aws

ユーザーガイド

Amazon Lightsail

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Lightsail: ユーザーガイド

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon の商標およびトレードドレスは Amazon 以外の製品およびサービスに使用することはできません。また、お客様に誤解を与える可能性がある形式で、または Amazon の信用を損なう形式で使用することもできません。Amazon が所有していないその他のすべての商標は Amazon との提携、関連、支援関係の有無にかかわらず、それら該当する所有者の資産です。

Table of Contents

Lightsail とは? ... 1
機能 ... 1
Lightsail の対象 ... 3
はじめに ... 3
関連サービス ... 3
見積もり、請求、コストの最適化 ... 4
セットアップする ... 5
にサインアップする AWS アカウント ... 5
管理アクセスを持つユーザーを作成する ... 5
Lightsail にアクセスする ... 8
Lightsail サービスインターフェイス ... 8
Lightsail サービスエンドポイント ... 9
IPv4 エンドポイント ... 9
デュアルスタック (IPv4 および IPv6) エンドポイント ... 10
リージョン別のサービスエンドポイント ... 10
エンドポイントを指定する例 ... 12
開始方法 ... 15
ステップ 1: 前提条件を満たす ... 15
ステップ 2: インスタンスを作成する ... 15
ステップ 3: インスタンスに接続する ... 17
ステップ 4: インスタンスにストレージを追加する ... 19
ステップ 5: スナップショットを作成する ... 20
ステップ 6: クリーンアップする ... 20
次の手順 ... 21
Lightsail で を使用する AWS CLI ... 21
前提条件 ... 22
SSH キーペアの生成 ... 22
インスタンスを作成して管理する ... 23
インスタンスへの接続 ... 27
インスタンスにストレージを追加する ... 29
スナップショットの作成と使用 ... 31
リソースをクリーンアップする ... 33
次の手順 ... 36
Lightsail リセラー ... 37

Lightsail 再販の利点 37
Lightsail リセラーの利点と増額されたデフォルトクォータがアカウントにどのように適用されるか 38
Lightsail リセラーになる方法 40
Lightsail リセラーになる 40
Lightsail リセラーになるために必要な情報 41
Lightsail リセラーになるためのリクエスト 41
Lightsail リセラーになる追加のアカウントをリクエストする 44
サービスクォータの引き上げ 46
リセラーとして Lightsail に連絡する 47
インスタンス 50
インスタンスを作成する 50
Linux インスタンス 51
Windows インスタンス 55
ブループリント 64
オペレーティングシステム 64
データベースアプリケーション 69
CMS アプリケーション 70
アプリケーションスタックおよびサーバー 72
e コマース アプリケーション 77
プロジェクト管理アプリケーション 77
バンドル 78
Linux/Unix バンドル (パブリック IPv4 アドレス機能付) 78
Linux/Unix バンドル (IPv6 のみ) 80
Windows バンドル (パブリック IPv4 アドレス機能付) 82
Windows バンドル (IPv6 のみ) 85
インスタンスのファイアウォール 87
Lightsail のファイアウォール 87
ファイアウォールルールを作成する 88
プロトコルを指定する 89
ポートの指定 90
アプリケーションレイヤーのプロトコルタイプを指定する 91
送信元 IP アドレスを指定する 93
Lightsail のデフォルトのファイアウォールルール 94
ファイアウォールルールを追加する 96
ファイアウォールルールを削除する 98

インスタンスのファイアウォールルール 99
バーストキャパシティとパフォーマンス 102
CPU パフォーマンス 103
バーストキャパシティの蓄積 106
インスタンスバーストを特定する 108
バーストキャパシティをモニタリングする 109
バーストキャパシティを表示する 111
CPU 使用率が高い場合のトラブルシューティング 114
インスタンス管理 115
インスタンスの開始、停止、または再起動 115
インスタンスを強制停止する 118
拡張ネットワーク 120
Lightsail で Windows Server ファイルシステムを拡張する 122
Linux シェルスクリプト 126
PowerShell スクリプト 128
Windows のセキュリティのベストプラクティス 130
インスタンスを削除する 135
Lightsail コンソールのホームページでインスタンスを削除する 135
Lightsail コンソールのインスタンス管理ページでインスタンスを削除する 136
を使用してインスタンスを削除する AWS CLI 137
次の手順 139
SSH およびインスタンスへの接続 140
キーペアオプションの選択 141
インスタンスに接続します 141
インスタンスに保存されているキーの管理 143
SSH キーの設定 143
SSH キーを管理 147
インスタンスの SSH キーの管理 162
Linux インスタンスに接続する 168
Windows インスタンスに接続する 188
インスタンスメタデータサービス 205
Instance Metadata Service を使う 205
IMDS 関連の追加のドキュメント 206
IMDS を設定する 207
Disks 214
ブロックストレージディスク 214

ディスククォータ 215
Linux インスタンスにディスクをアタッチする 215
ステップ 1: 新しいディスクを作成してインスタンスにアタッチする 215
ステップ 2: インスタンスに接続し、ディスクをフォーマットしてマウントする 217
ステップ 3: インスタンスを再起動するたびにディスクをマウントする 223
Windows インスタンスにディスクをアタッチする 223
ステップ 1: 新しいブロックストレージディスクを作成してインスタンスにアタッチする .. 224
ステップ 2: インスタンスに接続し、ブロックストレージディスクをオンラインにする 226
ステップ 3: ブロックストレージディスクを初期化する 228
ステップ 4: ディスクをファイルシステムでフォーマットする 230
ディスクをデタッチおよび削除する 232
前提条件 233
ディスクをデタッチおよび削除する 233
スナップショット 234
手動スナップショット 234
自動スナップショット 235
システムディスクのスナップショット 235
スナップショットからの新しいリソースの作成 235
スナップショットをコピーする 236
のスナップショットを Amazon EC2 にエクスポートする 236
スナップショットを削除する 236
自動スナップショット 237
自動スナップショットの制限 237
自動スナップショット保持 238
Lightsail コンソールを使用してインスタンスの自動スナップショットを有効または無効にする 238
を使用してインスタンスまたはブロックストレージディスクの自動スナップショットを有効または無効にする AWS CLI 240
スナップショット時間を変更する 244
自動スナップショットを削除する 249
自動スナップショットを保持する 253
Linux スナップショット 259
Windows スナップショットと sysprep 260
ステップ 1: Sysprep を実行する前にバックアップスナップショットを作成する 261
ステップ 2: Sysprep を使用してインスタンスに接続し、シャットダウンする 263
ステップ 3: Sysprep の実行後にスナップショットを作成する 265

次の手順 267
ブロックストレージディスクのスナップショットを作成する 267
スナップショットからディスクを作成します。 268
ステップ 1: ディスクスナップショットを検索して新しいディスクの作成を選択する 269
ステップ 2: ディスクスナップショットから新しいディスクを作成する 270
ルートボリュームのスナップショットを作成する 272
ステップ 1: 前提条件を満たす 273
ステップ 2: インスタンスのルートボリュームスナップショットを作成する 273
ステップ 3: スナップショットからブロックストレージディスクを作成し、インスタンスへアタッチする 275
ステップ 4: インスタンスからブロックストレージディスクにアクセスする 277
スナップショットからのインスタンスの作成 282
を使用して、スナップショットからより大きなリソースを作成する 285
前提条件 285
リソースを作成する 285
を使用してスナップショットからより大きなリソースを作成する AWS CLI 287
前提条件 287
ステップ 1: スナップショット名を取得する 287
ステップ 2: バンドルを選択する 287
ステップ 3: AWS CLI コマンドを記述して新しいインスタンスを作成する 291
次の手順 292
スナップショットを削除する 292
リージョンの間でスナップショットをコピーする 294
前提条件 294
のスナップショットをコピーする 294
次の手順 297
スナップショットを EC2 にエクスポートする 297
エクスポートした Lightsail スナップショットから Amazon EC2 リソースを作成する 299
Amazon EC2 インスタンスタイプを選択する 300
Amazon EC2 インスタンスに接続する 301
Amazon EC2 インスタンスを保護する 301
スナップショットをエクスポートする方法 302
エクスポートをモニタリングする 307
エクスポートしたスナップショットから EC2 インスタンスを作成する 308
エクスポートしたスナップショットから EBS ボリュームを作成する 317
Linux EC2 インスタンスに接続します。 319

Linux または Unix EC2 インスタンスを保護する 327
Windows EC2 インスタンスに接続する 336
Windows EC2 インスタンスを保護する 342
CloudFormation スタック 344
ドメインと DNS 347
ドメイン登録の仕組み 347
で登録できるドメインLightsail 348
ドメイン登録の料金 349
ドメインに関する追加情報 349
Lightsail での DNS 349
DNS の用語 350
Lightsail DNS ゾーンでサポートされる DNS レコードタイプ 352
DNS ゾーンの作成 354
DNS ゾーンを編集する 362
DNS ゾーンの削除 363
インターネットトラフィックのルーティング 363
ドメインをインスタンスにポイントする 366
ドメインをロードバランサーにポイントする 369
転送 DNS 管理 372
Route 53 を使用する 373
ドメインを登録する 377
Lightsail を使用して新しいドメインを登録する 378
ドメインの詳細 382
ドメイン名をフォーマットする 383
ドメイン名登録用のドメイン名をフォーマットする 383
DNS ゾーンとレコード用のドメイン名をフォーマットする 383
DNS ゾーンとレコードの名前でのアスタリスク (*) の使用 383
次の手順 385
R53 でドメインを管理する 385
ドメイン登録のステータスを表示する 386
別の登録への許可のない移管を防ぐためにドメインを口ックする 386
失効した、または削除されたドメインを復元する 386
ドメイン登録を移管する 386
ドメイン名の登録を削除する 387
登録に関する情報 387
言葉 388

ドメインの自動更新 388
登録者、管理者、テクニカル、請求先の連絡先 388
コンタクトのタイプ 389
姓名 389
組織 389
メール 390
電話 390
住所 1 390
住所 2 390
国 391
都道府県 391
市町村 391
郵便番号 391
プライバシー保護 391
登録更新 392
自動更新 392
ドメイン登録中のドメイン自動更新の設定 394
登録済みのドメイン自動更新の設定 394
プライバシー保護 394
の前提条件を満たす 395
ドメインのプライバシー保護を管理する 395
ドメイン連絡先情報の更新 395
ドメインの所有者は誰ですか。 396
ドメインの連絡先情報を更新 396
データベース 397
データベースの作成 397
次のステップ 401
データベースエンジンを選択する 402
Lightsail で管理されたデータベースを比較する 402
データのインポートを最適化する 403
高可用性データベース 404
MySQL に接続する 405
ステップ 1: MySQL データベース接続の詳細を取得する 405
ステップ 2: MySQL データベースのパブリック可用性を設定する 406
ステップ 3: MySQL データベースに接続するようにデータベースクライアントを設定する 407

次の手順 ... 409
SSL を使用して MySQL に接続する ... 409
サポートされている接続 ... 410
前提条件 ... 411
SSL を使用して MySQL データベースに接続する ... 411
PostgreSQL に接続する ... 413
ステップ 1: PostgreSQL データベース接続の詳細を取得する ... 413
ステップ 2: PostgreSQL データベースのパブリック可用性を設定する ... 414
ステップ 3: PostgreSQL データベースに接続するようにデータベースクライアントを設定する ... 415
次の手順 ... 417
SSL を使用して PostgreSQL に接続する ... 418
前提条件 ... 418
SSL を使用して Postgres データベースに接続する ... 418
データベースを削除する ... 419
データのインポートモード ... 421
SQL データをインポートする ... 422
データ PostgreSQL をインポートする ... 423
データベースログ ... 426
MySQL クエリログ ... 427
ポイントインタイムバックアップを無効にする ... 431
前提条件 ... 432
データベースポイントインタイムバックアップを無効にする ... 432
データベーススナップショット ... 433
次の手順 ... 435
データベースを復元する ... 435
スナップショットからデータベースを作成する ... 438
SSL 証明書をダウンロードします。 ... 442
すべての AWS リージョンの証明書バンドル ... 442
特定の AWS リージョンの証明書バンドル ... 442
CA 証明書を更新する ... 442
メンテナンスおよびバックアップ期間 ... 446
前提条件 ... 447
データベースのメンテナンスウィンドウを変更する ... 447
次の手順 ... 450
データベースのパスワードを管理する ... 451

次の手順 ... 452
パブリックモード ... 452
次のステップ ... 453
パラメータを更新する ... 454
前提条件 ... 454
使用可能なデータベースのパラメータのリストを取得します。 ... 454
データベースのパラメータを更新する ... 456
メジャーバージョンをアップグレードする ... 458
前提条件 ... 458
データベースのメジャーバージョンを更新する ... 459
次の手順 ... 462
MySQL 5.6 からの移行 ... 462
ステップ 1: 変更を確認する ... 463
ステップ 2: 前提条件を完了させる ... 463
ステップ 3: MySQL 5.6 データベースに接続してデータを工クスポートする ... 463
ステップ 4: MySQL 5.7 データベースに接続してデータをインポートする ... 468
ステップ 5: アプリケーションをテストして移行を完了する ... 470
ロードバランサー ... 472
ロードバランサーの機能 ... 472
ロードバランサーを使用するタイミング ... 473
ロードバランシングが推奨される アプリケーション ... 473
ロードバランサーの使用を開始する ... 474
ロードバランサーの作成 ... 474
前提条件 ... 474
ロードバランサーの作成 ... 474
インスタンスをロードバランサーにアタッチする ... 476
次の手順 ... 476
ロードバランサーの設定を更新する ... 477
ヘルスチェック ... 477
暗号化されたトラフィック (HTTPS) ... 478
セッション永続性 ... 478
インスタンスのロードバランシング ... 478
一般的なガイドライン: データベースを使用するアプリケーション ... 478
WordPress ... 479
Node.js ... 479
Magento ... 480

GitLab ... 480
Drupal ... 481
LAMP スタック ... 481
MEAN スタック ... 481
Redmine ... 482
Nginx ... 482
Joomla! ... 482
TLS のセキュリティポリシーを設定する ... 483
セキュリティポリシーの概要 ... 483
サポートされているセキュリティポリシーとプロトコル ... 483
前提条件を満たす ... 485
Lightsail コンソールを使用してセキュリティポリシーを設定する ... 485
を使用してセキュリティポリシーを設定する AWS CLI ... 485
HTTP から HTTPS へのリダイレクト ... 487
前提条件を満たす ... 487
Lightsail コンソールを使用してロードバランサーでの HTTPS リダイレクトを設定する ... 487
を使用してロードバランサーの HTTP から HTTPS へのリダイレクトを設定する AWS CLI ... 488
セッション永続性 ... 489
セッション永続性を有効にする ... 490
Cookie の有効期間を調整する ... 490
ヘルスチェック ... 491
ヘルスチェックのパスをカスタマイズする ... 492
ヘルスチェックメトリクス ... 493
ヘルスチェック ... 495
インスタンスのデタッチ ... 496
ロードバランサーを削除する ... 496
ディストリビューション ... 498
ユースケース ... 500
ディストリビューションを設定する ... 501
エッジロケーションと IP アドレス範囲 ... 503
ディストリビューションを作成する ... 503
前提条件 ... 504
オリジンリソース ... 505
オリジンプロトコルポリシー ... 506
キャッシュ動作とキャッシュプリセット ... 506

WordPress の最適キャッシュプリセット 507
デフォルトの 動作 508
ディレクトリとファイルの上書き 509
キャッシュの詳細設定 510
ディストリビューションプラン 513
ディストリビューションを作成する 514
次の手順 517
ディストリビューションを削除する 518
ディストリビューションを削除する 518
キャッシュの動作 518
キャッシュプリセット 519
WordPress の最適キャッシュプリセット 520
デフォルトの 動作 520
ディレクトリとファイルの上書き 521
キャッシュの詳細設定 522
ディストリビューションのキャッシュ動作を変更する 525
キャッシュのリセット 526
オリジンを変更する 526
オリジンプロトコルポリシー 527
ディストリビューションのオリジンを変更する 527
バケットをディストリビューションと共に使用する 529
ステップ 1: 前提条件を満たす 530
ステップ 2: バケットのアクセス許可を変更する 530
ステップ 3: オリジンとしてのバケットを持つディストリビューションを作成する 533
ステップ 4: ディストリビューションのカスタムサブドメインを有効にする 536
ステップ 5: WordPress ウェブサイトに WP Offload Media プラグインをインストールします。 536
ステップ 6: WordPress ウェブサイトとLightsail バケットとディストリビューションの接続をテストします。 543
バケットとオブジェクトを管理する 547
プラン変更 549
ディストリビューションプランを変更する 549
ディストリビューションカスタムドメイン 550
前提条件 550
ディストリビューションのカスタムドメインを有効にする 551
ドメインをディストリビューションにポイントする 552

カスタムドメインを変更する 554
ディストリビューションカスタムドメインを無効にする 555
コンテナサービスへディストリビューションのドメインを追加する 556
リクエストとレスポンスの動作 558
ディストリビューションがリクエストを処理してオリジンに転送する方法 559
ディストリビューションがオリジンからの応答を処理する仕組み 574
ディストリビューションのテスト 579
ディストリビューションをテスト 579
ネットワーク 581
ロードバランサー 581
静的 IP 581
IP アドレス 581
インスタンスのプライベートとパブリックの IPv4 アドレス 582
インスタンスの静的 IPv4 アドレス 584
インスタンス、コンテナーサービス、CDN ディストリビューション、およびロードバランサー用の IPv6 586
静的 IP アドレス 588
デュアルスタックネットワーク 594
IPv6 専用のネットワーク 598
リージョンとアベイラビリティーゾーン 602
Lightsail のリージョン 603
SSH キーと Lightsail リージョン 604
Lightsail リージョンを使用するためのヒント 604
Lightsail のアベイラビリティーゾーン 605
アベイラビリティーゾーンと Lightsail アプリケーション 605
リージョンの有効化 606
リージョンの無効化 608
VPC ピアリング 611
他の AWS サービスとの通信を許可する 612
SSL/TLS 証明書 612
HTTPS を使用する理由 613
プロセスの概要 613
ディストリビューションおよびコンテナサービスを使用した SSL/TLS 証明書を使用する .. 614
ロードバランサーでの SSL/TLS 証明書の使用 615
コンテナの証明書 616
ディストリビューション証明書 621

ロードバランサー証明書 633
逆引き DNS を設定する 643
前提条件 643
AWS Support に逆引き DNS の設定リクエストを送信する 644
バケット 646
オブジェクトストレージの概念 646
バケットとオブジェクトを管理する 648
バケットを作成する 649
バケットを作成する 650
バケットとオブジェクトを管理する 650
バケットの削除 653
バケットの強制削除 653
Lightsail コンソールを使用してバケットを削除する 653
を使用してバケットを削除する AWS CLI 654
バケットとオブジェクトを管理する 656
アクセスキーを作成する 658
バケットのアクセスキーを作成する 658
アクセスキーを削除します 659
バケットのアクセスキーを削除します 660
パブリックアクセスをブロックする 660
アカウントのブロックパブリックアクセス設定の構成 661
バケットとオブジェクトを管理する 664
バケットのアクセスログ 667
ログ配信を有効にするには何が必要ですか 667
ログオブジェクトのキーフォーマット 668
ログを配信する方法 668
ベストエフォート型のアクセスログ配信 668
バケットのログ記録ステータスの変更が有効になるまでには時間がかかる 669
アクセスログの形式 669
アクセスログを管理する 683
アクセスログの使用 687
バケットオブジェクト 692
Lightsail コンソールを使用してオブジェクトをフィルターする 692
を使用してオブジェクトを表示する AWS CLI 695
バケットとオブジェクトを管理する 697
オブジェクトをコピーまたは移動する 700

オブジェクトの削除 704
オブジェクトをダウンロードする 713
オブジェクトをフィルタリングする 717
オブジェクトのバージョニングを管理する 721
オブジェクトバージョンを復元する 727
オブジェクトをタグ付けする 732
バケットリソースアクセス 736
バケットのリソースアクセスの設定 737
バケットのプランを変更する 738
Lightsail コンソールを使用してバケットのストレージプランを変更する 738
を使用してバケットのストレージプランを変更する AWS CLI 739
アクセス許可を設定する 740
バケットのアクセス許可設定 741
クロスアカウントアクセス 742
バケットのクロスアカウントアクセスの設定 743
個々のオブジェクトのアクセス許可 743
個々のオブジェクトのアクセス許可の設定 744
マルチパートアップロード 746
マルチパートアップロードのプロセス 747
マルチパートアップロードの同時オペレーション 749
マルチパートアップロードの保持期間 750
Amazon シンプルストレージサービスのマルチパートアップロード制限 750
アップロードするファイルを分割します。 751
を使用してマルチパートアップロードを開始する AWS CLI 751
を使用してパートをアップロードする AWS CLI 752
AWS CLI を使用したマルチパートアップロードのパートのリスト化 753
マルチパートアップロード .json ファイルの作成 755
AWS CLI を使用したマルチパートアップロードの完了 757
AWS CLI を使用した、バケットのマルチパートアップロードのリスト化 758
AWS CLI を使用したマルチパートアップロードの停止 759
名前付けルール 760
バケット名の例 761
オブジェクトキー名 762
キー名 762
オブジェクトキーの命名のガイドライン 763
XML 関連のオブジェクトキーの制約 765

オブジェクトストレージのセキュリティのベストプラクティス ... 766
予防的セキュリティのベストプラクティス ... 767
モニタリングと監査のベストプラクティス ... 772
バケット許可 ... 773
バケットのアクセス許可 ... 775
個々のオブジェクトのアクセス許可 ... 775
クロスアカウントアクセス ... 776
アクセスキー ... 776
リソースアクセス ... 776
Amazon S3 パブリックアクセスブロック ... 777
バケットにファイルをアップロードする ... 777
オブジェクトキーの名前とバージョニング ... 778
Lightsail コンソールを使用してバケットにファイルをアップロードするには ... 778
を使用してバケットにファイルをアップロードする AWS CLI ... 779
IPv6 専用リクエスト用に AWS CLI を設定する ... 780
Lightsail でのバケットとオブジェクトの管理 ... 782
Lightsail の CORS ... 785
CORS のユースケース ... 785
Lightsail が CORS 設定を評価する方法 ... 785
AWS CLI を使用して CORS の設定 ... 787
CORS のトラブルシューティング ... 790
コンテナサービス ... 791
コンテナ ... 792
Lightsail コンテナサービスの要素 ... 792
Lightsail コンテナサービス ... 792
コンテナサービス容量 (スケールとパワー) ... 793
料金 ... 794
デプロイ ... 794
デプロイバージョン ... 795
コンテナイメージソース ... 796
コンテナサービス ARN ... 796
パブリックエンドポイントとデフォルトドメイン ... 797
カスタムドメインと SSL/TLS 証明書 ... 798
コンテナログ ... 798
メトリクス ... 798
Lightsail コンテナサービスを使用する ... 798

コンテナを作成する ... 800
コンテナサービス容量 (スケールとパワー) ... 800
料金 ... 801
コンテナサービスステータス ... 801
コンテナサービスの作成 ... 802
コンテナイメージ ... 805
ステップ 1: 前提条件を満たす ... 806
ステップ 2: Dockerfile を作成してコンテナイメージを構築する ... 806
ステップ 3: 新しいコンテナイメージを実行する ... 808
(オプション) ステップ 4: ローカルマシンで実行されているコンテナをクリーンアップする ... 809
コンテナイメージの作成後の次のステップ ... 810
コンテナイメージの管理 ... 810
コンテナーサービスプラグインをインストールする ... 814
ECR プライベートリポジトリへのアクセス ... 822
コンテナとデプロイを管理する ... 840
前提条件 ... 841
デプロイパラメータ ... 842
コンテナ間の通信 ... 846
コンテナログ ... 846
デプロイバージョン ... 847
デプロイのステータス ... 847
デプロイエラー ... 847
現在のコンテナサービスのデプロイの表示 ... 847
コンテナサービスのデプロイを作成または変更 ... 848
コンテナ容量を変更する ... 850
デプロイバージョンを管理する ... 851
コンテナログの表示 ... 853
コンテナサービスのカスタムドメイン ... 856
コンテナサービスのカスタムドメインの制限 ... 856
前提条件 ... 857
コンテナサービスのカスタムドメインの表示 ... 857
コンテナサービスのカスタムドメインを有効にする ... 858
コンテナサービスのカスタムドメインを無効化する ... 859
Lightsail ドメインをコンテナにポイントする ... 860
Route 53 ドメインをコンテナにポイントする ... 862

コンテナを削除する ... 867
コンテナサービスを削除 ... 868
セキュリティ ... 869
インフラストラクチャセキュリティ ... 869
耐障害性 ... 870
Identity and Access Management ... 870
オーディエンス ... 870
アイデンティティを使用した認証 ... 871
ポリシーを使用したアクセスの管理 ... 874
AWS マネージドポリシー ... 877
Lightsail のポリシーとロール ... 880
IAM ユーザーのアクセスを管理する ... 903
更新管理 ... 910
インスタンスブループリントソフトウェアのサポート ... 910
コンプライアンス検証 ... 911
AWS PrivateLink ... 912
考慮事項 ... 912
インターフェイスエンドポイントの作成 ... 912
AWS CLI の例 ... 913
エンドポイントポリシーを作成する ... 913
パフォーマンスをモニタリングする ... 915
リソースを効果的にモニタリングする ... 915
メトリクスの概念と用語 ... 916
メトリクス ... 916
メトリクスの保持 ... 916
統計 ... 916
単位 ... 917
期間 ... 917
アラーム ... 917
Lightsail で使用可能なメトリクス ... 918
インスタンスメトリクス ... 918
データベースメトリクス ... 919
ディストリビューションメトリクス ... 920
ロードバランサーのメトリクス ... 920
コンテナサービスのメトリクス ... 921
バケットメトリクス ... 922

リソースヘルスのメトリック 922
インスタンスメトリクス 922
データベースメトリクス 924
ディストリビューションメトリクス 924
ロードバランサーのメトリクス 925
コンテナサービスのメトリクス 926
バケットメトリクス 927
メトリクスの通知 927
のインスタンスメトリクスの表示 928
メトリクスのアラーム 933
インスタンスのアラームを作成する 944
アラームを削除または無効化する 949
バケットメトリクス 950
バケットメトリクス 951
Lightsail コンソールでのバケットメトリクスの表示 951
バケットとオブジェクトを管理する 952
アラームの作成 954
コンテナのメトリクス 958
コンテナサービスのメトリクス 959
Lightsail コンソールでコンテナサービスのメトリクスを表示する 959
データベースメトリクス 960
データベースメトリクス 960
Lightsail コンソールでのデータベースメトリクスの表示 961
データベースメトリクスの表示後の次のステップ 961
データベースアラームの作成 962
ディストリビューションメトリクス 967
ディストリビューションメトリクス 968
ディストリビューションメトリクスを Lightsail コンソールに表示する 968
ディストリビューションメトリクスの表示後の次のステップ 969
ディストリビューションにアラームを作成する 970
ロードバランサーのメトリクス 975
ロードバランサーのメトリクス 976
ロードバランサーメトリクスの表示 977
次の手順 978
ロードバランサーアラーム 978
通知連絡先を追加する 984

リージョンの通知の連絡先の制限 985
SMS テキストメッセージングのサポート 985
メールによる連絡先の確認 986
Lightsail コンソールを使用した通知連絡先の追加 986
を使用した通知連絡先の追加 AWS CLI 991
通知連絡先を追加した後の次の手順 993
通知連絡先を削除する 994
Lightsail コンソールを使用した通知連絡先の削除 994
を使用した通知連絡先の削除 AWS CLI 995
通知連絡先を削除した後の次の手順 995
Lightsail アラーム通知を確認する 996
アクティブなアラームのアラーム通知を確認する 996
検証保留中の E メール連絡先を確認する 996
タグ 998
タグを使用して請求を整理し、アクセスをコントロールする 998
タグ付けをサポートする Lightsail のリソース 999
タグの制限 1000
タグを追加 1000
次の手順 1002
タグの削除 1003
アクセス許可とタグに基づく承認 1004
タグを使用してアクセスを制御する 1004
ステップ 1: IAM ポリシーを作成する 1005
ステップ 2: ユーザーまたはグループにポリシーをアタッチする 1007
タグを使用したコストを整理する 1007
ステップ 1: キーと値のタグを リソースに追加する 1007
ステップ 2: ユーザー定義のコスト配分タグを有効にする 1008
ステップ 3: コスト配分レポートを設定して表示する 1008
タグを使用して、リソースを整理する 1008
リソースのタグを表示する 1009
タグを使用してリソースをフィルタ処理する 1010
トラブルシューティング 1011
WordPress の設定 1011
一般的なエラー 1012
設定失敗 1016
403 エラー (アクセス拒否) 1025

ブロックストレージディスク 1026
一般的なディスクエラー 1026
ブラウザベースの SSH または RDP クライアント 1028
エラーメッセージ: 接続できません 1028
エラーメッセージ: 現在接続できません。 1030
Ghost Service が使用できない 1031
Ghost サービスの開始 1031
IAM に関する問題 1034
Lightsail でアクションを実行する権限がない 1034
iam:PassRole を実行する権限がない 1035
アクセスキーを表示したい 1035
管理者として Lightsail へのアクセスを他のユーザーに許可したい 1036
自分の AWS アカウント以外のユーザーに自分のLightsailリソースへのアクセスを許可したい 1036
IPv6 到達可能性 1037
デュアルスタックインスタンスの IPv6 を有効にする 1037
インスタンスのファイアウォールを設定する 1039
インスタンスへの到達可能性をテストする 1040
インスタンス容量不足のエラー 1042
新しいインスタンスを起動するときの容量不足 1043
停止したインスタンスをスタートするときの容量不足 1043
関連情報 1044
ロードバランサー 1044
ロードバランサーの一般的なエラー 1044
通知 1045
SSL/TLS 証明書 1047
チュートリアル 1048
クイックスタートガイド 1049
AlmaLinux 1050
cPanel & WHM 1059
Drupal 1074
Ghost 1084
GitLab CE 1096
Joomla! 1108
LAMP 1121
Magento 1128

Nginx 1146
Node.js 1153
OpenClaw 1161
Plesk 1182
PrestaShop 1191
Redmine 1207
Ruby on Rails 1217
WordPress 1220
WordPress マルチサイト 1231
Bitnami 1240
Bitnami のユーザー名とパスワード 1241
Bitnami バナーを削除する 1248
WordPress 1252
WordPress の設定 1252
Amazon S3 に接続する 1267
Aurora DB に接続する 1277
MySQL に接続する 1285
ストレージバケットに接続する 1290
CDN を設定する 1306
メールを有効にする 1310
HTTPS を有効にする 1322
Lightsail に移行する 1333
WordPress マルチサイト 1342
WordPress Multisite : ブログをドメインとして追加する 1342
WordPress Multisite: ブログをサブドメインとして追加する 1349
WordPress Multisite: ドメインを定義 1353
Let's Encrypt 1356
LAMP の Let's Encrypt 証明書 1357
NGINX Let's Encrypt 証明書 1372
WordPress の Let's Encrypt 証明書 1399
IPv6 ネットワーク 1415
cPanel と WHM 用の IPv6 1416
GitLab 用 IPv6 1422
Nginx 用 IPv6 1425
Plesk の IPv6 1429
Lightsail の場合はAWS CLI 1433

ステップ 1: をインストールする AWS CLI 1433
ステップ 2: 新しいアクセスキーを作成する 1434
ステップ 3: を設定する AWS CLI 1435
次の手順 1435
AWS CloudShell 1436
永続ストレージ 1436
AWS リージョン 1436
を起動して使用する AWS CloudShell 1437
追加情報 1440
LAMP を起動して設定する 1440
ステップ 1: AWS にサインアップ 1441
ステップ 2: LAMP インスタンスを作成する 1441
ステップ 3: SSH 経由でインスタンスに接続し、LAMP インスタンスのアプリケーションパスワードを取得します。 1444
ステップ 4: LAMP インスタンス上にアプリケーションをインストールする 1445
ステップ 5: 静的 IP アドレスを作成して LAMP インスタンスにアタッチする 1445
ステップ 6: DNS ゾーンを作成し、ドメインを LAMP インスタンスにマッピングする 1447
次の手順 1448
LAMP インスタンスを Aurora データベースに接続する 1448
チュートリアル: Windows Server 2016 を起動して設定する 1453
ステップ 1: AWS にサインアップ 1454
ステップ 2: Lightsail で Windows Server 2016 インスタンスを作成する 1454
ステップ 3: RDP 経由で Windows Server 2016 インスタンスに接続する 1457
ステップ 4: 静的 IP アドレスを作成して Windows Server 2016 インスタンスにアタッチする 1458
ステップ 5: DNS ゾーンを作成し、ドメインを Windows Server 2016 インスタンスにマッピングする 1461
次の手順 1462
CloudTrail ロギング 1462
LightsailCloudTrailでの情報 1463
Lightsail ログファイルエントリの概要 1464
HAR ファイルを作成する 1464
ステップ 1: ブラウザで HAR ファイルを作成する 1465
ステップ 2: HAR ファイルを編集して機密情報を削除する 1467
ステップ 3: HAR ファイルをレビュー用に送信する 1467
Prometheus のインストール 1467

ステップ 1: 前提条件を満たす 1468
ステップ 2: Lightsail インスタンスにユーザーとローカルシステムディレクトリを追加する 1468
ステップ 3: Prometheus バイナリパッケージをダウンロードする 1469
ステップ 4: Prometheus を設定する 1472
ステップ 5: Prometheus をスタートする 1475
ステップ 6: Node Exporter をスタートする 1477
ステップ 7: Node Exporter データコレクタで Prometheus を設定する 1479
scp でファイルを転送する 1482
前提条件 1482
ステップ 1: プライベートキー (.pem) ファイルをローカルコンピュータに保存する 1482
ステップ 2: プライベートキーのアクセス許可を変更する 1484
ステップ 3: プライベートキーをインスタンスに転送する 1484
ステップ 4: Lightsail Linux インスタンスと Unix インスタンス間でファイルを安全に転送する 1486
他の AWS サービスと連携する 1487
仮想マシン (仮想プライベートサーバー) 1487
サーバーレスコンピューティング 1488
データベース 1489
ロードバランサー 1490
ビッグデータ 1491
Storage 1491
モニタリングとアラーム 1492
アプリケーションのデプロイ 1493
アプリケーションコンテナ 1493
セキュリティとユーザーサインイン 1494
ソース管理とアプリケーションライフサイクル管理 1494
キューとメッセージング 1495
ワークフロー 1496
ストリーミングアプリケーション 1496
AWS CloudFormation リソース 1497
Lightsail および CloudFormation テンプレート 1497
の詳細 CloudFormation 1497
Lightsail に関する追加情報 1498
ブログ 1498
チュートリアル 1501

動画 1503
料金 1505
Lightsail の請求明細を表示する 1505
請求の使用タイプ 1506
請求のリージョンコード 1508
よくある質問 1509
Lightsail について 1510
Amazon Lightsail とは? 1510
Lightsail の機能 1510
Lightsail で API を利用できますか? 1510
Lightsail にサインアップするには、どうすればよいですか? 1510
どの AWS リージョン でLightsail利用できますか? 1510
アベイラビリティーゾーンとは何ですか? 1511
Lightsail の Service Quotas とは何ですか? 1511
より詳細なヘルプを得るにはどうすればよいですか? 1511
請求とアカウント管理 1511
Lightsail プランの課金対象を教えてください。 1512
プランに対して課金されるのは、どのようなときですか? 1512
Lightsail インスタンスを無料で試すことはできますか？ 1512
Lightsail 無料体験はいつから始まりますか？ 1513
Lightsail マネージド型データベースの課金対象を教えてください。 1513
Lightsail マネージド型データベースを無料で試すことはできますか？ 1513
Lightsail ブロックストレージの課金対象を教えてください。 1513
Lightsail ロードバランサーの課金対象を教えてください。 1513
証明書管理の課金対象を教えてください。 1513
Lightsail 静的 IPv4 アドレスの費用を教えてください。 1513
Lightsail ドメインにはどれくらいの費用がかかりますか？ 1514
Lightsail の DNS 管理の課金対象を教えてください。 1514
Lightsail スナップショットの課金対象を教えてください。 1514
AWS アカウントを管理するにはどうすればよいですか? 1514
どのオプトインリージョンを有効または無効にするかについて、どのように管理すれば良いですか？ 1515
無効化されたオプトインリージョンのリソースはどうなりますか？ 1515
無効なオプトインリージョンのリソースを削除する方法を教えてください。 1515
Lightsail の法的な利用規約とは何ですか? 1515
Lightsail 請求の支払い方法を教えてください。 1515

データ転送 1516
インスタンスでプランのデータ転送許容量を超えた場合はどうなりますか？ 1516
インスタンスにはどの種類のデータ転送が課金されますか? 1516
インスタンスにおけるデータ転送枠はどのように機能しますか? 1517
インスタンスのデータ転送許容量は、 AWS リージョンによってどれぐらい異なりますか？ 1518
データ転送の課金対象を教えてください。 1519
データ転送枠はロードバランサーではどのように機能しますか? 1519
オブジェクトストレージでのデータ転送許容量の仕組みを教えてください。 1519
ディストリビューションにはどの種類のデータ転送が課金されますか? 1520
Lightsail のインスタンスのデータ転送クォータとディストリビューションのデータ転送クォータの違いについて教えてください。 1520
コンテナサービスとのデータ転送は課金されますか? 1521
ブロックストレージ (ディスク) 1521
Lightsail ブロックストレージでどのようなことができますか? 1521
アタッチ済みディスクは Lightsail プランに含まれているストレージとどのように異なりますか? 1522
アタッチ済みディスクの容量は、どれくらいまで増やせますか? 1522
Lightsail インスタンスごとにアタッチできるディスクの数を教えてください。 1522
1 台のディスクを複数のインスタンスにアタッチすることはできますか? 1522
ディスクはインスタンスにアタッチする必要がありますか? 1523
アタッチ済みディスクの容量を拡張することはできますか? 1523
Lightsail ブロックストレージは暗号化を提供しますか? 1523
Lightsail ブロックストレージにはどのような可用性を期待できましか? 1523
アタッチ済みディスクをバックアップするには、どうすればよいですか? 1523
証明書 1524
Lightsail によってプロビジョニングされた証明書はどのように使用できますか? 1524
証明書を認証するには、どうすればよいですか? 1524
ドメインを認証できない場合はどうなりますか? 1524
証明書に追加できるドメインおよびサブドメインの数を教えてください。 1524
証明書に関連付けられたドメインを変更するには、どうすればよいですか? 1524
証明書を更新するには、どうすればよいですか? 1525
ロードバランサーを削除すると、証明書はどうなりますか? 1525
Lightsail によって提供された証明書をダウンロードできますか? 1525
連絡先とモニタリング通知 1525
通知とは何ですか? 1525

連絡先はいくつ追加できますか? 1525
コンテナサービス 1526
Lightsail コンテナサービスでできることは何ですか? 1526
Lightsail コンテナサービスは Docker コンテナを実行できますか? 1526
公開コンテナイメージを Lightsail コンテナサービスで使うには、どうすればよいですか? 1526
プライベートコンテナレジストリからコンテナイメージをプルできますか? 1526
需要に応じてサービスのパワーとスケールを変更することはできますか? 1526
Lightsail コンテナサービスによって作成された HTTPS エンドポイントの名前をカスタマイズすることはできますか? 1526
Lightsail コンテナサービスの HTTPS エンドポイントにカスタムドメインを使用できますか? 1527
Lightsail コンテナサービスの料金を教えてください。 1527
コンテナサービスを数日しか実行なくても、1 か月分が請求されますか? 1527
コンテナサービスの停止と削除の違いは何ですか? 1527
コンテナサービスが無効状態でも、課金されますか? 1528
コンテナサービスを Lightsail コンテンツ配信ネットワーク (CDN) ディストリビューションのオリジンとして使用することはできますか? 1528
コンテナサービスを Lightsail ロードバランサーのターゲットとして使用できますか? 1528
HTTP リクエストを HTTPS にリダイレクトするよう、コンテナサービスのパブリックエンドポイントを設定できますか? 1528
コンテナサービスはモニタリングととアラートをサポートしていますか? 1528
Lightsail コンテナサービスは IPv6 をサポートしていますか? 1529
コンテンツ配信ネットワークディストリビューション 1529
Lightsail CDN ディストリビューションでできることは何ですか? 1529
ディストリビューションのオリジンとして、どのような種類のリソースを使用できますか? 1529
インスタンスを Lightsail ディストリビューションのオリジンに設定する場合、静的 IPv4 アドレスを Lightsail インスタンスにアタッチする必要はありますか ? 1529
WordPress のウェブサイトで Lightsail ディストリビューションを設定するには、どうすればよいですか ? 1530
複数のオリジンをアタッチできますか? 1530
Lightsail ディストリビューションは証明書の作成をサポートしていますか ? 1530
証明書は必要ですか? 1530
作成できる証明書の数に制限はあります? 1530
HTTP リクエストを HTTPS にリダイレクトするようにディストリビューションを設定するには、どうすればよいですか? 1530

apex ドメインが Lightsail ディストリビューションを指すように設定するには、どうすればよいですか？ 1531
ディストリビューションと関連付いているプランを変更することはできますか? 1531
自分のディストリビューションが機能しているかどうか、どうすれば分かりますか? 1531
Lightsail ディストリビューションのキャッシュされたコンテンツを削除できますか？ 1531
どのような場合に、Amazon CloudFront ディストリビューションではなく Lightsail ディストリビューションを使うべきですか？ 1531
Lightsail コンテンツ配信ネットワークディストリビューションを Amazon CloudFront に移動することはできますか? 1532
Lightsail CDN はどのような目的で使用されますか？ 1533
Lightsail CDN ディストリビューションは IPv6 をサポートしていますか? 1533
Lightsail CDN ディストリビューションで動作させるには、オリジンを IPv6 有効にする必要がありますか? 1533
データベース 1533
Lightsail マネージド型データベースとは何ですか？ 1533
Lightsail マネージド型データベースでできることは何ですか？ 1534
Lightsail の管理対象を教えてください。 1534
Lightsail がサポートしているデータベースの種類とバージョンを教えてください。 1535
Lightsail はどのようなマネージド型データベースプランを提供していますか？ 1535
高可用性プランとは何ですか? 1535
Lightsail マネージド型データベースをスケールアップおよびスケールダウンするには、どうすればよいですか？ 1535
Lightsail マネージド型データベースをバックアップするには、どうすればよいですか？ .. 1536
Lightsail マネージド型データベースを削除すると、データはどうなりますか？ 1536
異なるアベイラビリティーゾーン AWS リージョン または異なるアベイラビリティーゾーンで実行されているLightsailマネージドデータベースにインスタンス (複数可) を接続することはできますか? 1536
Lightsail マネージド型データベースにデータをロードするには、どうすればよいですか？ 1537
Lightsail マネージド型データベースのデータにアクセスするには、どうすればよいですか？ 1537
Lightsail マネージド型データベースは Lightsail インスタンスとどのように連携しますか? 1537
AWS アカウントで実行されている EC2 インスタンスにLightsailマネージドデータベースを接続するにはどうすればよいですか? 1537
Lightsail マネージド型データベースのパブリックモードとプライベートモードの違いは何ですか？ 1538

Lightsail マネージド型データベースが使用するポートを管理できますか？ 1538
Lightsail マネージドデータベースサービスは IPv6 をサポートしていますか? 1538
ドメイン 1538
Lightsail ドメインでは何ができますか？ 1538
どの最上位ドメイン (TLD) を使用できますか? 1538
Lightsail を既存ドメインの DNS サービスにできますか？ 1539
Lightsail のドメイン登録の使用を開始するには、どうすればよいですか？ 1539
どのような場合に Route 53 ではなく Lightsail にドメインを登録すればよいですか？ 1539
ドメインを Lightsail に移管できますか？ 1539
ドメインではどの Lightsail リソースを使用できますか？ 1539
リソースを Amazon EC2 にエクスポートする 1540
Amazon EC2 へのエクスポートとは 1540
Amazon EC2 にエクスポートする理由は何ですか? 1540
Amazon EC2 へのエクスポートの仕組みを教えてください。 1540
どのように請求されますか? 1541
マネージドデータベースやディスクのスナップショットはエクスポートできますか? 1541
どのような Lightsail リソースをエクスポートできますか？ 1541
インスタンス 1541
Lightsail インスタンスとは何ですか? 1541
Lightsail プランとは何ですか? 1542
インスタンスでは何のソフトウェアが実行できますか? 1542
Lightsail は、どのオペレーティングシステムで使用できますか? 1542
Lightsail インスタンスを使用するには自分のライセンスを持ち込む必要がありますか？ .. 1542
Lightsail インスタンスを作成するには、どうすればよいですか? 1542
Lightsail インスタンスは、どのように動作しますか? 1543
インスタンスがバーストしているかどうか、どうやって確認できますか? 1543
Lightsail インスタンスへ接続するには、どうすればよいですか? 1543
インスタンスをバックアップするには、どうすればよいですか? 1544
プランをアップグレードできますか? 1544
AWS アカウントの他のリソースにLightsailインスタンスを接続するにはどうすればよいですか? 1544
インスタンスの停止と削除の違いは何ですか? 1544
ロードバランサー 1545
Lightsail ロードバランサーでどのようなことができますか? 1545
異なるアベイラビリティーゾーン AWS リージョン または異なるアベイラビリティーゾーンのインスタンスでロードバランサーを使用できますか? 1545

Lightsail ロードバランサーはトラフィックの急増にどのように対応しますか? 1546
Lightsail ロードバランサーはターゲットインスタンスにどのようにトラフィックをルーティングしますか? 1546
Lightsail ロードバランサーはターゲットインスタンスが正常かどうかをどのように判別しますか? 1546
ロードバランサーにアタッチできるインスタンスの数を教えてください。 1546
1 つのインスタンスを複数のロードバランサーに割り当てることはできますか? 1547
ロードバランサーを削除すると、ターゲットインスタンスはどうなりますか? 1547
セッション永続性とは何ですか? 1547
Lightsail ロードバランサーではどのような種類の接続がサポートされますか? 1547
Lightsail ロードバランサーは IPv6 をサポートしていますか? 1547
IPv6 が有効になっているロードバランサーを使うには、そのロードバランサーの背後にあるインスタンスの IPv6 も有効になっている必要がありますか? 1548
スナップショット 1548
スナップショットとは何ですか? 1548
自動スナップショットとは何ですか? 1548
手動スナップショットと自動スナップショットの違いは何ですか? 1548
どのようなリソースがスナップショットをサポートしていますか? 1549
スナップショットはどれくらいの期間保存できますか? 1549
自動スナップショットを有効にするには、どうすればよいですか? 1549
自動スナップショットはいつ作成されますか? 1549
保存できるスナップショットの数を教えてください。 1550
スナップショットはどのように課金されますか? 1550
自動スナップショットを無効にすると、スナップショットは失われますか? 1550
自動スナップショットが置き換えられないようにする場合は、どうすればよいですか? ... 1550
自動スナップショットは削除できますか? 1550
スナップショットはどのように使用できますか? 1550
メトリクスおよびアラーム 1551
メトリクスとは何ですか? 1551
アラームとは何ですか? 1551
アラームはいくつ追加できますか? 1551
ネットワーク 1551
Lightsail で IP アドレスをどのように使用しますか ? 1551
Lightsail は IPv6 のみのインスタンスをサポートしていますか? 1552
静的 IP とは何ですか? 1552
インスタンスには静的 IP をいくつアタッチできますか? 1552

DNS レコードとは何ですか? 1552
インスタンスのファイアウォール設定を管理することはできますか? 1553
オブジェクトストレージ (バケット) 1553
Lightsail オブジェクトストレージではどのようなことができますか? 1553
Lightsail オブジェクトストレージのコストについて教えてください。 1553
Lightsail バケットに関連付けられているプランを変更することはできますか? 1553
Lightsail オブジェクトストレージから Amazon S3 にオブジェクトをコピーできますか? . 1554
Lightsail オブジェクトストレージの使用を開始するにはどうすればよいですか? 1554
バケットにオブジェクトをアップロードするにはどうすればよいですか? 1554
バケットへのパブリックアクセスをブロックできますか? 1554
バケットにプログラムによるアクセスを追加するにはどうすればよいですか? 1555
バケットを他の AWS アカウントと共有するにはどうすればよいですか? 1555
バージョニングとは何ですか? 1555
Lightsail バケットを Lightsail CDN ディストリビューションに関連付けるにはどうすればよいですか? 1555
Lightsail オブジェクトストレージサービスにはどのような制限がありますか? 1556
Lightsail オブジェクトストレージはモニタリングとアラートをサポートしていますか? ... 1556
Lightsail のタグ 1556
タグとは 1556
Lightsail でタグを使用するには、どうすればよいですか ? 1556
タグ付けできるのはどのようなリソースですか? 1557
Lightsail スナップショットにタグを付けるには、どうすればよいですか ? 1557
キー値タグとキーのみタグの違いは何ですか? 1558
ドキュメント履歴 1559
ヘルプの表示 1561
コンテキスト依存のヘルプパネル 1561
ユーザーガイドについて 1561
検索の使用 1562
Lightsail の CLI および API の使用 1562
AWS フォーラムおよびその他のコミュニティリソース 1562
..... mdlxiii

Amazon Lightsail とは?

Amazon Lightsail は、ウェブサイトやウェブアプリケーションを構築する必要がある開発者向けの Amazon Web Services (AWS)を開始する最も簡単な方法です。プロジェクトをすばやく開始するためのすべてが揃っています。インスタンス (仮想プライベートサーバー)、コンテナサービス、マネージドデータベース、コンテンツ配信ネットワーク (CDN) ディストリビューション、ロードバランサー、SSD ベースのブロックストレージ、静的 IP アドレス、登録済みドメインの DNS 管理、リソーススナップショット (バックアップ) が、1 か月単位の予測可能な低価格で提供されます。

Lightsail には研究用の Amazon Lightsail もあります。Lightsail for Research を使用すると、学者や研究者は、強力な仮想コンピュータを AWS クラウド内で作成できます。これらの仮想コンピュータには、RStudio や Scilab などの研究用アプリケーションがプリインストールされています。詳細については、「Amazon Lightsail for Research User Guide」を参照してください。

トピック

- Lightsail の機能
- Lightsail の対象
- Lightsail の開始方法
- 関連サービス
- 見積もり、請求、コストの最適化

Lightsail の機能

Lightsail には次の高度な機能があります:

インスタンス

Lightsail は、 のパワーと信頼性によって簡単にセットアップおよびバックアップできる仮想プライベートサーバー (インスタンス) を提供します AWS。ウェブサイト、ウェブアプリケーション、またはプロジェクトを数分で起動し、直感的な Lightsail コンソールまたは API からインスタンスを管理できます。

インスタンスを作成するときは、WordPress、Windows、Plesk、LAMP、Nginx などのシンプルなオペレーティングシステム (OS)、事前設定されたアプリケーション、または開発スタックをクリックして起動します。すべての Lightsail インスタンスには、ソース IP、ポート、プロトコルに基づいてインスタンスへのトラフィックを許可または制限できる組み込みのファイアウォールが付属しています。詳細はこちら

コンテナ

クラウドでコンテナ化されたアプリケーションを実行し、安全にアクセスします。コンテナは、コードとその依存関係をパッケージ化するソフトウェアのスタンダード単位で、アプリケーションが 1 つのコンピューティング環境から別のコンピューティング環境に迅速かつ確実に実行します。詳細はこちら

ロードバランサー

インスタンス間でウェブトラフィックをルーティングし、トラフィックの変動に対応、障害から保護し、シームレスな訪問者体験を提供します。詳細はこちら

マネージド型データベース

Lightsail は、メモリ、処理、ストレージ、転送の許容量を含む、完全に設定された MySQL または PostgreSQL データベースプランを提供します。Lightsail のマネージドデータベースを使用すると、仮想サーバーとは独立してデータベースを簡単にスケールさせたり、アプリケーションの可用性を向上させたり、クラウドでスタンドアロンデータベースを運用できます。詳細はこちら

ブロックとオブジェクトのストレージ

Lightsail は、ブロックストレージとオブジェクトストレージの両方を提供します。Linux または Windows 仮想サーバー用の高可用性 SSD バックアップストレージを使用すると、ストレージを迅速かつ簡単にスケールできます。詳細はこちら

Lightsail オブジェクトストレージサービスを使用して、インターネット上のどこからでも、いつでもオブジェクトの保存と取得が可能です。クラウドで静的コンテンツをホストすることもできます。詳細はこちら

CDN ディストリビューション

Lightsail は、Amazon CloudFront と同じインフラストラクチャ上に構築されたコンテンツ配信ネットワーク (CDN) ディストリビューションを有効にします。世界中にプロキシサーバーを設定することで、コンテンツをグローバルなユーザーに簡単に配信できます。これにより、ユーザーは地理的に近い場所からウェブサイトにアクセスでき、遅延が軽減されます。詳細はこちら

AWS サービスへのアクセス

Lightsail は、インスタンス、マネージド型データベース、ロードバランサーといった限定された機能セットを提供し、簡単に始められるように設計されています。ただし、これらのオプションに制限されているわけではありません。Amazon VPC ピアリング AWS を通じて、Lightsailプロジェクトを の 90 以上の他のサービスの一部と統合できます。詳細はこちら

Lightsail の詳細については、「Amazon Lightsail」を参照してください。

Lightsail の対象

Lightsail は全ユーザーが対象です。開発プロジェクトをすばやく開始する Lightsail インスタンスのイメージを選択できるため、ソフトウェアやフレームワークのインストールに時間を費やす必要はありません。

個人プロジェクトに携わっている個々の開発者または愛好家にとっては、Lightsail は基本的なクラウドリソースのデプロイおよび管理に役立ちます。仮想マシン、ドメイン、ネットワーキングなどのクラウドサービスの学習や試用に関心があるユーザーにとっては、Lightsail から開始するのが最も簡単です。

Lightsail には、基本オペレーティングシステム、LAMP、LEMP (Nginx)、SQL Server Express などの開発スタック、および WordPress、Drupal、Magento などのアプリケーションが含まれているイメージがあります。各イメージにインストールされているソフトウェアの詳細については、「Lightsail インスタンスのイメージの選択」を参照してください。

プロジェクトの成長に合わせて、ブロックストレージディスクを追加し、Lightsail インスタンスにアタッチすることができます。これらのインスタンスとディスクのスナップショットを作成すると、それらのスナップショットから新しいインスタンスを簡単に作成できます。VPC をピアリングして、Lightsailインスタンスが の外部で他の AWS リソースを使用できるようにすることもできます Lightsail。

Lightsail ロードバランサーを作成し、ターゲットインスタンスをアタッチして可用性の高いアプリケーションを作成することもできます。暗号化された (HTTPS) トラフィック、セッション永続性、ヘルスチェックなどを処理できるように、ロードバランサーを設定することもできます。

Lightsail の開始方法

Lightsail を使用するように設定したら、Lightsail での仮想プライベートサーバーの使用を開始する方法 を通じてインスタンスを起動、接続、クリーンアップできます。Lightsail のアクセス方法の詳細については、「Lightsail にアクセスする」を参照してください。

関連サービス

インスタンスやディスクなど、Lightsail のリソースは Lightsail を使用して直接プロビジョニングできます。さらに、次のような他の AWS サービスを使用してリソースをプロビジョニングできます。

- Amazon EC2

  ユーザーがソフトウェアシステムを構築しホストするための、自在に拡張および縮小できるコンピューティング能力 (実際には Amazon のデータセンター内のサーバー) です。Lightsail と Amazon EC2 を比較するには、「Amazon Lightsail または Amazon EC2」を参照してください。
- Amazon EC2 Auto Scaling

  アプリケーションの負荷を処理するために適切な数の Amazon EC2 インスタンスがあることを確認できます。
- Elastic Load Balancing

  アプリケーションの着信トラフィックを複数の インスタンスに自動的に分散できます。
- Amazon Relational Database Service (Amazon RDS)

  クラウド内でマネージドリレーショナルデータベースを簡単に設定、運用、およびスケールできます。
- Amazon Elastic Container Service (Amazon ECS)

  コンテナ化されたアプリケーションを Amazon EC2 インスタンスのクラスターにデプロイ、管理、スケールできます。

見積もり、請求、コストの最適化

AWS ユースケースの見積りを作成するには、 を使用しますAWS 料金見積りツール。

請求を表示するにはAWS Billing and Cost Management コンソールで請求およびコスト管理ダッシュボードに移動します。請求書には料金の明細が記載された使用状況レポートへのリンクが記載されています。 AWS アカウントの請求の詳細については、AWS 「請求とコスト管理ユーザーガイド」を参照してください。

AWS 請求、アカウント、イベントについてご質問がある場合は、 AWS サポートにお問い合わせください。

を使用して、 AWS 環境のコスト、セキュリティ、パフォーマンスを最適化できますAWS Trusted Advisor。

の設定 AWS アカウント と管理ユーザー Lightsail

新規の AWS お客様は、 の使用を開始する前に、このページに記載されているセットアップの前提条件を完了してくださいAmazon Lightsail。セットアップ手順には、 AWS Identity and Access Management (IAM) サービスを使用します。IAM の詳細については、「IAM ユーザーガイド」を参照してください。

にサインアップする AWS アカウント

がない場合は AWS アカウント、次の手順を実行して作成します。

にサインアップするには AWS アカウント

1. https://portal.aws.amazon.com/billing/signup を開きます。
2. オンラインの手順に従います。

   サインアップ手順の一環として、電話またはテキストメッセージを受け取り、電話キーパッドで検証コードを入力します。

   にサインアップすると AWS アカウント、 AWS アカウントのルートユーザー が作成されます。ルートユーザーには、アカウントのすべての AWS のサービス とリソースへのアクセス権があります。セキュリティベストプラクティスとして、ユーザーに管理アクセス権を割り当て、ルートユーザーアクセスが必要なタスクの実行にはルートユーザーのみを使用するようにしてください。

AWS サインアッププロセスが完了すると、 から確認メールが送信されます。https://aws.amazon.com/ の [マイアカウント] をクリックして、いつでもアカウントの現在のアクティビティを表示し、アカウントを管理することができます。

管理アクセスを持つユーザーを作成する

にサインアップしたら AWS アカウント、日常的なタスクにルートユーザーを使用しないように AWS アカウントのルートユーザー、 を保護し AWS IAM アイデンティティセンター、 を有効にして管理ユーザーを作成します。

を保護する AWS アカウントのルートユーザー

1. ルートユーザーを選択し、 AWS アカウント E メールアドレスを入力して、アカウント所有者AWS マネジメントコンソールとして にサインインします。次のページでパスワードを入力します。

   ルートユーザーを使用してサインインする方法については、「AWS サインイン ユーザーガイド」の「ルートユーザーとしてサインインする」を参照してください。
2. ルートユーザーの多要素認証 (MFA) を有効にします。

   手順については、IAM ユーザーガイドの AWS アカウント 「ルートユーザー (コンソール) の仮想 MFA デバイスを有効にする」を参照してください。

管理アクセスを持つユーザーを作成する

1. IAM アイデンティティセンターを有効にします。

   手順については、「AWS IAM アイデンティティセンター ユーザーガイド」の「AWS IAM アイデンティティセンターの有効化」を参照してください。
2. IAM アイデンティティセンターで、ユーザーに管理アクセスを付与します。

   を ID ソース IAM アイデンティティセンターディレクトリ として使用する方法のチュートリアルについては、「 AWS IAM アイデンティティセンター ユーザーガイド」の「デフォルトを使用してユーザーアクセスを設定する IAM アイデンティティセンターディレクトリ」を参照してください。

管理アクセス権を持つユーザーとしてサインインする

- IAM アイデンティティセンターのユーザーとしてサインインするには、IAM アイデンティティセンターのユーザーの作成時に E メールアドレスに送信されたサインイン URL を使用します。

  IAM Identity Center ユーザーを使用してサインインする方法については、AWS サインイン「 ユーザーガイド」の AWS 「 アクセスポータルにサインインする」を参照してください。

追加のユーザーにアクセス権を割り当てる

1. IAM アイデンティティセンターで、最小特権のアクセス許可を適用するというベストプラクティスに従ったアクセス許可セットを作成します。

手順については、「AWS IAM アイデンティティセンター ユーザーガイド」の「アクセス許可セットを作成する」を参照してください。

2. グループにユーザーを割り当て、そのグループにシングルサインオンアクセス権を割り当てます。

   手順については、「AWS IAM アイデンティティセンター ユーザーガイド」の「グループを追加する」を参照してください。

Lightsail にアクセスする

さまざまなインターフェイスおよびサービスエンドポイントを使用して Amazon Lightsail にアクセスできます。

トピック

- Lightsail サービスインターフェイス
- Lightsail サービスエンドポイント
- エンドポイントを指定する例

Lightsail サービスインターフェイス

次のインターフェイスを使用して Lightsail リソースの作成および管理ができます。

Amazon Lightsail コンソール

Lightsail インスタンスおよびリソースを作成、管理するためのシンプルなウェブインターフェイス。 AWS アカウントにサインアップしている場合は、Lightsailコンソールに直接アクセスするか、 にサインイン AWS マネジメントコンソール して、コンソールのホームページLightsailから選択できます。

AWS Command Line Interface

コマンドラインシェルのコマンドを使用して AWS サービスとやり取りできます。Windows、Mac、Linux でサポートされています。の詳細については AWS CLI、AWS Command Line Interface 「 ユーザーガイド」を参照してください。コマンドは、 CLI Lightsail コマンドリファレンスの セクションにあります。 Lightsail AWS

AWS CloudShell

CloudShell は、 から直接起動できるブラウザベースの事前認証済みシェルです AWS マネジメントコンソール。Bash、PowerShell、Z シェルなどの任意のシェルを使用して AWS CLI コマンドを実行できます。 AWS CloudShell を使用して Lightsailリソースを管理する方法の例については、「」を参照してくださいでLightsailリソースを管理する AWS CloudShell。

Query API

Lightsail はクエリ API を提供します。このリクエストはHTTP 動詞 (GET または POST) とクエリパラメータ `Action` で記述する HTTP または HTTPS リクエストです。Lightsail の API アク

ションの詳細については、Amazon Lightsail API リファレンスのアクションを参照してください。

AWS SDKs

HTTP または HTTPS 経由でリクエストを送信する代わりに言語固有の APIs を使用してアプリケーションを構築する場合、 はソフトウェア開発者向けにライブラリ、サンプルコード、チュートリアル、その他のリソース AWS を提供します。これらのライブラリにはリクエストの暗号化署名、リクエストの再試行、エラーレスポンスの処理などのタスクを自動化する基本機能が用意されているので、開発を簡単に始められます。詳細については、「AWSでの構築ツール」を参照してください。

AWS Tools for PowerShell

SDK for .NET から公開されている機能に基づいて構築された PowerShell モジュールのセットです。Tools for PowerShell を使用すると、PowerShell コマンドラインから AWS リソースに対するオペレーションをスクリプト化できます。使用を開始する方法については『AWS Tools for PowerShell ユーザーガイド』を参照してください。「AWS Tools for PowerShell Cmdlet Reference」で、Lightsail のコマンドレットを確認できます。

Lightsail サービスエンドポイント

エンドポイントは、 AWS ウェブサービスのエントリポイントとして機能する URL です。前述のインターフェイスのプログラム的なアクセス方法の場合、Lightsail は次のエンドポイントの種類をサポートします。

- IPv4 エンドポイント
- デュアルスタックのエンドポイント (IPv4 と IPv6 の両方をサポート)

リクエストを行うと、使用するエンドポイントを指定できます。エンドポイントを指定しない場合、デフォルトで IPv4 エンドポイントが使用されます。別のエンドポイントタイプを使用するには、リクエストで指定する必要があります。

IPv4 エンドポイント

IPv4 エンドポイントは IPv4 トラフィックのみをサポートします。IPv4 エンドポイントは、全リージョンで利用できます。リージョンのサービスエンドポイントの詳細については、「リージョン別のサービスエンドポイント」を参照してください。

IPv4 エンドポイント名では、次の命名規則が使用されます。

- *service*.*region*.amazonaws.com

例えば、us-east-2 リージョンの IPv4 エンドポイントは、lightsail.us-east-2.amazonaws.com です。

デュアルスタック (IPv4 および IPv6) エンドポイント

デュアルスタックエンドポイントは、IPv4 と IPv6 トラフィックの両方をサポートします。デュアルスタックエンドポイントにリクエストを行うと、エンドポイント URL は、ネットワークとクライアントが使用するプロトコルに応じて IPv6 または IPv4 アドレスに解決されます。

- lightsail.*region*.api.aws

例えば、us-east-2 リージョンのデュアルスタックエンドポイント名は、lightsail.us-east-2.api.aws です。

リージョン別のサービスエンドポイント

次の内容は、Lightsail のサービスエンドポイントです。Lightsail が利用できるリージョンの詳細については、「Lightsail のリージョンとアベイラビリティーゾーン」を参照してください。

リージョン名	リージョン	エンドポイント	プロトコル	
米国東部 (オハイオ)	us-east-2	lightsail.us-east-2.amazonaws.com	HTTPS	
		lightsail.us-east-2.api.aws	HTTPS	
米国東部 (バージニア北部)	us-east-1	lightsail.us-east-1.amazonaws.com	HTTPS	
		lightsail.us-east-1.api.aws	HTTPS	
米国西部 (オレゴン)	us-west-2	lightsail.us-west-2.amazonaws.com	HTTPS	
		lightsail.us-west-2.api.aws	HTTPS	

リージョン名	リージョン	エンドポイント	プロトコル
アジアパシフィック (ジャカルタ)	ap-southeast-3	lightsail.ap-southeast-3.amazonaws.com lightsail.ap-southeast-3.api.aws	HTTPS HTTPS
アジアパシフィック (ムンバイ)	ap-south-1	lightsail.ap-south-1.amazonaws.com lightsail.ap-south-1.api.aws	HTTPS HTTPS
アジアパシフィック (ソウル)	ap-northeast-2	lightsail.ap-northeast-2.amazonaws.com lightsail.ap-northeast-2.api.aws	HTTPS HTTPS
アジアパシフィック (シンガポール)	ap-southeast-1	lightsail.ap-southeast-1.amazonaws.com lightsail.ap-southeast-1.api.aws	HTTPS HTTPS
アジアパシフィック (シドニー)	ap-southeast-2	lightsail.ap-southeast-2.amazonaws.com lightsail.ap-southeast-2.api.aws	HTTPS HTTPS
アジアパシフィック (東京)	ap-northeast-1	lightsail.ap-northeast-1.amazonaws.com lightsail.ap-northeast-1.api.aws	HTTPS HTTPS
カナダ (中部)	ca-central-1	lightsail.ca-central-1.amazonaws.com lightsail.ca-central-1.api.aws	HTTPS HTTPS
欧州 (フランクフルト)	eu-central-1	lightsail.eu-central-1.amazonaws.com lightsail.eu-central-1.api.aws	HTTPS HTTPS

リージョン名	リージョン	エンドポイント	プロトコル
欧州 (アイルランド)	eu-west-1	lightsail.eu-west-1.amazonaws.com lightsail.eu-west-1.api.aws	HTTPS HTTPS
欧州 (ロンドン)	eu-west-2	lightsail.eu-west-2.amazonaws.com lightsail.eu-west-2.api.aws	HTTPS HTTPS
欧州 (パリ)	eu-west-3	lightsail.eu-west-3.amazonaws.com lightsail.eu-west-3.api.aws	HTTPS HTTPS
欧州 (ストックホルム)	eu-north-1	lightsail.eu-north-1.amazonaws.com lightsail.eu-north-1.api.aws	HTTPS HTTPS

エンドポイントを指定する例

このセクションでは、リクエストを行うときにエンドポイントを指定する方法を例で示します。

> **Note**
> エンドポイントを指定しない場合、デフォルトで IPv4 エンドポイントが使用されます。

AWS CLI

次の例では、 AWS CLIを使用して `us-east-2` リージョンのエンドポイントを指定する方法が示されます。

- IPv4

```
aws lightsail get-regions --region us-east-2 --endpoint-url https://lightsail.us-east-2.amazonaws.com
```

- デュアルスタック

```
aws lightsail get-regions --region us-east-2 --endpoint-url https://lightsail.us-east-2.api.aws
```

AWS SDK for Java 2.x

次の例では、 AWS SDK for Java 2.xを使用して us-east-2 リージョンのエンドポイントを指定する方法が示されます。

- IPv4

```
LightsailClient client = LightsailClient.builder()
    .region(Region.US_EAST_2)
    .endpointOverride(URI.create("https://lightsail.us-east-2.amazonaws.com"))
    .build();
```

- デュアルスタック

```
LightsailClient client = LightsailClient.builder()
    .region(Region.US_EAST_2)
    .endpointOverride(URI.create("https://lightsail.us-east-2.api.aws"))
    .build();
```

AWS SDK for Java 1.x

次の例は、 AWS SDK for Java 1.x を使用してus-east-2リージョンのエンドポイントを指定する方法を示しています。

- IPv4

```
AmazonLightsail lightsail = AmazonLightsailClientBuilder.standard()
     .withEndpointConfiguration(new EndpointConfiguration(
          "https://lightsail.us-east-2.amazonaws.com",
          "us-east-2"))
     .build();
```

- デュアルスタック

```
AmazonLightsail lightsail = AmazonLightsailClientBuilder.standard()
     .withEndpointConfiguration(new EndpointConfiguration(
```

```
            "https://lightsail.us-east-2.api.aws",
            "us-east-2"))
        .build();
```

AWS SDK for Go

次の例では、 AWS SDK for Goを使用して us-east-2 リージョンのエンドポイントを指定する方法が示されます。

- IPv4

```
sess := session.Must(session.NewSession())
svc := lightsail.New(sess, &aws.Config{
    Region: aws.String(endpoints.UsEast2RegionID),
    Endpoint: aws.String("https://lightsail.us-east-2.amazonaws.com")
})
```

- デュアルスタック

```
sess := session.Must(session.NewSession())
svc := lightsail.New(sess, &aws.Config{
    Region: aws.String(endpoints.UsEast2RegionID),
    Endpoint: aws.String("https://lightsail.us-east-2.api.aws")
})
```

Lightsail での仮想プライベートサーバーの使用を開始する方法

Lightsail では、インスタンスは仮想プライベートサーバー (仮想マシンとも呼ばれる) です。Lightsail インスタンスは、 AWS クラウドで作成して管理します。インスタンスの作成時に、そこのオペレーティングシステム (OS) が含まれているイメージを選択します。基本 OS だけでなくアプリケーションまたは開発スタックが含まれているインスタンスのイメージを選択することもできます。

このチュートリアルで作成したインスタンスには、インスタンスを作成してから削除するまでの間、使用料がかかります。削除はこのチュートリアルの最後に行う手順です。料金の詳細については、「Lightsail の料金」を参照してください。

トピック

- ステップ 1: 前提条件を満たす
- ステップ 2: インスタンスを作成する
- ステップ 3: インスタンスに接続する
- ステップ 4: インスタンスにストレージを追加する
- ステップ 5: スナップショットを作成する
- ステップ 6: クリーンアップする
- 次の手順
- Amazon Lightsail で を使用する AWS CLI

ステップ 1: 前提条件を満たす

新規の AWS お客様は、 の使用を開始する前にセットアップの前提条件を完了してくださいAmazon Lightsail。詳細については、「の設定 AWS アカウント と管理ユーザー Lightsail」を参照してください。

ステップ 2: インスタンスを作成する

以下の手順で説明しているように、Lightsail コンソールを使用してインスタンスを作成できます。このチュートリアルは、最初のインスタンスをすばやく起動できるようにすることを目的としていま

す。また、利用可能なアプリケーションとハードウェア プランを調べることをお勧めします。詳細については、「Lightsail インスタンスブループリントのサービスを確認する」を参照してください。

1. Lightsail コンソール にサインインします。
2. ホームページで [インスタンスの作成] を選択します。
3. インスタンスの場所 (AWS リージョン とアベイラビリティーゾーン) を選択します。レイテンシーを短縮するために、物理的な場所に最も AWS リージョン 近い を選択します。

   変更 AWS リージョン とアベイラビリティーゾーンを選択して、別の場所にインスタンスを作成します。
4. アプリケーション ([アプリ + OS]) またはオペレーティングシステム ([OS のみ]) を選択できます。

   Lightsail インスタンス イメージの詳細については、「Lightsail インスタンスブループリントのサービスを確認する」を参照してください。
5. インスタンスプランを選択します。

   インスタンスがデュアルスタック (IPv4 と IPv6) を使用するか IPv6 専用ネットワークを使用するかを選択します。一部の Lightsail ブループリントは、現在のところ IPv6 専用ネットワークをサポートしていません。どのブループリントが IPv6 専用のネットワーキングをサポートしているかを確認するには、Lightsail インスタンスブループリントのサービスを確認する を参照してください。

   5 米ドルの Lightsail プランを、1 か月間 (最大 750 時間) 無料でお試しいただけます。1 か月の無料期間分はアカウントに返金されます。詳細については、Lightsail の料金ページを参照してください。
6. インスタンスの名前を入力します。

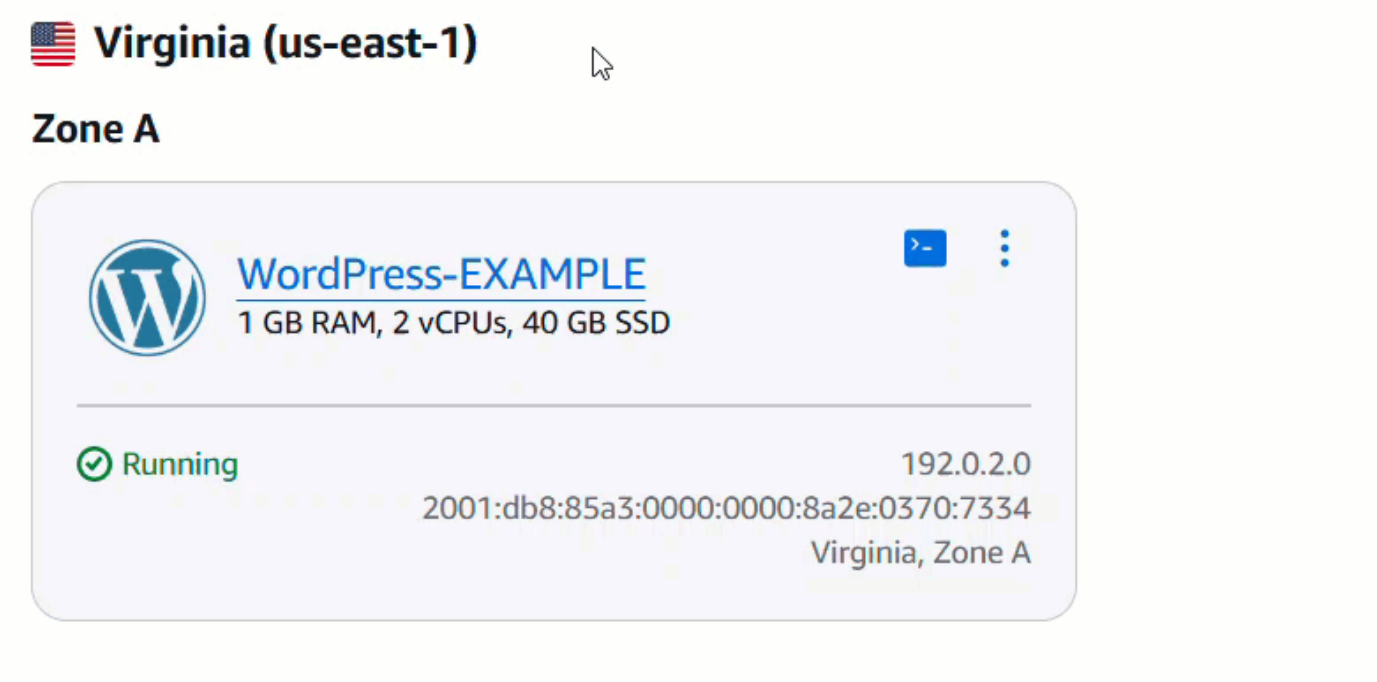
   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。
7. [インスタンスの作成] を選択します。

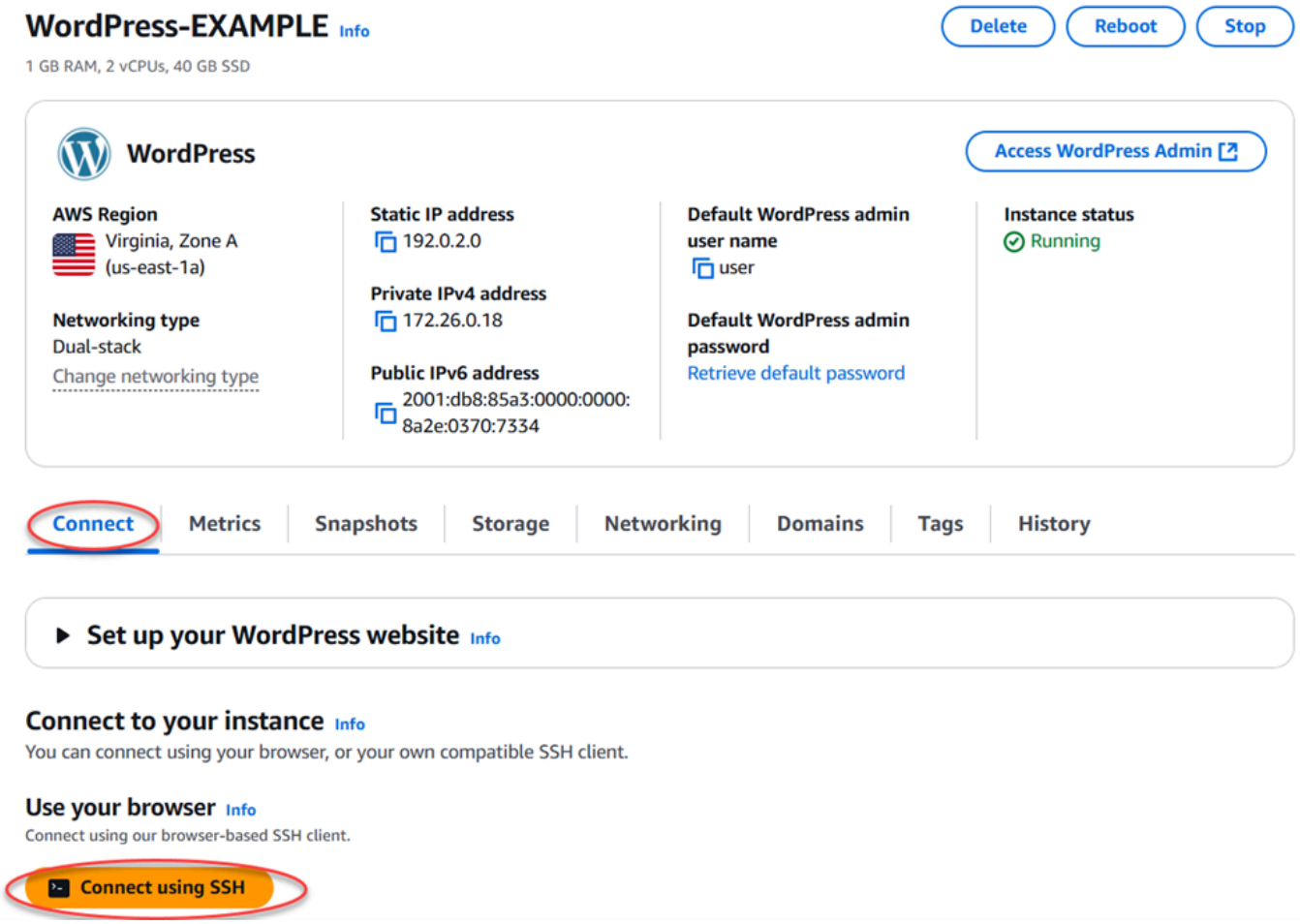
数分で Lightsail インスタンスの準備が整い、そのインスタンスに接続できます。

ステップ 3: インスタンスに接続する

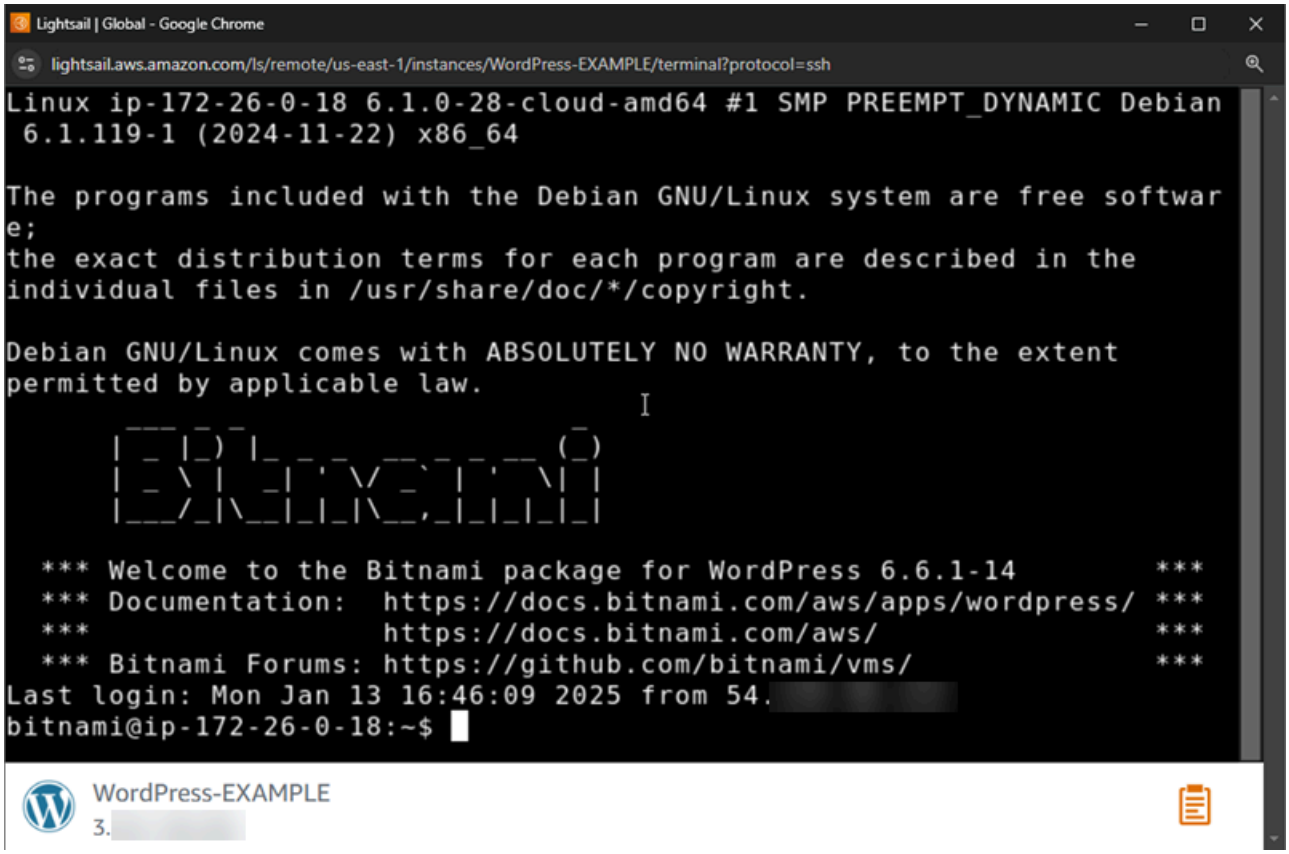
1. Lightsail ホームページからアクションメニューアイコン (⋮) を選択し、[接続] を選択します。



または、インスタンスの管理ページから接続することもできます。インスタンスの名前を選択し、[接続] タブを選択した後、[SSH を使用して接続] を選択します。



2. これで、SSH クライアントをセットアップせずに、ターミナルにコマンドを入力して Lightsail インスタンスを管理できるようになります。



仮想コンピュータに追加ストレージを接続する方法については、このチュートリアルの次のステップで説明します。

ステップ 4: インスタンスにストレージを追加する

Lightsail は、インスタンスにアタッチできる、ブロックレベルのストレージ容量を提供します。インスタンスにはシステムディスクが付属していますが、ニーズの変化に応じて追加のストレージディスクをアタッチできます。インスタンスからディスクをデタッチし、別のインスタンスにアタッチすることもできます。

追加ディスクを作成したら、Lightsail インスタンスに接続してディスクをフォーマットおよびマウントする必要があります。

ディスクの作成、アタッチおよび管理の詳細については、「Lightsail ブロックストレージディスクを作成して Linux ベースのインスタンスにアタッチする」を参照してください。

このチュートリアルの次の手順で、仮想コンピューターのバックアップについて説明します。

ステップ 5: スナップショットを作成する

スナップショットは、データのポイントインタイムコピーです。インスタンスのスナップショットを作成し、新しいインスタンスを作成したり、データをバックアップしたりするためのベースラインとして使用できます。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点からのデータ) が含まれます。

スナップショットの作成と管理に関する詳細は、「スナップショットを使用して Linux/Unix Lightsail インスタンスをバックアップする」を参照してください。

このチュートリアルの次の手順で、仮想コンピューターリソースのクリーンアップについて説明します。

ステップ 6: クリーンアップする

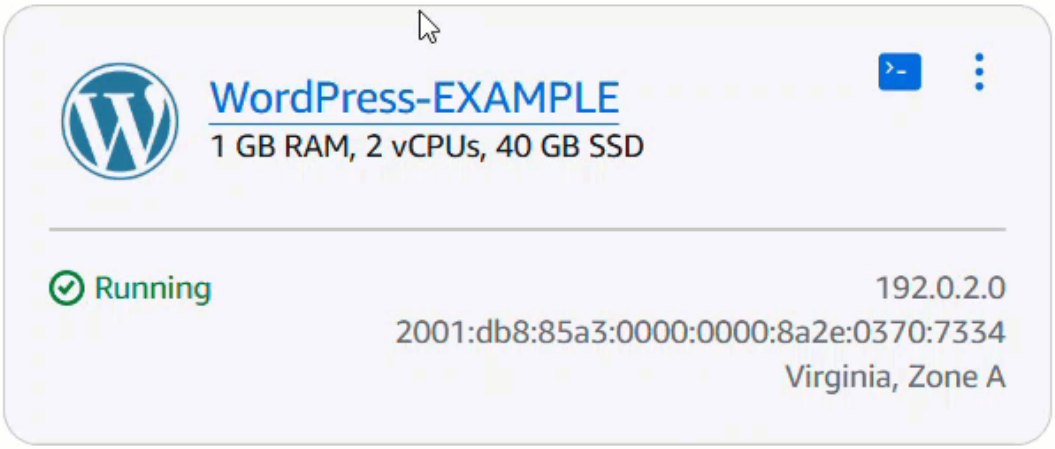
このチュートリアル用に作成したインスタンスを使用して操作した後に、削除することができます。これにより、不要になったインスタンスに対する料金は発生しなくなります。

インスタンスを削除しても、それに関連するスナップショットやアタッチされたディスクは削除されません。このチュートリアル用にスナップショットとディスクを作成した場合は、それらも削除する必要があります。

後のためにインスタンスを保存したいが料金を発生させたくない場合は、そのインスタンスを削除する代わりに停止することができます。後でそのインスタンスを再起動できます。料金の詳細については、「Lightsail の料金」を参照してください。

> **Important**
>
> Lightsail リソースの削除は永続的な操作です。削除されたデータは復元できません。後でデータが必要になるかもしれない場合は、削除する前に仮想コンピュータのスナップショットを作成してください。詳細については、「スナップショットを使用して Linux/Unix Lightsail インスタンスをバックアップする」を参照してください。

1. Lightsail コンソール にサインインします。
2. ナビゲーションペインで、[Instances (インスタンス)] を選択します。
3. 削除するインスタンスのアクションメニューアイコン (⋮) を選択し、[削除] を選択します。



4. [はい、削除します] を選択して削除を確定します。

次の手順

次のトピックを使用して Amazon Lightsail Linux および Windows ベースのインスタンスを開始します。

- Lightsail でアプリケーションを使用して Linux/Unix インスタンスを作成する
- Lightsail で Windows Server インスタンスを作成する

Amazon Lightsail で を使用する AWS CLI

このチュートリアルでは、 AWS Command Line Interface (AWS CLI) を使用した一般的な Amazon Lightsail オペレーションについて説明します。キーペア、インスタンス、ストレージ、スナップショットなどの Lightsail リソースを作成および管理する方法を学びます。

トピック

- 前提条件
- SSH キーペアの生成
- インスタンスを作成して管理する
- インスタンスへの接続
- インスタンスにストレージを追加する
- スナップショットの作成と使用
- リソースをクリーンアップする

- 次の手順

前提条件

このチュートリアルを開始する前に、以下の要件を満たしていることを確認してください。

1. AWS CLI。インストールする必要がある場合は、「AWS CLI の最新バージョンのインストールまたは更新」に従ってください。を含む を使用 AWS CloudShellすることもできます AWS CLI。
2. 適切な認証情報 AWS CLI を使用して を設定しました。認証情報をまだ設定していない場合は、`aws configure` を実行します。
3. コマンドラインインターフェイスと SSH の概念に関する基本的な知識。
4. AWS アカウントでLightsailリソースを作成および管理するための十分なアクセス許可。

まだ設定していない場合は、`AWS_REGION`環境変数を、開始する前に が使用する AWS CLI ように設定したのと同じリージョンに設定します。この環境変数は、Lightsail リソースのアベイラビリティーゾーンを指定するコマンド例で使用されます。

```
$ [ -z "${AWS_REGION}" ] && export AWS_REGION=$(aws configure get region)
```

CLI を使用した Amazon Lightsail リソースの作成と管理を開始しましょう。

SSH キーペアの生成

SSH キーペアを使用すると、パスワードを使用せずに Lightsail インスタンスに安全に接続できます。このセクションでは、新しいキーペアを作成し、その情報を取得します。

Example– 新しいキーペアの作成

次のコマンドで、「cli-tutorial-keys」という名前の新しい SSH キーペアを作成し、プライベートキーをローカルマシンに保存します。

```
$ aws lightsail create-key-pair --key-pair-name cli-tutorial-keys \
        --query privateKeyBase64 --output text > ~/.ssh/cli-tutorial-keys.pem
$ chmod 400 ~/.ssh/cli-tutorial-keys.pem
```

このコマンドを実行すると、プライベートキーは適切なアクセス許可で `~/.ssh` ディレクトリに保存されます。chmod コマンドによって、SSH のセキュリティ要件であるプライベートキーファイルをユーザーのみが読み取るようになります。

Example– キーペア情報を取得する

キーペアが正常に作成されたことを確認するには、その情報を取得します。

```
$ aws lightsail get-key-pair --key-pair-name cli-tutorial-keys
{
    "keyPair": {
        "name": "cli-tutorial-keys",
        "arn": "arn:aws:lightsail:us-east-2:123456789012:KeyPair/e00xmpl-6a6a-434a-
bff1-87f2bb815e21",
        "supportCode": "123456789012/cli-tutorial-keys",
        "createdAt": 1673596800.000,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-east-2"
        },
        "resourceType": "KeyPair",
        "tags": [],
        "fingerprint": "d0:0d:30:db:5a:24:df:f6:17:f0:e2:15:45:77:3d:bb:d0:6d:fc:81"
    }
}
```

出力には、名前、ARN、作成時刻、リージョン、フィンガープリントなど、キーペアの詳細が表示されます。このフィンガープリントを使用して、インスタンスに接続するときにキーの信頼性を検証できます。

インスタンスを作成して管理する

Lightsail インスタンスは、アプリケーションまたはウェブサイトを実行する仮想プライベートサーバーです。このセクションでは、WordPress インスタンスを作成し、その詳細を取得します。

Example– WordPress インスタンスを作成する

次のコマンドで、nano_3_0 バンドル (最小の Lightsail インスタンスサイズ) を使用して新しい WordPress インスタンスを作成し、キーペアに関連付けます。コマンドは、AWS_REGION 環境変数を使用して、設定されたリージョンのアベイラビリティーゾーンにインスタンスを作成します。

```
$ aws lightsail create-instances --instance-names cli-tutorial \
        --availability-zone ${AWS_REGION}a --blueprint-id wordpress \
        --bundle-id nano_3_0 --key-pair-name cli-tutorial-keys
{
    "operations": [
```

```
        {
            "id": "f30xmpl-3727-492a-9d42-5c94ad3ef9a8",
            "resourceName": "cli-tutorial",
            "resourceType": "Instance",
            "createdAt": 1673596800.000,
            "location": {
                "availabilityZone": "us-east-2a",
                "regionName": "us-east-2"
            },
            "isTerminal": false,
            "operationType": "CreateInstance",
            "status": "Started",
            "statusChangedAt": 1673596800.000
        }
    ]
}
```

レスポンスは、インスタンス作成オペレーションが開始されたことを示します。インスタンスが使用可能になるまで数分かかる場合があります。

Example– インスタンスの詳細を取得する

インスタンスが作成されたら、次のコマンドを使用してその詳細を取得できます。

```
$ aws lightsail get-instance --instance-name cli-tutorial
{
    "instance": {
        "name": "cli-tutorial",
        "arn": "arn:aws:lightsail:us-east-2:123456789012:Instance/7d3xmpl-ae2e-44d5-
bbd9-22f9ec2abe1f",
        "supportCode": "123456789012/i-099cxmpl5dad5923c",
        "createdAt": 1673596800.000,
        "location": {
            "availabilityZone": "us-east-2a",
            "regionName": "us-east-2"
        },
        "resourceType": "Instance",
        "tags": [],
        "blueprintId": "wordpress",
        "blueprintName": "WordPress",
        "bundleId": "nano_3_0",
        "isStaticIp": false,
        "privateIpAddress": "172.26.6.136",
        "publicIpAddress": "192.0.2.1",
```

```
        "ipv6Addresses": [
            "2001:db8:85a3:0000:0000:8a2e:0370:7334"
        ],
        "ipAddressType": "dualstack",
        "hardware": {
            "cpuCount": 2,
            "disks": [
                {
                    "createdAt": 1673596800.000,
                    "sizeInGb": 20,
                    "isSystemDisk": true,
                    "iops": 100,
                    "path": "/dev/xvda",
                    "attachedTo": "cli-tutorial",
                    "attachmentState": "attached"
                }
            ],
            "ramSizeInGb": 0.5
        },
        "networking": {
            "monthlyTransfer": {
                "gbPerMonthAllocated": 1024
            },
            "ports": [
                {
                    "fromPort": 80,
                    "toPort": 80,
                    "protocol": "tcp",
                    "accessFrom": "Anywhere (0.0.0.0/0 and ::/0)",
                    "accessType": "public",
                    "commonName": "",
                    "accessDirection": "inbound",
                    "cidrs": [
                        "0.0.0.0/0"
                    ],
                    "ipv6Cidrs": [
                        "::/0"
                    ],
                    "cidrListAliases": []
                },
                {
                    "fromPort": 22,
                    "toPort": 22,
                    "protocol": "tcp",
```

```
                    "accessFrom": "Anywhere (0.0.0.0/0 and ::/0)",
                    "accessType": "public",
                    "commonName": "",
                    "accessDirection": "inbound",
                    "cidrs": [
                        "0.0.0.0/0"
                    ],
                    "ipv6Cidrs": [
                        "::/0"
                    ],
                    "cidrListAliases": []
                },
                {
                    "fromPort": 443,
                    "toPort": 443,
                    "protocol": "tcp",
                    "accessFrom": "Anywhere (0.0.0.0/0 and ::/0)",
                    "accessType": "public",
                    "commonName": "",
                    "accessDirection": "inbound",
                    "cidrs": [
                        "0.0.0.0/0"
                    ],
                    "ipv6Cidrs": [
                        "::/0"
                    ],
                    "cidrListAliases": []
                }
            ]
        },
        "state": {
            "code": 16,
            "name": "running"
        },
        "username": "bitnami",
        "sshKeyName": "cli-tutorial-keys",
        "metadataOptions": {
            "state": "applied",
            "httpTokens": "optional",
            "httpEndpoint": "enabled",
            "httpPutResponseHopLimit": 1,
            "httpProtocolIpv6": "disabled"
        }
    }
```

```
}
```

出力には、IP アドレス、ハードウェア仕様、ネットワーク設定、状態など、インスタンスに関する包括的な情報が表示されます。パブリック IP アドレスとユーザー名を書き留めます。これらはインスタンスに接続するために必要になります。

インスタンスへの接続

インスタンスを作成したら、前に作成したキーペアで SSH を使用してインスタンスに接続できます。このセクションでは、SSH 接続を確立し、セキュリティ設定を管理する方法について説明します。

Example– インスタンスへの SSH

次のコマンドを使用して SSH 経由でインスタンスに接続し、IP アドレスをインスタンスのパブリック IP に置き換えます。

```
$ ssh -i ~/.ssh/cli-tutorial-keys.pem bitnami@192.0.2.1
Linux ip-172-26-6-136 6.1.0-32-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.129-1
 (2025-03-06) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
       ___ _ _                   _
      | _ |_) |_ _ _  __ _ _ _ __ (_)
      | _ \ |  _| ' \/ _` | '  \| |
      |___/_|\__|_|_|\__,_|_|_|_|_|_|

  *** Welcome to the Bitnami package for WordPress 6.7.2           ***
  *** Documentation:  https://docs.bitnami.com/aws/apps/wordpress/ ***
  ***                 https://docs.bitnami.com/aws/                ***
  *** Bitnami Forums: https://github.com/bitnami/vms/              ***

bitnami@ip-172-26-6-136:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev               217920       0    217920   0% /dev
tmpfs               45860     480     45380   2% /run
/dev/nvme0n1p1   20403592 3328832  16142256  18% /
```

```
tmpfs              229292       0    229292   0% /dev/shm
tmpfs                5120       0      5120   0% /run/lock
/dev/nvme0n1p15    126678   11840    114838  10% /boot/efi
tmpfs               45856       0     45856   0% /run/user/1000
```

接続したら、WordPress のインストールを管理したり、サーバーを設定したり、追加のソフトウェアをインストールしたりできます。上記の例は、df コマンドを使用したインスタンスのディスク使用量を示しています。

Example– パブリックポートを閉じる

SSH を使用していない場合は、インスタンスのパブリックポートを閉じることができます。これにより、不正アクセスの試みからインスタンスを保護することができます。

```
$ aws lightsail close-instance-public-ports --instance-name cli-tutorial \
        --port-info fromPort=22,protocol=TCP,toPort=22
{
    "operation": {
        "id": "6cdxmpl-9f39-4357-a66d-230096140b4f",
        "resourceName": "cli-tutorial",
        "resourceType": "Instance",
        "createdAt": 1673596800.000,
        "location": {
            "availabilityZone": "us-east-2a",
            "regionName": "us-east-2"
        },
        "isTerminal": true,
        "operationDetails": "22/tcp",
        "operationType": "CloseInstancePublicPorts",
        "status": "Succeeded",
        "statusChangedAt": 1673596800.000
    }
}
```

Note

ポート 22 を閉じると、Lightsail コンソールから開始された接続を含むすべての SSH 接続が禁止されます。詳細については、以下のトピックを参照してください。

- SSH キーペアを管理し、Lightsail インスタンスに接続する
- Lightsail のファイアウォールを使用してインスタンストラフィックを制御する

レスポンスは、ポート 22 が正常に閉じられたことを確認します。SSH 経由で再接続する必要がある場合は、open-instance-public-ports コマンドを使用してポートを再度開くことができます。

インスタンスにストレージを追加する

アプリケーションが大きくなるにつれて、追加のストレージ容量が必要になる場合があります。Lightsail では、追加のディスクを作成してインスタンスにアタッチできます。このセクションでは、ストレージを追加する方法を示します。

Example– ディスクを作成する

次のコマンドで、新しい 32 GB のディスクを作成します。

```
$ aws lightsail create-disk --disk-name cli-tutorial-disk \
        --availability-zone ${AWS_REGION}a --size-in-gb 32
{
    "operations": [
        {
            "id": "070xmpl-3364-4aa2-bff2-3c589de832fc",
            "resourceName": "cli-tutorial-disk",
            "resourceType": "Disk",
            "createdAt": 1673596800.000,
            "location": {
                "availabilityZone": "us-east-2a",
                "regionName": "us-east-2"
            },
            "isTerminal": false,
            "operationType": "CreateDisk",
            "status": "Started",
            "statusChangedAt": 1673596800.000
        }
    ]
}
```

レスポンスは、ディスク作成オペレーションが開始されたことを示します。ディスクが使用可能になるまでに少し時間がかかる場合があります。

Example– ディスクをインスタンスにアタッチする

ディスクが作成されたら、次のコマンドを使用してインスタンスにアタッチできます。

```
$ aws lightsail attach-disk --disk-name cli-tutorial-disk \
```

```
        --disk-path /dev/xvdf --instance-name cli-tutorial
{
    "operations": [
        {
            "id": "d17xmpl-2bdb-4292-ac63-ba5537522cea",
            "resourceName": "cli-tutorial-disk",
            "resourceType": "Disk",
            "createdAt": 1673596800.000,
            "location": {
                "availabilityZone": "us-east-2a",
                "regionName": "us-east-2"
            },
            "isTerminal": false,
            "operationDetails": "cli-tutorial",
            "operationType": "AttachDisk",
            "status": "Started",
            "statusChangedAt": 1673596800.000
        },
        {
            "id": "01exmpl-c04e-42d4-aa6b-45ce50562a54",
            "resourceName": "cli-tutorial",
            "resourceType": "Instance",
            "createdAt": 1673596800.000,
            "location": {
                "availabilityZone": "us-east-2a",
                "regionName": "us-east-2"
            },
            "isTerminal": false,
            "operationDetails": "cli-tutorial-disk",
            "operationType": "AttachDisk",
            "status": "Started",
            "statusChangedAt": 1673596800.000
        }
    ]
}
```

disk-path パラメータは、ディスクを Linux ファイルシステムにアタッチする場所を指定します。ディスクをアタッチしたら、インスタンス内からディスクをフォーマットしてマウントする必要があります。

Example– ディスクのアタッチを検証する

ディスクの詳細を取得することで、ディスクが正しくアタッチされていることを確認できます。

```
$ aws lightsail get-disk --disk-name cli-tutorial-disk
{
    "disk": {
        "name": "cli-tutorial-disk",
        "arn": "arn:aws:lightsail:us-east-2:123456789012:Disk/1a9xmpl-8a34-46a4-
b87e-19184f0cca9c",
        "supportCode": "123456789012/vol-0dacxmplc1c3108e2",
        "createdAt": 1673596800.000,
        "location": {
            "availabilityZone": "us-east-2a",
            "regionName": "us-east-2"
        },
        "resourceType": "Disk",
        "tags": [],
        "sizeInGb": 32,
        "isSystemDisk": false,
        "iops": 100,
        "path": "/dev/xvdf",
        "state": "in-use",
        "attachedTo": "cli-tutorial",
        "isAttached": true,
        "attachmentState": "attached"
    }
}
```

出力から、ディスクがインスタンスにアタッチされていることが確認できます。「state」フィールドが「in-use」を示し、「isAttached」は true に設定されていることから、アタッチが成功したことがわかります。

スナップショットの作成と使用

スナップショットを使用すると、インスタンスをバックアップし、バックアップから新しいインスタンスを作成できます。これは、ディザスタリカバリ、テスト、または重複環境の作成に役立ちます。

Example– インスタンスのスナップショットを作成する

次のコマンドで、インスタンスのスナップショットを作成します。

```
$ aws lightsail create-instance-snapshot --instance-name cli-tutorial \
        --instance-snapshot-name cli-tutorial-snapshot
{
    "operations": [
```

```
        {
            "id": "41bxmpl-7824-4591-bfcc-1b1c341613a4",
            "resourceName": "cli-tutorial-snapshot",
            "resourceType": "InstanceSnapshot",
            "createdAt": 1673596800.000,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-east-2"
            },
            "isTerminal": false,
            "operationDetails": "cli-tutorial",
            "operationType": "CreateInstanceSnapshot",
            "status": "Started",
            "statusChangedAt": 1673596800.000
        },
        {
            "id": "725xmpl-158e-46f6-bd49-27b0e6805aa2",
            "resourceName": "cli-tutorial",
            "resourceType": "Instance",
            "createdAt": 1673596800.000,
            "location": {
                "availabilityZone": "us-east-2a",
                "regionName": "us-east-2"
            },
            "isTerminal": false,
            "operationDetails": "cli-tutorial-snapshot",
            "operationType": "CreateInstanceSnapshot",
            "status": "Started",
            "statusChangedAt": 1673596800.000
        }
    ]
}
```


レスポンスは、スナップショットのプロセスが開始されたことを示します。スナップショットを取得するインスタンスには 1 つの非同期オペレーションがあり、作成されるスナップショットには 1 つの非同期オペレーションがあります。スナップショットには、インスタンスにアタッチされたすべてのディスクが含まれます。

Example– スナップショットから新しいインスタンスを作成する

スナップショットが完了したら、そのスナップショットを使用して新しいインスタンスを作成できます。

```
$ aws lightsail create-instances-from-snapshot --availability-zone ${AWS_REGION}b \
        --instance-snapshot-name cli-tutorial-snapshot --instance-name cli-tutorial-bup
 --bundle-id small_3_0
{
    "operations": [
        {
            "id": "a35xmpl-efa1-4d6c-958e-9d58fd258f5f",
            "resourceName": "cli-tutorial-bup",
            "resourceType": "Instance",
            "createdAt": 1673596800.000,
            "location": {
                "availabilityZone": "us-east-2b",
                "regionName": "us-east-2"
            },
            "isTerminal": false,
            "operationType": "CreateInstancesFromSnapshot",
            "status": "Started",
            "statusChangedAt": 1673596800.000
        }
    ]
}
```

このコマンドは、small_3_0 バンドルサイズを使用して、アベイラビリティーゾーン us-east-2b に cli-tutorial-bup という名前の新しいインスタンスを作成します。新しいインスタンスには別のバンドルサイズを選択できます。これは、スケールアップまたはスケールダウンに役立ちます。

リソースをクリーンアップする

Lightsail レスポンスを終了したら、追加料金が発生しないようにすべてのレスポンスを削除する必要があります。このセクションでは、このチュートリアルで作成したリソースをすべてクリーンアップする方法について説明します。

Example– インスタンスのスナップショットを削除する

不要になったスナップショットを削除するには、次のコマンドを使用します。

```
$ aws lightsail delete-instance-snapshot --instance-snapshot-name cli-tutorial-snapshot
{
    "operations": [
        {
            "id": "cf8xmpl-0ec7-43ec-9cbc-6dedd9d8eda8",
```

```
            "resourceName": "cli-tutorial-snapshot",
            "resourceType": "InstanceSnapshot",
            "createdAt": 1673596800.000,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-east-2"
            },
            "isTerminal": true,
            "operationDetails": "",
            "operationType": "DeleteInstanceSnapshot",
            "status": "Succeeded",
            "statusChangedAt": 1673596800.000
        }
    ]
}
```

レスポンスにより、スナップショットの削除オペレーションが成功したことが確認できます。

Example– インスタンスを削除する

インスタンスを削除するには、次のコマンドを使用します。

```
$ aws lightsail delete-instance --instance-name cli-tutorial
{
    "operations": [
        {
            "id": "f4bxmpl-2df1-4740-90d7-e30adaf7e3a1",
            "resourceName": "cli-tutorial",
            "resourceType": "Instance",
            "createdAt": 1673596800.000,
            "location": {
                "availabilityZone": "us-east-2a",
                "regionName": "us-east-2"
            },
            "isTerminal": true,
            "operationDetails": "",
            "operationType": "DeleteInstance",
            "status": "Succeeded",
            "statusChangedAt": 1673596800.000
        }
    ]
}
```

スナップショットから作成されたインスタンスを含め、作成したすべてのインスタンスを削除します。

Example– ディスクを削除する

不要になったディスクを削除するには、次のコマンドを使用します。

```
$ aws lightsail delete-disk --disk-name cli-tutorial-disk
{
    "operations": [
        {
            "id": "aacxmpl-8626-4edd-8b3b-bf108d6b279c",
            "resourceName": "cli-tutorial-disk",
            "resourceType": "Disk",
            "createdAt": 1673596800.000,
            "location": {
                "availabilityZone": "us-east-2a",
                "regionName": "us-east-2"
            },
            "isTerminal": true,
            "operationDetails": "",
            "operationType": "DeleteDisk",
            "status": "Succeeded",
            "statusChangedAt": 1673596800.000
        }
    ]
}
```

ディスクがインスタンスにアタッチされている場合は、まず detach-disk コマンドを使用してディスクをデタッチする必要があります。

Example– キーペアを削除する

最後に、このチュートリアルの冒頭で作成したキーペアを削除します。

```
$ aws lightsail delete-key-pair --key-pair-name cli-tutorial-keys
{
    "operation": {
        "id": "dbfxmpl-c954-4a45-93a4-ab3e627d2c23",
        "resourceName": "cli-tutorial-keys",
        "resourceType": "KeyPair",
        "createdAt": 1673596800.000,
        "location": {
```

```
            "availabilityZone": "all",
            "regionName": "us-east-2"
        },
        "isTerminal": true,
        "operationDetails": "",
        "operationType": "DeleteKeyPair",
        "status": "Succeeded",
        "statusChangedAt": 1673596800.000
    }
}
```

このコマンドは、キーペアのみを削除します AWS。これで、ローカルコピーも削除できます。

```
$ rm ~/.ssh/cli-tutorial-keys.pem
```

次の手順

を使用してLightsailリソースを管理するための基本を学習したので AWS CLI、他のLightsail機能について説明します。

1. ドメイン – アプリケーションにドメイン名を割り当てます。
2. ロードバランサー – トラフィックを複数のインスタンスにルーティングして、容量と耐障害性を向上させます。
3. 自動スナップショット – アプリケーションデータを自動的にバックアップします。
4. メトリクス – リソースの状態のモニタリング、通知の取得、アラームの設定を行います。
5. データベース – アプリケーションをリレーショナルデータベースに接続します。

使用可能な AWS CLI コマンドの詳細については、「 のAWS CLI コマンドリファレンスLightsail」を参照してください。

Lightsail リセラー

登録済みの Amazon Lightsail リセラーになると、自分の顧客に Lightsail 製品を提供できます。Lightsail リセラーになると、Lightsail インスタンスのデフォルトクォータが高くなり、登録されたリセラー専用のコンソール内フィードバックフォームを使用できるようになります。

トピック

- Lightsail 再販の利点
- Lightsail リセラーの利点と増額されたデフォルトクォータがアカウントにどのように適用されるか
- Lightsail リセラーになる方法
- Lightsail リセラーになる
- リセラーアカウントのサービスクォータの引き上げをリクエストする
- リセラーとして Lightsail に連絡する

Lightsail 再販の利点

Lightsail リセラーになることで、スケーリング、予算編成、支援の取得に関する Lightsail リソースついてさまざまな利点が得られます。

Lightsail でビジネスをスケールする

リセラーは、Lightsail のグローバルクラウドインフラストラクチャでビジネスをより迅速に拡張できます。リセラーのメリットにより、 AWS リージョン デフォルトでは、各 の登録済みアカウント全体のLightsailインスタンスのサービスクォータが高くなります。

予算を簡素化する

Lightsail には、メモリ、vCPU、ソリッドステートドライブ (SSD) ストレージがバンドルプランとして提供される予測可能な料金モデルがあります。このモデルを使用すると、Lightsail リソースを使用して大規模にビジネスを成長させ、管理しながらコストを簡単に予測できます。

信頼性

データの自動スナップショット、設定されたしきい値を超えるリソースのアラーム通知、IPv6 ネットワークのサポートなどの機能を使用して、リソースの効率と信頼性を向上させます。

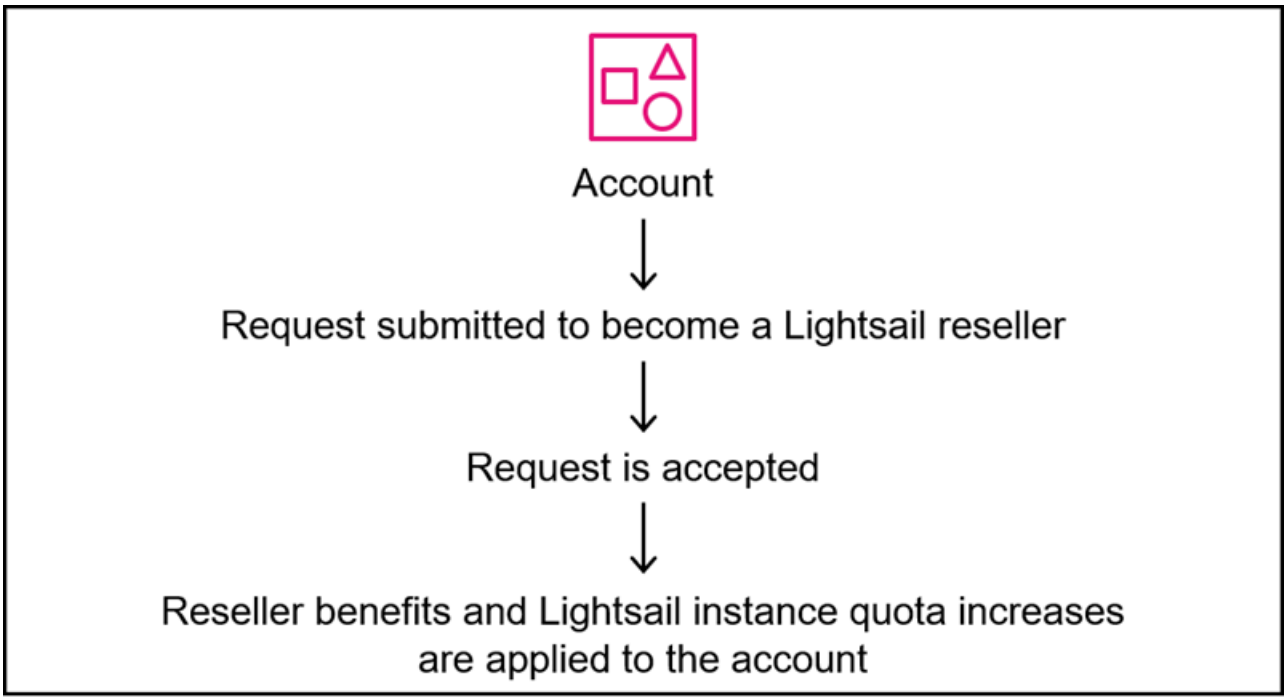
Lightsail リセラーの利点と増額されたデフォルトクォータがアカウントにどのように適用されるか

リセラーのメリット AWS アカウント は、リクエストの送信元の に適用されます。リクエストが承認された場合は、デフォルトのLightsailインスタンスクォータを増やす AWS アカウント ために を追加するようリクエストできます。を使用する場合 AWS Organizations、リセラーのメリットとデフォルトのLightsailインスタンスクォータの増加が管理アカウントに適用されます。メンバーアカウントの場合、Lightsail インスタンスのデフォルトクォータが増加します。Organizations の詳細については、AWS Organizations 「 ユーザーガイド」の「What is AWS Organizations?」を参照してください。

次の図は、Lightsailリセラーのメリットと増加したデフォルトLightsailインスタンスクォータがどのように適用されるかを示しています AWS アカウント。

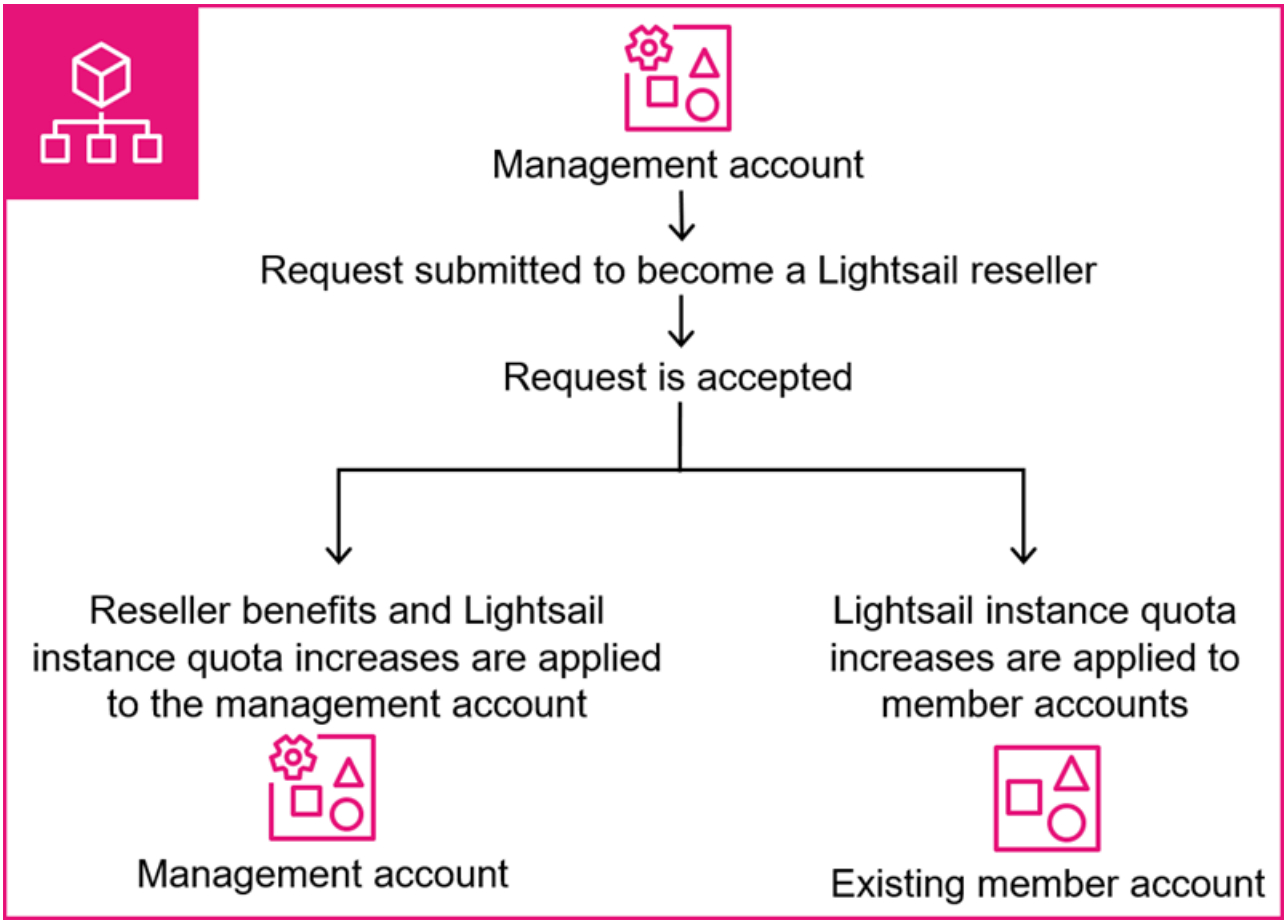
シングル AWS アカウント

次の図は、 の外部の 1 つのアカウントがLightsailリセラー AWS Organizations になるとどうなるかを示しています。



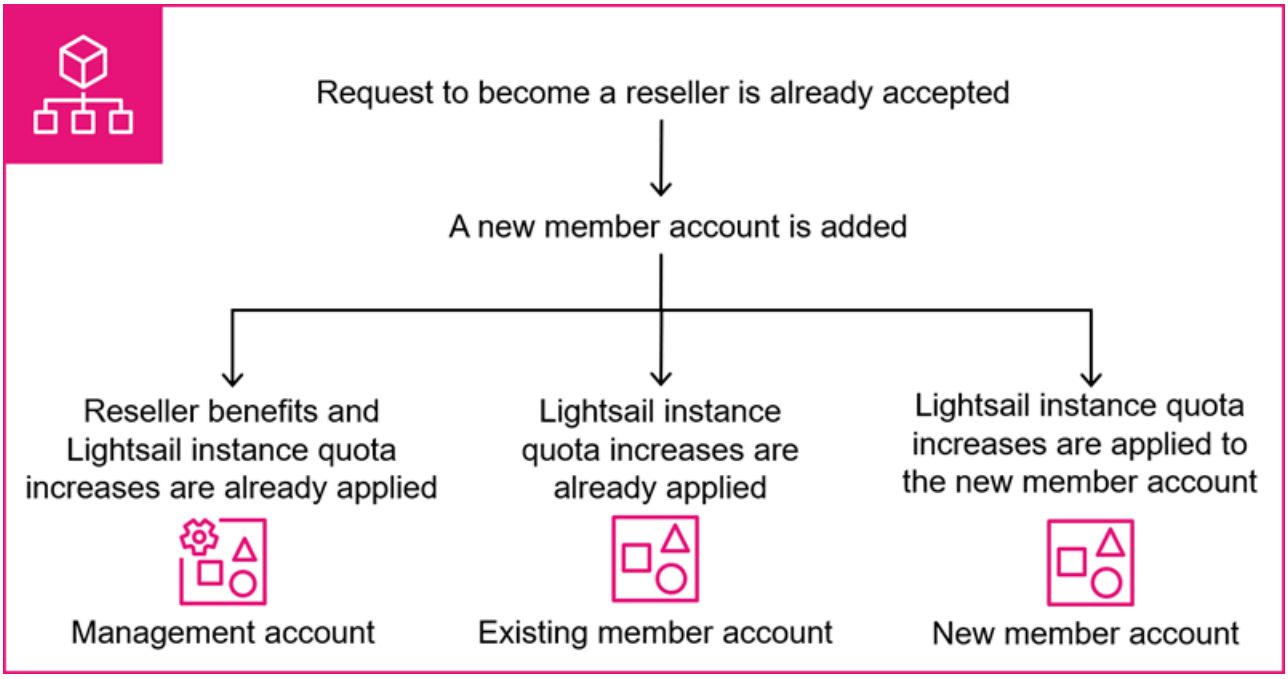
AWS アカウント Organizations の

次の図は、 AWS Organizations の管理アカウントが Lightsail リセラーになるとどうなるかを示しています。



AWS アカウント リセラーになった後に追加された Organizations の

次の図は、管理アカウントがすでに Lightsail リセラーとして登録されている組織に新しいメンバーアカウントが追加されたときに何が起こるかを示しています。



Lightsail リセラーになる方法

続行するには、Lightsail リセラーになるには、ビジネスニーズに関する詳細が記載されたフォームを送信する必要があります。詳細については、「Lightsail リセラーになる」を参照してください。

Lightsail リセラーになる

Amazon Lightsail リセラーになるための審査を受けるには、フォームを送信する必要があります。リクエストは、フォームの入力時にログイン AWS アカウント している を使用してファイルされます。 AWS Organizations を使用して を一元管理する場合は AWS アカウント、管理アカウントを使用してリセラーになるときにリクエストを送信する必要があります。管理アカウントを使用すると、組織内のメンバーアカウント全体でデフォルトの Lightsail インスタンスクォータが増加します。Lightsail リセラーのメリットが に与える影響の詳細については AWS アカウント、「」を参照してくださいLightsail リセラーの利点と増額されたデフォルトクォータがアカウントにどのように適用されるか。

リクエストが承認され、複数の組織がある場合は、追加のリクエストを送信して、各組織の管理アカウントの AWS アカウント ID を追加して、増加したデフォルトのLightsailインスタンスクォータをそれらの組織のメンバーアカウントにスケーリングできます。Organizations の詳細については、AWS Organizations 「 ユーザーガイド」の「What is AWS Organizations?」を参照してください。

トピック

- Lightsail リセラーになるために必要な情報
- Lightsail リセラーになるためのリクエスト
- Lightsail リセラーになる追加のアカウントをリクエストする

Lightsail リセラーになるために必要な情報

Amazon Lightsail リセラーになるリクエストを検討するために、計画された使用状況とユースケースに関する情報が必要になります。フォームは、Lightsail コンソールで作成し、検討のために送信できます。フォームに入力するには、ビジネスの詳細に加えて、次の情報が必要です。

- 使用する予定の Lightsail リソースのインスタンスバンドルのサイズと数量。利用可能なバンドルの詳細については、「Amazon Lightsail pricing」を参照してください。
- 登録するAWS アカウント IDs。を使用している場合は AWS Organizations、リクエストで管理アカウントのみを指定する必要があります。これにより、組織内の各メンバーアカウントも登録されます。詳細については、AWS Organizations ユーザーガイドの「Terminology and concepts for AWS Organizations」を参照してください。

Lightsail リセラーになるためのリクエスト

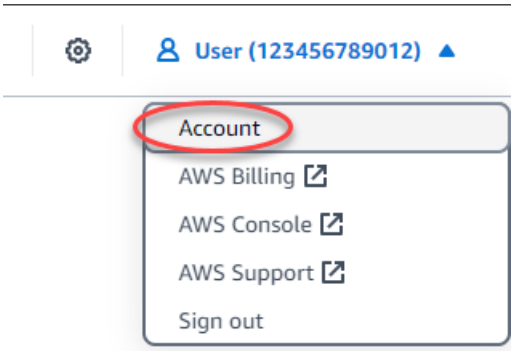
以下の手順では、リセラーになるためのリクエストを送信します。認証される AWS アカウント ID は、リセラーのメリットを受けるアカウントとして使用されます。リクエストが承認されたら、アカウントの追加をリクエストできます。

> **Tip**
>
> を使用している場合は AWS Organizations、この手順を組織の管理アカウントとして実行して、メンバーアカウントにデフォルトのLightsailインスタンスクォータも引き上げられるようにする必要があります。

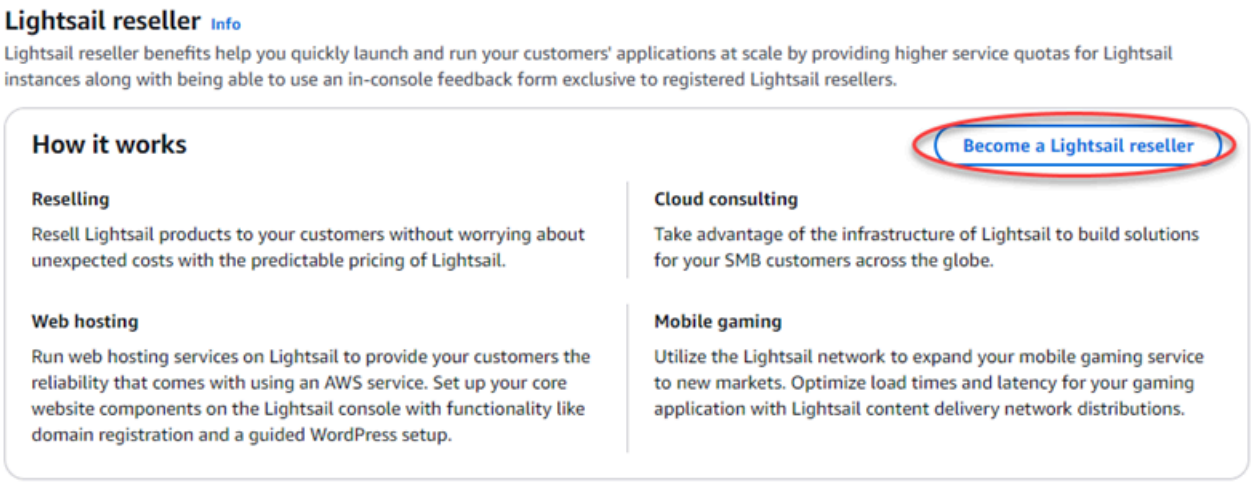
リセラーになることをリクエストするには

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。

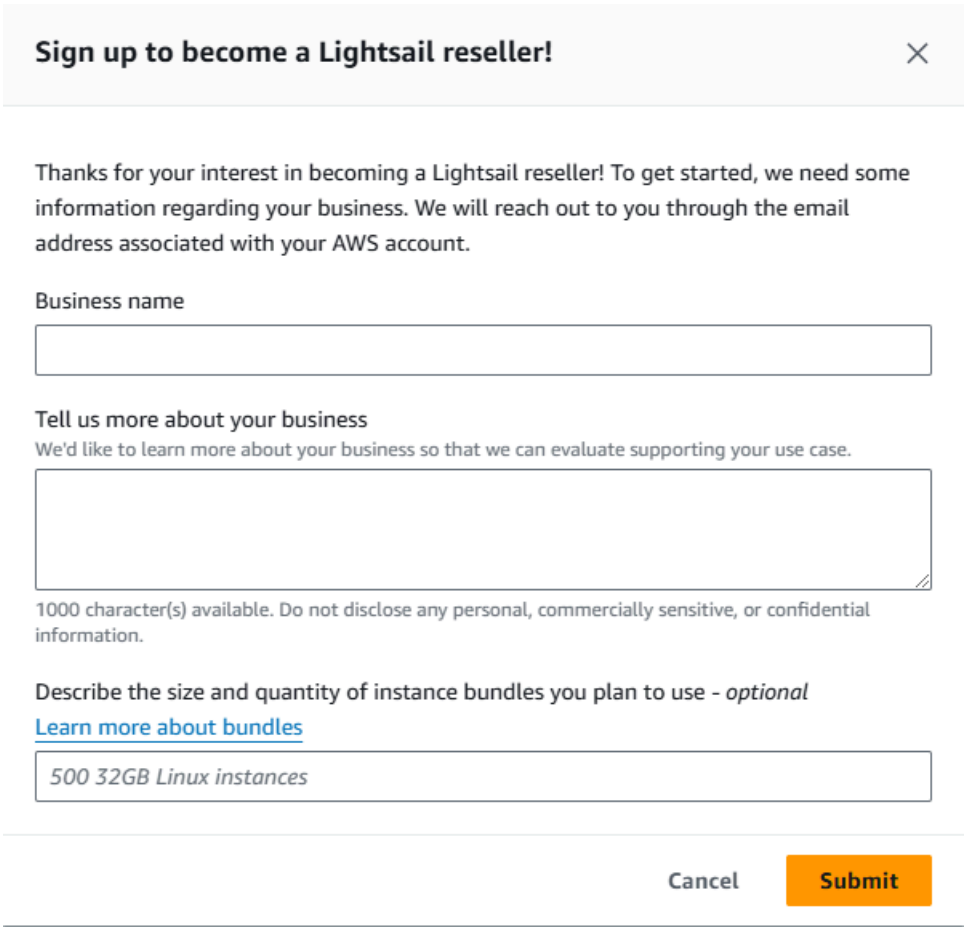
3. ドロップダウンメニューで [Account (アカウント)] を選択します。



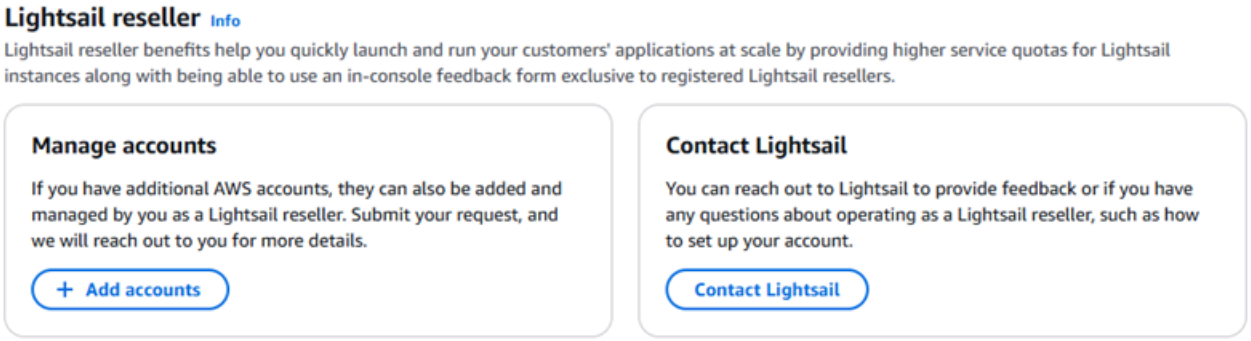
4. [プロファイル] タブの Lightsail リセラーセクションで、[Lightsail リセラーになる] を選択します。



5. 登録フォームで、フィールドに情報を入力し、[送信] を選択します。



リセラーになることへの関心について、送信内容の確認メールがアカウントの E メールに送信されます。リクエストが承認されると、Lightsail コンソールの [アカウント] ページに[Lightsail、改訂された リセラー]セクションが表示されます。ここには、リセラーアカウントを管理し、Lightsail リセラーとしてのフィードバックやクエリについて Lightsail チームに連絡するためのオプションが含まれます。このセクションは、Lightsail リセラーになるリクエストを送信したアカウントにのみ表示されます。また、Lightsail インスタンスのより高いサービスクォータを受け取り、Lightsail リセラーになるための追加 AWS アカウント をリクエストすることもできます。



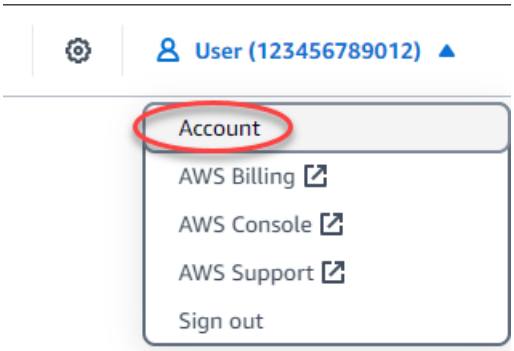
Lightsail リセラーになる追加のアカウントをリクエストする

次のステップでは、リセラーになる AWS アカウント ための追加リクエストを送信します。

> **Tip**
>
> を使用している場合は AWS Organizations、追加 AWS アカウント する として管理アカウントを指定する必要があります。このアプローチは、増加したデフォルトの Lightsail インスタンスクォータを管理アカウントの組織内のすべてのメンバーアカウントにスケーリングします。

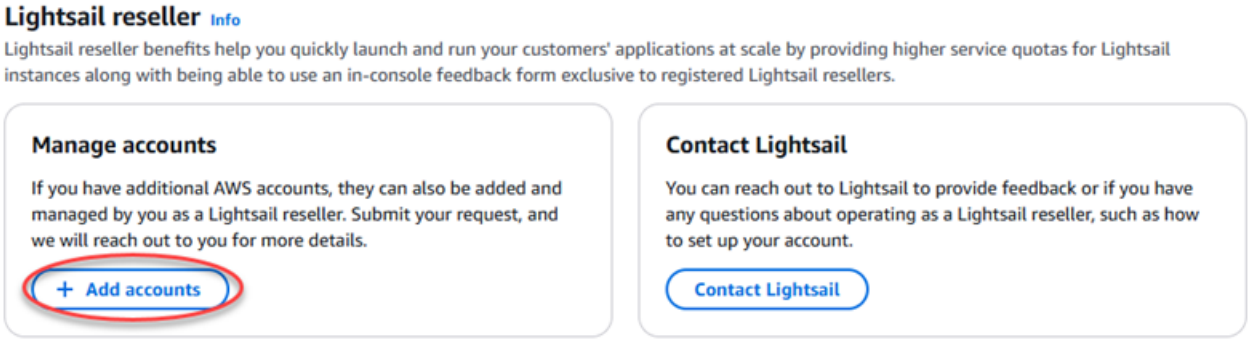
Lightsail リセラーになる追加のアカウントをリクエストするには

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
3. ドロップダウンメニューで [Account (アカウント)] を選択します。



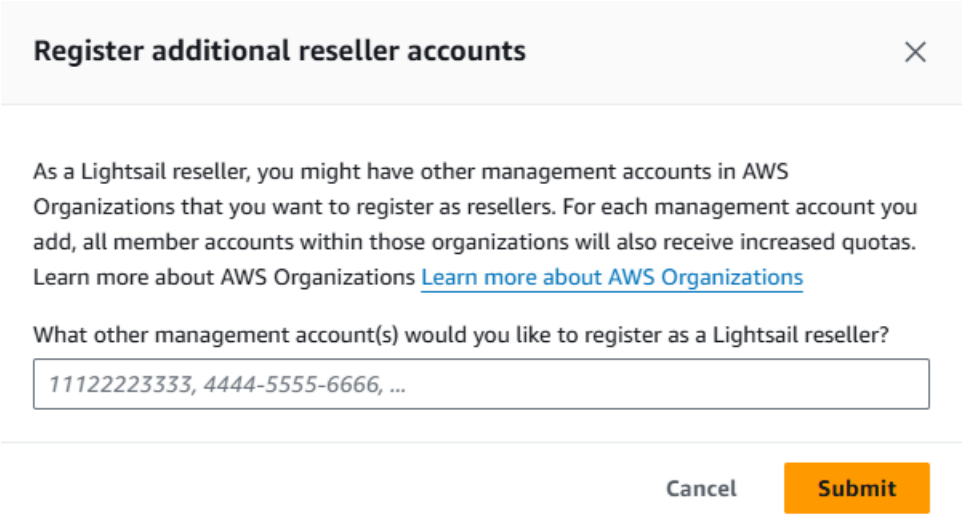
4. [プロファイル] タブの Lightsail リセラーセクションで、[アカウントを追加] を選択します。

> **Important**
>
> [アカウントを追加] のアクションは、Lightsail リセラーになることをリクエストし、承諾されたアカウントでのみ使用できます。



5. 登録フォームに、登録する組織の追加の AWS アカウント IDs または管理アカウントを入力します。

> **Note**
>
> Organizations を使用している場合は、メンバーアカウントをリクエストする必要はありません。



6. [Submit] を選択します。

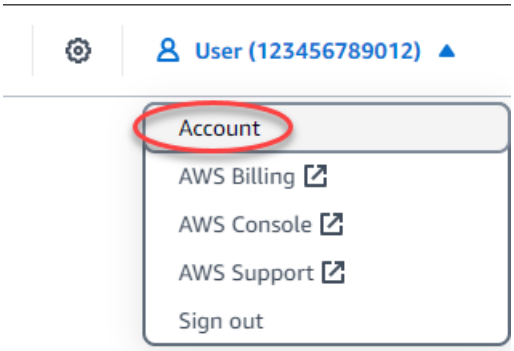
リセラーアカウントのサービスクォータの引き上げをリクエストする

Amazon Lightsail リセラーになると、Lightsail インスタンスのデフォルトのサービスクォータは、現在のアカウントと組織全体のメンバーアカウントに対して引き上げられます。メンバーアカウントの制限をさらに引き上げる場合は、次のプロセスを使用してクォータの引き上げをリクエストする必要があります。現在のクォータの表示と引き上げのリクエストは、Lightsail コンソールで行えます。

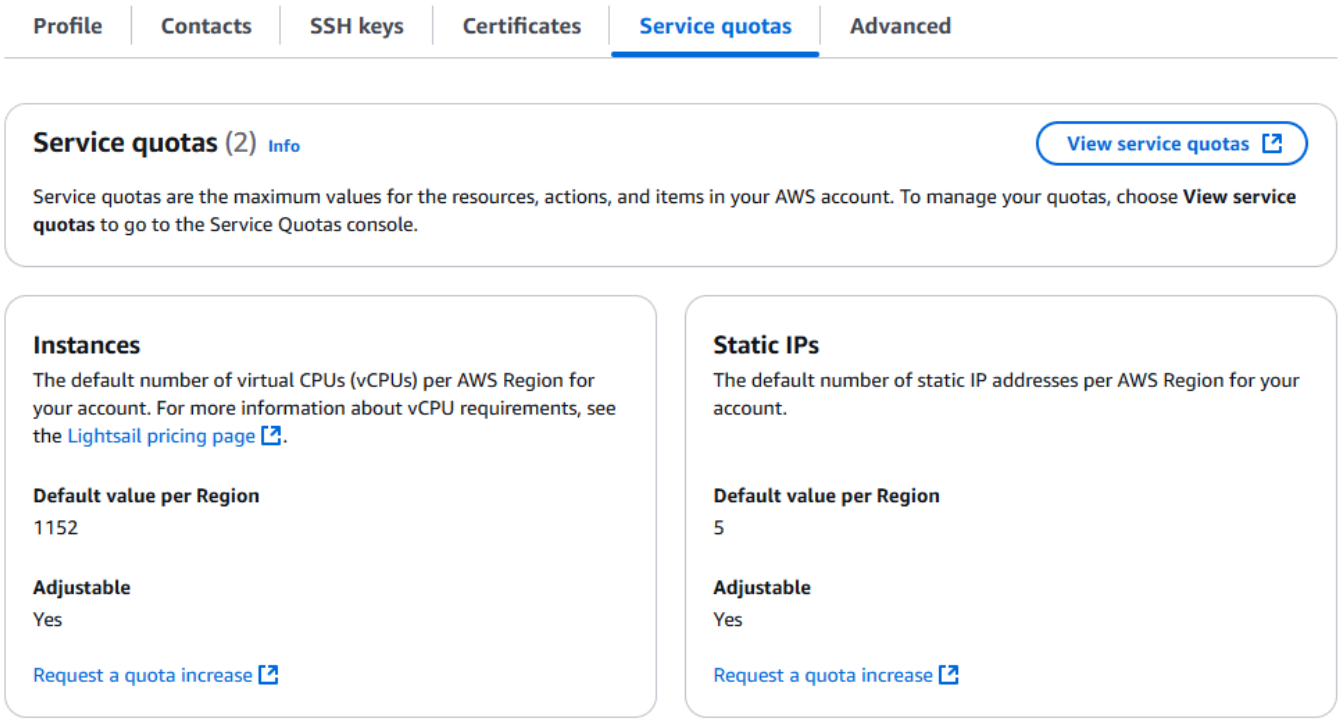
> **Note**
>
> Lightsail リセラーアカウントにリンクされている複数のメンバーアカウントについては、リセラーフィードバックフォームを使用してサービスクォータの引き上げをリクエストする必要があります。詳細については、「リセラーとして Lightsail に連絡する」を参照してください。

リセラーアカウントのサービスクォータの引き上げをリクエストするには

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
3. ドロップダウンメニューで [Account (アカウント)] を選択します。



4. [Service Quotas] タブを選択します。
5. 引き上げるクォータについては、[クォータの増額をリクエストする] を選択します。



6. Service Quotas コンソールで、[アカウントレベルでの引き上げをリクエスト] を選択します。
7. [クォータ値を引き上げる] に、数量を入力します。
8. リクエストを送信するには、[リクエスト] を選択します。

クォータ引き上げリクエストが送信されると、更新をモニタリングできる サポート ケースが生成されます。増額が承認されると、リージョンごとにすべてのリセラーアカウントに適用されます。リストにないクォータの増額については、「リセラーとして Lightsail に連絡する」を参照してください。

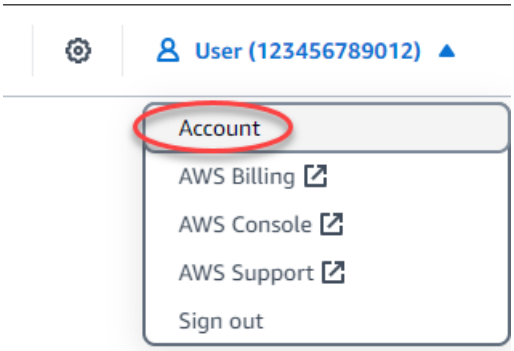
リセラーとして Lightsail に連絡する

Amazon Lightsail リセラーは、Lightsail コンソールから直接、リセラーとしての活動に関する質問やフィードバックについて Lightsail チームに連絡できます。これは、組織内のメンバーアカウント全体で Lightsail のサービスクォータの引き上げをリクエストする方法でもあります。

Lightsail チームに連絡するには

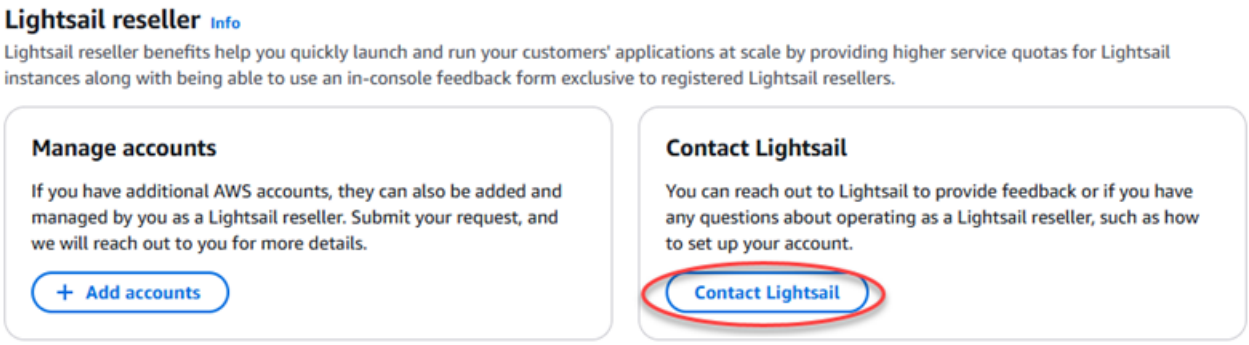
1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。

3. ドロップダウンメニューで [Account (アカウント)] を選択します。

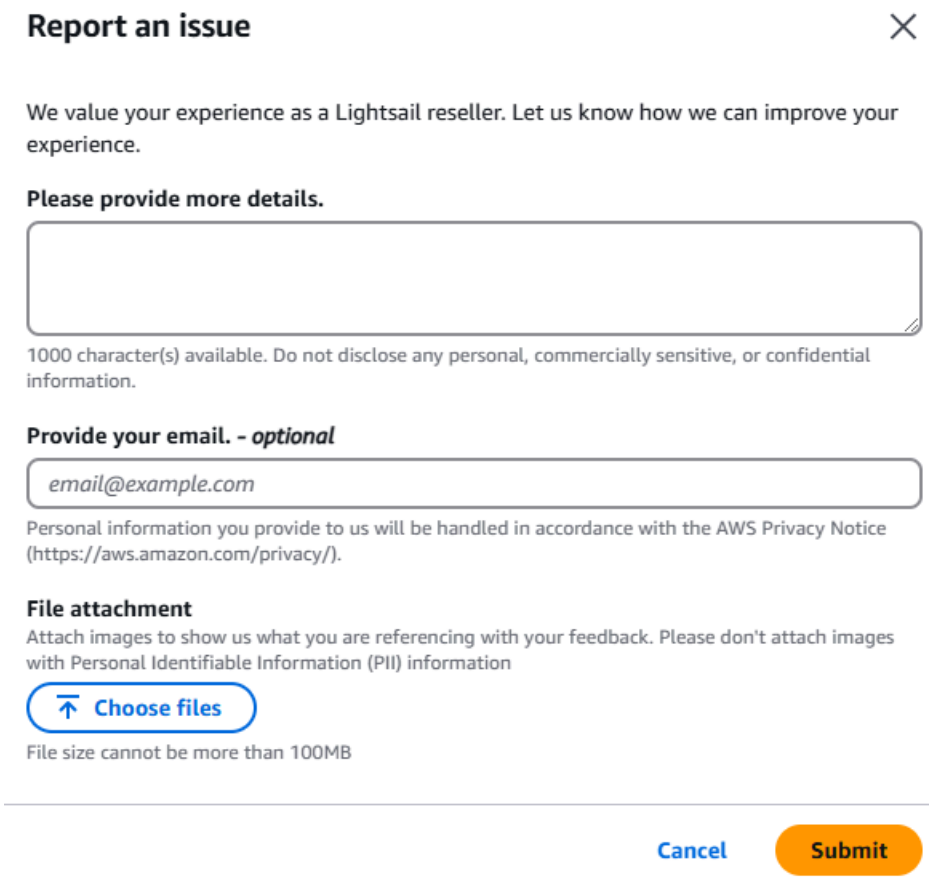


4. [プロファイル] タブの [Lightsail リセラー]セクションで、[コンタクト Lightsail] を選択します。

> **Important**
>
> [コンタクト Lightsail] アクションは、Lightsail リセラーになることをリクエストし、承諾されたアカウントでのみ使用できます。詳細については、「Lightsail リセラーになる」を参照してください。



5. リクエストに必要なフィールドに入力します。Lightsail のサービスクォータの引き上げをリクエストする場合は、複数のメンバーアカウントを指定できます。



6. [Submit] を選択してください。

E メールアドレスを指定すると、フィードバックについて連絡を受ける場合があります。

Lightsail の仮想プライベートサーバーインスタンス

Lightsail インスタンスは仮想プライベートサーバー (仮想マシンとも呼ばれる) です。インスタンスの作成時に、オペレーティングシステム (OS) が含まれているイメージを選択します。基本 OS だけでなくアプリケーションまたは開発スタックが含まれているインスタンスのイメージを選択することもできます。

オペレーティングシステム、アプリケーション、および開発フレームワークの一覧表については、「Lightsail インスタンスのイメージの選択」を参照してください。

インスタンスの詳細については、次のトピックを参照してください。

トピック

- Lightsail インスタンスを作成します。
- Lightsail インスタンスブループリントのサービスを確認する
- Lightsail インスタンスバンドル
- Lightsail のファイアウォールを使用してインスタンストラフィックを制御する
- Lightsail インスタンスバーストを検出して最適なパフォーマンスを得る
- Lightsail インスタンスに接続して管理する
- Lightsail インスタンスを削除する
- SSH キーペアを管理し、Lightsail インスタンスに接続する
- Lightsail におけるインスタンスメタデータサービス (IMDS) とユーザーデータをアクセスする

Lightsail インスタンスを作成します。

このセクションでは、Amazon Lightsail でのインスタンスの作成に関連する以下のトピックについて説明します:

トピック

- Lightsail でアプリケーションを使用して Linux/Unix インスタンスを作成する
- Lightsail で Windows Server インスタンスを作成する

Lightsail でアプリケーションを使用して Linux/Unix インスタンスを作成する

> **知っていましたか?**
>
> Lightsail は毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを最新のスナップショットに自動的に置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

WordPress などのアプリケーションまたは LAMP などの開発スタックを実行する Linux/Unix ベースの Amazon Lightsail インスタンス (仮想プライベートサーバー) を作成します。インスタンスの実行が開始されると、Lightsail を終了せずに SSH 経由でそのインスタンスに接続できます。その方法は次のとおりです。

Windows ベースのインスタンスを作成するには、「Amazon Lightsail で、Windows ベースのインスタンスの使用を開始する」を参照してください。

Linux ベースのインスタンスを作成する

1. ホームページで [インスタンスの作成] を選択します。
2. インスタンスの場所 (AWS リージョン とアベイラビリティーゾーン) を選択します。

   変更 AWS リージョン とアベイラビリティーゾーンを選択して、別の場所にインスタンスを作成します。
3. 必要に応じて、アベイラビリティーゾーンを変更できます。

   [アベイラビリティーゾーンの変更] を選択する。
4. Linux プラットフォームを選択します。
5. アプリケーション ([アプリ + OS]) またはオペレーティングシステム ([OS のみ]) を選択します。

   Lightsail インスタンスイメージの詳細については、「Amazon Lightsail インスタンスのイメージの選択」を参照してください。
6. インスタンスプランを選択します。

   インスタンスがデュアルスタック (IPv4 と IPv6) を使用するか IPv6 専用ネットワークを使用するかを選択します。一部の Lightsail ブループリントは、現在のところ IPv6 専用ネットワークを

サポートしていません。どのブループリントが IPv6 専用のネットワーキングをサポートしているかを確認するには、Lightsail インスタンスブループリントのサービスを確認する を参照してください。

5 米ドルの Lightsail プランを、1 か月間 (最大 750 時間) 無料でお試しいただけます。1 か月の無料期間分はアカウントに返金されます。詳細については、Lightsail の料金ページを参照してください。

> **Note**
>
> AWS 無料利用枠の一部として、一部のインスタンスバンドルで Amazon Lightsailを無料で開始できます。詳細については、「Amazon Lightsail 料金表」ページの「AWS Free Tier」を参照してください。

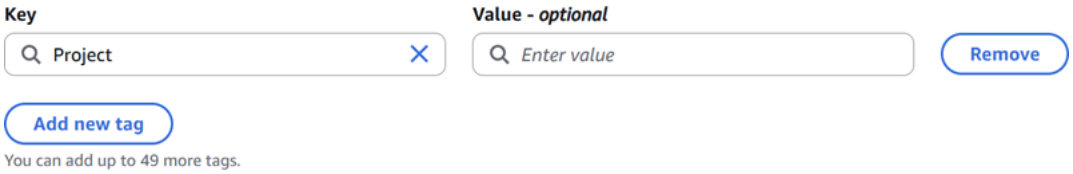
7. インスタンスの名前を入力します。

   リソース名:

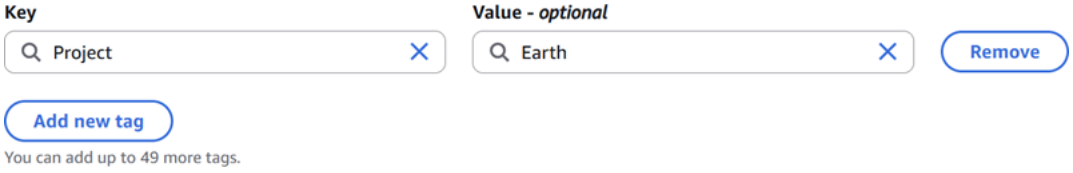
   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

8. (オプション)[新しいタグの追加] を選択してインスタンスにタグを追加します。必要に応じてこの手順を繰り返して、他のタグを追加します。タグの使用の詳細については、「タグ」を参照してください。

   a. [キー] には、タグキーを入力します。

   b. (オプション) [値] にタグの値を入力します。

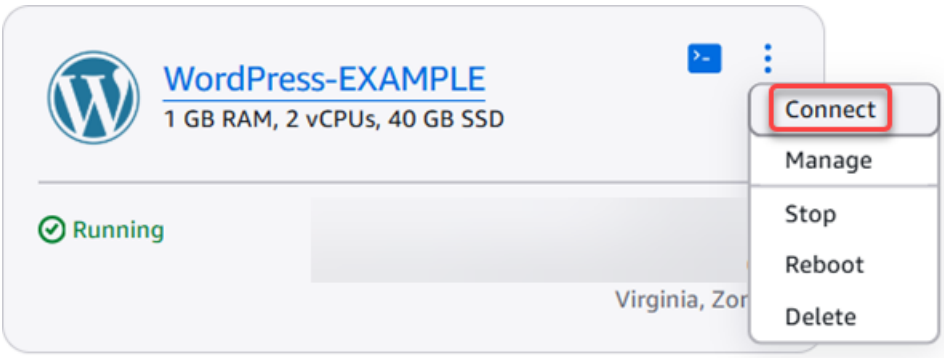
9. [インスタンスの作成] を選択します。

高度な作成オプションの詳細については、「Amazon Lightsail インスタンスの起動時にシェルスクリプトを使用して設定する」または「Linux/Unix ベースの Lightsail インスタンス向けに SSH をセットアップする」を参照してください。

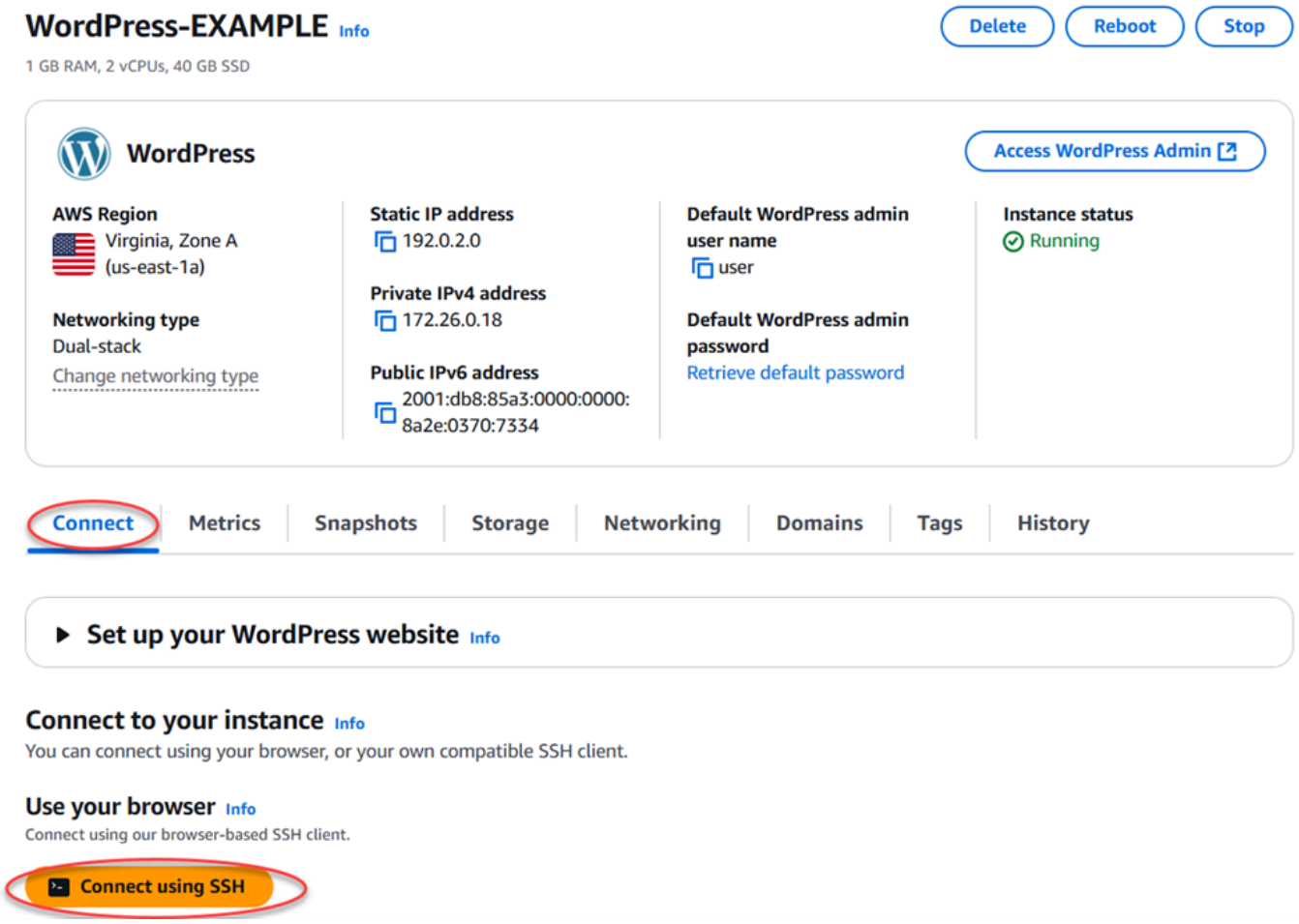
数分で Lightsail インスタンスの準備が整い、Lightsail を終了せずに SSH 経由でそのインスタンスに接続できます。

インスタンスへの接続

1. Lightsail のホームページで、インスタンス名の右側にあるメニューを選択して、[接続] を選択します。

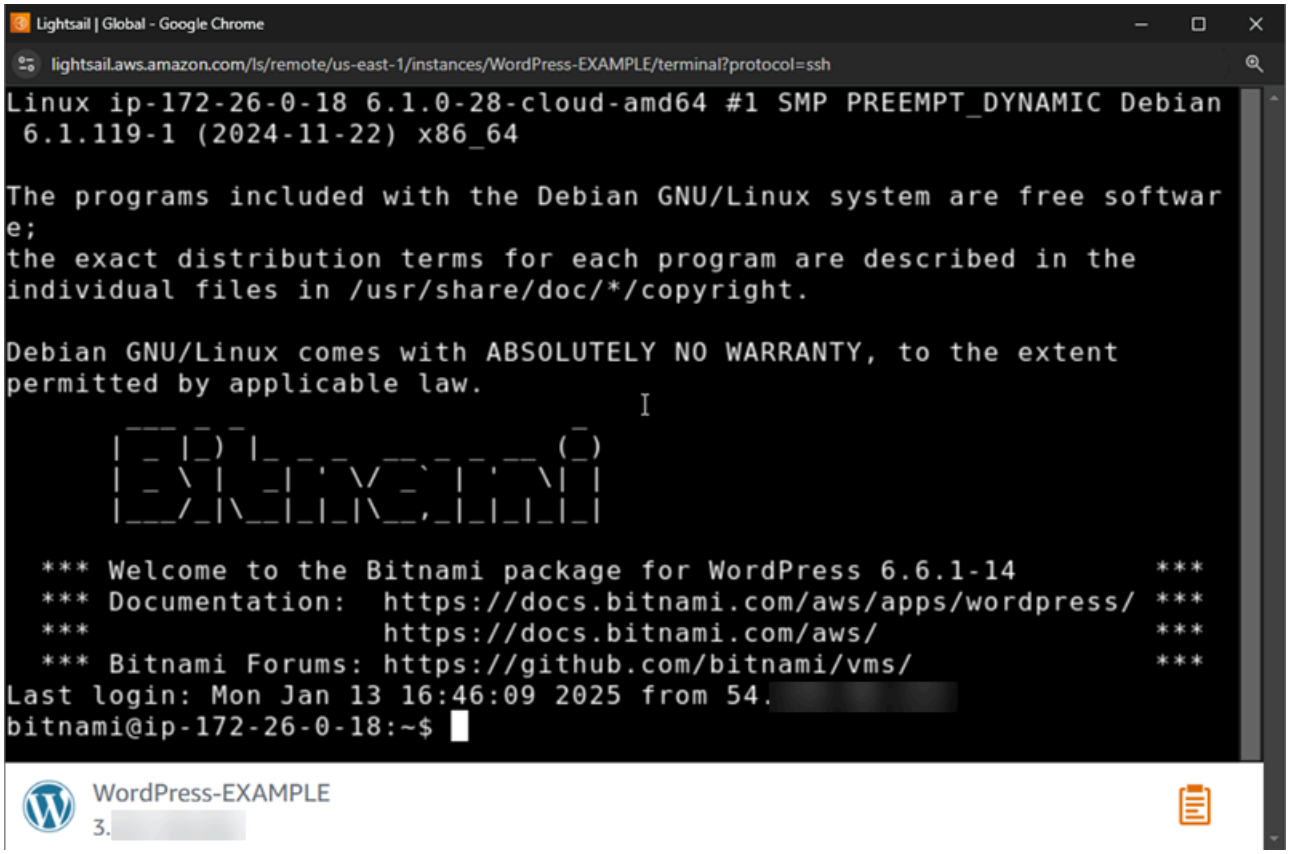


   または、インスタンス管理ページを開き、[接続] タブを選択した後、[SSH を使用して接続] を選択することもできます。



> **Note**
>
> PuTTy などの SSH クライアントを使用してインスタンスに接続するには、「PuTTy をセットアップして Lightsail インスタンスに接続する」の手順に従ってください。

2. これで、SSH クライアントをセットアップせずに、ターミナルにコマンドを入力して Lightsail インスタンスを管理できるようになりました。



次の手順

インスタンスに接続できるようになった後、次に行う作業はお客様の使用計画によって異なります。例えば、次のようになります。

- the section called “WordPress” (ブログを作成する場合)。
- インスタンスの再起動時にインスタンスで同じ IP アドレスを保持する場合は、静的 IP アドレスを作成Lightsailします。
- バックアップとしてインスタンスのスナップショットを作成します。

Lightsail で Windows Server インスタンスを作成する

> **知っていましたか?**
>
> Lightsail は毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを最新のスナップショットに自動的に置き換えま

す。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

Windows Server オペレーティングシステム (OS) を実行する Lightsail インスタンスを作成します。3 つの OS ブループリント (Windows Server 2022、Windows Server 2019、Windows Server 2016) を利用できます。また、SQL Server Express 2022、2019、2016 では、あらかじめ設定されたブループリントが用意されています。

このトピックでは、ソフトウェアの選択、Windows Server ベースのインスタンスの作成、それに接続する方法について説明します。

AWS での Windows Server の詳細

Windows Server ベースのインスタンスを選択する

Lightsail で Windows Server ベースのインスタンスを作成するオプションは 3 つあります。

[Windows Server 2022]

Windows Server を実行する Lightsail は、Microsoft Web Platform を使用してアプリケーションをデプロイするための高速で信頼できるプラットフォームです。Lightsail を利用すれば、高性能で信頼性があり、費用対効果の高い、 AWS クラウド コンピューティングプラットフォーム上で、互換性のある任意の Windows ベース ソリューションを実行することができます。一般的な Windows ユースケースには、Enterprise Windows ベースのアプリケーションホスティング、ウェブサイトおよびウェブサービスホスティング、データ処理、分散テスト、ASP.NET アプリケーションホスティング、そして Windows ソフトウェアを必要とする、他のすべてのアプリケーションが含まれます。

Windows Server 2022 イメージの詳細について説明します

Windows Server 2019

何らかの理由で Windows Server 2016 または Windows Server 2019 を実行する必要がある場合を除き、Windows Server 2022 の最新バージョンを使用することをお勧めします。

Lightsail Windows Server を実行することは、Microsoft Web Platform を使用してアプリケーションをデプロイするための高速で信頼性の高い環境です。 Lightsail を使用すると、互換性のある Windows ベースのソリューションを、 AWS高性能で信頼性が高く、費用対効果の高いクラウドコンピューティングプラットフォームで実行できます。一般的な Windows ユースケースには、Enterprise Windows ベースのアプリケーションホスティング、ウェブサイトおよびウェ

ブサービスホスティング、データ処理、分散テスト、ASP.NET アプリケーションホスティング、Windows ソフトウェアが必要な他のあらゆるアプリケーションが含まれます。

Windows Server 2019 イメージの詳細

Windows Server 2016

何らかの理由で Windows Server 2016 または Windows Server 2019 を実行する必要がある場合を除き、Windows Server 2022 の最新バージョンを使用することをお勧めします。

Windows Server を実行する Lightsail は、Microsoft Web Platform を使用してアプリケーションをデプロイするための高速で信頼できる環境です。Lightsail を利用すれば、AWS の高性能で信頼性が高く、費用対効果に優れたクラウドコンピューティングプラットフォームで、すべての互換性のある Windows ベースのソリューションを実行できます。一般的な Windows ユースケースには、Enterprise Windows ベースのアプリケーションホスティング、ウェブサイトおよびウェブサービスホスティング、データ処理、分散テスト、ASP.NET アプリケーションホスティング、Windows ソフトウェアが必要な他のあらゆるアプリケーションが含まれます。

Windows Server 2016 イメージの詳細

[SQL Server Express 2022]

SQL Server Express は、無料でダウンロード、配布、使用できるリレーショナルデータベース管理システムです。小規模な埋め込みアプリケーションを特にターゲットとするデータベースを構成しています。この Lightsail イメージは、Windows Server 2022 の基本 OS で実行されます。

SQL Server Express 2022 イメージの詳細

[SQL Server Express 2019]

SQL Server Express は、無料でダウンロード、配布、使用できるリレーショナルデータベース管理システムです。小規模な埋め込みアプリケーションを特にターゲットとするデータベースを構成しています。この Lightsail イメージは、Windows Server 2022 の基本 OS で実行されます。

SQL Server Express 2019 イメージの詳細

[SQL Server Express 2016]

SQL Server Express は、無料でダウンロード、配布、使用できるリレーショナルデータベース管理システムです。小規模な埋め込みアプリケーションを特にターゲットとするデータベースを構成しています。この Lightsail イメージは、Windows Server 2016 の基本 OS で実行されます。

SQL Server Express イメージの詳細

Windows Server ベースの インスタンスを作成する

Windows Server ベースのインスタンスは、 Lightsailコンソールまたは AWS Command Line Interface () を使用して作成できますAWS CLI。

コンソールを使用してインスタンスを作成するには

1. Lightsail にサインインし、ホームページに移動します。
2. [インスタンスの作成] を選択します。
3. Windows Server ベースのLightsailインスタンス AWS リージョン を作成する を選択します。

   例えば、Ohio (us-east-2)。
4. [Microsoft Windows] プラットフォームを選択します。
5. Windows Server 2022、Windows Server 2019、Windows Server 2016 のブループリントを選択するには、[OS のみ] を選択します。

   SQL Server Express 設計図を選択するには、[アプリ + OS] を選択します。
6. インスタンスプランを選択します。

   インスタンスがデュアルスタック (IPv4 と IPv6) を使用するか IPv6 専用ネットワークを使用するかを選択します。一部の Lightsail ブループリントは、現在のところ IPv6 専用ネットワークをサポートしていません。どのブループリントが IPV6 専用ネットワークをサポートしているかを確認するには、Lightsail インスタンスブループリントのサービスを確認する を参照してください。

   プランには、低額で予測可能なコスト、マシン設定 (RAM、SSD、vCPU)、データ転送も含まれます。

   > **Note**
   >
   > 設計図によっては、一部のインスタンスプランを使用できません。例えば、SQL Server Express ブループリントでは、少なくとも 4 GB のメモリと 80 GB の SSD ストレージを備えたプランを使用する必要があります。
7. インスタンスの名前を入力します。

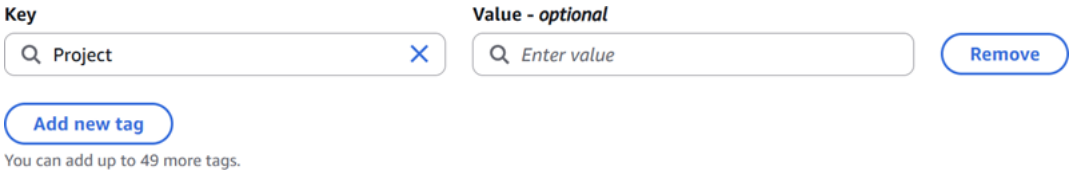
   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。

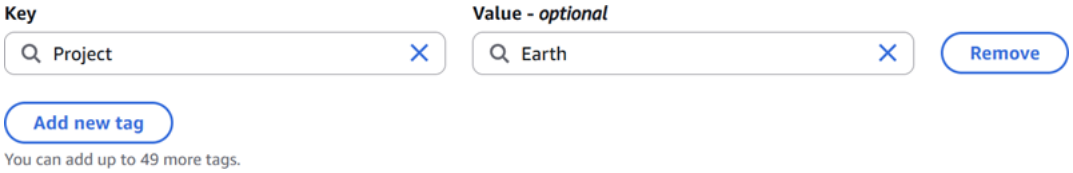
- 2〜255 文字を使用する必要があります。
- 先頭と末尾は英数字または数字を使用する必要があります。
- 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

8. (オプション)[新しいタグの追加] を選択してインスタンスにタグを追加します。必要に応じてこの手順を繰り返して、他のタグを追加します。タグの使用の詳細については、「タグ」を参照してください。

   a. [キー] には、タグキーを入力します。

   b. (オプション) [値] にタグの値を入力します。

9. [インスタンスの作成] を選択します。

を使用してインスタンスを作成するには AWS CLI

1. まだ AWS CLIをインストールして設定していない場合は、インストールして設定します。

   詳細については、「Amazon Lightsail で使用するために AWS Command Line Interface を設定する」を参照してください。

2. コマンドプロンプトまたはターミナルウィンドウを開きます。
3. まだ設定していない場合は、 AWS CLI を使用して を設定しaws configure、Lightsailリソースを作成する AWS リージョン を選択します。
4. 次の AWS CLI コマンドを入力して、オハイオリージョンで実行されている 1 か月あたり 44 USD の Windows Server 2022 インスタンスを作成します。

```
aws lightsail create-instances --instance-names InstanceName --availability-zone
 us-east-2a --blueprint-id windows_server_2022 --bundle-id medium_win_3_0
```

   コマンドで、*InstanceName* を新しいインスタンスの名前に置き換えます。

成功すると、 AWS CLIから次の出力が表示されます。

```
{
    "operations": [
        {
            "status": "Started",
            "resourceType": "Instance",
            "isTerminal": false,
            "statusChangedAt": 1508086226.4,
            "location": {
                "availabilityZone": "us-east-2a",
                "regionName": "us-east-2"
            },
            "operationType": "CreateInstance",
            "resourceName": "my-windows-instance",
            "id": "344acdc8-f9c4-4eda-8232-12345EXAMPLE",
            "createdAt": 1508086225.467
        }
    ]
}
```

> **Note**
>
> 使用可能な設計図の一覧を取得するには、get-blueprints コマンドを使用します。使用可能なバンドルの一覧を取得するには、get-bundles コマンドを使用します。get-instance-access-details コマンドを使用してインスタンスのパスワードを取得する方法について、詳細を確認してください。

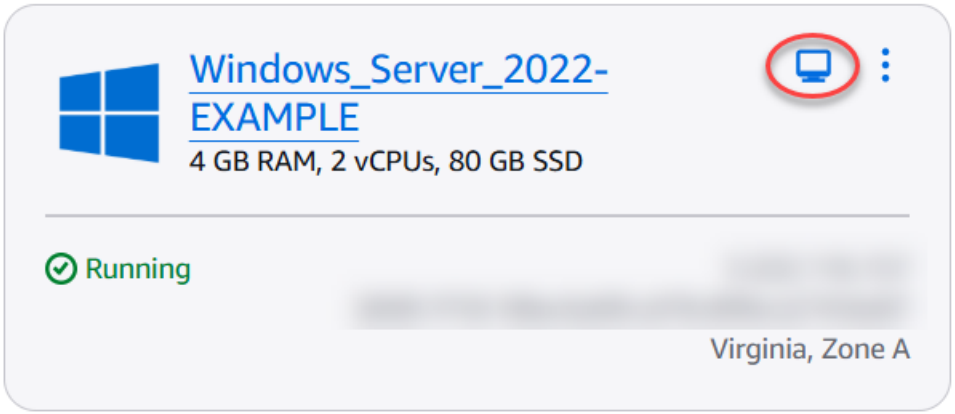
インスタンスへの接続

Windows Server ベースの Lightsail インスタンスを作成すると、ブラウザベースの RDP クライアントまたは選択したリモートデスクトップクライアントを使用してそのインスタンスに接続できます。

> **Note**
>
> インスタンスを作成した後、インスタンスに接続できるようになるまで最大 15 分かかります。

Lightsail のブラウザベースの RDP クライアントを使用して接続するには

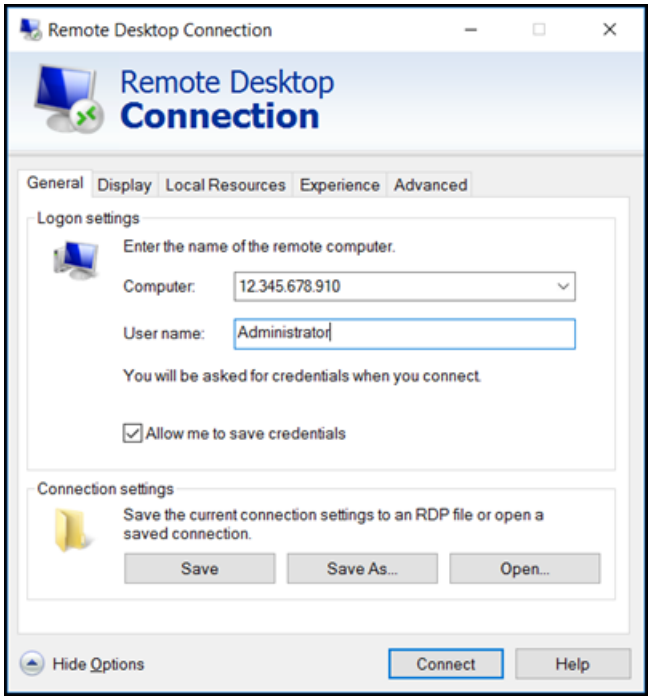
1. ホームページで、インスタンスの横の [RDP を使用して接続] アイコンを選択します。



2. または、ショートカットメニューやインスタンス管理ページからインスタンスに接続することもできます。

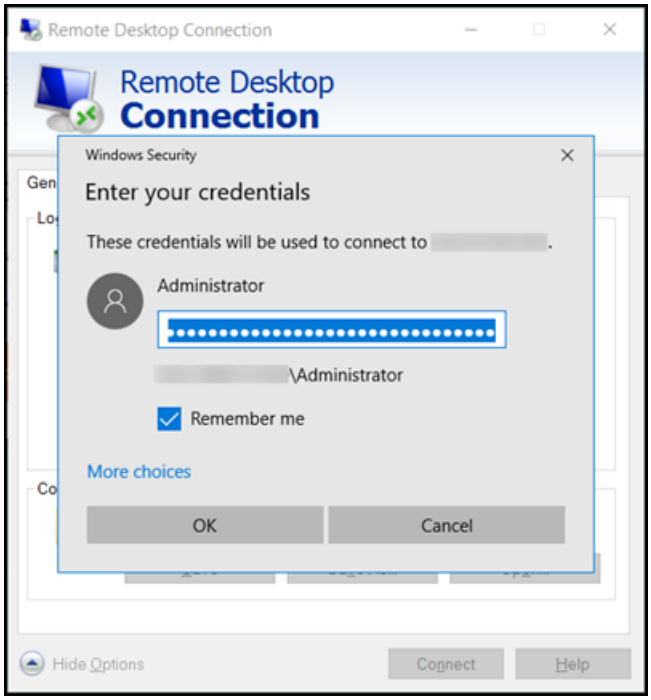
独自の RDP クライアントを使用して接続するには

1. IP アドレスを取得するには、Lightsail のホームページに移動します。
2. IP アドレスをクリップボードにコピーします。
3. Windows のリモートデスクトップ接続などの RDP クライアントを開きます。
4. IP アドレスを [コンピューター] フィールドに貼り付けます。
5. [オプションの表示] を選択し、[Administratorユーザー名] に「」と入力します。



6. 接続 を選択します。
7. パスワードを取得するには、Lightsail でインスタンス管理ページに移動します。

   インスタンス管理ページに移動するには、 ホームページでインスタンスの名前を選択します (またはショートカットメニューから [管理Lightsail] を選択します)。
8. [デフォルトのパスワードを表示] を選択します。
9. デフォルトパスワードをクリップボードにコピーします。
10. パスワードを [リモートデスクトップ接続] に貼り付け、[このアカウントを記憶する] を選択して今後はこのダイアログボックスが表示されないようにします。



11. [OK] を選択してください。
12. [Don't ask me again for connections to this computer (このコンピューターでは接続の確認をしない)] を選択し、[Yes (はい)] を選択します。

次の手順

インスタンスに接続できるようになった後、次に行う作業はお客様の使用計画によって異なります。例えば、次のようになります。

- インスタンスの再起動時にインスタンスで同じ IP アドレスを保持する場合は、静的 IP アドレスを作成Lightsailします。
- Lightsail Windows Server インスタンスのスナップショットを作成します。

ステップバイステップの手順に従って、Amazon Linux、Ubuntu、Debian、または Windows Server 2022、2019、2016 などの Windows Server オペレーティングシステムなどの Linux および Unix ディストリビューションを実行するインスタンスを作成します。

Linux および Unix インスタンスでは、WordPress、LAMP、LEMP などのさまざまなアプリケーション設計図から選択することも、オペレーティングシステムのみを選択することもできます。Windows Server インスタンスの場合、Windows Server ブループリントまたは SQL Server Express ブループリントから選択できます。

このガイドでは、 AWS リージョン とアベイラビリティーゾーンの選択、必要なコンピューティングリソースとストレージリソースを含むインスタンスプラン (バンドル) の選択、IPv4 や IPv6 などのネットワークオプションの設定、インスタンスの名前付け、タグの追加について説明します。インスタンスを作成したら、Lightsail ブラウザベースの SSH クライアントまたは RDP クライアントを使用してインスタンスに接続するか、提供された接続の詳細で独自の SSH クライアントまたは RDP クライアントを使用できます。このガイドに従うことで、特定の要件に合わせて、Lightsail で Linux および Unix または Windows Server インスタンスをすばやく起動してアクセスできます。

Lightsail インスタンスブループリントのサービスを確認する

Lightsail では、仮想プライベートサーバーを作成するための複数のオプションが用意されています。このトピックは、自分のプロジェクトに適したオペレーティングシステム (OS)、アプリケーションスタック、または開発スタックを決定するのに役立ちます。アプリケーションは、機能領域 (CMS や e コマースなど) 別に整理されています。

オペレーティングシステム

Lightsail には、いくつかの Linux/Unix ベースまたは Windows ベースのオペレーティングシステムがあります。

[Windows Server 2022]

Windows Server を実行する Lightsail は、Microsoft Web Platform を使用してアプリケーションをデプロイするための高速で信頼できるプラットフォームです。Lightsail を利用すれば、高性能で信頼性があり、費用対効果の高い、 AWS クラウド コンピューティングプラットフォーム上で、互換性のある任意の Windows ベース ソリューションを実行することができます。一般的な Windows ユースケースには、Enterprise Windows ベースのアプリケーションホスティング、ウェブサイトおよびウェブサービスホスティング、データ処理、分散テスト、ASP.NET アプリケーションホスティング、そして Windows ソフトウェアを必要とする、他のすべてのアプリケーションが含まれます。サポート終了情報については、「Microsoft」ウェブサイトを参照してください。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

Windows Server 2022 の詳細について。

Windows Server 2019

Windows Server を実行する Lightsail は、Microsoft Web Platform を使用してアプリケーションをデプロイするための高速で信頼できる環境です。Lightsail を利用すれば、高性能で信頼性が高

く、費用対効果に優れた AWS のクラウドコンピューティングプラットフォーム上で、互換性のある任意の Windows ベースソリューションを実行できます。一般的な Windows ユースケースには、Enterprise Windows ベースのアプリケーションホスティング、ウェブサイトおよびウェブサービスホスティング、データ処理、分散テスト、ASP.NET アプリケーションホスティング、そして Windows ソフトウェアを必要とする、他のすべてのアプリケーションが含まれます。サポート終了情報については、「Microsoft」ウェブサイトを参照してください。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

Windows Server 2019 の詳細について。

Windows Server 2016

Windows Server を実行する Lightsail は、Microsoft Web Platform を使用してアプリケーションをデプロイするための高速で信頼できる環境です。Lightsail を利用すれば、高性能で信頼性が高く、費用対効果に優れた AWS のクラウドコンピューティングプラットフォーム上で、互換性のある任意の Windows ベースソリューションを実行できます。一般的な Windows ユースケースには、Enterprise Windows ベースのアプリケーションホスティング、ウェブサイトおよびウェブサービスホスティング、データ処理、分散テスト、ASP.NET アプリケーションホスティング、そして Windows ソフトウェアを必要とする、他のすべてのアプリケーションが含まれます。サポート終了情報については、「Microsoft」ウェブサイトを参照してください。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

Windows Server 2016 の詳細について。

Amazon Linux 2023

Amazon Linux 2023 (AL2023) は、 AWSでの汎用ワークロード向けとして理想的な次世代型 Amazon Linux です。AL2023 には、一般公開の開始後 5 年間サポートが提供されます。AL2023 は Amazon Linux パッケージリポジトリの特定のバージョンにロックされるため、更新を適用するタイミングと方法を管理できます。また、AL2023 では、頻繁に行われる更新や、コンプライアンスのニーズを満たすのに役立つ機能を取得することもできます。

AL2023 から起動された Lightsail インスタンスには、Instance Metadata Service Version 2 (IMDSv2) がデフォルトで適用されます。詳細については、「インスタンスメタデータサービスバージョン 2 の仕組み」を参照してください。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

Amazon Linux 2023 の詳細について。

Amazon Linux 2

> **Important**
>
> Amazon Linux 2 は 2026 年 6 月 30 日に長期サポート終了になります。2026 年 6 月 30 日以降、このブループリントを使用して新しいLightsailインスタンスを作成することはできません。詳細については、Amazon Linux 2 ウェブサイトを参照してください。

Amazon Linux 2 は、 AWSの Linux サーバーオペレーティングシステムでは、前世代の Amazon Linux です。これはクラウドおよびエンタープライズアプリケーションの開発と実行のために設計された、安定した安全で高性能な実行環境を提供します。Amazon Linux 2 では、Linux での最新のイノベーションへのアクセスを含む長期的なサポートを提供する、アプリケーション環境を取得できます。Amazon Linux 2 には追加料金はかかりません。サポート終了情報については、「Amazon Linux 2 に関するよくある質問」を参照してください。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

Amazon Linux 2 の詳細について。

AlmaLinux OS 9

AlmaLinux OS 9 は、オープンソースでコミュニティによって所有· 運営される、永続的に無料のエンタープライズ向け Linux ディストリビューションです。長期的な安定性に重点を置き、堅牢なプロダクション向けプラットフォームを提供します。AlmaLinux は、RHEL® および pre-Stream CentOS と互換性があります。サポート終了情報については、「AlmaLinux OS Foundation」ウェブサイトを参照してください。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

AlmaLinux OS 9 の詳細について。

CentOS Stream 9

CentOS Stream 9 は、CentOS Stream ディストリビューションでの次のメジャーリリースです。CentOS Stream 9 は、Red Hat Enterprise Linux (RHEL) の開発の直前に一歩先んじて、継続的に配信されるディストリビューションであり、Fedora Linux と RHEL の中間的な位置付けとなっています。これは、RHEL と機能的な互換性を持つように設計されており、提供される Linux 環境は安定で、予測性、管理性、再現性を備えています。サポート終了情報については、「CentOS ウェブサイト」を参照してください。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

詳細については、CentOS Stream ウェブサイトを参照してください。

Debian 11、12、13

> **Important**
>
> Debian 11 は 2026 年 8 月 31 日に長期サポート終了になります。2026 年 8 月 31 日以降、このブループリントを使用して新しいLightsailインスタンスを作成することはできません。詳細については、Debian ウェブサイトを参照してください。

Debian は無料のオペレーティングシステムであり、インターネット上で協力しあう、世界中の何千人ものボランティアによって開発されました。Debian プロジェクトの主な強みは、ボランティアの基盤、Debian 社会契約およびフリーソフトウェアへの献身、可能な限り最高のオペレーティングシステムを提供するというコミットメントです。この新しいリリースは、その方向へ向かうためにもう 1 つの重要なステップです。サポート終了情報については、「Debian ウェブサイト」を参照してください。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

詳細については、「Debian」のウェブサイトを参照してください。

FreeBSD 13、14、15

> **Important**
>
> FreeBSD 13 は 2026 年 4 月 30 日にサポート終了になります。2026 年 4 月 30 日以降、このブループリントで新しいLightsailインスタンスを作成することはできません。詳細については、FreeBSD ウェブサイトを参照してください。

FreeBSD は、パワーサーバー、デスクトップ、および組み込みシステムに使用されるオペレーティングシステムです。BSD から派生した、カリフォルニア大学バークレー校で開発された UNIX のバージョンである FreeBSD は、30年以上にわたり大規模なコミュニティによって継続的に開発されています。pf ファイアウォール、Capsicum や CloudABI の能力フレームワーク、ZFS ファイルシステム、DTrace 動的トレーシングフレームワークなど、FreeBSD のネットワーキング、セキュリティ、ストレージ、監視機能により、FreeBSD は最も忙しいウェブサイトや広く普及している組み込み型ネットワークおよびストレージシステムにおいて選ばれるプラットフォー

ムとなっています。サポート終了情報については、「FreeBSD」のウェブサイトを参照してください。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

詳細については、「FreeBSD」のウェブサイトを参照してください。

openSUSE 15

openSUSE ディストリビューションは、安定していて使いやすい、包括的な汎用 Linux ディストリビューションです。openSUSE は、デスクトップやサーバーで作業するユーザーおよび開発者を対象としています。openSUSE は、初心者、経験豊富なユーザー、およびマニアックなユーザーなどに最適であり、つまり誰にとっても申し分ありません。サポート終了情報については、「openSUSE」のウェブサイトを参照してください。

このオペレーティングシステムでは、デフォルトでパスワード認証は無効になっています。つまり、パスワード認証が有効になっているインスタンスのスナップショットからインスタンスを作成しても、新しいインスタンスではパスワード認証が無効になります。SUSE Linux のパスワード認証の詳細については、SUSE ドキュメントのドキュメント 3404214 を参照してください。

パスワード認証が無効な状態でインスタンスにログインするには、Lightsail コンソールまたはキーペアにブラウザベースの SSH クライアントを使用できます。ログインの詳細については、「Lightsail で Linux または Unix インスタンスに接続」または「SSH コマンドで Lightsail Linux または Unix インスタンスに接続」を参照してください。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

詳細については、「openSUSE」のウェブサイトを参照してください。

Ubuntu 22 および 24

Ubuntu Server は Debian ベースの Linux オペレーティングシステムであり、仮想サーバーに使用されます。Ubuntu のデフォルトのインストールには、LibreOffice、Firefox、Thunderbird、Transmission などの幅広いソフトウェアが含まれています。APT ベースのパッケージ管理ツール (`apt-get`) を使用して、Evolution、GIMP、Pidgin、Synaptic など、多数の追加ソフトウェアパッケージをインストールできます。サポート終了情報については、「Ubuntu」ウェブサイトを参照してください。

Ubuntu 24 ブループリントで作成された Lightsail インスタンスには、デフォルトでインスタンスメタデータサービスバージョン 2 (IMDSv2) が実施されます。詳細については、「インスタンスメタデータサービスバージョン 2 の仕組み」を参照してください。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

詳細については、「Ubuntu」のウェブサイトを参照してください。

データベースアプリケーション

Lightsail では次のデータベースアプリケーションを使用できます:

SQL Server 2022 Express

SQL Server Express は、無料でダウンロード、配布、使用できるリレーショナルデータベース管理システムです。小規模な埋め込みアプリケーションを特にターゲットとするデータベースを構成しています。この Lightsail イメージは、Windows Server 2022 の基本 OS で実行されます。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

「SQL Server 2022 Express」の詳細について。

SQL Server 2019 Express

SQL Server Express は、無料でダウンロード、配布、使用できるリレーショナルデータベース管理システムです。小規模な埋め込みアプリケーションを特にターゲットとするデータベースを構成しています。この Lightsail イメージは、Windows Server 2022 の基本 OS で実行されます。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

「SQL Server 2019 Express」の詳細について。

SQL Server 2016 Express

> **Important**
>
> SQL Server 2016 Express は、2026 年 7 月 14 日に延長サポートを終了する予定です。2026 年 7 月 14 日以降、このブループリントを使用して新しいLightsailインスタンスを作成することはできません。詳細については、Microsoft のウェブサイトを参照してください。

SQL Server Express は、無料でダウンロード、配布、使用できるリレーショナルデータベース管理システムです。小規模な埋め込みアプリケーションを特にターゲットとするデータベースを構成しています。この Lightsail イメージは、Windows Server 2016 の基本 OS で実行されます。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

「SQL Server 2016 Express」の詳細について

CMS アプリケーション

Lightsail では、以下のコンテンツ管理システム (CMS) アプリケーションが利用可能です:

WordPress

WordPress ブループリントは、PHP、MariaDB、phpMyAdmin、WordPress を備えた完全な本番環境を提供します。 は、業界のベストプラクティスを使用して、安全でup-to-dateブループリントをLightsailパッケージ化します。

Lightsail WordPress から起動された インスタンスには、デフォルトでインスタンスメタデータサービスバージョン 2 (IMDSv2) が適用されます。詳細については、「インスタンスメタデータサービスバージョン 2 の仕組み」を参照してください。

での WordPress のデプロイと管理 Lightsail

WordPress スタックの詳細については、WordPress ウェブサイトを参照してください。

WordPress certified by Bitnami

Bitnami WordPress は、Lightsail で WordPress を実行するための、すぐに利用できる事前設定済みイメージです。WordPress は、ブログやウェブサイトの構築によく使用されているウェブパブリッシングプラットフォームです。提供されているさまざまなテーマ、拡張機能、プラグイン、およびウィジェットを使用して WordPress をカスタマイズできます。

WordPress では完全なテーマシステム機能が提供されているため、何度かクリックするだけで、自分のサイトのルックアンドフィールを変更できます。無料または商用の既存の WordPress テーマを使用することもできます。WordPress は World Wide Web Consortium (W3C) 標準に完全に準拠しています。

での WordPress のデプロイと管理 Lightsail

「WordPress」の詳細について、Bitnami ウェブサイトを参照してください。

WordPress マルチサイトcertified by Bitnami

WordPress マルチサイトを使用すると、管理者は同じ WordPress インスタンスから複数のウェブサイトをホストして管理できます。これらのウェブサイトは、すべてが一意のドメイン名を持

ち、所有者がカスタマイズできます。また、サーバー管理者が利用可能にするテーマやプラグインなどのアセットを共有できます。すべてのサイトに対する更新を同時にプッシュできるため、常に安全で保護された状態に保つことができます。

WordPress マルチサイトは、多数の人々が各自のウェブサイトをホストできる一方で、単一のサーバー管理者が全体を統制する必要がある大学、企業、機関などの組織に最適です。

Lightsail で WordPress マルチサイトを設定する

「WordPress マルチサイト」の詳細について、Bitnami ウェブサイトを参照してください。

cPanel & WebHost マネージャ (WHM)

cPanel & WHM は、Linux OS 用に構築されたツールのスイートであり、シンプルなグラフィカルユーザーインターフェイスを介してウェブホスティングタスクを自動化する機能を提供します。お客様のサーバー管理、および顧客によるウェブサイト管理を簡素化することを目的としています。

Lightsail で cPanel と WHM を使用してウェブサイト、メール、サービスをホストします。

「cPanel と WHM」の詳細については、cPanel のウェブサイトを参照してください。

PrestaShop (Bitnami によってパッケージ化)

PrestaShop は、世界で最も多用されている e コマース ソリューションの 1 つです。これは、100万人以上のアクティブなメンバーのコミュニティを持つ、フリーでオープンソースソフトウェアです。これは、構成済みのテーマを使用して、オンラインストアを迅速に立ち上げて稼働できるように設計されています。サイトの外観を簡単にカスタマイズできるライブコンフィグレータにより、ほぼ瞬時に販売を開始できます。PrestaShopは、マルチストアのサポート、カスタマイズ可能なURL、複数の支払いゲートウェイオプション (PayPalとStripeを含む)、Amazon、eBay、Facebookなどとのマーケットプレイス統合を備えています。

Lightsail で PrestaShop ウェブサイトを設定する

「PrestaShop」の詳細について、PrestaShop のウェブサイトを参照してください。

Ghost (Bitnami によってパッケージ化)

Ghostは、個人的なブログから主要なニュースサイトまで、あらゆるものに適したパブリッシングプラットフォームです。Node.js 上に構築された最新のテクノロジースタックは、コンテンツ作成者の使いやすさを維持しながら、他のアプリケーションやツールとの統合を求める開発者に汎用性と柔軟性を提供します。

Lightsail で Ghost ウェブサイトをデプロイする

「Bitnami Ghost」の詳細について、Bitnami ウェブサイトを参照してください。

Joomla! (Bitnami によってパッケージ化)

Bitnami Joomla! は、Lightsail で Joomla! を実行するための、すぐに利用できる事前設定済みイメージです。Joomla! は、さまざまなウェブサイトおよびポータルの構築に使用できる CMS です。個人、企業、中小企業、非営利団体、およびその他の組織のウェブサイトで利用できます。

Joomla! では、登録システム機能によってユーザーが個人用オプションを設定することもできます。認証はユーザー管理の重要な部分であり、Joomla! は LDAP や OpenID などの複数のプロトコルをサポートしています。Joomla! は多言語をサポートし、ウェブサイトや管理パネルで多言語を利用するためのガイダンスを提供しています。また、Banner Manager により、サイトのバナーを簡単にセットアップして管理できます。インプレッション数の設定や特別な URL などのメトリクスを追跡できます。

Lightsail で Joomla! の使用を開始する

「Joomla!」の詳細について、Bitnami ウェブサイトを参照してください。

Drupal (Bitnami によってパッケージ化)

Bitnami Drupal は、Lightsail で Drupal を実行するための、すぐに利用できる事前設定済みイメージです。Drupal は、ユーザーがコンテンツを簡単に公開、管理、および整理できるようにする、コンテンツ管理プラットフォームです。Drupal は、コミュニティのウェブポータル、ディスカッションサイト、企業のウェブサイトなどに使用されています。Drupal は、モジュールを接続することによって容易に拡張できます。Drupal は、高パフォーマンス用に構築されていて、多数のサーバーにスケーリング可能であり、REST、JSON、SOAP、およびその他の形式と簡単に統合できます。

Drupal では何千ものアドオンモジュールとデザインを無償で使用できます。Drupal は複数の言語でも使用できます。

Lightsail で Drupal ウェブサイトを設定およびカスタマイズする

「Drupal」の詳細について、Bitnami ウェブサイトを参照してください。

アプリケーションスタックおよびサーバー

Lightsail には、さまざまな開発プロジェクト用の複数のアプリケーションスタックとサーバーがあります。各イメージは Linux/Unix を基本オペレーティングシステムとして使用します。

OpenClaw

OpenClaw はオープンソースの自律型 AI エージェント (以前の Clawdbot/Moltbot) です。独自のサーバーでバックグラウンドで継続的に実行され、プライマリインターフェイスとして Slack、Telegram、WhatsApp、Discord などのメッセージングプラットフォームに接続します。OpenClaw には、プロアクティブなタスク実行、マルチチャネル統合、コードの実行、ファイルの管理、ウェブの参照機能があります。

Lightsail OpenClaw から起動された インスタンスには、デフォルトでインスタンスメタデータサービスバージョン 2 (IMDSv2) が適用されます。詳細については、「インスタンスメタデータサービスバージョン 2 の仕組み」を参照してください。

で OpenClaw の使用を開始する Lightsail

OpenClaw の詳細については、OpenClaw OpenClawウェブサイトを参照してください。

[Bitnami によってパッケージ化された LAMP スタック (PHP 8)]

Bitnami LAMP スタックでは、PHP アプリケーションの開発とデプロイが簡略化されます。このスタックには、すぐに使用できるバージョンの Apache、MySQL、PHP、phpMyAdmin に加え、これらのコンポーネントの実行に必要なソフトウェアまでが含まれています。Bitnami LAMP スタックは完全に統合および設定されているため、Lightsail でインスタンスを作成したらすぐにアプリケーションの開発を開始できます。Bitnami LAMP スタックは定期的に更新されているため、バンドルされている各コンポーネントの最新の安定版リリースを常に利用できます。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

での LAMP スタックのデプロイと管理 Lightsail

「Bitnami LAMP スタック」の詳細について、Bitnami ウェブサイトを参照してください。

LAMP

LAMP ブループリントは、Linux 上の PHP、Apache、および MariaDB を備えた完全な本番環境を提供します。このブループリントには、phpMyAdmin、PHP コアモジュール、Composer も含まれています。

Lightsail LAMP から起動された インスタンスでは、インスタンスメタデータサービスバージョン 2 (IMDSv2) がデフォルトで適用されます。詳細については、「インスタンスメタデータサービスバージョン 2 の仕組み」を参照してください。

での LAMP スタックのデプロイと管理 Lightsail

Django (Bitnami によってパッケージ化)

Django は、迅速な開発とクリーンで実用的な設計を奨励する高レベルの Python ウェブフレームワークです。Python は、ソフトウェア開発の多くの種類のために使用することができる動的なオブジェクト指向プログラミング言語です。Bitnami Django スタックは、Django とそのランタイム依存関係のデプロイを大幅に簡素化します。また、すぐ実行可能なバージョンの Python、Django、MySQL、Apache も包含されています。

「Bitnami Django スタック」の詳細について、Bitnami ウェブサイトを参照してください。

Node.js (Bitnami によってパッケージ化)

Bitnami Node.js は、Lightsail で Node.js を実行するための、すぐに利用できる事前設定済みイメージです。Node.js は、高速でスケーラブルなネットワークアプリケーションを簡単に作成するための、Chrome の JavaScript ランタイム上に構築されているプラットフォームです。イベント駆動型のノンブロッキング I/O モデルが使用されているため、軽量かつ効率的です。Node.js はデータ集約型のリアルタイムアプリケーションに適しています。

での Node.js スタックのデプロイと管理 Lightsail

「Node.js スタック」の詳細について、Bitnami ウェブサイトを参照してください。

Node.js

Node.js ブループリントは、MariaDB と Node.js を備えた完全な本番環境を提供します。 は、業界のベストプラクティスを使用して、安全でup-to-dateブループリントをLightsailパッケージ化します。

Lightsail Node.js から起動された インスタンスには、デフォルトでインスタンスメタデータサービスバージョン 2 (IMDSv2) が適用されます。詳細については、「インスタンスメタデータサービスバージョン 2 の仕組み」を参照してください。

での Node.js スタックのデプロイと管理 Lightsail

Node.js スタックの詳細については、Node.js ウェブサイトを参照してください。

[Bitnami によってパッケージ化された MEAN スタック]

Bitnami MEAN スタックでは、ワンクリックでデプロイできる、MongoDB および Node.js 用の完全な開発環境が提供されています。このスタックには、MongoDB、Express、Angular、Node.js、Git、PHP、および RockMongo の最新の安定版リリースが含まれています。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

「MEAN スタック」の詳細について、Bitnami ウェブサイトを参照してください。

[Bitnami によってパッケージ化された GitLab CE]

Bitnami GitLab Community Edition (CE) は、Lightsail で GitLab を実行するために事前設定済みですぐに利用できるイメージです。Gitlab は、高速かつ安全で、Ruby on Rails に基づいたセルフホスト型の Git 管理ソフトウェアです。GitLab CI (GitLab に付属) は、Git および GitLab と緊密に統合されているオープンソースの継続的インテグレーション (CI) サーバーです。

GitLab を使用することで、ユーザーのサーバー上のコードのセキュリティを確保でき、リポジトリ、ユーザー、およびアクセス許可を管理できます。これは自己完結型であるため、インストールした GitLab を別のサーバーに簡単にコピーまたは移動できます。

Lightsail で GitLab CE インスタンスを設定および構成する

「GitLab スタック」の詳細について、Bitnami ウェブサイトを参照してください。

Nginx (LEMP スタック) (Bitnami によってパッケージ化)

Bitnami NGINX スタックでは、ワンクリックで起動できる、PHP、MySQL、および NGINX の完全な開発環境が提供されま
す。phpMyAdmin、SQLite、ImageMagick、FastCGI、Memcache、GD、CURL、PEAR、PECL などのコンポーネントもバンドルされています。

NGINX は非同期サーバーであり、スケーラビリティが主な長所となっています。NGINX スタックは LEMP (Linux、NGINX、MySQL、および PHP) とも呼ばれます。

Lightsail で Nginx ウェブサーバーをデプロイして管理する

NGINX スタックの詳細については、Bitnami ウェブサイトを参照してください。

Nginx

Nginx ブループリントは、PHP、MariaDB、phpMyAdmin、NGINX を備えた完全な本番環境を提供します。 は、業界のベストプラクティスを使用して、安全でup-to-dateブループリントをLightsailパッケージ化します。

Lightsail Nginx から起動された インスタンスでは、デフォルトでインスタンスメタデータサービスバージョン 2 (IMDSv2) が適用されます。詳細については、「インスタンスメタデータサービスバージョン 2 の仕組み」を参照してください。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

Lightsail で Nginx ウェブサーバーをデプロイして管理する

NGINX スタックの詳細については、NGINX ウェブサイトを参照してください。

[Ubuntu で Plesk ホストスタック]、[Ubuntu で Plesk ホストスタック (BYOL)]

> **Important**
>
> 2024 年 8 月 1 日より、Plesk は有料ライセンスモデルに移行しました。Plesk を実行している Lightsail インスタンスには、次のライセンス動作が適用されます。
>
> - 2025 年 2 月 1 日以降、古い [Ubuntu で Plesk ホストスタック] ブループリントを使用するインスタンスには有料ライセンスが必要になります。
> - [Ubuntu で Plesk ホストスタック (BYOL)] ブループリントで起動されたインスタンスには、30 日間のトライアルライセンスが適用されます。30 日後に Plesk アプリケーションを引き続き使用するには、Plesk からライセンスを購入する必要があります。
>
> 詳細については、「Plesk ライセンスの購入」を参照してください。

Plesk が搭載されたホスティングスタックを使用して、Lightsail および AWS 上でウェブサイトやアプリケーションを構築、保護、実行できます。グラフィカルユーザーインターフェースにウェブベースのサーバー管理ツールやセキュリティツール、さらに WordPress 自動化ツールがすべて含まれています。ウェブの専門家の作業を簡易化し、顧客が必要とするスケーラビリティ、セキュリティ、パフォーマンスを提供します。

Plesk をセットアップおよび設定する。

「Plesk スタック」の詳細については、Plesk のウェブサイトを参照してください。

Ruby on Rails

Ruby on Rails ブループリントは、Amazon Linux 2023 の Rails で事前設定されているため、フレームワークを手動でインストールおよびセットアップする必要はありません。Ruby on Rails ブループリントを使用すると、 で最新のウェブアプリケーションを構築するための堅牢でスケーラブル、費用対効果の高いソリューションをデプロイできますLightsail。

で Rails に Ruby を設定する Lightsail

Ruby on Rails と Amazon Linux 2023 の詳細をご覧ください。

e コマース アプリケーション

現在 Lightsail には、1 つの e コマースアプリケーションのイメージ、 Magento が含まれています。この Magento イメージでは、Linux/Unix (Ubuntu) が基本オペレーティングシステムとして使用されます。

Magento (Bitnami によってパッケージ化)

Bitnami Magento は、Lightsail で Magento を実行するための、すぐに利用できる事前設定済みイメージです。Magento を使用して、魅力的で応答性が高くて安全なサイトを構築できます。Magento は、機能が豊富で柔軟性に優れた e コマースソリューションであり、トランザクションオプション、マルチストア機能、ロイヤルティプログラム、製品のカテゴリ化、買い物客のフィルタリング、プロモーションルールなどの機能を備えます。

Magento を使用することによって、自社のブランドを反映しつつ高度にカスタマイズした、e コマース用サイトを作成できます。Magento はユーザーのビジネスオペレーションと統合されるため、ご自分のビジネスニーズに合わせて e コマースサイトを管理できます。

Lightsail で Magento を設定および構成する

「Magento スタック」の詳細について、Bitnami ウェブサイトを参照してください。

プロジェクト管理アプリケーション

現在 Lightsail には、1 つのプロジェクト管理アプリケーションのイメージ、Redmine が含まれています。このイメージでは、Linux/Unix (Ubuntu) が基本オペレーティングシステムとして使用されます。

[Bitnami によってパッケージ化された Redmine]

Bitnami Redmine は、Lightsail で Redmine を実行するための、すぐに利用できる事前設定済みイメージです。Redmine は、柔軟性に優れたプロジェクト管理ウェブアプリケーションです。複数のプロジェクト、ロールベースのアクセスコントロール、ガントチャートとカレンダー、ニュース/ドキュメント/ファイルの管理、プロジェクトごとの Wiki とフォーラム、SCM 統合などがサポートされています。

このブループリントは、Lightsail IPV6 専用インスタンスプランに対応しています。

Lightsail の Redmine インスタンスを設定して保護する

「Redmine スタック」の詳細について、Bitnami ウェブサイトを参照してください。

Lightsail インスタンスバンドル

Lightsail には、さまざまなワークロード要件を満たすためのさまざまなインスタンスバンドル (インスタンスプランとも呼ばれる) が用意されています。各バンドルには、コンピューティング能力 (vCPU)、メモリ (RAM)、ストレージ、データ転送許容量があります。バンドルはオンデマンドの時間料金で請求されるため、ご利用分のみのお支払いとなります。ご利用のバンドルごとに固定の時間料金が課金され、プランの月額上限まで適用されます。

汎用プランは、ウェブサーバーやアプリケーションサーバー、仮想デスクトップ、マイクロサービス、データベース、バッチ処理など、ほとんどのアプリケーションに適したコンピューティング、メモリ、ネットワークリソースをバランスよく組み合わせます。メモリ最適化プランはメモリmemory-to-CPUの比率が高く、インメモリキャッシュ、リアルタイムのビッグデータ分析、高性能データベースなどのワークロードにメリットをもたらします。

各バンドルにはパブリック IP アドレス機能が含まれます。パブリック IPv4 アドレス、IPv6 アドレスとパブリック IPv4 アドレスの両方 (デュアルスタックと呼ばれる) 、IPv6 アドレス (単に IPv6 と呼ばれる) のみのいずれかが提供されます。バンドル料金は、パブリック IPv4 アドレスが含まれているかどうかによって異なります。

トピック

- Linux/Unix バンドル (パブリック IPv4 アドレス機能付)
- Linux/Unix バンドル (IPv6 のみ)
- Windows バンドル (パブリック IPv4 アドレス機能付)
- Windows バンドル (IPv6 のみ)

Linux/Unix バンドル (パブリック IPv4 アドレス機能付)

次のテーブルには、IPv4 アドレスを含む Linux/Unix インスタンスバンドルの仕様が一覧表示されています。

バンドル名	料金 (USD/月)	vCPU	メモリ	Storage	データ転送 *
パブリック IPv4 を使用する Nano-0.5GB Linux	5.00 USD	2	0.5 GB	20 GB	1 TB

バンドル名	料金 (USD/月)	vCPU	メモリ	Storage	データ転送 *
パブリック IPv4 を搭載した Micro-1GB Linux	7.00 USD	2	1 GB	40 GB	2 TB
パブリック IPv4 を搭載したSmall-2GB Linux	12.00 USD	2	2 GB	60 GB	3 TB
パブリック IPv4 を搭載した Medium-4GB Linux	24.00 USD	2	4 GB	80 GB	4 TB
パブリック IPv4 を搭載したLarge-8GB Linux	44.00 USD	2	8 GB	160 GB	5 TB
パブリック IPv4 を使用する Xlarge-16GB Linux	84.00 USD	4	16 GB	320 GB	6 TB
パブリック IPv4 を搭載した 2Xlarge-32GB Linux	164.00 USD	8	32 GB	640 GB	7 TB
パブリック IPv4 を搭載した 4Xlarge-64GB Linux	384.00 USD	16	64 GB	1,280 GB	8 TB
パブリック IPv4 を搭載した 8Xlarge-128GB Linux	884.00 USD	32	128 GB	1,280 GB	9 TB
パブリック IPv4 を搭載した 12Xlarge-192GB Linux	1,324.00 USD	48	192 GB	1,280 GB	10 TB
パブリック IPv4 を搭載した 16Xlarge-256GB Linux	1,764.00 USD	64	256 GB	1,280 GB	10 TB

バンドル名	料金 (USD/月)	vCPU	メモリ	Storage	データ転送 *
パブリック IPv4 を搭載したメモリ最適化 Large-16GB Linux	74.00 USD	2	16 GB	160 GB	5 TB
パブリック IPv4 を搭載したメモリ最適化 Xlarge-32GB Linux	144.00 USD	4	32 GB	320 GB	6 TB
パブリック IPv4 を搭載したメモリ最適化 2Xlarge-64GB Linux	294.00 USD	8	64 GB	640 GB	7 TB
パブリック IPv4 を搭載したメモリ最適化 4Xlarge-128GB Linux	584.00 USD	16	128 GB	1,280 GB	8 TB
パブリック IPv4 を搭載したメモリ最適化 8Xlarge-256GB Linux	1,174 USD	32	256 GB	1,280 GB	9 TB
パブリック IPv4 を搭載したメモリ最適化 12Xlarge-384GB Linux	1,764.00 USD	48	384 GB	1,280 GB	10 TB
パブリック IPv4 を搭載したメモリ最適化 16Xlarge-512GB Linux	2,344.00 USD	64	512 GB	1,280 GB	10 TB

* データ転送許容量はリージョンによって異なります。詳細については、「インスタンスのデータ転送許容量は、 AWS リージョンによってどれぐらい異なりますか？」を参照してください。

Linux/Unix バンドル (IPv6 のみ)

次のテーブルには、IPv6 アドレスのみを持つ Linux/Unix インスタンスバンドルの仕様が一覧表示されています。

バンドル名	料金 (USD/月)	vCPU	メモリ	Storage	データ転送 *
Nano-0.5GB Linux IPv6-only	3.50 USD	2	0.5 GB	20 GB	1 TB
Micro-1GB Linux IPv6-only	5.00 USD	2	1 GB	40 GB	2 TB
Small-2GB Linux IPv6-only	10.00 USD	2	2 GB	60 GB	3 TB
Medium-4GB Linux IPv6-only	20.00 USD	2	4 GB	80 GB	4 TB
Large-8GB Linux IPv6-only	40.00 USD	2	8 GB	160 GB	5 TB
Xlarge-16GB Linux IPv6-only	80.00 USD	4	16 GB	320 GB	6 TB
2Xlarge-32GB Linux IPv6-only	160.00 USD	8	32 GB	640 GB	7 TB
4Xlarge-64GB Linux IPv6-only	380.00 USD	16	64 GB	1,280 GB	8 TB
8Xlarge-128GB Linux IPv6-only	880.00 USD	32	128 GB	1,280 GB	9 TB
12Xlarge-192GB Linux IPv6-only	1,320.00 USD	48	192 GB	1,280 GB	10 TB
16Xlarge-256GB Linux IPv6-only	1,760.00 USD	64	256 GB	1,280 GB	10 TB
メモリ最適化 Large-16GB Linux IPv6-only	70.00 USD	2	16 GB	160 GB	5 TB

バンドル名	料金 (USD/月)	vCPU	メモリ	Storage	データ転送 *
メモリ最適化 Xlarge-32 GB Linux IPv6-only	140.00 USD	4	32 GB	320 GB	6 TB
メモリ最適化 2Xlarge-64GB Linux IPv6-only	290.00 USD	8	64 GB	640 GB	7 TB
メモリ最適化 4Xlarge-128GB Linux IPv6-only	580.00 USD	16	128 GB	1,280 GB	8 TB
メモリ最適化 8Xlarge-256GB Linux IPv6-only	1170.00 USD	32	256 GB	1,280 GB	9 TB
メモリ最適化 12Xlarge-384GB Linux IPv6-only	1,760.00 USD	48	384 GB	1,280 GB	10 TB
メモリ最適化 16Xlarge-512GB Linux IPv6-only	2,340.00 USD	64	512 GB	1,280 GB	10 TB

* データ転送許容量はリージョンによって異なります。詳細については、「インスタンスのデータ転送許容量は、 AWS リージョンによってどれぐらい異なりますか？」を参照してください。

Windows バンドル (パブリック IPv4 アドレス機能付)

次のテーブルには、IPv4 アドレスを含む Windows インスタンスバンドルの仕様が一覧表示されています。

バンドル名	料金 (USD/月)	vCPU	メモリ	Storage	データ転送 *
パブリック IPv4 を搭載した Nano-0.5GB Windows	9.50 USD	2	0.5 GB	30 GB	1 TB
パブリック IPv4 を備えた Micro-1GB Windows	14.00 USD	2	1 GB	40 GB	2 TB

バンドル名	料金 (USD/月)	vCPU	メモリ	Storage	データ転送 *
パブリック IPv4 を備えたSmall-2GB Windows	22.00 USD	2	2 GB	60 GB	3 TB
パブリック IPv4Medium-4GB Windows	44.00 USD	2	4 GB	80 GB	4 TB
パブリック IPv4 を搭載したLarge-8GB Windows	74.00 USD	2	8 GB	160 GB	5 TB
パブリック IPv4 を搭載した Xlarge-16GB Windows	124.00 USD	4	16 GB	320 GB	6 TB
パブリック IPv4 を搭載した 2Xlarge-32GB Windows	244.00 USD	8	32 GB	640 GB	7 TB
パブリック IPv4 を搭載した 4Xlarge-64GB Windows	574.00 USD	16	64 GB	1,280 GB	8 TB
パブリック IPv4 を搭載した 8Xlarge-128GB Windows	1,254.00 USD	32	128 GB	1,280 GB	9 TB
パブリック IPv4 を搭載した 12Xlarge-192GB Windows	1,884.00 USD	48	192 GB	1,280 GB	10 TB
パブリック IPv4 を搭載した 16Xlarge-256GB Windows	2,504.00 USD	64	256 GB	1,280 GB	10 TB

バンドル名	料金 (USD/月)	vCPU	メモリ	Storage	データ転送 *
パブリック IPv4 を搭載したメモリ最適化 Large-16GB Windows	134.00 USD	2	16 GB	160 GB	5 TB
パブリック IPv4 を搭載したメモリ最適化 Xlarge-32GB Windows	264.00 USD	4	32 GB	320 GB	6 TB
パブリック IPv4 を搭載したメモリ最適化 2Xlarge-64GB Windows	524.00 USD	8	64 GB	640 GB	7 TB
パブリック IPv4 を搭載したメモリ最適化 4Xlarge-128GB Windows	1044.00 USD	16	128 GB	1,280 GB	8 TB
パブリック IPv4 を搭載したメモリ最適化 8Xlarge-256GB Windows	2,104.00 USD	32	256 GB	1,280 GB	9 TB
パブリック IPv4 を搭載したメモリ最適化 12Xlarge-384GB Windows	3,164.00 USD	48	384 GB	1,280 GB	10 TB
パブリック IPv4 を搭載したメモリ最適化 16Xlarge-512GB Windows	4,204.00 USD	64	512 GB	1,280 GB	10 TB

* データ転送許容量はリージョンによって異なります。詳細については、「インスタンスのデータ転送許容量は、 AWS リージョンによってどれぐらい異なりますか？」を参照してください。

Windows バンドル (IPv6 のみ)

次のテーブルには、IPv6 アドレスのみを持つ Windows インスタンスバンドルの仕様が一覧表示されています。

バンドル名	料金 (USD/月)	vCPU	メモリ	Storage	データ転送 *
Nano-0.5 GB Windows IPv6-only	8.00 USD	2	0.5 GB	30 GB	1 TB
Micro-1GB Windows IPv6-only	12.00 USD	2	1 GB	40 GB	2 TB
Small-2GB Windows IPv6-only	20.00 USD	2	2 GB	60 GB	3 TB
Medium-4GB Windows IPv6-only	40.00 USD	2	4 GB	80 GB	4 TB
Large-8GB Windows IPv6-only	70.00 USD	2	8 GB	160 GB	5 TB
Xlarge-16GB Windows IPv6-only	120.00 USD	4	16 GB	320 GB	6 TB
2Xlarge-32GB Windows IPv6-only	240.00 USD	8	32 GB	640 GB	7 TB
4Xlarge-64GB Windows IPv6-only	570.00 USD	16	64 GB	1,280 GB	8 TB
8Xlarge-128GB Windows IPv6-only	1,250.00 USD	32	128 GB	1,280 GB	9 TB
12Xlarge-192GB Windows IPv6-only	1,880.00 USD	48	192 GB	1,280 GB	10 TB

バンドル名	料金 (USD/月)	vCPU	メモリ	Storage	データ転送 *
16Xlarge-256GB Windows IPv6-only	2,500.00 USD	64	256 GB	1,280 GB	10 TB
メモリ最適化 Large-16GB Windows IPv6-only	130.00 USD	2	16 GB	160 GB	5 TB
メモリ最適化 Xlarge-32 GB Windows IPv6-only	260.00 USD	4	32 GB	320 GB	6 TB
メモリ最適化 2Xlarge-64GB Windows IPv6-only	520.00 USD	8	64 GB	640 GB	7 TB
メモリ最適化 4Xlarge-128GB Windows IPv6-only	1,040.00 USD	16	128 GB	1,280 GB	8 TB
メモリ最適化 8Xlarge-256GB Windows IPv6-only	2,100.00 USD	32	256 GB	1,280 GB	9 TB
メモリ最適化 12Xlarge-384GB Windows IPv6-only	3,160.00 USD	48	384 GB	1,280 GB	10 TB
メモリ最適化 16Xlarge-512GB Windows IPv6-only	4,200.00 USD	64	512 GB	1,280 GB	10 TB

* データ転送許容量はリージョンによって異なります。詳細については、「インスタンスのデータ転送許容量は、 AWS リージョンによってどれぐらい異なりますか？」を参照してください。

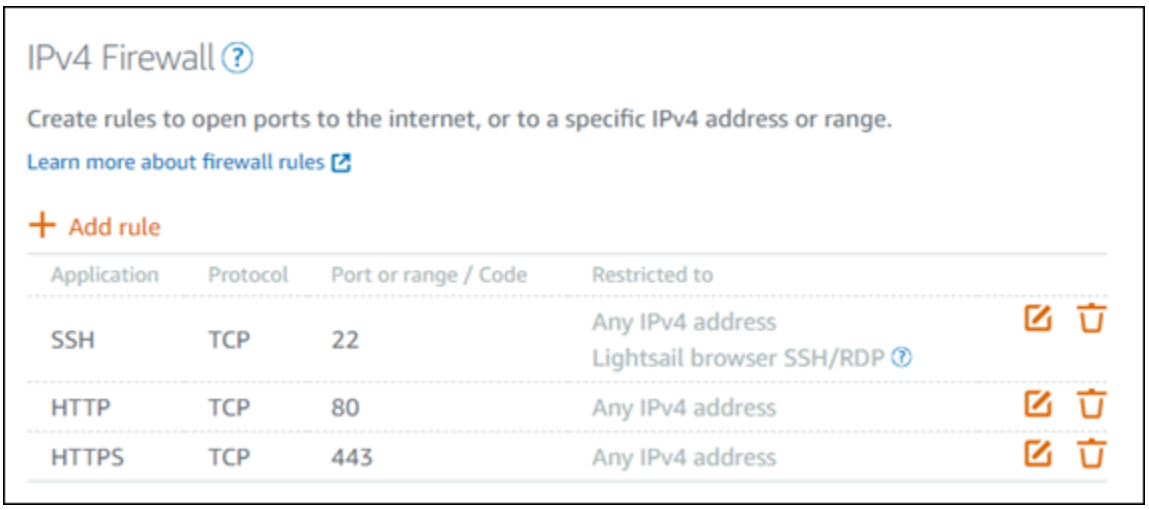
Lightsail のファイアウォールを使用してインスタンストラフィックを制御する

Amazon Lightsail コンソールのファイアウォールは、パブリック IP アドレス経由でインスタンスに接続することが許可されたトラフィックを制御する仮想ファイアウォールとして機能します。Lightsail で作成する各インスタンスには 2 つのファイアウォールがあります。1 つは IPv4 アドレス用、もう 1 つは IPv6 アドレス用です。各ファイアウォールには、インスタンスに着信するトラフィックをフィルタリングする一連のルールが含まれています。各ファイアウォールは、互いに独立しています。IPv4 と IPv6 のファイアウォールルールを別個に設定する必要があります。インスタンスのファイアウォールは、トラフィックを許可または制限するルールを追加および削除することで、いつでも編集できます。

Lightsail のファイアウォール

各 Lightsail インスタンスにはファイアウォールが 2 つあります。1 つは IPv4 アドレス用、もう 1 つは IPv6 アドレス用です。Lightsail インスタンスに出入りするすべてのインターネットトラフィックは、そのファイアウォールを通過します。インスタンスのファイアウォールは、インスタンスへの流入を許可されたインターネットトラフィックを制御します。ただし、送信トラフィックは制御しません。 ファイアウォールは、すべてのアウトバウンドトラフィックを許可します。インスタンスのファイアウォールは、受信ラフィックを許可または制限するルールを追加および削除することで、いつでも編集できます。各ファイアウォールは、互いに独立しています。IPv4 と IPv6 のファイアウォールルールを別個に設定する必要があります。

ファイアウォールルールは常にアクセスを許可します。アクセスを拒否するルールを作成することはできません。インスタンスへの着信トラフィックを許可するルールをインスタンスのファイアウォールに追加します。インスタンスのファイアウォールにルールを追加するときは、以下の例（IPv4）に示すように、使用するプロトコル、開くポート、インスタンスに接続できる IPv4 および IPv6 アドレスを指定します。アプリケーションレイヤーのプロトコルタイプも指定できます。これは、インスタンスで使用するサービスに応じて、プロトコルとポート範囲を自動的に指定するプリセットです。



> **Important**
>
> ファイアウォールルールが影響するのは、インスタンスのパブリック IP アドレス経由で受信されるトラフィックのみです。このルールは、同じ AWS リージョン内のアカウントにある Lightsail リソース、または同じ AWS リージョン内のピアリングされた仮想プライベートクラウド (VPC) にあるリソースを送信元とする、インスタンスのプライベート IP アドレス経由で受信されるトラフィックには影響しません。

ファイアウォールルールおよび設定可能なパラメータについては、このガイドの以降のセクションで説明します。

ファイアウォールルールを作成する

ファイアウォールルールを作成して、クライアントがインスタンスまたはインスタンスで実行されているアプリケーションとの接続を確立できるようにします。例えば、すべてのウェブブラウザがインスタンス上の WordPress アプリケーションに接続できるようにするには、任意の IP アドレスからのポート 80 経由での Transmission Control Protocol (TCP) を有効にするファイアウォールルールを設定します。このルールがインスタンスのファイアウォールに設定済みである場合は、このルールを削除することで、ウェブブラウザからインスタンスの WordPress アプリケーションへの接続をブロックできます。

> **Important**
>
> Lightsail コンソールを使用して、一度に 30 個までソース IP アドレスを追加できます。一度に最大 60 個の IP アドレスを追加するには、 Lightsail API、 AWS Command Line Interface

(AWS CLI)、または SDK AWS を使用します。このクォータは、IPv4 ルールと IPv6 ルールに対して個別に適用されます。例えば、ファイアウォールで、IPv4 トラフィックと IPv6 トラフィックのそれぞれに 60 個のインバウンドルールを設定できます。個々の IP アドレスを CIDR 範囲に統合することをお勧めします。詳細については、このガイドの「送信元 IP アドレスの指定」セクションを参照してください。

また、SSH クライアントからインスタンスへの接続を許可してサーバーで管理タスクを実行できます。そのためには、接続を確立する必要があるコンピュータの IP アドレスに限り、ポート 22 を介した TCP を許可するファイアウォールルールを設定します。この場合、インスタンスへの SSH 接続の確立をすべての IP アドレスに許可しないようにします。許可すると、インスタンスのセキュリティリスクが生じる可能性があります。

> **Note**
>
> このセクションで説明するファイアウォールルールの例は、インスタンスのファイアウォールにデフォルトで設定済みである場合があります。詳細については、このガイドの後半の「デフォルトのファイアウォールルール」を参照してください。

特定のポートに複数のルールがある場合、最も許容度の大きいルールを適用します。たとえば、IP アドレス 192.0.2.1 からの TCP ポート 22 (SSH) へのアクセスを許可するルールを追加したとします。次に、すべてのユーザーからの TCP ポート 22 へのアクセスを許可する別のルールを追加します。これにより、すべてのユーザーが TCP ポート 22 にアクセスできるようになります。

プロトコルを指定する

プロトコルは、2 台のコンピュータ間でデータを転送する形式です。Lightsail では、ファイアウォールルールで次のプロトコルを指定できます。

- Transmission Control Protocol (TCP) は主に、クライアントとインスタンスで実行されているアプリケーション間の接続を確立して、データの交換が完了するまで接続を維持するために使用されます。これは広く使用されており、ファイアウォールルールで指定することが多いプロトコルです。TCP は、送信されたデータに欠落がないこと、および送信されたすべてのデータが目的の受信者に到達することを保証します。ウェブブラウジング、金融取引、テキストメッセージングなど、高い信頼性が要求されるが、転送時間の重要度が低いネットワークアプリケーションに最適です。これらのユースケースでは、データの一部が失われると、重大な価値の損失となります。

- UDP (User Datagram Protocol) は、インスタンスで実行されているアプリケーションとクライアントとの間で低レイテンシーおよび損失許容接続を確立するために主に使用します。ゲーム、音声、ビデオ通信など、体感レイテンシーの重要度が高いネットワークアプリケーションに最適です。これらのユースケースでは、多少のデータ損失が生じる場合がありますが、体感品質を損なうほどではありません。
- ICMP (Internet Control Message Protocol) は、ネットワーク通信の問題を診断するために主に使用します。たとえば、データが送信先にタイムリーに到着しているかどうかを確認します。このプロトコルは Ping ユーティリティに最適です。このユーティリティでは、ローカルコンピュータとインスタンス間の接続速度をテストできます。データがインスタンスに到着してローカルコンピュータに戻ってくるまでの所要時間をレポートします。

  > **Note**
  >
  > Lightsail コンソールを使用してインスタンスの IPv6 ファイアウォールに ICMP ルールを追加すると、このルールは ICMPv6 を使用するように自動的に設定されます。詳細については、Wikipedia の「IPv6 のインターネット制御メッセージプロトコル」を参照してください。

- すべてでは、インスタンスへのすべてのプロトコルトラフィックの流入を許可します。どのプロトコルを指定すればよいかわからない場合は、このプロトコルを指定します。これには、上に示したプロトコルだけでなく、すべてのインターネットプロトコルが含まれます。詳細については、「Protocol Numbers」(Internet Assigned Numbers Authority ウェブサイト) を参照してください。

ポートの指定

コンピュータがキーボードやマウスなどの周辺機器と通信するためのコンピュータの物理ポートと同様に、ネットワークポートはインスタンスのインターネット通信エンドポイントとして機能します。コンピュータは、インスタンスと接続するときに、通信を確立するためのポートを公開します。

ファイアウォールルールで指定できるポートの範囲は 0～65535 です。インスタンスへの接続をクライアントに許可するファイアウォールを作成する場合、使用するプロトコル (このガイドの前半で説明) と、接続の確立に使用できるポート番号を指定します。プロトコルとポートを使用して接続を確立できる IP アドレスを指定することもできます。これについては、このガイドの次のセクションで説明します。

よく使用されるポートと、これらのポートを使用するサービスは以下のとおりです。

- FTP (File Transfer プロトコル) を介したデータ転送では、ポート 20 を使用します。

- FTP に介したコマンド制御では、ポート 21 を使用します。
- Secure Shell (SSH) では、ポート 22 を使用します。
- Telnet リモートログインサービス、および暗号化されていないテキストメッセージでは、ポート 23 を使用します。
- SMTP (Simple Mail Transfer Protocol) では、E メールの送信にポート 25 を使用します。

> **Important**
>
> インスタンスで SMTP を有効にするには、インスタンスにリバース DNS も設定する必要があります。そうしないと、E メールが TCP ポート 25 経由に制限される可能性があります。詳細については、「Amazon Lightsail インスタンスで E メールサーバーの逆引き DNS を設定する」を参照してください。

- ドメインネームシステム (DNS) サービスでは、ポート 53 を使用します。
- ウェブブラウザがウェブサイトに接続するために使用する HTTP (ハイパーテキスト転送プロトコル) では、ポート 80 を使用します。
- E メールクライアントがサーバーから E メールを取得するために使用する POP3 (Post Office Protocol) では、ポート 110 を使用します。
- NNTP (Network News Transfer Protocol) では、ポート 119 を使用します。
- NTP (Network Time Protocol) では、ポート 123 を使用します。
- デジタルメールを管理するために使用する IMAP (インターネットメッセージアクセスコントロール) では、ポート 143 を使用します。
- SNMP (簡易ネットワーク管理プロトコル) では、ポート 161 を使用します。
- ウェブブラウザがウェブサイトへの暗号化された接続を確立するために使用する TLS/SSL を介した HTTP Secure (HTTPS) HTTP では、ポート 443 を使用します。

詳細については、「Service Name and Transport Protocol Port Number Registry」(Internet Assigned Numbers Authority ウェブサイト) を参照してください。

アプリケーションレイヤーのプロトコルタイプを指定する

ファイアウォールルールの作成時に、アプリケーションレイヤーのプロトコルタイプを指定できます。プロトコルタイプは、インスタンスで有効にしたサービスに応じてルールのプロトコルとポート範囲を指定するプリセットです。これにより、SSH、RDP、HTTP などのサービスで使用するー

般的なプロトコルやポートを検索する必要がなくなります。これらのアプリケーションレイヤーのプロトコルタイプを選択するだけで、プロトコルとポートが自動的に指定されます。独自のプロトコルとポートを指定する場合は、アプリケーションレイヤーのプロトコルタイプとして [カスタムルール] を選択できます。これにより、該当するパラメータを制御できます。

> **Note**
>
> アプリケーションレイヤーのプロトコルタイプは、Lightsail コンソールでのみ指定できます。Lightsail API、 AWS Command Line Interface (AWS CLI)、または SDK を使用してアプリケーションレイヤーのプロトコルタイプを指定することはできません。

Lightsail コンソールでは、次のアプリケーションレイヤーのプロトコルタイプを使用できます。

- カスタム - 独自のプロトコルとポートを指定する場合は、このオプションを選択します。
- すべてのプロトコル – すべてのプロトコルを指定して、独自のポートを指定する場合は、このオプションを選択します。
- すべての TCP – TCP プロトコルを使用するが、どのポートを開けばよいかわからない場合は、このオプションを選択します。これにより、すべてのポート (0 ~ 65535) を介した TCP が有効になります。
- すべての UDP – UDP プロトコルを使用するが、どのポートを開けばよいかわからない場合は、このオプションを選択します。これにより、すべてのポート (0 ~ 65535) で UDP が有効になります。
- すべての ICMP — すべての ICMP タイプとコードを指定するには、このオプションを選択します。
- カスタム ICMP - ICMP プロトコルを使用し、ICMP のタイプとコードを定義する場合は、このオプションを選択します。ICMP タイプとコードの詳細については、Wikipedia の「制御メッセージ」を参照してください。
- DNS – インスタンスで DNS を有効にする場合は、このオプションを選択します。これにより、ポート 53 を介した TCP および UDP が有効になります。
- HTTP – インスタンスでホストされているウェブサイトへの接続をウェブブラウザに許可する場合は、このオプションを選択します。これにより、ポート 80 を介した TCP が有効になります。
- HTTPS – インスタンスでホストされているウェブサイトへの暗号化された接続の確立をウェブブラウザに許可する場合は、このオプションを選択します。これにより、ポート 443 を介した TCP が有効になります。

- MySQL/Aurora – インスタンスでホストされている MySQL または Aurora データベースへの接続をクライアントに許可する場合は、このオプションを選択します。これにより、ポート 3306 を介した TCP が有効になります。
- Oracle-RDS – インスタンスでホストされている Oracle または RDS データベースへの接続をクライアントに許可する場合は、このオプションを選択します。これにより、ポート 1521 を介した TCP が有効になります。
- Ping (ICMP) – Ping ユーティリティを使用してリクエストに応答することをインスタンスに許可する場合は、このオプションを選択します。IPv4 ファイアウォールでは、ICMP タイプ 8 (エコー) とコード -1 (すべてのコード) が有効になります。IPv6 ファイアウォールでは、ICMP タイプ 129 (エコー応答) とコード 0 が有効になります。
- RDP – インスタンスへの接続を RDP クライアントに許可する場合に、このオプションを選択します。これにより、ポート 3389 を介した TCP が有効になります。
- SSH – インスタンスへの接続を SSH クライアントに許可する場合に、このオプションを選択します。これにより、ポート 22 を介した TCP が有効になります。

送信元 IP アドレスを指定する

ファイアウォールルールは、デフォルトですべての IP アドレスに対して、指定したプロトコルとポートを介してインスタンスに接続することを許可します。これは、HTTP や HTTPS を経由するウェブブラウザなどのトラフィックに最適です。ただし、SSH や RDP などのトラフィックの場合、これらのアプリケーションを使用してインスタンスに接続することをすべての IP アドレスに許可することは、セキュリティ上のリスクとなります。そのため、ファイアウォールルールを IPv4 または IPv6 アドレス、あるいは IP アドレス範囲に制限できます。

- IPv4 アドレスについて - 単一の IPv4 アドレス(203.0.113.1 など) または IPv4 アドレス範囲を指定できます。Lightsail コンソールで範囲を指定するには、ダッシュを使用するか (192.0.2.0-192.0.2.255 など)、CIDR ブロック表記を使用します (192.0.2.0/24 など)。CIDR ブロックの表記の詳細については、Wikipedia の「Classless Inter-Domain Routing」記事を参照してください。
- IPv6ファイアウォールについて - 単一の IPv6 アドレス (2001:0db8:85a3:0000:0000:8a2e:0370:7334 など) または IPv6 アドレス範囲を指定できます。Lightsail コンソールでは、IPv6 の範囲は CIDR ブロック表記 (2001:db8:: /32 など) で指定できます。IPv6 CIDR ブロック表記の詳細については、Wikipedia の「IPv6 CIDR ブロック」を参照してください。

Lightsail のデフォルトのファイアウォールルール

新しいインスタンスを作成すると、そのインスタンスへの基本的なアクセスを許可する以下のデフォルトのルールが、IPv4 および IPv6 のファイアウォールに事前設定されます。デフォルトのルールは、作成するインスタンスのタイプに応じて異なります。これらのルールは、アプリケーション、プロトコル、ポート、および送信元 IP アドレスのリスト (アプリケーション - プロトコル - ポート - 送信元 IP アドレスなど) として示してあります。

AlmaLinux、Amazon Linux 2、Amazon Linux 2023、CentOS、Debian、FreeBSD、openSUSE、および Ubuntu (ベースオペレーティングシステム)

SSH - TCP - 22 - すべての IP アドレス

HTTP - TCP - 80 - すべての IP アドレス

WordPress、Ghost、Joomla!、PrestaShop、Drupal (CMS アプリケーション)

SSH - TCP - 22 - すべての IP アドレス

HTTP - TCP - 80 - すべての IP アドレス

HTTPS - TCP - 443 - すべての IP アドレス

cPanel & WHM (CMS アプリケーション)

SSH - TCP - 22 - すべての IP アドレス

DNS (UDP) - UDP - 53 - すべての IP アドレス

DNS (TCP) - TCP - 53 - すべての IP アドレス

HTTP - TCP - 80 - すべての IP アドレス

HTTPS - TCP - 443 - すべての IP アドレス

カスタム - TCP - 2078 - すべての IP アドレス

カスタム - TCP - 2083 - すべての IP アドレス

カスタム - TCP - 2087 - すべての IP アドレス

カスタム - TCP - 2089 - すべての IP アドレス

LAMP、Django、Node.js、MEAN、GitLab と Nginx (開発スタック)

SSH - TCP - 22 - すべての IP アドレス

HTTP - TCP - 80 - すべての IP アドレス

HTTPS - TCP - 443 - すべての IP アドレス

Magento (e コマースアプリケーション)

SSH - TCP - 22 - すべての IP アドレス

HTTP - TCP - 80 - すべての IP アドレス

HTTPS - TCP - 443 - すべての IP アドレス

Redmine (プロジェクト管理アプリケーション)

SSH - TCP - 22 - すべての IP アドレス

HTTP - TCP - 80 - すべての IP アドレス

HTTPS - TCP - 443 - すべての IP アドレス

Plesk (ホスティングスタック)

SSH - TCP - 22 - すべての IP アドレス

HTTP - TCP - 80 - すべての IP アドレス

HTTPS - TCP - 443 - すべての IP アドレス

カスタム - TCP - 53 - すべての IP アドレス

カスタム - UDP - 53 - すべての IP アドレス

カスタム - TCP -8443 - すべての IP アドレス

カスタム - TCP - 8447 - すべての IP アドレス

Windows Server 2022、Windows Server 2019、および Windows Server 2016

SSH - TCP - 22 - すべての IP アドレス

HTTP - TCP - 80 - すべての IP アドレス

RDP - TCP - 3389 - すべての IP アドレス

SQL Server Express 2022、SQL Server Express 2019、および SQL Server Express 2016

SSH - TCP - 22 - すべての IP アドレス

HTTP - TCP - 80 - すべての IP アドレス

RDP - TCP - 3389 - すべての IP アドレス

Lightsail インスタンスにファイアウォールルールを追加する

Amazon Lightsail インスタンスのファイアウォールにルールを追加して、インスタンスに接続できるトラフィックを制御できます。ファイアウォールルールを追加する場合、インスタンスの接続が許可されているアプリケーションレイヤーのプロトコルタイプ、プロトコル、ポート、および送信元 IPv4 or IPv6 アドレスを指定できます、ファイアウォールの詳細については、「ファイアウォールとポート」を参照してください。

インスタンスのファイアウォールルールを追加および編集する

Lightsail コンソールでファイアウォールルールを追加または編集するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [インスタンス] を選択してください。
3. ファイアウォールルールを追加または編集する対象のインスタンスの名前を選択します。
4. インスタンスの管理ページで [ネットワーキング] タブを選択します。

   [ネットワーキング] タブには、インスタンスのパブリック IP アドレスとプライベート IP アドレス、インスタンスに設定された IPv4 や IPv6 ファイアウォールルールが表示されます。

   > **Note**
   >
   > IPv6 ファイアウォールは、インスタンスで IPv6 を有効にしている場合にのみ表示されます。詳細については、「IPv6 を有効化または無効化する」を参照してください。

5. ルールの送信元 IP が IPv4 アドレスか IPv6 アドレスかに応じて、次のいずれかの手順を実行します。

   - ページの [ファイアウォール] セクションまでスクロールし、[ルールの追加] を選択します。

- ページの [ファイアウォール] セクションまでスクロールし、[ルールの追加] を選択します。

  既存のルールの横にある [編集] (鉛筆アイコン) を選択して編集することもできます。

6. [アプリケーション] ドロップダウンメニューからアプリケーションレイヤーのプロトコルタイプを選択します。

   アプリケーションレイヤーのプロトコルタイプを選択すると、プロトコルとポートのプリセットが自動的に指定されます。値の例は [Custom (カスタム)]、[All TCP (すべての TCP)]、[All UDP (すべての UDP)]、[Custom ICMP (カスタム ICMP)]、[SSH]、[RDP] です。

   選択したアプリケーションレイヤープロトコルタイプに応じて、以下のオプション設定を定義できます。

   - (オプション) [Custom (カスタム)] オプションを選択すると、[プロトコル] ドロップダウンメニューから値を選択できます。使用可能なプロトコル値は [TCP] および [UDP] です。

     [Port (ポート)] フィールドに 1 つのポート番号またはポート番号の範囲 (7000 ~ 8000 など) を入力することもできます。

   - (オプション) [Custom ICMP (カスタム ICMP)] オプションを選択した場合、[Type (タイプ)] フィールドで ICMP タイプを、[Code (コード)] フィールドで ICMP コードを指定できます。ICMP タイプとコードに関する詳細については、ウィキペディアの「コントロールメッセージ」を参照してください。

     > **Note**
     >
     > Lightsail コンソールを使用してインスタンスの IPv6 ファイアウォールに ICMP ルールを追加すると、このルールは ICMPv6 を使用するように自動的に設定されます。詳細については、ウィキペディアの「IPv6 のインターネットコントロールメッセージプロトコル」を参照してください。

   - (オプション) 指定したプロトコルとポートへのアクセスを特定の IP アドレスや IP アドレス範囲に制限するには、[IP アドレスに制限する] を選択します。指定したプロトコルとポートに対してすべての IP アドレスを許可する場合は、このオプションをオフのままにします。

     1 つの IPv4 アドレス (`203.0.113.1` など) または IPv4 アドレスの範囲を入力できます。その範囲は、ダッシュ表記 (`192.0.2.0-192.0.2.255` など) または CIDR ブロック表記 (`192.0.2.0/24` など) で指定できます。CIDR ブロック表記の詳細については、ウィキペディアの「Classless Inter-Domain Routing」記事を参照してください。

- (オプション) [SSH] または [RDP] アプリケーションレイヤーのプロトコルタイプを選択してから [IP アドレスに制限する] を選択した場合、[Lightsail ブラウザの SSH/RDP を許可する] を選択して、Lightsail コンソールで利用できるブラウザベースの SSH および RDP クライアントを使用してインスタンスへの接続を許可することができます。これらのブラウザベースのクライアントからのアクセスをブロックするには、このオプションをオフのままにします。

7. ルールをファイアウォールに追加するには、[作成] を選択します。

   しばらくすると、ファイアウォールルールが追加されます。

ファイアウォールルールを削除する

ファイアウォールルールの追加と編集に加えて、Amazon Lightsail インスタンスの既存のルールを削除することも可能です。特定のインバウンドトラフィックをインスタンスに許可する必要がなくなった場合、ファイアウォールルールの削除が必要になることがあります。IPv4 および IPv6 ファイアウォールルールを削除するプロセスは簡単で、Lightsail コンソールから直接実行できます。Lightsail コンソールでインスタンスのファイアウォールルールを削除するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [インスタンス] を選択してください。
3. ファイアウォールルールを削除する対象のインスタンスの名前を選択します。
4. インスタンスの管理ページで [ネットワーキング] タブを選択します。
5. ルールの送信元 IP が IPv4 アドレスか IPv6 アドレスかに応じて、次のいずれかの手順を実行します。

   - IPv4 ページの [ファイアウォール] セクションまでスクロールし、既存のルールの横にある [削除] (ゴミ箱アイコン) を選択して削除します。
   - IPv6 ページの [ファイアウォール] セクションまでスクロールし、既存のルールの横にある [削除] (ゴミ箱アイコン) を選択して削除します。

> **Important**
>
> ファイアウォールルールが影響するのは、インスタンスのパブリック IP アドレス経由で受信されるトラフィックのみです。このルールは、同じ AWS リージョン 内のアカウントにある Lightsail リソース、または同じ AWS リージョン 内のピアリングされた仮想プライベートクラウド (VPC) にあるリソースを送信元とする、インスタンスのプライベート IP アドレス経由で受信されるトラフィックには影響しません。例えば、インスタンス

のファイアウォールから SSH ルール (TCP ポート 22) を削除する場合、同じ AWS リージョン 内の同じ Lightsail アカウントにある他のインスタンスは、インスタンスのプライベート IP アドレスを指定することによって引き続き SSH を使用して接続することができます。

しばらくすると、ファイアウォールルールが削除されます。

Lightsail インスタンスのファイアウォールルールリファレンス

インスタンスのロールを反映する Amazon Lightsail インスタンスのファイアウォールにルールを追加できます。たとえば、ウェブサーバーとして設定するインスタンスには、インバウンドの HTTP および HTTPS アクセスを許可するファイアウォールルールが必要です。データベースのインスタンスには、データベースタイプへのアクセス (MySQL のポート 3306 を介したアクセスなど) を許可するルールが必要です。ファイアウォールの詳細については、「Lightsail のインスタンスファイアウォール」を参照してください。

このガイドでは、特定の種類のアクセスを対象として、インスタンスのファイアウォールに追加できるルールの種類を例として示します。ルールは、特に明記しない限り、アプリケーション、プロトコル、ポート、および送信元 IP アドレスのリスト (アプリケーション - プロトコル - ポート - 送信元 IP アドレスなど) として示します。

目次

- ウェブサーバールール
- コンピュータからインスタンスに接続するためのルール
- データベースサーバールール
- DNS サーバールール
- SMTP メール

ウェブサーバールール

次のインバウンドルールは、HTTP および HTTPS アクセスを許可します。

> **Note**
>
> Lightsail の一部のインスタンスには、デフォルトで次のファイアウォールルールが設定されます。詳細については、「ファイアウォールとポート」を参照してください。

HTTP

HTTP - TCP - 80 - すべての IP アドレス

HTTPS

HTTPS - TCP - 443 - すべての IP アドレス

コンピュータからインスタンスに接続するためのルール

インスタンスに接続するには、SSH アクセス (Linux インスタンスの場合) または RDP アクセス (Windows インスタンスの場合) を許可するルールを追加します。

> **Note**
>
> Lightsail のすべてのインスタンスには、デフォルトで次のいずれかのファイアウォールルールが設定されます。詳細については、「ファイアウォールとポート」を参照してください。

SSH

SSH - TCP - 22 - コンピュータのパブリック IP アドレス、またはローカルネットワークの IP アドレス範囲 (CIDR ブロック表記)。

RDP

RDP - TCP - 3389 - コンピュータのパブリック IP アドレス、またはローカルネットワークの IP アドレス範囲 (CIDR ブロック表記)。

データベースサーバールール

次のインバウンドルールは、インスタンスで実行中のデータベースのタイプに応じて、データベースアクセス用に追加できるルールの例です。

SQL Server

カスタム - TCP - 1433 - コンピュータのパブリック IP アドレス、またはローカルネットワークの IP アドレス範囲 (CIDR ブロック表記)。

MySQL/Aurora

MySQL/Aurora - TCP - 3306 - コンピュータのパブリック IP アドレス、またはローカルネットワークの IP アドレス範囲 (CIDR ブロック表記)。

[PostgreSQL]

PostgreSQL - TCP - 5432 - コンピュータのパブリック IP アドレス、またはローカルネットワークの IP アドレス範囲 (CIDR ブロック表記)。

Oracle-RDS

Oracle - RDS - TCP - 1521 - コンピュータのパブリック IP アドレス、またはローカルネットワークの IP アドレス範囲 (CIDR ブロック表記)。

Amazon Redshift

カスタム - TCP - 5439 - コンピュータのパブリック IP アドレス、またはローカルネットワークの IP アドレス範囲 (CIDR ブロック表記)。

DNS サーバールール

インスタンスを DNS サーバーとして設定した場合、TCP および UDP のトラフィックは、ポート 53 経由で DNS サーバーに到達できる必要があります。

DNS (TCP)

DNS (TCP) - 53 - コンピュータの IP アドレス、またはローカルネットワークの IP アドレス範囲 (CIDR ブロック表記)。

DNS (UDP)

DNS (UDP) - UDP - 53 - コンピュータの IP アドレス、またはローカルネットワークの IP アドレス範囲 (CIDR ブロック表記)。

SMTP メール

インスタンスで SMTP を有効にするには、次のファイアウォールルールを設定する必要があります。

> **Important**
>
> 次のルールを設定したら、インスタンスの逆引き DNS も設定する必要があります。そうしないと、E メールは TCP ポート 25 経由に制限される場合があります。詳細については、「E メールサーバーの逆引き DNS を設定する」を参照してください。

SMTP

カスタム - TCP - 25 - インスタンスと通信するホストの IP アドレス。

Lightsail インスタンスバーストを検出して最適なパフォーマンスを得る

> **知っていましたか?**
>
> インスタンススナップショットからインスタンスを作成するときに、インスタンスバンドルのサイズを大きく変更できます。詳細については、「Upsize a Lightsail instance, storage, or database from snapshots」を参照してください。

Amazon Lightsail インスタンスは、CPU パフォーマンスのベースライン量を提供しますが、必要に応じてベースラインを上回る CPU パフォーマンスを一時的に提供することもできます。これをバーストといいます。ベースラインパフォーマンスとバースト機能は以下のインスタンスメトリクスによって制御されます。

- CPU 使用率 - 割り当てられたコンピューティングユニットのうち、現在インスタンスで使用されているものの割合。このメトリクスは、インスタンスでアプリケーションを実行するために使用される処理能力を表します。
- CPU バースト容量の割合 - インスタンスで利用できる CPU パフォーマンスの割合。
- CPU バースト容量の分数 - インスタンスが 100% の CPU 使用率でバーストできる時間長。

次のトピックでは、インスタンスの可用性を最大化するためにこれらのメトリクスをモニタリングする方法について説明します。

トピック

- ベースライン CPU パフォーマンスと Lightsail インスタンスのバーストキャパシティの増加を理解する
- Lightsail インスタンスの CPU バーストキャパシティアクルーアルを表示する
- Lightsail インスタンスがバーストする時期を特定する
- Lightsail インスタンスの CPU バーストキャパシティをモニタリングする
- Lightsail インスタンスの CPU 使用率とバーストキャパシティを表示する
- Lightsail インスタンスの CPU 使用率が高い場合のトラブルシューティング

ベースライン CPU パフォーマンスと Lightsail インスタンスのバーストキャパシティの増加を理解する

> 知っていましたか?
>
> インスタンススナップショットからインスタンスを作成するときに、インスタンスバンドルのサイズを大きく変更できます。詳細については、「Upsize a Lightsail instance, storage, or database from snapshots」を参照してください。

Lightsail インスタンスは、1 時間あたり設定されたレートの CPU バースト容量を (ミリ秒レベルの細かさで) 継続的に蓄積します。また、このバースト容量は、インスタンスの CPU 使用率が 0% を超えると消費されます。バースト容量が蓄積されるか消費されるかの会計処理もミリ秒レベルの細かさで行われるため、CPU バースト容量の過剰消費について心配する必要はありません。CPU の短期バーストでは、バースト容量のごく一部が使用されます。

インスタンスが使用している CPU リソースがベースラインパフォーマンスに必要な数よりも少ない場合 (アイドル時など)、未使用の CPU バースト容量が CPU バースト容量の割合と分数という形で蓄積されます。インスタンスがベースラインパフォーマンスレベルを超えてバーストする必要がある場合、蓄積された CPU バースト容量を消費します。インスタンスが蓄積した CPU バースト容量が多いほど、より高いパフォーマンスが必要なときに、ベースラインを超えてバーストできる時間が長くなります。

ベースライン CPU パフォーマンス

以下の表では、Lightsail の各デュアルスタックインスタンスプランのパフォーマンスベースラインの概要を示しています。IPV6 専用のプランの価格は異なりますが、パフォーマンスの基準値は同じです。

インスタンスプラン	vCPU	メモリ	Storage	パフォーマンスのベースライン
Linux または Unix 5 ドル および Windows 9.50 ドル	2	512 MB	20 GB	5%
Linux または Unix 7 ドル および Windows 14 ドル	2	1 GB	40 GB	10%
Linux または Unix 12 ドル および Windows 22 ドル	2	2 GB	60 GB	20%
Linux または Unix 24 ドル および Windows 44 ドル	2	4 GB	80 GB	20%
Linux または Unix 44 ドル および Windows 74 ドル	2	8 GB	160 GB	30%
Linux または Unix 84 ドル および Windows 124 ドル	4	16 GB	320 GB	40%
Linux または Unix 164 ドル および Windows 244 ドル	8	32 GB	640 GB	40%
* Linux または Unix 384 ドル および Windows 574 ドル	16	64 GB	1,280 GB	40%
* Linux または Unix 884 USD および Windows 1,254 USD	32	128 GB	1,280 GB	40%
* Linux または Unix 1,324 USD および Windows 1,884 USD	48	192 GB	1,280 GB	40%
* Linux または Unix 1,764 USD および Windows 2,504 USD	64	256 GB	1,280 GB	40%

* これらのインスタンスプランは必要に応じて自動的にバーストし、バーストキャパシティを使用しません。

これらのパフォーマンス ベースラインは vCPU 単位です。Lightsail コンソールの CPU 使用率 メトリクス グラフは、複数の vCPU が割り当てられたインスタンスの平均の CPU 使用率ベースラインを表します。たとえば、44 米ドル/月のインスタンスには 2 つの vCPU が割り当てられ、平均の CPU 使用率ベースラインは 30% です。したがって、以下の場合が考えられます。

- 1 つの vCPU が 50% で動作し、もう 1 つの vCPU が 0% で動作すると、25% の平均 CPU 使用率がグラフに表示されます。これにより、インスタンスの CPU 使用率が 30% のベースラインを下回り、持続可能なゾーンに入ります。
- 1 つの vCPU が 30% で動作し、もう 1 つの vCPU が 20% で動作すると、25% の平均 CPU 使用率がグラフに表示されます。これにより、インスタンスの CPU 使用率が 30% のベースラインを下回り、持続可能なゾーンに入ります。
- 1 つの vCPU が 35% で動作し、もう 1 つの vCPU が 25% で動作すると、30% の平均 CPU 使用率がグラフに表示されます。これにより、インスタンスの CPU 使用率が 30% のベースラインになります。
- 1 つの vCPU が 100% で動作し、もう 1 つの vCPU が 90% で動作すると、95% の平均 CPU 使用率がグラフに表示されます。これにより、インスタンスの CPU 使用率が 30% のベースラインを超え、バースト可能なゾーンに入ります。

持続可能なゾーンとバースト可能なゾーンの詳細については、このガイドで後述される「インスタンスがバーストする時期の特定」を参照してください。

前世代の CPU パフォーマンス

次の表は、2023 年 6 月 29 日より前に作成された Lightsail インスタンスのパフォーマンスベースラインの概要を示しています。これらのパフォーマンス ベースラインは vCPU 単位です。

インスタンスプラン	vCPU	メモリ	Storage	パフォーマンスのベースライン
Linux または Unix 5 ドル および Windows 9.50 ドル	1	512 MB	20 GB	5%

インスタンスプラン	vCPU	メモリ	Storage	パフォーマンスのベースライン
Linux または Unix 7 ドル および Windows 14 ドル	1	1 GB	40 GB	10%
Linux または Unix 12 ドル および Windows 22 ドル	1	2 GB	60 GB	20%
Linux または Unix 24 ドル および Windows 44 ドル	2	4 GB	80 GB	20%
Linux または Unix 44 ドル および Windows 74 ドル	2	8 GB	160 GB	30%
Linux または Unix 84 ドル および Windows 124 ドル	4	16 GB	320 GB	22.5%
Linux または Unix 164 ドル および Windows 244 ドル	8	32 GB	640 GB	17%

Lightsail インスタンスの CPU バーストキャパシティアクルーアルを表示する

知っていましたか?

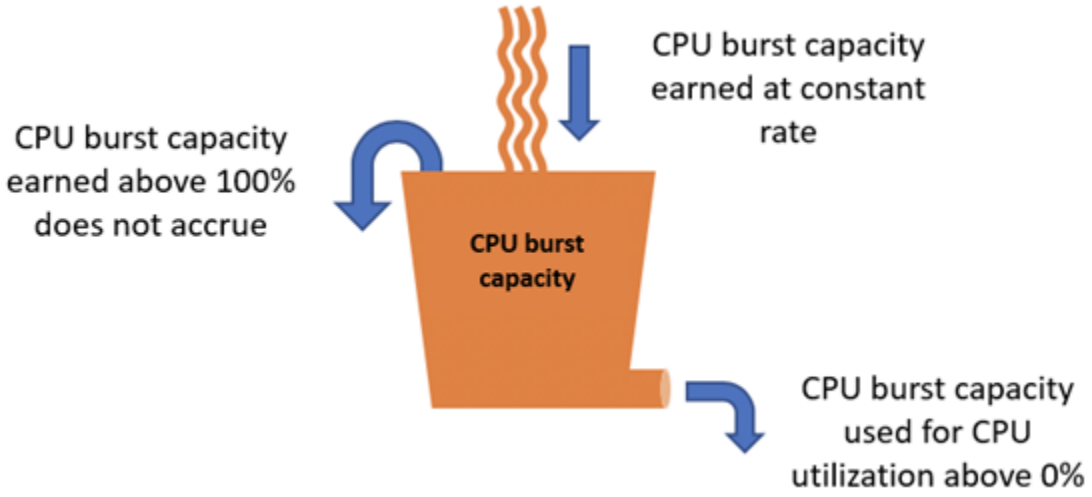
インスタンススナップショットからインスタンスを作成するときに、インスタンスバンドルのサイズを大きく変更できます。詳細については、「Upsize a Lightsail instance, storage, or database from snapshots」を参照してください。

Linux または Unix 380 ドル以上のプランおよび Windows 570 ドル以上のプランを除く Amazon Lightsail インスタンスプランでは、1 時間あたりの CPU バーストキャパシティの 4.17% を蓄積します。蓄積できる 最大 CPU バーストキャパシティの割合は、24 時間で蓄積できる CPU バーストキャパシティの割合と同じです。CPU バーストキャパシティのパーセンテージが 100% に達すると、インスタンスは CPU バーストキャパシティの蓄積を停止します。

Important

CPU バーストキャパシティの蓄積

- Linux または Unix 380 ドルおよび Windows 570 ドル以上のインスタンスプラン – これらのプランには CPU バーストキャパシティは発生しません。必要に応じて自動的にバーストします。
- 2023 年 6 月 29 日より前に作成されたインスタンス – インスタンスが停止しても、CPU バーストキャパシティは保持されません。インスタンスを停止すると、蓄積されたバースト容量がすべて失われます。
- 2023 年 6 月 29 日以降に作成されたインスタンス – CPU バーストキャパシティは、インスタンスの停止と起動の間に 7 日間保持されます。
- 実行中のインスタンスで蓄積された CPU バースト容量に有効期限はありません。



Lightsail インスタンスは、起動時に追加の CPU バースト容量を受け取ります。これは、起動 CPU バースト容量と呼ばれます。起動 CPU バースト容量を使用すると、インスタンスは起動直後にバーストしてから、追加のバースト容量を蓄積します。起動 CPU バースト容量はバースト容量の制限にカウントされません。インスタンスが起動 CPU バースト容量を消費せず、バースト容量が蓄積されている間に 24 時間以上アイドル状態のままである場合、その CPU バースト容量の割合メトリクスグラフは 100% 以上として表示されます。

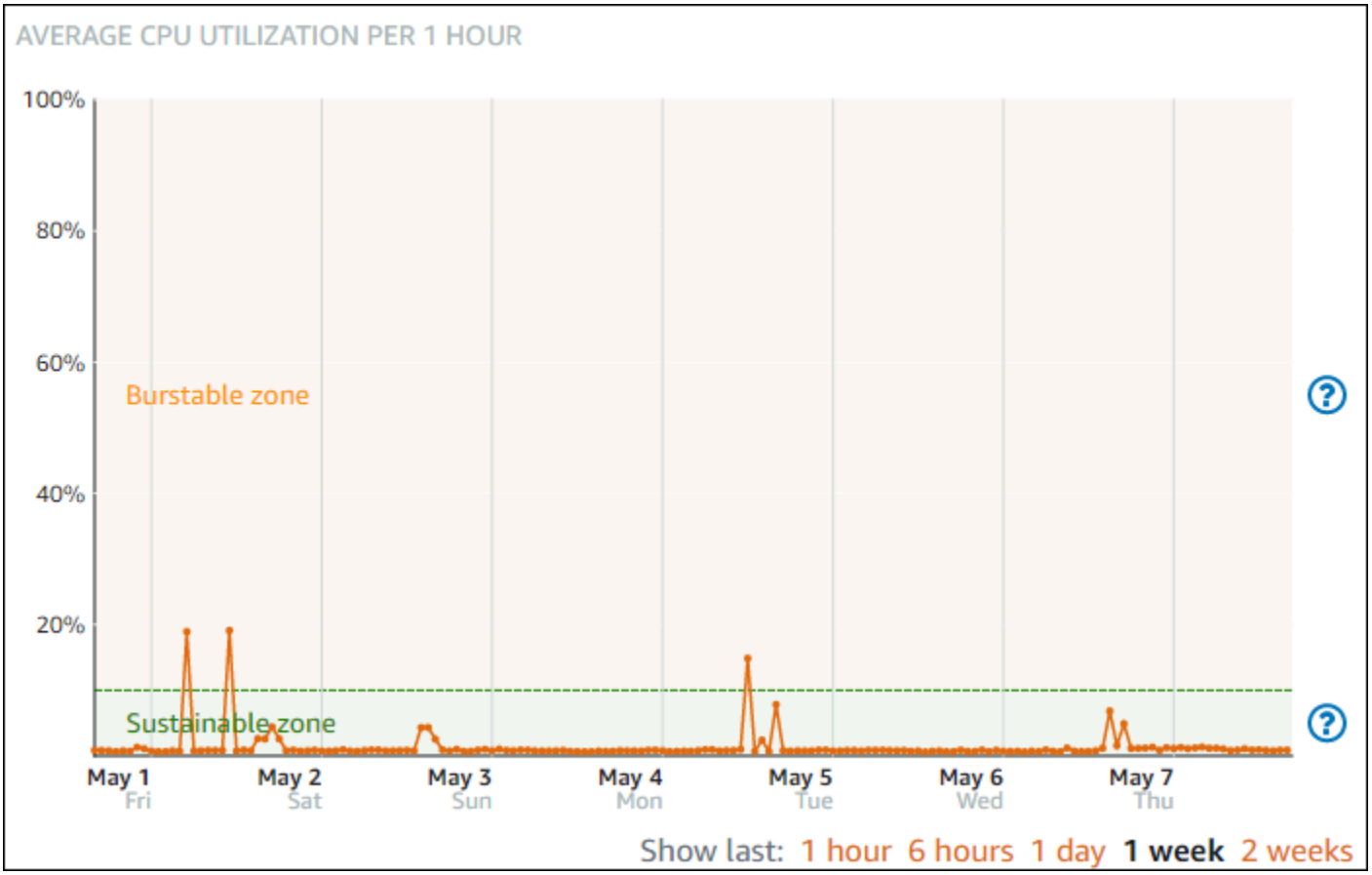
また、Lightsail一部のインスタンスは起動モードでスタートします。これにより、バースト可能なインスタンスに通常存在するパフォーマンス制限の一部が一時的に削除されます。起動モードでは、イ

ンスタンスの全体的なパフォーマンスに影響を与えずに、リソースを大量に消費するスクリプトを起動時に実行できます。

Lightsail インスタンスがバーストする時期を特定する

> 知っていましたか?
>
> インスタンススナップショットからインスタンスを作成するときに、インスタンスバンドルのサイズを大きく変更できます。詳細については、「Upsize a Lightsail instance, storage, or database from snapshots」を参照してください。

インスタンスの CPU 使用率メトリクスグラフに、持続可能なゾーンとバースト可能なゾーンが表示されます。次の CPU 使用率メトリクスグラフの例では、インスタンスが Linux または Unix ベースの 7 USD/月のインスタンスプランを使用しているため、パフォーマンスベースラインは 10% です。



Lightsail 持続可能領域内だと、インスタンスはシステムのオペレーションに影響を与えることなく無制限で作動することができます。コードのコンパイル、新しいソフトウェアのインストール、バッチ

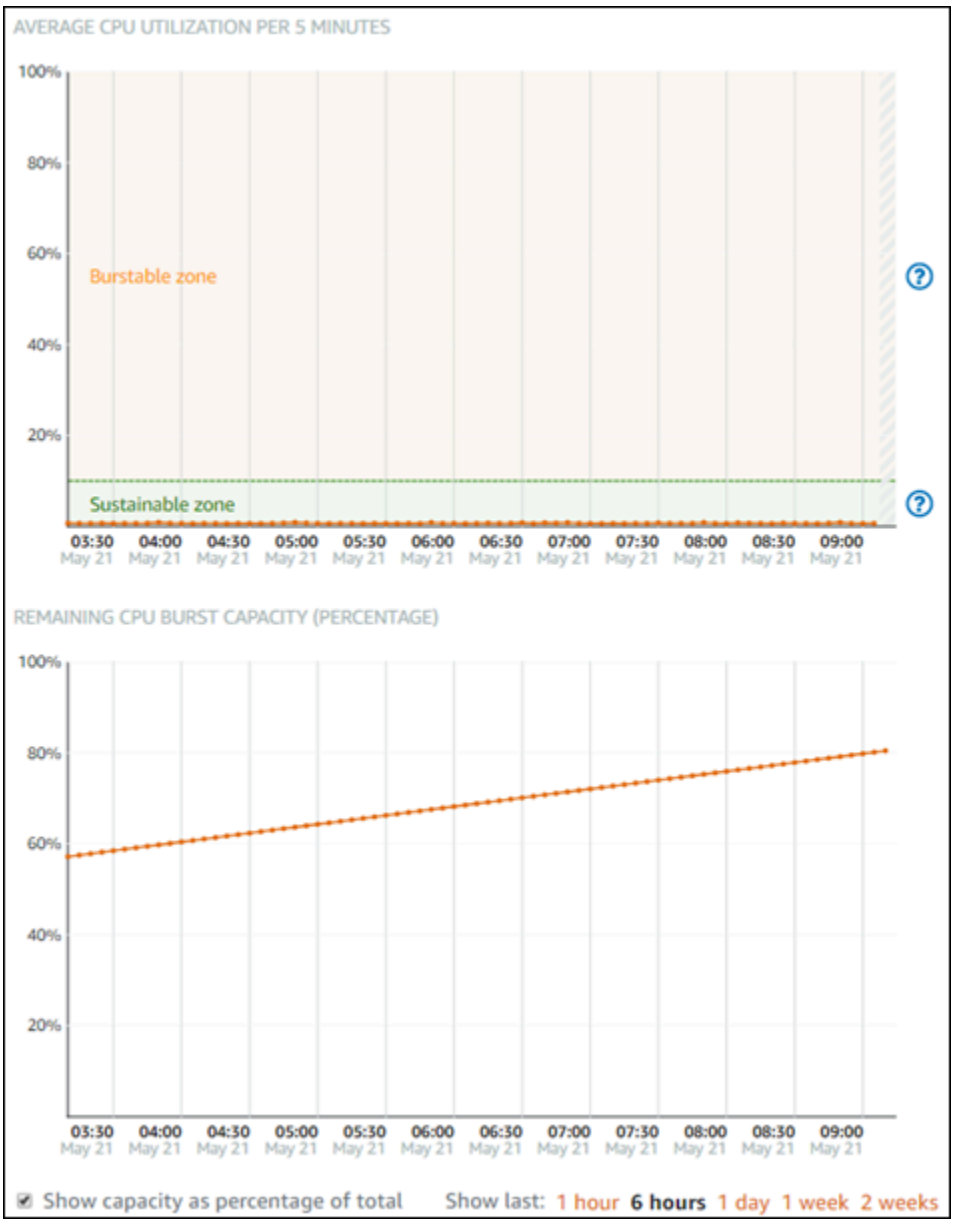
ジョブの実行、ピークの負荷リクエストの処理など、負荷が高い場合、インスタンスがバースト可能なゾーンで動作し始めることがあります。バーストな可能ゾーンで動作している間、インスタンスは大量の CPU サイクルを消費します。したがって、この領域では限られた期間しか作動できません。

インスタンスがバースト可能なゾーンで動作できる期間は、バースト可能なゾーンにどの程度入っているかによって異なります。バースト可能なゾーンの下限近くで動作しているインスタンスは、バースト可能なゾーンの上限近くで動作しているインスタンスよりも長い時間バーストできます。ただし、一定期間バースト可能なゾーンにあるインスタンスは、持続可能なゾーンで再び動作するまで、最終的にすべての CPU 容量を使い果たすことになります。したがって、このガイドの次のセクションで説明する残り CPU バースト容量もモニタリングすることが重要です。

Lightsail インスタンスの CPU バーストキャパシティをモニタリングする

> **知っていましたか?**
>
> インスタンススナップショットからインスタンスを作成するときに、インスタンスバンドルのサイズを大きく変更できます。詳細については、「Upsize a Lightsail instance, storage, or database from snapshots」を参照してください。

Lightsail コンソールの CPU 概要ページには、インスタンスの CPU 使用率が、使用可能な CPU バースト容量と比較して表示されます。以下の CPU 概要の例では、インスタンスが持続可能なゾーンでベースラインを下回って継続的に動作しているため、CPU バースト容量の割合が増加しています。



残り CPU バースト容量のグラフビューは、CPU バースト容量の割合と分数で切り替えることができます。バースト可能なゾーンで動作しているとき、インスタンスはより多くの CPU バースト容量を消費します。CPU バースト容量の分数メトリクスは、インスタンスが 100% の CPU 使用率でバーストできる時間長です。インスタンスがバースト可能なゾーンで動作しているとき、CPU バースト容量 (割合) がインスタンスの現在の CPU 使用率と同じレートで消費されます。例えば、Linux または Unix ベース 7 米ドル/月のインスタンスの CPU 使用状況のベースラインは 10% で、1 時間あたり 6 分の CPU バーストキャパシティ (分間数) が累積されます。したがって、以下の場合が考えられます。

- 60 分間、バースト可能なゾーンでの CPU 使用率が 100% のとき、その期間中、CPU バースト容量 (分数) が 100% のレートで消費されます。インスタンスは 60 分の CPU バーストキャパシティを消費し、6 分を累積するため、正味 54 分が消費されます。
- 60 分間、バースト可能なゾーンでの CPU 使用率が 50% のとき、その期間中、CPU バースト容量 (分数) が 50% のレートで消費されます。インスタンスは 30 分の CPU バーストキャパシティを消費し、6 分を累積するため、正味 24 分が消費されます。
- 60 分間、インスタンスのベースラインでの CPU 使用率が 10% のとき、その期間中、CPU バースト容量 (分数) が 10% のレートで消費されます。インスタンスは 6 分の CPU バースト容量を消費し、6 分を蓄積します。インスタンスがベースラインで動作しているとき、CPU バースト容量の分数は増減しません。
- 60 分間、持続可能なゾーンでの CPU 使用率が 5% のとき、その期間中、CPU バースト容量 (分数) が 5% のレートで消費されます。インスタンスは 3 分の CPU バーストキャパシティを消費し、6 分を累積するため、正味 3 分が累積されます。

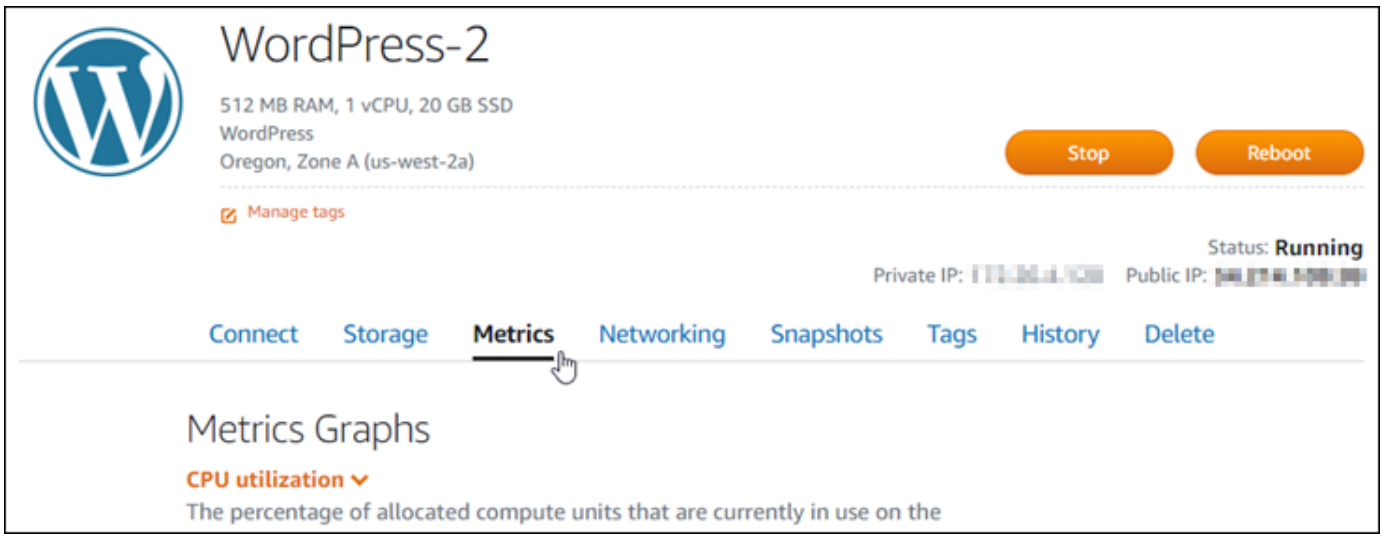
あるいは、インスタンスは 60 分の CPU バースト容量を蓄積した場合、CPU 使用率 100% で 60 分間、50% で 120 分間、または 25% で 150 分間動作できます。

Lightsail インスタンスの CPU 使用率とバーストキャパシティを表示する

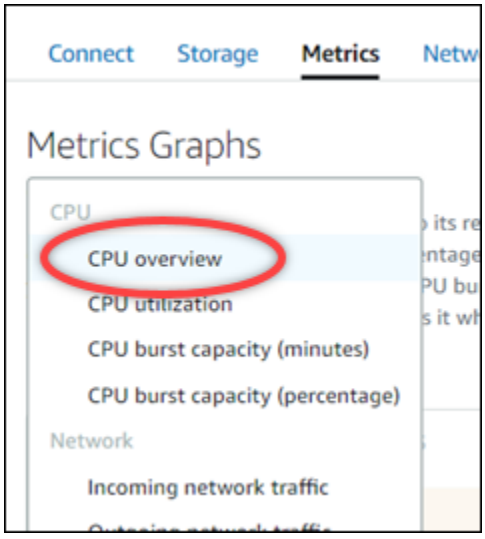
> 知っていましたか?
>
> インスタンススナップショットからインスタンスを作成するときに、インスタンスバンドルのサイズを大きく変更できます。詳細については、「Upsize a Lightsail instance, storage, or database from snapshots」を参照してください。

CPU 概要ページにアクセスし、インスタンスの CPU 使用率と残り CPU バースト容量を表示するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページで、CPU 使用率とバースト容量を表示するインスタンスの名前を選択します。
3. インスタンス管理ページで [Metrics (メトリクス)] タブを選択します。



4. [Metrics graphs (メトリクスグラフ)] 見出しの下のドロップダウンメニューで [CPU overview (CPU 概要)] を選択します。



このページには、[5分間の平均 CPU 使用率] と [残りの CPU バーストキャパシティ] のグラフが表示されます。

> **Note**
>
> インスタンスを作成した後、しばらくの間、[残りの CPU バーストキャパシティ] のグラフに [起動モード] ゾーンが表示されることがあります。ある程度の Lightsail インスタンスは起動モードで起動します。これにより、バースト可能なインスタンスに通常存在するパフォーマンス制限の一部が一時的に削除されます。起動モードでは、インスタンス

の全体的なパフォーマンスに影響を与えずに、リソースを大量に消費するスクリプトを起動時に実行できます。

5. メトリクスグラフでは、以下のアクションを実行できます。

   - バースト容量グラフで、[Show capacity as percentage of total (合計容量の割合として容量を表示)] を選択して、ビューを使用可能なバースト容量の分数から使用可能なバースト容量の割合に変更します。
   - グラフの表示を変更して、1 時間、6 時間、1 日、1 週間、2 週間のデータを表示します。

- データポイント上にカーソルを置くと、そのデータポイントに関する詳細情報が表示されます。
- CPU 使用率とバースト容量が指定したしきい値を超えたときに通知するアラームを追加します。CPU 概要ページでアラームを追加することはできません。それらのアラームは、個々の CPU 使用率、CPU バースト容量の割合、CPU バースト容量の分数メトリクスグラフのページで追加する必要があります。詳細については、「アラーム」および「インスタンスのメトリクスアラームを作成する」を参照してください。

Lightsail インスタンスの CPU 使用率が高い場合のトラブルシューティング

> **知っていましたか?**
>
> インスタンススナップショットからインスタンスを作成するときに、インスタンスバンドルのサイズを大きく変更できます。詳細については、「Upsize a Lightsail instance, storage, or database from snapshots」を参照してください。

インスタンスがバースト可能なゾーンで頻繁にまたは長期間にわたって動作する場合、インスタンスはすべてのバースト容量を使用します。これは、インスタンスがプロビジョニング不足であることを示している可能性があります。また、サービスの実行頻度が高すぎるか、インスタンスで不要なソフトウェアが実行されていることを示している可能性もあります。

Linux/Unix インスタンスの top や Windows Server インスタンスのタスクマネージャーなどのツールを使用して、インスタンスのバーストの原因を調査します。これらのツールでは、インスタンスでリソースを消費しているサービスが表示されます。最も多くのリソースを消費しているサービスを特定し、インスタンスのワークロードに影響を与えずにそれらのサービスを無効にできるかどうかを決定します。サービスを無効にするか、ソフトウェアをアンインストールすることで、インスタンスのバーストを減らすことができ、インスタンスのサイズを大きくする必要がなくなる場合があります。

インスタンスが実際にプロビジョニング不足で、CPU 使用率を下げることができない場合は、処理能力を増やすことでバースト容量の消費を減らすことができます。そのためには、インスタンスのスナップショットを作成し、Lightsail インスタンスのより大きなプランを使用して、スナップショットから新しいインスタンスを作成します。たとえば、以前のインスタンスで使用されていた Linux または Unix ベースの月額 12 USD プランの代わりに、新しいインスタンスで Linux または Unix ベースの月額 24 USD プランを使用します。新しいインスタンスが稼働中になったら、必要に応じてワークロードの DNS を変更して、古いインスタンスを新しいインスタンスと交換します。トラフィック

が新しいインスタンスへルーティングされ始めたら、プロビジョニング不足の古いインスタンスを削除します。詳細については、「スナップショット」を参照してください。

Lightsail インスタンスに接続して管理する

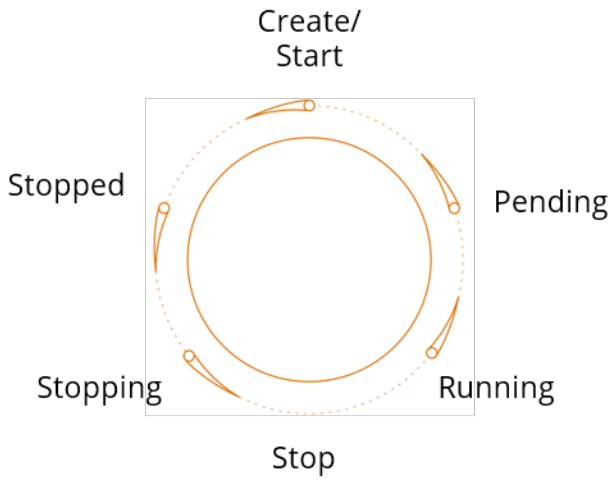
このガイドでは、Amazon Lightsail インスタンスの管理と接続に関する以下のトピックについて説明します:

トピック

- Lightsail インスタンスの開始、停止、または再起動
- スタックした Lightsail インスタンスを強制停止する
- Amazon EC2 インスタンスの拡張ネットワーキングを有効にする
- Lightsail Windows Server インスタンスのファイルシステムを拡張する
- Lightsail で起動スクリプトを使用して Linux/Unix インスタンスを設定する
- PowerShell またはバッチスクリプトを使用して Windows Lightsail インスタンスを設定する
- Lightsail で Windows Server インスタンスを保護する

Lightsail インスタンスの開始、停止、または再起動

Amazon Lightsail でインスタンスを作成すると、ユーザーのマシンは [保留中] 状態になった後で [実行中] 状態に移行します。インスタンスが実行中になると、そのインスタンスを再起動するか、停止して起動できます。そのサイクルは次のようになっています。

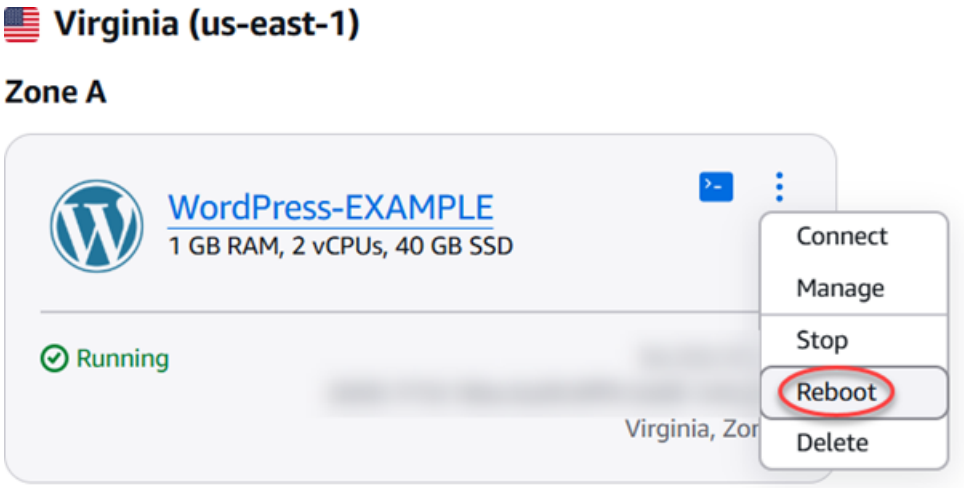


インスタンスの状態は、インスタンスを管理する場合やホームページでインスタンスを表示する場合に確認できます。

> **Important**
>
> インスタンスの作成時にインスタンスに割り当てられるデフォルトの公開 IPv4 アドレスは、インスタンスを停止してまた開始すると変更されます。必要に応じて、静的 IPv4 アドレスをインスタンスに作成し、アタッチできます。静的 IPv4 アドレスは、インスタンスのデフォルトの公開 IPv4 アドレスに置き換えられ、インスタンスを停止して起動しても変わりません。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。

実行中のインスタンスの再起動

- ホームページで再起動するインスタンスを選択するか、インスタンス管理メニューで [再起動] を選択します。

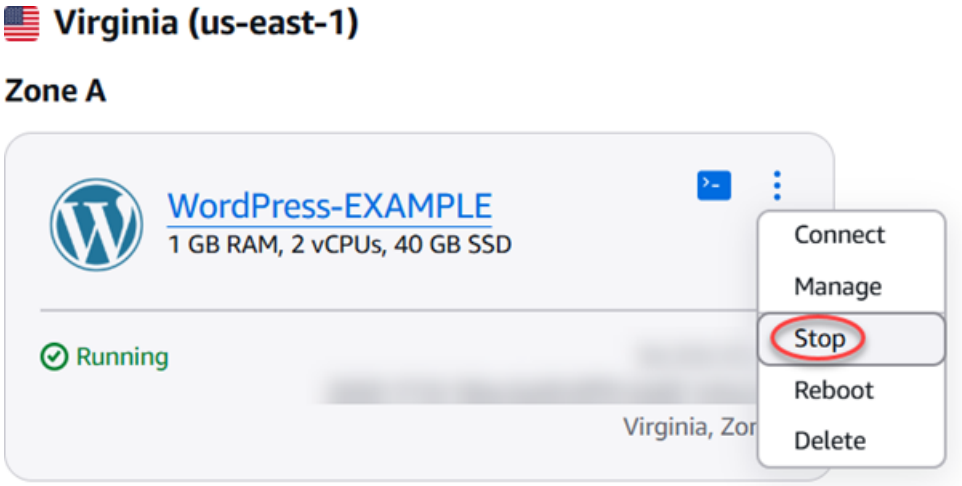


インスタンス管理ページでインスタンスを表示している場合は、[再起動] を選択し、プロンプトが表示されたら [確認] を選択します。

> **Note**
>
> インスタンスを[再起動]するには、そのインスタンスが [実行中] 状態である必要があります。

実行中のインスタンスの停止

- ホームページで、停止するインスタンスを選択し、インスタンス管理メニューで [停止] を選択します。

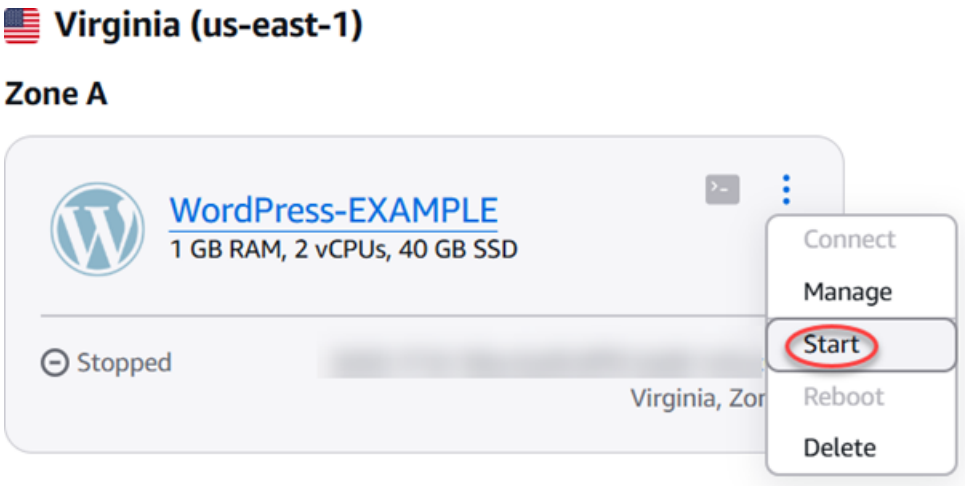


インスタンス管理ページでインスタンスを表示している場合は、[停止] を選択し、プロンプトが表示されたら [確認] を選択します。

> **Note**
>
> インスタンスを停止するには、そのインスタンスが [実行中] 状態である必要があります。

停止した後のインスタンスの開始

- ホームページで、開始するインスタンスを選択し、インスタンス管理メニューで [開始] を選択します。



インスタンス管理ページでインスタンスを表示している場合は、[開始] を選択します。

> **Note**
>
> インスタンスを開始するには、そのインスタンスが [停止] 状態である必要があります。

スタックした Lightsail インスタンスを強制停止する

まれに、インスタンスが Stopping 状態でスタックすることがあります。これが発生した場合は、Amazon Lightsail インスタンスをホストする基盤となるハードウェアに問題がある可能性があります。このガイドでは、stopping 状態でスタックしたインスタンスを強制停止する方法を説明します。インスタンスの状態の詳細については、「Lightsail インスタンスの開始、停止、または再起動」を参照してください。

インスタンスを強制停止する方法

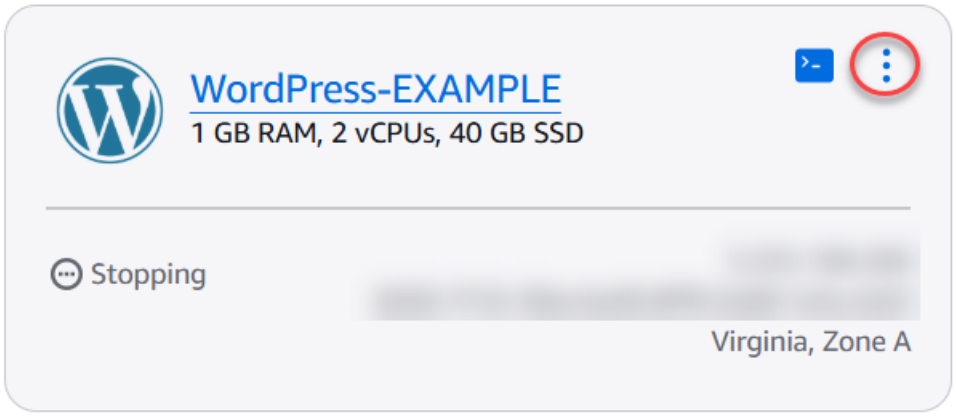
Lightsail コンソールを使用してインスタンスを強制停止できますが、これはインスタンスが stopping 状態にある間のみ可能です。または、インスタンスが shutting-down と terminated の状態にあるとき以外であれば、 AWS Command Line Interface (AWS CLI) を使用してインスタンスを強制停止することもできます。強制停止が完了するまでに数分かかる場合があります。10 分経ってもインスタンスが停止しない場合は、再度強制停止します。

インスタンスが強制的に停止される際に、ファイルシステムのキャッシュやファイルシステムのメタデータをフラッシュする機会はありません。インスタンスを強制停止した後、ファイルシステムのチェックと修復手順を実行する必要があります。

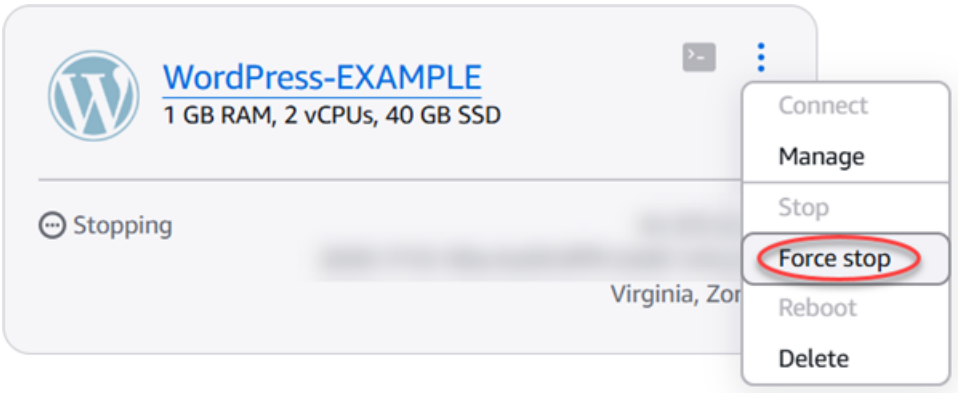
次の手順では、Lightsail インスタンスを強制停止するさまざまな方法について説明します。

Lightsail コンソールでインスタンスを強制停止する

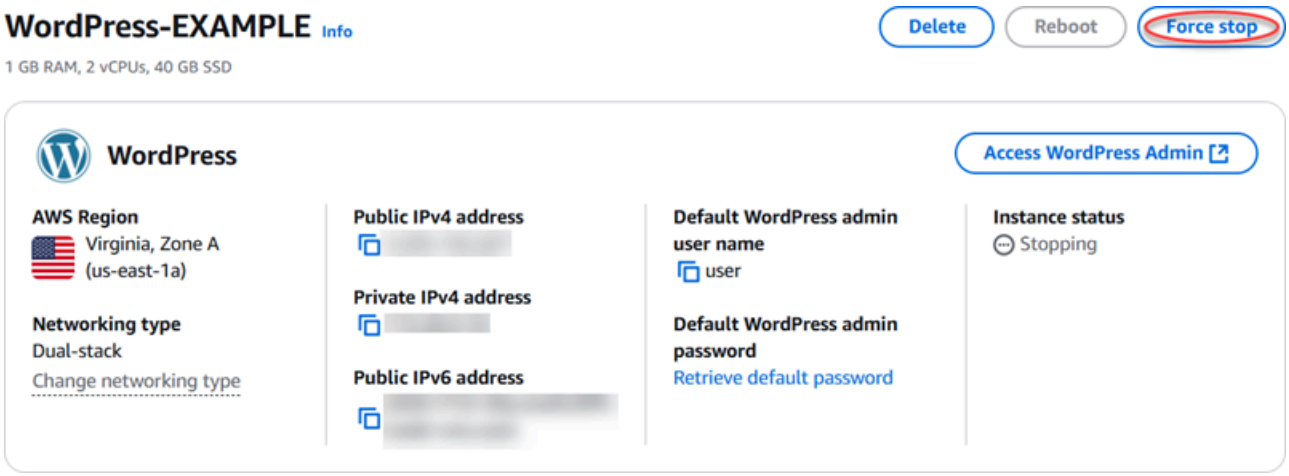
1. Lightsail コンソール にサインインします。
2. [Instances] タブを選択します。
3. `Stopping` 状態でスタックしているインスタンスを特定します。その後、インスタンス名の横に表示されるアクションメニューアイコン (⋮) を選択します。



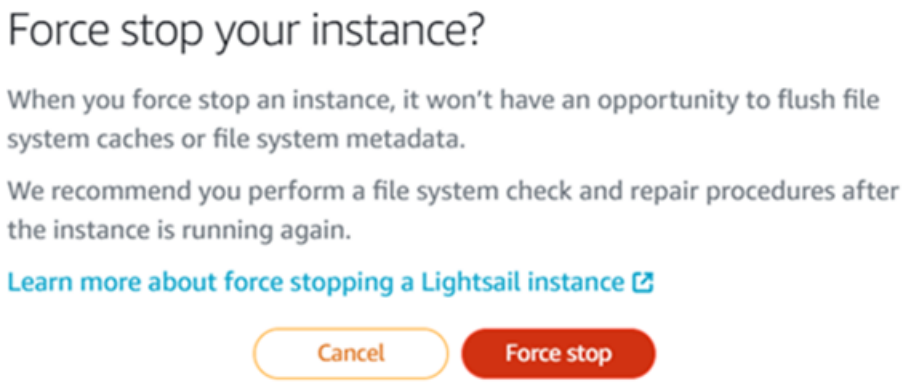
4. 表示されるドロップダウンリストで [強制停止] を選択します。



あるいは、インスタンス名を選択してインスタンス管理ページにアクセスすることもできます。その後、[強制停止] ボタンを選択します。



5. このオペレーションに関する考慮事項を確認してください。続行するには、[強制停止] を選択してください。



を使用してインスタンスを強制停止する AWS CLI

1. 開始する前に、 AWS CLIをインストールする必要があります。詳細については、「AWS Command Line Interfaceのインストール」を参照してください。インストール後は必ず AWS CLIを設定してください。
2. stop-instance コマンドと `--force` パラメータを次のように使用します。

```
aws lightsail stop-instance --instance-name Wordpress-1 --force
```

Amazon EC2 インスタンスの拡張ネットワーキングを有効にする

一部の Lightsail インスタンスは、拡張ネットワーキングに対応していないため、現行世代の EC2 インスタンスタイプ (T3、M5、C5、または R5) と互換性がありません。ソースの Lightsail インスタン

スに互換性がない場合は、エクスポートしたスナップショットから EC2 インスタンスを作成するときに、以前の世代のインスタンスタイプ (T2、M4、C4、または R4) から選択する必要があります。これらのインスタンスタイプオプションは、Lightsail コンソールの[Amazon EC2 インスタンス作成]ページを使用して EC2 インスタンスを作成するときに表示されます。

> **Note**
>
> 拡張ネットワークの詳細については、「Amazon EC2 ドキュメント」の「Linux での拡張ネットワーク」または「Windows での拡張ネットワーク」を参照してください。

ソースの Lightsail インスタンスに互換性がない場合に最新世代の EC2 インスタンスタイプを使用するには、まず以前の世代のインスタンスタイプ (T2、M4、C4、または R4) を使用して新しい EC2 インスタンスを作成し、インスタンスのネットワーキングドライバーを更新します。次に、インスタンスを目的の最新世代のインスタンスタイプに更新します。

前提条件

エクスポートした Lightsail スナップショットから Amazon EC2 インスタンスを作成する必要があります。Lightsail インスタンスに互換性がない場合は、Amazon EC2 インスタンスの作成時に以前の世代のインスタンスタイプ (T2、M4、C4、または R4) を選択します。詳細については、「Lightsail でエクスポートしたスナップショットから Amazon EC2 インスタンスを作成する」を参照してください。

新しい EC2 インスタンスが起動して実行中になったら、このガイドの「Elastic Network Adapter で拡張ネットワーキングを有効にする」セクションに進み、拡張ネットワーキングを有効にする方法を確認します。

Elastic Network Adapter で拡張ネットワーキングを有効にする

新しいインスタンスが起動して実行中になったら、以下のいずれかの「Amazon EC2 ドキュメント」のガイドを参照して Elastic Network Adapter (ENA) で拡張ネットワーキングを有効にします。

- Linux インスタンスにおける Elastic Network Adapter (ENA) を使用した拡張ネットワーキングの有効化
- Windows インスタンスにおける Elastic Network Adapter (ENA) を使用した拡張ネットワーキングの有効化

インスタンスタイプをアップグレードする

拡張ネットワーキングを有効にしたら、以下のいずれかのガイドの手順に従ってインスタンスタイプをアップグレードできます。

- Windows Server インスタンスの場合 — 最新世代のインスタンスタイプへの移行
- Linux または Unix インスタンスの場合 — インスタンスタイプを変更する

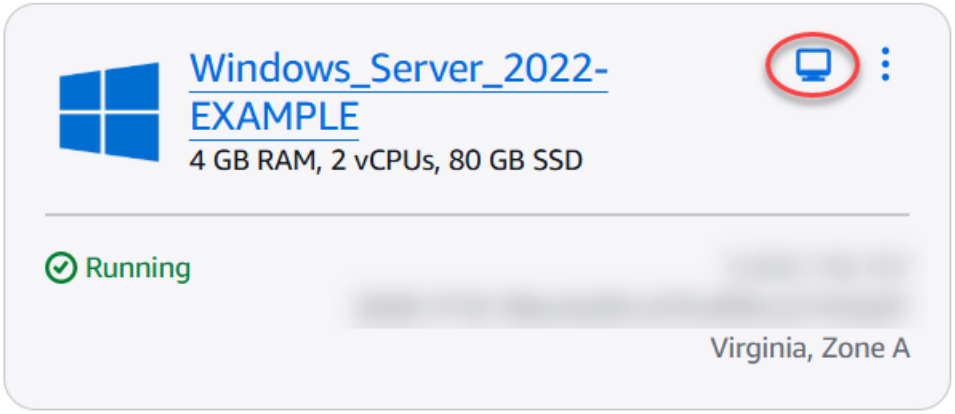
Lightsail Windows Server インスタンスのファイルシステムを拡張する

スナップショットを使用してより大きなプランの新しい Windows Server インスタンスを作成すると、使用可能なストレージ領域がプランに指定されている領域より小さいことに気づく場合があります。通常、より大きなプランに指定されている追加分のストレージ領域は未割り当てのため、アクティブボリュームで使用されないことが原因です。このトピックの手順では、Windows Server インスタンスのファイルシステムを拡張し、使用可能なストレージ領域を最大限に利用する方法を示します。

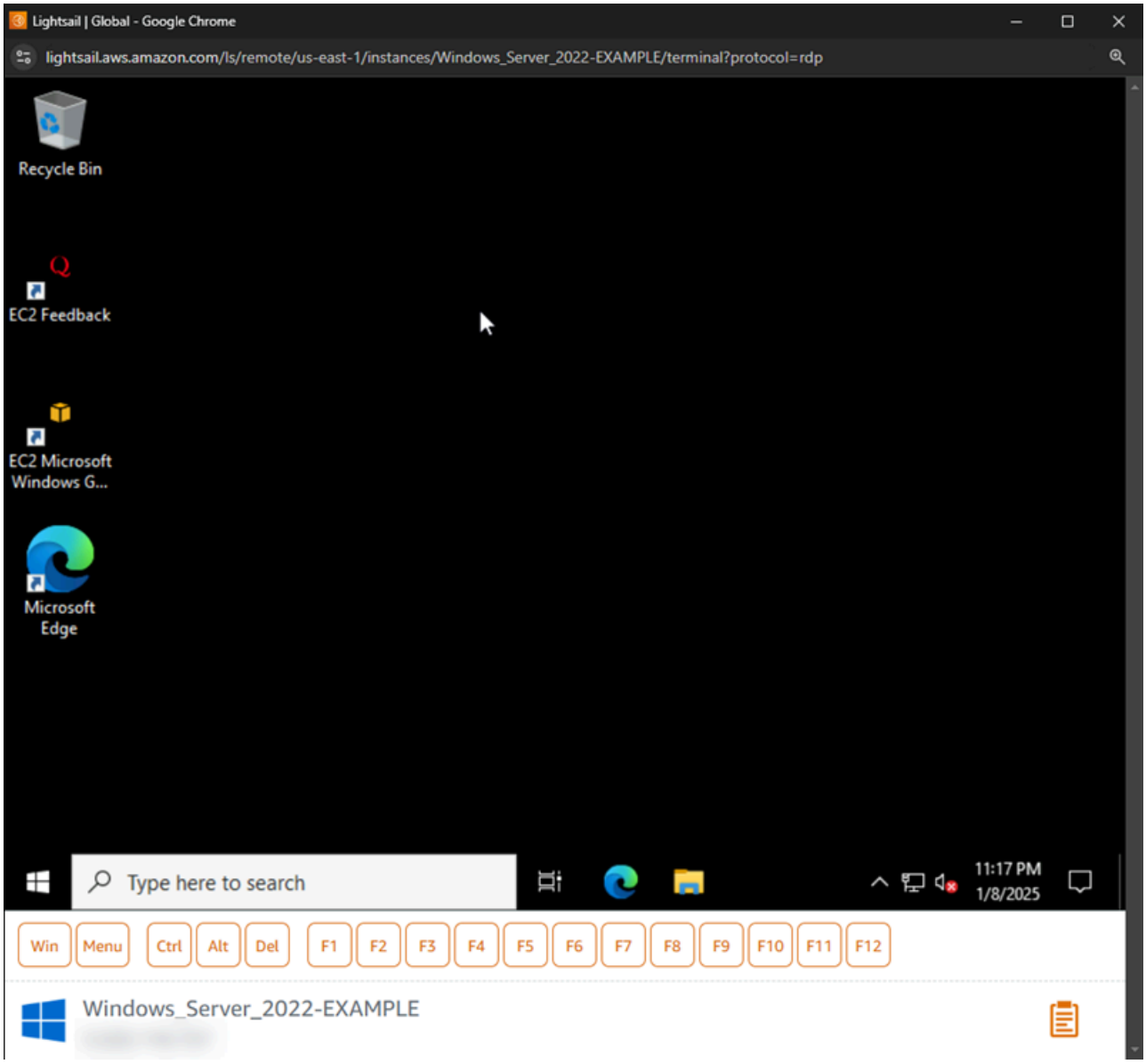
> **Note**
>
> この状況が発生するのは、システム準備 (Sysprep) ユーティリティの実行前に作成したスナップショットを使用して Windows Server インスタンスを作成した場合に限ります。詳細については、「Windows Server インスタンスのスナップショットを作成する」を参照してください。

Windows Server インスタンスのファイルシステムを拡張するには

1. Lightsail コンソール にサインインします。
2. Lightsail のホームページで、接続先のインスタンスの RDP クライアントアイコンを選択します。



次の例に示すように、ブラウザベースの RDP クライアントウィンドウが開きます。

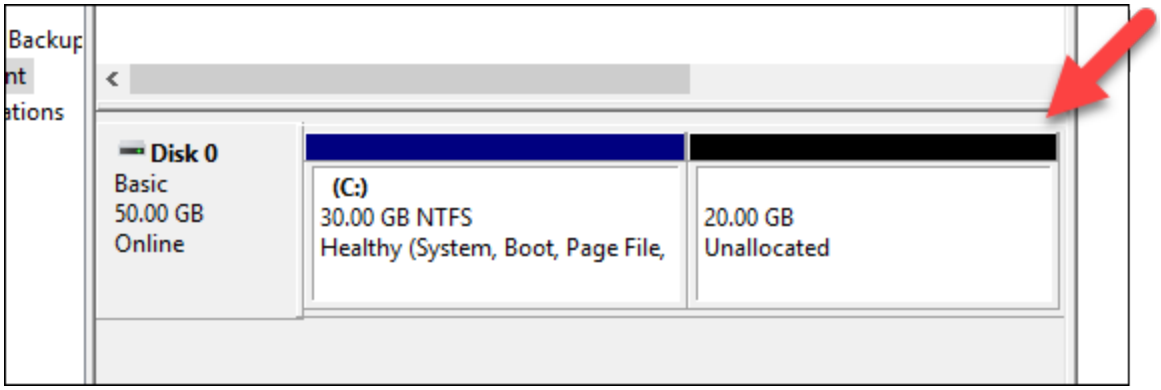


3. タスクバーで Windows アイコンを選択し、以下のいずれかのオプションを選択します。

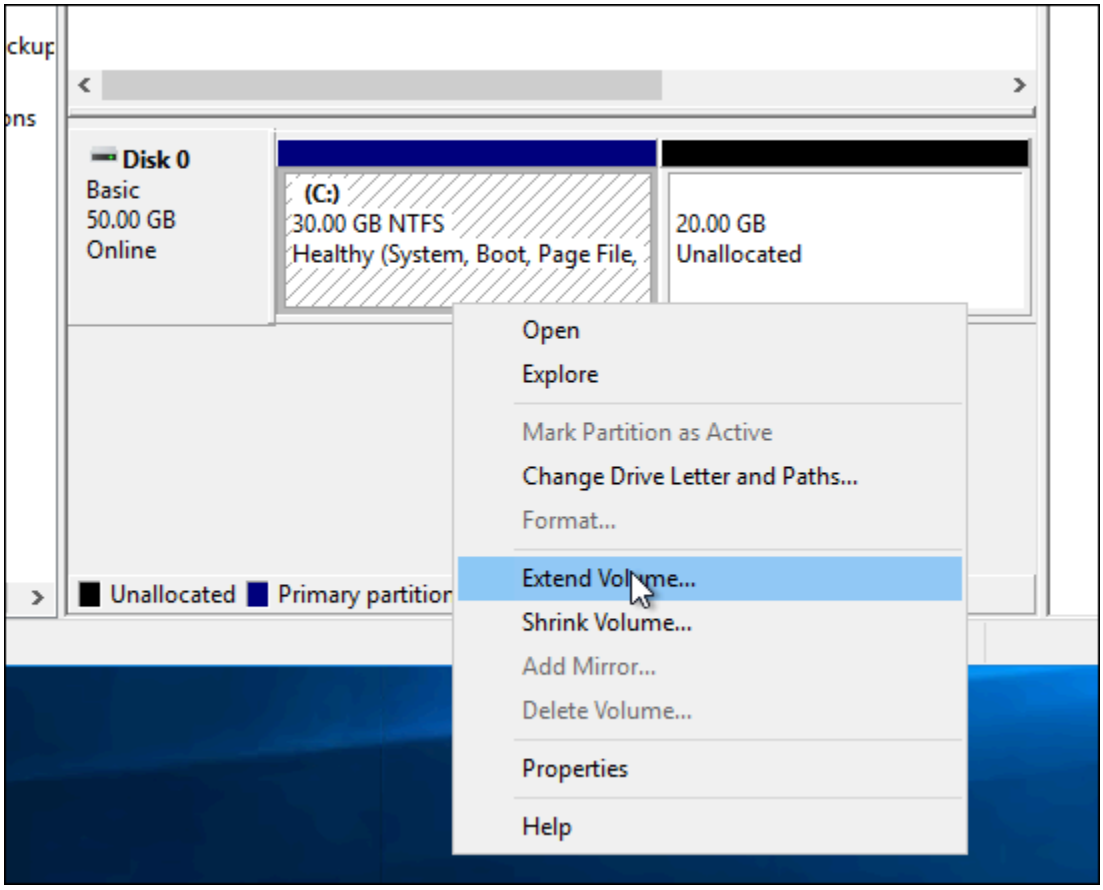
   - Windows Server 2022、Windows Server 2019 または Windows Server 2016 インスタンスの場合は、[スタート]、[Windows 管理ツール] の順に選択します。

4. [コンピューターの管理] を選択します。
5. [コンピューターの管理] コンソールの左側のペインで、[ディスクの管理] を選択します。
6. [操作] メニューの [ディスクの再スキャン] を選択します。

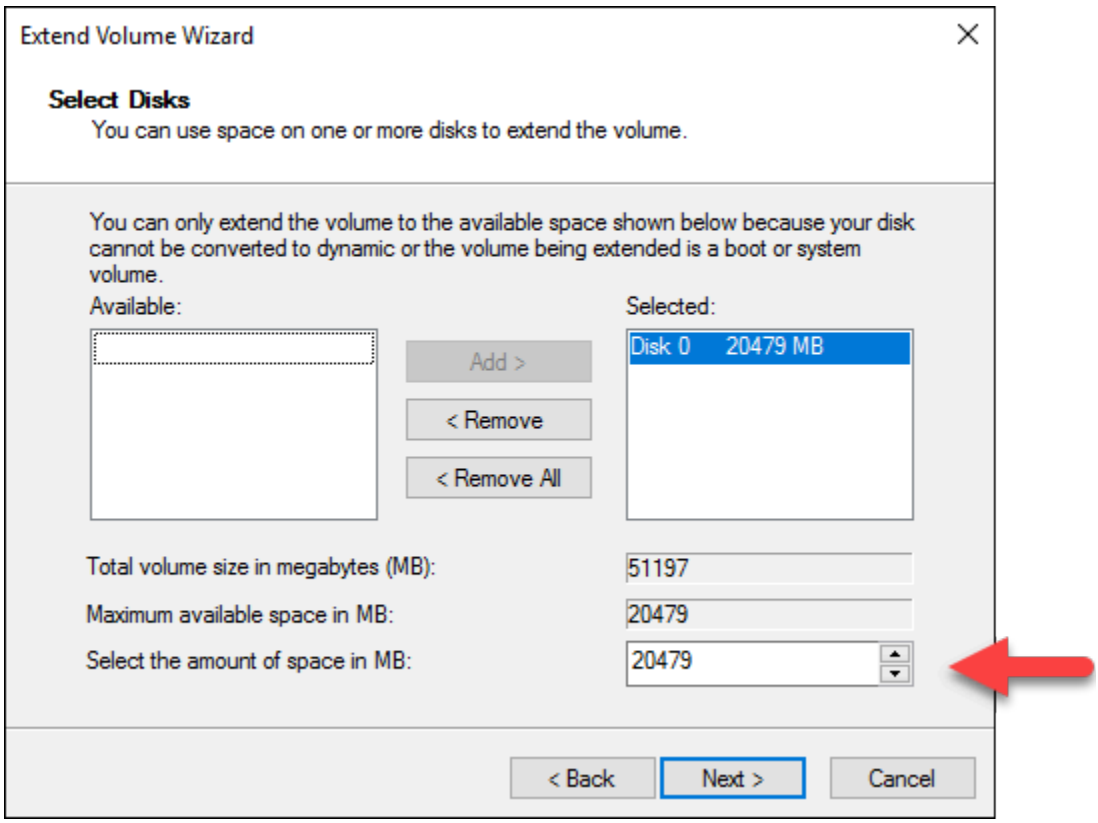
   ディスクに関連付けられている未割り当て領域が表示される場合があります。未割り当て領域を利用するようにディスクのアクティブボリュームを拡張します。



7. 未割り当て領域と同じディスクのアクティブボリュームを右クリックし、[ボリュームの拡張] を選択します。

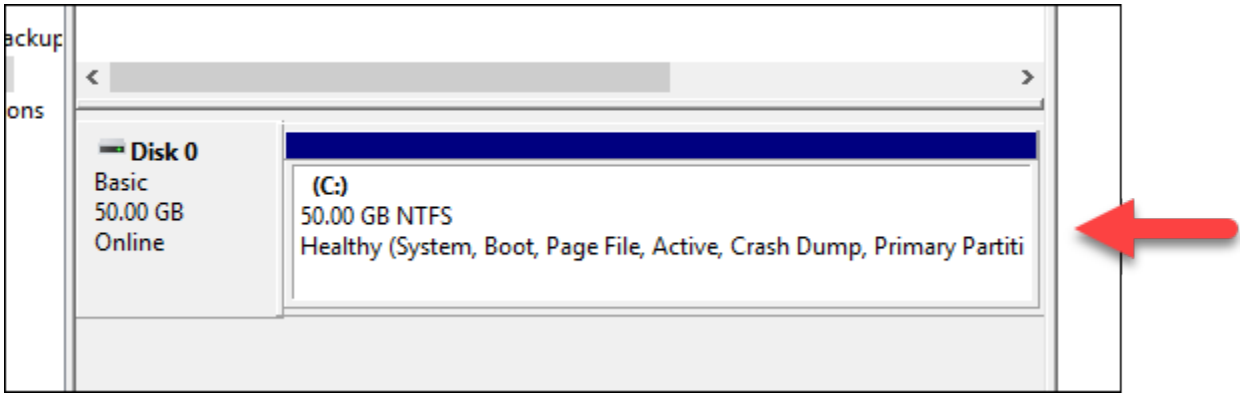


8. ボリュームの拡張ウィザードが開いたら、[次へ] を選択します。
9. [ディスク領域 (MB) を選択] フィールドで、ボリュームを拡張するメガバイト数を入力します。通常、これは最大の未割り当て領域に設定します。入力する値は、ボリュームの最終的なサイズではなく、追加する領域のサイズです。



10. ボリュームの拡張ウィザードを完了します。

    指定した未割り当て領域を利用するようにアクティブボリュームが拡張されます。次の例では、すべての未割り当て領域が選択されています。



Lightsail で起動スクリプトを使用して Linux/Unix インスタンスを設定する

Linux または Unix ベースのインスタンスの作成時に、ソフトウェアの追加、ソフトウェアの更新、インスタンスの設定などを他の方法で行うことで、起動スクリプトを追加できます。追加データで

Windows ベースのインスタンスを設定するには、「Windows PowerShell を使用して新しい Lightsail インスタンスを設定する」を参照してください。

> **Note**
>
> インスタンスでソフトウェアを取得するためのコマンドは、選択したマシンイメージに応じて異なります。Amazon Linux では yum が使用され、Debian および Ubuntu の両方では apt-get が使用されます。WordPress およびその他のアプリケーションイメージは、オペレーティングシステムとして Debian を使用しているため、apt-get を使用します。FreeBSD と openSUSE は、freebsd-update や zypper (openSUSE) などのカスタムツールを使用するために追加のユーザー設定が必要です。

例: Node.js をインストールするように Ubuntu サーバーを設定する

次の例では、apt-get コマンドを使用して、パッケージリストを更新し、Node.js をインストールしています。

1. [インスタンスを作成する] ページの [OS のみ] タブで [Ubuntu] を選択します。
2. 下にスクロールして [起動スクリプトの追加] を選択します。
3. 次の内容を入力します。

```
# update package list
apt-get update -y
# install some of my favorite tools
apt-get install nodejs -y
```

> **Note**
>
> サーバーを設定するために送信するコマンドは root として実行されるため、コマンドの前に sudo を付ける必要はありません。

4. [インスタンスの作成] を選択します。

例: プラグインをダウンロードしてインストールするように WordPress サーバーを設定する

次の例では、パッケージリストを更新し、WordPress 用の BuddyPress プラグインをダウンロードしてインストールしています。

1. [インスタンスを作成する] ページで [WordPress] を選択します。
2. [起動スクリプトの追加] を選択します。
3. 次の内容を入力します。

```
# update package list
apt-get update
# download wordpress plugin
wget "https://downloads.wordpress.org/plugin/buddypress.14.0.0.zip"
apt-get install unzip
# unzip into wordpress plugin directory
unzip buddypress.14.0.0.zip -d /bitnami/wordpress/wp-content/plugins
```

4. [インスタンスの作成] を選択します。

PowerShell またはバッチスクリプトを使用して Windows Lightsail インスタンスを設定する

Windows ベースのインスタンスを作成するとき、Windows PowerShell スクリプトまたは他のバッチスクリプトを使用して設定することができます。これは、インスタンスの起動直後に実行されるワンタイムスクリプトです。このトピックでは、スクリプトの構文と使用を開始するための例を示します。スクリプトが正常に実行されたかどうかをテストする方法も示します。

PowerShell スクリプトを起動して実行するインスタンスを作成する

次の手順では、インスタンスの起動直後に chocolatey というツールを新しいインスタンスにインストールします。

1. 左側のナビゲーションペインで、[インスタンス] を選択します。
2. インスタンスを作成する AWS リージョン とアベイラビリティーゾーンを選択します。
3. [プラットフォームの選択] で [Microsoft Windows] を選択します。
4. [OS のみ] を選択した後、[Windows Server 2022]、[Windows Server 2019]、[Windows Server 2016] を選択します。

5. [起動スクリプトの追加] を選択します。
6. 次の内容を入力します。

```
<powershell>
iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/
install.ps1'))
</powershell>
```

> **Note**
>
> PowerShell スクリプトは常に <powershell></powershell> タグで囲む必要があります。PowerShell コマンドまたはバッチスクリプトは、<script></script> タグを使用するか、タグをまったく使用せずに入力できます。

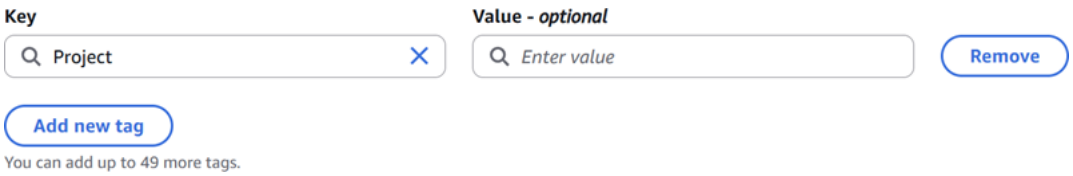
7. インスタンスの名前を入力します。

   リソース名:

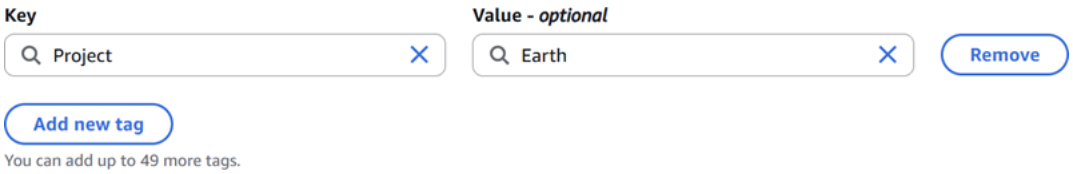
   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

8. (オプション)[新しいタグの追加] を選択してインスタンスにタグを追加します。必要に応じてこの手順を繰り返して、他のタグを追加します。タグの使用の詳細については、「タグ」を参照してください。

   a. [キー] には、タグキーを入力します。

   b. (オプション) [値] にタグの値を入力します。

9. [インスタンスの作成] を選択します。

スクリプトが正常に実行されたことを確認する

インスタンスにログインして、スクリプトが正常に実行されたことを確認できます。Windows ベースのインスタンスが RDP 接続を受け入れるようになるまで最大で 15 分かかります。準備ができたら、ブラウザベースの RDP クライアントを使用してログインするか、独自の RDP クライアントを設定します。詳細については、「Windows ベースのインスタンスに接続する」を参照してください。

1. Lightsail インスタンスに接続できるようになったら、コマンドプロントを開きます (または Windows エクスプローラーを開きます)。
2. 次のように入力して Log ディレクトリに移動します。

```
cd C:\ProgramData\Amazon\EC2-Windows\Launch\Log
```

3. テキストエディターで UserdataExecution.log を開くか、type UserdataExecution.log と入力します。

   ログファイルには次のように表示されます。

```
2017/10/11 20:32:12Z: <powershell> tag was provided.. running powershell content
2017/10/11 20:32:13Z: Message: The output from user scripts: iex ((New-Object
 System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))

2017/10/11 20:32:13Z: Userdata execution done
```

Lightsail で Windows Server インスタンスを保護する

この記事では、Windows Server を実行する Lightsail インスタンスを使用する際にセキュリティリスクを回避するために役立つヒントやテクニックについて説明します。

Lightsail のパスワードについて

Windows Server ベースのインスタンスを作成するとき、Lightsail は推測しにくい長いパスワードをランダムに生成します。新しいインスタンスではこのパスワードを一意に使用します。デフォルトのパスワードを使用すると、リモートデスクトップ (RDP) を使用してインスタンスにすばやく接続できます。 インスタンスには常に AdministratorLightsail として ログインします。

パスワードの管理

Windows Server ベースのインスタンスのパスワードを覚えやすいものに変更できます。これは、リモートデスクトップクライアントを使用して Lightsail インスタンスにアクセスする場合に役立ちます。生成したパスワードが Lightsail に保存されることはありません。

> **Note**
>
> Lightsail では、ブラウザベースの RDP クライアントに、Lightsail で生成されたパスワードまたは独自のカスタムパスワードを使用できます。カスタムパスワードを使用する場合、ログインするたびにパスワードの入力を求められます。インスタンスにすばやくアクセスする場合、ブラウザベースの RDP クライアントで Lightsail によって生成されたデフォルトパスワードを使用した方が簡単です。

管理者パスワードを安全に変更するには、Windows Server のパスワードマネージャーを使用します。`Ctrl + Alt + Del` を押し、[パスワードの変更] を選択します。Lightsail にはパスワードが保存されないため、必ずパスワードを記録しておいてください。パスワードを取得する必要がある場合は、以下の「Windows ベースのインスタンスの管理者パスワードを変更する」を参照してください。

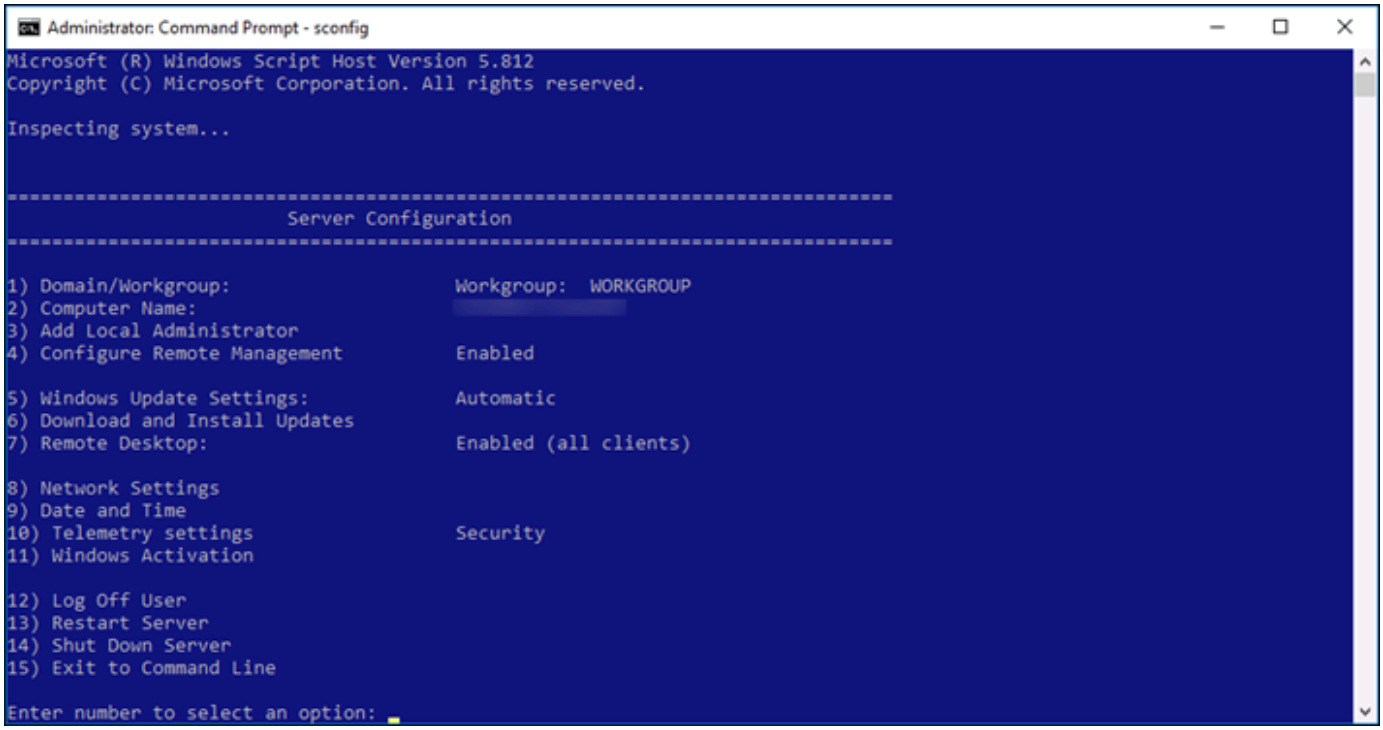
パスワードを一意のデフォルトパスワードから変更する場合、必ず強力なパスワードを使用してください。名前や辞書に載っている単語をベースとしたパスワードや、一連の文字の繰り返しは避けてください。

セキュリティパッチ

Windows Server ベースの Lightsail インスタンスを最新のセキュリティパッチで更新された状態に保つことをお勧めします。サーバーが更新をダウンロードおよびインストールするよう設定されていることを確認してください。次の手順では、Windows Server を実行する Lightsail インスタンスで直接これを行う方法について説明します。

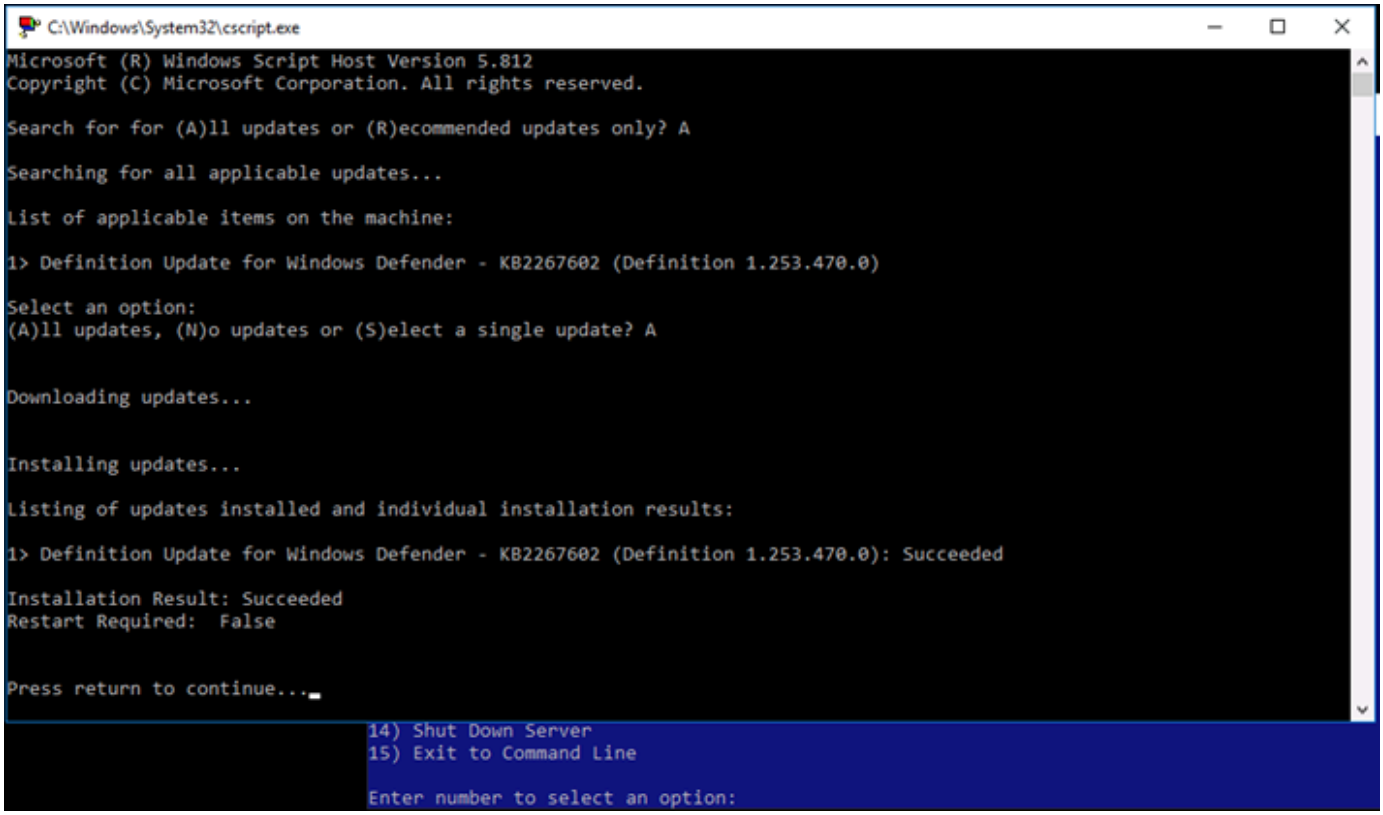
1. Windows Server ベースのインスタンスで、コマンドプロンプトを開きます。
2. 「`sconfig`」と入力し、`Enter` を押します。

   Windows Update Settings (5 番) はデフォルトで `Automatic` になっています。



3. 新しい更新プログラムをダウンロードしてインストールするには、6 と入力して Enter キーを押します。
4. 新しいコマンドウィンドウで「A」と入力して [(A)ll updates (すべての更新)] を検索し、Enter キーを押します。
5. もう一度「A」と入力して [(A)ll updates (すべての更新)] をインストールし、Enter キーを押します。

   完了したら、インストール結果と詳しい手順が記載されたメッセージが表示されます (該当する場合)。



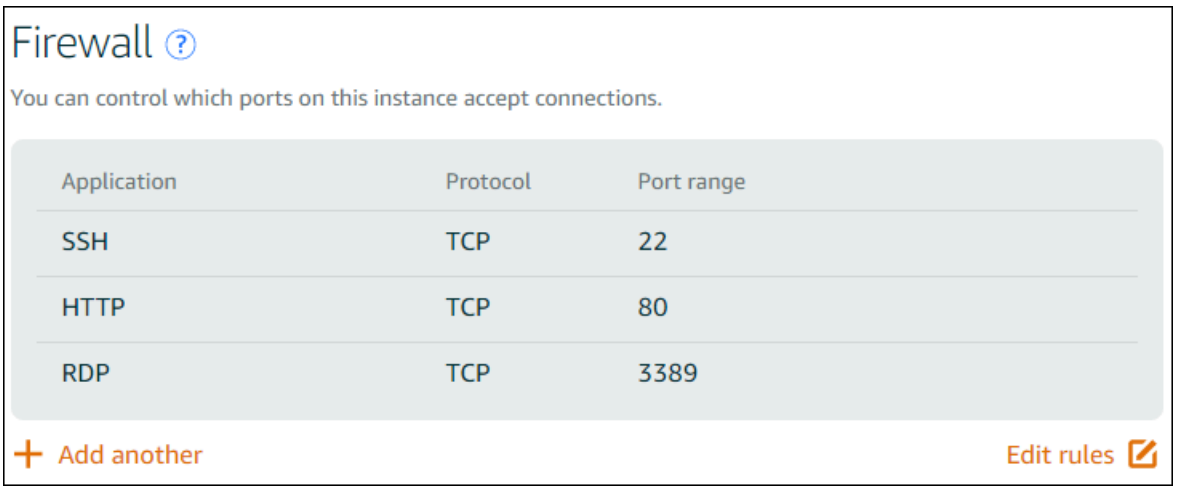
Windows Server でアカウントロックアウトポリシーを有効にする

ログイン試行に一定回数失敗した場合にアカウントを一時的または永続的に無効にするよう Windows Server を設定できます。たとえば、間違ったパスワードを 3 つ使用してインスタンスにログインしようとした場合にロックアウトすることができます。

詳細については、『Windows Server documentation』の「Account Lockout Policy」を参照してください。

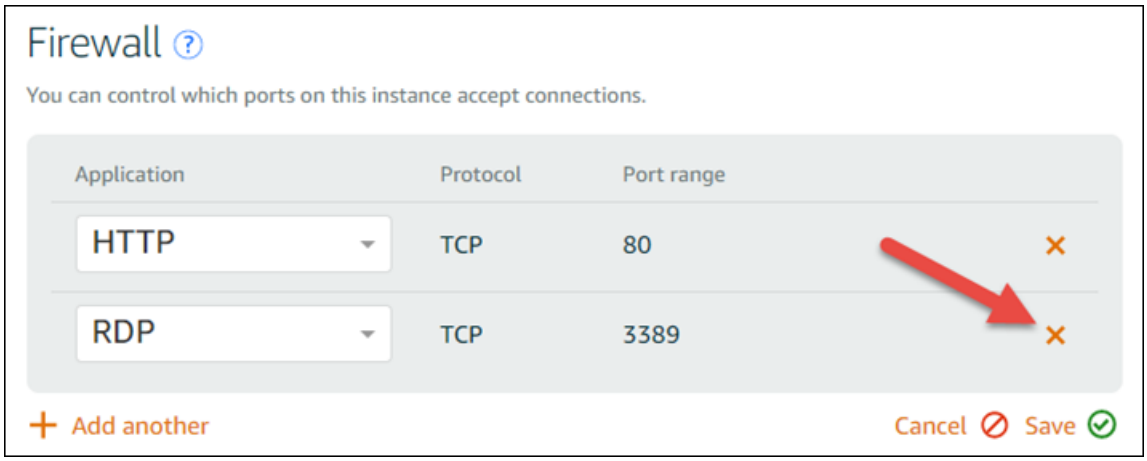
ポートとファイアウォールの設定

デフォルトでは、Windows Server ベースのインスタンスで次のポートを開きます。



有効にしたポートは世界中に公開され、ソース IP によって制限することはできません。インスタンスへのアクセスを制限するには、これらのポートを無効にし、インスタンスへのアクセスが必要なときにのみ有効にすることができます。その方法は次のとおりです。

1. Lightsail で管理するインスタンスを検索し、[管理] を選択します。
2. [ネットワーキング] を選択します。
3. インスタンスの [ネットワーキング] ページで、[ルールの編集] を選択します。
4. ルールの横にあるオレンジ色の [x] を選択することで、RDP/TCP/3389 ルールを削除します。



5. [保存] を選択します。

段階的な指示に従って、インスタンスの状態を制御し、スタックしているインスタンスを強制停止し、拡張ネットワーキングのためにインスタンスを更新し、Windows Server インスタンスのファイルシステムを拡張し、スクリプトを使用して起動時にインスタンスを設定し、Windows Server インスタンスを保護する方法について説明します。

このガイドでは、Linux または Unix と Windows Server インスタンスの両方を対象とし、ソフトウェアのインストール、設定の更新、パスワード管理、セキュリティパッチの有効化、ファイアウォール設定の構成といった作業に関するヒントやベストプラクティスを提供します。このガイドに従うことで、Lightsail インスタンスを効果的に管理および保護し、特定のユースケースに最適なパフォーマンス、セキュリティ、カスタマイズを確保できます。

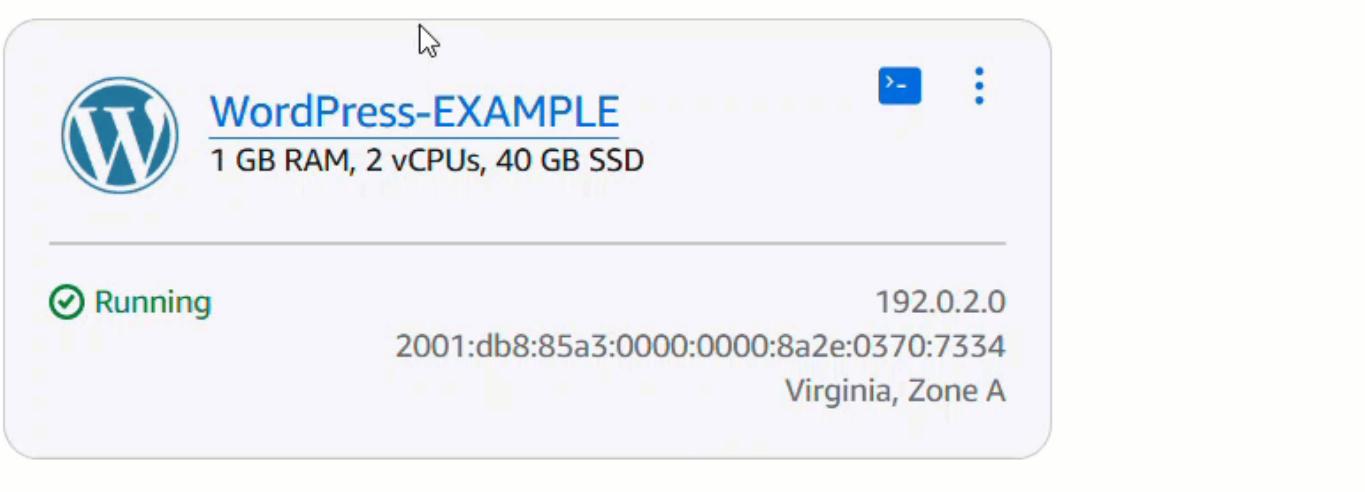
Lightsail インスタンスを削除する

インスタンスが不要になった場合は、 Amazon Lightsailコンソールまたは AWS Command Line Interface () を使用して削除できますAWS CLI。インスタンスを削除すると、インスタンスに対する課金も停止します。ただし、削除したインスタンスにアタッチされていたリソースに対しては、削除するまで料金が発生します。これらのリソースの詳細と、インスタンスを削除した後にリソースを削除する方法については、次の手順 を参照してください。

> **Warning**
>
> インスタンスを削除すると、復元できません。インスタンスの自動スナップショットも、このオペレーションの一部として削除されます。後で使用するためにデータを保持する場合は、まずインスタンスのスナップショットを作成するか、既存の自動スナップショットを保持することを選択する必要があります。詳細については、次のドキュメントを参照してください。
>
> - Lightsail で自動スナップショットが置き換えられないようにする
> - スナップショットを使用して Linux/Unix Lightsail インスタンスをバックアップする
> - Lightsail Windows Server インスタンスのスナップショットを作成する

Lightsail コンソールのホームページでインスタンスを削除する

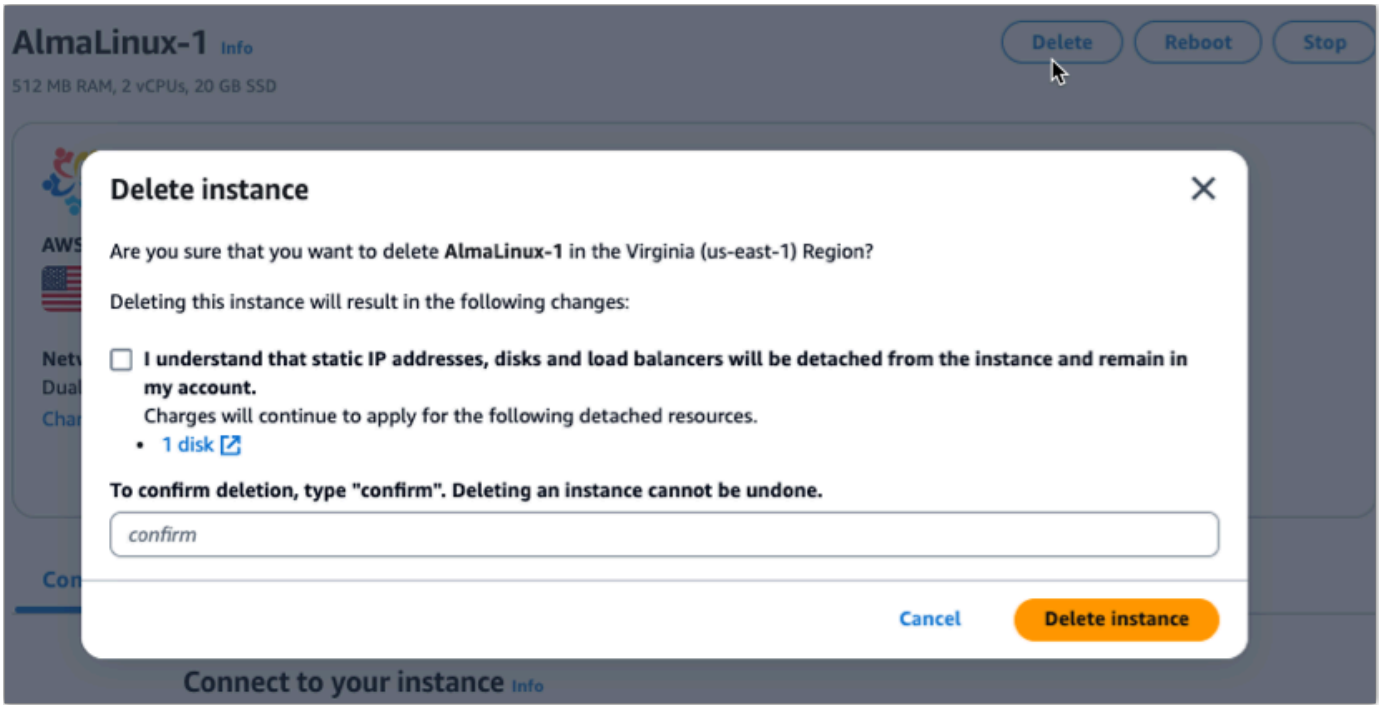
1. Lightsail コンソール にサインインします。
2. 削除するインスタンスのアクションメニューアイコン (⋮) を選択し、[削除] を選択します。



3. [はい、削除します] を選択して削除を確定します。

Lightsail コンソールのインスタンス管理ページでインスタンスを削除する

1. Lightsail コンソールのホームページで、削除するインスタンスを選択します。
2. [削除]、[インスタンスの削除] の順に選択します。



3. チェックボックスを選択し、入力フィールドに [確認] と入力して、インスタンスを削除することを確認します。
4. 削除を確認するには、[インスタンスを削除] を選択します。

を使用してインスタンスを削除する AWS CLI

1. 以下の前提条件を完了します (まだ完了していない場合)。

   a. をインストールします AWS CLI。詳細については、「AWS CLIをインストールする」を参照してください。

   b. AWS CLIを設定します。詳細については、「Configuring the AWS CLI」を参照してください。

   c. (オプション) を使用します AWS CloudShell。詳細については、「???」を参照してください。

2. ターミナル、コマンドプロンプト、または CloudShell ウィンドウを開き、次のコマンドを入力して、削除するインスタンスの名前を取得します:

   ```
   aws lightsail get-instances
   ```

   次のような結果が表示されます。

3. 削除するインスタンスの名前を選択してコピーします。この名前は次のステップで使用します。

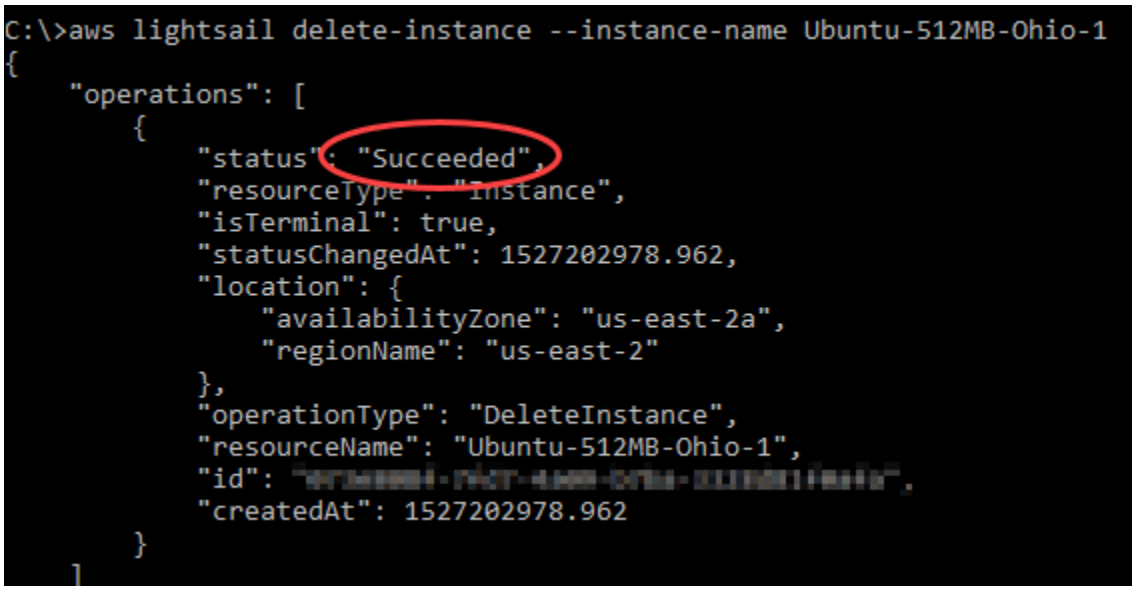
> **Note**
>
> 削除するインスタンスが表示されない場合は、 AWS CLI がインスタンス AWS リージョン がある 用に設定されていることを確認します。詳細については、「Configuring the AWS CLI」を参照してください。

4. 以下のコマンドを入力して、インスタンスを削除します。

```
aws lightsail delete-instance --instance-name InstanceName
```

コマンドで、*InstanceName* をインスタンスの名前に置き換えます。

削除が成功した場合は、次のような確認メッセージが表示されます。



> **Note**
>
> 削除が失敗した場合は、エラーメッセージが表示されます。インスタンス名を正確にコピーして貼り付けたことを確認し、もう一度試します。

次の手順

インスタンスの削除後に、インスタンスに関連付けられていた静的 IP、スナップショット、ブロックストレージディスク、およびロードバランサーは Lightsail に残るため、引き続き料金が発生します。これらのリソースの削除方法については、以下の記事を参照してください。

- 静的 IP の削除
- スナップショットを削除する
- ブロックストレージディスクをデタッチおよび削除する
- ロードバランサーの削除

SSH キーペアを管理し、Lightsail インスタンスに接続する

キーペアは、Amazon Lightsail インスタンスへの接続時にユーザーのアイデンティティを証明するために使用する一連のセキュリティ認証情報です。キーペアは公開キーとプライベートキーで構成されており、公開キーは Lightsail がインスタンスに保存し、プライベートキーはユーザーが保存します。

キーペアファイルには、以下のテキストが含まれています。

Public key example:

```
ssh-rsa
EXAMPLEzaC1yc2EAAAADAQABAAABAQCoYFOS10yNQ2AoRuvt2uM2LpuZXLGpNoHFxCAmXZjNIZ6t6s
sHCAWgiqzbp5fzRSZnPXjeuxQo2KsGkZCD6f81YHfEIBTSPWoiA6HPWAlAOR6K7E4ZGBkpYhOJKDK1
BYzCKUTgyRUvemmNmGme/c5O4ts50se0A/8m26YNt8TYgKqLV7mjl+Q1uMix0qS3wOim4x
+Iq5eV3cdTa0v0iuQJd01aXoCdJ1cdMW6qEDxZ5ILEMtle8FoLvvMe67JLqjCTxy8i/6x
+SiBWVITOgBKfeePPHsq2PceOQN/XfajeLd+CMAXYyRrvUo4HIiR443BJG1zevIvKYA7+yEXAMPLE
```

Private key example:

```
-----BEGIN RSA PRIVATE KEY-----
EXAMPLEBAAKCAQEAqGBTktdMjUNgKEbr7drjNi6bmVyxqTaBxcQgJl2YzSGererL
BwgFoIqs26eX80UmZz143rsUKNirBpGQg+n/JWB3xCAU0j1qIgOhz1gJQDkeiuxO
GRgZKWITiSgypQWMwilE4MkVL3ppjZhpnv3OTuLbOdLHtAP/JtumDbfE2ICqi1e5
o5fkNbjIsdKkt8DopuMfiKuXld3HU2tL9IrkCXdNWl6AnSdXHTFuqhA8WeSCxDLZ
XvBaC77zHuuyS6owk8cvIv+sfkogVlSEzoASn3njzx7Ktj3HjkDf132o3i3fgjAF
2Mka71KOByIkeONwSRtc3ryLymAO/smHHNQRzwIDAQABAoIBAGoipiu2uVOGd/OL
mSaKxpSd1olaq8atTCo8kcN9V1df7OVWTnp1LQ7gu0uOnjLDkQyc7DcCGBgTU+NF
GKJ+es21vGkNi/JmsiMUxQetR8+K8dzCTgx1a07xurzHcP0ivXKajwde2ZLfB/Aw
dcu50zVYvLX7TtUDe++jn02gXF3X3q98lqWmSPV+dtlZPctQqcmemjQg3onUdpZo
4yrAKUKJdrchIMHhBD0jisom86Zl3jEPXRY7iuOfa1bB76cmErja18rijUhMH5Pn
mjAsbvZ0CTxU7QGx5yHnFtSK73oLN4LoYKek0TA7JARc4lp0MELtkOTn9mj2IeEw
h2yygPECgYEA37mi3uGVBBAVLEU3Z2sAS/thF0+L2y6qcuBxjY/HeyPnwvuXied0
xJhb9wPpODRTShDkKLfHPiVYD7H6bXZLetZfNLjIu/IKvseL85zCX8fWz6cJ6IeS
3QKRYu2VdpQW2prs+58QyKDlDqQ0hfE3dHZvSayLmm/9/sBZ24+G/WcCgYEAwKqb
yYkDOZtXIHZyTt1UUHvKFzo9LFuuMwlHQdNpvy2QbNNw4iE706DzVjy9FNuMXzIs
Skhhn7m+wredBP+r8udX3+gA1vY329wJ/+c7W8IPN21RiWIT4VtawmoHgMeJHOv4
4mdxqMo6L44Nkny/4KLtGAuZCUrJzoLr+d+Fn1kCgYEAyA7MIdo+0r8+77OFc6kv
PsKvc5TiT0FPkiI56IilrOvSl307aUncFODZe+23Y1cHE7g/DloohN4H/SD9+1xI
6rM/t3l1pvstuKPf9hw7hELDSDTqm1CAd7mQIJKrkLmkJh9bwzXeYEngC1Oz1AJ7
wF0X7x2oSJXU3zVKJRgXcgkCgYEAn504DxC5YUI2Piiirn9iWIMVe4S+JT+W46Uu
KXSSSNXgrqfE/zH1NHBE6A7NvrfcZQlV8/xfFEp3pS0kon2F4GiUPmUgPPYidLyo
dB8G6A+vN4YTZLOiMLLUT/gzWxbzmshLmpWEbgeLiNYwnElJVTrlHWSOVkplQfbo
tEvfkZECgYAayAwDXa2gbZBmqInwCTNJyqu8XW/Kc4JBT6mugXzQqxMr6ZnXM70h
Fq0EAT7kAHt4wKfZyPkcgrrmj0Mej6VoL2GlJejPykNa20nxrPIi8ecJDYhjiaIp
zoO5rFDVcZhMctewa70OL3c1q+nDGf7Sd9pqw0q31K6MiJwEXAMPLE==
-----END RSA PRIVATE KEY-----
```

Linux および Unix インスタンスでは、プライベートキーを使用することでインスタンスへのセキュアな SSH 接続を確立できます。Windows インスタンスでは、インスタンスへのセキュアな RDP 接続を確立するために使用されるデフォルトの管理者パスワードを、プライベートキーが復号化します。

プライベートキーにアクセスできるユーザーならば誰でもインスタンスに接続できるため、プライベートキーは安全な場所に保存することが重要です。

目次

- キーペアオプションの選択
- インスタンスへの接続
- インスタンスに保存されているキーの管理

キーペアオプションの選択

Lightsail インスタンスの作成時に、以下のキーペアオプションのいずれかを選択できます。Windows インスタンスは常にデフォルトキーを使用します。このため、Windows インスタンスの作成時にキーペアを作成したり、キーをアップロードしたりすることはできません。

- デフォルトの SSH キー – インスタンスを作成する各 AWS リージョン にデフォルトのキーペア Lightsailが自動的に作成されます。インスタンスでデフォルトキーペアを使用すると、Lightsail がインスタンスに公開キーを保存します。デフォルトキーペアのプライベートキーは、Lightsail コンソールの [Account] (アカウント) ページからいつでもダウンロードできます。それぞれに最大 1 つのデフォルトキーペアを設定できます AWS リージョン。
- カスタムキーを作成する (Linux および Unix インスタンス) – Lightsail コンソールを使用して、インスタンスで使用する新しいカスタムキーペアを作成できます。カスタムキーペアを作成するときは、一意の名前を付けてください。公開キーは、Lightsail がインスタンスに保存します。カスタムキーペアのプライベートキーをダウンロードできるのは、最初の作成時のみです。
- キーをアップロードする (Linux および Unix インスタンス) – 独自の既存キーペアを使用するには、公開キーを Lightsail にアップロードします。インスタンスでの使用のために公開キーをアップロードするときは、一意の名前を付けてください。このキーは、Lightsail がインスタンスに保存します。キーペアのプライベートキーは、ユーザーが保持して保存します。

複数のインスタンスに単一の公開キーを設定する場合、これらのインスタンスへの接続には同じキーペアのプライベートキーを使用できます。キーペアの管理に関する詳細については、「Amazon Lightsail でのキーペアの管理」を参照してください。

インスタンスに接続します

Lightsail インスタンスには、以下のオプションのいずれかを使用して接続することができます。

Lightsail のブラウザベースの SSH および RDP クライアント

Lightsail コンソールでは、ブラウザベースの SSH クライアントを使用して Linux および Unix インスタンスに瞬時に接続し、ブラウザベースの RDP クライアントを使用して Windows インスタンスに接続することができます。ブラウザベースのクライアントを使用してインスタンスに接続するときは、コンピュータに SSH クライアントをインストール、キーペアを設定、または管理者パスワードを指定する必要はありません。これは、インスタンスに接続するための最も迅速な方法です。詳細については、「Amazon Lightsail での Linux または Unix インスタンスへの接続」または「Amazon Lightsail での Windows インスタンスへの接続」を参照してください。

ブラウザベースのクライアントは、インスタンスの作成時に設定するキーペア (デフォルトキーや、ユーザーが作成またはアップロードするキーなど) とは異なるキーペアを使用します。このため、当初設定したキーのいずれかを削除したり紛失したりした場合でも、ブラウザベースのクライアントを使用してインスタンスへの接続を継続することができます。

サードパーティーの SSH および RDP クライアント

サードパーティーの SSH クライアントを使用して Linux および Unix インスタンスに接続し、サードパーティーの RDP クライアントを使用して Windows インスタンスに接続することができます。SSH クライアントを使用するときは、インスタンスで設定したキーペアのプライベートキーを使用するようにクライアントを設定する必要があります。RDP クライアントを使用するときは、Windows インスタンスの管理者パスワードを指定する必要があります。

Windows コンピュータをローカルで使用する場合は、以下のクライアントを使用して Lightsail インスタンスに接続できます。

- PuTTY – Use PuTTY SSH を使用した Linux または Unix インスタンスへの接続には、PuTTY を使用します。詳細については、「PuTTY を設定してインスタンスに接続する」を参照してください。
- リモートデスクトップ接続 – RDP を使用した Windows インスタンスへの接続には、リモートデスクトップ接続クライアントを使用します。詳細については、「Windows コンピュータでリモートデスクトップ接続クライアントを使用して Windows インスタンスに接続する」を参照してください。

Mac コンピュータをローカルで使用する場合は、以下のクライアントを使用して Lightsail インスタンスに接続します。

- ターミナル内のネイティブ SSH クライアント – Linux および Unix インスタンスへの接続には、ターミナル内のネイティブ SSH クライアントを使用します。詳細については、「ターミナルで SSH を使用して Linux または Unix インスタンスに接続する」を参照してください。
- Microsoft リモートデスクトップ – RDP を使用した Windows インスタンスへの接続には、macOS 用の Microsoft リモートデスクトップクライアントを使用します。詳細については、「Mac で Microsoft リモートデスクトップクライアントを使用して Windows インスタンスへ接続する」を参照してください。

インスタンスに保存されているキーの管理

インスタンスが実行状態になったら、インスタンスに新しいキーを追加したり、最初に割り当てたキーを交換したりすることができます。例えば、組織内のユーザーが個別のキーを使用してインスタンスにアクセスする必要がある場合は、そのキーをインスタンスに追加できます。もう 1 つの例として、誰かが組織から脱退し、その人物がプライベートキー (.PEM) ファイルのコピーを持っているという場合があります。そのキーを新しいキーと交換する、または完全に削除することによって、この人物がインスタンスに接続できないようにすることが可能です。詳細については、「Amazon Lightsail 内のインスタンスに保存されているキーの管理」を参照してください。

トピック

- Lightsail の SSH キーの設定
- Lightsail SSH キーによる安全なインスタンス接続の制御
- Lightsail Linux インスタンスでの SSH キーを管理する
- Lightsail で Linux または Unix のインスタンスに接続する
- RDP を使用した Lightsail Windows インスタンスへの接続

Lightsail の SSH キーの設定

Secure SHell (SSH) は、仮想プライベートサーバー (または Lightsail インスタンス) に安全に接続するためのプロトコルです。SSH は、リモートサーバーを承認されたユーザーに一致させるパブリックキーとプライベートキーを作成することによって動作します。このキーペアを使用し、ブラウザベースの SSH ターミナルを使用して Lightsail インスタンスに接続できます。

SSH の詳細については、「SSH について」を参照してください。

Lightsail インスタンスの作成時のデフォルトオプションでは、Lightsail で SSH キーが自動的に管理されます。Lightsail は、Linux ベースのインスタンスに安全に接続するためのブラウザベースの SSH クライアントを提供します。このクライアントは完全に機能するターミナルであり、そこでコマンドを入力したりインスタンスへの変更を行ったりできます。

Windows ベースのインスタンスでは、SSH の代わりにリモートデスクトップ (RDP) プロトコルを使用します。Lightsail における Windows ベースのインスタンスの詳細については、「Lightsail で Windows ベースのインスタンスの使用を開始する」を参照してください。

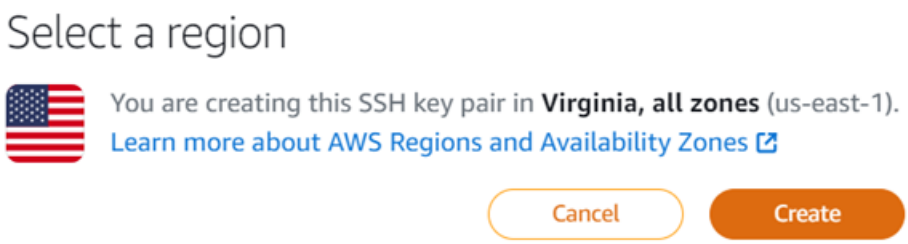
> **Important**
>
> SSH キーはリージョンごとに管理されます。新しい AWS リージョン でインスタンスを作成する際に、そのリージョンのデフォルトのキーペアを使用するオプションが提示されます。そのリージョンでカスタムキーを使用することもできます。独自のキーをアップロードする場合は、Lightsail インスタンスがあるリージョンごとにアップロードする必要があることに注意してください。

デフォルトのキーを使用している場合でも、保管用にプライベートキーをダウンロードできます。キーのダウンロードは、インスタンスの作成時または作成後に行うことができます。インスタンスを作成した後にキーをダウンロードすることを選択した場合、[アカウント] ページの [SSH キー] の下でダウンロードできます。

新規キーの作成

デフォルトのキーを使用することを選択しない場合は、Lightsail インスタンスの作成時に新規キーペアを作成できます。

1. まだ作成していない場合は [インスタンスの作成] を選択します。
2. [インスタンスを作成] ページで、[カスタムキーを作成] を選択します。
3. Lightsail新規キーを作成しているリージョンが によって表示されます。



[作成] を選択します。

4. キーペアの名前を入力します。

   リソース名:

   - Lightsail アカウントの各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。

- 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

5. [キーペアの生成] を選択します。

> **Important**
>
> 見つけやすい場所にキーを保存します。また、他のユーザーがそのキーを読み取りできないようにアクセス許可を設定することをお勧めします。

6. インスタンスの作成を続行します。

既存のキーのアップロード

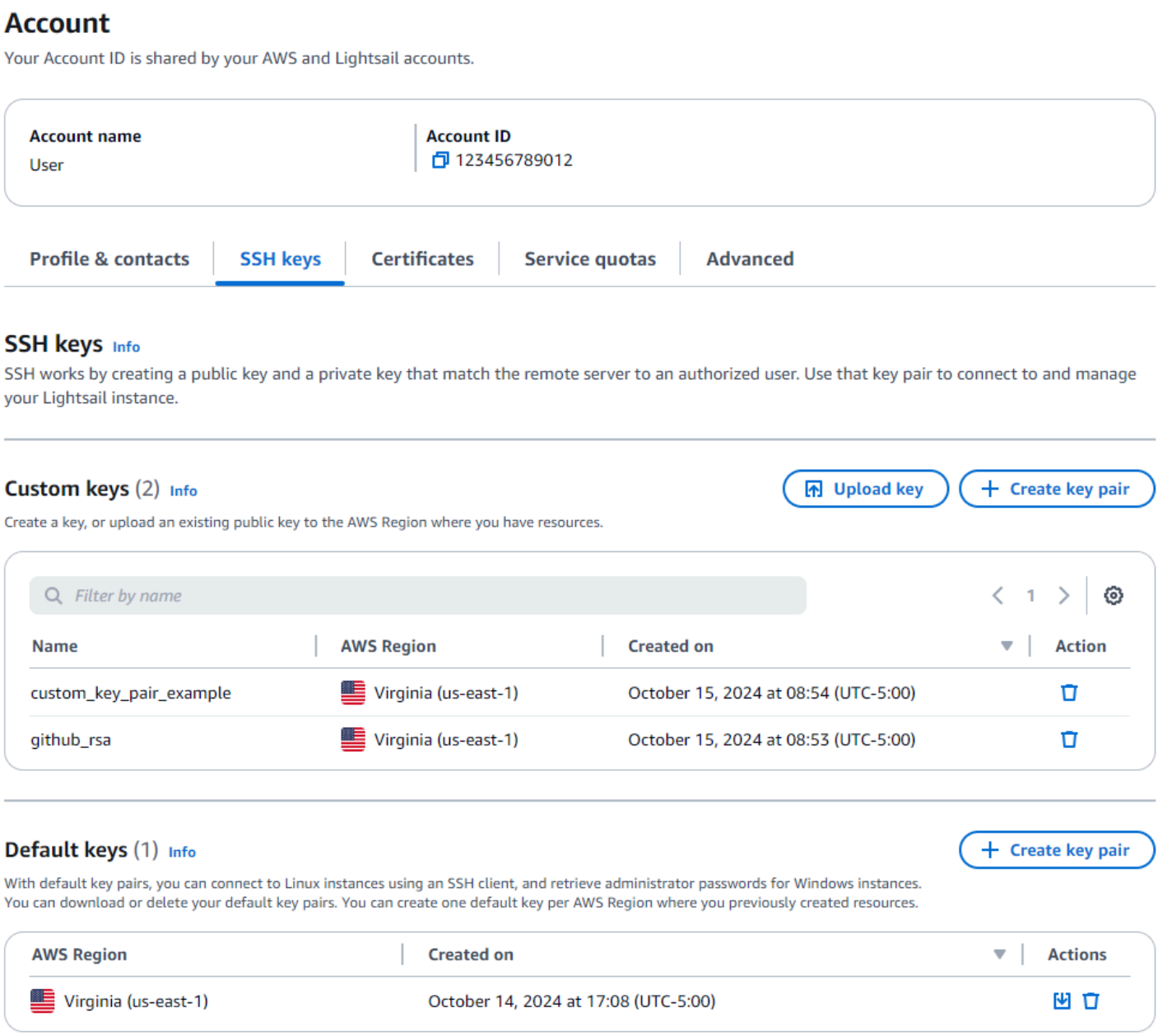
Lightsail インスタンスの作成時に、既存のキーをアップロードすることもできます。

1. まだ作成していない場合は [インスタンスの作成] を選択します。
2. [インスタンスを作成] ページで [キーのアップロード] を選択します。
3. [アップロード] を選択します。
4. Lightsail新規キーをアップロードしているリージョンが によって表示されます。
5. [ファイルを選択] を選択して、ローカルマシンでキーを見つけます。

   プライベートキーではなくパブリックキーをアップロードしていることを確認します。例えば、`github_rsa.pub`。
6. [Upload key] (キーのアップロード) をクリックします。
7. インスタンスの作成を続行します。

キーを管理する

キーは、[アカウント] ページの [SSH キー] タブで管理できます。各リージョンで使用中の各キーペアが表示されます。



このページでは、新しいキーの作成、既存のキーの削除、既存のキーのアップロード、プライベートキーのダウンロードを行うことができます。PuTTY と同様に SSH クライアントを使用して接続できますが、プライベートキーを持っている必要があります。プライベートキーは [アカウント] ページでダウンロードできます。Lightsail インスタンスに接続するように PuTTY をセットアップする方法の詳細を確認してください。

Lightsail SSH キーによる安全なインスタンス接続の制御

キーペアを使用することで、Amazon Lightsail インスタンスへのセキュアな接続を確立することができます。Amazon Lightsail インスタンスを新規作成するときに、Lightsail が作成するキーペア (Lightsail デフォルトキーペア) の使用、またはユーザーが作成するカスタムキーペアの使用を選択できます。詳細については、「キーペアと Amazon Lightsail 内のインスタンスへの接続」を参照してください。

Linux および Unix インスタンスでは、プライベートキーを使用することでインスタンスへのセキュアな SSH 接続を確立できます。Windows インスタンスでは、インスタンスへのセキュアな RDP 接続を確立するために使用されるデフォルトの管理者パスワードを、プライベートキーが復号化します。

このガイドでは、Lightsail インスタンスで使用できるキーの管理方法について説明します。キーの表示、既存キーの削除、および新しいキーの作成やアップロードを実行できます。

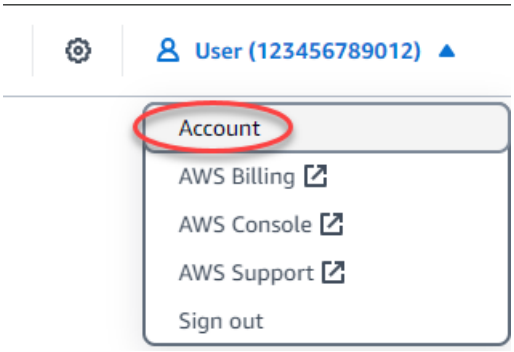
目次

- デフォルトキーとカスタムキーを表示する
- Lightsail コンソールからデフォルトキーのプライベートキーをダウンロードする
- Lightsail コンソールでカスタムキーを削除する
- Lightsail コンソールでデフォルトキーを削除して新しいキーを作成する
- Lightsail コンソールを使用してカスタムキーを作成する
- ssh-keygen を使用してカスタムキーを作成し、Lightsail にアップロードする

デフォルトキーとカスタムキーを表示する

Lightsail コンソールでデフォルトキーとカスタムキーを表示するには、以下の手順を実行します。

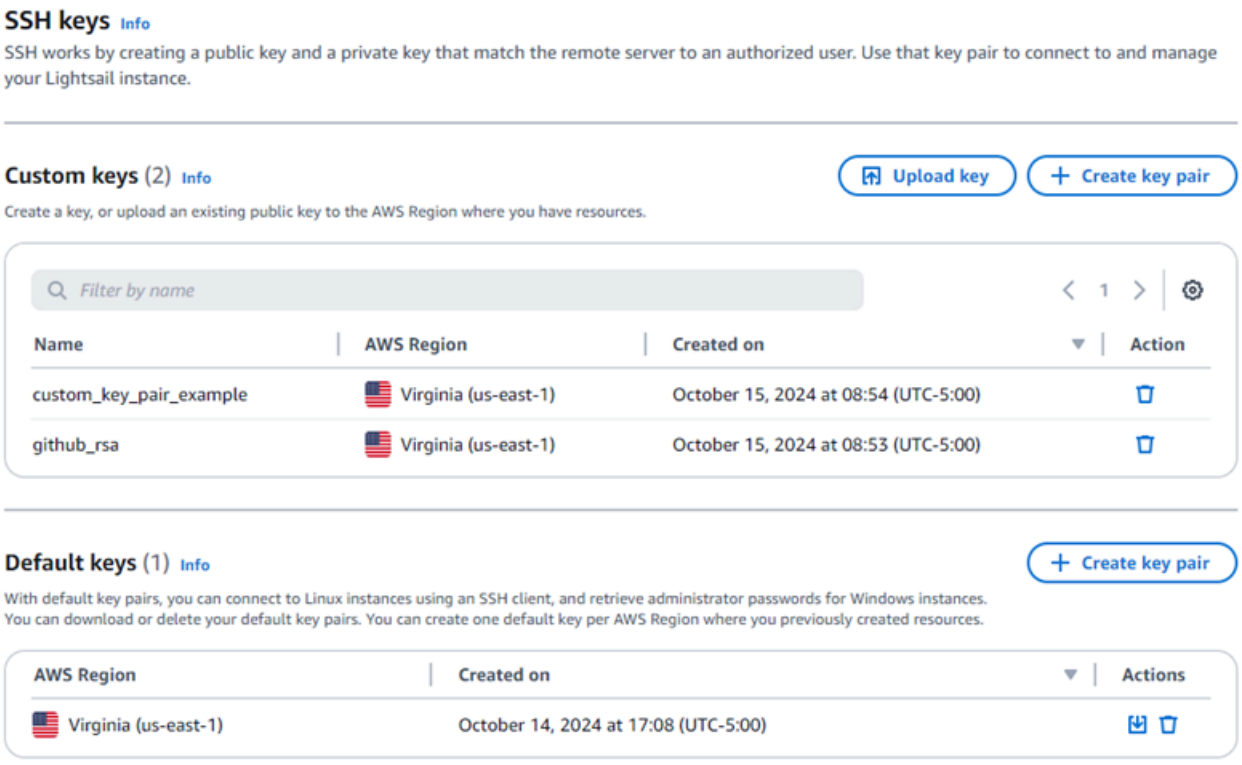
1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
3. ドロップダウンメニューで [Account (アカウント)] を選択します。



4. [SSH キー] タブを選択します。

   SSH キーページには、以下がリストされます。

   - カスタムキー – これらは、Lightsail コンソール、または ssh-keygen などのサードパーティーツールを使用してユーザーが作成するキーです。それぞれに多数のカスタムキーを持つことができます AWS リージョン。
   - デフォルトキー – これらは、Lightsail が作成するキーです。デフォルトキーは、 AWS リージョンごとに 1 つしか設定できません。



カスタムキーとデフォルトキーはリージョン別です。例えば、米国西部 (オレゴン) AWS リージョン内のキーを設定できるのは、そのリージョンで作成されたインスタンスだけです。キーの詳細については、「キーペアと Amazon Lightsail 内のインスタンスへの接続」を参照してください。

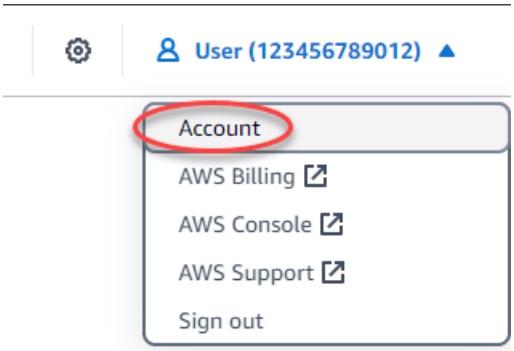
SSH キーページでは、キーペアの作成、キーのアップロード、キーの削除、および Lightsail デフォルトキーペアのプライベートキーのダウンロードを実行できます。

> **Note**
>
> Lightsail はカスタムキーペアのプライベートキーを保存しないため、プライベートキーをダウンロードすることはできません。カスタムキーペアのプライベートキーを紛失した場合は、新しいキーを作成して、それをインスタンスで設定する必要があります。その後、紛失したキーを削除します。詳細については、本ガイド後出の「Create a custom key using the Lightsail console」(Lightsail コンソールを使用してカスタムキーを作成する) または「Create a custom key using ssh-keygen and upload to 」(ssh-keygen を使用してカスタムキーを作成し、 にアップロードする) を参照してください。

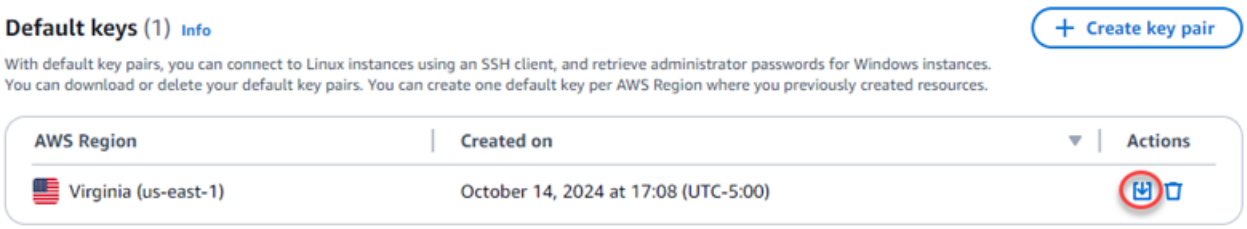
Lightsail コンソールからデフォルトキーのプライベートキーをダウンロードする

Lightsail コンソールからデフォルトキーペアのプライベートキーをダウンロードするには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
3. ドロップダウンメニューで [Account (アカウント)] を選択します。



4. [SSH キー] タブを選択します。
5. そのページの [Default keys] (デフォルトキー) セクションで、ダウンロードするキーのダウンロードアイコンを選択します。



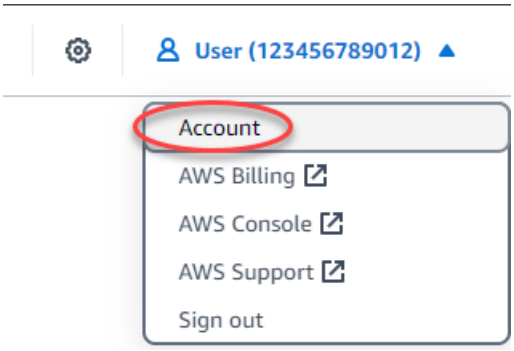
> **Important**
>
> プライベートキーは安全な場所に保存してください。このキーはインスタンスへの接続に使用できるため、公開しないでください。

SSH クライアントは、プライベートキーを使用してインスタンスに接続するように設定できます。詳細については、「インスタンスへの接続」を参照してください。

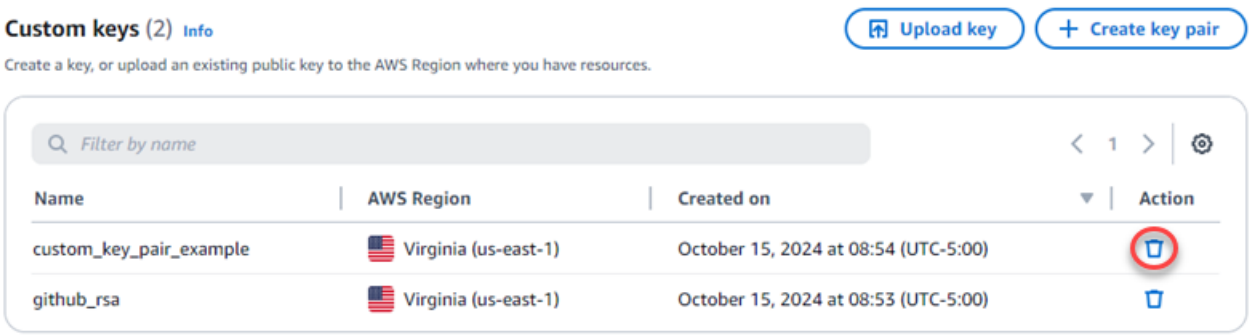
Lightsail コンソールでカスタムキーを削除する

Lightsail コンソールでカスタムキーを削除するには、以下の手順を実行します。これによって、Lightsail で作成される新しいインスタンスにカスタムキーが設定されないようになります。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
3. ドロップダウンメニューで [Account (アカウント)] を選択します。



4. [SSH キー] タブを選択します。
5. そのページの [Custom keys] (カスタムキー) セクションで、削除するキーの削除アイコンを選択します。

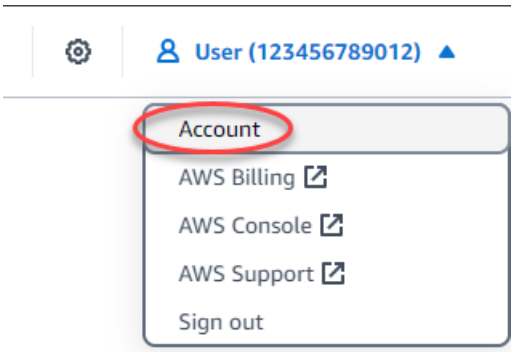


カスタムキーを削除しても、以前に作成された現在実行中のインスタンスからカスタムキーペアの公開キーが削除されることはありません。実行中のインスタンスに保存されている以前に設定された公開キーを削除するには、「Amazon Lightsail 内のインスタンスに保存されているキーの管理」を参照してください。

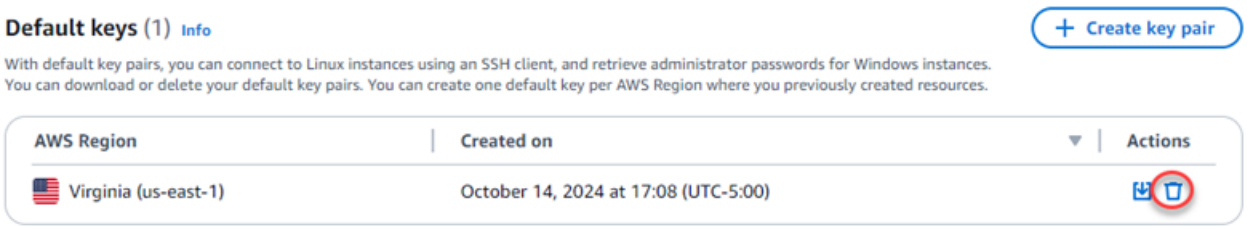
Lightsail コンソールでデフォルトキーを削除して新しいキーを作成する

Lightsail コンソールでデフォルトキーを削除するには、以下の手順を実行してください。削除することによって、Lightsail で作成される新しいインスタンスにデフォルトキーが設定されないようになります。削除後、削除したキーを置き換えるための新しいデフォルトキーを作成できます。Lightsail で作成する新しいインスタンスには、新しいデフォルトキーを設定することができます。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
3. ドロップダウンメニューで [Account (アカウント)] を選択します。

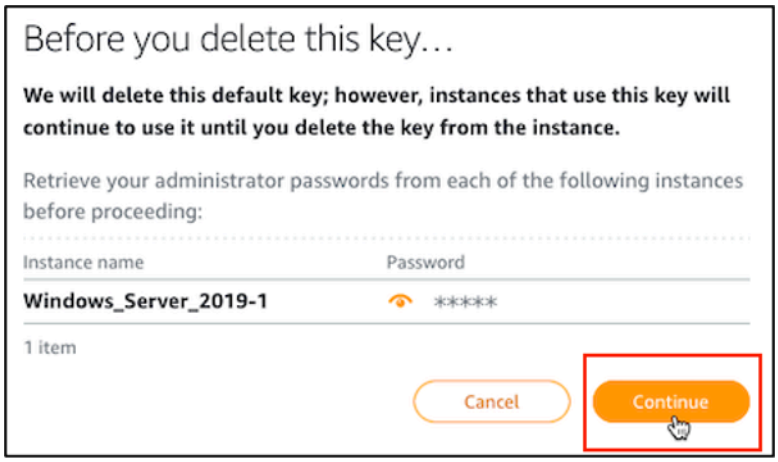


4. [SSH キー] タブを選択します。
5. そのページの [Default keys] (デフォルトキー) セクションで、削除するデフォルトキーの削除アイコンを選択します。

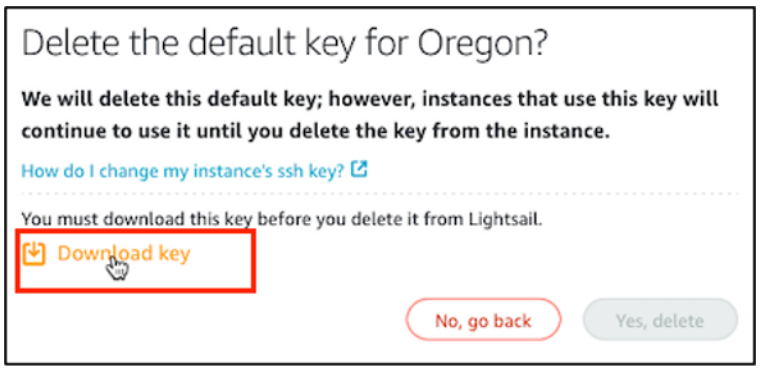


> **Important**
>
> デフォルトキーを削除しても、以前に作成された現在実行中のインスタンスからカスタムキーペアの公開キーが削除されることはありません。詳細については、「Amazon Lightsail 内のインスタンスに保存されているキーの管理」を参照してください。

6. デフォルトキーは、Windows インスタンスの管理者パスワードを生成するために使用されます。デフォルトキーを削除する前に、削除するデフォルトキーを使用するすべての Windows インスタンスから管理者パスワードを取得して保存する必要があります。
7. [Continue] (続行) を選択して、デフォルトキーを削除します。



8. デフォルトキーは、削除する前にダウンロードする必要があります。デフォルトキーをダウンロードしたら、[Yes, delete] (はい、削除します) を選択して、デフォルトキーを完全に削除することができるようになります。



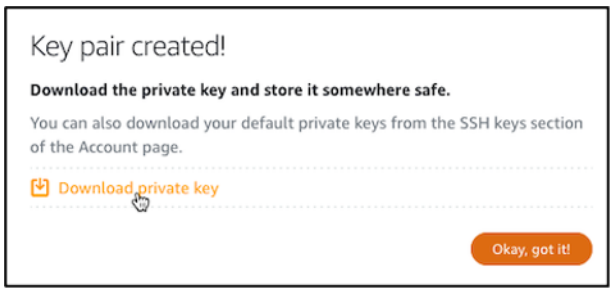
9. デフォルトキーが削除されました。[OK] を選択してください。

以下の手順はオプションで、削除したデフォルトキーペアを置き換える場合にのみ実行するようにしてください。

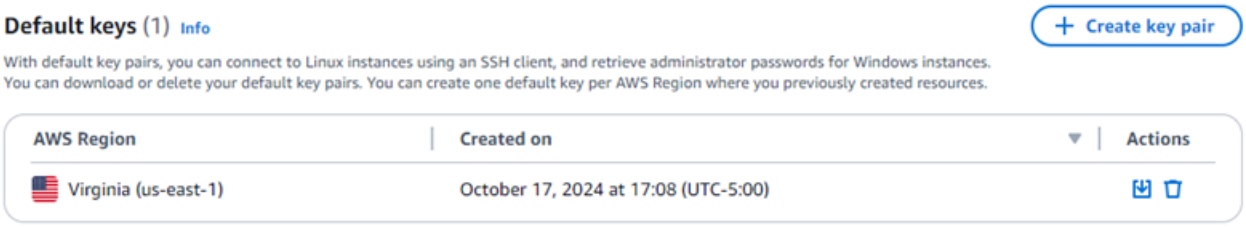
10. このページの [Default keys] (デフォルトキー) セクションで、[Create key pair] (キーペアを作成) を選択します。
11. 表示されるリージョンの選択プロンプト AWS リージョン で、新しいデフォルトキーを作成する を選択します。同じ AWS リージョン内の新しいインスタンスには、新しいデフォルトキーを設定することができます。

> **Note**
>
> これらのステップを使用して、Lightsailリソースを作成した AWS リージョンでのみデフォルトのキーペアを作成できます。新しいリージョンでデフォルトキーペアを作成するには、そのリージョンで Lightsail リソースを作成する必要があります。リソースを作成すると、デフォルトキーペアも作成されます。

12. プライベートキーをダウンロードして、安全な場所に保存します。
13. [Ok, got it!] (わかりました!) を選択して続行します。



14. Lightsail コンソールの SSH キーページで、新しいデフォルトキーを確認します。

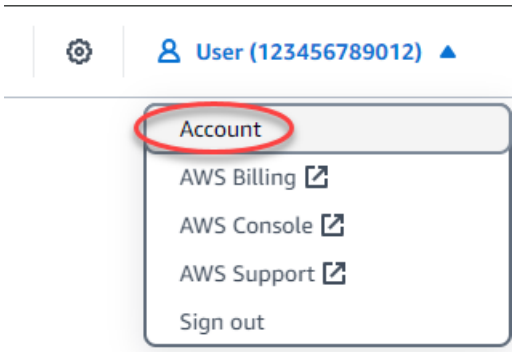


Lightsail で作成する新しいインスタンスには、新しいデフォルトキーを設定することができます。以前に作成されて、現在実行中のインスタンスで新しいデフォルトキーを設定するには、「Amazon Lightsail 内のインスタンスに保存されているキーを管理する」を参照してください。

Lightsail コンソールを使用してカスタムキーを作成する

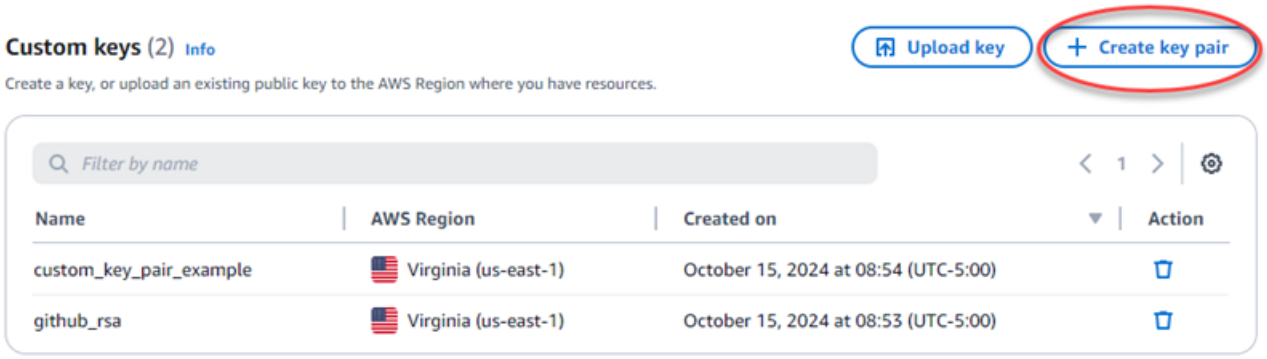
Lightsail コンソールを使用してカスタムキーペアを作成するには、以下の手順を実行します。Lightsail で作成する新しいインスタンスには、新しいカスタムキーを設定することができます。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
3. ドロップダウンメニューで [Account (アカウント)] を選択します。



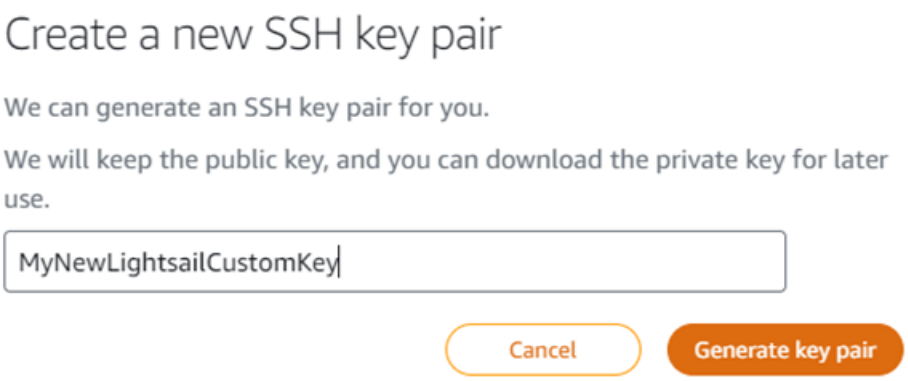
4. [SSH キー] タブを選択します。

5. そのページの [Custom keys] (カスタムキー) セクションで、[Create key pair] (キーペアを作成) をクリックします。



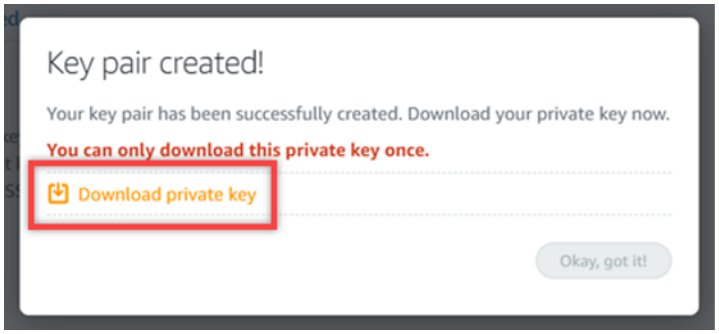
6. 表示される [Select a region] (リージョンの選択) プロンプトで、新しいカスタムキーを作成する AWS リージョン を選択します。同じ AWS リージョン内の新しいインスタンスには、新しいカスタムキーを設定することができます。

7. 表示される [Create a new SSH key pair] (新しい SSH キーペアの作成) プロンプトでカスタムキーに名前を付け、[Generate key pair] (キーペアの生成) を選択します。

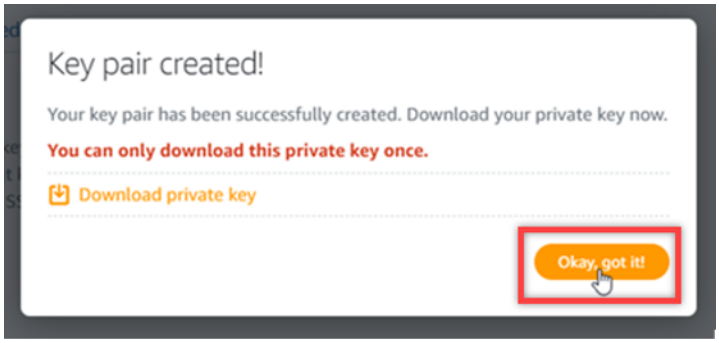


8. 表示される [Key pair created!] (キーペアが作成されました!) プロンプトで [Download private key] (プライベートキーのダウンロード) を選択して、プライベートキーをローカルコンピュータに保存します。

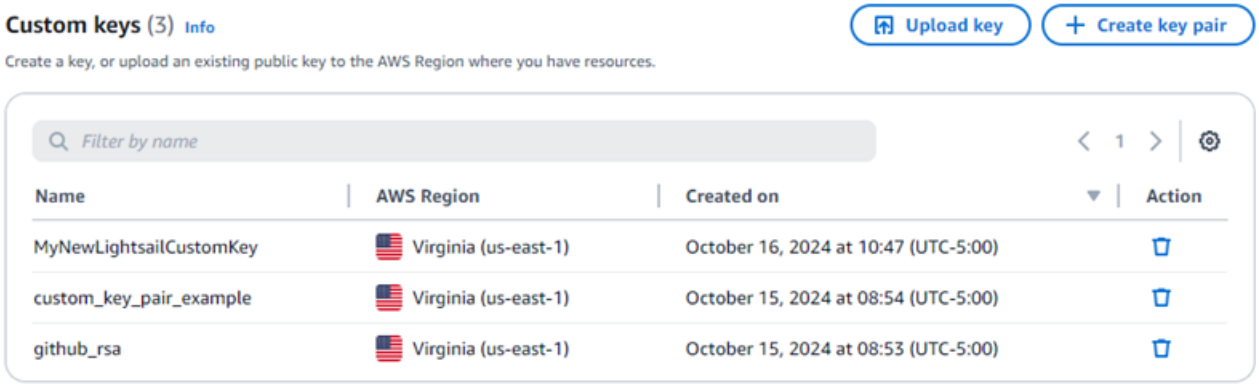
> **Important**
>
> プライベートキーは安全な場所に保存してください。このキーはインスタンスへの接続に使用できるため、公開しないでください。
> カスタムキーペアのプライベートキーをダウンロードできるのは、このときだけです。Lightsail は、カスタムキーペアのプライベートキーを保存しません。このプロンプトを閉じてしまうと、再度ダウンロードすることはできません。



9. [Ok, got it!] (わかりました!) を選択して、プロンプトを閉じます。



10. 新しいカスタムキーは、このページのカスタムキーセクションにリストされます。



Lightsail で作成する新しいインスタンスには、新しいカスタムキーを設定することができます。以前に作成された現在実行中のインスタンスで新しいカスタムキーを設定するには、「Amazon Lightsail 内のインスタンスに保存されているキーを管理する」を参照してください。

ssh-keygen を使用してカスタムキーを作成し、Lightsail にアップロードする

ssh-keygen などのサードパーティーツールを使用してローカルコンピュータでカスタムキーペアを作成するには、以下の手順を実行します。キーを作成したら、そのキーを Lightsail コンソールにアップロードできます。Lightsail で作成する新しいインスタンスには、新しいカスタムキーを設定することができます。

1. ローカルコンピュータで、コマンドプロンプトまたはターミナルを開きます。
2. 次のコマンドを入力して、新しいキーペアを作成します。

```
ssh-keygen -t rsa
```

3. キーペアを保存するコンピュータのディレクトリの場所を指定します。

   例えば、以下のディレクトリのいずれかを指定できます。

a. Windows の場合: C:\Users*<UserName>*\.ssh*<KeyPairName>*

b. macOS、Linux、または Unix の場合: /home/*<UserName>*/.ssh/*<KeyPairName>*

<UserName> を現在サインインしているユーザーの名前に置き換えて、*<KeyPairName>* を新しいキーペアの名前に置き換えます。

以下の例では、Windows コンピュータの C:\Keys ディレクトリを指定し、新しいキーに MyNewLightsailCustomKey という名前を付けました。

```
C:\Users\[redacted]>ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\[redacted]/.ssh/id_rsa): C:\Keys\MyNewLightsailCustomKey
```

4. キーのパスフレーズを入力して、Enter を押します。パスフレーズの入力中にパスフレーズは表示されません。

このパスフレーズは後ほど、キーペアの公開キーが設定されているインスタンスに接続するために SSH クライアントでキーペアのプライベートキーを設定するときに必要になります。

```
Enter passphrase (empty for no passphrase):
```

5. 確認のためパスフレーズをもう一度入力して、Enter を押します。パスフレーズの入力中にパスフレーズは表示されません。

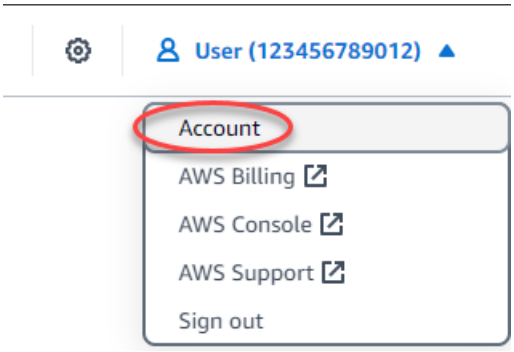
```
Enter same passphrase again:
```

6. 指定されたディレクトリにプライベートキーと公開キーが保存されたことを示すプロンプトが表示されます。

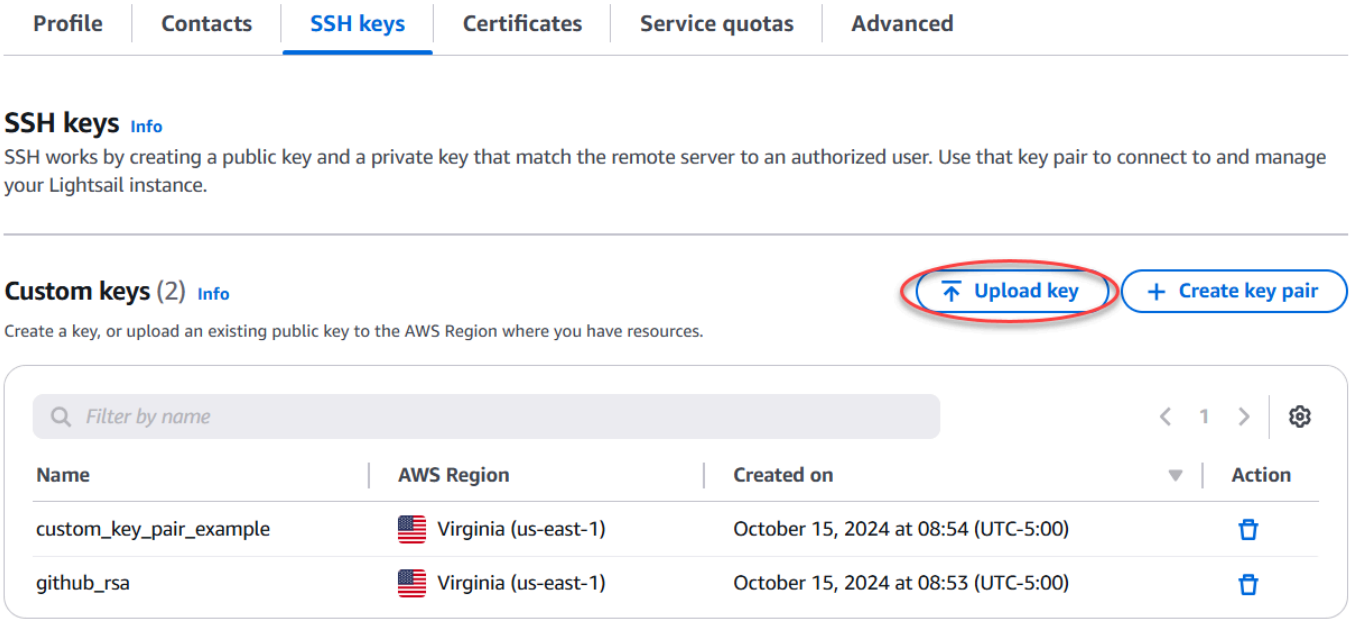
```
Your identification has been saved in C:\Keys\MyNewLightsailCustomKey.
Your public key has been saved in C:\Keys\MyNewLightsailCustomKey.pub.
```

次に、キーペアの公開キーを Lightsail コンソールにアップロードします。

7. Lightsail コンソール にサインインします。
8. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
9. ドロップダウンメニューで [Account (アカウント)] を選択します。

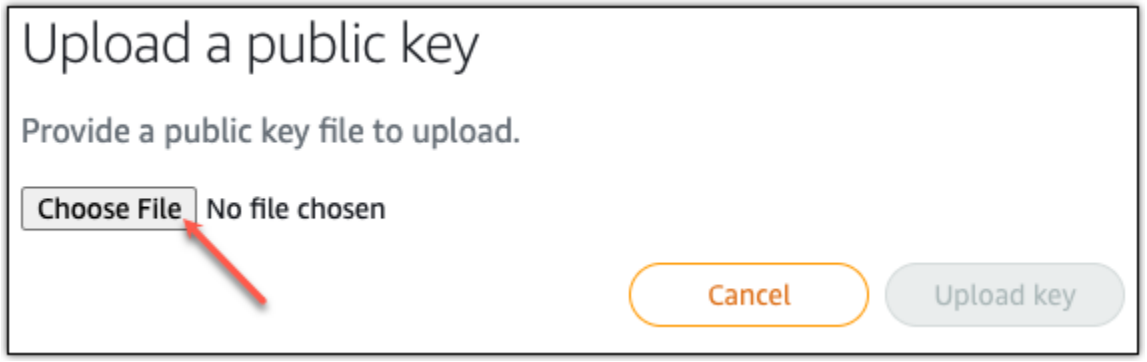


10. [SSH キー] タブを選択します。
11. そのページの [Custom keys] (カスタムキー) セクションで、[Upload key] (キーのアップロード) を選択します。

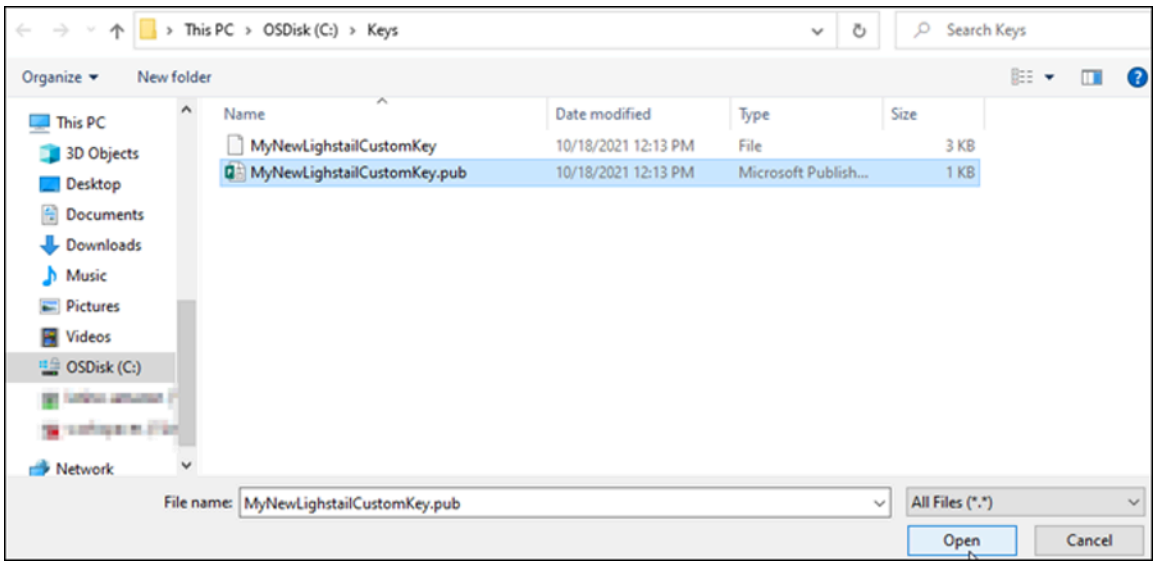


12. 表示されるリージョンの選択プロンプト AWS リージョン で、新しいカスタムキーをアップロードする を選択します。同じ AWS リージョン内の新しいインスタンスには、新しいカスタムキーを設定することができます。

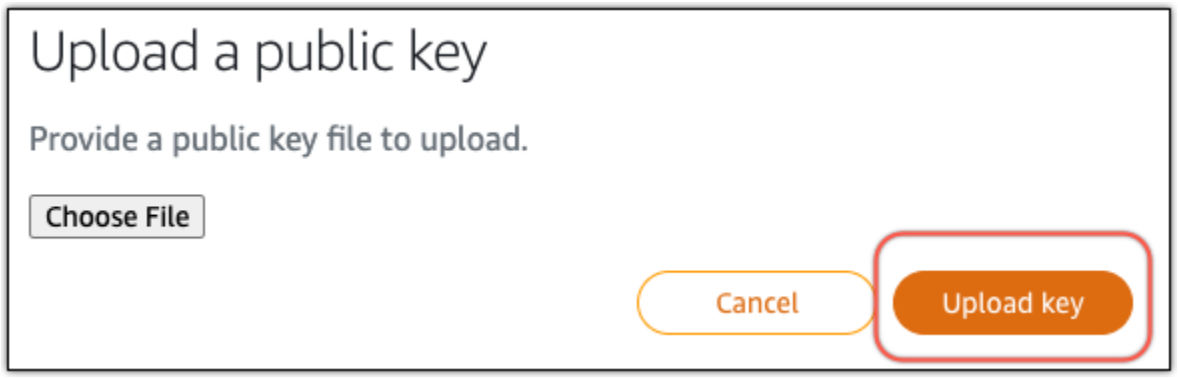
13. アップロード を選択します。
14. 表示される [Upload a public key] (公開キーのアップロード) プロンプトで、[Choose File] (ファイルを選択) をクリックします。



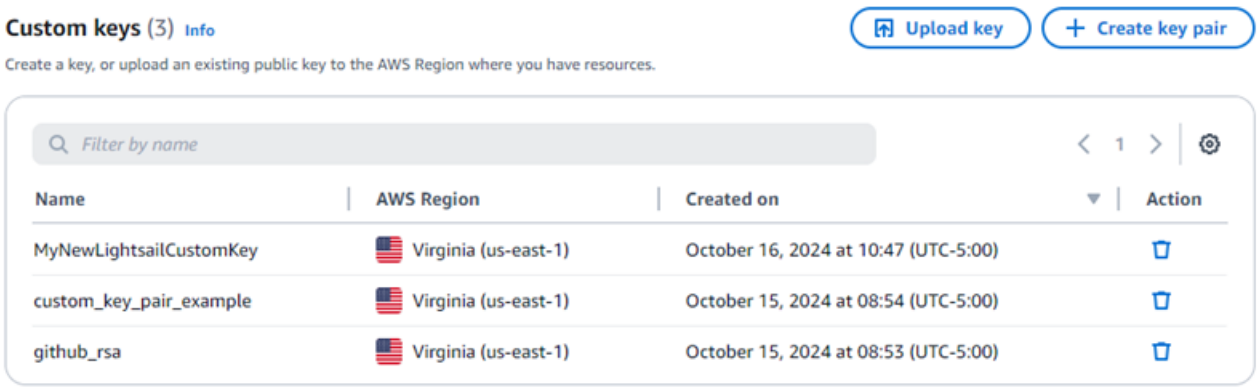
15. ローカルコンピュータで、この手順で先ほど作成したキーペアの公開キーを検索し、[Open] (開く) を選択します。キーペアの公開キーは、.PUB ファイル拡張子を持つファイルです。



16. [Upload key] (キーのアップロード) をクリックします。



17. 新しいカスタムキーは、このページの [Custom keys] (カスタムキー) セクションにリストされます。



キーをアップロードした AWS リージョンで作成する新しいインスタンスには、新しいカスタムキーを設定することができます。以前に作成された現在実行中のインスタンスで新しいカスタム

キーを設定するには、「Amazon Lightsail 内のインスタンスに保存されているキーを管理する」を参照してください。

Lightsail Linux インスタンスでの SSH キーを管理する

キーペアを使用して、Amazon Lightsail インスタンスへのセキュアな接続を確立することができます。Lightsail は、Linux または Unix インスタンスの新規作成時に、インスタンスでキーペアの公開キーを設定します。インスタンスに対する SSH 接続を確立するときは、キーペアのプライベートキーを使用してインスタンスへの認証を行います。キーの詳細については、「キーペアとインスタンスへの接続」を参照してください。

インスタンスが実行状態になったら、インスタンスに新しい公開キーを追加する、またはインスタンスの公開キーを交換 (既存の公開キーを削除して新しいものを追加) することで、インスタンスへの接続に使用されるキーペアを変更できます。次の理由から、これが必要になる場合があります。

- 組織内のユーザーが個別のキーペアを使用してインスタンスにアクセスする必要がある場合は、インスタンスに公開キーを追加できます。
- 漏洩したキーを使用していたインスタンスのスナップショットから作成された新しいインスタンスをセキュア化する必要がある場合。
- 誰かがプライベートキーのコピーを持っており、その人物がインスタンスに接続できないようにしたい場合 (例えば、その人物が組織から脱退した場合など)、インスタンスの公開キーを削除して、新しいものに交換することができます。

インスタンスのキーペアを追加または交換するには、インスタンスに接続できる必要があります。既存のプライベートキーを紛失した場合は、Lightsail のブラウザベースの SSH クライアントを使用してインスタンスに接続できます。詳細については、「Linux または Unix インスタンスへの接続」を参照してください。

目次

- ステップ 1: プロセスについて学ぶ
- ステップ 2: キーペアを作成する
- ステップ 3: インスタンスに公開キーを追加する
- ステップ 4: 新しいキーペアを使用してインスタンスに接続する
- ステップ 5: インスタンスから既存の公開キーを削除する

ステップ 1: プロセスについて学ぶ

以下は、インスタンスでキーを追加および削除するためのおおまかな手順です。新しいキーを追加せずにインスタンスからキーを削除する場合は、本ガイド後述の「ステップ 5: Delete an existing public key from your instance」(インスタンスから既存の公開キーを削除する) を参照してください。

1. キーペアを作成する – インスタンスに新しいキーを追加するには、まず新しいキーペアを作成する必要があります。Lightsail コンソールを使用してカスタムもしくはデフォルトのキーペアを作成する、または ssh-keygen などのサードパーティーツールを使用してローカルコンピュータでそれらを作成できます。どちらの方法でも、公開キーとプライベートキーで構成される新しいキーペアが生成されます。詳細については、本ガイド後述の「ステップ 2: キーペアを作成する」を参照してください。
2. インスタンスに公開キーを追加する – キーペアを作成したら、SSH を使用してインスタンスに接続し、キーペアの公開キーをインスタンスに追加します。詳細については、本ガイド後述の「ステップ 3:Add a public key to your instance」(インスタンスに公開キーを追加する) を参照してください。
3. 新しいキーペアを使用してインスタンスに接続できることをテストする – キーペアの公開キーがインスタンスに保存されたら、SSH を使用したインスタンスへの接続にキーペアのプライベートキーを使用できることをテストする必要があります。詳細については、本ガイド後述の「ステップ 4: Connect to your instance using the new key pair」(新しいキーペアを使用してインスタンスに接続する) を参照してください。
4. インスタンスから古い公開キーを削除する – 新しいキーを使用したインスタンスへの接続が正常に行われたら、インスタンスから古い公開キーを削除できます。このステップを実行して、ユーザーが古いキーペアを使用してインスタンスに接続できないようにします。詳細については、本ガイド後述の「ステップ 5:インスタンスから既存の公開キーを削除する」を参照してください。

ステップ 2: キーペアを作成する

ssh-keygen を使用してローカルコンピュータでキーペアを作成するには、以下の手順を実行します。

1. ローカルコンピュータで、コマンドプロンプトまたはターミナルを開きます。
2. 次のコマンドを入力して、新しいキーペアを作成します。

```
ssh-keygen -t rsa
```

3. キーペアを保存するコンピュータのディレクトリの場所を指定します。

   以下はその例です。

   - Windows の場合: C:\Users*<UserName>*\.ssh*<KeyPairName>*
   - macOS、Linux、または Unix の場合: /home/*<UserName>*/.ssh/*<KeyPairName>*

   <UserName> を現在サインインしているユーザーの名前に置き換えて、*<KeyPairName>* を新しいキーペアの名前に置き換えます。

   以下の例では、Windows コンピュータの C:\Keys ディレクトリを指定し、新しいキーに MyNewLightsailCustomKey という名前を付けました。

```
C:\Users\[redacted]>ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\[redacted]/.ssh/id_rsa): C:\Keys\MyNewLightsailCustomKey
```

4. キーのパスフレーズを入力して、Enter を押します。パスフレーズの入力中にパスフレーズは表示されません。

   このパスフレーズは後ほど、公開キーが設定されているインスタンスに接続するために SSH クライアントでプライベートキーを設定するときに必要になります。

```
Enter passphrase (empty for no passphrase):
```

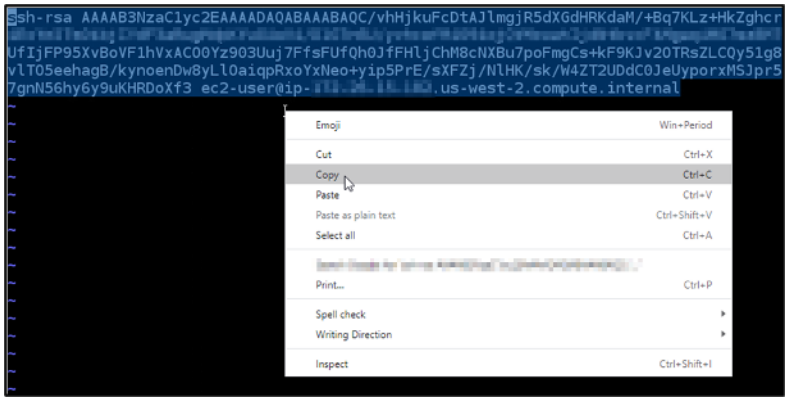
5. 確認のためパスフレーズをもう一度入力して、Enter を押します。パスフレーズの入力中にパスフレーズは表示されません。

```
Enter same passphrase again:
```

6. 指定されたディレクトリにプライベートキーと公開キーが保存されたことを示すプロンプトが表示されます。

```
Your identification has been saved in C:\Keys\MyNewLightsailCustomKey.
Your public key has been saved in C:\Keys\MyNewLightsailCustomKey.pub.
```

7. 公開キー (.PUB) ファイルを開いて、ファイル内のテキストをコピーします。

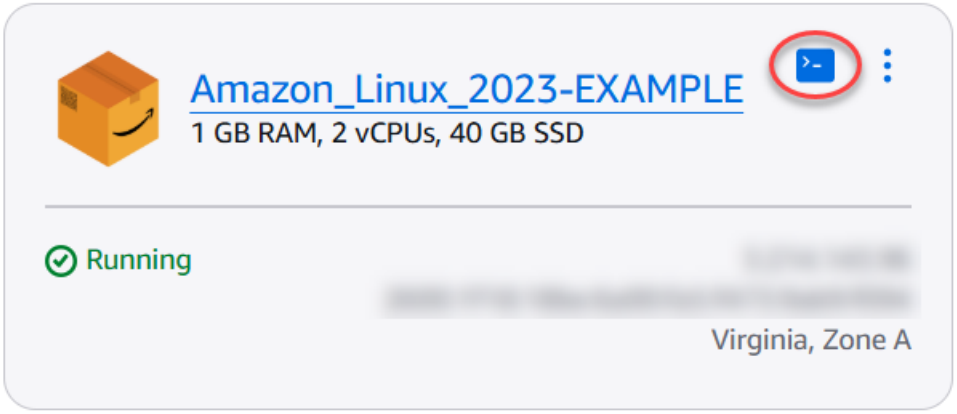


本ガイドの次のセクションに進み、Lightsail インスタンスに新しい公開キーを追加します。

ステップ 3: インスタンスに公開キーを追加する

インスタンスに公開キーを追加するには、次のステップを実行します。公開キーの内容は、Linux および Unix インスタンスの ~/.ssh/authorized_keys ファイルに保存されています。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページで [インスタンス] セクションを選択します。
3. 接続するインスタンスのブラウザベースの SSH クライアントアイコンをクリックします。



4. 接続されたら、任意のテキストエディタを使用して、authorized_keys ファイルを編集するための以下のコマンドを入力します。以下の手順では、デモ用に Vim を使用します。

```
sudo vim ~/.ssh/authorized_keys
```

インスタンス上で設定されている現在の公開キーは、次の例のような結果で表示されます。今回の場合、インスタンスが作成された AWS リージョン の Lightsail デフォルトキーが、インスタンスで設定されている唯一の公開キーになります。

5. [I] キーを押して、Vim エディタを挿入モードにします。
6. ファイルの最後の公開キーの後に改行を入力します。
7. このガイドの前のセクションで (新しいキーペアを作成した後) コピーした公開キーテキストを貼り付けます。以下の例のような結果が表示されるはずです。

8. ESC キーを押します。次に、:wq! を入力し、Enter キーを押して編集内容を保存して、Vim エディタを終了します。

これで、インスタンスに新しい公開キーが追加されました。本ガイドの次のセクションに進み、新しいキーペアを使用してインスタンスに接続します。

ステップ 4: 新しいキーペアを使用してインスタンスに接続する

新しいキーペアをテストするには、インスタンスとの接続を切断してから、このガイドで先ほど作成したプライベートキーを使用してインスタンスに再接続します。詳細については、「キーペアと Amazon Lightsail 内のインスタンスへの接続」を参照してください。新しいキーを使用してインスタンスに正常に接続したら、インスタンスから古いキーを削除できます。次のステップに進んで、インスタンスから公開キーを削除する方法を学びます。

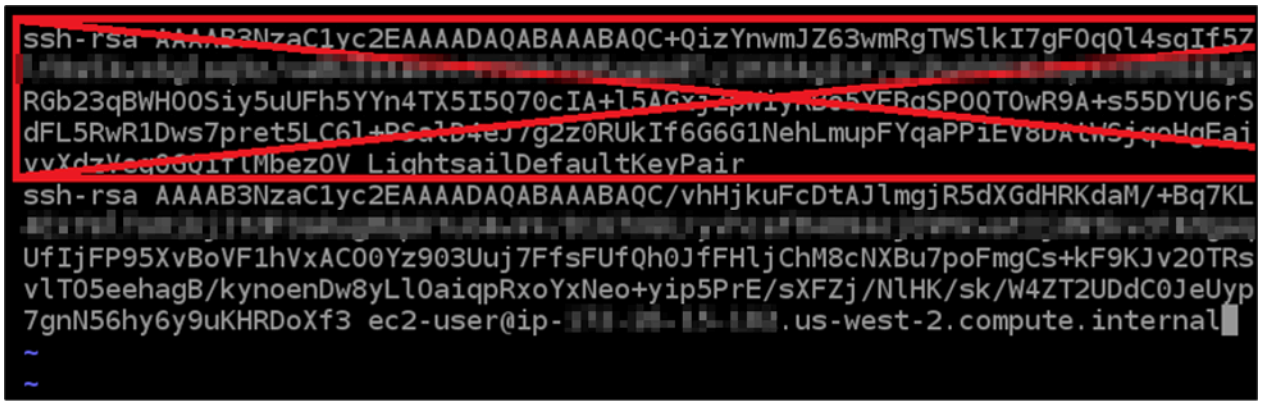
ステップ 5: インスタンスから既存の公開キーを削除する

インスタンスから公開キーを削除するには、以下の手順を実行します。これは、ユーザーが古いキーペアを使用してインスタンスに接続できないようにします。この手順は、新しいキーペアを使用したインスタンスへの接続が正常に行われた後で実行してください。

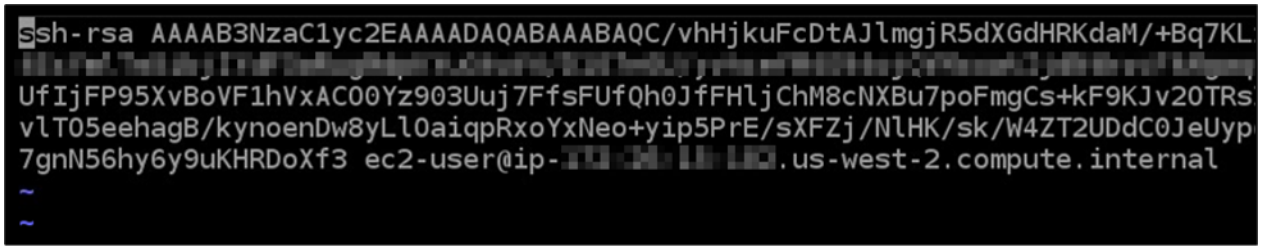
1. SSH を使用してインスタンスに接続します。
2. 任意のテキストエディアを使用して、authorized_keys ファイルを編集するための以下のコマンドを入力します。以下の手順では、デモ用に Vim を使用します。

```
sudo vim ~/.ssh/authorized_keys
```

3. I の文字キーを押して、Vim エディタを挿入モードにします。
4. インスタンスから削除したい公開キーを含んでいるテキストの行を削除します。



結果は以下の例のようになります。ここでは、新しい公開キーが、表示されている唯一のキーです。



5. ESC キーを押します。次に、:wq! を入力し、Enter キーを押して編集内容を保存して、Vim エディタを終了します。

削除された公開キーは、インスタンスから消去された状態です。インスタンスは、そのキーペアのプライベートキーを使用する接続を拒否します。

Lightsail で Linux または Unix のインスタンスに接続する

Amazon Lightsail が提供するブラウザベースの SSH クライアントは、Linux または UNIX インスタンスに接続する最も簡単な方法です。独自の SSH クライアントを使用してインスタンスに接続することもできます。詳細については、「PuTTY をダウンロードしてセットアップする」を参照してください。

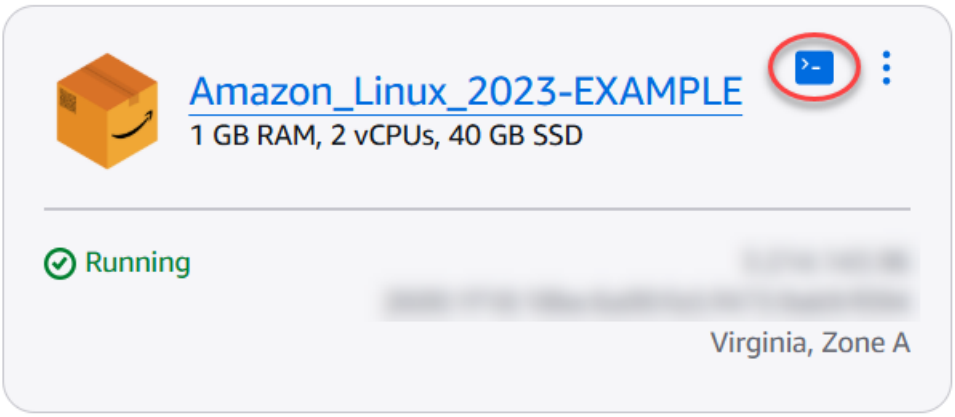
SSH を使用してインスタンスに接続し、ソフトウェアパッケージのインストールやウェブアプリケーションの設定などの管理タスクをサーバーで実行します。ブラウザベースの SSH クライアントは、ソフトウェアのインストールを必要とせず、インスタンスの作成後、ほぼ即座に使用できます。

Lightsail で Windows Server インスタンスに接続するには、「Windows ベースのインスタンスに接続する」を参照してください。

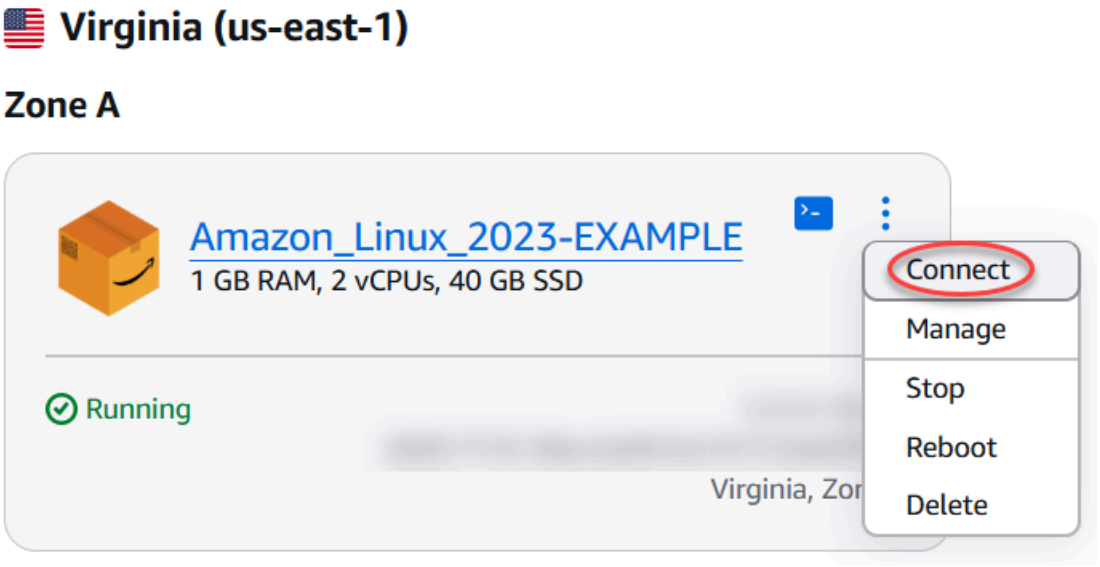
Linux または Unix インスタンスに接続するには

1. Lightsail コンソール にサインインします。
2. 接続先のインスタンスのブラウザベースの SSH クライアントにアクセスするには、以下のいずれかの操作を行います。

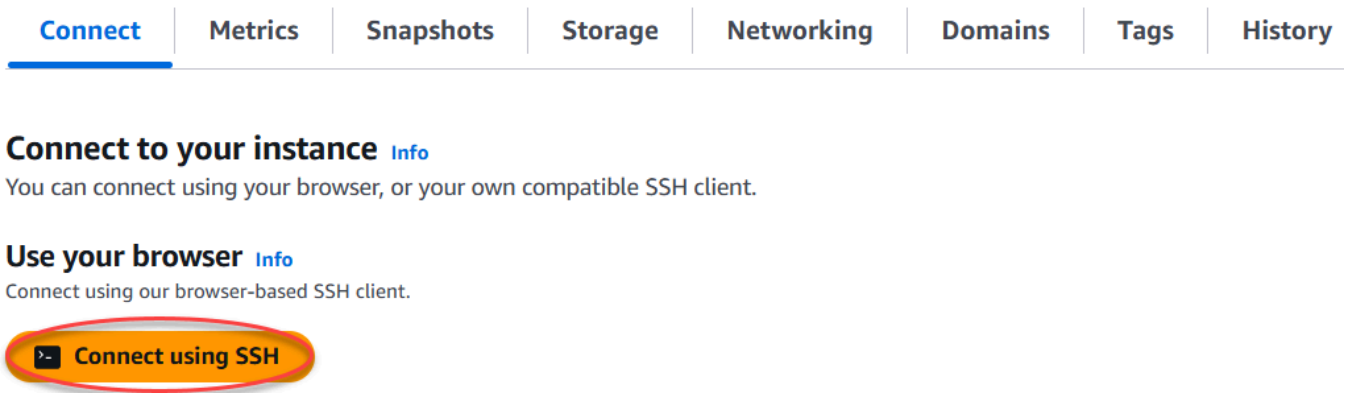
   - 次の例に示すように、クリック接続アイコンを選択します。



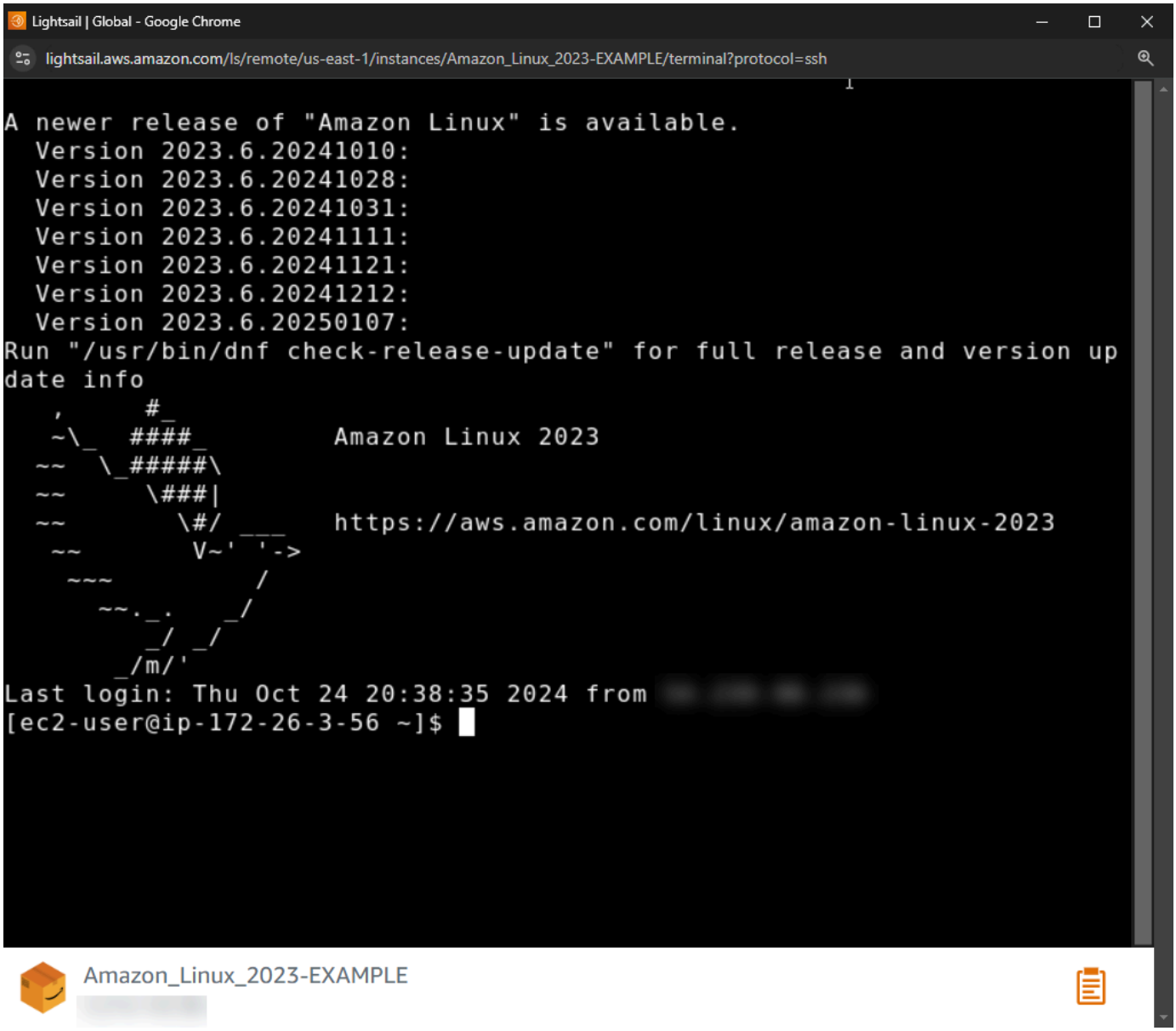
   - アクションメニューアイコン (⋮) を選択し、次に [接続] を選択します。



- インスタンスの名前を選択し、[接続] タブの [SSH を使用して接続] を選択します。



ブラウザベースの SSH クライアントを開いて、次の例に示すようなターミナル画面が表示されると、インスタンスとのやり取りを開始できます。



> **Note**
>
> [接続] タブには、独自の SSH クライアントを使用して接続するために必要な情報も表示されます。詳細については、「PuTTY をダウンロードしてセットアップする」を参照してください。

ブラウザベースの SSH クライアントを使用して Linux または Unix インスタンスとやり取りする

ブラウザベースの SSH クライアントのターミナル画面に直接 Linux または UNIX コマンドを入力したり、ターミナル画面からテキストをコピーしたりします。以下のセクションでは、SSH でクリップボードに (またはクリップボードから) テキストをコピーして貼り付ける方法を示します。

ブラウザベースの SSH クライアントにテキストを貼り付けるには

1. ローカルデスクトップのテキストを強調表示し、Ctrl+C または Cmd+C を押してテキストをローカルクリップボードにコピーします。
2. ブラウザベースの SSH クライアントの右下で、クリップボードアイコンを選択します。ブラウザベースの SSH クライアントのクリップボードテキストボックスが表示されます。
3. テキストボックス内をクリックし、Ctrl+V または Cmd+V を押して、ローカルクリップボードの内容をブラウザベースの SSH クライアントのクリップボードに貼り付けます。
4. SSH ターミナル画面の任意の領域を右クリックし、ブラウザベースの SSH クライアントのクリップボードからターミナル画面にテキストを貼り付けます。

ブラウザベースの SSH クライアントからテキストをコピーするには

1. ターミナル画面でテキストを強調表示します。
2. ブラウザベースの SSH クライアントの右下で、クリップボードアイコンを選択します。ブラウザベースの SSH クライアントのクリップボードテキストボックスが表示されます。
3. コピーするテキストを強調表示し、Ctrl+C または Cmd+C を押してテキストをローカルクリップボードにコピーします。これで、コピーしたテキストをローカルデスクトップの任意の場所に貼り付けることができます。

SSH コマンドを使用して Lightsail Linux または Unix インスタンスに接続する

macOS を含む Linux または Unix オペレーティングシステムが使用されるローカルマシンの場合、ターミナルウィンドウを介した SSH クライアントを使用すると、Amazon Lightsail の Linux または Unix インスタンスに接続することができます。

このガイドで説明されているインスタンスへの接続方法は、たくさんある接続方法のうちの 1 つです。他の方法に関する詳細は、「SSH のキーペア」を参照してください。

Lightsail でのLinux または Unix インスタンスに接続する最も簡単な方法は、Lightsail コンソールで利用可能なブラウザベースの SSH クライアントを使用する方法です。詳細については、「Linux または Unix インスタンスに接続する」を参照してください。

内容

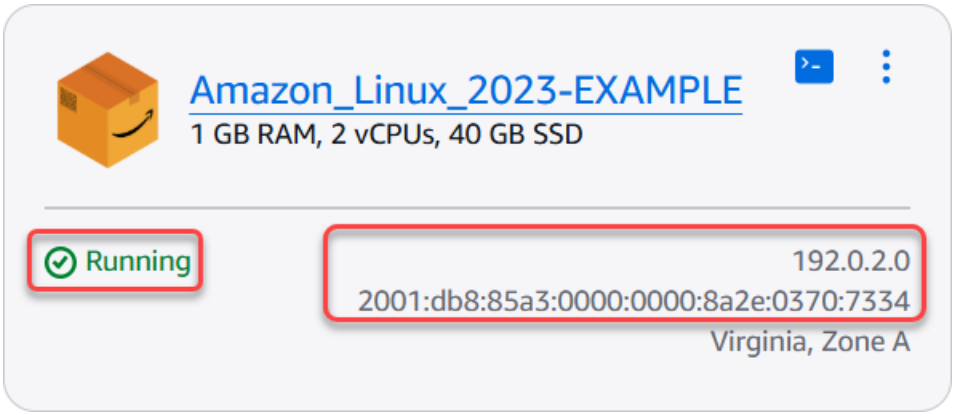
- ステップ 1: インスタンスが実行されていることを確認し、パブリック IP アドレスを取得する
- ステップ 2: インスタンスで使用されている SSH キーペアを確認する
- ステップ 3: プライベートキーのアクセス許可を変更し、SSH を使ってインスタンスに接続する

ステップ 1: インスタンスが実行されていることを確認し、パブリック IP アドレスを取得する

次の手順では、Lightsail コンソールにサインインしてインスタンスが実行中であることを確認し、インスタンスのパブリック IP アドレスを取得します。SSH 接続を確立するには、インスタンスが実行中になっている必要があります。またこのガイドの後半で SSH に接続する際、インスタンスのパブリック IP アドレスが必要になります。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの [インスタンス] セクションで、接続先のインスタンスを見つけます。
3. インスタンスが実行中であることを確認し、インスタンスのパブリック IP アドレスを書き留めます。

   次の例に示すように、インスタンスの状態とパブリック IP アドレスはインスタンス名の横に表示されます。

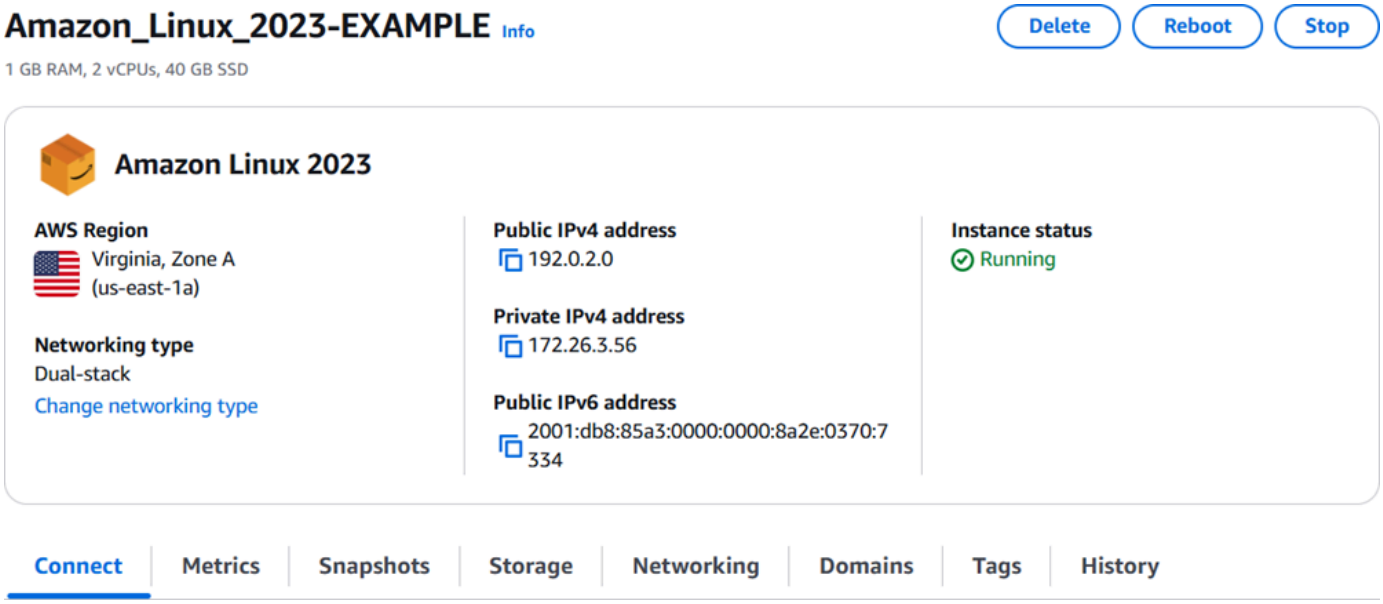


ステップ 2: インスタンスで使用されている SSH キーペアを確認する

次の手順では、インスタンスで使用されている SSH キーペアを確認します。インスタンスに対して認証し、SSH 接続を確立するには、キーペアのプライベートキーが必要になります。

1. Lightsail ホームページの [インスタンス] セクションで、接続先のインスタンス名を見つけます。

   インスタンス管理ページが表示され、インスタンスを管理するためのさまざまなタブオプションが表示されます。

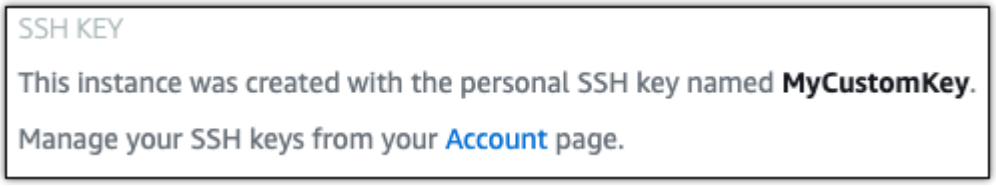


2. [接続] タブで、下にスクロールして、インスタンスで使用されているキーペアを確認します。考えられる可能性は 2 つあります。

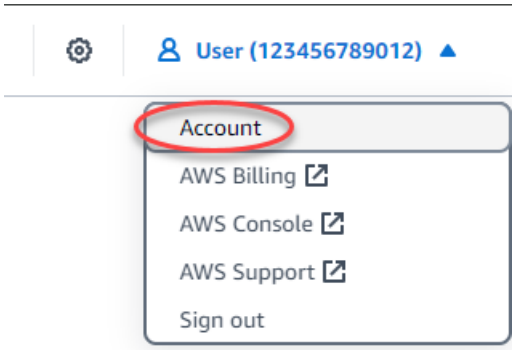
   1. 次の例は、インスタンスが作成された AWS リージョン用のデフォルトのキーペア を使用しているインスタンスを示しています。インスタンスがデフォルトのキーペアを使用している場合は、この手順のステップ 3 に進み、キーペアのプライベートキーをダウンロードします。Lightsail は、各 AWS リージョンのデフォルトキーペア専用のプライベートキーを保存します。

   2. 次の例は、ユーザによってアップロードまたは作成されたカスタムキーペアを使用しているインスタンスを示しています。インスタンスがカスタムキーペアを使用している場合は、キーを保存している、カスタムキーペアのプライベートキーの位置を特定する必要があります。カスタムキーペアのプライベートキーを無くした場合、インスタンスへの SSH 接続を、独自のクライアントを使って確立することができなくなります。ただし、ブラウザベースの SSH クライアントは Lightsail コンソールで引き続き使用できます。カスタムキーペアのプラ

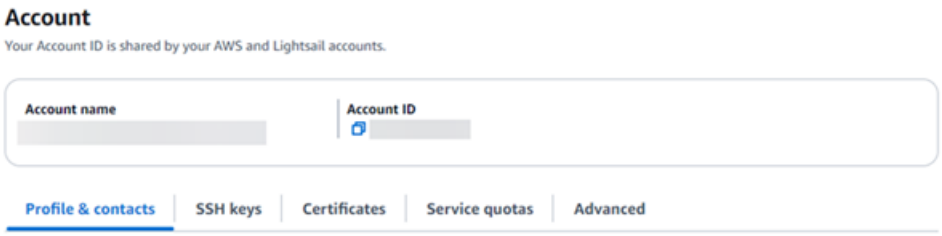
イベートキーの位置を特定したら、このガイドの次の「ステップ 3: プライベートキーの権限を変更し、SSH を使ってインスタンスに接続する」のセクションに進んでください。



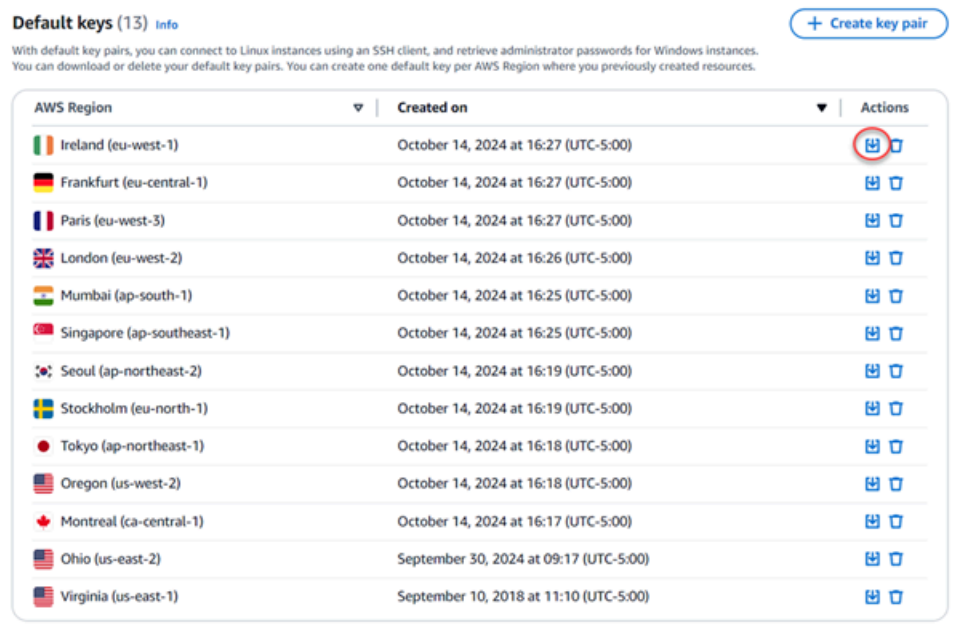
3. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
4. ドロップダウンメニューで [Account (アカウント)] を選択します。



アカウント管理ページが表示され、アカウント設定を管理するためのさまざまなタブオプションが表示されます。



5. [SSH キー] タブを選択します。
6. 下にスクロールし、接続先インスタンスの AWS リージョン のデフォルトキーの横にあるダウンロードアイコンを選択します。



プライベートキーはユーザーのローカルマシンにダウンロードされます。ユーザーのホームディレクトリにある「Keys」フォルダなど、SSH キーが保存されているディレクトリに、ダウンロードしたキーを移動することも可能です。このガイドの次のセクションで、プライベートキーが保存されるディレクトリを参照する必要があります。プライベートキーが .pem 以外の形式で保存しようとした場合、保存する前に手動で形式を .pem に変更する必要があります。

> **Note**
>
> Lightsail には、.pem ファイルやその他の証明書形式を操作するためのユーティリティはありません。プライベートキーファイルの形式変換する必要がある場合、OpenSSL などのフリーのオープンソースツールを容易に利用できます。

このガイドの次の「ステップ 3: プライベートキーのアクセス権限を変更して、SSH を使用してインスタンスに接続する」のセクションに進んで、ダウンロードしたプライベートキーを使ってインスタンスへの SSH 接続を確立します。

ステップ 3: プライベートキーのアクセス許可を変更し、SSH を使ってインスタンスに接続する

次の手順では、プライベートキーファイルの権限を変更して、お客様以外のユーザーが読み書きできないように変更します。次に、ローカルマシンでターミナルウィンドウを開き、SSH コマンドを実行して Lightsail のインスタンスとの接続を確立します。

1. ローカルマシンでターミナルウィンドウを開きます。
2. 次のコマンドを入力して、キーペアのプライベートキーが本人にしか読み書きできないようにします。これは、一部のオペレーティングシステムで要求される、セキュリティのベストプラクティスです。

```
sudo chmod 400 /path/to/private-key.pem
```

コマンドで */path/to/private-key.pem* を、インスタンスで使われるキーペアのプライベートキーが保存されている場所を向いたディレクトリパスに、置き換えます。

例:

```
sudo chmod 400 /Users/user/Keys/LightsailDefaultKey-us-west-2.pem
```

3. 次のコマンドを入力して、Lightsail のインスタンスに SSH を使用して接続します。

```
ssh -i /path/to/private-key.pem username@public-ip-address
```

コマンドを、以下のように置き換えます。

- */path/to/private-key.pem* を、インスタンスで使われるキーペアのプライベートキーが保存されている場所を向いたディレクトリパスに置き換えます。
- *username* をインスタンスのユーザーネームに置き換えます。インスタンスで使用されるブループリントに応じて、以下のいずれかのユーザー名を指定できます。
  - AlmaLinux OS 9、Amazon Linux 2、Amazon Linux 2023、CentOS Stream 9、FreeBSD、および openSUSE インスタンスの場合: `ec2-user`
  - Debian インスタンス: `admin`
  - Ubuntu インスタンス: `ubuntu`
  - Bitnami インスタンス: `bitnami`
  - Plesk インスタンス: `ubuntu`
  - cPanel & WHM インスタンス : `centos`
- *public-ip-address* を、このガイドの前の Lightsail コンソールのセクションで書き留めたインスタンスのパブリック IP アドレスに置き換えます。

絶対パスの例:

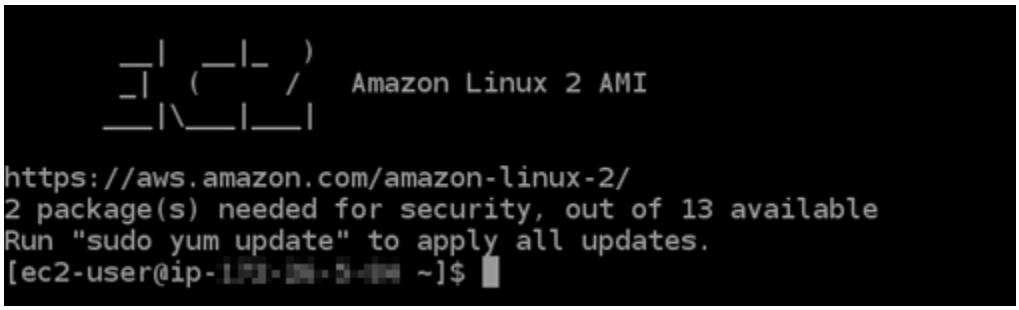
```
ssh -i /Users/user/Keys/LightsailDefaultKey-us-west-2.pem ec2-user@192.0.2.0
```

相対パスの例:

./ が .pem ファイルをプレフィックスすることに気を付けてください。./ を省略して単に LightsailDefaultKey-us-west-2.pem を書くだけでは動作しません。

```
ssh -i ./LightsailDefaultKey-us-west-2.pem ec2-user@192.0.2.0
```

インスタンスにようこそ、のメッセージが表示されたら、インスタンスには正常に接続された状態です。次の例は、Amazon Linux 2 インスタンスのウェルカムメッセージを示しています。他のインスタンスのブループリントでも、同様のウェルカムメッセージがあります。接続後、Lightsail でインスタンスにコマンドを実行することが可能です。接続を解除するには、exit を入力して Enter を押します。



PuTTY で Linux/Unix Lightsail インスタンスに接続する

Lightsail のブラウザベースの SSH ターミナルだけでなく、PuTTY などの SSH クライアントを使用して Linux ベースのインスタンスに接続することもできます。PuTTY のセットアップ方法については、「PuTTY をダウンロードし、Lightsail で SSH を使用して接続するようにセットアップする」を参照してください。

> **Note**
>
> RDP を使用して Windows ベースのインスタンスに接続するには、「Windows ベースの Lightsail インスタンスに接続する」を参照してください。

Lightsail に付属しているデフォルトのプライベートキー (Lightsail の新しいプライベートキー) を使用するか、別のサービスで使用している既存のプライベートキーを使用できます。

1. PuTTY を起動します (たとえば、[スタート] メニューで [すべてのプログラム]、[PuTTY]、[PuTTY] の順に選択します)。
2. [ロード] を選択し、保存済みセッションを見つけます。

   保存済みセッションがない場合は、「ステップ 4: プライベートキーとインスタンスの情報を使用して PuTTY の設定を完了する」を参照してください。
3. インスタンスのオペレーティングシステムに応じて、次のいずれかのデフォルトのユーザー名を使用してログインします。

   - AlmaLinux、Amazon Linux 2、Amazon Linux 2023、CentOS Stream 9、FreeBSD および openSUSE インスタンス: `ec2-user`
   - Debian インスタンス: `admin`
   - Ubuntu インスタンス: `ubuntu`
   - Bitnami インスタンス: `bitnami`
   - Plesk インスタンス: `ubuntu`
   - cPanel & WHM インスタンス: `centos`

   インスタンスのオペレーティングシステムの詳細については、「Lightsail のイメージの選択」を参照してください。

SSH の詳細については、「SSH および Amazon Lightsail インスタンスへの接続」を参照してください。

PuTTY で Lightsail Linux インスタンスに接続する

PuTTY などの SSH クライアントを使用して Amazon Lightsail インスタンスに接続できます。PuTTY ではプライベート SSH キーのコピーが必要です。すでに持っているキーを使用するか、または Lightsail によって作成されるキーペアを使用できます。どちらの方法についても説明しています。SSH の詳細については、「SSH キーペア」を参照してください。このトピックでは、キーペアをダウンロードし、インスタンスに接続するように PuTTY をセットアップするステップについて順を追って説明します。

このガイドで説明するインスタンスへの接続方法は、多数あるうちの 1 つです。他の方法に関する詳細は、「SSH のキーペア」を参照してください。

Lightsail でのLinux または Unix インスタンスに接続する最も簡単な方法は、Lightsail コンソールで利用可能なブラウザベースの SSH クライアントを使用する方法です。詳細については、「Amazon Lightsail のLinux インスタンスへの接続」を参照してください。

前提条件

- Lightsail で実行中のインスタンスが必要です。詳細については、「Amazon Lightsail でインスタンスを作成する」を参照してください。
- 静的 IP アドレスを作成してインスタンスにアタッチすることをお勧めします。これにより、後でパブリック IP アドレスが変わった場合に、PuTTY を再設定する必要がなくなります。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。

ステップ 1: PuTTY をダウンロードしてインストールする

PuTTY は、Windows 用の SSH の無料の実装です。PuTTY の詳細については、「PuTTY のウェブサイト」を参照してください。暗号化が許可されていない国に関する制約なども記載されています。すでに PuTTY を持っている場合は、この手順をスキップしてステップ 2 に進むことができます。

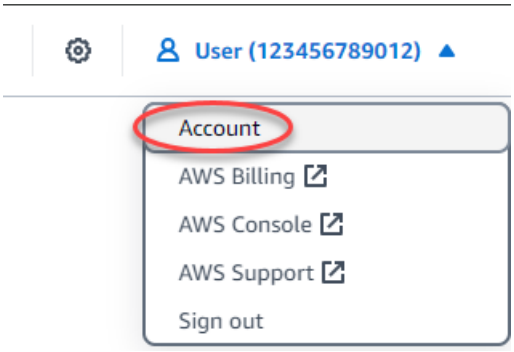
1. PuTTY のインストーラまたは実行可能ファイルを、「PuTTY のダウンロード」からダウンロードします。

   どのダウンロードを選択すればいいか判断できない場合は、PuTTY のドキュメントを参照してください。 の最新バージョンを使用することをお勧めします。
2. ステップ 2 に進み、PuTTY を設定する前にプライベートキーを取得します。

ステップ 2: プライベートキーの準備を整える

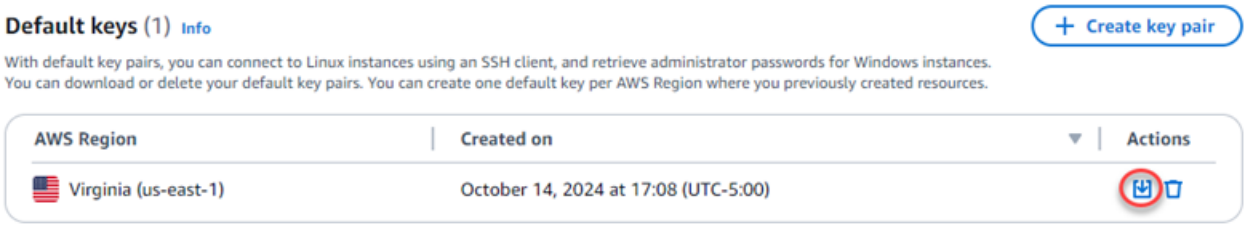
プライベートキーを取得する方法にはいくつかの選択肢があります。Lightsail が生成するデフォルトのプライベートキーを使用するか、Lightsail で新しいプライベートキーを作成するか、別のサービスから取得済みのプライベートキーを使用できます。各オプションの手順については、以下に概要を示します。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
3. ドロップダウンメニューで [Account (アカウント)] を選択します。



4. [SSH キー] タブを選択します。
5. 使用するプライベートキーに応じて、次のいずれかのオプションを選択します。

   - がLightsail生成するデフォルトのプライベートキーを使用するには、ページのデフォルトキーセクションで、インスタンス AWS リージョン が配置されている のデフォルトのプライベートキーの横にあるダウンロードアイコンを選択します。



   - Lightsail で新しいキーペアを作成するには、ページの [Custom keys] セクションで [Create key pair] (キーペアを作成) を選択します。インスタンス AWS リージョン がある を選択し、作成 を選択します。名前を入力し、[Generate key pair] (キーペアの生成) を選択します。プライベートキーをダウンロードするためのオプションが表示されます。

     > **Important**
     >
     > プライベートキーは一度だけダウンロードすることができます。セキュリティで保護されている場所に保存します。

   - 独自のキーペアを使用するには、[今すぐアップロード] を選択します。インスタンス AWS リージョン が配置されている を選択し、アップロードを選択します。[Upload file (ファイルのアップロード)] を選択し、ローカルドライブのファイルを見つけます。Lightsail に公開キーファイルをアップロードする準備ができたら、[キーのアップロード] を選択します。
6. Lightsail でプライベートキーをダウンロードした場合や新しいプライベートキーを作成した場合は、`.pem` キーファイルを見つけやすい場所に必ず保存します。

他のユーザーが読み取ることができないようにファイルのアクセス許可も設定することをお勧めします。

ステップ 3: Lightsail のプライベートキーを使用して PuTTYgen を設定する

`.pem` キーファイルのコピーを作成したら、PuTTY キージェネレータ (PuTTYgen) を使用して PuTTY をセットアップできます。

1. PuTTYgen を起動します (例: [スタート] メニューで [すべてのプログラム]、[PuTTY]、[PuTTYgen] の順に選択します)。
2. [ロード] を選択します。

   PuTTYgen では、デフォルトでは `.ppk` 拡張子を持つファイルだけが表示されます。`.pem` ファイルの場所を特定するには、すべてのタイプのファイルを表示するオプションを選択します。
3. `lightsailDefaultKey.pem` を選択して [Open (開く)] を選択します。

   キーの正常なインポートが PuTTYgen で確認されたら、[OK] を選択できます。
4. [Save private key (プライベートキーを保存)] を選択し、パスフレーズ付きで保存しないことを確認します。

   追加のセキュリティ対策としてパスフレーズを作成することを選択した場合は、PuTTY を使用してインスタンスに接続するたびにそのパスフレーズを入力する必要があります。
5. プライベートキーを保存する名前と場所を指定し、[Save (保存)] を選択します。
6. PuTTYgen を閉じます。

ステップ 4: プライベートキーとインスタンスの情報を使用して PuTTY の設定を完了する

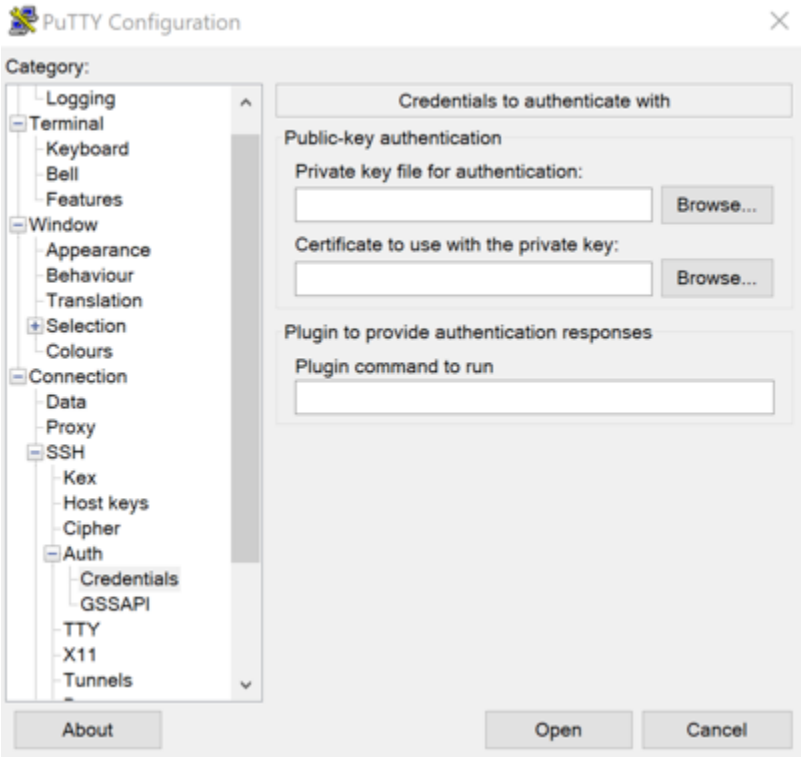
もう少しです。これから最後の変更を行います。

1. PuTTY を開きます。
2. Lightsail で、インスタンス管理ページからパブリック IP アドレス (静的 IP アドレスを使用することをお勧めします) を取得します。

   パブリック IP アドレスは、Lightsail ホームページで取得するか、またはインスタンスを選択してその詳細情報を表示することによって取得できます。
3. パブリック IP アドレスを [Host Name (or IP address) (ホスト名 (または IP アドレス))] フィールドに入力するか、貼り付けます。

> **Note**
>
> Lightsail インスタンスでは SSH 用にポート 22 がすでに開かれているため、そのデフォルトポートをそのまま使用します。

4. [接続] の下の [SSH] および [認証] を展開して、[認証情報] を選択します。



5. [Browse (参照)] を選択し、前のステップで作成した .ppk ファイルの場所に移動して選択し、[Open (開く)] を選択します。
6. [開く] を再度選択し、[承諾] を選択して、今後はこの接続を信頼することを示します。
7. インスタンスのオペレーティングシステムに応じて、次のいずれかのデフォルトのユーザー名を使用してログインします。

   - AlmaLinux、Amazon Linux 2、Amazon Linux 2023、CentOS Stream 9、FreeBSD および openSUSE インスタンス: `ec2-user`
   - Debian インスタンス: `admin`
   - Ubuntu インスタンス: `ubuntu`
   - Bitnami インスタンス: `bitnami`
   - Plesk インスタンス: `ubuntu`

- cPanel & WHM インスタンス : `centos`

インスタンスのオペレーティングシステムの詳細については、「イメージの選択」を参照してください。

8. 後で使用できるように接続を保存します。

次の手順

再度接続する必要がある場合は、「PuTTY を使用して Linux/Unix ベースのインスタンスに接続する」を参照してください。

SFTP を使用して Lightsail Linux インスタンスにファイルを安全に転送する

SFTP (SSH File Transfer プロトコル) を使用してインスタンスに接続することによって、ローカルコンピュータと Amazon Lightsail のLinux または UNIX インスタンスとの間にファイルを転送できます。これを行うには、インスタンスのプライベートキーを取得する必要があり、これを使用して FTP クライアントを設定します。このチュートリアルでは、FileZilla FTP クライアントを設定してインスタンスに接続する方法について説明します。これらのステップは、他の FTP クライアントにも適用されます。

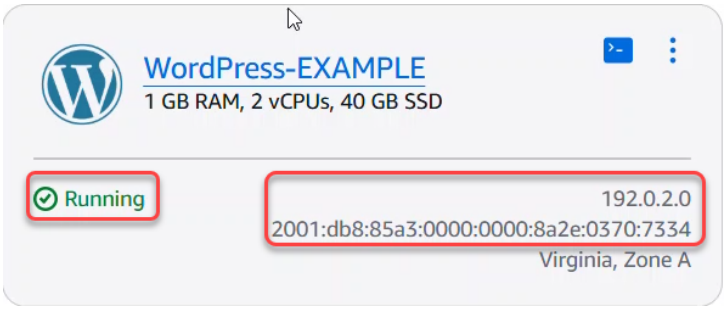
内容

- 前提条件
- インスタンスの SSH キーを取得する
- FileZilla を設定してインスタンスを接続する

前提条件

以下の前提条件を完了します (まだの場合)。

- FileZilla をローカルコンピュータにダウンロードしてインストールします。詳細については、次のダウンロードオプションを参照してください。
  - Windows 用 FileZilla クライアントをダウンロードする
  - Mac OS X 用 FileZilla クライアントをダウンロードする
  - Linux 用 FileZilla クライアントをダウンロードする
- インスタンスのパブリック IP アドレスを取得します。Lightsail コンソールにサインインし、以下の例のように、インスタンスの横に表示されるパブリック IP アドレスをコピーします。

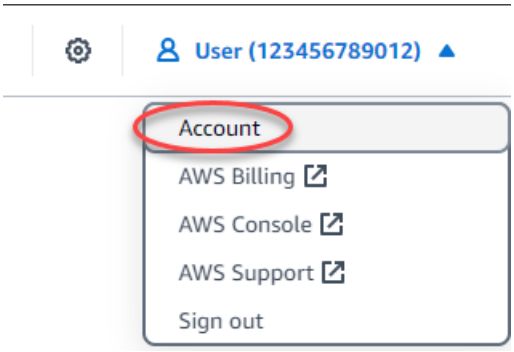


インスタンスの SSH キーを取得する

インスタンスの AWS リージョンのデフォルトのプライベートキーを取得するには、次のステップを実行します。これは、FileZilla を使用してインスタンスに接続するために必要です。

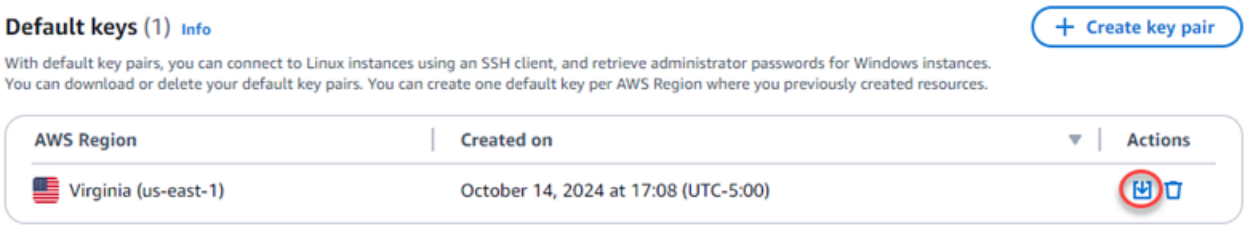
> **Note**
>
> 独自のキーペアを使用している場合、または Lightsail コンソールを使用してキーペアを作成した場合は、プライベートキーを見つけ、それを使用してインスタンスに接続します。独自のキーをアップロードする場合または Lightsail コンソールを使用してキーペアを作成する場合は Lightsail はプライベートキーを保存しません。プライベートキーなしで SFTP を使用してインスタンスに接続することはできません。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
3. ドロップダウンメニューで [Account (アカウント)] を選択します。



4. [SSH キー] タブを選択します。
5. ページの [Default keys] (デフォルトキー) セクションまで下にスクロールします。

6. インスタンスが配置されているリージョンのデフォルトのプライベートキーの横にある [ダウンロード] を選択します。

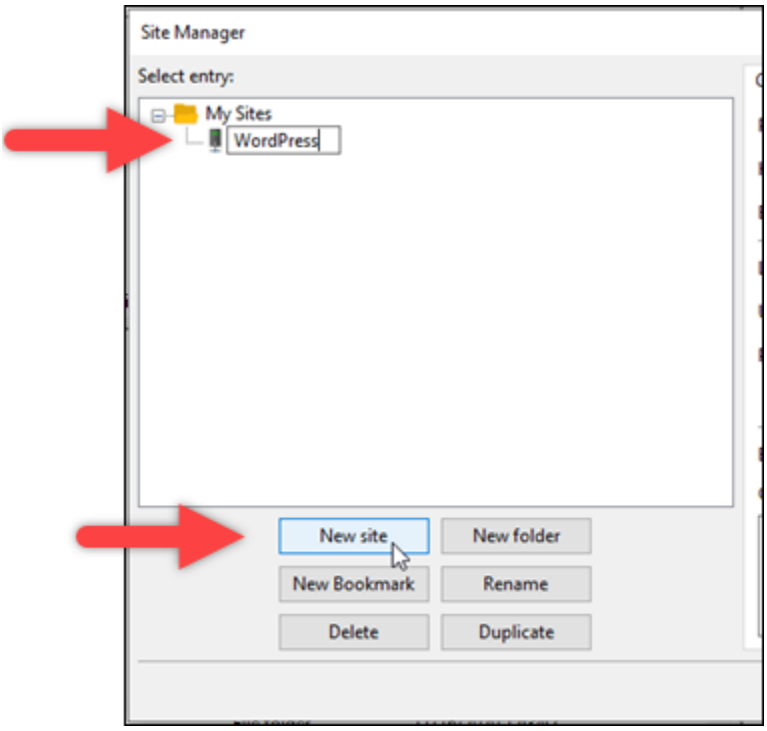


7. ローカルドライブのセキュリティが確保された場所にプライベートキーを保存します。

FileZilla を設定してインスタンスを接続する

以下のステップに従って、インスタンスに接続するよう FileZilla を設定します。

1. FileZilla を開きます。
2. [ファイルFile]、[サイトマネージャー] を選択します。
3. [新しいサイト] を選択してサイトに名前を付けます。



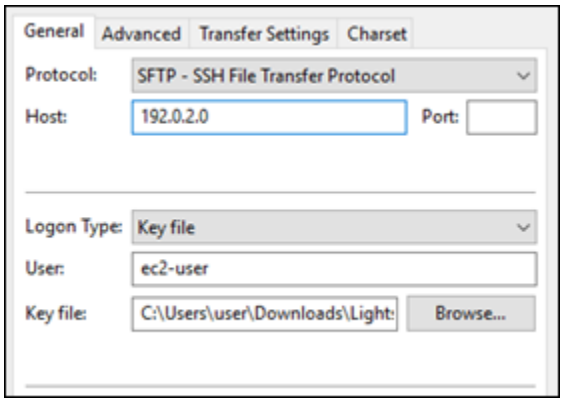
4. [プロトコル] ドロップダウンで、[SFTP – SSH File Transfer プロトコル] を選択します。
5. [ホスト] テキストボックスに、インスタンスのパブリック IP アドレスを入力するか、貼り付けます。

6. [ログオンタイプ] ドロップダウンで、[キーファイル] を選択します。
7. [ユーザー] テキストボックスに、インスタンスのオペレーティングシステムに応じて、次のいずれかのデフォルトのユーザー名を入力します。

   - AlmaLinux、Amazon Linux 2、Amazon Linux 2023、CentOS Stream 9、FreeBSD および openSUSE インスタンス: `ec2-user`
   - Debian インスタンス: `admin`
   - Ubuntu インスタンス: `ubuntu`
   - Bitnami インスタンス: `bitnami`
   - Plesk インスタンス: `ubuntu`
   - cPanel & WHM インスタンス : `centos`

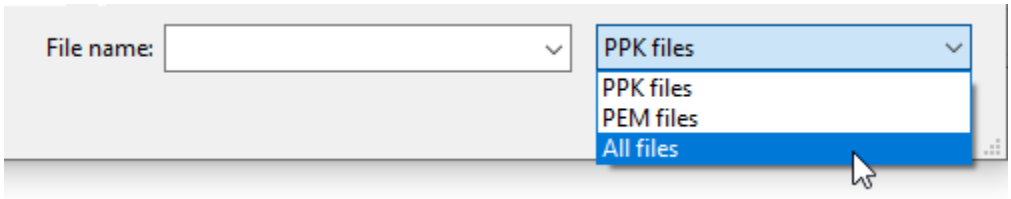
   > **Important**
   >
   > ここにリストされているデフォルトのユーザー名とは異なるユーザー名を使用している場合は、ユーザーにインスタンスへの書き込み許可を付与します。

8. [キーファイル] テキストボックスの横で、[参照] を選択します。

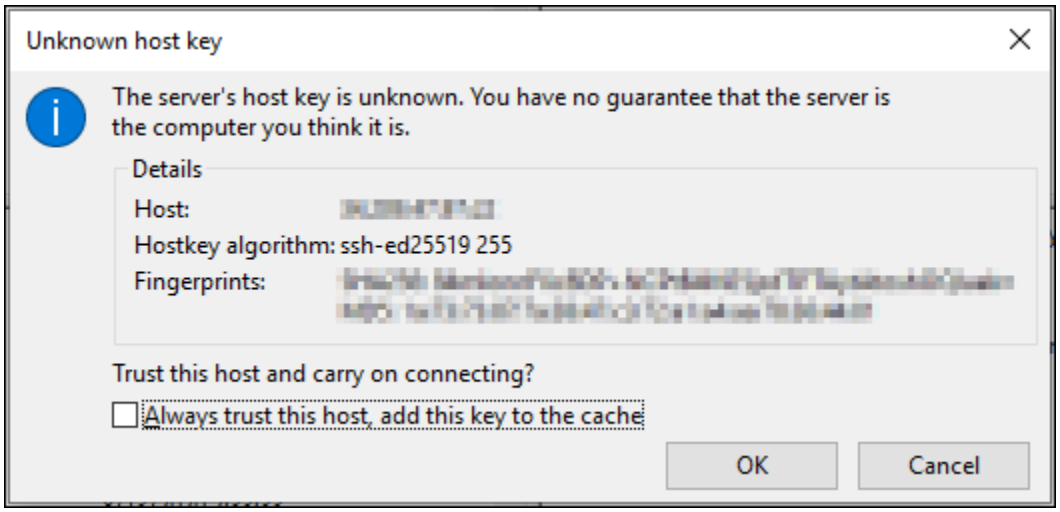


9. この手順で前に、Lightsail コンソールからダウンロードしたプライベートキーを探し、[開く] を選択します。

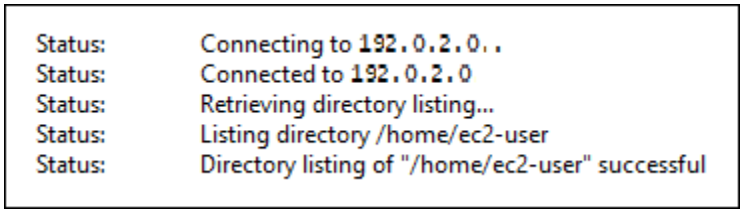
   > **Note**
   >
   > Windows を使用している場合は、pem ファイルを検索する際に既定のファイルの種類を [すべてのファイル] に変更します。



10. 接続 を選択します。
11. 以下の例のようなプロンプトが表示され、ホストキーが不明であることが分かります。[OK] を選択してプロンプトを確認し、インスタンスに接続します。



次の例のようなステータスメッセージが表示されている場合は、正常に接続されています。



ローカルコンピュータとインスタンス間でファイルを転送する方法を含む FileZilla の使用に関する詳細については、「FileZilla Wiki ページ」を参照してください。

RDP を使用した Lightsail Windows インスタンスへの接続

Lightsail コンソールでブラウザベースの RDP クライアントを使用して、Amazon Lightsail から Windows Server インスタンスに接続できます。ブラウザベースの RDP クライアントはソフトウェアのインストールを必要とせず、Windows Server インスタンスにその作成後すぐに接続でき、その

インスタンスを使用できます。インスタンスに接続し、ソフトウェアのインストールやウェブアプリケーションの設定などの管理タスクをサーバーで実行します。

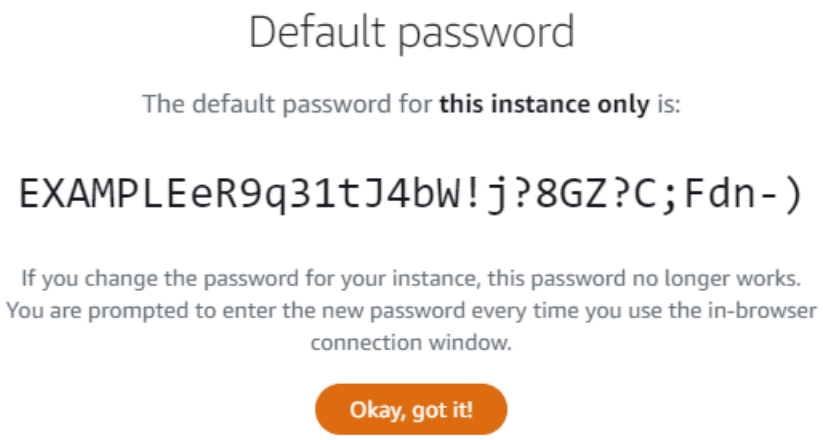
独自の RDP クライアントを使用して、Windows にバンドルされているリモートデスクトップ接続などのインスタンスに接続することもできます。独自の RDP クライアントの設定の詳細については、「リモートデスクトップ接続クライアントを使用しての Windows インスタンスへの接続」を参照してください。Lightsail の Linux または UNIX インスタンスに接続するには、「Linux または Unix インスタンスに接続する」を参照してください。

Windows Server インスタンスのデフォルトの管理者パスワード

ランダムに生成されたデフォルトの管理者パスワードは、Windows Server インスタンスにその作成時に割り当てられます。Lightsail コンソールのブラウザベースの RDP クライアントは、デフォルトの管理者パスワードを使用してインスタンスにサインインします。インスタンスの管理者パスワードを変更した場合、ブラウザベースの RDP クライアントを使用してインスタンスに接続しようとするたびに、新しいパスワードを手動で入力するように求められます。新しい管理者パスワードは、Lightsail によって保存されないため、インスタンスから取得できません。

> **Important**
>
> 管理者パスワードを紛失した場合、インスタンスにサインインできなくなり、パスワードをリセットできなくなります。新しい管理者パスワードは、 AWS Secrets Manager など、紛失した場合に後で取得できる安全な場所に保存します。詳細については、AWS Secrets Manager 「 ユーザーガイド」を参照してください。

管理者パスワードを元のデフォルトの管理者パスワードに変更することで、ブラウザベースの RDP クライアントを使用してインスタンスにアクセスするたびに、パスワードの入力を求められないようにできます。元のデフォルトの管理者パスワードは、Lightsail ホームページ にある [インスタンス] タブを選択すると確認できます。以下の例に示すように、Windows Server インスタンスの名前を選択し、[Connect (接続)] タブを選択し、[デフォルトのパスワードを表示] を選択して、元のデフォルトの管理者パスワードを表示します。

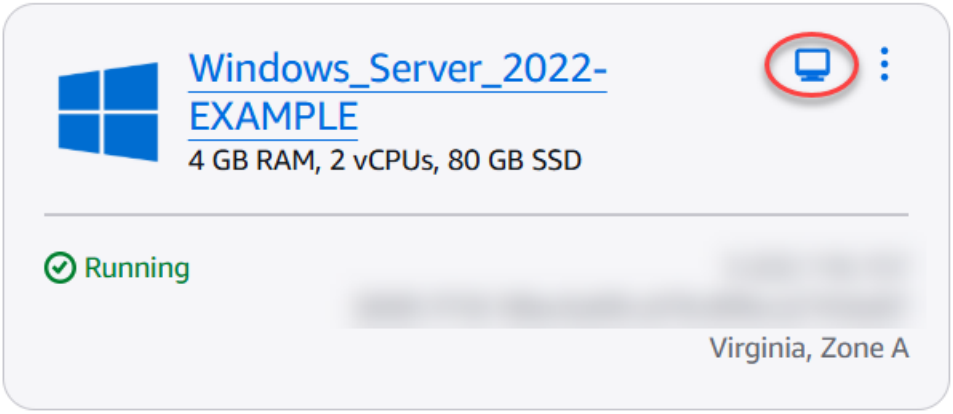


ブラウザベースの RDP クライアントを使用して Windows Server インスタンスに接続する

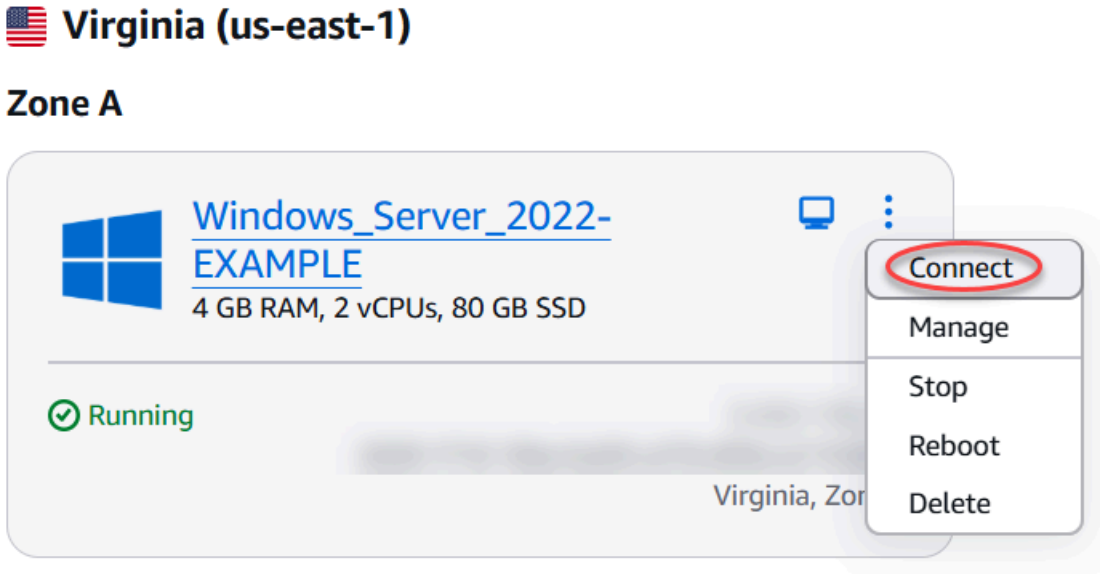
以下の手順に従って、Lightsail コンソールでブラウザベースの RDP クライアントを使用して Windows Server インスタンスに接続します。

1. Lightsail コンソール にサインインします。
2. 接続先のインスタンスのブラウザベースの RDP クライアントにアクセスするには、以下のいずれかのステップを実行します。

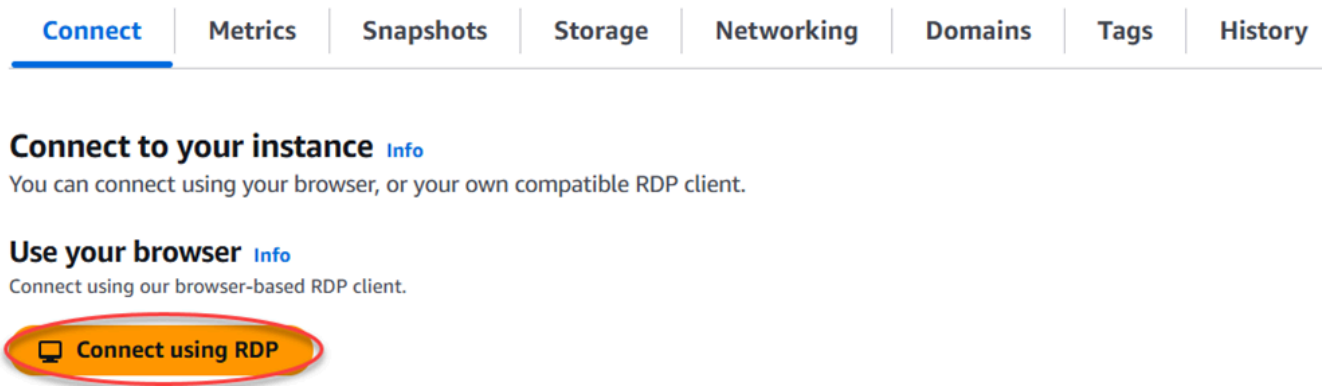
   - 以下の例に示すように、ブラウザベースの RDP クライアントアイコンを選択します。



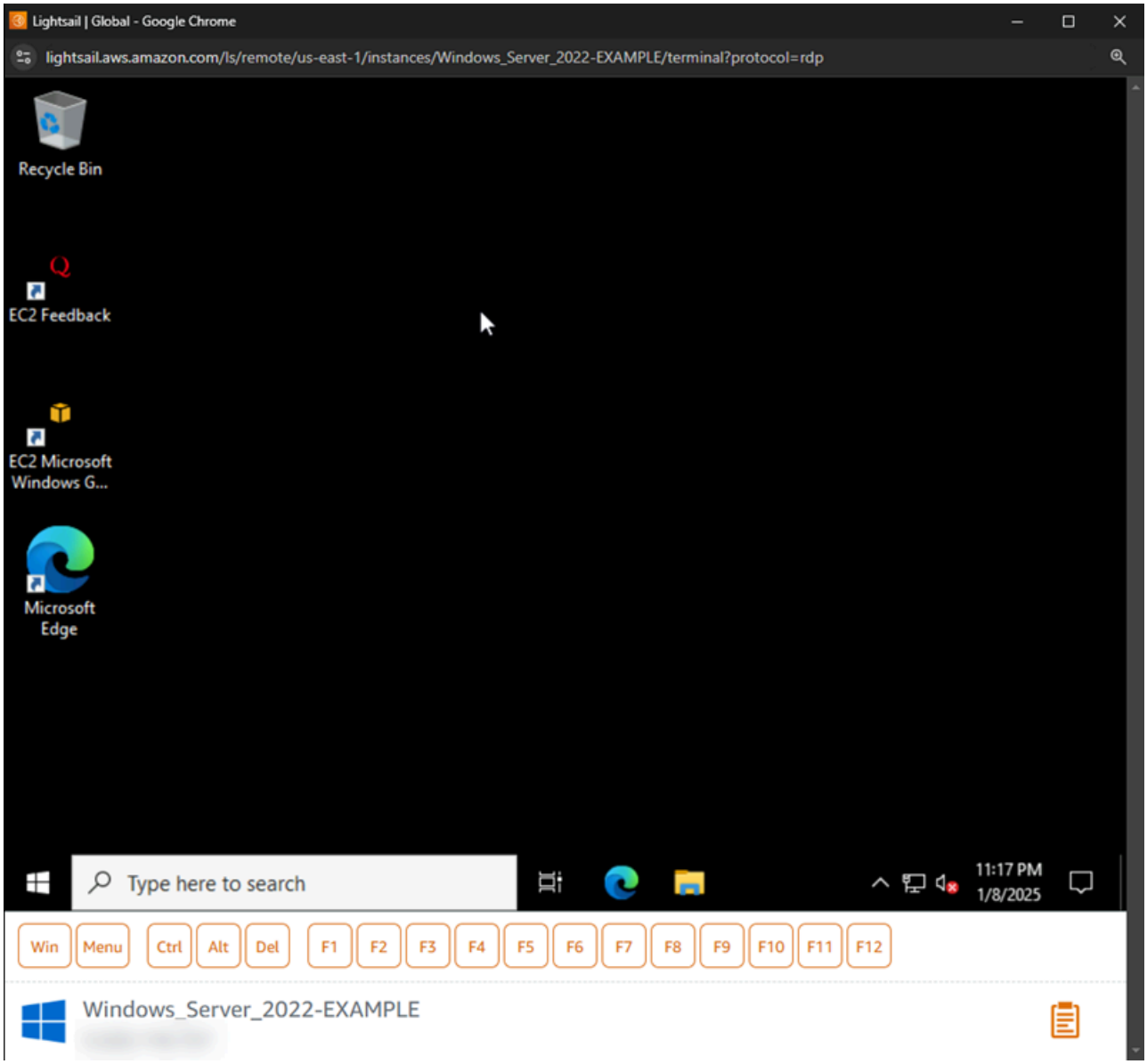
   - 以下の例に示すように、アクションメニューアイコン (⋮) を選択してから、[接続] を選択します。



- インスタンスの名前を選択し、[接続] タブの [RDP を使用して接続] を選択します。



ブラウザベースの RDP クライアントが開いて、以下の例に示すような Windows デスクトップが表示されると、インスタンスとのやり取りを開始できます。



> **Note**
>
> [接続] タブには、独自の RDP クライアントを使用して接続するために必要な情報 (Windows インスタンスのデフォルトのユーザー名やパスワードなど) も表示されます。独自の RDP クライアントの設定の詳細については、「リモートデスクトップ接続クライアントを使用した Amazon Lightsail での Windows インスタンスへの接続」を参照してください。

ブラウザベースの RDP クライアントを使用して Windows インスタンスとやり取りする

ブラウザベースの RDP クライアントは、独自のローカル Windows デスクトップと同じように使用します。RDP には、インスタンスとのやり取りに役立つ Windows 固有のキー (ファンクションキーなど) が含まれています。以下のセクションでは、RDP でクリップボードに (またはクリップボードから) テキストをコピーして貼り付ける方法を示します。

ブラウザベースの RDP クライアントにテキストを貼り付けるには

1. ローカルデスクトップのテキストを強調表示し、Ctrl+C または Cmd+C を押してテキストをローカルクリップボードにコピーします。
2. ブラウザベースの RDP クライアントの右下で、クリップボードアイコンを選択します。ブラウザベースの RDP クライアントのクリップボードテキストボックスが表示されます。
3. テキストボックス内をクリックし、Ctrl+V または Cmd+V を押して、ローカルクリップボードの内容をブラウザベースの RDP クライアントのクリップボードに貼り付けます。
4. リモートデスクトップ画面の任意の領域を右クリックし、ブラウザベースの RDP クライアントのクリップボードからリモートデスクトップ画面にテキストを貼り付けます。

ブラウザベースの RDP クライアントからテキストをコピーするには

1. リモートデスクトップ画面でテキストを強調表示します。
2. ブラウザベースの RDP クライアントの右下で、クリップボードアイコンを選択します。ブラウザベースの RDP クライアントのクリップボードテキストボックスが表示されます。
3. コピーするテキストを強調表示し、Ctrl+C または Cmd+C を押してテキストをローカルクリップボードにコピーします。これで、コピーしたテキストをローカルデスクトップの任意の場所に貼り付けることができます。

Lightsail Windows インスタンスの管理者パスワードの変更

Windows Server ベースのLightsailインスタンスを作成すると、インスタンスを作成する AWS リージョン のデフォルトのパスワードが使用されます。これにより、ブラウザベースのリモートデスクトップ (RDP) クライアントと、リモートデスクトップ接続などのクライアントを使用して接続しやすくなります。

> **Important**
>
> Lightsail がインスタンスのパスワードを生成できるようにすることを強くお勧めします。カスタムパスワードは保存されないため、管理者パスワードを変更した場合、Lightsail インスタンスへのアクセスが失われるリスクがあります。

Windows Server を使用して管理者パスワードを変更する

Windows Server の [パスワードの変更] ツールを使用して管理者パスワードを変更できます。Windows Server ベースの Lightsail インスタンスで `Ctrl + Alt + Del` を押し、[パスワードの変更] を選択します。

を使用してLightsailキーペアの暗号文を取得する AWS CLI

Windows Server ベースのLightsailインスタンスでパスワードを変更する場合は、 AWS Command Line Interface (AWS CLI) を使用して、パスワードの復号に役立つ情報を取得できます。

> **Note**
>
> Lightsail には、.pem ファイルを操作するためのユーティリティはありません。プライベートキーファイルの形式変換する必要がある場合、OpenSSL (Linux 用) や base64 (Windows 用) などの無料オープンソースツールを容易に利用できます。

暗号文を取得する

1. まだ AWS CLIをインストールして設定していない場合は、インストールして設定します。

   詳細については、「Amazon Lightsail で使用するために AWS Command Line Interface を設定する」を参照してください。
2. コマンドプロントまたはターミナルを開きます。
3. 次のコマンドを入力します。

   ```
   aws lightsail get-instance-access-details --instance-name my-instance
   ```

   ここで、*my-instance* は情報を取得するインスタンスの名前です。

   以下のような出力結果が表示されるはずです。

```
{
    "accessDetails": {
        "username": "Administrator",
        "protocol": "rdp",
        "ipAddress": "12.345.678.910",
        "passwordData": {
            "ciphertext": "cipher",
            "keyPairName": "my-ohio-key"
        },
        "password": "",
        "instanceName": "2016-ohio-windows"
    }
}
```

4. 暗号化テキストは、使用可能な任意のアプリケーションでパスワードの復号化に使用できます。

Remote Desktop を使用して Windows から Lightsail Windows インスタンスに接続する

Windows オペレーティングシステムに含まれているリモートデスクトップ接続 (RDC) クライアントを使用して、Amazon Lightsail の Windows インスタンスに接続できます。RDC では、Windows インスタンスの管理者ユーザー名とパスワードを使用する必要があります。この場合のパスワードは、インスタンスの作成時にインスタンスに割り当てられるデフォルトのパスワード、またはデフォルトのパスワードを変更した場合は自分のパスワードです。

このトピックでは、Lightsail コンソールからデフォルトの管理者パスワードを取得し、Windows インスタンスに接続するように RDC を設定するステップについて説明します。ブラウザを使用して、Lightsail コンソールからインスタンスに接続することもできます。詳細については、「ウェブベースの RDP クライアントを使用して Windows インスタンスに接続する」を参照してください。

Windows インスタンスのデフォルトの管理者パスワードを取得する

次のステップに従って、RDC を使用してインスタンスに接続するために必要な Windows インスタンスのデフォルトの管理者パスワードを取得します。

> **Note**
>
> デフォルトの管理者パスワードを変更した場合、インスタンスの Lightsail コンソールに表示されるパスワードは機能しません。パスワードは覚えておく必要があります。RDC で管理者パスワードを使用せずにインスタンスに接続することはできません。

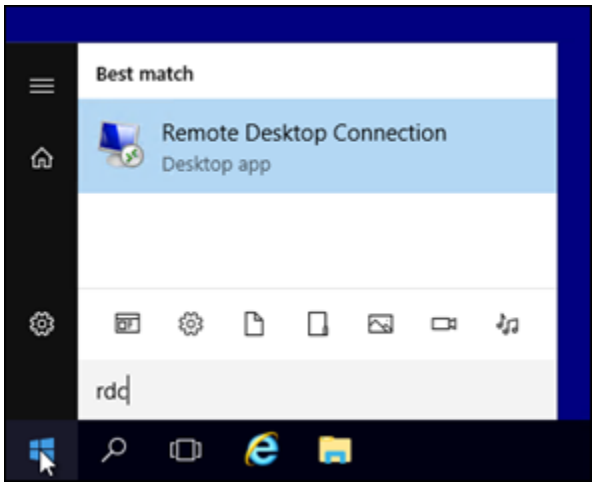
1. Lightsail コンソール にサインインします。
2. 接続先の Windows インスタンスを選択します。
3. インスタンス管理ページの [接続] タブで、[デフォルトのパスワードを表示] を選択します。
4. 表示されるデフォルトのパスワードをハイライト表示し、[Ctrl+C] または [Cmd+C] を押してコピーします。これで、パスワードがクリップボードにコピーされます。

   このガイドの次のセクションに進んで RDC を設定し、パスワードをクライアントに貼り付けます。

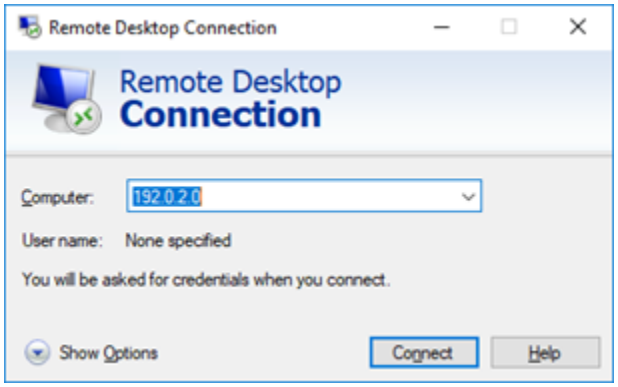
RDC を設定し、Windows インスタンスに接続する

以下のステップに従って、RDC を設定し、Windows インスタンスに接続します。

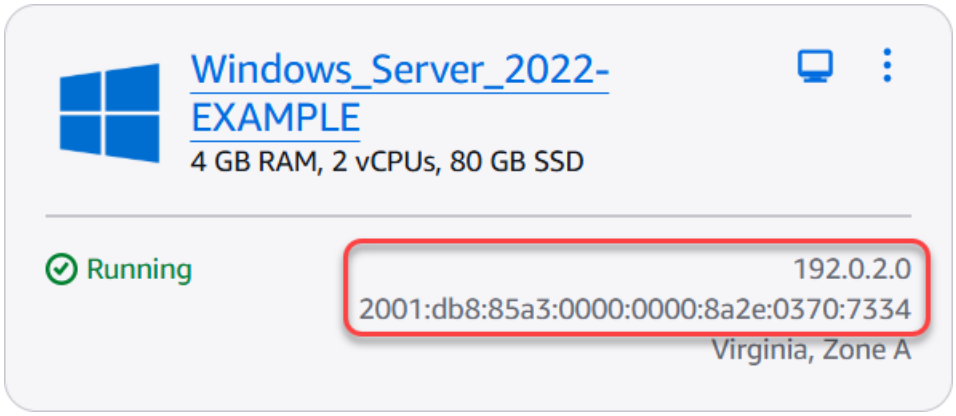
1. Windows メニューを開き、`Remote Desktop Connection` または `RDC` を検索します。
2. 検索結果の [リモートデスクトップ接続] を選択します。



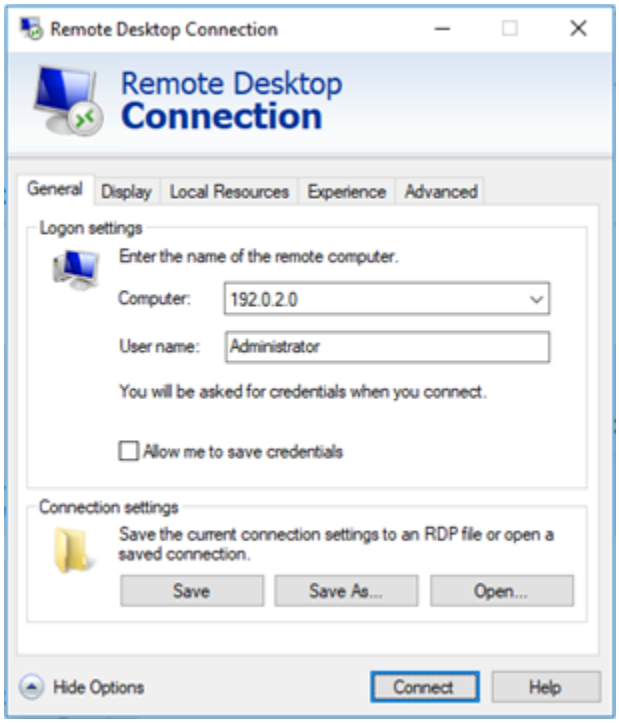
3. [コンピュータ] テキストボックスに、Windows インスタンスのパブリック IP アドレスを入力します。



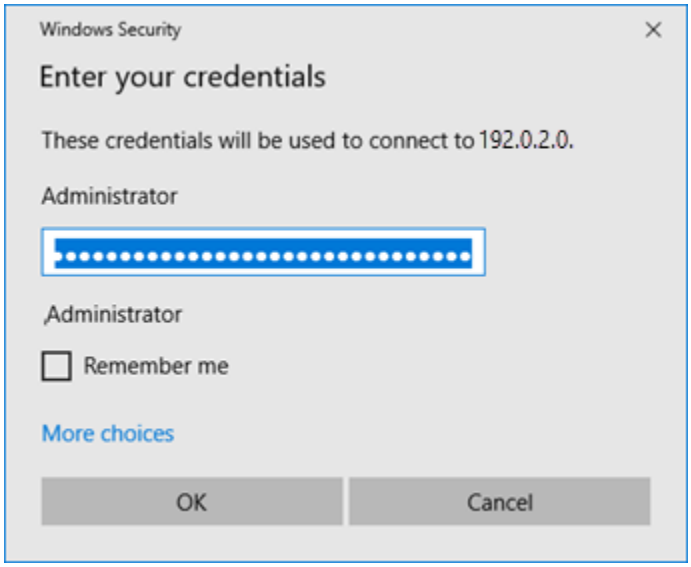
次の例に示すように、パブリック IP は、Lightsail コンソールのインスタンスの横に表示されます。



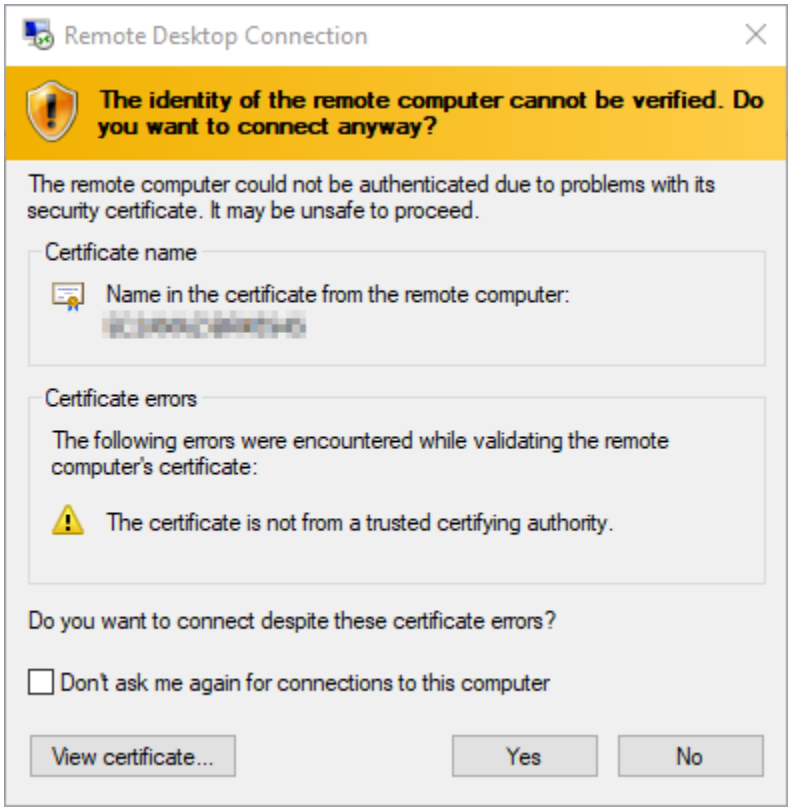
4. [オプションの表示] を選択して追加の接続オプションを表示します。
5. [User Name] (ユーザー名) テキストボックスに `Administrator` を入力します。これは、Lightsail 内のすべての Windows インスタンスのデフォルトユーザー名です。



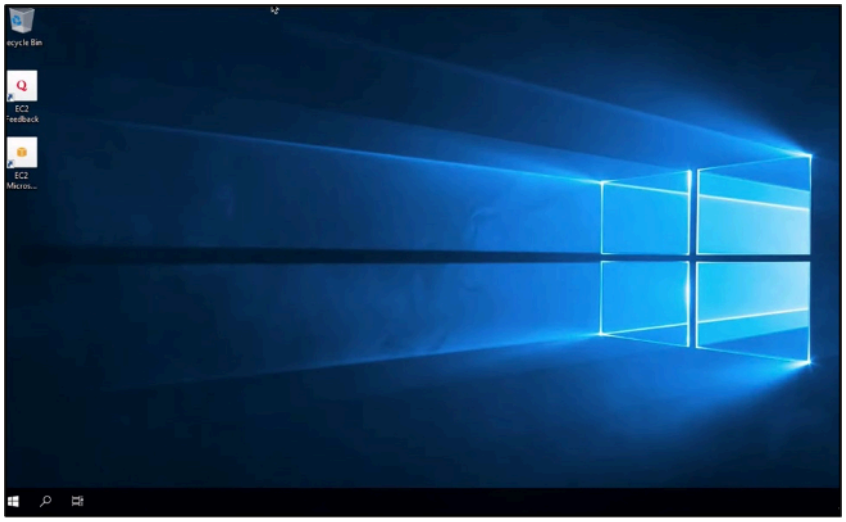
6. 接続 を選択します。
7. 表示されるプロンプトで、この手順の前半で Lightsail コンソールからコピーしたデフォルトの管理者パスワードを入力するか貼り付け、[OK] を選択します。



8. 表示されるプロンプトで、[はい] を選択して Windows インスタンスに接続します。証明書エラーが出ても無視します。



インスタンスに接続されると、次の例のような画面が表示されます。



Remote Desktop を使用して macOS から Lightsail Windows インスタンスに接続する

Microsoft リモートデスクトップクライアントを使用して、macOS コンピュータから Windows インスタンスに接続することができます。Microsoft リモートデスクトップでは、Lightsail Windows インスタンスの管理者ユーザー名とパスワードの使用が求められます。パスワードは、インスタンスの作

成時に割り当てられたデフォルトのパスワード、またはデフォルトのパスワードを変更した場合は独自のパスワードを使用できます。

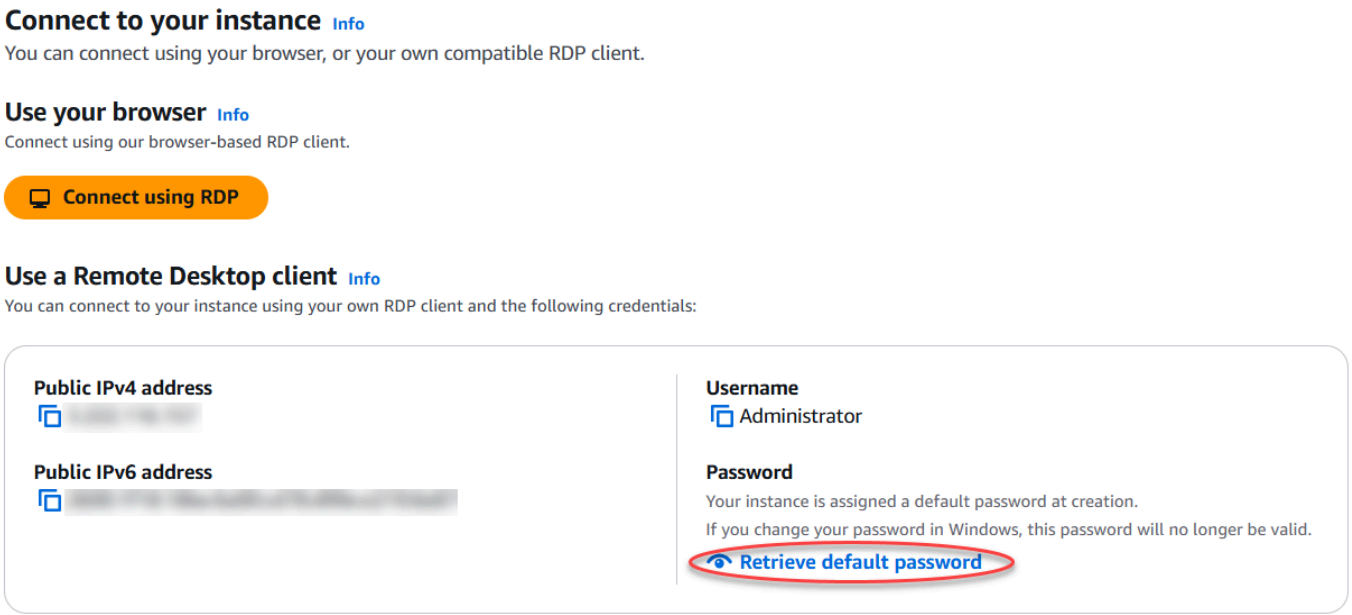
このトピックでは、Lightsail コンソールからデフォルトの管理者パスワードを取得して、Windows インスタンスに接続するように Microsoft リモートデスクトップを設定する手順を詳しく説明します。ブラウザを使用して、Lightsail コンソールからインスタンスに接続することもできます。詳細については、「Microsoft リモートデスクトップクライアントを使用して Windows インスタンスに接続する」を参照してください。

Windows インスタンスについて必要な接続情報を取得する

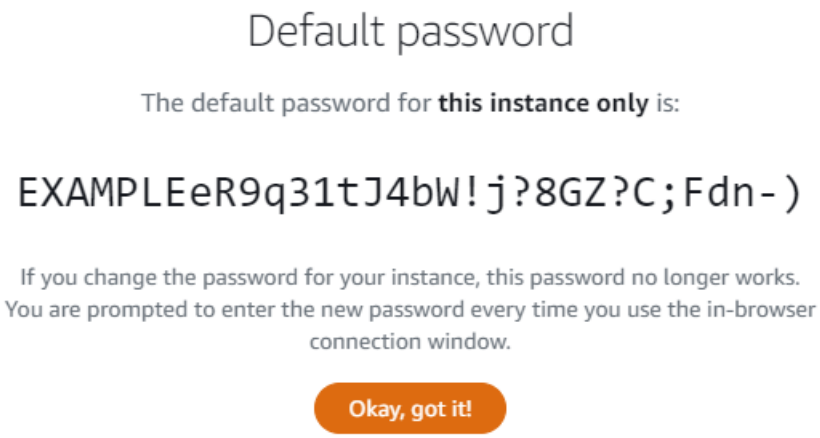
Microsoft リモートデスクトップクライアントを使用して Windows インスタンスに接続するには、そのインスタンスのパブリック IP アドレス、ユーザー名、および管理者パスワードが必要になります。

必要な情報を取得するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページで [インスタンス] セクションを選択します。
3. 接続するインスタンスのパブリック IP アドレスをメモします。
4. 接続するインスタンスの名前を選択します。
5. [Connect] (接続) タブを選択します。
6. [Show default password] (デフォルトのパスワードを表示) をクリックして、インスタンスの Windows 管理者パスワードを取得します。



プロンプトに Windows インスタンスのデフォルト管理者パスワードが表示されます。

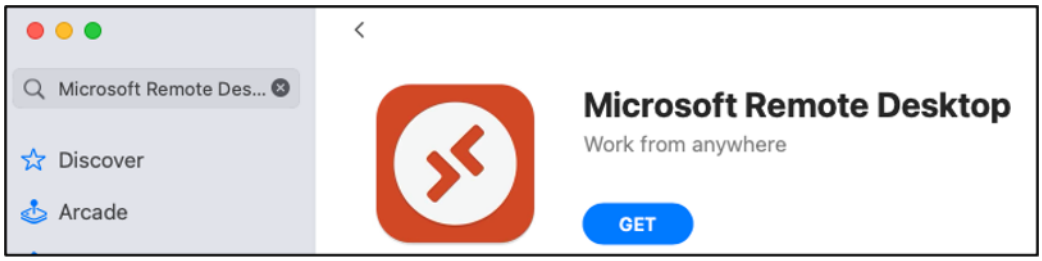


7. 管理者パスワードをコピーします。これは、本ガイドの後半で Microsoft リモートデスクトップクライアントを使用してインスタンスにサインインするために使用します。

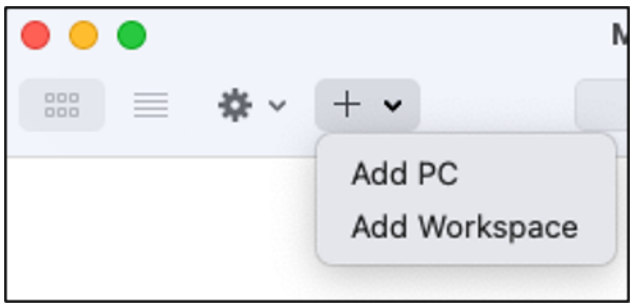
Microsoft リモートデスクトップを設定してインスタンスに接続する

Mac に Microsoft リモートデスクトップクライアントをインストールして、インスタンスに接続するようにクライアントを設定するには、以下の手順を実行します。

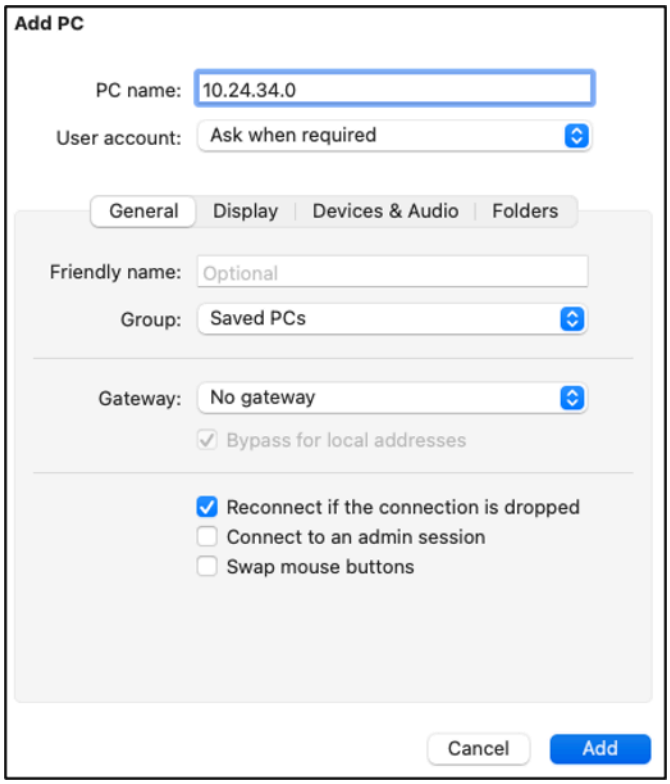
1. Mac で App Store を開き、[Microsoft Remote Desktop] (Microsoft リモートデスクトップ) を検索します。
2. 検索結果で [Microsoft Remote Desktop] (Microsoft リモートデスクトップ) を見つけ、[GET] (入手) をクリックしてアプリケーションをインストールします。



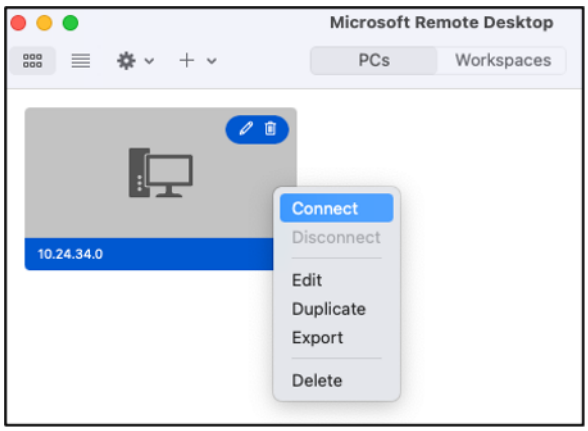
3. インストールが完了したら、[Microsoft Remote Desktop] (Microsoft リモートデスクトップ) を開きます。
4. 上部で [+] アイコンを選択し、[PC の追加] を選択します。



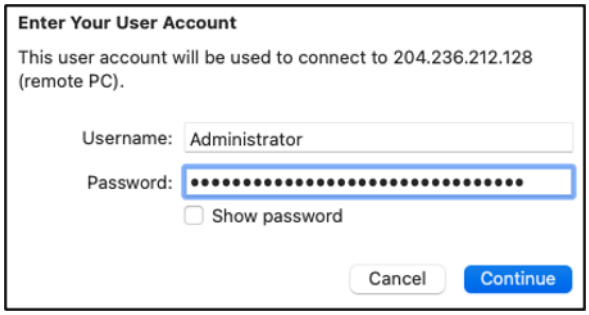
5. [PC name] (PC 名) テキストボックスに、インスタンスのパブリック IP アドレスを貼り付けます。
6. [Add] (追加) を選選択します。



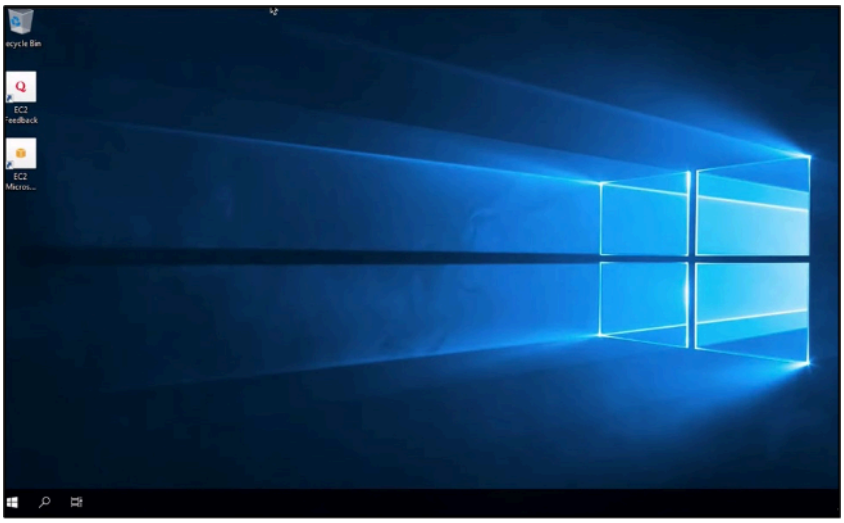
7. インスタンスのアイコンを右クリックし、[Connect] (接続) を選択します。



8. [Username]] (ユーザーネーム) テキストボックスに [Administrator (管理者) と入力し、[Password] (パスワード) テキストボックスに本ガイドで先ほど取得したデフォルトの管理者パスワードを入力します。
9. [Connect] (接続) をクリックしてインスタンスに接続します。



これで、Lightsail Windows インスタンスに接続されました。



Lightsail におけるインスタンスメタデータサービス (IMDS) とユーザーデータをアクセスする

インスタンスメタデータはインスタンスに関するデータで、実行中のインスタンスを設定または管理するために使用します。インスタンスメタデータは、ホスト名、イベント、およびセキュリティグループなどでカテゴリ分けされます。インスタンスメタデータを使用して、インスタンスの起動時に指定したユーザーデータにアクセスすることもできます。例えば、インスタンスを設定するためにパラメータを指定したり、単純なスクリプトを含めたりすることができます。インスタンスには、インスタンスの起動時に生成されるインスタンスアイデンティティドキュメントなどの動的データも含まれます。

> **Important**
>
> インスタンスメタデータおよびユーザーデータにはそのインスタンス自体内からのみアクセスできるものの、データは認証または暗号化手法によって保護されていません。インスタンス、そしてインスタンス上で実行される任意のソフトウェアに対して直接アクセス権がある可能性がある人はメタデータを表示できます。そのため、パスワードまたは存続期間の長い暗号化キーなどの機密データはユーザーデータとして保管しないようにしてください。

Instance Metadata Service を使う

次のいずれかのメソッドを使って、Lightsail 内の実行中のインスタンスからインスタンスメタデータにアクセスできます。

- インスタンスメタデータサービスバージョン 1 (IMDSv1) – リクエスト/レスポンスメソッド
- インスタンスメタデータサービスバージョン 2 (IMDSv2) – セッション指向メソッド

> **Important**
>
> Lightsail のすべてのインスタンスブループリントが IMDSv2 をサポートされているわけではありません。`MetadataNoToken` インスタンスメトリクスは、IMDSv1 を使用しているインスタンスメタデータサービスへの呼び出しの数を追跡します。詳細については、「インスタンスのメトリクスを表示する」を参照してください。

IDMS の詳細については、「インスタンスメタデータサービス (IMDS) の設定」を参照してください。

IMDS 関連の追加のドキュメント

次の IMDS ドキュメントは、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」と「Windows インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」で利用できます。

> **Note**
>
> Amazon EC2 では、インスタンスのブループリントは Amazon マシンイメージ (AMIs) と呼ばれます。

- Linux インスタンスの場合:
  - インスタンスメタデータオプションの設定
  - インスタンスメタデータの取得
  - インスタンスユーザーデータの使用
  - 動的データの取得
  - インスタンスメタデータのカテゴリ
  - 例: AMI 作成インデックス値
  - インスタンスアイデンティティドキュメント
- Windows インスタンスの場合:
  - インスタンスメタデータオプションの設定
  - インスタンスメタデータの取得
  - インスタンスユーザーデータの使用
  - 動的データの取得
  - インスタンスメタデータのカテゴリ
  - 例: AMI 作成インデックス値
  - インスタンスアイデンティティドキュメント

Lightsail でインスタンスメタデータサービス (IMDS) をアクセスおよび設定する

次のいずれかのメソッドを使用して、実行中のインスタンスからインスタンスメタデータにアクセスできます。

- インスタンスメタデータサービスバージョン 1 (IMDSv1) – リクエスト/レスポンスメソッド
- インスタンスメタデータサービスバージョン 2 (IMDSv2) – セッション指向メソッド

> **Important**
>
> Lightsail のすべてのインスタンスブループリントが IMDSv2 をサポートされているわけではありません。`MetadataNoToken` インスタンスメトリクスは、IMDSv1 を使用しているインスタンスメタデータサービスへの呼び出しの数を追跡します。詳細については、「インスタンスのメトリクスを表示する」を参照してください。

デフォルトではIMDSv1またはIMDSv2のいずれか、あるいは両方を使用できます。インスタンスメタデータサービスは、IMDSv2 に固有の PUT ヘッダーまたは GET ヘッダーがリクエストに存在するかどうかに基づいて、IMDSv1 リクエストと IMDSv2 リクエ ストを区別します。詳細については、「EC2 Instance Metadata Service の拡張により、オープンファイアウォール、リバースプロキシ、および SSRF の脆弱性に対して多層防御を追加する」を参照してください。

ローカルコードまたはユーザーに IMDSv2を使用させるように、各インスタンスのインスタンスメタデータサービスを構成することができます。IMDSv2を使用しなければならないように指定すると、IMDSv1はもう機能しなくなります。詳細については、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「インスタンスのメタデータオプションを設定する」を参照してください。

インスタンスのメタデータを取得するには、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」の「インスタンスのメタデータの取得」を参照してください。

> **Note**
>
> このセクションの例では、インスタンスメタデータサービスの IPv4 アドレスを使用します (`169.254.169.254`)。IPv6 アドレスを使用してインスタンスのインスタンスメタデータを

取得する場合は、IPv6 アドレスを有効にして使用してください (`fd00:ec2::254`)。インスタンスメタデータサービスの IPv6 アドレスは、IMDSv2 コマンドと互換性があります。

インスタンスメタデータサービスバージョン 2 の仕組み

IMDSv2 は、セッション指向リクエストを使用します。セッション指向リクエストを使用して、セッション期間 (1 秒～6 時間) を定義するセッショントークンを作成します。指定した期間中、それ以降のリクエストに同じセッショントークンを使用できます。指定した期間が期限切れになった後、将来のリクエストに使用する新しいセッショントークンを作成する必要があります。

Important

Amazon Linux 2023 および Ubuntu 24 から起動された Lightsail インスタンスには、デフォルトで IMDSv2 が設定されます。

次の例では、LinuxPowerShell シェルスクリプトと IMDSv2 を使用して、最上位インスタンスメタデータアイテムを取得しています。これらの例では、以下のことを行います。

- PUT リクエストを使用して、6 時間 (21,600 秒) のセッショントークンを作成する
- セッショントークンヘッダーを TOKEN (Linux の場合) または `token` (Windows の場合) という名前の変数に保管する
- トークンを使用して最上位メタデータアイテムをリクエストする

次のコマンドを使用してインストールして起動します。

- Linux の場合:
  - 最初に、次のコマンドを使用してトークンを生成します。

    ```
    [ec2-user ~]$ TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`
    ```

  - その後、次のコマンドを使用して、トークンを使用して上位レベルのメタデータアイテムを生成します。

    ```
    [ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/
    ```

- Windows の場合:
  - 最初に、次のコマンドを使用してトークンを生成します。

```
PS C:\> [string]$token = Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token-
ttl-seconds" = "21600"} -Method PUT -Uri http://169.254.169.254/latest/api/token
```

  - その後、次のコマンドを使用して、トークンを使用して上位レベルのメタデータアイテムを生成します。

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} -Method
 GET -Uri http://169.254.169.254/latest/meta-data/
```

トークンを作成した後、期限切れになるまで再使用することができます。次の例では、各コマンドはインスタンスの起動に使用されるブループリント (Amazon マシンイメージ (AMI)) の ID を取得します。前の例のトークンは再利用されます。$TOKEN (Linux) または $token (Windows) に保管されます。

- Linux の場合:

```
[ec2-user ~]$ curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/
latest/meta-data/ami-id
```

- Windows の場合:

```
PS C:\> Invoke-RestMethod -Headers @{"X-aws-ec2-metadata-token" = $token} `
-Method GET -uri http://169.254.169.254/latest/meta-data/ami-id
```

IMDSv2 を使用してインスタンスメタデータをリクエストする際は、リクエストに次の項目が含まれている必要があります。

- **PUT** リクエスト – PUT リクエストを使用して、インスタンスメタデータサービスに対してセッションを開始します。PUT リクエストは、インスタンスメタデータサービスに対する後続の GET リクエストに含まれている必要のあるトークンを返します。このトークンは、IMDSv2 を使用する際、メタデータにアクセスするのに必要です。
- トークン – トークンを、インスタンスメタデータサービスに対するすべての GET リクエストに含めます。トークンの使用が required に設定されている場合、有効なトークンがないリクエスト、または有効期限切れのトークンを持つリクエストで 401 - Unauthorized HTTP エラー

コードが発生します。トークン使用要件の変更に関する情報については、「AWS CLI コマンドリファレンス」の「インスタンスのメタデータのオプションの更新」を参照してください。

- トークンはインスタンス固有のキーです。トークンは他のインスタンスで有効ではなく、生成されたインスタンスの外で使用しようとすると拒否されます。
- PUT リクエストには、トークンの有効期限 (TTL) を秒単位で指定するヘッダーが含まれている必要があります。TTL は最大 6 時間 (21,600 秒) まで指定できます。トークンは論理的セッションを表します。TTL はトークンが有効な時間の長さ、つまりセッションの期間を指定します。
- トークンの期限が切れた後、インスタンスメタデータにアクセスし続けるためには、別の PUT リクエストを使用して新しいセッションを作成する必要があります。
- 各リクエストについてトークンを再使用するか、あるいは新しいトークンを作成することを選択できます。少数のリクエストでは、インスタンスメタデータサービスにアクセスする必要があるたびに、トークンを生成してすぐに使用するほうが簡単かもしれません。ただし、効率を重視するなら、インスタンスメタデータをリクエストする必要があるたびに PUT リクエストを書くのではなく、トークン期間を長く指定して再使用することができます。それぞれが独自のセッションを表すトークンを同時に使用できる数については、実質的な制限はありません。ただし、IMDSv2 では、通常のインスタンスメタデータサービス接続とスロットリングの制限によって制約を受けます。詳細については、「Linux インスタンス向け Amazon Elastic Compute Cloud ユーザーガイド」の「クエリスロットリング」を参照してください。

HTTP GET および HEAD メソッドは IMDSv2 インスタンスメタデータリクエストで許可されています。PUT リクエストは、X-Forwarded-For ヘッダーが含まれている場合、拒否されます。

デフォルトで、PUT リクエストに対するレスポンスには IP プロトコルレベルで 1 のレスポンスホップリミット (有効期限) があります。より大きなホップリミットが必要な場合は、update-instance-metadata-options コマンドを使用して調整できます。例えば、インスタンスで実行されているコンテナサービスとの下位互換性のためにホップリミットを拡大する必要があるかもしれません。詳細については、「AWS CLI コマンドリファレンス」の「update-instance-metadata-options」を参照してください。

インスタンスメタデータサービスバージョン 2 の使用への移行

インスタンスメタデータサービスバージョン 2 (IMDSv2) の使用は任意です。インスタンスメタデータサービスバージョン 1 (IMDSv1) は、終了の期限なく引き続きサポートされます。IMDSv2 の使用に移行する場合、次のツールと移行パスを使用することが推奨されます。

IMDSv2 への移行に役立つツール

お使いのソフトウェアで IMDSv1 が使用されている場合、次のツールを使用して IMDSv2 を使用するようソフトウェアを再構成することができます。

- AWS ソフトウェア: AWS SDKs の最新バージョンと は IMDSv2 AWS CLI をサポートしています。IMDSv2 を使用するには、インスタンスに最新バージョンの AWS SDKs と があることを確認します AWS CLI。の更新の詳細については AWS CLI、 AWS Command Line Interface ユーザーガイドの「 のインストール、更新、アンインストール AWS CLI」を参照してください。すべての Amazon Linux 2 ソフトウェアパッケージが IMDSv2 をサポートしています。
- インスタンスのメトリクス: IMDSv2 はトークンベースのセッションを使用しますが、IMDSv1 は使用しません。`MetadataNoToken` インスタンスのメトリクスは、IMDSv1 を使用しているインスタンスメタデータサービスへの呼び出しの数を追跡します。このメトリクスをゼロまでトラッキングすることにより、すべてのソフトウェアが IMDSv2 を使用するようアップグレードされたかどうか、およびいつアップデートが行われたかを測定できます。詳細については、「Amazon Lightsail でインスタンスのメトリクスを表示する」を参照してください。
- Lightsail API オペレーションと AWS CLI コマンドの更新: 既存のインスタンスでは、update-instance-metadata-options AWS CLI コマンド (または UpdateInstanceMetadataOptions API オペレーション) を使用してIMDSv2 の使用を要求できます。コマンドの例を次に示します。*`InstanceName`* をインスタンスの名前に置き換え、*`RegionName`* をインスタンスの に置き換え AWS リージョン てください。

```
aws lightsail update-instance-metadata-options --region RegionName --instance-
name InstanceName --http-tokens required
```

IMDSv2 アクセスを必要とする推奨パス

前述のツールを使用し、IMDSv2 への移行にこのパスに従うことを推奨します。

ステップ 1: 開始時

インスタンスでロール認証情報を使用する AWS SDKs AWS CLI、、およびソフトウェアを IMDSv2-compatibleバージョンに更新します。の更新の詳細については AWS CLI、「 AWS Command Line Interface ユーザーガイド」の「 の最新バージョンへのアップグレード AWS CLI」を参照してください。

次に、IMDSv2 リクエストを使用して、インスタンスメタデータに直接アクセスする (つまり、 AWS SDK を使用しない) ソフトウェアを変更します。

ステップ 2: 移行中

MetadataNoToken のインスタンスメトリクスを使用して、移行の進行状況を追跡します。このメトリクスは、インスタンスで IMDSv1 を使用しているインスタンスメタデータサービスへの呼び出しの数を示します。詳細については、「インスタンスのメトリクスを表示する」を参照してください。

ステップ 3: すべてのインスタンスですべての準備が完了した時点

インスタンスのメトリクス MetadataNoToken が IMDSv1 の使用ゼロを記録した時点で、すべてのインスタンスにおいてすべての準備が完了します。この段階で、インスタンスのメタデータのオプションの更新 コマンドから IMDSv2 の使用をリクエストできます。実行中のインスタンスでこれらの変更を行うことができます。インスタンスを再起動する必要はありません。

既存のインスタンスのインスタンスメタデータオプションの更新は、Lightsail API または AWS CLI を介してのみ使用できます。現在のところ、Lightsail コンソールでは使用できません。詳細については、「update-instance-metadata-options」を参照してください。

IMDS 関連の追加のドキュメント

次の IMDS ドキュメントは、「Linux インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」と「Windows インスタンス用 Amazon Elastic Compute Cloud ユーザーガイド」で利用できます。

> **Note**
>
> Amazon EC2 では、インスタンスのブループリントは Amazon マシンイメージ (AMIs) と呼ばれます。

- Linux インスタンスの場合:
  - インスタンスメタデータオプションの設定
  - インスタンスメタデータの取得
  - インスタンスユーザーデータの使用
  - 動的データの取得
  - インスタンスメタデータのカテゴリ
  - 例: AMI 作成インデックス値
  - インスタンスアイデンティティドキュメント

- Windows インスタンスの場合:
  - インスタンスメタデータオプションの設定
  - インスタンスメタデータの取得
  - インスタンスユーザーデータの使用
  - 動的データの取得
  - インスタンスメタデータのカテゴリ
  - 例: AMI 作成インデックス値
  - インスタンスアイデンティティドキュメント

Lightsail ブロックストレージディスクでストレージとパフォーマンスを拡張する

システムディスクでは、ワークロードの実行に必要な安定していて低レイテンシーのパフォーマンスが提供されます。Lightsail ディスクを使用すると、使用量の拡張または縮小を分単位で行うことができます。プロビジョニングしているサイズに合わせて、低料金でご利用いただけます。

Linux/Unix ベースまたは Windows Server ベースのインスタンスでは、最大 80 GB のシステムディスクを選択できます。「Lightsail で Linux ベースのインスタンスの使用を開始する」または「Windows Server ベースのインスタンスの使用を開始する」を参照してください。

追加のブロックストレージディスクを作成することで、仮想プライベートサーバーにストレージをさらに追加することもできます。「ブロックストレージディスクの作成と Linux ベースのインスタンスへのアタッチ」または「ブロックストレージディスクの作成と Windows Server インスタンスへのアタッチ」を参照してください。

ブロックストレージディスク

ブロックストレージは、データを「ブロック」として管理するストレージアーキテクチャです。各ストレージブロック (Lightsail では「ディスク」とも呼ばれます) は、サーバーにアタッチ可能な個々のハードディスクと同様に機能します。通常、特定のデータをコアサービスから分離し、インスタンスやブートストレージディスクで障害や他の問題が発生した場合にアプリケーションデータを保護する必要があるアプリケーションまたはソフトウェアに追加のブロックストレージを使用できます。

Lightsail では、ブロックストレージにソリッドステートドライブ (SSD) が使用されます。このタイプのブロックストレージは、リーズナブルな料金と良好なパフォーマンスのバランスが取れています。Lightsail で実行される大半のワークロードをサポートするためのものです。Lightsail の追加のブロックストレージディスクにより、保存データに頻繁にアクセスするアプリケーションやソフトウェアに必要な一貫したパフォーマンスと低レイテンシーが実現します。

> **Note**
>
> 一定の IOPS パフォーマンスまたはディスクあたりの高いスループットを必要とするアプリケーションや、MongoDB、Cassandra などの大規模データベースを実行するお客様は、Lightsail の代わりに GP2 やプロビジョンド IOPS SSD ストレージを搭載する Amazon EC2 を使用することをお勧めします。

「Amazon EBS ボリューム」の詳細は、「Amazon EC2 ユーザーガイド」でご覧いただけます。

ディスククォータ

- リージョンあたり 20,000 GB。
- ディスクあたり最大 16 TB、またはディスクあたり最小 8 GB。
- インスタンスあたり最大で 15 個までのアタッチされたディスクおよび 1 個のブートボリュームディスクを保持できます。

Lightsail ブロックストレージディスクを作成して Linux ベースのインスタンスにアタッチする

Amazon Lightsail インスタンス用に追加のブロックストレージディスクを作成してアタッチすることができます。追加ディスクを作成したら、Linux/Unix ベースの Lightsail インスタンスに接続してディスクをフォーマットおよびマウントする必要があります。

このトピックでは、Lightsail を使用して新しいディスクを作成し、アタッチする方法について説明します。さらに、アタッチされたディスクをフォーマットしてマウントできるように、SSH を使用して Linux/Unix ベースのインスタンスに接続する方法についても説明します。

Windows Server ベースのインスタンスを使用している場合は、代わりに「ブロックストレージディスクを作成して Windows Server インスタンスにアタッチする」を参照してください。

ステップ 1: 新しいディスクを作成してインスタンスにアタッチする

1. 左側のナビゲーションペインで [ストレージ] を選択します。
2. [ディスクの作成] を選択します。
3. Lightsail インスタンスが配置されている AWS リージョン とアベイラビリティーゾーンを選択します。
4. サイズを選択します。
5. ディスクの名前を入力します。

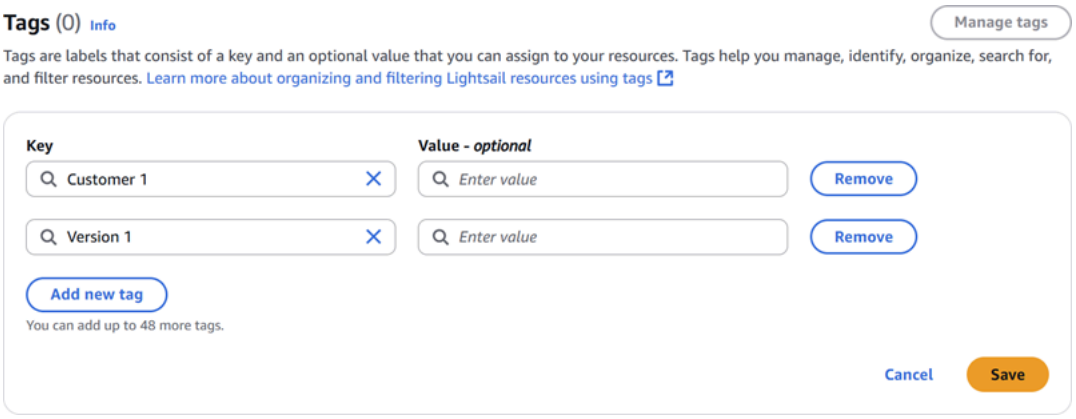
   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。

- 先頭と末尾は英数字または数字を使用する必要があります。
- 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

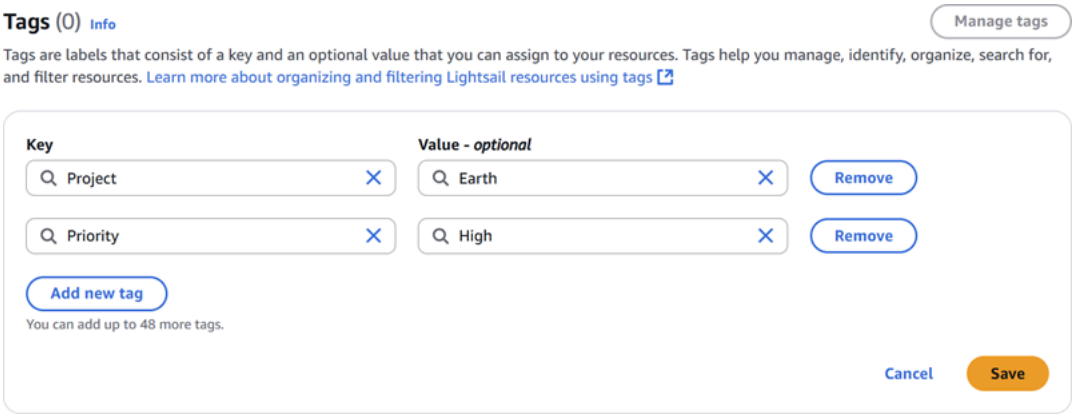
6. 以下のいずれかのオプションを選択して、ディスクにタグを追加します。

- [キーオンリータグの追加] または [タグを編集] (タグが追加済みの場合)。タグキーのテキストボックスに新しいタグを入力し、Enter キーを押します。タグの入力を完了したら、[保存] を選択してタグを追加し、追加しない場合は、[キャンセル] を選択します。



- [Create a key-value tag] (キーバリュータグを作成) から [Key] (キー) テキストボックスにキーを入力し、[Value] (値) テキストボックスに値を入力します。タグの入力を完了したら、[保存] を選択し、追加しない場合は、[キャンセル] を選択します。

  キーバリューのタグは、保存する際に一つずつ追加することができます。さらにキーバリュータグを追加するには、以上のステップを繰り返します。



> **Note**
>
> 「キーのみ」のタグと「キーバリュー」のタグの詳細については、「タグ」を参照してください。

7. [ディスクの作成] を選択します。

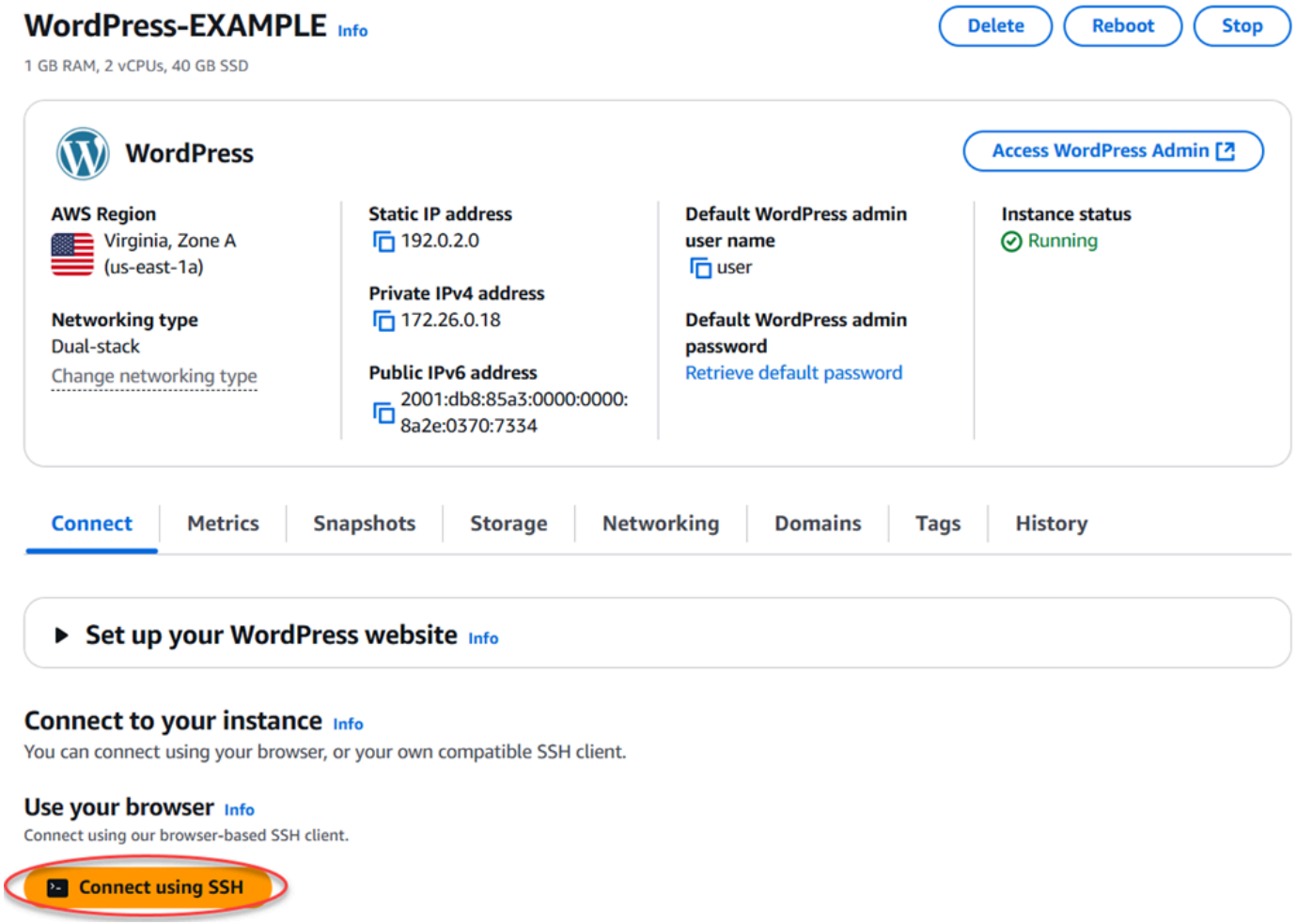
   数秒後、ディスクが作成され、新しいディスク管理ページが表示されます。

8. リストからインスタンスを選択し、[アタッチ] を選択して、新しいディスクをインスタンスにアタッチします。

ステップ 2: インスタンスに接続し、ディスクをフォーマットしてマウントする

1. ディスクを作成してアタッチしたら、Lightsail でインスタンス管理ページに戻ります。

   デフォルトでは、[接続] タブが表示されます。



2. [SSH を使用して接続] を選択してインスタンスに接続します。
3. ターミナルウィンドウで、以下のコマンドを入力します:

```
lsblk
```

`lsblk` の出力は、ディスクパスからプレフィックス `/dev/` を略します。

> **Note**
>
> 2023 年 6 月 29 日に、Lightsail インスタンスの基盤となるハードウェアを更新しました。次の例では、前世代インスタンスのデバイス名は `/dev/xvda` として表示されます。この日付以降に作成されたインスタンスのデバイス名は `/dev/nvme0n1` として表示されます。

Current generation instances

次の出力例では、ルートボリューム (nvme0n1) には 2 つのパーティション (nvme0n1p1 および nvme0n1p128) がありますが、追加のボリューム (nvme1n1) にはパーティションがありません。

```
[ec2-user ~]$ sudo lsblk
NAME          MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
nvme1n1       259:0    0  30G  0 disk /data
nvme0n1       259:1    0  16G  0 disk
##nvme0n1p1   259:2    0   8G  0 part /
##nvme0n1p128 259:3    0   1M  0 part
```

Previous generation instances

次の出力例では、ルートボリューム (xvda) にはパーティション (xvda1) がありますが、追加のボリューム (xvdf) にはパーティションがありません。

```
[ec2-user ~]$ sudo lsblk
NAME    MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda    202:0    0  16G  0 disk
##xvda1 202:1    0   8G  0 part /
xvdf    202:80   0  24G  0 disk
```

4. ディスクにファイルシステムを作成する必要があるかどうかを確認します。新しいディスクは未加工のブロックデバイスであるため、マウントして使用する前に、ボリュームにファイルシステムを作成する必要があります。スナップショットから復元されたディスクには、多くの場合既にファイルシステムがあります。既存のファイルシステムの上に新しいファイルシステムを作成した場合、データが上書きされます。

   ディスクにファイルシステムがあるかどうかを確認するには、以下を使用します。ディスクにファイルシステムがない場合は、[ステップ 2.5] に進みます。ディスクにファイルシステムがある場合は、[ステップ 2.6] に進みます。

   Current generation instances

   ```
   sudo file -s /dev/nvme1n1
   ```

   新しいディスクでは、次のような出力が表示されます。

```
/dev/nvme1n1: data
```

次のような出力が表示される場合、ディスクに既にファイルシステムがあることを意味します。

```
/dev/nvme1n1: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)
```

Previous generation instances

```
sudo file -s /dev/xvdf
```

新しいディスクでは、次のような出力が表示されます。

```
/dev/xvdf: data
```

次のような出力が表示される場合、ディスクに既にファイルシステムがあることを意味します。

```
/dev/xvda1: Linux rev 1.0 ext4 filesystem data, UUID=1701d228-e1bd-4094-
a14c-12345EXAMPLE (needs journal recovery) (extents) (large files) (huge files)
```

5. 次のコマンドを使用して、ディスクに新しいファイルシステムを作成します。*device_name* をデバイス名 (例: /dev/nvme1n1) に置き換えます。アプリケーションの要件またはオペレーティングシステムの制限に応じて、ext3、ext4 など、異なるファイルシステムの種類を選択できます。

> **Important**
>
> この手順では、空のディスクをマウントすることを前提としています。既にデータが含まれるディスク (スナップショットから復元したディスクなど) をマウントする場合は、ディスクのマウント前に mkfs を使用しないでください。代わりに、この手順の[ステップ 2.6] に進んでマウントポイントを作成します。そうでない場合、ディスクがフォーマットされ、既存のデータが削除されます。

Current generation instances

```
sudo mkfs -t xfs device_name
```

次のような出力が表示されます。

```
meta-data=/dev/nvme1n1           isize=512    agcount=16, agsize=1048576 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1
data     =                       bsize=4096   blocks=16777216, imaxpct=25
         =                       sunit=1      swidth=1 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1
log      =internal log           bsize=4096   blocks=16384, version=2
         =                       sectsz=512   sunit=1 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

Previous generation instances

```
sudo mkfs -t ext4 device_name
```

次のような出力が表示されます。

```
mke2fs 1.42.9 (4-Feb-2014)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
4194304 inodes, 16777216 blocks
838860 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
512 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
4096000, 7962624, 11239424
```

```
Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

6. 次のコマンドを使用して、ディスクのマウントポイントディレクトリを作成します。マウントポイントとは、ディスクをマウントした後、ファイルシステムツリー内でディスクが配置され、ファイルの読み書きが実行される場所です。*mount_point* の位置 (例: /data) を使用されていないスペースに置き換えます。

```
sudo mkdir mount_point
```

7. ディスクでファイルシステムが作成されたことを確認するには、次のコマンドを入力します。

   Current generation instances

   ```
   sudo file -s /dev/nvme1n1
   ```

   /dev/nvme1n1: data の代わりに、以下のような出力結果が表示されるはずです。

   ```
   /dev/nvme1n1: SGI XFS filesystem data (blksz 4096, inosz 512, v2 dirs)
   ```

   Previous generation instances

   ```
   sudo file -s /dev/xvdf
   ```

   /dev/xvdf: data の代わりに、以下のような出力結果が表示されるはずです。

   ```
   /dev/xvdf: Linux rev 1.0 ext4 filesystem data, UUID=0ee83fdf-e370-442e-
   ae38-12345EXAMPLE (extents) (large files) (huge files)
   ```

8. 最後に、次のコマンドを入力して、ディスクをマウントします。

   ```
   sudo mount device_name mount_point
   ```

   新しいディスクマウントのファイルのアクセス許可をプレビューして、ユーザーとアプリケーションがディスクに書き込みできることを確認します。ファイルのアクセス許可の詳細については、Amazon EC2 ユーザーガイドの「Making an Amazon EBS Volume Available for Use」を参照してください。

ステップ 3: インスタンスを再起動するたびにディスクをマウントする

通常は、Lightsail インスタンスを再起動するたびにこのディスクをマウントしてください。マウントしない場合、このステップは省略可能です。

1. システムブート時に常に、このディスクをマウントするには、/etc/fstab ファイルにデバイス用のエントリを追加します。

   /etc/fstab ファイルのバックアップコピーを作成すると、編集中に誤って破壊/削除してしまった場合にこのコピーを使用できます。

   ```
   sudo cp /etc/fstab /etc/fstab.orig
   ```

2. 任意のテキストエディタ (例: vim など) を使って /etc/fstab ファイルを開きます。

   変更を保存するには、ファイルを開く前に sudo と入力する必要があります。

3. 次のフォーマットを使って、ディスクのファイルの最後に新しい行を追加します。

   ```
   device_name  mount_point  file_system_type  fs_mntops  fs_freq  fs_passno
   ```

   たとえば、新しい行は以下のようになります。

   Current generation instances

   ```
   /dev/nvme1n1 /data xfs defaults,nofail 0 2
   ```

   Previous generation instances

   ```
   /dev/xvdf /data ext4 defaults,nofail 0 2
   ```

4. ファイルを保存し、テキストエディタを終了します。

Lightsail ブロックストレージディスクを作成して Windows Server インスタンスにアタッチする

ストレージ容量を追加する必要がある場合は、Amazon Lightsail でブロックストレージディスクを作成して Windows Server インスタンスにアタッチできます。ブロックストレージディスクの詳細については、「ブロックストレージディスク」を参照してください。

このガイドでは、Lightsail コンソールを使用して新しいブロックストレージディスクを作成し、Windows Server インスタンスにアタッチする方法について説明します。さらに、RDP を使用して Windows Server インスタンスに接続し、ディスクをオンラインにして初期化する方法についても説明します。

> **Note**
>
> Linux または Unix インスタンスを使用している場合は、「ディスクを作成して Linux または Unix インスタンスにアタッチする」を参照してください。

ステップ 1: 新しいブロックストレージディスクを作成してインスタンスにアタッチする

Amazon Lightsail コンソールを使用して新しいブロックストレージディスクを作成し、インスタンスにアタッチします。

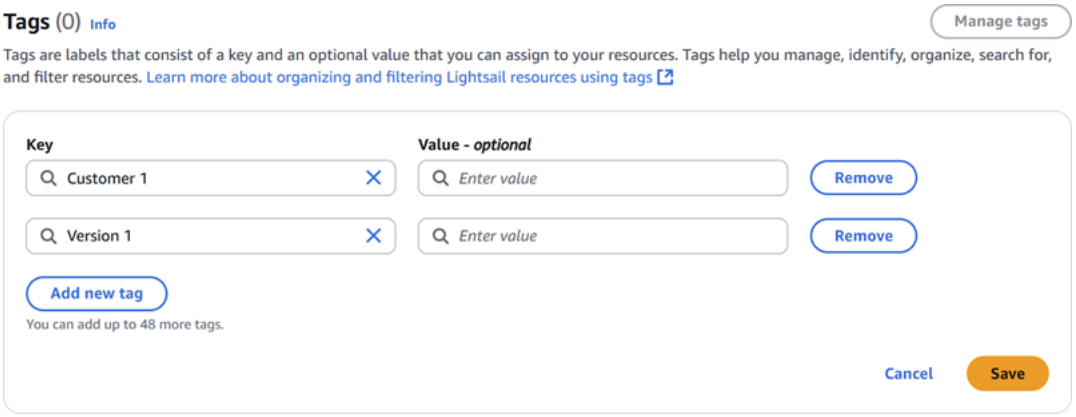
新しいブロックストレージディスクを作成してインスタンスにアタッチするには

1. Lightsail コンソール にサインインします。
2. [ストレージ] タブ、[ディスクの作成] の順に選択します。
3. Lightsail インスタンスが配置されている AWS リージョン とアベイラビリティーゾーンを選択します。
4. ディスクサイズを選択します。
5. ストレージディスクの名前を入力します。

   リソース名:

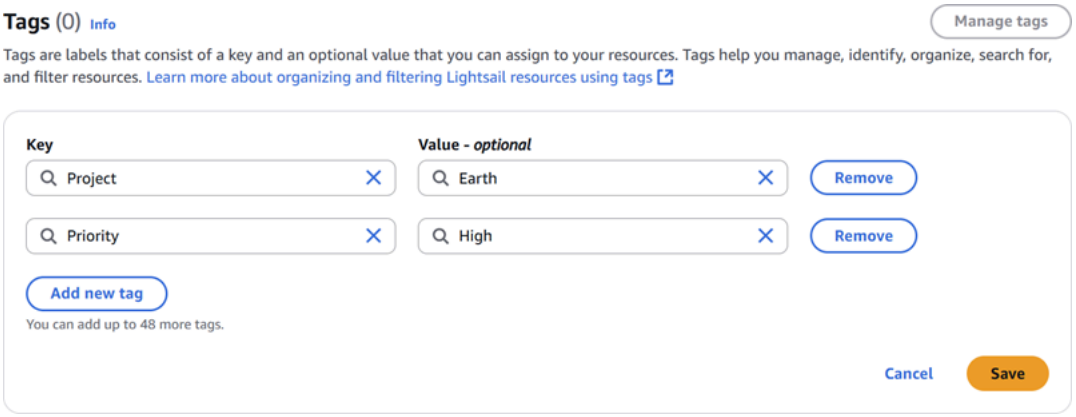
   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。
6. 以下のいずれかのオプションを選択して、ディスクにタグを追加します。

   - [キーオンリータグの追加] または [タグを編集] (タグが追加済みの場合)。タグキーのテキストボックスに新しいタグを入力し、Enter キーを押します。タグの入力を完了したら、[保存] を選択してタグを追加し、追加しない場合は、[キャンセル] を選択します。



- [Create a key-value tag] (キーバリュータグを作成) から [Key] (キー) テキストボックスにキーを入力し、[Value] (値) テキストボックスに値を入力します。タグの入力を完了したら、[保存] を選択し、追加しない場合は、[キャンセル] を選択します。

  キーバリューのタグは、保存する際に一つずつ追加することができます。さらにキーバリュータグを追加するには、以上のステップを繰り返します。



> **Note**
>
> 「キーのみ」のタグと「キーバリュー」のタグの詳細については、「タグ」を参照してください。

7. [ディスクの作成] を選択します。

   数秒後にディスクが作成され、新しいディスク管理ページでディスクの情報を確認できます。

8. リストからインスタンスを選択し、[アタッチ] を選択して、新しいディスクをインスタンスにアタッチします。

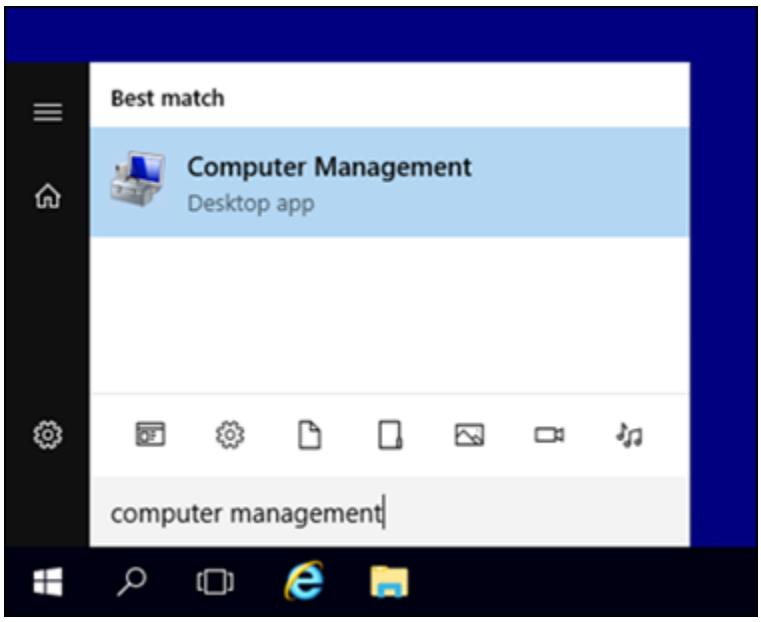
ブロックストレージディスクをオンラインにするには、このガイドの「ステップ 2: インスタンスに接続し、ブロックストレージディスクをオンラインにする」セクションに進みます。

ステップ 2: インスタンスに接続し、ブロックストレージディスクをオンラインにする

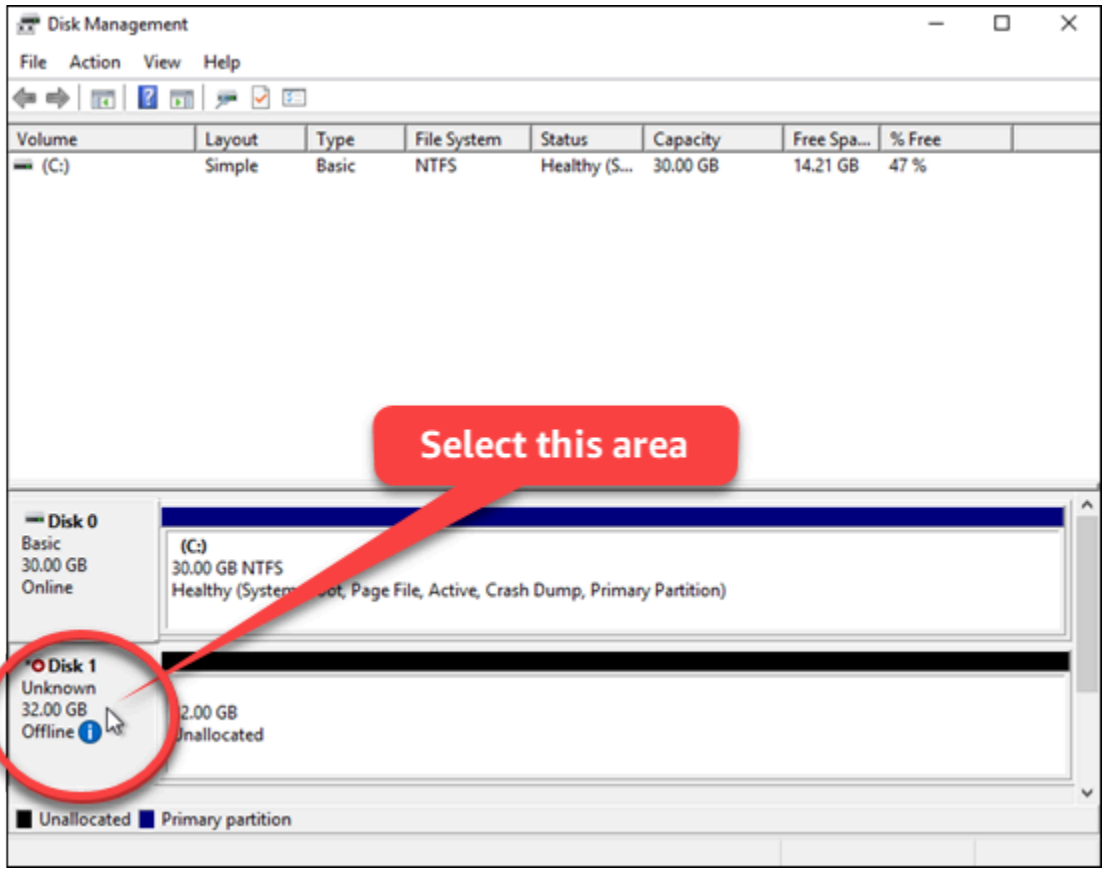
Windows Server インスタンスに接続し、ディスクの管理ユーティリティを使用して、前にアタッチしたブロックストレージディスクをオンラインにします。

インスタンスに接続してブロックストレージディスクをオンラインにするには

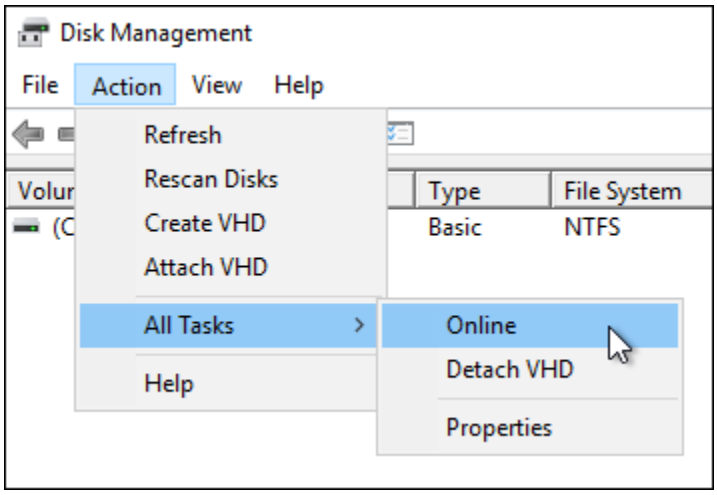
1. Lightsail コンソールのホームページに移動します。
2. このガイドで前に追加のストレージディスクをアタッチしたインスタンスの名前を選択します。
3. [接続] タブで、[RDP を使用して接続] を選択します。
4. Windows のスタートメニューで、コンピューターの管理を検索し、検索結果から [コンピューターの管理] を選択します。



5. [コンピューターの管理] の左側のペインで、[ディスクの管理] を選択します。
6. [ディスクの管理] ユーティリティの下部のペインで、[不明 / オフライン] というラベルが付いているディスクを選択します。これが、このガイドで前にインスタンスにアタッチしたブロックストレージディスクです。



7. ディスクを選択した状態で、[操作] メニューの [すべてのタスク] をポイントし、[オンライン] を選択します。



ブロックストレージディスクのステータスが [初期化されていません] に更新されるのがわかります。ブロックストレージディスクはまだオンラインになっていません。ブロックストレージディスクを初期化するには、このガイドの「ステップ 3: ブロックストレージディスクを初期化する」セクションに進みます。

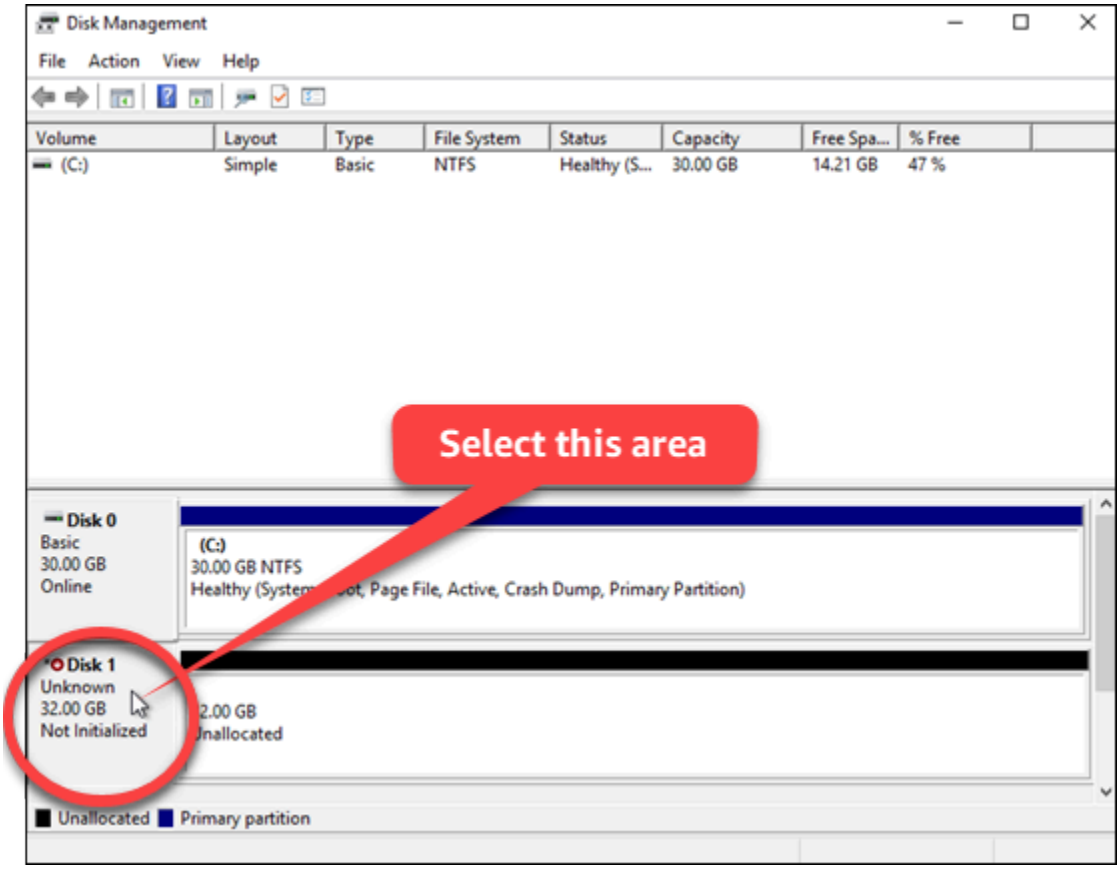
ステップ 3: ブロックストレージディスクを初期化する

フォーマットできるように、ブロックストレージディスクを初期化します。

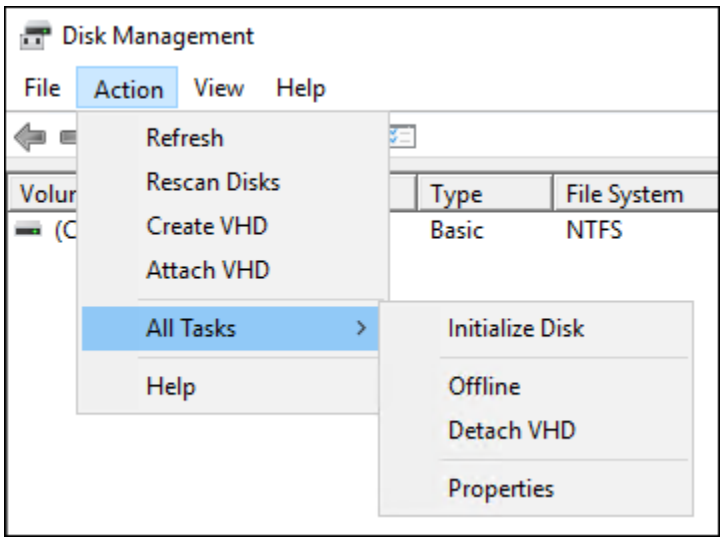
> **Important**
>
> すでにデータが含まれているディスク (スナップショットから作成したディスクなど) をマウントする場合は、ディスクを再フォーマットしないように注意してください。再フォーマットすると、既存のデータが削除されます。

ブロックストレージディスクを初期化するには

1. ディスクの管理ユーティリティの下部のペインで、[不明 / 初期化されていません] というラベルが付いているディスクを選択します。



2. ディスクを選択した状態で、[操作] メニューの [すべてのタスク] をポイントし、[ディスクの初期化] を選択します。



3. 新しいディスクのパーティションスタイルを選択し、[OK] を選択します。

> **Note**
>
> パーティションスタイルの詳細については、Microsoft の記事「About partition styles - GPT and MBR」を参照してください。

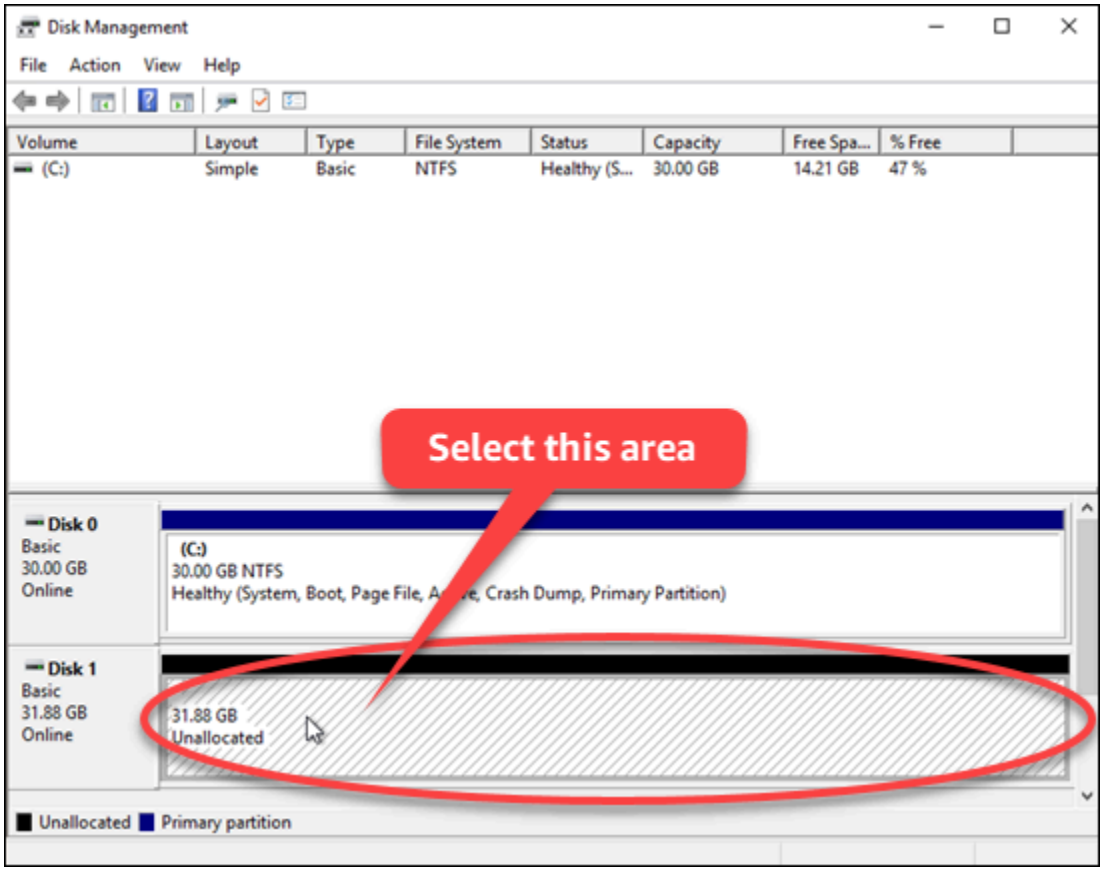
ブロックストレージディスクのステータスが [オンライン] に更新されるのがわかります。ファイルシステムでブロックストレージディスクをフォーマットにするには、このガイドの「ステップ 4: ディスクをファイルシステムでフォーマットする」セクションに進みます。

ステップ 4: ディスクをファイルシステムでフォーマットする

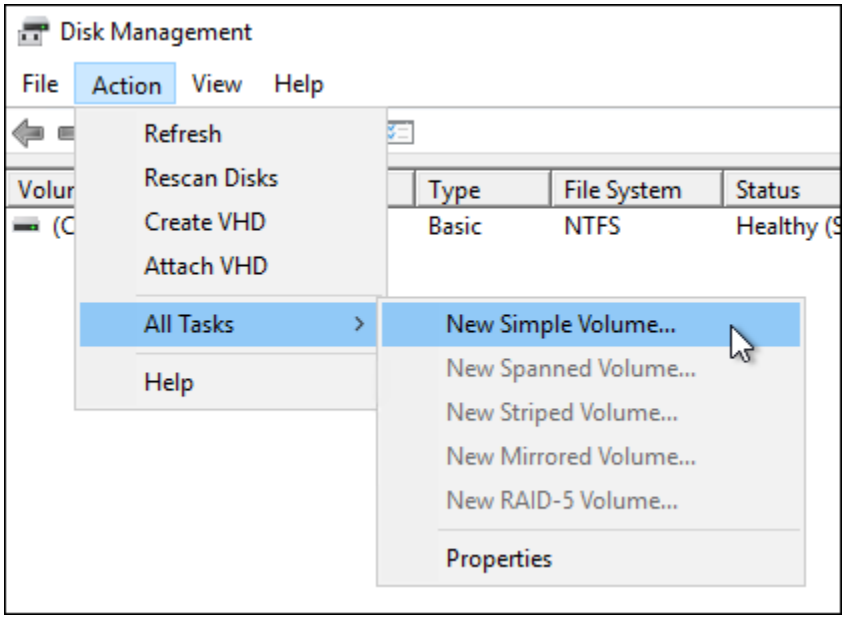
Windows Server の新しいシンプルボリュームウィザードを使用して、ドライブ文字を割り当て、ディスクをファイルシステムでフォーマットします。

ディスクをファイルシステムでフォーマットするには

1. ディスクの管理ユーティリティの下部のペインで、[未割り当て] というラベルが付いているブロックストレージディスクのパーティションを選択します。



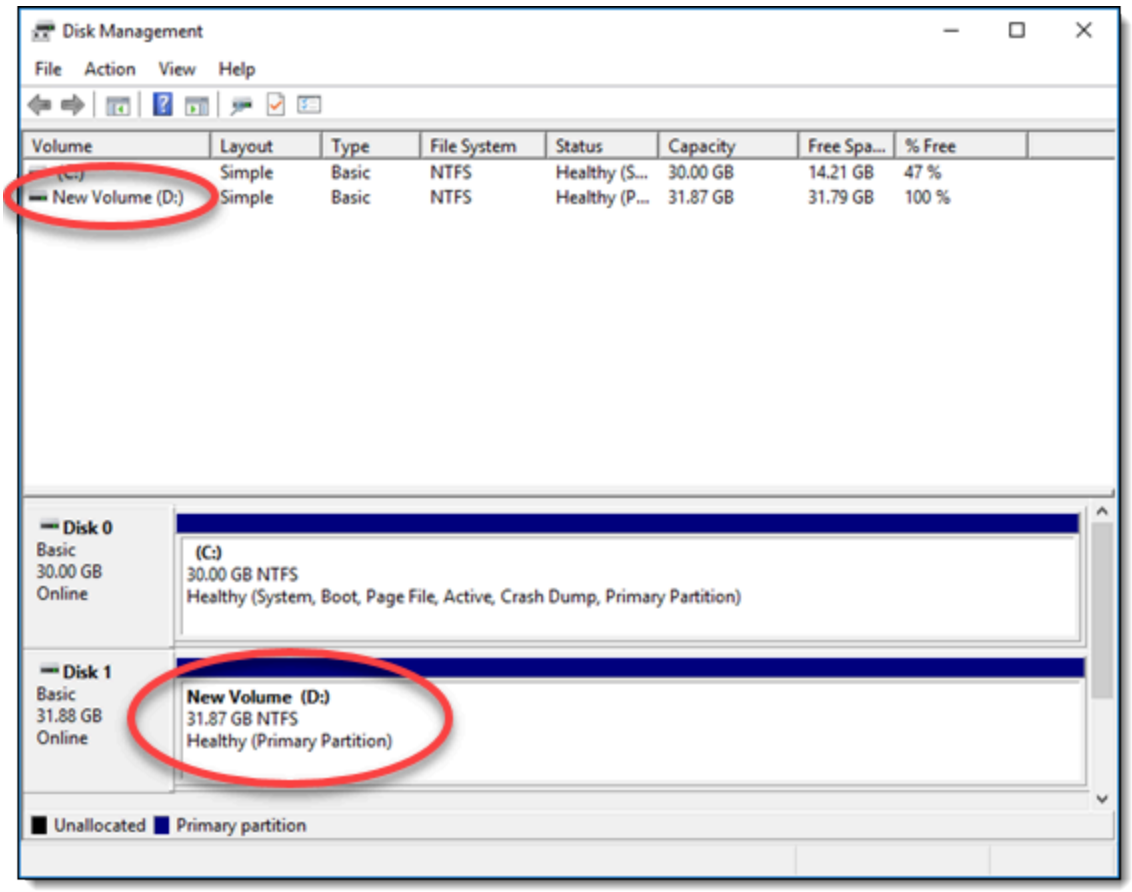
2. パーティションを選択した状態で、[アクション] メニューの [すべてのタスク] をポイントし、[新しいシンプルボリューム] を選択します。



3. 新しいシンプルボリュームウィザードの指示に従って、NTFS、FAT32、または ReFS ファイルシステムタイプを選択し、ディスクをフォーマットします。

> **Note**
>
> これらの各ファイルシステムの詳細については、Microsoft の記事「NTFS の概要」、「Resilient File System (ReFS) の概要」、および「FAT32 ファイル システムについて」を参照してください。

完了すると、ドライブ文字と次のメッセージがディスクの管理ユーティリティに表示されます。



Lightsail ブロックストレージディスクをデタッチおよび削除する

ブロックストレージディスクが不要になった場合、停止した Amazon Lightsail インスタンスからデタッチして削除できます。このトピックでは、データをバックアップしてディスクを安全に削除する方法について説明します。

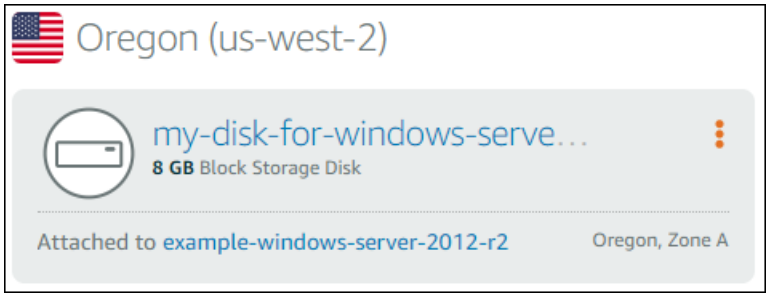
前提条件

- インスタンスの実行を停止します。これは、ディスクをデタッチして削除する前に実行する必要があります。インスタンスを停止する方法の詳細
- (オプション) ディスクのスナップショットを作成することをお勧めします。このようにして、状況が変わった場合もバックアップを利用できます。詳細については、「データベースのスナップショットを作成する」を参照してください。

ディスクをデタッチおよび削除する

Lightsail インスタンスを停止した後、ディスクを安全にデタッチして削除できます。

1. ホームページで [ストレージ] を選択します。
2. アタッチされたディスクの名前を選択して管理します。



3. ディスク管理ページで、[デタッチ] を選択します。

   数秒後、ディスクがデタッチされ、削除または再アタッチする準備ができます。
4. [削除] タブを選択します。
5. [ディスクの削除] を選択し、[はい、削除します] を選択して削除を確定します。

> **Important**
>
> これは永続的オペレーションで、取消すことはできません。削除するとディスク上のすべてのデータが失われます。

Amazon Lightsail のスナップショット

インスタンス、データベース、ブロックストレージディスクのポイントインタイムスナップショットを Amazon Lightsail で作成し、それらのスナップショットをベースラインとして使用して、新しいリソースを作成したり、データをバックアップしたりできます。スナップショットには、リソースの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。スナップショットからリソースを作成して復元すると、その新しいリソースはスナップショットの作成に使用された元のリソースの正確なレプリカとして始まります。Lightsail アカウント上のスナップショットには、スナップショットストレージ料金が請求されます (手動スナップショット、自動スナップショット、コピーされたスナップショット、システムディスクスナップショットのいずれか)。データの破損やディスク障害が発生した場合は、撮影したスナップショットからディスクを作成し、古いディスクを置き換えることができます。スナップショットを使用して新しいディスクをプロビジョニングし、新しいインスタンスの起動時にアタッチすることもできます。

目次

- 手動スナップショット
- 自動スナップショット
- システムディスクのスナップショット
- スナップショットからの新しいリソースの作成
- スナップショットをコピーする
- スナップショットを Amazon EC2 にエクスポートする
- スナップショットを削除する

手動スナップショット

インスタンス、マネージドデータベース、ブロックストレージディスクのスナップショットをいつでも手動で作成します。手動スナップショットは、削除するまで無期限に保存されます。

手動スナップショットの作成の詳細については、以下のガイドを参照してください。

- Linux または Unix インスタンスのスナップショットを作成する
- Windows Server インスタンスのスナップショットを作成する
- データベースのスナップショットを作成する
- ブロックストレージディスクのスナップショットを作成する

自動スナップショット

Lightsail インスタンスまたはブロックストレージディスクで重要な情報をホストしている場合は、手動スナップショットを作成して頻繁にバックアップする必要があります。ただし、管理タスクを頻繁に実行する時間を見つけるのが難しい場合があります。その場合は、自動スナップショットを使用して、手動での操作なしで、Lightsail によってインスタンスまたはブロックストレージディスクの毎日のバックアップが作成されるようにします。毎日 7 つの最新の自動スナップショットが保存されたあと、最も古いものから最新のものに置き換えられます。

自動スナップショットの詳細については、以下のガイドを参照してください。

- インスタンスの自動スナップショットを有効または無効にする
- インスタンスまたはディスクの自動スナップショット時間の変更
- 自動スナップショットを削除する

> **Important**
>
> ソースリソースを削除すると、リソースに関連付けられたすべての自動スナップショットが削除されます。この動作は、ソースリソースを削除した後でも Lightsail アカウントに保存される手動スナップショットとは異なります。ソースリソースを削除するときに自動スナップショットを保持するには、「自動スナップショットの保持」を参照してください。

システムディスクのスナップショット

インスタンスが応答しなくなり、システムディスク上のファイルにアクセスする必要がある場合は、インスタンスルートボリュームをバックアップするためにそのボリュームのスナップショットを作成できます。次に、スナップショットから新しいブロックストレージディスクを作成し、別のインスタンスにアタッチすることで、システムディスク内のファイルにアクセスできます。詳細については、「インスタンスルートボリュームのスナップショットを作成する」を参照してください。

スナップショットからの新しいリソースの作成

スナップショットを使用して、同じプラン、または元のリソースより大きなプランを使用している新しい Lightsail リソースを作成します。スナップショットを使用し、より小さい Lightsail プランを使用している新しいリソースを作成することはできません。スナップショットに基づいてリソースを作

成すると、新しいリソースは、スナップショットの作成に使用された元のリソースのレプリカとなります。

詳細については、以下のガイドを参照してください。

- スナップショットからのインスタンスの作成
- スナップショットからデータベースを作成する
- スナップショットから新しいブロックストレージディスクを作成する
- スナップショットからより大きなインスタンス、ブロックストレージディスク、またはデータベースを作成する

スナップショットをコピーする

インスタンスとブロックストレージディスクのスナップショットは、1 つの Amazon Web Services (AWS) リージョンから同じ Lightsail アカウント内の別のリージョンにコピーすることができます。データベーススナップショットをリージョン間でコピーすることはできません。詳細については、「Copy snapshots from one AWS リージョン to another」を参照してください。

のスナップショットを Amazon EC2 にエクスポートする

Lightsail は、 の使用を開始する最も簡単な方法です AWS。ただし、Amazon EC2 や他の AWS サービスにはLightsail存在しない の制限があります。Lightsail インスタンスとブロックストレージディスクのスナップショットを Amazon EC2 にエクスポートして、使用できるさまざまなインスタンスタイプを活用し、 AWSであらゆるサービスを使用します。詳細については、「スナップショットを Amazon EC2 にエクスポートする」を参照してください。

> **Note**
>
> cPanel & WHM (CentOS 7) インスタンスのスナップショットは Amazon EC2 にエクスポートできません。

スナップショットを削除する

不要になった Lightsail スナップショットは、毎月のスナップショットストレージ料金の発生を避けるために削除してください。詳細については、「スナップショットを削除する」を参照してください。

Lightsail インスタンスとディスクの自動スナップショットを設定する

インスタンスまたはブロックストレージディスクの自動スナップショット機能を有効化すると、Amazon Lightsail は、デフォルトの自動スナップショット時間、または指定した時刻にリソースのスナップショットを作成します。手動スナップショットと同様に、自動スナップショットをベースラインとして、新しいリソースを作成したり、データをバックアップできます。

自動スナップショットが作成されると、Lightsail アカウントに保存されている自動スナップショットはスナップショットストレージ料金が課金されます。

目次

- 自動スナップショットの制限
- 自動スナップショット保持
- Lightsail コンソールを使用してインスタンスの自動スナップショットを有効または無効にする
- AWS CLIを使用したインスタンスまたはブロックストレージディスクの自動スナップショットを有効または無効にする

自動スナップショットの制限

自動スナップショットには、以下の制限が適用されます。

- Lightsail コンソールを使用して、ブロックストレージディスクの自動スナップショットを有効または無効にすることはできません。ブロックストレージディスクの自動スナップショットを有効または無効にするには、 Lightsail API、 AWS Command Line Interface (AWS CLI)、または SDKsを使用する必要があります。詳細については、「AWS CLIを使用した自動スナップショットを有効または無効にする」を参照してください。
- 自動スナップショットは現在、Windows インスタンスまたはマネージドデータベースではサポートされていません。代わりに、Windows インスタンスまたはマネージドデータベースの手動スナップショットを作成して、それらをバックアップする必要があります。詳細については、「Windows Server インスタンスのスナップショットを作成する」および「データベースのスナップショットを作成する」を参照してください。マネージドデータベースには、デフォルトで有効になっているポイントインタイムバックアップ機能もあり、これを使用してデータを新しいデータベースに復元できます。詳細については、「ポイントインタイムバックアップからデータベースを作成する」を参照してください。

- 自動スナップショットでは、ソースリソースのタグは保持されません。自動スナップショットから作成される新しいリソースでソースリソースのタグを保持するには、自動スナップショットから新しいリソースを作成するときにタグを手動で追加する必要があります。詳細については、「リソースにタグを追加する」を参照してください。

自動スナップショット保持

毎日 7 つの最新の自動スナップショットが保存されたあと、最も古いものから最新のものに置き換えられます。さらに、ソースリソースを削除した場合、リソースに関連付けられたすべての自動スナップショットは削除されます。この動作は、ソースリソースを削除した後でも Lightsail アカウントに保存される手動スナップショットとは異なります。自動のスナップショットを置き換えられないようにしたり、ソースリソースを削除した際に削除されないようにしたい場合、自動スナップショットを手動スナップショットとしてコピーすることができます。

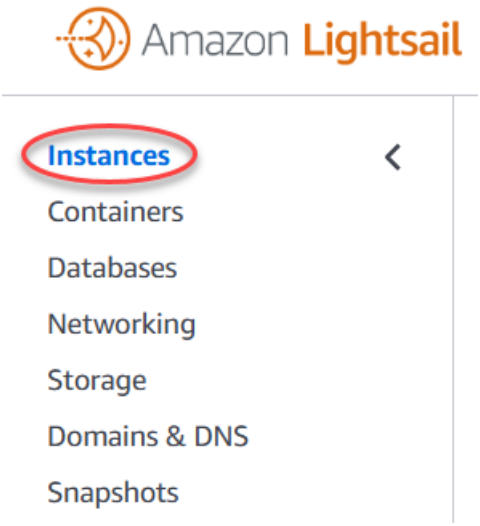
リソースの自動スナップショット機能を無効化した時、リソースの既存の自動スナップショットは、以下のいずれかの操作を行うまで、ソースリソースとともに保持されます。

- 自動スナップショットを再度有効化して、既存の自動スナップショットが新しいスナップショットに置き換える。
- 既存の自動スナップショットを手動で削除する。
- ソースリソースを削除して関連した自動スナップショットを削除する。

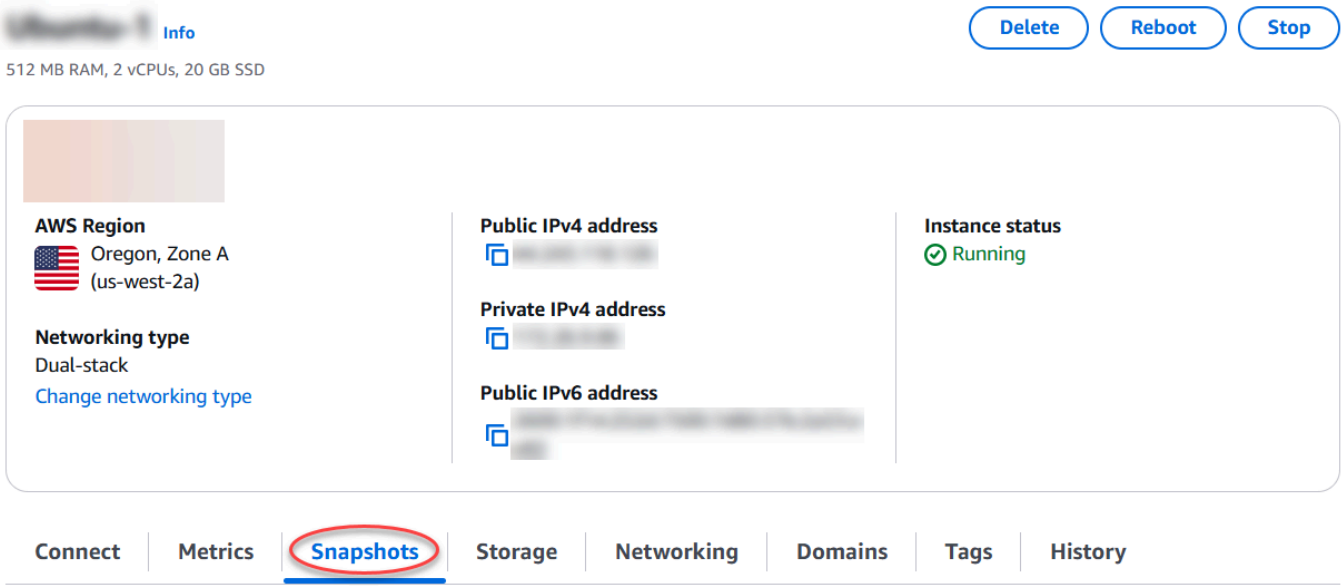
Lightsail コンソールを使用してインスタンスの自動スナップショットを有効または無効にする

Lightsail コンソールを使用してインスタンスの自動スナップショットを有効または無効にするには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [インスタンス] を選択してください。



3. 自動スナップショットを有効または無効にするインスタンスの名前を選択します。
4. インスタンス管理ページで、[Snapshots (スナップショット)] タブを選択します。



5. [自動スナップショット] セクションで、トグルを選択して有効にします。同様に、有効になっている場合は、トグルを選択して無効にします。
6. プロンプトで、[Yes, enable (はい、有効にする)] を選択して自動スナップショットを有効にするか、[Yes, disable (はい、無効にする)] を選択してこの機能を無効にします。

   しばらくすると、自動スナップショットが有効または無効になります。

   - 自動スナップショット機能を有効にした場合は、自動スナップショット時間の変更も必要になることがあります。詳細については、「インスタンスまたはブロックストレージディスクの自動スナップショット時間を変更する」を参照してください。

- 自動スナップショット機能を無効にする場合、リソースの既存の自動スナップショットは、この機能を再度有効にして新しいスナップショットに置き換えられるか、お客様が削除するまで、保持されます。Lightsail アカウントに保存されている自動スナップショットは、スナップショットストレージ料金が課金されます。自動スナップショットの削除の詳細については、「インスタンスの自動スナップショットを削除する」を参照してください。

を使用してインスタンスまたはブロックストレージディスクの自動スナップショットを有効または無効にする AWS CLI

AWS CLIを使用してインスタンスまたはブロックストレージディスクの自動スナップショットを有効または無効にするには、以下の手順を実行します。

1. ターミナルまたはコマンドプロントウィンドウを開きます。

   まだインストールしていない場合は、 をインストール AWS CLI し、 で動作するように設定しますLightsail。
2. 自動スナップショットを有効にするか無効にするかに応じて、この手順で説明するコマンドのいずれかを入力します。

   > **Note**
   >
   > これらのコマンドで
   > は、autoSnapshotAddOnRequest={snapshotTimeOfDay=HH:00} パラメータはオプションです。自動スナップショットを有効にするときに毎日の自動スナップショット時間を指定しない場合、Lightsail によってリソースにデフォルトのスナップショット時間が割り当てられます。詳細については、「インスタンスまたはブロックストレージディスクの自動スナップショット時間を変更する」を参照してください。

   - 以下のコマンドを入力して、既存のリソースの自動スナップショットを有効にします。

     ```
     aws lightsail enable-add-on --region Region --resource-name ResourceName --add-on-request
      addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=HH:00}
     ```

     コマンドを、以下のように置き換えます。

     - リソースが配置されている AWS リージョン がある#####。

- *ResourceName* は、リソースの名前に置き換えます。
- *HH:00* は、協定世界時 (UTC) での毎日の自動スナップショット時間 (1 時間単位) に置き換えます。

例:

```
aws lightsail enable-add-on --region us-west-2 --resource-name WordPress-1 --add-
on-request
 addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=18:00}
```

- 以下のコマンドを入力して、新しいインスタンスを作成するときに自動スナップショットを有効にします。

```
aws lightsail create-instances --region Region --availability-
zone AvailabilityZone --blueprint-id BlueprintID --
bundle-id BundleID --instance-name InstanceName --add-ons
 addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=HH:00}
```

コマンドを、以下のように置き換えます。

- インスタンスを作成する AWS リージョン がある*#####*。
- *AvailabilityZone* は、インスタンスが作成されるアベイラビリティーゾーンに置き換えます。
- *BlueprintID* は、インスタンスに使用する設計図 ID に置き換えます。
- *BundleID* は、インスタンスに使用するバンドル ID に置き換えます。
- *InstanceName* は、インスタンスに使用する名前に置き換えます。
- *HH:00* は、協定世界時 (UTC) での毎日の自動スナップショット時間 (1 時間単位) に置き換えます。

例:

```
aws lightsail create-instances --region us-west-2 --availability-
zone us-west-2a --blueprint-id wordpress_5_1_1_2 --bundle-
id medium_2_0 --instance-name WordPressInstance --add-ons
 addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=20:00}
```

- 以下のコマンドを入力して、新しいディスクを作成するときに自動スナップショットを有効にします。

```
aws lightsail create-disk --region Region --availability-
zone AvailabilityZone --size-in-gb Size --disk-name DiskName --add-ons
 addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=HH:00}
```

コマンドを、以下のように置き換えます。

- ディスクを作成する AWS リージョン がある*#####*。
- *AvailabilityZone* は、ディスクが作成されるアベイラビリティーゾーンに置き換えます。
- *Size* は、ディスクの希望サイズ (GB 単位) に置き換えます。
- *DiskName* は、ディスクに使用する名前に置き換えます。
- *HH:00* は、協定世界時 (UTC) での毎日の自動スナップショット時間 (1 時間単位) に置き換えます。

例:

```
aws lightsail create-disk --region us-west-2 --availability-
zone us-west-2a --size-in-gb 32 --disk-name Disk01 --add-ons
 addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=18:59}
```

- 以下のコマンドを入力して、リソースの自動スナップショットを無効にします。

```
aws lightsail disable-add-on --region Region --resource-name ResourceName --add-
on-type AutoSnapshot
```

コマンドを、以下のように置き換えます。

- リソース AWS リージョン が配置されている がある*#####*。
- *ResourceName* は、リソースの名前に置き換えます。

例:

```
aws lightsail disable-add-on --region us-west-1 --resource-
name MyFirstWordPressWebsite01 --add-on-type AutoSnapshot
```

以下の例のような結果が表示されるはずです。

しばらくすると、自動スナップショットが有効または無効になります。

- 自動スナップショット機能を有効にした場合は、自動スナップショット時間の変更も必要になることがあります。詳細については、「インスタンスまたはブロックストレージディスクの自動スナップショット時間を変更する」を参照してください。
- 自動スナップショット機能を無効にする場合、既存の自動スナップショットは、この機能を再度有効にして新しいスナップショットに置き換えられるか、お客様が削除するまで、保持されます。Lightsail アカウントに保存されている自動スナップショットは、スナップショットストレージ料金が課金されます。自動スナップショットの削除の詳細については、「インスタンスの自動スナップショットを削除する」を参照してください。

> **Note**
>
> EnableAddOn と DisableAddOn API オペレーションの詳細については、「Lightsail API ドキュメントの EnableAddOn および DisableAddOn」 を参照してください。

Lightsail インスタンスおよびディスクの自動スナップショットスケジュールを調整する

インスタンスまたはブロックストレージディスクの自動スナップショット特徴を有効化する際、Lightsail は、リソースのスナップショットを 1 日 1 回デフォルトの自動スナップショット時間、または指定した時刻に作成します。このガイドの手順に従って、リソースの自動スナップショット時間を変更します。

目次

- 自動スナップショット時間の制限
- のデフォルトの自動スナップショット時間 AWS リージョン
- Lightsail コンソールを使用して自動スナップショット時間を変更する
- を使用して自動スナップショット時間とブロックストレージディスクを変更する AWS CLI

自動スナップショット時間の制限

自動スナップショット時間には、以下の制限が適用されます。

- Lightsail コンソールを使用して、ブロックストレージディスクの自動スナップショット時間を変更することはできません。ブロックストレージディスクの自動スナップショット時間を変更するには、 Lightsail API、 AWS Command Line Interface (AWS CLI)、または SDKsを使用する必要があります。詳細については、「AWS CLIを使用して自動スナップショット時間を変更する」を参照してください。
- 自動スナップショット時間は 1 時間単位でのみ指定できます。また、現在の時間から 30 分を超える時間であることが必要です。Lightsail によって、指定した時間から 45 分以内に自動スナップショットが作成されます。

> **Important**
>
> 自動スナップショットの作成中は、手動スナップショットを作成できません。

- リソースの自動スナップショット時間を変更すると、以下の条件下でなければ、その時間は通常すぐに有効になります。
  - 現在の日に自動スナップショットがすでに作成されていて、スナップショット時間を後の時刻に変更した場合、新しいスナップショット時間は翌日に有効になります。その結果、現在の日に 2 つのスナップショットが作成されることはありません。

- 現在の日の自動スナップショットがまだ作成されておらず、スナップショット時間をその日の過去の時刻に変更した場合、新しいスナップショット時間は翌日に有効になります。また、スナップショットは、現在の日の以前に設定した時刻に自動的に作成されます。その結果、現在の日のスナップショットが作成されます。
- 現在の日の自動スナップショットがまだ作成されておらず、スナップショット時間を現在の時刻から 30 分以内の時刻に変更した場合、新しいスナップショット時間は翌日に有効になります。また、スナップショットは、現在の日の以前に設定した時刻に自動的に作成されます。現在の時間と指定した新しいスナップショット時間の間に 30 分が必要であるため、現在の日にスナップショットが作成されます。
- 現在の時間から 30 分以内に自動スナップショットが作成されるようにスケジュールされている場合、スナップショット時間を変更すると、新しいスナップショット時間は翌日に有効になります。また、スナップショットは、現在の日の以前に設定した時刻に自動的に作成されます。現在の時間と指定した新しいスナップショット時間の間に 30 分が必要であるため、現在の日にスナップショットが作成されます。

これらの条件のいずれかに当てはまると、Lightsail コンソールにメッセージが表示されて、新しいスナップショットが有効になるまでに最大 24 時間かかる場合があることが通知されます。

のデフォルトの自動スナップショット時間 AWS リージョン

自動スナップショットを有効にするときに自動スナップショット時間を指定しない場合は、Lightsail によって以下のデフォルトの自動スナップショット時間のいずれかが割り当てられます。時間は、インスタンスまたはブロックストレージディスク AWS リージョン が配置されている によって異なります。

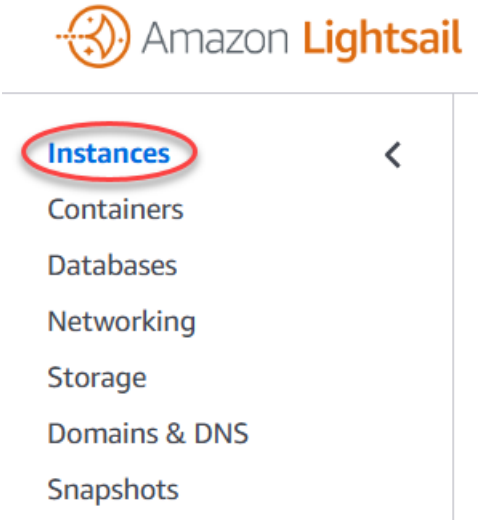
- 米国東部 (バージニア北部) (us-east-1): 06:00 UTC
- 米国東部 (オハイオ) (us-east-2): 03:00 UTC
- 米国西部 (オレゴン) (us-west-2): 06:00 UTC
- アジアパシフィック (ムンバイ) (ap-south-1): 17:00 UTC
- アジアパシフィック (ソウル) (ap-northeast-2): 13:00 UTC
- アジアパシフィック (シンガポール) (ap-southeast-1): 14:00 UTC
- アジアパシフィック (シドニー) (ap-southeast-2)): 12:00 UTC
- アジアパシフィック (ジャカルタ) (ap-southeast-3): 08:00 UTC
- アジアパシフィック (東京) (ap-northeast-1): 13:00 UTC
- カナダ (中部) (ca-central-1): 06:00 UTC

- 欧州 (フランクフルト) (eu-central-1): 20:00 UTC
- 欧州 (アイルランド) (eu-west-1): 22:00 UTC
- 欧州 (ロンドン) (eu-west-2): 06:00 UTC
- 欧州 (パリ) (eu-west-3): 07:00 UTC
- 欧州 (ストックホルム) (eu-north-1): 08:00 UTC

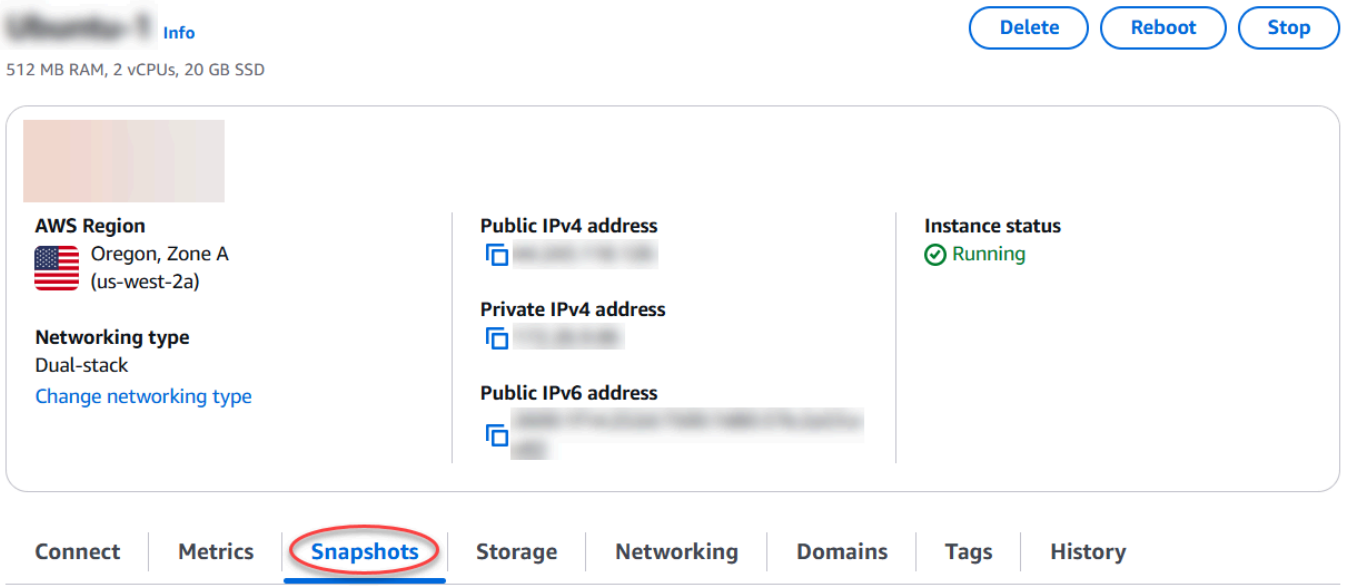
Lightsail コンソールを使用して自動スナップショット時間を変更する

Lightsail コンソールを使用してインスタンスの自動スナップショット時間を変更するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [インスタンス] を選択してください。



3. 自動スナップショット時間を変更するインスタンスの名前を選択します。
4. インスタンス管理ページで、[Snapshots (スナップショット)] タブを選択します。



5. [Automatic snapshots (自動スナップショット)] セクションで、[Change snapshot time (スナップショット時間の変更)] を選択します。
6. Lightsail によって自動スナップショットが作成される時刻を選択します。選択する時間は、協定世界時 (UTC) であることが必要です。
7. [変更] を選択して、新しいスナップショット時間を保存します。

   しばらくすると、自動スナップショット時間が更新されます。制限は新しい自動スナップショット時間の発効日に適用されるものとします。詳細については、「自動スナップショット時間の制限」を参照してください。

を使用してインスタンスとブロックストレージディスクの自動スナップショット時間を変更する AWS CLI

AWS CLIを使用してインスタンスまたはブロックストレージディスクの自動スナップショット時間を変更するには、以下の手順を実行します。

1. ターミナルまたはコマンドプロントウィンドウを開きます。

   まだインストールしていない場合は、 をインストール AWS CLI し、 と連携するように設定してくださいLightsail。
2. 以下のコマンドを入力して、リソースの自動スナップショット時間を変更します。

```
aws lightsail enable-add-on --region Region --resource-name ResourceName --add-on-
request addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=HH:00}
```

コマンドを、以下のように置き換えます。

- リソース AWS リージョン が配置されている がある*#####*。
- *ResourceName* は、リソースの名前に置き換えます。
- *HH:00* は、協定世界時 (UTC) での毎日の自動スナップショット時間 (1 時間単位) に置き換えます。

例:

```
aws lightsail enable-add-on --region us-west-1 --resource-
name MyFirstWordPressWebsite01 --add-on-request
 addOnType=AutoSnapshot,autoSnapshotAddOnRequest={snapshotTimeOfDay=12:00}
```

以下の例のような結果が表示されるはずです。

しばらくすると、自動スナップショット時間が更新されます。制限は新しい自動スナップショット時間の発効日に適用されるものとします。詳細については、「自動スナップショット時間の制限」を参照してください。

> **Note**
>
> EnableAddOn API オペレーションの詳細については、Lightsail API ドキュメントの EnableAddOn を参照してください。

未使用の Lightsail インスタンスとディスクスナップショットを削除する

Amazon Lightsail でインスタンスまたはブロックストレージディスクの自動スナップショットはいつでも削除できます。これは、自動スナップショットを有効にしていても、有効にしていた後に無効にしていても同じです。Lightsail アカウントに自動的に保存されているスナップショットに対してスナップショットストレージ料金が請求されます。自動スナップショットが不要になった場合は、このガイドの手順に従って削除します。たとえば、自動スナップショットを手動スナップショットにコピーして元のスナップショットが不要になった場合や、リソースの自動スナップショット機能を無効にしたため、保持している既存の自動スナップショットが不要になった場合です。

目次

- 自動スナップショットの削除に関する制限
- Lightsail コンソールを使用してインスタンスの自動スナップショットを削除する
- を使用してインスタンスまたはブロックストレージディスクの自動スナップショットを削除する AWS CLI

自動スナップショットの削除に関する制限

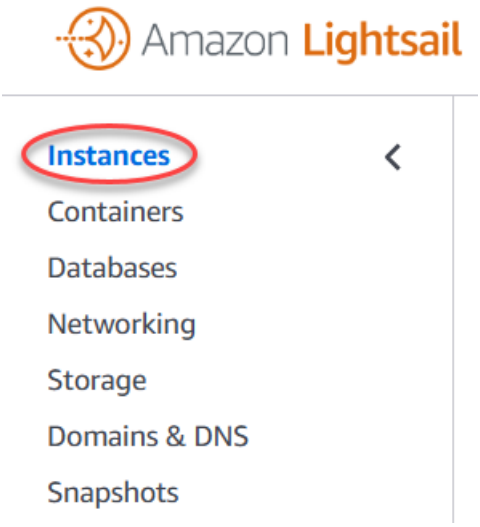
ブロックストレージディスクの自動スナップショットは、Lightsail コンソールを使用して削除できません。ブロックストレージディスクの自動スナップショットを削除するには、 Lightsail API、 AWS Command Line Interface (AWS CLI)、または SDKsを使用する必要があります。詳細については、「AWS CLIを使用してインスタンスまたはブロックストレージディスクの自動スナップショットを削除する」を参照してください。

Lightsail コンソールを使用してインスタンスの自動スナップショットを削除する

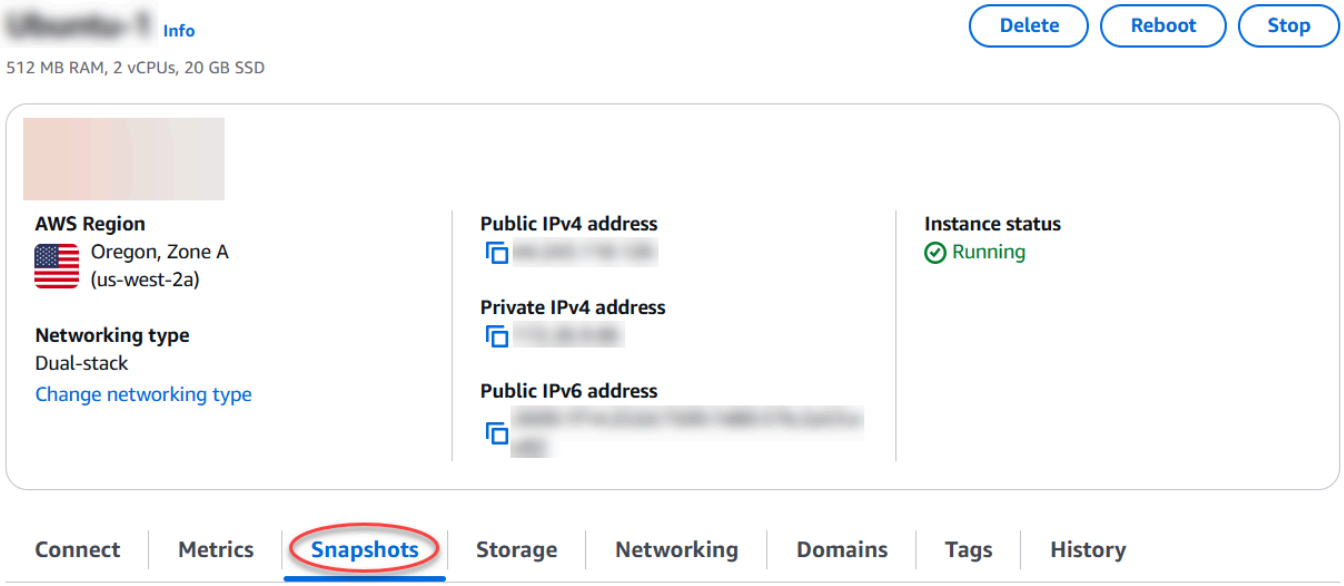
Lightsail コンソールを使用してインスタンスの自動スナップショットを削除するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。

2. 左のナビゲーションペインの [インスタンス] を選択してください。



3. 自動スナップショットを削除するインスタンスの名前を選択します。
4. インスタンス管理ページで、[Snapshots (スナップショット)] タブを選択します。



5. [Automatic snapshots (自動スナップショット)] セクションで、削除する自動スナップショットの横にある省略記号アイコンを選択し、[スナップショットの削除] を選択します。
6. プロンプトで、[はい] を選択して、スナップショットを削除することを確定します。

   しばらくすると、自動スナップショットが削除されます。

を使用してインスタンスまたはブロックストレージディスクの自動スナップショットを削除する AWS CLI

AWS CLIを使用してインスタンスまたはブロックストレージディスクの自動スナップショットを削除するには、以下の手順を実行します。

1. ターミナルまたはコマンドプロントウィンドウを開きます。

   まだインストールしていない場合は、 をインストール AWS CLI し、 と連携するように設定してくださいLightsail。

2. 以下のコマンドを入力して、特定のリソースの使用可能な自動スナップショットの日付を取得します。自動スナップショットの日付は、後続のコマンドで date パラメータとして指定するために必要です。

   ```
   aws lightsail --region Region get-auto-snapshots --resource-name ResourceName
   ```

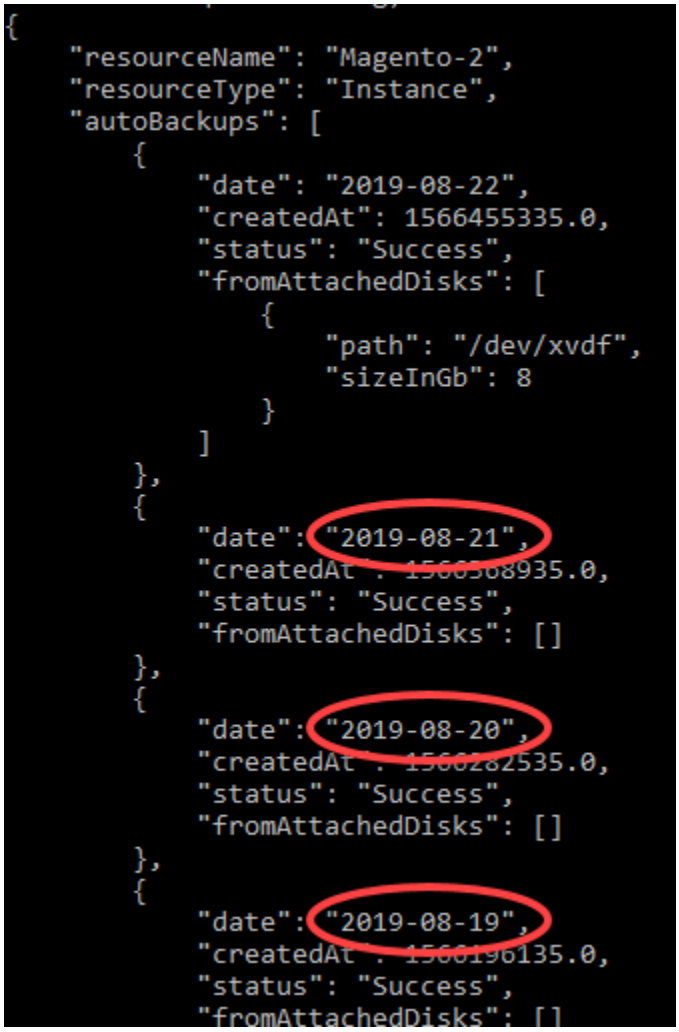
   コマンドを、以下のように置き換えます。

   - リソース AWS リージョン が配置されている がある*#####*。
   - *ResourceName* は、リソースの名前に置き換えます。

   例:

   ```
   aws lightsail --region us-west-2 get-auto-snapshots --resource-
   name MyFirstWordPressWebsite01
   ```

   以下のような結果が表示され、使用可能な自動スナップショットが一覧表示されます。



3. 以下のコマンドを入力して、自動スナップショットを削除します。

```
aws lightsail --region Region delete-auto-snapshot --resource-name ResourceName --
date YYYY-MM-DD
```

コマンドを、以下のように置き換えます。

- リソース AWS リージョン が配置されている がある*#####*。
- *ResourceName* は、リソースの名前に置き換えます。
- *YYYY-MM-DD* は、前のコマンドで取得した使用可能な自動スナップショットの日付に置き換えます。

例:

```
aws lightsail --region us-west-2 delete-auto-snapshot --resource-
name MyFirstWordPressWebsite01 --date 2019-09-16
```

以下の例のような結果が表示されるはずです。

しばらくすると、自動スナップショットが削除されます。

> **Note**
>
> これらのコマンドのスナップショットの GetAutoSnapshot および DeleteAutoSnapshot API オペレーションの詳細については、Lightsail API ドキュメント内の「スナップショットの取得」および「自動スナップショットの削除」を参照してください。

Lightsail で自動スナップショットが置き換えられないようにする

Amazon Lightsail でインスタンスまたはブロックストレージディスクに対して自動スナップショット機能を有効にする場合、リソースの自動スナップショットは最新の 7 日分のみが保存されます。その後、最も古いものが最新のものに置き換えられます。さらに、出典リソースを削除した場合、リソースに関連付けられたすべての自動スナップショットは削除されます。

特定の自動スナップショットを置き換えられないようにしたり、ソースリソースを削除した際にリソースも削除されないようにしたい場合は、手動スナップショットとしてコピーすることができます。手動スナップショットは、ユーザーがマニュアルで削除しない限り保持されます。

このガイドの手順に従って、自動スナップショットを手動スナップショットとしてコピーして保存します。Lightsailアカウントに保存されている自動スナップショットは、スナップショットストレージ料金が課金されます。

> **Note**
>
> リソースの自動スナップショット機能を無効にする場合、リソースの既存の自動スナップショットは、この機能を再度有効にして新しいスナップショットに置き換えられるか、お客様が自動スナップショットを削除するまで、保持されます。

目次

- 自動スナップショットの制限を保持する
- Lightsail コンソールを使用したインスタンスの自動スナップショットの保持
- を使用してインスタンスとブロックストレージディスクの自動スナップショットを保持する AWS CLI

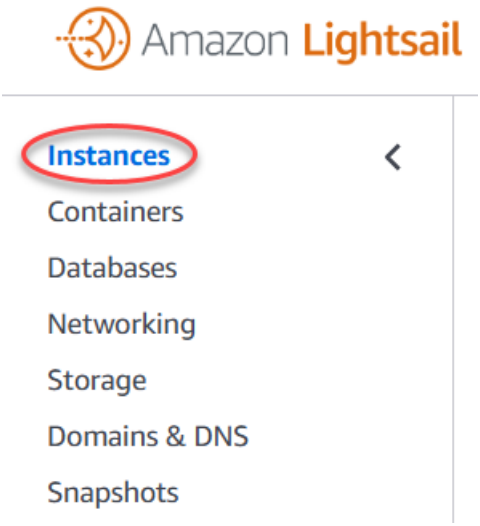
自動スナップショットの制限を保持する

Lightsail コンソールを使用して、ブロックストレージディスクの自動スナップショットを手動スナップショットにコピーすることはできません。ブロックストレージディスクの自動スナップショットをコピーするには、 Lightsail API、 AWS Command Line Interface (AWS CLI)、または SDKsを使用する必要があります。詳細については、「AWS CLIを使用したインスタンスおよびブロックストレージディスクの自動スナップショットの保持」を参照してください。

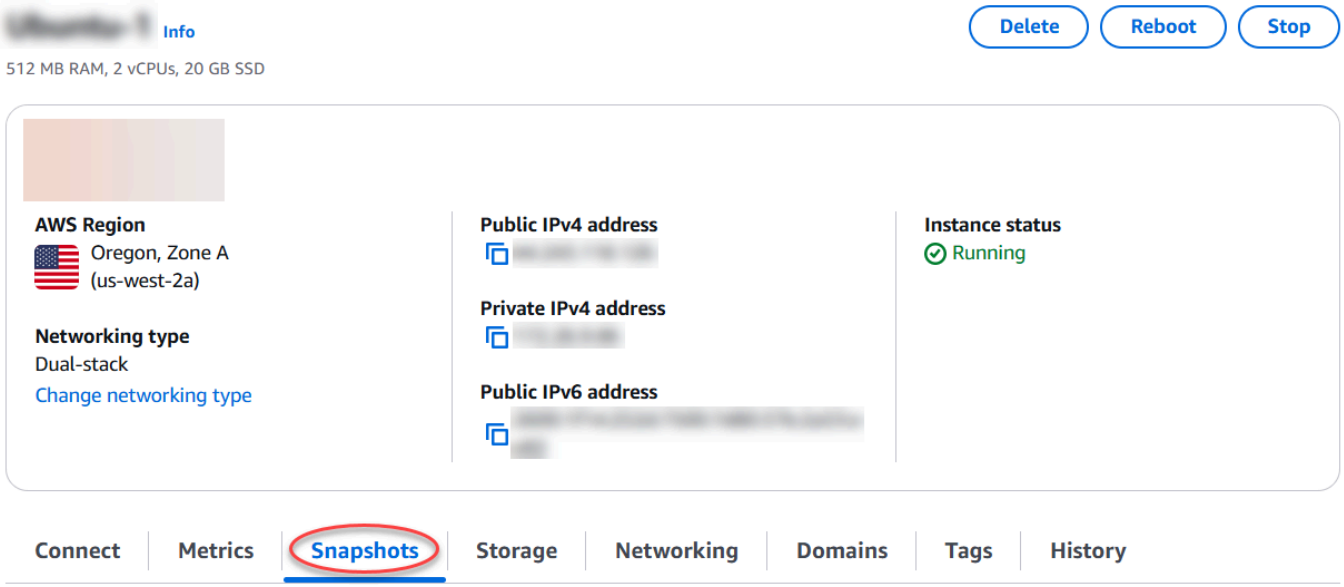
Lightsail コンソールを使用したインスタンスの自動スナップショットの保持

Lightsail コンソールを使用してインスタンスの自動スナップショットを保持するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [インスタンス] を選択してください。



3. 自動スナップショットを保持するインスタンスの名前を選択します。
4. インスタンス管理ページで、[Snapshots (スナップショット)] タブを選択します。



5. [Automatic snapshots (自動スナップショット)] セクションで、保持する自動スナップショットの横にある省略記号アイコンを選択し、[Keep snapshot (スナップショットの保持)] を選択します。
6. プロンプトで「Yes, save (はい、保存する)」を選択して、自動スナップショットを保持することを確定します。

   しばらくすると、自動スナップショットが手動スナップショットとしてコピーされます。手動スナップショットは、お客様が削除するまで保持されます。

> **Important**
>
> 自動スナップショットが不要になった場合は、削除することをお勧めします。そうでないと、Lightsailアカウントに保存される自動スナップショットと重複する手動スナップショットに対してスナップショットストレージ料金が課金されます。詳細については、「自動インスタンスのスナップショットを削除する」を参照してください。

を使用してインスタンスとブロックストレージディスクの自動スナップショットを保持する AWS CLI

AWS CLIを使用してインスタンスまたはブロックストレージディスクの自動スナップショットを保持するには、以下の手順を実行します。

1. ターミナルまたはコマンドプロントウィンドウを開きます。

   まだインストールしていない場合は、 をインストール AWS CLI し、 と連携するように設定してくださいLightsail。

2. 以下のコマンドを入力して、特定のリソースの使用可能な自動スナップショットの日付を取得します。自動スナップショットの日付は、後続のコマンドで `restore date` パラメータとして指定するために必要です。

   ```
   aws lightsail get-auto-snapshots --region Region --resource-name ResourceName
   ```

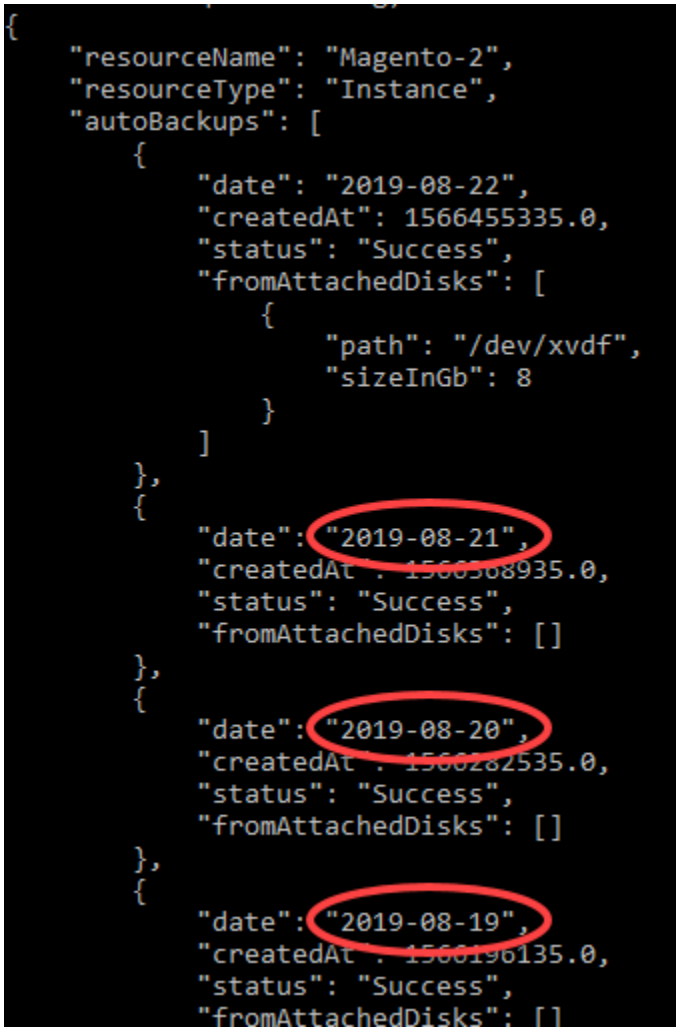
   コマンドを、以下のように置き換えます。

   - リソース AWS リージョン が配置されている がある*#####*。
   - *ResourceName* は、リソースの名前に置き換えます。

   例:

   ```
   aws lightsail get-auto-snapshots --region us-west-2 --resource-
   name MyFirstWordPressWebsite01
   ```

   以下のような結果が表示され、使用可能な自動スナップショットが一覧表示されます。



3. 以下のコマンドを入力して、特定のリソースの自動スナップショットを保持します。

```
aws lightsail copy-snapshot --region TargetRegion --source-resource-
name ResourceName --restore-date YYYY-MM-DD --source-region SourceRegion --target-
snapshot-name SnapshotName
```

コマンドを、以下のように置き換えます。

- *TargetRegion* AWS リージョン とスナップショットをコピーする 。
- *ResourceName* は、リソースの名前に置き換えます。
- *YYYY-MM-DD* は、前のコマンドで取得した使用可能な自動スナップショットの日付に置き換えます。
- 自動スナップショットが現在存在する AWS リージョン を持つ *SourceRegion*。
- *SnapshotName* は、作成される新しいスナップショットの名前に置き換えます。

例:

```
aws lightsail copy-snapshot --region us-west-2 --source-resource-
name MyFirstWordPressWebsite01 --restore-date 2019-09-16 --source-region us-west-2
 --target-snapshot-name Snapshot-Copied-From-Auto-Snapshot
```

以下の例のような結果が表示されるはずです。

しばらくすると、自動スナップショットが手動スナップショットとしてコピーされます。手動スナップショットは、お客様が削除するまで保持されます。

> **Important**
>
> 自動スナップショットが不要になった場合は、削除することをお勧めします。そうしないと、Lightsailアカウントに保存された自動スナップショットと複製の手動スナップショットに対してスナップショットストレージの料金が課金されます。詳細については、「自動インスタンスのスナップショットを削除する」を参照してください。

> **Note**
>
> これらのコマンドでの GetAutoSnapshots と CopySnapshot API オペレーションに関する詳細については、LightsailAPI ドキュメントのGetAutoSnapshots および CopySnapshot を参照してください。

スナップショットを使用して Linux/Unix Lightsail インスタンスをバックアップする

Linux/Unix ベースの Amazon Lightsail インスタンスのスナップショットを作成できます。インスタンススナップショットはシステムディスクのコピーであり、元のマシン構成 (メモリ、CPU、ディスクサイズ、およびデータ転送レート) を複製したものです。ブロックストレージディスクをインスタンスにアタッチした場合、Lightsail はスナップショットの一部としてそれらの追加のディスクをコピーします。詳細については、「スナップショット」を参照してください。

> **Note**
>
> Windows Server ベースの Lightsail インスタンスのスナップショットを作成する手順は異なります。詳細については、「Windows Server インスタンスのスナップショットを作成する」を参照してください。

スナップショットを作成するには、Lightsail でインスタンスが作成されている必要があります。インスタンスの準備が整ったら、次の手順に従ってスナップショットを作成します。

1. Lightsail のホームページで、スナップショットを作成するインスタンスの名前を選択します。
2. [スナップショット] タブを選択します。
3. このページの [手動スナップショット] セクションで、[スナップショットの作成] を選択し、スナップショットの名前を入力します。

   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。

- 先頭と末尾は英数字または数字を使用する必要があります。
- 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

4. [作成] を選択します。

   作成したスナップショットのステータスが [Snapshotting... (スナップショットの作成中)] と表示されることがあります。

   スナップショットが完了したら、スナップショットから別のインスタンスを作成することができます。たとえば、以前のものより大きいサイズのバンドルを選択できます。

> **Important**
>
> Lightsail では、スナップショットから新しいインスタンスを作成するときに、同じサイズまたはより大きいサイズのインスタンスバンドルを作成できます。現在は、スナップショットより小さいサイズのインスタンスの作成はサポートされていません。スナップショットから新しいインスタンスを作成する際、より小さいオプションは灰色で表示されます。

スナップショットを使用して、より大きいインスタンスサイズを作成するには、Lightsail コンソールで、CLI の [create-instances-from-snapshot] CLI コマンドを使用するか、[CreateInstancesFromSnapshot] API オペレーションを使用します。詳細については、「スナップショットからインスタンスを作成する」を参照してください。Lightsail バンドルについて詳しくは、「Lightsail 料金表」を参照してください。

Lightsail Windows Server インスタンスのスナップショットを作成する

スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、メモリ、CPU、ディスクサイズ、データ転送レートなどの情報が含まれています。詳細については、「スナップショット」を参照してください。

Lightsail で Windows Server インスタンスのスナップショットを作成するには、最初にバックアップスナップショットを作成します。次に、システム準備 (Sysprep) という特別なユーティリティを使用して、2 つ目のスナップショットを作成します。Sysprep は Windows Server のインストールを一般化するため、インスタンスをスナップショットとしてバックアップできます。次に、そのスナップ

ショットからインスタンスを作成すると、Windows インスタンスの最初の実行エクスペリエンスであるように感じます。

Linux または UNIX インスタンスのスナップショットを作成するには、「Linux または Unix インスタンスのスナップショットを作成する」を参照してください。

目次

- ステップ 1: Sysprep を実行する前にバックアップスナップショットを作成する
- ステップ 2: Sysprep を使用してインスタンスに接続し、シャットダウンする
- ステップ 3: Sysprep の実行後にスナップショットを作成する

ステップ 1: Sysprep を実行する前にバックアップスナップショットを作成する

Sysprep を実行してスナップショットを作成すると、システム固有の情報はインスタンスから削除されます。これは、インスタンスで実行されているアプリケーションに意図しない結果をもたらす場合があります。したがって、Sysprep を実行する前にバックアップスナップショットを必ず作成し、何か異常が発生した場合に備えて別のスナップショットを用意します。

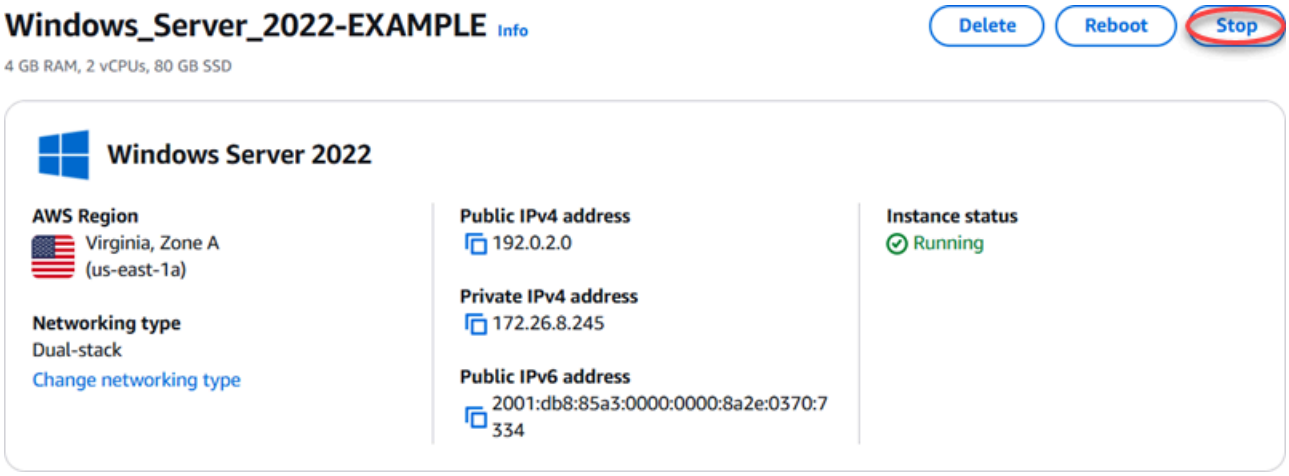
Sysprep を実行する前にスナップショットを作成すると、このバックアップスナップショットを使用して作成するインスタンスでは、元のインスタンスと同じ管理者パスワードが使用されます。これらのインスタンスには、Lightsail コンソールのブラウザベースの RDP クライアントを使用して接続できません。ただし、独自の RDP クライアントおよび元のインスタンスと同じ管理者パスワードを使用して接続できます。詳細については、「Windows コンピュータでリモートデスクトップ接続クライアントを使用して Amazon Lightsail の Windows インスタンスに接続する」を参照してください。

> **Important**
>
> 元の Windows インスタンスの管理者パスワードを保存して、安全な場所に保管します。後に問題が発生した場合に管理者パスワードが必要になります。その場合は、Sysprep を実行する前に作成したスナップショットからインスタンスを作成します。

Sysprep を実行する前にバックアップスナップショットを作成するには

1. Lightsail コンソール にサインインします。

2. Lightsail のホームページで、スナップショットを作成する Windows Server インスタンスの名前を選択します。
3. インスタンス管理ページの上部にある [停止] を選択してインスタンスを停止します。



> **Note**
>
> インスタンスを停止すると、再開するまでインスタンスのウェブサイトやサービスは使用できなくなります。

4. [スナップショット] タブを選択します。
5. このページの [手動スナップショット] セクションで、[スナップショットの作成] を選択し、スナップショットの名前を入力します。

   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。
6. [作成] を選択します。
7. 確認のために、プロンプトで [スナップショットの作成] をもう一度選択します。

   スナップショットプロセスの完了までには数分かかります。
8. スナップショットの作成後、インスタンス管理ページの上部にある [開始] を選択し、インスタンスを再開します。

ステップ 2: Sysprep を使用してインスタンスに接続し、シャットダウンする

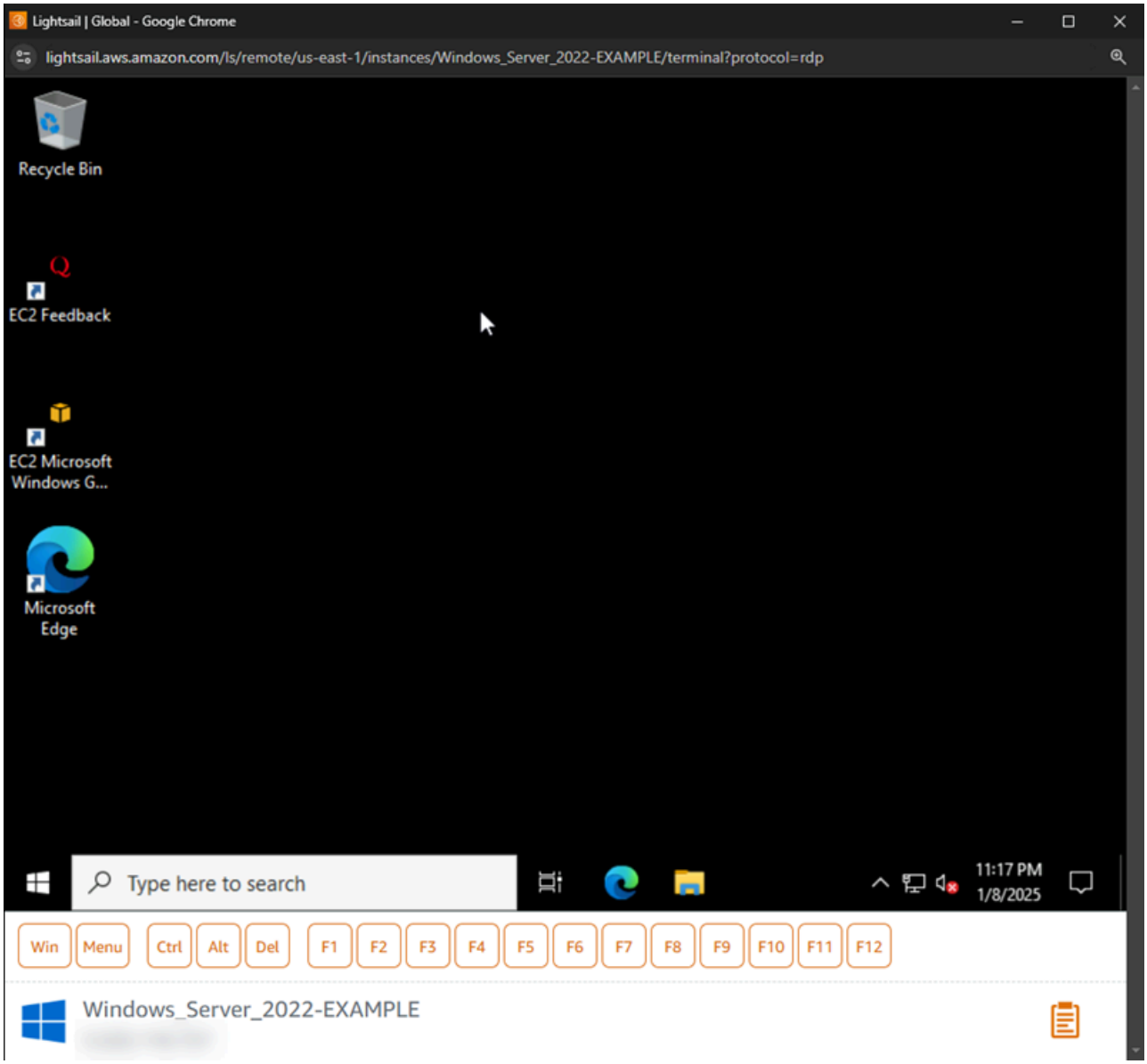
バックアップスナップショットを作成したので、次は Windows Server インスタンスで Sysprep を実行します。これに伴ってインスタンスがシャットダウンされ、スナップショットを作成できるようになります。Sysprep の詳細については、Microsoft のドキュメントで「Sysprep Overview」を参照してください。

このステップでは、プリインストール済みのアプリケーションを通じてインスタンスに接続し、Sysprep を実行します。アプリケーションは Windows Server 2019 および Windows Server 2016 のインスタンスでは「EC2LaunchSettings」で、Windows Server 2012 インスタンスでは「Ec2ConfigService の設定」です。

インスタンスに接続して Sysprep を実行するには

1. インスタンス管理ページで、[接続] タブ、[RDP を使用して接続] の順に選択します。

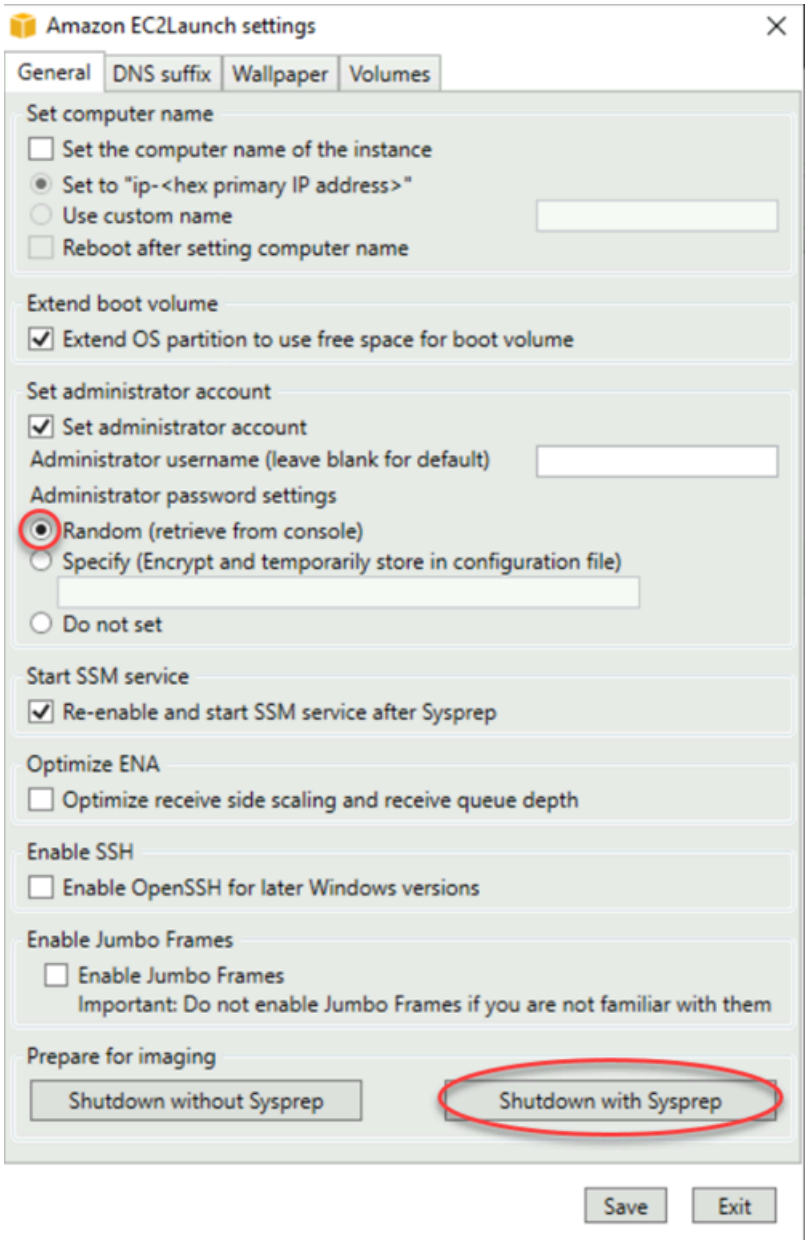
   以下の例に示すように、ブラウザベースの RDP ウィンドウが開きます。



2. タスクバーで、Windows アイコンを選択するか、[Win] を選択してスタートメニューを開きます。
3. 以下のいずれかのオプションを選択します：

   - Windows Server 2022、Windows Server 2019 および Windows Server 2016 インスタンスの場合、[スタート]、[Ec2LaunchSettings] の順に選択します。

4. [Administrator Password (管理者パスワード)] セクションで、[Random (Retrieve from console) (ランダム (コンソールから取得))]、[Shutdown with Sysprep (Sysprep でシャットダウン)] の順に選択します。



5. [Yes (はい)] をクリックし、Sysprep を実行してインスタンスをシャットダウンすることを確認します。

   インスタンスで Sysprep の実行が始まり、RDP 接続がシャットダウンし、Lightsail インスタンスが数分後に実行を停止します。

ステップ 3: Sysprep の実行後にスナップショットを作成する

インスタンスが停止状態になったら、Lightsail コンソールでスナップショットを作成します。Sysprep の実行後に Windows Server インスタンスのスナップショットを作成すると、このス

ナップショットから作成するすべてのインスタンスで一意の管理者パスワードが使用されます。これらのインスタンスには、Lightsail コンソールのブラウザベースの RDP クライアントを使用して接続できます。

Lightsail コンソールでスナップショットを作成するには

1. Lightsail コンソールに戻ります。
2. Windows Server インスタンスのインスタンス管理ページで、[Snapshots (スナップショット)] タブを選択します
3. このページの [手動スナップショット] セクションで、[スナップショットの作成] を選択し、スナップショットの名前を入力します。

   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。
4. [作成] を選択します。
5. スナップショットのインスタンスを作成する準備ができたら、プロンプトで [スナップショットの作成] を選択します。

   スナップショットプロセスの完了までには数分かかります。
6. スナップショットの作成後、インスタンス管理ページの上部にある [開始] を選択し、インスタンスを再開します。

   この時点で、次の例に示すように、Windows Server インスタンスの 2 つの作成済みスナップショットが表示されます。

   Sysprep スナップショットを使用して、新しいインスタンスを作成します。バックアップスナップショットは、Sysprep の実行後に元のインスタンスが予期どおりに機能しない場合にのみ使用します。

次の手順

Sysprep およびバックアップスナップショットの作成が完了したので、次のステップを実行できます。

- 元のインスタンスに接続し、インスタンスのアプリケーションが Sysprep の実行後に予期どおりに機能することを確認します。詳細については、「Amazon Lightsail を使用して Windows Server インスタンスに接続する」を参照してください。
- Sysprep を使用して新しいインスタンスを作成し、これに接続して、新しいインスタンスのアプリケーションが予期どおりに機能することを確認します。詳細については、「スナップショットからインスタンスを作成する」を参照してください。
- Sysprep の実行後に元のインスタンスが想定どおりに機能することを確認した後、バックアップスナップショットを削除します。詳細については、「スナップショットを削除する」を参照してください。
- Sysprep の実行後にインスタンスが予期したとおりに機能しない場合は、「スナップショットからインスタンスを作成する」のステップに従って、バックアップスナップショットから新しいインスタンスを作成します。

バックアップまたはベースライン用の Lightsail ブロックストレージディスクスナップショットを作成します

Amazon Lightsail では、ディスクスナップショットを追加のブロックストレージディスクのバックアップとして作成することができます。

ディスクのスナップショットを、新しいディスクまたはデータバックアップのためのベースラインとして使用できます。ディスクのスナップショットを定期的に作成する場合、スナップショットは差分になります。最後にスナップショットを作成した時点から、デバイス上で変更があったブロックだけが、新しいスナップショットに保存されます。スナップショットの保存は増分ベースで行われるものの、最新のスナップショットさえあればディスク全体を復元できるようにスナップショット削除プロセスは設計されています。

詳細については、「スナップショット」を参照してください。

1. 左側のナビゲーションペインで [ストレージ] を選択します。
2. スナップショットを作成するブロックストレージディスクの名前を選択します。
3. [スナップショット] タブを選択します。

4. このページの [手動スナップショット] セクションで、[スナップショットの作成] を選択し、スナップショットの名前を入力します。

   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

5. [作成] を選択します。

   作成したスナップショットのステータスが [Snapshotting... (スナップショットの作成中)] と表示されることがあります。

   スナップショットが完了したら、スナップショットから別のディスクを作成することができます。

Lightsail のブロックストレージディスクのスナップショットを作成する

ディスクスナップショットから新しいブロックストレージディスクを作成できます。完全に新規のディスクを作成する場合は、代わりに「追加のブロックストレージディスクを作成する (Linux/Unix)」または「ブロックストレージディスクを作成して Windows Server インスタンスにアタッチする」を参照してください。

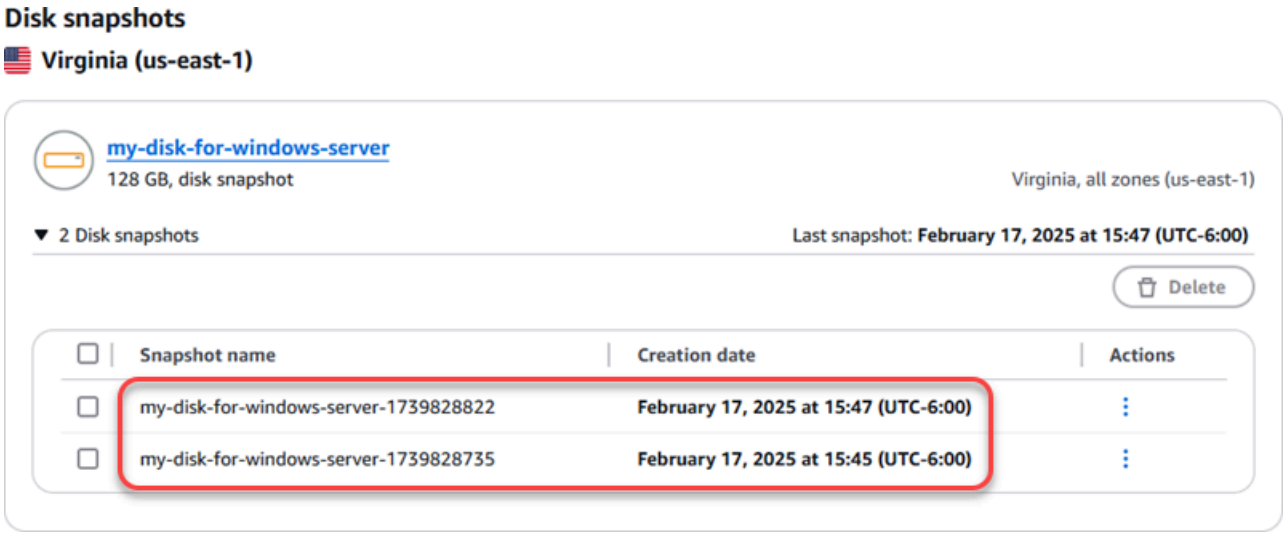
ブロックストレージディスクのスナップショットを、新しいディスクまたはデータバックアップのためのベースラインとして使用できます。ディスクのスナップショットを定期的に作成する場合、スナップショットは差分になります。最後にスナップショットを作成した時点から、ディスク上で変更があったブロックだけが、新しいスナップショットに保存されます。スナップショットの保存は増分ベースで行われるものの、最新のスナップショットさえあればディスク全体を復元できるようにスナップショット削除プロセスは設計されています。ブロックストレージディスクのスナップショットを作成するには、「ブロックストレージディスクのスナップショットを作成する」を参照してください。

ステップ 1: ディスクスナップショットを検索して新しいディスクの作成を選択する

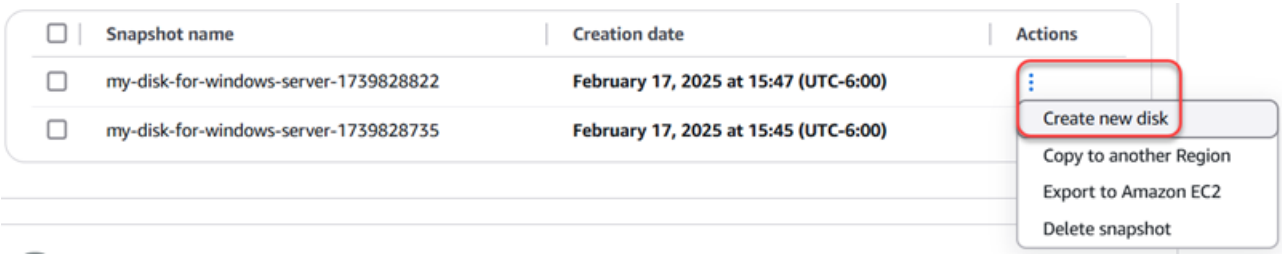
Lightsail の 2 つの場所 (Lightsail ホームページの [スナップショット] タブ、またはディスク管理ページの [スナップショット] タブ) のいずれかで、ディスクスナップショットから新しいインスタンスを作成できます。

Lightsail ホームページから

1. 左側のナビゲーションペインの左側のナビゲーションバーで、[スナップショット] を選択します。
2. ディスクの名前を見つけ、その下のノードを展開して、そのディスクの利用可能なすべてのスナップショットを表示します。



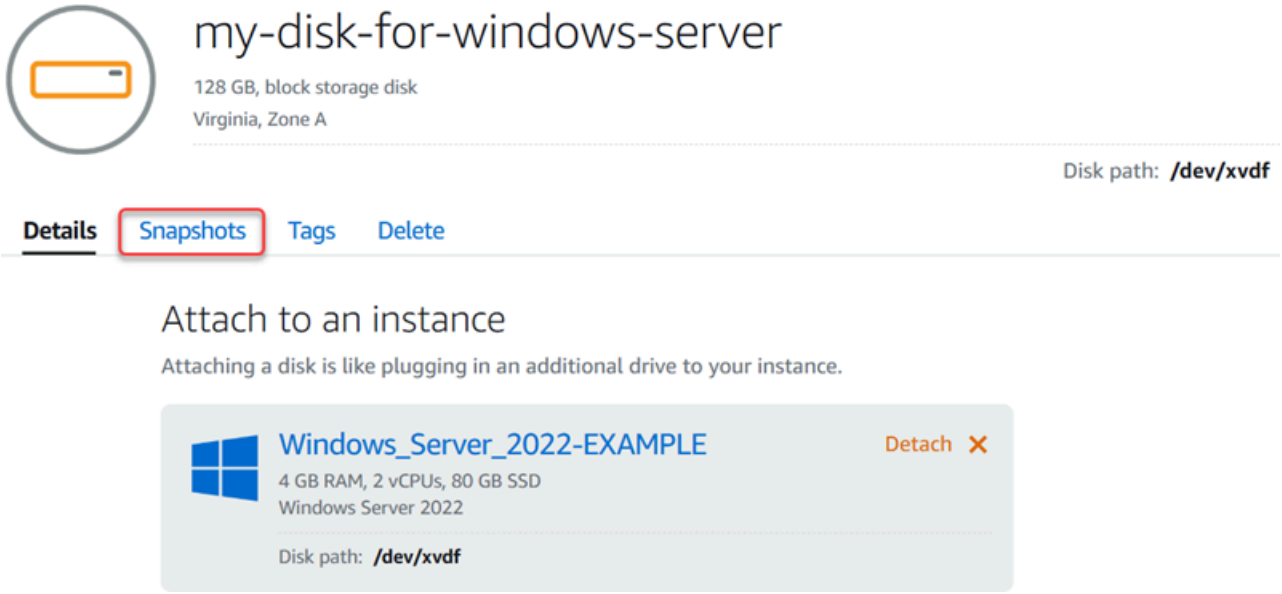
3. 新しいディスクを作成するスナップショットの横にあるアクションメニューアイコン (⋮) を選択し、[新しいディスクの作成] を選択します。



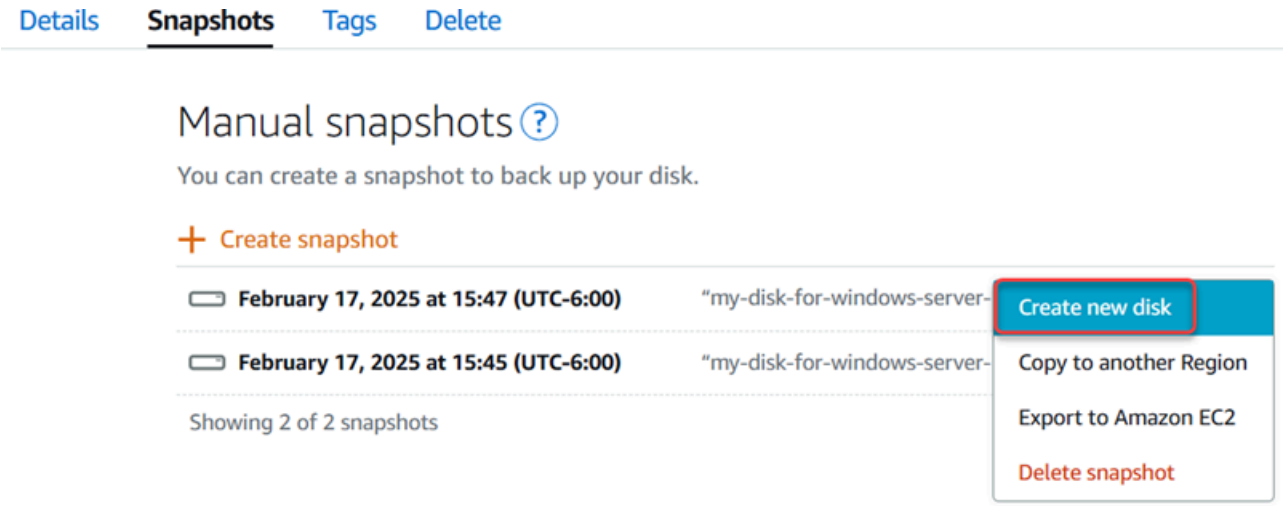
Lightsail のディスク管理ページから

1. 左側のナビゲーションペインの左側のナビゲーションバーで、[ストレージ] タブを選択します。

2. スナップショットを表示するディスクの名前を選択します。
3. [スナップショット] タブを選択します。



4. このページの [手動スナップショット] セクションで、新しいディスクを作成するスナップショットの横にあるアクションメニューアイコン (:) を選択してから、[新しいディスクの作成] を選択します。



ステップ 2: ディスクスナップショットから新しいディスクを作成する

1. 新しいディスクのアベイラビリティーゾーンを選択するか、デフォルトのままにします (us-east-2a)。

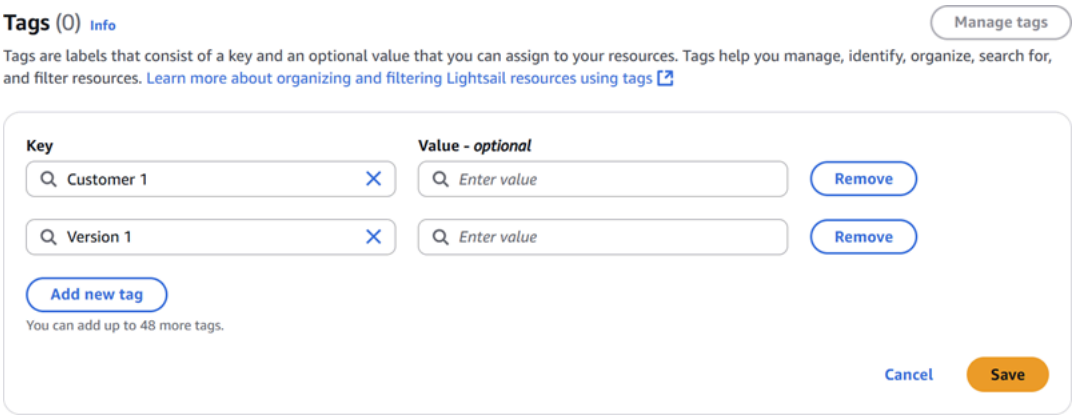
   ソースディスク AWS リージョン と同じ に新しいディスクを作成する必要があります。

2. 新しいディスクには、ソーススナップショット以上のサイズを選択してください。
3. ディスクの名前を入力します。

   リソース名:

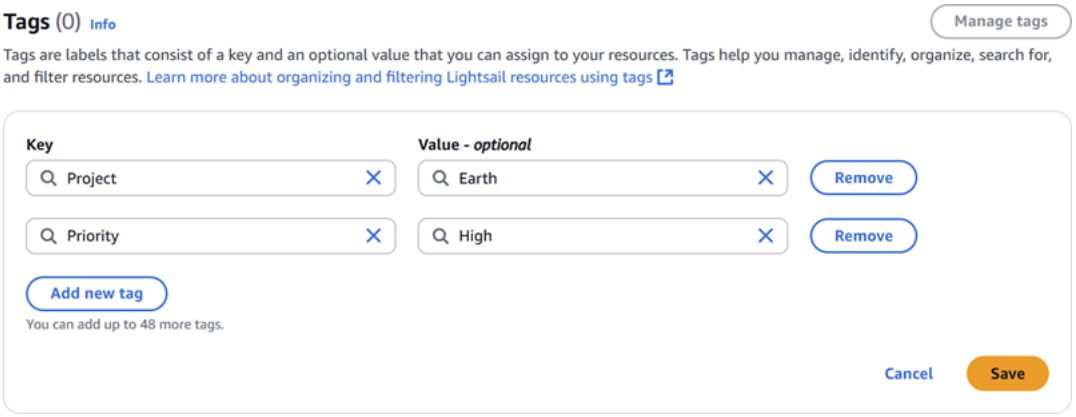
   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。
4. 以下のいずれかのオプションを選択して、ディスクにタグを追加します。

   - [キーオンリータグの追加] または [タグを編集] (タグが追加済みの場合)。タグキーのテキストボックスに新しいタグを入力し、Enter キーを押します。タグの入力を完了したら、[保存] を選択してタグを追加し、追加しない場合は、[キャンセル] を選択します。



   - [Create a key-value tag] (キーバリュータグを作成) から [Key] (キー) テキストボックスにキーを入力し、[Value] (値) テキストボックスに値を入力します。タグの入力を完了したら、[保存] を選択し、追加しない場合は、[キャンセル] を選択します。

     キーバリューのタグは、保存する際に一つずつ追加することができます。さらにキーバリュータグを追加するには、以上のステップを繰り返します。



> **Note**
>
> 「キーのみ」のタグと「キーバリュー」のタグの詳細については、「タグ」を参照してください。

5. [ディスクの作成] を選択します。

Lightsail インスタンスのルートボリュームのスナップショットを作成する

システムディスクのスナップショットを作成して、Amazon Lightsail インスタンスのルートボリュームをバックアップします。そして、スナップショットから新しいブロックストレージディスクを作成し、別のインスタンスにアタッチすることによって、バックアップされたファイルにアクセスします。必要な場合は、以下のステップを実行します。

- 失敗したインスタンスのルートボリュームからデータを復旧します。
- ブロックストレージディスクに対して行うように、インスタンスのルートボリュームのバックアップを作成します。

AWS Command Line Interface (AWS CLI) または を使用してインスタンスルートボリュームスナップショットを作成します AWS CloudShell。スナップショットを作成した後、Lightsail コンソールを使用してスナップショットからブロックストレージディスクを作成します。次に、それを実行中のインスタンスにアタッチし、そのインスタンスからアクセスします。

目次

- ステップ 1: 前提条件を満たす
- ステップ 2: インスタンスのルートボリュームスナップショットを作成する
- ステップ 3: スナップショットからブロックストレージディスクを作成し、インスタンスへアタッチする
- ステップ 4: インスタンスからブロックストレージディスクにアクセスする

ステップ 1: 前提条件を満たす

AWS Command Line Interface (AWS CLI)、または を使用して AWS CloudShell 、インスタンスのルートボリュームスナップショットを作成します。CloudShell はブラウザーベースの事前認証済みシェルで、Lightsail コンソールから直接起動できます。詳細については「Lightsail オペレーション AWS CLI 用の をセットアップおよび設定する」および「でLightsailリソースを管理する AWS CloudShell」を参照してください。

ステップ 2: インスタンスのルートボリュームスナップショットを作成する

ターミナル、CloudShell またはコマンドプロンプトウィンドウを開き、次のコマンドを入力して、ルートボリュームのスナップショットのインスタンスを作成します。

```
aws lightsail create-disk-snapshot --region AWSRegion --instance-name InstanceName --
disk-snapshot-name DiskSnapshotName
```

コマンドを、以下のように置き換えます。

- インスタンス AWS リージョン の を持つ *AWSRegion*。
- *InstanceName* は、ルートボリュームをバックアップするインスタンスの名前に置き換えます。
- *DiskSnapshotName* は、作成される新しいディスクスナップショットの名前に置き換えます。

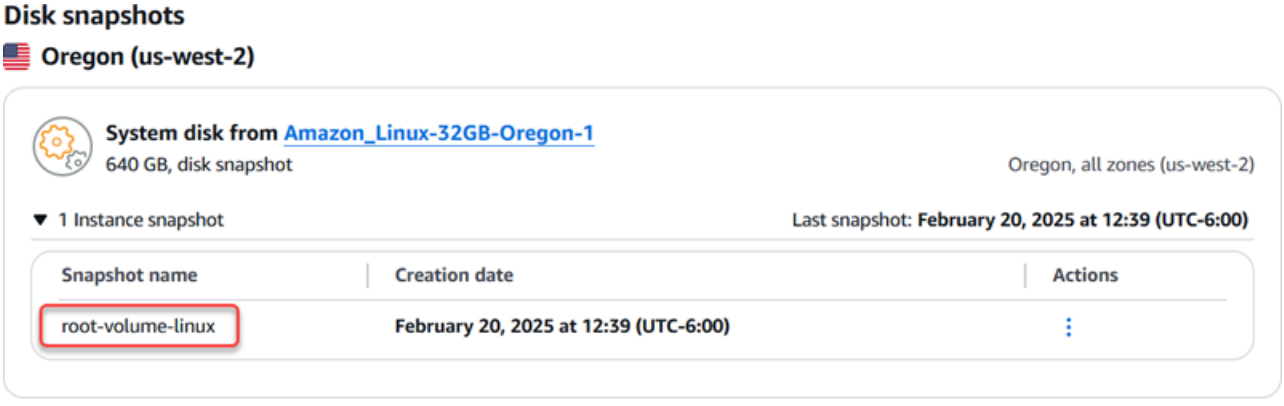
例:

```
aws lightsail create-disk-snapshot --region us-west-2 --instance-
name Amazon_Linux-32GB-Oregon-1 --disk-snapshot-name root-volume-linux
```

成功すると、以下のような結果が表示されます。

```
H:\>aws lightsail create-disk-snapshot --region us-west-2 --instance-name Amazon_Linux-32GB-Oregon-1
--disk-snapshot-name root-volume-linux

{
    "operations": [
        {
            "status": "Started",
            "resourceType": "DiskSnapshot",
            "isTerminal": false,
            "operationDetails": "Amazon_Linux-32GB-Oregon-1",
            "statusChangedAt": 1548799955.599,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "operationType": "CreateDiskSnapshot",
            "resourceName": "root-volume-linux",
            "id": "[redacted]",
            "createdAt": 1548799955.599
        },
        {
            "status": "Started",
            "resourceType": "Instance",
            "isTerminal": false,
            "operationDetails": "root-volume-linux",
            "statusChangedAt": 1548799955.599,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "operationType": "CreateDiskSnapshot",
            "resourceName": "Amazon_Linux-32GB-Oregon-1",
            "id": "[redacted]",
            "createdAt": 1548799955.599
        }
    ]
}
```

スナップショットが作成されるまで数分待ちます。作成後は、Lightsail ホームページの左側のナビゲーションペインで [スナップショット] を選択し、[ディスクスナップショット] セクションまで下にスクロールすることで確認できます (以下の例を参照)。



ステップ 3: スナップショットからブロックストレージディスクを作成し、インスタンスへアタッチする

インスタンスのルートボリュームのスナップショットから新しいブロックストレージディスクを作成し、そのコンテンツにアクセスするためには、別のインスタンスにアタッチします。失敗したインスタンスのルートボリュームからデータを復旧する必要がある場合は、以下を実行します。

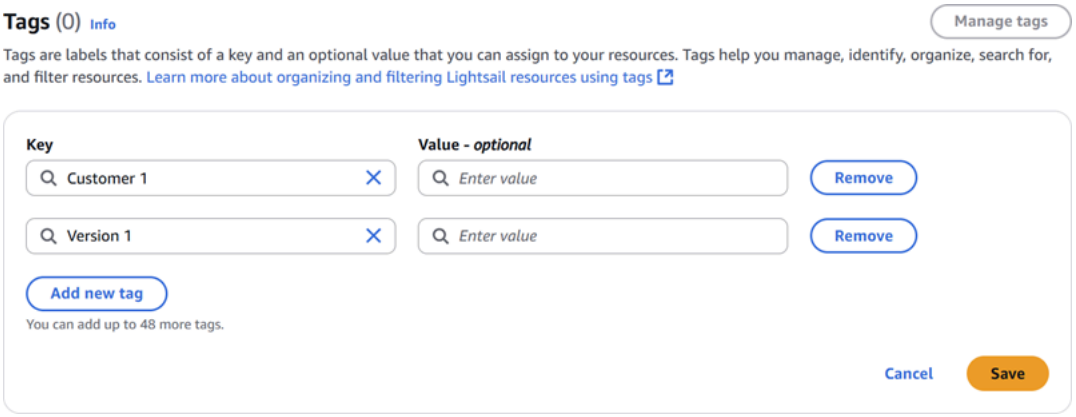
> **Note**
>
> 新しいブロックストレージディスクは、ソーススナップショット AWS リージョン と同じに作成されます。別のリージョンにブロックストレージディスクを作成するためには、目的のリージョンにスナップショットをコピーし、コピーしたスナップショットから新しいディスクを作成します。詳細については、「Copy snapshots from one AWS リージョン to another」を参照してください。

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインで [スナップショット]を選択します。
3. 使用するルートボリュームディスクスナップショットの横に表示されるアクションメニューアイコン (:) を選択し、[Create new disk (新しいディスクの作成)] を選択します。
4. ディスクのアベイラビリティーゾーンを選択するか、デフォルトのままにします。
5. ソースディスクと同等、もしくはそれ以上のサイズのディスクを選択してください。
6. ディスクの名前を入力します。

   リソース名:

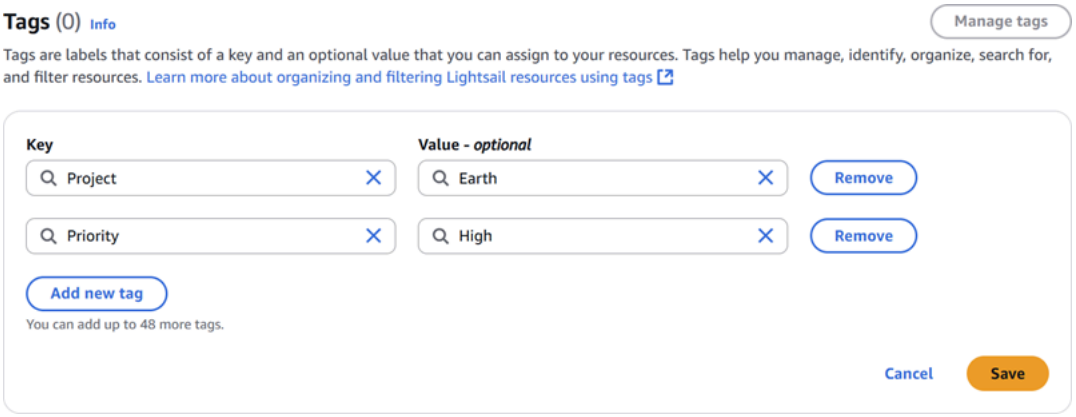
   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。
7. 以下のいずれかのオプションを選択して、ディスクにタグを追加します。

   - [キーオンリータグの追加] または [タグを編集] (タグが追加済みの場合)。タグキーのテキストボックスに新しいタグを入力し、Enter キーを押します。タグの入力を完了したら、[保存] を選択してタグを追加し、追加しない場合は、[キャンセル] を選択します。



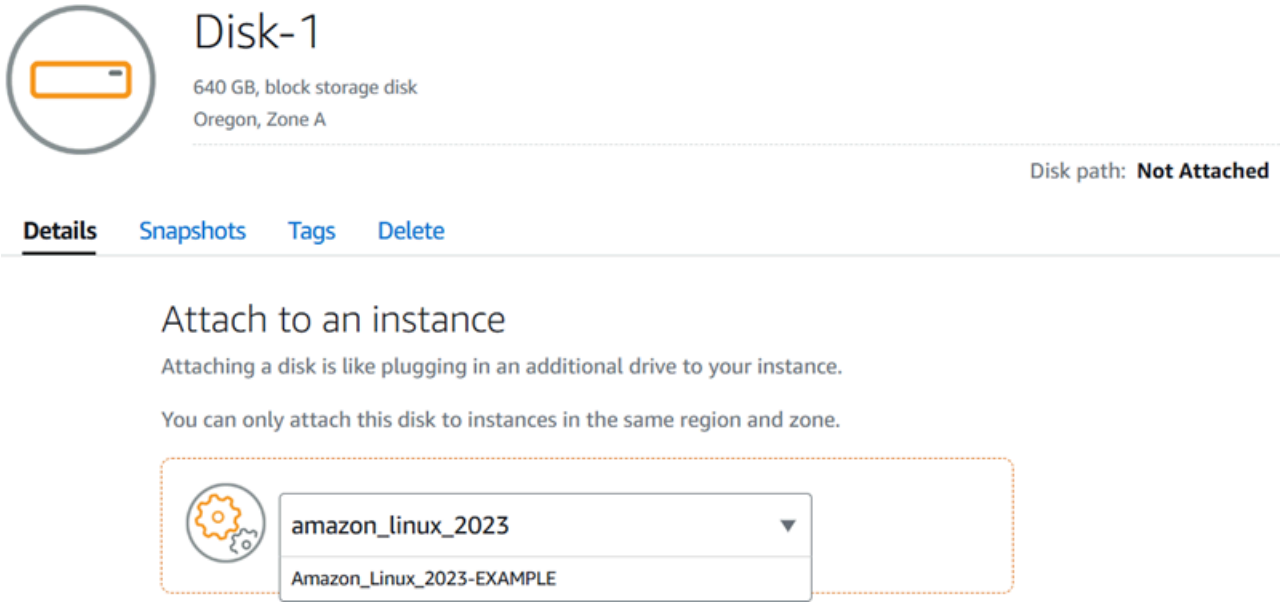
- [Create a key-value tag] (キーバリュータグを作成) から [Key] (キー) テキストボックスにキーを入力し、[Value] (値) テキストボックスに値を入力します。タグの入力を完了したら、[保存] を選択し、追加しない場合は、[キャンセル] を選択します。

  キーバリューのタグは、保存する際に一つずつ追加することができます。さらにキーバリュータグを追加するには、以上のステップを繰り返します。



> **Note**
>
> 「キーのみ」のタグと「キーバリュー」のタグの詳細については、「タグ」を参照してください。

8. [ディスクの作成] を選択します。
9. ディスクが作成されたら、ディスクをアタッチするインスタンスを [Select an instance (インスタンスの選択)] ドロップダウンメニューで選択します。これは次の例で示されます。



10. [アタッチ] を選択して、選択したインスタンスにディスクをアタッチします。

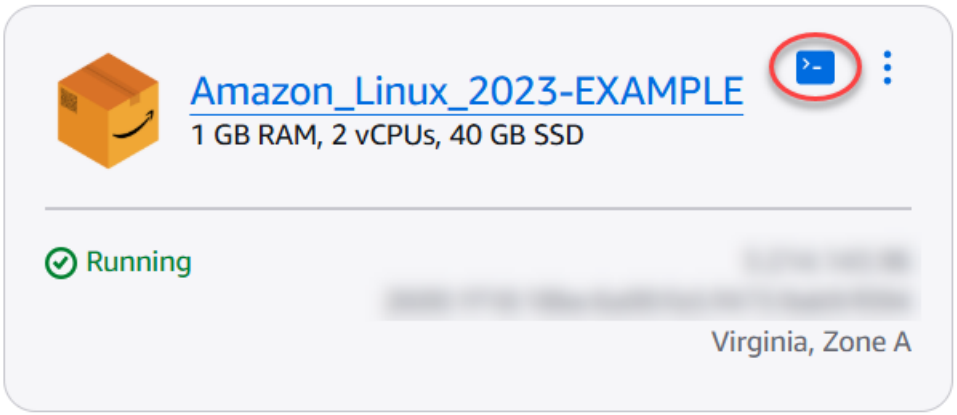
    ディスクがインスタンスにアタッチされます。次に、Linux にマウントするか、Windows でオンラインにすることによって、該当するオペレーティングシステムにアクセスできる状態にします。詳細については、このガイドの次の [インスタンスからブロックストレージにアクセスする] セクションを参照してください。

ステップ 4: インスタンスからブロックストレージディスクにアクセスする

インスタンスにアタッチした後でブロックストレージディスクにアクセスするには、Linux または Unix でマウントするか、Windows でオンラインにする必要があります。

Linux または Unix インスタンスにブロックストレージディスクをマウントしてアクセスする

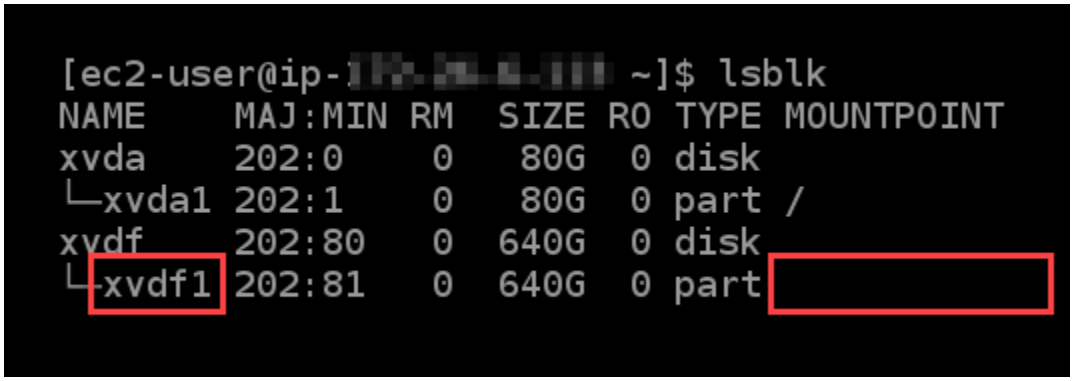
1. Lightsail ホームページで、ブロックストレージディスクをアタッチした Linux または Unix インスタンスのブラウザベースの SSH クライアントのアイコンを選択します。



2. ブラウザベースの SSH クライアントを接続したら、次のコマンドを入力して、インスタンスにアタッチしたブロックストレージディスクデバイスを表示します。

```
lsblk
```

次の例のような結果が表示されます。この例でxvdf1 は、マウントポイントがないため、マウントされていないインスタンスにアタッチされたブロックストレージディスクです。また、結果では、デバイス名から /dev/ が除外されているため、実際のデバイス名は /dev/xvdf1 となります。



3. 次のコマンドを入力して、ブロックストレージディスクのマウントポイントを作成します。

```
sudo mkdir MountPoint
```

コマンドで *MountPoint*を、ブロックストレージディスクをマウントし、アクセス可能にするディレクトリの名前に置き換えます。

例:

```
sudo mkdir xvdf
```

4. 次のコマンドを入力し、前のステップで作成したマウントポイントにブロックストレージディスクをマウントします。

```
sudo mount /dev/DeviceName MountPoint
```

コマンドは、以下のように置き換えます。

- *DeviceName* は、ブロックストレージディスクデバイスの名前に置き換えます。
- *MountPoint* は、前のステップで作成したマウントポイントディレクトリに置き換えます。

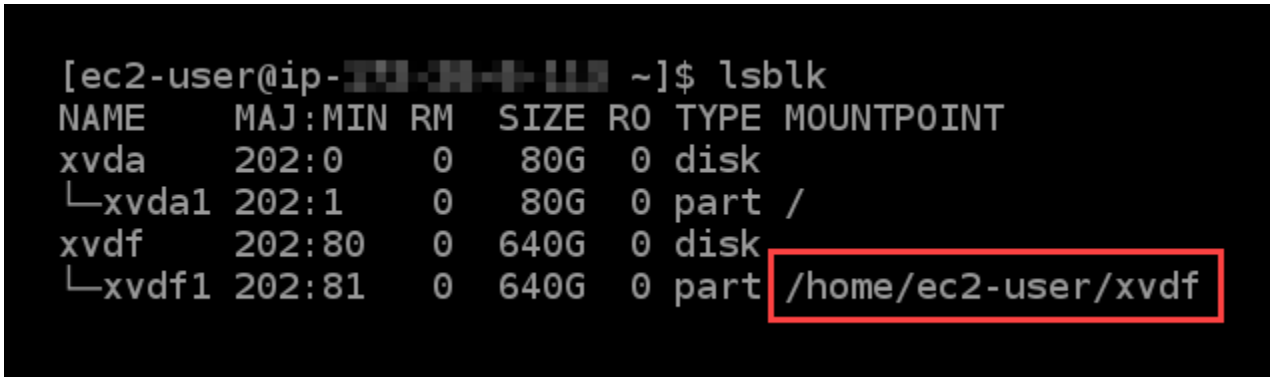
例:

```
sudo mount /dev/xvdf1 xvdf
```

5. 次のコマンドを入力して、インスタンスにアタッチしたブロックストレージディスクデバイスを表示します。

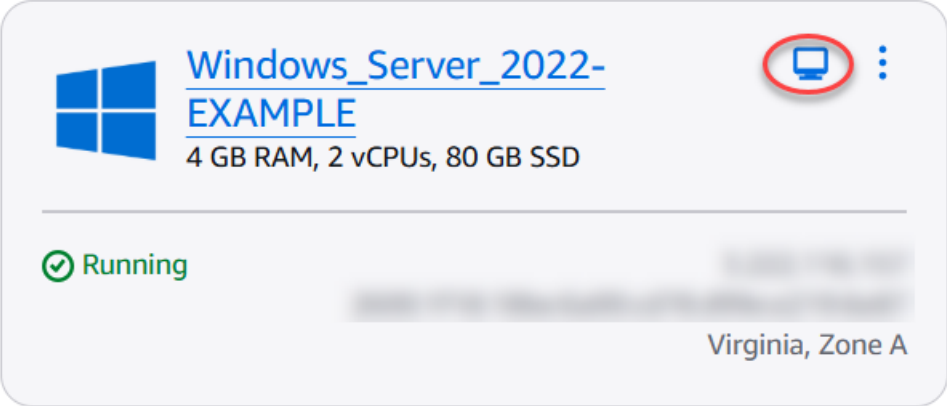
```
lsblk
```

次の例のような結果が表示されます。この例では、*xvdf1* デバイスは */home/ec2-user/xvdf* ディレクトリにマウントされ、アクセス可能になっています。マウントポイントのディレクトリで、ブロックストレージディスクとそのコンテンツにアクセスできるようになりました。

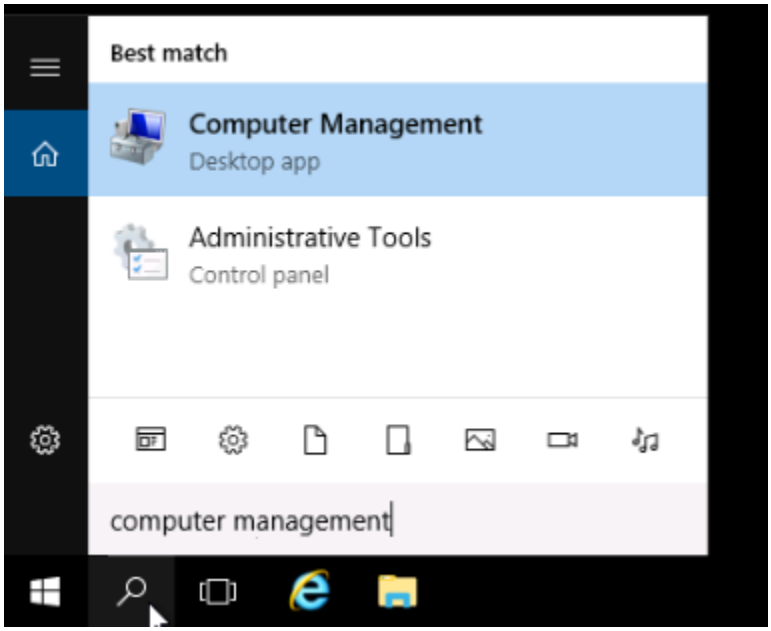


Windows インスタンスでブロックストレージディスクをオンラインにしてアクセスします。

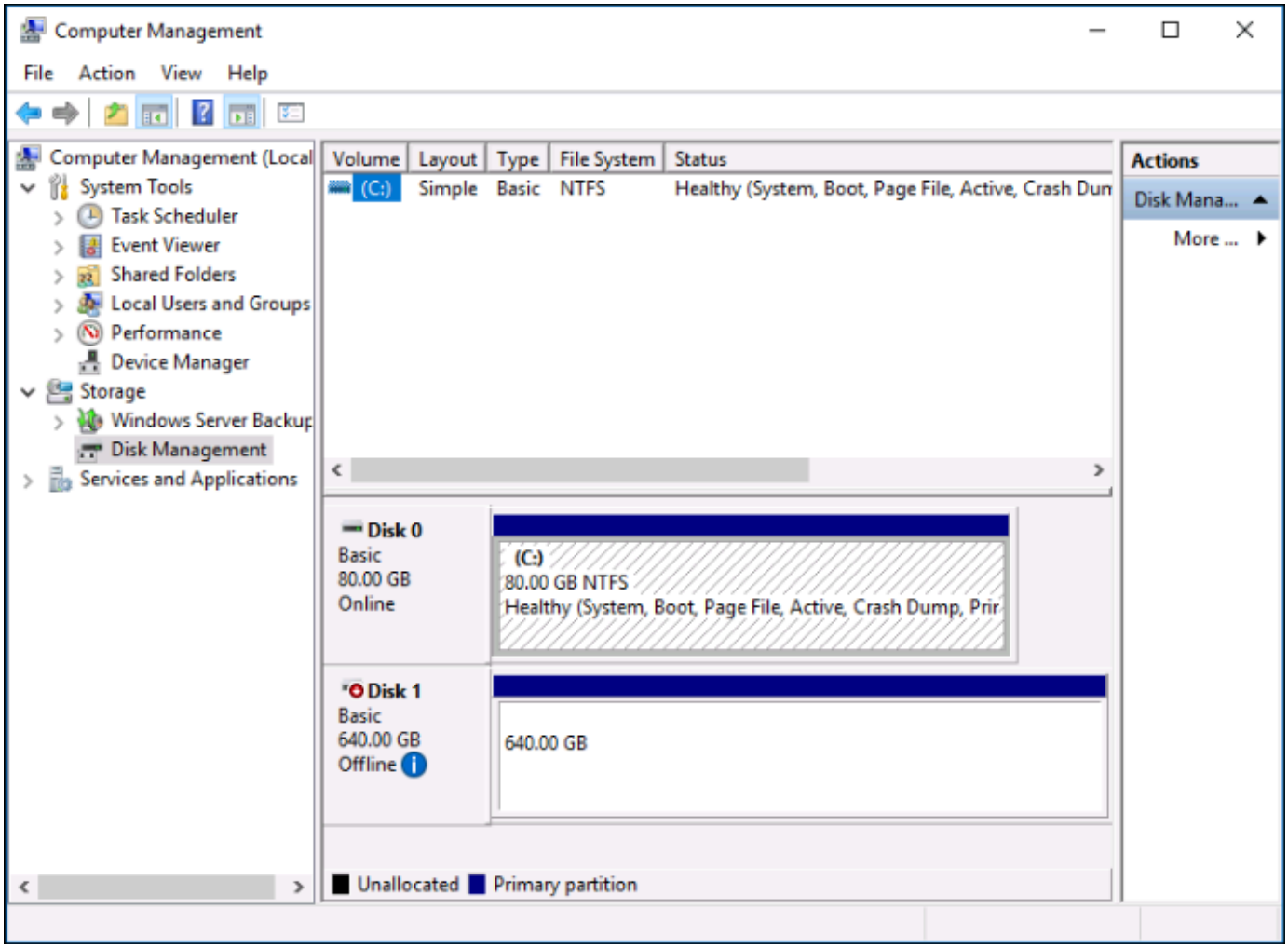
1. Lightsail ホームページで、ブロックストレージディスクをアタッチした Windows インスタンスのブラウザベースの RDP クライアントアイコンを選択します。



2. ブラウザベースの SSH クライアントが接続されたら、Windows タスクバーで [コンピュータの管理] を選択し、その結果から [コンピューターの管理] を選択します。

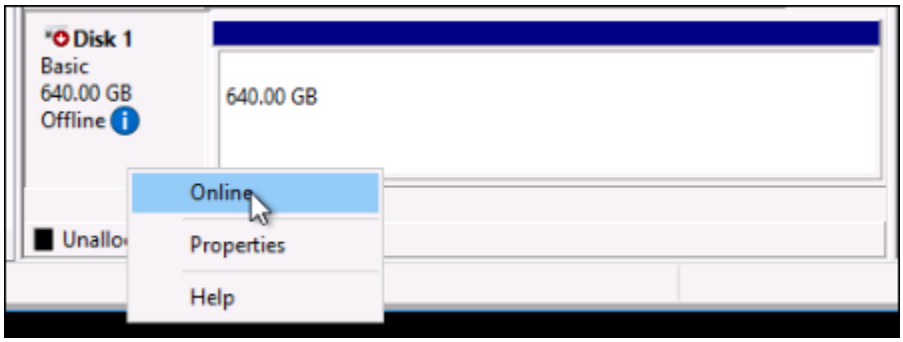


3. [コンピュータの管理] コンソールの左側のナビゲーションメニューで、以下の例のように [ディスク管理] を選択します。

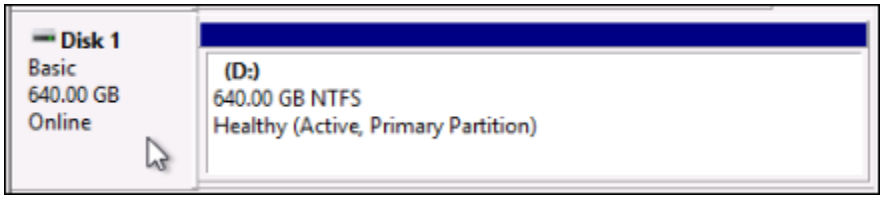


4. 最近インスタンスにアタッチしたディスクを見つけます。[オフライン] とラベル付けされているはずです。

5. [オフライン] ラベルを右クリックし、[オンライン] を選択します。



ディスクが [オンライン] として表示され、ドライブ文字が関連付けられているはずです。File Explorer を開いて指定したドライブ文字を参照することにより、ブロックストレージディスクとそのコンテンツにアクセスできるようになりました。

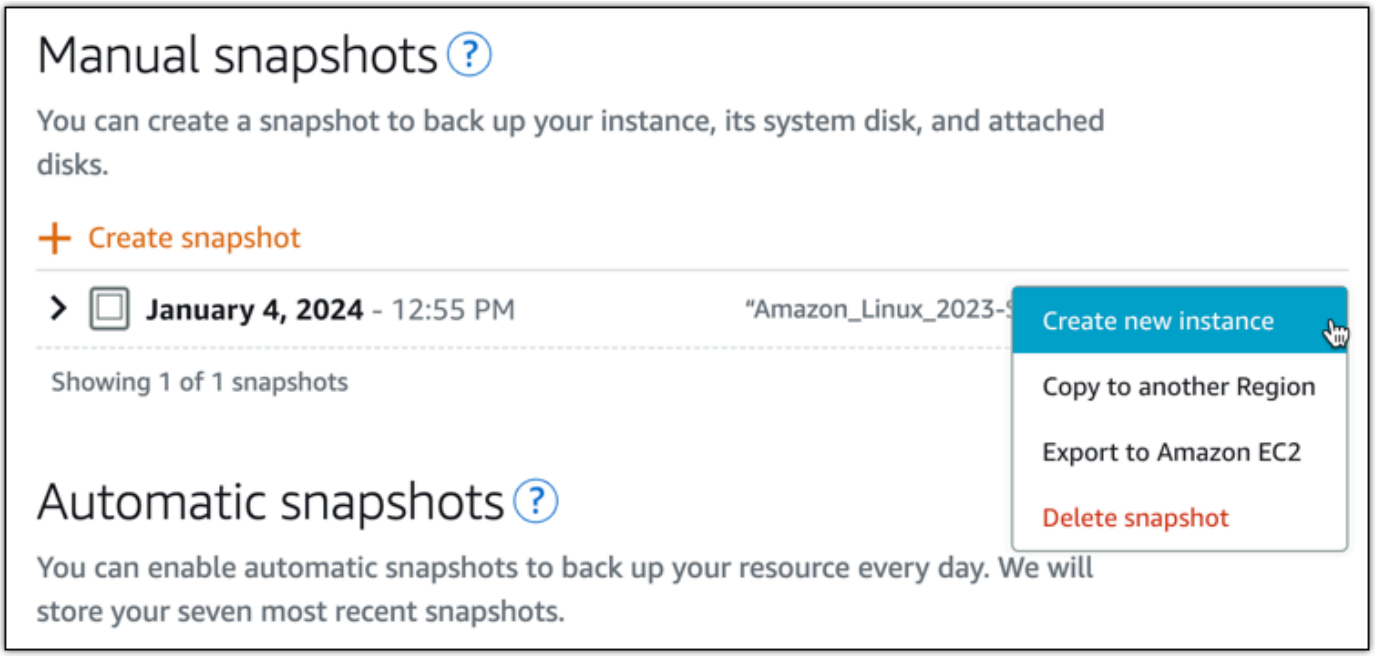


スナップショットから Lightsail インスタンスを作成する

Lightsail でスナップショットを作成した後に、そのスナップショットから新しいインスタンスを作成できます。インスタンスサイズやネットワークタイプなど、新しいインスタンスの属性をデュアルスタックまたは IPv6 専用で変更できます。新しいインスタンスには、システムディスクと追加したアタッチ済みのブロックストレージディスクが含められます。

あるスナップショットから別のインスタンスを作成するには、インスタンスのスナップショットが必要です。詳細については、スナップショットを使用して Linux/Unix Lightsail インスタンスをバックアップするまたはLightsail Windows Server インスタンスのスナップショットを作成するを参照してください。

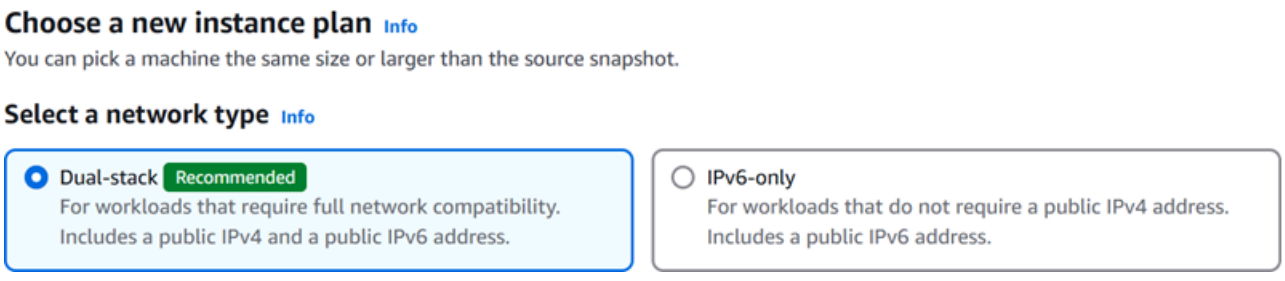
1. Lightsail コンソールで、スナップショットを作成して新しいインスタンスを作成するインスタンスを選択します。
2. [スナップショット] タブを選択します。
3. このページの [手動スナップショット] セクションで、新しいインスタンスを作成するスナップショットの横にあるアクションメニューアイコン (:) を選択してから、[新しいインスタンスの作成] を選択します。



4. [スナップショットからのインスタンスの作成] ページが表示されます。使用したい選択可能な設定を選択します。アベイラビリティーゾーンの変更、起動スクリプトの追加、インスタンスへの接続方法の変更などを行うことができます。

5. 新しいインスタンスのプラン (またはバンドル) を選択します。デュアルスタック (IPv4 および IPv6) インスタンスプランを使用するインスタンスを作成するか、IPv6 専用のプランを使用するインスタンスを作成するかを選択できます。また、元のインスタンスよりも大きなバンドルサイズを選択することもできます。IPv6 専用のインスタンスプランの詳細については、「Lightsail インスタンスの IPv6 専用ネットワークを設定する」を参照してください。

> **Note**
>
> 元のインスタンスよりも小さいバンドルサイズを使用して、インスタンスを作成することはできません。



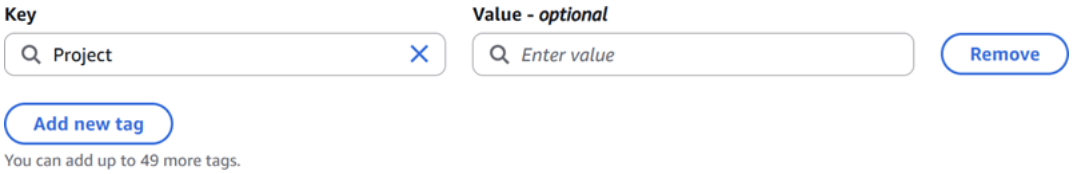
6. インスタンスの名前を入力します。

リソース名:

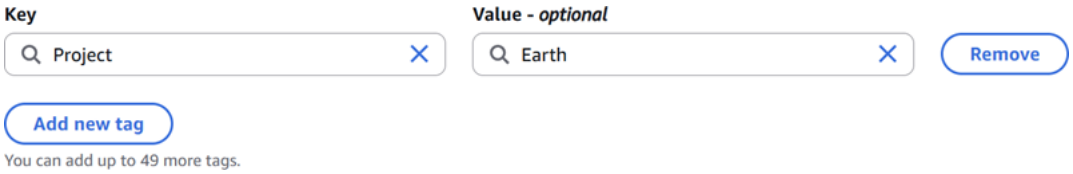
- AWS リージョン Lightsail アカウントごとに一意である必要があります。
- 2～255 文字を使用する必要があります。
- 先頭と末尾は英数字や文字を使用する必要があります。
- 英数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

7. (オプション)[新しいタグの追加] を選択してインスタンスにタグを追加します。必要に応じてこの手順を繰り返して、他のタグを追加します。タグの使用の詳細については、「タグ」を参照してください。

   a. [キー] には、タグキーを入力します。

   b. (オプション) [値] にタグの値を入力します。

8. [インスタンスの作成] を選択します。

   Lightsail を使用すると管理ページが開き、新しいインスタンスを管理できます。

> **Important**
>
> 元のインスタンスからのカスタムファイアウォールルールは、スナップショットから作成した新しいインスタンスにはコピーされません。デフォルトのルールのみが新しいインスタンスにコピーオーバーします。詳細については、このガイドの後半の「デフォルトのインスタンスのファイアウォールルール」を参照してください。

スナップショットから Lightsail インスタンス、ストレージ、またはデータベースをアップグレードする

これは、お客様のクラウドプロジェクトが増大して、より多くの処理能力がすぐに必要になった場合に必要です。その場合に、当社ではお客様を支援できます。Lightsail インスタンス、ブロックストレージディスク、またはデータベースのサイズを大きくするには、リソースのスナップショットを作成し、スナップショットを使用してそのリソースの新しく大きなバージョンを作成します。

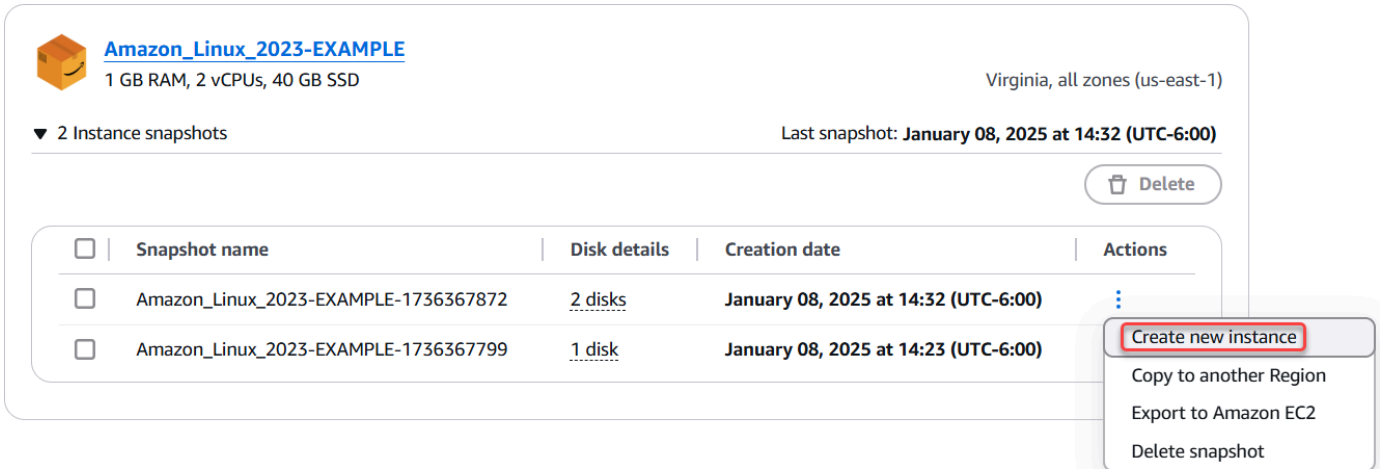
> **Note**
>
> 元のリソースよりも小さいプランサイズを使用して、スナップショットからリソースを作成することはできません。たとえば、8 GB のインスタンスから 2 GB のインスタンスに移行することはできません。
> インスタンスの作成時にインスタンスに割り当てられるデフォルトの公開 IPv4 アドレスは、インスタンスを停止してまた開始すると変更されます。必要に応じて、静的 IPv4 アドレスをインスタンスに作成し、アタッチできます。静的 IP アドレスを使用すると、アドレスをアカウント内の別のインスタンスに迅速に再マッピングすることで、インスタンスやソフトウェアの障害をマスクできます。または、ドメインがインスタンスを参照するように、ドメインの DNS レコードに静的 IP アドレスを指定することもできます。詳細については、「IP アドレス」を参照してください。

前提条件

Lightsail インスタンス、ブロックストレージディスク、またはデータベースのスナップショットが必要になります。詳細については、「スナップショット」を参照してください。

リソースを作成する

1. Lightsail コンソール にサインインします。
2. [スナップショット] タブを選択します。
3. 新しく大きなリソースを作成するためにスナップショットを使用する Lightsail リソースを見つけ、右矢印を選択してスナップショットのリストを展開します。
4. 使用するスナップショットの横にある省略記号アイコンを選択し、[新規インスタンスを作成] を選択します。



5. [作成] ページには、選択可能なオプション設定がいくつかあります。たとえば、アベイラビリティーゾーンを変更できます。インスタンスの場合は、起動スクリプトを追加したり、接続に使用する SSH キーを変更したりできます。

   すべてのデフォルト値をそのまま使用して、次のステップに進むことができます。

6. 新しいリソースのプラン (またはバンドル) を選択します。この時点で、必要に応じて、元のリソースよりも大きなバンドルサイズを選択できます。

   > **Note**
   >
   > 元のリソースよりも小さなプランサイズを使用してリソースを作成することはできません。元のリソースよりも小さなバンドルオプションは使用できません。

7. インスタンスの名前を入力します。

   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

8. [作成] を選択します。

   Lightsail では新しいリソースの管理ページに移動し、管理を開始できます。

Lightsail を使用してスナップショットからより大きなインスタンス、ブロックストレージディスク、またはデータベースを作成する AWS CLI

これは、お客様のクラウドプロジェクトが増大して、より多くの処理能力がすぐに必要になった場合に必要です。その場合に、当社ではお客様を支援できます。Lightsail コンソール内から、または AWS Command Line Interface (AWS CLI) を使用してすべてを実行できます。

現在の Lightsail インスタンスのスナップショットを作成し、そのスナップショットに基づいてお客様が必要とする処理能力を持つ、より大きいサイズの新規インスタンスを作成する方法を提示します。

> **Note**
>
> 現時点では、スナップショットより小さいサイズ (またはバンドル) のインスタンスの作成はサポートされていません。作成できるのは、同じサイズまたはより大きいサイズのインスタンスのみです。

前提条件

1. まず、まだインストールしていない場合は、 をインストールする必要があります AWS CLI。詳細については、「AWS Command Line Interfaceのインストール」を参照してください。AWS CLIを設定する必要があります。
2. 作業するインスタンスのスナップショットも必要です。詳細については、「Linux または Unix インスタンスのスナップショットを作成する」を参照してください 。

ステップ 1: スナップショット名を取得する

明らかだと思われるかもしれませんが、この AWS CLI コマンドを実行してより大きいインスタンスを作成する前に、スナップショット名を知っている必要があります。幸いなことに、その名前は簡単に取得できます。

1. に AWS CLI、次のように入力します。

```
aws lightsail get-instance-snapshots
```

次のような出力が表示されます。

```
{
    "instanceSnapshots": [
        {
            "fromInstanceName": "WordPress-512MB-EXAMPLE",
            "name": "WordPress-512MB-EXAMPLE-system-1234567891011",
            "sizeInGb": 20,
            "resourceType": "InstanceSnapshot",
            "fromInstanceArn":
            "arn:aws:lightsail:us-
east-1:123456789101:Instance/86f49ee4-26cc-4802-9b0d-12345EXAMPLE",
            "state": "available",
            "arn": "arn:aws:lightsail:us-east-1:123456789101:InstanceSnapshot/
c87acb5f-851e-4fbc-94f1-12345EXAMPLE",
            "fromBundleId": "nano_1_0",
            "fromBlueprintId": "wordpress_4_6_1",
            "createdAt": 1480898073.653,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-east-2"
            }
        }
    ]
}
```

2. [name] の値を、後で見つけやすい場所にコピーします。この名前は、 AWS CLI コマンドで `--instance-snapshot-name` の値として使用します。

ステップ 2: バンドルを選択する

バンドルは、実際にはインスタンスの料金プランおよび設定です。たとえば、Linux ベースの Medium バンドルは、1 か月あたり 24 米ドルの料金で、4.0 GB の RAM、80 GB の SSD ストレージなどが設定されています。

小さいバンドルから始めて、処理能力の追加が必要になった場合は、より大きなバンドルにアップグレードできます。詳細については、「スナップショットからより大きなインスタンス、ブロックストレージディスク、またはデータベースを作成する」を参照してください。

> **Important**
>
> スナップショットからより小さいサイズのバンドルに変更することはできません。より小さいバンドルを作成する場合は、最初からやり直す必要があります。

1. 次のコマンドを入力します AWS CLI 。

```
aws lightsail get-bundles
```

出力は次のようになります。

```
{
    "bundles": [
        {
            "price": 5.0,
            "cpuCount": 2,
            "diskSizeInGb": 20,
            "bundleId": "nano_3_0",
            "instanceType": "nano",
            "isActive": true,
            "name": "Nano",
            "power": 298,
            "ramSizeInGb": 0.5,
            "transferPerMonthInGb": 1024,
            "supportedPlatforms": [
                "LINUX_UNIX"
            ],
            },
        {
            "price": 7.0,
            "cpuCount": 2,
            "diskSizeInGb": 40,
            "bundleId": "micro_3_0",
            "instanceType": "micro",
            "isActive": true,
            "name": "Micro",
            "power": 500,
            "ramSizeInGb": 1.0,
            "transferPerMonthInGb": 2048,
            "supportedPlatforms": [
                "LINUX_UNIX"
```

```
            ],
            },
        {
            "price": 12.0,
            "cpuCount": 2,
            "diskSizeInGb": 60,
            "bundleId": "small_3_0",
            "instanceType": "small",
            "isActive": true,
            "name": "Small",
            "power": 1000,
            "ramSizeInGb": 2.0,
            "transferPerMonthInGb": 3072,
            "supportedPlatforms": [
                "LINUX_UNIX"
            ],
            },
        {
            "price": 24.0,
            "cpuCount": 2,
            "diskSizeInGb": 80,
            "bundleId": "medium_3_0",
            "instanceType": "medium",
            "isActive": true,
            "name": "Medium",
            "power": 2000,
            "ramSizeInGb": 4.0,
            "transferPerMonthInGb": 4096,
            "supportedPlatforms": [
                "LINUX_UNIX"
            ],
            },
        {
            "price": 44.0,
            "cpuCount": 2,
            "diskSizeInGb": 160,
            "bundleId": "large_3_0",
            "instanceType": "large",
            "isActive": true,
            "name": "Large",
            "power": 3000,
            "ramSizeInGb": 8.0,
            "transferPerMonthInGb": 5120,
            "supportedPlatforms": [
```

```
                "LINUX_UNIX"
            ],
            },
    ]
}
```

2. 目的のバンドルの [bundleId] の値を見つけます。詳細については、Lightsail の料金を参照してください。

ステップ 3: AWS CLI コマンドを記述して新しいインスタンスを作成する

これで、パラメーター値が取得済みであるため、インスタンスを作成するためのコマンドを記述して実行する準備ができました。

1. 次の内容を入力します。

```
aws lightsail create-instances-from-snapshot --instance-names
 MyNewInstanceFromSnapshot --availability-zone us-east-1a --instance-snapshot-name
 WordPress-512MB-EXAMPLE-system-1234567891011 --bundle-id medium_1_0
```

出力は次のようになります。

```
{
    "operations": [
        {
            "status": "Started",
            "resourceType": "Instance",
            "isTerminal": false,
            "statusChangedAt": 1486863990.961,
            "location": {
                "availabilityZone": "us-east-2a",
                "regionName": "us-east-2"
            },
            "operationType": "CreateInstance",
            "resourceName": "MyNewInstanceFromSnapshot",
            "id": "30fec45e-e7d7-4e18-96c8-12345EXAMPLE",
            "createdAt": 1486863989.784
        }
    ]
}
```

> **Note**
>
> を使用してリージョンとアベイラビリティーゾーンのリストを返すこともできます AWS CLI。`aws lightsail get-regions --include-availability-zones` リクエストでアベイラビリティーゾーンのリストを返すには、`get-regions` と入力します。

2. Lightsail コンソールで新しいインスタンスを開き、インスタンスの変更を開始します。

次の手順

スナップショットから新しいインスタンスを作成した後に、次のことを行うことができます。

- 以前のインスタンスを使用しない場合は削除できます。これを行うには、Lightsail コンソールまたは CLI コマンド delete-instance を使用します。
- 以前のスナップショットが必要ない場合は削除できます。これを行うには、Lightsail コンソールまたは CLI コマンド delete-instance-snapshot を使用します。
- 以前のインスタンスに静的 IP アドレスがアタッチされていた場合は、そのまま新しいインスタンスにアタッチできます。これを行うには、コンソールを使用します。「静的 IP を作成してインスタンスにアタッチする」を参照してください。

未使用の Lightsail スナップショットを削除して月額料金を回避する

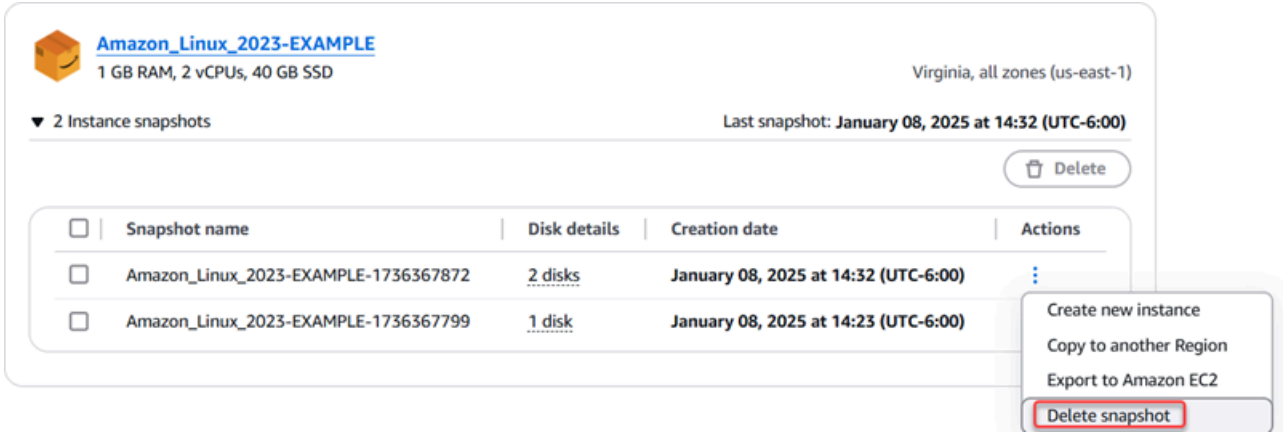
月額料金が発生しないように、不要になった Amazon Lightsail のインスタンス、データベース、ディスクスナップショットは削除します。

個々のスナップショットを削除する

> **Important**
>
> これは永続的オペレーションで、取消すことはできません。削除するとスナップショット上のすべてのデータが失われます。

1. Lightsail コンソールで、[スナップショット] タブを選択します。

2. スナップショットを削除する Lightsail リソースを見つけ、右矢印を選択して、そのリソースで使用可能なスナップショットのリストを展開します。
3. 削除するスナップショットの横にあるアクションメニューアイコン (⋮) を選択してから、[スナップショットの削除] を選択します。

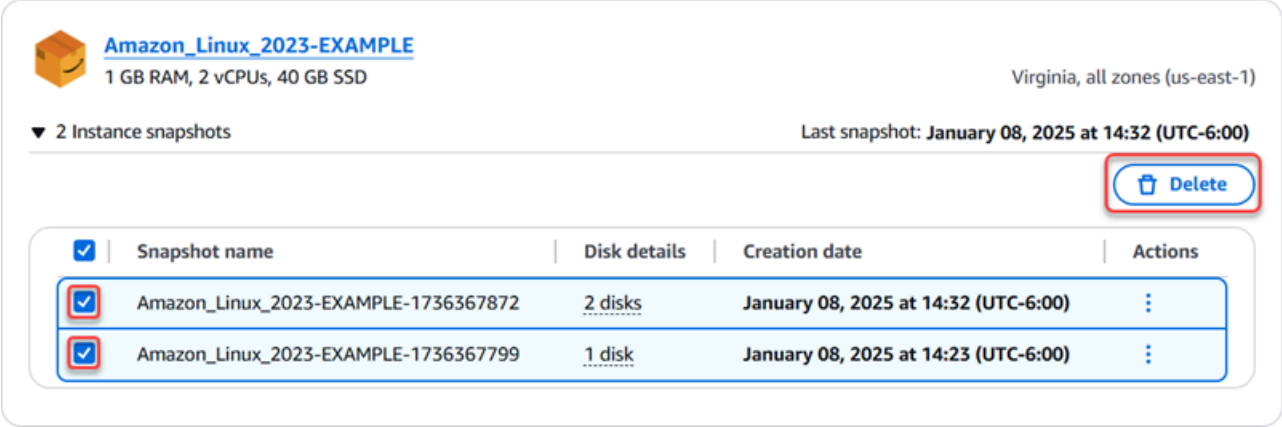


4. [はい] を選択して、スナップショットを削除することを確定します。

複数のスナップショットを削除する

> **Important**
>
> これは永続的オペレーションで、取消すことはできません。削除するとスナップショット上のすべてのデータが失われます。

1. Lightsail のホームページで [スナップショット] を選択します。
2. 削除するスナップショットの Lightsail リソースを探して、リソースのスナップショットセクションを展開します。
3. 削除するリソースのスナップショットを選択して、[削除] を選択します。



4. [はい] を選択して、スナップショットを削除することを確定します。

間でLightsailスナップショットをコピーする AWS リージョン

ではAmazon Lightsail、インスタンススナップショットとブロックストレージディスクスナップショット AWS リージョン を 間で、または同じリージョン内でコピーできます。例えば、あるリージョンでリソースを作成して設定した後、別のリージョンの方が適切だと判断した場合、リージョン間でスナップショットをコピーすることができます。複数のリージョン間でリソースを複製することもできます。

前提条件

コピーする Lightsail インスタンスまたはブロックストレージディスクのスナップショットを作成します。詳細については、以下のいずれかのガイドを参照してください。

- Linux または Unix インスタンスのスナップショットを作成する
- Windows Server インスタンスのスナップショットを作成する
- ブロックストレージディスクのスナップショットを作成する

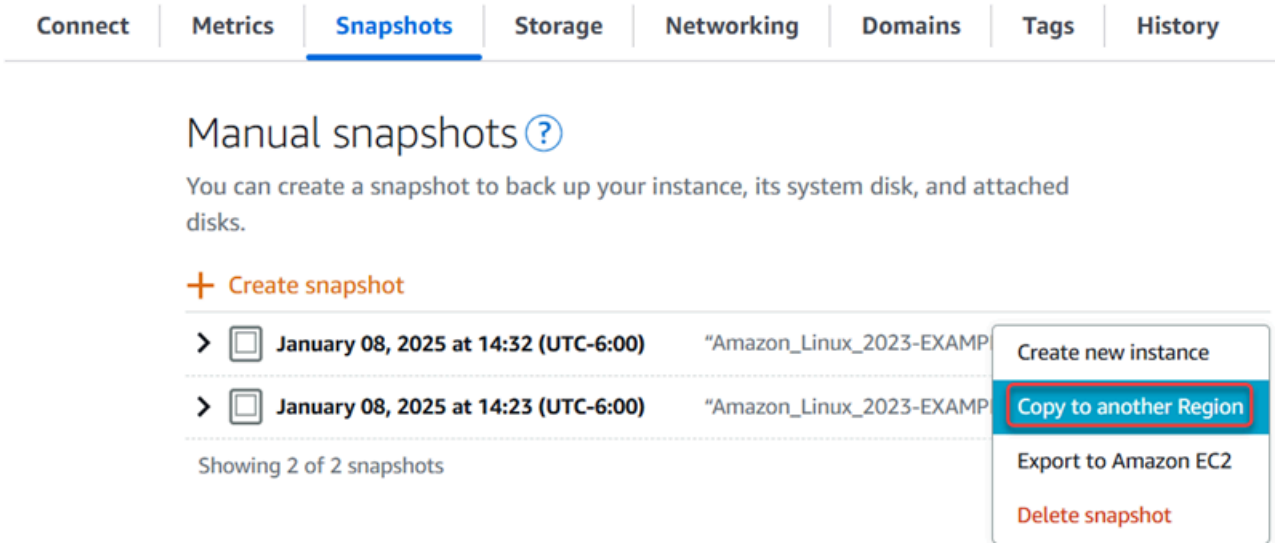
のスナップショットをコピーする

Lightsail インスタンスのスナップショットやブロックストレージディスクのスナップショットを、2 つの AWS リージョン 間または同じリージョン内でコピーできます。

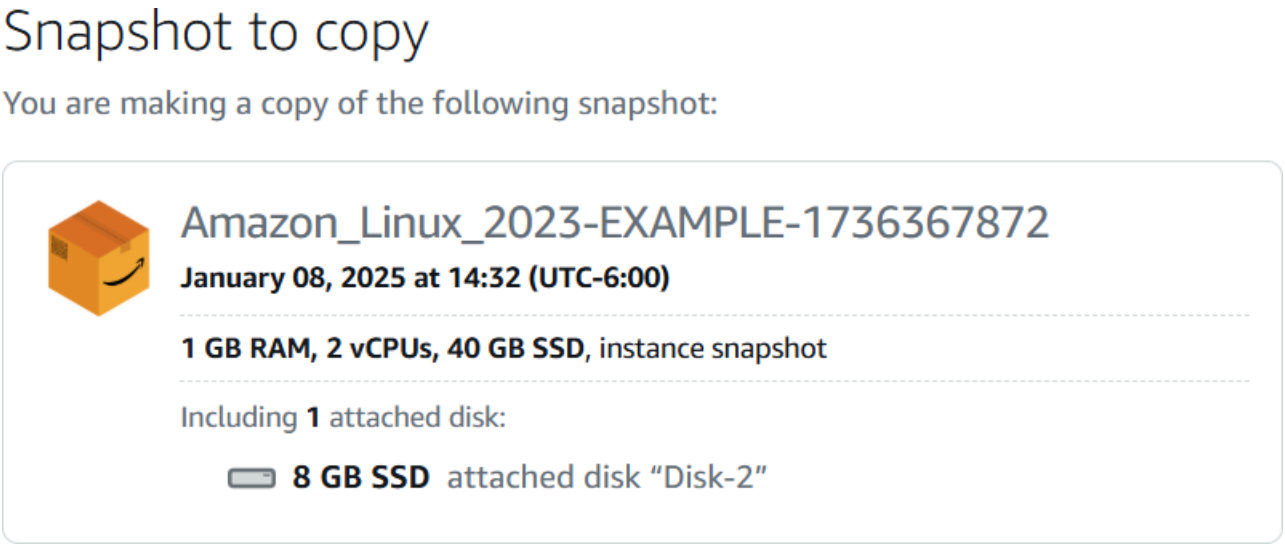
Lightsail スナップショットをコピーするには

1. Lightsail コンソール にサインインします。

2. Lightsail のホームページで [スナップショット] タブを選択します。
3. コピーするインスタンスまたはブロックストレージディスクを見つけてノードを展開し、そのリソースで使用可能なスナップショットを表示します。
4. 目的のスナップショットのアクションメニューアイコン (⋮) を選択し、[Copy to another Region (別のリージョンにコピー)] を選択します。



5. [Copy a snapshot (スナップショットをコピーする)] ページの [Snapshot to copy (コピーするスナップショット)] 部分で、表示されているスナップショットの詳細がコピー元のインスタンスやブロックストレージディスクの仕様と一致していることを確認します。

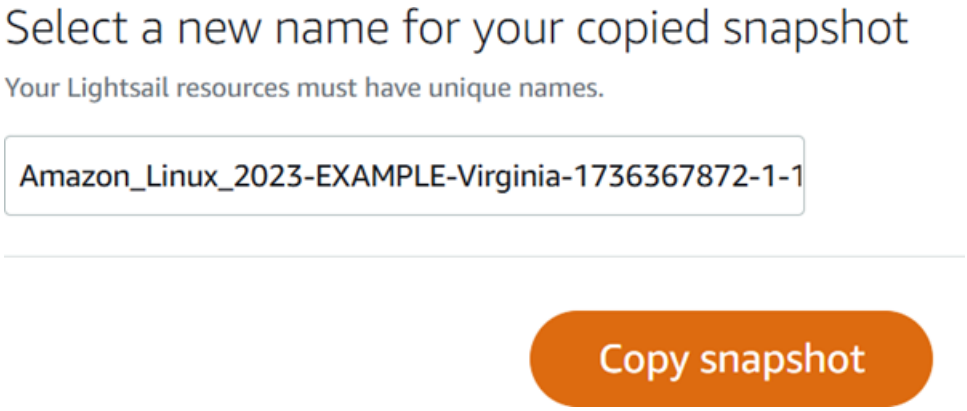


6. このページの [リージョンの選択] セクションで、スナップショットのコピー先のリージョンを選択します。
7. スナップショットコピーの名前を入力します。

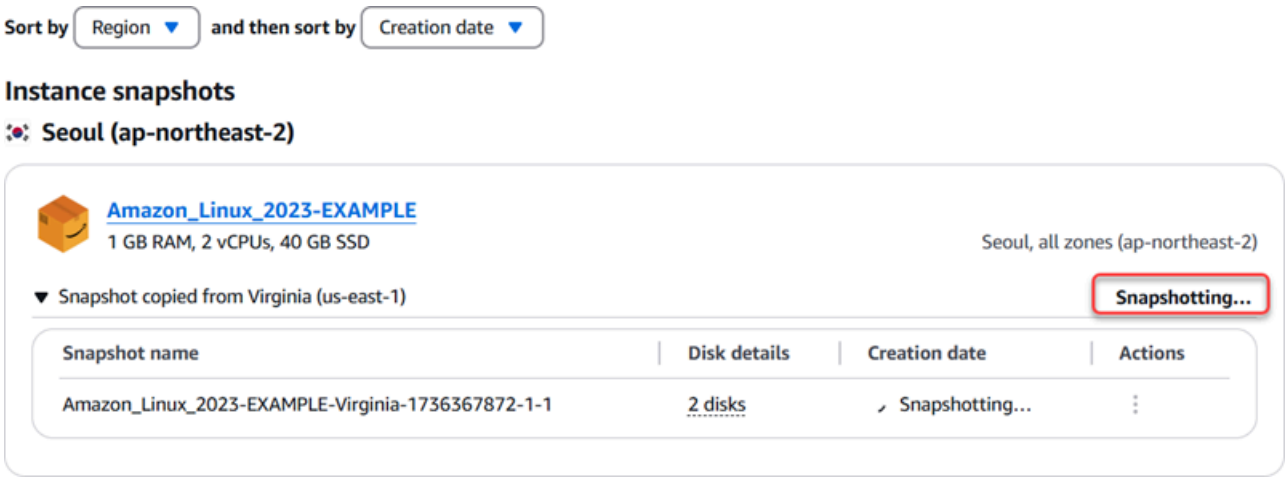
リソース名:

- Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
- 2〜255 文字を使用する必要があります。
- 先頭と末尾は英数字または数字を使用する必要があります。
- 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用できます。

8. [スナップショットをコピー] を選択します。



スナップショットのコピーはまもなく利用可能になります。所要時間は、ソースインスタンスのサイズと設定によります。左側のナビゲーションペインで [スナップショット] タブに移動してスナップショットのステータスを確認することで、スナップショットコピーのステータスを確認できます。次の画像で示されるように、[スナップショット作成中...] のステータスが表示されます。プロセスが完了してスナップショットを使用する準備ができたら、[コピー元] のタイムスタンプが表示されます。



次の手順

Lightsail の別のリージョンにスナップショットをコピーしたあと、追加で実行できるステップがいくつかあります。

- コピーしたスナップショットが利用可能になったら、このスナップショットから新しいインスタンスを作成します。詳細については、「スナップショットからインスタンスを作成する」を参照してください。
- コピー元のスナップショットが不要になった場合は、削除します。さもなければ、スナップショットの保存料金が発生します。

Lightsail のスナップショットを Amazon EC2 にエクスポートする方法について説明する

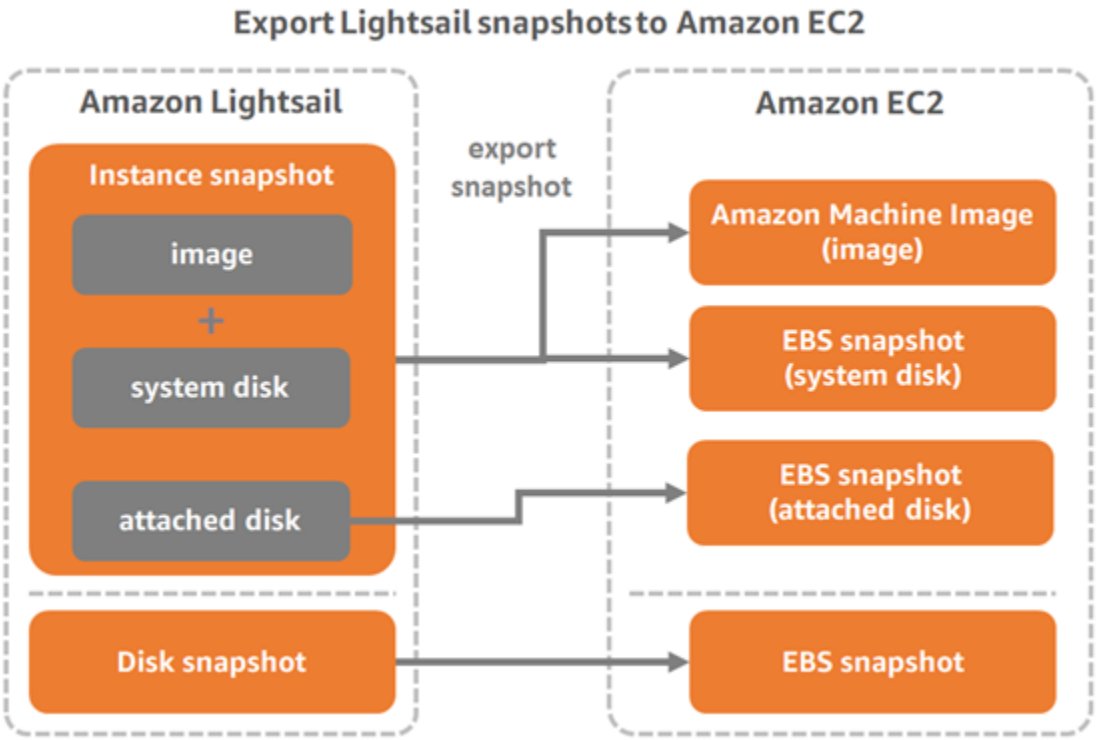
Amazon EC2 への Lightsail スナップショットのエクスポート、エクスポートされたスナップショットからの EC2 リソースの作成、互換性のある EC2 インスタンスタイプの選択、EC2 インスタンスへの接続、Lightsail スナップショットから作成された EC2 インスタンスの保護を行うことができます。Amazon Lightsail インスタンスおよびブロックストレージのディスクスナップショットは、次のいずれかの方法によって Amazon Elastic Compute Cloud (Amazon EC2) にエクスポートできます。

- Lightsail コンソール。詳細については、「スナップショットを Amazon EC2 にエクスポートする」を参照してください。
- Lightsail API、 AWS Command Line Interface (AWS CLI)、または SDKs。詳細については、Lightsail API ドキュメントの ExportSnapshot オペレーション、または AWS CLI ドキュメントの export-snapshot コマンドを参照してください。

インスタンスおよびブロックストレージディスクのスナップショットをエクスポートできます。ただし、cPanel & WHM (CentOS 7) インスタンスのスナップショットは Amazon EC2 にエクスポートできません。スナップショットは、同じ Lightsail の AWS リージョン から Amazon EC2 へとエクスポートされます。スナップショットを別のリージョンにエクスポートするには、まず Lightsail でスナップショットを別のリージョンにコピーしてからエクスポートを実行します。詳細については、「Copy snapshots from one AWS リージョン to another」を参照してください。

Lightsail インスタンスのスナップショットをエクスポートすると、Amazon マシンイメージ (AMI) および Amazon Elastic Block Store (Amazon EBS) のスナップショットが Amazon EC2 に作成されます。これは、Lightsail インスタンスはイメージとシステムディスクで構成されており、より効率

的な管理のために Lightsail コンソールで 1 つのインスタンスエンティティとしてグループ化されているためです。スナップショットの作成時に、ソースの Lightsail インスタンスに 1 つ以上のブロックストレージディスクがアタッチされている場合、アタッチされているディスクごとに追加の EBS スナップショットが Amazon EC2 に作成されます。Lightsail のブロックストレージディスクのスナップショットをエクスポートすると、1 つの EBS スナップショットが Amazon EC2 に作成されます。Amazon EC2 のすべてのエクスポートしたリソースには、Lightsail の対応するリソースとは異なる独自の一意な識別子が割り当てられます。



> **Note**
>
> Lightsail は、 AWS Identity and Access Management (IAM) サービスにリンクされたロール (SLR) を使用して、スナップショットを Amazon EC2 にエクスポートします。SLR の詳細については、「サービスにリンクされたロール」を参照してください。

エクスポートプロセスは時間がかかる場合があります。所要時間は、ソースのインスタンスやブロックストレージディスクのサイズと設定に応じて異なります。Lightsail コンソールの [エクスポート] を使用して、エクスポートのステータスを追跡します。詳細については、「Lightsail でスナップショットのエクスポートステータスを追跡する」を参照してください。

エクスポートした Lightsail スナップショットから Amazon EC2 リソースを作成する

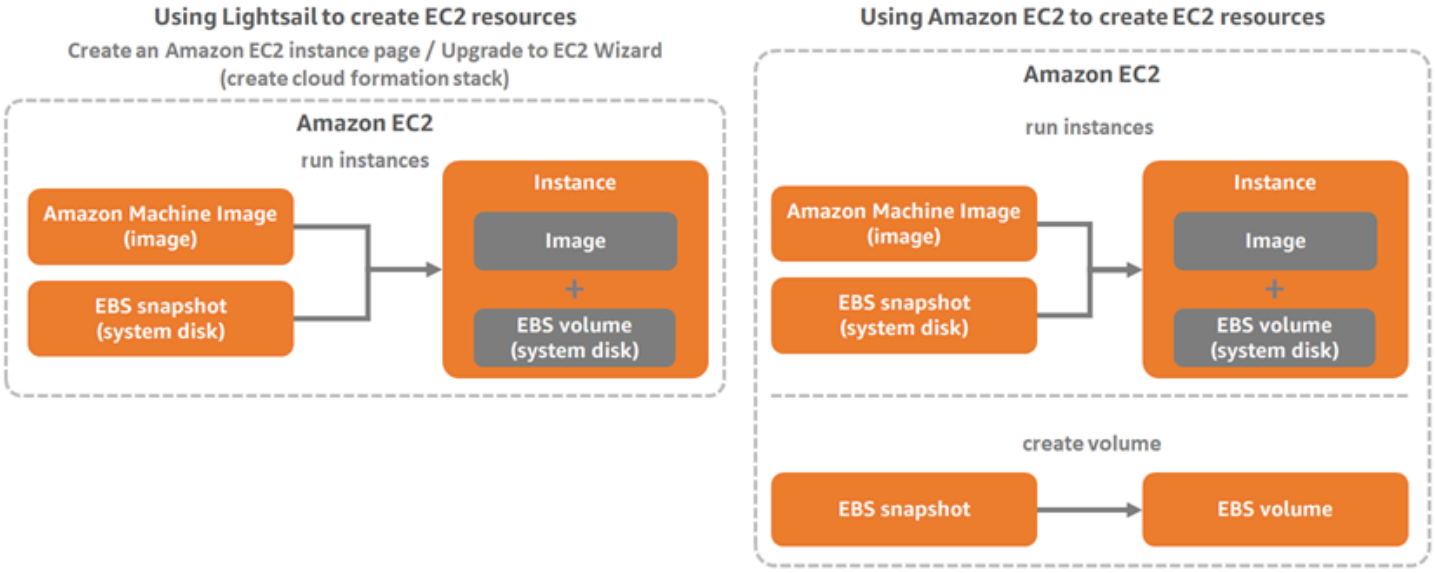
Lightsail スナップショットをエクスポートして、(AMI、EBS スナップショット、または両方として) Amazon EC2 で使用可能にした後は、以下のいずれかの方法を使用して、スナップショットから Amazon EC2 リソースを作成できます。

- Lightsail コンソールの「Amazon EC2 インスタンスを作成する」ページは、「Amazon EC2 へのアップグレードウィザード」とも呼ばれます。詳細については、「エクスポートしたスナップショットから Amazon EC2 インスタンスを作成する」を参照してください。
- Lightsail API、 AWS CLI、または SDKs。詳細については、 Lightsail API ドキュメントの CreateCloudFormationStack オペレーション、または AWS CLI ドキュメントの create-cloud-formation-stack コマンドを参照してください。

  > **Note**
  >
  > Lightsail では、エクスポートしたインスタンスから Amazon EC2 インスタンスを作成できますが、エクスポートしたブロックストレージディスクスナップショットから EBS ボリュームを作成することはできません。この目的には、Amazon EC2 コンソール、API、または AWS CLIを使用する必要があります。詳細については、「エクスポートしたディスクスナップショットから Amazon EBS ボリュームを作成する」を参照してください。

- Amazon EC2 コンソール、Amazon EC2 API AWS CLI、または SDKs。詳細については、Amazon EC2 ドキュメントの「インスタンス起動ウィザードを使用したインスタンスの起動」または「スナップショットからの Amazon EBS ボリュームの復元」を参照してください。

エクスポートしたインスタンススナップショット (AMI および EBS スナップショット) から Amazon EC2 インスタンスを作成すると、1 つの EC2 インスタンスが起動されます。Lightsail インスタンススナップショットのエクスポートで生成される AMI および EBS スナップショットは、相互に自動的にリンクされて EC2 インスタンスを形成します。Amazon EC2 で EBS ボリュームを作成するには、エクスポートした Lightsail ブロックストレージディスクのスナップショット (EBS スナップショット) を使用できます。



> **Note**
>
> Lightsail では、CloudFormation スタックを使用してインスタンスおよび関連リソースを EC2 に作成します。詳細については、「Lightsail のCloudFormation スタック」を参照してください。

エクスポートしたスナップショットから Amazon EC2 リソースを作成するプロセスは時間がかかる場合があります。所要時間は、ソースインスタンスのサイズと設定によります。Lightsail コンソールの [エクスポート] を使用して、エクスポートのステータスを追跡します。詳細については、「」を参照してくださいLightsail でスナップショットのエクスポートステータスを追跡する。

Amazon EC2 インスタンスタイプを選択する

Amazon EC2 が提供するインスタンスオプションは、Lightsail で使用可能なオプションよりも充実しています。Amazon EC2 では、コンピューティング (C5)、メモリ (R5)、両方 (T3 および M5) に最適化されたインスタンスタイプを選択できます。Lightsail により、これらのオプションは [Amazon EC2 インスタンスの作成] ページで提供されます。ただし、Amazon EC2 を使用してエクスポートされたスナップショットから新しいインスタンスを作成する場合、より多くのインスタンスタイプのオプションが利用できます。EC2 インスタンスタイプの詳細については、Amazon EC2 ドキュメントの「インスタンスタイプ」を参照してください。

エクスポートしたスナップショットから EC2 インスタンスを作成する前に、Lightsail と Amazon EC2 のインスタンス料金の違いを確認することが重要です。インスタンスの料金の詳細については、「Lightsail の料金」と「Amazon EC2 の料金」のページを参照してください。

Lightsail と Amazon EC2 のインスタンスタイプの互換性

一部の Lightsail インスタンスは、拡張ネットワーキングに対応していないため、現行世代の EC2 インスタンスタイプ (T3、M5、C5、または R5) と互換性がありません。ソースの Lightsail インスタンスに互換性がない場合は、エクスポートしたスナップショットから EC2 インスタンスを作成するときに、以前の世代のインスタンスタイプ (T2、M4、C4、または R4) から選択する必要があります。これらのオプションは、Lightsail コンソールの「Amazon EC2 インスタンスを作成する」ページを使用して、EC2 インスタンスを作成するときに表示されます。

ソースの Lightsail インスタンスに互換性がない場合に最新世代の EC2 インスタンスタイプを使用するには、まず以前の世代のインスタンスタイプ (T2、M4、C4、または R4) を使用して新しい EC2 インスタンスを作成し、ネットワーキングドライバーを更新します。次に、インスタンスを目的の現行世代のインスタンスタイプに更新します。詳細については、「Amazon EC2 インスタンスの拡張ネットワーキング」を参照してください。

Amazon EC2 インスタンスに接続する

Lightsail インスタンスに接続する場合と同様の方法で Amazon EC2 インスタンスに接続できます。つまり、Linux および Unix インスタンスには SSH を使用し、Windows Server インスタンスには RDP を使用します。ただし、Lightsail コンソールでブラウザベースの SSH/RDP クライアントを使用した場合、使用するブラウザのバージョンによっては、このクライアントを Amazon EC2 で使用できない場合があります。この場合は、独自の SSH/RDP クライアントを設定して EC2 インスタンスに接続する必要があります。詳細については、以下のガイドを参照してください。

- Lightsail スナップショットから作成された Amazon EC2 の Linux または Unix インスタンスに接続する
- Lightsail スナップショットから作成された Amazon EC2 の Windows Server インスタンスに接続する

Amazon EC2 インスタンスを保護する

エクスポートした Lightsail スナップショットから EC2 インスタンスを作成した後で、必要に応じて、いくつかのアクションを実行して新しいインスタンスのセキュリティを強化します。それらのアクションは、EC2 インスタンスのオペレーティングシステムによって異なります。

Amazon EC2 での Linux および Unix インスタンスの保護

EC2 (EC2 コンソール、EC2 API、EC2 用、または EC2 用 SDK) を使用してエクスポートされたスナップショットから Amazon EC2 EC2 で Linux または Unix インスタンスを作成すると、新しい

EC2 インスタンスにLightsailサービスからの残余 SSH キーが含まれる場合があります。 EC2 AWS CLI EC2 SDKs これらのキーを削除して新しいインスタンスのセキュリティを強化することをお勧めします。

詳細については、「Lightsail スナップショットから作成した Amazon EC2 の Linux または Unix インスタンスを保護する」を参照してください。

Amazon EC2 の Windows Server インスタンスの保護

エクスポートされたスナップショットから Amazon EC2 で Windows Server インスタンスを作成するLightsailと、 および EC2 へのアクセス権を持つ AWS アカウントのユーザーは、ソースインスタンスに最初に割り当てられたデフォルトの管理者パスワードを取得できます。これは、新しい EC2 インスタンスのパスワードでもあります。セキュリティを強化するために、Amazon EC2 インスタンスのデフォルトの管理者パスワードを変更することをお勧めします (まだ変更していない場合)。

詳細については、「Lightsail スナップショットから作成した Amazon EC2 の Windows Server インスタンスを保護する」を参照してください。

Lightsail スナップショットを Amazon EC2 にエクスポートする

Amazon Lightsail インスタンスおよびブロックストレージディスクのスナップショットを Amazon Elastic Compute Cloud (Amazon EC2) にエクスポートできます。Lightsail インスタンスのスナップショットをエクスポートすると、Amazon マシンイメージ (AMI) および Amazon Elastic Block Store (Amazon EBS) のスナップショットが Amazon EC2 に作成されます。これは、Lightsail インスタンスはイメージとシステムディスクで構成されており、より効率的な管理のために Lightsail コンソールで 1 つのインスタンスエンティティとしてグループ化されているためです。スナップショットの作成時に、ソースの Lightsail インスタンスに 1 つ以上のブロックストレージディスクがアタッチされている場合、アタッチされているディスクごとに追加の EBS スナップショットが Amazon EC2 に作成されます。

Lightsail のブロックストレージディスクのスナップショットをエクスポートすると、1 つの EBS スナップショットが Amazon EC2 に作成されます。Amazon EC2 のすべてのエクスポートしたリソースには、Lightsail の対応するリソースとは異なる独自の一意な識別子が割り当てられます。

このガイドでは、Lightsail スナップショットをエクスポートする方法、エクスポートのステータスを追跡する方法、およびエクスポートしたスナップショットを Amazon EC2 で (AMI、EBS スナップショット、またはその両方として) 利用可能にした後のステップについて説明します。

> **Important**
>
> このガイドの手順を実行する前に、Lightsail のエクスポートプロセスを再確認することをお勧めします。詳細については、「スナップショットを Amazon EC2 にエクスポートする」を参照してください。

目次

- Lightsail スナップショットをエクスポートするために必要なサービスにリンクされたロールおよび IAM アクセス許可
- 前提条件
- Amazon EC2 に Lightsail スナップショットをエクスポートする
- タスクのステータスを追跡する

Lightsail スナップショットをエクスポートするために必要なサービスにリンクされたロールおよび IAM アクセス許可

Lightsail は、 AWS Identity and Access Management (IAM) サービスにリンクされたロール (SLR) を使用して、スナップショットを Amazon EC2 にエクスポートします。SLR の詳細については、「サービスにリンクされたロール」を参照してください。

スナップショットのエクスポートを実行するユーザーによっては、必要に応じて以下の追加のアクセス許可を IAM で設定します。

- Amazon アカウントのルートユーザーがエクスポートを実行する場合は、このガイドの「前提条件」セクションに進みます。アカウントルートユーザーは、スナップショットのエクスポートを実行するために必要なアクセス許可をすでに持っています。
- IAM ユーザーがエクスポートを実行する場合、 AWS アカウント管理者はユーザーに次のポリシーを追加する必要があります。ユーザーのアクセス権限を変更する方法については、IAM ドキュメント内の「IAM ユーザーのアクセス権限の変更」を参照してください。

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
```

```
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
lightsail.amazonaws.com/AWSServiceRoleForLightsail*",
            "Condition": {"StringLike": {"iam:AWSServiceName":
 "lightsail.amazonaws.com"}}
        },
        {
            "Effect": "Allow",
            "Action": "iam:PutRolePolicy",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
lightsail.amazonaws.com/AWSServiceRoleForLightsail*"
        }
    ]
}
```

前提条件

Amazon EC2 にエクスポートする Lightsail インスタンスまたはブロックストレージディスクのスナップショットを作成します。詳細については、以下のいずれかのガイドを参照してください。

- Linux または Unix インスタンスのスナップショットを作成する
- Windows Server インスタンスのスナップショットを作成する
- ブロックストレージディスクのスナップショットを作成する

Amazon EC2 に Lightsail スナップショットをエクスポートする

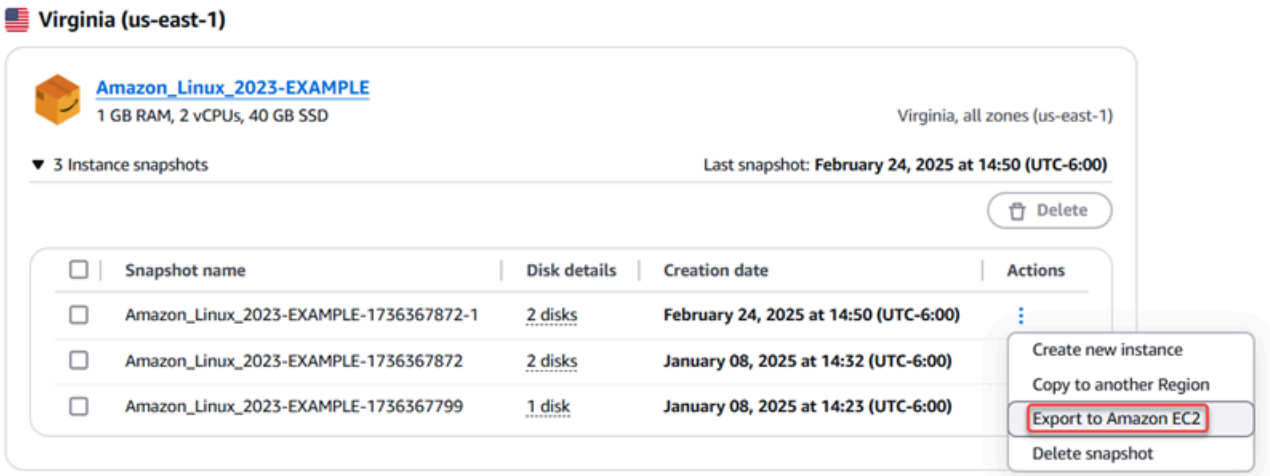
スナップショットを Amazon EC2 にエクスポートする最も効率的な方法は、Lightsail コンソールを使用することです。Lightsail API、 AWS Command Line Interface (AWS CLI)、または SDKs を使用してスナップショットをエクスポートすることもできます。詳細については、 Lightsail API ドキュメントの ExportSnapshot オペレーション、または AWS CLI ドキュメントの export-snapshot コマンドを参照してください。

> **Note**
>
> スナップショットは、 AWS リージョン から Amazon EC2 に同じ Lightsailにエクスポートされます。スナップショットを別のリージョンにエクスポートするには、まず Lightsail でス

ナップショットを別のリージョンにコピーしてからエクスポートを実行します。詳細については、「Copy snapshots from one AWS リージョン to another」を参照してください。

Amazon EC2 に Lightsail スナップショットをエクスポートする

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [スナップショット] を選択します。
3. エクスポートするインスタンスまたはブロックストレージディスクを見つけてノードを展開し、そのリソースの使用可能なスナップショットを表示します。
4. 目的のスナップショットの [アクション] メニューを選択し、[Amazon EC2 にエクスポート] を選択します。



> **Note**
>
> cPanel & WHM (CentOS 7) インスタンスのスナップショットは Amazon EC2 にエクスポートできません。

5. プロンプトに表示される重要な詳細情報を確認します。
6. Amazon EC2 にエクスポートすることに同意する場合は、[はい、続行します] を選択してプロセスを開始します。

   エクスポートプロセスは時間がかかる場合があります。所要時間は、ソースのインスタンスやブロックストレージディスクのサイズと設定に応じて異なります。Lightsail コンソールの [エクスポート] を使用して、エクスポートのステータスを追跡します。詳細については、「Lightsail でスナップショットのエクスポートステータスを追跡する」を参照してください。

タスクのステータスを追跡する

Lightsail コンソールの [エクスポート] セクションでエクスポートのステータスを追跡します。Lightsail コンソールの各ページの左側のナビゲーションペインからアクセスできます。詳細については、「Lightsail でスナップショットのエクスポートステータスを追跡する」を参照してください。

[エクスポート] では、以下の情報が表示されます:

- Snapshot name (スナップショット名) – ソースの Lightsail スナップショットの名前。
- ステータス - エクスポートのステータス。これは、`In progress`、`Successful`、または `Failed` です。
- Export started (エクスポートの開始日時) – スナップショットのエクスポートが開始された日付と時刻。
- ソースの詳細 – ソースの Lightsail インスタンスの仕様 (メモリ、処理、ストレージなど)。
- ソースインスタンス名 – スナップショットのソースインスタンスの名前。
- Snapshot type (スナップショットのタイプ) – Lightsail スナップショットのタイプ。インスタンススナップショットまたはディスクスナップショットのいずれかになります。
- Snapshot created (スナップショットの作成日時) – ソースの Lightsail スナップショットが作成された日付と時刻。

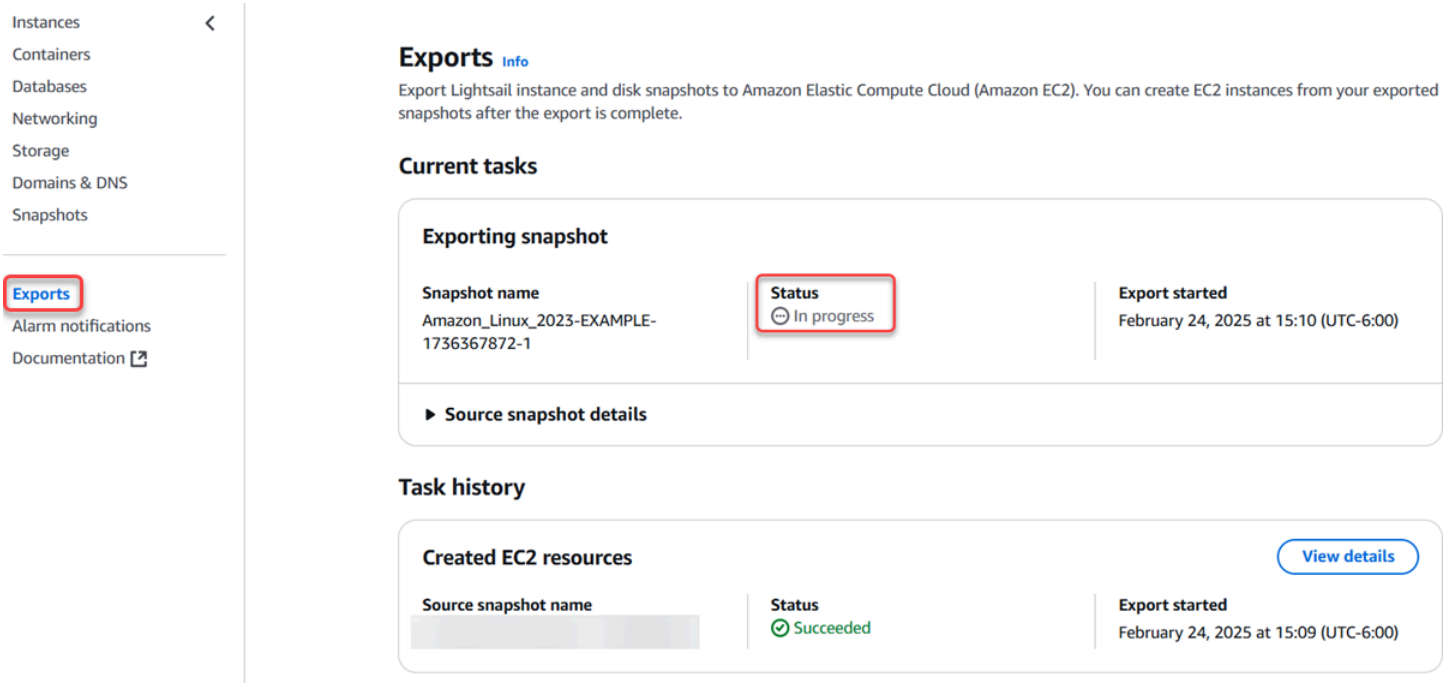
完了したエクスポートの以下の情報が [タスク履歴] セクションに表示されます:

- EC2 インスタンスの作成 – Lightsail コンソールを使用して Amazon EC2 で新しいインスタンスを作成する場合は、このオプションを選択します。詳細については、「エクスポートしたスナップショットから Amazon EC2 インスタンスを作成する」を参照してください。
- EC2 を開く – Amazon EC2 コンソールを使用して、エクスポートしたスナップショットから新しい EC2 リソースを作成する場合に、このオプションを選択します。Lightsail ブロックストレージディスクのスナップショットをエクスポートした場合は、Amazon EC2 を使用してスナップショット (EBS スナップショット) から EBS ボリュームを作成する必要があります。詳細については、Amazon EC2 ドキュメントの「インスタンス起動ウィザードを使用したインスタンスの起動」または「スナップショットからの Amazon EBS ボリュームの復元」を参照してください。

> **Note**
>
> ソースの Lightsail スナップショットが不要になった場合は、これを削除します。そうしないと、スナップショットの保存料金が発生します。

Lightsail でスナップショットのエクスポートステータスを追跡する

Amazon Lightsail コンソールの [エクスポート] セクションでは、Amazon EC2 への Lightsail スナップショットのエクスポートや、エクスポートしたインタンススナップショットからの新しい EC2 インスタンスの作成のステータスを追跡できます。エクスポートタスクの所要時間は、ソースインスタンスやブロックストレージディスクのサイズと設定によって異なります。Lightsail コンソールの各ページの左側のナビゲーションペインから [エクスポート] をアクセスできます。



Amazon EC2 への Lightsail スナップショットのエクスポートや、エクスポートしたスナップショットからの EC2 インスタンスの作成の詳細については、以下のガイドを参照してください。

- スナップショットを Amazon EC2 にエクスポートする
- エクスポートしたスナップショットから Amazon EC2 インスタンスを作成する

エクスポートした Lightsail スナップショットから Amazon EC2 インスタンスを作成する

Lightsail インスタンスのスナップショットがエクスポートされ、(AMI および EBS スナップショットとして) Amazon EC2 で利用可能になると、Amazon Lightsail コンソールの [Amazon EC2 インスタンスの作成] ページ (「Amazon EC2 へのアップグレードウィザード」とも呼ばれます) を使用して、スナップショットから Amazon EC2 インスタンスを作成することができます。このウィザードでは、要件に一致する EC2 インスタンスタイプの選択、セキュリティグループのポートの設定、起動スクリプトの追加など、EC2 インスタンスの設定を行うことができます。Lightsail コンソールのウィザードにより、新しい EC2 インスタンスおよび関連リソースの作成プロセスが簡素化されます。

> **Note**
>
> エクスポートしたブロックストレージディスクのスナップショットから Amazon Elastic Block Store (Amazon EBS) ボリュームを作成する場合は、「エクスポートされたディスクのスナップショットから Amazon EBS ボリュームを作成する」を参照してください。

Lightsail API、 AWS CLI、または SDKs を使用して新しい EC2 インスタンスを作成することもできます。詳細については、 Lightsail API ドキュメントの CreateCloudFormationStack オペレーション、または AWS CLI ドキュメントの create-cloud-formation-stack コマンドを参照してください。または、Amazon EC2 に慣れている場合は、EC2 コンソール、Amazon EC2 API AWS CLI、または SDKsを使用できます。詳細については、Amazon EC2 ドキュメントの「インスタンス起動ウィザードを使用したインスタンスの起動」または「スナップショットからの Amazon EBS ボリュームの復元」を参照してください。

> **Important**
>
> このガイドの手順を実行する前に、Lightsail のエクスポートプロセスを再確認することをお勧めします。詳細については、「スナップショットを Amazon EC2 にエクスポートする」を参照してください。

目次

- CloudFormation の スタック Lightsail
- 前提条件

- Lightsail コンソールの「Amazon EC2 インスタンスの作成」ページにアクセスする
- Amazon EC2 インスタンスを作成する
- 新しい Amazon EC2 インスタンスのステータスを追跡する

CloudFormation の スタック Lightsail

Lightsail は、 CloudFormation スタックを使用して EC2 インスタンスとその関連リソースを作成します。Lightsail の CloudFormation スタックの詳細については、「Lightsail のCloudFormation スタック」を参照してください。

「Amazon EC2 インスタンスの作成」ページで EC2 インスタンスを作成するユーザーによっては、以下の追加のアクセス許可を IAM で設定する必要があります。

- Amazon アカウントのルートユーザーが EC2 インスタンスを作成する場合は、このガイドの「前提条件」セクションに進みます。ルートユーザーは、Lightsail を使用して EC2 インスタンスを作成するために必要なアクセス許可をすでに持っています。
- IAM ユーザーが EC2 インスタンスを作成する場合、 AWS アカウント管理者はユーザーに次のアクセス許可を追加する必要があります。ユーザーのアクセス権限を変更する方法については、IAM ドキュメント内の「IAM ユーザーのアクセス権限の変更」を参照してください。
  - ユーザーが Lightsail を使用して Amazon EC2 インスタンスを作成するには、以下のアクセス許可が必要です。

    > **Note**
    >
    > 以下のアクセス許可により、CloudFormation スタックを作成できます。ただし、作成が失敗した場合は、ロールバックプロセスで追加のアクセス許可が必要になることがあります。アクセス許可が不足すると、残りのリソースは Amazon EC2 でロールバックされない可能性があります。この場合、 CloudFormation コンソールに移動し、EC2 リソースを手動で削除できます。詳細については、「Lightsail のCloudFormation スタック」を参照してください。

    - ec2:DescribeAvailabilityZones
    - ec2:DescribeSubnets
    - ec2:DescribeRouteTables
    - ec2:DescribeInternetGateways

    - ec2:DescribeVpcs
    - cloudformation:CreateStack
    - cloudformation:ValidateTemplate
    - iam:CreateServiceLinkedRole
    - iam:PutRolePolicy
- ユーザーが EC2 インスタンスのセキュリティグループでポートを設定する場合は、以下のアクセス許可が必要になります。
    - ec2:DescribeSecurityGroups
    - ec2:CreateSecurityGroup
    - ec2:AuthorizeSecurityGroupIngress
- ユーザーが Amazon EC2 で Windows Server インスタンスを作成する場合は、以下のアクセス許可が必要です。
    - ec2:DescribeKeyPairs
    - ec2:ImportKeyPair
- ユーザーが Amazon EC2 インスタンスを初めて作成する場合や、仮想プライベートクラウド (VPC) の設定が完全に失敗した場合は、以下のアクセス許可が必要です。
    - ec2:AssociateRouteTable
    - ec2:AttachInternetGateway
    - ec2:CreateInternetGateway
    - ec2:CreateRoute
    - ec2:CreateRouteTable
    - ec2:CreateSubnet
    - ec2:CreateVpc
    - ec2:ModifySubnetAttribute
    - ec2:ModifyVpcAttribute

前提条件

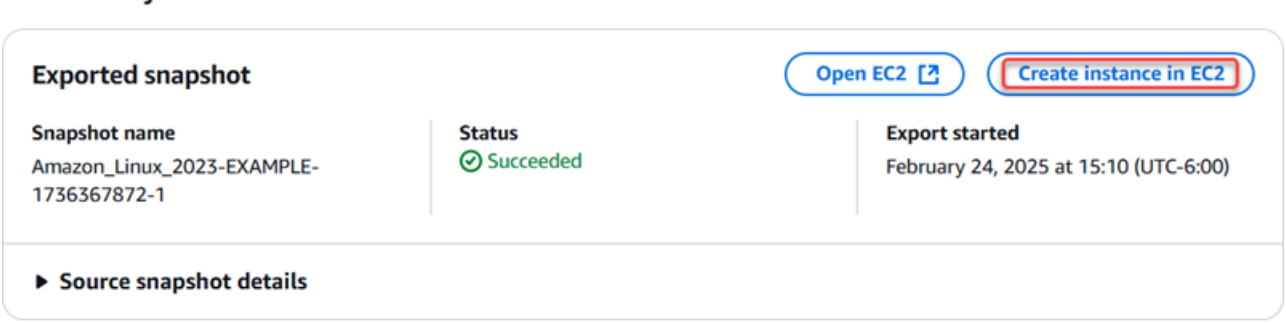
Lightsail インスタンスのスナップショットを Amazon EC2 にエクスポートします。詳細については、「スナップショットを Amazon EC2 にエクスポートする」を参照してください。

Lightsail コンソールの「Amazon EC2 インスタンスの作成」ページにアクセスする

Lightsail コンソールの「Amazon EC2 インスタンスの作成」ページは、インスタンスのスナップショットが EC2 に正常にエクスポートされた後にのみ、タスクモニターからアクセスできます。

Lightsail コンソールの「Amazon EC2 インスタンスの作成」ページにアクセスするには

1. Lightsail コンソール にサインインします。
2. 上部のナビゲーションペインでタスクモニターアイコンを選択します。
3. [タスク履歴] セクションでエクスポート完了済みのインスタンススナップショットを探して、[EC2 でインスタンス作成] を選択します。



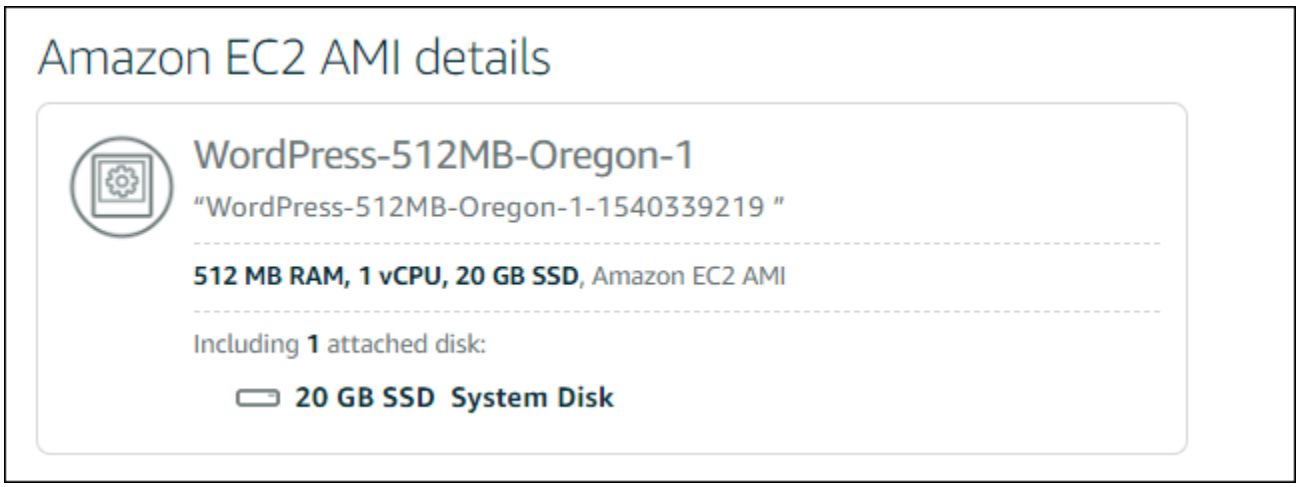
[Amazon EC2 インスタンスの作成] ページが表示されます。このガイドの次の「Amazon EC2 インスタンスの作成」セクションに進み、このページを使用して EC2 インスタンスを設定して作成する方法を確認します。

Amazon EC2 インスタンスを作成する

[Amazon EC2 インスタンスの作成] ページを使用して EC2 インスタンスを作成します。エクスポートされた Lightsail スナップショットから複数の EC2 インスタンスを作成するには、以下の手順を複数回繰り返します。ただし、各インスタンスが作成されるまで待ってから、次のインスタンスを作成します。

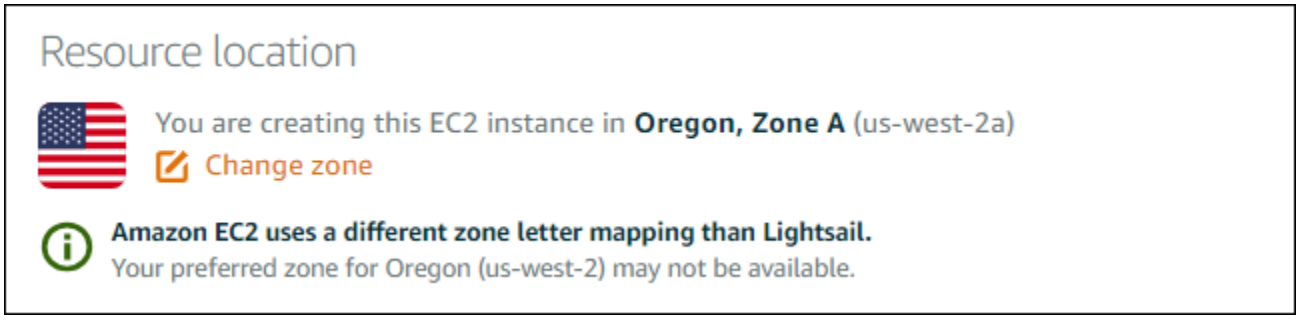
Amazon EC2 インスタンスを作成するには

1. ページの [Amazon EC2 AMI の詳細] セクションで、表示されている Amazon マシンイメージ (AMI) の詳細がソースの Lightsail インスタンスの仕様と一致していることを確認します。

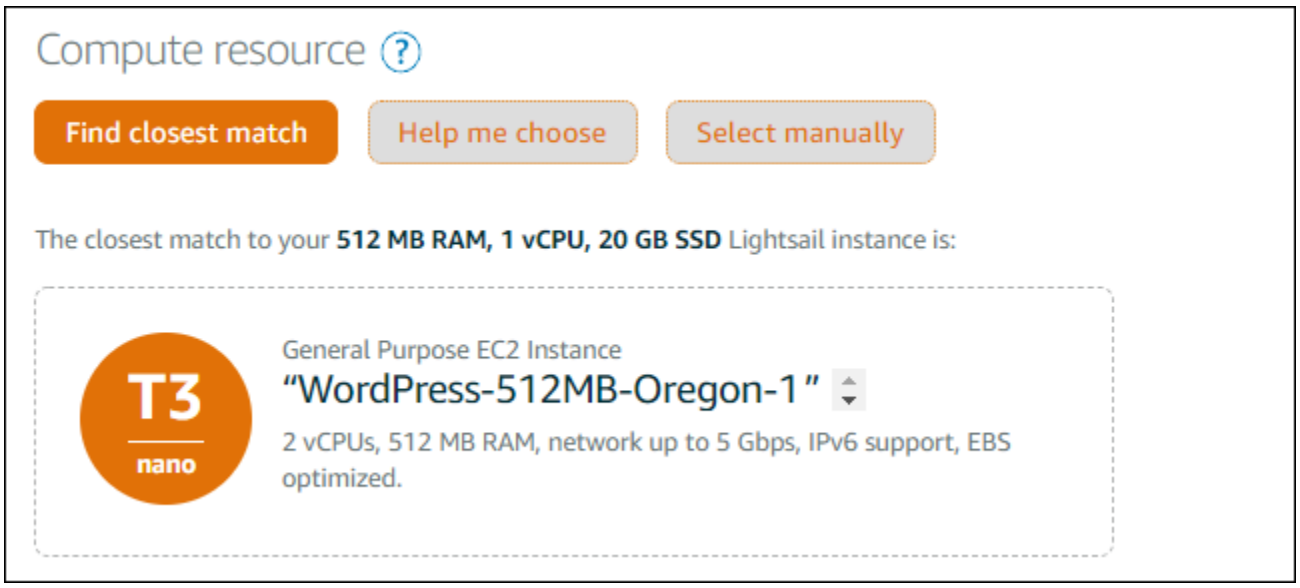


2. ページの [Resource location (リソースの場所)] セクションで、必要に応じてインスタンスのアベイラビリティーゾーンを変更します。Amazon EC2 リソースは、ソースLightsailスナップショット AWS リージョン と同じ に作成されます。

> **Note**
>
> すべてのユーザーがすべてのアベイラビリティーゾーンを使用できるとは限りません。使用できないアベイラビリティーゾーンを選択すると、EC2 インスタンスの作成時にエラーが発生します。



3. ページの [Compute resource (コンピューティングリソース)] セクションで、次のいずれかのオプションを選択します。



a. [最も適切なものを選択する] は、ソースの Lightsail インスタンスの仕様と厳密に一致する Amazon EC2 インスタンスタイプが自動的に選択されます。

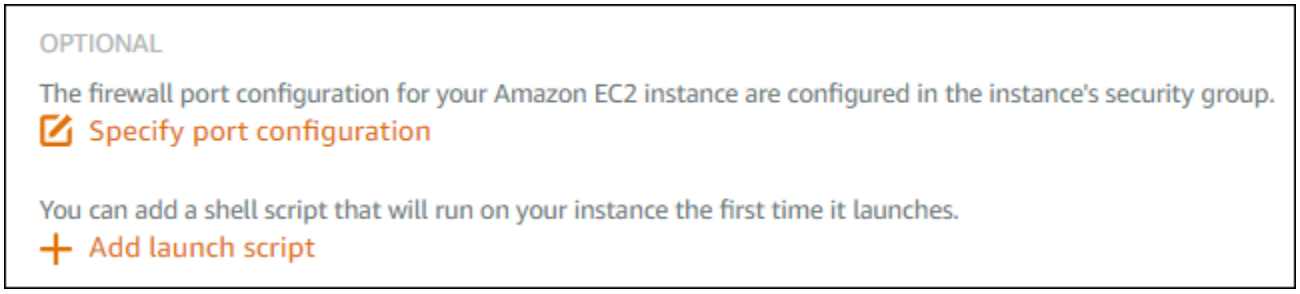
b. [選択のヘルプ] では、新しい Amazon EC2 インスタンスの仕様に関する簡単なアンケートに回答します。コンピューティングを最適化したインスタンスタイプ、メモリを最適化したインスタンスタイプ、または 2 つの間でバランスを取ったインスタンスタイプから選択できます。

c. [手動で選択] では、[Amazon EC2 インスタンスの作成] ページから利用可能なインスタンスタイプが一覧表示されます。

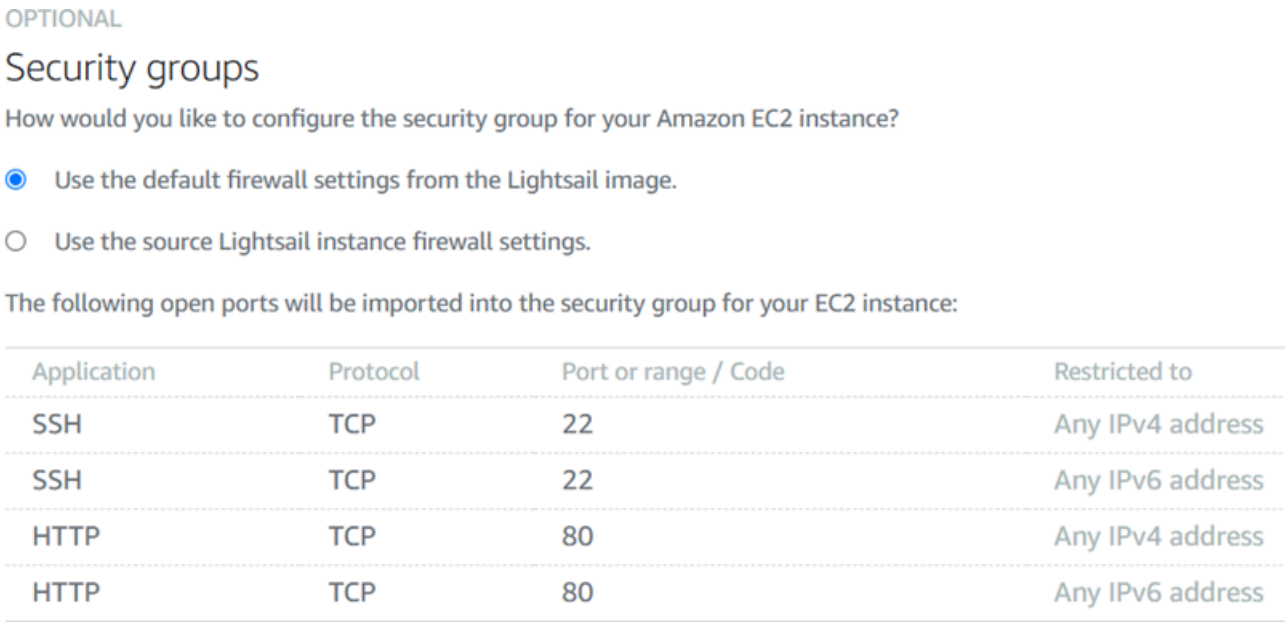
> **Note**
>
> 一部の Lightsail インスタンスは、拡張ネットワーキングに対応していないため、現行世代の EC2 インスタンスタイプ (T3、M5、C5、または R5) と互換性がありません。ソースの Lightsail インスタンスに互換性がない場合は、エクスポートしたスナップショットから EC2 インスタンスを作成するときに、以前の世代のインスタンスタイプ (T2、M4、C4、または R4) から選択する必要があります。これらのインスタンスタイプのオプションは、Lightsail コンソールの [Amazon EC2 インスタンスの作成] ページで表示されます。
>
> ソースの Lightsail インスタンスに互換性がない場合に最新世代の EC2 インスタンスタイプを使用するには、まず以前の世代のインスタンスタイプ (T2、M4、C4、または R4) を使用して新しい EC2 インスタンスを作成し、ネットワーキングドライバーを更新します。次に、インスタンスを目的の現行世代のインスタンスタイプに

更新します。詳細については、「拡張ネットワーキング用に Amazon EC2 インスタンスを更新する」を参照してください。

4. ページの [Optional (オプション)] セクションで以下の操作を行います。



a. [ポート設定の指定を] を選択して Amazon EC2 インスタンスのファイアウォール設定を選択し、次のいずれかのオプションを選択します。



i. Lightsail イメージのデフォルトのファイアウォール設定を使用し、新しい EC2 インスタンスでソースの Lightsail 設計図のデフォルトポートを設定します。Lightsail ブループリントのデフォルトポートの詳細については、「ファイアウォールとポート」を参照してください。

ii. ソースの Lightsail インスタンスのファイアウォール設定を使用し、新しい EC2 インスタンスでソースの Lightsail インスタンスのポートを設定します。このオプションは、ソースの Lightsail インスタンスが実行されている場合にのみ使用できます。

b. ページの [起動スクリプト] セクションで、起動時に EC2 インスタンスを設定するスクリプトを追加する場合は、[起動スクリプトの追加] を選択します。

5. ページの [Connection security (接続セキュリティ)] セクションで、ソースの Lightsail インスタンスに接続した方法を確認します。これにより、適切な SSH キーを取得して、新しい EC2 インスタンスに接続します。ソースの Lightsail インスタンスへの接続方法としては以下が挙げられます。

   a. ソースインスタンスのリージョンのデフォルトLightsailキーペアの使用 — EC2 インスタンス AWS リージョン に接続する の一意のデフォルトLightsailキーをダウンロードして使用します。

      > **Note**
      >
      > Lightsail の Windows Server インスタンスでは、常にデフォルトの Lightsail キーペアが使用されます。

   b. 独自のキーペアを使用する – プライベートキーを見つけて EC2 インスタンスへの接続に使用します。

      > **Note**
      >
      > 個人のプライベートキーは Lightsail に保存されません。したがって、プライベートキーをダウンロードするオプションは提供されていません。プライベートキーが見つからない場合は、EC2 インスタンスに接続できません。

6. ページの [ストレージリソース] セクションで、作成する EBS ボリュームがソースの Lightsail インスタンスのシステムディスクおよびアタッチ済みブロックストレージディスクと一致することを確認します。

7. Lightsail の外部におけるリソースの作成に関する重要な詳細を確認します。
8. Amazon EC2 でインスタンスを作成することに同意する場合は、[EC2 でリソースを作成する] を選択します。

   Lightsail でインスタンスが作成中であることが確認され、 CloudFormation スタックに関する情報が表示されます。Lightsail は CloudFormation スタックを使用して EC2 インスタンスおよび関連リソースを作成します。詳細については、「Lightsail のCloudFormation スタック」を参照してください。

   このガイドの「新しい Amazon EC2 インスタンスのステータスを追跡する」セクションに進んで、新しい EC2 インスタンスのステータスを追跡します。

> **Important**
>
> 新しい EC2 インスタンスが作成されるまで待ってから、同じエクスポートしたスナップショットから別の EC2 インスタンスを作成します。

新しい Amazon EC2 インスタンスのステータスを追跡する

Lightsail コンソールの [エクスポート] を使用して、EC2 インスタンスのステータスを追跡します。詳細については、「Lightsail でスナップショットのエクスポートステータスを追跡する」を参照してください。

作成中の EC2 インスタンスに関する以下の情報が表示されます:

- Source name (ソース名) – ソースの Lightsail スナップショットの名前。
- Started (開始日時) – 作成リクエストが開始された日付と時刻。

タスクモニターには、作成済みの EC2 インスタンスに関する以下の情報が表示されます。

- 作成済み – Amazon EC2 リソースが正常に作成された場合に表示されます。
- 失敗 – EC2 インスタンスの作成中に問題が発生した場合に表示されます。

エクスポートした Lightsail ディスクスナップショットから Amazon Elastic Block Store ボリュームを作成する

Lightsail ブロックストレージディスクのスナップショットをエクスポートして Amazon EC2 で (EBS スナップショットとして) 使用可能にすると、Amazon EC2 コンソールを使用してスナップショットから EBS ボリュームを作成できます。

> **Note**
>
> エクスポートされたインスタンススナップショットから EC2 インスタンスを作成するには、「Lightsail でエクスポートしたスナップショットから Amazon EC2 インスタンスを作成する」を参照してください。

Amazon EC2 API、 AWS CLI、または SDKs を使用して新しい EBS ボリュームを作成することもできます。詳細については、Amazon EC2 ドキュメントの「インスタンス起動ウィザードを使用してインスタンスを起動する」または「スナップショットからの Amazon EBS ボリュームの復元」を参照してください。

> **Important**
>
> このガイドの手順を実行する前に、Lightsail のエクスポートプロセスを再確認することをお勧めします。詳細については、「スナップショットを Amazon EC2 にエクスポートする」を参照してください。

前提条件

Lightsail ブロックストレージディスクのスナップショットを Amazon EC2 にエクスポートします。詳細については、「スナップショットを Amazon EC2 にエクスポートする」を参照してください。

エクスポートした Lightsail ブロックストレージディスクスナップショットから EBS ボリュームを作成する

Amazon EC2 コンソールを使用して、エクスポートした Lightsail ブロックストレージディスクのスナップショットから新しい EBS ボリュームを作成します。

> **Note**
>
> これらの手順は Amazon EC2 のドキュメントにも記載されています。詳細については、Amazon EC2 のドキュメントで「スナップショットからの Amazon EBS ボリュームの復元」を参照してください。

エクスポートされた Lightsail ブロックストレージディスクスナップショットから EBS ボリュームを作成するには

1. Amazon EC2 コンソールにサインインします。
2. ナビゲーションバーから、スナップショットが存在するリージョンを選択します。
3. 左サイドバーのナビゲーションペインで [Elastic Block Store] の [スナップショット] を選択してください。
4. エクスポートした Lightsail ブロックストレージディスクスナップショットを見つけて選択します。

   エクスポートしたディスクスナップショットは、次のスクリーンショットに示すように、EBS スナップショットの説明「A disk snapshot exported from Amazon Lightsail (からエクスポートされたディスクスナップショット)」で識別できます。

5. [アクション]、そして[ボリュームの作成] の順に選択します。
6. [ボリュームタイプ] ドロップダウンメニューからボリュームタイプを選択します。詳細については、Amazon EC2ドキュメントの「Amazon EBS ボリュームタイプ」を参照してください。
7. [Size (GiB)] に、ボリュームのサイズを入力するか、スナップショットのデフォルトサイズが適切であることを実証します。
8. プロビジョンド IOPS SSD ボリュームの場合は、[IOPS] に、ボリュームがサポートすべき 1 秒あたりの入力/出力オペレーション (IOPS) の最大数を入力します。
9. [Availability Zone] では、ボリュームを作成するアベイラビリティーゾーンを選択します。EBS ボリュームは、 EC2 インスタンスと同じアベイラビリティーゾーンに限りアタッチできます。
10. (オプション) [Create additional tags (追加のタグを作成)] を選択してボリュームにタグを追加します。タグごとに、タグキーとタグの値を指定します。
11. [Create Volume (ボリュームの作成)] を選択します。ボリュームが作成されると、そのボリュームは Amazon EC2 コンソールの [Elastic Block Store (EBS) > ボリューム] セクションにリストされます。

Lightsail スナップショットから作成した Linux Amazon EC2 インスタンスに接続する

Amazon Lightsail スナップショットから Amazon Elastic Compute Cloud (Amazon EC2)で Linux または Unix インスタンスを作成すると、ソース Lightsail インスタンスに接続したときと同じように、SSH 経由でインスタンスに接続できます。インスタンスに対して認証するには、ソースインスタンスのデフォルトLightsailキーペア、 AWS リージョンまたは独自のキーペアを使用します。このガイドでは、PuTTY を使用して EC2 の Linux または Unix インスタンスに接続する方法を示します。

> **Note**
>
> Windows Server インスタンスへの接続の詳細については、「Lightsail スナップショットから作成した Amazon EC2 Windows Server インスタンスに接続する」を参照してください。

目次

- インスタンスのキーを取得する
- インスタンスのパブリック DNS アドレスを取得する
- PuTTY をダウンロードしてインストールする
- PuTTYgen を使用してキーを設定する
- インスタンスに接続するように PuTTY を設定する
- 次のステップ

インスタンスのキーを取得する

新しい Amazon EC2 インスタンスに接続するために必要な適切なキーを取得します。必要なキーは、ソースの Lightsail インスタンスに接続した方法によって異なります。ソースの Lightsail インスタンスへの接続方法としては以下が挙げられます。

- ソースインスタンスのリージョンにおけるデフォルト Lightsail キーペアを使用 – Lightsail アカウントページの SSH キータブからデフォルトのプライベートキーをダウンロードします。デフォルトの Lightsail キーの詳細については、「SSH のキーペア」を参照してください。

  > **Note**
  >
  > EC2 インスタンスに接続したら、インスタンスから Lightsail のデフォルトキーを削除して、独自のキーペアに置き換えることをお勧めします。詳細については、「Lightsail スナップショットから作成した Amazon EC2 の Linux または Unix インスタンスを保護する」を参照してください。

- 独自のキーペアを使用する – プライベートキーを探して Amazon EC2 インスタンスの接続に使用します。独自のキーペアを使用した場合、Lightsail にはプライベートキーが保存されません。プライベートキーをなくすと、Amazon EC2 インスタンスに接続できなくなります。

インスタンスのパブリック DNS アドレスを取得する

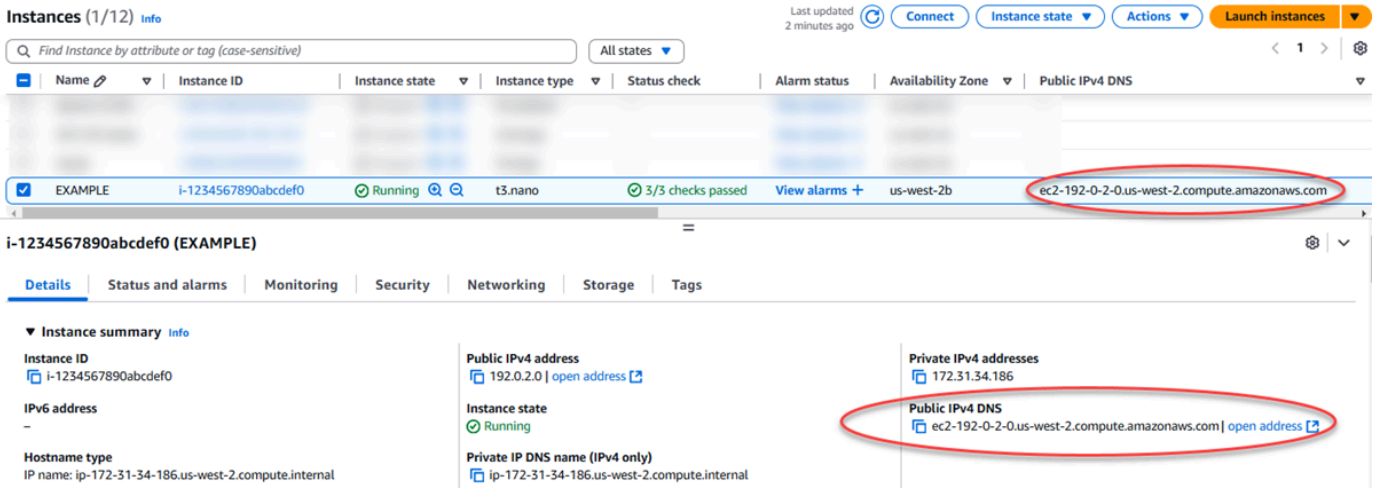
Amazon EC2 インスタンスのパブリック DNS アドレスを取得し、これを SSH クライアント (PuTTY など) の設定時に使用してインスタンスに接続します。

インスタンスのパブリック DNS アドレスを取得するには

1. Amazon EC2 コンソールにサインインします。
2. 左側のナビゲーションペインから、[インスタンス] を選択します。

3. 接続先である実行中の Linux または Unix をインスタンスを選択します。
4. 下部のペインで、インスタンスのパブリック DNS アドレスを見つけます。

   このアドレスを SSH クライアントの設定時に使用してインスタンスに接続します。このガイドの「PuTTY をダウンロードしてインストールする」セクションに進み、PuTTY SSH クライアントをダウンロードしてインストールします。



PuTTY をダウンロードしてインストールする

PuTTY は Windows 用の無料の SSH クライアントです。詳細については、「PuTTY: a free SSH and Telnet client」を参照してください。このウェブサイトでは、暗号化を許可しない諸国での制限についても説明しています。すでに PuTTY がある場合は、このガイドの後述の「PuTTYgen を使用してキーを設定する」セクションに進むことができます。

PuTTY インストーラまたは実行可能ファイルをダウンロードします。 の最新バージョンを使用することをお勧めします。ただし、どのダウンロードを選択するかの詳細については、PuTTY のドキュメントを参照してください。

このガイドの「PuTTYgen を使用してキーを設定する」セクションに進み、PuTTYgen でキーを設定します。

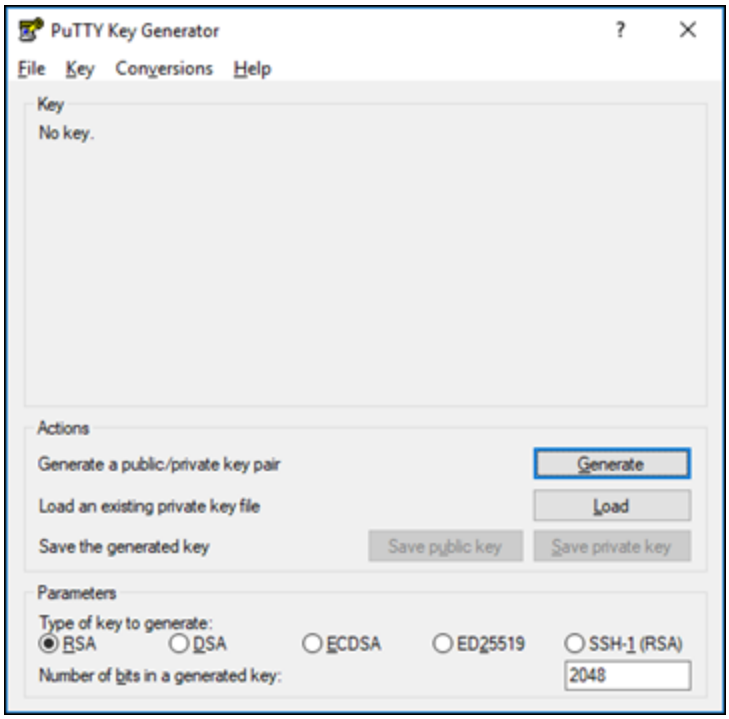
PuTTYgen を使用してキーを設定する

PuTTYgen は、PuTTY で使用するパブリックキーとプライベートキーのペアを生成します。このステップは、PuTTY が受け入れるキーファイルタイプ (.PPK) を使用するために必要です。

PuTTYgen を使用してキーを設定するには

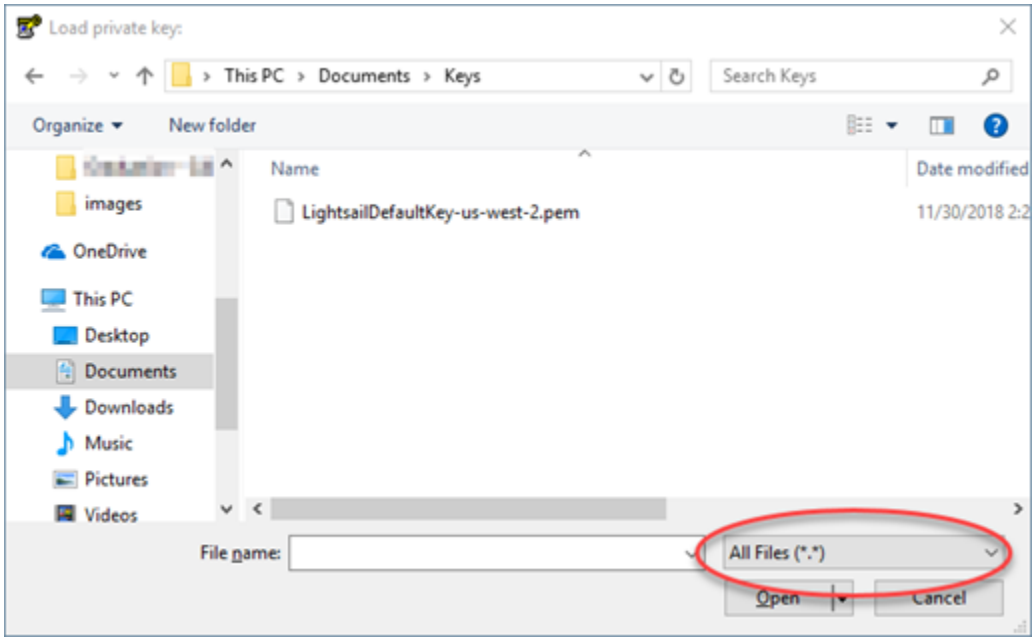
1. PuTTYgen を起動します。

   たとえば、Windows のスタートメニューで、[すべてのプログラム]、[PuTTY]、[PuTTYgen] の順に選択します。

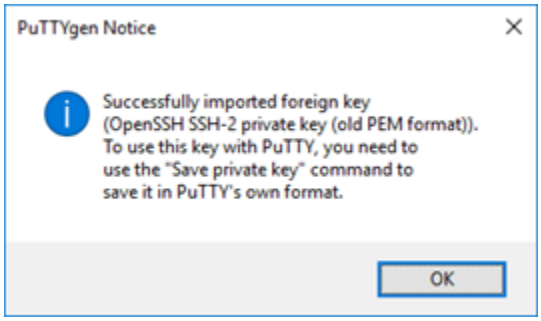


2. [ロード] を選択します。

   デフォルトでは、PuTTYgen には拡張子が .PPK のファイルだけが表示されます。.PEM ファイルを見つけるには、すべてのファイルの種類を表示するオプションを選択します。

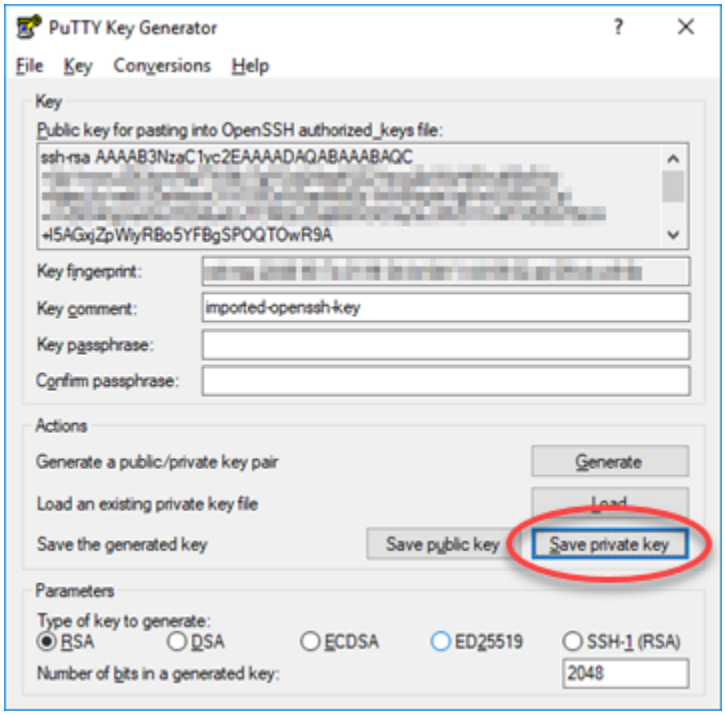


3. このガイドで前にダウンロードした Lightsail のデフォルトキーファイル (.PEM) を選択し、[Open (開く)] を選択します。
4. キーが正常にインポートされたことが PuTTYgen で確認されたら、[OK] を選択します。



5. [Save private key (プライベートキーの保存)] を選択し、パスフレーズ付きで保存しないことを確認します。

   追加のセキュリティ対策としてパスフレーズを作成すると、PuTTY を使用してインスタンスに接続するたびにパスフレーズを入力する必要があります。



6. プライベートキーを保存する名前と場所を指定し、[Save (保存)] を選択します。

   PuTTYgen に、新しいキーファイルが .PPK ファイルタイプとして保存されます。

7. PuTTYgen を閉じます。

   このガイドの「インスタンスに接続するように PuTTY を設定する」セクションに進み、生成した新しい.PPK ファイルを使用して PuTTY を設定し、Amazon EC2 の Linux または Unix インスタンスに接続します。

インスタンスに接続するように PuTTY を設定する

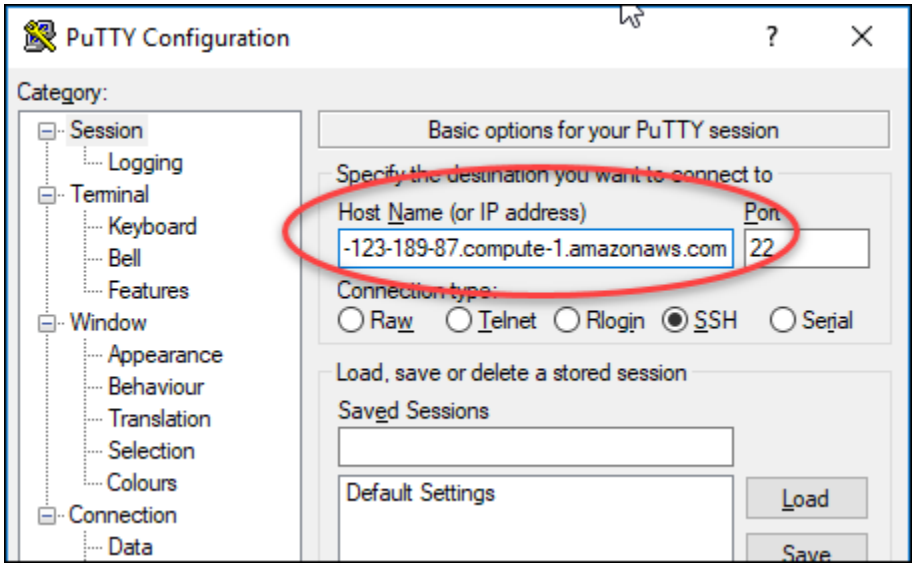
SSH を使用して Linux または Unix インスタンスに接続するためのすべての要件を満たしたので、次に PuTTY を設定します。

Linux または Unix インスタンスに接続するように PuTTY を設定するには

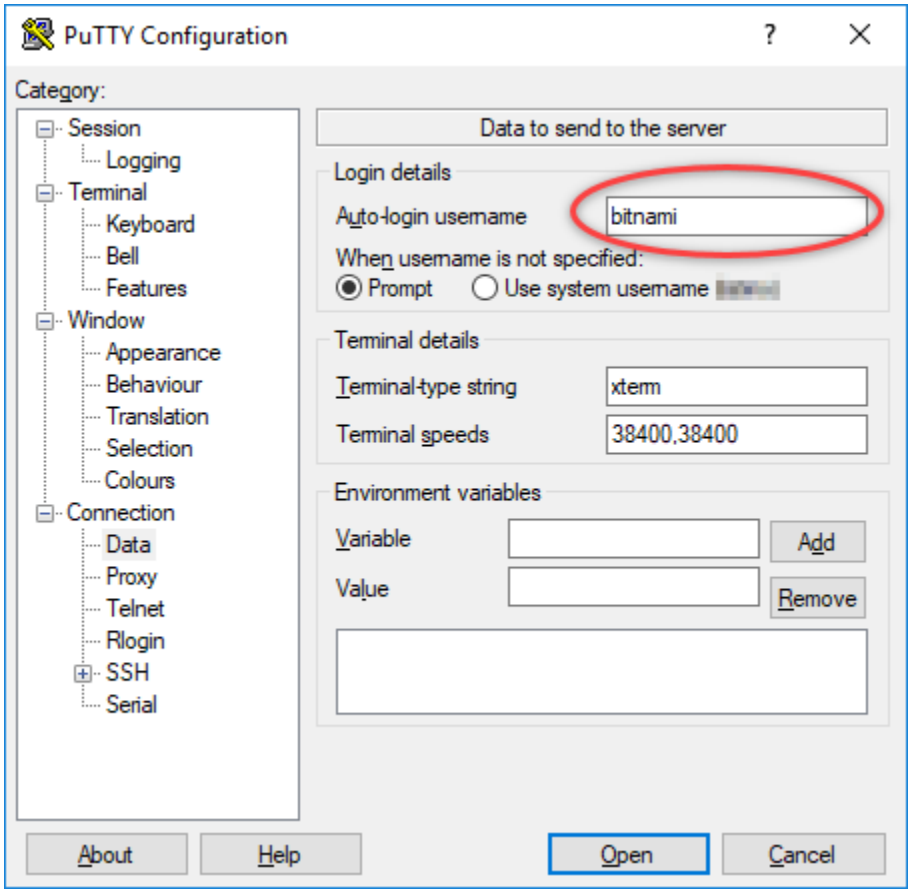
1. PuTTY を開きます。

   たとえば、Windows のスタートメニューで、[すべてのプログラム]、[PuTTY]、[PuTTY] の順に選択します。

2. [ホスト名] テキストボックスに、このガイドで前に Amazon EC2 コンソールから取得した、インスタンスのパブリック DNS アドレスを入力します。



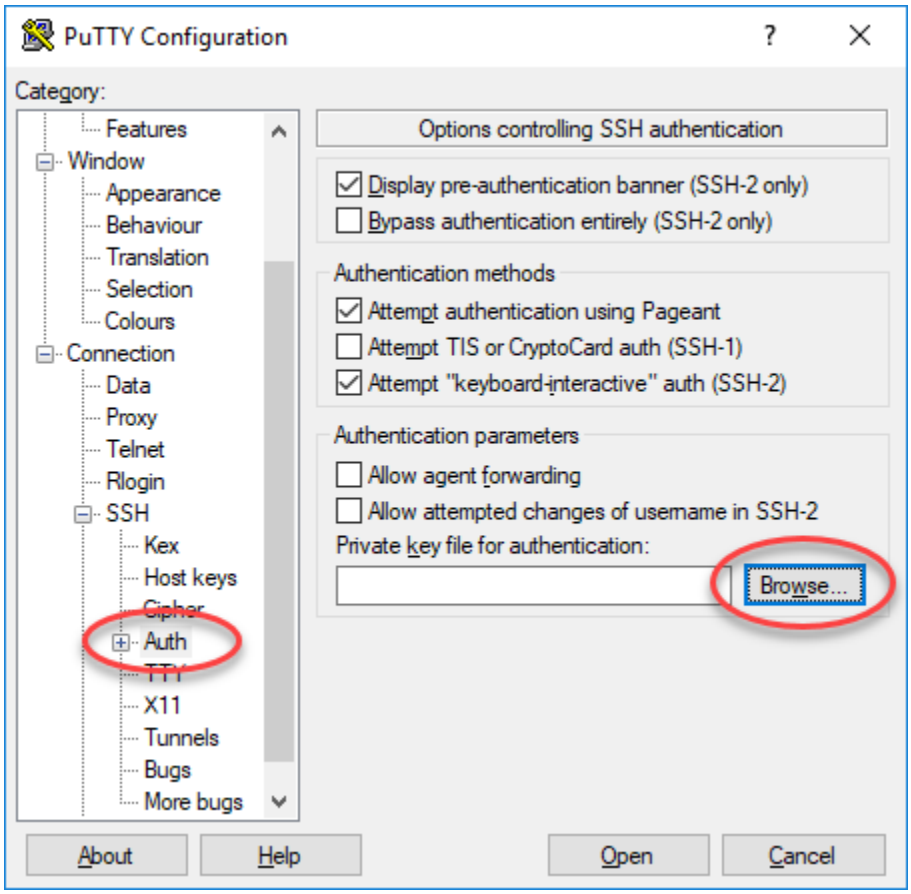
3. 左のナビゲーションペインの [接続] セクションで、[データ] を選択します。
4. [Auto-login username (自動ログインのユーザー名)] テキストボックスに、インスタンスにログインするときに使用するユーザー名を入力します。



ソースの Lightsail インスタンスの設計図に応じて、以下のいずれかのデフォルトユーザー名を入力します。

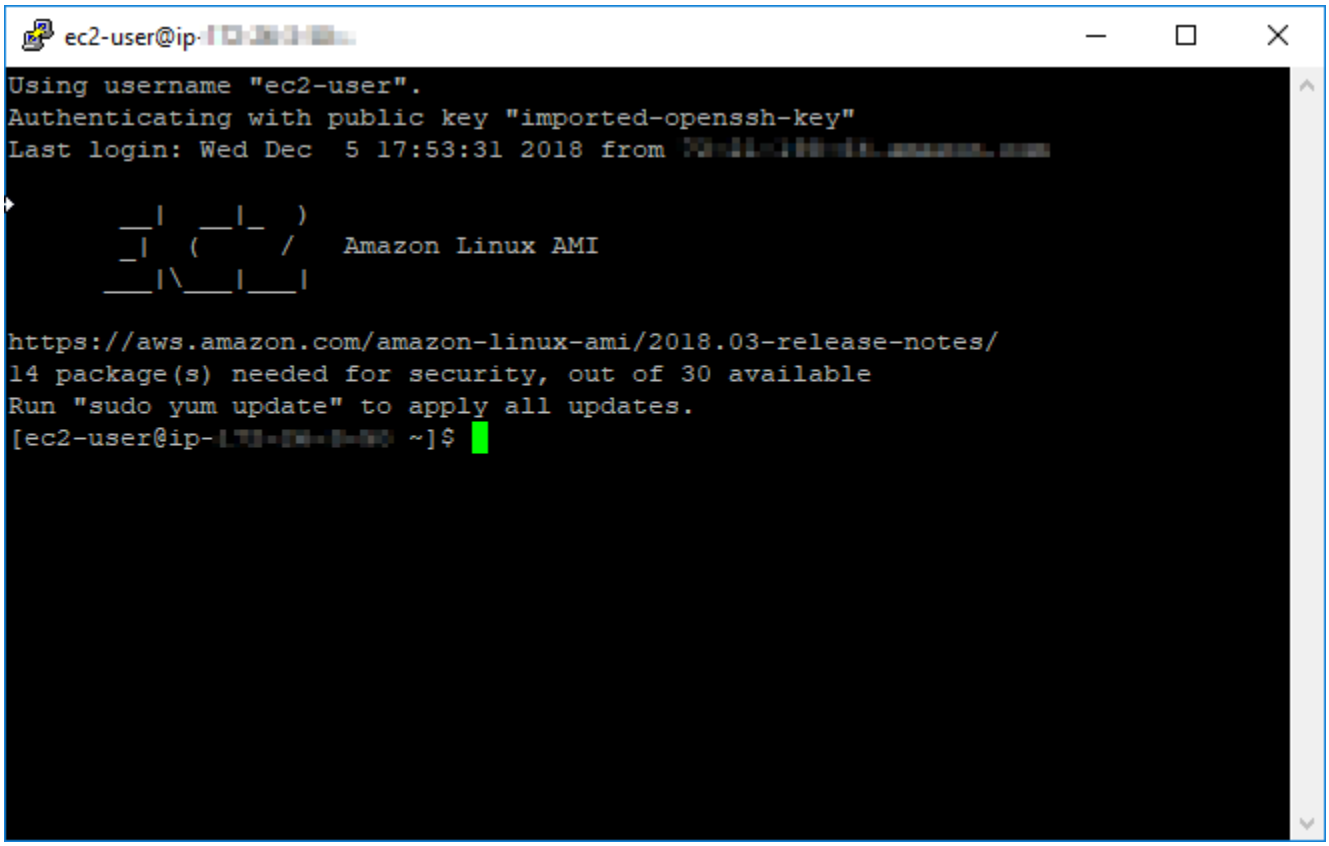
- AlmaLinux、Amazon Linux 2、Amazon Linux 2023、CentOS Stream 9、FreeBSD および openSUSE インスタンス: `ec2-user`
- Debian インスタンス: `admin`
- Ubuntu インスタンス: `ubuntu`
- Bitnami インスタンス: `bitnami`
- Plesk インスタンス: `ubuntu`
- cPanel と WHM インスタンス : `centos`

5. 左のナビゲーションペインの [接続] セクションで、[SSH] を展開して [認証] を選択します。
6. [Browse (参照)] を選択し、このガイドの前のセクションで作成した .PPK ファイルに移動して、[Open (開く)] を選択します。



7. [Open (開く)] を選択してインスタンスに接続し、今後はこの接続を信頼するために [Yes (はい)] を選択します。

インスタンスに正常に接続されると、次のような画面が表示されます。



次の手順

エクスポートされたスナップショットから Amazon EC2 で新しいインスタンスを作成すると、Amazon EC2 の Linux または Unix インスタンスに Lightsail サービスからのキーが含まれています。新しい Amazon EC2 インスタンスのセキュリティを強化するには、これらのキーを削除することをお勧めします。詳細については、「Lightsail スナップショットから作成した Amazon EC2 の Linux または Unix インスタンスを保護する」を参照してください。

Lightsail スナップショットから起動された安全な Amazon EC2 インスタンス

Amazon Lightsail および Amazon Elastic Compute Cloud (Amazon EC2) は公開キー暗号化を使用し、ログイン情報の暗号化と復号を行います。パブリックキー暗号はパブリックキーを使用してデータを暗号化し (パスワードなど)、受信者はプライベートキーを使用してデータを復号します。パブリックキーとプライベートキーは、キーペアと呼ばれます。

Linux または Unix インスタンスを Lightsail から EC2 にエクスポートすると、新しい EC2 インスタンスには Lightsail サービスからのキーが残ります。セキュリティ上のベストプラクティスとして、未使用のキーはインスタンスから削除してください。

Lightsail スナップショットから作成した EC2 の Linux または Unix インスタンスのセキュリティを向上させるには、インスタンスの作成後に以下のアクションを実行することをお勧めします。

- Lightsail のソースインスタンスに接続するために使用した Lightsail のデフォルトキーを削除して置き換えます。独自のキーを使用してインスタンスに接続した場合や、Lightsail コンソールでインスタンスのキーを作成した場合には、Lightsail のデフォルトキーは Amazon EC2 インスタンス内に存在しません。
- Lightsail のシステムキー (`lightsail_instance_ca.pub` キーとも呼ばれます) を削除します。Linux および Unix インスタンスのこのキーにより、Lightsail のブラウザベースの SSH クライアントはインスタンスに接続できます。Lightsail コンソールまたは Lightsail API の [Amazon EC2 インスタンスを作成する] ページを使用して EC2 インスタンスが作成すると、`lightsail_instance_ca.pub` キーは自動的に削除されます。

目次

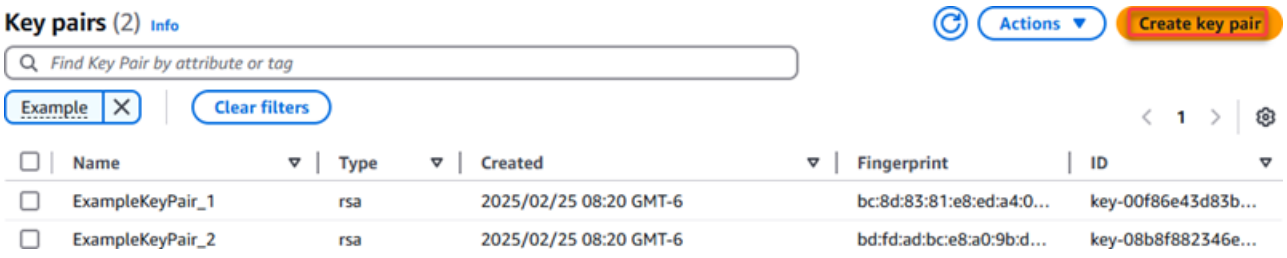
- Amazon EC2 を使用してプライベートキーを作成する
- PuTTYgen を使用してパブリックキーを作成する
- Amazon EC2 の Linux または Unix インスタンスに接続する
- インスタンスにパブリックキーを追加して接続をテストする
- Lightsail のデフォルトキーを削除する
- Lightsail のシステムキーを削除する

Amazon EC2 を使用してプライベートキーを作成する

Amazon EC2 コンソールを使用して、Lightsail のデフォルトキーを置き換えるために使用する新しいキーペアを作成します。

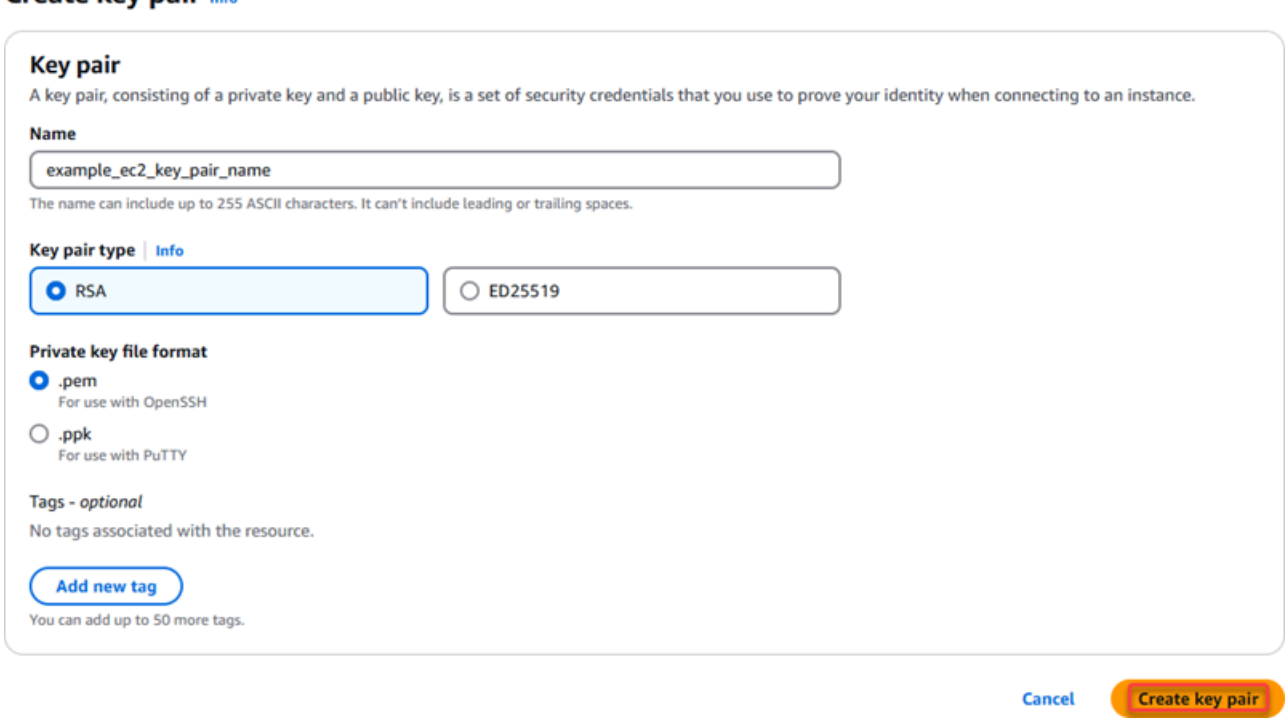
Amazon EC2 を使用してプライベート キーを作成するには

1. Amazon EC2 コンソールにサインインします。
2. 左のナビゲーションペインから、[キーペア] を選択します。
3. [キーペアの作成] を選択してください。



4. [キーペア名] テキストボックスにキー名を入力し、[キーペア作成] を選択します。Amazon EC2 でキーペア作成の詳細については、「Amazon Elastic Compute Cloud ユーザーガイド」の「Amazon EC2 インスタンスのキーペアの作成」を参照してください。

   新しいプライベートキーが自動的にダウンロードされます。プライベートキーの保存先を書き留めておきます。次の「PuTTYgen を使用してパブリックキーを作成する」セクションでパブリックキーを作成するときに必要になります。



PuTTYgen を使用してパブリックキーを作成する

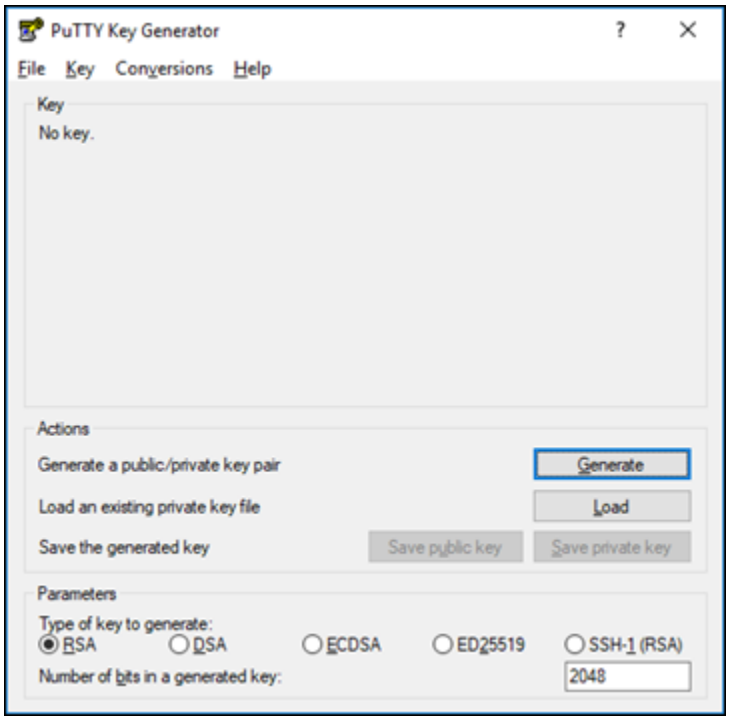
PuTTYgen は PuTTY に含まれているツールです。PuTTYgen では、このガイドで後ほどインスタンスに追加するパブリックキーテキストを生成します。

> **Note**
>
> Linux または Unix インスタンスに接続するように PuTTY を設定する方法の詳細については、「Lightsail スナップショットから作成した Amazon EC2 の Linux または Unix インスタンスに接続する」を参照してください。

PuTTYgen を使用してパブリックキーを作成するには

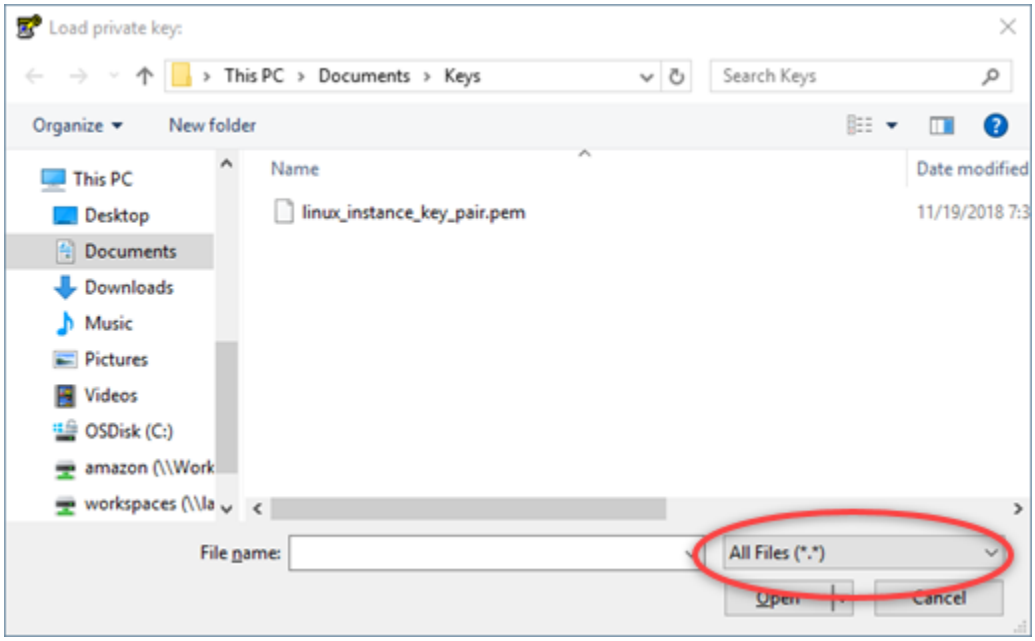
1. PuTTYgen を起動します。

   たとえば、Windows のスタートメニューで、[すべてのプログラム]、[PuTTY]、[PuTTYgen] の順に選択します。



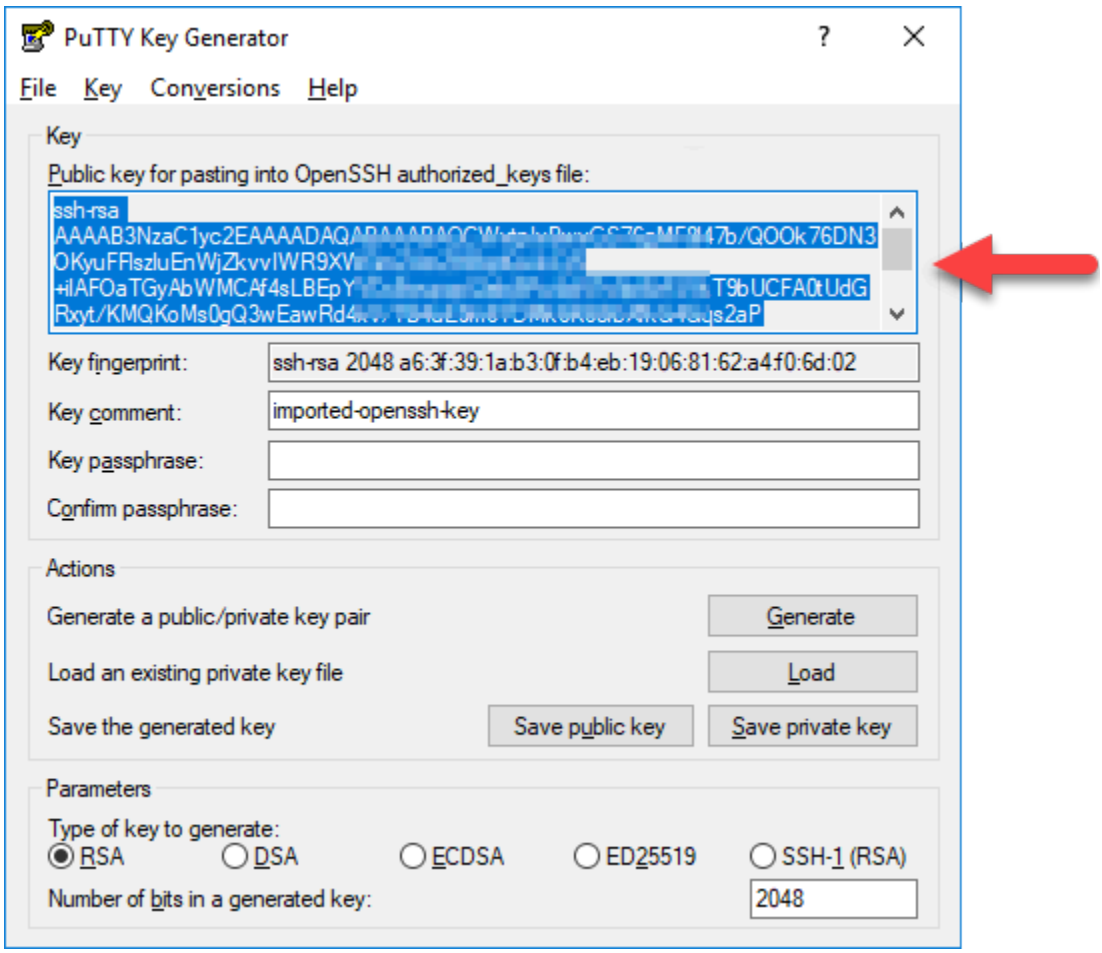
2. [ロード] を選択します。

   デフォルトでは、PuTTYgen には拡張子が .PPK のファイルだけが表示されます。.PEM ファイルを見つけるには、すべてのファイルの種類を表示するオプションを選択します。



3. このガイドで先ほど作成したプライベートキーの場所に移動します。プライベートキーを選択し、[開く] を選択します。
4. キーが正常にインポートされたことが PuTTYgen で確認されたら、[OK] を選択します。
5. [パブリックキー] テキストボックスの内容を強調表示し、Ctrl+C (Windows) または Cmd+C (macOS) を押してクリップボードにコピーします。

   メモ帳や TextEdit などのテキスト工ディタを開き、Ctrl+V (Windows) または Cmd+V (macOS) を押してパブリックキーテキストを貼り付けます。パブリックキーテキストのファイルを保存します。このガイドで後ほど必要になります。



6. 「Amazon EC2 の Linux または Unix インスタンスに接続する」セクションに進み、EC2 インスタンスに接続してパブリックキーを追加します。

Amazon EC2 の Linux または Unix インスタンスに接続する

SSH を使用して Amazon EC2 で Linux または Unix インスタンスに接続し、Lightsail のデフォルトキーとシステムキーを削除します。詳細については、「Amazon Lightsail スナップショットから作成した Amazon EC2 の Linux または Unix インスタンスに接続する」を参照してください。

Amazon EC2 でインスタンスに接続したら、このガイドの「公開キーをインスタンスに追加して接続テストをする」のセクションに進んでください。

インスタンスにパブリックキーを追加して接続をテストする

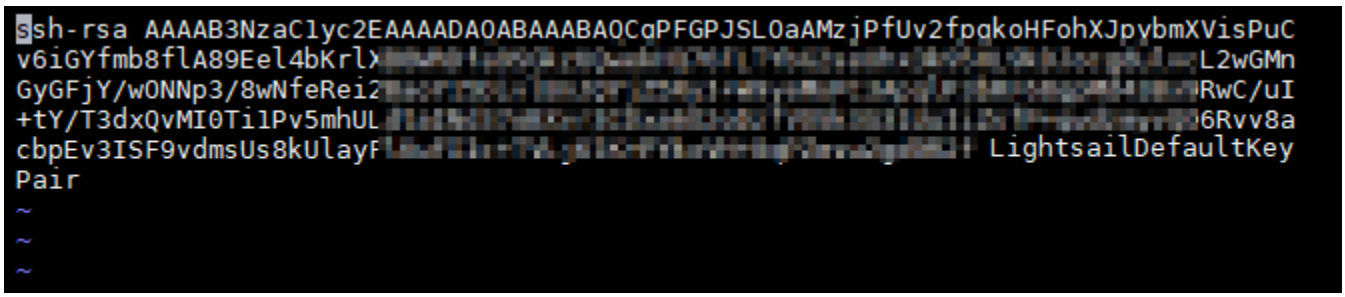
公開キーの内容は、Linux および Unix インスタンスの `~/.ssh/authorized_keys` ファイルに保存されています。このファイルを編集し、Lightsail のデフォルトキーを Amazon EC2 の Linux または Unix インスタンスから削除して置き換えます。

インスタンスにパブリックキーを追加して接続をテストするには

1. インスタンスへの SSH 接続を確立したら、次のコマンドを入力し、Vim テキストエディタを使用して authorized_keys ファイルを編集します。

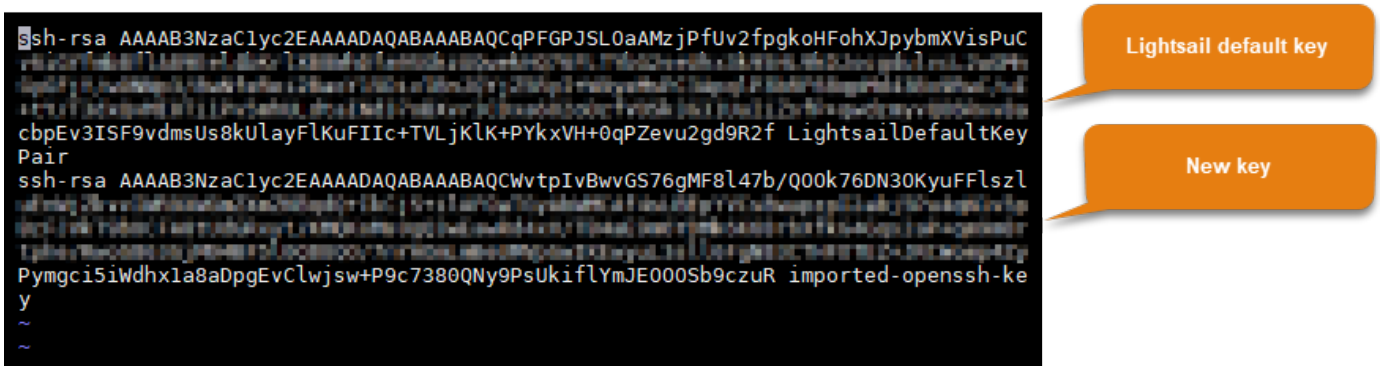
```
sudo vim ~/.ssh/authorized_keys
```

> **Note**
>
> 以下のステップでは、デモの目的で Vim を使用します。ただし、以下のステップでは任意のテキストエディタを使用できます。



2. I キーを押して Vim エディタを挿入モードにします。
3. Lightsail のデフォルトキーの後に追加の行を入力します。
4. このガイドで先ほど保存したパブリックキーテキストをコピーして貼り付けます。

   結果は次のようになります。



5. ESC キーを押して :wq! を入力すると、編集が保存され Vim が終了します。
6. 次のコマンドを入力して Open SSH サーバーを再起動します。

```
sudo /etc/init.d/sshd restart
```

次のような結果が表示されます。

```
[ec2-user@ip-172-26-11-173 ~]$ sudo /etc/init.d/sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
[ec2-user@ip-172-26-11-173 ~]$
```

新しいパブリックキーがインスタンスに追加されました。新しいキーペアをテストするには、インスタンスから切断します。Lightsail デフォルトキーの代わりに新しいプライベートキーを使用するように PuTTY を設定します。新しいキーペアを使用してインスタンスに正常に接続できる場合は、「Lightsail のデフォルトキーを削除する」セクションに進み、Lightsail のデフォルトキーを削除します。

Lightsail のデフォルトキーを削除する

インスタンスに新しいパブリックキーを追加し、新しいキーペアを使用してインスタンスに正常に接続したら、Lightsail のデフォルトキーを削除します。

Lightsail のデフォルトキーを削除するには

1. インスタンスへの SSH 接続を確立したら、次のコマンドを入力し、Vim テキストエディタを使用して `authorized_keys file` を編集します。

```
sudo vim ~/.ssh/authorized_keys
```

2. I キーを押して Vim エディタを挿入モードにします。
3. `LightsailDefaultKeyPair` で終わる行を削除します。これが Lightsail のデフォルトキーです。

4. ESC キーを押して :wq! を入力すると、編集が保存され Vim が終了します。
5. 次のコマンドを入力して Open SSH サーバーを再起動します。

```
sudo /etc/init.d/sshd restart
```

次のような結果が表示されます。

```
[ec2-user@ip-172-26-11-173 ~]$ sudo /etc/init.d/sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
[ec2-user@ip-172-26-11-173 ~]$
```

Lightsail のデフォルトキーがインスタンスから削除されました。これで Lightsail のデフォルトキーを使用する接続はインスタンスで拒否されます。「Lightsail のシステムキーを削除する」セクションに進み、Lightsail のシステムキーを削除します。

Lightsail のシステムキーを削除する

Linux および Unix インスタンスにある Lightsail のシステムキー (lightsail_instance_ca.pub キーとも呼ばれます) により、Lightsail のブラウザベースの SSH クライアントはインスタンスに接続できます。以下のステップを実行して、Amazon EC2 の Linux または Unix インスタンスから lightsail_instance_ca.pub キーを削除し、/etc/ssh/sshd_config ファイルを編集します。/etc/ssh/sshd_config ファイルは、インスタンスへの SSH 接続のパラメータを定義します。

Lightsail のシステムキーを削除するには

1. インスタンスに接続されている SSH のターミナルウィンドウで、次のコマンドを入力して lightsail_instance_ca.pub キーを削除します。

```
sudo rm -r /etc/ssh/lightsail_instance_ca.pub
```

2. 次のコマンドを入力し、Vim テキストエディタを使用して sshd_config ファイルを編集します。

```
sudo vim /etc/ssh/sshd_config
```

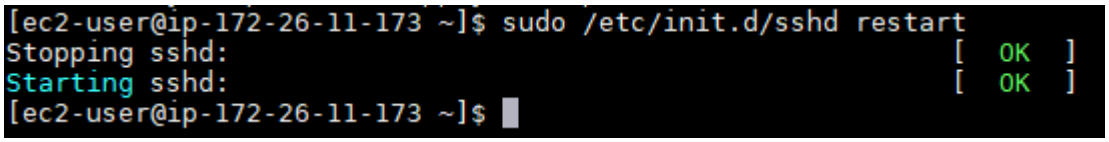
3. I キーを押して Vim エディタを挿入モードにします。
4. 次のテキストをファイルから削除します (ある場合)。

```
TrustedUserCAKeys /etc/ssh/lightsail_instance_ca.pub
```

5. ESC キーを押して :wq! を入力すると、編集が保存され Vim が終了します。
6. 次のコマンドを入力して Open SSH サーバーを再起動します。

```
sudo /etc/init.d/sshd restart
```

次のような結果が表示されます。



lightsail_instance_ca.pub キーがインスタンスから削除されました。関連する sshd_config ファイルが更新されて、このキーが除外されます。

Lightsail スナップショットから作成された Windows Server Amazon EC2 インスタンスに接続する

Amazon Elastic Compute Cloud (Amazon EC2) で新しい Windows Server インスタンスを作成すると、Remote Desktop Protocol (RDP) を使用して、このインスタンスに接続できます。接続方法はソースの Amazon Lightsail インスタンスに接続したときと同様です。ソースインスタンスのデフォルトのLightsailキーペアを使用して EC2 インスタンスに接続します AWS リージョン。このガイドでは、Microsoft リモートデスクトップ接続を使用して Windows Server インスタンスに接続する方法について説明します。

> **Note**
>
> Linux または Unix インスタンスへの接続の詳細については、「Lightsail スナップショットから作成した Amazon EC2 の Linux または Unix インスタンスに接続する」を参照してください。

目次

- インスタンスのキーを取得する
- インスタンスのパブリック DNS アドレスを取得する

- Windows Server インスタンスのパスワードを取得する
- Windows Server インスタンスに接続するようにリモートデスクトップ接続を設定する
- 次のステップ

インスタンスのキーを取得する

Amazon EC2 の Windows Server インスタンスは、ソースインスタンスのリージョンの Lightsail のデフォルトキーペアを使用して、デフォルトの管理者パスワードを取得します。

Lightsail アカウントページの SSH キータブからデフォルトのプライベートキーをダウンロードします。デフォルトの Lightsail SSH キーの詳細については、「SSH のキーペア」を参照してください。

> **Note**
>
> EC2 インスタンスに接続したら、Amazon EC2 で Windows Server インスタンスの管理者パスワードを変更することをお勧めします。これにより、Lightsail のデフォルトキーペアと Amazon EC2 の Windows Server インスタンスの関連付けが解除されます。詳細については、「Lightsail スナップショットから作成した Amazon EC2 の Windows Server インスタンスを保護する」を参照してください。

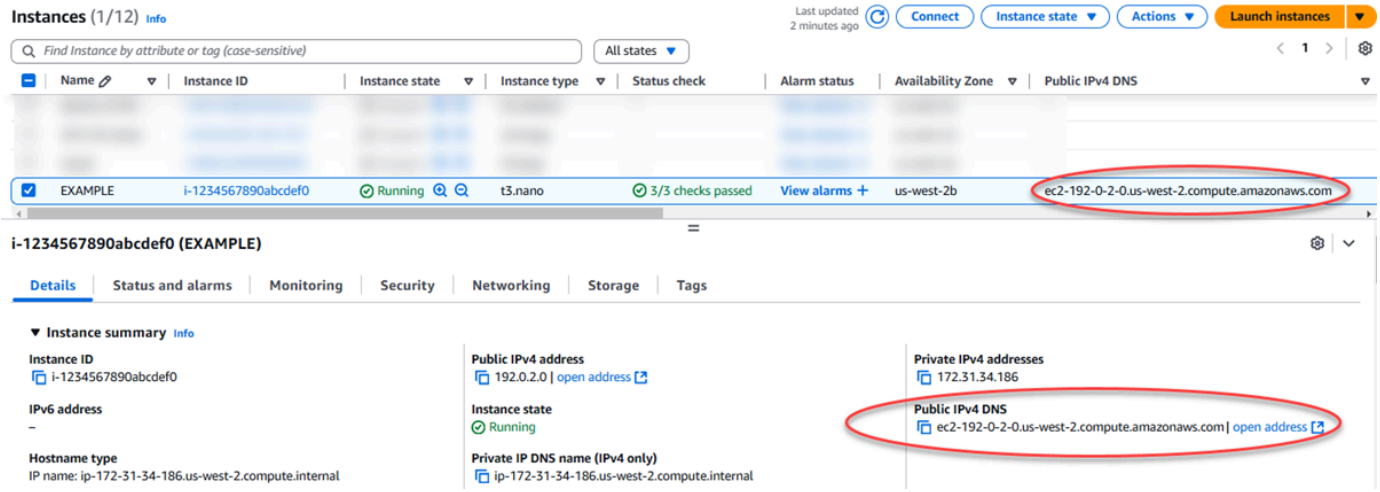
インスタンスのパブリック DNS アドレスを取得する

Amazon EC2 インスタンスのパブリック DNS アドレスを取得し、これを RDP クライアント (Microsoft リモートデスクトップ接続など) の設定時に使用してインスタンスに接続します。

インスタンスのパブリック DNS アドレスを取得するには

1. Amazon EC2 コンソールにサインインします。
2. 左側のナビゲーションペインから、[インスタンス] を選択します。
3. 接続先である実行中の Windows Server インスタンスを選択します。
4. 下部のペインで、インスタンスのパブリック DNS アドレスを見つけます。

   このアドレスを RDP クライアントの設定時に使用してインスタンスに接続します。「Windows Server インスタンスのパスワードを取得する」セクションに進み、Amazon EC2 で Windows Server インスタンスのデフォルトの管理者パスワードを取得する方法を確認します。

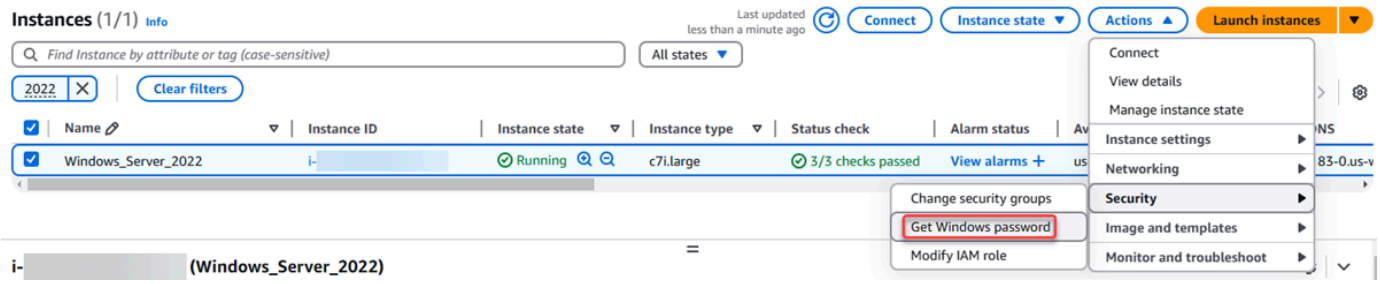


Windows Server インスタンスのパスワードを取得する

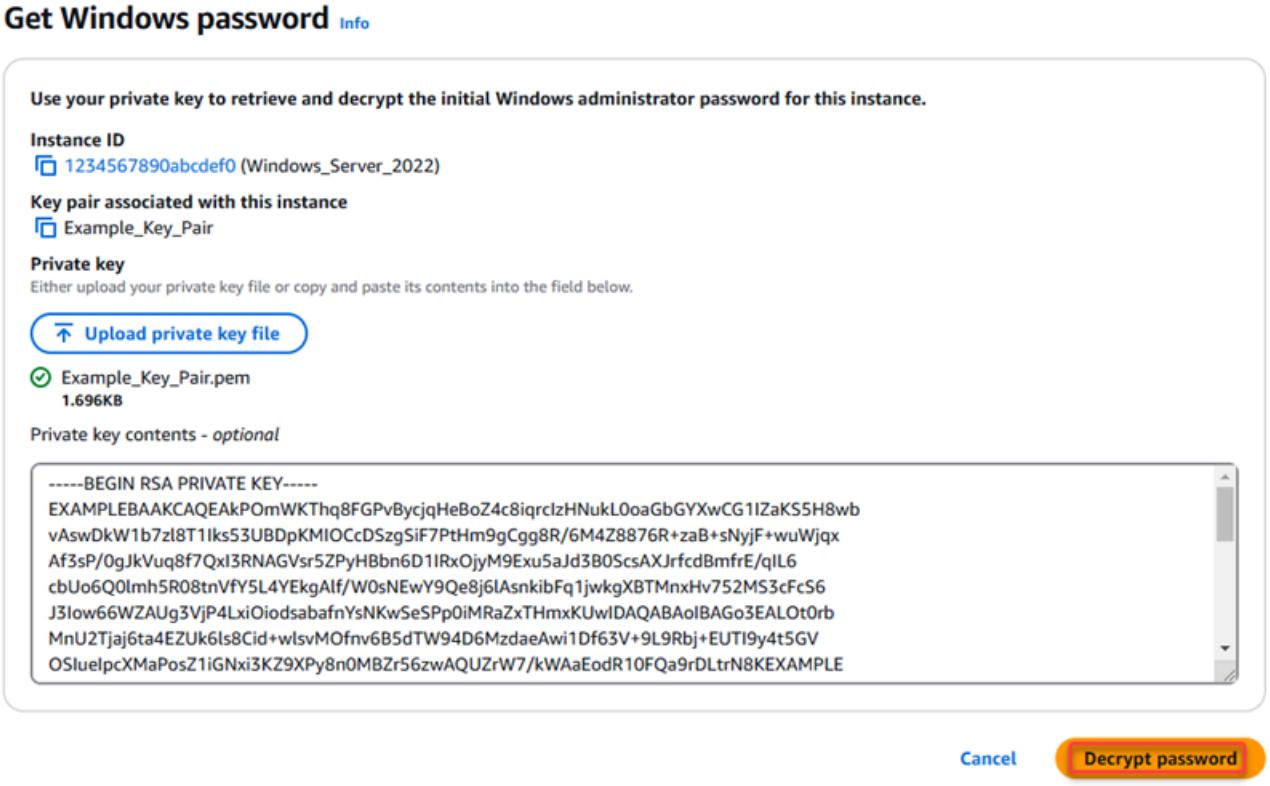
Amazon EC2 コンソールから Windows Server インスタンスのパスワードを取得します。このパスワードは、RDP を通じて Windows Server インスタンスに接続するときに、このインスタンスにサインインするために使用します。

Windows Server インスタンスのパスワードを取得するには

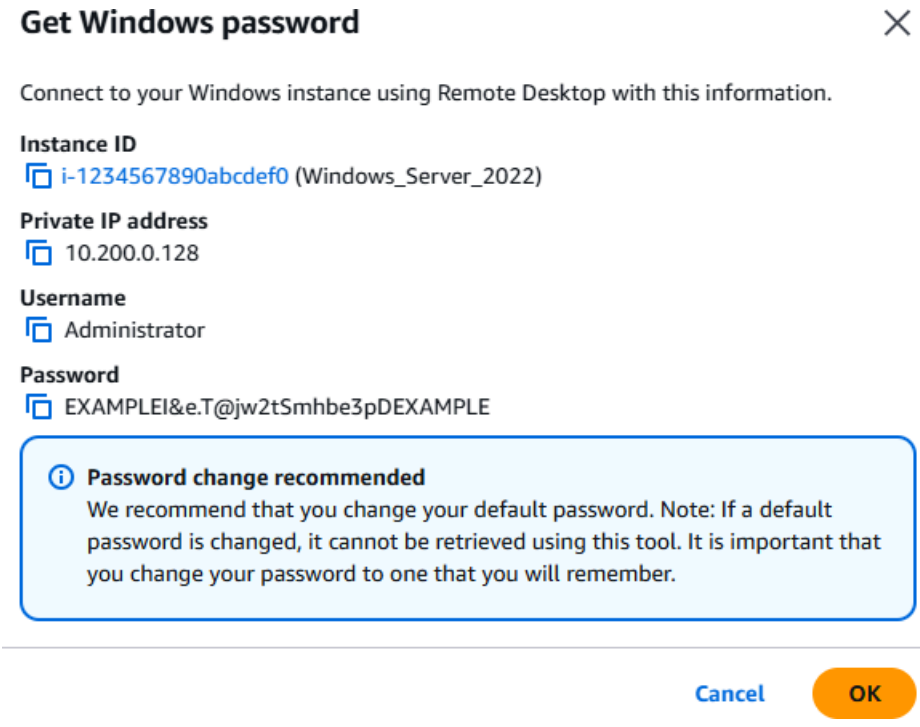
1. Amazon EC2 コンソールにサインインします。
2. 左のナビゲーションペインの [インスタンス] を選択します。
3. 接続先の Windows Server インスタンスを選択します。
4. [アクション] には、[セキュリティ] と [Windows パスワードの取得] を選択します。



5. プロンプトに応じて [参照] を選択し、このガイドで前に Lightsail からダウンロードしたデフォルトのプライベートキーファイルを開きます。
6. [Decrypt Password] (パスワードを復号化) を選択します。



パスワード、ユーザー名、プライベート IP アドレスが表示されます。パスワードをクリップボードにコピーします。このパスワードは、次の「Windows Server インスタンスに接続するようにリモートデスクトップ接続を設定する」セクションで使用します。パスワードを強調表示し、Ctrl+C (Windows) または Cmd+C (macOS) を押します。



このガイドの「Windows Server インスタンスに接続するようにリモートデスクトップ接続を設定する」セクションに進み、Amazon EC2 で Windows Server インスタンスに接続するようにリモートデスクトップ接続を設定する方法を確認します。

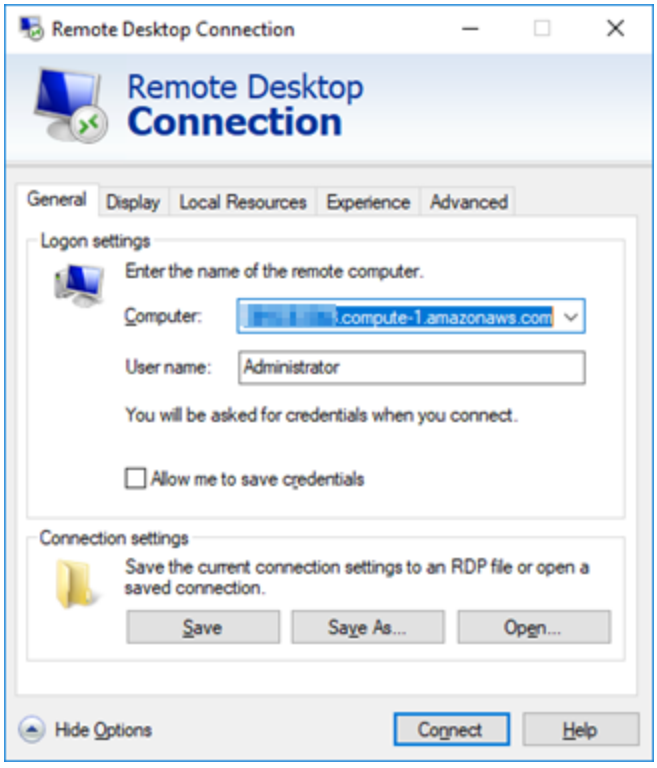
Windows Server インスタンスに接続するようにリモートデスクトップ接続を設定する

リモートデスクトップ接続は、ほとんどの Windows オペレーティングシステムにプリインストールされている RDP クライアントです。これを使用して、Amazon EC2 の Windows Server インスタンスにグラフィカルに接続します。

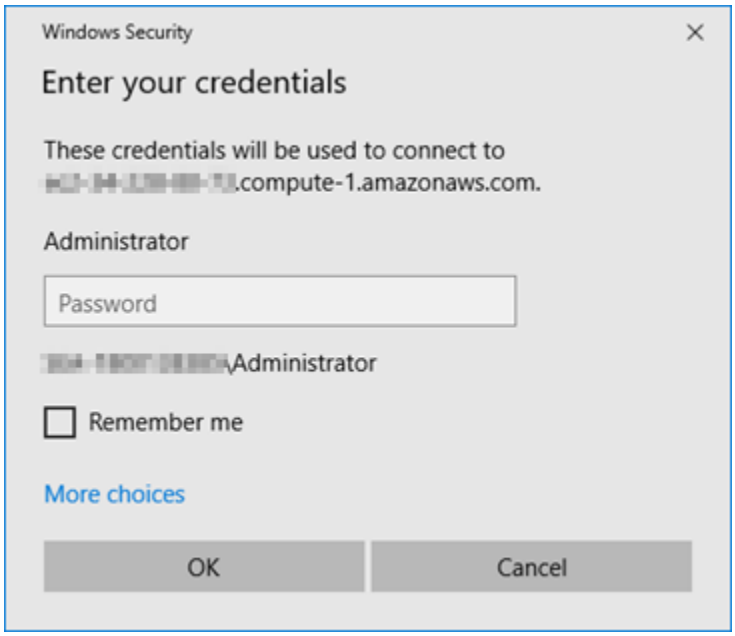
Windows Server インスタンスに接続するようにリモートデスクトップ接続を設定するには

1. リモートデスクトップ接続を開きます。

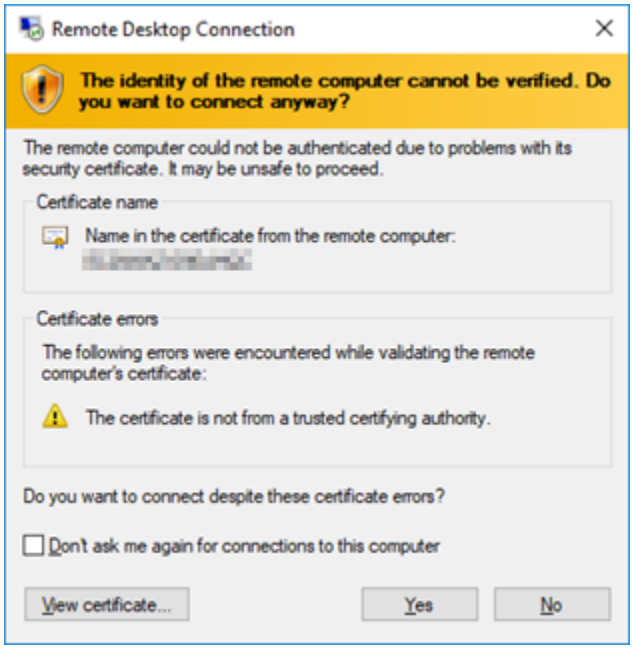
   たとえば、Windows のスタート メニューを選択し、[リモートデスクトップ接続] を見つけます。
2. [コンピューター] テキストボックスに、このガイドで前に取得した Amazon EC2 の Windows Server インスタンスのパブリック DNS アドレスを入力します。
3. [オプションの表示] を選択して追加のオプションを表示します。
4. `Administrator` をユーザー名テキストボックスに入力します。



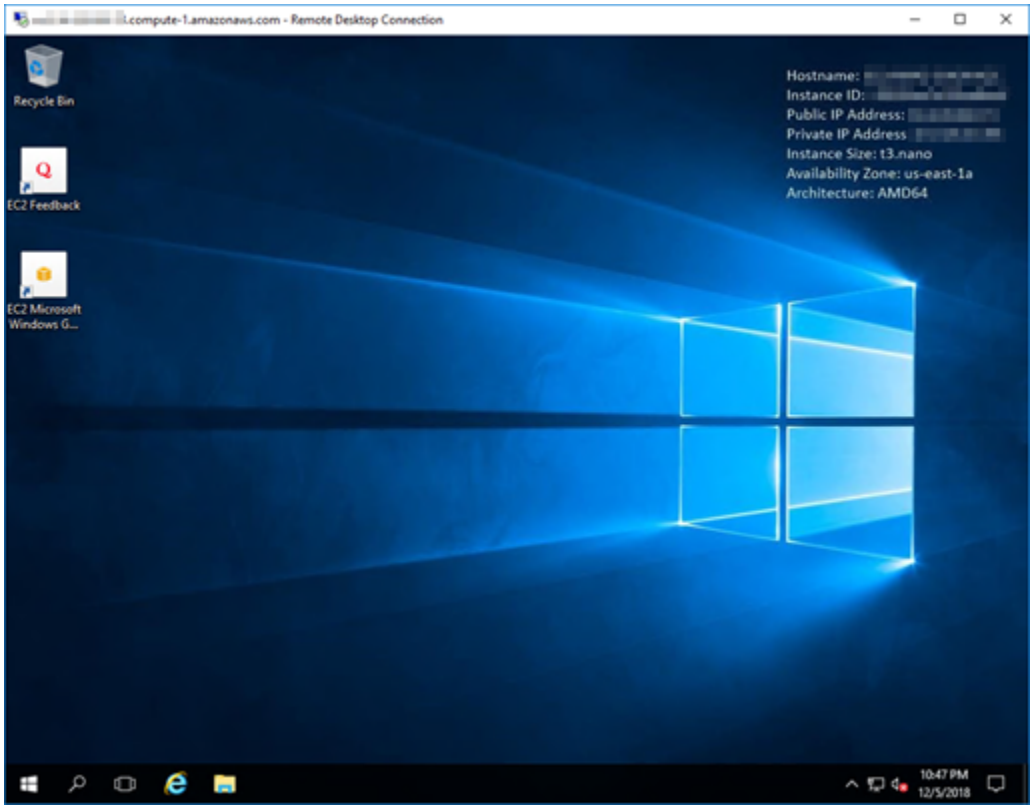
5. [接続] を選択して Windows Server インスタンスに接続します。
6. Windows セキュリティのプロンプトで、[パスワード] テキストボックスに Windows Server インスタンスのパスワードを入力し、[OK] を選択します。



7. リモートデスクトップ接続のプロンプトで、[はい] を選択して接続します。



インスタンスに正常に接続されると、次のような画面が表示されます。



次の手順

Amazon EC2 で Windows Server インスタンスの管理者パスワードを変更することをお勧めします。これにより、Lightsail のデフォルトキーペアと Amazon EC2 の Windows Server インスタンスの関連付けが解除されます。詳細については、「Lightsail スナップショットから作成した Amazon EC2 の Windows Server インスタンスを保護する」を参照してください。

Lightsail スナップショットから起動された Windows Server Amazon EC2 インスタンスを保護する

Amazon Lightsail スナップショットから作成した Amazon Elastic Compute Cloud (Amazon EC2) の Windows Server インスタンスのセキュリティを向上させるために、デフォルトの管理者パスワードを変更することをお勧めします。これにより、Lightsail のキーペアと Amazon EC2 の Windows Server インスタンスの関連付けが解除されます。

> **Note**
>
> Lightsail のスナップショットから Amazon EC2 の Linux または Unix インスタンスを作成した場合は、これらのインスタンスを保護するためにいくつかのステップを実行する必要があ

ります。詳細については、「Lightsail スナップショットから作成した Amazon EC2 の Linux または Unix インスタンスを保護する」を参照してください。

目次

- Amazon EC2 の Windows Server インスタンスに接続する
- Amazon EC2 の Windows Server インスタンスのデフォルトの管理者パスワードを変更する

Amazon EC2 の Windows Server インスタンスに接続する

Windows Server の管理者パスワードを変更するには、リモートデスクトッププロトコル (RDP) を使用して Amazon EC2 の Windows Service インスタンスに接続します。インスタンスに接続する方法については、「Lightsail スナップショットから作成した Amazon EC2 の Windows Server インスタンスに接続する」を参照してください。

Amazon EC2 でインスタンスに接続したら、このガイドの「Amazon EC2 で Windows サーバーインスタンスのデフォルト管理者パスワードを変更する」セクションに進んでください。

Amazon EC2 の Windows Server インスタンスのデフォルトの管理者パスワードを変更する

Windows Server インスタンスのデフォルトパスワードを変更し、Lightsail のキーペアと Amazon EC2 の新しい Windows Server インスタンスとの関連付けを解除します。

Amazon EC2 の Windows Server インスタンスのデフォルト管理者パスワードを変更する

1. インスタンスへの RDP 接続を確立したら、コマンドプロンプトを開いて次のコマンドを入力します。

   ```
   net user Administrator "Password"
   ```

   コマンドで、*Password* を新しいパスワードに置き換えます。

   例:

   ```
   net user Administrator "EXAMPLE%4=Bwk^GEAg8$u@5"
   ```

   次のような結果が表示されます。

```
C:\users\Administrator>net user Administrator "EXAMPLE%4=Bwk^GEAg8$u@5"
The command completed successfully.


C:\users\Administrator>
```

2. 新しいパスワードを安全な場所に保存します。Amazon EC2 コンソールを使用して新しいパスワードを取得することはできません。コンソールで取得できるのはデフォルトのパスワードのみです。パスワードの変更後に、デフォルトのパスワードを使用してインスタンスに接続しようとすると、認証情報が無効であるというエラーが表示されます。

   パスワードを忘れた場合、またはパスワードの有効期限が切れた場合は、新しいパスワードを生成できます。パスワードをリセットする手順については、Amazon EC2 ドキュメントの「紛失または期限切れの Windows 管理者パスワードのリセット」を参照してください。

Lightsail インスタンスの CloudFormation スタックを表示する

Amazon Lightsail は を使用して CloudFormation 、エクスポートされたスナップショットから Amazon Elastic Compute Cloud (Amazon EC2) インスタンスを作成します。Lightsail コンソールまたは Lightsail API を使用して Amazon EC2 インスタンスの作成をリクエストすると、CloudFormation スタックが作成されます。スタックは、Amazon Web Services (AWS) アカウントで一連のアクションを実行し、インスタンスに関連するすべてのリソースを作成します。たとえば、Amazon マシンイメージ (AMI) から Amazon EC2 インスタンスを作成し、EBS スナップショットから Elastic Block Store (EBS) システムボリュームを作成して、インスタンスのセキュリティグループを作成します。 CloudFormation スタックの詳細については、 CloudFormation ドキュメントの「スタックの使用」を参照してください。

CloudFormation スタックには、 Lightsailコンソールまたは CloudFormation コンソールからアクセスできます。このガイドでは両方のアクセス方法を示します。

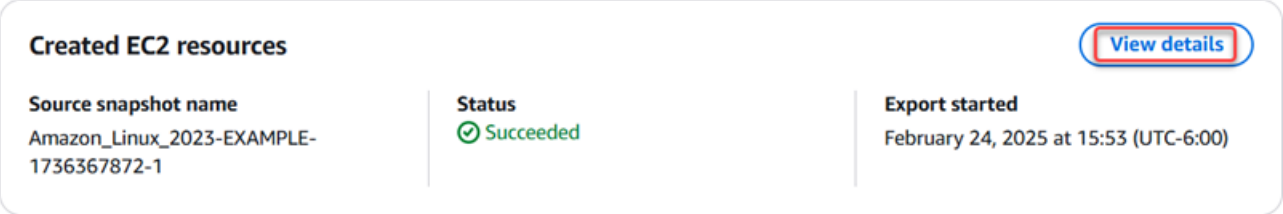
> **Note**
>
> Amazon EC2 リソースの作成に使用される CloudFormation スタックは、Amazon EC2 リソースに永続的にリンクされます。スタックを削除すると、すべての関連リソースが自動的に削除されます。このため、 によって作成された CloudFormation スタックを削除せず Lightsail、代わりに Amazon EC2 EC2 リソースを削除する必要があります。

Lightsail コンソールから CloudFormation スタックにアクセスする

Lightsail コンソールまたは Lightsail API を使用して Amazon EC2 でインスタンスを作成すると、CloudFormation スタックが作成され、そのステータスはLightsailコンソールのエクスポートセクションで追跡されます。[エクスポート] の詳細については、Lightsail でスナップショットのエクスポートステータスを追跡する を参照してください。

Lightsail コンソールで CloudFormation スタックを表示するには

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[エクスポート] を選択します。
3. 以前作成された Amazon EC2 インスタンスの CloudFormation スタックにアクセスするには、[Amazon EC2 で作成されたリソース] でラベル付けされたタスクの [詳細を表示] を選択してください。

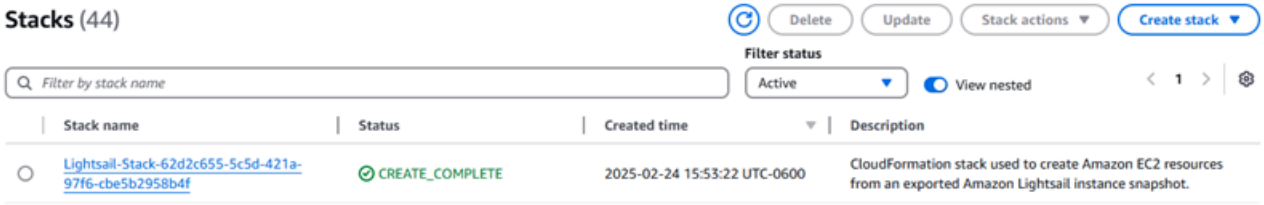


4. 確認ページにタスクの CloudFormation スタックが表示されます。スタック名を選択して、CloudFormation コンソールでスタックの詳細を開きます。

CloudFormation コンソールでのスタックへのアクセス

CloudFormation コンソールからスタックの詳細にアクセスすることもできます。Lightsail で作成されたスタックは、次のスクリーンショットに示すように、名前が「Lightsail -stack」で始まり、説明が「CloudFormation stack used to create Amazon EC2 resources」と表示されます。

スタックのステータスが [CREATE_IN_PROGRESS] である場合、エクスポートした Lightsail スナップショットから Amazon EC2 リソースが作成中です。スタックのステータスが [CREATE_COMPLETED] である場合、Amazon EC2 リソースの作成プロセスは完了しています。スタックで作成されたリソースを表示するには、スタック名の横にあるチェックボックスをオンにして、[リソース] タブを選択します。


Stacks (44)
Delete
Update
Stack actions
Create stack
Filter status
Filter by stack name
Active
View nested
1
Stack name
Status
Created time
Description
Lightsail-Stack-62d2c655-5c5d-421a-97f6-cbe5b2958b4f
CREATE_COMPLETE
2025-02-24 15:53:22 UTC-0600
CloudFormation stack used to create Amazon EC2 resources from an exported Amazon Lightsail instance snapshot.

Lightsail でウェブサイトのドメインを登録および管理します。

ウェブサイトには example.com などの名前が必要です。Amazon Lightsail では、ウェブサイトの名前 (つまりドメイン名) を登録できます。ウェブサイトにアクセスするには、ウェブブラウザにドメイン名を入力します。

Amazon Lightsail コンソールの [ドメインと DNS] タブを使用して、ドメイン名を登録および管理します。Lightsail は Amazon Route 53 という可用性およびスケーラビリティに優れたドメインネームシステム (DNS) のウェブサービスを使用して、ドメインを登録します。ドメインが登録されたら、Lightsail リソースに割り当てたり、DNS レコードを管理したりできます。DNS の一般的な情報については、「DNS」を参照してください。

Amazon Lightsail でのドメイン登録の詳細については、この先を参照してください。

内容

- ドメイン登録の仕組み
- で登録できるドメインLightsail
- ドメイン登録の料金

ドメイン登録の仕組み

Amazon Lightsail でドメイン名を登録する方法の概要を次に示します。

1. 目的のドメイン名がインターネットで使用できることを確認します。希望するドメイン名が使用できない場合は、他の名前を試したり、.com などの最上位ドメインのみを .org や .net などの別の最上位ドメインに変更したりできます。Lightsail でサポートされている最上位ドメイン (TLD) の一覧については、「Amazon Lightsail で登録できるドメイン」を参照してください。
2. Lightsail を使用してドメイン名を登録します。ドメインを登録するときは、ドメインの所有者の名前と連絡先情報、その他の連絡先の名前とその情報を提供します。

登録プロセスの最後に、お客様から提供された情報がドメインのレジストラに送信されます。ドメインレジストラは、特定の TLD のドメイン登録を処理する ICANN (Internet Corporation for Assigned Names and Numbers) から認定を受けている会社です。ドメインのレジストラは、Amazon Registrar か、当社のレジストラアソシエイトである Gandi のいずれかです。

Amazon Registrar と Gandi では、デフォルトで非表示になる情報が異なります。Amazon Registrar, Inc. はお客様の連絡先情報をすべて非表示にし、Gandi は組織名を除くすべての連絡先情報を非表示にします。

- ドメインのレジストラを特定するには、「Amazon Lightsail に登録できるドメイン」を参照してください。
- レジストラはお客様の情報をドメインのレジストリに送信します。レジストリとは、.com などの 1 つまたは複数の最上位ドメインのドメイン登録を販売する会社です。
- レジストリは、お客様のドメインに関する情報を自社のデータベースに保存し、その情報の一部をパブリック WHOIS データベースにも保存します。

ドメイン名を登録する方法の詳細については、「新しいドメインを登録する」を参照してください。

Lightsail を使用してドメインを登録すると、Route 53 はネームサーバーのセットをドメインに割り当てることにより、ドメインの DNS サービスになります。ネームサーバーとは、ドメイン名を IP アドレスに変換するのに役立つサーバーです。

Lightsail は自動的に次の処理を実行して、ドメインの DNS サービスになります。

- ドメインと同じ名前の Lightsail DNS ゾーンを作成します。
- 4 つの一連のネームサーバーを Lightsail DNS ゾーンに割り当てます。
- ドメインの Route 53 ネームサーバーを Lightsail DNS ゾーンのネームサーバーに置き換えます。

別のレジストラにドメイン名を既に登録している場合は、ドメインの DNS の管理を Lightsail に移管するように選択できます。この操作は、Lightsail の他の機能は使用する場合は不要です。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。

で登録できるドメインLightsail

Lightsail は、Route 53 と同じ汎用最上位ドメイン (TLD) を使用します。Lightsail でドメインの登録に使用できる汎用 TLD の一覧については、Amazon Route 53 デベロッパーガイドの「Amazon Route 53 に登録できるドメイン」を参照してください。

TLD がリストに含まれていない場合、または地理的ドメインを登録する場合は、Route 53 コンソールを使用することをお勧めします。地理的ドメインは、Route 53 を使用して登録してから Lightsail コンソールで使用できるようになります。詳細については、「Amazon Route 53 デベロッパーガイド」の「地理的最上位ドメイン」を参照してください。

ドメイン登録の料金

Lightsail では、ドメイン登録には Route 53 が使用されます。したがって、Route 53 の料金は Lightsail 登録にも適用されます。

ドメイン登録のコストの詳細については、Amazon Route 53 デベロッパーガイドの「Amazon Route 53 に登録できるドメイン」を参照してください。

ドメインに関する追加情報

次の記事は、Lightsail でのドメインの管理に役立ちます。

- DNS
- ドメイン名をフォーマットする
- Amazon Route 53 で Lightsail ドメインを管理する
- ドメインの DNS レコードを管理する DNS ゾーンを作成する
- ドメイン登録更新
- DNS ゾーンを編集または削除する
- ドメインをロードバランサーにポイントする
- ドメインをディストリビューションにポイントする
- ドメインをインスタンスにポイントする
- ドメインへのトラフィックをコンテナサービスにルーティングする

Lightsail の DNS について

ユーザーは、IPv4 または IPv6 となり得る Lightsail インスタンスのパブリックインターネットプロトコル (IP) アドレスを参照することで、このインスタンスのウェブアプリケーションにアクセスできます。ただし、IP アドレスは複雑で覚えにくいという欠点があります。覚えやすいドメイン名 (`example.com` など) に変換すると、ユーザーはインスタンスのウェブアプリケーションにアクセスしやすくなります。そのためには、ドメインネームシステム (DNS) を使用します。DNS は、登録されたドメイン名を IP アドレスにマッピングするディレクトリとして機能します。

ドメイン名のトラフィックを Lightsail インスタンスに送信するには、ドメイン名をインスタンスの静的 IPv4 アドレスに指すアドレス (A) レコードを追加するか、インスタンスの IPv6 アドレスを指す AAAA レコードを追加します。Lightsail を使用してドメイン名を登録した場合は、ドメイン名の登録

時に作成された DNS ゾーンから DNS レコードを管理できます。ドメインが別のレジストラを通じて登録されている場合は、そのレジストラで DNS レコードを管理することも、ドメインの DNS の管理を Lightsail に移管することもできます。

ドメイン名を Lightsail インスタンスに簡単にマッピングするには、DNS ゾーンを作成して、ドメインの DNS レコードの管理を Lightsail に引き渡すことをお勧めします。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。Lightsail では、最大 6 つの DNS ゾーンを作成できます。6 つ以上の DNS ゾーンが必要な場合は、すべてのドメインでの DNS 管理に、 Route 53 を使用することをお勧めします。ドメイン名に Lightsail インスタンスを参照させるには、 Route 53 を使用できます。Route 53 による DNS の管理の詳細については、「Amazon Route 53 を使用してドメインをインスタンスに参照させる」を参照してください。

DNS の用語

ドメインの DNS を管理するには、以下の用語を理解しておく必要があります。

Apex ドメイン/ルートドメイン

apex ドメイン (ルートドメインとも呼ばれます) は、サブドメインパートを含まないドメインです。apex ドメインの例は `example.com` です。サブドメインの例は `www.example.com` や `blog.example.com`です。これらがサブドメインであるのは、それぞれサブドメインパートとして `www` と `blog` を含んでいるためです。

ドメインネームシステム (DNS)

DNS は、覚えやすいドメイン名 (`example.com` など) であり、ウェブサーバーの IP アドレスに対応しています。

詳細については、Wikipedia の「Domain Name System」を参照してください。

DNS レコード

DNS レコードはマッピングパラメータです。DNS サーバーに対して、ドメインやサブドメインに対応する IP アドレスやホスト名を示します。

詳細については、Wikipedia の「DNS レコードタイプの一覧」を参照してください。

DNS ゾーン

DNS ゾーンは、特定のドメイン (`example.com` など) やそのサブドメイン (`blog.example.com` など) のトラフィックをインターネットでルーティングする方法に関する情報を保持するコンテナです。

詳細については、Wikipedia の「DNS ゾーン」を参照してください。

ドメイン名レジストラ

ドメイン名レジストラ (ドメイン名プロバイダーとも呼ばれます) は、ドメイン名の割り当てを管理する企業または組織です。Lightsail、Amazon Route 53 または他のドメイン名レジストラを使用して、ドメインを購入したり、既存のドメインを管理したりできます。

詳細については、Wikipedia の「Domain name registrar」を参照してください。

ネームサーバー

ネームサーバーは、トラフィックをドメインにルーティングします。ではLightsail、ネームサーバーは覚えeasy-to-rememberドメイン名を IP アドレスに変換するのに役立つネットワークサービスを実行する AWS インスタンスです。 Lightsailには、トラフィックをドメインにルーティングするための AWS ネームサーバーオプション (などns-NN.awsdns-NN.com) がいくつか用意されています。ドメインレジストラを使用してドメインを変更するときに、これらの AWS ネームサーバーから選択できます。

詳細については、Wikipedia の「Name server」を参照してください。

サブドメイン

サブドメインは、ドメイン階層内で、上位のドメインに属するドメインのことです (ルートドメインを除く)。たとえば、`blog` は `blog.example.com` サブドメインのサブドメインパートです。

詳細については、Wikipedia の「Subdomain」を参照してください。

有効期限 (TTL)

TTL は、ローカル解決ネームサーバーでの DNS レコードの有効期限を表します。たとえば、期限が短いほど変更が有効になるまでに待機する時間が短くなります。TTL は、Lightsail DNS ゾーンでは設定できません。代わりに、Lightsail のすべての DNS レコードはデフォルトで TTL が 60 秒に設定されます。

詳細については、Wikipedia の「Time to live」を参照してください。

ワイルドカード DNS レコード

ワイルドカード DNS レコードは、存在しないドメイン名に対するリクエストのマッチングを行います。ワイルドカード DNS レコードを指定するには、ドメイン名の左端にアスタリスク記号 (*) を使用します (例: `*.example.com` または `*example.com`)。

> **Note**
>
> Lightsail DNS ゾーンでは、ネームサーバー (NS) レコードに定義されている、ネームサーバードメイン (例: *awsdns.com) 向けのワイルドカードレコードがサポートされます。

Lightsail DNS ゾーンでサポートされる DNS レコードタイプ

アドレス (A) レコード

A レコードは、ドメイン (example.com など) やサブドメイン (blog.example.com など) をウェブサーバーの IP アドレスにマッピングします。

たとえば、Lightsail DNS ゾーンで、example.com (apex ドメイン) のウェブトラフィックをインスタンスにルーティングするとします。A レコードを作成し、@ シンボルを [サブドメイン] のテキストボックスに入力し、 ウェブサーバーの IP アドレスを [Resolves to address] (解決するアドレス) テキストボックスに入力します。

A レコードの詳細については、Wikipedia の「DNS レコードタイプの一覧」を参照してください。

AAAA レコード

AAAA レコードは、ドメイン (example.com など) やサブドメイン (blog.example.com など) をウェブサーバーの IPv6 アドレスにマッピングします。

たとえば、Lightsail DNS ゾーンで、IPv6 プロトコルで example.com (ドメインの apex) のウェブトラフィックをインスタンスに送信するとします。AAAA レコードを作成し、@ シンボルを [サブドメイン] のテキストボックスに入力し、 ウェブサーバーの IP アドレスを [Resolves to address] (解決するアドレス) テキストボックスに入力します。

AAAA レコードの詳細については、Wikipedia の「IPv6 のドメインネームシステム」を参照してください。

> **Note**
>
> Lightsail は、静的 IPv6 アドレスをサポートしません。Lightsail リソースを削除し新しいリソースを作成するか、同じリソースで IPv6 を無効にして再度有効にする場合は、リソースの最新の IPv6 アドレスを反映するために、AAAA レコードの更新が必要な場合があります。

正規名 (CNAME) レコード

CNAME レコードは、エイリアスまたはサブドメイン (`blog.example.com` など) を別のドメインまたはサブドメインにマッピングします。

たとえば、Lightsail DNS ゾーンで、`www.example.com` のウェブトラフィックを `example.com` に振り向けるとします。この場合、`www` のエイリアス CNAME レコードを作成し、「解決先」アドレスとして `example.com` を使用します。

詳細については、Wikipedia の「CNAME Record」を参照してください。

メールエクスチェンジャ (MX) レコード

MX レコードは、サブドメイン (`mail.example.com` など) を E メールサーバーアドレスにマッピングします。複数のサーバーを定義する場合は、優先度の値を設定します。

たとえば、Lightsail DNS ゾーンで、`mail.example.com` のメールを `10 inbound-smtp.us-west-2.amazonaws.com` Amazon WorkMail メールサーバーに振り向けるとします。この場合に作成する MX レコードでは、サブドメインとして `example.com`、優先度として `10`、「解決先」アドレスとして `inbound-smtp.us-west-2.amazonaws.com` を設定します。

詳細については、Wikipedia の「MX レコード」を参照してください。

ネームサーバー (NS) レコード

NS レコードは、サブドメイン (`test.example.com` など) をネームサーバー (`ns-NN.awsdns-NN.com` など) に委任します。

詳細については、Wikipedia の「Name server」を参照してください。

サービスロケーター (SRV) レコード

SRV レコードは、サブドメイン (`service.example.com` など) をサービスアドレスにマッピングします。優先度、重み、およびポート番号の値を設定します。通常 SRV レコードに関連付けられるサービスとして、テレフォニーやインスタントメッセージングなどがあります。

たとえば、Lightsail DNS ゾーンで、`service.example.com` のトラフィックを `1 10 5269 xmpp-server.example.com` に振り向けるとします。この場合に作成する SRV レコードでは、優先度として `1`、重みとして `10`、ポート番号として `5269`、「マッピング先」アドレスとして `xmpp-server.example.com` を設定します。

詳細については、Wikipedia の「SRV Record」を参照してください。

テキスト (TXT) レコード

TXT レコードは、サブドメインをプレーンテキストにマッピングします。TXT レコードを作成し、サービスプロバイダーに対してドメインの所有権を確認します。

たとえば、Lightsail DNS ゾーンで、`_amazonchime.example.com` ホスト名が照会されたときに `23223a30-7f1d-4sx7-84fb-31bdes7csdbb` を返すとします。この場合に作成する TXT レコードでは、サブドメイン値として `_amazonchime`、「応答先」値として `23223a30-7f1d-4sx7-84fb-31bdes7csdbb` を設定します。

詳細については、Wikipedia の「TXT Record」を参照してください。

Lightsail インスタンスのドメインレコードを管理する DNS ゾーンを作成する

`example.com` などのドメイン名のトラフィックを Amazon Lightsail インスタンスにルーティングするには、ドメインのドメインネームシステム (DNS) にレコードを追加します。ドメインの DNS レコードは、ドメインを登録したレジストラを使用して管理することも、Lightsail を使用して管理することもできます。

ドメインの DNS レコードの管理を Lightsail に引き渡すことをお勧めします。これにより、ドメインの管理とリソースのコンピューティングをまとめて 1 つの場所 Lightsail で効率的に行うことができます。ドメインの DNS レコードは、Lightsail を使用して、Lightsail DNS ゾーンを作成することで管理できます。最大 6 つの Lightsail DNS ゾーンを作成できます。6 つを超えるドメイン名を管理しているために、6 つ以上の DNS ゾーンが必要な場合は、すべてのドメインでの DNS 管理に、Amazon Route 53 を使用することをお勧めします。Route 53 を使用して、ドメインのトラフィックを Lightsail リソースにルーティングできます。Route 53 による DNS の管理の詳細については、「Amazon Route 53 を使用してドメインをインスタンスに参照させる」を参照してください。

このトピックでは、ドメインのための Lightsail DNS ゾーンを作成し、ドメインの DNS レコードの管理を Lightsail に引き渡す方法を示します。ドメインの DNS レコードの管理を Lightsail に引き渡した後でも、ドメインの更新や請求は引き続きレジストラで管理します。

> **Important**
>
> ドメインの DNS に対して行った変更は、インターネットの DNS を通じて伝播されるまで数時間かかる場合があります。このため、Lightsail が伝播するよう管理を転送させるあいだは、ドメインの DNS レコードは、ドメインの現在の DNS ホスティングプロバイダーで保管

します。これにより、転送の実行中も、ドメインのトラフィックが途切れることなくリソースにルーティングされます。

ステップ 1: 前提条件を満たす

以下の前提条件を満たします (まだ満たしていない場合)。

1. ドメイン名を登録します。次に、ドメインのネームサーバーを編集するための管理者アクセス権があることを確認します。

   登録済みのドメイン名が必要な場合、Lightsail を使用してドメインを登録します。詳細については、「ドメインの登録」を参照してください。
2. ドメインの必要な DNS レコードタイプが Lightsail の DNS ゾーンでサポートされていることを確認します。現在、Lightsail のDNS ゾーンでは、アドレス (A および AAAA)、正規名 (CNAME)、メールエクスチェンジャ (MX)、ネームサーバー (NS)、サービスロケーター (SRV)、およびテキスト (TXT) の各レコードタイプがサポートされています。NS レコードには、ワイルドカード DNS レコードエントリを使用できます。

   ドメインに必要な DNS レコードタイプが Lightsail の DNS ゾーンでサポートされていない場合は、より多くのレコードタイプがサポートされるため、ドメインの DNS ホスティングプロバイダーとして Route 53 を使用できます。詳細については、「Amazon Route 53 デベロッパーガイド」の「サポートされる DNS レコードタイプ」と「Amazon Route 53 を既存ドメインの DNS サービスとして使用する」を参照してください。
3. ドメインがポイントする Lightsail インスタンスを作成します。詳細については、「インスタンスを作成する」を参照してください。
4. 静的 IP を作成して Lightsail インスタンスにアタッチします。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。

ステップ 2: Lightsail コンソールで DNS ゾーンを作成する

次の手順を実行して Lightsail に DNS ゾーンを作成します。DNS ゾーンを作成するとき、DNS ゾーンが適用されるドメイン名を指定する必要があります。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ドメインと DNS] を選択します。[DNS ゾーンの作成] を選択します。

3. 以下のオプションのいずれかを選択してください。

   - [Amazon Route 53 に登録されているドメインを使用]して、Amazon Route 53 に登録されたドメインを指定します。
   - [Use a domain from another registrar] (別のレジストラのドメインを使用) で、別のレジストラを使用して登録されたドメインを指定します。

4. 登録済みドメイン名 (`example.com` など) を選択または入力します。

   ドメイン名を入力するときに `www` を含める必要はありません。このガイドの「ステップ 3: DNS ゾーンにレコードを追加する」セクションで、アドレス (A) レコードを使用して `www` を追加できます。

   > **Note**
   >
   > Lightsail DNS ゾーンは、バージニア (`us-east-1`) の AWS リージョンに作成されます。そのリージョンでリソースに Lightsail DNS ゾーンと同じ名前 (`example.com`) を付けると、リソース名の競合エラー (「一部の名前は既に使用されています」) が表示されます。
   > このエラーを解決するには、リソースのスナップショットを作成します。スナップショットから新しいリソースを作成して、新しい一意の名前を付けます。次に、Lightsail DNS ゾーンを作成するドメインと同じ名前の元のリソースを削除します。

5. [DNS ゾーンの作成] を選択します。

   ユーザーは DNS ゾーンの [Assignments] (割り当て) ページにリダイレクトされ、ドメインリソースの割り当てを管理することが可能になります。割り当てを使用して、ドメインを、ロードバランサーやインスタンスなどの Lightsail リソースにポイントします。

ステップ 3: DNS ゾーンにレコードを追加する

ドメインの DNS ゾーンにレコードを追加するには、以下のステップを実行します。DNS レコードは、そのドメインにインターネットトラフィックがルーティングされる方法を指定します。たとえば、ドメインの apex (`example.com`) のトラフィックを 1 つのインスタンスにルーティングし、サブドメイン (`blog.example.com` など) のトラフィックを異なるインスタンスにルーティングできます。

1. DNS ゾーンの割り当てページで、[DNS records] (DNS レコード) タブを選択します。

DNS ゾーンは、Lightsail コンソールの [Domains & DNS] (ドメイン & DNS) タブに一覧表示されます。

> **Note**
>
> DNS ゾーンの [割り当て] ページでは、ドメインが指す Lightsail リソースを追加、削除、または変更できます。Lightsail インスタンス、ディストリビューション、コンテナサービス、ロードバランサー、静的 IP アドレスなどでドメインに指定できます。[DNS records] (DNS レコード) のページでは、ドメインの DNS レコードを追加、編集、または削除できます。

2. 以下のいずれかのレコードタイプを選択します。

アドレス (A) レコード

A レコードは、ドメイン (`example.com` など) やサブドメイン (`blog.example.com` など) をウェブサーバーまたはインスタンスの IPv4 アドレス (`192.0.2.255` など) にマッピングします。

1. [Record name] (レコード名) テキストボックスに、レコードのターゲットサブドメインを入力するか、@ 記号を入力してドメインの最上位を定義します。
2. [Resolves to (解決先)] テキストボックスに、レコードのターゲット IP アドレスを入力し、実行中のインスタンスまたは設定済みのロードバランサーを選択します。実行中のインスタンスを選択すると、そのインスタンスのパブリック IP アドレスが自動的に追加されます。
3. AWS リソースエイリアスを選択して、ディストリビューションやコンテナサービスなどの Lightsailおよび AWS リソースにトラフィックをルーティングします。DNS ゾーン内のあるレコードから別のレコードにトラフィックをルーティングできます。

> **Note**
>
> Lightsail インスタンスに静的 IP をアタッチしてから、レコードが解決する値として静的 IP を選択することをお勧めします。詳細については、「静的 IP を作成する」を参照してください。

AAAA レコード

AAAA レコードは、ドメイン (example.com など) やサブドメイン (blog.example.com など) をウェブサーバーまたはインスタンスの IPv6 アドレス (2001:0db8:85a3:0000:0000:8a2e:0370:7334 など) にマッピングします。

> **Note**
>
> Lightsail は静的 IPv6 アドレスをサポートしていません。Lightsail リソースを削除し新しいリソースを作成するか、同じリソースで IPv6 を無効にして再度有効にする場合は、リソースの最新の IPv6 アドレスを反映するために、AAAA レコードの更新が必要な場合があります。

1. [Record name] (レコード名) テキストボックスに、レコードのターゲットサブドメインを入力するか、@ 記号を入力してドメインの最上位を定義します。
2. [Resolves to (解決先)] テキストボックスに、レコードのターゲット IPv6 アドレスを入力し、実行中のインスタンスまたは設定済みのロードバランサーを選択します。実行中のインスタンスを選択すると、そのインスタンスのパブリック IPv6 アドレスが自動的に追加されます。
3. AWS リソースエイリアスを選択して、ディストリビューションやコンテナサービスなどの Lightsailおよび AWS リソースにトラフィックをルーティングします。DNS ゾーン内のあるレコードから別のレコードにトラフィックをルーティングできます。

正規名 (CNAME) レコード

CNAME レコードは、www.example.com などのエイリアスまたはサブドメインを example.com などの別のドメイン、または blog.example.com などの別のサブドメインにマップします。

1. [Record name] (レコード名) テキストボックスに、レコードのサブドメインを入力します。
2. [Route traffic to] (トラフィックのルーティング先) テキストボックスに、レコードのターゲットドメインまたはサブドメインを入力します。

メールエクスチェンジャ (MX) レコード

MX レコードは、サブドメイン (mail.example.com など) を E メールサーバーアドレスにマッピングします。複数のサーバーを定義する場合は、優先度の値を設定します。

1. [Record name] (レコード名) テキストボックスに、レコードのサブドメインを入力します。
2. [優先度] テキストボックスに、レコードの優先度を入力します。これは、複数のサーバーにレコードを追加する場合に重要です。
3. [Route traffic to] (トラフィックのルーティング先) テキストボックスに、レコードのターゲットドメインまたはサブドメインを入力します。

サービスロケーター (SRV) レコード

SRV レコードは、サブドメイン (`service.example.com` など) をサービスアドレスにマッピングします。優先度、重み、およびポート番号の値を設定します。通常 SRV レコードに関連付けられるサービスとして、テレフォニーやインスタントメッセージングなどがあります。

1. [Record name] (レコード名) テキストボックスに、レコードのサブドメインを入力します。
2. [優先度] テキストボックスに、レコードの優先度を入力します。
3. [Weight (重み)] テキストボックスに、同じ優先度を持つ SRV レコードの相対的な重みを入力します。
4. [Route traffic to] (トラフィックのルーティング先) テキストボックスに、レコードのターゲットドメインまたはサブドメインを入力します。
5. [Port (ポート)] テキストボックスに、サービスへの接続を確立できるポート番号を入力します。

テキスト (TXT) レコード

TXT レコードは、サブドメインをプレーンテキストにマッピングします。TXT レコードを作成し、サービスプロバイダーに対してドメインの所有権を確認します。

1. [Record name] (レコード名) テキストボックスに、レコードのサブドメインを入力します。
2. [Responds with (応答内容)] テキストボックスに、サブドメインに対してクエリが実行されたときに返すテキストレスポンスを入力します。

> **Note**
>
> 入力テキストは、引用符で囲む必要はありません。

3. レコードの追加が終了したら、保存アイコンを選択して変更を保存します。

レコードが DNS ゾーンに追加されます。ドメインの DNS ゾーンに複数のレコードを追加する場合は、上の手順を繰り返します。

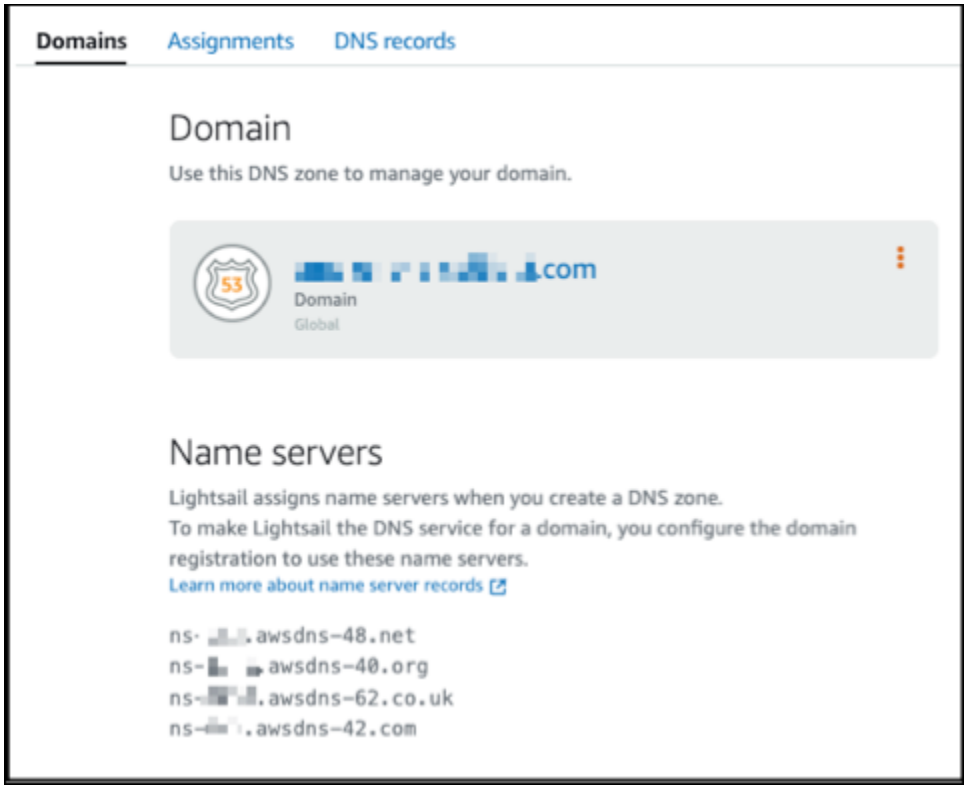
> **Note**
>
> DNS レコードの有効期限 (TTL) は、Lightsail の DNS ゾーンで設定できません。代わりに、Lightsail のすべての DNS レコードはデフォルトで TTL が 60 秒に設定されます。詳細については、Wikipedia の「Time to live」を参照してください。

ステップ 4: ドメインの現在の DNS ホスティングプロバイダーでネームサーバーを変更する

ドメインの DNS レコードの管理を Lightsail に引き渡すには、以下のステップを実行します。これを行うには、ドメインの現在の DNS ホスティングプロバイダーのウェブサイトにサインインし、ドメインのネームサーバーを Lightsail ネームサーバーに変更します。

> **Important**
>
> ウェブトラフィックが現在ドメインにルーティングされている場合は、すべての既存の DNS レコードが Lightsail DNS ゾーンに存在することを確認してから、ドメインの現在の DNS ホスティングプロバイダーでネームサーバーを変更します。これにより、Lightsail DNS ゾーンへの引き渡し後も、トラフィックは引き続き中断されることなく流れます。

1. ドメインの DNS ゾーンの管理ページに記載されている Lightsail ネームサーバーを書き留めてください。ネームサーバーは、Lightsail DNS ゾーンの [ドメイン] タブに表示されています。



2. ドメインの現在の DNS ホスティングプロバイダーのウェブサイトにサインインします。
3. ドメインのネームサーバーを編集できるページを見つけます。

   このページを見つける詳しい方法については、ドメインの現在の DNS ホスティングプロバイダーのドキュメントを参照してください。
4. Lightsail のネームサーバーを入力し、表示されている他のネームサーバーを削除します。
5. 変更内容を保存します。

   ネームサーバーの変更がインターネットの DNS を通じて伝播されるまで数時間かかる場合があります。完了すると、ドメインのインターネットトラフィックが Lightsail DNS ゾーンを経由してルーティングされ始めます。

次の手順

- DNS ゾーンを編集する
- ロードバランサーを作成してインスタンスをアタッチする

Lightsail DNS ゾーンを編集する

ドメインの DNS ゾーンの DNS レコードを編集します。ドメインの DNS レコードの管理を別の DNS ホスティングプロバイダーに引き渡したり、ドメインを登録したレジストラに戻したりする場合は、ドメインの DNS ゾーンを Amazon Lightsail で削除することもできます。詳細については、???を参照してください。

> **Note**
>
> Lightsail コンソールで DNS エディタを使用してレコードを編集するには、事前にドメインの DNS レコードの管理を Lightsail に引き渡す必要があります。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。

DNS レコードを編集する

ドメインの DNS ゾーンの DNS レコードは、Lightsail コンソールを使用していつでも編集できます。

DNS ゾーンを編集するには

1. Lightsail コンソールにサインインします。
2. Lightsail コンソールのホームページの左側のナビゲーションペインで [ドメインと DNS] を選択します。
3. 編集したい DNS ゾーンの名前を選択します。
4. DNS ゾーンの [DNS レコード] ページで、削除したいレコードの横にある [削除] アイコンを選択します。
5. 終了したら、保存アイコンを選択して変更を保存します。

> **Note**
>
> DNS レコードの変更がインターネットの DNS を通じて伝播されるまで数時間かかる場合があります。

Lightsail で DNS ゾーンを削除する

場合によっては、Amazon Lightsail で設定したドメインの DNS レコードを管理するための DNS ゾーンを完全に削除したいことがあります。DNS 管理を別のプロバイダーに移管するか、ドメインレジストラに戻したい場合があります。DNS ゾーンの削除は簡単なプロセスですが、ドメインのトラフィックが正しくルーティングされるように事前に計画することが重要です。Lightsail で DNS ゾーンを削除する手順を確認しましょう。

> **Important**
>
> 引き続きドメインを通じてトラフィックをルーティングする場合は、Lightsail でドメインの DNS ゾーンを削除する前に、別の DNS ホスティングプロバイダーを準備します。そうしないと、Lightsail の DNS ゾーンを削除したときに、ウェブサイトへのすべてのトラフィックが停止します。

DNS ゾーンを削除するには

1. Lightsail コンソールのホームページの左側のナビゲーションペインで [ドメインと DNS] を選択します。
2. 削除する DNS ゾーンの名前を選択します。
3. 縦三点リーダーメニュー (:) を選択します。次に、[Delete] (削除) オプションを選択します。
4. [Delete DNS zone] (DNS ゾーンを削除) を選択して削除を確定します。

   DNS ゾーンが Lightsail から削除されます。

Lightsail のウェブサイトにインターネットトラフィックがルーティングされる方法について説明する

スマートフォン、ラップトップ、ウェブサイトサーバーなど、インターネット上のすべてのコンピュータは、一意の文字列を使用して相互に通信します。これらの文字列は、IP アドレスと呼ばれ、次のいずれかの形式になります。

- インターネットプロトコルバージョン 4 (IPv4) 形式 (192.0.2.44 など)
- インターネットプロトコルバージョン 6 (IPv6) 形式 (2001:DB8::/32 など)

ブラウザを開いてウェブサイトにアクセスするときは、このような長い文字列を覚えて入力する必要はありません。代わりに、example.com のようなドメイン名を入力しても、正しい場所にアクセスできます。そのためには、ドメインネームシステム (DNS) を使用します。DNS は、登録されたドメイン名を IP アドレスにマッピングするディレクトリとして機能します。

目次

- ドメインのインターネットトラフィックをルーティングするように Lightsail を設定する方法の概要
- ドメインにインターネットトラフィックがルーティングされる方法
- 次のステップ

ドメインのインターネットトラフィックをルーティングするように Lightsail を設定する方法の概要

ここでは、Lightsail を使用して、ウェブサイトやウェブアプリケーションにインターネットトラフィックをルーティングするようにドメインを登録および設定する方法の概要を示します。

1. ドメイン名を登録する。概要については、「ドメイン登録」を参照してください。
2. ドメイン名を登録すると、Lightsail によってドメインと同じ名前の DNS ゾーンが自動的に作成されます。
3. Lightsail コンソールでは、インスタンスやロードバランサーなどの Lightsail リソースにドメインを簡単に割り当てることができます。DNS ゾーンに DNS レコードを作成して、リソースにトラフィックをルーティングすることもできます。各レコードには、ドメインのトラフィックをどのようにルーティングするかについて、以下のような情報が含まれます。

   名前

   レコードの名前は、ドメイン名 (example.com) またはサブドメイン名 (www.example.com、retail.example.com) に対応します。DNS ゾーン内の各レコードの名前は、DNS ゾーンの名前で終わる必要があります。例えば、DNS ゾーンの名前が example.com の場合、すべてのレコード名は example.com で終わる必要があります。

   タイプ

   レコードタイプは、通常、トラフィックをルーティングする先のリソースのタイプによって決まります。例えば、トラフィックを E メールサーバーにルーティングするには、[Type] (タイプ) で [MX] を指定します。ドメイン名のトラフィックを Lightsail インスタンスにルーティングするに

は、ドメイン名がインスタンスの静的 IPv4 アドレスを指す A レコードを追加するか、インスタンスの IPv6 アドレスを指す AAAA レコードを追加します。

4. ターゲット

   ターゲットとは、トラフィックのルーティング先です。トラフィックを Lightsail インスタンス、Lightsail コンテナサービス、その他の Lightsail リソースにルーティングするエイリアスレコードを作成できます。詳細については、「DNS」を参照してください。

ドメインにインターネットトラフィックがルーティングされる方法

インスタンス、ロードバランサー、ディストリビューション、コンテナサービスなどのリソースにインターネットトラフィックをルーティングするように Lightsail を設定した後に、www.example.com のコンテンツへのリクエストがあると、何が起こるかを次に示します。

1. ユーザーがウェブブラウザを開き、アドレスバーに「www.example.com」と入力して、Enter キーを押したとします。
2. www.example.com のリクエストは DNS リゾルバーにルーティングされます。DNS リゾルバーは通常、ユーザーのインターネットサービスプロバイダー (ISP) によって管理されています。ISP には、ケーブルインターネットプロバイダー、DSL ブロードバンドプロバイダー、企業ネットワークなどがあります。
3. ISP の DNS リゾルバーは、www.example.com のリクエストを DNS ルートネームサーバーに転送します。
4. DNS リゾルバーは www.example.com のリクエストが再びあると、今度は .com ドメインのいずれかの TLD ネームサーバーに転送します。.com ドメインのネームサーバーは、example.com ドメインに関連付けられている 4 つのネームサーバーの名前でリクエストに応答します。

   DNS リソルバーは、4 つのネームサーバーをキャッシュ (保存) します。次回に誰かが example.com を参照すると、既に example.com のネームサーバーがあるため、ステップ 3 およびステップ 4 はスキップされます。通常、ネームサーバーは 2 日間キャッシュされます。
5. DNS リゾルバーは、ネームサーバーを選択し、www.example.com のリクエストをそのネームサーバーに転送します。
6. ネームサーバーは、example.com DNS ゾーンで www.example.com レコードを検索し、関連付けられた値 (ウェブサーバーの IP アドレス 192.0.2.44 など) を取得します。次に、ネームサーバーは IP アドレスを DNS リゾルバーに返します。
7. DNS リゾルバーには最終的に、ユーザーが必要とする IP アドレスがあります。リゾルバーは、その値をウェブブラウザに返します。

8. ウェブブラウザは、DNS リゾルバーから取得した IP アドレスに www.example.com のリクエストを送信します。これは、Lightsail インスタンスで実行されているウェブサーバー、ウェブサイトエンドポイントとして設定されているコンテナサービスなど、コンテンツが置かれている場所です。
9. 192.0.2.44 にあるウェブサーバーなどのリソースは、www.example.com のウェブページをウェブブラウザに返し、ウェブブラウザはそのページを表示します。

次の手順

- DNS
- ドメインをインスタンスにポイントする
- ドメインをロードバランサーにポイントする
- ドメインをディストリビューションにポイントする

ドメイントラフィックを Lightsail インスタンスにルーティングする

Amazon Lightsail の DNS ゾーンを使用して、example.com などの登録済みドメイン名を、Lightsail インスタンス (別名、仮想プライベートサーバー (VPS)) 上で実行されているウェブサイトに向けてポイントすることができます。Lightsail アカウントには最大 6 つの DNS ゾーンを作成できます。DNS レコードタイプは、全種類サポートされているわけではありません。Lightsail DNS ゾーンの詳細については、「DNS」を参照してください。

6 つ以上の DNS ゾーンを作成する場合や、Lightsail でサポートされていない DNS レコードタイプを使用する場合は、Amazon Route 53 の ホストゾーンの使用をお勧めします。Route 53 を使用すると、DNS で、最大 500 個のドメインに対応させることができます。また、より多様な DNS レコードタイプをサポートするようになります。詳細については、Amazon Route 53 デベロッパーガイドの「ホストゾーンの使用」を参照してください。

このガイドでは、Lightsail で管理されているドメインの DNS レコードを編集して、Lightsail インスタンスに向けてポイントさせる方法を説明します。DNS の変更がインターネットの DNS を通じて伝播されるまで、最大 48 時間待機します。

前提条件

以下の前提条件を満たします (まだ満たしていない場合)。

- Lightsail を使用してドメイン名を登録する 詳細については、「新しいドメインを登録する」を参照してください。

- すでにドメインを登録しているけれども、そのレコードの管理に Lightsail を使用していない場合は、ドメインの DNS レコードの管理を Lightsail に転送する必要があります。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。
- Lightsail にアタッチされているデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して再起動するたびに変わります。パブリック IP アドレスが変わらないようにするには、静的 IP アドレスを作成して、それをインスタンスにアタッチします。このガイドでは、ドメイン内で静的 IP アドレスを解決する DNS ゾーンに DNS レコードを作成します。これにより、インスタンスの停止および再開のたびに、ドメインの DNS レコードを更新する必要がなくなります。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。

  オプション – Lightsail インスタンスでは IPv6 を有効のままにしておくことができます。この IPv6 アドレスは、インスタンスを停止および再開しても保持されます。詳細については、「IPv6 の有効化と無効化」を参照してください。

Lightsail インスタンスにドメインを割り当てる

Lightsail 内のインスタンスにドメインを割り当てるには、以下のいずれかの方法を使用します。

- インスタンスドメインタブ
- 静的 IP ドメインタブ
- DNS ゾーン割り当てタブ

インスタンスドメインタブ

Lightsail コンソールにあるインスタンスの [ドメインと DNS] セクションで、次の手順を実行してドメインを Lightsail インスタンスに割り当てます。

インスタンスの [Domains] (ドメイン) タブを使用してドメインを割り当てるには

1. Lightsail コンソール にサインインします。
2. ドメインの割り当て先となるインスタンス名を選択します。
3. [Domains] (ドメイン) タブで [Assign domain] (ドメインの割り当て) を選択します。
4. Lightsail インスタンスに割り当てるドメインを選択します。
5. ルーティング情報が正しいことを確認し、[Assign] (割り当て) を選択します。

オプション

インスタンスへのドメイン割り当てを編集また削除するには、ドメイン名の横にある編集アイコン、または、ごみ箱アイコンを選択します。

静的 IP ドメインタブ

Lightsail コンソールにある静的 IP の [ドメインと DNS] タブで、次の手順を実行してドメインを Lightsail インスタンスに割り当てます。

静的 IP の [Domains] (ドメイン) タブを使用してドメインを割り当てるには

1. Lightsail コンソール にサインインします。
2. [ネットワーキング] タブを選択してください。
3. ドメインの割り当て先となる静的 IP を選択します。
4. [Domains] (ドメイン) タブで [Assign domain] (ドメインの割り当て) を選択します。
5. 静的 IP に割り当てるドメインを選択します。
6. ルーティング情報が正しいことを確認し、[Assign] (割り当て) を選択します。

オプション

静的 IP アドレスへのドメイン割り当てを編集または削除するには、ドメイン名の横にある編集アイコン、または、ごみ箱アイコンを選択します。

DNS ゾーン割り当てタブ

DNS ゾーンの [Assignments] (割り当て) タブで、以下の手順を実行して、Lightsail インスタンスにドメインを割り当てます。

[Assignments] (割り当て) タブを使用してドメインを割り当てるには

1. Lightsail コンソール にサインインします。
2. [Domains & DNS] (ドメインと DNS) タブを選択します。
3. 対象のドメイン名に対して使用する DNS ゾーンを選択します。
4. [Assignments] (割り当て) タブで [Add assignment] (割り当てを追加) を選択します。
5. Lightsail インスタンスに割り当てるドメイン名を選択します。静的 IP がインスタンスにまだアタッチされていない場合は、アタッチするよう求められます。
6. ルーティング情報が正しいことを確認し、[Assign] (割り当て) を選択します。

オプション

このリソースでのドメイン割り当てを編集または削除するには、ドメイン名の横にある編集アイコン、または、ごみ箱アイコンを選択します。

ドメインを Lightsail ロードバランサーにポイントする

トラフィックを暗号化 (HTTPS) するドメインを自分が管理していることを確認したら、アドレス (A) レコードを、ドメインを Lightsail ロードバランサーにポイントするドメインの DNS ホスティングプロバイダーに追加する必要があります。このガイドでは、Lightsail DNS ゾーン、および Amazon Route 53 ホストゾーンに A レコードを追加する方法について説明します。

DNS ゾーン - アサインメントページを使用して A レコードを追加する

1. 左側のナビゲーションペインで [ドメインと DNS] を選択します。
2. 管理する DNS ゾーンを選択します。
3. [Assignments] (割り当て) タブを選択します。
4. [Add assignment] (割り当てを追加) を選択します。
5. [Select a domain name] (ドメイン名を選択) フィールドで、ドメイン名を使用するか、ドメインのサブドメインを使用するかを選択します。
6. [Select a resource] (リソースの選択) ドロップダウンで、ドメインを割り当てるロードバランサーを選択します。
7. [割り当てる] を選択します。

変更がインターネットの DNS を通じて伝播されるまで待ちます。これには数分から数時間かかる場合があります。

DNS ゾーン - DNS レコードページを使用して A レコードを追加する

1. 左側のナビゲーションペインで [ドメインと DNS] を選択します。
2. 管理する DNS ゾーンを選択します。
3. [DNS records] (DNS レコード) タブを選択します。
4. 現在の DNS ゾーンの状態に応じて、次のいずれかの手順を実行します。

   - A レコードが追加されていない場合、レコードを追加を選択します。
   - A レコードが既に追加されている場合、ページにリストされている既存のレコードの横にある編集アイコンを選択し、ステップ 5 に進みます。

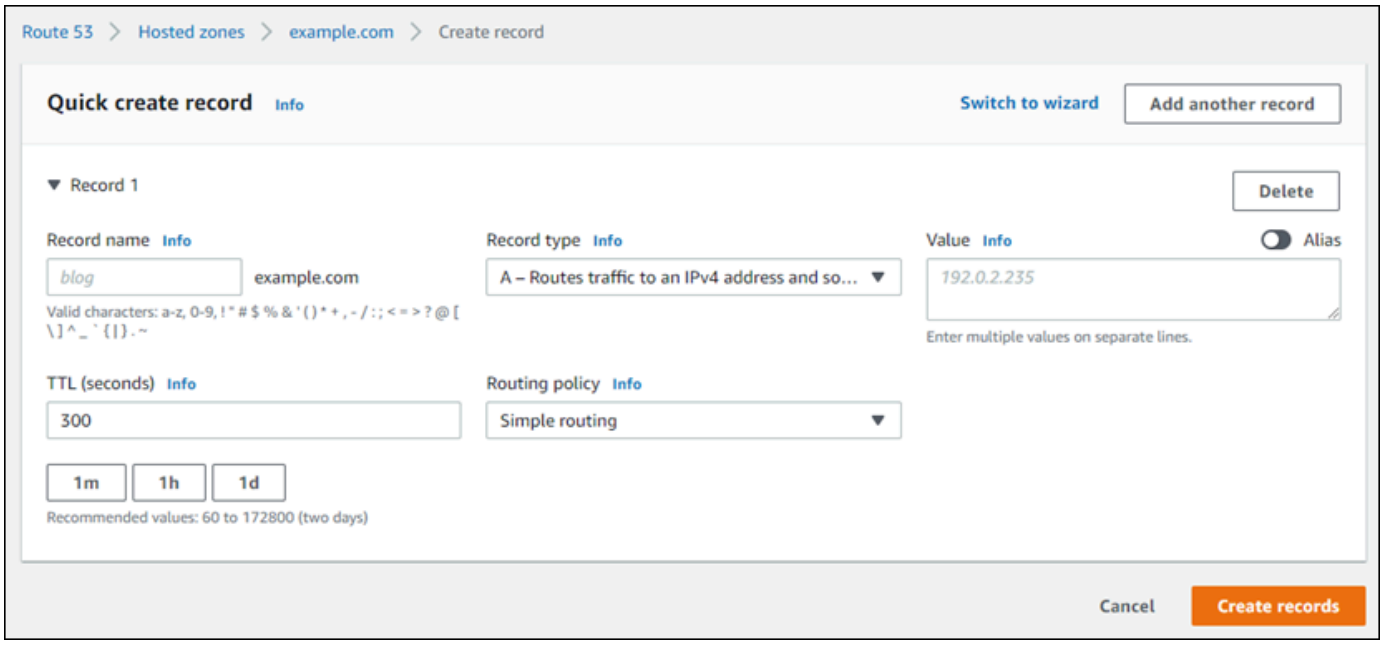
5. レコードタイプのドロップダウンメニューで A レコードを選択します。
6. [Record name] (レコード名) テキストボックスに、次のいずれかのオプションを入力します。
   - @を入力して、ドメインの頂点のトラフィックをロードバランサーにルーティングします。(例：example.com)
   - wwwを入力して、www サブドメインのトラフィックをロードバランサーにルーティングします。 (例:www.example.com)
7. 解決テキストボックスで、Lightsailロードバランサーの名前を選択します。
8. 保存アイコンを選択します。

変更がインターネットの DNS を通じて伝播されるまで待ちます。これには数分から数時間かかる場合があります。

Route 53 に A レコードを追加する

1. Route 53 コンソールにサインインします。
2. ナビゲーションペインで [Hosted zones] を選択します。
3. ロードバランサーへのトラフィックのルートに使用するドメイン名のホストゾーンを選択します。
4. [Create record] (レコードを作成) を選択します。

   「レコードのクイック作成」ページが表示されます。

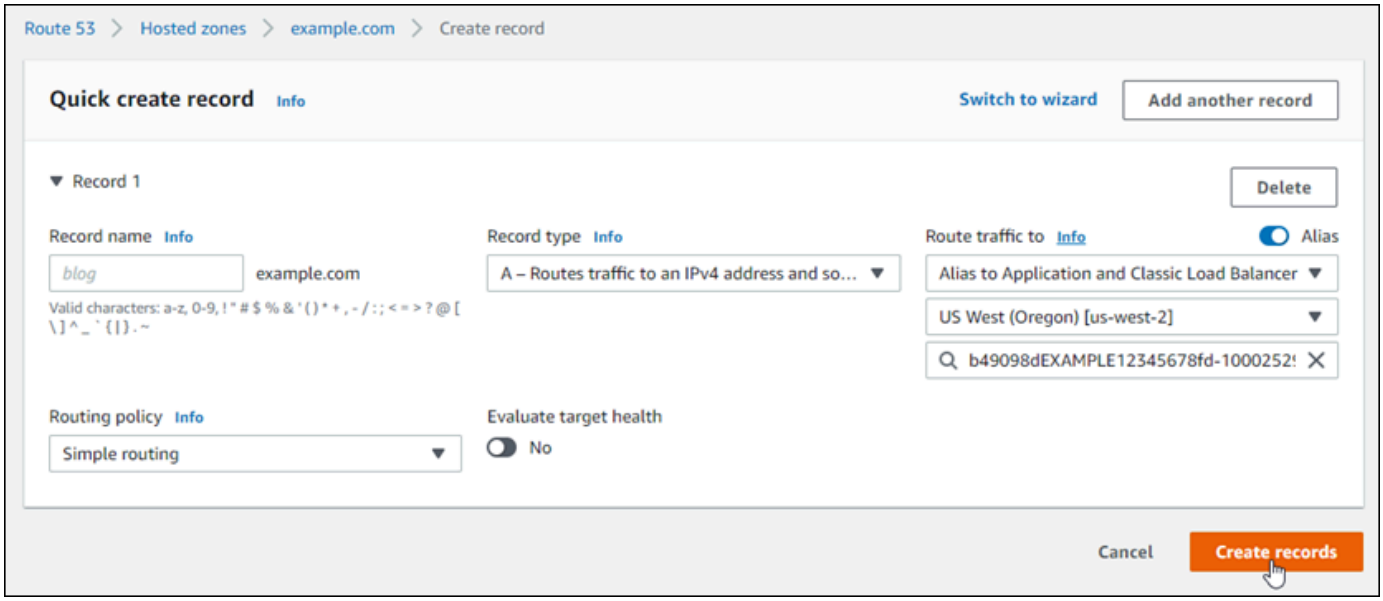


> **Note**
>
> ルーティングポリシーの選択ページが表示されている場合、クイック作成に切り替えるを選択し、次の手順を続行する前に、クイック作成ウィザードに切り替えます。

5. レコード名は、wwwサブドメイン (例:`www.example.com`) を使用する場合wwwを入力するか、ドメインの頂点を使用する場合は空白のままにします。 (例:`example.com`)
6. レコードタイプは、A – IPv4アドレスにトラフィックをルートしていくつかのAWSリソースを選択します。
7. エイリアス切り替えを選択して、エイリアスレコードを有効にします。
8. トラフィックのルーティング先は以下を選択します。
   a. エンドポイントの選択は、アプリケーションにエイリアスおよびClassic Load Balancerを選択します。
   b. リージョンの選択は、Lightsail ロードバランサーを作成したAWSリージョンを選択します。
   c. ロードバランサーの選択は、Lightsailロードバランサーのエンドポイント URL (例:DNS名) を入力、あるいはペーストします。
9. ルーティングポリシーは、シンプルルーティングを選択し、ターゲットの正常性の評価切り替えを無効化します。

   Lightsail は既にロードバランサーでヘルスチェックを実行しています。詳細については、ロードバランサーのヘルスチェック を参照してください。

   レコードは以下の例のようになります。



10. レコード作成を選択してホストゾーンにレコードを追加します。

> **Note**
>
> 変更がインターネットの DNS を通じて伝播されるまで待ちます。これには数分から数時間かかる場合があります。

Lightsail ドメインの DNS 管理を転送する

Lightsail で登録したドメインの DNS レコードは、Amazon Lightsail DNS ゾーンを使用して管理できます。または、必要に応じて、ドメインの DNS レコードの管理を別の DNS ホストプロバイダーに移管することもできます。このガイドでは、Lightsail に登録したドメインの DNS レコードの管理を別の DNS ホストプロバイダーに移管する方法を説明します。

> **Important**
>
> ドメインの DNS に対して行った変更は、インターネットの DNS を通じて伝播されるまで数時間かかる場合があります。このため、管理の移管が完了するまでは、ドメインの DNS レコードを現在の DNS ホストプロバイダーで保管しておく必要があります。これにより、転送の実行中も、ドメインのトラフィックが途切れることなくリソースにルーティングされます。

目次

- 前提条件を満たす
- DNS ゾーンにレコードを追加する

の前提条件を満たす

以下の前提条件を満たします (まだ満たしていない場合)。

1. ドメイン名を登録します。Lightsail を使用してドメイン名を登録できます。詳細については、「新しいドメインを登録する」を参照してください。
2. DNS サービスから提供されるプロセスを使用して、ドメインのネームサーバーを取得します。

DNS ゾーンにレコードを追加する

次の手順を実行して、別の DNS ホストプロバイダーのネームサーバーを Lightsail の登録ドメインに追加します。

1. Lightsail コンソール にサインインします。
2. [Domains & DNS] (ドメインと DNS) タブを選択します。
3. 他の DNS サービスを使用するように設定するドメインの名前を選択します。
4. [Edit Name Servers] (ネームサーバーを編集) を選択します。
5. ネームサーバーの名前を、前提条件を完了したときに DNS サービスから取得したネームサーバーに変更します。
6. [保存] を選択します。

Amazon Route 53 を使用してドメインを Lightsail インスタンスにポイントする

Amazon Lightsail の DNS ゾーンを使用すると、`example.com` などの登録されたドメイン名を Lightsail インスタンスで実行されているウェブサイトに簡単にポイントできます。Lightsail DNS ゾーンは最大 6 つまで作成できます。また、 DNS レコードタイプが全種類サポートされているわけではありません。Lightsail DNS ゾーンの詳細については、「DNS」を参照してください。

Lightsail DNS ゾーンが過度に制限されている場合、Amazon Route 53 ホストゾーンを使用して、ドメインの DNS レコードを管理することをお勧めします。DNS は、Route 53 を使用して、最大 500

のドメインを管理でき、非常に多様な DNS レコードタイプをサポートします。また、ドメインの DNS レコードを管理するためにすでに Route 53 を使用していて、引き続き使用することを希望する場合があるかも知れません。このガイドでは、Route 53 で管理されているドメインの DNS レコードを編集して、Lightsail インスタンスをポイントする方法を示します。

前提条件

以下の前提条件を満たします (まだ満たしていない場合)。

- Route 53 を使用してドメイン名を登録する。詳細については、「Route 53 ドキュメント」の「新しいドメインの登録」を参照してください。
- すでにドメインを登録しているけれども、そのレコードの管理に Route 53 を使用していない場合は、ドメインの DNS レコードの管理を Route 53 に転送する必要があります。詳細については、「Route 53 ドキュメント」の「Amazon Route 53 を既存ドメインの DNS サービスにする」を参照してください。
- Route 53 にドメインのパブリックホストゾーンを作成します。詳細については、「Route 53 ドキュメント」の「公開ホストゾーンの作成」を参照してください。
- 静的 IP を作成して Lightsail インスタンスにアタッチします。このガイドでは、インスタンスの静的 IP アドレス (パブリック IP アドレス) に解決される、ドメイン の Route 53 ホストゾーンで DNS レコードを作成します。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。

Route 53 を使用してドメインを Lightsail インスタンスにポイントする

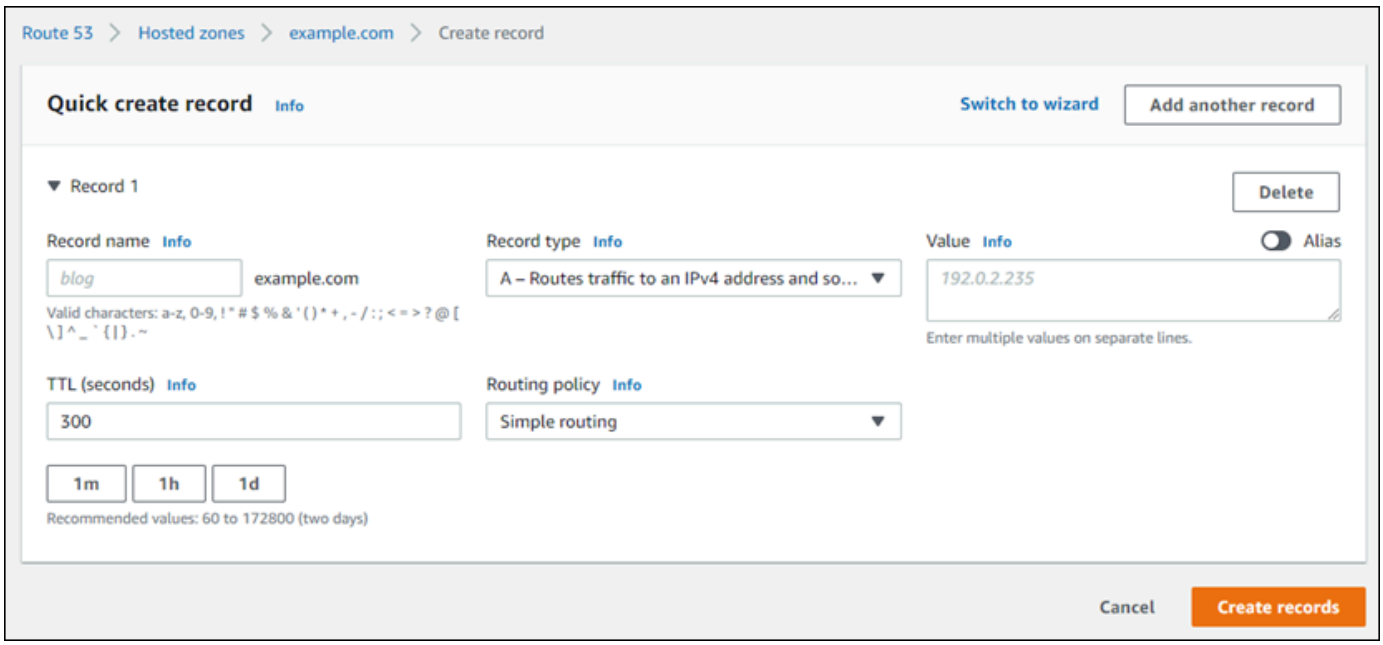
Route 53 で 2 つの代表的な DNS レコード、アドレス、および正規名を設定して、ドメインを Lightsail インスタンスにポイントするには、次の手順を実行します。

> **Note**
>
> この手順については Route 53 デベロッパーガイドでも説明しています。詳細については、「Route 53 ドキュメント」の「Amazon Route 53 コンソールを使用したレコードの作成」を参照してください。

1. Route 53 コンソールにサインインします。
2. ナビゲーションペインで [Hosted zones] を選択します。

3. ロードバランサーへのトラフィックのルートに使用するドメイン名のホストゾーンを選択します。
4. [Create record] (レコードを作成) を選択します。

   「レコードのクイック作成」ページが表示されます。



> **Note**
>
> 「ルーティングポリシーの選択」ページが表示された場合は、[クイック作成に切り替える] を選択し、クイック作成ウィザードに切り替えてから、次のステップを続行します。

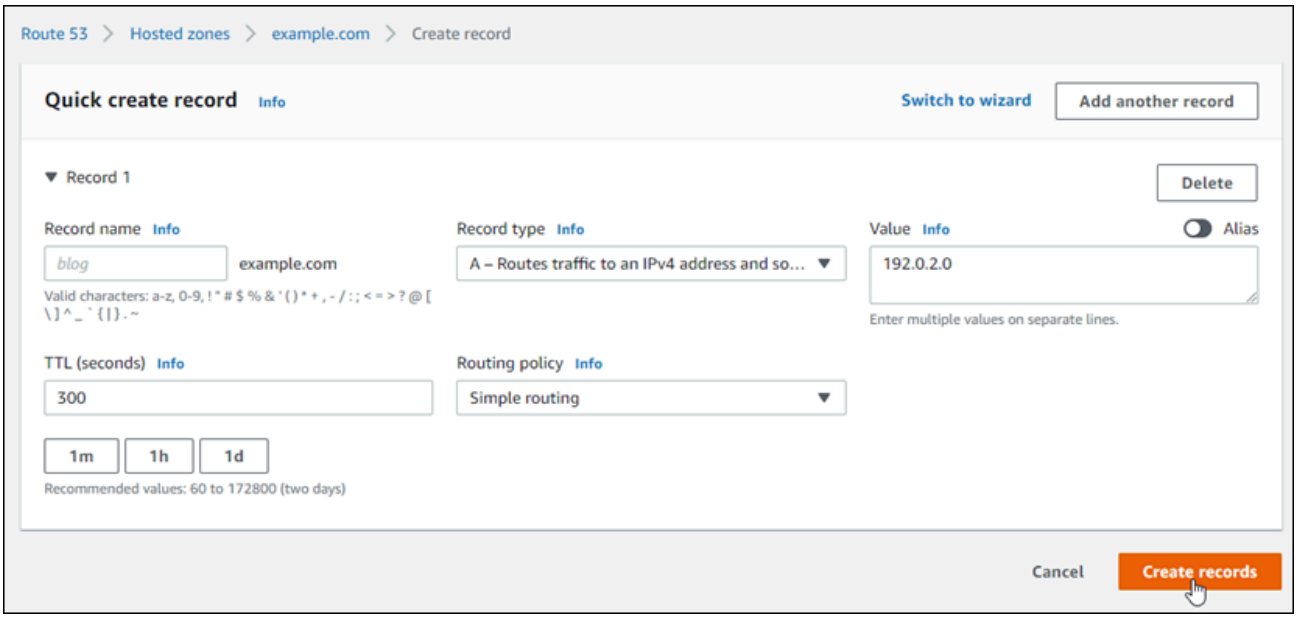
5. [レコードのタイプ] で、以下のいずれかのオプションを選択します。

   A - トラフィックを IPv4 アドレスと一部の AWS リソースにルーティング

   アドレス (A) レコードは、ドメイン (`example.com` など) やサブドメイン (`blog.example.com` など) をウェブサーバーの IP アドレス (`192.0.2.255` など) にマッピングします。

   1. [レコード名] テキストボックスを空のまま、`example.com` などのドメインの頂点が IP アドレスをポイントするようにするか、サブドメインを入力します。
   2. [レコードタイプ] ドロップダウンメニューで、[A – トラフィックをIPv4 アドレスと一部の AWS リソース にルーティング] を選択します。

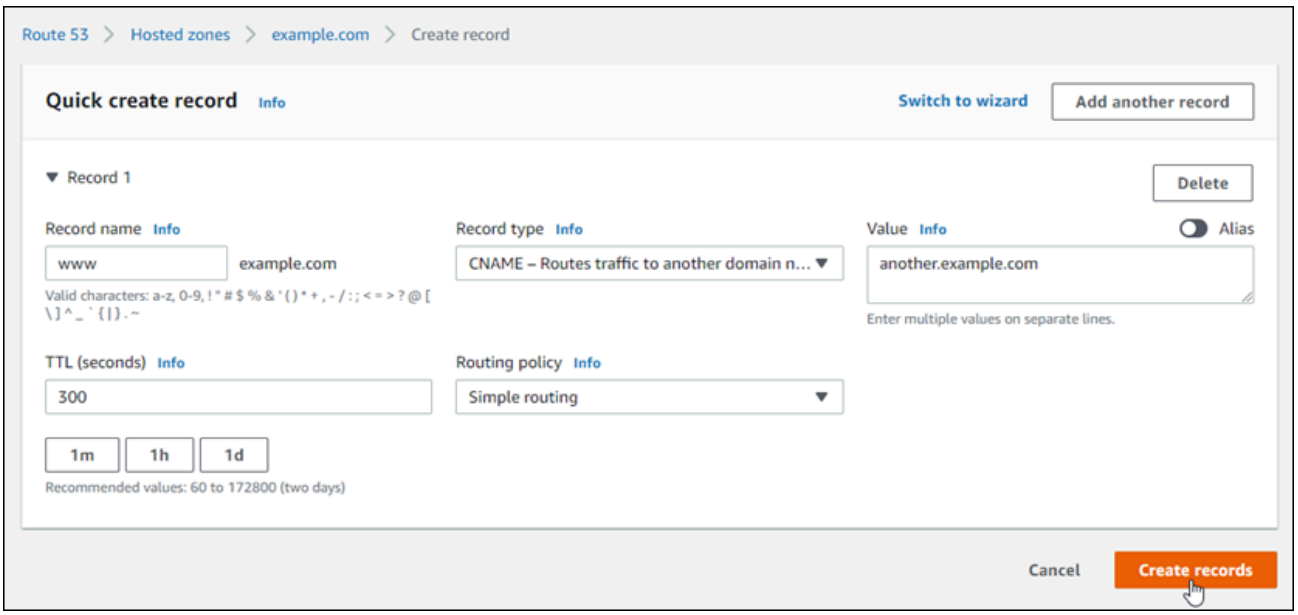
3. Lightsail インスタンスの静的 IP アドレス (パブリック IP アドレス) を [値] テキストボックスに入力します。
4. TTL を 300 に保ち、ルーティングポリシーを [シンプルルーティング] のままにしておきます。



CNAME - トラフィックを別のドメイン名および一部の AWS リソースにルーティング

正規名 (CNAME) レコードは、`www.example.com` などのエイリアスまたはサブドメインを `example.com` などのドメイン、または `www2.example.com` などのサブドメインにマップします。CNAME レコードは、あるドメインを別のドメインにリダイレクトします。

1. [レコード名] テキストボックスにサブドメインを入力します。
2. [レコードタイプ] ドロップダウンメニューで [CNAME - トラフィックを別のドメイン名および一部の AWS リソースにルーティング] を選択します。
3. [Value] (値) テキストボックスにドメイン (例：`example.com`) またはサブドメイン (例：`another.example.com`) を入力します。
4. TTL を 300 に保ち、ルーティングポリシーを [シンプルルーティング] のままにしておきます。



6. [レコード作成] を選択してホストゾーンにレコードを追加します。

> **Note**
>
> 変更がインターネットの DNS を通じて伝播されるまで待ちます。これには数分から数時間かかる場合があります。

Route 53 ホストゾーンで既存のレコードセットを編集するには、編集するレコードを選択し、変更内容を入力して、[保存] を選択します。

Lightsail のドメインの登録

Amazon Lightsail を使用して新しいドメインを登録することができます。Lightsail ドメインは、可用性とスケーラビリティが優れた DNS ウェブサービスである Amazon Route 53 を通じて登録が行われます。使用しているドメインの中に、他のプロバイダーに登録されたものがある場合は、それらのドメインの DNS 管理を Lightsail に移管できます。これらのドメインを Lightsail リソースにポイントさせることもできます。

Lightsail で新しいドメインを登録するには、以下の手順のいずれかを実行します。

- 新しいドメインの登録については、「Lightsail を使用して新しいドメインを登録する」を参照してください。

- 既存のドメインについては、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。
- ドメインの別のレジストラへの移管については、「Amazon Route 53 での Lightsail ドメインの管理」を参照してください。

ドメイン登録を開始する前に、以下の考慮事項に留意してください。

ドメイン登録の料金

ドメイン登録にかかるコストについては、「Amazon Route 53 料金表」を参照してください。

ドメインでのサービスクォータ

ユーザーが登録できるドメイン数には上限があります。詳細については、「Amazon Route 53 デベロッパーガイド」の「Service Quotas」を参照してください。上限の引き上げについては、Route 53 にお問い合わせください。

サポートされるドメイン

Lightsail では、すべての汎用最上位ドメイン (TLD) の登録をサポートしています。サポートされている TLD の一覧については、「Amazon Route 53 デベロッパーガイド」の「Amazon Route 53 に登録できるドメイン」を参照してください。

地理的最上位ドメインを登録するには Route 53 を使用する必要があります。詳細については、「Amazon Route 53 デベロッパーガイド」の「地理的最上位ドメイン」を参照してください。

登録の完了後、ドメイン名を変更することはできません。

誤って正しくないドメイン名を登録した場合でも、そのドメイン名を変更することはできません。代わりに、正しい名前を指定しながら、新たにドメイン名を登録する必要があります。誤って登録したドメイン名について、料金の払い戻しはありません。

DNS ゾーンの料金

Lightsail でドメインを登録すると、そのドメインの DNS ゾーンが自動的に作成されます。Lightsail では DNS ゾーンに料金はかかりません。

Lightsail を使用して新しいドメインを登録する

トピック

- 新しいドメインを登録する前提条件

- 新しいドメインを登録する
- ドメインの連絡先情報を検証する

新しいドメインを登録する前提条件

ドメインの必要な DNS レコードタイプが Lightsail の DNS ゾーンでサポートされていることを確認します。Lightsail DNS ゾーンでは現在、アドレス (A)、正規名 (CNAME)、メールエクスチェンジャ (MX)、ネームサーバー (NS)、サービスロケーター (SRV)、およびテキスト (TXT) のレコードタイプがサポートされています。NS レコードには、ワイルドカード DNS レコードエントリを使用できます。

ドメインに必要な DNS レコードタイプが Lightsail の DNS ゾーンでサポートされていない場合は、ドメインの DNS ホスティングプロバイダとして、Route 53 を使用するとよいでしょう。Route 53 はより多くのレコードタイプをサポートします。詳細については、「Amazon Route 53 デベロッパーガイド」の「サポートされる DNS レコードタイプ」と「Amazon Route 53 を既存ドメインの DNS サービスとして使用する」を参照してください。

新しいドメインを登録する

新しいドメインを登録するには

1. Lightsail コンソール にサインインします。
2. [Domains & DNS] (ドメインと DNS) タブを選択します。
3. [Register Domain] (ドメインの登録) を選択し、登録するドメインを指定します。
   a. 登録するドメイン名を入力し、[Check availability] (使用可能かチェック) を選択してそのドメイン名が使用できるかどうか確認します。そのドメインが使用可能な場合は、Automatic domain renewal (ドメインの自動更新) をそのまま続けます。
   b. ドメイン名が使用できない場合には、最初の選択肢の代わりとして、あるいはそれに加えて登録できる、他のドメインが一覧表示されます。登録に使用するドメインで、[Select] (選択) を選択します。
4. 有効期限の前に、ドメイン登録の自動更新を行うかどうかを選択します。自分のために登録したドメイン名は、デフォルトで 1 年間その所有権を維持できます。ドメイン名登録の更新を行わないと、有効期限が切れた後、そのドメイン名を他の誰かが登録に使用できるようになります。ドメイン名を確実に維持するためには、毎年自動更新を行うように設定するか、より長い所有期間を設定します。
5. [Domain contact information] (ドメインの連絡先情報) セクションで、ドメインの登録者、管理者、技術担当者の連絡先情報を入力します。詳細については、「Values that you specify when

you register or transfer a domain」(ドメインを登録または移管するときに指定する値) を参照してください。

以下の考慮事項に注意してください。

姓と名前

[First Name] (名) と [Last Name] (姓) には、ご自身の本名と同じ名前を指定することをお勧めします。ドメイン設定の変更に際しては、一部のドメインレジストリで、身分証明書の提供が求められる場合があります。お客様の ID の名前は、ドメイン登録者の連絡先の名前と完全に一致する必要があります。

他の連絡先

デフォルトでは、3 種類の連絡先すべてについて同じ情報が使用されます。連絡先として 1 つ以上の異なる情報を入力する場合は、[Same as registrant] (登録者と同じ) チェックボックスをオフにした後に、新しい連絡先情報を入力します。

6. [Privacy Protection] (プライバシー保護) セクションで、WHOIS クエリに対し連絡先情報を隠蔽するかどうかを選択します。

   詳細については、以下の各トピックを参照してください。

   - プライバシー保護
   - Amazon Route 53 に登録できるドメイン

7. [Register domain] (ドメインの登録) を選択して続行します。[DNS zones] (DNSゾーン) と [Summary] (概要) セクションに、ドメインのDNSゾーン、料金、および更新のスケジュールに関する情報が表示されます。
8. ドメインを登録する前に、「Amazon Route 53 ドメイン名登録規約」に同意する必要があります。

ドメインの連絡先情報を検証する

ドメイン登録の完了後は、登録者の連絡先として有効な E メールアドレスが指定されていることを確認する必要があります。

以下のいずれかの E メールアドレスを使用して、自動的に確認の E メールが送信されます。

- noreply@registrar.amazon - Amazon Registrar がレジストラになっているドメインの場合。

- noreply@domainnameverification.net - 当社のレジストラアソシエイトである Gandi がレジストラになっているドメインの場合。自分の TLD のレジストラを特定するには、「Amazon Route 53 デベロッパーガイド」の「Amazon Route 53 に登録できるドメイン」を参照してください。

以下の手順により、ドメイン検証のプロセスを完了します。

ドメインの検証を完了するには

1. 確認 E メールを受け取ったら、E メール内のリンクを選択し、指定した E メールアドレスが有効であることを確認します。E メールがすぐに届かない場合は、迷惑メールフォルダーを確認します。
2. Lightsail コンソールに戻ります。ステータスが自動的に [Verified] (検証済み) に更新されない場合は、[Refresh status] (ステータスの更新) を選択します。

> **Important**
>
> 連絡先となっている登録者は、E メールの指示に従って、そのメールの受信を確認する必要があります。これを行わない場合、対象のドメインは ICANN の規定に従い停止されます。ドメインが停止されると、インターネットでそのドメインにアクセスできなくなります。

3. ドメイン登録が完了したら、DNS サービスとして Lightsail を使用するか、他の DNS サービスを使用するかを選択します。
   - Lightsail

     ドメインおよびサブドメインへのトラフィックのルーティング方法を Lightsail に指示するためのレコードを、ドメイン登録時に Lightsail が作成した DNS ゾーン内に作成します。

     例えば、誰かがブラウザにドメイン名を入力し、そのクエリが Lightsail に転送された際、Lightsail がそのクエリに対して IP アドレスで応答するのか、ロードバランサーの名前で応答するのかを指定します。詳細については、「DNS ゾーンの編集または削除」を参照してください。
   - 別の DNS サービスの使用

     Lightsail 以外の DNS サービスに DNS クエリをルーティングするように新しいドメインを設定します。詳細については、「他の DNS サービスを使用するときにドメインのネームサーバーを更新するには」を参照してください。

Amazon Registrar に登録されているドメインの登録の詳細を表示する

Amazon Lightsail と Amazon Route 53 を使用して登録した .com、.net、および .org ドメインであり、かつ Amazon Registrar がレジストラであるドメインに関する情報を表示できます。これには、ドメインの初回登録時に提供した情報、ドメイン所有者、技術担当者、管理者の連絡先情報が含まれます。

次の点に注意してください。

プライバシー保護がアクティブになっている場合にドメインの連絡先にメールを送信

ドメインに対してプライバシー保護がアクティブになっている場合、登録者、技術担当者、管理者の連絡先情報は、Amazon Registrar プライバシーサービスの連絡先情報に置き換えられます。例えば、example.com ドメインが Amazon Registrar に登録されており、プライバシー保護がアクティブになっている場合、WHOIS クエリに対するレスポンスでは、[Registrant Email] (登録者 E メール) の値は `owner1234@example.com.whoisprivacyservice.org` のようになります。

プライバシー保護がアクティブになっている場合にドメインの連絡先に問い合わせるには、対応するメールアドレスにメールを送信します。E メールは該当する連絡先に自動的に転送されます。

不正使用を報告

不適切なコンテンツ、フィッシング、マルウェア、スパムなど、違法行為または適正使用ポリシー の違反を報告するには、trustandsafety@support.aws.com に E メールを送信してください。

Amazon Registrar に登録されているドメインに関する情報を表示するには

1. ウェブブラウザで、以下のウェブサイトのいずれかに移動します。両方のウェブサイトには同じ情報が表示されます。ただし、使用されるプロトコル、情報の表示形式は異なります。
   - WHOIS: https://registrar.amazon.com/whois
   - RDAP: https://registrar.amazon.com/rdap
2. 情報を表示するドメインの名前を入力し、[Search (検索)] を選択します。検索したドメインが Amazon Lightsail または Route 53 を使用して登録されていない場合は、そのドメインがレジストラデータベースにないことを示すメッセージが表示されます。

Lightsail のドメイン名をフォーマットする

ユーザーがウェブサイトやアプリケーションにアクセスしやすいように、覚えやすいドメイン名を選択します。ドメイン名 (および DNS ゾーンの名前、レコード名) は、ピリオド (.) で区切られた一連のラベルから構成されます。命名要件は、ドメイン名を登録するのか、DNS ゾーンまたはレコードの名前を指定するのかによって異なります。

ドメイン名は、次のガイドラインに従ってフォーマットします。

目次

- ドメイン名登録用のドメイン名をフォーマットする
- DNS ゾーンとレコード用のドメイン名をフォーマットする
- DNS ゾーンとレコードの名前でアスタリスク (*) を使用する
- 次のステップ

ドメイン名登録用のドメイン名をフォーマットする

ドメイン名登録では、ドメイン名は 1～255 文字でなければなりません。ドメイン名に使用できるのは、a - z、A - Z、0～9、ハイフン (-)、ピリオド (.) です。

ドメイン名の先頭または末尾にスペースとハイフンを使用することはできません。Lightsail は有効な汎用最上位ドメイン (TLD) 名をすべてサポートします。詳細については、Amazon Route 53 デベロッパーガイドの「汎用最上位ドメイン」を参照してください。

DNS ゾーンとレコード用のドメイン名をフォーマットする

DNS ゾーンとレコードの場合、ドメイン名は 1～255 文字でなければなりません。ドメイン名に使用できるのは、a - z、A - Z、0～9、ハイフン (-)、ピリオド (.) です。スペースは使用できません。

Lightsail では、大文字 (A - Z) で指定しても、英字は小文字 (a - z) として格納されます。

Lightsail は、汎用 TLD と地理的 TLD の両方の DNS ゾーンをサポートします。地理的 TLD のその他の例については、Amazon Route 53 デベロッパーガイドの「地理的最上位ドメイン」を参照してください。

DNS ゾーンとレコードの名前でのアスタリスク (*) の使用

DNS では、名前の中の位置によっては、アスタリスク (*) 文字はワイルドカード文字と見なされます。ワイルドカード DNS レコードとは、未定義のサブドメインの DNS リクエストに応答するレ

コードです。Lightsail では、次の条件で、名前にアスタリスク (*) を含む DNS ゾーンとレコードを作成できます。

DNS ゾーン

- アスタリスク (*) をドメイン名の左端のラベルに含めることはできません。例えば、「*.example.com」は使用できません。
- アスタリスク (*) が他の位置に含まれる場合、DNS はこれをワイルドカードとしてではなく、ASCII 42 文字として扱います。ASCII 文字の詳細については、「ウィキペディア」の「ASCII」を参照してください。

DNS レコード

DNS レコード名の中でアスタリスク (*) をワイルドカードとして使用する際は、次の制約に注意してください。

- ワイルドカードとして、アスタリスクはドメイン名の左端のラベルを置き換えるものである必要があります。例えば、*.example.com、*.acme.example.com などです。prod.*.example.com のようにアスタリスクを他の位置に含めると、DNS はこれをワイルドカードとしてではなく、ASCII 42 文字として扱います。
- アスタリスクで、ラベル全体を置き換える必要があります。例えば、*prod.example.com や prod*.example.com と指定することはできません。
- 特定のドメイン名が優先されます。例えば、*.example.com と acme.example.com のレコードを作成すると、acme.example.com の DNS クエリは、acme.example.com レコードの値で応答します。
- アスタリスクは、アスタリスクが含まれたサブドメインレベル、およびそのサブドメインのすべてのサブドメインの DNS クエリに適用されます。例えば、*.example.com という名前のレコードを作成すると、*.example.com の DNS クエリは次の名前に応答します。

  zenith.example.com

  acme.zenith.example.com

  pinnacle.acme.zenith.example.com (その DNS ゾーンにどのタイプのレコードもない場合)

「*.example.com」という名前のレコードを作成して「example.com」レコードがない場合、Lightsail は「example.com」の DNS クエリに `NXDOMAIN` (存在しないドメイン) で応答します。

同じレベルのすべてのサブドメインの DNS クエリに対して、さらにドメイン名の DNS クエリに対しても、同じレスポンスを返すように Lightsail を設定できます。例えば、Lightsail が「example.com」レコードを使用して、「acme.example.com」や「zenith.example.com」などの DNS クエリに応答するように設定できます。サブドメインのトラフィックを example.com の最上位ドメインにルーティングするには、次の手順を実行します。

1. ドメインのレコードを作成します (example.com など)。
2. サブドメインのエイリアスレコードを作成します (*.example.com など)。前のステップで作成したレコードの名前を、エイリアスレコードのターゲットとして指定します。

次の手順

詳細については、以下の各トピックを参照してください。

- ドメインの DNS レコードを管理する DNS ゾーンを作成する
- DNS

上級 Route 53 機能を使用して Lightsail ドメインを管理する

Amazon Lightsail では、可用性が高くスケーラブルな DNS ウェブサービスである Amazon Route 53 を通じてドメインを登録します。Lightsail を使用してドメインを登録することで、Lightsail と Route 53 の両方でドメインの管理が行えます。

ドメインの登録や、ドメインのトラフィックを Lightsail リソースにルーティングするなどのタスクは、Lightsail コンソールで実行します。詳細については、「Amazon Lightsail でのドメイン登録」を参照してください。

ドメインの移管や登録の削除などの高度なタスクは、Amazon Route 53 コンソールで実行する必要があります。

このガイドでは、Route 53 コンソールを使用して完了できる高度な管理タスクの一部について説明します。Route 53 の概要については、「Amazon Route 53 デベロッパーガイド」の「Amazon Route 53 とは」を参照してください。

目次

- ドメイン登録のステータスを表示する

- 別の登録への許可のない移管を防ぐためにドメインをロックする
- 失効した、または削除されたドメインを復元する
- ドメインを移管する
- ドメイン名の登録を削除する

ドメイン登録のステータスを表示する

ドメイン名には、拡張プロビジョニングプロトコル (EPP) ステータスコードとも呼ばれるステータスがあります。EPP ステータスコードは、ICANN (ドメイン名に関する中心的なデータベースを管理する組織) により開発されました。EPP ステータスコードは、各種オペレーションに関するステータスを表します。これらのステータスの例としては、ドメイン名の登録、ドメイン名の登録の更新などに関するものが挙げられます。すべてのレジストラは、この同じステータスコードを使用します。ドメインのステータスコードを確認するには、「Amazon Route 53 デベロッパーガイド」の「ドメイン登録のステータスの表示」を参照してください。

別の登録への許可のない移管を防ぐためにドメインをロックする

すべての汎用最上位ドメイン (TLD) のドメインレジストリでは、許可なく他者がドメインを別のレジストラに移管することを防止するために、ユーザーが自分のドメインをロックする手段を提供しています。詳細については、「Amazon Route 53 デベロッパーガイド」の「別の登録への許可のない移管を防ぐためにドメインをロックする」を参照してください。

失効した、または削除されたドメインを復元する

後期更新期間が終了する前にドメインを更新しないか、ドメインを誤って削除した場合、最上位ドメイン (TLD) のいくつかのレジストリにより、他のユーザーが登録できるようになる前に、ドメインを復元することができます。ドメイン登録の復元を試すには、以下からリンク先にある手順を使用してください。詳細については、「Amazon Route 53 デベロッパーガイド」の「失効した、または削除されたドメインの復元」を参照してください。

ドメイン登録を移管する

別の登録から Amazon Route 53 に、 AWS アカウントから別のアカウントに、または Route 53 から別の登録に、ドメインの登録を移管できます。詳細については、「Amazon Route 53 デベロッパーガイド」の「ドメインを移管する」を参照してください。

ドメイン名の登録を削除する

最上位ドメイン (TLD) では、必要がなくなった登録を削除できます。レジストリで登録を削除できる場合、このトピックの手順を実行します。詳細については、Amazon Route 53 デベロッパーガイドの「ドメイン名の登録を削除する」を参照してください。

Lightsail でドメインを登録または移管するときのドメイン情報を提供する

Amazon Lightsail を使用してドメインを登録すると、登録期間 (契約期間) やドメインの連絡先情報などのドメイン情報が提供されます。また、ドメインの自動更新とプライバシー保護も設定します。

Lightsail に現在登録されているドメインの情報を変更することもできます。

> **Note**
>
> - ドメインの連絡先情報を変更した場合は、登録者の連絡先に変更について通知メールが送信されます。この E メールは noreply@registrar.amazon から送信されます。ほとんどの変更について、登録者は応答する必要はありません。
> - 連絡先情報の変更が所有権の変更も含む場合は、登録者の連絡先に追加のメールが送信されます。ドメイン名の中央データベースを管理する組織である ICANN の規則では、メールを受け取ったことについて登録者の連絡先による確認が必要です。詳細については、このセクションの後半にある姓名および組織の項目を参照してください。

既存のドメインの連絡先情報の変更については、「ドメインの連絡先情報を更新する」を参照してください。

お客様が提供するドメイン情報

- 言葉
- ドメインの自動更新
- 登録者、管理者、テクニカル、請求先の連絡先
- コンタクトのタイプ
- 姓名
- 組織

- メール
- 電話
- 住所 1
- 住所 2
- 国
- 都道府県
- 市町村
- 郵便番号
- プライバシー保護

言葉

ドメインの登録期間。通常、期間は 1 年ですが、ドメインの登録時に最大 10 年まで延長できます。

ドメインの自動更新

ドメインを Lightsail に登録すると、ドメインが自動的に更新されるようになります。自動更新期間は通常 1 年間です。有効期限切れになる前に Lightsail で自動的にドメイン登録を更新するかどうかを選択します。登録料金は AWS アカウントに請求されます。詳細については、「ドメイン登録の更新」を参照してください。

> **Important**
>
> ドメイン自動更新を非アクティブにした場合、有効期限が過ぎるとドメイン登録は更新されません。その結果、ドメイン名のコントロールを失う可能性があります。

登録者、管理者、テクニカル、請求先の連絡先

ドメインを登録する際、次の連絡先が必要です。

- 登録者 – ドメインの所有者
- 管理者 – ドメインの管理を担当する連絡先。
- テクニカル – ドメインの技術的な変更を担当する連絡先。
- 請求 – ドメインに関する請求の問い合わせを担当する連絡先。

> **Note**
>
> デフォルトでは、登録者に指定した情報を使用して他の連絡先に適用します。ある連絡先の異なる情報を入力するには、[登録者と同じ] の選択を解除します。

コンタクトのタイプ

この連絡先のカテゴリ。

> **Note**
>
> - [Company] (会社) または [Association] (協会) オプションを選択した場合は、組織名を入力する必要があります。
> - 一部の最上位ドメイン (TLD) の場合、使用可能なプライバシー保護は、[Contact type] (連絡先のタイプ) で選択した値によって異なります。TLD のプライバシー保護設定については、「Amazon Route 53 に登録できるドメイン」を参照してください

姓名

連絡先の姓名。[First name] (名) と [Last name] には、公的な身分証明書の名前を使用することをお勧めします。一部のドメイン設定の変更については、身分証明書の提示が必要です。その場合、身分証明書の名前がドメイン登録者の連絡先の名前と一致する必要があります。

登録者の連絡先メールアドレスを変更した場合、このメールは以前のメールアドレスと新しいメールアドレスの両方に送信されます。

組織

連絡先と関連付けられている組織 (存在する場合)。登録者と管理者の連絡先の場合、これは通常、ドメインを登録する組織です。技術担当者の連絡先の場合、これはドメインを管理する組織のこともあります。

連絡先のタイプが [Person] (個人) 以外のときに、登録者の連絡先の [Organization] (組織) フィールドを変更すると、ドメインの所有者が変更されます。ICANN の規則では、登録者の連絡先にメールを送付して承認を得る必要があります。メールは次のメールアドレスの 1 つから送信されます。

- noreply@registrar.amazon - Amazon Registrar がレジストラになっているドメインの場合。
- noreply@domainnameverification.net - 当社のレジストラアソシエイトである Gandi がレジストラになっているドメインの場合。

お客様の TLD のレジストラを特定するには、「Amazon Route 53 に登録できるドメイン」を参照してください。

登録者の連絡先 E メールアドレスを変更した場合、この E メールは以前の E メールアドレスと新しい E メールアドレスの両方に送信されます。

メール

連絡先のメールアドレス。

> **Note**
>
> 登録者の連絡先のメールアドレスを変更すると、以前のメールアドレスと新しいメールアドレスの両方に通知メールが送信されます。この E メールは noreply@registrar.amazon から送信されます。

電話

連絡先の電話番号です。

- 米国またはカナダの電話番号を入力する場合は、「1」の後に市外局番を含む 10 桁の電話番号を入力します。
- その他の場所の電話番号を入力する場合は、国コードの後に残りの電話番号を入力します。国の電話コードのリストについては、「ウィキペディア」の「国際電話番号の一覧」を参照してください。

住所 1

連絡先の住所または私書箱。

住所 2

アパート、スイート、ユニット、ビル、フロア、配達先コードなど、連絡先の追加住所情報。

国

連絡先の国。

都道府県

連絡先の都道府県。

市町村

連絡先の市町村。

郵便番号

連絡先の郵便番号。

プライバシー保護

WHOIS クエリに対して連絡先情報を隠すかどうかを選択します。ドメインの連絡先情報のプライバシー保護をアクティブにすると、WHOIS (「who is」) クエリは、個人情報の代わりにドメインレジストラの連絡先情報を返します。ドメインレジストラは、ドメイン名登録を管理する会社です。

> **Note**
>
> 管理者、登録者、および技術担当者の連絡先に同じプライバシー設定が適用されます。

ドメインの連絡先情報のプライバシー保護を無効にした場合、指定したメールアドレスに送られてくる迷惑メールが増えます。

だれでもドメインの WHOIS クエリを送信して、そのドメインのすべての連絡先情報を取得することができます。WHOIS コマンドは多くのオペレーティングシステムで利用でき、多くのウェブサイトでウェブアプリケーションとしても利用できます。

> **Important**
>
> ドメイン連絡先情報の正当なユーザーもいますが、最も一般的なユーザーは、迷惑メールや詐欺メールをドメインの連絡先に送りつけるスパム業者です。一般的に、[Contact information] (連絡先情報) については [Privacy protection] (プライバシー保護) を有効のままにしておくことをお勧めします。

プライバシー保護の詳細については、以下のトピックを参照してください。

- ドメインのプライバシー保護を管理する
- Amazon Route 53 に登録できるドメイン

Lightsail でのドメイン登録の更新または無効化

Amazon Lightsail にドメインを登録すると、ドメインがデフォルトで自動的に更新されるようになります。デフォルトの自動更新期間は 1 年間ですが、一部の最上位ドメイン (TLD) のレジストリの更新期間はこれより長くなっています。すべての汎用 TLD では、ドメイン登録期間を、通常 1 年単位で最大 10 年まで延長することができます。

> **Note**
>
> を閉じる場合は、必ず自動更新を非アクティブ化してください AWS アカウント。そうしないと、アカウントを閉鎖した後でもドメイン登録が更新されます。

目次

- 自動更新
- ドメイン登録中のドメイン自動更新の設定
- 登録済みのドメイン自動更新の設定

自動更新

自動更新がアクティブな場合のタイムラインは次のとおりです。

有効期限の 45 日前

当社から登録者の連絡先に E メールを送信し、自動更新がアクティブになっていることを伝えます。E メールには、自動更新を非アクティブにする方法についての説明も記載されています。登録者の連絡先メールアドレスを最新状態にして、メールが届くようにしておきます。

有効期限の 35 日前または 30 日前

.com.ar、.com.br、.jp ドメインを除くすべてドメインでは、有効期限の 35 日前にドメイン登録の更新が行われます。これにより、ドメイン名の有効期限が切れる前に、更新に関する問題を解決する時間を確保できます。

.com.ar、.com.br、.jp のドメインのレジストリでは、有効期限の 30 日前にならないとドメインの更新ができません。当社のレジストラアソシエイトである Gandi から、有効期限の 30 日前に更新についての E メールが送信されます。自動更新がアクティブな場合、この E メールはドメインを更新したのと同じ日に送信されます。

自動更新が非アクティブの場合、ドメイン名の有効期限が近づいたときのタイムラインは次のとおりです。

有効期限の 45 日前

登録者の連絡先に E メールを送信し、自動更新が現在非アクティブになっていることを伝えます。E メールには、自動更新をアクティブにする方法についての説明も記載されています。登録者の連絡先メールアドレスを最新状態にして、メールが届くようにしておきます。

有効期限切れの 35 日前 または 7 日前

ドメインの自動更新が非アクティブの場合、ドメイン登録の運営組織である ICANN は、レジストラが登録者の連絡先に E メールを送信することを義務付けています。メールは次のメールアドレスの 1 つから送信されます。

noreply@registrar.amazon - Amazon Registrar がレジストラになっているドメインの場合。

noreply@domainnameverification.net - 当社のレジストラアソシエイトである Gandi がレジストラになっているドメインの場合。

有効期限まで 30 日未満の期間に自動更新をアクティブにすると、ドメインは 24 時間以内に更新されます。

更新期間の詳細については、Amazon Route 53 デベロッパーガイドの「Amazon Route 53 に登録できるドメイン」の中の該当する TLD についての「ドメインの更新と復元の期限」セクションを参照してください。

有効期限経過後

ほとんどのドメインは有効期限が切れても短期間保持されるため、有効期限後に失効したドメインを更新できる場合もありますが、ドメインの維持を希望する場合は、自動更新をアクティブにしておくことを強くお勧めします。有効期限後にドメインを更新しようとする場合についての詳細は、Amazon Route 53 デベロッパーガイドの「失効した、または削除されたドメインを復元する」を参照してください。

ドメインの有効期限が切れたが、ドメインで後期更新が許可されている場合は、標準更新価格でドメインを更新できます。ドメインがまだ期限切れ後の更新期間内であるかどうかを確認するには、Amazon Route 53 デベロッパーガイドの「ドメインの登録期間を延長する」に記載されている手順を実行します。ドメインがまだリストされる場合は、後期更新期間内です。

ドメイン登録中のドメイン自動更新の設定

新しいドメイン名を Lightsail に登録すると、ドメインが自動的に更新されるようになります。ドメイン登録手続き中に、自動ドメイン更新を非アクティブにすることもできます。

1. Lightsail コンソール にサインインします。
2. [Domains & DNS] (ドメインと DNS) タブを選択します。
3. [Register domain] (ドメインの登録) ボタンを選択します。
4. Lightsail に登録するドメイン名を指定し、[空き状況の確認] を選択します。
5. ドメイン名が利用可能な場合は、ドメイン登録ページが表示されます。[Automatic domain renewal] (自動ドメイン更新) セクションで、トグルスイッチをオンまたはオフにして、自動ドメイン更新をアクティブまたは非アクティブにします。

登録済みのドメイン自動更新の設定

Lightsail により、有効期限の少し前にドメインの登録を自動更新するかどうかの変更を行う場合、または自動更新の現在の設定を表示する場合には、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. [Domains & DNS] (ドメインと DNS) タブを選択します。
3. 表示または更新するドメインを選択します。
4. [Contact info] (連絡先情報) タブを選択します
5. 5. [Automatic domain renewal] (ドメインの自動更新) セクションで、トグルスイッチをオンまたはオフにして、ドメインの登録期間中の自動更新をアクティブまたは非アクティブにします。

Lightsail のドメイン連絡先のプライバシー保護を管理する

ドメインを Amazon Lightsail に登録すると、ドメインのすべての連絡先について、デフォルトでプライバシー保護がアクティブになります。これによって一般的に、WHOIS ("Who is") クエリから返される連絡先情報の大部分が非表示になり、送られてくるスパムの数が減少します。お客様の連絡先

情報は、レジストラの連絡先情報または "REDACTED FOR PRIVACY" という文言に置き換えられます。プライバシー保護の利用には料金はかかりません。

プライバシー保護を非アクティブにすると、誰でもドメインに関する WHOIS クエリを送信でき、ほとんどの最上位ドメイン (TLD) について、ドメインの登録時に指定したすべての連絡先情報を取得できる可能性があります。この情報には、名前、住所、電話番号、E メールアドレスが含まれています。WHOIS コマンドは広く利用可能です。このコマンドは、多くのオペレーティングシステムに含まれ、多くのウェブサイトでウェブアプリケーションとしても利用できます。

Lightsail を使用して登録したドメインのプライバシー保護を管理するには、次の手順を実行します。

目次

- 前提条件を満たす
- ドメインのプライバシー保護を管理する

の前提条件を満たす

ドメインを Lightsail に登録します。詳細については、「新しいドメインを登録する」を参照してください。

ドメインのプライバシー保護を管理する

1. Lightsail コンソール にサインインします。
2. [Domains & DNS] (ドメインと DNS) タブを選択します。
3. プライバシー保護を変更するドメインの名前を選択します。
4. [Contact info] (連絡先情報) を選択します。
5. [Privacy protection] (プライバシー保護) トグルスイッチをオンまたはオフにすることで、連絡先情報のプライバシー保護を管理できます。

Lightsail のドメイン連絡先情報の更新

Amazon Lightsail でドメインを登録すると、ドメインの連絡先情報を指定する必要があります。ドメインの連絡先情報は、ドメインの所有権を確認し、ドメイン名に関連する情報を最新の状態に保つために使用されます。ドメイン登録に必要な情報の詳細については、「Lightsail でドメインを登録または移管するときのドメイン情報を提供する」を参照してください。

トピック

- ドメインの所有者は誰ですか?
- ドメインの連絡先情報を更新

ドメインの所有者は誰ですか。

連絡先のタイプが [Person] で、登録者の連絡先の [First Name] または [Last Name] フィールドを変更すると、ドメインの所有者を変更したことになります。

連絡先のタイプが [Person] 以外のときに [Organization] を変更すると、ドメインの所有者が変更されます。

Lightsail に現在登録されているドメインの連絡先情報を変更すると、次のアクションが発生します。

- ドメインの連絡先情報を変更した場合は、登録者の連絡先に変更について通知メールが送信されます。この E メールは noreply@registrar.amazon から送信されます。ほとんどの変更について、登録者は応答する必要はありません。
- 連絡先情報の変更が所有権の変更も含む場合は、登録者の連絡先に追加のメールが送信されます。ドメイン名の中央データベースを管理する組織である ICANN の規則では、メールを受け取ったことについて登録者の連絡先による確認が必要です。

ドメインの連絡先情報を更新

ドメインの連絡先情報を更新するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. [Domains & DNS] (ドメインと DNS) タブを選択します。
3. 更新するドメインの名前を選択します。
4. [Contact info] (連絡先情報) タブを選択します。次に、[Edit contact] (連絡先を編集) を選択します。
5. 目的の値を更新します。詳細については、「Amazon Route 53 デベロッパーガイド」の「ドメインを登録または移管するときに指定する値」を参照してください。
6. [保存] を選択します。

Lightsail でのリレーショナルデータベースの作成と管理

Amazon Lightsail の MySQL または PostgreSQL マネージドデータベースは、わずかなステップ数で作成できます。Lightsail は、一般的なメンテナンスやセキュリティタスクを管理することによって、データベースの管理を効率化します。Lightsail コンソールでは、以下のことができます。

- データベースをスナップショットにバックアップする。
- スナップショットから新しいより大きなデータベースを作成する。
- ブラウザベースのログやメトリクスを使用して、一般的な問題のトラブルシューティングを行います。
- 特定の時点のバックアップおよび復元オペレーションを使用してデータを復元する。

Lightsail インスタンスでアプリケーションを構築して、Lightsail マネージドデータベースに接続できます。スタンドアロンデータベースを作成して、貴社の分析ツールやクエリツールを接続することもできます。スタンダートデータベースプランと高可用性データベースプランから選択できます。これらのプランでは、事前設定されたデータベース、SSD ベースのストレージ、およびデータ転送の割り当てが月額の固定料金で提供されます。Lightsail データベースは、AWS Command Line Interface (AWS CLI)、API、または SDK を使用して管理することもできます。

Lightsail データベースを作成する

Amazon Lightsail でマネージドデータベースを作成します。MySQL または PostgreSQL の最新メジャーバージョンから選択し、データベースをスタンダードプランまたは高可用性プランで設定できます。

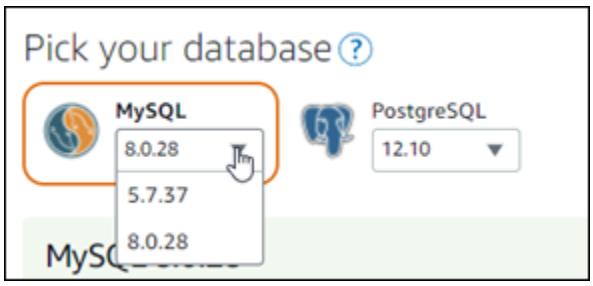
> **Note**
>
> Lightsail のマネージドデータベースの詳細については、「データベースを選択する」を参照してください。

データベースを作成する

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [データベース] を選択します。

3. [データベースの作成] を選択します。
4. データベースの AWS リージョン およびアベイラビリティーゾーンを選択します。

   1. [AWS リージョン とアベイラビリティーゾーンの変更] を選択し、リージョンを選択します。
   2. [アベイラビリティーゾーンの変更] を選択し、アベイラビリティーゾーンを選択します。
5. データベースのタイプを選択します。使用可能なエンジンオプションのいずれかから、ドロップダウンメニューを選択し、Lightsail でサポートされている最新のメジャーデータベースバージョンのどちらかを選択します。



6. 必要に応じて、以下のいずれかのオプションを選択します。

   - ログイン認証情報の指定 – 独自のデータベースユーザー名とパスワードを指定します。それ以外の場合は、Lightsail が代わりにユーザー名を指定し、強力なパスワードを作成します。
     - 独自のユーザー名を指定するには、 [Specify login credentials (ログイン認証情報の指定)] を選択し、テキストボックスにユーザー名を入力します。選択したデータベースエンジンに応じて、次の制約が適用されます。

       (MySQL)

       - MySQL に必要です。
       - 1 ~ 16 文字の英字または数字を使用することができます。
       - 1 字目は英字である必要があります。
       - 選択したデータベースエンジンの予約語は使用できません。MySQL の予約語の詳細については、MySQL 5.6、MySQL 5.7、または MySQL 8.0 のキーワードと予約語の記事を参照してください。

       PostgreSQL

       - PostgreSQL には必須です。
       - 1 ~ 63 文字の英字または数字が使用できます。
       - 1 字目は英字である必要があります。

- 選択したデータベースエンジンの予約語は使用できません。PostgreSQL の予約語の詳細については、PostgreSQL 9.6、PostgreSQL 10、PostgreSQL 11、または PostgreSQL 12 の SQL キーワードの記事を参照してください。

- 独自のパスワードを指定するには、[Create a strong password for me (強力なパスワードを作成する)] チェックボックスをオンにし、パスワードをテキストボックスに入力します。パスワードには「/」「"」または「@」を除く表示可能な任意の ASCII 文字を使用することができます。MySQL データベースの場合、パスワードには 8 ~ 41 文字の英数字を使用できます。PostgreSQL データベースの場合、パスワードには 8 ~ 128 文字の英数字を使用できます。

- マスターデータベース名の指定 – 独自のプライマリデータベース名を指定します。指定しないと、Lightsail で自動的に名前が指定されます。独自のプライマリデータベース名を指定するには、 [Specify the master database name (マスターデータベース名の指定)] を選択し、テキストボックスに名前を入力します。選択したデータベースエンジンに応じて、次の制約が適用されます。

  (MySQL)

  - 1 ~ 64 の文字または数字を使用する必要があります。
  - 先頭は文字を使用する必要があります。後続の文字には、英字、アンダースコア、または数字 (0 ~ 9) を使用することができます。
  - 選択したデータベースエンジンの予約語は使用できません。MySQL の予約語の詳細については、MySQL 5.6、MySQL 5.7、または MySQL 8.0 のキーワードと予約語の記事を参照してください。

  PostgreSQL

  - 1 ~ 63 文字の英字、数字、またはアンダースコアを使用する必要があります。
  - 先頭は文字を使用する必要があります。後続の文字には、英字、アンダースコア、または数字 (0 ~ 9) を使用することができます。
  - 選択したデータベースエンジンの予約語は使用できません。PostgreSQL の予約語の詳細については、PostgreSQL 9.6、PostgreSQL 10、PostgreSQL 11、または PostgreSQL 12 の SQL キーワードの記事を参照してください。

7. 高可用性データベースプランまたはスタンダードデータベースプランを選択します。

   高可用性プランでデータベースを作成すると、プライマリデータベースのほかに、フェイルオーバーのサポートとしてスタンバイ用のセカンダリデータベースが別のアベイラビリティーゾーンに作成されます。詳細については、「高可用性データベース」を参照してください。複数の異な

る価格のデータベースバンドルオプションを利用できます。オプションごとにメモリ、処理、ストレージ容量、および転送レートのレベルが異なります。

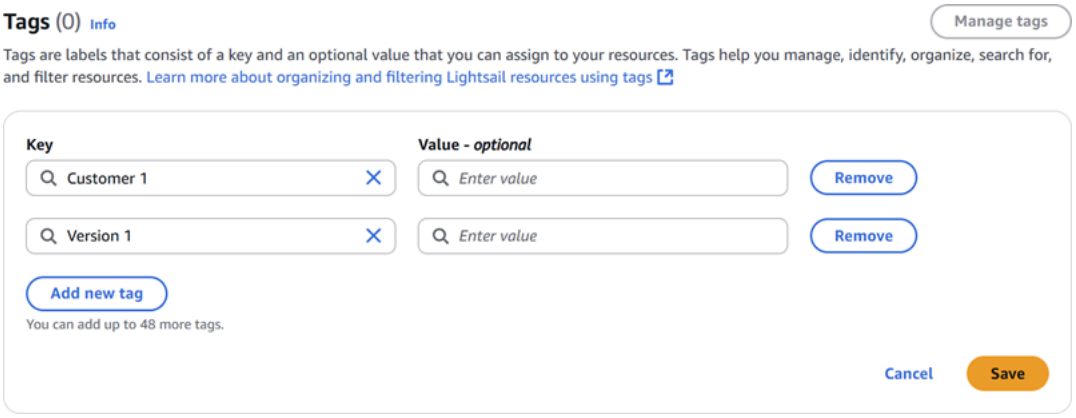
8. データベースの名前を入力します。

   リソース名:

   - Lightsail アカウントの各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

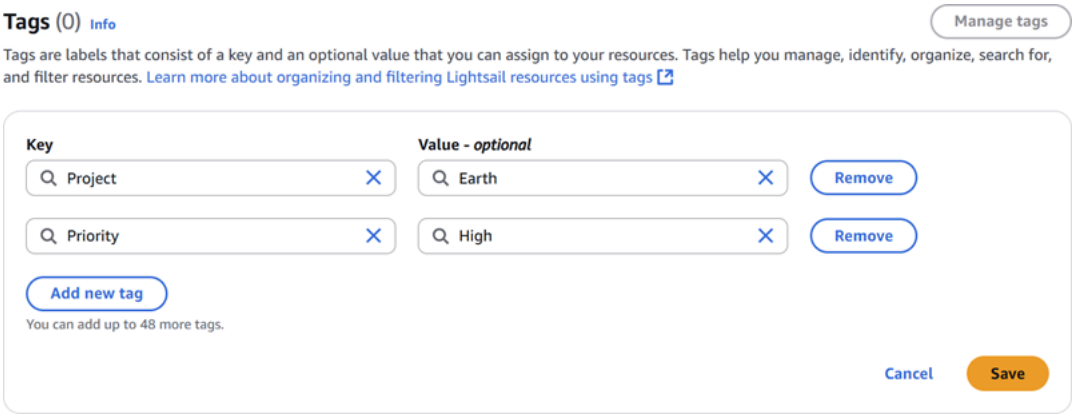
9. 以下のいずれかのオプションを選択して、データベースにタグを追加します。

   - [キーオンリータグの追加] または [タグを編集] (タグが追加済みの場合)。タグキーのテキストボックスに新しいタグを入力し、Enter キーを押します。タグの入力を完了したら、[保存] を選択してタグを追加し、追加しない場合は、[キャンセル] を選択します。



   - [Create a key-value tag] (キーバリュータグを作成) から [Key] (キー) テキストボックスにキーを入力し、[Value] (値) テキストボックスに値を入力します。タグの入力を完了したら、[保存] を選択し、追加しない場合は、[キャンセル] を選択します。

     キーバリューのタグは、保存する際に一つずつ追加することができます。さらにキーバリュータグを追加するには、以上のステップを繰り返します。



> **Note**
>
> 「キーのみ」のタグと「キーバリュー」のタグの詳細については、「タグ」を参照してください。

10. [データベースの作成] を選択します。

    数分以内に、Lightsail データベースの準備が完了します。データのインポートに関する設定を開始するか、データベースクライアントを使用して接続することができます。

次のステップ

Lightsail で新しいデータベースの使用を開始したら、データベースの管理に役立つ以下のガイドを参照してください。

- データベースのデータのインポートモードを設定する
- のデータベースのパブリックモードを設定するAmazon Lightsail
- データベースのパスワードを管理する
- MySQL データベースに接続する
- PostgreSQL データベースに接続する
- MySQL データベースにデータをインポートする
- データを PostgreSQL データベースにインポートする
- データベースのスナップショットを作成する

プロジェクトに適した Lightsail データベースを選択する

Amazon Lightsail は、MySQL データベースと PostgreSQL データベースの最新のメジャーバージョンを提供しています。このガイドでは、プロジェクトに適したデータベースの選択に役立つ情報を提供します。

Lightsail は、SQL Server で Windows Server 2022 インスタンスも提供します。詳細については、「Amazon Lightsail インスタンスのイメージの選択」を参照してください。

Lightsail で管理されたデータベースを比較する

MySQL

MySQL 5.7 および 8.0 は Lightsail で使用できます。MySQL は最も広く採用されているオープンソースのリレーショナルデータベースです。多くの一般的なウェブサイト、アプリケーション、および商用製品でプライマリのリレーショナルデータストアとして使用されています。MySQL は、高信頼性の安定した安全な SQL ベースのデータベース管理システムとして、20 年以上にわたってコミュニティからの開発の支援とサポートを受けています。MySQL データベースは、ミッションクリティカルなアプリケーションや動的なウェブサイトなど、さまざまなユースケースに適しています。また、ソフトウェア、ハードウェア、およびアプライアンスの埋め込みデータベースとしても機能します。

> **Important**
>
> 2024 年 6 月 30 日以降、Lightsail は MySQL 5.7 のサポートを終了し、このブループリントを使用した新規データベースの作成が不可となります。データベースインスタンスのメジャーバージョンをアップグレードする方法については、「Lightsail データベースのメジャーバージョンをアップグレードする」を参照してください。

詳細については、次の MySQL ドキュメントを参照してください。

- MySQL 5.7 のドキュメント
- MySQL 8.0 のドキュメント

PostgreSQL

Lightsail では、PostgreSQL 12、13、14、15 と 16 が利用可能です。PostgreSQL は、30 年以上もの間開発されてきた強力なオープンソースのオブジェクトリレーショナルデータベースシステムであり、信頼性、機能の堅牢性、およびパフォーマンスで高い評価を得ています。

公式ドキュメントには PostgreSQL のインストール方法と使用方法を説明する豊富な情報があります。PostgreSQL コミュニティでは、テクノロジーに精通し、その仕組みを理解して、そしてキャリアの機会を見つけるために役立つ多くの場所が提供されています。

> ⚠ Important
>
> - 2024 年 6 月 30 日以降、Lightsail は PostgreSQL 11 のサポートを終了し、このブループリントを使用した新規データベースの作成が不可となります。データベースインスタンスのメジャーバージョンをアップグレードする方法については、「Lightsail データベースのメジャーバージョンをアップグレードする」を参照してください。
> - PostgreSQL コミュニティは 2024 年 11 月 14 日に PostgreSQL 12 を廃止する予定であり、このブループリントから起動された Lightsail インスタンスは、この日付以降にセキュリティパッチを受信しません。そのため、Amazon Lightsail は 2025 年 2 月 28 日をもって PostgreSQL 12 への標準サポートを終了します。2025 年 2 月 28 日以降、PostgreSQL 12 を使用して新しい Lightsail データベースを作成することは不可となります。詳細については、PostgreSQL のウェブサイトを参照してください。

詳細については、次の PostgreSQLドキュメントを参照してください。

- PostgreSQL 11 ドキュメント
- PostgreSQL 12 ドキュメント
- PostgreSQL 13 ドキュメント
- PostgreSQL 14 ドキュメント
- PostgreSQL 15 ドキュメント
- PostgreSQL 16 ドキュメント

データのインポートを最適化する

Lightsail では、複数のデータベースプランを利用できます。プランごとにメモリ、vCPU、ストレージ、データ転送枠の仕様は異なります。データベースプランごとに仕様が異なるため、新しい Lightsail データベースにインポートするデータの量に応じて、適切なサイズのデータベースプランを

選択することが重要です。サイズの要件に満たないプランを選択すると、データのインポートが遅くなることがあります。以下のガイドラインに従って、データのインポート要件に応じた適切なデータベースプランを選択してください。

- Micro $15 USD/月データベースプラン – データの転送量が 10 GB を超えると、データのインポートが遅くなる場合があります。
- Small $30 USD/月データベースプラン – データの転送量が 20 GB を超えると、データのインポートが遅くなる場合があります。
- Medium $60 USD/月データベースプラン – データの転送量が 85 GB を超えると、データのインポートが遅くなる場合があります。
- Large $115 USD/月データベースプラン – データの転送量が 156 GB を超えると、データのインポートが遅くなる場合があります。
- Xlarge 230 USD/月のデータベースプラン — 156 GB を超えるデータを転送した場合、データのインポートが遅くなる可能性があります。
- 2Xlarge 460 USD/月のデータベースプラン — 156 GB を超えるデータを転送した場合、データのインポートが遅くなる可能性があります。

> **Note**
>
> データベースへのデータのインポートの詳細については、「MySQL データベースへのデータのインポート」または「PostgreSQL データベースへのデータへのインポート」を参照してください。

Lightsail の高可用性データベース

Lightsail の高可用性マネージドデータベースは、フェイルオーバー サポートを提供するために、1 つのアベイラビリティーゾーンでプライマリデータベースを保持し、別のアベイラビリティーゾーンでスタンバイ用のセカンダリデータベースを保持します。高可用性データベースは、使用負荷の高い、データの冗長性を必要とする本番稼働用のワークロードにお勧めします。開発およびテストの目的には、高可用性ではないスタンダードデータベースを使用できます。

高可用性データベースを作成するには、マネージドデータベースの作成時に Lightsail の高可用性データベースプランのいずれかを選択します。詳細については、「データベースの作成」を参照してください。また、スタンダードデータベースを高可用性データベースに変更することもできます。スタンダードデータベースのスナップショットを作成し、そのスナップショットから新しいデータベー

スを作成して、高可用性プランを選択します。詳細については、「スナップショットからデータベースを作成する」を参照してください。

クライアントアプリケーションから Lightsail MySQL データベースに接続する

Amazon Lightsail で MySQL マネージドデータベースを作成すると、標準の MySQL クライアントアプリケーションまたはユーティリティを使用して、このインスタンスに接続できます。Lightsail コンソールのデータベース管理ページから、データベースのエンドポイント、ポート、ユーザー名、およびパスワードを取得する必要があります。これらの値は、クライアントやウェブアプリケーションでデータベース接続を設定するときに指定します。

このガイドでは、必要な接続情報を取得する方法、およびマネージドデータベースに接続するように MySQL Workbench を設定する方法について説明します。

> **Note**
>
> PostgreSQL データベースへの接続の詳細については、「PostgreSQL データベースに接続する」を参照してください。

ステップ 1: MySQL データベース接続の詳細を取得する

Lightsail コンソールからデータベースのエンドポイントとポート情報を取得します。これらの情報は、データベースに接続するようにクライアントを設定するときに使用します。

データベース接続の詳細を取得するには

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [データベース] を選択します。
3. 接続先のデータベースの名前を選択します。
4. [接続] タブの [Endpoint and port (エンドポイントとポート)] セクションで、エンドポイントとポートの情報を書き留めます。

   間違えて入力しないように、エンドポイントをクリップボードにコピーすることをお勧めします。そのためには、エンドポイントを強調表示し、Ctrl+C (Windows) または Cmd+C (macOS) を押してクリップボードにコピーします。次に、Ctrl+V または Cmd+V を押して貼り付けます。

5. [接続] タブの [ユーザー名とパスワード] セクションでユーザー名を確認し、[パスワード] セクションの [表示] を選択します。現在のデータベースパスワードが表示されます。

   マネージドパスワードは複雑であるため、間違って入力しないように、これもコピーして貼り付けることをお勧めします。マネージドパスワードを強調表示し、Ctrl+C (Windows) または Cmd+C (macOS) を押してクリップボードにコピーします。次に、Ctrl+V または Cmd+V を押して貼り付けます。

ステップ 2: MySQL データベースのパブリック可用性を設定する

データベースを外部に接続するには、データベースのパブリックモードを有効にするか、データベース AWS リージョン とは異なる のLightsailインスタンスからパブリックモードを有効にする必要があります。パブリックモードを有効にすると、誰でもデータベースのユーザー名とパスワードを使用してデータベースに接続できます。データベースのパブリックでの可用性を設定するには、ガイドの「データベースのパブリックモードの設定」の手順に従います。

> **Note**
>
> データベースと同じリージョンにある Lightsail インスタンスのいずれかからデータベースに接続する場合は、ステップ 3 に進みます。

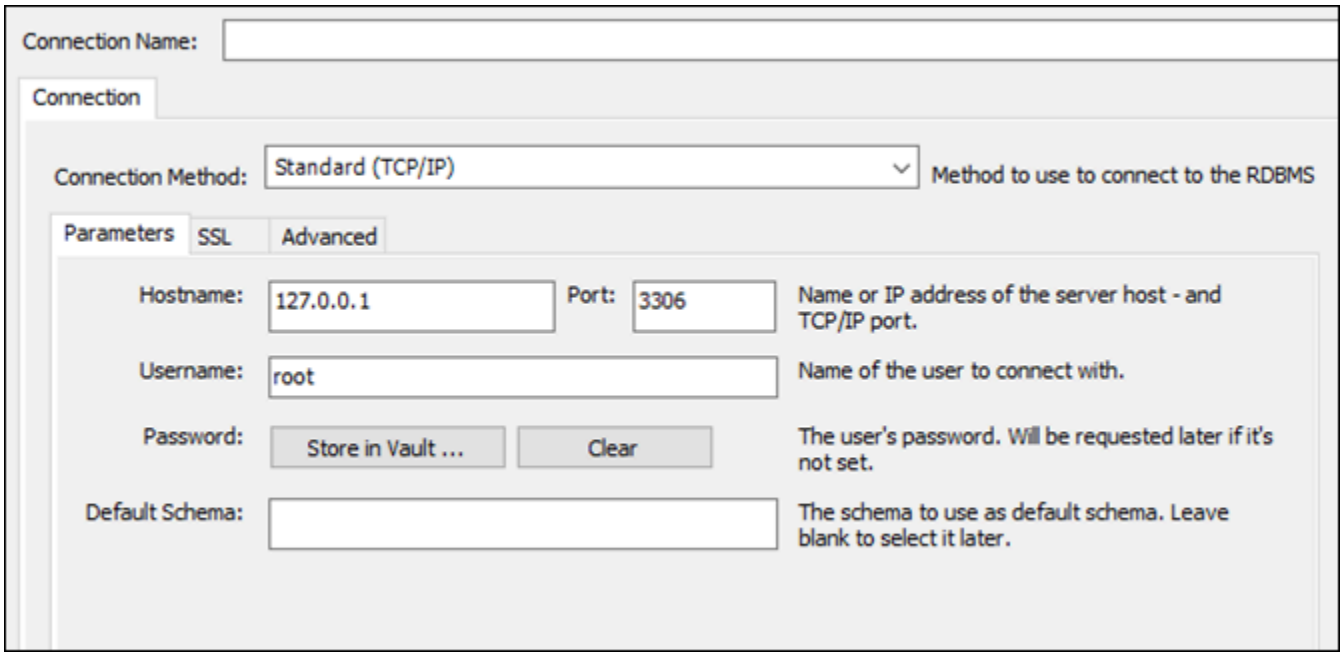
ステップ 3: MySQL データベースに接続するようにデータベースクライアントを設定する

MySQL データベースに接続するには、前に取得したエンドポイントとポートを使用するようにデータベースクライアントを設定します。以下のステップは、MySQL Workbench を設定する方法を示していますが、これらのステップは他のクライアントと同様の場合があります。

> **Note**
>
> MySQL Workbench の使用方法の詳細については、MySQL Workbench のマニュアルを参照してください。

データベースに接続するように MySQL Workbench を設定するには

1. MySQL Workbench を開きます。
2. [Database (データベース)] メニュー、[Manage connections (接続の管理)] の順に選択します。
3. 表示されるフォームに以下の情報を入力します。

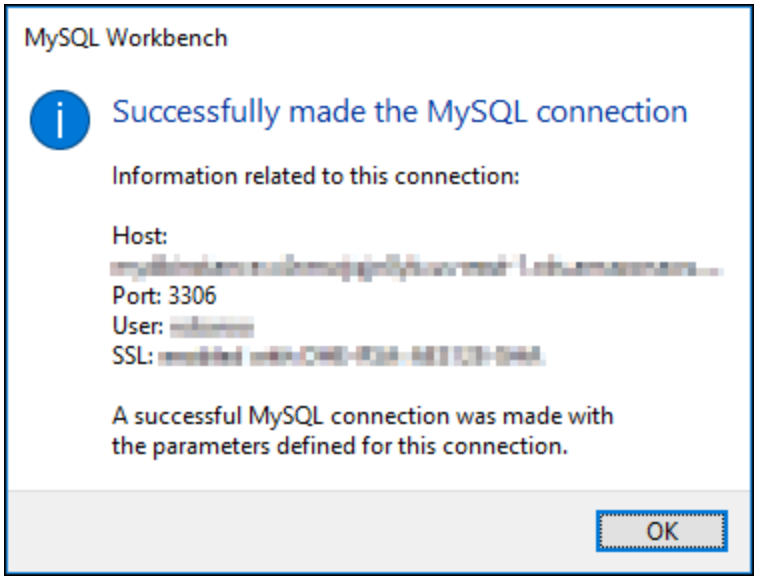


- 接続名: データベースに似た接続名を使用することをお勧めします。後で接続を識別しやすくなります。
- 接続方法: [標準 (TCP/IP)] を選択します。

- ポート — 先に取得したデータベースのポートを入力します。MySQL のデフォルトポートは 3306 です。
- ホスト名 — 先に取得したデータベースエンドポイントを入力します。データベースエンドポイントを Lightsail コンソールからコピーした場合は、まだクリップボード内に残っているため、Ctrl+V (Windows) または Cmd+V (macOS) を押して貼り付けます。
- ユーザー名: 前に取得したデータベースユーザー名を入力します。
- パスワード: [Store in Vault (ボールトに保存)] を選択します。表示されるウィンドウで、前に取得したデータベースパスワードを入力します。パスワードを Lightsail コンソールからコピーした場合は、まだクリップボード内に残っているため、Ctrl+V (Windows) または Cmd+V (macOS) を押して貼り付けます。[OK] をクリックしてパスワードを保存します。
- デフォルトスキーマ: このテキストボックスは空白のままにします。

4. [Test connection (テスト接続)] を選択し、クライアントからデータベースに接続できるかどうかを確認します。

   接続に成功すると、次の例に示すようなプロンプトが表示されます。情報を確認したら [OK] をクリックして閉じます。

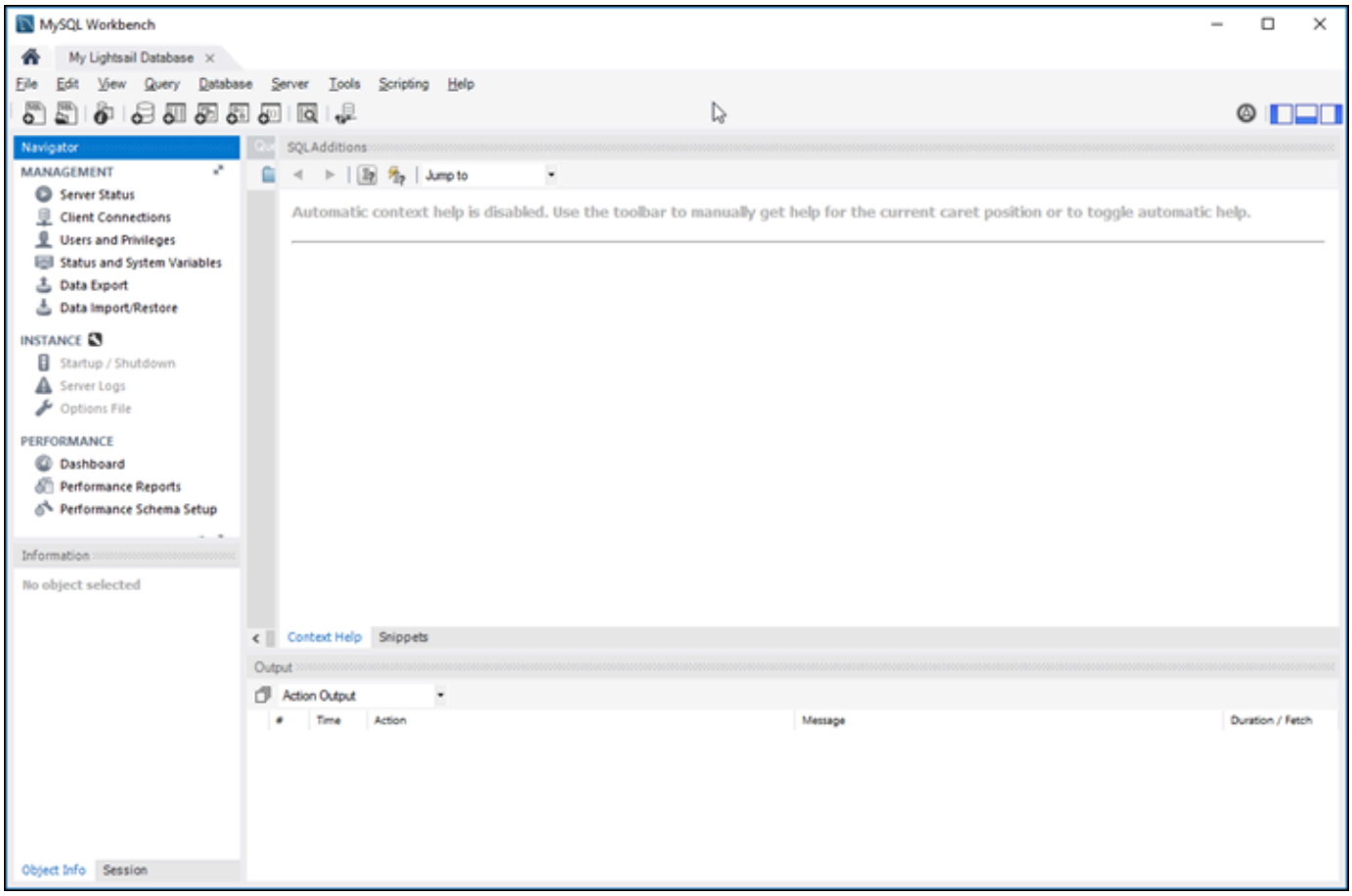


5. [New (新規)] を選択して新しい接続の詳細を保存し、[Close (閉じる)] を選択して接続管理ウィンドウを閉じます。

   新しいデータベース接続が、MySQL Workbench アプリケーションのホームページの [MySQL Connections (MySQL 接続)] セクションに表示されます。

6. データベースに接続するには、新しいデータベース接続を選択します。

接続に成功すると、次の例に示すようなウィンドウが表示されます。



次の手順

次のガイドは、Lightsail のデータベースにデータをインポートする際に役立ちます。

- MySQL データベースにデータをインポートする

SSL/TLS を使用して Lightsail MySQL データベースに安全に接続する

Amazon Lightsail によって SSL 証明書が作成され、プロビジョニング時に MySQL マネージド型データベースにインストールされます。証明書は認証機関 (CA) によって署名され、なりすまし攻撃から保護するために、SSL 証明書の共通名 (CN) としてデータベースエンドポイントが含まれます。

Lightsail によって作成された SSL 証明書は信頼されたルートエンティティであり、ほとんどの場合は使用できますが、アプリケーションが証明書チェーンを受け入れていない場合は使用できない可能性があります。アプリケーションが証明書チェーンを受け入れていない場合は、 AWS リージョンに接続している中間証明書の使用が必要になる場合があります。

マネージドデータベースの CA 証明書、サポートされる AWS リージョン、アプリケーションの中間証明書のダウンロード方法の詳細については、「マネージドデータベースの SSL 証明書をダウンロード」を参照してください。

サポートされている接続

MySQL は以下のバージョンで安全な接続のため yaSSL を使用します。

- MySQL バージョン 5.7.19 および 5.7 以前のバージョン
- MySQL バージョン 5.6.37 および 5.6 以前のバージョン
- MySQL バージョン 5.5.57 および 5.5 以前のバージョン

MySQL は以下のバージョンで安全な接続のため OpenSSL を使用します。

- MySQL バージョン 8.0
- MySQL バージョン 5.7.21 以降の 5.7 バージョン
- MySQL バージョン 5.6.39 以降の 5.6 バージョン
- MySQL バージョン 5.5.59 以降の 5.5 バージョン

MySQL マネージド型データベースは、Transport Layer Security (TLS) バージョン 1.0、1.1、1.2 をサポートしています。以下のリストでは、MySQL バージョンがサポートする TLS を示しています。

- MySQL 8.0 - TLS 1.0、TLS 1.1、および TLS 1.2
- MySQL 5.7 - TLS 1.0 と TLS 1.1 TLS 1.2 は、MySQL 5.7.21 以降でのみサポートされています。
- MySQL 5.6 - TLS1.0
- MySQL 5.5 - TLS1.0

前提条件

- データベースへの接続に使用するコンピュータに MySQL サーバーをインストールします。詳細については、MySQL ウェブサイトの「MySQL Community Server download」を参照してください。
- データベースの該当する証明書をダウンロードします。詳細については、「マネージドデータベースの SSL 証明書をダウンロード」を参照してください。

SSL を使用して MySQL データベースに接続する

SSL を使用して MySQL データベースに接続するには、以下の手順を実行します。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. MySQL データベースのバージョンに応じて、以下のコマンドのいずれかを入力します。

   - MySQL 5.7 以降のデータベースに接続するには、以下のコマンドを入力します。

     ```
     mysql -h DatabaseEndpoint --ssl-ca=/path/to/certificate/rds-combined-ca-bundle.pem --ssl-mode=VERIFY_IDENTITY -u UserName -p
     ```

     コマンドを、以下のように置き換えます。

     - *DatabaseEndpoint* は、データベースのエンドポイントに置き換えます。
     - */path/to/certificate/rds-combined-ca-bundle.pem* は、データベースの証明書をダウンロードして保存したローカルパスに置き換えます。
     - *UserName* は、データベースのユーザー名に置き換えます。

     例:

     ```
     mysql -h ls-1c51a7c70a4fb55e542829a4e4e0d735ba42.czowadgeezqi.us-west-2.rds.amazonaws.com --ssl-ca=/home/ec2-user/rds-combined-ca-bundle.pem --ssl-mode=VERIFY_IDENTITY -u dbmasteruser -p
     ```

   - MySQL 6.7 以降のデータベースに接続するには、以下のコマンドを入力します。

     ```
     mysql -h DatabaseEndpoint --ssl-ca=/path/to/certificate/rds-combined-ca-bundle.pem --ssl-verify-server-cert -u UserName -p
     ```

コマンドを、以下のように置き換えます。

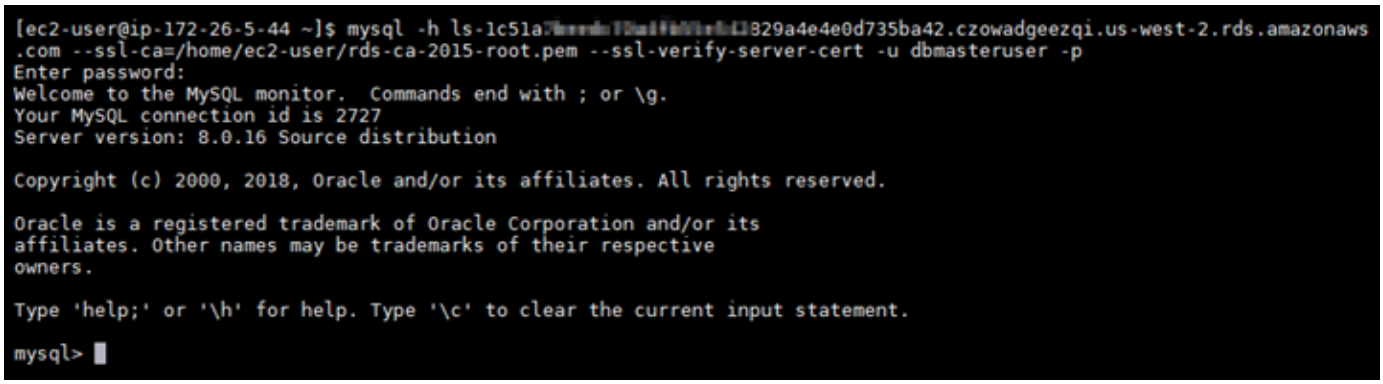
- *DatabaseEndpoint* は、データベースのエンドポイントに置き換えます。
- */path/to/certificate/rds-combined-ca-bundle.pem* は、データベースの証明書をダウンロードして保存したローカルパスに置き換えます。
- *UserName* は、データベースのユーザー名に置き換えます。

例:

```
mysql -h ls-1c51a7c70a4fb55e542829a4e4e0d735ba42.czowadgeezqi.us-
west-2.rds.amazonaws.com --ssl-ca=/home/ec2-user/rds-combined-ca-bundle.pem --
ssl-verify-server-cert -u dbmasteruser -p
```

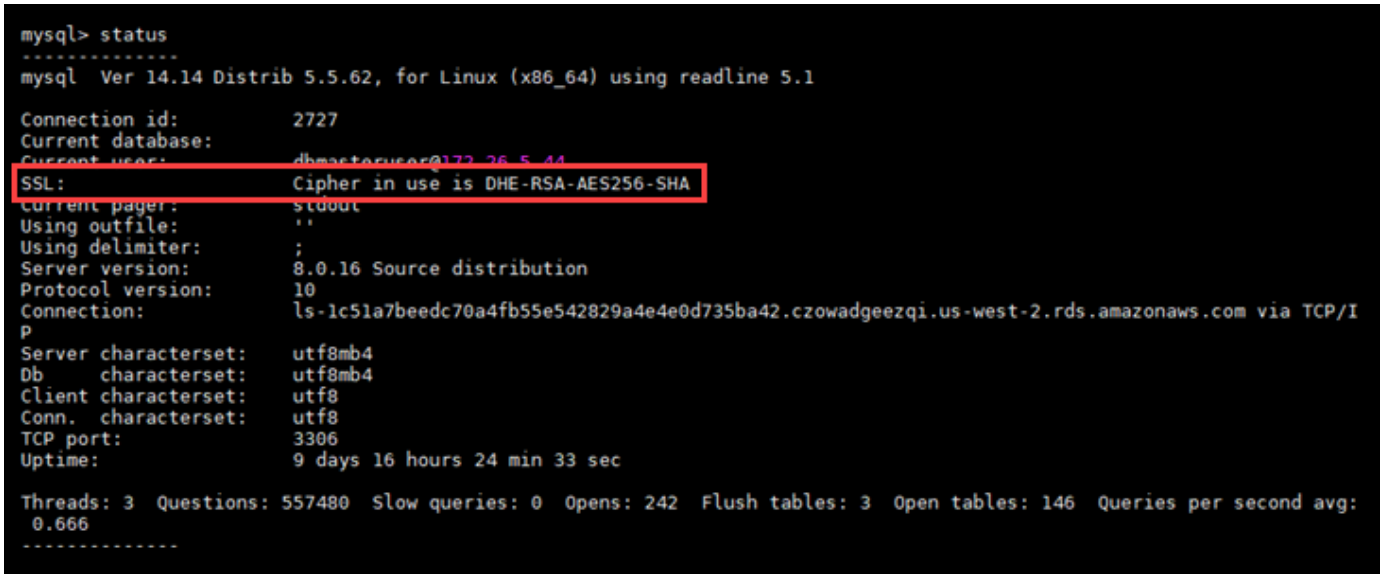
3. プロンプトが表示されたら、前のコマンドで指定したデータベースユーザーのパスワードを入力し、Enter キーを押します。

   以下の例のような結果が表示されるはずです。



4. 「**status**」と入力し、Enter キーを押して、接続のステータスを表示します。

   [SSL] の横で [Cipher in use is (使用中の暗号)] に値が表示されている場合、接続は暗号化されています。



Lightsail PostgreSQL データベースインスタンスに接続する

Amazon Lightsail で PostgreSQL マネージドデータベースを作成すると、標準の PostgreSQL クライアントアプリケーションまたはユーティリティを使用して、このインスタンスに接続できます。Lightsail コンソールのデータベース管理ページから、データベースのエンドポイント、ポート、ユーザー名、およびパスワードを取得する必要があります。これらの値は、クライアントやウェブアプリケーションでデータベース接続を設定するときに指定します。

このガイドでは、必要な接続情報を取得する方法、およびマネージドデータベースに接続するように pgAdmin クライアントを設定する方法について説明します。

> **Note**
>
> MySQL データベースへの接続の詳細については、「MySQL データベースに接続する」を参照してください。

ステップ 1: PostgreSQL データベース接続の詳細を取得する

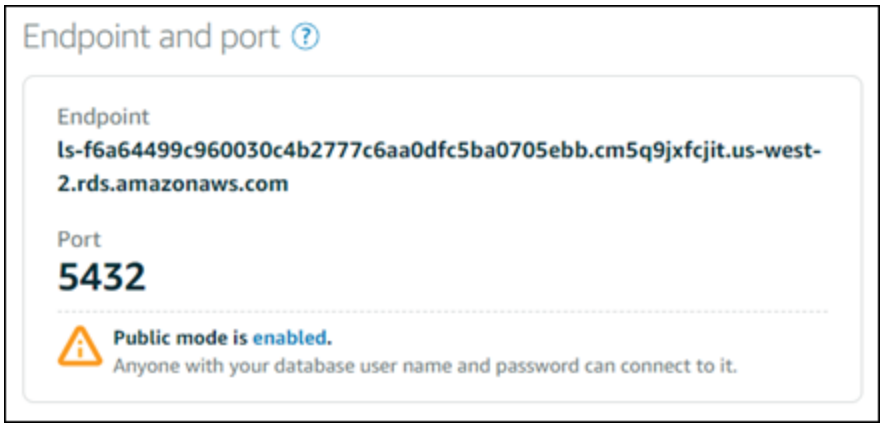
Lightsail コンソールからデータベースのエンドポイントとポート情報を取得します。これらの情報は、データベースに接続するようにクライアントを設定するときに使用します。

データベース接続の詳細を取得するには

1. Lightsail コンソール にサインインします。

2. 左のナビゲーションペインの [データベース] を選択します。
3. 接続先のデータベースの名前を選択します。
4. [接続] タブの [Endpoint and port (エンドポイントとポート)] セクションで、エンドポイントとポートの情報を書き留めます。

   間違えて入力しないように、エンドポイントをクリップボードにコピーすることをお勧めします。そのためには、エンドポイントを強調表示し、Ctrl+C (Windows) または Cmd+C (macOS) を押してクリップボードにコピーします。次に、Ctrl+V または Cmd+V を押して貼り付けます。



5. [接続] タブの [ユーザー名とパスワード] セクションでユーザー名を確認し、[パスワード] セクションの [表示] を選択します。現在のデータベースパスワードが表示されます。

   マネージドパスワードは複雑であるため、間違って入力しないように、これもコピーして貼り付けることをお勧めします。マネージドパスワードを強調表示し、Ctrl+C (Windows) または Cmd+C (macOS) を押してクリップボードにコピーします。次に、Ctrl+V または Cmd+V を押して貼り付けます。

ステップ 2: PostgreSQL データベースのパブリック可用性を設定する

データベースに外部から接続したり、データベースとは異なるリージョンの Lightsail インスタンスから接続するには、データベースのパブリックモードを有効にする必要があります。パブリックモードを有効にすると、誰でもデータベースのユーザー名とパスワードを使用してデータベースに接続できます。データベースのパブリックでの可用性を設定するには、ガイドの「データベースのパブリックモードの設定」の手順に従います。

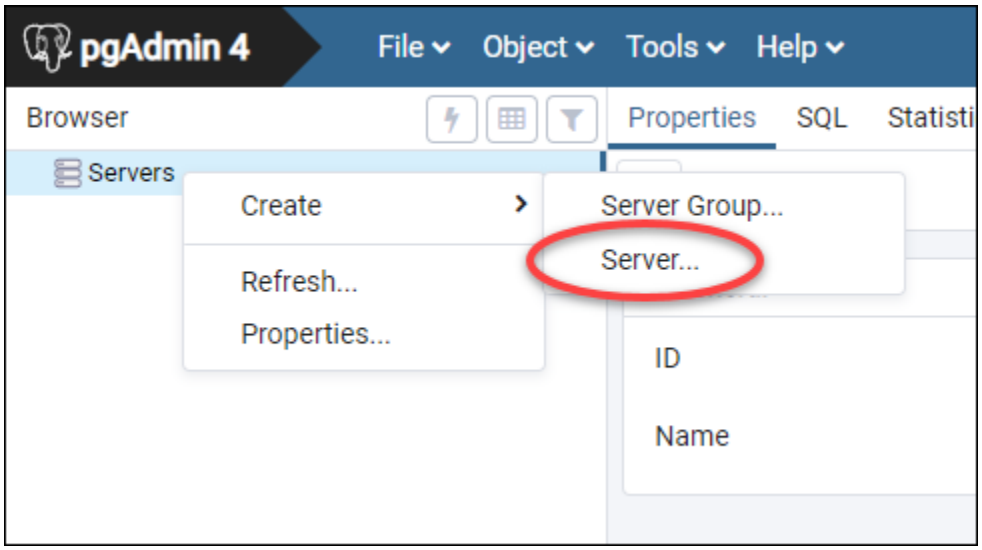
> **Note**
>
> データベースと同じリージョンにある Lightsail インスタンスのいずれかからデータベースに接続する場合は、ステップ 3 に進みます。

ステップ 3: PostgreSQL データベースに接続するようにデータベースクライアントを設定する

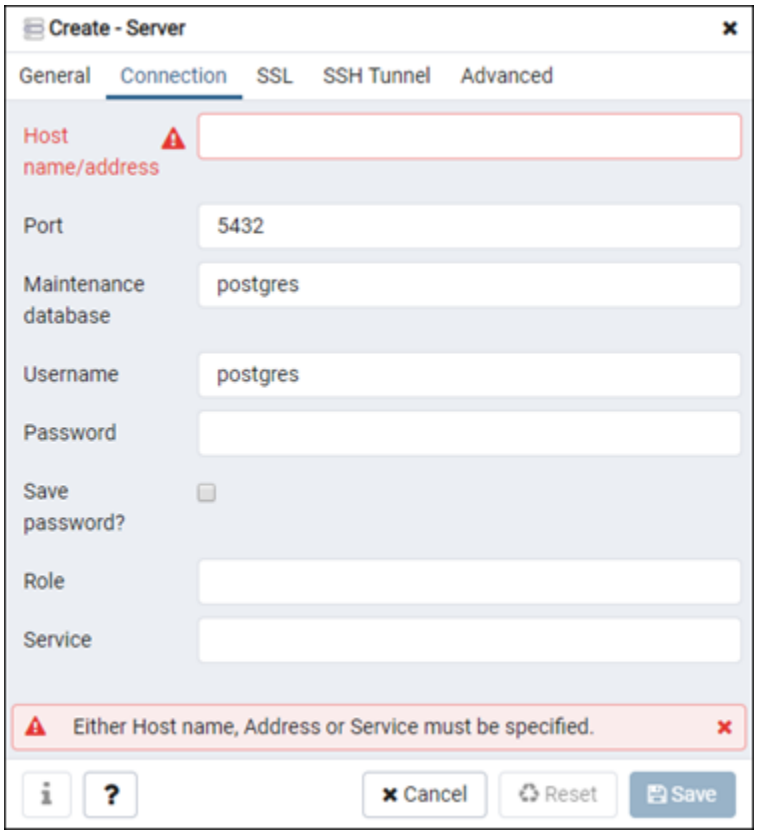
PostgreSQL データベースに接続するには、前に取得したエンドポイントとポートを使用するようにデータベースクライアントを設定します。以下のステップは、pgAdmin を設定する方法を示していますが、これらのステップは他のクライアントと同様の場合があります。

> **Note**
>
> pgAdmin を使用する方法の詳細については、「pgAdmin ドキュメント」を参照してください。

データベースに接続するように pgAdmin を設定するには

1. [pgAdmin] を開きます。
2. 左側のナビゲーションメニュー から [サーバー] を右クリックします。
3. [作成]、[サーバー] の順に選択します。
4. 

5. [Create-Server] フォームに、サーバー名を入力します。データベースに似た接続名を使用することをお勧めします。後で接続を識別しやすくなります。
6. [接続] タブを選択し、表示されるフォームに、次の情報を入力します。



- ホスト名/アドレス — 先に取得したデータベースエンドポイントを入力します。データベースエンドポイントを Lightsail コンソールからコピーした場合は、まだクリップボード内に残っているため、Ctrl+V (Windows) または Cmd+V (macOS) を押して貼り付けます。
- ポート — 先に取得したデータベースのポートを入力します。PostgreSQL のデフォルトポートは 5432 です。
- メンテナンスデータベース — クライアントが接続する初期データベースの名前を指定します。これは、Lightsail に PostgreSQL データベースを作成したときに指定したプライマリデータベースの名前です。

  プライマリデータベースの名前を覚えていない場合は、`postgres` を入力します。すべての PostgreSQL のマネージド型データベースには接続可能な `postgres` データベースがあり、後で PostgreSQL マネージド型データベースの他のすべてのデータベースにアクセスできるようになります。
- ユーザー名: 前に取得したデータベースユーザー名を入力します。

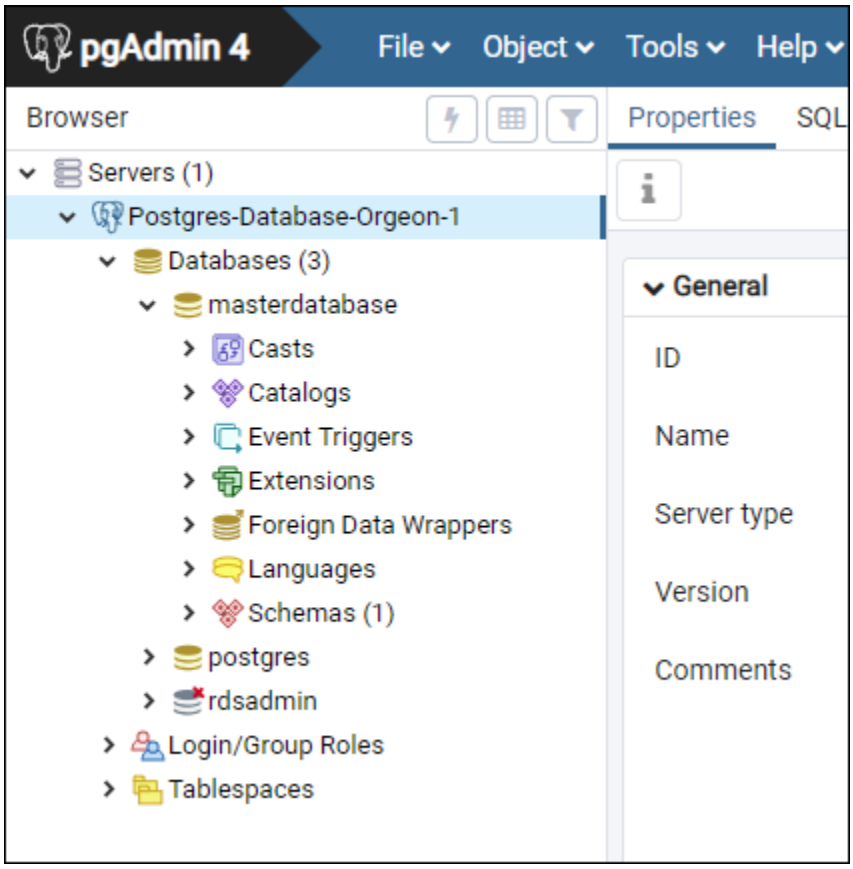
- パスワード — 先に取得したデータベースのパスワードを入力します。パスワードを Lightsail コンソールからコピーした場合は、まだクリップボード内に残っているため、Ctrl+V (Windows) または Cmd+V (macOS) を押して貼り付けます。[パスワードを保存] を選択してパスワードを保存します。
- ロールとサービス — これらのフィールドは空白のままにしておきます。

7. [保存] を選択して新しいサーバーの詳細を保存します。

   サーバーセクションの pgAdmin アプリケーションの左側のナビゲーションメニューに新しいデータベース接続が表示されます。

8. データベースに接続するには、新しいデータベース接続をダブルクリックします。

   接続に成功すると、そのデータベースの使用可能なリソースのリストが表示されます。



次の手順

次のガイドは、Lightsail のデータベースにデータをインポートする際に役立ちます。

- データを PostgreSQL データベースにインポートする

SSL を使用して Lightsail PostgreSQL データベースに安全に接続する

Amazon Lightsail によって SSL 証明書が作成され、プロビジョニング時に PostgreSQL (Postgres) マネージド型データベースにインストールされます。証明書は認証機関 (CA) によって署名され、なりすまし攻撃から保護するために、SSL 証明書の共通名 (CN) としてデータベースエンドポイントが含まれます。

Lightsail によって作成された SSL 証明書は信頼されたルートエンティティであり、ほとんどの場合は使用できますが、アプリケーションが証明書チェーンを受け入れていない場合は使用できない可能性があります。アプリケーションが証明書チェーンを受け入れていない場合は、 AWS リージョンに接続している中間証明書の使用が必要になる場合があります。

マネージドデータベースの CA 証明書、サポートされる AWS リージョン、アプリケーションの中間証明書のダウンロード方法の詳細については、「マネージドデータベースの SSL 証明書をダウンロード」を参照してください。

前提条件

- データベースへの接続に使用するコンピュータに PostgreSQL サーバーをインストールします。詳細については、Postgres ウェブサイトの「PostgreSQL Downloads」を参照してください。
- データベースの該当する証明書をダウンロードします。詳細については、「マネージドデータベースの SSL 証明書をダウンロード」を参照してください。

SSL を使用して Postgres データベースに接続する

SSL を使用して Postgres データベースに接続するには、以下の手順を実行します。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. PostgreSQL データベースに接続するには、以下のコマンドを入力します。

```
psql -h DatabaseEndpoint -p 5432 "dbname=DatabaseName user=UserName sslrootcert=/
path/to/certificate/rds-combined-ca-bundle.pem sslmode=verify-full"
```

コマンドを、以下のように置き換えます。

- *DatabaseEndpoint* は、データベースのエンドポイントに置き換えます。

- *DatabaseName* は、接続するデータベースの名前に置き換えます。
- *UserName* は、データベースのユーザー名に置き換えます。
- */path/to/certificate/rds-combined-ca-bundle.pem* は、データベースの証明書をダウンロードして保存したローカルパスに置き換えます。

例:

```
psql -h ls-8e81e07f8b821917b11e1c6a0e26cb73c203.czowadgeezqi.us-
west-2.rds.amazonaws.com -p 5432 "dbname=dbmaster user=dbmasteruser sslrootcert=/
home/ec2-user/rds-combined-ca-bundle.pem sslmode=verify-full"
```

3. プロンプトが表示されたら、前のコマンドで指定したデータベースユーザーのパスワードを入力し、Enter キーを押します。

   次の例のような結果が表示されます。[SSL connection (SSL 接続)] に値が表示されている場合、接続は暗号化されています。

Lightsail データベースを削除し、最終スナップショットを作成する

Amazon Lightsail でマネージドデータベースが不要になった場合は、これを削除します。データベースを削除すると、データベースに対する課金も停止します。

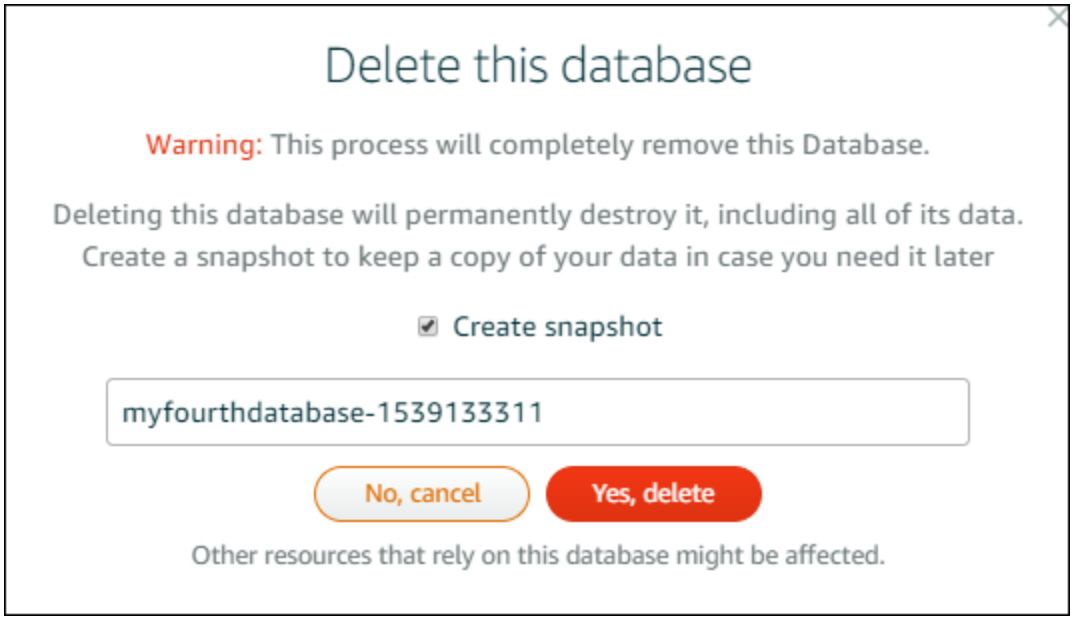
> **Note**
>
> 削除したデータベースは復元できません。このガイドで示す手順の一環としてデータベースの最終スナップショットを作成できます。または、削除プロセスとは関係なしにスナップショットを作成することもできます。詳細については、「データベースのスナップショットを作成する」を参照してください。

データベースを削除するには

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [データベース] を選択します。
3. 削除するデータベースの名前を選択します。
4. [削除] タブを選択します。
5. データベースを削除する前に最終スナップショットを作成するには、[削除前にスナップショットを作成する] をオンにします。次に、スナップショットの名前を入力します。

   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。
6. [データベースの削除] を選択します。
7. [はい、削除します] を選択して削除を確定します。



削除する前にスナップショットを作成することを選択した場合、Lightsail ホームページの [スナップショット] セクションで確認できます。

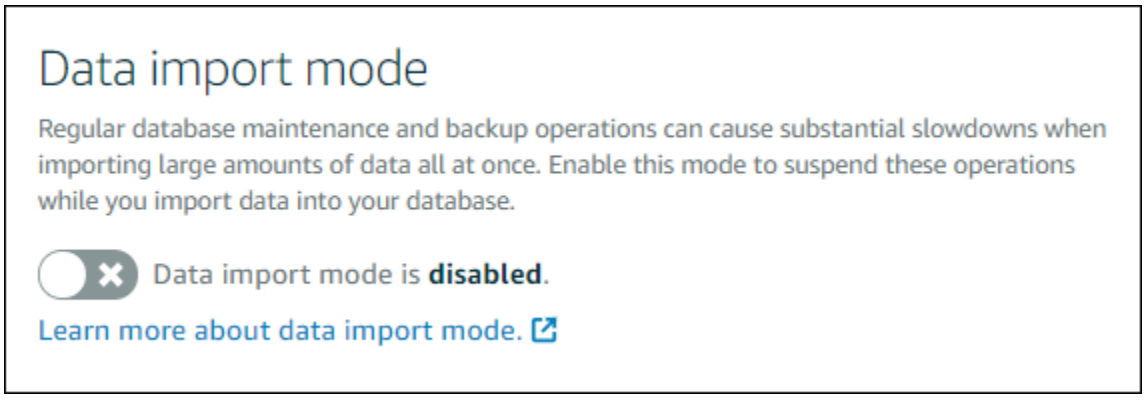
大きなデータセットを遅延なく Lightsail データベースにインポートする

大量のデータを一度にすべてインポートする場合、定期的なデータベースのバックアップオペレーションのせいで大幅な遅延や速度の低下が生じることがあります。大量のデータをインポートするときは、バックアップオペレーションを停止するように Amazon Lightsail マネージドデータベースのデータのインポートモードを設定できます。

> **Important**
>
> データのインポートモードが有効になるとすべての緊急復元バックアップが削除されます。データのインポートモードが有効になる前にバックアップする場合は、データベースのスナップショットを作成します。詳細については、「データベースのスナップショットを作成する」を参照してください。

データベースのデータのインポートモードを設定するには

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [データベース] を選択します。
3. データのインポートモードを設定するデータベースの名前を選択します。
4. [接続] タブの [Data import mode (データのインポートモード)] セクションで、トグルを使用してデータのインポートモードをオンにします。同様に、インポートの完了後は、トグルを使用してオフにします。



これでデータのインポートモードが有効になり、データベースのバックアップオペレーションが停止されます。データのインポートモードは一時的に有効にすることをお勧めします。大量のデータをデータベース内にインポートする必要がある場合に限り、使用してください。インポー

トが完了したらすぐにデータのインポートモードを無効にして、バックアップオペレーションを回復します。

> **Note**
>
> インポートするデータの量によっては、インポートが遅くなることがあります。詳細については、「データのインポートの最適化」を参照してください。

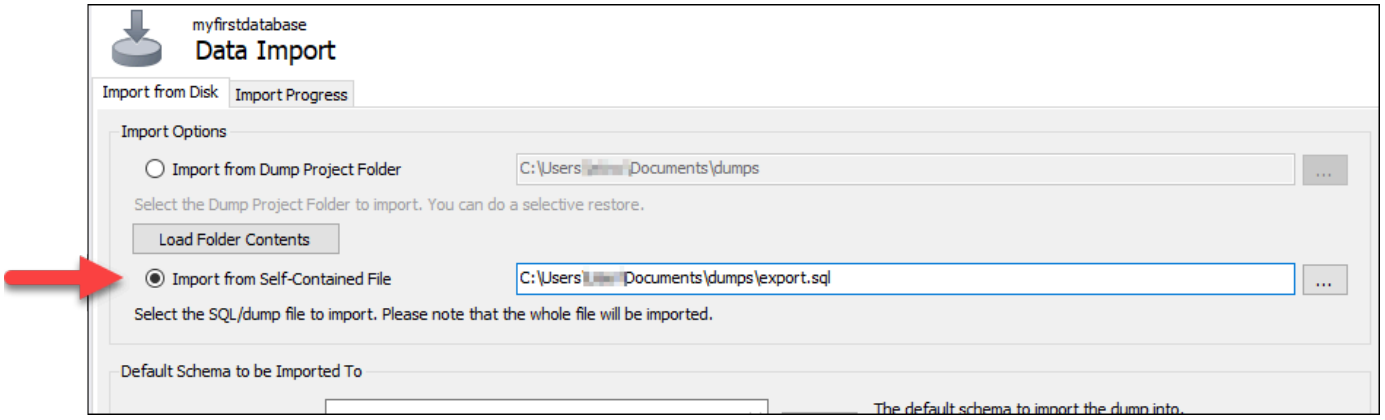
Lightsail MySQL データベースに SQL データをインポートする

MySQL Workbench を使用して、Amazon Lightsail の MySQL マネージドデータベースに SQL ファイル (.SQL) をインポートできます。

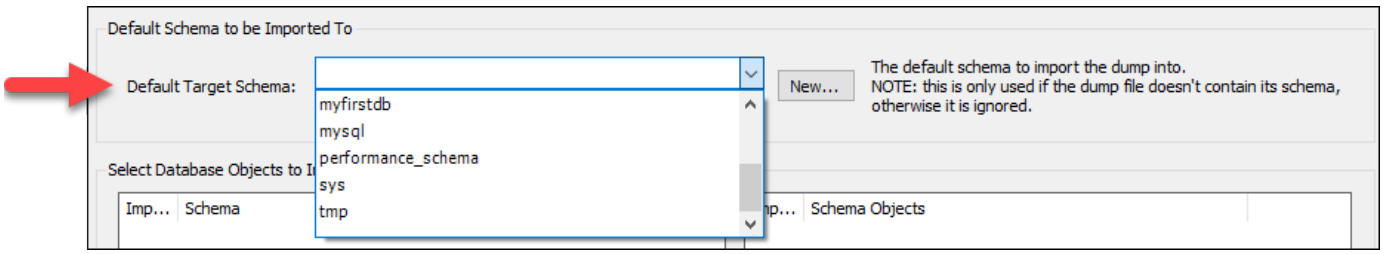
> **Note**
>
> MySQL Workbench をデータベースに接続する方法については、「MySQL データベースに接続する」を参照してください。

データベースにデータをインポートするには

1. MySQL Workbench を開きます。
2. MySQL 接続のリストで、MySQL マネージドデータベースを選択します。
3. 左のナビゲーションメニューから [Data Import/Restore (データのインポート/復元)] を選択します。
4. [Data Import] ペインで、[Import Option] セクションにある [Import from Self-Contained File] (自己完結型ファイルからインポート) を選択します。

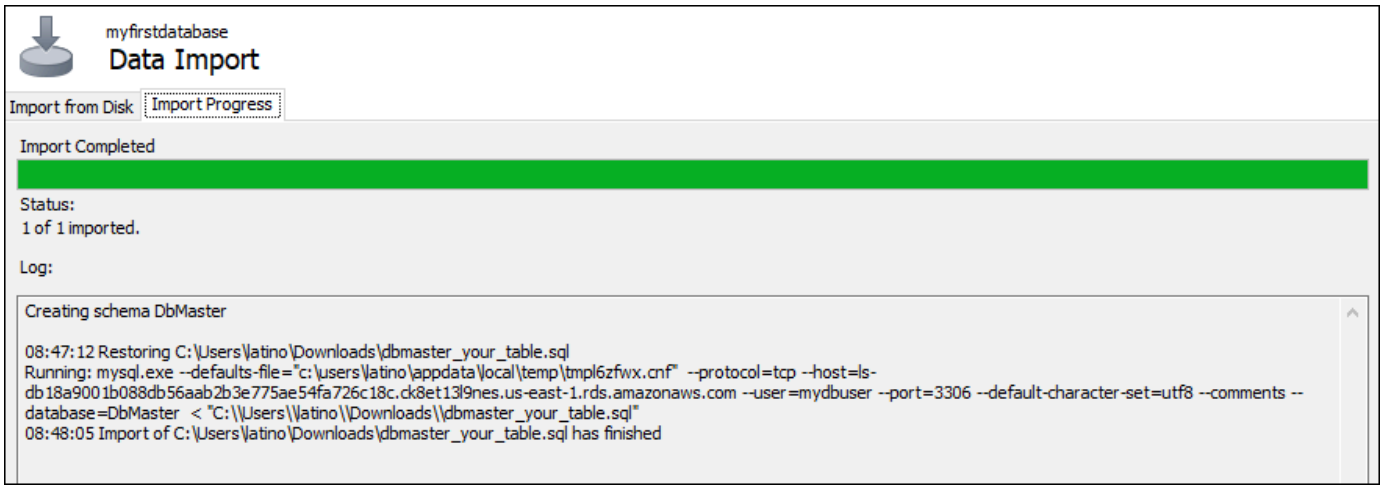


5. 省略記号ボタンを選択し、インポートする .SQL ファイルのローカルドライブを参照します。
6. インポートする .SQL ファイルを選択し、[Open (開く)] を選択します。
7. [Default Target Schema (デフォルトのターゲットスキーマ)] ドロップダウンメニューを選択し、ファイルをインポートする先の既存のデータベースを選択します。[New (新規)] を選択して新しいデータベースを作成することもできます。



8. [Start Import (インポートの開始)] を選択してインポートを開始します。

   .SQL ファイルのサイズに応じて、完了するまで数分かかる場合があります。インポートが完了した後、次のようなメッセージが表示されます。



PostgreSQL データベースバックアップを Lightsail マネージド型データベースにインポートします。

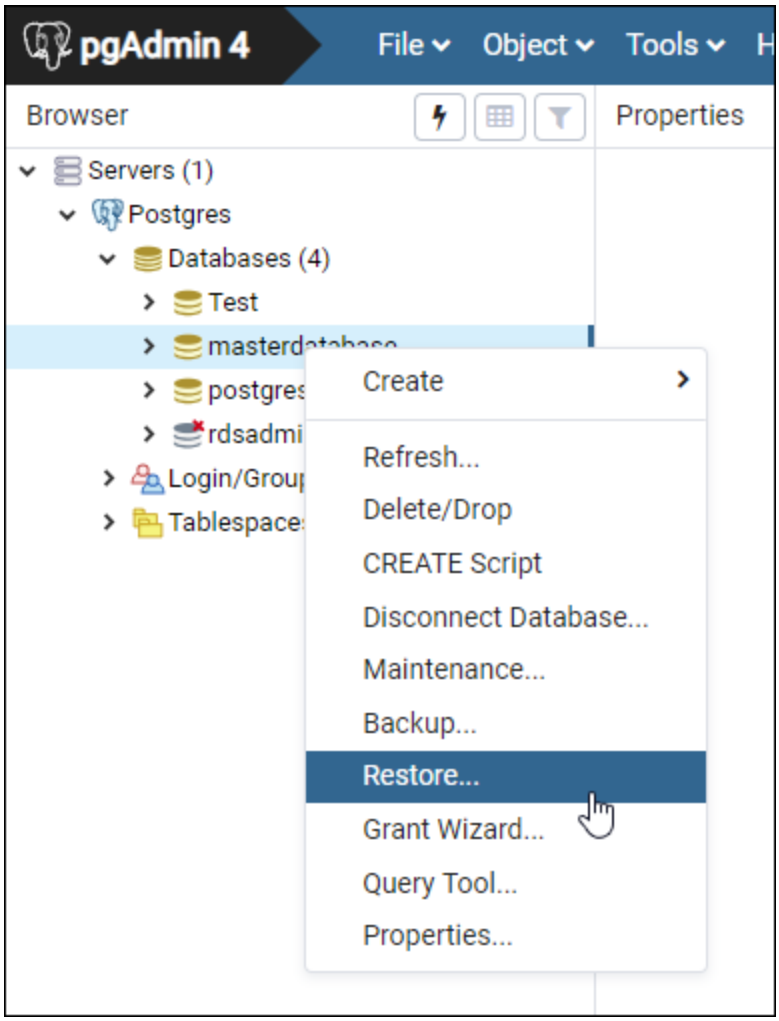
pgAdmin を使用して、Amazon Lightsail で PostgreSQL マネージドデータベースにデータベースバックアップファイルをインポートできます。

> **Note**
>
> pgAdmin をデータベースに接続する方法については、「PostgreSQL データベースに接続する」を参照してください。他のデータベースにインポートできる PostgreSQL データベース

バックアップを作成する方法について詳細は、pgAdmin ドキュメントの「バックアップダイアログ」を参照してください。

データベースにバックアップファイルをインポートするには

1. [pgAdmin] を開きます。
2. サーバー接続のリストで Amazon Lightsail でお使いの PostgreSQL マネージドデータベースをダブルクリックして、接続します。
3. Databases ノードを展開します。
4. データベースバックアップファイルからのデータをインポートするデータベースを右クリックして、その後 [Restore (復元)] を選択します。

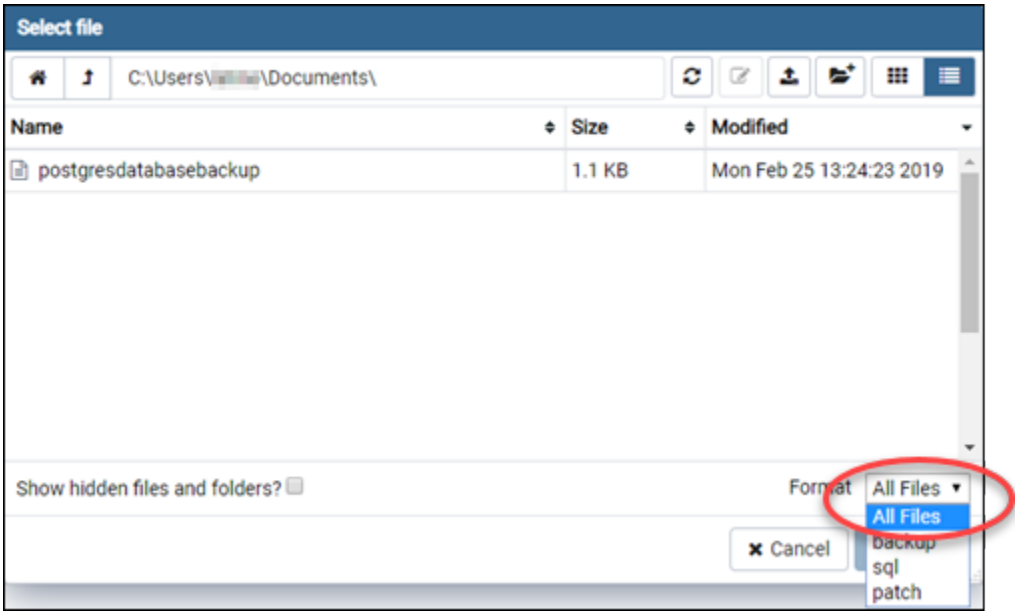


5. [Restore (復元)] フォームで、次のフィールドに入力します。

   - Format (形式) - バックアップファイルの形式を選択します。

- Filename (ファイル名) - 省略記号アイコンを選択し、ローカルドライブのデータベースバックアップファイルを見つけて選択します。ファイルがハイライトされたら、[Select (選択)] を選択して、[Restore (復元)] プロンプトに戻ります。

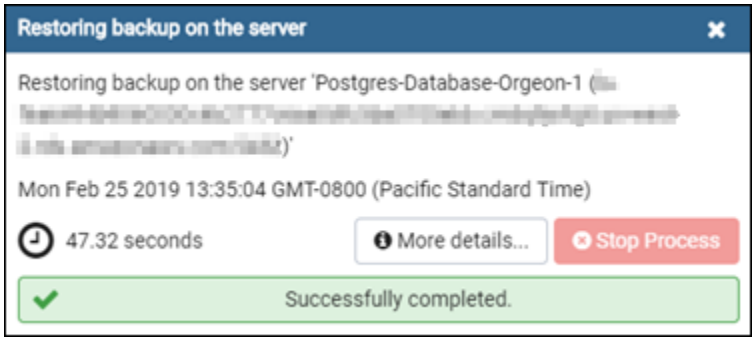
> **Note**
>
> [Format (形式)] ドロップダウンメニューを選択し、[All files (すべてのファイル)] を選択してローカルドライブにあるすべてのファイル形式を表示します。バックアップファイルは、デフォルトで選択されているファイル形式 (sql) とは異なる形式で保存されている場合があります。



- Number of jobs (ジョブの数) および Role name (ロール名) - これらのフィールドは空白のままにしておきます。

6. インポートを開始するには [Restore (復元)] を選択します。

   データベースバックアップファイルのサイズに応じて、完了するまで数分かかる場合があります。インポートが完了した後、次のようなメッセージが表示されます。



Lightsail データベースのログと履歴を表示する

Amazon Lightsail コンソールでデータベースのログと変更履歴を表示します。データベースのログは、データベースの問題の診断に役立つ情報を提供します。同様に、データベースの履歴は、データベースに加えられた変更を示します。これにより、最近の変更と問題の関連性を確認できます。

データベースのログを表示するには

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [データベース] を選択します。
3. ログを表示するデータベースの名前を選択します。
4. [Logs and history (ログと履歴)] タブを選択します。

   このページには、データベースのログとデータベースに加えられた変更の履歴が表示されます。
5. データベースのログを選択します。以下のデータベースのログを利用できます。

   MySQL データベースのログ

   - エラーログ — mysqld の起動時間およびシャットダウン時間の記録。これには、サーバーの起動とシャットダウン時、およびサーバーの実行中に発生するエラー、警告、注意などの診断メッセージも含まれます。詳細については、MySQL 5.6、MySQL 5.7、または MySQL 8.0 のドキュメントでエラーログに関する記事を参照してください。
   - 全般ログ — mysqld の動作に関する全般ログ。サーバーは、クライアントが接続または切断したときにこのログに情報を書き込みます。また、クライアントから受信した各 SQL ステートメントをログに記録します 詳細については、MySQL 5.6、MySQL 5.7、または MySQL 8.0 のドキュメントで全般クエリログに関する記事を参照してください。
   - スロークエリログ — 実行に long_query_time 秒を超える時間がかかり、検証に min_examined_row_limit 行以上を要した SQL ステートメントのレコード。詳細について

は、MySQL 5.6、MySQL 5.7、または MySQL 8.0 のドキュメントでスロークエリログに関する記事を参照してください。

> **Note**
>
> MySQL データベースの一般ログとスロークエリログはデフォルトで無効になっています。これらのログを有効にして、データの収集を開始するには、いくつかのデータベースパラメータを更新します。詳細については、「Amazon Lightsail で MySQL データベースの一般ログとスロークエリログを有効にする」を参照してください。

PostgreSQL データベースのログ

- Postgres ログ - データベースの起動時間およびシャットダウン時間の記録。これには、データベースの起動とシャットダウン時、およびデータベースが実行中に発生するエラー、警告、通知、デバッグなどの診断メッセージも含まれます。詳細については、PostgreSQL 9.6、PostgreSQL 10、PostgreSQL 11、PostgreSQL 12 のドキュメントでエラーレポートおよびログ記録の記事を参照してください。

トピック

- Lightsail で一般クエリログとスロークエリログを使用して MySQL クエリのパフォーマンスを監視します。

Lightsail で一般クエリログとスロークエリログを使用して MySQL クエリのパフォーマンスを監視します。

Amazon Lightsail の MySQL データベースの一般ログとスロークエリログは、デフォルトで無効になっています。これらのログを有効にして、データの収集を開始するには、いくつかのデータベースパラメータを更新します。Lightsail API、 AWS Command Line Interface (AWS CLI)、または SDKs を使用してデータベースパラメータを更新します。このガイドでは、 を使用してデータベースパラメータ AWS CLI を更新し、一般的なクエリログとスロークエリログを有効にする方法について説明します。また、一般ログとスロークエリログを制御するいくつかの追加オプションと、ログデータの保持期間がどのように処理されるかについても説明します。

前提条件

まだ AWS CLIをインストールして設定していない場合は、インストールして設定します。詳細については、「Amazon Lightsail で使用するために AWS Command Line Interface を設定する」を参照してください。

Lightsail コンソールで一般ログとスロークエリログを有効にする

Lightsail コンソールで一般ログとスロークエリログを有効にするには、general_log および slow_query_log データベースパラメータの値を 1 に、log_output パラメータの値を FILE に更新する必要があります。

Lightsail コンソールで一般ログとスロークエリログを有効にするには

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 次のコマンドを入力して、general_log パラメータの値を 1 に更新します。これは true または有効です。

   ```
   aws lightsail update-relational-database-parameters --
   region Region --relational-database-name DatabaseName --parameters
    "parameterName=general_log,parameterValue=1,applyMethod=pending-reboot"
   ```

   コマンドを、以下のように置き換えます。

   - *DatabaseName* は、データベースの名前に置き換えます。
   - データベース AWS リージョン の を持つ*#####*。
3. 次のコマンドを入力して、slow_query_log パラメータの値を 1 に更新します。これは true または有効です。

   ```
   aws lightsail update-relational-database-parameters --
   region Region --relational-database-name DatabaseName --parameters
    "parameterName=slow_query_log,parameterValue=1,applyMethod=pending-reboot"
   ```

   コマンドを、以下のように置き換えます。

   - *DatabaseName* は、データベースの名前に置き換えます。
   - データベース AWS リージョン の を持つ*#####*。
4. 次のコマンドを入力して log_output パラメータの値を FILE に更新します。これによりログデータがシステムファイルに書き込まれ、Lightsail コンソールに表示されるようになります。

```
aws lightsail update-relational-database-parameters --
region Region --relational-database-name DatabaseName --parameters
 "parameterName=log_output,parameterValue=FILE,applyMethod=pending-reboot"
```

コマンドを、以下のように置き換えます。

- *DatabaseName* は、データベースの名前に置き換えます。
- データベース AWS リージョン の を持つ*#####*。

5. 次のコマンドを入力してデータベースを再起動し、変更を反映させます。

```
aws lightsail reboot-relational-database --region Region --relational-database-
name DatabaseName
```

コマンドを、以下のように置き換えます。

- *DatabaseName* は、データベースの名前に置き換えます。
- データベース AWS リージョン の を持つ*#####*。

この時点で、データベースは再起動中使用できなくなります。数分間待ってから Lightsail コンソールにサインインして、データベースの一般ログとスロークエリログを表示します。詳細については、「Amazon Lightsail でデータベースのログと履歴を表示する」を参照してください。

> **Note**
>
> データベースパラメータの更新の詳細については、「Amazon Lightsail でデータベースのパラメータを更新する」を参照してください。

データベースログのその他のオプションの制御

MySQL の一般ログとスロークエリログのその他のオプションを制御するには、次のパラメータを更新します。

- `log_output` — このパラメータは `TABLE` に設定します。一般クエリが `mysql.general_log` テーブルに書き込まれ、スロークエリは `mysql.slow_log` テーブルに書き込まれます。`log_output` パラメータを `NONE` に設定して、ログ記録を無効にすることもできます。

> **Note**
>
> log_output パラメータを TABLE に設定すると、一般ログとスロークエリログの Lightsail コンソールでの表示が無効になります。 ログデータを表示するには、代わりにデータベースの mysql.general_log および mysql.slow_log テーブルを参照する必要があります。

- long_query_time — ファストクエリがスロークエリログに記録されないようにするために、ログに記録されるクエリの最短実行時間の値を秒単位で指定します。デフォルトは 10 秒であり、最小値は 0 です。log_output パラメータが FILE に設定されている場合は、マイクロ秒の精度になるように、浮動小数点値を指定できます。log_output パラメータが TABLE に設定されている場合は、秒の精度になるように、整数値を指定する必要があります。実行時間が long_query_time パラメータの値を超えたクエリのみがログに記録されます。例えば、long_query_time を 0.1 に設定すると、実行時間が 100 ミリ秒未満のすべてのクエリはログに記録されなくなります。
- log_queries_not_using_indexes — インデックスを使用しないすべてのクエリをスロークエリログに記録するには、1 に設定します。デフォルトは 0 です。インデックスを使用しないクエリは、その実行時間が long_query_time パラメーターの値未満であっても口グに記録されます。

ログデータの保持

ログ記録が有効になっている場合、テーブルログのローテーションまたはログファイルの削除が定期的に実行されます。これは、ログファイルが大きくなることでデータベースが使用できなくなったりパフォーマンスに影響する可能性を低く抑えるための予防措置です。log_output パラメータが FILE または TABLE に設定されている場合、ログ記録は次のように処理されます。

- FILE ログ記録が有効になっている場合、ログファイルの検査が 1 時間ごとに実行され、作成後 24 時間を超えた古いログファイルは削除されます。場合によっては、削除後の残りのログファイルの合計サイズが、データベースに割り当てられた領域のしきい値である 2% を超えることがあります。この場合、ログファイルのサイズがしきい値以下になるまで、最も大きいログファイルから順に削除されます。
- TABLE ログ記録を有効化すると、24 時間ごとにログテーブルのローテーションが実行される場合があります。

このログテーブルのローテーションは、テーブルログに使用されている領域が、割り当てられたストレージ領域の 20 % を超えるか、すべてのログの合計サイズが 10 GB を超えると、実行されます。

データベースに使用されている領域が、データベースに割り当てられたストレージ領域の 90% を超えている場合は、ログのローテーションを実行するためのしきい値が小さくなります。

テーブルログに使用されている領域が、割り当てられたストレージ領域の 10% を超えるか、すべてのログの合計サイズが 5 GB を超えると、ログテーブルのローテーションが実行されます。

low_free_storage にサブスクライブして、ログテーブルのローテーションが実行されて領域が解放されたときに通知を受け取ることができます。

- ログテーブルのローテーションが実行されると、現在のログテーブルがバックアップのログテーブルにコピーされ、現在のログテーブル内にあるエントリは削除されます。バックアップのログテーブルが既に存在する場合は、現在のログテーブルをバックアップにコピーする前に、削除されます。バックアップのログテーブルは、照会することができます。mysql.general_log テーブルに対するバックアップのログテーブルは、mysql.general_log_backup という名前になります。mysql.slow_log テーブルに対するバックアップのログテーブルは、mysql.slow_log_backup という名前になります。
- mysql.general_log テーブルのローテーションは、mysql.rds_rotate_general_logprocedure を呼び出すことで実行できます。mysql.slow_log テーブルのローテーションは、mysql.rds_rotate_slow_logprocedure を呼び出すことで実行できます。
- データベースバージョンのアップグレード時にも、テーブルログのローテーションが実行されます。

Lightsail データベースのポイントインタイムバックアップを無効にする

Lightsail マネージド型データベースのポイントインタイムバックアップを無効にするには、次の手順に従います。

> **Important**
>
> ポイントインタイムバックアップを使用すると、データベースに障害が発生した場合にデータを簡単に復元できます。Lightsail マネージド型データベースのポイントインタイムバックアップを有効にしておくことをお勧めします。

前提条件

AWS Command Line Interface (AWS CLI)、または を使用して AWS CloudShell 、Lightsailデータベースのpoint-in-timeバックアップを有効または無効にします。CloudShell はブラウザーベースの事前認証済みシェルで、Lightsail コンソールから直接起動できます。詳細については「Lightsail オペレーション AWS CLI 用の をセットアップおよび設定する」および「でLightsailリソースを管理する AWS CloudShell」を参照してください。

データベースポイントインタイムバックアップを無効にする

Lightsail でマネージド型データベースのポイントインタイムバックアップを無効にするには、 AWS CLIの update-relational-database Lightsail コマンドを使用してデータベースを更新する必要があります。詳細については、「AWS CLI Command Reference」の「update-container-service」を参照してください。

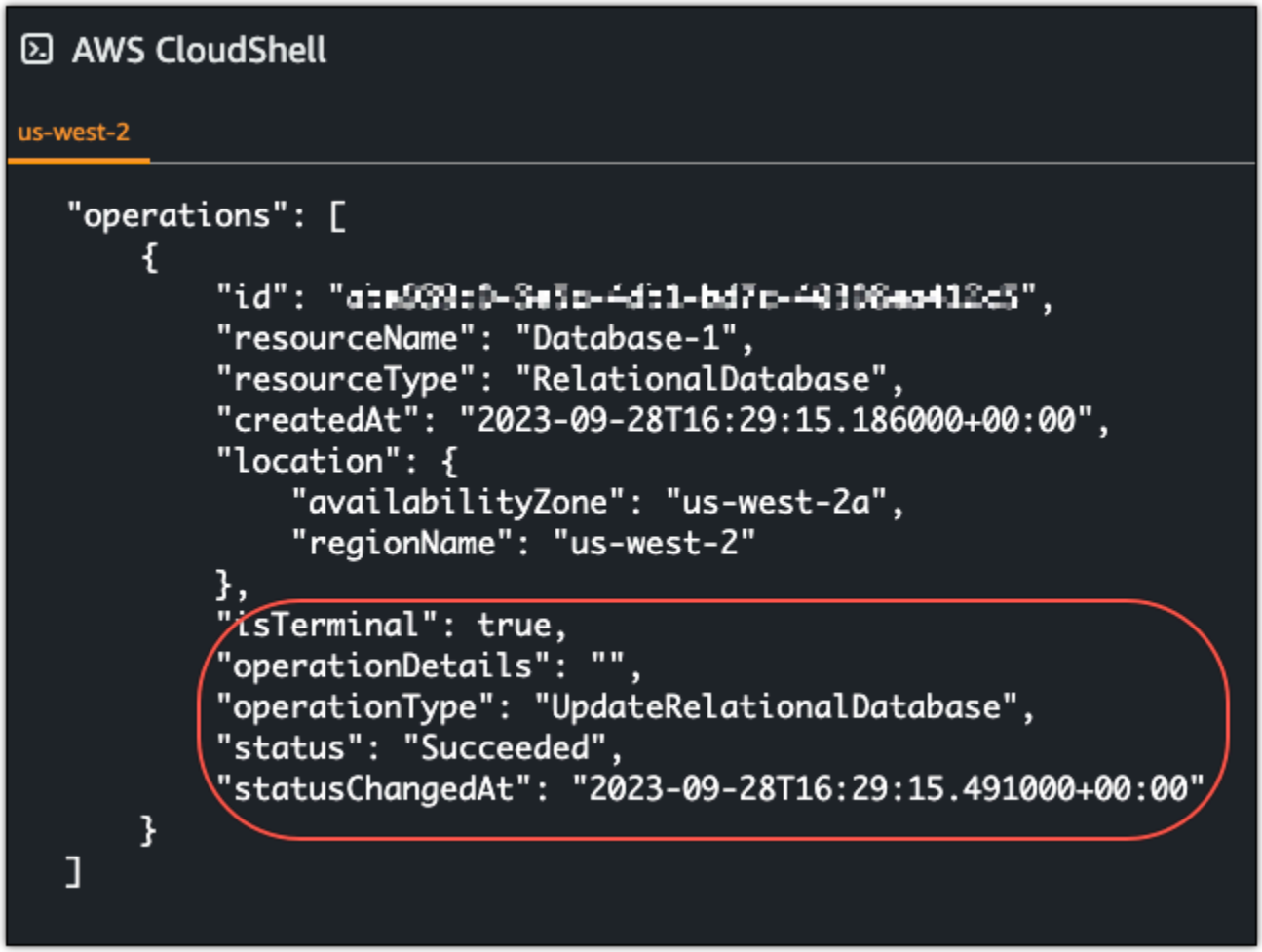
- ターミナル、コマンドプロンプトまたは CloudShell ウィンドウで、次のいずれかのコマンドを入力します:

  ```
  aws lightsail update-relational-database --region Region --relational-database-name DatabaseName --disable-backup-retention --apply-immediately
  ```

  コマンドの --disable-backup-retention 値は、指定されたデータベースのポイントインタイムバックアップをオフにします。コマンドを、以下のように置き換えます。

  - *DatabaseName* は、データベースの名前に置き換えます。
  - データベース AWS リージョン の を持つ*#####*。

Succeeded のステータスがオペレーションレスポンスで表示されるはずです。データベースのステータスは、更新された短期間に [変更中] に変わります。データベースのステータスが [利用可能] に戻ると、次の例に示すようにポイントインタイム復元オプションは無効になります。



> **Note**
>
> ポイントインタイムバックアップを有効にするには、前述のコマンドと同じコマンドを実行しますが、代わりに `--enable-backup-retention` パラメータを使用します。

スナップショットで Lightsail データベースをバックアップする

Amazon Lightsail でマネージドデータベースのスナップショットを作成できます。スナップショットは、問題が発生した場合にデータベースの復元に使用できるデータベースのコピーです。また、スナップショットを使用して、高可用性プランまたはスタンダードプランなどの別のプラン を使用する新しいデータベースを作成することもできます。

スタンダードデータベースのスナップショットを作成する場合、データベースのサイズに応じて、数秒から数分、データベースが使用不可になります。高可用性データベースの場合、スナップショット

はスタンバイデータベースを使用して作成されるため、スナップショットオペレーションによる影響はありません。

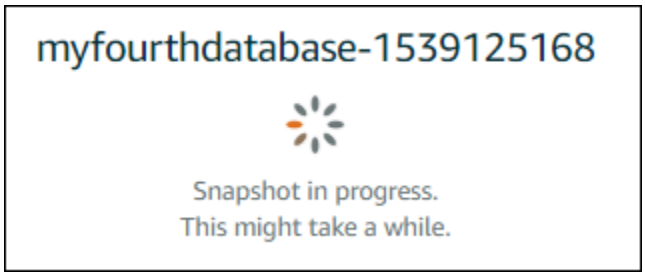
データベースのスナップショットを作成するには

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [データベース] を選択します。
3. スナップショットを作成するデータベースの名前を選択します。
4. [スナップショットと復元] タブを選択します。
5. このページの [手動スナップショット] セクションで、[スナップショットの作成] を選択し、スナップショットの名前を入力します。

   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。
6. [Create(作成)] を選択します。

   スナップショットの作成プロセスが開始され、ステータスとして [スナップショットを作成中です] と表示されます。



   スナップショットの作成プロセスが完了すると、新しいスナップショットが [最近のスナップショット] セクションに表示されます。アカウントのすべてのスナップショットは、Lightsail のホームページの [スナップショット] タブでも確認できます。

次の手順

スナップショットの準備が完了したら、スナップショットから新しいデータベース (元のデータベースの複製) を作成できます。詳細については、「スナップショットからデータベースを作成する」を参照してください。

トピック

- Lightsail で特定の時点のバックアップからデータベースを復元する
- Lightsail でスナップショットからマネージドデータベースを作成する

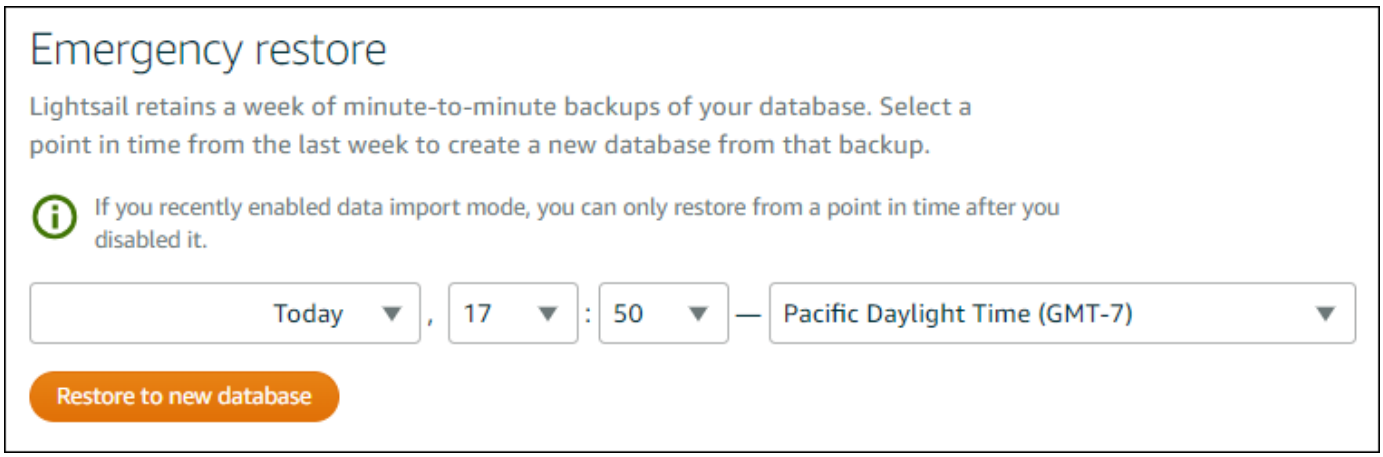
Lightsail で特定の時点のバックアップからデータベースを復元する

Amazon Lightsail で特定の時点のバックアップを使用して、新しいマネージドデータベースを作成できます。特定の時点のデータベースのバックアップは、5 分間単位で、過去を遡って 7 日分、利用できます。これにより、障害が発生したデータベースを過去 1 週間前までの特定の時点に復旧できます。

スナップショットから新しいデータベースを作成することもできます。詳細については、Amazon Lightsail でスナップショットからデータベースを作成するを参照してください。

特定の時点のバックアップからデータベースを作成するには

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [データベース] を選択します。
3. プランを変更するデータベースの名前を選択します。
4. [Snapshots and restore (スナップショットおよび復元)] タブを選択します。
5. [Emergency restore (緊急復元)] セクションで、新しいデータベースに使用するバックアップの日時を選択します。



6. [Restore to new database (新しいデータベースに復元)] を選択します。
7. [新しいデータベースを作成] ページで、[ゾーンの変更] を選択して別のアベイラビリティーゾーンを選択します。前に選択したスナップショットと同じ AWS リージョンに新しいデータベースが作成されます。
8. 新しいデータベースプランを選択します。

   高可用性データベースプランまたはスタンダードデータベースプランを選択します。高可用性プランでデータベースを作成すると、プライマリデータベースのほかに、フェイルオーバーのサポートとしてスタンバイ用のセカンダリデータベースが別のアベイラビリティーゾーンに作成されます。詳細については、「高可用性データベース」を参照してください。

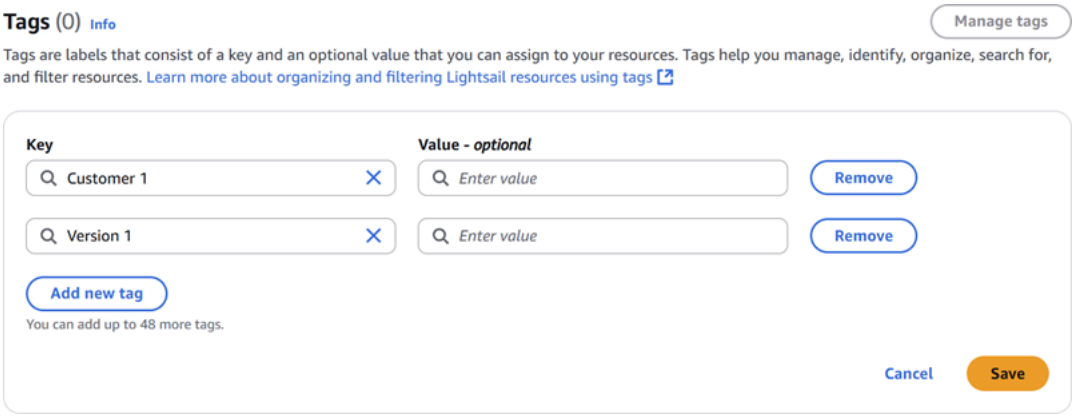
   > **Note**
   >
   > 元のデータベースプランより小さいデータベースプランを選択することはできません。

9. データベースの名前を入力します。

   リソース名:

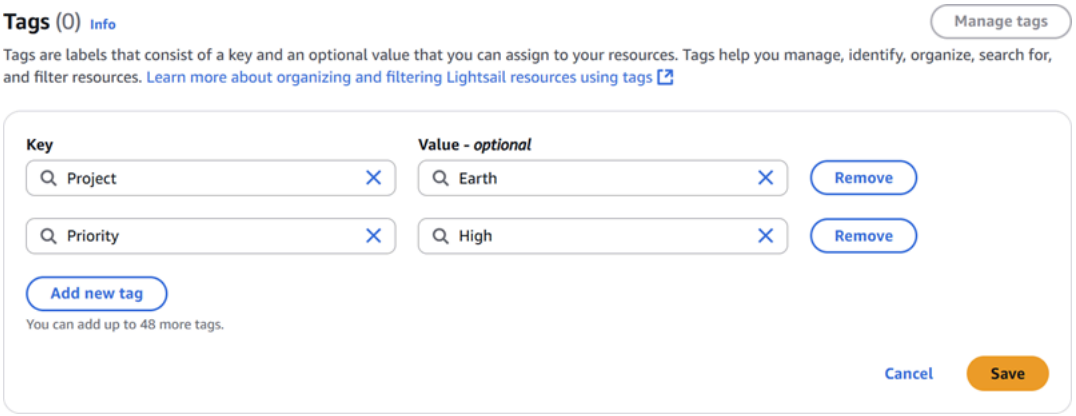
   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。
10. 以下のいずれかのオプションを選択して、データベースにタグを追加します。

- [キーオンリータグの追加] または [タグを編集] (タグが追加済みの場合)。タグキーのテキストボックスに新しいタグを入力し、Enter キーを押します。タグの入力を完了したら、[保存] を選択してタグを追加し、追加しない場合は、[キャンセル] を選択します。



- [Create a key-value tag] (キーバリュータグを作成) から [Key] (キー) テキストボックスにキーを入力し、[Value] (値) テキストボックスに値を入力します。タグの入力を完了したら、[保存] を選択し、追加しない場合は、[キャンセル] を選択します。

  キーバリューのタグは、保存する際に一つずつ追加することができます。さらにキーバリュータグを追加するには、以上のステップを繰り返します。



> **Note**
>
> 「キーのみ」のタグと「キーバリュー」のタグの詳細については、「タグ」を参照してください。

11. [データベースの作成] を選択します。

数分以内に、新しい Lightsail のデータベースが新しいデータベースプランまたはバンドルとして使用可能になります。

次の手順

新しいデータベースが使用可能になったら、次のアクションを実行します。

- 元のデータベースが不要な場合は、削除できます。詳細については、「データベースを削除する」を参照してください。
- 特定の時点のバックアップから作成したデータベースは、Lightsail で作成された強力なパスワードを使用するように設定します。詳細については、「データベースのパスワードを管理する」を参照してください。

Lightsail でスナップショットからマネージドデータベースを作成する

元のデータベースで障害が発生した場合は、Amazon Lightsail でスナップショットから新しいマネージドデータベースを作成できます。また、データベースを別のプラン (高可用性プランまたはスタンダードプラン) に変更することもできます。さらに、元のデータベースの特定時点のバックアップから新しいデータベースを作成することもできます。詳細については、「Amazon Lightsail で特定の時点のバックアップからデータベースを作成する」を参照してください。

データベースを複製する際は、元のデータベースとは異なるプランやよりサイズの大きなプランを選択できます。ただし、元のデータベースよりサイズの小さいプランを選択することはできません。

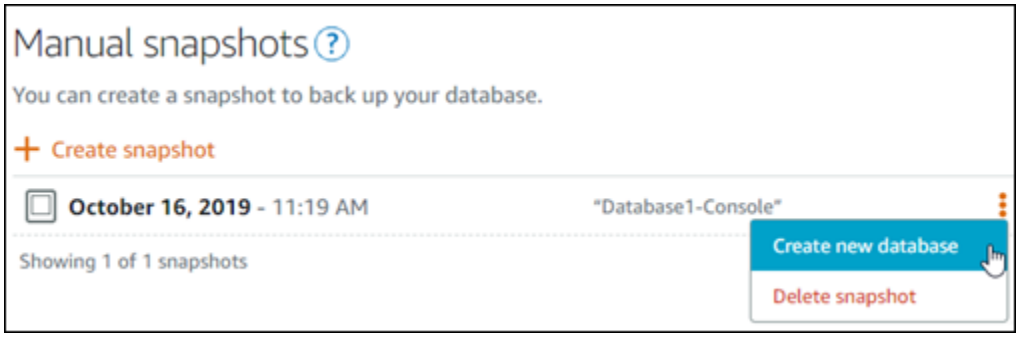
> **Note**
>
> 高可用性プランでデータベースを作成すると、プライマリデータベースのほかに、フェイルオーバーのサポートとしてスタンバイ用のセカンダリデータベースが別のアベイラビリティーゾーンに作成されます。詳細については、「高可用性データベース」を参照してください。

スナップショットからデータベースを作成するには

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [データベース] を選択します。

3. スナップショットから新しいデータベースを作成することによって複製する元のデータベースの名前を選択します。
4. [スナップショットと復元] タブを選択します。
5. このページの [手動スナップショット] セクションで、新しいデータベースを作成するスナップショットの横にあるアクションメニューアイコン (⋮) を選択してから、[Create new database (新しいデータベースの作成)] を選択します。

> **Note**
>
> データベースの既存のスナップショットが必要となります。スナップショットをまだ作成していない場合は、「データベースのスナップショットを作成する」を参照してください。



6. [Create new database (新しいデータベースの作成)] を選択します。
7. [新しいデータベースを作成] ページで、[ゾーンの変更] を選択して別のアベイラビリティーゾーンを選択します。前に選択したスナップショットと同じ AWS リージョンに新しいデータベースが作成されます。
8. 新しいデータベースプランを選択します。

   高可用性データベースプランまたはスタンダードデータベースプランを選択します。高可用性プランでデータベースを作成すると、プライマリデータベースのほかに、フェイルオーバーのサポートとしてスタンバイ用のセカンダリデータベースが別のアベイラビリティーゾーンに作成されます。詳細については、「高可用性データベース」を参照してください。

> **Note**
>
> スナップショットの作成に使用した元のデータベースのプランよりサイズの小さいデータベースプランを選択することはできません。

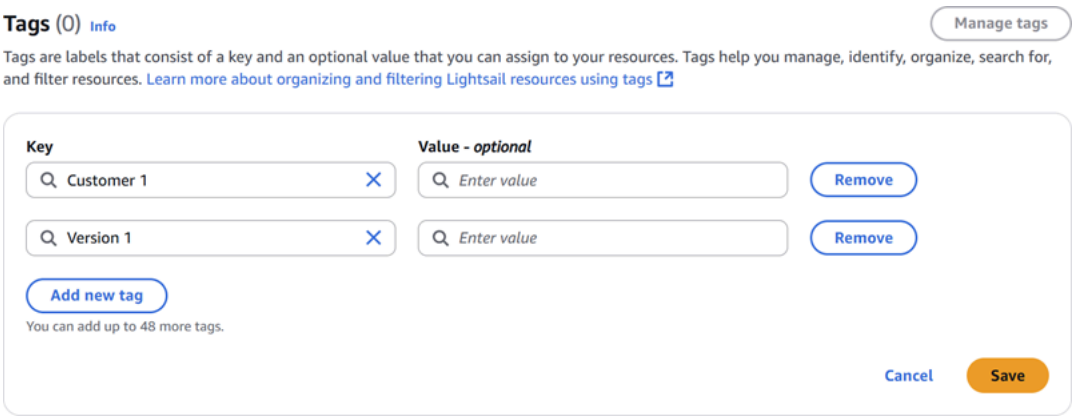
9. データベースの名前を入力します。

   リソース名:

   - Lightsail アカウントの各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

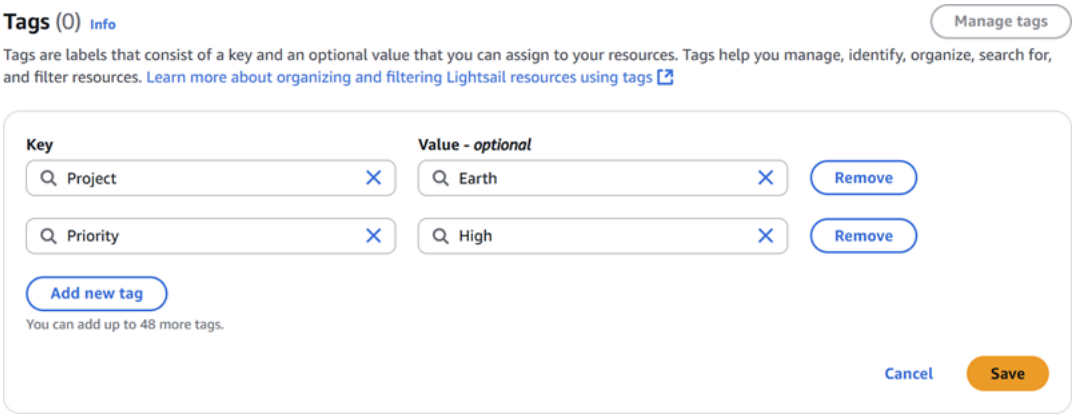
10. 以下のいずれかのオプションを選択して、データベースにタグを追加します。

    - [キーオンリータグの追加] または [タグを編集] (タグが追加済みの場合)。タグキーのテキストボックスに新しいタグを入力し、Enter キーを押します。タグの入力を完了したら、[保存] を選択してタグを追加し、追加しない場合は、[キャンセル] を選択します。

    - [Create a key-value tag] (キーバリュータグを作成) から [Key] (キー) テキストボックスにキーを入力し、[Value] (値) テキストボックスに値を入力します。タグの入力を完了したら、[保存] を選択し、追加しない場合は、[キャンセル] を選択します。

      キーバリューのタグは、保存する際に一つずつ追加することができます。さらにキーバリュータグを追加するには、以上のステップを繰り返します。



> **Note**
>
> 「キーのみ」のタグと「キーバリュー」のタグの詳細については、「タグ」を参照してください。

11. [データベースの作成] を選択します。

    数分以内に、新しい Lightsail のデータベースが新しいデータベースプランまたはバンドルとして使用可能になります。

次のステップ

新しいデータベースが使用可能になったら、次のアクションを実行します。

- 新しいデータベースを作成して既存のデータベースを置き換える場合、既存のデータベースに依存するアプリケーションがあるときは、アプリケーションの依存関係を新しいデータベースに必ず更新します。
- 元のデータベースが不要な場合は、削除できます。詳細については、「データベースを削除する」を参照してください。
- スナップショットから作成したデータベースは、Lightsail で作成された強力なパスワードを使用するように設定します。詳細については、「データベースのパスワードを管理する」を参照してください。

Lightsail データベースへの安全なアプリケーション接続のための SSL/TLS 証明書をダウンロードする

アプリケーションの Secure Socket Layer (SSL) または Transport Layer Security (TLS) を使用して、MySQL または PostgreSQL を実行している Amazon Lightsail のマネージド型データベースへの接続を暗号化できます。各 DB エンジンには SSL/TLS を実装する独自のプロセスがあります。詳細については、「SSL を使用した MySQL データベースの接続」または「SSL を使用した PostgreSQL データベースの接続」を参照してください。

> **Note**
>
> ダウンロード可能な証明書には Amazon Relational Database Service (Amazon RDS) のラベルが付いていますが、Lightsail のマネージド型データベースにも使用できます。

すべての AWS リージョンの証明書バンドル

すべての の中間証明書とルート証明書の両方を含む証明書バンドルを取得する場合、またはアプリケーションが Microsoft Windows にあり PKCS7 ファイルが必要な場合は、 AWS リージョン Amazon Relational Database Service ユーザーガイド」の「すべての AWS リージョンの証明書バンドル」を参照してください。

このルート証明書は信頼されたルートエンティティであり、ほとんどの場合は使用することができます。ただし、アプリケーションが証明書チェーンを受け入れていない場合は使用できない場合があります。アプリケーションが証明書チェーンを受け入れていない場合、このドキュメントの次のセクションに進みます。

特定の AWS リージョンの証明書バンドル

特定の の中間証明書とルート証明書の両方を含む証明書バンドルを取得するには AWS リージョン、「Amazon Relational Database Service ユーザーガイド」の「特定の AWS リージョンの証明書バンドル」を参照してください。 Amazon Relational Database Service

Lightsail データベースの CA 証明書バージョンを更新する

Amazon Lightsail で、SSL/TLS を使用してマネージド型データベースに接続するための新しい認証機関 (CA) 証明書が公開されました。このガイドでは、新しい CA 証明書にアップグレードする方法について説明します。update-relational-database API アクションを使用してのみ証明書をアップグ

レードできます。新しい証明書は、rds-ca-rsa2048-g1、rds-ca-rsa4096-g1、および rds-ca-ecc384-g1 と呼ばれます。古い証明書は rds-ca-2019 と呼ばれます。 AWS セキュリティのベストプラクティスとして CA 証明書を提供しています。マネージド型データベースの CA 証明書、およびサポートされている AWS リージョンの詳細については、「マネージドデータベースの SSL 証明書のダウンロード」を参照してください。

古い CA 証明書 (rds-ca-2019) の有効期限は 2024 年 8 月 22 日です。したがって、このガイドの手順をできる限り早く完了して、新しい証明書を使用するようにマネージド型データベースを変更することを強くお勧めします。アプリケーションが SSL/TLS を使用して Lightsail のマネージド型データベースに接続していない場合、対処は必要ありません。これらの手順が完了していない場合、2024 年 8 月 22 日以降、アプリケーションは SSL/TLS を使用したマネージド型データベースへの接続に失敗します。

2024 年 1 月 26 日以降に作成された新しいマネージド型データベースは、デフォルトで rds-ca-rsa2048-g1 証明書を使用します。古い証明書 (rds-ca-2019) を使用するように新しいマネージド型データベースを一時的に変更する場合、そのためには、 AWS Command Line Interface (AWS CLI) を使用します。2024 年 1 月 26 日より前に作成されたマネージド型データベースは、rds-ca-2019 証明書に更新するまでrds-ca-rsa2048-g1、rds-ca-rsa4096-g1、rds-ca-ecc384-g1 証明書を使用します。

> **Note**
>
> このガイドの手順は、本番稼働用環境で使用する前に、開発環境またはステージング環境でテストしてください。

前提条件

- 以下の手順のステップを実行する前に、新しい SSL/TLS 証明書を使用するようにデータベースアプリケーションを更新してください。

  新しい SSL/TLS 証明書のアプリケーションを更新する方法は、特定のアプリケーションにより異なります。アプリケーション開発者と協力して、アプリケーションの SSL/TLS 証明書を更新します。新しい SSL/TLS 証明書のアプリケーション更新の詳細については、「Amazon Relational Database Service ユーザーガイド」の「新しい SSL/TLS 証明書の使用によってアプリケーションを更新して MySQL DB インスタンスに接続」または「新しい SSL/TLS 証明書の使用によってアプリケーションを更新して PostgreSQL DB インスタンスに接続」を参照してください。

- このガイドでは、 AWS CloudShell を使用してアップグレードを実行します。CloudShell はブラウザーベースの事前認証済みシェルで、Lightsail コンソールから直接起動できます。CloudShell では、Bash、PowerShell、Z シェルなどの任意のシェルを使用して AWS Command Line Interface (AWS CLI) コマンドを実行できます。この手順は、コマンドラインツールのダウンロードもインストールも不要です。CloudShell をセットアップして使用する方法の詳細については、Lightsail のAWS CloudShell を参照してください。

マネージド型データベースのアクティブな CA 証明書を特定する

Lightsail データベースインスタンスのアクティブな CA 証明書を識別するには、次の手順を実行します。

1. ターミナル「AWS CloudShell」またはコマンドプロントウィンドウを開きます。
2. マネージド型データベースでアクティブな CA 証明書を使用するには、以下のコマンドを入力します。

   ```
   aws lightsail get-relational-database --relational-database-name DatabaseName --
   region DatabaseRegion | grep "caCertificateIdentifier"
   ```

   コマンドにて、*DatabaseName* を変更したいデータベースの名前に置き換え、*DatabaseRegion* をデータベースインスタンスが存在する AWS リージョン に置き換えます。

   例

   ```
   aws lightsail get-relational-database --relational-database-name Database-1 --
   region us-east-1 | grep "caCertificateIdentifier"
   ```

   コマンドは、データベースのアクティブな CA 証明書の ID を返します。

   例

   ```
   "caCertificateIdentifier": "rds-ca-rsa2048-g1"
   ```

新しい CA 証明書を使用するためにマネージド型データベースを変更する

新しい CA 証明書の 1 つ (rds-ca-rsa2048-g1、rds-ca-rsa4096-g1 と rds-ca-ecc384-g1) を使用するように Lightsail のマネージド型データベースを変更するには、以下の手順を実行します。

> **Important**
>
> データベースの CA 証明書を更新する前に、CA 証明書を使用するクライアントアプリケーションを更新します。

1. ターミナル「AWS CloudShell」またはコマンドプロントウィンドウを開きます。
2. マネージド型データベースで新しい証明書を使用するには、以下のコマンドを入力します。

```
aws lightsail update-relational-database --relational-database-name DatabaseName --region DatabaseRegion --ca-certificate-identifier rds-ca-rsa2048-g1
```

コマンドにて、*DatabaseName* を変更したいデータベースの名前に置き換え、*DatabaseRegion* をデータベースインスタンスが存在する AWS リージョン に置き換えます。

例

```
aws lightsail update-relational-database --relational-database-name Database-1 --region us-east-1 --ca-certificate-identifier rds-ca-rsa2048-g1
```

マネージド型データベースで使用される CA 証明書は、次回のデータベースメンテナンス期間中に更新されますが、コマンドの末尾に --apply-immediately パラメーターを追加すると即座に更新されます。

古い CA 証明書を使用するためにマネージド型データベースを変更する

古い CA 証明書 (rds-ca-2019) を使用するように Lightsail のマネージド型データベースを変更するには、以下の手順を実行します。新しい証明書 (rds-ca-rsa2048-g1、rds-ca-rsa4096-g1 と rds-ca-ecc384-g1) で重大な問題が発生し、古い証明書に一時的に戻す必要がある場合のみでこれを行ってください。

> **Important**
>
> データベースの CA 証明書を更新する前に、CA 証明書を使用するクライアントアプリケーションを更新します。

1. ターミナル「AWS CloudShell」またはコマンドプロントウィンドウを開きます。
2. マネージド型データベースで `rds-ca-2019` を使用するには、以下のコマンドを入力します。

```
aws lightsail update-relational-database --relational-database-name DatabaseName --region DatabaseRegion --ca-certificate-identifier rds-ca-2019
```

コマンドにて、*DatabaseName* を変更したいデータベースの名前に置き換え、*DatabaseRegion* をデータベースインスタンスが存在する AWS リージョン に置き換えます。

例

```
aws lightsail update-relational-database --relational-database-name Database-1 --region us-east-1 --ca-certificate-identifier rds-ca-2019
```

マネージド型データベースで使用される CA 証明書は、次回のデータベースメンテナンス期間中に更新されますが、コマンドの末尾に `--apply-immediately` パラメーターを追加すると即座に更新されます。

Lightsail データベースのメンテナンスとバックアップをスケジュールする

Amazon Lightsail で新バージョンのデータベースのサポートを開始した場合、既存のマネージドデータベースを新バージョンにアップグレードできます。アップグレードには、メジャーバージョンのアップグレードとマイナーバージョンのアップグレードの 2 種類があります。現在、Lightsail はマイナーバージョンのアップグレードのみサポートしています。

マイナーバージョンアップグレードおよび他のデータベースメンテナンスタスクは、データベースのメンテナンスウィンドウ中に自動的に実行されます。推奨されるメンテナンスウィンドウは、それぞれ 8 時間の時間ブロックからランダムに選択された 30 分のウィンドウです AWS リージョン。このウィンドウはランダムな曜日に発生します。データベースのバックアップは、バックアップウィンド

ウ中に実行されます。優先バックアップウィンドウは、それぞれ 8 時間の時間ブロックからランダムに選択された 30 分のウィンドウです AWS リージョン。このウィンドウもランダムな曜日に発生します。

> **Note**
>
> リージョン別のデフォルトメンテナンスウィンドウの時間ブロックの詳細については、Amazon Relational Database Service (Amazon RDS) ドキュメントの「DB インスタンスのメンテナンス」ガイドを参照してください。リージョン別のデフォルトバックアップウィンドウの時間ブロックの詳細については、Amazon RDS ドキュメントの「バックアップの使用」ガイドを参照してください。

このガイドでは、メンテナンスおよびバックアップウィンドウを、データベースの負荷が最も低い時間帯に変更する方法を示します。

前提条件

データベースの優先メンテナンスおよびバックアップウィンドウを変更するには、 AWS Command Line Interface (AWS CLI) を使用する必要があります。

以下の前提条件を満たしてください。

- のインストール AWS CLI — 詳細については、AWS CLI「I のインストール」を参照してください。
- の設定 AWS CLI — 詳細については、「 の設定 AWS CLI」を参照してください。

データベースのメンテナンスウィンドウを変更する

データベースは、メンテナンスまたはバックアップオペレーション中は利用できない場合があります。したがって、メンテナンスまたはバックアップウィンドウを、データベースの負荷が最も低い時間に変更する必要が生じる場合があります。

データベースのメンテナンスウィンドウを変更するには

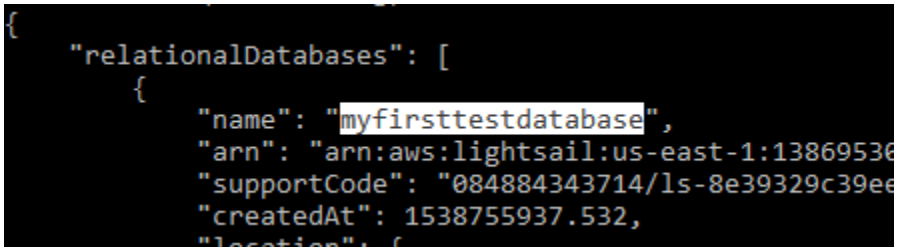
1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 次のコマンドを入力して、メンテナンスウィンドウを変更するデータベースの名前を取得します。

```
aws lightsail get-relational-databases
```

以下の例のような結果が表示されるはずです。

> **Note**
>
> 変更するデータベースが表示されない場合は、 AWS CLI がデータベース AWS リージョン がある 用に設定されていることを確認します。詳細については、「AWS CLIの設定」を参照してください。

3. 変更するデータベースの名前を強調表示し、Ctrl+C (Windows) または Cmd+C (macOS) を押して、クリップボードにコピーします。これを次のステップで使用します。



4. 変更するウィンドウに応じて、以下のいずれかのコマンドを入力します。

   - 次のコマンドを入力し、データベースのメンテナンスウィンドウを変更します。

     ```
     aws lightsail update-relational-database --relational-database-name DatabaseName
      --preferred-maintenance-window MaintenanceWindow
     ```

     コマンドを、以下のように置き換えます。

     - *DatabaseName* は、データベースの名前に置き換えます。
     - *MaintenanceWindow* は、新しいメンテナンスウィンドウの時間枠に置き換えます。

       メンテナンスウィンドウの時間を ddd:hh24:mi-ddd:hh24:mi 形式で定義します。また、協定世界時 (UTC) 形式で指定し、最低 30 分のウィンドウとして定義する必要があります。メンテナンスウィンドウは、バックアップウィンドウと重複できません。

     例:

     ```
     aws lightsail update-relational-database --relational-database-
     name myproductiondb --preferred-maintenance-window Tue:16:00-Tue:16:30
     ```

   - 次のコマンドを入力し、データベースのバックアップウィンドウを変更します。

     ```
     aws lightsail update-relational-database --relational-database-name DatabaseName
      --preferred-backup-window BackupWindow
     ```

     コマンドを、以下のように置き換えます。

     - *DatabaseName* は、データベースの名前に置き換えます。
     - *BackupWindow* は、新しいバックアップウィンドウの時間枠に置き換えます。

       バックアップウィンドウの時間を hh24:mi-hh24:mi 形式で定義します。また、協定世界時 (UTC) 形式で指定し、最低 30 分のウィンドウとして定義する必要があります。バックアップウィンドウは、メンテナンスウィンドウと重複できません。

例:

```
aws lightsail update-relational-database --relational-database-
name myproductiondb --preferred-backup-window 14:00-14:30
```

以下の例のような結果が表示されるはずです。

次の手順

データベースの管理に役立つ以下のガイドを参照してください。

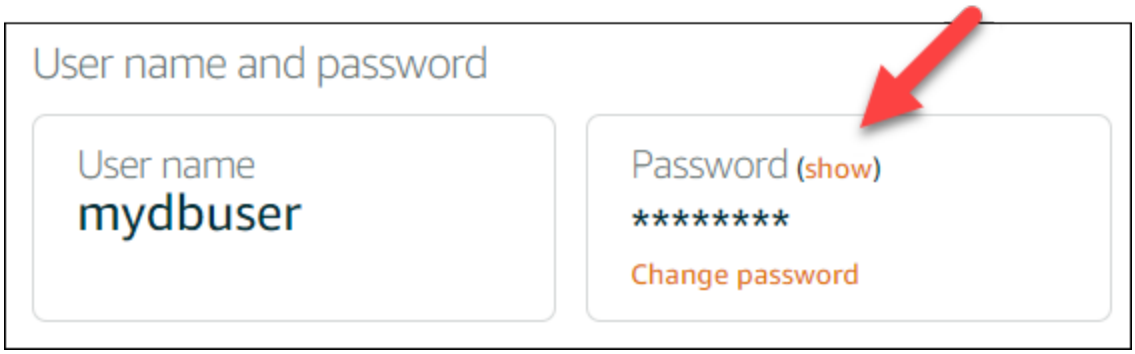
- データベースのデータのインポートモードを設定する
- データベースのパブリックモードを設定する
- データベースのパスワードを管理する
- MySQL データベースに接続する
- PostgreSQL データベースに接続する
- MySQL データベースにデータをインポートする
- データを PostgreSQL データベースにインポートする
- データベースのスナップショットを作成する

Lightsail データベースのパスワードを変更する

Amazon Lightsail で新しいデータベースを作成する場合、Lightsail で自動的に強力なパスワードを作成するか、独自にパスワードを指定できます。現在のデータベースのパスワードは、Lightsail コンソールでいつでも表示または変更できます。

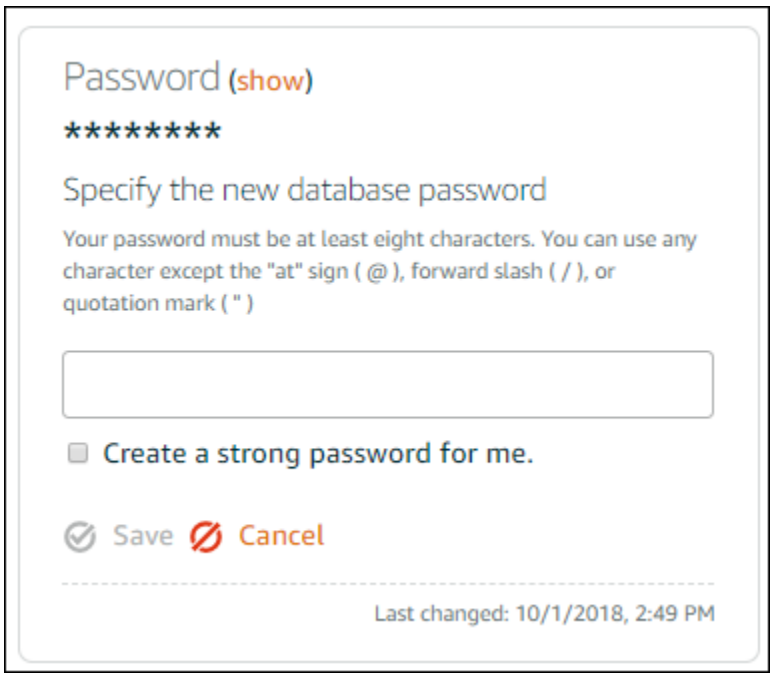
データベースのパスワードを管理するには

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [データベース] を選択します。
3. パスワードを管理するデータベースの名前を選択します。
4. [接続] タブの [User name and passwords (ユーザー名とパスワード)] セクションで、[表示] を選択して現在のデータベースパスワードを表示します。



5. データベースのパスワードを変更するには、[パスワードの変更] を選択します。

   Lightsail で自動的に強力なパスワードを作成するか、独自のパスワードをテキストボックスに入力するかを選択できます。パスワードには「/」「"」または「@」を除く表示可能な任意の ASCII 文字を使用することができます。MySQL データベースの場合、パスワードには 8 ~ 41 文字の英数字を使用する必要があります。PostgreSQL の場合、パスワードには 8 ~ 128 文字の英数字を使用する必要があります。



6. 完了したら、[保存] を選択します。

   データベースのパスワードの変更はすぐに適用されます。独自のパスワードを入力した場合、パスワードはすぐに保存されます。Lightsail で自動的にパスワードを作成した場合は、数秒以内に生成されます。新しいパスワードを表示するには、[表示] を選択します。

次の手順

Lightsail でのデータベースの管理に役立つ以下のガイドを参照してください。

- MySQL データベースに接続する
- PostgreSQL データベースに接続する
- データベースのスナップショットを作成する

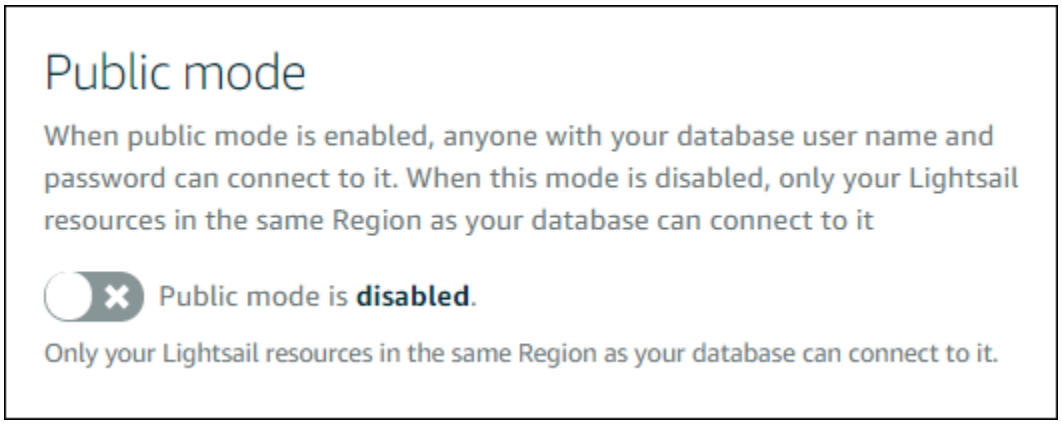
Lightsail データベースのパブリックアクセスを設定する

Amazon Lightsail のマネージドデータベースにアクセスできるのは、同じ Lightsail アカウント内にある Lightsail リソース (インスタンス、ロードバランサーなど) のみです。1 つの一般的なシナリオでは、一般向けのウェブアプリケーションを持つ Lightsail インスタンスと、パブリックアクセス可能ではない Lightsail データベースの両方を作成し、この 2 つを接続します。

データベースをパブリックアクセス可能にするには、パブリックモード機能を有効にします。これにより、すべてのユーザーがデータベースエンドポイント、ポート、ユーザー名、およびパスワードを使用してデータベースに接続できます。詳細については、「MySQL データベースに接続する」または「PostgreSQL データベースに接続する」を参照してください。

データベースのパブリックモードを設定するには

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [データベース] を選択します。
3. パブリックモードを設定するデータベースの名前を選択します。
4. [ネットワーキング] タブを選択します。
5. [Public mode (パブリックモード)] セクションで、トグルを使用してパブリックモードをオンにします。これをオフにする場合も、トグルを使用します。



パブリックアクセシビリティの設定の適用が即座に開始されますが、完了するまでに数分かかることがあります。この間に、データベースのステータスは [変更中] に変わります。パブリックアクセシビリティの設定が適用されると、データベースのステータスは [利用可能] に変わります。

次のステップ

データベースの管理に役立つ以下のガイドを参照してください。

- データベースのデータのインポートモードを設定する
- データベースのパスワードを管理する
- MySQL データベースに接続する
- PostgreSQL データベースに接続する

- MySQL データベースにデータをインポートする
- データを PostgreSQL データベースにインポートする
- データベースのスナップショットを作成する

パラメータの更新による Lightsail データベースのパフォーマンス最適化

データベースのパラメータ (データベースのシステム変数とも呼ばれます) は、Amazon Lightsail のマネージドインスタンスの基本的なプロパティを定義します。たとえば、データベース接続の数を制限するデータベースのパラメータを定義したり、データベースのバッファプールサイズを制限する別のパラメータを定義したりできます。このガイドでは、マネージドデータベースのパラメータのリストを取得する方法と、 AWS Command Line Interface () を使用してパラメータを更新する方法について説明しますAWS CLI。

> **Note**
>
> MySQL のシステム変数の詳細については、MySQL 5.6、MySQL 5.7、または MySQL 8.0 のドキュメントをご覧ください。PostgreSQL システム変数の詳細については、PostgreSQL 9.6、PostgreSQL 10、PostgreSQL 11、PostgreSQL 12 のドキュメントを参照してください。

前提条件

- まだ AWS CLIをインストールして設定していない場合は、インストールして設定します。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

使用可能なデータベースのパラメータのリストを取得します。

データベースのパラメータは、データベースエンジンによって異なります。そのため、使用しているマネージドデータベースに応じたパラメータのリストを取得する必要があります。これにより、どのパラメータを変更し、どのような方法でパラメータを有効にするかを決定できます。

使用可能なデータベースのパラメータのリストを取得するには

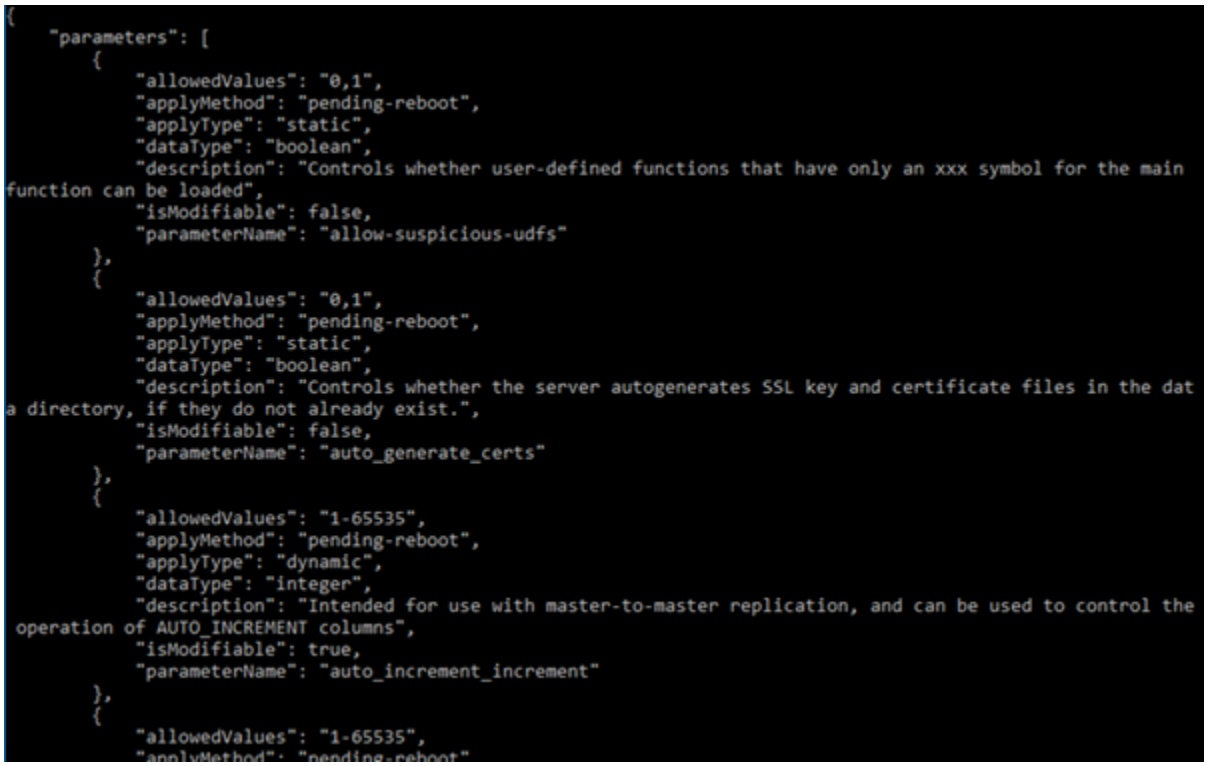
1. ターミナルまたはコマンドプロントウィンドウを開きます。

2. 以下のコマンドを入力して、データベースのパラメータのリストを取得します。

```
aws lightsail get-relational-database-parameters --relational-database-
name DatabaseName
```

コマンドで、*DatabaseName* をデータベースの名前に置き換えます。

以下の例のような結果が表示されるはずです。



> **Note**
>
> パラメータの結果がページ分割される場合は、「次ページトークン ID」が表示されます。この次ページトークン ID を書き留め、表示されたとおりに次のステップで使用して、パラメータ結果の次のページを表示します。

3. 結果がページ分割されている場合は、次のコマンドを使用して追加のパラメータセットを表示します。それ以外の場合は、次のステップに進みます。

```
aws lightsail get-relational-database-parameters --relational-database-
name DatabaseName --page-token NextPageTokenID
```

コマンドを、以下のように置き換えます。

- *DatabaseName* は、データベースの名前に置き換えます。
- *NextPageTokenID* は、次のページトークン ID に置き換えます。

結果には、データベースのパラメータごとに以下の情報が表示されます。

- 使用できる値 — パラメータの有効な値の範囲を指定します。
- 適用方法 — パラメータの変更を適用するタイミングを指定します。使用できるオプションは、immediate または pending-reboot です。適用方法の定義の詳細については、次の「適用タイプ」を参照してください。
- 適用タイプ — エンジン固有の送信タイプを指定します。dynamic が表示された場合は、適用方法として immediate を使用してパラメータを適用できます。データベースは新しいパラメータ値の使用を即座に開始します。static が表示された場合は、パラメータの適用方法として pending-reboot のみ使用できます。データベースは新しいパラメータ値の使用を再起動後にのみ開始します。
- データ型 — パラメータの有効なデータ型を指定します。
- 説明 — パラメータの説明です。
- 変更可能 — パラメータが変更可能であるかどうかを示すブール値。true が表示された場合、パラメータは変更可能です。
- パラメータ名 — パラメータの名前を指定します。この値は update relational database オペレーションおよび parameter name パラメータと組み合わせて使用します。

4. 変更するパラメータを検索し、パラメータ名、使用できる値、および適用方法を書き留めます。間違えて入力しないように、パラメータ名をクリップボードにコピーすることをお勧めします。これを行うには、パラメータ名を強調表示し、Ctrl+C (Windows) または Cmd+C (macOS) を押してクリップボードにコピーします。次に、Ctrl+V または Cmd+V を押して貼り付けます。

   変更するパラメータの名前を確認したら、このガイドの次のセクションに進み、パラメータを目的の値に変更します。

データベースのパラメータを更新する

変更するパラメータの名前を確認したら、次のステップを実行して Lightsail のマネージドデータベースのパラメータを変更し ます。

データベースのパラメータを更新するには

- 次のコマンドをターミナルまたはコマンドプロンプトウィンドウに入力し、マネージドデータベースのパラメータを更新します。

```
aws lightsail update-relational-database-parameters
 --relational-database-name DatabaseName --parameters
 "parameterName=ParameterName,parameterValue=NewParameterValue,applyMethod=ApplyMethod"
```

コマンドを、以下のように置き換えます。

- *DatabaseName* は、データベースの名前に置き換えます。
- *ParameterName* は、変更するパラメータの名前に置き換えます。
- *NewParameterValue* は、パラメータの新しい値に置き換えます。
- *ApplyMethod* は、パラメータの適用方法に置き換えます。

  パラメータの適用タイプが dynamic である場合は、適用方法として immediate を使用してパラメータを適用できます。データベースは新しいパラメータ値の使用を即座に開始します。ただし、パラメータの適用タイプが static である場合は、パラメータの適用方法として pending-reboot のみ使用できます。データベースは新しいパラメータ値の使用を再起動後にのみ開始します。

以下の例のような結果が表示されるはずです。


```
{
    "operations": [
        {
            "id": "2c650987-11e8-463f-94d5-0c15aacaf12b",
            "resourceName": "myfirsttestdatabase",
            "resourceType": "RelationalDatabase",
            "createdAt": 1539204831.214,
            "location": {
                "availabilityZone": "us-east-1a",
                "regionName": "us-east-1"
            },
            "isTerminal": true,
            "operationType": "UpdateRelationalDatabaseParameters",
            "status": "Succeeded",
            "statusChangedAt": 1539204831.214
        }
    ]
}
```

データベースのパラメータは、使用される適用方法に応じて更新されます。

Lightsail データベースのメジャーバージョンをアップグレードする

Amazon Lightsail がデータベースエンジンの新しいバージョンをサポートしている場合、データベースを新しいバージョンにアップグレードできます。Lightsail には、2 つのデータベースブループリント、MySQL、PostgreSQL が提供されます。このガイドでは、MySQL または PostgreSQL データベースインスタンスのメジャーバージョンをアップグレードする方法について説明します。データベースメジャーバージョンは、update-relational-database API アクションのみを使用してアップグレードできます。

AWS CloudShell を使用してアップグレードを実行します。CloudShell はブラウザーベースの事前認証済みシェルで、Lightsail コンソールから直接起動できます。CloudShell では、Bash、PowerShell、Z シェルなどの任意のシェルを使用して AWS Command Line Interface (AWS CLI) コマンドを実行できます。この手順は、コマンドラインツールのダウンロードもインストールも不要です。CloudShell をセットアップして使用する方法の詳細については、Lightsail の AWS CloudShell を参照してください。

変更を確認する

メジャーバージョンのアップグレードは、以前のバージョンとの間にさまざまな非互換性を引き起こす可能性があります。このような非互換性が原因でアップグレード中に問題が生じる可能性があります。アップグレードを成功させるには、データベースの準備が必要になる場合があります。データベースのメジャーバージョンのアップグレードについては、MySQL および PostgreSQL ウェブサイトの以下のトピックを参照してください。

- アップグレードのためのインストールの準備
- MySQL Upgrade Checker ユーティリティ
- PostgreSQL クラスターのアップグレード

前提条件

1. アプリケーションがデータベースの両方のメジャーバージョンをサポートしていることを確認します。

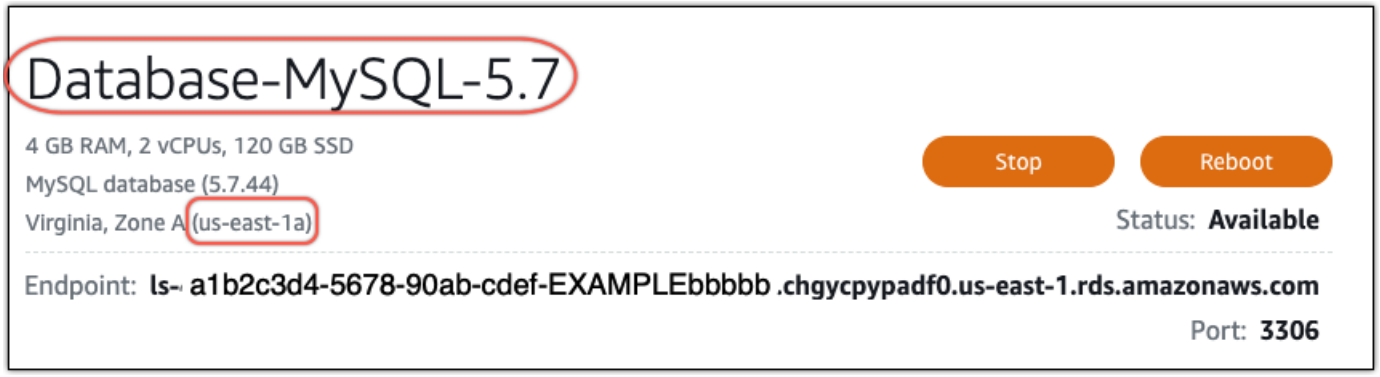
2. 変更を行う前に、データベースインスタンスのスナップショットを作成することをお勧めします。詳細については、「Lightsail データベースのスナップショットを作成する」を参照してください。
3. (選択可能) 作成したスナップショットから新しいデータベースインスタンスを作成します。データベースの更新にはダウンタイムが必要なため、現在アクティブなデータベースをアップグレードする前に、新しいデータベースでアップグレードをテストできます。データベースのコピー作成の詳細については、「Lightsail データベースのスナップショットを作成する」を参照してください。

データベースのメジャーバージョンを更新する

Lightsail は、MySQL および PostgreSQL データベースインスタンスのメジャーバージョンアップグレードをサポートしています。MySQL データベースは、次の手順で例として使用されます。ただし、PostgreSQL データベースに使用されるプロセスとコマンドは同じです。

Lightsail データベースのデータベースメジャーバージョンをアップグレードするには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [データベース] を選択します。
3. アップグレードするデータベースインスタンスの名前と AWS リージョン のメモ。



4. Lightsail コンソールの左下にある [CloudShell] アイコンを選択します。CloudShell ターミナルは、同じブラウザタブで開かれます。コマンドプロンプトが表示されたら、シェルは対話的な操作の準備ができています。
5. CloudShell プロンプトで次のコマンドを入力して、使用可能なデータベースブループリント ID のリストを取得します。

```
aws lightsail get-relational-database-blueprints
```

6. アップグレード先のメジャーバージョンのブループリント ID のメモ。例えば、mysql_8_0。

7. データベースのメージャーバージョンをアップグレードするには、以下のコマンドを入力します。アップグレードは、データベースの次のメンテナンス期間中に行われます。コマンドで、*DatabaseName* をデータベースの名前に、*blueprintId* をアップグレード先のメジャーバージョンのブループリント ID に、*DatabaseRegion* をデータベース AWS リージョン がある に置き換えます。

```
aws lightsail update-relational-database \
 --relational-database-name DatabaseName \
 --relational-database-blueprint-id blueprintId \
 --region DatabaseRegion
```

(選択可能) アップグレードをすぐに適用するには、コマンドに --apply-immediately パラメータを含めます。次の例のようなレスポンスが表示され、アップグレードの適用中にデータベースが使用できなくなります。詳細については、「Lightsail API リファレンス」の「update-relational-database」を参照してください。

```
% aws lightsail update-relational-database \
--relational-database-name "Database-Mysql-5.7" \
--relational-database-blueprint-id "mysql_8_0" \
--apply-immediately \
[--region us-east-1
{
    "operations": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
            "resourceName": "Database-Mysql-5.7",
            "resourceType": "RelationalDatabase",
            "createdAt": 2024-01-01T00:00:00.00000+00:00",
            "location": {
                "availabilityZone": "us-east-1a",
                "regionName": "us-east-1"
            },
            "isTerminal": true,
            "operationDetails": "",
            "operationType": "UpdateRelationalDatabase",
            "status": "Succeeded",
            "statusChangedAt": 2024-01-01T00:00:00.00000+00:00",
        }
```

8. 次のコマンドを入力して、メジャーバージョンアップグレードが次のデータベースメンテナンスウィンドウにスケジュールされていることを確認します。コマンドで、*DatabaseName* をデータベースの名前に置き換え、*DatabaseRegion* をデータベース AWS リージョン がある に置き換えます。

```
aws lightsail get-relational-database \
 --relational-database-name DatabaseName \
 --region DatabaseRegion
```

get-relational-database のレスポンスで、データベース state は次回のメンテナンスウィンドウ中に保留中のメジャーバージョンアップグレードがあることを通知します。次のメンテナンスウィンドウの日付と時刻は、レスポンスの preferredMaintenanceWindow セクションで確認できます。

データベースインスタンスの状態

```
"state": "upgrading",
  "backupRetentionEnabled": true,
  "pendingModifiedValues": {
  "engineVersion": "8.0.36"
```

メンテナンスウィンドウ

```
"preferredMaintenanceWindow": "wed: 09:22-wed: 09:52"
```

次の手順

テストデータベースを作成した場合、アプリケーションがアップグレード後のデータベースで動作することを確認した後に削除できます。前のデータベースに戻りたい場合は、前のデータベースで作成したスナップショットを保管します。アップグレード後のデータベースのスナップショットを作成して、新しい時点のコピーを確保することも推奨されます。

MySQL 5.6 データベースから Lightsail の新しいバージョンにデータを移行する

このチュートリアルでは、MySQL 5.6 データベースから Amazon Lightsail の新しい MySQL 5.7 データベースにデータを移行する方法について解説しています。移行を実行するには、MySQL 5.6 データベースに接続し、既存のデータをエクスポートします。次に、MySQL 5.7 データベースに接続し、データをインポートします。新しいデータベースに必要なデータを取得したら、アプリケーションを再設定して新しいデータベースに接続できるようにします。

目次

- ステップ 1: 変更を確認する
- ステップ 2: 前提条件を完了させる
- ステップ 3: MySQL 5.6 データベースに接続してデータをエクスポートする
- ステップ 4: MySQL 5.7 データベースに接続してデータをインポートする
- ステップ 5: アプリケーションをテストして移行を完了する

ステップ 1: 変更を確認する

MySQL 5.6 データベースから MySQL 5.7 データベースへの移行は、メジャーバージョンへのアップグレードと見なされます。メジャーバージョンのアップグレードには、既存のアプリケーションとの下位互換性のないデータベースの変更が含まれる場合があります。本稼働インスタンスへの適用前に、いずれのアップグレードも徹底的にテストすることをお勧めします。詳細については、「MySQL ドキュメント」の「MySQL 5.7 の変更」を参照してください。

まず、既存の MySQL 5.6 データベースから新しい MySQL 5.7 データベースにデータを移行することをお勧めします。次に、本番前のインスタンスで新しい MySQL 5.7 データベースを使用してアプリケーションをテストします。アプリケーションが期待どおりに動作する場合は、本番環境のインスタンスのアプリケーションに変更を適用します。さらなる措置を取る場合は、既存の MySQL 5.7 データベースから新しい MySQL 8.0 データベースにデータを移行し、本番前のアプリケーションで再度テストし、本番環境のアプリケーションに変更を適用します。

ステップ 2: 前提条件を完了させる

このチュートリアルの次のセクションに進むには、次の必要条件を満たす必要があります。

- ローカルコンピュータに MySQL Workbench をインストールします。このコンピュータを使用して、データベースに接続してデータをエクスポートおよびインポートします。詳細については、MySQL ウェブサイトのMySQL Workbench のダウンロードを参照してください。
- Lightsail で MySQL 5.7 データベースを作成します。詳細については、「Amazon Lightsail でデータベースを作成する」を参照してください。
- データベースのパブリックモードを有効にします。これにより、MySQL Workbench を使用してデータベースに接続することができるようになります。データのエクスポートとインポートが完了したら、データベースのパブリックモードを無効にすることができます。詳細については、「データベースのパブリックモードの設定」を参照してください。
- MySQL Workbench を設定してデータベースに接続する。詳細については、「MySQL データベースに接続する」を参照してください。

ステップ 3: MySQL 5.6 データベースに接続してデータをエクスポートする

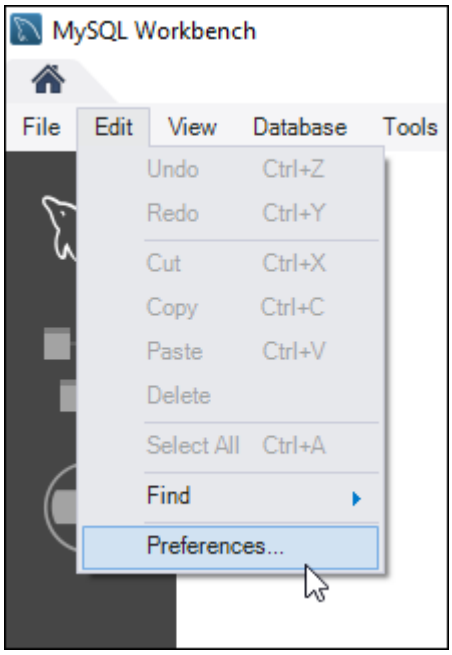
チュートリアルのこのセクションでは、MySQL 5.6 データベースに接続し、MySQL Workbench を使用してそのデータベースからデータをエクスポートします。MySQL Workbench を使用してデータをエクスポートする方法の詳細については、MySQL Workbench マニュアルの「SQL Data のエクスポートとインポートウィザード」を参照してください。

1. MySQL Workbench を使用して MySQL 5.6 データベースに接続する。

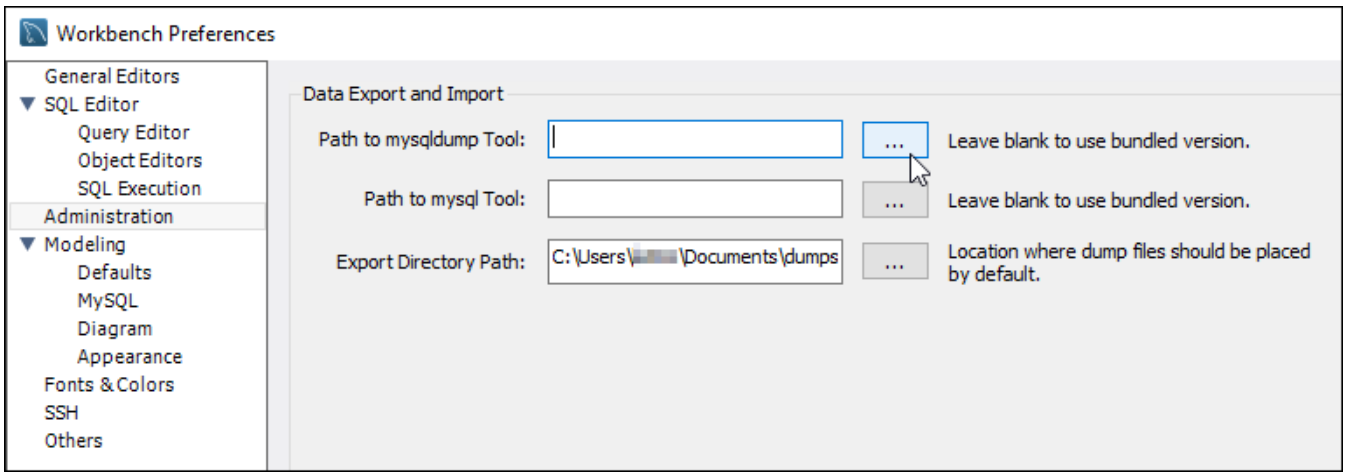
   MySQL ワークベンチでは、mysqldump を使用してデータをエクスポートします。MySQL Workbench で使用される mysqldump のバージョンは、データをエクスポートする MySQL データベースのバージョンと同じ (またはそれ以降) である必要があります。たとえば、MySQL 5.6.51 データベースからデータをエクスポートする場合は、バージョン 5.6.51 かそれ以降の mysqldump を使用する必要があります。正しいバージョンの mysqldump を使用しているか確認するために、ローカルコンピュータに適切なバージョンの MySQL サーバーをダウンロードしてインストールする必要がある場合があります。MySQL サーバーの特定バージョンをダウンロードするには、「MySQL ウェブサイト」の「MySQL コミュニティダウンロード」を参照してください。「Windows MSI 用の MySQL インストーラ」では、ダウンロードする MySQL サーバーのバージョンを選択できます。

   MySQL ワークベンチで使用する mysqldump の正しいバージョンを選択するには、以下の手順を実行します。

   1. MySQL ワークベンチで [Edit] (編集)、[Preferences] (設定) の順に選択します。

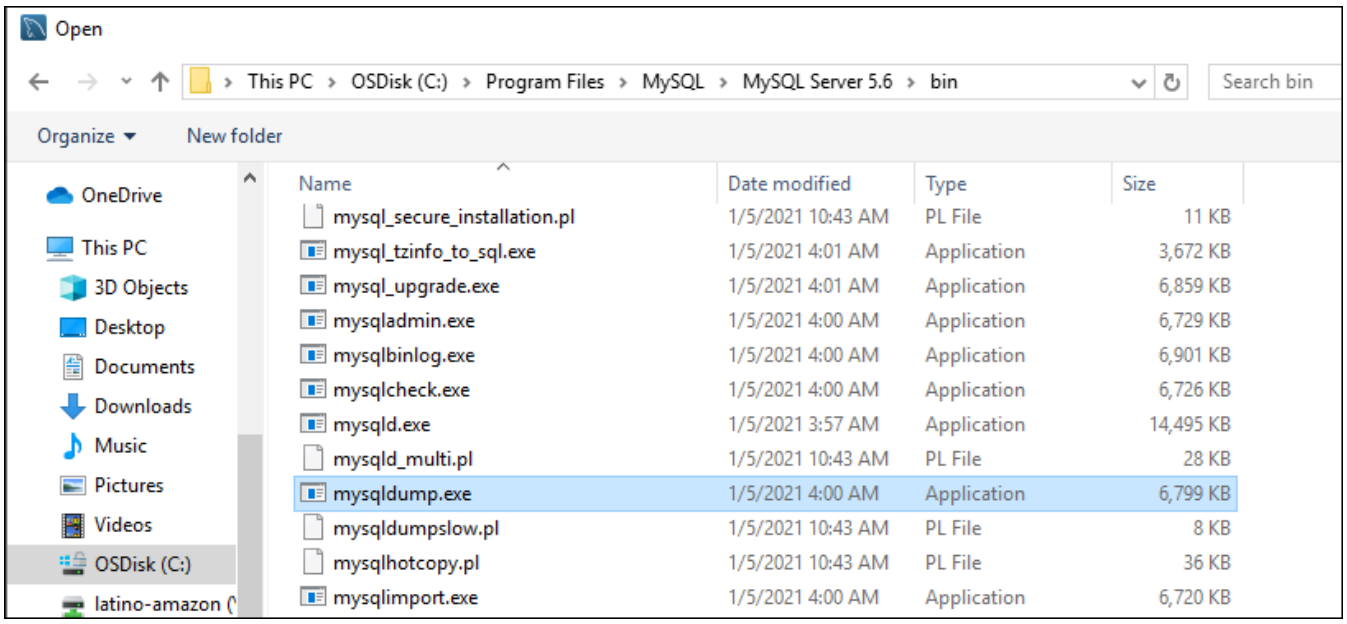
      

   2. ナビゲーションペインで [Administration] (管理) を選択します。
   3. Workbench Preferences ウィンドウが表示されたら、Path to mysqldump Tool のテキストボックスの横にある省略記号ボタンを選択します。

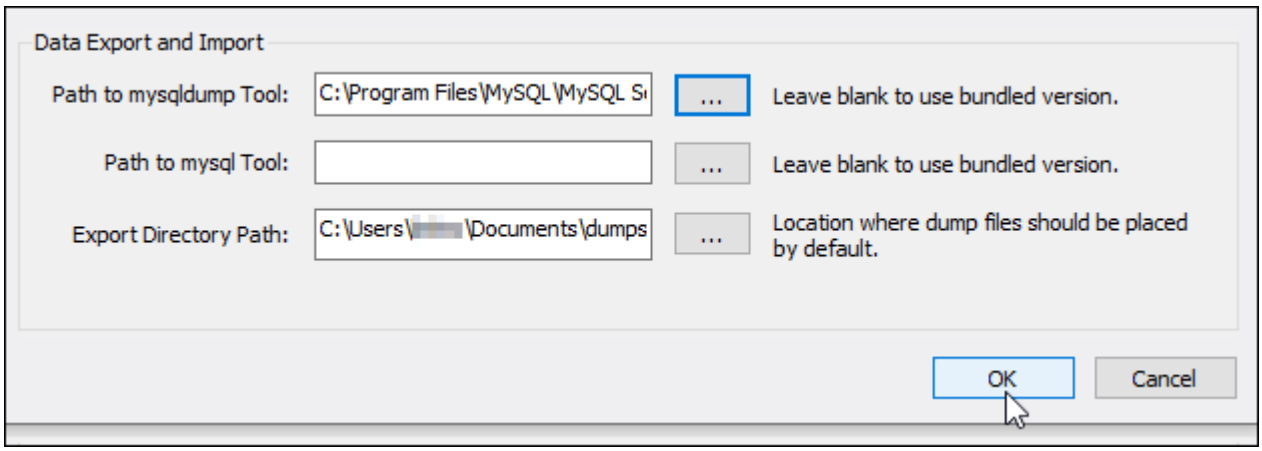


4. 適切なmysqldump 実行可能ファイルの場所まで移動したら、ダブルクリックします。

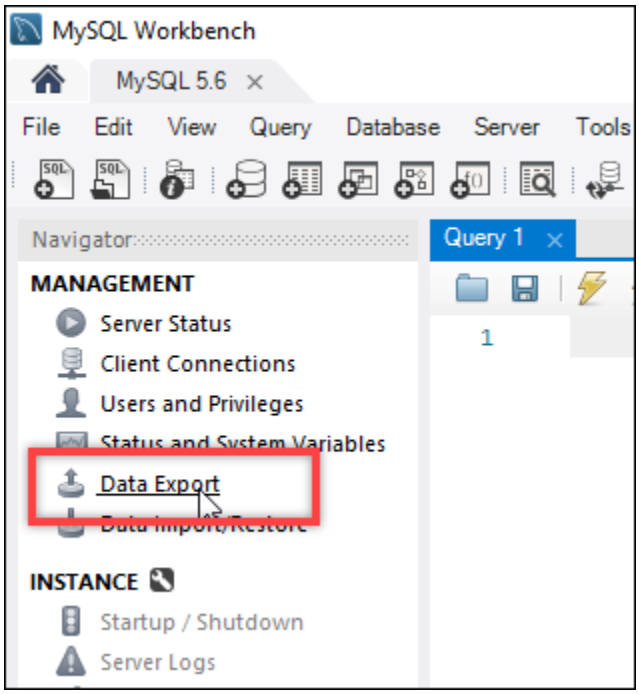
   Windows の場合、mysqldump.exe ファイルは通常 C:\Program Files\MySQL \MySQL Server 5.6\bin ディレクトリにあります。Linux の場合、ターミナルに which mysqldump を入力して mysqldump ファイルの位置を確認します。



5. Workbench Preferences ウィンドウで [OK] を選択します。

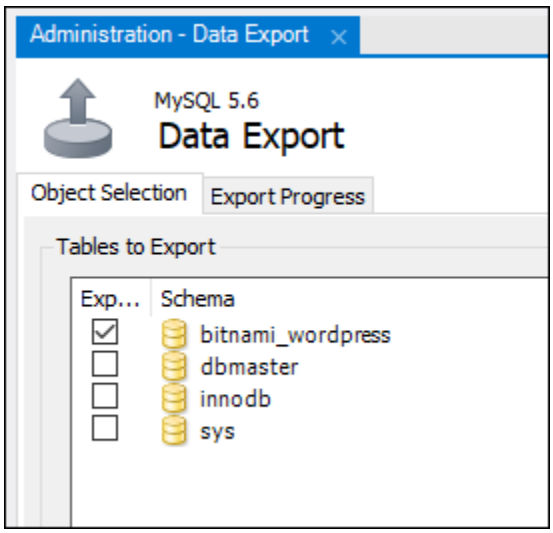


2. [Navigator] (ナビゲーター) ペインで [Data Export] (データエクスポート) を選択します。

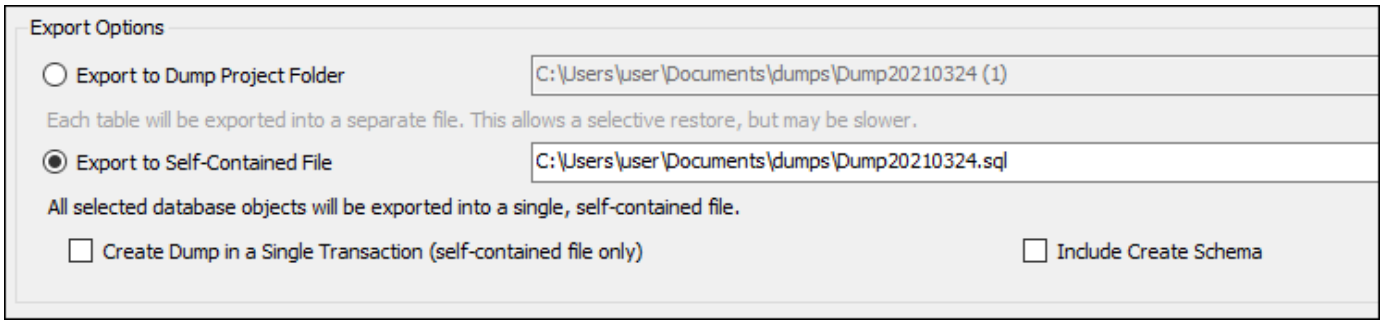


3. [データのエクスポート] タブが表示されたら、エクスポートするテーブルの横にチェックマークを追加します。

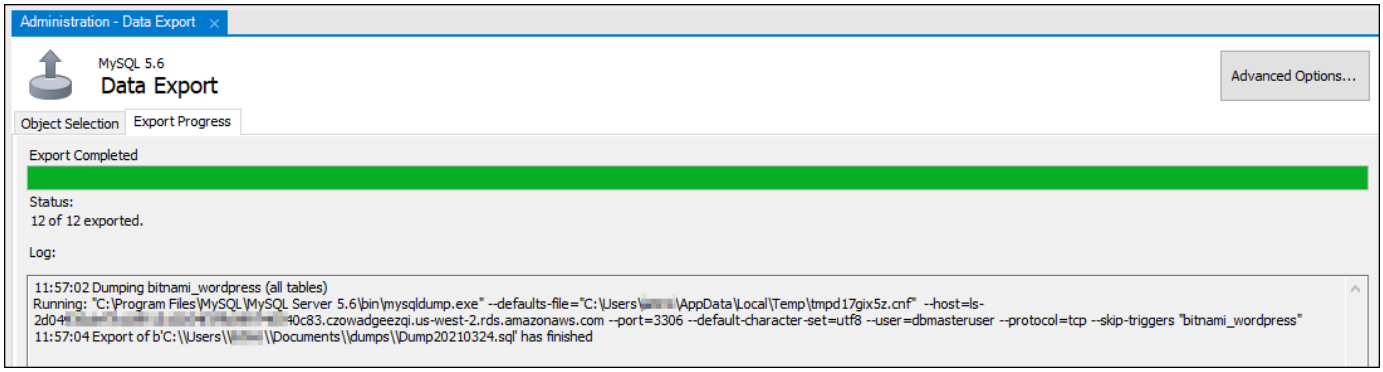
> **Note**
>
> この例で選択した `bitnami_wordpress` テーブルには、「Certified by Bitnami」WordPress インスタンスにある WordPress ウェブサイトのデータが含まれています。



4. [Export Options] (エクスポートオプション) セクションで、[Export to Self-Contained File] (自己完結型ファイルにエクスポート) を選択してエクスポートファイルが保存されるディレクトリを書き留めておきます。



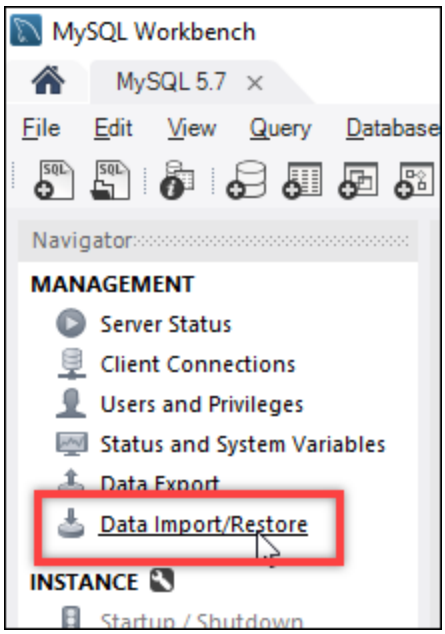
5. [Start Export] (エクスポートの開始) を選択します。
6. このチュートリアルの次のセクションに進む前に、エクスポートが完了するのを待ちます。



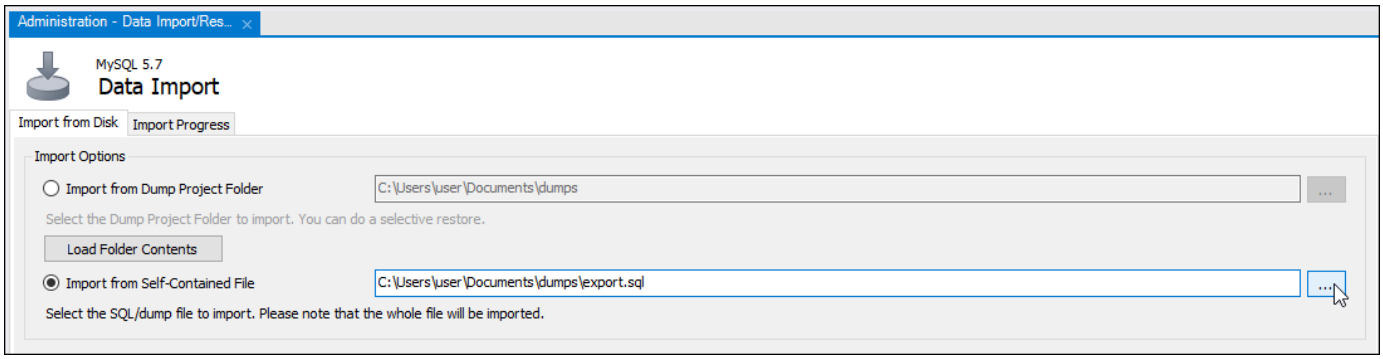
ステップ 4: MySQL 5.7 データベースに接続してデータをインポートする

チュートリアルのこのセクションでは、MySQL 5.7 データベースに接続し、MySQL Workbench を使用してそのデータベースにデータをエクスポートします。

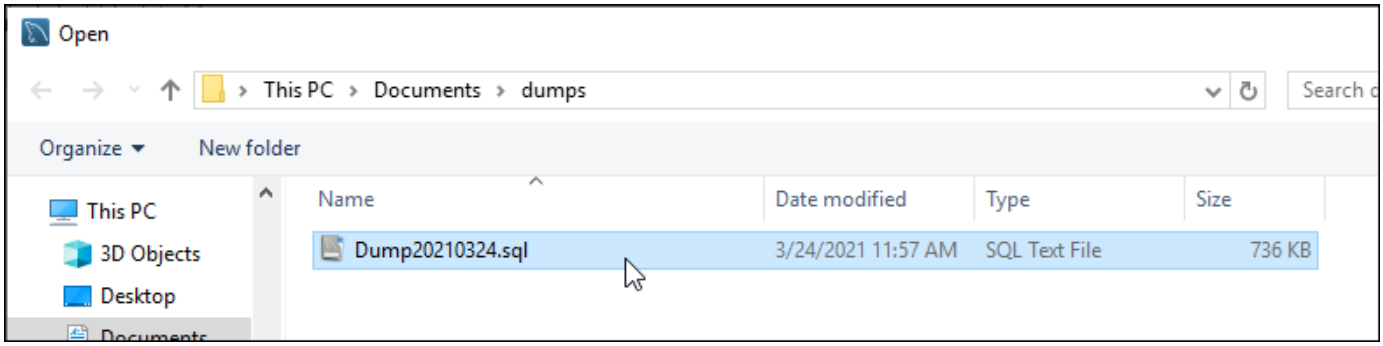
1. ローカルコンピュータの MySQL Workbench を使用して MySQL 5.7 データベースに接続する。
2. [Navigator] (ナビゲーター) ペインの [Data Import/Restore] (データのインポート/復元) を選択します。



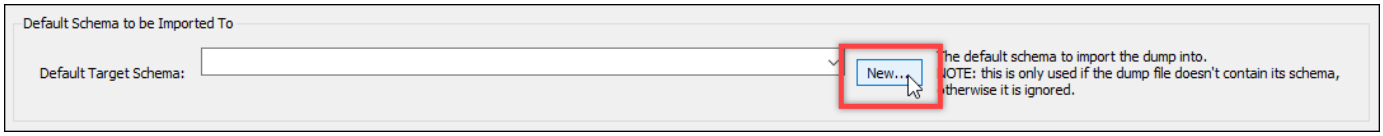
3. [Data Import] (データのインポート) タブが表示されたら、[Export to Self-Contained File] (自己完結型ファイルにエクスポート) を選択して、テキストボックスの横にある省略記号ボタンを選択します。



4. エクスポートファイルが保存された場所まで移動したら、ダブルクリックします。

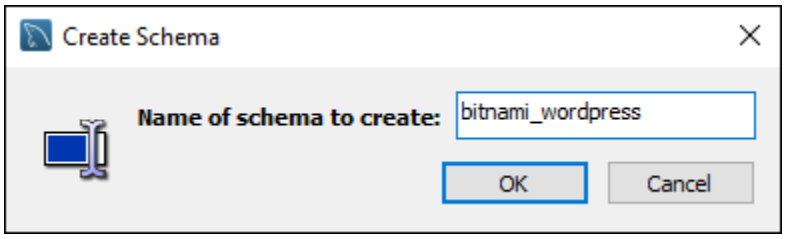


5. [Default Schema to be imported To] (インポート先のデフォルトスキーマ) のセクションで、[New] (新規) を選択します。

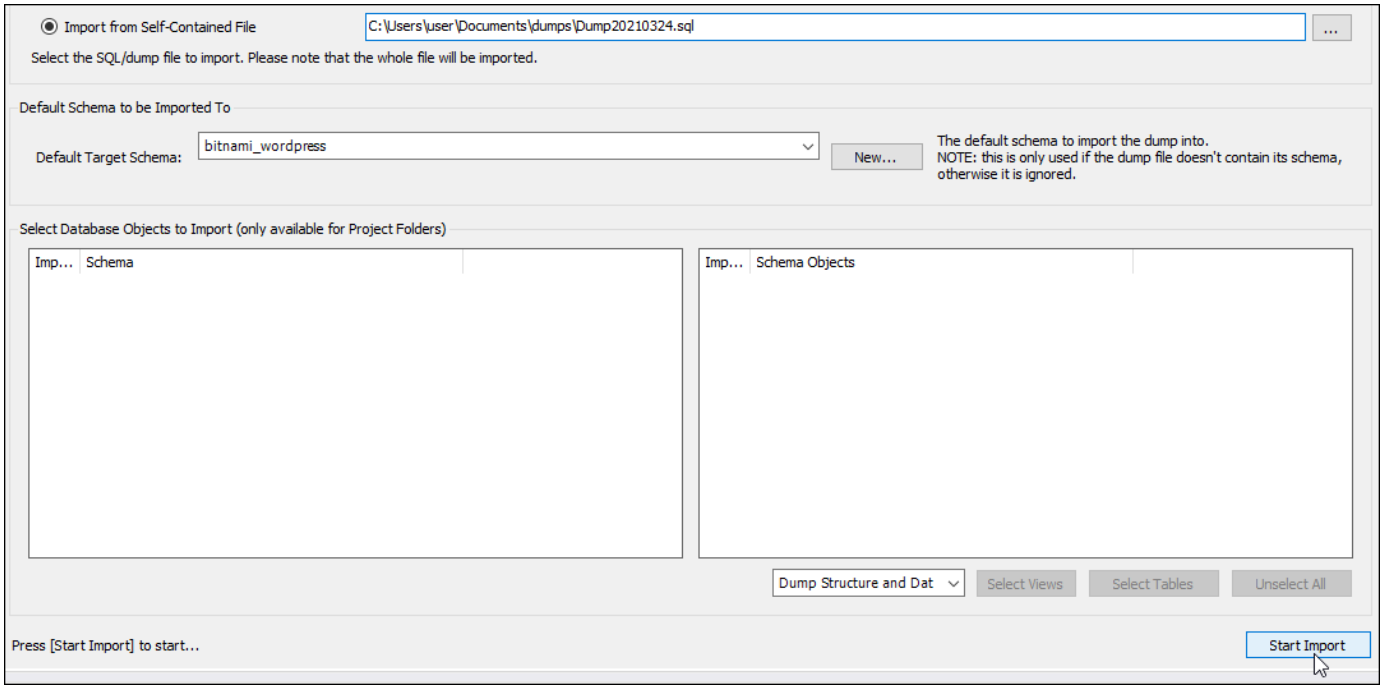


6. [Create Schema] (スキーマの作成) ウィンドウが表示されたら、スキーマの名前を入力します。

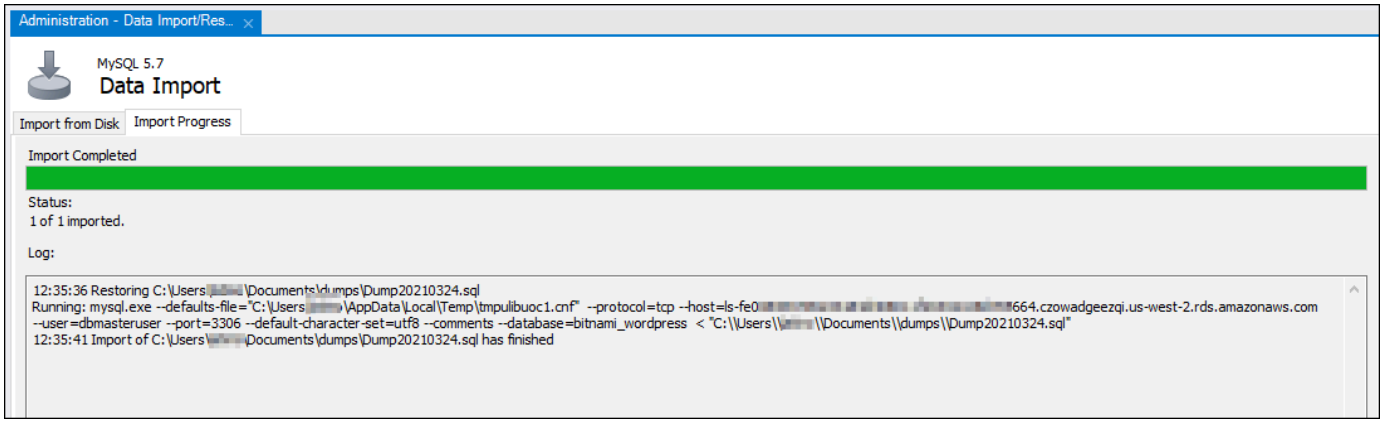
> **Note**
>
> この例では、エクスポートされたデータベーステーブルの名前が `bitnami_wordpress` であるため、それを入力しています。



7. [Start import] (インポートの開始) を選択します。



8. このチュートリアルの次のセクションに進む前に、インポートが完了するのを待ちます。



ステップ 5: アプリケーションをテストして移行を完了する

この時点で、データは新しい MySQL 5.7 データベースに格納されます。本番前の環境でアプリケーションを設定し、アプリケーションと新しい MySQL 5.7 データベース間で接続テストを行います。アプリケーションが期待どおりに動作する場合は、本番環境でアプリケーションに変更を反映します。

移行が完了したら、データベースのパブリックモードを無効にする必要があります。不要になった MySQL 5.6 データベースは削除できます。ただし、削除する前に MySQL 5.6 データベースのスナップショットを作成することをお勧めします。またこの作業をする際、新しい MySQL 5.7 データベー

スのスナップショットも作成しておくことをお勧めします。詳細については、「データベースのスナップショットを作成する」を参照してください。

Lightsail ロードバランサーを使用してウェブトラフィックを分散する

Lightsail ロードバランサーは、着信ウェブトラフィックを複数のアベイラビリティーゾーンの複数の Lightsail インスタンス間で分散します。ロードバランシングを使用すると、インスタンスでのアプリケーションの可用性と耐障害性が向上します。アプリケーションへのリクエストの流れを中断することなく、ニーズの変化に応じて Lightsail ロードバランサーに対して インスタンスの追加と削除を行うことができます。

Lightsail のロードバランシングを使用して、DNS ホスト名を作成し、このホスト名に送信されるすべてのリクエストを、ターゲットの Lightsail インスタンスのプールにルーティングします。インスタンスの合計数に関する Lightsail アカウントのクォータを超えない限り、任意の数のターゲットインスタンスをロードバランサーに追加できます。

ロードバランサーの機能

Lightsail ロードバランサーは、次の機能を備えています。

- HTTPS 暗号化 — デフォルトでは、Lightsail ロードバランサーは、ポート 80 を介して暗号化されていない (HTTP) トラフィックリクエストを処理します。検証済みの Lightsail SSL/TLS 証明書をロードバランサーにアタッチして HTTPS 暗号化をアクティブ化します。これにより、ロードバランサーは、ポート 443 を介して暗号化された (HTTPS) トラフィックリクエストを処理することができます。詳細については、「SSL/TLS 証明書」を参照してください。

  ロードバランサーで HTTPS 暗号化をアクティブ化すると、次の機能を使用できます。

  - HTTP から HTTPS へのリダイレクト — HTTP から HTTPS へのリダイレクトをアクティブ化して、HTTP リクエストを HTTPS 暗号化接続に自動的にリダイレクトします。詳細については、「ロードバランサーの HTTP から HTTPS へのリダイレクトの設定」を参照してください。
  - TLS セキュリティポリシー — ロードバランサーで TLS セキュリティポリシーを設定します。詳細については、「Amazon Lightsail ロードバランサーでの TLS セキュリティポリシーの設定」を参照してください。
- ヘルスチェック — デフォルトでは、ヘルスチェックは、アタッチされたインスタンスにおいて、それらのインスタンスで実行されているウェブアプリケーションのルートで実行されます。ヘルスチェックは、ロードバランサーから正常なインスタンスにのみリクエストを送信できるように、インスタンスのヘルス状態をモニタリングします。詳細については、「Lightsail ロードバランサーのヘルスチェック」を参照してください。

- セッション永続性 — ウェブサイトの訪問者のブラウザでセッション情報をローカルに保存する場合は、セッション永続性を設定します。例えば、ロードバランシング済みの Lightsail インスタンスで、ショッピングカート機能を備えた Magento e コマースアプリケーションを実行しているとします。ウェブサイトの訪問者がショッピングカートに商品を追加してセッションを終了した後に戻ってくると、セッション永続性を設定した場合はショッピングカートの商品が残っています。詳細については、「ロードバランサーのセッション永続性を有効にする」を参照してください。

ロードバランサーを使用するタイミング

ロードバランサーは、トラフィックがときどき急上昇するウェブサイトがある場合や、多くのユーザーが一度に利用したときにインスタンスで大きな負荷が発生するコストコンテンツがある場合に使用してください。たとえば、画像が多いウェブサイトがある場合、他のページリクエストを使ってイメージリクエストをロードバランシングすることができます。このようにすると、ページのロード時間が短縮され、ユーザーの満足度が向上します。

ロードバランサーを使用すると、可用性の高いウェブサイトを作成できます。高可用性とは、一定期間内にウェブサイトやアプリケーションが稼働している時間の長さを指します。これまでサイトが停止したことがある場合、ロードバランサーはアップタイムの増加に役立つ場合があります。Lightsail ロードバランサーを使用すると、複数のアベイラビリティーゾーンに分散されるターゲットインスタンスを追加することで、アプリケーションの可用性を高めることができます。

耐障害性は、関連する概念です。いずれかのインスタンスまたはデータベースで障害が発生した後もサイトが動作し続ける場合、耐障害性があると見なされます。ロードバランサーは、耐障害性を備えたアプリケーションまたはウェブサイトを作成するのに役立ちます。

ロードバランシングが推奨される アプリケーション

すべての Lightsail アプリケーションにロードバランサーが必要なわけではありません。ロードバランシングされたアプリケーションを作成することに決定した場合、最初にアプリケーションを設定する必要があります。例えば、ロードバランシング用の LAMP スタックアプリケーションを準備するには、最初にすべてのターゲットインスタンスが読み取り/書き込みを行うための一元的な専用データベースを作成する必要があります。Lightsail オブジェクトストレージバケットなどの一元化されたメディアストレージの作成を検討することも考えられます。詳細については、「ロードバランシング用のインスタンスを設定する」を参照してください。

ロードバランサーの使用を開始する

Lightsail コンソール、 AWS Command Line Interface (AWS CLI)、または Lightsail API を使用してロードバランサーを作成できます。ロードバランシング用のインスタンスも設定する必要があります。

ロードバランサーを作成して設定したインスタンスをアタッチすると、次のトピックを使用して HTTPS を有効にすることができます。詳細については、「ロードバランサー用の SSL/TLS 証明書を作成する」を参照してください。

Lightsail ロードバランサーを使用してウェブトラフィックを分散する

アプリケーションの冗長性を高め、より多くのウェブトラフィックを処理するには、 ロードバランサーを作成します。ロードバランサーが作成されたら、負荷分散する Lightsail インスタンスをアタッチできます。詳細については、「ロードバランサー」を参照してください。

前提条件

開始する前に、ロードバランシング用の Lightsail インスタンスを準備したことを確認します。詳細については、「ロードバランシング用の インスタンスを設定する」を参照してください。

ロードバランサーの作成

1. Lightsail コンソール にサインインします。
2. [ネットワーキング] タブを選択してください。
3. [ロードバランサーを作成] を選択します。
4. ロードバランサーが作成される AWS リージョン を確認するか、リージョンの変更を選択して別のリージョンを選択します。

> **Note**
>
> デフォルトでは、HTTP リクエストを受け入れるため、ポート 80 が開いたロードバランサーが作成されます。ロードバランサーを作成した後は、SSL/TLS 証明書を作成して HTTPS を設定できます。詳細については、「ロードバランサー用の SSL/TLS 証明書を作成する」を参照してください

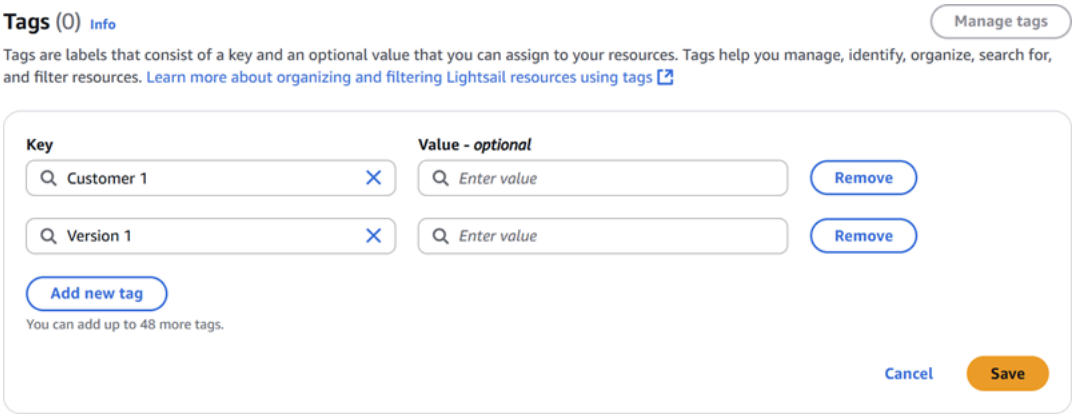
5. ロードバランサーの名前を入力します。

   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

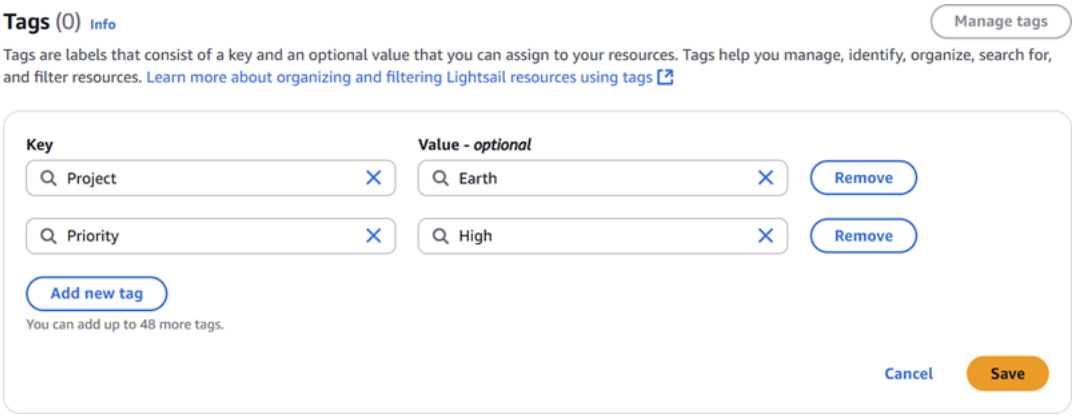
6. 以下のいずれかのオプションを選択して、ロードバランサーにタグを追加します。

   - [キーオンリータグの追加] または [タグを編集] (タグが追加済みの場合)。タグキーのテキストボックスに新しいタグを入力し、Enter キーを押します。タグの入力を完了したら、[保存] を選択してタグを追加し、追加しない場合は、[キャンセル] を選択します。



   - [Create a key-value tag] (キーバリュータグを作成) から [Key] (キー) テキストボックスにキーを入力し、[Value] (値) テキストボックスに値を入力します。タグの入力を完了したら、[保存] を選択し、追加しない場合は、[キャンセル] を選択します。

     キーバリューのタグは、保存する際に一つずつ追加することができます。さらにキーバリュータグを追加するには、以上のステップを繰り返します。



> **Note**
>
> 「キーのみ」のタグと「キーバリュー」のタグの詳細については、「タグ」を参照してください。

7. [ロードバランサーを作成] を選択します。

インスタンスをロードバランサーにアタッチする

ロードバランサーを作成した後、Lightsail のロードバランサー管理ページに移動します。そのページを再表示する必要がある場合は、Lightsail ホーム画面の[ネットワーク]タブを開き、Lightsailロードバランサーの名前を選択し、管理します。

> **Note**
>
> Lightsail インスタンスは、ロードバランサーにアタッチする前に実行されている必要があります。

1. ロードバランサー管理ページで、[ターゲットインスタンス] を選択します。
2. [ターゲットインスタンス] ドロップダウンリストでインスタンスを選択します。
3. [アタッチ] を選択してください。アタッチには数分かかる場合があります。

   [Attach another] を選択し、前述のステップを繰り返して、別のインスタンスをロードバランサーにアタッチします。

次の手順

ロードバランサーが作成され、インスタンスがアタッチされたら、続く次のステップを完了して、ロードバランサーを設定します。

- ロードバランサー用の SSL/TLS 証明書を作成する
- ロードバランサー用のヘルスチェックをカスタマイズする

ロードバランサーに関する問題が発生した場合は、「ロードバランサーに関するトラブルシューティング」を参照してください。

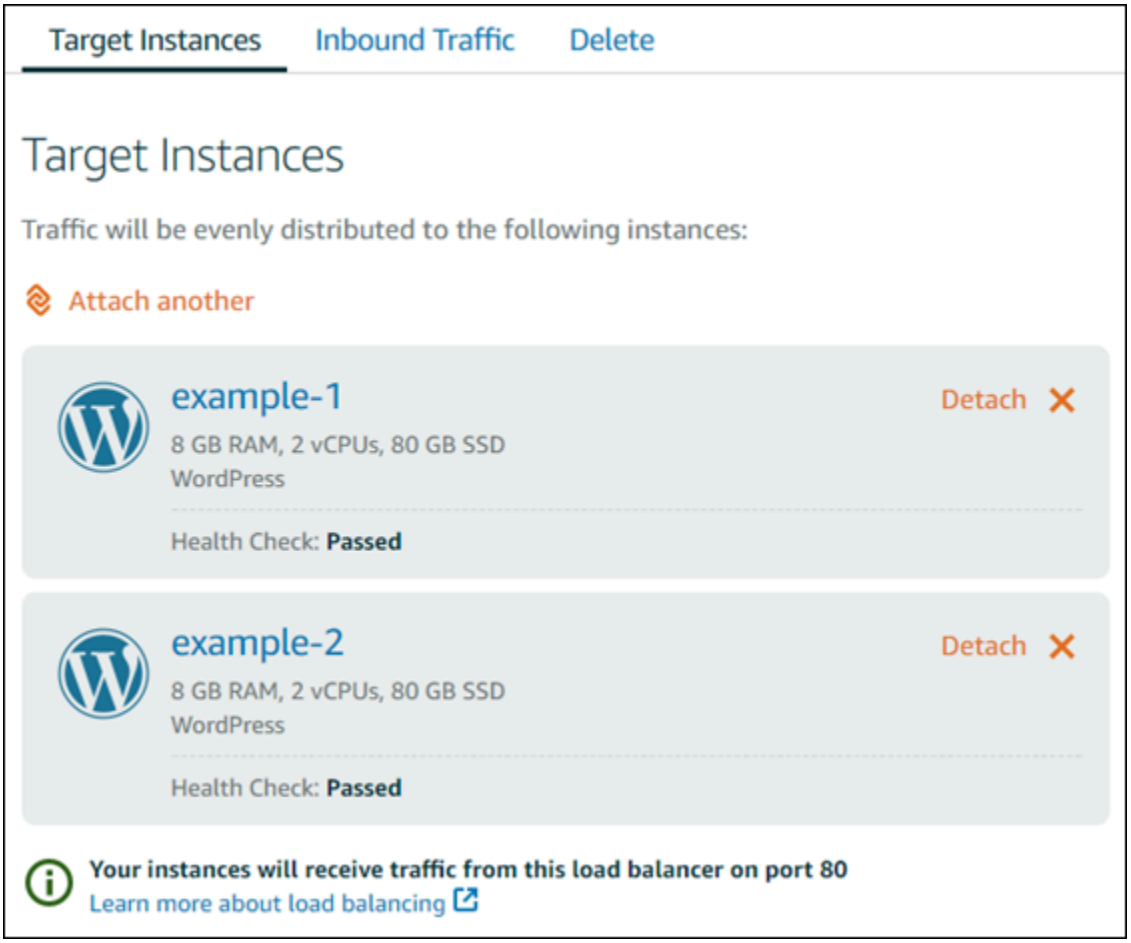
Lightsail ロードバランサーのヘルスチェックと HTTPS 設定をカスタマイズする

Lightsail ロードバランサーは、AWS リージョン と名前を選択するだけで作成できます。このトピックでは、ロードバランサーを更新して他のオプションを有効にする方法について説明します。

ロードバランサーをまだ作成していない場合は、作成する必要があります。ロードバランサーの作成

ヘルスチェック

まず、ロードバランシング用のインスタンスを設定 をします。完了したら、インスタンスをロードバランサーにアタッチできます。インスタンスをアタッチすると、ヘルスチェックプロセスが開始され、ロードバランサー管理ページに成功または失敗のメッセージが表示されます。



ヘルスチェックのパスをカスタマイズすることもできます。たとえば、ホームページのロードに時間がかかる場合や多くの画像が使用されている場合、すぐにロードされる別のページをチェックするように Lightsail を設定します。ロードバランサーのヘルスチェックパスのカスタマイズ

暗号化されたトラフィック (HTTPS)

ウェブサイトでより安全なユーザー体験を実現するため、HTTPS を設定できます。ロードバランサーを設定したら SSL/TLS 証明書を作成して検証する 3 ステップのプロセスです。

HTTPS の詳細

セッション永続性

セッション永続性は、ユーザーのブラウザでセッション情報をローカルに保存する場合に役立ちます。たとえば、Lightsail にショッピングカートのある Magento e コマースアプリケーションを実行しているとします。セッション永続性を有効にした場合、ユーザーがショッピングカートに商品を追加してセッションを終了しても、戻ってくるとカートに商品が残っています。

また、永続的なセッションの Cookie の有効期間を調整することもできます。これは、特に長い有効期間や短い有効期間が必要な場合に役立ちます。詳細については、「ロードバランサーのセッション永続性を有効にする」を参照してください。

ロードバランサー用の Lightsail インスタンスを設定する

インスタンスを Amazon Lightsail ロードバランサーにアタッチする前に、アプリケーションの設定を評価する必要があります。たとえば、ロードバランサーは多くの場合、データ層がアプリケーションの残りの部分と分離されている方が適切に動作します。このトピックでは、各 Lightsail インスタンスについて説明し、ロードバランシング (水平スケーリング) を行うかどうかとアプリケーションの最適な設定方法に関する推奨事項を示します。

一般的なガイドライン: データベースを使用するアプリケーション

データベースを使用する Lightsail アプリケーションの場合、データベースインスタンスが 1 つだけになるように、データベースインスタンスをアプリケーションの残りの部分から分離することをお勧めします。主な理由は、複数のデータベースにデータが書き込まれないようにすることです。1 つのデータベースインスタンスを作成しない場合、ユーザーがたまたまヒットしたインスタンス上のデータベースにデータが書き込まれます。

WordPress

水平スケーリングを行いますか? はい、WordPress ブログまたはウェブサイトで行います。

Lightsail ロードバランサーを使用する前の設定の推奨事項

- ロードバランサーの背後で実行されているあらゆる WordPress インスタンスが同じ場所に情報を保存および取得するように、データベースを分離します。データベースのより高いパフォーマンスが必要な場合、ウェブサーバーとは別個に処理能力やメモリをレプリケートまたは変更することができます。
- ファイルと静的コンテンツを Lightsail バケットにオフロードします。これを実行するには、WordPress のウェブサイトに WP Offload Media Lite プラグインをインストールし、Lightsail バケットに接続するように設定する必要があります。詳細については、「チュートリアル: WordPress インスタンスをストレージ バケットに接続する」を参照してください。

Node.js

水平スケーリングを行いますか? はい。ただし、いくつかの考慮事項があります。

Lightsail ロードバランサーを使用する前の設定の推奨事項

- Lightsail では、Bitnami によってパッケージ化されている Node.js スタックには、Node.js、Apache、Redis (メモリ内データベース)、Python が含まれています。デプロイするアプリケーションに応じて、いくつかのサーバー間でロードバランシングすることができます。ただし、すべてのウェブサーバー間でトラフィックが分散され、Redis が別のサーバーに移動されるようにロードバランサーを設定する必要があります。
- Redis サーバーを別のサーバーに分割して、すべてのインスタンスと通信します。必要に応じて、データベースサーバーを追加します。
- Redis の主なユースケースの 1 つは、データをローカルにキャッシュするため、中央のデータベースに継続的にヒットする必要はありません。Redis によるパフォーマンス向上を活用するには、セッション永続性を有効にすることをお勧めします。詳細については、「ロードバランサーのセッション永続性を有効にする」を参照してください。
- 共有 Redis ノードを作成することもできるため、セッション永続性を使用する各マシンでノードを共有したり、ローカルキャッシュを使用したりすることもできます。
- Apache を使用してロードバランサーをデプロイする場合は、`mod_proxy_balancer` を Apache サーバーに含めることを検討してください。

詳細については、「Scaling Node.js applications」を参照してください。

Magento

水平スケーリングを行いますか? はい。

Lightsail ロードバランサーを使用する前の設定の推奨事項

- Amazon RDS データベースなどの追加のコンポーネントを使用する Magento の AWS リファレンスデプロイを使用できます: Terraform Magento Adobe Commerce on AWS。
- 必ず、セッション永続性を有効にしてください。Magento はショッピングカートを使用しているため、セッションをまたいで複数回訪問するお客様が、新しいセッションで戻ってきたときもショッピングカート内に商品を保持することができます。詳細については、「ロードバランサーのセッション永続性を有効にする」を参照してください。

GitLab

水平スケーリングを行いますか? はい。ただし、考慮事項があります。

Lightsail ロードバランサーを使用する前の設定の推奨事項

以下を準備する必要があります。

- 実行されており、使用準備ができた Redis ノード
- 共有ネットワークストレージサーバー (NFS)
- アプリケーション用の一元化されたデータベース (MySQL または PostgreSQL)。上記のデータベースに関する一般的なガイドラインを参照してください。

詳細については、「GitLab」ウェブサイトの「高可用性」を参照してください。

> **Note**
>
> 上記で言及されている共有ネットワークストレージサーバー (NFS) は、GitLab ブループリントでは現在利用できません。

Drupal

水平スケーリングを行いますか? はい。Drupal には、アプリケーションを水平スケーリングする方法を説明する公式ドキュメント「Server Scaling」があります。

Lightsail ロードバランサーを使用する前の設定の推奨事項

異なるインスタンス間でファイルが同期されるように Drupal モジュールを設定する必要があります。Drupal ウェブサイトにはいくつかのモジュールがありますが、本稼働使用ではなくプロトタイプ作成の方に適している可能性があります。

ファイルを Amazon S3 に保存できるモジュールを使用します。これにより、ターゲットインスタンスごとに別個のコピーを保持するのではなく、ファイルを一元化された場所に保存できます。このようにして、ファイルを編集した場合、ヒットしたインスタンスに関係なく、一元化されたストアから更新を取得してユーザーに同じファイルを表示できます。

- Amazon S3 ファイルシステム
- コンテンツの同期

詳細については、「Drupal をクラウドで水平方向にスケーリング」を参照してください。

LAMP スタック

水平スケーリングを行いますか? はい。

Lightsail ロードバランサーを使用する前の設定の推奨事項

- 別のインスタンスにデータベースを作成する必要があります。ロードバランサーの背後にあるすべてのインスタンスは、この別個のデータベースインスタンスをポイントするため、同じ場所に情報を保存および取得できます。
- デプロイするアプリケーションに応じて、ファイルシステムを共有する方法を検討します (NFS、Lightsail ブロックストレージディスク、または Amazon S3 ストレージ)。

MEAN スタック

水平スケーリングを行いますか? はい。

Lightsail ロードバランサーを使用する前の設定の推奨事項

MongoDB を別のマシンに移動し、Lightsail インスタンス間でルートドキュメントを共有するメカニズムを設定します。

Redmine

水平スケーリングを行いますか? はい。

Lightsail ロードバランサーを使用する前の設定の推奨事項

- Redmine_s3 プラグインを取得し、添付ファイルをローカルファイルシステムではなく Amazon S3 に格納します。
- 別のインスタンスにデータベースを分離します。

Nginx

水平スケーリングを行いますか? はい。

Nginx を実行する Lightsail インスタンスを 1 つ以上、Lightsail ロードバランサーにアタッチできます。詳細については、「Scaling Web Applications with NGINX, Part 1: Load Balancing」を参照してください。

Joomla!

水平スケーリングを行いますか? はい。ただし、考慮事項があります。

Lightsail ロードバランサーを使用する前の設定の推奨事項

Joomla ウェブサイトに公式ドキュメントはありませんが、コミュニティフォーラムで議論されています。一部のユーザーは、次の設定のクラスターを作成して Joomla インスタンスを水平スケーリングしています。

- セッション永続性を有効にするように設定された Lightsail ロードバランサー。詳細については、「ロードバランサーのセッション永続性を有効にする」を参照してください。
- Joomla! のドキュメントルートが同期された状態でロードバランサーにアタッチされた Joomla を実行する複数の Lightsail。これを行うには、Rsync などのツールを使用する、すべての Lightsail インスタンス間でコンテンツを同期できる NFS サーバーを持つ、または AWSを使用してファイルを共有します。
- レプリケーションクラスターで設定されたいくつかのデータベースサーバー。

- Lightsail インスタンスごとに設定されている同じキャッシュシステム。JotCache など、役に立つ拡張機能がいくつかあります。

Lightsail ロードバランサーの TLS セキュリティポリシーを設定する

Amazon Lightsail ロードバランサーで HTTPS を有効にした後、暗号化された接続の TLS セキュリティポリシーを設定できます。このガイドでは、Lightsail ロードバランサーで設定できるセキュリティポリシー、およびロードバランサーのセキュリティポリシーを更新する手順について説明します。ロードバランサーの詳細については、「ロードバランサー」を参照してください。

セキュリティポリシーの概要

Lightsail ロードバランシングは、Secure Sockets Layer (SSL) ネゴシエーション設定 (セキュリティポリシーと呼ばれます) を使用して、クライアントとロードバランサー間の SSL 接続をネゴシエートします。セキュリティポリシーはプロトコルと暗号の組み合わせです。プロトコルは、クライアントとサーバーの間の安全な接続を確立し、クライアントとロードバランサーの間で受け渡しされるすべてのデータのプライバシーを保証します。暗号とは、暗号化キーを使用してコード化されたメッセージを作成する暗号化アルゴリズムです。プロトコルは、複数の暗号を使用し、インターネットを介してデータを暗号化します。接続ネゴシエーションのプロセスで、クライアントとロードバランサーでは、それぞれサポートされる暗号とプロトコルのリストが優先される順に表示されます。デフォルトでは、クライアントの暗号のいずれかに一致するサーバーのリストの最初の暗号が安全な接続のために選択されます。Lightsail ロードバランサーは、クライアントまたはターゲット接続の SSL 再ネゴシエーションをサポートしていません。

Lightsail ロードバランサーで HTTPS を有効にすると、`TLS-2016-08` セキュリティポリシーがデフォルトで設定されます。このガイドで後述するように、必要に応じて別のセキュリティポリシーを設定できます。フロントエンド接続のみに使用するセキュリティポリシーを選択できます。バックエンド接続には、常に `TLS-2016-08` セキュリティポリシーが使用されます。Lightsail ロードバランサーはカスタムセキュリティポリシーをサポートしていません。

サポートされているセキュリティポリシーとプロトコル

Lightsail ロードバランサーは、次のセキュリティポリシーおよびプロトコルを使用して設定できます。

Security policies	TLS-2016-08 (default)	TLS-FS-1-2-Res-2019-08
TLS Protocols		
Protocol-TLSv1	✓	
Protocol-TLSv1.1	✓	
Protocol-TLSv1.2	✓	✓
TLS Ciphers		
ECDHE-ECDSA-AES128-GCM-SHA256	✓	✓
ECDHE-RSA-AES128-GCM-SHA256	✓	✓
ECDHE-ECDSA-AES128-SHA256	✓	✓
ECDHE-RSA-AES128-SHA256	✓	✓
ECDHE-ECDSA-AES128-SHA	✓	
ECDHE-RSA-AES128-SHA	✓	
ECDHE-ECDSA-AES256-GCM-SHA384	✓	✓
ECDHE-RSA-AES256-GCM-SHA384	✓	✓
ECDHE-ECDSA-AES256-SHA384	✓	✓
ECDHE-RSA-AES256-SHA384	✓	✓
ECDHE-RSA-AES256-SHA	✓	
ECDHE-ECDSA-AES256-SHA	✓	
AES128-GCM-SHA256	✓	
AES128-SHA256	✓	
AES128-SHA	✓	

前提条件を満たす

以下の前提条件を完了します (まだの場合)。

- ロードバランサーを作成してインスタンスをアタッチする。詳細については、「ロードバランサーを作成してインスタンスをアタッチする」を参照してください。
- SSL/TLS 証明書を作成し、ロードバランサーにアタッチして HTTPS を有効にします。詳細については、「Lightsail ロードバランサー用の SSL/TLS 証明書を作成する」を参照してください。証明書の詳細については、「SSL/TLS 証明書」を参照してください。

Lightsail コンソールを使用してセキュリティポリシーを設定する

Lightsail コンソールを使用してセキュリティポリシーを設定するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. TLS セキュリティポリシーを設定するロードバランサーの名前を選択します。
4. [インバウンドトラフィック] タブを選択します。
5. ページの [TLS security protocols] (TLS セキュリティプロトコル) セクションで [Change protocols] (プロトコルを変更) を選択します。
6. [Supported protocols] (サポートされているプロトコル) ドロップダウンメニューで、次のいずれかのオプションを選択します。

   - [TLS バージョン 1.2] — このオプションは最も安全ですが、古いブラウザは接続できない可能性があります。
   - [TLS バージョン 1.0、1.1、および 1.2] — このオプションは、ブラウザとの互換性が最も高くなります。
7. [Save] (保存) を選択して、選択したプロトコルをロードバランサーに適用します。

   変更が有効になるまで、少し時間がかかります。

を使用してセキュリティポリシーを設定する AWS CLI

AWS Command Line Interface (AWS CLI)を使用してセキュリティポリシーを設定するには、次の手順を実行します。これは、`update-load-balancer-attribute` コマンドを使用して行います。

詳細については、「AWS CLI コマンドリファレンス」の「update-load-balancer-attribute」を参照してください。

> **Note**
>
> この手順を続行するLightsail前に、 をインストール AWS CLI して 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 次のコマンドを入力して、ロードバランサーの TLS セキュリティポリシーを変更します。

```
aws lightsail update-load-balancer-attribute --load-balancer-name LoadBalancerName
 --attribute-name TlsPolicyName --attribute-value AttributeValue
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- [*LoadBalancerName*] を、TLS セキュリティポリシーを変更するロードバランサーの名前に置き換えます。
- [*AttributeValue*] を、TLS-2016-08 または TLS-FS-1-2-Res-2019-08 セキュリティポリシーに置き換えます。

> **Note**
>
> コマンドの TlsPolicyName 属性は、ロードバランサーで設定されている TLS セキュリティポリシーを編集することを指定します。

例:

```
aws lightsail update-load-balancer-attribute --load-balancer-name MyLoadBalancer --
attribute-name TlsPolicyName --attribute-value TLS-2016-08
```

変更が有効になるまで、少し時間がかかります。

Lightsail ロードバランサーに HTTP から HTTPS へのリダイレクト

Amazon Lightsail ロードバランサーで HTTPS を設定した後、HTTP から HTTPS へのリダイレクトを設定して、HTTP 接続を使用してウェブサイトまたはウェブアプリケーションを閲覧するユーザーが暗号化された HTTPS 接続に自動的にリダイレクトされるようにすることができます。ロードバランサーの詳細については、「ロードバランサー」を参照してください。

前提条件を満たす

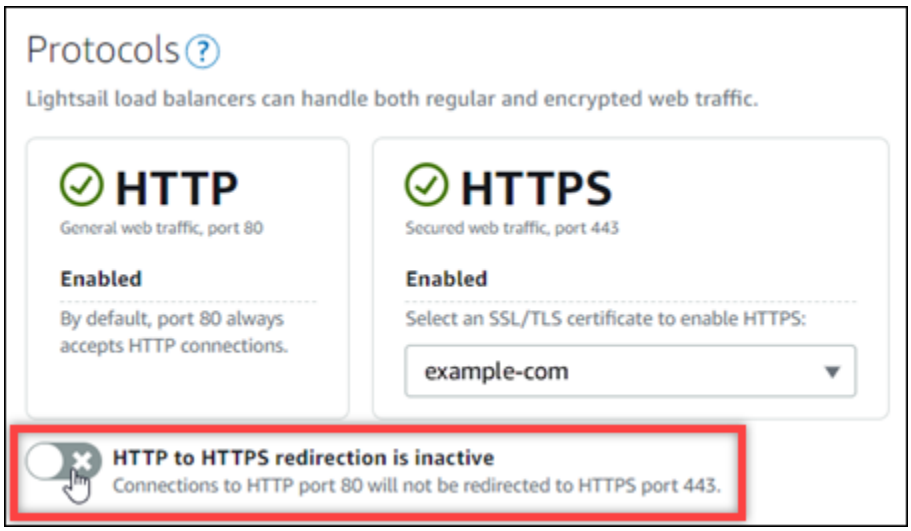
以下の前提条件を完了します (まだの場合)。

- ロードバランサーを作成してインスタンスをアタッチする。詳細については、「ロードバランサーを作成してインスタンスをアタッチする」を参照してください。
- SSL/TLS 証明書を作成し、ロードバランサーにアタッチして HTTPS を有効にします。詳細については、「Lightsail ロードバランサー用の SSL/TLS 証明書を作成する」を参照してください。証明書の詳細については、「SSL/TLS 証明書」を参照してください。

Lightsail コンソールを使用してロードバランサーでの HTTPS リダイレクトを設定する

Lightsail コンソールを使用してロードバランサーで HTTPS リダイレクトを設定するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. HTTPS リダイレクトを設定するロードバランサーの名前を選択します。
4. [インバウンドトラフィック] タブを選択します。
5. ページの [Protocols] (プロトコル) セクションでは、次のいずれかのアクションを実行できます。



- HTTP から HTTPS へのリダイレクトをオンにするには、方向オプションをアクティブに切り替えます。
- HTTP から HTTPS へのリダイレクトをオフにするには、方向オプションを非アクティブに切り替えます。

変更が有効になるまで、少し時間がかかります。

を使用してロードバランサーの HTTP から HTTPS へのリダイレクトを設定する AWS CLI

AWS Command Line Interface () を使用してロードバランサーで HTTPS リダイレクトを設定するには、次の手順を実行しますAWS CLI。これは、`update-load-balancer-attribute` コマンドを使用して行います。詳細については、「AWS CLI コマンドリファレンス」の「update-load-balancer-attribute」を参照してください。

> **Note**
>
> この手順を続行するLightsail前に、 をインストール AWS CLI し、 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 次のコマンドを入力して、ロードバランサーで HTTPS リダイレクトを設定します。

```
aws lightsail update-load-balancer-attribute --load-balancer-name LoadBalancerName
 --attribute-name HttpsRedirectionEnabled --attribute-value AttributeValue
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *LoadBalancerName* を、HTTP から HTTPS へのリダイレクトをアクティブ化または非アクティブ化するロードバランサーの名前に置き換えます。
- *AttributeValue* を、リダイレクトをアクティブ化する `true`、またはリダイレクトを非アクティブ化する `false` に置き換えます。

> **Note**
>
> コマンドの `HttpsRedirectionEnabled` 属性は、指定されたロードバランサーについて HTTPS リダイレクトが有効か無効かを編集することを指定します。

例:

- ロードバランサーで HTTP から HTTPS へのリダイレクトをアクティブ化するには、次の手順を実行します。

```
aws lightsail update-load-balancer-attribute --load-balancer-name MyLoadBalancer
 --attribute-name HttpsRedirectionEnabled --attribute-value true
```

- ロードバランサーで HTTP から HTTPS へのリダイレクトを非アクティブ化するには、次の手順を実行します。

```
aws lightsail update-load-balancer-attribute --load-balancer-name MyLoadBalancer
 --attribute-name HttpsRedirectionEnabled --attribute-value false
```

変更が有効になるまで、少し時間がかかります。

Lightsail ロードバランサーのセッション永続性を有効にする

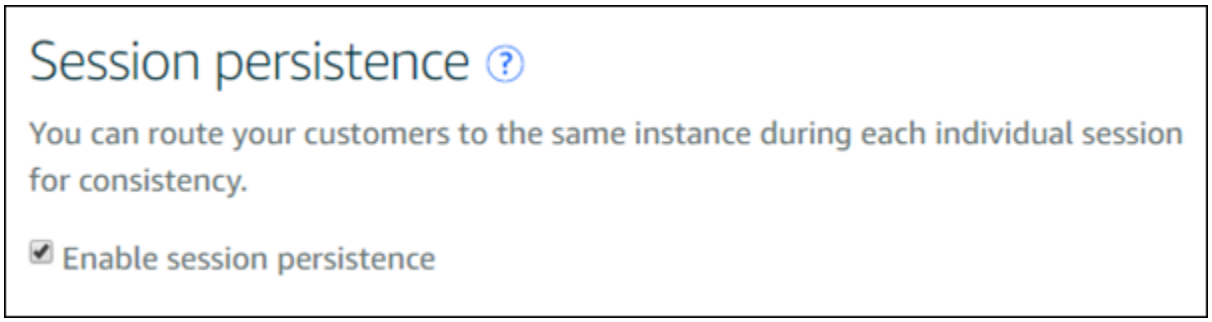
ユーザーのセッション永続性を有効にすることができます。これは、ユーザーのブラウザでセッション情報をローカルに保存する場合に役立ちます。たとえば、Amazon Lightsail にショッピングカート

のある Magento e コマースアプリケーションを実行しているとします。セッション永続性を有効にした場合、ユーザーがショッピングカートに商品を追加してサイトから離れても、戻ってくるとカートに商品が残っています。

AWS Command Line Interface (AWS CLI) または Lightsail API を使用して Cookie の期間を調整することもできます。

セッション永続性を有効にする

1. 左側のナビゲーションペインで、[ネットワーク] を選択します。
2. ロードバランサーを選択して管理します。
3. [インバウンドトラフィック] タブを選択します。
4. [セッション永続性を有効にする] を選択します。



Cookie の有効期間を調整する

また、永続的なセッションの Cookie の有効期間を調整することもできます。これは、特に長い有効期間や短い有効期間が必要な場合に役立ちます。たとえば、多くの e コマースサイト期間では有効期間が非常に長くなっています。これにより、顧客がサイトを離れて戻ってきても、ショッピングカート内の商品が失われません。

まだ設定していない場合は、 をセットアップ AWS CLI して設定します。

AWS Command Line Interface と連携するための Amazon Lightsail の設定

1. コマンドプロンプトまたはターミナルウィンドウを開きます。
2. 次の AWS CLI コマンドを入力して、Cookie の期間を 3 日間 (259,200 秒) に延長します。

```
aws lightsail update-load-balancer-attribute --load-balancer-name LoadBalancerName
 --attribute-name SessionStickiness_LB_CookieDurationSeconds --attribute-value
 259200
```

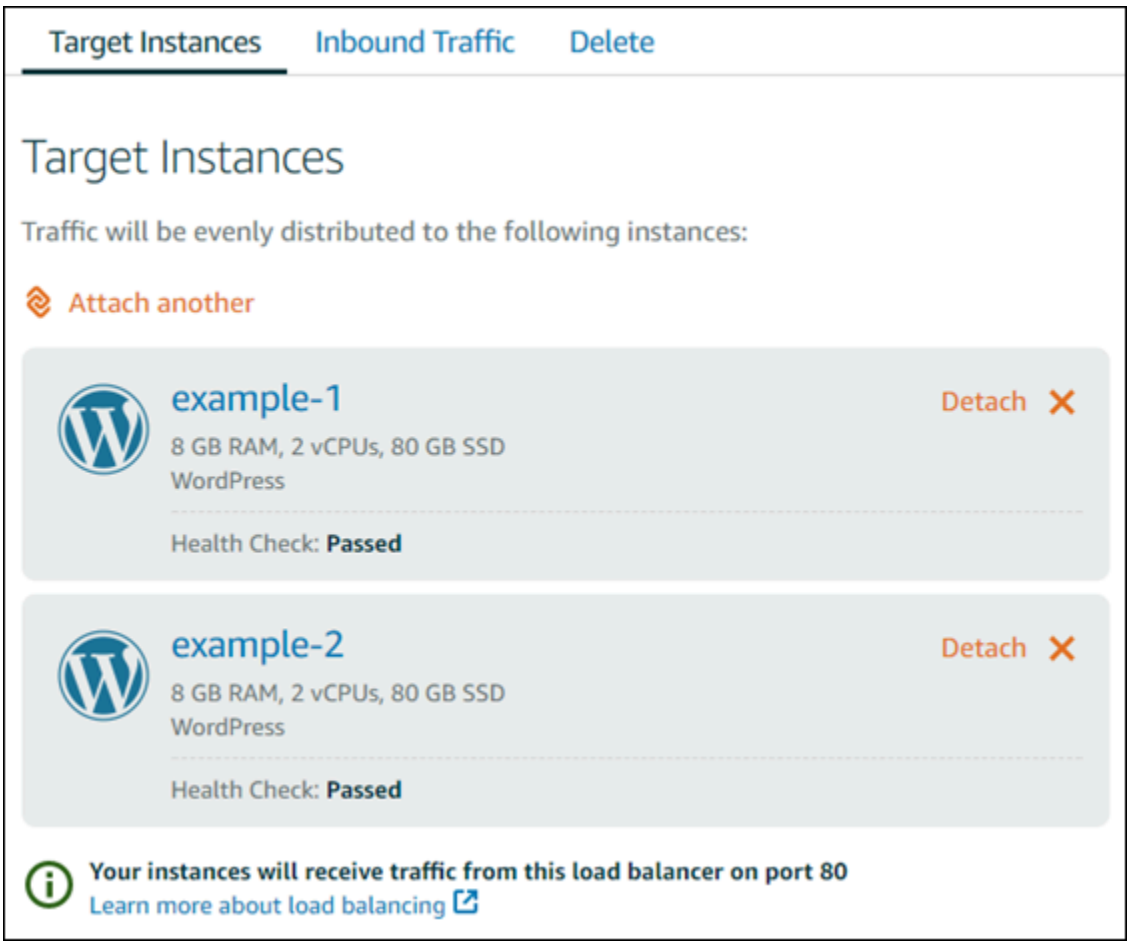
コマンドで、*LoadBalancerName* をロードバランサーの名前に置き換えます。

成功すると、次のような応答が表示されます。


```
{
    "operations": [
        {
            "status": "Succeeded",
            "resourceType": "LoadBalancer",
            "isTerminal": true,
            "operationDetails": "SessionStickiness_LB_CookieDurationSeconds",
            "statusChangedAt": 1511758936.174,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "operationType": "UpdateLoadBalancerAttribute",
            "resourceName": "example-load-balancer",
            "id": "681c2bd9-9a51-402b-8ad2-12345EXAMPLE",
            "createdAt": 1511758936.174
        }
    ]
}
```


ヘルスチェックに対する Lightsail ロードバランサーの設定

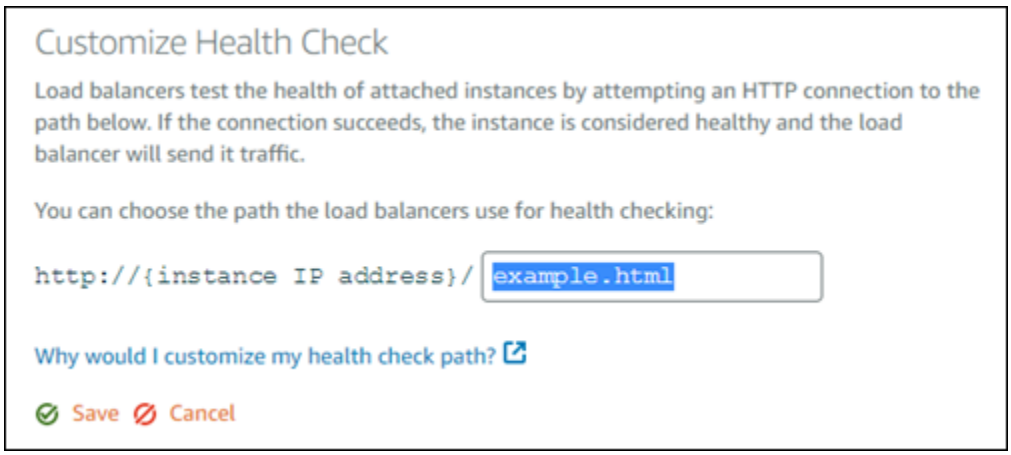
ヘルスチェックは、Lightsail インスタンスをロードバランサーにアタッチするとすぐに開始され、その後 30 秒ごとに実行されます。ヘルスチェックのステータスを表示するには、ロードバランサーの管理ページを参照してください。



ヘルスチェックのパスをカスタマイズする

ヘルスチェックのパスをカスタマイズすることが必要な場合があります。たとえば、ホームページのロードに時間がかかる場合や多くの画像が使用されている場合、すぐにロードされる別のページをチェックするように Lightsail を設定します。

1. 左側のナビゲーションペインで、[ネットワーク] を選択します。
2. ロードバランサーを選択して管理します。
3. [ターゲットインスタンス] タブで [ヘルスチェックのカスタマイズ] を選択します。
4. ヘルスチェックの有効なパスを入力し、[保存] を選択します。



ヘルスチェックメトリクス

次のメトリックスはヘルスチェックの問題を診断するのに役立ちます。 AWS Command Line Interface または Lightsail API を使用して、特定のヘルスチェックメトリクスに関する情報を返します。

- **ClientTLSNegotiationErrorCount** – クライアントで開始された TLS 接続のうち、ロードバランサーとのセッションの確立に失敗したものの数。暗号化またはプロトコルの不一致が原因である場合があります。

  Statistics: 最も有用な統計は Sum です。
- **HealthyHostCount** – 正常と見なされるターゲットインスタンスの数。

  Statistics: 最も有用な統計は Average、Minimum、Maximum です。
- **UnhealthyHostCount** – 異常と見なされるターゲットインスタンスの数。

  Statistics: 最も有用な統計は Average、Minimum、Maximum です。
- **HTTPCode_LB_4XX_Count** – ロードバランサーから送信される HTTP 4XX クライアントエラーコードの数。リクエストの形式が不正な場合、または不完全な場合は、クライアントエラーが生成されます。それらのリクエストはターゲットインスタンスで受信されません。この数には、ターゲットインスタンスによって生成される応答コードは含まれません。

  Statistics: 最も有用な統計は Sum です。Minimum、Maximum、および Average はすべて 1 を返すことに注意してください。
- **HTTPCode_LB_5XX_Count** – ロードバランサーから送信される HTTP 5XX サーバーエラーコードの数。この数には、ターゲットインスタンスによって生成される応答コードは含まれません。

Statistics: 最も有用な統計は Sum です。Minimum、Maximum、および Average はすべて 1 を返すことに注意してください。Minimum、Maximum、および Average はすべて 1 を返すことに注意してください。

- **HTTPCode_Instance_2XX_Count** – ターゲットインスタンスで生成された HTTP 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。

  Statistics: 最も有用な統計は Sum です。Minimum、Maximum、および Average はすべて 1 を返すことに注意してください。

- **HTTPCode_Instance_3XX_Count** – ターゲットインスタンスで生成された HTTP 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。

  Statistics: 最も有用な統計は Sum です。Minimum、Maximum、および Average はすべて 1 を返すことに注意してください。

- **HTTPCode_Instance_4XX_Count** – ターゲットインスタンスで生成された HTTP 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。

  Statistics: 最も有用な統計は Sum です。Minimum、Maximum、および Average はすべて 1 を返すことに注意してください。

- **HTTPCode_Instance_5XX_Count** – ターゲットインスタンスで生成された HTTP 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。

  Statistics: 最も有用な統計は Sum です。Minimum、Maximum、および Average はすべて 1 を返すことに注意してください。

- **InstanceResponseTime** – ロードバランサーからリクエストを送信してから、ターゲットインスタンスからの応答を受信するまでの経過時間 (秒)。

  Statistics: 最も有用な統計は Average です。

- **RejectedConnectionCount** – ロードバランサーが接続の最大数に達したため、拒否された接続の数。

  Statistics: 最も有用な統計は Sum です。

- **RequestCount** – IPv4 経由で処理されたリクエストの数。この数には、ロードバランサーのターゲットインスタンスによって生成されたレスポンスを含むリクエストのみが含まれます。

  Statistics: 最も有用な統計は Sum です。Minimum、Maximum、および Average はすべて 1 を返すことに注意してください。

トピック

- Lightsail ロードバランサーのヘルスチェックの設定

Lightsail ロードバランサーのヘルスチェックの設定

デフォルトでは、Lightsail はウェブアプリケーションのルート ("/") でインスタンスに対するヘルスチェックを実行します。ヘルスチェックは、ロードバランサーから正常なインスタンスにのみリクエストを送信できるように、登録されたインスタンスのヘルス状態をモニタリングするために使用されます。ヘルスチェックは、インスタンスをロードバランサーにアタッチするとすぐに開始します。

以下のいずれかのステータスが返されます。

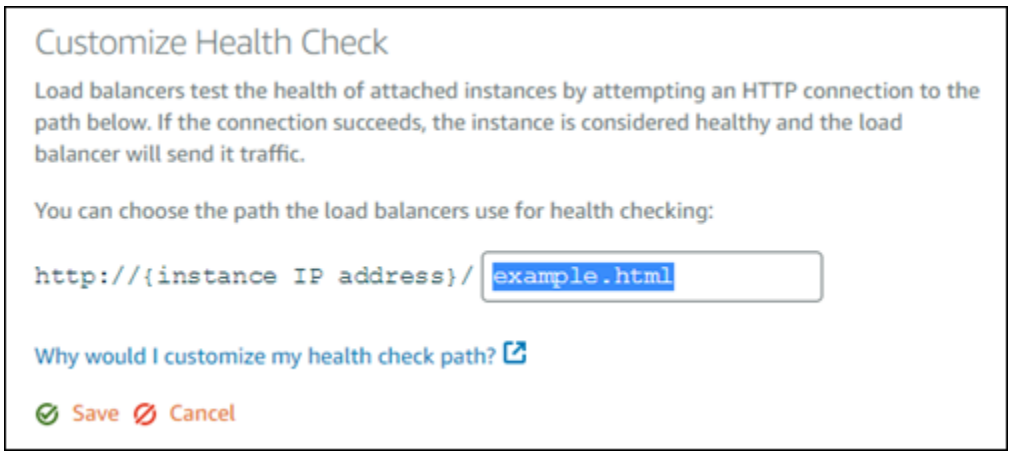
- 成功
- 失敗

ヘルスチェックが失敗した場合、 AWS Command Line Interface または Lightsail API を使用することで問題の原因を調べることができます。詳細については、トラブルシューティングガイドを参照してください。

ヘルスチェックのパスをカスタマイズする

ヘルスチェックのパスをカスタマイズすることが必要な場合があります。たとえば、ホームページのロードに時間がかかる場合や多くの画像が使用されている場合、すぐにロードされる別のページをチェックするように Lightsail を設定します。

1. 左側のナビゲーションペインで、[ネットワーク] を選択します。
2. ロードバランサーを選択して管理します。
3. [ターゲットインスタンス] タブで [ヘルスチェックのカスタマイズ] を選択します。
4. ヘルスチェックの有効なパスを入力し、[保存] を選択します。



Lightsail ロードバランサーからインスタンスをデタッチする

インスタンスを Amazon Lightsail ロードバランサーにアタッチする必要がなくなった場合、デタッチすることができます。ロードバランサーから Lightsail インスタンスをデタッチときは、デタッチする前に指定したインスタンスがこれ以上必要なくなるまで待機します。

1. 左側のナビゲーションペインで、[ネットワーク] を選択します。
2. 管理するロードバランサーを選択します。
3. [ターゲットインスタンス] タブで、デタッチするロードバランサーの横にある [デタッチ] を選択します。

Lightsail ロードバランサーを削除する

不要になった場合は、Lightsail ロードバランサーを削除できます。ロードバランサーを削除すると、それにアタッチされている Lightsail インスタンスもすべてデタッチされますが、Lightsail インスタンスは削除されません。SSL/TLS 証明書を使用して暗号化された (HTTPS) トラフィックを有効にした場合、ロードバランサーを削除するとロードバランサーに関連付けられた SSL/TLS 証明書も完全に削除されます。

> **Important**
>
> Lightsail ロードバランサーとそれに関連する証明書を削除すると、元に戻すことができません。

1. 左側のナビゲーションペインで、[ネットワーク] を選択します。

2. 削除するロードバランサーを選択します。
3. [削除] を選択します。
4. [ロードバランサーの削除] を選択します。
5. [Yes, delete] (はい、削除します) を選択します。

Lightsail コンテンツ配信ディストリビューションでウェブコンテンツをグローバルに提供します。

Lightsail ディストリビューションは、グローバルに分散したサーバーのネットワーク、エッジロケーションを使用して、コンテンツをすばやくユーザーに配信します。ディストリビューションを使用するには、まず 1 つの Lightsail インスタンスまたはコンテナサービス上、または Lightsail ロードバランサーにアタッチされた複数のインスタンス上でウェブサイトかウェブアプリケーションを作成してホストするか、Lightsail バケットに静的コンテンツを保存します。次に、Lightsail ディストリビューションを作成して、インスタンス、コンテナサービス、ロードバランサー、またはバケットからコンテンツをプルし、キャッシュし、提供するように設定します。インスタンス、コンテナサービス、ロードバランサー、バケット (ディストリビューションのオリジンとも呼ばれる) はコンテンツの決定的なソースです。

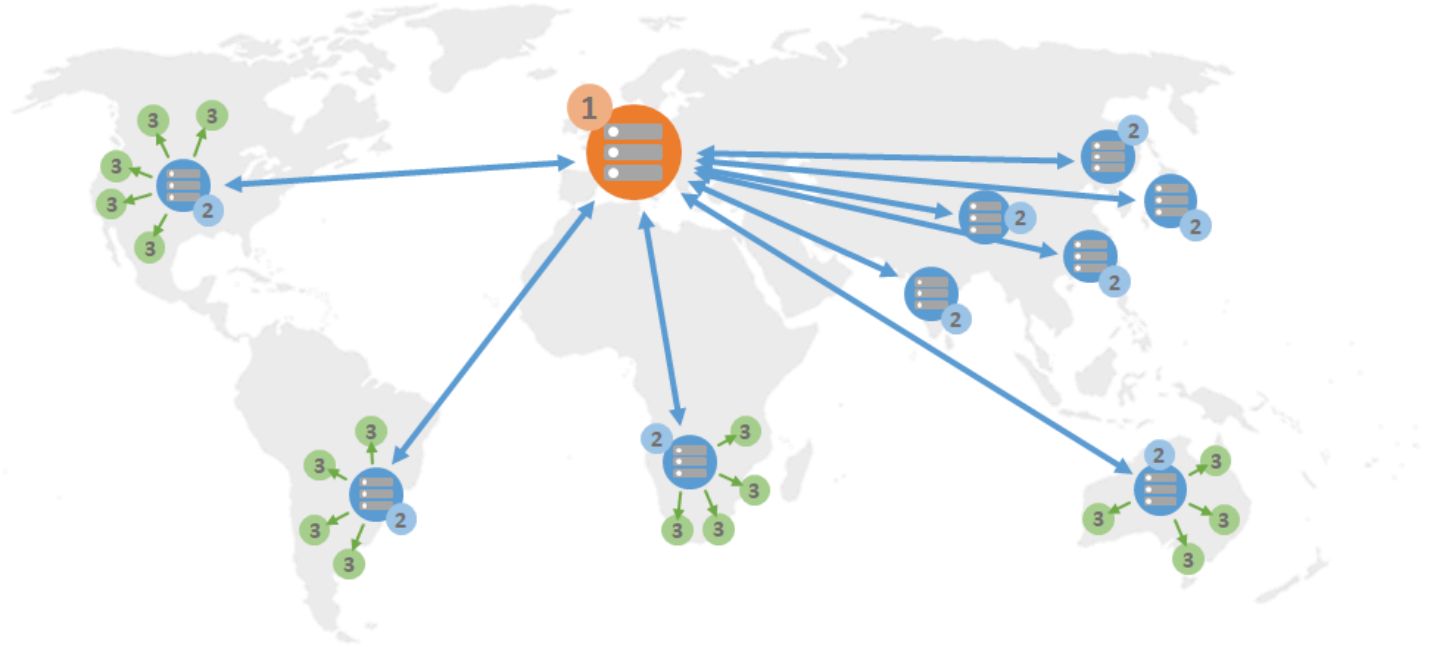
ユーザーがディストリビューションを通じて提供されているウェブサイトにアクセスしてコンテンツをリクエストすると、リクエストはレイテンシーの点から最も近い場所にルーティングされます。次に、ディストリビューションは以下のいずれかのアクションを実行します。

- すでにコンテンツがエッジロケーション内にキャッシュされている場合、ディストリビューションはそのコンテンツをユーザーに提供します。
- コンテンツがそのエッジロケーションにまだキャッシュされていない場合、ディストリビューションは指定されたオリジンからコンテンツを取得し、キャッシュし、ユーザーに提供します。

コンテンツは、ディストリビューションに指定するキャッシュのライフスパン (存続時間) の間、エッジロケーションにキャッシュされるため、同じロケーションにある他のリクエストは直ちに満たされます。キャッシュされたコンテンツは、キャッシュのライフスパンに達すると、エッジロケーションから削除されます。次回、コンテンツリクエストがエッジロケーションにルーティングされる際に、ディストリビューションがコンテンツを取得し、キャッシュ、および配信します。

以下の図表では、

- 1 は、ウェブサイトをホストしている Lightsail インスタンスまたはコンテナサービス、インスタンスがアタッチされているロードバランサー、静的コンテンツをホストしているバケットなど、ディストリビューションのオリジンを示します。
- 2 は、ディストリビューション、またはオリジンからコンテンツをプル、キャッシュし、配信するエッジロケーションを示します。
- 3 は、エッジロケーションからコンテンツを提供するユーザーを示します。

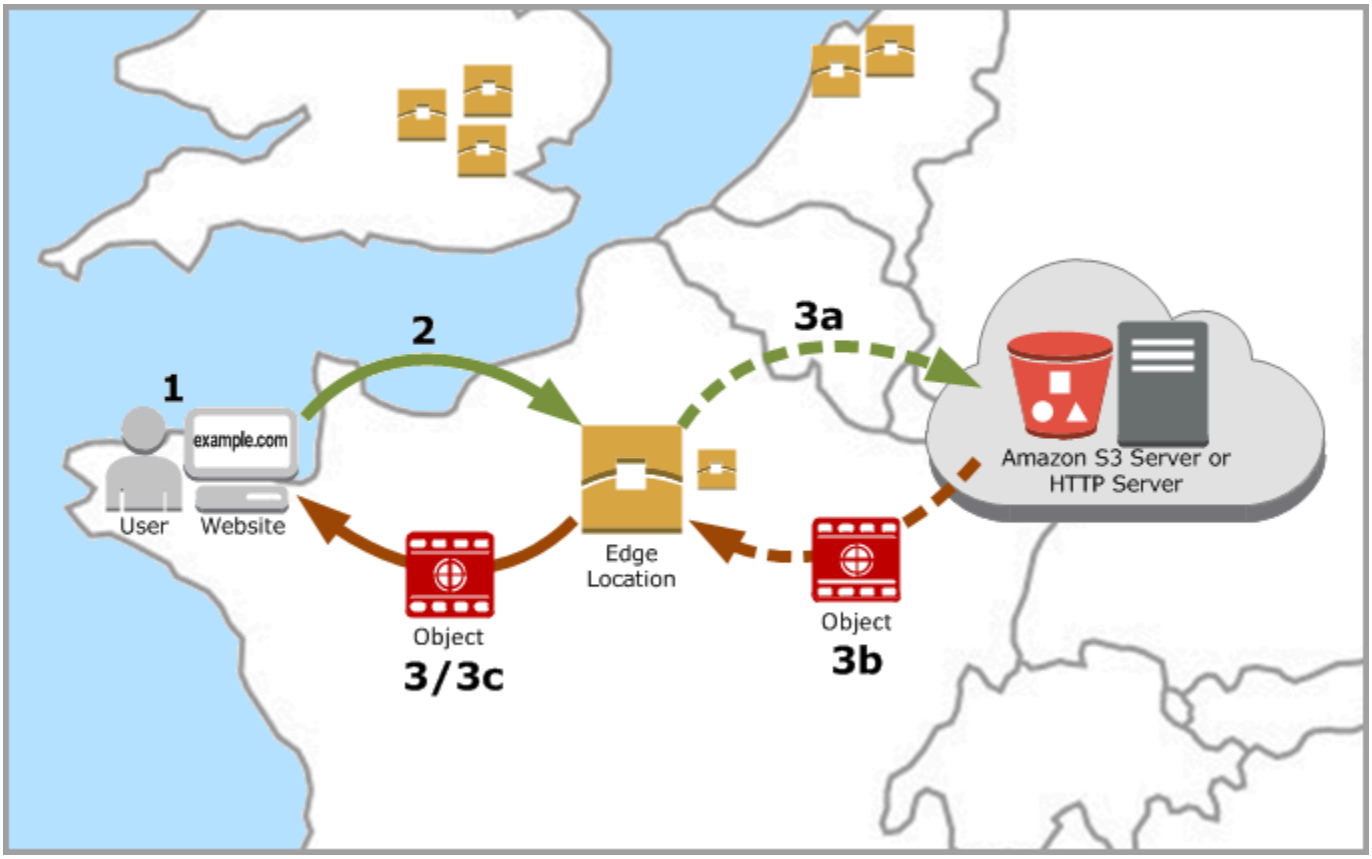


> **Note**
>
> この図表は図示のみを目的とし、実際のエッジロケーションは表示していません。エッジロケーションの詳細については、このガイドの後半のエッジロケーションと IP アドレスの範囲を参照してください。

例えば、ウェブサイトがフランスでホストされており、フランスの他の地域のユーザーがコンテンツを表示したい場合、ページはミリ秒単位の時間でロードされます。

訪問者が近くにいない場合は少し複雑です。

オーストラリアのユーザーがコンテンツを表示したい場合、ブラウザはフランスにあるサーバーからそれを取得し、数千マイル離れた場所にいるそのユーザーに表示する必要があります。異なる国のユーザーが同時に同じコンテンツをリクエストすると、それらのリクエストによってサーバーに負荷がかかり、コンテンツのロードと提供に時間がかかります。これは、エンドユーザーのためにコンテンツがロードされる速度に影響します。



CDN は、エッジロケーションでウェブサイトのコンテンツをキャッシュすることで、この状況を解決します。このコンテンツ提供方法は、1 つの中央リソースからコンテンツを提供する従来の方法よりも高速でかつ効率的です。ビューワーがウェブサイトで、またはアプリケーション経由でリクエストを実行すると、DNS はユーザーのリクエストに対応できる最適なロケーションにリクエストをルーティングします。すべてのユーザーが遠くにある同じ中央リソースにアクセスするのとは対照的に、ユーザーは近くの場所からコンテンツにアクセスします。

ユースケース

高速で安全なウェブサイトを配信する

Lightsail 配信は、世界中のビューワーへのコンテンツ (例えば、ウェブサイトページ、イメージ、スタイルシート、JavaScript など) の配信を高速化します。ディストリビューションを使用することで、 AWS バックボーンネットワークおよびエッジサーバーを活用でき、ウェブサイトを閲覧するビューワーに、高速で、安全で、信頼性の高い体験を提供できます。

サイトのセキュリティを改善する

TLS ターミネーションを利用して、ウェブサイトを強化し、パフォーマンスを改善します。これは、暗号化処理をディストリビューションにオフロードすることで、オリジンのロードを軽減します。登録済みドメイン名と Lightsail SSL/TLS 証明書を使用して、ディストリビューションの Hypertext Transfer Protocol Secure (HTTPS) を有効化できます。ユーザーはディストリビューションへの暗号化された HTTPS 接続を確立し、HTTP を使用してオリジンからコンテンツを取得します。

アプリケーションの最適化

WordPress や静的なウェブサイトなど、さまざまなアプリケーション向けにディストリビューションを簡単に最適化できます。ディストリビューションを使用してコンテンツをキャッシュし、配信すると、ほとんどのリクエストがインスタンス、コンテナサービス、ロードバランサー、またはバケットではなくディストリビューションによって処理されるため、オリジンへの負荷も軽減されます。

ディストリビューションを設定する

Lightsail インスタンスとディストリビューションを使用してウェブサイトやウェブアプリケーションを提供するためにの具体的なステップです。

1. ディストリビューションでインスタンス、コンテナサービス、バケットのどれを使用するか応じて、次のいずれかを実行します。
   - コンテンツをホストする Lightsail インスタンスを作成します。インスタンスは、ディストリビューションのオリジンとして機能します。オリジンサーバーには、コンテンツのオリジナルの最終バージョンが保存されます。詳細については、「インスタンスを作成する」を参照してください。

     Lightsail 静的 IP をインスタンスにアタッチします。インスタンスを停止して起動すると、インスタンスのデフォルトのパブリック IP アドレスが変更されるため、ディストリビューションとオリジンインスタンスの接続が切断されます。インスタンスを停止して開始しても、静的 IP は変更されません。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。

     インスタンスにコンテンツとファイルをアップロードします。ファイル (オブジェクト) には、ウェブページ、イメージ、メディアファイルに限らず、HTTP 経由で提供できるもの全てが含まれます。

- ウェブサイトやウェブアプリケーションをホストする Lightsail コンテナサービスを作成します。コンテナサービスは、ディストリビューションのオリジンとして機能します。オリジンサーバーには、コンテンツのオリジナルの最終バージョンが保存されます。詳細については、「Amazon Lightsail コンテナサービスを作成する」を参照してください。
- Lightsail バケットを作成して、静的コンテンツを保存します。バケットは、ディストリビューションのオリジンとして機能します。オリジンは、オリジナル、最終バージョンコンテンツを保存します。詳細については、「バケットの作成」を参照してください。

  Lightsail コンソール、 AWS Command Line Interface (AWS CLI)、および AWS APIs。ファイルのアップロードに関する詳細については、「バケットにファイルをアップロードする」を参照してください。

2. (オプション) インスタンスでホストされているウェブサイトが耐障害性を必要とする場合、Lightsail ロードバランサーを作成します。次に、インスタンスの複数のコピーをロードバランサーにアタッチします。インスタンスをオリジンとして設定する代わりに、ロードバランサー（複数のインスタンスが添付されている）をディストリビューションのオリジンとして設定することができます。詳細については、「ロードバランサーを作成してインスタンスをアタッチする」を参照してください。
3. Lightsail ディストリビューションを作成し、インスタンス、コンテナサービス、ロードバランサー、またはバケットをオリジンとして設定します。同時に、コンテンツのキャッシュライフスパン、ウェブサイトまたは ウェブアプリケーションのどの要素をキャッシュするかなどの、詳細を指定します。詳細については、「ディストリビューションを作成する」を参照してください。
4. (オプション) ディストリビューションのオリジンが WordPress インスタンスである場合は、インスタンスの WordPress 設定ファイルを編集して、WordPress ウェブサイトをディストリビューションと連携させる必要があります。詳細については、「ディストリビューションと動作するように WordPress インスタンスを設定する」を参照してください。
5. (オプション) Lightsail コンソールでドメインの DNS を管理するために Lightsail DNS ゾーンを作成します。これにより、Lightsail リソースにドメインを簡単にマッピングすることができます。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。または、現在ホストされているドメインの DNS をホストし続けることもできます。
6. ディストリビューションで使用するドメインの Lightsail SSL/TLS 証明書を作成します。Lightsail ディストリビューションには HTTPS が必要なため、ディストリビューションで使用する前に、ドメインの SSL/TLS 証明書を要求する必要があります。詳細については、「ディストリビューションの SSL/TLS 証明書を作成する」を参照してください。
7. ディストリビューションのカスタムドメインを有効化して、登録済みドメイン名をディストリビューションで使用できるようにします。カスタムドメインを有効化するには、ドメイン用に作

成したLightsail SSL/TLS 証明書を指定する必要があります。これにより、ドメインがディストリビューションに追加され、HTTPS が有効になります。詳細については、「ディストリビューション用のカスタムドメインを有効にする」を参照してください。

8. ドメインの DNS にエイリアスレコードを追加して、ドメインのトラフィックをディストリビューションにルーティングします。エイリアスレコードを追加したら、ドメインにアクセスしたユーザーはディストリビューションを通じてルーティングされます。詳細については、「ドメインをディストリビューションにポイントする」を参照してください。
9. ディストリビューションがコンテンツをキャッシュしていることをテストします。詳細については、「ディストリビューションをテストする」を参照してください。

エッジロケーションと IP アドレス範囲

Lightsail ディストリビューションは、Amazon CloudFront と同じエッジサーバーと IP アドレス範囲を使用します。CloudFront エッジサーバーの場所の一覧については、「Amazon CloudFront 製品の詳細情報ページ」をご覧ください。CloudFront の IP 範囲のリストについては、「CloudFront のグローバル IP リスト」を参照してください。

Lightsail コンテンツ配信ネットワークディストリビューションを作成する

このガイドでは、Lightsail コンソールを使用して Amazon Lightsail ディスディストリビューションを作成する方法と、ディストリビューションの設定について説明します。ディストリビューションの詳細については、「コンテンツ配信ネットワークディストリビューション」を参照してください。

目次

- 前提条件
- オリジンリソース
- オリジンプロトコルポリシー
- キャッシュ動作とキャッシュプリセット
- WordPress のキャッシュプリセットに最適
- デフォルトの動作
- ディレクトリとファイルの上書き
- ゲームのアドバンスト設定
- ディストリビューションプラン

- ディストリビューションの作成
- 次のステップ

前提条件

ディストリビューションの作成のスタート前に、前提条件として次の作業を完了してください。

1. ディストリビューションでインスタンス、コンテナサービス、バケットのどれを使用するか応じて、次のいずれかを実行します。
   - コンテンツをホストする Lightsail インスタンスを作成します。インスタンスは、ディストリビューションのオリジンとして機能します。オリジンサーバーには、コンテンツのオリジナルの最終バージョンが保存されます。詳細については、「インスタンスを作成する」を参照してください。

     Lightsail 静的 IP をインスタンスにアタッチします。インスタンスを停止して起動すると、インスタンスのデフォルトのパブリック IP アドレスが変更されるため、ディストリビューションとオリジンインスタンスの接続が切断されます。インスタンスを停止して開始しても、静的 IP は変更されません。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。

     インスタンスにコンテンツとファイルをアップロードします。ファイル (オブジェクト) には、ウェブページ、イメージ、メディアファイルに限らず、HTTP 経由で提供できるもの全てが含まれます。
   - ウェブサイトやウェブアプリケーションをホストする Lightsail コンテナサービスを作成します。コンテナサービスは、ディストリビューションのオリジンとして機能します。オリジンサーバーには、コンテンツのオリジナルの最終バージョンが保存されます。詳細については、「Amazon Lightsail コンテナサービスの作成」を参照してください。
   - Lightsail バケットを作成して、静的コンテンツを保存します。バケットは、ディストリビューションのオリジンとして機能します。オリジンは、オリジナル、最終バージョンコンテンツを保存します。詳細については、「バケットの作成」を参照してください。

     Lightsail コンソール、 AWS Command Line Interface (AWS CLI)、および AWS APIs。ファイルのアップロードに関する詳細については、「バケットにファイルをアップロードする」を参照してください。
2. (オプション) ウェブサイトに耐障害性が必要な場合、Lightsailロードバランサーを作成してください。次に、インスタンスの複数のコピーをロードバランサーにアタッチします。インスタンスをオリジンとして設定する代わりに、ロードバランサー (複数のインスタンスが添付されている)

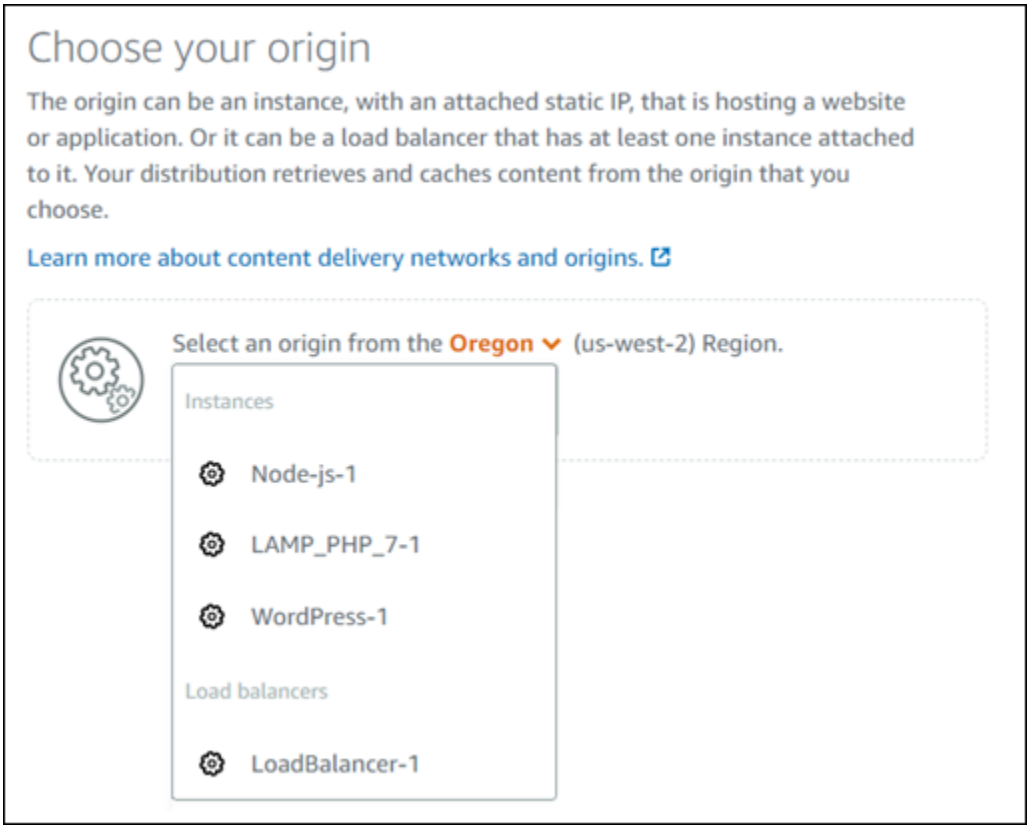
をディストリビューションのオリジンとして設定することができます。詳細については、「ロードバランサーを作成してインスタンスをアタッチする」を参照してください。

オリジンリソース

オリジンとは、あなたのディストリビューションのためのコンテンツの決定的なソースです。ディストリビューションを作成する際に、ウェブサイトまたはウェブアプリケーションのコンテンツをホストする Lightsail インスタンス、コンテナサービス、バケット、またはロードバランサー (1 つまたは複数のインスタンスがアタッチされている) を選択します。

> **Note**
>
> IPv6 専用インスタンスは、現時点では Lightsail コンテンツ配信ネットワーク (CDN) ディストリビューションのオリジンとして設定できません。

1 つのディストリビューションにつき 1 つのオリジンのみ選択できます。ディストリビューションを作成後、いつでもオリジンを変更できます。詳細については、「ディストリビューションのオリジンを変更する」を参照してください。



オリジンプロトコルポリシー

オリジンプロトコルポリシーは、オリジンからコンテンツを引き出す時にディストリビューションが使用するプロトコルポリシーです。ディストリビューションのオリジンを選択した後、オリジンからコンテンツを引き出す際に、Hypertext Transfer Protocol (HTTP) か、Hypertext Transfer Protocol Secure (HTTPS) どちらを使用すべきか決めます。オリジンが HTTPS 用に設定されていない場合は、HTTP を使用する必要があります。

ディストリビューションに対して、次のいずれかのオリジンプロトコルポリシーを選択できます。

- HTTP Only - オリジンへのアクセスに HTTP のみを使用します。これはデフォルトの設定です。
- HTTPS Only (HTTPS のみ) : オリジンへのアクセスに HTTPS のみを使用します。

オリジンプロトコルポリシーを編集するステップは、このガイドで後述する「ディストリビューションを作成する」セクションを参照してください。

> **Note**
>
> Lightsail バケットをディストリビューションのオリジンとして選択すると、[Origin protocol policy] (オリジンプロトコルポリシー) はデフォルトで [HTTPS only] (HTTPS のみ) に設定されます。バケットがディストリビューションのオリジンである場合、オリジンプロトコルポリシーを変更することはできません。

キャッシュ動作とキャッシュプリセット

キャッシュプリセットは、オリジンでホストするコンテンツの種類に応じて、ディストリビューションの設定を自動的に設定します。例えば、静的コンテンツに最適を選択すると、プリセットは、静的ウェブサイトで最適に動作する設定にディストリビューションを自動的に設定します。ウェブサイトが WordPress インスタンスでホストされている場合は、WordPress に最適プリセットを選択すると、WordPress ウェブサイトで動作するようにディストリビューションが自動的に設定されます。

> **Note**
>
> Lightsail バケットをディストリビューションのオリジンとして選択すると、キャッシュプリセットのオプションは利用できません。バケットに保存されている静的コンテンツに最適なディストリビューション設定が自動的に適用されます。

ディストリビューション用に、以下のいずれかのキャッシュプリセットを選択できます。

- 静的コンテンツに最適 - このプリセットは、ディストリビューションをすべてをキャッシュするように設定されます。このプリセットは、オリジンで静的コンテンツ (静的 HTML ページなど) をホストする場合や、ウェブサイトにアクセスするユーザーごとに変更されないコンテンツをホストする場合に最適です。このプリセットを選択すると、ディストリビューション上のすべてのコンテンツがキャッシュされます。
- 動的コンテンツに最適 - このプリセットは、ディストリビューションをディストリビューションを作成するページのセクションにあるディレクトリとファイルの上書きのキャッシュで指定されていないもの以外をキャッシュしないように設定されます。詳細については、ディレクトリとファイルの上書きを後ほど参照してください。このプリセットは、オリジンで動的なコンテンツや、ウェブサイトやウェブアプリケーションにアクセスするユーザーごとに変化するコンテンツをホストする場合に最適です。
- WordPress に最適 - このプリセットは、ディストリビューションを`wp-includes/`と`wp-content/` WordPress インスタンスのディレクトリーにあるファイル以外をキャッシュしないように設定されます。このプリセットは、オリジンのインスタンスが WordPress Certified by Bitnami and Automatic ブループリントの場合に最適です (マルチサイトブループリントを除く) 。このプリセットの詳細については、 WordPress の最適キャッシュプリセット

> **Note**
>
> カスタム設定プリセットは選択できません。プリセットはプリセットを選択後、自動で選択されますが、ディストリビューションの設定を手動で変更します。

Lightsail コンソールのみでキャッシュのプリセットを指定できます。Lightsail API、 AWS CLI、および SDKs を使用して指定することはできません。

WordPress の最適キャッシュプリセット

選択したインスタンスが、 WordPress Certified by Bitnami and Automattic 設計図をディストリビューションのオリジンとして使用している場合、Lightsail はディストリビューションのキャッシュプリセットに [WordPress に最適] を適用するかを確認するプロンプトが表示されます。プリセットを適用すると、ディストリビューションは WordPress のウェブサイトに最適した動作に自動的に設定されます。他に適用しなければいけないディストリビューション設定はありません。WordPress に最適は、`wp-includes/`内のファイルおよび`wp-content/` WordPress ウェブサイトのディレクトリ以外をキャッシュしないようにプリセットされます。また、毎日キャッシュをクリアするよ

うにディストリビューションを設定し（キャッシュ寿命は 1 日）、すべての HTTP メソッドを許可し、Host ヘッダーのみを転送し、Cookieを転送せず、すべてのクエリ文字列を転送します。

> **Important**
>
> WordPress ウェブサイトをディストリビューションで使用できるようにするには、インスタンスの WordPress 設定ファイルを編集する必要があります。詳細については、「ディストリビューションと動作するように WordPress インスタンスを設定する」を参照してください。

デフォルトの 動作

デフォルトの動作は、ディストリビューションがコンテンツキャッシュをどのように処理するかを指定します。ディストリビューションのデフォルトの動作は、選択したキャッシュプリセットによって自動的に決定されます。別のデフォルト動作を選択した場合、キャッシュプリセットは自動的にカスタム設定に変更されます。

> **Note**
>
> Lightsail バケットをディストリビューションのオリジンとして選択した場合、デフォルトの動作オプションは使用できません。バケットに保存されている静的コンテンツに最適ディストリビューション設定が自動的に適用されます。

ディストリビューションでは、以下のいずれかのデフォルト動作を選択できます。

- すべてをキャッシュする - この動作は、ウェブサイト全てを静的コンテンツとしてキャッシュ、対応するようにディストリビューションを設定します。このオプションは、閲覧者によって変更されないコンテンツをオリジンがホストしている場合、または ウェブサイトが cookie、ヘッダー、またはクエリ文字列を使用してコンテンツをパーソナライズしない場合に最適です。
- 何もキャッシュしない - この動作は、指定したオリジンファイルとフォルダーパスのみをキャッシュするようにディストリビューションを設定します。このオプションは、ウェブサイトやウェブアプリケーションが cookie、ヘッダー、クエリ文字列を使用して、個々のユーザー向けにコンテンツをパーソナライズする場合に最適です。このオプションを選択すると、キャッシュするには、ディレクトリとファイルパスの上書きを指定する必要があります。

ディレクトリとファイルの上書き

ディレクトリとファイルの上書きを使用して、選択したデフォルトの動作を上書きしたり、例外を追加することが可能です。例えば、すべてをキャッシュするを選択した場合、上書きを使用して、ディストリビューションがキャッシュしないディレクトリ、ファイル、またはファイルの種類を指定します。代わりに、何もキャッシュしないを選択した場合、上書きを使用して、ディストリビューションがキャッシュするディレクトリ、ファイル、またはファイルの種類を指定します。

ディレクトリとファイルの上書きセクションで、キャッシュするディレクトリまたはファイルへのパスを指定するか、キャッシュしないかを指定できます。アスタリスク記号を使用して、ワイルドカードディレクトリ (path/to/assets/*)、ファイルタイプ (*.html,*jpg,*js)を指定する。ディレクトリとファイルパスでは、大文字と小文字が区別されます。

> **Note**
>
> Lightsail バケットをディストリビューションのオリジンとしてを選択すると、ディレクトリとファイルの上書きオプションは利用できません。選択したバケットに保存されているものすべてがキャッシュされます。

以下は、ディレクトリとファイルの上書きを指定する方法の例です。

- 以下を指定することによって、Lightsail インスタンスで実行されている Apache ウェブサーバーのドキュメントルートのすべてのファイルをキャッシュします。

  ```
  var/www/html/
  ```

- Apache ウェブサーバーのドキュメントルートのインデックスページのみをキャッシュするには、次のファイルを指定します。

  ```
  var/www/html/index.html
  ```

- Apache ウェブサーバーのドキュメントルートの .html ファイルのみをキャッシュするには、次のように指定します。

  ```
  var/www/html/*.html
  ```

- 以下を指定して、 Apache ウェブサーバーのドキュメントルートにある images サブディレクトリの .jpg、.png、および.gif ファイルのみをキャッシュします。

```
var/www/html/images/*.jpg
```

```
var/www/html/images/*.png
```

```
var/www/html/images/*.gif
```

以下を指定して、 Apache ウェブサーバーのドキュメントルートにある images サブディレクトリのすべてのファイルをキャッシュするします。

```
var/www/html/images/
```

キャッシュの詳細設定

詳細設定を使用して、ディストリビューション上のコンテンツのキャッシュのライフスパン、許可されている HTTP メソッド、 HTTP ヘッダー転送、 cookie 転送、およびクエリ文字列転送を指定できます。指定したアドバンスト設定は、ディストリビューションがキャッシュするディレクトリとファイルにのみ適用されます。これには、キャッシュとして指定したディレクトリとファイルの上書きも含まれます。

> **Note**
>
> Lightsail バケットをディストリビューションのオリジンとして選択すると、「ディストリビューション作成」のページでキャッシュのアドバンスト設定を利用することはできません。バケットに保存される静的コンテンツに最適なディストリビューション設定が自動的に適用されます。ただし、ディストリビューションの作成後に、ディストリビューション管理ページでアドバンストキャッシュ設定を変更できます。

次のアドバンスト設定を編集できます。

キャッシュ寿命 (TTL)

コンテンツが更新されたかどうかを確認するためにディストリビューションからオリジンに別のリクエストを送るまで、コンテンツをディストリビューションのキャッシュに保持する期間を制御します。デフォルト値は 1 日です。この期間を短くすると、動的なコンテンツを供給できます。この期間を長くすると、ユーザー側のパフォーマンスは向上します。ファイルがエッジロケーションから直

接返される可能性が高くなるためです。期間を長くすると、ディストリビューションがコンテンツを引き出す頻度が低くなるため、オリジンの負荷も軽減されます。

> **Note**
>
> 指定するキャッシュのライフスパン値は、オリジンが `Cache-Control max-age`、`Cache-Control s-maxage`、`Expires` などの HTTP ヘッダーをコンテンツに追加しないときにのみ適用されます。

許可される HTTP メソッド

ディストリビューションが処理してオリジンに転送する HTTP メソッドをコントロールします。HTTP メソッドは、オリジンで実行されるべきパフォーマンスを示します。例えば、GET メソッドはオリジンからデータを取得し、PUT メソッドは、囲まれたエンティティをオリジンに保存することを要求します。

以下のディストリビューションの HTTP メソッドのオプションのいずれかを選択できます。

- 許可された GET、 HEAD、 OPTIONS、 PUT,、PATCH、 POST と DELETE メソッド
- GET、HEAD、OPTIONS メソッドを許可する
- GET と HEAD メソッドを許可する

ディストリビューションは、常に GET および HEAD の応答をキャッシュします。OPTIONS リクエストを許可するように選択した場合、ディストリビューションは OPTIONS の応答もキャッシュします。ディストリビューションは他の HTTP メソッドへのレスポンスをキャッシュしません。詳細については、「HTTP メソッド」を参照してください。

> **Important**
>
> サポートされているすべての HTTP メソッドを許可するようにディストリビューションを構成する場合、オリジンインスタンスにすべてのメソッドを処理させるように設定する必要があります。例えば POST を使用したいので、上記のメソッドを受け入れて転送するようにディストリビューションを構成する場合は、削除すべきでないリソースをビューワーが削除できないようにするために、DELETE リクエストを適切に処理するようオリジンサーバーを構成する必要があります。詳細については、ウェブサイトまたはウェブアプリケーションのドキュメントを検索してください。

HTTP ヘッダーの転送

ディストリビューションが、指定されたヘッダーの値に基づいてコンテンツをキャッシュするか否か、そしてどのヘッダーに基づくのかをコントロールします。HTTP ヘッダーは、クライアントブラウザ、要求されたページ、オリジンなどの情報を保持します。たとえば、Accept-Languageヘッダーはクライアントの言語を送信します (例えば、英語なら en-US) 。これにより、オリジンはクライアントの言語でコンテンツに対応できます (利用可能な場合) 。

ディストリビューションでは、次の HTTP ヘッダーのオプションのいずれかを選択できます。

- ヘッダーを転送しない
- 指定するヘッダーのみを転送する

ヘッダーを転送しないを選択すると、ディストリビューションはヘッダー値に基づいたコンテンツのキャッシュを行いません。選択したオプションにかかわらず、ディストリビューションは特定のヘッダーをオリジンに転送し、転送したヘッダーに基づいて特定のアクションを実行します。ディストリビューションがヘッダーの転送を処理する方法の詳細については、「HTTP リクエストヘッダーとディストリビューション動作」を参照してください。

Cookie の転送

ディストリビューションがオリジンに Cookie を転送するかどうか、および転送する場合はどれを転送するかをコントロールします。Cookieには、オリジンの ウェブページでの訪問者の行動に関する情報や、訪問者が提供した名前や関心事などの情報など、オリジンに送信される小さなデータが含まれます。

ディストリビューションでは、以下の Cookie 転送オプションのいずれかを選択できます。

- cookie を転送しない
- すべての Cookie を転送する
- 指定した Cookie を転送する

すべての cookie を転送するを選択した場合、ディストリビューションは、アプリケーションで使用されている Cookie の数に関係なく、すべての Cookie を転送します。指定した Cookie を転送するを選択した場合、ディストリビューションに転送して欲しい cookies 名を表示されるテキストボックスに入力します。以下のワイルドカード文字を使用して Cookie 名を指定することができます。

- * は、Cookie 名に含まれる 0 個以上の文字と一致します。

- ? は、Cookie 名に含まれる 1 文字と一致します。

例えば、オブジェクトに対するビューワーリクエストに `userid_`*`member-number`*Cookie 名が含まれているとします。各ユーザーに割り当てられた一意の値 `member-number` (`userid_123`,`userid_124`,`userid_125` など) 。ディストリビューションが、各メンバーについて個別バージョンのコンテンツでキャッシュするものとします。これは Cookie をオリジンに転送することで実行できますが、ビューワーのリクエストには、 ディストリビューションにキャッシュして欲しくない cookies が含まれます。これに代わる方法として、Cookie 名に以下の値を指定できます。その場合、ディストリビューションは `userid_` から始まるすべての Cookie をオリジン `userid_*` に転送します。

クエリ文字列の転送

ディストリビューションがオリジンにクエリ文字列を転送するかどうか、および転送する場合にどれを転送するかをコントロールします。クエリ文字列は、指定されたパラメータに値を割り当てる URL の一部です。例えば、`https://example.com/over/there?name=ferret` URL は `name=ferret` クエリ文字列を含みます。サーバーは、そのようなページのリクエストを受信すると、プログラムを実行し、`name=ferret` クエリ文字列を変更せずにプログラムに渡します。疑問符はセパレータとして使用され、クエリ文字列の一部ではありません。

ディストリビューションがクエリ文字列を転送しないようにするか、指定したクエリ文字列のみを転送するかを選択できます。オリジンがクエリ文字列パラメータの値に関係なくコンテンツの同じバージョンを返す場合、クエリ文字列を転送しないように選択します。これにより、 ディストリビューションがキャッシュからリクエストを処理できる可能性が高くなり、パフォーマンスが向上し、オリジンの負荷が軽減されます。オリジンサーバーが 1 つ以上のクエリ文字列パラメータに基づいてコンテンツの異なるバージョンを返す場合、選択したクエリ文字列のみを転送します。

ディストリビューションプラン

ディストリビューションプランは、毎月のデータ転送クォータとディストリビューションのコストを指定します。プランの月次データ転送クォータよりも多くのデータが配信される場合、超過分が課金されます。詳細については、Lightsail 料金表ページ を参照してください。

超過料金が発生しないようにするには、クォータを超える前に、現在のディストリビューションのプランを毎月のデータ転送量が多い別のプランに変更します。ディストリビューションのプランは、AWS 請求サイクルごとに 1 回だけ変更できます。作成後にディストリビューションプランを変更する方法の詳細については、「ディストリビューションのプランを変更する」を参照してください。

ディストリビューションを作成する

ディストリビューションを作成する手順は以下のとおりです。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. [ディストリビューションの作成] を選択します。
4. ページの [オリジンを選択する] セクションで、オリジンリソースが作成された AWS リージョンを選択します。

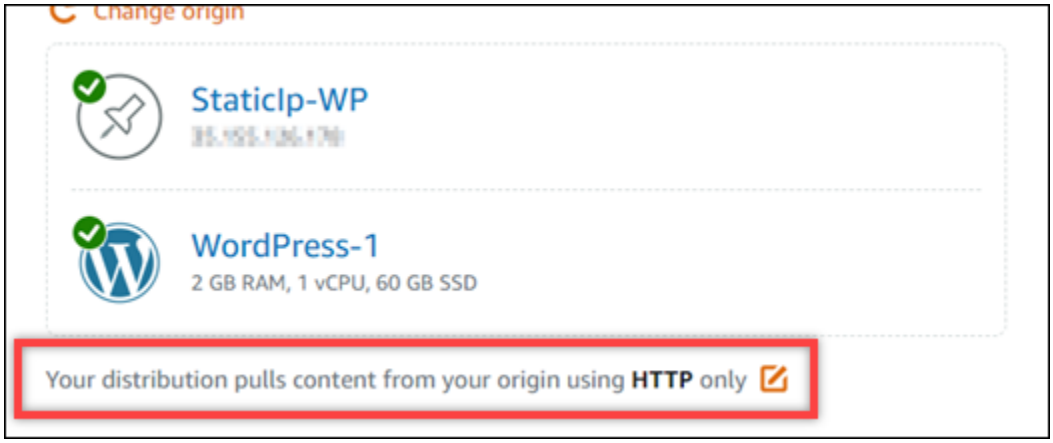
   ディストリビューションはグローバルリソースです。任意の でオリジンを参照し AWS リージョン、そのコンテンツをグローバルに配信できます。
5. オリジンの選択。オリジンは、Lightsail インスタンス、コンテナサービス、バケット、またはロードバランサー (1 つまたは複数のインスタンスがアタッチされている) のいずれかである可能性があります。詳細については、オリジンリソースを参照してください。

   > **Important**
   >
   > ディストリビューションのオリジンとして Lightsail コンテナサービスを選択すると、Lightsail はディストリビューションのデフォルトドメイン名をカスタムドメインとしてコンテナサービスに自動的に追加します。これにより、ディストリビューションとコンテナサービスの間でトラフィックをルーティングできます。ただし、場合によっては、ディストリビューションのデフォルトドメイン名をコンテナサービスに手動で追加する必要があります。詳細については、「ディストリビューションのデフォルトドメインをコンテナサービスに追加する」を参照してください。
6. （オプション）オリジンプロトコルポリシーを変更するには、ディストリビューションが使用する現在のオリジンプロトコルポリシーの横に表示される鉛筆アイコンを選択します。詳細については、オリジンプロトコルポリシーを参照してください。

   このオプションは、オリジンの選択セクションにあり、選択したディストリビューションのオリジンリソース下にあります。

   > **Note**
   >
   > Lightsail バケットをディストリビューションのオリジンとして選択すると、[Origin protocol policy] (オリジンプロトコルポリシー) はデフォルトで [HTTPS only] (HTTPS の

み) に設定されます。バケットがディストリビューションのオリジンである場合、オリジンプロトコルポリシーを変更することはできません。



7. ディストリビューションのキャッシュ動作 (キャッシングプリセットとも呼ばれます) を選択します。詳細については、「キャッシュ動作とキャッシングプリセット」を参照してください。

> **Note**
>
> Lightsail バケットをディストリビューションのオリジンとして選択した場合、キャッシングプリセットオプションは利用できません。バケットに保存される静的コンテンツに最適なディストリビューション設定が自動的に適用されます。

8. (オプション) [すべての設定を表示] を選択して、ディストリビューションの追加のキャッシュ動作設定を表示させます。

> **Note**
>
> Lightsail バケットをディストリビューションのオリジンとして選択すると、キャッシュ動作の設定は利用できません。バケットに保存される静的コンテンツに最適なディストリビューション設定が自動的に適用されます。

9. (オプション) ディストリビューションのデフォルトの動作を選択します。詳細については、デフォルト動作を参照してください。

> **Note**
>
> Lightsail バケットをディストリビューションのオリジンとして選択した場合、デフォルトの動作オプションは利用できません。バケットに保存される静的コンテンツに最適なディストリビューション設定が自動的に適用されます。

10. (オプション) [パスの追加] を選択して、ディストリビューションのキャッシュ動作を上書きするディレクトリとファイルを追加します。詳細については、「ディレクトリとファイルの上書き」を参照してください。

> **Note**
>
> Lightsail バケットをディストリビューションのオリジンとして選択すると、 ディレクトリとファイルの上書きオプションは利用できません バケットに保存される静的コンテンツに最適なディストリビューション設定が自動的に適用されます。

11. （オプション）編集するディストリビューションの横に表示されるアドバンスト設定用の鉛筆アイコンを選択します。詳細については、「アドバンストキャッシュ動作設定」を参照してください。

> **Note**
>
> Lightsail バケットをディストリビューションをオリジンとして選択すると、キャッシュのアドバンスト設定は、「ディストリビューションを作成する」ページで利用できません。バケットに保存される静的コンテンツに最適なディストリビューション設定が自動的に適用されます。ただし、ディストリビューションの作成後に、ディストリビューション管理ページでアドバンストキャッシュ設定を変更できます。

12. ディストリビューションプランを選択します。詳細については、「ディストリビューションプラン」を参照してください。
13. ディストリビューションの名前を入力します。

    リソース名:

    - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
    - 2〜255 文字を使用する必要があります。
    - 先頭と末尾は英数字または数字を使用する必要があります。

- 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

14. ディストリビューションのコストを確認します。
15. [ディストリビューションの作成] を選択します。

    しばらくすると、ディストリビューションが作成されます。

次の手順

ディストリビューションを起動して実行したら、次の手順を実行することをお勧めします。

1. ディストリビューションのオリジンが WordPress インスタンスである場合は、インスタンスの WordPress 設定ファイルを編集して、WordPress ウェブサイトをディストリビューションと連携させる必要があります。詳細については、「ディストリビューションと動作するように WordPress インスタンスを設定する」を参照してください。
2. (オプション) Lightsail コンソールでドメインの DNS を管理するためにLightsail DNS ゾーンを作成します。これにより、Lightsail リソースにドメインを簡単にマッピングすることができます。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。または、現在ホストされているドメインの DNS をホストし続けることもできます。
3. ディストリビューションで使用するドメインの Lightsail SSL/TLS 証明書を作成します。Lightsail ディストリビューションには HTTPS が必要なため、ディストリビューションで使用する前に、ドメインの SSL/TLS 証明書を要求する必要があります。詳細については、「ディストリビューションの SSL/TLS 証明書を作成する」を参照してください。
4. ディストリビューションでカスタムドメインを有効にして、ディストリビューションでドメインを使用できるようにします。カスタムドメインを有効にするには、ドメイン用に作成した Lightsail SSL/TLS 証明書を指定することが必要です。これにより、ドメインがディストリビューションに追加され、HTTPS が有効になります。詳細については、「ディストリビューション用のカスタムドメインを有効にする」を参照してください。
5. ドメインの DNS にエイリアスレコードを追加して、ドメインのトラフィックをディストリビューションにルーティングします。エイリアスレコードを追加したら、ドメインにアクセスしたユーザーはディストリビューションを通じてルーティングされます。詳細については、「ドメインをディストリビューションにポイントする」を参照してください。
6. ディストリビューションがコンテンツをキャッシュしていることをテストします。詳細については、「ディストリビューションをテストする」を参照してください。

Lightsail ディストリビューションを削除する

使用していない Amazon Lightsail ディストリビューションはいつでも削除できます。

ディストリビューションを削除する

ディストリビューションを削除するためには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. 削除するディストリビューションの名前を選択します。
4. ディストリビューション管理ページで [Delete] (削除) タブを選択します。
5. ディストリビューションを削除するためには、[ディストリビューションの削除] を選択します。
6. [はい、削除します] を選択して削除を確定します。

Lightsail ディストリビューションのキャッシュを設定する

キャッシュ動作は Amazon Lightsail ディストリビューションによってキャッシュされているもの、されていないもののオリジンを設定可能にします。例えば、オリジンから個々のディレクトリ、ファイル、またはファイルタイプをキャッシュするように指定できます。オリジンに転送される HTML メソッドとヘッダーを指定することもできます。このガイドでは、ディストリビューションのキャッシュ動作を変更する方法について説明します。ディストリビューションの詳細については、「コンテンツ配信ネットワークディストリビューション」を参照してください。

目次

- キャッシュプリセット
- WordPress の最適キャッシュプリセット
- デフォルトの動作
- ディレクトリとファイルの上書き
- キャッシュの詳細設定
- ディストリビューションのキャッシュ動作を変更する

キャッシュプリセット

キャッシュプリセットは、オリジンでホストするコンテンツの種類に応じて、ディストリビューションの設定を自動的に設定します。例えば、静的コンテンツに最適を選択すると、プリセットは、静的ウェブサイトで最適に動作する設定にディストリビューションを自動的に設定します。ウェブサイトが WordPress インスタンスでホストされている場合は、 WordPress に最適プリセットを使用して、WordPress ウェブサイトで最適に動作するようにディストリビューションが自動的に設定されます。

ディストリビューションでは、以下のいずれかのキャッシュプリセットを選択できます。

- 静的コンテンツに最適 - このプリセットは、ディストリビューションをすべてをキャッシュするように設定されます。このプリセットは、オリジンで静的コンテンツ (静的 HTML ページなど) をホストする場合や、ウェブサイトにアクセスするユーザーごとに変更されないコンテンツをホストする場合に最適です。このプリセットを選択すると、ディストリビューション上のすべてのコンテンツがキャッシュされます。
- 動的コンテンツに最適 - このプリセットは、ディストリビューションをディストリビューションを作成するページのセクションにあるディレクトリとファイルの上書きのキャッシュで指定されていないもの以外をキャッシュしないように設定されます。詳細については、ディレクトリとファイルの上書きを後ほど参照してください。このプリセットは、オリジンで動的なコンテンツや、ウェブサイトやウェブアプリケーションにアクセスするユーザーごとに変化するコンテンツをホストする場合に最適です。
- WordPress に最適 - このプリセットは、ディストリビューションを`wp-includes/`と`wp-content/` WordPress インスタンスのディレクトリーにあるファイル以外をキャッシュしないように設定されます。このプリセットは、オリジンのインスタンスが WordPress Certified by Bitnami and Automatic ブループリントの場合に最適です (マルチサイトブループリントを除く) 。このプリセットの詳細については、 WordPress の最適キャッシュプリセット

> **Note**
>
> カスタム設定プリセットは選択できません。プリセットはプリセットを選択後、自動で選択されますが、ディストリビューションの設定を手動で変更します。

Lightsail コンソールのみでキャッシュのプリセットを指定できます。Lightsail API、 AWS CLI、および SDKs を使用して指定することはできません。

WordPress の最適キャッシュプリセット

選択したインスタンスが、 WordPress Certified by Bitnami and Automattic 設計図をディストリビューションのオリジンとして使用している場合、Lightsail はディストリビューションのキャッシュプリセットに [WordPress に最適] を適用するかを確認するプロンプトが表示されます。プリセットを適用すると、ディストリビューションは WordPress のウェブサイトに最適した動作に自動的に設定されます。他に適用しなければいけないディストリビューション設定はありません。WordPress に最適は、`wp-includes/`内のファイルおよび`wp-content/` WordPress ウェブサイトのディレクトリ以外をキャッシュしないようにプリセットされます。また、毎日キャッシュをクリアするようにディストリビューションを設定し (キャッシュ寿命は 1 日) 、すべての HTTP メソッドを許可し、`Host` ヘッダーのみを転送し、Cookieを転送せず、すべてのクエリ文字列を転送します。

> **Important**
>
> WordPress ウェブサイトをディストリビューションで使用できるようにするには、インスタンスの WordPress 設定ファイルを編集する必要があります。詳細については、「ディストリビューションと動作するように WordPress インスタンスを設定する」を参照してください。

デフォルトの 動作

デフォルトの動作は、ディストリビューションがコンテンツキャッシュをどのように処理するかを指定します。ディストリビューションのデフォルトの動作は、選択したキャッシュプリセットによって自動的に決定されます。別のデフォルト動作を選択した場合、キャッシュプリセットは自動的にカスタム設定にされます。

ディストリビューションは、以下のデフォルトの動作のいずれかから選択できます。

- すべてをキャッシュする - この動作は、ウェブサイト全てを静的コンテンツとしてキャッシュ、対応するようにディストリビューションを設定します。このオプションは、閲覧者によって変更されないコンテンツをオリジンがホストしている場合、または ウェブサイトが cookie、ヘッダー、またはクエリ文字列を使用してコンテンツをパーソナライズしない場合に最適です。
- 何もキャッシュしない - この動作は、指定したオリジンファイルとフォルダーパスのみをキャッシュするようにディストリビューションを設定します。このオプションは、ウェブサイトやウェブアプリケーションが cookie、ヘッダー、クエリ文字列を使用して、個々のユーザー向けにコンテンツをパーソナライズする場合に最適です。このオプションを選択すると、キャッシュするには、ディレクトリとファイルパスの上書きを指定する必要があります。

ディレクトリとファイルの上書き

ディレクトリとファイルの上書きを使用して、選択したデフォルトの動作を上書きしたり、例外を追加することが可能です。例えば、すべてをキャッシュするを選択した場合、上書きを使用して、ディストリビューションがキャッシュしないディレクトリ、ファイル、またはファイルの種類を指定します。代わりに、何もキャッシュしないを選択した場合、上書きを使用して、ディストリビューションがキャッシュするディレクトリ、ファイル、またはファイルの種類を指定します。

ディレクトリとファイルの上書きセクションで、キャッシュするディレクトリまたはファイルへのパスを指定するか、キャッシュしないかを指定できます。アスタリスク記号を使用して、ワイルドカードディレクトリ (`path/to/assets/*`)、ファイルタイプ (`*.html`,`*jpg`,`*js`)を指定する。ディレクトリとファイルのパスでは、大文字と小文字が区別されます。

ディレクトリとファイルの上書きを指定する方法の例をいくつか紹介します。

- 以下を指定して、Lightsail インスタンスで動作している Apache ウェブサーバーのドキュメントルート内のすべてのファイルをキャッシュします。

  ```
  var/www/html/
  ```

- 以下を指定して、 Apache ウェブサーバーのドキュメントルートにあるインデックスページのみをキャッシュします。

  ```
  var/www/html/index.html
  ```

- 以下を指定して、 Apache ウェブサーバーのドキュメントルートにある. html ファイルのみをキャッシュします。

  ```
  var/www/html/*.html
  ```

- 以下を指定して、 Apache ウェブサーバーのドキュメントルートにある images サブディレクトリの .jpg、.png、および.gif ファイルのみをキャッシュします。

  ```
  var/www/html/images/*.jpg
  ```

  ```
  var/www/html/images/*.png
  ```

  ```
  var/www/html/images/*.gif
  ```

以下を指定して、 Apache ウェブサーバーのドキュメントルートにある images サブディレクトリのすべてのファイルをキャッシュするします。

```
var/www/html/images/
```

キャッシュの詳細設定

詳細設定を使用して、ディストリビューション上のコンテンツのキャッシュのライフスパン、許可されている HTTP メソッド、 HTTP ヘッダー転送、 cookie 転送、およびクエリ文字列転送を指定できます。指定したアドバンスト設定は、ディストリビューションがキャッシュするディレクトリとファイルにのみ適用されます。これには、キャッシュとして指定したディレクトリとファイルの上書きも含まれます。

次のアドバンスト設定を編集できます。

キャッシュ寿命 (TTL)

コンテンツが更新されたかどうかを確認するためにディストリビューションからオリジンに別のリクエストを送るまで、コンテンツをディストリビューションのキャッシュに保持する期間を制御します。デフォルト値は 1 日です。この期間を短くすると、動的なコンテンツを供給できます。この期間を長くすると、ユーザー側のパフォーマンスは向上します。ファイルがエッジロケーションから直接返される可能性が高くなるためです。期間を長くすると、ディストリビューションがコンテンツを引き出す頻度が低くなるため、オリジンの負荷も軽減されます。

> **Note**
>
> 指定するキャッシュのライフスパン値は、オリジンが `Cache-Control max-age`、`Cache-Control s-maxage`、`Expires` などの HTTP ヘッダーをコンテンツに追加しないときにのみ適用されます。

許可される HTTP メソッド

ディストリビューションが処理してオリジンに転送する HTTP メソッドをコントロールします。HTTP メソッドは、オリジンで実行されるべきパフォーマンスを示します。例えば、GET メソッドはオリジンからデータを取得し、PUT メソッドは、囲まれたエンティティをオリジンに保存することを要求します。

以下のディストリビューションの HTTP メソッドのオプションのいずれかを選択できます。

- 許可された GET、 HEAD、 OPTIONS、 PUT,、PATCH、 POST と DELETE メソッド
- GET、HEAD、OPTIONS メソッドを許可する
- GET と HEAD メソッドを許可する

ディストリビューションは、常に GET および HEAD の応答をキャッシュします。OPTIONS リクエストを許可するように選択した場合、ディストリビューションは OPTIONS の応答もキャッシュします。ディストリビューションは他の HTTP メソッドの応答をキャッシュしません。

> **Important**
>
> サポートされているすべての HTTP メソッドを許可するようにディストリビューションを設定した場合、オリジンインスタンスがすべてのメソッドを処理できるように設定する必要があります。例えば POST を使用したいので、上記のメソッドを受け入れて転送するようにディストリビューションを構成する場合は、削除すべきでないリソースをビューワーが削除できないようにするために、DELETE リクエストを適切に処理するようオリジンサーバーを構成する必要があります。詳細については、ウェブサイトまたはウェブアプリケーションのドキュメントを検索してください。

HTTP ヘッダーの転送

ディストリビューションが、指定されたヘッダーの値に基づいてコンテンツをキャッシュするか否か、そしてどのヘッダーに基づくのかをコントロールします。HTTP ヘッダーは、クライアントブラウザ、要求されたページ、オリジンなどの情報を保持します。たとえば、Accept-Languageヘッダーはクライアントの言語を送信します (例えば、英語なら en-US) 。これにより、オリジンはクライアントの言語でコンテンツに対応できます (利用可能な場合) 。

ディストリビューションでは、次の HTTP ヘッダーのオプションのいずれかを選択できます。

- ヘッダーを転送しない
- 指定するヘッダーのみを転送する

ヘッダーを転送しないを選択すると、ディストリビューションはヘッダー値に基づいたコンテンツのキャッシュを行いません。選択したオプションにかかわらず、ディストリビューションは特定のヘッダーをオリジンに転送し、転送したヘッダーに基づいて特定のアクションを実行します。

Cookie の転送

ディストリビューションがオリジンに Cookie を転送するかどうか、および転送する場合はどれを転送するかをコントロールします。Cookieには、オリジンの ウェブページでの訪問者の行動に関する情報や、訪問者が提供した名前や関心事などの情報など、オリジンに送信される小さなデータが含まれます。

ディストリビューションでは、以下の Cookie 転送オプションのいずれかを選択できます。

- cookie を転送しない
- すべての Cookie を転送する
- 指定した Cookie を転送する

すべての cookie を転送するを選択した場合、ディストリビューションは、アプリケーションで使用されている Cookie の数に関係なく、すべての Cookie を転送します。指定した Cookie を転送するを選択した場合、ディストリビューションに転送して欲しい cookies 名を表示されるテキストボックスに入力します。以下のワイルドカード文字を使用して Cookie 名を指定することができます。

- * は、Cookie 名に含まれる 0 個以上の文字と一致します。
- ? は、Cookie 名に含まれる 1 文字と一致します。

例えば、オブジェクトに対するビューワーリクエストに `userid_`*`member-number`*Cookie 名が含まれているとします。各ユーザーに割り当てられた一意の値 `member-number` (`userid_123`,`userid_124`,`userid_125` など) 。ディストリビューションが、各メンバーについて個別バージョンのコンテンツでキャッシュするものとします。これは Cookie をオリジンに転送することで実行できますが、ビューワーのリクエストには、 ディストリビューションにキャッシュして欲しくない cookies が含まれます。これに代わる方法として、Cookie 名に以下の値を指定できます。その場合、ディストリビューションは `userid_` から始まるすべての Cookie をオリジン `userid_*` に転送します。

クエリ文字列の転送

ディストリビューションがオリジンにクエリ文字列を転送するかどうか、および転送する場合にどれを転送するかをコントロールします。クエリ文字列は、指定されたパラメータに値を割り当てる URL の一部です。例えば、`https://example.com/over/there?name=ferret` URL は `name=ferret` クエリ文字列を含みます。サーバーは、そのようなページのリクエストを受信すると、プログラムを実行し、`name=ferret` クエリ文字列を変更せずにプログラムに渡します。疑問符はセパレータとして使用され、クエリ文字列の一部ではありません。

ディストリビューションがクエリ文字列を転送しないようにするか、指定したクエリ文字列のみを転送するかを選択できます。オリジンがクエリ文字列パラメータの値に関係なくコンテンツの同じバージョンを返す場合、クエリ文字列を転送しないように選択します。これにより、 ディストリビューションがキャッシュからリクエストを処理できる可能性が高くなり、パフォーマンスが向上し、オリジンの負荷が軽減されます。オリジンサーバーが 1 つ以上のクエリ文字列パラメータに基づいて、異なるバージョンのコンテンツを返す場合、指定したクエリ文字列のみを転送するように選択します。

ディストリビューションのキャッシュ動作を変更する

ディストリビューションのデフォルトのキャッシュ動作を変更するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. デフォルトのキャッシュ動作を変更するディストリビューションの名前を選択します。
4. ディストリビューション管理ページのキャッシュタブを開きます。
5. キャッシュを設定するセクションで、ディストリビューションのキャッシュプリセットを選択します。詳細については、キャッシュプリセットを参照してください。
6. デフォルトのキャッシュ動作を変更するを選択して、ディストリビューションのデフォルト動作を変更します。次に、ディストリビューションのデフォルト動作を選択します。詳細については、デフォルト動作を参照してください。
7. パスの追加を選択して、ディストリビューションのキャッシュ動作にディレクトリとファイルの上書きを追加します。詳細については、ディレクトリとファイルの上書きを参照してください。
8. 編集したいディストリビューションの詳細設定の横に表示される鉛筆アイコンを選択します。詳細については、キャッシュ動作詳細設定を参照してください。

ディストリビューションの設定を保存したら、すべてのエッジロケーションに伝達し始めます。エッジロケーションで設定が更新されるまでは、 以前の設定に基づいて、そののロケーションからコンテンツを引き続き供給します。エッジロケーションで設定がが更新されると、 新しい設定に基づいて、そのロケーションからコンテンツを直ちに供給し始めます。

変更は、すべてのエッジロケーションにすぐに伝達されるわけではありません。伝達が完了すると、ディストリビューションのステータスが進行中からデプロイ済みに変わります。ディストリビューションがが変更を伝達している間、特定のエッジロケーションでコンテンツが以前の設定または新しい設定のどちらに基づいて供給されるかを判別することはできません。

トピック

- Lightsail ディストリビューションのキャッシュをリセット

Lightsail ディストリビューションのキャッシュをリセット

キャッシュの寿命 (有効期限 [TTL]) 設定は、コンテンツが Amazon Lightsail ディストリビューションに残る期間をコントロールします。キャッシュの有効期限より前にキャッシュをクリアにする必要がある場合は、ディストリビューションのキャッシュを手動でリセットすることもできます。キャッシュをクリアにすると、次回ユーザーがコンテンツをリクエストした際、ディストリビューションはコンテンツの最新バージョンをオリジンから取り出し、キャッシュします。このガイドでは、ディストリビューションでキャッシュを手動でリセットする方法を説明します。ディストリビューションの詳細については、「コンテンツ配信ネットワークディストリビューション」を参照してください。

ディストリビューションのキャッシュをリセット

ディストリビューションのキャッシュをリセットするには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. キャッシュをリセットしたいディストリビューションの名前を選択します。
4. ディストリビューションの管理ページで [Cache] タブを選択します。
5. ページの [キャッシュのリセット] セクションまでスクロールして、[キャッシュのリセット] を選択します。
6. 確認プロンプトで [はい、リセットします] を選択してディストリビューションのキャッシュのリセットを確定します。または [いいえ、キャンセル] を選択して、ディストリビューションのキャッシュをリセットしないようにします。

Lightsail ディストリビューションのコンテンツオリジンを変更する

このガイドでは、Amazon Lightsail ディストリビューションを作成後、オリジンの変更方法を紹介します。オリジンとは、ディストリビューションのコンテンツの最終的なソースです。ディストリビューションを作成するときに、Lightsail インスタンス、Lightsail バケット、Lightsail ウェブサイトまたはウェブアプリケーションのコンテンツをホストするロードバランサー (1 つ以上のインスタンスが添付されている) を選択します。詳細については、「コンテンツ配信ネットワークディストリビューション」を参照してください。

ディストリビューションを作成後、いつでもオリジンを変更できます。オリジンを変更する時、直ちにディストリビューションはエッジロケーションに変更を適用します。ディストリビューションがエッジロケーションの新しいオリジンに更新されるまで、古いオリジンにリクエストが転送されます。

オリジンを変更しても、 ディストリビューションは新しいオリジンからのコンテンツでエッジキャッシュを生成し直す必要はありません。ウェブサイトまたはウェブアプリケーション内でユーザーのリクエストが変更されていない限り、コンテンツのキャッシュのライフスパンが切れるまで、ディストリビューションは、エッジキャッシュに既存するコンテンツを供給します。

オリジンプロトコルポリシー

オリジンプロトコルポリシーは、オリジンからコンテンツを引き出す時にディストリビューションが使用するプロトコルポリシーです。ディストリビューションのオリジンを選択した後、オリジンからコンテンツを引き出す際に、Hypertext Transfer Protocol (HTTP) か、Hypertext Transfer Protocol Secure (HTTPS) どちらを使用すべきか決めます。オリジンが HTTPS 用に設定されていない場合は、HTTP を使用する必要があります。

ディストリビューションに対して、次のいずれかのオリジンプロトコルポリシーを選択できます。

- HTTP Only - オリジンへのアクセスに HTTP のみを使用します。これはデフォルトの設定です。
- HTTPS Only - オリジンへのアクセスに HTTPS のみを使用します。

オリジンプロトコルポリシーを編集する手順は、このガイドのディストリビューションのオリジンを変更するセクションを参照してください。

ディストリビューションのオリジンを変更する

ディストリビューションのオリジンの変更を行うには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. オリジンを変更したいディストリビューションの名前を選択します。
4. ディストリビューション管理ページの詳細タブを選択して、オリジンの選択までスクロールします。

   オリジンの選択セクションには、ディストリビューションの現在のオリジンが表示されます。

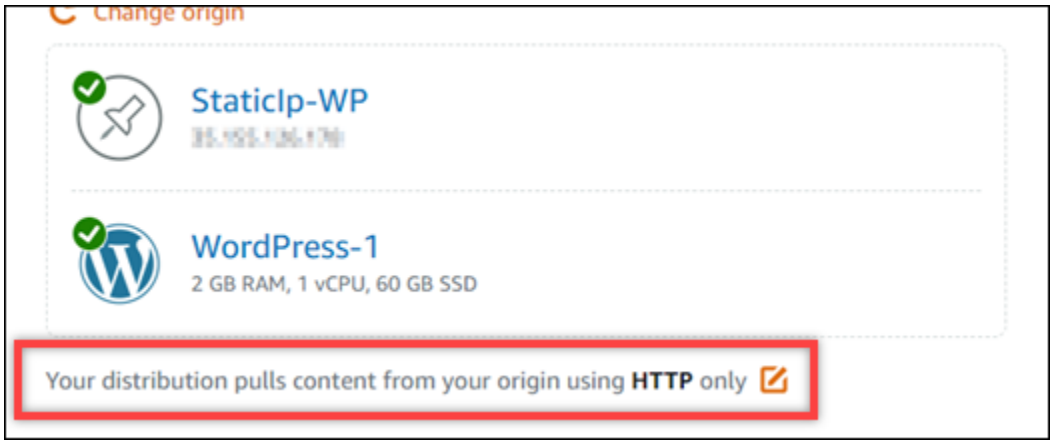
5. オリジンを変更を選択します。
6. オリジンのリソースが作成された AWS リージョンを選択します。

   ディストリビューションはグローバルリソースです。どの AWS リージョンでもオリジンをリファレンスでき、グローバルにそのコンテンツを配信することができます。
7. オリジンの選択。オリジンは、インスタンス、バケット、またはロードバランサー (1 つ以上のインスタンスが添付されている) にすることが可能です。
8. 保存を選択して、新しいオリジンでディストリビューションを更新します。

   ディストリビューションのオリジンを選択したら、オリジンからコンテンツを引き出す際に、Hypertext Transfer Protocol (HTTP) かHypertext Transfer Protocol Secure (HTTPS) のどちらを使用するかを決める必要があります。
9. (オプション) オリジンプロトコルポリシーを変更するには、ディストリビューションが使用する現在のオリジンプロトコルポリシーの横に表示される鉛筆アイコンを選択します。詳細については、オリジンプロトコルポリシーを参照してください。

   このオプションは、オリジンの選択セクションにあり、選択したディストリビューションのオリジンリソース下にあります。

> **Note**
>
> Lightsail バケットをディストリビューションのオリジンとして選択すると、[Origin protocol policy] (オリジンプロトコルポリシー) はデフォルトで [HTTPS only] (HTTPS のみ) に設定されます。バケットがディストリビューションのオリジンである場合、オリジンプロトコルポリシーを変更することはできません。



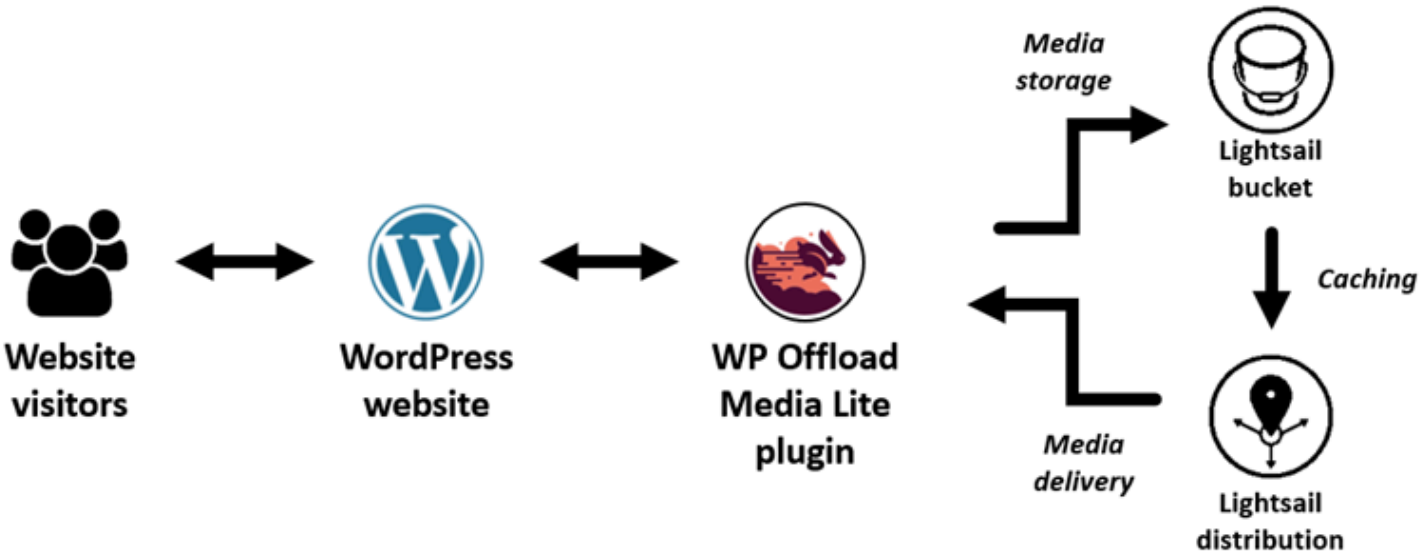
10. HTTP Only または HTTPS Only を選択して、保存を選択してオリジンプロトコルポリシーを保存します。

ディストリビューション設定に変更を保存すると、 変更をすべてのエッジロケーションに伝達し始めます。エッジロケーションで設定が更新されるまでは、 以前の設定に基づいて、そののロケーションからコンテンツを引き続き供給します。エッジロケーションで設定がが更新されると、 新しい設定に基づいて、そのロケーションからコンテンツを直ちに供給し始めます。

変更は、すべてのエッジロケーションにすぐに伝達されるわけではありません。伝達が完了すると、ディストリビューションのステータスが進行中からデプロイ済みに変わります。ディストリビューションがが変更を伝達している間、特定のエッジロケーションでコンテンツが以前の設定または新しい設定のどちらに基づいて供給されるかを判別することはできません。

Lightsail バケットと CDN ディストリビューションを使用してメディアファイルを効率的に提供する

このチュートリアルでは、Amazon Lightsail バケットを Lightsail コンテンツ配信ネットワーク(CDN)配信のオリジンとして設定するために必要な手順を説明します。また、バケットにメディア (イメージや動画ファイルなど) をアップロードして保存し、ディストリビューションからメディアを配信するために WordPress ウェブサイトに設定する方法も説明します。その方法の一例として、「WP Offload Media Lite プラグイン」があります。次の図にその概念を示します。



ウェブサイトメディアを Lightsail バケットに保存すると、これらのファイルを保存して提供する必要がなくなり、インスタンスの負荷が軽減されます。Lightsail ディストリビューションからメディア

をキャッシュし、提供すると、ウェブサイトの訪問者へのファイルの配信が高速化され、ウェブサイト全体のパフォーマンスが向上します。ディストリビューションの詳細については、「コンテンツ配信ネットワークディストリビューション」を参照してください。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

目次

- ステップ 1: 前提条件を満たす
- ステップ 2: バケットのアクセス許可を変更する
- ステップ 3: オリジンとしてのバケットを持つディストリビューションを作成する
- ステップ 4: ディストリビューションのカスタムサブドメインを有効にする
- ステップ 5: WordPress ウェブサイトに WP Offload Media プラグインをインストールする
- ステップ 6: WordPress ウェブサイトと Lightsail バケットとディストリビューションの接続をテストする

ステップ 1: 前提条件を満たす

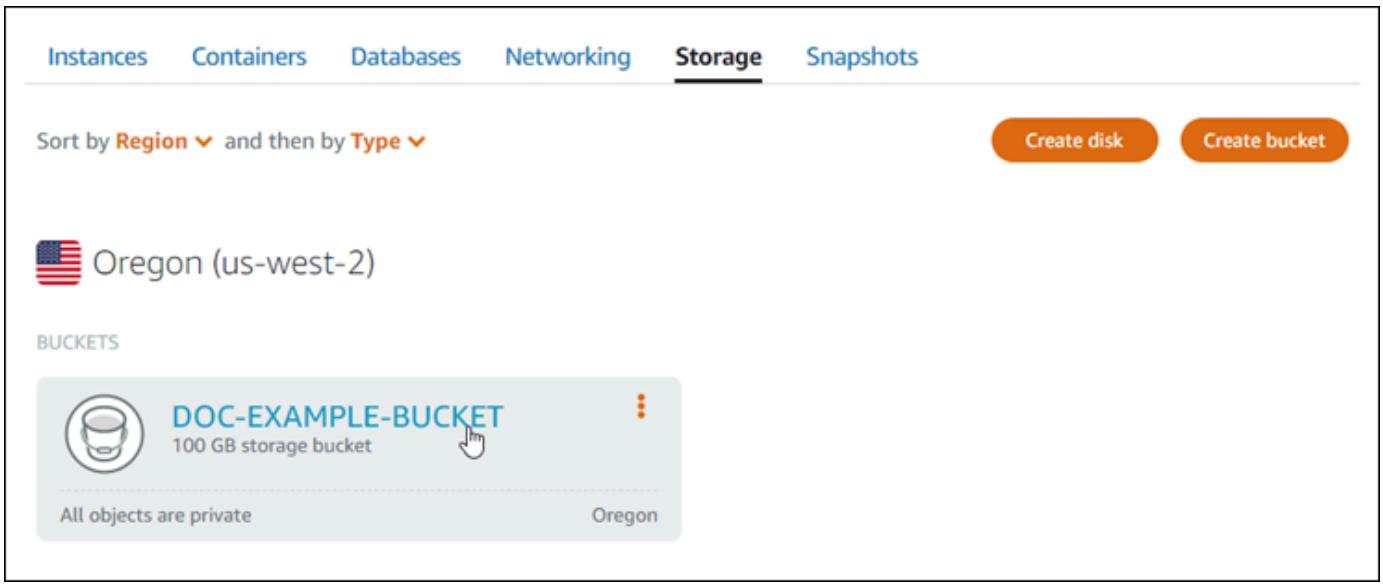
以下の前提条件を完了します (まだの場合)。

- Lightsail で WordPress のインスタンスを作成し設定したあと、管理用ダッシュボードにサインインするためのパスワードを取得します。詳細については、「チュートリアル: Amazon Lightsail で WordPress インスタンスを起動して設定する」を参照してください。
- Lightsail オブジェクトストレージサービスでバケットを作成します。詳細については、「Lightsail のバケットの作成」を参照してください。

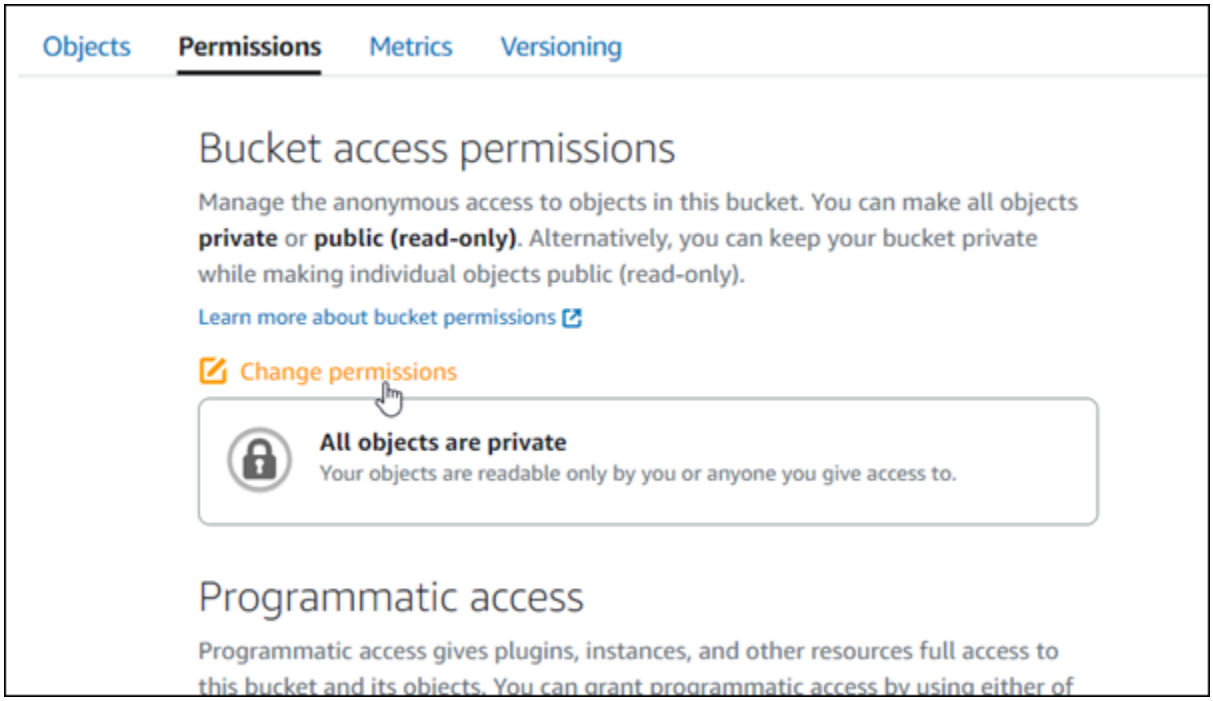
ステップ 2: バケットのアクセス許可を変更する

次の手順を実行して、WordPress インスタンスと WP Offload Media Lite プラグインにバケットへのアクセスを許可します。バケットのアクセス許可は個々のオブジェクトを公開 (読み取り専用) に設定する必要があります。また、WordPress インスタンスをバケットにアタッチする必要があります。バケット許可の詳細については、「バケットのアクセス許可」を参照してください。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. WordPress ウェブサイトで使用するバケットの名前を選択します。

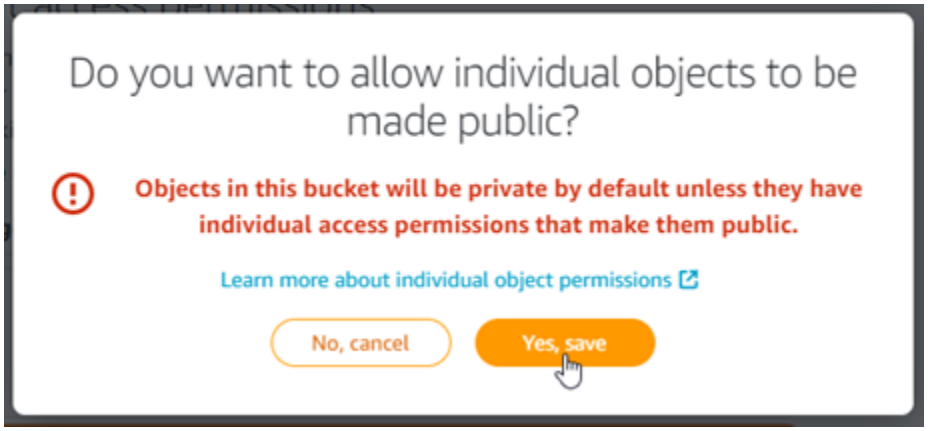


4. バケット管理ページで [Permissions] (許可) タブを選択します。
5. ページの「バケットのアクセス許可」セクションで [Change permissions](許可の変更) を選択します。



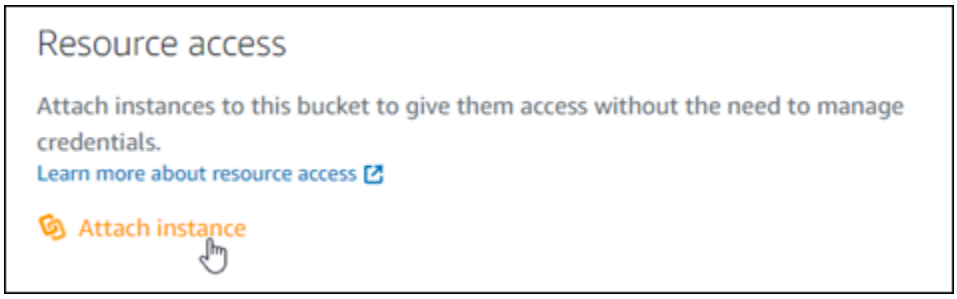
6. 個々のオブジェクトを選択して公開し、読み取り専用にすることができます。

7. [保存] を選択します。
8. 表示される確認プロンプトで、[はい、選択]を選択します。

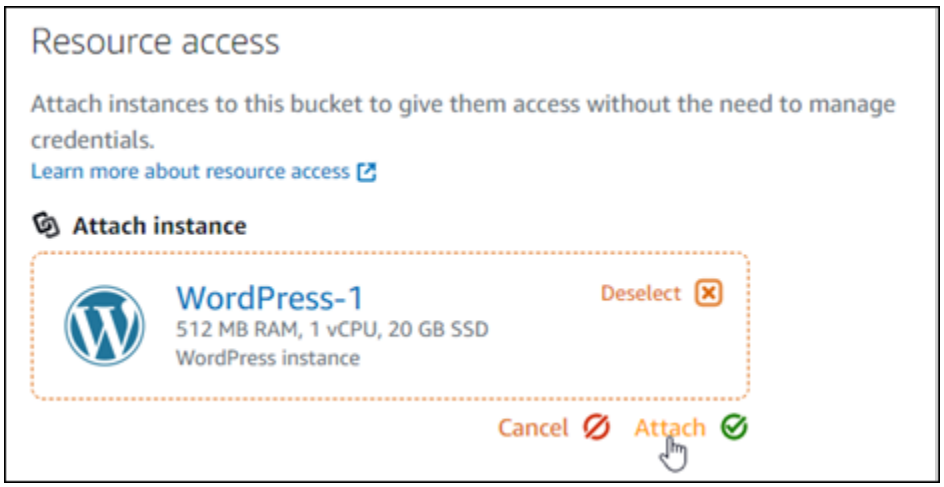


しばらくすると、バケットは個々のオブジェクトにアクセスを許可するように設定されます。これにより、Offload Media Lite プラグインを使用して、WordPress ウェブサイトからバケットにアップロードされたオブジェクトをユーザーが読み取れるようになります。

9. ページの [リソースアクセス] セクションまでスクロールし、[Attach instance] (インスタンスの添付) を選択します。



10. 表示されるドロップダウンで WordPress インスタンスの名前を選択し、[添付]を選択します。



しばらくすると、WordPress インスタンスがバケットに添付されます。これにより、WordPress インスタンスにアクセスして、バケットとそのオブジェクトを管理するアクセス権が与えられます。

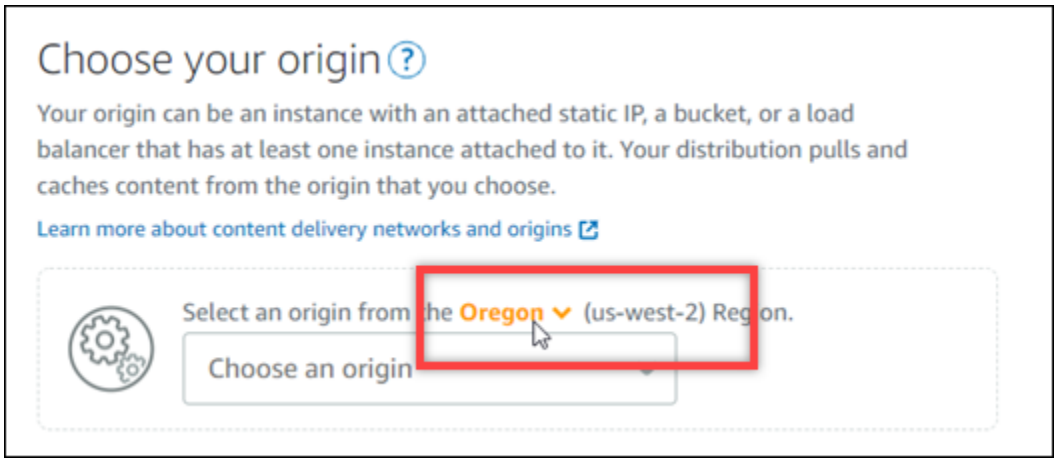
ステップ 3: オリジンとしてのバケットを持つディストリビューションを作成する

以下の手順に従って、Lightsail ディストリビューションを作成し、選択し、オリジンとしてLightsail バケットを選択します。

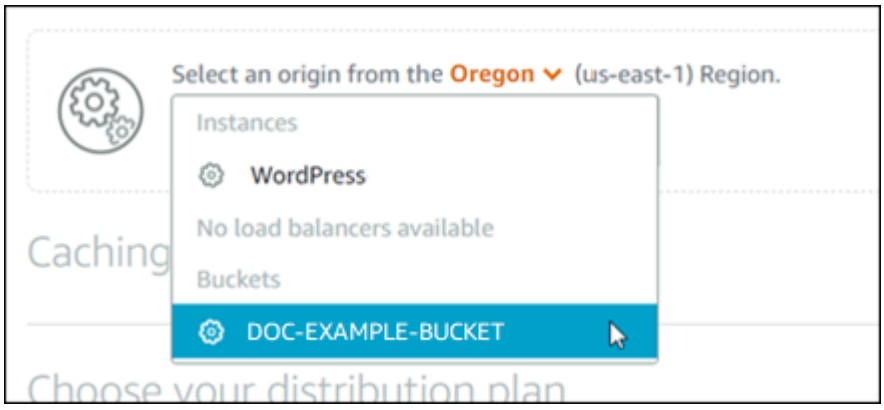
1. Lightsail コンソールの上部のナビゲーション メニューで [ホーム] を選択します。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. [ディストリビューションの作成] を選択します。

4. このページの [オリジンの選択] セクションで、バケットを作成した AWS リージョン を選択します。

   ディストリビューションはグローバルリソースです。任意の のバケットを参照し AWS リージョン、そのコンテンツをグローバルに配信できます。

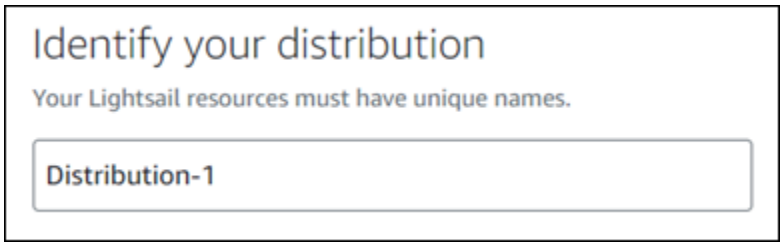


5. バケットをオリジンとして選択します。



> **Note**
>
> バケットのアクセス許可は個々のオブジェクトを公開 (読み取り専用) に設定する必要があります。公開として設定されている個々のオブジェクトだけがキャッシュされ、ディストリビューションで配信されます。ディストリビューションのオリジンとしてバケットを選択すると、オリジンプロトコルポリシー、キャッシュ動作、デフォルトの動作、ディレクトリとファイルの上書きを指定するオプションが使用できなくなり、編集もできなくなります。デフォルトでは、オリジンプロトコルポリシーはバケットに対して [HTTPS のみ] に設定され、キャッシュ動作のデフォルトは [すべてをキャッシュする] に設定されます。ディストリビューションのアドバンストキャッシュ設定は、ディストリビューションの作成後に変更できます。

6. ディストリビューションプランを選択します。
7. ディストリビューションの名前を入力します。



ディストリビューション名 :

- Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
- 2 ~ 255 文字を使用する必要があります。
- 先頭と末尾は英数字または数字を使用する必要があります。
- 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

8. [ディストリビューションの作成] を選択します。

しばらくすると、ディストリビューションが作成されます。新しいディストリビューションが [Enabled] (有効) になると、バケット内のオブジェクトを提供してキャッシュする準備が整った状態です。

ステップ 4: ディストリビューションのカスタムサブドメインを有効にする

ディストリビューションを作成すると、123abc.cloudfront.net と同様のデフォルトドメインで構成されます。WP Offload Media Lite プラグインを設定するときに、そのデフォルトドメインをメディアファイルのソースとして指定することができます。ただし、ディストリビューションのカスタムドメインを有効にすることを強くお勧めします。ディストリビューション用に有効にするカスタムドメインは、WordPress ウェブ サイトで使用しているドメインのサブドメインである必要があります。たとえば、使用しているmycustomdomain.comを WordPress のウェブサイトに追加すると、ディストリビューションでカスタムドメイン media.mycustomdomain.com を使用することを選択できます。WordPress ウェブサイトとディストリビューションの間で同じドメインとサブドメインの組み合わせを使用すると、ウェブサイトの検索エンジン最適化スコアを向上させるのに役立ちます。

ディストリビューション用のカスタムドメインを設定するには、以下のステップを実行します。

1. ディストリビューションで使用するために、Lightsail ドメインの Lightsail SSL/TLS証明書を作成してください。ディストリビューションには HTTPS が必要なため、ディストリビューションで使用する前に、ドメインの SSL/TLS 証明書をリクエストする必要があります。詳細については、「ディストリビューションの SSL/TLS 証明書を作成する」を参照してください。
2. ディストリビューションでカスタムドメインを有効にして、ディストリビューションでドメインを使用できるようにします。カスタムドメインを有効にするには、ドメイン用に作成した Lightsail SSL/TLS 証明書を指定することが必要です。これにより、ドメインがディストリビューションに追加され、HTTPS が有効になります。詳細については、「ディストリビューション用のカスタムドメインを有効にする」を参照してください。
3. ドメインの DNS ゾーンにエイリアス レコードを追加 エイリアスレコードを追加すると、ドメインにアクセスするユーザーはディストリビューションを通じてルーティングされます。詳細については、「ドメインをディストリビューションにポイントする」を参照してください。

ステップ 5: WordPress ウェブサイトに WP Offload Media プラグインをインストールします。

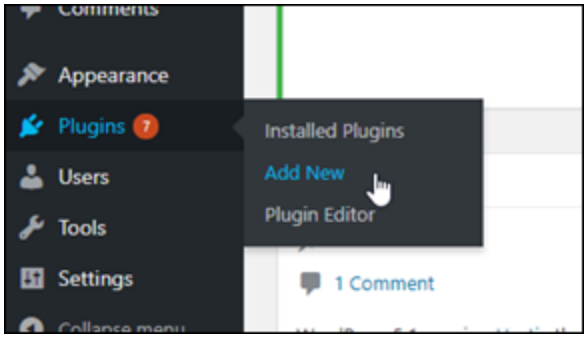
次のステップに従って、WordPress ウェブサイトに WP Offload Media Lite プラグインをインストールします。このプラグインは、WordPressのメディアアップローダーを介して追加されたイメージ、ビデオ、ドキュメント、およびその他のメディアを Lightsail バケットに自動的にコピーします。また、バケットから Lightsail ディストリビューションを通じてメディアを提供するように設定するこ

ともできます。詳細については、WordPress ウェブサイトの「WP Offload Media Lite」を参照してください。

1. 管理者として WordPress ウェブサイトのダッシュボードにサインインします。

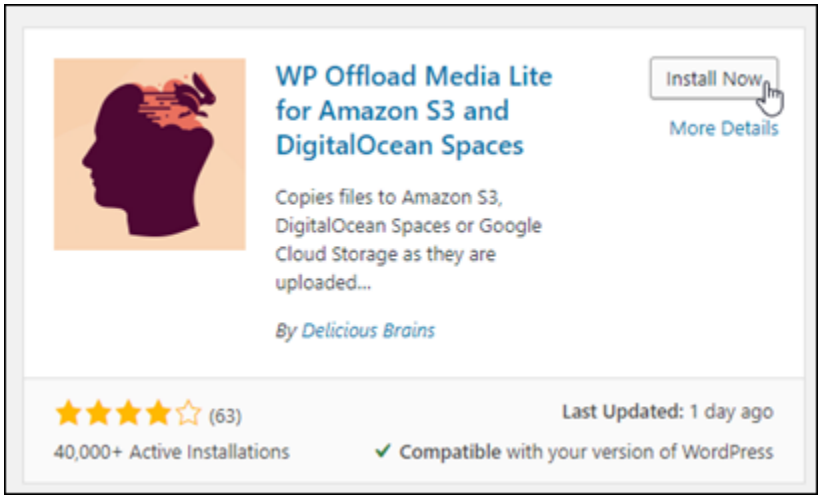
   詳細については、「Amazon Lightsailの Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

2. 左側のナビゲーションメニューのプラグインを一時停止し、[Add New] (新規追加) を選択します。

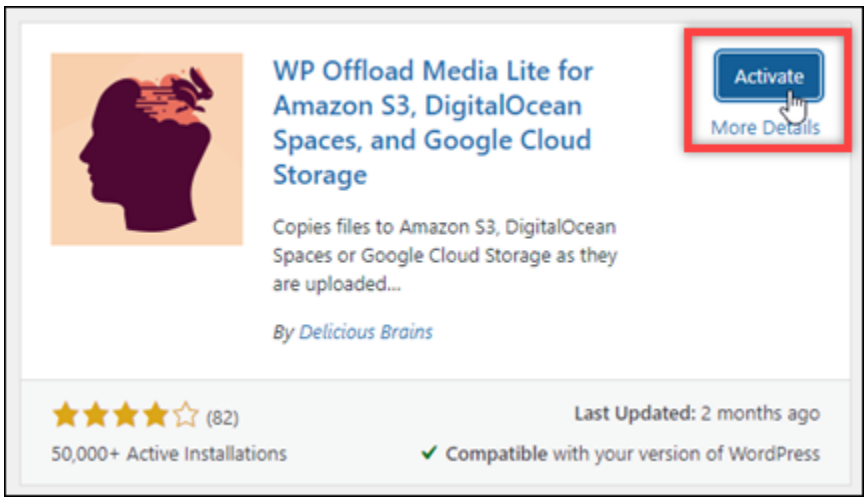


3. [WP Offload Media Lite] を検索します。

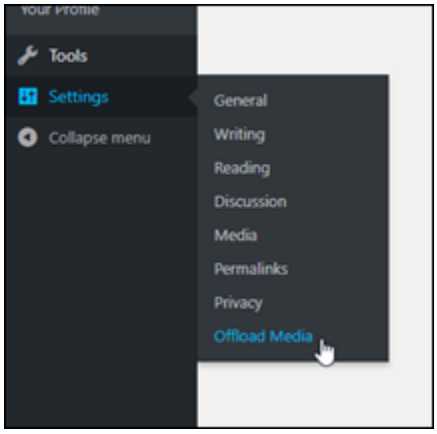
4. 検索結果で、[WP Offload MediaLite] プラグインの横にある [Install Now] (今すぐインストール) を選択します。



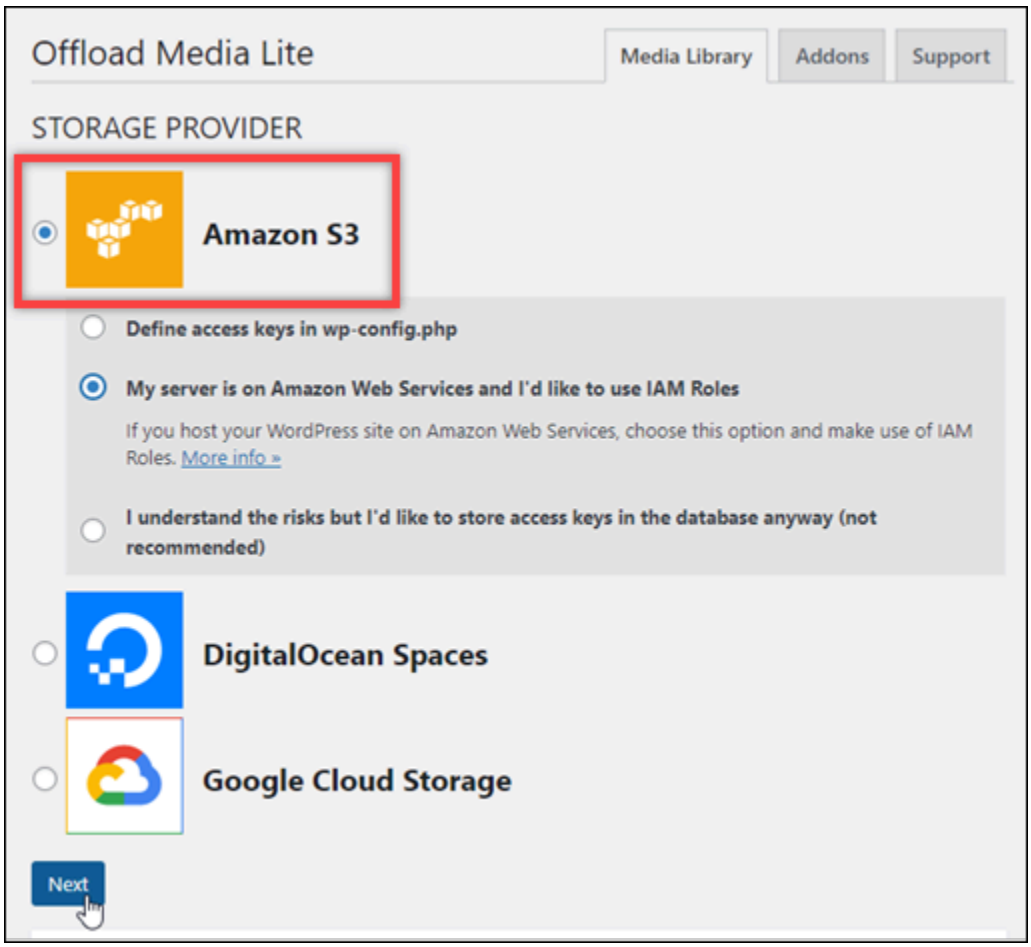
5. プラグインのインストールが完了したら、[アクティベート] を選択します。



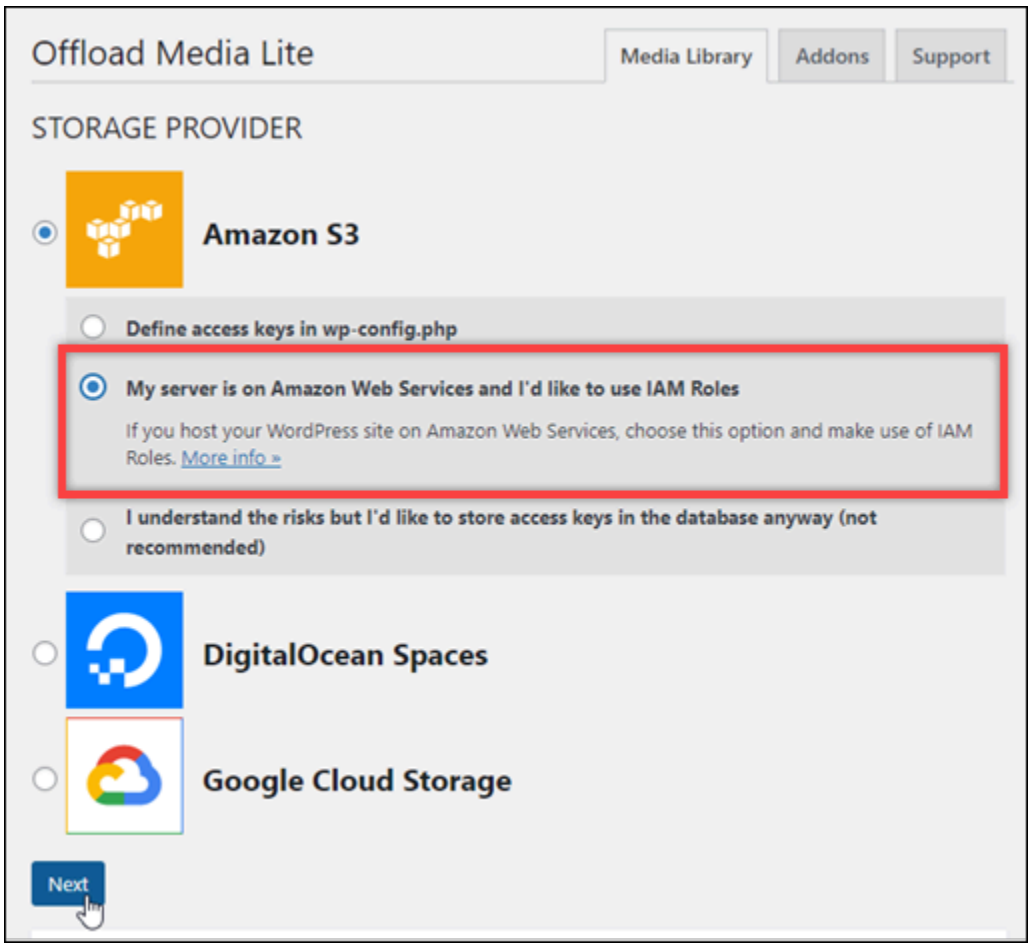
6. 左ナビゲーションメニューで、[設定]、[Offload Media] の順に選択します。



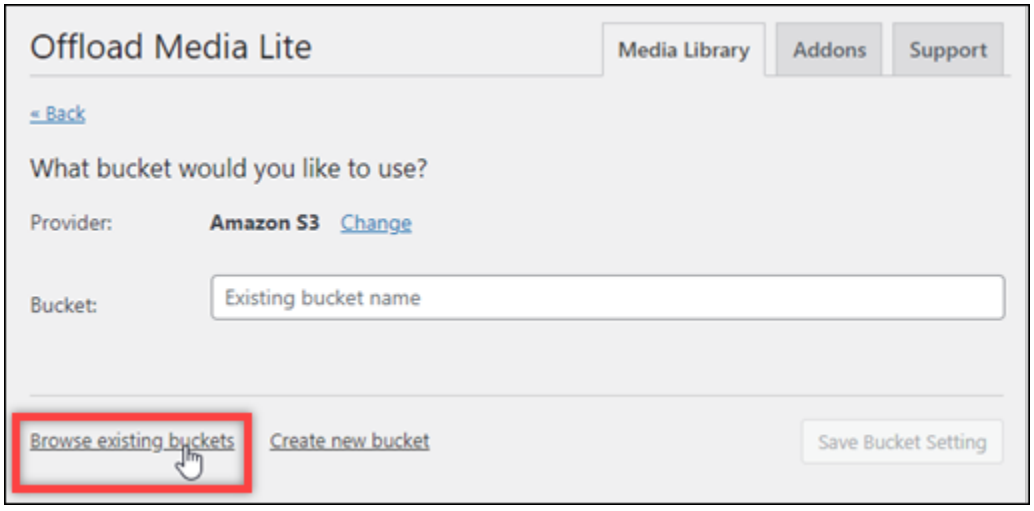
7. [Offload Media Lite] ページで、ストレージプロバイダーとして [Amazon S3] を選択します。



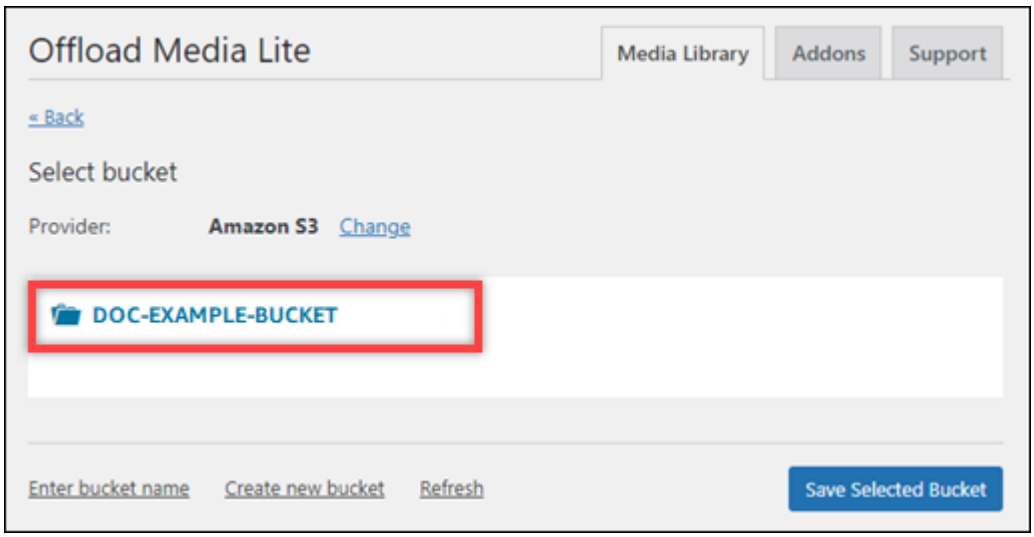
8. [私のサーバーは Amazon Web Services 上にあり、IAM ロールを使いたい] を選択します。



9. [次へ] を選択します。
10. [どのバケットを使用しますか?] と表示される画面で、[Browse existing buckets] (既存のバケットを参照する) を選択します。



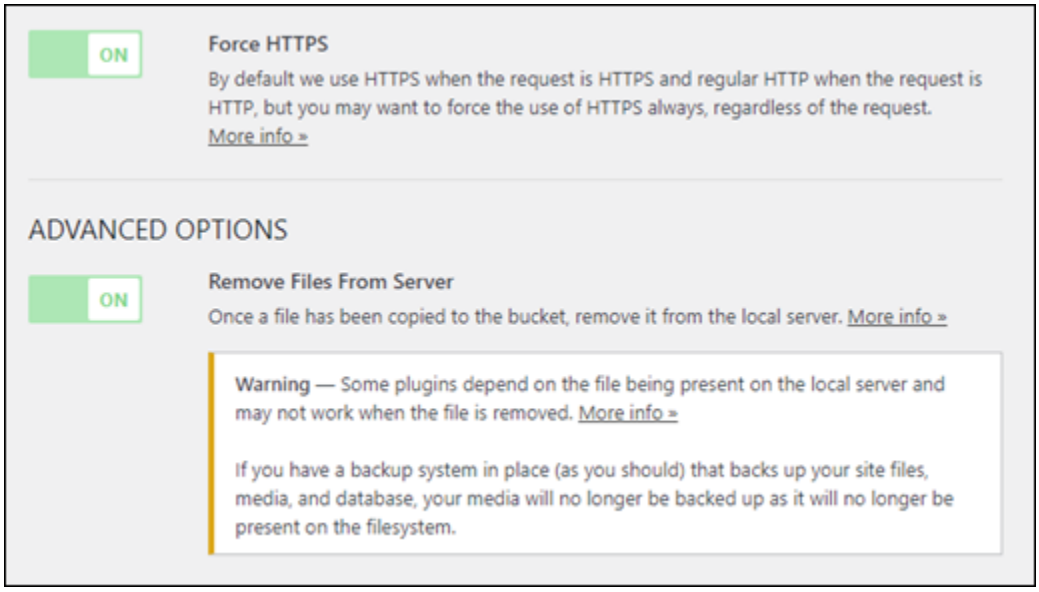
11. WordPress インスタンスで使用するために作成したバケットの名前を選択します。



12. 表示される [Offload Media Lite] 設定画面で、[Force HTTPS] (HTTPS の強制実行) と [Remove Files From Server](サーバーからファイルの削除) をオンにします。

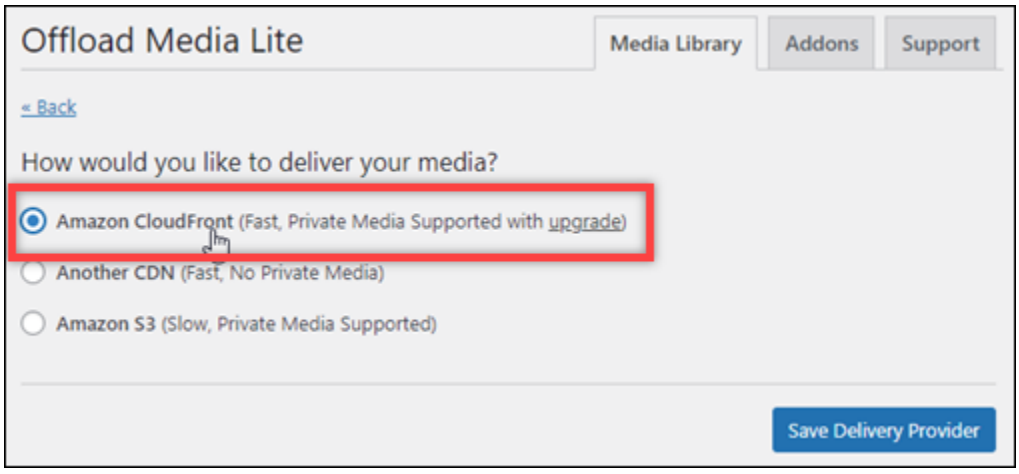
    - Lightsail バケットは、デフォルトで HTTPS を使用してメディアファイルを提供するため、[Force HTTPS] 設定をオンにする必要があります。この機能がオンになっていないと、WordPressのウェブサイトから Lightsail バケットにアップロードされたメディアファイルが、ウェブサイトの訪問者に正しく提供されません。

      [Remove Files From Server] は、Lightsail バケットにアップロードされたメディアがインスタンスのディスクにも保存されないようにする設定です。この機能をオンにしていない場合、Lightsail バケットにアップロードされたメディアファイルは、WordPress インスタンスのローカルストレージにも保存されます。



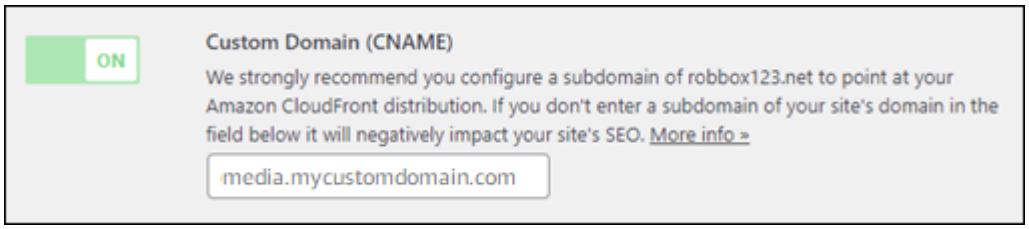
13. ページの [Delivery] セクションで、Amazon S3 ラベルの横にある [変更] を選択します。

14. 表示された [メディアをどのように配信したいですか] ページで、[Amazon CloudFront] を選択します。



15. 配信プロバイダを保存を選択。
16. 表示される [Offload Media Lite 設定] 画面で、[カスタムドメイン (CNAME)]をオンにします。次に、Lightsail ディストリビューションのドメインをテキストボックスに入力します。これは、ディストリビューションのデフォルトドメイン (例 : `123abc.cloudfront.net`) や、ディス

トリビューションのカスタムドメイン (例 : `media.mycustomdomain.com`) 有効にしている場合は、そのドメインになります。



17. [Save changes] (変更の保存) をクリックします。

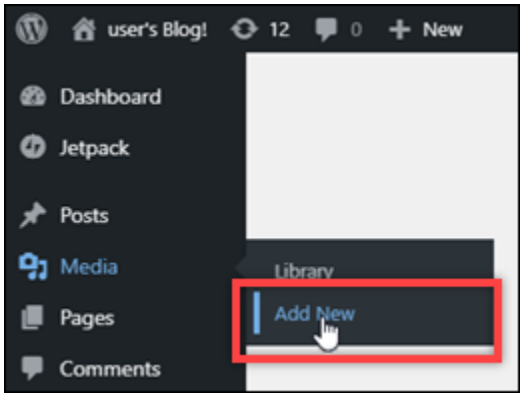
> **Note**
>
> 後で [Offload Media Lite 設定] ページに戻るには、左のナビゲーション メニューで [設定] を一時停止し、[Offload Media] を選択します。

これで、WordPress ウェブサイトは Media Lite プラグインを使用できるように設定されました。次回 WordPress を使用してメディアファイルをアップロードすると、そのファイルは自動的に Lightsail バケットにアップロードされ、ディストリビューションによって提供されます。設定をテストするには、このチュートリアルの次のセクションに進みます。

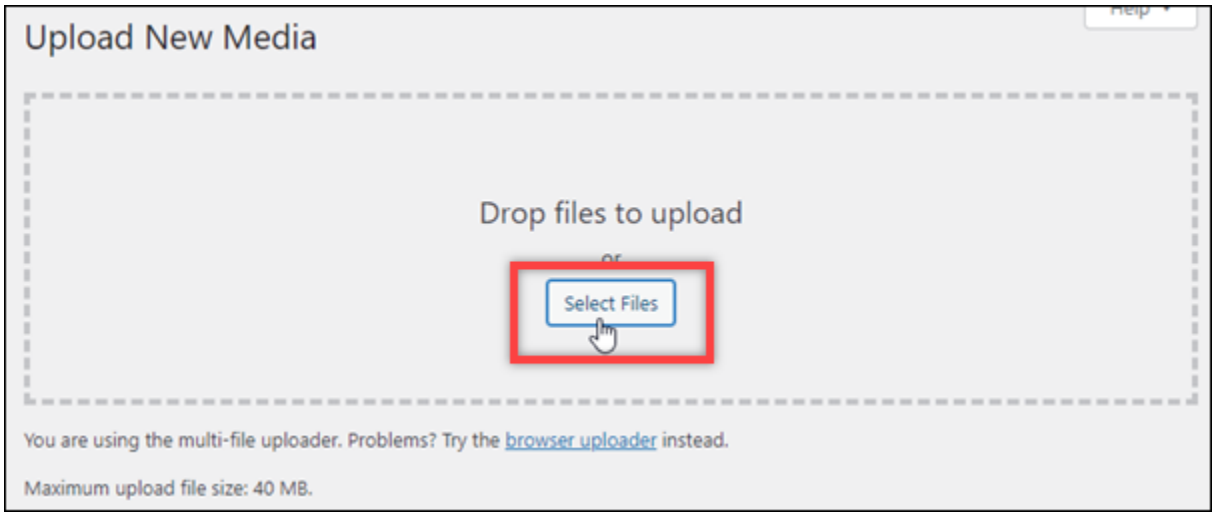
ステップ 6: WordPress ウェブサイトとLightsail バケットとディストリビューションの接続をテストします。

次の手順を実行して、WordPress インスタンスにメディアファイルをアップロードし、それが Lightsail バケットにアップロードされ、ディストリビューションから提供されることを確認します。

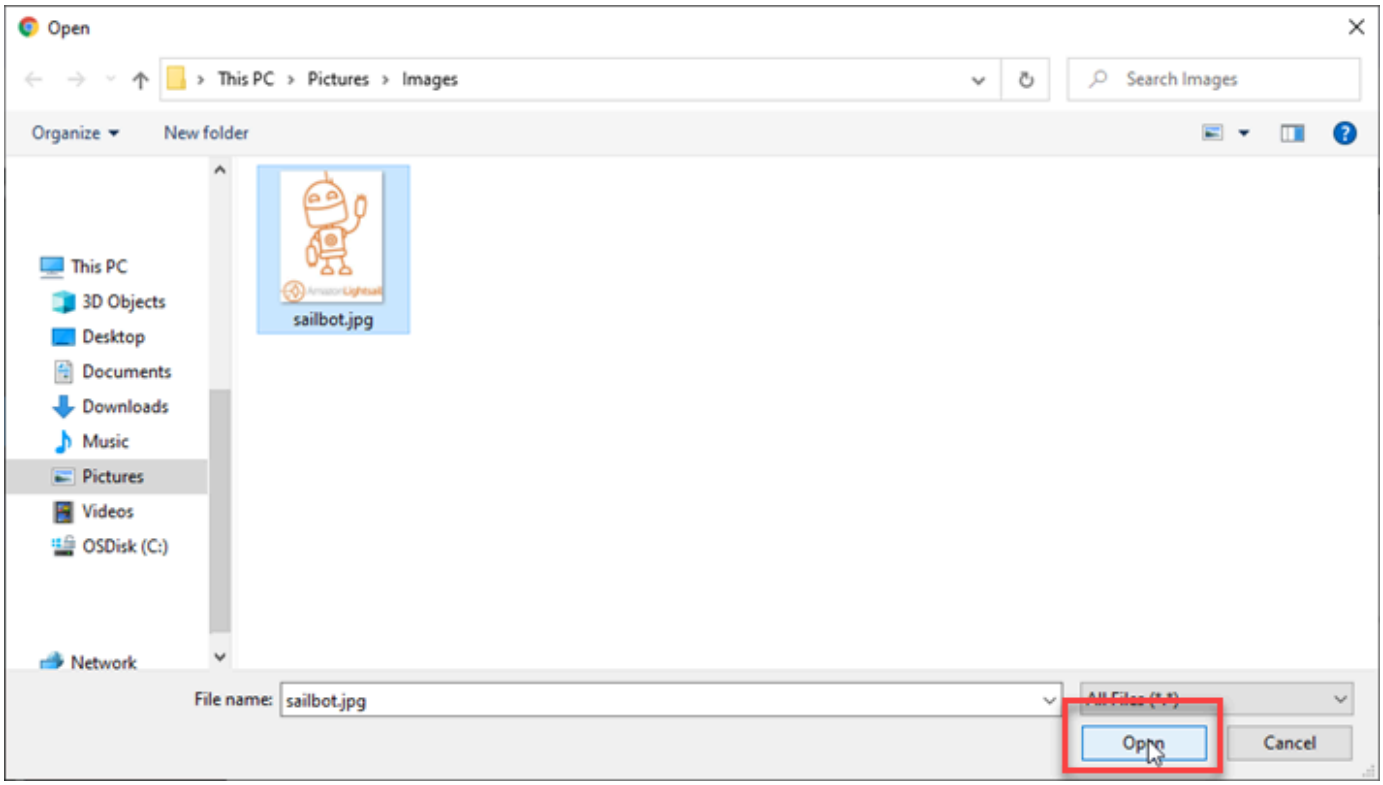
1. WordPress ダッシュボードの左側のナビゲーションメニューで [メディア] を一時停止し、[新規追加]を選択します。



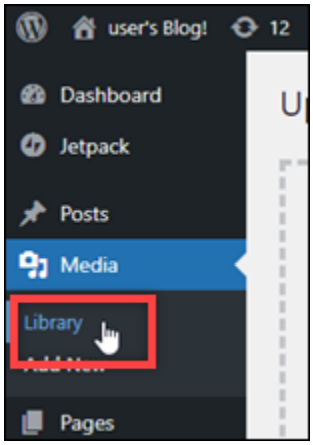
2. 表示される [新しいメディアのアップロード] ページで [ファイルを選択] を選択します。



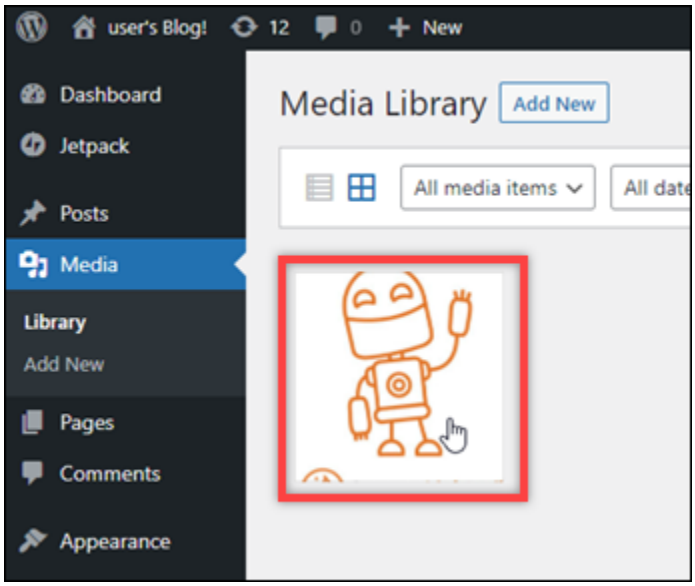
3. ローカルコンピュータからアップロードするメディアファイルを選択し、[開く] を選択します。



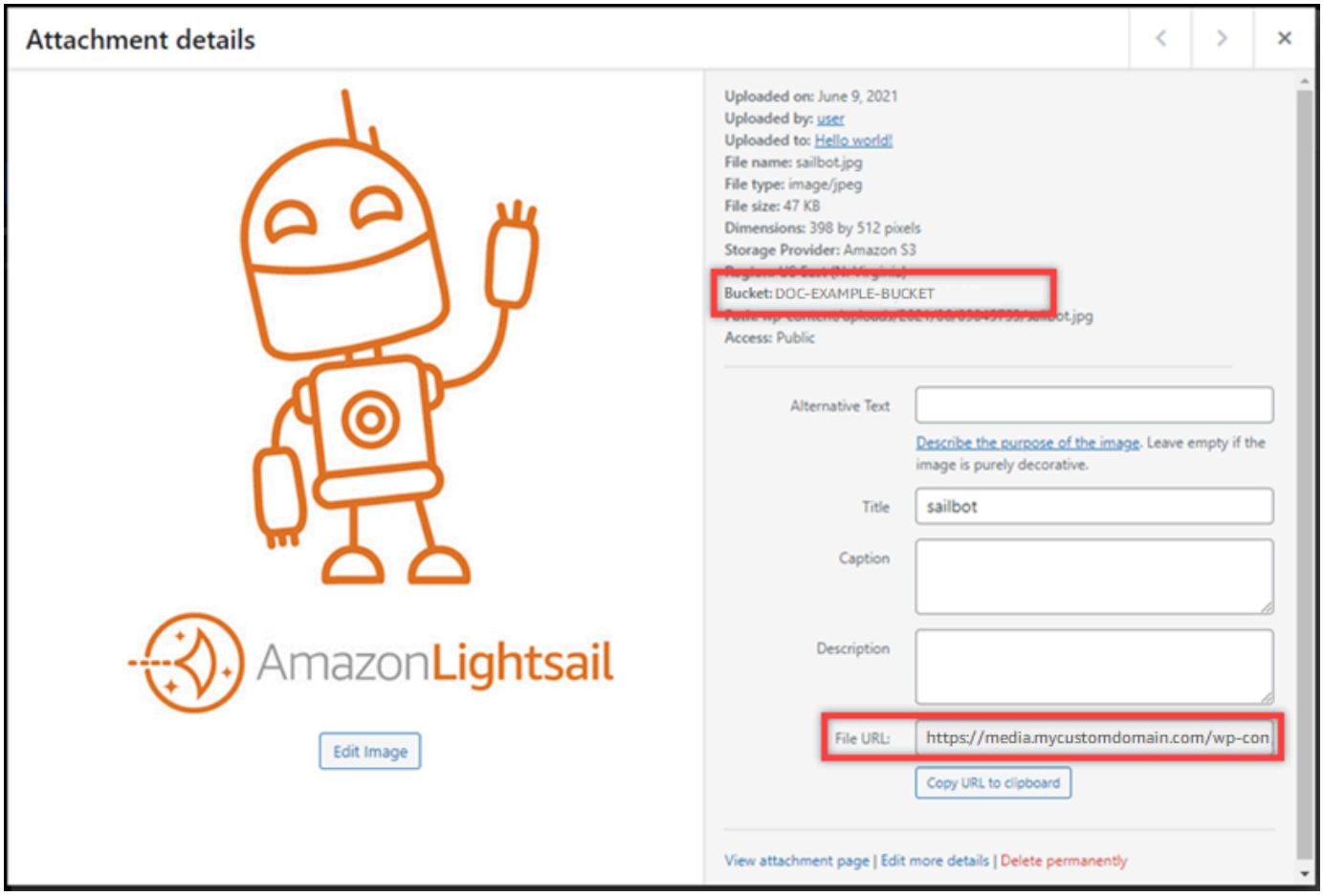
4. ファイルのアップロードが完了したら、左のナビゲーションメニューにある [メディア] の [ライブラリ] を選択します。



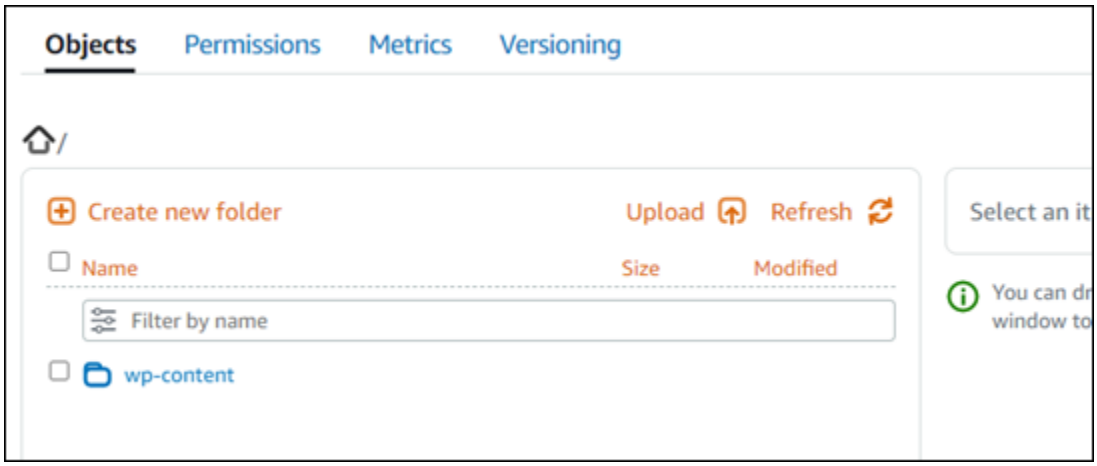
5. 最近アップロードしたファイルを選択します。



6. ファイルの詳細パネルで、[バケット] フィールドにバケットの名前が表示されます。[ファイルの URL」フィールドには、ディストリビューションの URL が表示されます。



7. Lightsail バケット管理ページの [オブジェクト] タブに移動すると、[wp-content] フォルダが表示されます。このフォルダは、Offload Media Lite プラグインによって作成され、アップロードしたメディアファイルを保存するために使用されます。



バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。
3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

   バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

   - Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
   - Amazon Lightsail でのバケットのアクセス許可の設定
   - Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
   - Amazon Lightsail でのバケットのアクセスキーの作成
   - Amazon Lightsail でのバケットのリソースアクセスの設定
   - Amazon Lightsail でのバケットのクロスアカウントアクセスの設定
5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。

   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスの口グ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスの口グ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定

6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail のバケットにファイルをアップロードする
   - マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
   - Amazon Lightsail のバケット内のオブジェクトの表示
   - Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
   - Amazon Lightsail のバケットからのオブジェクトのダウンロード
   - Amazon Lightsail のバケット内のオブジェクトのフィルタリング
   - Amazon Lightsail のバケット内のオブジェクトのタグ付け
   - Amazon Lightsail のバケット内のオブジェクトの削除
9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。
11. バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。
12. バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。
13. ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。
14. バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。
   - チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
   - チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用

15.使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

Lightsail ディストリビューションのデータ転送クォータを調整する

Amazon Lightsail ディストリビューションを作成する際は、毎月のデータ転送クォータとディストリビューションのコストに適したディストリビューションプランを選択します。ディストリビューションにより月間のデータ転送クォータプランを超えるデータが転送された場合、超過料金が発生します。超過料金の詳細については、Lightsail料金表を参照ください。

超過料金の発生を防ぐためには、転送量が月間のクォータを超える前に、現在のディストリビューションプランより転送量が多いプランに変更します。ディストリビューションのプランは、 AWS 請求サイクルごとに 1 回だけ変更できます。このガイドでは、ディストリビューションのプランの変更方法を説明します。

ディストリビューションの詳細については、「コンテンツ配信ネットワークディストリビューション」を参照してください。

ディストリビューションプランを変更する

ディストリビューションのプランを変更するには、以下の手順を行います。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. 参照したい現在の月間データ転送のディストリビューション名を選択します。
4. ディストリビューション管理ページにある詳細タブを選択します。
5. データ転送セクションのページで、ディストリビューションプラン変更を選択します。
6. 確認プロンプトではい、変更しますを選択し、確認します。
7. 次のプロンプトで、新しいディストリビューションプランを選択しプランの選択を選択します。
8. 次のプロンプトで、はい、適用しますを選択して、新しいディストリビューションプランを適用することを確認します。いいえ、戻るを選択すると、新しいプランは適用されません。

Lightsail ディストリビューションのカスタムドメインで内容を処理する

登録済みドメイン名をディストリビューションで使用するために、カスタムドメインを Amazon Lightsail ディストリビューションと動作するように設定します。カスタムドメインを有効にする前は、ディストリビューションを最初に作成したときに関連付けられたデフォルトドメイン (例: `123456abcdef.cloudfront.net`) に対してのみ、ディストリビューションはトラフィックを受け入れます。カスタムドメインを有効にする際、ディストリビューションと動作するように作成したドメインの Lightsail SSL/TLS 証明書を選択する必要があります。カスタムドメインを有効にすると、選択した証明書に関連付けられているすべてのドメインのトラフィックがディストリビューションで受け入れられます。

> **Important**
>
> ディストリビューション事に、一度に 1 つの証明書のみを使用することができます。ディストリビューションでカスタムドメインを無効にすると、カスタムドメインを再度有効にするまで、登録したドメインの HTTPS トラフィックをディストリビューションで処理できなくなります。
>
> SSL/TLS 証明書に関連付けられたドメイン名は、Amazon CloudFront サービスのディストリビューションを含む、すべての Amazon Web Services (AWS) アカウントのディストリビューションで使用することはできません。ドメインの証明書を作成することはできますが、ディストリビューションと使用することはできません。

ディストリビューションの詳細については、「コンテンツ配信ネットワークディストリビューション」を参照してください。

前提条件

開始する前に、Lightsail ディストリビューションを作成する必要があります。詳細については、「ディストリビューションを作成する」を参照してください。

ディストリビューション用の SSL/TLS 証明書の作成と検証が必要です。詳細については、「ディストリビューションの SSL/TLS 証明書を作成する」および「ディストリビューションの SSL/TLS 証明書を検証する」を参照してください。

ディストリビューションのカスタムドメインを有効にする

ディストリビューションのカスタムドメインを有効にするには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. カスタムドメインを有効にするディストリビューションの名前を選択します。
4. ディストリビューション管理ページで [カスタムドメイン] タブを選択します。
5. [証明書のアタッチ] を選択します。

   証明書がない場合は、ディストリビューションにアタッチする前に、ドメインの SSL/TLS 証明書を作成してから検証する必要があります。詳細については、「ディストリビューションの SSL/TLS 証明書を作成する」を参照してください。
6. 表示されるドロップダウンメニューで、ディストリビューションと使用するドメインの有効な証明書を選択します。
7. 証明書情報が正しいことを確認し、[アタッチ] を選択します。
8. ディストリビューションの [Status] (ステータス) が [Updating] (更新中) に変わります。ステータスが [Enabled] (使用可能) に変わると、証明書のドメインが [Custom domains] (カスタムドメイン) セクションに表示されます。
9. [Add domain assignment] (ドメイン割り当てを追加) を選択して、ドメインがディストリビューションを指すようにします。
10. 証明書と DNS 情報が正しいことを確認し、[Add assignment] (割り当てを追加) を選択します。しばらくすると、選択したドメインのトラフィックがディストリビューションによって受け入れられ始めます。

トピック

- カスタムドメインを Lightsail ディストリビューションにポイントする
- Lightsail ディストリビューションの SSL/TLS 証明書ドメインを更新する
- Lightsail ディストリビューションのカスタムドメインを無効にする
- ディストリビューションのデフォルトドメインを Lightsail コンテナサービスに追加する

カスタムドメインを Lightsail ディストリビューションにポイントする

ディストリビューションのカスタムドメインを有効にしたら、メンバードメイン名を Amazon Lightsail ディストリビューションに向ける必要があります。ディストリビューションで使用中の証明書に指定されている、各ドメインの DNS ゾーンにエイリアスレコードを追加して行います。追加するレコードはすべて、ディストリビューションのデフォルトのドメイン (例: `123456abcdef.cloudfront.net`) に向ける必要があります。

このガイドでは、Lightsail DNS ゾーンを使用してディストリビューションにドメインを向ける手順について説明しています。Domain.com や GoDaddy などの別の DNS ホスティングプロバイダーを使用して、ドメインをディストリビューションに向ける手順と類似しているかもしれません。Lightsail DNS ゾーンの詳細については、「DNS」を参照してください。

ディストリビューションの詳細は、「ディストリビューションを作成する」を参照してください。

目次

- ステップ 1: 前提条件を満たす
- ステップ 2: ディストリビューションのデフォルトドメインを取得する
- ステップ 3: ドメインの DNS ゾーンにレコードを追加する

ステップ 1: 前提条件を満たす

開始するには、Lightsail ディストリビューション用のカスタムドメインを有効にします。詳細については、「ディストリビューション用のカスタムドメインを有効にする」を参照してください。

ステップ 2: ディストリビューションのデフォルトドメインを取得する

以下の手順を実行して、ディストリビューションのデフォルトドメイン名を取得します。このドメイン名は、ドメインの DNS にエイリアスレコードを追加するときに指定します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. デフォルトのドメイン名を取得したいディストリビューション名を選択します。
4. ディストリビューションの管理ページのヘッダーセクションにある、ディストリビューションのデフォルトドメイン名を書き留めます。ディストリビューションのデフォルトドメイン名は `123456abcdef.cloudfront.net` と類似しています。

この値は、ドメイン DNS のエイリアスレコードのパートとして、追加する必要があります。この値はテキストファイルにコピー、ペーストして、後で参照できるようにしておくことをお勧めします。このチュートリアルの次の「ステップ 3: ドメインの DNS ゾーンにレコードを追加する」セクションに進みます。

ステップ 3: ドメインの DNS ゾーンにレコードを追加する

ドメインの DNS ゾーンにレコードを追加するには、次のステップを実行します。

1. 左側のナビゲーションペインで [ドメインと DNS] を選択します。
2. ページの [DNS zones] (DNS ゾーン) セクションで、レコードを追加したいドメイン名を選択します。そのレコードがユーザーのドメインへのトラフィックをディストリビューションに送信します。
3. [DNS records] (DNS レコード) タブを選択します。次に、[Add record] (レコードの追加) を選択します。
4. ディストリビューションにポイントするドメインのタイプに応じて、以下のいずれかの手順を実行します。

   - アドレス (A) レコードを選択して、頂点ドメイン (例: `example.com`) を ディストリビューション)にポイントします。

     ドメインの頂点の A レコードが DNS ゾーンにすでに存在する場合は、別の A レコードを追加するのではなく、既存のレコードを編集する必要があります。
   - 正規名 (CNAME) を選択して、`website.example.com` などのサブドメインをディストリビューションに対しポイントします。
5. A レコードを追加する場合は、[Resolves to] (解決先) テキストボックスでディストリビューション名を選択します。CNAME レコードを追加する場合は、[Maps to] (マッピング先) テキストボックスにディストリビューションのデフォルトドメイン名を入力します。

> **Note**
>
> DNS ゾーンに A レコードを追加してディストリビューション名を選択すると、アドレスレコードとは異なるエイリアスレコードが追加されます。Lightsail では、他の DNS ホスティングプロバイダーでは通常必要とされる追加ステップを行うことなく、エイリアスレコードを簡単に追加できます。

6. 保存アイコンを選択して、レコードを DNS ゾーンに保存します。

   これらの手順を繰り返すと、ディストリビューションで使用している証明書と紐づくドメイン用に、他の DNS レコードも追加できます。変更がインターネットの DNS を通じて伝達されるまで待ちます。数分後に、ドメインがディストリビューションをポイントしているか確認してください。なお、ディストリビューションもテストする必要があります。詳細については、次の「ディストリビューションのテスト」を参照してください。

Lightsail ディストリビューションの SSL/TLS 証明書ドメインを更新する

Amazon Lightsail ディストリビューションが使用しているカスタムドメインを他のドメインあるいはドメインのセットに変更することができます。変更するには、ディストリビューションで使用するドメイン用の新しい SSL/TLS 証明書を作成する必要があります。詳細については、「ディストリビューションの SSL/TLS 証明書を作成する」を参照してください。　新しい証明書が検証されたら、古い証明書を新しい証明書とスワップします。これにより、ディストリビューションのカスタムドメインが変更されます。

ディストリビューションの詳細は、「ディストリビューションを作成する」を参照してください。

ディストリビューションのカスタムドメインを変更する

ディストリビューションのカスタムドメインを変更するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. カスタムドメインを変更するディストリビューションの名前を選択します。
4. ディストリビューション管理ページのカスタムドメインタブを選択します。
5. ディストリビューションに現在アタッチされている SSL/TLS 証明書のアタッチを解除します。

   ディストリビューションのステータスが [In progress] (進行中) に変化します。
6. ディストリビューションのステータスが [Enabled] (有効) に戻ったら、[Attach certificate] (証明書をアタッチ) を選択します。
7. 表示されるドロップダウンメニューで、ディストリビューションと使用するドメインの有効な証明書を選択します。
8. 証明書情報が正しいことを確認し、[Attach] (アタッチ) を選択します。
9. ドメインの DNS にドメイン割り当てを追加して、ドメインがディストリビューションを指すようにします。

ディストリビューションの [Status] (ステータス) が [Updating] (更新中) に変わります。ステータスが [Ready] (準備完了) に変わると、証明書のドメインが [Custom domains] (カスタムドメイン) セクションに表示されます。[Add domain assignment] (ドメイン割り当てを追加) を選択して、ドメインがディストリビューションを指すようにします。

10. [Add assignment] (割り当てを追加) を選択します。しばらくすると、選択したドメインのトラフィックがディストリビューションによって受け入れられ始めます。
11. [保存] を選択します。

Lightsail ディストリビューションのカスタムドメインを無効にする

Amazon Lightsail ディストリビューションのカスタムドメインを無効にし、登録済みドメイン名のディストリビューションとの使用を停止します。カスタムドメインを無効にすると、ディストリビューションは、最初に作成したときにディストリビューションに関連付けられたデフォルトドメイン (例:`123456abcdef.cloudfront.net`) のみを受け入れます。以前に関連付けられたカスタムドメインのトラフィックには 403 エラーが表示されます。

ディストリビューションの詳細については、「コンテンツ配信ネットワークディストリビューション」を参照してください。

ディストリビューションのカスタムドメインを無効にする

ディストリビューションのカスタムドメインを無効にするには、以下の手順を行います。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. カスタムドメインを無効にするディストリビューションの名前を選択します。
4. ディストリビューションの管理ページのカスタムドメインタブを選択します。

   [Custom domains] (カスタムドメイン) ページには、ディストリビューションに現在アタッチされている SSL/TLS 証明書があれば表示されます。
5. 以下のオプションのいずれかを選択してください。

   1. [Configure distribution domains] (ディストリビューションドメインを設定する) を選択して、以前に選択したドメインの選択を解除するか、ディストリビューションに関連付けられているドメインをさらに選択します。

2. [デタッチ] を選択してディストリビューションから証明書をデタッチし、関連付けられているすべてのドメインを削除します。

6. カスタムドメインを無効にするリクエストが送信され、ディストリビューションのステータスが [In progress] (進行中) へ変更されます。しばらくすると、ディストリビューションのステータスが[Enabled] (有効) に変更されます。

カスタムドメインを無効にすると、ディストリビューションは、最初に作成したときにディストリビューションに関連付けられたデフォルトドメイン (例:123456abcdef.cloudfront.net) のみを受け入れます。以前に関連付けられたカスタムドメインのトラフィックには 403 エラーが表示されます。ドメインの DNS レコードを更新して、それらのドメインのトラフィックが別のリソースに送信されるようにする必要があります。

ディストリビューションのデフォルトドメインを Lightsail コンテナサービスに追加する

Amazon Lightsail コンテナサービスをコンテンツ配信ネットワーク (CDN) ディストリビューションのオリジンとして選択できます。そうすると、ディストリビューションでは、コンテナサービスでホストされているウェブサイトまたはウェブアプリケーションがキャッシュされ提供されます。Lightsail コンテナサービスを使用した Lightsail ディストリビューションを使用している場合、Lightsail によって、ディストリビューションのデフォルトドメイン名がカスタムドメインとしてコンテナサービスに自動的に追加されます。これにより、ディストリビューションとコンテナサービスの間でトラフィックをルーティングできます。しかし、以下の状況においては、このガイドで説明されている手順を実行して、ディストリビューションのデフォルトドメイン名をコンテナサービスに手動で追加する必要があります。

- 何らかの不具合により、ディストリビューションのデフォルトドメイン名がコンテナサービスに自動的に追加されない場合。
- Lightsail ディストリビューション以外のディストリビューションをコンテナサービスで使用している場合。

() を使用してのみ、ディストリビューションのデフォルトドメイン名をコンテナサービスに手動で追加できます AWS Command Line Interface AWS CLI。コンテナサービスの詳細については、「コンテナサービス」を参照してください。ディストリビューションの詳細については、「オブジェクトストレージ」を参照してください。

ディストリビューションのデフォルトドメインをコンテナサービスに追加する

AWS Command Line Interface () Lightsailを使用して のコンテナサービスにディストリビューションのデフォルトドメインを追加するには、次の手順を実行しますAWS CLI。これは、update-container-service コマンドを使用して実行できます。詳細については、「AWS CLI コマンドリファレンス」の「update-container-service」を参照してください。

> **Note**
>
> この手順を続行するLightsail前に、 をインストール AWS CLI して 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 以下のコマンドのいずれかを入力して、ディストリビューションのデフォルトドメインをコンテナサービスに追加します。

   > **Note**
   >
   > コンテナサービスにカスタムドメインを追加した場合は、カスタムドメインとディストリビューションのデフォルトドメインの両方を指定する必要があります。

   コンテナサービスにカスタムドメインが設定されていない場合:

   ```
   aws lightsail update-container-service --service-name ContainerServiceName --
   public-domain-names '{"_": ["DistributionDefaultDomain"]}'
   ```

   コンテナサービスに 1 つまたは複数のカスタムドメインが設定されている場合:

   ```
   aws lightsail update-container-service --service-name ContainerServiceName
    --public-domain-names '{"CertificateName": ["ExistingCustomDomain"],"_":
    ["DistributionDefaultDomain"]}'
   ```

   コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *ContainerServiceName* - ディストリビューションのオリジンとして指定された Lightsail コンテナサービスの名前。
- *DistributionDefaultDomain* - コンテナサービスをオリジンとして使用しているディストリビューションのデフォルトドメイン。例えば、example123.cloudfront.net。
- *CertificateName* - コンテナサービスに現在アタッチされているカスタムドメインがある場合、その Lightsail 証明書の名前。コンテナサービスにアタッチされたカスタムドメインがない場合は、コンテナサービスでカスタムドメインが設定されていないというラベルの付いたコマンドを使用します。
- *DistributionDefaultDomain* - コンテナサービスに現在アタッチされているカスタムドメイン。

例:

- コンテナサービスにカスタムドメインが設定されていない場合:

```
aws lightsail update-container-service --service-name ContainerServiceName --
public-domain-names '{"_": ["example123.cloudfront.net"]}'
```

- コンテナサービスに 1 つまたは複数のカスタムドメインが設定されている場合:

```
aws lightsail update-container-service --service-name ContainerServiceName
 --public-domain-names '{"example-com": ["example.com"],"_":
 ["example123.cloudfront.net"]}'
```

Lightsail ディストリビューションのリクエストとレスポンスの動作を管理する

このガイドでは、Amazon Lightsail ディストリビューションが、リクエストを処理してオリジンに転送し、オリジンからのレスポンスを処理するときの動作に関して説明します。ディストリビューションの詳細については、「コンテンツ配信ネットワークディストリビューション」を参照してください。

トピック

- ディストリビューションがリクエストを処理してオリジンに転送する方法
- ディストリビューションがオリジンからの応答を処理する方法

ディストリビューションがリクエストを処理してオリジンに転送する方法

このトピックには、ディストリビューションがビューワーリクエストを処理してオリジンに転送する方法に関する情報が含まれています。

目次

- 認証
- キャッシュ期間
- クライアント IP アドレス
- クライアント側の SSL 認証
- 圧縮
- 条件付きリクエスト
- cookie
- クロスオリジンリソース共有 (CORS)
- 暗号化
- 本文を含む GET リクエスト
- HTTP メソッド
- HTTP リクエストヘッダーとディストリビューション動作
- HTTP バージョン
- リクエストの最大長と URL の最大長
- OCSP Stapling
- 持続的接続
- プロトコル
- クエリ文字列
- オリジン接続のタイムアウトと試行
- オリジン応答タイムアウト
- 同じオブジェクト (トラフィックスパイク) の同時リクエスト
- ユーザーエージェントヘッダー

認証

DELETE、GET、HEAD、PATCH、POST、PUT リクエストの場合、Authorization ヘッダーをオリジンに転送するように ディストリビューションを設定すると、クライアント認証を要求するようにオリジンサーバーを設定できます。

OPTIONS リクエストの場合、次のディストリビューション設定を使用した場合のみ、クライアント認証を要求するようにオリジンサーバーを設定することができます。

- Authorization ヘッダーをオリジンに転送するようにディストリビューションを設定する。
- OPTIONS リクエストへの応答をキャッシュしないようにディストリビューションを設定する。

HTTP または HTTPS のいずれかを使用してオリジンにリクエストを転送するようにディストリビューションを構成することができます。

キャッシュ期間

ディストリビューションが別のリクエストをオリジンに転送するまでにオブジェクトをキャッシュに保持する時間をコントロールするには :

- Cache-Control または Expires ヘッダーフィールドを各オブジェクトに追加するようにオリジンを構成します。
- キャッシュ寿命 (TTL) には、デフォルト値の 1 日を使用します。

ディストリビューション設定の詳細については、「ディストリビューションアドバンスト設定」を参照してください。

クライアント IP アドレス

ビューワーがリクエストをディストリビューションに送信し、X-Forwarded-For リクエストヘッダーを含めない場合、ディストリビューションは TCP 接続からビューワーの IP アドレスを取得して、IP アドレスが含まれた X-Forwarded-For ヘッダーを追加し、リクエストをオリジンに転送します。たとえば、ディストリビューションが TCP 接続から IP アドレス 192.0.2.2 を取得する場合、以下のヘッダーをオリジンに転送します。

```
X-Forwarded-For: 192.0.2.2
```

ビューワーがリクエストをディストリビューションに転送して X-Forwarded-For リクエストヘッダーを含める場合、 ビューワーの IP アドレスを TCP 接続から取得してそれを X-Forwarded-For

ヘッダーの末尾に追加し、リクエストをオリジンに転送します。たとえば、ビューワーのリクエストに X-Forwarded-For: 192.0.2.4,192.0.2.3 が含まれ、ディストリビューションが TCP 接続から IP アドレス 192.0.2.2 を取得する場合、以下のヘッダーをオリジンに転送します。

```
X-Forwarded-For: 192.0.2.4,192.0.2.3,192.0.2.2
```

ロードバランサー (Elastic Load Balancing を含む)、ウェブアプリケーションファイアウォール、リバースプロキシ、侵入防御システム、API Gateway などの一部のアプリケーションでは、リクエストを転送したディストリビューションエッジサーバーの IP アドレスを X-Forwarded-For ヘッダーの末尾に付加します。たとえば、ディストリビューションから ELB に転送するリクエストに X-Forwarded-For: 192.0.2.2 が含まれていて、 エッジサーバーの IP アドレスが 192.0.2.199 である場合、インスタンスで受け取るリクエストのヘッダーは次のようになります。

```
X-Forwarded-For: 192.0.2.2,192.0.2.199
```

> **Note**
>
> X-Forwarded-For ヘッダーには、IPv4 アドレス (192.0.2.44 など) および IPv6 アドレス (2001:0db8:85a3:0000:0000:8a2e:0370:7334 など) が含まれます。

クライアント側の SSL 認証

Lightsail ディストリビューションはクライアント側の SSL 証明書を使用したクライアント認証をサポートしていません。オリジンがクライアント側証明書をリクエストした場合、 ディストリビューションはリクエストを削除します。

Compression

Lightsail ディストリビューションは、"identity" および "gzip" の Accept-Encoding フィールド値を持つリクエストを転送します。

条件付きリクエスト

ディストリビューションは、エッジキャッシュで有効期限切れになっているオブジェクトに対するリクエストを受け取ると、リクエストをオリジンに転送し、オブジェクトの最新バージョンを取得するか、 エッジキャッシュに最新バージョンが既に存在することをオリジンに確認します。通常、オリジンはオブジェクトをディストリビューションに最後に送信するときに、ETag 値または LastModified 値、あるいはその両方の値をレスポンスに含めます。ディストリビューションがオリジンに転送する新しいリクエストには、次のどちらかまたは両方を追加します。

- オブジェクトの有効期限切れバージョンの `If-Match` 値が含まれる `If-None-Match` または `ETag` ヘッダー。
- オブジェクトの有効期限切れバージョンの `If-Modified-Since` 値が含まれる `LastModified` ヘッダー。

オリジンは、この情報を使用して、オブジェクトが更新されているかどうかを判別します。つまり、オブジェクト全体をディストリビューションに返すか、または HTTP 304 ステータスコード (変更なし) のみを返すかを判別します。

cookie

Cookie をオリジンに転送するようにディストリビューションを構成できます。詳細については、「ディストリビューションアドバンスト設定」を参照してください。

クロスオリジンリソース共有 (CORS)

ディストリビューションで Cross-Origin Resource Sharing 設定を尊重する場合は、 `Origin` ヘッダーをオリジンに転送するように設定します。

暗号化

ビューワーに HTTPS を使用してディストリビューションに接続するように要求し、HTTP または HTTPS を使用してリクエストをオリジンに転送するようにディストリビューションに要求することができます。

ディストリビューションは、SSLv3、TLSv1.0、TLSv1.1、および TLSv1.2 プロトコルを使用して、HTTPS リクエストをオリジンに転送します。SSL と TLS のその他のバージョンはサポートされていません。

本文を含む GET リクエスト

ビューワーの `GET` リクエストの本文が含まれている場合、 ディストリビューションはビューワーに HTTP ステータスコード 403 (禁止) を返します。

HTTP メソッド

サポートするすべての HTTP メソッドを許可するようディストリビューションを構成すると、ディストリビューションはビューワーからの以下のリクエストを受け入れてオリジンに転送します。

- DELETE
- GET
- HEAD
- OPTIONS
- PATCH
- POST
- PUT

ディストリビューションは、GET リクエストと HEAD リクエストへの応答を常にキャッシュします。OPTIONS リクエストへの応答をキャッシュするように設定することもできます。ディストリビューションはその他のメソッドを使用するリクエストへのレスポンスをキャッシュしません。

オリジンが上記のメソッドを処理するかどうかを構成する方法の詳細については、オリジンのドキュメントを参照してください。

> **Important**
>
> ディストリビューションがサポートするすべての HTTP メソッドを受け入れてオリジンに転送するように設定する場合、オリジンサーバーがすべてのメソッドを処理するように構成します。たとえば、POST を使用するために、上記のメソッドを受け入れて転送するようにディストリビューションを構成する場合は、DELETE リクエストを適切に処理するようオリジンサーバーを設定して、削除すべきでないリソースをビューワーが削除できないようにする必要があります。詳細については、HTTP サーバーのドキュメントを参照してください。

HTTP リクエストヘッダーとディストリビューション動作

次の表は、オリジンに転送できる HTTP リクエストヘッダーを示しています (例外も注記されています)。この表には、各ヘッダーについて以下に関する情報も含まれています。

- サポート - そのヘッダーの値に基づいてオブジェクトをキャッシュするようにディストリビューションを設定できるかどうか。

  Date および User-Agent ヘッダーの値に基づいてオブジェクトをキャッシュするようにディストリビューションを設定できますが、これはお勧めできません。これらのヘッダーには可能な値が多数あり、その値に基づいてキャッシュすると、ディストリビューションがオリジンに転送するリクエストの数が大幅に増加します。

- 設定していない場合の動作 - ヘッダーをオリジンに転送するように設定していない場合、ディストリビューションはヘッダー値に基づいてオブジェクトをキャッシュします。

- ヘッダー - 他の定義されたヘッダー

  サポート - あり

  設定されていない場合の動作 - ディストリビューションはヘッダーをオリジンに転送します。
- ヘッダー - Accept

  サポート - あり

  設定されていない場合の動作 - ディストリビューションはヘッダーを削除します。
- ヘッダー - Accept-Charset

  サポート - あり

  設定されていない場合の動作 - ディストリビューションはヘッダーを削除します。
- ヘッダー - Accept-Encoding

  サポート - あり

  設定されていない場合の動作 - 値に gzip が含まれる場合、ディストリビューションは Accept-Encoding: gzip をオリジンに転送します。値に gzip が含まれない場合、ディストリビューションはリクエストをオリジンに転送する前に Accept-Encoding ヘッダーフィールドを削除します。
- ヘッダー - Accept-Language

  サポート - あり

  設定されていない場合の動作 - ディストリビューションはヘッダーを削除します。
- ヘッダー - Authorization

  サポート - あり

  設定されていない場合の動作:

  - GET、HEAD の各リクエスト – ディストリビューションは、リクエストをオリジンに転送する前に Authorization ヘッダーフィールドを削除します。

  - OPTIONS リクエスト – OPTIONS リクエストへの応答をキャッシュするようにディストリビューションを設定した場合、ディストリビューションは、リクエストをオリジンに転送する前に、Authorization ヘッダーフィールドを削除します。

    OPTIONS リクエストへの応答をキャッシュするようにディストリビューションを設定しなかった場合、ディストリビューションは Authorization ヘッダーフィールドをオリジンに転送します。

  - DELETE、PATCH、POST、PUT の各リクエスト – ディストリビューションは、リクエストをオリジンに転送する前にヘッダーフィールドを削除しません。

- ヘッダー - Cache-Control

  サポート - なし

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。

- ヘッダー - CloudFront-Forwarded-Proto

  サポート - あり

  設定されていない場合の動作 - ディストリビューションは、リクエストをオリジンに転送する前にヘッダーを追加しません。

- ヘッダー - CloudFront-Is-Desktop-Viewer

  サポート - あり

  設定されていない場合の動作 - ディストリビューションは、リクエストをオリジンに転送する前にヘッダーを追加しません。

- ヘッダー - CloudFront-Is-Mobile-Viewer

  サポート - あり

  設定されていない場合の動作 - ディストリビューションは、リクエストをオリジンに転送する前にヘッダーを追加しません。

- ヘッダー - CloudFront-Is-Tablet-Viewer

  サポート - あり

  設定されていない場合の動作 - ディストリビューションは、リクエストをオリジンに転送する前にヘッダーを追加しません。

- ヘッダー - CloudFront-Viewer-Country

  サポート - あり

  設定されていない場合の動作 - ディストリビューションは、リクエストをオリジンに転送する前にヘッダーを追加しません。
- ヘッダー - Connection

  サポート - なし

  設定されていない場合の動作 - ディストリビューションは、オリジンに転送する前に、このヘッダーをConnection: Keep-Aliveで置き換えます。
- ヘッダー - Content-Length

  サポート - なし

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。
- ヘッダー - Content-MD5

  サポート - あり

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。
- ヘッダー - Content-Type

  サポート - あり

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。
- ヘッダー - Cookie

  サポート - なし

  設定されていない場合の動作 - Cookie を転送するようにディストリビューションを設定している場合、Cookieヘッダーフィールドをオリジンに転送します。そうでない場合、ディストリビューションは Cookie ヘッダーフィールドを削除します。
- ヘッダー - Date

  サポート対象 - あり、ただし推奨されません

  設定されていない場合の動作 - ディストリビューションはヘッダーをオリジンに転送します。
- ヘッダー - Expect

サポート - あり

設定されていない場合の動作 - ディストリビューションはヘッダーを削除します。

- ヘッダー - From

  サポート - あり

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。

- ヘッダー - Host

  サポート - あり

  設定されていない場合の動作 - ディストリビューションは、リクエストされたオブジェクトに関連付けられたオリジンのドメイン名に値を設定します。

- ヘッダー - If-Match

  サポート - あり

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。

- ヘッダー - If-Modified-Since

  サポート - あり

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。

- ヘッダー - If-None-Match

  サポート - あり

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。

- ヘッダー - If-Range

  サポート - あり

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。

- ヘッダー - If-Unmodified-Since

  サポート - あり

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。

- ヘッダー - Max-Forwards

  サポート - なし

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。
- ヘッダー - Origin

  サポート - あり

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。
- ヘッダー - Pragma

  サポート - なし

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。
- ヘッダー - Proxy-Authenticate

  サポート - なし

  設定されていない場合の動作 - ディストリビューションはヘッダーを削除します。
- ヘッダー - Proxy-Authorization

  サポート - なし

  設定されていない場合の動作 - ディストリビューションはヘッダーを削除します。
- ヘッダー - Proxy-Connection

  サポート - なし

  設定されていない場合の動作 - ディストリビューションはヘッダーを削除します。
- ヘッダー - Range

  サポート対象 - あり (デフォルト)

  設定されていない場合の動作 - ディストリビューションはヘッダーをオリジンに転送します。
- ヘッダー - Referer

  サポート - あり

  設定されていない場合の動作 - ディストリビューションはヘッダーを削除します。

- ヘッダー - Request-Range

  サポート - なし

  設定されていない場合の動作 - > ディストリビューションはヘッダーをオリジンに転送します。
- ヘッダー - TE

  サポート - なし

  設定されていない場合の動作 - ディストリビューションはヘッダーを削除します。
- ヘッダー - Trailer

  サポート - なし

  設定されていない場合の動作 - ディストリビューションはヘッダーを削除します。
- ヘッダー - Transfer-Encoding

  サポート - なし

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。
- ヘッダー - Upgrade

  サポート - なし (WebSocket 接続の場合を除く)

  設定されていない場合の動作 - WebSocket 接続が確立されていない限り、ディストリビューションはヘッダーを削除します。
- ヘッダー - User-Agent

  サポート - あり、ただし推奨されません

  設定されていない場合の動作 - ディストリビューションはこのヘッダーフィールドの値をAmazon CloudFrontで置き換えます。
- ヘッダー - Via

  サポート - あり

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。
- ヘッダー - Warning

  サポート - あり

設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。

- ヘッダー - X-Amz-Cf-Id

  サポート - なし

  設定されていない場合の動作 - ディストリビューションは、リクエストをオリジンに転送する前に、ビューワーリクエストにヘッダーを追加します。ヘッダー値には、リクエストを一意に識別する暗号化された文字列が含められます。

- ヘッダー - X-Edge-*

  サポート - なし

  設定されていない場合の動作 - あなたのディストリビューションは、すべてのX-Edge-*ヘッダー。

- ヘッダー - X-Forwarded-For

  サポート - あり

  設定されていない場合の動作-ディストリビューションはヘッダーをオリジンに転送します。

- ヘッダー - X-Forwarded-Proto

  サポート - なし

  設定されていない場合の動作 - ディストリビューションはヘッダーを削除します。

- ヘッダー - X-Real-IP

  サポート - なし

  設定されていない場合の動作 - ディストリビューションはヘッダーを削除します。

HTTP バージョン

ディストリビューションは HTTP/1.1 を使用してオリジンにリクエストを転送します。

リクエストの最大長と URL の最大長

パス、クエリ文字列 (ある場合)、ヘッダーを含め、リクエストの最大長は 20480 バイトです。

ディストリビューションはリクエストから URL を構築します。この URL の最大長は 8192 文字です。

リクエストまたは URL がこの最大制限を超えると、ディストリビューションは、リクエストエンティティが長すぎることを示す HTTP ステータスコード 413 (Request Entity Too Large) をビューワーに返してから、ビューワーへの TCP 接続を終了します。

OCSP Stapling

オブジェクトに対する HTTPS リクエストをビューワーが送信する際には、ドメインの SSL 証明書が無効になっていないことをディストリビューションまたはビューワーが認証機関 (CA) に対して確認する必要があります。OCSP Stapling を使用すると、ディストリビューションで証明書を検証して CA からの応答をキャッシュできるため、クライアントが直接 CA に対して証明書を検証する必要がなくなり、証明書の検証速度が向上します。

同一ドメイン内のオブジェクトに対する多数の HTTPS リクエストをディストリビューションが受信した場合は、OCSP Stapling によるパフォーマンス向上がさらに顕著になります。エッジロケーション内の各サーバーは、別々の検証リクエストを送信する必要があります。同一ドメインに対する多数の HTTPS リクエストを が受信するとすぐに、エッジロケーション内のすべてのサーバーが、SSL ハンドシェイクでパケットに "ステープリング" できるという CA からの応答を受信します。証明書が有効であることをビューワーが確認すると、ディストリビューションはリクエストされたオブジェクトを提供できます。エッジロケーション内でディストリビューションが十分なトラフィックを確保できない場合、新しいリクエストは、CA に対して証明書がまだ検証されていないサーバーに誘導される可能性が高くなります。この場合は、ビューワーが検証ステップを別途実行し、ディストリビューションサーバーがオブジェクトを提供します。このディストリビューションサーバーも CA に検証リクエストを送信するため、同じドメイン名が含まれるリクエストを次に受信したときには、CA からの検証応答が既に存在しているということになります。

永続的接続

ディストリビューションがオリジンからレスポンスを取得すると、その期間中に別のリクエストが届くのに備え、数秒間、接続を維持しようとします。持続的接続を維持すると、TCP 接続の再構築に必要な時間と後続のリクエストに対する別の TLS ハンドシェイクの実行に必要な時間を節約できます。

プロトコル

ディストリビューションは、HTTP または HTTPS リクエストを、Lightsail コンソールの [オリジンプロトコルポリシー] フィールドの値に基づいてオリジンサーバーに転送します。Lightsail コンソールで使用できるオプションは、[HTTP のみ] と [HTTPS のみ] です。

[HTTP のみ] または [HTTPS のみ] を指定すると、ディストリビューションは、ビューワーリクエストのプロトコルに関係なく、指定されたプロトコルのみを使用してリクエストをオリジンに転送します。

> **Important**
>
> ディストリビューションが HTTPS プロトコルを使用してリクエストをオリジンに転送し、オリジンサーバーから無効な証明書または自己署名証明書が返された場合、ディストリビューションは TCP 接続を中断します。

クエリ文字列

ディストリビューションがクエリ文字列パラメータをオリジンに転送するかどうかを設定できます。

オリジン接続のタイムアウトと試行

デフォルトでは、ディストリビューションはセカンダリオリジンへの接続を試行したり、エラーレスポンスを返したりする前に 30 秒 (それぞれ 10 秒間の試行が 3 回) 待機します。

オリジン応答タイムアウト

オリジン応答タイムアウト (オリジンの読み取りタイムアウトまたはオリジンリクエストタイムアウトとも呼ばれます) は、次の両方に適用されます。

- ディストリビューションがリクエストをオリジンに転送してからレスポンスを受け取るまでの待機時間 (秒)
- ディストリビューションがオリジンからレスポンスのパケットを受け取ってから次のパケットを受け取るまでの待機時間 (秒)

ディストリビューションの動作は、ビューワーリクエストの HTTP メソッドによって決まります。

- GET および HEAD リクエスト – 応答タイムアウトの期間内にオリジンが応答しない場合、または応答を停止した場合、ディストリビューションは接続を中断します。指定されたオリジン接続の試行回数が 1 回を超える場合、ディストリビューションは完全な応答の取得を再試行します。オリジン接続の試行回数設定の値で決められているように、ディストリビューションは最大 3 回試行します。最後の試行でもオリジンが応答しない場合、ディストリビューションは同じオリジンのコンテンツに対する別のリクエストを受け取るまで接続を試みません。

- DELETE、OPTIONS、PATCH、PUT、POST の各リクエスト – オリジンが 30 秒以内に応答しない場合、ディストリビューションは接続を中断し、オリジンへの接続を再試行しません。クライアントは、必要に応じてリクエストを再送信できます。

同じオブジェクト (トラフィックスパイク) の同時リクエスト

ディストリビューションエッジロケーションがオブジェクトのリクエストを受け取り、オブジェクトが現在キャッシュにないか、有効期限が切れている場合、ディストリビューションはすぐにオリジンにリクエストを送信します。トラフィックスパイクがある場合 (同じオブジェクトへの追加のリクエストが、オリジンが最初のリクエストに応答する前にエッジロケーションに届く場合)、ディストリビューションは短時間一時停止してから、オブジェクトへの追加のリクエストをオリジンに転送します。通常、最初のリクエストへのレスポンスは、それ以降のリクエストに対するレスポンスの前に、ディストリビューションエッジロケーションに届きます。この短い停止により、オリジンサーバーでの不要な負荷が減ります。リクエストヘッダーや Cookie に基づいてキャッシュするようにディストリビューションを設定した場合など、追加のリクエストが同じでない場合、ディストリビューションはすべての一意のリクエストをオリジンに転送します。

ユーザーエージェントヘッダー

ユーザーがコンテンツの表示に使用しているデバイスに基づいて、オブジェクトの異なるバージョンを ディストリビューションでキャッシュするには、次の 1 つ以上のヘッダーをオリジンに転送するように ディストリビューションを設定することをお勧めします。

- CloudFront-Is-Desktop-Viewer
- CloudFront-Is-Mobile-Viewer
- CloudFront-Is-SmartTV-Viewer
- CloudFront-Is-Tablet-Viewer

ディストリビューションは、User-Agent ヘッダーの値に基づいて、これらのヘッダーの値を true または false に設定した後、リクエストをオリジンに転送します。デバイスが複数のカテゴリに属する場合は、複数の値が true になることがあります。たとえば、あるタブレットデバイスについて、ディストリビューションがCloudFront-Is-Mobile-Viewer と CloudFront-Is-Tablet-Viewer の両方を true に設定する場合があります。

User-Agent ヘッダーの値に基づいてオブジェクトをキャッシュするようにディストリビューションを設定できますが、これはお勧めできません。User-Agent ヘッダーには可能な値が多数あり、

その値に基づいてキャッシュすると、ディストリビューションがオリジンに転送するリクエストの数が大幅に増加します。

ディストリビューションが User-Agent ヘッダーの値に基づいてオブジェクトをキャッシュするように設定しない場合、ディストリビューションは以下の値を指定した User-Agent ヘッダーを追加して、リクエストをオリジンに転送します。

```
User-Agent = Amazon CloudFront
```

ディストリビューションは、ビューワーからのリクエストに User-Agent ヘッダーが含まれているかどうかに関係なく、このヘッダーを追加します。ビューワーからのリクエストに User-Agent ヘッダーが含まれる場合、ディストリビューションはそのヘッダーを削除します。

ディストリビューションがオリジンからの応答を処理する仕組み

このトピックには、 オリジンからのレスポンスを ディストリビューショが処理する方法に関する情報が含まれています。

目次

- 100-continue レスポンス
- キャッシュ
- キャンセルされたリクエスト
- コンテンツネゴシエーション
- cookie
- 切断された TCP 接続
- ディストリビューションが削除または置き換える HTTP レスポンスヘッダー
- 最大ファイルサイズ
- 使用できないオリジン
- リダイレクト
- 転送エンコード

100-continue レスポンス

オリジンは複数の 100-continue レスポンスをディストリビューションに送信することはできません。最初の 100-continue レスポンスの後で、ディストリビューションは HTTP 200 OK レスポンス

を予期します。オリジンが最初のレスポンスの後に別の 100-continue レスポンスを送信すると、ディストリビューションはエラーを返します。

キャッシュ

- オリジンが Date および Last-Modified ヘッダーフィールドに有効かつ正確な値を設定していることを確認します。
- ビューワーからのリクエストに If-Match または If-None-Match リクエストヘッダーフィールドが含まれる場合、ETag レスポンスヘッダーフィールドを設定します。ETag の値が指定されていない場合、ディストリビューションは以降の If-Match または If-None-Match ヘッダーを無視します。
- 通常、ディストリビューションはオリジンからのレスポンスの Cache-Control: no-cache ヘッダーを優先します。例外については、「同じオブジェクトに対する同時要求 (トラフィックの急増) 」を参照してください。

取り消されたリクエスト

オブジェクトがエッジキャッシュになく、ディストリビューションがオブジェクトをオリジンから取得したものの、リクエストされたそのオブジェクトを配信する前にビューワーがセッションを終了すると (ブラウザを閉じるなど)、ディストリビューションはそのオブジェクトをエッジロケーションにキャッシュしません。

コンテンツネゴシエーション

オリジンが応答で Vary:* を返し、対応するキャッシュ動作の [最小 TTL] の値が [0] の場合、ディストリビューションはオブジェクトをキャッシュしますが、そのオブジェクトの後続のすべてのリクエストをオリジンに転送して、キャッシュにオブジェクトの最新バージョンが含まれていることを確認します。ディストリビューションには、If-None-Match や If-Modified-Since などの条件付きヘッダーは含まれません。その結果、オリジンはすべてのリクエストに応じてディストリビューションにオブジェクトを返します。

オリジンが応答で Vary:* を返し、対応するキャッシュ動作の [最小 TTL] の値が別の値になっている場合、CloudFront は「ディストリビューションが削除または置き換える HTTP レスポンスヘッダー」に記述されている方法で Vary ヘッダーを処理します。

cookie

キャッシュ動作の Cookie を有効にしており、オリジンが Cookie とオブジェクトを返す場合、ディストリビューションはオブジェクトと Cookie の両方をキャッシュします。これにより、オブジェクトのキャッシュ可能性が低下します。

切断された TCP 接続

オリジンがオブジェクトをディストリビューションに返している間にディストリビューションとオリジン間の TCP 接続が中断した場合、ディストリビューションの動作は、オリジンが Content-Length ヘッダーをレスポンスに含めたかどうかによって異なります。

- Content-Length ヘッダー – ディストリビューションは、オブジェクトをオリジンから取得すると、ビューワーにオブジェクトを返します。ただし、Content-Length ヘッダーの値がオブジェクトのサイズに一致しない場合、ディストリビューションはオブジェクトをキャッシュしません。
- Transfer-Encoding: Chunked – ディストリビューションは、オブジェクトをオリジンから取得すると、ビューワーにオブジェクトを返します。ただし、チャンクレスポンスが完了していない場合、ディストリビューションはオブジェクトをキャッシュしません。
- Content-Length ヘッダーなし – ディストリビューションはオブジェクトをビューワーに返して、オブジェクトをキャッシュしますが、オブジェクトが完全でない可能性があります。Content-Length ヘッダーがない場合、ディストリビューションは、TCP 接続が誤って中断されたか、または故意に中断されたかを判断できません。

Content-Length ヘッダーを追加して、ディストリビューションが不完全なオブジェクトをキャッシュしないように HTTP サーバーを設定することをお勧めします。

ディストリビューションが削除または置き換える HTTP レスポンスヘッダー

ディストリビューションは、 オリジンからのレスポンスをビューワーに転送する前に、以下のヘッダーフィールドを削除または更新します。

- Set-Cookie - Cookie を転送するようにディストリビューションを設定している場合、Set-Cookie ヘッダーフィールドがクライアントに転送されます。
- Trailer
- Transfer-Encoding - オリジンがこのヘッダーフィールドを返す場合、ディストリビューションは値を chunked に設定してからビューワーにレスポンスを返します。
- Upgrade

- Vary – 次の点に注意してください。
  - デバイス固有のヘッダーのいずれかをオリジン (CloudFront-Is-Desktop-Viewer、CloudFront-Is-Mobile-Viewer、CloudFront-Is-SmartTV-Viewer、CloudFront-Is-Tablet-Viewer) に転送するようにディストリビューションを設定しており、オリジンが Vary:User-Agent をディストリビューションに返すように設定している場合、ディストリビューションは Vary:User-Agent をビューワーに返します。
  - Varyヘッダーに、Accept-Encoding または Cookie のいずれかを含めるよう設定した場合、ディストリビューションはビューワーへの応答にその値を含めます。
  - オリジンにヘッダーの許可リストを転送するように Vary ヘッダーを設定し、(たとえば Vary:Accept-Charset,Accept-Language など) ヘッダー名を返すようにオリジンを設定した場合、ディストリビューションはこれらの値を持つ Vary ヘッダーをビューワーに返します。
  - ディストリビューションが、* の Vary ヘッダーの値を処理するかについて詳しくは「コンテンツネゴシエーション」を参照してください。
  - Vary ヘッダーで他の値を返すようにオリジンを設定している場合、ディストリビューションは応答をビューワーに返す前にその値を削除します。
- Via – ディストリビューションは、ビューワーへの応答で値を次のように設定します。

  Via: *http-version alphanumeric-string*.cloudfront.net (CloudFront)

  たとえば、クライアントが HTTP/1.1 を介してリクエストを行った場合、値は次のようになります。

  Via: 1.1 1026589cc7887e7a0dc7827b4example.cloudfront.net (CloudFront)

最大ファイルサイズ

ディストリビューションがビューワーに返すレスポンス本文の最大サイズは 20 GB です。これには、Content-Length ヘッダーの値を指定しないチャンク転送レスポンスが含まれます。

使用できないオリジン

オリジンサーバーが使用できないときに、ディストリビューションがエッジキャッシュに存在するオブジェクトのリクエストを受け取り、そのオブジェクトが (たとえば Cache-Control max-age ディレクティブに指定された期間が経過しているために) 有効期限切れになっている場合、ディストリビューションは有効期限切れバージョンのオブジェクトを供給するか、またはカスタムエラーページを供給します。

場合によって、要求頻度の低いオブジェクトは削除されてエッジキャッシュで使用できなくなることがあります。ディストリビューションは、削除されたオブジェクトを提供することはできません。

リダイレクト

オリジンサーバーでオブジェクトの場所を変更した場合、リクエストを新しい場所にリダイレクトするようにウェブサーバーを構成できます。リダイレクトが構成された後、ビューワーがオブジェクトのリクエストを最初に送信したときに、ディストリビューションはリクエストをオリジンに送信し、オリジンはリダイレクトで応答します (例: `302 Moved Temporarily`)。ディストリビューションはリダイレクトをキャッシュし、ビューワーにリダイレクトを返します。ディストリビューションはリダイレクトに従いません。

リクエストを以下のどちらかの場所にリダイレクトするようにウェブサーバーを構成できます。

- オリジンサーバーのオブジェクトの新しい URL。ビューワーが新しい URL へのリダイレクトに従う場合、ビューワーはディストリビューションをバイパスし、オリジンに直接アクセスします。つまり、オリジンにあるオブジェクトの新しい URL にリクエストをリダイレクトしないことをお勧めします。
- オブジェクトの新しいディストリビューション URL。新しいディストリビューション URL を含むリクエストがビューワーから送信されると、ディストリビューションは、オリジンの新しい場所からオブジェクトを取得し、エッジロケーションにキャッシュした後、ビューワーにオブジェクトを返します。オブジェクトに対する以降のリクエストはエッジロケーションによって処理されます。これにより、オリジンのオブジェクトを要求するビューワーに関連するレイテンシーと負荷が回避されます。ただし、オブジェクトに対する新しいすべてのリクエストに、ディストリビューションへの 2 つのリクエストに対する料金がかかります。

転送エンコード

Lightsail ディストリビューションは、`Transfer-Encoding` ヘッダーの `chunked` 値のみをサポートします。オリジンが `Transfer-Encoding: chunked` を返した場合、ディストリビューションは、エッジロケーションで受け取ったオブジェクトをクライアントに返し、そのオブジェクトをチャンク形式でキャッシュして以降のリクエストに備えます。

ビューワーが `Range GET` をリクエストし、オリジンは `Transfer-Encoding: chunked` を返した場合、ディストリビューションはリクエストされた範囲ではなくオブジェクト全体をビューワーに返します。

レスポンスのコンテンツ長を事前に決定できない場合は、チャンクエンコーディングを使用することをお勧めします。詳細については、「中断された TCP 接続」を参照してください。

Lightsail ディストリビューションのコンテンツキャッシュを検証する

このガイドでは、Amazon Lightsail ディストリビューションが配信元からコンテンツをキャッシュして提供していることをテストする方法について説明します。このテストは、登録したドメイン名をディストリビューションに追加した後に行う必要があります。ディストリビューションの詳細については、「コンテンツ配信ネットワークディストリビューション」を参照してください。

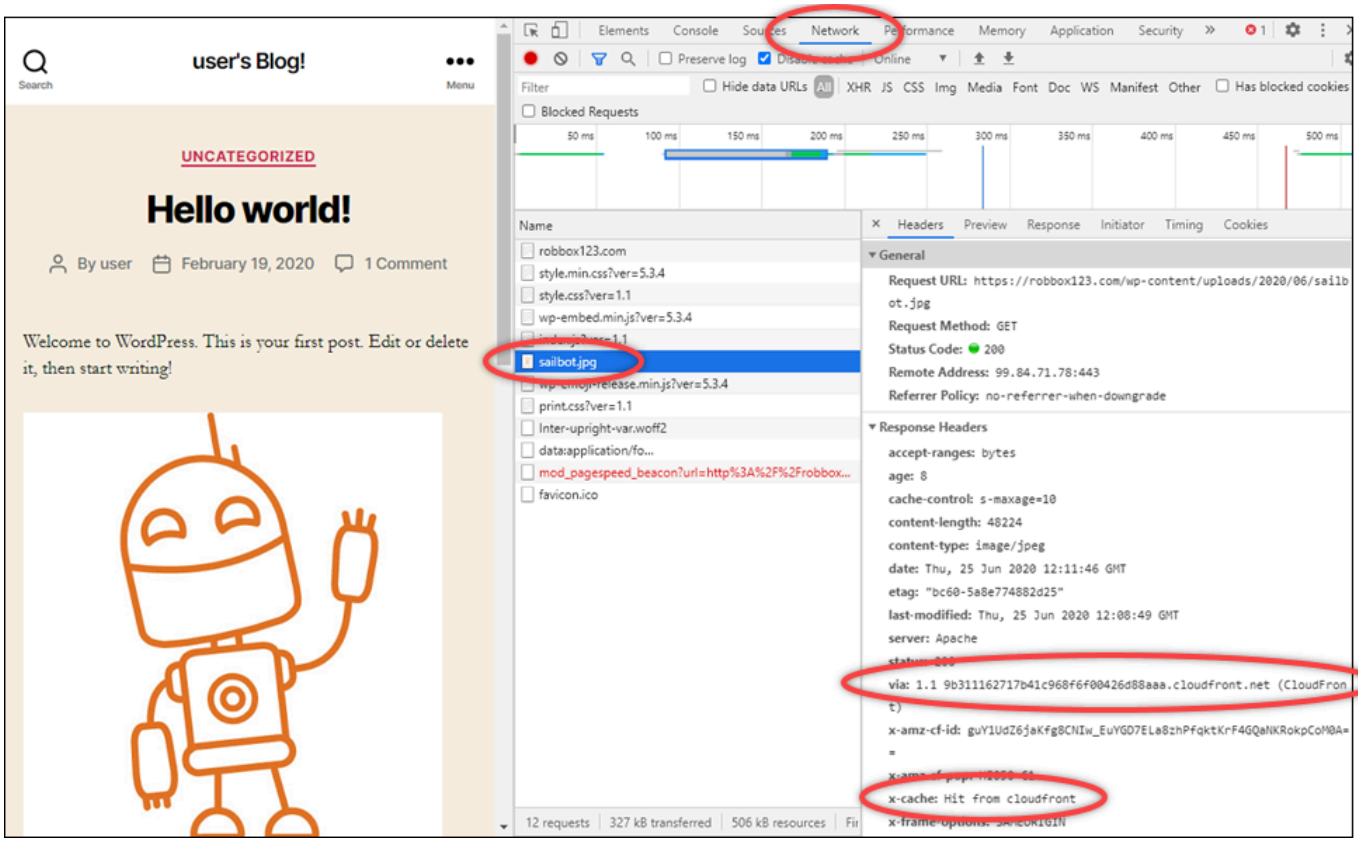
ディストリビューションをテスト

ディストリビューションをテストするには、以下の手順を行います。この手順では Chrome ウェブブラウザを使用していますが、他のブラウザでも同様の手順を使用することができます。

1. Chrome ウェブブラウザを開きます。
2. ブラウザウィンドウの右上にある Chrome メニュー を開き、[その他のツール]の [デベロッパーツール]を選択します。

   ショートカット Option + ⌘ + J (macOS の場合) 、Shift + Ctrl + J (Windows/Linux の場合) を使用することもできます。
3. デベロッパー ツールのペインで、[ネットワーク]タブを選択します。
4. ディストリビューションのドメインを参照します (例 : `https://www.example.com`)。

   Chrome のデベロッパーツールの[ネットワーク]タブには、ウェブサイトのオブジェクトのリストが表示されます。
5. イメージファイル (.jpg, .png, .gif) などの静的オブジェクトを選択します。
6. 表示される[ヘッダー] パネルで、`via` と `x-cache` のヘッダーの両方に CloudFront が記載されていることがわかります。これにより、ディストリビューションがオリジンからコンテンツをキャッシュして提供していることが確認されます。


user's Blog!
Search
Menu
UNCATEGORIZED
Hello world!
By user
February 19, 2020
1 Comment
Welcome to WordPress. This is your first post. Edit or delete it, then start writing!
Elements
Console
Network
Performance
Memory
Application
Security
Preserve log
Online
Filter
Hide data URLs
XHR JS CSS Img Media Font Doc WS Manifest Other
Has blocked cookies
Blocked Requests
Name
robbox123.com
style.min.css?ver=5.3.4
style.css?ver=1.1
wp-embed.min.js?ver=5.3.4
sailbot.jpg
print.css?ver=1.1
Inter-upright-var.woff2
data:application/fo...
favicon.ico
Headers
Preview
Response
Initiator
Timing
Cookies
General
Request URL: https://robbox123.com/wp-content/uploads/2020/06/sailbot.jpg
Request Method: GET
Status Code: 200
Remote Address: 99.84.71.78:443
Referrer Policy: no-referrer-when-downgrade
Response Headers
accept-ranges: bytes
age: 8
cache-control: s-maxage=10
content-length: 48224
content-type: image/jpeg
date: Thu, 25 Jun 2020 12:11:46 GMT
etag: "bc60-5a8e774882d25"
last-modified: Thu, 25 Jun 2020 12:08:49 GMT
server: Apache
via: 1.1 9b311162717b41c968f6f00426d88aaa.cloudfront.net (CloudFront)
x-amz-cf-id: guY1UdZ6jaKfg8CNIw_EuYGD7ELa8zhPfqktKrF4GQaNKRokpCoM0A==
x-cache: Hit from cloudfront
12 requests
327 kB transferred
506 kB resources

Amazon Lightsail のネットワーキングリソース

Lightsail ネットワーキングリソースは、ユーザーや外部サービスが Lightsail インスタンスに接続する方法を改善します。

ロードバランサー

ロードバランサーを作成すると、冗長性を追加したり、処理するトラフィック量を増やしたりできます。詳細については、「ロードバランサー」を参照してください。

静的 IP

静的 IP アドレスを作成すると、インスタンスを再起動しても同じ IP アドレスを保持できます。詳細については、「静的 IP アドレス」を参照してください。

Lightsail リソースの IP アドレスの表示と管理

IP アドレスを使用して、Lightsail インスタンス、およびその他の Lightsail リソースでコミュニケーションをとることができます。たとえば、インスタンスのパブリック IP アドレスを使用して、インスタンスのネットワークステータスを確認し (PING を使用)、インスタンスへの SSH 接続を確立し、カスタムドメイン名からインスタンスにトラフィックをルーティングできます。Lightsail リソースの IP アドレスで多くのことができます。

Lightsail インスタンス、コンテナサービス、ロードバランサーは、IPv4 と IPv6 の両方のアドレス設定プロトコルをサポートします。こういったリソースでは、デフォルトで IPv4 アドレス設定プロトコルを使用します。この動作を無効にすることはできません。必要に応じて、インスタンス、コンテナサービス、ロードバランサーに対して IPv6 を有効にできます。

このガイドでは、Lightsail のIP アドレスに関して理解する必要がある事項について説明します。

目次

- インスタンスのプライベートとパブリック IPv4 アドレス
- インスタンスの静的 IP アドレス
- インスタンス、コンテナーサービス、CDN ディストリビューション、およびロードバランサー用の IPv6

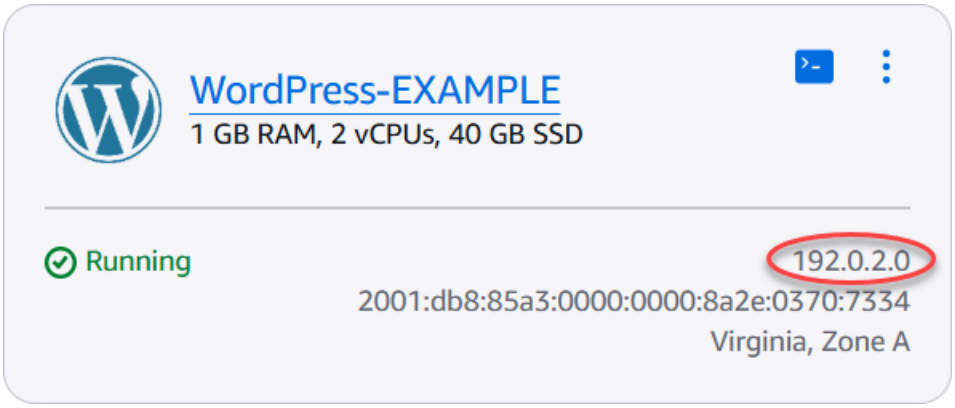
インスタンスのプライベートとパブリックの IPv4 アドレス

Lightsail インスタンスを作成する際、パブリックとプライベート IPv4 アドレスが割り当てられます。パブリック IP アドレスはインターネットへアクセスできますが、プライベート IP アドレスは同じ AWS リージョンの Lightsail アカウントのリソースにのみアクセス可能です。

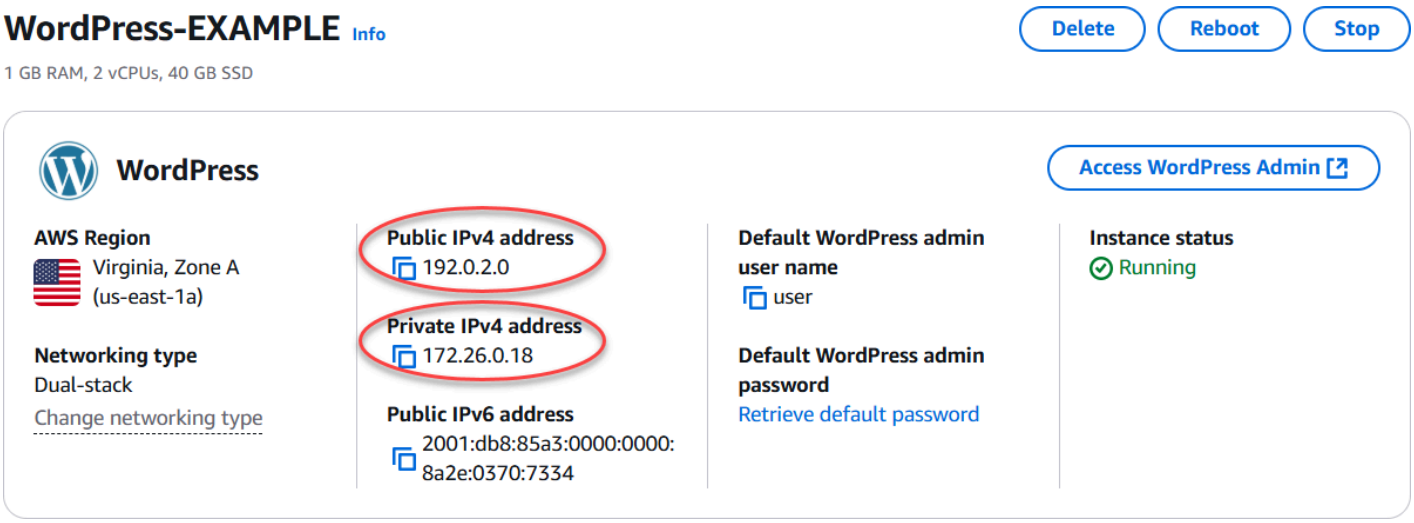
> **Note**
>
> インスタンスのプライベート IP アドレスは、同じ AWS リージョン内の他の AWS リソースにアクセスできますが、VPC ピアリングを有効にする場合は Lightsail アカウント外に限ります。詳細については、「Lightsail の外で AWS リソースを使用するために Amazon VPC ピア接続をセットアップする」を参照してください。

インスタンスの IP アドレスは、Lightsail コンソールの次のエリアに表示されています。

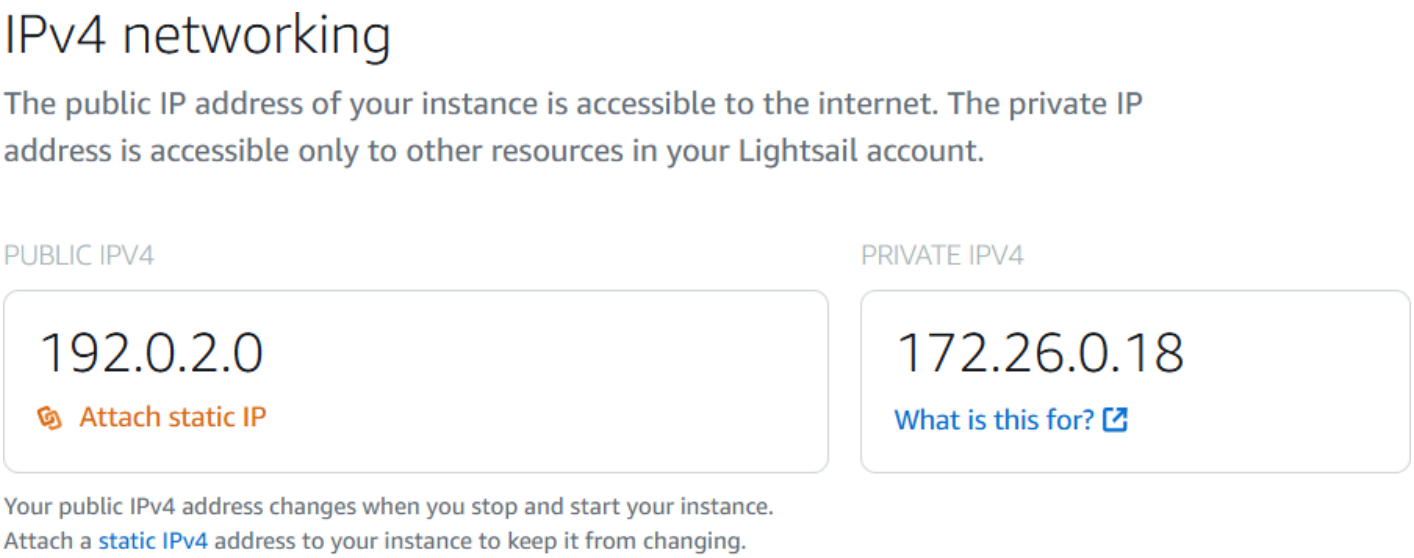
- 以下の例は、Lightsail ホームページ上のインスタンスのパブリック IPv4 アドレスを示しています。



- 次の例は、インスタンス管理ページのヘッダー領域にあるインスタンスのパブリックとプライベート IPv4 アドレスを示しています。



- 次の例は、インスタンス管理ページの [ネットワーク] タブにあるインスタンスのパブリック IP アドレスとプライベート IPv4 アドレスを示しています。



インスタンスの IPv4 アドレスを使用する場合は、以下の点に注意してください。

- インスタンスのパブリック IP アドレスは変わることがあります。インスタンスに静的 IP を添付することで、変更されない IP アドレスを割り当てます。詳細については、このガイドのインスタンスの静的 IP アドレスセクションを参照してください。
- Lightsail では、デフォルトで IPv4 アドレスを使用します。ただし、2021 年 1 月 12 日以前に作成されたいくつかの Lightsail リソースについては、オプションで IPv6 を有効にできます。2021 年 1 月 12 日以降に作成されたリソースでは、IPv6 がデフォルトで有効になっています。詳細については、このガイドのインスタンス、コンテナーサービス、CDN ディストリビューション、およびロードバランサー用の IPv6 セクションを参照してください。

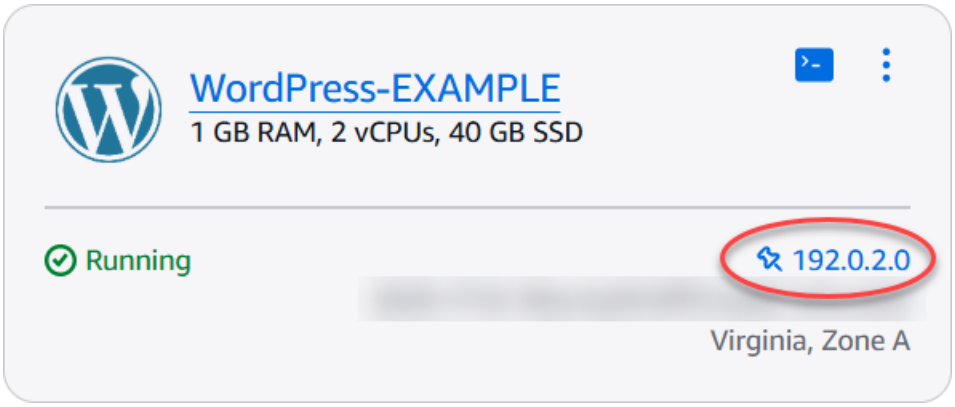
- インスタンスのファイアウォールにルールを追加して、インスタンスに接続できるトラフィックを制御できます。詳細については、「インスタンスのファイアウォール」を参照してください。

インスタンスの静的 IPv4 アドレス

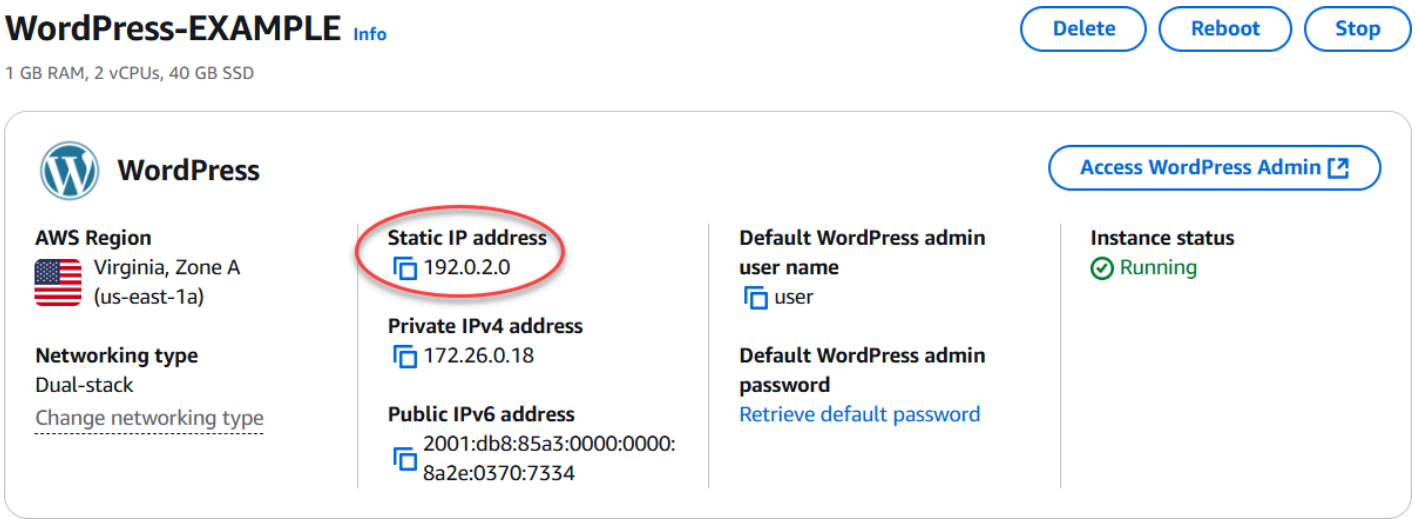
インスタンスの作成時にインスタンスに割り当てられるデフォルトの公開 IPv4 アドレスは、インスタンスを停止してまた開始すると変更されます。必要に応じて、静的 IPv4 アドレスをインスタンスに作成し、アタッチできます。静的 IPv4 アドレスは、インスタンスのデフォルトの公開 IPv4 アドレスに置き換えられ、インスタンスを停止して起動しても変わりません。1 つの静的 IP を 1 つのインスタンスにアタッチできます。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。

静的 IP を作成してインスタンスにアタッチすると、その静的 IP はLightsail コンソールの次のエリアに表示されます。

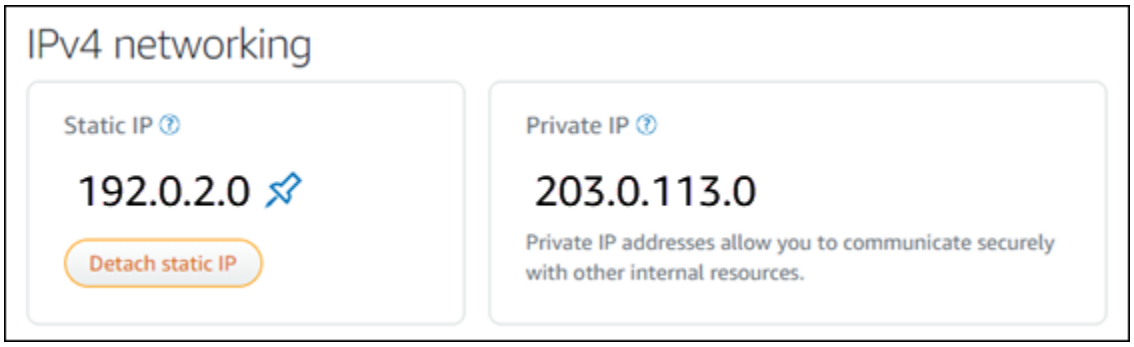
- 以下の例は、Lightsail ホームページ上のインスタンスのパブリック IP アドレスを示しています。画鋲アイコンは、パブリック IP アドレスが静的であることを示します。



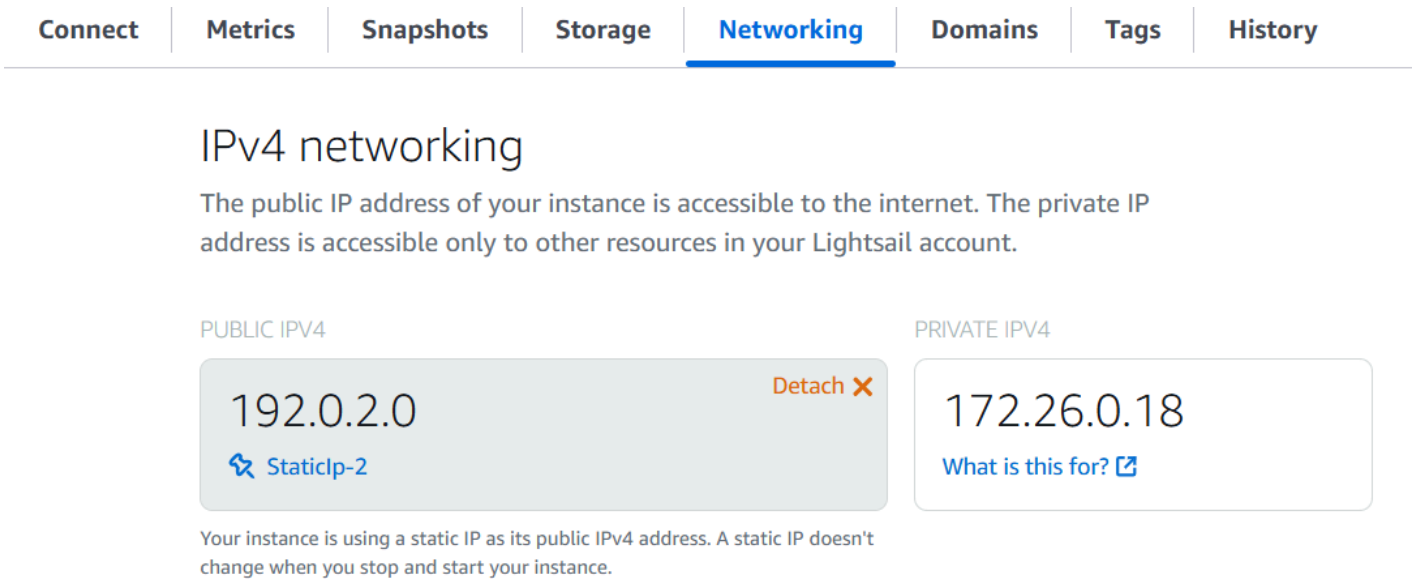
- 次の例は、インスタンス管理ページのヘッダー領域にあるインスタンスの静的 IP アドレスを示しています。画鋲アイコンは、パブリック IP アドレスが静的であることを示します。



- 次の例は、インスタンス管理ページの [ネットワーク] タブにあるインスタンスの静的 IP アドレスを示しています。デフォルトのパブリック IP アドレスは表示されなくなり、静的 IP アドレスに置き換えられました。画鋲アイコンは、パブリック IP アドレスが静的であることを示します。



- 作成したすべての静的 IP は、次の例のように Lightsail ホームページの [ネットワーク] タブで確認できます。



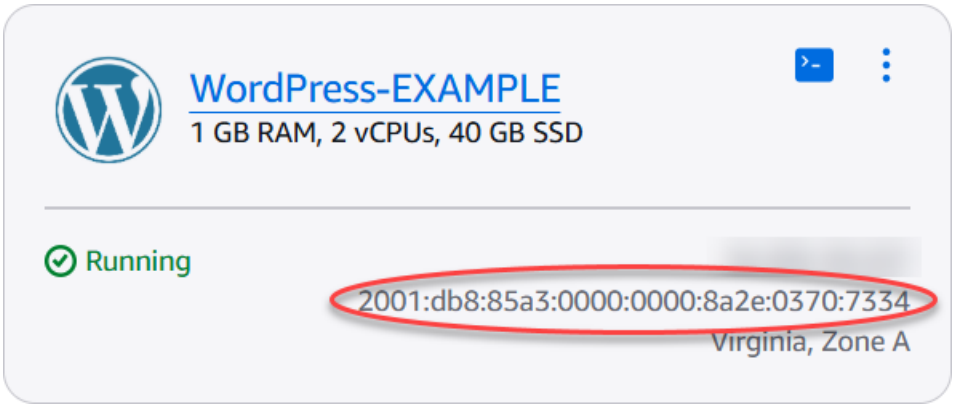
インスタンス、コンテナーサービス、CDN ディストリビューション、およびロードバランサー用の IPv6

IPv6 は、2021 年 1 月 12 日以降に作成された Lightsail インスタンス、コンテナーサービス、CDN ディストリビューション、およびロードバランサーでデフォルトで有効になっています。オプションで、2021 年 1 月 12 日より前に作成されたリソースの IPv6 を有効にできます。特定のリソースに対して IPv6 を有効にすると、Lightsail は自動的に IPv6 アドレスをそのリソースに割り当てます。IPv6 アドレスを自分で選択または指定することはできません。詳細については、「IPv6 を有効化または無効化する」を参照してください。

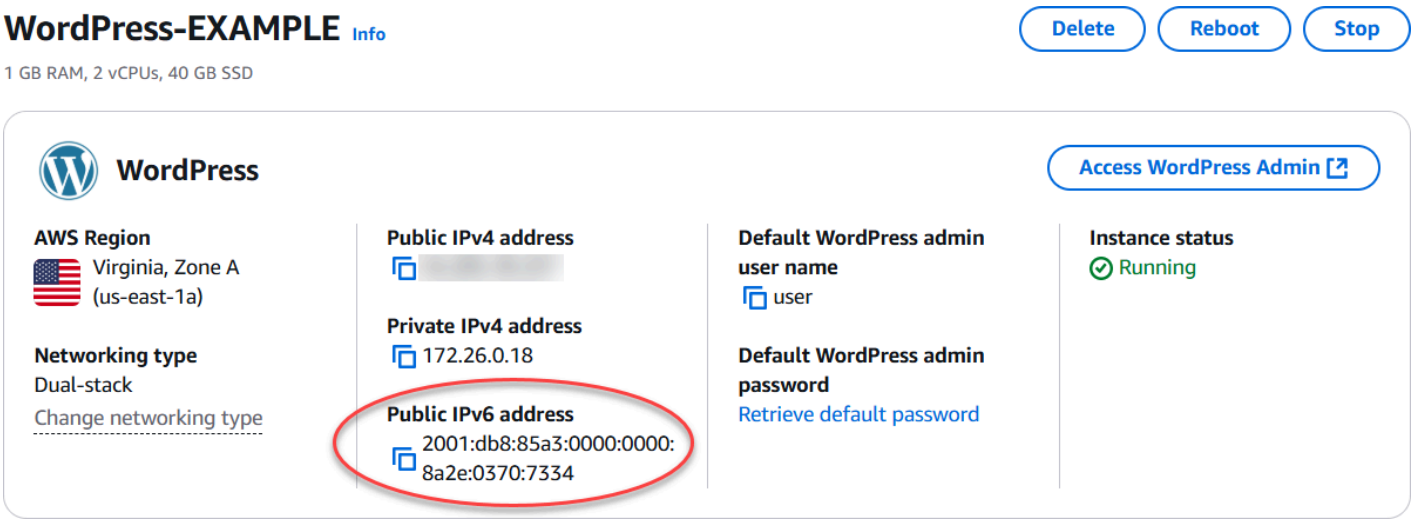
IPv6 専用インスタンスを作成することもできます。IPv6 専用インスタンスは IPv6 経由でのみパブリックに通信でき、パブリック IPv4 アドレスはありません。詳細については、Lightsail インスタンスの IPv6 専用ネットワークを設定するを参照してください。

インスタンスの IPv6 アドレスは、Lightsail コンソールの次のエリアに表示されています:

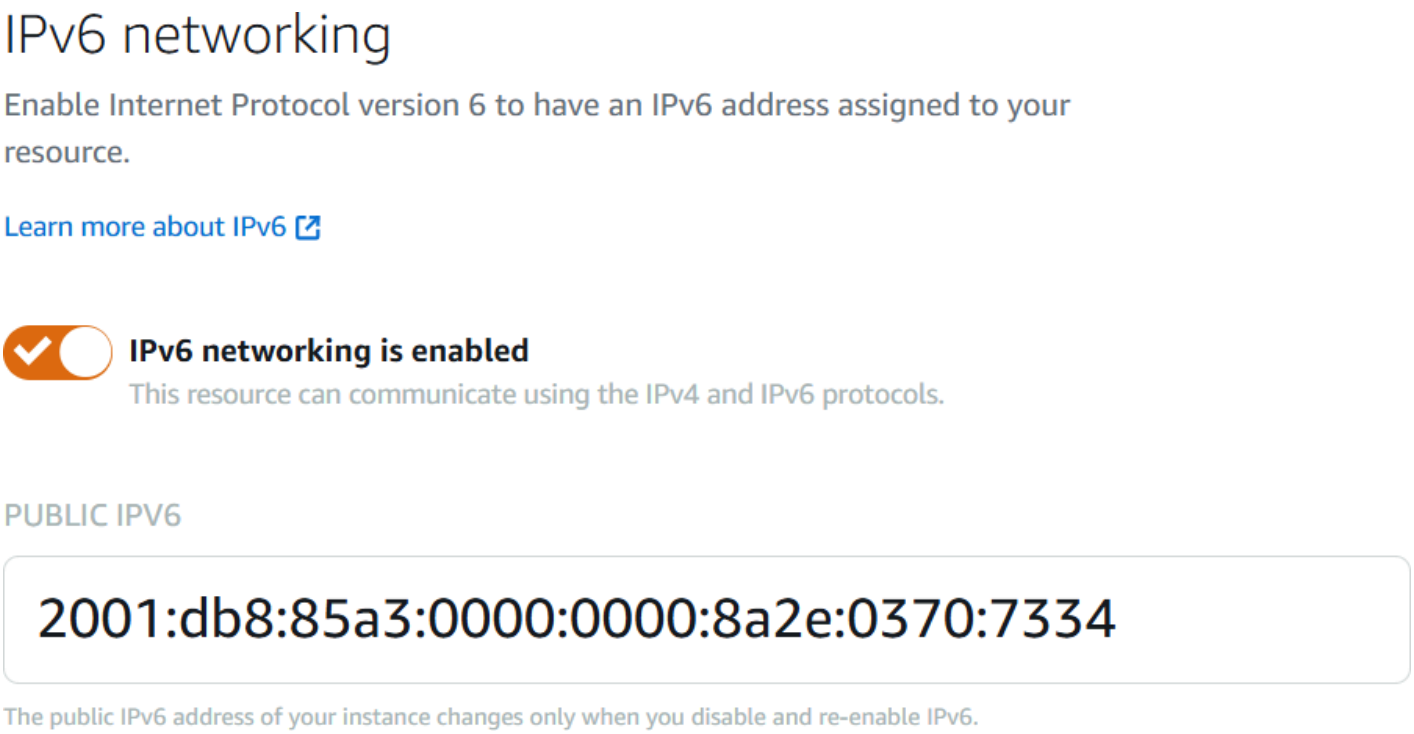
- 次の例は、Lightsail ホームページ上のインスタンスの IPv6 アドレスを示しています。



- 次の例は、リソースの管理ページのヘッダー領域にあるリソースの IPv6 アドレスを示しています。



- 次の例は、リソース管理ページの [ネットワーク] タブにあるリソースの IPv6 アドレスを示しています。



リソースで IPv6 を有効にして使用するときは、次の点に注意してください。

- リソースに対して IPv6 を有効にすると、リソースは IPv4 のみ、または IPv4 と IPv6 (デュアルスタックモード) で通信できます。

- リソースで IPv6 を有効にすると、Lightsail は自動的に IPv6 アドレスをそのリソースに割り当てます。IPv6 アドレスを自分で選択または指定することはできません。リソースに対して IPv6 を有効にすると、IPv6 プロトコル経由のネットワークトラフィックの受け入れが開始されます。
- インスタンスの IPv6 アドレスは、インスタンスを停止してまた開始しても保持されます。インスタンスを削除するか、インスタンスに対して IPv6 を無効にした場合にのみ解除されます。これらのアクションのいずれかを実行した後は、同じ IPv6 アドレスを取得することはできません。
- インスタンスに割り当てられるすべての IPv6 アドレスは公開されているため、インターネット経由で接続することができます。インスタンスに割り当てられるプライベート IPv6 アドレスは存在しません。
- インスタンスの IPv4 アドレスと IPv6 アドレスは互いに独立しています。従って、 IPv4 と IPv6 のインスタンスファイアウォールのルールは個別に設定する必要があります。詳細については、「インスタンスのファイアウォール」を参照してください。
- Lightsail のすべてのインスタンスの設計図は、IPv6 が有効なときに IPv6 用に自動的に構成されます。次の設計図を使用するインスタンスでは、IPv6 を有効にした後で、追加の設定手順が必要になります。
  - cPanel — 詳細については、「cPanel インスタンスに IPv6 を設定する」を参照してください。
  - GitLab — 詳細については、「GitLab インスタンスに IPv6 を設定する」を参照してください。
  - Nginx — 詳細については、「Nginx インスタンスに IPv6 を設定する」を参照してください。
  - Plesk — 詳細については、「Plesk インスタンスに IPv6 を設定する」を参照してください。

> **Note**
>
> PrestaShop は現在、IPv6 アドレスをサポートしていません。インスタンスに対して IPv6 を有効にすることはできますが、PrestaShop ソフトウェアは IPv6 ネットワーク経由のリクエストに応答しません。

Lightsail の静的 IP アドレス

静的 IP アドレスは固定のパブリック IP アドレスであり、インスタンスまたはその他リソースに割り当ておよび再割り当てできます。静的 IP アドレスを未設定の場合、インスタンスを停止または再起動するたびに、Lightsail で新しいパブリック IP アドレスが割り当てられます。

Lightsail インスタンスにアタッチされた静的 IP アドレスにはコストがかかりません。ただし、静的 IP アドレスがインスタンスにアタッチされていない場合は料金が発生します。詳細については、「Lightsail 静的 IPv4 アドレスの費用を教えてください。」を参照してください。

> **Important**
>
> 最初に静的 IP アドレスを作成してインスタンスに添付せずにインスタンスを停止または再起動すると、インスタンスの再起動時に IP アドレスが失われます。インスタンスが常に同じパブリック IP アドレスを持つように、静的 IP アドレスを作成してインスタンスにアタッチする必要があります。詳細については、「静的 IP アドレスを作成する」を参照してください。

内容

- 静的 IP を作成して Lightsail インスタンスにアタッチします。
- Lightsail での静的 IP アドレスの削除

静的 IP を作成して Lightsail インスタンスにアタッチします。

Amazon Lightsail にアタッチされているデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して再起動するたびに変わります。パブリック IP アドレスが変わらないようにするには、静的 IP アドレスを作成してインスタンスに接続します。その後、登録したドメイン名がインスタンスをポイントするように設定すると、インスタンスを停止して再起動するたびに、ドメインの DNS レコードを更新する必要がなくなります。1 つの静的 IP を 1 つのインスタンスにアタッチできます。詳細については、「静的 IP アドレス」を参照してください。

前提条件

Lightsail で実行されている少なくとも 1 つのデュアルスタックインスタンスが必要です。インスタンスを作成するには、「インスタンスを作成する」を参照してください。

静的 IP アドレスの作成およびインスタンスへの割り当て

以下の手順で、新しい静的 IP アドレスを作成し、Lightsail のインスタンスにアタッチできます。

1. https://lightsail.aws.amazon.com/ で Lightsailコンソールにサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. [静的 IP の作成] を選択します。
4. 静的 IP AWS リージョン を作成する を選択します。

> **Note**
>
> 静的 IP アドレスは、同じリージョンのインスタンスにのみアタッチできます。

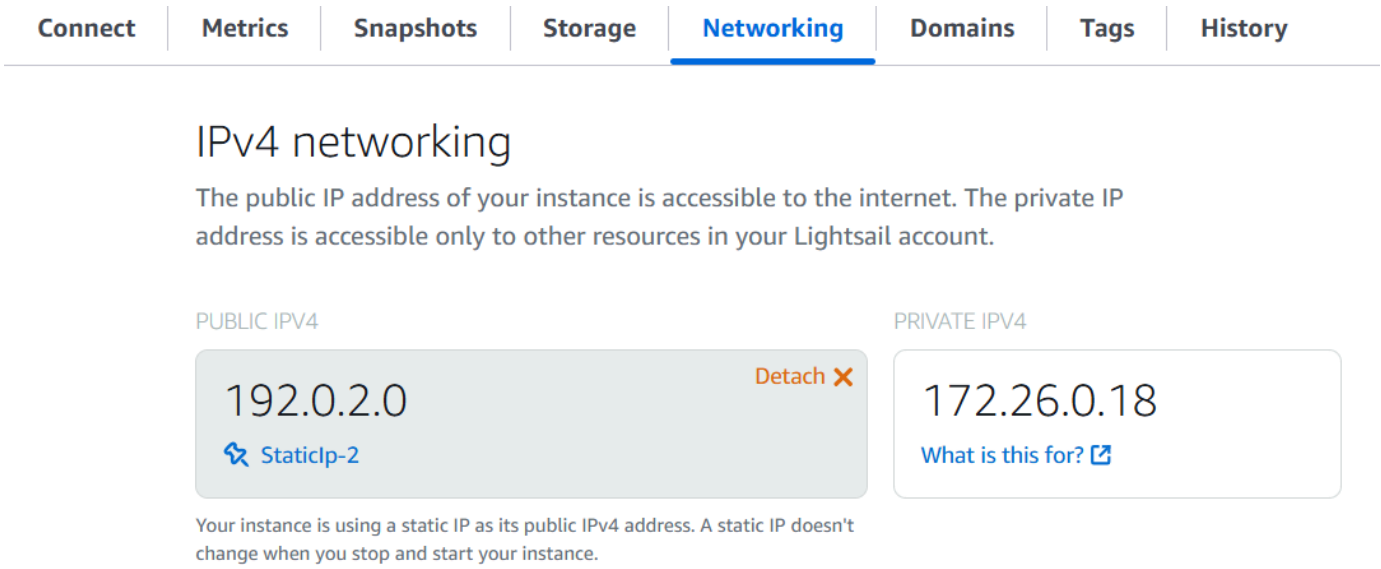
5. 静的 IP をアタッチする Lightsail リソースを選択します。
6. 静的 IP の名前を入力します。

   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。
7. [作成] を選択します。

   ホームページに移動すると、管理できる静的 IP アドレスを確認できます。

また、インスタンス管理ページでは、[ネットワーキング] タブのパブリック IP アドレスの横に青い押しピンが表示されます。これは、現在の IP アドレスが静的であることを示します。



詳細については、「パブリック IP アドレスとプライベート IP アドレス」を参照してください。

Lightsail での静的 IP アドレスの削除

Amazon Lightsail アカウント AWS リージョン では、 ごとに最大 5 IPs を作成できます。静的 IP アドレスが割り当てられているインスタンスを削除しても、静的 IP アドレスはアカウントに残ります。静的 IP アドレスが不要になった場合は、 Lightsail コンソールまたは AWS Command Line Interface () を使用して削除できますAWS CLI。このガイドでは静的 IP アドレスを Lightsail アカウントから削除する方法について説明します。静的 IP の詳細については、「IP アドレス」を参照してください。

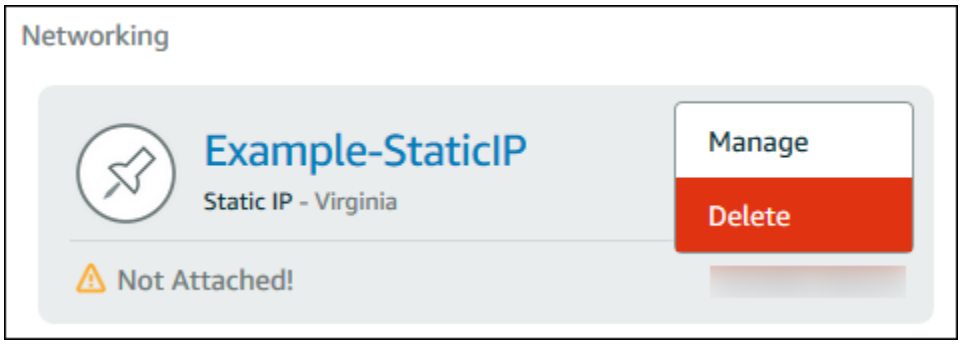
> **Important**
>
> 静的 IP を削除すると、その静的 IP が Lightsail から完全に削除されます。インスタンスなど、その静的 IP を使用するリソースは影響を受けます。静的 IP を削除すると、元に戻すことはできません。

Lightsail コンソールを使用した静的 IP の削除

Lightsail コンソールで静的 IP を削除するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。

3. [ネットワーク] ページで、削除したい静的 IP アドレスの横にある縦の省略記号アイコンを選択し、[削除] を選択します。



を使用して静的 IP を削除する AWS CLI

AWS CLIを使用して静的 IP を削除するには、以下の手順を完了してください。Lightsail アカウントから静的 IP を削除するコマンドは release-static-ip です。静的 IP を作成する場合、実際は、静的 IP を割り当てています。したがって、実際には静的 IP を削除するのではなく、解放することになります。

前提条件

まず、まだインストールしていない場合は、 をインストールする必要があります AWS CLI。詳細については、「AWS Command Line Interfaceのインストール」を参照してください。AWS CLIを設定する必要があります。

解放する静的 IP の名前が必要になります。これを取得するには、 `get-static-ips` AWS CLI コマンドを使用します。

1. 次のコマンドを入力します。

```
aws lightsail get-static-ips
```

次のような出力が表示されます。

```
{
    "staticIps": [
        {
            "name": "Example-StaticIP",
            "resourceType": "StaticIp",
            "attachedTo": "MyInstance",
```

```
            "arn": "arn:aws:lightsail:us-east-2:123456789101:StaticIp/5282f35e-
c720-4e5a-1234-12345EXAMPLE",
            "isAttached": true,
            "ipAddress": "192.0.2.0",
            "createdAt": 1489750629.026,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-east-2"
            }
        },
        {
            "name": "my-other-static-ip",
            "resourceType": "StaticIp",
            "arn": "arn:aws:lightsail:us-east-2:123456789101:StaticIp/
f5885e14-8984-49e5-1234-12345EXAMPLE",
            "isAttached": false,
            "ipAddress": "192.0.2.2",
            "createdAt": 1483653597.815,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-east-2"
            }
        }
    ]
}
```

2. 解放する静的 IP の [名前] フィールドの値を選択し、次のステップで使用できるようにその名前をメモします。

   たとえば、その値をクリップボードにコピーできます。

3. 次のコマンドを入力します。

```
aws lightsail release-static-ip --static-ip-name StaticIpName
```

   コマンドで、*StaticIpName* をスタティック IP の名前に置き換えます。

   成功すると、次のような出力が表示されます。

```
{
    "operations": [
        {
            "status": "Succeeded",
```

```
            "resourceType": "StaticIp",
            "isTerminal": true,
            "statusChangedAt": 1489860944.19,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-east-2"
            },
            "operationType": "ReleaseStaticIp",
            "resourceName": "Example-StaticIP",
            "id": "92a2f0d2-eef2-4e6f-1234-12345EXAMPLE",
            "createdAt": 1489860944.19
        }
    ]
}
```

Lightsail リソースのデュアルスタックネットワークを有効または無効にする

IPv6 は、2021 年 1 月 12 日以降に作成された Lightsail インスタンス、コンテナサービス、およびロードバランサーに対してデフォルトで有効になっています。オプションとして、2021 年 1 月 12 日より前に作成されたリソースの IPv6 を有効にすることもできます。このガイドでは、デュアルスタックインスタンスの IPv6 ネットワークを有効または無効にする方法を示します。IPv6 アドレスの詳細については、「IP アドレス」を参照してください。

デュアルスタックに関する考慮事項

IPv6 は 2021 年 1 月 12 日Lightsailに実装されました。そのため、次のガイドラインに従って、一部のリソースに対して IPv6 を手動で有効または無効にする必要がある場合があります。

- 1 月 12 日より前に作成されたインスタンス、およびロードバランサーは、有効にするまで IPv6 は無効になっています。ただし、1 月 12 日より後に作成されたインスタンス、およびロードバランサーは作成時に IPv6 が有効になります。
- 1 月 12 日より前または後に作成されたコンテナサービスでは IPv6 が有効になっています。
- IPv6 は、インスタンス、およびロードバランサーに対していつでも手動で有効または無効にできます。コンテナサービスに対して無効にすることはできません。

IPv6 を有効にして使用する場合、以下の点に注意してください。

- リソースに対して IPv6 を有効にすると、リソースは IPv4 のみ、または IPv4 と IPv6 (デュアルスタックモード) で通信できます。
- インスタンスに対して IPv6 を有効にすると、Lightsail はそのインスタンスに IPv6 アドレスを自動的に割り当てます。IPv6 アドレスを自分で選択したり指定することはできません。コンテナサービス、またはロードバランサーに対して IPv6 を有効にすると、そのリソースは IPv6 経由でインターネットトラフィックを受け入れ始めます。
- インスタンスの IPv6 アドレスは、インスタンスを停止してまた開始しても保持されます。インスタンスを削除するか、インスタンスに対して IPv6 を無効にした場合にのみ解除されます。これらのアクションのいずれかを実行した後は、同じ IPv6 アドレスを取得することはできません。
- インスタンスに割り当てられるすべての IPv6 アドレスは公開されているため、インターネット経由で接続することができます。インスタンスに割り当てられるプライベート IPv6 アドレスは存在しません。
- インスタンスの IPv4 アドレスと IPv6 アドレスは互いに独立しています。従って、 IPv4 と IPv6 のインスタンスファイアウォールのルールは個別に設定する必要があります。詳細については、「インスタンスのファイアウォール」を参照してください。
- Lightsail のすべてのインスタンスの設計図は、IPv6 が有効なときに IPv6 用に自動的に構成されます。次の設計図を使用するインスタンスでは、IPv6 を有効にした後で、追加の設定手順が必要になります。
  - cPanel — 詳細については、「cPanel インスタンスに IPv6 を設定する」を参照してください。
  - GitLab — 詳細については、「GitLab インスタンスに IPv6 を設定する」を参照してください。
  - Nginx — 詳細については、「Nginx インスタンスに IPv6 を設定する」を参照してください。
  - Plesk — 詳細については、「Plesk インスタンスに IPv6 を設定する」を参照してください。

トピック

- Lightsail リソースの IPv6 ネットワークを有効にする
- Lightsail リソースの IPv6 ネットワークを無効にする

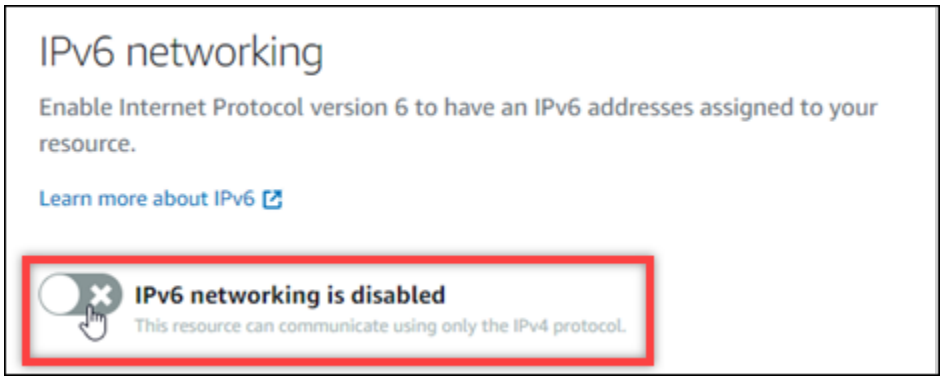
Lightsail リソースの IPv6 ネットワークを有効にする

インスタンス、CDN ディストリビューション、およびロードバランサーに対して IPv6 を有効にするには、以下の手順を実行してください。

1. Lightsail コンソール にサインインします。
2. IPv6 を有効にするリソースに応じて、次のいずれかの手順を実行してください。

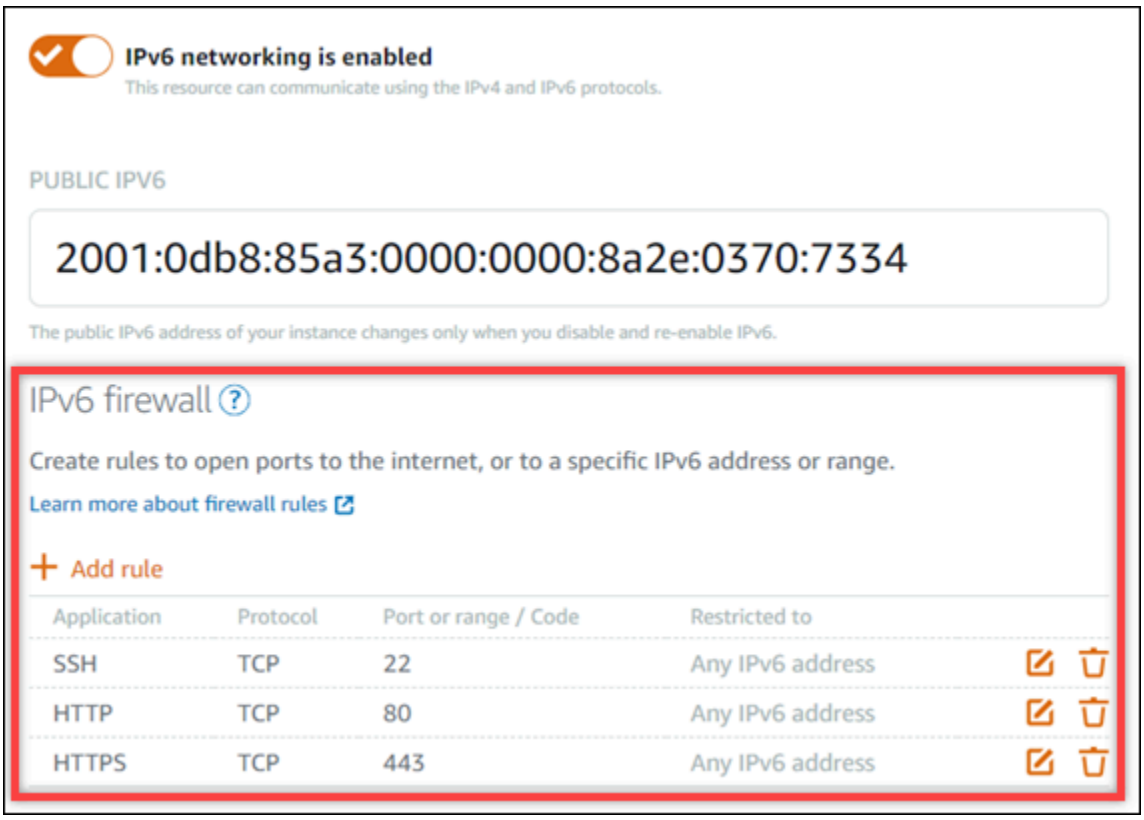
- インスタンスで IPv6 を有効にするには、Lightsail ホームページの [インスタンス] タブをクリックし、IPv6 を有効にするインスタンスの名前を選択します。
- CDN ディストリビューションまたはロードバランサーで IPv6 を有効にするには、左側のナビゲーションペインで [ネットワーク] タブをクリックし、IPv6 を有効にする CDN ディストリビューションまたはロードバランサーの名前を選択します。

3. リソースの管理ページで [Networking] タブを選択します。
4. ページのIPv6 ネットワーキングセクションで、 IPv6 のリソースを有効にするトグルを選択します。



リソースに対して IPv6 を有効にした後は、次の点に注意してください。

- CDN ディストリビューションまたはロードバランサーに対して IPv6 を有効にすると、そのリソースは IPv6 トラフィックを受け入れ始めます。インスタンスに対して IPv6 を有効にすると、IPv6 アドレスが割り当てられ、次の例に示すように、 IPv6 ファイアウォールが使用可能になります。

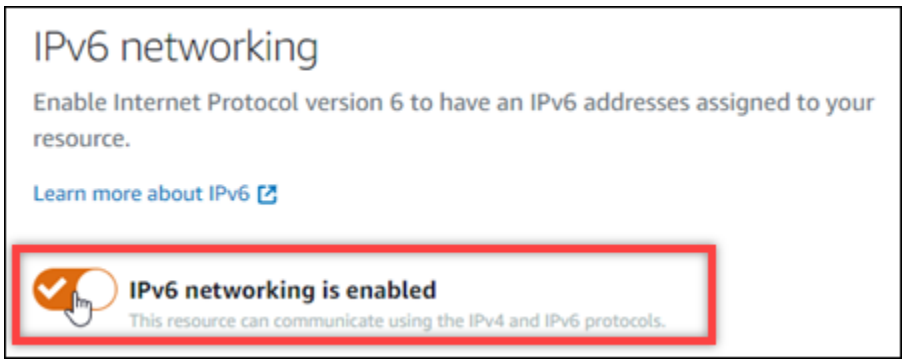


- 次のブループリントを使用するインスタンスでは、IPv6 を有効にした後、インスタンスが新しい IPv6 アドレスを認識できるように追加の手順が必要です。
  - cPanel — 詳細については、「cPanel インスタンスに IPv6 を設定する」を参照してください。
  - GitLab — 詳細については、「GitLab インスタンスに IPv6 を設定する」を参照してください。
  - Nginx — 詳細については、「Nginx インスタンスに IPv6 を設定する」を参照してください。
  - Plesk — 詳細については、「Plesk インスタンスに IPv6 を設定する」を参照してください。
- インスタンス、コンテナサービス、CDN ディストリビューション、またはロードバランサーのトラフィックを転送する登録済みのドメイン名を持つ場合、 IPv6 トラフィックをリソースにルーティングするために、ドメインの DNS に IPv6 アドレスレコード (AAAA) を作成するようにしてください。

Lightsail リソースの IPv6 ネットワークを無効にする

インスタンス、CDN ディストリビューション、およびロードバランサーの IPv6 を無効にするには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. IPv6 を無効にするリソースに応じて、次のいずれかの手順を実行します。
   - インスタンスの IPv6 を無効にするには、Lightsail ホームページの [インスタンス] タブをクリックし、IPv6 を無効にするインスタンスの名前を選択します。
   - CDN ディストリビューションまたはロードバランサーの IPv6 を無効にするには、左側のナビゲーションペインで [ネットワーキング] タブを選択し、IPv6 を無効にしたい CDN ディストリビューションまたはロードバランサーの名前を選択します。
3. リソースの管理ページで [Networking] タブを選択します。
4. ページの IPv6 ネットワーキングセクションで、 IPv6 を無効にするトグルを選択してください。



Lightsail インスタンスの IPv6 専用ネットワークを設定する

Lightsail インスタンスは 2 種類のネットワークをサポートしており、デュアルスタックネットワーク (IPv4・ IPv6) および IPv6 専用ネットワーク。デュアルスタックネットワークを使用すると、インスタンスにパブリック IPv4 およびパブリック IPv6 アドレスが割り当てられます。デュアルスタックネットワークを使用するインスタンスには、必要に応じて IPv6 を有効または無効にできます。

IPv6 専用ネットワークでは、インスタンスにパブリック IPv6 アドレスが割り当てられ、パブリック IPv4 トラフィックはサポートされていません。すべての Lightsail ブループリントが IPv6 と互換性があるわけではありません。どのブループリントが IPv6 専用をサポートしているかを確認するには、IPv6 互換性ブループリント を参照してください。また、IPv6 専用ネットワークを使用す

るインスタンスは、Lightsail コンテンツ配信ネットワーク (CDN) ディストリビューションのオリジンリソースとして設定することはできません。Lightsail ディストリビューションの詳細については、「Lightsail コンテンツ配信ディストリビューションでウェブコンテンツをグローバルに提供します。」を参照してください。

パブリック IPv4 アドレスが不要な場合は、IPv6 専用ネットワーキングを使用してください。ただし、まずはローカルネットワーク、コンピューター、デバイス、およびエンドユーザーが IPv6 を使用して通信できることを確認してください。詳細については、「Lightsail インスタンスの IPv6 到達可能性を検証する の IPv6 到達可能性」を参照してください。

ブループリントがサポートされている既存のインスタンスには、デュアルスタックネットワークと IPv6 専用ネットワークにネットワークタイプを切り替えることができます。IPv6 専用ネットワークに関する考慮事項を確認し、既存のインスタンスを変更するには、「Lightsail でインスタンスネットワークタイプを IPv6 またはデュアルスタックに切り替える」を参照してください。

トピック

- Lightsail でインスタンスネットワークタイプを IPv6 またはデュアルスタックに切り替える
- IPv6 互換性ブループリント

Lightsail でインスタンスネットワークタイプを IPv6 またはデュアルスタックに切り替える

インスタンスのネットワークタイプは、インターネットを介した通信に使用するプロトコルを決定します。インスタンスを作成するときは、デュアルスタックまたは IPv6 専用ネットワークのいずれかを選択します。また、既存のインスタンスのネットワークタイプをデュアルスタックから IPv6 専用に変更できます。また、その逆も可能です。ネットワークタイプを変更するには、ガイド付き、段階的なワークフローを使用するか、それぞれのステップを完了します。

ガイド付きのワークフローでは、新しいネットワークタイプが設定されている間、インスタンスは引き続き実行されます。このオプションは、変更の実行中、インスタンスがインターネット経由でアクセス可能な状態を保つために使用します。ただし、まずはローカルネットワーク、コンピュータ、デバイス、エンドユーザーが IPv6 を使用して通信できることを確認します。詳細については、「Lightsail インスタンスの IPv6 到達可能性を検証する」を参照してください。

個々のステップでは、インスタンスのスナップショットを作成し、スナップショットから新しいインスタンスを作成します。新しいインスタンスを作成するときに、別のネットワークタイプを選択できます。このオプションを使用して、他のインスタンスの設定を変更する前に IPv6 の互換性を検証します。開始する前に、IPv6 専用の考慮事項 を確認することをお勧めします。

IPv6 専用の考慮事項

次の考慮事項を確認してください。

- インスタンスプランは、ネットワークタイプが変更されるたびに変更されます。詳細については、「AWS Compute Blog」の「Announcing IPv6 instance bundles and pricing update on Amazon Lightsail」を参照してください。
- インスタンスは IPv6 経由でパブリックに通信します。受信または送信のパブリック IPv4 トラフィックはサポートされていません。Lightsail アカウント内の他のリソースと通信するためのプライベート IPv4 アドレスを受け取ります。詳細については、「Lightsail リソースの IP アドレスの表示と管理」を参照してください。
- IPv6 専用インスタンスは、Lightsail コンテンツ配信ネットワーク (CDN) ディストリビューションのオリジンとして設定することはできません。
- IPv6 専用インスタンスを Lightsail ロードバランサーに追加できます。
- インスタンスのデータ転送プランの許容量は、ネットワークタイプを変更すると引き継がれます。リセットされません。
- ローカルデバイス、ネットワーク、インターネットサービスプロバイダー (ISP) が IPv6 に対する互換性があることを確認します。詳細については、「Lightsail インスタンスの IPv6 到達可能性を検証する」を参照してください。

オプション: ガイド付きのワークフロー

ウィザードを使用してインスタンスネットワークタイプを設定するには

1. インスタンス管理ページの情報パネルで、[ネットワークタイプの変更] を選択します。
2. [ネットワークタイプの選択] では、[デュアルスタック] または [IPv6 専用] を選択します。選択したオプションの下に強調表示されている情報を確認して、[次へ] を選択します。
3. [リソースの確認] については、インスタンスに現在関連付けられているリソースに加えられる変更を確認します。リソースは、静的 IP アドレスまたは Lightsail ロードバランサーにすることができます。インスタンスにアタッチされたリソースがない場合、変更は行われません。リソースの変更は、次のステップでワークフローを完了するまで行われません。[次へ] を選択して続行します。
4. [変更の確認] については、新しいインスタンスネットワークタイプ、料金、リソースの変更を確認し、[変更の確認] を選択します。Lightsail リソースの設定を開始します。
5. (選択可能) ワークフローが完了した後、インスタンス設定を更新します。例えば、インスタンスに静的 IP をアタッチするか、IPv4 の DNS A レコードと IPv6 の AAAA レコードを更新しま

す。次の手順については、このガイドの the section called “次の手順” セクションを参照してください。

オプション: 個々のステップ

個々のステップを完了してインスタンスネットワークタイプを設定するには

1. インスタンス管理ページの [スナップショット] タブで、 [スナップショットの作成] を選択します。詳細については、次のトピックのいずれかを参照してください。
   - スナップショットを使用して Linux/Unix Lightsail インスタンスをバックアップする
   - Lightsail Windows Server インスタンスのスナップショットを作成する
2. スナップショットに名前を付け、[作成] を選択します。
3. スナップショットメニューで [新規インスタンスの作成] を選択します。詳細については、「スナップショットから Lightsail インスタンスを作成する」を参照してください。
4. [ネットワークタイプの選択] セクションで、[デュアルスタック] または [IPv6 専用] を選択します。
5. 残りのオプションを確認し、[インスタンスの作成] を選択します。新しいインスタンスが作成されました。
6. (選択可能) ワークフローが完了した後、インスタンス設定を更新します。例えば、インスタンスに静的 IP をアタッチするか、IPv4 の DNS A レコードと IPv6 の AAAA レコードを更新します。次の手順については、このガイドの the section called “次の手順” セクションを参照してください。

次の手順

インスタンスのネットワークタイプを変更した後に実行できる追加のタスクがいくつかあります:

- (IPv6 専用) アプリケーションとユーザーが IPv6 経由で通信できることを確認します。詳細については、「Lightsail インスタンスの IPv6 到達可能性を検証する」を参照してください。
- (デュアルスタック) インスタンスに静的 IP アドレスをアタッチします。詳細については、「静的 IP をインスタンスにアタッチする」を参照してください。
- (デュアルスタック) インスタンスを Lightsail ディストリビューションのオリジンとして設定します。詳細については、「CDN distributions in Lightsail」を参照してください。
- (両方) インスタンスのファイアウォール設定を追加または更新します。詳細については、「Lightsail のインスタンスファイアウォール」を参照してください。

- (両方) IPv4 の DNS A レコードと IPv6 の AAAA レコードを追加または更新します。詳細については、「Point your domain to an instance」を参照してください。
- (両方) Lightsail ロードバランサーにインスタンスを追加します。詳細については、「Lightsail のロードバランサー」を参照してください。

IPv6 互換性ブループリント

次の Lightsail ブループリントは、IPv6 専用のインスタンスプランと互換性があります。

- Windows Server 2022
- Windows Server 2019
- Windows Server 2016
- Amazon Linux 2023
- Amazon Linux 2
- AlmaLinux OS 9
- CentOS Stream 9
- Debian 11, 12, and 13
- FreeBSD 13, 14, and 15
- Ubuntu 20, 22, and 24
- SQL Server 2022 Express
- SQL Server 2019 Express
- SQL Server 2016 Express
- LAMP stack (PHP 8) packaged by Bitnami
- MEAN stack packaged by Bitnami
- Redmine packaged by Bitnami

Lightsail ブループリントの詳細については、the section called “ブループリント” を参照してください。

Lightsail のリージョンとアベイラビリティーゾーン

でリソースを作成するときはAmazon Lightsail、ユーザーに近い AWS リージョン でリソースを作成します。複数のアベイラビリティーゾーンを活用して、アプリケーションで耐障害性と高可用性を実

現することもできます。ユーザーの近くにリソースをプロビジョニングすることで、ユーザーはレイテンシーを短縮し、パフォーマンスを向上させることができます。たとえば、ブログのトラフィックが主にスイスで発生する場合は [フランクフルト] または [パリ] を選択します。

Lightsail でサポートされているオプトインリージョン (デフォルトで無効になっているリージョン) でアプリケーションを作成するとメリットが得られる場合は、リージョンを有効にすることができます。後でオプトインリージョンでリソースを運用しなくなった場合は、リージョンを無効にすることができます。オプトインリージョンの詳細については、AWS の用語集 の「Considerations before enabling and disabling Regions」および「opt-in Region entry」を参照してください。

> **Note**
>
> DNS ゾーンや Lightsail ディストリビューションなどのグローバルリソースは、米国東部 (バージニア北部) (us-east-1) リージョンでのみ作成されますが、サポートされているリソースはどの AWS リージョンでも参照できます。

トピック

- Lightsail のリージョン
- SSH キーと Lightsail リージョン
- Lightsail リージョンを使用するためのヒント
- Lightsail のアベイラビリティーゾーン
- アベイラビリティーゾーンと Lightsail アプリケーション
- Lightsail のオプトインリージョンを有効にする
- Lightsail のオプトインリージョンを無効にする

Lightsail のリージョン

Lightsail は、以下のリージョンで利用できます。

- 米国東部 (バージニア北部) (us-east-1)
- 米国東部 (オハイオ) (us-east-2)
- 米国西部 (オレゴン) (us-west-2)
- アジアパシフィック (ジャカルタ) (ap-southeast-3)*

- アジアパシフィック (ムンバイ) (ap-south-1)
- アジアパシフィック (ソウル) (ap-northeast-2)
- アジアパシフィック (シンガポール) (ap-southeast-1)
- アジアパシフィック (シドニー) (ap-southeast-2)
- アジアパシフィック (東京) (ap-northeast-1)
- カナダ (中部) (ca-central-1)
- 欧州 (フランクフルト) (eu-central-1)
- 欧州 (アイルランド) (eu-west-1)
- 欧州 (ロンドン) (eu-west-2)
- 欧州 (パリ) (eu-west-3)
- 欧州 (ストックホルム) (eu-north-1)

*このリージョンは、デフォルトでは無効になっています。*使用する前に、リージョンを有効にする必要があります。

SSH キーと Lightsail リージョン

Lightsail では、 AWS リージョンでインスタンスを作成すると同時に、そのリージョンで [デフォルト] の SSH キーが作成されます。このデフォルトキーは、その特定のリージョンのインスタンスに接続するためにのみ使用できます。詳細については、「SSH のキーペア」を参照してください。

Lightsail リージョンを使用するためのヒント

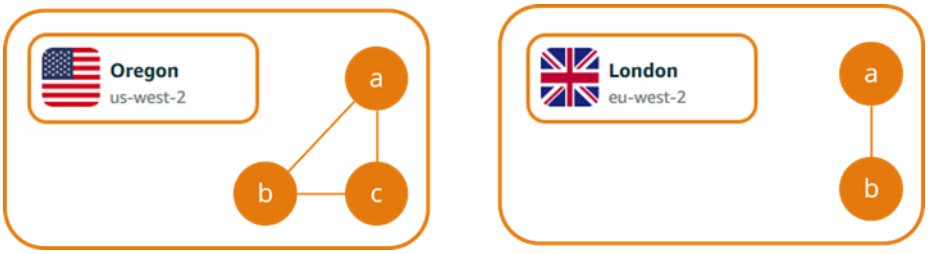
各 AWS リージョン は、他の から完全に分離されるように設計されています AWS リージョン。これにより、最大限の耐障害性と安定性が達成されます。

リージョン間のすべての通信は、パブリックインターネットを通して行われます。したがって、適切な暗号方式を使用してデータを保護する必要があります。リージョン間のデータ転送には料金がかかります。詳細についてはAmazon EC2 料金表 - データ転送を参照してください。

AWS Command Line Interface (AWS CLI) を使用してLightsailインスタンスを操作する場合は、 コマンドの AWS CLI `--region`オプションを使用してリージョンのコードを指定することをお勧めします。例えば、米国東部 (バージニア北部) (us-east-1) リージョンの DNS ゾーンとネットワークリソースに関する情報を返すように **`us-east-1`** を指定できます。オプションの使用 AWS CLI `--region`の詳細については、 AWS CLI リファレンスの「一般的なオプション」を参照してください。

Lightsail のアベイラビリティーゾーン

各 AWS リージョン には複数の分離されたアベイラビリティーゾーンがあり、リージョン名 () の後に文字で示されますus-east-2*a*。アベイラビリティーゾーンは、物理的に独立した独自のインフラストラクチャで実行されているデータセンターの集合です。アベイラビリティーゾーンは、高度な信頼性を実現できるよう設計されています。発電機や冷却装置などの一般的な障害発生点は、アベイラビリティーゾーン間では共有されていません。アベイラビリティーゾーンは物理的にも離れているため、火災、竜巻、洪水などの極度の災害であっても、影響を受けるのはその発生場所にある単一のアベイラビリティーゾーンのみです。



一度に Lightsail インスタンスを作成できるのは 1 つのアベイラビリティーゾーンだけです。インスタンスを作成した時点では、一部のアベイラビリティーゾーンが表示されないことがあります。アベイラビリティーゾーンのリストが全く表示されていない場合は、前のステップで選択したリージョンを確認してください。

アベイラビリティーゾーンと Lightsail アプリケーション

別のアベイラビリティーゾーンでもインスタンスを起動することにより、1 つの場所で障害が発生してもアプリケーションを保護できます。このような設定に進む前に、一般的なガイドラインとアプリケーション固有のガイドラインを確認してください。詳細については、「Configure Lightsail instances for load balancing」を参照してください。

既存のインスタンスのスケーリングを続行してアプリケーションを複数のアベイラビリティーゾーンで使用できるようにするには、まずインスタンスのスナップショットを作成します。次に、作成したスナップショットから新しいインスタンスを作成する際に、別のアベイラビリティーゾーンを選択します。スナップショットから作成された新しいインスタンスでは、Lightsail ロードバランサーを使用してウェブトラフィックをインスタンスに分散できます。

Lightsail のオプトインリージョンを有効にする

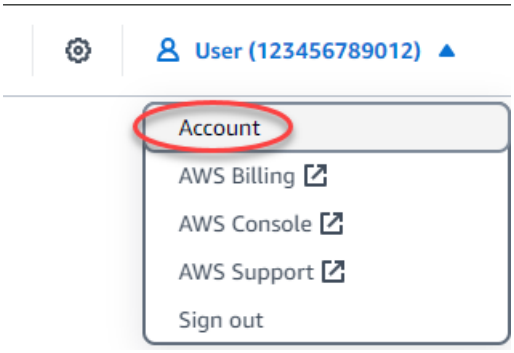
Lightsail でサポートされているオプトインリージョンは、追加料金なしで有効にできます。新しく無効化されたリージョンで作成するリソースに対してのみ、料金がかかります。このプロセスは、ほとんどのアカウントで数分かかります。リージョンの操作の詳細については、「Enable or disable AWS リージョン in your account」を参照してください。

> **Note**
>
> Lightsail でオプトインリージョンを使用する前に、Lightsail コンソールで[オプトインステータス]を[有効]にする必要があります。

この手順では、Lightsail コンソールからオプトインリージョンを有効にする方法について説明します。

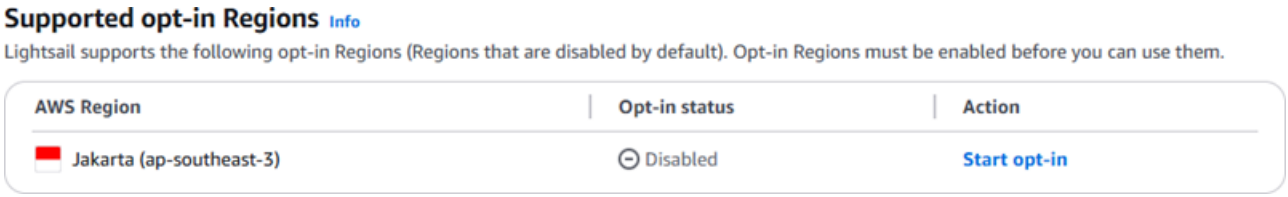
Lightsail のオプトインリージョンを有効にするには

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
3. ドロップダウンメニューで [Account (アカウント)] を選択します。



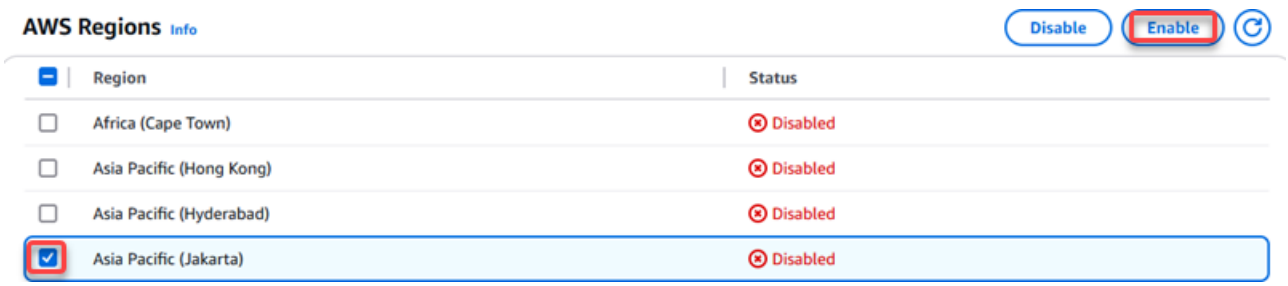
4. アカウントページで、[プロファイル] タブを選択します。

5. [サポートされるオプトインリージョン] セクションで、有効にするリージョンに対して [オプトインの開始] を選択します。

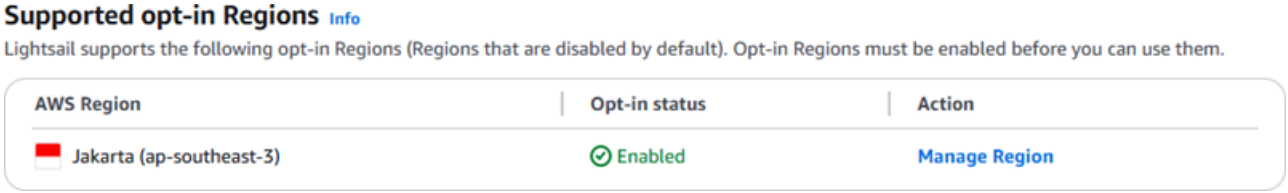


6. オプトイン情報を確認し、[オプトインの開始] を選択します。
7. 必要なステップを確認し、 AWS プロファイルの管理を選択して続行します。AWS アカウント管理 コンソールページが開きます。後のステップのために、Lightsail コンソールを開いたままにします。
8. AWS リージョン セクションで、有効にするリージョンを選択し、有効化を選択します。

> **Note**
>
> のリージョン名は、他の AWS サービスのリージョン名とは若干Lightsail異なります。たとえば、Lightsailコンソールのジャカルタ (ap-southeast-3) は、 AWS アカウント管理コンソールのアジアパシフィック (ジャカルタ) と同じです。



9. 表示される追加情報を確認してから、[リージョンを有効にする] を選択してオペレーションを続行します。
10. Lightsail コンソールのアカウントページに戻り、リージョンの[オプトインステータス]値を定期的に確認します。プロセスが完了し、[有効] に更新されるまで、[オプトインステータス]では [有効にしています] と表示されます。新しいリージョンでリソースをプロビジョニングできるようになりました。



Lightsail のオプトインリージョンを無効にする

オプトインリージョンでリソースを運用しなくなった場合は、リージョンを無効にすることができます。リージョンを無効にする前に、そのリージョンに削除するリソースがあるかどうかを評価する必要があります。これを行うには、Lightsail コンソールでリソースを確認するか、Billing and Cost Management コンソールを確認します。Billing and Cost Management コンソールからコストを確認すると、リージョン内のすべての AWS のサービス で使用されているリソースを特定できます。リージョンの操作の詳細については、アカウントの「有効または無効に AWS リージョン する」を参照してください。

無効なオプトインリージョンでのリソースの動作

まだ リソースが含まれているリージョンを無効にした場合、それらのリソースの料金は標準レートで引き続き発生します。さらに、リージョンでリソースを管理するためのアクセスは失われますが、リソースは引き続き実行されます。その結果、Lightsailコンソール、LightsailAPI、 AWS CLIまたは SDKs を使用してこれらのリソースを操作できなくなります。無効化されたリージョンのリソースを削除するには、リージョンがリソースに対してアクションを実行できるように再度有効にする必要があります。

例えば、WordPress インスタンスが実行中で、最初にインスタンスを削除せずにそのインスタンスが存在するオプトインリージョンを無効にした場合、料金は引き続き発生します。また、インスタンスは引き続き実行され、インターネットで使用できます。この例では、無効になっているリージョンで WordPress インスタンスを管理できなくなります。

リソースがリージョンにあるかどうかを確認する

次のステップを使用して、無効にするリージョンの Lightsail リソースに料金が発生しているかどうかを確認できます。リージョン内のリソースに料金が発生している場合は、先に進む前に削除できます。

リージョン別に Lightsail コストを確認するには

1. Billing and Cost Management コンソールにサインインします。
2. 左側のナビゲーションペインの [請求と支払い] で、[請求] を選択します。
3. [サービス別料金] タブの [Amazon Web Services, Inc. charges by service] で、Lightsail のエントリを展開します。
4. 無効にするリージョンで Lightsail の料金が発生している場合は、リージョン名の横にある展開アイコンを選択します。

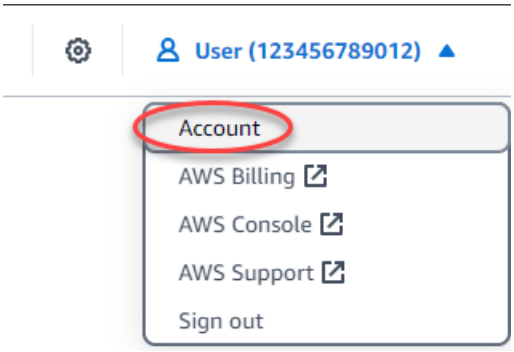
5. リージョン内で料金が発生するリソースタイプのリストを確認します。例えば、[Amazon Lightsail バンドル]の料金は、リージョンに Lightsail インスタンスが作成されていることを示します。
6. 追加料金の発生を停止するには、リージョン内のリソースをすべて削除してから無効にします。

オプトインリージョンを無効化する

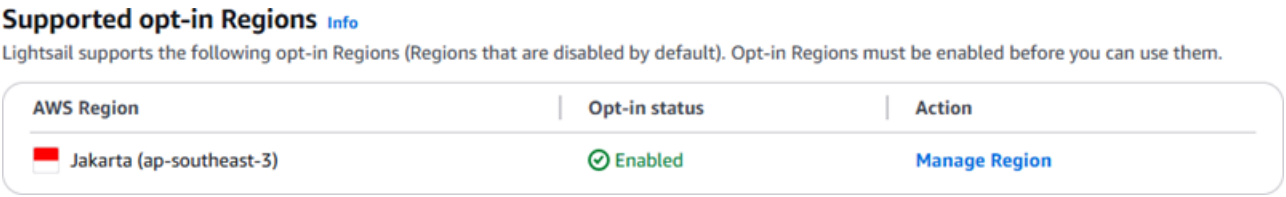
この手順を使用して、オプトインリージョンを無効にすることができます。先に進む前に、まず「リソースがリージョンにあるかどうかを確認する」セクションを確認してください。

Lightsail のオプトインリージョンを無効にするには

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
3. ドロップダウンメニューで [Account (アカウント)] を選択します。

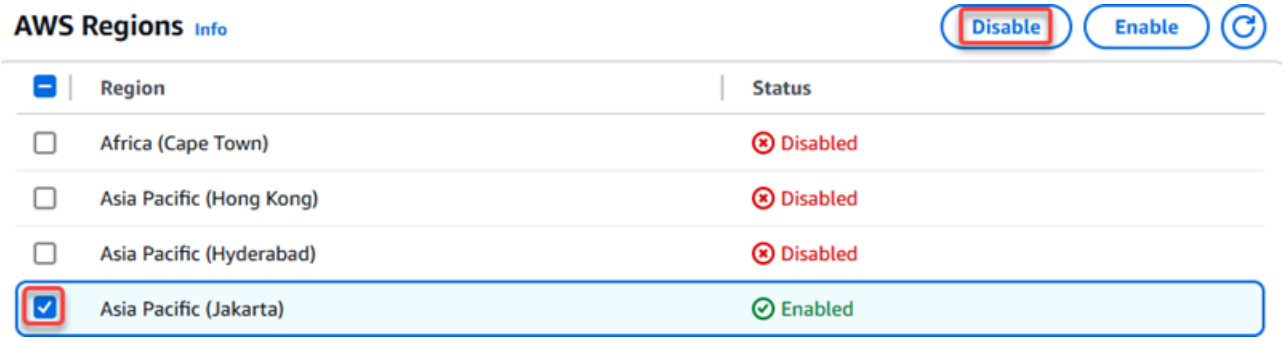


4. [プロファイル] タブの[オプトインリージョンのサポート]で、[リージョンの管理] を選択します。

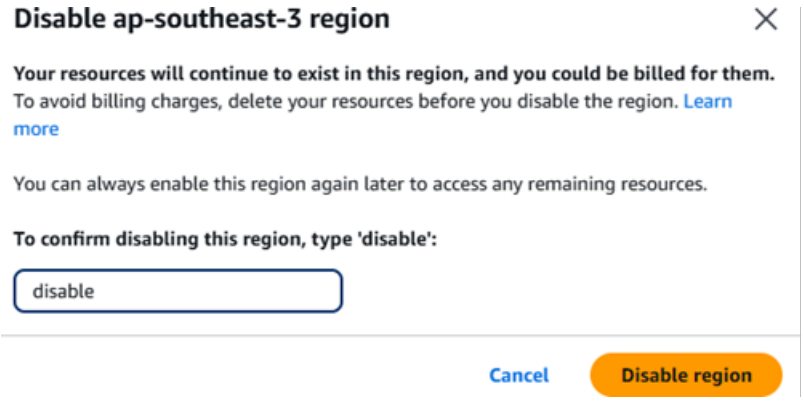


5. リージョン内のリソースを最初に削除するメッセージを確認します。続行する準備ができたら、AWS プロファイルの管理を選択します。
6. 必要なステップを確認し、[AWS アカウントの管理] を選択して続行します。AWS アカウント管理 ページが開きます。
7. AWS リージョン セクションで、無効にするリージョンを選択し、無効を選択します。

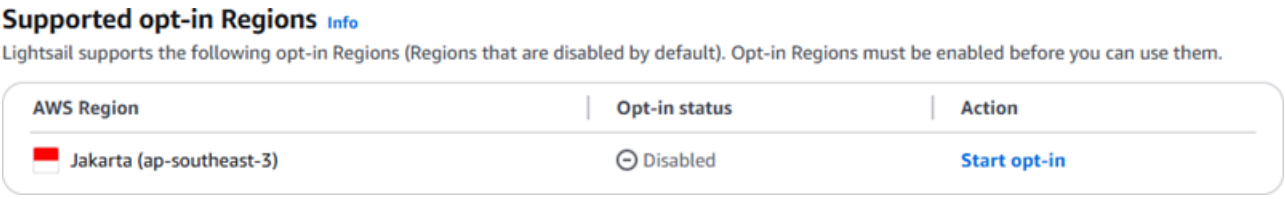
> **Note**
>
> のリージョン名は、他の AWS サービスのリージョン名とは若干Lightsail異なります。たとえば、Lightsailコンソールのジャカルタ (ap-southeast-3) は、 AWS アカウント管理コンソールのアジアパシフィック (ジャカルタ) と同じです。



8. 表示される追加情報を確認し、**`disable`** を入力して[リージョンの無効化] を選択し、オペレーションを続行します。



9. Lightsail コンソールのアカウントページに戻り、リージョンの[オプトインステータス]値を定期的に確認します。プロセスが完了し、[無効] に更新されるまで、オプトインステータスは [無効にしています] と表示されます。



VPC ピアリングを使用してLightsailリソースを AWS サービスに接続する

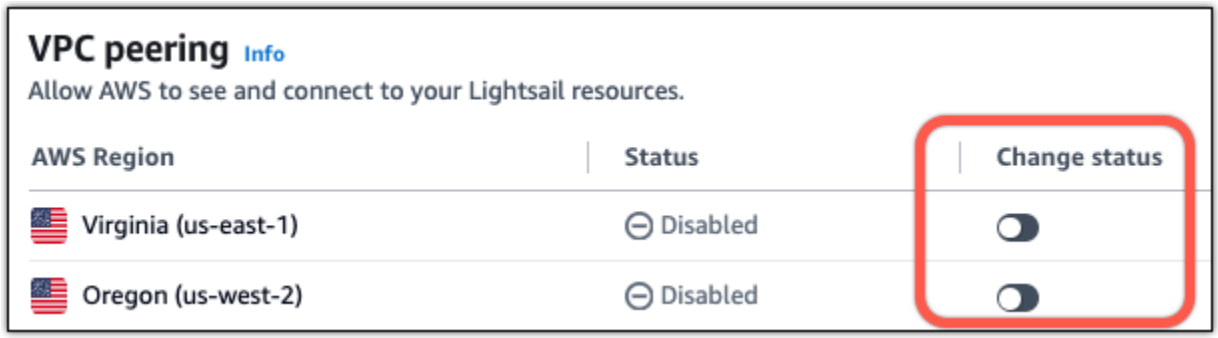
を使用するとAmazon Lightsail、Virtual Private Cloud (VPC) ピアリングを介して Amazon RDS データベースなどの AWS リソースに接続できます。VPC は、 AWS アカウント専用の仮想ネットワークです。Lightsail 内で作成するすべてのものは VPC 内に存在し、Lightsail VPC を Amazon VPC に接続できます。

Amazon S3、Amazon CloudFront、Amazon DynamoDB などの一部の AWS リソースでは、VPC ピアリングを有効にする必要はありません。

> **Note**
>
> で VPC ピアリングを有効にするにはLightsail、 にデフォルトの VPC が必要です AWS リージョン。ピアリング関係は、Lightsail 内のリソースと、VPC ピアリングを有効にしたリージョンのデフォルト VPC 内のリソースの間で確立されます。デフォルトの Amazon VPC がない場合は作成できます。詳細については、「Amazon VPC ユーザーガイド」の「デフォルト VPC」および「デフォルト VPC を作成する」を参照してください。
> AWS リージョンは互いに分離されているため、VPC は VPC を作成したリージョンでも分離されます。他のリソースを接続するLightsailリソースがある各 AWS リージョン で VPC ピアリングを有効にする必要があります。

デフォルトの Amazon VPC の準備が整ったら、以下の手順に従って、Lightsail VPC と Amazon VPC をピア接続します。

1. 「Lightsail コンソール」の上部にあるナビゲーションメニューで [ユーザー名] を選択します。
2. ドロップダウンから [アカウント] を選択します。
3. [Advanced] (アドバンス) タブを選択します。
4. VPC ピアリングを有効にする AWS リージョン の横にあるステータスを切り替えます。



ピア接続に失敗した場合は、VPC ピア接続を再度有効にしてみてください。それでも問題が解決しない場合は、AWS サポート までお問い合わせください。

ピアリングリクエストが成功 AWS すると、アカウントにピアリング接続が作成されます。ナビゲーションペインで、[Amazon VPC ダッシュボード] を選択し、[ピア接続] を選択して、作成されたピア接続を表示します。

Amazon VPC の詳細については、「Amazon VPC ユーザーガイド」の「VPC とサブネット」を参照してください。

他の AWS サービスとの通信を許可する

VPC ピアリングを有効にすると、接続する他の AWS サービスのリソースが Lightsail リソースからのインバウンドトラフィックを受け入れるようにする必要があります。他の AWS サービスのリソースをLightsailインスタンスに接続する場合は、ファイアウォールルールを追加して、必要なインバウンドトラフィックを許可できます。詳細については、「Add firewall rules to Lightsail instances」を参照してください。

実行する手順は、使用しているサービスとトラフィックのタイプによって異なります。Lightsail インスタンスを Amazon RDS データベースに接続する手順の例については、Amazon Lightsail「 Database Tips and Tricks AWS 」ブログ記事を参照してください。VPC ピアリングを使用して Lightsail と統合できるサービスの詳細については、VPC ピアリングを使用して他の Lightsail AWS サービスと統合する を参照してください。

Lightsail における SSL/TLS 証明書

Amazon Lightsail では SSL/TLS 証明書を使用して、Lightsail ロードバランサー、コンテンツ配信ネットワーク (CDN) ディストリビューション、コンテナサービスで使用できるカスタム (登録済み) ドメインを検証します。検証済みの証明書がそれらのいずれかに添付されかの Lightsail リソースに

添付された後、ドメインを介してそのリソースにルーティングされるトラフィックは、ハイパーテキスト転送プロトコルセキュア (HTTPS) を使用して暗号化されています。

Lightsail ロードバランサーのコンテンツ配信ネットワークディストリビューションおよびコンテナサービスで使用したいカスタム (登録済み) ドメイン用に暗号化されたウェブトラフィックを有効にするには、Amazon Lightsail に Transport Layer Security (TLS) 証明書を作成します。TLS は、Secure Socket Layer (SSL) の更新された、より安全なバージョンです。この Lightsail ドキュメントおよびコンソール全体を通じて、これらの証明書は [SSL/TLS] と呼ばれています。

> **Important**
>
> ロードバランサー、CDN ディストリビューション、コンテナサービスにアタッチできる Lightsail証明書は、 AWS Certificate Manager (ACM) サービスによって発行されます。2022 年 10 月 11 日以降、ロードバランサーの場合、CDN ディストリビューションとコンテナサービスのために Lightsail で取得した公的証明書は、ACM が管理する複数の中間認証局 (ICA) または下位 CA のいずれかから発行されます。詳細については、「AWS セキュリティブログ」の「Amazon introduces dynamic intermediate certificate authorities」を参照してください。

HTTPS を使用する理由

何よりも優先されるのはセキュリティです。HTTPS では、TLS を使用してデータが移動される、セキュリティの追加レイヤーが実現します。HTTPS 暗号化では、ウェブサーバーとクライアントのブラウザだけがトラフィックを復号化できるエンティティであるため、それらの間では内容が秘密に保たれます。HTTPS 接続では、クライアントがサーバーと交換するデータを別の当事者が変更できないため、HTTPS も安全です。

前述のセキュリティ上の利点を除き、HTTP に加えて HTTPS を使用する他の理由があります。たとえば、2014 年に Google は検索結果において安全なウェブサイトを上位にランク付けし始めました。つまり、HTTPS を使用するサイトは、HTTP しか使用していないサイト (他のすべては同じ) と比較して検索結果の上位にランク付けされます。

ランキングシグナルとしての HTTPS の詳細

プロセスの概要

Lightsail 証明書を使用するプロセスは簡単です。これには、次のステップが含まれます。

1. ロードバランサー、CDN ディストリビューション、コンテナサービスなど、Lightsail 証明書を使用できる Lightsail リソースを作成します。
2. Lightsail を使用してドメインの証明書を作成します。
3. ドメインの DNS に正規名 (CNME) レコードを追加して証明書を検証します
4. Lightsail リソースに検証済みの証明書を添付します。
5. ドメインのDNSを変更して、Lightsail リソースにトラフィックをルーティングします。

証明書がリソースに添付された後、ドメインを介してそのリソースにルーティングされるトラフィックは、HTTPS を使用して暗号化されます。

ディストリビューションおよびコンテナサービスを使用した SSL/TLS 証明書を使用する

Lightsail ディストリビューションおよびコンテナサービスには HTTPS が必要です。これらのリソースのいずれかを作成すると、HTTPS はデフォルトで、リソースのデフォルトドメインに有効化されます (例: `https://123456abcdef.cloudfront.net/` はディストリビューション用、`https://container-service-1.123456abcdef.us-west-2.cs.amazonlightsail.com/` はコンテナサービス用)。ユーザーのディストリビューションまたはコンテナサービスで登録済みのドメイン名 (例: `example.com`) を使用する場合は、Lightsail SSL/TLS 証明書を作成し、ドメイン名で検証し、ユーザーのリソースでカスタムドメインを有効にする必要があります。ディストリビューションまたはコンテナサービスでカスタムドメインを有効にすると、ドメインの検証済み証明書もリソースに添付されます。

ディストリビューションでカスタムドメインと HTTPS を有効化するには、次のリンクをクリックします。

- ディストリビューションの SSL/TLS 証明書を作成する
- ディストリビューションの SSL/TLS 証明書を表示する

- ディストリビューションの SSL/TLS 証明書を表示する
- ディストリビューションのカスタムドメインを有効にする
- ドメインをディストリビューションにポイントする

ディストリビューションの詳細については、「コンテンツ配信ネットワークディストリビューション」を参照してください。

コンテナサービスでカスタムドメインと HTTPS を有効化するには、次のリンクをクリックします。

- コンテナサービス用の SSL/TLS 証明書を作成する
- コンテナサービスの SSL/TLS 証明書を検証する
- カスタムドメインを有効にして管理する

コンテナサービスの詳細については、「コンテナサービス」を参照してください。

ロードバランサーでの SSL/TLS 証明書の使用

Lightsail ロードバランサーを作成すると、通常の HTTP トラフィックを処理するために、デフォルトでポート 80 が開きます。ポート 443 で HTTPS トラフィックを有効にするには、SSL/TLS 証明書を作成し、ドメイン名で検証し、ロードバランサーにアタッチする必要があります。

ロードバランサーあたり最大 2 つの SSL/TLS 証明書を作成できます。ロードバランサーごとに、一度に 1 つの証明書のみ使用できます。有効な使用中の証明書をロードバランサーから削除した場合、そのロードバランサーは、別の有効な証明書をアタッチするまで、指定されたドメインの HTTPS トラフィックを処理できなくなります。

ロードバランサーで HTTPS の使用を開始するには、次のリンクをクリックします。

- ロードバランサーを作成してインスタンスをアタッチする
- SSL/TLS 証明書を作成する
- ドメインの所有権を検証する
- 検証された証明書をアタッチして HTTPS を有効にする

ロードバランサーの詳細については、「ロードバランサー」を参照してください。

セキュア Lightsail コンテナサービスドメインの SSL/TLS 証明書を作成する

Lightsail コンテナサービス用の Amazon Lightsail TLS/SSL 証明書を作成できます。証明書を作成するときは、証明書のプライマリおよび代替ドメイン名を指定します。コンテナサービスのカスタムドメインを有効にして証明書を選択すると、コンテナサービスのカスタムドメインとして追加する証明書から最大 4 つのドメインを選択できます。ドメインの DNS レコードを更新してトラフィックをコンテナサービスに誘導すると、サービスはトラフィックを受け入れ、HTTPS を使用してコンテンツを提供します。アカウントに作成できる証明書の数にはクォータがあります。詳細については、「Lightsail サービスクォータ」を参照してください。

SSL/TLS 証明書の詳細については、「コンテナサービス証明書」を参照してください。

前提条件

開始する前に、Lightsail コンテナサービスを作成する必要があります。詳細については、「コンテナサービスを作成する」および「コンテナサービス」を参照してください。

コンテナサービス用の SSL/TLS 証明書を作成する

コンテナサービス用の SSL/TLS 証明書を作成するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. 証明書を作成するコンテナサービスの名前を選択します。
4. [Custom domains] (カスタムドメイン) のタブで、コンテナサービス管理ページを選択します。
5. ページの下部にある [Attached certificates] (アタッチされた証明書) セクションまで下にスクロールします。

   他の Lightsail リソース用に作成された証明書や、使用中あるいは使用されていない証明書など、すべての証明書がページの [Attached certificates] (アタッチされた証明書) セクションに表示されます。
6. [証明書の作成] を選択します。
7. 証明書を識別する一意の名前を [Certificate name] (証明書の名前) テキストボックスに入力します。次に、[Continue] (続行する) を選択します。
8. 証明書とともに使用するプライマリドメイン名 (例: `example.com`) を、[Specify up to 10 domains or subdomains] (最大 10 個のドメインまたはサブドメインを指定) フィールドに入力します。

9. (オプション) 別のドメイン名 (例: www.example.com) を [Specify up to 10 domains or subdomains] (最大 10 個のドメインまたはサブドメインを指定) フィールドに入力します。

   証明書には最大 9 個の代替ドメインを追加できます。カスタムドメインを有効にし、サービスの証明書を選択した後、コンテナサービスでは最大 4 つの証明書ドメインを使用できます。

10. [証明書の作成] を選択します。

    証明書のリクエストが送信されると、新しい証明書のステータスは [Attempting to validate your certificate] (証明書の検証を試行しています) に変更されます。この間、Lightsail は証明書の検証レコードをプライマリドメインの DNS に追加しようとします。しばらくすると、ステータスは [Valid] (有効) に変化します。

    自動検証に失敗した場合、コンテナサービスとともに使用する前に、ドメインで証明書を検証する必要があります。詳細については、「コンテナサービスの SSL/TLS 証明書を検証する」を参照してください。

トピック

- Lightsail コンテナサービスの SSL/TLS 証明書を検証する
- Lightsail コンテナサービスの SSL/TLS 証明書を表示する

Lightsail コンテナサービスの SSL/TLS 証明書を検証する

Amazon Lightsail SSL/TLS 証明書は、それを作成した後 Lightsail コンテナサービスで使用する前に、検証される必要があります。証明書に対するリクエストが送信されると、新しい証明書のステータスが[Attempting to validate your certificate] (証明書の検証を試みています) に変更されます。この間に Lightsail は、証明書に指定したドメイン名の DNS に証明書の検証レコードを追加することを試みます。しばらくすると、ステータスが [Valid] (有効) または [Validation timed out] (検証がタイムアウトしました) に変わります。

自動検証に失敗した場合は、証明書の作成時に指定したすべてのドメイン名を管理していることを確認します。そのためには、証明書で指定された各ドメインの DNS ゾーンに正規名 (CNAME) レコードを追加します。追加する必要のあるレコードが、[Validation details] (検証の詳細) セクションに一覧表示されます。

このガイドでは、Lightsail DNS ゾーンを使用して、証明書を手動で検証するための手順を説明します。Domain.com や GoDaddy などの別の DNS ホスティングプロバイダーを使用して、証明書を検

証する手順と類似しているかもしれません。Lightsail DNS ゾーンの詳細については、「DNS」を参照してください。

SSL/TLS 証明書の詳細については、「SSL/TLS 証明書」を参照してください。

前提条件

開始する前に、コンテナサービス用の SSL/TLS 証明書を作成する必要があります。詳細については、「コンテナサービスの SSL/TLS 証明書の作成」を参照してください。

CNAME レコードの値を取得して証明書を検証する

次の手順を実行して、証明書を検証するためにドメインに追加する必要があるCNAMEレコードを取得します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. 証明書を作成するコンテナサービスの名前を選択します。
4. [Custom domains] (カスタムドメイン) のタブで、コンテナサービス管理ページを選択します。
5. ページの下部にある [Attached certificates] (アタッチされた証明書) セクションまで下にスクロールします。

   他の Lightsail リソース用に作成された証明書や、検証が保留中の証明書など、すべての証明書が、このページの [Attached certificates] (アタッチされた証明書) セクションに一覧表示されます。
6. 検証する証明書を見つけて、[Validation details] (検証の詳細) を展開し、リストされているドメインごとに追加する必要がある CNAME レコードの [Name] (名前) と [Value] (値) をメモします。

   これらのレコードは、リストされているとおりに正確に追加する必要があります。この値はコピーしてテキストファイルに貼り付け、後で参照できるようにしておくことをお勧めします。詳細については、このガイドの「ドメインの DNS ゾーンに CNAME レコードを追加する」セクションを参照してください。

ドメインの DNS ゾーンに CNAME レコードを追加する

ドメインの DNS ゾーンにCNAMEレコードを追加するには、次のステップを実行します。

1. 左側のナビゲーションペインで [ドメインと DNS] を選択します。

2. ページの [DNS ゾーン] セクションで、証明書を検証するために CNAME レコードを追加するドメイン名を選択します。
3. [DNS records] (DNS レコード) タブを選択します。
4. DNS レコードの管理ページで、[Add record] (レコードの追加) を選択します。
5. [Record type] (レコードタイプ) のドロップダウンメニューから、[CNAME] を選択します。
6. [Record name] (レコード名) テキストボックスに、証明書から取得した値を使用して、CNAME レコードの [Name] (名前) を入力します。

   Lightsail コンソールには、ドメインの頂点部分があらかじめ入力されています。たとえば、`www.example.com` サブドメインを追加する場合は、`www` をテキストボックスに入力するだけで、レコードを保存するときに Lightsail が `.example.com` の部分を追加します。
7. [Route traffic to] (トラフィックのルーティング先) テキストボックスに、証明書から取得した CNAME レコード内にある [Value] (値) の部分を入力します。
8. 入力した値が、検証する証明書に記載されている値とまったく同じであることを確認します。
9. 保存アイコンを選択して、レコードを DNS ゾーンに保存します。

   これらのステップを繰り返して、検証が必要な証明書のドメインに CNAME レコードを追加します。変更がインターネットの DNS を通じて伝播されるまで待ちます。数分後に、証明書のステータスが [有効] に変わるはずです。詳細については、本ガイドの「証明書のステータスの検証」セクションを参照してください。

証明書のステータスを表示する

SSL/TLS 証明書のステータスを表示するには、次の手順を実行します。

1. 左側のナビゲーションペインで [コンテナ] を選択します。
2. 証明書のステータスを表示するコンテナサービスの名前を選択します。
3. コンテナサービス 管理ページで [カスタムドメイン] タブを選択します。
4. ページの下部にある [Attached certificates] (アタッチされた証明書) セクションまで下にスクロールします。

   ステータスが [Pending validation] (検証の保留中) や [Valid] (有効) の証明書を含む、すべての証明書が、このページの [Attached certificates] (アタッチされた証明書) セクションに一覧表示されます。

> **Note**
>
> 証明書の検証中、[Custom domains] (カスタムドメイン) ページを開けたままにしている場合は、証明書の更新ステータスを確認するためにページの更新が必要となる場合があります。

[有効] なステータスは、ドメインに追加した CNAME レコードで証明書が正常に検証されたことを確認します。証明書に関する重要な日付、暗号化の詳細、ID、検証レコードを表示するには、[Details] (詳細) を選択します。証明書は、有効にした日から 13 か月間有効で、その後、Lightsail は自動的に再検証を試みます。ドメインに追加した CNAME レコードは、リストされている有効期限日に証明書が再検証されるときに必要になるため、削除しないでください。

SSL/TLS 証明書を検証した後、コンテナサービスのカスタムドメインを有効にして、サービスで証明書のドメイン名を使用できるようにする必要があります。詳細については、「コンテナサービスでカスタムドメインを有効にして管理する」を参照してください。

Lightsail コンテナサービスの SSL/TLS 証明書を表示する

Lightsail コンテナサービス用に作成した Amazon Lightsail SSL/TLS 証明書を表示できます。これを行うには、Lightsail コンソールから、対象となるコンテナサービスの管理ページにアクセスします。

SSL/TLS 証明書の詳細については、「SSL/TLS 証明書」を参照してください。

前提条件

開始する前に、Lightsail コンテナサービスを作成する必要があります。詳細については、「Amazon Lightsail コンテナサービスの作成」および「コンテナサービス」を参照してください。

コンテナサービス用の SSL/TLS 証明書も作成しておく必要があります。詳細については、「コンテナサービスの SSL/TLS 証明書を作成する」を参照してください。

コンテナサービスの SSL/TLS 証明書を表示するには

以下の手順を実行して、コンテナサービスの SSL/TLS 証明書を表示します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. コンテナサービスの名前を選択します。

選択したコンテナサービスに関係なく、すべての証明書を表示できます。

4. コンテナサービス 管理ページで [Custom domains] (カスタムドメイン) タブを選択します。
5. ページの下部にある [Attached certificates] (アタッチされた証明書) セクションまで下にスクロールします。

   すべての証明書は、このページの [Attached certificates] (アタッチされた証明書) セクションに一覧表示されます。証明書に関する重要な日付、暗号化の詳細、ID、およびドメインを表示するには、[Details] (詳細) を選択します。証明書の検証レコードを表示するには、[Validation details] (検証の詳細) を選択します。証明書は、作成日から 13 か月間有効です。その後、Lightsail は自動的に証明書の再認証を試みます。ドメインに追加した CNAME レコードは、リストされている有効期限日に証明書が再検証されるときに必要になるため、削除しないでください。

   コンテナサービスで使用する有効な SSL/TLS 証明書を取得したら、サービスで証明書のドメイン名を使用できるようにカスタムドメインを有効にする必要があります。詳細については、「カスタムドメインの有効化と管理」を参照してください。

SSL/TLS 証明書を使用して Lightsail CDN ディストリビューションを保護する

Lightsailディストリビューションの Amazon Lightsail TLS/SSL 証明書を作成することができます。証明書を作成するときは、証明書のプライマリおよび代替ドメイン名を指定します。ディストリビューションのカスタムドメインを有効にして証明書を選択すると、それらのドメインはディストリビューションのカスタムドメインとして追加されます。ディストリビューションを指すようにドメインの DNS レコードを更新すると、ディストリビューションはトラフィックを受け入れ、HTTPS を使用してコンテンツを提供します。作成できる証明書の数にはクォータがあります。詳細については、「Lightsail サービスクォータ」を参照してください。

SSL/TLS 証明書の詳細については、「SSL/TLS 証明書」を参照してください。

> **Important**
>
> ディストリビューションの SSL/TLS 証明書を作成するときに指定したドメイン名は、Amazon CloudFront サービスのディストリビューションを含め、すべての Amazon Web Services (AWS) アカウントでの別のディストリビューションで使用することはできません。ドメインの証明書を作成することはできますが、その証明書をディストリビューションで使用することはできません。

前提条件

開始する前に、Lightsail ディストリビューションを作成する必要があります。詳細については、「ディストリビューションを作成する」および「コンテンツ配信ネットワークディストリビューション」を参照してください。

ディストリビューション用の SSL/TLS 証明書を作成する

ディストリビューション用の SSL/TLS 証明書を作成するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. 証明書を作成する対象のディストリビューションの名前を選択します。
4. ディストリビューションの管理ページのカスタムドメインタブを選択します。
5. ページ下部の [アタッチされた証明書] セクションまでスクロールします。

   ページの [Attached certificates] (アタッチされた証明書) セクションには、他のディストリビューション用に作成された証明書や、使用中の証明書も使用中でない証明書も含む、すべてのディストリビューション証明書が含まれます。
6. [証明書の作成] を選択します。
7. 証明書を識別する一意の名前を [Certificate name] (証明書の名前) テキストボックスに入力します。次に、[Continue] (続行する) を選択します。
8. 証明書とともに使用するプライマリドメイン名 (例: `example.com`) を、[Specify up to 10 domains or subdomains] (最大 10 個のドメインまたはサブドメインを指定) フィールドに入力します。
9. (オプション) 代替ドメイン名 (例: `www.example.com`) を、残りの [Specify up to 10 domains or subdomains] (最大 10 個のドメインまたはサブドメインを指定) フィールドに入力します。

   証明書には最大 9 個の代替ドメインを追加できます。カスタムドメインを有効にし、ディストリビューションの証明書を選択すると、すべての証明書ドメインをディストリビューションで使用できるようになります。
10. [作成] を選択します。

    証明書のリクエストが送信されると、新しい証明書のステータスは [Attempting to validate your certificate] (証明書の検証を試行しています) に変更されます。この間、Lightsail は証明書の検証レコードをプライマリドメインの DNS に追加しようとします。しばらくすると、ステータスは [Valid] (有効) に変化します。

自動検証に失敗した場合、ディストリビューションとともに使用する前に、ドメインで証明書を検証する必要があります。詳細については、「ディストリビューションの SSL/TLS 証明書を検証する」を参照してください。

トピック

- Lightsail ディストリビューションの SSL/TLS 証明書を表示する
- Lightsail ディストリビューションの SSL/TLS 証明書を検証する
- 最小 TLS プロトコルバージョンで Lightsail ディストリビューションを保護する
- Lightsail ディストリビューションから使用されていない SSL/TLS 証明書を削除する

Lightsail ディストリビューションの SSL/TLS 証明書を表示する

Lightsail ディストリビューションで作成した Amazon Lightsail SSL/TLS 証明書を表示できます。そのためには、Lightsail コンソールの任意のディストリビューションの管理ページにアクセスします。

SSL/TLS 証明書の詳細については、「SSL/TLS 証明書」を参照してください。

前提条件

開始する前に、Lightsail ディストリビューションを作成する必要があります。詳細については、「ディストリビューションを作成する」および「コンテンツ配信ネットワークディストリビューション」を参照してください。

ディストリビューションの SSL/TLS 証明書も作成しておく必要があります。詳細については、「ディストリビューションの SSL/TLS 証明書を作成する」を参照してください。

ディストリビューションの SSL/TLS 証明書を表示

以下の手順を実行して、ディストリビューションの SSL/TLS 証明書を表示します。

1. Lightsail コンソールにサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. ディストリビューションの名前を選択します。

   選択したディストリビューションに関係なく、すべての証明書を表示できます。
4. ディストリビューション管理ページで [カスタムドメイン] タブを選択します。

5. ページの下部にある [Attached certificates] (アタッチされた証明書) セクションまで下にスクロールします。

   すべてのディストリビューション証明書は、このページの [Attached certificates] (アタッチされた証明書) セクションに一覧表示されます。証明書に関する重要な日付、暗号化の詳細、ID、検証レコードを表示するには、[Validation details] (検証の詳細) を展開します。証明書は、作成日から 13 か月間有効です。その後、Lightsail は自動的に証明書の再認証を試みます。ドメインに追加した CNAME レコードは、リストされている有効期限日に証明書が再検証されるときに必要になるため、削除しないでください。

   ディストリビューションで使用する有効な SSL/TLS 証明書を取得したら、カスタムドメインを有効にして、ディストリビューションの証明書のドメイン名を使用できるようにする必要があります。詳細については、「ディストリビューション用のカスタムドメインを有効にする」を参照してください。

Lightsail ディストリビューションの SSL/TLS 証明書を検証する

Amazon Lightsail SSL / TLS 証明書は、作成した後 Lightsail ディストリビューションで使用する前に、検証を行う必要があります。証明書に対するリクエストが送信されると、新しい証明書のステータスが[Attempting to validate your certificate] (証明書の検証を試みています) に変更されます。この間に Lightsail は、証明書に指定したドメイン名の DNS に証明書の検証レコードを追加することを試みます。しばらくすると、ステータスが [Valid] (有効) または [Validation timed out] (検証がタイムアウトしました) に変わります。

自動検証に失敗した場合は、証明書の作成時に指定したすべてのドメイン名を管理していることを確認します。そのためには、証明書で指定された各ドメインの DNS ゾーンに正規名 (CNAME) レコードを追加します。追加する必要のあるレコードが、[Validation details] (検証の詳細) セクションに一覧表示されます。

このガイドでは、Lightsail DNS ゾーンを使用して、証明書を手動で検証するための手順を説明します。Domain.com や GoDaddy などの別の DNS ホスティングプロバイダーを使用して、証明書を検証する手順と類似しているかもしれません。Lightsail DNS ゾーンの詳細については、「DNS」を参照してください。

SSL/TLS 証明書の詳細については、「SSL/TLS 証明書」を参照してください。

目次

- 前提条件

- CNAME レコードの値を取得して証明書を検証する
- ドメインの DNS ゾーンに CNAME レコードを追加する
- ディストリビューション証明書のステータスを表示する

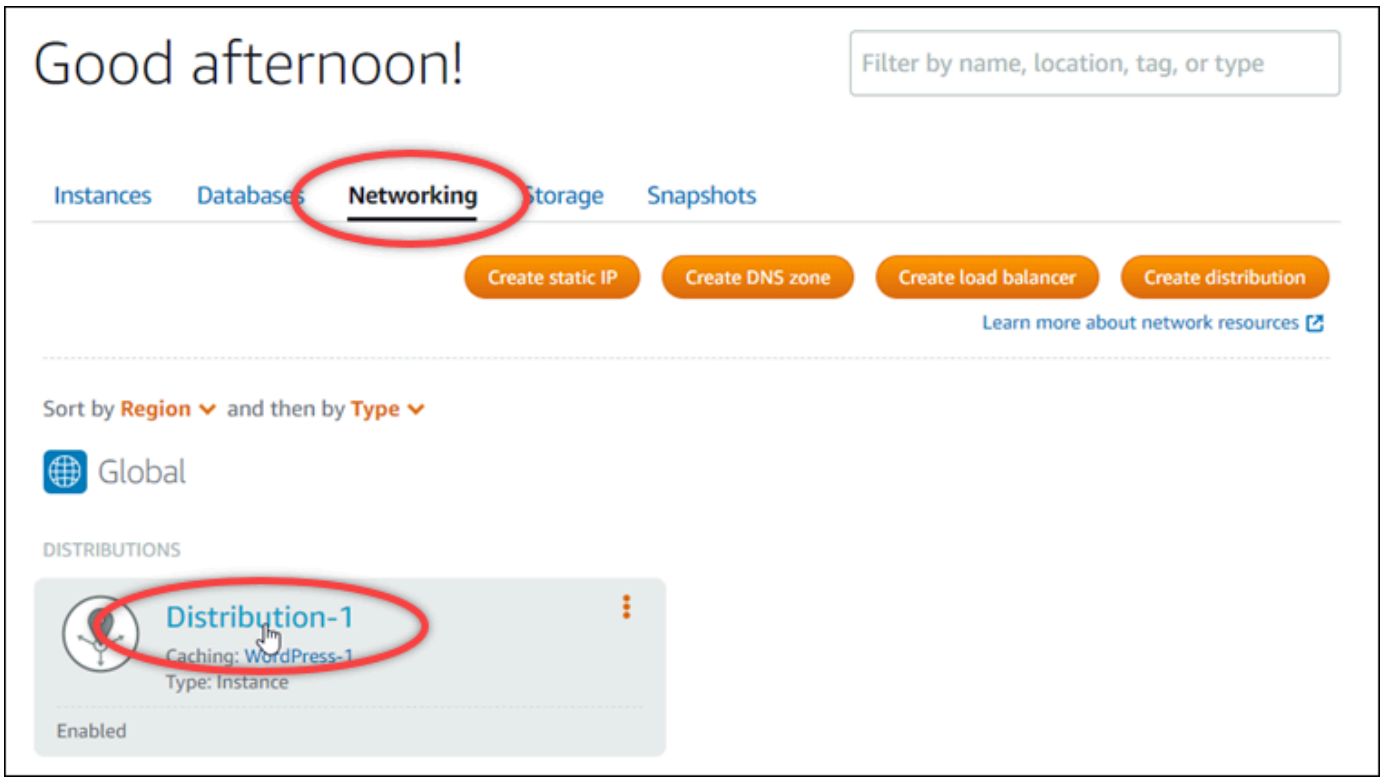
前提条件

開始する前に、ディストリビューション用の SSL/TLS 証明書を作成する必要があります。詳細については、「ディストリビューションの SSL/TLS 証明書を作成する」を参照してください。

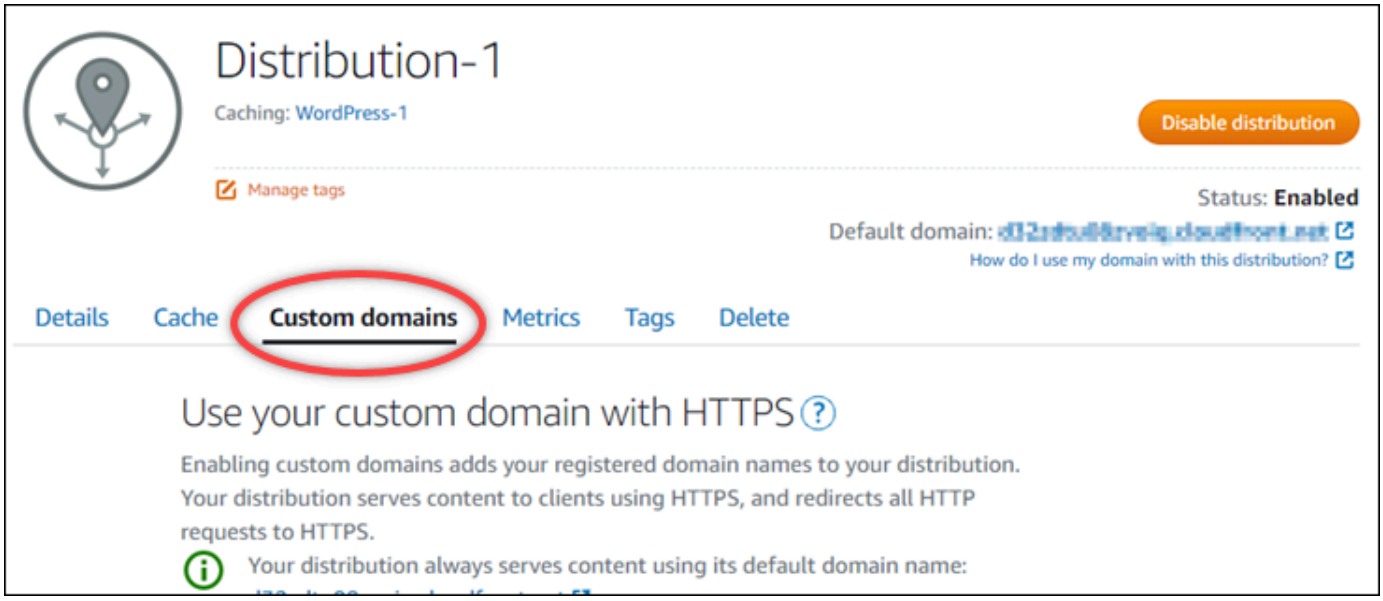
CNAME レコードの値を取得して証明書を検証する

次の手順を実行して、証明書を検証するためにドメインに追加する必要があるCNAMEレコードを取得します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. 証明書の CNAME レコード値を取得するディストリビューションの名前を選択します。



4. ディストリビューション管理ページで [カスタムドメイン] タブを選択します。



5. ページの下部にある [Attached certificates] (アタッチされた証明書) セクションまで下にスクロールします。

   他の Lightsail リソース用に作成された証明書や、検証が保留中の証明書など、すべてのディストリビューション証明書が、このページの [Attached certificates] (アタッチされた証明書) セクションに一覧表示されます。

6. 検証する証明書を見つけて、[Validation details] (検証の詳細) を展開し、リストされているドメインごとに追加する必要がある CNAME レコードの [Name] (名前) と [Value] (値) をメモします。

   これらのレコードは、リストされているとおりに正確に追加する必要があります。この値はコピーしてテキストファイルに貼り付け、後で参照できるようにしておくことをお勧めします。詳細については、このガイドの「ドメインの DNS ゾーンに CNAME レコードを追加する」セクションを参照してください。

ドメインの DNS ゾーンに CNAME レコードを追加する

ドメインの DNS ゾーンにCNAMEレコードを追加するには、次のステップを実行します。

1. 左側のナビゲーションペインで [ドメインと DNS] を選択します。
2. ページの [DNS ゾーン] セクションで、証明書を検証するために CNAME レコードを追加するドメイン名を選択します。
3. [DNS records] (DNS レコード) タブを選択します。
4. DNS レコードの管理ページで、[Add record] (レコードの追加) を選択します。

5. [Record type] (レコードタイプ) のドロップダウンメニューから、[CNAME] を選択します。
6. [Record name] (レコード名) テキストボックスに、証明書から取得した値を使用して、CNAME レコードの [Name] (名前) を入力します。

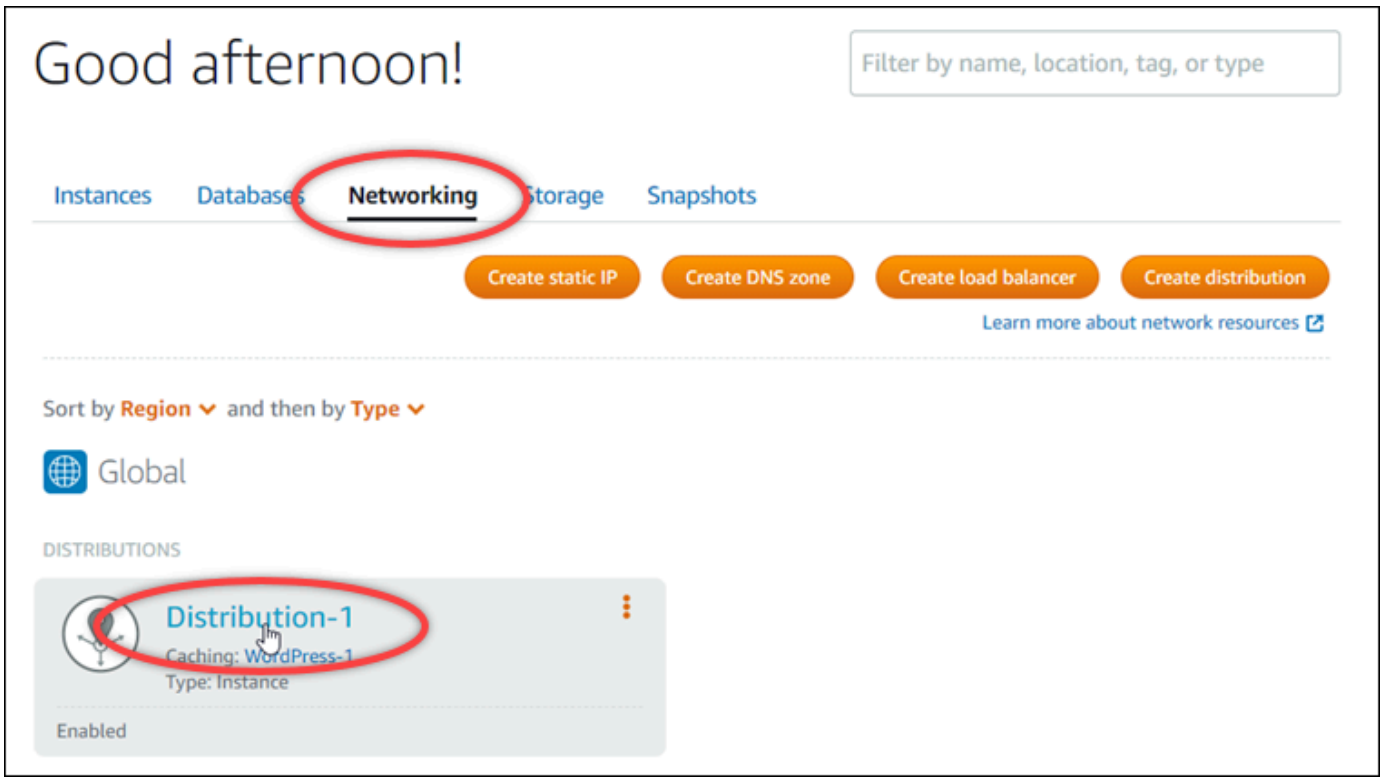
   Lightsail コンソールには、ドメインの頂点部分があらかじめ入力されています。たとえば、`www.example.com` サブドメインを追加する場合は、`www` をテキストボックスに入力するだけで、レコードを保存するときに Lightsail が `.example.com` の部分を追加します。
7. [Route traffic to] (トラフィックのルーティング先) テキストボックスに、証明書から取得した CNAME レコード内にある [Value] (値) の部分を入力します。
8. 入力した値が、検証する証明書に記載されている値とまったく同じであることを確認します。
9. 保存アイコンを選択して、レコードを DNS ゾーンに保存します。

   これらのステップを繰り返して、検証が必要な証明書のドメインに CNAME レコードを追加します。変更がインターネットの DNS を通じて伝播されるまで待ちます。数分後に、ディストリビューション証明書のステータスが [有効] に変わるはずです。詳細については、本ガイドの以下の「ディストリビューション証明書のステータスを表示する」セクションを参照してください。

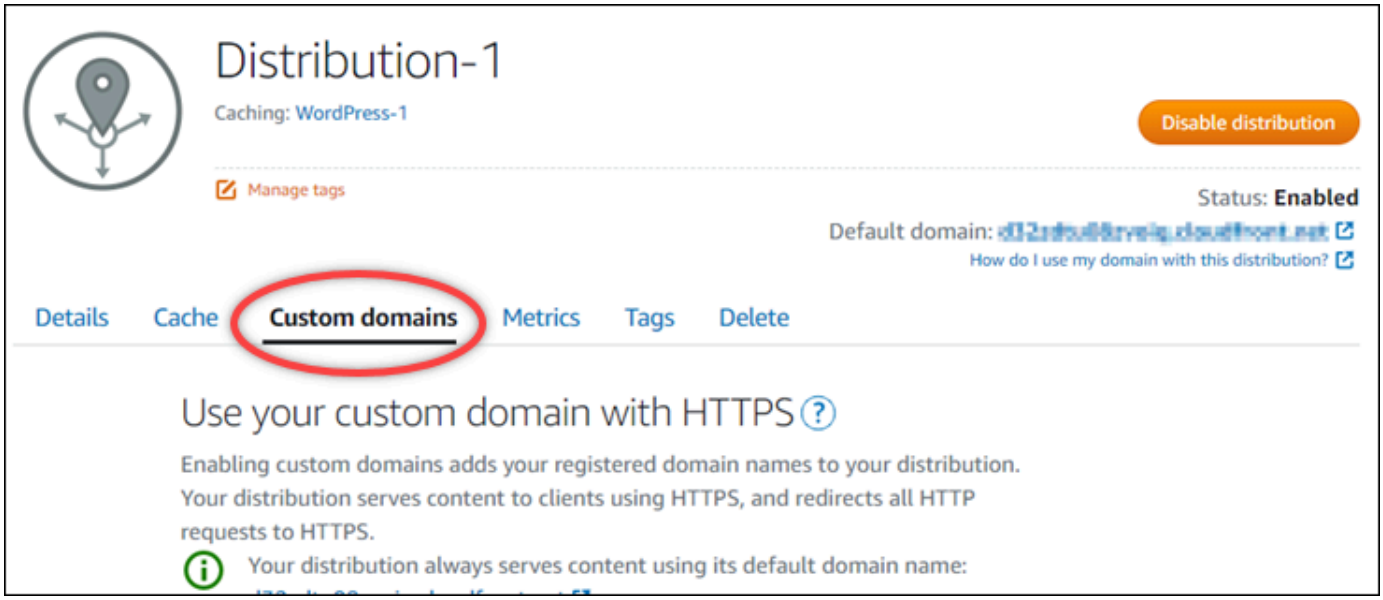
ディストリビューション証明書のステータスを表示する

以下の手順を実行して、ディストリビューションの SSL/TLS 証明書を表示します。

1. 左側のナビゲーションペインで、[ネットワーク] を選択します。
2. 証明書のステータスを表示するディストリビューションの名前を選択します。

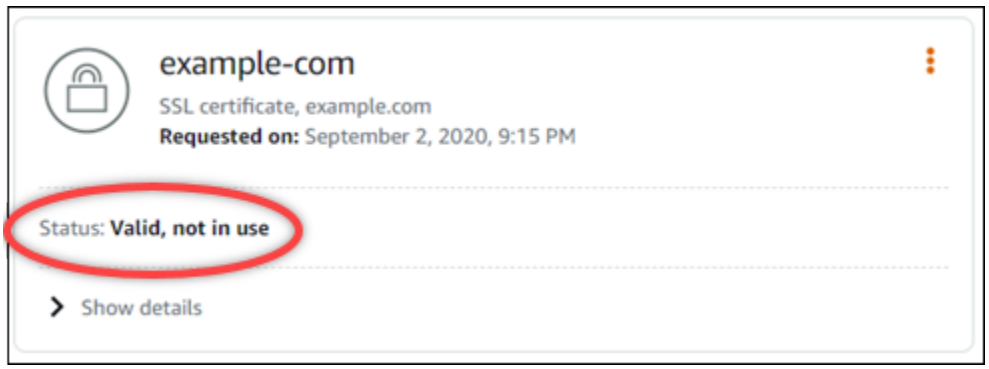


3. ディストリビューション管理ページで [カスタムドメイン] タブを選択します。



4. ページの下部にある [Attached certificates] (アタッチされた証明書) セクションまで下にスクロールします。

   ステータスが [Pending validation] (検証の保留中) および [Valid] (有効) の証明書を含む、すべてのディストリビューション証明書が、このページの [Attached certificates] (アタッチされた証明書) セクションに一覧表示されます。



[有効] なステータスは、ドメインに追加した CNAME レコードで証明書が正常に検証されたことを確認します。証明書に関する重要な日付、暗号化の詳細、ID、検証レコードを表示するには、[Details] (詳細) を選択します。証明書は、有効にした日から 13 か月間有効で、その後、Lightsail は自動的に再検証を試みます。ドメインに追加した CNAME レコードは、リストされている [有効期限] 日に証明書が再検証されるときに必要になるため、削除しないでください。

SSL/TLS 証明書を検証したら、ディストリビューションのカスタムドメインを有効にして、ディストリビューションの証明書のドメイン名を使用する必要があります。詳細については、「ディストリビューション用のカスタムドメインを有効にする」を参照してください。

最小 TLS プロトコルバージョンで Lightsail ディストリビューションを保護する

Amazon Lightsail は SSL/TLS 証明書を使用して、Lightsail ディストリビューションで使用できるカスタム (登録済み) ドメインを検証します。このガイドでは、SSL/TLS 証明書に設定可能なビューワーの最小 TLS プロトコルバージョン (プロトコルバージョン) に関する情報を提供します。SSL/TLS 証明書の詳細については、「Lightsail の SSL/TLS 証明書」を参照してください。ビューワーは、Lightsail ディストリビューションに関連付けられているエッジロケーションに HTTP リクエストを行うアプリケーションです。ディストリビューションの詳細については、Lightsail のコンテンツ配信ネットワークディストリビューションを参照してください。

`TLSv1.2_2021` プロトコルバージョンは、ディストリビューションのカスタムドメインを有効にすると、デフォルトで設定されます。このガイドで後述されるように、別のプロトコルバージョンを設定できます。Lightsail ディストリビューションはカスタム TLS プロトコルバージョンをサポートしていません。

サポートされるプロトコル

Lightsail ディストリビューションは、次の TLS プロトコルを使用して設定できます:

- (推奨) TLSv1.2_2021
- TLSv1.2_2019
- TLSv1.2_2018
- TLSv1.1_2016

前提条件

以下の前提条件を完了します (まだの場合)。

- Lightsail コンテンツ配信ネットワークディストリビューションを作成する
- ディストリビューションの SSL/TLS 証明書を作成する
- ディストリビューションの SSL/TLS 証明書を表示する
- ディストリビューションのカスタムドメインを有効にする
- ドメインをディストリビューションにポイントする

ディストリビューションの最小 TLS プロトコルバージョンを特定する

以下の手順に従い、Lightsail ディストリビューションの最小 TLS プロトコルバージョンを特定してください。

> **Note**
>
> このガイドでは、 AWS CloudShell を使用してアップグレードを実行します。CloudShell はブラウザーベースの事前認証済みシェルで、Lightsail コンソールから直接起動できます。CloudShell では、Bash、PowerShell、Z シェルなどの任意のシェルを使用して AWS CLI コマンドを実行できます。この手順は、コマンドラインツールのダウンロードもインストールも不要です。CloudShell をセットアップして使用する方法の詳細については、「Lightsail のAWS CloudShell」を参照してください。

1. ターミナル「AWS CloudShell」またはコマンドプロントウィンドウを開きます。
2. 次のコマンドを入力して、Lightsail ディストリビューションの最小 TLS プロトコルバージョンを特定します。

```
aws lightsail get-distributions --distribution-name DistributionName --region us-east-1 | grep "viewerMinimumTlsProtocolVersion"
```

コマンドで、*DistributionName* を変更したいディストリビューションの名前に置き換えます。

例

```
aws lightsail get-distributions --distribution-name Distribution-1 --region us-east-1 | grep "viewerMinimumTlsProtocolVersion"
```

コマンドは、ディストリビューションの最小 TLS プロトコルバージョンの ID を返します。

例

```
"viewerMinimumTlsProtocolVersion": "TLSv1.2_2021"
```

を使用して最小 TLS プロトコルバージョンを設定する AWS CLI

AWS Command Line Interface (AWS CLI) を使用して TLS プロトコルバージョンを設定するには、次の手順を実行します。これは、update-distribution コマンドを使用して実行できます。詳細については、「AWS CLI Command Reference」の「update-distribution 属性」を参照してください。

1. ターミナル「AWS CloudShell」またはコマンドプロントウィンドウを開きます。
2. ディストリビューションの最小 TLS プロトコルバージョンを変更するには、次のコマンドを入力します。

```
aws lightsail update-distribution --distribution-name DistributionName --viewer-minimum-tls-protocol-version ProtocolVersion
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- 更新したいディストリビューションの名前を含む *DistributionName*。
- 有効な TLS プロトコルバージョンを含む *ProtocolVersion*。例えば、TLSv1.2_2021、TLSv1.2_2019 などです。

例:

```
aws lightsail update-distribution --distribution-name  MyDistribution --viewer-
minimum-tls-protocol-version TLSv1.2_2021
```

変更が有効になるまで、少し時間がかかります。

Lightsail ディストリビューションから使用されていない SSL/TLS 証明書を削除する

> **Warning**
>
> SSL/TLS 証明書の削除は元に戻すことができません。

ディストリビューションで使用しなくなった Amazon Lightsail SSL/TLS 証明書を削除できます。たとえば、証明書の有効期限が切れており、検証済みの更新された証明書を既にアタッチしている場合などです。証明書の詳細については、「SSL/TLS 証明書」を参照してください。ディストリビューションの詳細については、「コンテンツ配信ネットワークディストリビューション」を参照してください。

365 日間に作成できる証明書の数にはクォータがあります。詳細については、AWS 全般のリファレンスの「Lightsail サービスクォータ」を参照してください。

ディストリビューション用の SSL/TLS 証明書を削除する

> **Important**
>
> [削除] オプションは、削除する証明書が使用中の場合は使用できません。使用中の証明書を削除するには、まず証明書を使用しているディストリビューションのカスタムドメインを変更するか、証明書を使用しているディストリビューションのカスタムドメインを無効にする必要があります。

ディストリビューション用の SSL/TLS 証明書を削除するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。

3. SSL/TLS 証明書を削除するディストリビューションの名前を選択します。証明書が現在使用中でない場合は、すべてのディストリビューションにすべての証明書がリストされるため、どのディストリビューションでも選択できます。
4. ディストリビューション管理ページで [カスタムドメイン] タブを選択します。
5. ページの [証明書] セクションで、削除する証明書の省略記号アイコン (⋮) を選択し、[削除] を選択します。

   [削除] オプションは、削除する証明書が使用中の場合は使用できません。使用中の証明書を削除するには、まず証明書を使用しているディストリビューションのカスタムドメインを変更するか、証明書を使用しているディストリビューションのカスタムドメインを無効にする必要があります。詳細については、「ディストリビューションのカスタムドメインを変更する」および「ディストリビューションのカスタムドメインを有効化する」を参照してください。
6. [はい、削除します] を選択して削除を確定します。

Lightsail ロードバランサー用の SSL/TLS 証明書を使用する HTTPS を有効にする

Lightsail ロードバランサーを作成したら、Transport Layer Security (TLS) 証明書をアタッチして HTTPS を有効にできます。SSL/TLS 証明書を使用すると、ロードバランサーが暗号化されたウェブトラフィックを処理できるようになるため、ユーザーに安全な体験を提供できます。詳細については、「SSL/TLS 証明書」を参照してください。

前提条件

開始する前に、以下のものが必要です。

- Lightsail ロードバランサー。詳細については、「ロードバランサーを作成する」を参照してください。

証明書リクエストを作成する

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. SSL/TLS 証明書を設定するロードバランサーの名前を選択します。
4. [Custom domains] (カスタムドメイン) タブを選択します。
5. [証明書の作成] を選択します。

6. 証明書の名前を入力するか、デフォルトをそのまま使用します。

   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。
7. プライマリドメイン (`www.example.com`) と、最大 9 つの代替ドメインまたはサブドメインを入力します。

   詳細については、「代替ドメインとサブドメインを SSL/TLS 証明書に追加する」を参照してください。
8. [証明書の作成] を選択します。

   Lightsail により検証プロセスが開始されます。72 時間以内にドメインを所有していることを検証してください。

   証明書を作成すると、ドメイン名とすべての代替ドメインおよびサブドメインと共に証明書が表示されます。ドメインおよびサブドメインごとに DNS レコードを作成する必要があります。

次のステップ

- 自分がドメインを所有していることを確認する

トピック

- 代替ドメインとサブドメインを Lightsail SSL/TLS 証明書に追加する
- Lightsail で CNAME レコードを使用して SSL/TLS 証明書ドメインを検証する
- 検証された SSL/TLS 証明書を Lightsail ロードバランサーにアタッチする
- Lightsail ロードバランサーから SSL/TLS 証明書を削除する

代替ドメインとサブドメインを Lightsail SSL/TLS 証明書に追加する

Lightsail ロードバランサーの SSL/TLS 証明書を作成するとき、代替ドメインとサブドメインを追加できます。これらの代替名を使用すると、ロードバランサーへのすべてのトラフィックが確実に暗号化されます。

プライマリドメインを指定する場合は、www.example.com などの完全修飾ドメイン名か、example.com などの apex ドメイン名を使用することができます。

ドメインおよびサブドメインの合計数が 10 を超えることはできないため、代替ドメインおよびサブドメインは証明書に最大 9 個追加できます。次のリストのようなエントリを追加することができます。

- example.com
- example.net
- blog.example.com
- myexamples.com

代替ドメインとサブドメインを含む証明書を作成するには

1. ロードバランサーを作成します (まだ作成していない場合)。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. Lightsail ロードバランサーを選択します。
4. [Custom domains] (カスタムドメイン) タブを選択します。
5. [証明書の作成] を選択します。
6. 証明書の名前を入力するか、デフォルトの名前をそのまま使用します。

   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。
7. プライマリドメイン (www.example.com) と、最大 9 つの代替ドメインまたはサブドメインを入力します。
8. [証明書の作成] を選択します。

   作成後、72 時間以内にドメインを所有していることを検証してください。

次の手順

- DNS を使用してドメインの所有権を検証する

検証後、検証された証明書を選択して Lightsail ロードバランサーに関連付けることができます。

- セッション永続性を有効にする

Lightsail で CNAME レコードを使用して SSL/TLS 証明書ドメインを検証する

Lightsail で SSL/TLS 証明書を作成した後は、その証明書に追加したすべてのドメインとサブドメインを、管理できていることを確認する必要があります。

目次

- ステップ 1: ドメインの Lightsail DNS ゾーンを作成する
- ステップ 2: ドメインの DNS ゾーンにレコードを追加する
- 次のステップ

ステップ 1: ドメインの Lightsail DNS ゾーンを作成する

まだ行っていない場合は、ドメインの Lightsail DNS ゾーンを作成します。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。

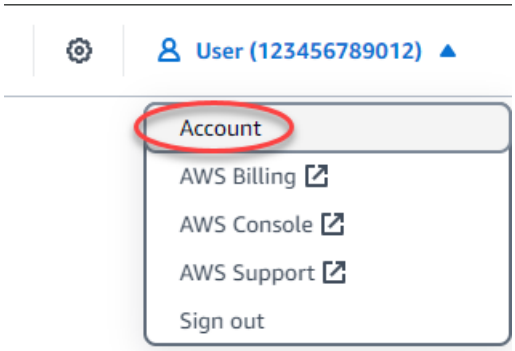
ステップ 2: ドメインの DNS ゾーンにレコードを追加する

作成した証明書により、一連の正規名 (CNAME) レコードが提供されます。これらのレコードをドメインの DNS ゾーンに追加して、そのドメインを所有し管理できていることを検証します。

> **Important**
>
> Lightsail は、ユーザーが証明書の作成時に指定したドメインまたはサブドメインが、そのユーザーにより管理されていることを自動的に検証しようとします。[Create certificate] (証明書を作成) を選択すると、CNAME レコードがドメインの DNS ゾーンに追加されます。自動検証が成功すると、証明書のステータスが [Attempting to validate your certificate] (証明書の検証を試行しています) から、[Valid, in use] (有効、使用中) に変わります。
> 自動検証が失敗した場合は、次の手順に進んでください。

以下のステップでは、Lightsail コンソールで CNAME レコードを取得して、ドメインの DNS ゾーンに追加する方法を説明します。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
3. ドロップダウンメニューで [Account (アカウント)] を選択します。



4. [証明書] タブを選択します。
5. 検証する証明書を見つけ、ドメインごとに追加する必要がある CNAME レコードの [Name] (名前) と [Value] (値) をメモします。

   Ctrl+C (Windows) または Cmd+C (Mac) を押してクリップボードにコピーします。

6. Windows を使用している場合はメモ帳、Mac を使用している場合は TextEdit などのテキストエディタを開きます。テキストファイルで、Windows を使用している場合は Ctrl+V、Mac を使用している場合は Cmd+V を押して、テキストファイルに値を貼り付けます。

   このテキストファイルは開いたままにしておきます。このガイドの後半でドメインの DNS ゾーンにレコードを追加するときに、これらの CNAME 値が必要になります。

7. Lightsail コンソールのトップナビゲーションバーの [ホーム] を選択します。
8. Lightsail のホームページで [Domains & DNS] (ドメイン & DNS) を選択します。
9. 証明書を使用するドメインの DNS ゾーンを選択します。
10. [DNS records] (DNS レコード) タブで [Add record] (レコードを追加) を選択します。
11. レコードタイプに、[CNAME] を選択します。
12. 証明書の CNAME レコードを含むテキストファイルに切り替えます。

    CNAME レコードの [名前] をコピーします。例えば、`_1bfb0b9ef15a50f9041e559d2c67b760`。
13. DNS レコードページに切り替え、[Name] (名前) の情報を [Record name] (レコード名) フィールドに貼り付けます。

> **Important**
>
> ドメイン名が含まれている CNAME レコード (`.example.com` など) を追加すると、ドメイン名が重複したレコード (`.example.com.example.com` など) になります。重複

を避けるために、必要な CNAME の一部だけが追加されるようにエントリを編集してください。_1bfb0b9ef15a50f9041e559d2c67b760 となります。

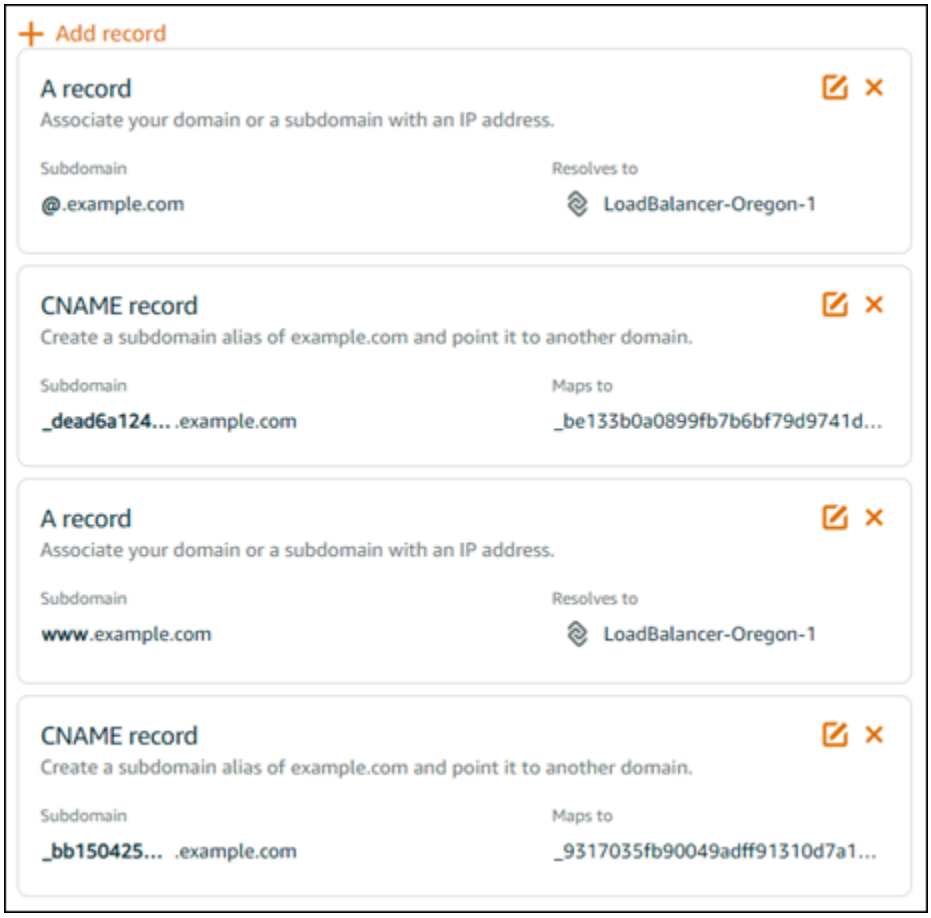
14. CNAME レコードの [値] をコピーします。例えば、_c9a0c385eda13283350e35f297469a13.hkvuiqjoua.acm-validations.aws.。
15. DNS レコードページに切り替え、[Value] (値) の情報を [Route traffic to] (トラフィックのルーティング先) フィールドに貼り付けます。
16. [Save] (保存) を選択して、レコードを追加します。
17. 代替サブドメインがある場合、[レコードの追加] を選択して別のレコードを追加します。

> **Note**
>
> 詳細については、「代替ドメインとサブドメインを Amazon Lightsail の SSL/TLS 証明書に追加する」を参照してください。

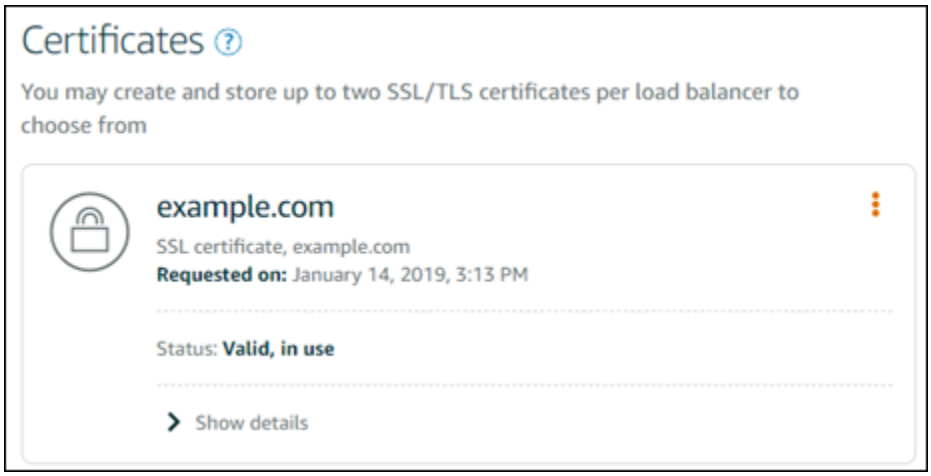
18. 代替サブドメインの CNAME レコードを (1 つ以上) 追加するには、ステップ 11〜17 を繰り返します。

DNS ゾーン管理ページからは、エイリアス (A) レコードを追加してロードバランサーをポイントしたり、他の Lightsail リソースを追加することもできます。

完了すると、DNS ゾーンは以下のスクリーンショットのようになります。



しばらくすると、ドメインの検証が完了し、証明書に次のメッセージが表示されます。

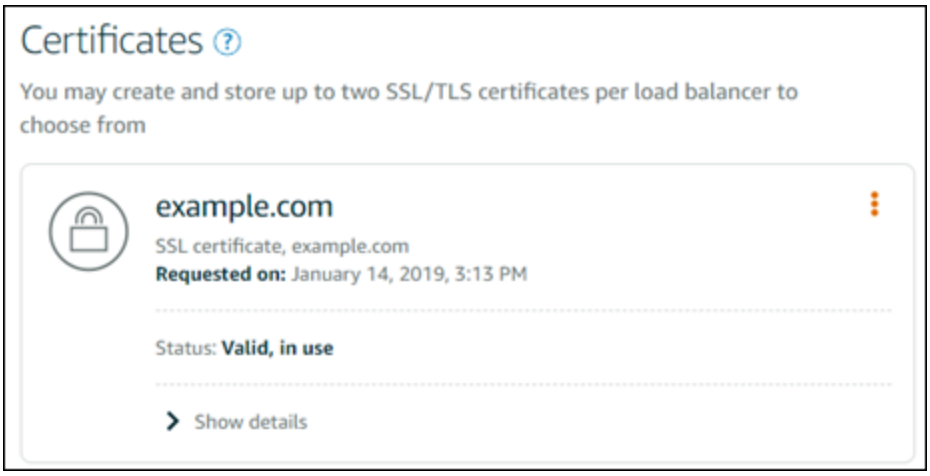


次のステップ

ドメインが確認されたら、「検証された SSL/TLS 証明書をロードバランサーにアタッチする」準備が整います。

検証された SSL/TLS 証明書を Lightsail ロードバランサーにアタッチする

ドメインを管理していることが確認されると、証明書のステータスが [Valid] (有効) に変わります。



次のステップでは、Lightsail ロードバランサーに証明書をアタッチします。

1. Lightsail のホームページで [ネットワーキング] を選択します。
2. ロードバランサーを選択します。
3. [Custom domains] (カスタムドメイン) タブを選択します。
4. [Certificates] (証明書) セクションで、[Attach certificate] (証明書のアタッチ) を選択します。
5. ドロップダウンリストから証明書を選択します。
6. [アタッチ] を選択して証明書をアタッチします。

Lightsail ロードバランサーから SSL/TLS 証明書を削除する

使用しなくなった SSL/TLS 証明書を削除できます。たとえば、証明書の有効期限が切れており、検証済みの更新された証明書を既にアタッチしている場合などです。証明書を削除する前に複製する場合、以下のステップ 5 と同じショートカットメニューから [重複] を選択します。

> **Important**
>
> 削除する証明書が有効であり、使用中の場合、ロードバランサーは暗号化 (HTTPS) トラフィックを処理できなくなります。Lightsail ロードバランサーでは、引き続き非暗号化 (HTTP) トラフィックがサポートされます。

SSL/TLS 証明書の削除は元に戻すことができません。365 日間に作成できる証明書の数にはクォータがあります。詳細については、AWS Certificate Manager ユーザーガイドの「クォータ」を参照してください。

1. 左側のナビゲーションペインで、[ネットワーク] を選択します。
2. SSL/TLS 証明書がアタッチされているロードバランサーを選択します。
3. ロードバランサーの管理ページで [インバウンドトラフィック] タブを選択します。
4. ページの [証明書] セクションで、削除する証明書の省略記号アイコン (:) を選択し、[削除] を選択します。

   [削除] オプションは、削除する証明書が使用中の場合は使用できません。使用中の証明書を削除するには、まず証明書を使用しているロードバランサーの証明書を変更するか、証明書を使用しているロードバランサーで HTTPS を無効にする必要があります。

Lightsail インスタンスに対するメールスパムを防ぐためにリバース DNS を設定する

E メールサーバーでは、ドメインネームシステム (DNS) 逆引き参照を使用して、メッセージの発信元を追跡し、それがスパムや悪意のあるメッセージではないことを確認します。DNS 逆引き参照は、IP アドレスのドメイン名を返します。これは、ドメインの IP アドレスを返す DNS 前方参照の反対です。

たとえば、IP アドレス `192.168.1.2` の DNS 逆引き参照がサブドメイン `mail.example.com` を返し、サブドメイン `mail.example.com` の DNS 前方参照が IP アドレス `192.168.1.2` を返すと、IP アドレス `192.168.1.2` の逆引き DNS は前方確認されます。詳細については、Wikipedia の「Forward-confirmed reverse DNS」を参照してください。

Amazon Lightsail インスタンスの逆引き DNS を設定するには、前提条件を満たした上で、アウトバウンドメッセージングのクォータの削除リクエストを AWS Support に送信します。これらのステップを以下のセクションで示します。

前提条件

逆引き DNS を設定するには、次の前提条件を以下の順に実行します。

1. E メールサーバーとして使用する Lightsail インスタンスを作成します。詳細については、「インスタンスを作成する」を参照してください。
2. 逆引き DNS レコードとして使用する静的 IP を作成し、これを実行中のインスタンスにアタッチします。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。

> **Important**
>
> インスタンスの初回作成時に割り当てられるデフォルトのパブリック IP を逆引き DNS として使用することはできません。インスタンスのデフォルトのパブリック IP は、インスタンスの停止や開始に伴って変わるためです。

3. ドメインの DNS ゾーンで、サブドメイン (`mail.example.com` など) をポイントするエイリアスレコード (A レコード) を、実行中のインスタンスの静的 IP アドレスに追加します。このサブドメインは、静的 IP アドレスの DNS 逆引き参照を実行したときに返されます。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。

> **Note**
>
> ドメインの DNS レコードの管理を Lightsail に引き渡すことをお勧めします。これにより、ドメインなどのすべてのリソースを 1 つの場所 (Lightsail コンソール) で管理できます。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。

4. 変更がインターネットの DNS を通じて伝播されるまで待ちます。次に、AWS Support に対して逆引き DNS の設定リクエストを送信します。

AWS Support に逆引き DNS の設定リクエストを送信する

セキュリティ上の理由から、Lightsail では、ポート 25 を介したアウトバウンドメッセージがデフォルトで制限されます。ただし、このクォータをアカウントから削除するリクエストを AWS Support に送信し、静的 IP の逆引き DNS を設定できます。

AWS Support にリクエストを送信するには

1. Lightsail コンソールに AWS アカウントのルートユーザーとしてサインインします。

> **Important**
>
> リクエストは AWS アカウントのルートユーザーを使用して送信する必要があります。AWS アカウントのルートユーザーの詳細については、「AWS アカウントのルートユーザー」を参照してください。

2. E メール送信制限解除リクエストフォームに移動し、以下の必須情報を入力します。

> **Note**
>
> このフォームは、Elastic IP (EIP) や EC2 インスタンスなどの Amazon Elastic Compute (EC2) リソースを参照します。ただし、静的 IP や Lightsail インスタンスなどの Lightsail リソース用のフォームを使用することもできます。

- E メールアドレス – リクエストに関するメッセージを受信できる E メールアドレスを入力します。このテキストボックスには、アカウントの E メールアドレスが事前設定されます。
- ユースケースの説明 - E メールのクォータの削除をリクエストする理由を入力します。
- Elastic IP アドレス – このガイドの前提条件のステップ 2 でインスタンスにアタッチした静的 IP アドレスを入力します。静的 IP アドレスを 2 つまで入力できます。
- EIP の逆引き DNS レコード – このガイドの前提条件のステップ 3 で定義したサブドメインを入力します。このドメインは、DNS 逆引き参照を実行したときに返されます。

3. 終了したら、[送信] を選択します。

AWS Support でリクエストが受理されると、静的 IP アドレスが DNS 逆引き参照で前方確認されます。

後で Lightsail アカウントから静的 IP アドレスを削除する場合は、逆引き DNS 設定の削除リクエストを AWS Support に送信する必要があります。逆引き DNS 設定が削除されると、Lightsail コンソールを使用して Lightsail アカウントから静的 IP アドレスを削除できます。詳細については、「静的 IP を削除する」を参照してください。

Lightsail オブジェクトストレージバケットを使用してデータを保存および管理する

Amazon Lightsail オブジェクトストレージサービスを使用して、インターネット上のどこからでも、いつでもオブジェクトの格納と取得ができます。ウェブスケールのコンピューティングをデベロッパーが簡単に実行できるように設計されており、Amazon Simple Storage Service (Amazon S3) を使用して構築されています。Lightsail オブジェクトストレージにより、Amazon が独自のグローバルなウェブサイトのネットワークを運用するために使用しているのと同じ、スケーラビリティと信頼性が高く、高速で低価格のデータストレージインフラストラクチャにアクセスできます。このサービスの目的は、規模の拡大や縮小のメリットを最大限に活かし、その利益をお客様に提供することです。

オブジェクトストレージの概念

次の概念と用語は Lightsail オブジェクトストレージに適用されます。

バケット

バケットとは、オブジェクトのコンテナで、Lightsail オブジェクトストレージサービスに保存されています。すべてのオブジェクトは、バケットに格納され、それぞれのバケットには、独自の URL があります。たとえば、`media/sailbot.jpg` という名前のオブジェクトが米国東部 (バージニア北部) リージョン (`us-east-1`) の `amzn-s3-demo-bucket` バケットに保存される場合、`https://amzn-s3-demo-bucket.us-east-1.amazonaws.com/media/sailbot.jpg` に類似した URL を使用してアドレスを指定できます。

Lightsail が利用可能な AWS リージョン でバケットを作成できます。で使用できる AWS リージョン Lightsail の詳細については、「 全般のリファレンス」の「リージョンとエンドポイント」を参照してください。 AWS

バケットストレージプラン

AWS API のバンドルと呼ばれるストレージプランは、バケットの月額コスト、ストレージ領域、データ転送クォータを指定します。最初にバケットを作成するときに、ストレージプランを選択する必要があります。バケットの起動後に変更することもできます。

バケットのプランは、毎月の AWS 請求サイクル内で 1 回だけ変更できます。バケットがストレージ領域またはデータ転送クォータを一貫して上回っている場合、またはバケットの使用量がストレージ

領域またはデータ転送クォータより低い範囲にある場合、バケットのプランを変更します。バケットの使用量の変動が予測できない場合があるため、バケットのプランの変更は、短期的な毎月のコスト削減策ではなく、長期的な戦略としてのみ行うことを強くお勧めします。今後長期にわたりバケットに十分なストレージ容量とデータ転送クォータを提供するストレージプランを選択します。

オブジェクト

オブジェクトは、バケットに格納される基本エンティティです。バケットにアップロードしたファイルは、格納されている間、オブジェクトと呼ばれます。オブジェクトは、データとメタデータで構成されます。データ部分は、Lightsail オブジェクトストレージサービスの不透明体です。メタデータは、オブジェクトを表現する名前と値のペアのセットです。これには最終更新日などのデフォルトのメタデータや、標準 HTTP メタデータ (Content-Type など) が含まれます。

オブジェクトは、キー名とバージョン ID によってバケット内で一意に識別されます。

オブジェクトキー名

キー名とは、バケット内のオブジェクトの固有の識別子です。バケット内のすべてのオブジェクトは、厳密に 1 個のキーを持ちます。バケット、キー、バージョン ID の組み合わせで、各オブジェクトを一意に識別します。そのため、Lightsail オブジェクトストレージを「バケット + キー + バージョン」とオブジェクト自体の間の基本データマップと考えることができます。Lightsail オブジェクトストレージ内の各オブジェクトは、ウェブサービスエンドポイント、バケット名、キー、およびオプションでバージョンを組み合わせることで一意にアクセスすることができます。たとえば、`https://amzn-s3-demo-bucket.us-east-1.amazonaws.com/media/sailbot.jpg` という URL では、「`amzn-s3-demo-bucket`」がバケットの名前で、「`media/sailbot.jpg`」がオブジェクトキー名になります。

オブジェクトのバージョニング

バージョニングとは、同じバケット内でオブジェクトの複数のバリアントを保持する機能です。バージョニングを有効にすると、 バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。バージョニングを使用すれば、意図しないユーザーアクションからもアプリケーション障害からも、より簡単に復旧できます。

バケットの作成時、デフォルトでは、バージョニングは無効になっています。バージョニングを有効にすると、バケットに格納するすべてのオブジェクトのすべてのバージョンは、保存されたバージョンを手動で削除するまで保持されます。たとえば、`media/sailbot.jpg` オブジェクトを格納した後で、同じオブジェクトキー名を持つより大きなファイルを格納すると、元の小さいオブジェクトが以前のバージョンとして保持されます。新しい大きなオブジェクトは現行のバージョンになりま

す。以前のバージョンのオブジェクトが必要ないと判断した場合、オブジェクトを削除できます。オブジェクトの最新バージョンを削除すると、そのオブジェクトの以前のバージョンはすべて削除されます。

格納されたオブジェクトバージョンは、格納されたオブジェクトの現在のバージョンと同じ方法で、バケットのストレージ領域を消費します。バージョニングを有効にした後は、そのバージョニングを中断して、オブジェクトバージョンの保存を停止できます。これにより、新しいオブジェクトバージョンをアップロードするときに、バケットのストレージ領域が消費されることも少なくなります。バージョニングを一時停止すると、保存されたオブジェクトバージョンは保持されますが、バージョニングを一時停止している間にアップロードした新しいオブジェクトバージョンは保持されません。

バケットとオブジェクトのアクセス

デフォルトでは、すべてのオブジェクトストレージリソース (バケットとオブジェクト) はプライベートです。バケットを作成した Lightsail アカウント (バケット所有者) のみが、バケットとそれに含まれるオブジェクトにアクセスできます。バケット所有者は、他のユーザーにアクセス許可を付与することもできます。これは、すべてのオブジェクトまたは個々のオブジェクトを公開に設定することで実行できます。これにより、世界中の誰でも読みやすくなります。また、プログラムによる完全なアクセスを付与するには、Lightsail インスタンスをバケットに追加するか、バケットのアクセスキーを作成します。最後に、他の AWS アカウントにバケットへの読み取り専用アクセスをプログラムで許可できます。

AWS リージョン

オブジェクトLightsailストレージバケットLightsailは、 が利用可能なすべての AWS リージョン で作成できます。レイテンシーを最適化し、コストを最小限に抑えて規制用件に対応できるリージョンを選ぶとよいでしょう。に保存されているオブジェクトは、明示的に別のリージョンに転送しない限り、リージョンを離れ AWS リージョン ません。たとえば、米国西部 (オレゴン) リージョンに格納されたオブジェクトがそこから移動することはありません。

バケットとオブジェクトを管理する

Lightsail オブジェクトストレージは、シンプルさと堅牢性を重視し、必要な機能に絞って提供しています。バケットとオブジェクトを管理する要素の一部を次に示します。

- バケットの作成 – データを格納するバケットを作成します。バケットとは、Lightsail オブジェクトストレージにおける基本的なコンテナです。詳細については、「バケットの作成」を参照してください。

- データの保存 – Lightsailコンソール、 AWS Command Line Interface (AWS CLI)、API AWS APIs。ファイルのアップロードに関する詳細については、「バケットにファイルをアップロードする」を参照してください。
- データのダウンロード — 保存したオブジェクトをいつでもダウンロードできます。詳細については、「バケットからオブジェクトをダウンロードする」を参照してください。
- アクセス権の付与 — 外部 (ソフトウェアや個人など) からの、バケット内のデータのアップロードまたはデータのダウンロードアクセスを許可または拒否します。認証メカニズムによって、データソースを不正アクセスから保護することができます。詳細については、「バケットのアクセス許可」を参照してください。
- バージョニング管理 — バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保持するために、バージョニングを有効にします。詳細については、「バケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
- 使用状況のモニタリング — バケットに格納されているオブジェクトの数と、使用されているストレージ領域の量を監視します。詳細については、「バケットメトリクスを表示」を参照してください。
- ストレージプランを変更する — バケットが過剰に使用されている場合はアップサイズを、使用されていない場合はダウンサイズします。詳細については、「バケットのプランの変更」を参照してください。
- バケットを接続する — Lightsail バケットを WordPress ウェブサイトに接続して、ウェブサイトの画像と添付ファイルを保存します。バケットを、Lightsail コンテンツ配信ネットワーク (CDN) ディストリビューションのオリジンとして指定します。これにより、世界中のユーザーへのバケット内のオブジェクトの配信が高速化されます。詳細については、「チュートリアル: バケットを WordPress インスタンスに接続する」および「チュートリアル: バケットをコンテンツ配信ネットワークディストリビューションと使用する」を参照してください。
- バケットの削除 — 使用しなくなったバケットを削除します。詳細については、「バケットの削除」を参照してください。

オブジェクトストレージ用の Lightsail バケットを作成する

クラウドへのファイルのアップロードを開始する準備が整ったら、Amazon Lightsailオブジェクトストレージサービスでバケットを作成します。Lightsail オブジェクトストレージサービスにアップロードしたすべてのファイルは Lightsail バケットに保存されます。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

バケットを作成する

Lightsail バケットを作成する手順は以下のとおりです。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. [バケットを作成] を選択します。
4. [AWS リージョンの変更] を選択して、バケットを作成するリージョンを選択します。

   バケットで使用する予定のリソース AWS リージョン と同じ にバケットを作成することをお勧めします。作成後にバケットのリージョンを変更することはできません。
5. バケットのストレージプランを選択します。

   ストレージプランでは、バケットの月額コスト、ストレージ領域のクォータ、データ転送クォータを指定します。

   バケットのプランは、毎月の AWS 請求サイクル内で 1 回だけ変更できます。バケットがストレージ領域またはデータ転送クォータを一貫して上回っている場合、またはバケットの使用量がストレージ領域またはデータ転送クォータより低い範囲にある場合、バケットのプランを変更します。詳細については、「バケットのプランを変更する」を参照してください。
6. バケットの名前を入力します。

   バケット名の詳細については、「Amazon Lightsail でのバケット命名規則」を参照してください。
7. [Create bucket] (バケットの作成) を選択します。

   新しいバケットの管理ページにリダイレクトされます。バケットを使用および管理するための追加ドキュメントについては、このガイドの「次のステップ」セクションに進んでください。

バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。

3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

   バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

   - Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
   - Amazon Lightsail でのバケットのアクセス許可の設定
   - Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
   - Amazon Lightsail でのバケットのアクセスキーの作成
   - Amazon Lightsail でのバケットのリソースアクセスの設定
   - Amazon Lightsail でのバケットのクロスアカウントアクセスの設定
5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。

   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定
6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。

   - Amazon Lightsail のバケットにファイルをアップロードする

- マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
- Amazon Lightsail のバケット内のオブジェクトの表示
- Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
- Amazon Lightsail のバケットからのオブジェクトのダウンロード
- Amazon Lightsail のバケット内のオブジェクトのフィルタリング
- Amazon Lightsail のバケット内のオブジェクトのタグ付け
- Amazon Lightsail のバケット内のオブジェクトの削除

9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。
11. バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。
12. バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。
13. ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。
14. バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。
    - チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
    - チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用
15. 使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

Lightsail オブジェクトストレージバケットを削除する

Amazon Lightsail オブジェクトストレージサービスで使用していないバケットを削除します。バケットを削除すると、保存されたバージョンのオブジェクトやアクセスキーなど、バケット内のすべてのオブジェクトが完全に削除されます。

バケットについての詳細は、「オブジェクトストレージ」を参照してください。

バケットの強制削除

次のいずれかの条件を持つバケットは、削除を承認しない限り、削除できません。

- ディストリビューションのオリジンのバケット。
- インスタンスが添付されたバケット。
- オブジェクトがあるバケット。
- アクセスキーがあるバケット。

バケットに依存する既存のワークフローが中断されないように、削除を承認する必要があります。たとえば、バケットにメディアを保存している WordPress ウェブサイトや、バケット内のオブジェクトをキャッシュして提供しているディストリビューションなどが該当します。

前述の条件のいずれかを持つバケットの削除を承認するには、バケットを強制的に削除する必要があります。バケットを削除する前に、Lightsail サービスはこれらのどの条件が該当するか、プロンプトで表示します。Lightsail コンソールを使用してバケットを削除すると、バケットを強制的に削除するオプションが表示されます。を使用する場合は AWS CLI、`delete-bucket`リクエストを行うときに `--force-delete`フラグを指定する必要があります。これらの手順については、このガイドの「Lightsail コンソールを使用してバケットを削除する」および「 AWS CLIを使用してバケットを削除する」のセクションを参照してください。

Lightsail コンソールを使用してバケットを削除する

Lightsail コンソールを使用してバケットを削除するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. 削除するバケットの名前を選択します。

4. タブメニューの省略記号 (:) アイコンを選択し、そして [削除] を選択します。
5. [Delete Bucket (バケットを削除)] を選択します。
6. 表示されるプロンプトで、バケットが次の条件のいずれかを満たしているかどうかを確認します。

   - オブジェクトを含む
   - アクセスキーを含む
   - インスタンスに添付されている
   - ディストリビューションのオリジンである

   これらの条件のいずれかが該当する場合は、バケットを強制的に削除するように選択する必要があります。
7. 以下のオプションのいずれかを選択してください。

   - [強制削除] を選択することで、この手順のステップ 6 の条件を有していてもバケットを削除することができます。
   - ステップ 6 に記された条件を有していない場合は、「はい、削除します」を選択してバケットを削除します。
   - 「いいえ、キャンセルします」を選択して削除をキャンセルします。

を使用してバケットを削除する AWS CLI

AWS Command Line Interface () を使用してバケットを削除するには、次の手順を実行しますAWS CLI。これは、`delete-bucket` コマンドを使用して実行できます。詳細については、「AWS CLI コマンドリファレンス」の「delete-bucket」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. コマンドプロンプトまたはターミナルウィンドウで、次のいずれかのコマンドを入力します。

- 本ガイドの[「バケットの強制削除」セクションに記されている条件が当てはまらないバケットを削除するためには、以下のコマンドを入力します。

```
aws lightsail delete-bucket --bucket-name BucketName
```

- 本ガイドの「バケットの強制削除」セクションに記されている条件が当てはまるバケットを削除するためには、以下のコマンドを入力します。

```
aws lightsail delete-bucket --bucket-name BucketName --force-delete
```

コマンドで、*BucketName* を削除するバケットの名前に置き換えます。

例:

```
aws lightsail delete-bucket --bucket-name amzn-s3-demo-bucket
```

以下の例のような結果が表示されるはずです。

```
C:\>aws lightsail delete-bucket --bucket-name DOC-EXAMPLE-BUCKET
{
    "operations": [
        {
            "id": "6example-4d30-4442-ae9a-examplef4f52",
            "resourceName": "DOC-EXAMPLE-BUCKET",
            "resourceType": "Bucket",
            "createdAt": "2021-06-30T13:42:43.873000-07:00",
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "62example362/DOC-EXAMPLE-BUCKET/small_1_0",
            "operationType": "DeleteBucket",
            "status": "Succeeded",
            "statusChangedAt": "2021-06-30T13:42:43.873000-07:00",
            "errorCode": "",
            "errorDetails": ""
        }
    ]
}
```

バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。
3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

   バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

   - Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
   - Amazon Lightsail でのバケットのアクセス許可の設定
   - Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
   - Amazon Lightsail でのバケットのアクセスキーの作成
   - Amazon Lightsail でのバケットのリソースアクセスの設定
   - Amazon Lightsail でのバケットのクロスアカウントアクセスの設定
5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。

   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定

6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail のバケットにファイルをアップロードする
   - マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
   - Amazon Lightsail のバケット内のオブジェクトの表示
   - Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
   - Amazon Lightsail のバケットからのオブジェクトのダウンロード
   - Amazon Lightsail のバケット内のオブジェクトのフィルタリング
   - Amazon Lightsail のバケット内のオブジェクトのタグ付け
   - Amazon Lightsail のバケット内のオブジェクトの削除
9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。
11. バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。
12. バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。
13. ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。
14. バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。
   - チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
   - チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用

15.使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

Lightsail オブジェクトストレージバケットのアクセスキーを作成する

アクセスキーを使用して、バケットとそのオブジェクトへのフルアクセスを付与する一連の認証情報を作成できます。アクセスキーは、アクセスキー ID とシークレットアクセスキーとのセットで構成されます。シークレットアクセスキーは、作成時にのみ使用できます。ソフトウェアまたはプラグインでアクセスキーを設定すると、 AWS APIs および AWS SDKs を使用してバケットへの完全な読み取りおよび書き込みアクセスを持つことができます。 AWS CLIでアクセスキーを設定することもできます。

> **Important**
>
> バケットごとに 2 つのアクセスキーを持つことができますが、一度に 1 つのバケットアクセスキーのみ作成することをお勧めします。定期的にキーをローテーションして、既存のキーのインベントリを作成することもお勧めします。シークレットアクセスキーがコピー、紛失、侵害された場合、アクセスキーを削除して新しいキーを作成する必要があります。バケットアクセスキーをローテーションするベストプラクティスの詳細については、「バケットのアクセスキーをローテーションする」を参照してください。

許可のオプションの詳細については、「バケットのアクセス許可」を参照してください。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

バケットのアクセスキーを作成する

バケットのアクセスキーを作成するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. アクセス権限を設定するバケットの名前を選択します。
4. [Permissions (許可)] タブを選択します。

   ページのアクセスキーセクションには、バケットの既存のアクセスキー（存在する場合）が表示されます。

5. バケットの新しいキーを作成するには、[Create access key (アクセスキーの作成)] を選択します。
6. 表示されるプロンプトで、「はい、作成します」を選択し、新しいアクセスキーの作成を確定します。それ以外の場合は、[キャンセル] を選択します。
7. 表示される成功プロンプトで、アクセスキー ID を書き留めます。
8. [Show secret access key (シークレットアクセスキーを表示)] を選択してシークレットアクセスキーを表示し、それをメモします。シークレットアクセスキーは再度表示されることはありません。

> **Important**
>
> アクセスキー ID とシークレットアクセスキーは安全な場所に保存します。これらが漏洩された場合は、削除し、新しいキーを作成する必要があります。詳細については、「Lightsail オブジェクトのストレージバケットのアクセスキーを削除します」を参照してください。

9. [Continue (続行)] を選択して終了します。

新しいアクセスキーはページのアクセスキーのセクションで操作します。アクセスキーが漏洩された場合、または紛失した場合は、キーを削除し、新しいキーを作成します。

> **Note**
>
> 各アクセスキーの隣に表示される [最終使用日] の列は、キーが最後に使用されたのがいつかを示します。キーが使用されていない場合は、ダッシュが表示されます。アクセスキーノードを展開して、サービスおよびキーが最後に使用された AWS リージョン 場所を表示します。

Lightsail オブジェクトのストレージバケットのアクセスキーを削除します

アクセスキーは、バケットとそのオブジェクトへのフルアクセスを付与する一連の認証情報です。アクセスキーは、アクセスキー ID とシークレットアクセスキーがセットです。シークレットアクセスキーがコピー、紛失、侵害された場合、アクセスキーを削除し、アクセスキーを削除する必要があります。

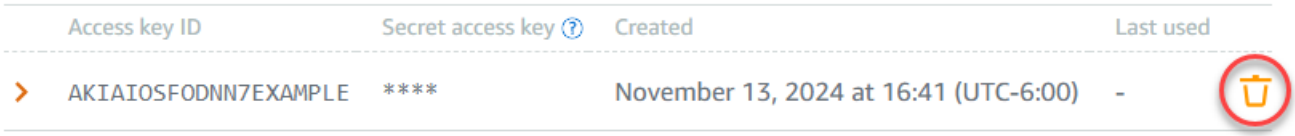
バケットのアクセスキーを削除します

次の手順でバケットアクセスキーを削除できます。

> **Warning**
> アクセスキーを削除すると、永久に削除されるため、再度取得することはできません。新しいアクセスキーでのみ置き換えることができます。

既存の Lightsail オブジェクトストレージバケットのアクセスキーを削除する方法

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. アクセスキーを削除するバケットの名前を選択します。
4. [アクセス許可] タブを選択します。
5. [アクセスキー] で、削除するアクセスキーの削除アイコンを選択します。



6. [はい、削除します] を選択し、アクセスキーの削除を続行します。

既存のキーが削除されたら、新しいアクセスキーを作成してソフトウェアまたはプラグイン用に設定できます。詳細については、「バケットのアクセスキーをローテーションする」を参照してください。

Lightsail バケットとオブジェクトへのパブリックアクセスを制限する

Amazon Simple Storage Service (Amazon S3) は、お客様がデータを保存して保護することができるオブジェクトストレージサービスです。Amazon Lightsail オブジェクトストレージサービスは Amazon S3 テクノロジーに基づいて構築されています。Amazon S3 はアカウントレベルのブロックパブリックアクセスを提供しており、これを使用して AWS アカウント内のすべての S3 バケットへのパブリックアクセスを制限できます。アカウントレベルのブロックパブリックアクセスは、既存の

個々のバケットとオブジェクトのアクセス許可に関係なく、 内のすべての S3 バケットを AWS アカウント プライベートにすることができます。

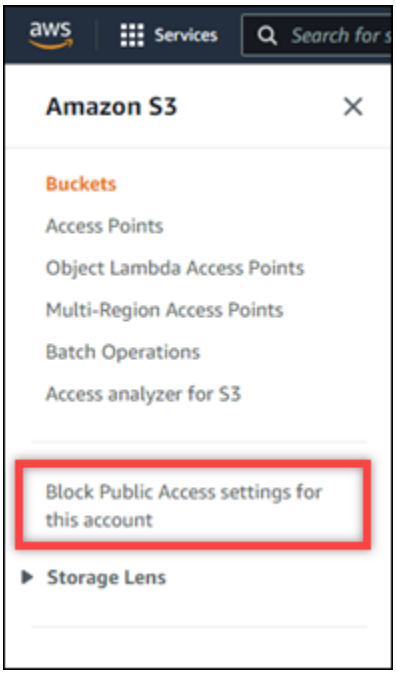
パブリックアクセスを許可または拒否する場合、Lightsail オブジェクトストレージバケットは次のことを考慮します。

- Lightsail バケットアクセス許可。詳細については、「バケットのアクセス許可」を参照してください。
- Amazon S3 アカウントレベルのパブリックアクセスのブロック設定。Lightsail バケットアクセス許可を上書きします。

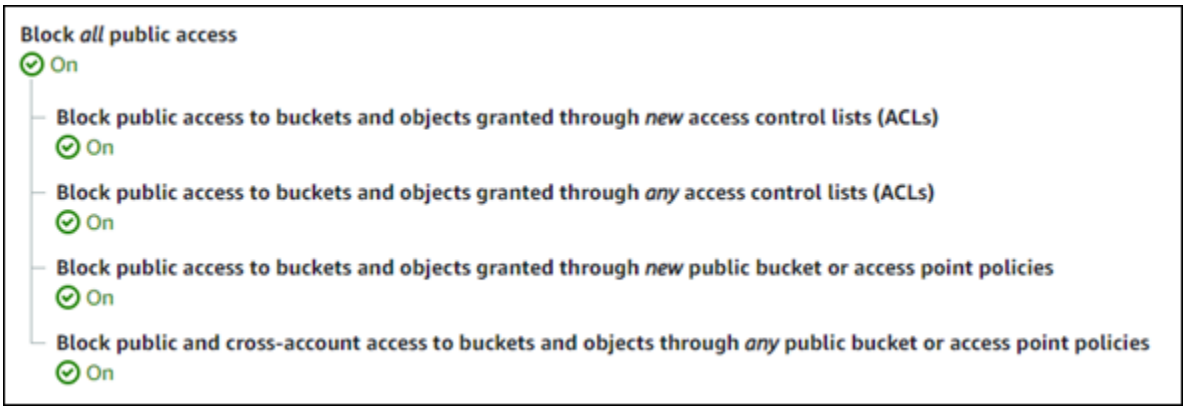
Amazon S3 でアカウントレベルの [すべてのパブリックアクセスのブロック] をオンにした場合、パブリックの Lightsail バケットおよびオブジェクトがプライベートになり、公開アクセスができなくなります。

アカウントのブロックパブリックアクセス設定の構成

Amazon S3 コンソール、 AWS Command Line Interface (AWS CLI)、 AWS SDKs、および REST API を使用して、パブリックアクセスのブロックを設定できます。次の例に示すように、Amazon S3 コンソールのナビゲーションペインでアカウントレベルのパブリックアクセスのブロック機能にアクセスできます。



Amazon S3 コンソールには、すべてのパブリックアクセスのブロック、新しいまたは任意のアクセスコントロールリストを通じて付与されたパブリックアクセスのブロック、新しいまたは任意のパブリックバケットまたはアクセスポイントポリシーを通じて付与されたバケットおよびオブジェクトへのパブリックアクセスのブロックの設定があります。



Amazon S3 コンソールで各設定を [オン] または [オフ] にできます。API では、対応する設定は TRUE (オン) または FALSE (オフ) です。次のセクションでは、S3 バケットと Lightsail バケットに対する各設定の影響について説明します。

> **Note**
>
> 次のセクションでは、アクセスコントロールリスト (ACL) について説明します。ACL は、バケットまたは個々のオブジェクトを所有している、またはそれらにアクセスできるユーザーを定義します。詳細については、「Amazon S3 ユーザーガイド」の「アクセスコントロールリストの概要」を参照してください。

- すべてのパブリックアクセスのブロック - この設定をオンにすると、S3 バケット、Lightsail バケット、それぞれの対応するオブジェクトへのパブリックアクセスがすべてブロックされます。この設定には、次の設定がすべて組み込まれています。この設定をオンにすると、あなた (バケット所有者) と許可されたユーザーのみが、バケットとそのオブジェクトにアクセスできます。この設定は、Amazon S3 コンソールでのみオンにできます。これは、 AWS CLI、Amazon S3 API、または SDK では使用できません。 AWS SDKs
  - 新しいアクセスコントロールリスト (ACL) を通じて付与されたバケットおよびオブジェクトへのパブリックアクセスをブロック — この設定をオンにすると、バケットおよびオブジェクトに対するパブリック ACL の配置がブロックされます。この設定は、既存の ACL には影響しません。したがって、既にパブリック ACL を持つオブジェクトはパブリックのままとなります。この設定は、バケットアクセス許可が [All objects are public and read-only] (すべてのオブジェクト

がパブリックかつ読み取り専用) に設定されているため、パブリックであるオブジェクトに影響を与えることもありません。この設定は、Amazon S3 API で `BlockPublicAcls` としてラベル付けされています。

> **Note**
>
> Offload Media Light プラグインなど、メディアを Lightsail バケットに配置する WordPress プラグインは、この設定をオンにすると動作を停止する場合があります。これは、ほとんどの WordPress プラグインがオブジェクトでパブリック読み取り ACL を設定するためです。オブジェクト ACL を切り替える WordPress プラグインも動作を停止する可能性があります。

- すべてのアクセスコントロールリスト (ACL) を通じて付与されたバケットおよびオブジェクトへのパブリックアクセスをブロック — この設定をオンにすると、パブリック ACL が無視され、バケットおよびオブジェクトへのパブリックアクセスがブロックされます。この設定では、パブリック ACL をバケットとオブジェクトに配置できますが、アクセス権を付与するときは無視されます。Lightsail バケットの場合、バケットのアクセス許可を [All objects are public and read-only] (すべてのオブジェクトがパブリックかつ読み取り専用) に設定するか、または個別のオブジェクトの許可を [Public (read-only)] (パブリック (読み取り専用)) に設定することは、パブリック ACL をいずれかに配置することと同等です。この設定は、Amazon S3 API で `IgnorePublicAcls` としてラベル付けされています。
- 新しいパブリックバケットまたはアクセスポイントポリシーを通じて付与されたバケットとオブジェクトへのパブリックアクセスのブロック — この設定をオンにすると、[すべてのオブジェクトをパブリックと読み取り専用にする] のバケットアクセス許可が、Lightsail バケットで設定されないようにブロックします。この設定は、[All objects are public and read-only] (すべてのオブジェクトがパブリックかつ読み取り専用) のバケットアクセス許可で既に設定されているバケットには影響しません。この設定は、Amazon S3 API で `BlockPublicPolicy` としてラベル付けされています。
- あらゆるパブリックバケットポリシーまたはアクセスポイントポリシーを介したバケットとオブジェクトへのパブリックアクセスおよびクロスアカウントアクセスをブロック — この設定をオンにすると、すべての Lightsail バケットがプライベートになります。これにより、[All objects are public and read-only] (すべてのオブジェクトがパブリックかつ読み取り専用) のバケットアクセス許可で設定されている場合でも、すべての Lightsail バケットがプライベートになります。この設定は、Amazon S3 API で `RestrictPublicBuckets` としてラベル付けされています。

> **Important**
>
> この設定は、Lightsail バケットで設定されているクロスアカウントアクセス (Lightsail で [All objects are public and read-only] (すべてのオブジェクトがパブリックかつ読み取り専用) のバケットアクセス許可でも設定されているもの) もブロックします。クロスアカウントアクセスを引き続き許可するには、Amazon S3 で [あらゆるパブリックバケットポリシーまたはアクセスポイントポリシーを介したバケットとオブジェクトへのパブリックアクセスおよびクロスアカウントアクセスをブロック] 設定をオンにする前に、必ず Lightsail で [すべてのオブジェクトをプライベートにする] のバケットアクセス許可を Lightsail バケットに設定してください。

ブロックパブリックアクセスとその設定方法の詳細については、「Amazon S3 ユーザーガイド」で以下のリソースを参照してください。

- Amazon S3 ストレージへのパブリックアクセスのブロック
- アカウントのブロックパブリックアクセス設定の構成

Lightsail コンソール、 AWS CLI、 AWS SDKs、および REST API を使用して、Lightsailバケットのアクセス許可を設定します。詳細については、「バケットのアクセス許可」を参照してください。

> **Note**
>
> Lightsail は、Amazon S3 から現在のアカウントレベルのブロックパブリックアクセス設定を取得し、それを Lightsail オブジェクトストレージリソースに適用するためにサービスリンクロールを使用します。Amazon S3 でパブリックアクセスのブロックを設定した後、Lightsail で有効になるまで少なくとも 1 時間待機します。詳細については、「サービスにリンクされたロール」を参照してください。

バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。

2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。
3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

   バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

   - Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
   - Amazon Lightsail でのバケットのアクセス許可の設定
   - Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
   - Amazon Lightsail でのバケットのアクセスキーの作成
   - Amazon Lightsail でのバケットのリソースアクセスの設定
   - Amazon Lightsail でのバケットのクロスアカウントアクセスの設定
5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。

   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定
6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。

- Amazon Lightsail のバケットにファイルをアップロードする
- マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
- Amazon Lightsail のバケット内のオブジェクトの表示
- Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
- Amazon Lightsail のバケットからのオブジェクトのダウンロード
- Amazon Lightsail のバケット内のオブジェクトのフィルタリング
- Amazon Lightsail のバケット内のオブジェクトのタグ付け
- Amazon Lightsail のバケット内のオブジェクトの削除

9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。

10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。

11. バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。

12. バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。

13. ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。

14. バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。

- チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
- チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用

15. 使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

アクセスログを使用してオブジェクトストレージバケットリクエストを追跡する

アクセスログは、Amazon Lightsail オブジェクトストレージサービスのバケットに対して行われたリクエストの詳細なレコードを提供します。この情報には、リクエストタイプ、リクエストで指定されたリソース、リクエストが処理された日時が含まれます。アクセスのログは、多くのアプリケーションに役立ちます。例えば、アクセスのログ情報は、セキュリティやアクセスの監査に役立ちます。また、顧客基盤について知るうえでも役立ちます。

目次

- ログ配信を有効にするには何が必要ですか
- ログオブジェクトのキーフォーマット
- ログを配信する方法
- ベストエフォート型のアクセスログ配信
- バケットのログ記録ステータスの変更が有効になるまでには時間がかかる

ログ配信を有効にするには何が必要ですか

ログ配信を有効にする前に、次の点を考慮してください。詳細は、「バケットアクセスのログ記録を有効にする」を参照してください。

1. ログのターゲットバケットを特定します。このバケットは、Lightsail がアクセスログをオブジェクトとして保存する場所です。ソースバケットとターゲットバケットの両方が同じ AWS リージョンにあり、同じアカウントによって所有されている必要があります。

   ログの保存先のバケットとして、ソースバケットと同じリージョンにあるユーザー所有のバケットを指定できます。これにはソースバケット自体も含まれます。ただし、ログを管理しやすくするため、アクセスログは別のバケットに保存することをお勧めします。

   ソースバケットとターゲットバケットが同じである場合、バケットに書き込まれるログに関する追加のログが作成されます。これは、ストレージの消費がいくらか増える可能性があるため、望ましくない場合があります。また、ログに関する追加のログのために、必要なログを見つけにくくなります。アクセスログの保存先をソースバケットにする場合は、ログオブジェクトを簡単に区別できるように、すべてのログオブジェクトキーにプレフィックスを指定し、オブジェクト名を共通の文字列で始めてください。キープレフィックスは、複数のバケットが同じターゲットバケットにログを記録する場合に、ソースバケットを区別するためにも役立ちます。

2. (オプション) ログオブジェクトキーのプレフィックスを特定します。プレフィックスを使用すると、ログオブジェクトを見つけやすくなります。例えば、プレフィックスの値として logs/ を指定すると、Lightsail で作成する各ログオブジェクトのキーの先頭に logs/ というプレフィックスが付きます。プレフィックスの末尾であることを示すには、末尾のスラッシュ / が必要です。logs/ プレフィックス付きのログオブジェクトキーの例を次に示します。

```
logs/2021-11-31-21-32-16-E568B2907131C0C0
```

ログオブジェクトのキーフォーマット

Lightsail では、ターゲットバケットにログオブジェクトをアップロードする際に、以下のオブジェクトキーフォーマットを使用します。

```
TargetPrefix/YYYY-mm-DD-HH-MM-SS-UniqueString
```

このキーで、YYYY、mm、DD、HH、MM、SS は、ログファイルを配信した年、月、日、時、分、秒をそれぞれ表す数字です。これらの日付と時刻は協定世界時 (UTC) です。

ある時点で配信されたログファイルには、その時点より前に書き込まれたレコードが含まれます。特定の期間のすべてのログレコードが配信されたかどうかを知る方法はありません。

キーの UniqueString コンポーネントは、ファイルの上書きを防止するためのものです。意味はないため、ログ処理ソフトウェアでは無視されます。

ログを配信する方法

Lightsail は、アクセスログレコードを定期的に収集してログファイルにまとめ、そのログファイルをログオブジェクトとしてターゲットバケットにアップロードします。複数のソースバケットでログ記録の配信先が同じターゲットバケットである場合、これらのすべてのソースバケットのアクセスログがターゲットバケットに収容されます。ただし、各ログオブジェクトは、ソースバケット別にアクセスログレコードをレポートします。

ベストエフォート型のアクセスログ配信

アクセスログレコードの配信は、ベストエフォートで行われます。ログ記録用に適切にバケットを設定した場合、そのバケットへのほとんどのリクエストについてログレコードが配信されます。ほとんどのログレコードは、記録された時間から数時間以内に配信されますが、配信間隔は短くなる場合もあります。

アクセスのログ記録の完全性や適時性は保証されません。リクエストのログレコードが、リクエストが実際に処理されてからかなり後に配信されたり、配信すらされなかったりすることもあり得ます。アクセスログの目的は、バケットに対するトラフィックの特性を理解することです。ログレコードが失われることはまれですが、すべてのリクエストが完全に報告されるとは限りません。

バケットのログ記録ステータスの変更が有効になるまでには時間がかかる

バケットのログ記録ステータスの変更がログファイルの配信に反映されるまでには時間がかかります。例えば、バケットのログを有効にする場合、その後数時間に行われるリクエストは記録されることもあれば、されないこともあります。ログ記録のターゲットバケットをバケット A からバケット B に変更すると、その後 1 時間は一部のログがバケット A に引き続き配信されたり、新しいターゲットバケット B に配信されたりします。いずれにしても、最終的に新しい設定が有効になるため、ユーザー側の操作は一切不要です。

トピック

- Lightsail バケットログを使用してオブジェクトストレージアクセスを分析する
- Lightsail のバケットアクセスログ記録の有効化
- Lightsail で Amazon Athena を使用してバケットアクセスログを分析する

Lightsail バケットログを使用してオブジェクトストレージアクセスを分析する

アクセスログは、Amazon Lightsail オブジェクトストレージサービスのバケットに対して行われたリクエストの詳細なレコードを提供します。アクセスのログ記録を使用して、セキュリティとアクセスを監査し、顧客ベースを確認することができます。このセクションでは、アクセスログファイルの形式およびその他の詳細について説明します。ログ記録の基本の詳細については、「バケットのアクセスログ」を参照してください。

アクセスのログファイルは、一連のログレコードを改行で区切って構成します。各ログレコードは 1 個のリクエストを表し、各フィールドをスペースで区切って構成します。

次に示すのは、5 個のログレコードで構成されるログの例です。

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 3E57427F3EXAMPLE
 REST.GET.VERSIONING - "GET /amzn-s3-demo-bucket?versioning HTTP/1.1" 200 - 113 - 7 -
```

```
 "-" "S3Console/0.4" - s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/
XV/VLi31234= SigV2 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-demo-bucket.s3.us-
west-1.amazonaws.com TLSV1.1
```

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 891CE47D2EXAMPLE
 REST.GET.LOGGING_STATUS - "GET /amzn-s3-demo-bucket?logging HTTP/1.1" 200 -
 242 - 11 - "-" "S3Console/0.4" - 9vKBE6vMhrNiWHZmb2L0mXOcqPGzQOI5XLnCtZNPxev+Hf
+7tpT6sxDwDty4LHBUOZJG96N1234= SigV2 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-
demo-bucket.s3.us-west-1.amazonaws.com TLSV1.1
```

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket [06/Feb/2019:00:00:38 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be A1206F460EXAMPLE
 REST.GET.BUCKETPOLICY - "GET /amzn-s3-demo-bucket?policy HTTP/1.1" 404
 NoSuchBucketPolicy 297 - 38 - "-" "S3Console/0.4" - BNaBsXZQQDbssi6xMBdBU2sLt
+Yf5kZDmeBUP35sFoKa3sLLeMC78iwEIWxs99CRUrbS4n11234= SigV2 ECDHE-RSA-AES128-GCM-SHA256
 AuthHeader amzn-s3-demo-bucket.s3.us-west-1.amazonaws.com TLSV1.1
```

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket [06/Feb/2019:00:01:00 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be 7B4A0FABBEXAMPLE
 REST.GET.VERSIONING - "GET /amzn-s3-demo-bucket?versioning HTTP/1.1" 200 -
 113 - 33 - "-" "S3Console/0.4" - Ke1bUcazaN1jWuUlPJaxF64cQVpUEhoZKEG/hmy/gijN/
I1DeWqDfFvnpybfEseEME/u7ME1234= SigV2 ECDHE-RSA-AES128-GCM-SHA256 AuthHeader amzn-s3-
demo-bucket.s3.us-west-1.amazonaws.com TLSV1.1
```

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 amzn-s3-demo-bucket [06/Feb/2019:00:01:57 +0000] 192.0.2.3
 79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
 DD6CC733AEXAMPLE REST.PUT.OBJECT s3-dg.pdf "PUT /amzn-s3-demo-bucket/
s3-dg.pdf HTTP/1.1" 200 - - 4406583 41754 28 "-" "S3Console/0.4" -
 10S62Zv81kBW7BB6SX4XJ48o6kpcl6LPwEoizZQQxJd5qDSCTLX0TgS37kYUBKQW3+bPdrg1234= SigV4
 ECDHE-RSA-AES128-SHA AuthHeader amzn-s3-demo-bucket.s3.us-west-1.amazonaws.com TLSV1.1
```

> **Note**
>
> 任意のログレコードフィールドを - (ダッシュ) に設定して、データが不明または使用不可であること、またはフィールドがこのリクエストに適用されなかったことを示すことができます。

目次

- ログレコードフィールド
- コピー操作の追加ログ記録
- カスタムアクセスログ情報
- 拡張可能なアクセスログの形式のプログラミングに関する考慮事項

ログレコードフィールド

次のリストには、ログレコードのフィールドが説明されています。

アクセスポイントの ARN (Amazon リソースネーム)

リクエストのアクセスポイントの Amazon リソースネーム (ARN) です。アクセスポイントの ARN の形式が不正、または使用されていない場合、このフィールドには「-」が含まれます。アクセスポイントの詳細については、「アクセスポイントを使用する」を参照してください。ARN の詳細については、AWS 全般リファレンスで「Amazon リソースネーム (ARN)」に関するトピックを参照してください。

エントリ例

```
arn:aws:s3:us-east-1:123456789012:accesspoint/example-AP
```

バケット所有者

ソースバケット所有者の正規ユーザー ID。正規ユーザー ID は、別の形式の AWS アカウント ID です。正規ユーザー ID の詳細については、「AWS 全般のリファレンス」の「AWS アカウント ID」を参照してください。アカウントの正規ユーザー ID を検索する方法については、「AWS アカウントの正規ユーザー ID の検索」を参照してください。

エントリ例

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

バケット

リクエストの処理ターゲットのバケットの名前。システムで受け取ったリクエストの形式に誤りがあり、バケットを特定できない場合、そのリクエストはアクセスログに表示されません。

エントリ例

```
amzn-s3-demo-bucket
```

時間

リクエストが受信された時間。これらの日付と時刻は協定世界時 (UTC) です。*strftime()* 用語を使用すると、形式は次のようになります: *[%d/%b/%Y:%H:%M:%S %z]*

エントリ例

```
[06/Feb/2019:00:00:38 +0000]
```

リモート IP

リクエスタの表面上のインターネットアドレス。中間プロキシやファイアウォールにより、リクエストを作成したマシンの実際のアドレスが不明確になる場合があります。

エントリ例

```
192.0.2.3
```

リクエスタ

リクエスタの正規ユーザー ID。認証されていないリクエストの場合は - です。リクエスタが IAM ユーザーの場合、このフィールドは、リクエスタの IAM ユーザー名と IAM ユーザーが所属する AWS ルートアカウントを返します。この識別子は、アクセスコントロールに使用されるものと同じです。

エントリ例

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

リクエスト ID

各リクエストを一意に識別するために Lightsail で生成される文字列。

エントリ例

```
3E57427F33A59F07
```

操作

ここに表示されているオペレーション
は、SOAP.*operation*、REST.*HTTP_method.resource_type*、WEBSITE.*HTTP_method.resource_t*
または BATCH.DELETE.OBJECT と表示されます。

エントリ例

```
REST.PUT.OBJECT
```

キー

リクエストの URL エンコードされた「key」部分、オペレーションがキーパラメータを取らない場合は「-」。

エントリ例

```
/photos/2019/08/puppy.jpg
```

Request-URI

HTTP リクエストメッセージの Request-URI の部分。

エントリ例

```
"GET /amzn-s3-demo-bucket/photos/2019/08/puppy.jpg?x-foo=bar HTTP/1.1"
```

HTTP ステータス

レスポンスの HTTP ステータスの数値。

エントリ例

```
200
```

エラーコード

Amazon S3 エラーコード、またはエラーが発生しなかった場合は「-」。

エントリ例

```
NoSuchBucket
```

送信バイト数

送信されたレスポンスのバイト数 (HTTP プロトコルオーバーヘッドを除きます)。ゼロの場合は「-」。

エントリ例

```
2662992
```

オブジェクトのサイズ

該当するオブジェクトの合計サイズ。

エントリ例

```
3462992
```

合計時間

バケットから見た、リクエストの転送の時間数 (ミリ秒単位)。これは、リクエストが受信されてから、レスポンスの最終バイトが送信されるまでの時間を計測した値です。クライアント側での計測値は、ネットワーク遅延により長くなる場合があります。

エントリ例

```
70
```

ターンアラウンド時間

Lightsail でリクエストの処理に要した時間数 (ミリ秒単位)。これは、リクエストの最終バイトが受信されてから、レスポンスの先頭バイトが送信されるまでの時間を計測した値です。

エントリ例

```
10
```

リファラー

HTTP Referer ヘッダーの値 (存在する場合)。一般に、HTTP ユーザーエージェント (ブラウザなど) は、このヘッダーをリクエスト作成時のリンクページや埋め込みページの URL に設定します。

エントリ例

```
"http://www.amazon.com/webservices"
```

ユーザーエージェント

HTTP User−Agent ヘッダーの値。

エントリ例

```
"curl/7.15.1"
```

バージョン ID

リクエストのバージョン ID、または オペレーションが `versionId` パラメータを取らない場合は「`-`」。

エントリ例

```
3HL4kqtJvjVBH40Nrjfkd
```

ホスト ID

x-amz-id-2 または Lightsail で拡張されたリクエスト ID。

エントリ例

```
s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

署名バージョン

署名バージョン SigV2 か SigV4 (リクエストの認証に使用)、または - (認証されていないリクエストの場合)。

エントリ例

```
SigV2
```

暗号スイート

HTTPS リクエストに対してネゴシエートされた Secure Sockets Layer (SSL) 暗号、または HTTP リクエストに対してネゴシエートされた -。

エントリ例

```
ECDHE-RSA-AES128-GCM-SHA256
```

認証タイプ

使用されるリクエスト認証のタイプ。認証ヘッダーは AuthHeader、クエリ文字列 (署名付き URL) は QueryString、認証されていないリクエストには -。

エントリ例

```
AuthHeader
```

ホストヘッダー

Lightsail への接続に使用するエンドポイント。

エントリ例

```
s3.us-west-2.amazonaws.com
```

TLS のバージョン

クライアントによってネゴシエートされた Transport Layer Security (TLS) バージョン。値は TLSv1、TLSv1.1、TLSv1.2、- のいずれかです (TLS を使用しなかった場合)。

エントリ例

```
TLSv1.2
```

コピーオペレーションの追加ログ記録

コピーオペレーションには GET と PUT が含まれます。このため、コピーオペレーションの実行時には 2 つのログレコードが記録されます。前述のセクションでは、コピーオペレーションの PUT 部分に関連するフィールドを説明しています。次のリストでは、コピーオペレーションの GET 部分に関連するフィールドを説明します。

バケット所有者

コピーされたオブジェクトを格納するバケットの正規ユーザー ID。正規ユーザー ID は、別の形式の AWS アカウント ID です。正規ユーザー ID の詳細については、「AWS 全般のリファレンス」の「AWS アカウント ID」を参照してください。アカウントの正規ユーザー ID を検索する方法については、「AWS アカウントの正規ユーザー ID の検索」を参照してください。

エントリ例

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

バケット

コピーターゲットのオブジェクトを格納するバケットの名前。

エントリ例

```
amzn-s3-demo-bucket
```

時間

リクエストが受信された時間。これらの日付と時刻は協定世界時 (UTC) です。strftime() terminology を使用した形式は次のようになります: [%d/%B/%Y:%H:%M:%S %z]

エントリ例

```
[06/Feb/2019:00:00:38 +0000]
```

リモート IP

リクエスタの表面上のインターネットアドレス。中間プロキシやファイアウォールにより、リクエストを作成したマシンの実際のアドレスが不明確になる場合があります。

エントリ例

```
192.0.2.3
```

リクエスタ

リクエスタの正規ユーザー ID。認証されていないリクエストの場合は - です。リクエスタが IAM ユーザーの場合、このフィールドは、リクエスタの IAM ユーザー名と IAM ユーザーが所属する AWS ルートアカウントと共に表示します。この識別子は、アクセスコントロールに使用されるものと同じです。

エントリ例

```
79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

リクエスト ID

各リクエストを一意に識別するために Lightsail で生成される文字列。

エントリ例

```
3E57427F33A59F07
```

操作

ここに表示されているオペレーション
は、SOAP.*operation*、REST.*HTTP_method.resource_type*、WEBSITE.*HTTP_method.resource_t*
または BATCH.DELETE.OBJECT と表示されます。

エントリ例

```
REST.COPY.OBJECT_GET
```

キー

コピーターゲットのオブジェクトのkey」部分。オペレーションがキーパラメータを取らない場合は「-」。

エントリ例

```
/photos/2019/08/puppy.jpg
```

Request-URI

HTTP リクエストメッセージの Request-URI の部分。

エントリ例

```
"GET /amzn-s3-demo-bucket/photos/2019/08/puppy.jpg?x-foo=bar"
```

HTTP ステータス

コピーオペレーションの GET 部分の HTTP ステータスの数値。

エントリ例

```
200
```

エラーコード

コピーオペレーションの GET 部分の Amazon S3 エラーコード、またはエラーがない場合は「-」。

エントリ例

```
NoSuchBucket
```

送信バイト数

送信されたレスポンスのバイト数 (HTTP プロトコルオーバーヘッドを除きます)。ゼロの場合は「-」。

エントリ例

```
2662992
```

オブジェクトのサイズ

該当するオブジェクトの合計サイズ。

エントリ例

```
3462992
```

合計時間

バケットから見た、リクエストの転送の時間数 (ミリ秒単位)。これは、リクエストが受信されてから、レスポンスの最終バイトが送信されるまでの時間を計測した値です。クライアント側での計測値は、ネットワーク遅延により長くなる場合があります。

エントリ例

```
70
```

ターンアラウンド時間

Lightsail でリクエストの処理に要した時間数 (ミリ秒単位)。これは、リクエストの最終バイトが受信されてから、レスポンスの先頭バイトが送信されるまでの時間を計測した値です。

エントリ例

```
10
```

リファラー

HTTP Referer ヘッダーの値 (存在する場合)。一般に、HTTP ユーザーエージェント (ブラウザなど) は、このヘッダーをリクエスト作成時のリンクページや埋め込みページの URL に設定します。

エントリ例

```
"http://www.amazon.com/webservices"
```

ユーザーエージェント

HTTP User-Agent ヘッダーの値。

エントリ例

```
"curl/7.15.1"
```

バージョン ID

コピー対象のオブジェクトのバージョン ID、または x-amz-copy-source ヘッダーでコピー元の一部として versionId パラメータを指定しなかった場合は「-」。

エントリ例

```
3HL4kqtJvjVBH40Nrjfkd
```

ホスト ID

x-amz-id-2 または Lightsail で拡張されたリクエスト ID。

エントリ例

```
s9lzHYrFp76ZVxRcpX9+5cjAnEH2ROuNkd2BHfIa6UkFVdtjf5mKR3/eTPFvsiP/XV/VLi31234=
```

署名バージョン

署名バージョン SigV2 か SigV4 (リクエストの認証に使用)、または - (認証されていないリクエストの場合)。

エントリ例

```
SigV2
```

暗号スイート

HTTPS リクエストに対してネゴシエートされた Secure Sockets Layer (SSL) 暗号、または HTTP リクエストに対してネゴシエートされた -。

エントリ例

```
ECDHE-RSA-AES128-GCM-SHA256
```

認証タイプ

使用されるリクエスト認証のタイプ。認証ヘッダーは AuthHeader、クエリ文字列 (署名付き URL) は QueryString、認証されていないリクエストには -。

エントリ例

```
AuthHeader
```

ホストヘッダー

Lightsail への接続に使用するエンドポイント。

エントリ例

```
s3.us-west-2.amazonaws.com
```

TLS のバージョン

クライアントによってネゴシエートされた Transport Layer Security (TLS) バージョン。値は TLSv1、TLSv1.1、TLSv1.2、- のいずれかです (TLS を使用しなかった場合)。

エントリ例

```
TLSv1.2
```

カスタムアクセスログ情報

リクエストのアクセスログレコードに保存するカスタム情報を含めることができます。これを行うには、リクエストの URL にカスタムクエリ文字列パラメータを追加します。Lightsail は「x-」で始まるクエリ文字列パラメータを無視しますが、ログレコードの Request-URI フィールドの一部として、リクエストのアクセスログレコードにそれらのパラメータを含めます。

例えば、GET の "s3.amazonaws.com/amzn-s3-demo-bucket/photos/2019/08/puppy.jpg?x-user=johndoe" リクエストは、"s3.amazonaws.com/amzn-s3-demo-bucket/photos/2019/08/puppy.jpg" のリクエストと同じように動作します。ただし 、"x-

user=johndoe" 文字列は関連付けられたログレコードの Request-URI フィールドに含まれている点が異なります。この機能は REST インターフェイスでのみ利用できます。

拡張可能なアクセスログの形式のプログラミングに関する考慮事項

場合によっては、新しいフィールドを各行末に追加することで、アクセスログレコードの形式を拡張することがあります。したがって、アクセスログを解析するコードは、理解できない可能性のある後続フィールドを処理するよう作成する必要があります。

Lightsail のバケットアクセスログ記録の有効化

アクセスログは、Amazon Lightsail オブジェクトストレージサービスのバケットに対して行われたリクエストの詳細なレコードを提供します。アクセスのログは、多くのアプリケーションに役立ちます。例えば、アクセスのログ情報は、セキュリティやアクセスの監査に役立ちます。また、顧客基盤について知るうえでも役立ちます。

デフォルトでは、Lightsail によってバケットへのアクセスのログは収集されません。ログ記録を有効にすると、Lightsail は、ソースバケットのアクセスログを選択されたターゲットバケットに配信します。ソースバケットとターゲットバケットの両方が同じ にあり AWS リージョン 、同じアカウントによって所有されている必要があります。

アクセスログのレコードには、バケットに対するリクエストの詳細が取り込まれます。この情報には、リクエストタイプ、リクエストで指定されたリソース、リクエストが処理された日時が含まれます。このガイドでは、 Lightsail API、 AWS Command Line Interface (AWS CLI)、または AWS SDKs を使用してバケットのアクセスログ記録を有効または無効にする方法を示します。

ログ記録の基本の詳細については、「バケットのアクセスログ」を参照してください。

目次

- アクセスログ記録のコスト
- AWS CLIを使用してアクセスログ記録を有効にする
- AWS CLIを使用してアクセスログ記録を無効にする

アクセスログ記録のコスト

バケットに対してアクセスのログ記録を有効にしても追加料金はかかりません。ただし、システムがバケットに配信するログファイルのストレージ領域は消費されます。ログはいつでも削除できます。

ログバケットのデータ転送が設定された月額許容範囲内にある場合、ログファイルの配信に対してデータ転送料金はかかりません。

ターゲットバケットでアクセスのログ記録が有効になっていない必要があります。ログの保存先のバケットとして、ソースバケットと同じリージョンにあるユーザー所有のバケットを指定できます。これにはソースバケット自体も含まれます。ただし、ログを管理しやすくするため、アクセスログは別のバケットに保存することをお勧めします。

を使用してアクセスログ記録を有効にする AWS CLI

バケットのアクセスログ記録を有効にするには、バケット AWS リージョン がある各 に専用のログ記録バケットを作成することをお勧めします。その後、 アクセスログをその専用のロギングバケットに配信します。

AWS CLIを使用してアクセスログ記録を有効にするには、次の手順を実行します。

> **Note**
>
> この手順を続行するLightsail前に、 をインストール AWS CLI し、 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ローカルコンピュータでコマンドプロンプトまたはターミナルウィンドウを開きます。
2. 次のコマンドを入力して、アクセスのログ記録を有効にします。

```
aws lightsail update-bucket --bucket-name SourceBucketName --access-log-config
 "{\"enabled\": true, \"destination\": \"TargetBucketName\", \"prefix\":
 \"ObjectKeyNamePrefix/\"}"
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

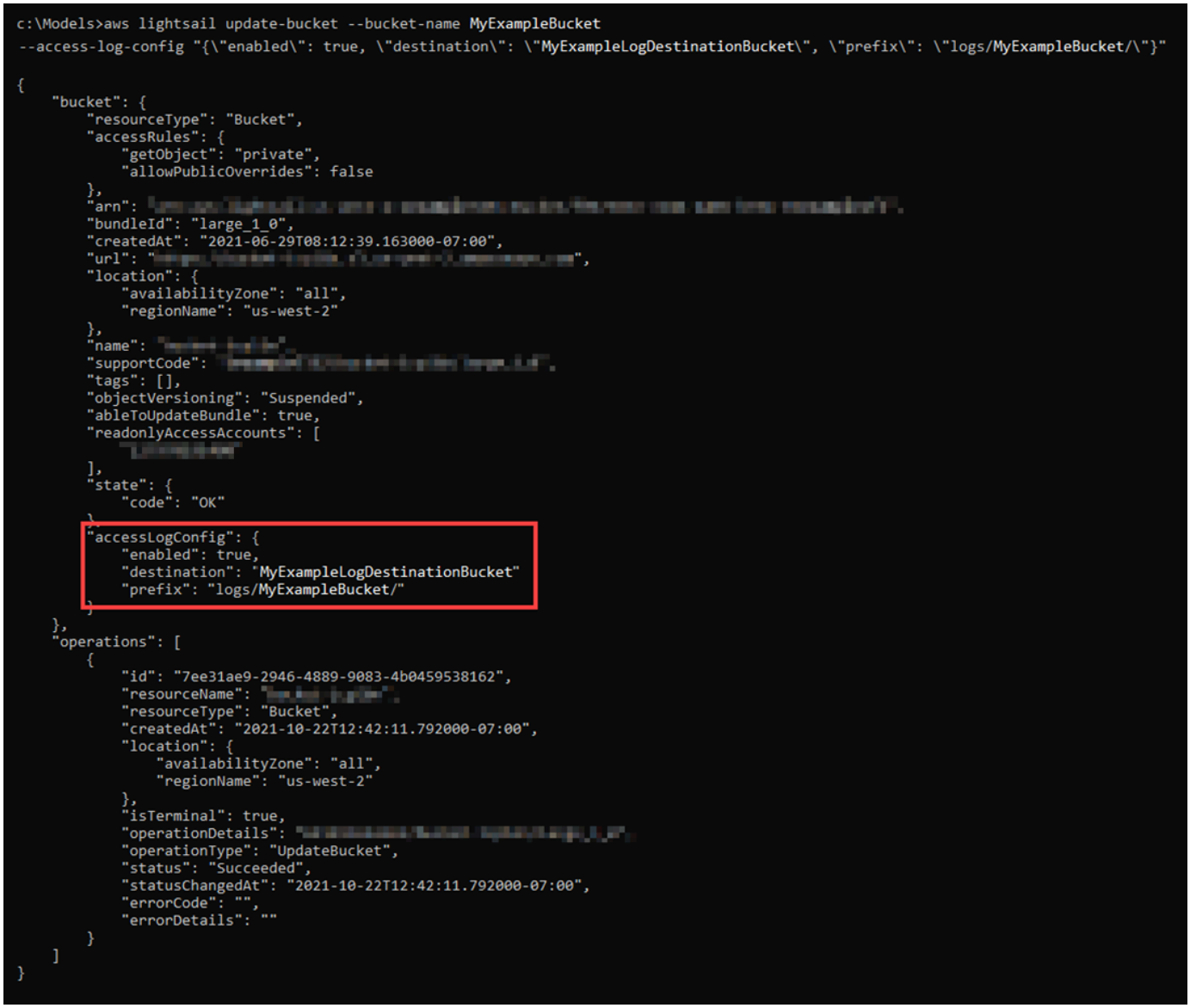
- *sourceBucketName* - アクセスログが作成されるソースバケットの名前。
- *TargetBucketName* - アクセスログを保存するターゲットバケットの名前。
- *ObjectKeyNamePrefix/* - アクセスログのオプションのオブジェクトキー名のプレフィックス。このプレフィックスは、スラッシュ (/) で終わる必要があります。

例

```
aws lightsail update-bucket --bucket-name amzn-s3-demo-bucket1 --access-log-config
 "{\"enabled\": true, \"destination\": \"amzn-s3-demo-bucket2\", \"prefix\":
 \"logs/amzn-s3-demo-bucket1/\"}"
```

この例では、*amzn-s3-demo-bucket1* はアクセスログを作成するソースバケット、*amzn-s3-demo-bucket2* はアクセスログを保存する送信先バケット、*logs/amzn-s3-demo-bucket1/* はアクセスログのオブジェクトキー名のプレフィックスです。

コマンドを実行すると、次の例のような結果が表示されます。ソースバケットが更新され、アクセスログの生成が開始され、保存先バケットに保存されます。



を使用したアクセスログ記録の無効化 AWS CLI

AWS CLIを使用してアクセスログ記録を無効にするには、次の手順を実行します。

> **Note**
>
> この手順を続行するLightsail前に、 をインストール AWS CLI し、 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ローカルコンピュータでコマンドプロンプトまたはターミナルウィンドウを開きます。
2. 次のコマンドを入力して、アクセスのログ記録を無効にします。

```
aws lightsail update-bucket --bucket-name SourceBucketName --access-log-config
 "{\"enabled\": false}"
```

コマンドで、*SourceBucketName* をアクセスログを無効にするソースバケットの名前に置き換えます。

例

```
aws lightsail update-bucket --bucket-name amzn-s3-demo-bucket --access-log-config
 "{\"enabled\": false}"
```

コマンドを実行すると、次の例のような結果が表示されます。

```
>aws lightsail update-bucket --bucket-name MyExampleBucket --access-log-config "{\"enabled\": false}"
{
    "bucket": {
        "resourceType": "Bucket",
        "accessRules": {
            "getObject": "private",
            "allowPublicOverrides": false
        },
        "arn": "
",
        "bundleId": "large_1_0",
        "createdAt": "2021-06-29T08:12:39.163000-07:00",
        "url":
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "name":
        "supportCode":
        "tags": [],
        "objectVersioning": "Suspended",
        "ableToUpdateBundle": true,
        "readonlyAccessAccounts": [
            "
"
        ],
        "state": {
            "code": "OK"
        },
        "accessLogConfig": {
            "enabled": false
        }
    },
    "operations": [
        {
            "id": "
",
            "resourceName":
            "resourceType": "Bucket",
            "createdAt": "2021-10-22T13:24:36.881000-07:00",
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails":
            "operationType": "UpdateBucket",
            "status": "Succeeded",
            "statusChangedAt": "2021-10-22T13:24:36.881000-07:00",
            "errorCode": "",
            "errorDetails": ""
        }
    ]
}
```

Lightsail で Amazon Athena を使用してバケットアクセスログを分析する

このガイドでは、アクセスログを使用したバケットへのリクエストの識別方法を説明します。詳細については、「バケットアクセスのログ」を参照してください。

目次

- Amazon Athena を使用してリクエストのアクセスログをクエリする
- Amazon S3 アクセスログを使用してオブジェクトアクセスリクエストの識別する

Amazon Athena を使用してリクエストのアクセスログをクエリする

Amazon Athena を使用して、アクセスログのバケットへのリクエストをクエリ、識別することができます。

Lightsail は、アクセスログをオブジェクトとして Lightsail バケットに保存します。多くの場合、ログを分析できるツールを使用する方が簡単です。Athena はオブジェクトの分析をサポートしてしるため、アクセスログに対してクエリを実行するのに使用できます。

例

次の例は、Amazon Athena でバケットサーバーアクセスログをクエリする方法を示しています。

> **Note**
>
> Athena クエリでバケットの場所を指定するには、ログが配信されるターゲットバケット名とターゲットプレフィックスを、次のような S3 URI でフォーマットする必要があります: `s3://amzn-s3-demo-bucket1-logs/prefix/`

1. https://console.aws.amazon.com/athena/ で Athena コンソールを開きます。
2. クエリエディタで、次のようなコマンドを実行します。

```
create database bucket_access_logs_db
```

> **Note**
>
> S3 バケット AWS リージョン と同じ にデータベースを作成するのがベストプラクティスです。

3. クエリエディタで、次のようなコマンドを実行して、ステップ 2 で作成したデータベースでテーブルスキーマを作成します。STRING および BIGINT データ型の値はアクセスログのプロパティです。これらのプロパティは Athena でクエリできます。LOCATION の場合は、前述のようにバケットとプレフィックスパスを入力します。

```
CREATE EXTERNAL TABLE `s3_access_logs_db.amzn-s3-demo-bucket_logs`(
  `bucketowner` STRING,
  `bucket_name` STRING,
```

```
  `requestdatetime` STRING,
  `remoteip` STRING,
  `requester` STRING,
  `requestid` STRING,
  `operation` STRING,
  `key` STRING,
  `request_uri` STRING,
  `httpstatus` STRING,
  `errorcode` STRING,
  `bytessent` BIGINT,
  `objectsize` BIGINT,
  `totaltime` STRING,
  `turnaroundtime` STRING,
  `referrer` STRING,
  `useragent` STRING,
  `versionid` STRING,
  `hostid` STRING,
  `sigv` STRING,
  `ciphersuite` STRING,
  `authtype` STRING,
  `endpoint` STRING,
  `tlsversion` STRING)
ROW FORMAT SERDE
  'org.apache.hadoop.hive.serde2.RegexSerDe'
WITH SERDEPROPERTIES (
  'input.regex'='([^ ]*) ([^ ]*) \\[(.*?)\\] ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 ([^ ]*) (\"[^\"]*\"|-) (-|[0-9]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*)
 (\"[^\"]*\"|-) ([^ ]*)(?: ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*) ([^ ]*))?.*$')
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://amzn-s3-demo-bucket1-logs/prefix/'
```

4. ナビゲーションペインにある、[データベース] で、データベースを選択します。
5. [テーブル] で、テーブル名の横にある、[Preview table (テーブルのプレビュー)] を選択します。

   [結果] ペインに、サーバーアクセスログのデータ (bucketowner、bucket、requestdatetime など) が表示されます。これは、Athena テーブルが正常に作成されたことを意味します。これで、バケットサーバーアクセスログのクエリを実行できます。

例 - 誰がいつオブジェクトを削除したか (タイムスタンプ、IP アドレス、および IAM ユーザー) を表示する

```
SELECT RequestDateTime, RemoteIP, Requester, Key
FROM s3_access_logs_db.amzn-s3-demo-bucket_logs
WHERE key = 'images/picture.jpg' AND operation like '%DELETE%';
```

例 - IAM ユーザーによって実行されたすべてのオペレーションを表示する

```
SELECT *
FROM s3_access_logs_db.amzn-s3-demo-bucket_logs
WHERE requester='arn:aws:iam::123456789123:user/user_name';
```

例 - 特定の期間にオブジェクトに対して実行されたすべてのオペレーションを表示する

```
SELECT *
FROM s3_access_logs_db.amzn-s3-demo-bucket_logs
WHERE Key='prefix/images/picture.jpg'
    AND parse_datetime(RequestDateTime,'dd/MMM/yyyy:HH:mm:ss Z')
    BETWEEN parse_datetime('2017-02-18:07:00:00','yyyy-MM-dd:HH:mm:ss')
    AND parse_datetime('2017-02-18:08:00:00','yyyy-MM-dd:HH:mm:ss');
```

例 - 特定の期間に特定の IP アドレスによって送信されたデータの量を表示する

```
SELECT SUM(bytessent) AS uploadTotal,
      SUM(objectsize) AS downloadTotal,
      SUM(bytessent + objectsize) AS Total
FROM s3_access_logs_db.amzn-s3-demo-bucket_logs
WHERE RemoteIP='1.2.3.4'
AND parse_datetime(RequestDateTime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2017-06-01','yyyy-MM-dd')
AND parse_datetime('2017-07-01','yyyy-MM-dd');
```

Amazon S3 アクセスログを使用してオブジェクトアクセスリクエストの識別する

アクセスログに対するクエリを使用して、GET、PUT、DELETE などのオペレーションに対するオブジェクトアクセスリクエストを識別し、それらのリクエストに関する詳細情報を確認することができます。

次の Amazon Athena クエリの例は、サーバーアクセスログからバケットに対するすべての PUT オブジェクトリクエストを取得する方法を示しています。

例 - 一定期間内に PUT オブジェクトリクエストを送信しているすべてのリクエスタを表示する

```
SELECT Bucket, Requester, RemoteIP, Key, HTTPStatus, ErrorCode, RequestDateTime
FROM s3_access_logs_db
WHERE Operation='REST.PUT.OBJECT' AND
parse_datetime(RequestDateTime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

次の Amazon Athena クエリの例は、サーバーアクセスログから Amazon S3 に対するすべての GET オブジェクトリクエストを取得する方法を示しています。

例 - 一定期間内に GET オブジェクトリクエストを送信しているすべてのリクエスタを表示する

```
SELECT Bucket, Requester, RemoteIP, Key, HTTPStatus, ErrorCode, RequestDateTime
FROM s3_access_logs_db
WHERE Operation='REST.GET.OBJECT' AND
parse_datetime(RequestDateTime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

次の Amazon Athena のクエリの例は、S3 バケットへのすべての匿名リクエストをサーバーアクセスログから取得する方法を示しています。

例 - 特定の期間にバケットにリクエストを行っているすべての匿名リクエスタを表示する

```
SELECT Bucket, Requester, RemoteIP, Key, HTTPStatus, ErrorCode, RequestDateTime
FROM s3_access_logs_db.amzn-s3-demo-bucket_logs
WHERE Requester IS NULL AND
parse_datetime(RequestDateTime,'dd/MMM/yyyy:HH:mm:ss Z')
BETWEEN parse_datetime('2019-07-01:00:42:42','yyyy-MM-dd:HH:mm:ss')
AND
parse_datetime('2019-07-02:00:42:42','yyyy-MM-dd:HH:mm:ss')
```

> **Note**
>
> - ニーズに合わせられるように、データ範囲を変更することができます。

- このクエリの例は、セキュリティのモニタリングにも役立つ場合があります。予期しないまたは不正な IP アドレス/リクエスタからの PutObject または GetObject コールの結果を確認し、バケットへの匿名リクエストを特定できます。
- このクエリでは、ログ記録が有効になった時間以降の情報のみ取得されます。

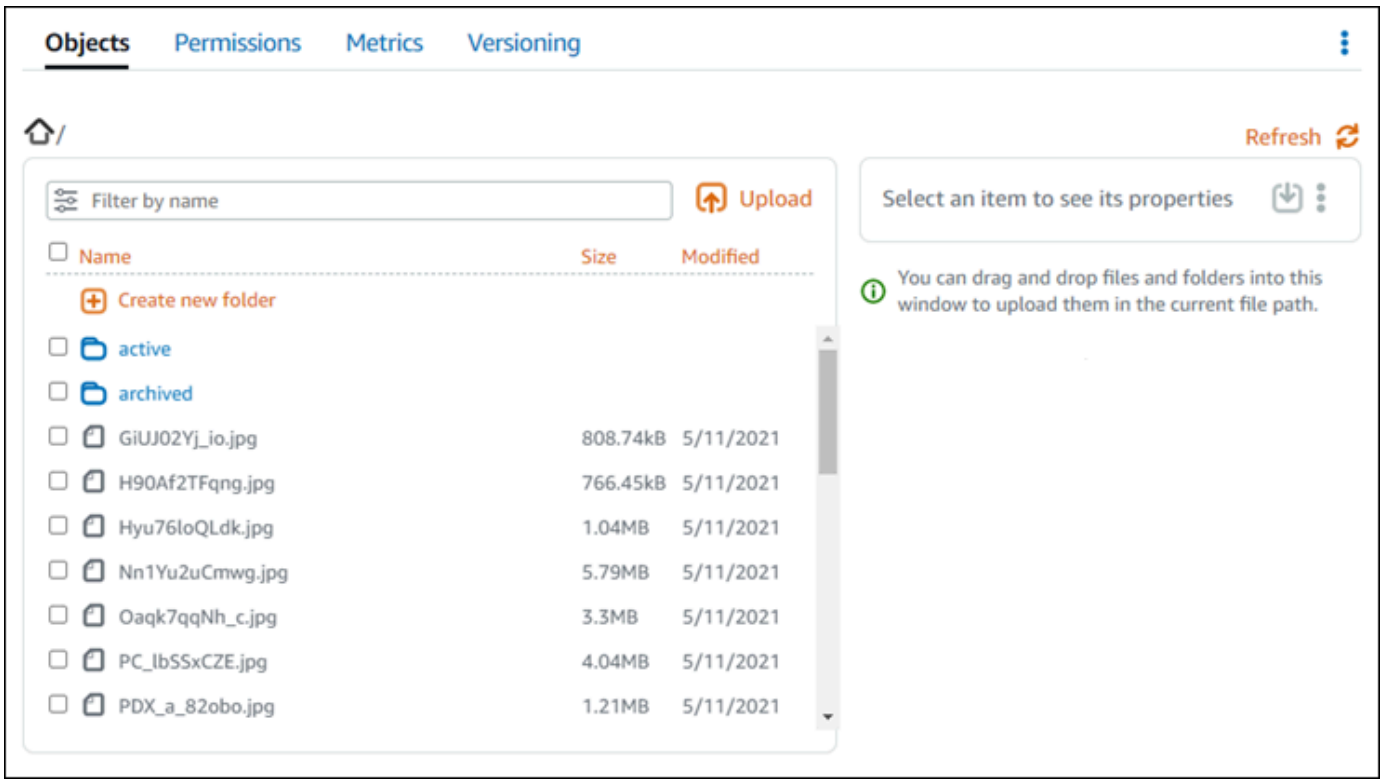
Lightsail バケット内のファイルとフォルダを管理する

Lightsail コンソールを使用して、Amazon Lightsail オブジェクトストレージサービスのバケットに保存されているすべてのオブジェクトを表示できます。(AWS Command Line Interface AWS CLI) と AWS SDKs を使用して、バケット内のオブジェクトキーを一覧表示することもできます。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

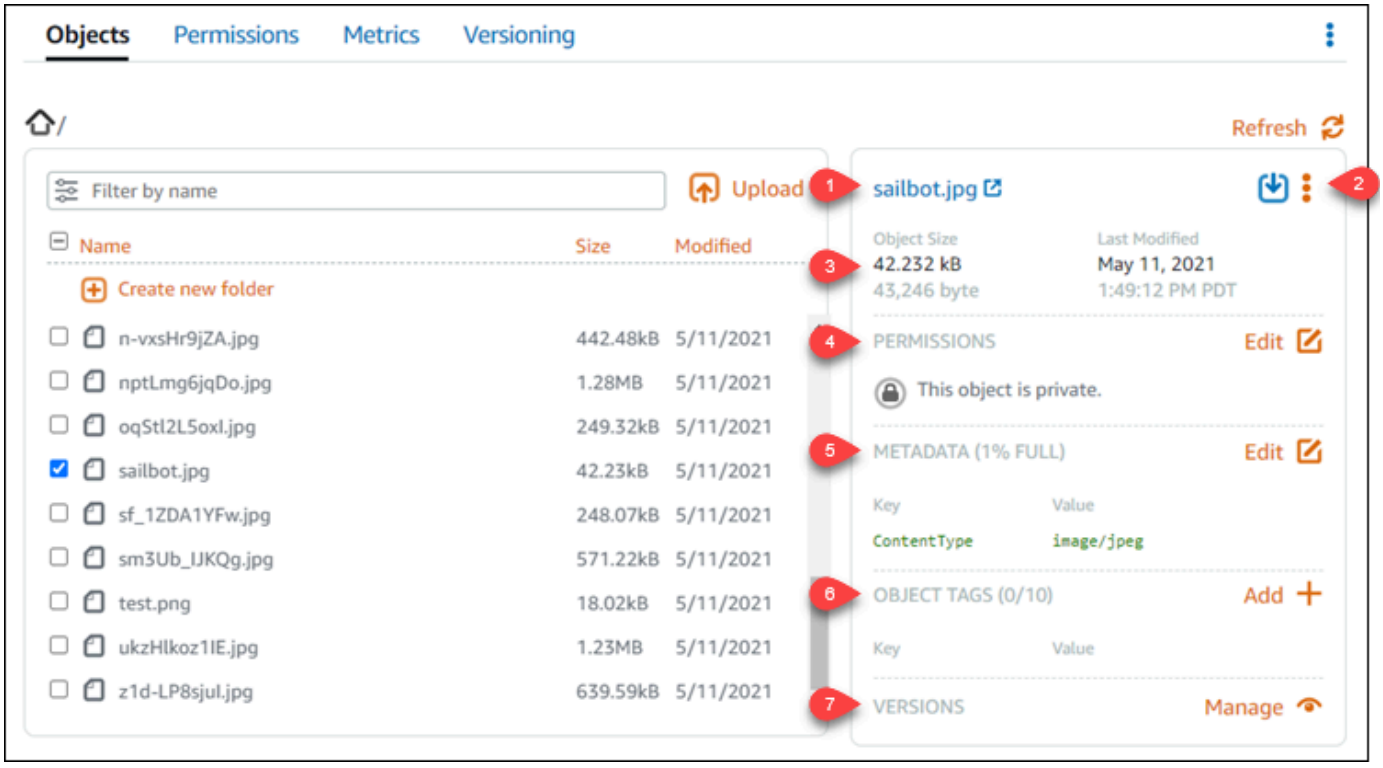
Lightsail コンソールを使用してオブジェクトをフィルターする

Lightsail コンソールを使用してバケットに格納されたオオブジェクトストレージを表示するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. オブジェクトを表示するバケットの名前を選択します。
4. [オブジェクト] タブの [オブジェクトブラウザ] ペインには、バケットに保存されているオブジェクトとフォルダが表示されます。



5. プロパティを表示するオブジェクトのロケーションを見つけます。
6. プロパティを表示するオブジェクトの横にチェックマークを追加します。
7. ページの右側にある [オブジェクトのプロパティ] ペインに、オブジェクトに関する情報が表示されます。



表示される情報には、次の情報が含まれます。

1. オブジェクトを表示およびダウンロードするリンク。
2. アクションメニュー (:) を使用して、オブジェクトをコピーまたは削除します。オブジェクトのコピーと削除の詳細については、「Amazon Lightsail バケット内のオブジェクトのコピーまたは移動」および「バケットのオブジェクトの削除」を参照してください。
3. オブジェクトのサイズ、および最終更新タイムスタンプ。
4. 個々のオブジェクトのアクセス許可は、プライベートまたは公開 (読み取り専用) です。バケットのアクセス許可の詳細については、「バケットのアクセス許可」を参照してください。
5. オブジェクトのメタデータ。コンテンツタイプ (ContentType) キーは、現時点で Lightsail オブジェクトストレージサービスがサポートする唯一のメタデータです。
6. オブジェクトキーバリュータグ 詳細については、「バケットオブジェクトにタグを付ける」を参照してください。
7. オブジェクトの保存されたバージョンを管理するオプション。詳細については、「バケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。

> **Note**
>
> 複数のオブジェクトを選択すると、[オブジェクトのプロパティ] ペインには、選択したオブジェクトの合計サイズのみが表示されます。

を使用してオブジェクトを表示する AWS CLI

AWS Command Line Interface (AWS CLI) を使用して、バケットのオブジェクトのキーをリスト化するには、次の手順を実行します。これは、`list-objects-v2` コマンドを使用して実行できます。詳細については、「AWS CLI コマンドリファレンス」の「list-objects-v2」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS Command Line Interface ように Amazon Lightsailを設定する」を参照してください。

1. コマンドプロンプトまたはターミナルウィンドウを開きます。
2. 以下のいずれかのコマンドを入力します。

   - 次のコマンドを入力して、バケット内のすべてのオブジェクトキーをリスト化します。

     ```
     aws s3api list-objects-v2 --bucket BucketName --query "Contents[].{Key: Key,
      Size: Size}"
     ```

     コマンドで、*BucketName* をすべてのオブジェクトをリスト化するバケットの名前に置き換えます。

   - 次のコマンドを入力して、特定のオブジェクトキー名のプレフィックスで始まるオブジェクトをリスト化します。

     ```
     aws s3api list-objects-v2 --bucket BucketName --prefix ObjectKeyNamePrefix --
     query "Contents[].{Key: Key, Size: Size}"
     ```

     コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *BucketName* - すべてのオブジェクトをリストするバケット名。
- *ObjectKeyNamePrefix* - オブジェクトキー名のプレフィックスで、指定されたプレフィックスで始まるキーへのレスポンスを制限します。

> **Note**
>
> これらのコマンドは、--query パラメータを使用して、list-objects-v2 リクエストのレスポンスを各オブジェクトのキーバリューとサイズにフィルタリングします。

例:

バケット内のすべてのオブジェクトキーをリスト化

```
aws s3api list-objects-v2 --bucket amzn-s3-demo-bucket --query "Contents[].{Key:
 Key, Size: Size}"
```

前述のコマンドでは、次の例に示すような結果が表示されます。

```
C:\>aws s3api list-objects-v2 --bucket DOC-EXAMPLE-BUCKET --query "Contents[].{Key: Key, Size: Size}"
[
    {
        "Key": "GiUJ02Yj_io.jpg",
        "Size": 828150
    },
    {
        "Key": "H90Af2TFqng.jpg",
        "Size": 784846
    },
    {
        "Key": "Hyu76loQLdk.jpg",
        "Size": 1086363
    },
    {
        "Key": "Nn1Yu2uCmwg.jpg",
        "Size": 6075006
    },
    {
        "Key": "Oaqk7qqNh_c.jpg",
        "Size": 3458557
    },
    {
        "Key": "PC_lbSSxCZE.jpg",
        "Size": 4239636
    },
    {
        "Key": "PDX_a_82obo.jpg"
```

オブジェクトキーのリストは、archived/オブジェクトキー名のプレフィックスで始まります:

```
aws s3api list-objects-v2 --bucket amzn-s3-demo-bucket --prefix archived/ --query
 "Contents[].{Key: Key, Size: Size}"
```

前述のコマンドでは、次の例に示すような結果が表示されます。

```
C:\>aws s3api list-objects-v2  --bucket DOC-EXAMPLE-BUCKET --prefix archived/ --query "Contents[].{Key: Key, Size: Size}"
[
    {
        "Key": "archived/",
        "Size": 0
    },
    {
        "Key": "archived/1_CMoFsPfso.jpg",
        "Size": 2561865
    },
    {
        "Key": "archived/3y1zF4hIPCg.jpg",
        "Size": 6404907
    },
    {
        "Key": "archived/5IHz5WhosQE.jpg",
        "Size": 2377975
    },
    {
        "Key": "archived/sailbot.jpg",
        "Size": 43246
    }
]
```

バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。
3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

- Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
- Amazon Lightsail でのバケットのアクセス許可の設定
- Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
- Amazon Lightsail でのバケットのアクセスキーの作成
- Amazon Lightsail でのバケットのリソースアクセスの設定
- Amazon Lightsail でのバケットのクロスアカウントアクセスの設定

5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定
6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail のバケットにファイルをアップロードする
   - マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
   - Amazon Lightsail のバケット内のオブジェクトの表示
   - Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
   - Amazon Lightsail のバケットからのオブジェクトのダウンロード
   - Amazon Lightsail のバケット内のオブジェクトのフィルタリング
   - Amazon Lightsail のバケット内のオブジェクトのタグ付け
   - Amazon Lightsail のバケット内のオブジェクトの削除

9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。

10.オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。

11.バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。

12.バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。

13.ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。

14.バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。

- チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
- チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用

15.使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

トピック

- Lightsail バケットの間でオブジェクトをコピーおよび移動する
- オブジェクトを削除して Lightsail バケットストレージをクリアする
- Lightsail バケットからオブジェクトをダウンロードする
- 名前プレフィックスで Lightsail バケット内のオブジェクトをフィルタリングする
- Lightsail のオブジェクトバージョニングの有効化と一時停止
- Lightsail バケット内の以前のオブジェクトバージョンを復元する
- Lightsail バケット内のオブジェクトにタグを付ける

Lightsail バケットの間でオブジェクトをコピーおよび移動する

Amazon Lightsail オブジェクトストレージサービスのバケットに既に保存されているオブジェクトをコピーできます。このガイドでは、 Lightsailコンソールと AWS Command Line Interface () を使用してオブジェクトをコピーする方法を示しますAWS CLI。バケット内のオブジェクトをコピーして、オブジェクトの複製コピーを作成したり、オブジェクトの名前を変更したり、場所間でオブジェクトを移動したりします (たとえば、オブジェクトをある場所から別のLightsail場所に移動 AWS リージョン し、 Lightsailが使用可能) 。ロケーション間でオブジェクトをコピーできるのは、 AWS APIs AWS SDKs、および AWS Command Line Interface () のみですAWS CLI。

バケットについての詳細は、「オブジェクトストレージ」を参照してください。

オブジェクトのコピーに関する制約事項

Lightsail コンソールでは、2 GB までのサイズのオブジェクトのコピーを作成することができます。AWS Command Line Interface (AWS CLI)、 AWS APIs、 AWS SDKs を使用して、1 つのコピーオブジェクトアクションで最大 5 GB のサイズのオブジェクトのコピーを作成できます。サイズが 5 GB を超えるオブジェクトをコピーするには、 AWS CLI、 AWS APIs、 AWS SDKs のマルチパートアップロードアクションを使用する必要があります。詳細については、「マルチパートアップロードを使用してバケットにファイルをアップロードする」を参照してください。

Lightsailコンソールを使用してオブジェクトをコピーする

Lightsailコンソールを使用し、バケットに格納されたオブジェクトをコピーするには、以下の手順を実行します。バケット内のオブジェクトを移動するには、そのオブジェクトを新しい場所にコピーし、元のオブジェクトを削除する必要があります。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. オブジェクトのコピー先のバケットの名前を選択します。
4. オブジェクトのタブで、オブジェクトブラウザペインを使用し、オブジェクトのコピー先の口ケーションを参照します。
5. コピーするオブジェクトの隣のチェックマークを入れます。
6. オブジェクト情報のウィンドウで、アクション (:) メニューを選択し、にコピーします。
7. 送信先の選択ペインで、選択したオブジェクトをコピーするバケット内の口ケーションを参照します。送信先パステキストボックスにフォルダ名を入力して、新しいパスを作成することもできます。

8. 選択したコピー先または指定したコピー先にオブジェクトをコピーするためには、コピーを選択します。それ以外の場合は、[いいえ、キャンセル] を選択します。

   オブジェクトが正常にコピーされると、コピー完了のメッセージが表示されます。オブジェクトの移動を目的としていた場合は、元のオブジェクトを削除する必要があります。詳細については、「バケットオブジェクトを削除する」を参照してください。

を使用してオブジェクトをコピーする AWS CLI

AWS Command Line Interface () を使用してバケット内のオブジェクトをコピーするには、次の手順を実行しますAWS CLI。これは、copy-object コマンドを使用して実行できます。詳細については、「AWS CLI コマンドリファレンス」の「copy-object」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. バケット内のオブジェクトをコピーするには、次のコマンドを入力します。

   ```
   aws s3api copy-object --copy-source SourceBucketNameAndObjectKey --
   key DestinationObjectKey --bucket DestinationBucketName --acl bucket-owner-full-
   control
   ```

   コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

   - *SourceBucketNameAndObjectKey* - ソースオブジェクトが現在存在するバケットの名前と、コピーされるオブジェクトのフルオブジェクトキー。たとえば、amzn-s3-demo-bucketバケットからオブジェクトimages/sailbot.jpgをコピーするには、amzn-s3-demo-bucket/images/sailbot.jpgを指定します。
   - *DestinationObjectKey* - 新しいオブジェクトのコピーのフルオブジェクトキー。
   - *DestinationBucket* - 送信先バケットの名前。

   例:

- バケット内のオブジェクトを同じバケット内にコピーする:

```
aws s3api copy-object --copy-source amzn-s3-demo-bucket1/images/sailbot.jpg
 --key media/sailbot.jpg --bucket amzn-s3-demo-bucket --acl bucket-owner-full-
control
```

- バケットから別のバケットへオブジェクトをコピーする:

```
aws s3api copy-object --copy-source amzn-s3-demo-bucket1/images/sailbot.jpg --
key images/sailbot.jpg --bucket amzn-s3-demo-bucket2 --acl bucket-owner-full-
control
```

以下の例のような結果が表示されるはずです。

```
C:\>aws s3api copy-object --copy-source DOC-EXAMPLE-BUCKET/images/sailbot.jpg --key images/archived/sailbot.jpg --bucket DOC-EXAMPLE-BUCKET
{
    "ServerSideEncryption": "AES256",
    "CopyObjectResult": {
        "ETag": "\"694d34example91d92d64f342aa234c3\"",
        "LastModified": "2021-05-10T05:35:42+00:00"
    }
}
```

バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。
3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

- Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
- Amazon Lightsail でのバケットのアクセス許可の設定
- Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
- Amazon Lightsail でのバケットのアクセスキーの作成
- Amazon Lightsail でのバケットのリソースアクセスの設定
- Amazon Lightsail でのバケットのクロスアカウントアクセスの設定

5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定
6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail のバケットにファイルをアップロードする
   - マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
   - Amazon Lightsail のバケット内のオブジェクトの表示
   - Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
   - Amazon Lightsail のバケットからのオブジェクトのダウンロード
   - Amazon Lightsail のバケット内のオブジェクトのフィルタリング
   - Amazon Lightsail のバケット内のオブジェクトのタグ付け
   - Amazon Lightsail のバケット内のオブジェクトの削除

9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。
11. バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。
12. バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。
13. ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。
14. バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。
    - チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
    - チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用
15. 使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

オブジェクトを削除して Lightsail バケットストレージをクリアする

Amazon Lightsail オブジェクトストレージサービス内のバケットからオブジェクトを削除することができます。ストレージ領域を解放するには、不要になったオブジェクトを削除します。たとえば、ログファイルを収集している場合は、不要になったファイルを削除することをお勧めします。

バケットについての詳細は、「オブジェクトストレージ」を参照してください。

目次

- バージョニングが有効なバケットからオブジェクトを削除する
- Lightsail コンソールを使用してオブジェクトを削除します。
- Lightsail コンソールを使用してオブジェクトバージョンを削除する
- を使用して単一のオブジェクトまたはオブジェクトバージョンを削除する AWS CLI

- を使用して複数のオブジェクトまたはオブジェクトバージョンを削除する AWS CLI

バージョニングが有効なバケットからオブジェクトを削除する

バージョニングがバケットで有効化されている場合、複数のバージョンのオブジェクトがバケット内に存在する可能性があります。Lightsail コンソール、 AWS APIs AWS CLI、または AWS SDK を使用して、オブジェクトの任意のバージョンを削除できます。ただし、次のオプションを検討する必要があります。

Lightsail コンソールを使用して、オブジェクトおよびオブジェクトバージョンを削除する

Lightsail コンソールの [オブジェクト] タブのオブジェクトブラウザペインでオブジェクトの最新バージョンを削除すると、以前のバージョンのオブジェクトもすべて削除されます。オブジェクトの特定のバージョンを削除するには、バージョンの管理ペインから実行してください。バージョン管理ペインを使用してオブジェクトの現在のバージョンを削除すると、以前の最新のバージョンが現在のバージョンとして復元されます。詳細については、このガイドで後述する「Lightsail コンソールを使用してオブジェクトバージョンを削除する」を参照してください。

Lightsail API、 AWS CLI、または AWS SDKs を使用してオブジェクトとオブジェクトバージョンを削除する

単一のオブジェクトとその保存されているすべてのバージョンを削除するには、削除リクエストでオブジェクトのキーのみを指定します。オブジェクトの特定のバージョンを削除するためには、オブジェクトのキー名とバージョン ID の両方を指定します。詳細については、このガイドで後述する「 AWS CLIで単一のオブジェクトまたはオブジェクトバージョンを削除するには」を参照してください。

Lightsail コンソールを使用してオブジェクトを削除します。

Lightsail コンソールを使い、保存された以前のバージョンを含めオブジェクトを削除するには、次の手順に従います。Lightsail コンソールでは、一度に 1 つずつしかオブジェクトを削除できません。を使用して AWS CLI 、複数のオブジェクトを一度に削除します。詳細については、このガイドで後述する「AWS CLIを使用して複数のオブジェクトまたはオブジェクトバージョンを削除する」を参照してください。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. オブジェクトを削除するバケットの名前を選択します。

4. [オブジェクト] タブのオブジェクトブラウザペインを使用して、削除するオブジェクトの場所を参照します。
5. 削除するオブジェクトの横にあるチェックマークを追加します。
6. オブジェクト情報ペインで、アクション (:) メニューを選択し、[削除] を選択します。
7. 表示される確認ペインで[はい、削除します]を選択し、オブジェクトを完全に削除することを確認します。

   フォルダ内の唯一のオブジェクトを削除すると、そのフォルダも削除されます。これは、フォルダがオブジェクトキー名の一部であり、バケット内の他のオブジェクトが同じオブジェクトプレフィックスを共有していない場合、オブジェクトを削除すると、先行するフォルダも削除されるために発生します。詳細については、「オブジェクトストレージバケットのキー名」を参照してください。

Lightsail コンソールを使用してオブジェクトバージョンを削除する。

オブジェクトの保存されたバージョンを削除するには、次の手順を実行します。これは、バージョニングが有効なバケットでのみ可能です。詳細については、「バケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. オブジェクトを削除するバケットの名前を選択します。
4. オブジェクトブラウザペインを使用して、削除するオブジェクトの場所を参照します。
5. 削除するオブジェクトの保存された旧バージョンの横にチェックマークを追加します。
6. オブジェクト情報ペインのバージョンのセクションで[Manage] (管理) を選択します。
7. 保存されたオブジェクトのバージョンを管理するペインで、削除するオブジェクトのバージョンの横にチェックマークを追加します。

   オブジェクトの現在のバージョンを削除するように選択することもできます。
8. [選択済みを削除] をクリックして、選択したバージョンを削除します。

   削除した場合:

   - オブジェクトの現在のバージョン-オブジェクトの以前の最新のバージョンが現在のバージョンとして復元されます。

- オブジェクトの唯一のバージョン-オブジェクトがバケットから削除されます。削除したバージョンが現在のフォルダ内の唯一のオブジェクトである場合、フォルダも削除されます。これは、フォルダがオブジェクトキー名の一部であり、バケット内の他のオブジェクトが同じオブジェクトキープレフィックス共有していない場合、オブジェクトを削除すると、先行するフォルダも削除されるため発生します。詳細については、「バケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。

を使用して単一のオブジェクトまたはオブジェクトバージョンを削除する AWS CLI

AWS Command Line Interface () を使用してバケット内の単一のオブジェクトまたはオブジェクトバージョンを削除するには、次の手順を実行しますAWS CLI。これは、`delete-object` コマンドを使用して実行できます。詳細については、「AWS CLI コマンドリファレンス」の「delete-object」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS Command Line Interface ように Amazon Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. バケット内のオブジェクトまたはオブジェクトバージョンを削除するには、次のコマンドを入力します。

   オブジェクトを削除するには:

   ```
   aws s3api delete-object --bucket BucketName --key ObjectKey
   ```

   オブジェクトバージョンを削除するには

   > **Note**
   >
   > オブジェクトバージョンの削除は、バージョニングが有効なバケットでのみ可能です。詳細については、「バケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。

```
aws s3api delete-object --bucket BucketName --key ObjectKey --version-id VersionID
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *BucketName* - リストで、オブジェクトを削除するバケットの名前を選択します。
- *ObjectKey* - 削除するオブジェクトの完全なオブジェクトキー。
- *VersionId* - 削除するオブジェクトバージョンの ID。

例:

オブジェクトの削除：

```
aws s3api delete-object --bucket amzn-s3-demo-bucket --key images/sailbot.jpg
```

オブジェクトバージョンの削除：

```
aws s3api delete-object --bucket amzn-s3-demo-bucket --key images/sailbot.jpg --
version-id YF0YMBlUvexampleO07l2vJi9hRz4ujX
```

以下の例のような結果が表示されるはずです。

```
C:\Users\latino>aws s3api delete-object --bucket DOC-EXAMPLE-BUCKET  --key  images/sailbot.jpg  --version-id YF0YMBlUvexampleO07l2vJi9hRz4ujX
{
    "VersionId": "YF0YMBexampleY7PO07l2vJi9hRz4ujX"
}
```

を使用して複数のオブジェクトまたはオブジェクトバージョンを削除する AWS CLI

AWS Command Line Interface (AWS CLI) を使用してバケット内の複数のオブジェクトを削除するには、以下の手順を実行します。これは、delete-objects コマンドを使用して実行できます。詳細については、 AWS CLI 「 コマンドリファレンス」の「delete-objects」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS Command Line Interface ように Amazon Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. バケット内の複数のオブジェクトまたは複数のオブジェクトバージョンを削除するには、次のコマンドを入力します。

```
aws s3api delete-objects --bucket BucketName --delete file://LocalDirectory
```

以下のコマンド例を使用するには、以下のテキストを独自のものに置き換えてください。

- *BucketName* - 複数のオブジェクトまたは複数のオブジェクトバージョンを削除するバケットの名前。
- *LocalDirectory* - 削除するオブジェクトまたはバージョンを指定する .json ドキュメントのコンピュータ上のディレクトリパス。.json ドキュメントは以下のようにフォーマットできます。

  オブジェクトを削除するには、.json ファイルに次のテキストを入力し、*ObjectKey* を削除するオブジェクトのオブジェクトキーに置き換えます。

```
{
  "Objects": [
    {
      "Key": "ObjectKey1"
    },
    {
      "Key": "ObjectKey2"
    }
  ],
  "Quiet": false
}
```

  オブジェクトのバージョンを削除するには、.json ファイルに次のテキストを入力します。*ObjectKey* および *VersionID* を削除するオブジェクトバージョンのオブジェクトキーとオブジェクト ID に置き換えます。

> **Note**
>
> オブジェクトバージョンの削除は、バージョンが有効なバケットでのみ可能です。詳細については、「バケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。

```
{
  "Objects": [
    {
      "Key": "ObjectKey1",
      "VersionId": "VersionID1"
    },
    {
      "Key": "ObjectKey2",
      "VersionId": "VersionID2"
    }
  ],
  "Quiet": false
}
```

例:

- Linux または Unix コンピュータの場合は、次の操作を行います。

```
aws s3api delete-objects --bucket amzn-s3-demo-bucket --delete file://home/user/Documents/delete-objects.json
```

- Windows コンピュータの場合:

```
aws s3api delete-objects --bucket amzn-s3-demo-bucket --delete file://C:\Users\user\Documents\delete-objects.json
```

以下の例のような結果が表示されるはずです。

```
C:\>aws s3api delete-objects --bucket DOC-EXAMPLE-BUCKET  --delete file://C:\Users\user\Documents\delete-objects.json
{
    "Deleted": [
        {
            "Key": "images/sailbot.jpg",
            "VersionId": "26sqexampleztRiT6TsGhMMz0FxQAEw."
        },
        {
            "Key": "images/sailbot.jpg",
            "VersionId": "QwDrexampleDJxJtZC1CrExbpNlEC5O4"
        }
    ]
}
```

バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。
3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

   バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

   - Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
   - Amazon Lightsail でのバケットのアクセス許可の設定
   - Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
   - Amazon Lightsail でのバケットのアクセスキーの作成
   - Amazon Lightsail でのバケットのリソースアクセスの設定
   - Amazon Lightsail でのバケットのクロスアカウントアクセスの設定
5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。

   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定

6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail のバケットにファイルをアップロードする
   - マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
   - Amazon Lightsail のバケット内のオブジェクトの表示
   - Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
   - Amazon Lightsail のバケットからのオブジェクトのダウンロード
   - Amazon Lightsail のバケット内のオブジェクトのフィルタリング
   - Amazon Lightsail のバケット内のオブジェクトのタグ付け
   - Amazon Lightsail のバケット内のオブジェクトの削除
9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。
11. バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。
12. バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。
13. ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。
14. バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。
   - チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
   - チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用

15.使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

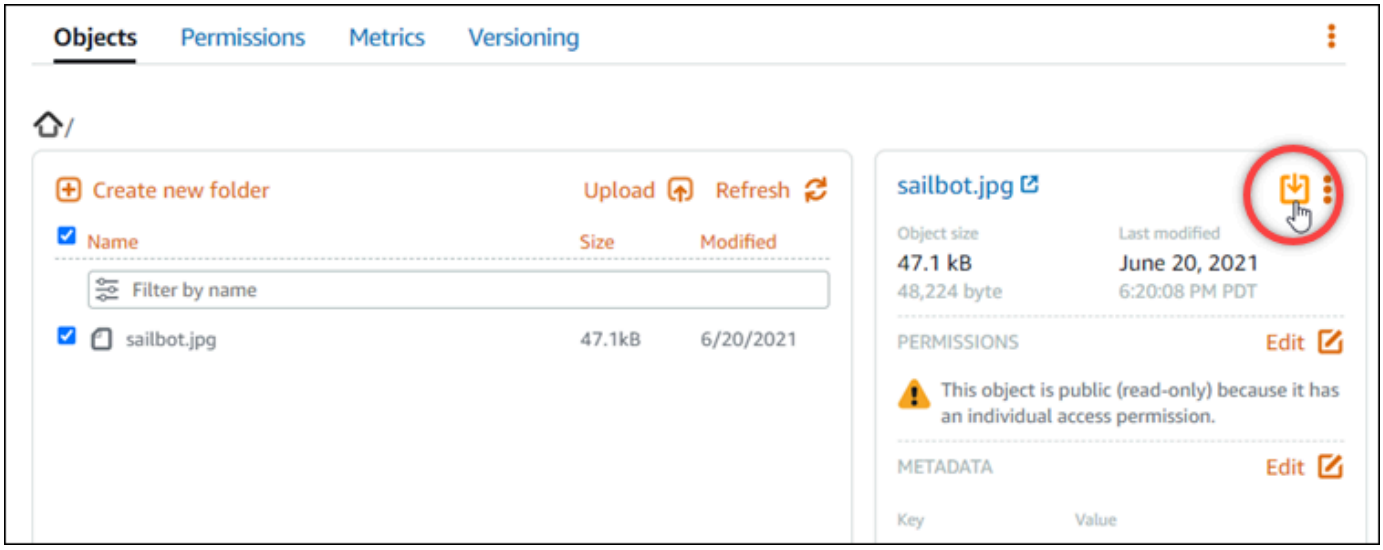
Lightsail バケットからオブジェクトをダウンロードする

Amazon Lightsail オブジェクトストレージサービスでは、アクセス可能なバケットまたはパブリック（読み取り専用）のバケットからオブジェクトをダウンロードすることができます。Lightsail コンソールを使用すれば、オブジェクトを一つずつダウンロードできます。1 つのリクエストで複数のオブジェクトをダウンロードするには、 AWS Command Line Interface (AWS CLI)、 AWS SDKs、または REST API を使用します。このガイドでは、Lightsail コンソールと AWS CLIを使用してオブジェクトをダウンロードする方法を紹介します。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

Lightsail コンソールを使用してオブジェクトをダウンロードする

Lightsail コンソールを使用してバケットからオブジェクトをダウンロードするには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. ファイルをダウンロードしたいバケットの名前を選択します。
4. オブジェクトタブのオブジェクトブラウザペインでダウンロードするオブジェクトの場所を参照します。
5. ダウンロードするオブジェクトの横にチェックマークを追加します。
6. 左オブジェクト情報ペインで、ダウンロードアイコンを選択します。



ブラウザの設定に応じて、選択したファイルはページに表示されるか、コンピュータにダウンロードされます。ファイルがページに表示されている場合は、ファイルを右クリックして、[Save as] を選択すると、コンピュータに保存されます。

を使用してオブジェクトをダウンロードする AWS CLI

AWS Command Line Interface (AWS CLI) を使用してバケットからオブジェクトをダウンロードするには、次の手順を実行します。これは、`get-object` コマンドを使用して実行できます。詳細については、「AWS CLI コマンドリファレンス」の「get-object」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS Command Line Interface ようにAmazon Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. バケットからオブジェクトをダウンロードするには、次のコマンドを入力します。

```
aws s3api get-object --bucket BucketName --key ObjectKey LocalFilePath
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *BucketName* - オブジェクトのダウンロード元となるバケット名。
- *ObjectKey* - ダウンロードするオブジェクトの完全なオブジェクトキー。
- *LocalFilePath* - ダウンロードしたファイルを保存するコンピュータ上の完全なファイルパス。

例:

```
aws s3api get-object --bucket amzn-s3-demo-bucket --key images/sailbot.jpg C:\Users
\user\Pictures\sailbot.jpg
```

以下の例のような結果が表示されるはずです。

```
C:\>aws s3api get-object --bucket DOC-EXAMPLE-BUCKET --key images/sailbot.jpg C:\Users\user\Pictures\sailbot.jpg
{
    "AcceptRanges": "bytes",
    "LastModified": "2021-05-10T05:09:31+00:00",
    "ContentLength": 48224,
    "ETag": "\"694d34example91d92d64f342aa234c3\"",
    "ContentType": "binary/octet-stream",
    "ServerSideEncryption": "AES256",
    "Metadata": {}
}
```

バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。
3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

   バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

   - Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
   - Amazon Lightsail でのバケットのアクセス許可の設定
   - Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
   - Amazon Lightsail でのバケットのアクセスキーの作成
   - Amazon Lightsail でのバケットのリソースアクセスの設定
   - Amazon Lightsail でのバケットのクロスアカウントアクセスの設定

5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定
6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail のバケットにファイルをアップロードする
   - マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
   - Amazon Lightsail のバケット内のオブジェクトの表示
   - Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
   - Amazon Lightsail のバケットからのオブジェクトのダウンロード
   - Amazon Lightsail のバケット内のオブジェクトのフィルタリング
   - Amazon Lightsail のバケット内のオブジェクトのタグ付け
   - Amazon Lightsail のバケット内のオブジェクトの削除
9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。
11. バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。

12バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。

13ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。

14バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。

- チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
- チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用

15使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

名前プレフィックスで Lightsail バケット内のオブジェクトをフィルタリングする

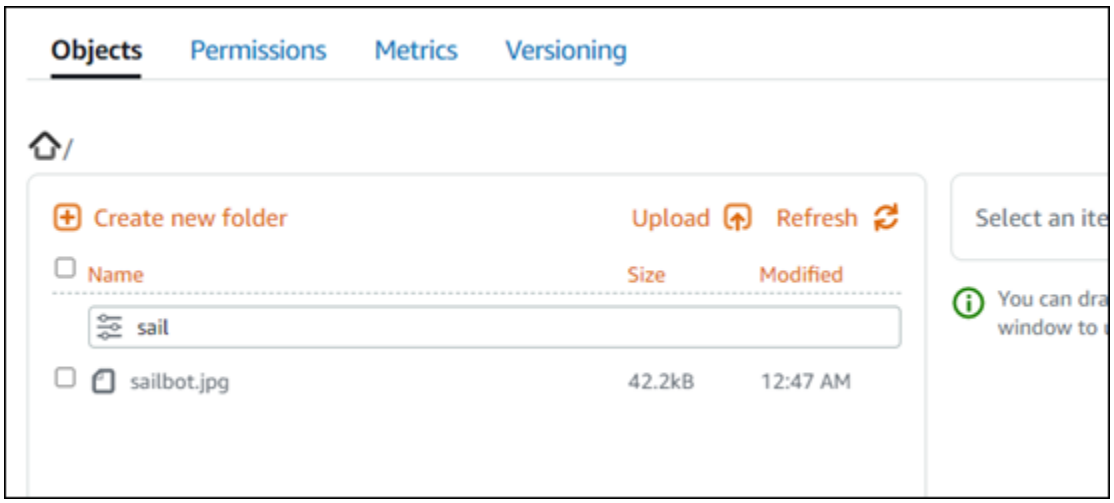
Amazon Lightsail オブジェクトストレージサービスでは、バケット内のオブジェクトをフィルタリングで検索できます。このガイドでは、Lightsail コンソールおよび AWS Command Line Interface (AWS CLI) を使用してオブジェクトをフィルタリングする方法を紹介します。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

Lightsail コンソールを使用してオブジェクトをフィルターする

Lightsail コンソールでバケットのオブジェクトをフィルタリングするには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. オブジェクトを検索するバケットの名前を選択します。
4. オブジェクトタブで、オブジェクトの接頭辞を [名前でフィルタリングする] テキストボックスに入力します。

   現在表示中のフォルダのオブジェクトのリストは、入力したテキストに合わせてフィルタされます。次の例は、`sail` と入力した場合、ページ上のオブジェクトのリストが `sail` で始まるものに限定してフィルタリングされることを示しています。



別のフォルダでオブジェクトのリストをフィルタするには、そのフォルダへ移動します。次に、オブジェクトの接頭辞を [名前でフィルタリングする] テキストボックスに入力します。

を使用してオブジェクトをフィルタリングする AWS CLI

AWS Command Line Interface (AWS CLI) を使用してバケットのオブジェクトをフィルタリングするには、以下の手順を実行します。これは、`list-objects-v2` コマンドを使用して実行できます。詳細については、「AWS CLI コマンドリファレンス」の「list-objects-v2」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS Command Line Interface ように Amazon Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 特定のオブジェクトキー名のプレフィックスで始まるオブジェクトをリストするには、次のコマンドを入力します。

```
aws s3api list-objects-v2 --bucket BucketName --prefix ObjectKeyNamePrefix --query
 "Contents[].{Key: Key, Size: Size}"
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *BucketName* - すべてのオブジェクトをリストするバケット名。

- *ObjectKeyNamePrefix* - 指定されたプレフィックスで始まるキーに応答を制限するオブジェクトキー名プレフィックス。

> **Note**
>
> このコマンドは、--query パラメーターを利用し、list-objects-v2 リクエストへの応答を各オブジェクトのキー値とサイズにフィルタリングします。

例:

```
aws s3api list-objects-v2 --bucket amzn-s3-demo-bucket --prefix archived/ --query
 "Contents[].{Key: Key, Size: Size}"
```

次の例のような結果が表示されます。

```
C:\>aws s3api list-objects-v2  --bucket DOC-EXAMPLE-BUCKET --prefix archived/ --query "Contents[].{Key: Key, Size: Size}"
[
    {
        "Key": "archived/",
        "Size": 0
    },
    {
        "Key": "archived/1_CMoFsPfso.jpg",
        "Size": 2561865
    },
    {
        "Key": "archived/3y1zF4hIPCg.jpg",
        "Size": 6404907
    },
    {
        "Key": "archived/5IHz5WhosQE.jpg",
        "Size": 2377975
    },
    {
        "Key": "archived/sailbot.jpg",
        "Size": 43246
    }
]
```

バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。

3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

   バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

   - Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
   - Amazon Lightsail でのバケットのアクセス許可の設定
   - Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
   - Amazon Lightsail でのバケットのアクセスキーの作成
   - Amazon Lightsail でのバケットのリソースアクセスの設定
   - Amazon Lightsail でのバケットのクロスアカウントアクセスの設定
5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。

   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定
6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。

   - Amazon Lightsail のバケットにファイルをアップロードする

- マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
- Amazon Lightsail のバケット内のオブジェクトの表示
- Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
- Amazon Lightsail のバケットからのオブジェクトのダウンロード
- Amazon Lightsail のバケット内のオブジェクトのフィルタリング
- Amazon Lightsail のバケット内のオブジェクトのタグ付け
- Amazon Lightsail のバケット内のオブジェクトの削除

9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。
11. バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。
12. バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。
13. ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。
14. バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。
    - チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
    - チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用
15. 使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

Lightsail のオブジェクトバージョニングの有効化と一時停止

Amazon Lightsail オブジェクトストレージサービスでのバージョニングとは、同じバケット内でオブジェクトの複数のバリアントを保持する手段です。バージョニング機能を使用すると、バケットに

保存されたすべてのオブジェクトのすべてのバージョンを、保存、取得、復元することができます。バージョニングを使用すれば、誤ったユーザーアクションやアプリケーション障害からより簡単に回復することができます。バケットのバージョニングが有効な場合に、Lightsail オブジェクトストレージサービスが同じオブジェクトに対する複数の書き込みリクエストを同時に受信すると、すべてのオブジェクトが保存されます。バージョニングは Lightsail オブジェクトストレージサービスのバケットではデフォルトで無効になっているため、明示的に有効にする必要があります。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

> **Important**
>
> 「個々のオブジェクトを公開可能 (読み取り専用)」のアクセス権が設定されているバケットでバージョニングを有効または一時停止にすると、アクセス権は「すべてのオブジェクトはプライベートです」にリセットされます。個々のオブジェクトをパブリックにするオプションを引き続き使用する場合は、バケットのアクセス権限を手動で「個々のオブジェクトを公開可能 (読み取り専用)」に変更する必要があります。詳細については、「バケットのアクセス許可を設定する」を参照してください。

バージョンが無効化、有効化、一時停止されたバケット

Lightsail コンソールのバケットバージョニングには、3 つの状態があります。

- 無効 (API と SDK では `NeverEnabled`)
- 有効 (API と SDK では `Enabled`)
- 一時停止 (API と SDK では `Suspended`)

一度バケットでバージョニングを有効にすると、無効状態に戻すことはできません。ただし、バージョニングを一時停止することは可能です。バージョニングは、バケットレベルで有効化および停止します。

バージョニングの状態は、バケット内のすべてのオブジェクト (一部ではない) に適用されます。バケットでバージョニングを有効にすると、すべての新しいオブジェクトがバージョニングされ、一意のバージョン ID が割り当てられます。バージョニングが有効されたときにバケット内にすでに存在したオブジェクトは、それ以降は常にバージョニングされます。将来のリクエストによってオブジェクトが修正された場合、固有のバージョン ID が割り当てられます。

バージョン ID

バケットのバージョニングを有効にすると、Lightsail オブジェクトストレージサービスは保存されてるオブジェクトに対して固有のバージョン ID を自動的に生成します。例えば、1 つのバケット内に、`photo.gif` (バージョン 111111) と `photo.gif` (バージョン 121212) のように、キーは同じだがバージョン ID が異なる 2 つのオブジェクトを保持することができます。

バージョン ID を編集することはできません。バージョン ID は、Unicode、UTF-8 エンコード、URL 対応の不透明型文字列で、長さは 1,024 バイト以下です。以下はバージョン ID の例です。

```
3sL4kqtJlcpXroDTDmJ+rmSpXd3dIbrHY+MTRCxf3vjVBH40Nr8X8gdRQBpUMLUo
```

Lightsail コンソールを使用してオブジェクトのバージョニングを有効化または一時停止する

Lightsail コンソールを使用してオブジェクトのバージョニングを有効または一時停止するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. バージョニングを有効または一時停止するバケットの名前を選択します。
4. [Versioning] (バージョニング) タブを選択します。
5. バケットの現在のバージョニング状態に応じて、次のいずれかのアクションを実行します。
   - バージョニングが現在停止されているか、有効になっていない場合は、 ページの [Object versioning] (オブジェクトのバージョニング) セクションにあるトグルを選択してバージョニングを有効にします。
   - バージョニングが現在有効になっている場合は、 ページの [Object versioning] セクションにあるトグルを選択してバージョニングを一時停止にします。

を使用してオブジェクトのバージョニングを有効化または停止する AWS CLI

AWS Command Line Interface (AWS CLI) を使用してオブジェクトのバージョニングを有効化または一時停止するには、次の手順を実行します。これは、update-bucket コマンドを使用して実行できます。詳細については、「AWS CLI コマンドリファレンス」の「update-bucket」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 次のコマンドを入力して、オブジェクトのバージョニングを有効または一時停止にします。

```
aws lightsail update-bucket --bucket-name BucketName --versioning VersioningState
```

コマンド内で、次のサンプルテキストを独自のテキストに置き換えます。

- *BucketName* - オブジェクトのバージョニングを有効にしたいバケットの名前。
- *VersioningState* - 以下のいずれかを指します。
  - Enabled - オブジェクトのバージョニングを有効にする。
  - Suspended - 有効になっているオブジェクトのバージョニングを一時停止する。

例:

```
aws lightsail update-bucket --bucket-name amzn-s3-demo-bucket --versioning Enabled
```

以下の例のような結果が表示されるはずです。

```
C:\>aws lightsail update-bucket --bucket-name DOC-EXAMPLE-BUCKET --versioning Enabled
{
    "bucket": {
        "resourceType": "Bucket",
        "accessRules": {
            "getObject": "private",
            "allowPublicOverrides": false
        },
        "arn": "arn:aws:lightsail:us-west-2:1example7491:Bucket/f067383e-ee41-4485-b934-example2e2fd",
        "bundleId": "small_1_0",
        "createdAt": "2021-06-29T08:12:39.163000-07:00",
        "url": "https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com",
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "name": "DOC-EXAMPLE-BUCKET",
        "supportCode": "621291663362/DOC-EXAMPLE-BUCKET/small_1_0",
        "tags": [],
        "objectVersioning": "Enabled",
        "ableToUpdateBundle": true
    },
    "operations": [
        {
            "id": "0d53d290-f4b2-43f0-89d2-example43448",
            "resourceName": "DOC-EXAMPLE-BUCKET",
            "resourceType": "Bucket",
            "createdAt": "2021-06-29T08:29:56.241000-07:00",
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "6example3362/DOC-EXAMPLE-BUCKET/small_1_0",
            "operationType": "UpdateBucket",
            "status": "Succeeded",
            "statusChangedAt": "2021-06-29T08:29:56.241000-07:00",
            "errorCode": "",
            "errorDetails": ""
        }
    ]
}
```

バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。
3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。

4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

   バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

   - Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
   - Amazon Lightsail でのバケットのアクセス許可の設定
   - Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
   - Amazon Lightsail でのバケットのアクセスキーの作成
   - Amazon Lightsail でのバケットのリソースアクセスの設定
   - Amazon Lightsail でのバケットのクロスアカウントアクセスの設定
5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。

   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定
6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。

   - Amazon Lightsail のバケットにファイルをアップロードする
   - マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード

- Amazon Lightsail のバケット内のオブジェクトの表示
- Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
- Amazon Lightsail のバケットからのオブジェクトのダウンロード
- Amazon Lightsail のバケット内のオブジェクトのフィルタリング
- Amazon Lightsail のバケット内のオブジェクトのタグ付け
- Amazon Lightsail のバケット内のオブジェクトの削除

9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。
11. バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。
12. バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。
13. ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。
14. バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。
    - チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
    - チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用
15. 使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

Lightsail バケット内の以前のオブジェクトバージョンを復元する

Amazon Lightsail オブジェクトストレージサービスのバケットがバージョニング対応の場合、オブジェクトの前のバージョンを復元できます。オブジェクトの前のバージョンを復元して、意図せぬユーザーアクションやアプリケーションの障害から回復します。

Lightsail コンソールを使ってオブジェクトの前のバージョンを復元できます。 AWS Command Line Interface (AWS CLI) と AWS SDKsを使用して、オブジェクトの以前のバージョンを復元することもできます。これを行うには、オブジェクトの特定のバージョンをバケットにコピーし、同じオブジェクトキー名を使用します。これにより、現在のバージョンが前のバージョンに置き換えられ、前のバージョンが現在のバージョンになります。バージョニングの詳細については、「バケット内のオブジェクトのバージョニングの有効化と一時停止」を参照してください。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

Lightsail コンソールを使ってオブジェクトの前のバージョンを復元する

Lightsail コンソールを使用してオブジェクトのバージョニングを有効または一時停止するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. 前のバージョンを復元したいオブジェクトが入っているバケットの名前を選択します。
4. [オブジェクト] タブにある [オブジェクトブラウザ] ペインを使用して、オブジェクトの場所を参照します。
5. 前のバージョンを復元したいオブジェクトの横にチェックマークを追加します。
6. [オブジェクト情報] ペインの [バージョン] セクションにある [管理] を選択します。
7. [復元] を選択します。
8. 表示される [保存されたバージョン] ペインの [オブジェクトの復元] で、復元したいオブジェクトのバージョンを選択します。
9. [続行] をクリックしてください。
10. 確認プロンプトが表示されたら、[はい、復元します] を選択して、オブジェクトのバージョンを復元します。復元しない場合は、[いいえ。キャンセルする] を選択します。

を使用してオブジェクトの以前のバージョンを復元する AWS CLI

オブジェクト AWS Command Line Interface (AWS CLI) の前のバージョンを復元するには、次の手順を実行します。これは、`copy-object` コマンドを使用して行います。同じオブジェクトキーを使用して、オブジェクトの前のバージョンを同じバケットにコピーする必要があります。詳細については、「AWS CLI コマンドリファレンス」の「copy-object」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS Command Line Interface ように Amazon Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. オブジェクトの前のバージョンを復元するには、次のコマンドを入力します。

```
aws s3api copy-object --copy-source "BucketName/ObjectKey?versionId=VersionId" --
key ObjectKey --bucket BucketName
```

コマンド内で、次のサンプルテキストを独自のテキストに置き換えます。

- *BucketName* - 前のバージョンを復元したいオブジェクトが入っているバケットの名前。同じバケット名を --copy-source および --bucket パラメータに指定する必要があります。
- *ObjectKey* - 復元するオブジェクトの名前。同じオブジェクトキーの名前を --copy-source および --key パラメータに指定する必要があります。
- *VersionId* - 現在のバージョンとして復元したい、前のオブジェクトのバージョンの ID。list-object-versions コマンドを使用して、バケット内のオブジェクトのバージョン ID 一覧を取得します。

例:

```
aws s3api copy-object --copy-source "amzn-s3-demo-bucket/sailbot.jpg?
versionId=GQWEexample87Mdl8Q_DKdVTiVMi_VyU" -key sailbot.jpg --bucket amzn-s3-demo-
bucket
```

以下の例のような結果が表示されるはずです。

```
C:\>aws s3api copy-object --copy-source "DOC-EXAMPLE-BUCKET/sailbot.jpg?versionId=GQWEexample87Mdl8Q_DKdVTiVMi_VyU"
--key sailbot.jpg --bucket DOC-EXAMPLE-BUCKET
{
    "CopySourceVersionId": "GQWEcouyrfexampleQ_DKdVTiVMi_VyU",
    "VersionId": "hjL8anKzI1xcXYyexampleDvvqMXSLoi",
    "ServerSideEncryption": "AES256",
    "CopyObjectResult": {
        "ETag": "\"dc5afd388fb3example20cda3fe41c54\"",
        "LastModified": "2021-05-16T06:45:35+00:00"
    }
}
```

バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。
3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

   バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

   - Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
   - Amazon Lightsail でのバケットのアクセス許可の設定
   - Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
   - Amazon Lightsail でのバケットのアクセスキーの作成
   - Amazon Lightsail でのバケットのリソースアクセスの設定
   - Amazon Lightsail でのバケットのクロスアカウントアクセスの設定
5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。

   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定

6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail のバケットにファイルをアップロードする
   - マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
   - Amazon Lightsail のバケット内のオブジェクトの表示
   - Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
   - Amazon Lightsail のバケットからのオブジェクトのダウンロード
   - Amazon Lightsail のバケット内のオブジェクトのフィルタリング
   - Amazon Lightsail のバケット内のオブジェクトのタグ付け
   - Amazon Lightsail のバケット内のオブジェクトの削除
9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。
11. バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。
12. バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。
13. ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。
14. バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。
   - チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
   - チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用

15.使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

Lightsail バケット内のオブジェクトにタグを付ける

バケット内のオブジェクトにタグ付けして、目的、所有者、環境、またはその他の基準で分類します。タグは、アップロード時またはアップロード後にオブジェクトに追加できます。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

Lightsail コンソールを使用してオブジェクトのタグを追加および削除

以下の手順で、Lightsail コンソールを使ってバケット内のオブジェクトにタグを追加・削除します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. オブジェクトにタグ付けするバケットの名前を選択します。
4. [オブジェクト]タブの[オブジェクト ブラウザ] ペインを使って、オブジェクトの場所を参照します。
5. タグを追加または削除するオブジェクトの横に、チェックマークを付けます。
6. [オブジェクト情報]ペインの[オブジェクトタグ] セクションで、次のいずれかのオプションを選択します。

   - [追加] または[編集] (タグが追加済みの場合)。[キー] テキストボックスにキーを入力し、[値] テキストボックスに値を入力します。次に、保存 を選択して、タグを追加します。または、[キャンセル] を選択します。
   - [編集] し、削除するキーバリュー タグの横にある [X] を選択します。タグの削除が完了したら [保存] を選択し、タグを削除しない場合は[キャンセル] を選択します。

を使用してオブジェクトのタグを追加および削除する AWS CLI

AWS Command Line Interface () を使用してオブジェクトにタグを追加したり、オブジェクトからタグを削除したりするには、次の手順を実行しますAWS CLI。これを行うには、`put-object-tagging` と`delete-object-tagging` コマンドを使用します。詳細については、「AWS CLI コマンドリファレンス」の「put-object-tagging」 および 「delete-object-tagging」 を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 次のいずれかのコマンドを入力します。

   - オブジェクトにタグを追加するには

     ```
     aws s3api put-object-tagging --bucket BucketName --key ObjectKey --tagging
      "{\"TagSet\":[{ \"Key\": \"KeyTag\", \"Value\": \"ValueTag\" }]}"
     ```

     コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

     - *BucketName* - タグ付けするオブジェクトを含むバケットの名前。
     - *ObjectKey* - タグ付けするオブジェクトの完全なオブジェクトキー。
     - *KeyTag* - タグのキー値。
     - *ValueTag* - タグの値。
   - オブジェクトにタグを追加するには

     ```
     aws s3api put-object-tagging --bucket BucketName --key ObjectKey --tagging
      "{\"TagSet\":[{ \"Key\": \"KeyTag1\", \"Value\": \"ValueTag1\" }, { \"Key\":
      \"KeyTag2\", \"Value\": \"ValueTag2\" }]}"
     ```

     コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

     - *BucketName* - タグ付けするオブジェクトを含むバケットの名前。
     - *ObjectKey* - タグ付けするオブジェクトの完全なオブジェクトキー。
     - *KeyTag1* - 最初のタグのキーバリュー。
     - *ValueTag1* - 最初のタグの値。
     - *KeyTag2* - 2番目のタグのキーバリュー。
     - *ValueTag2* - 2番目のタグの値。
   - オブジェクトからタグを削除します。

```
aws s3api delete-object-tagging --bucket BucketName --key ObjectKey
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *BucketName* - すべてのタグを削除するオブジェクトが含まれているバケットの名前。。
- *ObjectKey* - タグ付けするオブジェクトの完全なオブジェクトキー。

例:

```
aws s3api delete-object --bucket amzn-s3-demo-bucket --key nptLmg6jqDo.jpg --
tagging "{\"TagSet\":[{ \"Key\": \"Importance\", \"Value\": \"High\" }]}"
```

以下の例のような結果が表示されるはずです。

```
C:\>aws s3api put-object-tagging --bucket DOC-EXAMPLE-BUCKET  --key nptLmg6jqDo.jpg
--tagging "{\"TagSet\":[{ \"Key\": \"Importance\", \"Value\": \"High\" }]}"
{
    "VersionId": "9nL2d41NuZdhdk4HS3kZIwOxJeS1kCkm"
}
```

バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。
3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オ

ブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

- Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
- Amazon Lightsail でのバケットのアクセス許可の設定
- Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
- Amazon Lightsail でのバケットのアクセスキーの作成
- Amazon Lightsail でのバケットのリソースアクセスの設定
- Amazon Lightsail でのバケットのクロスアカウントアクセスの設定

5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定
6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail のバケットにファイルをアップロードする
   - マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
   - Amazon Lightsail のバケット内のオブジェクトの表示
   - Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
   - Amazon Lightsail のバケットからのオブジェクトのダウンロード
   - Amazon Lightsail のバケット内のオブジェクトのフィルタリング
   - Amazon Lightsail のバケット内のオブジェクトのタグ付け

- Amazon Lightsail のバケット内のオブジェクトの削除

9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。
11. バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。
12. バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。
13. ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。
14. バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。
    - チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
    - チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用
15. 使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

インスタンスの Lightsail バケットへのアクセスをコントロールする

Amazon Lightsail インスタンスを Lightsail バケットに添付して、バケットとそのオブジェクトに全プログラムアクセスを許可します。インスタンスをバケットにアタッチする場合、アクセスキーなどの認証情報を管理する必要はありません。アタッチするインスタンスおよびバケットは同じ AWS リージョンに存在する必要があります。インスタンスを別のリージョンにあるバケットにアタッチすることはできません。

リソースアクセスは、バケットに直接ファイルをアップロードするようにインスタンスでソフトウェアまたはプラグインが設定されている場合に適しています。例えば、バケットにメディアファイルを

保存するように WordPress インスタンスを設定する場合。詳細については、「チュートリアル: バケットを WordPress インスタンスに接続する」を参照してください。

許可のオプションの詳細については、「バケットのアクセス許可」を参照してください。セキュリティのベストプラクティスの詳細については、「オブジェクトストレージのセキュリティのベストプラクティス」を参照してください。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

バケットのリソースアクセスの設定

バケットのリソースアクセスを設定するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. リソースアクセスを設定するバケット名を選択します。
4. [Permissions] (許可) タブを選択します。

   リソースアクセスセクションには、バケットに現在添付されているインスタンスが表示されます。(存在する場合)
5. [Attach instance] (インスタンスをアタッチ) を選択してインスタンスをバケットにアタッチします。
6. [Select an instance] (インスタンスを選択) ドロップダウンメニューで、バケットにアタッチするインスタンスを選択します。

   > **Note**
   >
   > 実行中または停止状態のインスタンスのみをアタッチできます。さらに、バケット AWS リージョン と同じ にあるインスタンスのみをアタッチできます。
7. [Attach] (アタッチ) を選択してインスタンスをアタッチします。または、[キャンセル] を選択します。

   インスタンスは添付された後、バケットとそのオブジェクトへの全アクセス許可があります。インスタンスでソフトウェアまたはプラグインを設定して、プログラムでバケット上のファイルにアクセスしたりアップロードをすることが可能です。例えば、バケットにメディアファイルを保存するように WordPress インスタンスを設定する場合。詳細については、「チュートリアル: バケットを WordPress インスタンスに接続する」を参照してください。

使用状況の変動に応じて Lightsail バケットストレージプランを調整します。

Amazon Lightsail オブジェクトストレージサービスでは、バケットのストレージプランによって、月額コスト、ストレージクォータ、およびデータ転送クォータが変わります。バケットのストレージプランを更新できるのは、毎月の AWS 請求サイクル内で 1 回のみです。バケットのストレージプランを変更すると、ストレージおよびネットワーク転送クォータがリセットされます。ただし、以前のストレージプランを使用して発生したストレージ容量とデータ転送超過料金は対象外となります。

バケットのストレージプランがストレージまたはデータ転送クォータを一貫して上回っている場合、またはバケットの使用量が一貫してクォータを下回っている場合は、バケットのストレージプランを変更しましょう。バケットの使用量の変動が予測できない場合があるため、バケットのストレージプランは、短期的な月々のコスト削減策としてではなく、長期的な戦略としてのみ変更をすることをお勧めします。長期間バケットに十分なストレージ容量とデータ転送クォータを提供するストレージプランを選択しましょう。

バケットについての詳細は、「オブジェクトストレージ」を参照してください。

Lightsail コンソールを使用してバケットのストレージプランを変更する

Lightsail コンソールを使ってバケットのストレージプランを変更するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. プランを変更するバケットの名前を選択します。
4. バケット管理ページで メトリクス タブを選択します。
5. ストレージプランを変更するを選択します。
6. 表示される確認プロンプトで、はい、変更しますを選択して、バケットのストレージプランの変更を続行します。それ以外の場合は、[キャンセル] を選択します。
7. 使用したいプランを選択し、プラン選択を選択します。
8. 表示される確認プロンプトで、はい、適用しますを選択してバケットの変更を適用するか、いいえ、戻るを選択して適用をしないようにします。

を使用してバケットのストレージプランを変更する AWS CLI

AWS Command Line Interface () を使用してバケットのプランを変更するには、次の手順を実行しますAWS CLI。これを行うには、update-bucket-bundle コマンドを使用します。API において「バケットストレージプラン」は「バケットバンドル」と表記されるのでご注意ください。詳細については、「AWS CLI コマンドリファレンス」の「update-bucket-bundle」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. バケットのプランを変更するには、以下のコマンドを入力します。

```
aws lightsail update-bucket-bundle --bucket-name BucketName --bundle-id BundleID
```

コマンドで、以下のサンプルテキストを独自のテキストに置き換えます。

- *BucketName* - 変更したいストレージプランのバケットの名前
- *BundleId* - 適用したいバケットの新しいバケットバンドルの ID。get-bucket-bundles コマンドを使用して、使用可能なバケットバンドルとその ID を一覧表示します。詳細については、「AWS CLI コマンドリファレンス」の「get-bucket-bundles」を参照してください。

例:

```
aws lightsail update-bucket-bundle --bucket-name amzn-s3-demo-bucket --bundle-
id medium_1_0
```

以下の例のような結果が表示されるはずです。

```
C:\>aws lightsail update-bucket-bundle --bucket-name DOC-EXAMPLE-BUCKET --bundle-id medium_1_0

{
    "operations": [
        {
            "id": "8example-8176-48bd-b1da-exampleb8404",
            "resourceName": "DOC-EXAMPLE-BUCKET",
            "resourceType": "Bucket",
            "createdAt": "2021-06-30T12:05:57.362000-07:00",
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "62example362/DOC-EXAMPLE-BUCKET/medium_1_0",
            "operationType": "UpdateBucketBundle",
            "status": "Succeeded",
            "statusChangedAt": "2021-06-30T12:05:57.362000-07:00",
            "errorCode": "",
            "errorDetails": ""
        }
    ]
}
```

セキュリティを強化するための Lightsail バケットアクセス許可を管理する

バケットのアクセス許可を使用して、バケット内のオブジェクトへのパブリック (認証されていない) 読み取り専用アクセスを制御します。バケットはプライベートまたはパブリック (読み取り専用) にすることができます。また、バケットをプライベートにしつつ、個々のオブジェクトをパブリック (読み取り専用) にするオプションもあります。

> **Important**
>
> バケットをパブリック (読み取り専用) にすると、バケット内のすべてのオブジェクトが、バケットの URL を介してインターネット上の誰でも読み取り可能になります。 (例えば、`https://amzn-s3-demo-bucket.us-east-1.amazonaws.com/media/sailbot.jpg`) インターネット上の誰にもオブジェクトへのアクセスを許可したくない場合は、バケットをパブリック (読み取り専用) にしないでください。

許可のオプションの詳細については、「バケットのアクセス許可」を参照してください。セキュリティのベストプラクティスの詳細については、「オブジェクトストレージのセキュリティのベストプ

ラクティス」を参照してください。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

> **Important**
>
> Lightsail オブジェクトストレージリソースは、パブリックアクセスを許可または拒否するときに、Lightsail バケットのアクセス許可と Amazon S3 アカウントレベルのブロックパブリックアクセス設定の両方を考慮するようになります。詳細については、「バケットに対するブロックパブリックアクセス」を参照してください。

バケットのアクセス許可設定

バケットのアクセス許可を設定する手順は以下を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. アクセス権限を設定するバケットの名前を選択します。
4. [Permissions (許可)] タブを選択します。

   ページの [Bucket access permissions] (バケットアクセス許可) セクションに、バケットの現在設定されているアクセス権限が表示されます。
5. [Change permission] (権限の変更) を選択して、バケットのアクセス権限を変更します。
6. 以下のオプションのいずれかを選択してください。

   - すべてのオブジェクトはプライベートです – バケット内のすべてのオブジェクトは、ご自身またはアクセスを許可したユーザーのみが読み取ることができます。
   - 個々のオブジェクトを公開可能にする (読み取り専用) — バケット内のオブジェクトは、パブリックにする個々のオブジェクト (読み取り専用) を指定しない限り、ご自身またはアクセスを許可したユーザーのみが読み取ることができます。個々のオブジェクトのアクセス許可の詳細については「バケット内の個々のオブジェクトに対するアクセス許可の設定」を参照してください。

     [個々のオブジェクトを公開可能にする (読み取り専用)] オプションは、バケット内のオブジェクトの一部を公開する必要があり、その他全てをプライベートにする場合にお勧めします。例えば、一部の WordPress プラグインでは、バケットの個々のオブジェクトを公開する必要があります。詳細については、「チュートリアル: バケットを WordPress インスタンスに接

続する」および「チュートリアル: バケットをコンテンツ配信ネットワークディストリビューションと使用する」を参照してください。

- すべてのオブジェクトを公開 (読み取り専用) — バケット内のすべてのオブジェクトは、インターネット上の誰でも読み取り可能です。

> **Important**
>
> バケットをパブリック (読み取り専用) にすると、バケット内のすべてのオブジェクトが、バケットの URL を介してインターネット上の誰でも読み取り可能になります。 (例えば、`https://amzn-s3-demo-bucket.us-east-1.amazonaws.com/media/sailbot.jpg`) インターネット上の誰にもオブジェクトへのアクセスを許可したくない場合は、バケットをパブリック (読み取り専用) にしないでください。

7. [保存] を選択して変更を保存します。または、[キャンセル] を選択します。

   変更したバケットアクセス許可に応じて、以下の変更が適用されます。

   - すべてのオブジェクトはプライベート - バケット内のすべてのオブジェクトは、個々のオブジェクトのアクセス許可がパブリック (読み取り専用)に以前設定されていても、プライベートに設定されます。
   - 個々のオブジェクトを公開可能にする (読み取り専用) - 個々のオブジェクトのアクセス許可がパブリック (読み取り専用)に以前設定されていたのがパブリックに設定されます。オブジェクトに対して個々のオブジェクトにアクセス許可を設定することが可能になります。
   - オブジェクトを全て公開 (読み取り専用) - バケット内のすべてのオブジェクトは、個々のオブジェクトのアクセス許可がプライベートに以前設定されていても、パブリックに設定されます。

   個々のオブジェクトのアクセス許可の詳細については「バケット内の個々のオブジェクトに対するアクセス許可の設定」を参照してください。

AWS アカウント間でLightsailバケットへの読み取り専用アクセスを付与する

クロスアカウントアクセスを使用して、他の AWS アカウントとそのユーザーに対してバケット内のすべてのオブジェクトに読み取り専用アクセスを許可します。クロスアカウントアクセスは、オブジェクトを別の AWS アカウントと共有する場合に最適です。別のアカウントにクロスアカウン

トアクセスを許可すると AWS 、そのアカウントのユーザーは、バケットの URL とオブジェクト (例:) を介してバケット内のオブジェクトへの読み取り専用アクセスを持ちますhttps://amzn-s3-demo-bucket.us-east-1.amazonaws.com/media/sailbot.jpg。バケットに最大 10 個の AWS アカウントへのアクセスを許可できます。

許可のオプションの詳細については、「バケットのアクセス許可」を参照してください。セキュリティのベストプラクティスの詳細については、「オブジェクトストレージのセキュリティのベストプラクティス」を参照してください。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

バケットのクロスアカウントアクセスの設定

バケットのクロスアカウントアクセスを設定するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. クロスアカウントアクセスを設定するバケット名を選択します。
4. [アクセス許可] タブを選択します。

   ページの [クロスアカウントアクセス] セクションに、バケットにアクセスするように現在設定されている AWS アカウント ID が表示されています (存在する場合)。
5. クロスアカウントアクセスの追加 を選択して、別の AWS アカウントのバケットへのアクセスを許可します。
6. アクセスを許可する AWS アカウントの ID をアカウント ID テキストボックスに入力します。
7. 保存を選択してアクセスを許可します。それ以外の場合は、[キャンセル] を選択します。

   追加した AWS アカウント ID は、ページのクロスアカウントアクセスセクションに表示されます。 AWS アカウントのクロスアカウントアクセスを削除するには、削除する AWS アカウント ID の隣にある、削除 (ゴミ箱) アイコンを選択します。

Amazon Lightsail 内の単一バケットオブジェクトへのパブリックアクセスを付与する

個々のオブジェクトアクセス許可を使用して、認証なしで公開されたバケット内の個々のオブジェクトの読み取り専用アクセスを制御します。バケットはプライベートまたはパブリック (読み取り専用) に設定することができます。

> **Important**
>
> 個々のオブジェクトアクセス許可は、バケットのアクセス許可が個々のオブジェクトを公開可能にする (読み取り専用) に設定されている場合のみ設定が可能です。バケット許可のオプションの詳細については、「バケットのアクセス許可」を参照してください。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

個々のオブジェクトへのアクセス許可の設定は、バケット内のオブジェクトの一部を公開し、その他全てをプライベートにする必要があるなどの、特殊な場合にのみ行うことをお勧めします。例えば、一部の WordPress プラグインでは、バケットの個々のオブジェクトを公開する必要があります。詳細については、「チュートリアル: バケットを WordPress インスタンスに接続する」および「チュートリアル: バケットをコンテンツ配信ネットワークディストリビューションと使用する」を参照してください。

許可のオプションの詳細については、「バケットのアクセス許可」を参照してください。セキュリティのベストプラクティスの詳細については、「オブジェクトストレージのセキュリティのベストプラクティス」を参照してください。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

個々のオブジェクトのアクセス許可の設定

バケットの個々のオブジェクトのアクセス許可を設定するには、以下の手順を実行します。Lightsail でバケットを管理する権限をユーザーに付与する IAM ポリシーの例については、「バケットを管理する IAM ポリシー」を参照してください。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. 個々のオブジェクトのアクセス許可を設定するバケット名を選択します。
4. [オブジェクト] タブを選択します。
5. アクセス許可を設定するオブジェクトの横にチェックマークを追加します。

   オブジェク情報ペインがオブジェクトの現在のアクセス許可の状態を表示します。
6. オブジェクト情報ペインの [アクセス許可] セクションで [編集] を選択して、オブジェクトのアクセス許可を変更します。

> **Note**
>
> 編集オプションが使用できない場合は、そのバケットのアクセス許可では、個々のオブジェクトアクセス許可を設定することができないことを意味します。個々のオブジェクトアクセス許可を設定するには、バケットアクセス許可を「個々のオブジェクトを公開可能にする (読み取り専用) 」に設定する必要があります。詳細については、「バケットのアクセス許可を設定する」を参照してください。

7. [アクセス許可の選択] のドロップダウンメニューから以下のいずれかのオプションを選択します。

   - プライベート – オブジェクトはご自身とアクセスを許可したユーザーのみが読み取ることができます。
   - パブリック (読み取り専用) – オブジェクトは世界中の誰もが読み取ることができます。

8. [保存] を選択して変更を保存します。または、[キャンセル] を選択します。

   バケットのバケットアクセス許可設定は、個々のオブジェクトのアクセス許可に以下の影響を与えます。

   - バケットのアクセス許可が「すべてのオブジェクトはプライベートです」に設定されている場合、個々のオブジェクトのアクセス許可がパブリック (読み取り専用)に以前設定されていても、バケット内のすべてのオブジェクトはプライベートに設定されます。ただし、以前設定されていた個々のオブジェクトのアクセス許可は保持されます。例えば、バケットのアクセス許可を「個々のオブジェクトを公開可能にする (読み取り専用) 」に戻すと、すべてのオブジェクトで個々のアクセス許可がパブリック (読み取り専用) 再びパブリックになります。
   - バケットのアクセス許可が「すべてのオブジェクトをパブリック (読み取り専用) にする」に設定されている場合、個々のオブジェクトのアクセス許可がプライベートに以前設定されていても、バケット内の全てのオブジェクトはパブリックに設定されます。

     バケットアクセス許可の詳細については、「バケットアクセス許可を設定する」を参照してください。

Lightsail マルチパートアップロードによるバケットへのファイルのアップロード

マルチパートアップロードを使用すると、単一のファイルをパートのセットとしてバケットにアップロードできます。各パートは、ファイルのデータの連続する部分です。これらのファイルパートは、任意の順序で個別にアップロードできます。いずれかのパートの送信が失敗すると、他のパートに影響を与えることなくそのパートを再送することができます。ファイルのすべてのパートがアップロードされると、Amazon S3 はこれらのパートをアセンブルし、Amazon Lightsail のバケットにオブジェクトを作成します。通常、オブジェクトサイズが 100 MB 以上の場合は、単一のオペレーションでオブジェクトをアップロードする代わりに、マルチパートアップロードを使用することを考慮してください。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

マルチパートアップロードの使用には、次の利点があります。

- スループットの向上 – パートを並列にアップロードすることで、スループットを向上させることができます。
- ネットワークの問題からの素早い回復 – パートサイズは比較的小さいため、ネットワークエラーにより失敗したアップロードを再開する際の影響を最小限に抑えることができます。
- 時間をおいてアップロード - ファイルのパートを時間をおいてアップロードできます。マルチパートアップロードを開始してから、24 時間以内にマルチパートアップロードを完了します。
- ファイルの最終的なサイズが不明な状態でアップロードを開始 – ファイルの作成中でもアップロードを開始できます。

次の方法でマルチパートアップロードを使用することをお勧めします。

- 安定した高帯域幅ネットワーク経由でラージファイルをアップロードする場合は、複数スレッドのパフォーマンスのために並行してファイルパートをアップロードする「マルチパートアップロード」を使用すると、可能な帯域幅の使用を最大化します。
- むらがあるネットワークでアップロードを実行する場合は、マルチパートアップロードを使用して、アップロードの再開を回避することで、ネットワークエラーに対する弾力性を高めます。マルチパートアップロードを使用している場合、中断されたパートのみを対象にアップロードを再試行します。最初からやり直したり、ファイル全体を再度アップロードする必要はありません。

目次

- マルチパートアップロードのプロセス

- マルチパートアップロードの同時オペレーション
- マルチパートアップロードの保持期間
- Amazon シンプルストレージサービスのマルチパートアップロード制限
- アップロードするファイルを分割
- を使用してマルチパートアップロードを開始する AWS CLI
- を使用してパートをアップロードする AWS CLI
- を使用してマルチパートアップロードの一部を一覧表示する AWS CLI
- マルチパートアップロード .json ファイルの作成
- を使用してマルチパートアップロードを完了する AWS CLI
- を使用してバケットのマルチパートアップロードを一覧表示する AWS CLI
- AWS CLIを使用したマルチパートアップロードの停止

マルチパートアップロードのプロセス

マルチパートアップロードには、Amazon S3 アクションを使用して、Lightsail のバケットにファイルをアップロードする 3 つのステップがあります。

1. CreateMultipartUploadアクションを使って、マルチパートアップロードを開始します。
2. UploadPartアクションを使ってファイルパートをアップロードします。
3. CompleteMultipartUploadアクションを使ってマルチパートアップロードを実行します。

> **Note**
>
> AbortMultipartUploadアクションを使用して、マルチパートアップロードを開始した後に停止できます。

マルチパートアップロードのリクエストが完了すると、Amazon シンプルストレージサービスはアップロードされたパートからオブジェクトを構築します。その後、バケット内の他のオブジェクトにアクセスするのと同じ方法で、オブジェクトにアクセスできます。

進行中のすべてのマルチパートアップロードをリストしたり、特定のマルチパートアップロードにおいてアップロードが完了したパートのリスト表示を取得したりできます。このようなオペレーションのそれぞれについて、このセクションで説明します。

マルチパートアップロードの開始

マルチパートアップロードを開始するリクエストを送信すると、Amazon シンプルストレージサービスはアップロード ID を含むレスポンスを返します。これは、マルチパートアップロードの一意の識別子です。パートのアップロード、パートのリスト、アップロードの完了、アップロードの停止を行うときは常に、アップロード ID を指定する必要があります。アップロードされるオブジェクトを説明するメタデータを提供したい場合は、マルチパートアップロードを開始するリクエストでメタデータを指定する必要があります。

パートのアップロード

パートをアップロードするときは、アップロード ID に加えて、パート番号を指定する必要があります。1 ~ 10,000 の範囲で任意のパート番号を選択できます。パート番号によって、アップロードするオブジェクトに含まれるパートとその位置が一意に識別されます。選択するパート番号は、連続している必要はありません (例えば、1、5、14 など)。以前にアップロードしたパートと同じパート番号を使って新しいパートをアップロードした場合、以前のパートは上書きされます。

パートをアップロードするたびに Amazon シンプルストレージサービスはレスポンスに ETag ヘッダーを返します。パートのアップロードごとに、パート番号と ETag 値を記録する必要があります。マルチパートアップロードを完了するためには、残りのリクエストにこれらの値を含める必要があります。

> **Note**
>
> マルチパートアップロードのすべてのアップロードされたパートは、バケットに保存されます。アップロードを完了するか、アップロードを停止するか、アップロードがタイムアウトするまで、バケットのストレージ容量を消費します。詳細については、このガイドで後述する「マルチパートアップロードの保持期間」を参照してください。

マルチパートアップロードの完了

マルチパートアップロードを完了すると、パート番号に基づいて昇順に連結されたオブジェクトが Amazon シンプルストレージサービスによって作成されます。マルチパートアップロードの開始リクエストにオブジェクトメタデータが提供されている場合、Amazon シンプルストレージサービスによってそのメタデータはオブジェクトに関連付けられます。完了リクエストが正常に処理されると、個々のパートはなくなります。

マルチパートアップロードの完了リクエストには、アップロード ID と、パート番号およびそれに対応する ETag 値の両方のリストが含まれている必要があります。Amazon シンプルストレージサービ

スからのレスポンスには、結合されるオブジェクトデータを一意に識別する ETag が含まれます。この ETag が、オブジェクトデータの MD5 ハッシュになるとは限りません。

マルチパートアップロードは停止することもできます。マルチパートアップロードを停止した後は、再度同じアップロード ID を使ってパートをアップロードすることはできません。キャンセルされたマルチパートアップロードの任意の部分のすべてのストレージが解放されます。パートのアップロードが進行しているときにマルチパートアップロードを停止した場合は、停止後もそのパートのアップロードは成功または失敗する可能性があります。すべてのパートによって使用されているストレージを全部解放するには、すべてのパートのアップロードが完了した後で初めてマルチパートアップロードを停止する必要があります。

マルチパートアップロードのリスト化

特定のマルチパートアップロードのパートや、進行中のすべてのマルチパートアップロードをリスト表示できます。パートのリストオペレーションでは、特定のマルチパートアップロードについて既にアップロードしたパートの情報が返されます。パートのリストリクエストを送信するたびに、指定したマルチパートアップロードのパート情報 (最大で 1,000 個のパート) が Amazon シンプルストレージサービスから返されます。マルチパートアップロードに 1,000 個を超えるパートが含まれる場合、すべてのパートを取得するにはパートのリストリクエストを追加で送信する必要があります。返されるパートのリストには、アップロード中のパートは含まれていないことに注意してください。マルチパートアップロードのリストオペレーションを使用すると、進行中のマルチパートアップロードのリストを取得できます。

進行中のマルチパートアップロードとは、開始されているものの、まだ完了または停止されていないアップロードを意味します。各リクエストに最大 1,000 個のマルチパートアップロードが返されます。進行中のマルチパートアップロードが 1,000 個を超える場合、残りのマルチパートアップロードを取得するには、リクエストを追加で送信する必要があります。返されるリストは確認の目的でのみ使用します。マルチパートアップロードの完了リクエストを送信するときに、リストの結果を使用しないでください。その代わりに、パートのアップロード時に指定したパート番号と、それに対応する、Amazon シンプルストレージサービスから返される ETag 値で構成される独自のリストを維持しておいてください。

マルチパートアップロードの同時オペレーション

分散開発環境においては、アプリケーションから同じオブジェクトに対して複数の更新が同時に開始されることもありえます。同じオブジェクトキーを使ってアプリケーションから複数のマルチパートアップロードが開始される可能性もあります。そのようなアップロードごとに、アプリケーションからパートのアップロードが行われ、アップロードの完了リクエストが Amazon シンプルストレージ

サービスに送信されて、オブジェクトが作成されます。バケットでバージョニングが有効になっている場合は、マルチパートアップロードを完了するたびに新しいバージョンが作成されます。バージョニングが有効になっていないバケットの場合は、マルチパートアップロードの開始から完了までの間に受信されたリクエストなど、他のリクエストが優先される可能性もあります。

> **Note**
>
> マルチパートアップロードを開始してから完了する前に受信したリクエストなど、他のリクエストが優先される可能性があります。たとえば、あるキーを使ってマルチパートアップロードを開始した後、マルチパートアップロードが完了しないうちに別のオペレーションによってそのキーが削除されることがあります。この場合、マルチパートアップロードの完了レスポンスによって、オブジェクトを確認できなくても、オブジェクト作成の成功が示される可能性があります。

マルチパートアップロードの保持期間

マルチパートアップロードのすべてのアップロードパートは、バケットに保存されます。アップロードを完了するか、アップロードを停止するか、またはアップロードがタイムアウトするまで、バケットのストレージ容量を消費します。マルチパートアップロードはタイムアウトになり、マルチパートアップロードが作成されてから 24 時間後に削除されます。マルチパートアップロードを停止するか、タイムアウトすると、アップロードされたすべてのパートが削除され、バケットで使用するために使用したストレージ領域が解放されます。

Amazon シンプルストレージサービスのマルチパートアップロード制限

次の表は、マルチパートアップロードの主な仕様をまとめたものです。

- 最大オブジェクトサイズ: 50 TB
- アップロードあたりの最大パート数：10,000
- パート番号：1 ~ 10,000（両端を含む）
- パートサイズ：5 MB（最小）- 5 GB（最大）マルチパートアップロードの最後のパートには、サイズの制限はありません。
- パートのリストリクエストで返されるパートの最大数：1,000
- マルチパートアップロードのリストリクエストで返されるマルチパートアップロードの最大数：1,000

アップロードするファイルを分割します。

Linux または Unix オペレーティングシステムでsplitコマンドを使用して、ファイルを複数のパートに分割し、バケットにアップロードします。Windows オペレーティングシステムでファイルを分割するために使用できる同様のフリーウェア アプリケーションがあります。ファイルを複数のパートに分割した後、本ガイドの「マルチパートアップロードの開始」セクションに進んでください。

を使用してマルチパートアップロードを開始する AWS CLI

AWS Command Line Interface (AWS CLI) を使用してマルチパートアップロードを開始するには、以下の手順を実行してください。これは、create-multipart-upload コマンドを使用して行います。詳細については、「AWS CLI コマンドリファレンス」の「create-multipart-upload」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 次のコマンドを入力して、バケットのマルチパートアップロードを作成します。

```
aws s3api create-multipart-upload --bucket BucketName --key ObjectKey --acl bucket-owner-full-control
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *BucketName*#- マルチパートアップロードを作成するバケットの名前。
- *ObjectKey*#- アップロードするファイルに使用するオブジェクトキー。

例:

```
aws s3api create-multipart-upload --bucket amzn-s3-demo-bucket --key sailbot.mp4 --acl bucket-owner-full-control
```

次の例のような結果が表示されます。レスポンスにはUploadIDが含まれており、以降のコマンドでパーツをアップロードしたり、このオブジェクトのマルチパートアップロードを完了させるために指定する必要があります。

```
C:\ >aws s3api create-multipart-upload --bucket DOC-EXAMPLE-BUCKET  --key  sailbot.mp4
{
    "AbortDate": "2021-05-20T00:00:00+00:00",
    "AbortRuleId": "ExpireMultiPart",
    "ServerSideEncryption": "AES256",
    "Bucket": "DOC-EXAMPLE-BUCKET",
    "Key": "sailbot.mp4",
    "UploadId": "R4QU.mO.exampleiHWiLOeNw7JtXX7OotRhTLsXXCzF21CZdYlfj5lfjtiMnpzVw2WPj.exampleBTmL_N_.42.DlHYOTsITFsX.tO3XOUTTAHiCxY5VR8jWRGdkVkUG"
}
```

マルチパートアップロード用のUploadID ができたら、このガイドの「AWS CLIを使用してパートをアップロードする」のセクションに進み、パートのアップロードを開始します。

を使用してパートをアップロードする AWS CLI

AWS Command Line Interface (AWS CLI)を使用して、マルチパートアップロードのパートをアップロードするには、以下の手順を実行してください。これは、upload-part コマンドを使用して行います。詳細については、「AWS CLI コマンドリファレンス」の「upload-part」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 以下のコマンドを入力して、パートをバケットにアップロードします。

```
aws s3api upload-part --bucket BucketName --key ObjectKey --part-number Number --
body FilePart --upload-id "UploadID"  --acl bucket-owner-full-control
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *BucketName*#- マルチパートアップロードを作成するバケットの名前。
- *ObjectKey*#- アップロードするファイルに使用するオブジェクトキー。

- *Number* - アップロードするパートのパート番号。パート番号によって、アップロードするオブジェクトに含まれるパートとその位置が一意に識別されます。アップロードするパートごとに、--part-number パラメータを段階的に増やしてください。そのためには、マルチパートアップロードの完了時に Amazon Simple Storage Service がオブジェクトをアセンブルする順序で番号を付けてください。
- *FilePart* - コンピュータからアップロードするパートファイル。
- *UploadID* - このガイドで前半で作成したマルチパートアップロードのアップロード IDです。

例:

```
aws s3api upload-part --bucket amzn-s3-demo-bucket --
key sailbot.mp4 --part-number 1 --body sailbot.mp4.001 --upload-id
 "R4QU.mO.exampleiHWiLOeNw7JtXX7OotRhTLsXXCzF21CZdYlfj5lfjtiMnpzVw2WPj.exampleBTmL_N_.42.Dl
 --acl bucket-owner-full-control
```

次の例のような結果が表示されます。アップロードするパートごとに、upload-part コマンドを繰り返します。パーツのアップロードリクエストの応答には、アップロードしたパートの ETag 値が含まれます。アップロードした各パーツの ETag 値を記録する。このガイドで後述するマルチパートアップロードを完了するには、すべての ETag 値が必要です。

```
C:\ >aws s3api upload-part --bucket DOC-EXAMPLE-BUCKET  --key sailbot.mp4  --part-number 1 --body sailbot.mp4.001
 --upload-id "R4QU.mO.exampleiHWiLOeNw7JtXX7OotRhTLsXXCzF21CZdYlfj5lfjtiMnpzVw2WPj.exampleBTmL_N_.42.DlHYOTsITFsX.tO3XOUTTAHiCxY5VR8jWRGdkVkUG"
{
    "ServerSideEncryption": "AES256",
    "ETag": "\"4example7530246113e837a860a38bbb\""
}
```

AWS CLI を使用したマルチパートアップロードのパートのリスト化

AWS Command Line Interface (AWS CLI)を使用して、マルチパートアップロードのパートをリストにするには、以下の手順を実行してください。これは、list-parts コマンドを使用して行います。詳細については、「AWS CLI コマンドリファレンス」の「list-parts」を参照してください。

マルチパートアップロードでアップロードされたすべてのパーツのETag 値を取得するには、この手順を実行します。これらの値は、このガイドの後半でマルチパートアップロードを完了するために必要となります。ただし、パートのアップロードの応答からすべてのETag 値を記録した場合は、この手順をスキップして、このガイドの「マルチパートアップロード .json ファイルの作成」セクションに進むことができます。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. バケットのマルチパートアップロードのパートをリストにするには、次のコマンドを入力します。

```
aws s3api list-parts --bucket BucketName --key ObjectKey --upload-id "UploadID"
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *BucketName*#- マルチパートアップロードのパートをリストするバケットの名前。
- *ObjectKey*#- マルチパートアップロードのオブジェクトキー。
- *UploadID* - このガイドの前半で作成したマルチパートアップロードのアップロード ID です。

例:

```
aws s3api list-parts --bucket amzn-s3-demo-bucket --key sailbot.mp4 --upload-id
 "R4QU.mO.exampleiHWiLOeNw7JtXX7OotRhTLsXXCzF21CZdYlfj5lfjtiMnpzVw2WPj.exampleBTmL_N_.42.Dl
```

次の例のような結果が表示されます。レスポンスには、マルチパートアップロードでアップロードしたパーツのすべてのパート番号とETag 値がリスト表示されます。これらの値をクリップボードにコピーして、このガイドの「マルチパートアップロード .json の作成」セクションに進みます。

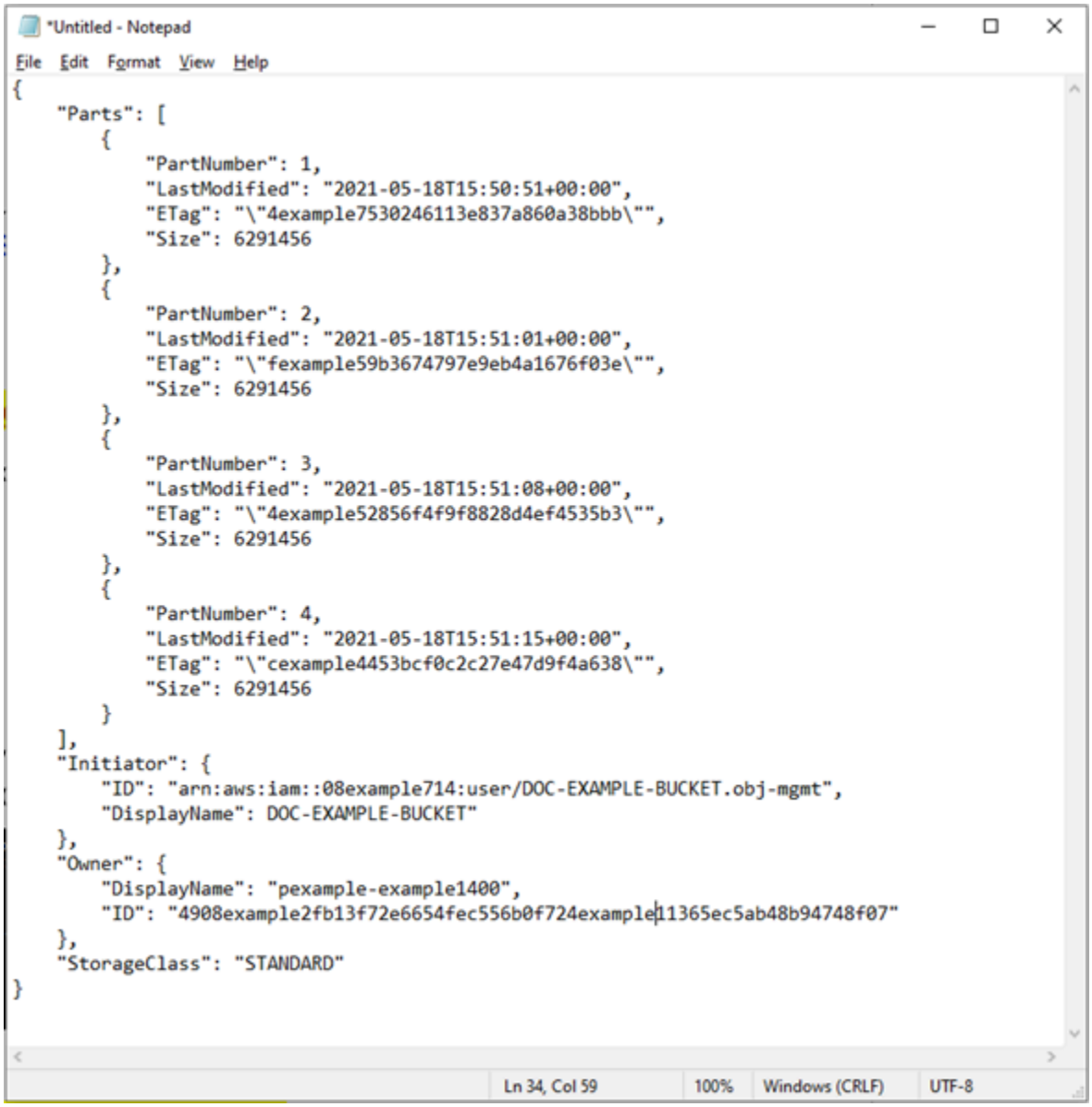
```
C:\>aws s3api list-parts --bucket DOC-EXAMPLE-BUCKET --key sailbot.mp4 --upload-id "R4QU.mO.exampleiHWiLOeNw7JtXX7OotR
hTLsXXCzF21CZdYlfj5lfjtiMnpzVw2WPj.exampleBTmL_N_.42.DlHYOTsITFsX.tO3XOUTTAHiCxY5VR8jWRGdkVkUG"
{
    "Parts": [
        {
            "PartNumber": 1,
            "LastModified": "2021-05-18T15:50:51+00:00",
            "ETag": "\"4example7530246113e837a860a38bbb\"",
            "Size": 6291456
        },
        {
            "PartNumber": 2,
            "LastModified": "2021-05-18T15:51:01+00:00",
            "ETag": "\"fexample59b3674797e9eb4a1676f03e\"",
            "Size": 6291456
        },
        {
            "PartNumber": 3,
            "LastModified": "2021-05-18T15:51:08+00:00",
            "ETag": "\"4example52856f4f9f8828d4ef4535b3\"",
            "Size": 6291456
        },
        {
            "PartNumber": 4,
            "LastModified": "2021-05-18T15:51:15+00:00",
            "ETag": "\"cexample4453bcf0c2c27e47d9f4a638\"",
            "Size": 6291456
        }
    ],
    "Initiator": {
        "ID": "arn:aws:iam::08example714:user/DOC-EXAMPLE-BUCKET.obj-mgmt",
        "DisplayName": "DOC-EXAMPLE-BUCKET"
    },
    "Owner": {
        "DisplayName": "pexample-example1400",
        "ID": "4908example2fb13f72e6654fec556b0f724example11365ec5ab48b94748f07"
    },
    "StorageClass": "STANDARD"
}
```

マルチパートアップロード .json ファイルの作成

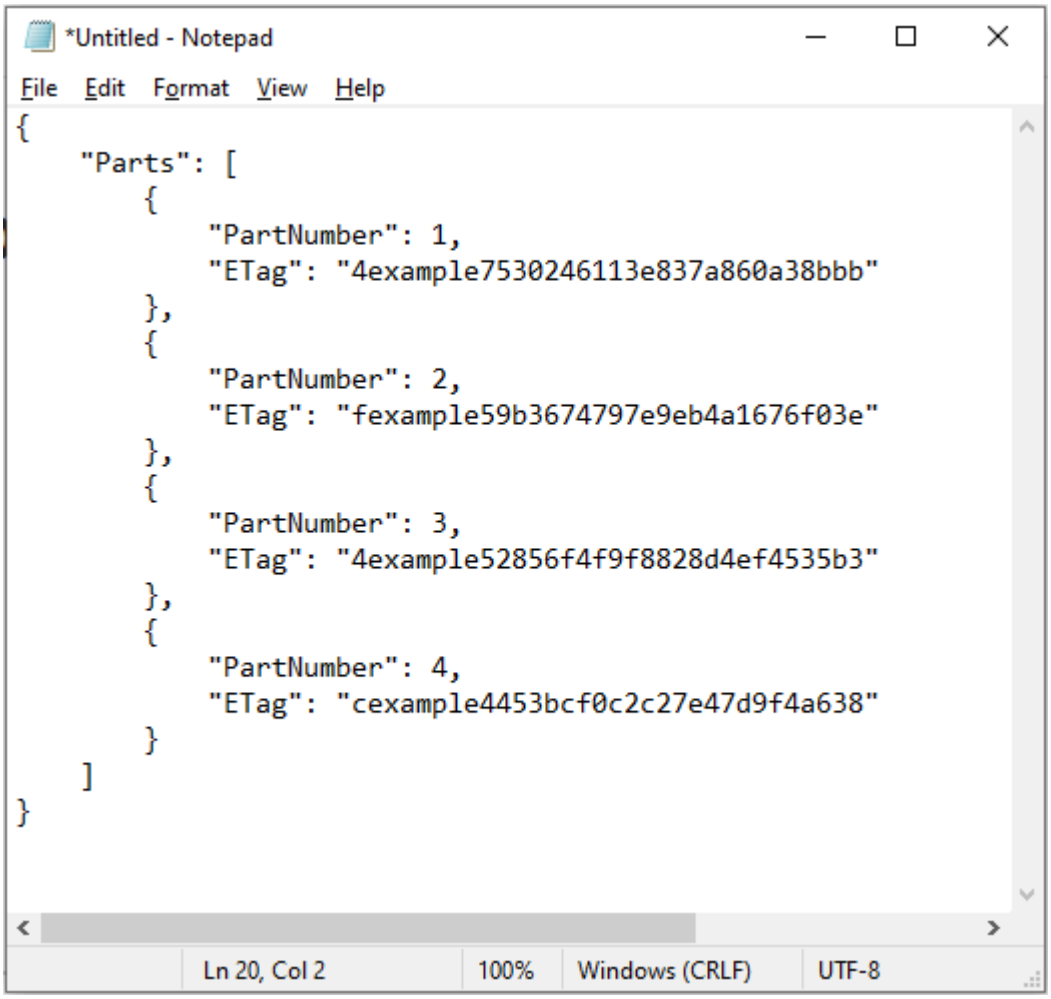
以下の手順で、アップロードしたすべてのパーツとそのETag 値を定義したマルチパートアップロード .json ファイルを作成します。これは、このガイドの後半で、マルチパートアップロードを完了するために必要です。

1. テキストエディターを開き、このガイドの前のセクションでリクエストしたlist-parts コマンドからのレスポンスを貼り付けます。

   結果は次の例のようになります。



2. 次の例に示すように、テキスト ファイルを再フォーマットします。



3. テキスト ファイルを、コンピュータに`mpstructure.json` のように保存し、本ガイドの「AWS CLI を使用したマルチパートアップロードを完了」に進みます。

AWS CLI を使用したマルチパートアップロードの完了

AWS Command Line Interface (AWS CLI) を使用してマルチパートアップロードを完了するには、以下の手順を実行してください。これは、`complete-multipart-upload` コマンドを使用して行います。詳細については、「AWS CLI コマンドリファレンス」の「complete-multipart-upload」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 以下のコマンドを入力して、パートをバケットにアップロードします。

```
aws s3api complete-multipart-upload --multipart-upload file://JSONFileName --
bucket BucketName --key ObjectKey --upload-id "UploadID"  --acl bucket-owner-full-
control
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *JSONFileName*#- このガイドの前半で作成した .json ファイルの名前 (例：mpstructure.json)
- *BucketName*#- マルチパートアップロードを完了するバケットの名前
- *ObjectKey*#- マルチパートアップロードのオブジェクトキー
- *UploadID* - このガイドの前半で作成したマルチパートアップロードのアップロード ID

Example:

```
aws s3api complete-multipart-upload --multipart-upload file://mpstructure.json
 --bucket amzn-s3-demo-bucket --key sailbot.mp4 --upload-id
 "R4QU.mO.exampleiHWiLOeNw7JtXX7OotRhTLsXXCzF21CZdYlfj5lfjtiMnpzVw2WPj.exampleBTmL_N_.42.Dl
 --acl bucket-owner-full-control
```

次の例に示すようなレスポンスが表示されます。これにより、マルチパートアップロードが完了したことを確認します。これで、オブジェクトがアセンブルされ、バケットで使用可能になります。

```
C:\>aws s3api complete-multipart-upload --multipart-upload file://mpstructure.json --bucket DOC-EXAMPLE-BUCKET  --key sailbot.mp4
--upload-id "R4QU.mO.exampleiHWiLOeNw7JtXX7OotRhTLsXXCzF21CZdYlfj5lfjtiMnpzVw2WPj.exampleBTmL_N_.42.DlHYOTsITFsX.tO3XOUTTAHiCxY5VR8jWRGdkVkUG"
{
    "ServerSideEncryption": "AES256",
    "VersionId": "MexampleKMdfPQb.2YZHqOvE_T.vSDtY",
    "Location": "https://DOC-EXAMPLE-BUCKET.s3.us-west-2.amazonaws.com/sailbot.mp4",
    "Bucket": "DOC-EXAMPLE-BUCKET",
    "Key": "sailbot.mp4",
    "ETag": "\"1example5964e3115e5d3f3c9a731585-4\""
}
```

AWS CLI を使用した、バケットのマルチパートアップロードのリスト化

AWS Command Line Interface (AWS CLI) を使用してバケットのマルチパートアップロードをリストにするには、以下の手順を実行します。これは、list-multipart-uploads コマンドを使用して

行います。詳細については、「AWS CLI コマンドリファレンス」の「list-multipart-uploads」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 以下のコマンドを入力して、パートをバケットにアップロードします。

```
aws s3api list-multipart-uploads --bucket BucketName
```

コマンドの中で、*BucketName*を、すべてのマルチパートアップロードをリストアップしたいバケットの名前に置き換えます。

例:

```
aws s3api list-multipart-uploads --bucket amzn-s3-demo-bucket
```

次の例に示すようなレスポンスが表示されます。

```
C:\>aws s3api list-multipart-uploads --bucket DOC-EXAMPLE-BUCKET
{
    "Uploads": [
        {
            "UploadId": "R4QU.mO.exampleiHWiLOeNw7JtXX7OotRhTLsXXCzF21CZdYlfj5lfjtiMnpzVw2WPj.exampleBTmL_N_.42.DlHYOTsITFsX.tO3XOUTTAHiCxY5VR8jWRGdkVkUG",
            "Key": "sailbot.mp4",
            "Initiated": "2021-05-18T15:49:11+00:00",
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "pexample-example1400",
                "ID": "4908example2fb13f72e6654fec556b0f724example11365ec5ab48b94748f07"
            },
            "Initiator": {
                "ID": "arn:aws:iam::08example714:user/DOC-EXAMPLE-BUCKET.obj-mgmt",
                "DisplayName": "DOC-EXAMPLE-BUCKET"
            }
        }
    ]
}
```

AWS CLI を使用したマルチパートアップロードの停止

AWS Command Line Interface () を使用してマルチパートアップロードを停止するには、次の手順を実行しますAWS CLI。マルチパートアップロード開始したものの、それを続行したくない場合に、

これを行います。これは、abort-multipart-upload コマンドを使用して行います。詳細については、「AWS CLI コマンドリファレンス」の「abort-multipart-upload」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 以下のコマンドを入力して、パートをバケットにアップロードします。

```
aws s3api abort-multipart-upload --bucket BucketName --key ObjectKey --upload-id
 "UploadID" --acl bucket-owner-full-control
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *BucketName*#- マルチパートアップロードを停止するバケットの名前。
- *ObjectKey*#- マルチパートアップロードのオブジェクトキー。
- *UploadID* - 停止するマルチパートアップロードのアップロード ID。

例:

```
aws s3api abort-multipart-upload --bucket amzn-s3-demo-bucket --key sailbot.mp4 --
upload-id
 "R4QU.mO.exampleiHWiLOeNw7JtXX7OotRhTLsXXCzF21CZdYlfj5lfjtiMnpzVw2WPj.exampleBTmL_N_.42.Dl
 --acl bucket-owner-full-control
```

このコマンドはレスポンスを返しません。以下のコマンドを実行するにはlist-multipart-uploadsコマンドを実行して、マルチパートアップロードが停止したことを確認します。

Lightsail オブジェクトストレージのバケット命名要件に従う

バケットを作成する場合、Amazon Lightsail オブジェクトストレージサービスで名前を指定する必要があります。バケットの名前は、バケットに保存されているオブジェクトにアクセスするときにカスタマーが使用する URL のパートです。たとえば、 amzn-s3-demo-bucketでバケット

に名前を付けるとus-east-1 AWS リージョン、バケットの URL は になりますamzn-s3-demo-bucket.s3.us-east-1.amazonaws.com。作成後にバケットの名前を変更することはできません。指定したバケット名をカスタマーが確認できることに留意してください。Lightsail オブジェクトストレージサービスの詳細については、「オブジェクトストレージ」を参照してください。バケットの作成の詳細については、「バケットの作成」を参照してください。

バケット名は、DNS に準拠している必要があります。このため、Lightsail でのバケットの命名には、次のルールが適用されます。

- バケット名は 3～54 文字の長さにする必要があります。
- バケット名は、小文字、数字、およびハイフン (-) のみで構成できます。
- バケット名は文字または数字で始まる必要があります。
- ハイフン (-) は単語を区切ることができますが、連続して指定することはできません。例えば、doc-example-bucket は許可されますが、doc--example--bucket は許可されません。
- バケット名は、Amazon Simple Storage Service (Amazon S3) のバケットを含む、aws (スタンダード リージョン) パーティション内で固有でなくてはなりません。
- バケット名のプレフィックスは amzn-s3-demo- で始まってはいけません。
- バケット名のプレフィックスは sthree- で始まってはいけません。
- バケット名のプレフィックスは sthree-configurator で始まってはいけません。
- バケット名のサフィックスは -s3alias で終わってはいけません。

バケット名の例

次の例に示すバケット名は有効であり、命名の推奨ガイドラインに従っています。

- docexamplebucket1
- log-delivery-march-2020
- my-hosted-content

次の例に示すバケット名は許可されません。

- doc.example.bucket (句点を含む)
- doc--example--bucket (連続する 2 つのハイフンを含む)
- doc-example-bucket- (ハイフンで終わっている)

Lightsail オブジェクトストレージバケットのキー名

バケットにアップロードしたファイルは、オブジェクトとして Amazon Lightsail オブジェクトストレージサービスに保存されます。オブジェクトキー (またはキー名) によって、バケット内に保存されたオブジェクトは一意に識別されます。このガイドでは、Lightsail コンソールを通して表示される、バケットのフォルダ構造を構成するキー名とキー名のプレフィックスの概念を説明します。バケットについての詳細は、「オブジェクトストレージ」を参照してください。

キー名

Lightsail オブジェクトストレージサービスのデータモデルは、ファイルシステムに表示される階層構造ではなく、フラットな構造を使用します。フォルダとサブフォルダの階層はありません。ただし、キー名のプレフィックスや区切り記号を使用して論理的な階層を暗示できます。Lightsail コンソールはキー名のプレフィックスを使用して、フォルダー構造のオブジェクトを表示します。

バケットに、次のようなオブジェクトキーを持つ 4 つのオブジェクトがあるとします。

- `Development/Projects.xls`
- `Finance/statement1.pdf`
- `Private/taxdocument.pdf`
- `to-dos.doc`

Lightsail コンソールでは、キー名のプレフィックス (`Development/`、`Finance/`、`Private/`) および区切り記号 (`/`) を使用して、フォルダ構造を表します。`to-dos.doc` キーにはプレフィックスがないため、そのオブジェクトはバケットのルートレベルに直接表示されます。Lightsail コンソールの `Development/` フォルダをブラウズすると、`Projects.xls` オブジェクトが表示されます。`Finance/` フォルダに `statement1.pdf` オブジェクト、および `Private/` フォルダに `taxdocument.pdf` オブジェクトが表示されます。

Lightsail コンソールは、キー名のプレフィックスと区切り記号の値をキー名としてゼロバイトのオブジェクトを作成することにより、フォルダーオブジェクトの作成を許可します これらのフォルダオブジェクトはコンソールに表示されません。ただし、他のオブジェクトと同様に動作します。Amazon S3 API、 AWS Command Line Interface (AWS CLI)、または AWS SDKs を使用して表示および操作できます。

オブジェクトキーの命名のガイドライン

オブジェクトキー名には UTF-8 文字を使用できます。ただし、キー名に特定の文字を使用すると、一部のアプリケーションやプロトコルで問題が発生することがあります。以下のガイドラインは、DNS、ウェブセーフ文字、XML パーサー、その他の API とのコンプライアンスを最大化するのに役立ちます。

セーフ文字

以下の文字セットは、一般的にキー名で使用しても安全です。

- アルファベットの文字
  - 0-9
  - a ~ z
  - A ~ Z
- 特殊文字
  - スラッシュ (/)
  - 感嘆符 (!)
  - ハイフン (-)
  - 下線 (_)
  - ピリオド (.)
  - アスタリスク (*)
  - 一重引用符 (')
  - 丸かっこ開き (()
  - 丸かっこ閉じ ())

有効なオブジェクトキー名の例を次に示します。

- `4my-organization`
- `my.great_photos-2014/jan/myvacation.jpg`
- `videos/2014/birthday/video1.wmv`

> **Important**
>
> オブジェクトキー名が単一のピリオド (.)、2 つのピリオド (..) で終了する場合は、Lightsail コンソールを使用してオブジェクトをダウンロードすることはできません。キー名が 1 つまたは 2 つのピリオドで終わるオブジェクトをダウンロードするには、Amazon S3 API、AWS CLI、および AWS SDKsを使用する必要があります。詳細については、「バケットオブジェクトをダウンロードする」を参照してください。

特殊な処理を必要とする可能性がある文字

キー名で以下の文字を使用すると、追加のコード処理が必要になる場合があり、16 進数として URL エンコードまたは参照することが必要になる可能性があります。これらの文字の一部は表示不可能な文字であり、ブラウザで処理されない場合があります。この場合も、特殊な処理が必要です。

- アンパサンド ("&")
- ドル記号 ("$")
- 16 進数の 00 ~ 1F (10 進数の 0 ~ 31) の範囲および 7F (10 進数の 127) の ASCII 文字
- アットマーク ("@")
- 等号 ("=")
- セミコロン (";")
- コロン (":")
- プラス記号 ("+")
- スペース – いくつかの用途 (特に複数のスペース) では、スペースの重要なシーケンスが失われる可能性があります。
- カンマ (",")
- 疑問符 ("?")

使用しない方がよい文字

すべてのアプリケーションで一貫性を維持するには相当な量の特殊な処理が必要になるため、キー名には以下の文字を使用しないでください。

- バックスラッシュ ("\")

- 左中括弧 ("{")
- 表示不可能な ASCII 文字 (10 進数の 128 ~ 255 の文字)
- カレット ("^")
- 右中括弧 ("}")
- パーセント記号 ("%")
- アクサングラーブ/バックティック ("`")
- 右角括弧 ("]")
- 引用符
- 大なり記号 (">")
- 左角括弧 ("[")
- チルダ ("~")
- 小なり記号 ("<")
- シャープ記号 ("#")
- 縦棒/パイプ ("|")

XML 関連のオブジェクトキーの制約

行末処理に関する XML 標準で規定されているとおり、すべての XML テキストは正規化され、1 つのキャリッジリターン (ASCII コード 13) と改行の直後のキャリッジリターン (ASCII コード 10) が単一の改行文字に置き換えられます。XML リクエストでオブジェクトキーを正しく解析するには、キャリッジリターンやその他の特殊文字を XML タグ内に挿入するときに、同等の XML エンティティコードに置き換える必要があります。以下では、当該特殊文字とそれに相当するエンティティコードのリストを示しています。

- `'` としての '
- `"` としての ”
- `&` としての &
- `<` としての <
- `>` としての <
- `
` または `
` としての `\r`
- `
` または `
` としての `\n`

次の例では、キャリッジリターンの代わりに XML エンティティコードを使用する方法を示しています。この DeleteObjects リクエストにより、 キーパラメータ /some/prefix/objectwith\rcarriagereturn (\r はキャリッジリターン) を持つオブジェクトが削除されます。

```
<Delete xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
      <Object>
        <Key>/some/prefix/objectwith&#13;carriagereturn</Key>
      </Object>
    </Delete>
```

Lightsail のオブジェクトストレージバケットを保護する

Amazon Lightsail オブジェクトストレージには、独自のセキュリティポリシーを開発および実装する際に考慮する必要のあるいくつかのセキュリティ機能が用意されています。以下のベストプラクティスは一般的なガイドラインであり、完全なセキュリティソリューションを説明するものではありません。これらのベストプラクティスはお客様の環境に適切ではないか、十分ではない場合があるため、これらは指示ではなく、有用な考慮事項と見なしてください。

目次

- 予防的セキュリティのベストプラクティス
  - 最小特権アクセスの実装
  - Lightsail バケットがパブリックにアクセス可能ではないことを確認する
  - Amazon S3 でパブリックアクセスのブロックを有効にする
  - バケットにインスタンスを添付して、プログラムによる完全なアクセスを付与する
  - バケットのアクセスキーをローテーションする
  - クロスアカウントアクセスを使用して、他の AWS アカウントにバケット内のオブジェクトへのアクセスを許可する
  - データの暗号化
  - バージョニングの有効化
- モニタリングと監査のベストプラクティス
  - アクセスログ記録を有効にし、セキュリティとアクセス監査を定期的に実行する
  - Lightsail バケットの特定、タグ付け、および監査
  - モニタリングツールを使用して AWS モニタリングを実装する
  - を使用する AWS CloudTrail

- AWS セキュリティアドバイザリのモニタリング

予防的セキュリティのベストプラクティス

以下のベストプラクティスは、Lightsail バケットでセキュリティ問題を防ぐのに役立ちます。

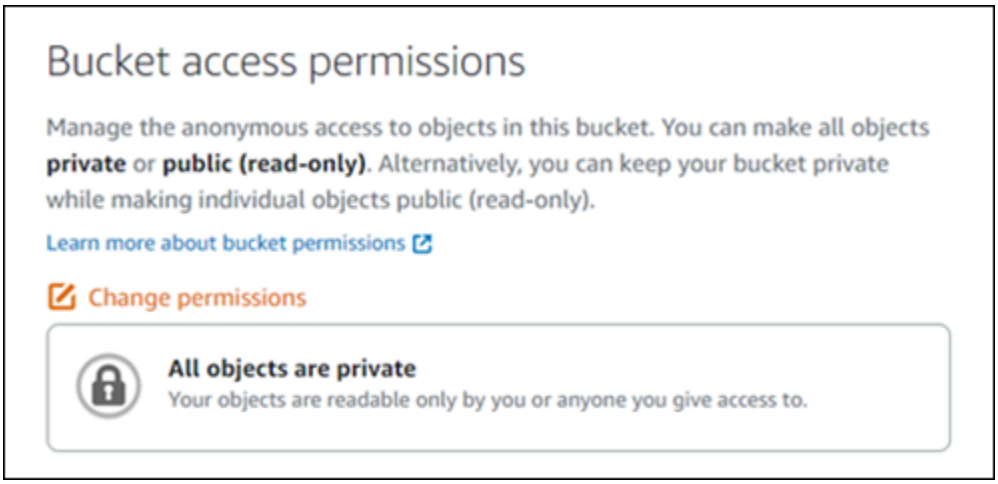
最小特権アクセスの実装

アクセス許可を付与する場合、どのユーザーにどの Lightsail リソースに対するアクセス許可を付与するかは、お客様が決定します。ユーザーは、それらのリソースで許可する特定のアクションを有効にします。そのため、タスクの実行に必要なアクセス許可のみを付与する必要があります。最小特権アクセスの実装は、セキュリティリスクと、エラーや悪意によってもたらされる可能性のある影響の低減における基本です。

バケットを管理するための IAM ポリシーの作成の詳細については、「バケットを管理する IAM ポリシー」を参照してください。Lightsail バケットでサポートされている Amazon S3 アクションの詳細については、Amazon Lightsail API リファレンスの「オブジェクトストレージのアクション」を参照してください。

Lightsail バケットがパブリックにアクセス可能ではないことを確認する

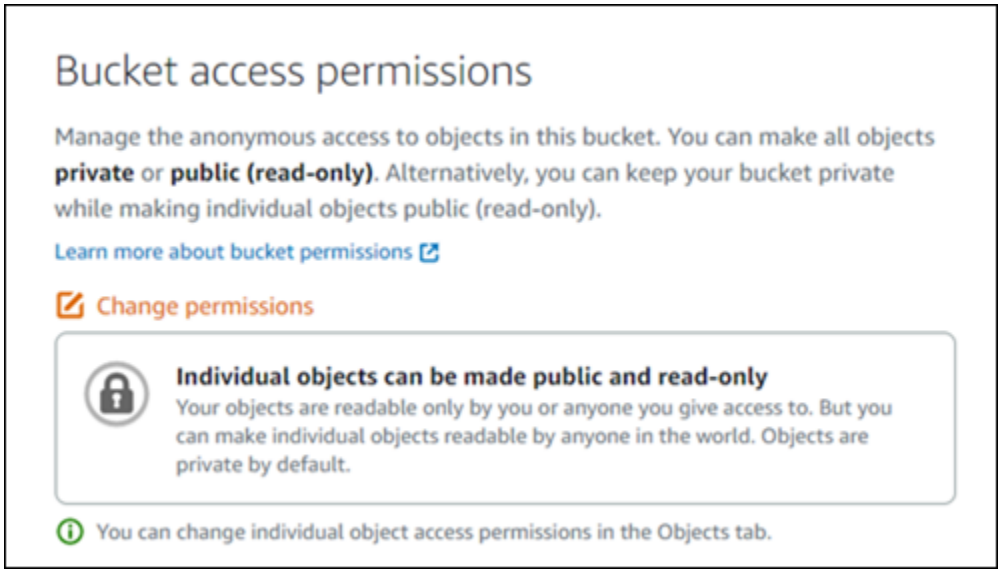
デフォルトでは、 バケットとオブジェクトはプライベートです。バケットのアクセス許可セットをすべてのオブジェクトはプライベートに設定して、バケットをプライベートに保ちます。大部分のユースケースでは、バケットや個々のオブジェクトをパブリックにする必要はありません。詳細については「バケット内の個々のオブジェクトに対するアクセス許可の設定」を参照してください。



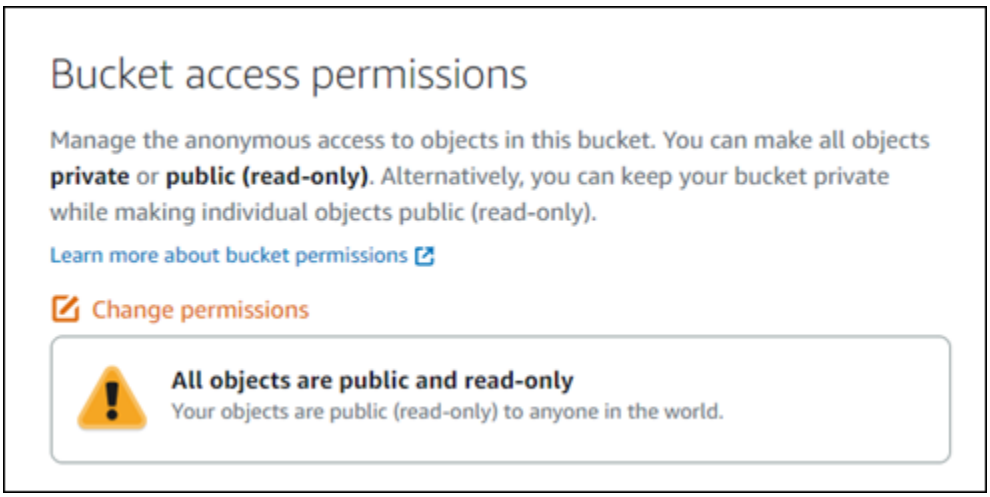
ただし、バケットを使用してウェブサイトやアプリケーションのメディアをホストしている場合は、特定のシナリオでは、バケットまたは個々のオブジェクトをパブリックにする必要があります。次の

いずれかのオプションを設定して、バケットまたは個々のオブジェクトをパブリックにすることができます。

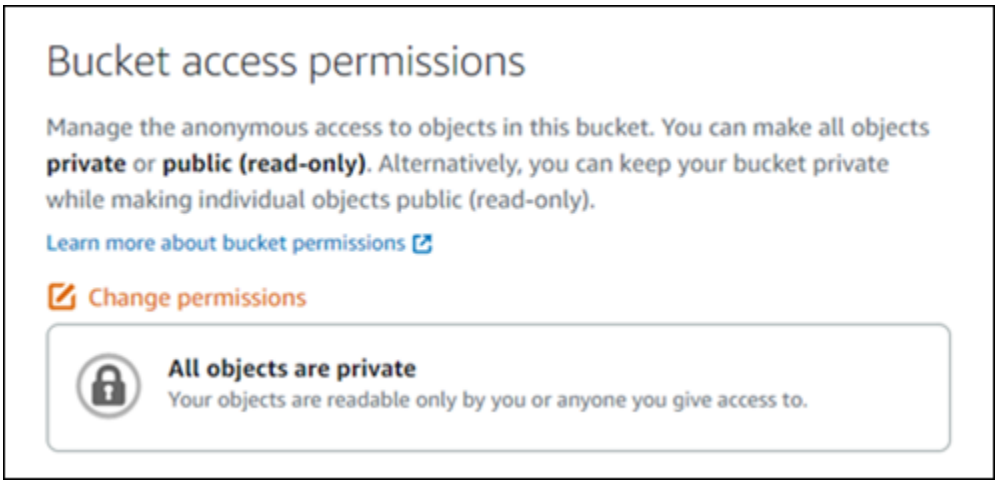
- バケット内のオブジェクトの一部のみをインターネット上の誰にでもパブリック (読み取り専用) する必要がある場合は、バケットのアクセス許可を個々のオブジェクトをパブリックにして読み取り専用にするに変更し、パブリックにする必要があるオブジェクトのみをパブリック (読み取り専用)に変更します。このオプションはバケットをプライベートにしますが、個々のオブジェクトをパブリックにするオプションも提供します。パブリックにアクセスしたくない機密情報または秘密情報が含まれている場合は、個々のオブジェクトを公開しないでください。個々のオブジェクトをパブリックにする場合は、個々のオブジェクトのパブリックアクセシビリティを定期的に検証する必要があります。



- バケット内のすべてのオブジェクトをインターネット上の誰にでもパブリック (読み取り専用) する必要がある場合は、バケットのアクセス許可をすべてのオブジェクトはパブリックで読み取り専用に変更します。バケット内のいずれかのオブジェクトに機密情報または秘密情報が含まれている場合は、このオプションを使用しないでください。



- 以前にバケットをパブリックに変更した場合、または個々のオブジェクトをパブリックに変更した場合は、バケットのアクセス許可をすべてのオブジェクトはプライベートに変更することで、バケットとそのすべてのオブジェクトをプライベートにすばやく変更できます。

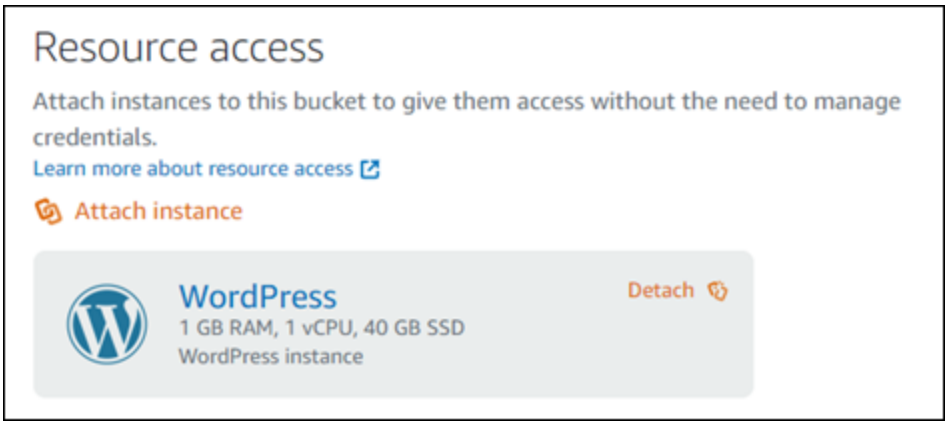


Amazon S3 でパブリックアクセスのブロックを有効にする

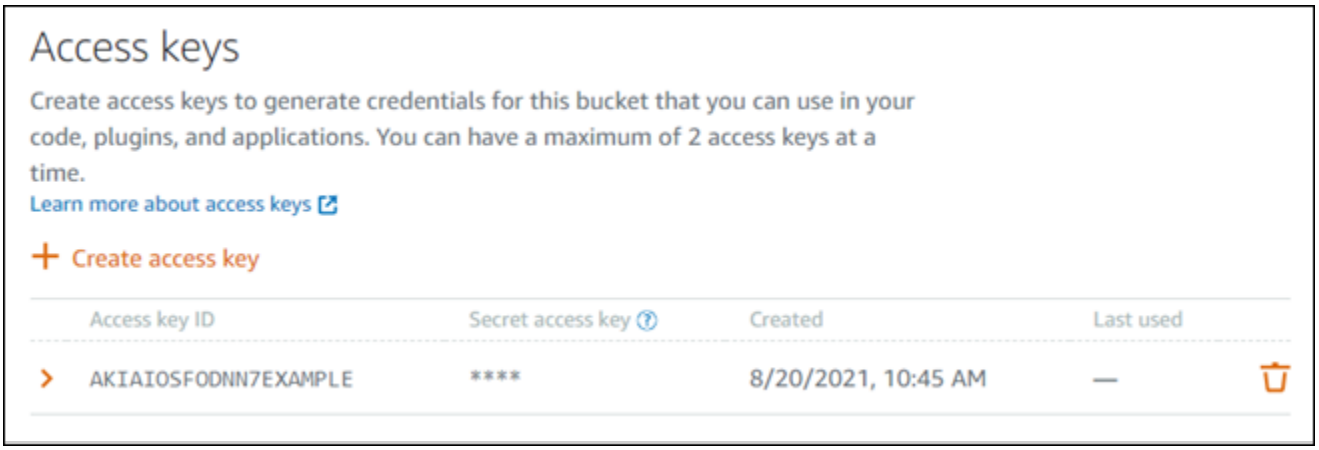
Lightsail オブジェクトストレージリソースは、パブリックアクセスを許可または拒否するときに、Lightsail バケットのアクセス許可と Amazon S3 アカウントレベルのブロックパブリックアクセス設定の両方を考慮するようになります。Amazon S3 アカウントレベルのブロックパブリックアクセスにより、アカウント管理者およびバケット所有者は、Amazon S3 および Lightsail バケットへのパブリックアクセスを一元的に制限できます。ブロックパブリックアクセスは、リソースがどのように作成され設定されているか、また設定されている可能性のある個々のバケットとオブジェクト権限にかかわらず、すべての Amazon S3 および Lightsail のバケットをプライベートにすることができます。詳細については、「バケットに対するブロックパブリックアクセス」を参照してください。

バケットにインスタンスを添付して、プログラムによる完全なアクセスを付与する

Lightsail オブジェクトストレージバケットにインスタンスを添付するのは、バケットへのアクセスを提供する最も安全な方法です。リソースアクセス機能は、インスタンスをバケットにアタッチする方法であり、インスタンスにバケットへの完全なプログラムによるアクセスを付与します。この方法では、バケット認証情報をインスタンスまたはアプリケーションに直接保存する必要はなく、定期的に認証情報をローテーションする必要もありません。例えば、一部の WordPress プラグインは、インスタンスがアクセスできるバケットにアクセスできます。詳細については、「バケットのリソースアクセスを設定する」および「チュートリアル: バケットを WordPress インスタンスに接続する」を参照してください。



ただし、アプリケーションが Lightsail インスタンスにない場合は、バケットアクセスキーを作成して設定することができます。バケットアクセスキーは、自動的にローテーションされない長期的な認証情報です。詳細については、「Lightsail オブジェクトストレージバケットのアクセスキーを作成する」を参照してください。



バケットのアクセスキーをローテーションする

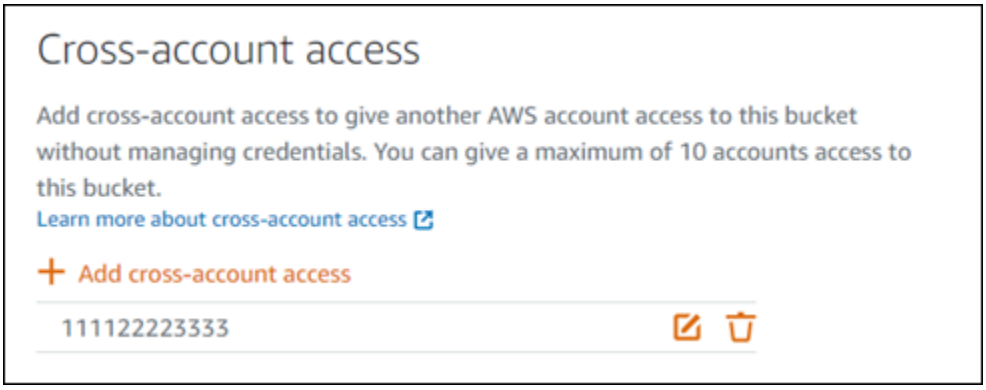
バケットごとに最大 2 つのアクセスキーを持つことができます。2 つの異なるアクセスキーを同時に持つことはできますが、キーローテーション時を除き、バケットに対し作成するアクセスキーは一度に 1 つのみにすることをお勧めします。このアプローチにより、新しいバケットアクセスキーが既に使用中である可能性を気にすることなく、いつでも作成できるようになります。例えば、既存のシークレットアクセスキーがコピー、紛失、侵害されたためにローテーションする必要がある場合、ローテーション用に 2 つ目のアクセスキーを作成すると便利です。

バケットでアクセスキーを使用する場合は、キーを定期的にローテーションし、既存のキーのインベントリを作成する必要があります。アクセスキーが最後に使用された日付と、そのキーが使用された AWS リージョン が、キーの使用方法に関する期待に沿っていることを確認します。アクセスキーが最後に使用された日付は、バケットの管理ページにある [アクセス許可] タブの [アクセスキー] セクションの Lightsail コンソールに表示されます。使用されていないアクセスキーを削除します。

アクセスキーをローテーションするには、新規のアクセスキーを作成し、ソフトウェアで設定してテストしてから、以前のアクセスキーを削除してください。アクセスキーを削除すると、永久に削除されるため、再度取得することはできません。新しいアクセスキーでのみ置き換えることができます。詳細については、「Lightsail オブジェクトストレージバケットのアクセスキーを作成する」および「Lightsail オブジェクトのストレージバケットのアクセスキーを削除します」を参照してください。

クロスアカウントアクセスを使用して、他の AWS アカウントにバケット内のオブジェクトへのアクセスを許可する

クロスアカウントアクセスを使用すると、バケットとそのオブジェクトを公開することなく、 AWS アカウントを持つ特定の個人がバケット内のオブジェクトにアクセスできるようになります。クロスアカウントアクセスを設定している場合は、リストされているアカウント ID が、バケット内のオブジェクトへのアクセスを許可する正しいアカウントであることを確認してください。詳細については、「バケットのクロスアカウントアクセスの設定」を参照してください。



データの暗号化

Lightsail は、Amazon マネージドキーを使用したサーバー側の暗号化と HTTPS (TLS) の強制による転送中のデータの暗号化を実行します。サーバー側の暗号化は、別のサービスに保存されているキーを使用してデータを暗号化することで、データへのリスクを軽減するのに役立ちます。さらに、送信中のデータの暗号化は、潜在的な攻撃者が中間者攻撃または同様の攻撃を使用してネットワークトラフィックを盗聴または操作することを防止するのに役立ちます。

バージョニングの有効化

バージョニングとは、同じバケット内でオブジェクトの複数のバリアントを保持する手段です。バージョニングを使用して、Lightsail バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元することができます。バージョニングを使用すれば、意図しないユーザーアクションからもアプリケーション障害からも、簡単に復旧できます。詳細については、「バケットのオブジェクトのバージョニングを有効化または一時停止する」を参照してください。

モニタリングと監査のベストプラクティス

以下のベストプラクティスは、Lightsail バケットの潜在的なセキュリティ上の弱点とインシデントを検出するのに役立ちます。

アクセスログ記録を有効にし、セキュリティとアクセス監査を定期的に実行する

アクセスのログ記録には、バケットに対するリクエストの詳細が記録されます。この情報には、リクエストタイプ (GET、PUT)、リクエストで指定したリソース、リクエストを処理した日時などが含まれます。バケットのアクセスログ記録を有効にし、セキュリティとアクセス監査を定期的に実行して、バケットにアクセスしているエンティティを特定します。デフォルトでは、Lightsail によってバケットへのアクセスのログは収集されません。アクセスログ記録を手動で有効にする必要があります。詳細については、「バケットのアクセスログ」と「バケットのアクセスログの記録を有効にする」を参照してください。

Lightsail バケットの特定、タグ付け、および監査

IT アセットの特定はガバナンスとセキュリティの重要な側面です。セキュリティ体制を評価し、潜在的な弱点に対処するには、すべての Lightsail バケットを可視化する必要があります。

タグ付けを使用してセキュリティまたは監査で注意を要するリソースを識別してから、それらのタグを、リソースを検索する必要があるときに使用します。詳細については、「タグ」を参照してください。

モニタリングツールを使用して AWS モニタリングを実装する

モニタリングは、Lightsail バケットおよびその他のリソースの信頼性、セキュリティ、可用性、パフォーマンスを維持する上で重要です。バケットサイズ (BucketSizeBytes) と Lightsail での Number of objects (NumberOfObjects) バケットメトリクスの通知アラームを作成してモニタリングすることができます。例えば、バケットのサイズが特定のサイズに増減したとき、またはバケット内のオブジェクト数が特定の数に増減したときに通知を受け取ることができます。詳細については、「バケットメトリクスアラームの作成」を参照してください。

を使用する AWS CloudTrail

AWS CloudTrail は、 のユーザー、ロール、または AWS のサービスによって実行されたアクションの記録を提供しますLightsail。CloudTrail で収集された情報を使用して、Lightsail に対するリクエスト、リクエスト元の IP アドレス、リクエストの実行者、リクエスト日時などの詳細を把握できます。例えば、データアクセスに影響するアクション (特に CreateBucketAccessKey、GetBucketAccessKeys、DeleteBucketAccessKey、SetResourceAcc および UpdateBucket) の CloudTrail エントリを特定できます。 AWS アカウントを設定すると、CloudTrail はデフォルトで有効になります。CloudTrail コンソールで最近のイベントを確認できます。Lightsail バケットのアクティビティとイベントの継続的なレコードを作成するには、CloudTrail コンソールで追跡作成できます。詳細については、https://docs.aws.amazon.com/awscloudtrail/latest/userguide/logging-data-events-with-cloudtrail.html ユーザーガイドのAWS CloudTrail 証跡へのデータイベントのログ記録 を参照してください。

AWS セキュリティアドバイザリのモニタリング

AWS アカウントに登録されているプライマリ E メールアドレスを積極的にモニタリングします。AWS は、この E メールアドレスを使用して、ユーザーに影響を与える可能性のある新たなセキュリティ上の問題について連絡します。

AWS 広範な影響を与える運用上の問題は、AWS Service Health Dashboard に投稿されます。運用上の問題は Personal Health Dashboard を介して個々のアカウントにも投稿されます。詳細については、AWS の正常性に関するドキュメントを参照してください。

Lightsail バケットとオブジェクトへのアクセスをコントロールする

デフォルトでは、すべての Amazon Lightsail オブジェクトストレージリソース (バケットとオブジェクト) はプライベートです。これは、バケット所有者、つまりバケットを作成した Lightsail アカウントのみがバケットとそのオブジェクトにアクセスできることを意味します。バケット所有者は、オプ

ションで他のユーザーにアクセス許可を付与できます。バケットとそのオブジェクトへのアクセスを許可するには、以下の方法があります。

- 読み取り専用アクセス — 以下のオプションは、バケットの URL (たとえば、`https://amzn-s3-demo-bucket.us-east-1.amazonaws.com/media/sailbot.jpg`) を介してバケットとそのオブジェクトへの読み取り専用アクセスを制御します。
  - バケットアクセス許可 — バケットのアクセス許可を使用して、インターネット上のすべてのオブジェクトへのアクセスを許可します。詳細については、このガイドで後述する「バケットのアクセス許可」を参照してください。
  - 個々のオブジェクトのアクセス許可 — 個々のオブジェクトアクセス許可を使用して、インターネット上のすべてのユーザーに、バケット内の個々のオブジェクトへのアクセスを許可します。詳細については、このガイドで後述する「個々のオブジェクトへのアクセス許可」を参照してください。
  - クロスアカウントアクセス – クロスアカウントアクセスを使用して、他の AWS アカウントのバケット内のすべてのオブジェクトへのアクセスを許可します。詳細については、このガイドで後述する「クロスアカウント アクセス」を参照してください。
- 読み取りおよび書き込みアクセス — バケットとそのオブジェクトへの完全な読み取りおよび書き込みアクセスを制御します。(AWS CLI)、 AWS Command Line Interface AWS APIs、 AWS SDKs でこれらのオプションを使用します。
  - アクセスキー — アクセスキーを使用して、アプリケーションやプラグインへのアクセスを許可します。詳細については、このガイドで後述する「アクセスキー」を参照してください。
  - リソースアクセス — リソースアクセスを使用して、Lightsail インスタンスへのアクセスを許可します。詳細については、このガイドで後述する。[リソースアクセス]]を参照してください。
- Amazon Simple Storage Service ブロックパブリックアクセス — Amazon Simple Storage Service (Amazon S3) アカウントレベルのブロックパブリックアクセス機能を使用して、Amazon S3 および Lightsail でバケットへのパブリックアクセスを一元的に制限します。ブロックパブリックアクセスは、設定された個々のバケットとオブジェクトの許可にかかわらず、すべての Amazon S3 および Lightsail のバケットをプライベートにすることができます。詳細については、このガイドで後述する「Amazon S3 パブリックアクセスブロック」を参照してください。

バケットについての詳細は、「オブジェクトストレージ」を参照してください。セキュリティのベストプラクティスの詳細については、「オブジェクトストレージのセキュリティのベストプラクティス」を参照してください。

バケットのアクセス許可

バケットのアクセス許可を使用して、バケット内のオブジェクトへの公開 (認証されていない) 読み取り専用アクセスを制御します。バケットのアクセス許可を設定する場合、以下のいずれかのオプションを選択します。

- すべてのオブジェクトがプライベート — バケット内のすべてのオブジェクトは、ご自身またはアクセスを許可したユーザーのみが読み取ることができます。このオプションでは、個々のオブジェクトを公開 (読み取り専用) にすることはできません。
- 個々のオブジェクトが公開可能 (読み取り専用) — バケット内のオブジェクトは、個々のオブジェクトを公開 (読み取り専用) として指定しない限り、自身またはアクセスを許可したユーザーのみが読み取ることができます。このオプションを使用すると、個々のオブジェクトを公開 (読み取り専用) にできます。詳細については、このガイドで後述する「個々のオブジェクトへのアクセス許可」を参照してください。
- すべてのオブジェクトが公開 (読み取り専用) — バケット内のすべてのオブジェクトは、インターネット上の誰でも読み取り可能です。このオプションを選択すると、バケット内のすべてのオブジェクトは、バケットのURL (たとえば、`https://amzn-s3-demo-bucket.us-east-1.amazonaws.com/media/sailbot.jpg`) を介してインターネット上の誰でも読み取り可能になります。

バケットアクセス許可の設定に関する詳細については、「バケットアクセス許可の設定」を参照してください。

個々のオブジェクトのアクセス許可

個々のオブジェクトアクセス許可を使用して、認証なしで公開されたバケット内の個々のオブジェクトの読み取り専用アクセスを制御します。個々のオブジェクトのアクセス権は、バケットのアクセス許可が、個々のオブジェクトの公開 (読み取り専用) を許可している場合にのみ設定できます。個々のオブジェクトのアクセス許可を設定する場合は、以下のいずれかのオプションを選択します。

- プライベート — このオブジェクトはご自身とアクセスを許可したユーザーのみが読み取ることができます。
- 公開 (読み取り専用) — このオブジェクトは、インターネット上の誰でも読み取り可能です。個々のオブジェクトは、インターネット上の誰でもバケットの URL を通じて読み取れるようになります (たとえば、`https://amzn-s3-demo-bucket.us-east-1.amazonaws.com/media/sailbot.jpg`)。

個々のオブジェクトのアクセス許可の設定に関する詳細については、「バケット内の個々のオブジェクトに対するアクセス許可の設定」を参照してください。

クロスアカウントアクセス

クロスアカウントアクセスを使用して、バケット内のすべてのオブジェクトへの認証された読み取り専用アクセスを他の AWS アカウントとそのユーザーに付与します。クロスアカウントアクセスは、オブジェクトを別の AWS アカウントと共有する場合に最適です。別の AWS アカウントにクロスアカウントアクセスを付与すると、そのアカウントのユーザーは、バケットの URL (例えば、`https://amzn-s3-demo-bucket.us-east-1.amazonaws.com/media/sailbot.jpg`) を通じてバケット内のオブジェクトに読み取り専用のアクセスが可能になります。最大 10 個の AWS アカウントへのアクセスを許可できます。

クロスアカウントアクセスの設定に関する詳細については、「バケットのクロスアカウント アクセスの設定」を参照してください。

アクセスキー

アクセスキーを使用して、バケットとそのオブジェクトへの完全な読み取りおよび書き込みアクセスを付与する認証情報セットを作成します。アクセスキーは、アクセスキー ID とシークレットアクセスキーがセットです。バケットごとに最大 2 つのアクセスキーを持つことができます。API と AWS SDKs を使用してバケットとそのオブジェクトにアクセスできるように AWS APIs 、アプリケーションでアクセスキーを設定できます。CLI AWS でアクセスキーを設定することもできます。

アクセスキーの作成に関する詳細については、「バケットのアクセスキーの作成」を参照してください。

リソースアクセス

リソースアクセスを使用し、Lightsail インスタンスのバケットとそのオブジェクトへの完全な読み取りおよび書き込みアクセスを許可します。リソースアクセスでは、アクセスキーなどの認証情報を管理する必要はありません。インスタンスへのアクセスを許可するには、インスタンスを同じ AWS リージョンのバケットにアタッチします。アクセスを拒否するには、インスタンスをバケットからデタッチします。リソースアクセスは、インスタンス上のアプリケーションで、バケット上のファイルをプログラムでアップロードしたりアクセスしたりするように設定する場合に最適です。このようなユースケースの 1 つに、メディアファイルをバケットに保存するように WordPress インスタンスを設定するものがあります。詳細については、「チュートリアル: バケットを WordPress インスタンスに接続する」および「チュートリアル: バケットをコンテンツ配信ネットワークディストリビューションと使用する」を参照してください。

リソースアクセスの設定に関する詳細については、「バケットのリソースアクセスの設定」を参照してください。

Amazon S3 パブリックアクセスブロック

Amazon S3 パブリックアクセスブロック機能を使用して、Amazon S3 および Lightsail でバケットへのパブリックアクセスを一元的に制限します。ブロックパブリックアクセスは、設定された個々のバケットとオブジェクトの許可にかかわらず、すべての Amazon S3 および Lightsail のバケットをプライベートにすることができます。Amazon S3 コンソール、 AWS CLI、 AWS SDKs、および REST API を使用して、Lightsailオブジェクトストレージサービスのバケットを含め、アカウント内のすべてのバケットのパブリックアクセスブロックを設定できます。詳細については、「バケットに対するブロックパブリックアクセス」を参照してください。

Lightsail オブジェクトストレージバケットのファイルをアップロードする

Amazon Lightsail オブジェクトストレージサービスのバケットにファイルをアップロードすると、そのファイルはオブジェクトとして保存されます。オブジェクトは、オブジェクトを記述するファイルデータとメタデータから構成されます。バケットには、オブジェクトをいくつでも保存できます。

ファイルタイプ (イメージ、バックアップ、データ、ムービーなど) を問わず、各種のファイルをバケットにアップロードできます。Lightsail コンソールを使用して、アップロードできるファイルの最大サイズは 2 GB です。より大きなファイルをアップロードするには、 Lightsail API、 AWS Command Line Interface (AWS CLI)、または AWS SDKsを使用します。

Lightsail では、アップロードするファイルのサイズに応じて以下のオプションが用意されています。

- Lightsail コンソールを使用して最大 2 GB のサイズのオブジェクトをアップロード — Lightsail コンソールを使用すると、最大 2 GB のサイズの単一のオブジェクトをアップロードできます。詳細については、このガイドで後述する「Lightsail コンソールを使用したバケットへのファイルのアップロード」を参照してください。
- AWS SDKs、REST API、または を使用した 1 回のオペレーションで最大 5 GB のサイズのオブジェクトをアップロードする AWS CLI — 1 回の PUT オペレーションで、最大 5 GB のサイズのオブジェクトを 1 つアップロードできます。詳細については、このガイドで後述する「AWS CLI を使用したバケットへのファイルのアップロード」を参照してください。
- AWS SDKs、REST API、または AWS CLI — マルチパートアップロード API を使用して、5 MB ～50 TB のサイズの単一の大きなオブジェクトをアップロードできます。マルチパートアップロード API は大容量オブジェクトのアップロードを効率よく行えるように設計されています。1 つのオ

ブジェクトをいくつかに分けてアップロードできます。オブジェクトのパートは、単独で、任意の順序で、または並行してアップロードできます。詳細については、「マルチパートアップロードを使用してバケットにファイルをアップロードする」を参照してください。

バケットについての詳細は、「オブジェクトストレージ」を参照してください。

オブジェクトキーの名前とバージョニング

Lightsail コンソールを使用してファイルをアップロードする場合、ファイル名がオブジェクトキー名として使用されます。オブジェクトキー (またはキー名) によって、バケットに保存されているオブジェクトを一意に識別します。ファイルがアップロードされたフォルダが存在する場合、キー名のプレフィックスとして使用されます。たとえば、`sailbot.jpg` という名前のファイルを `images` という名前のバケット内のフォルダーにアップロードすると、完全なオブジェクトキー名とプレフィックスは `images/sailbot.jpg` になります。ただし、オブジェクトはコンソールの `sailbot.jpg` フォルダ内で `images` として表示されます。オブジェクトキー名の詳細については、「オブジェクトストレージバケットのキー名」を参照してください。

Lightsail コンソールを使用してディレクトリをアップロードすると、ディレクトリ内のすべてのファイルとサブフォルダがバケットにアップロードされます。Lightsail は次に、アップロードされた各ファイル名とフォルダ名を組み合わせたオブジェクトキー名を割り当てます。たとえば、`sample1.jpg` と `sample2.jpg` の2つのファイルを含む `images` という名前のフォルダをアップロードすると、Lightsail はファイルをアップロードしてから、対応するキー名 、`images/sample1.jpg` と `images/sample2.jpg` を割り当てます。コンソールには、オブジェクトが `sample1.jpg` および `sample2.jpg` の `images` フォルダーとして表示されます。

すでに存在するキー名のファイルをアップロードし、バケットのバージョニングが有効になっていない場合、新しくアップロードされたオブジェクトが前のオブジェクトに置き換えられます。ただし、バケットでバージョニングが有効になっている場合は、Lightsail 既存のオブジェクトを置き換えるのではなく、オブジェクトの新しいバージョンを作成します。詳細については、「バケットのオブジェクトのバージョニングを有効化または一時停止する」を参照してください。

Lightsail コンソールを使用してバケットにファイルをアップロードするには

Lightsail コンソールを使用してファイルとディレクトリをアップロードするには、次の手順を実行します。

1. Lightsail コンソール にサインインします。

2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. ファイルとフォルダをアップロードするバケットの名前を選択します。
4. [Objects] (オブジェクト) タブで、次のいずれかのアクションを実行します。

   - ファイルとフォルダを [Objects] ページにドラッグアンドドロップします。
   - [Upload] (アップロード) を選択し、[File] (ファイル) を選択して個々のファイルをアップロードするか、[Directory] (ディレクトリ) を選択して、フォルダとそのすべてのコンテンツをアップロードします。

   > **Note**
   >
   > [Create new folder] (新しいフォルダの作成) を選択してフォルダを作成することもできます。その後、新しいフォルダを参照して、そのフォルダにファイルをアップロードできます。

   アップロードが完了すると、正常にアップロードしましたというメッセージが表示されます。

を使用してバケットにファイルをアップロードする AWS CLI

AWS Command Line Interface (AWS CLI) を使用してファイルやフォルダーをバケットにアップロードを完了するには、以下の手順を実行します。これは、`put-object` コマンドを使用して実行できます。詳細については、「AWS CLI コマンドリファレンス」の「put-object」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsail と Amazon S3 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 以下のコマンドを入力して、ファイルをバケットにアップロードします。

```
aws s3api put-object --bucket BucketName --key ObjectKey --body LocalDirectory --
acl bucket-owner-full-control
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *BucketName* は、ファイルをアップロードするバケットの名前に置き換えます。
- *ObjectKey* は、バケット内のオブジェクトの完全なオブジェクトキーに置き換えます。
- *LocalDirectory* は、アップロードするファイルのコンピュータ上のローカルディレクトリフォルダパスに置き換えます。

例:

- Linux または Unix コンピュータの場合：

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key images/sailbot.jpg --
body home/user/Pictures/sailbot.jpg --acl bucket-owner-full-control
```

- Windows コンピュータの場合:

```
aws s3api put-object --bucket amzn-s3-demo-bucket --key images/sailbot.jpg --
body "C:\Users\user\Pictures\sailbot.jpg" --acl bucket-owner-full-control
```

以下の例のような結果が表示されるはずです。

```
C:\>aws s3api put-object --bucket DOC-EXAMPLE-BUCKET --key images/sailbot.jpg --body "C:\Users\user\Pictures\sailbot.jpg"
{
    "ETag": "\"694d34edexampled92d64f342aa234c3\""
}
```

IPv6 専用リクエスト用に AWS CLI を設定する

Amazon S3 は IPv6 を介したバケットアクセスをサポートしています。IPv6 での Amazon S3 API コールを使用したリクエストは、デュアルスタックのエンドポイントを使用して行います。このセクションでは、IPv6 を介してデュアルスタックのエンドポイントへリクエストする方法の例を示します。詳細については、「Amazon S3 API ユーザーガイド」の「Using Amazon S3 dual-stack endpoints」を参照してください。の設定手順については AWS CLI、「 と連携 AWS Command Line Interface するように Amazon Lightsailを設定する」を参照してください。

> **Important**
>
> バケットにアクセスするクライアントやネットワークは、IPv6 の使用を有効にする必要があります。詳細については、「IPv6 の到達可能性」を参照してください。

IPv6 専用インスタンスから S3 リクエストを行うには、2 つの方法があります。すべての Amazon S3 リクエストを、指定された のデュアルスタックエンドポイントに転送 AWS CLI するように を設定できます AWS リージョン。または、指定した AWS CLI コマンド (すべてのコマンドではない) にのみデュアルスタックエンドポイントを使用する場合は、すべてのコマンドに S3 デュアルスタックエンドポイントを追加できます。

を設定する AWS CLI

AWS Config ファイルのプロファイルtrueで設定値を use_dualstack_endpointに設定して、Amazon S3 および s3api AWS CLI コマンドによって行われたすべての Amazon S3 リクエストを、指定されたリージョンのデュアルスタックエンドポイントに送信します。リージョンは、AWS CLI 設定ファイル、または --region オプションを使用してコマンドで指定します。

次のコマンドを入力して、 AWS CLIを設定します。

```
aws configure set default.s3.use_dualstack_endpoint true
```

```
aws configure set default.s3.addressing_style virtual
```

特定のコマンドにデュアルスタックエンドポイントを追加する

コマンドでデュアルスタックのエンドポイントを使用する。その場合は、すべての s3 または s3api コマンドの --endpoint-url パラメータを https://s3.dualstack.*aws-region*.amazonaws.com または http://s3.dualstack.*aws-region*.amazonaws.com に設定します。以下の例では、*bucketname* と *aws-region* をバケット名と AWS リージョンに置き換えます。

```
aws s3api list-objects --bucket bucketname --endpoint-url https://s3.dualstack.aws-region.amazonaws.com
```

Lightsail でのバケットとオブジェクトの管理

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。
3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

   バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

   - Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
   - Amazon Lightsail でのバケットのアクセス許可の設定
   - Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
   - Amazon Lightsail でのバケットのアクセスキーの作成
   - Amazon Lightsail でのバケットのリソースアクセスの設定
   - Amazon Lightsail でのバケットのクロスアカウントアクセスの設定
5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。

   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスの口グ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスの口グ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定

6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail のバケットにファイルをアップロードする
   - マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
   - Amazon Lightsail のバケット内のオブジェクトの表示
   - Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
   - Amazon Lightsail のバケットからのオブジェクトのダウンロード
   - Amazon Lightsail のバケット内のオブジェクトのフィルタリング
   - Amazon Lightsail のバケット内のオブジェクトのタグ付け
   - Amazon Lightsail のバケット内のオブジェクトの削除
9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。
11. バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。
12. バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。
13. ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。
14. バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。
   - チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
   - チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用

15.使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

Lightsail のクロスオリジンリソース共有 (CORS)

Cross-Origin Resource Sharing (CORS) は、特定のドメインにロードされたクライアントウェブアプリケーションが異なるドメイン内のリソースと通信する方法を定義します。CORS のサポートにより、Lightsail オブジェクトストレージを使用して機能豊富なクライアント側ウェブアプリケーションを構築し、バケットリソースへのクロスオリジンアクセスを選択的に許可できるようになります。CORS の詳細については、「CORS とは？」を参照してください。

このセクションでは、AWS Command Line Interface (AWS CLI) を使用して Lightsail バケットの CORS を設定する方法について説明します。クロスオリジンリクエストを許可するようにバケットを設定するには、JSON ドキュメントを使用して CORS 設定をバケットに追加します。この JSON ドキュメントでは、バケットへのアクセスを許可するオリジン、各オリジンに対してサポートされるオペレーション (HTTP メソッド)、その他のオペレーション固有の情報を識別するルールを定義します。

トピック

- CORS のユースケース
- Lightsail が CORS 設定を評価する方法
- AWS CLI を使用して CORS の設定
- CORS のトラブルシューティング

CORS のユースケース

次のシナリオ例では、Lightsail バケットで CORS を設定する方法の詳細について説明します。

シナリオ: ウェブフォントホスト

Lightsail バケットからウェブフォントをホストするとします。ブラウザでウェブフォントを読み込むため、CORS チェック (プリフライトチェックとも呼ばれる) が必要です。ウェブフォントをホストしているバケットを設定し、いずれかのオリジンがこれらのリクエストを実行できるように許可します。

Lightsail が CORS 設定を評価する方法

Lightsail オブジェクトストレージは、ブラウザからプリフライトリクエストを受け取ると、バケットの CORS 設定を評価し、受信ブラウザリクエストに一致する最初の CORS ルールを使用して、クロスオリジンリクエストを有効にします。ルールが一致するには、次の条件を満たしている必要があります。

- リクエストの Origin ヘッダーは、AllowedOrigins 要素のオリジンと一致する必要があります。
- Access-Control-Request-Method ヘッダーで指定された HTTP メソッドは、AllowedMethods 要素のメソッドと一致する必要があります。
- Access-Control-Request-Headers ヘッダーにリストされているヘッダーは、AllowedHeaders 要素のヘッダーと一致する必要があります。

> **Note**
>
> バケットで CORS を有効にすると、バケットのアクセス許可は引き続き適用されます。CORS 設定は、ブラウザがクロスオリジンリクエストの続行を許可するかどうかのみを決定します。詳細については、「Lightsail バケットとオブジェクトへのアクセスをコントロールする」を参照してください。

CORS 設定のエレメント

CORS 設定は、CORS ルールの配列を含む JSON ドキュメントです。各ルールは、バケットへのアクセスを許可するオリジン、許可される HTTP メソッド、およびその他の設定オプションを定義します。

CORS ルールには、次の要素を含めることができます。

allowedOrigins

バケットへのアクセスが許可されているオリジンを指定します。ワイルドカード (*) を使用してすべてのオリジンを許可するか、https://example.com などの特定のドメインを指定できます。

allowedMethods

指定されたオリジンに許可される HTTP メソッドを指定します。有効な値には、GET、PUT、POST、DELETE、HEAD などがあります。

allowedHeaders

Access-Control-Request-Headers ヘッダーによって、プリフライト OPTIONS リクエストで許可されるヘッダーを指定します。

exposeHeaders

レスポンス内のどのヘッダーにクライアントアプリケーションでアクセスできるかを指定します。

id

CORS ルール用の一意の識別子。

maxAgeSeconds

プリフライトリクエストに対して、ブラウザがレスポンスをキャッシュできる時間を秒単位で指定します。

これらのパラメータの詳細については、「BucketCorsRule」(Amazon Lightsail API リファレンス) を参照してください。

AWS CLI を使用して CORS の設定

`--cors` パラメータを持つ AWS CLI を使用して、Lightsail バケットの CORS を設定できます。このパラメータは CORS 設定を含む JSON ファイルを受け入れます。CORS 設定の要素の詳細については、「CORS 設定のエレメント」を参照してください。

トピック

- CORS 構成の適用
- CORS 設定例
- CORS 設定の削除

CORS 構成の適用

以下の手順は、JSON ファイルを指定して CORS 設定をバケットに適用する方法を示しています。その他の設定例については、「CORS 設定例」を参照してください。

AWS CLI を使用してバケットに CORS を設定する方法

1. CORS 設定を含む JSON ファイルを作成します。例えば、次の内容で `cors-config.json` という名前のファイルを作成します。

```
{
  "CORSRules": [
```

```
    {
      "AllowedOrigins": ["https://example.com"],
      "AllowedMethods": ["GET", "PUT", "POST"],
      "AllowedHeaders": ["*"],
      "MaxAgeSeconds": 3000
    }
  ]
}
```

2. AWS CLI を使用して CORS 設定をバケットに適用します。

```
aws lightsail update-bucket --bucket-name amzn-s3-demo-bucket --cors file://cors-config.json
```

3. CORS 設定が正常に適用されたことを確認してください。

```
aws lightsail get-buckets --bucket-name amzn-s3-demo-bucket --include-cors
```

> **Note**
>
> *amzn-s3-demo-bucket* を Lightsail のバケット名に置き換えます。

CORS 設定例

以下の例は、さまざまなユースケースに対する一般的な CORS 設定を示しています。

例 1: すべてのオリジンとメソッドを許可する

この設定により、任意の HTTP メソッドを使用してすべてのオリジンにバケットへのアクセスを許可します。

```
{
    "CORSRules": [
      {
        "AllowedOrigins": ["*"],
        "AllowedMethods": ["GET", "PUT", "POST", "DELETE", "HEAD"],
        "AllowedHeaders": ["*"],
        "MaxAgeSeconds": 3000
      }
    ]
```

```
  }
```

例 2: 特定のドメインに制限する

この設定では、https://mywebsite.com からのリクエストのみを許可します。

```
{
    "CORSRules": [
      {
        "AllowedOrigins": ["https://mywebsite.com"],
        "AllowedMethods": ["GET", "PUT"],
        "AllowedHeaders": ["Authorization", "Content-Type"],
        "ExposeHeaders": ["ETag"],
        "MaxAgeSeconds": 3600
      }
    ]
  }
```

例 3: さまざまなオリジンに複数のルール

この設定では、オリジンごとに異なるルールを定義します。

```
{
    "CORSRules": [
      {
        "AllowedOrigins": ["https://mywebsite.com"],
        "AllowedMethods": ["GET", "PUT", "POST"],
        "AllowedHeaders": ["*"],
        "MaxAgeSeconds": 3600
      },
      {
        "AllowedOrigins": ["https://cdn.mywebsite.com"],
        "AllowedMethods": ["GET"],
        "AllowedHeaders": ["Authorization"],
        "MaxAgeSeconds": 86400
      }
    ]
  }
```

CORS 設定の削除

バケットから CORS 設定を削除するには、次の AWS CLI コマンドを実行します。

```
aws lightsail update-bucket --bucket-name amzn-s3-demo-bucket --cors '{"rules":[]}'
```

> **Note**
>
> *amzn-s3-demo-bucket* を Lightsail のバケット名に置き換えます。

CORS 設定を削除すると、バケットへのクロスオリジンリクエストはブラウザによってブロックされます。

CORS のトラブルシューティング

CORS に問題がある場合は、以下を確認してください。

- CORS 設定の検証 – CORS 設定により JSON が適切にフォーマットされ、ユースケースに必要なルールが含まれていることを確認します。
- オリジン一致の確認 – リクエストのオリジンは、`AllowedOrigins` リストのエントリと完全に一致する必要があります。プロトコル (http/https)、サブドメイン、およびポートは完全に一致する必要があります。
- HTTP メソッドの検証 – 使用している HTTP メソッドが、一致するルールの `AllowedMethods` にリストされていることを確認します。
- ブラウザ開発者ツールの確認 – ブラウザの開発者ツールを使用して、プリフライト OPTIONS リクエストとレスポンスを検査し、CORS 関連のエラーを特定します。
- バケットのアクセス許可の検証 – バケットに CORS に加えて適切なアクセス許可が設定されていることを確認します。CORS は、バケットレベルのアクセス許可ではなく、ブラウザベースのクロスオリジンアクセスのみを制御します。

トラブルシューティング中にバケットから CORS 設定を削除する必要がある場合は、「CORS 設定の削除」を参照してください。

Amazon Lightsail にコンテナをデプロイして管理する

Amazon Lightsail コンテナサービスは、高度スケーラブルコンピューティングとネットワークリソースで、コンテナをデプロイ、実行、管理することができます。コンテナは、コードとその依存関係をパッケージ化するソフトウェアのスタンダード単位で、アプリケーションが 1 つのコンピューティング環境から別のコンピューティング環境に迅速かつ確実に実行します。

Lightsail コンテナサービスは、ローカルマシンで作成したイメージ、または Amazon ECR Public Gallery などのオンラインリポジトリのイメージを使用して、 AWS インフラストラクチャでコンテナを実行できるようにするコンピューティング環境と考えることができます。

Docker などのソフトウェアをインストールすることで、ローカルマシン上でコンテナをローカルで実行することもできます。Amazon Elastic Container Service (Amazon ECS) と Amazon Elastic Compute Cloud (Amazon EC2) は、コンテナを実行できる AWS インフラストラクチャ内の別のリソースです。詳細については、Amazon ECS 開発者ガイドを参照してください。

目次

- コンテナ
- Lightsail コンテナサービスの要素
  - Lightsail コンテナサービス
  - コンテナサービスの容量 (スケールとパワー)
  - 料金
  - デプロイ
  - デプロイバージョン
  - コンテナイメージソース
  - コンテナサービス ARN
  - パブリックエンドポイントとデフォルトドメイン
  - カスタムドメインと SSL/TLS 証明書
  - コンテナログ
  - メトリクス
- Lightsail コンテナサービスを使用する

コンテナ

コンテナは、コードと依存関係をパッケージ化するソフトウェアのスタンダード単位で、1 つのコンピューティング環境から別のコンピューティング環境へアプリケーションを迅速かつ確実に実行します。開発環境でコンテナを実行し、本番稼働前環境にデプロイしてから、本稼働環境にデプロイできます。開発環境がローカルマシンであるか、本番稼働前環境がデータセンターの物理サーバーであるか、運用環境がクラウドのバーチャルプライベートサーバーであるかにかかわらず、コンテナは確実に実行されます。

コンテナイメージは軽量で、スタンドアローンで実行可能な、アプリケーションの実行に必要なもの (コード、ランタイム、システムツール、システムライブラリ、設定) が全て含まれるソフトウェアのパッケージです。コンテナイメージは、ランタイム時にコンテナになります。アプリケーションとその依存関係をコンテナ化することで、ソフトウェアをデプロイするオペレーティングシステムとインフラストラクチャ上で正しく動作するかどうかを心配する必要がなくなり、コードに集中する時間を増やすことができます。

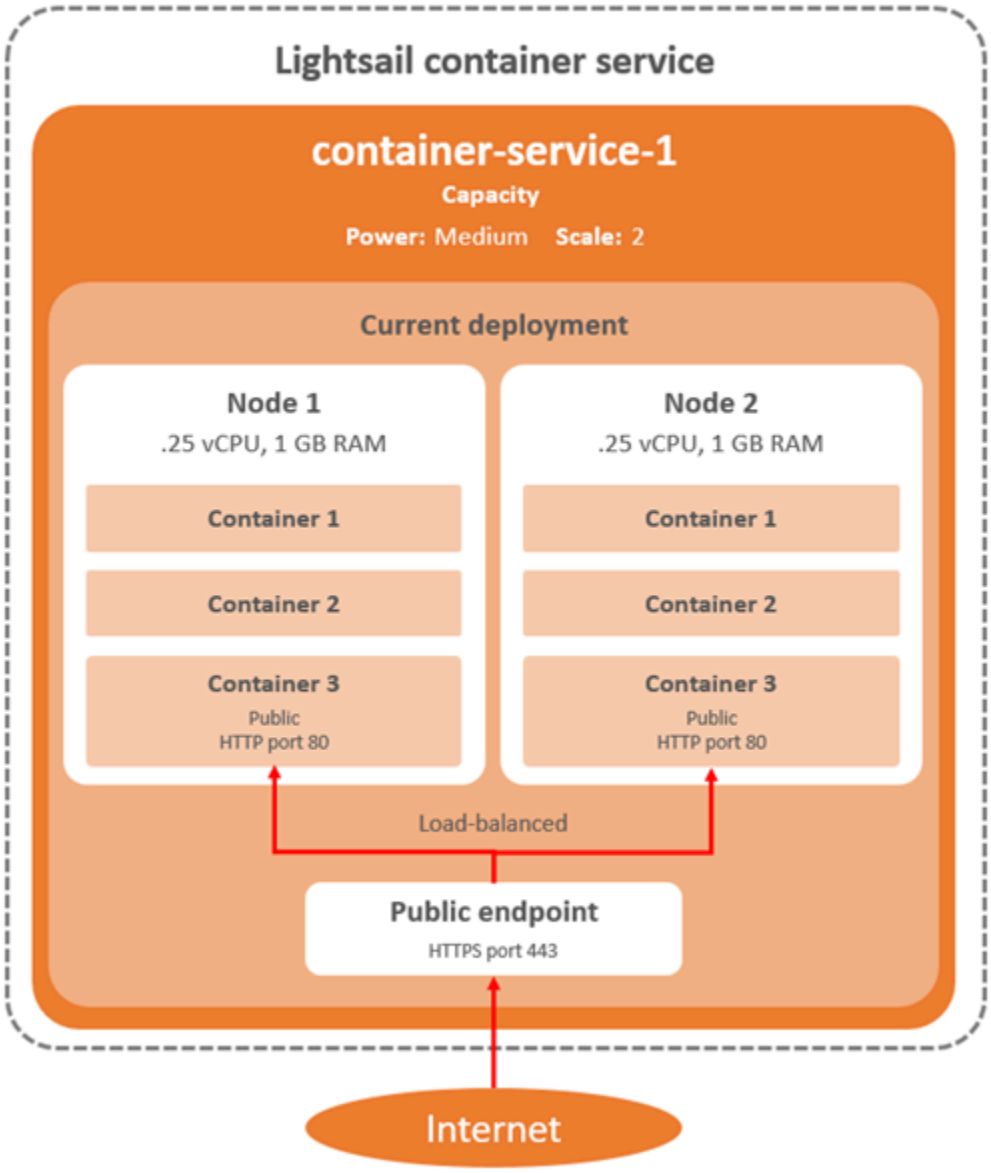
コンテナとコンテナイメージの詳細については、Docker ドキュメントの「コンテナとは」を参照してください。

Lightsail コンテナサービスの要素

Lightsail コンテナサービスの使用を開始する前に、以下の主な要素を理解しておく必要があります。

Lightsail コンテナサービス

コンテナサービスは、Lightsail コンピューティングリソースで、Lightsail が使用可能な AWS リージョン で作成できます。コンテナサービスは、いつでも作成および削除できます。詳細については、「Lightsail コンテナサービスを作成する」および「Lightsail コンテナサービスを削除する」を参照してください。



コンテナサービス容量 (スケールとパワー)

コンテナサービスを最初に作成するときは、以下の容量パラメータを選択する必要があります。

- スケール — コンテナのワークロードを実行するコンピューティングノードの数です。コンテナのワークロードは、サービスのコンピューティングノード間でコピーされます。コンテナサービスには最大 20 のコンピューティングノードを指定できます。可用性と容量を向上させるために、サービスを強化させるノードの数に応じてスケールを選択します。コンテナへのトラフィックは、ノード全体にロードバランスされます。

- パワー — コンテナサービス内の各ノードのメモリと vCPUs です。選択できるパワーは、Nano (Na)、Micro (Mi)、Small (Sm)、Medium (Md)、Large (Lg)、Xlarge (Xl) で、それぞれメモリと vCPUs の容量が徐々に大きくなっています。

コンテナサービスのスケールを 1 より大きく指定すると、コンテナワークロードはサービスの複数のコンピューティングノードにコピーされます。例えば、サービスのスケールが 3 で、パワーが Nano の場合、コンテナワークロードのコピーが 3 つあり、それぞれ 512 MB の RAM と 0.25 の vCPUs を持つ 3 つのコンピューティングリソースで実行されています。受信トラフィックは、3 つのリソース間でロードバランスされます。コンテナサービスに指定する容量が大きいほど、処理できるトラフィックが増えます。

コンテナサービスのパワーとスケールは、プロビジョンが不足している場合にダウンタイムに関係なくいつでも動的に増やすことができ、プロビジョンが過剰している場合は減らすことができます。Lightsail は、現在のデプロイとともに容量の変更を自動的に管理します。詳細については、「コンテナサービスの容量を変更する」を参照してください。

料金

コンテナサービスの月額料金は、そのパワーの価格にコンピューティングノード数 (サービスのスケール) を乗算して計算されます。例えば、ミディアムパワーのサービスの価格が 40 USD、コンピューティングノードが 3 である場合、1 か月あたり 120 USD になります。コンテナサービスが有効か無効か、デプロイがあるかどうかに関わらず、コンテナサービスに対して課金されます。コンテナサービスの課金を停止するには、コンテナサービスを削除する必要があります。

各コンテナサービスには、設定された容量に関係なく、500 GB の月次データ転送クォータが含まれます。データ転送クォータは、選択したサービスのパワーとスケールに関わらず変更されることはありません。クォータを超えるインターネットへのデータ転送では、超過料金が発生し、GB あたり AWS リージョン 0.09 USD から発生します。クォータを超過したインターネットからのデータ転送には、超過料金が発生しません。詳細については、Lightsail 料金表ページ を参照してください。

デプロイ

Lightsail コンテナサービスでデプロイを作成できます。デプロイは、サービスで起動するコンテナワークロードの仕様のセットです。

デプロイ内の各コンテナエントリに対して、以下のパラメータを指定できます。

- 起動されるコンテナ名

- コンテナで使用するソースコンテナイメージ
- コンテナの起動時に実行するコマンド
- コンテナに適用する環境可変
- コンテナで開くネットワークポート
- コンテナサービスのデフォルトドメインを介してパブリックにアクセスできるようにする、デプロイ内のコンテナ

> **Note**
>
> デプロイ内の 1 つのコンテナのみが、各コンテナサービスに対してパブリックにアクセス可能にすることができます。

次のヘルスチェックパラメータは、デプロイの起動後にデプロイのパブリックエンドポイントに適用されます。

- ヘルスチェックを実行するディレクトリパス。
- 間隔秒、タイムアウト秒、成功コード、正常しきい値、異常しきい値など、ヘルスチェックの高度な設定。

コンテナサービスは、一度に 1 つのアクティブなデプロイを持つことができ、デプロイは最大 10 個のコンテナエントリを持つことができます。デプロイは、コンテナサービスの作成と同時に作成する、あるいはサービスが起動され実行されてから作成することができます。詳細については、「コンテナサービスのデプロイの作成と管理」を参照してください。

デプロイバージョン

コンテナサービスで作成するすべてのデプロイは、デプロイバージョンとして保存されます。既存のデプロイのパラメータを変更すると、コンテナがサービスに再デプロイされ、デプロイが変更された場合は新しいデプロイバージョンが作成されます。各コンテナサービスの最新の 50 のデプロイバージョンが保存されます。50 のデプロイバージョンのいずれかを使用して、新しいデプロイを同じコンテナに作成できます。詳細については、「コンテナサービスのデプロイの作成と管理」を参照してください。

コンテナイメージソース

デプロイを作成するときは、デプロイ内の各コンテナエントリにソースコンテナイメージを指定する必要があります。デプロイを作成した直後に、コンテナサービスは指定したソースからイメージを取り出し、それらを使用してコンテナを作成します。

以下のソースから指定してイメージを取り出せます。

- Amazon ECR Public Gallery、その他のパブリックコンテナイメージレジストリなどのパブリックレジストリ。Amazon ECR Public の詳細については、「Amazon ECR Public ユーザーガイド」の「Amazon Elastic Container Registry Public とは？」を参照してください。
- ローカルマシンからプッシュされたイメージをコンテナサービスに追加します。ローカルマシンでコンテナイメージを作成する場合は、コンテナサービスにプッシュして、デプロイの作成時に使用できます。詳細については、「コンテナサービスイメージを作成する」および「コンテナイメージをプッシュして管理する」を参照してください。

Lightsail コンテナサービスは、Linux ベースのコンテナイメージをサポートしています。Windows ベースのコンテナイメージは現在サポートされていませんが、Windows で Docker、 AWS Command Line Interface (AWS CLI)、および Lightsail Control (lightsailctl) プラグインを実行して、Linux ベースのイメージを構築してLightsailコンテナサービスにプッシュできます。

コンテナサービス ARN

Amazon リソースネーム (ARNs) は AWS リソースを一意に識別します。IAM ポリシーや API コールなど AWS、すべての でリソースを明確に指定する必要がある場合は、ARN が必要です。

コンテナサービスの ARN を取得するには、`GetContainerServices` Lightsail API アクションを使用し、`serviceName` パラメータを使用してコンテナサービスの名前を指定します。コンテナサービス ARN は、次の例に示すように、そのアクションの結果に一覧表示されます。詳細については、「Amazon Lightsail API Reference」の「GetContainerServices」を参照してください。

以下のような出力結果が表示されるはずです。

```
{
    "containerServices": [
        {
            "containerServiceName": "container-service-1",
            "arn": "arn:aws:lightsail: :111122223333:ContainerService/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "createdAt": "2024-01-01T00:00:00+00:00",
```

```
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
        },
        .....
}
```

パブリックエンドポイントとデフォルトドメイン

デプロイを作成するときに、コンテナサービスのパブリックエンドポイントとして機能するデプロイ内のコンテナエントリを指定できます。パブリックエンドポイントコンテナ上のアプリケーションは、コンテナサービスのランダムに生成されたデフォルトドメインを介して、インターネット上でパブリックにアクセスできます。デフォルトのドメインは の形式でhttps://*<ServiceName>*.*<RandomGUID>*.*<AWSRegion>*.cs.amazonlightsail.com、*<Service*はコンテナサービスの名前、*<RandomGUID>* は AWS リージョン Lightsailアカウントの 内のコンテナサービスのランダムに生成されたグローバル一意識別子、*<AWSRegion>* はコンテナサービスが作成された AWS リージョン です。Lightsail コンテナサービスのパブリックエンドポイントは HTTPS のみをサポートし、TCP または UDP トラフィックはサポートされていません。サービスのパブリックエンドポイントにできるコンテナは 1 つだけです。したがって、アプリケーションのフロントエンドをホストしているコンテナをパブリックエンドポイントとして選択し、残りのコンテナは内部的にアクセス可能であることを確認してください。

コンテナサービスのデフォルトドメインか、独自のカスタムドメイン (登録されたドメイン名) を使用できます。コンテナサービスでのカスタムドメインの使用の詳細については、「コンテナサービスでカスタムドメインを有効にして管理する」を参照してください。

プライベートドメイン

また、すべてのコンテナサービスは、プライベートドメインを保持していて、*<ServiceName>*.service.local にフォーマットされており*<ServiceName>*はコンテナサービスの名前です。プライベートドメインを使用して、サービスと同じ AWS リージョンにある別の Lightsail リソースからコンテナサービスにアクセスします。プライベートドメインは、サービスのデプロイメントでパブリックエンドポイントを指定しない場合、コンテナサービスにアクセスする唯一の方法です。パブリックエンドポイントを指定しなくても、コンテナサービスに対してデフォルトのドメインが生成されますが、観覧しようとすると、404 No Such Service エラーメッセージが表示されます。

コンテナサービスのプライベートドメインを使用して特定のコンテナにアクセスするには、接続要求を受け入れるコンテナのオープンポートを指定する必要があります。これを実行するには、リク

エストのドメインを*<ServiceName>*.service.local:*<PortNumber>* にフォーマットし、この中で*<ServiceName>*はコンテナサービスの名前、*<PortNumber>*は、接続したいコンテナのオープンポートです。例えば、コンテナサービスにデプロイcontainer-service-1 を作成し、Redis コンテナを指定してポート 6379 が開いている場合は、リクエストのドメインを*container-service-1*.service.local:*6379* にフォーマットします。

カスタムドメインと SSL/TLS 証明書

デフォルトドメインを使用する代わりに、コンテナサービスでカスタムドメインを最大 4 つ使用できます。例えば、カスタムドメインのトラフィックを、example.com のようにパブリックエンドポイントとしてラベル付けされたデプロイ内のコンテナにルーティングできます。

カスタムドメインをサービスで使用するには、まず、使用するドメインの SSL/TLS 証明書をリクエストする必要があります。その後、ドメインの DNS に CNAME レコードのセットを追加して、SSL/TLS 証明書を検証する必要があります。SSL/TLS 証明書の検証した後、有効な SSL/TLS 証明書をサービスに添付して、コンテナサービスでカスタムドメインを有効化します。詳細については、「Lightsail コンテナサービスの SSL/TLS 証明書を作成する」、「Lightsail コンテナサービスの SSL/TLS 証明書を検証する」、および「Lightsail コンテナサービスでカスタムドメインを有効にして管理する」を参照してください。

コンテナログ

コンテナサービスのすべてのコンテナは、コンテナのオペレーションを診断するためにアクセスできるログを生成します。ログは、コンテナ内で実行されている stdout および stderr にプロセスの流れを提供します。詳細については、「コンテナサービスログを表示する」を参照してください。

メトリクス

コンテナサービスのメトリックスをモニタリングして、過剰使用が原因の可能性とする問題を診断します。メトリックスをモニタリングして、サービスのプロビジョニングが不足していないか、あるいは過剰にプロビジョニングされているかを判断することもできます。詳細については、「コンテナサービスメトリクスを表示する」を参照してください。

Lightsail コンテナサービスを使用する

次の内容は、Lightsail コンテナサービスを管理し、ローカルマシンからサービスにイメージをプッシュするか、パブリックレジストリからコンテナイメージを使用するための一般的なステップです。

Lightsail コンテナサービスを管理し、デプロイでコンテナイメージを使用する方法

1. Lightsail アカウントにコンテナサービスを作成する。詳細については、「Lightsail コンテナサービスを作成する」を参照してください。
2. 次のいずれかのオプションを使用し、Lightsail コンテナサービスでコンテナイメージを使用します。
   - ローカルマシンからコンテナイメージを使用する – ローカルマシンにソフトウェアをインストールして独自のコンテナイメージを作成し、Lightsail コンテナサービスにプッシュできます。詳細については、以下のガイドを参照してください。
     - Lightsail コンテナサービスのコンテナイメージを管理するためのソフトウェアをインストールする
     - Lightsail コンテナサービスのコンテナイメージを作成する
     - Lightsail コンテナサービスのコンテナイメージをプッシュして管理する
   - パブリックレジストリのコンテナイメージを使用する – Amazon ECR Public Gallery などのパブリックレジストリで Lightsail コンテナサービスのコンテナイメージを検索して使用できます。Amazon ECR Public Gallery の詳細については、「Amazon ECR Public ユーザーガイド」の「Amazon Elastic Container Registry Public とは?」を参照してください。
3. Lightsail コンテナサービスのコンテナイメージを管理するためにソフトウェアのインストール
4. Lightsail コンテナサービスのコンテナイメージの作成
5. Lightsail コンテナサービスでコンテナイメージのプッシュと管理
6. コンテナを設定して起動するデプロイをコンテナサービス内に作成します。詳細については、「Lightsail コンテナサービスのデプロイを作成して管理する」を参照してください。
7. コンテナサービスの以前のデプロイを表示します。以前のデプロイ バージョンを使用して、新しいデプロイを作成できます。詳細については、「Lightsail コンテナサービスのデプロイバージョンを表示して管理する」を参照してください。
8. コンテナサービスのコンテナのログを表示します。詳細については、「Lightsail コンテナサービスのコンテナログを表示する」を参照してください。
9. コンテナで使用するドメイン用の SSL/TLS 証明書を作成します。詳細については、「Lightsail コンテナサービスの SSL/TLS 証明書を作成する」を参照してください。
10. ドメインの DNS にレコードを追加して、SSL/TLS 証明書を検証します。詳細については、「Lightsail コンテナサービスの SSL/TLS 証明書を検証する」を参照してください。

11. 有効な SSL/TLS 証明書をコンテナサービスに添付して、カスタムドメインを有効にします。詳細については、「Lightsail コンテナサービスのカスタムドメインを有効にして管理する」を参照してください。
12. コンテナサービスの使用状況メトリクスをモニタリングします。詳細については、「コンテナサービスメトリクスを表示する」を参照してください。
13. (オプション) パワースペックを垂直方向に増やし、スケールスペックを水平方向に増やして、コンテナサービスの容量をスケールします。詳細については、「Lightsail コンテナサービスの容量を変更する」を参照してください。
14. コンテナサービスを使用していない場合は、月額料金が発生しないようにコンテナサービスを削除します。詳細については、「Lightsail コンテナサービスを削除する」を参照してください。

Lightsail で高可用性コンテナサービスを作成する

このガイドでは、Lightsail コンソールを使用して作成可能な Amazon Lightsail コンテナサービスとその設定について説明します。

開始する前に、Lightsail コンテナサービスの基礎を学んでおくことをお勧めします。詳細については、「コンテナサービス」を参照してください。

コンテナサービス容量 (スケールとパワー)

コンテナサービスの容量は、最初に作成するときに選択する必要があります。容量は、次に示すパラメータの組み合わせで構成されます。

- スケール: コンテナのワークロードを実行するコンピューティングノードの数。コンテナのワークロードは、サービスのコンピューティングノード間でコピーされます。コンテナサービスには最大 20 のコンピューティングノードを指定できます。可用性と容量を向上させるために、サービスを強化させるノードの数に応じてスケールを選択します。コンテナへのトラフィックは、すべてのノードでロードバランスされます。
- パワー: コンテナサービス内の各ノードのメモリと vCPUs。選択できるパワーは、ナノ (Na)、マイクロ (Mi)、スモール (Sm)、ミディアム (Md)、ラージ (Lg)、エクストララージ (Xl) で、それぞれメモリと vCPUs の容量が徐々に大きくなります。

着信トラフィックは、コンテナサービスのスケール (コンピューティングノードの数) 全体にわたってロードバランスされます。たとえば、Nano パワーでスケールが 3 のサービスの場合、コンテナワークロードのコピーが 3 つ実行されます。各ノードには 512 MB の RAM と 0.25 の vCPUs 容量

があります。受信トラフィックは、3 つのノード間でロードバランシングされます。選択したコンテナサービス容量が大きいほど、処理できるトラフィックが増えます。

プロビジョニングが不十分であることがわかった場合、コンテナサービスのパワーとスケールはダウンタイムなしでいつでも動的に増加でき、過剰な場合は減少することもできます。Lightsailは、現在のデプロイとともにキャパシティの変更を自動的に管理します。詳細については、「Lightsail コンテナサービスの容量を変更する」を参照してください。

料金

コンテナサービスの月額料金は、そのパワーの基本価格にスケール (コンピューティングノード数) を乗算して計算されます。たとえば、ミディアムレベルのパワーが 40 USD でスケールが 3 のサービスでは、月額 120 USD の費用がかかります。

各コンテナサービスには、構成された容量に関わらず、500 GB の月次データ転送クォータが含まれます。データ転送クォータは、選択したサービスのパワーとスケールに関わらず変更されることはありません。クォータを超えるデータをインターネットに転送すると、AWS リージョンによって異なる超過料金が、1 GB あたり 0.09 USD から発生します。クォータを超過したインターネットからのデータ転送には、超過料金が発生しません。詳細については、Lightsail 料金表ページ を参照してください。

コンテナサービスが有効か無効か、デプロイがあるかどうかに関係なく、コンテナサービスに対して課金されます。コンテナサービスの課金を停止するには、サービスを削除する必要があります。詳細については、「Lightsail コンテナサービスを削除する」を参照してください。

コンテナサービスステータス

コンテナサービスは、次に示す状態のいずれかになります。

- 保留中 – コンテナサービスを作成しています。
- 準備完了 – コンテナサービスは実行中ですが、アクティブなコンテナデプロイがありません。
- デプロイ – デプロイがコンテナサービスに対して起動されます。
- 実行中 – コンテナサービスが実行中で、アクティブなデプロイがあります。
- 更新中 – コンテナサービスの容量またはそのカスタムドメインが更新されています。
- 削除中 – コンテナサービスが削除されています。削除を選択した後、コンテナサービスはしばらくの間この状態を表示します。
- 無効 – コンテナサービスが無効になり、アクティブなデプロイとコンテナ (もし存在すれば) がシャットダウンされます。

コンテナサービスのサブステータス

コンテナサービスがデプロイまたは更新中の状態の場合、コンテナサービスの状態の下に、追加で次のサブ状態のいずれかが表示されます。

- システムリソースの作成 – コンテナサービスのシステムリソースが作成されています。
- ネットワークインフラストラクチャの作成 – コンテナサービスのネットワークインフラストラクチャが作成されています。
- プロビジョニング証明書 – コンテナサービス用の SSL/TLS 証明書が作成されています。
- プロビジョニングサービス – コンテナサービスがプロビジョニングされています。
- デプロイの作成 – デプロイがコンテナサービス上に作成されています。
- ヘルスチェック評価 – デプロイの正常性が評価されています。
- デプロイのアクティベーション – デプロイがアクティベーションされています。

コンテナサービスが保留中状態の場合、コンテナサービスの状態の下に、次の追加のサブ状態のいずれかが表示されます。

- 証明書の制限を超えました – コンテナサービスに必要な SSL/TLS 証明書の数はアカウントで許可されている証明書の最大数を超えています。
- 未知のエラー – コンテナサービスの作成中にエラーが発生しました。

コンテナサービスの作成

以下の手順に従って、Lightsail コンテナサービスを作成します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. コンテナサービスの作成を選択します。
4. 「コンテナサービスの作成」ページで「 の変更 AWS リージョン」を選択し、コンテナサービスの AWS リージョン を選択します。
5. コンテナサービスの容量を選択します。詳細については、本ガイドの「コンテナサービス容量 (スケールとパワー)」セクションを参照してください。
6. 以下のステップを実行して、コンテナサービスの作成と同時に起動されるデプロイを作成します。それ以外の場合は、手順 7 に進み、デプロイなしでコンテナサービスを作成します。

公開レジストリーからコンテナイメージを使用する予定の場合は、デプロイでコンテナサービスを作成します。ローカルマシン上のコンテナイメージを使用する予定の場合は、デプロイなしでサービスを作成します。サービスの起動と実行後に、ローカルマシンからコンテナサービスにコンテナイメージをプッシュできます。コンテナサービスに登録されているプッシュされたコンテナイメージを使用してデプロイを作成できます。

a. [Create a deployment] (デプロイを作成する) を選択します。

b. 以下のオプションのいずれかを選択してください。

- デプロイ例の選択 – Lightsail チームによって事前設定されたデプロイパラメータのセットを用いてキュレートされたコンテナイメージを使用し、デプロイを作成するためには、このオプションを選択します。このオプションは、一般的なコンテナをコンテナサービス上で起動して実行するための最速かつ最も簡単な方法を提供します。
- カスタムデプロイを指定 – 選択したコンテナを指定してデプロイを作成するには、このオプションを選択します。

デプロイフォームビューが開き、新しいデプロイパラメータを入力することができます。

c. デプロイのパラメータを入力します。指定できるデプロイパラメータの詳細については、「Lightsail コンテナサービスのデプロイを作成して管理する」ガイドの「デプロイパラメータ」セクションを参照してください。

d. [コンテナエントリの追加] を選択することによって、デプロイに複数のコンテナエントリを追加できます。デプロイには最大 10 のコンテナエントリを追加することができます。

e. デプロイのパラメータを入力し終わったら、[保存してデプロイ] を選択して、コンテナサービス上にデプロイを作成します。

7. コンテナサービスの名前を入力します。

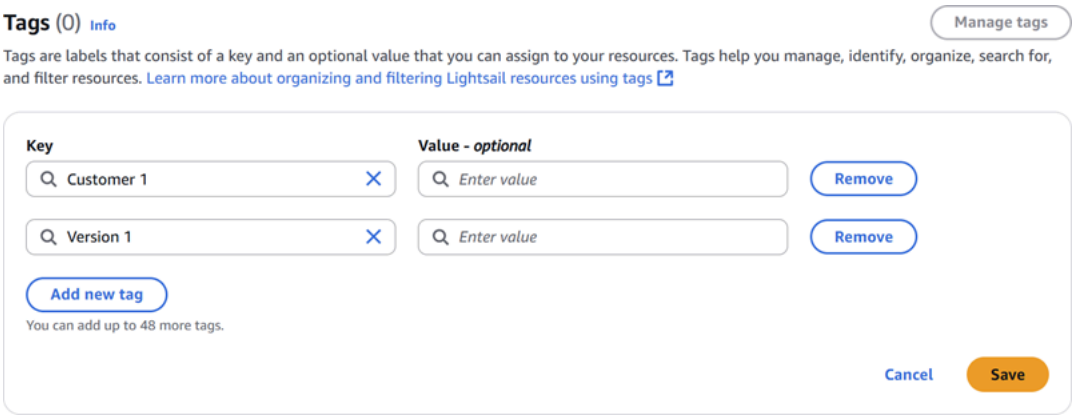
コンテナサービス名は、次のものである必要があります。

- Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
- 2〜63 文字を使用する必要があります。
- 英数字またはハイフンのみを使用する必要があります。
- ハイフン (-) で単語を区切ることができますが、名前の先頭または末尾に付けることはできません。

> **Note**
>
> 指定する名前は、コンテナサービスのデフォルトのドメイン名の一部となり、一般ユーザーに表示されます。

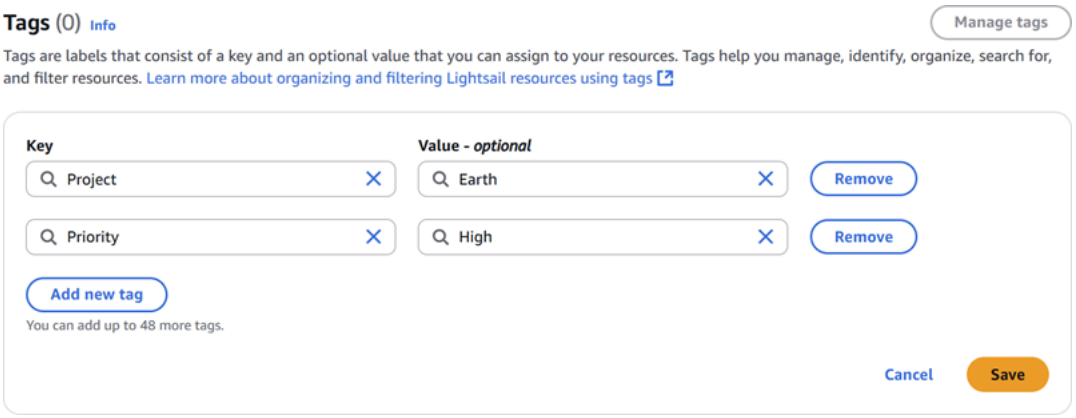
8. 以下のいずれかのオプションを選択して、コンテナサービスにタグを追加します。

   - [キーオンリータグの追加] または [タグを編集] (タグが追加済みの場合)。タグキーのテキストボックスに新しいタグを入力し、Enter キーを押します。タグの入力を完了したら、[保存] を選択してタグを追加し、追加しない場合は、[キャンセル] を選択します。



   - [Create a key-value tag] (キーバリュータグを作成) から [Key] (キー) テキストボックスにキーを入力し、[Value] (値) テキストボックスに値を入力します。タグの入力を完了したら、[保存] を選択し、追加しない場合は、[キャンセル] を選択します。

     キーバリューのタグは、保存する際に一つずつ追加することができます。さらにキーバリュータグを追加するには、以上のステップを繰り返します。



> **Note**
>
> 「キーのみ」のタグと「キーバリュー」のタグの詳細については、「タグ」を参照してください。

9. [コンテナサービスの作成] を選択します。

   新しいコンテナサービスの管理ページにリダイレクトされます。作成している間、新しいコンテナサービスのステータスは「保留中」となります。しばらくすると、現在のデプロイがない場合、サービスの状態は「準備完了」を表示し、デプロイが作成された場合、「実行中」が表示されます。

Lightsail コンテナサービスの Docker イメージの構築とテスト

Docker を使用すると、コンテナに基づく分散アプリケーションを構築、実行、テスト、デプロイできます。Amazon Lightsail コンテナサービスは、デプロイで Docker コンテナイメージを使用してコンテナを起動します。

このガイドでは、Dockerfile を使用してローカルマシン上にコンテナイメージを作成する方法を説明します。イメージが作成されたら、そのイメージを Lightsail コンテナサービスにプッシュしてデプロイできます。

このガイドの手順を完了するには、Docker の概要と機能についての基本的な理解が必要です。Docker の詳細については、「Docker とは」、「Docker の概要」を参照してください。

目次

- ステップ 1: 前提条件を満たす
- ステップ 2: Dockerfile を作成してコンテナイメージを構築する
- ステップ 3: 新しいコンテナイメージを実行する
- （オプション）ステップ 4: ローカルマシンで実行されているコンテナをクリーンアップする
- コンテナイメージの作成後の次のステップ

ステップ 1: 前提条件を満たす

作業を開始する前に、コンテナの作成に必要なソフトウェアをインストールし、Lightsail コンテナサービスにプッシュする必要があります。たとえば、Docker をインストールして使用して、Lightsail コンテナサービスで使用できるコンテナイメージを作成してビルドする必要があります。詳細については、「Amazon Lightsail コンテナサービス用のコンテナイメージを管理するソフトウェアのインストール」を参照してください。

ステップ 2: Dockerfile を作成してコンテナイメージを構築する

以下のステップを実行して Dockerfile を作成し、`mystaticwebsite` Docker コンテナイメージを構築します。コンテナイメージは、Ubuntu の Apache ウェブサーバーでホストされている単純な静的ウェブサイト用です。

1. `mystaticwebsite` の作成フォルダを作成し、Dockerfile を保存するローカルマシン上に配置します。
2. 先ほど作成したフォルダに Dockerfile を作成します。

   Dockerfile は、`.TXT` のようなファイル拡張子を使用しません。完全なファイル名は `Dockerfile` です。
3. コンテナイメージの設定方法に応じて次のコードブロックのいずれかをコピーし、Dockerfile に貼り付けます。

   - Hello World メッセージを含む単純な静的なウェブサイトコンテナイメージを作成する場合、次のコードブロックをコピーして Dockerfile に貼り付けます。このコードサンプルは Ubuntu 18.04 イメージを使用します。`RUN` の手順により、パッケージキャッシュが更新され、Apache がインストールされて設定されてから、Hello World のメッセージがウェブサーバーのドキュメントルートに出力されます。`EXPOSE` の命令はコンテナ上のポート 80 を公開し、`CMD` の命令はウェブサーバーを起動します。

     ```
     FROM ubuntu:18.04

     # Install dependencies
     RUN apt-get update && \
      apt-get -y install apache2

     # Write hello world message
     RUN echo 'Hello World!' > /var/www/html/index.html
     ```

```
# Open port 80
EXPOSE 80

# Start Apache service
CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

- 静的ウェブサイトコンテナイメージに独自の HTML ファイルセットを使用する場合には、html フォルダを Dockerfile の保存先と同じフォルダに配置します。次に、HTML ファイルをそのフォルダに入れます。

  HTML ファイルを html フォルダに保存したら、次のコードブロックをコピーして Dockerfile に貼り付けます。このコードサンプルは Ubuntu 18.04 イメージを使用します。RUN の命令はパッケージキャッシュを更新し、Apache をインストールして設定します。COPY の命令は html フォルダの内容をウェブサーバーのドキュメントルートにコピーします。EXPOSE の命令はコンテナ上のポート 80 を公開し、CMDの命令はウェブサーバーを起動します。

```
FROM ubuntu:18.04

# Install dependencies
RUN apt-get update && \
 apt-get -y install apache2

# Copy html directory files
COPY html /var/www/html/

# Open port 80
EXPOSE 80

CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

4. コマンドプロンプトまたはターミナルウィンドウを開き、 Dockerfile を格納しているフォルダにディレクトリを変更します。
5. 次のコマンドを入力して、フォルダ内の Dockerfile を使用してコンテナイメージを構築します。このコマンドは、 mystaticwebsite という新しい Docker コンテナイメージをビルドします。

```
docker build -t mystaticwebsite .
```

   イメージが正常に構築されたことを確認するメッセージが表示されます。

6. 次のコマンドを入力して、ローカルマシン上のコンテナイメージを表示します。

```
docker images --filter reference=mystaticwebsite
```

次の例に示すような結果が表示され、作成された新しいコンテナイメージが示されます。

```
C:\Users\[redacted]\Documents\Docker\Dockerfiles\mystaticwebsite>docker images --filter reference=mystaticwebsite
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
mystaticwebsite     latest              8f7ffd1013e0        8 minutes ago       199MB
```

新しく構築したコンテナイメージは、ローカルマシン上で新しいコンテナを実行させることによってテストすることができます。次の手順は、本ガイドの「ステップ 3: 新しいコンテナイメージを実行する」セクションを参照してください。

ステップ 3: 新しいコンテナイメージを実行する

作成した新しいコンテナイメージを実行するには、以下の手順に従います。

1. コマンドプロンプトまたはターミナルウィンドウに次のコマンドを入力して、本ガイドの「ステップ 2: Dockerfile を作成してコンテナイメージを構築する」のセクションで構築したコンテナーイメージを実行します。-p 8080:80 オプションは、コンテナ上の公開されたポート 80 をホストシステム上のポート 8080 にマッピングします。-d オプションは、コンテナをデタッチモードで実行するように指定します。

   ```
   docker container run -d -p 8080:80 --name mystaticwebsite mystaticwebsite:latest
   ```

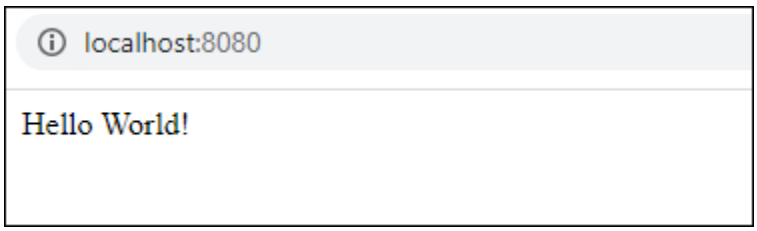
2. 次のコマンドを入力して、実行中のコンテナを表示します。

   ```
   docker container ls -a
   ```

   次の例に示すような結果が表示され、新しい実行中のコンテナが示されます。

   ```
   C:\Users\[redacted]\Documents\Docker\Dockerfiles\mystaticwebsite>docker container ls -a
   CONTAINER ID        IMAGE                   COMMAND                  CREATED             STATUS              PORTS                   NAMES
   62382081e06b        mystaticwebsite:latest  "/bin/sh -c /root/ru…"   6 minutes ago       Up 6 minutes        0.0.0.0:8080->80/tcp    mystaticwebsite
   ```

3. コンテナが起動して実行されていることを確認するには、新しいブラウザウィンドウで http://localhost:8080 を開きます。次の例に示すようなメッセージが表示されます。これにより、コンテナがローカルマシン上で稼働していることが確認されます。



新しく構築されたコンテナイメージを Lightsail アカウントにプッシュする準備が整いました。これにより、Lightsail コンテナサービスにデプロイできるようになります。詳細については、Amazon Lightsail コンテナサービスのコンテナイメージのプッシュと管理 を参照してください。

（オプション）ステップ 4: ローカルマシンで実行されているコンテナをクリーンアップする

Lightsail コンテナサービスにプッシュできるコンテナイメージを作成したので、このガイドの手順に従って、ローカルマシンで実行されているコンテナをクリーンアップします。

ローカルマシンで実行されているコンテナをクリーンアップするには、以下の手順にを実行します。

1. ローカルマシンで実行されているコンテナを表示するには、次のコマンドを実行します。

   ```
   docker container ls -a
   ```

   次のような結果が表示され、ローカルマシンで実行されているコンテナの名前が一覧表示されます。

   ```
   C:\Users\[redacted]\Documents\Docker\Dockerfiles\mystaticwebsite>docker container ls -a
   CONTAINER ID   IMAGE                     COMMAND                  CREATED          STATUS          PORTS                    NAMES
   62382081e06b   mystaticwebsite:latest    "/bin/sh -c /root/ru…"   6 minutes ago    Up 6 minutes    0.0.0.0:8080->80/tcp     mystaticwebsite
   ```

2. 次のコマンドを実行して、このガイドの前述の部分で作成した実行中のコンテナを削除します。これにより、コンテナは強制的に停止され、完全に削除されます。

   ```
   docker container rm <ContainerName> --force
   ```

   コマンドで、<ContainerName> (コンテナ名) を、削除する設定セットの名前で置き換えます。

   例:

```
docker container rm mystaticwebsite --force
```

このガイドを元に作成されたコンテナは削除されます。

コンテナイメージの作成後の次のステップ

コンテナイメージを作成した後、デプロイの準備が整ったらそれらを Lightsail コンテナサービスにプッシュします。詳細については、「Lightsail コンテナサービスイメージを管理する」を参照してください。

トピック

- コンテナサービスの Lightsail コンテナイメージをプッシュ、表示、削除する
- Docker AWS CLI、およびコンテナ用のLightsailコントロールプラグインをインストールする
- Lightsail コンテナサービスに対して、Amazon ECR プライベートリポジトリへのアクセスを付与する

コンテナサービスの Lightsail コンテナイメージをプッシュ、表示、削除する

Amazon Lightsail コンテナサービスでデプロイを作成する場合は、コンテナエントリごとに出典コンテナイメージを指定する必要があります。Amazon ECR Public Gallery などの公開レジストリのイメージを使用することができます。または、ローカルマシンで作成したイメージを使用できます。このガイドでは、コンテナイメージをローカルマシンから Lightsail コンテナサービスにプッシュする方法を説明しています。コンテナイメージの作成に関する詳細については、「コンテナサービスイメージの作成」を参照してください。

目次

- 前提条件
- コンテナイメージをローカルマシンからコンテナサービスにプッシュする
- コンテナサービスに保存されているコンテナイメージを表示する
- コンテナサービスに保存されているコンテナイメージを削除する

前提条件

コンテナサービスへのコンテナイメージのプッシュを開始する前に、次の必要条件を完了します。

- Lightsail アカウントにコンテナサービスを作成する。詳細については、Amazon Lightsail コンテナサービスの作成 を参照してください。
- ローカルマシンにソフトウェアをインストールし、独自のコンテナイメージを作成し、Lightsail コンテナサービスにプッシュします。詳細については、「Amazon Lightsail コンテナサービス用のコンテナイメージを管理するソフトウェアのインストール」を参照してください。
- Lightsail コンテナサービスにプッシュしたい独自のコンテナイメージを、ローカルマシンに作成する。詳細については、「Amazon Lightsailコンテナサービスでのコンテナイメージの作成」を参照してください。

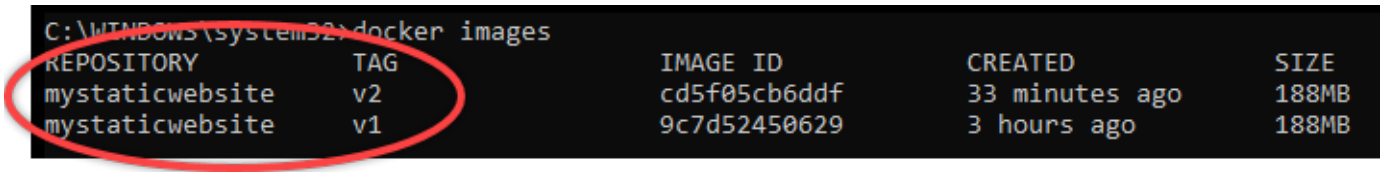
コンテナイメージをローカルマシンからコンテナサービスにプッシュする。

コンテナイメージをコンテナサービスにプッシュするには、以下の手順を実行します。

1. コマンドプロンプトまたはターミナルウィンドウを開きます。
2. コマンドプロンプトまたはターミナルウィンドウで、次のコマンドを入力して、現在ローカルマシン上にある Docker イメージ を表示します。

```
docker images
```

3. その結果、コンテナサービスにプッシュしたいコンテナイメージ名 (リポジトリ名) とそのタグが見つけられます。これは次のステップで必要になるため、書きとめておきます。



4. 次のコマンドを入力して、ローカルマシン上のコンテナイメージをコンテナサービスにプッシュします。

```
aws lightsail push-container-image --region <Region> --service-
name <ContainerServiceName> --label <ContainerImageLabel> --
image <LocalContainerImageName>:<ImageTag>
```

コマンドを、以下のように置き換えます。

- *<Region>* をコンテナサービスが作成された AWS リージョンに置き換えます。
- *<ContainerServiceName>* をコンテナサービス名に置き換えます。
- *<ContainerImageLabel>* を、コンテナサービスに保存される際にコンテナイメージに与えたいラベルに置き換えます。登録しているコンテナイメージの異なるバージョンを追跡する際に使用できる記述的ラベルを指定します。

  このラベルは、コンテナサービスによって生成されたコンテナイメージ名の一部になります。例えば、コンテナサービス名が container-service-1 の場合には、コンテナイメージラベルは mystaticsite になり、これがユーザーがプッシュするコンテナイメージの最初のバージョンになります。そしてコンテナサービスによって生成されたイメージ名は :container-service-1.mystaticsite.1 になります。
- *<LocalContainerImageName>* を、コンテナサービスにプッシュしたいコンテナイメージ名に置き換えます。この手順の前のステップで、コンテナイメージ名は取得しています。
- *<Image Tag>* を、コンテナサービスにプッシュしたいコンテナイメージのタグに置き換えます。この手順の前のステップで、コンテナイメージのタグは取得しています。

例:

```
aws lightsail push-container-image --region us-west-2 --service-name myservice --
label mystaticwebsite --image mystaticwebsite:v2
```

次の例のような結果が表示されていれば、コンテナイメージがコンテナサービスにプッシュされたことを確認できます。

このガイドの以下の「コンテナサービスに保存されているコンテナイメージを表示する」セクションを参照して、Lightsail コンソールでコンテナサービスにプッシュされたコンテナイメージを確認してください。

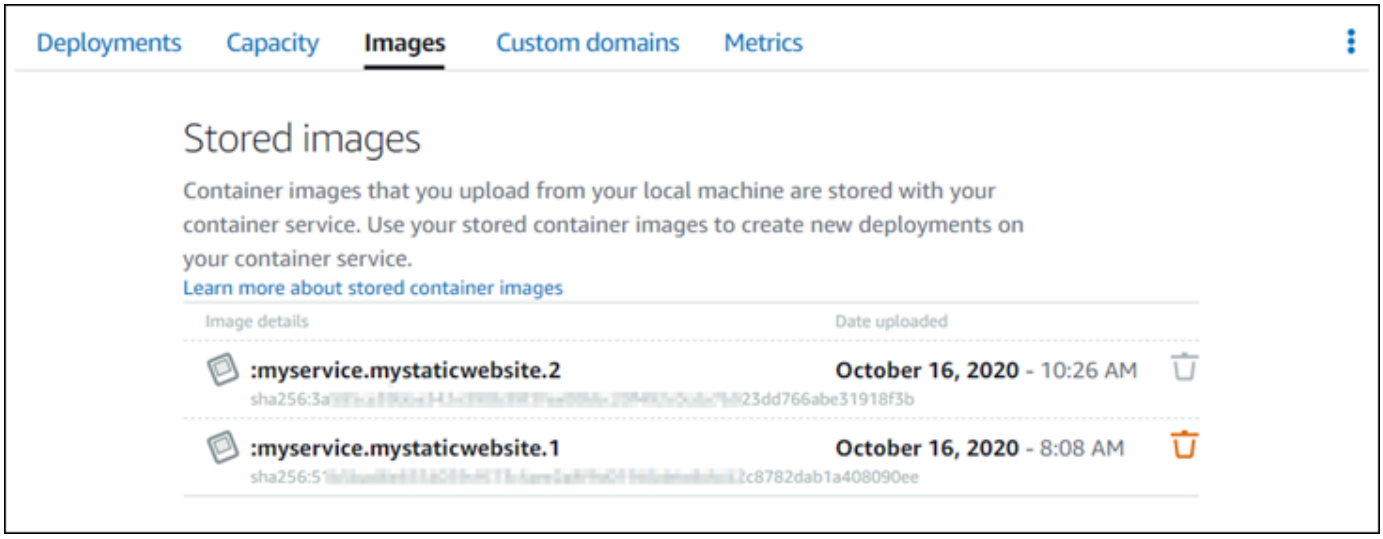
コンテナサービスに保存されているコンテナイメージを表示する

コンテナサービスにプッシュ、保存されているコンテナイメージを表示するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. 表示したい保存されたコンテナイメージのコンテナサービス名を選択します。
4. コンテナサービス管理ページで、[イメージ] タブを選択します。

> **Note**
>
> コンテナサービスにイメージをプッシュしていない場合、[イメージ] タブは表示されません。コンテナサービスのイメージタブを表示するには、まずコンテナイメージをサービスにプッシュする必要があります。

[イメージ] ページには、コンテナサービスにプッシュされ、現在ユーザーのサービス内に保存されているコンテナイメージの一覧が表示されます。現在のデプロイで使用されているコンテナイメージは削除できないため、削除アイコンは灰色に表示されます。



サービスに保存されているコンテナイメージを使用して、デプロイが作成できます。詳細については、「Amazon Lightsail コンテナサービスのデプロイの作成と管理」を参照してください。

コンテナサービスに保存されているコンテナイメージを削除する

コンテナサービスにプッシュ、保存されているコンテナイメージを削除するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. 現在のデプロイを表示したいコンテナサービス名を選択します。
4. コンテナサービス管理ページで、[イメージ] タブを選択します。

> **Note**
>
> コンテナサービスにイメージをプッシュしていない場合、[イメージ] タブは表示されません。コンテナサービスのイメージタブを表示するには、まずコンテナイメージをサービスにプッシュする必要があります。

5. 削除したいコンテナイメージを見つけ、削除アイコン (ごみ箱) を選択します。

> **Note**
>
> 現在のデプロイで使用されているコンテナイメージは削除できないため、削除アイコンは灰色に表示されます。

6. 確認プロンプトが表示されたら、[はい、削除します] を選択して保存されたイメージの完全な削除を確定します。

   保存されたコンテナイメージは、コンテナサービスからただちに削除されます。

Docker AWS CLI、およびコンテナ用のLightsailコントロールプラグインをインストールする

Amazon Lightsail コンソールを使用して Lightsail コンテナサービスを作成すると、Amazon ECR Public Gallery などのオンライン公開レジストリからコンテナイメージを使ったデプロイを作成できます。独自のコンテナイメージを作成してコンテナサービスにプッシュするには、コンテナイメージを作成する予定のコンピューター上に、以下の追加ソフトウェアをインストールする必要があります。

- Docker – 独自のコンテナイメージを実行、テスト、作成することが可能で、イメージは Lightsail コンテナサービスで使用することができます。
- AWS Command Line Interface (AWS CLI) – 作成したコンテナイメージのパラメータを指定し、Lightsailコンテナサービスにプッシュします。バージョン 2.1.1 とそれ以降のバージョンで、Lightsail コントロールプラグインは機能します。
- Lightsail コントロール (lightsailctl) プラグイン – がローカルマシンにあるコンテナイメージにアクセス AWS CLI できるようにします。

このガイドの次のセクションでは、これらのソフトウェアパッケージをダウンロードする場所と、インストール方法について説明しています。コンテナサービスの詳細については、「コンテナサービス」を参照してください。

目次

- Docker をインストールする
- のインストール AWS CLI
- Lightsail コントローラー プラグインをインストールする
  - Windows に lightsailctl プラグインをインストールする
  - macOS に lightsailctl プラグインをインストールする
  - Linux に lightsailctl プラグインをインストールする

Docker をインストールする

Docker は、Linux コンテナをベースにしている配信されたアプリケーションの構築、実行、テスト、そしてデプロイを可能にするテクノロジーです。Lightsail コンテナサービスで使用することができる独自のコンテナイメージを作成したい場合は、Docker ソフトウェアをインストールして使用する必要があります。詳細については、「Lightsail コンテナサービスでのコンテナイメージの作成」を参照してください。

Docker はさまざまなオペレーティングシステムで使用できます。Ubuntu のような最新の Linux ディストリビューションや、 macOS や Windows でも使用できます。特定のオペレーティングシステムに Docker をインストールする方法の詳細については、Docker インストールガイド を参照してください。

> **Note**
>
> Docker の最新バージョンがインストールされている必要があります。古いバージョンの Docker が、このガイドで後述する AWS CLI および Lightsail Control (lightsailctl) プラグインで動作するとは限りません。

のインストール AWS CLI

AWS CLI は、コマンドラインシェルのコマンドLightsailを使用して、 などの AWS サービスとやり取りできるオープンソースツールです。ローカルマシンで作成されたコンテナイメージをLightsailコンテナサービスにプッシュ AWS CLI するには、 をインストールして使用する必要があります。

AWS CLI は、次のバージョンで使用できます。

- バージョン 2.x — 現在一般的にご利用いただける AWS CLIのリリース。これは の最新バージョン AWS CLI であり、コンテナイメージをLightsailコンテナサービスにプッシュする機能など、すべての最新機能をサポートしています。バージョン 2.1.1 とそれ以降のバージョンで、Lightsail コントロールプラグインは機能します。
- バージョン 1.x – 下位互換性のために AWS CLI 利用可能な の以前のバージョン。このバージョンでは、コンテナイメージを Lightsail コンテナサービスにプッシュする機能がサポートされていません。したがって、代わりに AWS CLI バージョン 2 をインストールする必要があります。

AWS CLI バージョン 2 は、Linux、macOS、および Windows オペレーティングシステムで使用できます。 AWS CLI これらのオペレーティングシステムに をインストールする方法については、「AWS CLI ユーザーガイド」の「バージョン 2 AWS CLIのインストール」を参照してください。

Lightsail コントローラー プラグインをインストールする

Lightsail Control (lightsailctl) プラグインは、 がローカルマシンで作成したコンテナイメージ AWS CLI にアクセスできるようにする軽量アプリケーションです。このプラグインは、コンテナイメージを Lightsail コンテナサービスにプッシュすることで、それらをご自身のサービスにデプロイできるようにします。

システム要件

- 64 ビット対応の Windows、macOS、および Linux オペレーティングシステム。

- AWS CLI lightsailctl プラグインを使用するには、バージョン 2 をローカルマシンにインストールする必要があります。詳細については、このガイドの前のセクションにあった「AWS CLIをインストールする」を参照してください。

最新バージョンの lightsailctl プラグインを使用する

lightsailctl プラグインは、拡張機能で更新されることがあります。lightsailctl プラグインを使用するたびに、最新バージョンを使用していることを確認するチェックが実行されます。新しいバージョンが利用可能であることが判明した場合は、最新バージョンに更新して新しい機能を利用するように求められます。更新バージョンが利用可能になったら、インストールプロセスを繰り返して lightsailctl プラグインの最新バージョンを取得する必要があります。

次の内容では lightsail ctl プラグインのすべてのリリース、ならびに各バージョンに含まれる機能および強化機能が一覧表示されす。

- v1.0.0 (2020 年 11 月 12 日にリリース) – 初回リリースでは、Lightsailコンテナイメージをコンテナサービスにプッシュするための AWS CLI バージョン 2 の機能が追加されました。

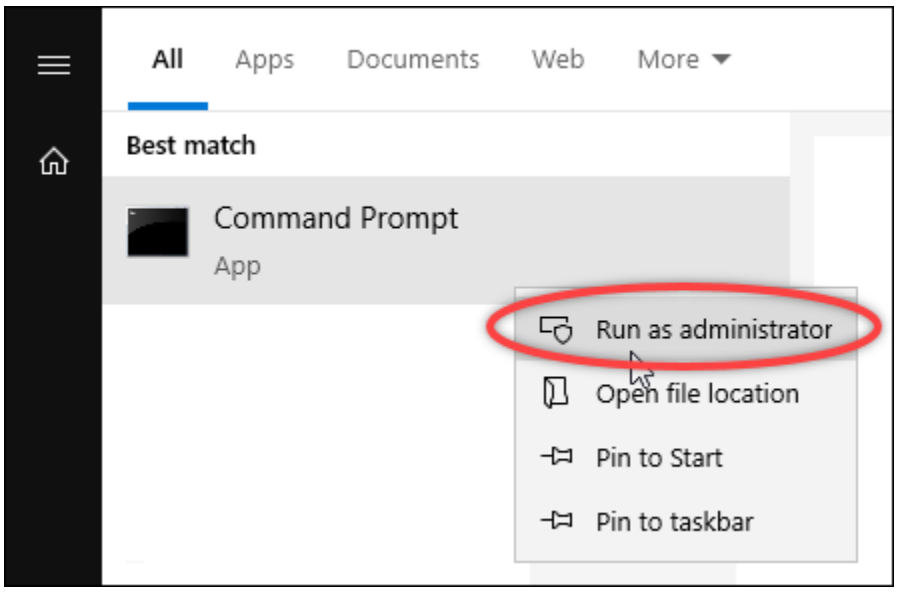
Windows に lightsailctl プラグインをインストールする

Windows に lightsailctl プラグインをインストールするには、次の手順を実行します。

1. 次の URL から実行可能ファイルをダウンロードして、`C:\Temp\lightsailctl\` ディレクトリに保存します。

   ```
   https://s3.us-west-2.amazonaws.com/lightsailctl/latest/windows-amd64/
   lightsailctl.exe
   ```

2. Windows Start ボタンを選択して、`cmd` を検索します。
3. 検索結果から Command Prompt アプリケーションを右クリックし、[Run as administrator] を選択します。



> **Note**
>
> デバイスに変更を加えることを Command Prompt に許可するかの確認プロンプトが表示される場合があります。はい を選択してインストールを続行します。

4. 次のコマンドを入力して、lightsailctl プラグインを保存したC:\Temp\lightsailctl\ディレクトリを指すパス環境変数を設定します。

```
setx PATH "%PATH%;C:\Temp\lightsailctl" /M
```

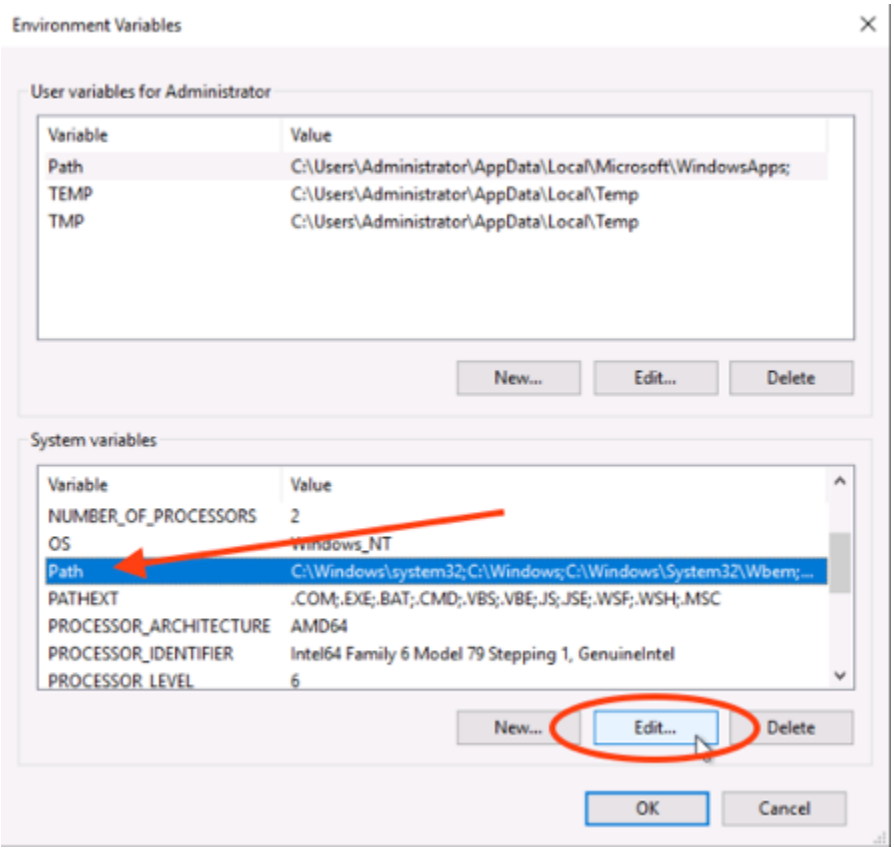
次の例のような結果が表示されます。

```
C:\WINDOWS\system32>setx PATH "%PATH%;C:\Temp\lightsailctl\" /M

SUCCESS: Specified value was saved.
```

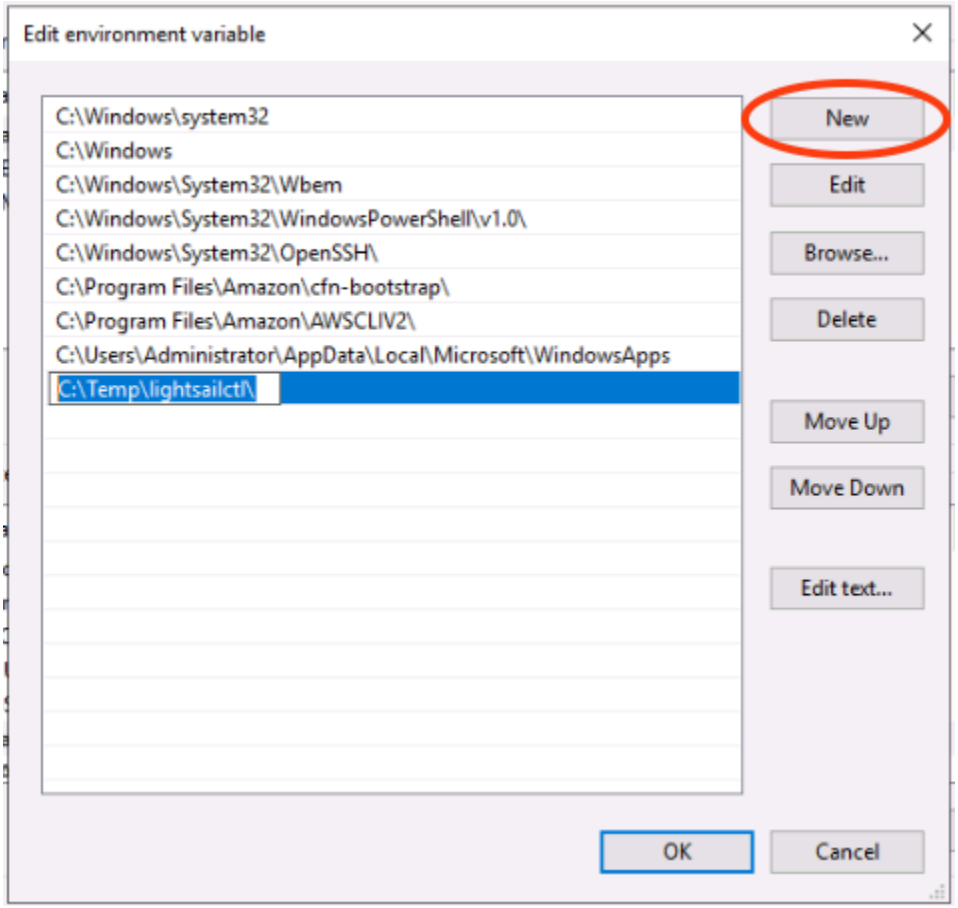
setx コマンドは 1,024 文字を超えると切り捨てられます。PATH に複数の変数がすでに設定されている場合は、以下の手順を使用して PATH 環境変数を手動で設定します。

1. [Start] (スタート) メニューから [Control Panel] (コントロールパネル) を開きます。
2. [System and Security] (システムとセキュリティ) を選択し、[System] (システム) を選択します。
3. [システムの詳細設定] を選択します。

4. [System Properties] (システムのプロパティ) ダイアログボックスで、[Advanced] (詳細設定) タブを開き、[Environment Variables] (環境変数) を選択します。
5. [Environment Variables] (環境変数) ダイアログボックスの [System Variables] (システム変数) ボックスで、[Path] (パス) を選択します。
6. [System Variables] (システム変数) ボックスの下にある [Edit] (編集) ボタンを選択します。



7. [New] (新規) を選択し、次のパスを入力します。`C:\Temp\lightsailctl\`



8. 3 つの連続したダイアログボックスで [OK] を選択し、[System] (システム) ダイアログボックスを閉じます。

これで、 AWS Command Line Interface (AWS CLI) を使用してコンテナイメージをLightsailコンテナサービスにプッシュする準備が整いました。詳しくは、「コンテナイメージをプッシュして管理する」を参照してください。

macOS に lightsailctl プラグインをインストールする

次のいずれかの手順を実行して、macOS に lightsailctl プラグインをダウンロードしてインストールします。

ホームブリューダウンロードとインストール

1. ターミナルウィンドウを開きます。
2. 次のコマンドを入力して、lightsailctl プラグインをダウンロードしてインストールします。

```
brew install aws/tap/lightsailctl
```

> **Note**
>
> Homebrew の詳細については、Homebrew ウェブサイトを参照してください。

手動のダウンロードとインストール

1. ターミナルウィンドウを開きます。
2. 次のコマンドを入力して、lightsailctl プラグインをダウンロードし、bin フォルダにコピーします。

```
curl "https://s3.us-west-2.amazonaws.com/lightsailctl/latest/darwin-amd64/
lightsailctl" -o "/usr/local/bin/lightsailctl"
```

3. 次のコマンドを入力して、実行可能なプラグインを作成します。

```
chmod +x /usr/local/bin/lightsailctl
```

4. 次のコマンドを入力して、プラグインの拡張属性をクリアにします。

```
xattr -c /usr/local/bin/lightsailctl
```

これで、 を使用してコンテナイメージをLightsailコンテナサービスに AWS CLI プッシュする準備が整いました。詳しくは、「コンテナイメージをプッシュして管理する」を参照してください。

Linux に lightsailctl プラグインをインストールする

Linux に Lightsail コンテナサービスプラグインをインストールするには、次の手順を実行します。

1. ターミナルウィンドウを開きます。
2. 次のコマンドを入力して、lightsailctl プラグインをダウンロードします。

   - AMD 64 ビットのアーキテクチャーバージョンのプラグインの場合：

   ```
   curl "https://s3.us-west-2.amazonaws.com/lightsailctl/latest/linux-amd64/
   lightsailctl" -o "/usr/local/bin/lightsailctl"
   ```

   - ARM 64 ビットのアーキテクチャーバージョンのプラグインの場合：

```
curl "https://s3.us-west-2.amazonaws.com/lightsailctl/latest/linux-arm64/
lightsailctl" -o "/usr/local/bin/lightsailctl"
```

3. 次のコマンドを入力して、実行可能なプラグインを作成します。

```
sudo chmod +x /usr/local/bin/lightsailctl
```

これで、 を使用してコンテナイメージをLightsailコンテナサービスに AWS CLI プッシュする準備が整いました。詳しくは、「コンテナイメージをプッシュして管理する」を参照してください。

Lightsail コンテナサービスに対して、Amazon ECR プライベートリポジトリへのアクセスを付与する

Amazon Elastic Container Registry (Amazon ECR) は、 AWS Identity and Access Management (IAM) を使用したリソースベースのアクセス許可を持つプライベートリポジトリをサポートする AWS マネージドコンテナイメージレジストリサービスです。Amazon Lightsail コンテナサービスに対して、Amazon ECR プライベートリポジトリ AWS リージョンへのアクセスを許可することができます。その後、プライベートリポジトリからコンテナサービスにイメージをデプロイすることができます。

Lightsail コンソールまたは AWS Command Line Interface () を使用して、Lightsailコンテナサービスと Amazon ECR プライベートリポジトリへのアクセスを管理できますAWS CLI。ただし、プロセスを簡素化するため Lightsail コンソールの使用をお勧めします。

コンテナサービスの詳細については、「コンテナサービス」を参照してください。Amazon ECR の詳細については、「Amazon ECR ユーザーガイド」を参照してください。

目次

- 必要な許可
- Lightsail コンソールを使用してプライベートリポジトリへのアクセスを管理する
- を使用してプライベートリポジトリへのアクセス AWS CLI を管理する
  - Amazon ECR イメージプラー IAM ロールを有効または無効にする
  - Amazon ECR プライベートリポジトリにポリシーステートメントがあるかどうかを見極める
    - ポリシーステートメントを持たないプライベートリポジトリにポリシーを追加する

- ポリシーステートメントを有するプライベートリポジトリにポリシーを追加する

必要なアクセス許可

Lightsail コンテナサービスの Amazon ECR プライベートリポジトリへのアクセスを管理するユーザーは、IAM で以下のいずれかのアクセス許可ポリシーを持つ必要があります。詳細については、[AWS Identity and Access Management ユーザーガイド]の「IAM ID アクセス許可の追加と削除」を参照してください。

任意の Amazon ECR プライベートリポジトリにアクセス権を付与する

以下のアクセス許可ポリシーは、任意の Amazon ECR プライベートリポジトリへのアクセスを設定する権限をユーザーに付与します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ManageEcrPrivateRepositoriesAccess",
            "Effect": "Allow",
            "Action": [
                "ecr:SetRepositoryPolicy",
                "ecr:DescribeRepositories",
                "ecr:DeleteRepositoryPolicy",
                "ecr:GetRepositoryPolicy"
            ],
            "Resource": "arn:aws:ecr:*:111122223333:repository/*"
        }
    ]
}
```

ポリシーで、*AwsAccountId* を AWS アカウント ID 番号に置き換えます。

特定の Amazon ECR プライベートリポジトリにアクセス権を付与する

以下のアクセス許可ポリシーは、特定の AWS リージョン内の特定の Amazon ECR プライベートリポジトリへのアクセスを設定する権限をユーザーに付与します。

ポリシー内で、次のサンプルテキストを独自のテキストに置き換えます。

- *AwsRegion* — プライベートリポジトリの AWS リージョン コード (例: us-east-1)。Lightsail コンテナサービスは、アクセスするプライベートリポジトリ AWS リージョン と同じ にある必要があります。
- *AwsAccountId* — AWS アカウント ID 番号。
- *RepositoryName* — アクセスを管理するプライベートリポジトリの名前。

以下は、アクセス許可ポリシーに例の値を入力した一例です。

JSON

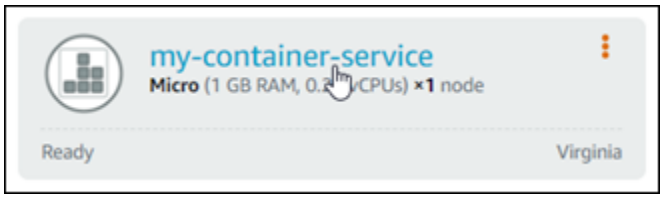
```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ManageEcrPrivateRepositoriesAccess",
            "Effect": "Allow",
            "Action": [
                "ecr:SetRepositoryPolicy",
                "ecr:DescribeRepositories",
                "ecr:DeleteRepositoryPolicy",
                "ecr:GetRepositoryPolicy"
            ],
            "Resource": "arn:aws:ecr:us-east-1:111122223333:repository/my-
private-repo"
        }
    ]
}
```

Lightsail コンソールを使用してプライベートリポジトリへのアクセスを管理する

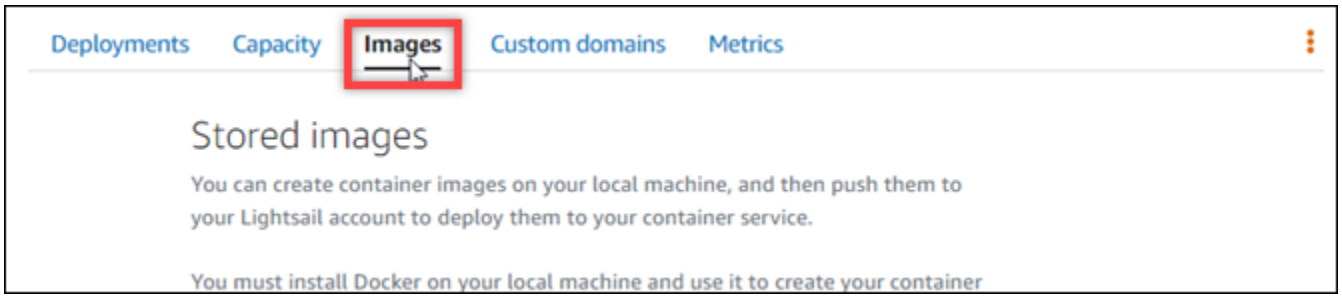
Lightsail コンソールを使用して Lightsail コンテナサービスの Amazon ECR プライベートリポジトリへのアクセスを管理するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。

3. Amazon ECR プライベートリポジトリへのアクセスを設定したいコンテナサービスの名前を選択します。

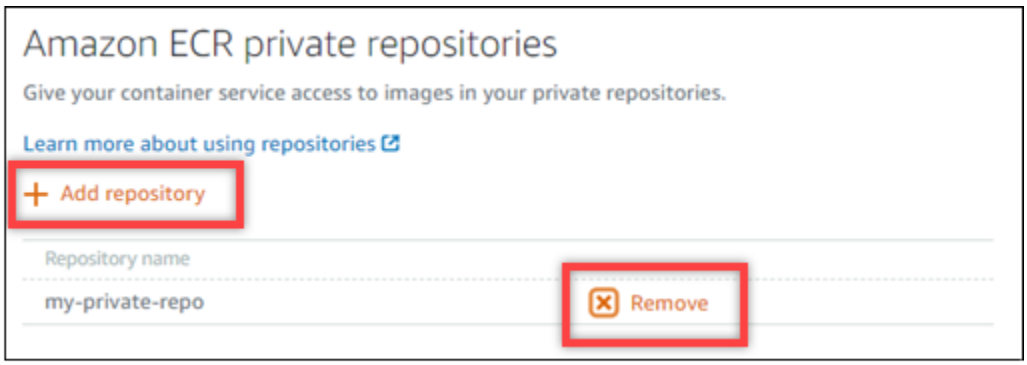


4. [Images] (イメージ) タブを選択します。

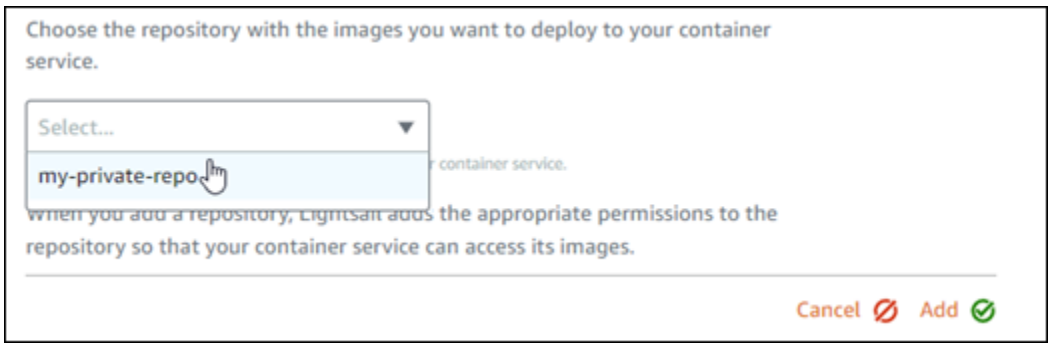


5. [リポジトリの追加] を選択すると、コンテナサービスの Amazon ECR プライベートリポジトリへのアクセス権が付与されます。

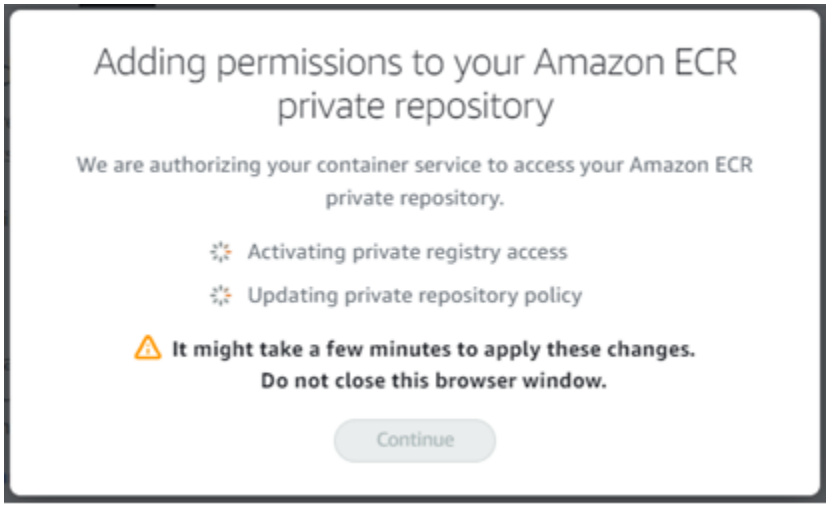
> **Note**
>
> [削除] を選択すると、以前に追加した Amazon ECR プライベートリポジトリからコンテナサービスのアクセスが削除されます。



6. 表示されるドロップダウンから、アクセスするプライベートリポジトリを選択し、[Add] (追加) を選択します。

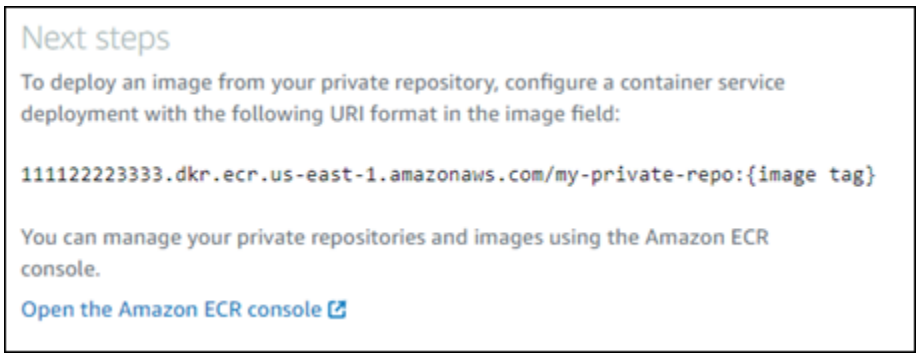


Lightsail は、コンテナサービスの Amazon ECR イメージプーラー IAM ロールを有効にするために数分かかります。この IAM ロールにはプリンシパル Amazon リソースネーム (ARN)が含まれています。その後 Lightsail は、選択した Amazon ECR プライベートリポジトリのアクセス許可ポリシーに、IAM ロールのプリンシパル ARN を自動的に追加します。　これにより、コンテナサービスはプライベートリポジトリとそのイメージにアクセスできるようになります。プロセスが完了し、[Continue] (続行) を選択できることを示すモーダルが表示されるまで、ブラウザウィンドウは閉じないでください。



7. アクティベーションが完了したら、[Continue] (続行) を選択します。

選択した Amazon ECR プライベートリポジトリが追加されると、このページの [Amazon ECR プライベートリポジトリ] セクションに表示されます。このページには、プライベートリポジトリから Lightsail コンテナサービスにイメージをデプロイする方法の手順が含まれています。リポジトリにあるイメージを使用するには、コンテナサービスのデプロイの作成時に、ページに [Image] (イメージ) の値として表示される URI 形式を指定します。指定する URI では、*#####*の例を、デプロイしたいイメージのタグに置き換えます。詳細については、「コンテナサービスのデプロイの作成と管理」を参照してください。



を使用してプライベートリポジトリへのアクセス AWS CLI を管理する

AWS Command Line Interface (AWS CLI) を使用して Amazon ECR プライベートリポジトリへの Lightsailコンテナサービスのアクセスを管理するには、次のステップが必要です。

> **Important**
>
> Lightsail コンテナサービスの Amazon ECR プライベートリポジトリへのアクセスを管理する際には、プロセスが簡潔になるため Lightsail コンソールを使用することを推奨します。詳細については、本ガイドの前半の「Lightsail コンソールを使用してプライベートリポジトリへのアクセスを管理する」を参照してください。

1. Amazon ECR イメージプーラー IAM ロールをアクティブ化または非アクティブ化する — AWS CLI `update-container-service`の コマンドLightsailを使用して、Amazon ECR イメージプーラー IAM ロールをアクティブ化または非アクティブ化します。有効にすると、Amazon ECR イメージプーラー IAM ロールにプリンシパル Amazon リソースネーム (ARN) が作成されます。詳細については、このガイドの「Amazon ECR イメージプーラー IAM ロールを有効または無効にする」セクションを参照してください。
2. Amazon ECR プライベートリポジトリにポリシーステートメントがあるかどうかを判断する — Amazon ECR イメージプーラー IAM ロールを有効にした後、コンテナサービスでアクセスしたい Amazon ECR プライベートリポジトリに既存のポリシーステートメントがあるかどうかを判断する必要があります。詳細については、このガイドの後半にある「Amazon ECR プライベートリポジトリにポリシーステートメントがあるかどうかを判断する」を参照してください。

   リポジトリに既存のポリシーステートメントがあるかどうかに応じて、次のいずれかの方法を使用して IAM ロールプリンシパル ARN をリポジトリに追加します。

a. ポリシーステートメントを持たないプライベートリポジトリにポリシーを追加する — Amazon ECR の `set-repository-policy` コマンドを使用して AWS CLI 、コンテナサービスの Amazon ECR イメージプーラーロールプリンシパル ARN を既存のポリシーを持つプライベートリポジトリに追加します。詳細については、本ガイドの後半にある「ポリシーステートメントを持たないプライベートリポジトリにポリシーを追加する」を参照してください。

b. ポリシーステートメントを持つプライベートリポジトリにポリシーを追加する — Amazon ECR の `set-repository-policy` コマンドを使用して AWS CLI 、コンテナサービスの Amazon ECR イメージプーラーロールを、既存のポリシーを持たないプライベートリポジトリに追加します。詳細については、本ガイドの後半にある「ポリシーステートメントを有するプライベートリポジトリにポリシーを追加する」を参照してください。

Amazon ECR イメージプラー IAM ロールを有効または無効にする

以下の手順を完了し、Lightsail コンテナサービスの Amazon ECR イメージプーラー IAM ロールを有効または無効にします。の `update-container-service` コマンドを使用して、Amazon ECR イメージプーラー IAM ロールを AWS CLI アクティブ化または非アクティブ化できますLightsail。詳細については、「AWS CLI コマンドリファレンス」の「update-container-service」を参照してください。

> **Note**
>
> この手順を続行するLightsail前に、 をインストール AWS CLI して 用に設定する必要があります。詳細については、「 Lightsail で使用するために AWS CLI を設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 次のコマンドを入力して、コンテナサービスを更新し、Amazon ECR イメージプーラー IAM ロールを有効または無効にします。

```
aws lightsail update-container-service --service-name ContainerServiceName --
private-registry-access ecrImagePullerRole={isActive=RoleActivationState} --
region AwsRegionCode
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *ContainerServiceName* — Amazon ECR イメージプーラー IAM ロールを有効または無効にするコンテナサービスの名前です。
- *RoleActivationState* — Amazon ECR イメージプーラー IAM ロールのアクティブ化状態です。ロールを有効にするには `true` を指定し、無効にするには `false` を指定します。
- *AwsRegionCode* — コンテナサービスの AWS リージョン コード (例えば、`us-east-1`)。

例:

- Amazon ECR イメージプーラー IAM ロールを有効にするには:

```
aws lightsail update-container-service --service-name my-container-service --
private-registry-access ecrImagePullerRole={isActive=true} --region us-east-1
```

- Amazon ECR イメージプーラー IAM ロールを無効にするには:

```
aws lightsail update-container-service --service-name my-container-service --
private-registry-access ecrImagePullerRole={isActive=false} --region us-east-1
```

3. オプション:

- Amazon ECR イメージプーラーロールを有効にした場合 — 前のレスポンスを取得後は、少なくとも 30 秒待機します。その後、次のステップに進んで、コンテナサービスの Amazon ECR イメージプーラー IAM ロールのプリンシパル ARN を取得します。
- Amazon ECR イメージプーラーロールを無効にした場合 — 以前に Amazon ECR イメージプーラー IAM ロールのプリンシパル ARN を Amazon ECR プライベートリポジトリのアクセス許可ポリシーに追加している場合、リポジトリからこのアクセス許可ポリシーを削除する必要があります。詳細については、「Amazon ECR ユーザーガイド」の「プライベートリポジトリポリシーステートメントを削除する」を参照してください。

4. 次のコマンドを入力して、コンテナサービスの Amazon ECR イメージプーラー IAM ロールのプリンシパル ARN を取得します。

```
aws lightsail get-container-services --service-name ContainerServiceName --
region AwsRegionCode
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *ContainerServiceName* — Amazon ECR イメージプーラー IAM ロールのプリンシパル ARN を取得するためのコンテナサービスの名前です。
- *AwsRegionCode* — コンテナサービスの AWS リージョン コード (例えば、us-east-1)。

例:

```
aws lightsail get-container-services --service-name my-container-service --
region us-east-1
```

レスポンスに ECR イメージプーラー IAM ロールのプリンシパル ARN がないか探します。ロールがリストにある場合は、コピーして書き留めます。本ガイドの次のセクションで必要になります。次に、コンテナサービスでアクセスしたい Amazon ECR プライベートリポジトリに、既存のポリシーステートメントがあるかどうかを見極める必要があります。本ガイドの「Amazon ECR プライベートリポジトリにポリシーステートメントがあるかどうかを見極める」のセクションに進んでください。

Amazon ECR プライベートリポジトリにポリシーステートメントがあるかどうかを見極める

以下の手順で、Amazon ECR プライベートリポジトリにポリシーステートメントがあるかどうかを見極めます。Amazon ECR には コマンドを使用できます AWS CLI get-repository-policy。詳細については、「AWS CLI コマンドリファレンス」の「update-container-service」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、Amazon ECR 用に設定する必要があります。詳細については、「Amazon ECR ユーザーガイド」の「Amazon ECR でのセットアップ」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 以下のコマンドを入力して、特定のプライベートリポジトリのポリシーステートメントを取得します。

```
aws ecr get-repository-policy --repository-name RepositoryName --
region AwsRegionCode
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *RepositoryName* — Lightsail コンテナサービスへのアクセスを設定するプライベートリポジトリの名前です。
- *AwsRegion* — プライベートリポジトリの AWS リージョン コード (例: us-east-1) です。

例:

```
aws ecr get-repository-policy --repository-name my-private-repo --region us-east-1
```

以下のレスポンスのいずれかが表示されます。

- RepositoryPolicyNotFoundException — プライベートリポジトリにポリシーステートメントがありません。リポジトリにポリシーステートメントがない場合は、本ガイドの後半にある「ポリシーステートメントを持たないプライベートリポジトリにポリシーを追加する」のセクションにある手順に従います。

```
C:\>aws ecr get-repository-policy --repository-name my-private-repo

An error occurred (RepositoryPolicyNotFoundException) when calling the GetRepositoryPolicy operation: Repository policy
does not exist for the repository with name 'my-private-repo' in the registry with id '[redacted]'
```

- リポジトリポリシーが見つかった場合 - プライベートリポジトリにはポリシーステートメントがあり、その内容がリクエストに対するレスポンスに表示されます。リポジトリにポリシーステートメントがある場合は、既存のポリシーをコピーして、本ガイドの後半にある「ポリシーステートメントを有するプライベートリポジトリにポリシーを追加する」のセクションにある手順に従います。

```
C:\>aws ecr get-repository-policy --repository-name my-private-repo
{
    "registryId": "[redacted]",
    "repositoryName": "my-private-repo",
    "policyText": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement\" : [ {\n    \"Sid\" : \"AllowUserPushPull\",\n
\"Effect\" : \"Allow\",\n    \"Principal\" : {\n      \"AWS\" : \"arn:aws:iam::[redacted]:user/example-user\"\n    },\
n    \"Action\" : [ \"ecr:BatchGetImage\", \"ecr:BatchCheckLayerAvailability\", \"ecr:CompleteLayerUpload\", \"ecr:GetDo
wnloadUrlForLayer\", \"ecr:InitiateLayerUpload\", \"ecr:PutImage\", \"ecr:UploadLayerPart\" ]\n  } ]\n}"
}
```

ポリシーステートメントを持たないプライベートリポジトリにポリシーを追加する

以下の手順に従って、ポリシーステートメントを持たない Amazon ECR プライベートリポジトリにポリシーを追加します。追加するポリシーには、Lightsail コンテナサービスの Amazon ECR イメー

ジプーラー IAM ロール プリンシパル ARN を含める必要があります。これにより、コンテナサービスにアクセス権が付与され、プライベートリポジトリからイメージをデプロイできるようになります。

> **Important**
>
> Lightsail コンソールを使用してアクセスを設定すると、Lightsail は Amazon ECR イメージプーラーロールを Amazon ECR プライベートリポジトリに自動的に追加します。この場合、このセクションの手順を使用して、プライベートリポジトリに Amazon ECR イメージプーラーロールを手動で追加する必要はありません。詳細については、本ガイドの前半の「Lightsail コンソールを使用してプライベートリポジトリへのアクセスを管理する」を参照してください。

AWS CLIを使用して、プライベートリポジトリにポリシーを追加できます。これを行うには、ポリシーを含む JSON ファイルを作成し、Amazon ECR の `set-repository-policy` コマンドでそのファイルを参照します。詳細については、「AWS CLI コマンドリファレンス」の「set-repository-policy」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、Amazon ECR 用に設定する必要があります。詳細については、「Amazon ECR ユーザーガイド」の「Amazon ECR でのセットアップ」を参照してください。

1. テキストエディタを開き、次のポリシーステートメントを新しいテキストファイルに貼り付けます。

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
  {
      "Sid": "AllowLightsailPull-ecr-private-repo-demo",
      "Effect": "Allow",
      "Principal": {
```

```
        "AWS": "IamRolePrincipalArn"
      },
      "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer"
      ]
    }
  ]
}
```

テキストファイル内では、*IamRolePrincipalArn* を、本ガイドの前半で取得したコンテナサービスの Amazon ECR イメージプーラー IAM ロールのプリンシパル ARN に置き換えてください。

2. ファイルを ecr-policy.json という名前で、コンピュータ上のアクセス可能な場所 (例: Windows では C:\Temp\ecr-policy.json、macOS や Linux では /tmp/ecr-policy.json) に保存します。
3. 作成された ecr-policy.json ファイルのファイルパスの場所を書き留めます。この手順の後半に出てくるコマンドで、これを指定します。
4. ターミナルまたはコマンドプロントウィンドウを開きます。
5. 以下のコマンドを入力して、コンテナサービスを使ってアクセスしたいプライベートリポジトリのポリシーステートメントを設定します。

```
aws ecr set-repository-policy --repository-name RepositoryName --policy-text
 file://path/to/ecr-policy.json --region AwsRegionCode
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *RepositoryName* — ポリシーを追加するプライベートリポジトリの名前です。
- *path/to/* — 本ガイドの前半部分で作成した、コンピュータ上の ecr-policy.json ファイルへのパスです。
- *AwsRegion* — プライベートリポジトリの AWS リージョン コード (例: us-east-1) です。

例:

- Windows の場合:

```
aws ecr set-repository-policy --repository-name my-private-repo --policy-text
 file://C:\Temp\ecr-policy.json --region us-east-1
```

- macOS または Linux の場合:

```
aws ecr set-repository-policy --repository-name my-private-repo --policy-text
 file:///tmp/ecr-policy.json --region us-east-1
```

これで、コンテナサービスはプライベートリポジトリとそのイメージにアクセスできるようになります。リポジトリにあるイメージを使用するには、コンテナサービスのデプロイのイメージ値として以下の URI を指定します。URI 内の##例を、デプロイしたいイメージのタグに置き換えます。詳細については、「コンテナサービスのデプロイの作成と管理」を参照してください。

```
AwsAccountId.dkr.ecr.AwsRegionCode.amazonaws.com/RepositoryName:ImageTag
```

URI 内の次のサンプルテキストを自身が使用するテキストに置き換えます。

- *AwsAccountId* — AWS アカウント ID ナンバー。
- *AwsRegion* — プライベートリポジトリの AWS リージョン コード (例: us-east-1) です。

- *RepositoryName* — コンテナイメージをデプロイするプライベートリポジトリの名前です。
- *ImageTag* — コンテナサービスにデプロイするプライベートリポジトリのコンテナイメージのタグです。

例:

```
111122223333.dkr.ecr.us-east-1.amazonaws.com/my-private-repo:myappimage
```

ポリシーステートメントを有するプライベートリポジトリにポリシーを追加する

以下の手順に従って、ポリシーステートメントを有する Amazon ECR プライベートリポジトリにポリシーを追加します。追加するポリシーには、既存のポリシーと、Lightsail コンテナサービスの Amazon ECR イメージプーラー IAM ロール プリンシパル ARN が含まれる新しいポリシーを含める必要があります。これにより、プライベートリポジトリ上にある既存のアクセス許可が維持されなが

ら、同時にプライベートリポジトリからイメージをデプロイするためのコンテナサービスへのアクセス権も付与されます。

> **Important**
>
> Lightsail コンソールを使用してアクセスを設定すると、Lightsail は Amazon ECR イメージプーラーロールを Amazon ECR プライベートリポジトリに自動的に追加します。この場合、このセクションの手順を使用して、プライベートリポジトリに Amazon ECR イメージプーラーロールを手動で追加する必要はありません。詳細については、本ガイドの前半の「Lightsail コンソールを使用してプライベートリポジトリへのアクセスを管理する」を参照してください。

AWS CLIを使用して、プライベートリポジトリにポリシーを追加できます。これを行うには、既存のポリシーと新しいポリシーが含まれる JSON ファイルを作成します。その後、そのファイルを Amazon ECR の `set-repository-policy` コマンドで参照します。詳細については、「AWS CLI コマンドリファレンス」の「set-repository-policy」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、Amazon ECR 用に設定する必要があります。詳細については、「Amazon ECR ユーザーガイド」の「Amazon ECR でのセットアップ」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 以下のコマンドを入力して、特定のプライベートリポジトリのポリシーステートメントを取得します。

   ```
   aws ecr get-repository-policy --repository-name RepositoryName --
   region AwsRegionCode
   ```

   コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

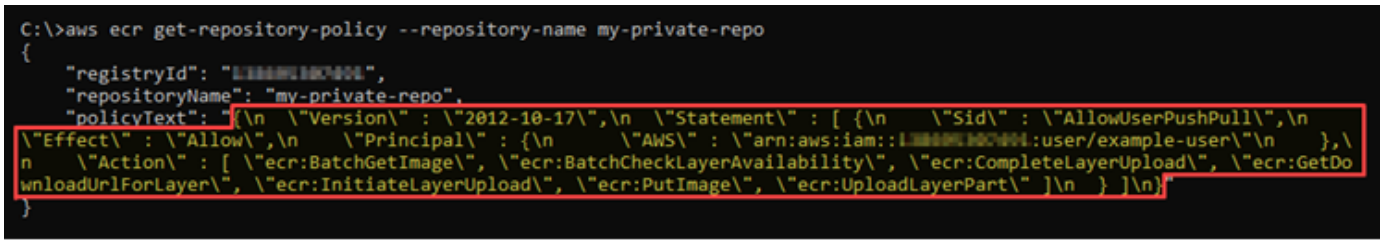
   - *RepositoryName* — Lightsail コンテナサービスへのアクセスを設定するプライベートリポジトリの名前です。
   - *AwsRegion* — プライベートリポジトリの AWS リージョン コード (例: `us-east-1`) です。

例:

```
aws ecr get-repository-policy --repository-name my-private-repo --region us-east-1
```

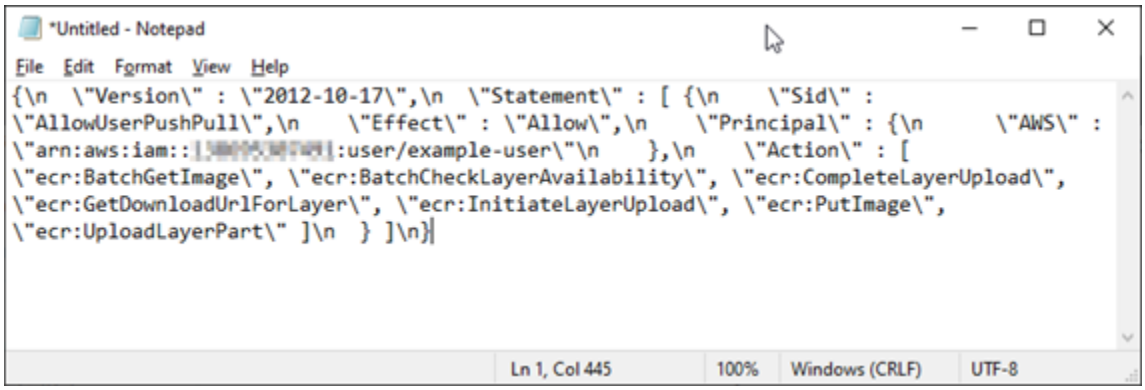
3. レスポンスに、既存のポリシーをコピーし、次のステップに進みます。

   次の例でハイライトされている部分のように、二重引用符で囲まれた policyText の内容のみをコピーする必要があります。



4. テキストエディタを開き、前の手順でコピーしたプライベートリポジトリの既存のポリシーを貼り付けます。

   結果は次の例のようになります。



5. 貼り付けたテキスト内の \n を改行に置き換え、残りの \ は削除します。

   結果は次の例のようになります。

6. テキストファイルの末尾に、次のポリシーステートメントを貼り付けます。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowLightsailPull-ecr-private-repo-demo",
            "Effect": "Allow",
            "Principal": {
                "AWS": "IamRolePrincipalArn"
            },
            "Action": [
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ]
        }
    ]
}
```

7. テキストファイル内では、*IamRolePrincipalArn* を、本ガイドの前半で取得したコンテナサービスの Amazon ECR イメージプーラー IAM ロールのプリンシパル ARN に置き換えてください。

   結果は次の例のようになります。

8. ファイルを `ecr-policy.json` という名前で、コンピュータ上のアクセス可能な場所 (例: Windows では `C:\Temp\ecr-policy.json`、macOS や Linux では `/tmp/ecr-policy.json`) に保存します。
9. `ecr-policy.json` ファイルのファイルパスの場所を書き留めます。この手順の後半に出てくるコマンドで、これを指定します。

10. ターミナルまたはコマンドプロントウィンドウを開きます。
11. 以下のコマンドを入力して、コンテナサービスを使ってアクセスしたいプライベートリポジトリのポリシーステートメントを設定します。

```
aws ecr set-repository-policy --repository-name RepositoryName --policy-text
 file://path/to/ecr-policy.json --region AwsRegionCode
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *RepositoryName* — ポリシーを追加するプライベートリポジトリの名前です。
- *path/to/* — 本ガイドの前半部分で作成した、コンピュータ上の ecr-policy.json ファイルへのパスです。
- *AwsRegion* — プライベートリポジトリの AWS リージョン コード (例: us-east-1) です。

例:

- Windows の場合:

```
aws ecr set-repository-policy --repository-name my-private-repo --policy-text
 file://C:\Temp\ecr-policy.json --region us-east-1
```

- macOS または Linux の場合:

```
aws ecr set-repository-policy --repository-name my-private-repo --policy-text
 file:///tmp/ecr-policy.json --region us-east-1
```

次の例に示すようなレスポンスが表示されます。

```
C:\>aws ecr set-repository-policy --repository-name my-private-repo --policy-text file://C:\Temp\ecr-policy.json --regio
n us-west-2
{
    "registryId": "[redacted]",
    "repositoryName": "my-private-repo",
    "policyText": "{\n  \"Version\" : \"2012-10-17\",\n  \"Statement\" : [ {\n    \"Sid\" : \"AllowLightsailPull-my-cont
ainer-service\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" : {\n      \"AWS\" : \"arn:aws:iam::[redacted]:role/a
mazon/lightsail/[redacted]/containers/my-container-service/private-repo-access/[redacted]\"\n    },\n    \"Action\" : [ \"ecr:BatchGetImage\", \"ecr:GetDownloadUrlForLayer\" ]\n  }, {\n    \"Sid\" :
\"AllowUserPushPull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" : {\n      \"AWS\" : \"arn:aws:iam::[redacted]:
user/example-user\"\n    },\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage\", \"ecr:Comple
teLayerUpload\", \"ecr:GetDownloadUrlForLayer\", \"ecr:InitiateLayerUpload\", \"ecr:PutImage\", \"ecr:UploadLayerPart\"
]\n  } ]\n}"
}
```

get-repository-policy コマンドをもう一度実行すると、プライベートリポジトリに新しく追加されたポリシーステートメントが表示されます。これで、コンテナサービスはプライベートリポジトリとそのイメージにアクセスできるようになります。リポジトリにあるイメージを使用するには、コンテナサービスのデプロイのイメージ値として以下の URI を指定します。URI 内の##例を、デプロイしたいイメージのタグに置き換えます。詳細については、「コンテナサービスのデプロイの作成と管理」を参照してください。

```
AwsAccountId.dkr.ecr.AwsRegionCode.amazonaws.com/RepositoryName:ImageTag
```

URI 内の次のサンプルテキストを自身が使用するテキストに置き換えます。

- *AwsAccountId* — AWS アカウント ID ナンバー。
- *AwsRegion* — プライベートリポジトリの AWS リージョン コード (例: us-east-1) です。
- *RepositoryName* — コンテナイメージをデプロイするプライベートリポジトリの名前です。
- *ImageTag* — コンテナサービスにデプロイするプライベートリポジトリのコンテナイメージのタグです。

例:

```
111122223333.dkr.ecr.us-east-1.amazonaws.com/my-private-repo:myappimage
```

Lightsail でコンテナサービスのデプロイを作成して管理する

Amazon Lightsail コンテナサービスでコンテナを起動する準備ができたら、デプロイを作成します。デプロイは、サービスに起動させたいコンテナの仕様セットです。コンテナサービスは、一度に 1 つのデプロイを実行することが可能で、デプロイは最大 10 個のコンテナエントリを持つことができます。デプロイは、コンテナサービスと同時に作成でき、あるいはサービスの起動と実行後にも作成できます。

> **Note**
>
> 新しいデプロイを作成すると、コンテナサービスの既存の使用率メトリクスが消え、新しい現在のデプロイのメトリクスだけが表示されます。

コンテナサービスの詳細については、「Amazon Lightsail のコンテナサービス」を参照してください。

目次

- 前提条件
- デプロイパラメータ
  - コンテナエントリーパラメータ
  - パブリックエンドポイントパラメータ
- コンテナ間の通信
- コンテナログ
- デプロイバージョン
- デプロイのステータス
- デプロイエラー
- 現在のコンテナサービスのデプロイの表示
- コンテナサービスのデプロイを作成または変更

前提条件

コンテナサービスにデプロイの作成を開始する前に、前提条件として以下を完了します。

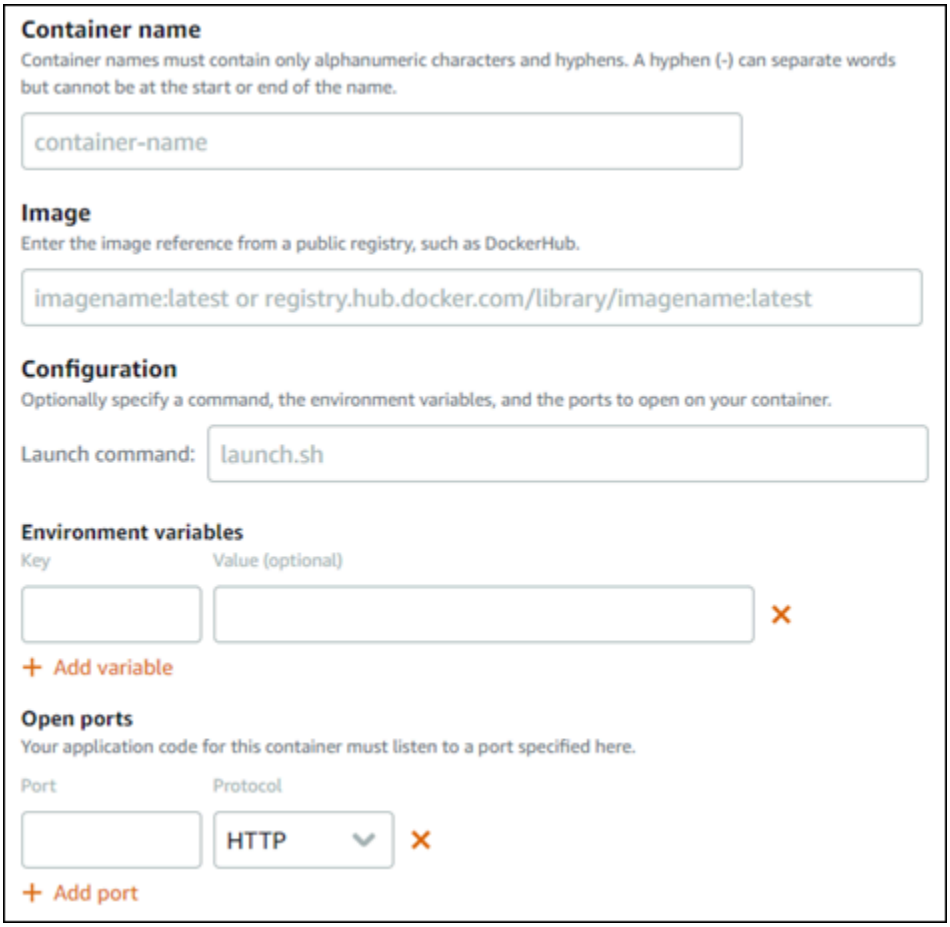
- Lightsail アカウントにコンテナサービスを作成する。詳細については、「Amazon Lightsail コンテナサービスの作成」を参照してください。
- コンテナサービスでコンテナを起動する際に使用するコンテナイメージを特定します。
  - Amazon ECR Public Gallery などのパブリックレジストリでコンテナイメージを検索します。詳細については、「Amazon ECR Public ユーザーガイド」の「Amazon ECR Public Gallery」を参照してください。
  - ローカルマシンでコンテナイメージを作成し、Lightsail コンテナサービスにプッシュします。詳細については、以下のガイドを参照してください。
    - Amazon Lightsail コンテナサービスのコンテナイメージを管理するためのソフトウェアのインストール
    - コンテナサービスのイメージを作成する
    - コンテナイメージをプッシュして管理する

デプロイパラメータ

このセクションでは、コンテナエントリと、デプロイのパブリックエンドポイントに指定できるパラメータについて説明します。

コンテナエントリパラメータ

デプロイには最大 10 個のコンテナエントリを追加できます。各コンテナエントリには、指定できる以下のパラメータがあります。



- コンテナ名 – コンテナ名を入力します。デプロイ内のすべてのコンテナは一意の名前を持つ必要があり、英数字とハイフンのみが使用可能です。ハイフンは単語を区別するために使用することができますが、名前の先頭または末尾には使用できません。
- ソースイメージ – コンテナのソースコンテナイメージを指定します。以下のソースからコンテナイメージを指定することができます。
  - Amazon ECR Public Gallery、その他のパブリックコンテナイメージレジストリなどのパブリックレジストリ。

Amazon ECR Public の詳細については、「Amazon ECR Public ユーザーガイド」の「Amazon Elastic Container Registry Public とは?」を参照してください。

- ローカルマシンからコンテナサービスにプッシュされたイメージ。保存されたイメージを指定するには、[保存されたイメージを選択] を選択し、使用するイメージを選択します。

  ローカルマシンでコンテナイメージを作成する場合は、コンテナサービスにプッシュして、デプロイの作成時に使用できます。詳細については、Amazon Lightsail コンテナサービスのコンテナイメージの作成およびAmazon Lightsail コンテナサービスでコンテナイメージをプッシュして管理するを参照してください。

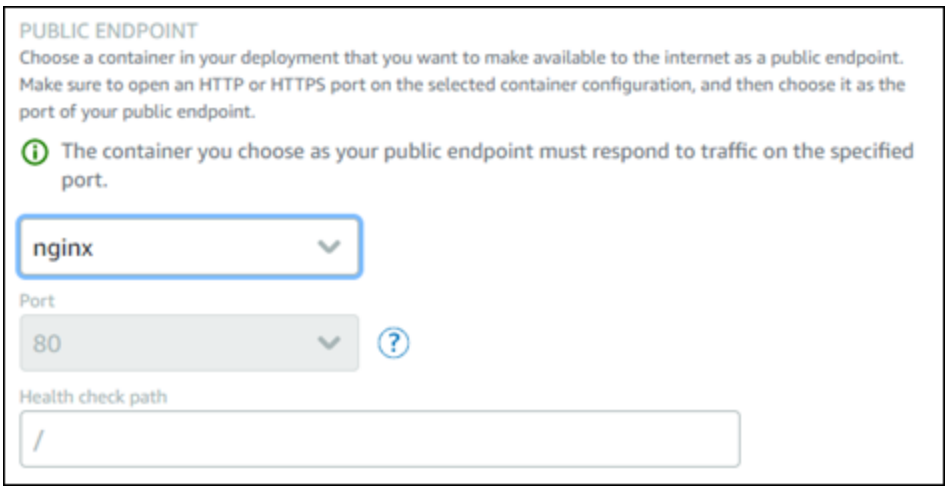
- 起動コマンド – シェルスクリプト、または コンテナの作成時にコンテナを設定する bash スクリプトを実行するための起動コマンドを指定します。起動コマンドでは、ソフトウェアの追加、ソフトウェアの更新、あるいはコンテナの設定などを他の方法で行うことができます。
- 環境可変 – 環境可変を指定します。環境可変は、コンテナによって実行されるアプリケーションまたはスクリプトの動的設定を提供するキーバリューパラメーターです。
- オープンポート – コンテナで開くポートとプロトコルを指定します。HTTP、HTTPS、TCP、および UDP 経由でポートが開くように指定できます。コンテナサービスのパブリックエンドポイントとして使用する予定のコンテナの HTTP ポートまたは HTTPS ポートを開く必要があります。詳細については、このガイドの以下のセクションを参照してください。

パブリックエンドポイントパラメータ

コンテナサービスのパブリックエンドポイントとして機能するデプロイ内のコンテナエントリを指定できます。パブリックエンドポイントコンテナ上のアプリケーションは、コンテナサービスのランダムに生成されたデフォルトドメインを介して、インターネット上でパブリックにアクセスできます。デフォルトのドメインは `https://<ServiceName>.<RandomGUID>.<AWSRegion>.cs.amazonlightsail.com` という形式になっています。*<ServiceName>* はコンテナサービスの名前、*<RandomGUID>* は、Lightsail アカウントの AWS リージョン内のコンテナサービスに対してランダムに生成されるグローバルに一意な識別子、*<AWSRegion>* は、コンテナサービスが作成された AWS リージョンです。Lightsail のパブリックエンドポイントコンテナサービスは HTTPS のみをサポートし、TCP または UDP traffic はサポートされていません。サービスのパブリックエンドポイントにできるコンテナは 1 つだけです。したがって、アプリケーションのフロントエンドをホストしているコンテナをパブリックエンドポイントとして選択し、残りのコンテナは内部的にアクセス可能であることを確認してください。

> **Note**
>
> コンテナサービスでは、独自のカスタムドメイン名を使用できます。詳細については、Amazon Lightsail コンテナサービスでのカスタムドメインの有効化と管理 を参照してください。

デプロイのパブリックエンドポイントとコンテナサービスには、以下のパラメータを指定できます。



- エンドポイントコンテナ – コンテナサービスのパブリックエンドポイントとして機能するデプロイ内のコンテナ名を選択します。デプロイで HTTP ポートまたは HTTPS ポートが開いているコンテナのみがドロップダウンメニューに表示されます。
- ポート – パブリックエンドポイントに使用する HTTP ポートまたは HTTPS ポートを選択します。選択したコンテナで開かれている HTTP ポートと HTTPS ポートのみがドロップダウンメニューに表示されます。選択したコンテナが最初に起動したときに HTTPS 接続をサポートするように設定されていない場合は、HTTP ポートを選択します。

> **Note**
>
> パブリックエンドポイントポートとして HTTP ポートを選択した場合でも、コンテナサービスのデフォルトドメインはデフォルトで HTTPS が使用されます。これは、コンテナサービスのロードバランサーがデフォルトで HTTPS に設定されていますが、HTTP を使用してコンテナとの接続を確立するためです。
> コンテナサービスのロードバランサーは HTTP を使用してコンテナに接続しますが、HTTPS を使用してユーザーにコンテンツを提供します。

- ヘルスチェックパス – コンテナサービスのロードバランサーが定期的にチェックして正常であることを確認するために選択された、パブリックエンドポイントコンテナのパスを指定します。
- ヘルスチェックの詳細設定 - 選択したパブリックエンドポイントコンテナに対して、次のヘルスチェック設定を設定できます。
  - ヘルスチェックのタイムアウト (秒) - ヘルスチェックのレスポンスを待つ時間 (秒単位)。この間にレスポンスが受信されない場合、ヘルスチェックは失敗します。2～60 秒を指定できます。
  - ヘルスチェックの間隔 (秒)-コンテナのヘルスチェックのおおよその間隔 (秒単位)。5～300 秒を指定できます。
  - ヘルスチェックの成功コード-コンテナからの正常なレスポンスを確認するために使用する HTTP コード。200～499 の値を指定できます。複数の値 (例: 200,202) または値の範囲 (例: 200-299) を指定できます。
  - ヘルスチェックの健全性しきい値 - コンテナをヘルス状態に移行するために必要な連続したヘルスチェックの成功数。
  - ヘルスチェックの異常しきい値 - コンテナを異常状態に移行するために必要な連続したヘルスチェックの成功数。

プライベートドメイン

また、すべてのコンテナサービスは、プライベートドメインを保持していて、*<ServiceName>*.service.local にフォーマットされており*<ServiceName>*はコンテナサービスの名前です。プライベートドメインを使用して、サービスと同じ AWS リージョンにある別の Lightsail リソースからコンテナサービスにアクセスします。プライベートドメインは、サービスのデプロイメントでパブリックエンドポイントを指定しない場合、コンテナサービスにアクセスする唯一の方法です。パブリックエンドポイントを指定しなくても、コンテナサービスに対してデフォルトのドメインが生成されますが、観覧しようとすると、`404 No Such Service` エラーメッセージが表示されます。

コンテナサービスのプライベートドメインを使用して特定のコンテナにアクセスするには、接続要求を受け入れるコンテナのオープンポートを指定する必要があります。これを実行するには、リクエストのドメインを*<ServiceName>*.service.local:*<PortNumber>* にフォーマットし、この中で*<ServiceName>*はコンテナサービスの名前、*<PortNumber>*は、接続したいコンテナのオープンポートです。例えば、コンテナサービスにデプロイcontainer-service-1 を作成し、Redis コンテナを指定してポート 6379 が開いている場合は、リクエストのドメインを*container-service-1*.service.local:*6379* にフォーマットします。

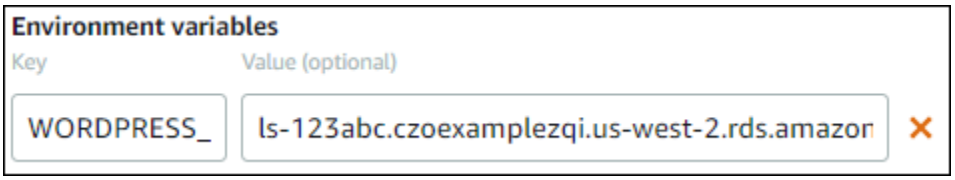
コンテナ間の通信

環境変数を使用すると、同じコンテナサービス内のコンテナ間、異なるコンテナサービス内のコンテナ、またはコンテナと他のリソース間 (コンテナとマネージドデータベース間など) の通信を開くことができます。

同じコンテナサービス内のコンテナ間の通信を開くには、次の例のように、`localhost` を参照する環境変数をコンテナのデプロイに追加します。

異なるコンテナサービスにあるコンテナ間の通信を開くには、次の例のように、プライベートドメイン (`container-service-1.service.local` など) を参照する環境変数をコンテナのデプロイに追加します。

コンテナと他のリソース間の通信を開くには、リソースのパブリックエンドポイント URL を参照する環境変数をコンテナのデプロイに追加します。例えば、Lightsail マネージドデータベースのパブリックエンドポイントは通常 `ls-123abc.czoexamplezqi.us-west-2.rds.amazonaws.com` です。したがって、次の例に示すように、環境変数でそのことを参照する必要があります。



コンテナログ

デプロイ内のすべてのコンテナがログを生成します。コンテナログは、コンテナ内で実行されている stdout および stderr にプロセスの流れを提供します。コンテナのログに定期的にアクセスして、オペレーションを診断します。詳細については、「Amazon Lightsail コンテナサービスのコンテナのログを表示」を参照してください。

デプロイバージョン

コンテナサービスで作成するすべてのデプロイは、デプロイバージョンとして保存されます。既存のデプロイのパラメータを変更すると、コンテナがサービスに再デプロイされ、デプロイが変更された場合は新しいデプロイバージョンが作成されます。各コンテナサービスの最新の 50 のデプロイバージョンが保存されます。50 のデプロイバージョンのいずれかを使用して、新しいデプロイを同じコンテナに作成できます。詳細については、「Amazon Lightsail コンテナサービスのデプロイバージョンの表示と管理」を参照してください。

デプロイのステータス

デプロイの作成後、デプロイは以下のいずれかの状態になります。

- アクティブ化中 – デプロイがアクティブ化されており、コンテナが作成されています。
- アクティブ – デプロイは正常に作成され、コンテナサービスで現在実行されています。
- 非アクティブ – 以前に正常に作成されたデプロイは、コンテナ上で実行されていません。
- 失敗 – デプロイで指定された 1 つ以上のコンテナが起動できなかったため、デプロイが失敗しました。

デプロイエラー

デプロイ内の 1 つ以上のコンテナの起動に失敗すると、デプロイは失敗します。デプロイが失敗し、コンテナサービスで以前のデプロイが実行されていた場合、コンテナサービスは以前のデプロイをアクティブなデプロイとして維持します。以前のデプロイがない場合、コンテナサービスは準備完了状態のままになり、現在アクティブなデプロイはありません。

失敗したデプロイのコンテナログを表示して、問題の診断とトラブルシューティングを行います。詳細については、「Amazon Lightsail コンテナサービスのコンテナのログを表示」を参照してください。

現在のコンテナサービスのデプロイの表示

以下の手順を実行して、Lightsail コンテナサービスの現在のデプロイを表示します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. 現在のデプロイを表示したいコンテナサービス名を選択します。

4. コンテナサービス管理ページで、[デプロイ] タブを選択します。

   デプロイページには、現在のデプロイとデプロイバージョンが一覧表示されます。コンテナサービスでデプロイをまだ作成していない場合、ページの両方のセクションは空です。

コンテナサービスのデプロイを作成または変更

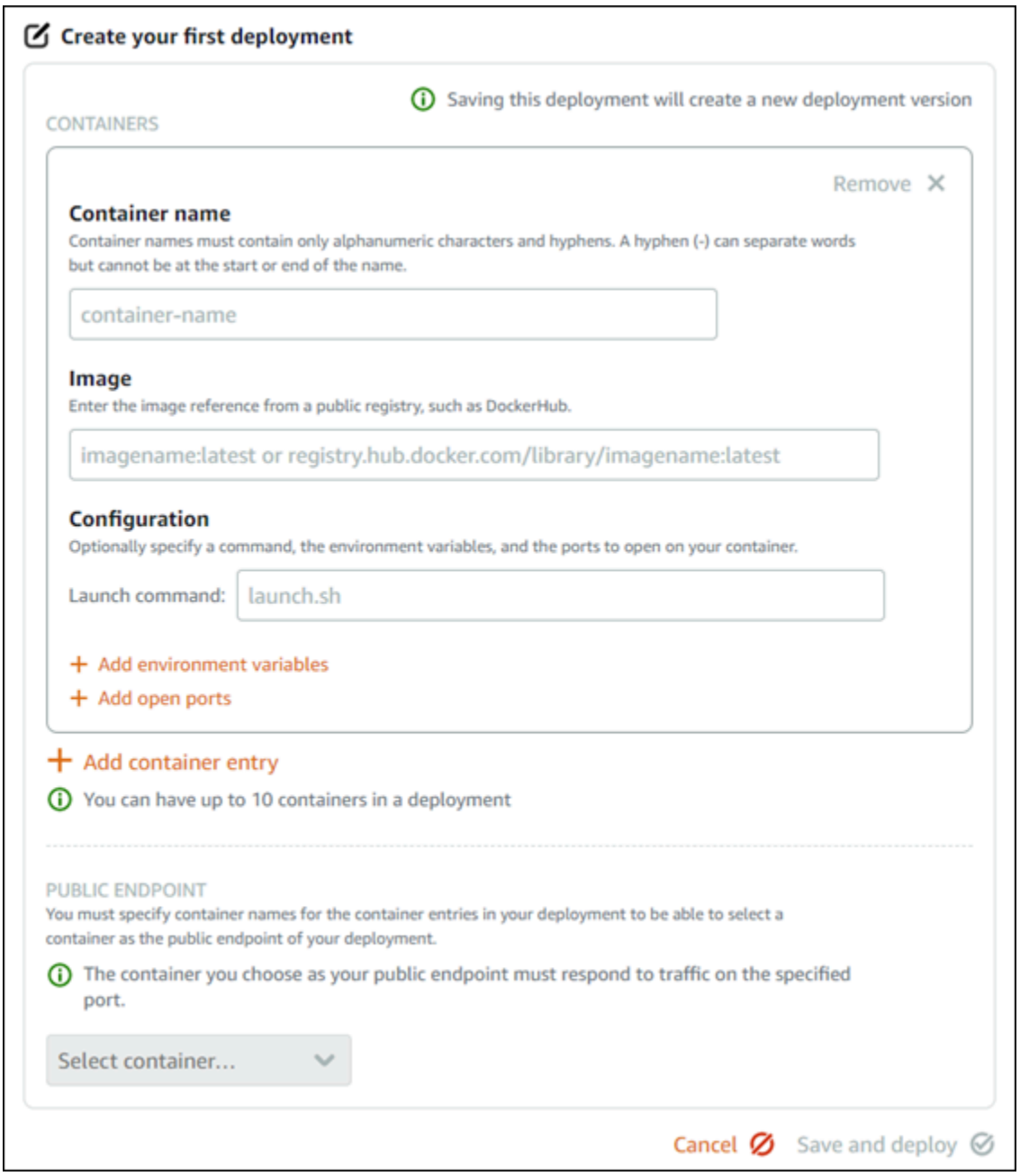
Lightsail コンテナサービスのデプロイを作成または変更するには、以下の手順を実行します。新しいデプロイを作成する場合も、既存のデプロイを変更する場合も、コンテナサービスでは、すべてのデプロイが新しいデプロイバージョンとして保存されます。詳細については、「Amazon Lightsail コンテナサービスのデプロイ バージョンの表示と管理」を参照してください。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. コンテナサービスのデプロイを作成または変更するコンテナサービス名を選択します。
4. コンテナサービスの管理ページで、デプロイタブを選択します。

   デプロイページには、現在のデプロイとデプロイのバージョンが一覧表示されます。(存在する場合)
5. 以下のオプションのいずれかを選択してください。

   - コンテナサービスに既存のデプロイがある場合は、デプロイの変更を選択します。
   - コンテナサービスにデプロイがない場合は、デプロイの作成を選択します。

     デプロイフォームが開き、既存のデプロイパラメータを編集したり、新しいデプロイパラメータを入力することができます。



6. デプロイのパラメータを入力します。指定できるデプロイパラメータの詳細については、このガイドの前半のデプロイパラメータセクションを参照してください。
7. [コンテナエントリを追加] を選択して、デプロイに複数のコンテナエントリを追加します。デプロイには最大 10 のコンテナエントリを追加することができます。
8. コンテナサービスのパブリックエンドポイントとして機能するデプロイ内のコンテナエントリを指定します。これには、HTTP または HTTPS ポート、選択したコンテナエントリのヘルスチェックパス、および詳細なヘルスチェック設定の指定が含まれます。詳細については、このガイドの前半にある「パブリックエンドポイントパラメータ」を参照してください。

9. デプロイのパラメータの入力が終了したら [保存してデプロイ] を選択して、コンテナサービス上にデプロイを作成します。

   コンテナサービスのステータスが [デプロイ中] に変わり、デプロイが作成されます。しばらくすると、デプロイのステータスに応じて、コンテナサービスのステータスが以下のいずれかに変わります。

   - デプロイが成功すると、コンテナサービスのステータスが [実行中] に変わり、デプロイのステータスが [アクティブ] に変わります。デプロイメントでパブリックエンドポイントを設定した場合、パブリックエンドポイントとして選択されたコンテナは、コンテナサービスのデフォルトドメインを介して使用できます。
   - デプロイが失敗し、コンテナサービスで以前のデプロイが実行されている場合、コンテナサービスのステータスが [実行中] に変わり、コンテナサービスは、以前のデプロイをアクティブデプロイとして維持します。以前のデプロイがない場合、コンテナサービスのステータスが [準備完了] に変わり、現在アクティブなデプロイはありません。失敗したデプロイのコンテナログを表示して、問題の診断とトラブルシューティングを行います。詳細については、「Amazon Lightsail コンテナサービスのコンテナログの表示」を参照してください。

トピック

- Lightsail コンテナサービスの容量をスケールする
- Lightsail コンテナサービスのデプロイバージョンの表示と管理
- Lightsail コンテナサービスログの分析

Lightsail コンテナサービスの容量をスケールする

Amazon Lightsail コンテナサービスの容量は、そのスケールと能力で構成されています。スケールはコンテナーサービス内のコンピューティングノードの数を指定し、能力はサービス内の各ノードのメモリと vCPUs を指定します。スケールは高可用性と大きな容量のために供給するノードの量を基に選択します。

このガイドの手順に従うことで、ダウンタイムを発生させることなくいつでもコンテナサービスの能力とスケールが不足している場合は動的に増やすことができ、過剰している場合は減らすことができます。Lightsail は、現在のデプロイと一緒に自動的に容量の変更を管理します。

> **Note**
>
> 新しいデプロイを作成すると、コンテナサービスの既存の使用率メトリックが消え、現在のデプロイのメトリクスのみが表示されます。

コンテナサービスの詳細については、「コンテナサービス」を参照してください。

コンテナサービスの容量を変更する

以下の手順を実行して Lightsail コンテナサービスの容量を変更します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. 容量を変更するコンテナサービスの名前を選択します。
4. コンテナサービス管理ページで、容量タブを選択します。

   現在のコンテナサービスの能力、スケール、月額料金は容量ページにあります。
5. 能力とスケールを変更するには、容量の変更を選択します。
6. 表示される確認プロンプトで、はい、続行しますを選択して、コンテナサービスの容量が変更されると現在のデプロイが再デプロイされることを確認します。
7. コンテナサービスの新しい能力とスケールを選択します。
8. はい、適用しますを選択して、新しい容量をコンテナサービスに適用します。

   コンテナサービスのステータスが更新中に変わります。数秒後、サービスのステータスが有効に変わり、新しい容量でのオペレーションが開始されます。

Lightsail コンテナサービスのデプロイバージョンの表示と管理

Amazon Lightsail コンテナサービスで作成するすべてのデプロイは、デプロイバージョンとして保存されます。既存のデプロイのパラメータを変更すると、コンテナがサービスに再デプロイされ、デプロイが変更された場合は新しいデプロイバージョンが作成されます。各コンテナサービスの最新の 50 のデプロイバージョンが保存されます。50 のデプロイバージョンのいずれかを使用して、新しいデプロイを同じコンテナに作成できます。このガイドでは、コンテナサービスのデプロイバージョンの表示および管理の方法を説明します。

コンテナサービスの詳細については、「コンテナサービス」を参照してください。

デプロイバージョンのステータス

各デプロイバージョンは、作成後、以下のいずれかのスタータスになります。

- デプロイ (アクティベート中) – デプロイは起動中です。
- アクティブ – デプロイは正常に作成され、コンテナサービスで現在実行されています。コンテナサービスでは、1 回につき 1 つのデプロイのみを実行することができます。
- 非アクティブ – 以前正常に作成されたデプロイは、コンテナ上で実行されなくなりました。
- 失敗 – デプロイで指定された 1 つ以上のコンテナが起動されなかったため、デプロイが失敗しました。

前提条件

開始する前に、Lightsail コンテナサービスを作成する必要があります。詳細については、「コンテナサービスを作成する」を参照してください。

コンテナを設定して起動するデプロイをコンテナサービス内に作成します。詳細については、「Amazon Lightsail コンテナサービスのデプロイの作成と管理」を参照してください。

コンテナサービスのデプロイバージョンを表示する

Lightsail コンテナサービスのデプロイバージョンを表示するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. デプロイバージョンを表示するコンテナサービスの名前を選択します。
4. コンテナサービス管理ページで、[デプロイ] タブを選択します。

   デプロイページには、現在のデプロイとデプロイバージョンが一覧表示されます。(存在する場合)
5. コンテナサービスのデプロイバージョンは、デプロイバージョンセクションにリストされています。

   各デプロイには、作成日、ステータス、アクションメニューがあります。
6. デプロイバージョンのアクションメニューで、以下のいずれかのオプションを選択します。

- 新しいデプロイを作成 – 選択したデプロイバージョンから新しいデプロイを作成するには、以下のオプションを選択します。デプロイの作成の詳細については、コンテナサービスのデプロイを作成または変更するを参照してください。

  > **Note**
  >
  > 失敗ステータスがあるバージョンから新しくデプロイを作成する場合、不具合を修正してからデプロイを作成する必要があります。修正されていない場合、デプロイは再び失敗する可能性が高いです。

- 詳細を表示 – このオプションを選択して、選択したデプロイバージョンのコンテナエントリとパブリックエンドポイントパラメーターを表示します。失敗したデプロイを診断する必要がある場合は、デプロイのコンテナログを表示することも可能です。詳細については、「コンテナサービスログを表示する」を参照してください。

Lightsail コンテナサービスログの分析

Amazon Lightsail コンテナサービスのデプロイのすべてのコンテナは ログを生成します。コンテナログは、コンテナ内で実行されるプロセスの stdout ストリームと stderr ストリームを提供します。コンテナのログに定期的にアクセスして、オペレーションを診断します。最新の 3 日間のログエントリが保存され、最も古いエントリが最新のエントリに置き換えられます。

コンテナログのフィルタ処理

コンテナログには、1 日に数百のエントリを含めることができます。フィルタリングオプションを使用すると、ログウィンドウに表示されるエントリ数が減り、探しているものを見つけやすくなります。コンテナログは、開始日と終了日 (現地時間)、および特定の期間でフィルタリングできます。期間でフィルタリングする場合、指定した期間のログエントリを含めるか除外するかを選択できます。

[include] (含む) または [exclude] (除く) のフィルター用語は、大文字と小文字を区別する完全一致を検索します。たとえば、「HTTP がメッセージに含まれるログイベントのみを含める」と指定した場合、HTTP がメッセージに含まれるログイベントはすべて表示されますが、http がメッセージに含まれないログイベントは表示されません。Error を除くと指定した場合、Error がメッセージに含まれていないすべてのログイベントが表示され、ERROR がメッセージに含まれているログイベントも表示されます。

前提条件

開始する前に、Lightsail コンテナサービスを作成する必要があります。詳細については、「Amazon Lightsail コンテナサービスの作成」を参照してください。

コンテナを設定して起動するデプロイをコンテナサービス内に作成します。詳細については、「Amazon Lightsail コンテナサービスのデプロイの作成と管理」を参照してください。

コンテナのログの表示

Lightsail コンテナ サービスのコンテナログを表示するには、以下の手順を実行します。

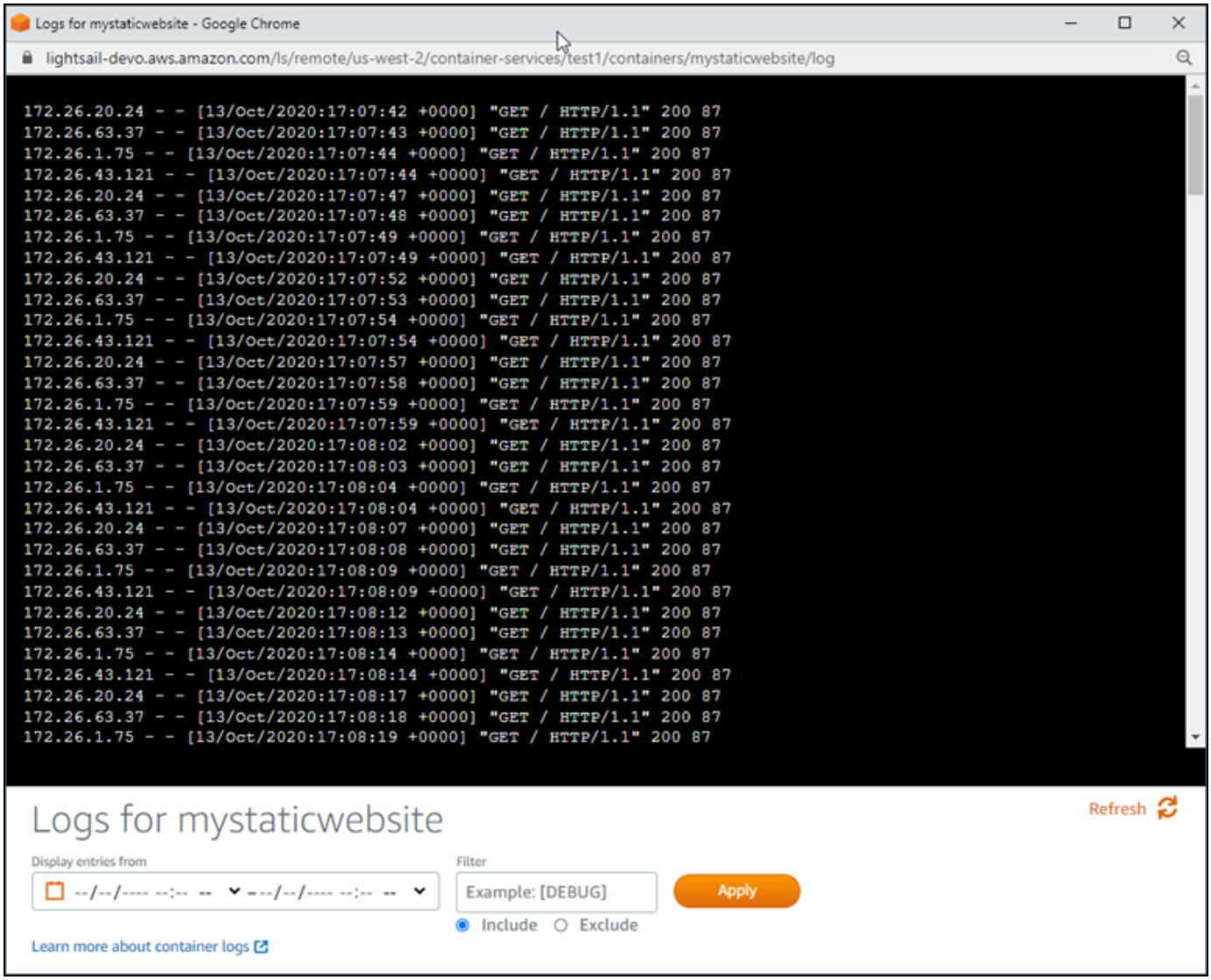
1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. コンテナログを表示するコンテナサービスの名前を選択します。
4. コンテナサービス管理ページで、[デプロイ] タブを選択します。

   [デプロイ] ページには、現在のデプロイとデプロイのバージョンが一覧表示されます。(存在する場合)
5. 次のいずれかのオプションを選択して、コンテナログを表示します。

   - 現在のデプロイのコンテナログにアクセスするには、ページの[Current deployment] (現在のデプロイ) セクションでコンテナエントリの[ログを開く] を選択します。
   - 以前のデプロイのコンテナログにアクセスするには、以前のデプロイのアクションメニューアイコン (⋮) を選択し、[デプロイバージョン] セクションを選択し、[詳細を表示] を選択します。表示される [バージョンの詳細] ページで、一覧表示されているコンテナエントリの [ログを開く] を選択します。

   ブラウザの新規ウィンドウでコンテナログが開きます。下にスクロールしてさらに多くのログエントリを表示したり、ページを更新して最新のエントリセットをロードしたりできます。フィルタオプションが、ページの下部に表示されます。

> **Note**
>
> ログエントリは昇順で、協定世界時 (UTC) で表示されます。つまり、最も古いログエントリが一番上に表示され、新しいログエントリを表示するには、下にスクロールする必要があります。



Lightsail でカスタムドメインを使用して安全なウェブアクセスを有効にする

登録済みドメイン名をサービスで使用するためには、Amazon Lightsail コンテナサービスのカスタムドメインを有効化します。カスタムドメインを有効にする前に、コンテナサービスは、最初の作成時にサービスに関連付けられたデフォルトドメインのトラフィックのみ受け入れます (例: `containerservicename.123456abcdef.us-west-2.cs.amazonlightsail.com`)。カスタムドメインを有効にする場合は、コンテナサービスで使用するドメイン用に作成した Lightsail SSL/TLS 証明書を選択し、その証明書から使用するドメインを選択します。カスタムドメインを有効にすると、コンテナサービスは、選択された証明書に関連付けられているすべてのドメインのトラフィックを受け入れます。

> **Important**
>
> ディストリビューションのオリジンとして Lightsail コンテナサービスを選択すると、Lightsail はディストリビューションのデフォルトドメイン名をカスタムドメインとしてコンテナサービスに自動的に追加します。これにより、ディストリビューションとコンテナサービスの間でトラフィックをルーティングできます。ただし、場合によっては、ディストリビューションのデフォルトドメイン名をコンテナサービスに手動で追加する必要があります。詳細については、「ディストリビューションのデフォルトドメインをコンテナサービスに追加する」を参照してください。

目次

- コンテナサービスのカスタムドメインの制限
- 前提条件
- コンテナサービスのカスタムドメインの表示
- コンテナサービスのカスタムドメインを有効にする
- コンテナサービスのカスタムドメインを無効にする

コンテナサービスのカスタムドメインの制限

コンテナサービスのカスタムドメインには、以下の制限が当てはまります。

- Lightsail コンテナサービスそれぞれに最大 4 つのカスタムドメインを使用できます。複数のサービスで同じドメインを使用することはできません。
- Lightsail DNS ゾーンを使用してドメインの DNS を管理する場合、ドメインの apex (例: `example.com`) とサブドメイン (例: `www.example.com`) のトラフィックをコンテナサービスにルーティングできます。

前提条件

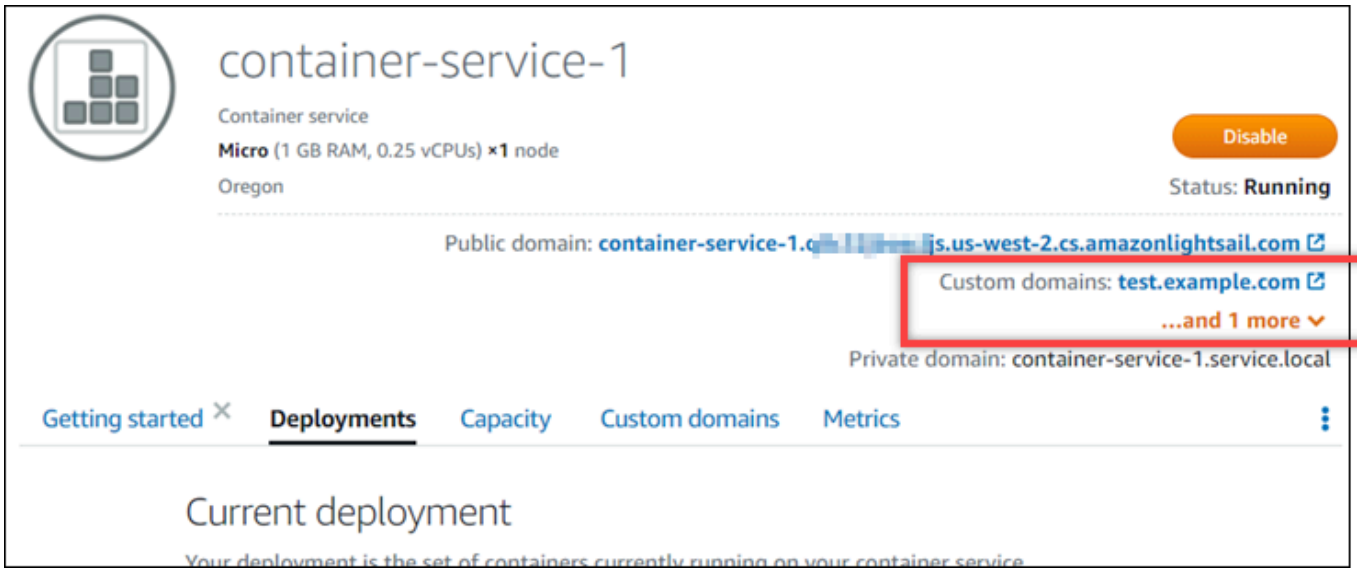
開始する前に、Lightsail コンテナサービスを作成する必要があります。詳細については、「Amazon Lightsail コンテナサービスの作成」を参照してください。

コンテナサービス用の SSL/TLS 証明書が作成され、検証されている必要があります。詳細については、「コンテナサービスの SSL/TLS 証明書を作成する」および「コンテナサービスの SSL/TLS 証明書を検証する」を参照してください。

コンテナサービスのカスタムドメインの表示

コンテナサービスで現在有効になっているカスタムドメインを表示するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. 有効にしたカスタムドメインを表示させたいコンテナサービスの名前を選択します。
4. 次の例に示すように、コンテナサービス管理ページの見出しでカスタムドメインの値を見つけます。これらは、コンテナサービスで現在有効になっているカスタムドメインです。



5. コンテナサービス管理ページで、[カスタムドメイン] タブを選択します。

   アタッチされた各証明書で使用されているカスタムドメインは、このページの [Custom domain SSL/TLS certificates] (カスタムドメイン SSL/TLS 証明書) セクションに一覧表示されています。コンテナサービスに現在アタッチされている証明書は、[Attached certificates] (アタッチされた証明書) セクションに一覧表示されています。

コンテナサービスのカスタムドメインを有効にする

Lightsail コンテナサービスのカスタムドメインを有効にするには、以下の手順を実行してサービスに証明書をアタッチします。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. カスタムドメインを有効にするコンテナサービスの名前を選択します。
4. コンテナサービス管理ページで [カスタムドメイン] タブを選択します。

   [カスタムドメイン] ページは、コンテナサービスに現在添付されている SSL/TLS 証明書 (存在する場合) を表示します。

5. [証明書のアタッチ] を選択します。

   証明書がない場合は、コンテナサービスにアタッチする前に、ドメインの SSL/TLS 証明書を作成してから検証する必要があります。詳細については、「コンテナサービスの SSL/TLS 証明書を作成する」を参照してください。

6. 表示されるドロップダウンメニューで、コンテナサービスとともに使用するドメインの有効な証明書を選択します。
7. 証明書情報が正しいことを確認し、[Attach] (アタッチ) を選択します。
8. コンテナサービスの [Status] (ステータス) が [Updating] (更新中) に変わります。ステータスが [Ready] (準備完了) に変わると、証明書のドメインが [Custom domains] (カスタムドメイン) セクションに表示されます。
9. [Add domain assignment] (ドメイン割り当ての追加) を選択して、ドメインがコンテナサービスを指すようにします
10. 証明書と DNS 情報が正しいことを確認し、[Add assignment] (割り当てを追加) を選択します。しばらくすると、選択したドメインのトラフィックがコンテナサービスによって受け入れられ始めます。

11. ドメイン割り当てを追加したら、新しいブラウザウィンドウを開き、コンテナサービスに対して有効にしたカスタムドメインを参照します。コンテナサービスで実行されているアプリケーション (存在する場合) がロードされます。

コンテナサービスのカスタムドメインを無効化する

Lightsail コンテナサービスのカスタムドメインを無効にするには、次の手順を実行します。サービスから証明書をデタッチするか、以前に選択したドメインの選択を解除します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. カスタムドメインを無効にするコンテナサービスの名前を選択します。
4. コンテナサービス管理ページで、[カスタムドメイン] タブを選択します。

   [カスタムドメイン] ページは、コンテナサービスに現在添付されている SSL/TLS 証明書 (存在する場合) を表示します。
5. 以下のオプションのいずれかを選択してください。

   1. [Configure container service domains] (コンテナサービスドメインを設定する) を選択して、以前に選択したドメインの選択を解除するか、コンテナサービスに関連付けられているドメインをさらに選択します。
   2. [デタッチ] を選択してコンテナサービスから証明書をデタッチし、関連付けられているすべてのドメインをサービスから削除します。

> **Important**
>
> トラフィックルートがコンテナサービスへのルーティングを停止し、代わりに別のリソースにルーティングするように、まだ行っていない場合は、ドメインの DNS レコードを変更します。

トピック

- ドメインのトラフィックを Lightsail コンテナサービスにルーティングする
- Route 53 を使用して、ドメインのトラフィックを Lightsail コンテナサービスにルーティングする

ドメインのトラフィックを Lightsail コンテナサービスにルーティングする

使用しているサービスでカスタムドメインを有効にした場合は、登録したドメイン名を Amazon Lightsail コンテナサービスに向けてポイントする必要があります。これを行うには、コンテナサービスで使用している証明書に指定されている各ドメインの DNS ゾーンに、エイリアスレコードを追加します。追加するレコードはすべて、コンテナサービスのデフォルトのドメイン (例:`https://<ServiceName>.<RandomGUID>.<AWSRegion>.cs.amazonlightsail.com`) に紐づけする必要があります。

このガイドでは、Lightsail DNS ゾーンを使用してコンテナサービスにドメインを指定する手順について説明します。Lightsail DNS ゾーンの詳細については、「Amazon Lightsail での DNS」を参照してください。

コンテナサービスの詳細については、「コンテナサービス」を参照してください。

> **Note**
>
> Route 53 を使用してドメインの DNS をホストする場合は、Route 53 のドメインのホストゾーンにエイリアスレコードを追加する必要があります。詳細については、「Route 53 のドメインへのトラフィックを Amazon Lightsail コンテナサービスにルーティング」を参照してください。

前提条件

開始する前に、Lightsail コンテナサービスのカスタムドメインを有効にする必要があります。詳細については、Amazon Lightsail コンテナサービスでのカスタムドメインの有効化と管理 を参照してください。

コンテナサービスのデフォルトドメインを取得する

以下の手順を実行して、コンテナサービスのデフォルトのドメイン名を取得します。このドメイン名は、ドメインの DNS にエイリアスレコードを追加するときに指定します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. デフォルトのドメイン名を取得するコンテナサービスの名前を選択します。
4. コンテナサービス管理ページのヘッダー部分にある、デフォルトのドメイン名を書き留めます。コンテナサービスのデフォルトのドメイン名

は、*<ServiceName>*.*<RandomGUID>*.*<AWSRegion>*.cs.amazonlightsail.com と似ています。

この値は、ドメイン DNS の正規名 (CNAME) レコードの一部として、追加する必要があります。この値はテキストファイルにコピー、ペーストして、後で参照できるようにしておくことをお勧めします。詳細については、このガイドの「ドメインの DNS ゾーンに CNAME レコードを追加する」セクションを参照してください。

ドメインの DNS ゾーンにレコードを追加する

アドレス (IPv4 の場合は A、IPv6 の場合は AAAA) および正規 (CNAME) レコードをドメインの DNS ゾーンに追加するには、次の手順を実行します。

1. 左側のナビゲーションペインで [ドメインと DNS] を選択します。
2. ページの [DNS zones] (DNS ゾーン) セクションで、レコードを追加したいドメイン名を選択します。そのレコードがユーザーのドメインへのトラフィックをコンテナサービスに送信します。
3. [DNS records] (DNS レコード) タブを選択します。
4. DNS ゾーンの現在の状態に応じて、次のいずれかのステップを実行します。
   - A、AAAA、ないし CNAME レコードを追加していない場合は、[Add record] (レコードの追加) を選択します。
   - 以前に A、AAAA、または CNAME レコードを追加している場合は、ページに記載されている既存の A、AAAA、ないし CNAME レコードの横にある編集アイコンを選択し、この手順のステップ 5 まで進んでください。
5. [レコードタイプ] のドロップダウンメニューで [A レコード]、[AAAA レコード]、[CNAME レコード] を選択します。
   - A レコードを追加して、ドメインの頂点 (例: `example.com`) をマッピングします。または、IPv4 ネットワーク下のコンテナサービスにサブドメイン (例: `www.example.com`) をマッピングします。
   - AAA レコードを追加して、ドメインの頂点 (例: `example.com`) をマッピングします。または、IPv6 ネットワーク下のコンテナサービスにサブドメイン (例: `www.example.com`) をマッピングします。
   - CNAME レコードを追加して、コンテナサービスのパブリックドメイン (デフォルト DNS) にサブドメイン (例: `www.example.com`) をマッピングします。
6. [Record name] (レコード名) テキストボックスに、次のいずれかのオプションを入力します。

- A レコードないし AAAA レコードの場合は、@ を入力してドメインの頂点 (例: example.com) へのトラフィックをコンテナサービスに送信します。またはサブドメイン (例: www) を入力して、サブドメイン (例: www.example.com) へのトラフィックをコンテナサービスにルーティングします。
- CNAME レコードの場合は、サブドメイン (例: www) を入力してサブドメイン (例: www.example.com)へのトラフィックをコンテナサービスに送信します。

7. 追加するレコードに応じて、次のいずれかの手順を実行します。

   - A レコードまたは AAAA レコードの場合は、[Resolves to] (解決先) テキストボックスにあるコンテナサービス名を選択します。
   - CNAME レコードの場合は、コンテナサービスのデフォルトのドメイン名を [Maps to] (マッピング先) テキストボックスに入力します。

8. 保存アイコンを選択して、レコードを DNS ゾーンに保存します。

   これらの手順を繰り返して、コンテナサービスで使用している証明書が紐づくドメイン用の DNS レコードを追加します。変更がインターネットの DNS を通じて伝達されるまで待ちます。数分後に、ドメインがコンテナサービスを指しているかどうか確認してください。

Route 53 を使用して、ドメインのトラフィックを Lightsail コンテナサービスにルーティングする

example.com などの登録済みドメインのトラフィックを、Amazon Lightsail コンテナサービスで実行されているアプリケーションにルーティングできます。これを行うには、Lightsail コンテナサービスのデフォルトドメインを指すドメインのホストゾーンに、エイリアスレコードを追加します。

このチュートリアルでは、Lightsail コンテナサービスのエイリアスレコードを Route 53 のホストゾーンに追加する方法を説明ます。これを行うには、 AWS Command Line Interface () を使用しますAWS CLI。Route 53 コンソールを使用しても実行されません。

> **Note**
>
> Lightsail を使用してドメインの DNS をホストする場合は、Lightsail のドメインのホストゾーンにエイリアスレコードを追加する必要があります。詳細については、「Amazon Lightsail のドメインへのトラフィックをLightsail コンテナサービスにルーティング」を参照してください。

目次

- ステップ 1: 前提条件を満たす
- ステップ 2: Lightsail コンテナサービスのホストゾーン ID を取得する
- ステップ 3: レコードセット JSON ファイルを作成する
- ステップ 4: Route 53 で、ドメインのホストゾーンにレコードを追加する

ステップ 1: 前提条件を満たす

以下の前提条件を完了します (まだの場合)。

- Route 53 でドメイン名を登録するか、Route 53 を登録された (既存の) ドメイン名の DNS サービスにします。詳細については、「Amazon Route 53 デベロッパーガイド」の「Amazon Route 53 を使用したドメイン名の登録」か「Amazon Route 53 を既存のドメインの DNS サービスにする」を参照してください。
- アプリケーションを Lightsail コンテナサービスにデプロイします。詳細については、「コンテナサービスのデプロイの作成と管理」を参照してください。
- Lightsail コンテナサービスで、登録済みのドメイン名を有効にします。詳細については、「カスタムドメインの有効化と管理」を参照してください。
- アカウント AWS CLI で を設定します。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

ステップ 2: Lightsail コンテナサービスのホストゾーン ID を取得する

Route 53 のホストゾーンにエイリアスレコードを追加するときは、Lightsail コンテナサービスのホストゾーン ID を指定する必要があります。たとえば、Lightsailコンテナサービスが米国西部 (オレゴン) (us-west-2) にある場合は AWS リージョン、Lightsailコンテナサービスのエイリアスレコードを Route 53 のホストゾーンに追加するZ0959753D43BBB908BAVときにホストゾーン ID を指定する必要があります。

Lightsail コンテナサービスを作成できる各 AWS リージョンのホストゾーン ID を以下に示します。

欧州 (ロンドン) (eu-west-2): Z0624918ZXDYQZLOXA66

米国東部 (バージニア北部) (us-east-1): Z06246771KYU0IRHI74W4

アジアパシフィック (シンガポール) (ap-southeast-1): Z0625921354DRJH4EY9V0

欧州 (アイルランド) (eu-west-1): Z0624732FELAMMKW3Y21

アジアパシフィック (東京) (ap-northeast-1): Z0626125UAU4JWQ9JSKN

アジアパシフィック (ソウル) (ap-northeast-2): Z06260262XZM84B2WPLHH

アジアパシフィック (ジャカルタ) (ap-southeast-3): Z03072883T5HFTY4T7CDL

アジアパシフィック (ムンバイ) (ap-south-1): Z10460781IQMISS0I0VVY

アジアパシフィック (シドニー) (ap-southeast-2): Z09597943PQQZATPFE96E

カナダ (中部) (ca-central-1): Z10450993RIRIJJUUMA5W

ヨーロッパ (フランクフルト) (eu-Central-1): Z06137433FV04OY4EC6L0

欧州 (ストックホルム) (eu-north-1): Z016970523TDG2TZMUXKK

欧州 (パリ) (eu-west-3): Z09594631DSW2QUR7CFGO

米国東部 (オハイオ) (us-east-2): Z10362273VJ548563IY84

米国西部 (オレゴン) (us-west-2): Z0959753D43BBB908BAV

ステップ 3: レコードセット JSON ファイルを作成する

を使用して Route 53 のドメインのホストゾーンに DNS レコードを追加する場合は AWS CLI、レコードの設定パラメータのセットを指定する必要があります。これを行う最も簡単な方法は、すべてのパラメータを含む JSON (.json) ファイルを作成し、 AWS CLI リクエストで JSON ファイルを参照することです。

次の手順を完了させ、エイリアスレコードのレコードセットパラメータを持つ JSON ファイルを作成します。

1. Windows の場合は Notepad、Linux の場合は Nano などのテキストエディタを開きます。
2. 次のテキストをコピーし、テキストエディターに貼り付けます。

```
{
  "Comment": "Comment",
  "Changes": [
    {
      "Action": "CREATE",
      "ResourceRecordSet": {
        "Name": "Domain.",
```

```
        "Type": "A",
        "AliasTarget": {
          "HostedZoneId": "LightsailContainerServiceHostedZoneID",
          "DNSName": " LightsailContainerServiceAddress.",
          "EvaluateTargetHealth": true
        }
      }
    }
  ]
}
```

ファイルで、次のサンプルテキストを独自のテキストに置き換えます。

- *Comment* を、レコードセットに関する個人的なメモまたはコメントで。
- *Domain* を、Lightsail コンテナサービスで使用したい登録済みのドメイン名で (例: example.com または www.example.com)。ドメインのルートを Lightsail コンテナサービスで使用する場合は、ドメインのサブドメイン空間で記号 @ を指定する必要があります (例えば、@.example.com)。
- *LightsailContainerServiceHostedZoneID* は、Lightsailコンテナサービスを作成した AWS リージョンのホストゾーン ID です。詳細については、このガイドの前半にある「ステップ 2: Lightsail コンテナサービスのホストゾーン ID を取得する」を参照してください。
- *LightsailContainerServiceAddress* とLightsailコンテナサービスのパブリックドメイン名。これを取得するには、Lightsail コンソールにサインインし、コンテナーサービスを参照し、コンテナーサービス管理ページのヘッダーセクションにリストされているパブリックドメインをコピーします (例えば、container-service-1.q8cexampleljs.us-west-2.cs.amazonlightsail.com)。

例:

```
{
  "Comment": "Alias record for Lightsail container service",
  "Changes": [
    {
      "Action": "CREATE",
      "ResourceRecordSet": {
        "Name": "@.example.com.",
        "Type": "A",
        "AliasTarget": {
          "HostedZoneId": "Z0959753D43BBB908BAV",
```

```
          "DNSName": "container-service-1.q8cexampleljs.us-
west-2.cs.amazonlightsail.com.",
          "EvaluateTargetHealth": true
        }
      }
    }
  ]
}
```

3. ローカルディレクトリに change-resource-record-sets.json としてファイルを保存します。

ステップ 4: Route 53 で、ドメインのホストゾーンにレコードを追加する

次の手順を完了させ、 AWS CLIを使用して、Route 53 のドメインのホストゾーンにレコードを追加します。これは、change-resource-record-sets コマンドを使用して行います。詳細については、「AWS CLI コマンドリファレンス」の「change-resource-record-sets」を参照してください。

> **Note**
>
> この手順を続行する前に、 をインストール AWS CLI し、 Lightsailと Route 53 用に設定する必要があります。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

1. ターミナルまたはコマンドプロントウィンドウを開きます。
2. 次のコマンドを入力して、Route 53 のドメインのホストゾーンにレコードを追加します。

```
aws route53 change-resource-record-sets --hosted-zone-id HostedZoneID --change-
batch PathToJsonFile
```

コマンドで、次のサンプルテキストを独自のテキストに置き換えます。

- *HostedZoneId* を、Route 53 で登録済みドメインのホストゾーンの ID で置き換えます。list-hosted-zones コマンドを使用して、Route 53 アカウントのホストゾーンの ID のリストを取得します。
- *PathToJsonFile* を、レコードパラメータを含む .json ファイルのコンピュータ上のローカルディレクトリフォルダパスで。詳細については、このガイドの前半にある「ステップ 3: レコードセット JSON ファイルを作成する」セクションを参照してください。

例:

Linux または Unix コンピュータの場合は、次の操作を行います。

```
aws route53 change-resource-record-sets --hosted-zone-id Z123456789ABCDEFGHIJ --
change-batch home/user/awscli/route53/change-resource-record-sets.json
```

Windows コンピュータの場合:

```
aws route53 change-resource-record-sets --hosted-zone-id Z123456789ABCDEFGHIJ --
change-batch file://C:\awscli\route53\change-resource-record-sets.json
```

以下の例のような結果が表示されるはずです。

```
H:\>aws route53 change-resource-record-sets --hosted-zone-id Z123456789ABCDEFGHIJ
--change-batch file://C:\awscli\route53\change-resource-record-sets.json
{
    "ChangeInfo": {
        "Id": "/change/C05953EXAMPLEZ4V4LOAC",
        "Status": "PENDING",
        "SubmittedAt": "2021-08-11T20:58:30.960000+00:00",
        "Comment": "Alias record for Lightsail container service"
    }
}
```

変更がインターネットの DNS を通じて伝播されるまで数時間かかる場合があります。完了すると、Route 53 の登録ドメインのインターネットトラフィックが Lightsail コンテナサービスへルーティング開始されるはずです。

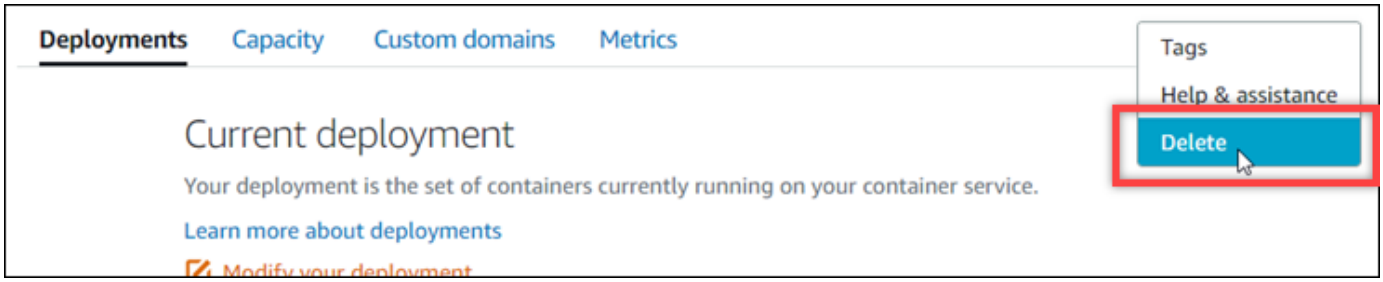
Lightsail コンテナサービスを削除する

Amazon Lightsail コンテナサービスを使用しなくなった場合、いつでも削除することができます。コンテナサービスを削除すると、そのサービスに関連付けられているすべてのデプロイと登録済みコンテナイメージが完全に破棄されます。ただし、作成した SSL/TLS 証明書やドメインは Lightsail アカウントに残るので、別のリソースで使用することができます。コンテナサービスの詳細については、「Amazon Lightsail のコンテナサービス」を参照してください。

コンテナサービスを削除

以下の手順を実行して、コンテナサービスを削除します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. 削除するコンテナサービスの名前を選択します。
4. タブメニューで省略記号アイコンを選択し、[削除] を選択します。



5. [コンテナサービスを削除する] をクリックしてサービスを削除します。
6. 表示されるプロンプトで、「はい、削除します」を選択して、削除が永続的であることを確認します。

   しばらくすると、コンテナサービスが削除されます。

「Amazon Lightsail」 のセキュリティ

「AWS」 ではクラウドセキュリティが最優先事項です。セキュリティを最も重視する組織の要件を満たすために構築された 「AWS」 のデータセンターとネットワークアーキテクチャは、お客様に大きく貢献します。

セキュリティは、「AWS」 と顧客の間の責任共有です。責任共有モデルでは、この責任がクラウドのセキュリティおよびクラウド内のセキュリティとして説明されています。

- クラウドのセキュリティ - 「AWS」 は、「AWS」 クラウドで 「AWS」 のサービスを実行するインフラストラクチャを保護する責任を負います。また、「AWS」 は、使用するサービスを安全に提供します。コンプライアンスプログラム、およびそれらが適用されるサービスについては、「コンプライアンスプログラムによる AWS 対象範囲内のサービス」を参照してください。
- クラウド内のセキュリティ - ユーザーの責任は、使用する 「AWS」 のサービスに応じて異なります。また、お客様は、データの機密性、お客様の会社の要件、および適用される法律および規制など、その他の要因についても責任を負います。

このドキュメントは、Amazon Lightsail を使用する際に責任共有モデルを適用する方法を理解するのに役立ちます。以下のトピックでは、セキュリティおよびコンプライアンスの目的を達成するために Amazon Lightsail を設定する方法を示します。また、Amazon Lightsail リソースのモニタリングや保護に役立つ他の AWS サービスの使用方法についても説明します。

Amazon Lightsail 内のインフラストラクチャセキュリティ

マネージドサービスである Amazon Lightsail は AWS グローバルネットワークセキュリティで保護されています。AWS セキュリティサービスと AWS がインフラストラクチャを保護する方法については、「AWS クラウドセキュリティ」を参照してください。インフラストラクチャセキュリティのベストプラクティスを使用して AWS 環境を設計するには、セキュリティの柱 - AWS Well-Architected Frameworkのインフラストラクチャ保護を参照してください。

AWS の発行済み API コールを使用して、ネットワーク経由で Lightsail にアクセスします。クライアントは以下をサポートする必要があります。

- Transport Layer Security (TLS)。TLS 1.2 が必須で、TLS 1.3 をお勧めします。
- DHE (楕円ディフィー・ヘルマン鍵共有) や ECDHE (楕円曲線ディフィー・ヘルマン鍵共有) などの完全前方秘匿性 (PFS) による暗号スイート。これらのモードは Java 7 以降など、ほとんどの最新システムでサポートされています。

また、リクエストにはアクセスキー ID と、IAM プリンシパルに関連付けられているシークレットアクセスキーを使用して署名する必要があります。または、AWS Security Token ServiceAWS STS を使用して、一時的なセキュリティ認証情報を生成し、リクエストに署名することもできます。

Amazon Lightsail での耐障害性

AWS グローバルインフラストラクチャは AWS リージョン およびアベイラビリティーゾーンを中心に構築されています。AWS リージョン には、低レイテンシー、高いスループット、そして高度の冗長ネットワークで接続されている複数の物理的に独立・隔離されたアベイラビリティーゾーンがあります。アベイラビリティーゾーンでは、ゾーン間で中断することなく自動的にフェールオーバーするアプリケーションとデータベースを設計および運用することができます。アベイラビリティーゾーンは、従来の単一または複数のデータセンターインフラストラクチャよりも可用性が高く、フォールトトレラントで、スケーラブルです。

AWS リージョンとアベイラビリティーゾーンの詳細については、「AWS グローバルインフラストラクチャ」を参照してください。

AWS では、Amazon Lightsail グローバルインフラストラクチャに加えて、データの回復性とバックアップのニーズに対応できるように複数の機能を提供しています。

- リージョン間でインスタンスとディスクのスナップショットをコピーする。詳細については、「スナップショット」を参照してください。
- インスタンスおよびディスクのスナップショットを自動化します。詳細については、「スナップショット」を参照してください。
- ロードバランサーを使用して、単一のアベイラビリティーゾーンまたは複数のアベイラビリティーゾーンにある複数のインスタンスの間で受信トラフィックを分散する。詳細については、「ロードバランサー」を参照してください。

Amazon Lightsail のためのアイデンティティおよびアクセス管理

オーディエンス

AWS Identity and Access Management (IAM) の使用方法は、 で行う作業によって異なります Amazon Lightsail。

サービスユーザー - Amazon Lightsail サービスを使用してジョブを実行する場合は、必要な権限と認証情報を管理者が用意します。作業を実行するためにさらに多くの Amazon Lightsail 機能を使用す

るとき、追加の権限が必要になる場合があります。アクセスの管理方法を理解すると、管理者から適切な権限をリクエストするのに役に立ちます。Amazon Lightsail の機能にアクセスできない場合は、「Identity and Access Management (IAM)のトラブルシューティング」を参照してください。

サービス管理者 - 社内の Amazon Lightsail リソースを担当している場合は、通常、Amazon Lightsail へのフルアクセスがあります。従業員がどの Amazon Lightsail特徴とリソースアクセスする必要があるかを決定するのは管理者のジョブです。その後、IAM 管理者にリクエストを送信して、サービスユーザーの許可を変更する必要があります。このページの情報を点検して、IAM の基本概念を理解してください。会社で Amazon Lightsail ともに IAM を使用する方法の詳細については、「Amazon Lightsail と IAM を連携する方法」を参照してください。

IAM 管理者 - 管理者は、Amazon Lightsail へのアクセスを管理するポリシーの書き込み方法の詳細について確認する場合があります。IAM で使用できる Amazon Lightsail アイデンティベースのポリシーの例については、「Amazon Lightsail アイデンティベースのポリシーの例」を参照してください。

アイデンティティを使用した認証

認証とは、ID 認証情報 AWS を使用して にサインインする方法です。を使用したサインインの詳細については AWS マネジメントコンソール、IAM ユーザーガイドの「IAM コンソールとサインインページ」を参照してください。

AWS アカウント ルートユーザー、IAM ユーザー、または IAM ロールを引き受けることで、認証 (にサインイン AWS) される必要があります。会社のシングルサインオン認証を使用することも、Google や Facebook を使用してサインインすることもできます。このような場合、管理者は以前に IAM ロールを使用して ID フェデレーションを設定しました。別の会社の認証情報 AWS を使用して にアクセスすると、間接的にロールを引き受けることになります。

AWS マネジメントコンソール に直接サインインするには、パスワード、およびルートユーザーの E メールまたは IAM ユーザー名を使用します。ルートユーザーまたは IAM ユーザーアクセスキーを使用して、 AWS プログラムで にアクセスできます。 AWS には、認証情報を使用してリクエストを暗号化して署名するための SDK およびコマンドラインツールが用意されています。 AWS ツールを使用しない場合は、自分でリクエストに署名する必要があります。これには、インバウンド API リクエストを認証するためのプロトコルである署名バージョン 4 を使用します。リクエストの認証の詳細については、『』の「署名バージョン 4 の署名プロセスAWS 全般のリファレンス」を参照してください。

使用する認証方法を問わず、追加のセキュリティ情報の提供を要求される場合もあります。たとえば、 では、アカウントのセキュリティを高めるために多要素認証 (MFA) を使用する AWS ことをお

勧めします。詳細については、「IAM ユーザーガイド」の 「AWSでの多要素認証 (MFA) の使用」を参照してください。

AWS アカウント ルートユーザー

を作成するときは AWS アカウント、まず、すべての AWS のサービス および リソースへの完全なアクセス権を持つ AWS アカウント root ユーザーと呼ばれる 1 つのサインインアイデンティティから始めます。日常的なタスクには、ルートユーザーを使用しないことを強くお勧めします。ルートユーザー認証情報を必要とするタスクについては、「IAM ユーザーガイド」の「ルートユーザー認証情報が必要なタスク」を参照してください。

IAM ユーザーとグループ

IAM ユーザーは、特定の個人やアプリケーションに対する特定のアクセス許可を持つアイデンティティです。長期認証情報を持つ IAM ユーザーの代わりに一時的な認証情報を使用することをお勧めします。詳細については、IAM ユーザーガイドの「ID プロバイダーとのフェデレーションを使用して にアクセスする必要がある AWS」を参照してください。

IAM グループは、IAM ユーザーの集合を指定し、大量のユーザーに対するアクセス許可の管理を容易にします。詳細については、「IAM ユーザーガイド」の「IAM ユーザーに関するユースケース」を参照してください。

IAM ロール

IAM ロールは、特定のアクセス許可を持つアイデンティであり、一時的な認証情報を提供します。ユーザーから IAM ロール (コンソール) に切り替えるか、 または API オペレーションを呼び出すことで、ロールを引き受けることができます。 AWS CLI AWS 詳細については、「IAM ユーザーガイド」の「ロールを引き受けるための各種方法」を参照してください。

IAM ロールは、フェデレーションユーザーアクセス、一時的な IAM ユーザーのアクセス許可、クロスアカウントアクセス、クロスサービスアクセス、および Amazon EC2 で実行するアプリケーションに役立ちます。詳細については、IAM ユーザーガイド の IAM でのクロスアカウントリソースアクセス を参照してください。

IAM ロールと一時的な認証情報は、次の状況で役立ちます:

- 一時的な IAM ユーザー許可 - IAM ユーザーは、特定のタスクに対して複数の異なる許可を一時的に IAM ロールで引き受けることができます。
- フェデレーションユーザーアクセス – フェデレーテッド ID に許可を割り当てるには、ロールを作成してそのロールの許可を定義します。フェデレーテッド ID が認証されると、その ID はロール

に関連付けられ、ロールで定義されている許可が付与されます。フェデレーションのロールについては、IAM ユーザーガイド の サードパーティー ID プロバイダー (フェデレーション) 用のロールを作成する を参照してください。IAM Identity Center を使用する場合は、許可セットを設定します。アイデンティティが認証後にアクセスできるものを制御するため、IAM Identity Center は、権限セットを IAM のロールに関連付けます。アクセス許可セットの詳細については、AWS IAM アイデンティティセンター User Guide の Permission sets を参照してください。

- クロスアカウントアクセス - IAM ロールを使用して、自分のアカウントのリソースにアクセスすることを、別のアカウントの人物 (信頼済みプリンシパル) に許可できます。クロスアカウントアクセス権を付与する主な方法は、ロールを使用することです。ただし、一部の では AWS のサービス、 (プロキシとしてロールを使用する代わりに) リソースに直接ポリシーをアタッチできます。クロスアカウントアクセスにおけるロールとリソースベースのポリシーの違いについては、「IAM ユーザーガイド」の「IAM ロールとリソースベースのポリシーとの相違点」を参照してください。
- クロスサービスアクセス — 一部の は他の の機能 AWS のサービス を使用します AWS のサービス。例えば、あるサービスで呼び出しを行うと、通常そのサービスによって Amazon EC2 でアプリケーションが実行されたり、Amazon S3 にオブジェクトが保存されたりします。サービスでは、呼び出し元プリンシパルの許可、サービスロール、またはサービスリンクロールを使用してこれを行う場合があります。
  - 転送アクセスセッション (FAS) – IAM ユーザーまたはロールを使用してアクションを実行すると AWS、プリンシパルと見なされます。ポリシーによって、プリンシパルに許可が付与されます。一部のサービスを使用する際に、アクションを実行することで、別サービスの別アクションがトリガーされることがあります。この場合、両方のアクションを実行するための権限が必要です。アクションがポリシーで追加の依存アクションを必要とするかどうかを確認するには、「サービス認可リファレンス」の「 Amazon Lightsail のアクション、リソース、および条件キー」をご参照ください。
  - サービスロール - サービスがユーザーに代わってアクションを実行するために引き受ける IAM ロールです。IAM 管理者は、IAM 内からサービスロールを作成、変更、削除できます。詳細については、IAM ユーザーガイドの AWS のサービスに許可を委任するロールを作成するを参照してください。
  - サービスにリンクされたロール – サービスにリンクされたロールは、 にリンクされたサービスロールの一種です AWS のサービス。サービスは、ユーザーに代わってアクションを実行するロールを引き受けることができます。サービスにリンクされたロールは に表示され AWS アカウント 、サービスによって所有されます。IAM 管理者は、サービスリンクロールのアクセス許可を表示できますが、編集することはできません。

- Amazon EC2 で実行されているアプリケーション – IAM ロールを使用して、EC2 インスタンスで実行され、 AWS CLI または AWS API リクエストを行うアプリケーションの一時的な認証情報を管理できます。これは、EC2 インスタンス内でのアクセスキーの保存に推奨されます。EC2 インスタンスに AWS ロールを割り当て、そのすべてのアプリケーションで使用できるようにするには、インスタンスにアタッチされたインスタンスプロファイルを作成します。インスタンスプロファイルにはロールが含まれ、EC2 インスタンスで実行されるプログラムは一時的な認証情報を取得できます。詳細については、「IAM ユーザーガイド」の「Amazon EC2 インスタンスで実行されるアプリケーションに IAM ロールを使用して許可を付与する」を参照してください。

IAM ロールと IAM ユーザーのどちらを使用するかについては、IAM ユーザーガイドの(IAM ユーザーではなく) IAM ロールをいつ作成したら良いのか?を参照してください。

ポリシーを使用したアクセスの管理

でアクセスを制御する AWS には、ポリシーを作成し、ID AWS またはリソースにアタッチします。ポリシーは、アイデンティティまたはリソースに関連付けられたときにアクセス許可を定義します。は、プリンシパルがリクエストを行うときにこれらのポリシー AWS を評価します。ほとんどのポリシーは JSON ドキュメント AWS として に保存されます。JSON ポリシードキュメントの詳細については、「IAM ユーザーガイド」の「JSON ポリシー概要」を参照してください。

管理者は、ポリシーを使用して、どのプリンシパルがどのリソースに対して、どのような条件でアクションを実行できるかを定義することで、誰が何にアクセスできるかを指定します。

デフォルトでは、ユーザーやロールにアクセス許可はありません。IAM 管理者は IAM ポリシーを作成してロールに追加し、このロールをユーザーが引き受けられるようにします。IAM ポリシーは、オペレーションの実行方法を問わず、アクセス許可を定義します。

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

すべての IAM エンティティ (ユーザーまたはロール) は、許可のない状態からスタートします。言い換えると、デフォルト設定では、ユーザーは何もできず、自分のパスワードを変更することすらできません。何かを実行する許可をユーザーに付与するには、管理者がユーザーに許可ポリシーをアタッチする必要があります。また、管理者は、必要な許可があるグループにユーザーを追加できます。管理者がグループに許可を付与すると、そのグループ内のすべてのユーザーにこれらの許可が付与されます。

IAM ポリシーは、オペレーションの実行方法を問わず、アクションの許可を定義します。例えば、`iam:GetRole` アクションを許可するポリシーがあるとします。そのポリシーを持つユーザーは、 AWS マネジメントコンソール、、 AWS CLIまたは AWS API からロール情報を取得できます。

アイデンティティベースのポリシー

アイデンティティベースのポリシーは、アイデンティティ (ユーザー、グループ、またはロール) にアタッチできる JSON アクセス許可ポリシードキュメントです。これらのポリシーは、アイデンティティがどのリソースに対してどのような条件下でどのようなアクションを実行できるかを制御します。アイデンティティベースポリシーの作成方法については、IAM ユーザーガイド の カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する を参照してください。

アイデンティティベースのポリシーは、インラインポリシー (単一の ID に直接埋め込む) または管理ポリシー (複数の ID にアタッチされたスタンドアロンポリシー) にすることができます。管理ポリシーとインラインポリシーのいずれかを選択する方法については、「IAM ユーザーガイド」の「管理ポリシーとインラインポリシーのいずれかを選択する」を参照してください。

アイデンティティベースのポリシーは、IAM ユーザーグループ、ユーザーのグループ、ロールなど、アイデンティティにアタッチできる JSON 許可ポリシードキュメントです。これらのポリシーは、ユーザーとロールが実行できるアクション、リソース、および条件をコントロールします。アイデンティティベースポリシーの作成方法については、「IAM ユーザーガイド」の「カスタマー管理ポリシーでカスタム IAM アクセス許可を定義する」を参照してください。

リソースベースのポリシー

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。例としては、IAM ロール信頼ポリシーや Amazon S3 バケットポリシーなどがあります。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスを制御できます。リソースベースのポリシーでは、プリンシパルを指定する必要があります。

リソースベースのポリシーは、そのサービス内にあるインラインポリシーです。リソースベースのポリシーでは、IAM の AWS マネージドポリシーを使用できません。

リソースベースのポリシーは、リソースに添付する JSON ポリシードキュメントです。リソースベースのポリシーには例として、IAM ロールの信頼ポリシーや Amazon S3 バケットポリシーがあげられます。リソースベースのポリシーをサポートするサービスでは、サービス管理者はポリシーを使用して特定のリソースへのアクセスをコントロールできます。ポリシーがアタッチされているリソースの場合、指定されたプリンシパルがそのリソースに対して実行できるアクションと条件は、ポリシーによって定義されます。リソースベースのポリシーで、プリンシパルを指定する必要がありま

す。プリンシパルには、アカウント、ユーザー、ロール、フェデレーティッドユーザー、または を含めることができます AWS のサービス。

アクセスコントロールリスト (ACL)

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、または ロール) がリソースにアクセスするためのアクセス許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

Amazon S3、および Amazon VPC は AWS WAF、ACLs。ACL の詳細については、Amazon Simple Storage Service デベロッパーガイド の アクセスコントロールリスト (ACL) の概要 を参照してください。

アクセスコントロールリスト (ACL) は、どのプリンシパル (アカウントメンバー、ユーザー、または ロール) がリソースにアクセスするためのアクセス許可を持つかを制御します。ACL はリソースベースのポリシーに似ていますが、JSON ポリシードキュメント形式は使用しません。

その他のポリシータイプ

AWS は、より一般的なポリシータイプによって付与されるアクセス許可の最大数を設定できる追加のポリシータイプをサポートしています。

- アクセス許可の境界 – アイデンティティベースのポリシーで IAM エンティティに付与することのできるアクセス許可の数の上限を設定します。詳細については、「IAM ユーザーガイド」の「IAM エンティティのアクセス許可境界」を参照してください。
- サービスコントロールポリシー (SCP) - AWS Organizations内の組織または組織単位の最大のアクセス許可を指定します。詳細については、「AWS Organizations ユーザーガイド」の「サービスコントロールポリシー」を参照してください。
- リソースコントロールポリシー (RCP) – は、アカウント内のリソースで利用できる最大数のアクセス許可を定義します。詳細については、「AWS Organizations ユーザーガイド」の「リソースコントロールポリシー (RCP)」を参照してください。
- セッションポリシー – ロールまたはフェデレーションユーザーの一時セッションを作成する際にパラメータとして渡される高度なポリシーです。詳細については、「IAM ユーザーガイド」の「セッションポリシー」を参照してください。

- アクセス許可の境界 - アクセス許可の境界は、アイデンティティベースポリシーによって IAM エンティティ (IAM ユーザーまたはロール) に付与できる権限の上限を設定する高度な機能です。エンティティに許可の境界を設定できます。結果として許可される範囲は、エンティティのアイデン

ティティベースポリシーとその許可の境界の共通部分になります。Principal フィールドでユーザーまたはロールを指定するリソースベースのポリシーでは、許可の境界は制限されません。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。アクセス許可の境界の詳細については、「IAM ユーザーガイド」の「IAM エンティティのアクセス許可の境界」を参照してください。

- サービスコントロールポリシー (SCPs) – SCPsは、 の組織または組織単位 (OU) の最大アクセス許可を指定する JSON ポリシーです AWS Organizations。 AWS Organizations は、ビジネスが所有する複数の AWS アカウント をグループ化して一元管理するためのサービスです。組織内のすべての機能を有効にすると、サービスコントロールポリシー (SCP) を一部またはすべてのアカウントに適用できます。SCP は、各 AWS アカウント ルートユーザーを含むメンバーアカウントのエンティティのアクセス許可を制限します。Organizations と SCP の詳細については、AWS Organizations ユーザーガイドの「SCP の仕組み」を参照してください。
- セッションポリシー - セッションポリシーは、ロールまたはフェデレーションユーザーの一時的なセッションをプログラムで作成する際にパラメータとして渡す高度なポリシーです。結果としてセッションの権限は、ユーザーまたはロールのアイデンティティベースポリシーとセッションポリシーの共通部分になります。また、リソースベースのポリシーから権限が派生する場合もあります。これらのポリシーのいずれかを明示的に拒否した場合、権限は無効になります。詳細については、「IAM ユーザーガイド」の「セッションポリシー」を参照してください。

複数のポリシータイプ

1 つのリクエストに複数のタイプのポリシーが適用されると、結果として作成されるアクセス許可を理解するのがさらに難しくなります。が複数のポリシータイプが関与する場合にリクエストを許可するかどうか AWS を決定する方法については、「IAM ユーザーガイド」の「ポリシー評価ロジック」を参照してください。

トピック

- AWS の 管理ポリシー Amazon Lightsail
- Amazon Lightsail と IAM の連携方法
- IAM ユーザーに Lightsail アクセスを付与する

AWS の 管理ポリシー Amazon Lightsail

ユーザー、グループ、ロールにアクセス許可を追加するには、自分でポリシーを記述するよりも AWS 管理ポリシーを使用する方が簡単です。チームに必要な権限のみを提供する IAM カスタマーマネージドポリシーを作成するには時間と専門知識が必要です。すぐに開始するには、 AWS マネージ

ドポリシーを使用できます。これらのポリシーは、一般的なユースケースをターゲット範囲に含めており、 AWS アカウントで利用できます。 AWS 管理ポリシーの詳細については、IAM ユーザーガイドの「 AWS 管理ポリシー」を参照してください。

AWS サービスは、 AWS 管理ポリシーを維持および更新します。 AWS 管理ポリシーのアクセス許可は変更できません。サービスは、新機能をサポートするために、 AWS 管理ポリシーに追加のアクセス許可を追加することがあります。この種類の更新はポリシーがアタッチされている、すべてのアイデンティティ (ユーザー、グループおよびロール) に影響を与えます。サービスは、新機能が起動されたとき、または新しいオペレーションが利用可能になったときに、 AWS マネージドポリシーを更新する可能性が最も高いです。サービスは AWS マネージドポリシーからアクセス許可を削除しないため、ポリシーの更新によって既存のアクセス許可が損なわれることはありません。

さらに、 は、複数のサービスにまたがるジョブ関数の マネージドポリシー AWS をサポートしています。例えば、ReadOnlyAccess AWS 管理ポリシーは、すべての AWS サービスとリソースへの読み取り専用アクセスを提供します。サービスが新機能を起動すると、 は新しいオペレーションとリソースの読み取り専用アクセス許可 AWS を追加します。ジョブ機能のポリシーの一覧および詳細については、「IAM ユーザーガイド」の「AWS のジョブ機能のマネージドポリシー」を参照してください。

AWS マネージドポリシー: LightsailExportAccess

LightsailExportAccess を IAM エンティティにアタッチすることはできません。このポリシーは、ユーザーに代わって Lightsail がアクションを実行することを許可する、サービスにリンクされたロールにアタッチされます。詳細については、「サービスにリンクされたロール」を参照してください。

このポリシーは、Lightsail がインスタンスおよびディスクスナップショットを Amazon Elastic Compute Cloud にエクスポートし、現在のアカウントレベルで Block Public Access 設定を Amazon Simple Storage Service (Amazon S3) から取得できるようにするアクセス許可を付与します。

アクセス許可の詳細

このポリシーには、以下のアクセス許可が含まれています。

- `ec2` – インスタンスイメージとディスクスナップショットの一覧表示とコピーをするためのアクセスを許可します。
- `iam` – サービスにリンクされたロールの削除と、サービスにリンクされたロールの削除のステータスを取得するためのアクセスを許可します。
- `s3` – AWS アカウント`PublicAccessBlock`の設定を取得するためのアクセスを許可します。

JSON

```
{
 "Version":"2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "iam:DeleteServiceLinkedRole",
    "iam:GetServiceLinkedRoleDeletionStatus"
   ],
   "Resource": "arn:aws:iam::*:role/aws-service-role/lightsail.amazonaws.com/
AWSServiceRoleForLightsail*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "ec2:CopySnapshot",
    "ec2:DescribeSnapshots",
    "ec2:CopyImage",
    "ec2:DescribeImages"
   ],
   "Resource": "*"
  },
  {
   "Effect": "Allow",
   "Action": [
    "s3:GetAccountPublicAccessBlock"
   ],
   "Resource": "*"
  }
 ]
}
```

Lightsail AWS 管理ポリシーの更新

- `LightsailExportAccess` 管理ポリシーの編集

  `LightsailExportAccess` マネージドポリシーに `s3:GetAccountPublicAccessBlock` アクションが追加されました。現在のアカウントレベルの Block Public Access 設定を Amazon S3 から取得する許可を Lightsail に付与します。

2022 年 1 月 14 日

- Lightsail は変更の追跡を開始しました

  Lightsail は、 AWS 管理ポリシーの変更の追跡を開始しました。

  2022 年 1 月 14 日

Amazon Lightsail と IAM の連携方法

Lightsail へのアクセスを管理するために IAM を使用する前に、Lightsail でどの IAM特徴が使用できるかを理解しておく必要があります。Lightsail およびその他の AWS のサービスが IAM と連携する方法の概要を把握するには、IAM ユーザーガイドのAWS 「IAM と連携する のサービス」を参照してください。

Lightsail アイデンティティベースのポリシー

IAM アイデンティティベースのポリシーでは、許可または拒否されたアクションとリソースを指定でき、さらにアクションが許可または拒否された条件を指定できます。Lightsail は、特定のアクション、リソース、および条件キーをサポートします。JSON ポリシーで使用するすべての要素については、「IAM ユーザーガイド」の「IAM JSON ポリシー要素のリファレンス」を参照してください。

アクション

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

JSON ポリシーの `Action` 要素にはポリシー内のアクセスを許可または拒否するために使用できるアクションが記述されます。このアクションは関連付けられたオペレーションを実行するためのアクセス許可を付与するポリシーで使用されます。

Lightsail のポリシーアクションは、アクションの前に次のプレフィックスを使用します: `lightsail:`。たとえば、Lightsail `CreateInstances` API オペレーションで Lightsail インスタンスを実行するためのアクセス許可をユーザーに付与するには、ポリシーに `lightsail:CreateInstances` アクションを含めます。ポリシーステートメントには、`Action` 要素または `NotAction` 要素のいずれかを含める必要があります。 Lightsail は、このサービスで実行できるタスクを説明する独自の一連のアクションを定義します。

単一のステートメントに複数のアクションを指定するには、次のようにカンマで区切ります。

```
"Action": [
      "lightsail:action1",
      "lightsail:action2"
```

ワイルドカード (*) を使用して複数アクションを指定できます。例えば、Create という単語で始まるすべてのアクションを指定するには、次のアクションを含めます。

```
"Action": "lightsail:Create*"
```

Lightsail アクションのリストを表示するには、IAM ユーザーガイドのAmazon Lightsail によって定義されたアクションを参照してください。

リソース

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Resource JSON ポリシー要素はアクションが適用されるオブジェクトを指定します。ベストプラクティスとして、Amazon リソースネーム (ARN) を使用してリソースを指定します。リソースレベルのアクセス許可をサポートしないアクションの場合は、ステートメントがすべてのリソースに適用されることを示すために、ワイルドカード (*) を使用します。

```
"Resource": "*"
```

> **Important**
>
> Lightsail は API アクションによっては、リソースレベルのアクセス許可をサポートしません。詳細については、「リソースレベルのアクセス許可およびタグに基づく承認のサポート」を参照してください。

Lightsail インスタンスリソースには次のような ARN があります。

```
arn:${Partition}:lightsail:${Region}:${Account}:Instance/${InstanceId}
```

ARN の形式の詳細については、「Amazon リソースネーム (ARNs AWS 「サービス名前空間」を参照してください。

例えば、ステートメントで ea123456-e6b9-4f1d-b518-3ad1234567e6 インスタンスを指定するには、次の ARN を使用します。

```
"Resource": "arn:aws:lightsail:us-east-1:123456789012:Instance/ea123456-e6b9-4f1d-b518-3ad1234567e6"
```

特定のアカウントに属するすべてのインスタンスを指定するには、ワイルドカード (*) を使用します。

```
"Resource": "arn:aws:lightsail:us-east-1:123456789012:Instance/*"
```

リソースを作成するためのアクションなど、一部の Lightsail アクションは、特定のリソースでは実行できません。このような場合は、ワイルドカード *を使用する必要があります。

```
"Resource": "*"
```

Lightsail API アクションの多くが複数のリソースと関連します。たとえば、AttachDisk では Lightsail ブロックストレージディスクをインスタンスにアタッチするため、IAM ユーザーにはディスクおよびインスタンスを使用するアクセス許可が必要になります。複数のリソースを単一のステートメントで指定するには、ARN をカンマで区切ります。

```
"Resource": [
      "resource1",
      "resource2"
```

Lightsail リソースタイプとその ARN のリストを表示するには、IAM ユーザーガイドのAmazon Lightsail で定義されるリソースを参照してください。どのアクションで各リソースの ARN を指定できるかについては、Amazon Lightsail で定義されるアクションを参照してください。

条件キー

管理者は JSON AWS ポリシーを使用して、誰が何にアクセスできるかを指定できます。つまり、どのプリンシパルがどのリソースに対してどのような条件下でアクションを実行できるかということです。

Condition 要素は、定義された基準に基づいてステートメントが実行される時期を指定します。イコールや未満などの条件演算子を使用して条件式を作成して、ポリシーの条件とリクエスト内の値を一致させることができます。すべての AWS グローバル条件キーを確認するには、「IAM ユーザーガイド」のAWS 「グローバル条件コンテキストキー」を参照してください。

Lightsail にはサービス固有条件キーがありませんが、いくつかのグローバル条件キーの使用がサポートされています。すべての AWS グローバル条件キーを確認するには、IAM ユーザーガイドのAWS「グローバル条件コンテキストキー」を参照してください。

Lightsail 条件キーのリストを確認するには、IAM ユーザーガイドの「Amazon Lightsail の条件キー」を参照してください。どのアクションおよびリソースと条件キーを使用できるかについては、「Amazon Lightsail で定義されるアクション」を参照してください。

例

Lightsail のアイデンティティベースのポリシーの例については、「Amazon Lightsail アイデンティティベースのポリシーの例」を参照してください。

Lightsail リソースベースのポリシー

Lightsail は、リソースベースのポリシーをサポートしません。

アクセスコントロールリスト (ACL)

Lightsail はアクセスコントロールリスト (ACL) をサポートしません。

Lightsail タグに基づいた承認

タグを Lightsail リソースにアタッチすることも、Lightsail へのリクエストでタグを渡すこともできます。タグに基づいてアクセスを制御するには、`lightsail:ResourceTag/key-name`、`aws:RequestTag/key-name`、または `aws:TagKeys` の条件キーを使用して、ポリシーの条件要素でタグ情報を提供します。

> **Important**
>
> Lightsail では、一部の API アクションのタグに基づく認可はサポートされていません。詳細については、「リソースレベルのアクセス許可およびタグに基づく承認のサポート」を参照してください。

Lightsail リソースのタグ付けの詳細については、「タグ」を参照してください。

リソース上のタグに基づいてリソースへのアクセスを制限するためのアイデンティティベースのポリシーの例については、「タグに基づく Lightsail リソースの作成と削除の許可」を参照してください。

Lightsail IAM ロール

IAM 役割 は特定のアクセス許可を持つ、 AWS アカウント内のエンティティです。

Lightsail を使用した一時的な認証情報の使用

一時的な認証情報を使用して、フェデレーションでサインイン、IAM ロールを引き受ける、またはクロスアカウントロールを引き受けることができます。一時的なセキュリティ認証情報を取得するには、AssumeRole または GetFederationToken などの AWS STS API オペレーションを呼び出します。

Lightsail では、一時認証情報の使用をサポートしています。

サービスリンクロール

サービスにリンクされたロールを使用すると、 AWS サービスは他の サービスのリソースにアクセスして、ユーザーに代わってアクションを実行できます。サービスリンクロールは IAM アカウント内に表示され、サービスによって所有されます。IAM 管理者は、サービスリンクロールの許可を表示できますが、編集することはできません。

Lightsail はサービスにリンクされたロールをサポートします。Lightsail サービスにリンクされたロールの作成または管理の詳細については、「サービスにリンクされたロール」を参照してください。

サービスロール

Lightsail はサービスロールをサポートしていません。

トピック

- Lightsail の IAM アイデンティティポリシーで最小特権のアクセス許可を付与する
- IAM ポリシーを使用して特定の Lightsail リソースへのアクセスを許可する
- Amazon Lightsail のサービスにリンクされたロールの使用
- IAM ポリシーを使用して Lightsail バケットを管理する

Lightsail の IAM アイデンティティポリシーで最小特権のアクセス許可を付与する

デフォルトでは、IAM ユーザーおよびロールには、Lightsail リソースを作成または変更するアクセス許可はありません。また、 AWS マネジメントコンソール、 AWS CLI、または AWS API を使用してタスクを実行することはできません。IAM 管理者は、ユーザーとロールに必要な、指定されたリ

ソースで特定の API オペレーションを実行する権限をユーザーとロールに付与する IAM ポリシーを作成する必要があります。続いて、管理者はそれらの権限が必要な IAM ユーザーまたはグループにそのポリシーをアタッチする必要があります。

JSON ポリシードキュメントのこれらの例を使用して、IAM アイデンティティベースのポリシーを作成する方法については、「IAM ユーザーガイド」の「JSON タブでのポリシーの作成」を参照してください。

ポリシーのベストプラクティス

ID ベースのポリシーは、ユーザーのアカウントで誰かが Amazon Lightsail リソースを作成、アクセス、または削除できるどうかを決定します。これらのアクションを実行すると、 AWS アカウントに料金が発生する可能性があります。アイデンティティベースポリシーを作成したり編集したりする際には、以下のガイドラインと推奨事項に従ってください:

- AWS 管理ポリシーを開始し、最小特権のアクセス許可に移行する – ユーザーとワークロードにアクセス許可の付与を開始するには、多くの一般的なユースケースにアクセス許可を付与するAWS 管理ポリシーを使用します。これらは で使用できます AWS アカウント。ユースケースに固有の AWS カスタマー管理ポリシーを定義することで、アクセス許可をさらに減らすことをお勧めします。詳細については、IAM ユーザーガイド の AWS マネージドポリシー または ジョブ機能のAWS マネージドポリシー を参照してください。
- 最小特権を適用する – IAM ポリシーでアクセス許可を設定する場合は、タスクの実行に必要な許可のみを付与します。これを行うには、特定の条件下で特定のリソースに対して実行できるアクションを定義します。これは、最小特権アクセス許可とも呼ばれています。IAM を使用して許可を適用する方法の詳細については、IAM ユーザーガイド の IAM でのポリシーとアクセス許可 を参照してください。
- IAM ポリシーで条件を使用してアクセスをさらに制限する - ポリシーに条件を追加して、アクションやリソースへのアクセスを制限できます。たとえば、ポリシー条件を記述して、すべてのリクエストを SSL を使用して送信するように指定できます。条件を使用して、サービスアクションが などの特定の を通じて使用されている場合に AWS のサービス、サービスアクションへのアクセスを許可することもできます CloudFormation。詳細については、IAM ユーザーガイド の IAM JSON ポリシー要素:条件 を参照してください。
- IAM アクセスアナライザー を使用して IAM ポリシーを検証し、安全で機能的な権限を確保する - IAM アクセスアナライザー は、新規および既存のポリシーを検証して、ポリシーが IAM ポリシー言語 (JSON) および IAM のベストプラクティスに準拠するようにします。IAM アクセスアナライザーは 100 を超えるポリシーチェックと実用的な推奨事項を提供し、安全で機能的なポリシーの作成をサポートします。詳細については、IAM ユーザーガイド の IAM Access Analyzer でポリシーを検証する を参照してください。

- 多要素認証 (MFA) を要求する – で IAM ユーザーまたはルートユーザーを必要とするシナリオがある場合は AWS アカウント、MFA をオンにしてセキュリティを強化します。API オペレーションが呼び出されるときに MFA を必須にするには、ポリシーに MFA 条件を追加します。詳細については、IAM ユーザーガイド の MFA を使用した安全な API アクセス を参照してください。

IAM でのベストプラクティスの詳細については、IAM ユーザーガイド の IAM でのセキュリティのベストプラクティス を参照してください。

Lightsail コンソールの使用

Amazon Lightsail コンソールにアクセスするには、すべての Lightsail アクションとリソースに対するフルアクセス許可が必要です。これらのアクセス許可により、 AWS アカウントのLightsailリソースの詳細を一覧表示および表示できます。最小限必要なアクセス許可よりも制限されたアイデンティティベースのポリシーを作成すると (つまり、フルアクセスではない)、そのポリシーをもつエンティティ (IAM ユーザーまたはロール) に対してはコンソールが意図したとおりに機能しません。

これらのエンティティが Lightsail コンソールを使用できるように、以下の 次のポリシーもそれらのエンティティにアタッチします。詳細については、「IAM ユーザーガイド」の「ユーザーへのアクセス許可の追加」を参照してください。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lightsail:*"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS CLI または AWS API のみを呼び出すユーザーには、最小限のコンソールアクセス許可を付与する必要はありません。代わりに、実行しようとしている API オペレーションに一致するアクションのみへのアクセスが許可されます。

ユーザーが自分のアクセス許可を表示できるようにする

この例では、ユーザーアイデンティティにアタッチされたインラインおよびマネージドポリシーの表示を IAM ユーザーに許可するポリシーの作成方法を示します。このポリシーには、コンソールで、または AWS CLI または AWS API を使用してプログラムでこのアクションを実行するアクセス許可が含まれています。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

タグに基づく Lightsail リソースの作成と削除の許可

アイデンティティベースのポリシーの条件を使用して、タグに基づいて Lightsail リソースへのアクセスをコントロールできます。この例では、作成リクエストで `allow` のキータグと `true` の値が定義されていない限り、ユーザーが新しい Lightsail リソースを作成できないようにするポリシーを作成する方法を示します。このポリシーは、 `allow/true` のキーバリューのタグが定義されていない限り、ユーザーによるリソースの削除も禁止します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lightsail:Create*",
                "lightsail:TagResource",
                "lightsail:UntagResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/allow": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "lightsail:Delete*",
                "lightsail:TagResource",
                "lightsail:UntagResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/allow": "true"
                }
            }
        }
    ]
```

```
}
```

次のポリシーでは、キーと値のタグが allow/false ではないリソースのタグの変更をユーザーに禁止します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "lightsail:TagResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:ResourceTag/allow": "false"
                }
            }
        }
    ]
}
```

これらのポリシーは、アカウントの IAM ユーザーにアタッチできます。詳細については、「IAM ユーザーガイド」の「IAM JSON ポリシー要素: 条件」を参照してください。

IAM ポリシーを使用して特定の Lightsail リソースへのアクセスを許可する

「リソースレベルのアクセス許可」とは、ユーザーがアクションを実行可能なリソースを指定できることを意味します。Amazon Lightsail では、リソースレベルのアクセス許可がサポートされます。これは、特定の Lightsail アクションでは、満たす必要がある条件、またはユーザーが使用したり、編集したりできる特定のリソースに基づいて、ユーザーがそれらのアクションをいつ使用できるかを制御できることを意味します。たとえば、特定の Amazon リソースネーム (ARN) を使用して、インスタンスまたはデータベースを管理するアクセス許可をユーザーに付与することができます。

> **Important**
>
> Lightsail は、一部の API アクションにはリソースレベルのアクセス許可をサポートしません。詳細については、「リソースレベルのアクセス許可およびタグに基づく承認のサポート」を参照してください。

Lightsail アクション、ARN および IAM ポリシーステートメントで使用できる Lightsail 条件キーによって作成または変更されるリソースの詳細については、IAM ユーザーガイドの「Actions, Resources, and Condition Keys for Amazon Lightsail 」を参照してください。

特定のインスタンスの管理を許可する

次のポリシーでは、インスタンスの再起動/開始/停止、インスタンスポートの管理、特定のインスタンスのインスタンススナップショットの作成へのアクセス権を付与します。また、Lightsail アカウント内の他のインスタンス関連の情報やリソースへの読み取り専用アクセスも提供します。ポリシーで、 *InstanceARN* をインスタンスの Amazon リソースネーム ((ARN) に置き換えます。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "lightsail:GetActiveNames",
                "lightsail:GetAlarms",
                "lightsail:GetAutoSnapshots",
                "lightsail:GetBlueprints",
                "lightsail:GetBundles",
                "lightsail:GetCertificates",
                "lightsail:GetCloudFormationStackRecords",
                "lightsail:GetContactMethods",
                "lightsail:GetDisk",
                "lightsail:GetDisks",
                "lightsail:GetDiskSnapshot",
                "lightsail:GetDiskSnapshots",
                "lightsail:GetDistributionBundles",
                "lightsail:GetDistributionLatestCacheReset",
```

```
                "lightsail:GetDistributionMetricData",
                "lightsail:GetDistributions",
                "lightsail:GetDomain",
                "lightsail:GetDomains",
                "lightsail:GetExportSnapshotRecords",
                "lightsail:GetInstance",
                "lightsail:GetInstanceAccessDetails",
                "lightsail:GetInstanceMetricData",
                "lightsail:GetInstancePortStates",
                "lightsail:GetInstances",
                "lightsail:GetInstanceSnapshot",
                "lightsail:GetInstanceSnapshots",
                "lightsail:GetInstanceState",
                "lightsail:GetKeyPair",
                "lightsail:GetKeyPairs",
                "lightsail:GetLoadBalancer",
                "lightsail:GetLoadBalancerMetricData",
                "lightsail:GetLoadBalancers",
                "lightsail:GetLoadBalancerTlsCertificates",
                "lightsail:GetOperation",
                "lightsail:GetOperations",
                "lightsail:GetOperationsForResource",
                "lightsail:GetRegions",
                "lightsail:GetRelationalDatabase",
                "lightsail:GetRelationalDatabaseBlueprints",
                "lightsail:GetRelationalDatabaseBundles",
                "lightsail:GetRelationalDatabaseEvents",
                "lightsail:GetRelationalDatabaseLogEvents",
                "lightsail:GetRelationalDatabaseLogStreams",
                "lightsail:GetRelationalDatabaseMetricData",
                "lightsail:GetRelationalDatabaseParameters",
                "lightsail:GetRelationalDatabases",
                "lightsail:GetRelationalDatabaseSnapshot",
                "lightsail:GetRelationalDatabaseSnapshots",
                "lightsail:GetStaticIp",
                "lightsail:GetStaticIps",
                "lightsail:IsVpcPeered"
            ],
            "Resource": "*"
        },
        {
            "Sid": "VisualEditor2",
            "Effect": "Allow",
            "Action": [
```

```
                "lightsail:CloseInstancePublicPorts",
                "lightsail:CreateInstanceSnapshot",
                "lightsail:OpenInstancePublicPorts",
                "lightsail:PutInstancePublicPorts",
                "lightsail:RebootInstance",
                "lightsail:StartInstance",
                "lightsail:StopInstance"
            ],
            "Resource": "arn:aws:lightsail:us-
east-2:123456789012:Instance/244ad76f-8aad-4741-809f-12345EXAMPLE"
        }
    ]
}
```

インスタンスの ARN を取得するには、`GetInstance` Lightsail API アクションを使用し、`instanceName` パラメータを使用してインスタンスの名前を指定します。インスタンス ARN は、次の例に示すように、そのアクションの結果に一覧表示されます。詳細については、Amazon Lightsail API リファレンスの GetInstance を参照してください。

```
C:\>aws lightsail get-instance --instance-name WordPress-1
{
    "instance": {
        "arn": "arn:aws:lightsail:us-west-2:138[redacted]1:Instance/1361427a-3982-[redacted]-98c5-[redacted]5591fcd",
        "createdAt": 1581469097.179,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "resourceType": "Instance",
        "tags": [],
        "blueprintId": "wordpress",
        "blueprintName": "WordPress",
        "bundleId": "nano_2_0",
```

特定のデータベースの管理を許可する

次のポリシーは、特定のデータベースの再起動/開始/停止および更新へのアクセス権を付与します。また、Lightsail アカウント内の他のデータベース関連の情報やリソースへの読み取り専用アクセスも提供します。ポリシーで、*DatabaseARN* をデータベースの Amazon リソースネーム (ARN) に置き換えます。

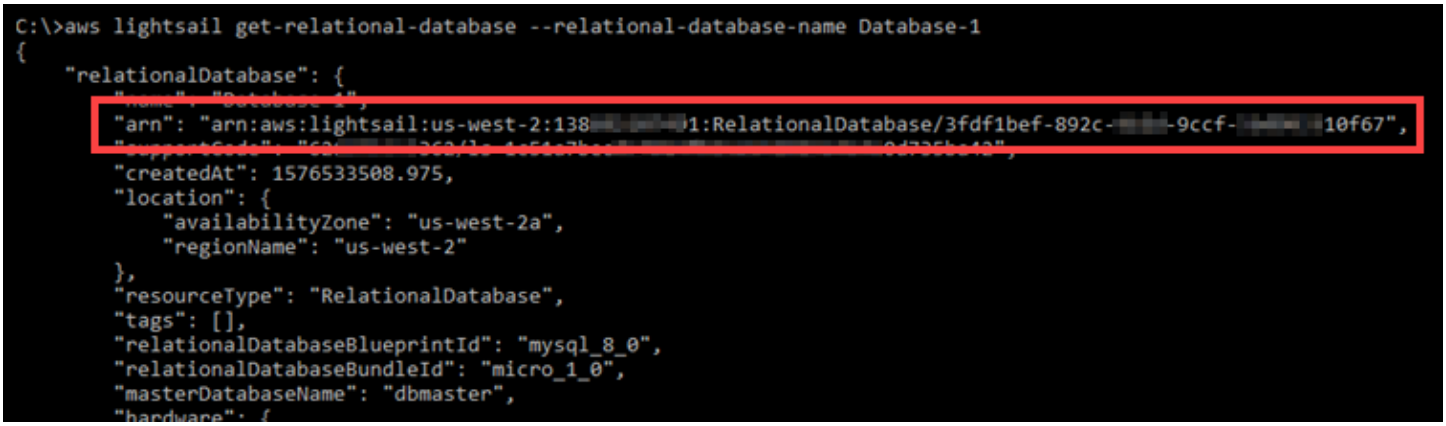
JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "lightsail:GetActiveNames",
                "lightsail:GetAlarms",
                "lightsail:GetAutoSnapshots",
                "lightsail:GetBlueprints",
                "lightsail:GetBundles",
                "lightsail:GetCertificates",
                "lightsail:GetCloudFormationStackRecords",
                "lightsail:GetContactMethods",
                "lightsail:GetDisk",
                "lightsail:GetDisks",
                "lightsail:GetDiskSnapshot",
                "lightsail:GetDiskSnapshots",
                "lightsail:GetDistributionBundles",
                "lightsail:GetDistributionLatestCacheReset",
                "lightsail:GetDistributionMetricData",
                "lightsail:GetDistributions",
                "lightsail:GetDomain",
                "lightsail:GetDomains",
                "lightsail:GetExportSnapshotRecords",
                "lightsail:GetInstance",
                "lightsail:GetInstanceAccessDetails",
                "lightsail:GetInstanceMetricData",
                "lightsail:GetInstancePortStates",
                "lightsail:GetInstances",
                "lightsail:GetInstanceSnapshot",
                "lightsail:GetInstanceSnapshots",
                "lightsail:GetInstanceState",
                "lightsail:GetKeyPair",
                "lightsail:GetKeyPairs",
                "lightsail:GetLoadBalancer",
                "lightsail:GetLoadBalancerMetricData",
                "lightsail:GetLoadBalancers",
                "lightsail:GetLoadBalancerTlsCertificates",
                "lightsail:GetOperation",
                "lightsail:GetOperations",
                "lightsail:GetOperationsForResource",
                "lightsail:GetRegions",
                "lightsail:GetRelationalDatabase",
```

```
                "lightsail:GetRelationalDatabaseBlueprints",
                "lightsail:GetRelationalDatabaseBundles",
                "lightsail:GetRelationalDatabaseEvents",
                "lightsail:GetRelationalDatabaseLogEvents",
                "lightsail:GetRelationalDatabaseLogStreams",
                "lightsail:GetRelationalDatabaseMetricData",
                "lightsail:GetRelationalDatabaseParameters",
                "lightsail:GetRelationalDatabases",
                "lightsail:GetRelationalDatabaseSnapshot",
                "lightsail:GetRelationalDatabaseSnapshots",
                "lightsail:GetStaticIp",
                "lightsail:GetStaticIps",
                "lightsail:IsVpcPeered"
            ],
            "Resource": "*"
        },
        {
            "Sid": "VisualEditor2",
            "Effect": "Allow",
            "Action": [
                "lightsail:RebootRelationalDatabase",
                "lightsail:StartRelationalDatabase",
                "lightsail:StopRelationalDatabase",
                "lightsail:UpdateRelationalDatabase"
            ],
            "Resource": "arn:aws:lightsail:us-
east-2:123456789012:RelationalDatabase/244ad76f-8aad-4741-809f-12345EXAMPLE"
        }
    ]
}
```

データベースの ARN を取得するには、`GetRelationalDatabase` Lightsail API アクションを使用し、`relationalDatabaseName` パラメータを使用してデータベースの名前を指定します。データベース ARN は、次の例に示すように、そのアクションの結果に一覧表示されます。詳細については、Amazon Lightsail API リファレンス の GetBucketEncryption を参照してください。



Amazon Lightsail のサービスにリンクされたロールの使用

Amazon Lightsail は AWS Identity and Access Management (IAM) サービスにリンクされたロールを使用します。サービスリンクロールは、Amazon Lightsail に直接リンクされた一意のタイプの IAM ロールです。サービスにリンクされたロールは によって事前定義Amazon Lightsailされており、 がユーザーに代わって他の AWS サービスを呼び出すLightsailために必要なすべてのアクセス許可が含まれています。

サービスリンクロールを使用すると、必要なアクセス許可を手動で追加する必要がなくなるため、Amazon Lightsail の設定が簡単になります。このサービスリンクロールのアクセス許可は Amazon Lightsail で定義します。特に定義されている場合を除き、Amazon Lightsail のみがその ロールを引き受けることができます。定義されたアクセス許可には、他の IAM エンティティにアタッチできない信頼ポリシーとアクセス許可ポリシーが含まれます。

サービスリンクロールを削除するには、最初に関連リソースを削除する必要があります。これにより、リソースへの意図しないアクセスによる権限の削除が防止され、Amazon Lightsail リソースは保護されます。

サービスリンクロールをサポートする他のサービスについては、「IAM と連携する AWS サービス」で「サービスリンクロール」列が「はい」になっているサービスを探してください。サービスにリンクされたロールに関するサービスのドキュメントを表示するには、「はい」のリンクをクリックします。

Amazon Lightsail のサービスにリンクされたロールのアクセス許可

Amazon Lightsail は、AWSServiceRoleForLightsail という名前のサービスにリンクされたロールを使用します – Lightsailインスタンスとブロックストレージディスクスナップショットを Amazon Elastic Compute Cloud (Amazon EC2) にエクスポートし、Amazon Simple Storage Service (Amazon S3) から現在のアカウントレベルのブロックパブリックアクセス設定を取得します。

AWSServiceRoleForLightsail サービスにリンクされたロールは、次のサービスを信頼してロールを引き受けます。

- `lightsail.amazonaws.com`

ロールのアクセス許可ポリシーは、指定したリソースに対して以下のアクションを実行することを Amazon Lightsail に許可します。

- アクション: すべての AWS リソース`ec2:CopySnapshot`で。
- アクション: すべての AWS リソース`ec2:DescribeSnapshots`で。
- アクション: すべての AWS リソース`ec2:CopyImage`で。
- アクション: すべての AWS リソース`ec2:DescribeImages`で。
- アクション: すべての AWS CloudFormation スタック`cloudformation:DescribeStacks`で。
- アクション: すべての AWS リソース`s3:GetAccountPublicAccessBlock`で。

サービスリンクロールのアクセス許可

IAM; エンティティ (ユーザー、グループ、ロールなど) がサービスにリンクされたロールの説明を作成または編集できるようにするには、アクセス許可を設定する必要があります。

特定のサービスにリンクされたロールの作成を IAM エンティティに許可するには

サービスにリンクされたロールを作成する必要のある IAM エンティティに、次のポリシーを追加します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
lightsail.amazonaws.com/AWSServiceRoleForLightsail*",
            "Condition": {"StringLike": {"iam:AWSServiceName":
 "lightsail.amazonaws.com"}}
        },
```

```
        {
            "Effect": "Allow",
            "Action": "iam:PutRolePolicy",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
lightsail.amazonaws.com/AWSServiceRoleForLightsail*"
        }
    ]
}
```

IAM エンティティがサービスにリンクされた任意のロールを作成することを許可するには

サービスにリンクされたロール、または必要なポリシーを含む任意のサービスロールを作成する必要のある IAM エンティティのアクセス許可ポリシーに、次のステートメントを追加します。このポリシーにより、ロールにポリシーがアタッチされます。

```
{
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "arn:aws:iam::*:role/aws-service-role/*"
}
```

IAM エンティティが任意のサービスロールの説明を編集することを許可するには

サービスにリンクされたロール、または任意のサービスロールの説明を編集する必要のある IAM エンティティのアクセス許可ポリシーに、次のステートメントを追加します。

```
{
    "Effect": "Allow",
    "Action": "iam:UpdateRoleDescription",
    "Resource": "arn:aws:iam::*:role/aws-service-role/*"
}
```

IAM エンティティがサービスにリンクされた特定のロールを削除することを許可するには

サービスにリンクされたロールを削除する必要のある IAM エンティティのアクセス許可ポリシーに、次のステートメントを追加します。

```
{
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/lightsail.amazonaws.com/
AWSServiceRoleForLightsail*"
}
```

IAM エンティティがサービスロールを削除することを許可するには

サービスにリンクされたロール、または任意のサービスロールを削除する必要のある IAM; エンティティのアクセス許可ポリシーに、次のステートメントを追加します。

```
{
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/*"
}
```

または、 AWS マネージドポリシーを使用して、サービスへのフルアクセスを提供することもできます。

Amazon Lightsail のサービスにリンクされたロールの作成

サービスリンクロールを手動で作成する必要はありません。Lightsail インスタンスまたはブロックストレージディスクスナップショットを Amazon EC2 にエクスポートするか、 AWS AWS マネジメントコンソール、 AWS CLI、または AWS API でLightsailバケットを作成または更新すると、 によってサービスにリンクされたロールAmazon Lightsailが作成されます。

このサービスにリンクされたロールを削除した後に、そのロールを再作成する必要がある場合は、同じプロセスを使用してアカウントでロールを再作成することができます。Lightsail インスタンスまたはブロックストレージのディスクスナップショットを Amazon EC2 にエクスポートした

り、Lightsail バケットを作成または更新したりすると、Amazon Lightsail には、サービスにリンクされたロールが作成されます。

> **Important**
>
> サービスにリンクされたロールの作成を Amazon Lightsail に許可するように IAM アクセス許可を設定する必要があります。これを行うには、次の「サービスにリンクされたロールのアクセス許可」セクションのステップを実行します。

Amazon Lightsail のサービスにリンクされたロールの編集

Amazon Lightsail では、AWSServiceRoleForLightsail サービスにリンクされたロールを編集することはできません。サービスリンクロールの作成後は、さまざまなエンティティがロールを参照する可能性があるため、ロール名を変更することはできません。ただし、IAM を使用してロールの説明を編集することはできます。詳細については、「IAM ユーザーガイド」の「サービスにリンクされたロールの編集」を参照してください。

Amazon Lightsail のサービスにリンクされたロールの削除

サービスリンクロールが必要な機能またはサービスが不要になった場合には、そのロールを削除することをお勧めします。そうすることで、モニタリングや保守が積極的に行われていない未使用のエンティティを排除できます。ただし、AWSServiceRoleForLightsail サービスにリンクされたロールを削除する前に、保留中のコピー状態のAmazon Lightsailインスタンスまたはディスクスナップショットがないことを確認する必要があります。詳細については、「スナップショットを Amazon EC2 にエクスポートする」を参照してください。

サービスリンクロールを IAM で手動削除するには

IAM コンソール、 AWS CLI、または AWS API を使用して、AWSServiceRoleForLightsail サービスにリンクされたロールを削除します。詳細については、IAM ユーザーガイドの「サービスにリンクされたロールの削除」を参照してください。

Amazon Lightsail のサービスにリンクされたロールをサポートするリージョン

Amazon Lightsail は、サービスを利用できるすべてのリージョンで、サービスにリンクされたロールの使用をサポートします。Lightsail が使用できるリージョンの詳細については、「Amazon Lightsail リージョン」 を参照してください。

IAM ポリシーを使用して Lightsail バケットを管理する

以下のポリシーは、Amazon Lightsail オブジェクトストレージサービスにて特定のバケットを管理するためのアクセス権限をユーザーに付与します。このポリシーは、Lightsailコンソール、 AWS Command Line Interface (AWS CLI)、 AWS API、 AWS SDKs を介してバケットへのアクセスを許可します。ポリシーで、*<BucketName>*を管理するバケット名に置き換えます。IAM ポリシーの詳細については、「AWS Identity and Access Management ユーザーガイド」の「IAM ポリシーの作成」を参照してください。IAM ユーザーとユーザーグループの作成の詳細は、「AWS Identity and Access Management ユーザーガイド」の「最初の IAM 委任ユーザーとユーザーグループの作成」を参照してください。

> **Important**
>
> このポリシーを持たないユーザーは、Lightsail コンソールのバケット管理ページにあるオブジェクトタブを表示する際にエラーが発生します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "LightsailAccess",
            "Effect": "Allow",
            "Action": "lightsail:*",
            "Resource": "*"
        },
        {
            "Sid": "S3BucketAccess",
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::<BucketName>/*",
                "arn:aws:s3:::<BucketName>"
            ]
        }
    ]
}
```

バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。
3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

   バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

   - Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
   - Amazon Lightsail でのバケットのアクセス許可の設定
   - Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
   - Amazon Lightsail でのバケットのアクセスキーの作成
   - Amazon Lightsail でのバケットのリソースアクセスの設定
   - Amazon Lightsail でのバケットのクロスアカウントアクセスの設定
5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。

   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定

6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail のバケットにファイルをアップロードする
   - マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
   - Amazon Lightsail のバケット内のオブジェクトの表示
   - Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
   - Amazon Lightsail のバケットからのオブジェクトのダウンロード
   - Amazon Lightsail のバケット内のオブジェクトのフィルタリング
   - Amazon Lightsail のバケット内のオブジェクトのタグ付け
   - Amazon Lightsail のバケット内のオブジェクトの削除
9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。
11. バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。
12. バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。
13. ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。
14. バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。
   - チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
   - チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用

15.使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

IAM ユーザーに Lightsail アクセスを付与する

AWS アカウントのルートユーザー、または管理者権限を持つ AWS Identity and Access Management (IAM) ユーザーとして、 AWS アカウントに 1 つ以上の IAM ユーザーを作成し、それらのユーザーを が提供するサービスへのさまざまなレベルのアクセスで設定できます AWS。

Amazon Lightsail では、Lightsail にのみアクセスできる IAM ユーザーを作成するとよいでしょう。これは、Lightsailリソースを表示、作成、編集、または削除するためのアクセスを必要とするが、が提供する他のサービスへのアクセスを必要としないユーザーがチームに参加するときに行います AWS。これを設定するにはまず、Lightsail へのアクセス許可を付与する IAM ポリシーを作成する必要があります。その後、IAM グループを作成し、ポリシーをそのグループにアタッチします。その後、IAM ユーザーを作成してグループのメンバーにします。これにより、Lightsail へのアクセス許可が付与されます。

誰かがチームを去るとき、たとえばその人物が、チームを離れるが引き続き会社に勤める場合、Lightsail アクセスグループからそのユーザーを削除して、Lightsail へのアクセスを取り消すことができます。あるいは、たとえばユーザーが退社して今後アクセス権限を必要としない場合、IAM からそのユーザーを削除できます。

> **Warning**
>
> このシナリオでは、プログラムによるアクセスと長期的な認証情報を持つ IAM ユーザーが必要です。これはセキュリティ上のリスクをもたらします。このリスクを軽減するために、これらのユーザーにはタスクの実行に必要な権限のみを付与し、不要になったユーザーは削除することをお勧めします。アクセスキーは、必要に応じて更新できます。詳細については、「IAM ユーザーガイド」の「アクセスキーの更新」を参照してください。

目次

- Lightsail アクセス用の IAM ポリシーを作成する
- Lightsail アクセス許可の IAM グループを作成し、Lightsail アクセスポリシーをアタッチする
- IAM ユーザーを作成して、そのユーザーを Lightsail アクセスグループに追加する

Lightsail アクセス用の IAM ポリシーを作成する

Lightsail アクセス権のための IAM ポリシーを作成するには、以下の手順に従います。詳細については、IAM ドキュメントの IAM ポリシーの作成を参照してください。

1. IAM コンソールにサインインします。
2. 左のナビゲーションペインの [ポリシー] を選択します。
3. [ポリシーを作成] を選択します。
4. [Create Policy (ポリシーの作成)] ページで、[JSON] タブを選択します。

5. テキストボックスの内容をハイライトしてから、次のポリシー構成テキストをコピーして貼り付けます。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lightsail:*"
            ],
            "Resource": "*"
        }
    ]
}
```

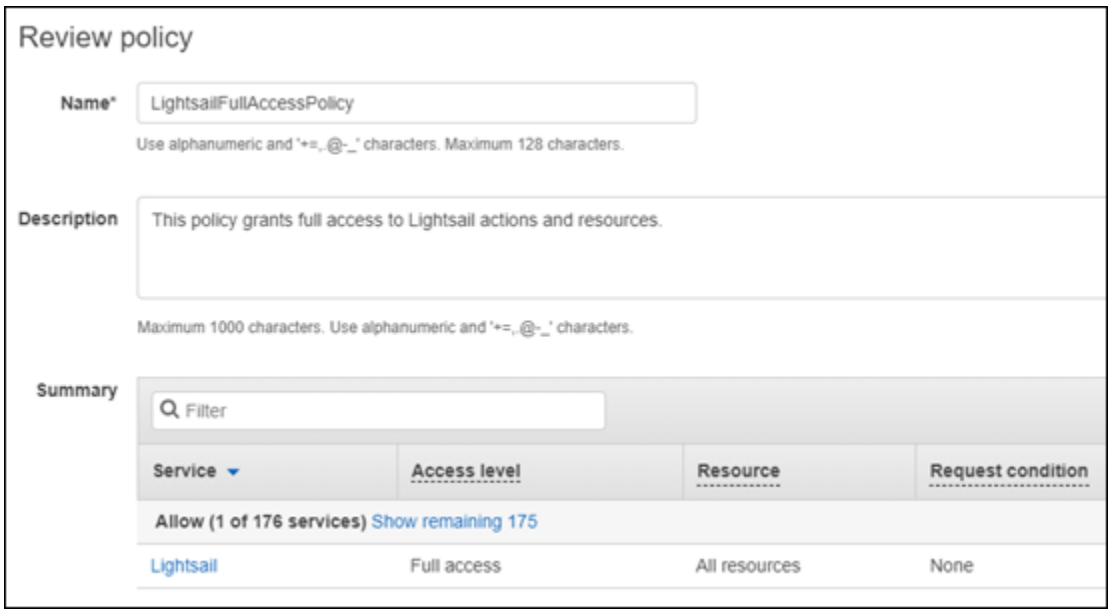
   結果は次の例のようになります。

これにより、すべての Lightsail アクションおよびリソースへのアクセス許可を付与します。VPC ピアリングの有効化 AWS、Amazon EC2 へのLightsailスナップショットのエクスポート、 を使用した Amazon EC2 リソースの作成など、 が提供する他のサービスへのアクセスを必要とするアクションにはLightsail、このポリシーに含まれていない追加のアクセス許可が必要です。詳細については、以下のガイドを参照してください。

- の外部の AWS リソースを操作するように Amazon VPC ピアリングを設定する Amazon Lightsail
- Amazon Lightsail のスナップショットを Amazon EC2 にエクスポートする
- Lightsail でエクスポートしたスナップショットから Amazon EC2 インスタンスを作成する

アクション固有およびリソース固有の許可が与えられる例については、Amazon Lightsail リソースレベルの許可ポリシー例 を参照してください。

6. ポリシーのレビュー を選択します。
7. [ポリシーの確認] ページで、ポリシー名を選択します。分かりやすい名前 (例: `LightsailFullAccessPolicy`) をつけます。
8. 説明を追加し、ポリシー設定を確認します。変更が必要な場合は、[戻る] を選択してポリシーを変更します。



9. ポリシーの設定が正しいことを確認したら、[Create Policy (ポリシーの作成)] を選択します。

   これでポリシーが作成され、既存の IAM グループに追加することも、このガイドの次のセクションの手順に従って、新しい IAM グループを作成することもできます。

Lightsail アクセス許可の IAM グループを作成し、Lightsail アクセスポリシーをアタッチする

Lightsail アクセス許可のために IAM グループを作成し、本ガイドの前出のセクションで作成した Lightsail アクセスポリシーをアタッチするには、この手順に従います。詳細については、IAM ドキュメント内にある「IAM グループの作成」および「IAM グループへのポリシーのアタッチ」を参照してください。

1. IAM コンソールの左側のナビゲーションペインで [グループ] を選択します。
2. [Create New Group (新しいグループの作成)] を選択します。
3. [Set Group Name (グループ名の設定)] ページで、グループを選択します。分かりやすい名前 (例: `LightsailFullAccessGroup`) をつけます。
4. [Attach Policy (ポリシーのアタッチ)] ページで、本ガイドで作成した Lightsail ポリシー (例: `LightsailFullAccessPolicy`) を検索します。
5. ポリシーの横にチェックマークを追加し、[Next step (次のステップ)] を選択します。
6. グループの設定を確認します。変更が必要な場合は、[戻る] を選択してグループのポリシーを変更します。

7. グループの設定が正しいことを確認したら、[グループの作成] を選択します。

   これでグループが作成され、グループに追加されたユーザーは Lightsail のアクションとリソースにアクセスできるようになります。本ガイドの次のセクションのステップに従って、既存の IAM ユーザーをグループに追加するか、新しい IAM ユーザーを作成することができます。

IAM ユーザーを作成して、そのユーザーを Lightsail アクセスグループに追加する

IAM ユーザーを作成して、そのユーザーを Lightsail アクセスグループに追加するには、次のステップに従います。詳細については、IAM ドキュメントの「AWS アカウントで IAM ユーザーを作成する」および「IAM グループでユーザーを追加または削除する」を参照してください。

1. IAM コンソールの左側のナビゲーションペインで、[ユーザー] を選択します。
2. [ユーザーを追加] を選択します。
3. ページの [Set user details (ユーザー詳細の設定)] セクションで、ユーザー名をつけます。
4. ページの AWS アクセスタイプの選択セクションで、次のオプションから選択します。

   a. プログラムによるアクセスを選択すると、 AWS API、CLI、SDK、およびその他の開発ツールのアクセスキー ID とシークレットアクセスキーが有効になり、Lightsailアクションとリソースに使用できます。詳細については、「 を使用する AWS CLI ように Lightsailを設定する」を参照してください。

   b. AWS マネジメントコンソールアクセスを選択して、ユーザーが AWS マネジメントコンソール、つまりLightsailコンソールにサインインできるようにするパスワードを有効にします。このオプションが選択されたとき、次のパスワードオプションが表示されます。

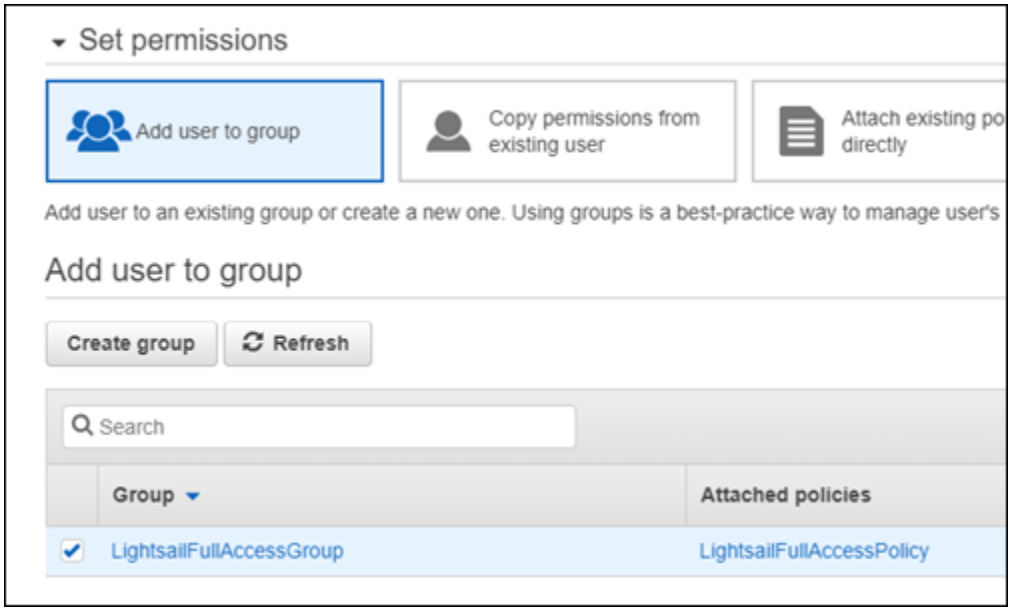
      i. [自動生成パスワード] を選択すると IAM がパスワードを生成し、または、[カスタムパスワード] を選択すると独自のパスワードを入力できます。

      ii. [Require password reset (パスワードのリセットが必要)] を選択すると、次回のログイン時にユーザーが新しいパスワードを作成します (パスワードをリセットする)。

   > **Note**
   >
   > プログラムによるアクセスオプションのみを選択した場合、ユーザーは AWS コンソールと Lightsailコンソールにサインインできません。

5. [Next: Permissions] (次のステップ: 許可) を選択します。

6. ページの [Set permissions (許可を設定)] セクションで [ユーザーをグループに追加] を選択して、本ガイドの前半で作成した Lightsail アクセスグループ (例: `LightsailFullAccessGroup`) を選択します。



7. [Next: Tags] (次へ: タグ) を選択します。
8. (オプション) タグをキーバリューペアとしてアタッチして、メタデータをユーザーに追加します。IAM でのタグの仕様について詳細は、「IAM エンティティのタグ付け」を参照してください。
9. [次へ: レビュー] を選択します。
10. ユーザー設定を確認します。変更が必要な場合は、[戻る] を選択してユーザーのグループまたはポリシーを変更します。
11. ユーザーの設定が正しいことを確認したら、[ユーザーの作成] を選択します。

    ユーザーが作成され、作成されたユーザーに Lightsail へのアクセスが付与されます。ユーザーの Lightsail アクセスを取り消すには、Lightsail アクセスグループからユーザーを削除します。詳細については、IAM ドキュメントの「IAM グループへのユーザーの追加と削除」を参照してください。
12. ユーザーの認証情報を取得するには、以下のオプションを選択します。

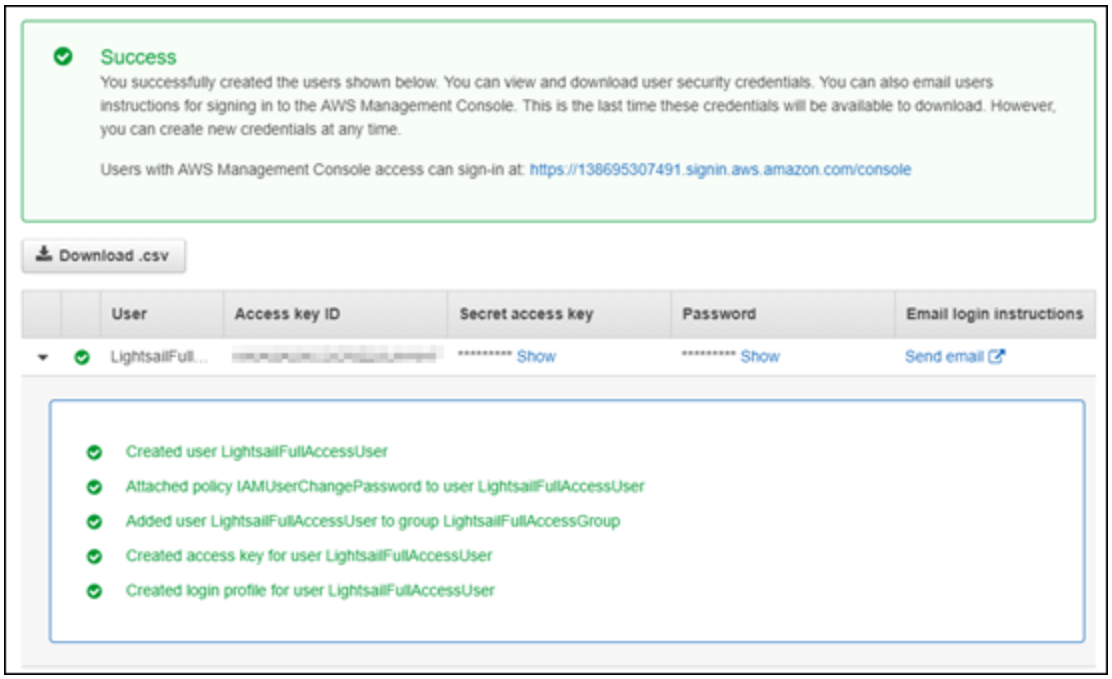
    a. Download .csv を選択して、アカウントのユーザー名、パスワード、アクセスキー ID、シークレットアクセスキー、 AWS コンソールログインリンクを含むファイルをダウンロードします。

b. [シークレットアクセスキー] で [表示] を選択すると、プログラム (AWS API、CLI、SDK、その他の開発ツール) を使用して Lightsail にアクセスするのに使用できるアクセスキーが表示されます。

> **Important**
>
> これはシークレットアクセスキーを表示またはダウンロードする唯一の機会であり、ユーザーが AWS API を使用する前にこの情報を提供する必要があります。ユーザーの新しいアクセスキー ID とシークレットアクセスキーは、安全な場所に保存してください。このステップを行った後に、シークレットキーに再度アクセスすることはできません。

c. ユーザーのパスワードが IAM によって生成されている場合、[パスワード] で [表示] を選択するとユーザーのパスワードが表示されます。ユーザーが初回サインインできるように、ユーザーにパスワードを提供する必要があります。

d. [E メールを送信する] を選択すると、Lightsail へのアクセス許可が付与されたことを知らせる E メールがユーザーに送られます。



アップデート管理で Lightsail インスタンスとコンテナの安全を保障する

アマゾン ウェブ サービス (AWS)Amazon Lightsail、、およびサードパーティーアプリケーションベンダーは、 で利用可能なインスタンスイメージ (ブループリントとも呼ばれます) を定期的に更新およびパッチ適用しますLightsail。 AWS インスタンスの作成後にオペレーティングシステムまたはアプリケーションを更新またはパッチLightsail適用することはありません。Lightsailまた、は、Lightsailコンテナサービスで設定したオペレーティングシステムおよびソフトウェアを更新またはパッチ適用しません。したがって、Amazon Lightsail インスタンスのオペレーティングシステムやアプリケーションに対するパッチ適用やそれらの更新およびセキュリティ確保を定期的に行うことが推奨されます。詳細については、AWS 責任共有モデルを参照してください。

インスタンスブループリントソフトウェアのサポート

以下の Amazon Lightsail プラットフォームとブループリントのリストは、各ベンダーのサポートページにリンクしています。そこで、ハウツーガイド、オペレーティングシステムとアプリケーションを最新の状態に保つなどの情報を表示できます。アプリケーションベンダーが提供している、自動更新サービスまたは推奨更新インストールプロセスを使用することもできます。

Windows

- Windows Server 2022、Windows Server 2019、および Windows Server 2016
- Microsoft SQL Server

Linux と Unix - オペレーティングシステムのみ

- Amazon Linux 2023
- Amazon Linux 2
- Ubuntu
- Debian
- FreeBSD
- openSUSE
- CentOS

Linux と Unix - オペレーティングシステムとアプリケーション

- Ubuntu の Plesk ホストスタック
- cPanel & WHM for Linux
- WordPress
- WordPress マルチサイト
- LAMP (PHP 8)
- Node.js
- Joomla!
- Magento
- MEAN
- Drupal
- GitLab CE
- Redmine
- Nginx
- Ghost
- Django
- PrestaShop

Amazon Lightsail リソースのコンプライアンスを検証する

AWS には、コンプライアンスに役立つ以下のリソースが用意されています。

- セキュリティとコンプライアンスのクイックスタートガイド – これらのデプロイガイドでは、アーキテクチャ上の考慮事項について説明し、セキュリティとコンプライアンスに重点を置いたベースライン環境をデプロイする手順について説明します AWS。
- AWS コンプライアンスリソース – このワークブックとガイドのコレクションは、お客様の業界や地域に適用される場合があります。
- 「 デベロッパーガイド」の「ルールによるリソースの評価」 – この AWS Config サービスは、リソース設定が内部プラクティス、業界ガイドライン、および規制にどの程度準拠しているかを評価します。 AWS Config
- AWS Security Hub CSPM – この AWS サービスは、 内のセキュリティ状態を包括的に把握 AWS し、セキュリティ業界標準とベストプラクティスへの準拠を確認するのに役立ちます。

インターフェイスエンドポイントを使用して Amazon Lightsail にアクセス (AWS PrivateLink)

AWS PrivateLink を使用して、VPC と Amazon Lightsail の間にプライベート接続を作成できます。インターネットゲートウェイ、NATデバイス、VPN接続、またはDirect Connect接続を使用せずに、VPC内にあるかのようにAmazon Lightsailにアクセスすることができます。VPC のインスタンスは、パブリック IP アドレスがなくても Amazon Lightsail にアクセスできます。

このプライベート接続を確立するには、AWS PrivateLink を利用したインターフェイスエンドポイントを作成します。インターフェイスエンドポイントに対して有効にする各サブネットにエンドポイントネットワークインターフェイスを作成します。これらは、Amazon Lightsail 宛てのトラフィックのエントリポイントとして機能するリクエスタ管理型ネットワークインターフェイスです。

詳細については、『AWS PrivateLink ガイド』の「AWS のサービス でアクセスする」を参照してください。

Amazon Lightsailに関する考慮事項

Amazon Lightsail のインターフェイスエンドポイントを設定する前に、仮想プライベートクラウド (VPC) を作成する必要があります。詳細については、「Amazon Virtual Private Cloud ユーザーガイド」の「VPC を作成する」を参照してください。さらに、「AWS PrivateLink ガイド」の「Considerations」も確認してください。

Amazon Lightsail は、インターフェイスエンドポイントを介してすべての API アクションの呼び出しをサポートしています。Lightsail で利用可能な API アクションの詳細については「Amazon Lightsail API reference」を参照してください。

Amazon Lightsail 用のインターフェイスエンドポイントの作成

Amazon Lightsail のインターフェイスエンドポイントは、Amazon VPC コンソールまたは AWS Command Line Interface (AWS CLI) を使用して作成できます。詳細については、「AWS PrivateLink ガイド」の「インターフェイスエンドポイントを作成」を参照してください。

以下のサービス名を使用して、Amazon Lightsail 用のインターフェース· エンドポイントを作成します。 :

```
com.amazonaws.region.lightsail
```

インターフェイスエンドポイントのプライベート DNS を有効にすると、リージョンのデフォルト DNS 名を使用して、Amazon Lightsail への API リクエストを実行できます。例えば、`lightsail.us-east-1.amazonaws.com`。使用できるリージョンコードについては、「Lightsail のリージョンとアベイラビリティーゾーン」を参照してください。

AWS CLI の例

インターフェイスエンドポイントを使用して Lightsail にアクセスするには、AWS CLI コマンドで `--region` および `--endpoint-url` パラメータを使用します。Lightsail で実行できるオペレーションの完全なリストについては、「Amazon Lightsail API Reference」の「Actions」を参照してください。

次の例では、AWS リージョン *us-east-1* と VPC エンドポイント ID の DNS 名 *vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com* をユーザー自身の情報に置き換えます。

例: エンドポイント URL を使用して Lightsail インスタンスを一覧表示する

次の例では、インターフェイスエンドポイントを使用してインスタンスを一覧表示します。

```
aws lightsail get-instances --region us-east-1 --endpoint-url
 https://vpce-1a2b3c4d-5e6f.lightsail.us-east-1.vpce.amazonaws.com
```

例: エンドポイント URL を使用して Lightsail ディスクを一覧表示する

次の例では、インターフェイスエンドポイントを使用してディスクを一覧表示します。

```
aws lightsail get-disks --region us-east-1 --endpoint-url
 https://vpce-1a2b3c4d-5e6f.lightsail.us-east-1.vpce.amazonaws.com
```

インターフェイスエンドポイントのエンドポイントポリシーを作成する

エンドポイントポリシーは、インターフェイスエンドポイントにアタッチできる IAM リソースです。デフォルトのエンドポイントポリシーでは、インターフェイスエンドポイントを介した Amazon Lightsail へのフルアクセスが許可されています。VPC から Amazon Lightsailに許可されるアクセスを制御するには、カスタム・エンドポイント・ポリシーをインターフェースのエンドポイントにアタッチします。

エンドポイントポリシーは以下の情報を指定します。

- アクションを実行できるプリンシパル (AWS アカウント、IAM ユーザー、IAM ロール)。
- 実行可能なアクション。
- このアクションを実行できるリソース。

詳細については、AWS PrivateLink ガイドのControl access to services using endpoint policies (エンドポイントポリシーを使用してサービスへのアクセスをコントロールする)を参照してください。

例: Amazon Lightsail アクション用の VPC エンドポイントポリシー

以下は、カスタムエンドポイントポリシーの例です。このポリシーをインターフェイスエンドポイントにアタッチすると、エンドポイントを介して Lightsail のブロックストレージディスクを削除するアクセス許可がすべてのユーザーに対して拒否され、他のすべての Lightsail アクションを実行するアクセス許可がすべてのユーザーに付与されます。

```
{
  "Statement": [
    {
      "Action": "lightsail:*",
      "Effect": "Allow",
      "Principal": "*",
      "Resource": "*"
    },
    {
      "Action": "lightsail:DeleteDisk",
      "Effect": "Deny",
      "Principal": "*",
      "Resource": "*"
    }
  ]
}
```

Lightsail リソースメトリクスをモニタリングする

Amazon Lightsail はインスタンス、データベース、ディストリビューション、ロードバランサー、コンテナサービス、およびバケットのメトリクスデータをチェック、収集し、それぞれのパフォーマンスをモニタリングします。時間の経過とともにベースラインを確立し、リソースのパフォーマンスに関する異常や問題をより簡単に検出できるようにアラームを設定できます。

Amazon Lightsail は、インスタンス、データベース、コンテンツ配信ネットワーク (CDN) ディストリビューション、ロードバランサー、コンテナサービス、バケットのメトリクスデータをレポートします。このデータは、Lightsail コンソールで表示およびモニタリングできます。モニタリングは、リソースの信頼性、可用性、パフォーマンスを維持する上で重要な部分です。リソースから定期的にメトリクスデータをモニタリングして収集し、マルチポイント障害が発生した場合に、より簡単にデバッグできるようにします。

目次

- リソースを効果的にモニタリングする
- メトリクスの概念と用語
- Lightsail の使用可能なメトリクス

リソースを効果的にモニタリングする

環境内の通常のリソースパフォーマンスのベースラインを確立する必要があります。さまざまな時間帯に、さまざまな負荷条件でパフォーマンスを測定します。リソースをモニタリングするときは、時間の経過に伴うリソースのパフォーマンスの履歴を書き留めて記録する必要があります。収集した履歴データに対して、リソースの現在のパフォーマンスを比較します。これにより、通常のパフォーマンスパターンとパフォーマンスの異常を特定し、それらに対処するための方法を考案することができます。

たとえば、インスタンスの CPU 使用率、ネットワーク使用率、ステータスチェックをモニタリングできます。確立したベースラインからパフォーマンスが外れた場合は、インスタンスの再設定または最適化を行って CPU 使用率の抑制、またはネットワークトラフィックの低減を行うことが必要な場合があります。インスタンスが CPU 使用率のしきい値を超えて動作し続ける場合は、インスタンスのより大きなプランに切り替えることができます (たとえば、7 米ドル/月プランではなく 5 米ドル/月プランを使用します)。インスタンスの新しいスナップショットを作成し、大きなプランを使用してスナップショットから新しいインスタンスを作成することで、より大きなプランに切り替えることができます。

ベースラインを確立したら、リソースが指定のしきい値を超えたときに通知するように Lightsail コンソールでアラームを設定できます。詳細については、「通知」および「アラーム」を参照してください。

メトリクスの概念と用語

次の用語と概念は、Lightsail でのメトリクスの使用をよりよく理解するのに役立ちます。

メトリクス

メトリクスは、時間順に並んだ一連のデータポイントを表します。メトリクスはモニタリング対象の変数と考え、データポイントは時間の経過と共に変数の値を表します。メトリクスは、名前によって一意に定義されます。たとえば、Lightsail によって提供される一部のインスタンスメトリクスには、CPU 使用率 (`CPUUtilization`)、着信ネットワークトラフィック (`NetworkIn`)、発信ネットワークトラフィック (`NetworkOut`) などがあります。Lightsail で使用可能なすべてのリソースメトリクスの詳細については、「Lightsail でのメトリクス」を参照してください。

メトリクスの保持

期間が 60 秒 (1 分の解像度) のデータポイントは、15 日間使用できます。期間が 300 秒 (5 分の解像度) のデータポイントは、63 日間使用できます。期間が 3600 秒 (1 時間の解像度) のデータポイントは、455 日 (15 か月) 間使用できます。

最初は短い期間で発行されるデータポイントは、長期的なストレージのため一緒に集計されます。たとえば、1 分の精度を持つデータポイントは、1 分の解像度で 15 日間使用できます。15 日を過ぎてもこのデータはまだ利用できますが、集計され、5 分の解像度のみで取得可能になります。63 日を過ぎるとこのデータはさらに集計され、1 時間の解像度のみで利用できます。これらの期間よりも長いメトリクスを使用する必要がある場合は、 Lightsail API、 AWS Command Line Interface (AWS CLI)、および SDKs を使用して、オフラインまたは異なるストレージのデータポイントを取得できます。

詳細については、「Lightsail API リファレンス」の「GetInstanceMetricData」、「GetBucketMetricData」、「GetLoadBalancerMetricData」、「GetDistributionMetricData」、「GetRelationalDatabaseMetricData」を参照してください。

統計

メトリクス統計は、一定期間にわたってデータを集計する手段です。統計情報の例としては、`Average`、`Sum`、`Maximum` などがあります。たとえば、`Average` 統計を使用してインスタン

スの CPU 使用率メトリクスデータを平均化し、Sum 統計を使用してデータベース接続を追加したり、Maximum 統計を使用してロードバランサーの最大応答時間を取得したりできます。

利用可能なメトリクス統計のリストは、「Lightsail API リファレンス」の「GetInstanceMetricData の統計」、「GetBucketMetricData の統計」、「GetLoadBalancerMetricData の統計」、「GetDistributionMetricData の統計」、「GetRelationalDatabaseMetricData の統計」を参照してください。

単位

各統計には、測定単位があります。単位の例は、Bytes、Seconds、Count、Percent などです。ユニットの全一覧は、Lightsail API リファレンスの GetInstanceMetricData のユニット、GetLoadBalancerMetricData のユニット、GetDistributionMetricData のユニットおよび GetRelationalDatabaseMetricData のユニットを参照してください。

期間

期間とは、返されたデータポイントの粒度を示す特定のデータポイントに関連付けられた時間の長さです。各データポイントは、指定された期間に収集されたメトリクスデータの集約を表しています。期間は秒単位で定義され、期間の有効値は 60 秒 (1 分) と 300 秒 (5 分) の倍数です。

Lightsail API を使用してデータポイントを取得する場合、期間、開始時刻、終了時刻を指定できます。これらのパラメータは、データポイントに関連付けられた全体の時間の長さを決定します。Lightsail は、メトリクスデータを 1 分または 5 分単位でレポートします。したがって、期間は 60 秒と 300 秒の倍数で指定する必要があります。開始時刻と終了時刻に指定した値により、Lightsail により返される期間が決まります。10 分区切りで集約された統計を取得する場合は、期間を 600 に指定します。1 時間分の集約された統計の場合は、期間を 3600 などに設定します。

Lightsail アラームには期間も重要です。Lightsail はアラームのデータポイントを 5 分ごとに評価し、アラームの各データポイントは 5 分間の集約データを表します。特定のメトリクスをモニタリングするアラームを作成したら、そのメトリクスと指定したしきい値を比較するよう Lightsail に依頼していることになります。ユーザーは、Lightsail がその比較を行う方法を広範囲に制御できます。比較を行う期間を指定し、結論に達するために使用する評価期間の数を指定することもできます。詳細については、「アラーム」を参照してください。

アラーム

アラームは、指定した期間に 1 つのメトリクスをモニタリングし、メトリクスが指定したしきい値を超えたときに通知します。通知は、Lightsail コンソールに表示されるバナー、指定したメールアド

レスに送信された メール、指定した携帯電話番号に送信された SMS テキストメッセージです。詳細については、「アラーム」を参照してください。

Lightsail で使用可能なメトリクス

インスタンスメトリクス

次のインスタンスメトリクスを使用できます。詳細については、「Amazon Lightsail でインスタンスのメトリクスを表示する」を参照してください。

- CPU 使用率 (**CPUUtilization**) — 割り当てられたコンピューティングユニットのうち、現在インスタンス上で使用されているものの割合。このメトリクスは、インスタンスでアプリケーションを実行するための処理能力を識別します。インスタンスがフルプロセッサコアに割り当てられていない場合に、オペレーティングシステムのツールが Lightsail よりも低い割合を示す場合があります。

  Lightsail コンソールでインスタンスの CPU 使用率メトリクスグラフを表示すると、持続可能なゾーンとバースト可能なゾーンが表示されます。これらのゾーンの意味の詳細については、「CPU 使用率の持続可能なゾーンとバースト可能なゾーン」を参照してください。
- バーストキャパシティ (**BurstCapacityTime**) および割合 (**BurstCapacityPercentage**) — バーストキャパシティ分数は、インスタンスが CPU 使用率 100% でバーストできる時間を表します。バーストキャパシティの割合は、インスタンスで利用できる CPU パフォーマンスの割合です。インスタンスはバースト容量を継続的に消費し、蓄積します。インスタンスが 100% の CPU 使用率で動作しているときにのみ、バーストキャパシティの分数がフルレートで消費されます。インスタンスのバースト容量の詳細については、「Amazon Lightsail でのインスタンスのバースト容量の表示」を参照してください。
- 受信ネットワークトラフィック (**NetworkIn**) — すべてのネットワークインターフェイスでの、このインスタンスによって受信されたバイト数。このメトリクスは、1 つのインスタンスへの着信ネットワークトラフィックの量を表しています。報告された数は期間中に受信されたバイト数です。このメトリクスは 5 分間隔でレポートされるため、レポートされた数を 300 で割ると、バイト/秒を算出できます。
- 送信ネットワークトラフィック (**NetworkOut**) — すべてのネットワークインターフェイスでの、このインスタンスから送信されたバイト数。このメトリクスは、1 つのインスタンスからの発信ネットワークトラフィックの量を表しています。報告された数は期間中に送信されたバイト数です。このメトリクスは 5 分間隔でレポートされるため、レポートされた数を 300 で割ると、バイト/秒を算出できます。

- ステータスチェックの失敗 (**StatusCheckFailed**) — インスタンスが、インスタンスステータスチェックとシステムステータスチェックの両方に合格したか失敗したかをレポートします。このメトリクスは 0 (合格) または 1 (失敗) となります。このメトリクスは、1 分間の頻度で利用できます。
- インスタンスステータスチェックの失敗 (**StatusCheckFailed_Instance**) — インスタンスがインスタンスステータスチェックに合格したか、失敗したかをレポートします。このメトリクスは 0 (合格) または 1 (失敗) となります。このメトリクスは、1 分間の頻度で利用できます。
- ステータスチェックの失敗 (**StatusCheckFailed_System**) — インスタンスが、システムステータスチェックに合格したか失敗したかをレポートします。このメトリクスは 0 (合格) または 1 (失敗) となります。このメトリクスは、1 分間の頻度で利用できます。
- トークンメタデータなしのリクエスト (**MetadataNoToken**) — トークンなしでインスタンスのメタデータサービスに正常にアクセスした回数。このメトリクスにより、トークンを使用しない Instance Metadata Service バージョン 1 を使用してインスタンスメタデータにアクセスするプロセスがあるかどうかがわかります。すべてのリクエストがトークン支援のセッション (Instance Metadata Service バージョン 2 など) を使用している場合、値は 0 になります。詳細については、「Amazon Lightsail のインスタンスメタデータとユーザーデータ」を参照してください。

データベースメトリクス

次のデータベースメトリクスを使用できます。詳細については、「Amazon Lightsail でのデータベースメトリクスの表示」を参照してください。

- CPU 使用率 (**CPUUtilization**) — データベースで現在使用されている CPU 使用率の割合。
- データベース接続 (**DatabaseConnections**) — 使用中のデータベース接続の数。
- ディスクのキューの深度 (**DiskQueueDepth**) — ディスクへのアクセスを待機している未処理の IO (読み取り/書き込みリクエスト) の数。
- 空きストレージ容量 (**FreeStorageSpace**) — 使用可能なストレージの容量。
- ネットワーク受信スループット (**NetworkReceiveThroughput**) — モニタリングとレプリケーションに使用する顧客データベーストラフィックと AWS トラフィックの両方を含む、データベースの受信ネットワークトラフィック。
- ネットワーク送信スループット (**NetworkTransmitThroughput**) — モニタリングとレプリケーションに使用する顧客データベーストラフィックと AWS トラフィックの両方を含む、データベースの送信ネットワークトラフィック。

ディストリビューションメトリクス

以下のディストリビューションメトリクスが利用可能です。詳細については、「Amazon Lightsail のディストリビューションメトリクスの表示」を参照してください。

- リクエスト (**Requests**) — すべての HTTP メソッド、および HTTP と HTTPS 両方のリクエストについて、ディストリビューションが受信したビューワーリクエストの総数。
- アップロードされたバイト数 (**BytesUploaded**) — POST リクエストと PUT リクエストを使用して、ディストリビューションによってオリジンにアップロードされたバイト数。
- ダウンロードされたバイト数 (**BytesDownloaded**) — GET リクエスト、HEAD リクエスト、および OPTIONS リクエストに対してビューワーがダウンロードしたバイト数。
- トータルエラー率 (**TotalErrorRate**) — レスポンスの HTTP ステータスコードが 4xx または 5xx であったすべてのビューワーリクエストの割合 (%)。
- HTTP 4xx トータルエラー率 (**4xxErrorRate**) — レスポンスの HTTP ステータスコードが 4xx であったすべてのビューワーリクエストの割合 (%)。このような場合、クライアントまたはクライアントビューワーでエラーが発生した可能性があります。たとえば、ステータスコード 404 (Not Found) は、クライアントが、検出できないオブジェクトをリクエストしたことを意味します。
- HTTP 5xx トータルエラー率 (**5xxErrorRate**) — レスポンスの HTTP ステータスコードが 5xx であったすべてのビューワーリクエストの割合 (%)。このような場合、オリジンサーバーはリクエストを満たしませんでした。たとえば、ステータスコード 503 (Service Unavailable) は、オリジンサーバーが現在利用できないことを意味します。

ロードバランサーのメトリクス

次のロードバランサーメトリクスを使用できます。詳細については、「Amazon Lightsail でのロードバランサーメトリクスの表示」を参照してください。

- 正常ホスト数 (**HealthyHostCount**) — 正常と見なされるターゲットインスタンスの数。
- 異常ホスト数 (**UnhealthyHostCount**) — 異常と見なされるターゲットインスタンスの数。
- ロードバランサー HTTP 4XX (**HTTPCode_LB_4XX_Count**) — ロードバランサーから発生した HTTP 4XX クライアントエラーコードの数。リクエストの形式が不正な場合、または不完全な場合は、クライアントエラーが生成されます。これらのリクエストは、ターゲットインスタンスによって受信されませんでした。この数には、ターゲットインスタンスによって生成される応答コードは含まれません。

- ロードバランサー HTTP 5XX (**HTTPCode_LB_5XX_Count**) — ロードバランサーから発生した HTTP 5XX サーバーエラーコードの数。これには、ターゲットインスタンスによって生成される応答コードは含まれません。ロードバランサーにアタッチされている正常なインスタンスがない場合、またはリクエストレートがインスタンスやロードバランサーの容量を超える場合 (スピルオーバー)、このメトリクスが報告されます。
- インスタンス HTTP 2XX (**HTTPCode_Instance_2XX_Count**) — ターゲットインスタンスによって生成された HTTP 2XX 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。
- インスタンス HTTP 3XX (**HTTPCode_Instance_3XX_Count**) — ターゲットインスタンスによって生成された HTTP 3XX 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。
- インスタンス HTTP 4XX (**HTTPCode_Instance_4XX_Count**) — ターゲットインスタンスによって生成された HTTP 4XX 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。
- インスタンス HTTP 5XX (**HTTPCode_Instance_5XX_Count**) — ターゲットインスタンスによって生成された HTTP 5XX 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。
- インスタンスからの応答時間 (**InstanceResponseTime**) — ロードバランサーがリクエストを送信してから、ターゲットインスタンスからの応答を受信するまでの経過時間 (秒)。
- クライアント TLS ネゴシエーションエラー数 (**ClientTLSNegotiationErrorCount**) — クライアントにより開始され、ロードバランサーによって生成された TLS エラーのためにロードバランサーとのセッションを確立しなかった、TLS 接続の数。暗号化またはプロトコルの不一致が原因である場合があります。
- リクエストの数 (**RequestCount**) — IPv4 経由で処理されたリクエストの数。この数には、ロードバランサーのターゲットインスタンスによって生成されたレスポンスを含むリクエストのみが含まれます。
- 拒否された接続数 (**RejectedConnectionCount**) — ロードバランサーが接続の最大数に達したため、拒否された接続の数。

コンテナサービスのメトリクス

以下のコンテナサービスメトリクスが利用可能です。詳細については、「コンテナサービスメトリクスを表示する」を参照してください。

- CPU 使用率 (**CPUUtilization**) — コンテナサービスの全ノードで現在使用されているコンピューティングユニットの平均比率。このメトリクスは、コンテナサービス上のコンテナを実行するのに必要な処理能力を特定します。
- メモリ使用率 (**MemoryUtilization**) — コンテナサービスの全ノードで現在使用されているメモリの平均比率。このメトリクスは、コンテナサービス上のコンテナを実行するのに必要なメモリを特定します。

バケットメトリクス

次のバケットメトリクスが利用可能です。詳細については、「Amazon Lightsail のバケットメトリクスを表示」を参照してください。

- [バケットサイズ (**BucketSizeBytes**)] — バケットに保存されたデータの量。この値を計算するには、バケット内のすべてのオブジェクト (最新のオブジェクトと最新でないオブジェクトの両方) のサイズを合計します。これには、バケットに対するすべての不完全なマルチパートアップロードのすべてのパートのサイズも含めます。
- [オブジェクトの数 (**NumberOfObjects**)] — バケットに保存されたオブジェクトの総数。この値を計算するには、バケット内のすべてのオブジェクト (最新のオブジェクトと最新でないオブジェクトの両方) と、バケットに対するすべての不完全なマルチパートアップロードの合計パート数をカウントします。

> **Note**
>
> バケットが空の場合、バケットメトリックスデータはレポートされません。

ヘルスメトリクスを使用して Lightsail リソースをモニタリングする

さまざまな期間で以下の Amazon Lightsail のリソースメトリクスを表示できます。Lightsail のリソースメトリクスの詳細については、「リソースメトリクス」を参照してください。

インスタンスメトリクス

次のインスタンスメトリクスを使用できます。詳細については、「Amazon Lightsail でインスタンスのメトリクスを表示する」を参照してください。

- CPU 使用率 (**CPUUtilization**) — 割り当てられたコンピューティングユニットのうち、現在インスタンス上で使用されているものの割合。このメトリクスは、インスタンスでアプリケーションを実行するための処理能力を識別します。インスタンスがフルプロセッサコアに割り当てられていない場合に、オペレーティングシステムのツールが Lightsail よりも低い割合を示す場合があります。

  Lightsail コンソールでインスタンスの CPU 使用率メトリクスグラフを表示すると、持続可能なゾーンとバースト可能なゾーンが表示されます。これらのゾーンの意味の詳細については、「CPU 使用率の持続可能なゾーンとバースト可能なゾーン」を参照してください。
- バーストキャパシティ (**BurstCapacityTime**) および割合 (**BurstCapacityPercentage**) — バーストキャパシティ分数は、インスタンスが CPU 使用率 100% でバーストできる時間を表します。バーストキャパシティの割合は、インスタンスで利用できる CPU パフォーマンスの割合です。インスタンスはバースト容量を継続的に消費し、蓄積します。インスタンスが 100% の CPU 使用率で動作しているときにのみ、バーストキャパシティの分数がフルレートで消費されます。インスタンスのバーストキャパシティの詳細については、「インスタンスのバーストキャパシティの表示」を参照してください。
- 受信ネットワークトラフィック (**NetworkIn**) — すべてのネットワークインターフェイスでの、このインスタンスによって受信されたバイト数。このメトリクスは、1 つのインスタンスへの着信ネットワークトラフィックの量を表しています。報告された数は期間中に受信されたバイト数です。このメトリクスは 5 分間隔でレポートされるため、レポートされた数を 300 で割ると、バイト/秒を算出できます。
- 送信ネットワークトラフィック (**NetworkOut**) — すべてのネットワークインターフェイスでの、このインスタンスから送信されたバイト数。このメトリクスは、1 つのインスタンスからの発信ネットワークトラフィックの量を表しています。報告された数は期間中に送信されたバイト数です。このメトリクスは 5 分間隔でレポートされるため、レポートされた数を 300 で割ると、バイト/秒を算出できます。
- ステータスチェックの失敗 (**StatusCheckFailed**) — インスタンスが、インスタンスステータスチェックとシステムステータスチェックの両方に合格したか失敗したかをレポートします。このメトリクスは 0 (合格) または 1 (失敗) となります。このメトリクスは、1 分間の頻度で利用できます。
- インスタンスステータスチェックの失敗 (**StatusCheckFailed_Instance**) — インスタンスがインスタンスステータスチェックに合格したか、失敗したかをレポートします。このメトリクスは 0 (合格) または 1 (失敗) となります。このメトリクスは、1 分間の頻度で利用できます。
- ステータスチェックの失敗 (**StatusCheckFailed_System**) — インスタンスが、システムステータスチェックに合格したか失敗したかをレポートします。このメトリクスは 0 (合格) または 1 (失敗) となります。このメトリクスは、1 分間の頻度で利用できます。

- ステータスチェックの失敗 (**StatusCheckFailed_System**) — インスタンスが、システムステータスチェックに合格したか失敗したかをレポートします。このメトリクスは 0 (合格) または 1 (失敗) となります。このメトリクスは、1 分間の頻度で利用できます。
- トークンメタデータなしのリクエスト (**MetadataNoToken**) — トークンなしでインスタンスのメタデータサービスに正常にアクセスした回数。このメトリクスにより、トークンを使用しない Instance Metadata Service バージョン 1 を使用してインスタンスメタデータにアクセスするプロセスがあるかどうかがわかります。すべてのリクエストが、Instance Metadata Service バージョン 2 あどのトークン支援のセッションを使用している場合、値は 0 になります。詳細については、「インスタンスメタデータとユーザーデータ」を参照してください。

データベースメトリクス

次のデータベースメトリクスを使用できます。詳細については、「データベースメトリクスの表示」を参照してください。

- CPU 使用率 (**CPUUtilization**) — データベースで現在使用されている CPU 使用率の割合。
- データベース接続 (**DatabaseConnections**) — 使用中のデータベース接続の数。
- ディスクのキューの深度 (**DiskQueueDepth**) — ディスクへのアクセスを待機している未処理の IO (読み取り/書き込みリクエスト) の数。
- 空きストレージ容量 (**FreeStorageSpace**) — 使用可能なストレージの容量。
- ネットワーク受信スループット (**NetworkReceiveThroughput**) — モニタリングとレプリケーションに使用する顧客データベーストラフィックと AWS トラフィックの両方を含む、データベースの受信ネットワークトラフィック。
- ネットワーク送信スループット (**NetworkTransmitThroughput**) — モニタリングとレプリケーションに使用する顧客データベーストラフィックと AWS トラフィックの両方を含む、データベースの送信ネットワークトラフィック。

ディストリビューションメトリクス

以下のディストリビューションメトリクスが利用可能です。詳細については、「Amazon Lightsail のディストリビューションメトリクスの表示」を参照してください。

- リクエスト — すべての HTTP メソッド、および HTTP と HTTPS 両方のリクエストについて、ディストリビューションが受信したビューワーリクエストの総数。

- アップロードされたバイト数 — POST リクエストと PUT リクエストを使用して、ディストリビューションによってオリジンにアップロードされたバイト数。
- ダウンロードされたバイト数 — GET リクエスト、HEAD リクエスト、および OPTIONS リクエストに対してビューワーがダウンロードしたバイト数。
- トータルエラー率 — レスポンスの HTTP ステータスコードが 4xx または 5xx であったすべてのビューワーリクエストの割合 (%)。
- HTTP 4xx トータルエラー率 — レスポンスの HTTP ステータスコードが 4xx であったすべてのビューワーリクエストの割合 (%)。このような場合、クライアントまたはクライアントビューワーでエラーが発生した可能性があります。たとえば、ステータスコード 404 (Not Found) は、クライアントが、検出できないオブジェクトをリクエストしたことを意味します。
- HTTP 5xx トータルエラー率 — レスポンスの HTTP ステータスコードが 5xx であったすべてのビューワー リクエストの割合 (%)。このような場合、オリジンサーバーはリクエストを満たしませんでした。たとえば、ステータスコード 503 (Service Unavailable) は、オリジンサーバーが現在利用できないことを意味します。

ロードバランサーのメトリクス

次のロードバランサーメトリクスを使用できます。詳細については、「ロードバランサーメトリクスの表示」を参照してください。

- 正常ホスト数 (**HealthyHostCount**) — 正常と見なされるターゲットインスタンスの数。
- 異常ホスト数 (**UnhealthyHostCount**) — 異常と見なされるターゲットインスタンスの数。
- ロードバランサー HTTP 4XX (**HTTPCode_LB_4XX_Count**) — ロードバランサーから発生した HTTP 4XX クライアントエラーコードの数。リクエストの形式が不正な場合、または不完全な場合は、クライアントエラーが生成されます。これらのリクエストは、ターゲットインスタンスによって受信されませんでした。この数には、ターゲットインスタンスによって生成される応答コードは含まれません。
- ロードバランサー HTTP 5XX (**HTTPCode_LB_5XX_Count**) — ロードバランサーから発生した HTTP 5XX サーバーエラーコードの数。これには、ターゲットインスタンスによって生成される応答コードは含まれません。ロードバランサーにアタッチされている正常なインスタンスがない場合、またはリクエストレートがインスタンスやロードバランサーの容量を超える場合 (スピルオーバー)、このメトリクスが報告されます。
- インスタンス HTTP 2XX (**HTTPCode_Instance_2XX_Count**) — ターゲットインスタンスによって生成された HTTP 2XX 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。

- インスタンス HTTP 3XX (**HTTPCode_Instance_3XX_Count**) — ターゲットインスタンスによって生成された HTTP 3XX 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。
- インスタンス HTTP 4XX (**HTTPCode_Instance_4XX_Count**) — ターゲットインスタンスによって生成された HTTP 4XX 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。
- インスタンス HTTP 5XX (**HTTPCode_Instance_5XX_Count**) — ターゲットインスタンスによって生成された HTTP 5XX 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。
- インスタンスからの応答時間 (**InstanceResponseTime**) — ロードバランサーがリクエストを送信してから、ターゲットインスタンスからの応答を受信するまでの経過時間 (秒)。
- リクエストの数 (**RequestCount**) — IPv4 経由で処理されたリクエストの数。この数には、ロードバランサーのターゲットインスタンスによって生成されたレスポンスを含むリクエストのみが含まれます。
- クライアント TLS ネゴシエーションエラー数 (**ClientTLSNegotiationErrorCount**) — クライアントにより開始され、ロードバランサーによって生成された TLS エラーのためにロードバランサーとのセッションを確立しなかった、TLS 接続の数。暗号化またはプロトコルの不一致が原因である場合があります。
- 拒否された接続数 (**RejectedConnectionCount**) — ロードバランサーが接続の最大数に達したため、拒否された接続の数。

コンテナサービスのメトリクス

以下のコンテナサービスメトリクスが利用可能です。詳細については、「コンテナサービスメトリクスを表示する」を参照してください。

- CPU 使用率 — コンテナサービスの全ノードで現在使用されているコンピューティングユニットの平均比率。このメトリクスは、コンテナサービス上のコンテナを実行するのに必要な処理能力を特定します。
- メモリ使用率 — コンテナサービスの全ノードで現在使用されているメモリの平均比率。このメトリクスは、コンテナサービス上のコンテナを実行するのに必要なメモリを特定します。

バケットメトリクス

次のバケットメトリクスが利用可能です。詳細については、「バケットメトリクスを表示」を参照してください。

- [バケットサイズ] — バケットに保存されたデータの量。この値を計算するには、バケット内のすべてのオブジェクト (最新のオブジェクトと最新でないオブジェクトの両方) のサイズを合計します。これには、バケットに対するすべての不完全なマルチパートアップロードのすべてのパートのサイズも含めます。
- オブジェクトの数 — バケットに保存されたオブジェクトの総数。この値を計算するには、バケット内のすべてのオブジェクト (最新のオブジェクトと最新でないオブジェクトの両方) と、バケットに対するすべての不完全なマルチパートアップロードの合計パート数をカウントします。

> **Note**
>
> バケットが空の場合、バケットメトリックスデータはレポートされません。

トピック

- Lightsail リソースのメトリクス通知を設定する
- メトリクスを使用して Lightsail インスタンスのパフォーマンスをモニタリングする
- Lightsail のメトリクスアラーム
- Lightsail インスタンスのメトリクスアラームを作成する
- Lightsail メトリクスアラームの削除または無効化

Lightsail リソースのメトリクス通知を設定する

インスタンス、データベース、ロードバランサーないしコンテンツ配信ネットワーク (CDN) ディストリビューションのメトリクスが指定した値を超えた場合に通知を受けられるよう、Lightsail を設定することが可能す。通知は、Lightsail コンソールに表示されるバナー、指定したアドレスに送信される メール、指定した携帯電話番号に送信される SMS テキストメッセージの形式にすることができます。通知の検証保留中の連絡先を確認する方法の詳細については、「検証保留中の E メール連絡先を確認する」を参照してください。

通知を取得するには、リソースの 1 つのメトリクスを監視するアラームを設定する必要があります。たとえば、指定した時間内にインスタンスの発信ネットワークトラフィックが 500 KB を超えた

場合に通知するアラームを設定できます。詳細については、「メトリクスのアラーム」を参照してください。

アラームが作動すると、通知バナーが Lightsail コンソールに表示されます。E メールと SMS テキストメッセージで通知を受け取るには、リソースをモニタリングする各 AWS リージョン に通知連絡先として E メールアドレスと携帯電話番号を追加する必要があります。詳細については、「通知連絡先を追加する」を参照してください。

> **Note**
>
> SMS テキストメッセージは AWS リージョン、Lightsailリソースを作成できるすべての でサポートされているわけではなく、テキストメッセージを世界の一部の国や地域に送信することはできません。詳細については、「通知連絡先を追加する」を参照してください。

通知が予定されているときに通知を受け取らない場合は、通知の連絡先が正しく設定されていることを確認するためにいくつかの点を確認してください。詳細については、「通知のトラブルシューティング」を参照してください。

通知の受信を停止するには、Lightsail から メールと携帯電話を削除します。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。また、アラームを無効化または削除して、特定のアラームの通知の受信を停止することもできます。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。

メトリクスを使用して Lightsail インスタンスのパフォーマンスをモニタリングする

Amazon Lightsail でインスタンスを起動すると、インスタンスの管理ページの [Metrics (メトリクス)] タブでメトリクスグラフを表示できるようになります。メトリクスのモニタリングは、リソースの信頼性、可用性、パフォーマンスを維持する上で重要な要素です。リソースから定期的にメトリクスデータをモニタリングして収集し、マルチポイント障害が発生した場合に、より簡単にデバッグできるようにします。メトリクスの詳細については、「Amazon Lightsail のメトリクス」を参照してください。

リソースを監視するときは、環境内の通常のリソースパフォーマンスのベースラインを確立する必要があります。その後、リソースのパフォーマンスが指定のしきい値を超えたときに通知するように、Lightsail コンソールでアラームを設定できます。詳細については、「通知」および「アラーム」を参照してください。

目次

- Lightsail で利用可能なインスタンスメトリクス
- CPU 使用率の持続可能なゾーンとバースト可能なゾーン
- Lightsail コンソールでインスタンスメトリクスを表示する
- インスタンスメトリクスの表示後の次のステップ

利用可能なインスタンスメトリクス

次のインスタンスメトリクスを使用できます。

- CPU 使用率 (**CPUUtilization**) — 割り当てられたコンピューティングユニットのうち、現在インスタンス上で使用されているものの割合。このメトリクスは、インスタンスでアプリケーションを実行するための処理能力を識別します。インスタンスがフルプロセッサコアに割り当てられていない場合に、オペレーティングシステムのツールが Lightsail よりも低い割合を示す場合があります。

  Lightsail コンソールでインスタンスの CPU 使用率メトリクスグラフを表示すると、持続可能なゾーンとバースト可能なゾーンが表示されます。これらのゾーンの意味の詳細については、「CPU 使用率の持続可能なゾーンとバースト可能なゾーン」を参照してください。
- バーストキャパシティ (**BurstCapacityTime**) および割合 (**BurstCapacityPercentage**) — バーストキャパシティ分数は、インスタンスが CPU 使用率 100% でバーストできる時間を表します。バーストキャパシティの割合は、インスタンスで利用できる CPU パフォーマンスの割合です。インスタンスはバースト容量を継続的に消費し、蓄積します。インスタンスが 100% の CPU 使用率で動作しているときにのみ、バーストキャパシティの分数がフルレートで消費されます。インスタンスのバーストキャパシティの詳細については、「インスタンスのバーストキャパシティの表示」を参照してください。
- 受信ネットワークトラフィック (**NetworkIn**) — すべてのネットワークインターフェイスでの、このインスタンスによって受信されたバイト数。このメトリクスは、1 つのインスタンスへの着信ネットワークトラフィックの量を表しています。報告された数は期間中に受信されたバイト数です。このメトリクスは 5 分間隔でレポートされるため、レポートされた数を 300 で割ると、バイト/秒を算出できます。
- 送信ネットワークトラフィック (**NetworkOut**) — すべてのネットワークインターフェイスでの、このインスタンスから送信されたバイト数。このメトリクスは、1 つのインスタンスからの発信ネットワークトラフィックの量を表しています。報告された数は期間中に送信されたバイト数で

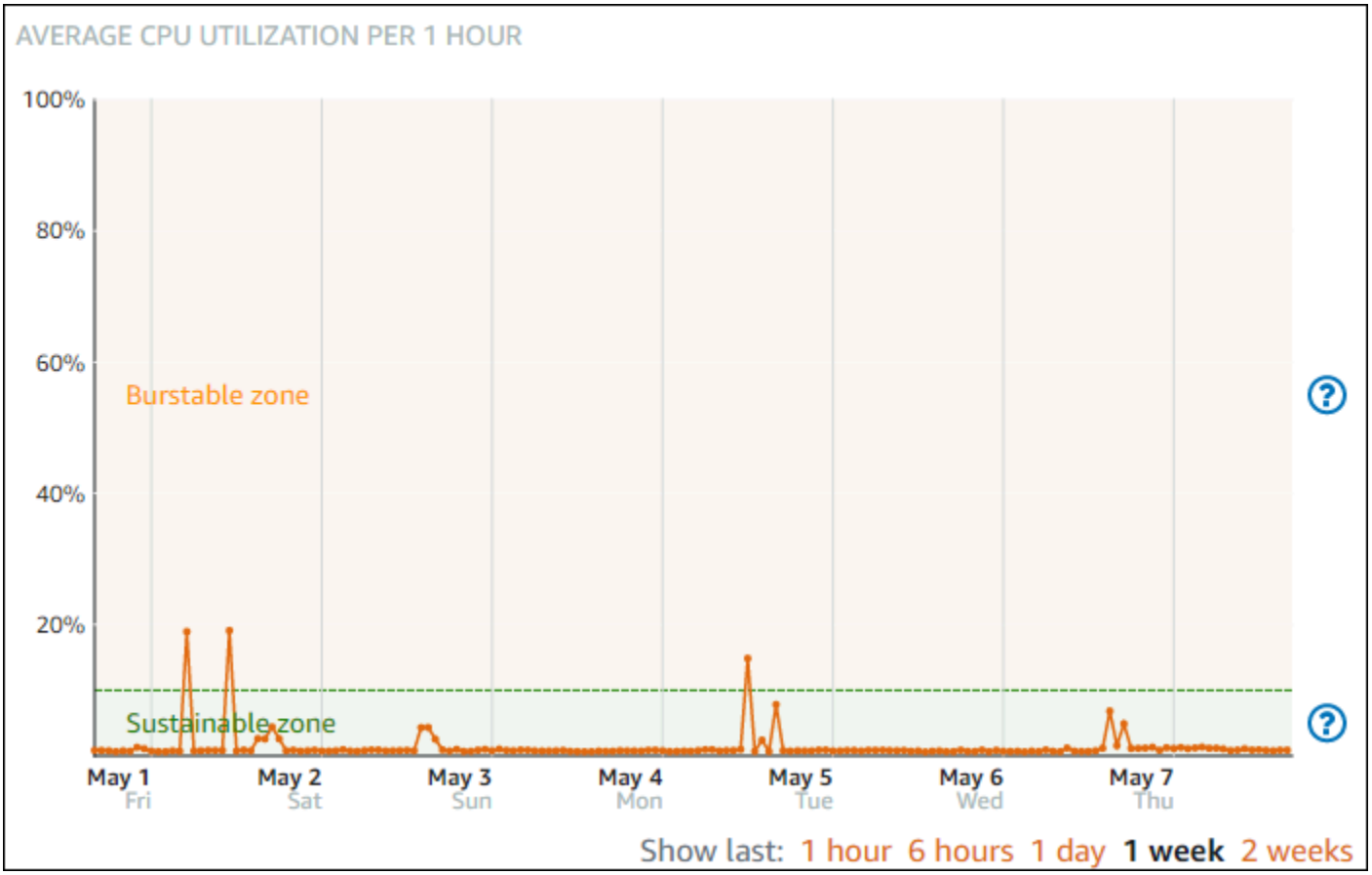
す。このメトリクスは 5 分間隔でレポートされるため、レポートされた数を 300 で割ると、バイト/秒を算出できます。

- ステータスチェックの失敗 (**StatusCheckFailed**) — インスタンスが、インスタンスステータスチェックとシステムステータスチェックの両方に合格したか失敗したかをレポートします。このメトリクスは 0 (合格) または 1 (失敗) となります。このメトリクスは、1 分間の頻度で利用できます。
- インスタンスステータスチェックの失敗 (**StatusCheckFailed_Instance**) — インスタンスがインスタンスステータスチェックに合格したか、失敗したかをレポートします。このメトリクスは 0 (合格) または 1 (失敗) となります。このメトリクスは、1 分間の頻度で利用できます。
- ステータスチェックの失敗 (**StatusCheckFailed_System**) — インスタンスが、システムステータスチェックに合格したか失敗したかをレポートします。このメトリクスは 0 (合格) または 1 (失敗) となります。このメトリクスは、1 分間の頻度で利用できます。
- トークンメタデータなしのリクエスト (**MetadataNoToken**) — トークンなしでインスタンスのメタデータサービスに正常にアクセスした回数。このメトリクスにより、トークンを使用しない Instance Metadata Service バージョン 1 を使用してインスタンスメタデータにアクセスするプロセスがあるかどうかがわかります。すべてのリクエストがトークン支援のセッション (Instance Metadata Service バージョン 2 など) を使用している場合、値は 0 になります。詳細については、「インスタンスメタデータとユーザーデータ」を参照してください。

CPU 使用率の持続可能なゾーンとバースト可能なゾーン

Lightsail は、バースト可能なインスタンスを使用します。これは CPU パフォーマンスのベースライン量を提供しますが、必要に応じてベースラインを上回る CPU パフォーマンスを一時的に提供することもできます。これをバーストといいます。バースト可能なインスタンスを使用すると、時々発生するパフォーマンスの急上昇 (スパイク) に対応するためにインスタンスを過剰にプロビジョンする必要がありません。つまり、使用しない容量に対して料金を支払う必要がありません。

インスタンスの CPU 使用率メトリクスグラフに、持続可能なゾーンとバースト可能なゾーンが表示されます。Lightsail 持続可能領域内だと、インスタンスはシステムのオペレーションに影響を与えることなく無制限で作動することができます。



コードのコンパイル、新しいソフトウェアのインストール、バッチジョブの実行、ピークの負荷リクエストの処理など、負荷が高い場合、インスタンスがバースト可能なゾーンで動作し始めることがあります。バーストな可能ゾーンで動作している間、インスタンスは大量の CPU サイクルを消費します。したがって、この領域では限られた期間しか作動できません。

インスタンスがバースト可能なゾーンで動作できる期間は、バースト可能なゾーンにどの程度入っているかによって異なります。バースト可能なゾーンの下限近くで動作しているインスタンスは、バースト可能なゾーンの上限近くで動作しているインスタンスよりも長い時間バーストできます。ただし、一定期間バースト可能なゾーンにあるインスタンスは、持続可能なゾーンで再び動作するまで、最終的にすべての CPU 容量を使い果たすことになります。

インスタンスの CPU 使用率メトリクスを監視して、持続可能なゾーンとバーストな可能ゾーン間でパフォーマンスがどのように分散されているかを確認してください。システムが時折バースト可能なゾーンに移動するだけの場合は、実行中のインスタンスを引き続き使用しても問題ありません。ただし、インスタンスがバーストな可能ゾーンでかなりの時間を費やしている場合は、インスタンスのより大きなプランに切り替えることを検討してください (たとえば、12 米ドル/月プランではなく 5 米ドル/月プランを使用します)。インスタンスの新しいスナップショットを作成し、スナップショットから新しいインスタンスを作成することで、より大きなプランに切り替えることができます。

Lightsail コンソールでインスタンスメトリクスを表示する

Lightsail コンソールでインスタンスメトリクスを表示するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [インスタンス] を選択してください。
3. メトリクスを表示するインスタンスの名前を選択します。
4. インスタンス管理ページで [Metrics (メトリクス)] タブを選択します。
5. [Metrics graph (メトリクスグラフ)] 見出しの下のドロップダウンメニューで、表示するメトリクスを選択します。

   グラフには、選択したメトリクスのデータポイントが視覚的に表示されます。

   > **Note**
   >
   > Lightsail コンソールでインスタンスの CPU 使用率メトリクスグラフを表示すると、持続可能なゾーンとバースト可能なゾーンが表示されます。これらのゾーンの詳細については、「CPU 使用率の持続可能なゾーンとバースト可能なゾーン」を参照してください。

6. メトリクスグラフでは、次のアクションを実行できます。

   - グラフの表示を変更して、1 時間、6 時間、1 日、1 週間、2 週間のデータを表示します。
   - データポイント上にカーソルを置くと、そのデータポイントに関する詳細情報が表示されます。
   - 指定したしきい値をメトリクスが超えたときに通知される、選択したメトリクスのアラームを追加します。詳細については、「アラーム」および「インスタンスのメトリクスアラームを作成する」を参照してください。

次の手順

インスタンスメトリクスに対して実行できる追加のタスクがいくつかあります。

- 指定したしきい値をメトリクスが超えたときに通知される、選択したメトリクスのアラームを追加します。詳細については、「メトリクスのアラーム」および「インスタンスのメトリクスアラームを作成する」を参照してください。

- アラームが作動すると、通知バナーが Lightsail コンソールに表示されます。E メールと SMS テキストメッセージで通知を受け取るには、リソースをモニタリングする各 AWS リージョン に通知連絡先として E メールアドレスと携帯電話番号を追加する必要があります。詳細については、「通知連絡先を追加する」を参照してください。
- 通知の受信を停止するには、Lightsail から メールと携帯電話を削除します。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。また、アラームを無効化または削除して、特定のアラームの通知の受信を停止することもできます。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。

Lightsail のメトリクスアラーム

Amazon Lightsail では、インスタンス、データベース、ロードバランサー、コンテンツ配信ネットワーク (CDN) ディストリビューションでの 1 つのメトリクスを監視するアラームを作成できます。アラームは、指定したしきい値を基準にしたメトリクスの値に基づいて通知するように設定できます。通知は、Lightsail コンソールに表示されるバナー、メールアドレスに送信される メール、携帯電話番号に送信される SMS テキストメッセージです。このガイドでは、アラームの条件と設定について説明します。すべての Lightsail リソースでアクティブなアラームを確認する方法の詳細については、「アクティブなアラームのアラーム通知を確認する」を参照してください。

内容

- アラームを設定する
- アラームの状態
- アラームの例
- アラームによる欠落データの処理方法の設定
- データが欠落した場合のアラーム状態の評価方法
- グラフ化された例の欠落データ
- アラームの詳細

アラームの設定

Lightsail コンソールでアラームを追加するには、インスタンス、データベース、ロードバランサーあるいは CDN ディストリビューションの [メトリクス] タブを参照します。次に、モニタリングするメトリクスを選択し、[Add alarm (アラームの追加)] を選択します。メトリクスごとに 2 つのアラームを追加できます。メトリクスの詳細については、「リソースのメトリクス」を参照してください。

アラームを設定するには、まずしきい値を特定します。しきい値は、アラームが状態を変更する時点のメトリクス値です (OK 状態から ALARM 状態への変更、またはその逆の変更など)。詳細については、「アラームの状態」を参照してください。次に、メトリクスとしきい値の比較に使用する比較演算子を選択します。使用できる演算子は、greater than or equal to、greater than、less than、less than or equal to です。

次に、アラームの状態を変更するために、しきい値を超える必要のある回数と、メトリクスの評価期間を指定します。Lightsail はアラームのデータポイントを 5 分ごとに評価し、各データポイントは集約データの 5 分間の期間を表します。たとえば、しきい値が 2 回を超えたときにトリガーするアラームを指定した場合、評価期間は過去 10 分以上 (最大 24 時間) である必要があります。しきい値を 10 回超えたときにトリガーするアラームを指定した場合、評価期間は過去 50 分以上 (最大 24 時間) である必要があります。

アラームの条件を設定したら、通知方法を設定できます。アラームが Lightsail 状態から OK 状態に変化すると、通知バナーは常に ALARM コンソールに表示されます。メールおよび SMS テキストメッセージによる通知を選択することもできますが、それらの通知連絡先を設定する必要があります。詳細については、「メトリクスの通知」を参照してください。メール、または SMS テキストメッセージによる通知を選択する場合、アラームの状態が ALARM 状態から OK 状態に変化したときに通知を受けるように選択することもできます。これは、すべてクリアな通知と見なされます。

アラームの [Advanced settings (詳細設定)] では、Lightsail が欠落しているメトリクスデータを処理する方法を選択できます。詳細については、「アラームが欠落データを処理する方法の設定」を参照してください。

アラームの状態

アラームは、常に次の状態のいずれかになります。

- ALARM — メトリクスの値が、定義したしきい値の範囲外にあります。

  たとえば、greater than 比較演算子を選択した場合、メトリクスが指定したしきい値を超えると、アラームは ALARM 状態になります。less than 比較演算子を選択した場合、メトリクスが指定したしきい値を下回ると、アラームは ALARM 状態になります。

- OK: — メトリクスの値が、定義されたしきい値の範囲内にあります。

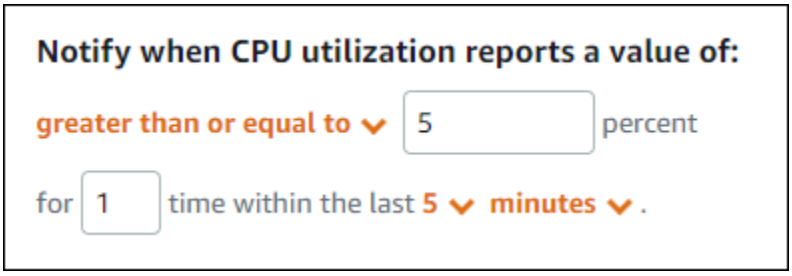
  たとえば、greater than 比較演算子を選択した場合、メトリクスが指定したしきい値を下回ると、アラームは OK 状態になります。less than 比較演算子を選択した場合、メトリクスが指定したしきい値を超えると、アラームは OK 状態になります。

- INSUFFICIENT_DATA — アラームが開始されたか、メトリクスが利用可能でないか、またはメトリクスがアラームの状態を決定するためのデータが不足しています。

アラームは、状態変更に対してのみトリガーされます。アラームは単に、特定の状態にあるだけでは作動しません — 状態が変更されていることが条件です。アラームがトリガーされると、Lightsail コンソールにバナーが表示されます。E メールや SMS テキストメッセージで通知するようにアラームを設定することもできます。

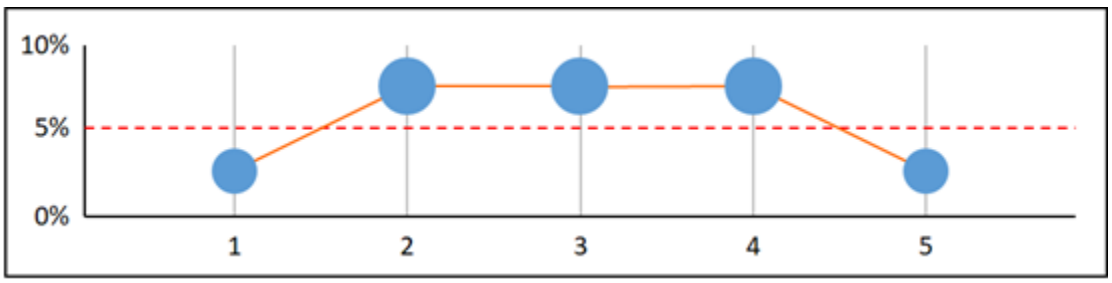
アラームの例

前述のアラーム条件を考慮して、インスタンスの CPU 使用率が 5 分間隔の 1 回で 5% 以上になったときに ALARM 状態になるアラームを設定できます。次の例は、Lightsail コンソールでのこのアラームの設定を示しています。



この例では、インスタンスの CPU 使用率メトリクスが 1 つのデータポイントで 5% 以上の使用率を報告した場合、アラームは OK 状態から ALARM 状態に変化します。使用率が 5% 以上と報告された後続の各データポイントは、アラームをある ALARM 状態で維持します。インスタンスの CPU 使用率メトリクスが、1 つのデータポイントでの使用率を 4.9% 以下と報告すると、アラームは ALARM 状態から OK 状態に変わります。

次のグラフは、このアラームをさらに示しています。赤い点線は 5% の CPU 使用率のしきい値を表し、青い点はメトリクスデータポイントを表します。アラームは、最初のデータポイントの OK 状態です。2 番目のデータポイントは、データポイントがしきい値を超えているため、アラームを ALARM 状態に変更します。データポイントはしきい値よりも大きくなるため、3 番目と 4 番目のデータポイントは ALARM 状態を維持します。5 番目のデータポイントは、データポイントがしきい値を下回っているため、アラームを OK 状態に変更します。



アラームによる欠落データの処理方法の設定

場合によっては、アラームを設定したメトリクスのデータポイントの一部が報告されません。例えば、接続が失われたり、サーバーがダウンしたりした場合に発生します。

Lightsail では、アラームの設定時に欠落したデータポイントを処理する方法を指定できます。監視するデータの種類に応じて適切な場合、ALARM 状態に遷移するようにアラームを設定するときに便利です。欠落データが問題を示すものではない場合の誤検出を避けることができます。

各アラームが常に 3 つの状態のいずれかであるように、データポイントはそれぞれ、次の 3 つのカテゴリのいずれかの状態に該当します。

- Not breaching — データポイントがしきい値の範囲内です。

  たとえば、greater than 比較演算子を選択した場合、指定したしきい値を下回ったときにデータポイントが `Not breaching` になります。less than 比較演算子を選択した場合、指定したしきい値を超えたときにデータポイントは `Not breaching` になります。
- Breaching — データポイントがしきい値の範囲外です。

  たとえば、greater than 比較演算子を選択した場合、指定したしきい値を超えたときにデータポイントが `Breaching` になります。less than 比較演算子を選択すると、指定したしきい値を下回ったときにデータポイントは `Breaching` になります。
- Missing — 欠落しているデータポイントに対する動作は、`treat missing data` パラメータによって指定されます。

アラームごとに、Lightsail が欠落データポイントを次のいずれかとして処理するように指定できます。

- Not breaching — 欠落データポイントは「正常」とされ、しきい値内として扱われます。
- Breaching — 欠落データポイントは「不良」とされ、しきい値超過として扱われます。
- Ignore — 現在のアラーム状態が維持されます。
- Missing — 状態を変更するかどうかを評価する際に、アラームは欠落データポイントを考慮に入れません。これは、アラームのデフォルトの動作です。

最適な選択は、メトリクスの種類によって異なります。インスタンスの CPU 使用率などのメトリクスでは、欠落しているデータポイントをしきい値を超過として扱うことができます。これは、欠落しているデータポイントが、何かが間違っていることを示している可能性があるためです。ただし、

ロードバランサーの HTTP 500 サーバーエラー数など、エラーが発生したときにのみデータポイントを生成するメトリクスでは、欠落したデータをしきい値内として扱うことができます。

アラームに最適なオプションを選択すると、不必要で誤解を招くアラームの状態の変更を防ぐことができます。また、システムの正常性をより正確に示します。

データが欠落した場合のアラーム状態の評価方法

欠落データの処理方法に設定した値にかかわらず、アラームが状態を変更するかどうかを評価する際に、Lightsail は評価期間の指定よりも多くのデータポイントを取得しようとします。取得しようとするデータポイントの正確な数は、アラーム期間の長さによって異なります。取得を試みるデータポイントのタイムフレームは評価範囲です。

Lightsail がこれらのデータポイントを取得すると、次の処理が実行されます。

- 評価範囲内のデータポイントが欠落していない場合、Lightsail は収集された最新のデータポイントに基づいてアラームを評価します。
- 評価範囲のデータポイントの一部が欠落しているが、取得された既存のデータポイントの数がアラームの評価期間以上である場合、正常に取得された最新の既存のデータポイントに基づいてアラームの状態が Lightsail で評価されます。この場合、欠落データを処理する方法に設定した値は不要であり、無視されます。
- 評価範囲のデータポイントの一部が欠落しており、取得された既存のデータポイントの数がアラームの評価期間の数を下回る場合、Lightsail によって、欠落データ部分に欠落データの処理方法に指定された結果が入力され、アラームが評価されます。ただし、評価範囲内の実際のデータポイントは、報告されたタイミングにかかわらず、評価に含まれます。Lightsail は、欠落データポイントの使用を最小限に抑えます。

これらのすべての状況で、評価されるデータポイントの数は、評価期間の値と同じです。超過している数がアラームを発生させるデータポイント数の値よりも少ない場合、アラームの状態は OK に設定されます。それ以外の場合、状態は ALARM に設定されます。

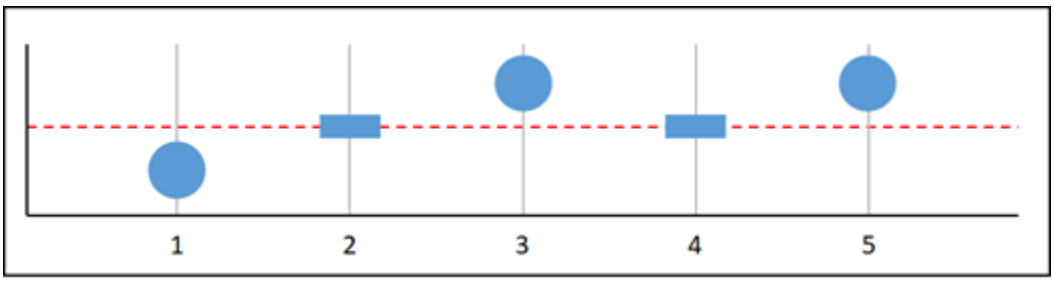
> **Note**
>
> この動作の特殊なケースは、メトリクスのフローが停止した後の一定期間、Lightsail アラームが最後のデータポイントのセットを繰り返し再評価する可能性があることです。この再評価により、メトリクスのストリームが停止する直前にアラームの状態が変更されていた場合に、アクションが再実行される可能性があります。この動作を軽減するには、より短い期間を使用します。

グラフ化された例の欠落データ

このセクションの次のグラフは、アラーム評価動作の例を示しています。グラフ A、B、C、D、E では、確実にアラームに違反しているデータポイントの数と評価期間は両方とも 3 になります。赤い点線はしきい値、青い点は有効なデータポイントを表し、ダッシュは欠落データを表します。しきい値ラインより上のデータポイントはしきい値を超過しており、しきい値を下回るデータポイントはしきい値内です。最新の 3 つのデータポイントの一部が欠落している場合、Lightsail は追加の有効なデータポイントを取得しようとします。

> **Note**
>
> アラームの作成直後にデータポイントが欠落し、アラームの作成前にメトリクスが Lightsail に報告されている場合、Lightsail はアラームを評価する際にアラーム作成前の直近のデータポイントを取得します。

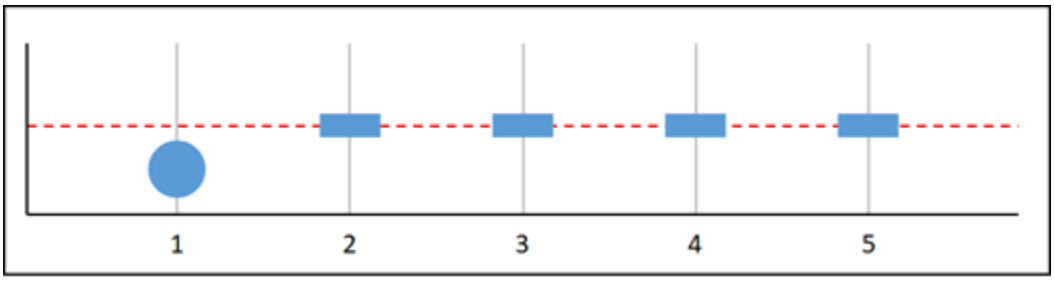
グラフ A



前述のグラフ化メトリクスでは、データポイント 1 がしきい値内、データポイント 2 が欠落し、データポイント 3 がしきい値を超過し、データポイント 4 が欠落し、データポイント 5 がしきい値を超過しています。評価範囲内に有効なデータポイントが 3 つあるので、このメトリクスの欠落しているデータポイントはゼロになります。欠落しているデータポイントを次のように扱うようにアラームを設定した場合:

- Not breaching — アラームは OK 状態になります。
- Breaching — アラームは OK 状態になります。
- Ignore — アラームは OK 状態になります。
- Missing — アラームは OK 状態になります。

グラフ B

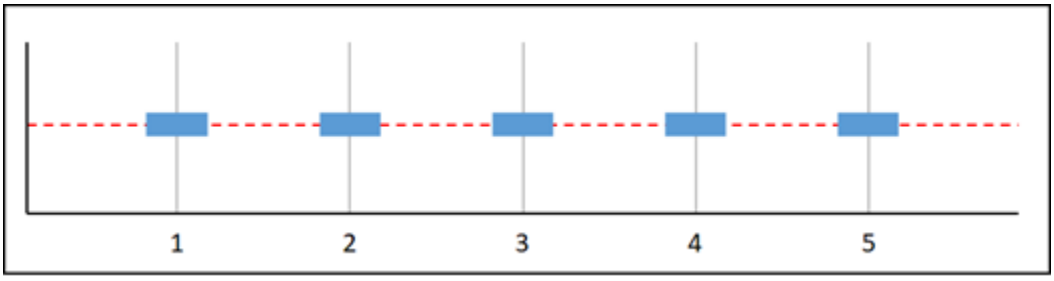


前述のグラフ化メトリクスでは、データポイント 1 がしきい値内にあり、データポイント 2~5 が欠落しています。評価範囲内にデータポイントが 1 つしかないので、このメトリクスには 2 つの欠落データポイントがあります。欠落しているデータポイントを次のように扱うようにアラームを設定した場合:

- Not breaching — アラームは OK 状態になります。
- Breaching — アラームは OK 状態になります。
- Ignore — アラームは OK 状態になります。
- Missing — アラームは OK 状態になります。

このシナリオでは、失われたデータがしきい値を超過として扱われる場合でも、アラームは OK 状態のままになります。これは、1 つの既存のデータポイントがしきい値内であるため、しきい値を超過として扱われる 2 つの欠落データポイントとともに評価されるためです。次回このアラームが評価されるときに、データがまだ欠落している場合は、ALARM に送られます。これは、しきい値内のデータポイントが取得された 5 つの最新のデータポイントに含まれることがなくなったためです。

グラフ C

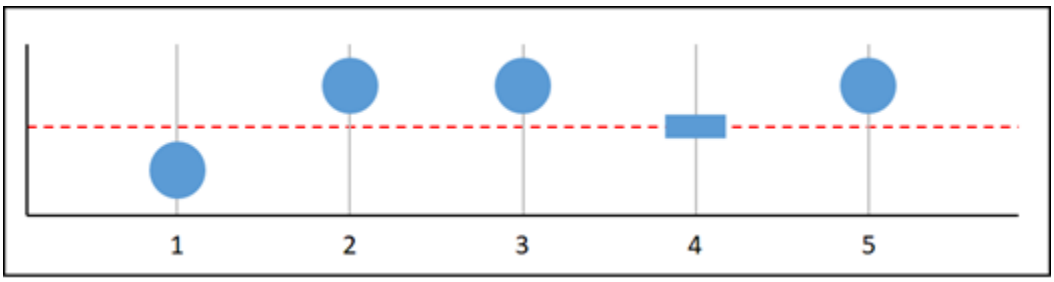


前述のグラフ化メトリクスでは、すべてのデータポイントが欠落しています。評価範囲内のすべてのデータポイントが欠落している場合、このメトリクスには 3 つの欠落データポイントがあります。欠落しているデータポイントを次のように扱うようにアラームを設定した場合:

- Not breaching — アラームは OK 状態になります。

- Breaching — アラームは ALARM 状態になります。
- Ignore — アラームは現在の状態を維持します。
- Missing — アラームは INSUFFICIENT_DATA 状態になります。

グラフ D

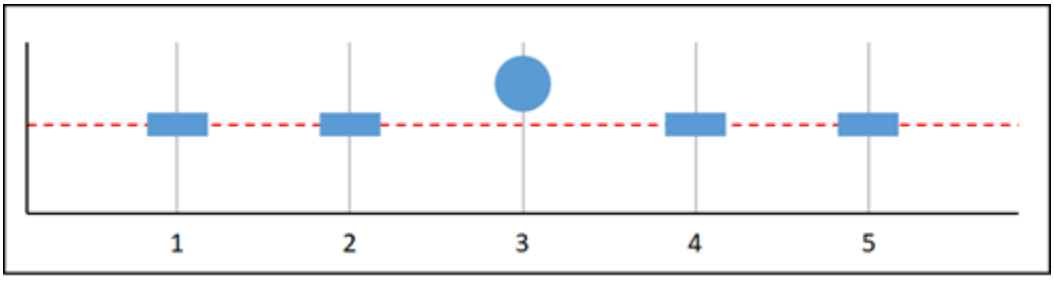


前述のグラフ化メトリクスでは、データポイント 1 がしきい値内、データポイント 2 がしきい値を超過し、データポイント 3 がしきい値を超過し、データポイント 4 が欠落し、データポイント 5 がしきい値を超過しています。評価範囲内に有効なデータポイントが 4 つあるので、このメトリクスの欠落しているデータポイントはゼロになります。欠落しているデータポイントを次のように扱うようにアラームを設定した場合:

- Not breaching — アラームは ALARM 状態になります。
- Breaching — アラームは ALARM 状態になります。
- Ignore — アラームは ALARM 状態になります。
- Missing — アラームは ALARM 状態になります。

このシナリオでは、アラームはすべてのケースで ALARM 状態になります。これは、欠落したデータの処理方法の設定が不要で、無視される十分な実際のデータポイントがあるためです。

グラフ E



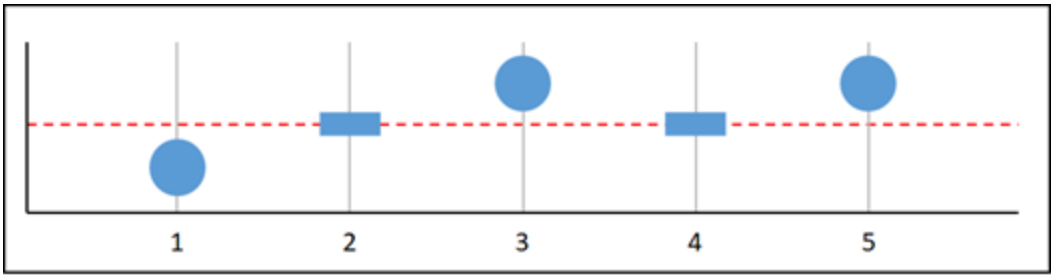
前述のグラフ化メトリクスでは、データポイント 1 と 2 が欠落し、データポイント 3 がしきい値を超過し、データポイント 4 と 5 が欠落しています。評価範囲内にデータポイントが 1 つしかないの

で、このメトリクスには 2 つの欠落データポイントがあります。欠落しているデータポイントを次のように扱うようにアラームを設定した場合:

- Not breaching — アラームは OK 状態になります。
- Breaching — アラームは ALARM 状態になります。
- Ignore — アラームは現在の状態を維持します。
- Missing — アラームは ALARM 状態になります。

グラフF、G、H、I、J では、 アラームへのデータポイントは 2 で、評価期間は 3 です。3 中 2 、N 中 M のアラームです。5 はアラームの評価範囲です。

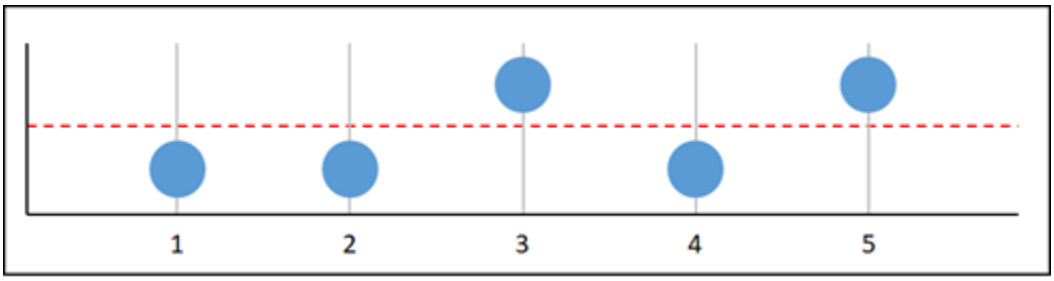
グラフ F



前述のグラフ化メトリクスでは、データポイント 1 がしきい値内で、データポイント 2 が欠落し、データポイント 3 がしきい値を超過し、データポイント 4 が欠落し、データポイント 5 がしきい値を超過しています。評価範囲に 3 つのデータポイントがあるので、このメトリクスの欠落したデータポイントはゼロになります。欠落しているデータポイントを次のように扱うようにアラームを設定した場合:

- Not breaching — アラームは ALARM 状態になります。
- Breaching — アラームは ALARM 状態になります。
- Ignore — アラームは ALARM 状態になります。
- Missing — アラームは ALARM 状態になります。

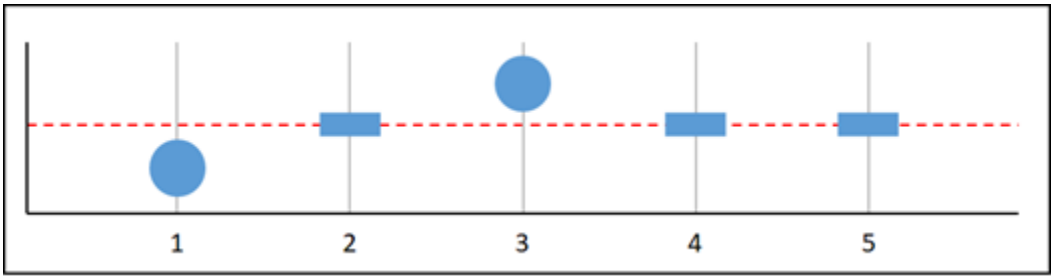
グラフ G



前述のグラフ化メトリクスでは、データポイント 1 と 2 がしきい値内で、データポイント 3 がしきい値を超過し、データポイント 4 がしきい値内で、データポイント 5 がしきい値を超過しています。評価範囲に 5 つのデータポイントがあるので、このメトリクスの欠落データポイントはゼロになります。欠落しているデータポイントを次のように扱うようにアラームを設定した場合:

- Not breaching — アラームは ALARM 状態になります。
- Breaching — アラームは ALARM 状態になります。
- Ignore — アラームは ALARM 状態になります。
- Missing — アラームは ALARM 状態になります。

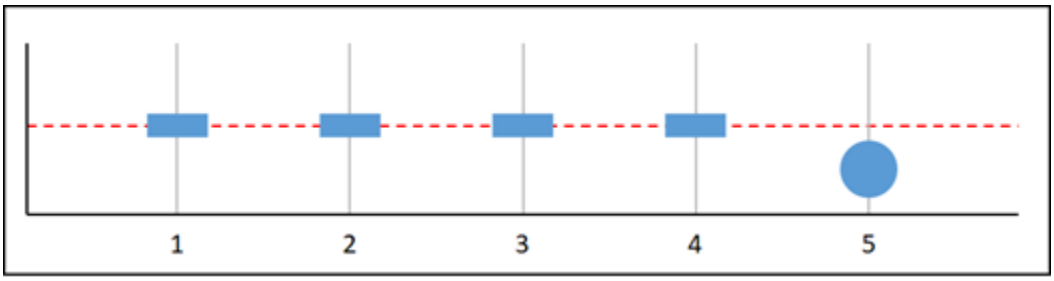
グラフ H



前述のグラフ化メトリクスでは、データポイント 1 がしきい値内で、データポイント 2 が欠落し、データポイント 3 がしきい値を超過し、データポイント 4 と 5 が欠落しています。評価範囲に 2 つのデータポイントがある場合ので、このメトリクスには欠落したデータポイントが 1 つあります。欠落しているデータポイントを次のように扱うようにアラームを設定した場合:

- Not breaching — アラームは OK 状態になります。
- Breaching — アラームは ALARM 状態になります。
- Ignore — アラームは OK 状態になります。
- Missing — アラームは OK 状態になります。

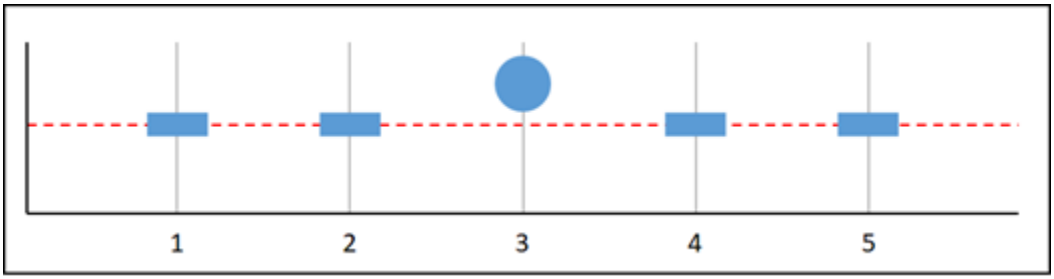
グラフ I



前述のグラフ化メトリクスでは、データポイント 1~4 が欠落し、データポイント 5 がしきい値内です。評価範囲に 1 つのデータポイントがあるので、このメトリクスには 2 つの欠落データポイントがあります。欠落しているデータポイントを次のように扱うようにアラームを設定した場合:

- Not breaching — アラームは OK 状態になります。
- Breaching — アラームは ALARM 状態になります。
- Ignore — アラームは OK 状態になります。
- Missing — アラームは OK 状態になります。

グラフ J



前述のグラフ化メトリクスでは、データポイント 1 と 2 が欠落し、データポイント 3 がしきい値を超過し、データポイント 4 と 5 が欠落しています。評価範囲に 1 つのデータポイントがあるので、このメトリクスには 2 つの欠落データポイントがあります。欠落しているデータポイントを次のように扱うようにアラームを設定した場合:

- Not breaching — アラームは OK 状態になります。
- Breaching — アラームは ALARM 状態になります。
- Ignore — アラームは現在の状態を維持します。
- Missing — アラームは ALARM 状態になります。

アラームの詳細

ここでは、Lightsail でアラームを管理するのに役立つ記事をいくつか紹介します。

- インスタンスのメトリクスアラームを作成する
- データベースのメトリクスにアラームを作成する
- ロードバランサーのメトリックスアラームを作成する
- ディストリビューションのメトリクスにアラームを作成する
- メトリクスアラームの削除または無効化

Lightsail インスタンスのメトリクスアラームを作成する

単一のインスタンスメトリクスを監視する Amazon Lightsail アラームを作成できます。アラームは、指定したしきい値を基準にしたメトリクスの値に基づいて通知するように設定できます。通知は、Lightsail コンソールに表示されるバナー、メールアドレスに送信される メール、携帯電話番号に送信される SMS テキストメッセージです。アラームの詳細については、「アラーム」を参照してください。

目次

- インスタンスアラームの制限
- インスタンスアラームの設定に関するベストプラクティス
- デフォルトのアラーム設定
- Lightsail コンソールを使用してインスタンスのメトリクスアラームを作成する
- Lightsail コンソールを使用してインスタンスのメトリクスアラームをテストする
- インスタンスアラームの作成後の次のステップ

インスタンスアラームの制限

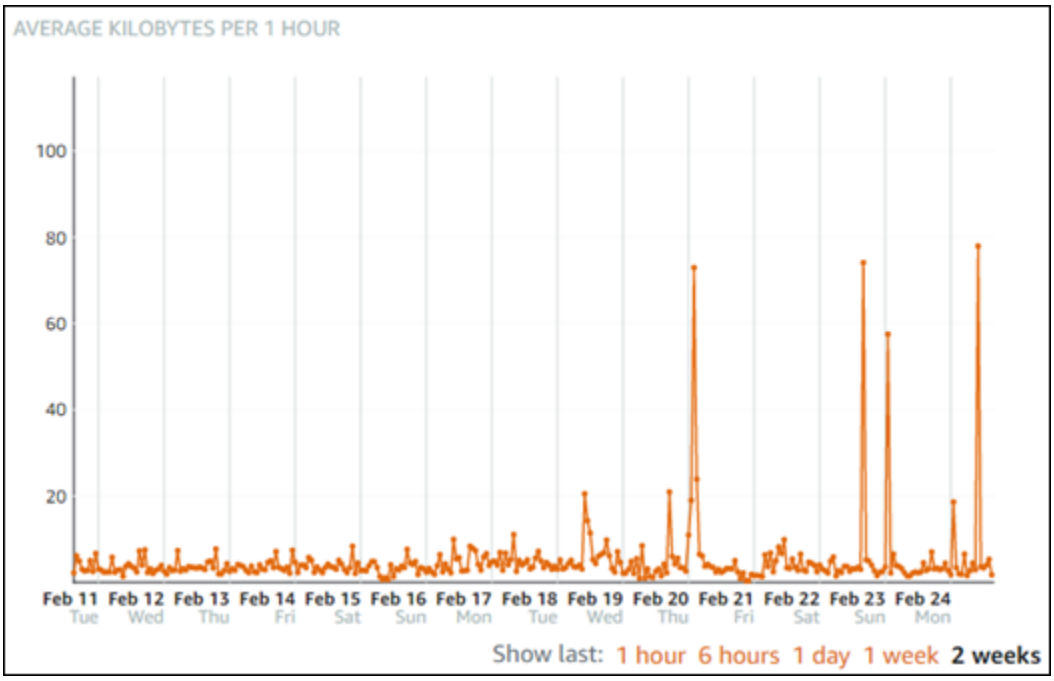
アラームには、以下の制限が適用されます。

- メトリクスごとに 2 つのアラームを設定できます。
- アラームは 5 分間隔で評価され、アラームの各データポイントは、集計されたメトリクスデータの 5 分間隔を表します。
- アラームの状態が OK に変わったときに通知するようにアラームを設定できるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合だけです。

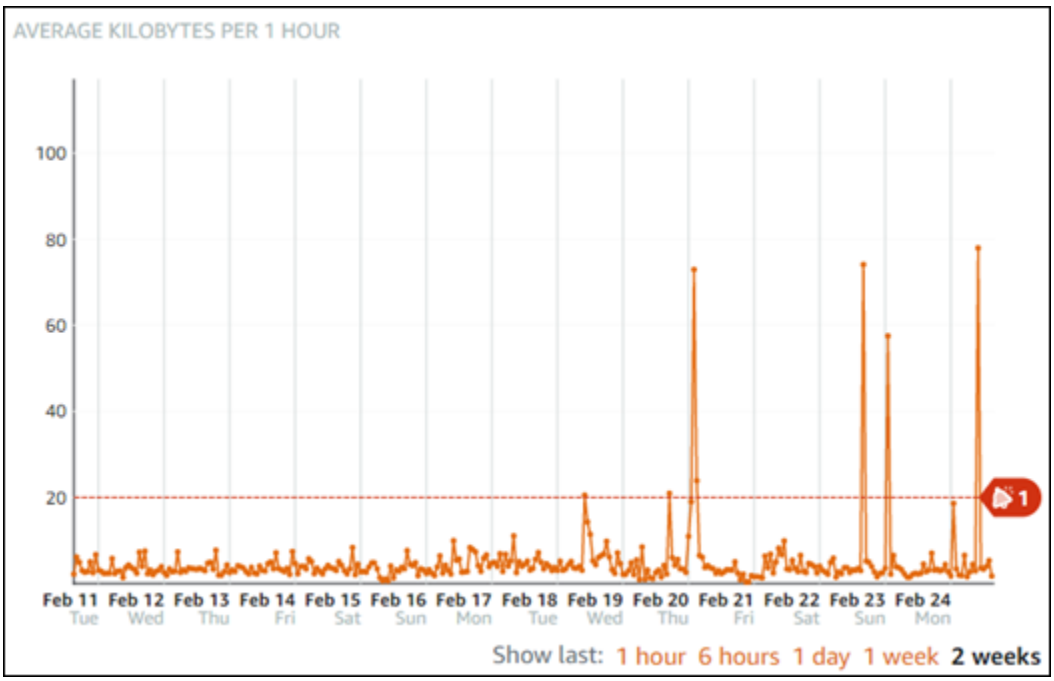
- OK アラーム通知をテストできるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合だけです。
- アラームの状態が INSUFFICIENT_DATA に変わったときに通知するようにアラームを設定できるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合と、欠落しているデータポイントに対して [欠落しているデータを評価しない] オプションを選択した場合だけです。
- 通知をテストできるのは、アラームが OK 状態の場合だけです。

インスタンスアラームの設定に関するベストプラクティス

インスタンスのメトリクスアラームを設定する前に、メトリクスの履歴データを表示する必要があります。過去 2 週間のメトリクスの低レベル、中間レベル、高レベルを識別します。次の発信ネットワークトラフィック (NetworkOut) メトリクスグラフの例では、低レベルは 1 時間あたり 0 ~ 10 KB、中間レベルは 1 時間あたり 10 ~ 20 KB、高レベルは 1 時間あたり 20 ~ 80 KB です。



アラームしきい値を低レベル範囲 (1 時間あたり 5 KB など) 以上に設定すると、アラーム通知が頻繁に送信され、不要なアラーム通知が発生する可能性があります。アラームしきい値を高レベルの範囲 (たとえば、1 時間あたり 20 KB) 以上に設定した場合、アラーム通知の頻度は低くなりますが、調査がさらに意義のあるものになる場合があります。アラームを設定して有効にすると、次の例に示すように、しきい値を表すアラームラインがグラフに表示されます。1 というラベルの付いたアラームラインはアラーム 1 のしきい値を表し、2 というラベルのアラームラインはアラーム 2 のしきい値を表します。



デフォルトのアラーム設定

Lightsail コンソールで新しいアラームを追加すると、デフォルトのアラーム設定があらかじめ入力されています。これは、選択したメトリクスの推奨アラーム設定です。ただし、デフォルトのアラーム設定がリソースに適していることを確認する必要があります。たとえば、インスタンスの発信ネットワークトラフィック (NetworkOut) メトリクスのデフォルトのアラームしきい値は、過去 10 分以内に 2 回 0 バイト[以下]です。ただし、トラフィックの多いイベントの通知を受ける場合は、アラームのしきい値を過去 10 分以内に 2 回 50 KB [以上]に変更するか、これらの設定に 2 番目のアラームを追加して、トラフィックがない場合および、トラフィックが多い場合に通知を受け取るようにします。　指定するしきい値は、このガイドの「インスタンスアラームの設定に関するベストプラクティス」セクションで説明されているように、メトリクスの高レベルと低レベルと一致するように調整する必要があります。

Lightsail コンソールを使用してインスタンスのメトリクスアラームを作成する

Lightsail コンソールを使用してインスタンスメトリクスアラームを作成するには、以下のステップを実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[インスタンス] をクリックします。
3. アラームを作成するインスタンスの名前を選択します。
4. インスタンス管理ページで [Metrics (メトリクス)] タブを選択します。

5. [Metrics Graphs (メトリクスグラフ)] 見出しの下のドロップダウンメニューで、アラームを作成するメトリクスを選択します。詳細については、「リソースのメトリクス」を参照してください。
6. アラームセクションのページでアラームの追加を選択します。
7. ドロップダウンメニューから比較演算子の値を選択します。例の値は、greater than or equal to、greater than、less than、less than or equal to です。
8. アラームのしきい値を入力します。
9. アラームへのデータポイントを入力します。
10. 評価期間を選択します。期間は、5 分から 24 時間まで 5 分単位で指定できます。
11. 次のいずれかの通知方法を選択します。

    - メール — アラームの状態が ALARM に変わると、メールで通知されます。
    - SMS テキストメッセージ — アラームの状態が ALARM に変わると、SMS テキストメッセージによって通知されます。SMS メッセージングは、Lightsail リソースを作成できるすべての AWS リージョンでサポートされているわけではなく、また、一部の国/地域には SMS テキストメッセージを送信できません。詳細については、「SMS テキストメッセージングのサポート」を参照してください。

    > **Note**
    >
    > メールまたは SMS による通知を選択したが、リソースの AWSリージョンで通知の連絡先をまだ設定していない場合は、メールアドレスまたは携帯電話番号を追加する必要があります。詳細については、「メトリクスの通知」を参照してください。

12. (オプション) [Send me a notification when the alarm state change to OK (アラームの状態が OK に変わったときに通知を送信する)] を選択して、アラームの状態が OK に変わったときに通知を受け取ります。このオプションは、メールまたは SMS テキストメッセージで通知されるように選択した場合にのみ使用できます。
13. (オプション) [Advanced settings (詳細設定)] を選択し、次のいずれかのオプションを選択します。

    - アラームが欠落データをどのように扱うかを選択してください。以下のオプションが利用できます。
        - しきい値の範囲内にないと仮定する(しきい値を超過) — 欠落しているデータポイントは「不良」として扱われ、しきい値を超えています。

- しきい値内にあると仮定する (しきい値を超過していない) — 欠落しているデータポイントは、「良い」と見なされ、しきい値の範囲内で処理されます。
- 最後の正常なデータポイントの値を使用する (無視して現在のアラーム状態を維持する) — 現在のアラーム状態が維持されます。
- 評価しない (欠落しているデータを欠落しているデータとして扱う) — 状態を変更するかどうかを評価する際に、欠落データポイントを考慮に入れません。

- [Send a notification if there is insufficient data (データが不足している場合に通知を送信)] を選択して、アラームの状態が INSUFFICIENT_DATA に変わったときに通知を受け取ります。このオプションは、メールまたは SMS テキストメッセージで通知されるように選択した場合にのみ使用できます。

14. [Create (作成)] を選択してアラームを追加します。

    後でアラームを編集するには、編集するアラームの横にある省略記号アイコン (⋮) を選択し、[アラームの編集] を選択します。

Lightsail コンソールを使用してインスタンスのメトリクスアラームをテストする

Lightsail コンソールを使用してアラームをテストするには、次の手順を実行します。アラームをテストして、アラームがトリガーされたときに メールや SMS テキストメッセージを受信するなど、設定された通知オプションが機能していることを確認します。

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [インスタンス] を選択してください。
3. アラームをテストするインスタンスの名前を選択します。
4. インスタンス管理ページで [Metrics (メトリクス)] タブを選択します。
5. [Metrics Graphs (メトリクスグラフ)] 見出しの下のドロップダウンメニューで、アラームをテストするメトリクスを選択します。
6. ページの [アラーム] セクションまでスクロールし、テストするアラームの横にある省略記号アイコン (⋮) を選択します。
7. 以下のオプションのいずれかを選択してください。

    - [アラーム通知のテスト] - このオプションを選択すると、アラームの状態が ALARM に変わったときの通知をテストできます。
    - [OK 通知のテスト] — このオプションを選択すると、アラームの状態が OK に変わったときの通知をテストできます。

> **Note**
>
> これらのオプションのいずれかが使用できない場合は、アラームの通知オプションが設定されていないか、アラームが現在 ALARM 状態になっている可能性があります。詳細については、「インスタンスのアラーム制限」を参照してください。

選択したテストオプションに応じて、アラームが一時的に ALARM または OK 状態に変化し、アラームの通知方法として設定した内容に応じて、メールまたは SMS テキストメッセージが送信されます。通知バナーは、ALARM 通知のテストを選択した場合にのみ Lightsail コンソールに表示されます。OK 通知のテストを選択した場合、通知バナーは表示されません。アラームは、通常、数秒後に実際の状態に戻ります。

次の手順

インスタンスアラームに対して実行できる追加のタスクがいくつかあります。

- 通知の受信を停止するには、Lightsail から メールと携帯電話を削除します。詳細については、「通知連絡先の削除」を参照してください。また、アラームを無効化または削除して、特定のアラームの通知の受信を停止することもできます。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。

Lightsail メトリクスアラームの削除または無効化

アラームを削除して、Amazon Lightsail アラームによってモニタリングされているメトリクスがしきい値を超えたときの通知を停止できます。アラームを無効にして、通知の受信を停止することもできます。詳細については、「アラーム」を参照してください。

目次

- Lightsail コンソールを使用してメトリクスアラームを削除する
- Lightsail コンソールを使用してメトリクスアラームを無効または有効にする

Lightsail コンソールを使用してメトリクスアラームを削除する

Lightsail コンソールを使用してメトリクスアラームを削除するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[インスタンス]、[データベース]、または [ネットワーキング] を選択します。
3. アラームを削除するリソース (インスタンス、データベース、ロードバランサー) の名前を選択します。
4. リソースの管理ページで [メトリクス] タブを選択します。
5. [Metrics Graphs (メトリクスグラフ)] 見出しの下のドロップダウンで、アラームを削除するメトリクスを選択します。
6. ページの [アラーム] セクションまで下にスクロールし、削除するアラームの横にある省略記号アイコン (:) を選択します。
7. [削除] を選択します。
8. プロンプトで、[Delete (削除)] を選択して、アラームを削除することを確定します。

Lightsail コンソールを使用するメトリクスアラームの無効化および有効化

Lightsail コンソールを使用してメトリクスアラームを無効にするには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[インスタンス]、[データベース]、または [ネットワーキング] を選択します。
3. アラームを無効にするリソース (インスタンス、データベース、ロードバランサー) の名前を選択します。
4. リソースの管理ページで [メトリクス] タブを選択します。
5. [Metrics Graphs (メトリクスグラフ)] 見出しの下のドロップダウンで、アラームを無効にするメトリクスを選択します。
6. ページの [Alarms (アラーム)] セクションまで下にスクロールし、無効にするアラームを探し、トグルを選択して無効にします。同様に、無効になっている場合は、トグルを選択して有効にします。

Lightsail バケットのパフォーマンスと使用状況をモニタリングする

Amazon Lightsail オブジェクトストレージサービスでバケットを作成した後、バケットの管理ページの [メトリクス] タブでそのメトリクスグラフを表示できます。メトリクスのモニタリングは、バ

ケットの可用性、パフォーマンスを維持する上で重要なパートです。バケットから定期的にメトリクスデータをモニタリングして収集し、必要に応じてバケットのストレージスペースとネットワーク転送クォータをアップサイズまたはダウンサイズできるようにします。メトリクスの詳細については、「リソースのメトリクス」を参照してください。

リソースを監視するときは、環境内の通常のリソースパフォーマンスのベースラインを確立する必要があります。その後、リソースのパフォーマンスが指定のしきい値を超えたときに通知するように、Lightsail コンソールでアラームを設定できます。詳細については、「通知」および「アラーム」を参照してください。

バケットメトリクス

次のバケットメトリクスが利用可能です。

- [バケットサイズ] — バケットに保存されたデータの量。この値を計算するには、バケット内のすべてのオブジェクト (最新のオブジェクトと最新でないオブジェクトの両方) のサイズを合計します。これには、バケットに対するすべての不完全なマルチパートアップロードのすべてのパートのサイズも含めます。
- オブジェクトの数 — バケットに保存されたオブジェクトの総数。この値を計算するには、バケット内のすべてのオブジェクト (最新のオブジェクトと最新でないオブジェクトの両方) と、バケットに対するすべての不完全なマルチパートアップロードの合計パート数をカウントします。

> **Note**
>
> バケットが空の場合、バケット メトリクス データはレポートされません。

Lightsail コンソールでのバケットメトリクスの表示

Lightsail コンソールでバケットメトリクスを表示するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. メトリクスを表示するバケットの名前を選択します。
4. バケット管理ページで [Metrics] (メトリクス) タブを選択します。
5. [Metrics graphs] (メトリクスグラフ) の見出しの下のドロップダウンメニューで、表示するメトリクスを選択します。

グラフには、選択したメトリクスのデータポイントが視覚的に表示されます。

ScreenshotTBD

メトリクスグラフでは、次のアクションを実行できます。

- グラフの表示を変更して、1 時間、6 時間、1 日、1 週間、2 週間のデータを表示します。
- データポイント上にカーソルを置くと、そのデータポイントに関する詳細情報が表示されます。
- 指定したしきい値をメトリクスが超えたときに通知される、選択したメトリクスのアラームを追加します。詳細については、「アラーム」および「バケットメトリクスアラームの作成」を参照してください。

バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。
3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

   バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

   - Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
   - Amazon Lightsail でのバケットのアクセス許可の設定
   - Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定

- Amazon Lightsail でのバケットのアクセスキーの作成
- Amazon Lightsail でのバケットのリソースアクセスの設定
- Amazon Lightsail でのバケットのクロスアカウントアクセスの設定

5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定
6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail のバケットにファイルをアップロードする
   - マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
   - Amazon Lightsail のバケット内のオブジェクトの表示
   - Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
   - Amazon Lightsail のバケットからのオブジェクトのダウンロード
   - Amazon Lightsail のバケット内のオブジェクトのフィルタリング
   - Amazon Lightsail のバケット内のオブジェクトのタグ付け
   - Amazon Lightsail のバケット内のオブジェクトの削除
9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。

11バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。

12バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。

13ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。

14バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。

- チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
- チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用

15.使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

トピック

- メトリクスアラームによる Lightsail バケットストレージのモニタリング

メトリクスアラームによる Lightsail バケットストレージのモニタリング

単一のバケットメトリクスを監視するAmazon Lightsail アラームを作成できます。アラームは、指定したしきい値を基準にしたメトリクスの値に基づいて通知するように設定できます。通知は、Lightsail コンソールに表示されるバナー、メールアドレスに送信される メール、携帯電話番号に送信される SMS テキストメッセージです。アラームの詳細については、「アラーム」を参照してください。

目次

- バケットアラームの制限
- バケットアラーム設定に関するベストプラクティス
- デフォルトのアラーム設定
- Lightsail コンソールを使用してバケットメトリクスアラームを作成する
- Lightsail コンソールを使用してバケットメトリクスアラームをテストする
- バケットアラームの作成後の次のステップ

バケットアラームの制限

アラームには、以下の制限が適用されます。

- メトリクスごとに 2 つのアラームを設定できます。
- アラームは 5 分間隔で評価され、アラームの各データポイントは、集計されたメトリクスデータの 5 分間隔を表します。
- アラームの状態が OK に変わったときに通知するようにアラームを設定できるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合だけです。
- OK アラーム通知をテストできるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合だけです。
- アラームの状態が INSUFFICIENT_DATA に変わったときに通知するようにアラームを設定できるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合と、欠落しているデータポイントに対して [欠落しているデータを評価しない] オプションを選択した場合だけです。
- 通知をテストできるのは、アラームが OK 状態の場合だけです。

バケットアラームの設定に関するベストプラクティス

バケットのメトリクスアラームを設定する前に、受けたい通知を決めておきます。例えば、バケットサイズメトリクスを念頭に置くと、バケットがほぼ満杯になったときに通知を受けることが可能です。バケットの現在のプランに 5 GB のストレージスペースが含まれている場合は、バケットサイズメトリクスが 4.5 GB に達すると通知されます。バケットプランを増量しなければいけなくなる前に通知されます。

デフォルトのアラーム設定

Lightsail コンソールで新しいアラームを追加すると、デフォルトのアラーム設定があらかじめ入力されています。　これは、選択したメトリクスの推奨アラーム設定です。ただし、デフォルトのアラーム設定がリソースに適していることを確認する必要があります。例えば、バケットサイズのバイトのメトリクスに対するデフォルトのアラームしきい値は 75 GB [以上]です。ただし、ストレージスペースが 5 GB しかないように設定されている場合、リクエストのしきい値はバケットに対して高すぎる可能性があります。アラームのしきい値を、4.5 GB以上にするとよいでしょう。

Lightsail コンソールを使用してバケットメトリクスアラームを作成する

Lightsail コンソールを使用してバケットメトリクスアラームを作成するには、次のステップを実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. 作成するアラームのバケットの名前を選択します。
4. バケット管理ページで メトリクスタブを選択します。
5. [Metrics Graphs (メトリクスグラフ)] 見出しの下のドロップダウンメニューで、アラームを作成するメトリクスを選択します。詳細については、「リソースのメトリクス」を参照してください。
6. アラームセクションのページでアラームの追加を選択します。
7. ドロップダウンメニューから比較演算子の値を選択します。例の値は、greater than or equal to、greater than、less than、less than or equal to です。
8. アラームのしきい値を入力します。
9. アラームへのデータポイントを入力します。
10. 評価期間を選択します。期間は、5 分から 24 時間まで 5 分単位で指定できます。
11. 次のいずれかの通知方法を選択します。

    - メール — アラームの状態が ALARM に変わると、メールで通知されます。
    - SMS テキストメッセージ — アラームの状態が ALARM に変わると、SMS テキストメッセージによって通知されます。SMS メッセージングは、すべての AWS リージョンでサポートされているわけではなく、また、一部の国/地域には SMS テキストメッセージを送信できません。詳細については、「SMS テキストメッセージングのサポート」を参照してください。

    > **Note**
    >
    > メールまたは SMS による通知を選択したが、リソースの AWS リージョンで通知の連絡先をまだ設定していない場合は、メールアドレスまたは携帯電話番号を追加する必要があります。詳細については、「通知」を参照してください。

12. (オプション) [Send me a notification when the alarm state change to OK (アラームの状態が OK に変わったときに通知を送信する)] を選択して、アラームの状態が OK に変わったときに通知を受け取ります。このオプションは、メールまたは SMS テキストメッセージで通知されるように選択した場合にのみ使用できます。
13. (オプション) [Advanced settings (詳細設定)] を選択し、次のいずれかのオプションを選択します。

    - アラームによる欠落データの処理方法を選択します。次のオプションを使用できます。

- しきい値の範囲内にないと仮定する(しきい値を超過) — 欠落しているデータポイントは「不良」として扱われ、しきい値を超えています。
- しきい値内にあると仮定する (しきい値を超過していない) — 欠落しているデータポイントは、「良い」と見なされ、しきい値の範囲内で処理されます。
- 最後の正常なデータポイントの値を使用する (無視して現在のアラーム状態を維持する) — 現在のアラーム状態が維持されます。
- 評価しない (欠落しているデータを欠落しているデータとして扱う) — 状態を変更するかどうかを評価する際に、欠落データポイントを考慮に入れません。

- [Send a notification if there is insufficient data (データが不足している場合に通知を送信)] を選択して、アラームの状態が INSUFFICIENT_DATA に変わったときに通知を受け取ります。このオプションは、メールまたは SMS テキストメッセージで通知されるように選択した場合にのみ使用できます。

14. [Create (作成)] を選択してアラームを追加します。

    後でアラームを編集するには、編集するアラームの横にある省略記号アイコン (⋮) を選択し、[アラームの編集] を選択します。

Lightsail コンソールを使用してバケットメトリクスアラームをテストする

Lightsail コンソールを使用してアラームをテストするには、次の手順を実行します。アラームをテストして、アラームがトリガーされたときに メールや SMS テキストメッセージを受信するなど、設定された通知オプションが機能していることを確認します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. アラームをテストするバケットの名前を選択します。
4. バケット管理ページで メトリクスタブを選択します。
5. [Metrics Graphs (メトリクスグラフ)] 見出しの下のドロップダウンメニューで、アラームをテストするメトリクスを選択します。
6. ページの [アラーム] セクションまでスクロールし、テストするアラームの横にある省略記号アイコン (⋮) を選択します。
7. 以下のオプションのいずれかを選択してください。

    - [アラーム通知のテスト] - このオプションを選択すると、アラームの状態が ALARM に変わったときの通知をテストできます。

- [OK 通知のテスト] — このオプションを選択すると、アラームの状態が OK に変わったときの通知をテストできます。

> **Note**
>
> これらのオプションのいずれかが使用できない場合は、アラームの通知オプションが設定されていないか、アラームが現在 ALARM 状態になっている可能性があります。詳細については、バケットアラームの制限を参照してください。

選択したテストオプションに応じて、アラームが一時的に ALARM または OK 状態に変化し、アラームの通知方法として設定した内容に応じて、メールまたは SMS テキストメッセージが送信されます。通知バナーは、ALARM 通知のテストを選択した場合にのみ Lightsail コンソールに表示されます。OK 通知のテストを選択した場合、通知バナーは表示されません。アラームは、通常、数秒後に実際の状態に戻ります。

バケットアラームの作成後の次のステップ

バケットアラームに対して実行できる追加のタスクがいくつかあります。

- 通知の受信を停止するには、Lightsail から メールと携帯電話を削除します。詳細については、「通知連絡先の削除」を参照してください。また、アラームを無効化または削除して、特定のアラームの通知の受信を停止することもできます。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。

Lightsail コンテナサービスリソースの使用率をモニタリングする

Amazon Lightsail コンテナサービスの作成後、サービスの管理ページの [メトリクス] タブでメトリクスグラフを表示できます。メトリクスのモニタリングは、リソースの信頼性、可用性、パフォーマンスを維持する上で重要な要素です。リソースから定期的にメトリクスデータをモニタリングして収集し、マルチポイント障害が発生した場合に、より簡単にデバッグできるようにします。メトリクスの詳細については、「Amazon Lightsail のメトリクス」を参照してください。

リソースを監視するときは、環境内の通常のリソースパフォーマンスのベースラインを確立する必要があります。

> **Note**
>
> アラームと通知は、現在、コンテナサービスメトリクスではサポートされていません。

コンテナサービスのメトリクス

以下のコンテナサービスメトリクスを使用できます。

- CPU 使用率 — コンテナサービスの全ノードで現在使用されている、コンピュータ単位の平均比率。このメトリクスは、コンテナサービス上のコンテナを実行するのに必要な処理能力を特定します。
- メモリ使用率 — コンテナサービスの全ノードで現在使用されているメモリの平均比率。このメトリクスは、コンテナサービスでコンテナを実行するのに必要なメモリを表します。

> **Note**
>
> 新しいデプロイを作成すると、コンテナサービスの既存の使用率メトリクスが消え、新しい現在のデプロイのメトリクスだけが表示されます。

Lightsail コンソールでコンテナサービスのメトリクスを表示する

Lightsail コンソールでコンテナサービスメトリクスを表示するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [コンテナ] を選択します。
3. メトリクスを表示するコンテナの名前を選択します。
4. コンテナ サービス 管理ページで [Metrics] (メトリクス) タブを選択します。
5. [メトリクス] グラフの見出しの下のドロップダウンメニューで、表示するメトリクスを選択します。

   グラフには、選択したメトリクスのデータポイントが視覚的に表示されます。
6. メトリクスグラフでは、次のアクションを実行できます。

   - グラフの表示を変更して、1 時間、6 時間、1 日、1 週間、2 週間のデータを表示します。

- データポイント上にカーソルを置くと、そのデータポイントに関する詳細情報が表示されます。

> **Note**
>
> アラームと通知は、現在、コンテナサービスメトリクスではサポートされていません。

Lightsail データベースパフォーマンスメトリクスをモニタリングする

Amazon Lightsail でデータベースを起動すると、データベースの管理ページの [Metrics (メトリクス)] タブでメトリクスグラフを表示できるようになります。メトリクスのモニタリングは、リソースの信頼性、可用性、パフォーマンスを維持する上で重要な要素です。リソースから定期的にメトリクスデータをモニタリングして収集し、マルチポイント障害が発生した場合に、より簡単にデバッグできるようにします。メトリクスの詳細については、「メトリクス」を参照してください。

リソースを監視するときは、環境内の通常のリソースパフォーマンスのベースラインを確立する必要があります。ベースラインを確立したら、リソースのパフォーマンスが指定のしきい値を超えたときに通知するように Lightsail コンソールでアラームを設定できます。詳細については、「通知」および「アラーム」を参照してください。

目次

- データベースメトリクス
- データベースメトリクスを表示する
- データベースメトリクスの表示後の次のステップ

データベースメトリクス

次のデータベースメトリクスを使用できます。

- CPU 使用率 (**CPUUtilization**) — データベースで現在使用されている CPU 使用率の割合。
- データベース接続 (**DatabaseConnections**) — 使用中のデータベース接続の数。
- ディスクのキューの深度 (**DiskQueueDepth**) — ディスクへのアクセスを待機している未処理の IO (読み取り/書き込みリクエスト) の数。

- 空きストレージ容量 (**FreeStorageSpace**) — 使用可能なストレージの容量。
- ネットワーク受信スループット (**NetworkReceiveThroughput**) — モニタリングとレプリケーションに使用する顧客データベーストラフィックと AWS トラフィックの両方を含む、データベースの受信ネットワークトラフィック。
- ネットワーク送信スループット (**NetworkTransmitThroughput**) — モニタリングとレプリケーションに使用する顧客データベーストラフィックと AWS トラフィックの両方を含む、データベースの送信ネットワークトラフィック。

Lightsail コンソールでのデータベースメトリクスの表示

Lightsail コンソールでデータベースメトリクスを表示するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [データベース] を選択します。
3. メトリクスを表示するデータベースの名前を選択します。
4. データベース管理ページで [Metrics (メトリクス)] タブを選択します。
5. [Metrics graph (メトリクスグラフ)] 見出しの下のドロップダウンメニューで、表示するメトリクスを選択します。

   グラフには、選択したメトリクスのデータポイントが視覚的に表示されます。
6. メトリクスグラフでは、次のアクションを実行できます。

   - グラフの表示を変更して、1 時間、6 時間、1 日、1 週間、2 週間のデータを表示します。
   - データポイント上にカーソルを置くと、そのデータポイントに関する詳細情報が表示されます。
   - 指定したしきい値をメトリクスが超えたときに通知される、選択したメトリクスのアラームを追加します。詳細については、「アラーム」および「データベースメトリクスアラームの作成」を参照してください。

データベースメトリクスの表示後の次のステップ

データベースメトリクスに対して実行できる追加のタスクがいくつかあります。

- 指定したしきい値をメトリクスが超えたときに通知される、選択したメトリクスのアラームを追加します。詳細については、「アラーム」および「データベースメトリクスアラームの作成」を参照してください。

- アラームが作動すると、通知バナーが Lightsail コンソールに表示されます。E メールと SMS テキストメッセージで通知を受け取るには、リソースをモニタリングする各 AWS リージョン に通知連絡先として E メールアドレスと携帯電話番号を追加する必要があります。詳細については、「通知連絡先の追加」を参照してください 。
- 通知の受信を停止するには、Lightsail から メールと携帯電話を削除します。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。また、アラームを無効化または削除して、特定のアラームの通知の受信を停止することもできます。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。

トピック

- メトリクスアラームで Lightsail データベースの状態をモニタリングする

メトリクスアラームで Lightsail データベースの状態をモニタリングする

1 つのデータベースメトリクスを監視する Amazon Lightsail アラームを作成できます。アラームは、指定したしきい値を基準にしたメトリクスの値に基づいて通知するように設定できます。通知は、Lightsail コンソールに表示されるバナー、メールアドレスに送信される メール、携帯電話番号に送信される SMS テキストメッセージです。アラームの詳細については、「アラーム」を参照してください。

目次

- データベースアラームの制限
- データベースアラームの設定に関するベストプラクティス
- デフォルトのアラーム設定
- Lightsail コンソールを使用してデータベースのメトリクスアラームを作成する
- Lightsail コンソールを使用してデータベースのメトリクスアラームをテストする
- データベースアラームの作成後の次の手順

データベースアラームの制限

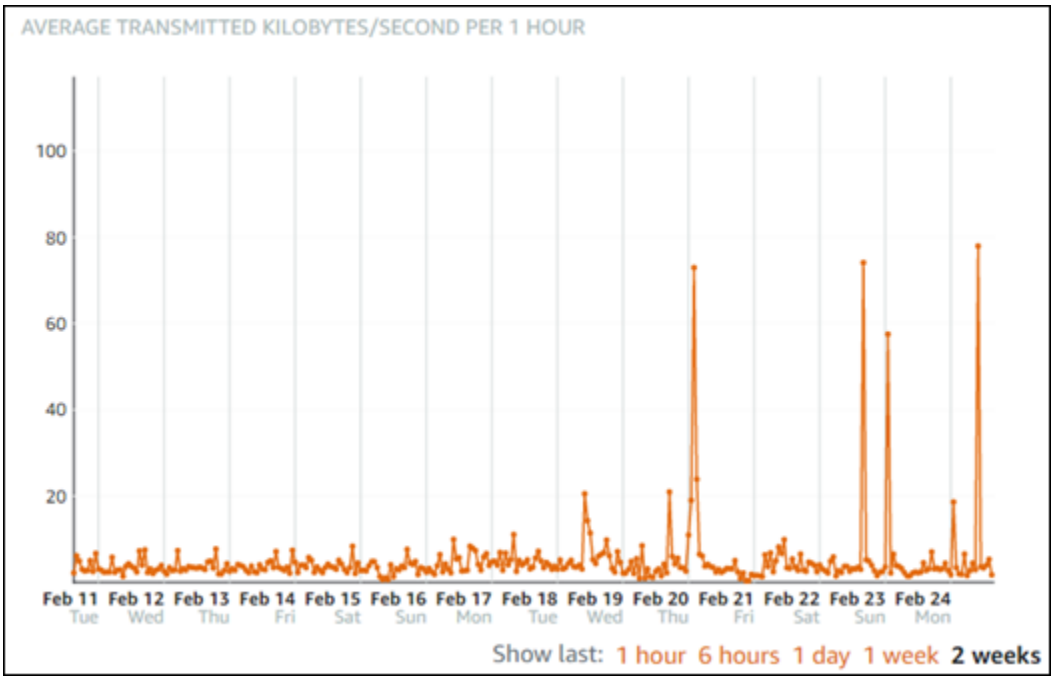
アラームには、以下の制限が適用されます。

- メトリクスごとに 2 つのアラームを設定できます。

- アラームは 5 分間隔で評価され、アラームの各データポイントは、集計されたメトリクスデータの 5 分間隔を表します。
- アラームの状態が OK に変わったときに通知するようにアラームを設定できるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合だけです。
- OK アラーム通知をテストできるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合だけです。
- アラームの状態が INSUFFICIENT_DATA に変わったときに通知するようにアラームを設定できるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合と、欠落しているデータポイントに対して [欠落しているデータを評価しない] オプションを選択した場合だけです。
- 通知をテストできるのは、アラームが OK 状態の場合だけです。

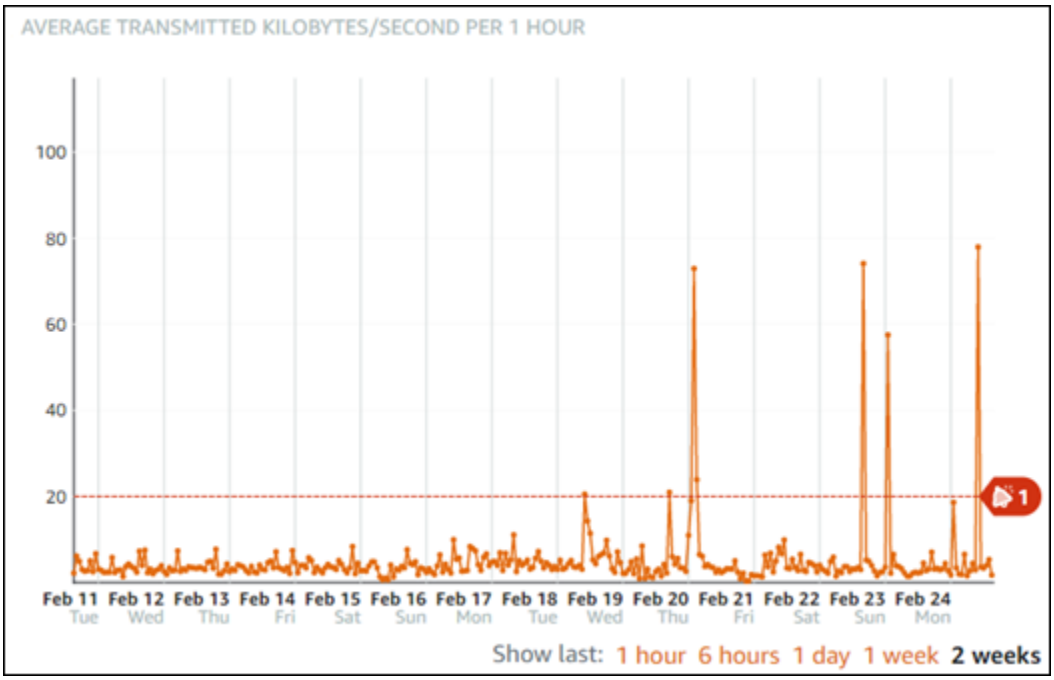
データベースアラームの設定に関するベストプラクティス

データベースのメトリクスアラームを設定する前に、メトリクスの履歴データを表示する必要があります。過去 2 週間のメトリクスの低レベル、中間レベル、高レベルを識別します。次のネットワーク送信スループット (NetworkTransmitThroughput) メトリクスグラフの例では、低レベルは 0 ~ 10 KB/秒、中間レベルは 1 時間あたり 10 ~ 20 KB/秒、高レベルは 1 時間あたり 20 ~ 80 KB/秒の間です。



アラームしきい値を低レベル範囲 (1 時間あたり 5 KB/秒など) 以上に設定すると、より頻繁に、不要なアラーム通知が送信されます。アラームしきい値を高レベルの範囲 (たとえば、1 時間あたり 20

KB) 以上に設定した場合、アラーム通知の頻度は低くなりますが、調査がさらに意義のあるものになる場合があります。アラームを設定して有効にすると、次の例に示すように、しきい値を表すアラームラインがグラフに表示されます。1 というラベルの付いたアラームラインはアラーム 1 のしきい値を表し、2 というラベルのアラームラインはアラーム 2 のしきい値を表します。



デフォルトのアラーム設定

Lightsail コンソールで新しいアラームを追加すると、デフォルトのアラーム設定があらかじめ入力されています。これは、選択したメトリクスの推奨アラーム設定です。ただし、デフォルトのアラーム設定がリソースに適していることを確認する必要があります。たとえば、空きストレージ容量 (`FreeStorageSpace`) メトリクスのデフォルトのアラームしきい値は、過去 5 分間 1 回で 5 バイト [未満]です。ただし、その空きストレージ容量のしきい値は、データベースに対して低すぎる可能性があります。アラームのしきい値を、過去 5 分以内に 1 回で 4 GB 未満に変更することもできます。

Lightsail コンソールを使用してデータベースのメトリクスアラームを作成する

Lightsail コンソールを使用してデータベースメトリクスアラームを作成するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [データベース] を選択します。
3. アラームを作成するデータベースの名前を選択します。

4. データベース管理ページで [Metrics (メトリクス)] タブを選択します。
5. [Metrics Graphs (メトリクスグラフ)] 見出しの下のドロップダウンメニューで、アラームを作成するメトリクスを選択します。詳細については、「リソースのメトリクス」を参照してください。
6. アラームセクションのページでアラームの追加を選択します。
7. ドロップダウンメニューから比較演算子の値を選択します。例の値は、greater than or equal to、greater than、less than、less than or equal to です。
8. アラームのしきい値を入力します。
9. アラームへのデータポイントを入力します。
10. 評価期間を選択します。期間は、5 分から 24 時間まで 5 分単位で指定できます。
11. 次のいずれかの通知方法を選択します。

    - メール — アラームの状態が ALARM に変わると、メールで通知されます。
    - SMS テキストメッセージ — アラームの状態が ALARM に変わると、SMS テキストメッセージによって通知されます。SMS メッセージングは、Lightsail リソースを作成できるすべての AWS リージョンでサポートされているわけではなく、また、一部の国/地域には SMS テキストメッセージを送信できません。詳細については、「SMS テキストメッセージングのサポート」を参照してください。

    > **Note**
    >
    > メールまたは SMS による通知を選択したが、リソースの AWSリージョンで通知の連絡先をまだ設定していない場合は、メールアドレスまたは携帯電話番号を追加する必要があります。詳細については、「通知」を参照してください。

12. (オプション) [Send me a notification when the alarm state change to OK (アラームの状態が OK に変わったときに通知を送信する)] を選択して、アラームの状態が OK に変わったときに通知を受け取ります。このオプションは、メールまたは SMS テキストメッセージで通知されるように選択した場合にのみ使用できます。
13. (オプション) [Advanced settings (詳細設定)] を選択し、次のいずれかのオプションを選択します。

    - アラームによる欠落データの処理方法を選択します。次のオプションを使用できます。
      - しきい値の範囲内にないと仮定する(しきい値を超過) — 欠落しているデータポイントは「不良」として扱われ、しきい値を超えています。

- しきい値内にあると仮定する (しきい値を超過していない) — 欠落しているデータポイントは、「良い」と見なされ、しきい値の範囲内で処理されます。
- 最後の正常なデータポイントの値を使用する (無視して現在のアラーム状態を維持する) — 現在のアラーム状態が維持されます。
- 評価しない (欠落しているデータを欠落しているデータとして扱う) — 状態を変更するかどうかを評価する際に、欠落データポイントを考慮に入れません。

- [Send a notification if there is insufficient data (データが不足している場合に通知を送信)] を選択して、アラームの状態が INSUFFICIENT_DATA に変わったときに通知を受け取ります。このオプションは、メールまたは SMS テキストメッセージで通知されるように選択した場合にのみ使用できます。

14. [Create (作成)] を選択してアラームを追加します。

    後でアラームを編集するには、編集するアラームの横にある省略記号アイコン (⋮) を選択し、[アラームの編集] を選択します。

Lightsail コンソールを使用したデータベースメトリクスアラームのテスト

Lightsail コンソールを使用してアラームをテストするには、次の手順を実行します。アラームをテストして、アラームがトリガーされたときに メールや SMS テキストメッセージを受信するなど、設定された通知オプションが機能していることを確認します。

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [データベース] を選択します。
3. アラームを表示するデータベースの名前を選択します。
4. データベース管理ページで [Metrics (メトリクス)] タブを選択します。
5. [Metrics Graphs (メトリクスグラフ)] 見出しの下のドロップダウンメニューで、アラームをテストするメトリクスを選択します。
6. ページの [アラーム] セクションまでスクロールし、テストするアラームの横にある省略記号アイコン (⋮) を選択します。
7. 以下のオプションのいずれかを選択してください。

    - [アラーム通知のテスト] - このオプションを選択すると、アラームの状態が ALARM に変わったときの通知をテストできます。
    - [OK 通知のテスト] — このオプションを選択すると、アラームの状態が OK に変わったときの通知をテストできます。

> **Note**
>
> これらのオプションのいずれかが使用できない場合は、アラームの通知オプションが設定されていないか、アラームが現在 ALARM 状態になっている可能性があります。詳細については、「データベースアラームの制限」を参照してください。

選択したテストオプションに応じて、アラームが一時的に ALARM または OK 状態に変化し、アラームの通知方法として設定した内容に応じて、メールまたは SMS テキストメッセージが送信されます。通知バナーは、ALARM 通知のテストを選択した場合にのみ Lightsail コンソールに表示されます。OK 通知のテストを選択した場合、通知バナーは表示されません。アラームは、通常、数秒後に実際の状態に戻ります。

データベースアラームの作成後の次の手順

データベースアラームに対して実行できる追加のタスクがいくつかあります。

- 通知の受信を停止するには、Lightsail から メールと携帯電話を削除します。詳細については、「通知連絡先の削除」を参照してください。また、アラームを無効化または削除して、特定のアラームの通知の受信を停止することもできます。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。

Lightsail ディストリビューションパフォーマンスメトリクスをモニタリングする

Amazon Lightsail でディストリビューションを作成した後、 ディストリビューションの管理ページの [Metrics] (メトリクス) タブで、そのメトリクスのグラフを見ることができます。メトリクスのモニタリングは、リソースの信頼性、可用性、パフォーマンスを維持する上で重要な要素です。リソースから定期的にメトリクスデータをモニタリングして収集し、マルチポイント障害が発生した場合に、より簡単にデバッグできるようにします。メトリクスの詳細については、「メトリクス」を参照してください。

リソースを監視するときは、環境内の通常のリソースパフォーマンスのベースラインを確立する必要があります。その後、リソースのパフォーマンスが指定のしきい値を超えたときに通知するよう

に、Lightsail コンソールでアラームを設定できます。詳細については、「通知」および「アラーム」を参照してください。

目次

- ディストリビューションメトリクス
- ディストリビューションメトリクスを Lightsail コンソールに表示する
- ディストリビューションメトリクス表示後の次のステップ

ディストリビューションメトリクス

次のディストリビューションメトリクスが利用可能です。

- リクエスト — すべての HTTP メソッド、および HTTP と HTTPS 両方のリクエストについて、ディストリビューションが受信したビューワーリクエストの総数。
- アップロードされたバイト数 — POST リクエストと PUT リクエストを使用して、ディストリビューションによってオリジンにアップロードされたバイト数。
- ダウンロードされたバイト数 — GET リクエスト、HEAD リクエスト、および OPTIONS リクエストに対してビューワーがダウンロードしたバイト数。
- トータルエラー率 — レスポンスの HTTP ステータスコードが 4xx または 5xx であったすべてのビューワーリクエストの割合 (%)。
- HTTP 4xx トータルエラー率 — レスポンスの HTTP ステータスコードが 4xx であったすべてのビューワーリクエストの割合 (%)。このような場合、クライアントまたはクライアントビューワーでエラーが発生した可能性があります。たとえば、ステータスコード 404 (Not Found) は、クライアントが、検出できないオブジェクトをリクエストしたことを意味します。
- HTTP 5xx トータルエラー率 — レスポンスの HTTP ステータスコードが 5xx であったすべてのビューワー リクエストの割合 (%)。このような場合、オリジンサーバーはリクエストを満たしませんでした。たとえば、ステータスコード 503 (Service Unavailable) は、オリジンサーバーが現在利用できないことを意味します。

ディストリビューションメトリクスを Lightsail コンソールに表示する

Lightsail コンソールでディストリビューションメトリクスを表示するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。

3. メトリクスを表示するディストリビューションの名前を選択します。
4. ディストリビューション管理ページで [Metrics] (メトリクス) タブを選択します。
5. [Metrics graph (メトリクスグラフ)] 見出しの下のドロップダウンメニューで、表示するメトリクスを選択します。

   グラフには、選択したメトリクスのデータポイントが視覚的に表示されます。
6. メトリクスグラフでは、次のアクションを実行できます。

   - グラフの表示を変更して、1 時間、6 時間、1 日、1 週間、2 週間のデータを表示します。
   - データポイント上にカーソルを置くと、そのデータポイントに関する詳細情報が表示されます。
   - 指定したしきい値をメトリクスが超えたときに通知される、選択したメトリクスのアラームを追加します。詳細については、「アラーム」および「インスタンスのメトリクスアラームを作成する」を参照してください。

ディストリビューションメトリクスの表示後の次のステップ

ディストリビューションメトリクスに対して実行できる追加のタスクがいくつかあります。

- 指定したしきい値をメトリクスが超えたときに通知される、選択したメトリクスのアラームを追加します。詳細については、「アラーム」および「ディストリビューションメトリクスアラームの作成」を参照してください。
- アラームが作動すると、通知バナーが Lightsail コンソールに表示されます。E メールと SMS テキストメッセージで通知を受け取るには、リソースをモニタリングする各 AWS リージョン に通知連絡先として E メールアドレスと携帯電話番号を追加する必要があります。詳細については、「通知連絡先を追加する」を参照してください。
- 通知の受信を停止するには、Lightsail から メールと携帯電話を削除します。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。また、アラームを無効化または削除して、特定のアラームの通知の受信を停止することもできます。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。

トピック

- メトリクスアラームで Lightsail ディストリビューションの状態をモニタリングする

メトリクスアラームで Lightsail ディストリビューションの状態をモニタリングする

単一のディストリビューションメトリクスをモニタリングする Amazon Lightsail アラームを作成できます。アラームは、指定したしきい値を基準にしたメトリクスの値に基づいて通知するように設定できます。通知は、Lightsail コンソールに表示されるバナー、メールアドレスに送信される メール、携帯電話番号に送信される SMS テキストメッセージです。アラームの詳細については、「アラーム」を参照してください。

目次

- ディストリビューションアラームの制限
- ディストリビューションアラームの設定に関するベストプラクティス
- デフォルトのアラーム設定
- Lightsail コンソールを使用してディストリビューションメトリクスアラームを作成する
- ディストリビューションメトリクスアラームをテストする
- ディストリビューションアラーム作成後の次のステップ

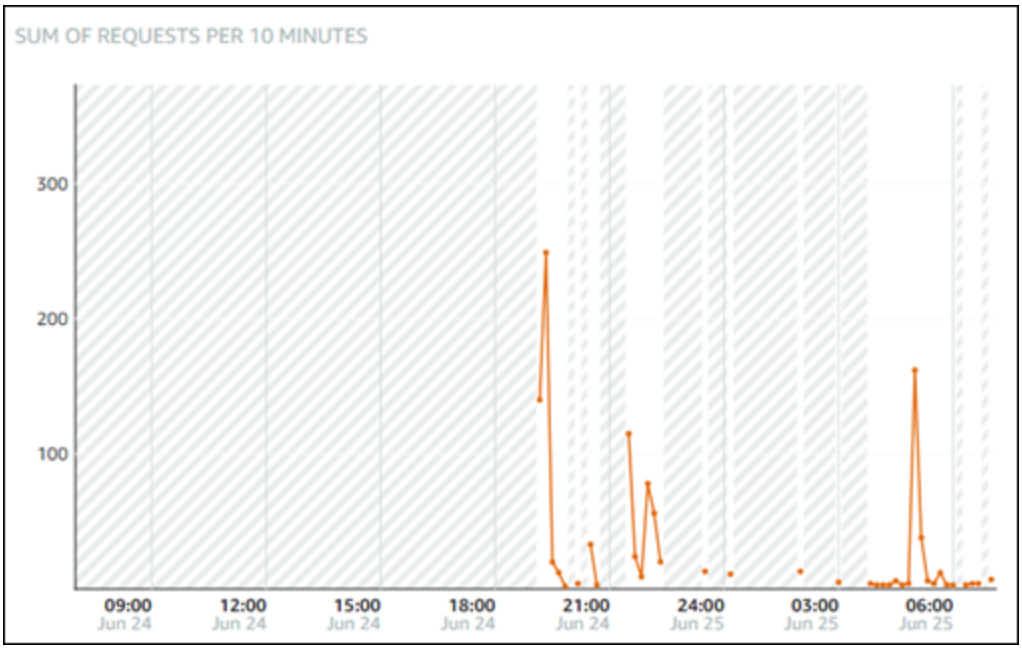
ディストリビューションアラームの制限

アラームには、以下の制限が適用されます。

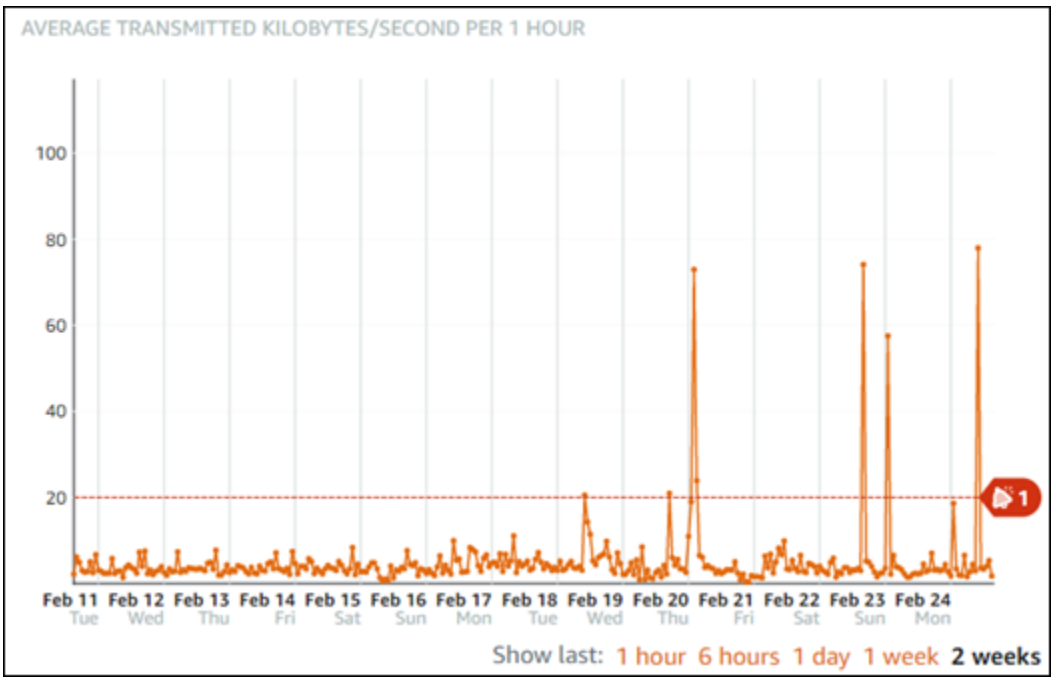
- メトリクスごとに 2 つのアラームを設定できます。
- アラームは 5 分間隔で評価され、アラームの各データポイントは、集計されたメトリクスデータの 5 分間隔を表します。
- アラームの状態が OK に変わったときに通知するようにアラームを設定できるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合だけです。
- OK アラーム通知をテストできるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合だけです。
- アラームの状態が `INSUFFICIENT_DATA` に変わったときに通知するようにアラームを設定できるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合と、欠落しているデータポイントに対して [欠落しているデータを評価しない] オプションを選択した場合だけです。
- 通知をテストできるのは、アラームが OK 状態の場合だけです。

ディストリビューションアラームの設定に関するベストプラクティス

ディストリビューションのメトリクスアラームを設定する前に、メトリクスのデータ履歴を表示する必要があります。過去 2 週間のメトリクスの低レベル、中間レベル、高レベルを識別します。以下のリクエストでメトリクスグラフの例では、低レベルは 0～10 のリクエスト、中間レベルは 10～50 リクエスト、高レベルは 50～250 リクエストとなります。



アラームを低レベル範囲 (例えば 5 リクエスト) 以上に設定すると、アラーム通知が頻繁に送信され、不要なアラーム通知が発生する可能性があります。アラームしきい値を高レベルの範囲 (例えば 150 リクエスト) 以上に設定した場合、アラーム通知の頻度は低くなりますが、さらに調査することが重要になります。アラームを設定して有効にすると、次の例に示すように、しきい値を表すアラームラインがグラフに表示されます。1 というラベルの付いたアラームラインはアラーム 1 のしきい値を表し、2 というラベルのアラームラインはアラーム 2 のしきい値を表します。



デフォルトのアラーム設定

Lightsail コンソールで新しいアラームを追加すると、デフォルトのアラーム設定があらかじめ入力されています。これは、選択したメトリクスの推奨アラーム設定です。ただし、デフォルトのアラーム設定がリソースに適していることを確認する必要があります。リクエストのメトリクスのデフォルトのアラームしきい値が、過去 15 分以内に 3 回で 45 リクエスト[以上]の場合。リクエストのしきい値は、ディストリビューションに対して低すぎる可能性があります。アラームのしきい値を、過去 15 分以内に 3 回で 150 リクエスト以上に変更することが望ましいです。

Lightsail コンソールを使用してディストリビューションメトリクスアラームを作成する

Lightsail コンソールを使用してディストリビューションメトリクスアラームを作成するには、以下のステップを実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. アラームを作成する対象のディストリビューションを選択します。
4. ディストリビューション管理ページでメトリクスタブを選択します。
5. [Metrics Graphs (メトリクスグラフ)] 見出しの下のドロップダウンメニューで、アラームを作成するメトリクスを選択します。詳細については、「リソースのメトリクス」を参照してください。

6. アラームセクションのページでアラームの追加を選択します。
7. ドロップダウンメニューから比較演算子の値を選択します。例の値は、greater than or equal to、greater than、less than、less than or equal to です。
8. アラームのしきい値を入力します。
9. アラームへのデータポイントを入力します。
10. 評価期間を選択します。期間は、5 分から 24 時間まで 5 分単位で指定できます。
11. 次のいずれかの通知方法を選択します。

    - メール — アラームの状態が ALARM に変わると、メールで通知されます。
    - SMS テキストメッセージ — アラームの状態が ALARM に変わると、SMS テキストメッセージによって通知されます。SMS メッセージングは、Lightsail リソースを作成できるすべての AWS リージョンでサポートされているわけではなく、また、一部の国/地域には SMS テキストメッセージを送信できません。詳細については、「SMS テキストメッセージングのサポート」を参照してください。

    > **Note**
    >
    > メールまたは SMS による通知を選択したが、リソースの AWS リージョンで通知の連絡先をまだ設定していない場合は、メールアドレスまたは携帯電話番号を追加する必要があります。詳細については、「通知」を参照してください。

12. (オプション) [Send me a notification when the alarm state change to OK (アラームの状態が OK に変わったときに通知を送信する)] を選択して、アラームの状態が OK に変わったときに通知を受け取ります。このオプションは、メールまたは SMS テキストメッセージで通知されるように選択した場合にのみ使用できます。
13. (オプション) [Advanced settings (詳細設定)] を選択し、次のいずれかのオプションを選択します。

    - アラームによる欠落データの処理方法を選択します。次のオプションを使用できます。
      - しきい値の範囲内にないと仮定する(しきい値を超過) — 欠落しているデータポイントは「不良」として扱われ、しきい値を超えています。
      - しきい値内にあると仮定する (しきい値を超過していない) — 欠落しているデータポイントは、「良い」と見なされ、しきい値の範囲内で処理されます。
      - 最後の正常なデータポイントの値を使用する (無視して現在のアラーム状態を維持する) — 現在のアラーム状態が維持されます。

- 評価しない (欠落しているデータを欠落しているデータとして扱う) — 状態を変更するかどうかを評価する際に、欠落データポイントを考慮に入れません。

- [Send a notification if there is insufficient data (データが不足している場合に通知を送信)] を選択して、アラームの状態が INSUFFICIENT_DATA に変わったときに通知を受け取ります。このオプションは、メールまたは SMS テキストメッセージで通知されるように選択した場合にのみ使用できます。

14. [Create (作成)] を選択してアラームを追加します。

    後でアラームを編集するには、編集するアラームの横にある省略記号アイコン (⋮) を選択し、[アラームの編集] を選択します。

ディストリビューションメトリクスアラームをテストする

Lightsail コンソールを使用してアラームをテストするには、次の手順を実行します。アラームをテストして、アラームがトリガーされたときに メールや SMS テキストメッセージを受信するなど、設定された通知オプションが機能していることを確認します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. アラームをテストしたいディストリビューションの名前を選択します。
4. ディストリビューション管理ページでメトリクスタブを選択します。
5. [Metrics Graphs (メトリクスグラフ)] 見出しの下のドロップダウンメニューで、アラームをテストするメトリクスを選択します。
6. ページの [アラーム] セクションまでスクロールし、テストするアラームの横にある省略記号アイコン (⋮) を選択します。
7. 以下のオプションのいずれかを選択してください。

    - [アラーム通知のテスト] - このオプションを選択すると、アラームの状態が ALARM に変わったときの通知をテストできます。
    - [OK 通知のテスト] — このオプションを選択すると、アラームの状態が OK に変わったときの通知をテストできます。

> **Note**
>
> これらのオプションのいずれかが使用できない場合は、アラームの通知オプションが設定されていないか、アラームが現在 ALARM 状態になっている可能性があります。詳細については、ディストリビューションアラーム制限を参照してください。

選択したテストオプションに応じて、アラームが一時的に ALARM または OK 状態に変化し、アラームの通知方法として設定した内容に応じて、メールまたは SMS テキストメッセージが送信されます。通知バナーは、ALARM 通知のテストを選択した場合にのみ Lightsail コンソールに表示されます。OK 通知のテストを選択した場合、通知バナーは表示されません。アラームは、通常、数秒後に実際の状態に戻ります。

ディストリビューションアラームの作成後の次のステップ

ディストリビューションアラームに対して実行できる追加のタスクがいくつかあります。

- 通知の受信を停止するには、Lightsail から メールと携帯電話を削除します。詳細については、「通知連絡先の削除」を参照してください。また、アラームを無効化または削除して、特定のアラームの通知の受信を停止することもできます。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。

Lightsail ロードバランサーのヘルスメトリクスをモニタリングする

Amazon Lightsail でロードバランサーを作成してインスタンスにアタッチすると、ロードバランサーの管理ページの [メトリクス] タブでメトリクスグラフを表示できます。メトリクスのモニタリングは、リソースの信頼性、可用性、パフォーマンスを維持する上で重要な要素です。リソースから定期的にメトリクスデータをモニタリングして収集し、マルチポイント障害が発生した場合に、より簡単にデバッグできるようにします。メトリクスの詳細については、「メトリクス」を参照してください。

リソースを監視するときは、環境内の通常のリソースパフォーマンスのベースラインを確立する必要があります。ベースラインを確立したら、リソースのパフォーマンスが指定のしきい値を超えたときに通知するように Lightsail コンソールでアラームを設定できます。詳細については、「通知」および「アラーム」を参照してください。

目次

- ロードバランサーのメトリクス
- ロードバランサーメトリクスの表示
- 次のステップ

ロードバランサーのメトリクス

次のロードバランサーメトリクスを使用できます。

- 正常ホスト数 (**HealthyHostCount**) — 正常と見なされるターゲットインスタンスの数。
- 異常ホスト数 (**UnhealthyHostCount**) — 異常と見なされるターゲットインスタンスの数。
- ロードバランサー HTTP 4XX (**HTTPCode_LB_4XX_Count**) — ロードバランサーから発生した HTTP 4XX クライアントエラーコードの数。リクエストの形式が不正な場合、または不完全な場合は、クライアントエラーが生成されます。これらのリクエストは、ターゲットインスタンスによって受信されませんでした。この数には、ターゲットインスタンスによって生成される応答コードは含まれません。
- ロードバランサー HTTP 5XX (**HTTPCode_LB_5XX_Count**) — ロードバランサーから発生した HTTP 5XX サーバーエラーコードの数。これには、ターゲットインスタンスによって生成される応答コードは含まれません。ロードバランサーにアタッチされている正常なインスタンスがない場合、またはリクエストレートがインスタンスやロードバランサーの容量を超える場合 (スピルオーバー)、このメトリクスが報告されます。
- インスタンス HTTP 2XX (**HTTPCode_Instance_2XX_Count**) — ターゲットインスタンスによって生成された HTTP 2XX 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。
- インスタンス HTTP 3XX (**HTTPCode_Instance_3XX_Count**) — ターゲットインスタンスによって生成された HTTP 3XX 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。
- インスタンス HTTP 4XX (**HTTPCode_Instance_4XX_Count**) — ターゲットインスタンスによって生成された HTTP 4XX 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。
- インスタンス HTTP 5XX (**HTTPCode_Instance_5XX_Count**) — ターゲットインスタンスによって生成された HTTP 5XX 応答コードの数。これには、ロードバランサーによって生成される応答コードは含まれません。

- インスタンスからの応答時間 (**InstanceResponseTime**) — ロードバランサーがリクエストを送信してから、ターゲットインスタンスからの応答を受信するまでの経過時間 (秒)。
- クライアント TLS ネゴシエーションエラー数 (**ClientTLSNegotiationErrorCount**) — クライアントにより開始され、ロードバランサーによって生成された TLS エラーのためにロードバランサーとのセッションを確立しなかった、TLS 接続の数。暗号化またはプロトコルの不一致が原因である場合があります。
- リクエストの数 (**RequestCount**) — IPv4 経由で処理されたリクエストの数。この数には、ロードバランサーのターゲットインスタンスによって生成されたレスポンスを含むリクエストのみが含まれます。
- 拒否された接続数 (**RejectedConnectionCount**) — ロードバランサーが接続の最大数に達したため、拒否された接続の数。

ロードバランサーメトリクスの表示

Lightsail コンソールでロードバランサーのメトリクスを表示するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. メトリクスを表示するロードバランサーの名前を選択します。
4. ロードバランサーの管理ページで [Metrics (メトリクス)] タブを選択します。
5. [Metrics graph (メトリクスグラフ)] 見出しの下のドロップダウンメニューで、表示するメトリクスを選択します。

   グラフには、選択したメトリクスのデータポイントが視覚的に表示されます。
6. メトリクスグラフでは、次のアクションを実行できます。

   - グラフの表示を変更して、1 時間、6 時間、1 日、1 週間、2 週間のデータを表示します。
   - データポイント上にカーソルを置くと、そのデータポイントに関する詳細情報が表示されます。
   - 指定したしきい値をメトリクスが超えたときに通知される、選択したメトリクスのアラームを追加します。詳細については、「アラーム」および「ロードバランサーのメトリクスアラームの作成」を参照してください。

次の手順

ロードバランサーのメトリクスに対して実行できる追加のタスクがいくつかあります。

- 指定したしきい値をメトリクスが超えたときに通知される、選択したメトリクスのアラームを追加します。詳細については、「アラーム」および「ロードバランサーのメトリクスアラームの作成」を参照してください。
- アラームが作動すると、通知バナーが Lightsail コンソールに表示されます。E メールと SMS テキストメッセージで通知を受け取るには、リソースをモニタリングする各 AWS リージョン に通知連絡先として E メールアドレスと携帯電話番号を追加する必要があります。詳細については、「通知連絡先を追加する」を参照してください。
- 通知の受信を停止するには、Lightsail から メールと携帯電話を削除します。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。また、アラームを無効化または削除して、特定のアラームの通知の受信を停止することもできます。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。

トピック

- アラームで Lightsail ロードバランサーメトリクスをモニタリングする

アラームで Lightsail ロードバランサーメトリクスをモニタリングする

単一のロードバランサーメトリクスを監視する Amazon Lightsail アラームを作成できます。アラームは、指定したしきい値を基準にしたメトリクスの値に基づいて通知するように設定できます。通知は、Lightsail コンソールに表示されるバナー、メールアドレスに送信される メール、携帯電話番号に送信される SMS テキストメッセージです。アラームの詳細については、「アラーム」を参照してください。

目次

- ロードバランサーのアラーム制限
- ロードバランサーアラームの設定に関するベストプラクティス
- デフォルトのアラーム設定
- Lightsail コンソールを使用してロードバランサーのメトリクスアラームを作成する
- Lightsail コンソールを使用してロードバランサーのメトリクスアラームをテストする
- 次のステップ

ロードバランサーのアラーム制限

アラームには、以下の制限が適用されます。

- メトリクスごとに 2 つのアラームを設定できます。
- アラームは 5 分間隔で評価され、アラームの各データポイントは、集計されたメトリクスデータの 5 分間隔を表します。
- アラームの状態が OK に変わったときに通知するようにアラームを設定できるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合だけです。
- OK アラーム通知をテストできるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合だけです。
- アラームの状態が INSUFFICIENT_DATA に変わったときに通知するようにアラームを設定できるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合と、欠落しているデータポイントに対して [欠落しているデータを評価しない] オプションを選択した場合だけです。
- 通知をテストできるのは、アラームが OK 状態の場合だけです。

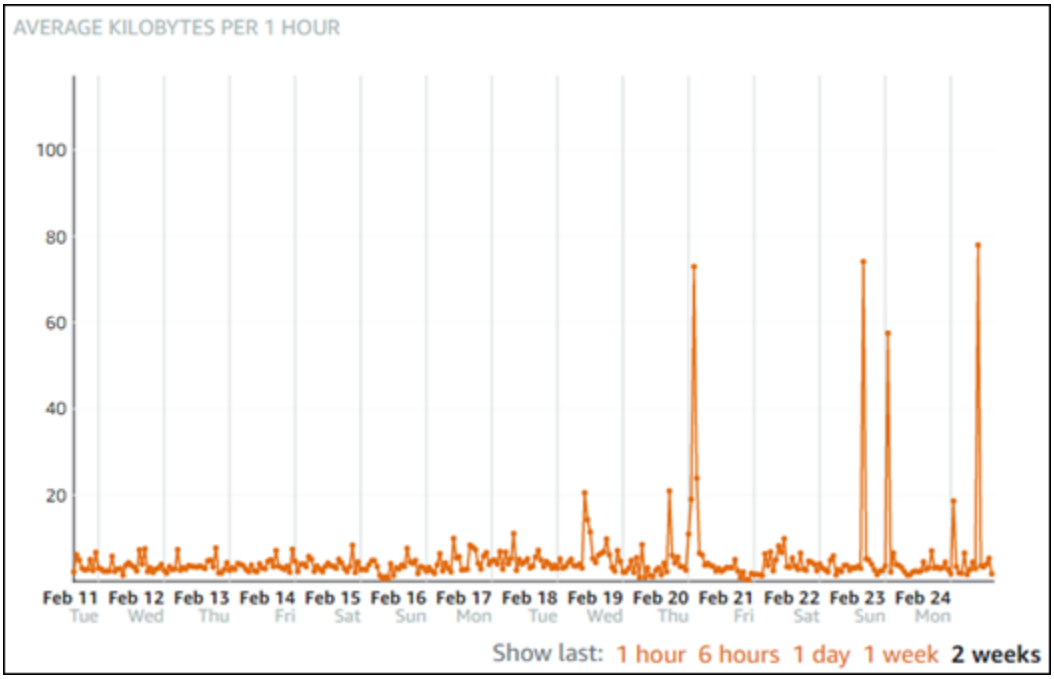
ロードバランサーアラームの設定に関するベストプラクティス

アラームには、以下の制限が適用されます。

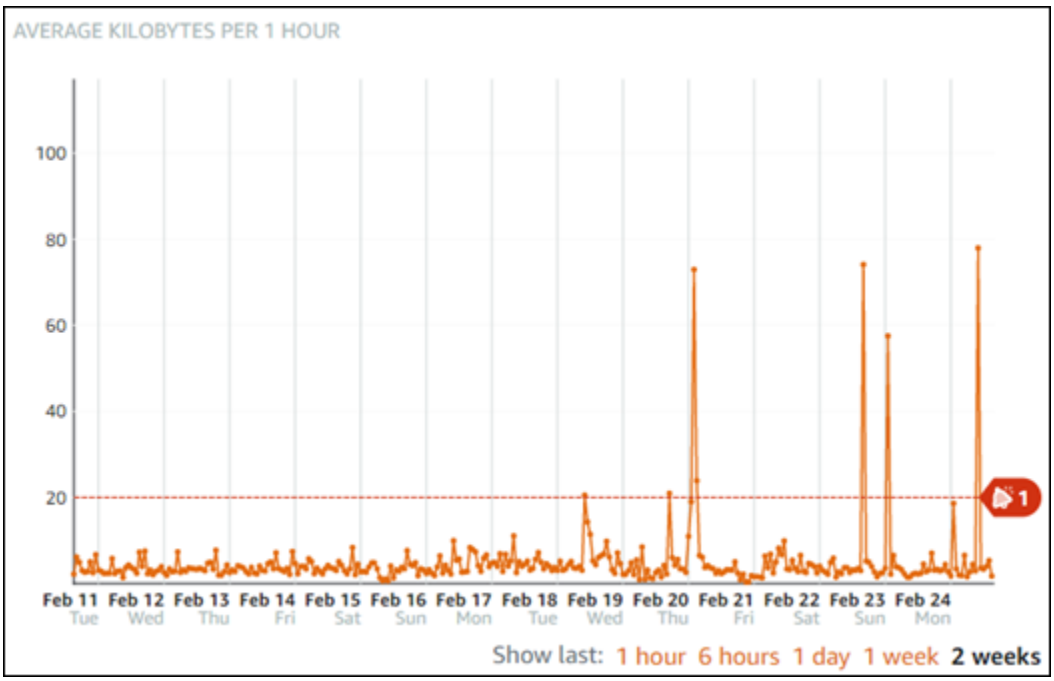
- メトリクスごとに 2 つのアラームを設定できます。
- アラームは 5 分間隔で評価され、アラームの各データポイントは、集計されたメトリクスデータの 5 分間隔を表します。
- アラームの状態が OK に変わったときに通知するようにアラームを設定できるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合だけです。
- OK アラーム通知をテストできるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合だけです。
- アラームの状態が INSUFFICIENT_DATA に変わったときに通知するようにアラームを設定できるのは、メールや SMS テキストメッセージで通知するようにアラームを設定した場合と、欠落しているデータポイントに対して [欠落しているデータを評価しない] オプションを選択した場合だけです。
- 通知をテストできるのは、アラームが OK 状態の場合だけです。

デフォルトのアラーム設定

メトリクスアラームを設定する前に、メトリクスの履歴データを表示する必要があります。過去 2 週間のメトリクスの低レベル、中間レベル、高レベルを識別します。次のインスタンスの発信ネットワークトラフィック (NetworkOut) メトリクスグラフの例では、低レベルは 1 時間あたり 0 ~ 10 KB、中間レベルは 1 時間あたり 10 ~ 20 KB、高レベルは 1 時間あたり 20 ~ 80 KB です。



アラームしきい値を低レベル範囲 (1 時間あたり 5 KB など) 以上に設定すると、アラーム通知が頻繁に送信され、不要なアラーム通知が発生する可能性があります。アラームしきい値を高レベルの範囲 (たとえば、1 時間あたり 20 KB) 以上に設定した場合、アラーム通知の頻度は低くなりますが、調査がさらに意義のあるものになる場合があります。アラームを設定して有効にすると、次の例に示すように、しきい値を表すアラームラインがグラフに表示されます。1 というラベルの付いたアラームラインはアラーム 1 のしきい値を表し、2 というラベルのアラームラインはアラーム 2 のしきい値を表します。



Lightsail コンソールを使用してロードバランサーのメトリクスアラームを作成する

Lightsail コンソールを使用してロードバランサーのメトリクスアラームを作成するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. アラームを作成するロードバランサーの名前を選択します。
4. ロードバランサーの管理ページで [Metrics (メトリクス)] タブを選択します。
5. [Metrics Graphs (メトリクスグラフ)] 見出しの下のドロップダウンメニューで、アラームを作成するメトリクスを選択します。詳細については、「リソースのメトリクス」を参照してください。
6. アラームセクションのページでアラームの追加を選択します。
7. ドロップダウンメニューから比較演算子の値を選択します。例の値は、greater than or equal to、greater than、less than、less than or equal to です。
8. アラームのしきい値を入力します。
9. アラームへのデータポイントを入力します。
10. 評価期間を選択します。期間は、5 分から 24 時間まで 5 分単位で指定できます。
11. 次のいずれかの通知方法を選択します。

- メール — アラームの状態が ALARM に変わると、メールで通知されます。
- SMS テキストメッセージ — アラームの状態が ALARM に変わると、SMS テキストメッセージによって通知されます。SMS メッセージングは、Lightsail リソースを作成できるすべての AWS リージョンでサポートされているわけではなく、また、一部の国/地域には SMS テキストメッセージを送信できません。詳細については、「SMS テキストメッセージングのサポート」を参照してください。

> **Note**
>
> メールまたは SMS による通知を選択したが、リソースの AWSリージョンで通知の連絡先をまだ設定していない場合は、メールアドレスまたは携帯電話番号を追加する必要があります。詳細については、「通知」を参照してください。

12. (オプション) [Send me a notification when the alarm state change to OK (アラームの状態が OK に変わったときに通知を送信する)] を選択して、アラームの状態が OK に変わったときに通知を受け取ります。このオプションは、メールまたは SMS テキストメッセージで通知されるように選択した場合にのみ使用できます。
13. (オプション) [Advanced settings (詳細設定)] を選択し、次のいずれかのオプションを選択します。

    - アラームによる欠落データの処理方法を選択します。次のオプションを使用できます。
      - しきい値の範囲内にないと仮定する(しきい値を超過) — 欠落しているデータポイントは「不良」として扱われ、しきい値を超えています。
      - しきい値内にあると仮定する (しきい値を超過していない) — 欠落しているデータポイントは、「良い」と見なされ、しきい値の範囲内で処理されます。
      - 最後の正常なデータポイントの値を使用する (無視して現在のアラーム状態を維持する) — 現在のアラーム状態が維持されます。
      - 評価しない (欠落しているデータを欠落しているデータとして扱う) — 状態を変更するかどうかを評価する際に、欠落データポイントを考慮に入れません。
    - [Send a notification if there is insufficient data (データが不足している場合に通知を送信)] を選択して、アラームの状態が INSUFFICIENT_DATA に変わったときに通知を受け取ります。このオプションは、メールまたは SMS テキストメッセージで通知されるように選択した場合にのみ使用できます。
14. [Create (作成)] を選択してアラームを追加します。

後でアラームを編集するには、編集するアラームの横にある省略記号アイコン (⋮) を選択し、[アラームの編集] を選択します。

Lightsail コンソールを使用してロードバランサーのメトリクスアラームをテストする

Lightsail コンソールを使用してアラームをテストするには、次の手順を実行します。アラームをテストして、アラームがトリガーされたときに メールや SMS テキストメッセージを受信するなど、設定された通知オプションが機能していることを確認します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[ネットワーク] を選択します。
3. アラームをテストするロードバランサーの名前を選択します。
4. ロードバランサーの管理ページで [Metrics (メトリクス)] タブを選択します。
5. [Metrics Graphs (メトリクスグラフ)] 見出しの下のドロップダウンメニューで、アラームをテストするメトリクスを選択します。
6. ページの [アラーム] セクションまでスクロールし、テストするアラームの横にある省略記号アイコン (⋮) を選択します。
7. 以下のオプションのいずれかを選択してください。

   - [アラーム通知のテスト] - このオプションを選択すると、アラームの状態が ALARM に変わったときの通知をテストできます。
   - [OK 通知のテスト] — このオプションを選択すると、アラームの状態が OK に変わったときの通知をテストできます。

> **Note**
>
> これらのオプションのいずれかが使用できない場合は、アラームの通知オプションが設定されていないか、アラームが現在 ALARM 状態になっている可能性があります。詳細については、「ロードバランサーのアラーム制限」を参照してください。

選択したテストオプションに応じて、アラームが一時的に ALARM または OK 状態に変化し、アラームの通知方法として設定した内容に応じて、メールまたは SMS テキストメッセージが送信されます。通知バナーは、ALARM 通知のテストを選択した場合にのみ Lightsail コンソールに表

示されます。OK 通知のテストを選択した場合、通知バナーは表示されません。アラームは、通常、数秒後に実際の状態に戻ります。

ロードバランサーアラームの作成後の次のステップ

ロードバランサーのアラームに対して実行できる追加のタスクがいくつかあります。

- 通知の受信を停止するには、Lightsail から メールと携帯電話を削除します。詳細については、「通知連絡先の削除」を参照してください。また、アラームを無効化または削除して、特定のアラームの通知の受信を停止することもできます。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。

Lightsail モニタリングの通知連絡先を設定する

インスタンス、データベース、ロードバランサーないしコンテンツ配信ネットワーク (CDN) ディストリビューションのメトリクスが指定した値を超えた場合に通知を受けられるよう、Amazon Lightsail を設定することが可能す。通知は、Lightsail コンソールに表示されるバナー、指定したアドレスに送信される メール、指定した携帯電話番号に送信される SMS テキストメッセージの形式にすることができます。E メールと SMS テキストメッセージで通知を受け取るには、リソースをモニタリングする各 AWS リージョン に通知連絡先として E メールアドレスと携帯電話番号を追加する必要があります。通知の詳細については、「通知」を参照してください。

> **Important**
>
> SMS テキストメッセージ機能は一時的に無効になっており、Lightsailリソースを作成できる AWS リージョン では現在サポートされていません。詳細については、「SMS テキストメッセージングのサポート」を参照してください。

目次

- リージョンの通知の連絡先の制限
- SMS テキストメッセージングのサポート
- メールによる連絡先の確認
- Lightsail コンソールを使用した通知連絡先の追加
- を使用した通知連絡先の追加 AWS CLI

- 通知連絡先を追加した後の次の手順

リージョンの通知の連絡先の制限

それぞれに 1 つの E メールアドレスと 1 つの携帯電話番号のみを追加できます AWS リージョン。すでにメールアドレスや携帯電話番号追加されているリージョンにメールアドレスまたは携帯電話番号を追加すると、既存の通知連絡先を新しい連絡先に置き換えるかどうか尋ねられます。

で複数の E メール受信者が必要な場合は AWS リージョン、複数の受信者に転送するディストリビューションリストを設定し、ディストリビューションリストの E メールアドレスを通知連絡先として追加できます。

SMS テキストメッセージングのサポート

> **Important**
>
> SMS テキストメッセージ機能は一時的に無効になっており、Lightsailリソースを作成できる AWS リージョン では現在サポートされていません。または、メールのメッセージを設定したり、Lightsail コンソールに表示される通知バナーを利用したりすることも可能です。SMS テキストメッセージサポートに関する次の情報は、この機能を無効にする前に SMS テキストメッセージを設定したお客様向けに公開されています。

SMS テキストメッセージは、Lightsailリソースを作成できるすべての AWS リージョンでサポートされているわけではありません。また、SMSテキストメッセージは、世界の一部の国や地域に送信することはできません。SMS メッセージングがサポートされていない AWS リージョンでは、E メール通知連絡先のみを設定できます。

SMS メッセージングは、次の AWS リージョンでサポートされています。これらは Amazon Simple Notification Service (Amazon SNS)が SMS テキストメッセージングをサポートしているリージョンで、このリージョンは、Lightsail が通知を送信するために使用されます。

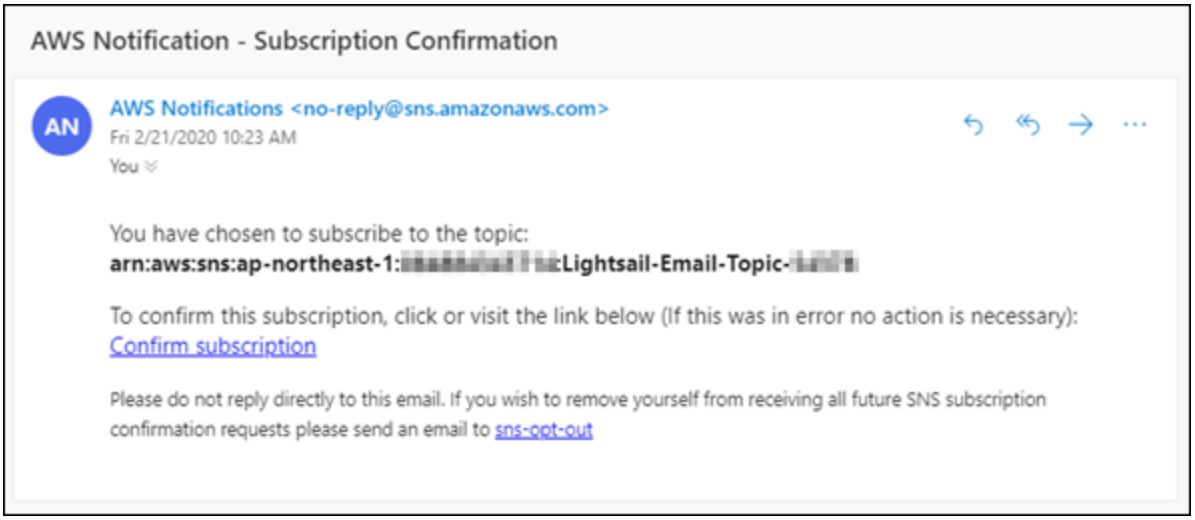
- 米国東部 (バージニア北部) (us-east-1)
- 米国西部 (オレゴン) (us-west-2)
- アジアパシフィック (シンガポール) (ap-southeast-1)
- アジアパシフィック (シドニー) (ap-southeast-2)
- アジアパシフィック (東京) (ap-northeast-1)

- 欧州 (アイルランド) (eu-west-1)

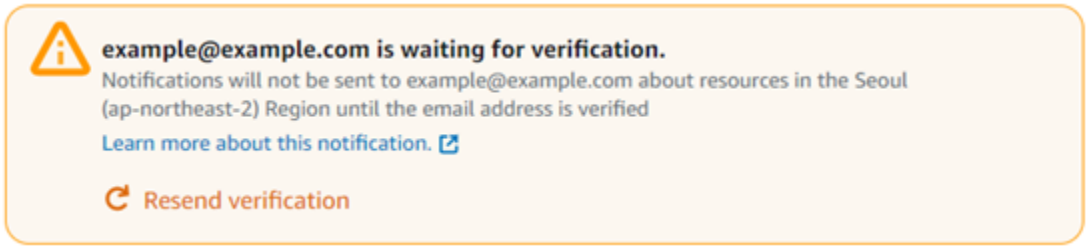
SMS テキストメッセージを送信できる世界の国とリージョン、および SMS テキストメッセージがサポートされている最新の のリストについては、「Amazon SNS AWS リージョンデベロッパーガイド」の「サポートされているリージョンと国」を参照してください。 Amazon SNS

メールによる連絡先の確認

Lightsail でメールアドレスを通知連絡先として追加すると、そのアドレスに確認要求が送信されます。確認要求の メールには、受信者が Lightsail 通知を受信することを確認するためにクリックする必要があるリンクが含まれています。通知は、確認が完了するまでメールアドレスに送信されません。検証は、AWS 通知 <no-reply@sns.amazonaws.com> から行われ、件名は AWS 通知 - サブスクリプションの確認です。SMS メッセージングには検証は不要です。



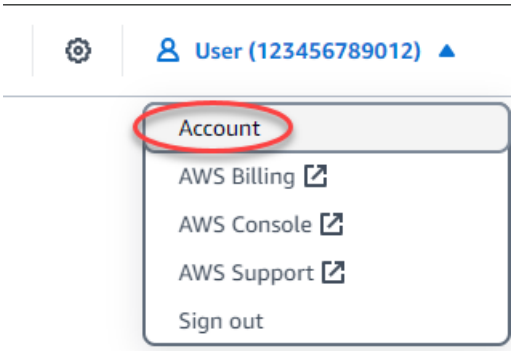
確認要求が受信トレイフォルダにない場合は、メールボックスのスパムフォルダと迷惑メールフォルダを確認してください。検証リクエストが失われた、または削除された場合は、Lightsail コンソールやアカウントページに表示される通知バナーで検証の再送を選択してください。



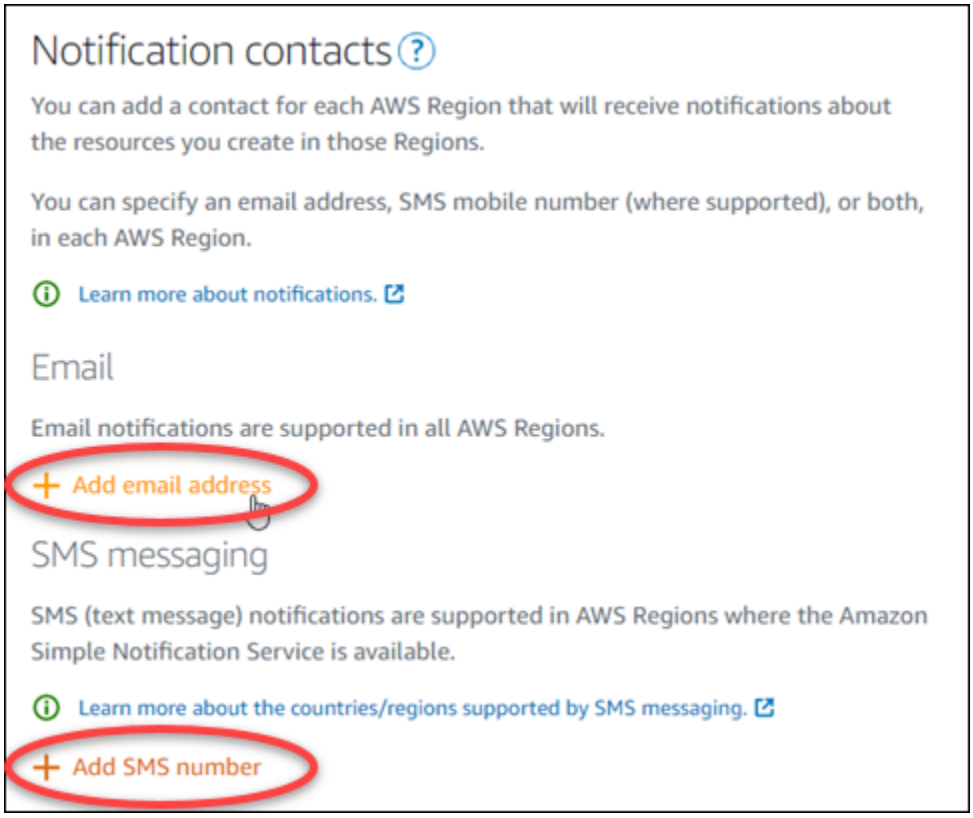
Lightsail コンソールを使用した通知連絡先の追加

Lightsail コンソールを使用して通知連絡先を追加するには、次の手順を完了します。

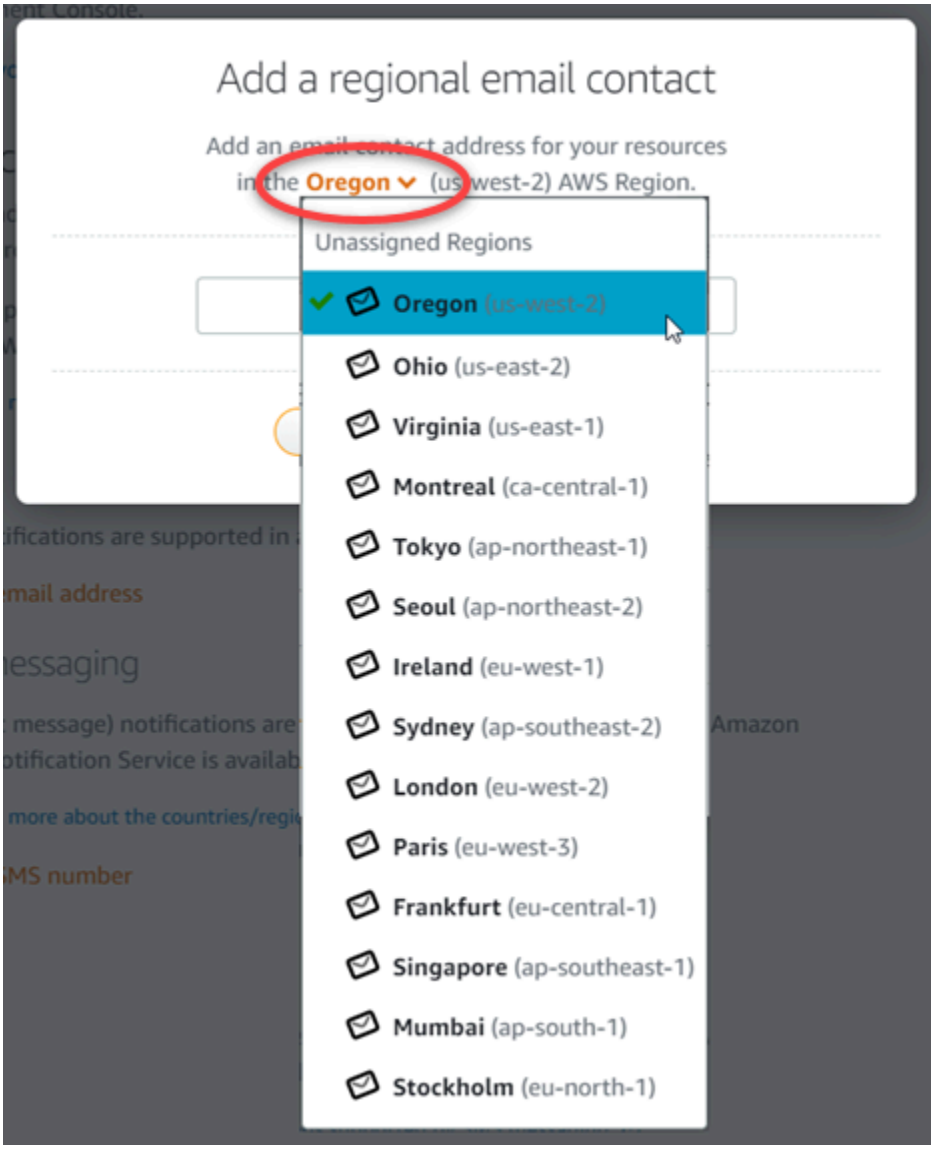
1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
3. ドロップダウンメニューで [Account (アカウント)] を選択します。



4. [プロフィールと連絡先] タブの [通知連絡先] セクションで、[メールアドレスの追加] または [SMS 番号の追加] を選択します。



5. 次のいずれかのステップを完了します。
   - E メールアドレスを追加する場合は、通知連絡先を追加する AWS リージョン を選択します。テキストボックスにメールアドレスを入力します。

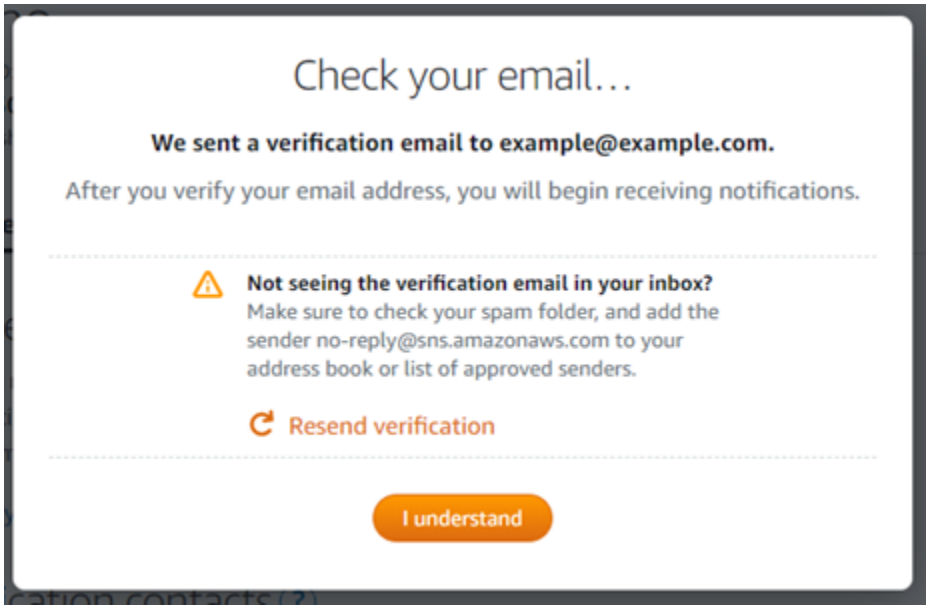


- SMS 番号を追加する場合は、通知連絡先を追加する AWS リージョン を選択します。携帯電話番号の国を選択し、テキストボックスに入力します。国コードは既に入力されています。

> **Important**
>
> SMS テキストメッセージ機能は一時的に無効になっており、Lightsailリソースを作成できる AWS リージョン では現在サポートされていません。詳細については、「SMS テキストメッセージングのサポート」を参照してください。

6. [Add Contact (連絡先を追加)] を選択します。

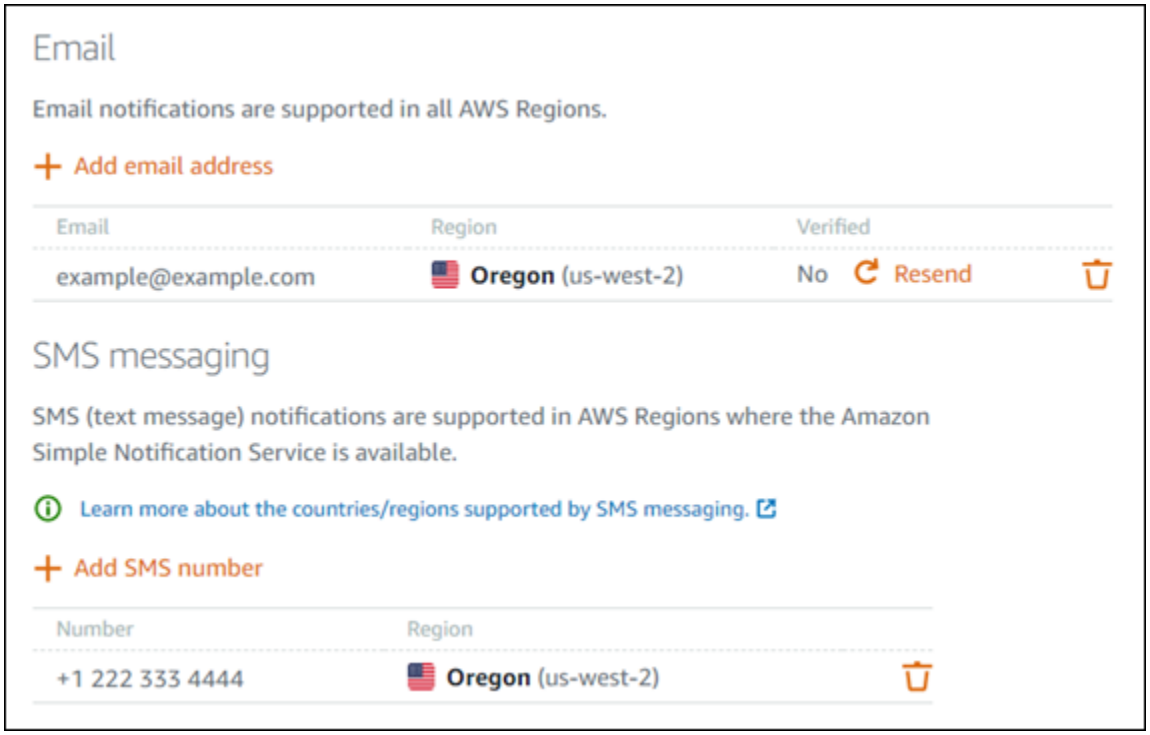
   メールアドレスを通知連絡先として追加すると、そのアドレスに確認要求が送信されます。確認要求の メールには、受信者が Lightsail 通知を受信することを確認するためにクリックする必要があるリンクが含まれています。SMS メッセージングには検証は不要です。



7. [I understand (理解する)] を選択します。

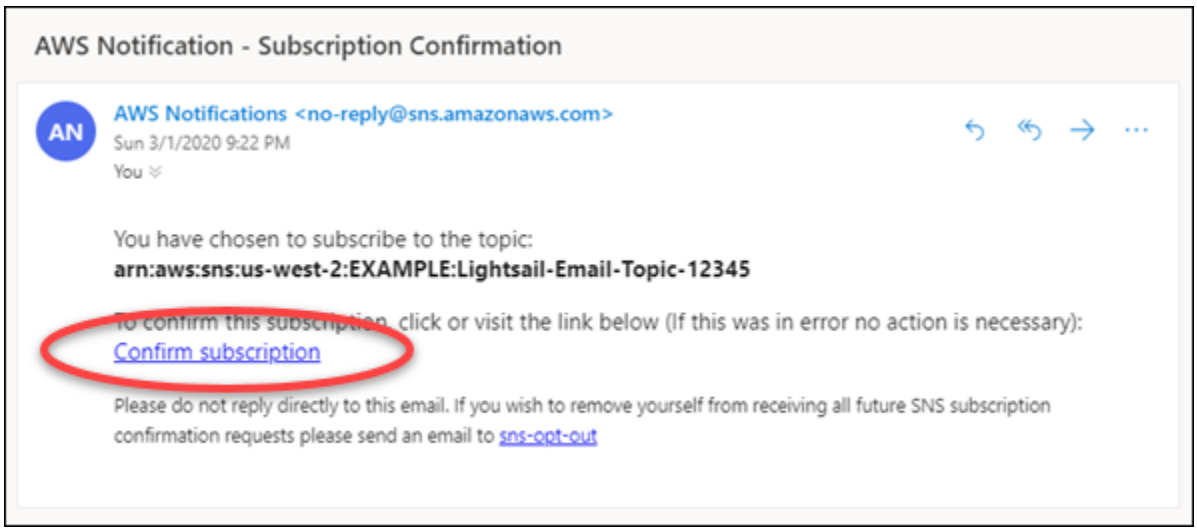
メールアドレスまたは携帯電話番号が [通知連絡先] セクションに追加されます。次の手順で確認プロセスを完了するまで、メールアドレスは確認されません。確認が完了するまで、通知はメールアドレスに送信されません。検証リクエストが紛失したか、削除された場合は、リージョンのメールアドレスの横にある [Resend (再送信)] を選択して、別の検証リクエストを送信します。

> **Note**
>
> SMS メッセージングには検証は不要です。したがって、SMS 通知の連絡先を追加した後、この手順の手順 8 ~ 10 を完了する必要はありません。



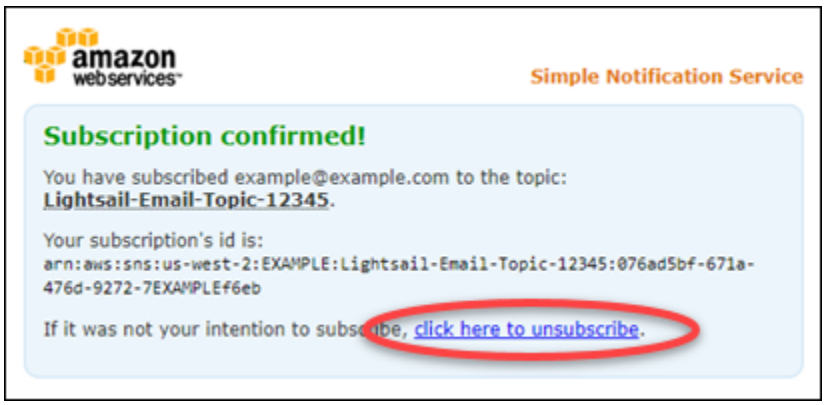
8. Lightsail で通知連絡先として追加したメールアドレスの受信トレイを開きます。
9. no-reply@sns.amazonaws.com からの AWS 通知 - サブスクリプションの確認 メールを開きます。

> **Note**
>
> 確認要求が受信トレイフォルダにない場合は、メールボックスのスパムフォルダと迷惑メールフォルダを確認してください。



10. メールで [Confirm subscription (サブスクリプションを確認)] を選択して、Lightsail 通知を受信することを確認します。

    ブラウザウィンドウが開き、サブスクリプションを確認する次のページが表示されます。登録を解除するには、ページの [click here to unsubscribe (ここをクリックしてページから登録を解除します)] を選択します。または、ページを閉じた場合は、通知連絡先を削除する手順を実行します。



を使用した通知連絡先の追加 AWS CLI

Lightsail に AWS Command Line Interface (AWS CLI) を使用するための通知連絡先を追加するには、次の手順を完了します。

1. ターミナルまたはコマンドプロントウィンドウを開きます。

まだインストールしていない場合は、 をインストール AWS CLI し、 と連携するように設定してくださいLightsail。

2. 次のコマンドを入力して、通知連絡先を追加します。

```
aws lightsail create-contact-method --region Region --notificationProtocol Protocol
 --contact-endpoint Destination
```

コマンドを、以下のように置き換えます。

- 通知連絡先を追加する必要がある AWS リージョン がある*#####*。
- 連絡先の通知プロトコルを使用する*#####*。メール、または SMS にする必要があります。
- メールアドレスまたは携帯電話番号の*##* 。

> **Note**
>
> 携帯電話番号を指定する場合は、E.164 形式を使用します。E.164 は、国際的な音声通信に使用される電話番号の構造の規格です。この形式に従う電話番号には最大 15 桁を設定でき、プラス記号 (+) および国コードのプレフィックスがついています。たとえば、E.164 形式の米国の電話番号は +1XXX5550100 として表示されます。詳細については、Wikipedia の E.164 を参照してください。

例:

```
aws lightsail create-contact-method --region us-west-2 --notificationProtocol Email
 --contact-endpoint example@example.com
```

```
aws lightsail create-contact-method --region us-east-1 --notificationProtocol SMS
 --contact-endpoint +14445556666
```

Enter キーを押すと、オペレーションの応答にリクエストの詳細が表示されます。

通知の連絡先として指定したメールアドレスに確認リクエストが送信されます。これにより、受信者が Lightsail 通知のサブスクリプションを希望していることが確認されます。メールアドレスは、以下の手順で確認処理が完了するまで確認されません。メールアドレスが確認されるま

で、通知はメールアドレスに送信されません。元の通知が間違っている場合は、リージョンのメールアドレスの横にある [Resend (再送信)] を選択して、別の確認リクエストを送信します。

> **Note**
>
> SMS メッセージングには検証は不要です。したがって、SMS 通知の連絡先を追加するときに、この手順の手順 8 ~ 10 を完了する必要はありません。

3. 通知連絡先として追加したメールアドレスの受信トレイを開きます。
4. no-reply@sns.amazonaws.com からの AWS 通知 - サブスクリプションの確認 メールを開きます。
5. メールで [Confirm subscription (サブスクリプションの確認)] を選択して、Lightsail からの メール通知を受信することを確認します。

   ブラウザウィンドウが開き、サブスクリプションを確認する次のページが表示されます。登録を解除するには、ページの [click here to unsubscribe (ここをクリックしてページから登録を解除します)] を選択します。または、ページを閉じた場合は、通知連絡先を削除する手順を実行します。

通知連絡先を追加した後の次の手順

通知連絡先に対して実行できる追加のタスクがいくつかあります。

- 通知連絡先を追加 AWS リージョン した にアラームを追加します。アラームの開始時に、メールおよび SMS テキストメッセージで通知されるように選択できます。詳細については、「アラーム」を参照してください。
- 通知が予定されているときに通知を受け取らない場合は、通知の連絡先が正しく設定されていることを確認するためにいくつかの点を確認してください。詳細については、「通知のトラブルシューティング」を参照してください。
- 通知の受信を停止するには、Lightsail から メールと携帯電話を削除します。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。また、アラームを無効化または削除して、特定のアラームの通知の受信を停止することもできます。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。

Lightsail の通知連絡先を削除する

メールおよび携帯電話番号の通知連絡先を Amazon Lightsail から削除して、Lightsail リソースのメールおよび SMS テキストメッセージ通知の受信を停止します。通知の詳細については、「通知」を参照してください。

また、アラームを無効にするか、削除して、特定のアラームの通知の受信を停止することもできます。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。

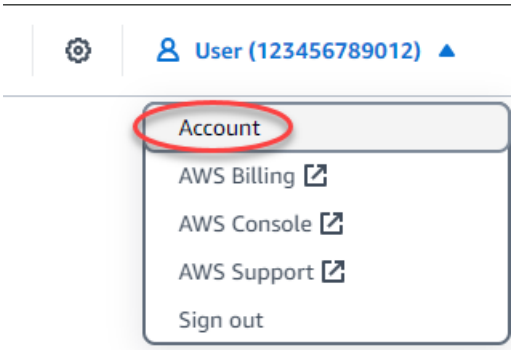
目次

- Lightsail コンソールを使用した通知連絡先の削除
- を使用した通知連絡先の削除 AWS CLI
- 通知連絡先を削除した後の次の手順

Lightsail コンソールを使用した通知連絡先の削除

Lightsail コンソールを使用して通知連絡先を削除するには、次の手順を実行します。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの上部にあるナビゲーションメニューでユーザーまたはロールを選択します。
3. ドロップダウンメニューで [Account (アカウント)] を選択します。



4. 削除するメールアドレスまたは携帯電話番号の横にある削除アイコンを [Profile & contacts] (プロファイルと連絡先) タブの [Notification contacts] (通知連絡先) セクションで選択します。
5. [Yes (はい)] を選択して、通知連絡先を削除することを確認します。

を使用した通知連絡先の削除 AWS CLI

AWS Command Line Interface (AWS CLI) を使用して Lightsail の通知連絡先を削除するには、次の手順を実行します。

1. ターミナルまたはコマンドプロントウィンドウを開きます。

   まだインストールしていない場合は、 をインストール AWS CLI し、 で動作するように設定しますLightsail。

2. 通知連絡先を削除するには、次のコマンドを入力します。

   ```
   aws lightsail delete-contact-method --region Region --notificationProtocol Protocol
   ```

   コマンドを、以下のように置き換えます。

   - 通知連絡先を削除する AWS リージョン がある*#####*。
   - メールや SMS など、削除する連絡先の通知プロトコルを使用する*#####*。

   例:

   ```
   aws lightsail delete-contact-method --region us-west-2 --notificationProtocol SMS
   ```

   Enter キーを押すと、オペレーションの応答にリクエストの詳細が表示されます。

通知連絡先を削除した後の次の手順

通知連絡先の削除後に実行できる追加のタスクがいくつかあります。

- 通知連絡先を削除すると、メールおよび SMS テキストメッセージングの通知は停止しますが、通知バナーが Lightsail コンソールに表示されるのを停止することはできません。通知バナーを停止し、メールおよび SMS テキストメッセージング通知も停止するには、バナーの原因となっているアラームを無効にするか削除します。詳細については、「メトリクスアラームを削除または無効化する」を参照してください。
- メールおよび SMS テキストメッセージング通知の受信を再開するには、通知連絡先 Lightsail としてメールアドレスと携帯電話番号を追加します。詳細については、「通知連絡先を追加する」を参照してください。

Lightsail アラーム通知と検証保留中の連絡先を確認する

[アラーム通知] ページの Lightsail コンソールで、すべての Amazon Lightsail リソースのアクティブなアラームおよび通知を確認できます。このページでは、`In alarm` 状態のアラーム (有効になっていて、定義されたしきい値を超えているアラーム) がまとめられています。検証保留中の E メール連絡先を確認することもできます。アラームの詳細については、「Lightsail のメトリクスアラーム」を参照してください。　アラームの通知の詳細については、「Lightsail リソースのメトリクス通知を設定する」を参照してください。

トピック

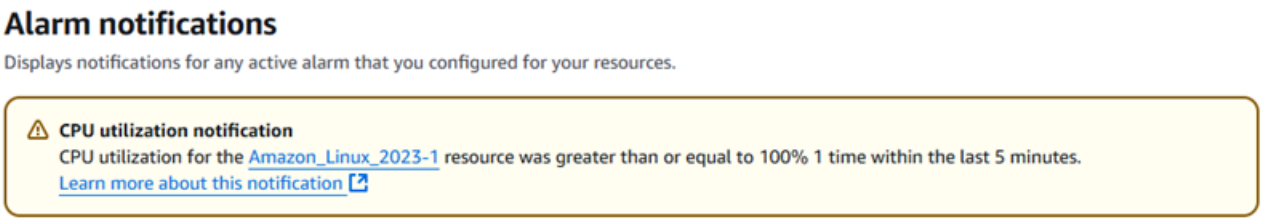
- アクティブなアラームのアラーム通知を確認する
- 検証保留中の E メール連絡先を確認する

アクティブなアラームのアラーム通知を確認する

Lightsail コンソールで、すべてのリソースの Lightsail のアラーム通知を確認できます。各エントリには、アラームがアクティブである理由および関連するリソースに関する追加の詳細が含まれます。アラームを追加する方法の詳細については、「アラームの設定」を参照してください。

アクティブなアラームのアラーム通知を確認する方法

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[アラーム通知] を選択します。
3. [アラーム通知] では、アクティブなアラームを確認できます。



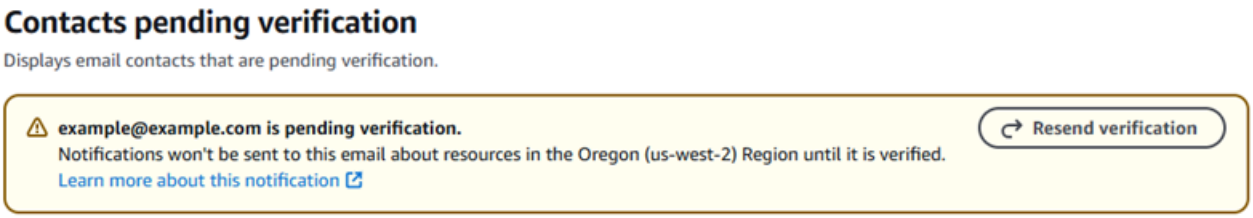
検証保留中の E メール連絡先を確認する

検証保留中の E メール連絡先は、Lightsail コンソールで確認できます。各エントリには、E メールアドレス、 AWS リージョン 通知の対象、検証を再送信する機能が含まれます。E メール連絡先の

追加方法の詳細については、「Lightsail モニタリングの通知連絡先を設定する」を参照してください。

検証保留中の E メール連絡先を確認する方法

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、[アラーム通知] を選択します。
3. [検証保留中の連絡先] では、検証保留中の E メール連絡先を確認できます。

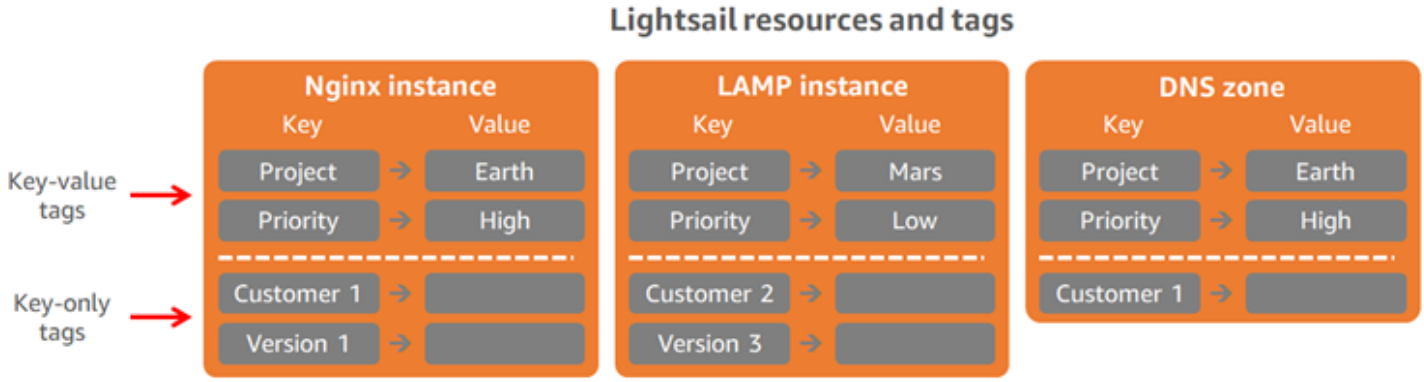


タグを使用して Lightsail リソースを整理してフィルタ処理する

Amazon Lightsail は、ラベルをタグとしてリソースに割り当てることができます。各タグは、キーおよび値 (省略可能) で構成されるラベルです。タグを使うと、リソースの管理、検出、およびフィルタ処理が容易になります。

Amazon Lightsail は、ラベルをタグとしてリソースに割り当てることができます。各タグは、キーとオプションの値で構成されるラベルです。使用すると効率的にリソースの管理、検索、フィルタリングができます。タグには、固有なタイプはありませんが、Lightsail リソースを用途、所有者、環境などの基準で分類できます。これは、同じ種類のリソースが多い場合に役立ちます。リソースに割り当てたタグに基づいて、特定のリソースをすばやく識別できます。たとえば、各リソースのプロジェクトや優先度の追跡に役立つ一連のタグを定義できます。

値のないキーは、Lightsail でキーのみのタグと呼ばれ ます。値のあるキーは、キーと値のタグと呼ばれ ます。次の図はタグの機能を示しています。この例では、各リソースに複数の「キーと値のタグ」があり、1 つ以上の「キーのみのタグ」があります。キーと値のタグはプロジェクトと優先度を識別し、キーのみのタグは顧客とアプリケーションバージョンを識別します。



タグを使用して請求を整理し、アクセスをコントロールする

タグを使用して、請求の整理、Lightsail のリソースとリクエストへのアクセスコントロール、およびタグキーへのアクセスコントロールを行うこともできます。詳細については、以下のいずれかのガイドを参照してください。

- タグを使用してリソースのコストを整理する
- タグを使用してリソースアクセスを制御する

タグ付けをサポートする Lightsail のリソース

ほとんどの Lightsail リソースには、作成時または作成後にタグを付けることができます。リソースの作成時にタグを適用できない場合、Lightsail はリソースの作成プロセスをロールバックします。これにより、リソースはタグ付きで作成されるか、まったく作成されないことになり、タグ付けを要するリソースにタグが付いていない状態はなくなります。

以下の Lightsail リソースには、Lightsail コンソールでタグを付けることができます。

- インスタンス
- コンテナサービス
- コンテンツ配信ネットワーク (CDN) の配信
- バケット
- データベース
- Disks
- DNS ゾーン
- ロードバランサー

> **Important**
>
> Lightsail コンソールを使用して作成されたスナップショットは、ソースリソースからタグを自動的に継承します。スナップショットから作成した Lightsail リソースには、スナップショットの作成時にソースリソースと同じタグが適用されます。

次のリソースは Lightsail API、AWS Command Line Interface (AWS CLI)、SDK を使用してタグ付けできます。

- データベーススナップショット
- データベース
- ディスクスナップショット
- Disks
- ドメイン (DNS ゾーン)
- インスタンススナップショット

- インスタンス
- キーペア
- ロードバランサーの TLS 証明書 (Lightsail で作成された TLS 証明書)
- ロードバランサー

> **Important**
>
> Lightsail API、、または SDKs を使用して作成されたスナップショットは AWS CLI、ソースリソースからタグを自動的に継承しません。代わりに、`tags` パラメータを使用してソースリソースのタグを手動で指定する必要があります。

タグの制限

タグには以下のベーシックな制限があります。

- リソースあたりのタグの最大数 – 50。
- リソースごとに各タグキーを一意にする必要があります。各タグキーに保持できる値は 1 つのみです。
- キーの最大長 – 128 Unicode 文字 (UTF-8)
- 値の最大長 - 256 Unicode 文字 (UTF-8)。
- 複数の のサービス間およびリソース間でタグ付けスキーマを使用する場合、他のサービスにも許容される文字数に制限がある可能性があることに注意してください。通常使用できる文字は、文字、数字、スペース、および特殊文字 +、-、= . _ : / @ です。
- タグキーと値は大文字と小文字が区別されます。
- キーや値には `aws:` プレフィックスは使用しないでください。このプレフィックスは AWS 専用として予約されています。

タグで Lightsail リソースを分類する

Amazon Lightsail のタグを使用し、リソースを目的、所有者、環境などの基準別に分類します。タグは、リソースの作成時または作成後に追加できます。作成後のリソースにタグを追加するには、以下の手順を実行します。

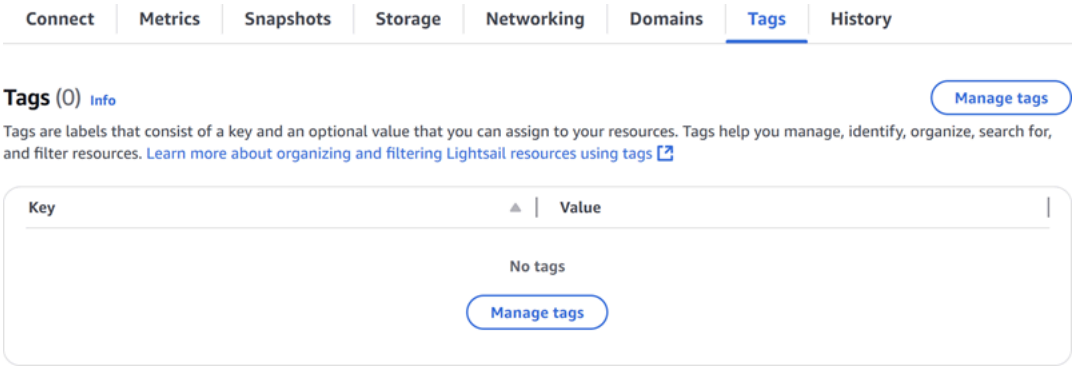
> **Note**
>
> タグ、タグを追加できるリソース、および制限の詳細については、「タグ」を参照してください。

リソースにタグを追加するには

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、タグを付けたいリソースタイプを選択します。たとえば、DNS ゾーンにタグを追加するには、[ネットワーキング] タブを選択します。インスタンスにタグを追加するには、[インスタンス] タブを選択します。

   > **Note**
   >
   > Lightsail コンソールを使用して、インスタンス、コンテナサービス、 CDN ディストリビューションバケット、データベース、ディスク、DNS ゾーン、およびロードバランサーをタグ付けすることができます。ただし、Lightsail API オペレーション、AWS Command Line Interface (AWS CLI)、または SDK を使用することで、より多くの Lightsail リソースにタグ付けすることができます。タグ付けをサポートする Lightsail リソースの詳細なリストについては、「タグ」を参照してください。

3. タグを追加するリソースを選択します。
4. 選択したリソースの管理ページで、[タグ] タブを選択します。



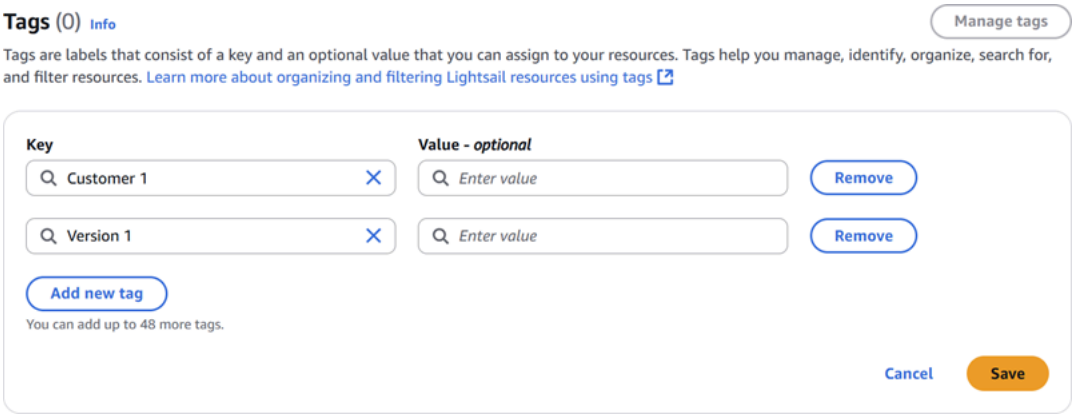
5. [Manage tags (タグの管理)] を選択します。
6. 作成するタグの種類に応じて、次のいずれかのオプションを使用します。また、このタブから既に追加されているタグを編集することもできます。

   - キーのみのタグを作成する

a. [新しいタグを追加] をクリックします。

b. 作成する各タグの [キー] の値を入力します。

c. タグの入力が完了したら [保存] を選択し、追加しない場合は [キャンセル] を選択します。

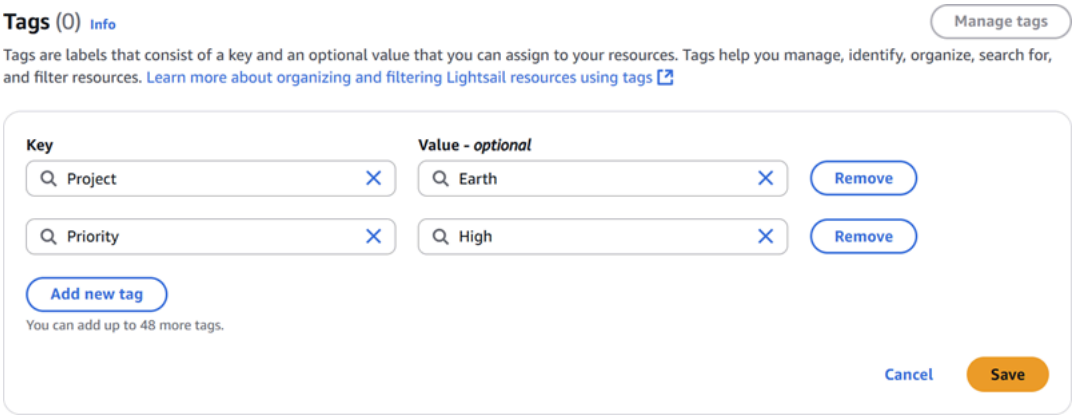


- キーと値のタグを作成する

a. [新しいタグを追加] をクリックします。

b. 作成する各タグの [キー] と [値] に値を入力します。

c. タグの入力が完了したら [保存] を選択し、追加しない場合は [キャンセル] を選択します。



次の手順

リソースにタグを追加した後で実行できるタスクの詳細については、以下のガイドを参照してください。

- タグを使用して、リソースを整理する

- タグを使用してリソースのコストを整理する
- タグを使用してリソースへのアクセスを制御する
- タグの削除

Lightsail リソースからタグを削除します

Amazon Lightsail リソースからタグを削除できます。1 つのリソースからタグを削除しても、他のすべてのリソースから同じタグが削除されるわけではありません。すべてのリソースからタグを完全に削除するには、そのタグを各リソースから削除する必要があります。このガイドでは、リソースからタグを削除する手順を示します。

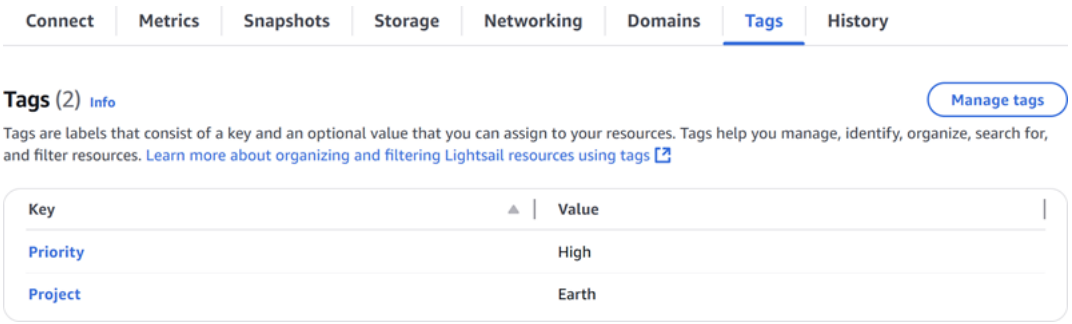
> **Note**
>
> タグ、タグを追加できるリソース、およびタグの制限の詳細については、「タグ」を参照してください。

リソースからタグを削除するには

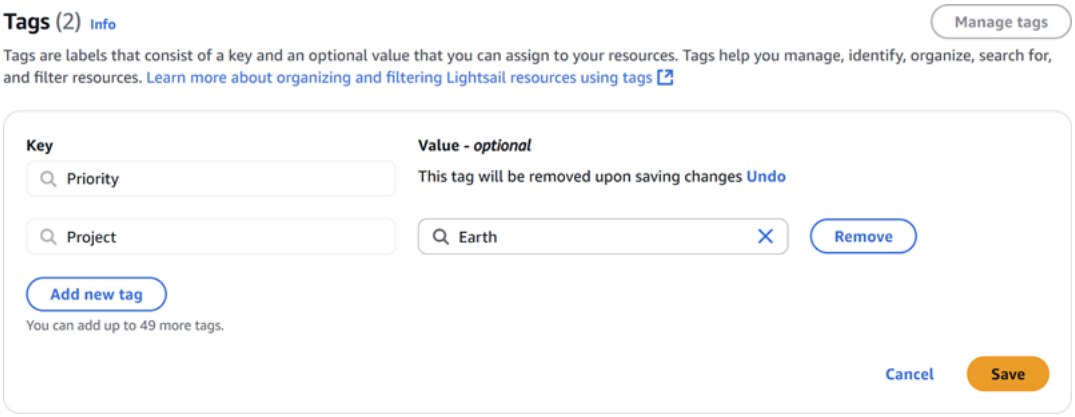
1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、タグを削除しようとするリソースのタイプを選択します。たとえば、DNS ゾーンからタグを削除するには、[ネットワーク] タブを選択します。インスタンスからタグを削除するには、[インスタンス] タブを選択します。

   > **Note**
   >
   > Lightsail コンソールを使用して、インスタンス、コンテナサービス、 CDN ディストリビューションバケット、データベース、ディスク、DNS ゾーン、およびロードバランサーをタグ付けすることができます。ただし、Lightsail API オペレーション、または AWS コマンドラインインターフェイス (AWS CLI) か SDK を使用すると、より多くの Lightsail リソースにタグ付けできます。タグ付けをサポートする Lightsail リソースの詳細なリストについては、「タグ」を参照してください。

3. タグを削除するリソースを選択します。
4. 選択したリソースの管理ページで、[タグ] タブを選択します。



5. [Manage tags (タグの管理)] を選択します。
6. リソースから削除するタグには、[削除] を選択します。
7. [保存] を選択して選択したタグをリソースから削除するか、[キャンセル] を選択して保持します。



Lightsail リソースレベルのアクセス許可とタグベースの承認でリソースへのアクセスを制御する

Lightsail は、リソースレベルのアクセス許可および一部の API アクションのタグに基づく承認をサポートします。詳細については、サービス認可リファレンスの「Actions, resources, and condition keys for Amazon Lightsail」を参照してください。

タグで Lightsail リソースへのアクセスをコントロールする

Amazon Lightsail のタグを使用して、リソース、リクエスト、およびタグキーへのアクセスをコントロールできます。このガイドでは、Lightsail リソースの作成や削除に必要な key-value タグを指定する AWS Identity and Access Management (IAM) ポリシーを作成し、また、これらのリクエストを行うユーザーやグループにポリシーをアタッチする方法について説明します。

> **Note**
>
> Lightsail のタグの詳細については、「タグ」を参照してください。

ステップ 1: IAM ポリシーを作成する

まず、IAM コンソールで以下の IAM ポリシーを作成します。IAM ポリシーの作成の詳細については、IAM ドキュメントの「IAM ポリシーの作成」を参照してください。

次に続くポリシーは、作成リクエストに `allow` のキータグと、 `true` の値が定義されていない限り、ユーザーによる新しい Lightsail リソースの作成も禁止します。このポリシーは、 `allow/true` のキーバリューのタグが定義されていない限り、ユーザーによるリソースの削除も禁止します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "lightsail:Create*",
                "lightsail:TagResource",
                "lightsail:UntagResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/allow": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "lightsail:Delete*",
                "lightsail:TagResource",
                "lightsail:UntagResource"
            ],
```

```
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/allow": "true"
                }
            }
        }
    ]
}
```

次に続くポリシーでは、キーバリューのタグが allow/false ではないリソースのタグの変更をユーザーに禁止します。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "lightsail:TagResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:ResourceTag/allow": "false"
                }
            }
        }
    ]
}
```

ステップ 2: ユーザーまたはグループにポリシーをアタッチする

IAM ポリシーを作成したら、キーと値のペアを使用して、Lightsail リソースを作成する必要があるユーザーやグループにアタッチします。ユーザーまたはグループに IAM ポリシーをアタッチする方法の詳細については、IAM ドキュメントの「IAM ポリシーの追加と削除」を参照してください。

タグを使用して Lightsail リソースコストを整理する

でタグを使用してAmazon Lightsail、独自のコスト構造を反映するように AWS 請求を整理できます。そのためには、Lightsail リソースにキーと値のタグを追加します。次に、これらのタグを AWS Billing and Cost Management コンソールでアクティブ化します。最後に、サインアップして、コスト配分レポートに含まれるタグキー値を含む AWS アカウント請求書を取得します。このセットアップ手順について以下に説明します。

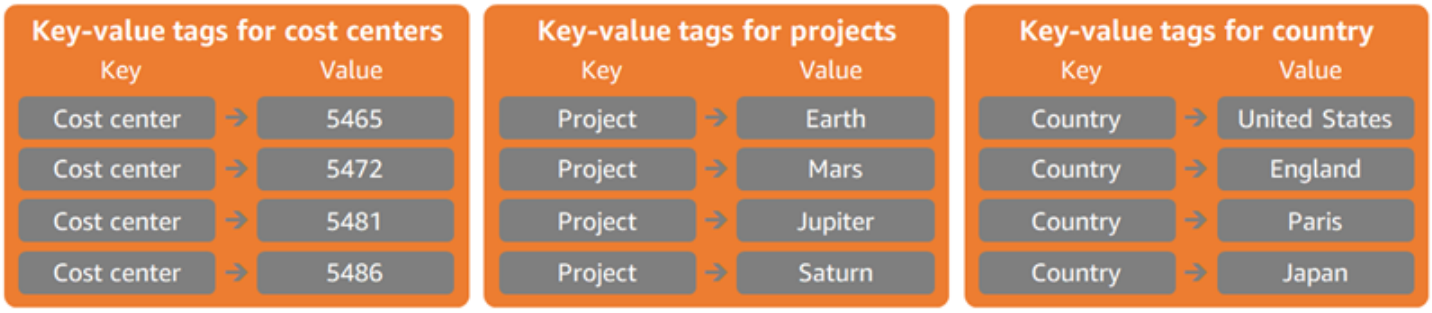
> **Note**
>
> Lightsail のタグ、タグ付けできるリソース、およびタグの制限の詳細については、「タグ」を参照してください。

> **Important**
>
> 現時点では、コスト配分タグを追加しても、Lightsail データベーススナップショットをコスト配分レポートで追跡することはできません。

ステップ 1: キーと値のタグを リソースに追加する

請求コンソールで整理する対象の Lightsail リソースにキーと値のタグを追加します。キーバリュー型のタグの詳細については、「リソースにタグを追加する」を参照してください。

コストの分類方法を表すタグキーのセットを規定しておくことをお勧めします。コスト配分レポートの追加の列にタグキー、各行に該当値が表示されます。したがって、一貫したタグキーのセットを使用すると、より効率的にコストを追跡できます。たとえば、複数の Lightsail リソースに特定のコストセンターをタグとして追加できます。これを行うには、「Cost center」キーと数値のペアを使用します。次に、このコストセンターに関する請求を複数のリソースをまたいで表示するように請求情報を整理します。次の例は、コスト配分を整理するために使用できるキーと値のタグを示しています。



ステップ 2: ユーザー定義のコスト配分タグを有効にする

必要なタグを Lightsail リソースに追加したら、これらのタグを[請求とコスト管理]コンソールでコスト配分用に有効化します。たとえば、「Cost center」キータグを作成したら、このキータグを [請求とコスト管理] コンソールで有効にして、このタグのコスト配分レポートを生成します。詳細については、 AWS Billing and Cost Management ドキュメントの「ユーザー定義のコスト配分タグのアクティブ化」を参照してください。

ステップ 3: コスト配分レポートを設定して表示する

月別コスト配分レポートには、製品カテゴリ別および連結アカウントユーザー別のアカウントの AWS 使用状況が一覧表示されます。このレポートには、詳細な請求レポートと同じ明細項目が表示され、さらに追加してタグキー用の列が表示されます。毎月のコスト配分レポートを設定するには、AWS Billing and Cost Management ドキュメントの「毎月のコスト配分レポートの設定」を参照してください。

レポートの保存先の Amazon Simple Storage Service (Amazon S3) バケットは、コスト配分レポートの設定時に定義済みです。この定義済みの Amazon S3 バケットを開き、利用可能になったコスト配分レポートを開きます。コスト配分レポートの内容の詳細については、 AWS Billing and Cost Management ドキュメントの「コスト配分レポートの表示」を参照してください。

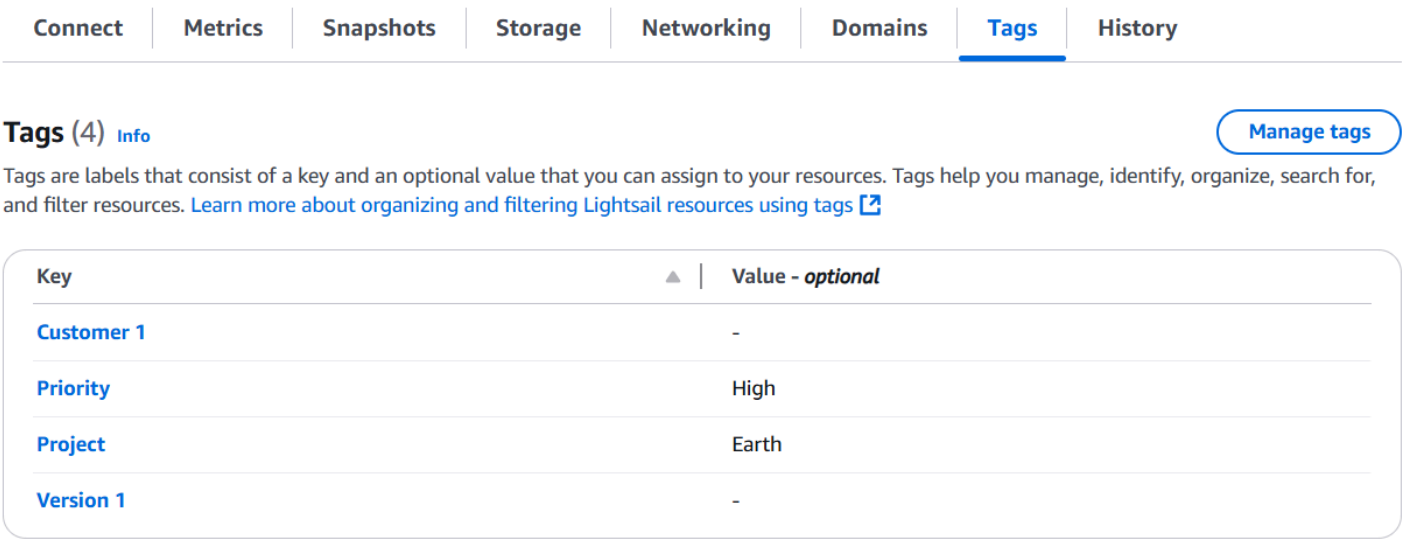
整理とフィルタリングのためのタグ Lightsail リソース

Amazon Lightsail リソースにタグを追加し、追加したタグでリソースをフィルタ処理できます。この操作を行うには、Lightsail コンソールでタグを選択または検索します。このガイドでは、タグを使用して Lightsail リソースを表示およびフィルタ処理する方法を示します。

> **Note**
>
> タグ、タグ付けできるリソース、およびタグの制限の詳細については、「タグ」を参照してください。

リソースのタグを表示する

インスタンス、コンテナサービス、CDN ディストリビューション、バケット、データベース、ディスク、DNS ゾーン、およびロードバランサーには、Lightsail コンソールを使ってタグ付けできるので、[Tags (タグ)] タブを含みます。このタブには、リソースの管理ページからアクセスできます。次の例は、インスタンスリソースの [タグ] タブです。[タグ] タブでは、タグを追加、編集、または削除できます。詳細については、「リソースにタグを追加する」と「タグを削除する」を参照してください。



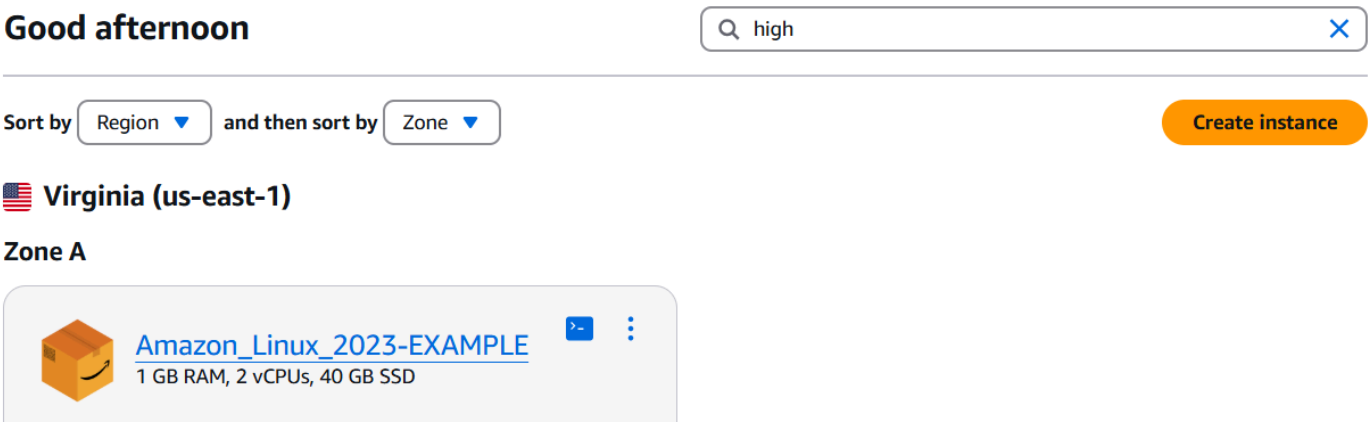
> **Note**
>
> インスタンス、コンテナサービス、CDN ディストリビューション、バケット、データベース、ディスク、DNS ゾーン、およびロードバランサーには、Lightsail コンソールを使ってタグ付けできます。ただし、Lightsail API オペレーション、AWS Command Line Interface (AWS CLI)、または SDK を使用することで、より多くの Lightsail リソースにタグ付けすることができます。タグ付けをサポートする Lightsail リソースの詳細なリストについては、「タグ」を参照してください。

タグを使用してリソースをフィルタ処理する

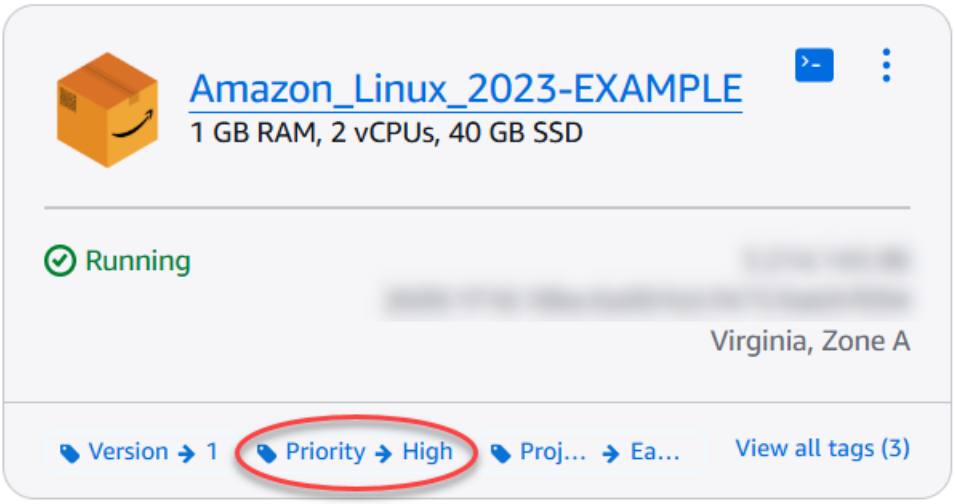
Lightsail コンソールでは、タグを使用してリソースをフィルタ処理するための以下のオプションを利用できます。どのオプションを使用した場合でも、Lightsail のホームページが更新され、検索または選択したタグのみが表示されます。

> **Note**
>
> フィルタ処理の各オプションは永続的です。タグでフィルタ処理した後で、Lightsail のホームページでセクション間を移動すると、フィルタが継続して適用されます。

- Lightsail のホームページで、[検索] テキストボックスにキーのみのタグまたはフィルタ処理用の値を入力し、Enter キーを押します。



- Lightsail のホームページでリソースの下に表示されるタグを選択します。



一般的な Lightsail リソース問題をトラブルシューティングする

このセクションでは、次の Amazon Lightsail リソースのトラブルシューティングトピックについて説明します。ステップバイステップの手順とガイダンスに従って、Lightsail インスタンス、データベース、ネットワーク、ロードバランサー、その他のリソースの操作中に発生する可能性のある一般的な問題を診断し、解決します。

トラブルシューティングのトピックでは、WordPress 設定の失敗、IAM アクセス許可の問題、ディスクエラー、接続の問題、サービスの利用不能、IPv6 接続、インスタンス容量の制限、ロードバランサーエラー、通知配信の失敗、SSL/TLS 証明書の問題など、さまざまなシナリオについて説明します。このガイドに従うことで、Lightsail リソースに関連するさまざまな問題を効果的にトラブルシューティングして解決し、アプリケーションとワークロードの円滑な運用と最適なパフォーマンスを確保できます。

トピック

- Lightsail インスタンスでの WordPress セットアップ問題のトラブルシューティング
- Lightsail コンソールで 403 (未承認) エラーを解決する
- Lightsail ディスクのアタッチと使用状況の問題を解決する
- Lightsail ブラウザベースの SSH および RDP クライアントで接続エラーを解決する
- Lightsail における Ghost インスタンスの 503 サービス使用不可エラーのトラブルシューティング
- Lightsail の Identity and Access Management (IAM) のトラブルシューティング
- Lightsail インスタンスの IPv6 到達可能性を検証する
- Lightsail のインスタンス容量不足エラーの解決
- Lightsail ロードバランサーに関する問題をトラブルシューティングする
- Lightsail での通知配信のトラブルシューティング
- Lightsail の SSL/TLS 証明書のトラブルシューティング

Lightsail インスタンスでの WordPress セットアップ問題のトラブルシューティング

Amazon Lightsail の WordPress 設定ワークフロー中に、次の 2 種類のエラーメッセージが表示されることがあります:

一般的なエラー

これらのタイプのエラーは、ワークフローの最終ステップで [証明書の作成] を選択するとすぐに発生します。これらのエラーは、Lightsail コンソールの上部にあるバナーに表示されます。通常、古い WordPress インスタンスで設定ワークフローを実行するか、誤った情報を送信することで発生します。例えば、インスタンスのパブリック IP アドレスを指さない DNS レコードを選択したこと。

設定失敗

これらのタイプのエラーは、ワークフローの最終ステップを完了してから数分以内に発生します。これらの失敗メッセージは、インスタンス[接続] タブの [WordPress ウェブサイトの設定] セクションに表示されます。これらのエラーは、Let's Encrypt HTTPS 証明書をインスタンスに設定できない場合に発生します。

以下のトピックの情報を使用して、WordPress セットアップガイド付きワークフローで発生する可能性のあるエラーを診断して修正します。

トピック

- Lightsail での WordPress セットアップエラーの解決
- Lightsail での WordPress 設定失敗のトラブルシューティング

Amazon Lightsail の WordPress 設定ガイド付きワークフローの詳細については、「Configure your WordPress instance」を参照してください。

Lightsail での WordPress セットアップエラーの解決

ワークフロー中に送信された情報に問題がある場合、Lightsail コンソールの上部にエラーメッセージが表示されます。

メッセージの最初の行は、セットアップでエラーが発生したことを知らせます:

InstanceRegion リージョン内のインスタンス *InstanceName* の設定を完了できませんでした。

2 行目には、設定で発生したエラーが含まれています:

エラーが発生し、インスタンスに接続したり、接続したままにしたりできませんでした

トラブルシューティングを開始するには、メッセージに表示されるエラーを、次のいずれかのエラーと一致させます。

エラー

- DNS レコードが見つかりません。ドメインの DNS レコードがインスタンスのパブリック IP アドレスを指し、DNS の変更が伝播する時間を確保します。
- DNS レコードが一致しません。ドメインの DNS レコードがインスタンスのパブリック IP アドレスを指し、DNS の変更が伝播する時間を確保します。
- インスタンスに接続できません。SSH 接続の準備が整うまで数分待ちます。次に、設定を再度開始します。
- サポートされていない WordPress バージョン。設定は WordPress バージョン 6 以降のみをサポートします。
- セットアップは、2023 年 1 月 1 日以降に作成された WordPress インスタンスのみをサポートします。
- インスタンスファイアウォールポート 22、80、および 443 は、セットアップワークフロー中に任意の IP アドレスからの TCP 接続を許可する必要があります。これらの設定は、インスタンスの [ネットワーク] タブから変更できます。

DNS レコードが見つかりません。ドメインの DNS レコードがインスタンスのパブリック IP アドレスを指し、DNS の変更が伝播する時間を確保します。

Reason

このエラーは、DNS レコードの設定ミスや、インターネット上の DNS に完全に伝播するまでの時間が不足していることが原因で発生します。

修正

[A] または [AAAA] DNS レコードが DNS ゾーンに存在し、インスタンスのパブリック IP アドレスを指していることを確認します。詳細については、「Lightsail のDNS」を参照してください。

頂点ドメイン (`example.com`) とその `www` サブドメイン (`www.example.com`) からのトラフィックをポイントする DNS レコードを追加または更新する場合、それらはインターネットの DNS 全体に伝播する必要があります。「nslookup」や「MxToolbox」の「DNS Lookup」などのツールを使用して、DNS の変更が有効になったことを確認できます。

> **Note**
>
> すべての DNS レコードの変更がインターネットの DNS を通じて伝播されるまで数時間かかる場合があります。

DNS レコードが一致しません。ドメインの DNS レコードがインスタンスのパブリック IP アドレスを指し、DNS の変更が伝播する時間を確保します。

Reason

[A] または [AAAA] DNS レコードは、インスタンスのパブリック IP アドレスを指していません。

修正

[A] または [AAAA] DNS レコードが DNS ゾーンに存在し、インスタンスのパブリック IP アドレスを指していることを確認します。詳細については、「Lightsail のDNS」を参照してください。

> **Note**
>
> すべての DNS レコードの変更がインターネットの DNS を通じて伝播されるまで数時間かかる場合があります。

インスタンスに接続できません。SSH 接続の準備が整うまで数分待ちます。次に、設定を再度開始します。

Reason

インスタンスが作成されたばかり、または再起動されたため、SSH 接続の準備が整っていません。

修正

SSH 接続の準備が整うまで数分待ちます。次に、ガイド付きのワークフローを再試行します。詳細については、「Lightsail のトラブルシューティング SSH」を参照してください。

サポートされていない WordPress バージョン。設定は WordPress バージョン 6 以降のみをサポートします。

Reason

インスタンスにインストールされている WordPress のバージョンが WordPress 6 よりも古いものです。古いバージョンの WordPress には互換できないソフトウェアや依存関係が含まれており、HTTPS 証明書の生成を妨げます。

Fix

Lightsail コンソールから新しい WordPress インスタンスを作成します。次に、WordPress ウェブサイトを古いインスタンスから新しいインスタンスに移行します。詳細については、「Migrate an existing WordPress blog」を参照してください。

既存のインスタンスを置き換えるために新しいインスタンスを作成する場合は、アプリケーションの依存関係を新しいインスタンスに更新してください。

セットアップは、2023 年 1 月 1 日以降に作成された WordPress インスタンスのみをサポートします。

Reason

セットアップで使用されているインスタンスには、古いソフトウェアが含まれている場合があります。古いソフトウェアを使用すると、HTTPS 証明書が生成できなくなります。

Fix

Lightsail コンソールから新しい WordPress インスタンスを作成します。次に、WordPress ウェブサイトを古いインスタンスから新しいインスタンスに移行します。詳細については、「Migrate an existing WordPress blog」を参照してください。

既存のインスタンスを置き換えるために新しいインスタンスを作成する場合は、アプリケーションの依存関係を新しいインスタンスに更新してください。

インスタンスファイアウォールポート 22、80、および 443 は、セットアップワークフロー中に任意の IP アドレスからの TCP 接続を許可する必要があります。これらの設定は、インスタンスの [ネットワーク] タブから変更できます。

Reason

インスタンスファイアウォールポート 22、80、および 443 は、セットアップの実行中に任意の IP アドレスからの TCP 接続を許可する必要があります。このエラーは、これらのポートの 1 つ以上が閉じられたときに発生します。詳細については、「インスタンスのファイアウォール」を参照してください。

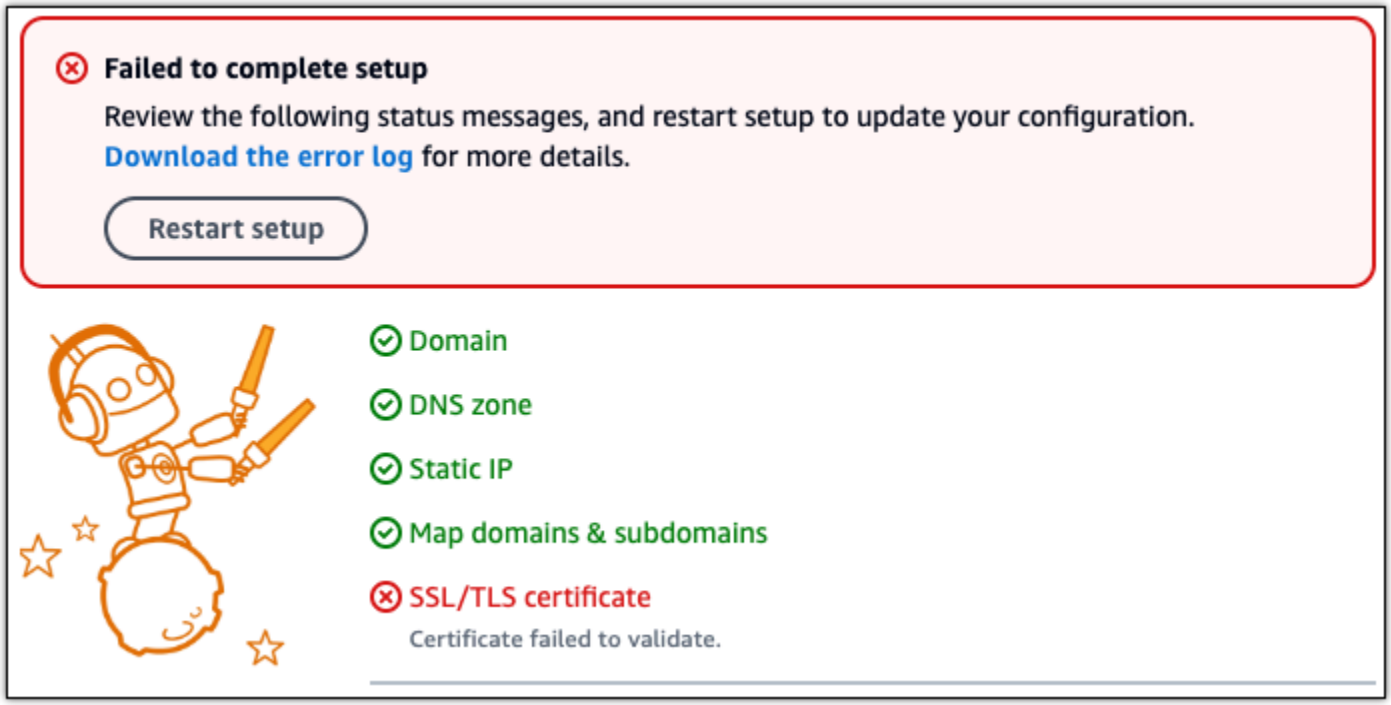
修正

インスタンスの IPv4 および IPv6 ファイアウォールルールを追加または編集して、ポート 22、80、443 を介した TCP 接続を許可します。詳細については、「Add and edit instance firewall rules」を参照してください。

Lightsail での WordPress 設定失敗のトラブルシューティング

以下の情報は、インスタンス [接続] タブの [WordPress ウェブサイトの設定] セクションに表示される可能性のある障害メッセージのトラブルシューティングに役立ちます。設定失敗は、ワークフローの最終ステップを完了してから数分以内に発生する可能性があります。これは、Let's Encrypt HTTPS 証明書をインスタンスに設定できない場合に発生します。

設定は完了できませんでした – 次のステータスメッセージを確認し、セットアップを再起動して設定を更新します。詳細については、エラーログをダウンロードしてください。



失敗メッセージから、[エラーログのダウンロード] リンクを選択して、セットアップで生成されたエラーログをダウンロードして表示します。トラブルシューティングを開始するには、ログのエラーメッセージを以下のエラーのいずれかと一致してください。

Bitnami

Certbot.errors.AuthorizationError: 一部のチャレンジが失敗しました

Reason

このエラーは、DNS レコードの設定ミス、またはインターネット全体に伝播するのに十分な時間がない DNS レコードが原因で発生します。

修正

[A] または [AAAA] DNS レコードが DNS ゾーンに存在し、インスタンスのパブリック IP アドレスを指すことを確認します。詳細については、「Lightsail のDNS」を参照してください。

頂点ドメイン (`example.com`) とその `www` サブドメイン (`www.example.com`) からのトラフィックをポイントする DNS レコードを追加または更新する場合、それらはインターネット全体に伝播する必要があります。「nslookup」や「MxToolbox」の「DNS Lookup」などのツールを使用して、DNS の変更が有効になったことを確認できます。

> **Note**
>
> すべての DNS レコードの変更がインターネットの DNS を通じて伝播されるまで数時間かかる場合があります。

Certbot が一部のドメインの認証に失敗しました

Reason

このエラーは、インスタンスで HTTPS 証明書を構成している間に、他のプロセスがポート 80 を使用している場合に発生する可能性があります。

修正

WordPress インスタンスを再起動します。次に、ガイド付きのワークフローを再度実行します。再起動で問題が解決しない場合は、次の手順を使用して、ポート 80 で実行されているインスタンス上のすべての実行プロセスを終了してください。

手順

1. Lightsail の「ブラウザベースのSSHクライアント」または「AWS CloudShell」を使用してインスタンスに接続します。
2. インスタンスで実行されている Bitnami プロセスを停止します:

```
$ sudo /opt/bitnami/ctlscript.sh stop
```

Bitnami プロセスが停止していることを確認します:

```
$ sudo /opt/bitnami/ctlscript.sh status
```

3. ポート 80 を使用しているプロセスが他にもあるかどうかを確認します:

```
$ fuser -n tcp 80
```

4. 別のアプリケーションで必要とされないプロセスをすべて終了します:

```
$ fuser -k -n tcp 80
```

5. WordPress の設定を再起動します。

リポジトリ http://cdn-aws.deb.debian.org/debian のbuster-backports にリリースファイルがありません

Reason

インスタンスには、更新できない非推奨の Debian リポジトリがあります。

修正

次の手順を使用して、Debian リポジトリファイルに記載されているリポジトリ URL を編集します。

手順

1. Lightsail の「ブラウザベースのSSHクライアント」または「AWS CloudShell」を使用してインスタンスに接続します。
2. /etc/apt/sources.list.d/ ディレクトリに移動します。

```
$ cd /etc/apt/sources.list.d/
```

3. 任意のテキストエディタを使用して、buster-backports.list ファイルを開きます。このディレクトリにファイルが見つからない場合は、/etc/apt/sources.list で確認することもできます。事前インストールされた Vim テキストエディタは、コマンド例で使用されます。詳細については、Vim ドキュメントを参照してください。

```
$ vim buster-backports.list
```

4. 次のテキストを含む行を見つけます: http://deb.debian.org/debian buster-backports main。

   deb.debian.org を archive.debian.org に置き換えます。例えば、http://**deb**.debian.org/debian buster-backports main contrib non-free は http://**archive**.debian.org/debian buster-backports main contrib non-free になります。

5. ファイルを保存して閉じます。
6. WordPress の設定を再起動します。

リポジトリ http://ppa.launchpad.net/certbot/certbot/ubuntu 月のリリースにリリースファイルがありません

Reason

インスタンスには、更新できない廃止された Certbot Personal Package Archive (PPA) リポジトリがあります。

修正

非推奨の PPA リポジトリをインスタンスから手動で削除するには、次の手順に従います。

手順

1. Lightsail の「ブラウザベースのSSHクライアント」または「AWS CloudShell」を使用してインスタンスに接続します。
2. `/etc/apt/sources.list.d/` ディレクトリに移動します。

```
$ cd /etc/apt/sources.list.d/
```

3. 任意のテキストエディタを使用して、`certbot-ubuntu-certbot-`**`version`**`.list` ファイルを開きます。事前インストールされた Vim テキストエディタは、コマンド例で使用されます。詳細については、Vim ドキュメントを参照してください。

   コマンド内の **`version`** をリポジトリが非互換である Ubuntu のバージョンに置き換えます。このバージョンはエラーメッセージに表示されるものと同じです。例えば、**`lunar`**、**`mantic`** です。

```
$ vim certbot-ubuntu-certbot-version.list
```

4. 次のテキストを含む行をすべて削除します: `http://ppa.launchpad.net/certbot/certbot/ubuntu`。
5. ファイルを保存して閉じます。
6. WordPress の設定を再起動します。

過去 168 時間にこの正確なドメインセットに対して発行された証明書 (5) が多すぎます

Reason

1 つ以上のドメインまたはサブドメインが、過去 1 週間以内に 5 つの証明書を作成するために既に使用されています。詳細については、Let's Encrypt ウェブサイトの「Rate Limits」を参照してください。

修正

1 週間 (168 時間) 待ってから、このドメインのガイド付きワークフローを再起動します。

失敗した認証が多すぎます

Reason

リクエスト内の 1 つ以上のドメインまたはサブドメインが、1 時間あたり 5 つの検証の制限を超えています。詳細については、Let's Encrypt ウェブサイトの「Rate Limits」を参照してください。

修正

1 時間待ってから、WordPress のセットアップを再度実行します。設定を再起動する前に、他の検証エラーが修正されていることを確認します。

Lightsail

Certbot.errors.AuthorizationError: 一部のチャレンジが失敗しました

Reason

このエラーは、DNS レコードの設定ミス、またはインターネット全体に伝播するのに十分な時間がない DNS レコードが原因で発生します。

修正

[A] または [AAAA] DNS レコードが DNS ゾーンに存在し、インスタンスのパブリック IP アドレスを指すことを確認します。詳細については、「Lightsail のDNS」を参照してください。

頂点ドメイン (`example.com`) とその `www` サブドメイン (`www.example.com`) からのトラフィックをポイントする DNS レコードを追加または更新する場合、それらはインターネッ

ト全体に伝播する必要があります。「nslookup」や「MxToolbox」の「DNS Lookup」などのツールを使用して、DNS の変更が有効になったことを確認できます。

> **Note**
>
> すべての DNS レコードの変更がインターネットの DNS を通じて伝播されるまで数時間かかる場合があります。

Certbot が一部のドメインの認証に失敗しました

Reason

このエラーは、インスタンスで HTTPS 証明書を構成している間に、他のプロセスがポート 80 を使用している場合に発生する可能性があります。

修正

WordPress インスタンスを再起動します。次に、ガイド付きのワークフローを再度実行します。再起動しても問題が解決しない場合は、ポート 80 で実行されているインスタンスで実行中のプロセスを終了するには、次の手順に従います。

手順

1. Lightsail の「ブラウザベースのSSHクライアント」または「AWS CloudShell」を使用してインスタンスに接続します。
2. インスタンスで実行されている Apache サービスを停止します。

```
$ sudo systemctl stop apache2
```

Apache サービスが停止していることを確認します。

```
$ sudo systemctl status apache2
```

3. ポート 80 を使用しているプロセスが他にもあるかどうかを確認します:

```
$ fuser -n tcp 80
```

4. 別のアプリケーションで必要とされないプロセスをすべて終了します:

```
$ fuser -k -n tcp 80
```

5. WordPress の設定を再起動します。

リポジトリ http://cdn-aws.deb.debian.org/debian のbuster-backports にリリースファイルがありません

Reason

インスタンスには、更新できない非推奨の Debian リポジトリがあります。

修正

次の手順を使用して、Debian リポジトリファイルに記載されているリポジトリ URL を編集します。

手順

1. Lightsail の「ブラウザベースのSSHクライアント」または「AWS CloudShell」を使用してインスタンスに接続します。
2. /etc/apt/sources.list.d/ ディレクトリに移動します。

```
$ cd /etc/apt/sources.list.d/
```

3. 任意のテキストエディタを使用して、buster-backports.list ファイルを開きます。このディレクトリにファイルが見つからない場合は、/etc/apt/sources.list で確認することもできます。事前インストールされた Vim テキストエディタは、コマンド例で使用されます。詳細については、Vim ドキュメントを参照してください。

```
$ vim buster-backports.list
```

4. 次のテキストを含む行を見つけます: http://deb.debian.org/debian buster-backports main。

   deb.debian.org を archive.debian.org に置き換えます。例えば、http://**deb**.debian.org/debian buster-backports main contrib non-free は http://**archive**.debian.org/debian buster-backports main contrib non-free になります。
5. ファイルを保存して閉じます。

6. WordPress の設定を再起動します。

リポジトリ http://ppa.launchpad.net/certbot/certbot/ubuntu 月のリリースにリリースファイルがありません

Reason

インスタンスには、更新できない廃止された Certbot Personal Package Archive (PPA) リポジトリがあります。

修正

非推奨の PPA リポジトリをインスタンスから手動で削除するには、次の手順に従います。

手順

1. Lightsail の「ブラウザベースのSSHクライアント」または「AWS CloudShell」を使用してインスタンスに接続します。
2. `/etc/apt/sources.list.d/` ディレクトリに移動します。

```
$ cd /etc/apt/sources.list.d/
```

3. 任意のテキストエディタを使用して、`certbot-ubuntu-certbot-`**`version`**`.list` ファイルを開きます。事前インストールされた Vim テキストエディタは、コマンド例で使用されます。詳細については、Vim ドキュメントを参照してください。

   コマンド内の **`version`** をリポジトリが非互換である Ubuntu のバージョンに置き換えます。このバージョンはエラーメッセージに表示されるものと同じです。例えば、**`lunar`**、**`mantic`** です。

```
$ vim certbot-ubuntu-certbot-version.list
```

4. 次のテキストを含む行をすべて削除します: `http://ppa.launchpad.net/certbot/certbot/ubuntu`。
5. ファイルを保存して閉じます。
6. WordPress の設定を再起動します。

過去 168 時間にこの正確なドメインセットに対して発行された証明書 (5) が多すぎます

Reason

1 つ以上のドメインまたはサブドメインが、過去 1 週間以内に 5 つの証明書を作成するために既に使用されています。詳細については、Let's Encrypt ウェブサイトの「レート制限」を参照してください。

Fix

1 週間 (168 時間) 待ってから、このドメインのガイド付きワークフローを再起動します。

失敗した認証が多すぎます

Reason

リクエスト内の 1 つ以上のドメインまたはサブドメインが、1 時間あたり 5 つの検証の制限を超えています。詳細については、Let's Encrypt ウェブサイトの「レート制限」を参照してください。

修正

1 時間待ってから、WordPress のセットアップを再度実行します。設定を再起動する前に、他の検証エラーが修正されていることを確認します。

Lightsail コンソールで 403 (未承認) エラーを解決する

Lightsail コンソールにアクセスしようとしたときに403 エラーが表示された場合、あわてる必要はありません。問題のトラブルシューティングを行うには、以下のステップを試してください。

- AWS アカウントまたは AWS Identity and Access Management (IAM) ユーザーが最近作成された場合は、数分待ってからブラウザを更新します。
- 最後にサインインしてから時間が経っている場合は、ブラウザを更新します。再度サインインするように求められた場合は、必ず Lightsail にアクセスできる IAM ユーザーを使用してください。
- IAM ユーザーに Lightsail にアクセスする権利がない場合は、AWS アカウントのルートユーザーまたは管理者アクセス権を持つ IAM ユーザーに問い合わせて、Lightsail へのアクセス権をリクエストします。詳細については、「IAM ユーザーの Amazon Lightsail へのアクセスを管理します。」を参照してください。

- 上記のステップを試した後で 403 エラーが続く場合は、AWS カスタマーサポートにお問い合わせください。2011 年より前に作成された AWS アカウントの場合、まれに、サポートはアカウントを に手動でサブスクライブする必要がありますLightsail。

Lightsail ディスクのアタッチと使用状況の問題を解決する

Lightsail のブロックストレージディスクでエラーが発生する可能性があります。このトピックでは、一般的な問題とそれらのエラーの回避策について説明します。

一般的なディスクエラー

以下の問題の中から発生している問題に最も近いものを選択し、リンク先に移動して問題を解決します。リストにない問題が発生した場合、このページの一番下にある [ご質問は? [意見などありますか ?] このページ下部のリンクにアクセスしてフィードバックを送信するか、AWS サポートにお問い合わせください。

インスタンスにアタッチされたままのためディスクを削除できない。

まず、ディスクをインスタンスからデタッチし、その後ディスクを削除してください。詳細については、「ブロックストレージディスクをデタッチおよび削除する」を参照してください。

実際のエラーメッセージ: You can't perform this operation because the disk is still attached to a Lightsail instance: ***YOUR_INSTANCE***

ディスクのステータスがエラーです。

[エラー] ステータスは、Lightsail ディスクに関連する基盤ハードウェアに障害が発生したことを示します。ディスクを最新のスナップショットから復元できます。そうしない場合、ディスクに関連するデータを回復できません。詳細については、「スナップショットからブロックストレージディスクを作成する」を参照してください。

[エラー] のステータスのディスクについては請求されません。

Lightsail インスタンスはまだ実行中のためディスクをデタッチできない。

まず、インスタンスを停止し、その後ディスクをデタッチしてください。詳細については、「インスタンスの停止」を参照してください。

実際のエラーメッセージ:You can't detach this disk right now. このディスクの状態: ***DISK_STATE***

16 TB (16,384 GB) より大きいカスタムディスクサイズを指定できない。

小さいディスクを作成してみます。追加ディスクは、最大 16 TB です。ディスクが 16 TB 未満の場合でも作成できない場合、リスト内の次のエラーが発生する可能性があります (大きいディスクが多すぎる)。これは、AWS アカウント全体の追加ディスクストレージが 20 TB を超えることができないためです。詳細については、「ブロックストレージディスク」を参照してください。

実際のエラーメッセージ:The size of a block storage disk must be between 8 and 16384 GB.

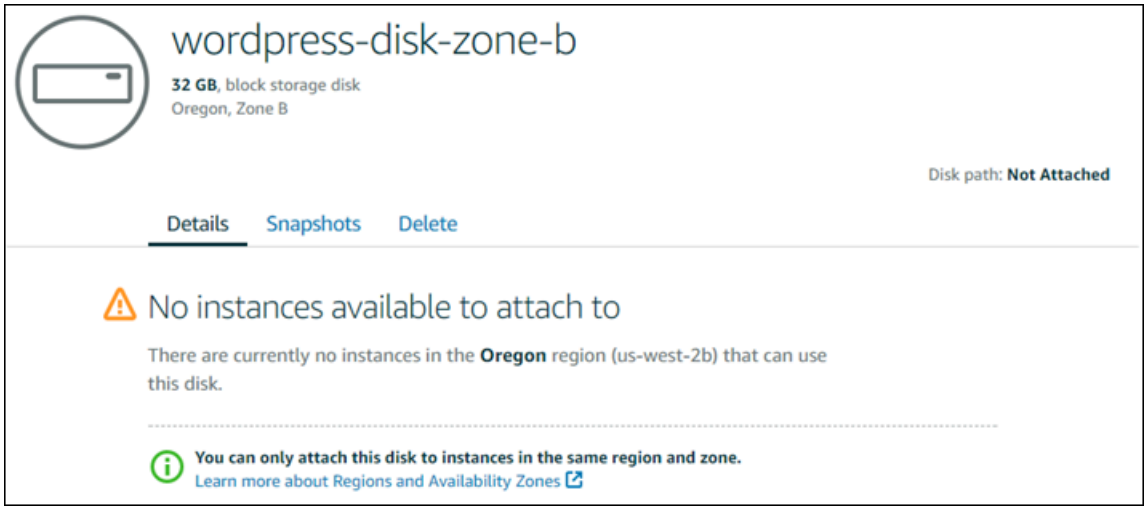
Lightsail でさらにディスクを作成できない。

作成できるディスク数のクォータに達した可能性があります。または、AWS アカウントで大きいディスクを多く作成しすぎた可能性があります (ディスクストレージの合計サイズが 20 TB を超えることはできません)。詳細については、「ブロックストレージディスク」を参照してください。

実際のエラーメッセージ: You've reached the maximum size limit of all disks in this account. or You've reached the limit of disks in this account.

ディスクを Lightsail インスタンスにアタッチできない

次のエラーが発生した場合、ディスクをアタッチする予定のインスタンスと同じ AWS リージョンおよびアベイラビリティーゾーンにディスクを再作成する必要があります。



実際のエラーメッセージ:There are currently no instances in the ***AWS Region*** that can use this disk.

Lightsail ブラウザベースの SSH および RDP クライアントで接続エラーを解決する

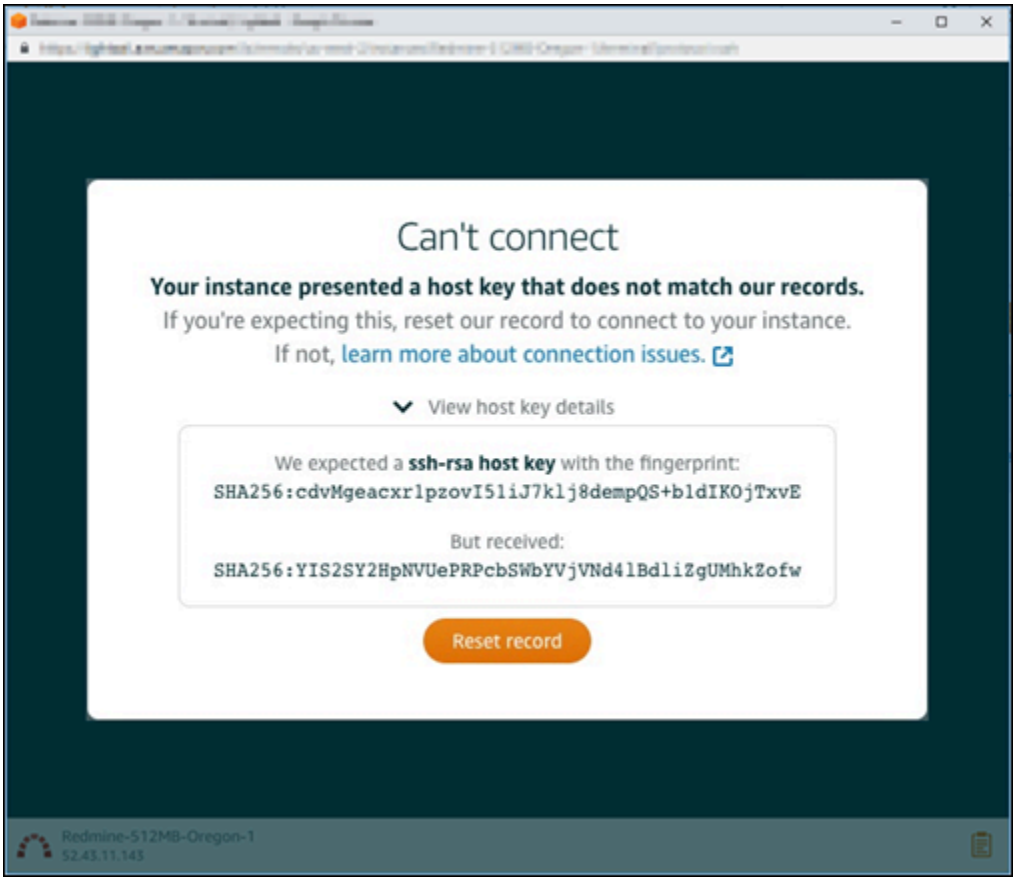
Amazon Lightsail コンソールで使用可能なブラウザベースの SSH または RDP クライアントを使用してインスタンスに接続しようとすると、エラーメッセージが表示されることがあります。このエラーが表示される理由として考えられるものは、次のセクションで説明します。

エラーメッセージ: 接続できません

接続を試みると、SSH および RDP ブラウザベースのクライアントは、ホストキーまたは証明書の検証を使用してインスタンスを認証します。インスタンスにより表示されるホストキーまたは証明書が、Lightsail がレコードに保持するものと一致しない場合、2 つのエラーメッセージのうちいずれか 1 つが表示されます。このセクションでは、両方のエラーメッセージが表示・説明されています。

接続できません。レコードをリセットしてください

次のエラーメッセージは、ホストキーまたは証明書が一致せず、その不一致の理由が、オペレーティングシステムの最近のアップグレードやユーザーまたは他のユーザーによるホストキーまたは証明書の意図的な更新が原因であると Lightsail が判断した場合に表示されます。この場合 Lightsail は、ホストキーまたは証明書の不一致が、ブラウザ - インスタンス間のネットワークの不正なアクターによって引き起こされたものではないと判断しました。



不一致が予想される場合、[Reset record (レコードをリセット)] を選択します。このアクションにより、Lightsail がインスタンスのレコードに保持するホストキーまたは証明書が削除され、ブラウザベースの SSH または RDP セッションによるインスタンスへの接続が許可されます。

次の Lightsail (AWS Command Line Interface) コマンドを使用して、AWS CLI がレコードに保持するホストキーまたは証明書を削除することもできます。[*InstanceName*] では、既知のホストキーまたは証明書を削除するインスタンスの名前を入力します。[*Region*] では、インスタンスの AWS リージョンを入力します。

```
aws lightsail delete-known-host-keys --region Region --instance-name InstanceName
```

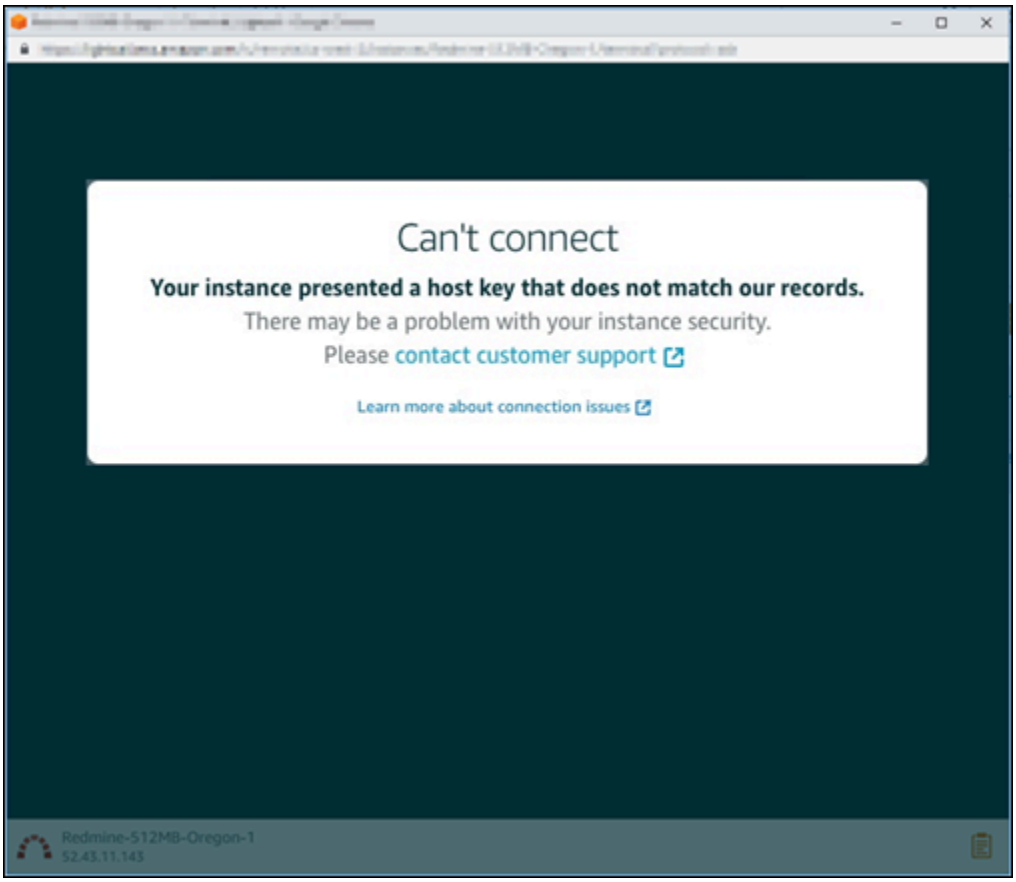
例:

```
aws lightsail delete-known-host-keys --region us-west-2 --instance-
name WordPress-512MB-Oregon-1
```

> **Note**
>
> AWS CLI の詳細については、「AWS CLI で使用するために Lightsail を設定する」を参照してください。

接続できません。カスタマーサポートにご連絡ください

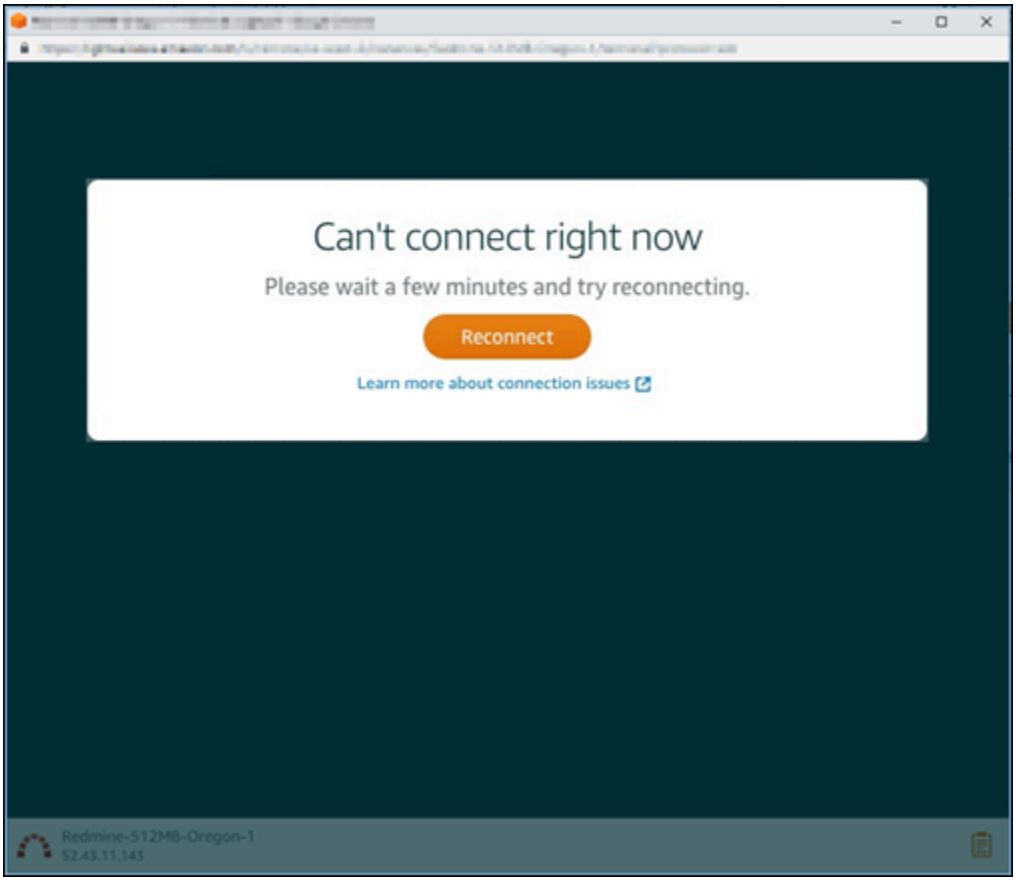
次のエラーメッセージは、ホストキーまたは証明書が一致せず、中間者 (MITM) 攻撃など、さらに調査を必要とする疑わしいアクティビティがあると Lightsail が判断した場合に表示されます。



このエラーメッセージは、ブラウザベースの SSH または RDP クライアントを使用してインスタンスに接続できないことを意味します。サポートにご連絡ください。

エラーメッセージ: 現在接続できません。

インスタンスを作成、再起動、または再起動のいずれかを行った後にまだ起動していないインスタンスに接続しようとすると、次のエラーメッセージが表示されます。数分間待機した後、[Reconnect (再接続)] を選択して再度試してください。



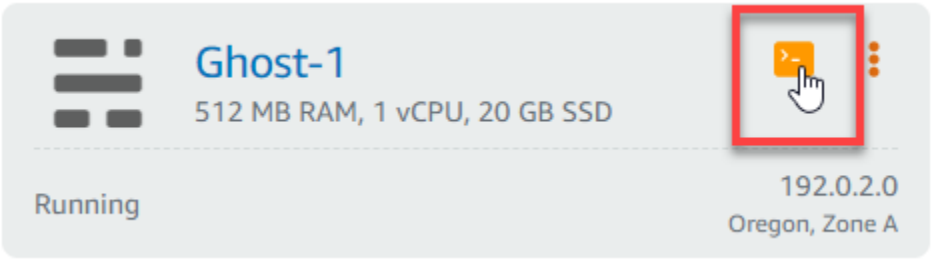
上記の手順を試しても接続できない場合は、AWS サポートにお問い合わせください。

Lightsail における Ghost インスタンスの 503 サービス使用不可エラーのトラブルシューティング

Amazon Lightsail で新しい Ghost インスタンスを作成し、ウェブサイトにアクセスしようとすると、サービスが利用できないことを示すエラー (503) が表示される場合があります。場合によっては、インスタンスの作成時にインスタンスの Ghost サービスが自動的に開始されないことがあります。このエラーは、インスタンスに 5 USD/月のバンドルを選択したときに発生する可能性があります。以下の手順を使用して Ghost サービスを開始し、サービス使用不可エラーを解決します。

Ghost サービスの開始

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [インスタンス] を選択してください。
3. Ghost インスタンスのブラウザベースの SSH クライアントアイコンを選択します。



4. SSH クライアントが接続されたら、以下のコマンドを入力してインスタンスですべてのサービスを再起動します。

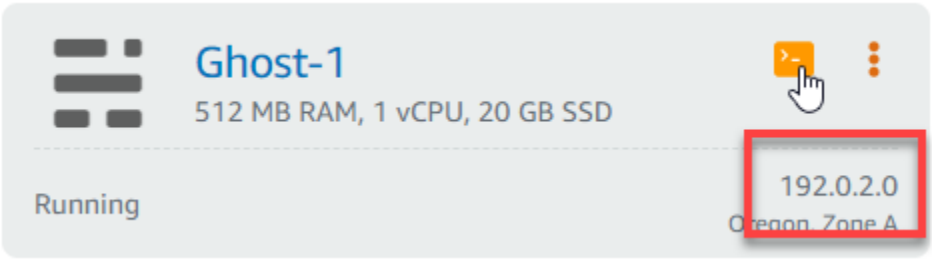
```
sudo /opt/bitnami/ctlscript.sh restart
```

以下の例のような結果が表示されるはずです。

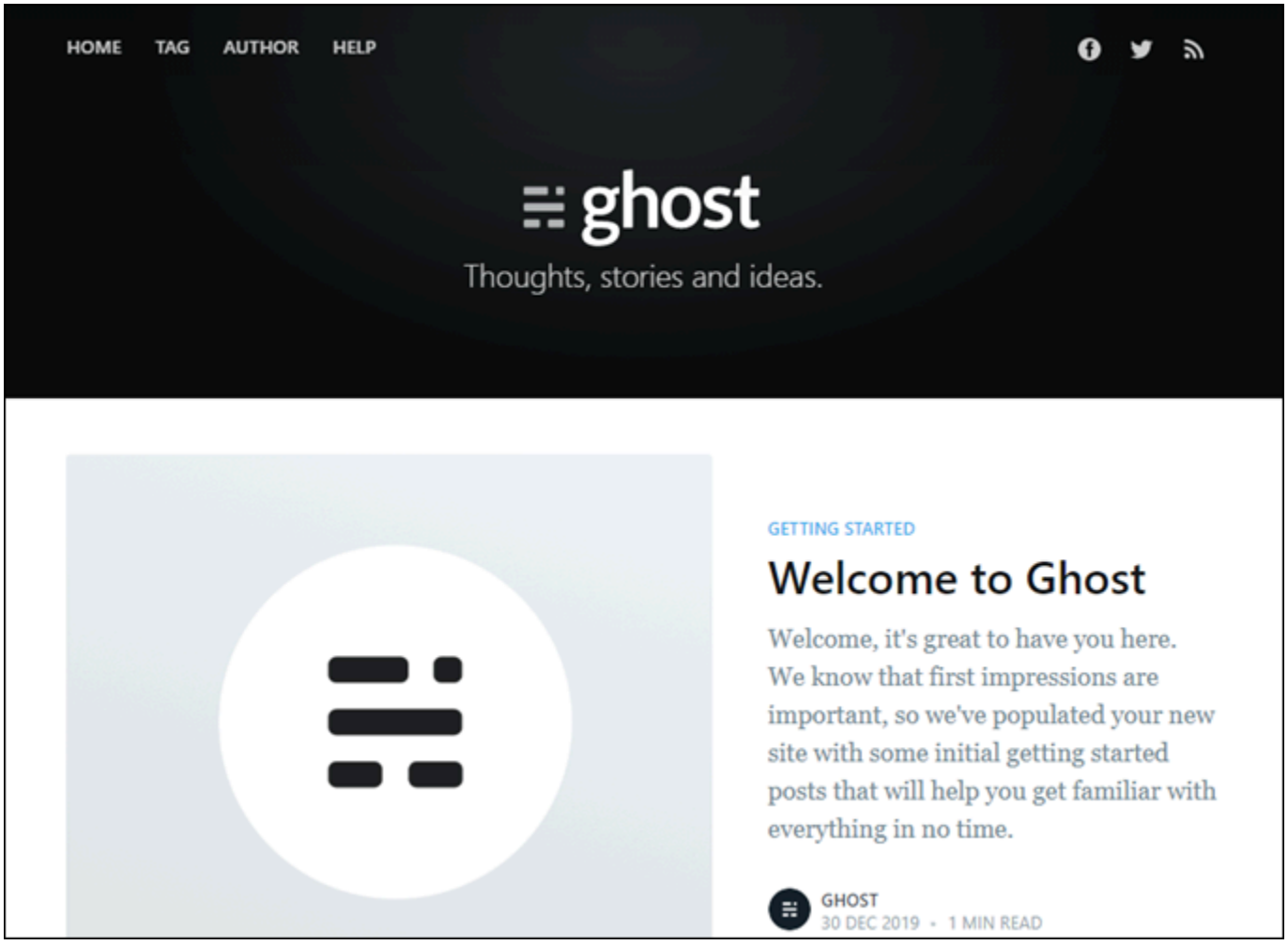
```
bitnami@ip-172-26-11-214:~$ sudo /opt/bitnami/ctlscript.sh restart
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd stopped
/opt/bitnami/apps/ghost/scripts/ctl.sh : ghost not running
/opt/bitnami/mysql/scripts/ctl.sh : mysql stopped
/opt/bitnami/mysql/scripts/ctl.sh : mysql  started at port 3306
» Ensuring user is not logged in as ghost user [skipped]
» Checking if logged in user is directory owner [skipped]
✔ Checking current folder permissions
✔ Validating config
✔ Checking memory availability
✔ Checking binary dependencies
✔ Starting Ghost: 127-0-0-1

------------------------------------------------------------------------------

Your admin interface is located at:

    http://18.237.117.48:80/ghost/

/opt/bitnami/apps/ghost/scripts/ctl.sh : ghost started
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd started at port 80
```

5. インスタンスのパブリック IP アドレスを参照して、Ghost ウェブサイトが稼働中であることを確認します。

インスタンスのパブリック IP アドレスは、Lightsail コンソールの [インスタンス] セクションのインスタンス名の横に記載されています。



新しい Ghost インスタンスのパブリック IP を参照すると、デフォルトの Ghost ウェブサイトテンプレートが表示されます。



Lightsail の Identity and Access Management (IAM) のトラブルシューティング

以下の情報は、Lightsail と IAM の使用に伴って発生する可能性がある一般的な問題の診断や修復に役立ちます。

Lightsail でアクションを実行する権限がない

にアクションを実行する権限がないと AWS マネジメントコンソール 通知された場合は、管理者に連絡してサポートを依頼する必要があります。担当の管理者はお客様のユーザー名とパスワードを発行した人です。

次のエラー例は、`mateojackson` IAM ユーザーが Lightsail コンソールにアクセスしようとしたが、`lightsail:*` (フルアクセス) アクセス許可がない場合に発生します。

この場合、Mateo は、`lightsail:*` (フルアクセス) アクセス許可を使用して Lightsail コンソールにアクセスできるように、ポリシーの更新を管理者に依頼します。

iam:PassRole を実行する権限がない

iam:PassRole アクションを実行する権限がないというエラーが表示された場合は、ポリシーを更新して Amazon Lightsail にロールを渡すことができるようにする必要があります。

一部の AWS のサービス では、新しいサービスロールまたはサービスにリンクされたロールを作成する代わりに、そのサービスに既存のロールを渡すことができます。そのためには、サービスにロールを渡す権限が必要です。

以下の例のエラーは、marymajor という IAM ユーザーがコンソールを使用して Amazon Lightsail でアクションを実行しようとする場合に発生します。ただし、このアクションをサービスが実行するには、サービスロールから付与された権限が必要です。Mary には、ロールをサービスに渡すアクセス許可がありません。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

この場合、Mary のポリシーを更新してメアリーに iam:PassRole アクションの実行を許可する必要があります。

サポートが必要な場合は、 AWS 管理者にお問い合わせください。サインイン資格情報を提供した担当者が管理者です。

アクセスキーを表示したい

IAM ユーザーアクセスキーを作成した後は、いつでもアクセスキー ID を表示できます。ただし、シークレットアクセスキーを再表示することはできません。シークレットアクセスキーを紛失した場合は、新しいアクセスキーペアを作成する必要があります。

アクセスキーは、アクセスキー ID (例: AKIAIOSFODNN7EXAMPLE) とシークレットアクセスキー (例: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY) の 2 つで構成されています。ユーザー名とパスワードと同様に、リクエストを認証するために、アクセスキー ID とシークレットアクセスキーの両方を使用する必要があります。ユーザー名とパスワードと同様に、アクセスキーは安全に管理してください。

> **Important**
>
> 正規のユーザー ID を確認するためであっても、アクセスキーを第三者に提供しないでください。これにより、 への永続的なアクセス権をユーザーに付与できます AWS アカウント。

アクセスキーペアを作成する場合、アクセスキー ID とシークレットアクセスキーを安全な場所に保存するように求めるプロンプトが表示されます。このシークレットアクセスキーは、作成時にのみ使用できます。シークレットアクセスキーを紛失した場合、IAM ユーザーに新規アクセスキーを追加する必要があります。アクセスキーは最大 2 つまで持つことができます。既に 2 つある場合は、新規キーペアを作成する前に、いずれかを削除する必要があります。手順を表示するには、IAM ユーザーガイドの「アクセスキーの管理」を参照してください。

管理者として Lightsail へのアクセスを他のユーザーに許可したい

他のユーザーが Amazon Lightsail にアクセスすることを許可するには、アクセスが必要なユーザーまたはアプリケーションにアクセス許可を付与する必要があります。 AWS IAM アイデンティティセンター を使用してユーザーとアプリケーションを管理する場合は、アクセスレベルを定義するアクセス許可セットをユーザーまたはグループに割り当てます。アクセス許可セットは、ユーザーまたはアプリケーションに関連付けられている IAM ロールに自動的に IAM ポリシーを作成して割り当てます。詳細については、「AWS IAM アイデンティティセンター ユーザーガイド」の「アクセス許可セット」を参照してください。

IAM アイデンティティセンターを使用していない場合は、アクセスを必要としているユーザーまたはアプリケーションの IAM エンティティ (ユーザーまたはロール) を作成する必要があります。次に、Amazon Lightsail の適切なアクセス許可を付与するポリシーを、そのエンティティにアタッチする必要があります。アクセス許可が付与されたら、ユーザーまたはアプリケーション開発者に認証情報を提供します。これらの認証情報を使用して AWSにアクセスします。IAM ユーザー、グループ、ポリシー、アクセス許可の作成の詳細については、「IAM ユーザーガイド」の「IAM アイデンティティ」と「IAM のポリシーとアクセス許可」を参照してください。

自分の AWS アカウント以外のユーザーに自分のLightsailリソースへのアクセスを許可したい

他のアカウントのユーザーや組織外の人が、リソースにアクセスするために使用できるロールを作成できます。ロールの引き受けを委託するユーザーを指定できます。リソースベースのポリシーまたはアクセスコントロールリスト (ACL) をサポートするサービスの場合、それらのポリシーを使用して、リソースへのアクセスを付与できます。

詳細については、以下を参照してください。

- Amazon Lightsail がこれらの機能をサポートしているかどうかを確認するには、「Amazon Lightsail と IAM の連携方法」をご参照ください。

- 所有 AWS アカウント している のリソースへのアクセスを提供する方法については、「IAM ユーザーガイド」の「所有 AWS アカウント している別の の IAM ユーザーへのアクセスを提供する」を参照してください。
- リソースへのアクセスをサードパーティーに提供する方法については AWS アカウント、IAM ユーザーガイドの「サードパーティー AWS アカウント が所有する へのアクセスを提供する」を参照してください。
- ID フェデレーションを介してアクセスを提供する方法については、IAM ユーザーガイド の 外部で認証されたユーザー (ID フェデレーション) へのアクセスの許可 を参照してください。
- クロスアカウントアクセスにおけるロールとリソースベースのポリシーの使用方法の違いについては、「IAM ユーザーガイド」の「IAM でのクロスアカウントのリソースへのアクセス」を参照してください。

Lightsail インスタンスの IPv6 到達可能性を検証する

ping ツールを使用して、ローカルコンピュータから Amazon Lightsail インスタンスへの IPv6 接続を確認できます。Ping は、2 つ以上のネットワークデバイス間の接続問題のトラブルシューティングに使用されるネットワーク診断ユーティリティです。ping が成功した場合、IPv6 経由でインスタンスに接続できるはずです。ネットワーク設定またはデバイスが IPv6 を許可するように設定されていない場合、ping コマンドは失敗します。詳細については、IPv6 専用の考慮事項を参照してください。

内容

- デュアルスタックインスタンスの IPv6 を有効にする
- インスタンスのファイアウォールを設定する
- インスタンスへの到達可能性をテストする

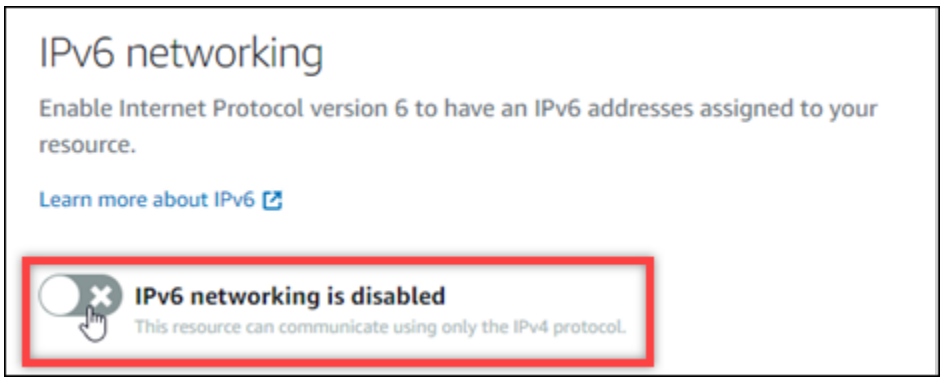
デュアルスタックインスタンスの IPv6 を有効にする

テストを開始する前に、デュアルスタックインスタンスの IPv6 を有効にします。IPv6 は、IPv6 専用インスタンスでは常にオンになっています。

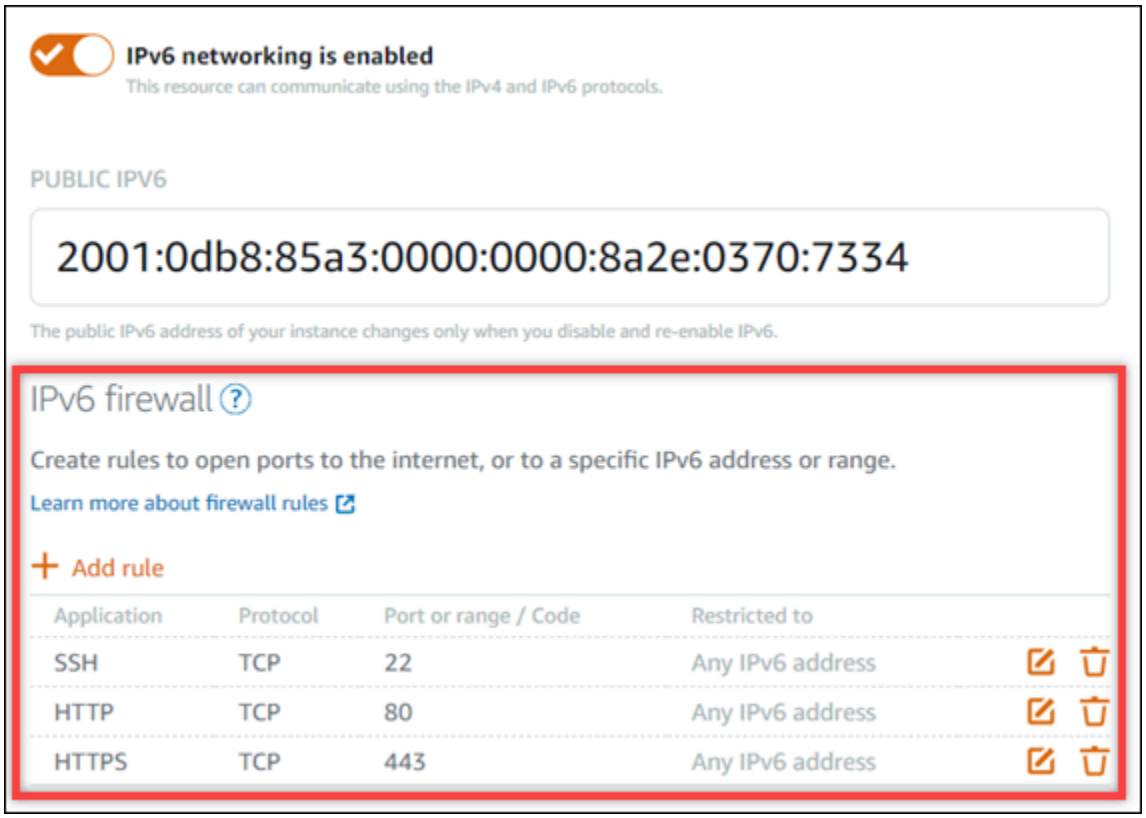
デュアルスタックインスタンスが有効でない場合は、次の手順を実行して IPv6 を有効にします。

1. Lightsail コンソール にサインインします。
2. IPv6 を有効にするインスタンスの名前を選択します。インスタンスが実行されていることを確認します。

3. インスタンスの管理ページで [ネットワーク] タブを選択します。
4. ページの [IPv6 ネットワーキング] セクションで IPv6 を有効にします。



IPv6 を有効にすると、パブリック IPv6 アドレスがインスタンスに割り当てられ、IPv6 ファイアウォールが使用可能になります。



5. ページの上部にあるインスタンスの[パブリック IPv4] アドレスと [パブリック IPv6] アドレスを書き留めます。次のセクションで使用します。

インスタンスのファイアウォールを設定する

Lightsail コンソール内のファイアウォールは、仮想ファイアウォールとして機能します。つまり、パブリック IP アドレスを介してインスタンスに接続できるトラフィックを制御します。Lightsail で作成する各デュアルスタックインスタンスには、独立なファイアウォールがあります。1 つは IPv4 アドレス用、もう 1 つは IPv6 アドレス用です。各ファイアウォールには、インスタンスに着信するトラフィックをフィルタリングする一連のルールが含まれています。各ファイアウォールは、互いに独立しています。IPv4 と IPv6 のファイアウォールルールを別個に設定する必要があります。IPv6 専用インスタンスプランを持つインスタンスには、設定できる IPv4 ファイアウォールがありません。

インターネットコントロールメッセージプロトコル (ICMP) トラフィック用にインスタンスのファイアウォールを設定するには、次の手順を実行します。ping ユーティリティは ICMP プロトコルを使用してインスタンスと通信します。詳細については、「Lightsail のファイアウォールを使用してインスタンストラフィックを制御する」を参照してください。

> **Important**
>
> Windows と Linux には、ping コマンドをブロックできるオペレーティングシステム (OS) レベルのファイアウォールが含まれています。続行する前に、インスタンスの OS ファイアウォールが IPv4 および IPv6 経由の ICMP トラフィックを受け入れることができることを確認します。詳細については、次のドキュメントを参照してください。
>
> - RDP を使用した Lightsail Windows インスタンスへの接続
> - Lightsail で Linux または Unix のインスタンスに接続する

1. Lightsail コンソール にサインインします。
2. ファイアウォールを設定する対象のインスタンスの名前を選択します。
3. インスタンス管理ページから [ネットワーク] タブを選択し、使用するファイアウォールのタイプに適したセクションの残りのステップを完了します。IPv4 の場合は、[IPv4 Firewall] セクションのステップを完了します。IPv6 の場合は、[IPv6 Firewall] セクションのステップを完了します。

   a. [アプリケーション] ドロップダウンメニューから、[Ping (ICMP)] を選択します。

   b. [IP アドレスに制限] ボックスを選択して、ローカルの送信元 IP アドレスまたは範囲からの接続を許可し、送信元 IP アドレスを入力します。(選択可能) 任意の IP アドレスからの接続を許可するには、ボックスを選択解除したままにすることができます。このオプションはテスト環境でのみ使用することをお勧めします。

c. [作成] を選択して、新しいルールをインスタンスに適用します。

インスタンスへの到達可能性をテストする

ローカルコンピュータまたはネットワークから Lightsail インスタンスへの IPv4 または IPv6 の到達可能性をテストするには、次の手順を実行します。Step 5 でメモしたインスタンスのパブリック IPv4 アドレスと IPv6 アドレスが必要です。

Linux、Unix、または macOS デバイスから

1. ローカルデバイスでターミナルウィンドウを開きます。
2. 次のいずれかのコマンドを使用し、Lightsail インスタンスを ping します。コマンド内の *IP ### ###*の例を、インスタンスのパブリック IPv4 または IPv6 アドレスに置き換えます。

   IPv4 経由でテストするには

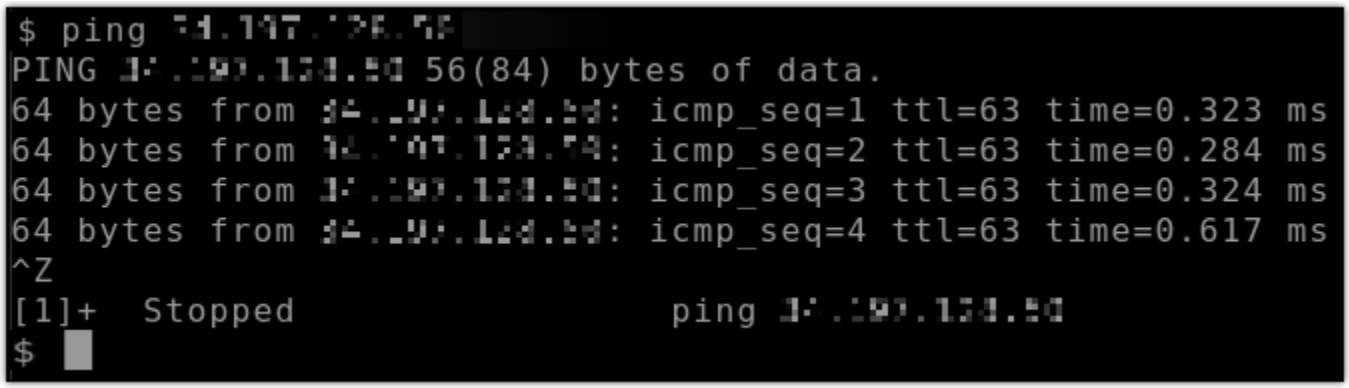
   ```
   ping 192.0.2.0
   ```

   IPv6 経由でテストするには

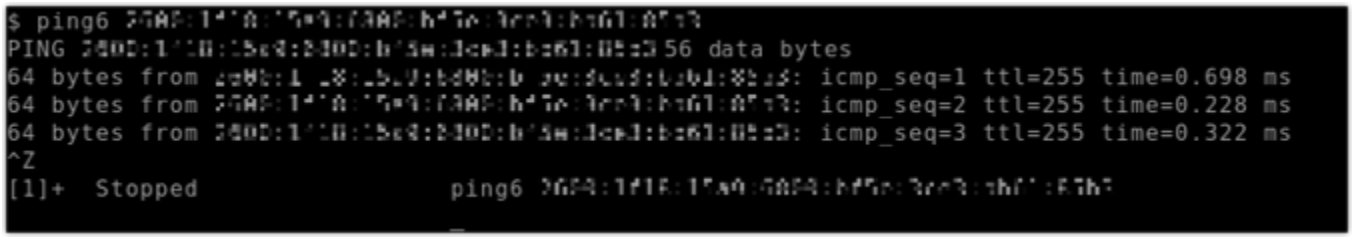
   ```
   ping6 2001:db8::
   ```

3. コマンドがいくつかの返信を返したら、デバイスのキーボードに `ctrl+z` を入力してコマンドを停止します。

ping コマンドは、成功するとインスタンスの IPv4 アドレスから成功した返信を返します。結果は次の例のようになります。



ping6 コマンドが成功すると、インスタンスの IPv6 アドレスから正常な応答が返されます。結果は次の例のようになります。



インスタンスに到達できない場合、どちらのコマンドも [リクエストタイムアウト] を返します。

Windows デバイスから

1. コマンドプロントを開きます。
2. 次のいずれかのコマンドを使用し、Lightsail インスタンスを ping します。コマンド内の *IP ###* の例を、インスタンスのパブリック IPv4 または IPv6 アドレスに置き換えます。

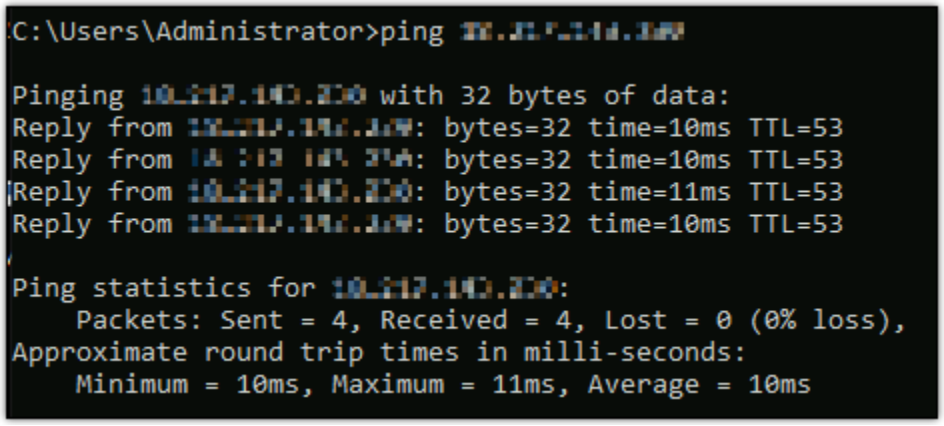
   IPv4 経由でテストするには

   ```
   ping 192.0.2.0
   ```

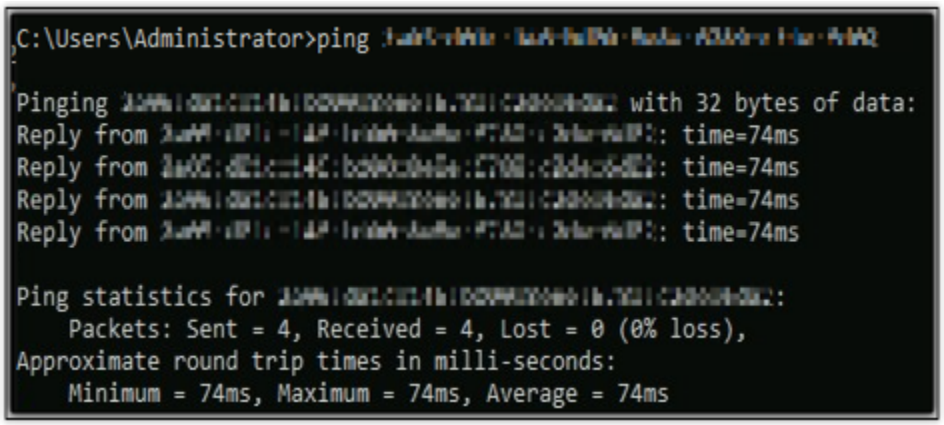
   IPv6 経由でテストするには

   ```
   ping 2001:db8::
   ```
3. コマンドがいくつかの返信を返したら、デバイスのキーボードに `ctrl+z` を入力してコマンドを停止します。

ping コマンドは、成功するとインスタンスの IPv4 アドレスから成功した返信を返します。結果は次の例のようになります。



ping コマンドは、成功するとインスタンスの IPv6 アドレスから成功した返信を返します。結果は次の例のようになります。



インスタンスに到達できない場合、どちらのコマンドも [リクエストタイムアウト] を返します。

Lightsail のインスタンス容量不足エラーの解決

インスタンスを起動するか、停止したインスタンスを再度スタートしようとするとき、不足エラーが発生する場合があります。つまり AWS 、 には、現在リクエストを満たすために使用可能なインスタンス容量がありません。次の内容はインスタンス容量不足エラーの例です。

InsufficientInstanceCapacity: インスタンスリクエストを満たすために十分な容量がありません。リクエスト内のインスタンス数を減らすか、追加の容量が利用可能になるまでお待ちください。より小さな Lightsail プラン 後でサイズ変更ができます) を選択することで、インスタンスを起動することもできます」。

このガイドでは、インスタンス容量不足エラーが発生した場合に実行できるアクションについて説明します。

目次

- 新しいインスタンスを起動するときの容量不足
- 停止したインスタンスをスタートするときの容量不足
- 関連情報

新しいインスタンスを起動するときの容量不足

新しいインスタンスを起動するときにインスタンス容量不足エラーが発生した場合、次のオプションを使用してください。各オプションを順番に入力することも、合ったオプションを選択することもできます。

1. 数分間待ってからリクエストを再度送信してください。インスタンス容量は頻繁に変化します。数分待ってもインスタンスを作成できない場合、オプション 2 に進みます。
2. インスタンスを作成するときは、別のアベイラビリティーゾーン (AZ) を選択します。各 AWS リージョン には 3 つ以上の AZ が含まれ、各 AZ が維持しているインスタンス容量は異なります。別の AZ を選択することにより、現在のインスタンス容量を活用できます。別の AWS リージョン または AZ にインスタンスを作成できない場合は、オプション 3 に進みます。
3. リクエスト内のインスタンスの数を減らします。複数のインスタンスを同時に作成する場合、インスタンスの数を減らしてリクエストを再送信してください。インスタンスの数を減らしても問題が解決しない場合、オプション 4 に進みます。
4. インスタンスを作成するときは別のインスタンスプランを選択してください。別の AZ または リージョンにインスタンスを作成できない場合、別のインスタンスプランを選択してください。インスタンスのサイズ変更は後で行えます。インスタンスのサイズ変更の詳細については、「スナップショットからインスタンスを作成する」を参照してください。

停止したインスタンスをスタートするときの容量不足

以前に停止した既存のインスタンスをスタートしたとき、インスタンス容量不足エラーが発生した場合、次のオプションを使用してください。

1. 数分間待ってからリクエストを再度送信してください。インスタンス容量は頻繁に変化します。数分待ってもインスタンスを作成できない場合、オプション 2 に進みます。
2. スナップショットから新しいインスタンスを作成します。停止したインスタンスのスナップショットを作成します。次に、スナップショットを使用し、元のインスタンスとは異なる新しいインスタンスを AZ に作成します。例えば、インスタンスが現在 us-east-2a (ゾーン A) にある場

合、新しいインスタンスを作成するときに us-east-2c (ゾーン C) を選択します。詳細については、「スナップショットからインスタンスを作成する」を参照してください。

3. スナップショットから新しいインスタンスを作成するとき、別のインスタンスプランを選択することもできます。このアクションはオプションです。

> **Important**
>
> 新しいインスタンスが実行状態になった後、新しいインスタンスにアクセスできてすべてが正常に動作していることを確認します。例えば、インスタンスがアプリケーションを実行していた場合、アプリケーションが期待どおりに動作していることを確認してください。このような場合、以前のインスタンスを削除できます。

関連情報

よくある質問

Lightsail の耐障害性

Lightsail ロードバランサーに関する問題をトラブルシューティングする

Lightsail ロードバランサーでエラーが発生する可能性があります。このトピックでは、一般的な問題とそれらのエラーの回避策について説明します。

ロードバランサーの一般的なエラー

以下の問題の中から発生している問題に最も近いものを選択し、リンク先に移動して問題を解決します。リストにない問題が発生した場合、このページの一番下にある [ご質問は? コメント?] リンクを使用してフィードバックを送信するか、AWS カスタマーサポートにお問い合わせください。

証明書を作成できません。

AWS アカウントに作成できる証明書の数にはクォータがあります。詳細については、AWS Certificate Manager ユーザーガイドの「クォータ」を参照してください。同じクォータがロードバランサーの Lightsail 証明書に適用されます。

実際のエラーメッセージ: Sorry, you've requested too many certificates for your account.

ロードバランサーに追加のインスタンスをアタッチできません。

Lightsail アカウントあたり合計 20 Lightsail インスタンスというクォータを超えない限り、任意の数の AWS インスタンスをロードバランサーにアタッチできます。

実際のエラーメッセージ: Sorry, you've reached the maximum number of instances you can attach to this load balancer.

ロードバランサーに特定のインスタンスをアタッチできません。

まず、Lightsail インスタンスが実行されていることをチェックします。停止している場合は、インスタンス管理ページから開始できます。Lightsail インスタンスは、実行されている場合に限り、ロードバランサーに正常にアタッチされます。

同じインスタンスを多数のロードバランサーにアタッチした可能性があります。

実際のエラーメッセージ: Sorry, you've reached the maximum number of times an instance can be registered with a load balancer.

Lightsailロードバランサーをアタッチしようとしているインスタンスを が見つけることができません

存在しなくなったか、ターゲットグループと同じ VPC 内に存在しないインスタンスをアタッチしようとしている可能性があります。

実際のエラーメッセージ: Sorry, the instance you specified doesn't exist, isn't in the same VPC as the target group, or has an unsupported instance type.

Lightsail での通知配信のトラブルシューティング

通知が予定されているときに通知を受け取らない場合は、通知の連絡先が正しく設定されていることを確認するためにいくつかの点を確認してください。通知の詳細については、「??? の通知」を参照してください 。

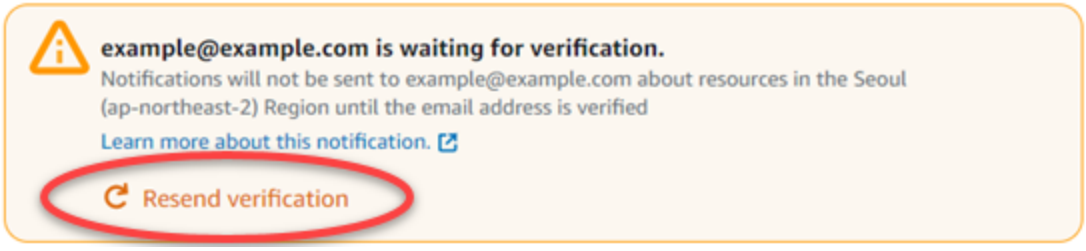
次の一覧では、発生する可能性がある一般的な通知連絡先の問題、原因とその解決方法について説明します。リストにない問題が発生した場合、このページの一番下にある 「ご質問は？ このページ下部の [フィードバック] リンクにアクセスしてフィードバックを送信するか、「AWS サポート センター」にお問い合わせください。

メールアドレスを通知連絡先として追加しましたが、メール通知が届きません

Lightsail でメールアドレスを通知連絡先として追加すると、そのアドレスに確認要求が送信されます。確認要求の メールには、受信者が Lightsail 通知を受信することを確認するためにクリック

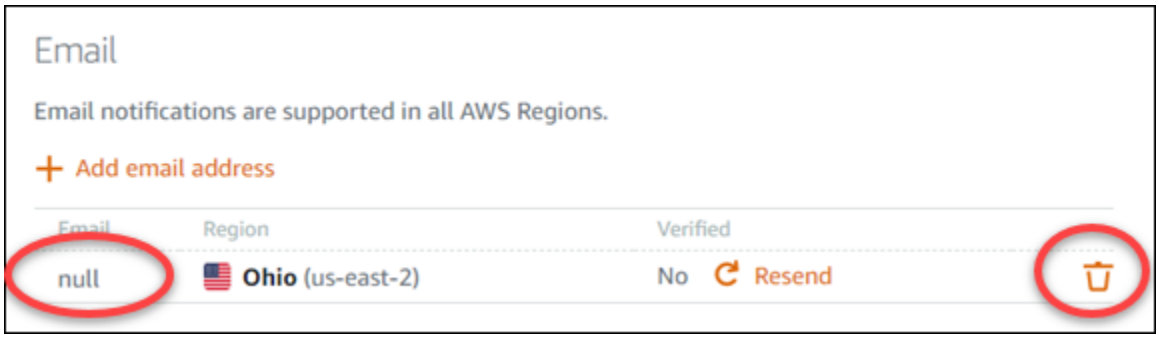
する必要があるリンクが含まれています。通知は、確認が完了するまでメールアドレスに送信されません。検証は、AWS 通知 <no-reply@sns.amazonaws.com> から行われ、件名は AWS 通知 - サブスクリプションの確認です。SMS メッセージングには検証は不要です。

確認要求が受信トレイフォルダにない場合は、メールボックスのスパムフォルダと迷惑メールフォルダを確認してください。検証リクエストが失われた、または削除された場合は、Lightsail コンソールやアカウントページに表示される通知バナーで検証の再送を選択してください。



メール通知の連絡先として null が表示されています。

メールアドレスは、追加後 24 時間以内に確認する必要があります。24 時間以内に メールの確認に失敗した場合、その メールは自動的に `invalid` のステータスになり、Lightsail から削除されます。そのため、1 つ以上の メール通知連絡先に null の値が表示されることがあります。



この問題を解決するには、null メール通知の連絡先を削除し、正しいメールアドレスを再度追加します。メールアドレスは、Lightsail に追加した直後に確認してください。詳細については、「通知」を参照してください。

SMS テキストメッセージの通知が届かない、または最近受信が停止した

SMS テキストメッセージ通知の受信をオプトアウトしている可能性があります。ARRET (フランス語)、CANCEL、END、OPT-OUT、OPTOUT、QUIT、REMOVE、STOP、TD、UNSUBSCRIBE を使用して SMS テキストメッセージ通知に応答することでオプトアウトできます。携帯電話番号をオプトアウトする場合、その携帯電話番号を Lightsail で通知連絡先として再度追加できるようになるまで 30 日待つ必要があります 。

Lightsail の SSL/TLS 証明書のトラブルシューティング

Lightsail ロードバランサーでエラーが発生する可能性があります。このトピックでは、一般的な問題とそれらのエラーの回避策について説明します。

以下の問題の中から発生している問題に最も近いものを選択し、リンク先に移動して問題を解決します。リストにない問題が発生した場合、このページの一番下にある [ご質問は? コメント?] リンクを使用してフィードバックを送信するか、AWS カスタマーサポートにお問い合わせください。

証明書を作成できません。

AWS アカウントで作成できる証明書の数にはクォータがあります。詳細については、AWS Certificate Manager ユーザーガイドの「クォータ」を参照してください。同じクォータがロードバランサーの Lightsail 証明書に適用されます。

実際のエラーメッセージ:Sorry, you've requested too many certificates for your account.

証明書リクエストに失敗しました。

証明書リクエストに失敗した場合、ロードバランサー管理ページの [インバウンドトラフィック] タブで [再試行] を実行できます。

原因がまだわからない場合は、AWS カスタマーサポートまでご連絡ください。

ドメインが無効と表示されました。

ドメインを管理していることを確認できない場合、DNS 管理にアクセスできることを確認します。アクセスできる場合は、こちらの手順に従います。この手順でも検証できない場合は、AWS カスタマーサポートにご連絡ください。

チュートリアルで Lightsail 機能を探索する

このセクションでは、次の Amazon Lightsail に関するトピックについて説明します:

トピック

- Lightsail ブループリントを使用してアプリケーションをすばやくデプロイする
- Lightsail での Bitnami アプリケーションとスタックの操作
- WordPress Lightsail インスタンスの設定と管理
- Lightsail で マルチサイトを使用して複数の WordPress サイトを管理する
- Let's Encrypt で Lightsail リソースの暗号化された通信を有効にする
- Lightsail インスタンスの IPv6 ネットワークを設定する
- Lightsail オペレーション AWS CLI 用の をセットアップおよび設定する
- でLightsailリソースを管理する AWS CloudShell
- Lightsail LAMP インスタンスに PHP アプリケーションをデプロイする
- Lightsail の Windows Server 2016 インスタンスを起動して設定する
- で Lightsail API アクティビティをモニタリングする AWS CloudTrail
- HAR ファイルを作成して Lightsail 問題をトラブルシューティングする
- Lightsail で Prometheus を使用してシステムリソースとアプリケーションをモニタリングする
- scp を使用して Lightsail の Linux インスタンス間でファイルを転送する
- VPC ピアリングを使用して他の Lightsail AWS サービスと統合する
- を使用してLightsailリソースを作成する AWS CloudFormation
- アプリケーションのデプロイのために Lightsail リソースを探索する

各カテゴリで提供されているリンクに従って、Lightsail の操作に関するさまざまな側面に関するステップバイステップガイド、ベストプラクティス、追加情報にアクセスします。

各トピックでは、アプリケーションのデプロイ、ネットワークの設定、モニタリングとログ記録、他の AWS サービスとの統合などの情報について説明します。このセクションでは、 を効果的に活用しLightsail、他の AWS サービスとの統合を活用し、豊富なチュートリアルやリソースにアクセスしてクラウドコンピューティングエクスペリエンスを向上させる方法について説明します。

Lightsail ブループリントを使用してアプリケーションをすばやくデプロイする

次のクイックスタートガイドを使用して、Lightsail ブループリントを始めましょう。Lightsail では、ブループリントはオペレーティングシステムとアプリケーションにあらかじめパッケージ化された仮想イメージです。アプリケーションには、WordPress、WordPress マルチサイト、cPanel & WHM、PrestaShop、Drupal、Ghost、Joomla!、Magento、Redmine、Ruby on Rails、LAMP、Nginx (LEMP)、Node.js などがあります。

トピック

- Lightsail で AlmaLinux インスタンスを起動および設定する
- Lightsail で cPanel と WHM を使用してウェブサイト、メール、サービスをホストします。
- Lightsail で Drupal ウェブサイトを設定およびカスタマイズする
- Lightsail で Ghost ウェブサイトをデプロイする
- Lightsail で GitLab CE インスタンスを設定および構成する
- Lightsail で Joomla! の使用を開始する
- での LAMP スタックのデプロイと管理 Lightsail
- Lightsail で Magento を設定および構成する
- Lightsail で Nginx ウェブサーバーをデプロイして管理する
- での Node.js スタックのデプロイと管理 Lightsail
- で OpenClaw の使用を開始する Lightsail
- Lightsail に Plesk ホストスタックをデプロイする。
- Lightsail で PrestaShop ウェブサイトを設定する
- Lightsail の Redmine インスタンスを設定して保護する
- で Rails に Ruby を設定する Lightsail
- での WordPress のデプロイと管理 Lightsail
- Lightsail で WordPress マルチサイトを設定する

Lightsail で AlmaLinux インスタンスを起動および設定する

> 知っていましたか?
>
> Lightsail は毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを最新のスナップショットに自動的に置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

このクイックスタートガイドでは、Amazon Lightsail プラットフォームで AlmaLinux インスタンスを作成および設定するためのステップバイステップの手順を説明します。このトピックでは、インスタンスの場所と計画の選択、ネットワークとセキュリティの設定、CentOS から AlmaLinux への移行など、主要なステップについて説明します。これらのステップに従うことで、AlmaLinux インスタンスを Lightsail ですばやく起動して実行できます。

トピック

- 前提条件
- Lightsail で AlmaLinux インスタンスを作成する
- (選択可能) 追加設定
- Lightsail で CentOS から AlmaLinux にデータを移行する

前提条件

- 新規 AWS のお客様の場合は、 の使用を開始する前にセットアップの前提条件を完了してくださいAmazon Lightsail。詳細については、「の設定 AWS アカウント と管理ユーザー Lightsail」を参照してください。
- 「AlmaLinux Wiki」サイトの AlmaLinux ドキュメントをお読みください。

Lightsail で AlmaLinux インスタンスを作成する

「Lightsail コンソール」を使用して AlmaLinux インスタンスを作成するには、次の手順を実行します。

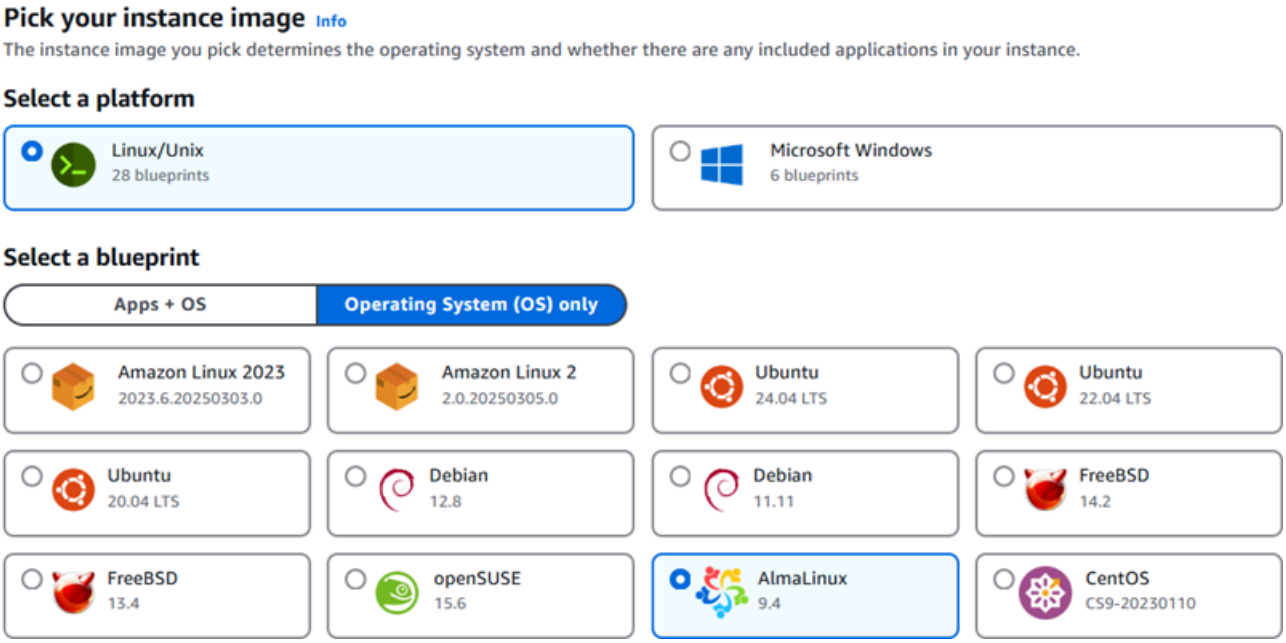
1. Lightsail コンソール にサインインします。
2. ホームページで [インスタンスの作成] を選択します。

3. インスタンスの場所 (AWS リージョン とアベイラビリティーゾーン) を選択します。レイテンシーを短縮するために、物理的な場所に最も AWS リージョン 近い を選択します。

   別の場所でインスタンスを作成するには、[アベイラビリティーゾーンの変更] を選択します。

4. Linux プラットフォームを選択します。

5. [オペレーティングシステム (OS) のみ] を選択し、[AlmaLinux] ブループリントを選択します。



6. 必要に応じて、次の操作も行えます:

   a. インスタンスの初回起動時に実行されるシェルスクリプトを追加するには、[起動スクリプトを追加] を選択します。詳細については、「Lightsail で起動スクリプトを使用して Linux/Unix インスタンスを設定する」を参照してください。

   b. インスタンスの SSH キーペアを変更するには、[SSH キー] の下にあるドロップダウンリストからキーを選択します。詳細については、「Lightsail の SSH キーの設定」を参照してください。

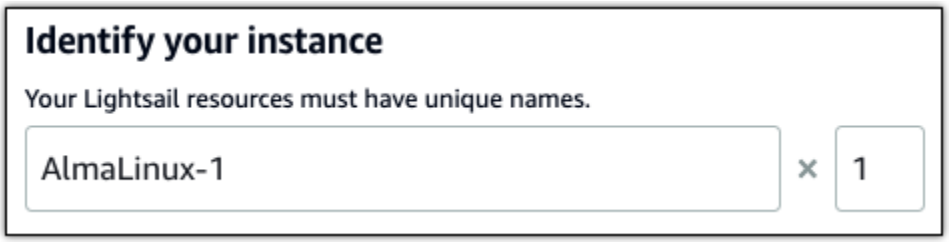
   c. [自動スナップショットを有効にする] を選択して、インスタンスとアタッチされたディスクの [自動スナップショット] を有効にします。詳細については、「Lightsail インスタンスとディスクの自動スナップショットを設定する」を参照してください。

7. インスタンスプランを選択します。インスタンスがデュアルスタック (IPv4 と IPv6) を使用するか IPv6 専用のネットワークを使用するかを選択できます。AlmaLinux ブループリントは、デュアルスタックバンドルと IPv6 専用バンドルの両方をサポートしています。IPv6 専用ネットワークの詳細については、Lightsail インスタンスの IPv6 専用ネットワークを設定する を参照してください。

8. インスタンスの名前を入力します。

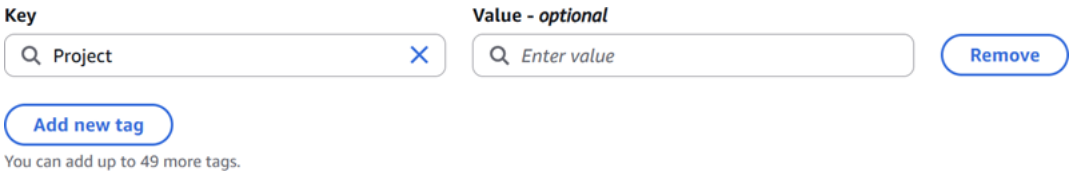
   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

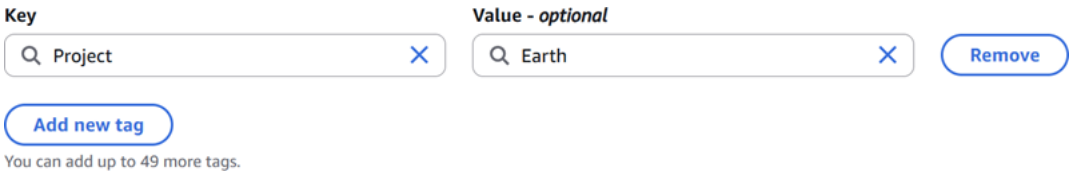


9. (オプション)[新しいタグの追加] を選択してインスタンスにタグを追加します。必要に応じてこの手順を繰り返して、他のタグを追加します。タグの使用の詳細については、「タグ」を参照してください。

   a. [キー] には、タグキーを入力します。

   b. (オプション) [値] にタグの値を入力します。

10. [インスタンスの作成] を選択します。

数分で Lightsail インスタンスの準備が整い、そのインスタンスに接続できます。

(選択可能) 追加設定

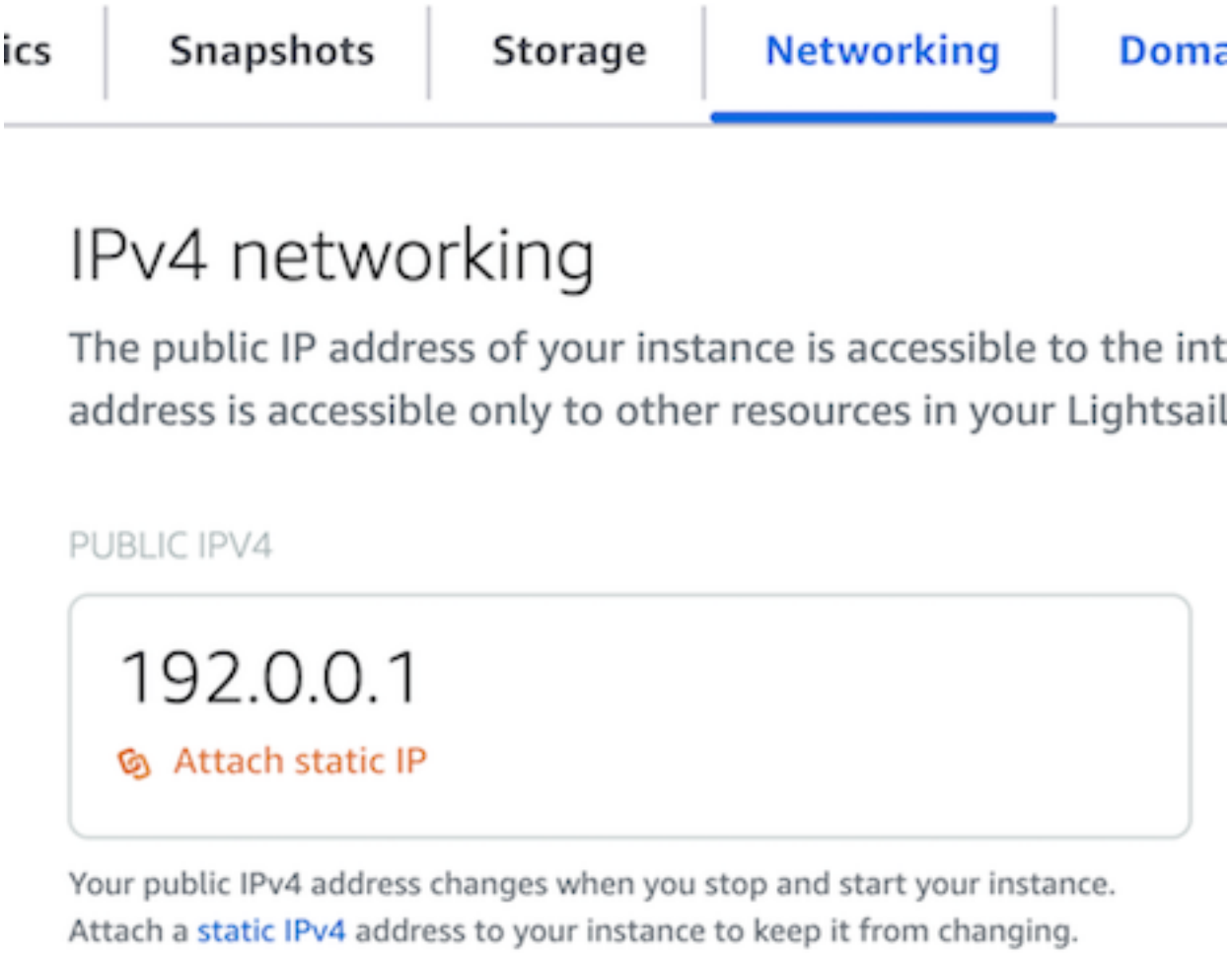
Lightsail で起動した AlmaLinux インスタンスの使用を開始するステップについて説明します:

- インスタンスに静的 IP アドレスをアタッチ – インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。パブリック IP アドレスが変わらないようにするには、静的 IP アドレスを作成してインスタンスにアタッチします。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。1 つの静的 IP を 1 つのインスタンスにアタッチできます。

  インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブ

リック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



- Lightsail でドメインの登録 Lightsail でドメイン名を登録および管理します。Lightsail は Amazon Route 53 という可用性た高くてスケーリング可能なドメインネームシステム (DNS) のウェブサービスを使用し、ドメインを登録します。ドメインが登録されたら、Lightsail リソースに割り当てたり、DNS レコードを管理したりできます。詳細については、「Lightsail でウェブサイトのドメインを登録および管理します。」を参照してください。
- インスタンスにドメイン名をマッピングする – ドメイン名 (`example.com` など) をインスタンスにマッピングするには、ドメインのドメインネームシステム (DNS) にレコードを追加しま

す。DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページの [ドメインと DNS] セクションで、[DNS ゾーンの作成] を選択し、ページに記載される手順に従います。詳細については、「Lightsail インスタンスのドメインレコードを管理する DNS ゾーンを作成する」を参照してください。

- インスタンスのスナップショットを作成する – スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、メモリ、CPU、ディスクサイズ、データ転送レートなどの情報が含まれています。スナップショットを、新しいインスタンスのベースラインまたはデータのバックアップとして使用できます。

  インスタンス管理ページの [スナップショット] タブで、スナップショット名を入力して [スナップショットの作成] を選択します。詳細については、「スナップショットを使用して Linux/Unix Lightsail インスタンスをバックアップする」を参照してください。

CentOS から AlmaLinux に移行する方法については、次のトピック Lightsail で CentOS から AlmaLinux にデータを移行する に進みます:

Lightsail で CentOS から AlmaLinux にデータを移行する

CentOS から AlmaLinux への移行は簡単なプロセスで、Lightsail のインスタンス間でデータを移動することで実現します。このトピックでは、データの移行に使用できる 2 つのオプションについて概説します。

詳細については、「AlmaLinux Wiki サイト」にある AlmaLinux ドキュメントを参照してください。

目次

- 前提条件
- (選択可能) セキュアコピー (scp) を使用してインスタンスの間でファイルを転送します。
- (選択可能) ブロックストレージディスクを CentOS インスタンスから AlmaLinux インスタンスに移動します。

前提条件

- まだ作成していない場合は、AlmaLinux Lightsail インスタンスを作成します。詳細については、「Lightsail で AlmaLinux インスタンスを起動および設定する」を参照してください。

- AlmaLinux インスタンスに移動するディスクのスナップショットを作成します。詳細については、「バックアップまたはベースライン用の Lightsail ブロックストレージディスクスナップショットを作成します」を参照してください。

(選択可能) セキュアコピー (scp) を使用してインスタンスの間でファイルを転送します。

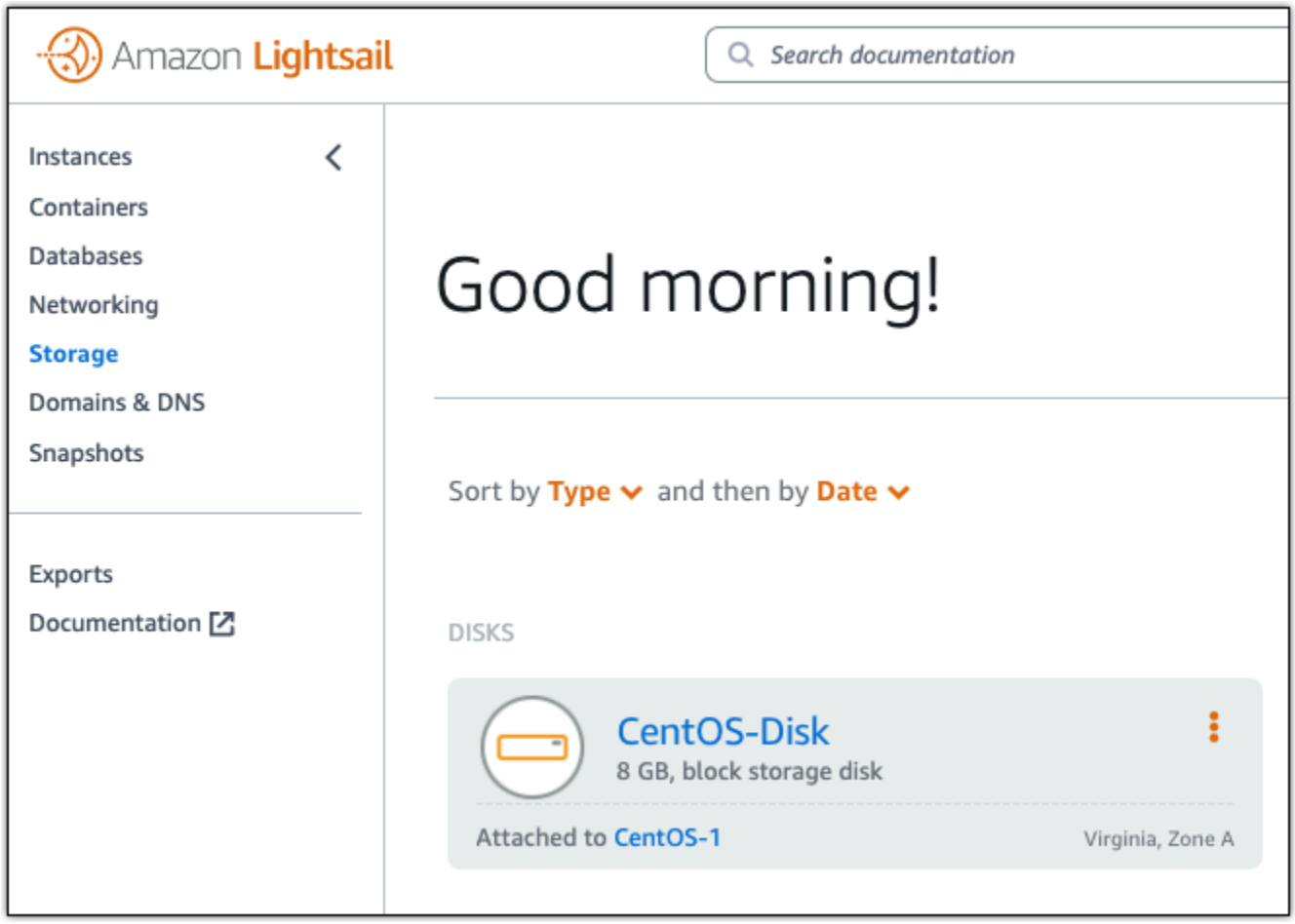
Linux のセキュアコピーコマンドを使用して、CentOS インスタンスから新しい AlmaLinux インスタンスにファイルを安全に転送できます。詳細については、「scp を使用して Lightsail の Linux インスタンス間でファイルを転送する」を参照してください。

(選択可能) ブロックストレージディスクを CentOS インスタンスから AlmaLinux インスタンスに移動します。

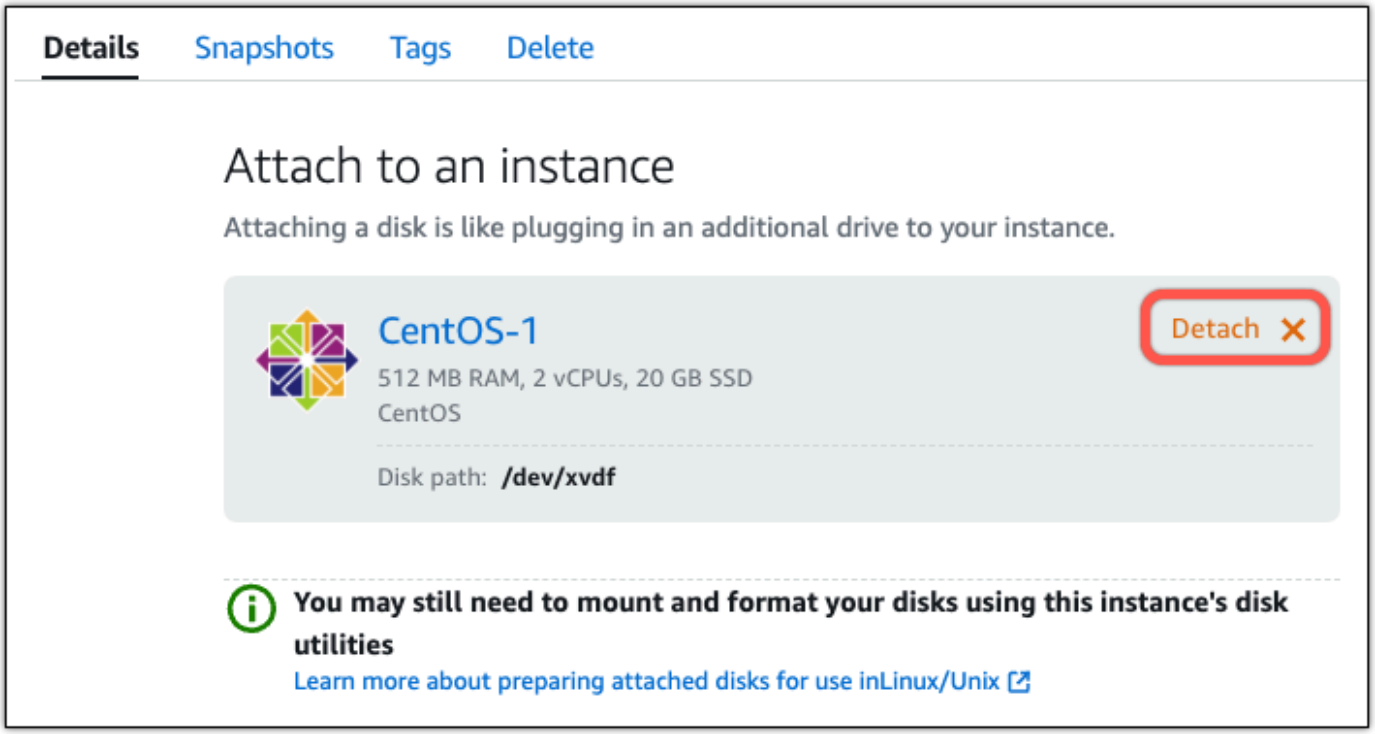
セカンダリブロックストレージディスクを CentOS インスタンスバンドルから AlmaLinux バンドルに移動するには、次の手順に従います。インスタンスのブートボリュームディスク (オペレーティングシステムを含むディスク) は切り離すことができません。AlmaLinux インスタンスにディスクをアタッチしたら、そのインスタンスに接続してディスクをマウントする必要があります。詳細については、「Lightsail ブロックストレージディスクでストレージとパフォーマンスを拡張する」を参照してください。

CentOS インスタンスが実行されている場合は、ディスクをデタッチする前に停止する必要があります。詳細については、「インスタンス実行の停止」を参照してください。

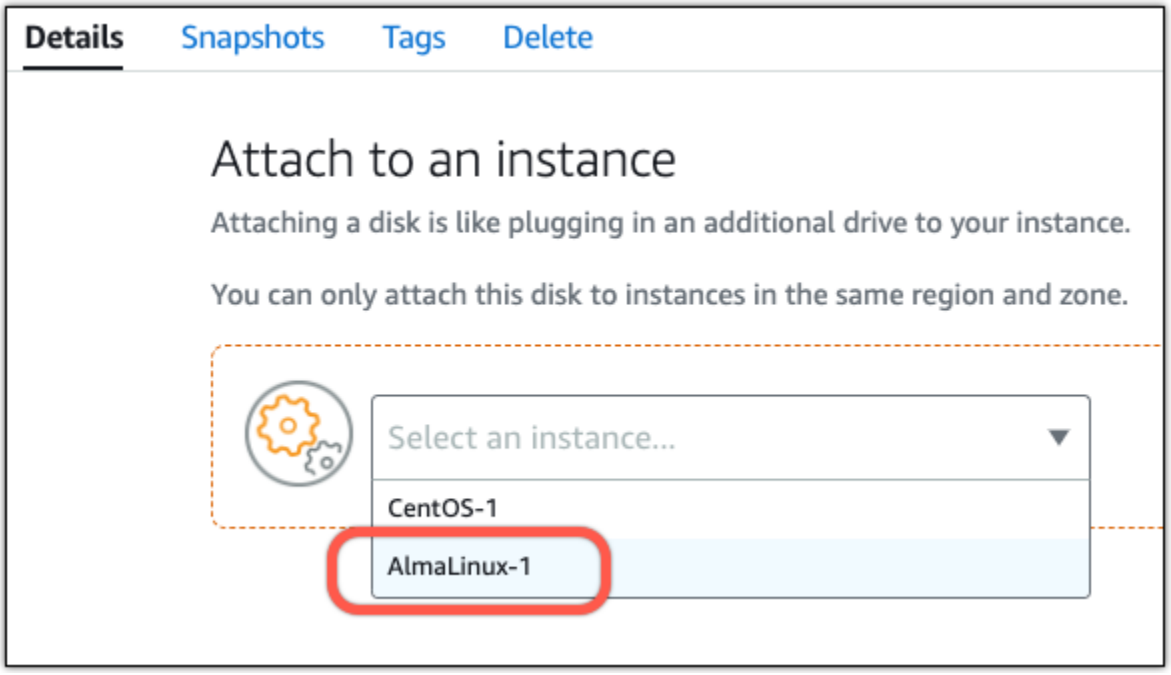
1. Lightsail コンソールの [ストレージ] セクションから、CentOS インスタンスからデタッチするディスクを選択します。



2. [詳細] タブで、[デタッチ] を選択します。



3. ディスクの [詳細] ページから、[インスタンスにアタッチ] ドロップダウンメニューを選択します。次に、AlmaLinux インスタンスの名前を選択します。



4. [アタッチ] を選択してください。
5. (選択可能) データにアクセスする前に、AlmaLinux インスタンスに接続し、ディスクをマウントする必要となる場合があります。詳細については、「インスタンスに接続してディスクをフォーマットおよびマウントする」を参照してください。

> **Warning**
>
> 上記のリンクは、アタッチされたディスクをマウントしてフォーマットする方法を示しています。AlmaLinux インスタンスにアタッチしたディスクをフォーマットしないでください。フォーマットすると、ディスクに保存されているすべての情報が完全に消去されます。

Lightsail で cPanel と WHM を使用してウェブサイト、メール、サービスをホストします。

> **知っていましたか?**
>
> Lightsail は、毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを自動的に最新のスナップショットに置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

Amazon Lightsail で起動した cPanel & WHM インスタンスの使用を開始するステップについて説明します。

> **Important**
>
> - cPanel & WHM インスタンスには、15日間のトライアルライセンスが含まれています。15日後も継続して cPanel & WHM を使用するには cPanel からライセンスを購入する必要があります。ライセンスの購入をご検討の際は、ライセンスを購入する前にこのガイドのステップ 1~7 を完了してください。
> - このブループリントを使用するには、2 GB 以上のメモリを持つインスタンスプランを選択する必要があります。

目次

- ステップ 1: ルートユーザーパスワードの変更
- ステップ 2: cPanel & WHM インスタンスに静的 IP アドレスをアタッチする
- ステップ 3: ウェブホストマネージャーに初めてサインインする

- ステップ 4: cPanel & WHM インスタンスのホスト名と IPアドレスを変更する
- ステップ 5: cPanel & WHM インスタンスにドメイン名をマッピングする
- ステップ 6: インスタンスのファイアウォールを編集する
- ステップ 7: Lightsail インスタンスから SMTP 制限の削除
- ステップ 8: cPanel および WHM のドキュメントを読み込んでサポートを受ける
- ステップ 9: cPanel および WHM のライセンスの購入
- ステップ 10: cPanel および WHM のインスタンスのスナップショットを作成する

ステップ 1: ルートユーザーパスワードの変更

cPanel インスタンスのルートユーザーのパスワードを変更するには、次の手順を実行します。ルートユーザーとパスワードは、ウェブホストマネージャー (WHM) コンソールに後ほどサインインする際に使用します。

1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。
2. 接続後、次のコマンドを入力してルートユーザーのパスワードを変更します。

```
sudo passwd
```

3. 強力なパスワードを入力し、もう一度入力してパスワードを確認します。

> **Note**
>
> パスワードは 7 文字以上で、一般的な単語は含まない必要があります。このガイドラインに従わない場合、BAD PASSWORD の警告が表示されます。

このパスワードは、このガイドの後半で WHM コンソールにサインインする際に使用するので、覚えておいてください。

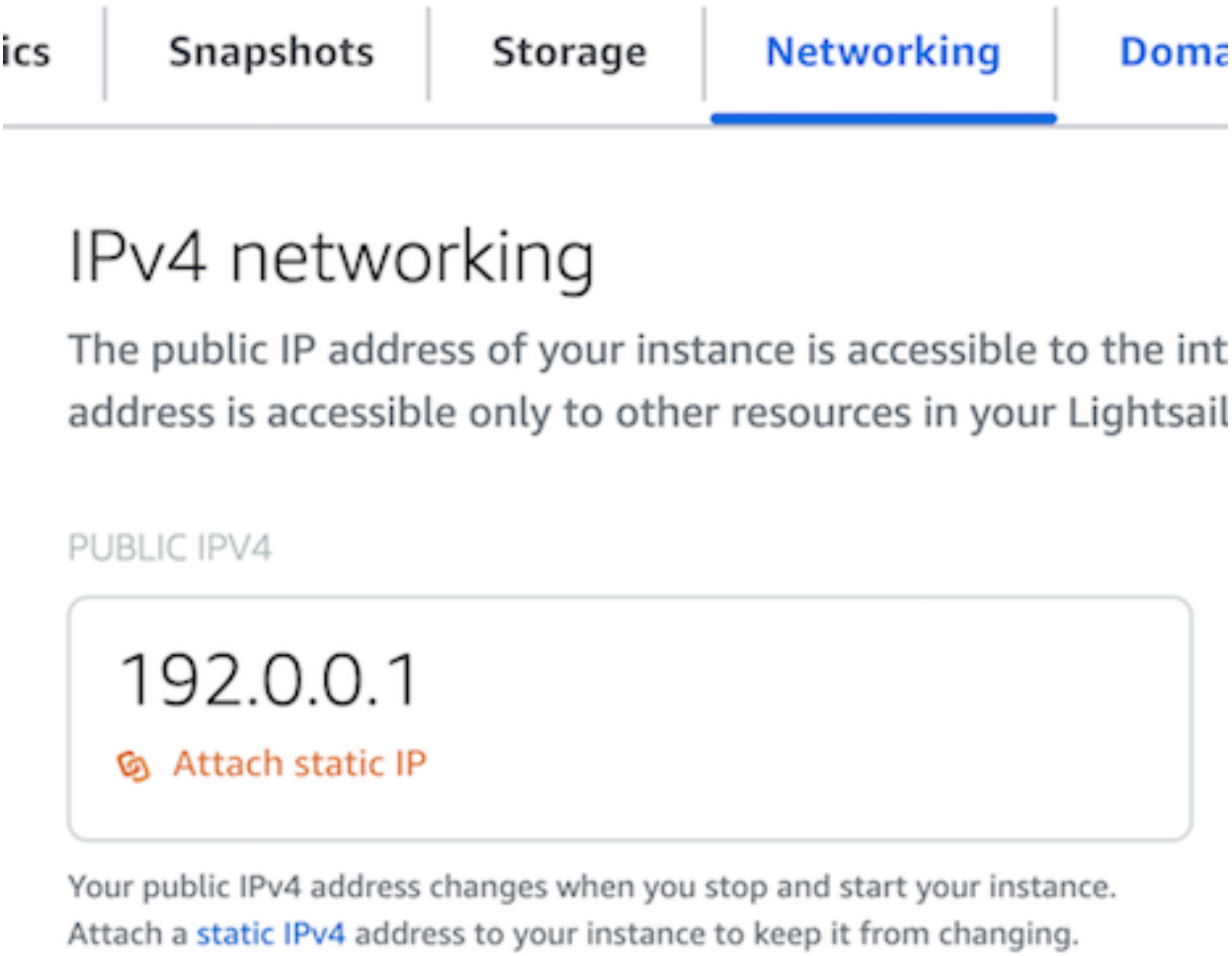
ステップ 2: cPanel & WHM インスタンスに静的 IP アドレスをアタッチする

> **Important**
>
> cPanel からライセンスを購入する際は、cPanel & WHM インスタンスのパブリック IP アドレスを指定する必要があります。購入したライセンスは、その IP アドレスに関連付けられま

す。このため、cPanel からのライセンス購入をご検討される場合は、cPanel & WHM インスタンスに静的 IP をアタッチする必要があります。cPanel からライセンスを購入する際は、静的 IP を指定し、Lightsail インスタンスで cPanel & WHM を使用する限りその静的 IP を保持し続けます。後に別の IP アドレスにライセンスを移転する必要がある場合は、cPanel にリクエストを送信することができます。詳細については、WHM ドキュメント の「ライセンスの移転」を参照してください。

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



ステップ 3: ウェブホストマネージャーに初めてサインインする

WHM コンソールに初めてサインインするには、次の手順を使用します。

1. ウェブブラウザを開き、次のウェブアドレスに移動します。*<StaticIP>* をインスタンスの静的 IP アドレスに置き換えます。アドレスの末尾に :2087 を必ず追加してください。これがインスタンスへの接続を確立するためのポートになります。

   ```
   https://<StaticIP>:2087
   ```

   例:

   ```
   https://192.0.2.0:2087
   ```

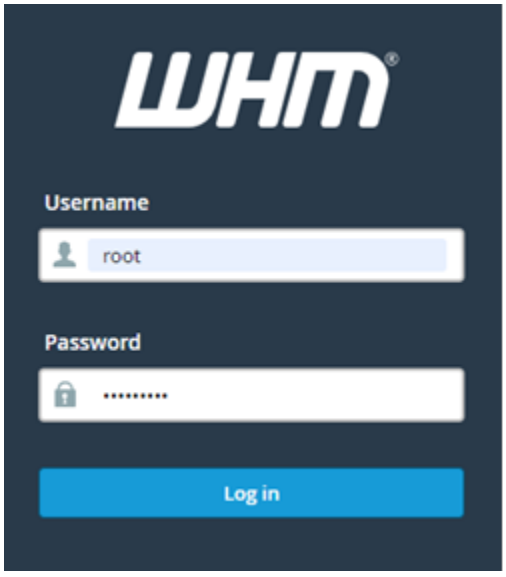
> **Important**
>
> インスタンスの IP アドレスとポートに移動する際は、ブラウザのアドレスバーに `https://` を含める必要があります。そうしないと、サイトにアクセスできないというエラーが表示されます。

ポート 2087 経由でインスタンスの静的 IP アドレスにブラウジングする際に、接続を確立できなかった場合は、ルーター、VPN、またはインターネットサービスプロバイダーがポート 2087 経由の HTTP/HTTPS 接続を許可しているかを確認してください。許可していない場合は、別のネットワークを使用して接続を試みてください。

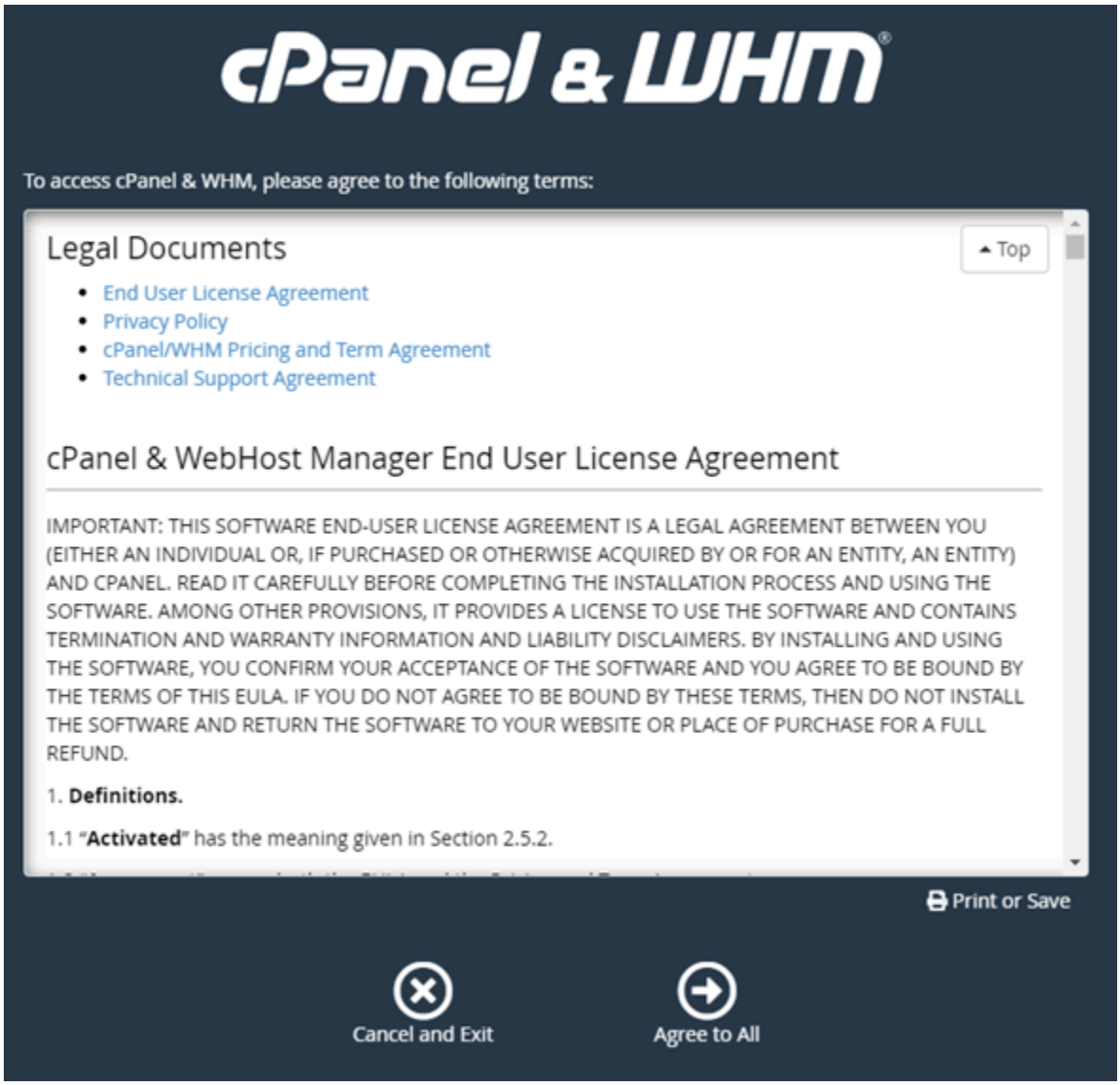
接続がプライベートではない、セキュリティで保護されていない、またはセキュリティ上のリスクがある、などの警告がブラウザに表示されることがあります。これは、 SSL/TLS 証明書がまだ cPanel インスタンスに適用されていない場合に発生します。ブラウザウィンドウで、[Advanced] (詳細設定)、[Details] (詳細)、または [More information] (詳細情報) を選択して、使用可能なオプションを表示します。次に、プライベートまたは安全でない場合でも、ウェブサイトにアクセスすることを選択します。

2. [ユーザーネーム] テキストボックスに `root` を入力します。
3. ルートユーザーパスワードを [パスワード] テキストボックスに入力します。

   これは、このガイドの「ステップ 1: ルートユーザーパスワードを変更する」セクションで先ほど指定したパスワードになります。
4. [ログイン] を選択します。

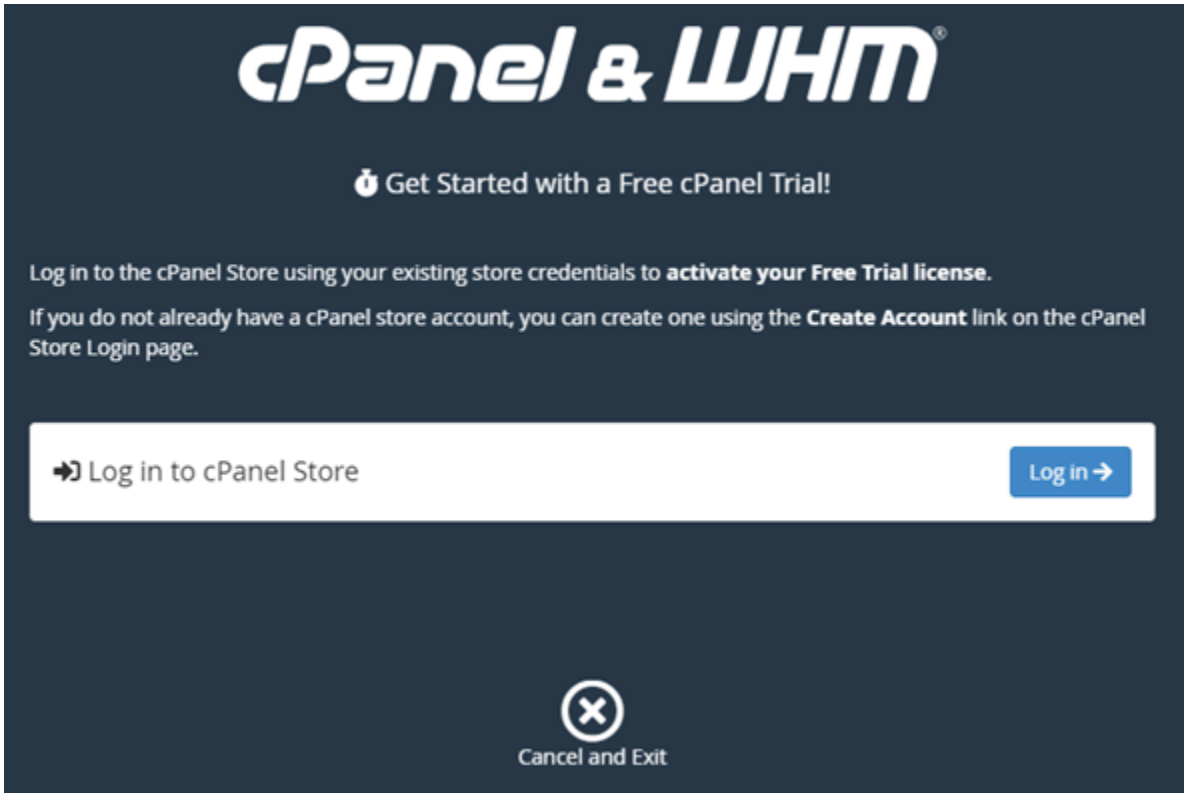


5. 続行する場合は、cPanel & WHM 規約を読んで、「すべてに同意する」を選択します。



6. cPanel の無料トライアルを開始するページで [ログイン] を選択して cPanel ストアにログインします。

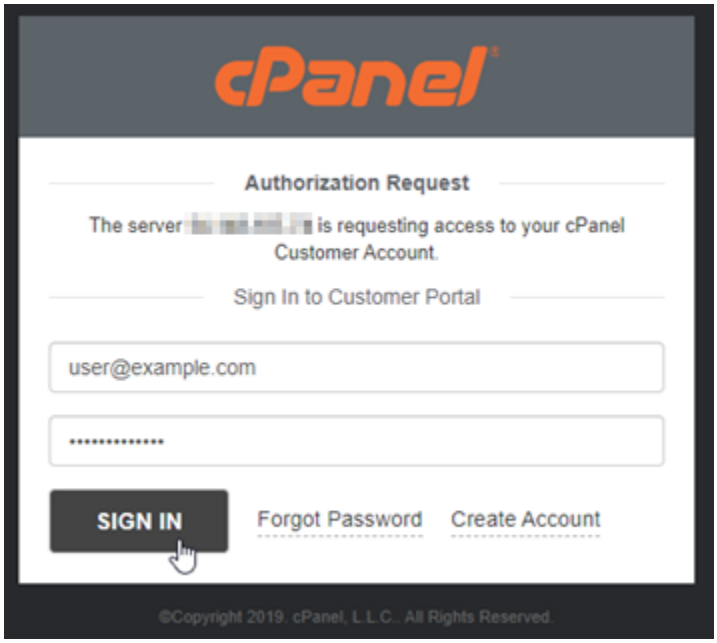
   ユーザーのアカウントにトライアルライセンスを関連付けるには、cPanel ストアにサインインする必要があります。cPanelストアのアカウントをお持ちでない場合も ログイン を選択します。アカウント作成のオプションが表示されます。



7. [認可リクエスト] ページが表示されたら、cPanel ストアのアカウント用のメールアドレスまたはユーザーネーム、およびパスワードを入力します。

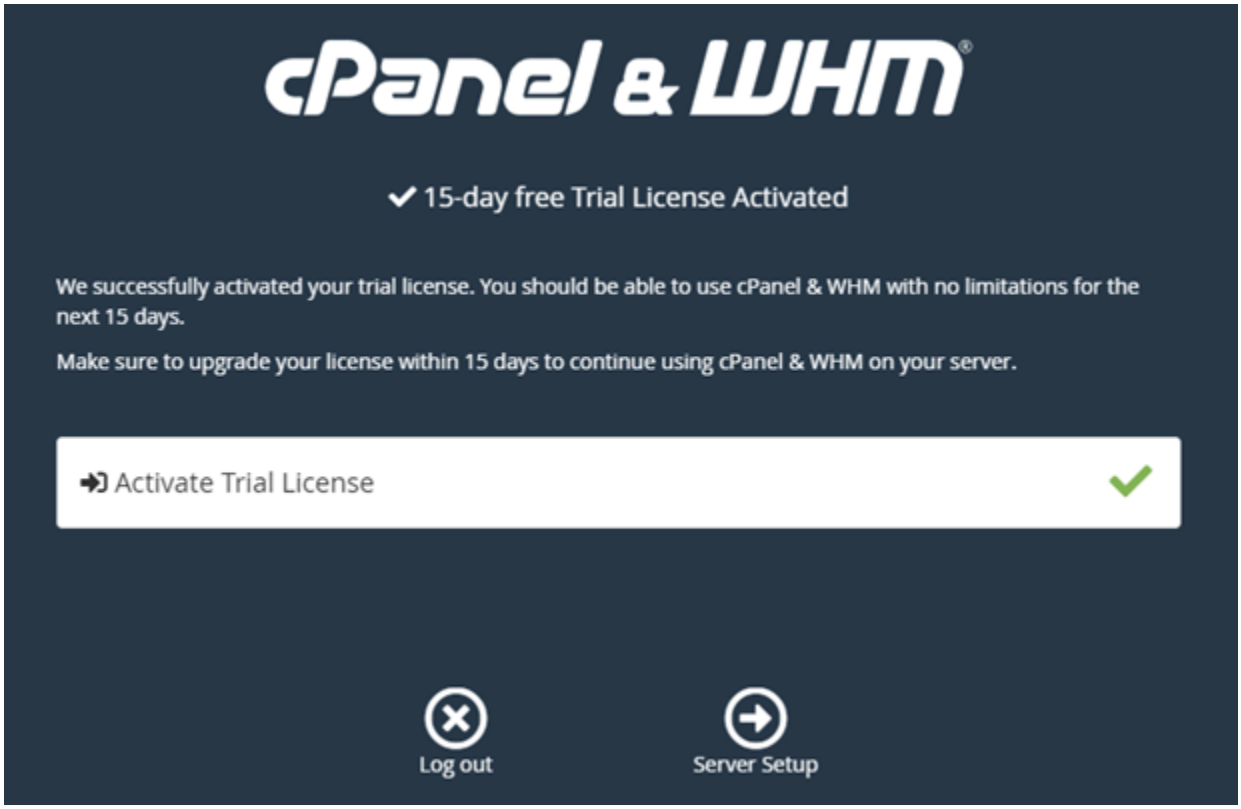
   cPanel ストアのアカウントをお持ちでない場合は、[アカウントの作成] を選択し、プロンプトに従って新しい cPanel ストアのアカウントを作成します。メールアドレスを入力するように求められます。cPanelストアアカウントのパスワードを設定するためのメールが送信されます。新しいブラウザを使用して cPanel ストアアカウントのパスワード設定を行うことをお勧めします。パスワードが設定されたらタブを閉じて、インスタンスに戻ってアカウント認可を行い、この手順の次のステップに進みます。

8. [サインイン] を選択します。

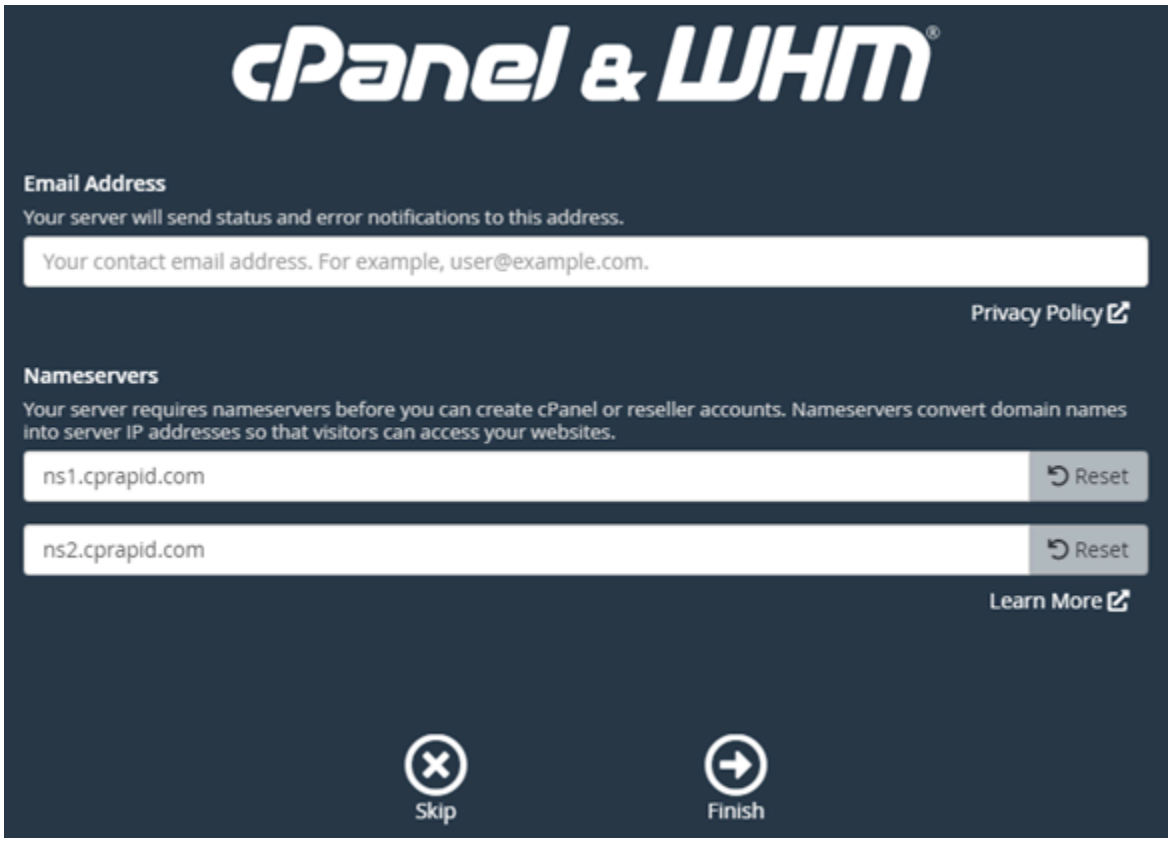


サインイン後、cPanel & WHM インスタンスは 15 日間のトライアルライセンスを取得します。これはユーザーの cPanel ストアのアカウントに関連付けられています。cPanel ストアの [ライセンスの管理] ページに移動して、トライアルライセンスを含む発行されたライセンスを確認します。

9. [サーバーのセットアップ] を選択して続行します。



10. メールアドレスとネームサーバーページの [スキップ] を選択します。これは後から設定することが可能です。

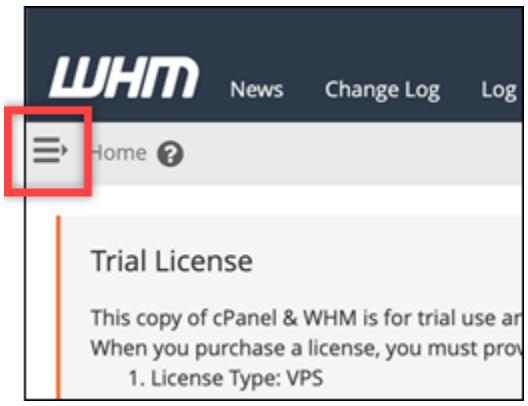


cPanel の設定と機能を管理することができる WHM コンソールが表示されます。

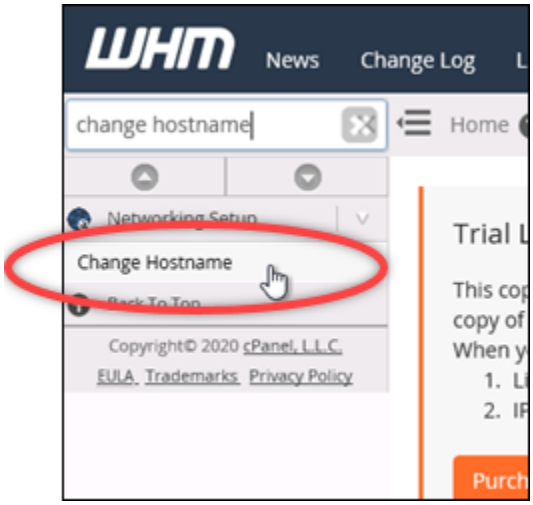
ステップ4：cPanel＆WHM インスタンスのホスト名と IPアドレスを変更する

インスタンスのホスト名を変更して、パブリック IP アドレスを使用しなくても WHM コンソールにアクセスできるようにするには、次の手順を実行します。このガイドの「ステップ 2: cPanel & WHM インスタンスに静的 IP アドレスをアタッチする」のセクションでインスタンスにアタッチした新しい静的 IP アドレスに、インスタンスの IP アドレスを変更する必要があります。

1. WHM コンソールの左上のセクションにある、ナビゲーションメニューアイコンを選択します。

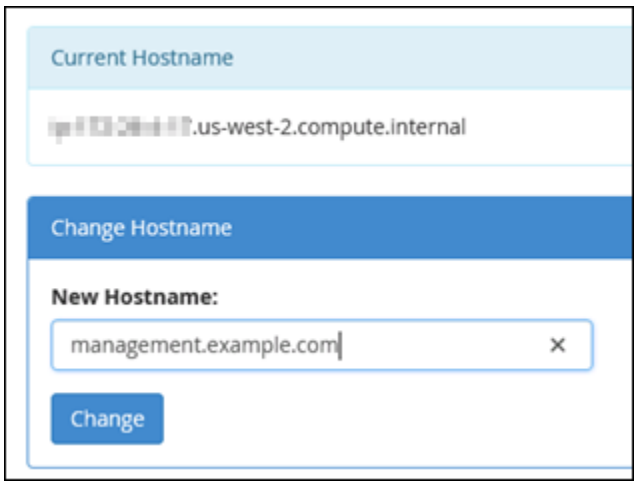


2. WHM コンソールの検索テキストボックスに Change hostname を入力して、検索結果の [ホスト名を変更] のオプションを選択します。

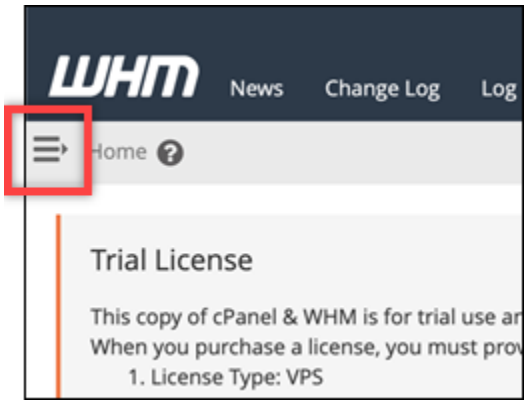


3. WHM コンソールへのアクセスに使用したいホスト名を、[新しいホスト名] テキストボックスに入力します。たとえば、management.example.com ないし administration.example.com を入力します。

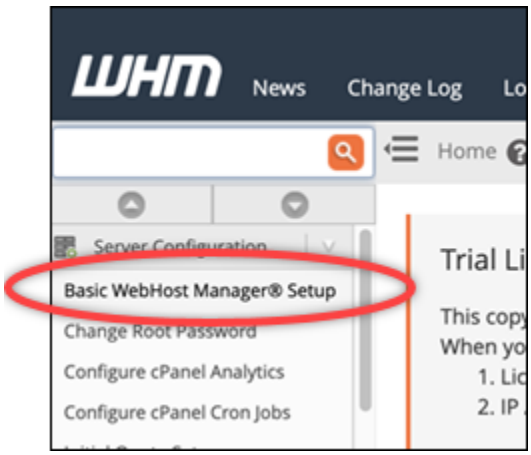
> **Note**
>
> サブドメインは、ホスト名としてのみ指定できます。whm や cpanel はサブドメインとして指定できません。



4. [Change] を選択します。
5. WHM コンソールの左上のセクションにあるナビゲーションメニューアイコンを選択します。

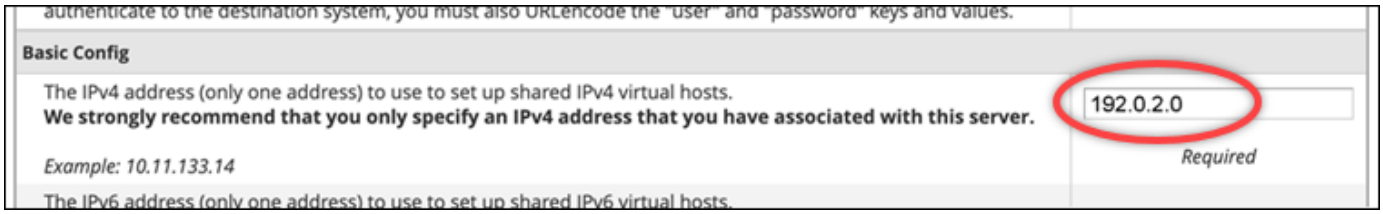


6. [Basic WebHost Manager Setup] を選択します。



7. [すべて] のタブで下にスクロールして、ページの [ベーシック Config] セクションを見つけます。

8. IPv4 アドレステキストボックスに、インスタンスの新しい静的 IP アドレスを入力します。IPv6 の情報については、「cPanel インスタンスで IPv6 の設定」を参照してください。



9. ページの最下部までスクロールして [保存] を選択します。

> **Note**
>
> [無効なライセンスファイル] のエラーメッセージを受けた場合は、しばらく待ってから再度 IP アドレスの変更を試みてください。

インスタンスのホスト名と IP アドレスは変更されましたが、まだ cPanel & WHM インスタンスにドメイン名をマッピングする必要があります。これは、登録済みドメイン名のドメインネームシステム (DNS) に、アドレス (A) レコードを追加することで可能です。A レコードは、インスタンスの静的 IP アドレスにインスタンスのホスト名を解決します。このガイドの次のセクションで、これを行う方法を解説します。

ステップ 5: cPanel & WHM インスタンスにドメイン名をマッピングする

> **Note**
>
> cPanel & WHM インスタンスにドメインをマッピングすることが可能で、これは WHM コンソールにアクセスする際に使用します。WHM 内で複数のドメインをマッピングすることも可能で、WHM 内のウェブサイトを管理する際に使用します。このセクションでは、cPanel & WHM インスタンスにドメインをマッピングする方法について説明します。新しいアカウントを作成する際、 WHM コンソール内に複数のドメインをマッピングしますが、この詳細については、WHMドキュメントの「新しいアカウントを作成」を参照してください。

`management.example.com` や `administration.example.com` などのドメイン名をインスタンスにマッピングするには、ドメインの DNS に A レコードを追加します。A レコードは、cPanel & WHM インスタンスのホスト名をインスタンスの静的 IP アドレスに マッピングします。A レコードで指定するサブドメインは、このガイドの「ステップ 4: cPanel & WHM インスタンスのホスト名と IP アドレスを変更する」のセクションで指定したホスト名と一致する必要があります。A レコード

が追加されたら、インスタンスの静的 IP アドレスを使用する代わりに、次のアドレスを使用してインスタンスの WHM コンソールにアクセスできます。*<InstanceHostName>* をインスタンスのホスト名に置き換えます。

```
https://<InstanceHostName>/whm
```

例:

```
https//management.example.com/whm
```

DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。これを行うには、Lightsail コンソールにサインインします。Lightsail コンソールのホームページで [Domains & DNS] (ドメインと DNS) タブを選択し、[Create DNS zone] (DNS ゾーンの作成) を選択します。ページの手順に従って、ドメイン名を Lightsail に追加します。詳細については、「Lightsail で DNS ゾーンを作成し、ドメインの DNS レコードを管理する」を参照してください。

ステップ 6: インスタンスのファイアウォールを編集する

次のファイアウォールポートはデフォルトで cPanel & WHM インスタンスで開いています。

- SSH - TCP - 22
- DNS (UDP) - UDP - 53
- DNS (TCP) - TCP - 53
- HTTP - TCP - 80
- HTTPS - TCP - 443
- カスタム - TCP - 2078
- カスタム - TCP - 2083
- カスタム - TCP - 2087
- カスタム - TCP - 2089

インスタンスで使用する予定のサービスやアプリケーションによっては、追加でポートを開く必要がある場合もあります。例えば、電子メールサービスの場合はポート 25、143、465、587、993、995、2096 を開き、カレンダーサービスの場合はポート 2080、2091 を開きます。インスタンス管理ページの [ネットワーク] タブで、ページのファイアウォールのセ

クションまでスクロールして [追加ルール] を選択します。アプリケーション、プロトコル、そしてポートまたは開けるポート範囲を選択します。完了したら、[作成] を選択します。

開くべきポートの詳細については、cPanel ドキュメント の「cPanel サービスのファイアウォールを設定する方法」を参照してください。Lightsail におけるインスタンスのファイアウォール編集の詳細については、「Amazon Lightsail でのインスタンスファイアウォールルールの追加と編集」を参照してください。

ステップ 7: Lightsail インスタンスから SMTP 制限の削除

AWS は、すべてのLightsailインスタンスでポート 25 のアウトバウンドトラフィックをブロックします。ポート 25 でアウトバウンドトラフィックを送信するには、この制限の解除をリクエストします。詳細については、「Lightsail インスタンスからポート 25 の制限を解除する方法」を参照してください。

> **Important**
>
> ポート 25、465、または 587 を使用するように SMTP を設定する場合、Lightsail コンソールでインスタンスファイアウォールのポートを開ける必要があります。詳細については、「Amazon Lightsail でインスタンスファイアウォールルールの追加および編集」を参照してください。

ステップ 8: cPanel および WHM のドキュメントを読み込んでサポートを受ける

cPanel & WHM のドキュメントを読んで、cPanel と WHM を使ってウェブサイトを管理する方法を確認ください。詳細については、cPanel & WHM ドキュメント を参照してください。

cPanel & WHM についての質問がある場合やサポートが必要な際は、次のリソースを使用して cPanel にお問い合わせ頂けます。

- インストールの cPanel トラブルシューティング
- cPanel ディスコード チャンネル

ステップ 9: cPanel および WHM のライセンスの購入

cPanel & WHM インスタンスには、15日間のトライアルライセンスが含まれています。15 日後も継続して cPanel & WHM を使用するには cPanel からライセンスを購入する必要があります。詳細については、cPanel ドキュメントの「cPanel のライセンスを購入する方法」を参照してください。

> **Important**
>
> cPanel からライセンスを購入する際は、cPanel & WHM インスタンスのパブリック IP アドレスを指定する必要があります。購入したライセンスは、その IP アドレスに関連付けられます。そのため、このガイドの「ステップ 2: cPanel & WHM インスタンスに静的 IP アドレスをアタッチする」のセクションで解説されているように、cPanel & WHM インスタンスに静的 IP をアタッチする必要があります。cPanel からライセンスを購入する際に静的 IP を指定し、Lightsail インスタンスで cPanel & WHM を使用する限り同じ静的IPを保持することになります。後に別の IP アドレスにライセンスを移転する必要がある場合は、cPanel にリクエストを送信することができます。詳細については、WHM ドキュメント の「ライセンスの移転」を参照してください。

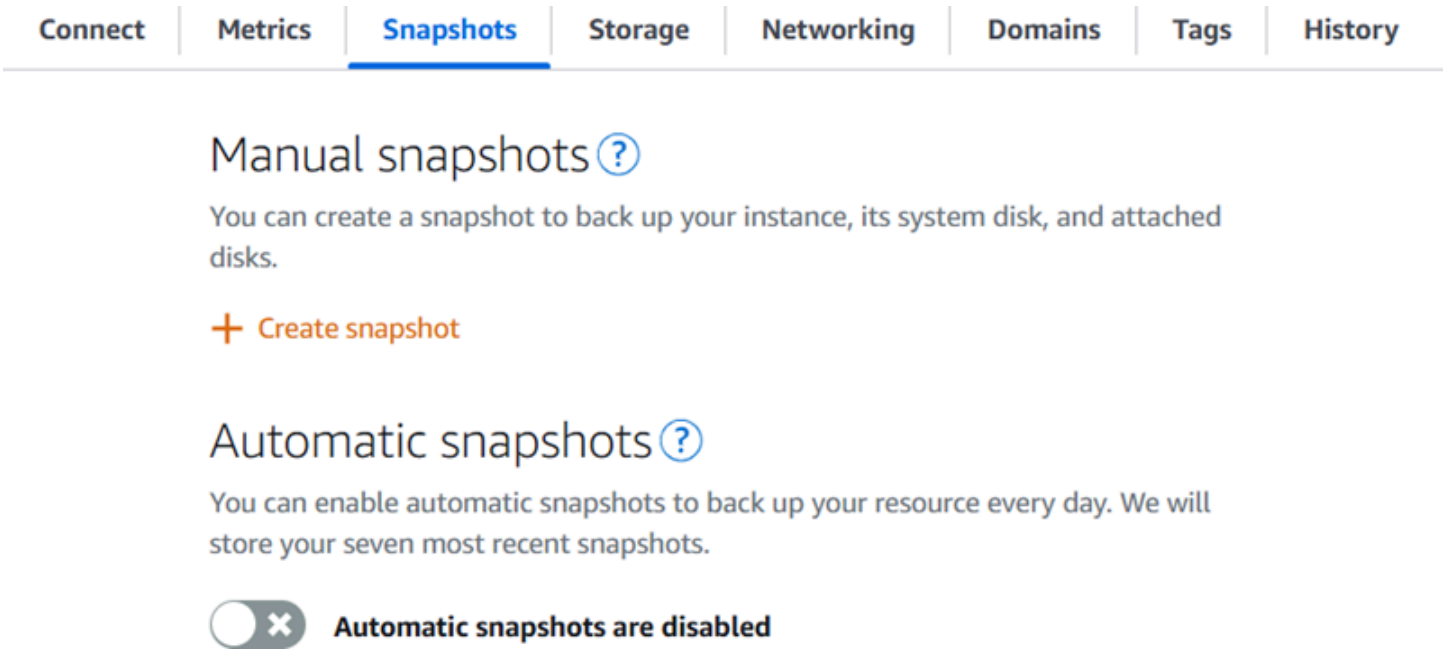
ステップ 10: cPanel および WHM のインスタンスのスナップショットを作成する

> **Note**
>
> - 現世代のブループリントである cPanel & WHM for AlmaLinux のインスタンススナップショットは、Amazon EC2 にエクスポートできます。
> - 前世代のブループリントである cPanel & WHM for AlmaLinux のインスタンススナップショットは、現時点では Amazon EC2 にエクスポートできません。
> - スナップショットから新しいインスタンスを作成する場合、ステップ 3 で説明したように、インスタンスが完全に起動するまでしばらく待ってから WHM にサインインしてください。

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。



Lightsail で Drupal ウェブサイトを設定およびカスタマイズする

> 知っていましたか?
>
> Lightsail は毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを最新のスナップショットに自動的に置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

Amazon Lightsail で起動した Drupal インスタンスの使用を開始するステップについて説明します。

目次

- ステップ 1: Bitnami のドキュメントを確認する
- ステップ 2: Drupal の管理ダッシュボードにアクセスするため、デフォルトのアプリケーションパスワードを取得する
- ステップ 3: インスタンスに静的 IP アドレスをアタッチする
- ステップ 4: Drupal ウェブサイトの管理ダッシュボードにサインインする
- ステップ 5: 登録済みドメイン名へのトラフィックを Drupal ウェブサイトに送信する
- ステップ 6: Drupal ウェブサイトの HTTPS を設定する

- ステップ 7: Drupal のドキュメントを読み、引き続きウェブサイトの設定を続行する
- ステップ 8: インスタンスのスナップショットを作成する

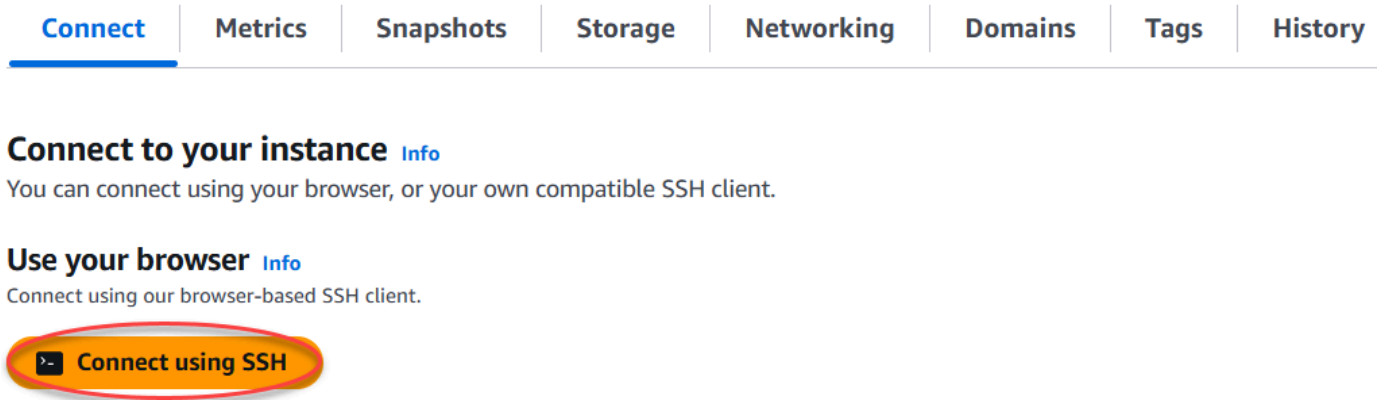
ステップ 1: Bitnami のドキュメントを確認する

Drupal アプリケーションの設定方法については、Bitnami ドキュメントを参照してください。詳細については、「AWS クラウド用に Bitnami がパッケージ化した Drupal」を参照してください。

ステップ 2: Drupal の管理ダッシュボードにアクセスするため、デフォルトのアプリケーションパスワードを取得する

次の手順を完了して、Drupal ウェブサイトの管理ダッシュボードにアクセスする際に必要となるデフォルトのアプリケーションパスワードを取得します。詳細については、「Amazon LightsailのBitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

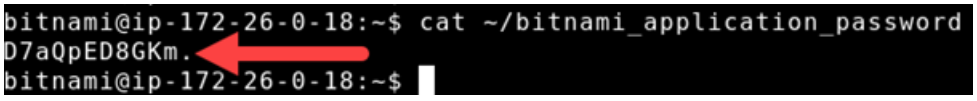
1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。



2. 接続後に、次のコマンドを入力してアプリケーションのパスワードを取得します。

```
cat $HOME/bitnami_application_password
```

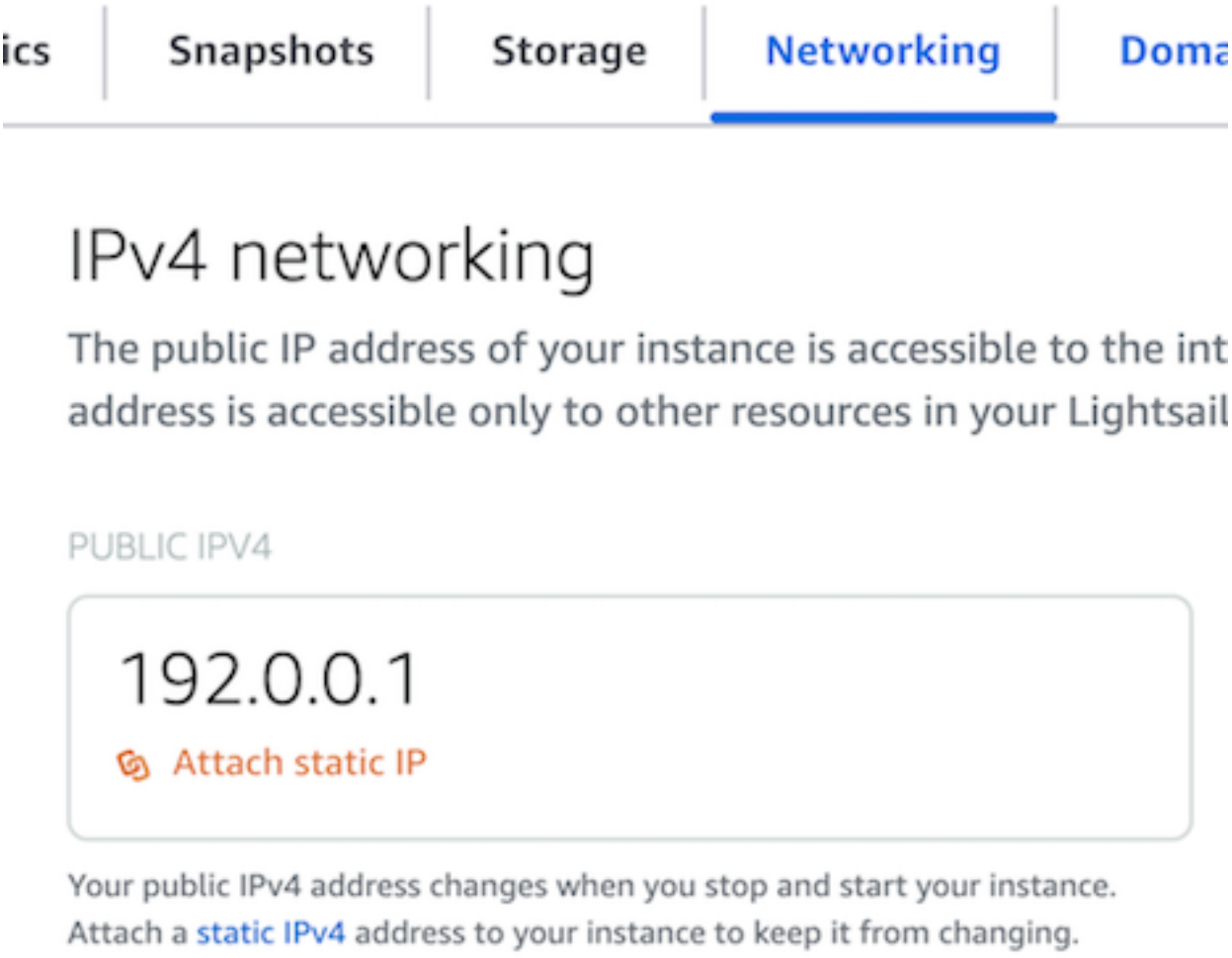
アプリケーションのデフォルトパスワードを含んだ、次の例のようなレスポンスが表示されます。



ステップ 3: インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



ステップ 4: Drupal ウェブサイトの管理ダッシュボードにサインインする

デフォルトのユーザーパスワードを取得したら、Drupal ウェブサイトのホームページに移動し、管理ダッシュボードにサインインします。サインイン後に、ウェブサイトをカスタマイズしたり管理上の変更を行うことができます。Drupal で実行できる事項の詳細については、本ガイドの後半にある

「ステップ 7: Drupal のドキュメントを読み、引き続きウェブサイトの設定を続行する」のセクションを参照してください。

1. インスタンス管理ページの [Connect] (接続) タブにあるパブリック IP アドレスを書き留めます。パブリック IP アドレスは、インスタンス管理ページのヘッダーセクションにも表示されます。

2. インスタンスのパブリック IP アドレスを参照します (例: `http://203.0.113.0` に移動します)。

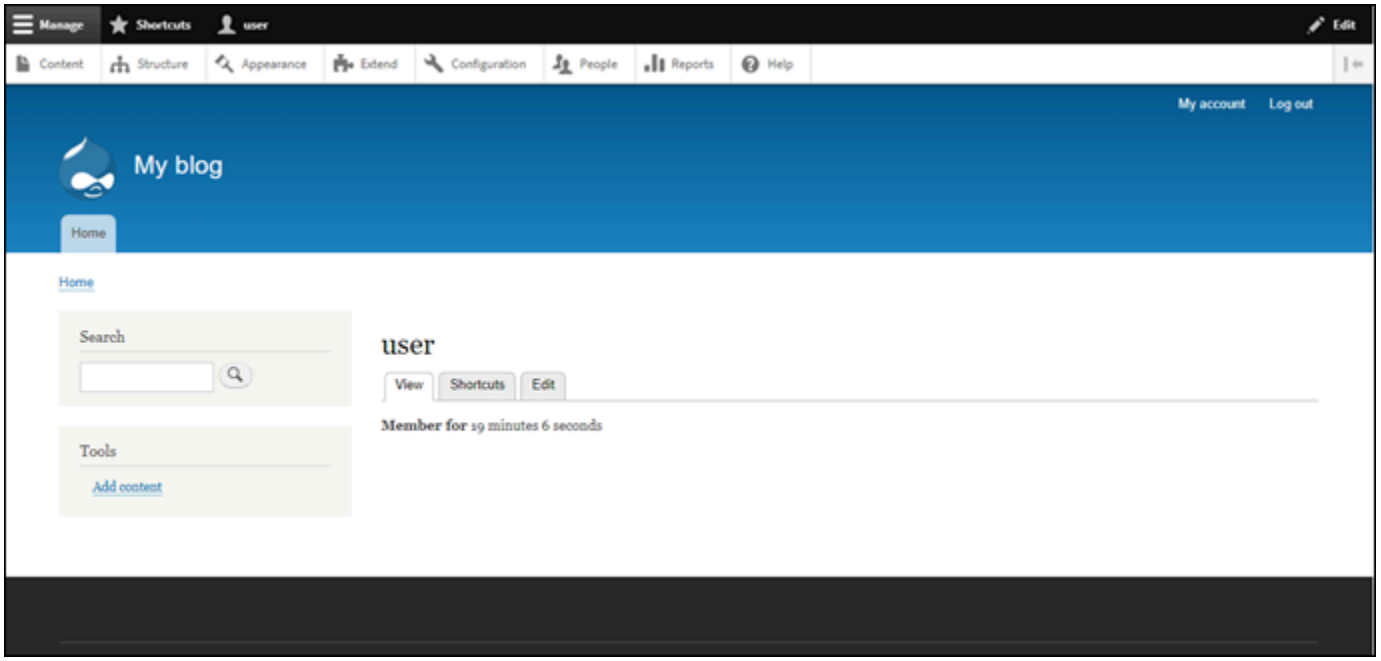
   Drupal ウェブサイトのホームページが表示されます。

3. Drupal ウェブサイトのホームページで、右下にある [Manage] (管理) を選択します。

   [Manage] (管理) バナーが表示されない場合は、`http://<PublicIP>/user/login` を参照することでサインインページにアクセスすることができます。*<PublicIP>* を、インスタンスのパブリック IP アドレスに置き換えます。

4. デフォルトのユーザー名 (`user`) と、先ほど取得したデフォルトのパスワードを使用してサインインします。

   Drupal の管理ダッシュボードが表示されます。



ステップ 5: 登録済みドメイン名へのトラフィックを Drupal ウェブサイトに送信する

`example.com` などの登録済みドメイン名のトラフィックを Drupal ウェブサイトに送信するには、ドメインのドメインネームシステム (DNS) にレコードを追加します。DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページの [Domains & DNS] (ドメインと DNS) タブで、[Create DNS zone] (DNS ゾーンの作成) を選択し、ページに記載される手順に従います。詳細については、「Lightsail で DNS ゾーンを作成し、ドメインの DNS レコードを管理する」を参照してください。

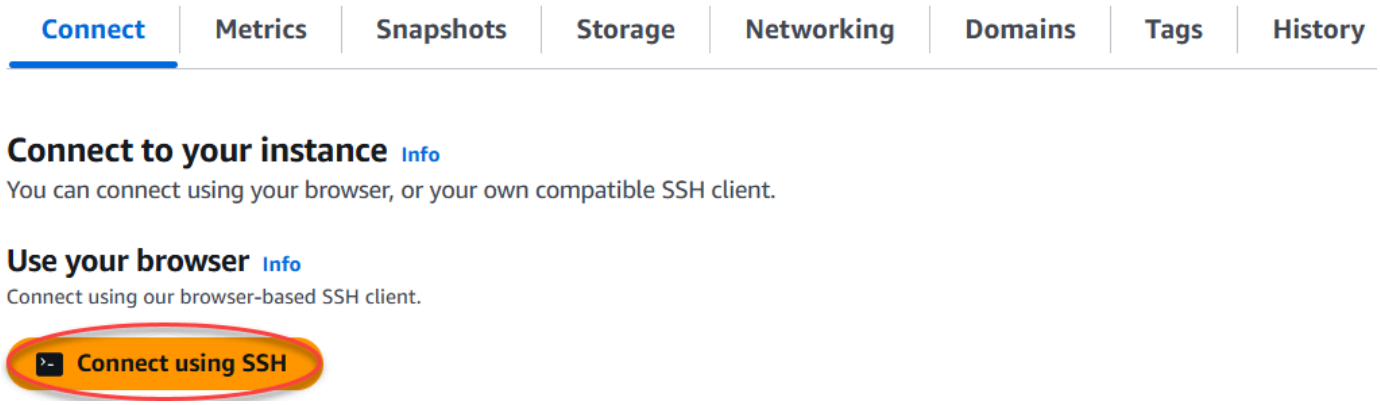
インスタンスに設定したドメイン名を参照すると、Drupal ウェブサイトのホームページへと移動します。次に、SSL/TLS 証明書を生成して設定し、Drupal ウェブサイトの HTTPS 接続を有効にします。詳細については、本ガイドの次の「ステップ 6: Drupal ウェブサイトの HTTPS を設定する」のセクションを参照してください。

ステップ 6: Drupal ウェブサイトの HTTPS を設定する

Drupal ウェブサイトで HTTPS を設定するには、以下の手順を実行します。次の手順では、Bitnami HTTPS 設定ツール (`bncert-tool`) の使い方を説明しています。これは、Let's Encrypt SSL/TLS 証明書を要求するコマンドラインツールです。詳細については、「Bitnami ドキュメント」の「Bitnami HTTPS 設定ツールの詳細」を参照してください。

> **Important**
>
> この手順を開始する前に、Drupal インスタンスにトラフィックがルーティングされるようにドメインが設定済みであることを確認してください。設定されていない場合、SSL/TLS 証明書の検証プロセスが失敗します。

1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。



2. 接続されたら、以下のコマンドを入力し、インスタンスに bncert ツールがインストールされていることを確認します。

```
sudo /opt/bitnami/bncert-tool
```

以下のレスポンスのいずれかが表示されます。

- レスポンスにコマンドが見つからないと表示された場合、bncert ツールがインスタンスにインストールされていないことを示しています。この手順の次のステップに進み、bncert ツールをインスタンスにインストールします。
- レスポンスに「Bitnami HTTPS 設定ツールへようこそ」と表示された場合は、インスタンスに bncert ツールがインストールされています。この手順のステップ 8 に進んでください。
- bncert ツールがインスタンスにインストールされてからしばらく経っている場合、ツールのアップデートバージョンが利用可能であることを示すメッセージが表示されることがあります。ダウンロードすることを選択し、`sudo /opt/bitnami/bncert-tool` コマンドを入力して bncert ツールを再度実行してください。この手順のステップ 8 に進んでください。

3. 以下のコマンドを入力して、bncert の実行ファイルをインスタンスにダウンロードします。

```
wget -O bncert-linux-x64.run https://downloads.bitnami.com/files/bncert/latest/
bncert-linux-x64.run
```

4. 以下のコマンドを入力して、インスタンスに bncert ツールの実行ファイル用のディレクトリを作成します。

```
sudo mkdir /opt/bitnami/bncert
```

5. 以下のコマンドを入力して、プログラムとして実行できるファイルを bncert に実行させます。

```
sudo chmod +x /opt/bitnami/bncert/bncert-linux-x64.run
```

6. 次のコマンドを入力して、sudo /opt/bitnami/bncert-tool コマンドを入力すると bncert ツールを実行するシンボリックリンクを作成します。

```
sudo ln -s /opt/bitnami/bncert/bncert-linux-x64.run /opt/bitnami/bncert-tool
```

これでインスタンスに bncert ツールをインストールする手順は完了です。

7. 次のコマンドを入力して、bncert ツールを実行しましょう。

```
sudo /opt/bitnami/bncert-tool
```

8. 次の例に示すように、プライマリドメイン名と代替ドメイン名の間はスペースで区切って入力します。

ドメインがインスタンスのパブリック IP アドレスにトラフィックをルーティングするように設定されていない場合、bncert ツールは、続行する前にその設定を行うように要求します。ドメインは、bncert ツールを使用して HTTPS を有効にしているインスタンスでのパブリック IP アドレスにトラフィックをルーティングする必要があります。これはドメインを所有していることを確認し、証明書の検証として機能します。

```
------------------------------------------------------------------------------
Welcome to the Bitnami HTTPS Configuration tool.

------------------------------------------------------------------------------
Domains

Please provide a valid space-separated list of domains for which you wish to
configure your web server.

Domain list []: example.com www.example.com
```

9. bncert ツールは、ウェブサイトのリダイレクトの設定方法を尋ねます。使用できるオプションは次のとおりです。

- HTTP から HTTPS へのリダイレクトを有効にする - HTTP バージョンのウェブサイトを閲覧するユーザー (例: `http:/example.com`) を自動的に HTTPS バージョン (例: `https://example.com`) にリダイレクトするかどうかを決定します。すべての訪問者が暗号化された接続を使用するように強制されるため、このオプションを有効にすることをお勧めします。Y を入力して Enter を押すと、有効になります。
- www なしから www ありへのリダイレクトの有効化 - ドメインの頂点 (例: `https://example.com`) まで閲覧するユーザー を自動的にドメインの `www` サブドメイン (例:

https://www.example.com) にリダイレクトするかを指定します。このオプションを有効にすることをお勧めします。ただし、ドメインの頂点を Google のウェブマスターツールなどの検索エンジンツールで希望のウェブサイトアドレスとして指定した場合、または頂点が IP を直接指しており、www のサブドメインが CNAME レコードを介してリファレンスしている場合は、無効にして代替オプションを有効にすることをお勧めします (www ありから www なしへのリダイレクトを有効化) 。Y を入力し、Enter を押して有効にします。

- www ありから www なしへのリダイレクトを有効にする - ドメインの www サブドメイン (例: https://www.example.com) まで閲覧するユーザーを、自動的にドメインの頂点 (例: https://example.com) にリダイレクトするかを指定します。www なしから www ありへのリダイレクトを有効にした場合は、これを無効にすることをお勧めします。N を入力し、Enter を押して無効にします。

選択した結果は次の例のようになります。

```
Enable/disable redirections

Please select the redirections you wish to enable or disable on your Bitnami
installation.



Enable HTTP to HTTPS redirection [Y/n]: Y



Enable non-www to www redirection [Y/n]: Y



Enable www to non-www redirection [y/N]: N
```

10. これから実行される変更が一覧表示されます。Y と入力し、Enter を押して確認し、続行します。

```
Changes to perform

The following changes will be performed to your Bitnami installation:

1. Stop web server
2. Configure web server to use a free Let's Encrypt certificate for the domains:
example.com www.example.com
3. Configure a cron job to automatically renew the certificate each month
4. Configure web server name to: example.com
5. Enable HTTP to HTTPS redirection (example: redirect http://example.com to
https://example.com)
6. Enable non-www to www redirection (example: redirect example.com to
www.example.com)
7. Start web server once all changes have been performed



Do you agree to these changes? [Y/n]: Y
```

11. Let's Encrypt 証明書に関連付けるメールアドレスを入力し、Enter を押します。

```
Create a free HTTPS certificate with Let's Encrypt

Please provide a valid e-mail address for which to associate your Let's Encrypt
certificate.

Domain list: example.com www.example.com

Server name: example.com

E-mail address []:
```

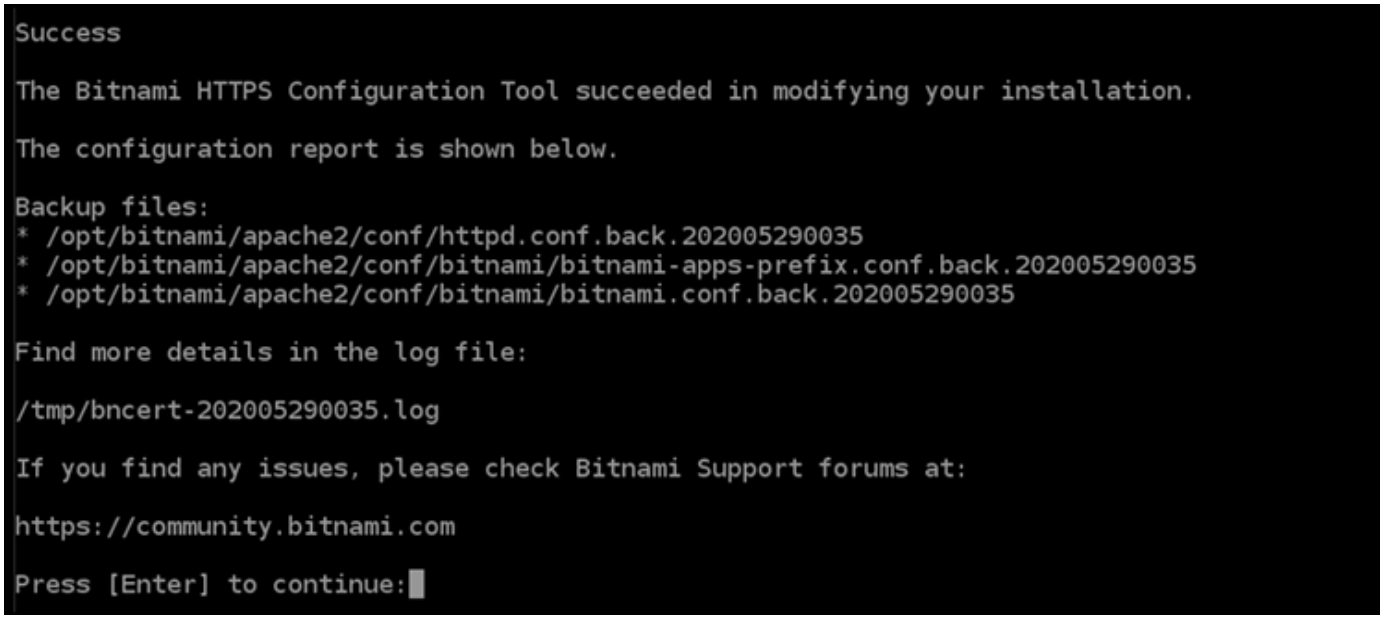
12. Let's Encrypt サブスクライバー合意書を確認します。Y と入力し、Enter を押して契約に同意し、続行します。

```
The Let's Encrypt Subscriber Agreement can be found at:

https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf

Do you agree to the Let's Encrypt Subscriber Agreement? [Y/n]:
```

これらのアクションは、証明書のリクエストや指定したリダイレクトの設定など、インスタンスで HTTPS を有効にするために実行されます。

```
Performing changes to your installation

The Bitnami HTTPS Configuration Tool will perform any necessary actions to your
Bitnami installation. This may take some time, please be patient.

|
```

次の例のようなメッセージが表示された場合は、証明書は正常に発行され、検証され、インスタンスでリダイレクトが正常に設定されています。



bncert ツールは、有効期限が切れる前、 80 日ごとに証明書の自動更新を実行します。インスタンスで追加のドメインやサブドメインを使用し、それらのドメインで HTTPS を有効にする場合は、上記のステップを繰り返します。

これで、Drupal インスタンスでの HTTPS の有効化が完了しました。次回に、設定したドメインを使用して Drupal ウェブサイトを参照するときには、HTTPS 接続にリダイレクトされるはずです。

ステップ 7: Drupal のドキュメントを読み、引き続きウェブサイトの設定を続行する

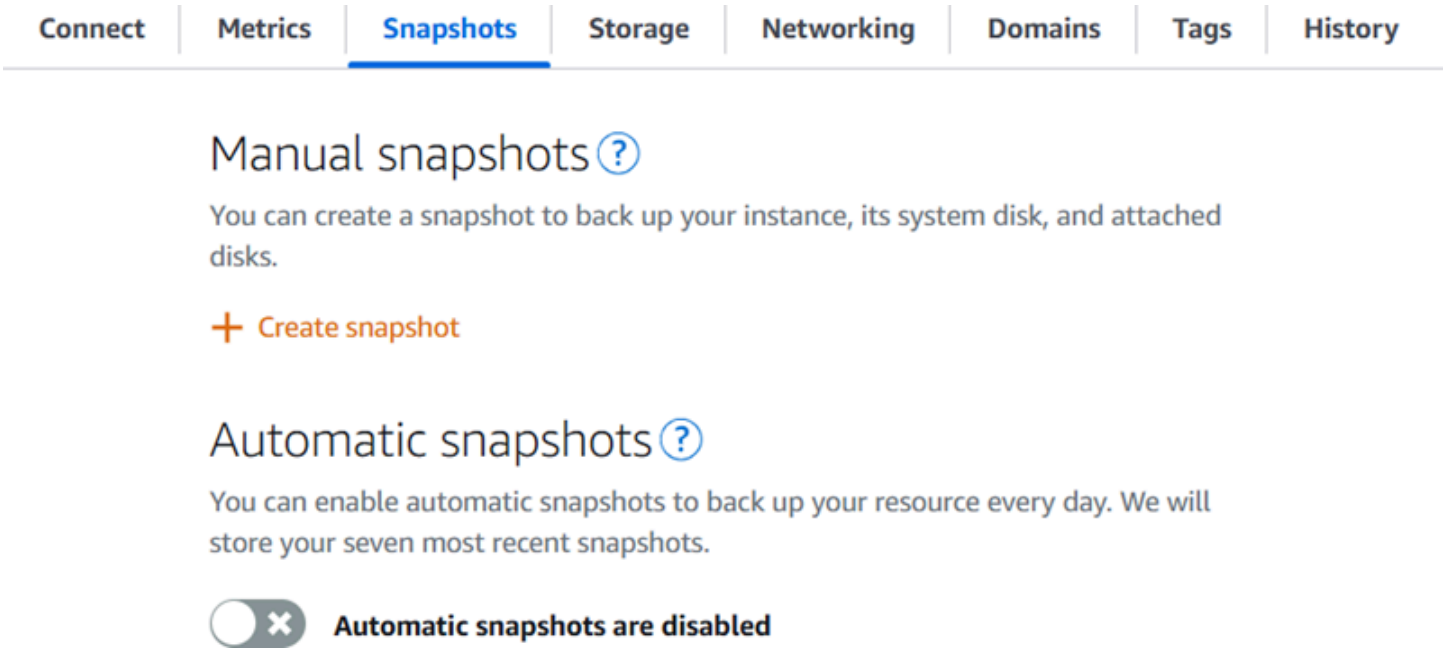
Drupal のドキュメントを読み、ウェブサイトを管理およびカスタマイズする方法を確認します。詳細については、「Drupal Documentation」(Drupal ドキュメント) を参照してください。

ステップ 8: インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。



Lightsail で Ghost ウェブサイトをデプロイする

> 知っていましたか?
>
> Lightsail は、毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを自動的に最新のスナップショットに置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

Amazon Lightsail で起動した Ghost インスタンスの使用を開始するステップについて説明します。

目次

- ステップ 1: Bitnami のドキュメントを確認する
- ステップ 2: Ghost の管理ダッシュボードにアクセスするため、デフォルトのアプリケーションパスワードを取得する
- ステップ 3: インスタンスに静的 IP アドレスをアタッチする
- ステップ 4: Ghost ウェブサイトの管理ダッシュボードにサインインする
- ステップ 5: 登録済みドメイン名へのトラフィックを Ghost ウェブサイトに送信する

- ステップ 6: Ghost ウェブサイトの HTTPS を設定する
- ステップ 7: Ghost のドキュメントを読み、引き続きウェブサイトの設定を続行する
- ステップ 8: インスタンスのスナップショットを作成する

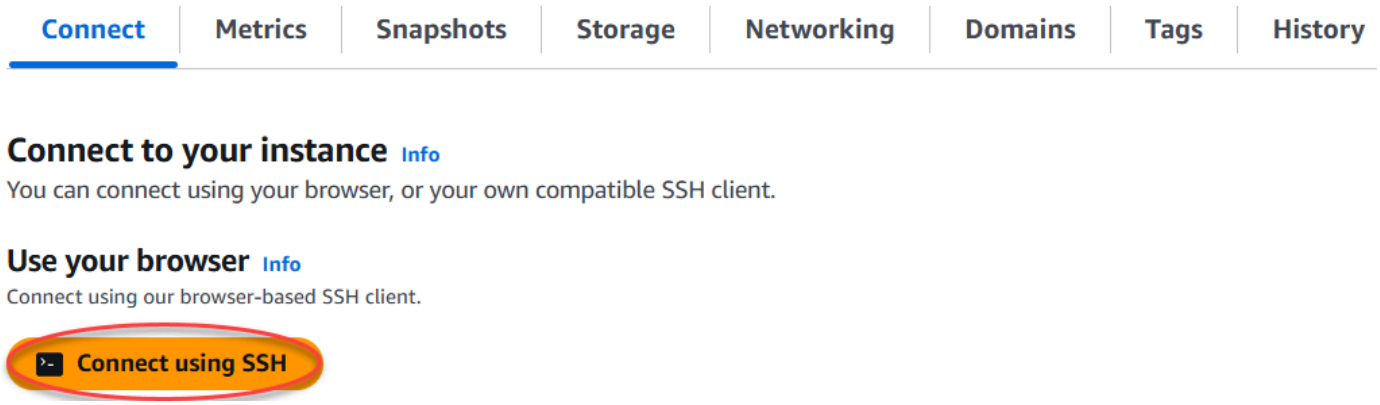
ステップ 1: Bitnami のドキュメントを確認する

Ghost アプリケーションの設定方法については、Bitnami ドキュメントを参照してください。詳細については、「AWS クラウド用に Bitnami がパッケージ化した Ghost」を参照してください。

ステップ 2: Ghost の管理ダッシュボードにアクセスするため、デフォルトのアプリケーションパスワードを取得する

次の手順を完了して、Ghost ウェブサイトの管理ダッシュボードにアクセスする際に必要となるデフォルトのアプリケーションパスワードを取得します。詳細については、「Amazon Lightsail の Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。



2. 接続後に、次のコマンドを入力してアプリケーションのパスワードを取得します。

```
$ cat $HOME/bitnami_application_password
```

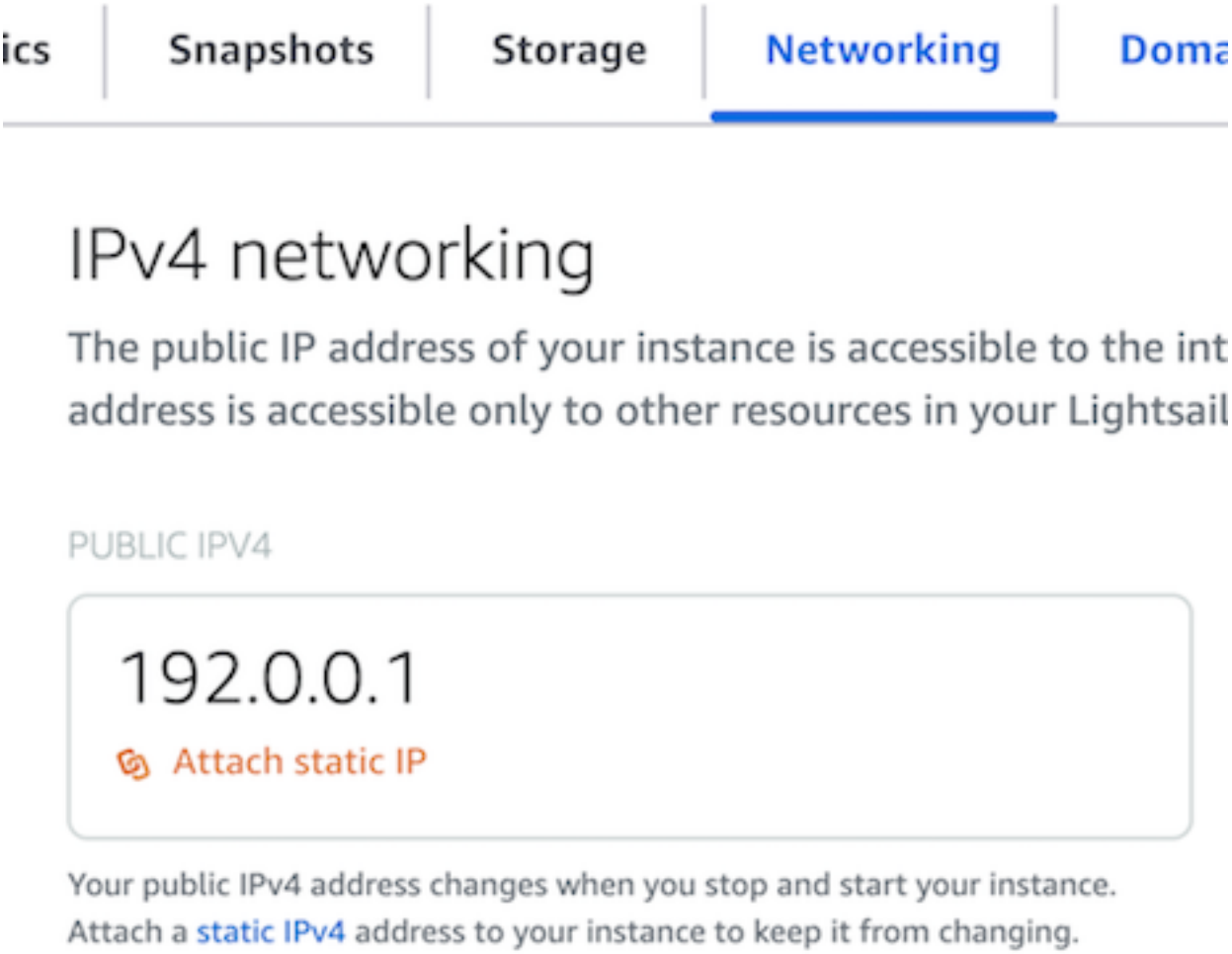
アプリケーションのデフォルトパスワードを含んだ、次のようなレスポンスが表示されます:

```
bitnami@ip-192-0-2-0:~$ cat $HOME/bitnami_application_password
wB2Ex@mplEK6
```

ステップ 3: インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

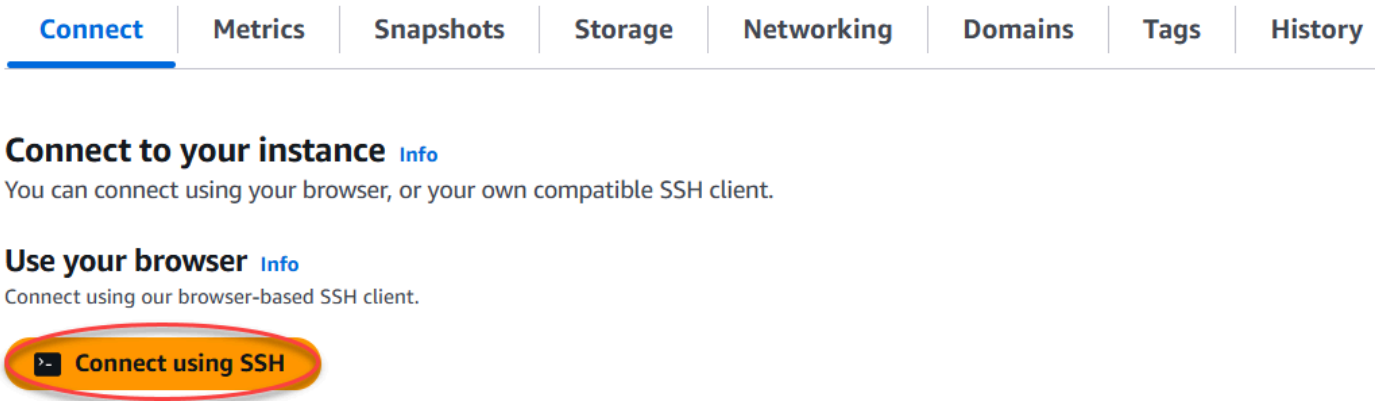
インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



新しい静的 IP アドレスがインスタンスに添付されたら、次の手順を実行して、アプリケーションに新しい静的 IP アドレスを認識させる必要があります。

1. インスタンスの静的 IP アドレスは書き留めておきます。このIP アドレスはインスタンス管理ページの ヘッダーセクションに表示されます。

2. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。



3. 接続後に、次のコマンドを入力します。*<StaticIP>* をインスタンスの新しい静的 IP アドレスに置き換えます。

```
sudo /opt/bitnami/configure_app_domain --domain <StaticIP>
```

例:

```
sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
```

次のようなレスポンスが表示されます。これで、インスタンス上のアプリケーションが新しい静的 IP アドレスを認識するようになります。

```
bitnami@ip-203.0.113.0:~$ sudo /opt/bitnami/configure_app_domain --domain
 203.0.113.0
Configuring domain to 203.0.113.0
2024-06-06T21:43:42.393Z - info: Saving configuration info to disk
ghost 21:43:42.78 INFO  ==> Configuring Ghost URL to http://203.0.113.0
Disabling automatic domain update for IP address changes
```

ステップ 4: Ghost ウェブサイトの管理ダッシュボードにサインインする

デフォルトのユーザーパスワードを取得したら、以下の手順に従って Ghost ウェブサイトのホームページに移動し、管理ダッシュボードにサインインします。サインイン後に、ウェブサイトをカスタマイズしたり管理上の変更を行うことができます。Ghost で実行できる事項の詳細については、本ガイドの後半にある「ステップ 6: Ghost のドキュメントを読み、引き続きウェブサイトの設定を続行する」のセクションを参照してください。

1. インスタンス管理ページの [Connect] (接続) タブにあるパブリック IP アドレスを書き留めます。インスタンスに静的 IP を以前にアタッチした場合、これは静的 IP アドレスになります。パブリック IP アドレスは、インスタンス管理ページのヘッダーセクションにも表示されます。

2. インスタンスのパブリック IP アドレスを参照します (例: `http://203.0.113.0` に移動します)。

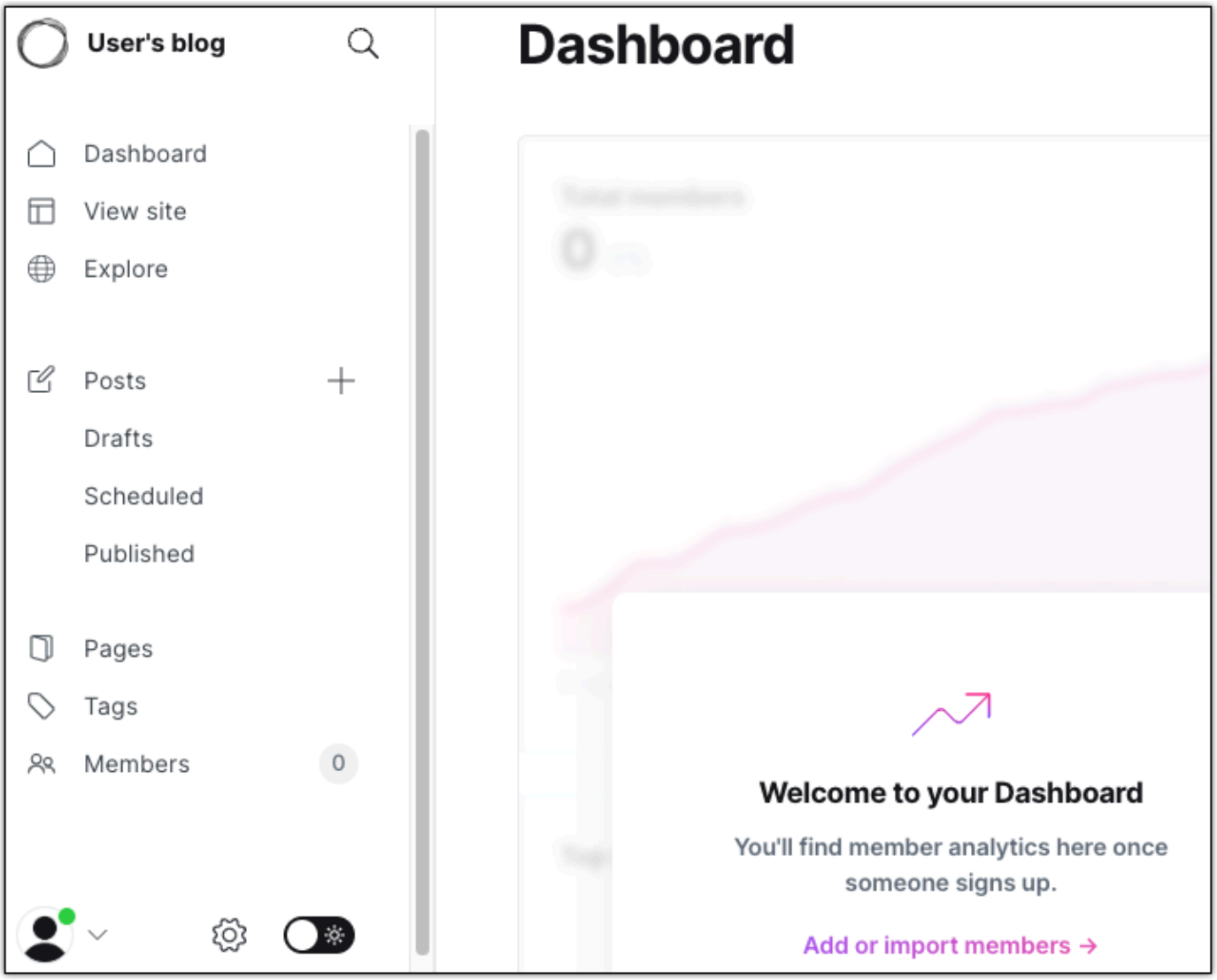
   Ghost ウェブサイトのホームページが表示されます。

3. Ghost ウェブサイトのホームページで、右下にある [Manage] (管理) を選択します。

   [Manage] (管理) バナーが表示されない場合は、`http://<PublicIP>/ghost` を参照することでサインインページにアクセスすることができます。*<PublicIP>* を、インスタンスのパブリック IP アドレスに置き換えます。

4. デフォルトのユーザー名 (`user@example.com`) と、先ほど取得したデフォルトのパスワードを使用してサインインします。

   Ghost の管理ダッシュボードが表示されます。



ステップ 5: 登録済みドメイン名へのトラフィックを Ghost ウェブサイトに送信する

`example.com` などの登録済みドメイン名のトラフィックを Ghost ウェブサイトに送信するには、ドメインの DNS にレコードを追加します。DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページの [ドメインと DNS] セクションで、[DNS ゾーンの作成] を選択し、ページに記載される手順に従います。詳細については、「Lightsail で DNS ゾーンを作成し、ドメインの DNS レコードを管理する」を参照してください。

ドメイン名へのトラフィックがインスタンスにルーティングされたら、次の手順を実行して、Ghost アプリケーションにドメイン名を認識させます。

1. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。
2. 接続後に、次のコマンドを入力します。*<DomainName>* は、Ghost インスタンスにトラフィックをルーティングするドメイン名に置き換えてください。

```
$ sudo /opt/bitnami/configure_app_domain --domain <DomainName>
```

例:

```
$ sudo /opt/bitnami/configure_app_domain --domain example.com
```

次の例のようなレスポンスが表示されます。これで、Ghost アプリケーションがドメインを認識するようになりました。

```
bitnami@ip-203.0.113.0:~$ sudo /opt/bitnami/configure_app_domain --domain
 example.com
Configuring domain to example.com
2024-06-06T21:50:00.393Z - info: Saving configuration info to disk
ghost 21:50:25.78 INFO  ==> Configuring Ghost URL to http://example.com
Disabling automatic domain update for IP address changes
```

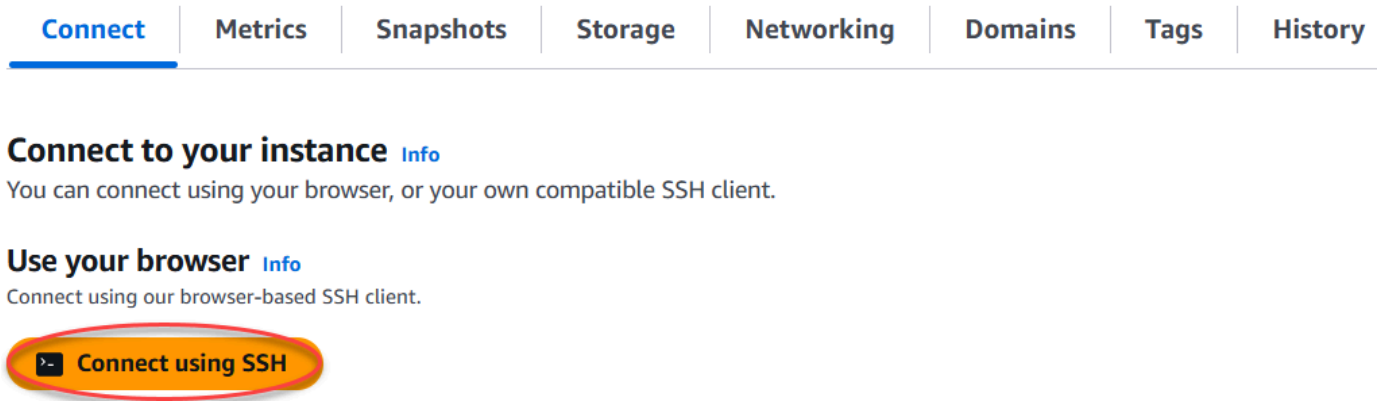
インスタンスに設定したドメイン名を参照すると、Ghost ウェブサイトのホームページへと移動します。次に、SSL/TLS 証明書を生成して設定し、Ghost ウェブサイトの HTTPS 接続を有効にします。詳細については、本ガイドの次の「ステップ 6: Ghost ウェブサイトの HTTPS を設定する」のセクションを参照してください。

ステップ 6: Ghost ウェブサイトの HTTPS を設定する

Ghost ウェブサイトで HTTPS を設定するには、以下の手順を実行します。次の手順では、Bitnami HTTPS 設定ツール (`bncert-tool`) の使い方を説明しています。これは、Let's Encrypt SSL/TLS 証明書を要求するコマンドラインツールです。詳細については、「Bitnami ドキュメント」の「Bitnami HTTPS 設定ツールの詳細」を参照してください。

> **Important**
>
> この手順を開始する前に、Ghost インスタンスにトラフィックがルーティングされるようにドメインが設定済みであることを確認してください。設定されていない場合、SSL/TLS 証明書の検証プロセスが失敗します。

1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。



2. 接続されたら、以下のコマンドを入力し、インスタンスに bncert ツールがインストールされていることを確認します。

```
sudo /opt/bitnami/bncert-tool
```

以下のレスポンスのいずれかが表示されます。

- レスポンスにコマンドが見つからないと表示された場合、bncert ツールがインスタンスにインストールされていないことを示しています。この手順の次のステップに進み、bncert ツールをインスタンスにインストールします。
- レスポンスに「Bitnami HTTPS 設定ツールへようこそ」と表示された場合は、インスタンスに bncert ツールがインストールされています。この手順のステップ 8 に進んでください。
- bncert ツールがインスタンスにインストールされてからしばらく経っている場合、ツールのアップデートバージョンが利用可能であることを示すメッセージが表示されることがあります。ダウンロードすることを選択し、`sudo /opt/bitnami/bncert-tool` コマンドを入力して bncert ツールを再度実行してください。この手順のステップ 8 に進んでください。

3. 以下のコマンドを入力して、bncert の実行ファイルをインスタンスにダウンロードします。

```
wget -O bncert-linux-x64.run https://downloads.bitnami.com/files/bncert/latest/
bncert-linux-x64.run
```

4. 以下のコマンドを入力して、インスタンスに bncert ツールの実行ファイル用のディレクトリを作成します。

```
sudo mkdir /opt/bitnami/bncert
```

5. 以下のコマンドを入力して、プログラムとして実行できるファイルを bncert に実行させます。

```
sudo chmod +x /opt/bitnami/bncert/bncert-linux-x64.run
```

6. 次のコマンドを入力して、sudo /opt/bitnami/bncert-tool コマンドを入力すると bncert ツールを実行するシンボリックリンクを作成します。

```
sudo ln -s /opt/bitnami/bncert/bncert-linux-x64.run /opt/bitnami/bncert-tool
```

これでインスタンスに bncert ツールをインストールする手順は完了です。

7. 次のコマンドを入力して、bncert ツールを実行しましょう。

```
sudo /opt/bitnami/bncert-tool
```

8. 次の例に示すように、プライマリドメイン名と代替ドメイン名の間はスペースで区切って入力します。

ドメインがインスタンスのパブリック IP アドレスにトラフィックをルーティングするように設定されていない場合、bncert ツールは、続行する前にその設定を行うように要求します。ドメインは、bncert ツールを使用して HTTPS を有効にしているインスタンスでのパブリック IP アドレスにトラフィックをルーティングする必要があります。これはドメインを所有していることを確認し、証明書の検証として機能します。

```
------------------------------------------------------------------------------
Welcome to the Bitnami HTTPS Configuration tool.

------------------------------------------------------------------------------
Domains

Please provide a valid space-separated list of domains for which you wish to
configure your web server.

Domain list []: example.com www.example.com
```

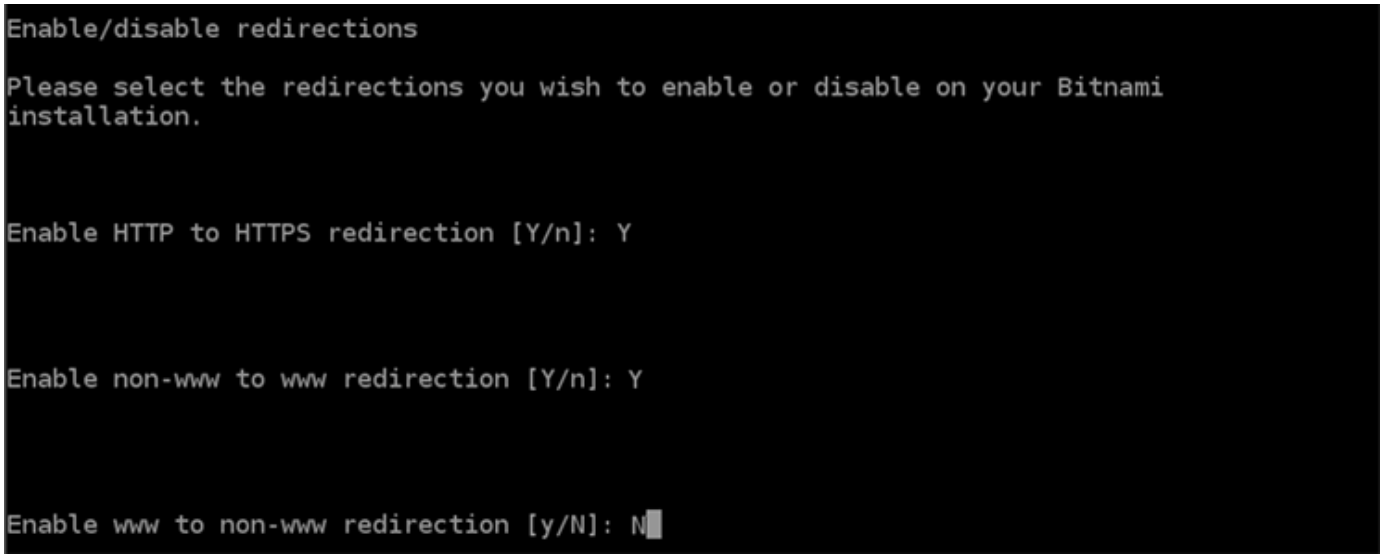
9. bncert ツールは、ウェブサイトのリダイレクトの設定方法を尋ねます。使用できるオプションは次のとおりです。

- HTTP から HTTPS へのリダイレクトを有効にする - HTTP バージョンのウェブサイトを閲覧するユーザー (例: `http:/example.com`) を自動的に HTTPS バージョン (例: `https://example.com`) にリダイレクトするかどうかを決定します。すべての訪問者が暗号化された接続を使用するように強制されるため、このオプションを有効にすることをお勧めします。Y を入力して Enter を押すと、有効になります。
- www なしから www ありへのリダイレクトの有効化 - ドメインの頂点 (例: `https://example.com`) まで閲覧するユーザー を自動的にドメインの `www` サブドメイン (例:

https://www.example.com) にリダイレクトするかを指定します。このオプションを有効にすることをお勧めします。ただし、ドメインの頂点を Google のウェブマスターツールなどの検索エンジンツールで希望のウェブサイトアドレスとして指定した場合、または頂点が IP を直接指しており、www のサブドメインが CNAME レコードを介してリファレンスしている場合は、無効にして代替オプションを有効にすることをお勧めします (www ありから www なしへのリダイレクトを有効化) 。Y を入力し、Enter を押して有効にします。

- www ありから www なしへのリダイレクトを有効にする - ドメインの www サブドメイン (例: https://www.example.com) まで閲覧するユーザーを、自動的にドメインの頂点 (例: https://example.com) にリダイレクトするかを指定します。www なしから www ありへのリダイレクトを有効にした場合は、これを無効にすることをお勧めします。N を入力し、Enter を押して無効にします。

選択した結果は次の例のようになります。



10. これから実行される変更が一覧表示されます。Y と入力し、Enter を押して確認し、続行します。

```
Changes to perform

The following changes will be performed to your Bitnami installation:

1. Stop web server
2. Configure web server to use a free Let's Encrypt certificate for the domains:
example.com www.example.com
3. Configure a cron job to automatically renew the certificate each month
4. Configure web server name to: example.com
5. Enable HTTP to HTTPS redirection (example: redirect http://example.com to
https://example.com)
6. Enable non-www to www redirection (example: redirect example.com to
www.example.com)
7. Start web server once all changes have been performed



Do you agree to these changes? [Y/n]: Y
```

11. Let's Encrypt 証明書に関連付けるメールアドレスを入力し、Enter を押します。

```
Create a free HTTPS certificate with Let's Encrypt

Please provide a valid e-mail address for which to associate your Let's Encrypt
certificate.

Domain list: example.com www.example.com

Server name: example.com

E-mail address []:
```

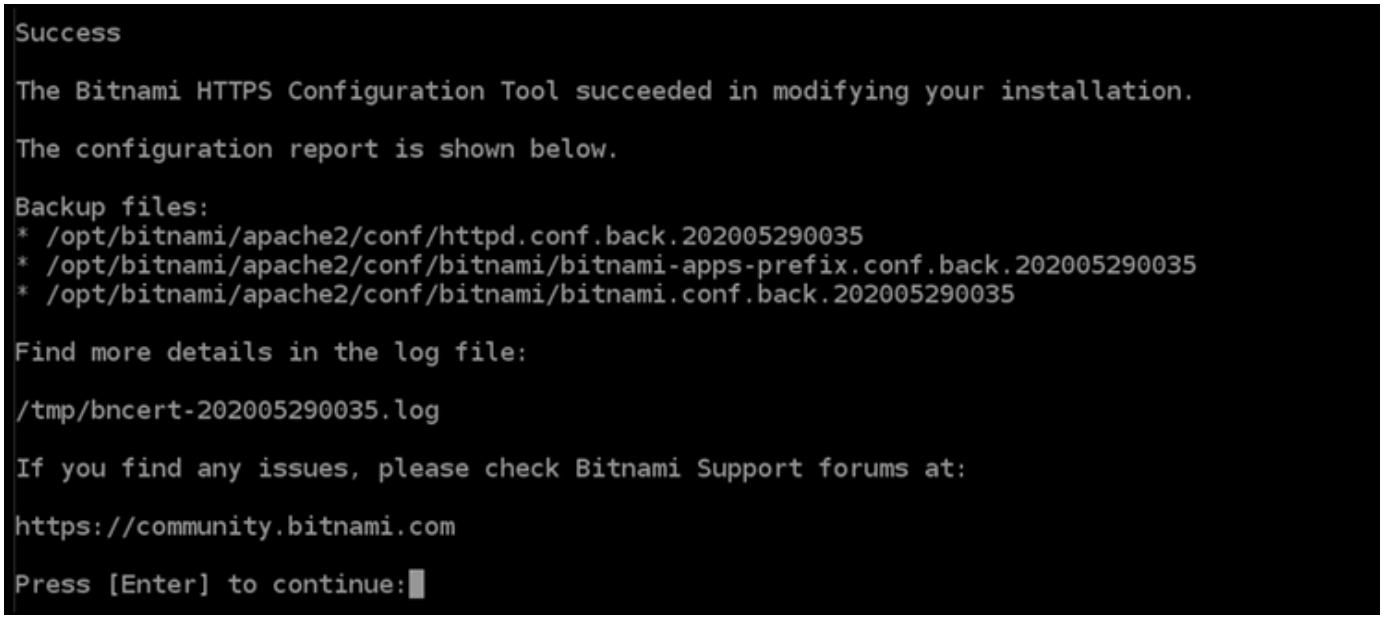
12. Let's Encrypt サブスクライバー合意書を確認します。Y と入力し、Enter を押して契約に同意し、続行します。

```
The Let's Encrypt Subscriber Agreement can be found at:

https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf

Do you agree to the Let's Encrypt Subscriber Agreement? [Y/n]:
```

これらのアクションは、証明書のリクエストや指定したリダイレクトの設定など、インスタンスで HTTPS を有効にするために実行されます。

```
Performing changes to your installation

The Bitnami HTTPS Configuration Tool will perform any necessary actions to your
Bitnami installation. This may take some time, please be patient.

|
```

次の例のようなメッセージが表示された場合は、証明書は正常に発行され、検証され、インスタンスでリダイレクトが正常に設定されています。



bncert ツールは、有効期限が切れる前、 80 日ごとに証明書の自動更新を実行します。インスタンスで追加のドメインやサブドメインを使用し、それらのドメインで HTTPS を有効にする場合は、上記のステップを繰り返します。

> **Tip**
>
> 以下のコマンドを入力して、インスタンスのサービスを再起動します。
>
> ```
> sudo /opt/bitnami/ctlscript.sh restart
> ```

これで、Ghost インスタンスでの HTTPS の有効化が完了しました。次回に、設定したドメインを使用して Ghost ウェブサイトを参照するときには、HTTPS 接続にリダイレクトされるはずです。

ステップ 7: Ghost のドキュメントを読み、引き続きウェブサイトの設定を続行する

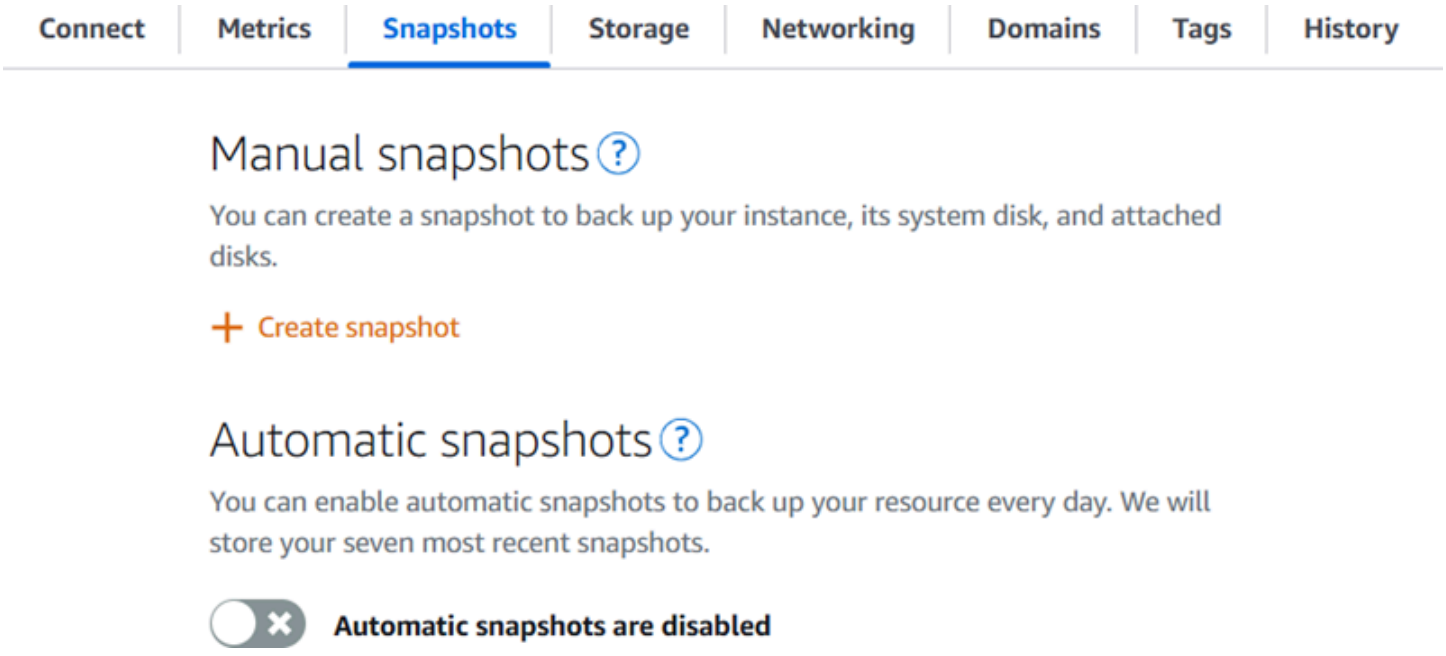
Ghost のドキュメントを読み、ウェブサイトを管理およびカスタマイズする方法を確認します。詳細については、 Ghost のドキュメントを参照してください。

ステップ 8: インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。



Lightsail で GitLab CE インスタンスを設定および構成する

> **知っていましたか?**
>
> Lightsail は毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを最新のスナップショットに自動的に置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

Amazon Lightsail で起動した GitLab CE インスタンスの使用を開始するステップについて説明します。

目次

- ステップ 1: Bitnami のドキュメントを確認する
- ステップ 2: GitLab CE の管理エリアにアクセスするため、デフォルトのアプリケーションパスワードを取得する
- ステップ 3: インスタンスに静的 IP アドレスをアタッチする
- ステップ 4: GitLab CE ウェブサイトの管理エリアにサインインする
- ステップ 5: 登録済みドメイン名へのトラフィックを GitLab CE ウェブサイトに送信する
- ステップ 6: GitLab CE ウェブサイトの HTTPS を設定する
- ステップ 7: GitLab CE のドキュメントを読み、引き続きウェブサイトの設定を続行する
- ステップ 8: インスタンスのスナップショットを作成する

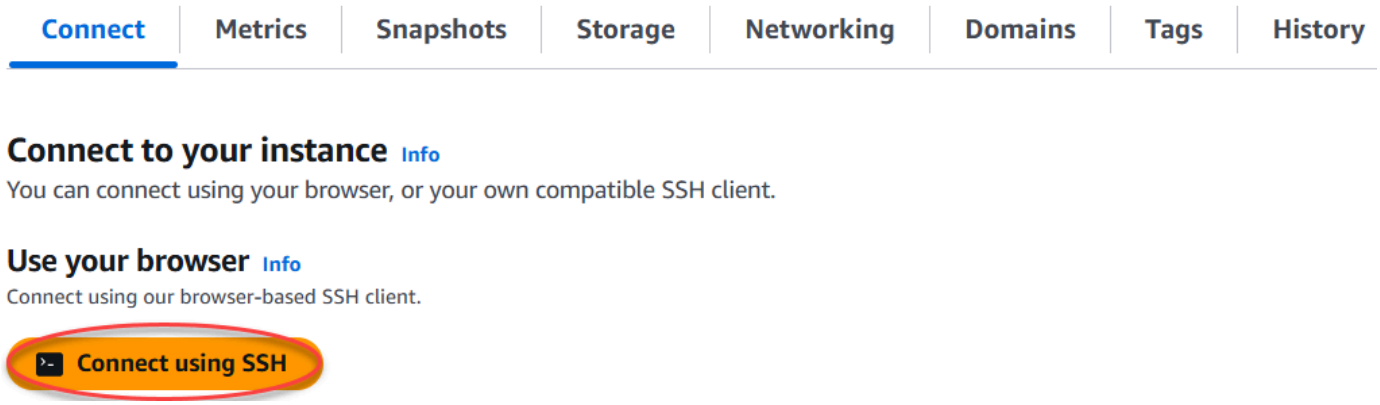
ステップ 1: Bitnami のドキュメントを確認する

GitLab CE アプリケーションの設定方法については、Bitnami ドキュメントを参照してください。詳細については、「AWS クラウド用に Bitnami がパッケージ化した GitLab CE」を参照してください。

ステップ 2: GitLab CE の管理エリアにアクセスするため、デフォルトのアプリケーションパスワードを取得する

次の手順を完了して、GitLab CE ウェブサイトの管理エリアにアクセスする際に必要となるデフォルトのアプリケーションパスワードを取得します。詳細については、「Amazon Lightsailの Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

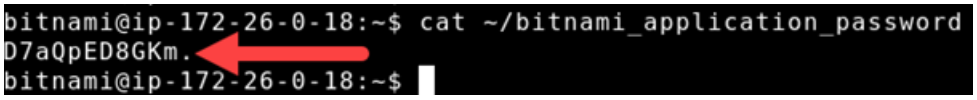
1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。



2. 接続後に、次のコマンドを入力してアプリケーションのパスワードを取得します。

```
cat $HOME/bitnami_application_password
```

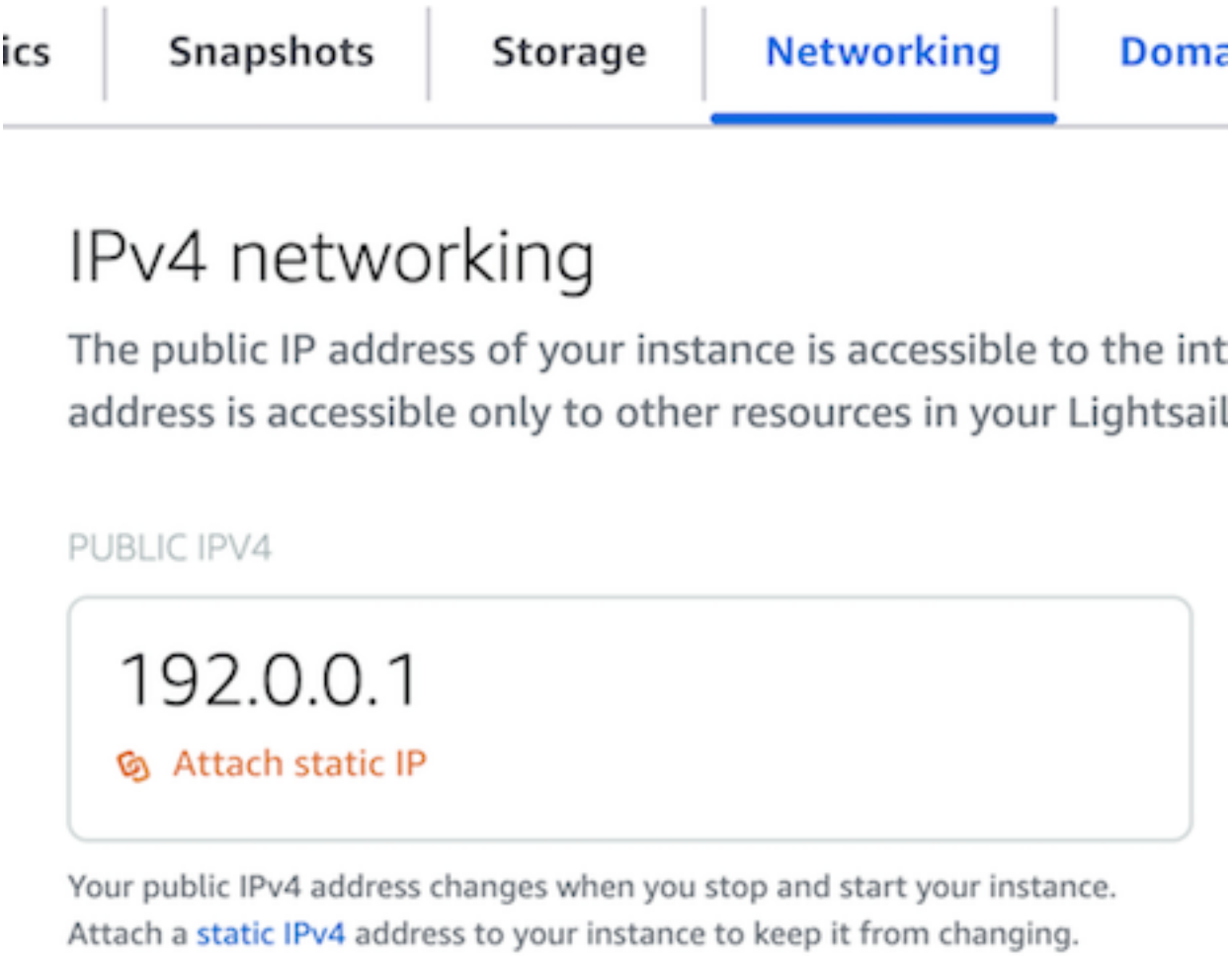
アプリケーションのデフォルトパスワードを含んだ、次の例のようなレスポンスが表示されます。



ステップ 3: インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

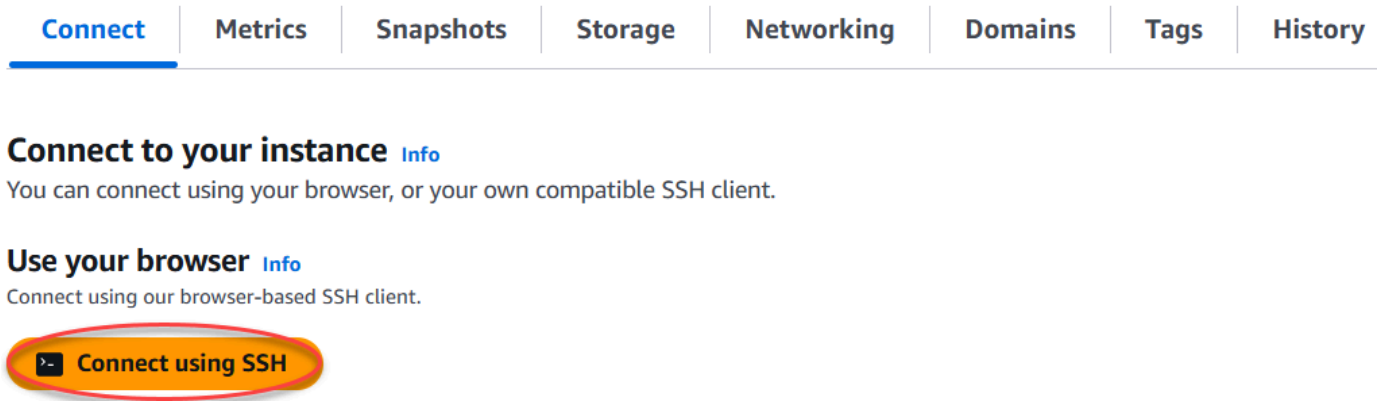
インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



新しい静的 IP アドレスがインスタンスに添付されたら、次の手順を実行して、アプリケーションに新しい静的 IP アドレスを認識させる必要があります。

1. インスタンスの静的 IP アドレスは書き留めておきます。このIP アドレスはインスタンス管理ページの ヘッダーセクションに表示されます。

2. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。



3. 接続後に、次のコマンドを入力します。*<StaticIP>* をインスタンスの新しい静的 IP アドレスに置き換えます。

```
sudo /opt/bitnami/configure_app_domain --domain <StaticIP>
```

例:

```
sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
```

次の例のようなレスポンスが表示されます。これで、インスタンス上のアプリケーションが新しい静的 IP アドレスを認識するようになります。

```
bitnami@ip-[redacted]:~$ sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
Configuring domain to 203.0.113.0
2022-06-09T16:47:06.737Z - info: Saving configuration info to disk
gitlab 16:47:06.86 INFO  ==> Updating external URL in GitLab configuration
gitlab 16:47:06.88 INFO  ==> Reconfiguring GitLab
gitlab 16:47:45.29 INFO  ==> Starting GitLab services
Disabling automatic domain update for IP address changes
```

ステップ 4: GitLab CE ウェブサイトの管理エリアにサインインする

デフォルトのユーザーパスワードを取得したら、GitLab CE ウェブサイトのホームページに移動し、管理エリアにサインインします。サインイン後に、ウェブサイトをカスタマイズしたり管理上の変更を行うことができます。GitLab CE で実行できる事項の詳細については、本ガイドの後半にある「ステップ 7: GitLab CE のドキュメントを読み、引き続きウェブサイトの設定を続行する」のセクションを参照してください。

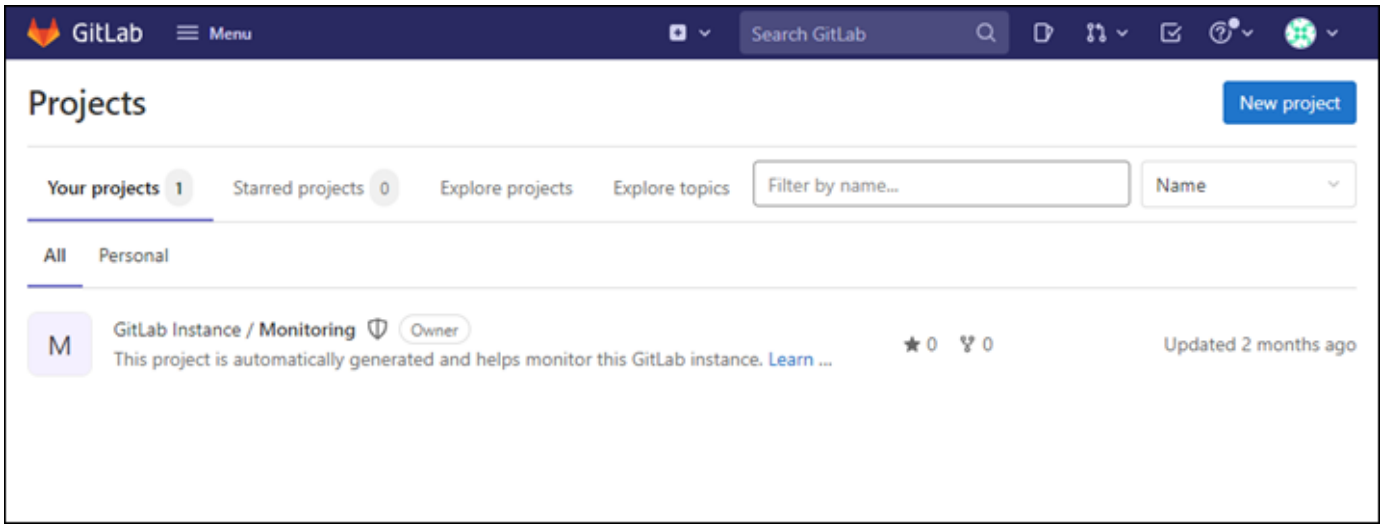
1. インスタンス管理ページの [Connect] (接続) タブにあるパブリック IP アドレスを書き留めます。パブリック IP アドレスは、インスタンス管理ページのヘッダーセクションにも表示されます。

2. インスタンスのパブリック IP アドレスを参照します (例: `http://203.0.113.0` に移動します)。

   GitLab CE ウェブサイトのホームページが表示されます。接続がプライベートではない、セキュリティで保護されていない、またはセキュリティ上のリスクがある、などの警告がブラウザに表示されることがあります。これは、GitLab CE インスタンスに SSL/TLS 証明書がまだ適用されていない場合に発生します。ブラウザウィンドウで、[Advanced] (詳細設定)、[Details] (詳細)、または [More information] (詳細情報) を選択して、使用可能なオプションを表示します。次に、プライベートまたは安全でない場合でも、ウェブサイトにアクセスすることを選択します。

3. デフォルトのユーザー名 (`root`) と、先ほど取得したデフォルトのパスワードを使用してサインインします。

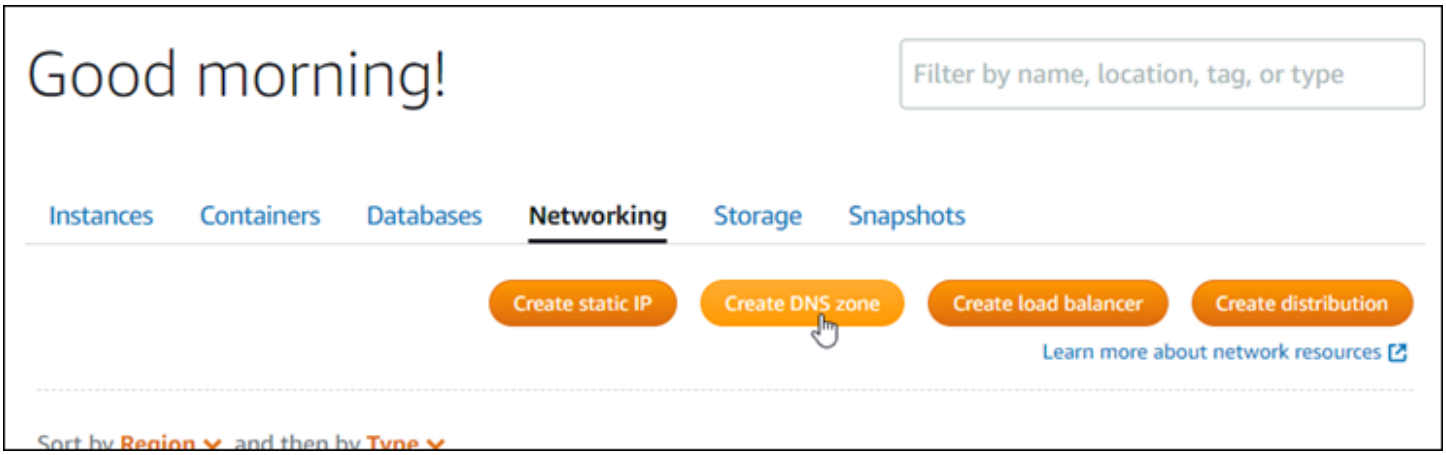
   Gitlab CE の管理ダッシュボードが表示されます。



ステップ 5: 登録済みドメイン名へのトラフィックを GitLab CE ウェブサイトに送信する

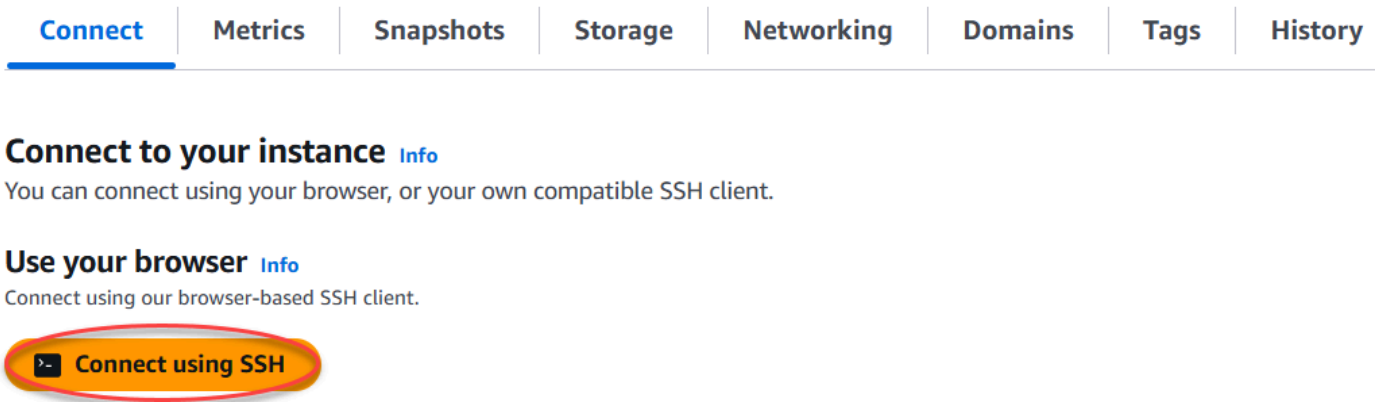
`example.com` などの登録済みドメイン名のトラフィックを GitLab CE ウェブサイトに送信するには、ドメインのドメインネームシステム (DNS) にレコードを追加します。DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページの [ネットワーク] タブで、[DNS ゾーンの作成] を選択し、ページに記載される手順に従います。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。



ドメイン名へのトラフィックがインスタンスにルーティングされたら、次の手順を実行して、GitLab CE にドメイン名を認識させます。

1. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。



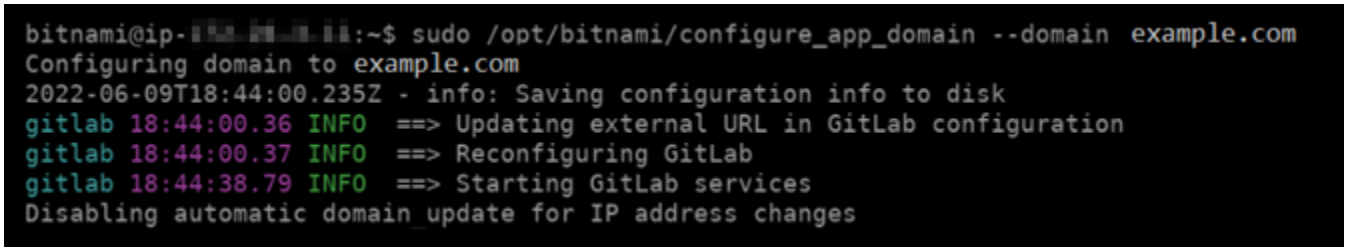
2. 接続後に、次のコマンドを入力します。*<DomainName>* は、インスタンスにトラフィックをルーティングするドメイン名に置き換えてください。

```
sudo /opt/bitnami/configure_app_domain --domain <DomainName>
```

例:

```
sudo /opt/bitnami/configure_app_domain --domain example.com
```

次の例のようなレスポンスが表示されます。これで、GitLab CE インスタンスがドメイン名を認識するようになります。



> **Note**
>
> このコマンドが失敗した場合、古いバージョンの GitLab CE インスタンスを使用している可能性があります。代わりに次のコマンドを実行してみてください。*<DomainName>* は、インスタンスにトラフィックをルーティングするドメイン名に置き換えてください。
>
> ```
> cd /opt/bitnami/apps/gitlab && \
> sudo ./bnconfig --machine_hostname <DomainName>
> ```
>
> 上記のコマンドを実行した後、次のコマンドを入力して、サーバーが再起動するたびに bnconfig ツールが自動的に実行されないようにします。
>
> ```
> sudo mv bnconfig bnconfig.disabled
> ```

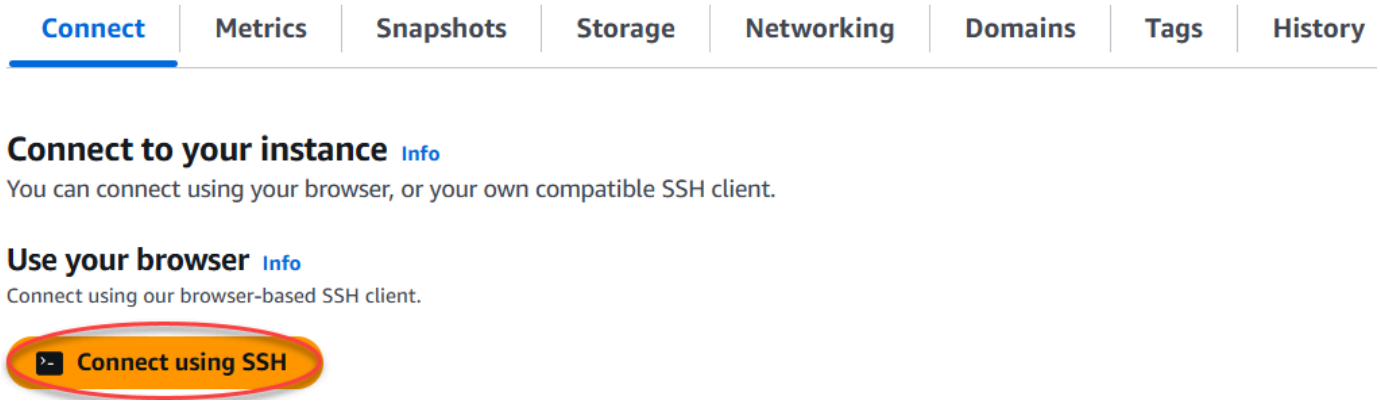
次に、SSL/TLS 証明書を生成して設定し、GitLab CE ウェブサイトの HTTPS 接続を有効にします。詳細については、本ガイドの次の「ステップ 6: GitLab CE ウェブサイトの HTTPS を設定する」のセクションを参照してください。

ステップ 6: GitLab CE ウェブサイトの HTTPS を設定する

GitLab CE ウェブサイトで HTTPS を設定するには、以下の手順を実行します。次の手順では、Lego クライアント の使い方を説明しています。これは、Let's Encrypt SSL/TLS 証明書を要求するコマンドラインツールです。

> **Important**
>
> この手順を開始する前に、GitLab CE インスタンスにトラフィックがルーティングされるようにドメインが設定済みであることを確認してください。設定されていない場合、SSL/TLS 証明書の検証プロセスが失敗します。登録したドメイン名のトラフィックをルーティングするために、ドメインの DNS にレコードを追加します。DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。
>
> Lightsail コンソールのホームページの [Domains & DNS] (ドメインと DNS) タブで、[Create DNS zone] (DNS ゾーンの作成) を選択し、ページに記載される手順に従います。詳細については、「Lightsail で DNS ゾーンを作成し、ドメインの DNS レコードを管理する」を参照してください。

1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。



2. 接続したら、次のコマンドを入力して、ディレクトリを一時ディレクトリ (/tmp) に変更します。

```
cd /tmp
```

3. 次のコマンドを入力して、Lego クライアントの最新バージョンをダウンロードします。このコマンドは、テープアーカイブ (tar) ファイルをダウンロードします。

```
curl -Ls https://api.github.com/repos/xenolf/lego/releases/latest | grep
 browser_download_url | grep linux_amd64 | cut -d '"' -f 4 | wget -i -
```

4. 次のコマンドを入力して tar ファイルからファイルを抽出します。*X.Y.Z* をダウンロードした Lego クライアントのバージョンに置き換えます。

```
tar xf lego_v X.Y.Z _linux_amd64.tar.gz
```

例:

```
tar xf lego_v4.7.0_linux_amd64.tar.gz
```

5. 以下のコマンドを入力して、Lego クライアントファイルを移動させる `/opt/bitnami/letsencrypt` ディレクトリを作成します。

```
sudo mkdir -p /opt/bitnami/letsencrypt
```

6. 以下のコマンドを入力して、Lego クライアントファイルを作成したディレクトリに移動します。

```
sudo mv lego /opt/bitnami/letsencrypt/lego
```

7. インスタンスで実行されているアプリケーションサービスを停止するには、次のコマンドを入力します。

```
sudo service bitnami stop && \
sudo service gitlab-runsvdir stop
```

8. 次のコマンドを入力して、Lego クライアントを使って Let's Encrypt SSL/TLS 証明書を要求します。

```
sudo /opt/bitnami/letsencrypt/lego --tls --email="EmailAddress" --
domains="RootDomain" --domains="WwwSubDomain" --path="/opt/bitnami/letsencrypt" run
```

コマンド内の次のサンプルテキストを独自のテキストに置き換えます。

- *EmailAddress* — 登録通知用のメールアドレスです。
- *RootDomain* — GitLab CE ウェブサイトにトラフィックをルーティングするプライマリルートドメイン (例: `example.com`) です。

- *WwwSubDomain* — GitLab CE ウェブサイトにトラフィックをルーティングするプライマリルートドメインの www サブドメイン (例: www.example.com) です。

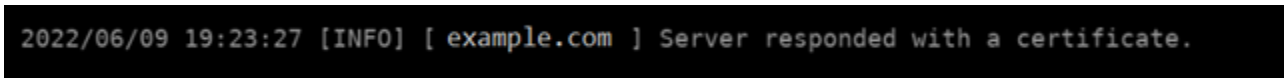
  コマンドに --domains パラメータを追加して指定することで、証明書に複数のドメインを指定することができます。複数のドメインを指定すると、Lego はサブジェクト別名 (SAN) 証明書を作成します。これにより、指定したすべてのドメインに対する有効な証明書が 1 つのみになります。リストの最初のドメインが証明書の「CommonName」として追加され、残りは「DNSNames」として証明書内の SAN 拡張に追加されます。

例:

```
sudo /opt/bitnami/letsencrypt/lego --tls --email="user@example.com" --
domains="example.com" --domains="www.example.com" --path="/opt/bitnami/letsencrypt"
 run
```

9. プロンプトが表示されたら、Y と Enter を押して利用規約に同意します。

   次の例に示すようなレスポンスが表示されます。



   成功した場合、一連の証明書が /opt/bitnami/letsencrypt/certificates ディレクトリに保存されます。このセットには、サーバー証明書ファイル (例: example.com.crt) とサーバ証明書キーファイル (例: example.com.key) が含まれています。

10. インスタンスの既存の証明書の名前を変更するには、次のコマンドを入力します。後で、これらの既存の証明書は新しい Let's Encrypt 証明書に置き換えます。

```
sudo mv /etc/gitlab/ssl/tls.crt /etc/gitlab/ssl/tls.crt.old && \
sudo mv /etc/gitlab/ssl/tls.key /etc/gitlab/ssl/tls.key.old
```

11. 次のコマンドを入力して、GitLab CE インスタンスのデフォルトの証明書ディレクトリである /etc/gitlab/ssl ディレクトリに新しい Let's Encript 証明書のシンボリックリンクを作成します。

```
sudo ln -sf /opt/bitnami/letsencrypt/certificates/Domain.key /etc/gitlab/ssl/
tls.key && \
sudo ln -sf /opt/bitnami/letsencrypt/certificates/Domain.crt /etc/gitlab/ssl/
tls.crt
```

コマンド内の *Domain* を Let's Encrypt 証明書を要求するときに指定したプライマリルートドメインに置き換えます。

例:

```
sudo ln -sf /opt/bitnami/letsencrypt/certificates/example.com.key /etc/gitlab/ssl/
tls.key && \
sudo ln -sf /opt/bitnami/letsencrypt/certificates/example.com.crt /etc/gitlab/ssl/
tls.crt
```

12. 次のコマンドを入力して、移動先のディレクトリにある新しい Let's Encrypt 証明書のアクセス許可を変更します。

```
sudo chown root:root /etc/gitlab/ssl/tls* && \
sudo chmod 600 /etc/gitlab/ssl/tls*
```

13. 次のコマンドを入力して、GitLab CE インスタンス上のアプリケーションサービスを再起動します。

```
sudo service bitnami start && \
sudo service gitlab-runsvdir start
```

次回に、設定したドメインを使用して GitLab CE ウェブサイトを参照するときには、HTTPS 接続にリダイレクトされるはずです。GitLab CE インスタンスが新しい証明書を認識するまでには、最大で 1 時間かかる可能性があります。GitLab CE ウェブサイトが接続を拒否した場合は、インスタンスを停止してから起動して、もう一度試してください。

ステップ 7: GitLab CE のドキュメントを読み、引き続きウェブサイトの設定を続行する

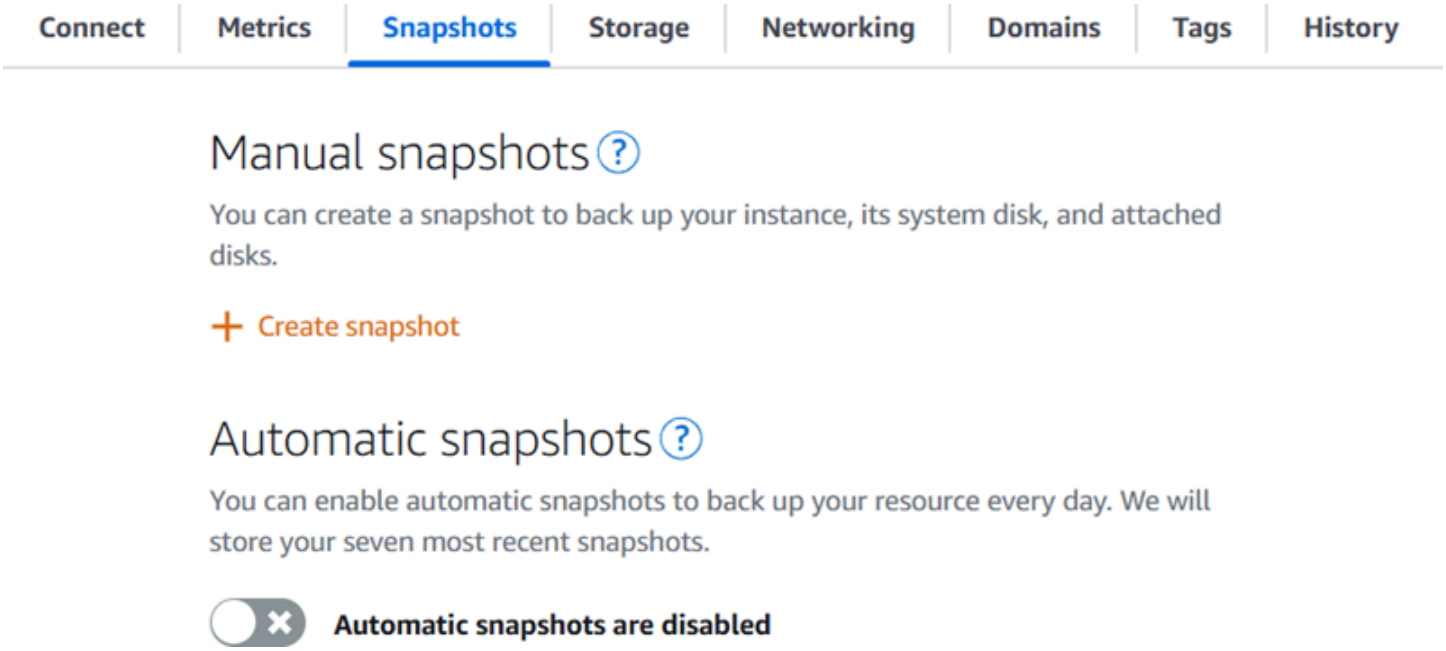
GitLab CE のドキュメントを読み、ウェブサイトを管理およびカスタマイズする方法を確認します。詳細については、「GitLab ドキュメント」を参照してください。

ステップ 8: インスタンスのスナップショットを作成する

GitLab CE ウェブサイトを希望どおりに設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生

した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。詳細については、「スナップショット」を参照してください。

インスタンス管理ページの [スナップショット] タブで [スナップショットを作成する] を選択するか、[自動スナップショットを有効にする] を選択します。



詳細については、「Amazon Lightsail で Linux または Unix インスタンスのスナップショットを作成」および「Amazon Lightsail のインスタンスまたはディスクの自動スナップショットの有効化と無効化」を参照してください。

Lightsail で Joomla! の使用を開始する

> 知っていましたか?
>
> Lightsail は、毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを自動的に最新のスナップショットに置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

Amazon Lightsail で起動した Joomla! インスタンスの使用を開始するステップについて説明します。

目次

- ステップ 1: Bitnami のドキュメントを確認する

- ステップ 2: Joomla! コントロールパネルにアクセスするためのデフォルトのアプリケーションパスワードを取得する
- ステップ 3: インスタンスに静的 IP アドレスをアタッチする
- ステップ 4: Joomla! ウェブサイトのコントロールパネルにサインインする
- ステップ 5: 登録済みドメイン名へのトラフィックを Joomla! ウェブサイトに送信する
- ステップ 6: Joomla! ウェブサイトの HTTPS を設定する
- ステップ 7: Joomla! のドキュメントを読み、引き続きウェブサイトの設定を続行する
- ステップ 8: インスタンスのスナップショットを作成する

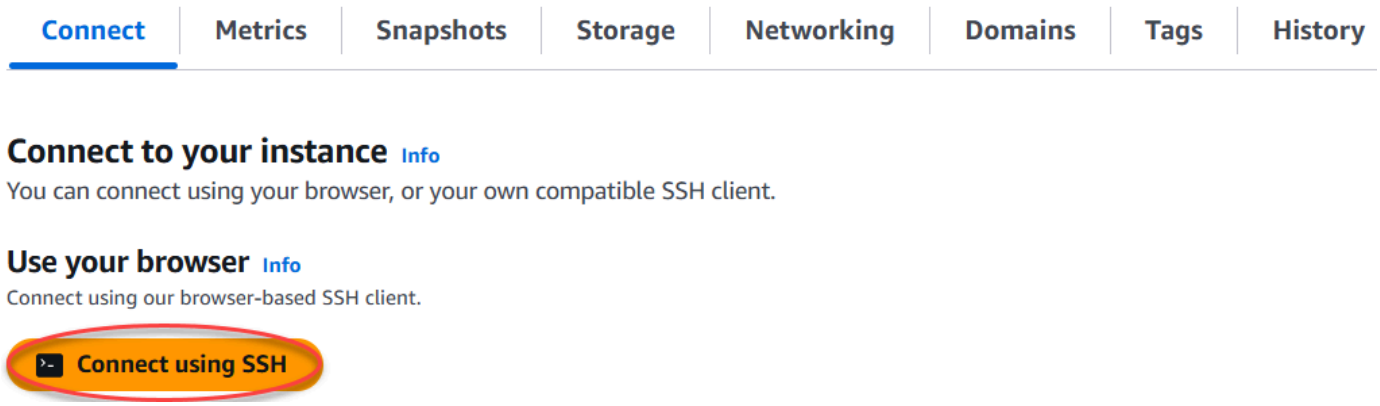
ステップ 1: Bitnami のドキュメントを確認する

Joomla! アプリケーションの設定方法については、Bitnami ドキュメントを参照してください。詳細については、Joomla! を参照してください。Bitnami によってパッケージ化 AWS クラウドされました。

ステップ 2: Joomla! コントロールパネルにアクセスするためのデフォルトのアプリケーションパスワードを取得する

次の手順を完了して、Joomla! ウェブサイトのコントロールパネルにアクセスする際に必要となるデフォルトのアプリケーションパスワードを取得します。詳細については、「Amazon LightsailのBitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。



2. 接続後に、次のコマンドを入力してアプリケーションのパスワードを取得します。

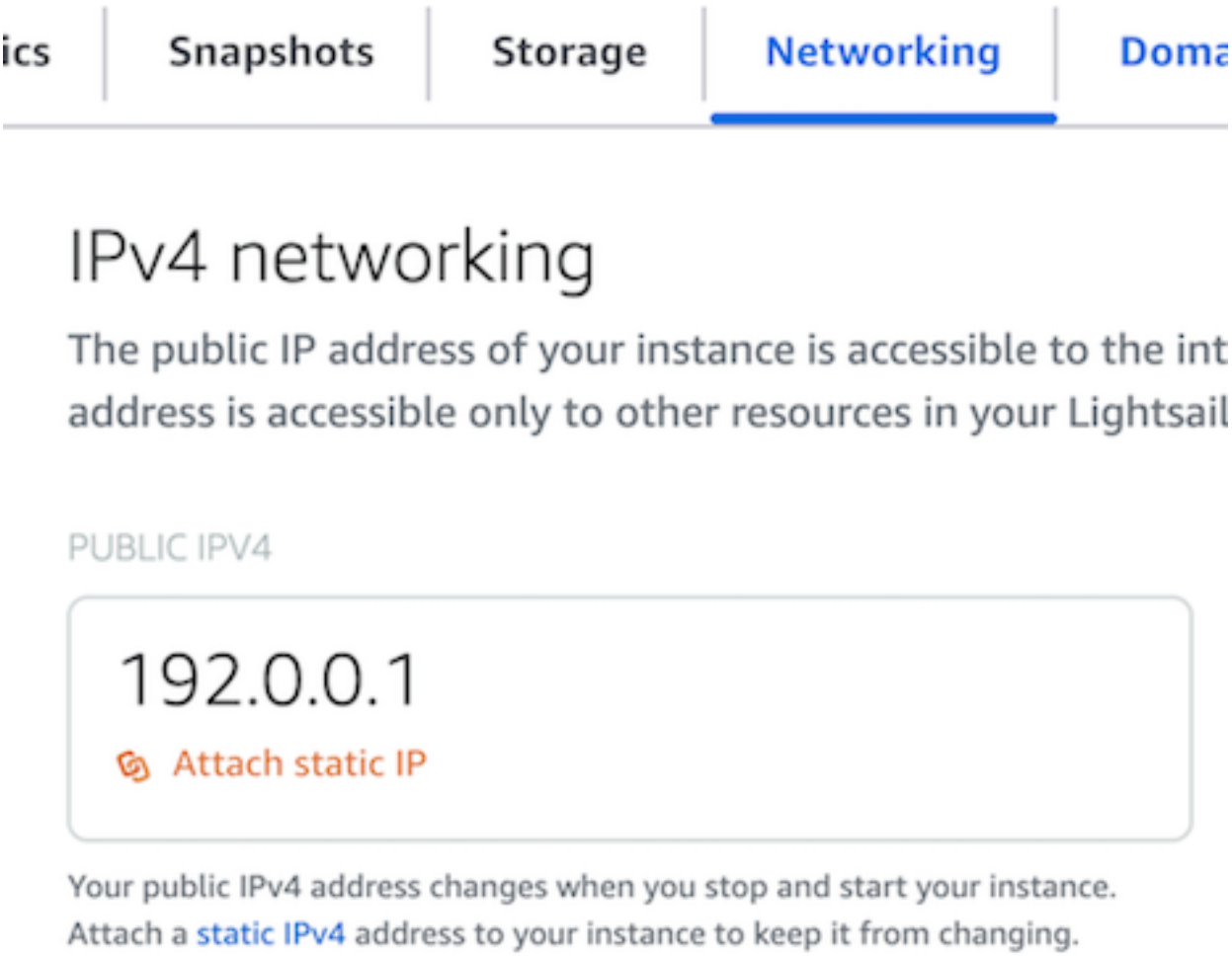
```
cat $HOME/bitnami_application_password
```

アプリケーションのデフォルトパスワードを含んだ、次の例のようなレスポンスが表示されます。

ステップ 3: インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



ステップ 4: Joomla! ウェブサイトのコントロールパネルにサインインする

デフォルトのユーザーパスワードを取得したら、以下の手順に従って Joomla! ウェブサイトのホームページに移動し、コントロールパネルにサインインします。サインイン後に、ウェブサイトをカスタマイズしたり管理上の変更を行うことができます。Joomla! で実行できる事項の詳細については、「ステップ 7: Joomla! のドキュメントを読み、引き続きウェブサイトの設定を続行する」のセクションを参照してください。

1. インスタンス管理ページの [Connect] (接続) タブにあるパブリック IP アドレスを書き留めます。パブリック IP アドレスは、インスタンス管理ページのヘッダーセクションにも表示されます。

2. インスタンスのパブリック IP アドレスを参照します (例: `http://203.0.113.0` に移動します)。

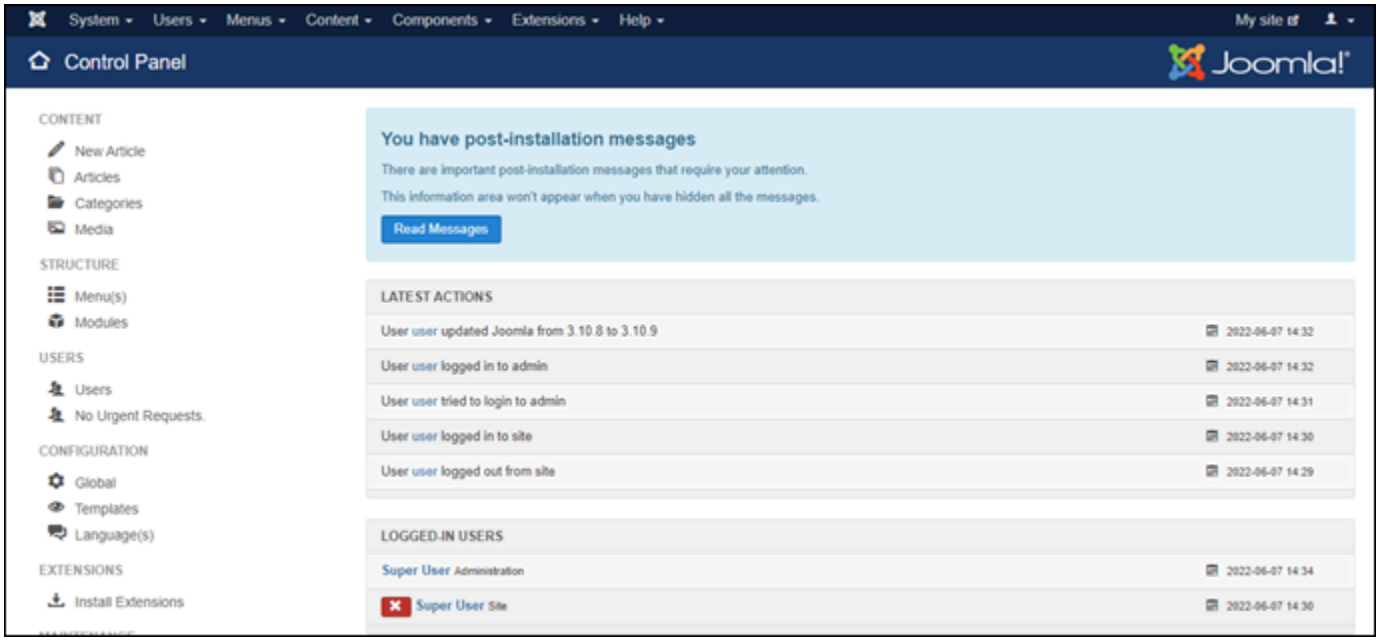
   Joomla! ウェブサイトのホームページが表示されます。

3. Joomla! ウェブサイトのホームページで、右下にある [Manage] (管理) を選択します。

   [Manage] (管理) バナーが表示されない場合は、`http://<PublicIP>/administrator/` を参照することでサインインページにアクセスすることができます。*<PublicIP>* を、インスタンスのパブリック IP アドレスに置き換えます。

4. デフォルトのユーザー名 (`user`) と、先ほど取得したデフォルトのパスワードを使用してサインインします。

   Joomla! の管理コントロールパネルが表示されます。



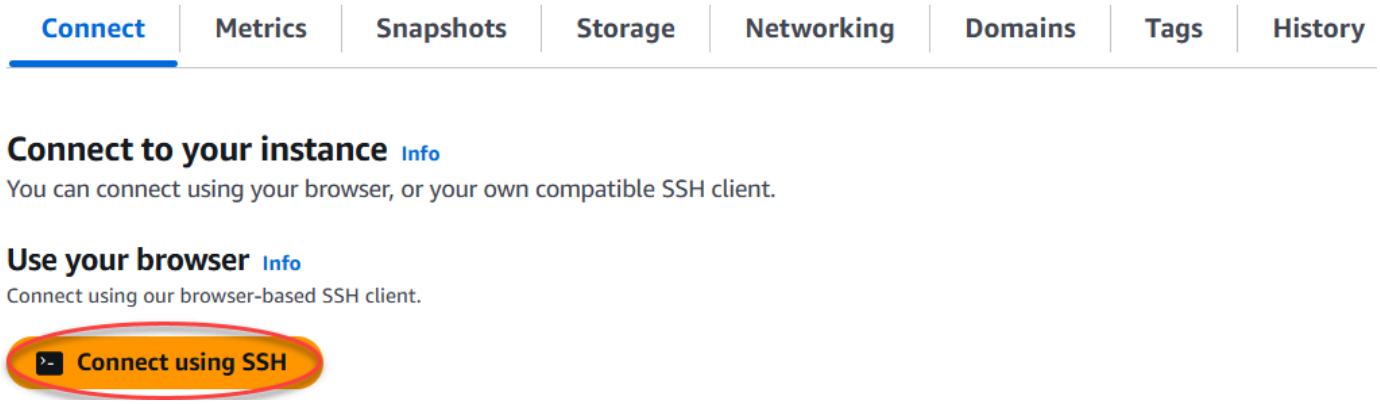
ステップ 5: 登録済みドメイン名へのトラフィックを Joomla! ウェブサイトに送信する

`example.com` などの登録済みドメイン名のトラフィックを Joomla! ウェブサイトに送信するには、ドメインのドメインネームシステム (DNS) にレコードを追加します。DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページの [Domains & DNS] (ドメインと DNS) タブで、[Create DNS zone] (DNS ゾーンの作成) を選択し、ページに記載される手順に従います。詳細については、「Lightsail で DNS ゾーンを作成し、ドメインの DNS レコードを管理する」を参照してください。

ドメイン名へのトラフィックがインスタンスにルーティングされたら、次の手順を実行して、Joomla! ソフトウェアにドメイン名を認識させます。

1. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。



2. Bitnami とは、多くのブループリントのファイル構造を変更するプロセスです。この手順にあるファイルパスは、Bitnami ブループリントがネイティブ Linux システムパッケージを使用しているか (アプローチ A)、または自己完結型インストール (アプローチ B) であるかによって変わる場合があります。Bitnami のインストールタイプと従うべき方法を特定するには、接続後に次のコマンドを実行します。

```
test ! -f "/opt/bitnami/common/bin/openssl" && echo "Approach A: Using system
 packages." || echo "Approach B: Self-contained installation."
```

3. 前のコマンドで得られる結果に、アプローチ A を使用すべきと示されている場合は次の手順を実行します。そうでない場合、前のコマンドで得られる結果にアプローチ B を使用すべきと示されている場合は、ステップ 4 に進みます。

   1. 以下のコマンドを入力して、Vim を使用して Apache 仮想ホスト設定ファイルを開き、ドメイン名の仮想ホストを作成します。

      ```
      sudo vim /opt/bitnami/apache2/conf/vhosts/joomla-vhost.conf
      ```

   2. I キーを押して Vim の挿入モードに移ります。
   3. 次の例に示されているように、ドメイン名をファイルに追加します。この例では、`example.com` および `www.example.com` ドメインを使用しています。

```
<VirtualHost 127.0.0.1:80 _default_:80>
  ServerName www.example.com
  ServerAlias example.com
  DocumentRoot /opt/bitnami/joomla
  <Directory "/opt/bitnami/joomla">
    Options -Indexes +FollowSymLinks -MultiViews
    AllowOverride None
    Require all granted
  </Directory>
  Include "/opt/bitnami/apache/conf/vhosts/htaccess/joomla-htaccess.conf"
</VirtualHost>
```

4. ESC キーを押して「:wq!」と入力し、編集内容を保存 (書き込んで) Vim を終了します。
5. 次のコマンドを入力して Apache サーバーを再起動します。

```
sudo /opt/bitnami/ctlscript.sh restart apache
```

4. 前のコマンドで得られる結果にアプローチ B を使用すべきと示された場合は、次の手順を実行します。

1. 以下のコマンドを入力して、Vim を使用して Apache 仮想ホスト設定ファイルを開き、ドメイン名の仮想ホストを作成します。

```
sudo vim /opt/bitnami/apps/joomla/conf/httpd-vhosts.conf
```

2. I キーを押して Vim の挿入モードに移ります。
3. 次の例に示されているように、ドメイン名をファイルに追加します。この例では、example.com および www.example.com ドメインを使用しています。

```
<VirtualHost *:80>
  ServerName example.com
  ServerAlias www.example.com
  ...
```

4. ESC キーを押して「:wq!」と入力し、編集内容を保存 (書き込んで) Vim を終了します。
5. 以下のコマンドを入力して、bitnami-apps-vhosts.conf ファイルに Joomla! の httpd-vhosts.conf ファイルが含まれていることを確認します。

```
sudo vim /opt/bitnami/apache2/conf/bitnami/bitnami-apps-vhosts.conf
```

ファイル内で、次の行を見つけます。ない場合には追加してください。

```
Include "/opt/bitnami/apps/joomla/conf/httpd-vhosts.conf"
```

6. 次のコマンドを入力して Apache サーバーを再起動します。

```
sudo /opt/bitnami/ctlscript.sh restart apache
```

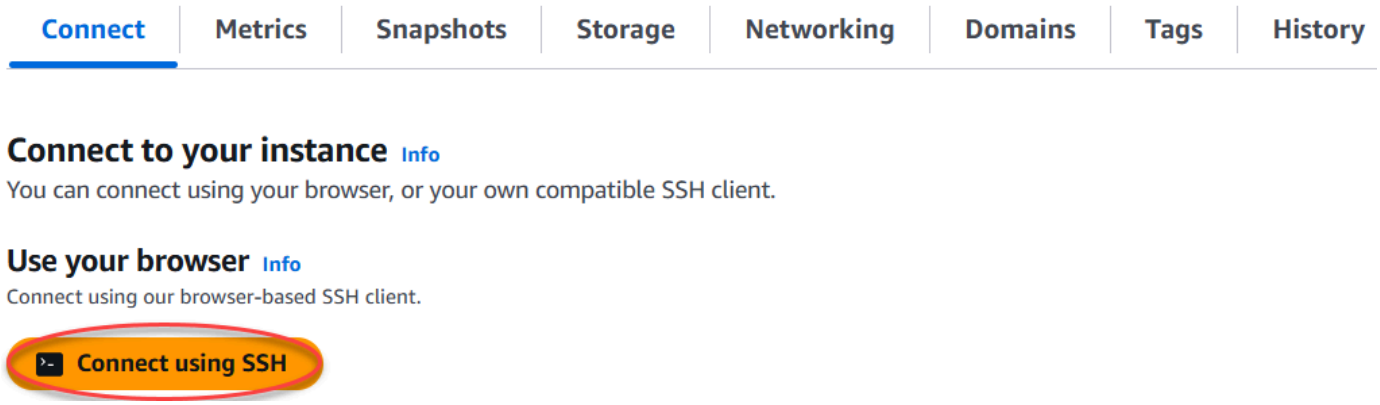
インスタンスに設定したドメイン名を参照すると、Joomla! ウェブサイトのホームページへと移動します。次に、SSL/TLS 証明書を生成して設定し、Joomla! ウェブサイトの HTTPS 接続を有効にします。詳細については、本ガイドの次の「ステップ 6: Joomla! ウェブサイトの HTTPS を設定する」のセクションを参照してください。

ステップ 6: Joomla! ウェブサイトの HTTPS を設定する

Joomla! ウェブサイトで HTTPS を設定するには、以下の手順を実行します。次の手順では、Bitnami HTTPS 設定ツール (`bncert-tool`) の使い方を説明しています。これは、Let's Encrypt SSL/TLS 証明書を要求するコマンドラインツールです。詳細については、「Bitnami ドキュメント」の「Bitnami HTTPS 設定ツールの詳細」を参照してください。

> **Important**
>
> この手順を開始する前に、Joomla! インスタンスにトラフィックがルーティングされるようにドメインが設定済みであることを確認してください。設定されていない場合、SSL/TLS 証明書の検証プロセスが失敗します。

1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。



2. 接続されたら、以下のコマンドを入力し、インスタンスに bncert ツールがインストールされていることを確認します。

```
sudo /opt/bitnami/bncert-tool
```

以下のレスポンスのいずれかが表示されます。

- レスポンスにコマンドが見つからないと表示された場合、bncert ツールがインスタンスにインストールされていないことを示しています。この手順の次のステップに進み、bncert ツールをインスタンスにインストールします。
- レスポンスに「Bitnami HTTPS 設定ツールへようこそ」と表示された場合は、インスタンスに bncert ツールがインストールされています。この手順のステップ 8 に進んでください。
- bncert ツールがインスタンスにインストールされてからしばらく経っている場合、ツールのアップデートバージョンが利用可能であることを示すメッセージが表示されることがあります。ダウンロードすることを選択し、`sudo /opt/bitnami/bncert-tool` コマンドを入力して bncert ツールを再度実行してください。この手順のステップ 8 に進んでください。

3. 以下のコマンドを入力して、bncert の実行ファイルをインスタンスにダウンロードします。

```
wget -O bncert-linux-x64.run https://downloads.bitnami.com/files/bncert/latest/
bncert-linux-x64.run
```

4. 以下のコマンドを入力して、インスタンスに bncert ツールの実行ファイル用のディレクトリを作成します。

```
sudo mkdir /opt/bitnami/bncert
```

5. 以下のコマンドを入力して、プログラムとして実行できるファイルを bncert に実行させます。

```
sudo chmod +x /opt/bitnami/bncert/bncert-linux-x64.run
```

6. 次のコマンドを入力して、sudo /opt/bitnami/bncert-tool コマンドを入力すると bncert ツールを実行するシンボリックリンクを作成します。

```
sudo ln -s /opt/bitnami/bncert/bncert-linux-x64.run /opt/bitnami/bncert-tool
```

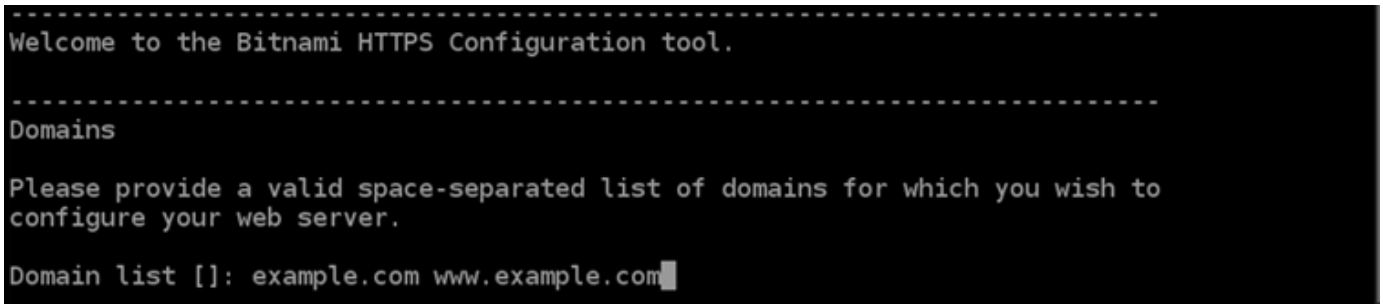
これでインスタンスに bncert ツールをインストールする手順は完了です。

7. 次のコマンドを入力して、bncert ツールを実行しましょう。

```
sudo /opt/bitnami/bncert-tool
```

8. 次の例に示すように、プライマリドメイン名と代替ドメイン名の間はスペースで区切って入力します。

   ドメインがインスタンスのパブリック IP アドレスにトラフィックをルーティングするように設定されていない場合、bncert ツールは、続行する前にその設定を行うように要求します。ドメインは、bncert ツールを使用して HTTPS を有効にしているインスタンスでのパブリック IP アドレスにトラフィックをルーティングする必要があります。これはドメインを所有していることを確認し、証明書の検証として機能します。



9. bncert ツールは、ウェブサイトのリダイレクトの設定方法を尋ねます。使用できるオプションは次のとおりです。

   - HTTP から HTTPS へのリダイレクトを有効にする - HTTP バージョンのウェブサイトを閲覧するユーザー (例: http:/example.com) を自動的に HTTPS バージョン (例: https://example.com) にリダイレクトするかどうかを決定します。すべての訪問者が暗号化された接続を使用するように強制されるため、このオプションを有効にすることをお勧めします。Y を入力して Enter を押すと、有効になります。
   - www なしから www ありへのリダイレクトの有効化 - ドメインの頂点 (例: https://example.com) まで閲覧するユーザー を自動的にドメインの www サブドメイン (例: https://www.example.com) にリダイレクトするかを指定します。このオプションを有効にすることをお勧めします。ただし、ドメインの頂点を Google のウェブマスターツールなどの検索エンジンツールで希望のウェブサイトアドレスとして指定した場合、または頂点が IP を直接指しており、www のサブドメインが CNAME レコードを介してリファレンスしている場合は、無効にして代替オプションを有効にすることをお勧めします (www ありから www なしへのリダイレクトを有効化) 。Y を入力し、Enter を押して有効にします。
   - www ありから www なしへのリダイレクトを有効にする - ドメインの www サブドメイン (例: https://www.example.com) まで閲覧するユーザーを、自動的にドメインの頂点 (例: https://example.com) にリダイレクトするかを指定します。www なしから www ありへのリダイレクトを有効にした場合は、これを無効にすることをお勧めします。N を入力し、Enter を押して無効にします。

選択した結果は次の例のようになります。

```
Enable/disable redirections

Please select the redirections you wish to enable or disable on your Bitnami
installation.



Enable HTTP to HTTPS redirection [Y/n]: Y




Enable non-www to www redirection [Y/n]: Y




Enable www to non-www redirection [y/N]: N
```

10. これから実行される変更が一覧表示されます。Y と入力し、Enter を押して確認し、続行します。

```
Changes to perform

The following changes will be performed to your Bitnami installation:

1. Stop web server
2. Configure web server to use a free Let's Encrypt certificate for the domains:
example.com www.example.com
3. Configure a cron job to automatically renew the certificate each month
4. Configure web server name to: example.com
5. Enable HTTP to HTTPS redirection (example: redirect http://example.com to
https://example.com)
6. Enable non-www to www redirection (example: redirect example.com to
 www.example.com)
7. Start web server once all changes have been performed



Do you agree to these changes? [Y/n]: Y
```

11. Let's Encrypt 証明書に関連付けるメールアドレスを入力し、Enter を押します。

```
Create a free HTTPS certificate with Let's Encrypt

Please provide a valid e-mail address for which to associate your Let's Encrypt
certificate.

Domain list: example.com www.example.com

Server name: example.com

E-mail address []:
```

12. Let's Encrypt サブスクライバー合意書を確認します。Y と入力し、Enter を押して契約に同意し、続行します。

```
The Let's Encrypt Subscriber Agreement can be found at:

https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf

Do you agree to the Let's Encrypt Subscriber Agreement? [Y/n]:
```

これらのアクションは、証明書のリクエストや指定したリダイレクトの設定など、インスタンスで HTTPS を有効にするために実行されます。

```
Performing changes to your installation

The Bitnami HTTPS Configuration Tool will perform any necessary actions to your
Bitnami installation. This may take some time, please be patient.
```

次の例のようなメッセージが表示された場合は、証明書は正常に発行され、検証され、インスタンスでリダイレクトが正常に設定されています。

```
Success

The Bitnami HTTPS Configuration Tool succeeded in modifying your installation.

The configuration report is shown below.

Backup files:
* /opt/bitnami/apache2/conf/httpd.conf.back.202005290035
* /opt/bitnami/apache2/conf/bitnami/bitnami-apps-prefix.conf.back.202005290035
* /opt/bitnami/apache2/conf/bitnami/bitnami.conf.back.202005290035

Find more details in the log file:

/tmp/bncert-202005290035.log

If you find any issues, please check Bitnami Support forums at:

https://community.bitnami.com

Press [Enter] to continue:
```

bncert ツールは、有効期限が切れる前、 80 日ごとに証明書の自動更新を実行します。インスタンスで追加のドメインやサブドメインを使用し、それらのドメインで HTTPS を有効にする場合は、上記のステップを繰り返します。

これで、Joomla! インスタンスでの HTTPS の有効化が完了しました。次回に、設定したドメインを使用して Joomla! ウェブサイトを参照するときには、HTTPS 接続にリダイレクトされるはずです。

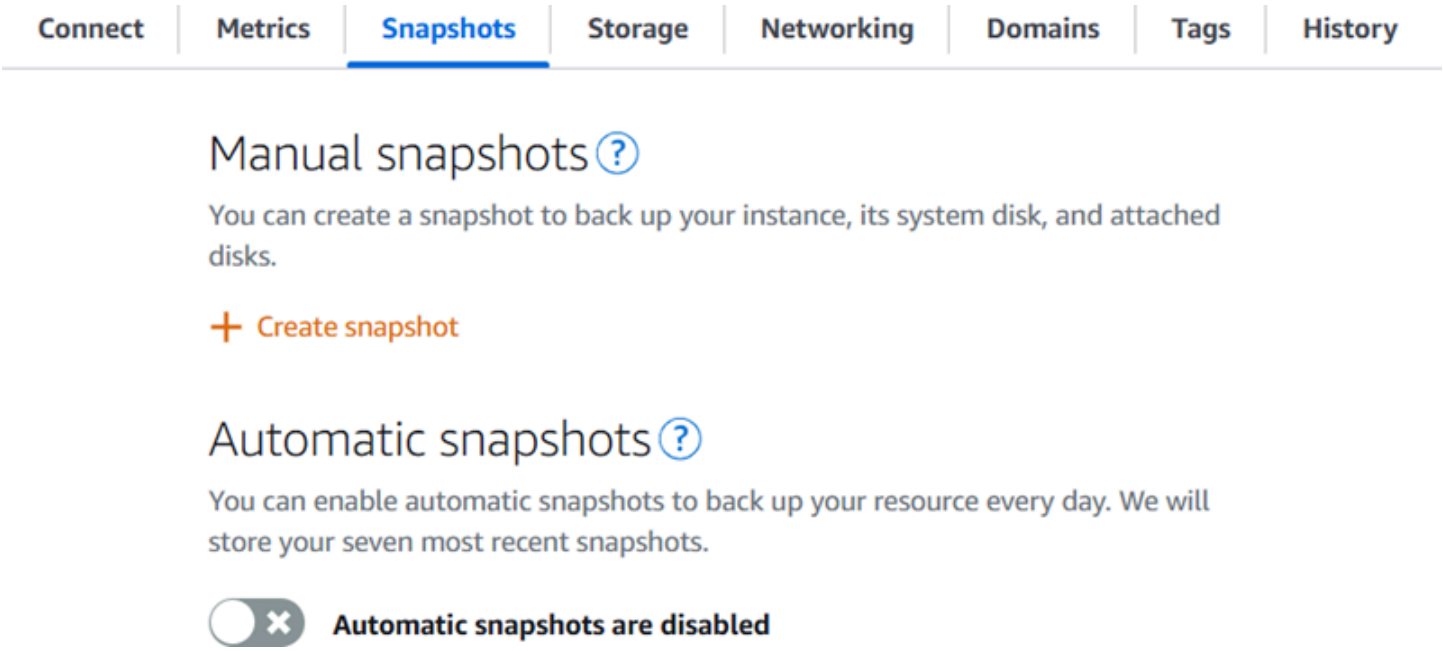
ステップ 7: Joomla! のドキュメントを読み、引き続きウェブサイトの設定を続行する

Joomla! のドキュメントを読み、ウェブサイトを管理およびカスタマイズする方法を確認します。詳細については、Joomla! を参照してください。ドキュメント

ステップ 8: インスタンスのスナップショットを作成する

Joomla! ウェブサイトを希望どおりに設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。詳細については、「スナップショット」を参照してください。

インスタンス管理ページの [スナップショット] タブで [スナップショットを作成する] を選択するか、[自動スナップショットを有効にする] を選択します。

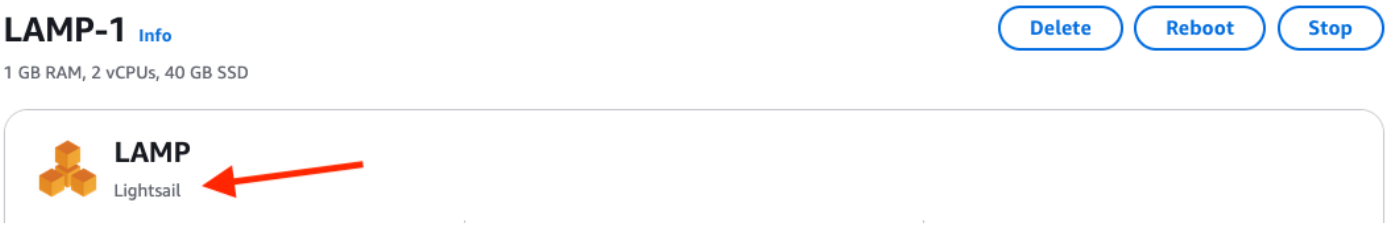


詳細については、「Amazon Lightsail で Linux または Unix インスタンスのスナップショットを作成」および「Amazon Lightsail のインスタンスまたはディスクの自動スナップショットの有効化と無効化」を参照してください。

での LAMP スタックのデプロイと管理 Lightsail

> **知っていましたか?**
>
> Lightsail は毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを最新のスナップショットに自動的に置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

LAMP インスタンスが で起動および実行された後に開始するには、いくつかのステップを実行する必要がありますAmazon Lightsail。開始する前に、インスタンス管理ページでブループリントベンダーを特定します。



LAMP インスタンスに適したガイドを選択します。

Bitnami

ステップ 1: LAMP インスタンスのデフォルトのアプリケーションパスワードを取得する

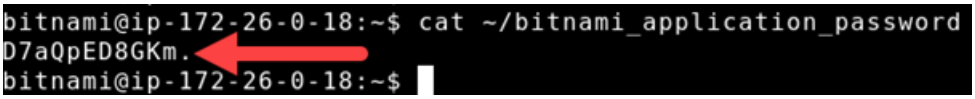
インスタンスにプリインストールされているアプリケーションやサービスにアクセスするには、アプリケーションのデフォルトパスワードが必要です。

1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。
2. 接続後に、次のコマンドを入力してアプリケーションのパスワードを取得します。

```
cat bitnami_application_password
```

> **Note**
>
> ユーザーのホームディレクトリ以外のディレクトリで作業している場合は、「`cat $HOME/bitnami_application_password`」と入力します。

次のようなレスポンスにアプリケーションのデフォルトパスワードが表示されます。

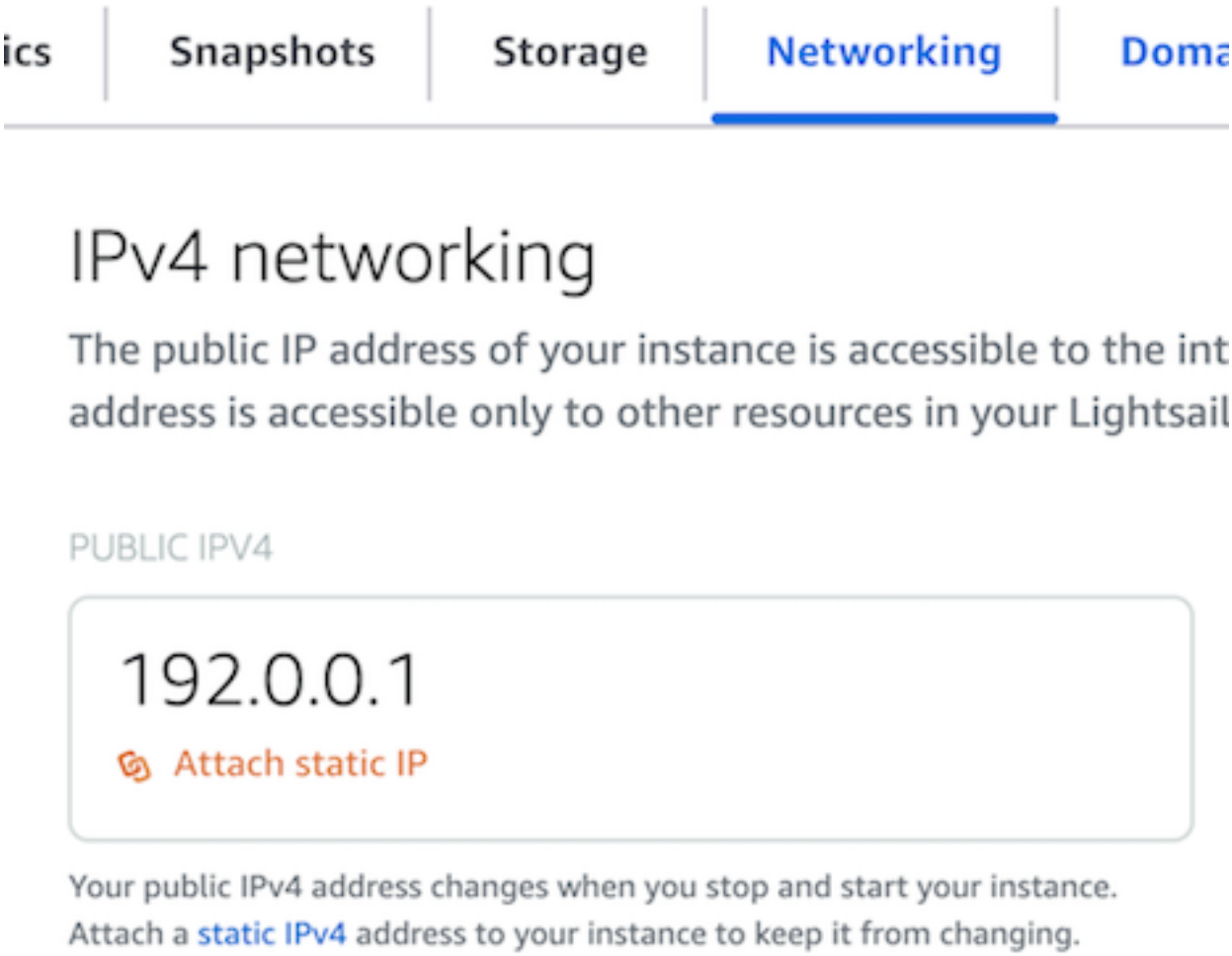


詳細については、「Amazon Lightsailの Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

ステップ 2: LAMP インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



ステップ 3: LAMP インスタンスのウェルカムページにアクセスする

インスタンスのパブリック IP アドレスに移動してインスタンスにインストールされているアプリケーションにアクセスし、phpMyAdmin または Bitnami ドキュメントを参照します。

1. インスタンス管理ページの [接続] タブで、パブリック IP を書き留めます。
2. パブリック IP アドレスを参照します (例: `http://192.0.2.3` に移動します)。

詳細については、「Amazon Lightsailの Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

ステップ 4: ドメイン名を LAMP インスタンスにマッピングする

ドメイン名 (`example.com` など) をインスタンスにマッピングするには、ドメインのドメインネームシステム (DNS) にレコードを追加します。DNS レコードは、通常、ドメインの登録

先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページのネットワークタブで、DNS ゾーンランプの作成を選択し、ページの指示に従います。

詳細については、「ドメインの DNS レコードを管理する DNS ゾーンを作成する」を参照してください。

ステップ 5: Bitnami のドキュメントを確認する

Bitnami ドキュメントを読んで、アプリケーションのデプロイ、SSL 証明書による HTTPS サポートの有効化、SFTP によるサーバーへのファイルのアップロードなどを行う方法を確認してください。

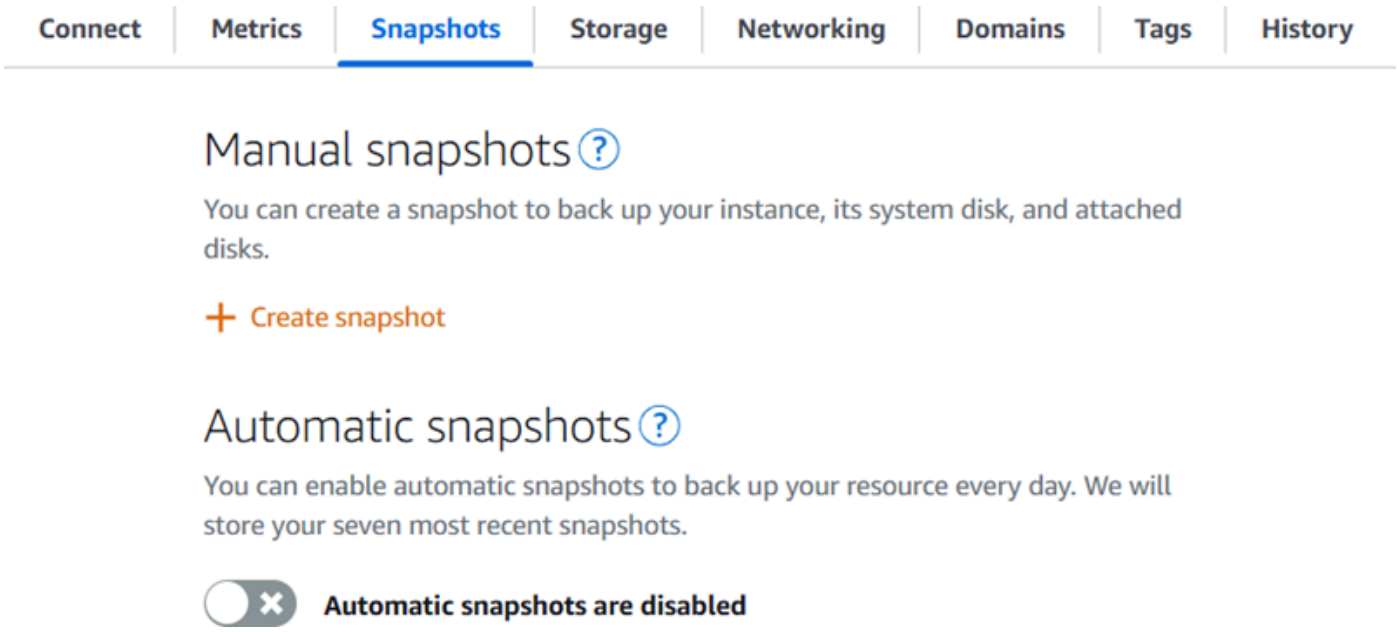
詳細については、「AWS クラウド用の Bitnami LAMP」を参照してください。

ステップ 6: LAMP インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。



Lightsail

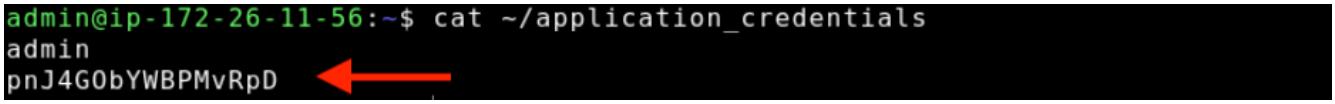
ステップ 1: LAMP インスタンスのデフォルトのアプリケーションパスワードを取得する

インスタンスにプリインストールされているアプリケーションやサービスにアクセスするには、アプリケーションのデフォルトパスワードが必要です。

1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。
2. 接続後に、次のコマンドを入力してデフォルトのアプリケーションのパスワードを取得します。

```
cat ~/application_credentials
```

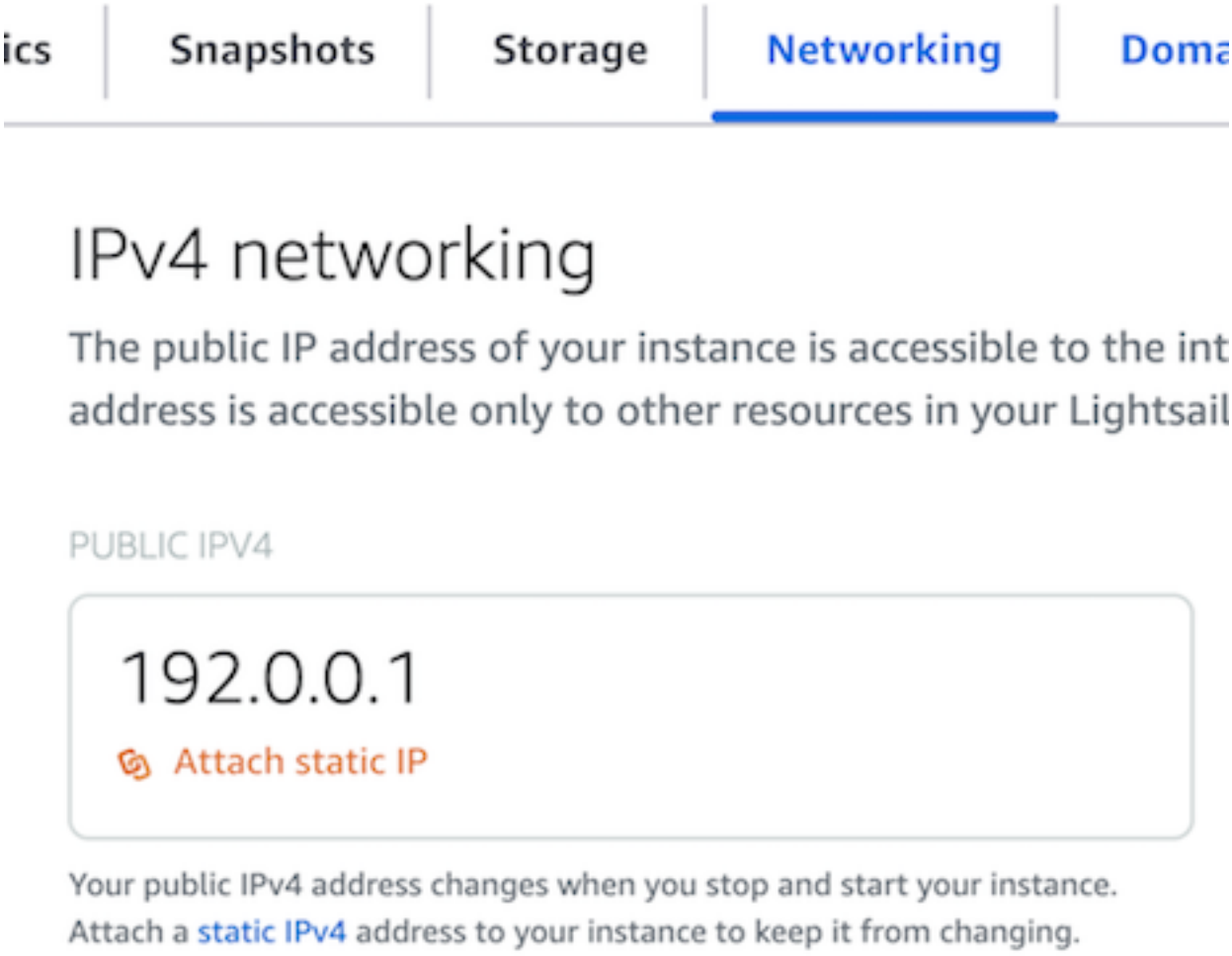
次のようなレスポンスにアプリケーションのデフォルトパスワードが表示されます。



ステップ 2: LAMP インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



ステップ 3: LAMP インスタンスのウェルカムページにアクセスする

インスタンスの静的 IP アドレスに移動して、インスタンスにインストールされているアプリケーションにアクセスします。

1. インスタンス管理ページで、静的 IP アドレスをコピーします。

2. 静的 IP アドレスをブラウザアドレスに貼り付けます。例: `http://192.0.0.1`。

ステップ 4: ドメイン名を LAMP インスタンスにマッピングする

ドメイン名 (`example.com` など) をインスタンスにマッピングするには、ドメインのドメインネームシステム (DNS) にレコードを追加します。DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページで、[ネットワーキング] タブの [DNS ゾーンの作成] を選択し、ページに表示される手順に従います。

詳細については、「ドメインの DNS レコードを管理する DNS ゾーンを作成する」を参照してください。

HTTPS を有効にするには、「Let's Encrypt Lightsail SSL/TLS を使用して LAMP ウェブサイトを保護する」を参照してください。

ステップ 5: アプリケーションをデプロイする

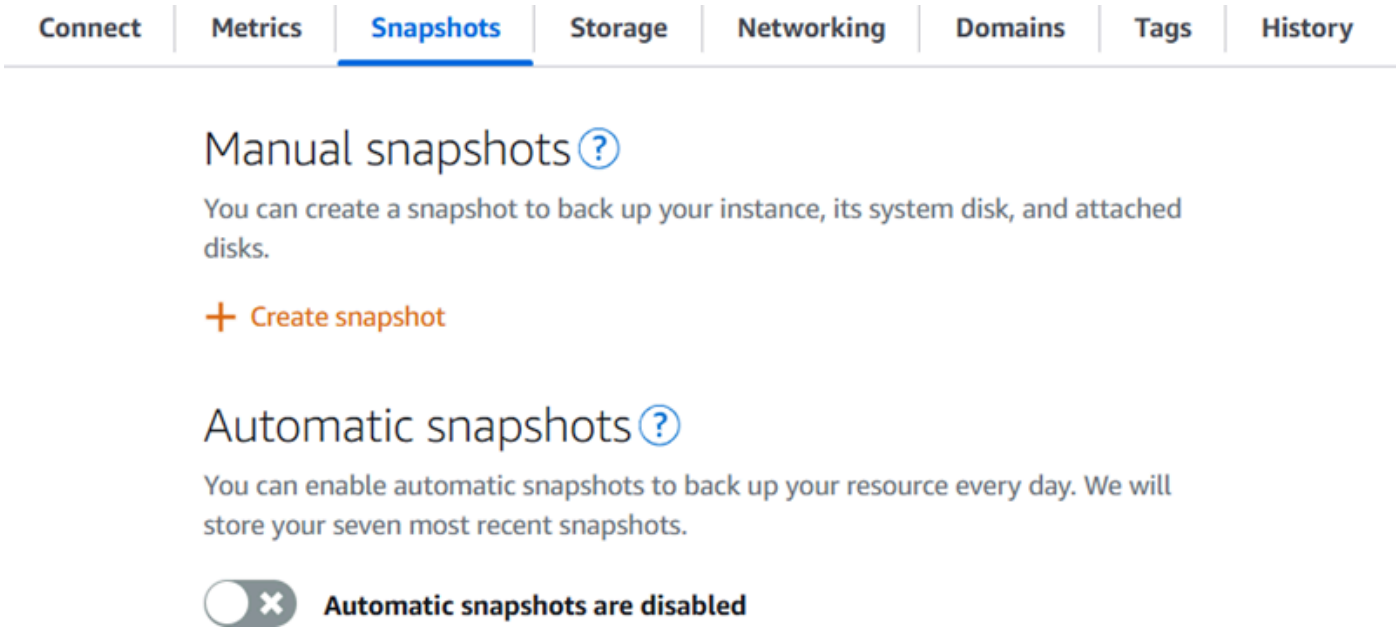
1. 「scp Lightsailを使用してアプリケーションを にコピーする」の「Linux インスタンス間でファイルを転送する」の手順に従います。 `/var/www/html`
2. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。
3. `sudo systemctl restart apache2` を実行する
4. インスタンスの静的 IP アドレスに移動する

ステップ 6: LAMP インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。



Lightsail で Magento を設定および構成する

> **知っていましたか?**
>
> Lightsail は、毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを自動的に最新のスナップショットに置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

Amazon Lightsail で起動した Magento インスタンスの使用を開始するステップについて説明します。

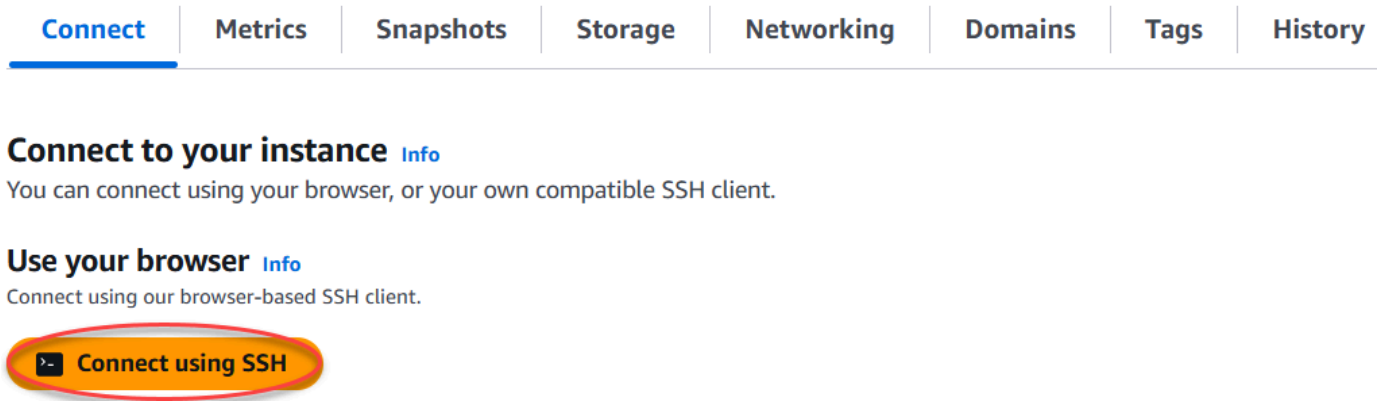
目次

- ステップ 1: Magento ウェブサイトのデフォルトのアプリケーションパスワードを取得する
- ステップ 2: Magento インスタンスに静的 IP アドレスをアタッチする
- ステップ 3: Magento ウェブサイトの管理ダッシュボードにサインインする
- ステップ 4: 登録済みドメイン名へのトラフィックを Magento ウェブサイトに送信する
- ステップ 5: Magento ウェブサイトの HTTPS を設定する
- ステップ 6: メール通知用の SMTP を設定する
- ステップ 7: Bitnami と Magento のドキュメントを読む
- ステップ 8: Magento インスタンスのスナップショットを作成する

ステップ 1: Magento ウェブサイトのデフォルトのアプリケーションパスワードを取得する

次のステップを完了して、Magento ウェブサイトのデフォルトのアプリケーションパスワードを取得します。詳細については、「Amazon Lightsailの Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

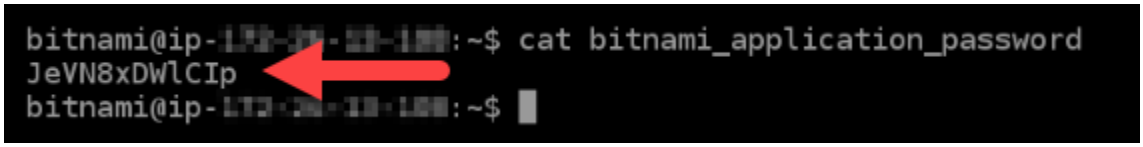
1. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。



2. 接続後に、次のコマンドを入力してデフォルトのアプリケーションのパスワードを取得します。

```
cat $HOME/bitnami_application_password
```

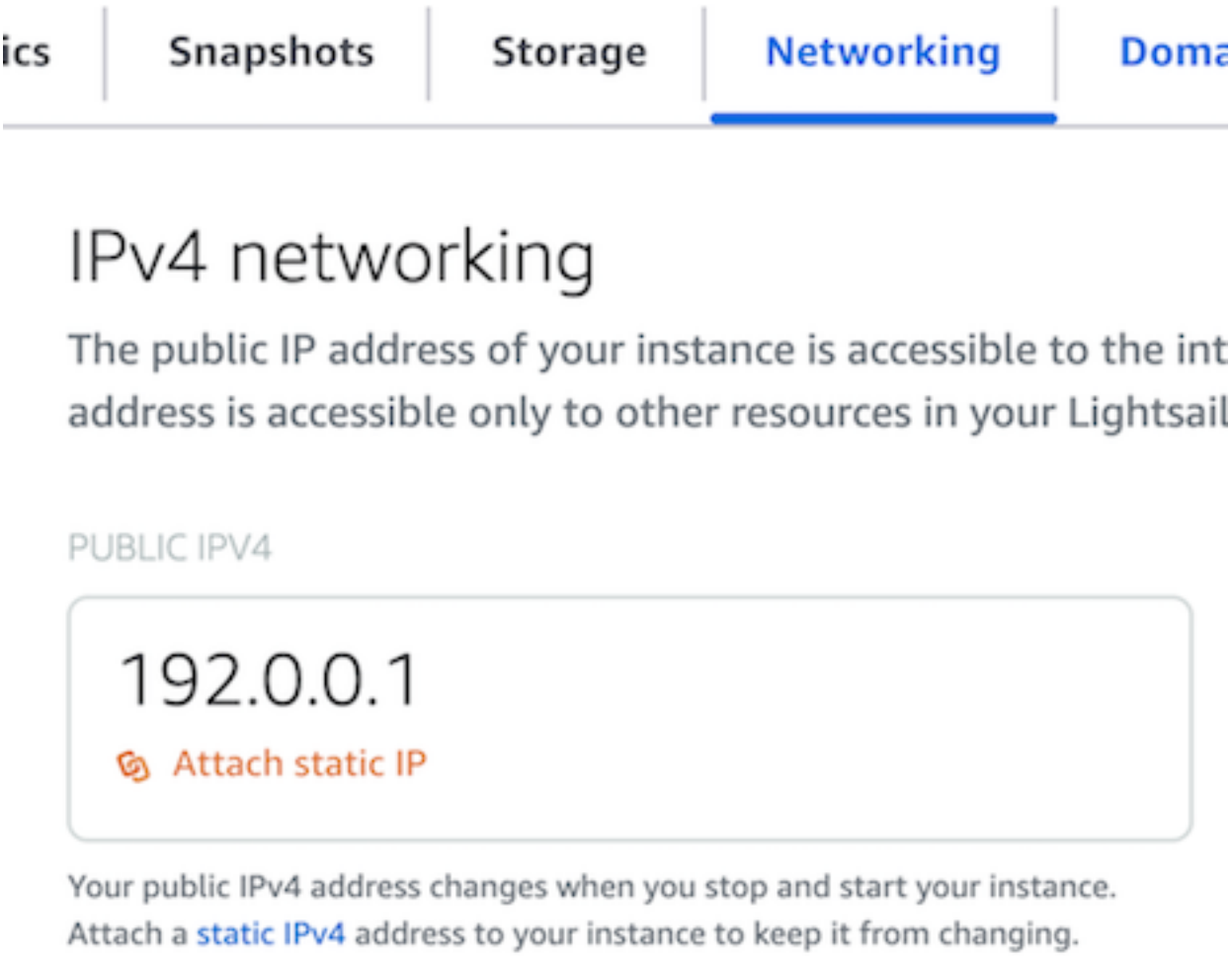
アプリケーションのデフォルトパスワードを含んだ、次の例のようなレスポンスが表示されます。このパスワードを安全な場所に保存します。このチュートリアルの次のセクションで、Magento ウェブサイトの管理ダッシュボードにサインインする際に使用します。



ステップ 2: Magento インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

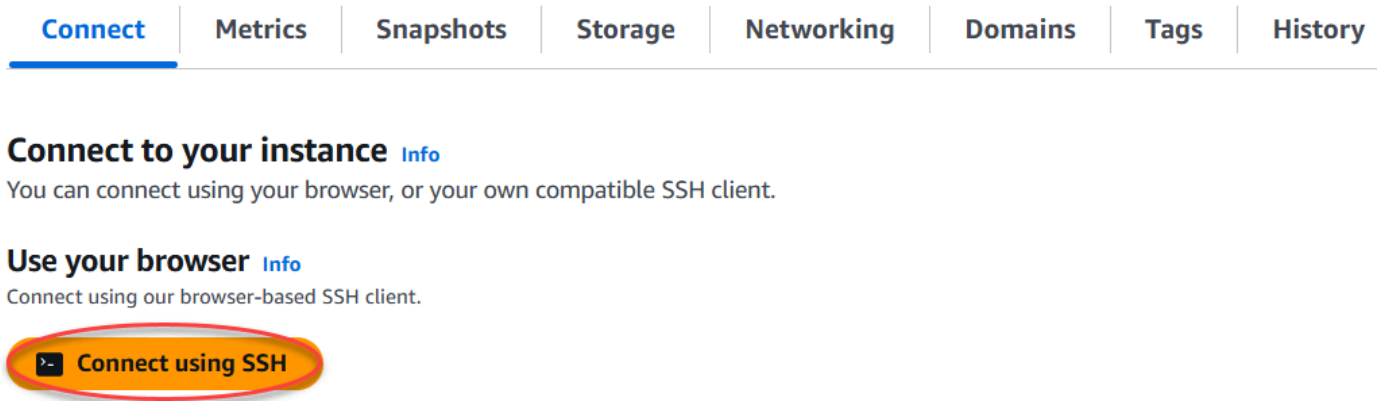
インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



新しい静的 IP アドレスがインスタンスに添付されたら、次の手順を実行して、Magento ソフトウェアに新しい静的 IP アドレスを認識させる必要があります。

1. インスタンスの静的 IP アドレスは書き留めておきます。このIP アドレスはインスタンス管理ページの ヘッダーセクションに表示されます。

2. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。



3. 接続後に、次のコマンドを入力します。*<StaticIP>* はインスタンスの新しい静的 IP アドレス必ず置き換えてください。

```
sudo /opt/bitnami/configure_app_domain --domain <StaticIP>
```

例:

```
sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
```

次の例のようなレスポンスが表示されます。これで、Magento ソフトウェアは新しい静的 IP アドレスを認識するようになります。

```
bitnami@ip-[redacted]:~$ sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
Configuring domain to 203.0.113.0
2021-03-12T15:49:22.000Z - info: Saving configuration info to disk
prestashop 15:49:22.41 INFO  ==> Trying to connect to the database server
prestashop 15:49:22.44 INFO  ==> Updating hostname in database
prestashop 15:49:22.46 INFO  ==> Purging cache
Disabling automatic domain update for IP address changes
```

> **Note**
>
> Magento は現在、IPv6 アドレスをサポートしていません。インスタンスの IPv6 を有効にすることはできますが、Magento ソフトウェアは IPv6 ネットワーク経由のリクエストに応答しません。

ステップ 3: Magento ウェブサイトの管理ダッシュボードにサインインする

Magento ウェブサイトにアクセスして管理ダッシュボードにサインインするには、以下のステップを実行します。サインインするには、このガイドの前のセクションで取得したデフォルトのユーザー名 (user) とデフォルトのアプリケーションパスワードを使用します。

1. Lightsail コンソールで、インスタンス管理ページのヘッダー部分に記載されているパブリック IP アドレスないし静的 IP アドレスを書き留めます。

2. 以下のアドレスまで移動して、Magento ウェブサイトの管理ダッシュボードのサインインページにアクセスします。*PublicIPAddress* は、インスタンスのパブリック IP アドレスないし静的 IP アドレスに必ず置き換えます。

```
http://<InstanceIpAddress>/admin
```

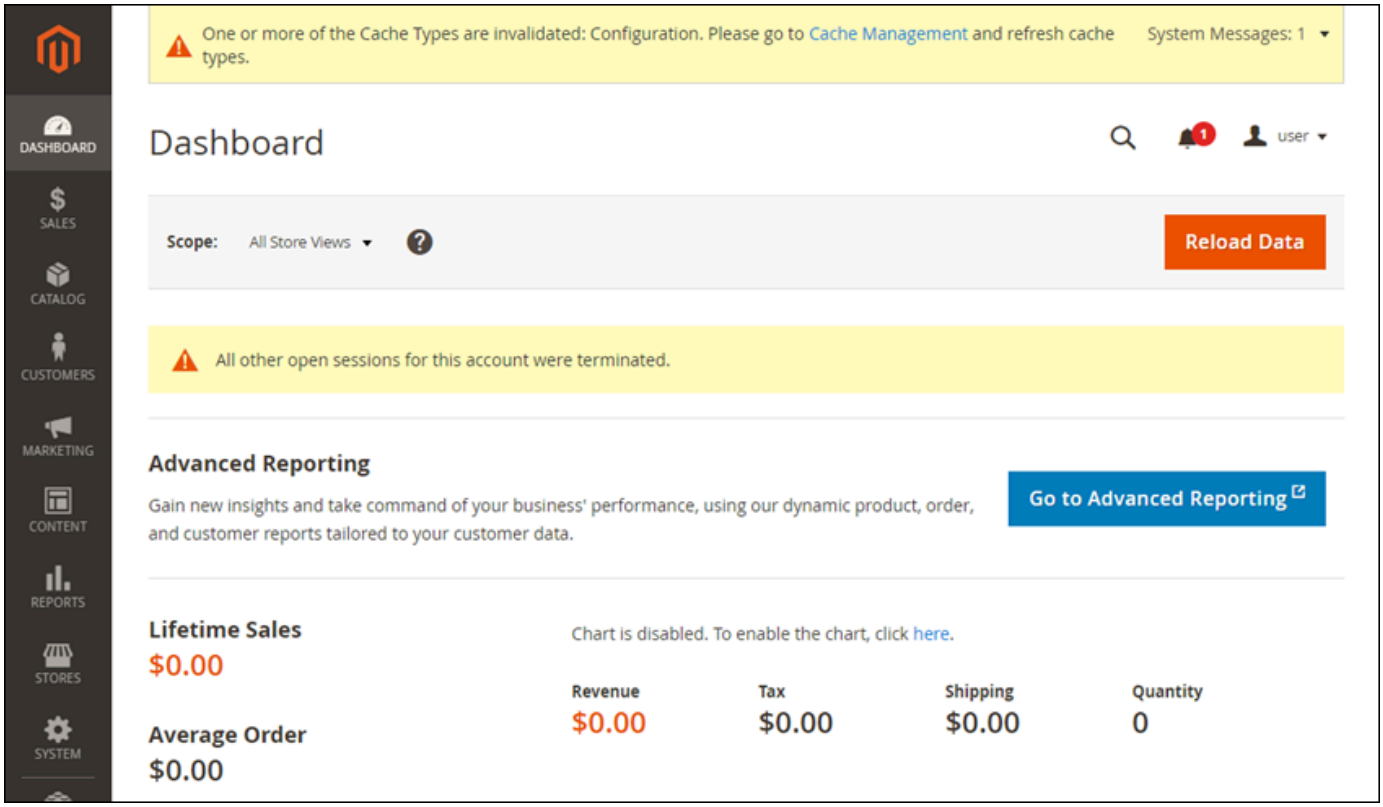
例:

```
http://203.0.113.0/admin
```

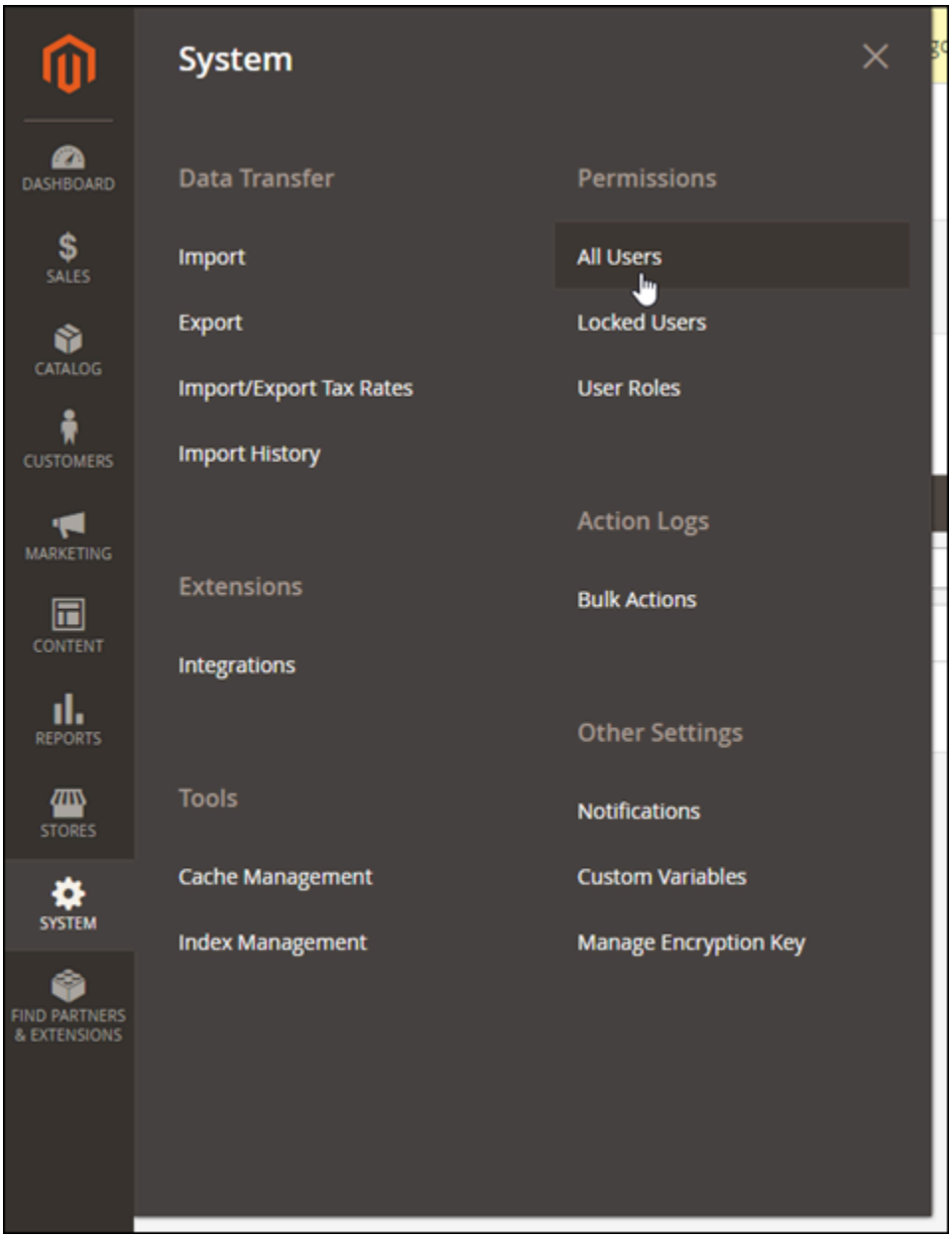
> **Note**
>
> Magento 管理用ダッシュボードのサインインページにアクセスできない場合、インスタンスを再起動する必要があるかもしれません。

3. デフォルトのユーザー名 (user) と、このガイドの前半のセクションで取得したデフォルトのアプリケーションパスワードを入力して [Sign in] (サインイン) を選択します。

Magento の管理ダッシュボードが表示されます。



Magento ウェブサイトの管理ダッシュボードへサインインする際に使用するデフォルトのユーザー名またはパスワードを変更するには、ナビゲーションペインの [System] (システム) を選択し、[All Users] (すべてのユーザー) を選択します。詳細については、「Magento ドキュメント」の「ユーザーの追加」を参照してください。



管理ダッシュボードの詳細については、「Magento 2.4 ユーザーガイド」を参照してください。

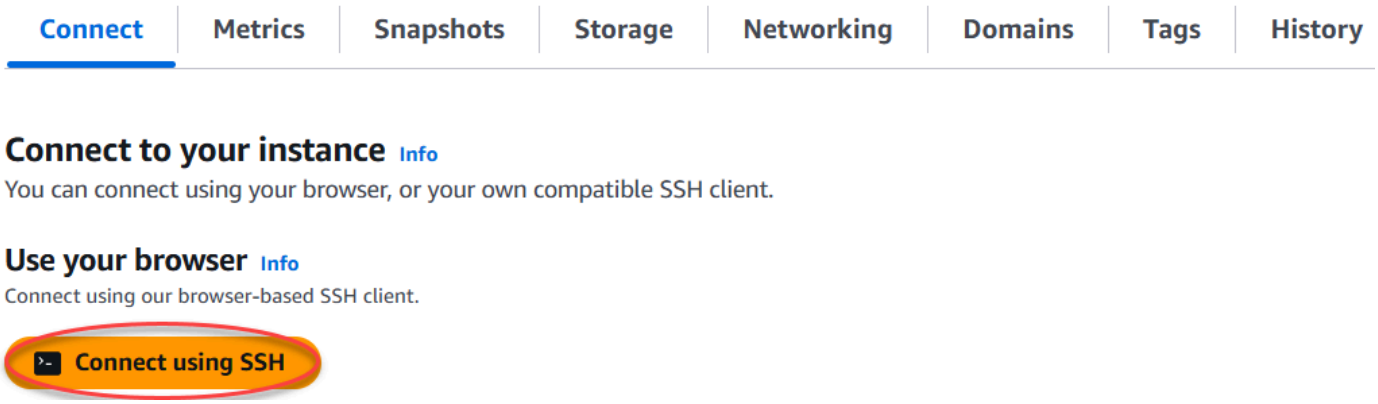
ステップ 4: 登録済みドメイン名へのトラフィックを Magento ウェブサイトに送信する

`example.com` などの登録済みドメイン名のトラフィックを Magento ウェブサイトに送信するには、ドメインのドメインネームシステム (DNS) にレコードを追加します。DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページの [Domains & DNS] (ドメインと DNS) タブで、[Create DNS zone] (DNS ゾーンの作成) を選択し、ページに記載される手順に従います。詳細については、「Lightsail で DNS ゾーンを作成し、ドメインの DNS レコードを管理する」を参照してください。

ドメイン名へのトラフィックがインスタンスにルーティングされたら、次の手順を実行して、Magento ソフトウェアにドメイン名を認識させます。

1. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。



2. 接続後に、次のコマンドを入力します。*<DomainName>* は、インスタンスにトラフィックを送信しているドメイン名に必ず置き換えてください。

```
sudo /opt/bitnami/configure_app_domain --domain <DomainName>
```

例:

```
sudo /opt/bitnami/configure_app_domain --domain www.example.com
```

次の例のようなレスポンスが表示されます。これで、Magento ソフトウェアはドメイン名を認識できるようになります。

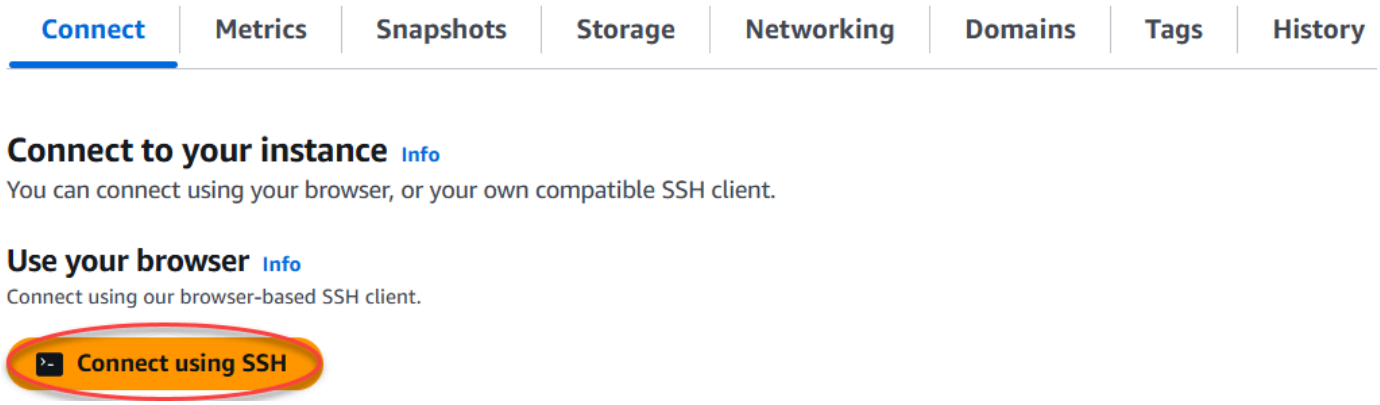
```
bitnami@ip-[redacted]:~$ sudo /opt/bitnami/configure_app_domain --domain www.example.com
Configuring domain to www.example.com
2021-03-12T15:49:22.000Z - info: Saving configuration info to disk
prestashop 15:49:22.41 INFO  ==> Trying to connect to the database server
prestashop 15:49:22.44 INFO  ==> Updating hostname in database
prestashop 15:49:22.46 INFO  ==> Purging cache
Disabling automatic domain update for IP address changes
```

ステップ 5: Magento ウェブサイトの HTTPS を設定する

Magento ウェブサイトで HTTPS を設定するには、以下の手順を実行します。次の手順では、Bitnami HTTPS 設定ツール (bncert) を使用する方法を示しています。このツールは、SSL/TLS 証明書のリクエスト、リダイレクトの設定 (例: HTTP から HTTPS)、および証明書の更新を行うためのコマンドラインツールです。

> **Important**
>
> bncert ツールは現在、 Magento インスタンスのパブリック IP アドレスにトラフィックをルーティングしているドメインに対してのみ証明書を発行します。これらの手順を開始する前に、Magento ウェブサイトで使用するすべてのドメインの DNS に、 DNS レコードが追加されていることを確認してください。

1. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。



2. 接続後に、次のコマンドを入力して bncert-tool をスタートします。

```
sudo /opt/bitnami/bncert-tool
```

次の例のようなレスポンスが表示されます:

```
bitnami@ip-[redacted]:~$ sudo /opt/bitnami/bncert-tool
Warning: Custom redirections are not supported in the Bitnami Magento Stack.
This tool will not be able to enable/disable redirections.
Press [Enter] to continue:
```

3. 次の例に記載されているように、プライマリドメイン名と代替ドメイン名をスペースで区切って入力します。

```
------------------------------------------------------------------------
Welcome to the Bitnami HTTPS Configuration tool.

------------------------------------------------------------------------
Domains

Please provide a valid space-separated list of domains for which you wish to
configure your web server.

Domain list []: example.com www.example.com
```

4. これから実行される変更が一覧表示されます。Y と入力し、Enter を押して確認し、続行します。

```
------------------------------------------------------------------------
Changes to perform

The following changes will be performed to your Bitnami installation:

1. Stop web server
2. Configure web server to use a free Let's Encrypt certificate for the domains:
 example.com www.example.com
3. Configure a cron job to automatically renew the certificate each month
4. Configure web server name to: example.com
5. Start web server once all changes have been performed



Do you agree to these changes? [Y/n]: Y
```

5. Let's Encrypt 証明書に関連付けるメールアドレスを入力し、Enter を押します。

```
Create a free HTTPS certificate with Let's Encrypt

Please provide a valid e-mail address for which to associate your Let's Encrypt
certificate.

Domain list: example.com www.example.com

Server name: example.com

E-mail address []:
```

6. Let's Encrypt サブスクライバー合意書を確認します。Y と入力し、Enter を押して契約に同意し、続行します。

```
The Let's Encrypt Subscriber Agreement can be found at:

https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf

Do you agree to the Let's Encrypt Subscriber Agreement? [Y/n]:
```

これらのアクションは、証明書のリクエストや指定したリダイレクトの設定など、インスタンスで HTTPS を有効にするために実行されます。

```
Performing changes to your installation

The Bitnami HTTPS Configuration Tool will perform any necessary actions to your
Bitnami installation. This may take some time, please be patient.
```

次の例のようなメッセージが表示された場合は、証明書は正常に発行され、検証され、インスタンスでリダイレクトが正常に設定されています。

```
Success

The Bitnami HTTPS Configuration Tool succeeded in modifying your installation.

The configuration report is shown below.

Backup files:
* /opt/bitnami/apache/conf/httpd.conf.back.202104052147
* /opt/bitnami/apache/conf/bitnami/bitnami.conf.back.202104052147
* /opt/bitnami/apache/conf/bitnami/bitnami-ssl.conf.back.202104052147
* /opt/bitnami/apache/conf/vhosts/magento-https-vhost.conf.back.202104052147
* /opt/bitnami/apache/conf/vhosts/magento-vhost.conf.back.202104052147

Find more details in the log file:

/tmp/bncert-202104052147.log

If you find any issues, please check Bitnami Support forums at:

https://community.bitnami.com

Press [Enter] to continue:

bitnami@ip-[redacted]:~$
```

bncert ツールは、有効期限が切れる前、 80 日ごとに証明書の自動更新を実行します。次の一連のステップに進み、Magento ウェブサイトでの HTTPS の有効化を完了します。

7. 以下のアドレスまで移動して、Magento ウェブサイトの管理ダッシュボードのサインインページにアクセスします。*<DomainName>* は、インスタンスにトラフィックを送信している登録済みドメイン名に必ず置き換えてください。

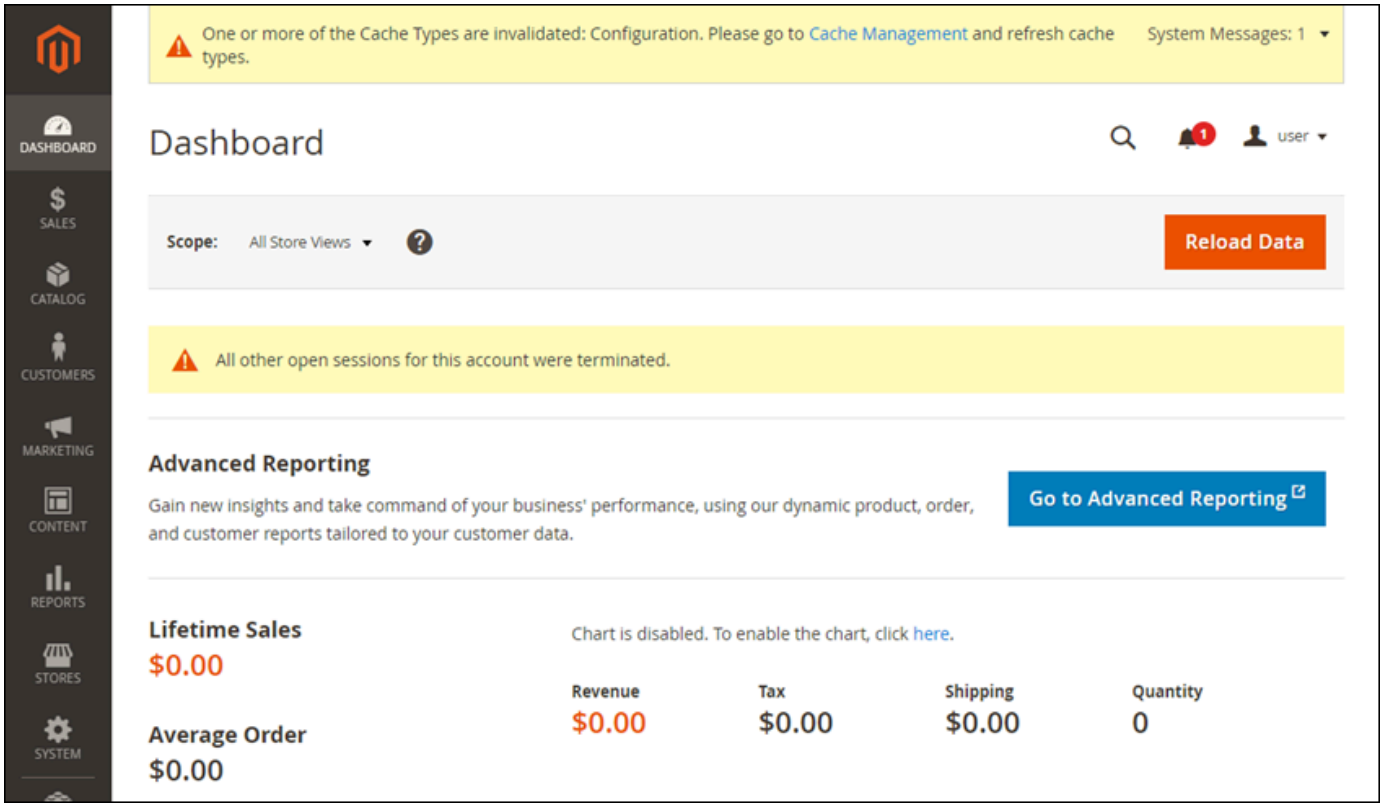
```
http://<DomainName>/admin
```

例:

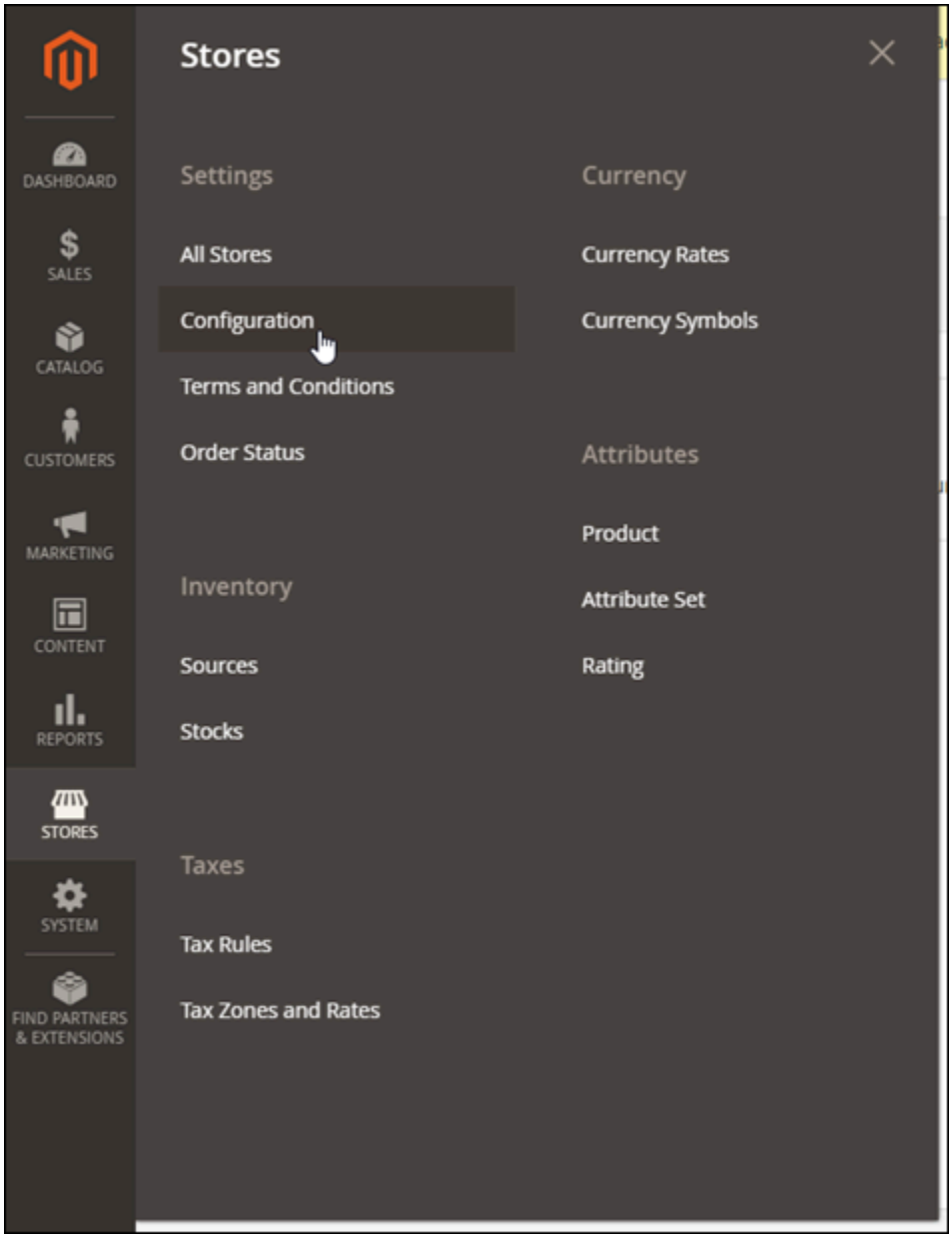
```
http://www.example.com/admin
```

8. デフォルトのユーザー名 (user) と、このガイドの前半のセクションで取得したデフォルトのアプリケーションパスワードを入力して [Sign in] (サインイン) を選択します。

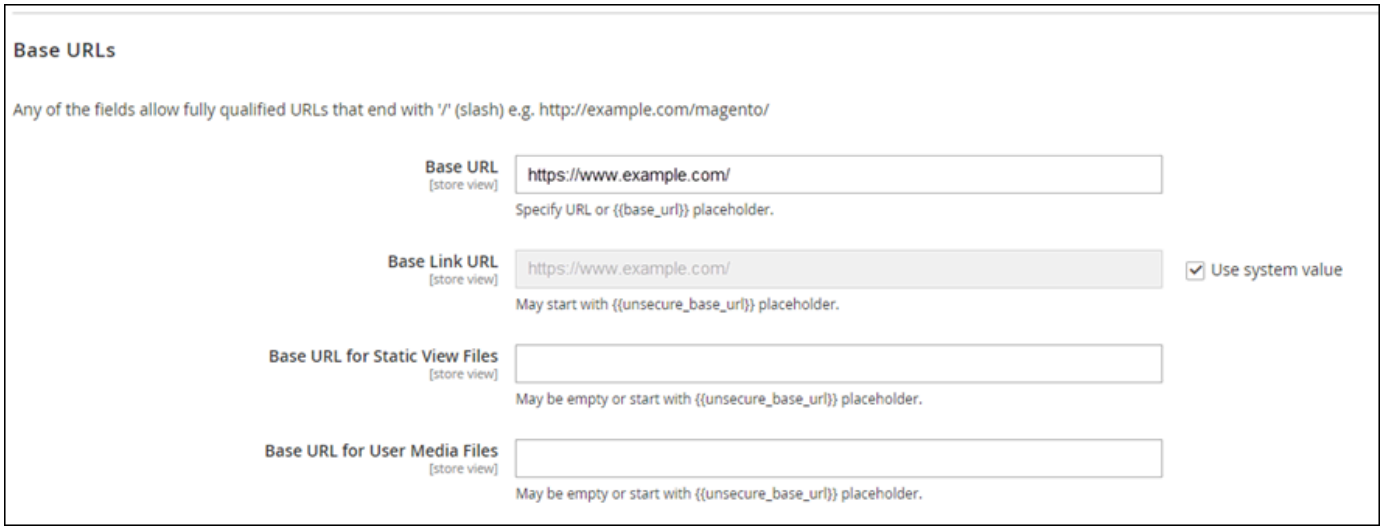
Magento の管理ダッシュボードが表示されます。



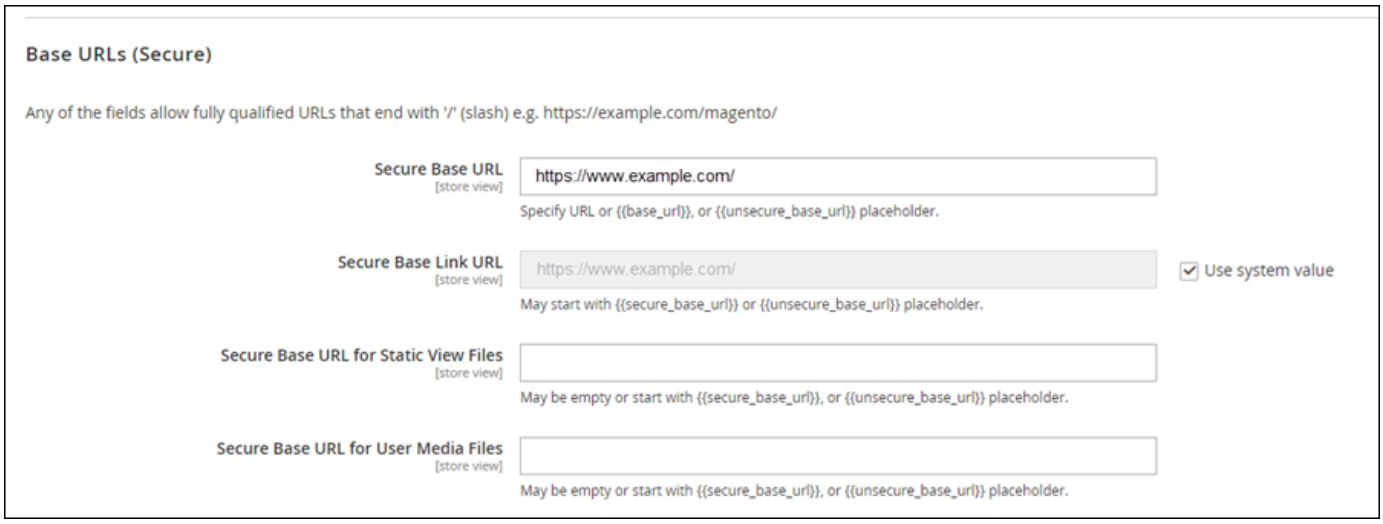
9. ナビゲーションペインで [Stores] (ストア)、[Configuration] (設定) の順に選択します。



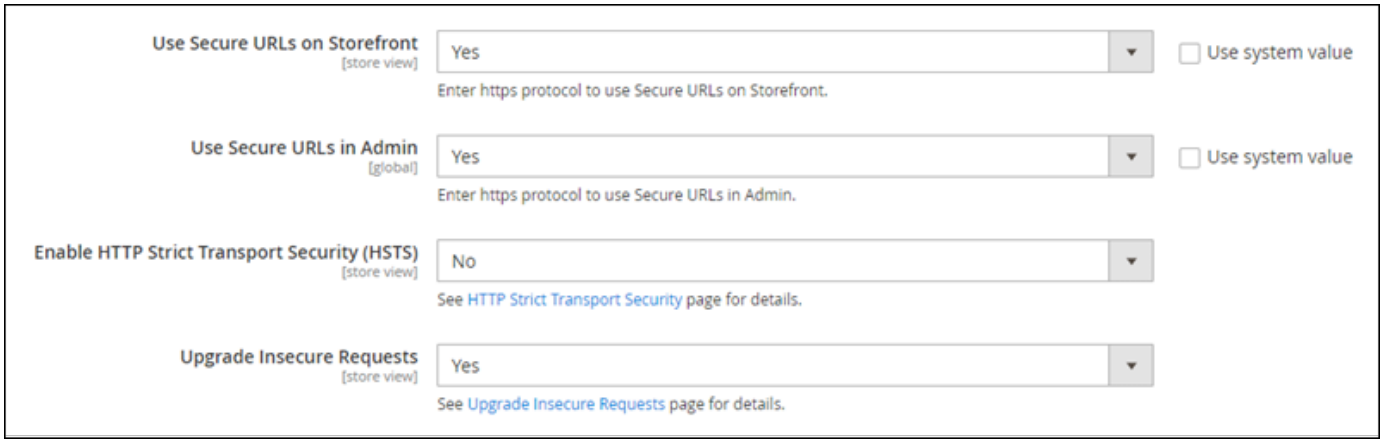
10. [Web] (ウェブ) を選択し、[Base URLs] (ベース URL) ノードを展開します。
11. [Base URLs] (ベース URL) テキストボックスに、ウェブサイトの完全な URL を入力します (例: `https://www.example.com/`)。



12. ベース URL (セキュア) ノードを展開します。
13. [Secure Base URL] (セキュアベース URL) テキストボックスに、ウェブサイトの完全な URL を入力します (例: `https://www.example.com/`)。



14. [Use Secure URLs on Storefront] (ストアフロントでセキュリティで保護された URL を使用する)、[Use Secure URLs in Admin] (管理者でセキュア URL を使用する)、および [Upgrade Insecure Requests] (安全でないリクエストをアップグレードする) で [Yes] (はい) を選択します。



15. ページの上部にある [Save Config] (設定の保存) を選択します。

    これで、Magento ウェブサイトに HTTPS が設定されました。ユーザーが HTTP バージョン (例: `http://www.example.com`) の Magento ウェブサイトを参照すると、ユーザーは自動的に HTTPS バージョン (例: `https://www.example.com`) にリダイレクトされます。

ステップ 6: メール通知用の SMTP を設定する

Magento ウェブサイトの SMTP 設定を構成して、メール通知を有効にします。詳細については、「Bitnami ドキュメント」の「Magento Magepal SMTP 拡張機能のインストール」を参照してください。

> **Important**
>
> ポート 25、465、または 587 を使用するように SMTP を設定する場合、Lightsail コンソールでインスタンスファイアウォールのポートを開ける必要があります。詳細については、「Amazon Lightsail でインスタンスファイアウォールルールの追加および編集」を参照してください。
>
> Gmail アカウントを設定して Magento ウェブサイトでメールを送信できるようにする場合は、Gmail のログインに使用する通常のパスワードではなく、アプリケーションのパスワードを使用する必要があります。詳細については、「アプリケーションのパスワードでサインイン」を参照してください。

ステップ 7: Bitnami と Magento のドキュメントを読む

Bitnami のドキュメントを読んで、Magento インスタンスとウェブサイト上でプラグインのインストールやテーマのカスタマイズなどの管理タスクを実行する方法を確認します。詳細について

は、Bitnami ドキュメントの「AWS クラウド用の Bitnami PrestaShop スタック」を参照してください。

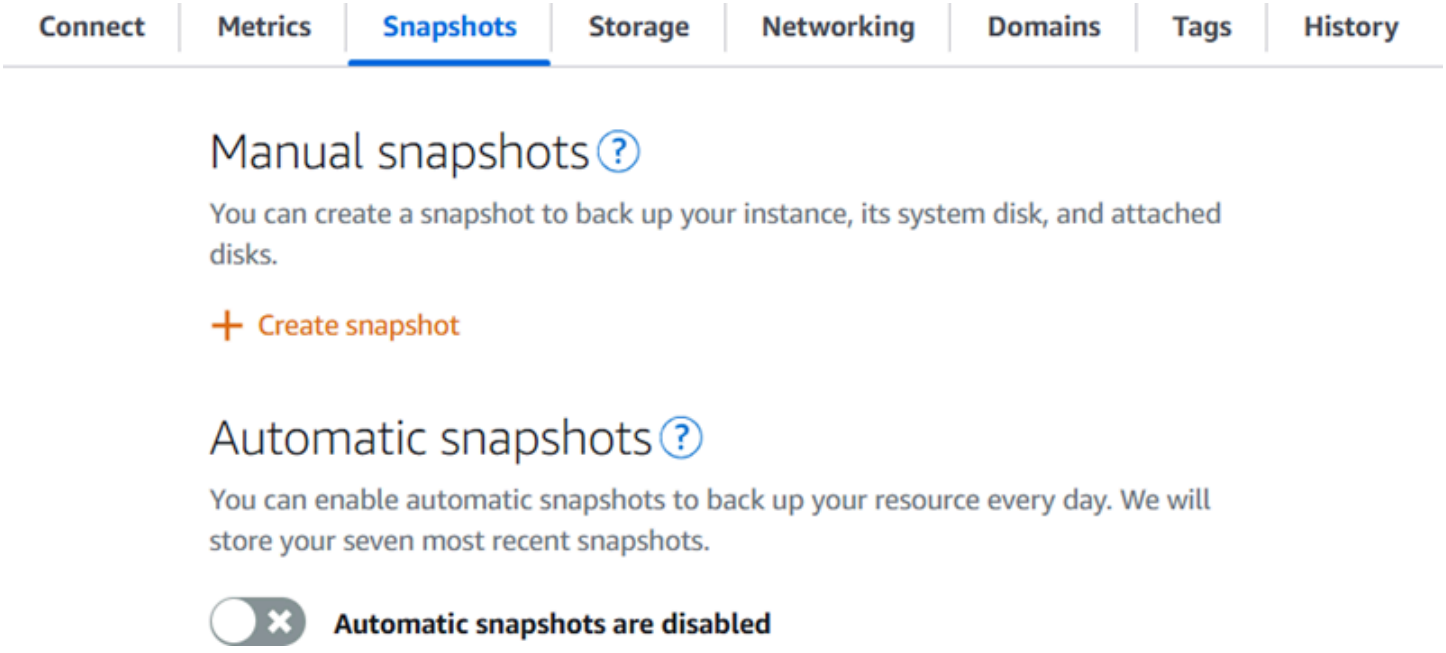
Magento のドキュメントを読んで、Magento のウェブサイトの管理方法も確認してください。詳細については、「Magento 2.4ユーザーガイド」を参照してください。

ステップ 8: Magento インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。

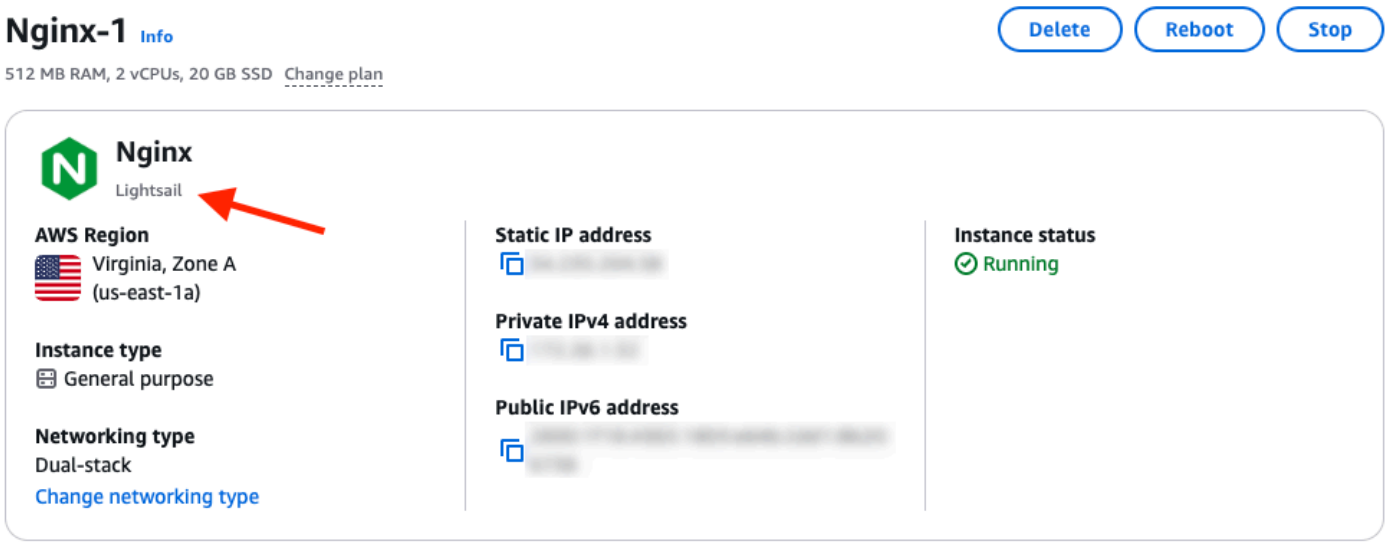


Lightsail で Nginx ウェブサーバーをデプロイして管理する

> 知っていましたか?
>
> Lightsail は毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを最新のスナップショットに自動的に置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

Nginx ブループリントベンダーを特定する

Nginx インスタンスが で起動および実行された後に開始するために実行する必要があるいくつかのステップを次に示しますAmazon Lightsail。開始する前に、インスタンス管理ページでブループリントベンダーを特定します。



Nginx インスタンスに適したガイドを選択します。

Bitnami

ステップ 1: Nginx インスタンスのデフォルトのアプリケーションパスワードを取得する

インスタンスにプリインストールされているアプリケーションやサービスにアクセスするには、アプリケーションのデフォルトパスワードが必要です。

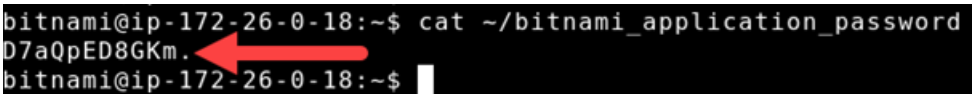
1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。

2. 接続後に、次のコマンドを入力してデフォルトのアプリケーションのパスワードを取得します。

```
cat bitnami_application_password
```

> **Note**
>
> ユーザーのホームディレクトリ以外のディレクトリで作業している場合は、「`cat $HOME/bitnami_application_password`」と入力します。

次のようなレスポンスにアプリケーションのデフォルトパスワードが表示されます。

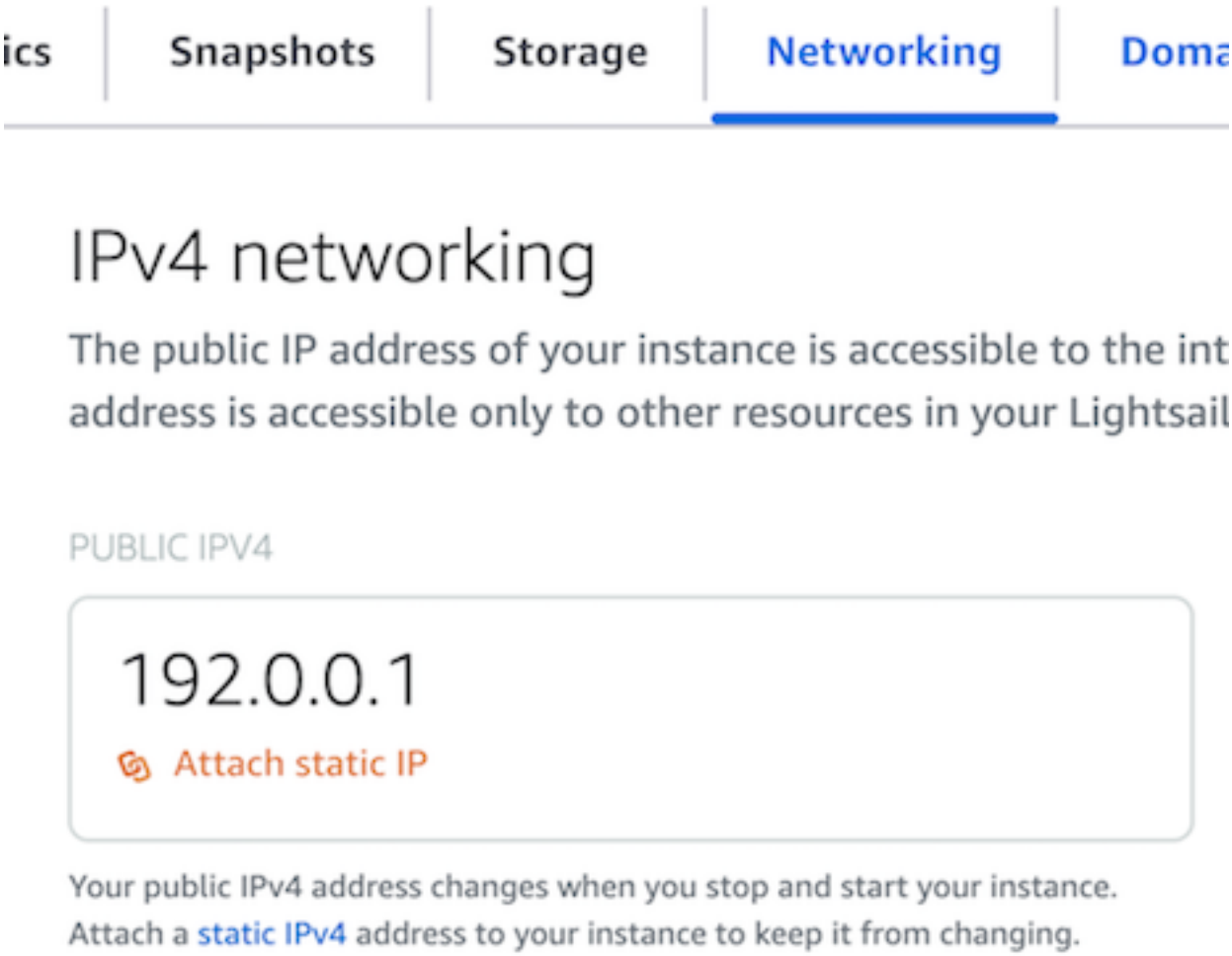


詳細については、「Amazon Lightsailの Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

ステップ 2: Nginx インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



ステップ 3: Nginx インスタンスのウェルカムページにアクセスする

インスタンスのパブリック IP アドレスに移動してインスタンスにインストールされているアプリケーションにアクセスし、phpMyAdmin または Bitnami ドキュメントを参照します。

1. インスタンス管理ページの [接続] タブで、パブリック IP を書き留めます。
2. パブリック IP アドレスを参照します (例: `http://192.0.2.3` に移動します)。

詳細については、「Amazon Lightsailの Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

ステップ 4: ドメイン名を Nginx インスタンスにマッピングする

ドメイン名 (`example.com` など) をインスタンスにマッピングするには、ドメインのドメインネームシステム (DNS) にレコードを追加します。DNS レコードは、通常、ドメインの登録

先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページで、[ネットワーキング] タブの [DNS ゾーンの作成] を選択し、ページに表示される手順に従います。

詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。

ステップ 5: Bitnami のドキュメントを確認する

Bitnami のドキュメントで、Nginx アプリケーションのデプロイ、SSL 証明書による HTTPS サポートの有効化、SFTP を使用したファイルのサーバーへのアップロードなどの方法を確認します。

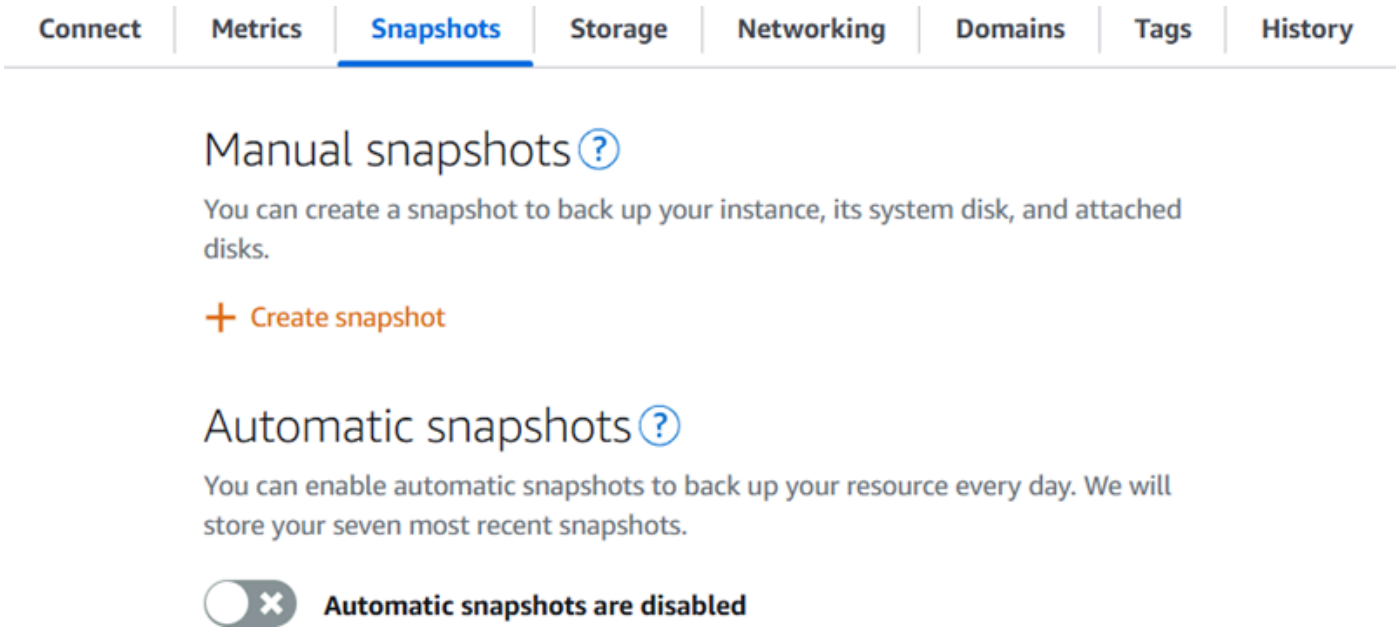
詳細については、「AWS クラウド用の Bitnami Nginx」を参照してください。

ステップ 6: Nginx インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。



Lightsail

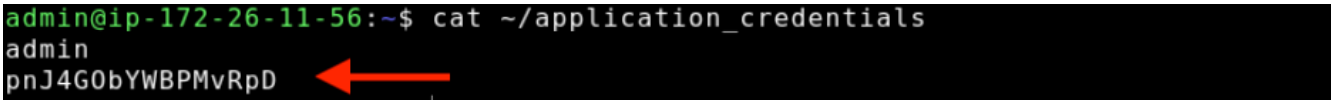
ステップ 1: Nginx インスタンスのデフォルトのアプリケーションパスワードを取得する

インスタンスにプリインストールされているアプリケーションやサービスにアクセスするには、アプリケーションのデフォルトパスワードが必要です。

1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。
2. 接続後に、次のコマンドを入力してデフォルトのアプリケーションのパスワードを取得します。

```
cat ~/application_credentials
```

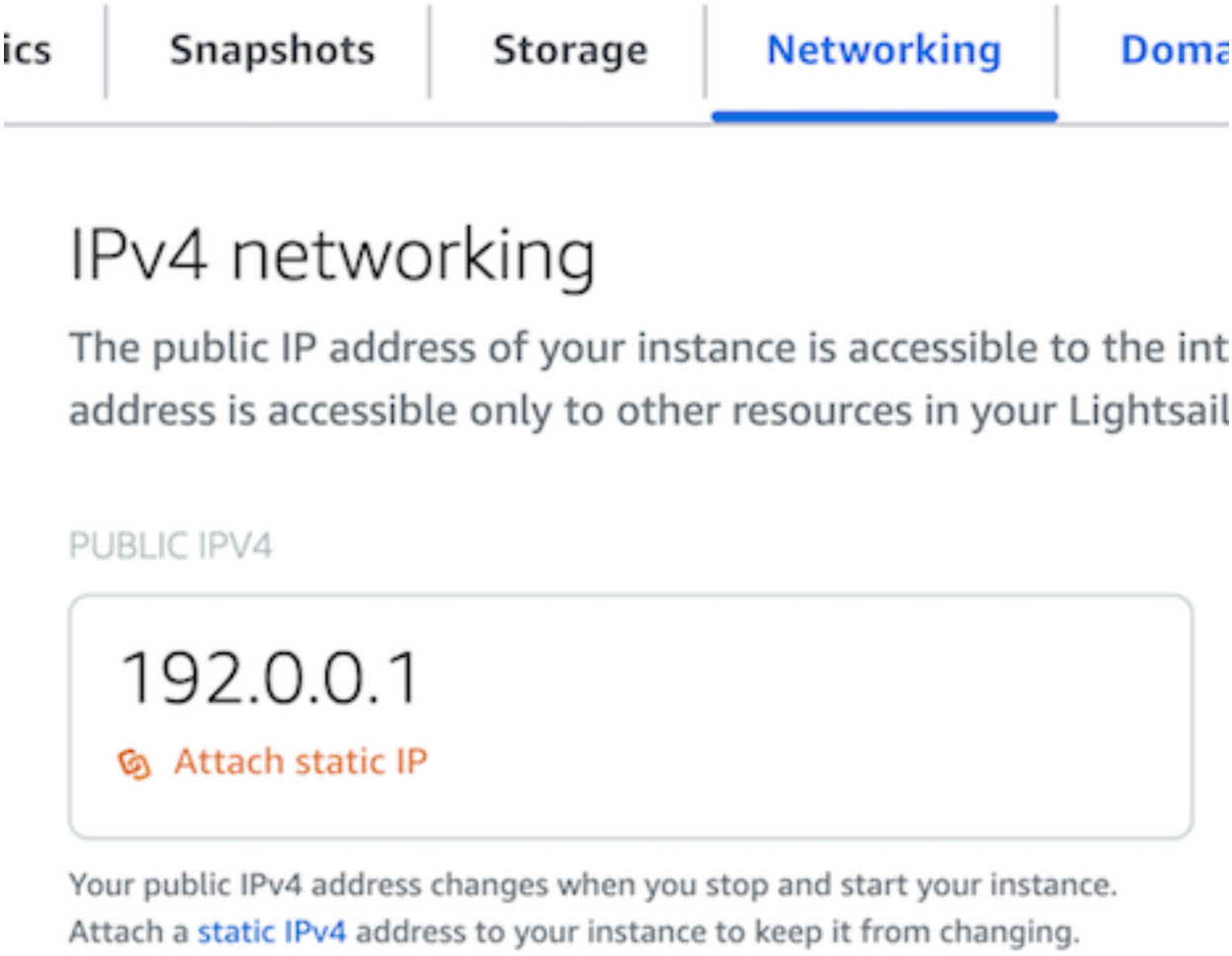
次のようなレスポンスにアプリケーションのデフォルトパスワードが表示されます。



ステップ 2: Nginx インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

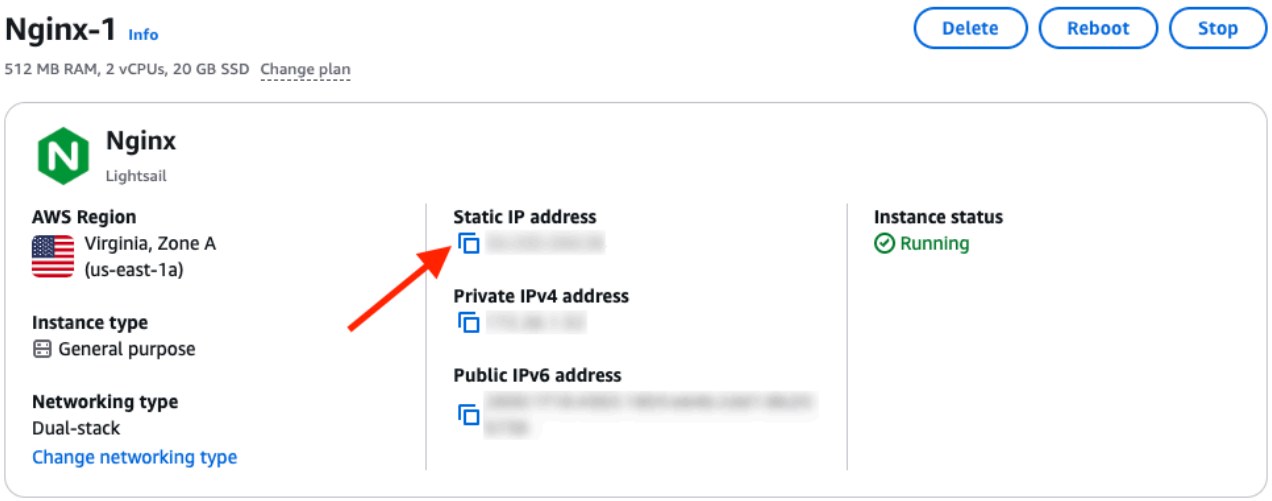
インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



ステップ 3: Nginx インスタンスのウェルカムページにアクセスする

インスタンスの静的 IP アドレスに移動して、インスタンスにインストールされているアプリケーションにアクセスします。

1. インスタンス管理ページで、静的 IP アドレスをコピーします。



2. 静的 IP アドレスをブラウザアドレスに貼り付けます。例: `http://192.0.0.1`。

詳細については、Nginx 管理者ガイドを参照してください。

ステップ 4: ドメイン名を Nginx インスタンスにマッピングする

ドメイン名 (`example.com` など) をインスタンスにマッピングするには、ドメインのドメインネームシステム (DNS) にレコードを追加します。DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページで、[ネットワーキング] タブの [DNS ゾーンの作成] を選択し、ページに表示される手順に従います。

詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。

HTTPS を有効にするには、「Let's Encrypt SSL/TLS を使用して Lightsail Nginx ウェブサイトを保護する」を参照してください。

ステップ 5: アプリケーションをデプロイする

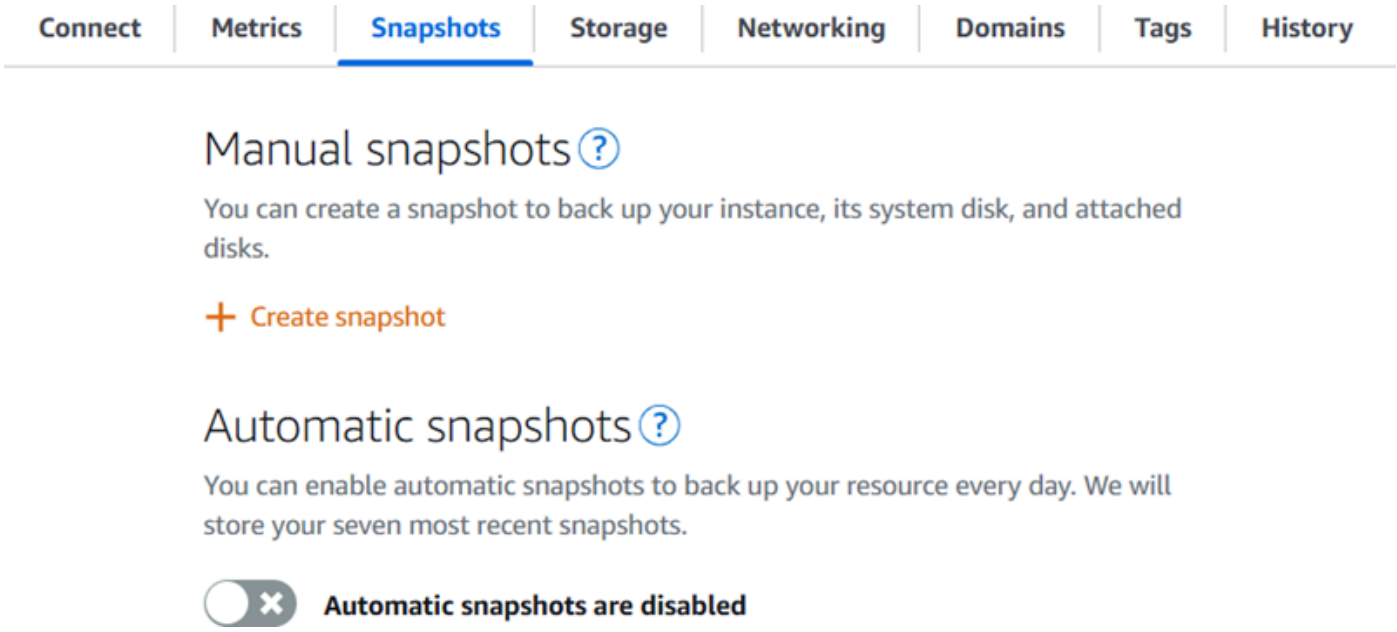
1. 「scp Lightsailを使用してアプリケーションを にコピーする」の「Linux インスタンス間でファイルを転送する」の手順に従います。 `/usr/share/nginx/html`
2. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。
3. `sudo systemctl restart nginx` を実行する
4. インスタンスの静的 IP アドレスに移動する

ステップ 6: Nginx インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

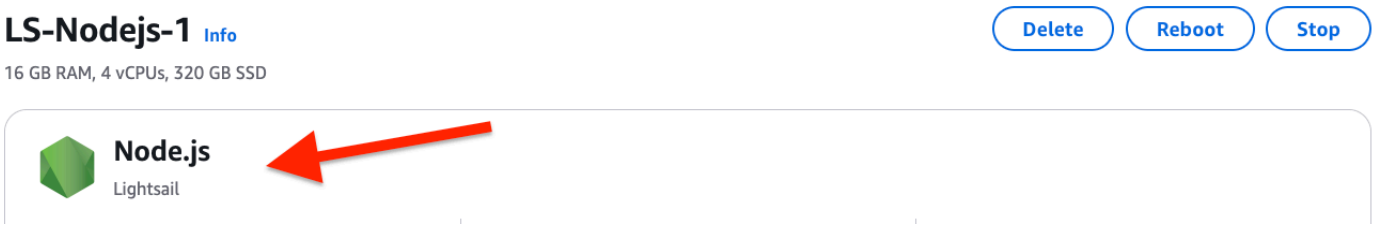
[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。



での Node.js スタックのデプロイと管理 Lightsail

> 知っていましたか?
>
> Lightsail は毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを最新のスナップショットに自動的に置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

Node.js インスタンスが で起動および実行された後に開始するには、いくつかのステップを実行する必要がありますAmazon Lightsail。開始する前に、インスタンス管理ページでブループリントベンダーを特定します。



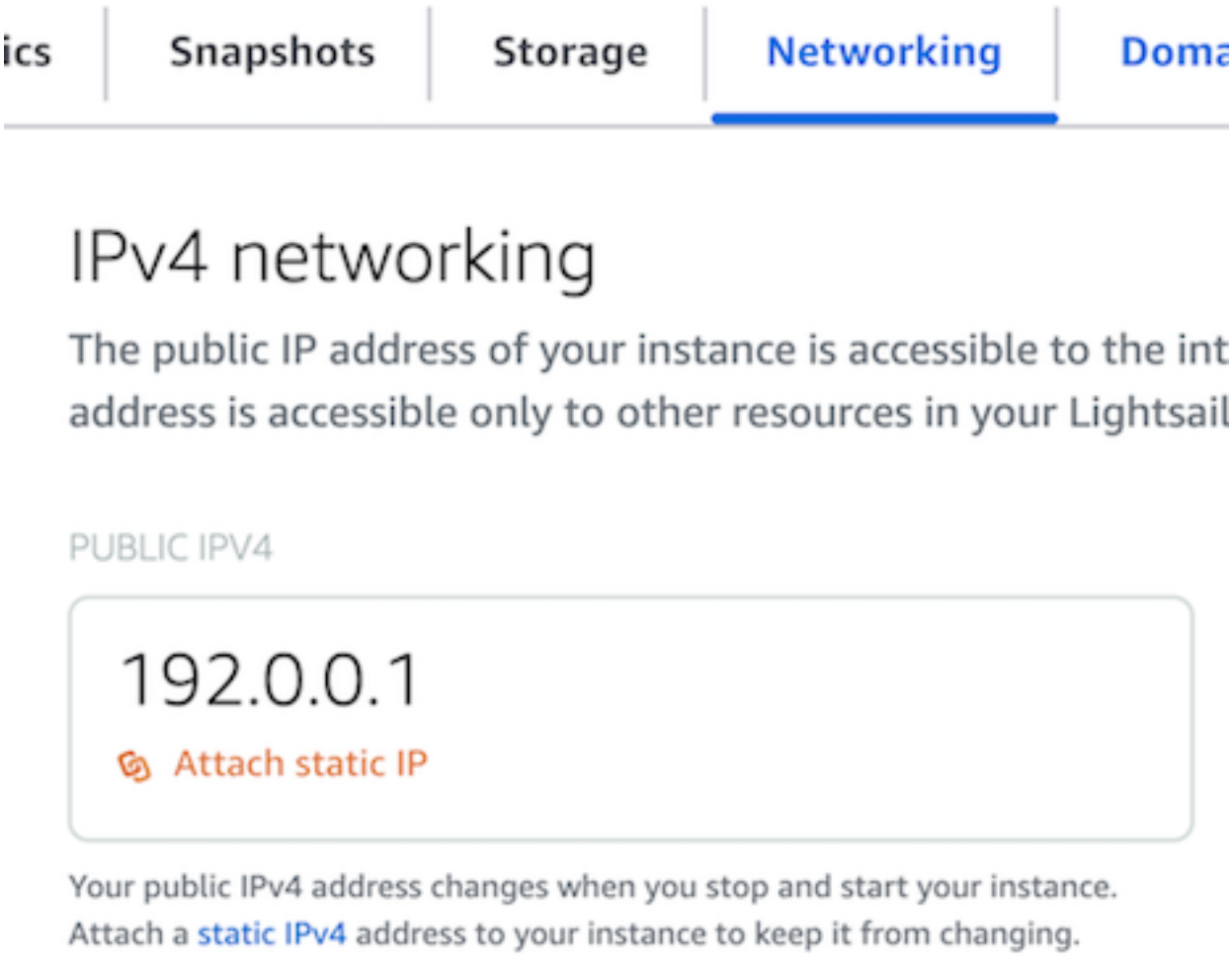
Node.js インスタンスに適したガイドを選択します。

Bitnami

ステップ 1: Node.js インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



ステップ 2: Node.js インスタンスのウェルカムページにアクセスする

インスタンスのパブリック IP アドレスに移動してインスタンスにインストールされているアプリケーションにアクセスし、phpMyAdmin または Bitnami ドキュメントを参照します。

1. インスタンス管理ページの [接続] タブで、パブリック IP を書き留めます。
2. パブリック IP アドレスを参照します (例: `http://192.0.2.3` に移動します)。

詳細については、「Amazon Lightsailの Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

ステップ 3: ドメイン名を Node.js インスタンスにマッピングする

ドメイン名 (`example.com` など) をインスタンスにマッピングするには、ドメインのドメインネームシステム (DNS) にレコードを追加します。DNS レコードは、通常、ドメインの登録

先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページで、[ネットワーキング] タブの [DNS ゾーンの作成] を選択し、ページに表示される手順に従います。

詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。

ステップ 4: Bitnami ドキュメントを読む

Bitnami のドキュメントで、Node.js アプリケーションのデプロイ、SSL 証明書による HTTPS サポートの有効化、SFTP を使用したファイルのサーバーへのアップロードなどの方法を確認します。

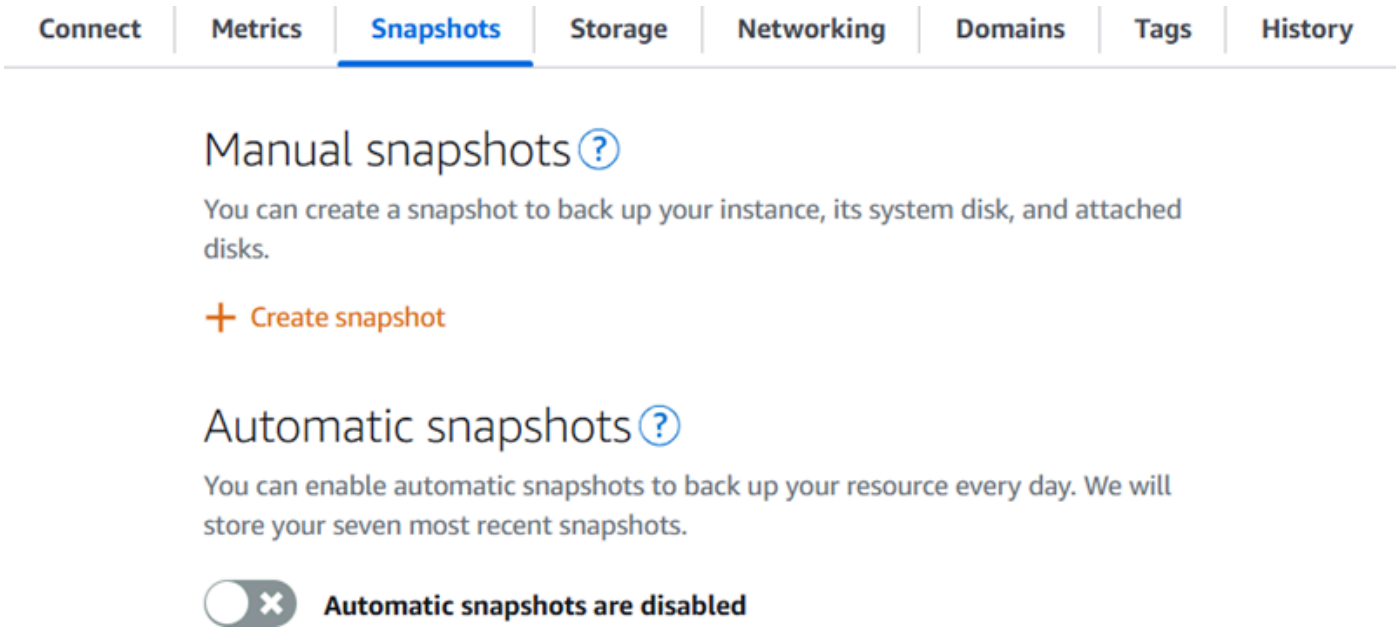
詳細については、「AWS クラウド用の Bitnami Node.js」を参照してください。

ステップ 5: Node.js インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。



Lightsail

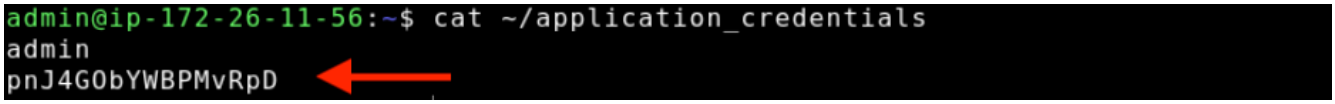
ステップ 1: Node.js インスタンスのデフォルトのアプリケーションパスワードを取得する

インスタンスにプリインストールされているアプリケーションやサービスにアクセスするには、アプリケーションのデフォルトパスワードが必要です。

1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。
2. 接続後に、次のコマンドを入力してデフォルトのアプリケーションのパスワードを取得します。

```
cat ~/application_credentials
```

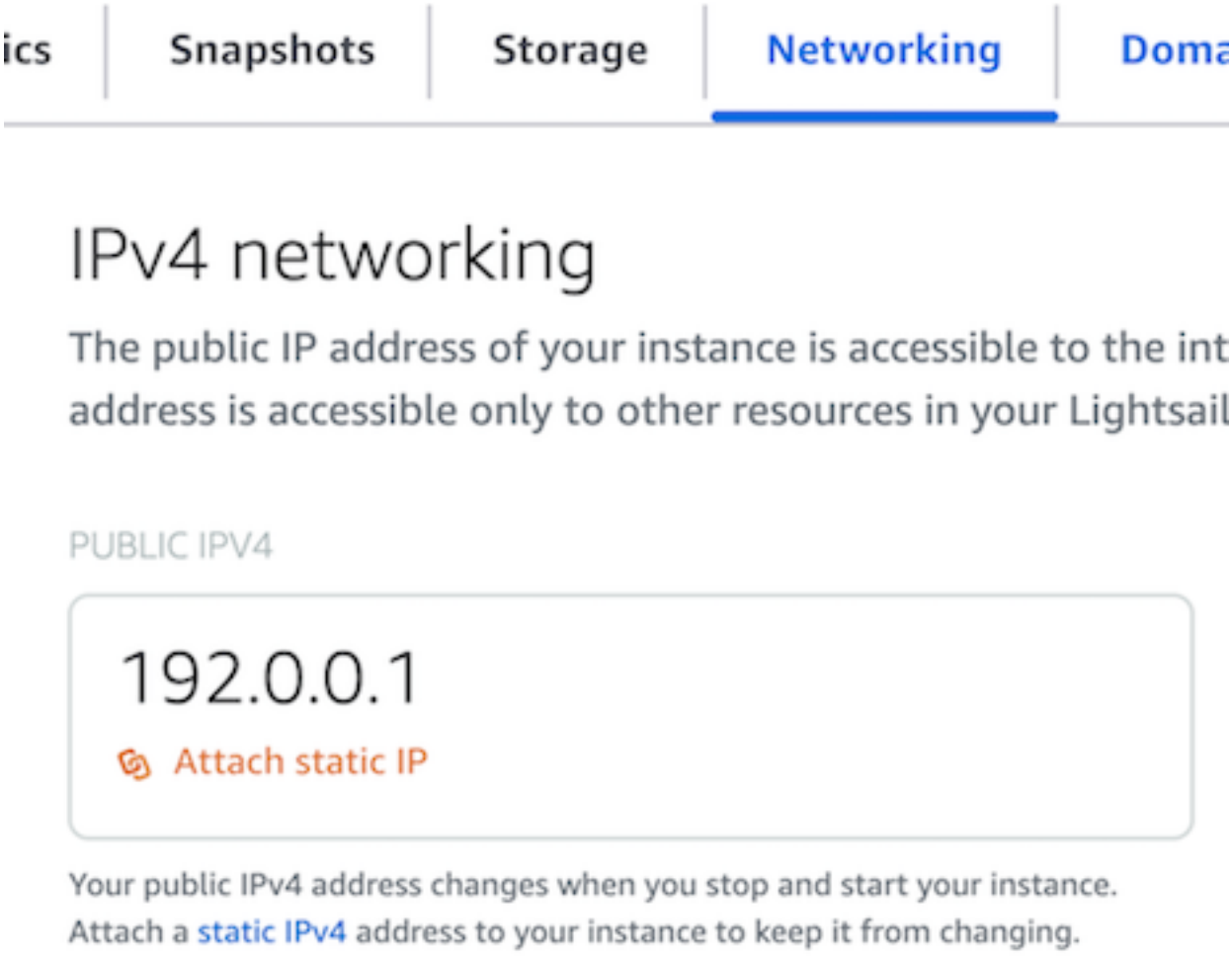
次のようなレスポンスにアプリケーションのデフォルトパスワードが表示されます。



ステップ 2: Node.js インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

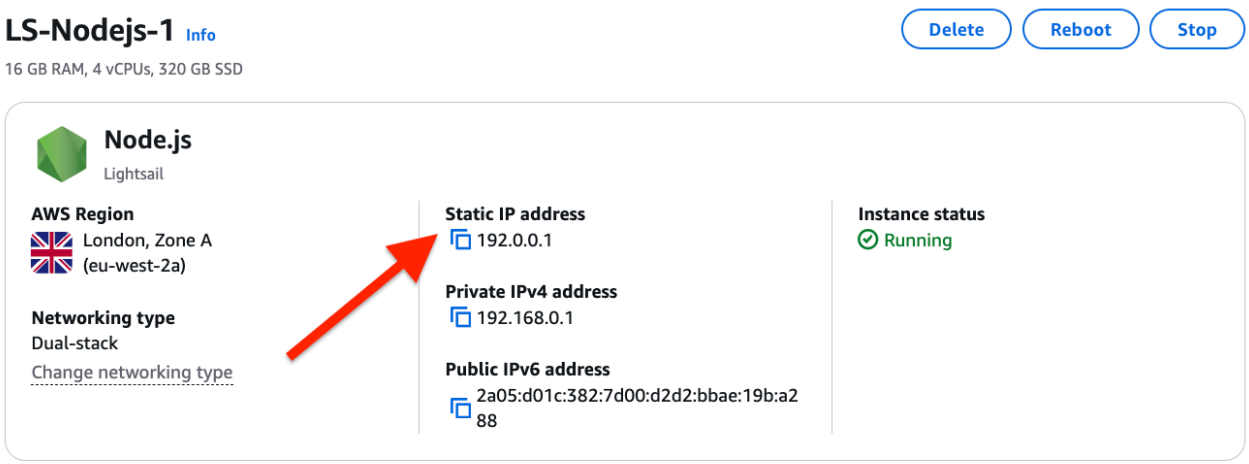
インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



ステップ 3: Node.js インスタンスのウェルカムページにアクセスする

インスタンスの静的 IP アドレスに移動して、インスタンスにインストールされているアプリケーションにアクセスするか、phpMyAdmin にアクセスします。

1. インスタンス管理ページで、静的 IP アドレスをコピーします。



2. 静的 IP アドレスをブラウザアドレスに貼り付けます。例: `http://192.0.0.1`。

詳細については、Node.js 管理者ガイドを参照してください。

ステップ 4: ドメイン名を Node.js インスタンスにマッピングする

ドメイン名 (`example.com` など) をインスタンスにマッピングするには、ドメインのドメインネームシステム (DNS) にレコードを追加します。DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページで、[ネットワーキング] タブの [DNS ゾーンの作成] を選択し、ページに表示される手順に従います。

詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。

ステップ 5: アプリケーションをデプロイする

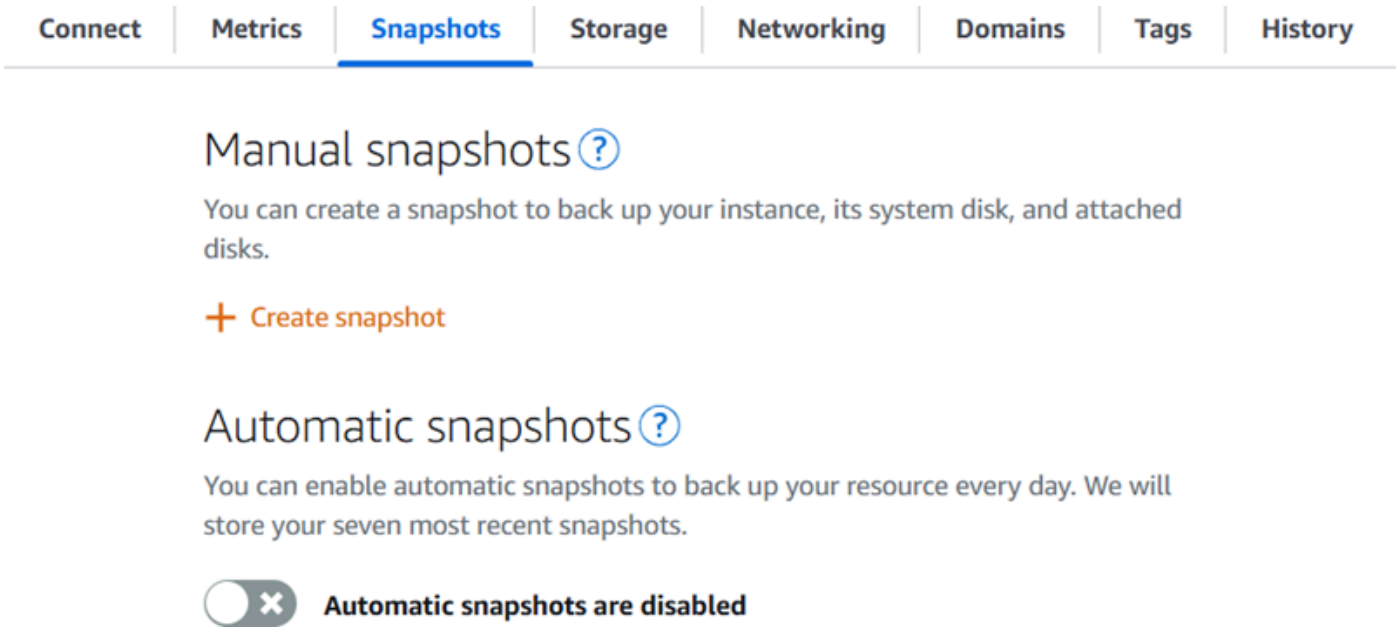
1. 「scp Lightsailを使用してアプリケーションを にコピーする」の「Linux インスタンス間でファイルを転送する」の手順に従います。 `/var/www/html/`
2. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。
3. `sudo pm2 restart nodeapp` を実行する
4. インスタンスの静的 IP アドレスに移動する

ステップ 6: Node.js インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。



で OpenClaw の使用を開始する Lightsail

OpenClaw は、 で実行される AI を活用したチャットゲートウェイでAmazon Lightsail、ブラウザ、Telegram、WhatsApp などからアクセスできるプライベートのセルフホスト型 AI アシスタントを提供します。このチュートリアルでは、Amazon LightsailOpenClaw インスタンスの起動、ブラウザのペアリング、AI 機能の有効化、オプションでメッセージングチャネルの接続について説明します。

> **知っていましたか?**
>
> Lightsail OpenClaw インスタンスには、デフォルトの AI モデルプロバイダーとして Amazon Bedrock が事前設定されています。セットアップが完了したら、AI アシスタントとのチャットをすぐに開始できます。追加の設定は必要ありません。

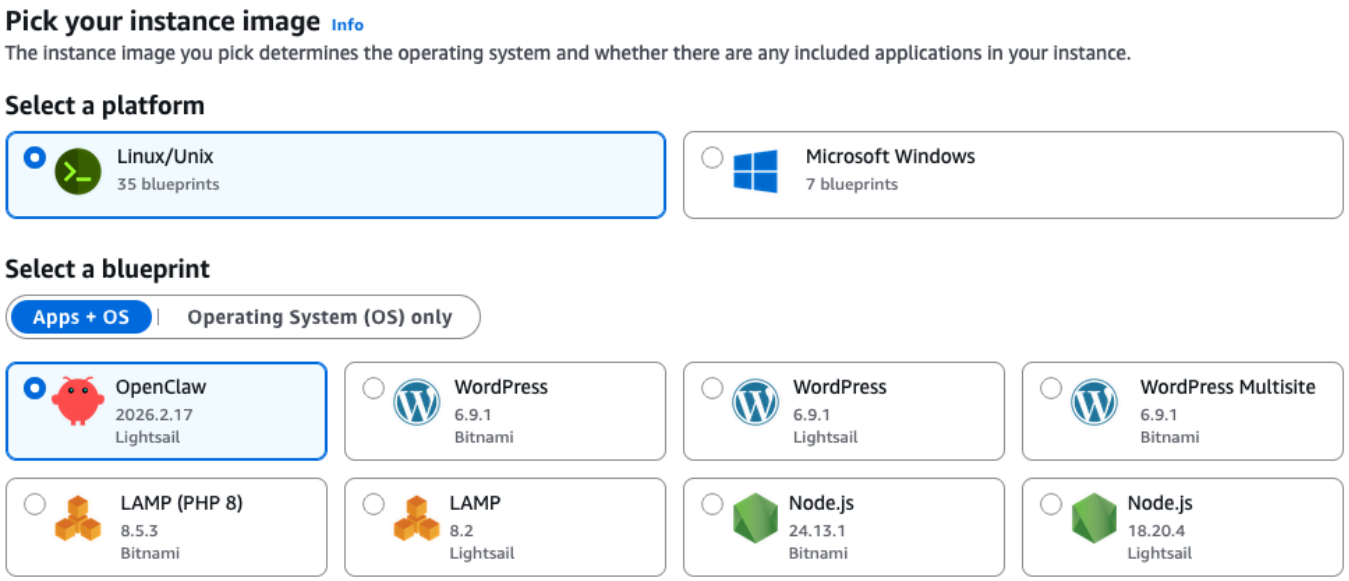
ステップ 1: OpenClaw インスタンスを作成する

このステップでは、OpenClaw を実行するLightsailインスタンスを作成します。

OpenClaw でLightsailインスタンスを作成するには

1. Lightsail コンソール にサインインします。
2. Lightsail ホーム画面の [インスタンス] セクションで、[インスタンスの作成] を選択します。

3. インスタンスの AWS リージョンとアベイラビリティーゾーンを選択します。
4. 次のようにインスタンスのイメージを選択します:
   - プラットフォームの選択 で、Linux/Unix を選択します。
   - 設計図の選択 で、OpenClaw を選択します。



5. 「インスタンスプランの選択」で、インスタンスプランを選択します (最適なパフォーマンスを得るには、4 GB のメモリプランをお勧めします) 。
6. インスタンスの名前を入力するか、指定されたデフォルト名を使用します。

> **Note**
>
> インスタンス名はLightsailアカウント内で一意で、2～255 文字で、英数字で開始および終了し、英数字、ピリオド、ダッシュ、またはアンダースコアのみを含める必要があります。

7. [インスタンスの作成] を選択します。

インスタンスは起動中に数分間開始状態になります。ステータスが実行中になるまで待ってから、次のステップに進みます。

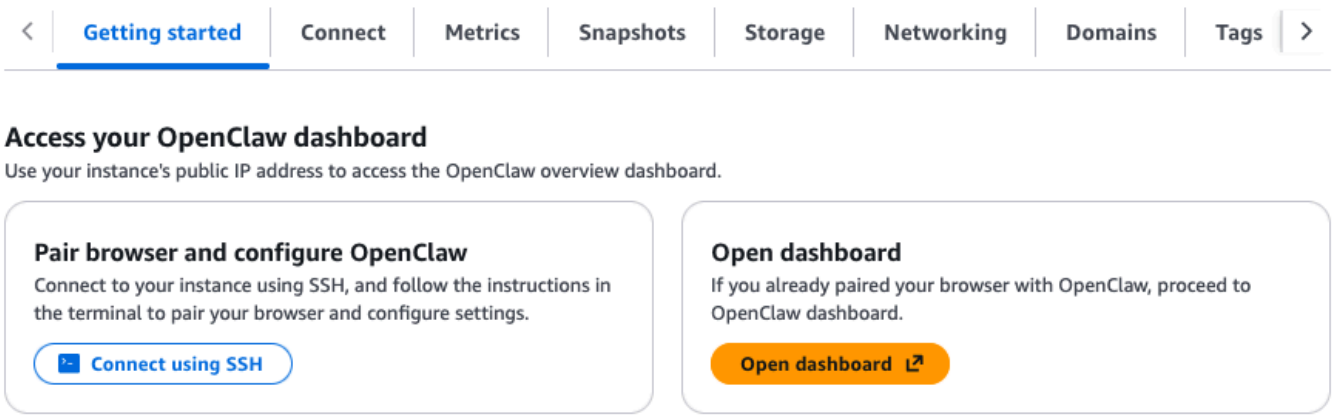
ステップ 2: ブラウザを OpenClaw とペアリングする

OpenClaw ダッシュボードを使用する前に、ブラウザを OpenClaw とペアリングする必要があります。これにより、ブラウザセッションと OpenClaw の間に安全な接続が作成されます。

> **Tip**
>
> OpenClaw ダッシュボードへのアクセスに使用するのと同じデバイスでブラウザを準備します。トークンを からコピーしLightsail、このステップ中に OpenClaw ダッシュボードに貼り付けます。
> インスタンスを停止して起動すると、OpenClaw インスタンスのデフォルトのパブリック IP アドレスが変更されます。インスタンスに静的 IP アドレスをアタッチすると、インスタンスを停止して開始してもそのアドレスが変わりません。詳細については、「Lightsailリソースの IP アドレスの表示と管理」を参照してください。

ブラウザを OpenClaw とペアリングするには

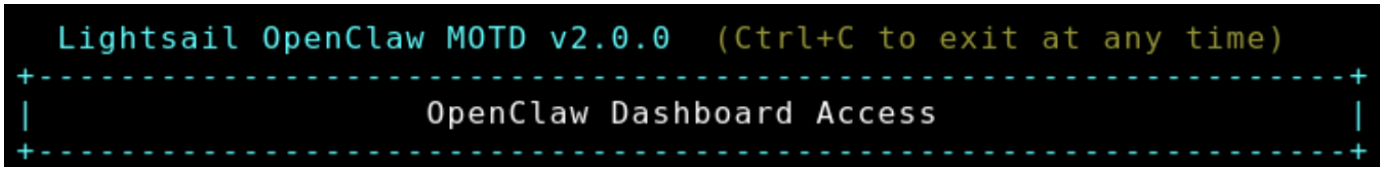
1. Lightsail コンソールのインスタンスセクションで、OpenClaw インスタンスの名前を選択してインスタンス管理ページを開きます。
2. 「開始方法」タブの「ブラウザを OpenClaw にペアリングする」で、「SSH を使用して接続する」を選択します。ブラウザベースの SSH ターミナルが開きます。



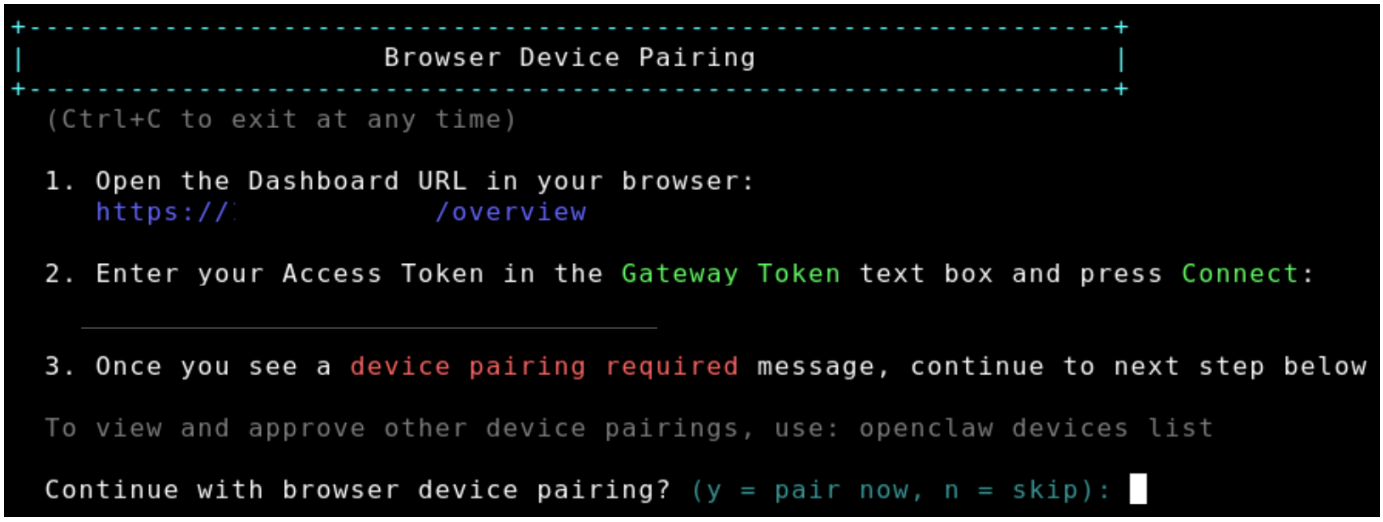
> **知っていましたか?**
>
> OpenClaw インスタンスで実行されている Message of the Day (MOTD) サービスは、オリジン検出、証明書管理、トークンローテーションなど、いくつかの自動設定タスクを

管理します。MOTD バージョンを確認するには、SSH 経由でインスタンスに接続します。

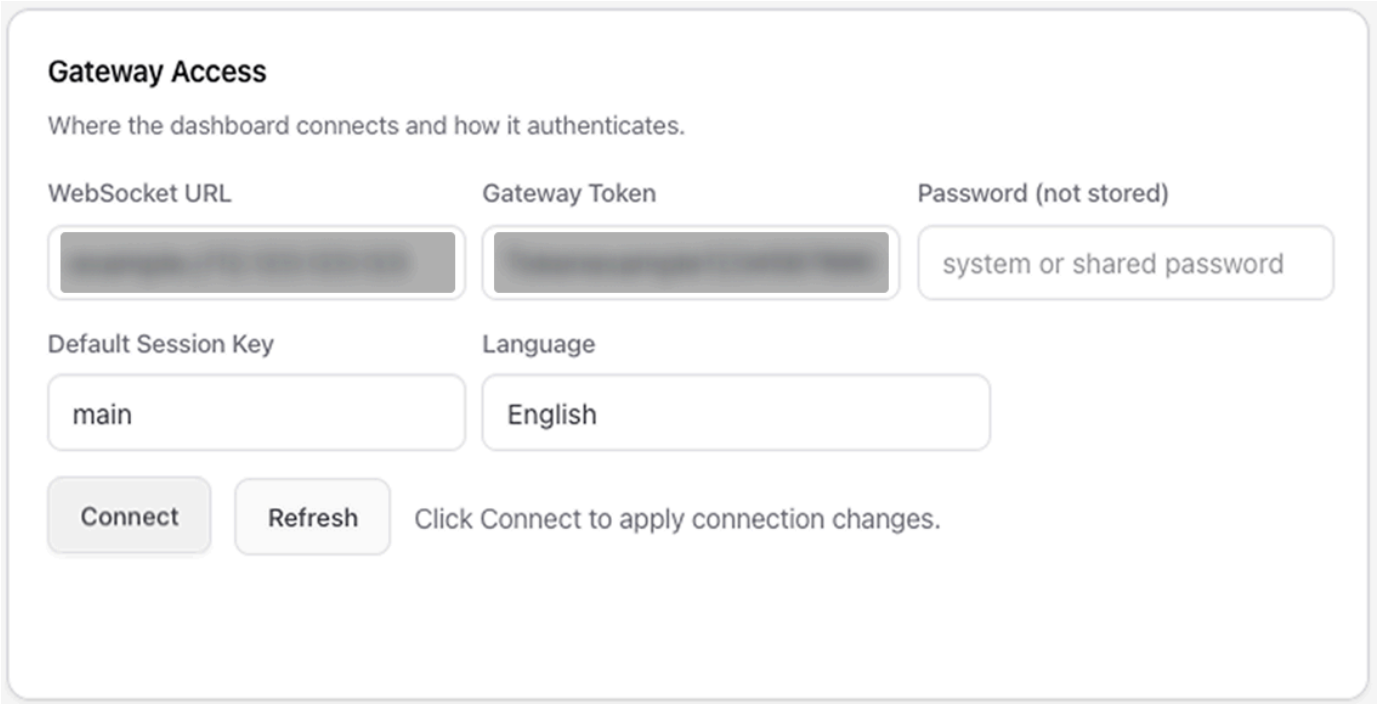
OpenClaw インスタンスは、インスタンスの IP アドレスからの接続を受け入れるようにゲートウェイを自動的に設定します。MOTD バージョン 2.0.0 には、インスタンスの起動時に実行される自動オリジン検出機能が含まれており、許可されたオリジンをインスタンスの現在の IP アドレスとして設定します。インスタンスに静的 IP アドレスをアタッチすると、システムは代わりに静的 IP アドレスを使用するように許可されたオリジンを自動的に更新します。



3. SSH ターミナルで、ウェルカムメッセージに表示されるダッシュボード URL を見つけます。この URL をコピーし、新しいブラウザタブで開きます。

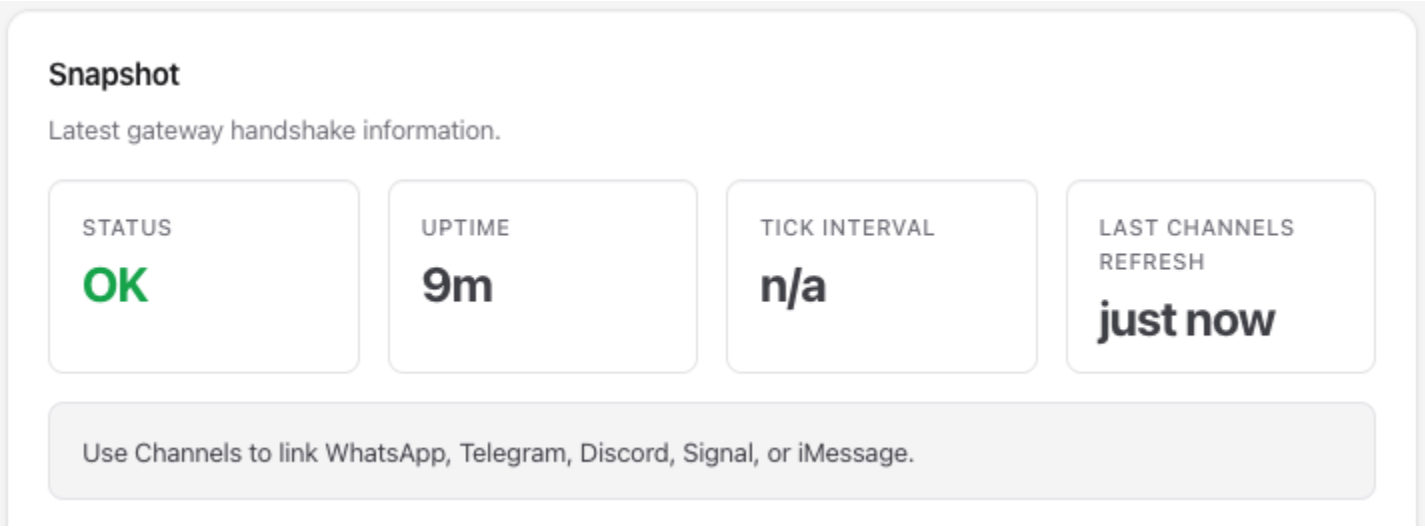


4. 開いた OpenClaw ダッシュボードで、ゲートウェイトークンフィールドを見つけます。
5. SSH ターミナルに戻り、表示されたアクセストークンをコピーします。
6. コピーしたアクセストークンを OpenClaw ダッシュボードのゲートウェイトークンフィールドに貼り付け、Connect をクリックします。



7. SSH ターミナルに戻ります。プロンプトが表示されたら、 **y** を押してデバイスのペアリングを続行します。
8. **a** を押して、デバイスのペアリングリクエストを承認します。

ペアリングが完了すると、OpenClaw ダッシュボードのステータスに OK が表示されます。これで、ブラウザが OpenClaw インスタンスに接続されました。



> **Tip**
>
> 後で追加のブラウザをペアリングする必要がある場合は、インスタンスに再度 SSH 接続し、上記のペアリング手順を繰り返します。

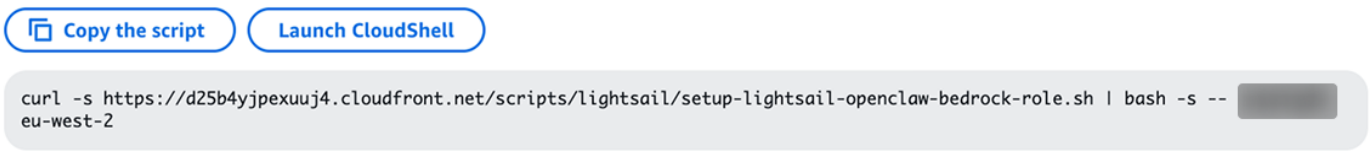
ステップ 3: Amazon Bedrock で AI 機能を有効にする

Lightsail OpenClaw インスタンスは、Amazon Bedrock を使用して AI アシスタントに電力を供給するように設定されています。このステップでは、Bedrock API を呼び出すために必要なアクセス許可をインスタンスに付与します。

Bedrock API アクセスを有効にするには

1. OpenClaw インスタンス管理ページで、開始方法タブを選択します。
2. モデルプロバイダーとして Amazon Bedrock を有効にする で、スクリプトのコピーボタンをクリックします。次に、CloudShell を起動ボタンをクリックして CloudShell を開きます。

> **セットアップスクリプトの動作**
>
> セットアップスクリプトは、OpenClaw インスタンス専用の IAM ロールを作成し、Amazon Bedrock APIs、AWS Marketplace アクセス許可を付与するポリシーをアタッチし (サードパーティーモデルに必要) 、このロールを使用するようにインスタンスプロファイルを設定します。スクリプトの実行後、IAM コンソールで IAM ポリシーの詳細を確認できます。IAM ロールの名前は になりますLightsailRoleFor-[your-instance-id]。



3. コピーしたコマンドを CloudShell ターミナルに貼り付け、 を押しますEnter。
4. スクリプトが完了するまで待ちます。出力に「完了」と表示されると、アクセス許可は正常に適用されました。

このステップが完了したら、OpenClaw ダッシュボードのチャットに移動して、AI アシスタントの使用を開始します。

注: Lightsail OpenClaw インスタンスは、デフォルトで Anthropic Claude Sonnet 4.6 を使用します。Amazon Bedrock で Anthropic モデルを初めて使用する場合は、初回使用 (FTU) フォームに記入してアクセスする必要があります。Anthropic モデルにアクセスする方法について詳しく説明します。

ステップ 4: メッセージングチャネルを接続する (オプション)

OpenClaw を拡張して Telegram や WhatsApp などのメッセージングアプリと連携できるため、AI アシスタントを電話やメッセージングクライアントから直接操作できます。OpenClaw をメッセージングチャネルに接続する前に、ブラウザを OpenClaw とペアリングする必要があります (ステップ 2 を参照) 。

テレグラムを接続する

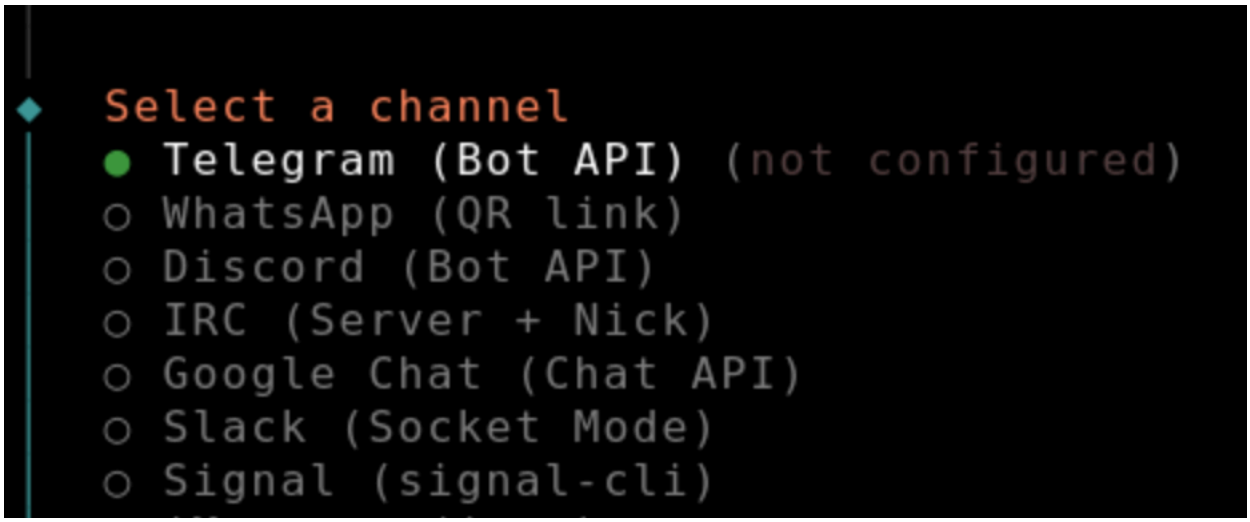
テレグラムチャネルを追加するには

1. Telegram を開き、 を検索します`@BotFather`。
2. コマンドを送信`/newbot`し、プロンプトに従って新しいボットを作成します。BotFather は、ボットトークンとボットのディープリンクを提供します。

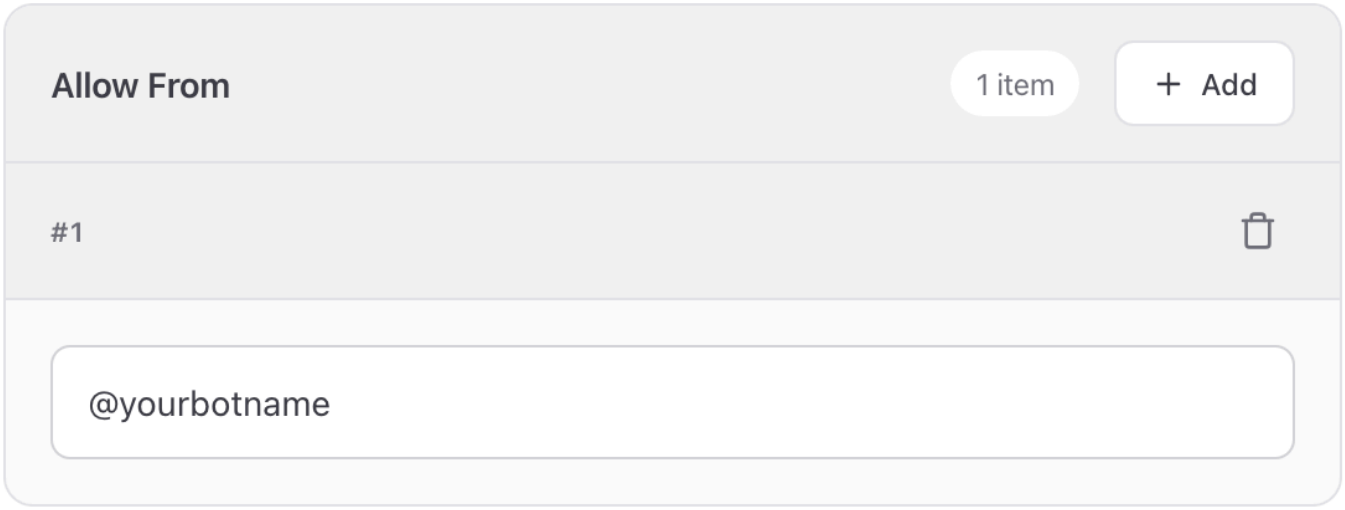
3. SSH を使用して OpenClaw インスタンスに接続します。ブラウザベースの SSH ターミナルが開きます。
4. OpenClaw インスタンスに接続されている SSH ターミナルで、以下を実行します。

```
openclaw channels add
```

5. 使用可能なチャネルのリストから Telegram を選択します。

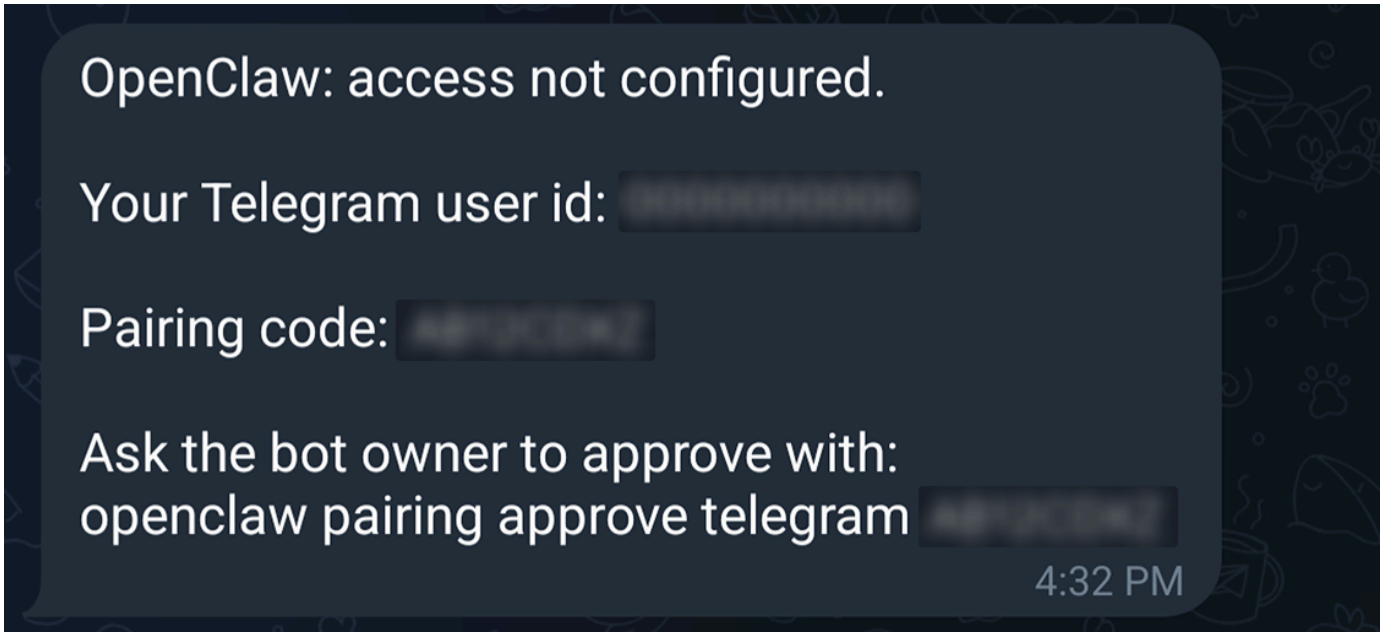


6. プロンプトが表示されたら、ステップ 2 で BotFather から受け取ったボットトークンを入力します。
7. OpenClaw ダッシュボードで、チャネルセクションに移動し、Telegram ユーザー ID を許可リストに追加します。

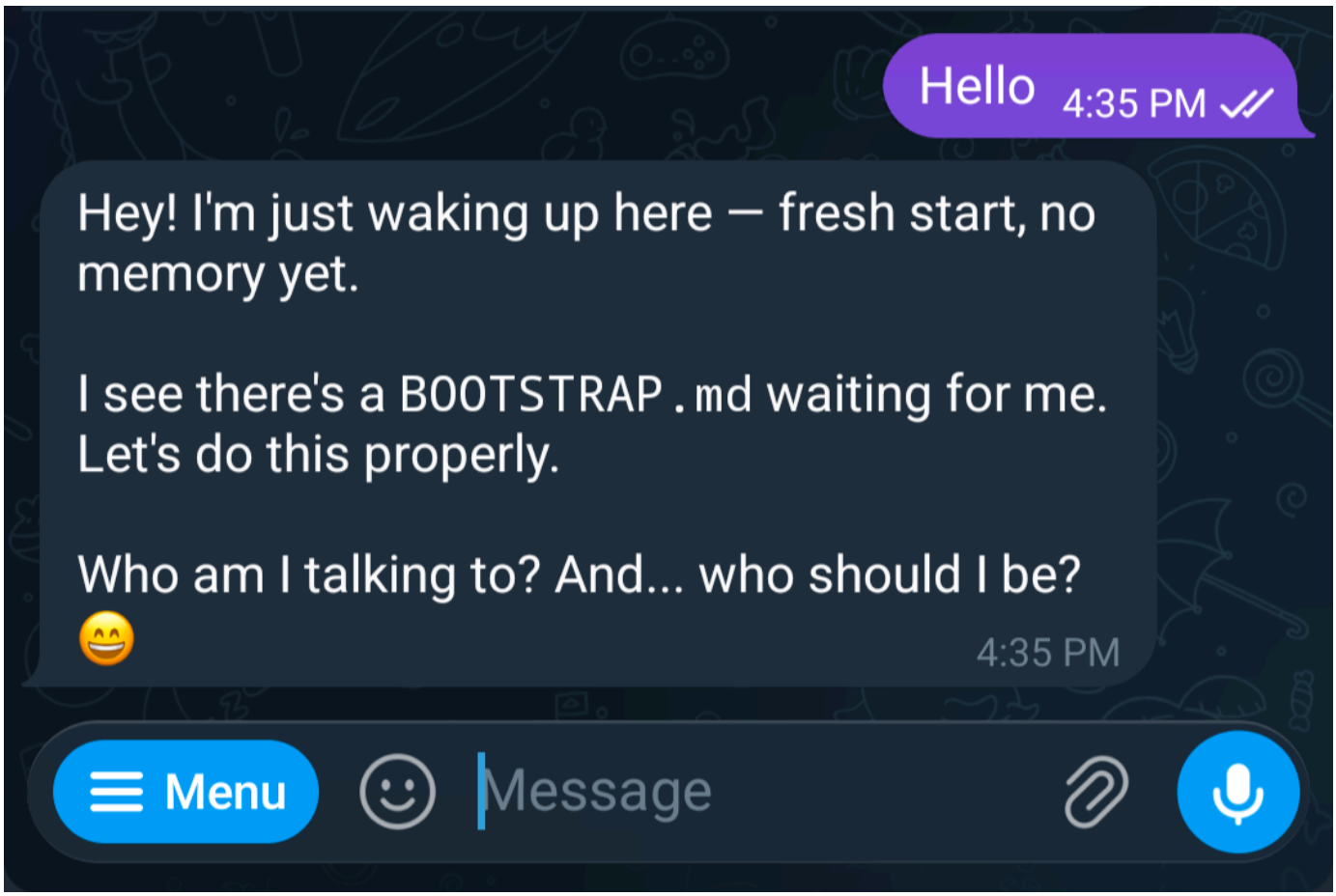


8. ステップ 2 の Telegram でボットにメッセージを送信して、統合をテストします。
9. OpenClaw のペアリングを承認するメッセージが Telegram に表示されます。SSH ターミナルで、以下を実行します。

```
openclaw pairing approve telegram [pairing code]
```



10. ステップ 2 の Telegram で作成したボットにメッセージを送信して、統合を再度テストする



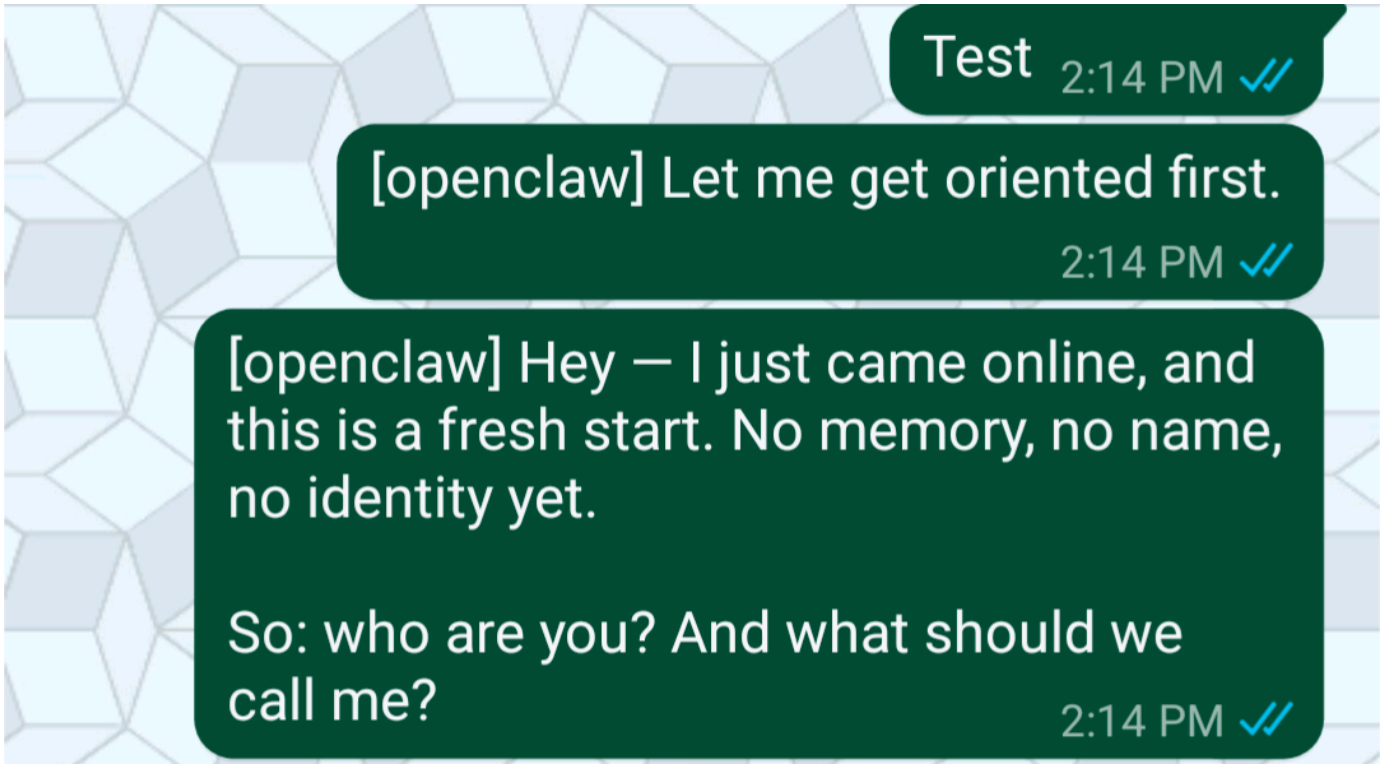
WhatsApp を接続する

WhatsApp チャネルを追加するには

1. SSH を使用して OpenClaw インスタンスに接続します。ブラウザベースの SSH ターミナルが開きます。
2. OpenClaw インスタンスに接続されている SSH ターミナルで、以下を実行します。

```
openclaw channels add
```

3. 使用可能なチャネルのリストから WhatsApp を選択します。
4. 画面上の指示に従ってください。ターミナルに QR コードが表示されます。
5. スマートフォンで WhatsApp を開き、リンクされたデバイス機能を使用して QR コードをスキャンします。
6. スマートフォンでペアリングを完了します。
7. 上記のステップでペアリングした連絡先番号をメッセージングして、WhatsApp から直接 OpenClaw アシスタントにメッセージを送信して、統合をテストします。 WhatsApp



ステップ 5: インスタンスのスナップショットを作成する (オプション)

セットアップが完了したら、OpenClaw インスタンスのスナップショットを作成することをお勧めします。スナップショットは、インスタンスを既知の状態から復元し、信頼性の高い復旧メカニズムを提供するpoint-in-timeバックアップです。スナップショットから目的のプランの新しいインスタンスを作成することもできます。

スナップショットを手動で作成するには

1. Lightsail コンソールのインスタンスセクションで、OpenClaw インスタンスの名前を選択します。
2. [スナップショット] タブを選択します。
3. 手動スナップショットで、スナップショットの作成を選択します。
4. スナップショットの名前を入力し、作成を選択します。

> 知っていましたか?
>
> Lightsail は毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを自動的に最新のスナップショットに置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

よくある質問 (FAQ)

追加のブラウザを OpenClaw インスタンスとペアリングするにはどうすればよいですか?

必要な数のブラウザをペアリングできます。新しいブラウザをペアリングするには、もう一度 OpenClaw インスタンスに SSH 接続します (Lightsailコンソールのインスタンス管理ページから、SSH を使用して接続を選択します) 。ステップ 2: ダッシュボード URL に移動し、ターミナルからアクセストークンをコピーし、新しいブラウザの OpenClaw ダッシュボードのゲートウェイトークンフィールドに貼り付け、ペアリングリクエストを承認します。

OpenClaw インスタンスに付与された IAM アクセス許可をカスタマイズできますか?

はい。ステップ 3 のセットアップスクリプトは、Amazon Bedrock へのアクセスを許可するポリシーを持つ IAM ロールを作成します。このポリシーはいつでも表示、編集、または制限できます。

- IAM コンソールを開き、ロールに移動します。

- OpenClaw インスタンス用に作成されたロールを見つけます。例: `LightsailRoleFor-i-0d15d5483571b95bb`。
- アタッチされたポリシーを表示するロールを選択します。
- アクセス許可を編集するポリシー名を選択します。

アクセス許可を変更するときは注意が必要です。必要な Bedrock アクセス許可を削除すると、OpenClaw が AI レスポンスを生成できなくなります。詳細については、AWS ドキュメントの「IAM ポリシー」を参照してください。

できることや使用方法など、OpenClaw 自体について質問できますか?

はい。OpenClaw の組み込みチャットアシスタントは、OpenClaw 自体に関する質問に回答できます。OpenClaw で何ができるかわからない場合は、チャットインターフェイスで直接質問してください。たとえば、次のように入力できます。

- 「何が役に立ちますか?」
- OpenClaw に接続できるチャネルは何ですか?」
- 「新しいメッセージングチャネルを追加するにはどうすればよいですか?」

OpenClaw は、その機能に基づいてガイダンスで応答します。これは、OpenClaw ダッシュボードを離れることなく機能を調べるのに最適な方法です。

注: Amazon Bedrock で AI 機能を有効にするステップ (入門ガイドのステップ 3) を完了するか、チャットが機能するように独自のモデルプロバイダーを設定する必要があります。Bedrock のセットアップでは、インスタンスの開始方法タブからワンクリックスクリプトを実行して必要なアクセス許可を付与し、Anthropic モデルを初めて使用する場合は、Amazon Bedrock コンソールで簡単な初回使用フォームを送信します。このステップがないと、チャットインターフェイスには接続する AI モデルがありません。

OpenClaw の実行にはどのようなLightsailコストがかかりますか?

コストの内訳は次のとおりです。

- Lightsail インスタンス — 選択したインスタンスプラン (4 GB プランなど) に対して料金が発生します。 Lightsail プランはオンデマンドの時間料金で請求されるため、使用した分に対してのみ料金が発生します。ご利用の Lightsail プランごとに固定の時間料金が課金され、プランの月額上限まで適用されます。

- AI モデルの使用 (トークン) — OpenClaw アシスタントとの間で送受信されるすべてのメッセージは、トークンベースの料金モデルを使用して Amazon Bedrock を通じて処理されます。コストはモデルによって異なります。一部のモデルは、他のモデルよりもトークンごとにコストが高くなります。
- サードパーティーモデルのサブスクリプション — AWS Marketplace を通じて配布されるサードパーティーモデル (Anthropic Claude や Cohere など) を選択した場合、トークンあたりのコストに加えて追加のソフトウェア料金が発生する場合があります。これらは、請求書の AWS Marketplace の下に個別の明細項目として表示されます。
- データ転送超過 — 各Lightsailプランには毎月のデータ転送許容量が含まれます。OpenClaw インスタンスがプランに含まれるデータよりも多くのデータを送受信する場合、データ転送には超過料金が適用されます。
- スナップショット — Lightsailインスタンスの手動スナップショットと自動スナップショットは、使用されたストレージの量に基づいて請求されます。

Anthropic モデルを使用します。他に必要なことはありますか?

Anthropic には、標準のアクセス許可以外にも 1 つの追加要件があります。Anthropic モデルを初めて呼び出す前に、初回使用 (FTU) フォームに記入する必要があります。これは Anthropic の要件であり、AWS アカウントごとに 1 回、または組織内のすべてのメンバーアカウントによって継承される AWS Organization の管理アカウントレベルで 1 回適用されます。OpenClaw インスタンスは、デフォルトで Anthropic Claude-Sonnet 4.6 を使用します。

Lightsail は、ステップ 3 のセットアップの一環として、基盤となる IAM および Marketplace のアクセス許可を処理します。CloudShell スクリプトは、3 つの必要な AWS Marketplace アクセス許可 (`aws-marketplace:Subscribe`、`aws-marketplace:Unsubscribe`、および) を含む IAM ロールを作成します`aws-marketplace:ViewSubscriptions`。これらは、Amazon Bedrock がサードパーティーモデルを初めて呼び出すときに自動的に有効にするために必要です。アカウントでモデルを有効にすると、アカウント内のすべてのユーザーが Marketplace のアクセス許可なしでモデルを呼び出すことができます。サブスクリプションは 1 回のみ実行する必要があります。

Anthropic 固有の FTU 要件を完了するには:

- Amazon Bedrock コンソールを開きます。
- Model カタログに移動し、Anthropic モデル (Claude など) を選択します。
- ユースケースの詳細を送信するように求められます。フォームに入力して送信します。

Anthropic モデルへのアクセスは、フォームが正常に送信された直後に付与されます。完了したら、OpenClaw ダッシュボードで任意の Anthropic モデルを選択し、すぐに使用を開始できます。

> **Note**
>
> Amazon、Meta、Mistral AI、DeepSeek、Qwen のモデルは AWS Marketplace を通じて販売されないため、このステップは必要ありません。

詳細については、「Amazon Bedrock ユーザーガイド」の「Amazon Bedrock 基盤モデルへのアクセス」を参照してください。

HTTPS は OpenClaw インスタンスとどのように連携しますか?

OpenClaw インスタンスには、Let's Encrypt 証明書で保護された組み込み HTTPS エンドポイントが付属しています。インスタンスが作成されると、インスタンスの IPv4 アドレスに対して Let's Encrypt 証明書が自動的に発行されます。手動セットアップは必要ありません。

インスタンスの IP アドレスが変更された場合、SSL 証明書はどうなりますか?

OpenClaw インスタンスには、インスタンスの IP アドレスをモニタリングする証明書管理デーモン (`lightsail-manage-certd`) が組み込まれています。静的 IP をアタッチまたはデタッチするなど、IP アドレスが変更された場合、デーモンは変更を自動的に検出し、新しい IP アドレスの新しい Let's Encrypt 証明書を発行します。SSL 証明書に手動アクションは必要ありません。

注: ゲートウェイアクセストークンは変わりませんが、「ステップ 2: ブラウザを OpenClaw とペアリングする」の手順に従って、ブラウザを再度ペアリングする必要があります。

SSL 証明書はどのくらいの頻度で更新されますか?

OpenClaw インスタンス用に発行された Let's Encrypt 証明書は 7 日間有効です。証明書管理デーモンは、証明書の有効期限が切れる 2 日前に証明書を自動的に更新するため、インスタンスは中断や手動による介入なしに保護されます。

OpenClaw にプラグインをインストールできますか?

はい。OpenClaw はプラグインのインストールをサポートしており、一部のプラグインまたは設定の変更を有効にするにはOpenClaw ゲートウェイサービスを手動で再起動する必要がある場合があります。

プラグインのインストールまたは設定の更新後にゲートウェイを管理するには、OpenClaw インスタンスに SSH 接続し、次のコマンドを使用します。

- OpenClaw ゲートウェイサービスを停止します。 `openclaw gateway stop`
- OpenClaw ゲートウェイサービスを起動します。 `openclaw gateway start`
- サービスの現在のステータスを確認します。 `openclaw gateway status`

注: MOTD 1.0.0 (OpenClaw 2026.2.17) を使用している場合は、代わりに次のコマンドを使用します。

- OpenClaw ゲートウェイサービスを停止します。 `sudo systemctl stop openclaw-gateway`
- OpenClaw ゲートウェイサービスを起動します。 `sudo systemctl start openclaw-gateway`
- サービスの現在のステータスを確認します。 `sudo systemctl status openclaw-gateway`

ゲートウェイトークンが侵害された場合はどうなりますか?

トークンが公開された場合、例えば、ログに漏れたり、誤って共有されたり、プロンプトインジェクション攻撃によって公開されたりした場合、トークンを持つユーザーは、手動で再生成するまで OpenClaw ゲートウェイにアクセスできます。

ゲートウェイトークンは自動的にローテーションされますか?

はい。毎日 3:00 UTC に自動的にローテーションされます。このローテーションでは、ブラウザを OpenClaw インスタンスと再ペアリングする必要があります。

ゲートウェイトークンを手動でローテーションする方法を教えてください。

ゲートウェイトークンをローテーションするには:

- Lightsail コンソールから OpenClaw インスタンスに SSH 接続します。
- トークンを再生成するには、次のコマンドを実行します。

```
openclaw token rotate
```

- 古いトークンはすぐに無効になります。現在古いトークンとペアリングされているブラウザまたはクライアントは切断されます。

- ステップ 2: ブラウザを OpenClaw とペアリングするの手順に従って、新しいトークンを使用してブラウザを再度ペアリングします。

> **Tip**
>
> トークンをローテーションしたら、使用を再開する前に、すべての信頼されたデバイスが再ペアリングされていることを確認します。

自動トークンローテーションの仕組み

MOTD 2.0.0 には、ゲートウェイアクセストークンを毎日ローテーションすることでセキュリティを強化する自動トークンローテーションが含まれています。

重要な影響:

- トークンが自動的にローテーションされると、ペアリングされたすべてのブラウザとデバイスが切断されます。
- ステップ 2: ブラウザを OpenClaw とペアリングするの手順に従って、ブラウザを再度ペアリングする必要があります。

トークンをローテーションしない場合は、セキュリティ設定を変更して MOTD で無効にできます。

```
4. Daily token rotation
   Automatically rotates your access token every day at 03:00.
   Status: ✓ Enabled — token rotates daily at 03:00
```

メッセージングチャネル認証情報 (Telegram、WhatsApp、Slack) を更新するにはどうすればよいですか?

OpenClaw インスタンスに保存されているメッセージングプラットフォームトークンまたは認証情報、例えば、Telegram ボットトークン、WhatsApp セッション認証情報、Slack トークンが侵害された場合は、メッセージングチャネルへの不正アクセスを防ぐために、すぐに更新する必要があります。

接続されたチャネルの認証情報は、インスタンス`~/.openclaw/credentials/`の に保存されます。認証情報をローテーションするには:

- ソースで侵害されたトークンを取り消します。

  - Telegram: Telegram、メッセージ を開き@BotFather、 /revoke を使用して既存のボットトークンを無効にし、新しいトークンを生成します。
  - WhatsApp: スマートフォンの WhatsApp からリンクされたデバイスセッションをログアウトします (Settings → Linked Devices → OpenClaw セッションを選択 → Log out)。次に、QR コードペアリングフローを使用して再リンクします。
- OpenClaw インスタンスの認証情報を更新する: インスタンスに SSH 接続し、以下を実行します。

```
openclaw channels update
```

  更新するチャネルを選択し、プロンプトが表示されたら新しいトークンまたは認証情報を入力します。
- 更新されたチャネルを介してテストメッセージを送信して、チャネルが動作していることを確認します。

> **Note**
>
> メッセージング認証情報をローテーションしても、ゲートウェイトークンやその他の接続されたチャネルには影響しません。各認証情報は個別に管理されます。

setup-lightsail-openclaw-bedrock-role.shスクリプトの動作

これにより、OpenClaw インスタンスのみが Amazon Bedrock と AWS Marketplace で利用できる基盤モデルを使用できるようにする IAM ロールが作成されます。

スナップショットから OpenClaw インスタンスを復元するにはどうすればよいですか?

- 既存の OpenClaw スナップショットから新しいインスタンスを作成します。詳細については、「スナップショットからインスタンスの作成」を参照してください。
- Lightsail コンソールから新しい OpenClaw インスタンスに SSH 接続する
- 次のコマンドを実行して、*i-1234567890abcdef1* などのインスタンスのLightsailインスタンス ID を取得します。

```
TOKEN=`curl -s -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600"` && curl -w "\n" -s -H "X-aws-ec2-metadata-token:
 $TOKEN" http://169.254.169.254/latest/meta-data/instance-id
```

- 次のコマンドを実行して、インスタンスに関連付けられた IAM ロールを取得します。

```
grep 'role_arn' /home/ubuntu/.aws/config | head -1 | awk '{print $3}'
```

- 前のステップで取得したロールを IAM コンソールで検索します。例: `LightsailRoleFor-i-0d15d5483571b95bb`
- `Trust relationships` の選択
- `Edit trust policy` の選択
- 先ほど取得したインスタンス ID の ARN で信頼ポリシーを更新します。例: *`"arn:aws:sts::0123456789012:assumed-role/Amazon LightsailInstance/i-1234567890abcdef1"`*。
- `Update policy` の選択

OpenClaw インスタンスに AllowedOrigin を設定するにはどうすればよいですか?

AllowedOrigin は、OpenClaw ゲートウェイへの接続が許可されるウェブアドレス (オリジン) を制御するセキュリティ設定です。これにより、不正なウェブサイトがインスタンスにアクセスできなくなり、クロスオリジンセキュリティの問題から保護されます。

MOTD 2.0.0 (OpenClaw 2026.3.2 以降): AllowedOrigin は MOTD サービスによって自動的に管理されます。インスタンスの起動時または IP アドレスの変更時に、サービスは自動的に正しいオリジンを検出し、設定を更新します。手動アクションは必要ありません。

MOTD 1.0.0 (OpenClaw 2026.2.17): 特定のドメインから OpenClaw にアクセスする場合は、AllowedOrigin を手動で設定する必要があります。インスタンスに SSH 接続し、以下の手順に従って OpenClaw 設定ファイルを編集して、許可されたオリジンを追加します。

- Lightsail コンソールから OpenClaw インスタンスに SSH 接続する
- 設定ファイルを開きます。 `~/.openclaw/openclaw.json`
- AllowedOrigin 設定を追加または変更します。

```
{
    "gateway": {
```

```
        "controlUi": {
            "allowedOrigins": [
                "https://<your-domain.com>"
            ]
        }
    }
}
```

- OpenClaw ゲートウェイサービスを再起動します。 `sudo systemctl restart openclaw-gateway`

OpenClaw を最新バージョンに更新するにはどうすればよいですか?

OpenClaw ゲートウェイを最新バージョンに更新するには:

- OpenClaw インスタンスへの SSH
- update コマンドを実行します。 `sudo openclaw update`

重要な注意事項:

- OpenClaw ブループリントは、ゲートウェイをインスタンスにグローバルにインストールするため、sudo 権限が必要です。
- update コマンドが完了すると、ルートユーザーとしてゲートウェイを起動しようとしますが、ゲートウェイをubuntuユーザーとして実行する必要があるため、失敗します。
- 更新後にゲートウェイを手動で再起動する必要があります。 `openclaw gateway restart`
- OpenClaw コントロール UI ダッシュボードの「更新」ボタンは、sudo権限がないため機能しません。

静的 IP アドレスをアタッチすると、デバイスのペアリングはどうなりますか?

OpenClaw インスタンスに静的 IP アドレスをアタッチすると、インスタンスの IP アドレスが変更されます。これは、デバイスのペアリングに重要な影響を与えます。

- 以前にペアリングされたすべてのブラウザとデバイスは切断されます
- ゲートウェイトークンは有効ですが、接続エンドポイントは変更されています
- 静的 IP をアタッチした後、すべてのブラウザとデバイスを再度明示的にペアリングする必要があります

デバイスを再ペアリングするには:

- インスタンスへの SSH (SSH 接続は新しい静的 IP で機能します)
- ステップ 2 のペアリング手順に従って、各ブラウザを再接続します。
- メッセージングチャネル (Telegram、WhatsApp) では、ペアリングの再承認が必要になる場合もあります。

ツールを有効にするためのサンドボックスアクセス許可を付与する方法を教えてください。

デフォルトでは、OpenClaw は分離された Docker コンテナ環境 (サンドボックス) でツールとプラグインを実行し、潜在的に有害なオペレーションからインスタンスを保護します。この分離により、システムコマンド、ファイルシステムアクセス、ネットワーク接続、ホストシステムリソースなど、アクセスできるツールが制限されます。

これにより強力なセキュリティが提供されますが、一部のツールでは、適切に機能するために制限の少ない設定が必要になる場合があります。たとえば、ウェブスクレイピングツールにはネットワークアクセスが必要であり、ファイル管理ツールにはより広範なファイルシステムアクセスが必要です。これらのアクセス許可がない場合、サンドボックスは主に機能が制限された基本的なチャットボットとして機能します。

サンドボックスの制限を小さくするには:

- Lightsail コンソールから OpenClaw インスタンスに SSH 接続する
- 次のコマンドを実行して、ツールの実行設定を構成します。

```
openclaw config set tools.exec.host gateway
openclaw config set tools.exec.ask off
openclaw config set tools.exec.security full
```

- 変更を有効にするには、OpenClaw ゲートウェイサービスを再起動します。

```
openclaw gateway restart
```

これらの設定の動作:

- `tools.exec.host gateway` - 分離された Docker コンテナではなくゲートウェイホストでツールを直接実行できるようにし、システムコマンドとリソースへのアクセスを許可します。

- `tools.exec.ask off` - ツールの実行前にアクセス許可プロンプトを無効にし、手動承認なしでツールを自動的に実行できるようにします
- `tools.exec.security full` - ツール実行のセキュリティレベルを設定します

セキュリティ上の考慮事項: これらの設定により、ツールとシステム間の分離が大幅に削減されます。これらの設定は、使用しているツールを信頼する場合にのみ設定します。制限の少ないサンドボックス設定でツールを実行すると、ツールが侵害されたり悪意のある場合に、インスタンスがセキュリティリスクにさらされる可能性があります。

MOTD バージョンの違いは何ですか?

OpenClaw インスタンスは、作成された日時に応じて異なる MOTD (Message of the Day) バージョンを使用します。知っておくべきことは次のとおりです。

MOTD 1.0.0 (OpenClaw 2026.2.17):

- ゲートウェイ管理: を使用する `sudo systemctl start/stop/status openclaw-gateway`
- トークンローテーション: 手動のみ (を使用`openclaw token rotate`)

MOTD 2.0.0 (OpenClaw 2026.3.2 以降):

- ゲートウェイ管理: 簡易コマンドを使用する `openclaw gateway start/stop/status`
- トークンローテーション: 毎日の自動ローテーション

MOTD バージョンを確認する方法: インスタンスに SSH 接続し、表示されるウェルカムメッセージを確認します。MOTD バージョンが上部に表示されます。

Lightsail に Plesk ホストスタックをデプロイする。

> **知っていましたか?**
>
> Lightsail は、毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを自動的に最新のスナップショットに置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

Amazon Lightsail で Plesk インスタンスを作成する方法と、ユーザー名とパスワードを作成して Plesk パネルに初めてサインインする方法について説明します。また、Plesk インスタンスが起動して実行された後に、Plesk インスタンスに接続して設定する方法も説明します。

> **Important**
>
> [Ubuntu で Plesk ホストスタック (BYOL)] ブループリントで起動されたインスタンスには、30 日間のトライアルライセンスが適用されます。30 日後に Plesk アプリケーションを引き続き使用するには、Plesk からライセンスを購入する必要があります。詳細については、「ステップ 8: Plesk ライセンスを購入する」を参照してください。

Lightsail で Plesk ホスティングスタックを使用すると、次のようなタスクを実行できます。

- WP Toolkit のグラフィカルインターフェイスを使用して WordPress サイト管理を自動化する
- Let's Encrypt を使用して無料の SSL 証明書でサイトを保護し、HTTPS トラフィックを設定する
- FTP を使用してインスタンス間でファイルを転送する
- ウェブベースのツール (Plesk Firewall、Logs、ModSecurity) を使用してサーバーをモニタリングおよび保護する
- Docker プロキシルールを使用して、特定のドメインから指定されたコンテナポートにインバウンドトラフィックをルーティングする

Plesk ホスティングスタックをデプロイする際の考慮事項

Plesk インスタンスのデプロイを開始する前に、ウェブサイトのドメインを登録する必要があるかどうか、およびその管理方法を確認してください。IP アドレスではなく論理名 (`http://`

example.com など) でウェブサイトにアクセスするには、登録済みドメインが必要です。また、証明書 (HTTPS トラフィックに必要) を使用してウェブサイトへの接続を保護するには、ドメイン内に DNS 名を作成する必要があります。

ドメインの設定と管理に関する以下のオプションを確認してください。

- 使用するドメインが Lightsail に登録済みである場合は、このチュートリアルのステップを開始できます。
- DNS レコードを管理する別のレジストラを持つドメインがある場合は、このチュートリアルのステップを開始できます。それ以外の場合は、ドメインの DNS レコードの管理を引き渡すことができます。これにより、Lightsail 内のすべてのコンピューティングリソースと DNS リソースをより効率的に管理できます。
- 使用するドメインがない場合は、ウェブサイトでドメインを検索して登録できます。Lightsail でドメインを登録する方法の詳細については、「Register and manage domains for your website in Lightsail」を参照してください。

ステップ 1: Plesk インスタンスの作成

Lightsail で Plesk インスタンスを作成するには、以下の手順を実行します。

1. https://lightsail.aws.amazon.com/ でLightsailコンソールにサインインします。
2. [インスタンス] ホームページで、[インスタンスの作成] を選択します。
3. インスタンスの場所 (AWS リージョン とアベイラビリティーゾーン) を選択します。

   変更 AWS リージョン とアベイラビリティーゾーンを選択して、別の場所にインスタンスを作成します。
4. 必要に応じて、アベイラビリティーゾーンを変更できます。

   [アベイラビリティーゾーンの変更] を選択する。
5. [アプリ + OS] で、[Ubuntu の Plesk ホスティングスタック (BYOL)] を選択します。
6. 使用可能なインスタンスプランを選択します。

> **Tip**
>
> ウェブサイトのパフォーマンスを最適化するには、インストールする予定のプラグインと拡張機能のリソース需要に対応できるインスタンスプランを選択します。後でより多くの処理能力が必要な場合は、スナップショットからインスタンスをサイズアップでき

ます。プラグインと拡張機能の要件の詳細については、それぞれのベンダーのドキュメントを参照してください。

7. インスタンスの名前を入力します。

   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

8. (オプション)[新しいタグの追加] を選択してインスタンスにタグを追加します。必要に応じてこの手順を繰り返して、他のタグを追加します。請求やリソースの整理など、タグの使用方法の詳細については、「タグ」を参照してください。

   - [キー] には、タグキーを入力します。
   - (オプション) [値] にタグの値を入力します。

9. [インスタンスの作成] を選択します。

インスタンスの作成後、プロビジョニングして使用可能になるまで数分かかります。

> **Tip**
>
> Plesk の初期セットアッププロセスを合理化するために、現時点ではインスタンスに静的 IP を割り当てないでください。これは後のステップで行います。現時点で Plesk インスタンスに静的 IP をアタッチした場合、古いパブリック IP アドレスを使用するワンタイムログイン URL を取得することがあります。これは、インスタンスが静的 IP アドレスではなく、最初に割り当てられたパブリック IP アドレスを使用する可能性があるため、発生する可能性があります。

ステップ 2: 初めて Plesk ユーザーインターフェースにサインインする

> **Tip**
>
> このチュートリアルを完了するまで、Lightsail コンソールと Plesk ユーザーインターフェイスのブラウザタブを開いたままにします。これは、後に各タブで複数のステップを実行するためです。

以下の手順を使用して、管理者として Plesk ユーザーインターフェイスにアクセスするためのワンタイムログイン URL を取得します。このインターフェイスでは、ウェブサイトを作成および管理します。

ワンタイムログイン URL を取得するには

1. 左のナビゲーションペインの [インスタンス] を選択してください。
2. 作成した Plesk インスタンスの名前を選択します。
3. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。

   > **Note**
   >
   > インスタンスの起動後に接続速度が速すぎると、接続または SSH キーのエラーが発生する可能性があります。接続中にこれらの問題が発生した場合は、数分待ってからもう一度試してください。

4. 接続後に、次のコマンドを入力してワンタイムログイン URL を取得します。

   ```
   sudo plesk login | grep plesk.page
   ```

   ワンタイムログイン URL を含む以下の例のようなレスポンスが表示されます。

   ```
   https://heuristic-bassi.192-0-2-0.plesk.page/login?secret=ce-
   EXAMPLE298fc1c149afbf4c8996fb92427
   ```

> **Note**
>
> パブリック IP アドレスが変更された場合 (例えば、起動後に静的 IP を割り当てた場合)、ログインページにアクセスするには、返された URL を新しいアドレスと一致するように変更する必要があります。

5. ブラウザベースの接続ウィンドウでワンタイムログイン URL を選択し、コピーします。
6. ウェブブラウザに URL を貼り付けて、Plesk ログインページにアクセスします。
7. ページの手順に従って Plesk のサインイン認証情報を作成します。初めてサインインするときに、Plesk にドメインを追加するオプションが表示されます。

ステップ 3: Plesk インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。パブリック IP アドレスが変わらないようにするには、静的 IP アドレスを作成してインスタンスにアタッチします。後のステップでは、静的 IP アドレスをドメイン名にマッピングします。このマッピングを設定すると、インスタンスを停止および起動するたびにドメインの DNS レコードを更新する必要はありません。

> **Note**
>
> Lightsail インスタンスにアタッチできる静的 IP は 1 つのみです。インスタンスにアタッチされている場合のみ、静的 IP アドレスは無料です。

Plesk で静的 IP をアタッチしてパブリック IP アドレスを更新するには

1. https://lightsail.aws.amazon.com/ でLightsailコンソールにサインインします。
2. Lightsail コンソールの[インスタンス]のホームページで、Plesk インスタンスの名前を選択します。
3. [ネットワーキング] で、[静的 IP をアタッチする] を選択します。
4. 静的 IP アドレスを作成してアタッチする
   - リージョンに静的 IP が存在しない場合に作成してアタッチするには:
     a. 静的 IP を識別する名前を入力します。

b. [作成およびアタッチ] を選択します。
c. [続行] をクリックしてください。

- リージョンに既に静的 IP がある場合に、新しい静的 IP を作成してアタッチするには:

a. [静的 IP を選択する] メニューを選択して、使用可能なオプションを表示します。
b. ドロップダウンメニューで [新しい静的 IP を作成] を選択します。
c. 静的 IP を識別する名前を入力します。
d. [作成およびアタッチ] を選択します。
e. [続行] をクリックしてください。

- リージョンで既存の静的 IP を使用するには:

a. [静的 IP を選択する] メニューを選択して、使用可能なオプションを表示します。
b. ドロップダウンメニューで既に使用可能な静的 IP を選択します。
c. [アタッチ] を選択してください。

パブリック IP アドレスが変更されると、`https://StaticIPAddress:8443` に似た URL を使用して Plesk ユーザーインターフェイスにアクセスします。`StaticIPAddress` をインスタンスにアタッチされた静的 IP アドレスに置き換えます。例えば、`https://192.0.2.0:8443`。ログインページで、前に作成したユーザー名とパスワードを入力して Plesk ユーザーインターフェースにサインインします。

> **Note**
>
> IP アドレスを使用して URL に接続すると、接続がプライベートではないか、セキュリティで保護されていないか、またはセキュリティ上のリスクがあることを示すブラウザの警告が表示されることがあります。これは、Plesk インスタンスに適用されている SSL/TLS 証明書が、使用されている新しいパブリック IP と一致しないために発生します。ブラウザウィンドウで、[Advanced] (詳細設定)、[Details] (詳細)、または [More information] (詳細情報) を選択して、使用可能なオプションを表示します。次に、プライベートまたは安全でない場合でも、ウェブサイトにアクセスすることを選択します。

ステップ 4: Plesk でパブリック IP アドレスを更新する

インスタンスに静的 IP アドレスが割り当てられたので、Plesk で使用されるパブリック IP アドレスを更新します。これにより、Plesk はインスタンスに関連付けられなくなった動的アドレスでは

なく、更新されたパブリック IP アドレスを使用します。このステップを実行しないと、Plesk ユーザーインターフェイスに不一致に関する警告が表示されることがあります。

Plesk でパブリック IP アドレスを更新するには

1. 割り当てられた新しいパブリック IP アドレスを使用して Plesk ユーザーインターフェイスにアクセスします。例えば、`https://192.0.2.0:8443`。
2. ステップ 2: 初めて Plesk ユーザーインターフェースにサインインする で以前に作成したサインイン認証情報を使用して認証します。
3. Plesk ユーザーインターフェイスの左下隅で、[ビューを変更] を選択し、[パワーユーザービューに切り替え] を選択します。
4. Plesk ユーザーインターフェイスの左側のナビゲーションペインで、[ツールと設定] を選択します。
5. [ツールとリソース] で、[IP アドレス] を選択します。
6. [パブリック IP の更新] を選択します。

[パブリック IP アドレス]列に、インスタンスにアタッチされた静的 IP と一致する値が表示されるようになりました。

ステップ 5: Plesk インスタンスにウェブサイトを追加する

Plesk ユーザーインターフェイス内に 1 つ以上のドメイン名 (ウェブサイト) をマッピングできます。Plesk インスタンスにドメインを追加すると、コンテンツのアップロード、Presence Builder の使用、およびコンテンツ管理システム (CMS) のインストールが可能になります。

Plesk に新しいドメインを追加するには

1. Plesk ユーザーインターフェイスの左側のナビゲーションペインで、「ウェブサイトとドメイン」 を選択します。
2. [ドメインを追加する] を選択します。
3. ウェブサイトの作成方法のオプションを選択します。
4. ドメイン名を入力し、追加の必須情報フィールドに入力します。
5. [ドメインを追加する] を選択します。

詳細については、Plesk 管理者ガイドの「Adding and Removing Domains」を参照してください。

ステップ 6: ドメイン名を Plesk インスタンスにマッピングする

作成したドメイン名を使用して、ドメインを Plesk インスタンスにマッピングして Plesk ユーザーインターフェイスにアクセスできます。Plesk ユーザーインターフェース内で複数のドメインをマッピングすることも可能で、ウェブサイトを管理する際に使用できます。Plesk ユーザーインターフェース内の複数のドメインのマッピングについての詳細は、Plesk Documentation and Help Portalの「Adding a Domain in Plesk」を参照してください。

ドメイン名 (`example.com` など) をインスタンスにマッピングするには、ドメインのドメインネームシステム (DNS) に A レコードを追加します。DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。詳細については、「Lightsail ドメインの DNS 管理を転送する」を参照してください。

ドメイン名を Lightsail の Plesk インスタンスにマッピングするには

1. Amazon Lightsail コンソール (https://console.aws.amazon.com/lightsail/) を開きます。
2. Plesk ユーザーインターフェイスの左側のナビゲーションペインで、[ドメインと DNS] を選択します。
3. DNS ゾーンの名前を選択します。
4. [DNS records] (DNS レコード) タブを選択します。
5. [レコードの追加] を選択します。
6. [レコード名] に、値 (`www` など) を入力します。
7. [解決先] に、Plesk インスタンスにアタッチされている静的 IP アドレスを入力します。
8. [保存] を選択します。

これで、設定したドメイン名を使用して Plesk ウェブサイトにアクセスできるようになります。

ステップ 7: Plesk ドキュメントを読む

Plesk のドキュメントで、ウェブサイトの管理、Plesk パネルのカスタマイズなどの方法を確認します。

詳細については、Plesk ドキュメントとヘルプポータルの「Plesk ウェブサイト管理の開始」を参照してください。

ステップ 8: Plesk ライセンスを購入する

Plesk インスタンスには、30 日間のトライアルライセンスが含まれています。30 日後に引き続き使用するには、Plesk からライセンスを購入する必要があります。詳細については、Plesk ウェブサイトの「価格設定」を参照してください。

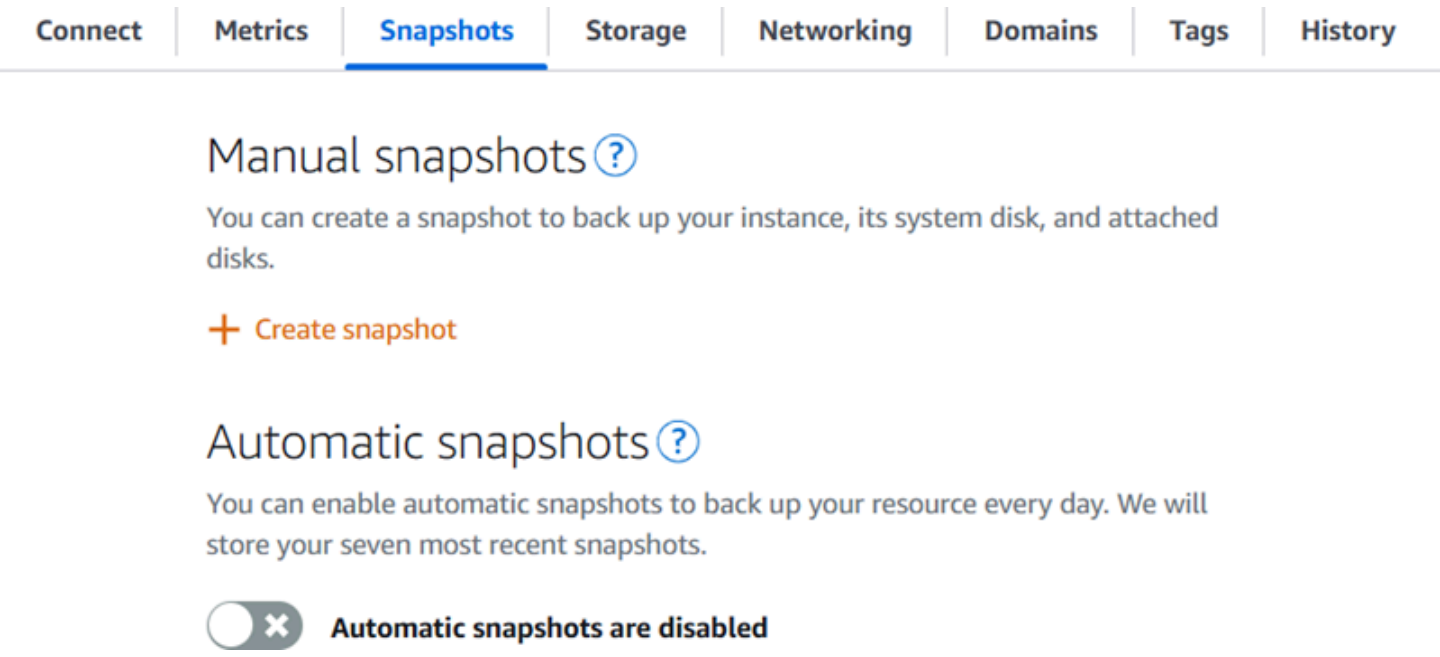
Plesk からライセンスを購入した後、ライセンスをインストールする必要があります。Plesk ライセンスをインストールするには、Plesk サポートウェブサイトの「How to install the Plesk license」を参照してください。

ステップ 9: Plesk インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。



Lightsail で PrestaShop ウェブサイトを設定する

> **知っていましたか?**
>
> Lightsail は毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを最新のスナップショットに自動的に置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

Amazon Lightsail で起動した PrestaShop インスタンスの使用を開始するステップについて説明します。

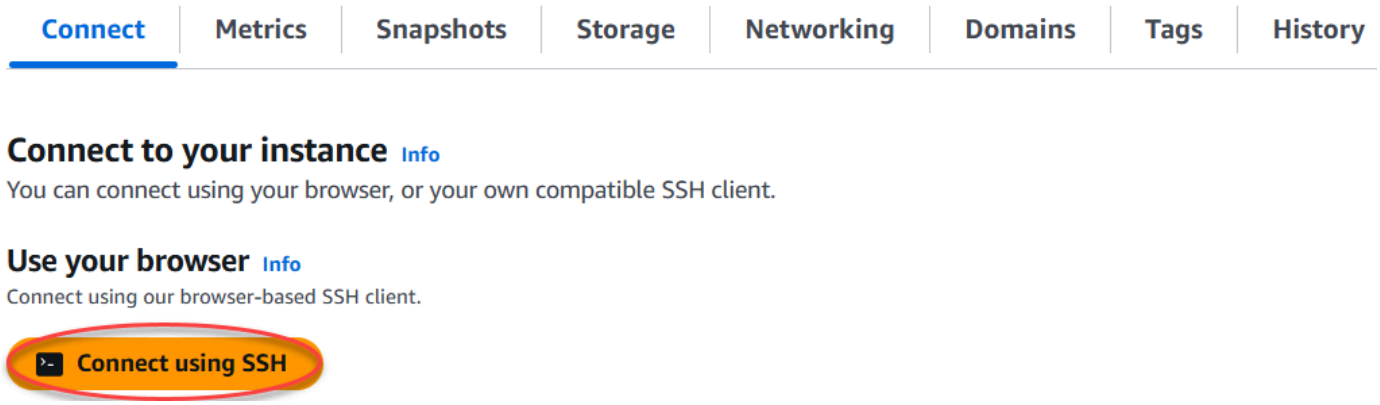
目次

- ステップ 1: PrestaShop ウェブサイトのデフォルトのアプリケーションパスワードを取得する
- ステップ 2: PrestaShop インスタンスに静的 IP アドレスを添付する
- ステップ 3: PrestaShop ウェブサイトの管理ダッシュボードにサインインする
- ステップ 4: 登録済みドメイン名へのトラフィックを PrestaShop ウェブサイトに送信する
- ステップ 5: PrestaShop ウェブサイトの HTTPS を設定する
- ステップ 6: メール通知用の SMTP を設定する
- ステップ 7: Bitnami と PrestaShop のドキュメントを読む
- ステップ 8: PrestaShop インスタンスのスナップショットを作成する

ステップ 1: PrestaShop ウェブサイトのデフォルトのアプリケーションパスワードを取得する

次のステップを完了して、PrestaShop ウェブサイトのデフォルトのアプリケーションパスワードを取得します。

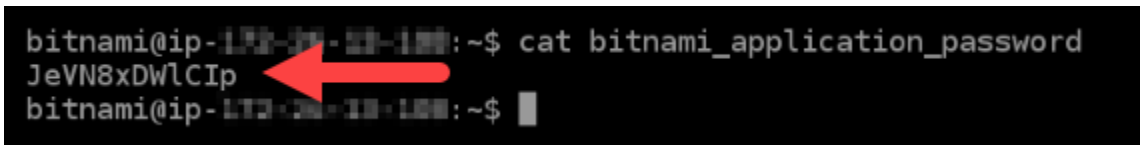
1. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。



2. 接続後に、次のコマンドを入力してデフォルトのアプリケーションのパスワードを取得します。

```
cat $HOME/bitnami_application_password
```

アプリケーションのデフォルトパスワードを含んだ、次の例のようなレスポンスが表示されます。このパスワードを安全な場所に保存します。このチュートリアルの次のセクションで、PresTaShop ウェブサイトの管理ダッシュボードにサインインする際に使用します。

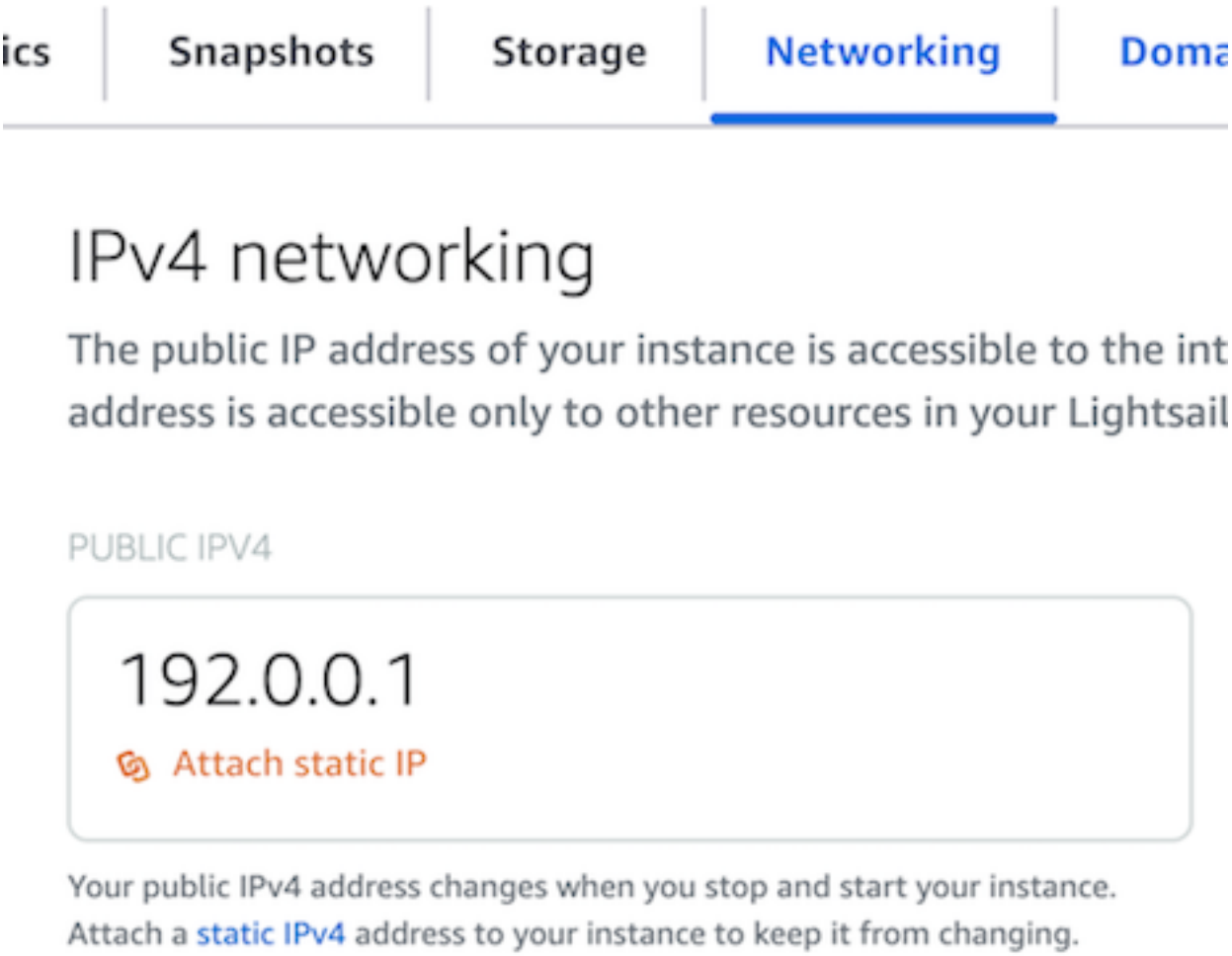


詳細については、「Amazon Lightsailの Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

ステップ 2: PrestaShop インスタンスに静的 IP アドレスを添付する

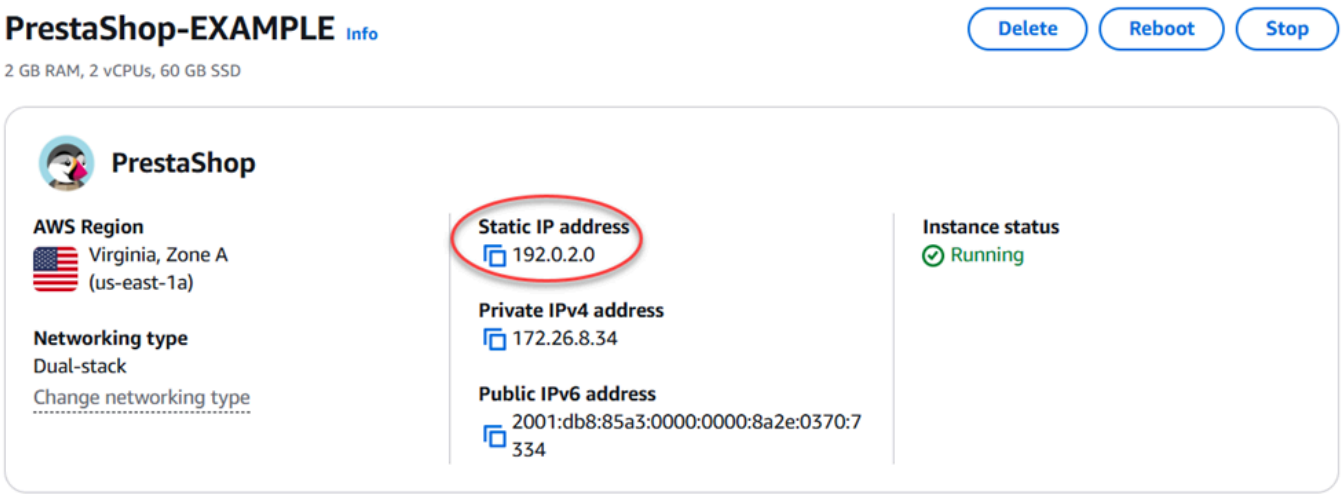
インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。

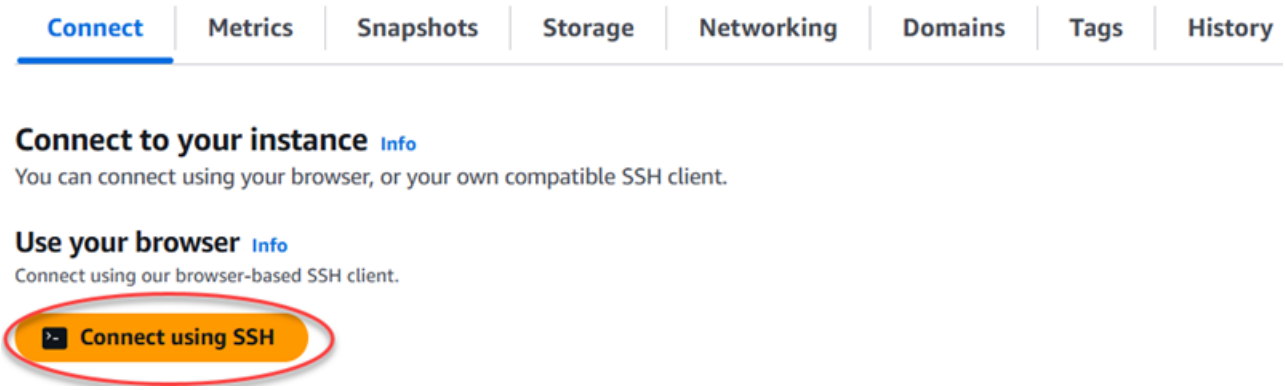


新しい静的 IP アドレスがインスタンスに添付されたら、次の手順を実行して、PrestaShop ソフトウェアに新しい静的 IP アドレスを認識させる必要があります。

1. インスタンスの静的 IP アドレスは書き留めておきます。このIP アドレスはインスタンス管理ページの ヘッダーセクションに表示されます。



2. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。



3. 接続後に、次のコマンドを入力します。*<StaticIP>* はインスタンスの新しい静的 IP アドレス必ず置き換えてください。

```
sudo /opt/bitnami/configure_app_domain --domain <StaticIP>
```

例:

```
sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
```

次の例のようなレスポンスが表示されます。これで、PrestaShop ソフトウェアは新しい静的 IP アドレスを認識している状態になります。

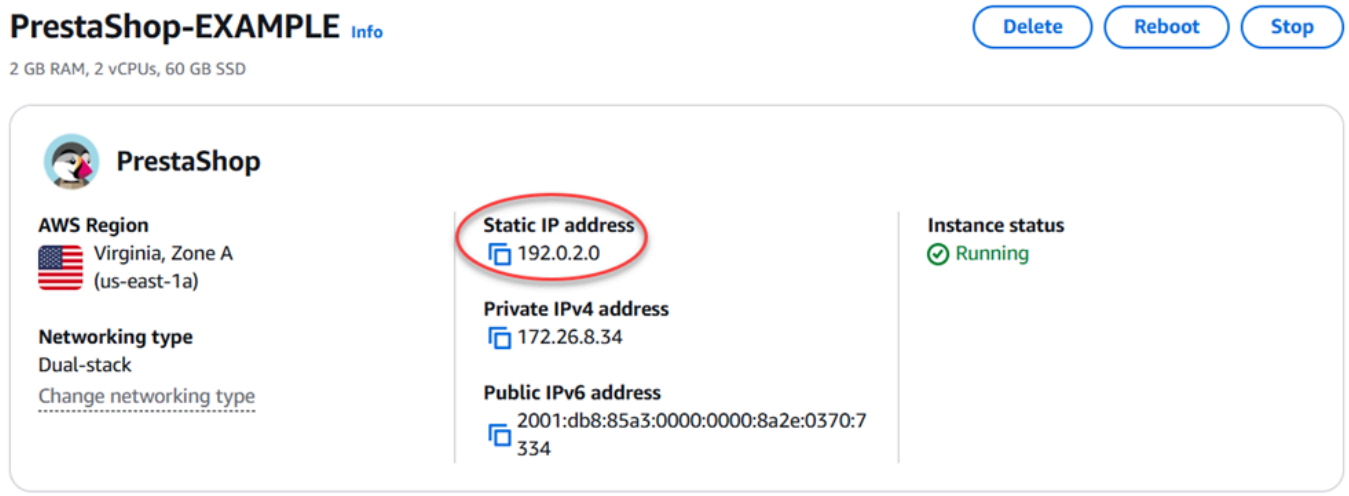
```
bitnami@ip-[redacted]:~$ sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
Configuring domain to 203.0.113.0
2021-03-12T15:49:22.000Z - info: Saving configuration info to disk
prestashop 15:49:22.41 INFO  ==> Trying to connect to the database server
prestashop 15:49:22.44 INFO  ==> Updating hostname in database
prestashop 15:49:22.46 INFO  ==> Purging cache
Disabling automatic domain update for IP address changes
```

> **Note**
>
> PrestaShop は現在、IPv6 アドレスをサポートしていません。インスタンスの IPv6 を有効にすることはできますが、PrestaShop ソフトウェアは IPv6 ネットワーク経由のリクエストに応答しません。

ステップ 3: PrestaShop ウェブサイトの管理ダッシュボードにサインインする

PrestaShop のウェブサイトにアクセスして管理ダッシュボードにサインインするには、以下のステップを実行します。サインインするには、このガイドの前のセクションで取得したデフォルトのユーザー名 (`user@example.com`) とデフォルトのアプリケーションパスワードを使用します。

1. Lightsail コンソールで、インスタンス管理ページのヘッダー部分に記載されているパブリック IP アドレスないし静的 IP アドレスを書き留めます。



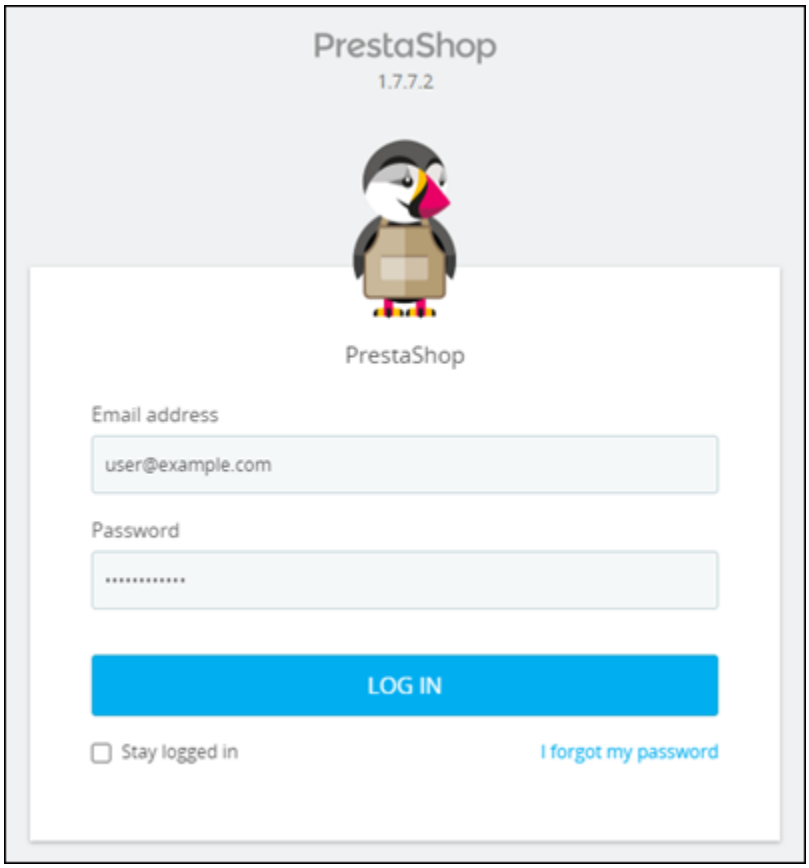
2. 以下のアドレスまで移動して、PrestaShop ウェブサイトの管理ダッシュボードのサインインページにアクセスします。*`PublicIPAddress`* は、インスタンスのパブリック IP アドレスないし静的 IP アドレスに必ず置き換えます。

```
http://<InstanceIpAddress>/administration
```

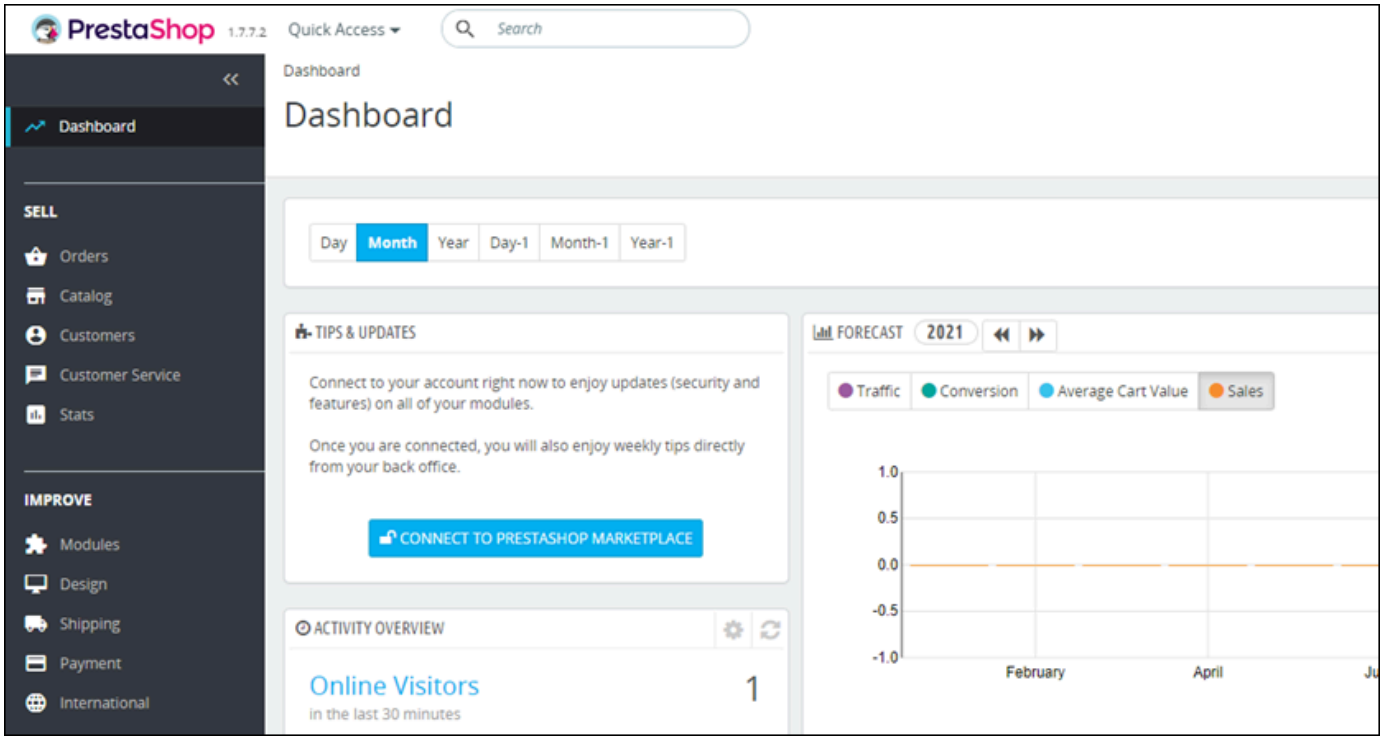
例:

```
http://203.0.113.0/administration
```

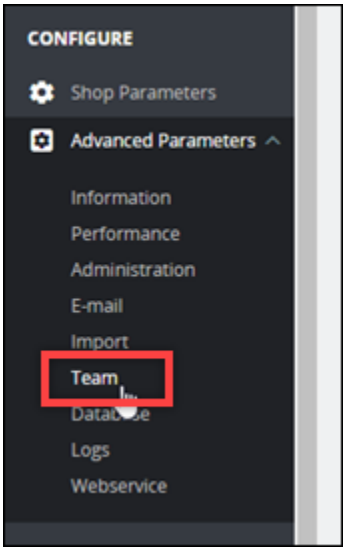
3. デフォルトのユーザー名 (user@example.com) と、このガイドの前のセクションで取得したデフォルトのアプリケーションパスワードを入力して [ログイン] を選択します。



PrestaShop の管理ダッシュボードが表示されます。



PrestaShop ウェブサイトの管理ダッシュボードへサインインする際に使用するデフォルトのユーザー名またはパスワードを変更するには、ナビゲーションペインの [アドバンストパラメータ] を選択し、[チーム] を選択します。詳細については、「PrestaShop ドキュメント」の「User Guide PrestaShop」を参照してください。



管理ダッシュボードの詳細については「PrestaShop ドキュメント」の「User Guide PrestaShop」を参照してください。

ステップ 4: 登録済みドメイン名へのトラフィックを PrestaShop ウェブサイトに送信する

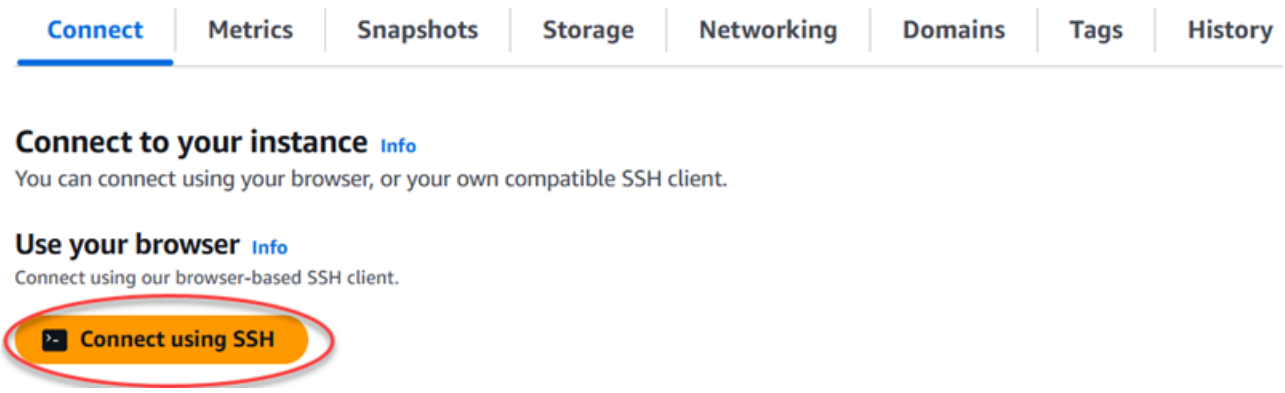
example.com などの登録済みドメイン名のトラフィックを PresTaShop ウェブサイトに送信するには、ドメインのドメインネームシステム (DNS) にレコードを追加します。DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページの [ドメインと DNS] タブで、[DNS ゾーンの作成] を選択し、ページに記載される手順に従います。

詳細については、「Lightsail で DNS ゾーンを作成し、ドメインの DNS レコードを管理する」を参照してください。

ドメイン名へのトラフィックがインスタンスにルーティングされたら、次の手順を実行して、PrestaShop ソフトウェアにドメイン名を認識させます。

1. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。



2. 接続後に、次のコマンドを入力します。*<DomainName>* は、インスタンスにトラフィックを送信しているドメイン名に必ず置き換えてください。

```
sudo /opt/bitnami/configure_app_domain --domain <DomainName>
```

例:

```
sudo /opt/bitnami/configure_app_domain --domain www.example.com
```

次の例のようなレスポンスが表示されます。これで、PrestaShop ソフトウェアはドメイン名を認識した状態になります。

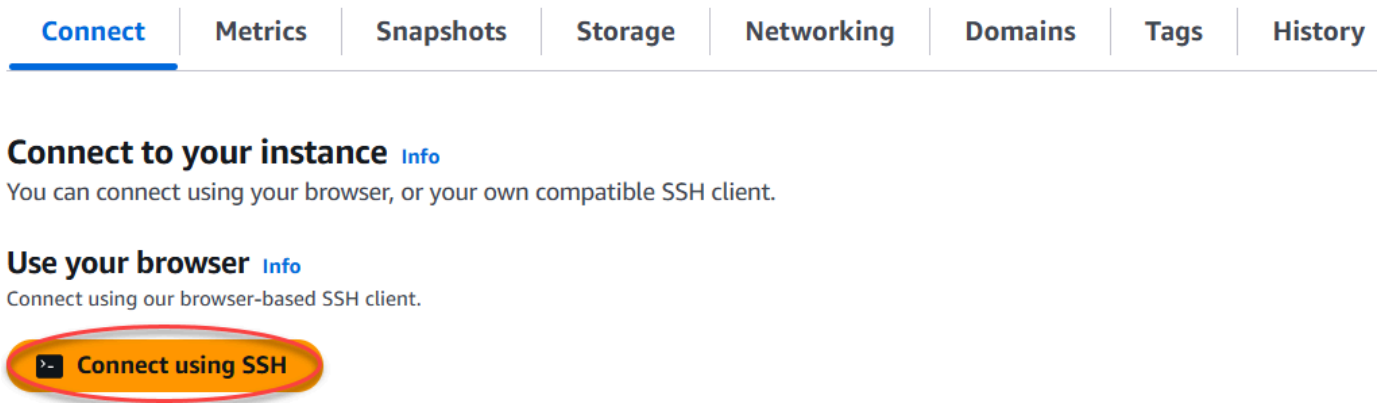
```
bitnami@ip-[redacted]:~$ sudo /opt/bitnami/configure_app_domain --domain www.example.com
Configuring domain to www.example.com
2021-03-12T15:49:22.000Z - info: Saving configuration info to disk
prestashop 15:49:22.41 INFO  ==> Trying to connect to the database server
prestashop 15:49:22.44 INFO  ==> Updating hostname in database
prestashop 15:49:22.46 INFO  ==> Purging cache
Disabling automatic domain update for IP address changes
```

ステップ 5: PrestaShop ウェブサイトの HTTPS を設定する

PrestaShop ウェブサイトで HTTPS を設定するには、以下の手順を実行します。次の手順では、Bitnami HTTPS 設定ツール (bncert) を使用する方法を示しています。このツールは、SSL/TLS 証明書のリクエスト、リダイレクトの設定 (例: HTTP から HTTPS)、および証明書の更新を行うためのコマンドラインツールです。

> **Important**
>
> bncert ツールは現在、 PrestaShop インスタンスのパブリック IP アドレスにトラフィックをルーティングしているドメインに対してのみ証明書を発行します。これらの手順を開始する前に、PrestaShop Web サイトで使用したいすべてのドメインの DNS に、 DNS レコードを追加してください。

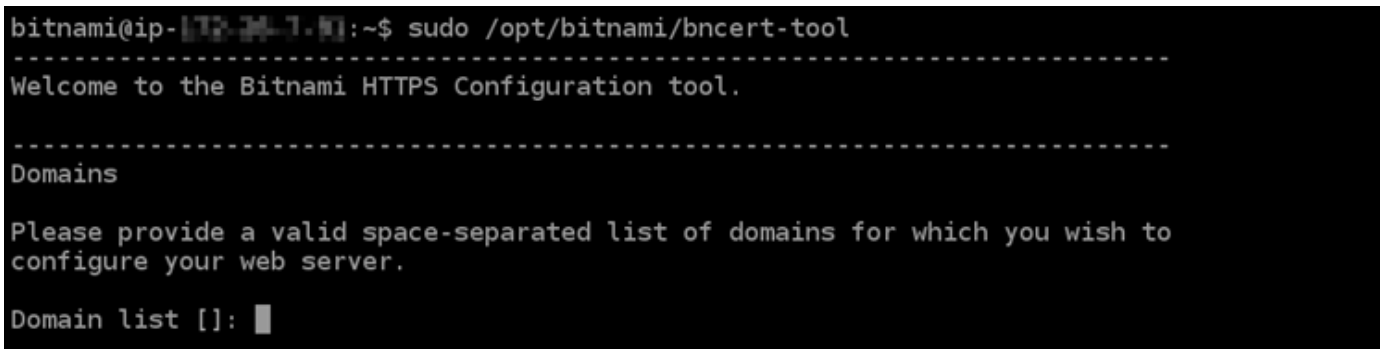
1. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。



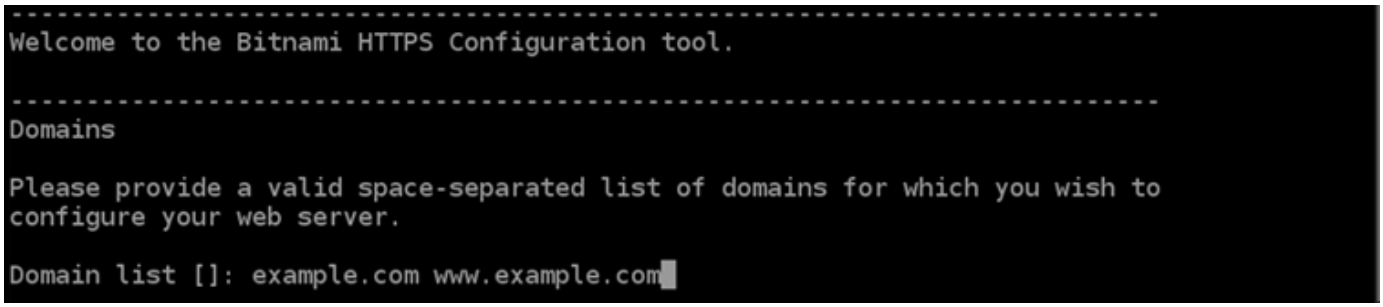
2. 接続後に、次のコマンドを入力して bncert-tool をスタートします。

```
sudo /opt/bitnami/bncert-tool
```

次の例のようなレスポンスが表示されます:



3. 次の例に記載されているように、プライマリドメイン名と代替ドメイン名をスペースで区切って入力します。



4. bncert ツールは、ウェブサイトのリダイレクトをどのように設定したいかをユーザーに確認します。使用できるオプションは、次のとおりです。

   - HTTP から HTTPS へのリダイレクトを有効にする - HTTP バージョンのウェブサイトを閲覧するユーザー (例: `http:/example.com`) を自動的に HTTPS バージョン (例: `https://example.com`) にリダイレクトするかどうかを決定します。すべての訪問者が暗号化された接続を使用するように強制されるため、このオプションを有効にすることをお勧めします。Y を入力して Enter を押すと、有効になります。
   - www なしから www ありへのリダイレクトの有効化 - ドメインの頂点 (例: `https://example.com`) まで閲覧するユーザー を自動的にドメインの www サブドメイン (例: `https://www.example.com`) にリダイレクトするかを指定します。このオプションを有効にすることをお勧めします。ただし、ドメインの頂点を Google のウェブマスターツールなどの検索エンジンツールで希望のウェブサイトアドレスとして指定した場合、または頂点が IP を直接指しており、www のサブドメインが CNAME レコードを介してリファレンスしている場合は、無効にして代替オプションを有効にすることをお勧めします (www ありから www なしへのリダイレクトを有効化) 。Y を入力し、Enter を押して有効にします。

- www ありから www なしへのリダイレクトを有効にする - ドメインの www サブドメイン (例: `https://www.example.com`) まで閲覧するユーザーを、自動的にドメインの頂点 (例: `https://example.com`) にリダイレクトするかを指定します。www なしから www ありへのリダイレクトを有効にした場合は、これを無効にすることをお勧めします。N を入力し、Enter を押して無効にします。

選択した結果は次の例のようになります。

```
Enable/disable redirections

Please select the redirections you wish to enable or disable on your Bitnami
installation.



Enable HTTP to HTTPS redirection [Y/n]: Y



Enable non-www to www redirection [Y/n]: Y



Enable www to non-www redirection [y/N]: N
```

5. これから実行される変更が一覧表示されます。Y と入力し、Enter を押して確認し、続行します。

```
Changes to perform

The following changes will be performed to your Bitnami installation:

1. Stop web server
2. Configure web server to use a free Let's Encrypt certificate for the domains:
example.com www.example.com
3. Configure a cron job to automatically renew the certificate each month
4. Configure web server name to: example.com
5. Enable HTTP to HTTPS redirection (example: redirect http://example.com to
https://example.com)
6. Enable non-www to www redirection (example: redirect example.com to
 www.example.com)
7. Start web server once all changes have been performed



Do you agree to these changes? [Y/n]: Y
```

6. Let's Encrypt 証明書に関連付けるメールアドレスを入力し、Enter を押します。

```
Create a free HTTPS certificate with Let's Encrypt

Please provide a valid e-mail address for which to associate your Let's Encrypt
certificate.

Domain list: example.com www.example.com

Server name: example.com

E-mail address []:
```

7. Let's Encrypt サブスクライバー合意書を確認します。Y と入力し、Enter を押して契約に同意し、続行します。

```
The Let's Encrypt Subscriber Agreement can be found at:

https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf

Do you agree to the Let's Encrypt Subscriber Agreement? [Y/n]:
```

これらのアクションは、証明書のリクエストや指定したリダイレクトの設定など、インスタンスで HTTPS を有効にするために実行されます。

```
Performing changes to your installation

The Bitnami HTTPS Configuration Tool will perform any necessary actions to your
Bitnami installation. This may take some time, please be patient.

|
```

次の例のようなメッセージが表示された場合は、証明書は正常に発行され、検証され、インスタンスでリダイレクトが正常に設定されています。

```
Success

The Bitnami HTTPS Configuration Tool succeeded in modifying your installation.

The configuration report is shown below.

Backup files:
* /opt/bitnami/apache2/conf/httpd.conf.back.202005290035
* /opt/bitnami/apache2/conf/bitnami/bitnami-apps-prefix.conf.back.202005290035
* /opt/bitnami/apache2/conf/bitnami/bitnami.conf.back.202005290035

Find more details in the log file:

/tmp/bncert-202005290035.log

If you find any issues, please check Bitnami Support forums at:

https://community.bitnami.com

Press [Enter] to continue:
```

bncert ツールは、有効期限が切れる前、 80 日ごとに証明書の自動更新を実行します。次の一連のステップに進み、PrestaShop ウェブサイトで HTTPS の有効化を完了します。

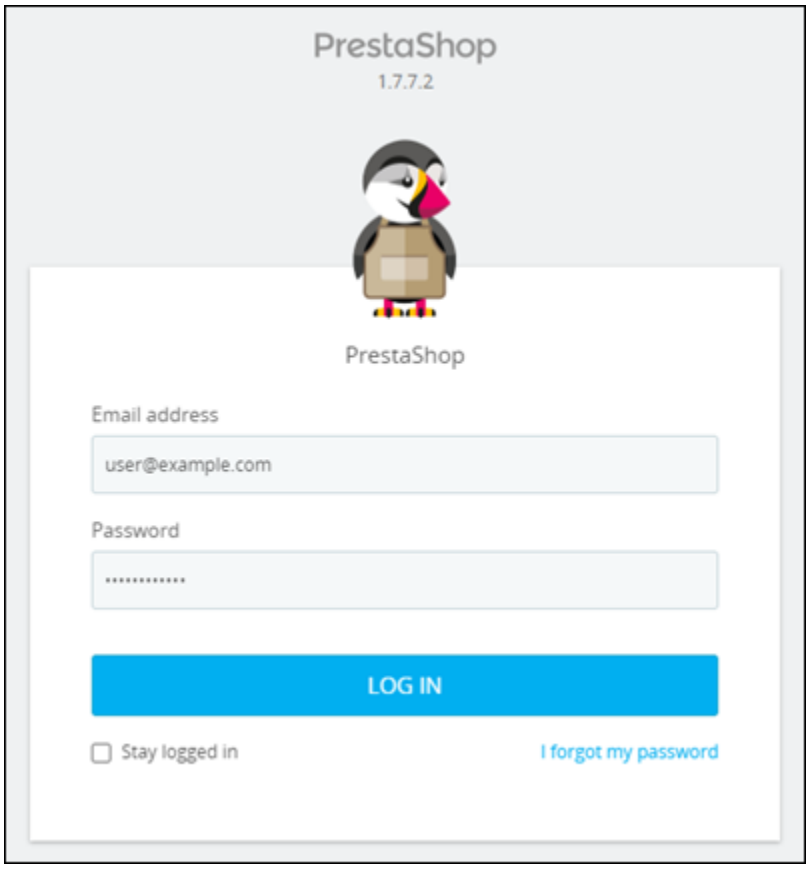
8. 以下のアドレスまで移動して、PrestaShop ウェブサイトの管理ダッシュボードのサインインページにアクセスします。*<DomainName>* は、インスタンスにトラフィックを送信している登録済みドメイン名に必ず置き換えてください。

```
http://<DomainName>/administration
```

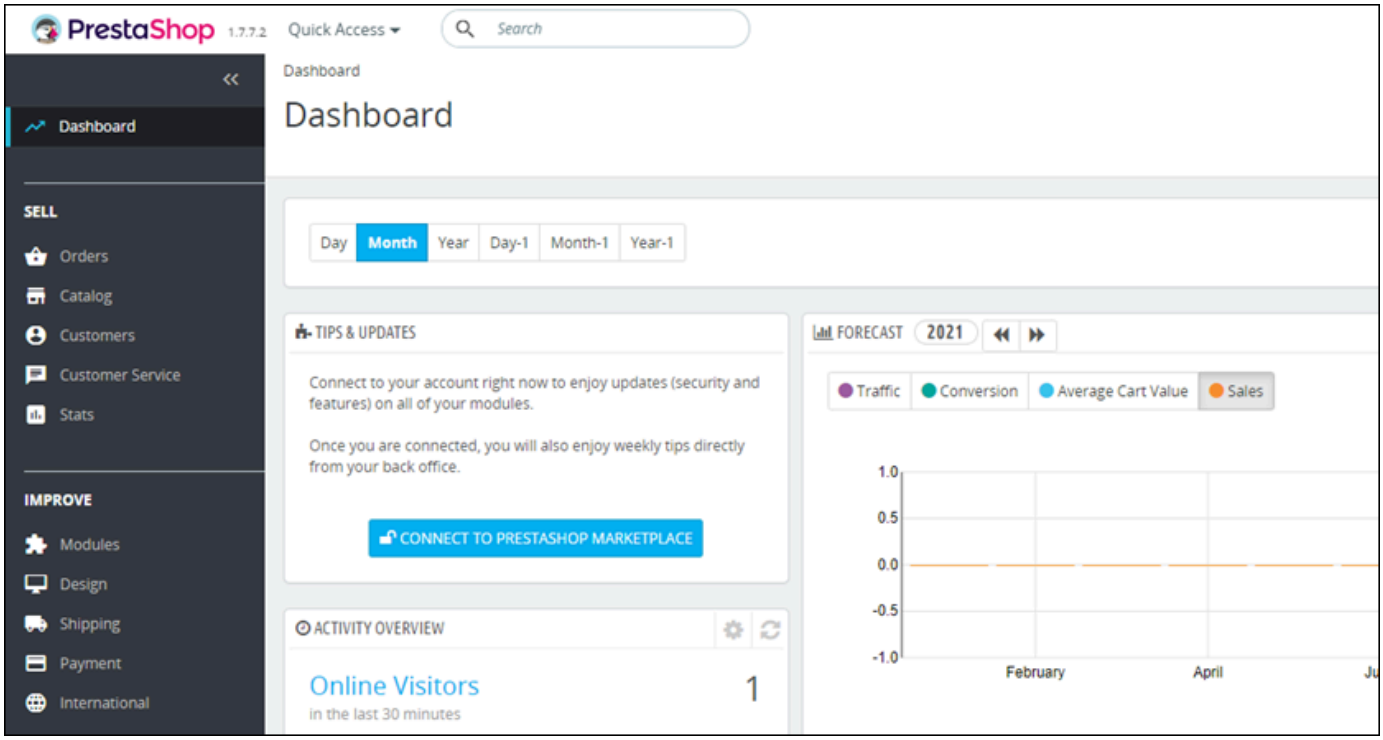
例:

```
http://www.example.com/administration
```

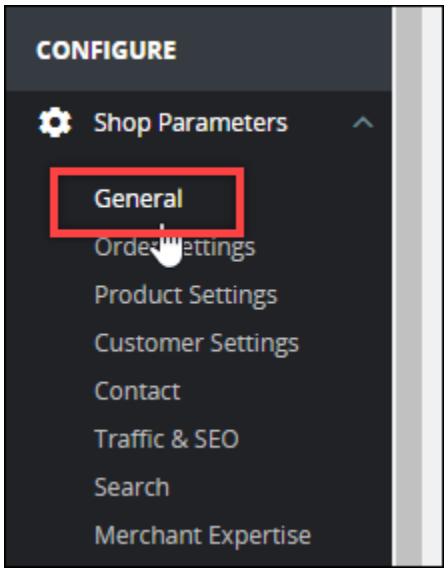
9. デフォルトのユーザー名 (user@example.com) と、このガイドの前のセクションで取得したデフォルトのアプリケーションパスワードを入力して [ログイン] を選択します。



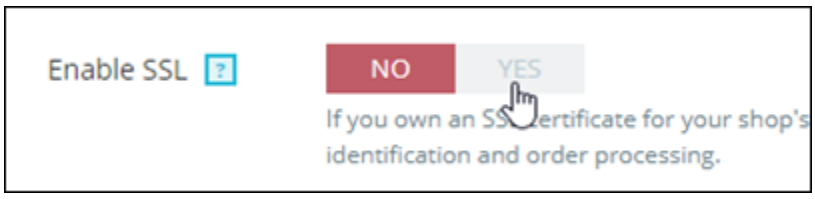
PrestaShop の管理ダッシュボードが表示されます。



10. ナビゲーションペインの [ショップパラメータ] を選択して、次に [General] (一般) を選択します。



11. [SSL を有効にする] の横にある [はい] を選択します。



12. ページの下部までスクロールし、[保存] を選択します。

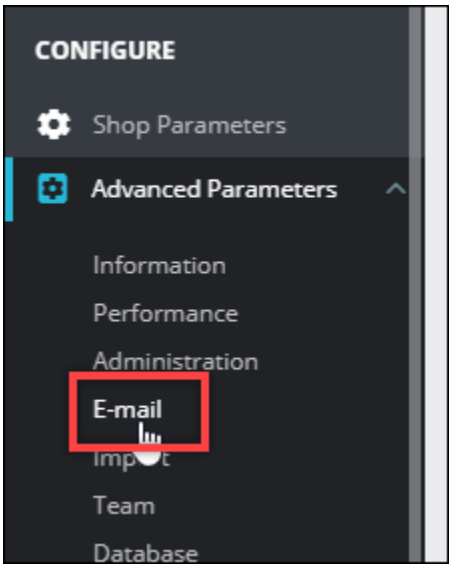
13. [General] ページが再読み込みしたら、[すべてのページでSSLを有効にする] の横にある [はい] を選択します。

14. ページの最下部までスクロールし、[Save] (保存) を選択します。

    これで、PrestaShop ウェブサイトに HTTPS が設定されました。訪問者が HTTP バージョン (例: `http://www.example.com`) の PrestaShop ウェブサイトにブラウジングした際、訪問者は自動的に HTTPS バージョン (例: `https://www.example.com`) にリダイレクトされます。

ステップ 6: メール通知用の SMTP を設定する

PrestaShop ウェブサイトの SMTP 設定を構成して、メール通知を有効にします。これを実行するには、PrestaShop ウェブサイトの管理ダッシュボードにサインインします。ナビゲーションペインの [アドバンストパラメータ] を選択して、[E メール] を選択します。E メールの連絡先もこれに応じて調整する必要があります。ナビゲーションペインの [Shop Parameters] (ショップパラメータ) をクリックしてから、[Contact] (連絡先) を選択します。



詳細については、「PrestaShop ドキュメント」の「User Guide PrestaShop」および Bitnami ドキュメントの「Configure SMTP for outbound emails」を参照してください。

> **Important**
>
> ポート 25、465、または 587 を使用するように SMTP を設定する場合、Lightsail コンソールでインスタンスファイアウォールのポートを開ける必要があります。詳細については、「Amazon Lightsail でインスタンスファイアウォールルールの追加および編集」を参照してください。
>
> Gmail アカウントを設定して PrestaShop ウェブサイトでメールを送信できるようにする場合は、Gmail のログインに使用するスタンダードパスワードではなく、アプリケーションのパスワードを使用する必要があります。詳細については、「アプリケーションのパスワードでサインイン」を参照してください。

ステップ 7: Bitnami と PrestaShop のドキュメントを読む

Bitnami のドキュメントを読んで、PrestaShop インスタンスとウェブサイト上でプラグインのインストールやテーマのカスタマイズなどの管理タスクを実行する方法を確認します。詳細については、Bitnami ドキュメントの「AWS クラウド用の Bitnami PrestaShop スタック」を参照してください。

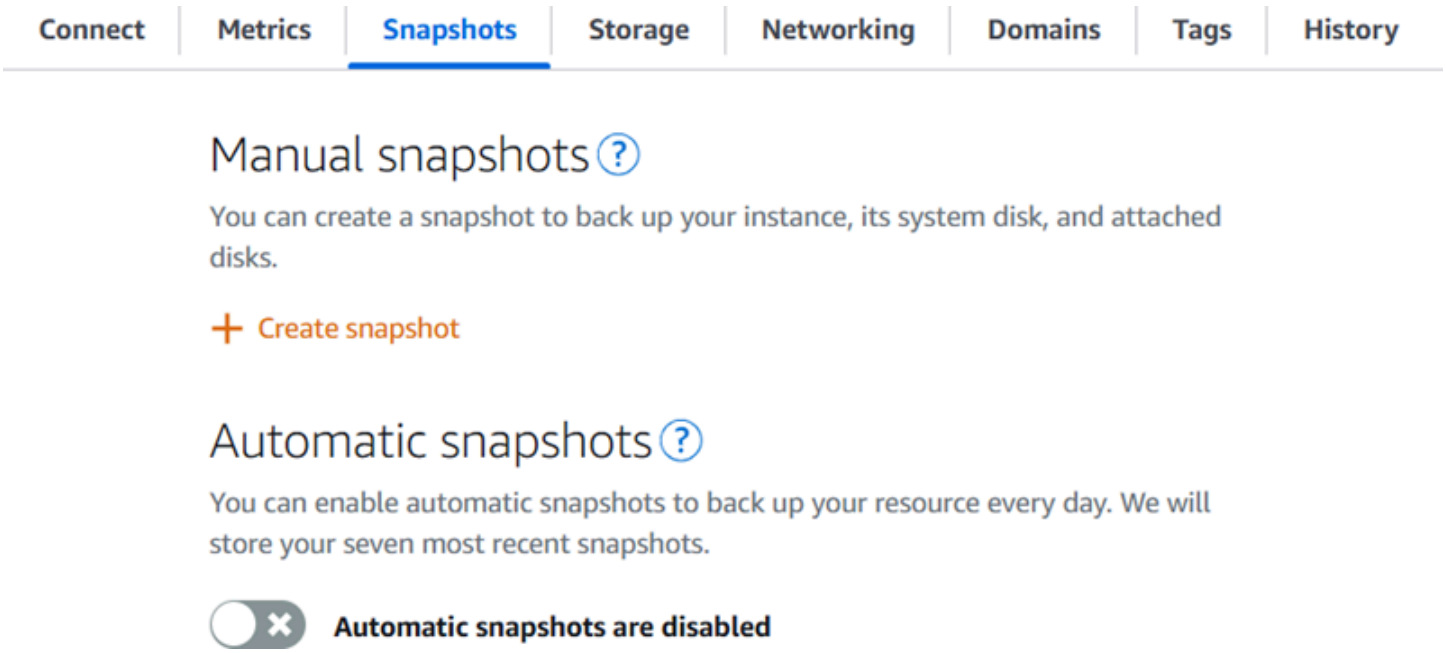
PrestaShop のドキュメントを読んで、PrestaShop のウェブサイトの管理方法も確認してください。詳細については、「PrestaShop ドキュメント」の「User Guide PrestaShop」を参照してください。

ステップ 8: PrestaShop インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。



Lightsail の Redmine インスタンスを設定して保護する

> **知っていましたか?**
>
> Lightsail は毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを最新のスナップショットに自動的に置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

Amazon Lightsail で起動した Redmine インスタンスの使用を開始するステップについて説明します。

目次

- ステップ 1: Bitnami のドキュメントを確認する
- ステップ 2: Redmine の管理ダッシュボードにアクセスするため、デフォルトのアプリケーションパスワードを取得する
- ステップ 3: インスタンスに静的 IP アドレスをアタッチする
- ステップ 4: Redmine ウェブサイトの管理ダッシュボードにサインインする
- ステップ 5: 登録済みドメイン名へのトラフィックを Redmine ウェブサイトに送信する
- ステップ 6: Redmine ウェブサイトの HTTPS を設定する

- ステップ 7: Redmine のドキュメントを読み、引き続きウェブサイトの設定を続行する
- ステップ 8: インスタンスのスナップショットを作成する

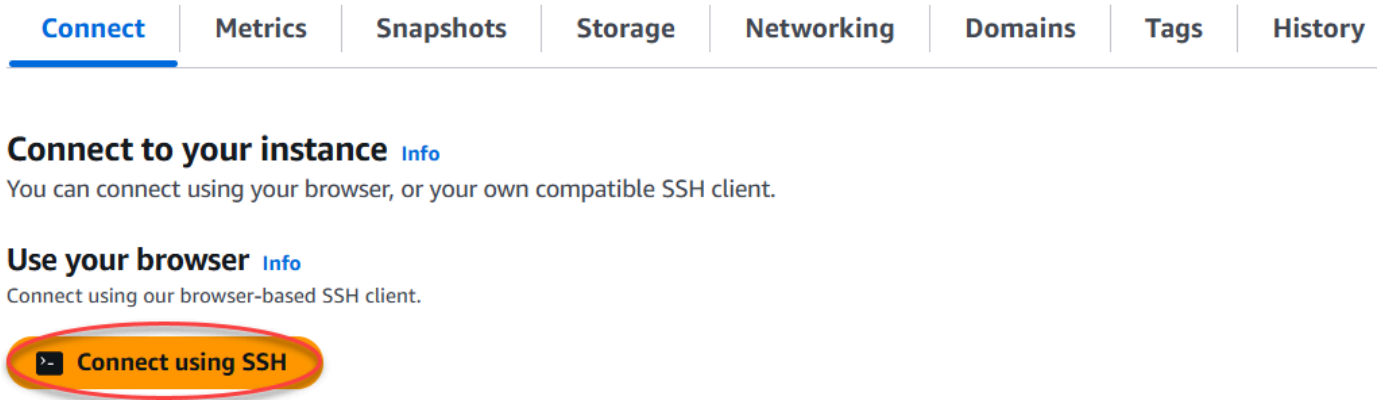
ステップ 1: Bitnami のドキュメントを確認する

Bitnami のドキュメントを読み、Redmine アプリケーションの設定方法については確認します。詳細については、「AWS クラウド用に Bitnami がパッケージ化した Redmine」を参照してください。

ステップ 2: Redmine の管理ダッシュボードにアクセスするため、デフォルトのアプリケーションパスワードを取得する

次の手順を完了して、Redmine ウェブサイトの管理ダッシュボードにアクセスする際に必要となるデフォルトのアプリケーションパスワードを取得します。詳細については、「Amazon LightsailのBitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

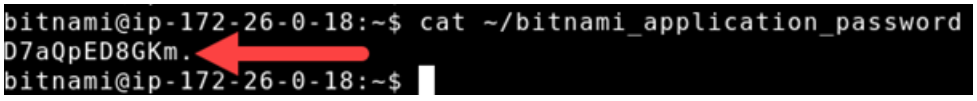
1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。



2. 接続後に、次のコマンドを入力してアプリケーションのパスワードを取得します。

```
cat $HOME/bitnami_application_password
```

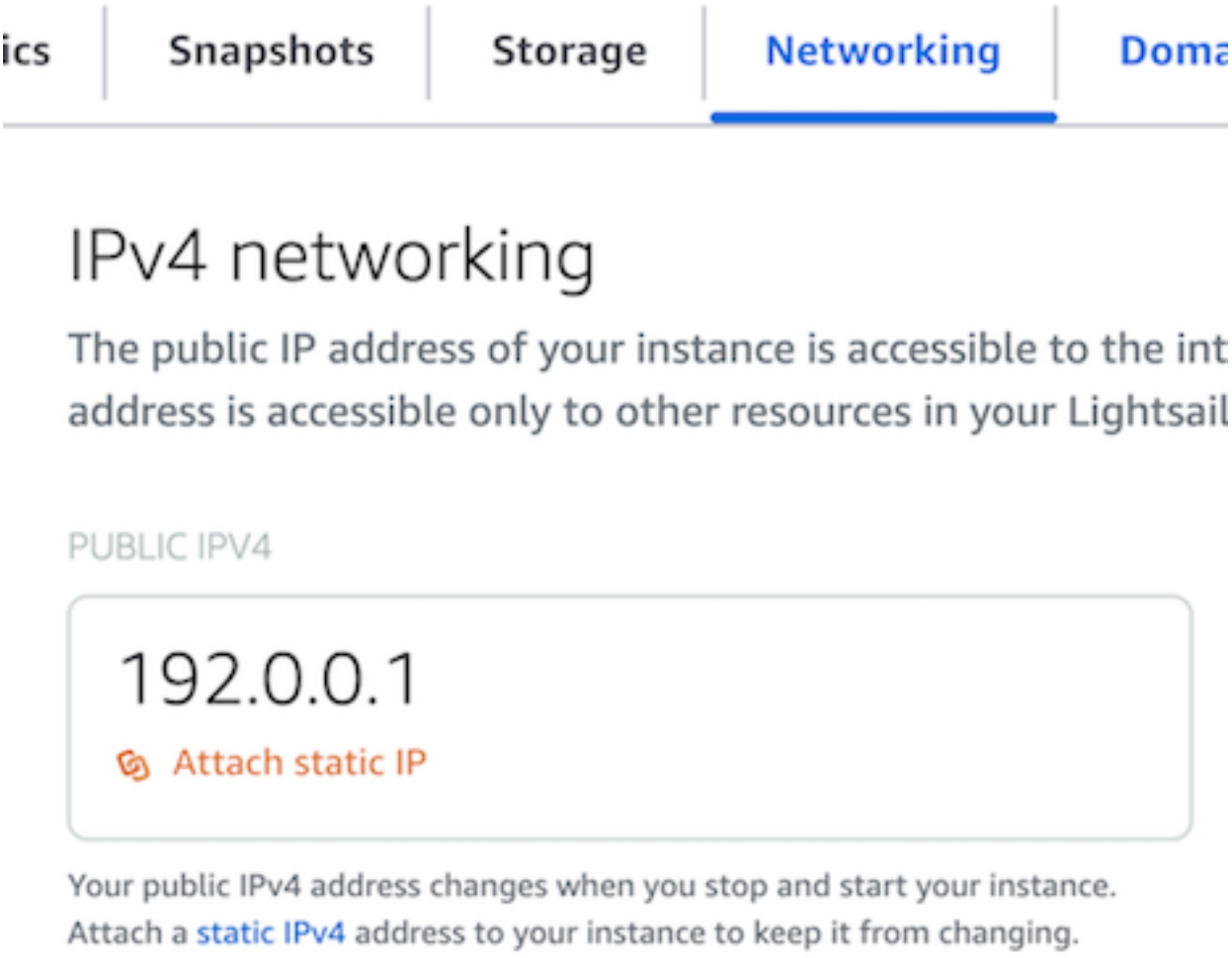
アプリケーションのデフォルトパスワードを含んだ、次の例のようなレスポンスが表示されます。



ステップ 3: インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



ステップ 4: Redmine ウェブサイトの管理ダッシュボードにサインインする

デフォルトのユーザーパスワードを取得したら、以下の手順に従ってRedmine ウェブサイトのホームページに移動し、管理ダッシュボードにサインインします。サインイン後に、ウェブサイトをカスタマイズしたり管理上の変更を行うことができます。Joomla! で実行できる事項の詳細について

は、「ステップ 7: Redmine のドキュメントを読み、引き続きウェブサイトの設定を続行する」のセクションを参照してください。

1. インスタンス管理ページの [Connect] (接続) タブにあるパブリック IP アドレスを書き留めます。パブリック IP アドレスは、インスタンス管理ページのヘッダーセクションにも表示されます。

2. インスタンスのパブリック IP アドレスを参照します (例: `http://203.0.113.0` に移動します)。

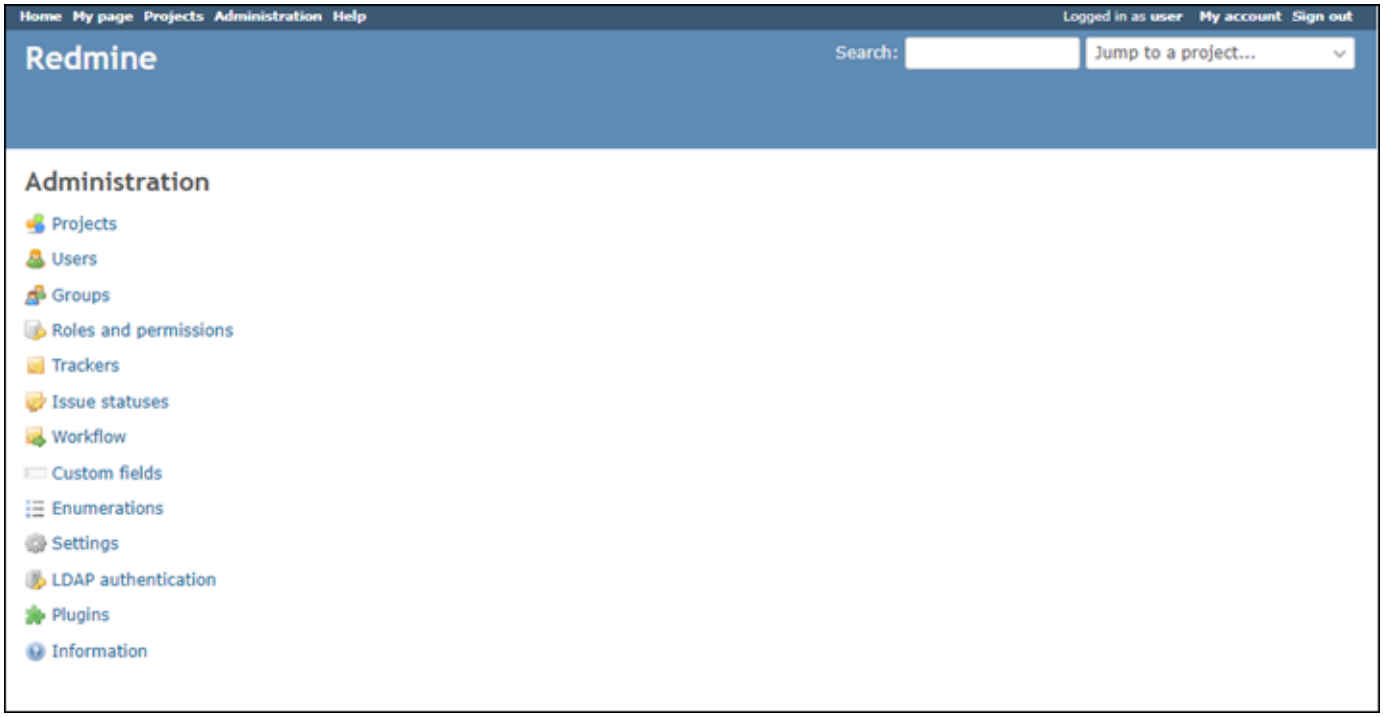
   Redmine ウェブサイトのホームページが表示されます。

3. Redmine ウェブサイトのホームページで、右下にある [Manage](管理) を選択します。

   [Manage] (管理) バナーが表示されない場合は、`http://<PublicIP>/admin` を参照することでサインインページにアクセスすることができます。*`<PublicIP>`* を、インスタンスのパブリック IP アドレスに置き換えます。

4. デフォルトのユーザー名 (`user`) と、先ほど取得したデフォルトのパスワードを使用してサインインします。

   Redmine の管理ダッシュボードが表示されます。



ステップ 5: 登録済みドメイン名へのトラフィックを Redmine ウェブサイトに送信する

`example.com` などの登録済みドメイン名のトラフィックを Redmine ウェブサイトに送信するには、ドメインの DNS にレコードを追加します。DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページの [Domains & DNS] (ドメインと DNS) タブで、[Create DNS zone] (DNS ゾーンの作成) を選択し、ページに記載される手順に従います。詳細については、「Lightsail で DNS ゾーンを作成し、ドメインの DNS レコードを管理する」を参照してください。

インスタンスに設定したドメイン名を参照すると、Redmine ウェブサイトのホームページへと移動します。次に、SSL/TLS 証明書を生成して設定し、Redmine ウェブサイトの HTTPS 接続を有効にします。詳細については、本ガイドの次の「ステップ 6: Redmine ウェブサイトの HTTPS を設定する」のセクションを参照してください。

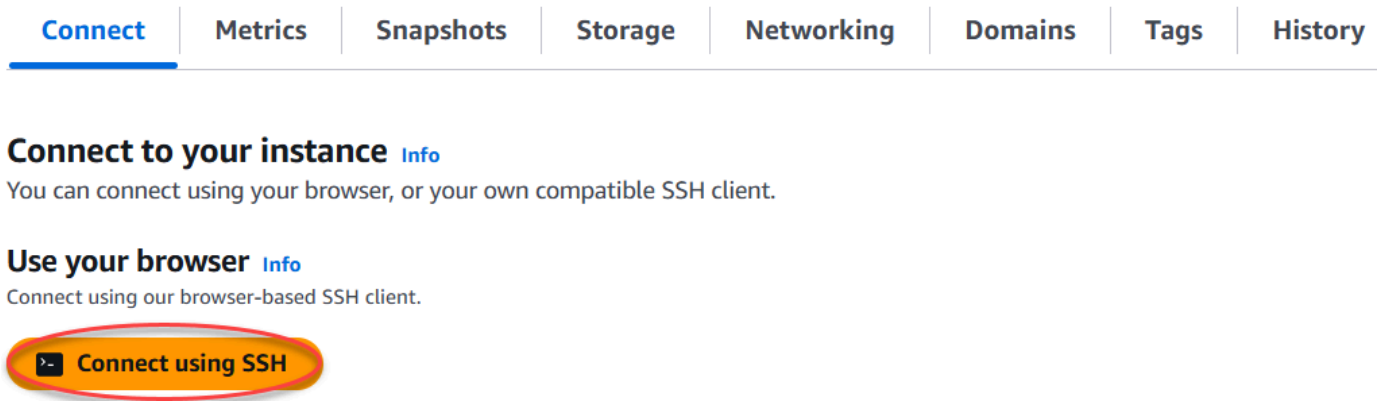
ステップ 6: Redmine ウェブサイトの HTTPS を設定する

Redmine ウェブサイトで HTTPS を設定するには、以下の手順を実行します。次の手順では、Bitnami HTTPS 設定ツール (`bncert-tool`) の使い方を説明しています。これは、Let's Encrypt

SSL/TLS 証明書を要求するコマンドラインツールです。詳細については、「Bitnami ドキュメント」の「Bitnami HTTPS 設定ツールの詳細」を参照してください。

> **Important**
>
> この手順を開始する前に、Redmine インスタンスにトラフィックがルーティングされるようにドメインが設定済みであることを確認してください。設定されていない場合、SSL/TLS 証明書の検証プロセスが失敗します。

1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。



2. 接続されたら、以下のコマンドを入力し、インスタンスに bncert ツールがインストールされていることを確認します。

```
sudo /opt/bitnami/bncert-tool
```

   以下のレスポンスのいずれかが表示されます。

   - レスポンスにコマンドが見つからないと表示された場合、bncert ツールがインスタンスにインストールされていないことを示しています。この手順の次のステップに進み、bncert ツールをインスタンスにインストールします。
   - レスポンスに「Bitnami HTTPS 設定ツールへようこそ」と表示された場合は、インスタンスに bncert ツールがインストールされています。この手順のステップ 8 に進んでください。
   - bncert ツールがインスタンスにインストールされてからしばらく経っている場合、ツールのアップデートバージョンが利用可能であることを示すメッセージが表示されることがあります。ダウンロードすることを選択し、`sudo /opt/bitnami/bncert-tool` コマンドを入力して bncert ツールを再度実行してください。この手順のステップ 8 に進んでください。

3. 以下のコマンドを入力して、bncert の実行ファイルをインスタンスにダウンロードします。

```
wget -O bncert-linux-x64.run https://downloads.bitnami.com/files/bncert/latest/
bncert-linux-x64.run
```

4. 以下のコマンドを入力して、インスタンスに bncert ツールの実行ファイル用のディレクトリを作成します。

```
sudo mkdir /opt/bitnami/bncert
```

5. 以下のコマンドを入力して、プログラムとして実行できるファイルを bncert に実行させます。

```
sudo chmod +x /opt/bitnami/bncert/bncert-linux-x64.run
```

6. 次のコマンドを入力して、sudo /opt/bitnami/bncert-tool コマンドを入力すると bncert ツールを実行するシンボリックリンクを作成します。

```
sudo ln -s /opt/bitnami/bncert/bncert-linux-x64.run /opt/bitnami/bncert-tool
```

これでインスタンスに bncert ツールをインストールする手順は完了です。

7. 次のコマンドを入力して、bncert ツールを実行しましょう。

```
sudo /opt/bitnami/bncert-tool
```

8. 次の例に示すように、プライマリドメイン名と代替ドメイン名の間はスペースで区切って入力します。

ドメインがインスタンスのパブリック IP アドレスにトラフィックをルーティングするように設定されていない場合、bncert ツールは、続行する前にその設定を行うように要求します。ドメインは、bncert ツールを使用して HTTPS を有効にしているインスタンスでのパブリック IP アドレスにトラフィックをルーティングする必要があります。これはドメインを所有していることを確認し、証明書の検証として機能します。

```
------------------------------------------------------------------------------
Welcome to the Bitnami HTTPS Configuration tool.

------------------------------------------------------------------------------
Domains

Please provide a valid space-separated list of domains for which you wish to
configure your web server.

Domain list []: example.com www.example.com
```

9. bncert ツールは、ウェブサイトのリダイレクトの設定方法を尋ねます。使用できるオプションは次のとおりです。

   - HTTP から HTTPS へのリダイレクトを有効にする - HTTP バージョンのウェブサイトを閲覧するユーザー (例: http:/example.com) を自動的に HTTPS バージョン (例: https://example.com) にリダイレクトするかどうかを決定します。すべての訪問者が暗号化された接続を使用するように強制されるため、このオプションを有効にすることをお勧めします。Y を入力して Enter を押すと、有効になります。
   - www なしから www ありへのリダイレクトの有効化 - ドメインの頂点 (例: https://example.com) まで閲覧するユーザー を自動的にドメインの www サブドメイン (例: https://www.example.com) にリダイレクトするかを指定します。このオプションを有効にすることをお勧めします。ただし、ドメインの頂点を Google のウェブマスターツールなどの検索エンジンツールで希望のウェブサイトアドレスとして指定した場合、または頂点が IP を直接指しており、www のサブドメインが CNAME レコードを介してリファレンスしている場合は、無効にして代替オプションを有効にすることをお勧めします (www ありから www なしへのリダイレクトを有効化) 。Y を入力し、Enter を押して有効にします。
   - www ありから www なしへのリダイレクトを有効にする - ドメインの www サブドメイン (例: https://www.example.com) まで閲覧するユーザーを、自動的にドメインの頂点 (例: https://example.com) にリダイレクトするかを指定します。www なしから www ありへのリダイレクトを有効にした場合は、これを無効にすることをお勧めします。N を入力し、Enter を押して無効にします。

   選択した結果は次の例のようになります。

```
Enable/disable redirections

Please select the redirections you wish to enable or disable on your Bitnami
installation.



Enable HTTP to HTTPS redirection [Y/n]: Y



Enable non-www to www redirection [Y/n]: Y



Enable www to non-www redirection [y/N]: N
```

10. これから実行される変更が一覧表示されます。Y と入力し、Enter を押して確認し、続行します。

```
Changes to perform

The following changes will be performed to your Bitnami installation:

1. Stop web server
2. Configure web server to use a free Let's Encrypt certificate for the domains:
example.com www.example.com
3. Configure a cron job to automatically renew the certificate each month
4. Configure web server name to: example.com
5. Enable HTTP to HTTPS redirection (example: redirect http://example.com to
https://example.com)
6. Enable non-www to www redirection (example: redirect example.com to
 www.example.com)
7. Start web server once all changes have been performed



Do you agree to these changes? [Y/n]: Y
```

11. Let's Encrypt 証明書に関連付けるメールアドレスを入力し、Enter を押します。

```
Create a free HTTPS certificate with Let's Encrypt

Please provide a valid e-mail address for which to associate your Let's Encrypt
certificate.

Domain list: example.com www.example.com

Server name: example.com

E-mail address []:
```

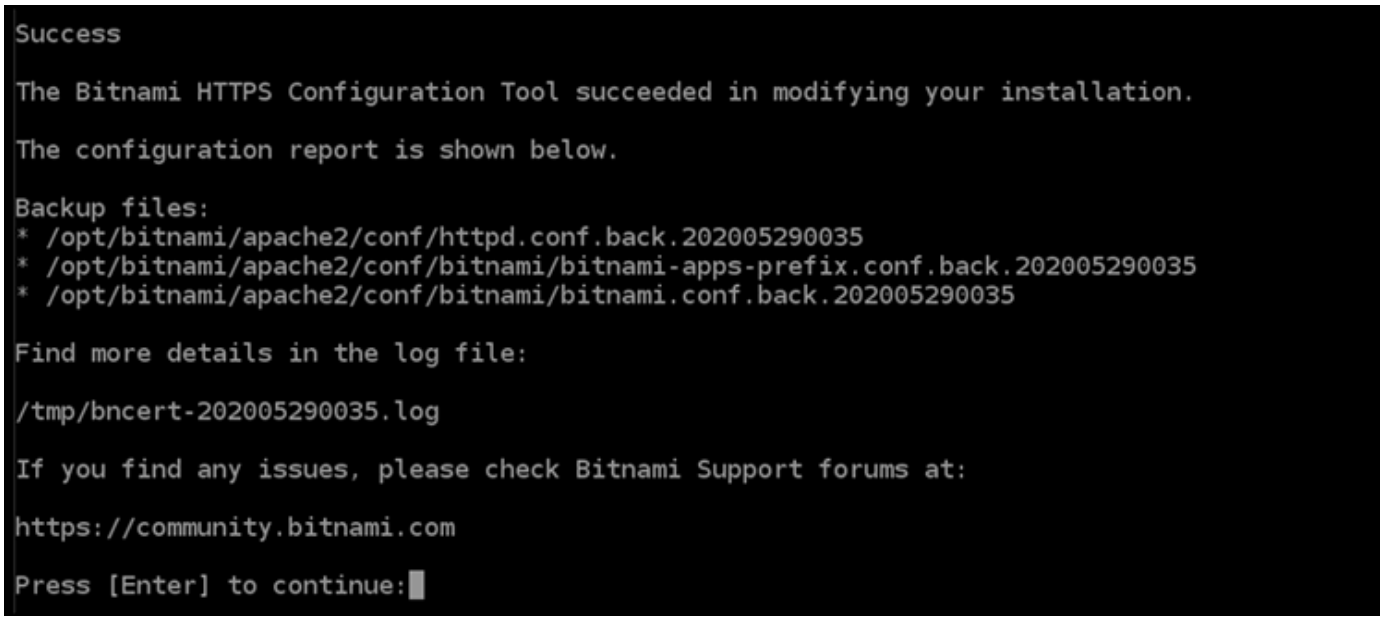
12. Let's Encrypt サブスクライバー合意書を確認します。Y と入力し、Enter を押して契約に同意し、続行します。

```
The Let's Encrypt Subscriber Agreement can be found at:

https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf

Do you agree to the Let's Encrypt Subscriber Agreement? [Y/n]:
```

これらのアクションは、証明書のリクエストや指定したリダイレクトの設定など、インスタンスで HTTPS を有効にするために実行されます。

```
Performing changes to your installation

The Bitnami HTTPS Configuration Tool will perform any necessary actions to your
Bitnami installation. This may take some time, please be patient.

|
```

次の例のようなメッセージが表示された場合は、証明書は正常に発行され、検証され、インスタンスでリダイレクトが正常に設定されています。



bncert ツールは、有効期限が切れる前、 80 日ごとに証明書の自動更新を実行します。インスタンスで追加のドメインやサブドメインを使用し、それらのドメインで HTTPS を有効にする場合は、上記のステップを繰り返します。

これで、Redmine インスタンスでの HTTPS の有効化が完了しました。次回、設定したドメインを使用して Redmine ウェブサイトを閲覧する際には、HTTPS 接続にリダイレクトされるはずです。

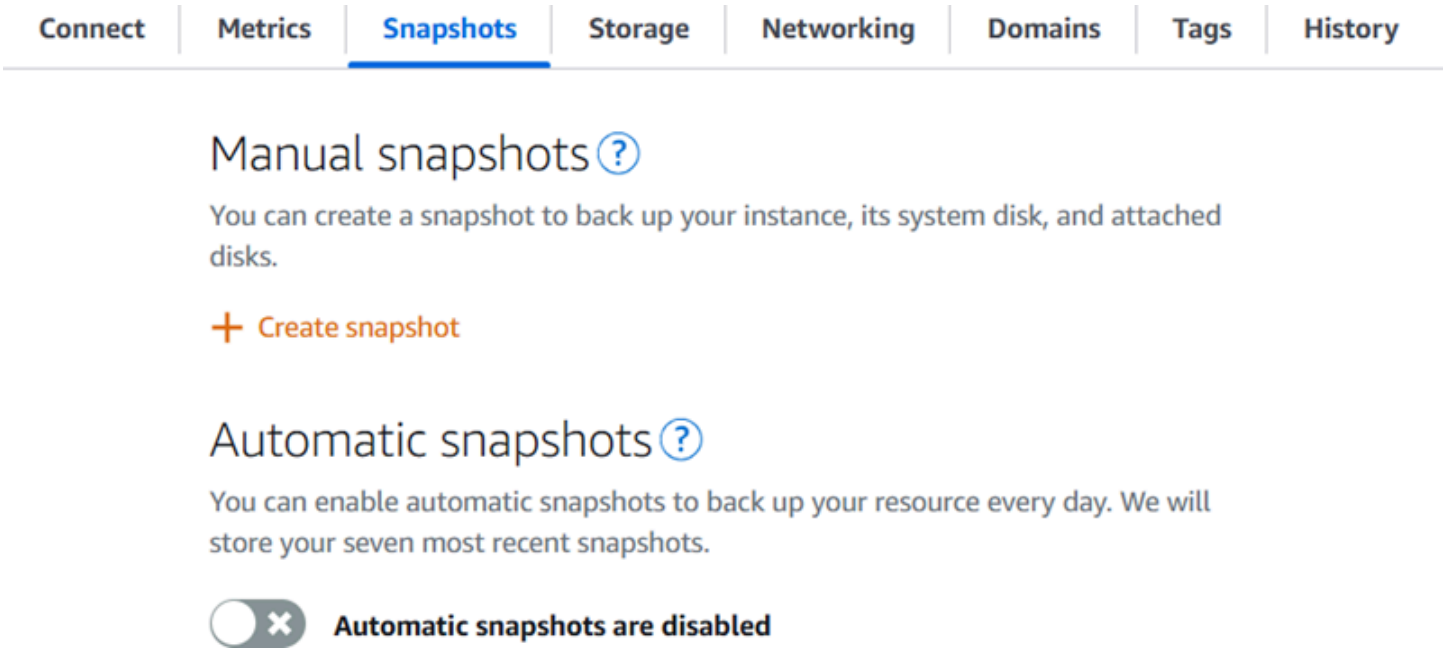
ステップ 7: Redmine のドキュメントを読み、引き続きウェブサイトの設定を続行する

Redmine のドキュメントを読み、ウェブサイトを管理およびカスタマイズする方法を確認します。詳細については、「Redmine ガイド」を参照してください。

ステップ 8: インスタンスのスナップショットを作成する

Redmine ウェブサイトを希望どおりに設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。詳細については、「スナップショット」を参照してください。

インスタンス管理ページの [スナップショット] タブで [スナップショットを作成する] を選択するか、[自動スナップショットを有効にする] を選択します。



詳細については、「Amazon Lightsail で Linux または Unix インスタンスのスナップショットを作成」および「Amazon Lightsail のインスタンスまたはディスクの自動スナップショットの有効化と無効化」を参照してください。

で Rails に Ruby を設定する Lightsail

> 知っていましたか?
>
> Lightsail は、毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを自動的に最新のスナップショットに置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

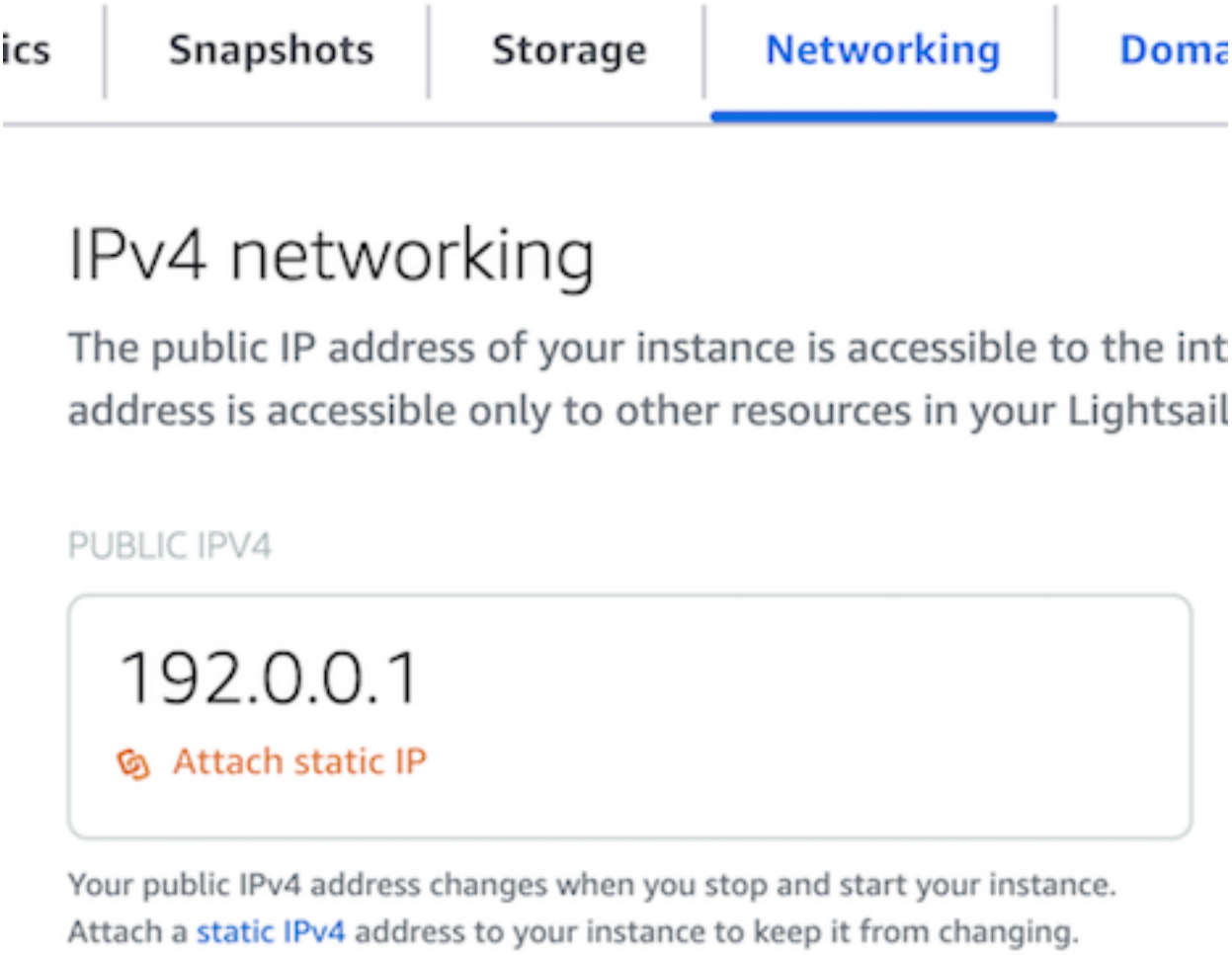
Ruby on Rails インスタンスが で起動および実行された後に開始するには、いくつかのステップを実行する必要がありますAmazon Lightsail。

ステップ 1: Ruby on Rails インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンス

を停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



ステップ 2: Ruby on Rails インスタンスのウェルカムページにアクセスする

インスタンスの静的 IP アドレスに移動して、インスタンスにインストールされているアプリケーションにアクセスします。

1. インスタンス管理ページの Connect タブで、静的 IP を書き留めます。
2. 静的 IP アドレスを参照します。例: `http://192.0.0.1:3000`。

詳細については、「Ruby on Rails ガイド」を参照してください。

ステップ 3: アプリケーションをデプロイする

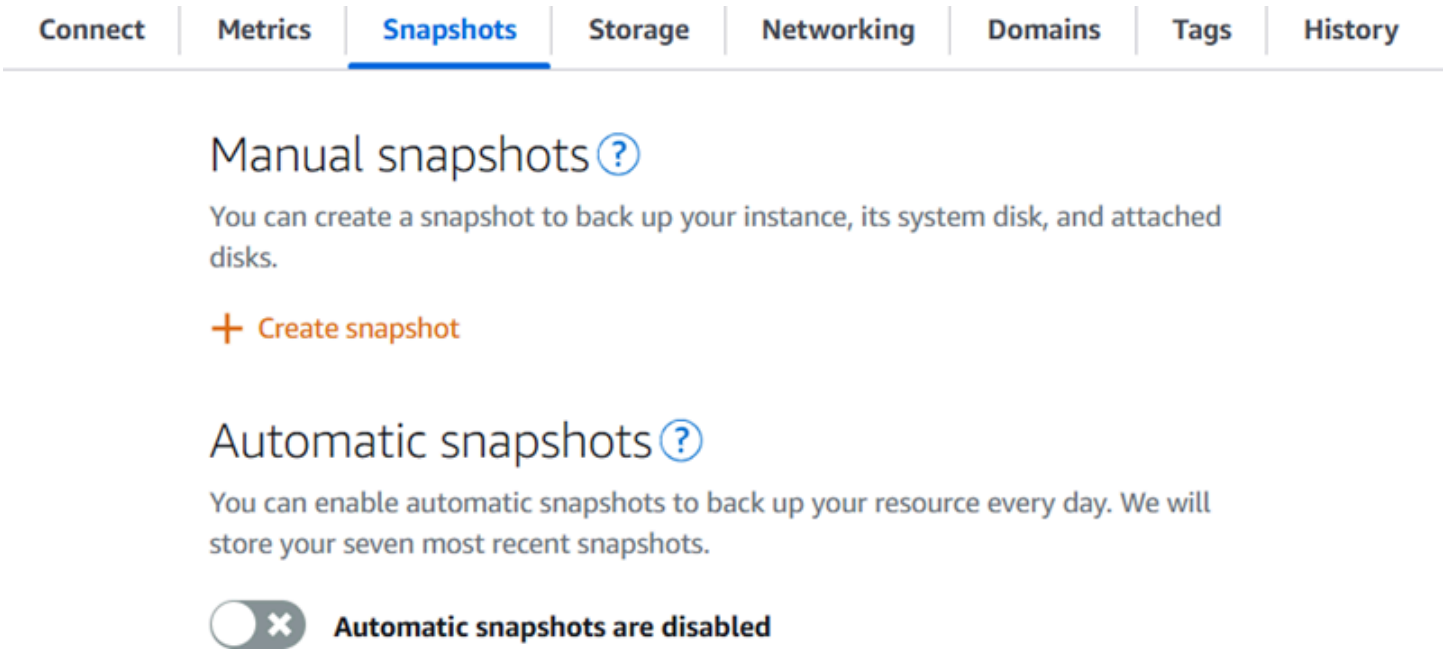
1. SFTP を使用して Lightsail Linux インスタンスにファイルを安全に転送するの指示に従って、アプリケーションを にコピーします。 `/home/ec2-user/my_app`
2. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。
3. `sudo systemctl restart rails-server` を実行する
4. インスタンスの静的 IP アドレスに移動する

ステップ 4: Ruby on Rails インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。

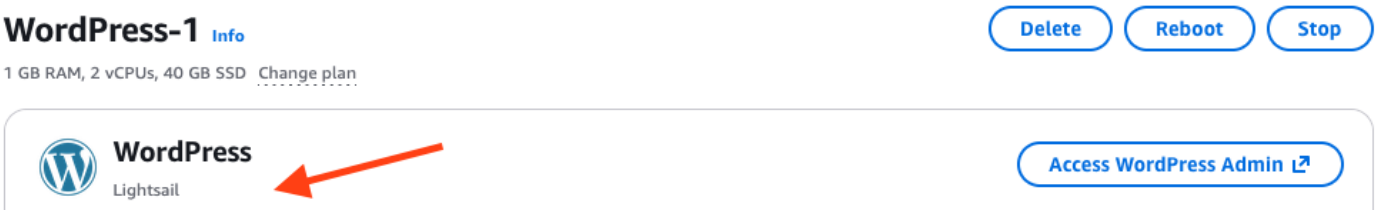


での WordPress のデプロイと管理 Lightsail

> **知っていましたか?**
>
> Lightsail は、毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを自動的に最新のスナップショットに置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

このクイックスタートガイドでは、Amazon Lightsail で WordPress インスタンスの起動および設定方法について説明します。

ここでは、WordPress インスタンスが で起動および実行された後に開始するための手順をいくつか示しますAmazon Lightsail。開始する前に、インスタンス管理ページでブループリントベンダーを特定します。



WordPress インスタンスに適したガイドを選択します。

Bitnami

ステップ 1: WordPress インスタンスを設定する

WordPress インスタンスは、以下を設定するガイド付きステップバイステップワークフローを使用して設定できます:

- 登録済みドメイン名 – WordPress サイトには、覚えやすいドメイン名が必要です。ユーザーは、WordPress サイトにアクセスするためにこのドメイン名を指定します。詳細については、「ドメインと DNS」を参照してください。
- DNS 管理 – ドメインの DNS レコードを管理する方法を決定する必要があります。DNS レコードは、DNS サーバーに対して、ドメインやサブドメインに対応する IP アドレスやホスト名を示します。DNS ゾーンには、ドメインの DNS レコードが含まれます。詳細については、「the section called “Lightsail での DNS”」を参照してください。

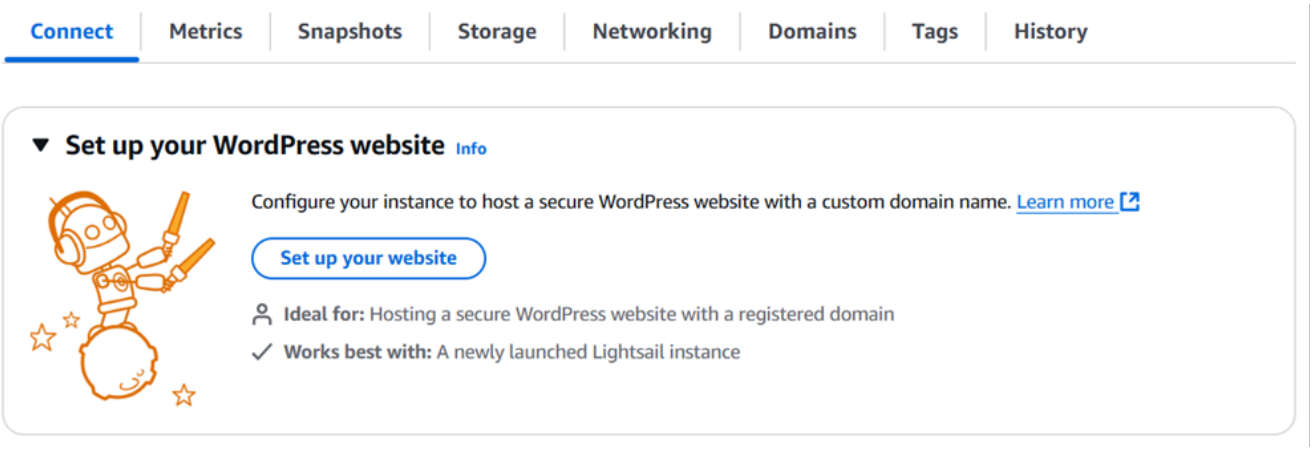
- 静的 IP アドレス – WordPress インスタンスのデフォルトのパブリック IP アドレスは、インスタンスを停止して開始すると変わります。インスタンスに静的 IP アドレスをアタッチすると、インスタンスを停止して開始してもそのアドレスが変わりません。詳細については、「the section called “IP アドレス”」を参照してください。
- SSL/TLS 証明書 – 検証済みの証明書を作成してインスタンスにインストールしたら、WordPress ウェブサイトの HTTPS を有効にして、登録済みドメインを介してインスタンスにルーティングされるトラフィックを HTTPS を使用して暗号化できます。詳細については、「the section called “HTTPS を有効にする”」を参照してください。

> **Tip**
>
> 開始する前に、以下のヒントを確認します。トラブルシューティングの詳細については、「Troubleshooting WordPress setup」を参照してください。
>
> - セットアップは、2023 年 1 月 1 日以降に作成された WordPress バージョン 6 以降の Lightsail インスタンスをサポートします。
> - 設定中に実行される Certbot 依存関係ファイル、HTTPS 書き換えスクリプト、および証明書更新スクリプトは、インスタンスの `/opt/bitnami/lightsail/scripts/` ディレクトリに保存されます。
> - インスタンスを [実行中] 状態にする必要があります。インスタンスが開始したばかりの場合、SSH 接続の準備が整うまで数分かかります。
> - インスタンスファイアウォールのポート 22、80、443 では、設定の実行中に任意の IP アドレスからの TCP 接続を許可する必要があります。詳細については、「インスタンスのファイアウォール」を参照してください。
> - 頂点ドメイン (`example.com`) とその `www` サブドメイン (`www.example.com`) からのトラフィックをポイントする DNS レコードを追加または更新する場合、それらはインターネット全体に伝播する必要があります。「nslookup」や「MxToolbox」の「DNS Lookup」などのツールを使用して、DNS の変更が有効になったことを確認できます。
> - 2023 年 1 月 1 日より前に作成された Wordpress インスタンスには、ウェブサイトのセットアップが失敗する、廃止された Certbot Personal Package Archive (PPA) リポジトリが含まれている場合があります。セットアップ中にこのリポジトリが存在する場合、既存のパスから削除され、インスタンス上の次の場所にバックアップされます: `~/opt/bitnami/lightsail/repo.backup`。廃止された PPA の詳細については、正規ウェブサイトの「Certbot PPA」を参照してください。
> - Let's Encrypt 証明書は 60 ~ 90 日ごとに自動的に更新されます。

- 設定が進行中の場合、インスタンスを停止したり変更を加えたりしないでください。インスタンスの設定には最大 15 分かかる場合があります。インスタンス接続タブで、各ステップの進行状況を表示できます。

ウェブサイト設定ウィザードを使用してインスタンスを設定するには

1. インスタンス管理ページの [Connect] タブで、[ウェブサイトの設定] を選択します。



2. [ドメイン名を指定する] には、既存の Lightsail マネージドドメインを使用するか、新しいドメインを Lightsail に登録するか、別のドメインレジストラを使用して登録したドメインを使用します。[このドメインを使用] を選択して次のステップに進みます。
3. [接続] で、次のいずれかを実行します:
   - [Lightsail マネージドドメイン] を選択して Lightsail DNS ゾーンを使用します。[DNS ゾーンを使用する] を選択して、次の手順に進みます。
   - [サードパーティードメイン] を選択して、ドメインの DNS レコードを管理するホスティングサービスを使用します。後で使用する場合に備えて、Lightsail アカウントに一致する DNS ゾーンを作成することに注意してください。[サードパーティーの DNS を使用する] を選択して次の手順に進みます。
4. [静的 IP アドレスを作成する] には、静的 IP アドレスの名前を入力し、[静的 IP を作成する] を選択します。
5. [ドメイン割り当ての管理] では、[割り当ての追加] を選択し、ドメインタイプを選択し、[追加] を選択します。[続行] を選択して、次の手順に進みます。
6. [SSL/TLS 証明書の作成] では、ドメインおよびサブドメインを指定してメールアドレスを入力します。その後、[Lightsail がインスタンスで Let's Encrypt 証明書を構成することを許可する] を選択して、[証明書の作成] をクリックします。Lightsail リソースの構成を開始します。

設定が進行中の場合、インスタンスを停止したり変更を加えたりしないでください。インスタンスの設定には最大 15 分かかる場合があります。インスタンス接続タブで、各ステップの進行状況を表示できます。

7. ウェブサイトのセットアップが完了したら、ドメイン割り当てステップで指定した URL が WordPress サイトを開いていることを確認します。

ステップ 2: WordPress ウェブサイトのデフォルトのアプリケーションパスワードを取得する

WordPress ウェブサイトの管理ダッシュボードにサインインするには、デフォルトのアプリケーションパスワードが必要です。

WordPress 管理者のデフォルトパスワードを取得するには

1. WordPress インスタンスのインスタンス管理ページを開きます。
2. [WordPress] パネルで、[デフォルトのパスワードの取得] を選択します。これにより、ページの下部にある [アクセスのデフォルトパスワード] が展開されます。

3. [CloudShell の起動] を選択します。そうすることで、ページの下部に分割されたペインが開きます。
4. [コピー] を選択し、コンテンツを CloudShell ウィンドウに貼り付けます。CloudShell プロンプトにカーソルを置いて Ctrl+V を押すか、右クリックしてメニューを開き、[貼り付ける] を選択します。
5. CloudShell ウィンドウに表示されるパスワードを書き留めます。後で WordPress ウェブサイトの管理ダッシュボードにサインインするには必要です。

```
[cloudshell-user@ip-[redacted] ~]$ AWS_REGION=us-east-1 ~/lightsail_connect WordPress-1 cat bitnami_applic
ation_password
JKzhKzhKzhR!
```

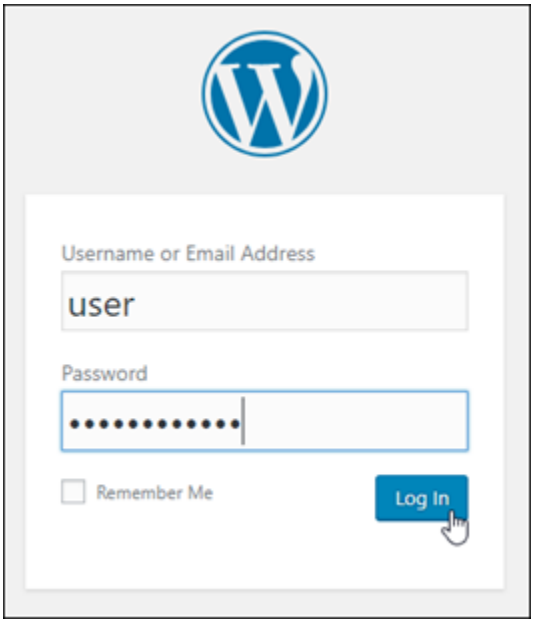
ステップ 3: WordPress ウェブサイトにサインインする

デフォルトのユーザーパスワードを取得したら、WordPress ウェブサイトのホームページに移動し、管理ダッシュボードにサインインします。サインインしたら、デフォルトのパスワードを変更できます。

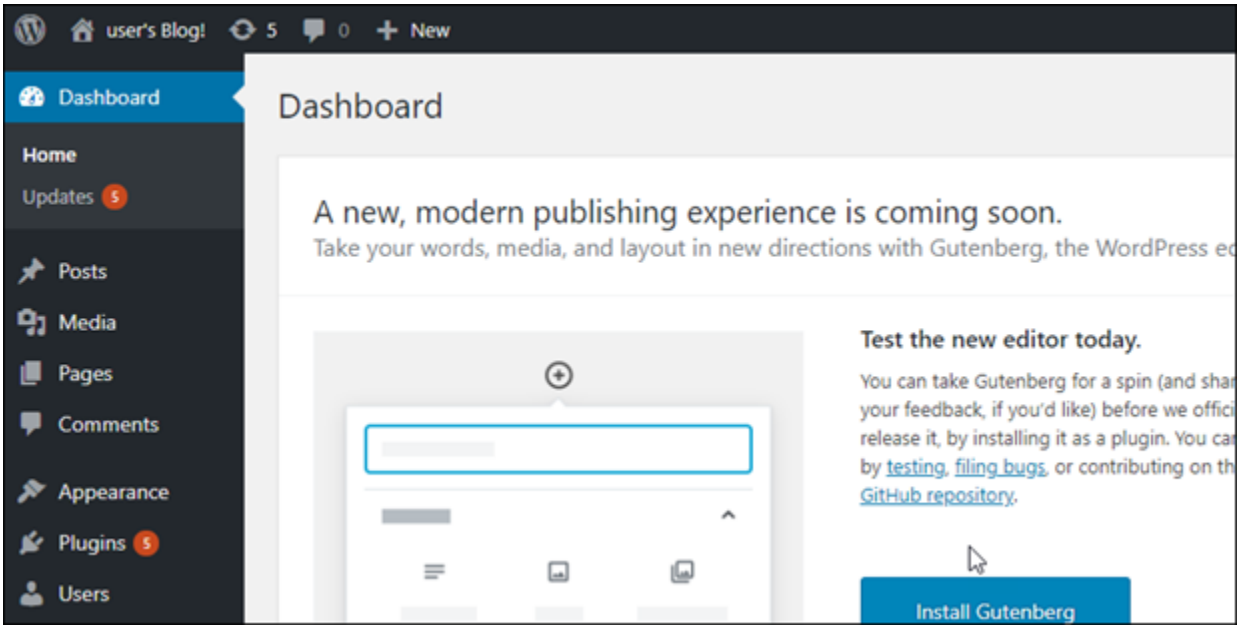
管理ダッシュボードにサインインするには

1. WordPress インスタンスのインスタンス管理ページを開きます。
2. [WordPress] パネルで、[WordPress 管理者へのアクセス] を選択します。
3. [WordPress 管理者ダッシュボードへのアクセス] パネルで、[パブリック IP アドレスを使用する] で、次のフォーマットを使用するリンクを選択します:

   http://*public-ipv4-address* /wp-admin
4. [ユーザー名またはメールアドレス] ボックスに **user** と入力します。
5. [パスワード] には、前の手順で取得したパスワードを入力します。
6. [ログイン] を選択します。



WordPress ウェブサイトの管理ダッシュボードにサインインしたら、管理アクションを実行できます。WordPress ウェブサイトの管理の詳細については、WordPress のドキュメントで「WordPress Codex」を参照してください。



ステップ 4: Bitnami ドキュメントを読む

Bitnami のドキュメントで、WordPress ウェブサイトでのプラグインのインストール、テーマのカスタマイズ、WordPress バージョンのアップグレードなどの管理タスクを実行する方法を確認します。

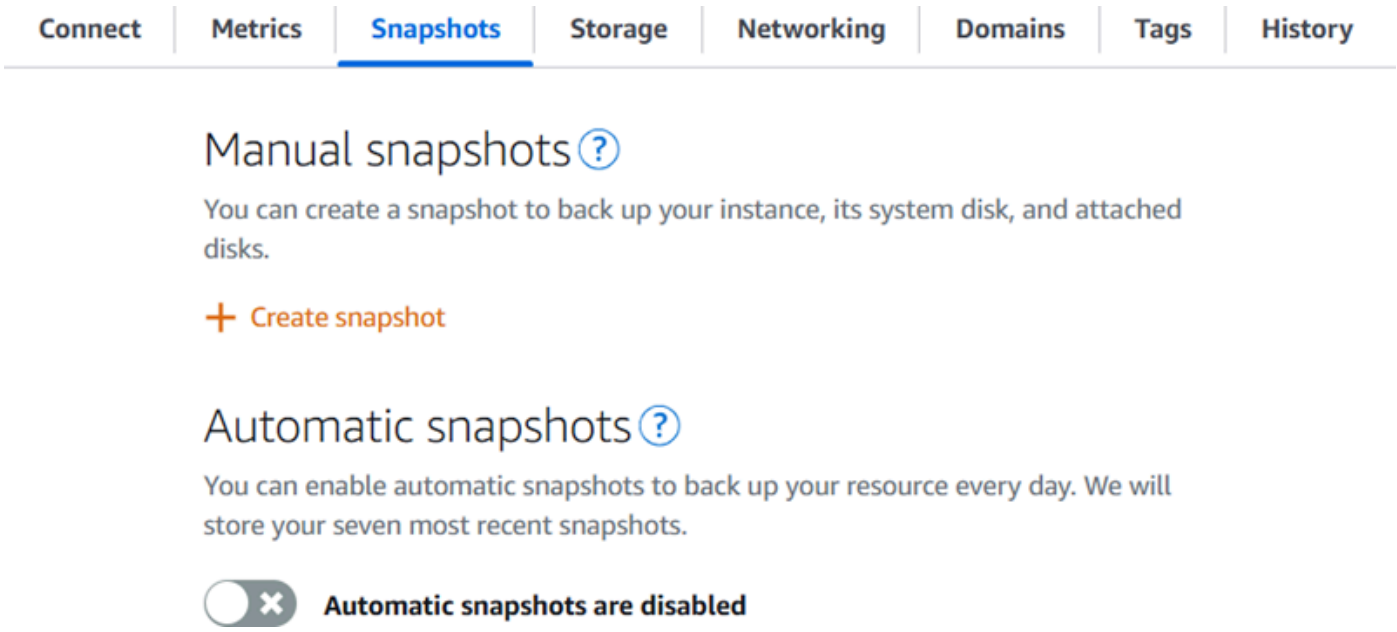
詳細については、「AWS クラウド用の Bitnami WordPress」を参照してください。

ステップ 5: インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。



Lightsail

ステップ 1: WordPress インスタンスを設定する

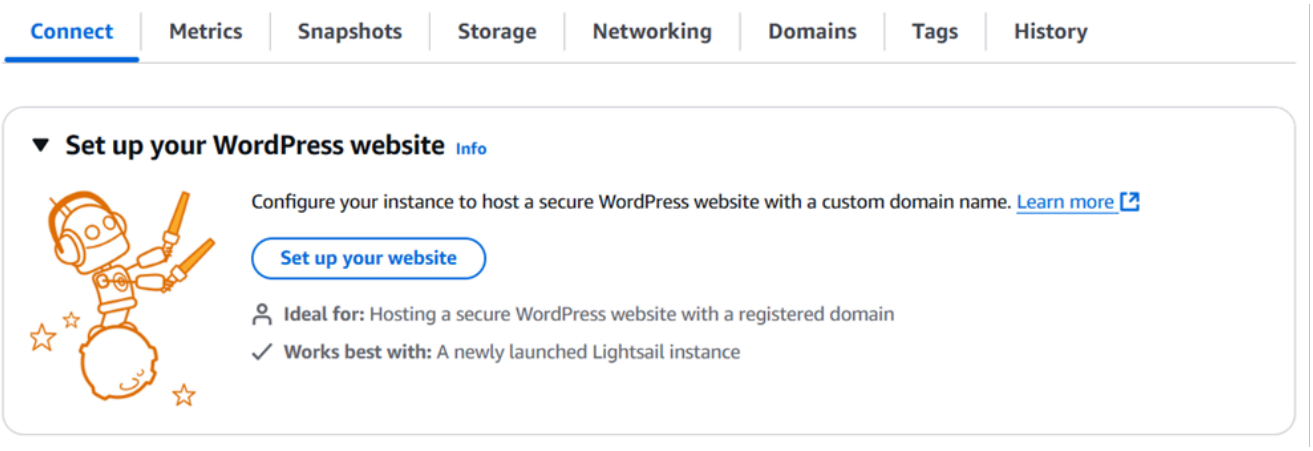
WordPress インスタンスは、以下を設定するガイド付きステップバイステップワークフローを使用して設定できます:

- 登録済みドメイン名 – WordPress サイトには、覚えやすいドメイン名が必要です。ユーザーは、WordPress サイトにアクセスするためにこのドメイン名を指定します。詳細については、「ドメインと DNS」を参照してください。
- DNS 管理 – ドメインの DNS レコードを管理する方法を決定する必要があります。DNS レコードは、DNS サーバーに対して、ドメインやサブドメインに対応する IP アドレスやホスト名を示します。DNS ゾーンには、ドメインの DNS レコードが含まれます。詳細については、「the section called “Lightsail での DNS”」を参照してください。
- 静的 IP アドレス – WordPress インスタンスのデフォルトのパブリック IP アドレスは、インスタンスを停止して開始すると変わります。インスタンスに静的 IP アドレスをアタッチすると、インスタンスを停止して開始してもそのアドレスが変わりません。詳細については、「the section called “IP アドレス”」を参照してください。
- SSL/TLS 証明書 – 検証済みの証明書を作成してインスタンスにインストールしたら、WordPress ウェブサイトの HTTPS を有効にして、登録済みドメインを介してインスタンスにルーティングされるトラフィックを HTTPS を使用して暗号化できます。詳細については、「the section called “HTTPS を有効にする”」を参照してください。

> **Tip**
>
> 開始する前に、以下のヒントを確認します。トラブルシューティングの詳細については、「Troubleshooting WordPress setup」を参照してください。
>
> - インスタンスを [実行中] 状態にする必要があります。インスタンスが開始したばかりの場合、SSH 接続の準備が整うまで数分かかります。
> - インスタンスファイアウォールのポート 22、80、443 では、設定の実行中に任意の IP アドレスからの TCP 接続を許可する必要があります。詳細については、「インスタンスのファイアウォール」を参照してください。
> - 頂点ドメイン (`example.com`) とその `www` サブドメイン (`www.example.com`) からのトラフィックをポイントする DNS レコードを追加または更新する場合、それらはインターネット全体に伝播する必要があります。「nslookup」や「MxToolbox」の「DNS Lookup」などのツールを使用して、DNS の変更が有効になったことを確認できます。
> - Let's Encrypt 証明書は 60～90 日ごとに自動的に更新されます。
> - 設定が進行中の場合、インスタンスを停止したり変更を加えたりしないでください。インスタンスの設定には最大 15 分かかる場合があります。インスタンス接続タブで、各ステップの進行状況を表示できます。

ウェブサイト設定ウィザードを使用してインスタンスを設定するには

1. インスタンス管理ページの [Connect] タブで、[ウェブサイトの設定] を選択します。



2. [ドメイン名を指定する] には、既存の Lightsail マネージドドメインを使用するか、新しいドメインを Lightsail に登録するか、別のドメインレジストラを使用して登録したドメインを使用します。[このドメインを使用] を選択して次のステップに進みます。

3. [接続] で、次のいずれかを実行します:
   - [Lightsail マネージドドメイン] を選択して Lightsail DNS ゾーンを使用します。[DNS ゾーンを使用する] を選択して、次の手順に進みます。
   - [サードパーティードメイン] を選択して、ドメインの DNS レコードを管理するホスティングサービスを使用します。後で使用する場合に備えて、Lightsail アカウントに一致する DNS ゾーンを作成することに注意してください。[サードパーティーの DNS を使用する] を選択して次の手順に進みます。
4. [静的 IP アドレスを作成する] には、静的 IP アドレスの名前を入力し、[静的 IP を作成する] を選択します。
5. [ドメイン割り当ての管理] では、[割り当ての追加] を選択し、ドメインタイプを選択し、[追加] を選択します。[続行] を選択して、次の手順に進みます。
6. [SSL/TLS 証明書の作成] では、ドメインおよびサブドメインを指定してメールアドレスを入力します。その後、[Lightsail がインスタンスで Let's Encrypt 証明書を構成することを許可する] を選択して、[証明書の作成] をクリックします。Lightsail リソースの構成を開始します。

   設定が進行中の場合、インスタンスを停止したり変更を加えたりしないでください。インスタンスの設定には最大 15 分かかる場合があります。インスタンス接続タブで、各ステップの進行状況を表示できます。
7. ウェブサイトのセットアップが完了したら、ドメイン割り当てステップで指定した URL が WordPress サイトを開いていることを確認します。

ステップ 2: WordPress ウェブサイトのデフォルトのアプリケーションパスワードを取得する

WordPress ウェブサイトの管理ダッシュボードにサインインするには、デフォルトのアプリケーションパスワードが必要です。

WordPress 管理者のデフォルトパスワードを取得するには

1. WordPress インスタンスのインスタンス管理ページを開きます。
2. [WordPress] パネルで、[デフォルトのパスワードの取得] を選択します。これにより、ページの下部にある [アクセスのデフォルトパスワード] が展開されます。

3. [CloudShell の起動] を選択します。そうすることで、ページの下部に分割されたペインが開きます。
4. [コピー] を選択し、コンテンツを CloudShell ウィンドウに貼り付けます。CloudShell プロンプトにカーソルを置いて Ctrl+V を押すか、右クリックしてメニューを開き、[貼り付ける] を選択します。
5. CloudShell ウィンドウに表示されるパスワードを書き留めます。後で WordPress ウェブサイトの管理ダッシュボードにサインインするには必要です。

```
~ $ AWS_REGION=us-east-1 ~/lightsail_connect WordPress-1 tail -n 1 application_credentials
LnPLnPLnPLnPLnP9
```

ステップ 3: WordPress ウェブサイトにサインインする

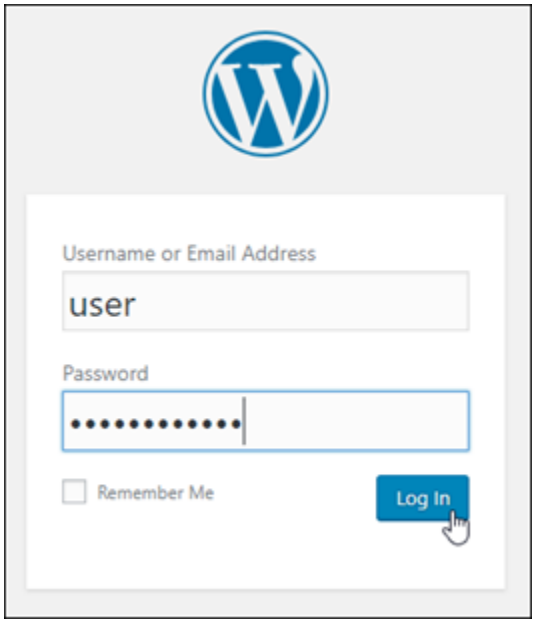
デフォルトのユーザーパスワードを取得したら、WordPress ウェブサイトのホームページに移動し、管理ダッシュボードにサインインします。サインインしたら、デフォルトのパスワードを変更できます。

管理ダッシュボードにサインインするには

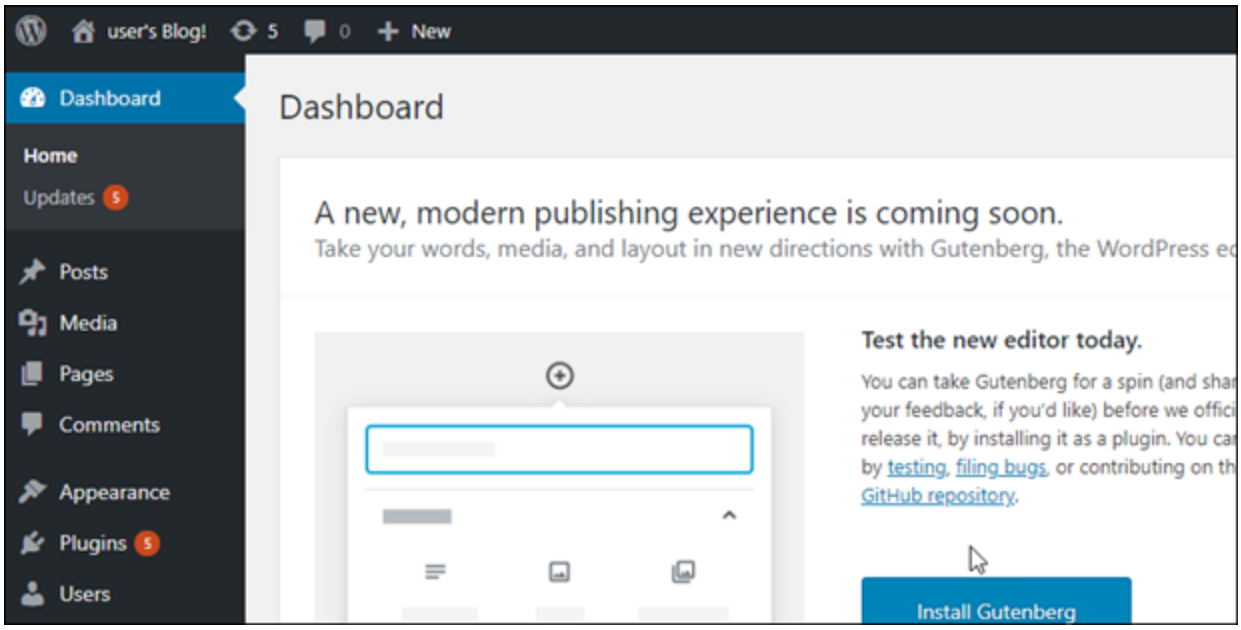
1. WordPress インスタンスのインスタンス管理ページを開きます。
2. [WordPress] パネルで、[WordPress 管理者へのアクセス] を選択します。
3. [WordPress 管理者ダッシュボードへのアクセス] パネルで、[パブリック IP アドレスを使用する] で、次のフォーマットを使用するリンクを選択します:

   http://*public-ipv4-address* ./wp-admin

4. [ユーザー名またはメールアドレス] ボックスに **user** と入力します。
5. [パスワード] には、前の手順で取得したパスワードを入力します。
6. [ログイン] を選択します。



WordPress ウェブサイトの管理ダッシュボードにサインインしたら、管理アクションを実行できます。WordPress ウェブサイトの管理の詳細については、WordPress のドキュメントで「WordPress Codex」を参照してください。

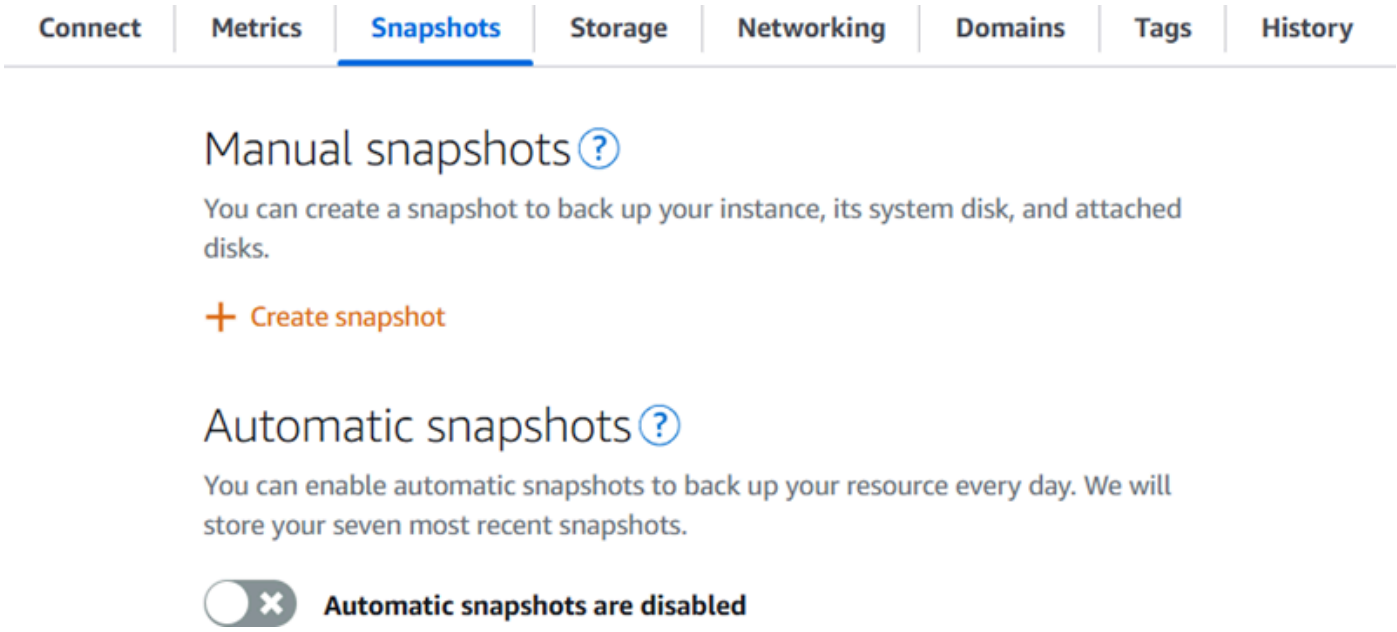


ステップ 4: インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。



Lightsail で WordPress マルチサイトを設定する

> **知っていましたか?**
>
> Lightsail は、毎日 7 つのスナップショットを保存し、インスタンスの自動スナップショットを有効にすると、最も古いスナップショットを自動的に最新のスナップショットに置き換えます。詳細については、Lightsail「インスタンスとディスクの自動スナップショットを設定する」を参照してください。

Amazon Lightsail で起動した WordPress Multisite インスタンスの使用を開始するステップについて説明します。

目次

- ステップ 1: Bitnami のドキュメントを確認する
- ステップ 2: WordPress の管理ダッシュボードにアクセスするため、デフォルトのアプリケーションパスワードを取得する
- ステップ 3: インスタンスに静的 IP アドレスをアタッチする
- ステップ 4: WordPress Multisite ウェブサイトの管理ダッシュボードにサインインする
- ステップ 5: 登録済みドメイン名へのトラフィックを WordPress Multisite ウェブサイトに送信する
- ステップ 6: WordPress Multisite ウェブサイトにブログをドメインまたはサブドメインとして追加する
- ステップ 7: WordPress Multisite のドキュメントを読み、引き続きウェブサイトの設定を続行する
- ステップ 8: インスタンスのスナップショットを作成する

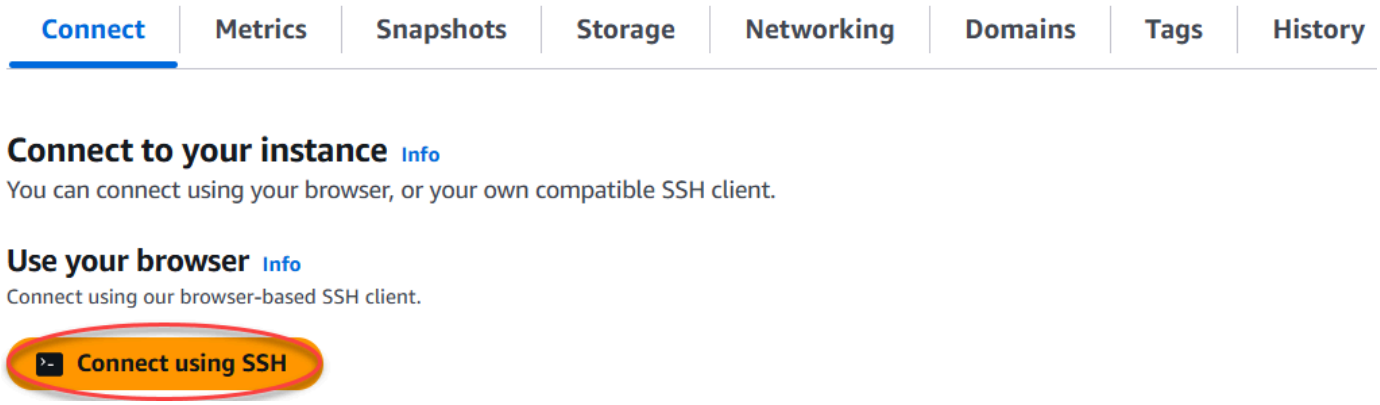
ステップ 1: Bitnami のドキュメントを確認する

Bitnami のドキュメントを読んで WordPress Multisite インスタンスを設定する方法を確認します。詳細については、「AWS クラウド用に Bitnami がパッケージ化した WordPress マルチサイト」を参照してください。

ステップ 2: WordPress の管理ダッシュボードにアクセスするため、デフォルトのアプリケーションパスワードを取得する

次の手順を完了して、WordPress Multisite ウェブサイトの管理ダッシュボードにアクセスする際に必要となるデフォルトのアプリケーションパスワードを取得します。詳細については、「Amazon Lightsailの Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

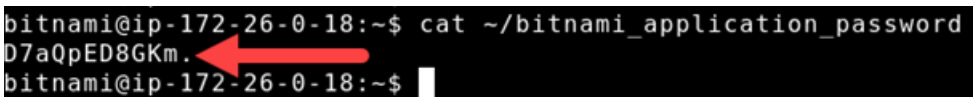
1. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。



2. 接続後に、次のコマンドを入力してデフォルトのアプリケーションのパスワードを取得します。

```
cat $HOME/bitnami_application_password
```

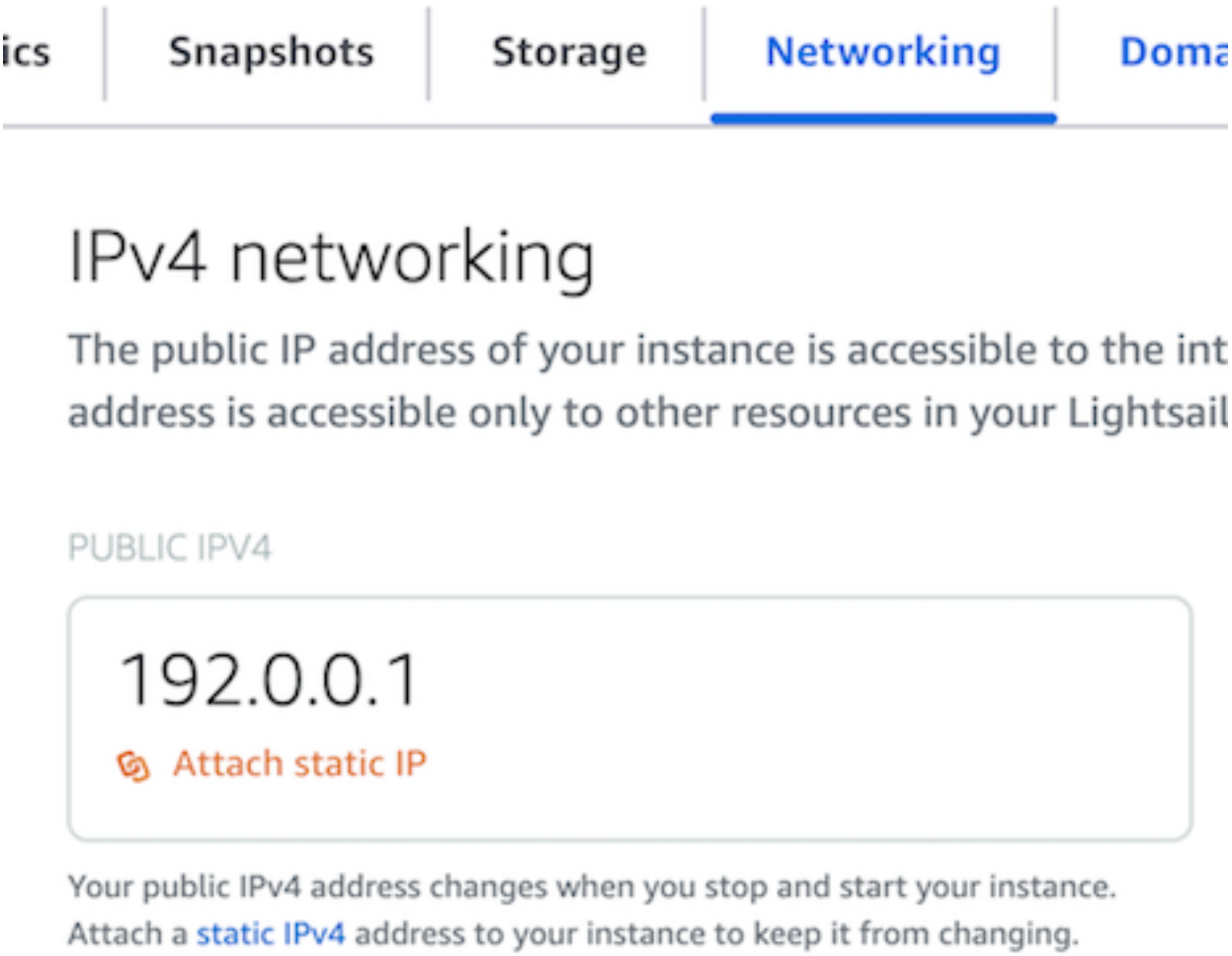
アプリケーションのデフォルトパスワードを含んだ、次の例のようなレスポンスが表示されます。このパスワードを使用して、WordPress マルチサイトウェブサイトの管理ダッシュボードにサインインします。



ステップ 3: インスタンスに静的 IP アドレスをアタッチする

インスタンスにアタッチしたデフォルトの動的なパブリック IP アドレスは、インスタンスを停止して開始するたびに変わります。静的 IP アドレスを作成してインスタンスにアタッチし、パブリック IP アドレスの変更を防止します。その後にドメイン名をインスタンスで使用すると、インスタンスを停止して開始するたびにドメインの DNS レコードを更新する必要がなくなります。各インスタンスには、1 つの静的 IP アドレスしか追加できません。

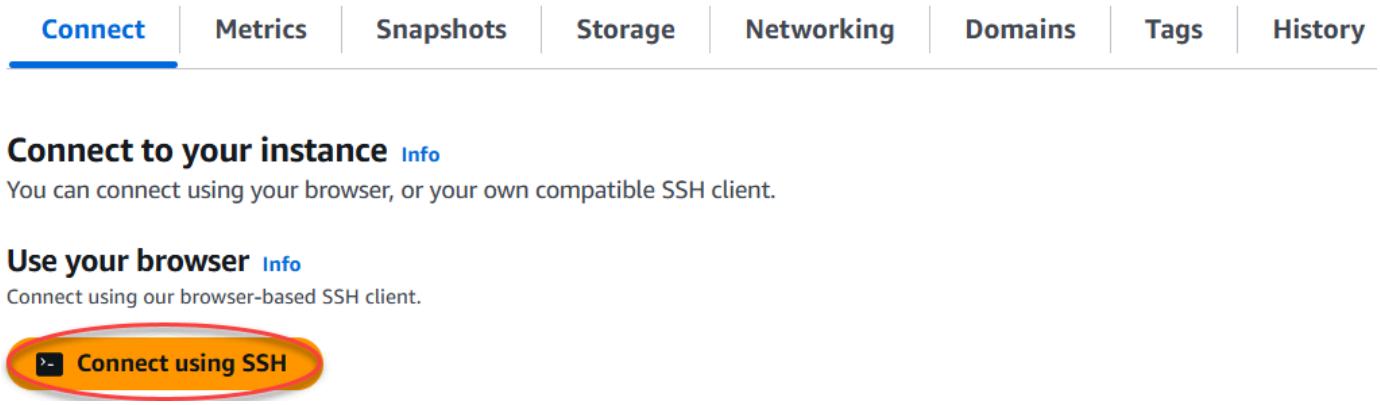
インスタンス管理ページの [ネットワーク] タブで、[静的 IP の作成] または [静的 IP のアタッチ] (インスタンスにアタッチできる静的 IP を既に作成している場合) を選択して、ページの手順に従います。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。



新しい静的 IP アドレスがインスタンスに添付されたら、次の手順を実行して、WordPress に新しい静的 IP アドレスを認識させる必要があります。

1. インスタンスの新しい静的 IP アドレスは書き留めておきます。このIP アドレスはインスタンス管理ページの ヘッダーセクションに表示されます。

2. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。



3. 接続後に、次のコマンドを入力します。*<StaticIP>* をインスタンスの新しい静的 IP アドレスに置き換えます。

```
sudo /opt/bitnami/configure_app_domain --domain <StaticIP>
```

例:

```
sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
```

次の例のようなレスポンスが表示されます。これで、インスタンスの WordPress ウェブサイトが新しい静的 IP アドレスを認識するようになります。

```
bitnami@ip-[redacted]:~$ sudo /opt/bitnami/configure_app_domain --domain 203.0.113.0
Configuring domain to 203.0.113.0
2021-03-12T15:49:22.000Z - info: Saving configuration info to disk
prestashop 15:49:22.41 INFO  ==> Trying to connect to the database server
prestashop 15:49:22.44 INFO  ==> Updating hostname in database
prestashop 15:49:22.46 INFO  ==> Purging cache
Disabling automatic domain update for IP address changes
```

> **Note**
>
> このコマンドが失敗した場合、古いバージョンの WordPress マルチサイトインスタンスを使用している可能性があります。代わりに次のコマンドを実行してみてください。*<StaticIP>* をインスタンスの新しい静的 IP アドレスに置き換えます。
>
> ```
> cd /opt/bitnami/apps/wordpress
> sudo ./bnconfig --machine_hostname <StaticIP>
> ```

コマンドの実行が終了したら次のコマンドを入力し、サーバーが再起動するたびに bnconfig ツールが自動的に実行されないようにします。

```
sudo mv bnconfig bnconfig.disabled
```

ステップ 4: WordPress マルチサイトウェブサイトの管理ダッシュボードにサインインする

デフォルトのユーザーパスワードを取得したら、以下の手順に従ってWordPress マルチサイトウェブサイトのホームページに移動し、管理ダッシュボードにサインインします。サインイン後に、ウェブサイトをカスタマイズしたり管理上の変更を行うことができます。WordPress で実行できる事項の詳細については、本ガイドの後半にある「ステップ 7: WordPress Multisite のドキュメントを読み、引き続きウェブサイトの設定を続行する」のセクションを参照してください。

1. インスタンス管理ページの [Connect] (接続) タブにあるパブリック IP アドレスを書き留めます。パブリック IP アドレスは、インスタンス管理ページのヘッダーセクションにも表示されます。

2. インスタンスのパブリック IP アドレスを参照します (例: `http://203.0.113.0` に移動します)。

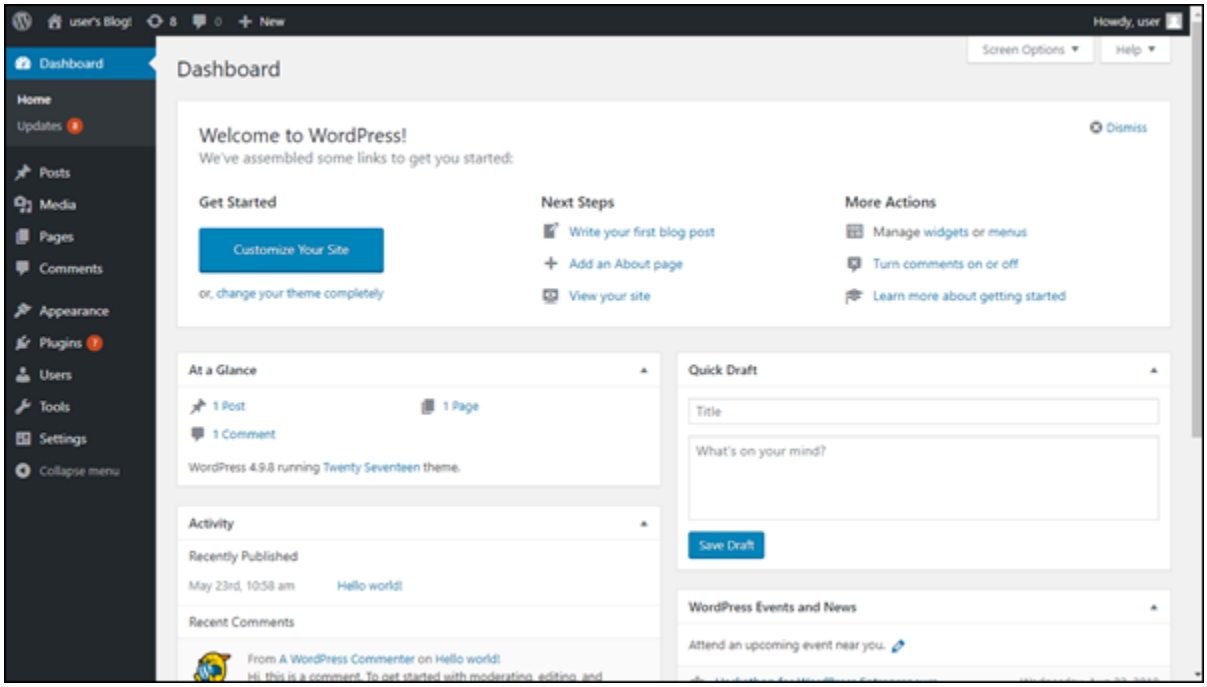
   WordPress ウェブサイトのホームページが表示されます。

3. WordPress ウェブサイトのホームページで、右下にある [Manage] (管理) を選択します。

   [Manage] (管理) バナーが表示されない場合は、`http://<PublicIP>/wp-login.php` を参照することでサインインページにアクセスすることができます。*<PublicIP>* を、インスタンスのパブリック IP アドレスに置き換えます。

4. デフォルトのユーザー名 (`user`) と、先ほど取得したデフォルトのパスワードを使用してサインインします。

   WordPress の管理ダッシュボードが表示されます。



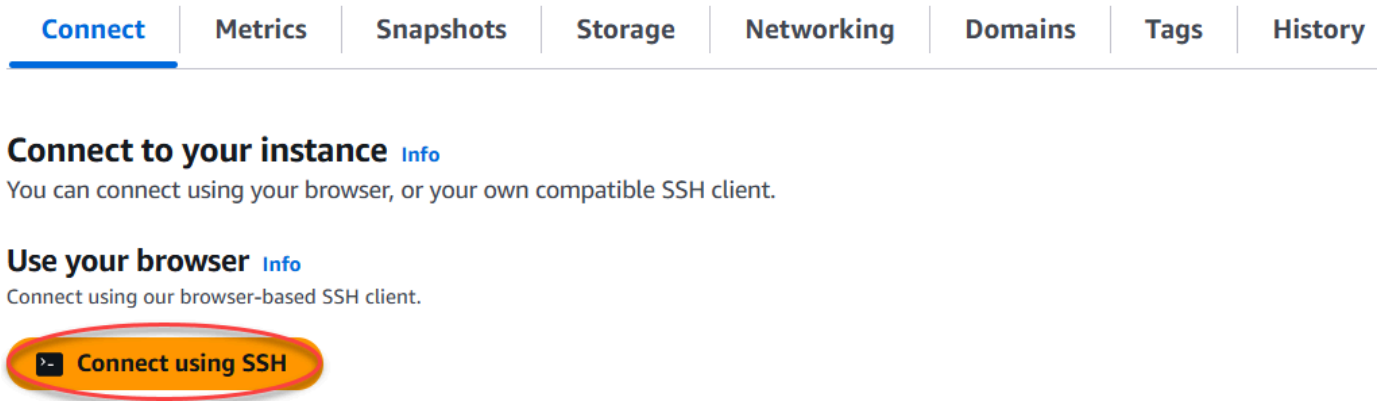
ステップ 5: 登録済みドメイン名へのトラフィックを WordPress マルチサイトウェブサイトに送信する

`example.com` などの登録済みドメイン名のトラフィックを WordPress Multisite ウェブサイトに送信するには、ドメインの DNS にレコードを追加します。DNS レコードは、通常、ドメインの登録先であるレジストラが管理またはホストします。ただし、ドメインの DNS レコードの管理を Lightsail に引き渡して、Lightsail コンソールで管理できるようにすることをお勧めします。

Lightsail コンソールのホームページの [Domains & DNS] (ドメインと DNS) タブで、[Create DNS zone] (DNS ゾーンの作成) を選択し、ページに記載される手順に従います。詳細については、「Lightsail で DNS ゾーンを作成し、ドメインの DNS レコードを管理する」を参照してください。

ドメイン名へのトラフィックがインスタンスにルーティングされたら、次の手順を実行して、WordPress にドメイン名を認識させます。

1. インスタンス管理ページの [Connect] (接続) タブで、[SSH を使用して接続] を選択します。



2. 接続後に、次のコマンドを入力します。*<DomainName>* は、インスタンスにトラフィックをルーティングするドメイン名に置き換えてください。

```
sudo /opt/bitnami/configure_app_domain --domain <DomainName>
```

例:

```
sudo /opt/bitnami/configure_app_domain --domain www.example.com
```

次の例のようなレスポンスが表示されます。これで、 WordPress マルチサイトソフトウェアがドメイン名を認識できるようになります。

```
bitnami@ip-[redacted]:~$ sudo /opt/bitnami/configure_app_domain --domain www.example.com
Configuring domain to www.example.com
2021-03-12T15:49:22.000Z - info: Saving configuration info to disk
prestashop 15:49:22.41 INFO  ==> Trying to connect to the database server
prestashop 15:49:22.44 INFO  ==> Updating hostname in database
prestashop 15:49:22.46 INFO  ==> Purging cache
Disabling automatic domain update for IP address changes
```

> **Note**
>
> このコマンドが失敗した場合、古いバージョンの WordPress マルチサイトインスタンスを使用している可能性があります。代わりに次のコマンドを実行してみてください。*<DomainName>* は、インスタンスにトラフィックをルーティングするドメイン名に置き換えてください。
>
> ```
> cd /opt/bitnami/apps/wordpress
> sudo ./bnconfig --machine_hostname <DomainName>
> ```

コマンドの実行が終了したら次のコマンドを入力し、サーバーが再起動するたびに bnconfig ツールが自動的に実行されないようにします。

```
sudo mv bnconfig bnconfig.disabled
```

インスタンスに設定したドメイン名を参照すると、WordPress マルチサイトウェブサイトのホームページへと移動します。次に、WordPress マルチサイトウェブサイトにブログをドメインとして追加するか、サブドメインとして追加するかを決定する必要があります。詳細については、本ガイドの次のステップにある「ステップ 6: WordPress Multisite ウェブサイトにブログをドメインまたはサブドメインとして追加する」セクションを参照してください。

ステップ 6: WordPress マルチサイトウェブサイトにブログをドメインまたはサブドメインとして追加する

WordPress マルチサイトは、WordPress の 1 つのインスタンスで複数のブログウェブサイトをホストするように設計されています。新しいブログウェブサイトを WordPress マルチサイトに追加すると、独自のドメインまたは WordPress マルチサイトのプライマリドメインのサブドメインを使用するように設定できます。WordPress マルチサイトで使用するように設定できるのは、これらのオプションのうちの 1 つのみです。例えば、ブログサイトをドメインとして追加する場合、ブログサイトをサブドメインとして追加することはできず、またその逆も同様です。これらのオプションのいずれかを設定するには、次の説明のいずれかに従います。

- ブログサイトを `example1.com` や `example2.com` などのドメインとして追加する場合は、「Lightsail の WordPress Multisite インスタンスにブログをドメインとして追加する」を参照してください。
- ブログサイトを `one.example.com` や `two.example.com` などの WordPress Multisite のプライマリドメインのドメインとして追加する場合は、「Lightsail の WordPress Multisite インスタンスにブログをサブドメインとして追加する」を参照してください。

ステップ 7: WordPress Multisite のドキュメントを読み、引き続きウェブサイトの設定を続行する

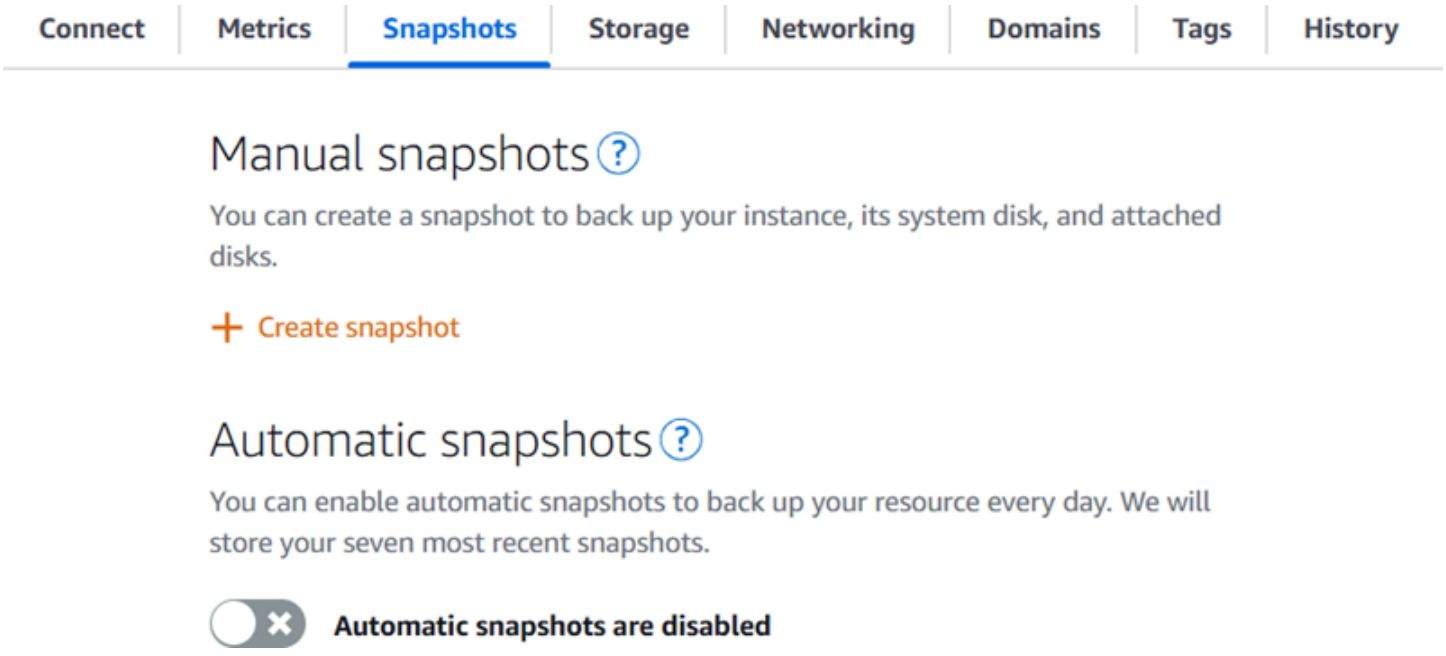
WordPress マルチサイトのドキュメントを読み、ウェブサイトを管理およびカスタマイズする方法を確認します。詳細については、「WordPress Multisite ネットワーク管理ドキュメント」を参照してください。

ステップ 8: インスタンスのスナップショットを作成する

ウェブサイトを必要に応じて設定したら、インスタンスの定期的なスナップショットを作成してバックアップします。スナップショットは、インスタンスのシステムディスクおよびオリジナル設定のコピーです。スナップショットには、インスタンスの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。

スナップショットは手動で作成するか、自動スナップショットを有効にして Lightsail に毎日のスナップショットを作成させることができます。インスタンスに問題が発生した場合は、スナップショットを使用して新しい代替インスタンスを作成できます。

[スナップショット] タブにあるインスタンスの管理ページでスナップショットを操作できます。詳細については、Amazon Lightsail のスナップショット を参照してください。



Lightsail での Bitnami アプリケーションとスタックの操作

このセクションでは、Amazon Lightsail インスタンスの Bitnami アプリケーションに関連する以下のトピックについて説明します:

トピック

- Lightsail Bitnami インスタンス用のアプリケーションのユーザー名とパスワードの取得
- Lightsail インスタンスから Bitnami バナーを削除する

Lightsail Bitnami インスタンス用のアプリケーションのユーザー名とパスワードの取得

Bitnami は、Amazon Lightsail インスタンス (仮想プライベートサーバー) として作成できるアプリケーションインスタンスイメージ (設計図) の多くを提供します。これらの設計図は、Lightsail コンソールのインスタンス作成ページに「Packaged by Bitnami」として示されます。

Bitnami ブループリントを使用してインスタンスを作成したら、このインスタンスにサインインして管理します。これを行うには、インスタンスで実行されるアプリケーションやデータベースのデフォルトのユーザー名とパスワードを取得する必要があります。この記事では、以下の設計図から作成された Lightsail インスタンスにサインインして管理するために必要な情報を取得する方法を示します。

- WordPress ブログおよびコンテンツ管理アプリケーション
- 同じインスタンスで複数のウェブサイトをサポートする WordPress マルチサイトブログおよびコンテンツ管理アプリケーション
- Django 開発スタック
- Ghost ブログおよびコンテンツ管理アプリケーション
- LAMP 開発スタック (PHP 7)
- Node.js 開発スタック
- Joomla コンテンツ管理アプリケーション
- Magento e コマースアプリケーション
- MEAN 開発スタック
- Drupal コンテンツ管理アプリケーション
- GitLab CE リポジトリアプリケーション
- Redmine プロジェクト管理アプリケーション
- Nginx (LEMP) 開発スタック

Bitnami アプリケーションおよびデータベースのデフォルトユーザー名を取得する

以下は、Bitnami 設計図を使用して作成された Lightsail インスタンスのアプリケーションおよびデータベースのデフォルトユーザー名です。

> **Note**
>
> すべての Bitnami 設計図にアプリケーションやデータベースが含まれているわけではありません。設計図にアプリケーションやデータベースが含まれていない場合、ユーザー名は該当なし (N/A) として表示されます。

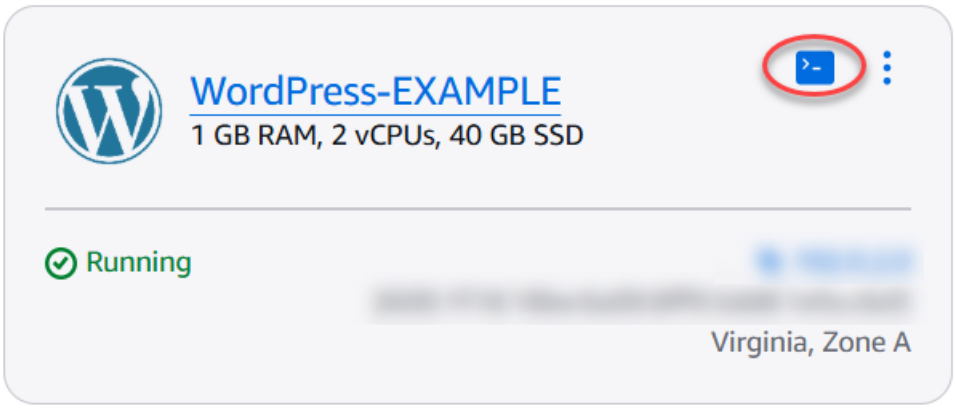
アプリケーション名	アプリケーションユーザー名	Database user name
WordPress (WordPress Multisite を含む)	ユーザー	root
PrestaShop	user@example.com	root
Django	該当なし	root
Ghost	user@example.com	root
LAMP スタック (PHP 5 および PHP 7)	該当なし	root
Node.js	該当なし	該当なし
Joomla	ユーザー	root
Magento	ユーザー	root
MEAN	該当なし	root
Drupal	ユーザー	root
GitLab CE	ユーザー	postgres
Redmine	ユーザー	root
Nginx	該当なし	root

Bitnami アプリケーションおよびデータベースのデフォルトパスワードを取得する

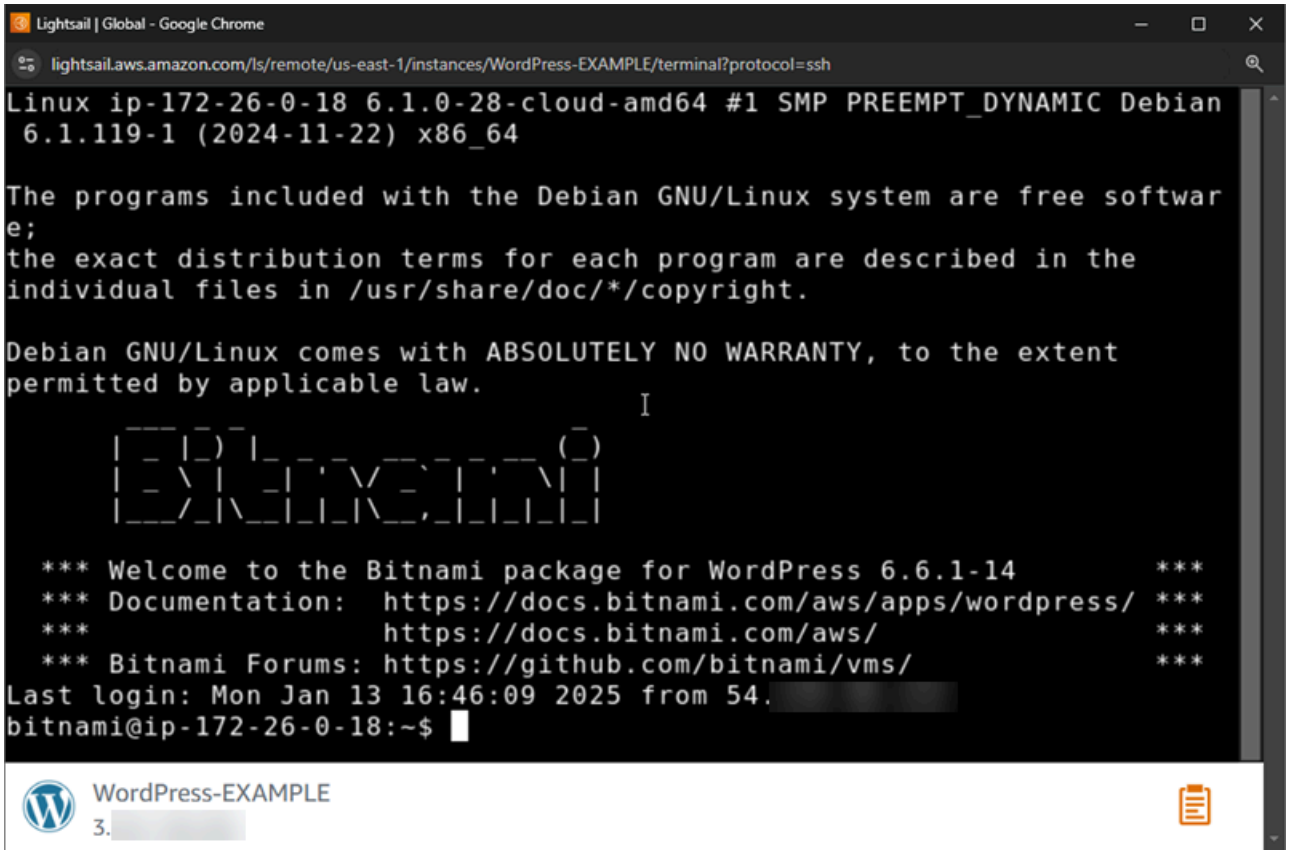
アプリケーションおよびデータベースのデフォルトパスワードはインスタンスに保存されています。これを取得するには、Lightsail コンソールのブラウザベースの SSH ターミナルを使用し、専用のコマンドを実行して接続します。

Bitnami アプリケーションおよびデータベースのデフォルトパスワードを取得するには

1. Lightsail コンソール にサインインします。
2. Bitnami ブループリントを使用してインスタンスを作成します (まだ作成していない場合)。詳細については、「Amazon Lightsail VPS を作成する」を参照してください。
3. Lightsail のホームページで、接続先のインスタンスのクイック接続アイコンを選択します。



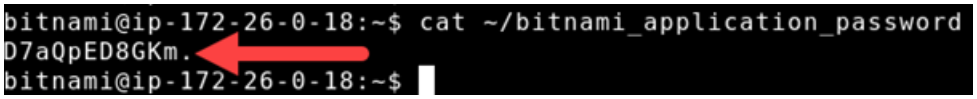
以下の例に示すように、ブラウザベースの SSH クラインとウィンドウが開きます。



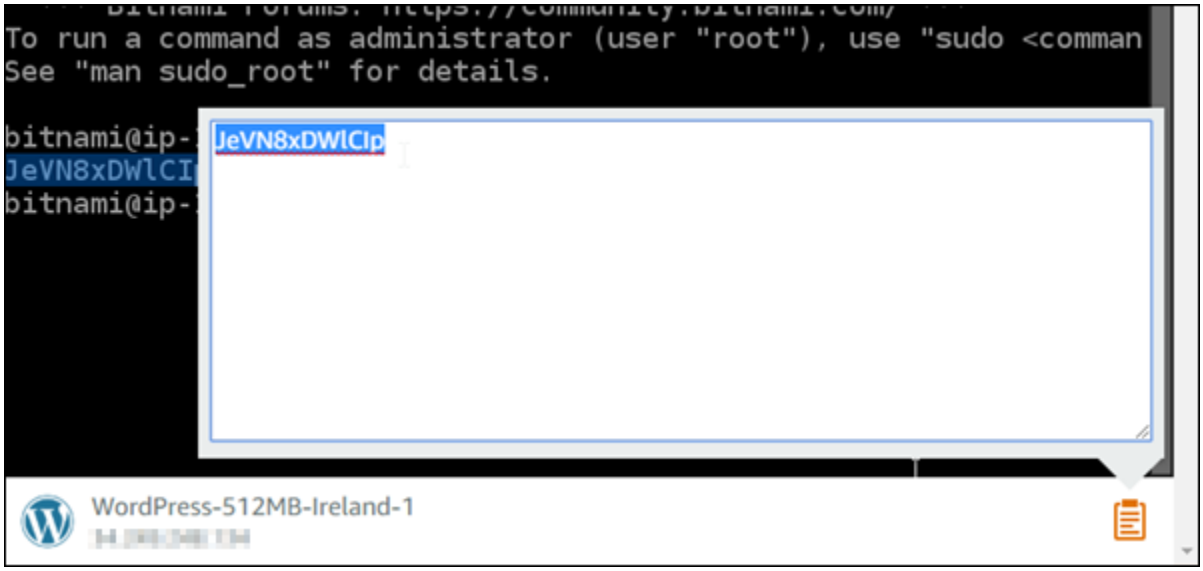
4. 次のコマンドを入力してデフォルトのアプリケーションのパスワードを取得します。

```
cat ~/bitnami_application_password
```

次のようなレスポンスにアプリケーションのパスワードが表示されます。



5. ターミナル画面でパスワードを強調表示し、ブラウザベースの SSH クライアントウィンドウの右下でクリップボードアイコンを選択します。
6. クリップボードテキストボックスで、コピーするテキストを強調表示し、Ctrl+C または Cmd+C を押してテキストをローカルクリップボードにコピーします。



> **Important**
>
> この時点で、任意の場所にパスワードを保存します。インスタンスの Bitnami アプリケーションにサインインした後に変更することもできます。

インスタンスで Bitnami アプリケーションにサインインする

WordPress、Joomla、Magento、Drupal、GitLab CE、および Redmine 設計図から作成したインスタンスの場合は、インスタンスのパブリック IP アドレスを参照してアプリケーションにサインインします。

Bitnami アプリケーションにサインインするには

1. ブラウザウィンドウで、インスタンスのパブリック IP アドレスに移動します。

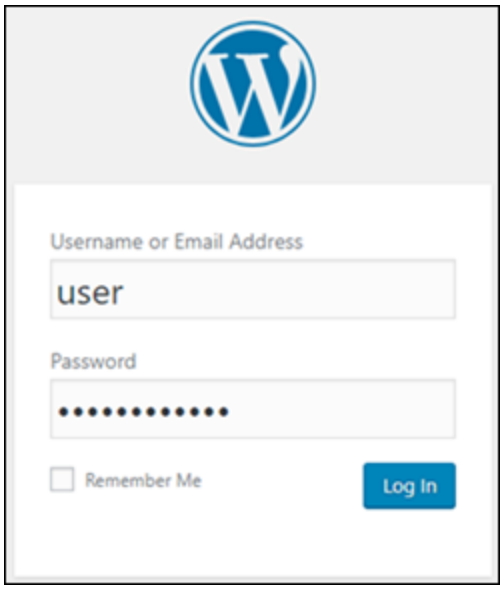
   Bitnami アプリケーションのホームページが開きます。ホームページは、インスタンスで選択した Bitnami 設計図に応じて表示されます。たとえば、これは WordPress アプリケーションのホームページです。

2. アプリケーションのホームページの右下にある Bitnami ロゴを選択し、アプリケーション情報ページに移動します。

> **Note**
>
> GitLab CE アプリケーションには Bitnami ロゴが表示されません。代わりに、GitLab CE ホームページに表示されるユーザー名とパスワードのテキストフィールドを使用してサインインします。

アプリケーション情報ページには、インスタンスのアプリケーションのユーザー名とログインページへのリンクが表示されます。

3. ページのログインリンクを選択し、インスタンスのアプリケーションのログインページに移動します。
4. 取得したユーザー名とパスワードを入力し、[Log In (ログイン)] を選択します。



次の手順

以下のリンクを使用して、Bitnami 設計図の詳細を確認し、チュートリアルを表示します。たとえば、WordPress インスタンスのプラグインをインストールしたり、SSL 証明書を使用して HTTPS サポートを有効化したりできます。

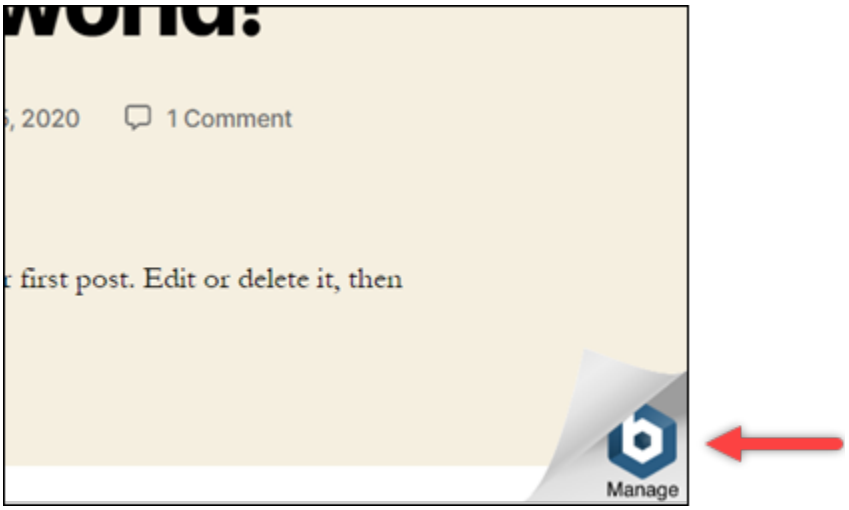
- Bitnami WordPress for Amazon Web Services

- Bitnami LAMP stack for Amazon Web Services
- Bitnami Node.js for Amazon Web Services
- Bitnami Joomla for Amazon Web Services
- Bitnami Magento for Amazon Web Services
- Bitnami MEAN stack for Amazon Web Services
- Amazon Web Services Bitnami Drupal
- Bitnami GitLab for Amazon Web Services
- Bitnami Redmine for Amazon Web Services
- Bitnami Nginx (LEMP stack) for Amazon Web Services

詳細については、「Get Started with Bitnami Applications using Amazon Lightsail」または「Using Amazon Lightsail FAQ」を参照してください。

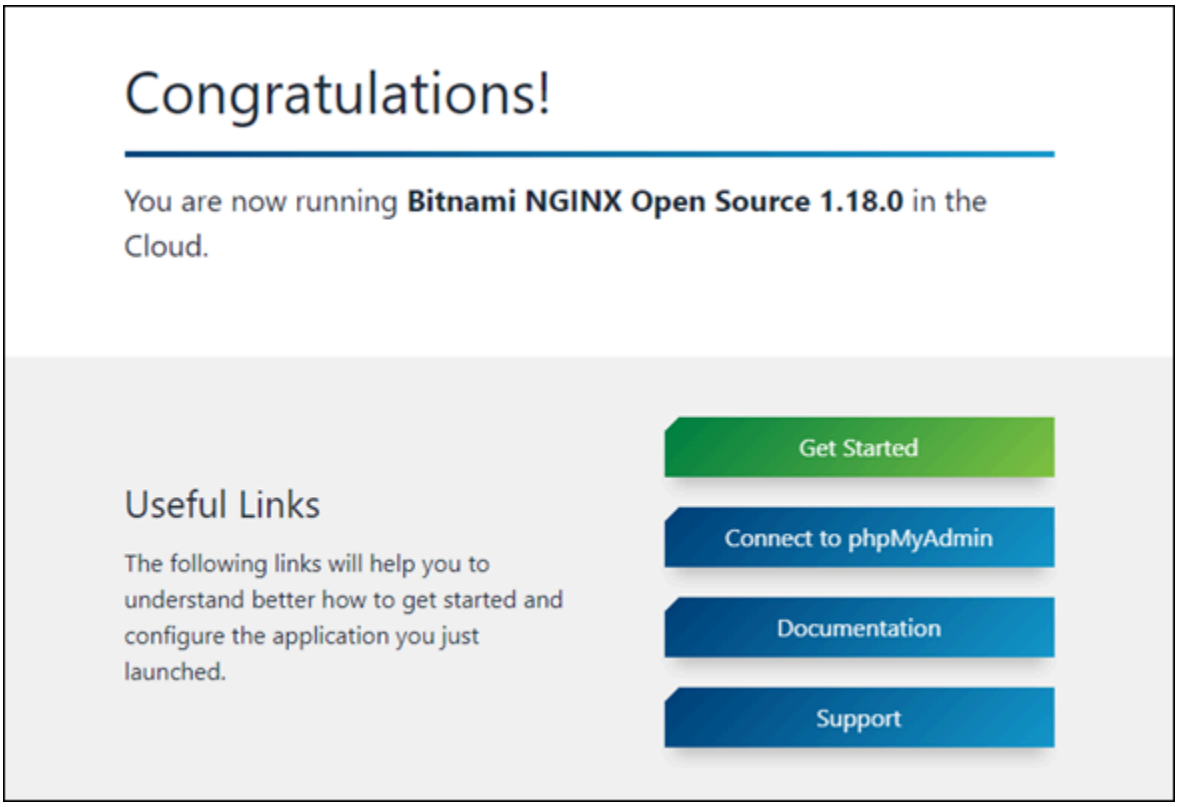
Lightsail インスタンスから Bitnami バナーを削除する

Amazon Lightsail インスタンス用に選択することができる Bitnami ブループリントの一部は、アプリケーションのホームページに Bitnami バナーを表示します。「Certified by Bitnami」WordPress インスタンスの以下の例では、Bitnami バナーはホームページの右下隅に表示されています。このガイドでは、インスタンスのアプリケーションのホームページから Bitnami アイコンを完全に削除する方法を解説しています。



すべての Bitnami ブループリントアプリケーションが、アプリケーションのホームページに Bitnami バナーを表示するわけではありません。Lightsail インスタンスのホームページに移動して、Bitnami バナーが表示されるかどうかを確認してください。「Packaged by Bitnami」Nginx インスタンスの

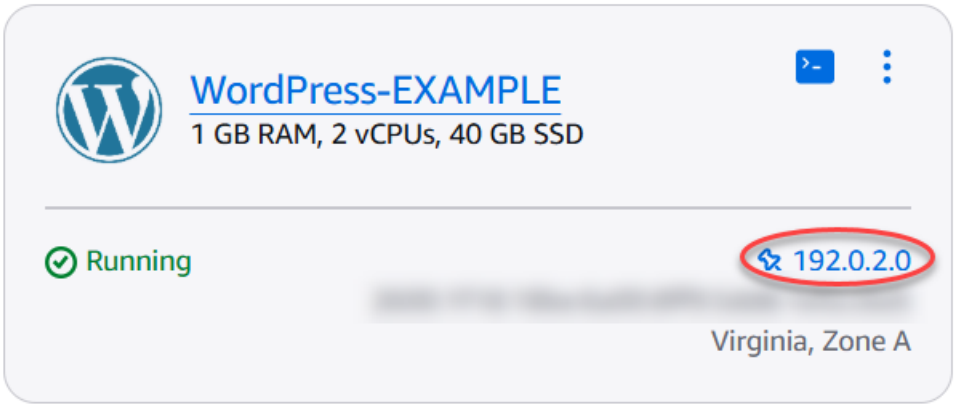
次の例では、Bitnami アイコンは表示されていません。代わりに、プレースホルダー情報ページが表示されます。このページは、最終的にインスタンスにデプロイすることを選択したアプリケーションに置き換えられます。インスタンスに Bitnami バナーが表示されない場合は、このガイドの手順に従う必要はありません。



インスタンスから Bitnami バナーを削除する

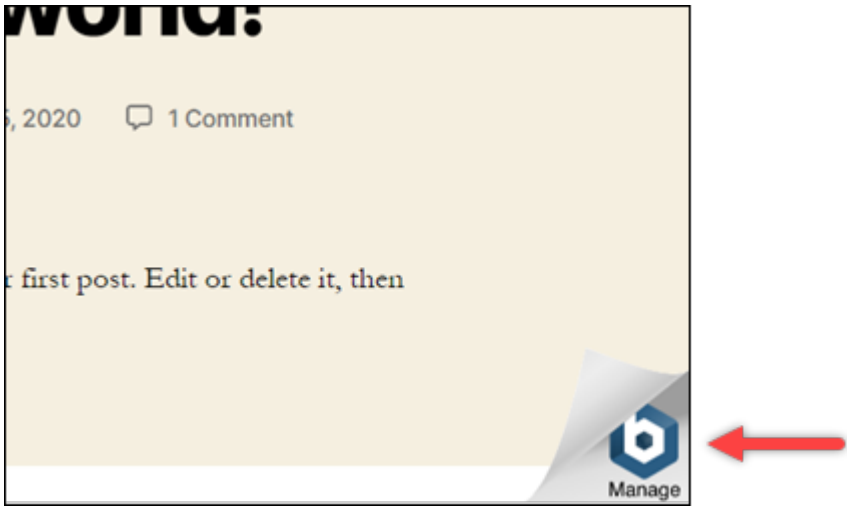
次の手順を実行して、インスタンスのアプリケーションのホームページに Bitnami アイコンが表示されていることを確認し、削除します。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの [インスタンス] セクションで、確認するインスタンスのパブリック IP アドレスをコピーします。



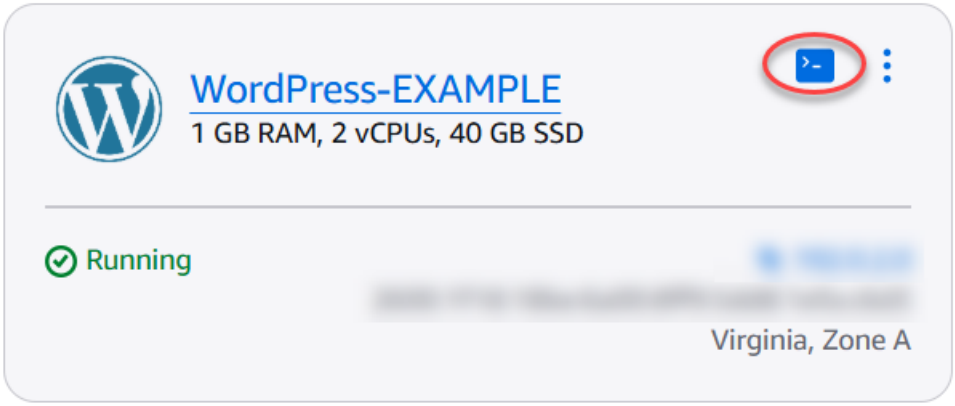
3. 新しいブラウザタブを開き、インスタンスのパブリック IP アドレスをアドレスバーに入力し、Enter を押します。
4. 以下のいずれかのオプションを確認します:

   1. Bitnami アイコンがページに表示されていない場合は、以下の手順に従う必要はありません。アプリケーションのホームページから Bitnami アイコンを削除する必要はありません。
   2. 次の例に示すように、Bitnami アイコンがページの右下隅に表示されている場合は、以下の一連のステップに従ってアイコンを削除します。



   以下の一連のステップでは、Lightsail ブラウザベースの SSH クライアントを使用してインスタンスに接続します。接続後、Bitnami 設定ツール (bnconfig) ツールを実行して、アプリケーションのホームページから Bitnami アイコンを削除します。bnconfig ツールは、 Bitnami ブループリントインスタンス上のアプリケーションを設定できるコマンドラインツールです。詳細については、Bitnami ドキュメントの「Bitnami 設定ツールの詳細を確認」を参照してください。
5. Lightsail ホームページ上にある [ブラウザ] タブに戻ります。

6. 接続先のインスタンス名の横に表示されているブラウザベースの SSH クライアントアイコンを選択します。



7. SSH クライアントがインスタンスに接続されたら、以下のいずれかのコマンドを入力します。

   1. インスタンスが Apache を使用している場合は、以下のコマンドのいずれかを入力します。一方のコマンドが失敗した場合は、他方のコマンドを試してください。このコマンドの前半部分が Bitnami バナーを無効にし、後半部分が Apache サービスを再起動させます。

      ```
      sudo /opt/bitnami/apps/wordpress/bnconfig --disable_banner 1 && sudo /opt/
      bitnami/ctlscript.sh restart apache
      ```

      ```
      sudo /opt/bitnami/wordpress/bnconfig --disable_banner 1 && sudo /opt/bitnami/
      ctlscript.sh restart apache
      ```

      プロセスが成功したことを確認するには、インスタンスのパブリック IP アドレスを参照し、Bitnami アイコンが表示されていないことを確認します。

段階的な手順に従い、Bitnami アプリケーションおよびデータベースのデフォルト資格情報を取得し、アプリケーションの管理パネルにサインインする方法を説明します。また、必要に応じてアプリケーションのホームページから Bitnami ブランドのバナーを削除することも可能です。

このガイドでは、WordPress 、Joomla、Drupal、Ghost、LAMP、LEMP、MEAN、Node.js など、Lightsail で利用可能なさまざまな Bitnami ブループリントについて説明します。アプリケーションとデータベースの両方のデフォルトのユーザー名と、デフォルトのパスワードを安全に取得するためのコマンドを提供します。このガイドに従うことで、Lightsail インスタンスで実行されている Bitnami アプリケーションに簡単にアクセスして管理し、要件に応じてカスタマイズし、不要なブランド要素を削除できます。

WordPress Lightsail インスタンスの設定と管理

このガイドでは、Lightsail の WordPress インスタンスに関連する以下のトピックについて説明します:

トピック

- Lightsail で WordPress インスタンスを起動して設定する
- WP Offload Media を使用して Lightsail の WordPress ウェブサイトを Amazon S3 に接続する
- Lightsail WordPress インスタンスを Amazon Aurora データベースに接続する
- Lightsail の MySQL マネージド型データベースに WordPress データを転送する
- WordPress インスタンスを静的コンテンツ用の Lightsail バケットに接続する
- Lightsail コンテンツ配信ネットワークで WordPress を設定する
- Lightsail の WordPress インスタンスのメールを有効にする
- Lightsail で HTTPS を使用して WordPress サイトを保護する
- WordPress ブログを Lightsail に移行する。

Lightsail で WordPress インスタンスを起動して設定する

Amazon Lightsail は Amazon Web Services (AWS) を始める最も簡単な方法です。Lightsail にはプロジェクトを迅速に立ち上げるために必要なものがすべて揃っています。予測可能な低料金でインスタンス (仮装プライベートサーバー)、マネージドデータベース、SSD ベースのストレージ、バックアップ (スナップショット)、データ転送、ドメイン DNS 管理、静的 IP、ロードバランサーが提供されます。

このチュートリアルでは、Lightsail で WordPress インスタンスを起動および設定する方法について説明します。カスタムドメイン名の設定、HTTPS によるインターネットトラフィックの保護、SSH を使用したインスタンスへの接続、WordPress ウェブサイトへのサインインなどの手順が含まれています。このチュートリアルを完了すると、Lightsail でインスタンスを起動して使用するための基礎固めができます。

> **Note**
>
> AWS 無料利用枠の一部として、一部のインスタンスバンドルで Amazon Lightsailを無料で開始できます。詳細については、「Amazon Lightsail 料金表」ページの「AWS Free Tier」を参照してください。

前提条件

ステップ 1: にサインアップする AWS

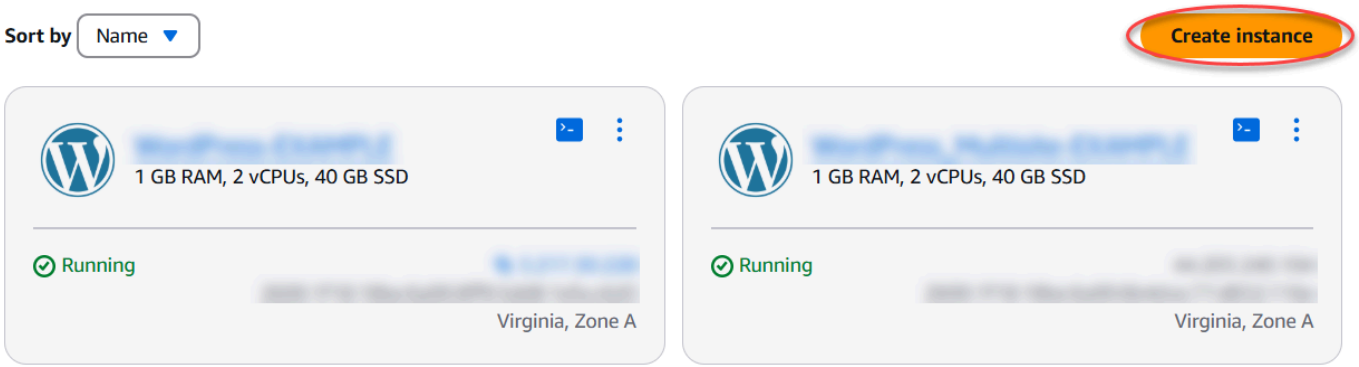
Amazon Lightsail には が必要です AWS アカウント。にサインアップ AWSするか、既にアカウントをお持ちの場合は にサインイン AWSします。

ステップ 2: WordPress インスタンスを作成する

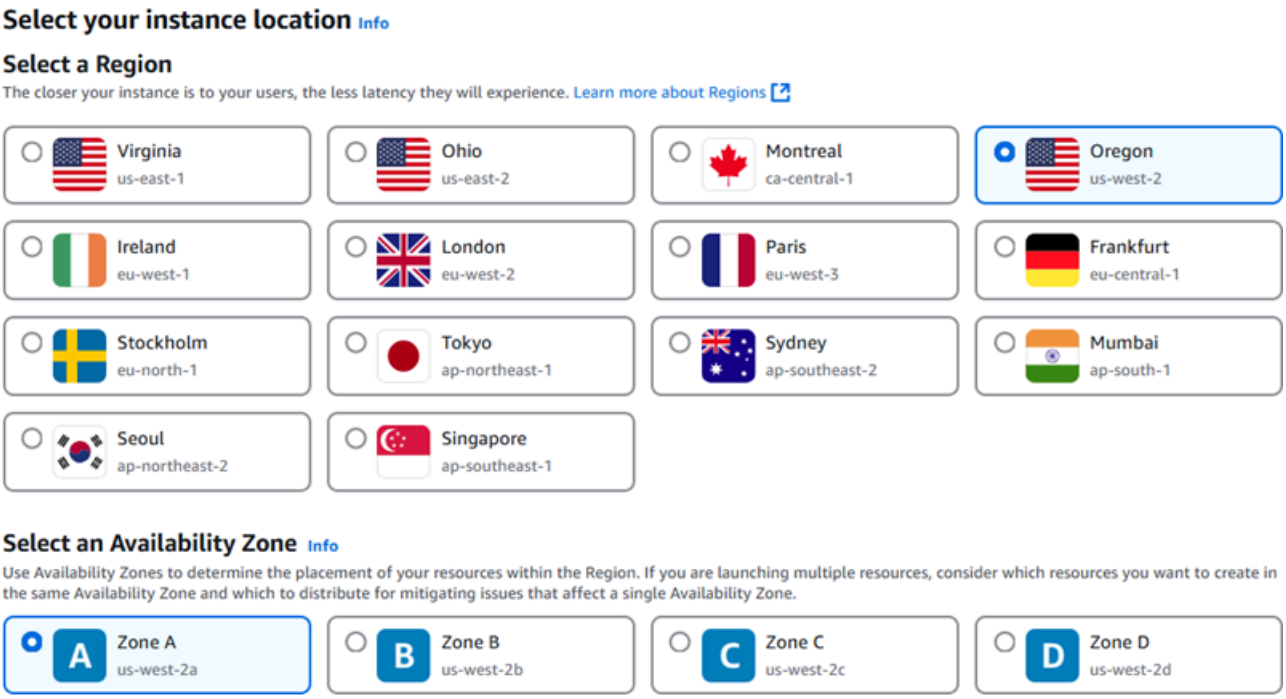
WordPress インスタンスを稼働状態にするには、以下の手順を実行します。詳細については、「the section called “インスタンスを作成する”」を参照してください。

WordPress Lightsail インスタンスを作成するには

1. Lightsail コンソール にサインインします。
2. Lightsail ホーム画面の [インスタンス] セクションで、[インスタンスの作成] を選択します。



3. インスタンスの AWS リージョン とアベイラビリティーゾーンを選択します。



4. 次のようにインスタンスのイメージを選択します:

   a. [プラットフォームを選択する] で、[Linux/Unix] を選択します。

   b. WordPress ブループリントを選択します。

      Amazon Lightsail または Bitnami プロバイダーを選択できます。

5. インスタンスプランを選択します。

   プランには、低額で予測可能なコストを消費するマシン設定 (RAM、SSD、vCPU)、およびデータ許容量が含まれます。

6. インスタンスの名前を入力します。リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

7. [インスタンスの作成] を選択します。

8. テストブログ記事を表示するには、インスタンス管理ページに移動し、ページの右上隅に表示されるパブリック IPv4 アドレスをコピーします。インターネットに接続したウェブブラウザのアドレスフィールドにアドレスを貼り付けます。ブラウザにテストブログ記事が表示されます。

インスタンスが実行されたら、インスタンス管理ページで WordPress ブループリントベンダーを特定し、以下の適切なガイドを選択します。

Bitnami

ステップ 3: WordPress インスタンスを設定する

WordPress インスタンスは、ガイド付きのステップバイステップワークフローを使用して設定することも、個々のタスクを完了することもできます。いずれかのオプションを使用して、以下を設定します:

- 登録済みドメイン名 – WordPress サイトには、覚えやすいドメイン名が必要です。ユーザーは、WordPress サイトにアクセスするためにこのドメイン名を指定します。詳細については、「ドメインと DNS」を参照してください。
- DNS 管理 – ドメインの DNS レコードを管理する方法を決定する必要があります。DNS レコードは、DNS サーバーに対して、ドメインやサブドメインに対応する IP アドレスやホスト名を示します。DNS ゾーンには、ドメインの DNS レコードが含まれます。詳細については、「the section called “Lightsail での DNS”」を参照してください。
- 静的 IP アドレス – WordPress インスタンスのデフォルトのパブリック IP アドレスは、インスタンスを停止して開始すると変わります。インスタンスに静的 IP アドレスをアタッチすると、インスタンスを停止して開始してもそのアドレスが変わりません。詳細については、「the section called “IP アドレス”」を参照してください。
- SSL/TLS 証明書 – 検証済みの証明書を作成してインスタンスにインストールしたら、WordPress ウェブサイトの HTTPS を有効にして、登録済みドメインを介してインスタンスにルーティングされるトラフィックを HTTPS を使用して暗号化できます。詳細については、「the section called “HTTPS を有効にする”」を参照してください。

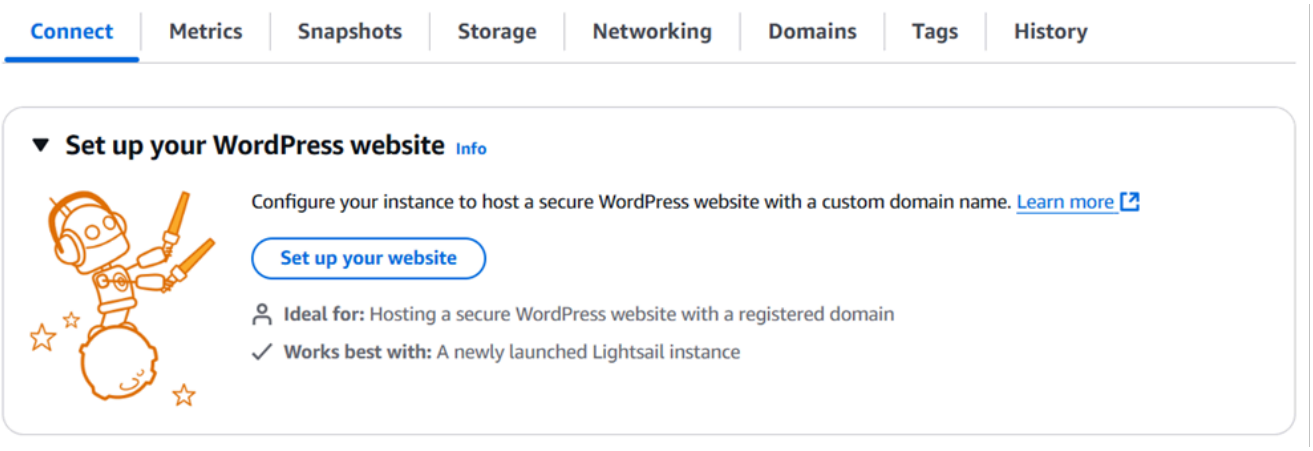
オプション: ガイド付きのワークフロー

> **Tip**
>
> 開始する前に、以下のヒントを確認します。トラブルシューティングの詳細については、「Troubleshooting WordPress setup」を参照してください。
>
> - セットアップは、2023 年 1 月 1 日以降に作成された WordPress バージョン 6 以降の Lightsail インスタンスをサポートします。

- 設定中に実行される Certbot 依存関係ファイル、HTTPS 書き換えスクリプト、および証明書更新スクリプトは、インスタンスの `/opt/bitnami/lightsail/scripts/` ディレクトリに保存されます。
- インスタンスを [実行中] 状態にする必要があります。インスタンスが開始したばかりの場合、SSH 接続の準備が整うまで数分かかります。
- インスタンスファイアウォールのポート 22、80、443 では、設定の実行中に任意の IP アドレスからの TCP 接続を許可する必要があります。詳細については、「インスタンスのファイアウォール」を参照してください。
- 頂点ドメイン (`example.com`) とその `www` サブドメイン (`www.example.com`) からのトラフィックをポイントする DNS レコードを追加または更新する場合、それらはインターネット全体に伝播する必要があります。「nslookup」や「MxToolbox」の「DNS Lookup」などのツールを使用して、DNS の変更が有効になったことを確認できます。
- 2023 年 1 月 1 日より前に作成された Wordpress インスタンスには、ウェブサイトのセットアップが失敗する、廃止された Certbot Personal Package Archive (PPA) リポジトリが含まれている場合があります。セットアップ中にこのリポジトリが存在する場合、既存のパスから削除され、インスタンス上の次の場所にバックアップされます: `~/opt/bitnami/lightsail/repo.backup`。廃止された PPA の詳細については、正規ウェブサイトの「Certbot PPA」を参照してください。
- Let's Encrypt 証明書は 60 ~ 90 日ごとに自動的に更新されます。
- 設定が進行中の場合、インスタンスを停止したり変更を加えたりしないでください。インスタンスの設定には最大 15 分かかる場合があります。インスタンス接続タブで、各ステップの進行状況を表示できます。

ウェブサイト設定ウィザードを使用してインスタンスを設定するには

1. インスタンス管理ページの [Connect] タブで、[ウェブサイトの設定] を選択します。



2. [ドメイン名を指定する] には、既存の Lightsail マネージドドメインを使用するか、新しいドメインを Lightsail に登録するか、別のドメインレジストラを使用して登録したドメインを使用します。[このドメインを使用] を選択して次のステップに進みます。
3. [接続] で、次のいずれかを実行します:
   - [Lightsail マネージドドメイン] を選択して Lightsail DNS ゾーンを使用します。[DNS ゾーンを使用する] を選択して、次の手順に進みます。
   - [サードパーティードメイン] を選択して、ドメインの DNS レコードを管理するホスティングサービスを使用します。後で使用する場合に備えて、Lightsail アカウントに一致する DNS ゾーンを作成することに注意してください。[サードパーティーの DNS を使用する] を選択して次の手順に進みます。
4. [静的 IP アドレスを作成する] には、静的 IP アドレスの名前を入力し、[静的 IP を作成する] を選択します。
5. [ドメイン割り当ての管理] では、[割り当ての追加] を選択し、ドメインタイプを選択し、[追加] を選択します。[続行] を選択して、次の手順に進みます。
6. [SSL/TLS 証明書の作成] では、ドメインおよびサブドメインを指定してメールアドレスを入力します。その後、[Lightsail がインスタンスで Let's Encrypt 証明書を構成することを許可する] を選択して、[証明書の作成] をクリックします。Lightsail リソースの構成を開始します。

   設定が進行中の場合、インスタンスを停止したり変更を加えたりしないでください。インスタンスの設定には最大 15 分かかる場合があります。インスタンス接続タブで、各ステップの進行状況を表示できます。
7. ウェブサイトのセットアップが完了したら、ドメイン割り当てステップで指定した URL が WordPress サイトを開いていることを確認します。

オプション: 個別タスク

個々のタスクを完了することでインスタンスを設定するには

1. 静的 IP アドレスの作成

   インスタンス管理ページで、[ネットワーク] タブの [静的 IP の作成] を選択します。静的 IP の場所とインスタンスが選択されます。静的 IP アドレスの名前を指定し、[作成とアタッチ] を選択します。

2. DNS ゾーンの作成

   ナビゲーションペインで、[ドメインと DNS] を選択します。[DNS ゾーンの作成] を選択し、ドメインを入力し、それから[DNS ゾーンの作成] を選択します。ウェブトラフィックが現在ドメインにルーティングされている場合は、ドメインの現在の DNS ホスティングプロバイダーのネームサーバーを変更する前に、既存の LightsailDNS レコードがすべて DNS ゾーンに存在することを確認してください。これにより、Lightsail DNS ゾーンへの引き渡し後も、トラフィックは引き続き中断されることなく流れます。

3. ドメイン割り当ての管理

   DNS ゾーンページの [割り当て] タブで [割り当てを追加] を選択します。ドメインまたはサブドメインを選択し、インスタンスを選択し、静的 IP アドレスをアタッチしてから、[割り当て] を選択します。

   > **Tip**
   >
   > これらの変更内容がインターネットを通じて伝播されるまで待ってから、ドメインから WordPress インスタンスへのトラフィックのルーティングを開始します。

4. SSL/TLS 証明書を作成およびインストールする

   段階的な手順については、the section called “HTTPS を有効にする” を参照してください。

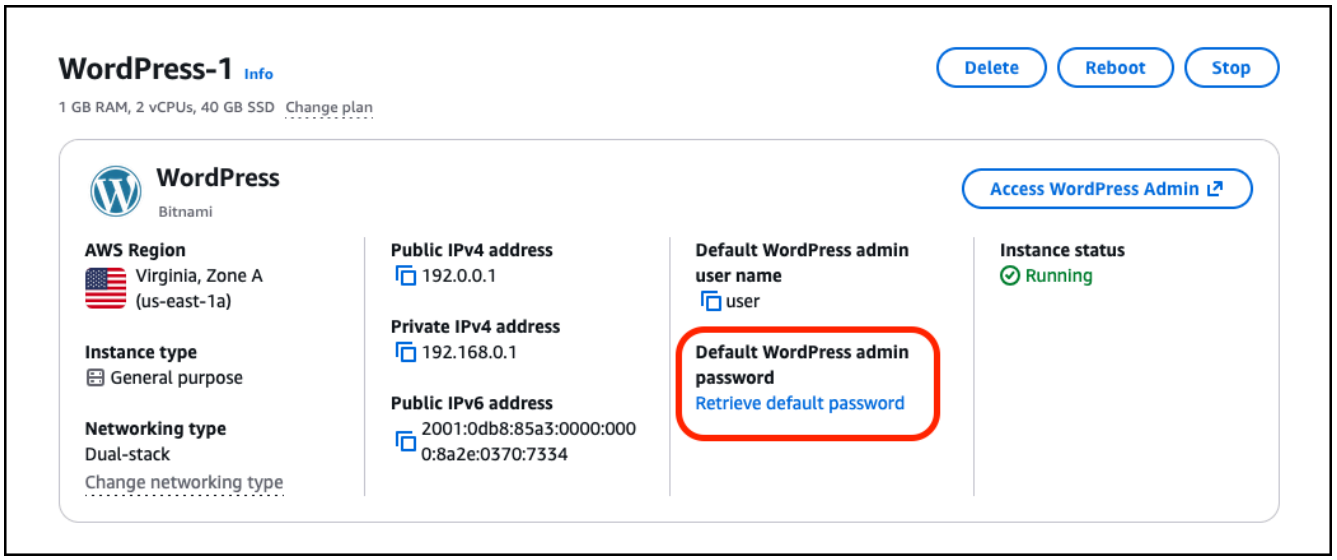
5. ドメイン割り当てステップで指定した URL で WordPress サイトが開いていることを確認します。

ステップ 4: WordPress ウェブサイトの管理者パスワードを取得する

WordPress ウェブサイトの管理ダッシュボードにサインインするためのデフォルトのパスワードは、インスタンスに保存されます。パスワードを取得するには、次の手順を実行します。

WordPress 管理者のデフォルトパスワードを取得するには

1. WordPress インスタンスのインスタンス管理ページを開きます。
2. [WordPress] パネルで、[デフォルトのパスワードの取得] を選択します。これにより、ページの下部にある [アクセスのデフォルトパスワード] が展開されます。



3. [CloudShell の起動] を選択します。そうすることで、ページの下部に分割されたペインが開きます。
4. [コピー] を選択し、コンテンツを CloudShell ウィンドウに貼り付けます。CloudShell プロンプトにカーソルを置いて Ctrl+V を押すか、右クリックしてメニューを開き、[貼り付ける] を選択します。
5. CloudShell ウィンドウに表示されるパスワードを書き留めます。後で WordPress ウェブサイトの管理ダッシュボードにサインインするには必要です。

```
[cloudshell-user@ip-[redacted] ~]$ AWS_REGION=us-east-1 ~/lightsail_connect WordPress-1 cat bitnami_application_password
JKzhKzhKzhR!
```

ステップ 5: WordPress ウェブサイトの管理ダッシュボードにサインインする

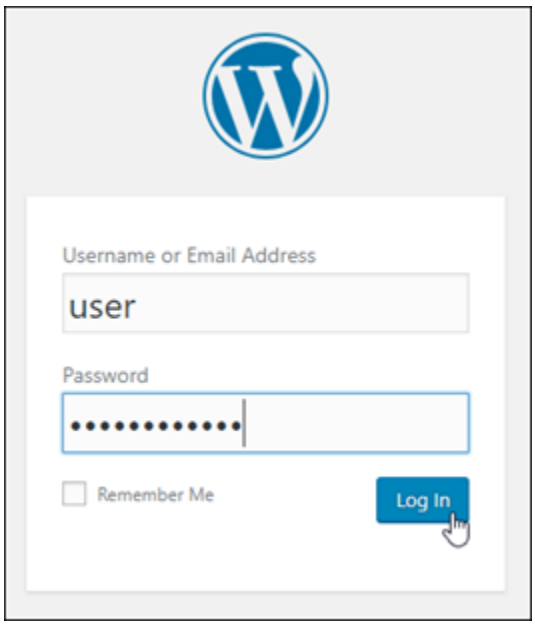
WordPress ウェブサイトの管理ダッシュボードのパスワードを取得したので、サインインできます。管理ダッシュボードでは、ユーザーパスワードの変更、プラグインのインストール、ウェブサイトのテーマの変更などを行うことができます。

WordPress ウェブサイトの管理ダッシュボードにサインインするには、以下の手順を実行します。

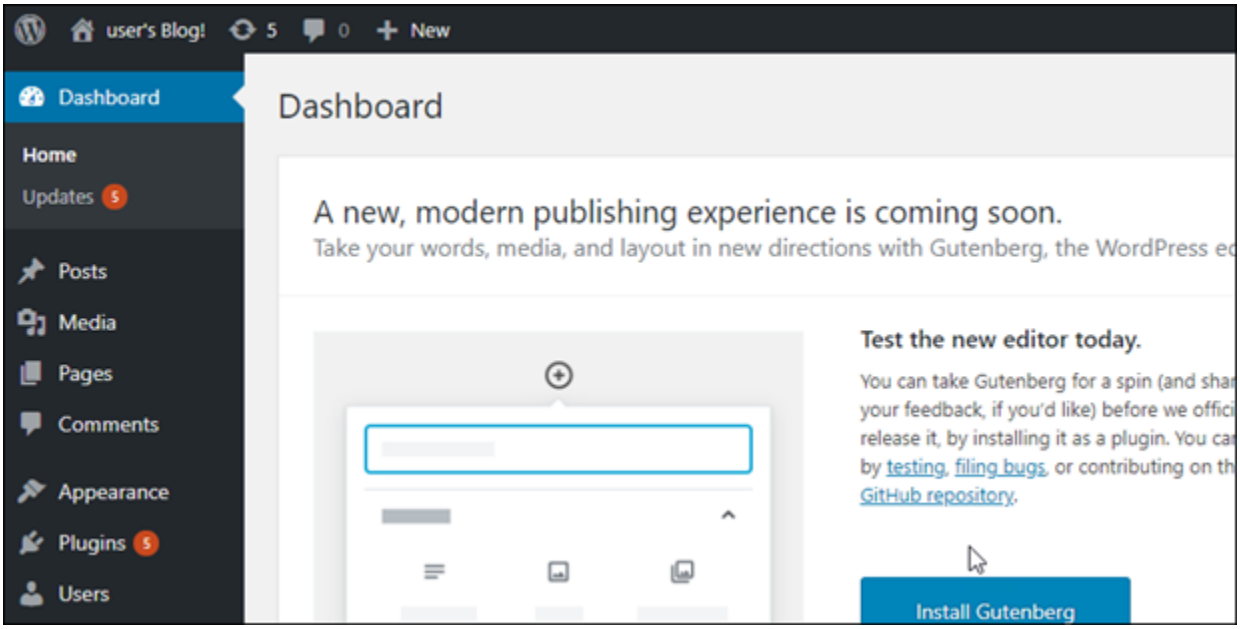
管理ダッシュボードにサインインするには

1. WordPress インスタンスのインスタンス管理ページを開きます。
2. [WordPress] パネルで、[WordPress 管理者へのアクセス] を選択します。
3. [WordPress 管理者ダッシュボードへのアクセス] パネルで、[パブリック IP アドレスを使用する] で、次のフォーマットを使用するリンクを選択します:

   http://*public-ipv4-address* /wp-admin
4. [ユーザー名またはメールアドレス] ボックスに **user** と入力します。
5. [パスワード] には、前の手順で取得したパスワードを入力します。
6. [ログイン] を選択します。



WordPress ウェブサイトの管理ダッシュボードにサインインしたら、管理アクションを実行できます。WordPress ウェブサイトの管理の詳細については、WordPress のドキュメントで「WordPress Codex」を参照してください。



Lightsail

ステップ 3: WordPress インスタンスを設定する

WordPress インスタンスは、ガイド付きのステップバイステップワークフローを使用して設定することも、個々のタスクを完了することもできます。いずれかのオプションを使用して、以下を設定します:

- 登録済みドメイン名 – WordPress サイトには、覚えやすいドメイン名が必要です。ユーザーは、WordPress サイトにアクセスするためにこのドメイン名を指定します。詳細については、「ドメインと DNS」を参照してください。
- DNS 管理 – ドメインの DNS レコードを管理する方法を決定する必要があります。DNS レコードは、DNS サーバーに対して、ドメインやサブドメインに対応する IP アドレスやホスト名を示します。DNS ゾーンには、ドメインの DNS レコードが含まれます。詳細については、「the section called “Lightsail での DNS”」を参照してください。
- 静的 IP アドレス – WordPress インスタンスのデフォルトのパブリック IP アドレスは、インスタンスを停止して開始すると変わります。インスタンスに静的 IP アドレスをアタッチすると、インスタンスを停止して開始してもそのアドレスが変わりません。詳細については、「the section called “IP アドレス”」を参照してください。
- SSL/TLS 証明書 – 検証済みの証明書を作成してインスタンスにインストールしたら、WordPress ウェブサイトの HTTPS を有効にして、登録済みドメインを介してインスタ

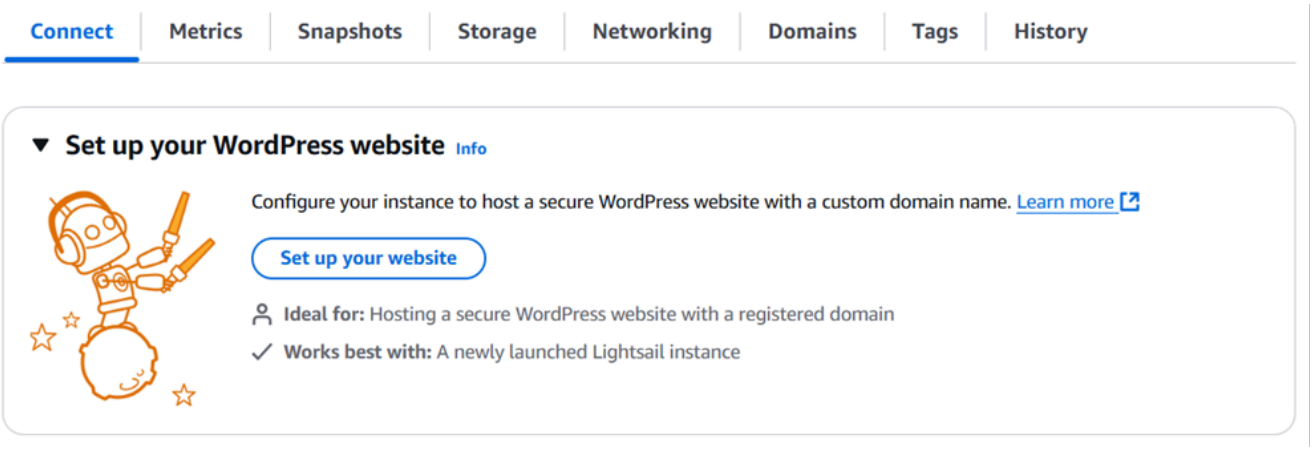
ンスにルーティングされるトラフィックを HTTPS を使用して暗号化できます。詳細については、「the section called “HTTPS を有効にする”」を参照してください。

オプション: ガイド付きのワークフロー

> **Tip**
>
> 開始する前に、以下のヒントを確認します。トラブルシューティングの詳細については、「Troubleshooting WordPress setup」を参照してください。
>
> - インスタンスを [実行中] 状態にする必要があります。インスタンスが開始したばかりの場合、SSH 接続の準備が整うまで数分かかります。
> - インスタンスファイアウォールのポート 22、80、443 では、設定の実行中に任意の IP アドレスからの TCP 接続を許可する必要があります。詳細については、「インスタンスのファイアウォール」を参照してください。
> - 頂点ドメイン (`example.com`) とその `www` サブドメイン (`www.example.com`) からのトラフィックをポイントする DNS レコードを追加または更新する場合、それらはインターネット全体に伝播する必要があります。「nslookup」や「MxToolbox」の「DNS Lookup」などのツールを使用して、DNS の変更が有効になったことを確認できます。
> - Let's Encrypt 証明書は 60 ~ 90 日ごとに自動的に更新されます。
> - 設定が進行中の場合、インスタンスを停止したり変更を加えたりしないでください。インスタンスの設定には最大 15 分かかる場合があります。インスタンス接続タブで、各ステップの進行状況を表示できます。

ウェブサイト設定ウィザードを使用してインスタンスを設定するには

1. インスタンス管理ページの [Connect] タブで、[ウェブサイトの設定] を選択します。



2. [ドメイン名を指定する] には、既存の Lightsail マネージドドメインを使用するか、新しいドメインを Lightsail に登録するか、別のドメインレジストラを使用して登録したドメインを使用します。[このドメインを使用] を選択して次のステップに進みます。
3. [接続] で、次のいずれかを実行します:
   - [Lightsail マネージドドメイン] を選択して Lightsail DNS ゾーンを使用します。[DNS ゾーンを使用する] を選択して、次の手順に進みます。
   - [サードパーティードメイン] を選択して、ドメインの DNS レコードを管理するホスティングサービスを使用します。後で使用する場合に備えて、Lightsail アカウントに一致する DNS ゾーンを作成することに注意してください。[サードパーティーの DNS を使用する] を選択して次の手順に進みます。
4. [静的 IP アドレスを作成する] には、静的 IP アドレスの名前を入力し、[静的 IP を作成する] を選択します。
5. [ドメイン割り当ての管理] では、[割り当ての追加] を選択し、ドメインタイプを選択し、[追加] を選択します。[続行] を選択して、次の手順に進みます。
6. [SSL/TLS 証明書の作成] では、ドメインおよびサブドメインを指定してメールアドレスを入力します。その後、[Lightsail がインスタンスで Let's Encrypt 証明書を構成することを許可する] を選択して、[証明書の作成] をクリックします。Lightsail リソースの構成を開始します。

   設定が進行中の場合、インスタンスを停止したり変更を加えたりしないでください。インスタンスの設定には最大 15 分かかる場合があります。インスタンス接続タブで、各ステップの進行状況を表示できます。
7. ウェブサイトのセットアップが完了したら、ドメイン割り当てステップで指定した URL が WordPress サイトを開いていることを確認します。

オプション: 個別タスク

個々のタスクを完了することでインスタンスを設定するには

1. 静的 IP アドレスの作成

   インスタンス管理ページで、[ネットワーク] タブの [静的 IP の作成] を選択します。静的 IP の場所とインスタンスが選択されます。静的 IP アドレスの名前を指定し、[作成とアタッチ] を選択します。

2. DNS ゾーンの作成

   ナビゲーションペインで、[ドメインと DNS] を選択します。[DNS ゾーンの作成] を選択し、ドメインを入力し、それから[DNS ゾーンの作成] を選択します。ウェブトラフィックが現在ドメインにルーティングされている場合は、ドメインの現在の DNS ホスティングプロバイダーのネームサーバーを変更する前に、既存の LightsailDNS レコードがすべて DNS ゾーンに存在することを確認してください。これにより、Lightsail DNS ゾーンへの引き渡し後も、トラフィックは引き続き中断されることなく流れます。

3. ドメイン割り当ての管理

   DNS ゾーンページの [割り当て] タブで [割り当てを追加] を選択します。ドメインまたはサブドメインを選択し、インスタンスを選択し、静的 IP アドレスをアタッチしてから、[割り当て] を選択します。

   > **Tip**
   >
   > これらの変更内容がインターネットを通じて伝播されるまで待ってから、ドメインから WordPress インスタンスへのトラフィックのルーティングを開始します。

4. SSL/TLS 証明書を作成およびインストールする

   段階的な手順については、the section called “HTTPS を有効にする” を参照してください。

5. ドメイン割り当てステップで指定した URL で WordPress サイトが開いていることを確認します。

ステップ 4: WordPress ウェブサイトの管理者パスワードを取得する

WordPress ウェブサイトの管理ダッシュボードにサインインするためのデフォルトのパスワードは、インスタンスに保存されます。パスワードを取得するには、次の手順を実行します。

WordPress 管理者のデフォルトパスワードを取得するには

1. WordPress インスタンスのインスタンス管理ページを開きます。
2. [WordPress] パネルで、[デフォルトのパスワードの取得] を選択します。これにより、ページの下部にある [アクセスのデフォルトパスワード] が展開されます。

3. [CloudShell の起動] を選択します。そうすることで、ページの下部に分割されたペインが開きます。
4. [コピー] を選択し、コンテンツを CloudShell ウィンドウに貼り付けます。CloudShell プロンプトにカーソルを置いて Ctrl+V を押すか、右クリックしてメニューを開き、[貼り付ける] を選択します。
5. CloudShell ウィンドウに表示されるパスワードを書き留めます。後で WordPress ウェブサイトの管理ダッシュボードにサインインするには必要です。

```
~ $ AWS_REGION=us-east-1 ~/lightsail_connect WordPress-1 tail -n 1 application_credentials
LnPLnPLnPLnPLnP9
```

ステップ 5: WordPress ウェブサイトの管理ダッシュボードにサインインする

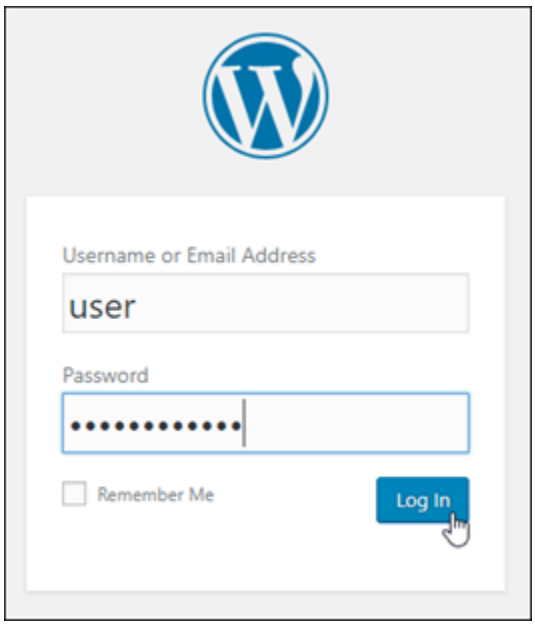
WordPress ウェブサイトの管理ダッシュボードのパスワードを取得したので、サインインできます。管理ダッシュボードでは、ユーザーパスワードの変更、プラグインのインストール、ウェブサイトのテーマの変更などを行うことができます。

WordPress ウェブサイトの管理ダッシュボードにサインインするには、以下の手順を実行します。

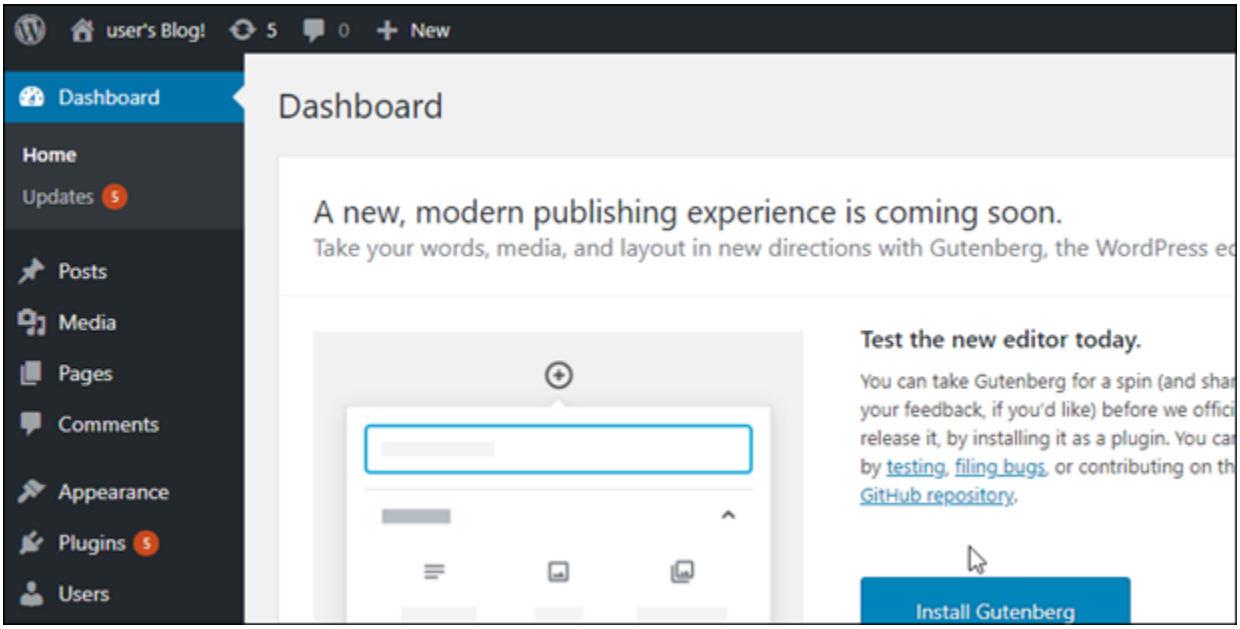
管理ダッシュボードにサインインするには

1. WordPress インスタンスのインスタンス管理ページを開きます。
2. [WordPress] パネルで、[WordPress 管理者へのアクセス] を選択します。
3. [WordPress 管理者ダッシュボードへのアクセス] パネルで、[パブリック IP アドレスを使用する] で、次のフォーマットを使用するリンクを選択します:

   http://*public-ipv4-address* /wp-admin
4. [ユーザー名またはメールアドレス] ボックスに **user** と入力します。
5. [パスワード] には、前の手順で取得したパスワードを入力します。
6. [ログイン] を選択します。



WordPress ウェブサイトの管理ダッシュボードにサインインしたら、管理アクションを実行できます。WordPress ウェブサイトの管理の詳細については、WordPress のドキュメントで「WordPress Codex」を参照してください。



追加情報

Amazon Lightsail で WordPress インスタンスを起動した後で、以下の追加のステップを実行できます:

- the section called “CDN を設定する”
- Linux または Unix インスタンスのスナップショットを作成する
- インスタンスまたはディスクの自動スナップショットの有効化または無効化
- 追加のブロックストレージディスクを作成して Linux ベースの インスタンスにアタッチする

WP Offload Media を使用して Lightsail の WordPress ウェブサイトを Amazon S3 に接続する

このチュートリアルでは、Amazon Lightsail インスタンスで実行されている WordPress ウェブサイトを Amazon Simple Storage Service (Amazon S3) に接続して、ウェブサイトの画像とアタッチメントを保存するのに必要なステップについて説明します。そのためには、Amazon Web Services (AWS) アカウントの認証情報のセットを使用して、WordPress プラグインを設定します。これで、プラグインによって Amazon S3 バケットが作成され、インスタンスのディスクの代わりに、バケットをウェブサイトの画像とアタッチメントに使用するようにウェブサイトが設定されます。

トピック

- ステップ 1: 前提条件を満たす
- ステップ 2: WordPress ウェブサイトに WP Offload Media プラグインをインストールする
- ステップ 3: IAM ポリシーを作成する
- ステップ 4: IAM ユーザーを作成する
- ステップ 5: IAM ユーザーのアクセスキーを作成する
- ステップ 6: WordPress 設定ファイルを編集する
- ステップ 7: WP Offload Media プラグインを使用して Amazon S3 バケットを作成する
- ステップ 8: 次のステップ

ステップ 1: 前提条件を満たす

開始する前に、Lightsail に WordPress インスタンスを作成し、実行状態になっていることを確認します。詳細については、「チュートリアル: WordPress インスタンスを起動して設定する」を参照してください。

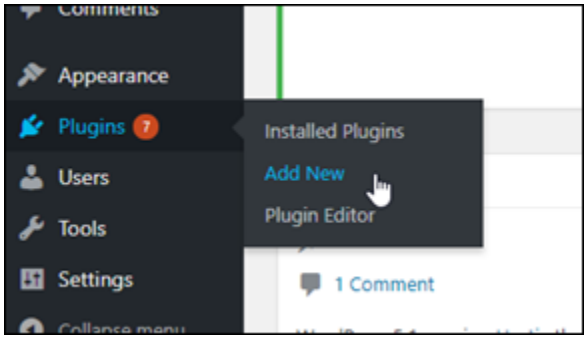
ステップ 2: WordPress ウェブサイトに WP Offload Media プラグインをインストールする

プラグインを使用して、Amazon S3 バケットを使用するようにウェブサイトを設定する必要があります。設定するために利用できるプラグインは多数あります。そのようなプラグインのひとつに WP Offload Media Lite があります。

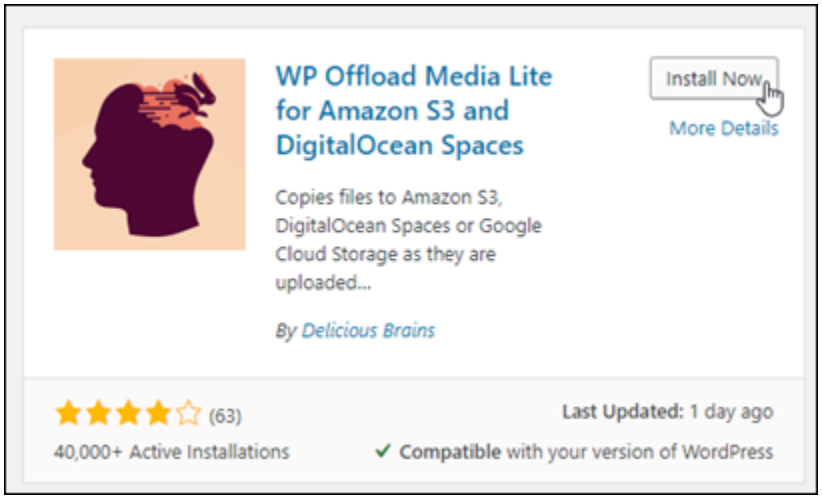
WordPress ウェブサイトに WP Offload Media プラグインをインストールする方法

1. 管理者として WordPress のダッシュボードにサインインします。

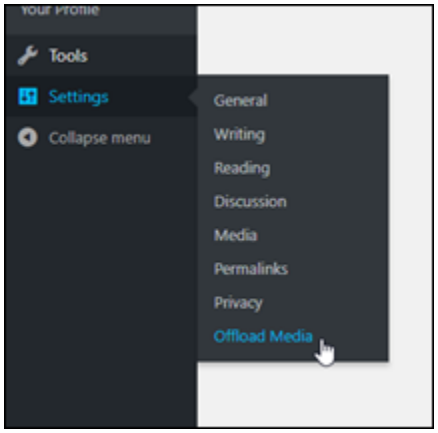
   詳細については、「Amazon Lightsailの Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。
2. 左側のナビゲーションメニューの [プラグイン] にカーソルを合わせ、[Add New (新規追加)] を選択します。



3. [WP Offload Media Lite] を検索します。
4. 検索結果の中から WP Offload Media プラグインの横の [Install Now] (今すぐインストール)を選択します。



5. プラグインのインストールが完了したら、[アクティベート] を選択します。
6. 左ナビゲーションメニューで、[設定]、[Offload Media] の順に選択します。



7. [オフロードメディア] ページで、ストレージプロバイダーとして [Amazon S3] を選択し、[wp-config.php でアクセスキーを定義する] を選択します。

このオプションでは、 AWS アカウント認証情報をインスタンスwp-config.phpの に追加する必要があります。これらのステップについては、このチュートリアルの後半で説明します。

[Offload Media] ページは開いたままにします。このチュートリアルの後半で使用します。このチュートリアルの ステップ 3: IAM ポリシーを作成する セクションに進みます。

ステップ 3: IAM ポリシーを作成する

> **Warning**
>
> このシナリオでは、プログラムによるアクセスと長期的な認証情報を持つ IAM ユーザーが必要です。これはセキュリティ上のリスクをもたらします。このリスクを軽減するために、これらのユーザーにはタスクの実行に必要な権限のみを付与し、不要になったユーザーは削除することをお勧めします。アクセスキーは、必要に応じて更新できます。詳細については、「IAM ユーザーガイド」の「アクセスキーの更新」を参照してください。

WP Offload Media プラグインでは、Amazon S3 バケットを作成し、ウェブサイトのイメージと添付ファイルをアップロードするために、 AWS アカウントにアクセスする必要があります。

WP Offload Media プラグインの新しい AWS Identity and Access Management (IAM) ポリシーを作成するには

1. 新しいブラウザタブを開き、IAM コンソールにサインインします。

2. 左のナビゲーションメニューの [アクセス管理] で、[ポリシー] を選択します。
3. [Create policy] (ポリシーの作成) を選択します。
4. [ポリシーの作成] ページで [JSON] を選択し、ポリシーエディタ内のコンテンツをすべて削除します。
5. ポリシーエディタで次のコンテンツを指定し、*amzn-s3-demo-bucket* のサンプルバケット名を独自のバケット名に置き換えます。

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*",
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        }
    ]
}
```

6. [次へ] を選択します。
7. [Policy Name] にこのポリシーの名前を入力してください。

   > **Tip**
   >
   > **wp_s3_user_policy** や **wp_offload_media_plugin_user_policy** など、分かりやすい名前を入力して、将来メンテナンスを実行するときに識別しやすくします。

8. [Create policy] (ポリシーの作成) を選択します。

   次のステップでは、IAM コンソールを開いたままにします。

ステップ 4: IAM ユーザーを作成する

新しい IAM ユーザーを作成し、以前に作成したポリシーをアタッチして、WP Offload Media プラグインの使用に必要なアクセス許可を付与します。

WP Offload Media プラグインの新しい AWS Identity and Access Management (IAM) ユーザーを作成するには

1. 必要に応じて、IAM コンソールを開きます。
2. ナビゲーションメニューの [アクセス管理] で、[ユーザー] を選択します。
3. [ユーザーの作成] を選択します。
4. [ユーザー名] には、新規ユーザーの名前を入力して [次へ] を選択します。

   > **Tip**
   >
   > **wp_s3_user** や **wp_offload_media_plugin_user** など、分かりやすい名前を入力して、将来メンテナンスを実行するときに識別しやすくします。

5. [ポリシーを直接アタッチ] を選択します。
6. [アクセス許可ポリシー] で、以前に作成したポリシーの名前を検索バーに入力します。
7. ポリシーを選択したら、[次へ] を選択します。
8. [ユーザーの作成] を選択します。

   次のステップでは、IAM コンソールを開いたままにします。

ステップ 5: IAM ユーザーのアクセスキーを作成する

WP Offload Media プラグインで使用される IAM ユーザーのアクセスキーを作成します。

WP Offload Media プラグインの新しい AWS Identity and Access Management (IAM) ユーザーを作成するには

1. 必要に応じて、IAM コンソールを開きます。
2. ナビゲーションメニューの [アクセス管理] で、[ユーザー] を選択します。
3. ユーザー詳細ページを開くには、ユーザー名を選択します。
4. [セキュリティ認証情報] タブの [アクセスキー] セクションで、[アクセスキーの作成] を選択します。
5. [その他] を選択したら、[次へ] を選択します。
6. [アクセスキーを作成] を選択します。
7. IAM ユーザー用に [アクセスキー ID] および[シークレットアクセスキー] を書き留めます。[.csv のダウンロード] を選択して、これらの値のコピーをローカルドライブに保存することもできま

す。これらの値は、WordPress インスタンスで wp-config.php ファイルを編集する際、次のいくつかの手順で必要になります。

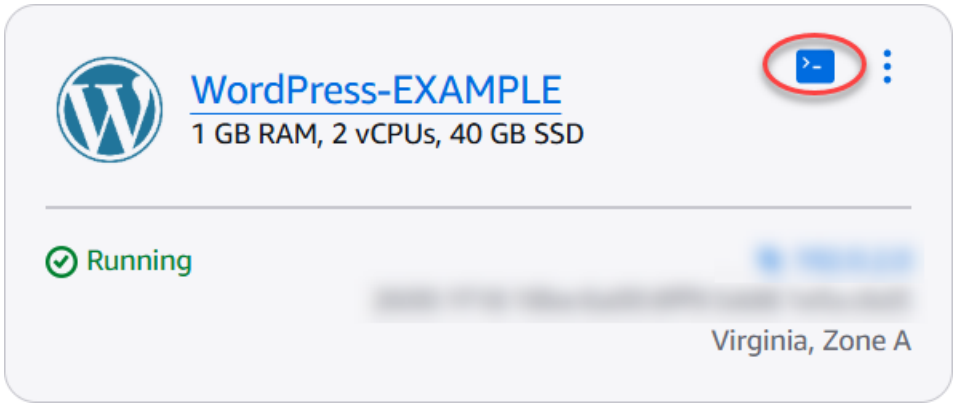
これで IAM コンソールを閉じて、Lightsail コンソールで次のステップを続行できます。

ステップ 6: WordPress 設定ファイルを編集する

wp-config.php ファイルには、データベース接続情報など、ウェブサイトの基本設定の詳細が含まれています。

WordPress インスタンスで **wp-config.php** ファイルを編集する方法

1. Lightsail コンソール にサインインします。
2. WordPress インスタンスのブラウザベースの SSH クライアントアイコンを選択します。



> **Note**
>
> 独自の SSH クライアントを使用してインスタンスに接続することもできます。詳細については、「PuTTY をダウンロードし、Lightsail で SSH を使用して接続するようにセットアップする」を参照してください。

3. 表示される SSH クライアントウィンドウで、次のコマンドを入力して、問題が発生した場合に備えて wp-config.php ファイルのバックアップを作成します。

```
sudo cp /opt/bitnami/wordpress/wp-config.php /opt/bitnami/wordpress/wp-config.php.backup
```

4. 次のコマンドを入力して、テキストエディタ nano を使用し、wp-config.php ファイルを開きます。

```
nano /opt/bitnami/wordpress/wp-config.php
```

5. テキスト `/* That's all, stop editing! Happy blogging. */` の上に次のテキストを入力します。

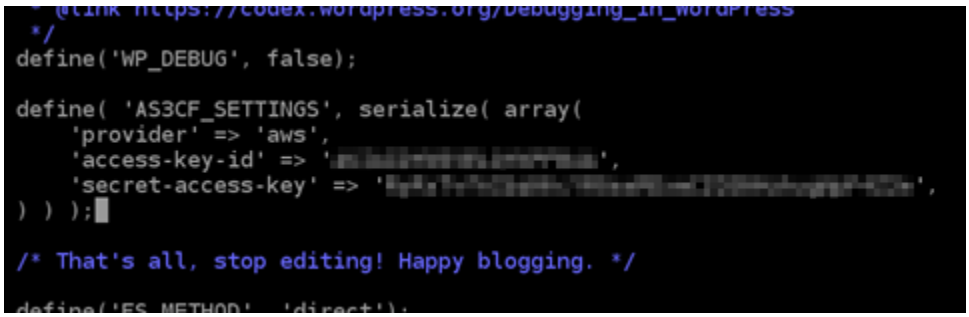
   AccessKeyID をアクセスキー ID に、*SecretAccessKey* をこれらのステップの前半で作成した IAM ユーザーのシークレットアクセスキーに置き換えます。

   ```
   define( 'AS3CF_SETTINGS', serialize( array(
       'provider' => 'aws',
       'access-key-id' => 'AccessKeyID',
       'secret-access-key' => 'SecretAccessKey',
   ) ) );
   ```

   例:

   ```
   define( 'AS3CF_SETTINGS', serialize( array(
       'provider' => 'aws',
       'access-key-id' => 'AKIAIOSFODNN7EXAMPLE',
       'secret-access-key' => 'wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY',
   ) ) );
   ```

   結果は次の例のようになります。

6. **Ctrl+X** を押して Nano を終了してから **Y**、**Enter** の順に押して編集内容を `wp-config.php` ファイルに保存します。
7. 次のコマンドを入力して、インスタンス上のサービスを再起動します。

   ```
   sudo /opt/bitnami/ctlscript.sh restart
   ```

   サービスが再起動されると次のような結果が表示されます。

```
bitnami@ip-172-26-13-236:~$ sudo /opt/bitnami/ctlscript.sh restart
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd stopped
/opt/bitnami/php/scripts/ctl.sh : php-fpm stopped
/opt/bitnami/mysql/scripts/ctl.sh : mysql stopped
/opt/bitnami/mysql/scripts/ctl.sh : mysql  started at port 3306
/opt/bitnami/php/scripts/ctl.sh : php-fpm started
Syntax OK
/opt/bitnami/apache2/scripts/ctl.sh : httpd started at port 80
bitnami@ip-172-26-13-236:~$
```

SSH ウィンドウを閉じ、このチュートリアルの前半で開いたままにした [Offload Media] ページに戻ります。これで、WP Offload Media プラグインを使用して Amazon S3 バケットを作成する準備ができました。

ステップ 7: WP Offload Media プラグインを使用して Amazon S3 バケットを作成する

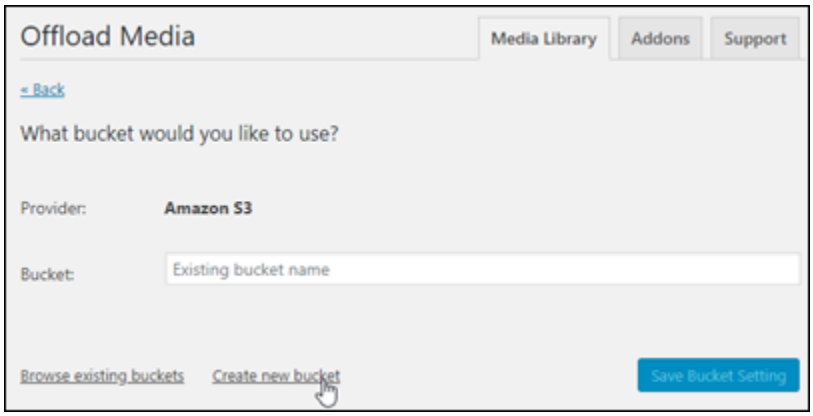
AWS 認証情報で `wp-config.php` ファイルが設定されたので、[Offload Media] ページに戻ってプロセスを完了します。

WP Offload Media プラグインを使用して Amazon S3 バケットを作成する方法

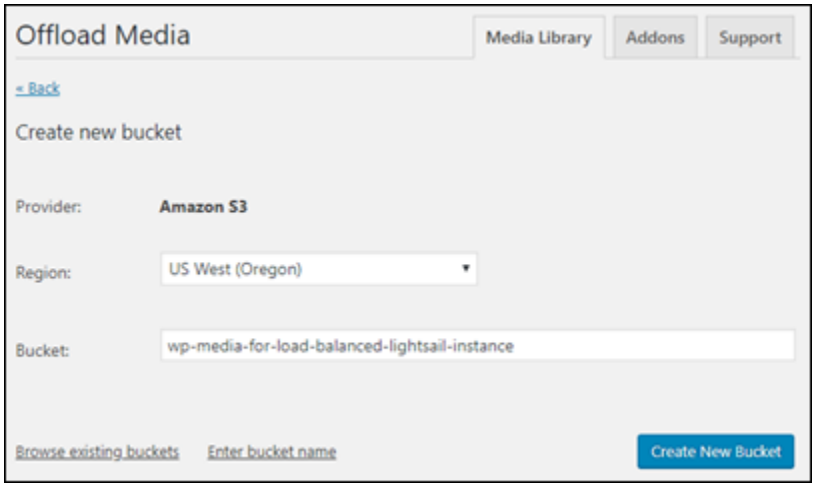
1. [Offload Media] ページを更新するか、[Nesxt] を選択します。

   Amazon S3 プロバイダーが設定されていることがわかります。

2. [新しいバケットの作成] を選択します。

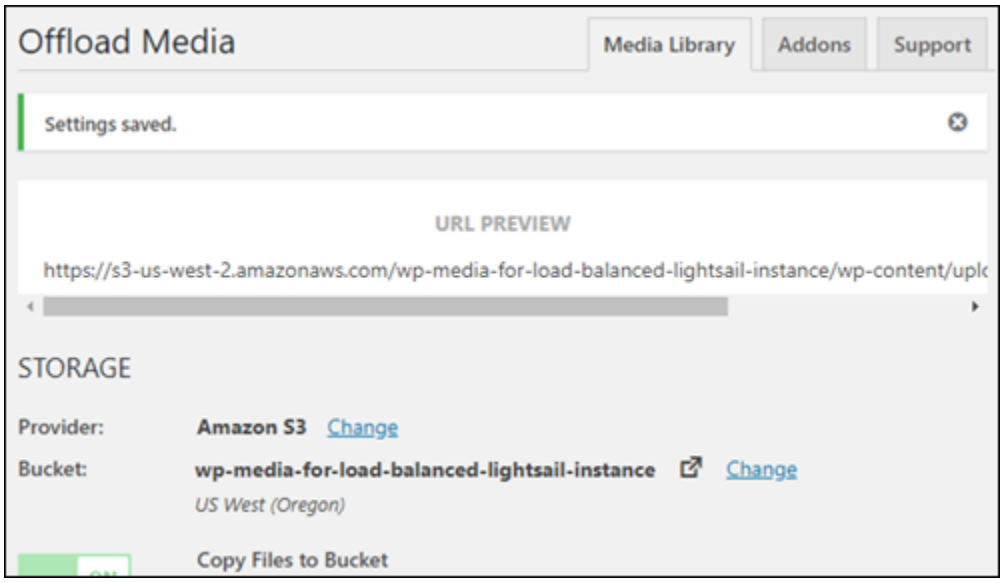


3. [リージョン] ドロップダウンメニューで、目的の AWS リージョンを選択します。WordPress インスタンスがあるリージョンと同じリージョンを選択することをお勧めします。
4. [バケット] テキストボックスに、新しい S3 バケットの名前を入力します。



5. [新しいバケットの作成] を選択します。

   ページが更新され、新しいバケットが作成されたことを確認します。表示される設定を確認し、WordPress ウェブサイトの動作に合わせて調整します。



   今後、ブログ投稿に追加された画像やアタッチメントは、作成した Amazon S3 バケットに自動的にアップロードされます。

ステップ 8: 次のステップ

WordPress ウェブサイトを Amazon S3 バケットに接続したら、WordPress インスタンスのスナップショットを作成して、行った変更をバックアップする必要があります。詳細については、「Linux または Unix インスタンスのスナップショットを作成する」を参照してください。

Lightsail WordPress インスタンスを Amazon Aurora データベースに接続する

投稿、ページ、およびユーザーのウェブサイトデータは、Amazon Lightsail の WordPress インスタンス上で実行するデータベースに保存されています。WordPress インスタンスに障害が発生した場合、データが回復不可能になる場合があります。このシナリオを回避するには、Amazon Relational Database Service (Amazon RDS) の Amazon Aurora データベースにウェブサイトのデータを転送する必要があります。

Amazon Aurora はクラウド用に構築された MySQL と PostgreSQL 互換のリレーショナルデータベースです。これは従来のエンタープライズデータベースのパフォーマンスと可用性に、オープンソースデータベースのシンプルさと費用対効果を組み合わせています。Aurora は Amazon RDS の一部として提供されています。Amazon RDS は、クラウドでリレーショナルデータベースを簡単に設定、運用、およびスケールすることができるマネージドデータベースサービスです。詳細については、「Amazon Relational Database Service ユーザーガイド」と「Aurora の Amazon Aurora ユーザーガイド」を参照してください。

このチュートリアルでは、Lightsail 内の WordPress インスタンスからウェブサイトデータベースを Amazon RDS 内の Aurora マネージドデータベースに接続する方法について説明します。

目次

- ステップ 1: 前提条件を満たす
- ステップ 2: Aurora データベースのセキュリティグループを設定する
- ステップ 3: Lightsail インスタンスから Aurora データベースに接続する
- ステップ 4: MySQL データベースを WordPress インスタンスから Aurora データベースに転送する
- ステップ 5: Aurora マネージドデータベースに WordPress を接続するように設定する

ステップ 1: 前提条件を満たす

開始する前に次の前提条件を完了します。

1. Lightsail で WordPress インスタンスを作成し、アプリケーションを設定します。続行する前に、インスタンスは実行中状態になっていることを確認してください。詳細については、「チュートリアル: Amazon Lightsail で WordPress インスタンスを起動して設定する」を参照してください。

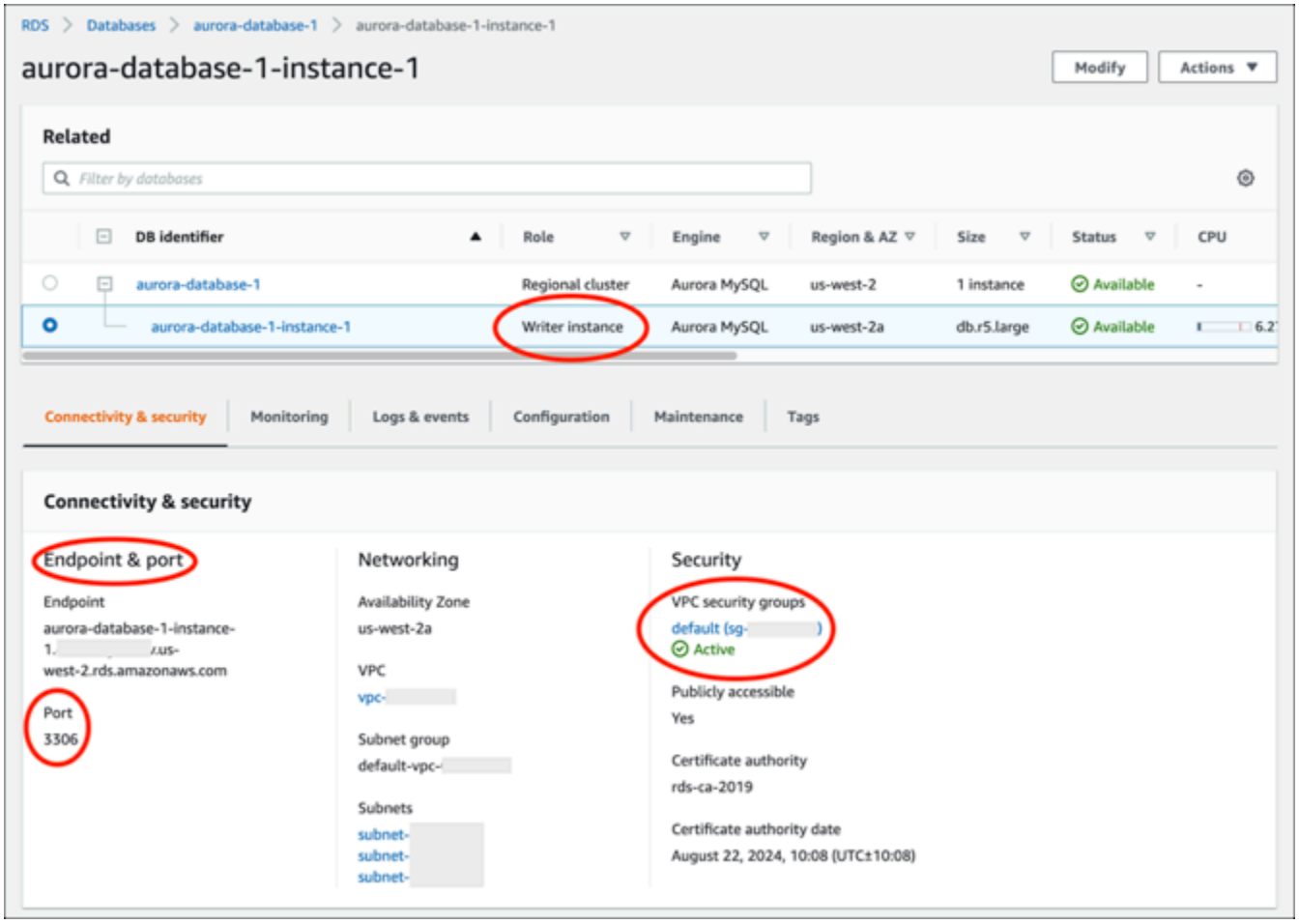
2. Lightsail アカウントで VPC ピアリングを有効にします。詳細については、「 以外の AWS リソースを操作するようにピア接続を設定するLightsail」を参照してください。
3. Amazon RDS に Aurora マネージドデータベースを作成します。データベースは、WordPress リソースと同じ AWS リージョン にある必要があります。続行する前に、データベースが実行中状態になっていることを確認してください。詳細については、「Amazon Aurora ユーザーガイド」の「Amazon Aurora で使用開始」を参照してください。

ステップ 2: Aurora データベースのセキュリティグループを設定する

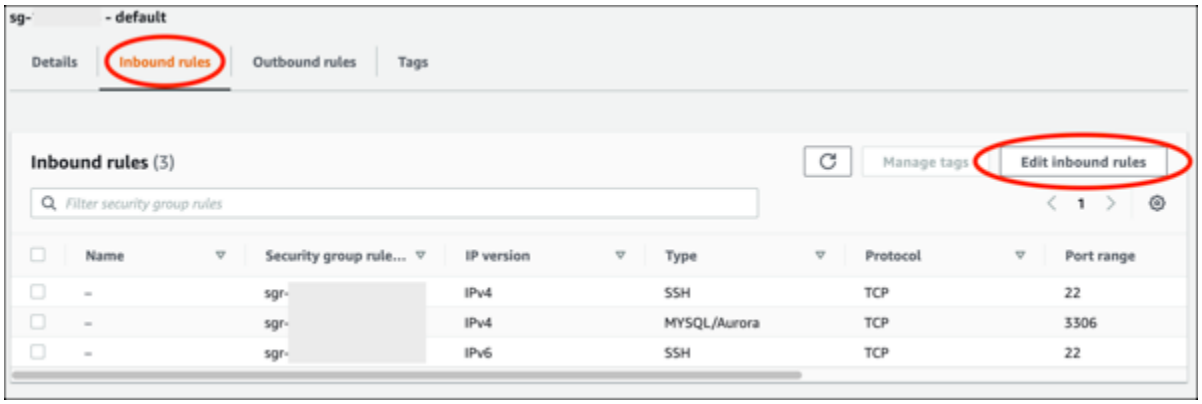
AWS セキュリティグループは、 AWS リソースの仮想ファイアウォールとして機能します。Amazon RDS 内の Aurora データベースに接続できる送受信トラフィックを制御します。詳細については、「Amazon Virtual Private Cloud ユーザーガイド」の「セキュリティグループを使用してリソースへのトラフィックを制御する」を参照してください。

WordPress インスタンスが Aurora データベースへの接続を確立できるよう、以下の手順を完了してセキュリティグループを設定します。

1. Amazon RDS コンソールにサインインします。
2. ナビゲーションペインで、[Databases] (データベース) を選択します。
3. WordPress インスタンスが接続する Aurora データベースの[ライターインスタンス]を選択します。
4. [Connectivity & security (接続とセキュリティ)] タブを選択します。
5. [Endpoint & port] (エンドポイントとポート) セクションに表示されるライターインスタンスのエンドポイント名とポートを記録します。これらの情報は、データベースに接続する Lightsail インスタンスを設定するときに必要になります。
6. [Security] (セキュリティ) セクションでアクティブな VPC セキュリティグループのリンクを選択します。データベースのセキュリティグループにリダイレクトされます。



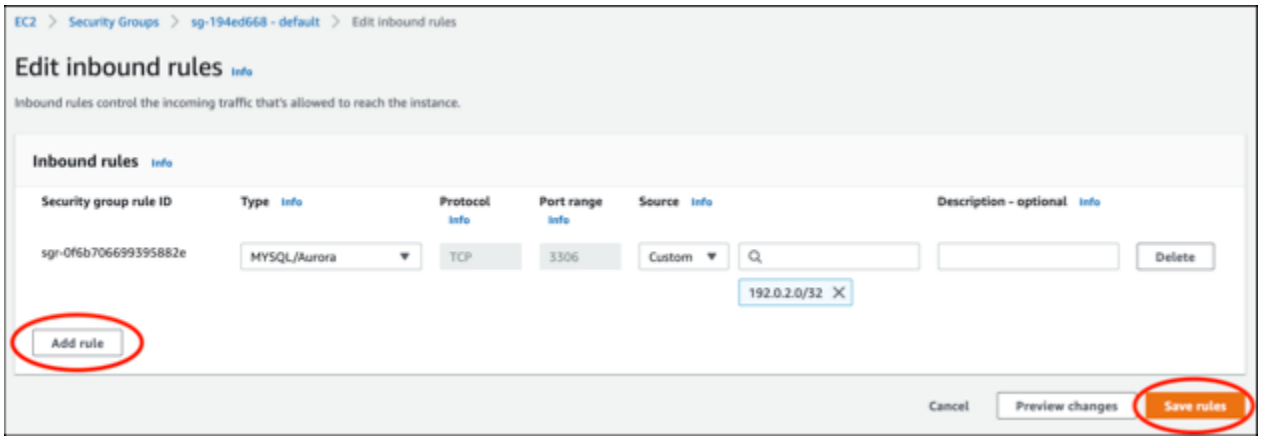
7. Aurora データベースのセキュリティグループが選択されていることを確認します。
8. [Inbound rules] (インバウンドルール) タブを開きます。
9. インバウンドのルールを編集 を選択します。



10. [Edit inbound rules] (インバウンドルールの編集) ページで [Add rule] (ルールの追加) を選択します。
11. 次のいずれかのステップを完了します。

- デフォルトの MySQL ポート 3306 を使用する場合は、[Type] (タイプ) ドロップダウンメニューから [MySQL/Aurora] を選択します。
- データベースのカスタムポートを使用する場合は、[Type] (タイプ) ドロップダウンメニューから [Custom TCP] (カスタム TCP) を選択し、[Port Range] (ポート範囲) テキストボックスにポート番号を入力します。

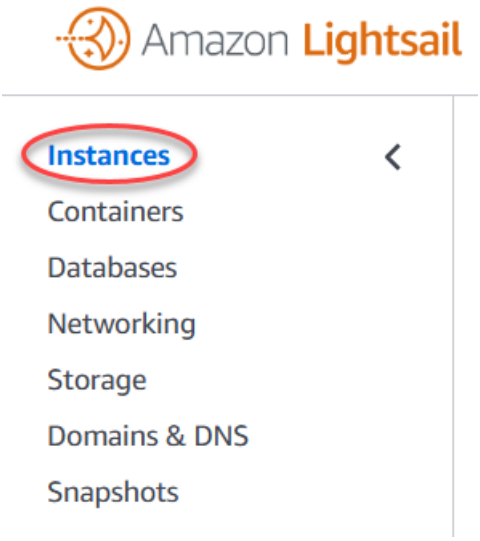
12. [Source] (ソース) テキストボックスに WordPress インスタンスのプライベート IP アドレスを追加します。IP アドレスは、 CIDR 表記で入力する必要があります (/32 を追加する必要があります)。例えば、`192.0.2.0` を許可するには「`192.0.2.0/32`」と入力します。
13. [ルールの保存] を選択します。



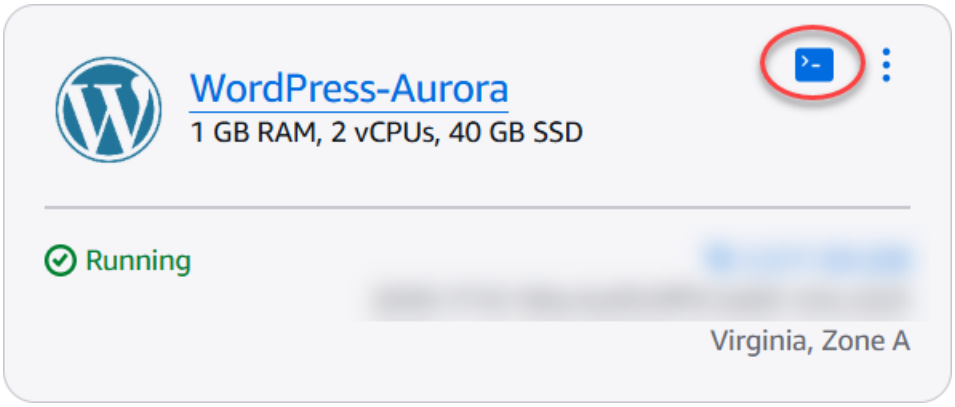
ステップ 3: Lightsail インスタンスから Aurora データベースに接続する

以下の手順を完了して、Lightsail インスタンスから Aurora データベースに接続できることを確認します。

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [インスタンス] を選択してください。



3. SSH を使用して接続する WordPress インスタンスのブラウザベースの SSH クライアントアイコンを選択します。



4. インスタンスに接続したら、次のコマンドを入力して、Aurora データベースに接続します。このコマンドで、*DatabaseEndpoint* を実際の Aurora データベースのエンドポイントアドレスで置き換え、*Port* をデータベースのポートで置き換えます。*MyUsername* は、データベースを作成したときに入力したユーザーの名前で置き換えます。

```
mysql -h DatabaseEndpoint -P Port -u MyUserName -p
```

インスタンスが Aurora データベースにアクセスおよび接続できれば、次の例のような応答が表示されます。

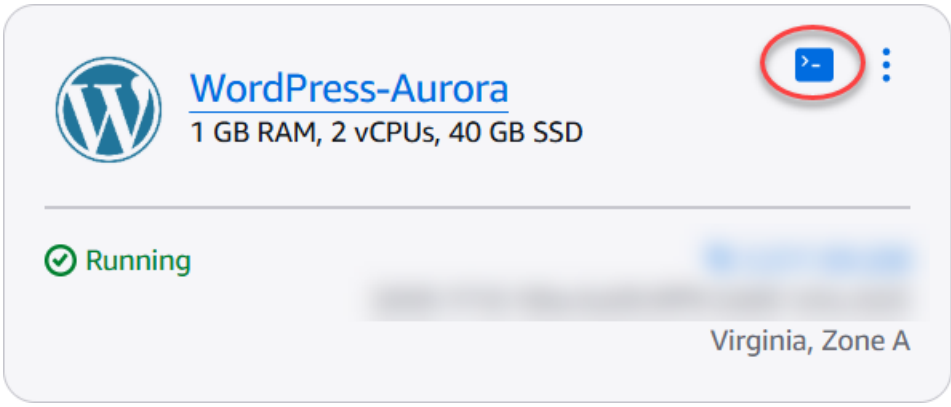
```
bitnami@ip-            $ mysql -h database.cluster-            .us-west-2.rds.amazonaws.
com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 215
Server version: 5.6.10 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

この応答が表示されない場合、またはエラーメッセージが表示された場合は、Aurora データベースのセキュリティグループを設定して、Lightsail インスタンスのプライベート IP アドレスからその接続を許可する必要がある場合があります。詳細については、このガイドの「Aurora データベースのセキュリティグループを設定する」を参照してください。

ステップ 4: データベースを WordPress インスタンスから Aurora データベースに転送する

インスタンスからデータベースに接続できることを確認した後は、WordPress ウェブサイトデータを Aurora データベースに転送する必要があります。

1. Lightsail コンソール にサインインします。
2. [Instances] (インスタンス) タブで、WordPress インスタンス用ブラウザベースの SSH クライアントを選択します。



3. ブラウザベースの SSH クライアントが WordPress インスタンスに接続されたら、以下のコマンドを入力します。このコマンドは、インスタンス上の `bitnami_wordpress` データベースのデータを転送し、Aurora データベースに移動します。このコマンドでは、*`DatabaseUserName`* は、Aurora データベースを作成したときに入力したプライマリユーザーの名前で置き換えます。*`DatabaseEndpoint`* は、Aurora データベースのエンドポイントアドレスに置き換えます。

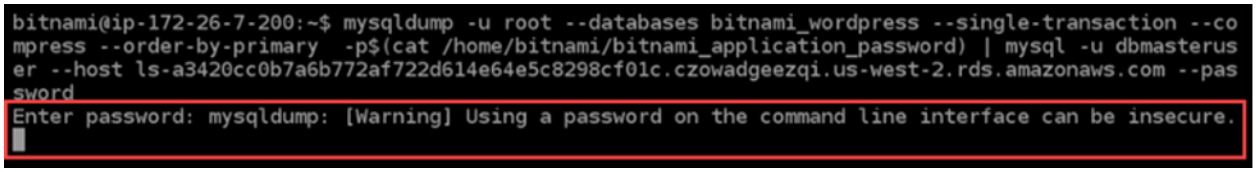
```
sudo mysqldump -u root --databases bitnami_wordpress --single-transaction --
compress --order-by-primary -p$(cat /home/bitnami/bitnami_application_password) |
 sudo mysql -u DatabaseUserName --host  DatabaseEndpoint --password
```

例

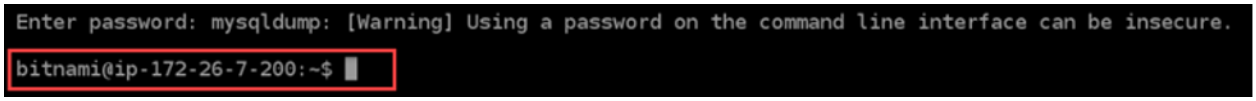
```
sudo mysqldump -u root --databases bitnami_wordpress --single-transaction --
compress --order-by-primary -p$(cat /home/bitnami/bitnami_application_password)
 | sudo mysql -u DBuser --host  abc123exampleE67890.czowadgeezqi.us-
west-2.rds.amazonaws.com --password
```

4. Enter password プロンプトで、Aurora データベースのパスワードを入力し、Enter キーを押します。

   入力中にパスワードを表示することはできません。



   データが正常に転送されると、次の例のような応答が表示されます。



   エラーが表示される場合は、正しいデータベース、ユーザー名、パスワード、およびエンドポイントが使用されていることを確認して、もう一度試してください。

ステップ 5: Aurora データベースに WordPress を接続するように設定する

アプリケーションデータを Aurora データベースに転送した後、WordPress を設定して接続する必要があります。ウェブサイトが Aurora データベースに接続されるように、以下の手順を実行して WordPress 設定ファイル (wp-config.php) を編集します。

1. WordPress インスタンスに接続されているブラウザベースの SSH クライアントで、以下のコマンドを入力し、wp-config.php ファイルのバックアップを作成します。

```
cp /opt/bitnami/wordpress/wp-config.php /opt/bitnami/wordpress/wp-config.php-backup
```

2. 次のコマンドを入力して wp-config.php ファイルを書き込み可能にします。

```
sudo chmod 664 /opt/bitnami/wordpress/wp-config.php
```

3. config ファイル内のデータベースユーザー名を Aurora データベースを作成したときに入力したプライマリユーザーの名前に編集します。

```
sudo wp config set DB_USER DatabaseUserName
```

4. config ファイル内のデータベースホストを Aurora データベースのエンドポイントアドレスとポート番号で編集します。例えば、abc123exampleE67890.czowadgeezqi.us-west-2.rds.amazonaws.com:3306。

```
sudo wp config set DB_HOST DatabaseEndpoint:Port
```

5. config ファイル内のデータベースパスワードを Aurora データベースのパスワードで編集します。

```
sudo wp config set DB_PASSWORD DatabasePassword
```

6. wp config list コマンドを入力して、wp-config.php ファイルに入力した情報が正しいことを確認します。

```
sudo wp config list
```

次の例のような結果で設定の詳細が表示されます。

```
bitnami@ip-1           :~$ sudo wp config list
+--------------------+-----------------------------------------------------+----------+
| name               | value                                               | type     |
+--------------------+-----------------------------------------------------+----------+
| table_prefix       | wp_                                                 | variable |
| DB_NAME            | bitnami_wordpress                                   | constant |
| DB_USER            | admin                                               | constant |
| DB_PASSWORD        | Password1                                           | constant |
| DB_HOST            | database.cluster-           .us-west-2.rds.amazonaws | constant |
|                    | .com:3306                                           |          |
```

7. 以下のコマンドを入力して、インスタンス上のウェブサービスを再起動します。

```
sudo /opt/bitnami/ctlscript.sh restart
```

サービスが再起動されると、次の例のような結果が表示されます。

お疲れ様でした。これで、WordPress サイトが Aurora データベースを使用できるように設定されました。

> **Note**
>
> 元の `wp-config.php` ファイルを復元する必要がある場合は、以下のコマンドを入力し、前にこのチュートリアルで作成したバックアップを使用して復元します。
>
> ```
> cp /opt/bitnami/wordpress/wp-config.php-backup /opt/bitnami/wordpress/wp-config.php
> ```

Lightsail の MySQL マネージド型データベースに WordPress データを転送する

投稿、ページ、およびユーザーの重要な WordPress ウェブサイトデータは、Amazon Lightsail のインスタンスで実行されている MySQL データベースに保存されています。WordPress インスタンスに障害が発生した場合、データが回復不可能になる場合があります。このシナリオを回避するには、MySQL マネージドデータベースにウェブサイトのデータを転送する必要があります。

このチュートリアルでは、WordPress ウェブサイトデータを Lightsail の MySQL マネージドデータベースに転送する方法について説明します。ウェブサイトがマネージドデータベースに接続され、インスタンスで実行されているデータベースへの接続を停止するために、インスタンス上の WordPress 設定 (`wp-config.php`) ファイルを編集する方法についても説明します。

目次

- ステップ 1: 前提条件を満たす
- ステップ 2: MySQL マネージドデータベースに WordPress データベースを転送する
- ステップ 3: MySQL マネージド型データベースに WordPress を接続するように設定する
- ステップ 4: 次のステップを完了する

ステップ 1: 前提条件を満たす

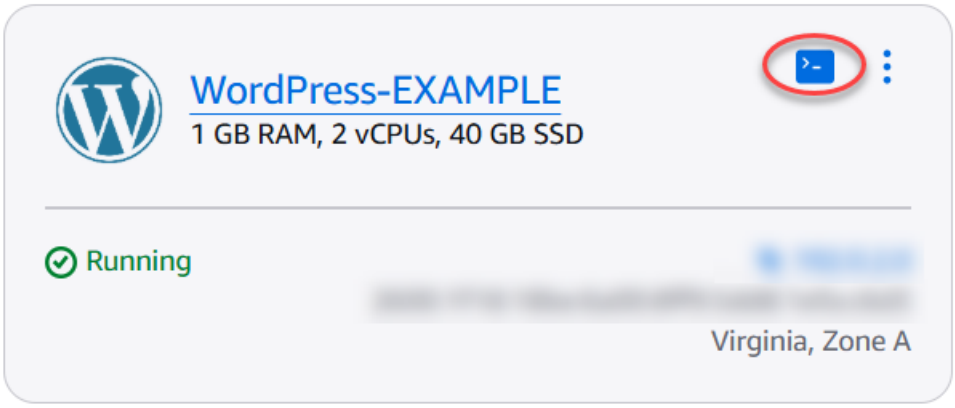
開始する前に、前提条件として以下の作業を実行します。

- Lightsail に WordPress インスタンスを作成し、実行状態になっていることを確認します。詳細については、「チュートリアル: Amazon Lightsail で WordPress インスタンスを起動して設定する」を参照してください。
- WordPress インスタンスと同じ AWS リージョン の Lightsail にMySQL マネージド型のデータベースを作成し、実行状態にあることを確認します。WordPress は、Lightsail で使用できるすべての MySQL データベースオプションで使用できます。詳細については、「Amazon Lightsail でデータベースを作成する」を参照してください。
- MySQL マネージドデータベースのパブリックおよびデータインポートモードを有効化します。このチュートリアルの手順を完了した後でこれらのモードを無効にできます。詳細については、「データベースのパブリックモードを設定する」および「データベースのデータインポートモードを設定する」を参照してください。

ステップ 2: MySQL マネージドデータベースに WordPress データベースを転送する

次の手順を実行して、WordPress ウェブサイトデータを、Lightsail の MySQL マネージドデータベースに転送します。

1. Lightsail コンソール にサインインします。
2. [Instances] (インスタンス) タブで、WordPress インスタンス用ブラウザベースの SSH クライアントアイコンを選択します。



3. ブラウザベースの SSH クライアントを WordPress インスタンスに接続後、以下のコマンドを入力して、`bitnami_wordpress` データベースのデータを MySQL マネージドデータベースに

転送します。*DbUserName* をマネージドデータベースのユーザー名に置き換え、*DbEndpoint* をマネージドデータベースのエンドポイントアドレスに置き換えてください。

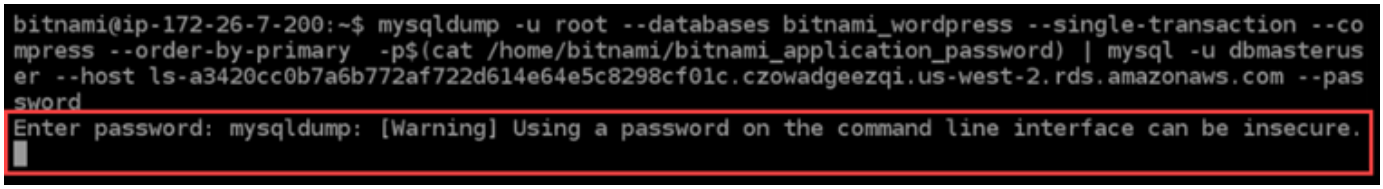
```
sudo mysqldump -u root --databases bitnami_wordpress --single-transaction --
compress --order-by-primary  -p$(cat /home/bitnami/bitnami_application_password) |
 sudo mysql -u DbUserName --host DbEndpoint --password
```

例

```
sudo mysqldump -u root --databases bitnami_wordpress --single-transaction --
compress --order-by-primary -p$(cat /home/bitnami/bitnami_application_password)
 | sudo mysql -u dbmasteruser --host ls-abc123exampleE67890.czowadgeezqi.us-
west-2.rds.amazonaws.com --password
```

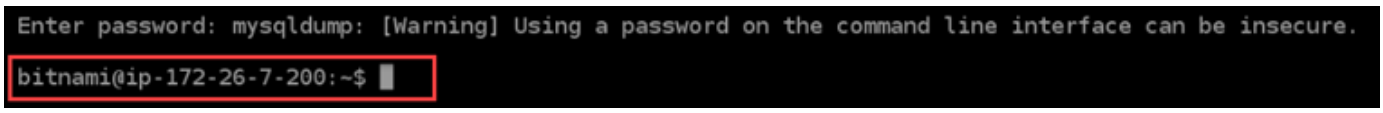
4. プロンプトで、MySQL マネージド型データベースのパスワードを入力し、Enter を押します。

   入力中にパスワードを表示することはできません。



5. データが正常に転送されると以下のような表示が出ます。

   エラーが表示される場合は、正しいデータベース、ユーザー名、パスワード、またはエンドポイントが使用されていることを確認して、もう一度試してください。



ステップ 3: MySQL マネージド型データベースに WordPress を接続するように設定する

ウェブサイトが MySQL マネージドデータベースに接続されるように、以下の手順を実行して WordPress 設定ファイル (`wp-config.php`) を編集します。

1. 問題が発生した場合は、WordPress インスタンスに接続されているブラウザベースの SSH クライアントで、以下のコマンドを入力し、`wp-config.php` ファイルのバックアップを作成します。

```
cp /opt/bitnami/wordpress/wp-config.php /opt/bitnami/wordpress/wp-config.php-backup
```

2. 以下のコマンドを入力して、Nano テキストエディタ を使用し、wp-config.php ファイルを開きます。

```
nano /opt/bitnami/wordpress/wp-config.php
```

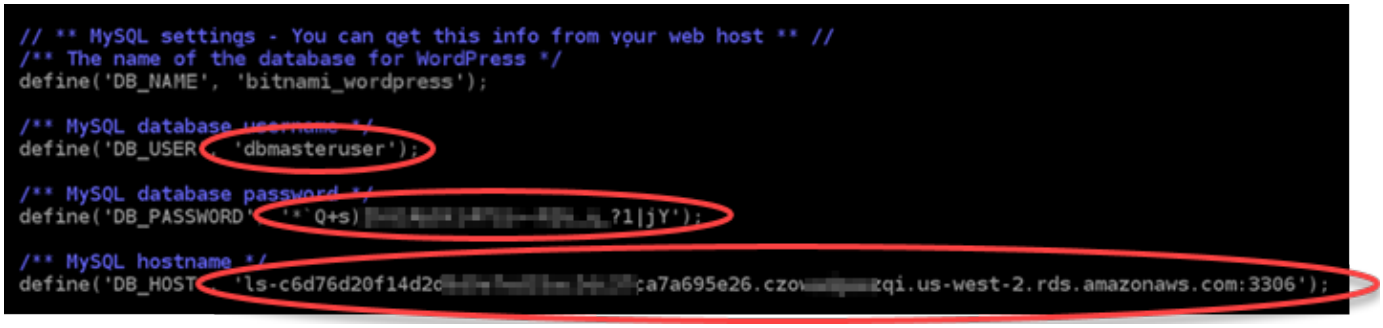
3. 以下の例のようにDB_USER、DB_PASSWORD、およびDB_HOST 値が見つかるまで下にスクロールします。

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'bitnami_wordpress');

/** MySQL database username */
define('DB_USER', 'bn_wordpress');

/** MySQL database password */
define('DB_PASSWORD', 'd6ab501583');

/** MySQL hostname */
define('DB_HOST', 'localhost:3306');
```

4. 次の値を変更します。

   - DB_USER — これを編集して MySQL マネージドデータベースのユーザー名に一致させます。Lightsail マネージドデータベースのデフォルトプライマリーユーザー名はdbmasteruser です。
   - DB_PASSWORD — これを編集して MySQL マネージドデータベースのパスワードに一致させます。詳細については、「データベースのパスワードを管理する」を参照してください。
   - DB_HOST — これを編集して MySQL マネージドデータベースのエンドポイントに一致させます。ホストアドレスの末尾に必ず :3306 ポート番号を入力します。例: ls-abc123exampleE67890.czowadgeezqi.us-west-2.rds.amazonaws.com:3306。

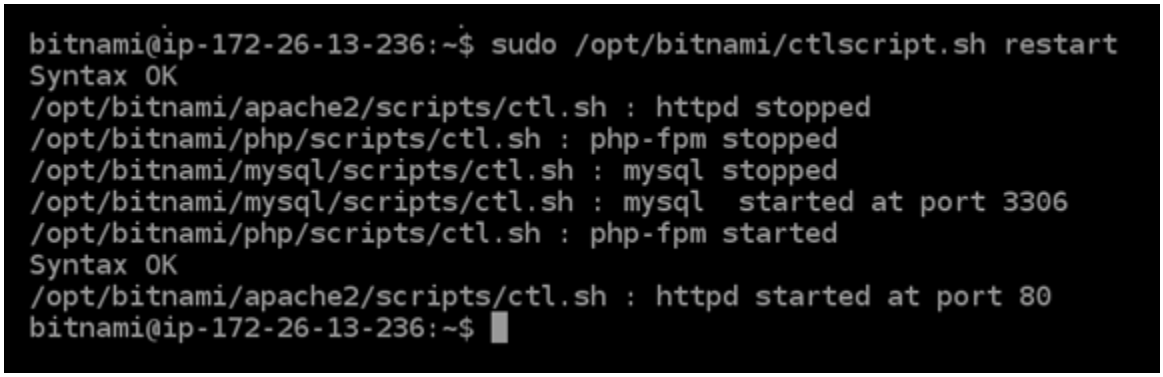
   結果は次の例のようになります。



5. Ctrl+X を押して Nano を終了し、Y および Enter を押して編集内容を保存します。
6. 以下のコマンドを入力して、インスタンス上のウェブサービスを再起動します。

```
sudo /opt/bitnami/ctlscript.sh restart
```

サービスが再起動されると以下のような結果が表示されます。



お疲れ様でした。これで、WordPress サイトは MySQL マネージド型データベースを使用できるように設定されました。

> **Note**
>
> 何らかの理由で、元の wp-config.php ファイルを復元する必要がある場合は、以下のコマンドを入力し、前にこのチュートリアルで作成したバックアップを使用して復元します。
>
> ```
> cp /opt/bitnami/wordpress/wp-config.php-backup /opt/bitnami/wordpress/wp-config.php
> ```

ステップ 4: 次のステップを完了する

WordPress ウェブサイトを MySQL マネージドデータベースに接続したら、以下の追加ステップを実行します。

- WordPress インスタンスのスナップショットを作成します。詳細については、「Linux または Unix インスタンスのスナップショットを作成する」を参照してください。
- MySQL マネージドデータベースのスナップショットを作成します。詳細については、「データベースのスナップショットを作成する」を参照してください。
- MySQL マネージドデータベースのパブリックおよびデータインポートモードを無効化します。詳細については、「データベースのパブリックモードを設定する」および「データベースのデータインポートモードを設定する」を参照してください。

WordPress インスタンスを静的コンテンツ用の Lightsail バケットに接続する

このチュートリアルでは、Amazon Lightsail インスタンスで実行されている WordPress ウェブサイトを Lightsail バケットに接続するために必要なステップを説明します。バケットを使用して、画像や添付ファイルなどの静的コンテンツをホストすることが可能です。これを実行するには、WordPress のウェブサイトに WP Offload Media Lite プラグインをインストールし、Lightsail バケットに接続するように設定する必要があります。プラグインが設定された後、WordPress ウェブサイトにアップロードされるすべてのメディアは、インスタンスのディスクではなく、バケットに自動的に追加されます。

目次

- ステップ 1: 前提条件を満たす
- ステップ 2: バケットのアクセス許可を変更する
- ステップ 3: WordPress ウェブサイトに WP Offload Media プラグインをインストールする
- ステップ 4: WordPress ウェブサイトと Lightsail バケットの接続をテストします

ステップ 1: 前提条件を満たす

以下の前提条件を完了します (まだの場合)。

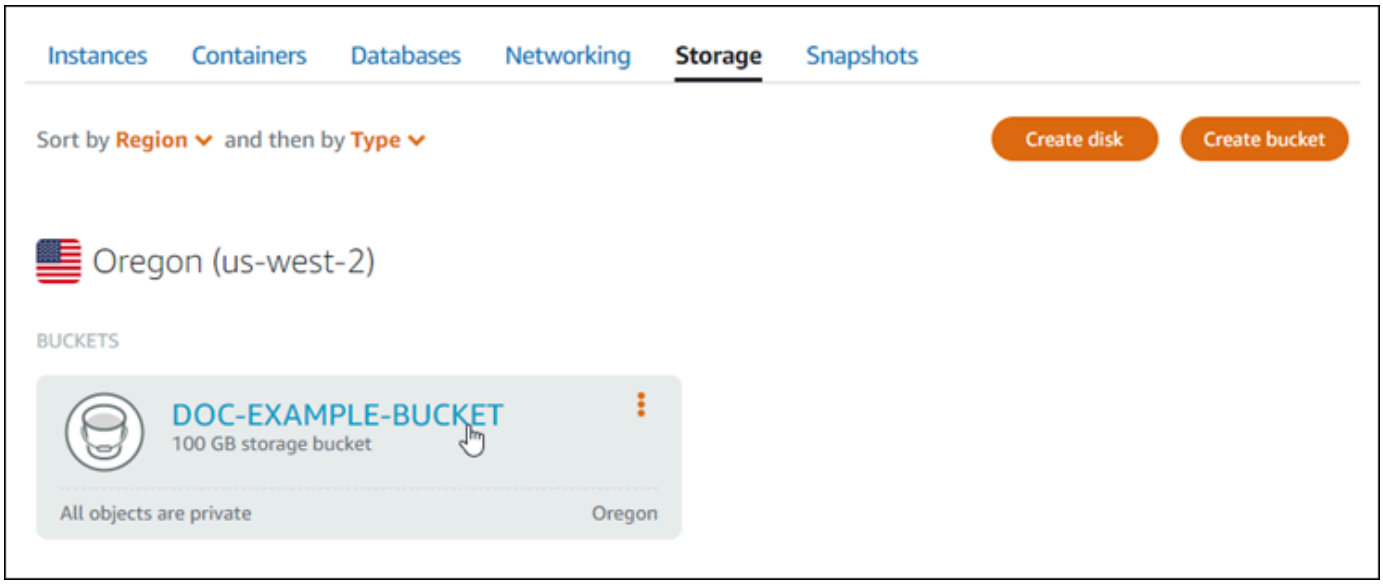
- Lightsail で WordPress インスタンスを作成します。詳細については、「チュートリアル: Amazon Lightsail で WordPress インスタンスを起動して設定する」を参照してください。

- Lightsail オブジェクトストレージサービスでバケットを作成します。詳細については、「バケットの作成」を参照してください。

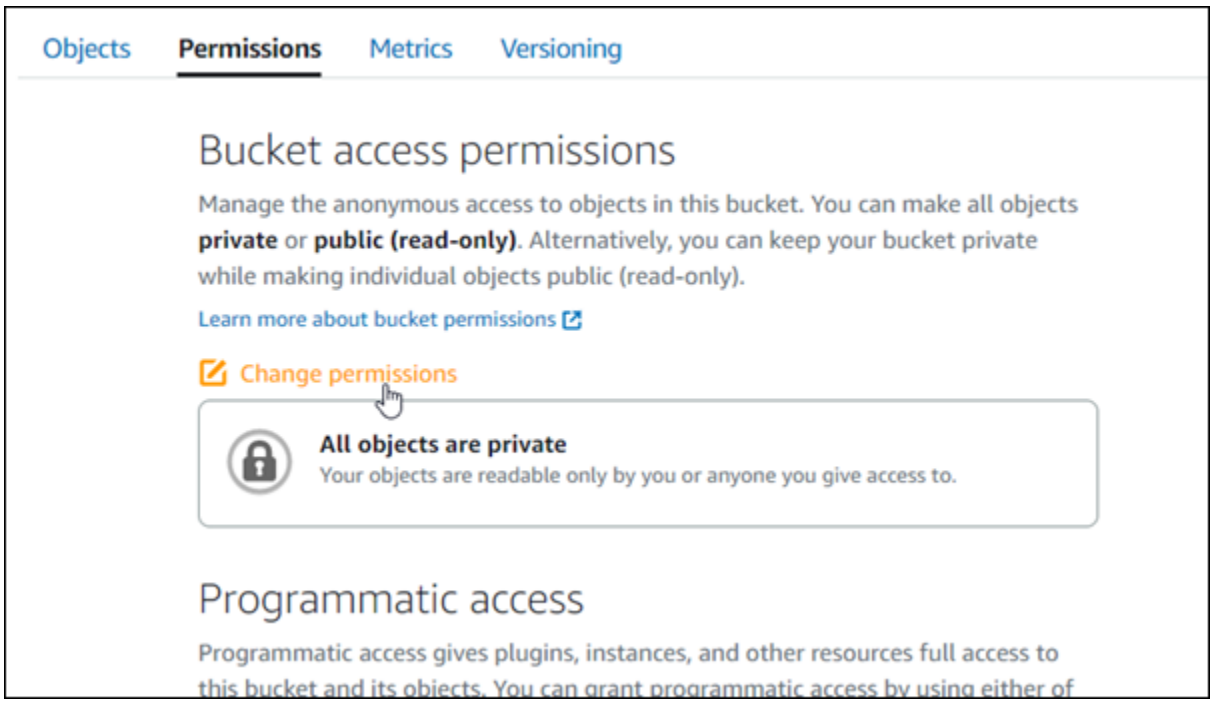
ステップ 2: バケットのアクセス許可を変更する

次の手順を実行して、WordPress インスタンスと Offload Media Lite プラグインのバケットへのアクセス許可を編集します。バケットのアクセス許可は個々のオブジェクトを公開 (読み取り専用)に設定する必要があります。また、WordPress インスタンスをバケットのアクセスロールにアタッチする必要があります。バケット許可の詳細については、「バケットのアクセス許可」を参照してください。

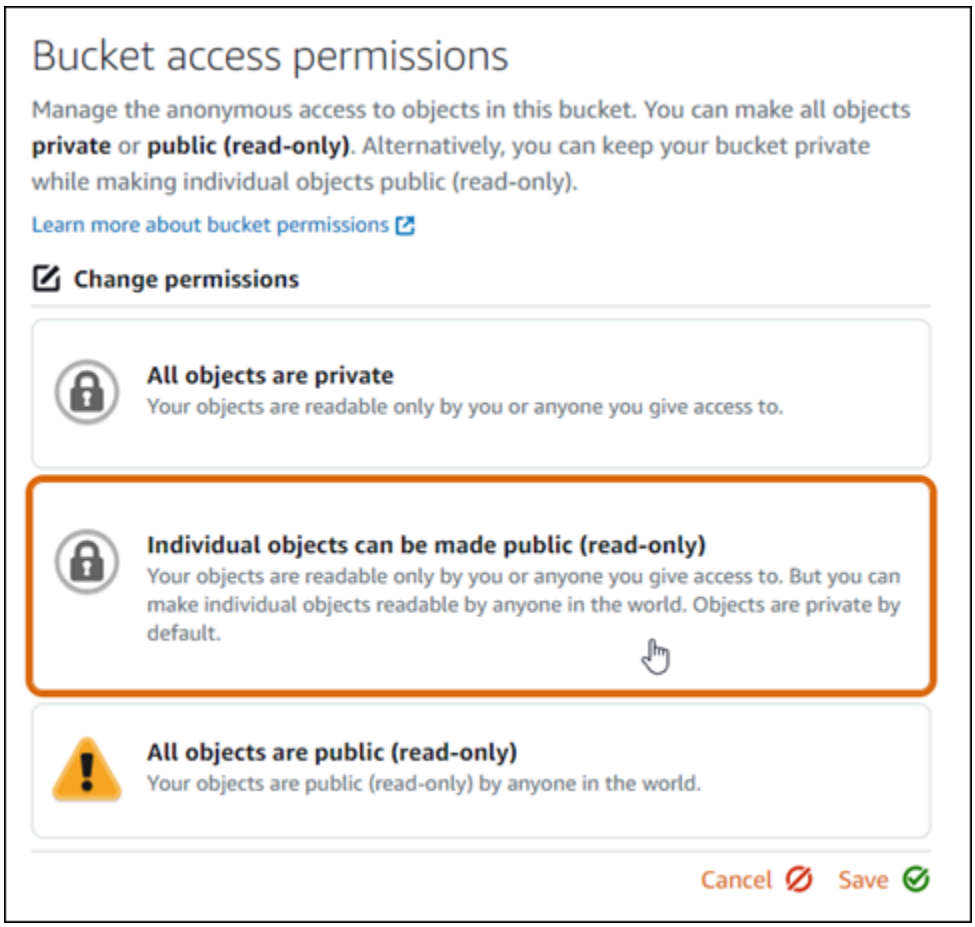
1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ストレージ] を選択します。
3. WordPress ウェブサイトで使用するバケットの名前を選択します。



4. バケット管理ページで [Permissions] (許可) タブを選択します。
5. ページの「バケットのアクセス許可」セクションで [Change permissions](許可の変更) を選択します。

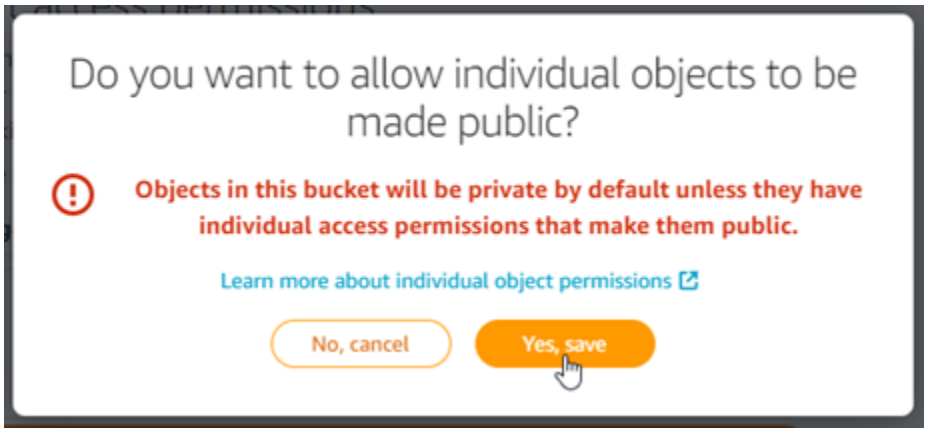


6. 個々のオブジェクトを選択して公開し、読み取り専用にすることができます。



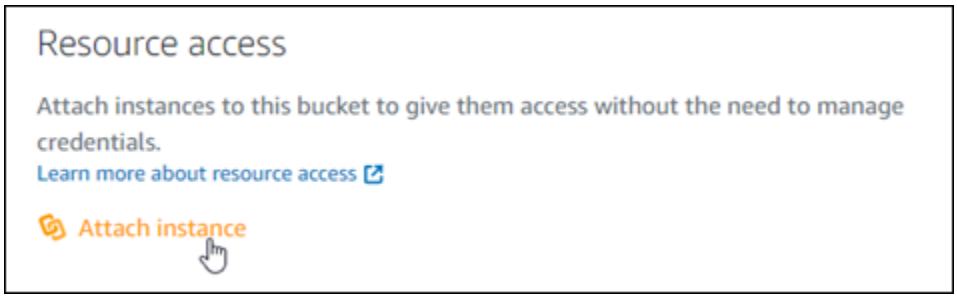
7. [保存] を選択します。

8. 表示される確認プロンプトで、[はい、選択]を選択します。

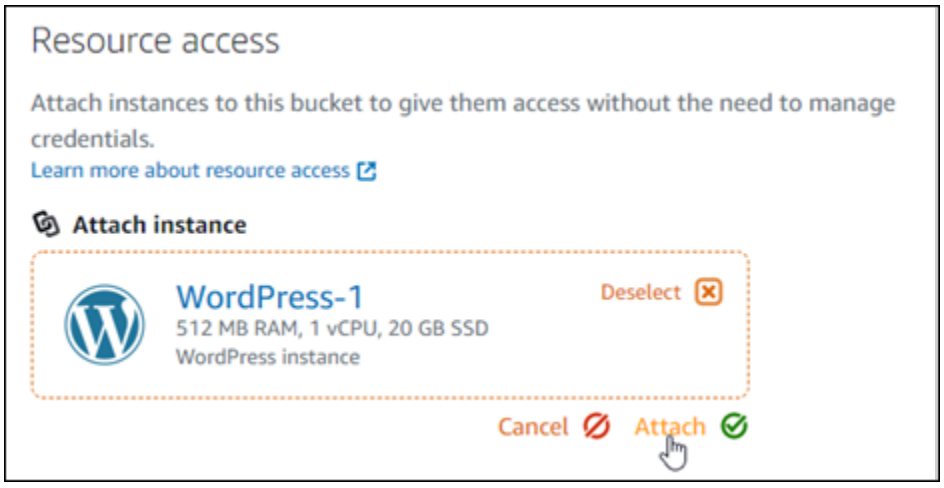


しばらくすると、バケットは個々のオブジェクトにアクセスを許可するように設定されます。これにより、Offload Media Lite プラグインを使用して、WordPress ウェブサイトからバケットにアップロードされたオブジェクトをユーザーが読み取れるようになります。

9. ページの [リソースアクセス] セクションまでスクロールし、[Attach instance] (インスタンスの添付) を選択します。



10. 表示されるドロップダウンリストで WordPress インスタンス名を選択し、[Attach] (添付) を選択します。



しばらくすると、WordPress インスタンスがバケットに添付されます。これにより、WordPress インスタンスにアクセスして、バケットとそのオブジェクトを管理するアクセス権が与えられます。

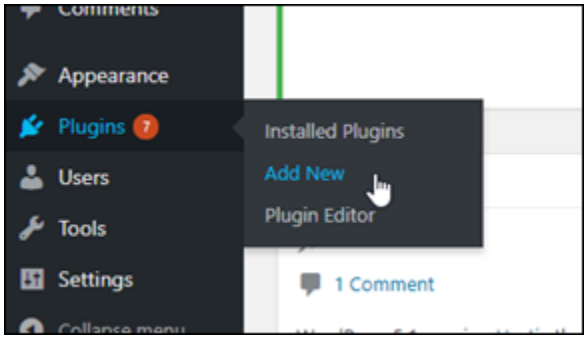
ステップ 3: WordPress ウェブサイトに WP Offload Media プラグインをインストールする

次のステップに従って、WordPress ウェブサイトに WP Offload Media Lite プラグインをインストールします。このプラグインは、WordPressのメディアアップローダーを介して追加されたイメージ、ビデオ、ドキュメント、およびその他のメディアを Lightsail バケットに自動的にコピーします。詳細については、WordPress ウェブサイトの WP Offload Media Lite を参照してください。

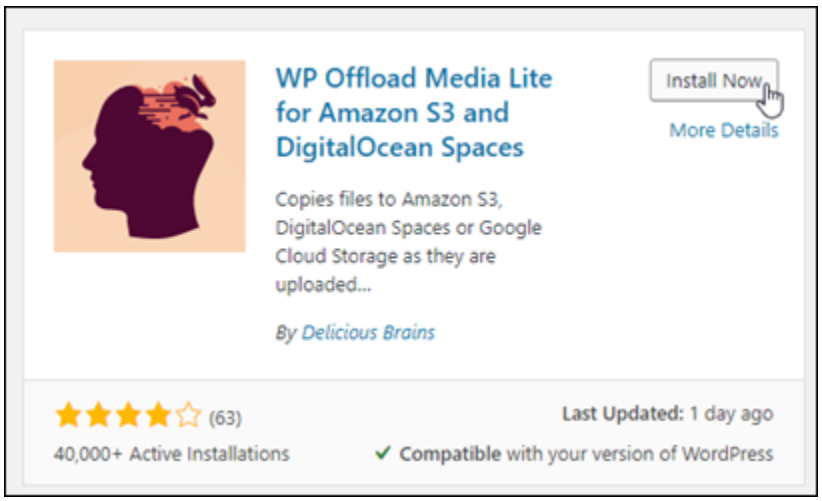
1. 管理者として WordPress ウェブサイトのダッシュボードにサインインします。

   詳細については、「Amazon Lightsailの Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

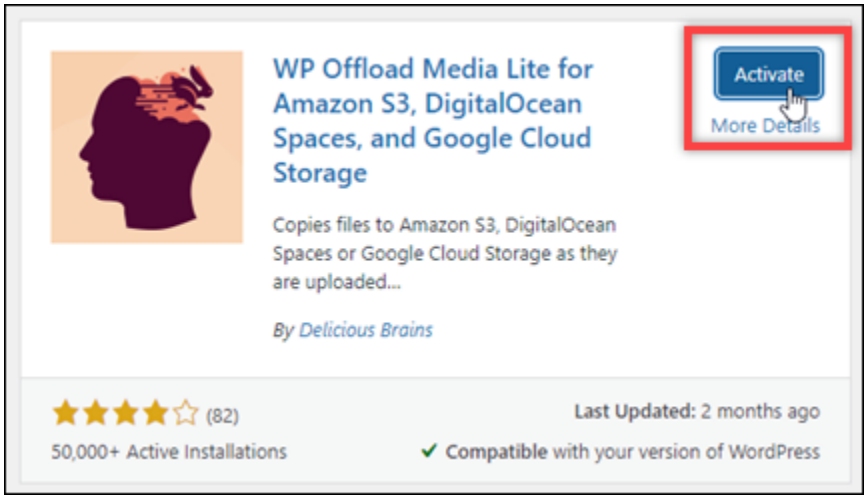
2. 左側のナビゲーションメニューのプラグインを一時停止し、[Add New] (新規追加) を選択します。



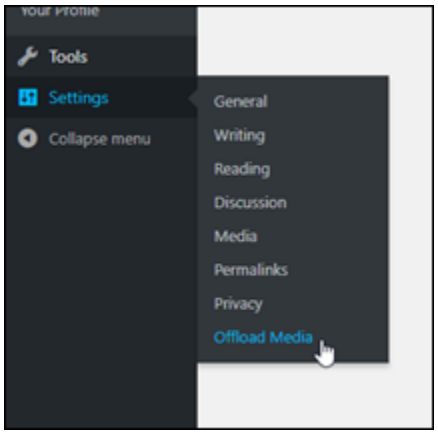
3. [WP Offload Media Lite] を検索します。
4. 検索結果の中から WP Offload Media プラグインの横の [Install Now] (今すぐインストール)を選択します。



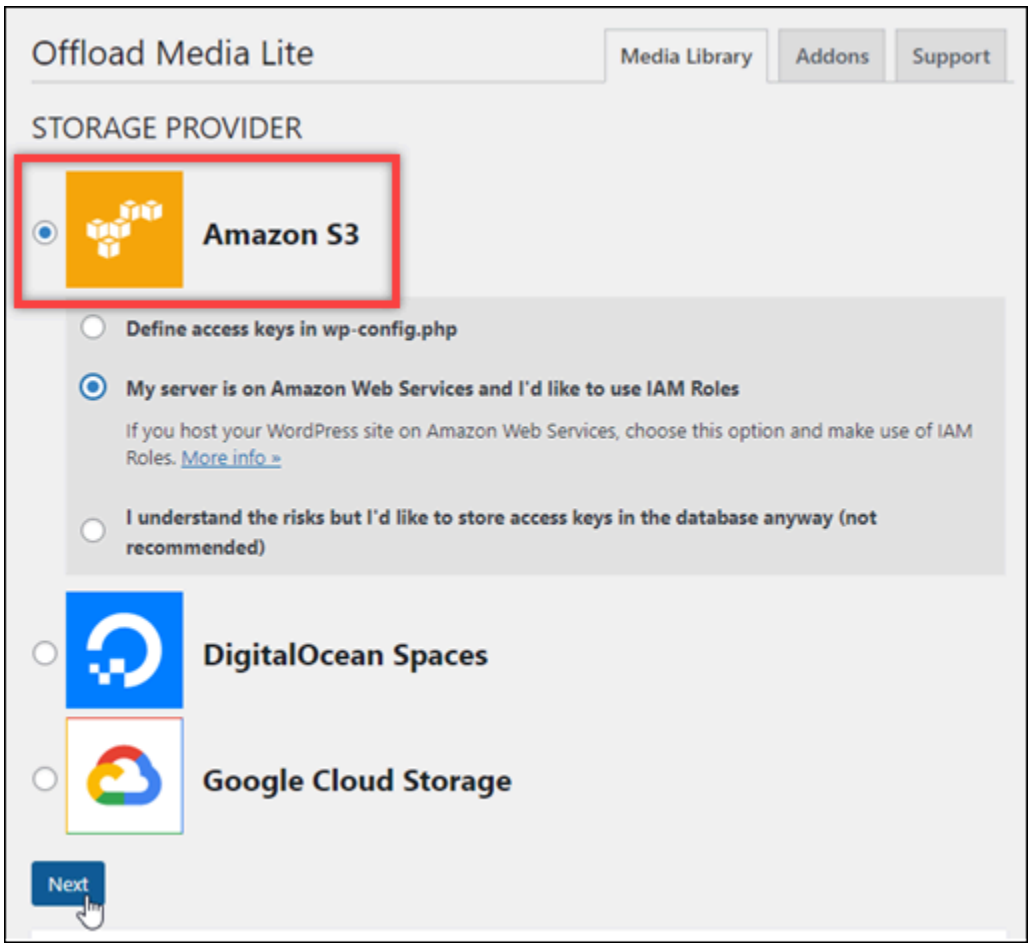
5. プラグインのインストールが完了したら、[アクティベート] を選択します。



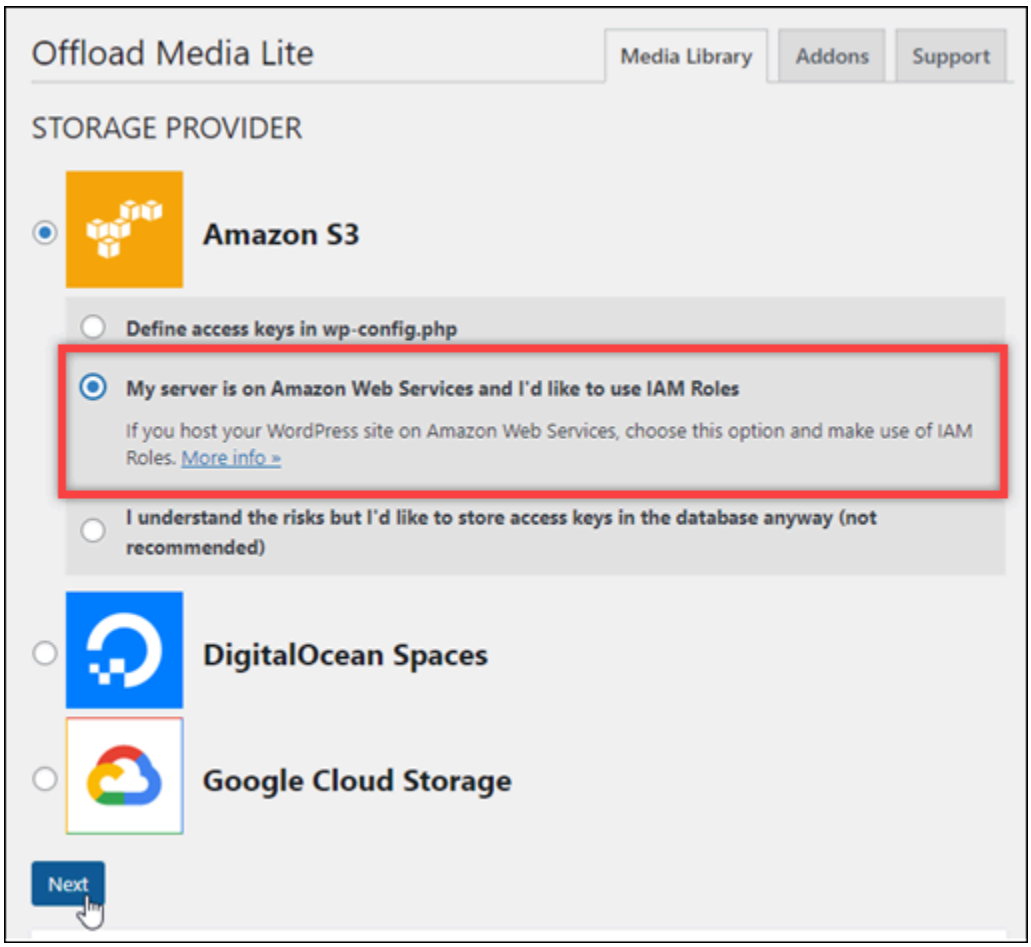
6. 左のナビゲーションメニューで、[Settings] (設定) を選択し、[Offload Media] を選択します。



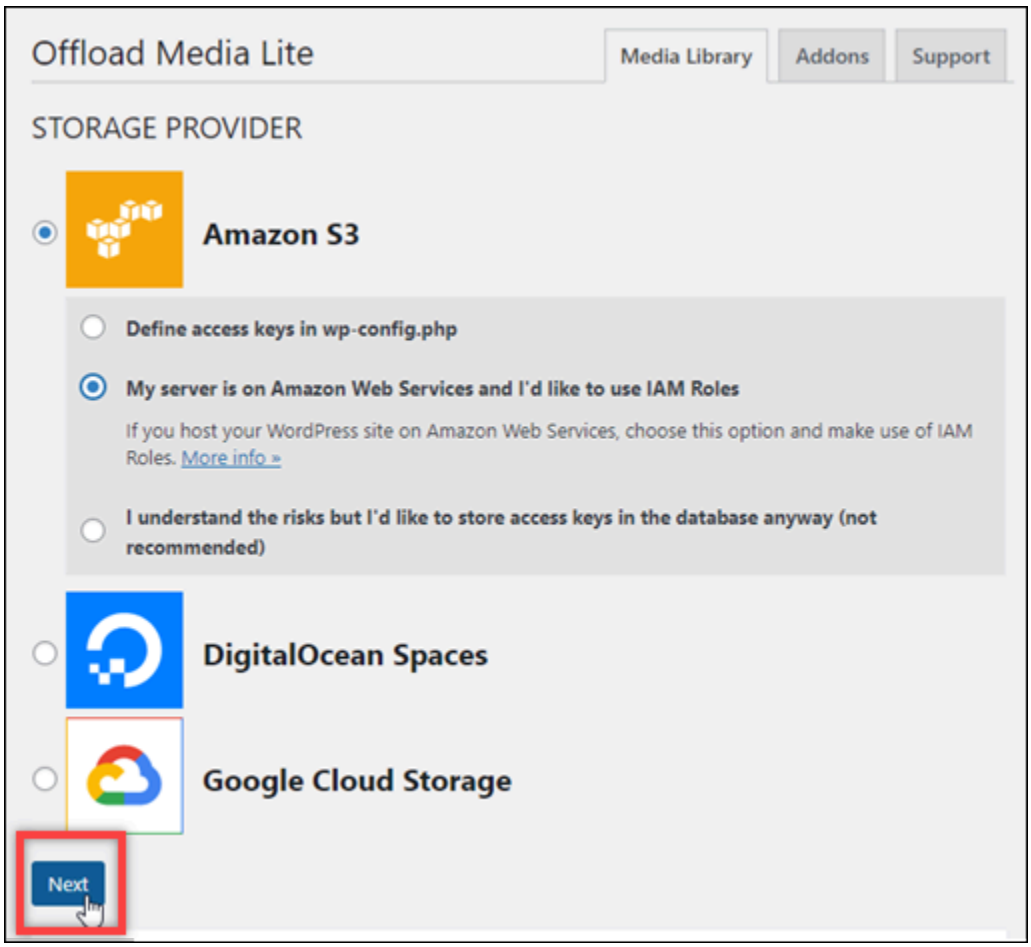
7. Offload Media ページで [Amazon S3] をストレージプロバイダとして選択します。



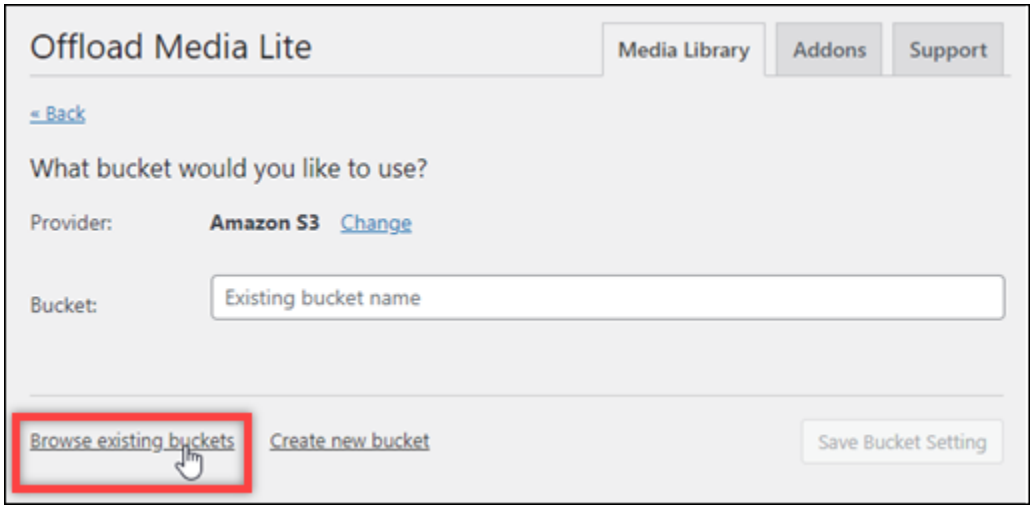
8. [私のサーバーは Amazon Web Services 上にあり、IAM ロールを使いたい] を選択します。


Offload Media Lite
Media Library
Addons
Support
STORAGE PROVIDER
Amazon S3
Define access keys in wp-config.php
My server is on Amazon Web Services and I'd like to use IAM Roles
If you host your WordPress site on Amazon Web Services, choose this option and make use of IAM Roles. More info »
I understand the risks but I'd like to store access keys in the database anyway (not recommended)
DigitalOcean Spaces
Google Cloud Storage
Next

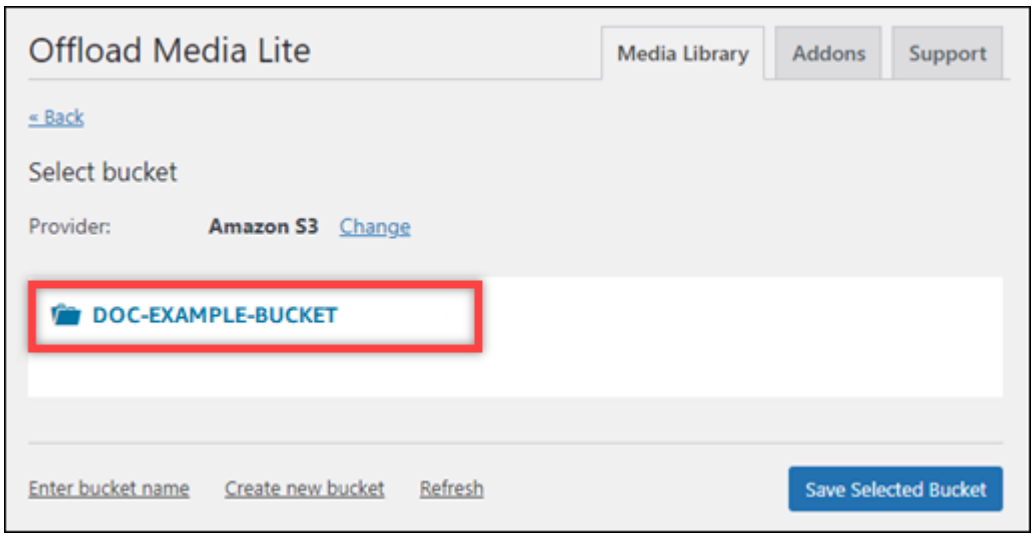
9. [次へ] を選択します。



10. 「どのバケットを使用しますか?」のページで [Browse existing buckets] (既存のバケットを参照する) を選択します。

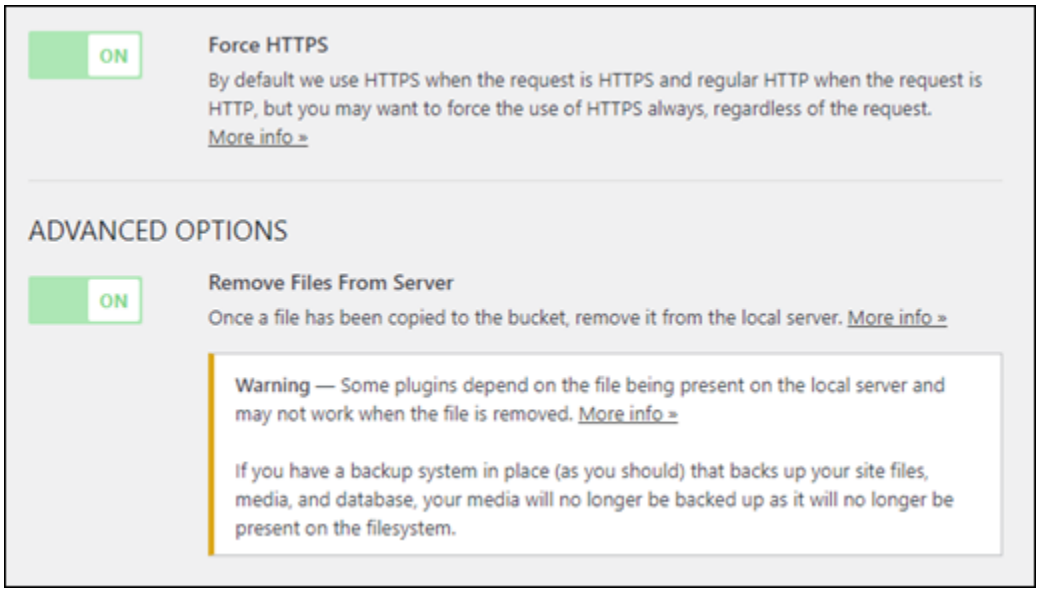


11. WordPress インスタンスで使用したいバケット名を選択します。



12. 表示されるOffload Media Lite Settings画面で、HTTPS の強制実行およびサーバーからファイルを削除をオンにします。

    - Lightsail バケットは、デフォルトで HTTPS を使用してメディアファイルを提供するため、[Force HTTPS] 設定をオンにする必要があります。この機能がオンになっていないと、WordPressのウェブサイトから Lightsail バケットにアップロードされたメディアファイルが、ウェブサイトの訪問者に正しく提供されません。
    - [Remove Files From Server] は、Lightsail バケットにアップロードされたメディアがインスタンスのディスクにも保存されないようにする設定です。この機能をオンにしていない場合、Lightsail バケットにアップロードされたメディアファイルは、WordPress インスタンスのローカルストレージにも保存されます。



13. [Save changes] (変更の保存) をクリックします。

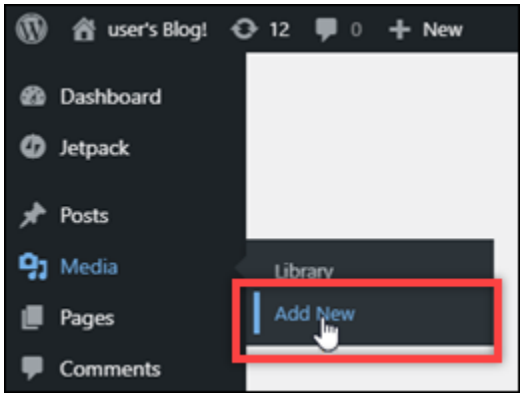
> **Note**
>
> 後でOffload Media Lite Settings画面に戻るには、左のナビゲーションメニューで [設定] を一時停止し、[Offload Media Lite] を選択します。

これで、WordPress ウェブサイトは Media Lite プラグインを使用できるように設定されました。次回 WordPress を使用してメディアファイルをアップロードすると、そのファイルは自動的に Lightsail バケットにアップロードされ、バケットによって提供されます。設定をテストするには、このチュートリアルの次のセクションに進みます。

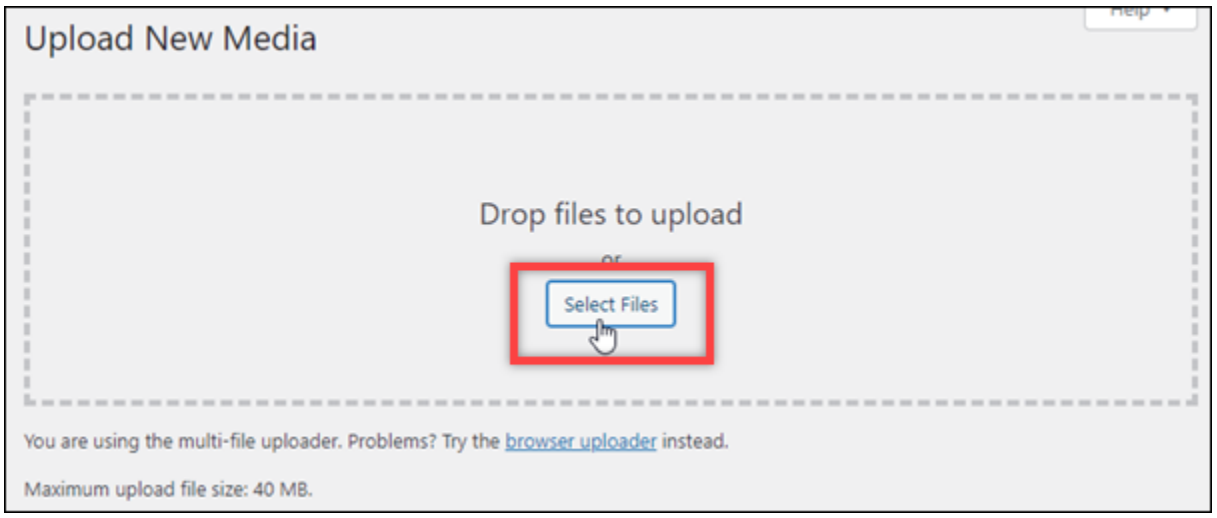
ステップ 4: Lightsail バケットと WordPress ウェブサイトの接続をテストする

次の手順を実行して、WordPress インスタンスにメディアファイルをアップロードして、無事にアップロードされ Lightsail バケットから提供されていることを確認します。

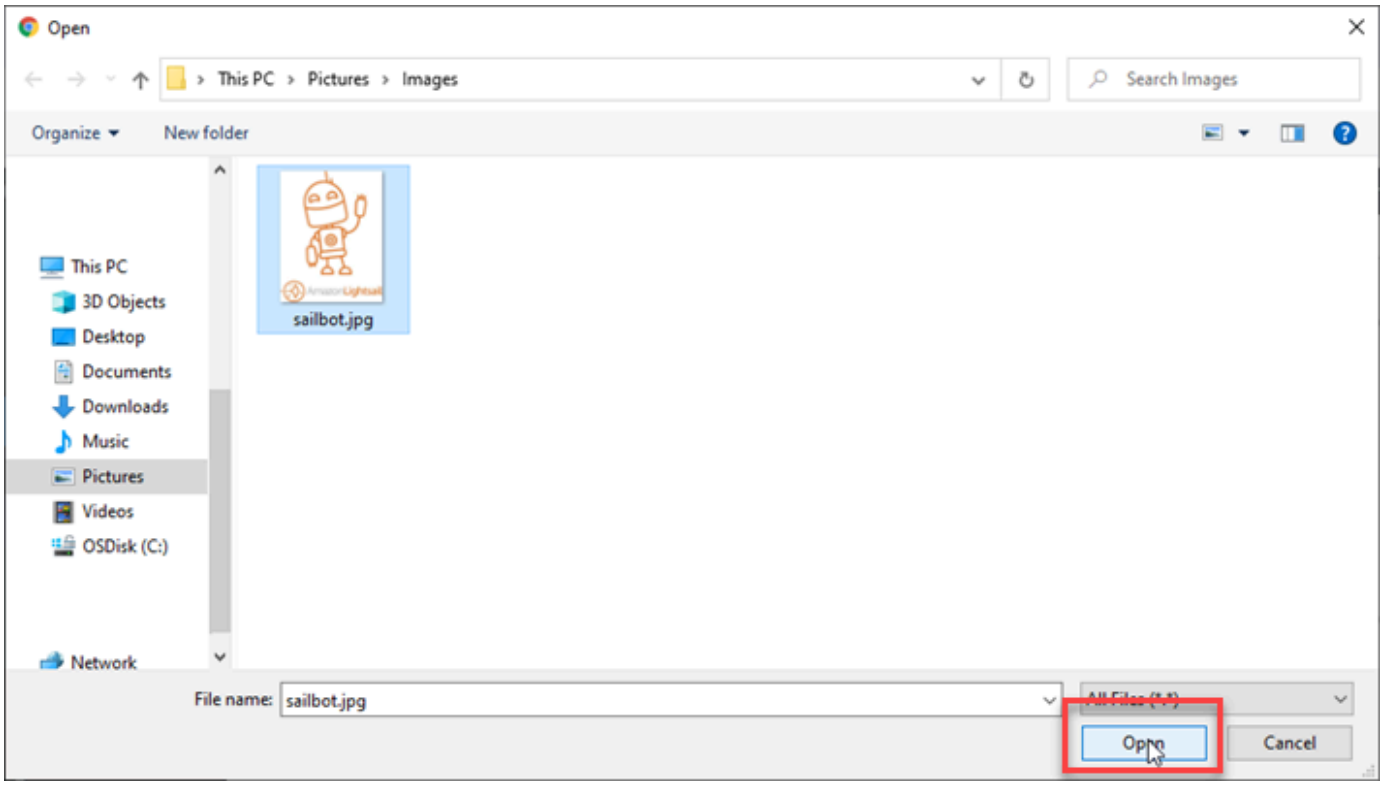
1. WordPress ダッシュボードの左側のナビゲーションメニューで [メディア] を一時停止し、[Add New] (新規追加) を選択します。



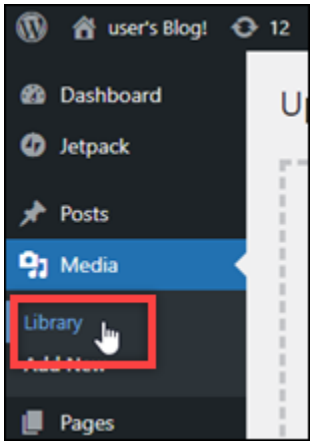
2. 表示される、[新しいメディアをアップロード] 画面で [ファイルを選択] を選択します。



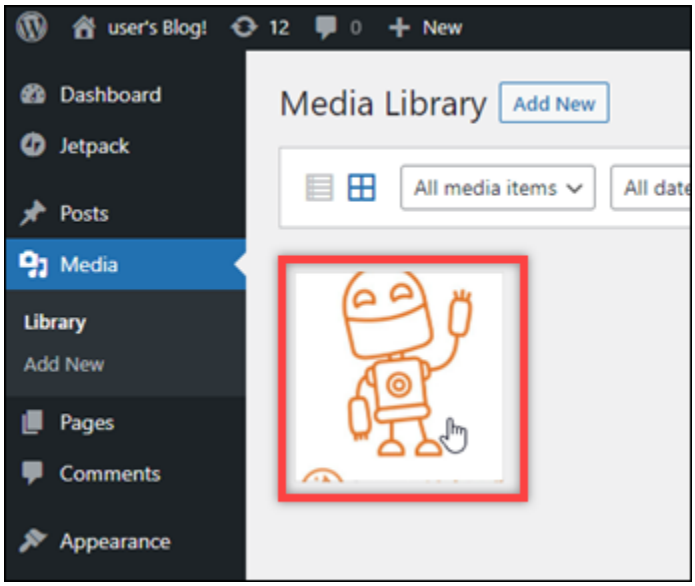
3. ローカルコンピュータからアップロードするメディアファイルを選択し、[開く] を選択します。



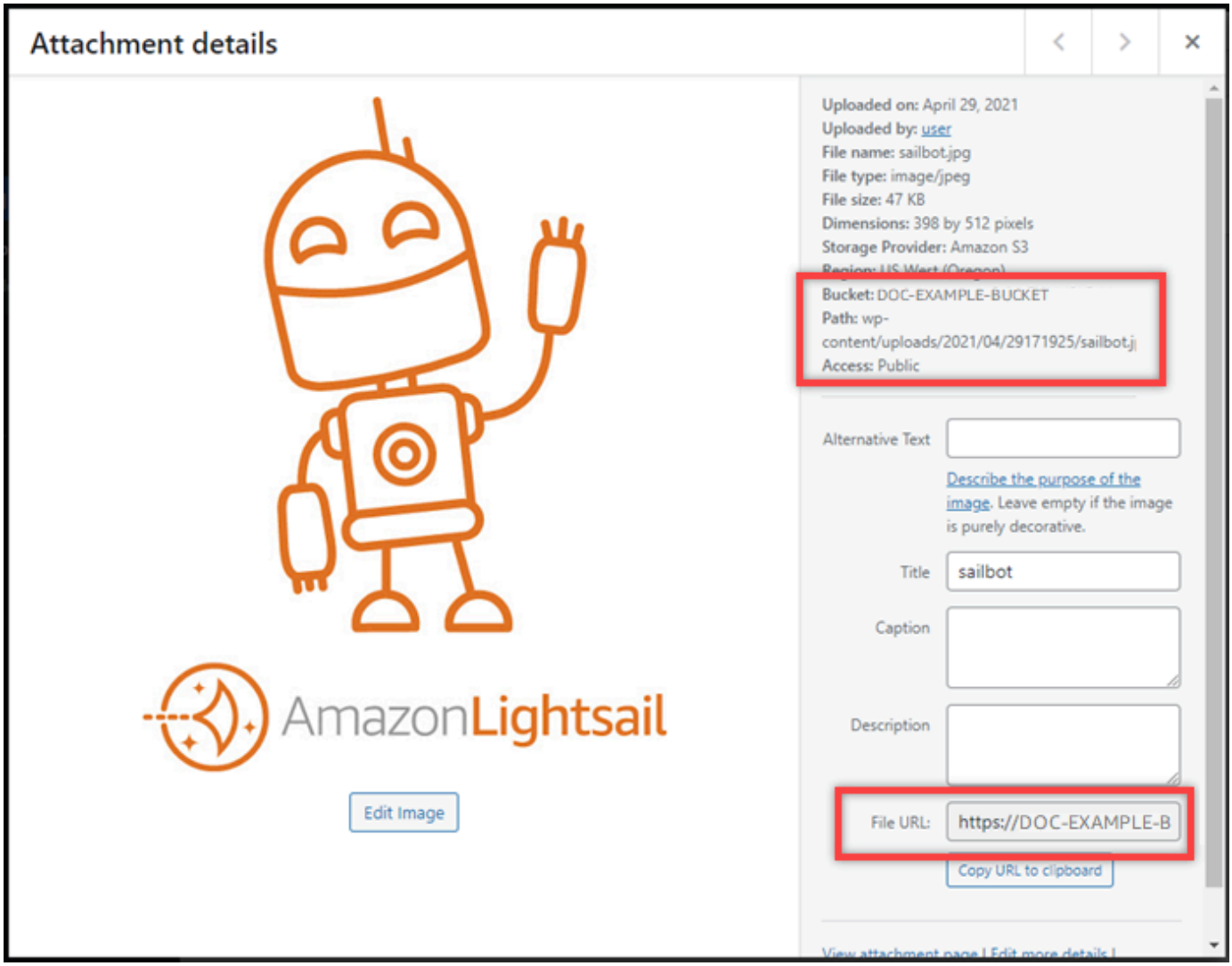
4. ファイルのアップロードが完了したら、左のナビゲーションメニューにある [メディア] の [ライブラリ] を選択します。



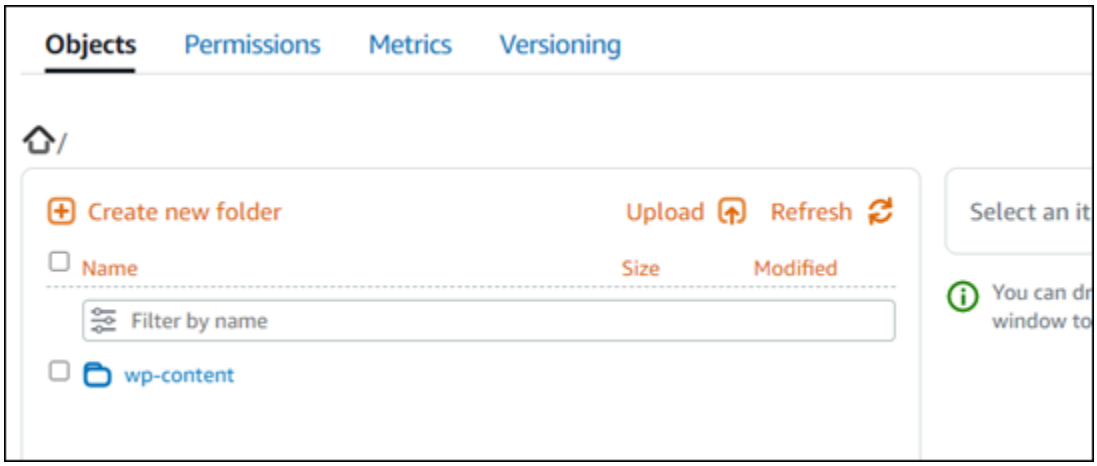
5. 最近アップロードしたファイルを選択します。



6. ファイルの詳細パネルで、バケットおよびファイルの URL フィールドにバケット名が表示されます。



7. Lightsail バケット管理ページの [オブジェクト] タブに移動すると、[wp-content] フォルダーが表示されます。このフォルダは、Offload Media Lite プラグインによって作成され、アップロードしたメディアファイルを保存するために使用されます。



バケットとオブジェクトを管理する

これらは、Lightsail オブジェクトストレージバケットを管理する一般的な手順です。

1. Amazon Lightsail オブジェクトストレージサービスでのオブジェクトとバケットについて説明します。詳細については、「Amazon Lightsail のオブジェクトストレージ」を参照してください。
2. Amazon Lightsail でバケットに付けることができる名前について説明します。詳細については、「Amazon Lightsail でのバケットの命名規則」をご参照ください。
3. バケットを作成して、Lightsail オブジェクトストレージサービスの使用を開始します。詳細については、「Amazon Lightsail におけるバケットの作成 」を参照してください。
4. バケットのセキュリティのベストプラクティスと、バケットに設定できるアクセス許可について説明します。バケット内のすべてのオブジェクトをパブリックまたはプライベートにすることも、オブジェクトを個別に選択してパブリックにすることもできます。また、アクセスキーを作成し、インスタンスをバケットに追加し、他の AWS アカウントにアクセス権を付与することで、バケットへのアクセスを許可することもできます。詳細については、「Amazon Lightsail オブジェクトストレージのセキュリティベストプラクティス」と「Amazon Lightsail でのバケットのアクセス許可を理解する」を参照してください。

   バケットのアクセス許可について理解したら、以下のガイドを参照してバケットへのアクセスを許可してください。

   - Amazon Lightsail でのバケットへのパブリックアクセスをブロックする
   - Amazon Lightsail でのバケットのアクセス許可の設定
   - Amazon Lightsail でのバケット内の個々のオブジェクトに対するアクセス許可の設定
   - Amazon Lightsail でのバケットのアクセスキーの作成
   - Amazon Lightsail でのバケットのリソースアクセスの設定
   - Amazon Lightsail でのバケットのクロスアカウントアクセスの設定
5. バケットのアクセスログの記録を有効にする方法と、アクセスログを使用してバケットのセキュリティを監査する方法について説明します。詳細については、以下のガイドを参照してください。

   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録
   - Amazon Lightsail オブジェクトストレージサービスのバケットのアクセスログの形式
   - Amazon Lightsail オブジェクトストレージサービスのバケットへのアクセスのログ記録を有効にする
   - Amazon Lightsailでのバケットのアクセスログを使用するリクエストの特定

6. Lightsail でバケットを管理する機能をユーザーに付与する IAM ポリシーを作成します。詳細については、「Amazon Lightsail でバケットを管理する IAM ポリシー」を参照してください。
7. バケット内のオブジェクトにラベルを付けて識別する方法について説明します。詳細については、「Amazon Lightsail でのオブジェクトキー名を理解する」を参照してください。
8. ファイルをアップロードしてバケット内のオブジェクトを管理する方法について説明します。詳細については、以下のガイドを参照してください。
   - Amazon Lightsail のバケットにファイルをアップロードする
   - マルチパートアップロードを使用した Amazon Lightsail のバケットへのファイルのアップロード
   - Amazon Lightsail のバケット内のオブジェクトの表示
   - Amazon Lightsail のバケット内のオブジェクトのコピーまたは移動
   - Amazon Lightsail のバケットからのオブジェクトのダウンロード
   - Amazon Lightsail のバケット内のオブジェクトのフィルタリング
   - Amazon Lightsail のバケット内のオブジェクトのタグ付け
   - Amazon Lightsail のバケット内のオブジェクトの削除
9. オブジェクトのバージョニングを有効にすると、バケットに保存されたあらゆるオブジェクトのあらゆるバージョンを保存、取得、復元します。詳細については、「Amazon Lightsail のバケットでのオブジェクトのバージョニングの有効化と一時停止」を参照してください。
10. オブジェクトのバージョニングを有効にすると、バケット内のオブジェクトの以前のバージョンを復元できます。詳細については、「Amazon Lightsail のバケット内のオブジェクトの以前のバージョンの復元」を参照してください。
11. バケットの使用率を監視します。詳細については、「Amazon Lightsail でのバケットのメトリクスの表示」を参照してください。
12. バケットの使用率がしきい値を超えたときにバケットメトリクスが通知されるよう、アラームを設定します。詳細については、「Amazon Lightsail でのバケットメトリクスアラームの作成」を参照してください。
13. ストレージとネットワーク転送量が不足している場合は、バケットのストレージプランを変更します。詳細については、「Amazon Lightsail のバケットのプランの変更」を参照してください。
14. バケットを他のリソースに接続する方法について説明します。詳細については、以下のチュートリアルを参照してください。
   - チュートリアル: WordPress インスタンスの Amazon Lightsail バケットへの接続
   - チュートリアル: Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用

15.使用しなくなったバケットを削除します。詳細については、「Amazon Lightsail でのバケットの削除」を参照してください。

Lightsail コンテンツ配信ネットワークで WordPress を設定する

このガイドでは、Amazon Lightsail ディストリビューションと動作するように WordPress インスタンスを設定する方法について説明します。

すべての Lightsail ディストリビューションは、デフォルトドメインとして HTTPS がデフォルトで有効になっています (例えば: `123456abcdef.cloudfront.net`)。ディストリビューションの設定により、ディストリビューションとインスタンス間の接続が暗号化されているかどうかが決まります。

- WordPress ウェブサイトは HTTP のみを使用しています – ウェブサイトがディストリビューションのオリジンとして HTTP のみを使用し、HTTPS を使用するように設定されていない場合、暗号化されていない接続を使用して SSL/TLS を終了し、すべてのコンテンツリクエストをインスタンスに転送するようにディストリビューションを設定できます。
- HTTPS を使用する WordPress ウェブサイト – ウェブサイトがディストリビューションのオリジンとして HTTPS を使用する場合、暗号化された接続を使用してすべてのコンテンツリクエストをインスタンスに転送するようにディストリビューションを設定できます。この設定は、エンドツーエンド暗号化と呼ばれます。

ディストリビューションを作成する

WordPress インスタンスの Lightsail ディストリビューションを設定するには、次の手順を実行します。詳細については、「the section called “ディストリビューションを作成する”」を参照してください。

前提条件

the section called “WordPress” の説明に従って WordPress インスタンスを作成して設定します。

WordPress インスタンスのディストリビューションを作成するには

1. 左側のナビゲーションペインで、[ネットワーク] を選択します。
2. [ディストリビューションの作成] を選択します。

3. [オリジンを選択する] では、WordPress インスタンスを実行しているリージョンを選択し、WordPress インスタンスを選択します。インスタンスにアタッチした静的 IP アドレスが自動的に使用されます。
4. [キャッシュ動作] で、[WordPress に最適] を選択します。
5. (選択可能) エンドツーエンドの暗号化を設定するには、オリジンプロトコルポリシーを [HTTPS のみ] に変更します。詳細については、「the section called “オリジンプロトコルポリシー”」を参照してください。
6. 残りのオプションを設定して、[ディストリビューションの作成] を選択します。
7. [カスタムドメイン] タブで、[証明書の作成] を選択します。証明書の一意の名前を入力し、ドメインとサブドメインの名前を入力してから、[証明書の作成] を選択します。
8. [証明書のアタッチ] を選択します。
9. [DNS レコードを更新する] には、[理解できる] を選択します。

DNS レコードの更新

次の手順を実行して、Lightsail DNS ゾーンの DNS レコードを更新します。

ディストリビューションの DNS レコードを更新するには

1. 左側のナビゲーションペインで [ドメインと DNS] を選択します。
2. DNS ゾーンを選択し、[DNS レコード] タブを選択します。
3. 証明書で指定したドメインの A レコードと AAAA レコードを削除します。
4. [レコードの追加] を選択し、ドメインをディストリビューションのドメインに解決する CNAME レコードを作成します (例: d2vbec9EXAMPLE.cloudfront.net)。
5. [保存] を選択します。

静的コンテンツをディストリビューションでキャッシュできるようにする

次の手順を実行して、WordPress インスタンスの `wp-config.php` ファイルを編集して、ディストリビューションで動作するようにします。

> **Note**
>
> この手順を開始する前に、WordPress インスタンスのスナップショットを作成することをお勧めします。スナップショットは、何か問題が発生した場合、これを元に別のインスタンス

を作成できるバックアップとして使用できます。詳細については、「Linux または Unix インスタンスのスナップショットを作成する」を参照してください。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、WordPress インスタンスの横に表示されるブラウザベースの SSH クライアントのアイコンを選択します。
3. インスタンスに接続したら、次のコマンドを入力して、wp-config.php ファイルのバックアップを作成します。何らかの問題が発生した場合は、バックアップを使用してファイルを復元することができます。

```
sudo cp /opt/bitnami/wordpress/wp-config.php /opt/bitnami/wordpress/wp-config.php.backup
```

4. 次のコマンドを入力して、 Vim を使用し、wp-config.php ファイルを開きます。

```
sudo vim /opt/bitnami/wordpress/wp-config.php
```

5. I キーを押して Vim の挿入モードに移ります。
6. ファイルで、次のコード行を削除します。

```
define('WP_SITEURL', 'http://' . $_SERVER['HTTP_HOST'] . '/');
define('WP_HOME', 'http://' . $_SERVER['HTTP_HOST'] . '/');
```

7. 使用している WordPress のバージョンに応じて、次のいずれかのコードの行をファイルに追加します。

   - バージョン 3.3 以前を使用している場合、以前にコードを削除した箇所に次のコード行を追加します。

```
define('WP_SITEURL', 'https://' . $_SERVER['HTTP_HOST'] . '/');
define('WP_HOME', 'https://' . $_SERVER['HTTP_HOST'] . '/');
if (isset($_SERVER['HTTP_CLOUDFRONT_FORWARDED_PROTO'])
&& $_SERVER['HTTP_CLOUDFRONT_FORWARDED_PROTO'] === 'https') {
$_SERVER['HTTPS'] = 'on';
}
```

   - バージョン 3.3.1-5 以降を使用している場合、ファイルの削除した箇所に、次のコード行を追加します。

```
define('WP_SITEURL', 'http://DOMAIN/');
define('WP_HOME', 'http://DOMAIN/');
if (isset($_SERVER['HTTP_CLOUDFRONT_FORWARDED_PROTO'])
&& $_SERVER['HTTP_CLOUDFRONT_FORWARDED_PROTO'] === 'https') {
$_SERVER['HTTPS'] = 'on';
}
```

8. ESC キーを押して Vim の挿入モードを終了し、:wq! を入力して Enter キーで編集内容を保存しで (書き込んで) Vim を終了します。
9. 次のコマンドを入力して、インスタンス上の Apache サービスを再起動します。

```
sudo /opt/bitnami/ctlscript.sh restart apache
```

10. Apache サービスが再起動するまでしばらく待ってから、ディストリビューションがコンテンツをキャッシュしているかどうかをテストします。詳細については、「Amazon Lightsail ディストリビューションをテストする」を参照してください。
11. 何らかの問題が発生した場合は、ブラウザベースの SSH クライアントを使用してインスタンスに再接続します。次のコマンドを実行して、このガイドで先に作成したバックアップを使用して wp-config.php ファイルで復元します。

```
sudo cp /opt/bitnami/wordpress/wp-config.php.backup /opt/bitnami/wordpress/wp-config.php
```

ファイルを復元したら、次のコマンドを入力して Apache サービスを再起動します:

```
sudo /opt/bitnami/ctlscript.sh restart apache
```

ディストリビューションに関する追加情報

ここでは、Lightsail でディストリビューションを管理するのに役立つ記事をいくつか紹介します。

- コンテンツ配信ネットワークディストリビューション
- ディストリビューションの作成
- ディストリビューションのリクエストとレスポンスを理解する
- ディストリビューションをテストする
- ディストリビューションのオリジンを変更する

- ディストリビューションのキャッシュ動作を変更する
- ディストリビューションのキャッシュをリセットする
- ディストリビューションのプランを変更する
- ディストリビューションのカスタムドメインを有効にする
- ドメインをディストリビューションにポイントする
- ディストリビューションのカスタムドメインを変更する
- ディストリビューションのカスタムドメインを無効にする
- ディストリビューションのメトリクスを表示する
- ディストリビューションを削除する

Lightsail の WordPress インスタンスのメールを有効にする

Amazon Lightsail の WordPress インスタンスで E メール を有効にできます。Amazon Simple Email Service (Amazon SES) で SMTP サービスを設定します。次に、インスタンスで WP Mail SMTP プラグインを有効化して設定します。E メールが有効になると、WordPress 管理者はユーザープロファイルのパスワードリセットをリクエストできます。ブログの投稿、ウェブサイトの更新、その他のプラグインメッセージに関する E メール通知が送信されます。このガイドでは、Amazon SES を使用して、Amazon Lightsail の WordPress インスタンスでメールを有効にする方法について説明します。

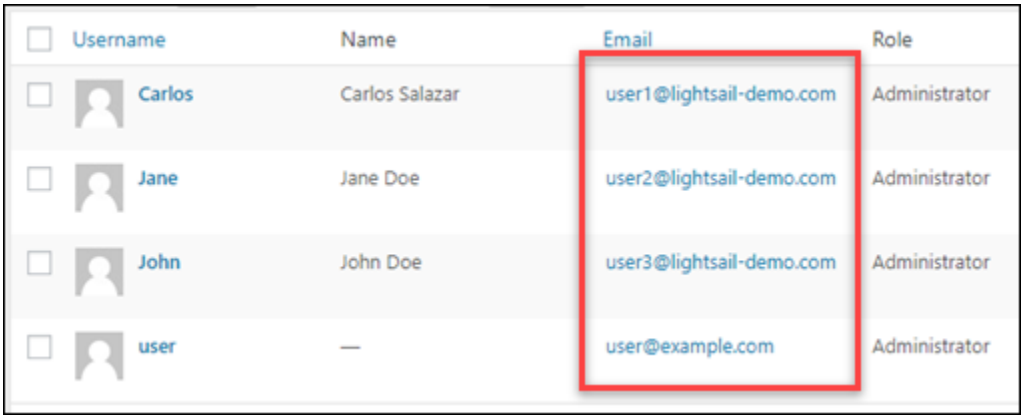
目次

- ステップ 1: 制限のレビュー
- ステップ 2: 前提条件を完了させる
- ステップ 3: Amazon SES で SMTP 認証情報を作成する
- ステップ 4: Amazon SES のドメインを検証する
- ステップ 5: Amazon SES のメールアドレスを検証する
- ステップ 6: WordPress インスタンスに WP Mail SMTP プラグインを設定する

詳細については、Amazon SES ドキュメントの「メールを送信するための Amazon SES SMTP インターフェイスの使用」を参照してください。

ステップ 1: 制限のレビュー

Amazon SES サンドボックスの新しい Amazon Web Services (AWS) アカウントは、確認済みのアドレスおよびドメインのみにメールを送信することができます。このような場合は、ウェブサイトのドメインを確認し、WordPress 管理者のメールアドレスを確認することをお勧めします。自分のメールアドレスを取得するには、WordPress ウェブサイトのダッシュボードにサインインし、左側のナビゲーションメニューで [ユーザー] を選択します。以下の例のように、[メール] 列に管理者のメールアドレスが表示されます。



> **Note**
>
> デフォルトの `user` プロファイルは `user@example.com`メールアドレスを使用して設定されます。これを有効なメールアドレスに変更する必要があります。詳細については、WordPress ドキュメントの「ユーザープロファイル画面」を参照してください。

任意のアドレスおよびドメインにメールを送信するには、アカウントを Amazon SES サンドボックスの外に移動するようにリクエストする必要があります。詳細については、Amazon SES ドキュメントの「Amazon SES サンドボックスの外への移動」を参照してください。

ステップ 2: 前提条件を完了させる

WordPress インスタンスで E メールを有効にするには、次のタスクを完了する必要があります。

- Lightsail で WordPress インスタンスを作成します。詳細については、「チュートリアル: Amazon Lightsail で WordPress インスタンスを起動して設定する」を参照してください。
- Lightsail DNS ゾーンを使用して登録されたドメインを WordPress インスタンスに選択します。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。

- Amazon SES にサインアップし、サービスの詳細について参照します。Amazon SES へのサインアップの詳細については、Amazon SES ドキュメントの「Amazon SES クイックスタート」を参照してください。Amazon SES の詳細については、Amazon SES ドキュメントの以下のガイドを参照してください。
  - Amazon SES デベロッパーガイド
  - Amazon SES のよくある質問
  - Amazon SES 料金表
  - Amazon SES Service Quotas

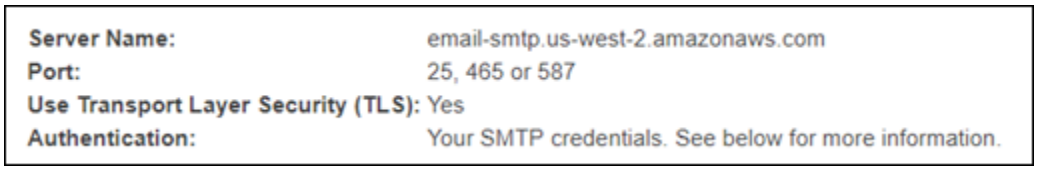
ステップ 3: Amazon SES で SMTP 認証情報を作成する

このガイドの後半で設定する WP Mail SMTP プラグインを設定するには、Amazon SES アカウントで SMTP 認証情報を作成する必要があります。詳細については、Amazon SES ドキュメントの「Amazon SES の SMTP 認証情報の取得」を参照してください。

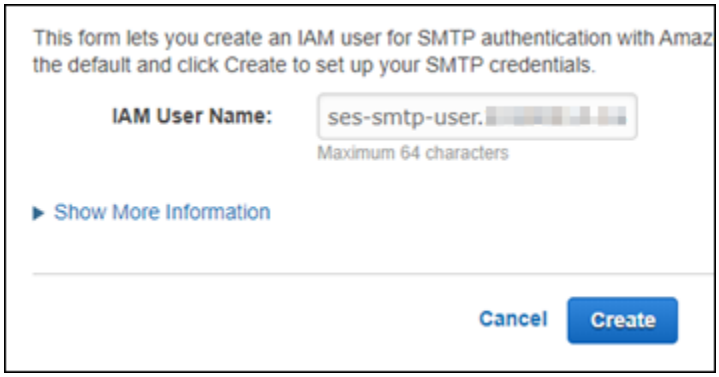
Amazon SES で SMTP 認証情報を作成するには

1. Amazon SES コンソールにサインインします。
2. 左側のナビゲーションメニューから、[SMTP 設定] を選択します。

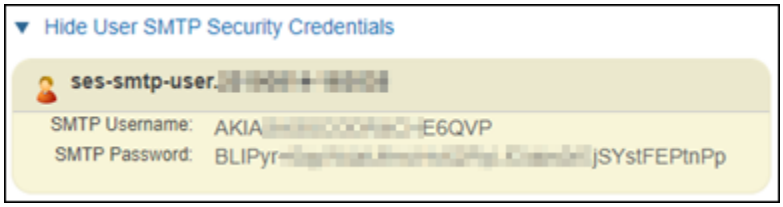
   [SMTP 設定] ページには、SMTP サーバー名、ポート、および TLS 設定が表示されます。WordPress インスタンスの WP Mail SMTP プラグインを設定する際は、後で必要になるため、これらの値を記録します。

3. [SMTP 認証情報の作成] を選択します。
4. [IAM ユーザー名] テキストボックスで [作成] を選択します。ユーザー名はデフォルトのままにします。



5. [Show User SMTP Security Credentials (ユーザー SMTP セキュリティ認証情報の表示)] を選択して SMTP ユーザー名とパスワードを表示するか、[認証情報のダウンロード] を選択して同じ情報を含む CSV ファイルをダウンロードします。WordPress インスタンスで WP Mail SMTP プラグインを設定するときに、後でこれらの認証情報が必要になります。



> **Note**
>
> Amazon SES コンソールに作成された認証情報は、アカウントの AWS Identity and Access Management (IAM) に自動的に追加されます。

ステップ 4: Amazon SES のドメインを検証する

Amazon SES では、ドメインを検証して、それを所有していることを確認し、他のユーザーに使用されないようにする必要があります。ドメインを検証すると、そのドメインのすべてのメールアドレスを検証することになるため、そのドメインのメールアドレスを個別に検証する必要はありません。たとえば、ドメイン `example.com` を検証する場合、`user1@example.com`、`user2@example.com`、または `example.com` の他の任意のユーザーから E メールを送信できます。詳細については、Amazon SES ドキュメントの「Amazon SES のドメインの検証」を参照してください。

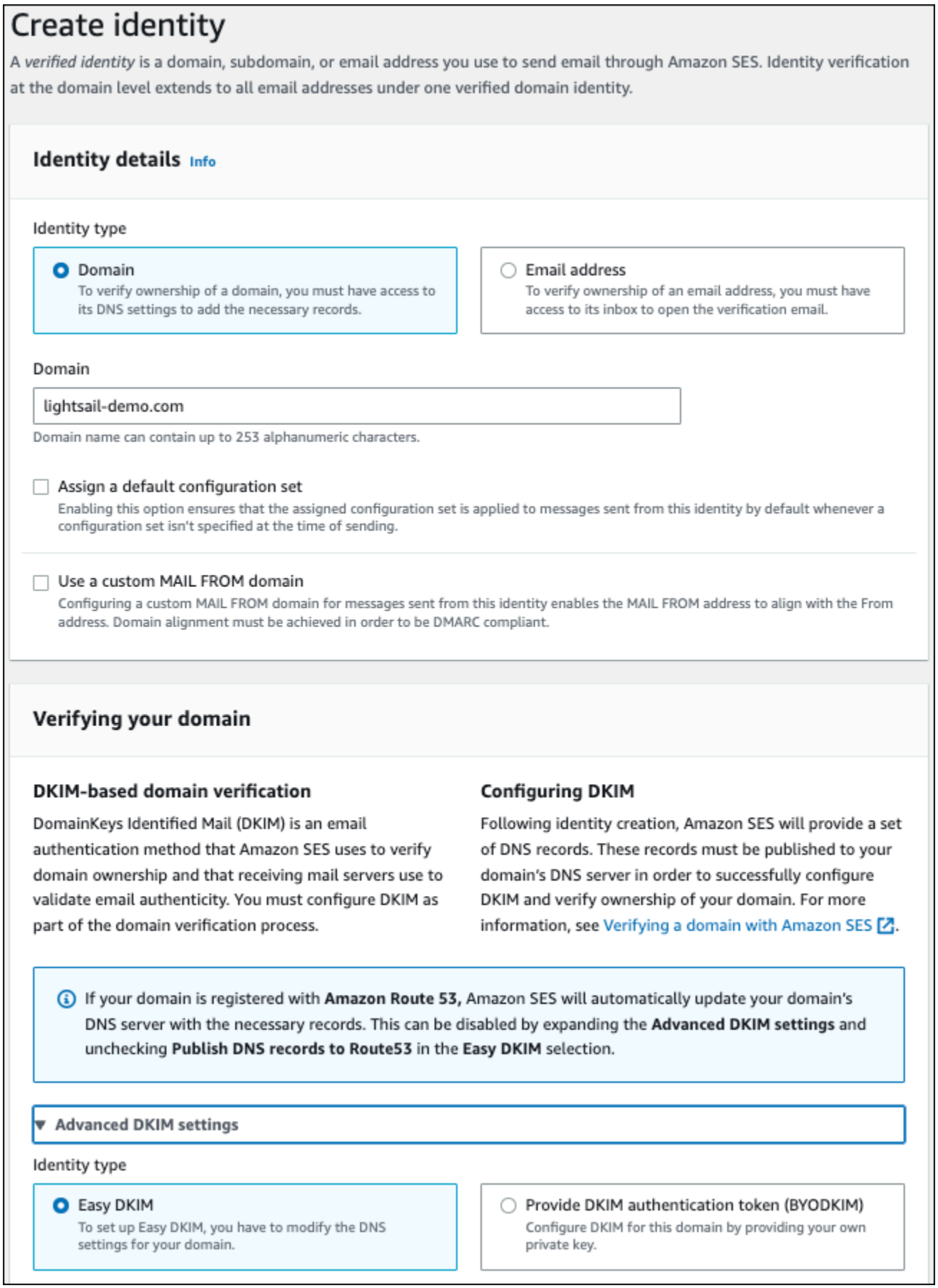
Amazon SES のドメインを検証するには

1. Amazon SES コンソールで、左ナビゲーションメニューから [検証済み ID] を選択します。
2. [ID の作成] を選択します。

3. 検証するドメインを入力し、[アイデンティティの作成] を選択します。

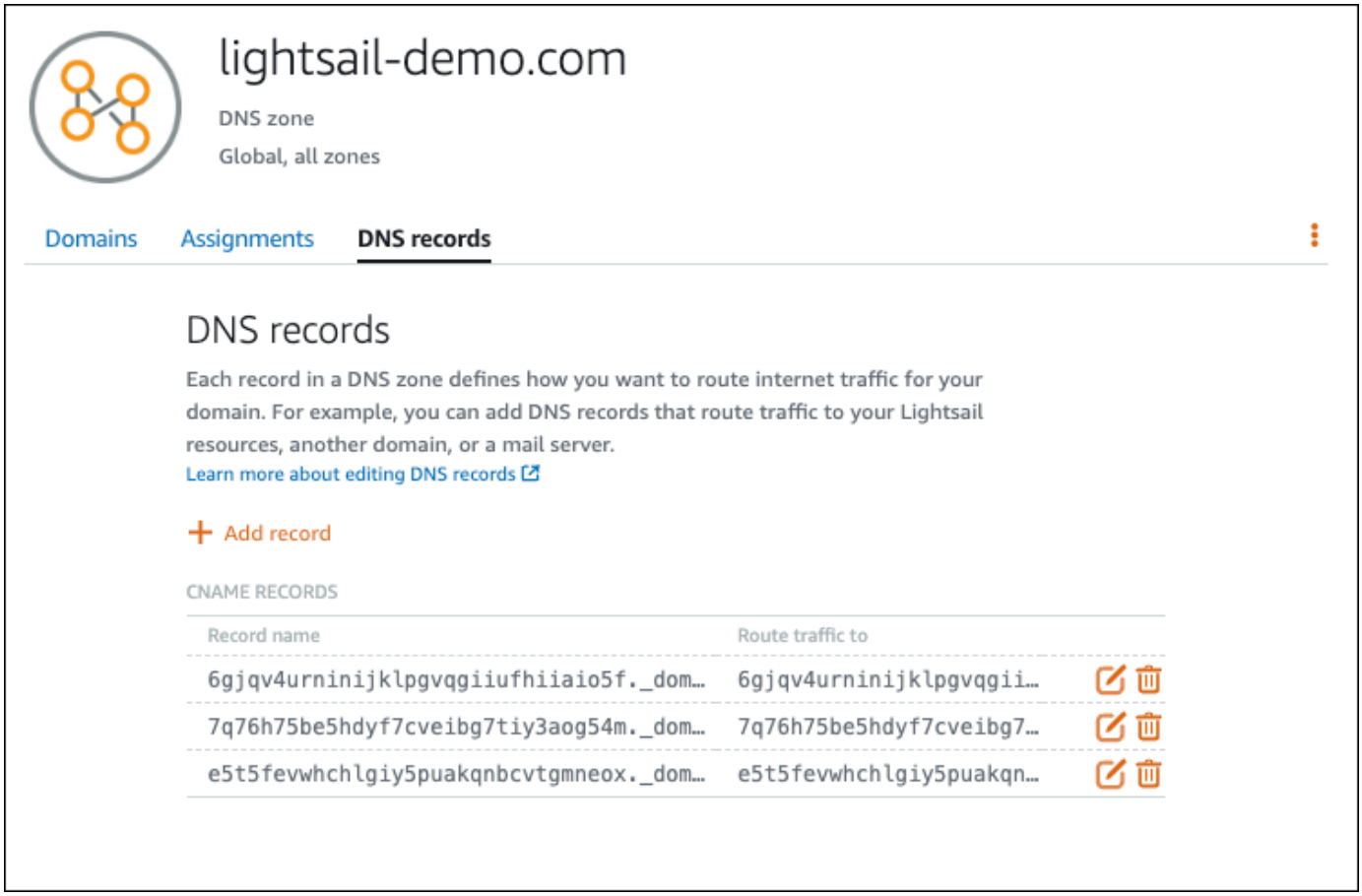
   検証するドメインは、Lightsail の WordPress インスタンスで使用しているドメインと一致している必要があります。

> **Important**
>
> レガシー TXT レコード
> Amazon SES のドメイン検証は、現在は DomainKeys Identified Mail (DKIM) に基づいています。これは受信メールサーバーがメールの信頼性を検証するために使用するメール認証規格です。ドメインの DNS 設定で DKIM を設定すると、ユーザーがアイデンティティの所有者であることが SES に確認されるため、TXT レコードは不要になります。TXT レコードを使用して検証されたドメインアイデンティティは再検証する必要はありませんが、DKIM 準拠のメールプロバイダーでメールを配信しやすくするため、DKIM 署名を有効にすることをお勧めします。


Create identity
A verified identity is a domain, subdomain, or email address you use to send email through Amazon SES. Identity verification at the domain level extends to all email addresses under one verified domain identity.
Identity details Info
Identity type
Domain
To verify ownership of a domain, you must have access to its DNS settings to add the necessary records.
Email address
To verify ownership of an email address, you must have access to its inbox to open the verification email.
Domain
lightsail-demo.com
Domain name can contain up to 253 alphanumeric characters.
Assign a default configuration set
Enabling this option ensures that the assigned configuration set is applied to messages sent from this identity by default whenever a configuration set isn't specified at the time of sending.
Use a custom MAIL FROM domain
Configuring a custom MAIL FROM domain for messages sent from this identity enables the MAIL FROM address to align with the From address. Domain alignment must be achieved in order to be DMARC compliant.
Verifying your domain
DKIM-based domain verification
DomainKeys Identified Mail (DKIM) is an email authentication method that Amazon SES uses to verify domain ownership and that receiving mail servers use to validate email authenticity. You must configure DKIM as part of the domain verification process.
Configuring DKIM
Following identity creation, Amazon SES will provide a set of DNS records. These records must be published to your domain's DNS server in order to successfully configure DKIM and verify ownership of your domain. For more information, see Verifying a domain with Amazon SES.
If your domain is registered with Amazon Route 53, Amazon SES will automatically update your domain's DNS server with the necessary records. This can be disabled by expanding the Advanced DKIM settings and unchecking Publish DNS records to Route53 in the Easy DKIM selection.
Advanced DKIM settings
Identity type
Easy DKIM
To set up Easy DKIM, you have to modify the DNS settings for your domain.
Provide DKIM authentication token (BYODKIM)
Configure DKIM for this domain by providing your own private key.

4. Easy DKIM でドメインアイデンティティを作成した後、ドメインの DNS プロバイダーに公開するため、次の生成した CNAME レコードをコピーして DKIM 認証での検証プロセスを完了する必要があります。これらのレコードの検出には最大 72 時間かかる場合があります。詳細については、「DKIM でドメインアイデンティティの検証」および「Easy DKIM」を参照してください。
5. 新しいブラウザタブを開き、Lightsail コンソールに移動します。
6. 左側のナビゲーションペインで [ドメインと DNS] を選択し、ドメインの DNS ゾーンを選択します。
7. Amazon SES コンソールから DNS レコードを追加します。Lightsail のDNS ゾーンの編集の詳細については、「Edit a DNS zone in Amazon Lightsail の DNS ゾーンを編集する」を参照してください。

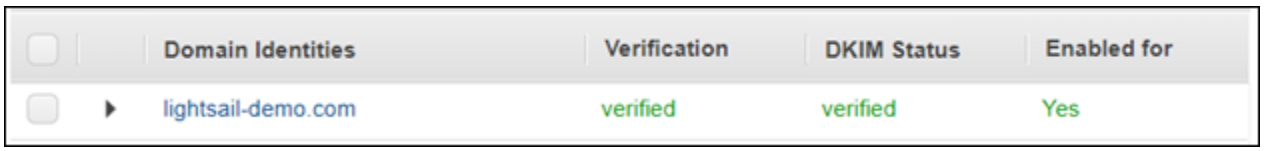
   結果は次の例のようになります。



> **Note**
>
> [サブドメイン] テキストボックスに @ のシンボルを入力して、MX レコードにドメインの頂点を使用します。また、Amazon SES で指定された MX レコード値は `10 inbound-smtp.us-west-2.amazonaws.com` になります。`10` を [Priority] (優先)、および `inbound-smtp.us-west-2.amazonaws.com` を [Maps to] (マップ先) としてドメインに入力します。

8. Amazon SES コンソールで、[新しいドメインを検証する] ページを閉じます。

   数分後、以下の例のように、Amazon SES コンソールに表示されるドメインには検証済みのラベルが付き、送信できるようになります。



Domain Identities	Verification	DKIM Status	Enabled for
lightsail-demo.com	verified	verified	Yes

   Amazon SES の SMTP サービスは、ドメインからメールを送信する準備ができました。

ステップ 5: Amazon SES のメールアドレスを検証する

Amazon SES の新規ユーザーの場合は、メールを送信する宛先のメールアドレスを検証する必要があります。これを行うには、Amazon SES コンソールにメールアドレスを追加します。詳細については、Amazon SES ドキュメントの「Amazon SES のメールアドレスの検証」を参照してください。

WordPress ウェブサイトの管理者のメールアドレスを追加することをお勧めします。こうすることで WordPress 管理者はユーザープロファイルのパスワードリセットをリクエストできます。またブログの投稿、ウェブサイトの更新、その他のプラグインメッセージに関する E メール通知を受信できます。

> **Note**
>
> 検証なしで任意のアドレスにメールを送信する場合は、Amazon SES アカウントをサンドボックスの外に移動するようリクエストする必要があります。詳細については、Amazon SES ドキュメントの「Amazon SES サンドボックスの外への移動」を参照してください。

E メールアドレスの ID を作成するには

1. Amazon SES コンソールで、左ナビゲーションメニューから [検証済み ID] を選択します。
2. [ID の作成] を選択します。
3. [メールアドレス] を選択します。次に、検証するメールアドレスを入力します。
4. [ID の作成] を選択します。

検証するメールアドレスすべてに対し、ステップ 1～4 を繰り返します。確認メールが入力したメールアドレスに送信されます。アドレスが「検証待ち」ステータスで検証済みの E メール ID のリストに追加されます。ユーザーが E メールメッセージを開いて検証プロセスを完了すると、ステータスは「検証済み」と表示されます。

E メールアドレス ID を検証するには

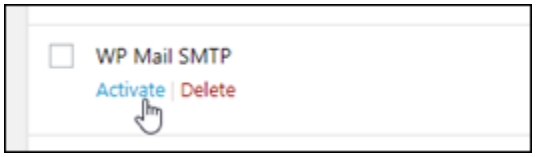
1. ID の作成に使用したメールアドレスの受信トレイを確認し、no-reply-aws@amazon.com からのメールを探します。
2. E メールを開き、E メールのリンクをクリックして、E メールアドレスの検証プロセスを完了します。完了したら、ID の状態が検証済みに更新されます。

ステップ 6: WordPress インスタンスに WP Mail SMTP プラグインを設定する

最後のステップでは、WordPress インスタンスに WP Mail SMTP プラグインを設定します。Amazon SES コンソールで、このガイドで先に作成した SMTP 認証情報を使用します。

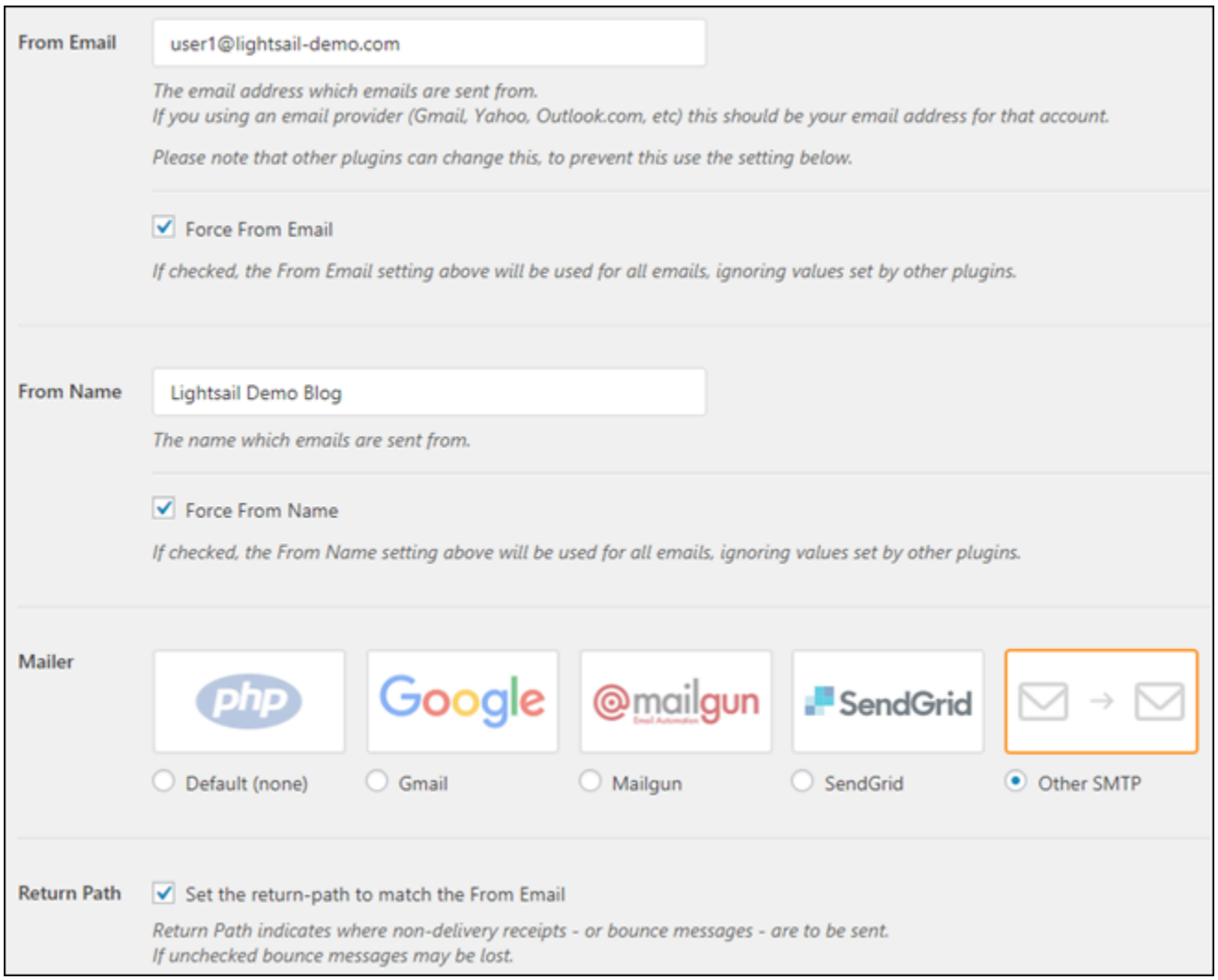
WordPress インスタンスで WP Mail SMTP プラグインを設定するには

1. 管理者として WordPress ウェブサイトのダッシュボードにサインインします。
2. 左側のナビゲーションメニューから、[プラグイン] を選択し、続いて [Installed Plugins (インストール済みのプラグイン)] を選択します。
3. WP Mail SMTP プラグインまで下にスクロールし、[有効化] を選択します。プラグインの新しいバージョンがある場合は、次のステップに進む前に更新してください。

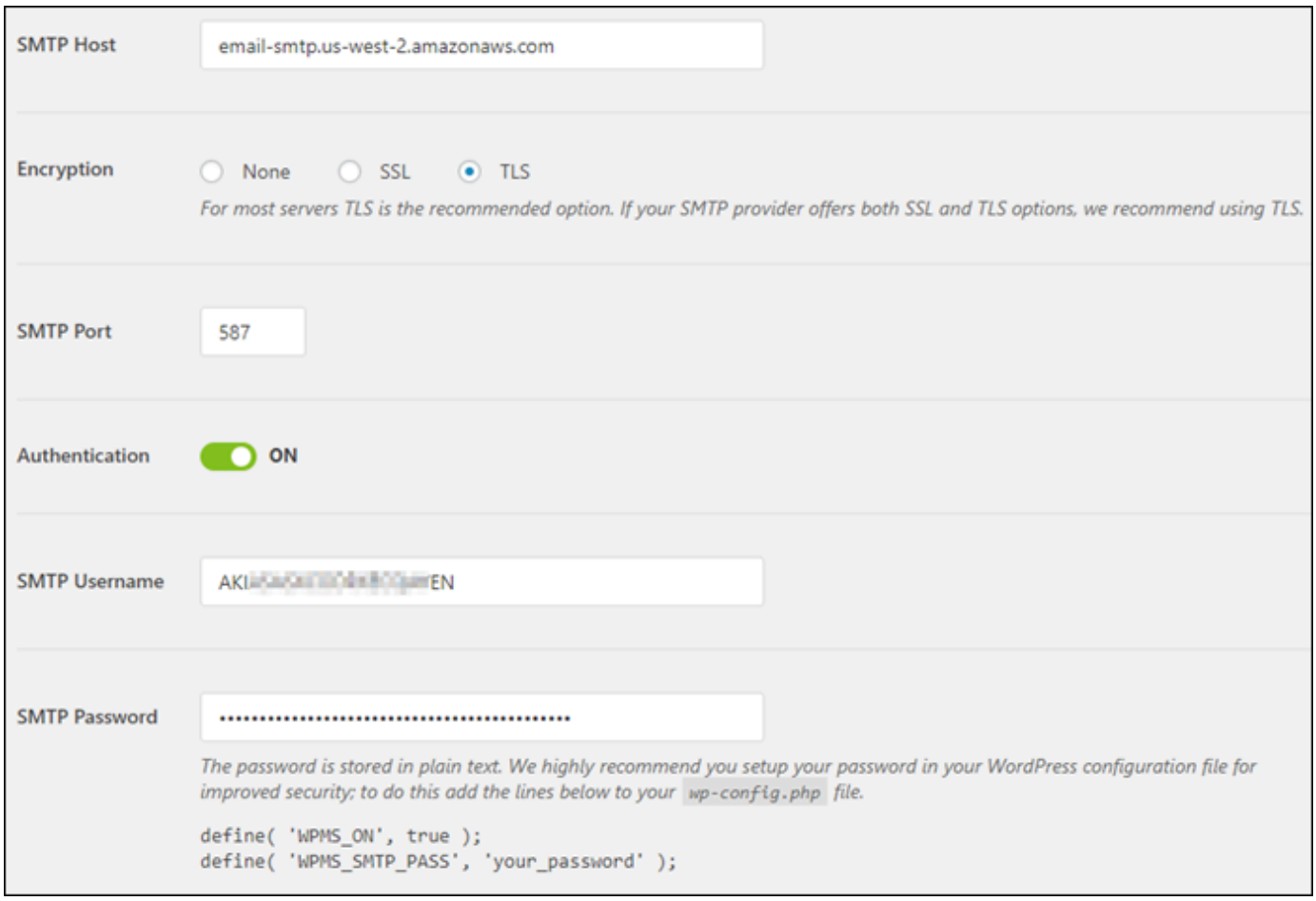


4. WP Mail SMTP プラグインが有効になったら、[設定] を選択します。下にスクロールしてプラグインを見つける必要がある場合があります。

5. [送信元メールアドレス] テキストボックスに、メールの送信元のメールアドレスを入力します。入力するメールアドレスは、このガイドの先のステップを使用して、Amazon SES で確認されている必要があります。
6. [メールから実行] を選択して、[送信元メールアドレス] テキストボックスで入力するメールアドレスを使用して実行し、他のプラグインで設定された「送信元メールアドレス」値を無視します。
7. [From Name (送信元名)] テキストボックスに E メールを送信する送信元の名前を入力するか、そのままにして WordPress ブログの名前を使用します。
8. [Force From Name (名前から実行)] を選択して、[From Name (送信元名)] テキストボックスに入力した名前を使用して実行します。このオプションを選択すると他のプラグインによって設定された「送信元名」値は無視され、WordPress は [From Name (送信元名)] テキストボックスに入力する名前を使用します。
9. ページのメーラーセクションで、[Other SMTP mailer (その他の SMTP メーラー)] を選択します。
10. [Return-Path を送信元メールアドレスに一致するよう設定する] を選択して、配信不能レシートが [送信元メールアドレス] テキストボックスで入力するメールアドレスに送信されるように設定します。



11. [SMTP ホスト] テキストボックスに、このガイドの前半で Amazon SES コンソールの [SMTP の設定] ページから取得した、SMTP サーバー名を入力します。
12. このページの [暗号化] セクションで [TLS] を選択して、Amazon SES の SMTP サービスが TLS 暗号化を使用していることを確認します。
13. [SMTP ポート] テキストボックスは、デフォルトの値 [587] のままにしておきます。
14. [認証] トグルを [オン] に切り替え、このガイドの前半で Amazon SES コンソールから取得した、SMTP ユーザー名とパスワードを入力します。

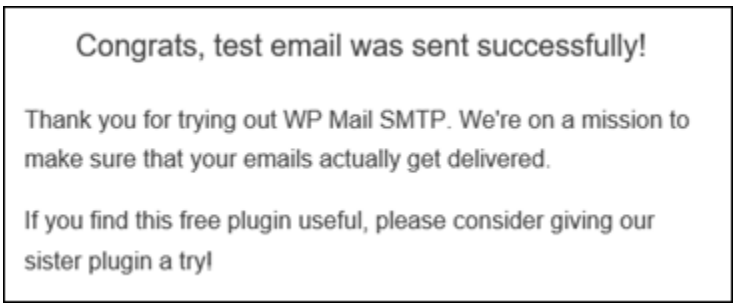


15. [Save settings (設定を保存)] を選択します。設定が正常に保存されたことを確認するプロンプトが表示されます。
16. [Email Test (E メールテスト)] タブを選択します。

    次のステップで、テスト用の E メールを送信して E メールサービスが動作していることを確認します。
17. [送信先] テキストボックスにメールアドレスを入力し、[メールの送信] を選択します。入力するメールアドレスは、このガイドの先のステップを使用して、Amazon SES で確認されている必要があります。

    2 つの可能な結果が表示されるはずです。

    - 成功の確認が表示された場合は、WordPress ウェブサイトは E メールが有効になっています。以下のテスト E メールが指定されたメールボックスに到達することを確認します。



WordPress ウェブサイトのサインインページで [Lost your password? (パスワードを忘れましたか?)] が選択できるようになりました。Amazon SES で WordPress ユーザープロファイルのメールアドレスが確認されると、新しいパスワードがメールで送信されます。

- 失敗通知が表示された場合は、WP Mail SMTP プラグインに入力した SMTP 設定が、Amazon SES アカウントの SMTP サービスのものと一致していることを確認します。また、Amazon SES で検証したメールアドレスを使用していることを確認します。

Lightsail で HTTPS を使用して WordPress サイトを保護する

WordPress のウェブサイトでハイパーテキスト転送プロトコルセキュア (HTTPS) を有効にすることによって、訪問者がウェブサイトが安全であること、暗号化されたデータを送受信していることを保証します。セキュリティで保護されていないウェブサイトのアドレスは `http://example.com` などの、`http` で始まり、セキュリティで保護されたウェブサイトのアドレスは `https://example.com` などの `https` で始まります。ウェブサイトが主に情報提供を目的としたものでも、HTTPS を有効にすることをお勧めします。これは、HTTPS が有効になっていない場合、ほとんどのウェブブラウザがウェブサイトの訪問者にウェブサイトが安全でないことを通知し、その結果ウェブサイトの検索エンジンの結果でランクが下がるためです。

> **Tip**
>
> Lightsail は、WordPress インスタンスでの SSL/TLS Let's Encrypt 証明書のインストールと設定を自動化するガイド付きワークフローを提供します。このチュートリアルの手動手順を実行する代わりに、このワークフローを使用することが強く推奨されます。詳細については、「Launch and configure a WordPress instance」を参照してください。

このガイドでは、Bitnami HTTPS 設定ツール (`bncert`) を使用して、HTTPS をAmazon LightsailのBitnam により保証された WordPress インスタンスで有効化する方法を説明します。これは、リクエスト時に指定するドメインおよびサブドメインに対してのみ証明書を要求することを許可します。ま

た Certbot を使用して、ドメインに証明書を、そしてサブドメインにワイルドカード証明書をリクエストできます。ワイルドカード証明書はドメインのすべてのサブドメインに使用できます。これは、トラフィックをインスタンスに誘導するために使用するサブドメインがどれかわからない場合に役立ちます。ただし、bncert ツールと違い、Certbot は証明書を自動的に更新しません。Certbot を使用する場合は、90 日ごとに証明書を手動で更新する必要があります。Certbot を使用して HTTPS を有効にする方法の詳細については、「チュートリアル: WordPress インスタンスで Let's Encrypt の SSL 証明書を使用する」を参照してください。

目次

- ステップ 1: プロセスについて学ぶ
- ステップ 2: 前提条件を完了させる
- ステップ 3: インスタンスに接続する
- ステップ 4: インスタンスに bncert ツールがインストールされていることを確認
- ステップ 5: WordPress インスタンスで HTTPS を有効にする
- ステップ 6: ウェブサイトで HTTPS を使用しているかテストする

ステップ 1: プロセスについて学ぶ

> **Note**
>
> このセクションでは、プロセスの高度な概要を説明します。このプロセスを実行する具体的なステップについては、このガイドの以降のステップで説明します。

WordPress のウェブサイトで HTTPS を有効にするには、SSH を使用してLightsail インスタンスに接続して、bncert ツールを使用して SSL/TLS 証明書を Let's Encrypt 認証機関から要求します。証明書をリクエストする際は、ウェブサイトのプライマリドメイン (example.com) や代替ドメイン (www.example.com や blog.example.com など) を指定します。Let's Encrypt は、ドメインの DNS で TXT レコードを作成するように求めるか、またはそれらのドメインがリクエスト元のインスタンスのパブリック IP アドレスにトラフィックをすでに送信していることを確認することによって、ドメインを所有していることを確認します。

証明書が検証されたら、WordPress のウェブサイトを構成して、HTTP から HTTPS へ訪問者を自動的にリダイレクト (http://example.com から https://example.com にリダイレクト) し、訪問者が暗号化された接続を使用するように強制します。また、www サブドメインをドメインの

頂点 (`https://www.example.com` を `https://example.com` にリダイレクト) またはその逆 (`https://example.com` を `https://www.example.com` にリダイレクト) に自動的にリダイレクトするようにウェブサイトを設定することもできます。これらのリダイレクトは、`bncert` ツールを使って設定することもできます。

Let's Encrypt では、ウェブサイトで HTTPS を維持するために 90 日ごとに証明書を更新する必要があります。`bncert` ツールは証明書を自動的に更新するので、ウェブサイトに専念する時間を増やすことができます。

bncert ツールの制限事項

`bncert` ツールには次の制約事項があります。

- Certified by Bitnam WordPress インスタンスの作成時にプリインストールされていません。Lightsail で作成されてしばらく経っている WordPress のインスタンスでは、`bncert` ツールを手動でインストールする必要があります このガイドのステップ 4 は、ツールがインスタンスにインストールされていることを確認する方法と、されていない場合にインストールする方法を示します。
- 証明書をリクエストできるのは、リクエスト時に指定したドメインおよびサブドメインに対してのみです。ドメインの証明書とサブドメインのワイルドカード証明書のリクエストを可能にする Certbot ツールとは異なります。ワイルドカード証明書はどのサブドメインにも使用できます。これは、トラフィックをインスタンスに誘導するために使用するサブドメインがわからない場合に役立ちます。ただし、`bncert` ツールと違い、Certbot は証明書を自動的に更新しません。Certbot を使用する場合は、90 日ごとに証明書を手動で更新する必要があります。Certbot を使用して HTTPS を有効にする方法の詳細については、「チュートリアル: WordPress インスタンスで Let's Encrypt の SSL 証明書を使用するAmazon Lightsail」を参照してください。

ステップ 2: 前提条件を完了させる

以下の前提条件を満たします (まだ満たしていない場合)。

- Lightsail で WordPress インスタンスを作成し、インスタンスと動作するようにウェブサイトを設定します。詳細については、「Amazon Lightsail で Linux/Unix ベースのインスタンスの使用を開始する」を参照してください。
- 静的 IP をインスタンスに添付します。インスタンスを停止してまた開始すると、インスタンスのパブリック IP アドレスは変わります。インスタンスを停止して開始しても、静的 IP は変更されません。詳細については、「静的 IP を作成して Amazon Lightsail インスタンスにアタッチする」を参照してください。

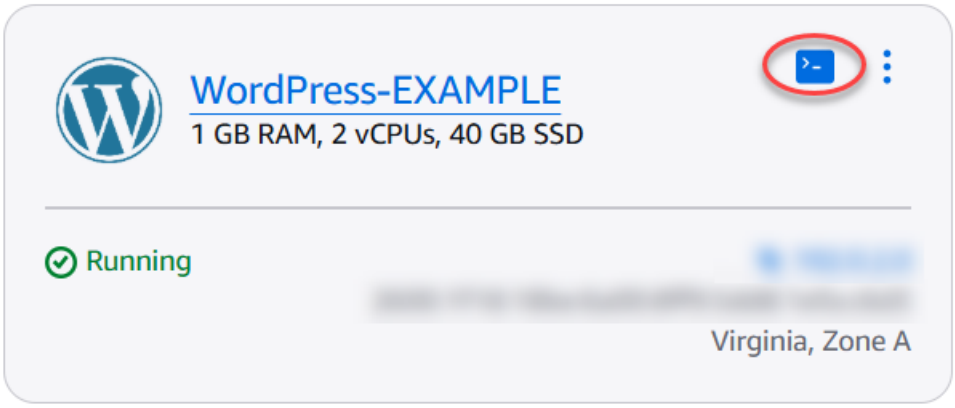
- WordPress インスタンスの設定が完了したら、そのスナップショットを作成するか、自動スナップショットを有効にします。スナップショットは、インスタンスに何か問題が発生した場合、これを元に別のインスタンスを作成できるバックアップとして使用できます。詳細については、「Linux または Unix インスタンスのスナップショットを作成する」または「Amazon Lightsail のインスタンスまたはディスクの自動スナップショットの有効化と無効化」を参照してください。
- ドメインの頂点 (`example.com`) とその `www` サブドメイン (`www.example.com`) のトラフィックを、Lightsail のWordPress インスタンスのパブリックIPアドレスに誘導するDNSレコードを、ドメインのDNSに追加します。これらのアクションは、ドメインの現在の DNS ホスティングプロバイダーで実行することができます。ドメインの DNS の管理を Lightsail に引き渡した場合は、Lightsail の DNS ゾーンを使用してこれらのアクションを完了できます。詳細については、「DNS」を参照してください。

> **Important**
>
> WordPress のウェブサイトで使用するすべてのドメインの DNS に DNS レコードを追加します。これらのドメインはすべて、WordPress ウェブサイトのパブリック IP アドレスにトラフィックをルーティングしている必要があります。`bncert` ツールは、WordPress インスタンスのパブリック IP アドレスに現在トラフィックを誘導しているドメインに対してのみ証明書を発行します。

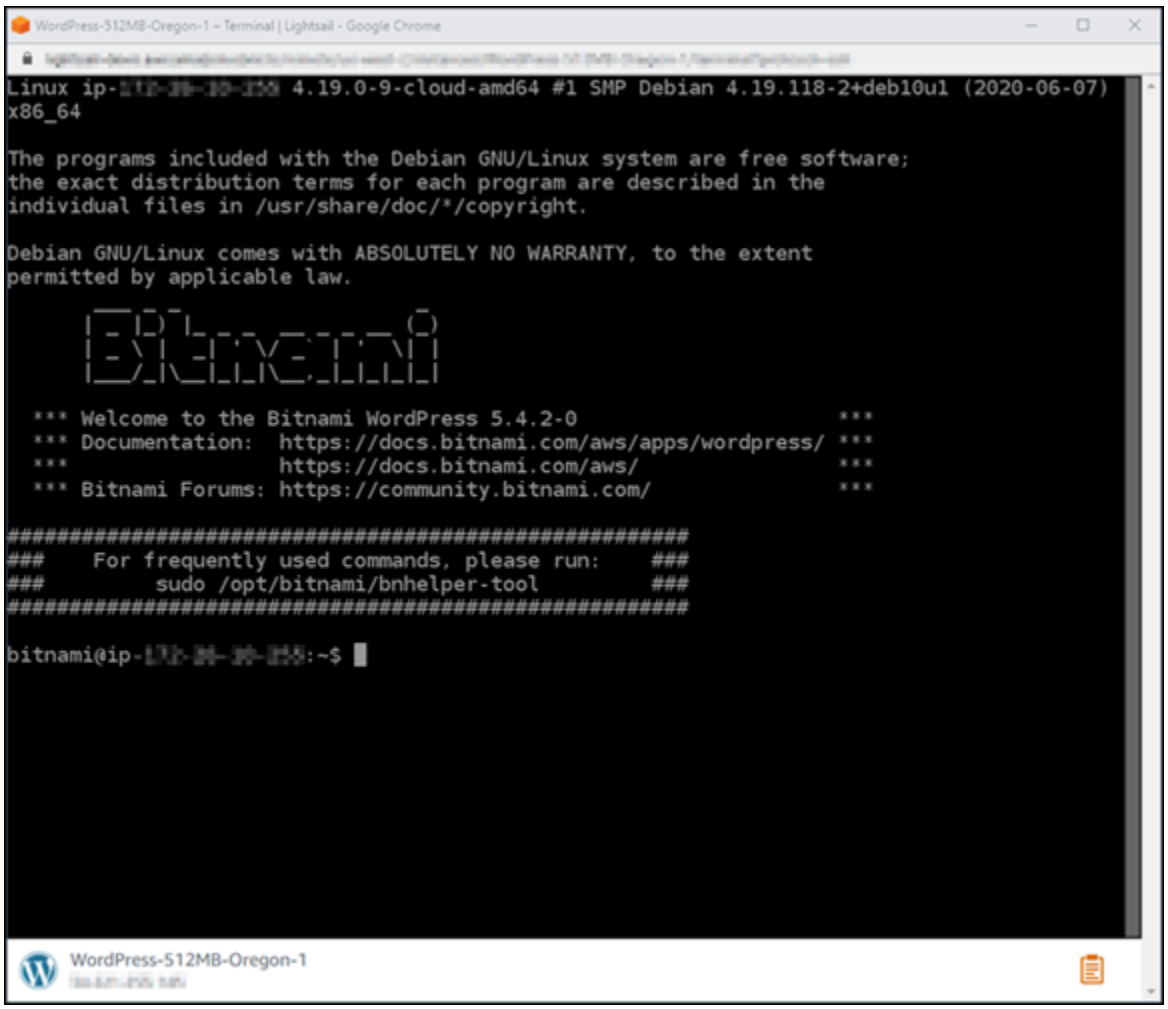
ステップ 3: インスタンスに接続する

次のステップに従って、Lightsail コンソールでブラウザベースの SSH クライアントを使用して WordPress インスタンスに接続します。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、WordPress インスタンスの SSH クイック接続アイコンを選択します。



ブラウザベースの SSH クライアントターミナルウィンドウが開きます。SSH 経由でインスタンスに正常に接続されていると、次の例に示すように Bitnami 口ゴが表示されます。



ステップ 4: インスタンスに bncert ツールがインストールされていることを確認

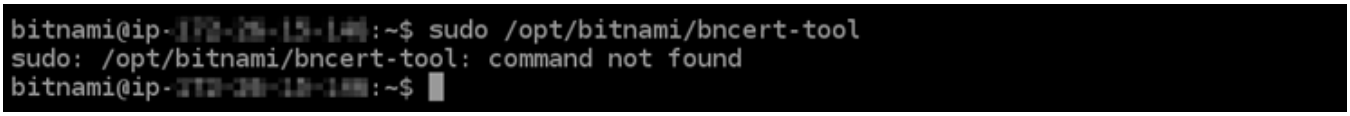
次のステップを完了して Bitnami HTTPS 設定ツール (bncert) がインスタンスにインストールされていることを確認します。このツールは Certified by Bitnam WordPress インスタンスの作成時にプリインストールされません。しばらく前に Lightsail で作成されたWordPress のインスタンスは、bncert ツールを手動でインストールすることを必要とします。このステップでは、ツールがインストールされていない場合にツールをインストールする方法を説明します。

1. bncert ツールを実行するには、次のコマンドを入力します。

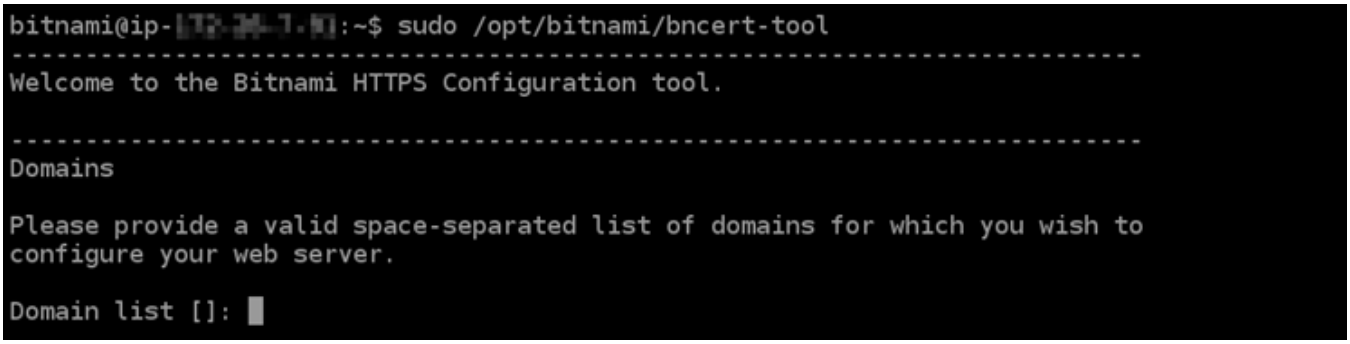
```
sudo /opt/bitnami/bncert-tool
```

- 次の例に示すように、command not found が応答で表示された場合、これは bncert ツールがインストールされていないことを示します。このステップの次のステップに進み、bncert ツールをインスタンスにインストールします。

> **Important**
>
> bncert ツールを使用できるのは、Certified by Bitnam の WordPress インスタンスのみです。または、Certbot ツールを使用して、WordPress インスタンスで HTTPS を有効化することもできます。詳細に関しては、「チュートリアル: WordPress インスタンスで Let's Encrypt の SSL 証明書を使用する」を参照してください。



- 次の例に示すように、Welcome to the Bitnami HTTPS configuration tool がレスポンスで表示された場合は、bncert ツールがインストールされていることを示します。本ガイドの、「ステップ 5: WordPress インスタンスで HTTPS を有効にする」セクションに進んでください。



2. 以下のコマンドを入力して、bncert 実行ファイルをインスタンスにダウンロードします。

```
wget -O bncert-linux-x64.run https://downloads.bitnami.com/files/bncert/latest/bncert-linux-x64.run
```

3. 以下のコマンドを入力して、bncert 実行ファイルへのディレクトリを作成します。

```
sudo mkdir /opt/bitnami/bncert
```

4. 以下のコマンドを入力して、ダウンロードした bncert 実行ファイルを、作成した新しいディレクトリに移動させます。

```
sudo mv bncert-linux-x64.run /opt/bitnami/bncert/
```

5. 以下のコマンドを入力して、プログラムとして実行できるファイルを bncert に実行させます。

```
sudo chmod +x /opt/bitnami/bncert/bncert-linux-x64.run
```

6. 次のコマンドを入力することによって、sudo /opt/bitnami/bncert-tool コマンドを入力すると bncert ツールを実行するシンボリックリンクを作成します。

```
sudo ln -s /opt/bitnami/bncert/bncert-linux-x64.run /opt/bitnami/bncert-tool
```

これで bncert ツールのインストールは完了しました。本ガイドの「ステップ 5: WordPress インスタンスで HTTPS を有効にする」のセクションに進んでください。

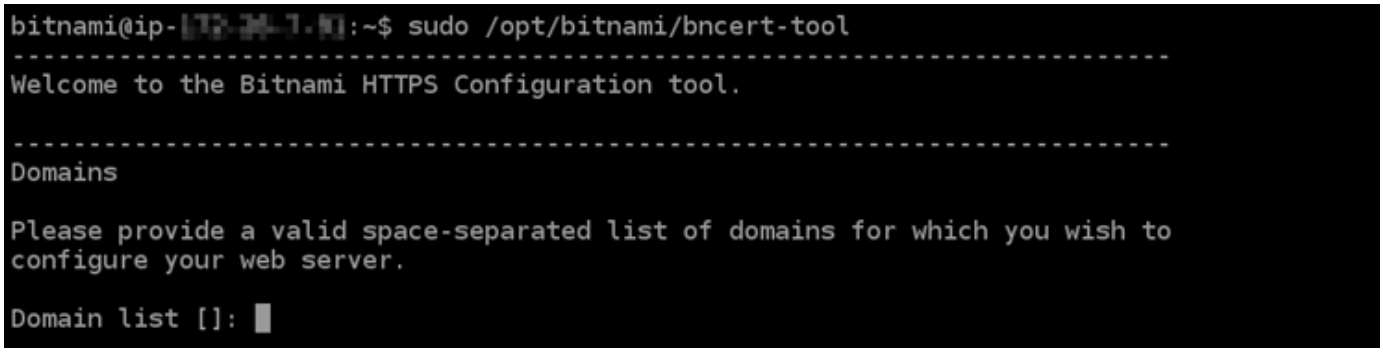
ステップ 5: WordPress インスタンスで HTTPS を有効にする

bncert ツールがインスタンスにインストールされていることを確認したあと、次のステップを実行して WordPress インスタンスで HTTPS を有効にします。

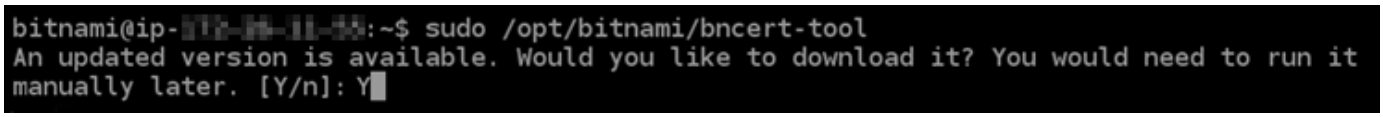
1. bncert ツールを実行するには、次のコマンドを入力します。

```
sudo /opt/bitnami/bncert-tool
```

次の例に示すようなメッセージが表示されます。

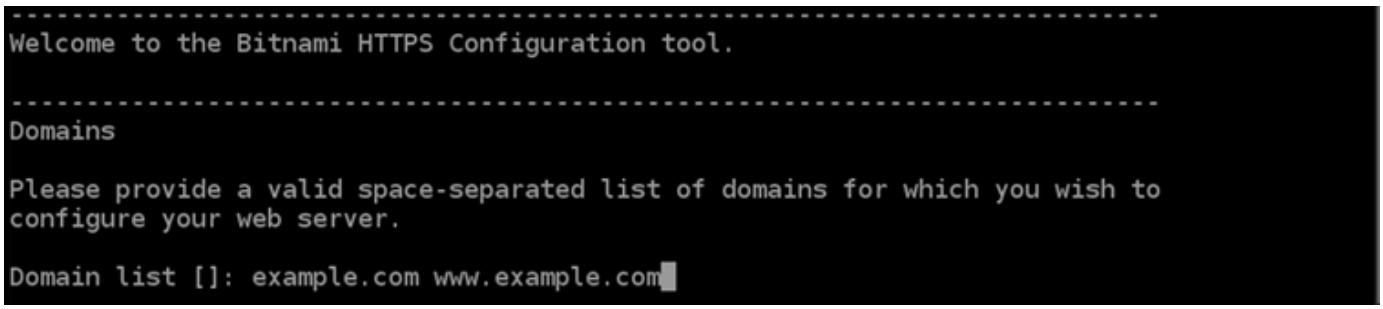


bncert ツールがしばらく前にインスタンスにインストールされていると、ツールの更新バージョンが利用可能であることを示すメッセージが表示される場合があります。次の例に示すように、ダウンロードすることを選択し、sudo /opt/bitnami/bncert-tool コマンドを入力して bncert ツールを再度実行します。



2. 次の例に示すように、プライマリドメイン名と代替ドメイン名の間はスペースで区切って入力します。

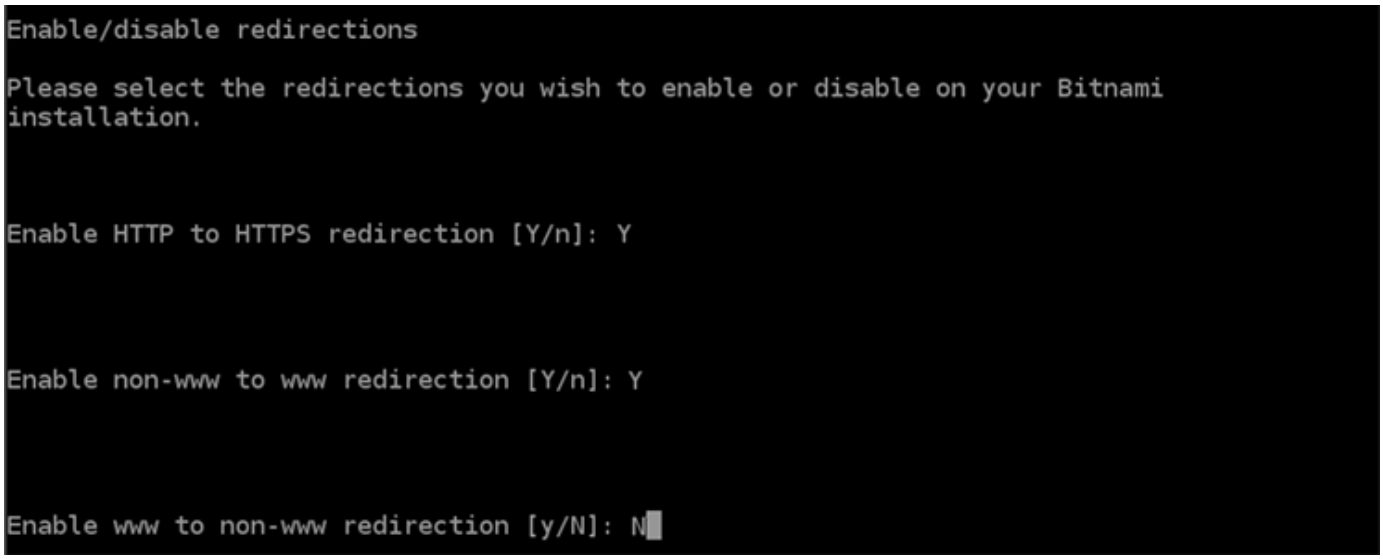
ドメインがインスタンスのパブリック IP アドレスにトラフィックをルーティングするように設定されていない場合、bncert ツールは、続行する前にその設定を行うように要求します。ドメインは、bncert ツールを使用して HTTPS を有効にしているインスタンスでのパブリック IP アドレスにトラフィックをルーティングする必要があります。これはドメインを所有していることを確認し、証明書の検証として機能します。



3. bncert ツールは、ウェブサイトのリダイレクトの設定方法を尋ねます。使用できるオプションは次のとおりです。

- HTTP から HTTPS へのリダイレクトを有効にする - HTTP バージョンのウェブサイトを閲覧するユーザー (例: `http:/example.com`) を自動的に HTTPS バージョン (例: `https://example.com`) にリダイレクトするかどうかを決定します。すべての訪問者が暗号化された接続を使用するように強制されるため、このオプションを有効にすることをお勧めします。Y を入力して Enter を押すと、有効になります。
- www なしから www ありへのリダイレクトの有効化 - ドメインの頂点 (例: `https://example.com`) まで閲覧するユーザー を自動的にドメインの www サブドメイン (例: `https://www.example.com`) にリダイレクトするかを指定します。このオプションを有効にすることをお勧めします。ただし、ドメインの頂点を Google のウェブマスターツールなどの検索エンジンツールで希望のウェブサイトアドレスとして指定した場合、または頂点が IP を直接指しており、www のサブドメインが CNAME レコードを介してリファレンスしている場合は、無効にして代替オプションを有効にすることをお勧めします (www ありから www なしへのリダイレクトを有効化) 。Y を入力し、Enter を押して有効にします。
- www ありから www なしへのリダイレクトを有効にする - ドメインの www サブドメイン (例: `https://www.example.com`) まで閲覧するユーザーを、自動的にドメインの頂点 (例: `https://example.com`) にリダイレクトするかを指定します。www なしから www ありへのリダイレクトを有効にした場合は、これを無効にすることをお勧めします。N を入力し、Enter を押して無効にします。

選択した結果は次の例のようになります。



4. これから実行される変更が一覧表示されます。Y と入力し、Enter を押して確認し、続行します。

```
Changes to perform

The following changes will be performed to your Bitnami installation:

1. Stop web server
2. Configure web server to use a free Let's Encrypt certificate for the domains:
example.com www.example.com
3. Configure a cron job to automatically renew the certificate each month
4. Configure web server name to: example.com
5. Enable HTTP to HTTPS redirection (example: redirect http://example.com to
https://example.com)
6. Enable non-www to www redirection (example: redirect example.com to
www.example.com)
7. Start web server once all changes have been performed



Do you agree to these changes? [Y/n]: Y
```

5. Let's Encrypt 証明書に関連付けるメールアドレスを入力し、Enter を押します。

```
Create a free HTTPS certificate with Let's Encrypt

Please provide a valid e-mail address for which to associate your Let's Encrypt
certificate.

Domain list: example.com www.example.com

Server name: example.com

E-mail address []:
```

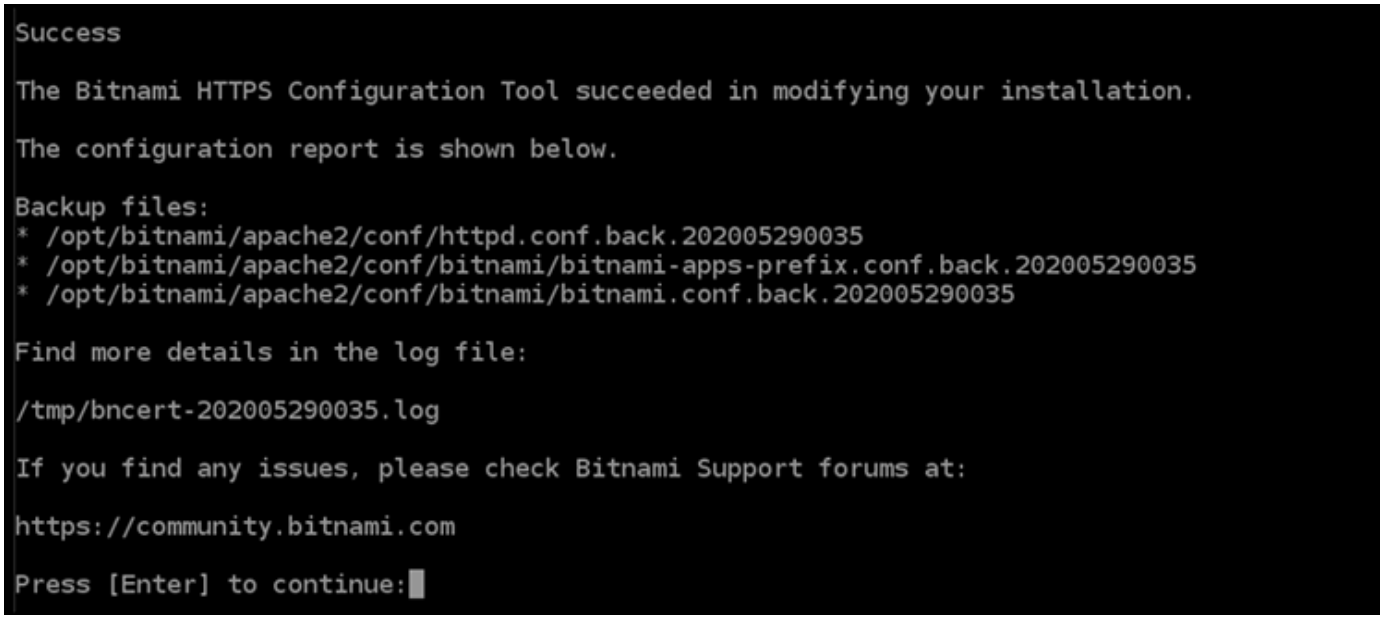
6. Let's Encrypt サブスクライバー合意書を確認します。Y と入力し、Enter を押して契約に同意し、続行します。

```
The Let's Encrypt Subscriber Agreement can be found at:

https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf

Do you agree to the Let's Encrypt Subscriber Agreement? [Y/n]:
```

これらのアクションは、証明書のリクエストや指定したリダイレクトの設定など、インスタンスで HTTPS を有効にするために実行されます。

```
Performing changes to your installation

The Bitnami HTTPS Configuration Tool will perform any necessary actions to your
Bitnami installation. This may take some time, please be patient.

|
```

次の例のようなメッセージが表示された場合は、証明書は正常に発行され、検証され、インスタンスでリダイレクトが正常に設定されています。



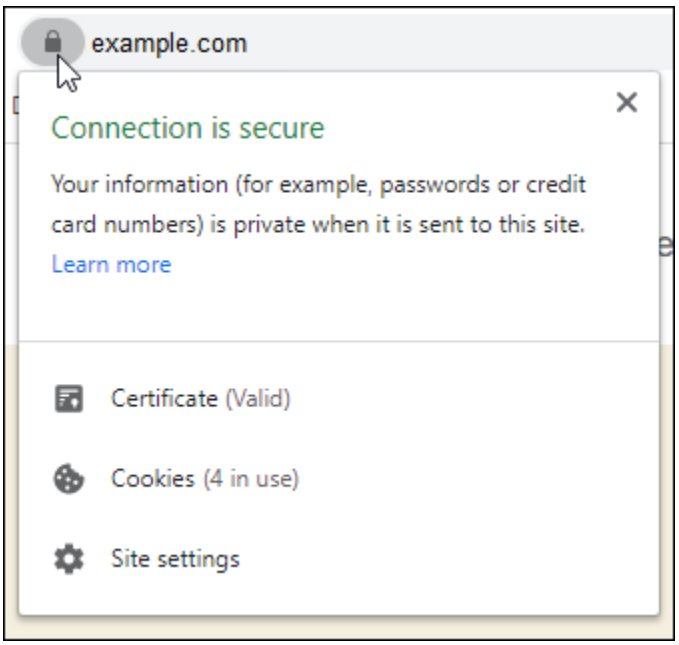
bncert ツールは、有効期限が切れる前、 80 日ごとに証明書の自動更新を実行します。インスタンスで追加のドメインやサブドメインを使用し、それらのドメインで HTTPS を有効にする場合は、上記のステップを繰り返します。

これで、WordPress インスタンスでの HTTPS の有効化が完了しました。本ガイドのステップ 6: ウェブサイトで HTTPS を使用しているかどうかをテストするセクションに進んでください。

ステップ 6: ウェブサイトで HTTPS を使用しているかどうかをテストする

WordPress インスタンスで HTTPS を有効にしたら、bncert ツールを使用して指定したドメインを全て参照し、ウェブサイトが HTTPS を使用していることを確認する必要があります。次の例に示すように、各ドメインにアクセスすると、セキュリティで保護された接続を使用していることがわかります。

> **Note**
>
> 変更を確認するには、ブラウザのキャッシュを更新し、消去する必要がある場合もあります。



bncert ツールの実行時に選択したオプションに応じて、www なしアドレスがドメインの www ありサブドメインへリダイレクトするか、その逆が実行されます。

WordPress ブログを Lightsail に移行する。

WordPress ホスティングプロバイダーを変更する場合に、以下の情報を参考にしてください。Amazon Lightsail は AWSで WordPress サイトを実行する最も簡単な方法です。

当社のいずれかの料金プラン (1 か月あたり 5 米ドルから) を選択すると、プラグイン、テーマなども含めて WordPress のインストールを完全に制御できます。

Lightsail の WordPress インスタンスの作成は数分で済みます。このチュートリアルに従って、既存の WordPress ブログをバックアップし、それを Lightsail で実行する新しいインスタンスにインポートします。

プロセスの簡単な概要を次に示します。

続きを読んで、始めてみましょう。

前提条件

開始するには、以下が必要です。

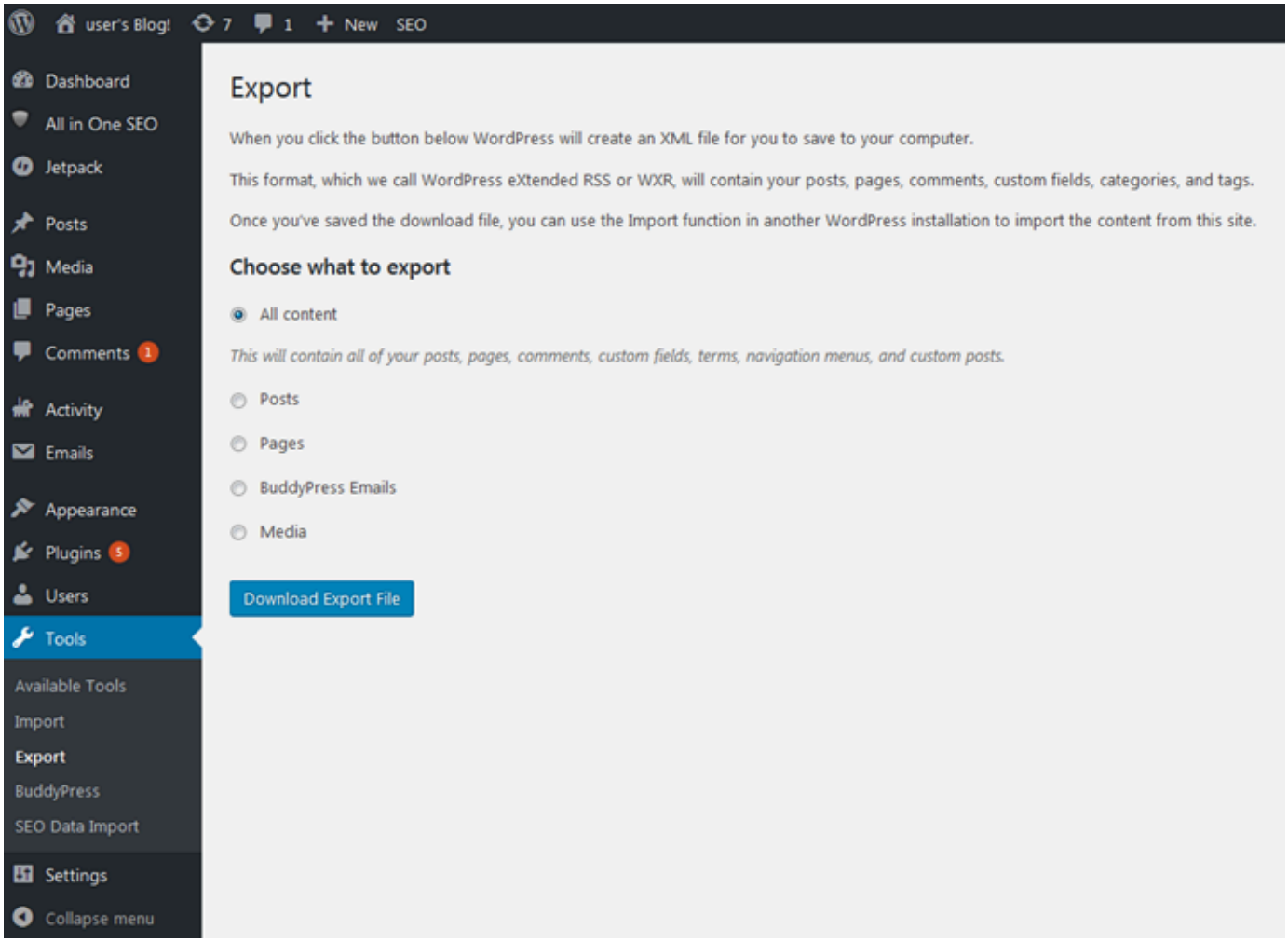
1. AWS アカウントが必要です。にサインアップ AWSするか、既に アカウントをお持ちの場合はにサインイン AWSします。
2. Lightsail を使用するようにアカウントをセットアップします。アカウントを作成してからしばらく経過している場合、またはクレジットカードをまだ提供していない場合は、まず にログイン AWS マネジメントコンソール してアカウントを更新する必要があります。

ステップ 1: 既存の WordPress ブログをバックアップする

WordPress を使用して既存のブログをバックアップできます。WordPress 管理コンソールにログインしてブログを管理できる必要があります。

1. ブログに移動して [管理] を選択します。

   [Manage] (管理) バナーが表示されない場合は、`http://<PublicIP>/wp-login.php` を参照することでサインインページにアクセスすることができます。*<PublicIP>* を、インスタンスのパブリック IP アドレスに置き換えます。
2. ユーザー名とパスワードを入力して WordPress 管理コンソールにログインします。
3. WordPress の [ダッシュボード] で、[ツール]、[エクスポート] の順に選択します。
4. [エクスポート] ページで、[すべてのコンテンツ] を選択して、すべてを XML ファイルとしてエクスポートします。



5. [エクスポートファイルをダウンロード] を選択して、以前のブログを XML ファイルとしてダウンロードします。

   その XML ファイルを見つけやすい場所に保存します。このファイルはステップ 4 で必要になります。

ステップ 2: Lightsail で新しい WordPress インスタンスを作成する

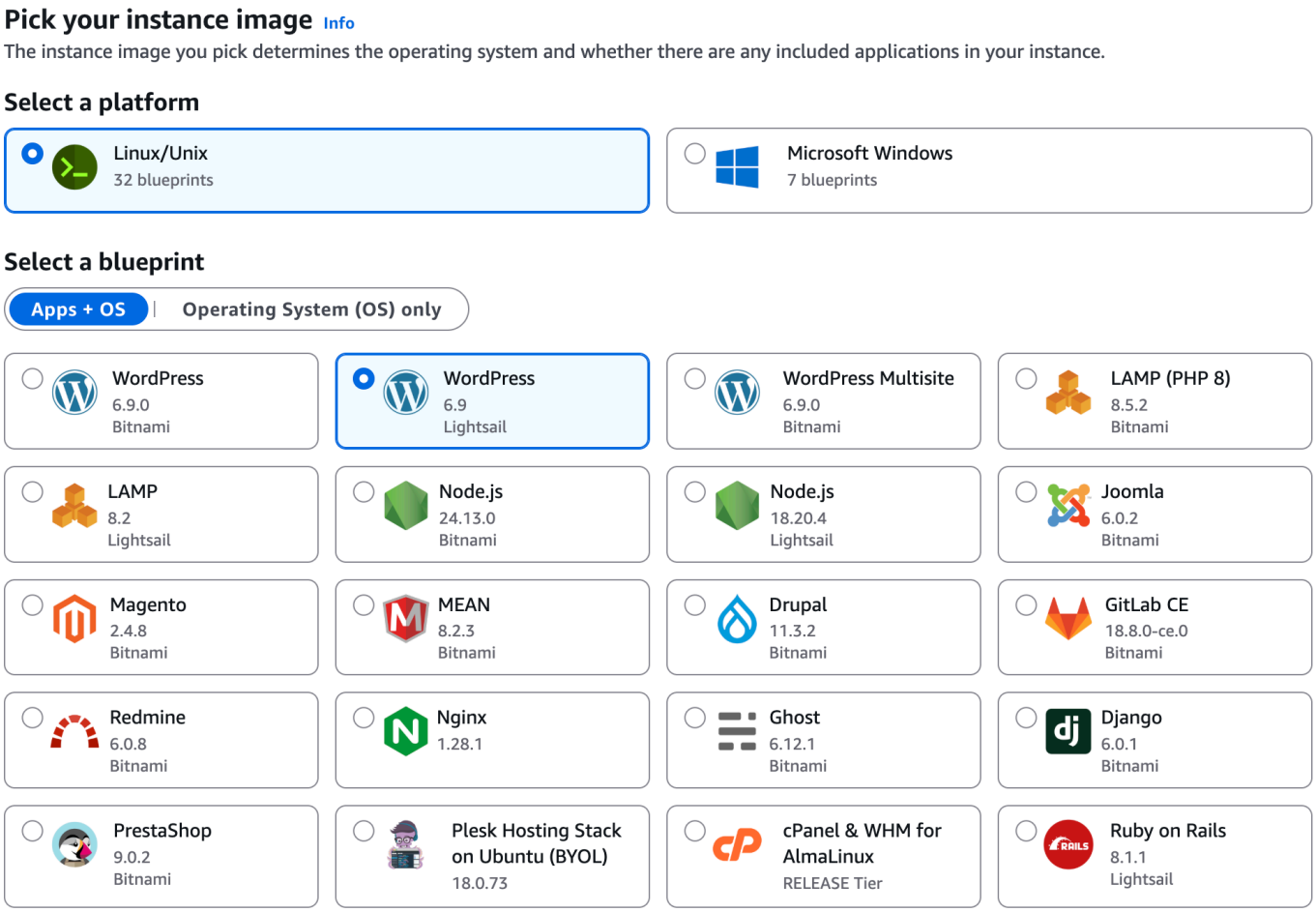
Lightsail では、新しい WordPress インスタンスを数分で作成できます。その方法は次のとおりです。

1. Lightsail のホームページに移動してログインします。
2. [インスタンスの作成] を選択します。
3. ブログを作成する AWS リージョン を選択します。

AWS リージョンを選択したら、デフォルトのアベイラビリティーゾーンを選択または変更できます。

4. WordPress ブループリントを選択します。

   Amazon Lightsail または Bitnami プロバイダーを選択できます。



5. インスタンスプラン (またはバンドル) を選択します。

   Lightsail プランは、後で必要に応じてアップグレードできます。詳細については、「Create an instance from a snapshot in Lightsail」を参照してください。

6. インスタンスの名前を入力します。

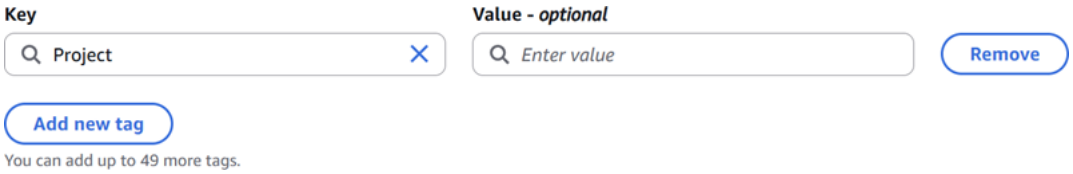
   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2 ~ 255 文字を使用する必要があります。
   - 先頭と末尾は英数字や文字を使用する必要があります。

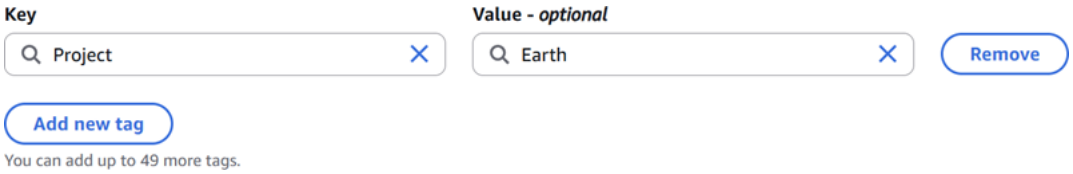
- 英数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

7. (オプション)[新しいタグの追加] を選択してインスタンスにタグを追加します。必要に応じてこの手順を繰り返して、他のタグを追加します。タグの使用の詳細については、「タグ」を参照してください。

   a. [キー] には、タグキーを入力します。

   b. (オプション) [値] にタグの値を入力します。

8. [インスタンスの作成] を選択します。

ステップ 3: Lightsail の新規 WordPress ブログにログインする

Lightsail で新規ブログを作成できたので、WordPress ダッシュボードにアクセスして、以前のブログデータをインポートする必要があります。WordPress ウェブサイトの管理ダッシュボードにサインインするためのデフォルトのパスワードは、インスタンスに保存されます。パスワードを取得するには、次の手順を実行します。

Bitnami

WordPress 管理者のデフォルトパスワードを取得するには

1. WordPress インスタンスのインスタンス管理ページを開きます。
2. [WordPress] パネルで、[デフォルトのパスワードの取得] を選択します。これにより、ページの下部にある [アクセスのデフォルトパスワード] が展開されます。

3. [CloudShell の起動] を選択します。そうすることで、ページの下部に分割されたペインが開きます。
4. [コピー] を選択し、コンテンツを CloudShell ウィンドウに貼り付けます。CloudShell プロンプトにカーソルを置いて Ctrl+V を押すか、右クリックしてメニューを開き、[貼り付ける] を選択します。
5. CloudShell ウィンドウに表示されるパスワードを書き留めます。後で WordPress ウェブサイトの管理ダッシュボードにサインインするには必要です。

```
[cloudshell-user@ip-[redacted] ~]$ AWS_REGION=us-east-1 ~/lightsail_connect WordPress-1 cat bitnami_application_password
JKzhKzhKzhR!
```

Lightsail

WordPress 管理者のデフォルトパスワードを取得するには

1. WordPress インスタンスのインスタンス管理ページを開きます。
2. [WordPress] パネルで、[デフォルトのパスワードの取得] を選択します。これにより、ページの下部にある [アクセスのデフォルトパスワード] が展開されます。

3. [CloudShell の起動] を選択します。そうすることで、ページの下部に分割されたペインが開きます。
4. [コピー] を選択し、コンテンツを CloudShell ウィンドウに貼り付けます。CloudShell プロンプトにカーソルを置いて Ctrl+V を押すか、右クリックしてメニューを開き、[貼り付ける] を選択します。
5. CloudShell ウィンドウに表示されるパスワードを書き留めます。後で WordPress ウェブサイトの管理ダッシュボードにサインインするには必要です。

```
~ $ AWS_REGION=us-east-1 ~/lightsail_connect WordPress-1 tail -n 1 application_credentials
LnPLnPLnPLnPLnP9
```

WordPress ウェブサイトの管理ダッシュボードのパスワードを取得したので、サインインできます。管理ダッシュボードでは、ユーザーパスワードの変更、プラグインのインストール、ウェブサイトのテーマの変更などを行うことができます。

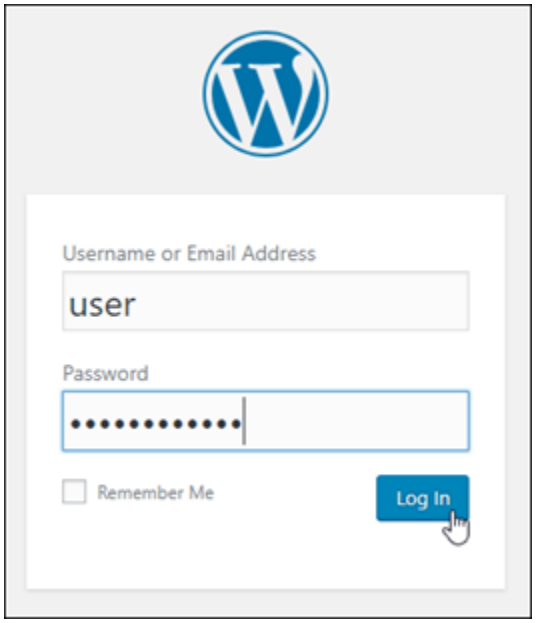
WordPress ウェブサイトの管理ダッシュボードにサインインするには、以下の手順を実行します。

管理ダッシュボードにサインインするには

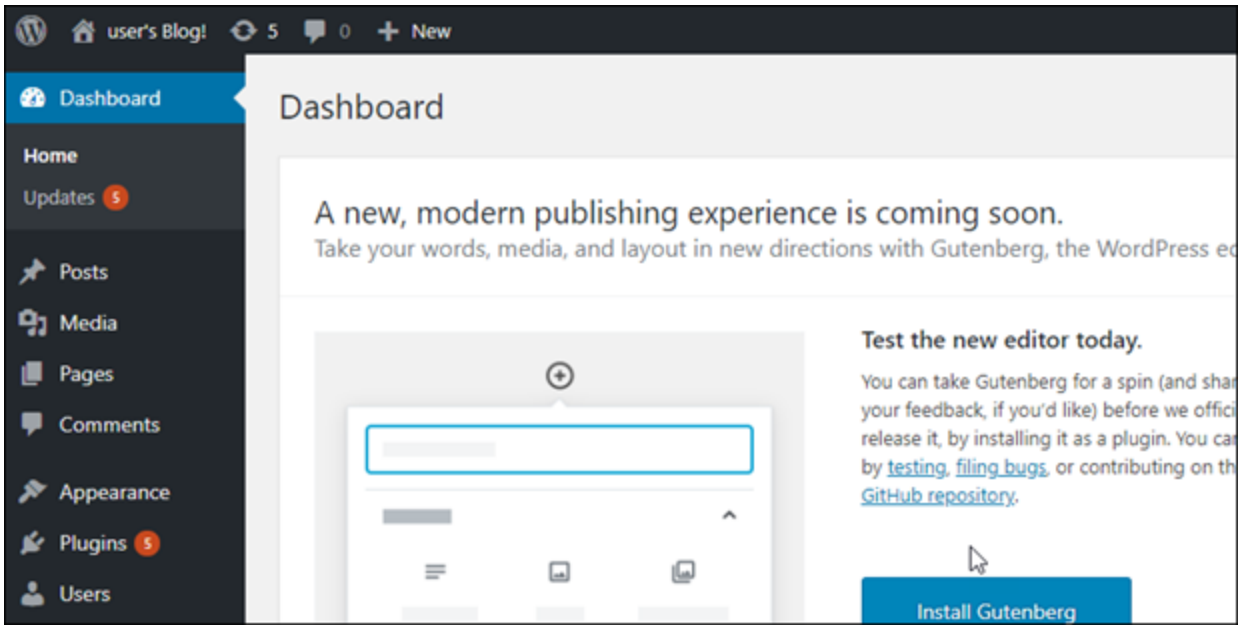
1. WordPress インスタンスのインスタンス管理ページを開きます。
2. [WordPress] パネルで、[WordPress 管理者へのアクセス] を選択します。
3. [WordPress 管理者ダッシュボードへのアクセス] パネルで、[パブリック IP アドレスを使用する] で、次のフォーマットを使用するリンクを選択します:

   http://*public-ipv4-address* ./wp-admin

4. [ユーザー名またはメールアドレス] ボックスに **user** と入力します。
5. [パスワード] には、前の手順で取得したパスワードを入力します。
6. [ログイン] を選択します。



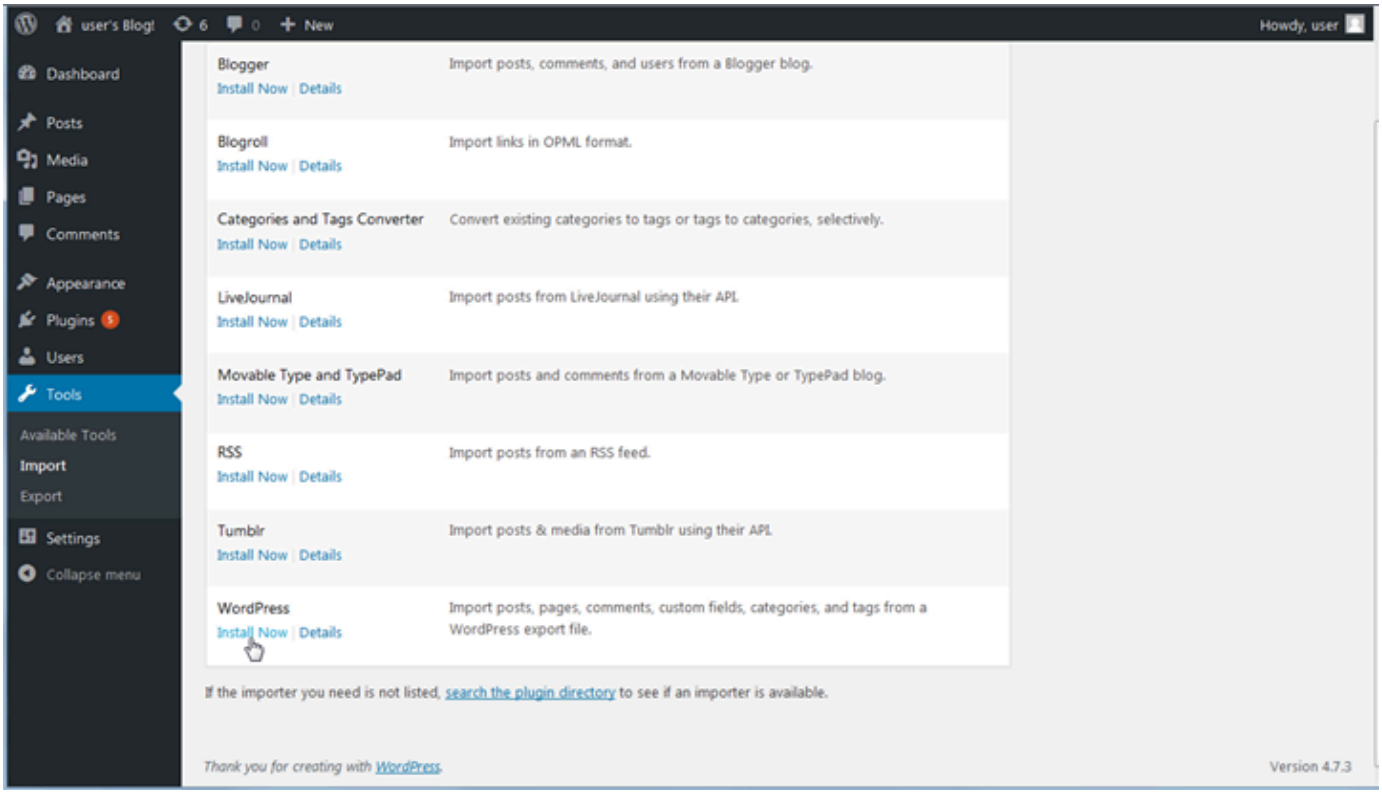
WordPress ウェブサイトの管理ダッシュボードにサインインしたら、管理アクションを実行できます。WordPress ウェブサイトの管理の詳細については、WordPress のドキュメントで「WordPress Codex」を参照してください。



ステップ 4: XML データを新しい Lightsail ブログにインポートする

新しい Lightsail インスタンスの WordPress ダッシュボードに正常にログインしたら、以下のステップに従って、XML ファイルを新しい Lightsail ブログにインポートします。

1. 新しい インスタンスの WordPress ダッシュボードLightsailで [ツール] を選択します。
2. [インポート] を選択し、[今すぐインストール] を選択して、WordPress のインポートツールをインストールします。



3. ツールのインストールが完了したら、[インポーターの実行] を選択してインポートツールを実行します。
4. [WordPress のインポート] ページで [参照] を選択します。
5. 「ステップ 1: 既存の WordPress ブログをバックアップする」で保存した XML ファイルを見つけて [開く] を選択します。
6. [ファイルをアップロードしてインポート] を選択します。

   残りはデフォルトのままにして、[送信] を選択します。

次の手順

ホームアイコンの横にあるブログを選択し、WordPress ダッシュボードで [Visit Site (サイトを訪問)] を選択して、すべてが動作することを確認できます。ブラウザに IP アドレスを入力してブログを表示することもできます。

次のステップを以下に示します。

- ドメインネームサーバーが新しいバージョンのブログを指すように、DNS を移行します。
- 新しいブログの外観をカスタマイズし、いくつかの WordPress プラグインをインストールします。
- で WordPress インスタンスを起動して設定する Lightsail

段階的な手順に従って、WordPress インスタンスを起動して設定し、HTTPS で保護し、外部データベースまたはストレージサービスに接続し、既存のブログを Lightsail に移行します。このチュートリアルでは、WordPress 管理者認証情報の取得、プラグインのインストール、DNS とドメインの設定、Amazon S3、Amazon Aurora、Amazon SES AWS のサービス などの他の との統合などの重要なタスクについて説明します。このガイドに従うことで、Lightsail プラットフォーム上で安全性が高く、スケーラブルで高性能な WordPress ウェブサイトを簡単に設定および管理できます。

Lightsail で マルチサイトを使用して複数の WordPress サイトを管理する

このセクションでは、Amazon Lightsail のWordPress マルチサイトインスタンスでのブログの管理に関連する以下のトピックについて説明します:

トピック

- Lightsail の WordPress マルチサイトにブログをドメインとして追加する
- Lightsail の WordPress マルチサイトにブログをサブドメインとして追加する
- Lightsail で WordPress マルチサイトインスタンスのプライマリドメインを定義する

Lightsail の WordPress マルチサイトにブログをドメインとして追加する

Amazon Lightsail の WordPress マルチサイトインスタンスは、インスタンス内で作成するブログサイトごとに、複数のドメインまたはサブドメインを使用するように設計されています。このガイドでは、WordPress マルチサイトインスタンスでメインブログのプライマリドメインとは異なるドメイ

ンを使用してブログサイトを追加する方法を示します。たとえば、メインブログのプライマリドメインが example.com である場合、同じインスタンスで another-example.com ドメインや third-example.com ドメインを使用する新しいブログサイトを作成できます。

> **Note**
>
> また、サブドメインを使用するサイトを WordPress マルチサイトインスタンスに追加することもできます。詳細については、「WordPress マルチサイトインスタンスにブログをサブドメインとして追加する」を参照してください。

前提条件

次の前提条件を以下に示す順に実行してください。

1. Lightsail で WordPress マルチサイトインスタンスを作成します。詳細については、「インスタンスを作成する」を参照してください。
2. 静的 IP を作成して Lightsail の WordPress マルチサイトインスタンスにアタッチします。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。
3. DNS ゾーンを作成してドメインを Lightsail に追加し、このドメインが WordPress マルチサイトインスタンスにアタッチした静的 IP をポイントするように設定します。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。
4. WordPress マルチサイトインスタンスのプライマリドメインを定義する。詳細については、「WordPress マルチサイトインスタンスのプライマリドメインを定義する」を参照してください。

WordPress マルチサイトインスタンスにブログをドメインとして追加する

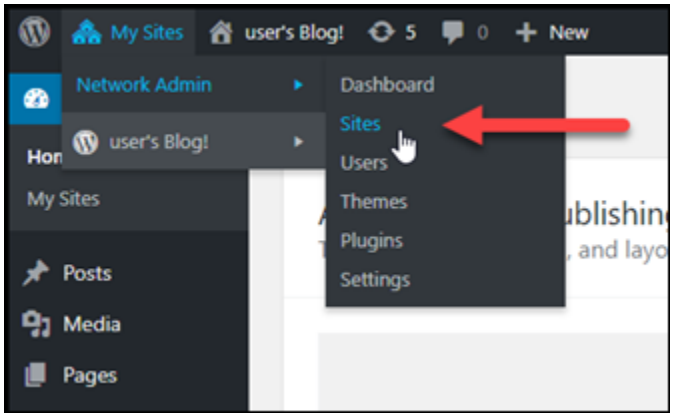
以下の手順を実行し、メインブログのプライマリドメインとは異なるドメインを使用するブログサイトを WordPress マルチサイトインスタンスで作成します。

> **Important**
>
> 次の手順を実行する前に、このガイドの前提条件のセクションに記載されているステップ 4 を完了する必要があります。

1. WordPress マルチサイトインスタンスの管理ダッシュボードにサインインします。

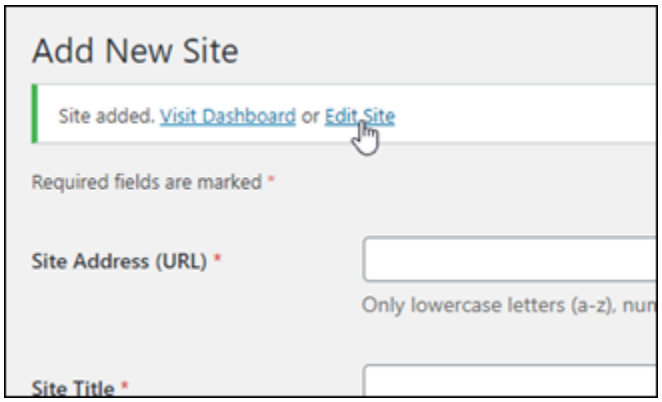
> **Note**
>
> 詳細については、「Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

2. 上部のナビゲーションペインで [My Sites] (自分のサイト)、[Network Admin] (ネットワーク管理者)、[Sites] (サイト) の順に選択します。

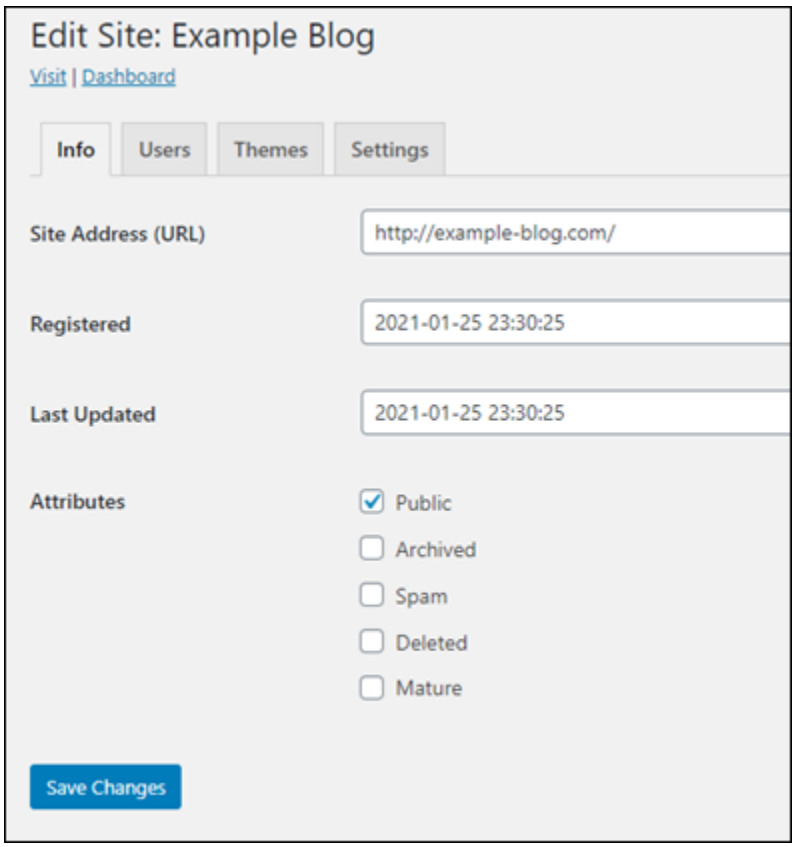


3. [Add New] (新規追加) を選択して新しいブログサイトを追加します。
4. サイトのアドレスをサイトアドレス (URL)テキストボックスに入力します。こちらが新しいブログサイトに使われるドメインになります。例えば、新しいブログサイトで `example-blog.com` をドメインとして使用する場合は、サイトアドレス (URL) テキストボックスに`example-blog`と入力します。ページに表示されるプライマリドメインのサフィックスは無視します。

5. サイトのタイトルを入力し、サイトの言語を選択して、管理者の E メールアドレスを入力します。
6. [Add Site] (サイトの追加) を選択します。
7. ページに表示させる確認バナーでサイトの編集を選択します。最近作成したサイトの詳細編集にリダイレクトされます。

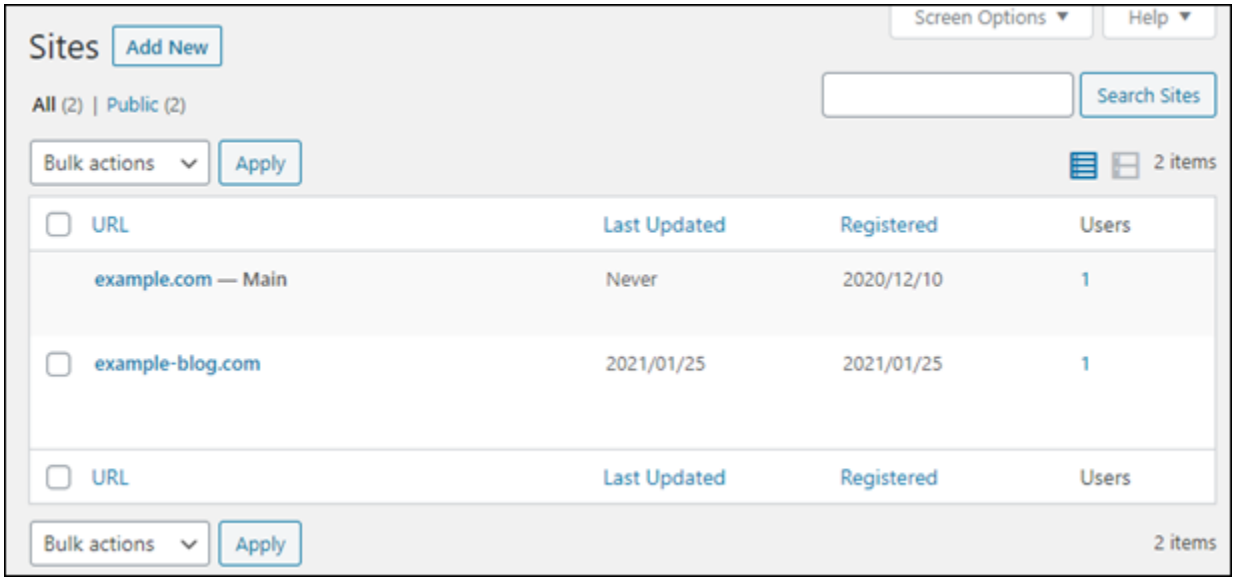


8. サイトの編集ページ上で、サイトアドレス (URL)テキストボックスにリストされているサブドメインを使用したい apex ドメインに変更します。この例では、`http://example-blog.com`を指定しました。



9. [Save Changes] (変更を保存する) を選択します。

この時点で、新しいブログサイトは WordPress マルチサイトインスタンスに作成されましたが、ドメインは新しいブログサイトにルーティングされるようにまだ設定されていません。次のステップに進み、アドレスレコード (A レコード) をドメインの DNS ゾーンに追加します。



アドレスレコード (A レコード) をドメインの DNS ゾーンに追加する

以下の手順を実行し、新しいブログサイトのドメインが WordPress マルチサイトインスタンスをポイントするように設定します。以下の手順は、WordPress Multisite インスタンスで作成するブログサイトごとに実行する必要があります。

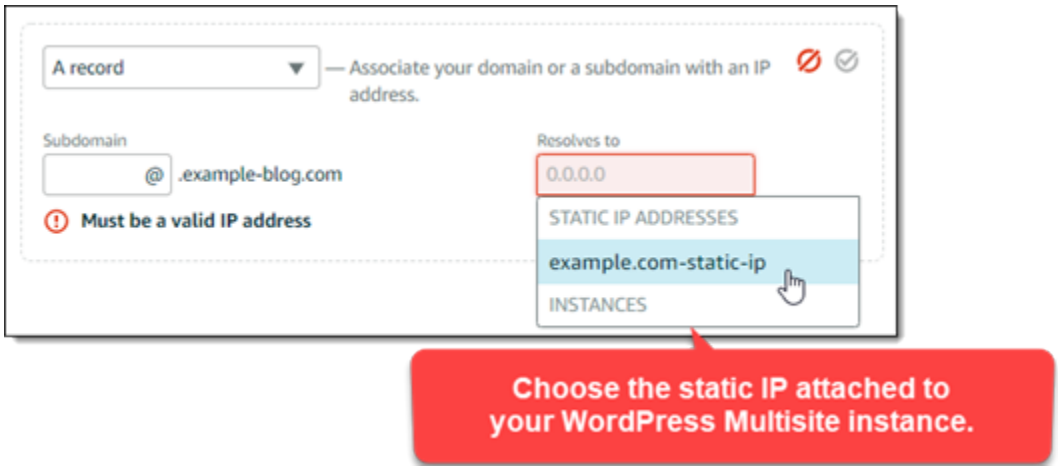
デモの目的で、Lightsail の DNS ゾーンを使用します。ただし、ドメインレジストラがホストする他の一般的な DNS ゾーンでも手順は同様です。

> **Important**
>
> Lightsail コンソールでは、最大 6 つの DNS ゾーンを作成できます。さらに DNS ゾーンを増やす場合は、Amazon Route 53 を使用してドメインの DNS レコードを管理することをお勧めします。詳細については、「Amazon Route 53 を既存ドメインの DNS サービスにする」を参照してください。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ドメインと DNS] を選択します。

3. ページの [DNS ゾーン] セクションで、新しいブログサイトのドメインの DNS ゾーンを選択します。
4. DNS ゾーンエディタで [DNS records] (DNS レコード) タブを選択します。次に、[Add record] (レコードの追加) を選択します。

5. レコードタイプのドロップダウンメニューで [A レコード] を選択します。
6. [Record name] (レコード名) テキストボックスに、「at」記号 (@) を入力し、ドメインのルートのレコードを作成します。
7. [Resolves to] (解決先) テキストボックスで、WordPress Multisite インスタンスにアタッチされている静的 IP アドレスを選択します。



8. 保存アイコンを選択します。

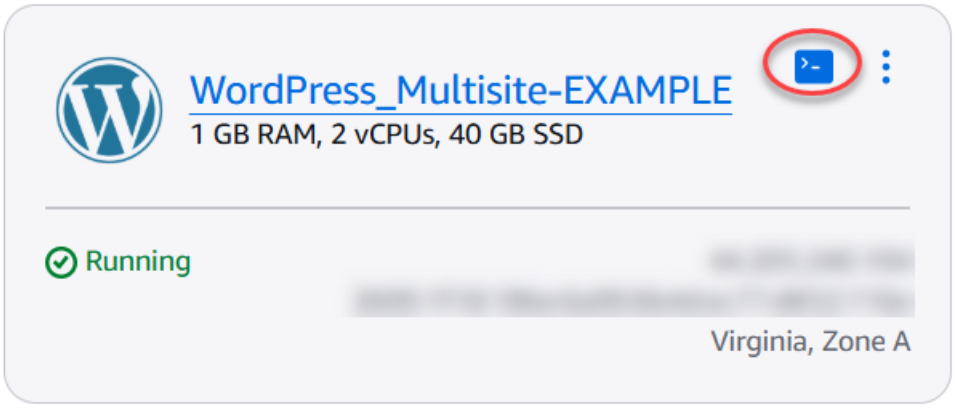
   変更がインターネットの DNS を通じて伝播されると、ドメインは WordPress マルチサイトインスタンスの新しいブログサイトにルートされます。

Cookie Support を有効にして、ブログサイトへのサインインを許可する

ブログサイトをドメインとして WordPress Multisite インスタンスに追加する場合は、cookie support を有効化するために、インスタンスにある WordPress の設定 (`wp-config`) ファイルをアップデートする必要があります。Cookie support を有効にしない場合、ユーザーがブログサイトの

WordPress 管理ダッシュボードにサインインしようとすると、「エラー: Cookie がブロックされているかサポートされていません」とエラーが表示されることがあります。

1. Lightsail コンソール にサインインします。
2. Lightsail のホームページで、WordPress Multisite インスタンスの SSH クイック接続アイコンを選択します。



3. Lightsail ブラウザベースの SSH セッションに接続後、以下のコマンドを入力して Vim を利用し、インスタンスの`wp-config.php`ファイルを開けて編集します。

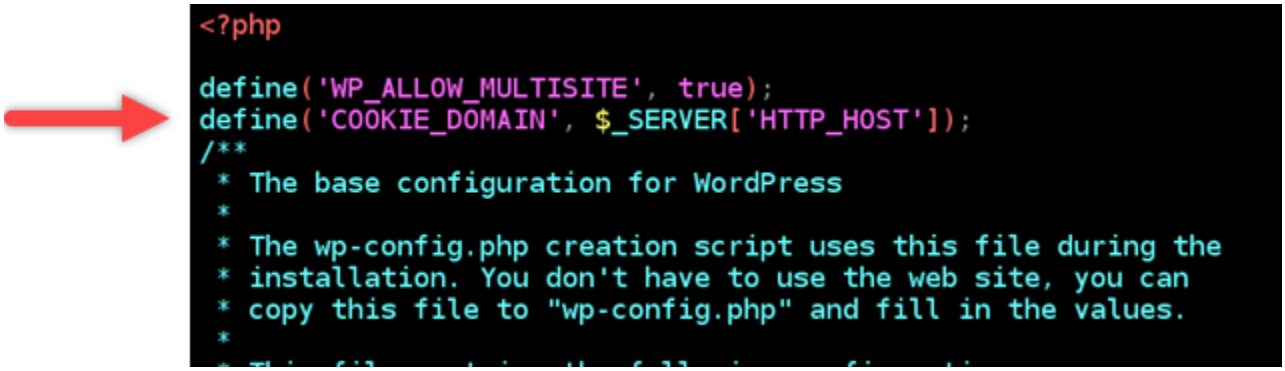
```
sudo vim /opt/bitnami/wordpress/wp-config.php
```

> **Note**
>
> このコマンドが失敗した場合は、古いバージョンの WordPress Multisite インスタンスを使用している可能性があります。代わりに次のコマンドを実行してみてください。
>
> ```
> sudo vim /opt/bitnami/wordpress/wp-config.php
> ```

4. I を押して Vim モード挿入を入力します。
5. 以下のテキストの行を `define('WP_ALLOW_MULTISITE', true);` テキストの行の下に追加します。

```
define('COOKIE_DOMAIN', $_SERVER['HTTP_HOST']);
```

完了すると、ファイルは次のようになります。



6. ESC キーを押して Vim モード挿入を終了後、`:wq!` を入力して Enter を押して編集 (書き込み) を保存して Vim を終了します。
7. 次のコマンドを入力して、WordPress インスタンス上の基本サービスを再起動します。

   ```
   sudo /opt/bitnami/ctlscript.sh restart
   ```

   これで、WordPress マルチサイトインスタンスで cookie が有効化されて、ブログサイトにサインインしようとしているユーザーに「エラー:Cookie がブロックされているかサポートされていません」というエラーが発生しなくなります。

次の手順

WordPress マルチサイトインスタンスにブログをドメインとして追加した後、WordPress マルチサイトの管理に慣れることをお勧めします。詳細については 、WordPress ドキュメントにある Multisite ネットワーク管理 を参照してください。

Lightsail の WordPress マルチサイトにブログをサブドメインとして追加する

Amazon Lightsail の WordPress マルチサイトインスタンスは、インスタンス内で作成するブログサイトごとに、複数のドメインまたはサブドメインを使用するように設計されています。このガイドでは、WordPress マルチサイトインスタンスのサブドメインとしてブログサイトを追加する方法について説明します。たとえば、メインブログのプライマリドメインが `example.com` である場合、同じインスタンスで `earth.example.com` サブドメインや `moon.example.com` サブドメインを使用する新しいブログサイトを作成できます。

> **Note**
>
> また、ドメインを使用するサイトを WordPress マルチサイトインスタンスに追加することもできます。詳細については、「WordPress マルチサイトインスタンスにブログをドメインとして追加する」を参照してください。

前提条件

次の前提条件を以下に示す順に実行してください。

1. WordPress マルチサイトインスタンスを作成する。詳細については、「インスタンスを作成する」を参照してください。
2. 静的 IP を作成して WordPress マルチサイトインスタンスにアタッチします。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。
3. DNS ゾーンを作成してドメインを Lightsail に追加し、このドメインが WordPress マルチサイトインスタンスにアタッチした静的 IP をポイントするように設定します。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。
4. WordPress マルチサイトインスタンスのプライマリドメインを定義する。詳細については、「WordPress マルチサイトインスタンスのプライマリドメインを定義する」を参照してください。

WordPress マルチサイトインスタンスにブログをサブドメインとして追加する

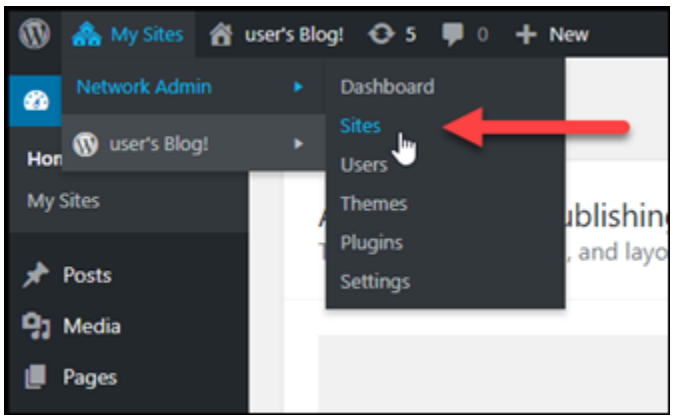
以下の手順を実行し、メインブログのプライマリドメインのサブドメインを使用する新しいブログを WordPress マルチサイトインスタンスで作成します。

> **Important**
>
> 次の手順を実行する前に、このガイドの前提条件のセクションに記載されているステップ 4 を完了する必要があります。

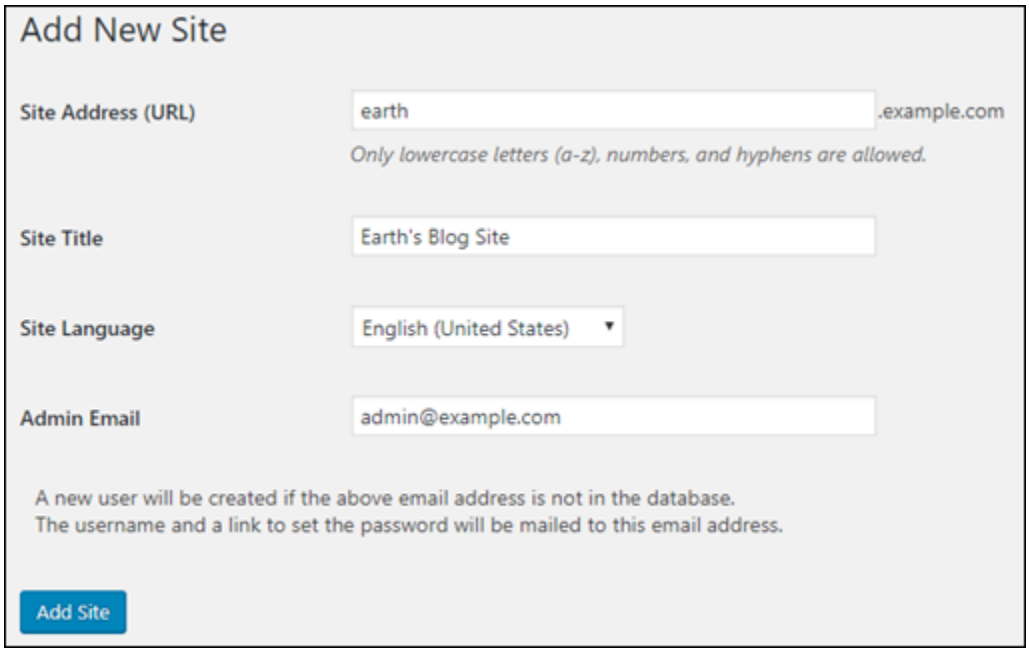
1. WordPress マルチサイトインスタンスの管理ダッシュボードにサインインします。

> **Note**
>
> 詳細については、「Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

2. 上部のナビゲーションペインで [My Sites] (自分のサイト)、[Network Admin] (ネットワーク管理者)、[Sites] (サイト) の順に選択します。

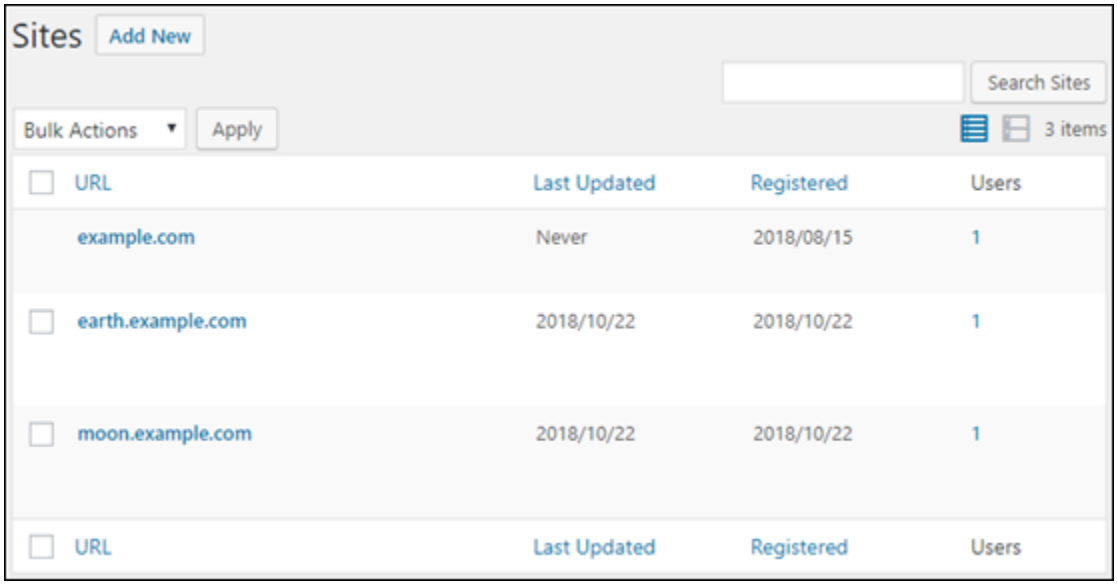


3. [Add New] (新規追加) を選択して新しいブログサイトを追加します。
4. 新しいブログサイトのサブドメインとして使用するサイトアドレスを入力します。



5. サイトのタイトルを入力し、サイトの言語を選択して、管理者の E メールアドレスを入力します。
6. [Add Site] (サイトの追加) を選択します。

この時点で、新しいブログサイトは WordPress マルチサイトインスタンスに作成されましたが、サブドメインは新しいブログサイトにルーティングされるようにまだ設定されていません。次のステップに進み、アドレスレコード (A レコード) をドメインの DNS ゾーンに追加します。

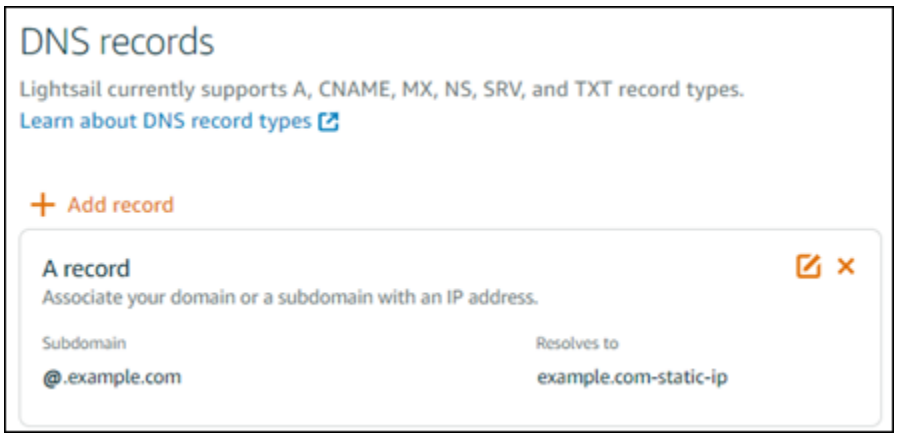


アドレスレコード (A レコード) をドメインの DNS ゾーンに追加する

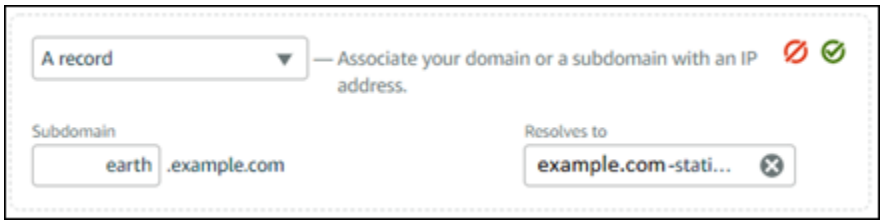
以下の手順を実行し、新しいブログサイトのサブドメインが WordPress マルチサイトインスタンスをポイントするように設定します。以下の手順は、WordPress マルチサイトインスタンスで作成するブログサイトごとに実行する必要があります。

デモの目的で、Lightsail の DNS ゾーンを使用します。ただし、ドメインレジストラがホストする他の一般的な DNS ゾーンでも手順は同様です。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで [ドメインと DNS] を選択します。
3. ページの [DNS ゾーン] セクションで、WordPress マルチサイトインスタンスのプライマリドメインとして定義したドメインの DNS ゾーンを選択します。
4. DNS ゾーンエディタで [DNS records] (DNS レコード) タブを選択します。次に、[Add record] (レコードの追加) を選択します。



5. レコードタイプのドロップダウンメニューで [A レコード] を選択します。
6. [Record name] (レコード名) テキストボックスに、WordPress マルチサイトインスタンスで新しいブログサイトを作成したときにサイトアドレスとして指定したサブドメインを入力します。
7. [Resolves to] (解決先) テキストボックスで、WordPress Multisite インスタンスにアタッチされている静的 IP アドレスを選択します。



8. 保存アイコンを選択します。

   必要な操作は以上のみです。変更がインターネットの DNS を通じて伝播されると、ドメインは WordPress マルチサイトインスタンスの新しいブログサイトにリダイレクトされます。

次の手順

WordPress マルチサイトインスタンスにブログをサブドメインとして追加した後は、WordPress マルチサイトの管理に慣れることをお勧めします。詳細については、WordPress ドキュメントにあるマルチサイトネットワーク管理を参照してください。

Lightsail で WordPress マルチサイトインスタンスのプライマリドメインを定義する

Amazon Lightsail の WordPress Multisite インスタンスは、インスタンス内で作成するブログサイトごとに、複数のドメインまたはサブドメインを使用するように設計されています。このた

め、WordPress マルチサイトインスタンスのメインブログで使用するプライマリドメインを定義する必要があります。

前提条件

次の前提条件を以下に示す順に実行してください。

1. Lightsail で WordPress マルチサイトインスタンスを作成します。詳細については、「インスタンスを作成する」を参照してください。
2. 静的 IP を作成して Lightsail の WordPress マルチサイトインスタンスにアタッチします。詳細については、「静的 IP を作成してインスタンスにアタッチする」を参照してください。

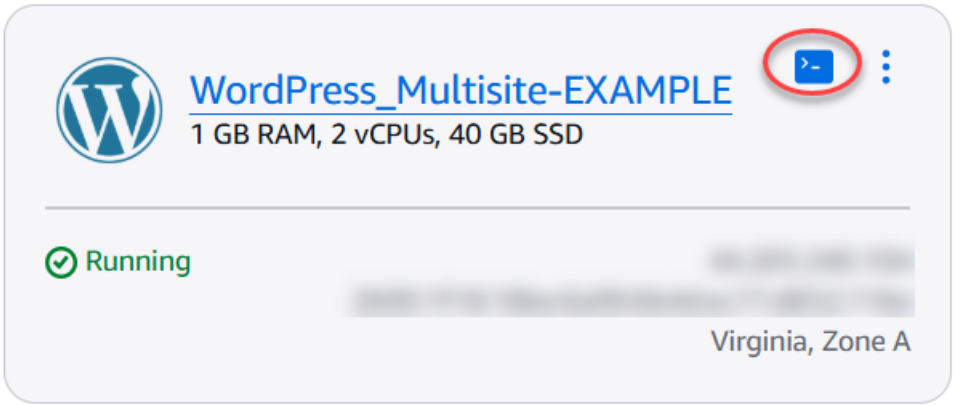
> **Important**
>
> WordPress マルチサイトインスタンスを再起動するには、静的 IP をアタッチした後に再起動する必要があります。これにより、インスタンスは、それに関連付けられた新しい静的 IP を認識できるようになります。

3. DNS ゾーンを作成してドメインを Lightsail に追加し、このドメインが WordPress マルチサイトインスタンスにアタッチした静的 IP をポイントするように設定します。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。
4. DNS の変更がインターネットの DNS を通じて伝播されるまで待ちます。その後、このガイドの「WordPress マルチサイトインスタンスのプライマリドメインを定義する」セクションに進みます。

WordPress マルチサイトインスタンスのプライマリドメインを定義する

以下のステップを実行し、ドメイン (`example.com` など) が WordPress マルチサイトインスタンスのメインブログにリダイレクトされるようにします。

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、WordPress マルチサイトインスタンスの SSH クイック接続アイコンを選択します。



3. 次のコマンドを入力して、WordPress マルチサイトインスタンスのプライマリドメイン名を定義します。*<domain>* は WordPress マルチサイトの正しいドメイン名に置き換えてください。

```
sudo /opt/bitnami/configure_app_domain --domain <domain>
```

例:

```
sudo /opt/bitnami/configure_app_domain --domain example.com
```

> **Note**
>
> このコマンドが失敗した場合は、古いバージョンの WordPress マルチサイトインスタンスを使用している可能性があります。*<domain>* を WordPress マルチサイトの正しいドメイン名で置き換えて以下のコマンドを実行してください。
>
> ```
> cd /opt/bitnami/apps/wordpress
> sudo ./bnconfig --machine_hostname <domain>
> ```
>
> コマンドの実行が終了したら次のコマンドを入力し、サーバーが再起動するたびに bnconfig ツールが自動的に実行されないようにします。
>
> ```
> sudo mv bnconfig bnconfig.disabled
> ```

この時点で、定義したドメインを参照すると、WordPress マルチサイトインスタンスのメインブログにリダイレクトされます。

次の手順

WordPress マルチサイトインスタンスのプライマリドメインを定義したら、以下のステップを実行します。

- WordPress マルチサイトインスタンスにブログをサブドメインとして追加する
- WordPress マルチサイトインスタンスにブログをドメインとして追加する

別のドメインまたはサブドメインを使用して新しいブログサイトを追加する方法、および WordPress マルチサイトインスタンスでメインブログのプライマリドメインを定義する方法については、ステップバイステップの手順に従います。

このガイドでは、WordPress マルチサイトインスタンスの作成、静的 IP のアタッチ、DNS ゾーンの作成、プライマリドメインの設定などの前提条件について説明します。その後、ブログをドメインやサブドメインとして追加する手順、DNS レコードの更新、クッキーサポートの有効化、その他必要な設定を行う具体的な手順を提供します。このガイドに従うことで、WordPress マルチサイトインスタンス内で複数のブログを効果的に管理および整理し、ブログサイトごとに個別のドメインまたはサブドメインを使用する柔軟性を活用できます。

Let's Encrypt で Lightsail リソースの暗号化された通信を有効にする

このガイドでは、Amazon Lightsail の Let's Encrypt に関連する以下のトピックについて説明します。開始するには、以下の前提条件を満たしていることを確認する必要があります:

前提条件

- LAMP、Nginx、または WordPress を実行する Lightsail インスタンスを作成する
- ドメイン名を登録し、その DNS レコードを編集するためのアクセスを取得する
- Lightsail ブラウザベースの SSH ターミナルまたは独自の SSH クライアントを使用する。

トピック

- Let's Encrypt の SSL 証明書で Lightsail LAMP インスタンスを保護する
- Let's Encrypt Lightsail SSL/TLS を使用して NGINX ウェブサイトを保護する
- Let's Encrypt の無料 SSL 証明書で Lightsail の WordPress インスタンスを保護する

Let's Encrypt の SSL 証明書で Lightsail LAMP インスタンスを保護する

Amazon Lightsail では、Lightsail ロードバランサーを使用すると、SSL/TLS でウェブサイトとアプリケーションのセキュリティを簡単に強化できます。ただし、Lightsail ロードバランサーの使用は一般的に最適な選択肢ではない場合があります。ロードバランサーが提供するスケーラビリティや耐障害性がサイトでは必要ない場合や、コストを最適化するためにロードバランサーを使用しない場合があります。

後者の場合は、Let's Encrypt で無料の SSL 証明書を入手できます。無料の証明書を使用することに問題はありません。これらの証明書は Lightsail インスタンスに統合できます。このチュートリアルでは、Certbot を使用して Let's Encrypt ワイルドカード証明書をリクエストし、これを LAMP インスタンスに統合する方法を示します。

Important

- Bitnami インスタンスで使用されている Linux ディストリビューションは、2020 年 7 月に Ubuntu から Debian に変更されました。この変更により、このチュートリアルのいくつかのステップは、インスタンスの Linux ディストリビューションによって異なります。変更後に作成された Bitnami ブループリントインスタンスはすべて Debian Linux ディストリビューションを使用します。変更前に作成されたインスタンスは、Ubuntu Linux ディストリビューションを引き続き使用します。インスタンスのディストリビューションをチェックするには、`uname -a` コマンドを実行します。応答には、インスタンスの Linux ディストリビューションとして Ubuntu または Debian のいずれかが表示されます。
- Bitnami は、多くのスタックのファイル構造を変更するプロセスです。このチュートリアルのファイルパスは、Bitnami スタックがネイティブ Linux システムパッケージを使用しているか (アプローチ A)、または自己完結型インストール (アプローチ B) であるかによって、変更される場合があります。Bitnami のインストールタイプと取るべき方法を特定するには、次のコマンドを実行します。

  ```
  test ! -f "/opt/bitnami/common/bin/openssl" && echo "Approach
  A: Using system packages." || echo "Approach B: Self-contained
  installation."
  ```

目次

- ステップ 1: 前提条件を満たす

- ステップ 2: インスタンスに Certbot をインストールする
- ステップ 3: Let's Encrypt の SSL ワイルドカード証明書をリクエストする
- ステップ 4: ドメインの DNS ゾーンに TXT レコードを追加する
- ステップ 5: TXT レコードが反映されたことを確認する
- ステップ 6: Let's Encrypt の SSL 証明書リクエストを完了する
- ステップ 7: Apache サーバーディレクトリで Let's Encrypt の証明書ファイルへのリンクを作成する
- ステップ 8: ウェブアプリケーションの HTTP から HTTPS へのリダイレクトを設定する
- ステップ 9: Let's Encrypt 証明書を 90 日ごとに更新する

ステップ 1: 前提条件を満たす

以下の前提条件を満たします (まだ満たしていない場合)。

- Lightsail で LAMP インスタンスを作成します。詳細については、「インスタンスを作成する」を参照してください。
- ドメイン名を登録し、その DNS レコードを編集するための管理アクセスを取得します。詳細については、「Amazon Lightsail の DNS」を参照してください。

> **Note**
>
> ドメインの DNS レコードは、Lightsail の DNS ゾーンを使用して管理することをお勧めします。詳細については、「 DNS ゾーンを作成し、ドメインの DNS レコードを管理する」を参照してください。

- Lightsail コンソールでブラウザベースの SSH ターミナルを使用して、このチュートリアルのステップを実行します。ただし、独自の SSH クライアント (PuTTY など) を使用することもできます。PuTTY の設定の詳細については、「PuTTY をダウンロード、SSH を使用して接続するようにセットアップする」を参照してください。

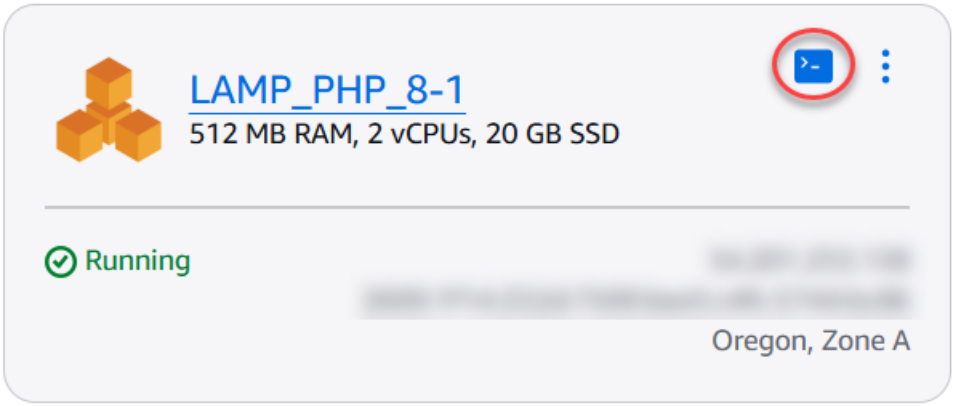
前提条件が完了したら、このチュートリアルの「次のセクション」に進みます。

ステップ 2: インスタンスに Certbot をインストールする

Certbot は、Let's Encrypt の証明書をリクエストしてウェブサーバーにデプロイするために使用するクライアントです。Let's Encrypt は ACME プロトコルを使用して証明書を発行します。Certbot は、Let's Encrypt とやり取りする ACME 対応のクライアントです。

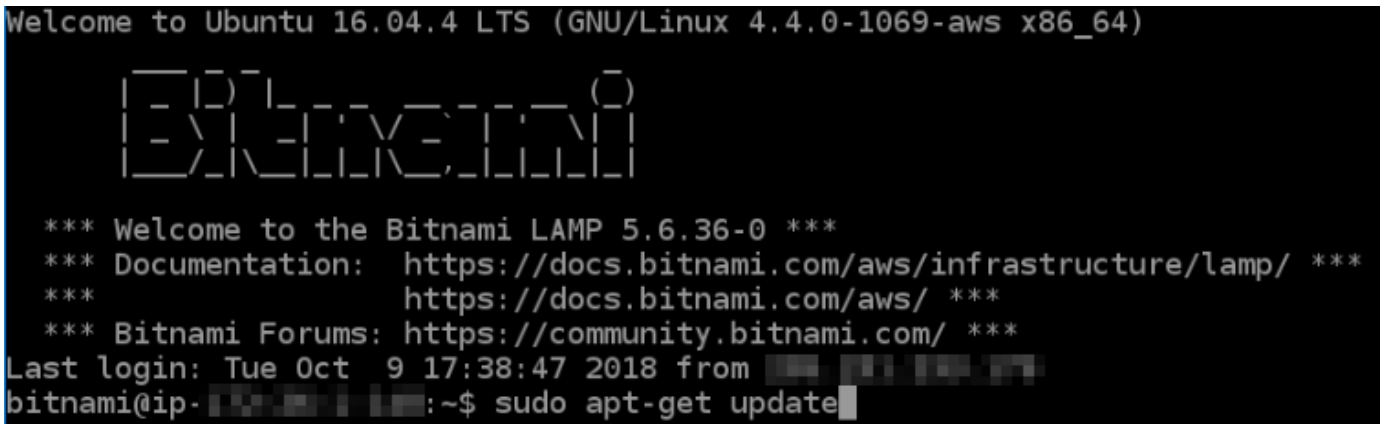
Lightsail インスタンスに Certbot をインストールするには

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、接続先のインスタンスの SSH クイック接続アイコンを選択します。



3. Lightsail のブラウザベースの SSH セッションに接続したら、次のコマンドを入力してインスタンスのパッケージを更新します。

```
sudo apt-get update
```



4. 次のコマンドを入力してソフトウェアプロパティパッケージをインストールします。Certbot の開発者は、Personal Package Archive (PPA) を使用して Cerbot を配信します。ソフトウェアプロパティパッケージを使用すると、PPA をより効率的に操作できます。

```
sudo apt-get install software-properties-common
```

> **Note**
>
> `sudo apt-get install` コマンドを実行したときに `Could not get lock` エラーが発生した場合は、約 15 分待ってから再試行してください。このエラーは、自動アップグレードをインストールするために Apt パッケージ管理ツールを使用している cron ジョブが原因で発生している可能性があります。

5. 次のコマンドを入力して Certbot をローカル apt リポジトリに追加します。

> **Note**
>
> ステップ 5 は、Ubuntu Linux ディストリビューションを使用するインスタンスにのみ適用されます。インスタンスが Debian Linux ディストリビューションを使用している場合は、このステップをスキップしてください。

```
sudo apt-add-repository ppa:certbot/certbot -y
```

6. 次のコマンドを入力して apt を更新し、新しいリポジトリを含めます。

```
sudo apt-get update -y
```

7. 次のコマンドを入力して Cerbot をインストールします。

```
sudo apt-get install certbot -y
```

これで Lightsail インスタンスに Cerbot がインストールされました。

8. ブラウザベースの SSH ターミナルウィンドウは開いたままにします。このチュートリアルで後ほど戻ります。このチュートリアルの「次のセクション」に進みます。

ステップ 3: Let's Encrypt の SSL ワイルドカード証明書をリクエストする

Let's Encrypt の証明書をリクエストするプロセスを開始します。Certbot を使用してワイルドカード証明書をリクエストします。この 1 つの証明書をドメインとそのサブドメインの両

方に使用できます。たとえば、1 つのワイルドカード証明書を example.com 最上位ドメイン、blog.example.com サブドメイン、および stuff.example.com サブドメインに使用できます。

Let's Encrypt の SSL ワイルドカード証明書をリクエストするには

1. このチュートリアルのステップ 2 で使用した同じブラウザベースの SSH ターミナルウィンドウで、以下のコマンドを入力してドメインの環境変数を設定します。より効率的にコマンドをコピーして貼り付け、証明書を取得できます。

   ```
   DOMAIN=Domain
   ```

   ```
   WILDCARD=*.$DOMAIN
   ```

   コマンドで、*Domain* を登録済みのドメイン名に置き換えます。

   例:

   ```
   DOMAIN=example.com
   ```

   ```
   WILDCARD=*.$DOMAIN
   ```

2. 次のコマンドを入力し、変数が正しい値を返すことを確認します。

   ```
   echo $DOMAIN && echo $WILDCARD
   ```

   次のような結果が表示されます。

3. 次のコマンドを入力して Certbot をインタラクティブモードで起動します。このコマンドでは、DNS チャレンジで手動認可を使用してドメインの所有権を検証することを Certbot に指示します。また、最上位ドメインとそのサブドメイン用にワイルドカード証明書をリクエストします。

```
sudo certbot -d $DOMAIN -d $WILDCARD --manual --preferred-challenges dns certonly
```

4. プロンプトに応じて E メールアドレスを入力します。これで更新とセキュリティに関する通知を受信します。
5. Let's Encrypt のサービス利用規約を読みます。読み終わり、同意する場合は A キーを押します。同意しない場合は、Let's Encrypt の証明書を取得できません。
6. E メールアドレスの共有と IP アドレスのログ記録に関するプロンプトに適宜応答します。
7. Let's Encrypt から、指定されたドメインの所有者であることの検証を求められます。これを行うには、ドメインの DNS レコードに TXT レコードを追加します。以下の例に示すように 2 組の TXT レコード値が提供されます。

> **Note**
>
> Let's Encrypt では検証に必要な TXT レコードを 1 つまたは複数提供する場合があります。この例では、検証に使用する 2 つの TXT レコードが提供されました。

8. Lightsail のブラウザベースの SSH セッションは開いたままにします。このチュートリアルで後ほど戻ります。このチュートリアルの「次のセクション」に進みます。

ステップ 4: ドメインの DNS ゾーンに TXT レコードを追加する

ドメインの DNS ゾーンに TXT レコードを追加すると、自分がドメインを所有していることが検証されます。ここでは、デモの目的で Lightsail の DNS ゾーンを使用します。ただし、ドメインレジストラがホストする他の一般的な DNS ゾーンでも手順はほぼ同じです。

> **Note**
>
> ドメインの Lightsail DNS ゾーンの作成方法の詳細については、「Lightsail で DNS ゾーンを作成し、ドメインの DNS レコードを管理する」を参照してください。

Lightsail でドメインの DNS ゾーンに TXT レコードを追加するには

1. 左側のナビゲーションペインで [ドメインと DNS] を選択します。

2. ページの [DNS ゾーン] セクションで、Certbot 証明書リクエストで指定したドメインの DNS ゾーンを選択します。
3. DNS ゾーンエディタで [DNS records] (DNS レコード) を選択します。
4. [レコードの追加] を選択します。
5. [Record type] (レコードタイプ) のドロップダウンメニューで [TXT record] (TXT レコード) を選択します。
6. Let's Encrypt 証明書のリクエストで指定された値を [Record name] (レコード名) と [Responds with] (応答) フィールドに入力します。

> **Note**
>
> Lightsail コンソールには、ドメインの頂点部分があらかじめ入力されています。たとえば、*_acme-challenge.example.com* サブドメインを追加する場合は、*_acme-challenge* をテキストボックスに入力するだけで、レコードを保存するときに Lightsail が `.example.com` の部分を追加します。

7. [保存] を選択します。
8. ステップ 4 ~ 7 を繰り返して、Let's Encrypt の証明書リクエストで指定された 2 番目の TXT レコードのセットを追加します。
9. Lightsail コンソールのブラウザウィンドウは、このチュートリアルで後ほど戻るので開いたままにします。このチュートリアルの「次のセクション」に進みます。

ステップ 5: TXT レコードが反映されたことを確認する

MxToolbox ユーティリティを使用して TXT レコードがインターネットの DNS に反映されたことを確認します。DNS レコードの反映には、DNS ホスティングプロバイダーと DNS レコードの有効期限 (TTL) の設定によって時間がかかる場合があります。このステップを完了し、TXT レコードが反映されたことを確認した上で、Certbot 証明書のリクエストに進むことが重要です。そうしないと、証明書のリクエストは失敗します。

TXT レコードがインターネットの DNS に反映されたことを確認するには

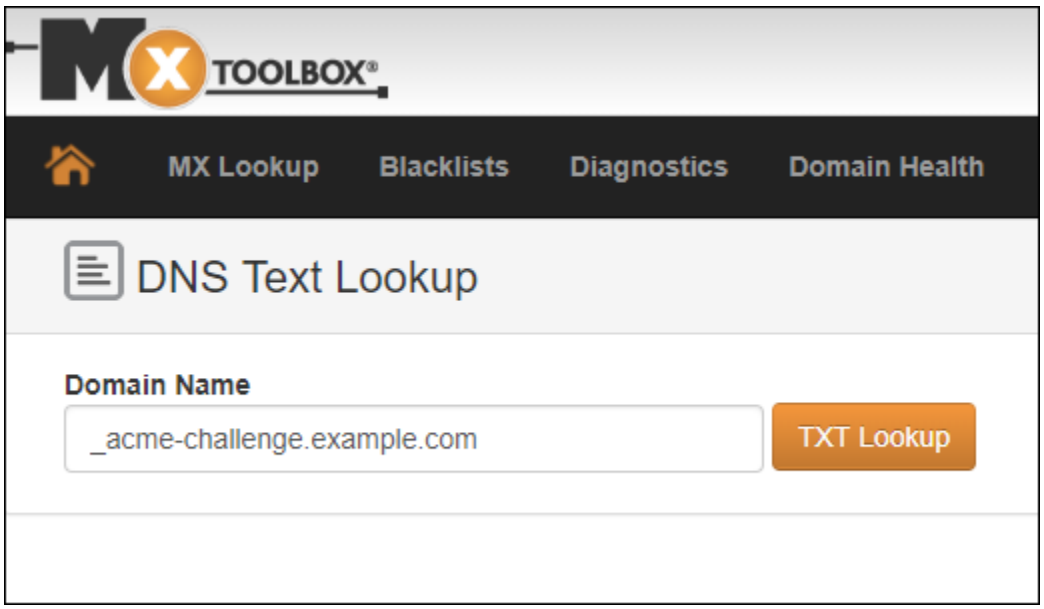
1. 新しいブラウザウィンドウを開き、https://mxtoolbox.com/TXTLookup.aspx に移動します。
2. 次の内容をテキストボックスに入力します。

```
_acme-challenge.Domain
```

Domain は、登録済みのドメイン名に置き換えます。

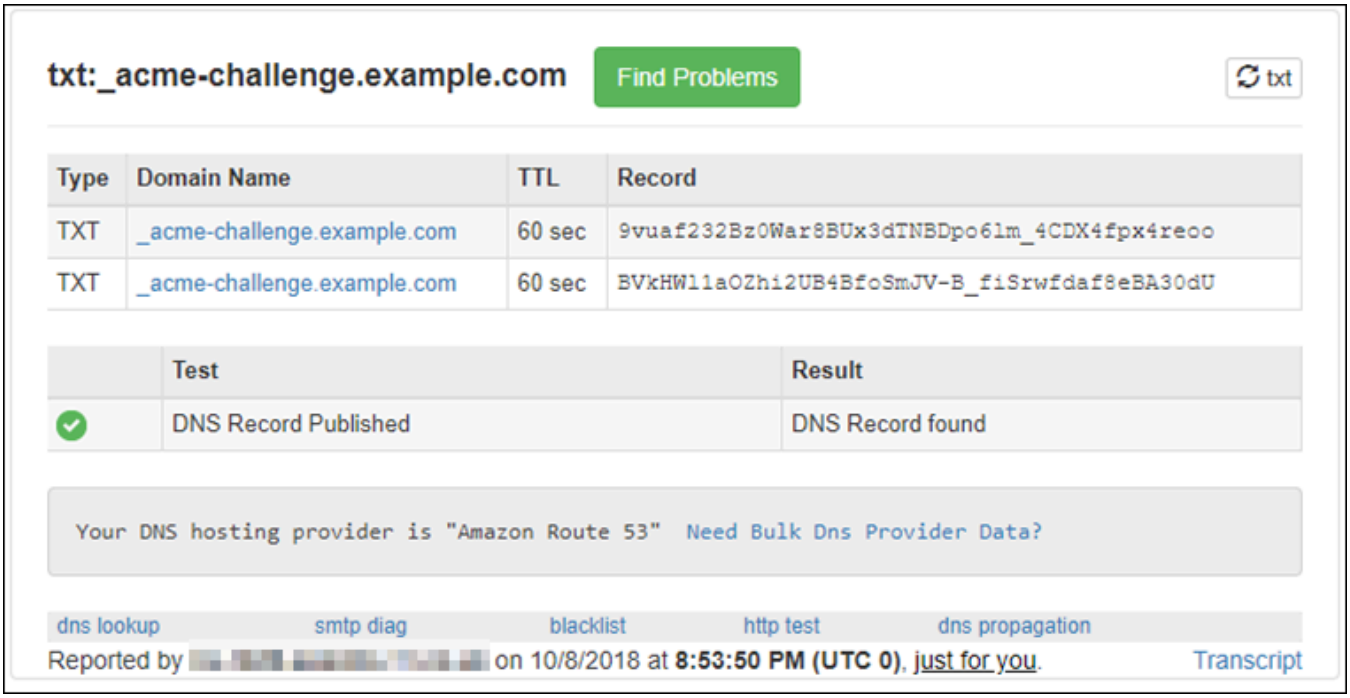
例:

```
_acme-challenge.example.com
```



3. [TXT Lookup (TXT ルックアップ)] を選択して確認を行います。
4. 以下のいずれかのレスポンスが返されます。

   - TXT レコードがインターネットの DNS に反映された場合は、次のスクリーンショットに示すようなレスポンスが表示されます。ブラウザウィンドウを閉じて、このチュートリアルの「次のセクション」に進みます。



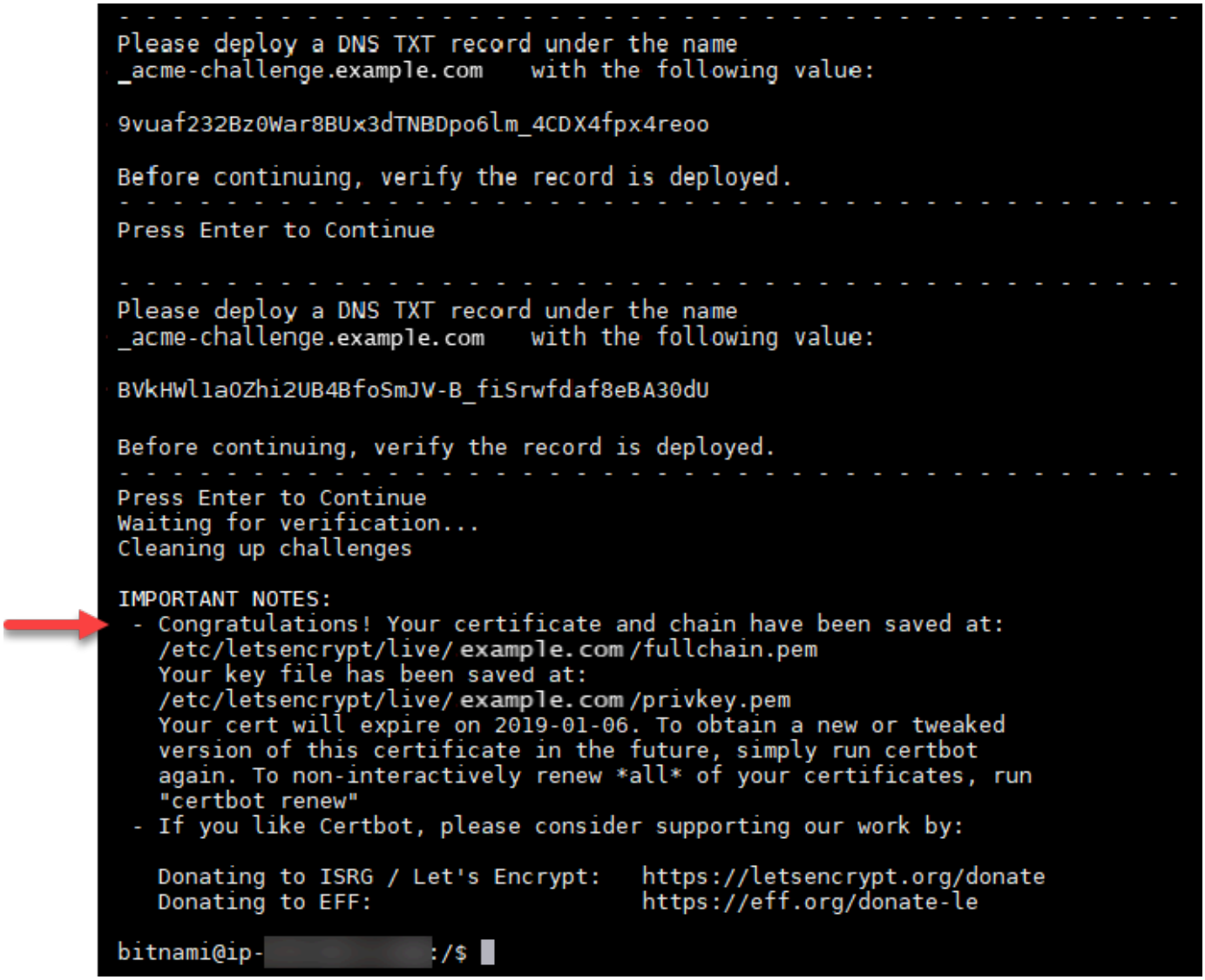
- TXT レコードがインターネットの DNS に反映されていない場合は、[DNS Record not found (DNS レコードが見つかりません)] というレスポンスが返されます。適切な DNS レコードをドメインの DNS ゾーンに追加したことを確認してください。適切なレコードを追加した場合は、ドメインの DNS レコードが反映されるまでしばらく待ってから、TXT のルックアップを再実行します。

ステップ 6: Let's Encrypt の SSL 証明書リクエストを完了する

LAMP インスタンスの Lightsail ブラウザベースの SSH セッションに戻り、Let's Encrypt 証明書のリクエストを完了します。Certbot は、SSL 証明書、チェーン、およびキーファイルを LAMP インスタンスの特定のディレクトリに保存します。

Let's Encrypt の SSL 証明書リクエストを完了するには

1. LAMP インスタンスの Lightsail ブラウザベースの SSH セッションで、Enter キーを押し、Let's Encrypt SSL 証明書のリクエストを続行します。成功すると、次のスクリーンショットに示すようなレスポンスが表示されます。



証明書、チェーン、およびキーファイルが /etc/letsencrypt/live/*Domain*/ ディレクトリに保存されたことを確認するメッセージが表示されます。*[Domain]* (ドメイン) は、登録済みのドメイン名 (/etc/letsencrypt/live/*example.com*/ など) になります。

2. メッセージに記載されている有効期限を書き留めておきます。この期限日までに証明書を更新する必要があります。

```
IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/example.com/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/example.com/privkey.pem
   Your cert will expire on 2019-01-06. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot
   again. To non-interactively renew *all* of your certificates, run
   "certbot renew"
 - If you like Certbot, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
   Donating to EFF:                    https://eff.org/donate-le
```

3. これで Let's Encrypt SSL 証明書が手に入ったので、このチュートリアルの「次のセクション」に進みます。

ステップ 7: Apache サーバーディレクトリで Let's Encrypt の証明書ファイルへのリンクを作成する

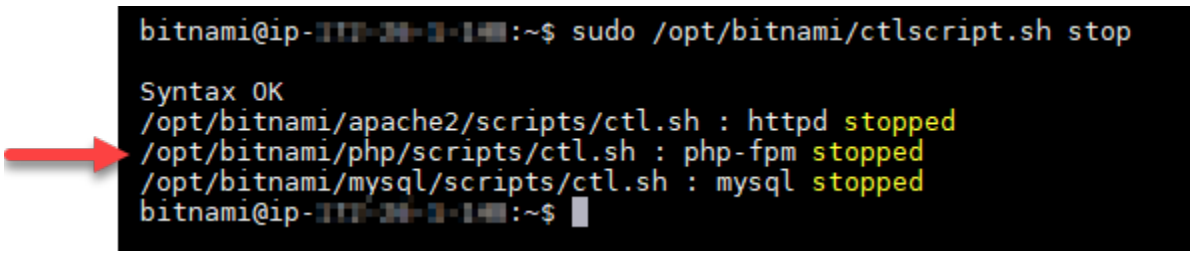
LAMP インスタンスの Apache サーバーディレクトリにある Let's Encrypt の SSL 証明書ファイルへのリンクを作成します。また、必要になる場合に備えて既存の証明書をバックアップします。

Apache サーバーディレクトリで Let's Encrypt の証明書ファイルへのリンクを作成するには

1. LAMP インスタンスの Lightsail ブラウザベースの SSH セッションで、次のコマンドを入力して基盤となる LAMP スタックサービスを停止します。

   ```
   sudo /opt/bitnami/ctlscript.sh stop
   ```

   次のようなレスポンスが表示されます。

2. 次のコマンドを入力してドメインの環境変数を設定します。

   ```
   DOMAIN=Domain
   ```

   コマンドで、*Domain* を登録済みのドメイン名に置き換えます。

例:

```
DOMAIN=example.com
```

3. 次のコマンドを入力し、変数が正しい値を返すことを確認します。

```
echo $DOMAIN
```

次のような結果が表示されます。

4. バックアップとして既存の証明書ファイルがある場合、以下のコマンドを個別に入力して名前を書き換えます。さまざまなディストリビューションとファイル構造の詳細については、このチュートリアルの冒頭の重要ブロックを参照してください。

   - Debian Linux ディストリビューションの場合

     アプローチ A (システムパッケージを使用した Bitnami インストール):

     ```
     sudo mv /opt/bitnami/apache2/conf/bitnami/certs/server.crt /opt/bitnami/apache2/
     conf/bitnami/certs/server.crt.old
     ```

     ```
     sudo mv /opt/bitnami/apache2/conf/bitnami/certs/server.key /opt/bitnami/apache2/
     conf/bitnami/certs/server.key.old
     ```

     アプローチ B (自己完結型 Bitnami インストール):

     ```
     sudo mv /opt/bitnami/apache2/conf/server.crt /opt/bitnami/apache2/conf/
     server.crt.old
     ```

     ```
     sudo mv /opt/bitnami/apache2/conf/server.key /opt/bitnami/apache2/conf/
     server.key.old
     ```

   - Ubuntu Linux ディストリビューションを使用する古いインスタンスの場合：

```
sudo mv /opt/bitnami/apache/conf/bitnami/certs/server.crt /opt/bitnami/apache/
conf/bitnami/certs/server.crt.old
```

```
sudo mv /opt/bitnami/apache/conf/bitnami/certs/server.key /opt/bitnami/apache/
conf/bitnami/certs/server.key.old
```

5. 以下のコマンドを個別に入力し、Apache2 ディレクトリで Let's Encrypt の証明書ファイルへのリンクを作成します。さまざまなディストリビューションとファイル構造の詳細については、このチュートリアルの冒頭の重要ブロックを参照してください。

   - Debian Linux ディストリビューションの場合

     アプローチ A (システムパッケージを使用した Bitnami インストール):

     ```
     sudo ln -sf /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/apache2/conf/
     bitnami/certs/server.key
     ```

     ```
     sudo ln -sf /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/apache2/
     conf/bitnami/certs/server.crt
     ```

     アプローチ B (自己完結型 Bitnami インストール):

     ```
     sudo ln -sf /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/apache2/conf/
     server.key
     ```

     ```
     sudo ln -sf /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/apache2/
     conf/server.crt
     ```

   - Ubuntu Linux ディストリビューションを使用する古いインスタンスの場合：

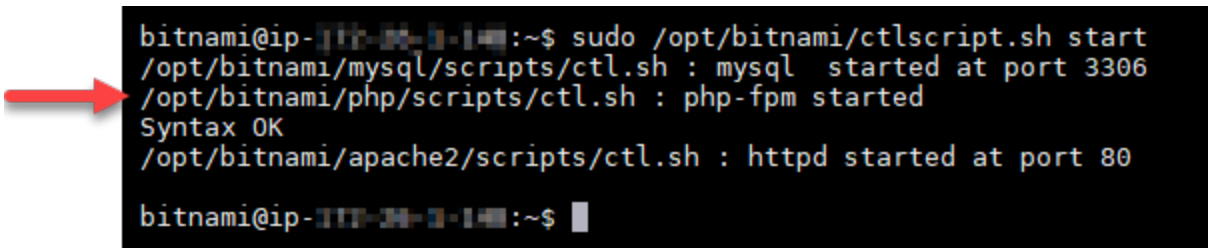
     ```
     sudo ln -s /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/apache/conf/
     bitnami/certs/server.key
     ```

     ```
     sudo ln -s /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/apache/conf/
     bitnami/certs/server.crt
     ```

6. 次のコマンドを入力して、以前に停止した基盤となる LAMP スタックサービスを開始します。

```
sudo /opt/bitnami/ctlscript.sh start
```

次のような結果が表示されます。



これで SSL 暗号化を使用するように LAMP インスタンスが設定されました。ただし、トラフィックは HTTP から HTTPS に自動的にはリダイレクトされません。

7. このチュートリアルの「次のセクション」に進みます。

ステップ 8: ウェブアプリケーションの HTTP から HTTPS へのリダイレクトを設定する

LAMP インスタンスの HTTP から HTTPS へのリダイレクトを設定できます。HTTP から HTTPS へのリダイレクトを自動的に行うことで、SSL を使用するユーザーにのみ (HTTP を使用して接続した場合でも) サイトへのアクセスを許可できます。

ウェブアプリケーションの HTTP から HTTPS へのリダイレクトを設定するには

1. LAMP インスタンスの Lightsail ブラウザベースの SSH セッションで、次のコマンドを入力し、Vim テキストエディタを使用して Apache ウェブサーバー設定ファイルを編集します。

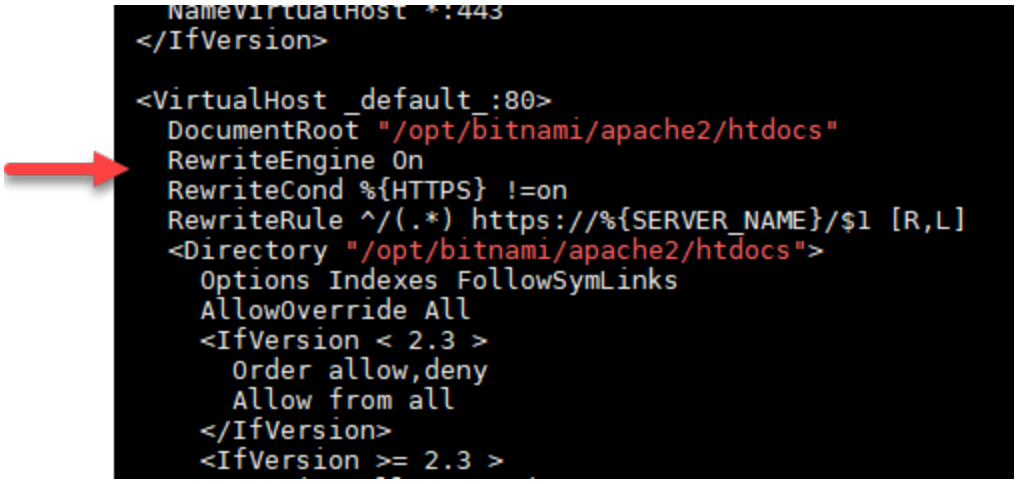
```
sudo vim /opt/bitnami/apache2/conf/bitnami/bitnami.conf
```

> **Note**
>
> このチュートリアルではデモの目的で Vim を使用していますが、任意のテキストエディタを使用できます。

2. `i` キーを押して Vim エディタを挿入モードにします。
3. このファイルで、`DocumentRoot "/opt/bitnami/apache2/htdocs"` と `<Directory "/opt/bitnami/apache2/htdocs">` の間に次のテキストを入力します。

```
RewriteEngine On
RewriteCond %{HTTPS} !=on
RewriteRule ^/(.*) https://%{SERVER_NAME}/$1 [R,L]
```

結果は次のようになります。



4. ESC キーを押して「:wq」と入力し、編集内容を書き込んで (保存して) Vim を終了します。
5. 次のコマンドを入力して基盤となる LAMP スタックサービスを再開し、編集内容を反映します。

```
sudo /opt/bitnami/ctlscript.sh restart
```

これで、HTTP から HTTPS へ自動的に接続をリダイレクトするように LAMP インスタンスが設定されました。訪問者が `http://www.example.com` にアクセスすると、暗号化された `https://www.example.com` アドレスに自動的にリダイレクトされます。

ステップ 9: Let's Encrypt 証明書を 90 日ごとに更新する

Let's Encrypt 証明書の有効期間は 90 日間です。証明書は有効期限が切れる 30 日前から更新できます。Let's Encrypt 証明書を更新するには、取得するために使用した元のコマンドを実行します。このチュートリアルの「Let's Encrypt の SSL ワイルドカード証明書をリクエストする」セクションのステップを繰り返します。

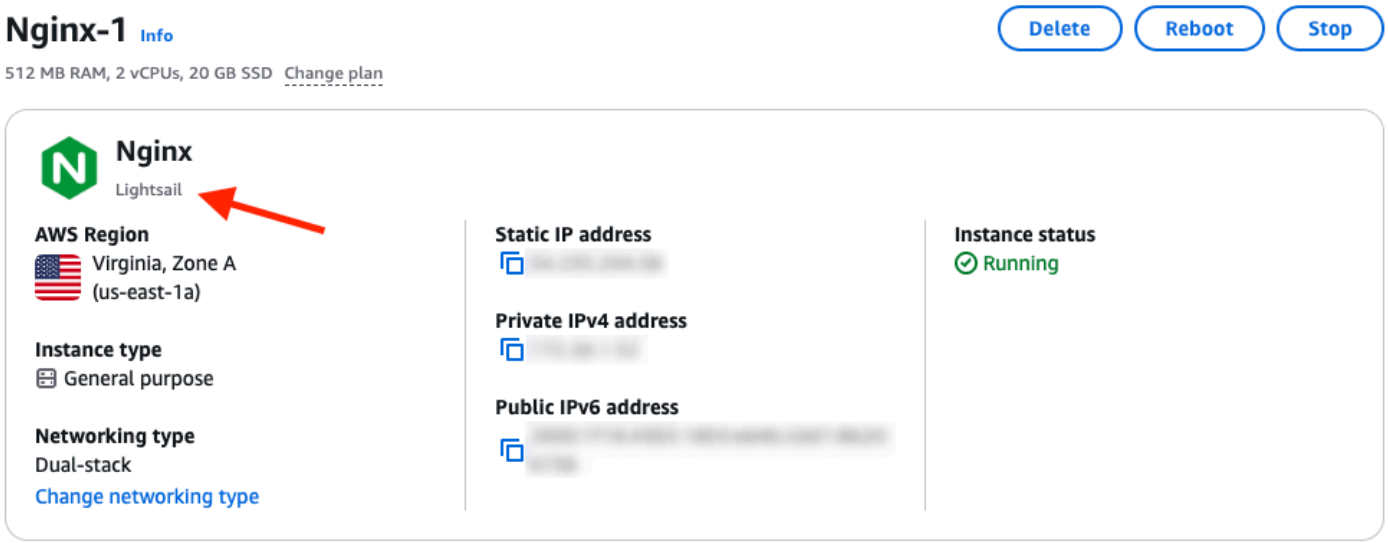
Let's Encrypt Lightsail SSL/TLS を使用して NGINX ウェブサイトを保護する

Amazon Lightsail では、Lightsail ロードバランサーを使用すると、SSL/TLS でウェブサイトとアプリケーションのセキュリティを簡単に強化できます。ただし、Lightsail ロードバランサーの使用は一般的に最適な選択肢ではない場合があります。ロードバランサーが提供するスケーラビリティや耐障害性がサイトでは必要ない場合や、コストを最適化するためにロードバランサーを使用しない場合があります。

後者の場合は、Let's Encrypt で無料の SSL 証明書を入手できます。無料の証明書を使用することに問題はありません。これらの証明書は Lightsail インスタンスに統合できます。このチュートリアルでは、Certbot を使用して Let's Encrypt ワイルドカード証明書をリクエストし、これを Nginx インスタンスに統合する方法を示します。

Nginx ブループリントベンダーを特定する

Nginx インスタンスが で起動および実行された後に開始するために、いくつかのステップを実行する必要がありますAmazon Lightsail。開始する前に、インスタンス管理ページでブループリントベンダーを特定します。



Nginx インスタンスに適したガイドを選択します。

Bitnami

> **Important**
>
> - Bitnami インスタンスで使用されている Linux ディストリビューションは、2020 年 7 月に Ubuntu から Debian に変更されました。この変更により、このチュートリアルのいくつかのステップは、インスタンスの Linux ディストリビューションによって異なります。変更後に作成された Bitnami ブループリントインスタンスはすべて Debian Linux ディストリビューションを使用します。変更前に作成されたインスタンスは、Ubuntu Linux ディストリビューションを引き続き使用します。インスタンスのディストリビューションをチェックするには、`uname -a` コマンドを実行します。応答には、インスタンスの Linux ディストリビューションとして Ubuntu または Debian のいずれかが表示されます。
> - Bitnami は、多くのスタックのファイル構造を変更するプロセスです。このチュートリアルのファイルパスは、Bitnami スタックがネイティブ Linux システムパッケージを使用しているか (アプローチ A)、または自己完結型インストール (アプローチ B) であるかによって、変更される場合があります。Bitnami のインストールタイプと取るべき方法を特定するには、次のコマンドを実行します。
>
>   ```
>   test ! -f "/opt/bitnami/common/bin/openssl" && echo "Approach A: Using system packages." || echo "Approach B: Self-contained installation."
>   ```

目次

- ステップ 1: 前提条件を満たす
- ステップ 2: Lightsail インスタンスに Certbot をインストールする
- ステップ 3: Let's Encrypt の SSL ワイルドカード証明書をリクエストする
- ステップ 4: ドメインの DNS ゾーンに TXT レコードを追加する
- ステップ 5: TXT レコードが反映されたことを確認する
- ステップ 6: Let's Encrypt の SSL 証明書リクエストを完了する
- ステップ 7: NGINX サーバーディレクトリに Let's Encrypt 証明書ファイルへのリンクを作成する
- ステップ 8: ウェブアプリケーションの HTTP から HTTPS へのリダイレクトを設定する

- ステップ 9: Let's Encrypt 証明書を 90 日ごとに更新する

ステップ 1: 前提条件を満たす

以下の前提条件を満たします (まだ満たしていない場合)。

- Lightsail に Nginx インスタンスを作成します。詳細については、「インスタンスを作成する」を参照してください。
- ドメイン名を登録し、その DNS レコードを編集するための管理アクセスを取得します。詳細については、「DNS」を参照してください。

> **Note**
>
> ドメインの DNS レコードは、Lightsail の DNS ゾーンを使用して管理することをお勧めします。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。

- Lightsail コンソールでブラウザベースの SSH ターミナルを使用して、このチュートリアルのステップを実行します。ただし、独自の SSH クライアント (PuTTY など) を使用することもできます。PuTTY の設定の詳細については、「Amazon Lightsail で PuTTY をダウンロードし、SSH を使用して接続するようにセットアップする」を参照してください。

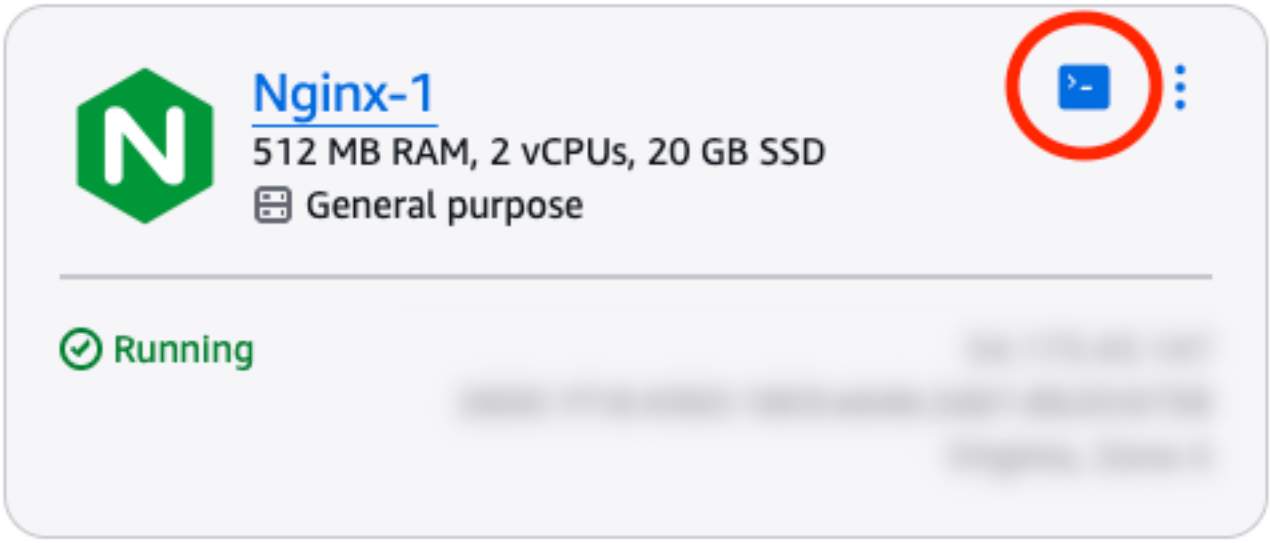
前提条件が完了したら、このチュートリアルの「次のセクション」に進みます。

ステップ 2: Lightsail インスタンスに Certbot をインストールする

Certbot は、Let's Encrypt の証明書をリクエストしてウェブサーバーにデプロイするために使用するクライアントです。Let's Encrypt は ACME プロトコルを使用して証明書を発行します。Certbot は、Let's Encrypt とやり取りする ACME 対応のクライアントです。

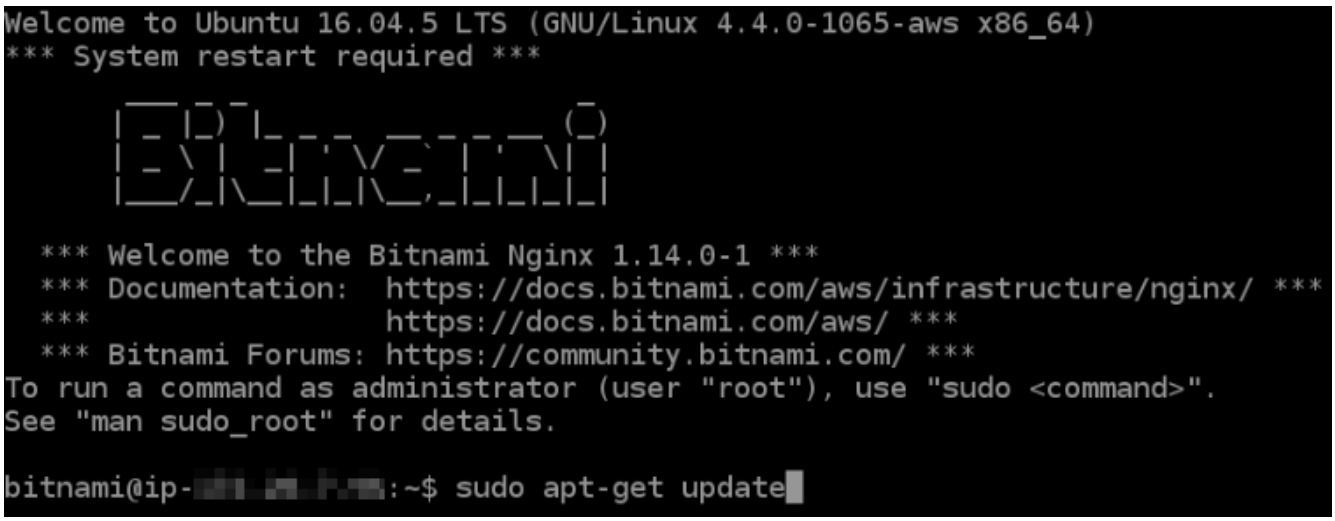
Lightsail インスタンスに Certbot をインストールするには

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、接続先のインスタンスの SSH クイック接続アイコンを選択します。



3. Lightsail のブラウザベースの SSH セッションに接続したら、次のコマンドを入力してインスタンスのパッケージを更新します。

```
sudo apt-get update
```



4. 次のコマンドを入力してソフトウェアプロパティパッケージをインストールします。Certbot の開発者は、Personal Package Archive (PPA) を使用して Cerbot を配信します。ソフトウェアプロパティパッケージを使用すると、PPA をより効率的に操作できます。

```
sudo apt-get install software-properties-common
```

> **Note**
>
> `sudo apt-get install` コマンドを実行したときに `Could not get lock` エラーが発生した場合は、約 15 分待ってから再試行してください。このエラーは、自動アップグレードをインストールするために Apt パッケージ管理ツールを使用している cron ジョブが原因で発生している可能性があります。

5. 次のコマンドを入力して Certbot をローカル apt リポジトリに追加します。

> **Note**
>
> ステップ 5 は、Ubuntu Linux ディストリビューションを使用するインスタンスにのみ適用されます。インスタンスが Debian Linux ディストリビューションを使用している場合は、このステップをスキップしてください。

```
sudo apt-add-repository ppa:certbot/certbot -y
```

6. 次のコマンドを入力して apt を更新し、新しいリポジトリを含めます。

```
sudo apt-get update -y
```

7. 次のコマンドを入力して Certbot をインストールします。

```
sudo apt-get install certbot -y
```

これで Lightsail インスタンスに Certbot がインストールされました。

8. ブラウザベースの SSH ターミナルウィンドウは開いたままにします。このチュートリアルで後ほど戻ります。このチュートリアルの「次のセクション」に進みます。

ステップ 3: Let's Encrypt の SSL ワイルドカード証明書をリクエストする

Let's Encrypt の証明書をリクエストするプロセスを開始します。Certbot を使用してワイルドカード証明書をリクエストします。この 1 つの証明書をドメインとそのサブドメインの両方に使用できます。たとえば、1 つのワイルドカード証明書を `example.com` 最上位ドメイン、`blog.example.com` サブドメイン、および `stuff.example.com` サブドメインに使用できます。

Let's Encrypt の SSL ワイルドカード証明書をリクエストするには

1. このチュートリアルのステップ 2 で使用した同じブラウザベースの SSH ターミナルウィンドウで、以下のコマンドを入力してドメインの環境変数を設定します。より効率的にコマンドをコピーして貼り付け、証明書を取得できます。*domain* を登録済みのドメイン名に置き換えます。

```
DOMAIN=domain
```

```
WILDCARD=*.$DOMAIN
```

例:

```
DOMAIN=example.com
```

```
WILDCARD=*.$DOMAIN
```

2. 次のコマンドを入力し、変数が正しい値を返すことを確認します。

```
echo $DOMAIN && echo $WILDCARD
```

次のような結果が表示されます。

3. 次のコマンドを入力して Certbot をインタラクティブモードで起動します。このコマンドでは、DNS チャレンジで手動認可を使用してドメインの所有権を検証することを Certbot に指示します。また、最上位ドメインとそのサブドメイン用にワイルドカード証明書をリクエストします。

```
sudo certbot -d $DOMAIN -d $WILDCARD --manual --preferred-challenges dns
 certonly
```

4. プロンプトに応じて E メールアドレスを入力します。これで更新とセキュリティに関する通知を受信します。

5. Let's Encrypt のサービス利用規約を読みます。読み終わり、同意する場合は A キーを押します。同意しない場合は、Let's Encrypt の証明書を取得できません。
6. E メールアドレスの共有と IP アドレスのログ記録に関するプロンプトに適宜応答します。
7. Let's Encrypt から、指定されたドメインの所有者であることの検証を求められます。これを行うには、ドメインの DNS レコードに TXT レコードを追加します。以下の例に示すように 2 組の TXT レコード値が提供されます。

   > **Note**
   >
   > Let's Encrypt では検証に必要な TXT レコードを 1 つまたは複数提供する場合があります。この例では、検証に使用する 2 つの TXT レコードが提供されました。

8. Lightsail のブラウザベースの SSH セッションは開いたままにします。このチュートリアルで後ほど戻ります。このチュートリアルの「次のセクション」に進みます。

ステップ 4: ドメインの DNS ゾーンに TXT レコードを追加する

ドメインの DNS ゾーンに TXT レコードを追加すると、自分がドメインを所有していることが検証されます。ここでは、デモの目的で Lightsail の DNS ゾーンを使用します。ただし、ドメインレジストラがホストする他の一般的な DNS ゾーンでも手順はほぼ同じです。

> **Note**
>
> ドメインの Lightsail DNS ゾーンの作成方法の詳細については、「Lightsail で DNS ゾーンを作成し、ドメインの DNS レコードを管理する」を参照してください。

Lightsail でドメインの DNS ゾーンに TXT レコードを追加するには

1. 左側のナビゲーションペインで [ドメインと DNS] を選択します。
2. ページの [DNS ゾーン] セクションで、Certbot 証明書リクエストで指定したドメインの DNS ゾーンを選択します。
3. DNS ゾーンエディタで [DNS records] (DNS レコード) を選択します。
4. [レコードの追加] を選択します。

5. [Record type] (レコードタイプ) のドロップダウンメニューで [TXT record] (TXT レコード) を選択します。
6. Let's Encrypt 証明書のリクエストで指定された値を [Record name] (レコード名) と [Responds with] (応答) フィールドに入力します。

   > **Note**
   >
   > Lightsail コンソールには、ドメインの頂点部分があらかじめ入力されています。たとえば、*_acme-challenge.example.com* サブドメインを追加する場合は、*_acme-challenge* をテキストボックスに入力するだけで、レコードを保存するときに Lightsail が `.example.com` の部分を追加します。

7. [保存] を選択します。
8. ステップ 4 ~ 7 を繰り返して、Let's Encrypt の証明書リクエストで指定された 2 番目の TXT レコードのセットを追加します。
9. Lightsail コンソールのブラウザウィンドウは、このチュートリアルで後ほど戻るので開いたままにします。このチュートリアルの「次のセクション」に進みます。

ステップ 5: TXT レコードが反映されたことを確認する

MxToolbox ユーティリティを使用して TXT レコードがインターネットの DNS に反映されたことを確認します。DNS レコードの反映には、DNS ホスティングプロバイダーと DNS レコードの有効期限 (TTL) の設定によって時間がかかる場合があります。このステップを完了し、TXT レコードが反映されたことを確認した上で、Certbot 証明書のリクエストに進むことが重要です。そうしないと、証明書のリクエストは失敗します。

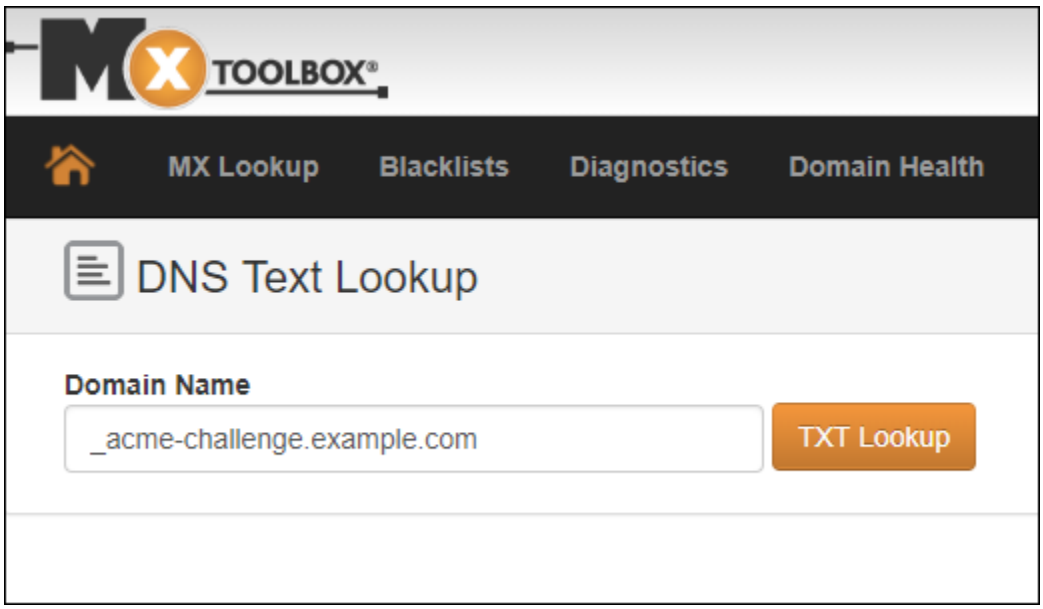
TXT レコードがインターネットの DNS に反映されたことを確認するには

1. 新しいブラウザウィンドウを開き、https://mxtoolbox.com/TXTLookup.aspx に移動します。
2. 次の内容をテキストボックスに入力します。*domain* は実際のドメインに置き換えてください。

   ```
   _acme-challenge.domain
   ```

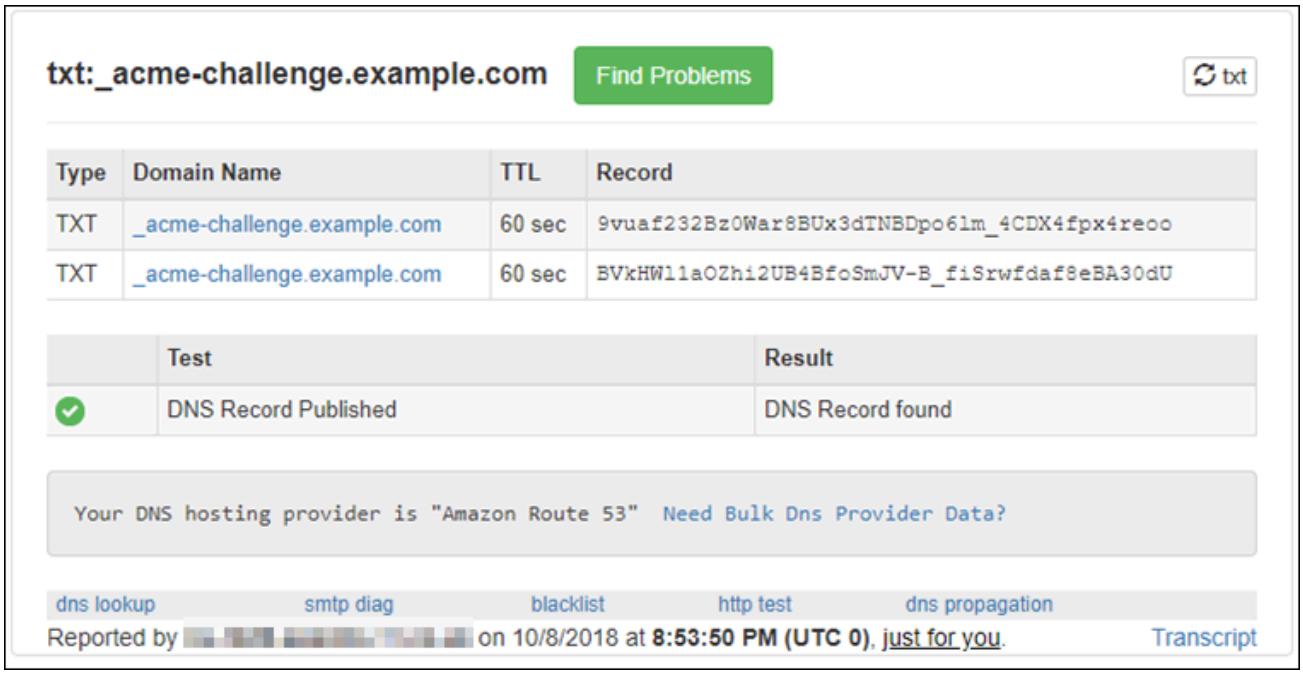
   例:

   ```
   _acme-challenge.example.com
   ```



3. [TXT Lookup (TXT ルックアップ)] を選択して確認を行います。
4. 以下のいずれかのレスポンスが返されます。

   - TXT レコードがインターネットの DNS に反映された場合は、次のスクリーンショットに示すようなレスポンスが表示されます。ブラウザウィンドウを閉じて、このチュートリアルの「次のセクション」に進みます。



   - TXT レコードがインターネットの DNS に反映されていない場合は、[DNS Record not found (DNS レコードが見つかりません)] というレスポンスが返されます。適切な DNS レ

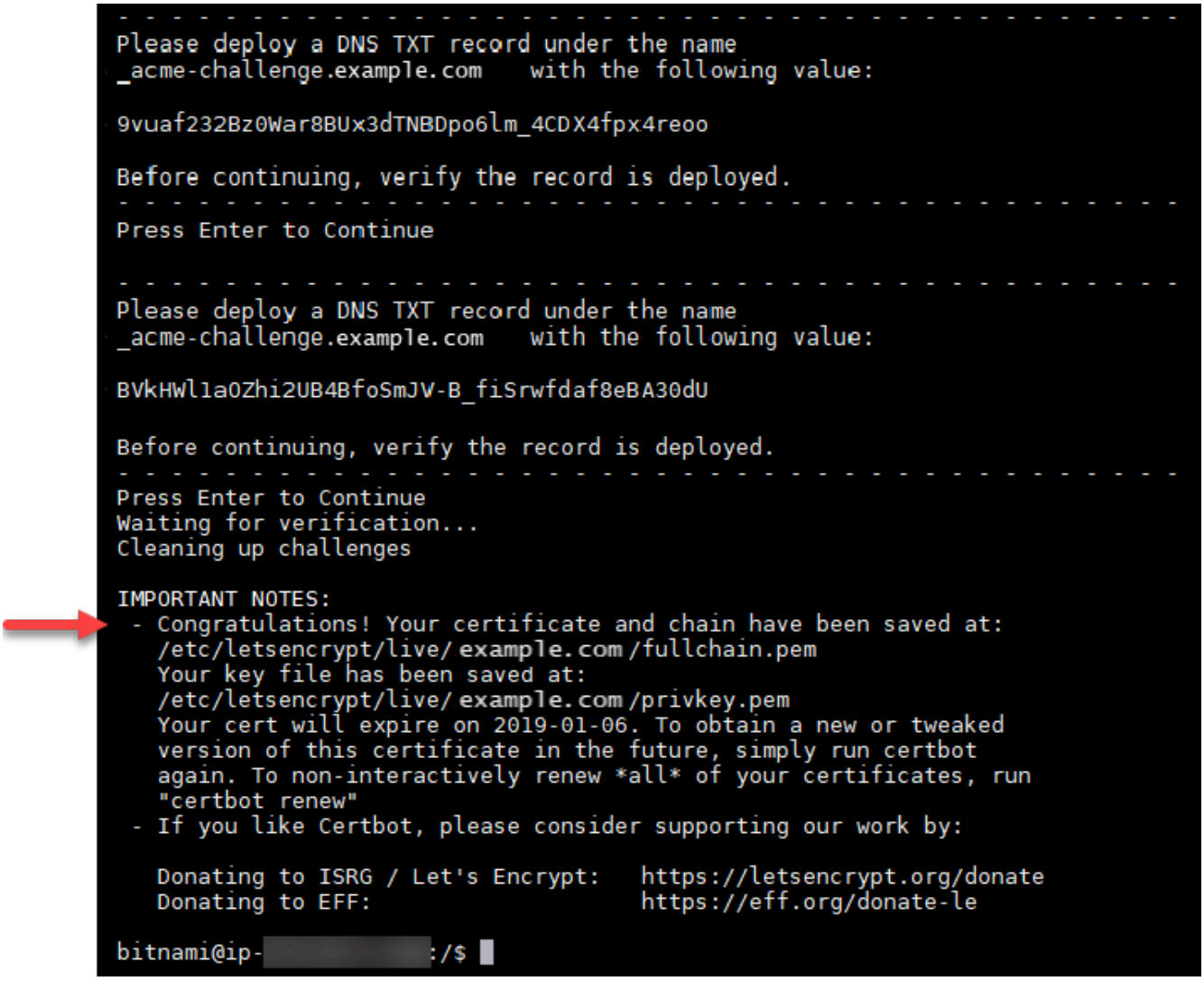
コードをドメインの DNS ゾーンに追加したことを確認してください。適切なレコードを追加した場合は、ドメインの DNS レコードが反映されるまでしばらく待ってから、TXT のルックアップを再実行します。

ステップ 6: Let's Encrypt の SSL 証明書リクエストを完了する

Nginx インスタンスの Lightsail ブラウザベースの SSH セッションに戻り、Let's Encrypt 証明書のリクエストを完了します。Certbot は、SSL 証明書、チェーン、およびキーファイルを Nginx インスタンスの特定のディレクトリに保存します。

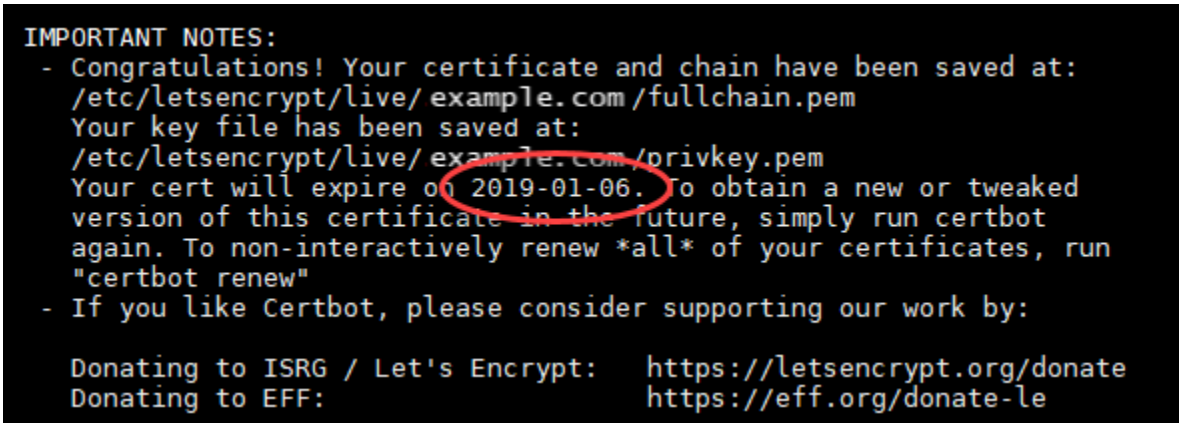
Let's Encrypt の SSL 証明書リクエストを完了するには

1. Nginx インスタンスの Lightsail ブラウザベースの SSH セッションで、Enter キーを押し、Let's Encrypt SSL 証明書のリクエストを続行します。成功すると、次のスクリーンショットに示すようなレスポンスが表示されます。



証明書、チェーン、およびキーファイルが /etc/letsencrypt/live/*domain*/ ディレクトリに保存されたことを確認するメッセージが表示されます。*domain* は、実際のドメイン (/etc/letsencrypt/live/*example.com*/ など) に置き換えてください。

2. メッセージに記載されている有効期限を書き留めておきます。この期限日までに証明書を更新する必要があります。



3. これで Let's Encrypt SSL 証明書が手に入ったので、このチュートリアルの「次のセクション」に進みます。

ステップ 7: NGINX サーバーディレクトリに Let's Encrypt 証明書ファイルへのリンクを作成する

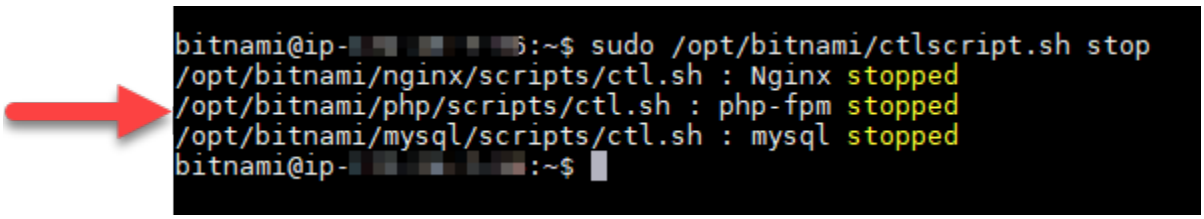
Nginx インスタンスの NGINX サーバーディレクトリに Let's Encrypt SSL 証明書ファイルへのリンクを作成します。また、必要になる場合に備えて既存の証明書をバックアップします。

NGINX サーバーディレクトリに Let's Encrypt 証明書ファイルへのリンクを作成するには

1. Nginx インスタンスの Lightsail ブラウザベースの SSH セッションで、次のコマンドを入力して基本サービスを停止します。

```
sudo /opt/bitnami/ctlscript.sh stop
```

次のようなレスポンスが表示されます。



2. 次のコマンドを入力してドメインの環境変数を設定します。コマンドのコピー＆ペーストで、より効率的に証明書ファイルにリンクを張れます。*domain* は登録済みのドメイン名に置き換えてください。

```
DOMAIN=domain
```

例:

```
DOMAIN=example.com
```

3. 次のコマンドを入力し、変数が正しい値を返すことを確認します。

```
echo $DOMAIN
```

次のような結果が表示されます。

4. バックアップとして既存の証明書ファイルがある場合、以下のコマンドを個別に入力して名前を書き換えます。さまざまなディストリビューションとファイル構造の詳細については、このチュートリアルの冒頭の重要ブロックを参照してください。

- Debian Linux ディストリビューションの場合

  アプローチ A (システムパッケージを使用した Bitnami インストール):

```
sudo mv /opt/bitnami/nginx/conf/bitnami/certs/server.crt /opt/bitnami/nginx/
conf/bitnami/certs/server.crt.old
```

```
sudo mv /opt/bitnami/nginx/conf/bitnami/certs/server.key /opt/bitnami/nginx/
conf/bitnami/certs/server.key.old
```

  アプローチ B (自己完結型 Bitnami インストール):

```
sudo mv /opt/bitnami/nginx/conf/server.crt /opt/bitnami/nginx/conf/
server.crt.old
```

```
sudo mv /opt/bitnami/nginx/conf/server.key /opt/bitnami/nginx/conf/
server.key.old
```

- Ubuntu Linux ディストリビューションを使用する古いインスタンスの場合：

```
sudo mv /opt/bitnami/nginx/conf/bitnami/certs/server.crt /opt/bitnami/nginx/
conf/bitnami/certs/server.crt.old
```

```
sudo mv /opt/bitnami/nginx/conf/bitnami/certs/server.key /opt/bitnami/nginx/
conf/bitnami/certs/server.key.old
```

5. 次のコマンドを個別に入力して、NGINX サーバーディレクトリに Let's Encrypt 証明書ファイルへのリンクを作成します。さまざまなディストリビューションとファイル構造の詳細については、このチュートリアルの冒頭の重要ブロックを参照してください。

   - Debian Linux ディストリビューションの場合

     アプローチ A (システムパッケージを使用した Bitnami インストール):

     ```
     sudo ln -sf /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/nginx/conf/
     bitnami/certs/server.key
     ```

     ```
     sudo ln -sf /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/nginx/
     conf/bitnami/certs/server.crt
     ```

     アプローチ B (自己完結型 Bitnami インストール):

     ```
     sudo ln -sf /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/nginx/conf/
     server.key
     ```

     ```
     sudo ln -sf /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/nginx/
     conf/server.crt
     ```

   - Ubuntu Linux ディストリビューションを使用する古いインスタンスの場合：

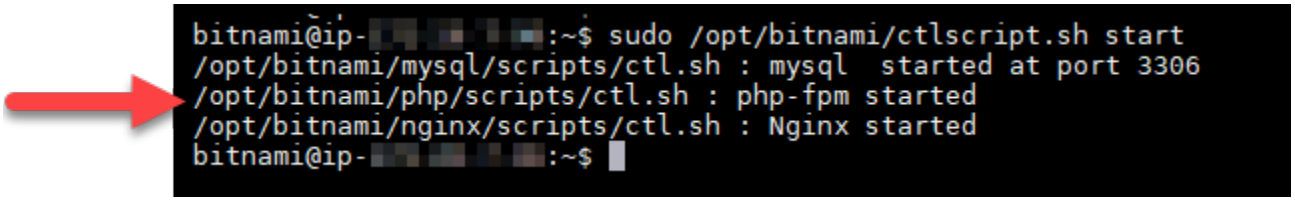
     ```
     sudo ln -s /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/nginx/conf/
     bitnami/certs/server.key
     ```

     ```
     sudo ln -s /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/nginx/
     conf/bitnami/certs/server.crt
     ```

6. 次のコマンドを入力して、先ほど停止した基本サービスを開始します。

   ```
   sudo /opt/bitnami/ctlscript.sh start
   ```

   次のような結果が表示されます。



これで SSL 暗号化を使用するように Nginx インスタンスが設定されました。ただし、トラフィックは HTTP から HTTPS に自動的にはリダイレクトされません。

7. このチュートリアルの「次のセクション」に進みます。

ステップ 8: ウェブアプリケーションの HTTP から HTTPS へのリダイレクトを設定する

Nginx インスタンスの HTTP から HTTPS へのリダイレクトを設定することができます。HTTP から HTTPS へのリダイレクトを自動的に行うことで、SSL を使用するユーザーにのみ (HTTP を使用して接続した場合でも) サイトへのアクセスを許可できます。さまざまなディストリビューションとファイル構造の詳細については、このチュートリアルの冒頭の重要ブロックを参照してください。

このチュートリアルではデモの目的で Vim を使用していますが、任意のテキストエディタを使用できます。

Debian Linux ディストリビューション – ウェブアプリケーションの HTTP から HTTPS へのリダイレクトを設定する

1. Nginx インスタンスの Lightsail ブラウザベースの SSH セッションで、次のコマンドを入力し、サーバーブロック設定ファイルを変更します。アプリケーションの名前を <ApplicationName> に置き換えます。

```
sudo vim /opt/bitnami/nginx/conf/server_blocks/<ApplicationName>-server-
block.conf
```

2. i キーを押して Vim エディタを挿入モードにします。
3. 次の例の情報を使用してファイルを編集します。

```
server {
    listen 80 default_server;
    root /opt/bitnami/APPNAME;
    return 301 https://$host$request_uri;
}
```

4. ESC キーを押して「:wq」と入力し、編集内容を書き込んで (保存して) Vim を終了します。
5. 次のコマンドを入力して、NGINX 設定ファイルのサーバーセクションを変更します。

```
sudo vim /opt/bitnami/nginx/conf/nginx.conf
```

6. i キーを押して Vim エディタを挿入モードにします。
7. 次の例の情報を使用してファイルを編集します。

```
server {
      listen 80;
      server_name localhost;
      return 301 https://$host$request_uri;
  }
```

8. ESC キーを押して「:wq」と入力し、編集内容を書き込んで (保存して) Vim を終了します。
9. 次のコマンドを入力して基本サービスを再開し、編集内容を反映させます。

```
sudo /opt/bitnami/ctlscript.sh restart
```

アプローチ B (自己完結型 Bitnami インストール):

1. Nginx インスタンスのLightsailブラウザベースの SSH セッションで、次のコマンドを入力して NGINX 設定ファイルのサーバーセクションを変更します。

```
sudo vim /opt/bitnami/nginx/conf/nginx.conf
```

2. i キーを押して Vim エディタを挿入モードにします。
3. 次の例の情報を使用してファイルを編集します。

```
server {
      listen 80;
      server_name localhost;
      return 301 https://$host$request_uri;
  }
```

4. ESC キーを押して「:wq」と入力し、編集内容を書き込んで (保存して) Vim を終了します。
5. 次のコマンドを入力して基本サービスを再開し、編集内容を反映させます。

```
sudo /opt/bitnami/ctlscript.sh restart
```

Ubuntu Linux ディストリビューションを使用する古いインスタンスの場合 – ウェブアプリケーションの HTTP から HTTPS へのリダイレクトを設定する

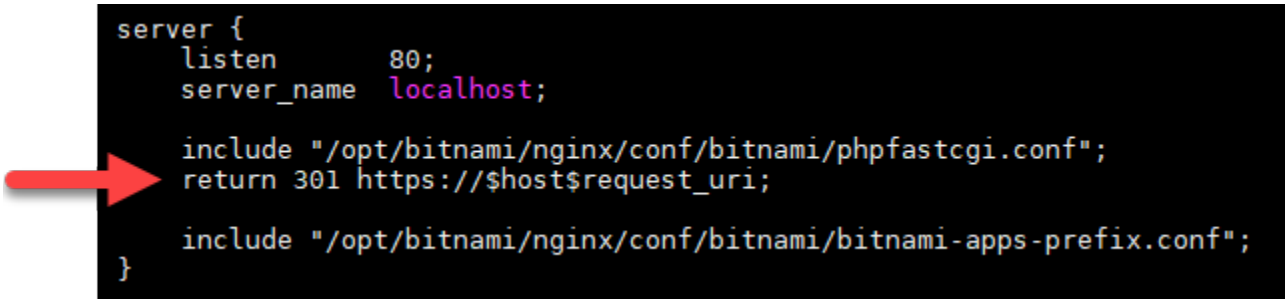
1. Nginx インスタンスのLightsailブラウザベースの SSH セッションで、次のコマンドを入力して、Vim テキストエディタを使用して NGINX ウェブサーバー設定ファイルを編集します。

   ```
   sudo vim /opt/bitnami/nginx/conf/bitnami/bitnami.conf
   ```

2. i キーを押して Vim エディタを挿入モードにします。
3. このファイルで、`server_name localhost;` と `include "/opt/bitnami/nginx/conf/bitnami/bitnami-apps-prefix.conf";` の間に次のテキストを入力します。

   ```
   return 301 https://$host$request_uri;
   ```

   結果は次のようになります。

4. ESC キーを押して「:wq」と入力し、編集内容を書き込んで (保存して) Vim を終了します。
5. 次のコマンドを入力して基本サービスを再開し、編集内容を反映させます。

   ```
   sudo /opt/bitnami/ctlscript.sh restart
   ```

   これで、HTTP から HTTPS へ自動的に接続をリダイレクトするように Nginx インスタンスが設定されました。訪問者が `http://www.example.com` にアクセスすると、暗号化された `https://www.example.com` アドレスに自動的にリダイレクトされます。

ステップ 9: Let's Encrypt 証明書を 90 日ごとに更新する

Let's Encrypt 証明書の有効期間は 90 日間です。証明書は有効期限が切れる 30 日前から更新できます。Let's Encrypt 証明書を更新するには、取得するために使用した元のコマンドを実行します。このチュートリアルの「Let's Encrypt の SSL ワイルドカード証明書をリクエストする」セクションのステップを繰り返します。

Lightsail

目次

- ステップ 1: 前提条件を完了する
- ステップ 2: Lightsailインスタンスに Certbot をインストールする
- ステップ 3: Let's Encrypt SSL ワイルドカード証明書をリクエストする
- ステップ 4: TXT レコードをドメインの DNS ゾーンに追加する
- ステップ 5: TXT レコードが伝播されたことを確認する
- ステップ 6: Let's Encrypt SSL 証明書リクエストを完了する
- ステップ 7: NGINX で SSL 設定を更新し、トラフィックを HTTP から HTTPS にリダイレクトする
- ステップ 8: Let's Encrypt 証明書を 90 日ごとに更新する

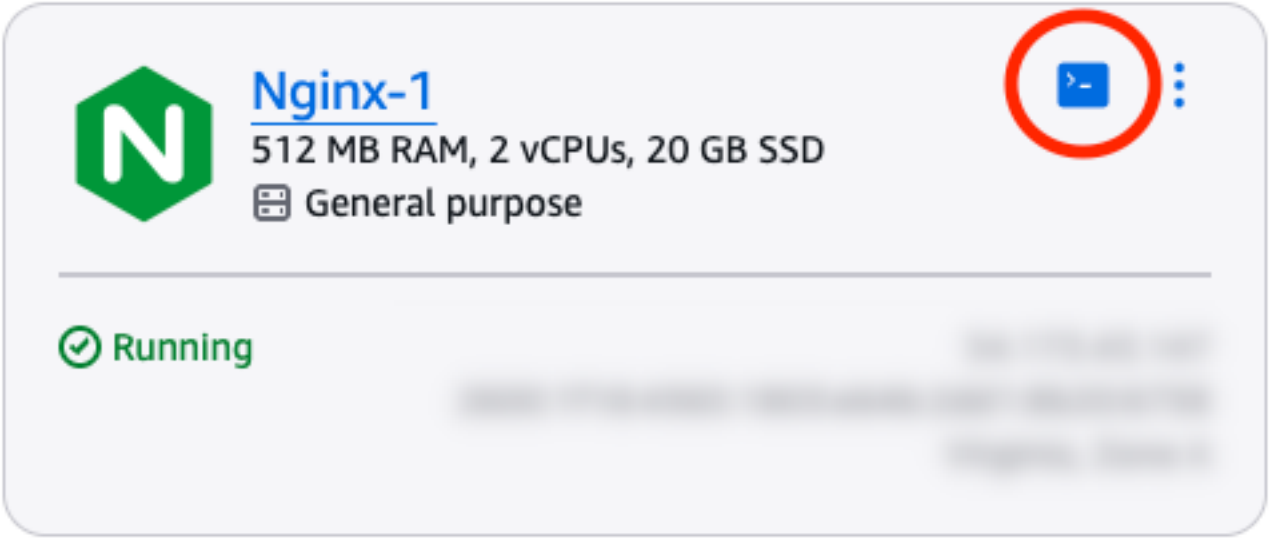
ステップ 1: 前提条件を満たす

以下の前提条件を満たします (まだ満たしていない場合)。

- Lightsail に Nginx インスタンスを作成します。詳細については、「インスタンスの作成」を参照してください。
- ドメイン名を登録し、その DNS レコードを編集するための管理アクセスを取得します。詳細については、「DNS」を参照してください。

> **Note**
>
> ドメインの DNS レコードは、Lightsail の DNS ゾーンを使用して管理することをお勧めします。詳細については、「ドメインの DNS レコードを管理する DNS ゾーンを作成する」を参照してください。

- Lightsail コンソールでブラウザベースの SSH ターミナルを使用して、このチュートリアルの手順を実行します。



> **Note**
>
> PuTTY などの独自の SSH クライアントを使用することもできます。PuTTY の設定の詳細については、「」の「Download and set up PuTTY to connect using SSH」を参照してくださいAmazon Lightsail。

ステップ 2: Lightsail インスタンスに Certbot をインストールする

Certbot は、Let's Encrypt の証明書をリクエストしてウェブサーバーにデプロイするために使用するクライアントです。Let's Encrypt は ACME プロトコルを使用して証明書を発行します。Certbot は Let's Encrypt とやり取りする ACME 対応クライアントです。

Lightsail インスタンスに Certbot をインストールするには

1. Lightsail ブラウザベースの SSH ターミナルなど、SSH クライアントを使用してインスタンスに接続します。インスタンスのパッケージを更新するには、次のコマンドを入力します。

```
sudo apt-get update
```

2. 次のコマンドを入力してソフトウェアプロパティパッケージをインストールします。Certbot の開発者は、Personal Package Archive (PPA) を使用して Cerbot を配信します。ソフトウェアプロパティパッケージを使用すると、PPA をより効率的に操作できます。

```
sudo apt-get install software-properties-common -y
```

3. 次のコマンドを入力して apt を更新し、新しいリポジトリを含めます。

```
sudo apt-get update -y
```

4. 次のコマンドを入力して Certbot をインストールします。

```
sudo apt-get install certbot -y
```

これで Lightsail インスタンスに Certbot がインストールされました。

ステップ 3: Let's Encrypt の SSL ワイルドカード証明書をリクエストする

Let's Encrypt の証明書をリクエストするプロセスを開始します。Certbot を使用してワイルドカード証明書をリクエストします。この 1 つの証明書をドメインとそのサブドメインの両方に使用できます。たとえば、1 つのワイルドカード証明書を `example.com` 最上位ドメイン、`blog.example.com` サブドメイン、および `stuff.example.com` サブドメインに使用できます。

Let's Encrypt の SSL ワイルドカード証明書をリクエストするには

1. このチュートリアルのステップ 2 で使用した同じブラウザベースの SSH ターミナルウィンドウで、以下のコマンドを入力してドメインの環境変数を設定します。より効率的にコマンドをコピーして貼り付け、証明書を取得できます。*`domain`* を登録済みのドメイン名に置き換えます。

```
DOMAIN=domain
WILDCARD=*.$DOMAIN
```

例:

```
DOMAIN=example.com
WILDCARD=*.$DOMAIN
```

2. 次のコマンドを入力し、変数が正しい値を返すことを確認します。

```
echo $DOMAIN && echo $WILDCARD
```

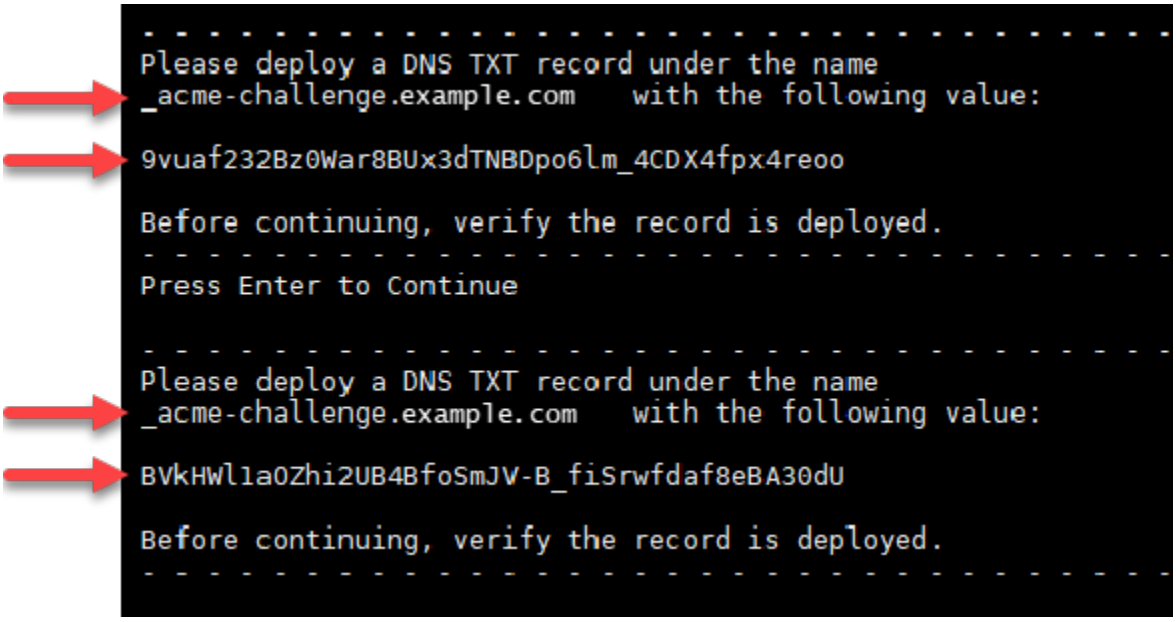
次のような結果が表示されます。

3. 次のコマンドを入力して Certbot をインタラクティブモードで起動します。このコマンドでは、DNS チャレンジで手動認可を使用してドメインの所有権を検証することを Certbot に指示します。また、最上位ドメインとそのサブドメイン用にワイルドカード証明書をリクエストします。

```
sudo certbot -d $DOMAIN -d $WILDCARD --manual --preferred-challenges dns
 certonly
```

4. プロンプトに応じて E メールアドレスを入力します。これで更新とセキュリティに関する通知を受信します。
5. Let's Encrypt のサービス利用規約を読みます。完了したら、同意したら Y を押します。同意しない場合は、Let's Encrypt の証明書を取得できません。
6. E メールアドレスの共有と IP アドレスのログ記録に関するプロンプトに適宜応答します。
7. Let's Encrypt から、指定されたドメインの所有者であることの検証を求められます。これを行うには、ドメインの DNS レコードに TXT レコードを追加します。以下の例に示すように 2 組の TXT レコード値が提供されます。

> **Note**
>
> Let's Encrypt は、検証に使用する必要がある 1 つまたは複数の TXT レコードを提供する場合があります。この例では、検証に使用する 2 つの TXT レコードが提供されました。



8. Lightsail のブラウザベースの SSH セッションは開いたままにします。このチュートリアルで後ほど戻ります。このチュートリアルの次のセクションに進みます。

ステップ 4: ドメインの DNS ゾーンに TXT レコードを追加する

ドメインの DNS ゾーンに TXT レコードを追加すると、自分がドメインを所有していることが検証されます。ここでは、デモの目的で Lightsail の DNS ゾーンを使用します。ただし、ドメインレジストラがホストする他の一般的な DNS ゾーンでも手順はほぼ同じです。

> **Note**
>
> ドメインの LightsailDNS ゾーンを作成する方法の詳細については、「 でドメインの DNS レコードを管理する DNS ゾーンの作成」を参照してくださいLightsail。

Lightsail でドメインの DNS ゾーンに TXT レコードを追加するには

1. 左側のナビゲーションペインで [ドメインと DNS] を選択します。
2. ページの [DNS ゾーン] セクションで、Certbot 証明書リクエストで指定したドメインの DNS ゾーンを選択します。
3. DNS ゾーンエディタで [DNS records] (DNS レコード) を選択します。
4. [レコードの追加] を選択します。

5. [Record type] (レコードタイプ) のドロップダウンメニューで [TXT record] (TXT レコード) を選択します。
6. Let's Encrypt 証明書のリクエストで指定された値を [Record name] (レコード名) と [Responds with] (応答) フィールドに入力します。

   > **Note**
   >
   > Lightsail コンソールには、ドメインの頂点部分があらかじめ入力されています。たとえば、*_acme-challenge.example.com* サブドメインを追加する場合は、*_acme-challenge* をテキストボックスに入力するだけで、レコードを保存するときに Lightsail が `.example.com` の部分を追加します。

7. [保存] を選択します。
8. ステップ 4 ~ 7 を繰り返して、Let's Encrypt の証明書リクエストで指定された 2 番目の TXT レコードのセットを追加します。
9. Lightsail コンソールのブラウザウィンドウは、このチュートリアルで後ほど戻るので開いたままにします。このチュートリアルの次のセクションに進みます。

ステップ 5: TXT レコードが反映されたことを確認する

MxToolbox ユーティリティを使用して TXT レコードがインターネットの DNS に反映されたことを確認します。DNS レコードの反映には、DNS ホスティングプロバイダーと DNS レコードの有効期限 (TTL) の設定によって時間がかかる場合があります。このステップを完了し、TXT レコードが反映されたことを確認した上で、Certbot 証明書のリクエストに進むことが重要です。そうしないと、証明書のリクエストは失敗します。

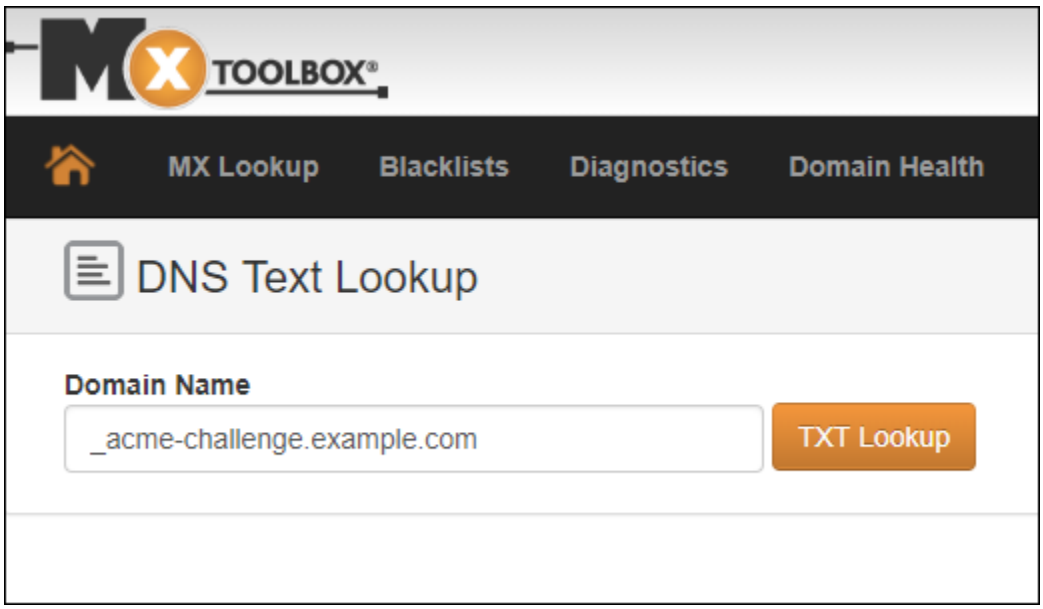
TXT レコードがインターネットの DNS に反映されたことを確認するには

1. 新しいブラウザウィンドウを開き、https://mxtoolbox.com/TXTLookup.aspx に移動します。
2. 次の内容をテキストボックスに入力します。*domain* は実際のドメインに置き換えてください。

   ```
   _acme-challenge.domain
   ```

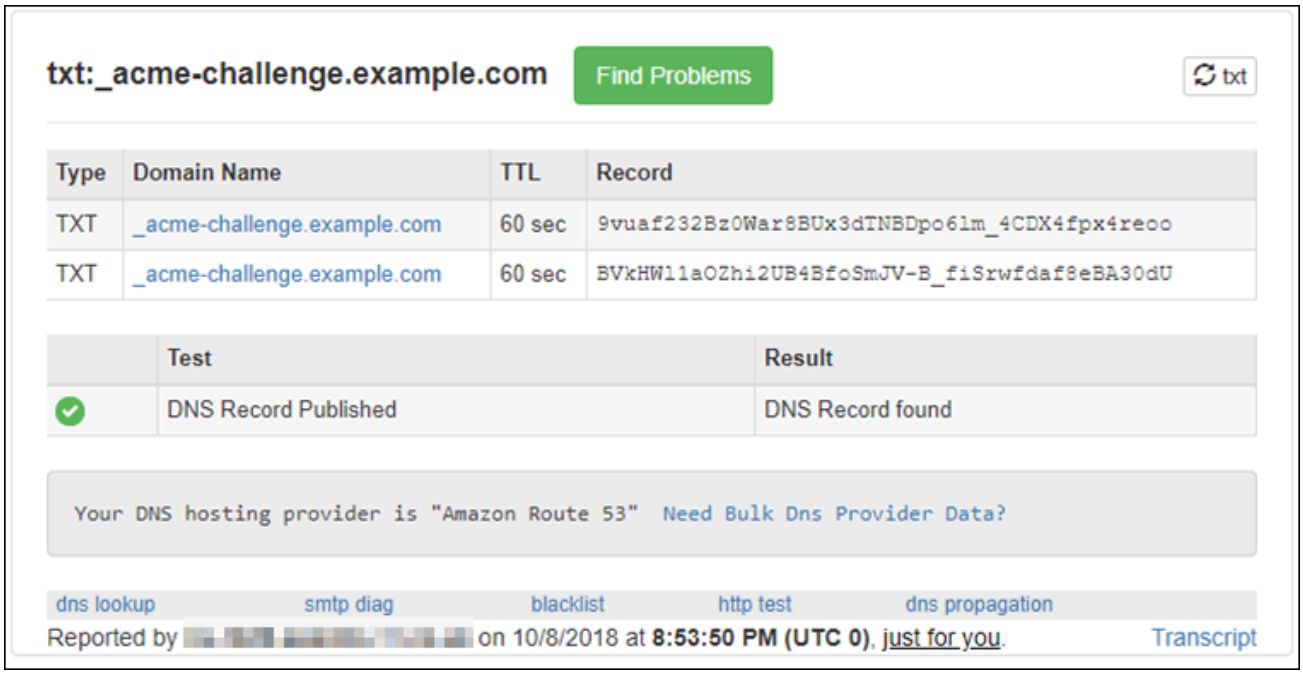
   例:

   ```
   _acme-challenge.example.com
   ```



3. [TXT Lookup (TXT ルックアップ)] を選択して確認を行います。
4. 以下のいずれかのレスポンスが返されます。

   - TXT レコードがインターネットの DNS に反映された場合は、次のスクリーンショットに示すようなレスポンスが表示されます。ブラウザウィンドウを閉じて、このチュートリアルの「次のセクション」に進みます。



   - TXT レコードがインターネットの DNS に反映されていない場合は、[DNS Record not found (DNS レコードが見つかりません)] というレスポンスが返されます。適切な DNS レ

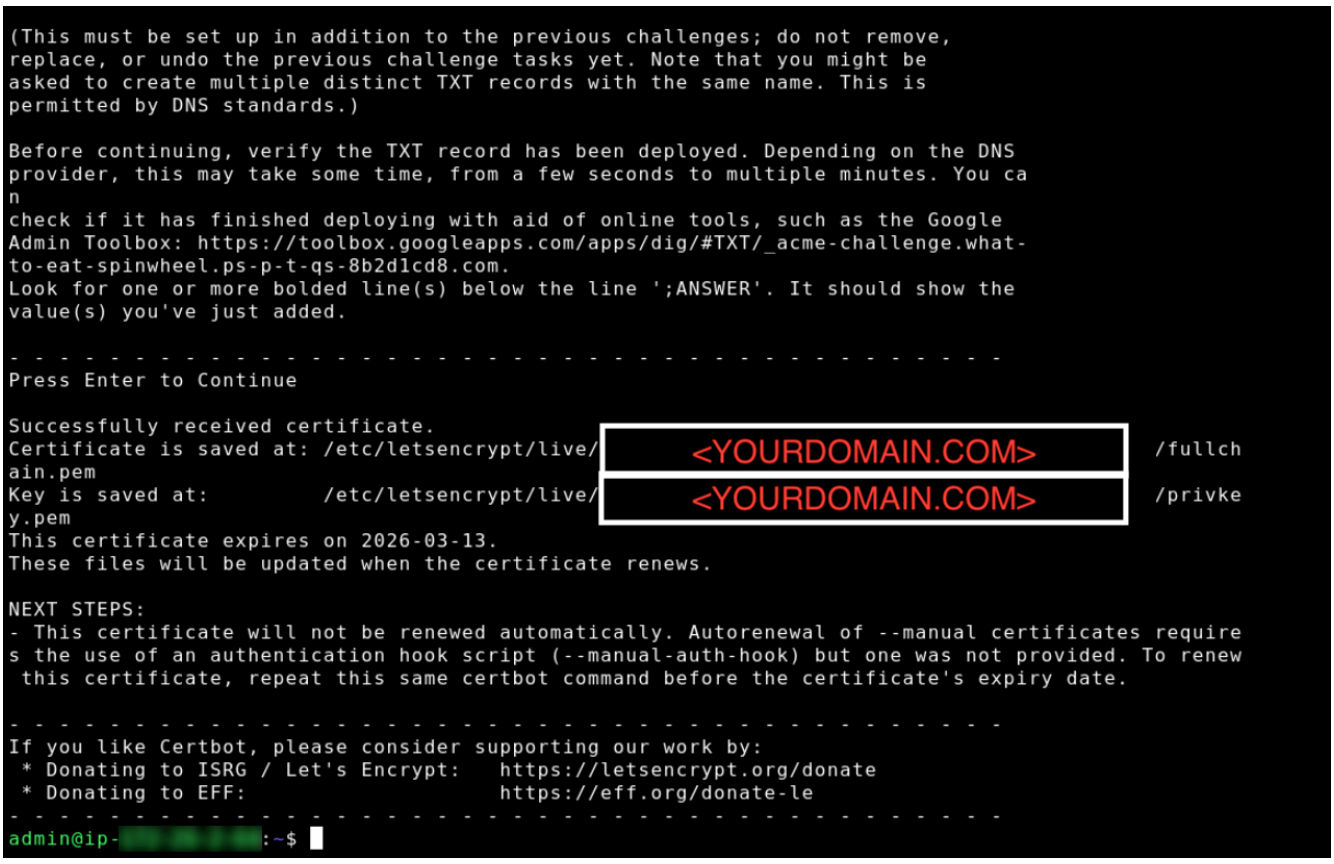
コードをドメインの DNS ゾーンに追加したことを確認してください。適切なレコードを追加した場合は、ドメインの DNS レコードが反映されるまでしばらく待ってから、TXT のルックアップを再実行します。

ステップ 6: Let's Encrypt の SSL 証明書リクエストを完了する

インスタンスのLightsailブラウザベースの SSH セッションに戻り、Let's Encrypt 証明書リクエストを完了します。Certbot は、SSL 証明書、チェーン、およびキーファイルをインスタンス上の特定のディレクトリに保存します。

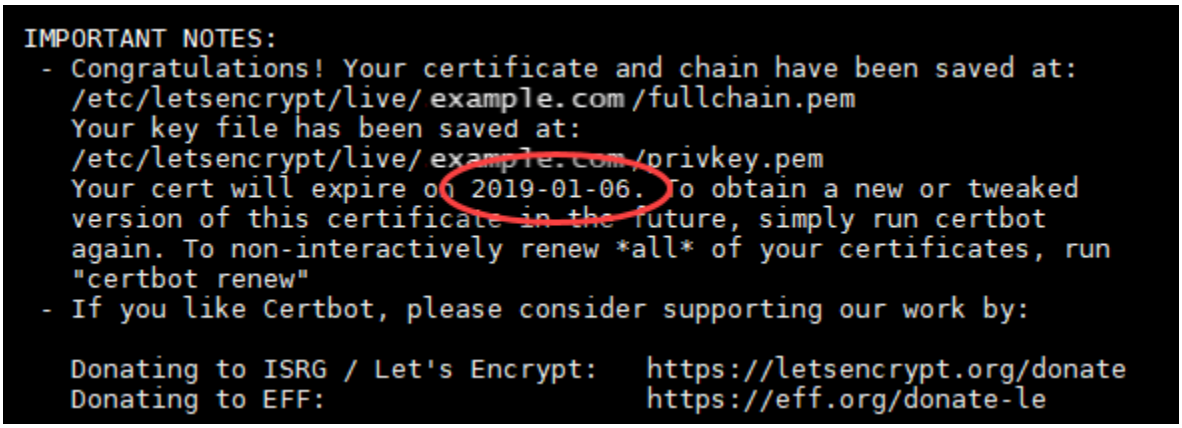
Let's Encrypt の SSL 証明書リクエストを完了するには

1. インスタンスのLightsailブラウザベースの SSH セッションで、Enter キーを押して Let's Encrypt SSL 証明書リクエストを続行します。成功すると、次のスクリーンショットに示すようなレスポンスが表示されます。



証明書、チェーン、およびキーファイルが /etc/letsencrypt/live/*domain*/ ディレクトリに保存されたことを確認するメッセージが表示されます。*domain* は、実際のドメイン (/etc/letsencrypt/live/*example.com*/ など) に置き換えてください。

2. メッセージに記載されている有効期限を書き留めておきます。この期限日までに証明書を更新する必要があります。



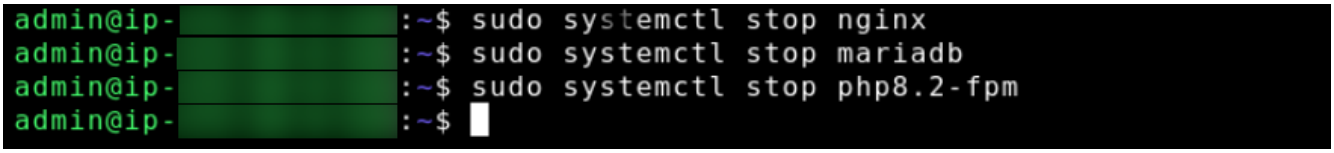
ステップ 7: NGINX で SSL 設定を更新し、トラフィックを HTTP から HTTPS にリダイレクトする

NGINX の default.conf で SSL 設定を更新するには

1. Nginx インスタンスの SSH セッションで、次のコマンドを入力して基盤となるサービスを停止します。

```
sudo systemctl stop nginx
sudo systemctl stop mariadb
sudo systemctl stop php8.2-fpm
```

次のようなレスポンスが表示されます。



2. 次のコマンドを入力してドメインの環境変数を設定します。コマンドのコピー＆ペーストで、より効率的に証明書ファイルにリンクを張れます。*domain* は登録済みのドメイン名に置き換えてください。

```
DOMAIN=domain
```

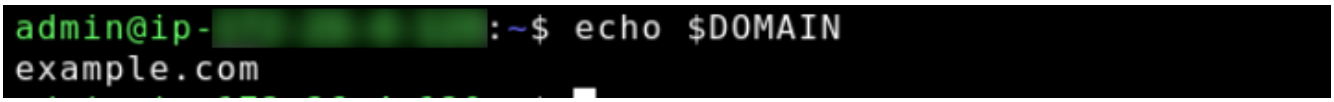
例:

```
DOMAIN=example.com
```

3. 次のコマンドを入力し、変数が正しい値を返すことを確認します。

```
echo $DOMAIN
```

次のような結果が表示されます。



4. 以下のコマンドを実行して SSL 設定を変更します。

```
sudo sed \
-i -e "s|ssl_certificate /etc/ssl/certs/ssl-cert-snakeoil.pem|ssl_certificate /
etc/letsencrypt/live/$DOMAIN/fullchain.pem|g" \
-i -e "s|ssl_certificate_key /etc/ssl/private/ssl-cert-snakeoil.key|
ssl_certificate_key /etc/letsencrypt/live/$DOMAIN/privkey.pem|g" \
/etc/nginx/conf.d/default.conf
```

5. default.conf ファイルを上書きしたら、以下のコマンドを実行して設定を確認し、NGINX を再起動します。

```
sudo nginx -t
sudo systemctl restart nginx
sudo systemctl restart mariadb
sudo systemctl restart php8.2-fpm
```

次のような結果が表示されます。

Nginx インスタンスが SSL 暗号化を使用するように設定され、トラフィックが HTTP から HTTPS にリダイレクトされるようになりました。

ステップ 8: Let's Encrypt 証明書を 90 日ごとに更新する

Let's Encrypt 証明書の有効期間は 90 日間です。証明書は有効期限が切れる 30 日前から更新できます。Let's Encrypt 証明書を更新するには、ステップ 3: Let's Encrypt SSL ワイルドカード証明書をリクエストする を繰り返します。

Let's Encrypt の無料 SSL 証明書で Lightsail の WordPress インスタンスを保護する

> **Tip**
>
> Amazon Lightsail は、WordPress インスタンスでの Let's Encrypt 証明書のインストールと設定を自動化するガイド付きのワークフローを提供します。このチュートリアルの手動手順を実行する代わりに、このワークフローを使用することが強く推奨されます。詳細については、「Launch and configure a WordPress instance」を参照してください。

Lightsail では、Lightsail ロードバランサーを使用すると、SSL/TLS でウェブサイトとアプリケーションのセキュリティを簡単に強化できます。ただし、Lightsail ロードバランサーの使用は一般的に最適な選択肢ではない場合があります。お使いのサイトではロードバランサーが提供するスケーラビリティや耐障害性が必要ない、またはコストのために最適化しているという可能性があります。後者の場合は、Let's Encrypt で無料の SSL 証明書を入手できます。無料の証明書を使用することに問題はありません。これらの証明書は Lightsail インスタンスに統合できます。

このガイドでは、Certbot を使用して Let's Encrypt ワイルドカード証明書をリクエストし、Really Simple SSL プラグインを使用して WordPress インスタンスと統合する方法について説明します。

- Bitnami インスタンスで使用されている Linux ディストリビューションは、2020 年 7 月に Ubuntu から Debian に変更されました。この変更により、このチュートリアルのいくつかのステップは、インスタンスの Linux ディストリビューションによって異なります。変更後に作成された Bitnami ブループリントインスタンスはすべて Debian Linux ディストリビューションを使用します。変更前に作成されたインスタンスは、Ubuntu Linux ディストリビューションを引き続き使用します。インスタンスのディストリビューションをチェックするには、`uname -a` コマンドを実行します。応答には、インスタンスの Linux ディストリビューションとして Ubuntu または Debian のいずれかが表示されます。
- Bitnami は、多くのスタックのファイル構造を変更するプロセスです。このチュートリアルのファイルパスは、Bitnami スタックがネイティブ Linux システムパッケージを使用しているか (アプローチ A)、または自己完結型インストール (アプローチ B) であるかによって、変更される場合があります。Bitnami のインストールタイプと取るべき方法を特定するには、次のコマンドを実行します。

  ```
  test ! -f "/opt/bitnami/common/bin/openssl" && echo "Approach A: Using system packages." || echo "Approach B: Self-contained installation."
  ```

目次

- チュートリアルを開始する前に
- ステップ 1: 前提条件を満たす
- ステップ 2: Lightsail インスタンスに Certbot をインストールする
- ステップ 3: Let's Encrypt の SSL ワイルドカード証明書をリクエストする
- ステップ 4: ドメインの DNS ゾーンに TXT レコードを追加する
- ステップ 5: TXT レコードが反映されたことを確認する
- ステップ 6: Let's Encrypt の SSL 証明書リクエストを完了する
- ステップ 7: Apache サーバーディレクトリで Let's Encrypt の証明書ファイルへのリンクを作成する
- ステップ 8: Really Simple SSL プラグインを使用して SSL 証明書を WordPress サイトに統合する
- ステップ 9: Let's Encrypt 証明書を 90 日ごとに更新する

チュートリアルを開始する前に

このチュートリアルを開始する前に、以下の点を考慮する必要があります。

Bitnami HTTPS 設定 (**`bncert`**) ツールを代わりに使用

このチュートリアルに記載されている手順は、手動プロセスを使用して SSL/TLS 証明書を実装する方法を説明しています。ただし、Bitnami は、通常 Lightsail 内の WordPress インスタンスにプリインストールされている Bitnami HTTPS 設定 (`bncert`) ツールを使用する、より自動化されたプロセスを提供します。このチュートリアルの手動手順を実行する代わりに、このツールを使用することが強く推奨されます。このチュートリアルは、`bncert` ツールが利用可能になる前に作成されたものです。`bncert` を使ったツールの詳細については、「Amazon Lightsail のWordPress インスタンスでの HTTPS の有効化」を参照してください。

WordPress インスタンスの Linux ディストリビューションを特定する

Bitnami インスタンスで使用されている Linux ディストリビューションは、2020 年 7 月に Ubuntu から Debian に変更されました。変更後に作成された Bitnami ブループリントインスタンスはすべて Debian Linux ディストリビューションを使用します。変更前に作成されたインスタンスは、Ubuntu Linux ディストリビューションを引き続き使用します。この変更により、このチュートリアルのいくつかのステップは、インスタンスの Linux ディストリビューションによって異なります。このチュートリアルでどの手順を使用するのかを把握するために、インスタンスの Linux ディストリビューショ

ンを特定する必要があります。インスタンスのディストリビューションを特定するには、`uname -a` コマンドを実行します。応答には、インスタンスの Linux ディストリビューションとして Ubuntu または Debian のいずれかが表示されます。

インスタンスに適用されるチュートリアルアプローチを特定する

Bitnami は、多くのスタックのファイル構造を変更するプロセスです。このチュートリアルのファイルパスは、Bitnami スタックがネイティブ Linux システムパッケージを使用しているか (アプローチ A)、または自己完結型インストール (アプローチ B) であるかによって、変更される場合があります。Bitnami のインストールタイプと取るべき方法を特定するには、次のコマンドを実行します。

```
test ! -f "/opt/bitnami/common/bin/openssl" && echo "Approach A: Using system packages." || echo "Approach B: Self-contained installation."
```

ステップ 1: 前提条件を満たす

以下の前提条件を満たします (まだ満たしていない場合)。

- Lightsail で WordPress インスタンスを作成します。詳細については、「インスタンスを作成する」を参照してください。
- ドメイン名を登録し、その DNS レコードを編集するための管理アクセスを取得します。詳細については、「DNS」を参照してください。

  ドメインの DNS レコードは、Lightsail の DNS ゾーンを使用して管理することをお勧めします。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。
- Lightsail コンソールでブラウザベースの SSH ターミナルを使用して、このチュートリアルのステップを実行します。ただし、独自の SSH クライアント (PuTTY など) を使用することもできます。PuTTY の設定の詳細については、「Amazon Lightsail で PuTTY をダウンロードし、SSH を使用して接続するようにセットアップする」を参照してください。

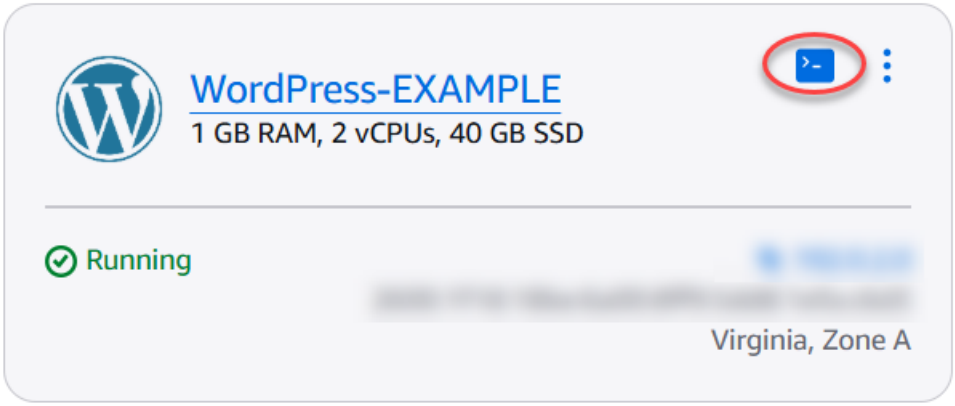
前提条件が完了したら、このチュートリアルの「次のセクション」に進みます。

ステップ 2: Lightsail インスタンスに Certbot をインストールする

Certbot は、Let's Encrypt の証明書をリクエストしてウェブサーバーにデプロイするために使用するクライアントです。Let's Encrypt は ACME プロトコルを使用して証明書を発行します。Certbot は、Let's Encrypt とやり取りする ACME 対応のクライアントです。

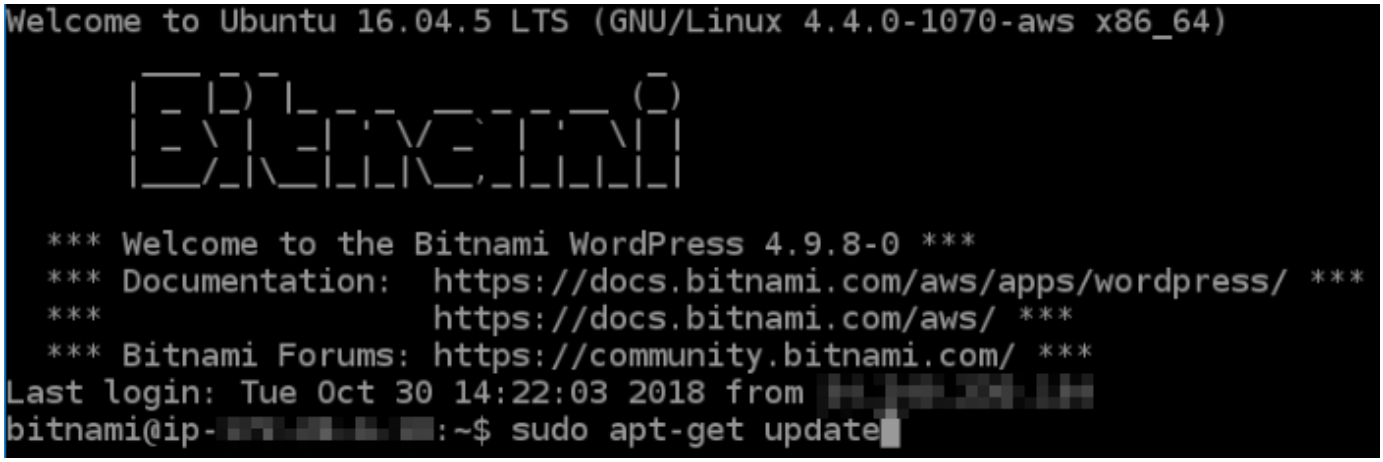
Lightsail インスタンスに Certbot をインストールするには

1. Lightsail コンソール にサインインします。
2. 左側のナビゲーションペインで、接続先のインスタンスの SSH クイック接続アイコンを選択します。



3. Lightsail のブラウザベースの SSH セッションに接続したら、次のコマンドを入力してインスタンスのパッケージを更新します。

```
sudo apt-get update
```



4. 次のコマンドを入力してソフトウェアプロパティパッケージをインストールします。Certbot の開発者は、Personal Package Archive (PPA) を使用して Cerbot を配信します。ソフトウェアプロパティパッケージを使用すると、PPA をより効率的に操作できます。

```
sudo apt-get install software-properties-common
```

> **Note**
>
> `sudo apt-get install` コマンドを実行したときに `Could not get lock` エラーが発生した場合は、約 15 分待ってから再試行してください。このエラーは、自動アップグレードをインストールするために Apt パッケージ管理ツールを使用している cron ジョブが原因で発生している可能性があります。

5. 次のコマンドを入力して GPG パッケージをインストールし、Certbot をローカルの apt リポジトリに追加します。

> **Note**
>
> ステップ 5 は、Ubuntu Linux ディストリビューションを使用するインスタンスにのみ適用されます。インスタンスが Debian Linux ディストリビューションを使用している場合は、このステップをスキップしてください。

```
sudo apt-get install gpg -y
```

```
sudo apt-add-repository ppa:certbot/certbot -y
```

6. 次のコマンドを入力して apt を更新し、新しいリポジトリを含めます。

```
sudo apt-get update -y
```

7. 次のコマンドを入力して Cerbot をインストールします。

```
sudo apt-get install certbot -y
```

これで Lightsail インスタンスに Cerbot がインストールされました。

8. ブラウザベースの SSH ターミナルウィンドウは開いたままにします。このチュートリアルで後ほど戻ります。このチュートリアルの「次のセクション」に進みます。

ステップ 3: Let's Encrypt の SSL ワイルドカード証明書をリクエストする

Let's Encrypt の証明書をリクエストするプロセスを開始します。Certbot を使用してワイルドカード証明書をリクエストします。この 1 つの証明書をドメインとそのサブドメインの両方に使用できます。たとえば、1 つのワイルドカード証明書を `example.com` 最上位ドメイン、`blog.example.com` サブドメイン、および `stuff.example.com` サブドメインに使用できます。

Let's Encrypt の SSL ワイルドカード証明書をリクエストするには

1. このチュートリアルのステップ 2 で使用した同じブラウザベースの SSH ターミナルウィンドウで、以下のコマンドを入力してドメインの環境変数を設定します。より効率的にコマンドをコピーして貼り付け、証明書を取得できます。*domain* は登録済みのドメイン名に置き換えてください。

   ```
   DOMAIN=domain
   ```

   ```
   WILDCARD=*.$DOMAIN
   ```

   例:

   ```
   DOMAIN=example.com
   ```

   ```
   WILDCARD=*.$DOMAIN
   ```

2. 次のコマンドを入力し、変数が正しい値を返すことを確認します。

   ```
   echo $DOMAIN && echo $WILDCARD
   ```

   次のような結果が表示されます。

3. 次のコマンドを入力して Certbot をインタラクティブモードで起動します。このコマンドでは、DNS チャレンジで手動認可を使用してドメインの所有権を検証することを Certbot に指示

します。また、最上位ドメインとそのサブドメイン用にワイルドカード証明書をリクエストします。

```
sudo certbot -d $DOMAIN -d $WILDCARD --manual --preferred-challenges dns certonly
```

4. プロンプトに応じて E メールアドレスを入力します。これで更新とセキュリティに関する通知を受信します。
5. Let's Encrypt のサービス利用規約を読みます。読み終わり、同意する場合は A キーを押します。同意しない場合は、Let's Encrypt の証明書を取得できません。
6. E メールアドレスの共有と IP アドレスのログ記録に関するプロンプトに適宜応答します。
7. Let's Encrypt から、指定されたドメインの所有者であることの検証を求められます。これを行うには、ドメインの DNS レコードに TXT レコードを追加します。以下の例に示すように 2 組の TXT レコード値が提供されます。

> **Note**
>
> Let's Encrypt では検証に必要な TXT レコードを 1 つまたは複数提供する場合があります。この例では、検証に使用する 2 つの TXT レコードが提供されました。

8. Lightsail のブラウザベースの SSH セッションは開いたままにします。このチュートリアルで後ほど戻ります。このチュートリアルの「次のセクション」に進みます。

ステップ 4: ドメインの DNS ゾーンに TXT レコードを追加する

ドメインの DNS ゾーンに TXT レコードを追加すると、自分がドメインを所有していることが検証されます。ここでは、デモの目的で Lightsail の DNS ゾーンを使用します。ただし、ドメインレジストラがホストする他の一般的な DNS ゾーンでも手順はほぼ同じです。

> **Note**
>
> ドメインの Lightsail DNS ゾーンの作成方法の詳細については、「Lightsail で DNS ゾーンを作成し、ドメインの DNS レコードを管理する」を参照してください。

Lightsail でドメインの DNS ゾーンに TXT レコードを追加するには

1. 左側のナビゲーションペインで [ドメインと DNS] を選択します。
2. ページの [DNS ゾーン] セクションで、Certbot 証明書リクエストで指定したドメインの DNS ゾーンを選択します。
3. DNS ゾーンエディタで [DNS records] (DNS レコード) を選択します。
4. [レコードの追加] を選択します。
5. [Record type] (レコードタイプ) のドロップダウンメニューで [TXT record] (TXT レコード) を選択します。
6. Let's Encrypt 証明書のリクエストで指定された値を [Record name] (レコード名) と [Responds with] (応答) フィールドに入力します。

   > **Note**
   >
   > Lightsail コンソールには、ドメインの頂点部分があらかじめ入力されています。たとえば、*_acme-challenge.example.com* サブドメインを追加する場合は、*_acme-challenge* をテキストボックスに入力するだけで、レコードを保存するときに Lightsail が `.example.com` の部分を追加します。

7. [保存] を選択します。
8. ステップ 4 ~ 7 を繰り返して、Let's Encrypt の証明書リクエストで指定された 2 番目の TXT レコードのセットを追加します。
9. Lightsail コンソールのブラウザウィンドウは、このチュートリアルで後ほど戻るので開いたままにします。このチュートリアルの「次のセクション」に進みます。

ステップ 5: TXT レコードが反映されたことを確認する

MxToolbox ユーティリティを使用して TXT レコードがインターネットの DNS に反映されたことを確認します。DNS レコードの反映には、DNS ホスティングプロバイダーと DNS レコードの有効期限 (TTL) の設定によって時間がかかる場合があります。このステップを完了し、TXT レコードが反映されたことを確認した上で、Certbot 証明書のリクエストに進むことが重要です。そうしないと、証明書のリクエストは失敗します。

TXT レコードがインターネットの DNS に反映されたことを確認するには

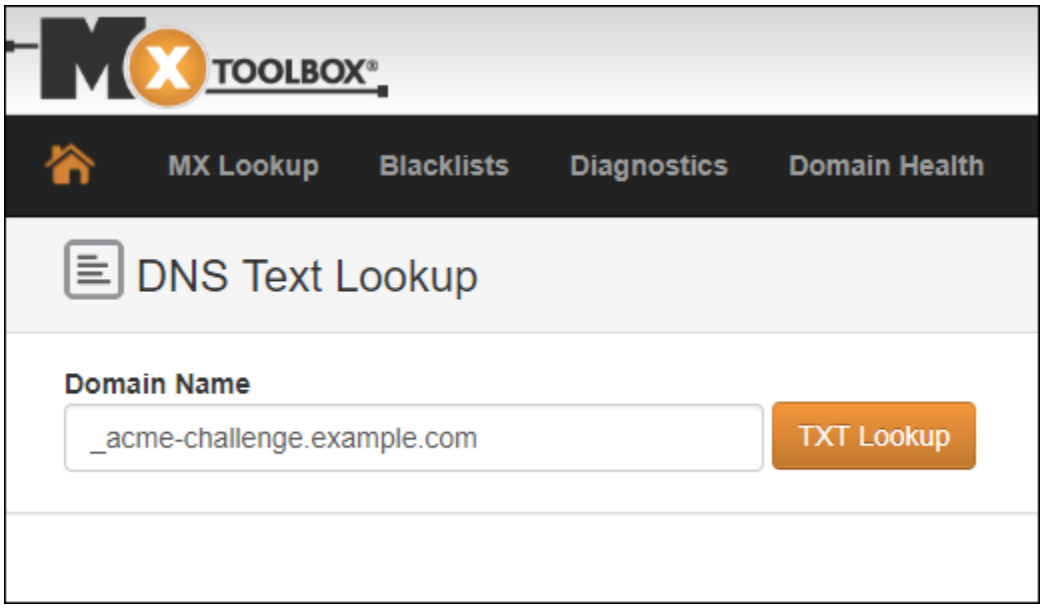
1. 新しいブラウザウィンドウを開き、https://mxtoolbox.com/TXTLookup.aspx に移動します。

2. 次の内容をテキストボックスに入力します。*domain* は実際のドメインに置き換えてください。

```
_acme-challenge.domain
```

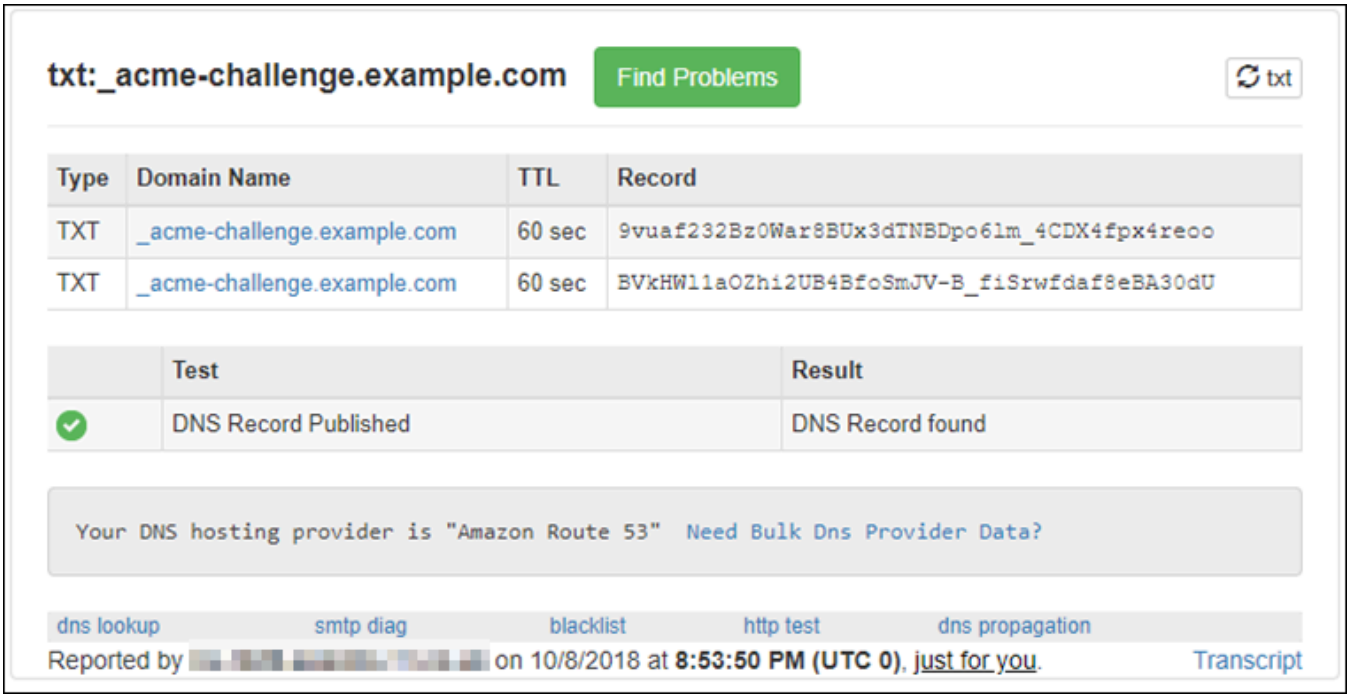
例:

```
_acme-challenge.example.com
```



3. [TXT Lookup (TXT ルックアップ)] を選択して確認を行います。
4. 以下のいずれかのレスポンスが返されます。

   - TXT レコードがインターネットの DNS に反映された場合は、次のスクリーンショットに示すようなレスポンスが表示されます。ブラウザウィンドウを閉じて、このチュートリアルの「次のセクション」に進みます。



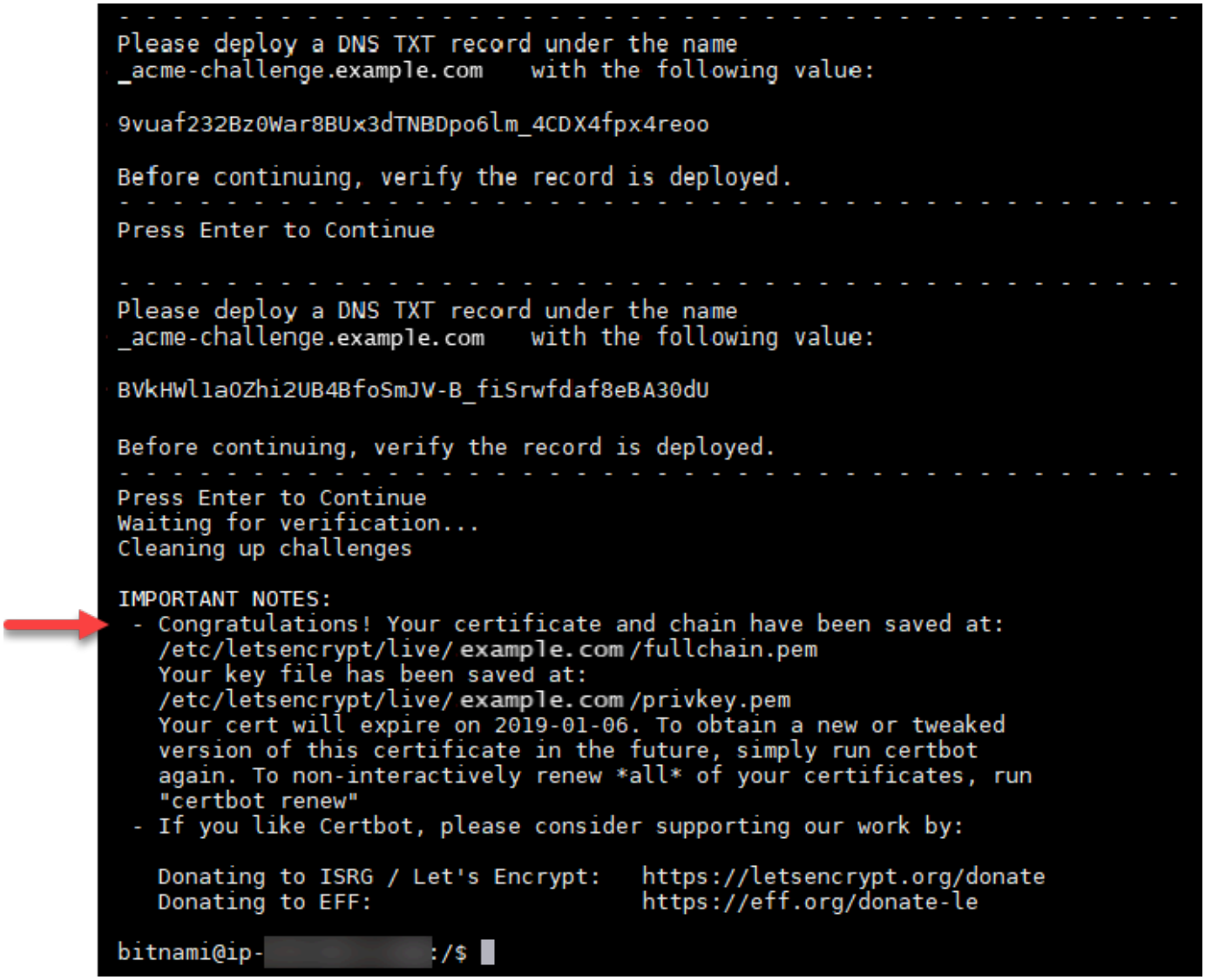
- TXT レコードがインターネットの DNS に反映されていない場合は、[DNS Record not found (DNS レコードが見つかりません)] というレスポンスが返されます。適切な DNS レコードをドメインの DNS ゾーンに追加したことを確認してください。適切なレコードを追加した場合は、ドメインの DNS レコードが反映されるまでしばらく待ってから、TXT のルックアップを再実行します。

ステップ 6: Let's Encrypt の SSL 証明書リクエストを完了する

WordPress インスタンスの Lightsail ブラウザベースの SSH セッションに戻り、Let's Encrypt 証明書のリクエストを完了します。Certbot は、SSL 証明書、チェーン、およびキーファイルを WordPress インスタンスの特定のディレクトリに保存します。

Let's Encrypt の SSL 証明書リクエストを完了するには

1. WordPress インスタンスの Lightsail ブラウザベースの SSH セッションで、Enter キーを押し、Let's Encrypt SSL 証明書のリクエストを続行します。成功すると、次のスクリーンショットに示すようなレスポンスが表示されます。



証明書、チェーン、およびキーファイルが /etc/letsencrypt/live/*domain*/ ディレクトリに保存されたことを確認するメッセージが表示されます。*domain* は、実際のドメイン (/etc/letsencrypt/live/*example.com*/ など) に置き換えてください。

2. メッセージに記載されている有効期限を書き留めておきます。この期限日までに証明書を更新する必要があります。

```
IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at:
   /etc/letsencrypt/live/example.com/fullchain.pem
   Your key file has been saved at:
   /etc/letsencrypt/live/example.com/privkey.pem
   Your cert will expire on 2019-01-06. To obtain a new or tweaked
   version of this certificate in the future, simply run certbot
   again. To non-interactively renew *all* of your certificates, run
   "certbot renew"
 - If you like Certbot, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
   Donating to EFF:                    https://eff.org/donate-le
```

3. これで Let's Encrypt SSL 証明書が手に入ったので、このチュートリアルの「次のセクション」に進みます。

ステップ 7: Apache サーバーディレクトリで Let's Encrypt の証明書ファイルへのリンクを作成する

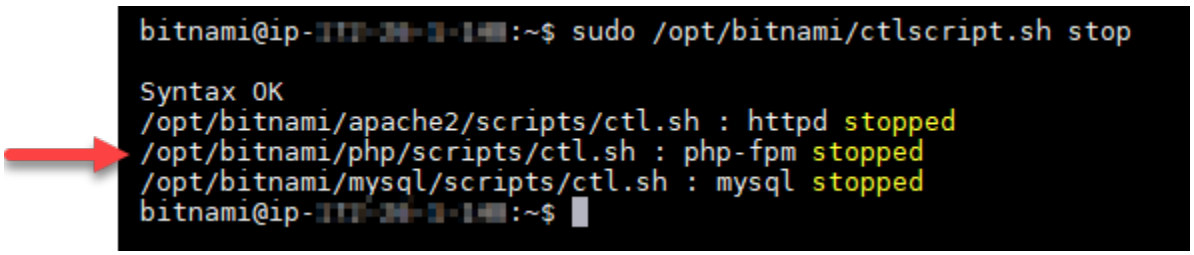
WordPress インスタンスの Apache サーバーディレクトリにある Let's Encrypt の SSL 証明書ファイルへのリンクを作成します。また、必要になる場合に備えて既存の証明書をバックアップします。

Apache サーバーディレクトリで Let's Encrypt の証明書ファイルへのリンクを作成するには

1. WordPress インスタンスの Lightsail ブラウザベースの SSH セッションで、次のコマンドを入力して基盤となるサービスを停止します。

   ```
   sudo /opt/bitnami/ctlscript.sh stop
   ```

   次のようなレスポンスが表示されます。

2. 次のコマンドを入力してドメインの環境変数を設定します。コマンドのコピー＆ペーストで、より効率的に証明書ファイルにリンクを張れます。*domain* を登録済みのドメイン名に置き換えます。

```
DOMAIN=domain
```

例:

```
DOMAIN=example.com
```

3. 次のコマンドを入力し、変数が正しい値を返すことを確認します。

```
echo $DOMAIN
```

次のような結果が表示されます。

4. バックアップとして既存の証明書ファイルがある場合、以下のコマンドを個別に入力して名前を書き換えます。さまざまなディストリビューションとファイル構造の詳細については、このチュートリアルの冒頭の重要ブロックを参照してください。

- Debian Linux ディストリビューションの場合

  アプローチ A (システムパッケージを使用した Bitnami インストール):

  ```
  sudo mv /opt/bitnami/apache2/conf/bitnami/certs/server.crt /opt/bitnami/apache2/
  conf/bitnami/certs/server.crt.old
  ```

  ```
  sudo mv /opt/bitnami/apache2/conf/bitnami/certs/server.key /opt/bitnami/apache2/
  conf/bitnami/certs/server.key.old
  ```

  アプローチ B (自己完結型 Bitnami インストール):

  ```
  sudo mv /opt/bitnami/apache2/conf/server.crt /opt/bitnami/apache2/conf/
  server.crt.old
  ```

  ```
  sudo mv /opt/bitnami/apache2/conf/server.key /opt/bitnami/apache2/conf/
  server.key.old
  ```

- Ubuntu Linux ディストリビューションを使用する古いインスタンスの場合 :

```
sudo mv /opt/bitnami/apache/conf/bitnami/certs/server.crt /opt/bitnami/apache/
conf/bitnami/certs/server.crt.old
```

```
sudo mv /opt/bitnami/apache/conf/bitnami/certs/server.key /opt/bitnami/apache/
conf/bitnami/certs/server.key.old
```

```
sudo mv /opt/bitnami/apache/conf/bitnami/certs/server.csr /opt/bitnami/apache/
conf/bitnami/certs/server.csr.old
```

5. 以下のコマンドを個別に入力し、Apache ディレクトリで Let's Encrypt の証明書ファイルへのリンクを作成します。さまざまなディストリビューションとファイル構造の詳細については、このチュートリアルの冒頭の重要ブロックを参照してください。

   - Debian Linux ディストリビューションの場合

     アプローチ A (システムパッケージを使用した Bitnami インストール):

     ```
     sudo ln -sf /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/apache2/conf/
     bitnami/certs/server.key
     ```

     ```
     sudo ln -sf /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/apache2/
     conf/bitnami/certs/server.crt
     ```

     アプローチ B (自己完結型 Bitnami インストール):

     ```
     sudo ln -sf /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/apache2/conf/
     server.key
     ```

     ```
     sudo ln -sf /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/apache2/
     conf/server.crt
     ```

   - Ubuntu Linux ディストリビューションを使用した古いインスタンスの場合:

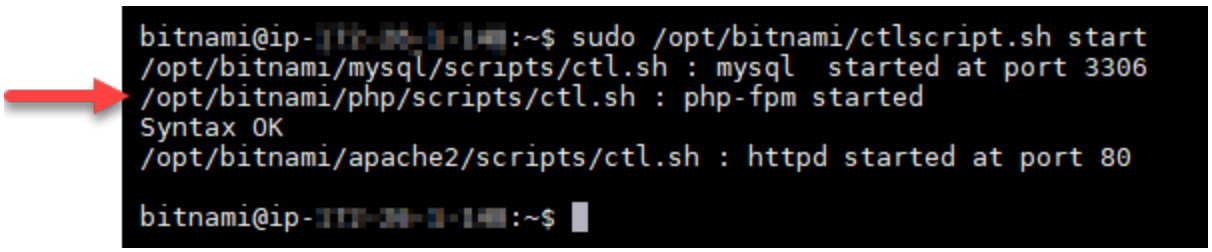
     ```
     sudo ln -s /etc/letsencrypt/live/$DOMAIN/privkey.pem /opt/bitnami/apache/conf/
     bitnami/certs/server.key
     ```

```
sudo ln -s /etc/letsencrypt/live/$DOMAIN/fullchain.pem /opt/bitnami/apache/conf/
bitnami/certs/server.crt
```

6. 次のコマンドを入力して、先ほど停止した基本サービスを開始します。

```
sudo /opt/bitnami/ctlscript.sh start
```

次のような結果が表示されます。



WordPress インスタンスの SSL 証明書ファイルが適切なディレクトリに配置されました。

7. このチュートリアルの「次のセクション」に進みます。

ステップ 8: Really Simple SSL プラグインを使用して SSL 証明書を WordPress サイトに統合する

Really Simple SSL プラグインを WordPress サイトにインストールし、これを使用して SSL 証明書を統合します。Really Simple SSL では、サイトを訪問するユーザーが常に HTTPS 接続を利用できるように、HTTP から HTTPS へのリダイレクトも設定します。

Really Simple SSL プラグインを使用して SSL 証明書を WordPress サイトに統合するには

1. WordPress インスタンスの Lightsail ブラウザ ベースの SSH セッションで、次のコマンドを入力して `wp-config.php` および `htaccess.conf` ファイルを書き込み可能に設定します。Really Simple SSL プラグインは、wp-config.php ファイルに書き込むことで証明書を設定します。

   - Debian Linux ディストリビューションを使用する新しいインスタンスの場合：

```
sudo chmod 666 /opt/bitnami/wordpress/wp-config.php && sudo chmod 666 /opt/
bitnami/apache/conf/vhosts/htaccess/wordpress-htaccess.conf
```

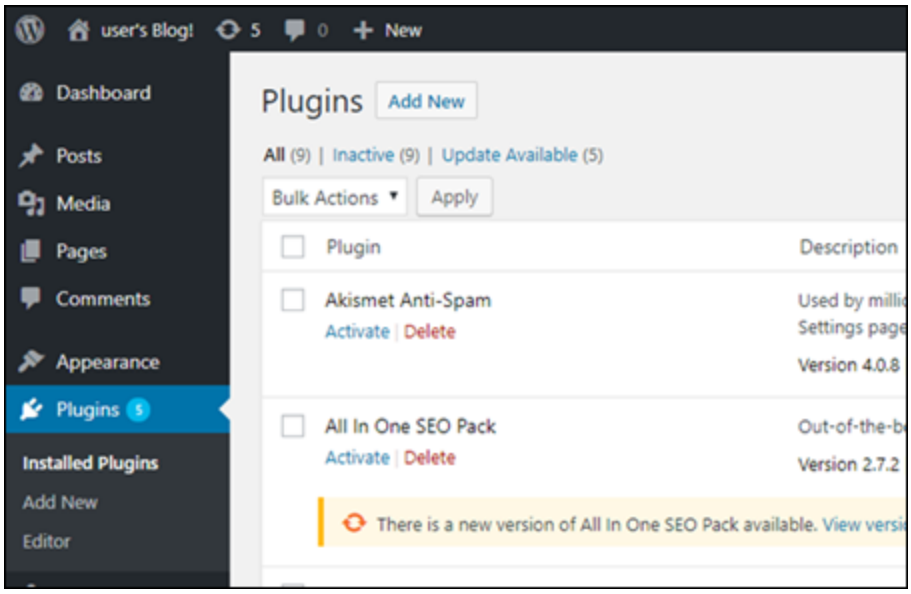
   - Ubuntu Linux ディストリビューションを使用する古いインスタンスの場合：

```
sudo chmod 666 /opt/bitnami/apps/wordpress/htdocs/wp-config.php && sudo chmod
 666 /opt/bitnami/apps/wordpress/conf/htaccess.conf
```

2. 新しいブラウザウィンドウを開き、WordPress インスタンスの管理ダッシュボードにサインインします。

> **Note**
>
> 詳細については、「Amazon Lightsailの Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

3. 左のナビゲーションペインから、[Plugins] (プラグイン) を選択します。
4. プラグインページの上部で、[Add New] (新規追加) を選択します。



5. [Really Simple SSL] を探します。
6. 検索結果の Really Simple SSL プラグインの横にある [Install Now (今すぐインストール)] を選択します。

7. インストールが完了したら、[Activate] (有効化) を選択します。
8. 表示されるプロンプトで [Go ahead, activate SSL!] (SSL の有効化を開始!) を選択します。WordPress インスタンスの管理ダッシュボードのサインインページにリダイレクトされる場合があります。

   これで SSL 暗号化を使用するように WordPress インスタンスが設定されました。さらに、HTTP から HTTPS へ自動的に接続をリダイレクトするように WordPress インスタンスが設定されました。訪問者が `http://example.com` にアクセスすると、暗号化された HTTPS 接続 (`https://example.com`) に自動的にリダイレクトされます。

ステップ 9: Let's Encrypt 証明書を 90 日ごとに更新する

Let's Encrypt 証明書の有効期間は 90 日間です。証明書は有効期限が切れる 30 日前から更新できます。Let's Encrypt 証明書を更新するには、取得するために使用した元のコマンドを実行します。このチュートリアルの「Let's Encrypt の SSL ワイルドカード証明書をリクエストする」セクションのステップを繰り返します。

特定のインスタンスタイプの段階的な手順に従います。各トピックでは、インスタンスの Linux ディストリビューション (Ubuntu または Debian) と Bitnami インストールタイプ (システムパッケージまたは自己完結型) に合わせた詳細なコマンドと設定手順について説明します。このトピックに従うことで、Let's Encrypt の無料の SSL/TLS 証明書を使用して Lightsail ウェブサイトとアプリケーションを保護することができ、暗号化された通信を確保し、訪問者のセキュリティを向上させることができます。

Lightsail インスタンスの IPv6 ネットワークを設定する

このセクションでは、Lightsail インスタンスブループリントでの IPv6 の設定に関連する以下のトピックについて説明します:

トピック

- Lightsail で cPanel インスタンスに IPv6 接続を設定する
- Lightsail で GitLab インスタンスに IPv6 接続を設定する
- Lightsail で Nginx インスタンスに IPv6 接続を設定する
- Lightsail で Plesk インスタンスに IPv6 接続を設定する

Lightsail で cPanel インスタンスに IPv6 接続を設定する

Amazon Lightsail 内のすべてのインスタンスにはデフォルトでパブリックアドレスとプライベート IPv4 アドレスが割り当てられています。オプションで、インスタンスの IPv6 を有効にして、パブリック IPv6 アドレスを割り当てることができます。詳細については「Amazon Lightsail の IP アドレス」および「IPv6 の有効化または無効化」を参照してください。

cPanel と WHM ブループリントを使用するインスタンスで IPv6 を有効化した後、インスタンスに IPv6 アドレスを認識させるために追加のステップを実行する必要があります。このガイドでは、cPanel と WHM インスタンスで実行する必要がある追加の手順を説明します。

前提条件

以下の前提条件を完了します (まだの場合)。

- Lightsail で cPanel と WHM インスタンスを作成する。詳細については、「インスタンスを作成する」を参照してください。
- cPanel と WHM インスタンスを設定します。詳細については、「クイックスタートガイド: Amazon Lightsail 上の cPanel と WHM」を参照してください。

> **Important**
>
> このガイドの手順を続行する前に、すべてのソフトウェアの更新と必要なシステムの再起動が実行されていることを確認してください。

- cPanel と WHMインスタンスの IPv6 を有効化します。詳細については、「IPv6 を有効化または無効化する」を参照してください。

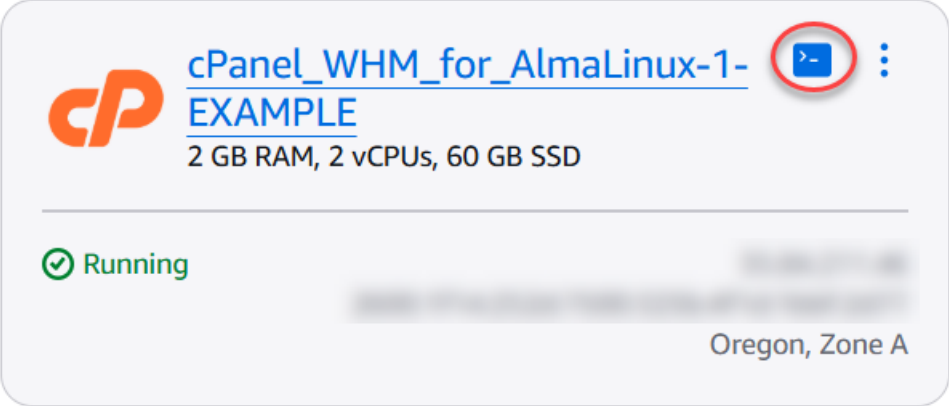
> **Note**
>
> 2021 年 1 月 12 日以降に作成された新しい cPanel と WHM インスタンスは、Lightsail コンソールで作成されると、IPv6 がデフォルトで有効になっています。インスタンスの作成

時に IPv6 がデフォルトで有効になっていても、インスタンスで IPv6 を設定するには、このガイドの以下の手順を実行する必要があります。

cPanel と WHM インスタンスで Pv6 を設定する。

Lightsail で cPanel と WHM インスタンス上で IPv6 を設定するには、以下の手順を実行します。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの [インスタンス] セクションから設定したい cPanel と WHM インスタンスを探し、ブラウザベース SSH クライアントアイコンを選択して SSH を使用して接続します。



3. インスタンスに接続後、次のコマンドを入力して`ifcfg-eth0` ネットワークインターフェース設定ファイルを Nano を使用して開きます。

```
sudo nano /etc/sysconfig/network-scripts/ifcfg-eth0
```

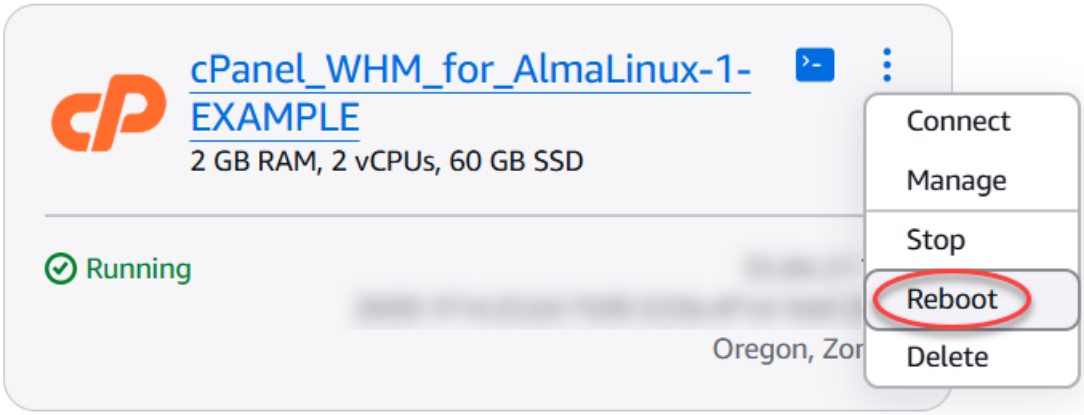
4. ファイルにテキストが追加されていない場合、次のテキストを追加します。

```
IPV6INIT=yes
IPV6_AUTOCONF=yes
DHCPV6C=yes
```

結果は次の例のようになります。

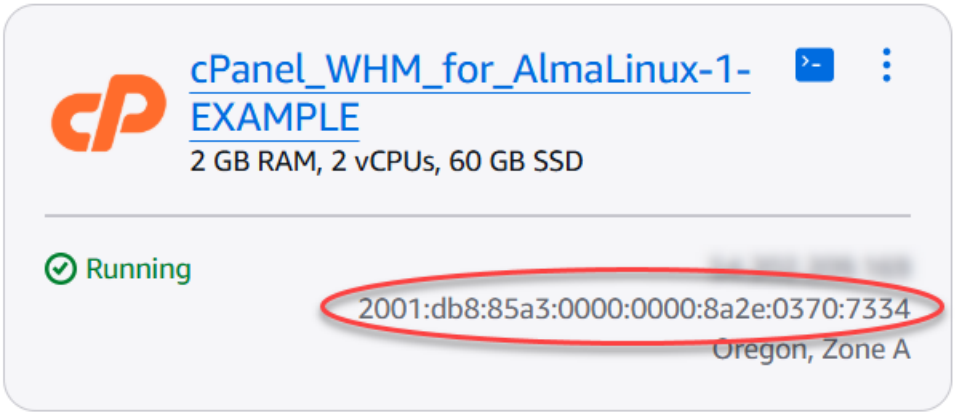
```
# Automatically generated by the vm import process
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
NAME=eth0
DEVICE=eth0
ONBOOT=yes
IPV6INIT=yes
IPV6_FAILURE_FATAL=no
DHCPV6C=yes
IPV6_AUTOCONF=yes
```

5. CTRL+C を押してファイルを終了します。
6. Y を修正したバッファを保存するプロンプトが表示されたら押します。Enter を押してして既存のファイルに保存します。これにより、`ifcfg-eth0`ネットワークインターフェース設定ファイルに編集が保存されます。
7. ブラウザベースの SSH ウィンドウを閉じ、Lightsail コンソールに戻ります。
8. Lightsail ホームページの [インスタンス] セクションで、cPanel および WHM インスタンスのアクションメニュー (:) を選択し、[再起動] を選択します。

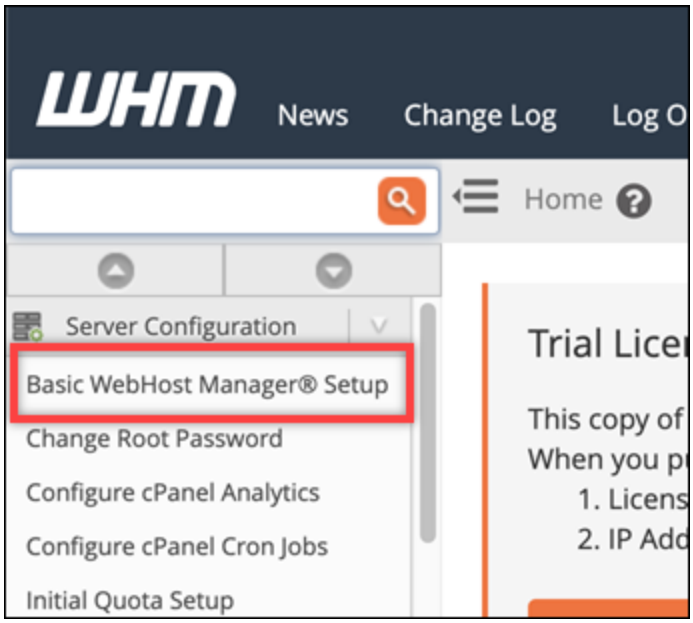


インスタンスが再起動するまで数分待ってから、次のステップに進みます。

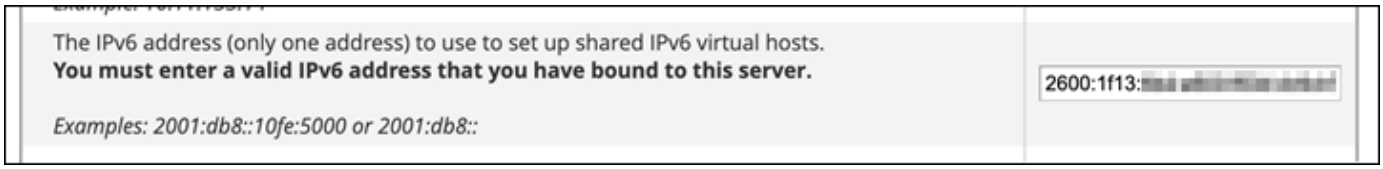
9. Lightsail ホームページの [インスタンス] セクションで、cPanel および WHM インスタンスに割り当てられた IPv6 アドレスを書き留めます。



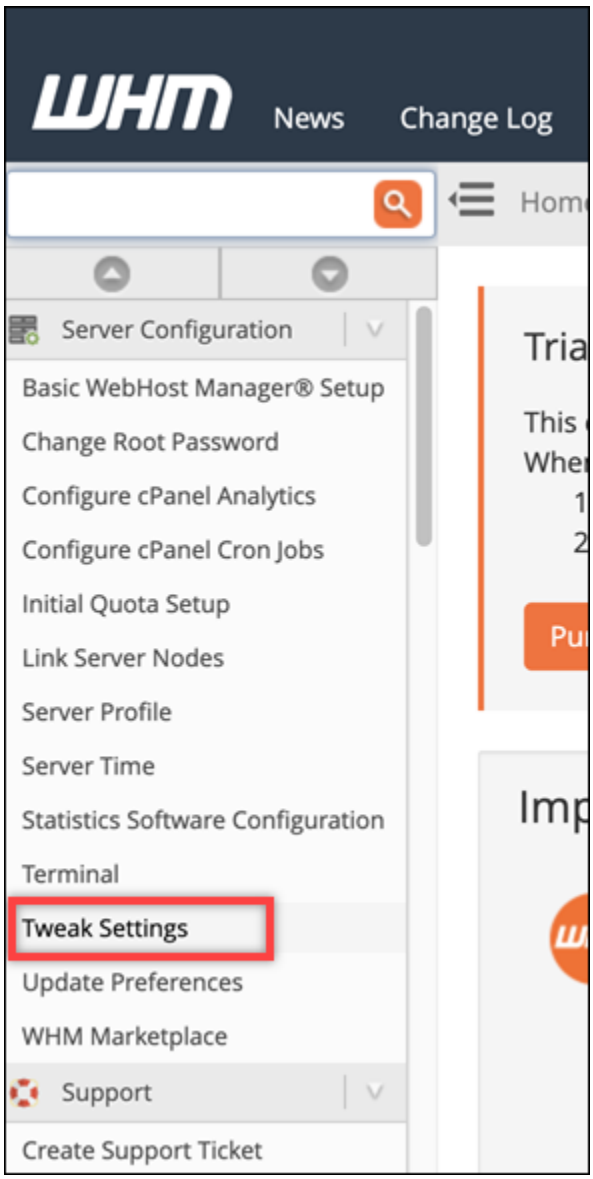
10. 新しいブラウザタブを開き、cPanel と WHM インスタンスのウェブホストマネージャ（WHM）にサインインします。
11. WHM コンソールの左のナビゲーションペインで、[WebHost Manager の基本設定] を選択します。



12. [All] (すべて)タブで、使用する IPv6 アドレスのテキストを探して、インスタンスに割り当てられている IPv6 アドレスを入力します。手順のステップ 9 からインスタンスに割り当てられた IPv6 アドレスを書き留めておく必要があります。

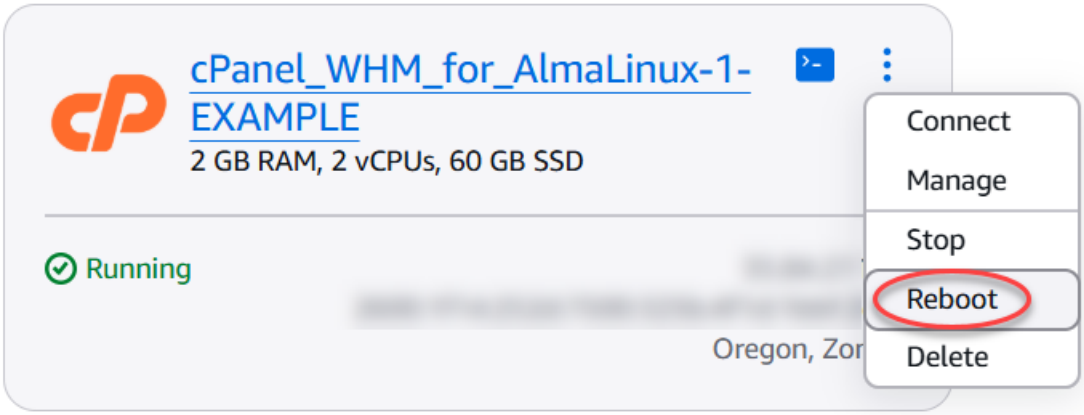


13. ページの最下部までスクロールし、[変更の保存] を選択します。
14. WHM コンソールの左のナビゲーションペインで [設定を微調整する] を選択します。



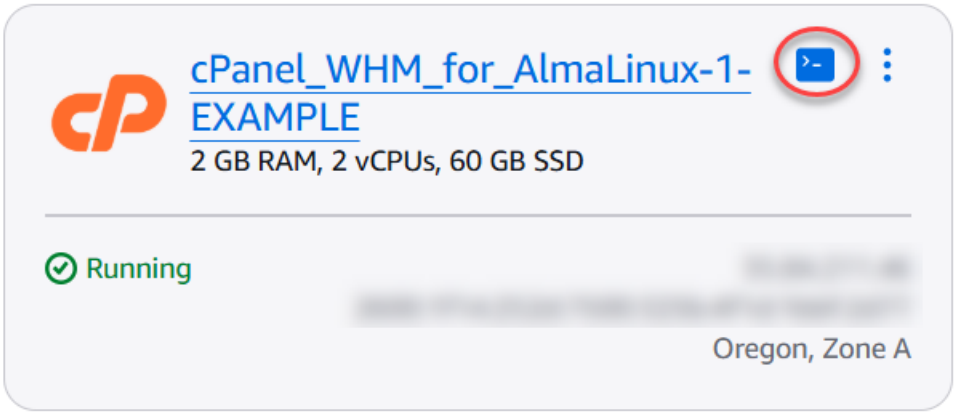
15. [All] (すべて)タブで、下にスクロールしてIPv6 アドレスをリッスンする設定を探して、Onに設定します。

16. ページの下部にスクロールし、保存を選択します。
17. Lightsail コンソールに戻ります。
18. Lightsail ホームページの [インスタンス] セクションで、cPanel および WHM インスタンスのアクションメニュー (:) を選択し、[再起動] を選択します。



インスタンスが再起動するまで数分待ってから、次のステップに進みます。

19. SSH を使用して接続する cPanel と WHM インスタンスのブラウザベースの SSH クライアントアイコンを選択します。



20. インスタンスに接続後、以下のコマンドを入力して、インスタンスに設定された IP アドレスを表示し、割り当てられた IPv6 アドレスを認識していることを確認します。

```
ip addr
```

次のようなレスポンスが表示されます。インスタンスが IPv6 アドレスを認識している場合はこの例のように、レスポンスに [scope global] のラベルでリスト表示されます。

```
[centos@ ~]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP group default qlen 1000
    link/ether 02:9b:51:92:50:45 brd ff:ff:ff:ff:ff:ff
    inet /20 brd  scope global dynamic eth0
    inet6 2600:1f13:  :6d59:4ac0/128 scope global dynamic
    inet6 fe80:  :5045/64 scope link
       valid_lft forever preferred_lft forever
```

21. 以下のコマンドを入力して、インスタンスが IPv6 アドレスに ping されているかを確認します。

```
ping6 ipv6.google.com -c 6
```

結果は、以下の例のように表示され、インスタンスが IPv6 アドレスに ping されていることが確認できます。

```
[centos@ ~]$ ping6 ipv6.google.com
PING ipv6.google.com(sea15s12-in-x0e.1e100.net (2607:f8b0:400a:809::200e)) 56 data bytes
64 bytes from sea15s12-in-x0e.1e100.net (2607:f8b0:400a:809::200e): icmp_seq=1 ttl=103 tim
e=7.66 ms
64 bytes from sea15s12-in-x0e.1e100.net (2607:f8b0:400a:809::200e): icmp_seq=2 ttl=103 tim
e=7.70 ms
64 bytes from sea15s12-in-x0e.1e100.net (2607:f8b0:400a:809::200e): icmp_seq=3 ttl=103 tim
e=7.68 ms
64 bytes from sea15s12-in-x0e.1e100.net (2607:f8b0:400a:809::200e): icmp_seq=4 ttl=103 tim
e=7.69 ms
64 bytes from sea15s12-in-x0e.1e100.net (2607:f8b0:400a:809::200e): icmp_seq=5 ttl=103 tim
e=7.70 ms
64 bytes from sea15s12-in-x0e.1e100.net (2607:f8b0:400a:809::200e): icmp_seq=6 ttl=103 tim
e=7.68 ms
^C
--- ipv6.google.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 7.667/7.690/7.702/0.052 ms
```

Lightsail で GitLab インスタンスに IPv6 接続を設定する

Amazon Lightsail 内のすべてのインスタンスにはデフォルトでパブリックアドレスとプライベート IPv4 アドレスが割り当てられています。オプションで、インスタンスの IPv6 を有効にして、パブリック IPv6 アドレスを割り当てることができます。詳細については「Amazon Lightsail の IP アドレス」および「IPv6 の有効化または無効化」を参照してください。

GitLab ブループリントを使用するインスタンスで IPv6 を有効化した後、インスタンスに IPv6 アドレスを認識させるために追加のステップを実行する必要があります。このガイドでは、GitLab インスタンスで実行しなければいけない追加のステップを説明します。

前提条件

以下の前提条件を完了します (まだの場合)。

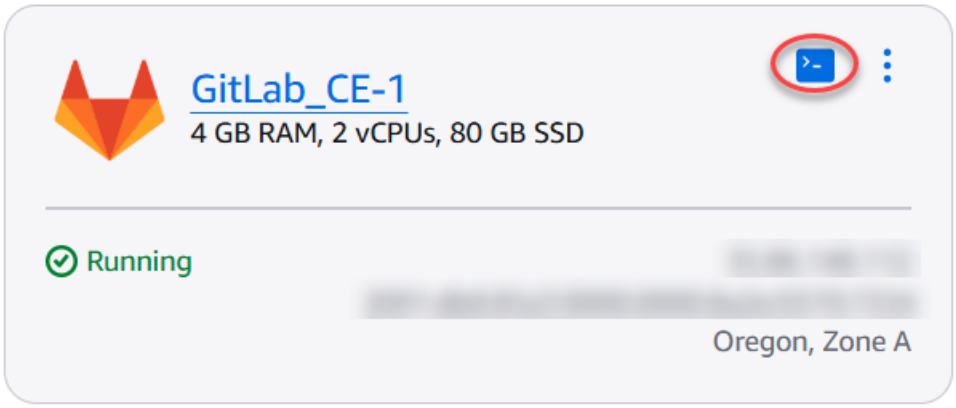
- Lightsail で GitLab インスタンスを作成します。詳細については、「インスタンスを作成する」を参照してください。
- GitLab インスタンスで IPv6 を有効化します。詳細については、「IPv6 を有効化または無効化する」を参照してください。

> **Note**
>
> 2021 年 1 月 12 日以降に作成された新しい GitLab インスタンスは、Lightsail コンソールで作成されている場合、 IPv6 がデフォルトで有効になっています。インスタンスの作成時に IPv6 がデフォルトで有効になっていても、インスタンスで IPv6 を設定するには、このガイドの以下の手順を実行する必要があります。

GitLab インスタンスで IPv6 を設定する

Lightsail の GitLab インスタンスで IPv6 を設定するには、以下のステップを実行します。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの [インスタンス] セクションで設定したい GitLab インスタンスを探し、ブラウザベース SSH クライアントアイコンを選択し、SSH を使用して接続します。

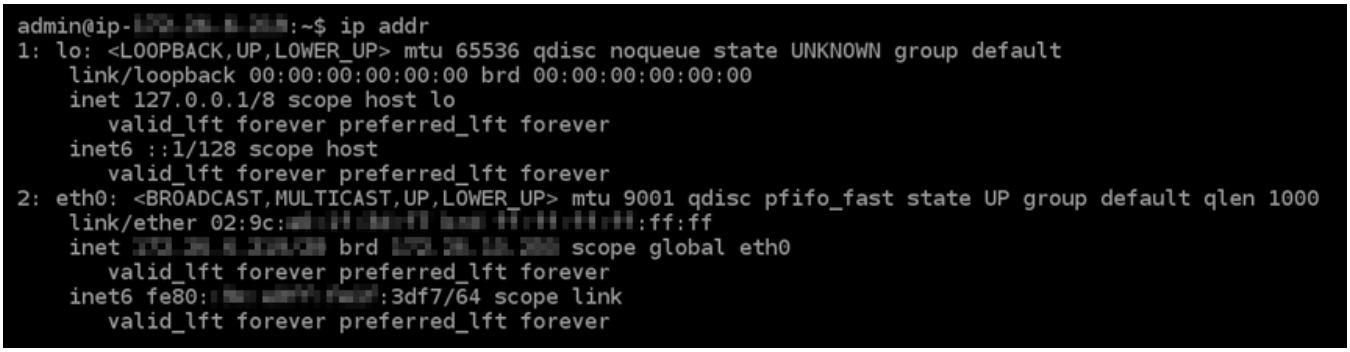


3. インスタンスに接続したら、以下のコマンドを入力し、インスタンスに設定されている IP アドレスを表示します。

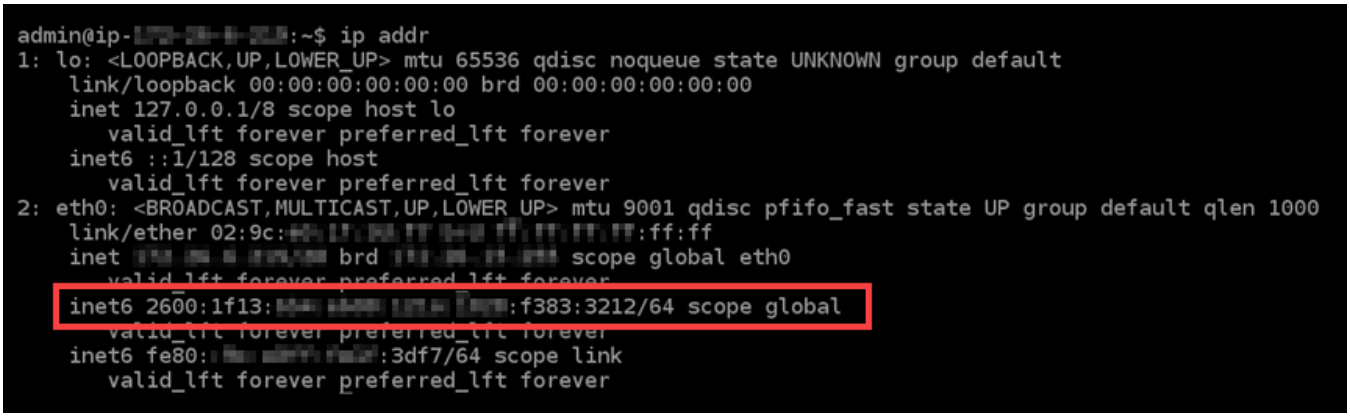
```
ip addr
```

以下のようなレスポンスが表示されます。

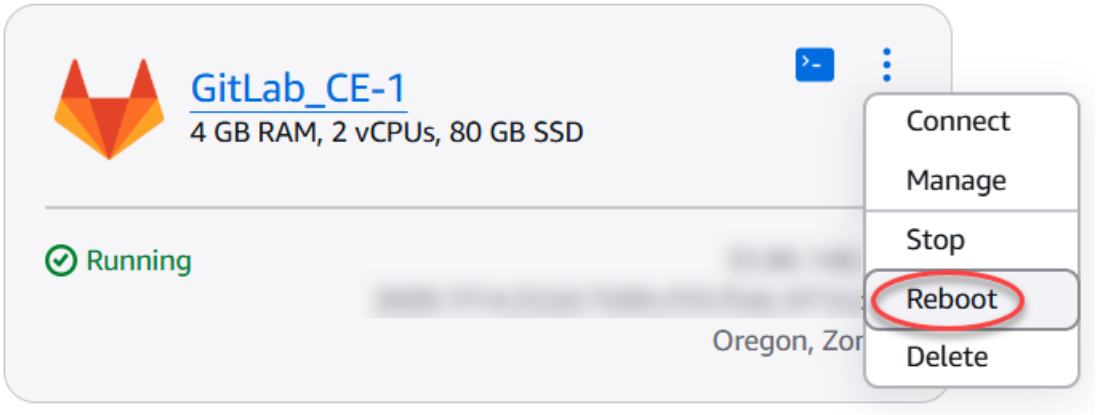
- インスタンスが IPv6 アドレスを認識しない場合、レスポンスに表示されません。この手順のステップ 4～9 を続行する必要があります。



- インスタンスが IPv6 アドレスを認識している場合は、レスポンスに `scope global` の例のように表示されます。ここで停止します。インスタンスが IPv6 アドレスを認識するように設定済みなので、この手順のステップ 4～9 を実行する必要はありません。



4. Lightsail コンソールに戻ります。
5. Lightsail ホームページの [インスタンス] セクションで、GitLab インスタンスのアクションメニュー (⋮) を選択し、[再起動] を選択します。

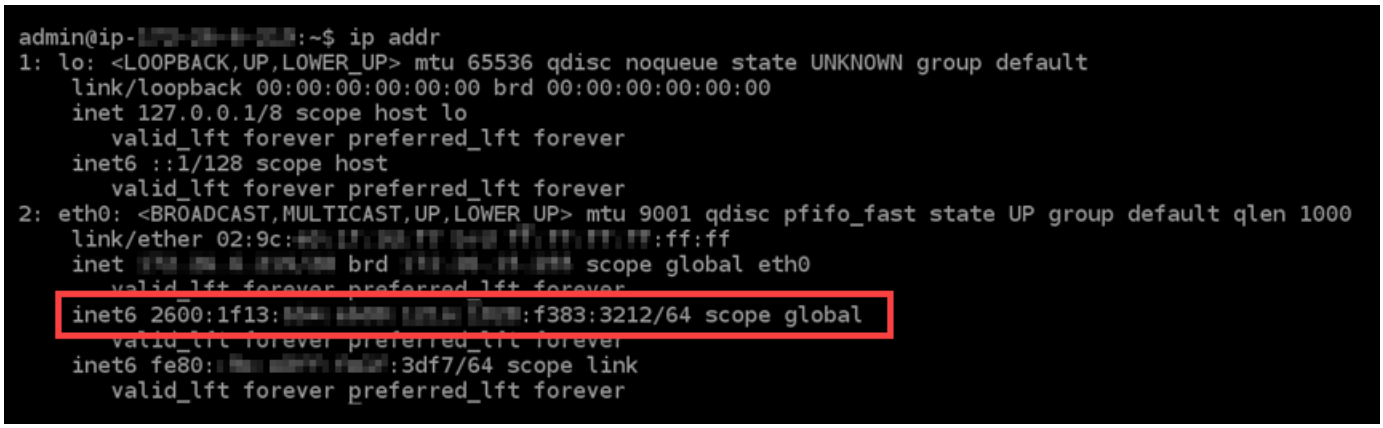


インスタンスが再起動するまで数分待ってから、次のステップに進みます。

6. GitLab インスタンスの SSH セッションに戻ります。
7. 以下のコマンドを入力して、インスタンスに設定された IP アドレスを表示し、割り当てられた IPv6 アドレスを認識していることを確認します。

```
ip addr
```

以下のようなレスポンスが表示されます。インスタンスが IPv6 アドレスを認識している場合は、レスポンスにラベルscope global の例ように表示されます。



Lightsail で Nginx インスタンスに IPv6 接続を設定する

Amazon Lightsail 内のすべてのインスタンスにはデフォルトでパブリックアドレスとプライベート IPv4 アドレスが割り当てられています。オプションで、インスタンスの IPv6 を有効にして、パブリック IPv6 アドレスを割り当てることができます。詳細については「Amazon Lightsail の IP アドレス」および「IPv6 の有効化または無効化」を参照してください。

Nginx ブループリントを使用するインスタンスで IPv6 を有効化した後、インスタンスに IPv6 アドレスを認識させる追加のステップを実行する必要があります。このガイドでは、Nginx インスタンスで実行しなければいけない追加のステップを説明します。

前提条件

以下の前提条件を完了します (まだの場合)。

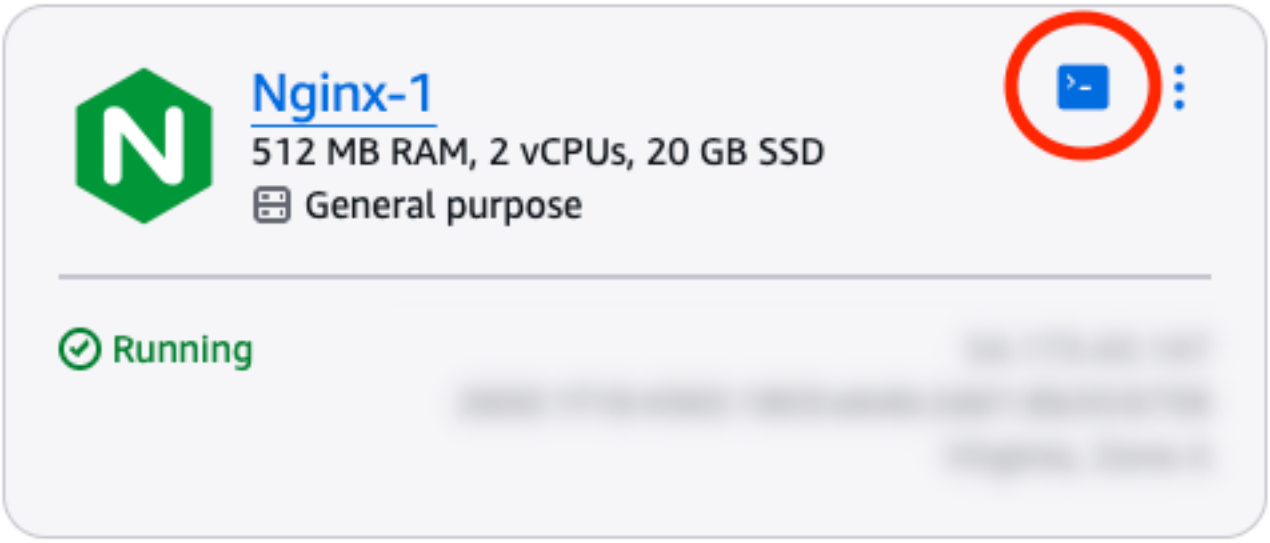
- Lightsail で Nginx インスタンスを作成します。詳細については、「インスタンスを作成する」を参照してください。
- Nginx インスタンスでI Pv6 を有効化します。詳細については、「IPv6 を有効化または無効化する」を参照してください。

> **Note**
>
> 2021 年 1 月 12 日以降に作成された新しい Nginx インスタンスは、Lightsail コンソールで作成されている場合、IPv6 がデフォルトで有効になっています。インスタンスの作成時に IPv6 がデフォルトで有効になっていても、インスタンスで IPv6 を設定するには、このガイドの以下の手順を実行する必要があります。

Nginx インスタンスで IPv6 を構成する

Lightsail の Nginx インスタンスで IPv6 を設定するには、以下のステップを実行します。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの [インスタンス] セクションで、設定する Ubuntu インスタンスを探し、SSH を使用して接続するブラウザベース SSH クライアントアイコンを選択します。



3. インスタンスに接続後、以下のコマンドを入力し、インスタンスがポート 80 で IPv6 リクエストをリッスンしているかを判断します。*<IPv6Address>*をインスタンスに割り当てられた IPv6 アドレスに必ず置き換えてください。

```
curl -g -6 'http://[<IPv6Address>]'
```

例:

```
curl -g -6 'http://[2001:0db8:85a3:0000:0000:8a2e:0370:7334]'
```

以下のようなレスポンスが表示されます。

- インスタンスがポート 80 で IPv6 リクエストをリッスンしていない場合は、接続失敗というエラーメッセージが表示されます。この手順のステップ 4 ~ 9 を続行する必要があります。

```
bitnami@ip-[redacted]:~$ curl -g -6 'http://[2600:1f13:[redacted]:985b:25d9]:80'
curl: (7) Failed to connect to 2600:1f13:[redacted]:985b:25d9 port 80: Connection refused
```

- インスタンスがポート 80 で IPv6 リクエストをリッスンしている場合、以下のようにインスタンスのホームページの HTML コードを含んだレスポンスが表示されます。ここで停止します。インスタンスが IPv6 アドレスを認識するように設定済みなので、この手順のステップ 4 ~ 9 を実行する必要はありません。

```
bitnami@ip-[redacted]:~$ curl -g -6 'http://[2600:[redacted]:985b:25d9]:80'
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Bitnami NGINX Open Source</title>
    <meta name="description" content="Bitnami: Open Source. Simplified.">
    <meta name="author" content="Bitnami">
    <link rel="stylesheet" media="screen" href="//unpkg.com/@bitnami/hex/dist/hex.min.css">
  </head>
  <body>
    <main class="margin-t-huge">
      <section aria-labelledby="installation-title" aria-describedby="installation-desc" class="container container-tiny margin-b-gi
        <h1 id="installation-title">Congratulations!</h1>
        <div aria-hidden="true" style="height: 4px; width: 100%;" class="gradient-135-brand"></div>
        <p id="installation-desc" class="type-big">You are now running <strong>Bitnami NGINX Open Source 1.18.0</strong> in the Clou
      </section>
      <section aria-labelledby="links-title" aria-describedby="links-desc" class="bg-light padding-v-bigger margin-v-enormous">
        <div class="container container-tiny">
          <div class="row row-collapse-b-tablet align-center ">
            <div class="col-6">
              <h3 id="links-title" class="margin-t-reset">Useful Links</h3>
              <p id="links-desc" class="margin-b-reset">The following links will help you to understand better how to get started an
unched.</p>
```

4. 次のコマンドを入力し、Vim を使用して nginx.conf 設定ファイルを開きます。

```
sudo vim /opt/bitnami/nginx/conf/nginx.conf
```

5. I キーを押して Vim 挿入モードにします。
6. 以下のテキストを`listen 80;` ファイルに既に存在しているテキストに追加します。Vim で下にスクロールして、 テキストを追加するセクションを見つける必要があるかもしれません。

```
listen [::]:80;
```

完了したらファイルは以下のようになります。

```
    server_tokens off;

    include  "/opt/bitnami/nginx/conf/server_blocks/*.conf";

    # HTTP Server
    server {
        # Port to listen on, can also be set in IP:PORT format
        listen  80;
        listen [::]:80;

        include  "/opt/bitnami/nginx/conf/bitnami/*.conf";

        location /status {
            stub_status on;
            access_log   off;
            allow 127.0.0.1;
            deny all;
        }
    }
```

7. ESC キーを押して Vim 挿入モードを終了して、:wq! を入力してEnterを押して編集内容を保存し、Vim を終了します。
8. 以下のコマンドを入力して、インスタンスのサービスを再起動します。

```
sudo /opt/bitnami/ctlscript.sh restart
```

9. 以下のコマンドを入力して、インスタンスがポート 80 で IPv6 リクエストをリッスンしているかを判断します。*<IPv6Address>*をインスタンスに割り当てられた IPv6 アドレスに必ず置き換えてください。

```
curl -g -6 'http://[<IPv6Address>]'
```

例:

```
curl -g -6 'http://[2001:0db8:85a3:0000:0000:8a2e:0370:7334]'
```

以下のようなレスポンスが表示されます。インスタンスがポート 80 で IPv6 リクエストをリッスンしている場合、インスタンスのホームページの HTML コードを含むレスポンスが表示されます。

Lightsail で Plesk インスタンスに IPv6 接続を設定する

Plesk ブループリントを使用するインスタンスに IPv6 アドレスを認識させる追加のステップを実行する必要があります。このガイドでは、Plesk インスタンスで実行しなければいけない追加のステップを説明します。

前提条件

以下の前提条件を完了します (まだの場合)。

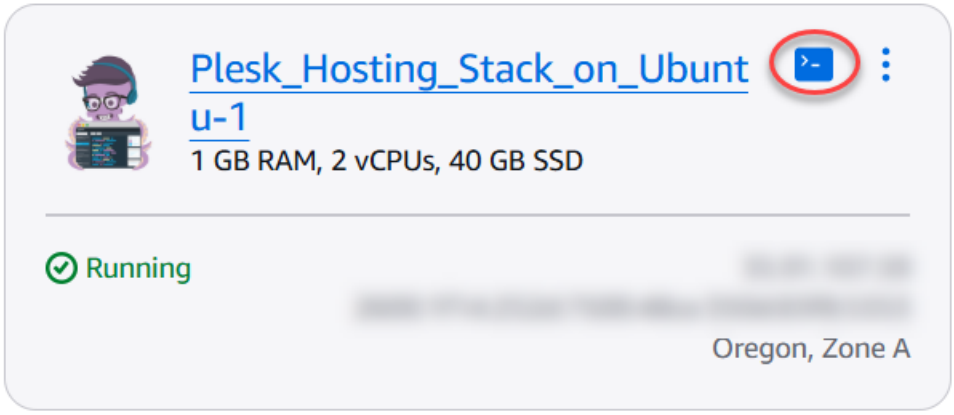
- Lightsail で Plesk インスタンスを作成する。詳細については、「インスタンスを作成する」を参照してください。
- Plesk インスタンスで IPv6 を有効化します。詳細については、「IPv6 を有効化または無効化する」を参照してください。

> **Note**
>
> 2021 年 1 月 12 日以降に作成された Lightsail Plesk インスタンスでは、IPv6 がデフォルトで有効になっています。インスタンスの作成時に IPv6 がデフォルトで有効になっていても、インスタンスで IPv6 を設定するには、このガイドの以下のステップを実行する必要があります。

Plesk インスタンスで IPv6 を設定する

Lightsail の Plesk インスタンスで IPv6 を設定するには、以下のステップを実行します。

1. Lightsail コンソール にサインインします。
2. Lightsail ホームページの [インスタンス] セクションで設定したい Plesk インスタンスを探し、ブラウザベース SSH クライアントアイコンを選択し、SSH を使用して接続します。

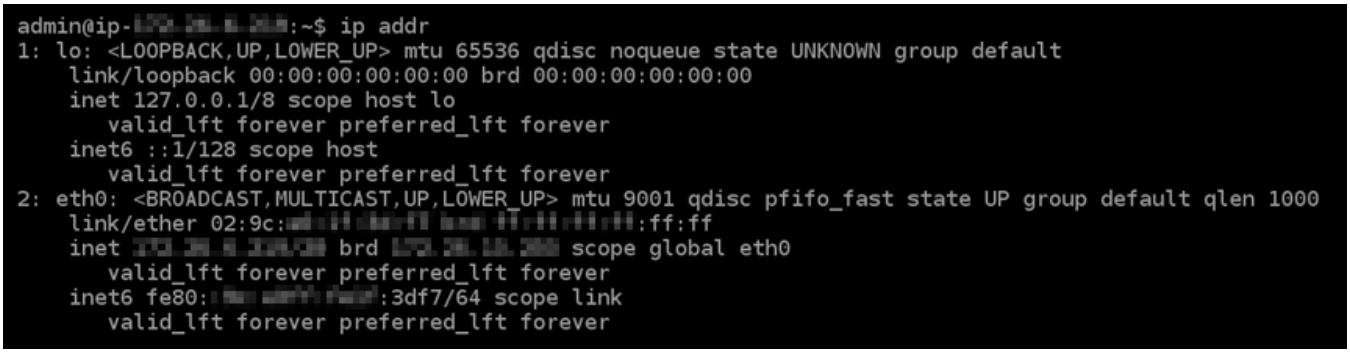


3. インスタンスに接続したら、以下のコマンドを入力し、インスタンスに設定されている IP アドレスを表示します。

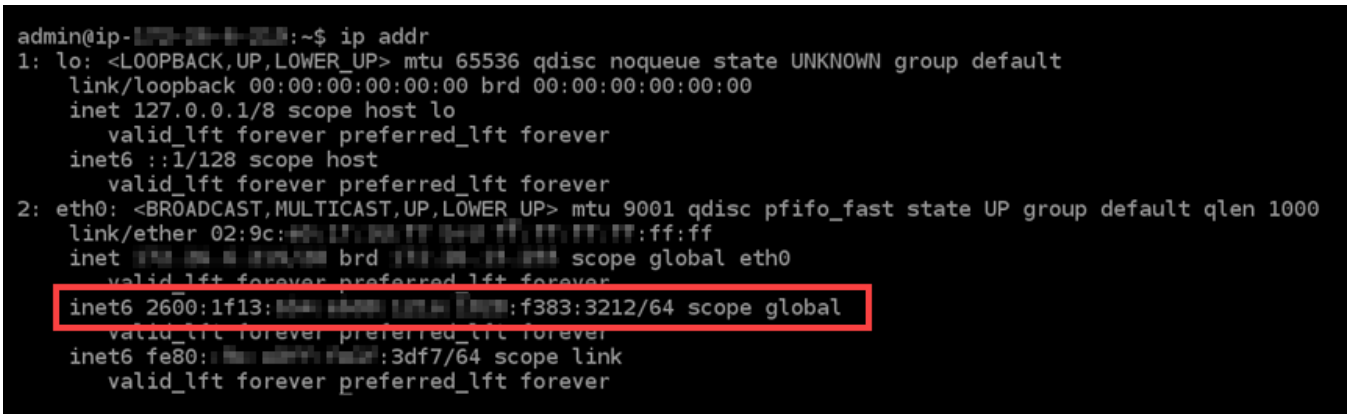
```
ip addr
```

以下のようなレスポンスが表示されます。

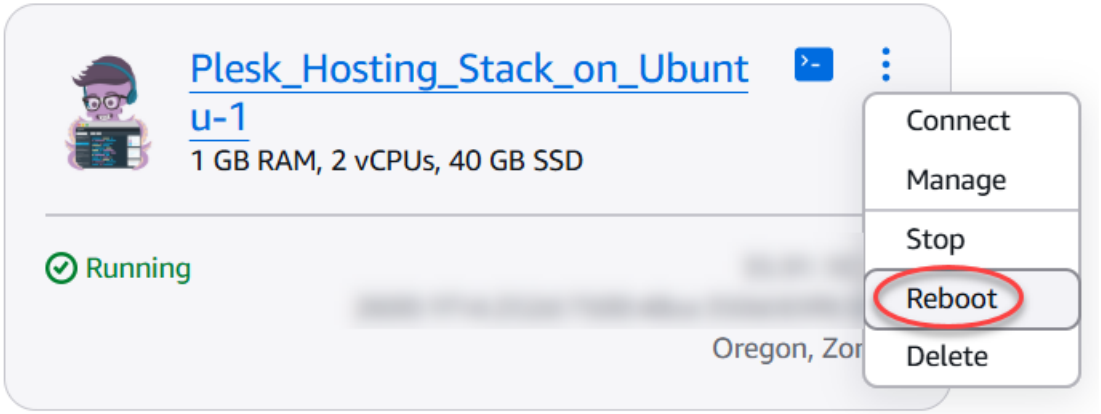
- インスタンスが IPv6 アドレスを認識しない場合、レスポンスに表示されません。この手順のステップ 4～7 を続行する必要があります。



- インスタンスが IPv6 アドレスを認識している場合は、レスポンスに`scope global` の例のように表示されます。ここで停止します。インスタンスが IPv6 アドレスを認識するように設定済みなので、この手順のステップ 4～7 を実行する必要はありません。



4. Lightsail コンソールに戻ります。
5. Lightsail ホームページの [インスタンス] セクションで、Plesk インスタンスのアクションメニュー (⋮) を選択し、[再起動] を選択します。

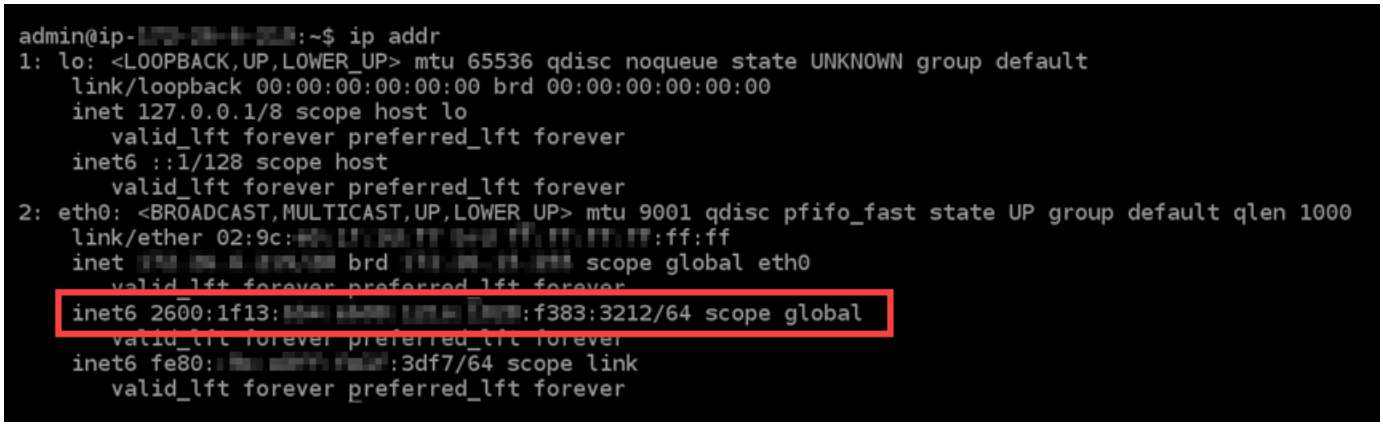


インスタンスが再起動するまで数分待ってから、次のステップに進みます。

6. Plesk インスタンスの SSH セッションに戻ります。
7. 以下のコマンドを入力して、インスタンスに設定された IP アドレスを表示し、割り当てられた IPv6 アドレスを認識していることを確認します。

```
ip addr
```

以下のようなレスポンスが表示されます。インスタンスが IPv6 アドレスを認識している場合は、レスポンスにラベルscope global の例ように表示されます。



Lightsail インスタンスブループリントで IPv6 を設定する方法については、ステップバイステップの手順に従います。

このガイドでは、cPanel、GitLab、Nginx、Plesk など、さまざまなインスタンスのブループリントについて説明します。手順には、SSH 経由でインスタンスに接続し、ネットワーク設定ファイルを変更し、サービスを再起動し、インスタンスが割り当てられた IPv6 アドレスを認識することを確認することが含まれます。このガイドに従うことで、Lightsail インスタンスが IPv4 アドレスと IPv6

アドレスの両方を使用するように適切に設定され、より良い接続が可能になり、インターネットの将来に向けてアプリケーションを準備することができます。

Lightsail オペレーション AWS CLI 用の をセットアップおよび設定する

AWS Command Line Interface (AWS CLI) は、上級ユーザーと開発者がターミナル (Linux および Unix) または コマンドプロンプト (Windows) にコマンドを入力してAmazon Lightsailサービスを制御できるようにするツールです。また、Lightsail コンソール、グラフィカルユーザーインターフェイス、および Lightsail アプリケーションプログラムインターフェイス (API) を使用して、Lightsail を制御することもできます。

> **Tip**
>
> AWS CloudShell を使用して、 AWS CLI コマンドラインツールをダウンロードまたはインストールせずにコマンドを実行してLightsailリソースを管理することもできます。CloudShell はブラウザーベースの事前認証済みシェルで、Lightsail コンソールから直接起動できます。詳細については、「でLightsailリソースを管理する AWS CloudShell」を参照してください。

トピック

- ステップ 1: をインストールする AWS CLI
- ステップ 2: 新しいアクセスキーを作成する
- ステップ 3: を設定する AWS CLI
- 次の手順

ステップ 1: をインストールする AWS CLI

をローカルデスクトップ AWS CLI にインストールすることも、Lightsailインスタンスにインストールすることもできます。の詳細については AWS CLI、AWS Command Line Interface 「 ユーザーガイド」を参照してください。

- をローカルデスクトップ AWS CLI にインストールするには、 AWS Command Line Interface ドキュメントの「 AWS CLIのインストール」を参照してください。
- Ubuntu ベースのLightsailインスタンス AWS CLI に をインストールするには、インスタンスに接続し、 と入力しますsudo apt-get -y install awscli。

> **Note**
>
> は Amazon Linux Lightsailインスタンスに既にインストール AWS CLI されている必要があります。再インストールする必要がある場合は、インスタンスに接続し、「`sudo yum install aws-cli`」と入力します。

をインストールしたら AWS CLI、アクセスキーを生成し、それらを使用するように AWS CLI を設定する必要があります。

ステップ 2: 新しいアクセスキーを作成する

Lightsail API または AWS Command Line Interface (AWS CLI) を使用するには、新しいアクセスキーを作成する必要があります。アクセスキーは、アクセスキー ID とシークレットアクセスキーで構成されます。次の手順に従ってキーを作成します。

1. IAM コンソールにサインインします。
2. アクセスキーを作成するユーザーの名前を選択します。選択するユーザーには、Lightsail アクションを行うフルアクセス権または特定のアクセス権が必要です。
3. [Security credentials] タブを選択します。
4. [アクセスキー] セクションで、[アクセスキーの作成] を選択します。

   > **Note**
   >
   > 一度に 1 人のユーザーが持つことができるのは、最大 2 つのアクセスキー (有効または無効) です。すでに 2 つある場合は、新しいものを作成する前に、いずれかを削除する必要があります。アクセスキーを削除する前に、アクセスキーがアクティブに使用されていないことを確認してください。

5. アクセスキー ID とリストされたシークレットアクセスキーを記録します。[シークレットアクセスキー] 列の [Show] を選択して、シークレットアクセスキーを表示します。

   これらをこの画面からコピーするか、[キーファイルのダウンロード] を選択して、アクセスキー ID とシークレットアクセスキーが含まれている `.csv` ファイルをダウンロードします。

> **Important**
>
> アクセスキーを安全な場所に保管します。後で見つけやすいように、そのファイルに `MyLightsailKeys.csv` のような名前を付けます。IAM コンソールから CSV ファイルをダウンロードした場合は、次のステップの完了後に削除する必要があります。後で新しいアクセスキーを作成できます。

ステップ 3: を設定する AWS CLI

アクセスキーを使用する AWS CLI ように を設定する必要があります。

1. ターミナルウィンドウまたはコマンドプロントを開きます。
2. 「`aws configure`」と入力します。
3. 前のステップで作成した `.csv` ファイルから AWS アクセスキー ID を貼り付けます。
4. 入力を求められたら、AWS シークレットアクセスキーを貼り付けます。
5. リソース AWS リージョン が配置されている を入力します。たとえば、リソースが主に Ohio に置かれている場合は、[`us-east-2`デフォルトリージョン名] の入力を求められたときに [] を選択します。

   オプションの使用の詳細については、 AWS CLI リファレンスの AWS CLI `--region`「一般的なオプション」を参照してください。
6. [Default output format (デフォルトの出力形式)] (`json` など) を選択します。

を使用してLightsailプログラムで を操作できるようになりました AWS CLI。「AWS CLI コマンドリファレンス」で Amazon Lightsail コマンドを確認できます。

次の手順

以下のリソースは、言語固有の AWS SDK を開始し、Lightsail API に慣れるのに役立ちます。

- 言語固有の AWS SDKs をインストールする
- Lightsail API リファレンスを確認する

でLightsailリソースを管理する AWS CloudShell

AWS CloudShell は、 Amazon Lightsailコンソールから直接起動できるブラウザベースの事前認証済みシェルです。CloudShell を使用して、コマンドラインインターフェイスから Lightsail リソースを管理します。Bash、PowerShell、Z シェルなど、任意のシェルを使用して AWS Command Line Interface (AWS CLI) コマンドを実行できます。この手順は、コマンドラインツールのダウンロードもインストールも不要です。詳細については、「 とは AWS CloudShell」を参照してください。

CloudShell を起動すると、Amazon Linux 2 に基づいた コンピューティング環境 が作成されます。この環境では、 AWS CLIなど、プリインストールされている広範な開発ツールにアクセスできます。プリインストールされているツールの全リストについては、「CloudShell ユーザーガイド」の「プリインストールソフトウェア」を参照してください。

永続ストレージ

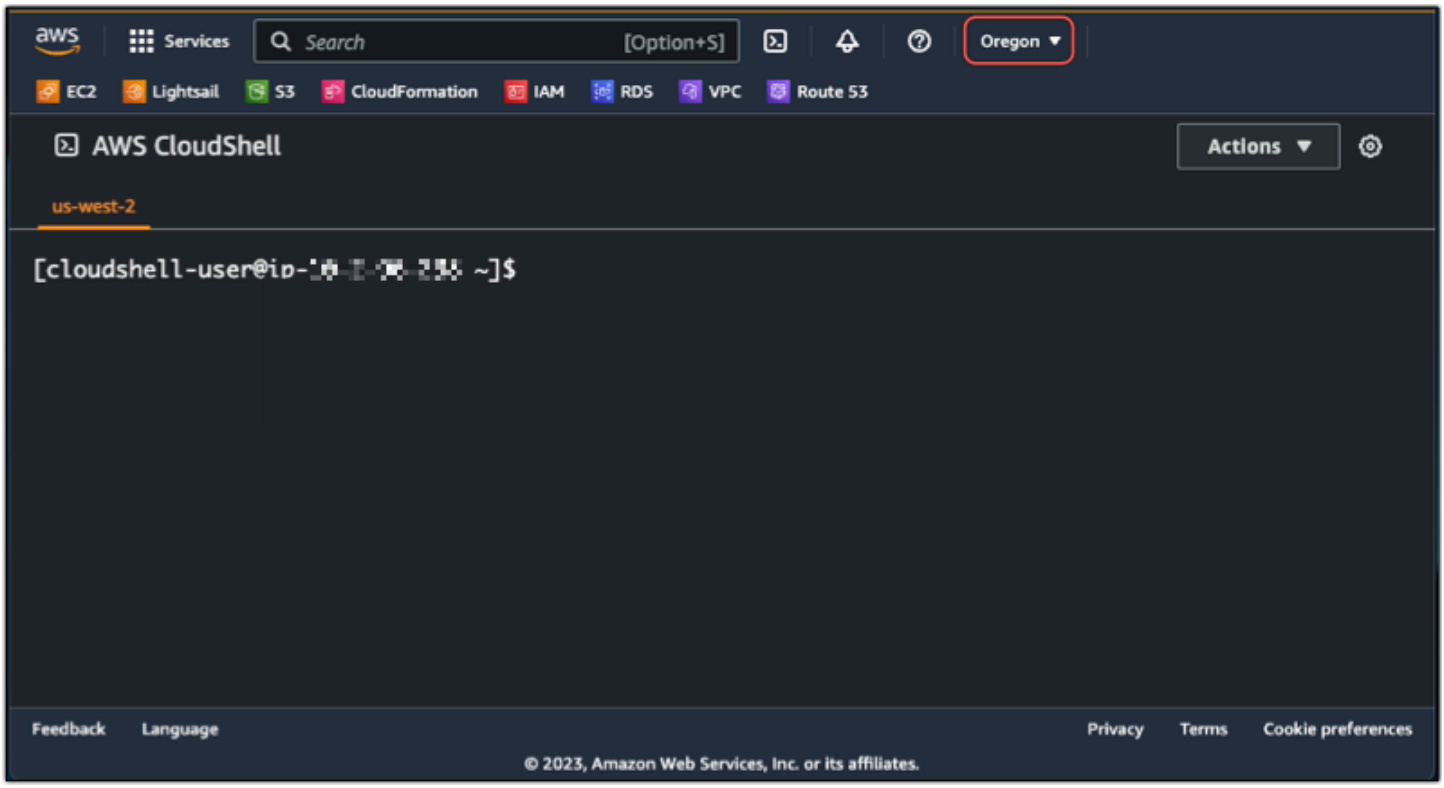
を使用すると AWS CloudShell、追加料金 AWS リージョン なしで各 で最大 1 GB の永続的ストレージを使用できます。永続的ストレージはホームディレクトリ (`$HOME`) にあり、ユーザーのプライベートな記憶域です。各シェルセッションが終了した後に削除されるエフェメラル環境リソースとは異なり、ホームディレクトリ内のデータはセッション間で保持されます。

AWS CloudShell で の使用を停止すると AWS リージョン、データは最後のセッションの終了から 120 日間、そのリージョンの永続ストレージに保持されます。アクションを実行しない限り、データは 120 日後にそのリージョンの永続的ストレージから自動的に削除されます。その AWS CloudShell で AWS リージョンを再度起動すれば削除を防止することができます。永続ストレージでのデータ保持の詳細については、「CloudShell ユーザーガイド」の「永続ストレージ」を参照してください。

AWS リージョン

ではLightsail、CloudShell セッションが で開き AWS リージョン 、物理的な場所へのレイテンシーが最も少なくなります。つまり、 はセッション間で変更 AWS リージョン される可能性があります。1 GB の永続ストレージを使用できるように、CloudShell セッションがどの AWS リージョンにあるかに注意してください。セッションの AWS リージョンを変更するには、[新しいブラウザタブで開く] アイコンを選択します。これにより、新しいブラウザウィンドウで CloudShell セッションにアクセスするオプションを使用できます。

新しいブラウザタブのナビゲーションバーで、現在表示されている AWS リージョン の名前を選択します。次に、切り替え AWS リージョン る を選択します。



CloudShell の詳細については、「CloudShell ユーザーガイド」を参照してください。

を起動して使用する AWS CloudShell

内で AWS CloudShell セッションを起動して使用する方法について説明しますLightsail。CloudShell を実行するアクセス許可がない場合は、使用している AWS Identity and Access Management (IAM)

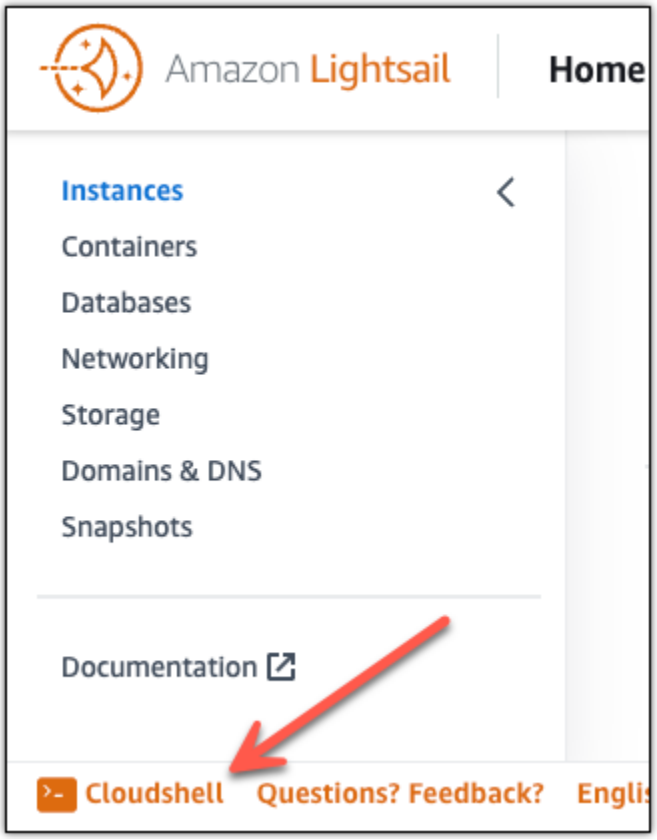
ID にarn:aws:iam::aws:policy/AWSCloudShellFullAccessポリシーを追加する必要があります。arn:aws:iam::aws:policy/AdministratorAccess ポリシーがすでにアタッチされている場合は、CloudShell にアクセスできるはずです。詳細については、「???」を参照してください。

起動 AWS CloudShell

Amazon Lightsail コンソールから CloudShell を起動できます。セッションが開始されたら、Bash、PowerShell、または Z shell などのお好みのシェルに切り替えることができます。

で新しい AWS CloudShell セッションを起動するには、次の手順を実行しますLightsail。

1. https://lightsail.aws.amazon.com/ で Lightsailコンソールにサインインします。
2. コンソールの左下にあるコンソールツールバーで [CloudShell] を選択します。コマンドプロンプトが表示されたら、シェルは対話的な操作の準備ができています。



3. (オプション) 使用するプリインストールされたシェルを選択するには、コマンドラインプロンプトで次のプログラム名のいずれかを入力します。

Bash: **bash**

Bash に切り替えると、コマンドプロンプトの記号が$にアップロードします。Bash はデフォルトのシェルです AWS CloudShell。

PowerShell: **pwsh**

PowerShell に切り替えると、コマンドプロンプトの記号が更新されて PS> になります。

Z シェル: **zsh**

Z シェルに切り替えると、コマンドプロンプトの記号が%に講アップロードします。

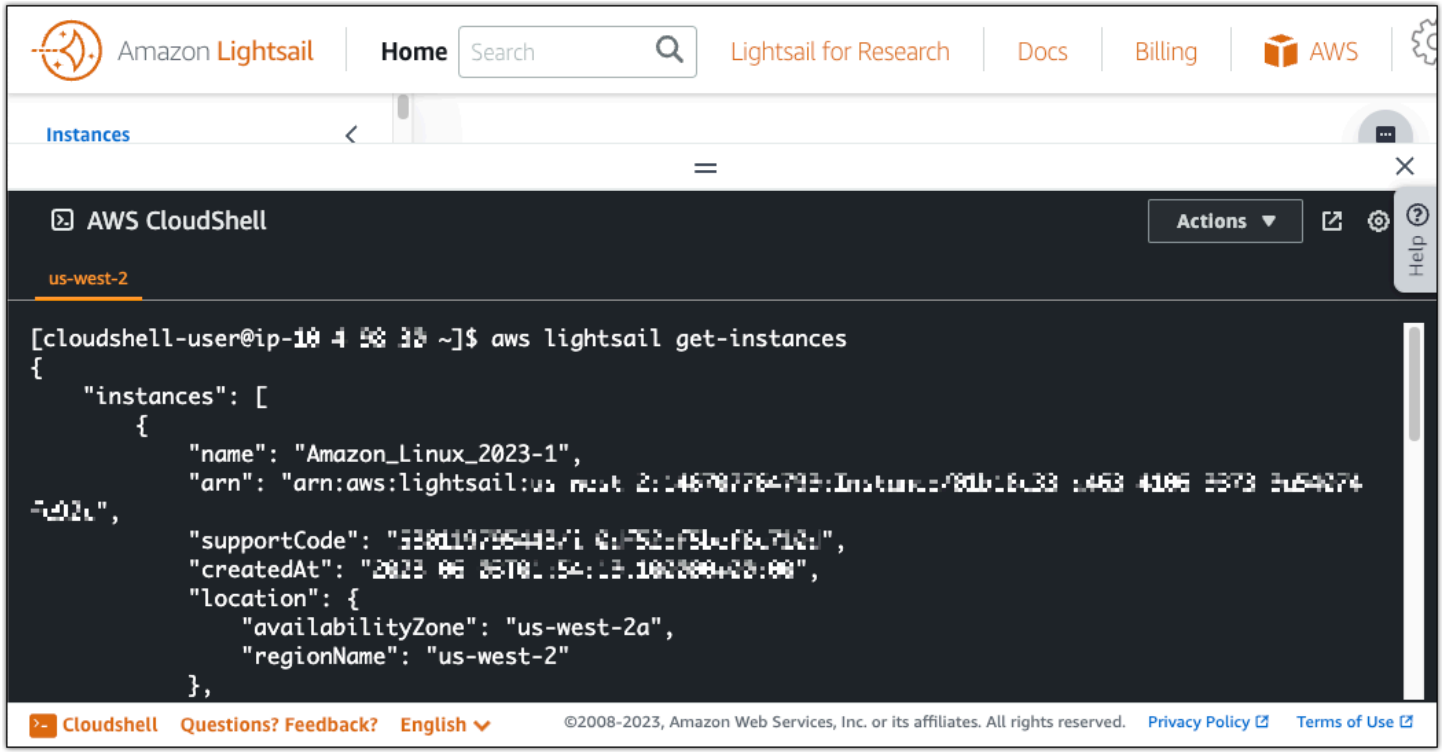
Exampleの Lightsail API コマンドの例 AWS CloudShell

CloudShell セッションには、使用できる複数のコマンドラインツールがプリインストールされています。この例では、Lightsail `GetInstances` API オペレーションを使用して Lightsail アカウント内にあるインスタンスを表示します。`GetInstances` API オペレーションの詳細については、「Amazon Lightsail API リファレンス」の「GetInstances」を参照してください。

1. https://lightsail.aws.amazon.com/ で Lightsailコンソールにサインインします。
2. コンソールの左下にあるコンソールツールバーで [CloudShell] を選択します。
3. AWS CloudShell プロンプトの後に次のコマンドを入力します。

```
aws lightsail get-instances
```

これで、Lightsail アカウントにあるインスタンスの全リストが表示されるはずです。



追加情報

詳細については、次のドキュメントを参照してください AWS CloudShell。

- Amazon Lightsail API リファレンス
- でのよくある質問 AWS CloudShell
- でサポートされているブラウザ AWS CloudShell
- でのトラブルシューティング AWS CloudShell
- AWS のサービス での の使用 AWS CloudShell

Lightsail LAMP インスタンスに PHP アプリケーションをデプロイする

仮想プライベートサーバーだけが必要な場合、Amazon Lightsail は Amazon Web Services (AWS) を始める最も簡単な方法です。Lightsail には、プロジェクトを迅速に立ち上げるために必要なものがすべて揃っています。予測可能な低料金で仮想マシン、SSD ベースのストレージ、データ転送、DNS 管理、静的 IP が提供されます。

このチュートリアルでは、Lightsail で LAMP インスタンスを起動して設定する方法を示します。SSH 経由でのインスタンスへの接続、インスタンスのアプリケーションパスワードの取得、静的 IP の作成とインスタンスへのアタッチ、DNS ゾーンの作成とドメインのマッピングに関するステップが含まれています。このチュートリアルを完了すると、Lightsail でインスタンスを起動して使用するための基礎固めができます。

目次

- ステップ 1: AWS にサインアップ
- ステップ 2: LAMP インスタンスを作成する
- ステップ 3: SSH 経由でインスタンスに接続し、LAMP インスタンスのアプリケーションパスワードを取得します。
- ステップ 4: LAMP インスタンス上にアプリケーションをインストールする
- ステップ 5: 静的 IP アドレスを作成して LAMP インスタンスにアタッチする
- ステップ 6: DNS ゾーンを作成し、ドメインを LAMP インスタンスにマッピングする
- 次のステップ

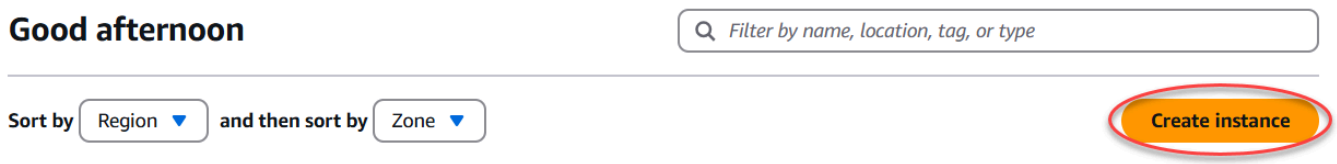
ステップ 1: AWS にサインアップ

このチュートリアルでは、 AWS アカウントが必要です。にサインアップ AWSするか、すでに アカウントをお持ちの場合は にサインイン AWSします。

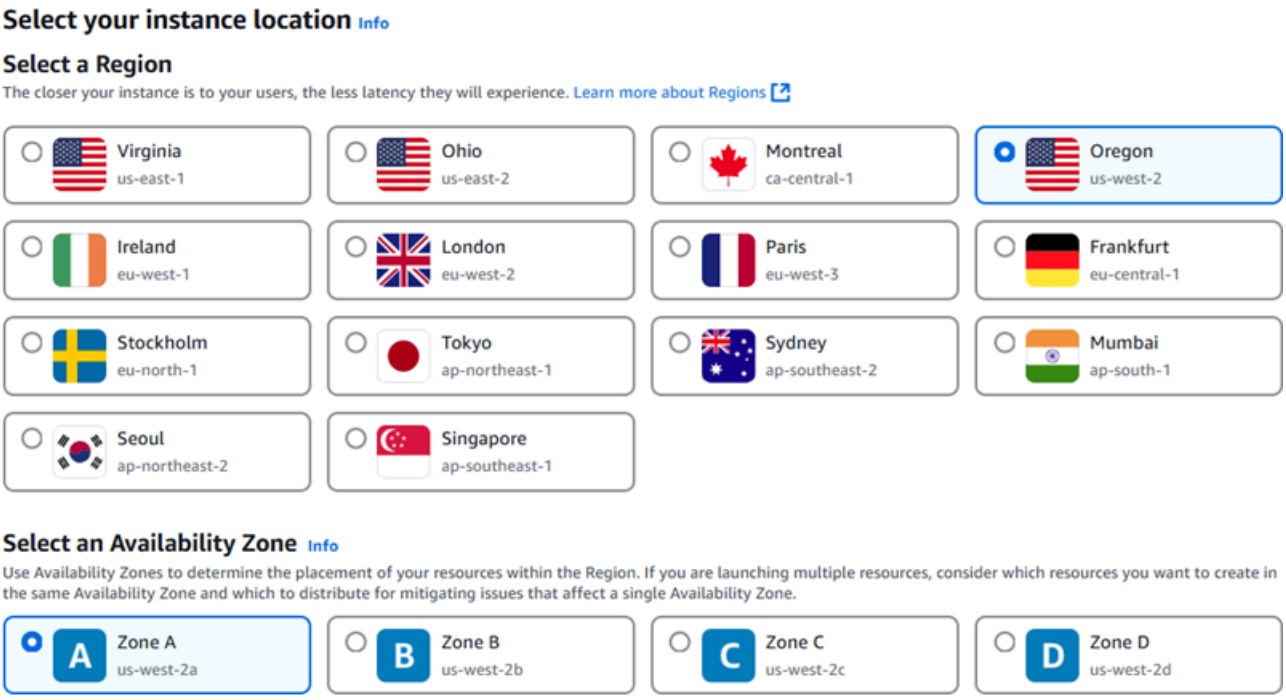
ステップ 2: LAMP インスタンスを作成する

Lightsail で LAMP インスタンスを起動します。Lightsail でのインスタンス作成の詳細については、「Lightsail でのAmazon Lightsail インスタンスの作成」ドキュメントを参照してください。

1. Lightsail コンソール にサインインします。
2. Lightsail ホーム画面の [インスタンス] セクションで、[インスタンスの作成] を選択します。



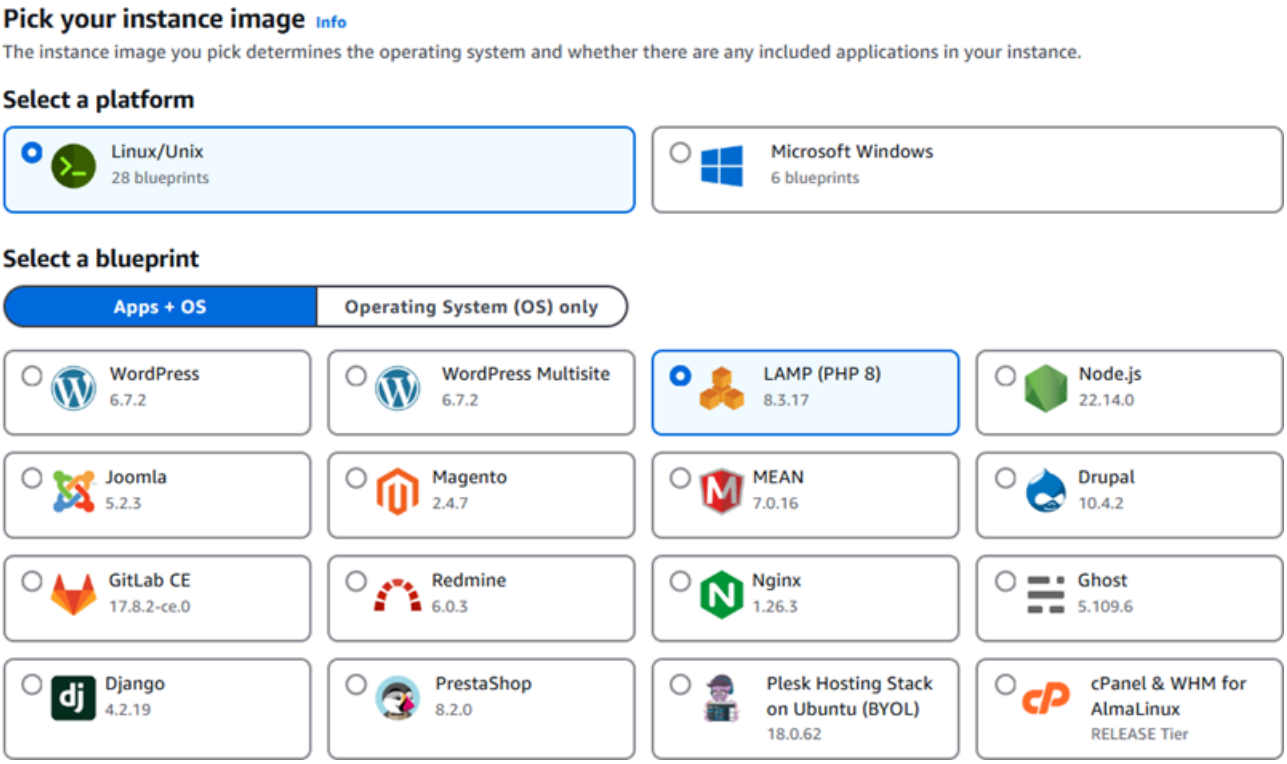
3. インスタンスの AWS リージョン とアベイラビリティーゾーンを選択します。



4. インスタンスイメージを選択します。

   a. プラットフォームとして [Linux/Unix] を選択します。

   b. ブループリントとして [LAMP (PHP 8)] を選択します。



5. インスタンスプランを選択します。

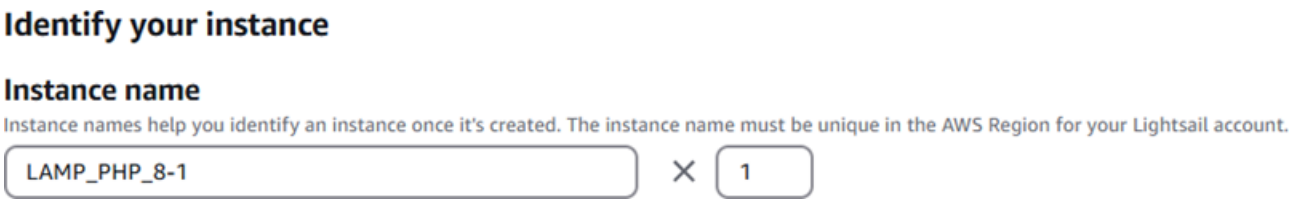
   プランには、低額で予測可能なコスト、マシン設定 (RAM、SSD、vCPU)、およびデータ転送枠が含まれます。5 USD Lightsailプランを 1 か月間 (最大 750 時間) 無料で試すことができます。は 1 か月分の無料クレジットをアカウントに AWS 付与します。

   > **Note**
   >
   > AWS 無料利用枠の一部として、一部のインスタンスバンドルで Amazon Lightsailを無料で開始できます。詳細については、「Amazon Lightsail 料金表」ページの「AWS Free Tier」を参照してください。

6. インスタンスの名前を入力します。

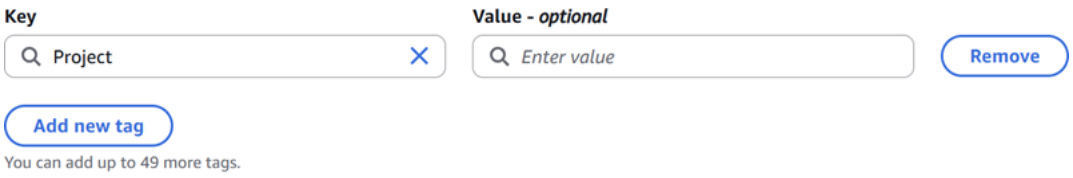
   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

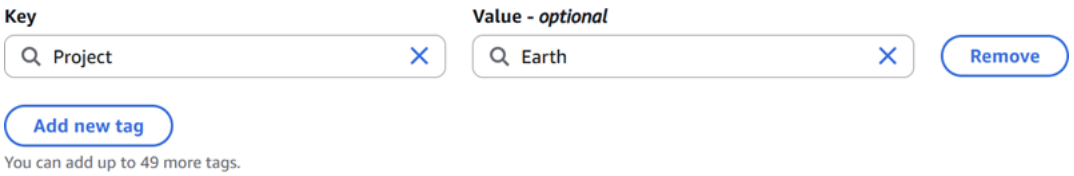
   

7. (オプション)[新しいタグの追加] を選択してインスタンスにタグを追加します。必要に応じてこの手順を繰り返して、他のタグを追加します。タグの使用の詳細については、「タグ」を参照してください。

   a. [キー] には、タグキーを入力します。

   b. (オプション) [値] にタグの値を入力します。

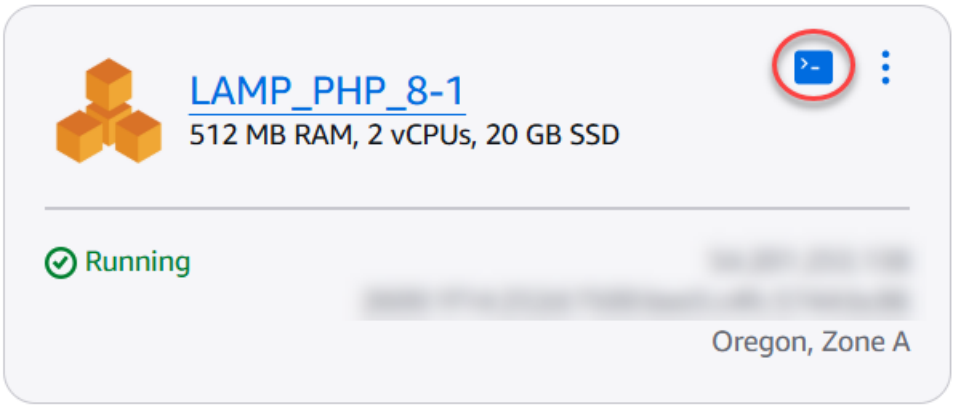


8. [インスタンスの作成] を選択します。

ステップ 3: SSH 経由でインスタンスに接続し、LAMP インスタンスのアプリケーションパスワードを取得します。

LAMP のデータベースにサインインするためのデフォルトのパスワードがインスタンスに保存されます。これを取得するには、Lightsail コンソールでブラウザベースの SSH ターミナルを使用してインスタンスに接続し、専用のコマンドを実行します。詳細については、「Amazon Lightsailの Bitnami インスタンス向けにアプリケーションのユーザー名とパスワードを取得する」を参照してください。

1. Lightsail ホームページの [インスタンス] セクションで、LAMP インスタンスの [SSH クイック接続] アイコンを選択します。



2. ブラウザベースの SSH クライアントのウィンドウが表示されたら、次のコマンドを入力してデフォルトのアプリケーションのパスワードを取得します。

```
cat bitnami_application_password
```

> **Note**
>
> ユーザーのホームディレクトリ以外のディレクトリで作業している場合は、「`cat $HOME/bitnami_application_password`」と入力します。

3. 画面に表示されたパスワードを書き留めておきます。このパスワードを後で使用して Bitnami アプリケーションをインスタンスにインストールします。または、ユーザー名 `root` を使用して MySQL データベースにアクセスします。

ステップ 4: LAMP インスタンス上にアプリケーションをインストールする

LAMP インスタンス上に PHP アプリケーションをデプロイするか、Bitnami アプリケーションをインストールします。PHP アプリケーションをデプロイするメインディレクトリは `/opt/bitnami/apache2/htdocs` です。このディレクトリに PHP アプリケーションのファイルをコピーし、インスタンスのパブリック IP アドレスを参照してアプリケーションにアクセスします。

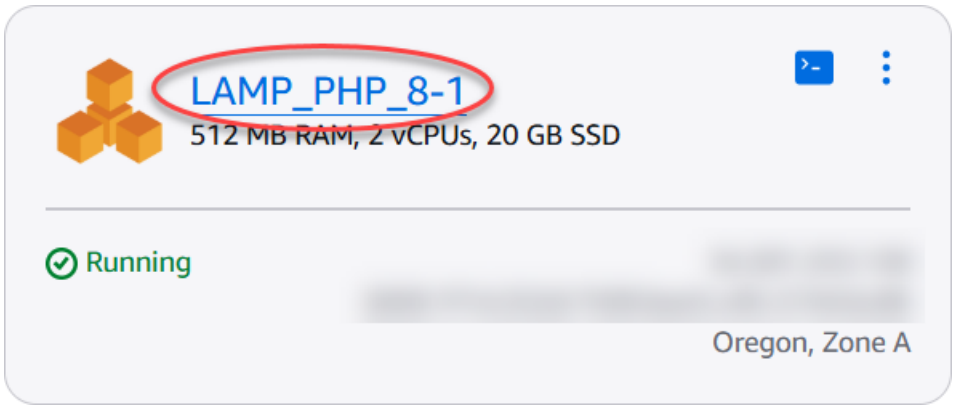
Bitnami アプリケーションは、モジュールのインストーラを使用してインストールすることもできます。WordPress、Drupal、Magento、Moodle などのアプリケーションを Bitnami ウェブサイトからダウンロードし、サーバーの機能を拡張します。Bitnami アプリケーションのインストールの詳細については、Bitnami のドキュメントで「開始」を参照してください。

ステップ 5: 静的 IP アドレスを作成して LAMP インスタンスにアタッチする

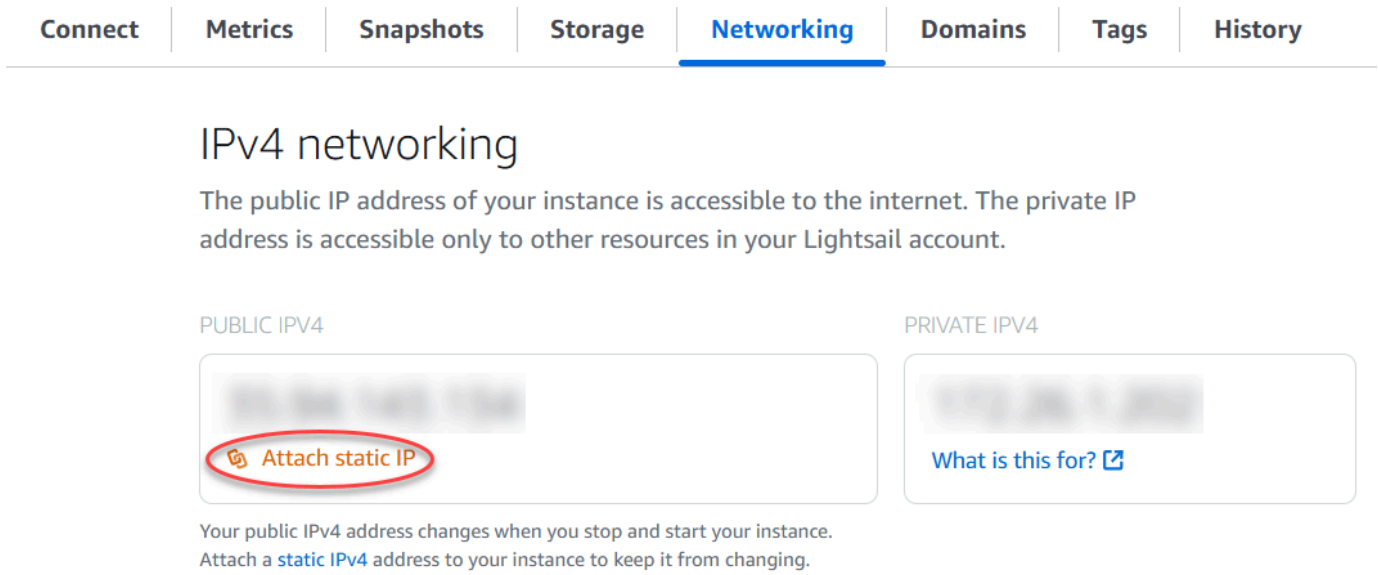
LAMP インスタンスのデフォルトのパブリック IP は、インスタンスを停止して開始すると変わります。インスタンスにアタッチした静的 IP アドレスは、インスタンスを停止して開始しても変わりません。

静的 IP アドレスを作成して LAMP インスタンスにアタッチします。詳細については、Lightsail ドキュメントの「静的 IP を作成してインスタンスにアタッチする」を参照してください。

1. Lightsail ホームページの [インスタンス] セクションで、実行している LAMP インスタンスを選択します。



2. [ネットワーキング] タブを開き、次に [静的 IP をアタッチする] を選択します。



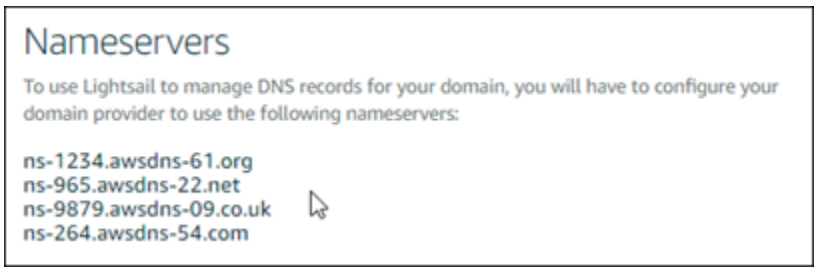
3. 静的 IP に名前を付け、[作成してアタッチ] を選択します。

ステップ 6: DNS ゾーンを作成し、ドメインを LAMP インスタンスにマッピングする

ドメインの DNS レコードの管理を Lightsail に引き渡します。これにより、ドメインを LAMP インスタンスにマッピングしやすくなり、ウェブサイトのすべてのリソースを Lightsail コンソールで管理できます。詳細については、「DNS ゾーンを作成し、ドメインの DNS レコードを管理する」を参照してください。

1. Lightsail ホームページの [ドメインと DNS] セクションで、[DNS ゾーンの作成] を選択します。
2. ドメインを入力し、[DNS ゾーンの作成] を選択します。
3. ページに表示されたネームサーバーのアドレスを書き留めておきます。

   ドメインの DNS レコードの管理を Lightsail に引き渡すには、これらのネームサーバーのアドレスをドメイン名のレジストラに追加します。



4. ドメインの DNS レコードの管理を Lightsail に引き渡したら、次に示すように、A レコードを追加してドメインの apex が LAMP インスタンスをポイントするように設定します。

   a. DNS ゾーンの [Assignments] (割り当て) タブで [Add assignment] (割り当てを追加) を選択します。

   b. [Select a domain] (ドメインの選択) フィールドで、ドメインまたはサブドメインを選択します。

   c. [Select a resource] (リソースの選択) ドロップダウンで、このチュートリアルで以前に作成した LAMP インスタンスを選択します。

   d. [Assign] (割り当て) を選択します。

   変更内容がインターネットの DNS を通じて伝播されるまで待ってから、LAMP インスタンスへのトラフィックのルーティングを開始します。

次の手順

Amazon Lightsail で LAMP インスタンスを起動した後で、以下の追加のステップを実行できます。

- Linux または Unix インスタンスのスナップショットを作成する
- 追加のブロックストレージディスクを作成して Linux ベースの インスタンスにアタッチする

Lightsail LAMP インスタンスを Aurora データベースに接続する

投稿、ページ、およびユーザーのアプリケーションデータは、Amazon Lightsail の LAMP インスタンスで実行している MariaDB データベースに保存されています。WordPress インスタンスに障害が発生した場合、データが回復不可能になる場合があります。このシナリオを回避するには、 MySQL マネージドデータベースにアプリケーションのデータを転送する必要があります。

Amazon Aurora はクラウド用に構築された MySQL と PostgreSQL 互換のリレーショナルデータベースです。これは従来のエンタープライズデータベースのパフォーマンスと可用性に、オープンソースデータベースのシンプルさと費用対効果を組み合わせています。Aurora は、Amazon Relational Database Service (Amazon RDS) の一部として提供されています。Amazon RDS は、クラウドでリレーショナルデータベースを簡単に設定、運用、およびスケールすることができるマネージドデータベースサービスです。詳細については、「Amazon Relational Database Service ユーザーガイド」と「Aurora の Amazon Aurora ユーザーガイド」を参照してください。

このチュートリアルでは、Lightsail 内の LAMP インスタンスからアプリケーションデータベースを Amazon RDS 内の Aurora マネージドデータベースに接続する方法について説明します。

目次

- ステップ 1: 前提条件を満たす
- ステップ 2: Aurora データベースのセキュリティグループを設定する
- ステップ 3: Lightsail インスタンスから Aurora データベースに接続する
- ステップ 4: MariaDB データベースを LAMP インスタンスから Aurora データベースに転送する
- ステップ 5: Aurora マネージドデータベースに接続するようアプリケーションを設定する

ステップ 1: 前提条件を満たす

開始する前に次の前提条件を完了します。

1. Lightsail で LAMP インスタンスを作成し、アプリケーションを設定します。続行する前に、インスタンスは実行中状態になっていることを確認してください。詳細については、「チュートリアル: Lightsail で LAMP インスタンスを起動して設定する」を参照してください。
2. Lightsail アカウントで VPC ピアリングを有効にします。詳細については、「 以外の AWS リソースを操作するように Amazon VPC ピア接続を設定するLightsail」を参照してください。
3. Amazon RDS に Aurora マネージドデータベースを作成します。データベースは、LAMP リソースと同じ AWS リージョン にある必要があります。続行する前に、データベースが実行中状態になっていることを確認してください。詳細については、「Aurora の Amazon Aurora ユーザーガイド」の「Amazon Aurora の使用開始」を参照してください。

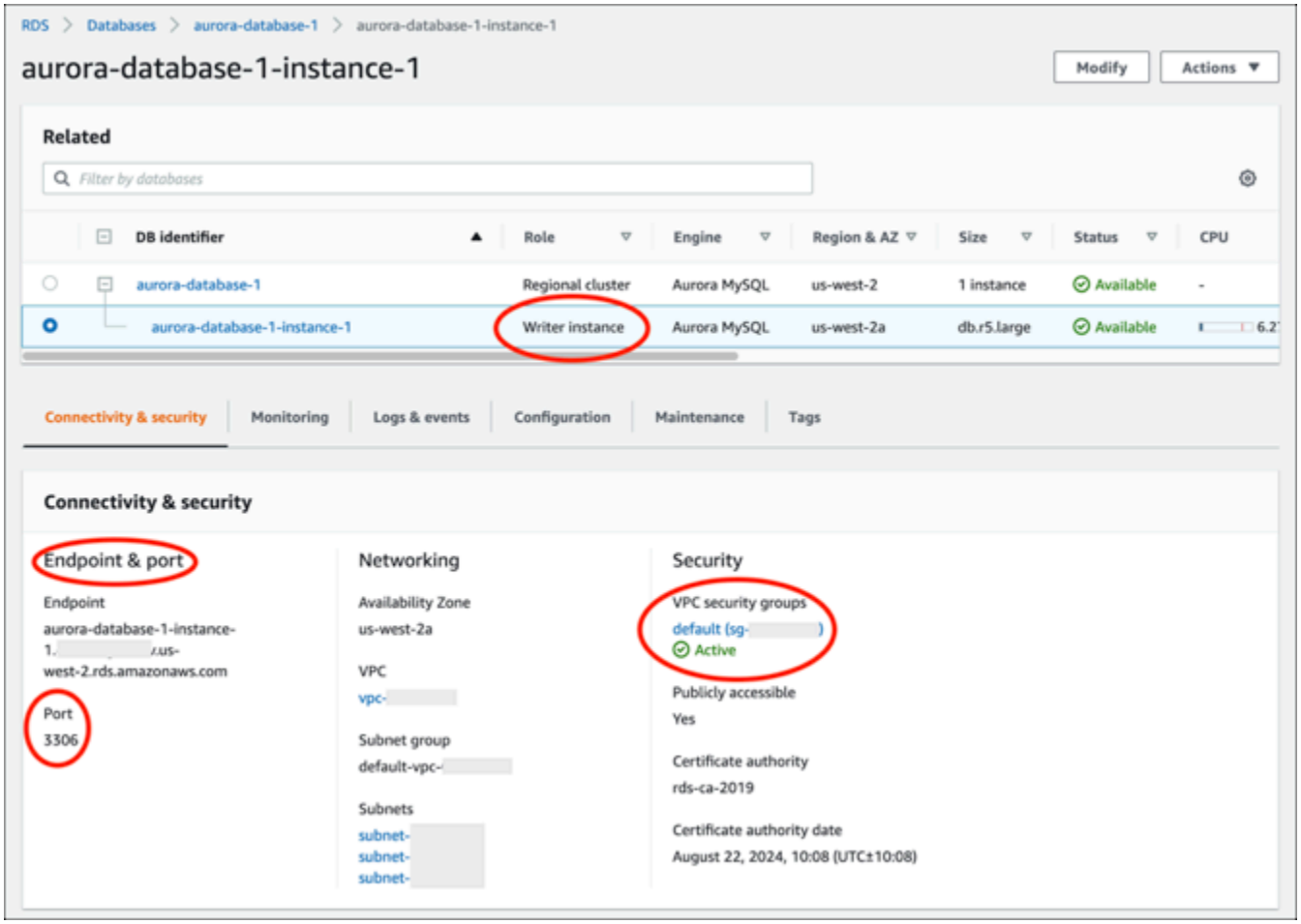
ステップ 2: Aurora データベースのセキュリティグループを設定する

AWS セキュリティグループは、 AWS リソースの仮想ファイアウォールとして機能します。Amazon RDS 内の Aurora データベースに接続できる送受信トラフィックを制御します。詳細については、「Amazon Virtual Private Cloud ユーザーガイドのセキュリティグループを使用してリソースへのトラフィックを制御する」を参照してください。

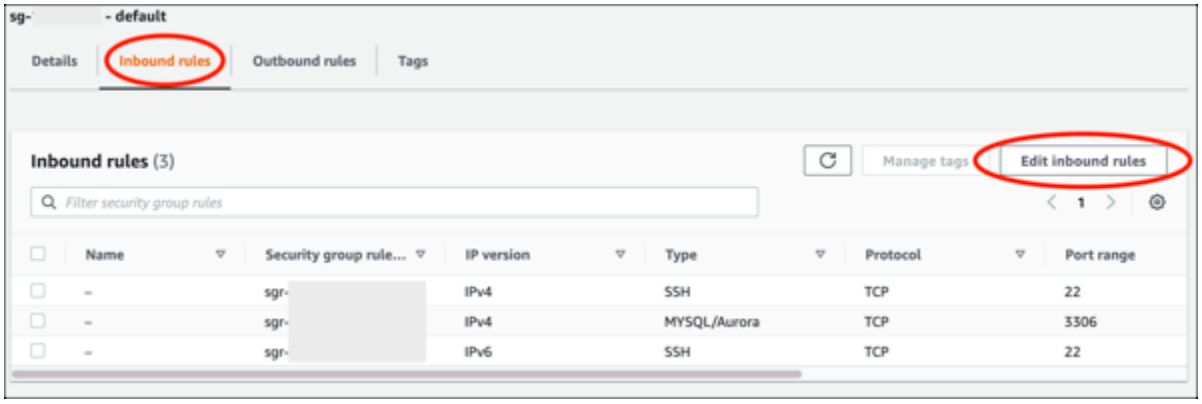
LAMP インスタンスが Aurora データベースへの接続を確立できるよう、以下の手順を完了してセキュリティグループを設定します。

1. Amazon RDS コンソールにサインインします。

2. ナビゲーションペインで、[Databases] (データベース) を選択します。
3. LAMP インスタンスが接続する Aurora データベースの[ライターインスタンス]を選択します。
4. [Connectivity & security (接続とセキュリティ)] タブを選択します。
5. [Endpoint & port] (エンドポイントとポート) セクションに表示されるライターインスタンスのエンドポイント名とポートを記録します。これらの情報は、データベースに接続する Lightsail インスタンスを設定するときに必要になります。
6. [Security] (セキュリティ) セクションでアクティブな VPC セキュリティグループのリンクを選択します。データベースのセキュリティグループにリダイレクトされます。

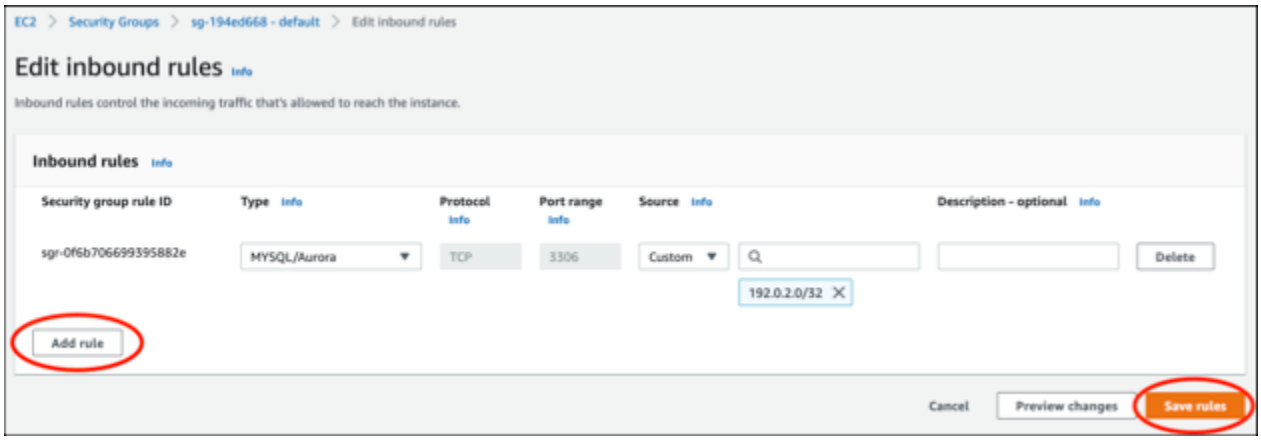


7. Aurora データベースのセキュリティグループが選択されていることを確認します。
8. [Inbound rules] (インバウンドルール) タブを開きます。
9. インバウンドのルールを編集 を選択します。



10. [Edit inbound rules] (インバウンドルールの編集) ページで [Add rule] (ルールの追加) を選択します。
11. 次のいずれかのステップを完了します。

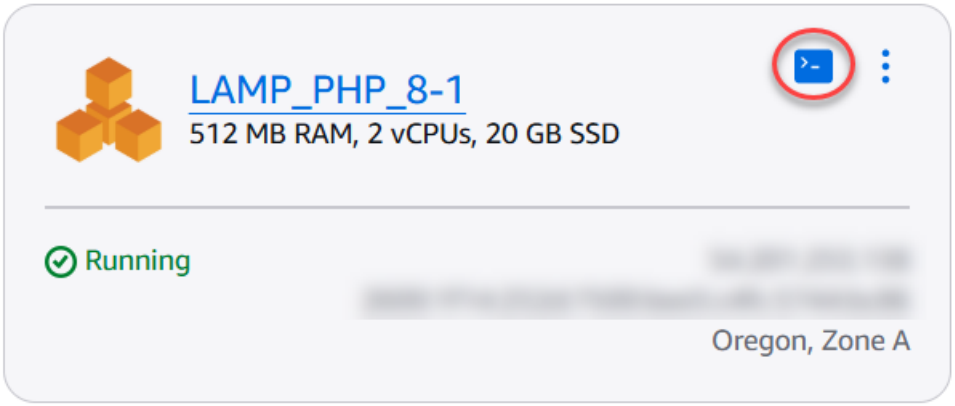
    - デフォルトの MySQL ポート 3306 を使用する場合は、[Type] (タイプ) ドロップダウンメニューから [MySQL/Aurora] を選択します。
    - データベースのカスタムポートを使用する場合は、[Type] (タイプ) ドロップダウンメニューから [Custom TCP] (カスタム TCP) を選択し、[Port Range] (ポート範囲) テキストボックスにポート番号を入力します。
12. [Source] (ソース) テキストボックスに LAMP インスタンスのプライベート IP アドレスを追加します。IP アドレスは、 CIDR 表記で入力する必要があります (/32 を追加する必要があります)。例えば、`192.0.2.0` を許可するには「`192.0.2.0/32`」と入力します。
13. [ルールの保存] を選択します。



ステップ 3: Lightsail インスタンスから Aurora データベースに接続する

以下の手順を完了して、Lightsail インスタンスから Aurora データベースに接続できることを確認します。

1. Lightsail コンソール にサインインします。
2. 左のナビゲーションペインの [インスタンス] を選択してください。
3. SSH を使用して接続する LAMP インスタンスのブラウザベースの SSH クライアントアイコンを選択します。



4. インスタンスに接続したら、次のコマンドを入力して、Aurora データベースに接続します。このコマンドでは、*DatabaseEndpoint* を実際の Aurora データベースのエンドポイントアドレスで置き換え、*Port* をデータベースのポートで置き換えます。*MyUsername* は、データベースを作成したときに入力したユーザーの名前で置き換えます。

```
mysql -h DatabaseEndpoint -P Port -u MyUserName -p
```

インスタンスが Aurora データベースにアクセスおよび接続できれば、次の例のような応答が表示されます。

```
bitnami@ip-            $ mysql -h database.cluster-            .us-west-2.rds.amazonaws.
com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 215
Server version: 5.6.10 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

この応答が表示されない場合、またはエラーメッセージが表示された場合は、データベースのセキュリティグループを設定して、Lightsail インスタンスのプライベート IP アドレスからの接続を許可する必要があることがあります。詳細については、このガイドの「Aurora データベースのセキュリティグループを設定する」を参照してください。

ステップ 4: MariaDB データベースを LAMP インスタンスから Aurora データベースに転送する

インスタンスからデータベースに接続できることを確認した後は、データを LAMP インスタンスデータベースから Aurora データベースに移行する必要があります。詳細については、「Aurora の Amazon Aurora ユーザーガイド」の「Amazon Aurora MySQL DB クラスターのモニタリングデータ」を参照してください。

ステップ 5: Aurora マネージドデータベースに接続するようアプリケーションを設定する

アプリケーションデータを Aurora データベースに転送した後、LAMP インスタンス上で実行するアプリケーションを設定して Aurora データベースに接続します。SSH を使用して LAMP インスタンスに接続し、アプリケーションのデータベース設定ファイルにアクセスします。設定ファイルで、Aurora データベースのエンドポイントアドレス、データベースユーザー名、およびパスワードを定義します。設定ファイルの例を以下に示します。

```
bitnami@ip-            :~/htdocs$ cat connectvalues.php
<?php
$host           = 'database.cluster-           .us-west-2.rds.amazonaws.com';
$username       = 'admin';
$password       = 'Password1';
```

Lightsail の Windows Server 2016 インスタンスを起動して設定する

仮想プライベートサーバーだけが必要な場合、Amazon Lightsail は Amazon Web Services (AWS) を始める最も簡単な方法です。Lightsail には、プロジェクトを迅速に立ち上げるために必要なものがすべて揃っています。予測可能な低料金で仮想マシン、SSD ベースのストレージ、データ転送、DNS 管理、静的 IP が提供されます。

このチュートリアルでは、Lightsail で Windows Server 2016 インスタンスを起動して設定する方法を示します。RDP 経由でのインスタンスへの接続、静的 IP の作成とインスタンスへのアタッチ、DNS ゾーンの作成とドメインのマッピングに関するステップが含まれています。このチュートリアルを完了すると、Lightsail でインスタンスを起動して使用するための基礎固めができます。

目次

- ステップ 1: AWS にサインアップ

- ステップ 2: Windows Server 2016 インスタンスを作成する
- ステップ 3: RDP 経由で Windows Server 2016 インスタンスに接続する
- ステップ 4: 静的 IP アドレスを作成して Windows Server 2016 インスタンスにアタッチする
- ステップ 5: DNS ゾーンを作成し、ドメインを Windows Server 2016 インスタンスにマッピングする
- 次のステップ

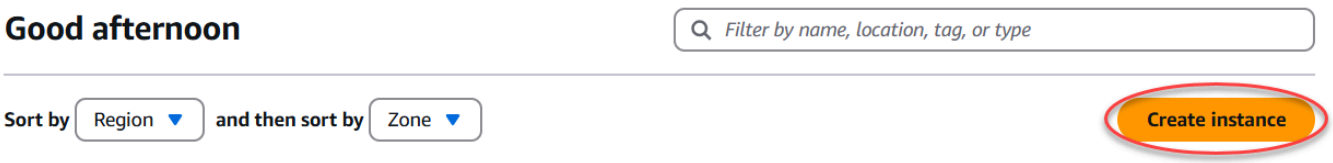
ステップ 1: AWS にサインアップ

このチュートリアルでは、 AWS アカウントが必要です。にサインアップ AWSするか、既に アカウントをお持ちの場合は にサインイン AWSします。

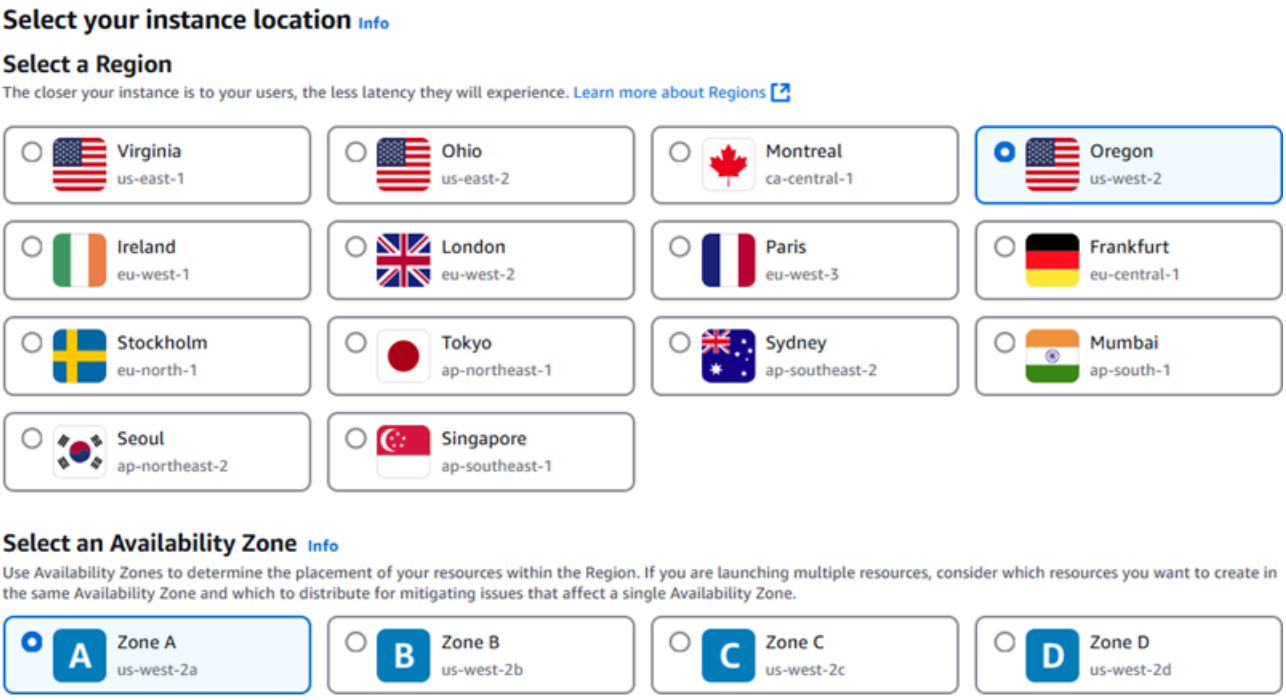
ステップ 2: Lightsail で Windows Server 2016 インスタンスを作成する

Lightsail で Windows Server 2016 インスタンスを起動します。詳細については、「Windows Server ベースのインスタンスの使用を開始する」を参照してください。

1. Lightsail コンソール にサインインします。
2. Lightsail ホーム画面の [インスタンス] セクションで、[インスタンスの作成] を選択します。



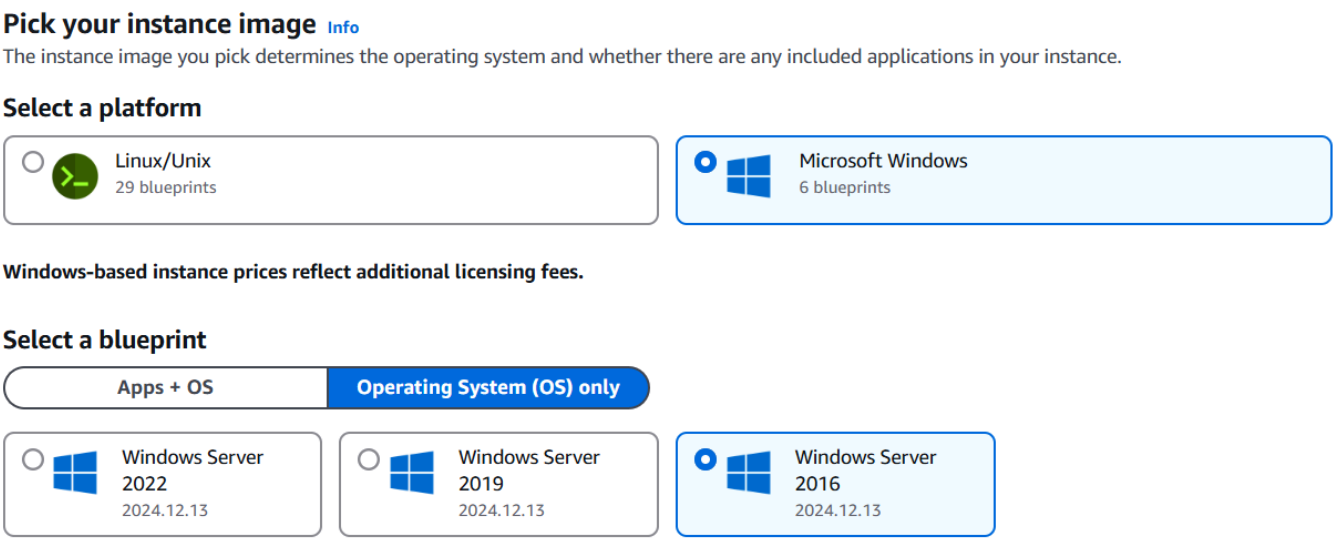
3. インスタンスの AWS リージョン とアベイラビリティーゾーンを選択します。



4. インスタンスイメージを選択します。

   a. プラットフォームとして [Microsoft Windows] を選択します。

   b. [OS のみ] を選択し、設計図として [Windows Server 2016] を選択します。



5. インスタンスプランを選択します。

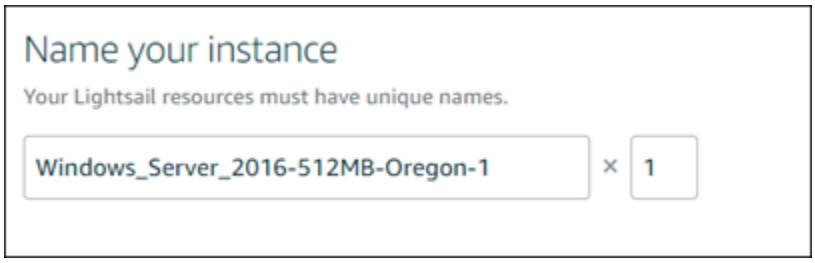
プランには、低額で予測可能なコスト、マシン設定 (RAM、SSD、vCPU)、およびデータ転送枠が含まれます。9.50 USD Lightsailプランを 1 か月間 (最大 750 時間) 無料で試すことができます。 は、アカウントに 1 か月の無料 AWS クレジットを付与します。

> **Note**
>
> AWS 無料利用枠の一部として、一部のインスタンスバンドルで Amazon Lightsailを無料で開始できます。詳細については、「Amazon Lightsail 料金表」ページの「AWS Free Tier」を参照してください。

6. インスタンスの名前を入力します。

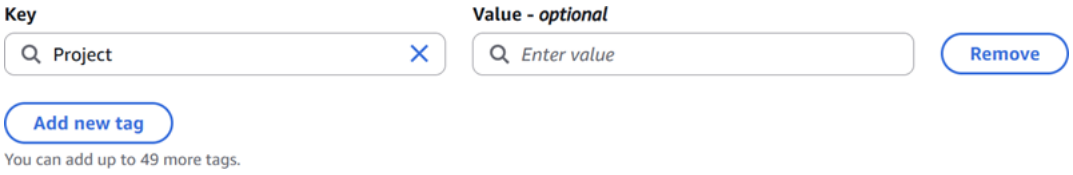
   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

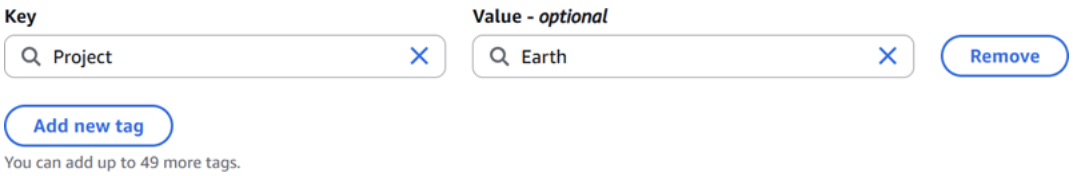


7. (オプション)[新しいタグの追加] を選択してインスタンスにタグを追加します。必要に応じてこの手順を繰り返して、他のタグを追加します。タグの使用の詳細については、「タグ」を参照してください。

   a. [キー] には、タグキーを入力します。



   b. (オプション) [値] にタグの値を入力します。

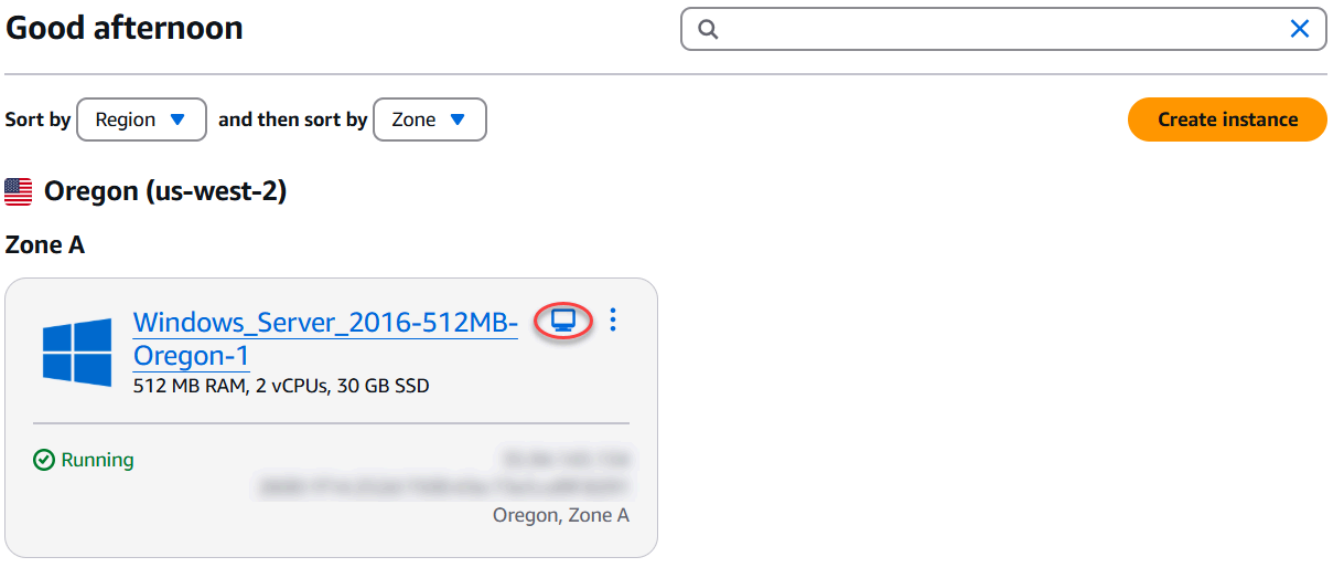


8. [インスタンスの作成] を選択します。

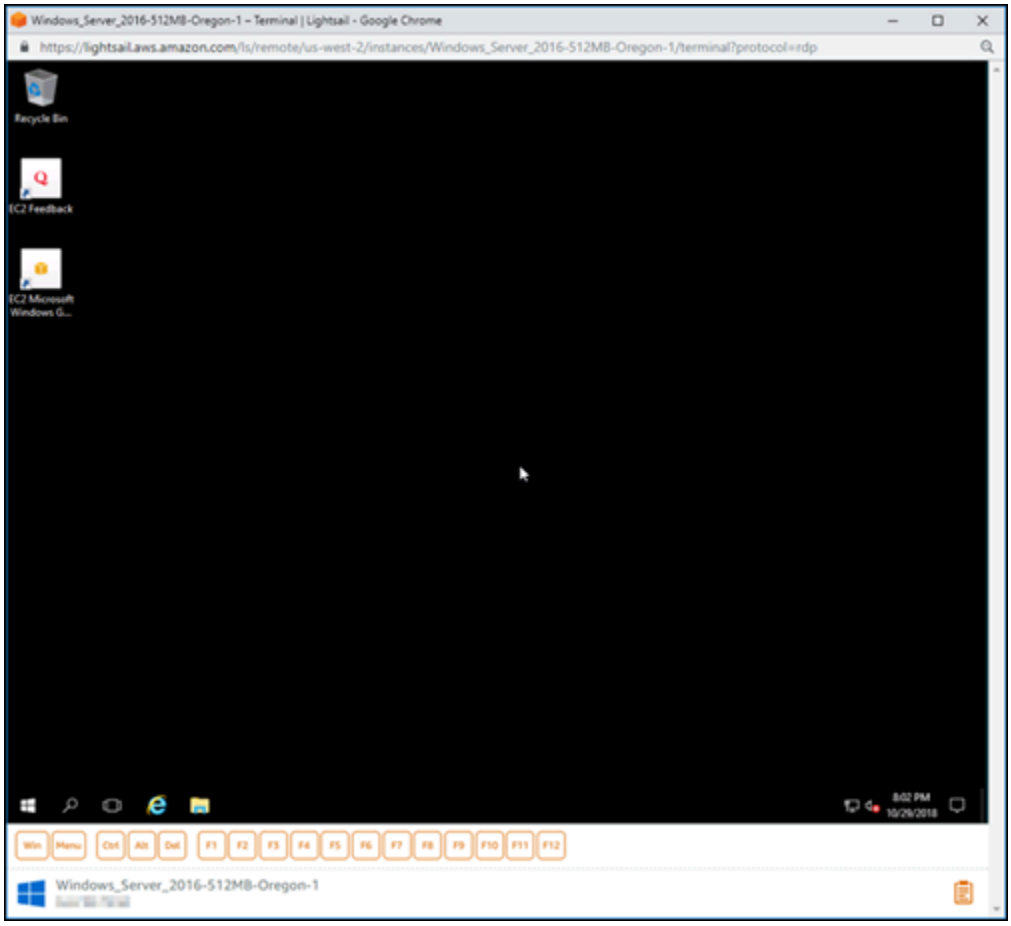
ステップ 3: RDP 経由で Windows Server 2016 インスタンスに接続する

Lightsail コンソールのブラウザベースの RDP クライアントを使用して Windows Server 2016 インスタンスに接続します。詳細については、「Windows インスタンスに接続する」を参照してください。

1. Lightsail ホームページの [インスタンス] セクションで、Windows Server 2016 インスタンスの [RDP クイック接続] アイコンを選択します。



2. ブラウザベースの RDP クライアントウィンドウが表示されたら、Windows Server 2016 インスタンスの設定を開始できます。

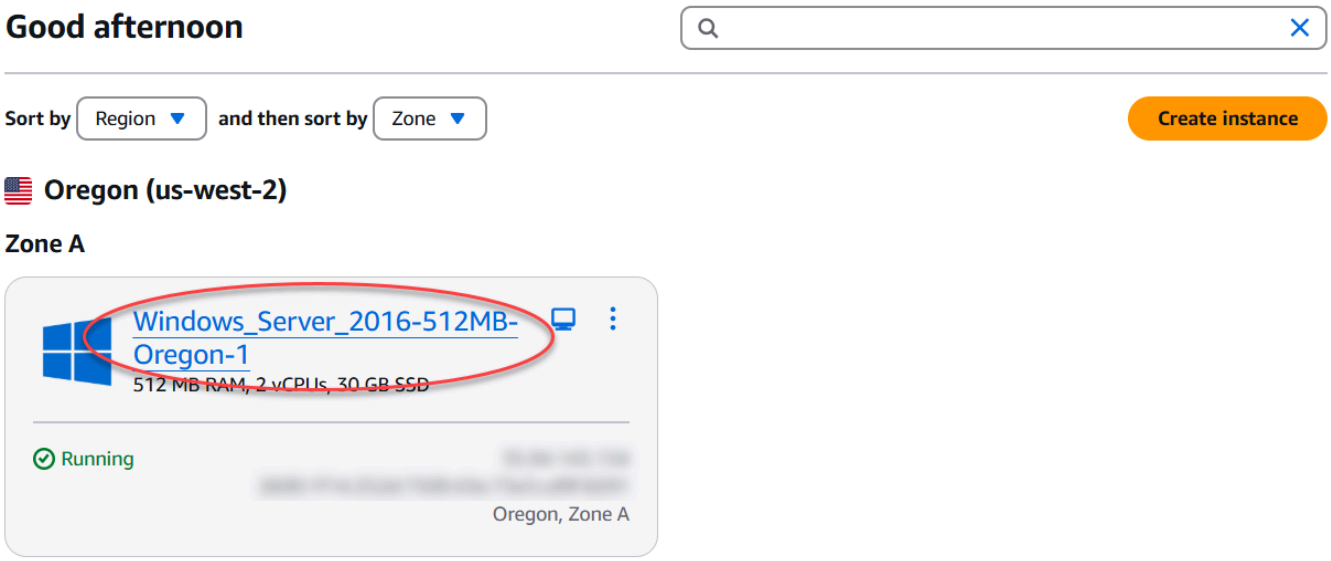


ステップ 4: 静的 IP アドレスを作成して Windows Server 2016 インスタンスにアタッチする

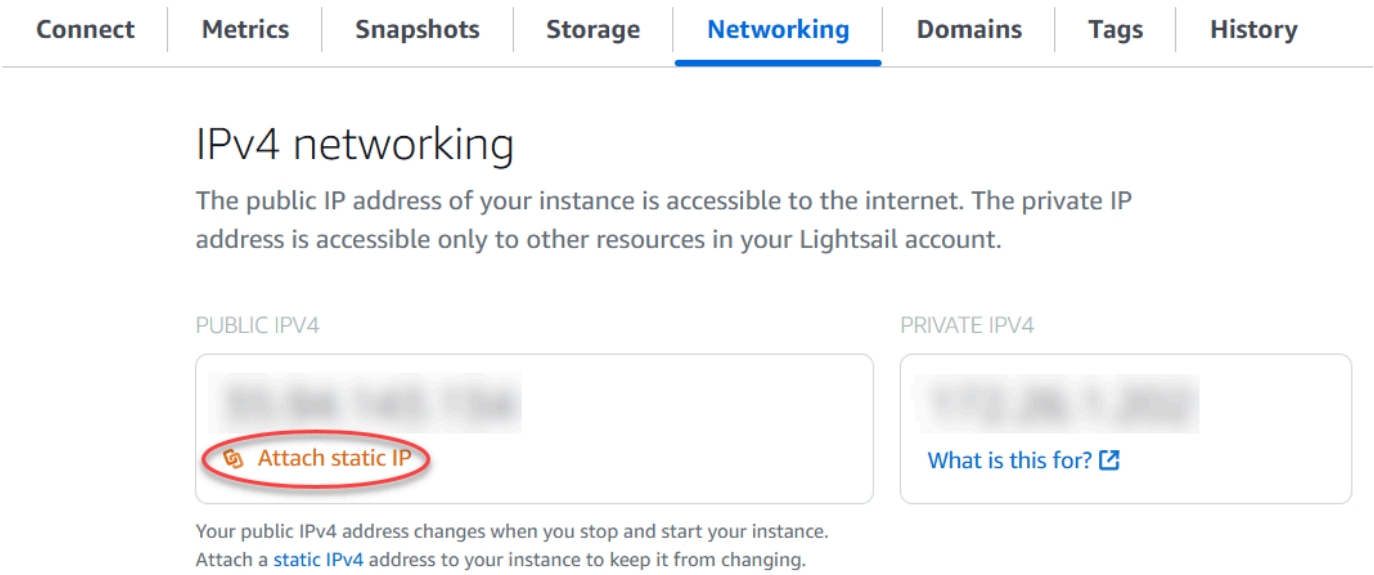
Windows Server 2016 インスタンスのデフォルトのパブリック IP は、インスタンスを停止して開始すると変わります。インスタンスにアタッチした静的 IP アドレスは、インスタンスを停止して開始しても変わりません。

静的 IP アドレスを作成して Windows Server 2016 インスタンスにアタッチします。詳細については、Lightsail ドキュメントの「静的 IP を作成してインスタンスにアタッチする」を参照してください。

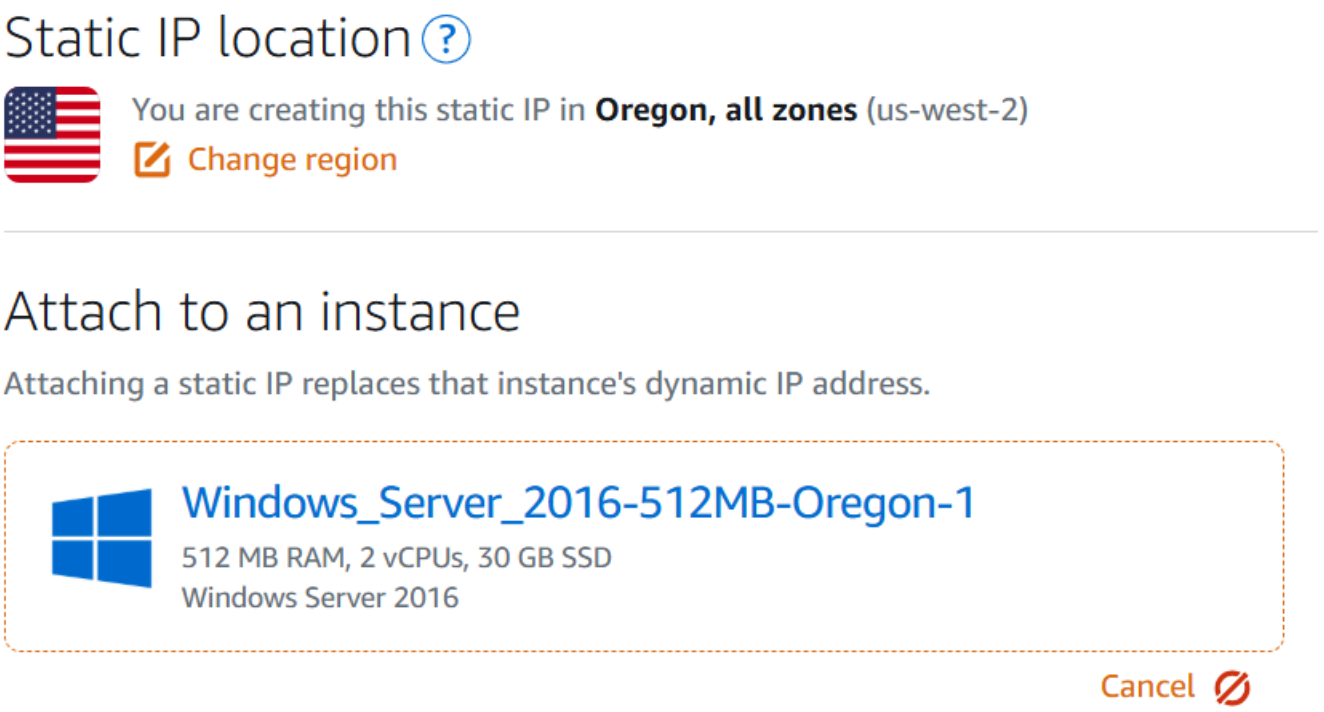
1. Lightsail ホームページの [インスタンス] セクションで、実行中の Windows Server 2016 インスタンスを選択します。



2. [ネットワーキング] タブ、[静的 IP の作成] の順に選択します。



3. このチュートリアルで前に選択したインスタンスに基づいて、静的 IP の場所とアタッチ済みインスタンスが事前に選択されます。



4. 静的 IP の名前を入力します。

   リソース名:

   - Lightsail アカウント内の各 AWS リージョン 内で一意である必要があります。
   - 2〜255 文字を使用する必要があります。
   - 先頭と末尾は英数字または数字を使用する必要があります。
   - 英数字、数字、ピリオド、ダッシュ、アンダースコアを使用することができます。

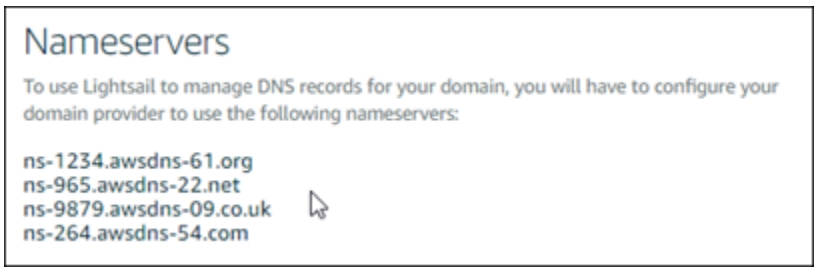
5. [作成] を選択します。

ステップ 5: DNS ゾーンを作成し、ドメインを Windows Server 2016 インスタンスにマッピングする

ドメインの DNS レコードの管理を Lightsail に引き渡します。これにより、ドメインを Windows Server 2016 インスタンスにマッピングしやすくなり、ウェブサイトのすべてのリソースを Lightsail コンソールで管理できます。詳細については、Lightsail ドキュメントの「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。

1. Lightsail ホームページの [ドメインと DNS] セクションで、[DNS ゾーンの作成] を選択します。
2. ドメインを入力し、[DNS ゾーンの作成] を選択します。
3. ページに表示されたネームサーバーのアドレスを書き留めておきます。

   ドメインの DNS レコードの管理を Lightsail に引き渡すには、これらのネームサーバーのアドレスをドメイン名のレジストラに追加します。



4. ドメインの DNS レコードの管理を Lightsail に引き渡したら、次に示すように、A レコードを追加してドメインの apex が LAMP インスタンスをポイントするように設定します。

   a. DNS ゾーンの [Assignments] (割り当て) タブで [Add assignment] (割り当てを追加) を選択します。
   b. [Select a domain] (ドメインの選択) フィールドで、ドメインまたはサブドメインを選択します。
   c. [Select a resource] (リソースの選択) ドロップダウンで、このチュートリアルで以前に作成した LAMP インスタンスを選択します。
   d. [Assign] (割り当て) を選択します。

   変更内容がインターネットの DNS を通じて伝播されるまで待ってから、LAMP インスタンスへのトラフィックのルーティングを開始します。

次の手順

Amazon Lightsail で Windows Server 2016 インスタンスを起動した後で、以下の追加のステップを実行できます。

- Windows Server インスタンスのスナップショットの作成
- Windows Server ベースの Lightsail インスタンスをセキュリティで保護するためのベストプラクティス
- ブロックストレージディスクを作成して Windows Server インスタンスにアタッチする
- Windows Server インスタンスのストレージ領域を拡張する

で Lightsail API アクティビティをモニタリングする AWS CloudTrail

Amazon Lightsail は、 のユーザー AWS CloudTrail、ロール、または のサービスによって実行されたアクションを記録する AWS サービスである と統合されていますLightsail。CloudTrail は、Lightsail のすべての API コールをイベントとしてキャプチャします。キャプチャされたコールには、Lightsail コンソールのコールと、Lightsail API オペレーションへのコードのコールが含まれます。証跡を作成する場合は、Lightsail のイベントなど、Amazon S3 バケットへの CloudTrail イベントの継続的デリバリーを有効にすることができます。証跡を設定しない場合でも、「CloudTrail」コンソールの「イベント履歴」で最新のイベントを表示できます。CloudTrail が収集した情報を使用して、Lightsail に

対して行われた要求、要求が行われた IP アドレス、要求を行った人、要求が行われた日時、および追加の詳細を判別できます。

CloudTrail の詳細については、「AWS CloudTrail ユーザーガイド」を参照してください。

LightsailCloudTrailでの情報

CloudTrail は、 AWS アカウントの作成時にアカウントで有効になります。でアクティビティが発生するとLightsail、そのアクティビティはイベント履歴の他の AWS サービスイベントとともに CloudTrail イベントに記録されます。 AWS アカウントで最近のイベントを表示、検索、ダウンロードできます。詳細については、CloudTrail イベント履歴でのイベントの表示を参照してください。

のイベントなど、 AWS アカウントのイベントの継続的な記録についてはLightsail、証跡を作成します。追跡により、CloudTrail はログファイルを Simple Storage Service (Amazon S3) バケットに配信できます。デフォルトでは、コンソールで作成した証跡がすべての AWS リージョンに適用されます。証跡は、 AWS パーティション内のすべてのリージョンからのイベントをログに記録し、指定した Amazon S3 バケットにログファイルを配信します。さらに、CloudTrail ログで収集したイベントデータをより詳細に分析し、それに基づいて対応するため、他の AWS サービスを構成できます。詳細については、次を参照してください:

- 証跡の作成のための概要
- CloudTrail がサポートするサービスと統合
- CloudTrail 用 Amazon SNS 通知の構成
- 複数のリージョンから CloudTrail ログファイルを受け取るおよび複数のアカウントから CloudTrail ログファイルを受け取る

すべての Lightsail アクションは CloudTrail によってログに記録され、「Amazon Lightsail API リファレンス」に記録されます。たとえば、[GetInstance]、[AttachStaticIp]、[RebootInstance] の各セクションの呼び出しにより、CloudTrail ログファイルにエントリが生成されます。

各イベントまたはログエントリには、誰がリクエストを生成したかという情報が含まれます。アイデンティティ情報は、以下を判別するのに役立ちます。

- リクエストがルートまたは AWS Identity and Access Management (IAM) ユーザー認証情報を使用して行われたかどうか。
- リクエストがロールまたはフェデレーションユーザーのテンポラリなセキュリティ認証情報を使用して行われたかどうか。

- リクエストが別の AWS サービスによって行われたかどうか。

詳細については、「CloudTrail userIdentity 要素」を参照してください。

Lightsail ログファイルエントリの概要

追跡は、指定したAmazon S3バケットにイベントをログファイルとして配信するように設定できます。CloudTrail のログファイルは、単一か複数のログエントリを含みます。イベントは、任意の出典からの単一のリクエストを表し、リクエストされたアクション、アクションの日時、リクエストパラメータなどに関する情報が含まれます。CloudTrail ログファイルは、パブリック API コールの順序付けられたスタックトレースではないため、特定の順序では表示されません。

HAR ファイルを作成して Lightsail 問題をトラブルシューティングする

Amazon Lightsail コンソールまたはLightsail仮想プライベートサーバー (VPS) で問題が発生した場合は サポート 、ウェブブラウザから HAR ファイルを送信するように求められることがあります。HAR ファイルには、一般的で診断が難しい問題のトラブルシューティングに役立つ重要な情報が含まれています。HAR ファイルでは、 サポート がこれらの問題を調査またはレプリケートすることもできます。

> **Important**
>
> HAR ファイルには、ユーザー名、パスワード、キーなどの機密情報が取り込まれることがあります。共有する前に、必ず HAR ファイルから機密情報を削除してください。

このガイドでは、ウェブブラウザから HAR ファイルを作成する方法を説明します。HTTP アーカイブ (HAR) ファイルとは、ブラウザによって記録された最新のネットワークアクティビティを含む JSON ファイルです。HAR ファイルを作成するには、次の手順に従います。

目次

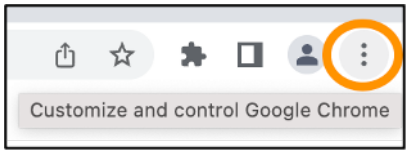
- ステップ 1: ブラウザで HAR ファイルを作成する
- ステップ 2: HAR ファイルを編集して機密情報を削除する
- ステップ 3: HAR ファイルをレビュー用に送信する

ステップ 1: ブラウザで HAR ファイルを作成する

> **Note**
>
> これらの手順の最新のテストは、Google Chrome バージョン 101.0.4951.64、Microsoft Edge (Chromium) バージョン 101.0.1210.47、および Mozilla Firefox バージョン 91.9 で行いました。これらのブラウザはサードパーティ製品であるため、これらの手順は最新バージョンまたは使用しているバージョンでの実際と一致しない場合があります。古い Microsoft Edge (EdgeHTML) や Apple Safari for macOS などの別のブラウザでは、HAR ファイルを生成するプロセスは似ているかもしれませんが、手順は異なります。

Google Chrome

1. ブラウザの右上にある [Customize and control Google Chrome] (Google Chrome の設定) を選択します。

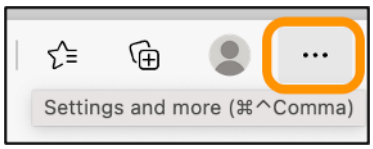


2. [More tools] (その他のツール) で、[Developer tools] (デベロッパーツール) を選択します。
3. ブラウザで DevTools を開いた状態で、[Network] (ネットワーク) パネルを選択します。
4. [Preserve log] (ログを保持) チェックボックスを選択します。
5. 現在のネットワークリクエストをすべてクリアするには、[Clear] (クリア) を選択します。
6. 直面している問題を再現します
7. DevTools で、ネットワークリクエストでコンテキスト (右クリック) メニューを開きます。
8. [Save all as HAR with content] (コンテンツと一緒に HAR としてすべて保存) を選択し、そのファイルを保存します。

詳しくは、Google Developers ウェブサイトの「Open Chrome DevTools」(Chrome DevTools を開く) と「Save all network requests to a HAR file」(すべてのネットワークリクエストを HAR ファイルに保存する) を参照してください。

Microsoft Edge (Chromium)

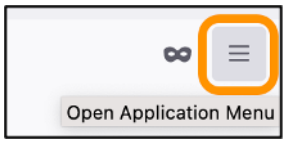
1. ブラウザの右上にある [設定など] を選択します。



2. [More tools] (その他のツール) で、[Developer tools] (デベロッパーツール) を選択します。
3. ブラウザで DevTools を開いた状態で、[Network] (ネットワーク) パネルを選択します。
4. [Preserve log] (ログを保持) チェックボックスを選択します。
5. 現在のネットワークリクエストをすべてクリアするには、[Clear] (クリア) を選択します。
6. 直面している問題を再現します
7. DevTools で、ネットワークリクエストでコンテキスト (右クリック) メニューを開きます。
8. [Save all as HAR with content] (コンテンツと一緒に HAR としてすべて保存) を選択し、そのファイルを保存します。

Mozilla Firefox

1. ブラウザの右上にある [Open Application Menu] (アプリケーションメニューを開きます) を選択します。



2. [More tools] (その他のツール) を選択し、[Web Developer tools] (ウェブ開発ツール) を選択します。
3. [Web Developer] (ウェブ開発) メニューで、[Network] (ネットワーク) を選択します。(Firefox の一部のバージョンでは、[Web Developer] (ウェブ開発) メニューは [Tools] (ツール) メニューの中にあります)。
4. 歯車アイコンを選択し、[Persist Logs] (永続ログ) を選択します。
5. ゴミ箱アイコン ([Clear] (消去)) を選択すると、現在のネットワークリクエストがすべてクリアされます。
6. 直面している問題を再現します。
7. [Network Monitor] (ネットワークモニター) で、リクエストリスト内のネットワークリクエストでコンテキストメニュー (右クリック) を開きます。
8. [Save All As HAR] (HAR 形式ですべて保存) を選択し、ファイルを保存します。

ステップ 2: HAR ファイルを編集して機密情報を削除する

1. テキストエディタアプリケーションで HAR ファイルを開きます。
2. テキストエディタの検索および置換ツールを使用して、HAR ファイルに取り込まれたすべての機密情報を特定して置換します。これには、ファイルの作成時にブラウザに入力したユーザー名、パスワード、およびキーが含まれます。
3. 編集した HAR ファイルを、機密情報を削除した状態で保存します。

ステップ 3: HAR ファイルをレビュー用に送信する

1. AWS Support Center Console の [サポートケースをオープンする] で、サポートケースを選択します。
2. サポートケースで、希望の連絡オプションを選択し、編集した HAR ファイルをアタッチして送信します。

Lightsail で Prometheus を使用してシステムリソースとアプリケーションをモニタリングする

Prometheus は、さまざまなシステムリソースとアプリケーションを管理するためのオープンソースの時系列監視ツールです。多次元データモデル、収集されたデータのクエリ機能、Grafana による詳細なレポート作成とデータの視覚化を提供します。

デフォルトでは、Prometheus はインストール先のサーバーでメトリクスを収集できるようになっています。ノードエクスポータを使用すると、ウェブサーバー、コンテナ、データベース、カスタムアプリケーション、その他のサードパーティシステムなどの他のリソースからメトリクスを収集できます。このチュートリアルでは、Lightsail インスタンスでノードエクスポーターを使用して Prometheus をインストールして設定する方法を説明します。利用可能なエクスポーターの全リストについては、「Prometheus ドキュメント」の「エクスポーターと統合Node Exporter のデフォルトのマシンメトリックを無効にします」を参照してください。

目次

- ステップ 1: 前提条件を満たす
- ステップ 2: Lightsail インスタンスにユーザーとローカルシステムディレクトリを追加する
- ステップ 3: Prometheus バイナリパッケージをダウンロードする

- ステップ 4: Prometheus を設定する
- ステップ 5: Prometheus をスタートする
- ステップ 6: Node Exporter をスタートする
- ステップ 7: Node Exporter データコレクタで Prometheus を設定する

ステップ 1: 前提条件を満たす

Prometheus を Amazon Lightsail インスタンスにインストールするには、次の作業を行う必要があります。

- Lightsail のインスタンスを作成します。インスタンスには Ubuntu 20.04 LTS ブループリントを使用することをお勧めします。詳細については、「Amazon Lightsail でインスタンスを作成する」を参照してください。
- 静的 IP アドレスを作成して新規インスタンスにアタッチします。詳細については、「Amazon Lightsail で静的 IP アドレスを作成する」を参照してください。
- 新しいインスタンスのファイアウォールのポート 9090 と 9100 を開きます。Prometheus では、ポート 9090 と 9100 が開いている必要があります。詳細については、「Amazon Lightsail でインスタンスファイアウォールルールの追加および編集」を参照してください。

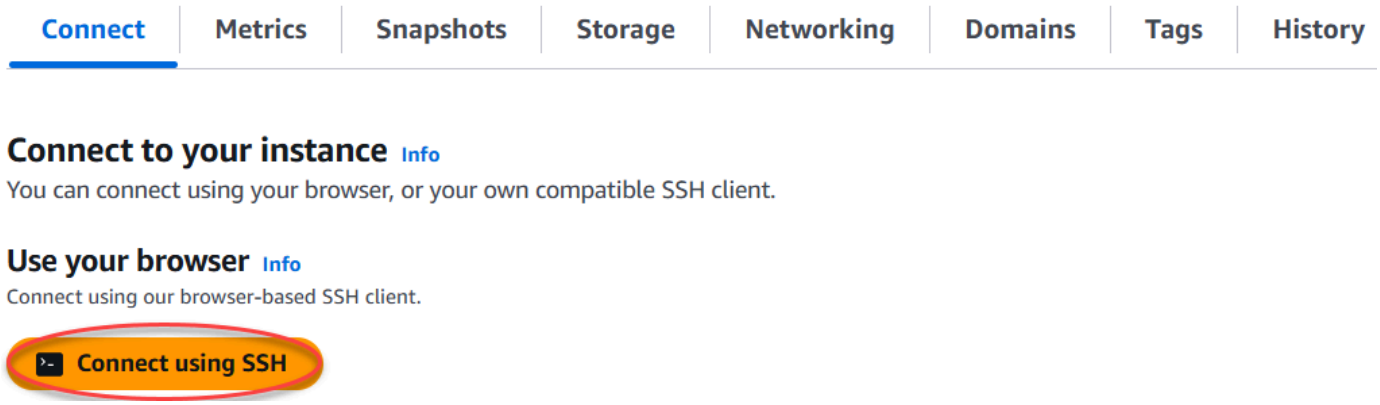
ステップ 2: Lightsail インスタンスにユーザーとローカルシステムディレクトリを追加する

以下の手順を実行して Lightsail インスタンスに接続し SSH を使用してユーザーとシステムディレクトリを追加します。この手順では、次の Linux ユーザーアカウントを作成します。

- `prometheus` – このアカウントは、サーバー環境のインストールと構成に使用されます。
- `exporter` – このアカウントは、`node_exporter` 拡張の構成に使用されます。

これらのユーザーアカウントは管理のみを目的として作成されるため、この設定の範囲を超える追加のユーザーサービスや権限は必要ありません。この手順では、Prometheus がリソースを監視するために使用するファイル、サービス設定、およびデータを保存および管理するためのディレクトリも作成します。

1. Lightsail コンソール にサインインします。
2. インスタンス管理ページの [接続] タブで、[SSH を使用して接続] を選択します。



3. 接続後に、次のコマンドを 1 つずつ入力して、2 つの Linux ユーザーアカウント (`prometheus` および `exporter`) を作成します。

```
sudo useradd --no-create-home --shell /bin/false prometheus
```

```
sudo useradd --no-create-home --shell /bin/false exporter
```

4. 次のコマンドを 1 つずつ入力して、ローカルシステムディレクトリを作成します。

```
sudo mkdir /etc/prometheus /var/lib/prometheus
```

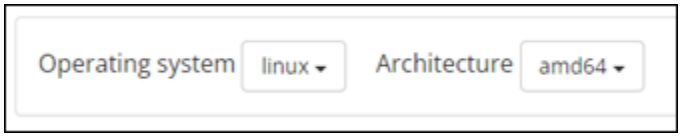
```
sudo chown prometheus:prometheus /etc/prometheus
```

```
sudo chown prometheus:prometheus /var/lib/prometheus
```

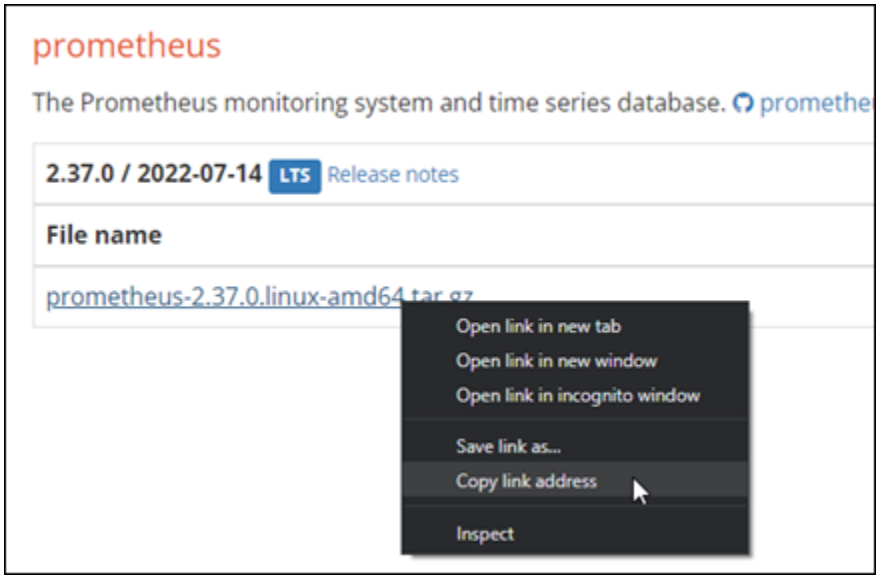
ステップ 3: Prometheus バイナリパッケージをダウンロードする

以下の手順を実行して、Prometheus バイナリパッケージを Lightsail インスタンスにダウンロードします。

1. ローカルコンピュータでウェブブラウザを開き、Prometheus のダウンロードページに移動します。
2. ページの上部で、[Operating system] (オペレーティングシステム) ドロップダウンから [Linux] を選択します。[Architecture] (アーキテクチャ) で [amd64] を選択します。



3. 表示される [Prometheus] ダウンロードリンクを選択または右クリックし、リンクアドレスをコンピュータ上のテキストファイルにコピーします。表示される [node_exporter] ダウンロードリンクにも同じ操作を行います。この手順の後半で、コピーした両方のアドレスを使用します。



4. SSH を使用して Lightsail インスタンスに接続します。
5. 次のコマンドを入力して、ディレクトリをホームディレクトリに変更します。

```
cd ~
```

6. 以下のコマンドを入力して、Prometheus バイナリパッケージをインスタンスにダウンロードします。

```
curl -LO prometheus-download-address
```

prometheus-download-address を、この手順で先ほどでコピーしたアドレスに置き換えます。アドレスの追加時は、コマンドは次の例のようになります。

```
curl -LO https://github.com/prometheus/prometheus/releases/download/v2.37.0/
prometheus-2.37.0.linux-amd64.tar.gz
```

7. 以下のコマンドを入力して、node_exporter バイナリパッケージをインスタンスにダウンロードします。

```
curl -LO node_exporter-download-address
```

node_exporter-download-address を、この手順の前のステップでコピーしたアドレスに置き換えます。アドレスの追加時は、コマンドは次の例のようになります。

```
curl -LO https://github.com/prometheus/node_exporter/releases/download/v1.3.1/
node_exporter-1.3.1.linux-amd64.tar.gz
```

8. 次のコマンドを 1 つずつ実行して、ダウンロードされた Prometheus と Node Exporter ファイルの内容を抽出します。

```
tar -xvf prometheus-2.37.0.linux-amd64.tar.gz
```

```
tar -xvf node_exporter-1.3.1.linux-amd64.tar.gz
```

ダウンロードしたファイルの内容が抽出された後、いくつかのサブディレクトリが作成されます。

9. 次のコマンドを 1 つずつ入力して、prometheus および promtool の抽出されたファイルを /usr/local/bin プログラムのディレクトリにコピーします。

```
sudo cp -p ./prometheus-2.37.0.linux-amd64/prometheus /usr/local/bin
```

```
sudo cp -p ./prometheus-2.37.0.linux-amd64/promtool /usr/local/bin
```

10. 次のコマンドを入力して、prometheus および promtool のファイルをこのチュートリアルの前半で作成した prometheus ユーザーに変更します。

```
sudo chown prometheus:prometheus /usr/local/bin/prom*
```

11. 次のコマンドを 1 つずつ入力して、consoles と console_libraries のサブディレクトリを /etc/prometheus にコピーします。-r オプションは階層内のすべてのディレクトリを再帰的にコピーします。

```
sudo cp -r ./prometheus-2.37.0.linux-amd64/consoles /etc/prometheus
```

```
sudo cp -r ./prometheus-2.37.0.linux-amd64/console_libraries /etc/prometheus
```

12. 次のコマンドを 1 つずつ入力して、コピーしたファイルの所有権をこのチュートリアルの前半で作成した prometheus ユーザーに変更します。-R オプションは階層内のすべてのファイルおよびディレクトリの所有権を再帰的に変更します。

```
sudo chown -R prometheus:prometheus /etc/prometheus/consoles
```

```
sudo chown -R prometheus:prometheus /etc/prometheus/console_libraries
```

13. 次のコマンドを 1 つずつ入力して、設定ファイル prometheus.yml を /etc/prometheus ディレクトリにコピーし、コピーしたファイルの所有権をこのチュートリアルの前半で作成した prometheus ユーザーに変更します。

```
sudo cp -p ./prometheus-2.37.0.linux-amd64/prometheus.yml /etc/prometheus
```

```
sudo chown prometheus:prometheus /etc/prometheus/prometheus.yml
```

14. 以下のコマンドを入力して、./node_exporter* サブディレクトリから /usr/local/bin プログラムのディレクトリに node_exporter ファイルをコピーします。

```
sudo cp -p ./node_exporter-1.3.1.linux-amd64/node_exporter /usr/local/bin
```

15. 次のコマンドを入力して、このチュートリアルの前半で作成した exporter ユーザーにファイルの所有権を変更します。

```
sudo chown exporter:exporter /usr/local/bin/node_exporter
```

ステップ 4: Prometheus を設定する

Prometheus を設定するには、次の手順を実行します。この手順では、prometheus.yml ファイルを開いて編集します。このファイルには、Prometheus ツールのさまざまな設定が含まれています。Prometheus は、ファイルに設定した設定に基づいて監視環境を確立します。

1. SSH を使用して Lightsail インスタンスに接続します。
2. prometheus.yml ファイルを開いて編集する前に、次のコマンドを入力してこのファイルのバックアップコピーを作成します。

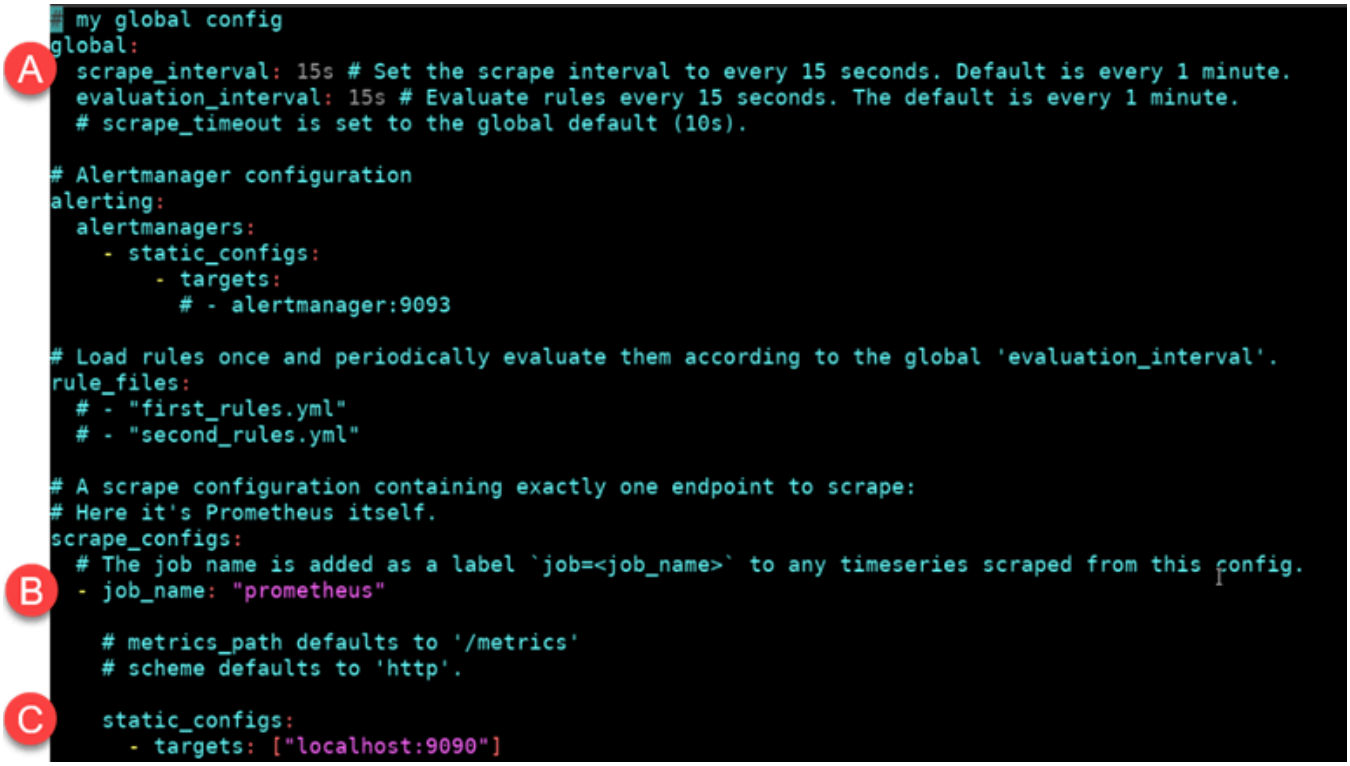
```
sudo cp /etc/prometheus/prometheus.yml /etc/prometheus/prometheus.yml.backup
```

3. 次のコマンドを入力して、 Vim を使用し、prometheus.yml ファイルを開きます。

```
sudo vim /etc/prometheus/prometheus.yml
```

以下に、prometheus.yml ファイルに設定する必要があるいくつかの重要なパラメータを示します。

- scrape_interval — このパラメータは、global ヘッダーの下に置かれ、Prometheus が特定のターゲットのメトリックスデータをどの頻度で収集するか、またはスクレイプするかの時間間隔 (秒) を定義します。global タグで示されているように、この設定は Prometheus が監視するすべてのリソースに共通です。この設定は、個々のエクスポーターがグローバル値をオーバーライドする別の値を提供しない限り、エクスポーターにも適用されます。このパラメータは、現在の 15 秒に設定したままにしておくことができます。
- job_name — scrape_configs ヘッダー下に配置されるこのパラメータは、データクエリまたはビジュアルディスプレイの結果セット内のエクスポーターを識別するラベルです。ジョブ名の値は、環境内で監視されているリソースを最もよく反映するように指定できます。たとえば、ウェブサイトを管理するジョブに business-web-app というラベルを付けたり、データベースに mysql-db-1 というラベルを付けることができます。この初期設定では、Prometheus サーバーのみを監視しているので、最新の prometheus 値を保つことができます。
- targets — static_configs ヘッダー下に配置される targets 設定では、特定のエクスポーターが実行されている場所を識別するために ip_addr:port キーバリューのペアを使用します。この手順のステップ 4 ~ 7 で、デフォルト設定を変更できます。



> **Note**
>
> この初期セットアップでは、`alerting` および `rule_files` のパラメータを設定する必要はありません。

4. Vim で開いている `prometheus.yml` ファイルでは、[I] キーを押して Vim 挿入モードに移ります。
5. `static_configs` ヘッダーの下に置かれている `targets` パラメータをスクロールして見つけます。
6. デフォルト設定を *<ip_addr>*:9090 に変更します。インスタンスの静的 IP アドレスを *<ip_addr>* に置き換えます。変更されたパラメータは、次の例のようになります。

```
static_configs:
  - targets: ["192.0.2.0:9090"]
```

7. [Esc] キーを押して挿入モードを終了し、[:wq!] と入力して変更内容を保存して Vim を終了します。
8. (オプション) 何か問題が発生した場合は、次のコマンドを入力してこの手順で前に作成したバックアップと `prometheus.yml` を置き換えます。

```
sudo cp /etc/prometheus/prometheus.yml.backup /etc/prometheus/prometheus.yml
```

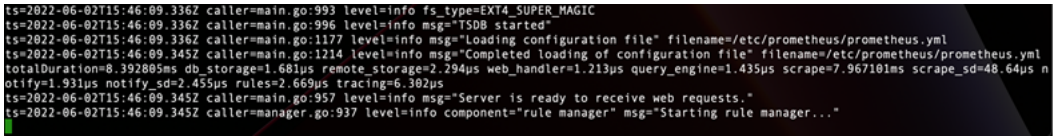
ステップ 5: Prometheus をスタートする

インスタンスで Prometheus サービスを開始するには、次のステップを実行します。

1. SSH を使用して Lightsail インスタンスに接続します。
2. 次のコマンドを入力して Prometheus サービスを開始します。

   ```
   sudo -u prometheus /usr/local/bin/prometheus --config.file /etc/prometheus/
   prometheus.yml --storage.tsdb.path /var/lib/prometheus --web.console.templates=/
   etc/prometheus/consoles --web.console.libraries=/etc/prometheus/console_libraries
   ```

   コマンドラインは、起動プロセスやその他のサービスの詳細を出力します。また、サービスがポート 9090 でリッスンしていることも示しているはずです。

   サービスが起動しない場合、このポートでのトラフィックを許可するインスタンスファイアウォールルールの作成については、このチュートリアルの「ステップ 1: 前提条件を満たす」セクションを参照してください。その他のエラーについては、prometheus.yml ファイルを見直して構文エラーがないことを確認します。
3. 実行中のサービスが検証されたら、[Ctrl+C] を押してストップします。
4. 以下のコマンドを入力し、Vim を使用して systemd 設定ファイルを開きます。このファイルは Prometheus を起動するために使用されます。

   ```
   sudo vim /etc/systemd/system/prometheus.service
   ```
5. ファイルに以下の行を挿入します。

   ```
   [Unit]
   Description=PromServer
   Wants=network-online.target
   After=network-online.target

   [Service]
   ```

```
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
--config.file /etc/prometheus/prometheus.yml \
--storage.tsdb.path /var/lib/prometheus/ \
--web.console.templates=/etc/prometheus/consoles \
--web.console.libraries=/etc/prometheus/console_libraries

[Install]
WantedBy=multi-user.target
```

前述の手順は Linux `systemd` サービスマネージャーがサーバー上で Prometheus を起動するために使用されます。Prometheus は、呼び出されると `prometheus` ユーザーとして実行され `prometheus.yml` ファイルを参照して、設定を読み込み `/var/lib/prometheus` ディレクトリの時系列データを保存するします。コマンドラインから `man systemd` を実行し、サービスの詳細情報を確認できます。

6. [Esc] キーを押して挿入モードを終了し、[:wq!] と入力して変更内容を保存して Vim を終了します。
7. 次のコマンドを入力して、`systemd` サービスマネージャに情報を読み込みます。

```
sudo systemctl daemon-reload
```

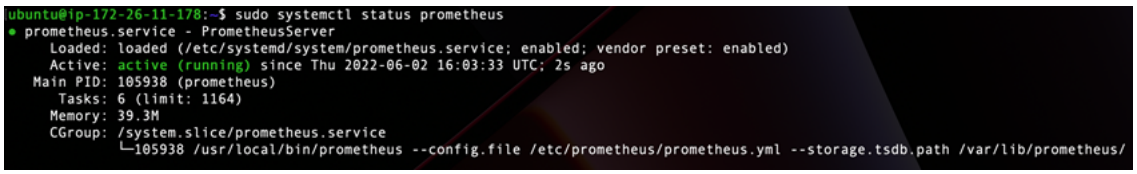
8. 次のコマンドを入力して Prometheus を再起動します。

```
sudo systemctl start prometheus
```

9. Prometheus サービスの状態を確認するには、次のコマンドを入力します。

```
sudo systemctl status prometheus
```

サービスが正常に起動された場合は、次の例のような出力が表示されます。



10. [Q] を押して、ステータスコマンドを終了します。

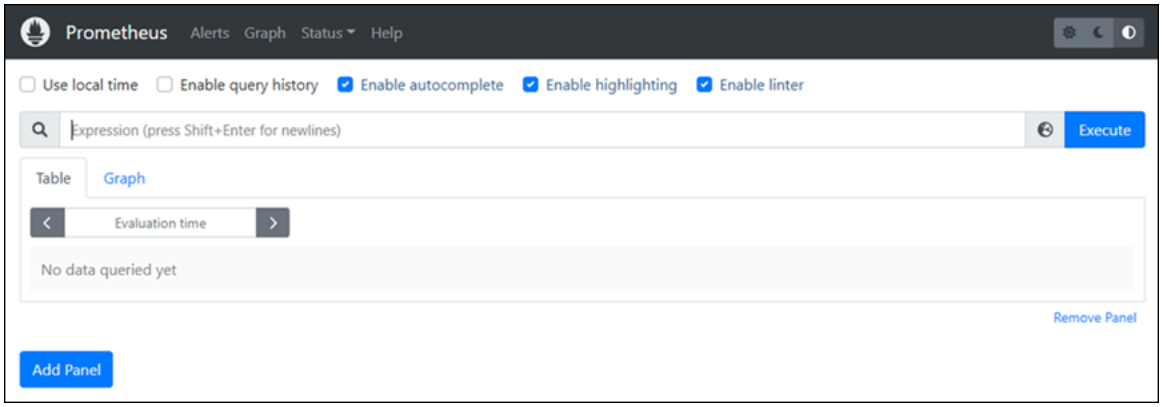
11. 次のコマンドを入力して、インスタンスが起動されたときに Prometheus が起動できるようにします。

```
sudo systemctl enable prometheus
```

12. ローカルコンピュータで Web ブラウザを開き、次の Web アドレスにアクセスして Prometheus 管理インターフェイスを表示します。

```
http:<ip_addr>:9090
```

<ip_addr> を Lightsail インスタンスの静的 IP アドレスに置き換えます。次の例に示すようなダッシュボードが表示されます。



ステップ 6: Node Exporter をスタートする

Node Exporter サービスを開始するには、以下の手順を実行します。

1. SSH を使用して Lightsail インスタンスに接続します。
2. 次のコマンドを入力し、Vim を使用して node_exporter の systemd サービスファイルを作成します。

```
sudo vim /etc/systemd/system/node_exporter.service
```

3. [I] キーを押して、Vim を挿入モードにします。
4. ファイルの末尾に次の行を追加します。これにより、CPU 負荷、ファイルシステムの使用状況、およびメモリリソースの監視コレクターを使用して node_exporter を設定します。

```
[Unit]
Description=NodeExporter
```

```
Wants=network-online.target
After=network-online.target

[Service]
User=exporter
Group=exporter
Type=simple
ExecStart=/usr/local/bin/node_exporter --collector.disable-defaults \
--collector.meminfo \
--collector.loadavg \
--collector.filesystem

[Install]
WantedBy=multi-user.target
```

> **Note**
>
> これらの手順では、Node Exporter のデフォルトのマシンメトリックを無効にします。Ubuntu で利用できるメトリクスの全リストについては、「Ubuntu ドキュメンテーション」の「Prometheus node_exporter のマニュアルページ」を参照してください。

5. [Esc] キーを押して挿入モードを終了し、[:wq!] と入力して変更内容を保存して Vim を終了します。
6. 次のコマンドを入力して、systemd プロセスをリロードします。

```
sudo systemctl daemon-reload
```

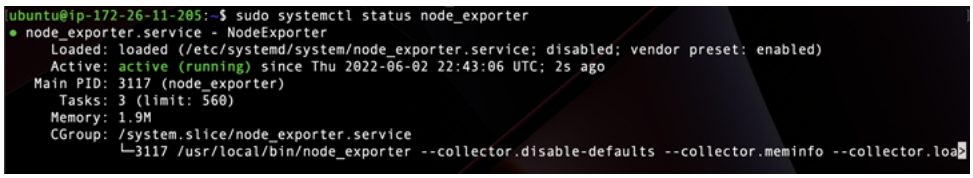
7. 次のコマンドを入力して node_exporter サービスを開始します。

```
sudo systemctl start node_exporter
```

8. node_exporter サービスの状態を確認するには、次のコマンドを入力します。

```
sudo systemctl status node_exporter
```

サービスが正常に起動された場合は、次の例のような出力が表示されます。



9. [Q] を押して、ステータスコマンドを終了します。
10. 次のコマンドを入力して、インスタンスが起動されたときに Node Exporter が起動できるようにします。

```
sudo systemctl enable node_exporter
```

ステップ 7: Node Exporter データコレクタで Prometheus を設定する

以下の手順を実行して、Node Exporter データコレクタで Prometheus を設定します。そのためには、prometheus.yml ファイルの node_exporter に新しい job_name パラメータを追加します。

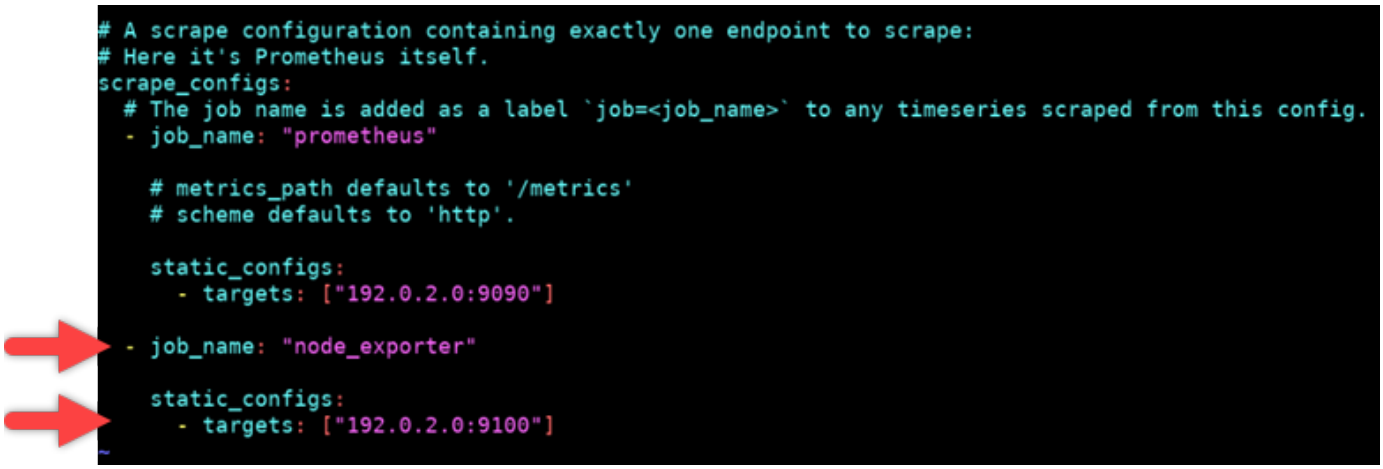
1. SSH を使用して Lightsail インスタンスに接続します。
2. 次のコマンドを入力して、 Vim を使用し、prometheus.yml ファイルを開きます。

```
sudo vim /etc/prometheus/prometheus.yml
```

3. [I] キーを押して、Vim を挿入モードにします。
4. 既存の - targets: ["*<ip_addr>*:9090"] パラメータの下で、次のテキスト行をファイルに追加します。

```
- job_name: "node_exporter"

static_configs:
- targets: ["<ip_addr>:9100"]
```

prometheus.yml ファイルの変更されたパラメータは、次の例のようになります。



次の点に注意してください。

- Node Exporter は、prometheus サーバーのポート 9100 をリッスンしてデータをスクレイピングします。このチュートリアルの「Step 1: Complete the prerequisites」(ステップ 1: 前提条件を満たす) セクションで説明されているように、インスタンスのファイアウォールルールを作成する手順に従っていることを確認します。
- prometheus job_name の設定と同様に、Lightsail インスタンスに添付される性的 IP アドレスと *<ip_addr>* を置き換えます。

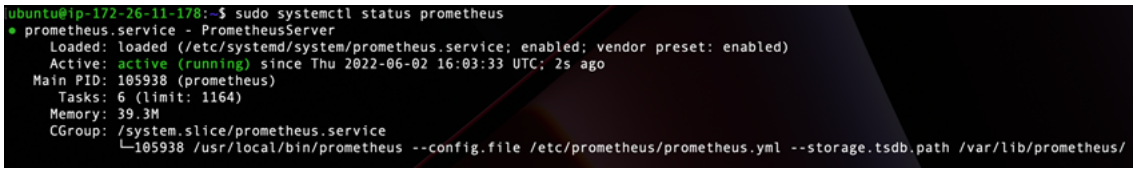
5. [Esc] キーを押して挿入モードを終了し、[:wq!] と入力して変更内容を保存して Vim を終了します。
6. 以下のコマンドを入力して Prometheus サービスを再起動し、設定ファイルへの変更を確定します。

```
sudo systemctl restart prometheus
```

7. Prometheus サービスの状態を確認するには、次のコマンドを入力します。

```
sudo systemctl status prometheus
```

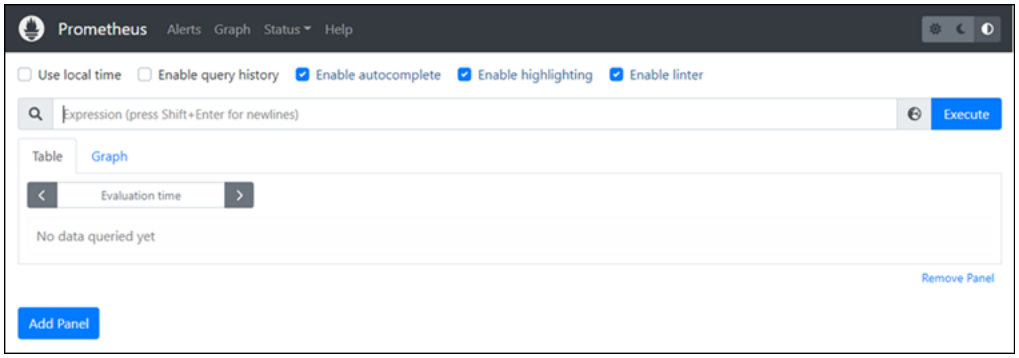
サービスが正常に再起動された場合は、次のような出力が表示されます。



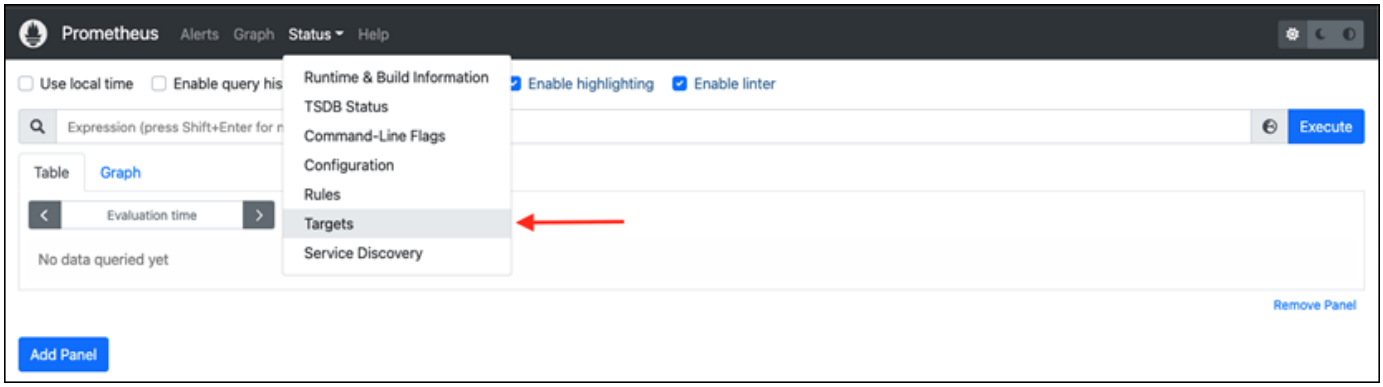
8. [Q] を押して、ステータスコマンドを終了します。
9. ローカルコンピュータで Web ブラウザを開き、次の Web アドレスにアクセスして Prometheus 管理インターフェイスを表示します。

```
http:<ip_addr>:9090
```

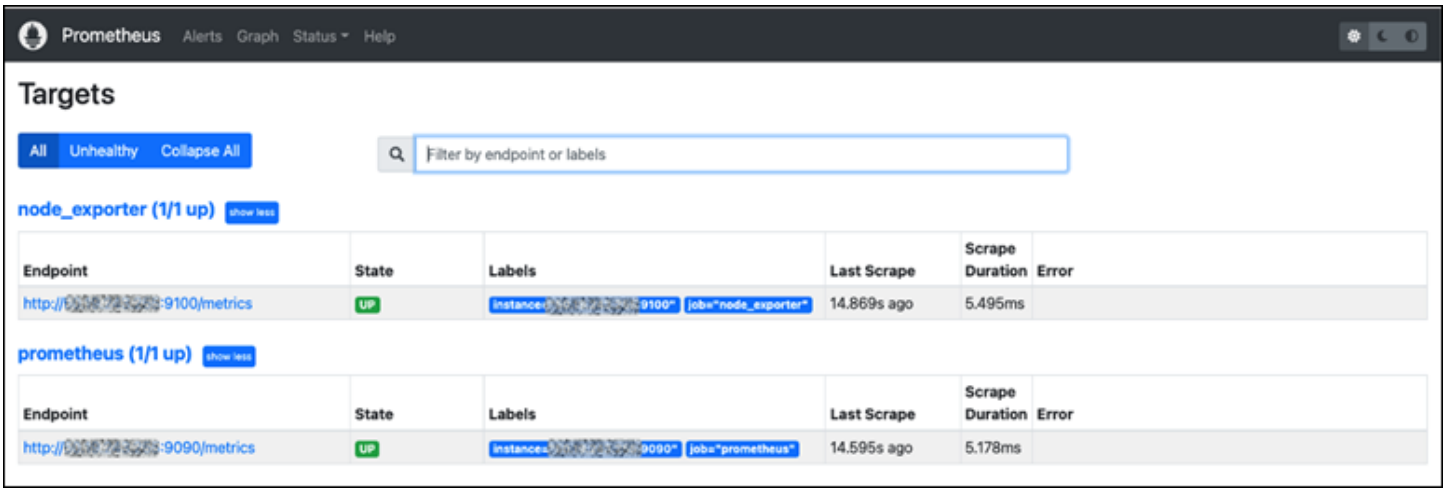
<ip_addr> を Lightsail インスタンスの静的 IP アドレスに置き換えます。次の例に示すようなダッシュボードが表示されます。



10. メインメニューで、[Status] (ステータス) ドロップダウンを選択し、[Targets] (ターゲット) を選択します。



次の画面には、2 つのターゲットが表示されます。最初のターゲットは [node_exporter] メトリクスコレクターのジョブで、2 つ目のターゲットは [Prometheus] ジョブです。



これで、メトリックの収集とサーバーの監視のための環境が適切に設定されました。

scp を使用して Lightsail の Linux インスタンス間でファイルを転送する

Linux のセキュアコピー (scp) コマンドを使用して、ローカルコンピュータから Linux または Unix インスタンスに、および Amazon Lightsail 内のあるインスタンスから別のインスタンスにファイルを転送します。scp コマンドの詳細については、「man7」ウェブサイトの「scp(1) — Linux manual」を参照してください。

このチュートリアルでは、ある Lightsail インスタンスから別のインスタンスにファイルをコピーする手順を説明します。

内容

- 前提条件
- ステップ 1: プライベートキー (.pem) ファイルをローカルコンピュータに保存する
- ステップ 2: プライベートキーのアクセス許可を変更する
- ステップ 3: プライベートキーをインスタンスに転送する
- ステップ 4: Lightsail Linux インスタンスと Unix インスタンス間でファイルを安全に転送する

前提条件

- 両方の Lightsail インスタンスのパブリック IP アドレスで 2 つのインスタンスを実行しています。インスタンスのパブリック IP アドレスを取得するには。「Lightsail コンソール」にサインインし、以下の例のように、インスタンスの横に表示されるパブリック IP アドレスをコピーします。
- SSH キーペアを使用して両方のインスタンスにアクセスできます。詳細については、「Linux インスタンスに接続する」を参照してください。

ステップ 1: プライベートキー (.pem) ファイルをローカルコンピュータに保存する

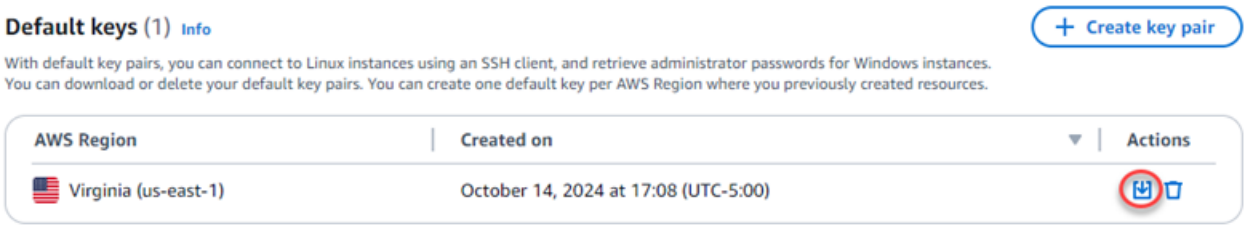
プライベートキー (.pem) ファイルをローカルコンピュータに保存するには、次の手順を実行します。ターゲットインスタンスのプライベートキーファイルは、あるインスタンスから別のインスタンスにファイルを安全に転送するために使用されます。同じ AWS リージョンのインスタンス間でファイルをコピーするには、そのリージョンのデフォルトキーを使用します。異なるリージョンのインスタンス間でファイルをコピーするには、ターゲットインスタンスがあるリージョンのデフォルトキー

を使用します。キーポリシーの詳細については、SSH およびインスタンスへの接続 を参照してください。

> **Note**
>
> 独自のキーペアを使用している場合、または Lightsail コンソールを使用してキーペアを作成した場合は、プライベートキーを見つけ、それを使用してインスタンスに接続します。独自のキーをアップロードする場合または Lightsail コンソールを使用してキーペアを作成する場合は Lightsail はプライベートキーを保存しません。プライベートキーなしで scp を使用してインスタンスに接続することはできません。

プライベートキー (.pem) をローカルコンピュータに保存するには

1. Lightsail コンソール にサインインします。
2. 上部のナビゲーションバーで [ユーザー名] を選択し、ドロップダウンメニューから [アカウント] を選択します。
3. [SSH キー] タブを選択します。
4. ページの [Default keys] (デフォルトキー) セクションまで下にスクロールします。
5. ファイルを転送したいインスタンスが配置されている AWS リージョン のデフォルトのプライベートキーの横にある [ダウンロード] を選択します。



6. ローカルドライブのセキュリティが確保された場所にプライベートキーを保存します。

   ユーザーのホームディレクトリにある「Keys」フォルダなど、SSH キーが保存されているディレクトリに、ダウンロードしたキーを移動することも可能です。このガイドの次のセクションで、プライベートキーが保存されるディレクトリを参照する必要があります。プライベートキーが .pem 以外の形式で保存しようとした場合、保存する前に手動で形式を .pem に変更する必要があります。

ステップ 2: プライベートキーのアクセス許可を変更する

次の手順では、プライベートキーファイルの権限を変更して、お客様以外のユーザーが読み書きできないように変更します。

プライベートキーのアクセス許可を変更するには

1. ローカルマシンでターミナルウィンドウを開きます。
2. 次のコマンドを入力して、キーペアのプライベートキーが本人にしか読み書きできないようにします。これは、一部のオペレーティングシステムで要求される、セキュリティのベストプラクティスです。

   ```
   sudo chmod 400 /path/to/private-key.pem
   ```

   コマンドで */path/to/private-key* を、インスタンスで使われるキーペアのプライベートキーが保存されている場所を向いたディレクトリパスに、置き換えます。

   例:

   ```
   sudo chmod 400 /Users/user/Keys/LightsailDefaultKey-us-west-2.pem
   ```

ステップ 3: プライベートキーをインスタンスに転送する

次の手順では、ローカルコンピュータから scp コマンドを実行して、プライベートキーをソースインスタンスに転送します。

scp を使用してプライベートキーをコンピュータからソースインスタンスに転送するには

1. コンピュータ上のプライベートキーファイルの場所と、インスタンス上の送信先パスを決定します。次の例では、プライベートキーファイルの名前は *private-key.pem*、ソースインスタンスのユーザー名は *ec2-user*、ソースインスタンスの IPv4 アドレスは *public-ipv4-address*、ソースインスタンスの IPv6 アドレスは *public-ipv6-address* です。*destination-path/* は、プライベートキーを転送するソースインスタンスの場所です。

   > **Note**
   >
   > インスタンスで使用されるブループリントに応じて、以下のいずれかのユーザー名を指定できます。

- AlmaLinux OS 9、Amazon Linux 2、Amazon Linux 2023、CentOS Stream 9、FreeBSD、および openSUSE インスタンスの場合: `ec2-user`
- Debian インスタンス: `admin`
- Ubuntu インスタンス: `ubuntu`
- Bitnami インスタンス: `bitnami`
- Plesk インスタンス: `ubuntu`
- cPanel & WHM インスタンス : `centos`

- (IPv4) インスタンスにプライベートファイルを転送するには、コンピュータから次のコマンドを入力します。

```
scp -i /path/private-key.pem /path/private-key.pem ec2-user@public-ipv4-address:path/
```

- (IPv6) インスタンスが IPv6 アドレスのみを使用する場合、インスタンスにプライベートファイルを転送するには、コンピュータから次のコマンドを入力します。IPv6 アドレスは、(\) でエスケープした角かっこ ([ ]) で囲む必要があります。

```
scp -i /path/private-key.pem /path/private-key.pem ec2-user@\[public-ipv6-address\]:path/
```

2. SSH を使用してインスタンスに接続していない場合は、次のようなレスポンスが表示されます。

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)'
can't be established.
RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
Are you sure you want to continue connecting (yes/no)?
```

yes と入力します。

3. 転送が成功した場合、レスポンスは以下のようになります。

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
private-key.pem                                 100%   480     24.4KB/s   00:00
```

プライベートキーをソースインスタンスに転送したため、ターゲットインスタンスに安全に接続し、ファイルを転送できます。次の手順に進み、その方法を確認してください。

ステップ 4: Lightsail Linux インスタンスと Unix インスタンス間でファイルを安全に転送する

次の手順では、あるインスタンス ([ソースインスタンス]) から scp コマンドを実行して、ファイルを別のインスタンス ([ターゲットインスタンス]) に転送します。

scp を使用してインスタンスの間でファイルを転送するには

1. SSH を使用して [ソースインスタンス] に接続します。接続するには、ローカルコンピュータのターミナルプログラムを使用するか、Lightsail でブラウザベースの SSH クライアントを使用します。詳細については、「Linux インスタンスに接続する」を参照してください。
2. [ソースインスタンス] 上のファイルの場所と、[ターゲットインスタンス] 上の送信先パスを決定します。次の例では、プライベートキーファイルの名前は *private-key.pem*、インスタンスのユーザー名は *ec2-user*、インスタンスの IPv4 アドレスは *public-ipv4-address*、インスタンスの IPv6 アドレスは *public-ipv6-address* です。*destination-path/* は、ファイルを転送する [ターゲットインスタンス] 上の場所です。
   - (IPv4) [ソースインスタンス] から [ターゲットインスタンス] にファイルを転送するには、[ソースインスタンス] から次のコマンドを入力します。

     ```
     scp -i /path/private-key.pem /path/my-file.txt ec2-user@public-ipv4-address:destination-path/
     ```

   - (IPv6) [ソースインスタンス] から [ターゲットインスタンス] にファイルを転送するには、[ソースインスタンス] から次のコマンドを入力します。IPv6 アドレスは、(\) でエスケープした角かっこ ([ ]) で囲む必要があります。

     ```
     scp -i /path/private-key.pem /path/my-file.txt ec2-user@\[public-ipv6-address\]:destination-path/
     ```

3. SSH を使用して[ターゲットインスタンス] に接続していない場合は、次のようなレスポンスが表示されます:

   ```
   The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com (10.254.142.33)' can't be established.
   RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
   ```

```
Are you sure you want to continue connecting (yes/no)?
```

yes と入力します。

4. 転送が成功した場合、レスポンスは以下のようになります。

```
Warning: Permanently added 'ec2-198-51-100-1.compute-1.amazonaws.com' (RSA)
to the list of known hosts.
my-file.txt                                   100%   480     24.4KB/s   00:00
```

VPC ピアリングを使用して他の Lightsail AWS サービスと統合する

Amazon Lightsail は、Amazon EC2 や などの集中型 AWS サービスを使用して AWS Identity and Access Management 、開始を容易にします。ただし、サービスがこれらに限定されるわけではありません。

VPC ピアリングを使用してLightsail、リソースを他の AWS サービスと統合できます。VPC ピアリングを有効にしたら、ピアリング接続を介して接続するリソースが、必要なインバウンドトラフィックを受け入れることを確認する必要があります。詳細については、「VPC ピアリングを使用して Lightsailリソースを AWS サービスに接続する」を参照してください。

Amazon Simple Storage Service、Amazon CloudFront、Amazon DynamoDB などの一部の AWS リソースでは、VPC ピアリングを有効にする必要はありません。他の AWS サービスの詳細については、以下のリンクに従ってください。

仮想マシン (仮想プライベートサーバー)

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) は、クラウド内で安心でサイズを変更できるコンピューティング性能を提供するウェブサービスです。ウェブスケールのクラウドコンピューティングを開発者が簡単に利用できるように設計されています。

Amazon EC2 では、最小限の手間で容量を取得して設定できます。使用するコンピューティングリソースのあらゆる面をお客様自身でコントロールできることと、Amazon の実績あるコンピューティング環境で実行できることが特徴です。Amazon EC2 であれば、新規サーバーインスタンスの取得と起動に要する時間が数分にまで短縮されるため、キャパシティーの拡張や縮

小も、コンピューティング要件の変化に合わせてすばやく実行できます。Amazon EC2 はコンピューティングの経済性も変革します。料金のお支払いは、実際に使用したキャパシティーの分だけです。Amazon EC2 には、障害に強いアプリケーションを構築して、一般的な障害シナリオに影響されないようにするための開発者向けツールが用意されています。

Amazon EC2 の詳細をご覧ください。

Amazon VPC

Amazon Virtual Private Cloud (Amazon VPC) では、 AWS クラウドの論理的に隔離されたセクションをプロビジョニングすることで、ユーザーが定義した仮想ネットワーク内で AWS リソースを起動できます。独自の IP アドレス範囲の選択、サブネットの作成、ルートテーブル、ネットワークゲートウェイの設定など、仮想ネットワーク環境全体をお客様がコントロールできます。

Amazon VPC のネットワーク設定は容易にカスタマイズすることができます。たとえば、インターネットにアクセスが可能なウェブサーバーのパブリックサブネットを作成し、データベースやアプリケーションサーバーなどのバックエンドシステムをインターネットにアクセスできないプライベートサブネットに配置できます。セキュリティグループやネットワークアクセスコントロールリストなどの複数のセキュリティレイヤーを活用し、各サブネットの Amazon EC2 インスタンスへのアクセスをコントロールすることができます。

加えて、既存のデータセンターと自分の VPC 間にハードウェア Virtual Private Network (VPN) 接続を作成することができるので、AWS クラウドを既存のデータセンターを拡張するかのように活用することができます。

Amazon VPC の詳細をご覧ください。

サーバーレスコンピューティング

AWS Lambda

AWS Lambda では、サーバーのプロビジョニングや管理を行わずにコードを実行できます。使用したコンピューティング時間に対してのみお支払いいただきます。コードが実行中でなければ料金はかかりません。Lambda を使用すれば、実質どのようなタイプのアプリケーションやバックエンドサービスでも、管理をまったく必要とせずに実行できます。コードをアップロードするだけで、コードの実行とスケールに必要な処理はすべて Lambda により自動的に実行され、高い可用性が維持されます。コードは、他の AWS サービスから自動的にトリガーするよう設定することも、ウェブやモバイルアプリケーションから直接呼び出すよう設定することもできます。

詳細については、「」を参照してください AWS Lambda。

Amazon API Gateway

Amazon API Gateway は、デベロッパーがあらゆる規模で API の公開、保守、モニタリング、セキュリティ保護を簡単に行えるフルマネージドサービスです。 AWS マネジメントコンソールで数回クリックするだけで、アプリケーションの「玄関」として機能する API を作成できます。これによってバックエンドサービスからデータ、ビジネスロジック、機能にアクセスできます。これには Lambda または任意のウェブアプリケーションで実行中のコードを、Amazon EC2 で実行しているワークロードが含まれます。Amazon API Gateway では、最大で数十万個の同時 API コールの受け入れ処理に伴うすべてのタスクを取り扱います。これにはトラフィック管理、認可とアクセスコントロール、モニタリング、API バージョン管理が含まれます。Amazon API Gateway には最低利用料金がなく、初期費用はかかりません。受信した API 呼び出しと、送出されたデータ量に対してのみ料金が発生します。

Amazon API Gateway の詳細をご覧ください。

データベース

Amazon DynamoDB

Amazon DynamoDB は、あらゆるスケールで 1 桁台のミリ秒単位の安定したレイテンシーを必要とする全アプリケーションに対応した、高速かつフレキシブルな NoSQL データベースサービスです。完全マネージド型のクラウドデータベースで、ドキュメントとキー値のストアモデルの両方をサポートしています。データモデルの柔軟性が高く、パフォーマンスが信頼できるため、モバイル、ウェブ、ゲーム、広告、IoT、他の多くのアプリケーションに最適です。

DynamoDB の詳細をご覧ください。

Amazon RDS

Amazon Relational Database Service (Amazon RDS) を使用して、クラウドでリレーショナルデータベースをセットアップ、運用、スケーリングできます。これにより、時間のかかるデータベース管理作業をお客様の代わりに実行して、お客様を管理業務から解放し、アプリケーションとビジネスに集中させることができます。このサービスはコスト効率もよく、データベース容量の変更にも柔軟に対応します。Amazon RDS では、Amazon Aurora、PostgreSQL、MySQL、MariaDB、Oracle、および Microsoft SQL Server の 6 つの使い慣れたデータベースエンジンから選択できます。

Amazon Aurora の詳細をご覧ください。

Amazon Aurora

Amazon Aurora は、MySQL と互換性のあるリレーショナルデータベースエンジンで、高性能の商用データベースの可用性とスピード、およびオープンソースデータベースのシンプルさとコスト効果性を併せ持っています。Aurora は、MySQL よりも最大 5 倍のパフォーマンスを発揮するだけでなく、商用データベースのセキュリティ、可用性、および信頼性を 10 分の 1 のコストで実現します。

Amazon Aurora の詳細を確認してください。

ロードバランサー

エラスティックロードバランシング

Elastic Load Balancing は、受信アプリケーショントラフィックを複数の Amazon EC2 インスタンスに自動的に分散します。これにより、アプリケーションの耐障害性の向上を可能にし、アプリケーショントラフィックのルーティングに必要なロードバランシング能力をシームレスに提供します。

Elastic Load Balancing では、2 種類のロードバランサーがサポートされています。いずれも高可用性、自動スケーリング、および強固なセキュリティを備えています。これには、アプリケーションまたはネットワークレベルの情報に基づいてトラフィックをルーティングする Classic Load Balancer と、リクエストのコンテンツを含むアプリケーションレベルの詳細情報に基づいてトラフィックをルーティングする Application Load Balancer が含まれます。Classic Load Balancer は、複数の Amazon EC2 インスタンス間でのシンプルなトラフィックのロードバランシングに最適です。Application Load Balancer は、高度なルーティング機能、マイクロサービス、およびコンテナベースのアーキテクチャが必要なアプリケーションに最適です。Application Load Balancer は、トラフィックを複数のサービスにルーティングしたり、負荷を同じ Amazon EC2 インスタンスの複数のポートに分散したりする機能を提供します。

Elastic Load Balancing の詳細をご覧ください。

Application Load Balancer

Application Load Balancer は、アプリケーションレイヤーで動作する サービスのロードバランシングオプションで、1 つ以上の Amazon EC2 インスタンスで実行されている複数のサービスやコンテナのコンテンツに基づいてルーティングルールを定義できます。

Application Load Balancer の詳細をご覧ください。

ビッグデータ

Amazon Kinesis のサービス

Amazon Kinesis サービスにより、AWS クラウドのリアルタイムストリーミングデータとの連携が容易になります。Amazon Kinesis サービスには、大量のストリーミングデータを AWS に簡単にロードする Amazon Data Firehose、標準 SQL でストリーミングデータを分析する Amazon Managed Service for Apache Flink、およびストリーミングデータを処理、または分析する独自のカスタムアプリケーションを構築する Amazon Kinesis Data Streams などが含まれます。

Amazon Kinesis サービスの詳細をご覧ください。

Amazon EMR

Amazon EMR は、動的にスケーラブルな Amazon EC2 インスタンス間で大量のデータを簡単、迅速、費用効率よく処理できるマネージド型 Hadoop フレームワークを提供します。また、Apache Spark、HBase、Presto、Flink などの他の一般的なフレームワークを Amazon EMR で実行したり、Amazon S や DynamoDB などの他の AWS データストア内のデータを操作したりもできます。

Amazon EMR では、ログの解析、ウェブインデックス作成、データ変換 (ETL)、機械学習、財務分析、科学シミュレーション、バイオインフォマティクスを含む、さまざまなビッグデータのユースケースが安全かつ確実に処理されます。

Amazon EMR の詳細をご覧ください。

Amazon Redshift

Amazon Redshift は、高速で完全マネージド型のペタバイト規模を誇るデータウェアハウスです。シンプルで費用対効果の高さが特長であり、お客様はすべてのデータを既存のビジネスインテリジェンスツールで分析できます。

Amazon Redshift の詳細をご覧ください。

Storage

Amazon Simple Storage Service (Amazon S3)

Amazon S3 では、開発者や IT チームのための安全で耐久性に優れ、高度にスケーラブルなクラウドストレージが用意されています。Amazon S3 は簡単に使用できるオブジェクトストレージです。シンプルなウェブサービスインターフェイスが用意されており、ウェブ上のどこからでも容

量に関係なくデータを保存、取得できます。Amazon S3 のお支払いは、実際に使用したストレージ分のみです。最低料金もセットアップ費用も不要です。

Amazon S3 は、頻繁にアクセスするデータの汎用ストレージのための Amazon S3 Standard 、長期保存を要し、かつアクセス頻度の低いデータのための Amazon S3 Standard -Infrequent Access (Standard - IA)、長期アーカイブのための Amazon Glacier など、さまざまなユースケースに応じて設計された各種のストレージクラスを提供します。Amazon S3 はまた、データのライフサイクルを通したデータ管理のために設定可能なライフサイクルポリシーを提供します。ポリシーを設定すると、データは自動的に最も適切なストレージクラスに移行します。アプリケーションの変更は一切必要ありません。

Amazon S3 は単独で使用することも、Amazon EC2 や IAM など他の AWS サービス、および、初回のみまたは継続的にデータの取り込みを行うクラウドデータ移行サービスやゲートウェイと一緒に使用することもできます。Amazon S3 では、バックアップと復元、nearline アーカイブ、ビッグデータ分析、ディザスタリカバリ、クラウドアプリケーション、コンテンツ配信など、さまざまなユースケースにおいてコスト効率に優れたオブジェクトストレージを利用できます。

Amazon S3 の詳細をご覧ください。

Amazon Elastic Block Store (Amazon EBS)

Amazon EBS は、AWS クラウドの Amazon EC2 インスタンス用の永続的なブロックストレージボリュームを提供します。コンポーネントに障害が発生した場合でも高い可用性と耐久性を提供できるように、各 Amazon EBS ボリュームはアベイラビリティーゾーン内で自動的にレプリケートされます。Amazon EBS のボリュームは、ワークロードの実行に必要な一貫した低レイテンシーのパフォーマンスを実現します。Amazon EBS を使用すると、使用量の拡張または縮小を数分で行うことができます – プロビジョニングしているサイズに合わせて、低料金でご利用いただけます。

Amazon EBS の詳細をご覧ください。

モニタリングとアラーム

Amazon CloudWatch

Amazon CloudWatch は、AWS クラウドのリソースと、AWS で実行されるアプリケーションをモニタリングするサービスです。CloudWatch を使用して、メトリクスの収集とトラッキング、ログファイルの収集とモニタリング、アラームの設定、AWS リソースの変更への自動対応を行うことができます。CloudWatch では、Amazon EC2 インスタンス、Amazon DynamoDB テーブ

ル、Amazon RDS DB インスタンスなどの AWS リソース、およびアプリケーションやサービスに生成されたカスタムメトリクス、アプリケーションが生成するあらゆるログファイルをモニタリングできます。CloudWatch を使用して、リソースの使用率、アプリケーションのパフォーマンス、運用の状況をシステム全体で把握できます。これらの洞察を使用して対応し、アプリケーションのスムーズな動作を維持できます。

Amazon CloudWatch の詳細をご覧ください。

アプリケーションのデプロイ

AWS Elastic Beanstalk

AWS Elastic Beanstalk は、Java、.NET、PHP、Node.js、Python、Ruby、Go、Docker で開発されたウェブアプリケーションとサービスを、Apache、Nginx、Passenger、IIS などの使い慣れたサーバーにデプロイおよびスケーリングするためのeasy-to-useサービスです。

お客様はコードをアップロードするだけで、Elastic Beanstalk が、キャパシティーのプロビジョニング、ロードバランシング、自動スケーリング からアプリケーションの状態モニタリングまで、デプロイを自動的に処理します。同時に、お客様のアプリケーションが稼動している AWS リソースの完全なコントロールを維持でき、いつでも基本的なリソースにアクセスすることができます。

Elastic Beanstalk の詳細をご覧ください。

アプリケーションコンテナ

Amazon Elastic Container Service (Amazon ECS)

Amazon ECS は非常にスケーラブルかつ高性能なコンテナ管理サービスで、Docker コンテナに対応しており、Amazon EC2 インスタンスのマネージド型クラスターでアプリケーションを簡単に実行できます。Amazon ECS により、クラスター管理インフラストラクチャのインストール、運用、スケールを行う必要がなくなります。簡単な API コールを使用して、Docker 対応のアプリケーションを起動/停止したり、クラスターの詳細な状態を問い合わせたり、多くの使い慣れた機能 (セキュリティグループ、Elastic Load Balancing、Amazon EBS ボリューム、 IAM ロールなど) にアクセスしたりできます。Amazon ECS を使用することで、リソースニーズと可用性要件に基づいて、クラスター全体のコンテナの配置をスケジューリングできます。また、ビジネスやアプリケーションに固有のニーズに合わせた独自のスケジューラやサードパーティー製スケジューラを統合することもできます。

Amazon ECS の詳細をご覧ください。

セキュリティとユーザーサインイン

AWS Identity and Access Management (IAM)

IAM を利用すると、AWS のサービスおよびリソースに対するお客様のユーザーのアクセスを安全にコントロールすることができます。IAM を使用すると、AWS のユーザーとグループを作成および管理し、アクセス権を使用して AWS リソースへのアクセスを許可および拒否できます。

IAM の詳細をご覧ください。

Amazon Cognito ユーザープール

Amazon Cognito でモバイルアプリやウェブアプリに、ユーザーのサインアップやサインインを簡単に追加できます。Amazon Cognito を使用すると、Facebook、Twitter、Amazon などのソーシャル ID プロバイダ経由で、SAML ID ソリューションを使用して、または独自の ID システムを使用してユーザーを認証することもできます。さらに、Amazon Cognito では、ユーザーのデバイスにローカルでデータを保存し、デバイスがオフラインであってもアプリケーションが機能するようにもできます。その後、ユーザーのデバイス間でデータを同期して、使用するデバイスを問わずアプリのエクスペリエンスに整合性を持たせることができます。

Amazon Cognito を使用すると、ユーザーの管理、認証、デバイス間の同期を行うソリューションの構築、安全性の確保、スケーリングに煩わされることなく、優れたアプリのエクスペリエンスを作成することに集中できます。

Amazon Cognito の詳細をご覧ください。

ソース管理とアプリケーションライフサイクル管理

AWS CodeCommit

AWS CodeCommit は、企業が安全でスケーラブルなプライベート Git リポジトリを簡単にホストできるようにするフルマネージド型のソースコントロールサービスです。 は、独自のソース管理システムを運用したり、インフラストラクチャのスケーリングを心配したりする必要性 AWS CodeCommit を排除します。を使用すると AWS CodeCommit 、ソースコードからバイナリまで、あらゆるものを安全に保存でき、既存の Git ツールとシームレスに連携します。

AWS CodeCommitの詳細は、こちらを参照してください。

キューとメッセージング

Amazon SQS

Amazon Simple Queue Service (Amazon SQS) は、高速で、信頼性が高く、スケーラブルな、完全マネージド型のメッセージキューイングサービスです。Amazon SQS を利用すると、簡単かつコスト効率良く、クラウドアプリケーションのコンポーネントを切り離すことができます。Amazon SQS を使用すると、メッセージを失うことなく、どのような量のデータでも転送できます。他のサービスが常に利用可能である必要はありません。Amazon SQS には、高スループットおよび少なくとも 1 回処理の標準キュー、および FIFO (先入先出) 配信および正確に 1 回のみ処理の FIFO キューが含まれています。

Amazon SQS を使用すると、可用性の高いメッセージングクラスターの運用とスケーリングという管理上の負担を軽減しつつ、使用した分だけ低額の料金を支払うことができます。

Amazon SQS の詳細をご覧ください。

Amazon SNS

Amazon Simple Notification Service (Amazon SNS) は、高速かつ柔軟な完全マネージド型のプッシュ通知サービスです。このサービスを使用すると、個々のメッセージを送信したり、多数の受信者にメッセージをファンアウトしたりできます。Amazon SNS により、簡単かつコスト効率の高い方法で、モバイルデバイスユーザーやメール受信者にプッシュ通知を送信したり、他の分散型サービスにメッセージを送信したりもできます。

Amazon SNS を使用すれば、Apple Push Notification Service (APNS)、Google クラウドメッセージング (GCM)、Fire OS、Windows の各デバイス、さらには中国では Baidu Cloud Push を使用して、Android デバイスにも通知を送信できます。Amazon SNS を使用して、世界中のモバイルデバイスユーザーに SMS メッセージを送信できます。

Amazon SNS では、これらのエンドポイント以外にも、Amazon SQS、 AWS Lambda 関数、または任意の HTTP エンドポイントにメッセージを配信できます。

Amazon SNS の詳細をご覧ください。

Amazon SES

Amazon Simple Email Service (Amazon SES) は、Amazon.com が自社の顧客ベースを対象に開発した信頼性の高いスケーラブルなインフラストラクチャを基盤とする費用対効果の高い E メールサービスです。Amazon SES を使用すると、最低料金が適用されることなく、E メールを送受信できます。使用したときに使用した分のみのお支払いとなります。

Amazon SES の詳細をご覧ください。

ワークフロー

Amazon Simple Workflow Service (Amazon SWF)

Amazon SWF は、開発者が並列またはシーケンシャルステップを含むバックグラウンドジョブを構築、実行、拡張するのを支援します。Amazon SWF は、クラウド内の完全マネージド型ステータストラッカーやタスクコーディネーターと考えることができます。

アプリケーションのステップが完了するまでに 500 ミリ秒以上かかる場合は、処理の状態を追跡する必要があります。タスクが失敗した場合は、復旧または再試行する必要があります。Amazon SWF はそのお手伝いをします。

Amazon SWF の詳細をご覧ください。

ストリーミングアプリケーション

Amazon AppStream

Amazon AppStream では、Windows アプリケーションを任意のデバイスで使用できます。

Amazon AppStream を利用すれば、コードを変更することなく、クラウドから既存の Windows アプリケーションをストリーミングして、さまざまなデバイスを使用する多くのユーザーがアプリケーションにアクセスできるようになります。Amazon AppStream を使用すると、アプリケーションは AWS インフラストラクチャにデプロイおよびレンダリングされ、出力はパーソナルコンピュータ、タブレット、携帯電話などのマスマーケットデバイスにストリーミングされます。アプリケーションはクラウドで実行されるため、お客様が使用するデバイスと関係なく、処理とストレージの膨大なニーズに応じてスケールできます。Amazon AppStream では、クラウドからアプリケーションをストリーミングする SDK を利用できます。カスタムクライアント、サブスクリプション、アイデンティティ、およびストレージソリューションを Amazon AppStream と統合し、ビジネスニーズに合わせたカスタムストリーミングソリューションを構築できます。

Amazon AppStream 詳細をご覧ください。

を使用してLightsailリソースを作成する AWS CloudFormation

Amazon Lightsail は と統合されています。これは AWS CloudFormation、 AWS リソースとインフラストラクチャの作成と管理に費やす時間を短縮できるように、リソースのモデル化とセットアップに役立つサービスです。必要なすべての AWS リソース (インスタンスやディスクなど) を記述するテンプレートを作成し、それらのリソースを CloudFormation プロビジョニングして設定します。

を使用すると CloudFormation、テンプレートを再利用してLightsailリソースを一貫して繰り返しセットアップできます。リソースを一度記述し、複数の AWS アカウント およびリージョンで同じリソースを何度もプロビジョニングします。

Lightsail および CloudFormation テンプレート

Lightsail および関連サービスのリソースをプロビジョニングして設定するには、CloudFormation テンプレートについて理解しておく必要があります。テンプレートは、JSON や YAML でフォーマットされたテキストファイルです。これらのテンプレートは、 CloudFormation スタックにプロビジョニングするリソースを記述します。JSON または YAML に慣れていない場合は、 CloudFormation デザイナー を使用してテンプレートの使用 CloudFormation を開始できます。詳細については、 AWS CloudFormation ユーザーガイドのCloudFormation 「デザイナーとは」を参照してください。

Lightsail は CloudFormationでのインスタンスとディスクの作成をサポートします。詳細については、AWS CloudFormation ユーザーガイドの「Lightsail resource type reference」を参照してください。

の詳細 CloudFormation

詳細については CloudFormation、次のリソースを参照してください。

- AWS CloudFormation
- AWS CloudFormation ユーザーガイド
- CloudFormation API リファレンス
- AWS CloudFormation コマンドラインインターフェイスユーザーガイド

アプリケーションのデプロイのために Lightsail リソースを探索する

次のリストには、Lightsail ユーザーガイドで公開されていない Amazon Lightsail の追加情報へのリンクが含まれています。

内容

- ブログ
- チュートリアル
- 動画

ブログ

- Datadog を使用した Amazon Lightsail インスタンスの正常性のモニタリング

  2022 年 3 月 30 日 - Datadog を使用した Lightsail ワークロードのモニタリングにより、アプリケーションのパフォーマンス確保とコスト管理を支援する方法について説明します。

- Amazon Lightsail を使用して AWS での研究のために Galaxy を設定する方法

  2022 年 1 月 13 日 – 科学的ワークフロー、データ統合、デジタル保存プラットフォームである Galaxy を Lightsail でデプロイします。

- What happens when you type a URL into your browser (ブラウザに URL を入力したときの挙動

  2021 年 8 月 26 日 – ブラウザに URL を入力して Enter キーを押すとどうなりますか?

- Amazon Lightsail インスタンスでのメモリの使用状況のモニタリング

  2021 年 6 月 14 日 – モニタリング、アラーム、通知のためにメモリの使用状況を Amazon CloudWatch に送信するように Lightsail インスタンスを設定します。

- を使用したコンテナ化済み ASP.NET ウェブアプリケーションのフリクションレスホスティング Amazon Lightsail

  2021 年 6 月 10 日 – PostgreSQL データベースに接続するコンテナ化された ASP.NET ウェブアプリケーションを取得し、Lightsail にデプロイする方法。

- Amazon Lightsail コンテナを使用した WordPress ウェブサイトの立ち上げ

2021 年 4 月 5 日 – Lightsail コンテナと Lightsail データベースを使用して WordPress ウェブサイトを起動します。

- Lightsail コンテナ: クラウドでコンテナを実行する簡単な方法

2020 年 11 月 13 日 – Lightsail でコンテナベースのワークロードをデプロイします。

- Amazon Lightsail から Amazon EC2 へのウェブサービスの移行

2020 年 10 月 16 日 – Amazon EC2 で本番稼働環境を設定し、ウェブサービスを Lightsail からその環境に移行します。

- Amazon Lightsail インスタンス上で実行する Graylog サーバーの構築

2020 年 7 月 28 日 – Lightsail で Graylog サーバーを構築する方法。

- Lightsail コンテンツ配信ネットワークを使用したウェブサイトのパフォーマンスの向上

2020 年 7 月 23 日 – WordPress と標準のウェブサーバーの両方で動作するように Lightsail ディストリビューションを設定します。

- Amazon Lightsail インスタンスでシステムパフォーマンスをプロアクティブにモニタリングする

2020 年 6 月 4 日 – ユーザーに影響を与える前にシステムパフォーマンスの問題を防ぐことができるように、バースト可能な容量アラートを設定します。

- 新しい Lightsail ファイアウォール機能を使用したサイトセキュリティの強化

2020 年 5 月 7 日 – SSH を使用したリモートアクセスを単一の送信元 IP アドレスに制限します。

- CodeDeploy と CodePipeline を使用してアプリケーションを Amazon Lightsail にデプロイする

2020 年 4 月 23 日 – GitHub に変更をプッシュするたびに、CodeDeploy および CodePipeline と連携してアプリケーションを自動的にデプロイ (または更新) するように Lightsail を設定します。

- でのロードバランサーの使用Amazon Lightsail

2020 年 4 月 21 日 – Amazon Lightsail ロードバランサーを使用して単純な Node.js ウェブアプリケーションのロードバランスを実行する方法

- Ghost を使用して Amazon Lightsail で写真日記を作成する

2020 年 3 月 23 日 – Lightsail で Ghost を使用して写真日記を開始します。

- Amazon Lightsail データベースのヒントとコツ

2020 年 3 月 23 日 — Amazon Relational Database Service (Amazon RDS) に搭載されている高度な機能を使用します。

- モニタリングと通知の設定と使用

  2020 年 2 月 27 日 – 通知先の作成、新しいアラームの作成、およびリソースモニタリングを使用した通知のテスト。

- Amazon Lightsail での高可用性 WordPress サイトのデプロイ、パート 1: WordPress を使用した高可用性 Lightsail データベースの実装

  2019 年 10 月 22 日 – Lightsail で可用性の高い WordPress ウェブサイトを構築します (パート 1)。

- Amazon Lightsail で高可用性の WordPress サイトのデプロイ、パート 2: Amazon S3 と WordPress を併用したメディアファイルの安全な配信

  2019 年 10 月 31 日 – Lightsail で可用性の高い WordPress ウェブサイトを構築します (パート 2)。

- Amazon Lightsail での高可用性の WordPress サイトのデプロイ、パート 3: Amazon CloudFront を使用したセキュリティとパフォーマンスの向上

  2019 年 11 月 7 日 – Lightsail で可用性の高い WordPress ウェブサイトを構築します (パート 3)。

- Amazon Lightsail での高可用性の WordPress サイトのデプロイ、パート 4: Lightsail ロードバランサーを使用したパフォーマンスとスケーラビリティの向上

  2019 年 11 月 14 日 – Lightsail で可用性の高い WordPress ウェブサイトを構築します (パート 4)。

- によるポケットの Platform as a Service の構築Amazon Lightsail

  2019 年 10 月 8 日 – Lightsail でポケットプラットフォームを構築します。

- を使用して Nginx ベースの HTTP/HTTPS ロードバランサーをデプロイするAmazon Lightsail

  2019 年 7 月 8 日 – Lightsail インスタンス内で NGINX ベースのロードバランサーを設定します。

- AWS クラウドは初めてですか? Amazon Lightsail がサポートします

  2019 年 3 月 27 日 – Amazon Lightsail の開始方法。

- 新規 – Amazon Lightsail 用マネージドデータベース

  2018 年 10 月 16 日 – 数回クリックするだけでマネージドデータベースを作成できます。

- Amazon Lightsail の更新: インスタンスサイズの増加と値下げ

  2018 年 8 月 23 日 – Lightsail インスタンスの概要。

- Amazon Lightsail: AWS のパワー、VPS のシンプルさ

  2016 年 11 月 30 日 – Lightsail リリースの発表。

チュートリアル

上位 5 位の実践チュートリアル:

1. 負荷分散された WordPress ウェブサイトを作成する

   2021 年 9 月 8 日 – Lightsail で高可用性の WordPress ウェブサイトを立ち上げます。

2. を使用して WordPress ウェブサイトを移行して管理するAmazon Lightsail

   2021 年 2 月 22 日 – Seahorse ソフトウェアを使用して Lightsail に WordPress ウェブサイトのクローンを立ち上げます。

3. Linux 仮想マシンを起動する

   2020 年 9 月 11 日 – Lightsail を使用して Linux インスタンスを起動、設定、接続します。

4. Windows 仮想マシンを起動する

   2020 年 9 月 11 日 – Lightsail を使用して Windows インスタンスを起動、設定、接続します。

5. で cPanel と WHM インスタンスを起動するAmazon Lightsail

   2020 年 7 月 27 日 – このチュートリアルでは、Lightsail で cPanel および WHM インスタンスを起動して実行した後に実践できるいくつかのステップについて説明します。

- で Magento をセットアップして設定する方法Amazon Lightsail

  2021 年 8 月 11 日 – e コマースサイトを立ち上げて稼働させます。

- How to connect your WordPress site to an object storage bucket (WordPress サイトをオブジェクトストレージバケットに接続する方法

  2021 年 7 月 14 日 – Lightsail で WordPress サイトを設定し、このウェブサイトを Lightsail バケットに接続します。

- Create object storage buckets (オブジェクトストレージバケットを作成する

  2021 年 7 月 14 日 – Amazon Lightsail でオブジェクトストレージバケットを作成します。

- WordPress ウェブサイトを Amazon Lightsail バケットに接続し、配信する

  2021 年 7 月 14 日 – Lightsail コンテンツ配信ネットワーク (CDN) ディストリビューションのオリジンとして Lightsail バケットを設定します。

- How to setup and configure Plesk (Plesk のセットアップおよび設定方法

  2021 年 4 月 22 日 – Lightsail で Plesk ホスティングスタックを起動して実行します。

- Prestashop e コマースサイトの設定方法

  2021 年 4 月 1 日 – Bitnami ブループリントによって認定された PrestaShop を使用し、Lightsail インスタンスを起動して設定します。

- で Amazon EFS を使用する方法Amazon Lightsail

  2021 年 3 月 15 日 – VPC ピアリングを使用して、Lightsail インスタンスから Amazon EFS ファイルシステムを作成して接続します。

- How to setup a Nginx reverse proxy (Nginx リバースプロキシを設定する方法

  2021 年 2 月 10 日 – Lightsail コンテナを使用して Nginx リバースプロキシを設定します。

- How to Serve a Flask pp (Flask pp を提供する方法

  2021 年 2 月 3 日 – Lightsail コンテナを使用して Flask アプリケーションを保守する方法について説明します。

- でのコンテナイメージの作成、プッシュ、デプロイAmazon Lightsail

  2020 年 11 月 11 日 – Dockerfile を使用して、ローカルマシンにコンテナイメージを作成します。

- Build a Drupal website (Drupal のウェブサイトを構築する

  2020 年 9 月 11 日 – Lightsail で本番稼働に対応した Drupal ウェブサイトをデプロイしてホストします。

- Build a LAMP stack web App (LAMP スタックウェブアプリケーションを構築する

  2020 年 9 月 9 日 – Lightsail で高可用性の PHP ウェブアプリケーションを起動して実行します。

- ディストリビューションと動作するように WordPress インスタンスを設定する

2020 年 7 月 16 日 – Lightsail ディストリビューションで動作するように WordPress インスタンスを設定します。

- Launch a WordPress website (WordPress ウェブサイトを起動する

2020 年 3 月 23 日 – Lightsail 仮想マシンに WordPress をインストールしてウェブサイトを起動して実行します。

- Host a .NET application (.NET アプリケーションをホストする

2020 年 3 月 20 日 – Lightsail を使用して .NET アプリケーションを構築してデプロイします。

- Amazon Route 53 のドメインを Lightsail リソースにマッピングする

example.com などのドメインのトラフィックを Lightsail リソースにルーティングします。

動画

- Amazon Lightsail チュートリアル: Django アプリケーションをデプロイする

2021 年 7 月 14 日 – このチュートリアルでは、Django アプリケーションを作成します。

- Amazon Lightsail チュートリアル: Flask アプリケーションをデプロイする

2021 年 7 月 14 日 – このチュートリアルでは、Flask アプリケーションを作成します。

- Amazon Lightsail チュートリアル: NGINX リバースプロキシをデプロイする

2021 年 7 月 14 日 – Flask アプリケーションを作成し、Docker コンテナを構築し、Lightsail にコンテナサービスを作成したら、アプリケーションをデプロイします。

- Amazon Lightsail チュートリアル: e コマースサイトをデプロイする

2021 年 7 月 14 日 – Bitnami ブループリントによって認定された PrestaShop を使用して Lightsail インスタンスを起動し、設定します。

- でコンテナ化されたアプリケーションをデプロイするAmazon Lightsail

2020 年 12 月 29 日 – Lightsail でコンテナ化されたアプリケーションをデプロイする方法について説明します。

- Amazon Lightsail チュートリアル: Drupal ウェブサイトを構築する

2020 年 8 月 31 日 – Drupal インスタンスを起動して設定します。

- Amazon Lightsail チュートリアル: LAMP スタックアプリケーションをデプロイする

2020 年 8 月 31 日 – LAMP (Linux Apache MySQL PHP) スタックアプリケーションを単一の Lightsail インスタンスにデプロイします。

- Amazon Lightsail チュートリアル: Linux インスタンスを起動する

  2020 年 8 月 31 日 – Linux インスタンスを起動する方法について説明します。

- Amazon Lightsail チュートリアル: Windows インスタンスを起動する

  2020 年 8 月 31 日 – Windows インスタンスを起動する方法について説明します。

- Amazon Lightsail チュートリアル: 独自の Minecraft サーバーを実行する

  2020 年 8 月 31 日 – 専用の Minecraft サーバーを設定する方法について説明します。

- Amazon Lightsail チュートリアルのご紹介

  2020 年 8 月 31 日 – Lightsail を使用してクラウドジャーニーを開始しましょう。

- Amazon Lightsail:を始める最も簡単な方法AWS

  2020 年 3 月 20 日 – Lightsail は AWS を開始する最も簡単な方法です。仮想サーバー、ストレージ、データベース、ネットワークに加えて、コスト効率の良い月額プランをご利用いただけます。

- での Plesk インスタンスの設定Amazon Lightsail

  2019 年 3 月 27 日 – Lightsail で Plesk インスタンスを設定する方法について説明します。

- で WordPress Multisite を設定するAmazon Lightsail

  2019 年 1 月 15 日 – Lightsail で WordPress マルチサイトインスタンスを設定する方法について説明します。

- の管理Lightsail

  2018 年 10 月 9 日 – Lightsail の主な機能を簡単に確認します。

- で MEAN スタックアプリケーションをデプロイするAmazon Lightsail

  2018 年 6 月 5 日 – Lightsail の MEAN ブループリントを使用して、カスタムアプリケーションをクラウドにデプロイします。

- で WordPress インスタンスをデプロイするAmazon Lightsail

  2018 年 6 月 5 日 – Lightsail に WordPress インスタンスをデプロイします。

Lightsail の詳細な請求情報と使用状況を表示します。

Amazon Lightsail の請求は、Amazon Web Services (AWS) の請求によって処理されます。Lightsail 請求書を表示するには、[AWS Billing and Cost Management ダッシュボード] を選択するか、Lightsail コンソールのナビゲーションバーにある [請求] を選択します。料金の詳細については、「Lightsail の料金」を参照してください。

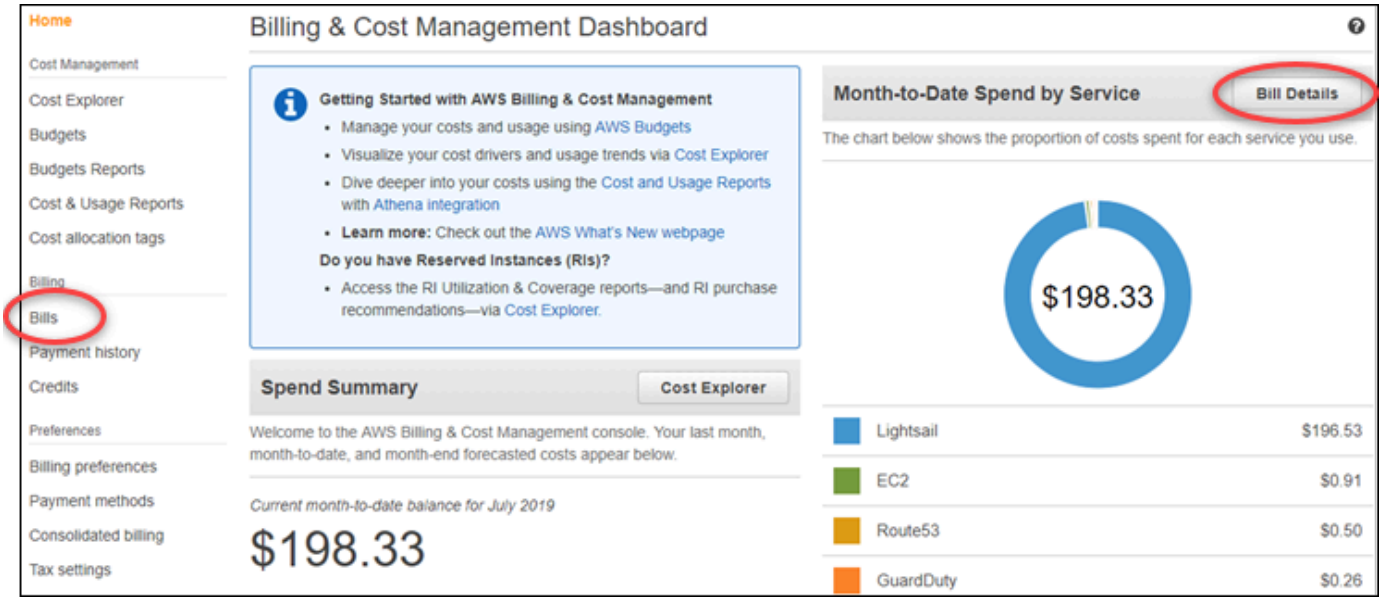
Lightsail の請求明細を表示する

月額の Lightsail の請求の詳細な内訳を表示するには:

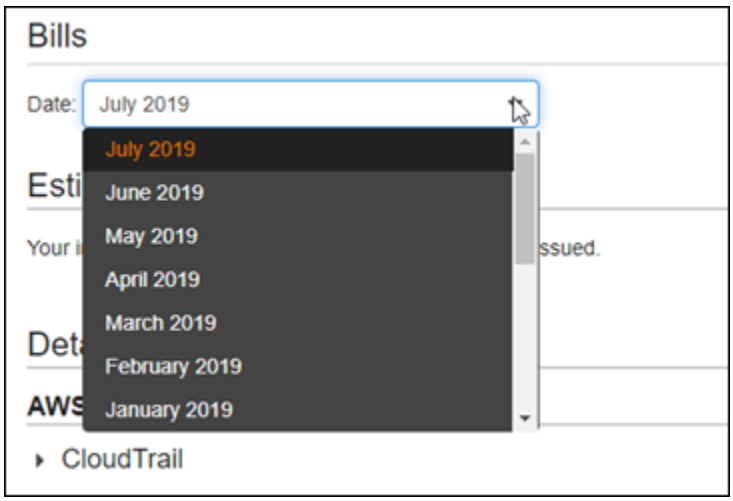
1. AWS Billing and Cost Management ダッシュボードにサインインします。

   請求ダッシュボードのホームページには、請求書の過去 1 か月の内訳が表示されます。

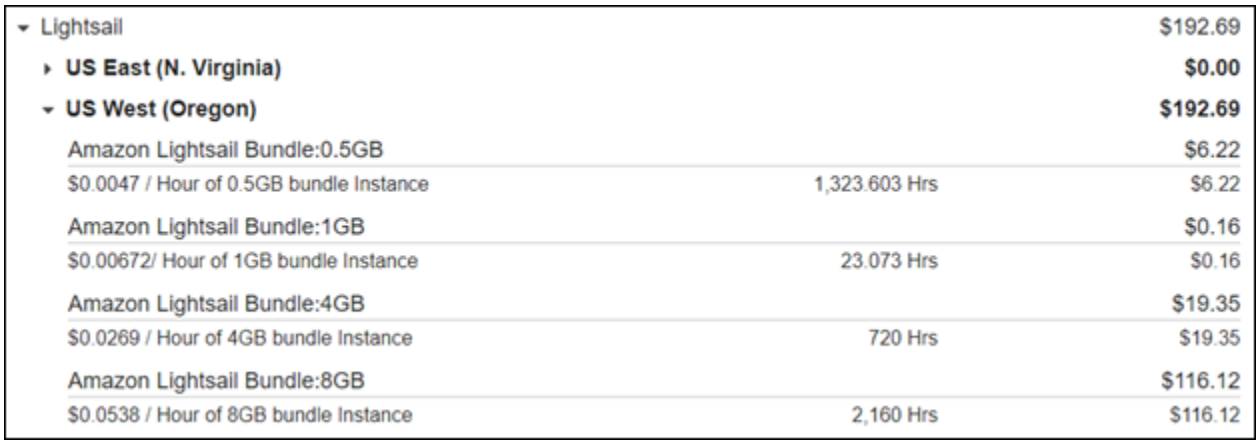
2. 月額の料金の詳細バージョンを表示するには、ダッシュボードのホームページで [料金明細] を選択するか、左側のナビゲーションペインで [請求] を選択します。



3. [Date (日付)] ドロップダウンメニューを選択して、現在の月以外の月を選択します。



4. [請求] ページを下にスクロールし、Lightsail の明細項目を展開して、各リージョンの詳細な使用状況を表示します。



請求の使用タイプ

次のリストは、Lightsail の請求および使用状況レポートに表示される使用タイプを示します。これらの使用タイプは、Lightsail リソースに対する月額料金を識別するのに役立ちます。

> **Note**
>
> リージョンコードを指定する以下の使用タイプについては、このガイドの「請求のリージョンコード」を参照して、対応する AWS リージョンを識別します。

- Amazon Lightsail Bundle:{Size}GB: 使用された Linux または Unix インスタンスプラン (時間単位) 。{Size} は、使用するインスタンスプランのメモリ仕様を定義します。たとえば、4 GB のメ

モリが指定されている場合、24 USD/月の Linux または Unix インスタンスプランの請求時間が表示されます。

- Amazon Lightsail Bundle:{Size}GB (Windows): 使用された Windows インスタンスプラン (時間単位) 。{Size} は、使用するインスタンスプランのメモリ仕様を定義します。たとえば、4 GB のメモリが指定されている場合、44 USD/月の Windows インスタンスプランの請求時間が表示されます。
- Amazon Lightsail MemoryOptimizedBundle:{Size}GB: 使用された Linux または Unix のメモリ最適化インスタンスプラン (時間単位) 。{Size} は、使用するメモリ最適化インスタンスプランのメモリ仕様を定義します。たとえば、16GB のメモリが指定されている場合、Linux または Unix のメモリ最適化インスタンスプランの請求時間が表示されます。
- Amazon Lightsail MemoryOptimizedBundle:{Size}GB (Windows): 使用される Windows メモリ最適化インスタンスプラン (時間単位) 。{Size} は、使用するメモリ最適化インスタンスプランのメモリ仕様を定義します。たとえば、16GB のメモリが指定されている場合、Windows メモリ最適化インスタンスプランの月額 134 USD の請求時間が表示されます。
- Amazon Lightsail RelationalDatabase:{Size}GB: 使用される標準データベースプラン (時間単位) 。{Size} は、使用するデータベースプランのメモリ仕様を定義します。たとえば、4 GB のメモリが指定されている場合、60 USD/月の標準データベースの請求時間が表示されます。
- Amazon Lightsail RelationalDatabase:{Size}GB (高可用性): 使用される高可用性データベースプラン (時間単位) 。{Size} は、使用するデータベースプランのメモリ仕様を定義します。たとえば、4 GB のメモリが指定されている場合、120 USD/月の高可用性データベースプランの請求時間が表示されます。
- Amazon Lightsail Region-DiskUsage: 使用されるブロックストレージディスクの量 (GB/月)。
- Amazon Lightsail DNS-Queries: 当月の DNS クエリの数 (カウント)。
- Amazon Lightsail Load Balancer: 使用されるロードバランサーの量 (時間単位)。
- Amazon Lightsail Region-SnapshotUsage: 保存されたスナップショットデータの量 (GB/月)。
- Amazon Lightsail Region-UnusedStaticIP: 接続されていない静的 IP の量 (時間単位)。
- Amazon Lightsail Region-TotalDataXfer-In-Bytes: 転送 (in) されたデータの合計量 (GB 単位)。
- Amazon Lightsail Region-TotalDataXfer-Out-Bytes: 転送 (out) されたデータの合計量 (GB 単位)。
- Amazon Lightsail Region-DataXfer-Out-Overage-Bytes: 使用されたインスタンスまたはデータベースプランの許容範囲を超えてインターネットまたはパブリック IP に転送されたデータの量 (GB 単位)。

請求のリージョンコード

Lightsail 請求および使用状況レポートはコードおよび略名を使用します。たとえば、使用タイプの場合、リージョンは次の略語のいずれかに置き換えられます。

- APN1: アジアパシフィック (東京) (ap-northeast-1)
- APN2: アジアパシフィック (ソウル) (ap-northeast-2)
- APS1: アジアパシフィック (シンガポール)(ap-southeast-1)
- APS2: アジアパシフィック (シドニー)(ap-southeast-2)
- APS3: アジアパシフィック (ムンバイ)(ap-south-1)
- APS4: アジアパシフィック (ジャカルタ) (ap-southeast-3)
- CAN1: カナダ (中部)(ca-central-1)
- EU: 欧州 (アイルランド)(eu-west-1)
- EUC1: 欧州 (フランクフルト)(eu-central-1)
- EUW2: 欧州 (ロンドン)(eu-west-2)
- EUW3: 欧州 (パリ) (eu-west-3)
- EUN1: 欧州 (ストックホルム) (eu-north-1)
- USE1: 米国東部 (バージニア北部) (us-east-1)
- USE2: 米国東部 (オハイオ)(us-east-2)
- USW2: 米国西部 (オレゴン)(us-west-2)

Lightsail に関するよくある質問への回答について

このセクションでは、Lightsail に関連する一般的な質問と回答を、以下のカテゴリに分類して説明します。

トピック

- Lightsail とそのグローバル可用性について説明します。
- 請求とアカウント管理
- Lightsail でのデータ転送
- ブロックストレージ (ディスク)
- 証明書
- 連絡先とモニタリング通知
- コンテナサービス
- コンテンツ配信ネットワークディストリビューション
- データベース
- ドメイン
- Amazon Elastic Compute Cloud (Amazon EC2) に Lightsail リソースをエクスポートする
- インスタンス
- ロードバランサー
- 手動および自動スナップショット
- リソースのヘルスメトリクスとアラーム
- ネットワーク
- オブジェクトストレージとバケット
- Lightsail のタグ

各カテゴリに記載されているリンクに従って、Lightsail に関するよくある質問に対する詳細な回答を見つけてください。

Lightsail とそのグローバル可用性について説明します。

Amazon Lightsail とは?

Amazon Lightsail は、開発者、中小企業、学生、およびクラウドでウェブサイトやウェブアプリケーションを構築してホストするソリューションを必要とする他のユーザー AWS にとって、 の使用を開始する最も簡単な方法です。 Lightsailは、開発者にコンピューティング、ストレージ、ネットワーク容量を提供します。 Lightsailには、仮想マシン、コンテナ、データベース、CDN、ロードバランサー、DNS 管理など、プロジェクトをすばやく起動するために必要なすべてが含まれており、予測可能な月額料金で提供されます。

Lightsail の機能

アプリケーションのデプロイと管理を容易に行うためのすべてが含まれた、事前設定済みの仮想プライベートサーバー (インスタンス) を作成できます。また基盤インフラストラクチャやオペレーティングシステムのセキュリティとヘルスを Lightsail で管理するデータベースも作成できます。Lightsail は、インスタンス数が数十以下のプロジェクトや、シンプルな管理インターフェイスを必要とするデベロッパーに最適です。Lightsail の一般的ユースケースには、ウェブサイト、ウェブアプリケーション、ビジネスソフトウェア、ブログ、e コマースサイトなどの運営が挙げられます。プロジェクトが大きくなるにつれて、インスタンスでロードバランサーとアタッチされたブロックストレージを使用して冗長性と稼働時間を向上させ、多数の他の AWS サービスにアクセスして新しい機能を追加できます。

Lightsail で API を利用できますか?

はい。Lightsail コンソールでのアクションはすべて、一般公開された API でサポートされます。Lightsail CLI および API のインストール方法と使い方を確認する。

Lightsail にサインアップするには、どうすればよいですか?

Lightsail の使用を開始するには、[Get Started] (使用開始) を選択してログインください。Amazon Web Services アカウントを使用して Lightsail にアクセスできます。まだアカウントがない場合、アカウントを作成するように求められます。

どの AWS リージョン でLightsail利用できますか?

Lightsail は、世界中のさまざまなリージョンで利用できます。利用可能なリージョンの詳細については、「Lightsail のリージョンとアベイラビリティーゾーン」を参照してください。

アベイラビリティーゾーンとは何ですか?

アベイラビリティーゾーンは、物理的独自性を持った独立したインフラストラクチャで実行されるデータセンターの集合体で、高度な信頼性を実現できるよう設計されています。発電機や冷却装置などの一般的な障害発生点は、アベイラビリティーゾーン間では共有されていません。加えて、アベイラビリティーゾーンは物理的に離れているため、火災や竜巻、洪水などの極めてまれな災害は、単独のアベイラビリティーゾーンにしか影響しません。

Lightsail の Service Quotas とは何ですか?

どのクォータを増やすことができるかなど、最新のLightsail Service Quotas については AWS 全般のリファレンス の「Lightsail Service Quotas」を参照ください。Service Quotas を増やすには、サポート でケースを開きます。

より詳細なヘルプを得るにはどうすればよいですか?

Lightsail のコンテキストに敏感なヘルプパネルは、コンソール上のアクションに関するすぐに使える有力なヒントを提供しています。ヘルプパネルを開くには、Lightsail コンソールの右上隅にあるヘルプパネルアイコン ⓘ を選択します。Lightsail コンソールから、入門ガイド、概要、ハウツートピックのライブラリにもアクセスできます。また、 Lightsail API を使用する場合、または Lightsail には AWS CLI、サポートされているすべてのプログラミング言語の完全な API リファレンスがあります。また Lightsail サポートリソースを利用することもできます。

アカウントや請求に関して問題がある場合は、 サポート までオンラインでお問い合わせください。お客様のLightsailアカウントから年中無休で無料アクセスいただけます。

Lightsail の使用方法に関する一般的な質問については、Lightsail ドキュメントおよび「サポートフォーラム」で検索してください。

さらに、 は、個々のニーズを満たすための有料プランの配列 サポート を提供します。

請求とアカウント管理

> **Note**
>
> Lightsail でのデータ転送に関するトピックは、Lightsail でのデータ転送 に移動されました。

Lightsail プランの課金対象を教えてください。

Lightsail プランはオンデマンドの時間料金で請求されるため、ご利用分のみのお支払いとなります。ご利用の Lightsail プランごとに固定の時間料金が課金され、プランの月額上限まで適用されます。最も安価なLightsailプランは、0.0067 USD/時間 (5 USD/月) から始まります。Windows Server ライセンスを含むLightsailプランは、0.0127 USD/時間 (9.50 USD/月) から始まります。

プランに対して課金されるのは、どのようなときですか?

Lightsail インスタンスとマネージドデータベースは、削除されるまで料金が発生します。これらのリソースは、停止状態であっても料金が発生します。月末までに Lightsail インスタンスまたはマネージドデータベースを削除した場合、その月に使用した Lightsail インスタンスまたはマネージドデータベースの合計時間数に基づいて、比例配分された料金のみが課金されます。例えば、最も低価の Lightsail インスタンスプランを 1 か月で 100 時間使用した場合、46 セント (100*0.0046) が課金されます。

Lightsail インスタンスを無料で試すことはできますか？

はい。既存または新規の AWS いずれのお客様でも、5 USD Lightsailプランを 750 時間無料で利用できます。また、9.50 米ドルの Windows プランを利用いただくと、Windows Server ライセンスを含んだ Lightsail プランも無料で試用いただけます。750 時間内で使用するインスタンスの数に制限はありません。例えば、1 つの Lightsail インスタンスを 1 か月間実行したり、10 個の Lightsail インスタンスを 75 時間実行したりできます。無料トライアルは Lightsail にサインアップされた最初の月の間だけ、ご利用頂けます。アカウントが組織 (AWS Organizations の下) にリンクされている場合、組織内の 1 つのアカウントのみが AWS 無料利用枠 オファーの恩恵を受けることができます。

インスタンスプランには、データ転送許容量が含まれています。インスタンスのデータ転送の入力および出力の両方はデータ転送許容量にカウントされます。データ転送許容量を超えると、インスタンス.(無料トライアル期間中のものを含む) には外部に転送された超過データに対してのみ料金が発生します。データ転送費用の詳細については、「データ転送の課金対象を教えてください。」を参照してください。

> **Note**
>
> AWS 無料利用枠の一部として、一部のインスタンスバンドルで Amazon Lightsailを無料で開始できます。詳細については、「Amazon Lightsail 料金表」ページの「AWS Free Tier」を参照してください。

Lightsail 無料体験はいつから始まりますか？

Lightsail 無料トライアルの特典は、最初の無料トライアル対象リソースが開始された時点で開始されます。

インスタンスとデータベースの 90 日間の延長無料トライアルは、一部のプラン (バンドル) にのみ適用されます。このオファーは、2021 年 7 月 8 日以降Lightsailに の使用を開始した新規または既存の AWS アカウントに適用されます。詳細については、Lightsail 料金表ページ を参照してください。

Lightsail マネージド型データベースの課金対象を教えてください。

Lightsail マネージドデータベースには 4 つのプランサイズがあります。最初のプランは 1 か月あたり 15 USD で、 1 GB の RAM データベースインスタンス、40 GB の SSD ストレージ、100 GB のデータ転送枠がご利用いただけます。高可用性プランはスタンダードプランの 2 倍の費用がかかります。これは、冗長性のために別のアベイラビリティーゾーンで追加のデータベースインスタンスとストレージが実行されるためです。

Lightsail マネージド型データベースを無料で試すことはできますか？

はい。Lightsail の新規のお客様は 15 USD の Lightsail プランを 1 か月間無料でお試しいただけます。

Lightsail ブロックストレージの課金対象を教えてください。

Lightsailブロックストレージは 1 か月あたり 0.10 USD/GB の費用がかかります。

Lightsail ロードバランサーの課金対象を教えてください。

Lightsail ロードバランサーは 1 か月あたり 18 USD の費用がかかります。

証明書管理の課金対象を教えてください。

Lightsail 証明書および証明書管理は Lightsail ロードバランサーを使用する場合、無料でご利用いただけます。

Lightsail 静的 IPv4 アドレスの費用を教えてください。

Lightsail インスタンスにアタッチされた静的 IP アドレスにはコストがかかりません。静的 IP は IPv6 専用インスタンスに割り当てることはできません。IPv4 アドレスは希少なリソースのた

め、Lightsail では効率的な利用に努めています。そのため、インスタンスに 1 時間以上アタッチされていない静的 IP に対しては、0.005 米ドル/時間という少額の料金を課金しています。

Lightsail ドメインにはどれくらいの費用がかかりますか？

リンク先の .pdf ファイルに記載されている料金は、2021 年 12 月 22 日以降の新規ドメイン名登録、既存のドメイン名登録の更新に適用されます。すべての料金には DNS ゾーンとプライバシー保護が含まれています。ドメイン登録のコストの詳細については、「Amazon Route 53 のドメイン登録の料金」および「ドメイン登録」を参照してください。

Lightsail の DNS 管理の課金対象を教えてください。

DNS 管理は Lightsail 内で無料でご利用いただけます。最大で 6 つの DNS ゾーンと、各 DNS ゾーンに必要な数だけのレコードを作成できます。1 か月あたり 300 万 DNS クエリの月間許容枠が、お客様のゾーンでご利用いただけます。1 か月のクエリ数が最初の 300 万を超えると、100万 DNS クエリあたり 0.40 USD の料金が適用されます。

Lightsail スナップショットの課金対象を教えてください。

Lightsail スナップショット (手動および自動) の保存にかかる料金は、 1 か月あたり 0.05 USD/GB です。つまり、28 GB のスペースを使用しているインスタンスのスナップショットを作成して、それを1 か月間保持した場合、その月は 1.40 USD のお支払いとなります。

同じインスタンスに対して複数の連続的なスナップショットを作成すると、スナップショットのコストは Lightsail によって自動的に最適化されます。新しいスナップショットを作成するたびに、変更されたデータ部分に対してのみ課金されます。上記の例でいうと、インスタンスのデータが 2 GB 分のみ変更された場合、2 番目のインスタンススナップショットにかかる料金は 1 か月あたり 0.10 USD のみとなります。

AWS アカウントを管理するにはどうすればよいですか?

Lightsail は AWS サービスであり、 AWS クラウドインフラストラクチャで実行されます。同じ AWS アカウントと認証情報を使用して、Lightsail および AWS マネジメントコンソールにログインします。

AWS アカウントのパスワード、ユーザー名、問い合わせ方法、オプトインリージョン (デフォルトで無効になっているリージョン) 、請求情報の変更など、 AWS アカウントを AWS Billing and Cost Management コンソールから管理できます。

どのオプトインリージョンを有効または無効にするかについて、どのように管理すれば良いですか？

オプトインリージョン (デフォルトで無効になっているリージョン) は有効または無効にできます。オプトインリージョンを使用するには、まずそれを有効にする必要があります。利用可能なリージョンおよびオプトインリージョンの管理方法の詳細については、「Lightsail のリージョンとアベイラビリティーゾーン」を参照してください。

無効化されたオプトインリージョンのリソースはどうなりますか？

無効になっているオプトインリージョンのリソースは引き続き実行され、通常のレートで料金が発生します。無効なオプトインリージョンのリソースは、リージョンが無効になっているLightsail間、Lightsail API、 AWS CLIまたは SDKs で管理することはできません。このようなリソースを削除するには、まずリージョンを再度一時的に有効にして、リソースを管理できるようにする必要があります。詳細については、「無効なオプトインリージョンのリソースを削除する方法を教えてください。」を参照してください。

無効なオプトインリージョンのリソースを削除する方法を教えてください。

オプトインリージョンを無効にしてからリソースを削除する場合、リージョンを再度一時的に有効にして、当該リソースを削除する必要があります。詳細については、Lightsail のオプトインリージョンを無効にするを参照してください。

Lightsail の法的な利用規約とは何ですか?

Lightsail は Amazon のウェブサービスであるため、Lightsail を使用するには、最初に AWS カスタマーアグリーメントおよびサービス条件に同意していただく必要があります。Lightsail インスタンスを作成する場合は、お客様のソフトウェア使用に対して販売者のエンドユーザーライセンス契約が適用されることにも同意いただく必要があります。詳細は、インスタンスの作成ページで確認できます。

Lightsail 請求の支払い方法を教えてください。

Billing AWS and Cost Management コンソールを使用して、請求書の支払いと管理を行うことができます。 はほとんどの主要なクレジットカード AWS を受け入れます。お支払い方法の管理についての詳細は、こちらを参照してください。

Lightsail でのデータ転送

インスタンスでプランのデータ転送許容量を超えた場合はどうなりますか？

データ転送プランは、大部分のお客様の使用量が枠内で収まり、追加料金の請求が発生しないように設計されています。インスタンスがデータ転送枠を超えた場合は、使用したデータ転送量に対して GB 単位で超過料金が発生します (インターネットへのデータ転送に対してのみ)。

インスタンスがプランのデータ転送枠を超過した場合でも、多くのタイプのデータ転送が無料です。Lightsail インスタンスおよびデータベースへのデータ転送は常に無料です。プライベート IP アドレスが使用されている場合、Lightsailインスタンスから別のLightsailインスタンス、LightsailインスタンスとLightsailマネージドデータベース間、または同じリージョンの AWS リソースへのデータ転送も無料です。

インスタンスにはどの種類のデータ転送が課金されますか?

インスタンスプランの毎月の無料データ転送枠を超えると、パブリック IP アドレスを使用する場合、Lightsailインスタンスからインターネット、別の AWS リージョン 、または同じリージョンの AWS リソースへのデータ転送に対して課金されます。無料許容量を超過した際のこれらのデータ転送料金は、以下の通りです。

- 米国東部 (オハイオ) (us-east-2): 0.09 米ドル/GB
- 米国東部 (バージニア北部) (us-east-1): 0.09 米ドル/GB
- 米国西部 (オレゴン) (us-west-2): 0.09 米ドル/GB
- アジアパシフィック (ムンバイ) (ap-south-1): 0.13 米ドル/GB
- アジアパシフィック (ソウル) (ap-northeast-2): 0.13 米ドル/GB
- アジアパシフィック (シンガポール) (ap-southeast-1): 0.12 米ドル/GB
- アジアパシフィック (シドニー) (ap-southeast-2): 0.17 米ドル/GB
- アジアパシフィック (ジャカルタ) (ap-southeast-3): 0.132 USD/GB
- アジアパシフィック (東京) (ap-northeast-1): 0.14 米ドル/GB
- カナダ (中部) (ca-central-1): 0.09 米ドル/GB
- 欧州 (フランクフルト) (eu-central-1): 0.09 米ドル/GB
- 欧州 (アイルランド) (eu-west-1): 0.09 米ドル/GB

- 欧州 (ロンドン) (eu-west-2): 0.09 米ドル/GB
- 欧州 (パリ) (eu-west-3): 0.09 米ドル/GB
- 欧州 (ストックホルム) (eu-north-1): 0.09 米ドル/GB

複数のアベイラビリティーゾーンで作成されたインスタンスは、ゾーン間でプライベートに無料通信でき、同時に障害が発生しにくくなります。アベイラビリティーゾーンでは、データ転送コストが増加したり、アプリケーションの安全性を損なうことなく、可用性の高いアプリケーションまたはウェブサイトを構築できます。

インスタンスにおけるデータ転送枠はどのように機能しますか?

すべての Lightsail インスタンスプランには、データ転送枠が含まれています。インスタンスのデータ転送の受信と送信の両方がデータ転送量としてカウントされます。データ転送許容量を超えた場合、Lightsailインスタンスからインターネット、またはインスタンスのパブリック IP アドレスを使用する AWS リソースへの超過データ転送に対してのみ課金されます。許容量を超えるデータ転送に対するこの追加課金は、無料トライアル期間中のリソースも対象になります。データ転送枠は毎月リセットされますが、その月の間であれば必要なときにいつでも消費できます。

Lightsail インスタンスへの過剰なデータ転送 IN に対して課金されることはありません (例 1 を参照) 。データ転送許容量は、リージョン内の同じバンドル (bundleId) のインスタンスについて集計されます (例 2 と例 3 を参照) 。データ転送許容量は、同じサイズの IPv4 インスタンスと IPv6 インスタンスについても集計されます (例 4 を参照) 。インスタンスを削除して新しいインスタンスを作成しても、データ転送許容量はリセットされません (例 5 を参照) 。新しいインスタンスを作成しても、既存のデータ転送の超過は相殺されません ([例 6] を参照)。Lightsail バンドルの詳細については、Amazon Lightsail API Reference の「バンドル」を参照してください。

- 例 1 – 1 か月あたり 5 米ドルのインスタンスバンドル (bundleId nano_3_0) があり、データ転送許容量は 1 TB です。500 GB のデータをインターネット (データ転送 OUT) に送信し、400 GB のデータをインスタンス (データ転送 IN) に送信すると、1 TB の許容量の 900 GB を消費します。さらに 200 GB のデータをインターネットに送信すると、許容量を 100 GB 超過し、100 GB のデータ転送超過料金が請求されます。次に 200 GB のデータをインスタンスに送信する場合、超過分に対して課金されません。
- 例 2 – リージョンで 1 か月あたり 5 米ドルのインスタンスバンドル (bundleId nano_3_0) が 1 か月あたり 2 つあり、それぞれに 1 TB のデータ転送手当がある場合、2 TB のデータ転送手当が合計で付与されます。1 番目のインスタンスで 1.5 TB のデータをインターネットに送信し、2 番目のインスタンスで 100 GB のデータをインターネットに送信する場合、合計許容量である 2 TB では 400 GB のままとなり、データ転送出力超過料金は請求されません。

- 例 3 – インスタンスバンドルの 2 つのセットを作成します。米国西部 (オレゴン) リージョンで、A を 2 つの 1 か月あたり 5 米ドルのインスタンスバンドル (bundleId nano_3_0) に設定し、B を 3 つの 1 か月あたり 7 米ドルのインスタンスバンドル (bundleId micro_3_0) に設定します。これにより、セット A に対して 2 TB のデータ転送許容量、セット B に対して 6 TB のデータ転送許容量が得られます。セット A インスタンスを介してインターネットに 3 TB のデータを、セット B インスタンスを介してインターネットに 4 TB のデータを転送した場合、セット A インスタンスのデータ転送許容量を超え、1 TB のデータ転送出力超過料金が請求されます。セット B インスタンスの 2 TB の許容量を超えていません。
- 例 4 – 請求月の 20 日以内に、1 か月あたり 3.50 米ドルの IPv6 インスタンスバンドル (bundleId nano_ipv6_3_0) の 1 TB データ転送許容量の合計 600 GB を消費しました。インスタンスのネットワークタイプをデュアルスタック (bundleId nano_3_0、月間料金 5 米ドル) に切り替えることを 21 日目に決定しました。その月のデータ転送使用率はリセットされず、600 GB のままになり、400 GB の許容量が残ります。請求月の残りの期間に 500 GB のデータをインターネットに送信すると、100 GB のデータ転送超過料金が発生します。
- 例 5 – インスタンスバンドル (bundleId nano_3_0) は 1 か月あたり 5 米ドルで、それぞれ 1 TB のデータ転送許容量があります。請求月内に合計 3 TB のデータ転送許容量のうち 1 TB を消費したと仮定すると、残りのデータ転送許容量は 2 TB になります。すべてのインスタンスを削除し、同じ請求月内に同じリージョンに同じバンドル (bundleId nano_3_0) の 3 つの新しいインスタンスを作成する場合、データ転送使用率は 1 TB のままで、残りのデータ転送許容量は 2 TB のままになります。データ送信の超過料金が発生する前の同じ月内に、インスタンスを介してさらに 2 TB のデータを転送できます。
- 例 6 – 請求月の最初の 20 日間に月間 5 USD のインスタンスバンドル (bundleIdnano_3_0) に対して毎月 1 TB のデータ転送許容量を使用した後、追加の 100 GB のデータをインターネットで送信したとします。この 100 GB にはデータ転送出力超過料金が発生します。同じバンドル (bundleId nano_3_0) の別の新しいインスタンスを今すぐ作成する場合、以前に発生したデータ転送出力超過料金が引き続き課金されます。これらのインスタンスを介した追加のデータ転送出力には、データ転送出力超過料金が引き続き発生します。

インスタンスのデータ転送許容量は、 AWS リージョンによってどれぐらい異なりますか？

Lightsail インスタンスのリージョン別のデータ転送許容量は「Amazon Lightsail の料金表」に記載されています。許容量は、アジアパシフィック (ムンバイ、シドニー、ジャカルタ) リージョンを除き AWS リージョン、すべての で同じです。ムンバイ、シドニー、ジャカルタの各リージョンのプランには、他のリージョンのデータ転送許容量の半分が含まれています。

Lightsail マネージドデータベースのデータ転送許容量は、すべての AWS リージョンで同じです。

データ転送の課金対象を教えてください。

インスタンス、データベース、およびコンテンツ配信ネットワーク (CDN) の配信プランには、データ転送枠が含まれます。

Lightsail インスタンスの場合、インスタンス内へのデータ転送とインスタンス外へのデータ転送の両方が、データ転送枠の利用にカウントされます。データ転送許容量を超えた場合、Lightsailインスタンスからインターネット、またはインスタンスのパブリック IP アドレスを使用する AWS リソースへの超過データ転送に対してのみ課金されます。Lightsail インスタンスへの過剰なデータ転送 IN に対して課金されることはありません。インスタンスのプライベート IP アドレスを使用してデータ転送すると、Lightsail インスタンスのデータ転送の入力および出力はデータ転送クォータにカウントされます。ただし、プライベート IP アドレスを使用して転送されたデータは、データ転送許容量を超えても無料です。

Lightsail マネージドデータベースの場合、データベース外へのデータ転送のみが許容枠の利用にカウントされます。データ転送許容量を超えると、Lightsail マネージドデータベースからインターネットへのデータ転送に対してのみ課金されます。

Lightsail CDN ディストリビューションでは、ディストリビューションからのデータ転送はすべて許容枠の使用にカウントされます。ディストリビューションのデータ転送許容量を超えると、ディストリビューションからのすべてのデータ転送に料金が発生します。

データ転送枠はロードバランサーではどのように機能しますか?

ロードバランサーはデータ転送枠を消費しません。ロードバランサーとターゲットインスタンスおよびディストリビューション間のトラフィックは測定され、インスタンスおよびディストリビューションのデータ転送枠の利用としてカウントされます。これは、ロードバランサーの背後に配置されていない Lightsail インスタンスにおいて、インターネットとのトラフィックの流入がデータ転送枠の利用にカウントされるのと同じです。ロードバランサーとインターネットの間を行き来するトラフィックは、インスタンスのデータ転送枠の利用としてカウントされません。

オブジェクトストレージでのデータ転送許容量の仕組みを教えてください。

データ転送許容量は、以下の場合を除き、データの Lightsail オブジェクトストレージからの転送、またはそれに対する転送によって消費されます。

- インターネットから Lightsail オブジェクトストレージに転送されたデータ
- Lightsail オブジェクトストレージリソース間でのデータ転送
- Lightsail オブジェクトストレージから同じ 内の別のLightsailリソースに転送されたデータ AWS リージョン (別の AWS アカウントのリソースを含むが、同じ AWS リージョン)
- Lightsail オブジェクトストレージから Lightsail CDN ディストリビューションに転送されたデータ

ディストリビューションにはどの種類のデータ転送が課金されますか?

Lightsail CDN ディストリビューションプランのデータ転送許容量を超過した場合は、流出されるすべてのデータ転送に対して課金されます。ディストリビューションの許容量を超えるデータ転送の課金内容は、次の通りです。

- アジアパシフィック: 0.13 米ドル/GB
- カナダ: 0.09 米ドル/GB
- 欧州: 0.09 米ドル/GB
- インド: 0.13 米ドル/GB
- 日本: 0.14 米ドル/GB
- 中東: 0.11 米ドル/GB
- 南アフリカ: 0.11 米ドル/GB
- 南アメリカ: 0.11 米ドル/GB
- 米国: 0.09 米ドル/GB

Lightsail のインスタンスのデータ転送クォータとディストリビューションのデータ転送クォータの違いについて教えてください。

ディストリビューションの許容量を超えるデータ転送の課金は、Lightsail インスタンスと異なります。データ転送の入力と出力はインスタンスのデータ転送クォータにカウントされますが、オリジンおよびビューアへのデータ転送の出力のみがディストリビューションクォータにカウントされます。また、ディストリビューションのクォータを超えたデータ転送の出力はすべて超過料金が課金されますが、インスタンスの一部におけるデータ転送の出力は無料になります。そして、Lightsail ディストリビューションでは異なる地域超過モデルが使用されます。ただし料金の大部分はインスタンスの超過で課金される場合の料金と同じです。

コンテナサービスとのデータ転送は課金されますか?

すべてのコンテナサービスには、データ転送クォータ (月々 500 GB) が付属しています。サービスにおけるデータ転送の入力および出力の両方にカウントされます。クォータを超えると、パブリック IP アドレスを使用する場合、Lightsailコンテナサービスからインターネット、別の AWS リージョン 、または同じリージョンの AWS リソースへのデータ転送に対して課金されます。無料許容量を超過した際のこれらのデータ転送料金は、以下の通りです。

- 米国東部 (オハイオ) (us-east-2): 0.09 米ドル/GB
- 米国東部 (バージニア北部) (us-east-1): 0.09 米ドル/GB
- 米国西部 (オレゴン) (us-west-2): 0.09 米ドル/GB
- アジアパシフィック (ムンバイ) (ap-south-1): 0.13 米ドル/GB
- アジアパシフィック (ソウル) (ap-northeast-2): 0.13 米ドル/GB
- アジアパシフィック (シンガポール) (ap-southeast-1): 0.12 米ドル/GB
- アジアパシフィック (シドニー) (ap-southeast-2): 0.17 米ドル/GB
- アジアパシフィック (東京) (ap-northeast-1): 0.14 米ドル/GB
- カナダ (中部) (ca-central-1): 0.09 米ドル/GB
- 欧州 (フランクフルト) (eu-central-1): 0.09 米ドル/GB
- 欧州 (アイルランド) (eu-west-1): 0.09 米ドル/GB
- 欧州 (ロンドン) (eu-west-2): 0.09 米ドル/GB
- 欧州 (パリ) (eu-west-3): 0.09 米ドル/GB
- 欧州 (ストックホルム) (eu-north-1): 0.09 米ドル/GB

ブロックストレージ (ディスク)

Lightsail ブロックストレージでどのようなことができますか?

Lightsail ブロックストレージは追加のストレージボリューム (Lightsail では「アタッチ済みディスク」と呼ばれます) を提供します。個別のハードドライブのように、Lightsail インスタンスにアタッチすることが可能です。アタッチ済みディスクは、特定のデータをコアサービスから分離する必要のあるアプリケーションやソフトウェアに役立ちます。インスタンスやその他のシステムディスクに障害や不具合が発生した場合に、アプリケーションデータを保護することが可能です。保存されたデータに頻繁にアクセスするアプリケーションやソフトウェアは、一貫したパフォーマンスと低レイテンシーを必要としますが、アタッチ済みディスクはそれを実現します。

Lightsail ブロックストレージディスクにはソリッドステートドライブ (SSD) が使用されています。このタイプのブロックストレージは、コストとパフォーマンスのバランスを重視しており、Lightsail で実行されるワークロードの大部分をサポートすることが目的です。持続的な IOPS パフォーマンスや、ディスクに対し多量のスループットを必要とするアプリケーション、または MongoDB や Cassandra などの大規模データベースを実行するアプリケーションをご利用のお客様には、Lightsail ではなく GP2 やプロビジョンド IOPS SSD ストレージを備えた Amazon EC2 のご使用をお勧めします。

アタッチ済みディスクは Lightsail プランに含まれているストレージとどのように異なりますか?

Lightsail プランに含まれるシステムディスクはインスタンスのルートデバイスです。インスタンスを終了すると、システムディスクも削除されます。インスタンスに障害が発生した場合、システムディスクにも影響が及ぶ可能性があります。またシステムディスクをデタッチしたり、インスタンスと切り離してバックアップすることができません。アタッチ済みディスクに保存されたデータは、インスタンスから独立て存続します。アタッチ済みディスクはデタッチしたり、インスタンス間で移動させることができます。またディスクの手動スナップショットを作成することで、インスタンスから独立してバックアップできます。データ保護のため、 Lightsail インスタンスのシステムディスクは一時データにだけ使用することをお勧めします。より高いレベルの耐久性が必要なデータには、アタッチ済みディスクを使用す、ディスクまたはインスタンスのスナップショットでディスクを定期的にバックアップすることをお勧めします。

アタッチ済みディスクの容量は、どれくらいまで増やせますか?

各アタッチ済みディスクの最大容量は 16 TB で、 Lightsail アカウントにおけるアタッチ済みブロックストレージの最大容量は 20 TB 以下となります。

Lightsail インスタンスごとにアタッチできるディスクの数を教えてください。

Lightsail インスタンスには最大 15 台のディスクをアタッチできます。

1 台のディスクを複数のインスタンスにアタッチすることはできますか?

できません。ディスクは一度に 1 つのインスタンスにだけアタッチできます。

ディスクはインスタンスにアタッチする必要がありますか?

いいえ、ディスクをインスタンスにアタッチしない選択も可能です。ディスクは、アタッチされていない状態でアカウントに残ります。ディスクがインスタンスにアタッチされていなくても、料金の違いはありません。

アタッチ済みディスクの容量を拡張することはできますか?

はい、ディスクの容量を拡張するには、ディスクのスナップショットを取得し、そのスナップショットからより大きいディスクを新規作成します。

Lightsail ブロックストレージは暗号化を提供しますか?

はい、データの安全を維持するため、すべての Lightsail アタッチ済みディスクとディスクスナップショットは保存される際にデフォルトで暗号化されます。その際、Lightsail がユーザーの代わりに管理するキーを使用します。またLightsail は、Lightsail インスタンスとアタッチ済みディスク間を移動するデータの暗号化も行います。

Lightsail ブロックストレージにはどのような可用性を期待できましか?

Lightsail ブロックストレージは、可用性と信頼性が高くなるように設計されています。各アタッチ済みディスクはアベイラビリティーゾーン内で自動的にレプリケートされるので、コンポーネント障害から保護されます。Lightsail ブロックストレージディスクは可用性が 99.99% になるよう設計されています。また Lightsail はディスクスナップショットをサポートしているので、通常のデータバックアップも可能です。

アタッチ済みディスクをバックアップするには、どうすればよいですか?

ディスクの手動スナップショットを作成することで、ディスクをバックアップできます。またインスタンスの手動スナップショットを作成すれば、インスタンス全体とアタッチされたすべてのディスクをバックアップできます。なお、ディスクがアタッチされているインスタンスの自動スナップショットを有効にするとでも、バックアップは可能です。インスタンスにアタッチされたディスクはインスタンスの手動および自動スナップショットに含まれます。

証明書

Lightsail によってプロビジョニングされた証明書はどのように使用できますか?

SSL/TLS 証明書では、ウェブサイトまたはアプリケーションのアイデンティティを確立し、ブラウザとウェブサイトの間の接続を保護します。Lightsail は、ロードバランサーで使用する署名証明書を提供し、ロードバランサーはセキュア AWS ネットワーク経由で検証済みトラフィックをターゲットインスタンスにルーティングする前に SSL/TLS ターミネーションを行います。Lightsail 証明書は、Lightsail ロードバランサーでのみ使用できます。個々の Lightsail インスタンスでは使用できません。

証明書を認証するには、どうすればよいですか?

Lightsail 証明書はドメイン認証のため、ユーザーがウェブサイトのドメインの所有者でありアクセス権を保有していることを証明して、本人確認を行う必要があります。その後、証明書は認証局によってプロビジョニングされます。新しい証明書をリクエストすると、Lightsail で自動的に証明書の検証が試行されます。証明書を自動的に検証できないときは、検証するドメインまたはドメインの DNS ゾーンに CNAME を追加するように Lightsail から求められます。72 時間以内に、現在 DNS ゾーンを管理している場所のいずれか (Lightsail DNS 管理または外部の DNS ホストプロバイダー) に CNAME レコードを追加します。

ドメインを認証できない場合はどうなりますか?

安全上の理由で、ユーザーはドメイン所有者であることを認証できる必要があります。つまり、何らかの理由でユーザまたはユーザーが所属する組織の人間が DNS レコードを追加してユーザーの証明書を認証できない場合、ユーザーは Lightsail で HTTPS 対応のロードバランサーを使用することはできません。

証明書に追加できるドメインおよびサブドメインの数を教えてください。

1 つの証明書には最大 10 個のドメインまたはサブドメインを追加できます。現在、Lightsail はワイルドカードドメインをサポートしていません。

証明書に関連付けられたドメインを変更するには、どうすればよいですか?

証明書に関連付けられたドメインを変更 (追加/削除)する場合は、証明書を再提出してドメインの所有権を再認証する必要があります。証明書管理画面のステップに沿って証明書を再発行し、促しに応じてドメインを追加または削除します。

証明書を更新するには、どうすればよいですか?

Lightsail では、SSL/TLS 証明書のマネージド更新が提供されます。これにより、お客様がアクションを実行しなくても、Lightsail は有効期限が切れる前に証明書の更新を自動的に試みます。Lightsail 証明書はロードバランサーに有効な状態で関連付けられていないと、自動更新されません。

ロードバランサーを削除すると、証明書はどうなりますか?

ロードバランサーが削除された場合、証明書も削除されます。その後、同じドメインに証明書を使用する必要がある場合、新しい証明書をリクエストして検証する必要があります。

Lightsail によって提供された証明書をダウンロードできますか?

いいえ、Lightsail 証明書はお客様の Lightsail アカウントに紐づけられているため、解除して Lightsail 外で使用することはできません。

連絡先とモニタリング通知

通知とは何ですか?

Lightsail でアラームを設定すると、インスタンス、データベース、またはロードバランサーのいずれかのメトリクスが指定した基準値を超えたときに、通知を受けることができます。通知は、Lightsail コンソールに表示されるバナー、指定したアドレスに送信される メール、指定した携帯電話番号に送信される SMS テキストメッセージの形式にすることができます。メールおよび SMS テキストメッセージで通知を受けるには、リソースを監視する各 AWS リージョン で、メールアドレスと携帯電話番号を通知連絡先として追加する必要があります。通知の詳細については、「通知」を参照してください。

連絡先はいくつ追加できますか?

リソースをモニタリングする AWS リージョン ごとに、1 つの E メールアドレスと 1 つの携帯電話番号を追加できます。SMS テキストメッセージングは、Lightsail リソースを作成できるすべての AWS リージョン でサポートされているわけではありません。また、一部の国や地域にテキストメッセージを送信することはできません。通知の詳細については、「通知」を参照してください。

コンテナサービス

Lightsail コンテナサービスでできることは何ですか?

Lightsail コンテナサービスは、コンテナ化されたアプリケーションをクラウドで実行する簡単な方法を提供します。コンテナサービスでは、シンプルなウェブアプリから多階層のマイクロサービスまで、さまざまなアプリケーションを実行できます。コンテナサービスに必要なコンテナイメージ、パワー (CPU、RAM)、スケール (ノード数) を指定するだけです。Lightsailは、基盤インフラストラクチャの管理を有することなく、コンテナサービスを実行します。Lightsail は負荷分散された TLS エンドポイントを提供するので、コンテナサービスで実行しているアプリケーションにアクセスすることができます。

Lightsail コンテナサービスは Docker コンテナを実行できますか?

はい。Lightsail は Linux ベースの Docker コンテナをサポートしています。Windows コンテナは現在サポートされていません。

公開コンテナイメージを Lightsail コンテナサービスで使うには、どうすればよいですか?

Amazon ECR Public Registry などのオンライン公開レジストリからコンテナイメージを使用するか、独自のカスタムイメージを構築して、 AWS CLIを使用した簡単な手順で Lightsail にプッシュすることができます。詳しくは、「コンテナイメージをプッシュして管理する」を参照してください。

プライベートコンテナレジストリからコンテナイメージをプルできますか?

現在、パブリックコンテナレジストリのみが Lightsail コンテナサービスにサポートされています。代案として、カスタムコンテナイメージをローカルマシンから Lightsail にプッシュしすると、非公開の状態にすることができます。

需要に応じてサービスのパワーとスケールを変更することはできますか?

はい。コンテナサービスのパワーとスケールは、サービスの作成後であっても常時変更できます。

Lightsail コンテナサービスによって作成された HTTPS エンドポイントの名前をカスタマイズすることはできますか?

Lightsail は、すべてのコンテナサービスに対する HTTPS エンドポイントを *<service-name>*.*<random-guid>*.*<aws-region-name>*.cs.amazonlightsail.com 形式で提供しま

す。カスタマイズできるのは、サービス名だけです。代案として、カスタムドメイン名を使用することができます。詳細については、「カスタムドメインの有効化と管理」を参照してください。

Lightsail コンテナサービスの HTTPS エンドポイントにカスタムドメインを使用できますか?

はい。カスタムドメイン名を持つ SSL/TLS 証明書を作成し、Lightsail のコンテナサービスにアタッチすることができます。証明書はドメイン検証済みである必要があります。ドメインの DNS がLightsail DNS ゾーンを使用する場合、ドメイン (`example.com`) またはサブドメイン (`www.example.com`) の apex のトラフィックをコンテナサービスにルーティングできます。または、ALIAS レコードの追加をサポートする DNS ホストプロバイダーを使用して、ドメインの apex (`example.com`) を Lightsail コンテナサービスのデフォルトドメイン (パブリック DNS) にマッピングすることもできます。詳細については、「カスタムドメインの有効化と管理」を参照してください。

Lightsail コンテナサービスの料金を教えてください。

Lightsail コンテナサービスはオンデマンドの時間料金で請求されるため、ご利用分のみのお支払いとなります。ご利用の Lightsail コンテナサービスごとに固定の時間料金が課金され、月間最大サービス料金に達するまで適用されます。月間最大サービス料金は、サービスのパワーの基本料金にサービスのスケールを掛けることによって算出できます。例えば、マイクロパワーとスケールが 2 のサービスの場合、最大 10 USD*2=20 USD/月のコストがかかります。最も安い Lightsail コンテナサービスは、0.0094 USD/時 (7 USD/月) となります。各サービスについて、月間 500 GB の無料クォータを超える使用については、追加データ転送料金が請求される場合があります。詳細については、「コンテナサービスとのデータ転送は課金されますか?」を参照してください。

コンテナサービスを数日しか実行なくても、1 か月分が請求されますか?

あなたのLightsailコンテナサービスは、実行中または無効状態のときにのみ、課金されます。Lightsail コンテナサービスを月末前に削除した場合は、Lightsail コンテナサービスを使用した合計時間に基づいて比例配分された料金が請求されます。例えば、パワーがマイクロでスケール 1の Lightsail コンテナサービスを 1 か月で 100 時間使用した場合、1.34 USD (0.0134 USD *100) が課金されます。

コンテナサービスの停止と削除の違いは何ですか?

コンテナサービスを無効にすると、コンテナノードは無効状態となり、サービスのパブリックエンドポイントは HTTP ステータスコード「503」を出力します。サービスを有効にすると、最後にアク

ティブだったデプロイが復元されます。パワーとスケールの設定も保持されます。パブリックエンドポイントの名前は、再有効にした後も変更されません。デプロイ履歴とコンテナイメージは保持されます。

コンテナサービスの削除は、破壊的なアクションです。サービスのすべてのコンテナノードは永久的に削除されます。サービスと関連付いている HTTPS パブリックエンドポイントのアドレス、コンテナイメージ、デプロイ履歴、ログも永久的に削除されます。エンドポイントアドレスを復元することはできません。

コンテナサービスが無効状態でも、課金されますか?

はい。コンテナサービスが無効な状態であっても、コンテナサービスのパワーとスケールの設定に従って課金されます。

コンテナサービスを Lightsail コンテンツ配信ネットワーク (CDN) ディストリビューションのオリジンとして使用することはできますか?

コンテナサービスは現在、Lightsail CDN ディストリビューションのオリジンとしてサポートされていません。

コンテナサービスを Lightsail ロードバランサーのターゲットとして使用できますか?

いいえ。 現在、コンテナサービスは Lightsail ロードバランサーのターゲットとして使用することができません。ただし、コンテナサービスのパブリックエンドポイントにはビルトインロードバランシングが備わっています。

HTTP リクエストを HTTPS にリダイレクトするよう、コンテナサービスのパブリックエンドポイントを設定できますか?

Lightsail コンテナサービスのパブリックエンドポイントは、すべての HTTP リクエストを自動的に HTTPS にリダイレクトするので、ユーザーのコンテンツは安全に配信されます。

コンテナサービスはモニタリングととアラートをサポートしていますか?

コンテナサービスは、サービスのノード全体の CPU 使用率とメモリ使用率に関するメトリクスを表示します。これらのメトリクスに基づくアラートは、現在サポートされていません。

Lightsail コンテナサービスは IPv6 をサポートしていますか?

Lightsail コンテナサービスの HTTPS エンドポイントは、IPv4 と IPv6 の両方をサポートしています。コンテナサービスでは Pv6 を無効にすることはできません。

コンテンツ配信ネットワークディストリビューション

Lightsail CDN ディストリビューションでできることは何ですか?

Lightsail コンテンツ配信ネットワーク (CDN) ディストリビューションを利用すると、Amazon CloudFront を利用したアマゾンのグローバルデリバリーネットワークにコンテンツを保存、供給することで、ユーザーの Lightsail リソースにホストされたコンテンツの配信を加速させることが容易になります。またディストリビューションでは、簡単な SSL 証明書の作成とホスティングが可能なので、ユーザーのウェブサイトにおける HTTPS トラフィックのサポートをするのにも役立ちます。そして、ディストリビューションは Lightsail リソースへの負担を軽減し、大規模なトラフィックの急増にもユーザーのウェブサイトが対応できるようにします。Lightsail のすべての特徴と同様に、数回のクリックだけでセットアップは完了します。支払いはシンプルな月額料金となります。

ディストリビューションのオリジンとして、どのような種類のリソースを使用できますか?

Lightsail ディストリビューションでは、Lightsail インスタンスとロードバランサーをオリジンとして使用することができます。Lightsail コンテナは現在、オリジンとしてサポートされていません。S3 バケットなどの Lightsail 外のリソースはサポートされていません。

インスタンスを Lightsail ディストリビューションのオリジンに設定する場合、静的 IPv4 アドレスを Lightsail インスタンスにアタッチする必要はありますか ?

はい。静的 IPv4 アドレスは、オリジンとして指定されたインスタンスにアタッチする必要があります。現在、Lightsail ディストリビューションは IPv6 をサポートしていません。

WordPress のウェブサイトで Lightsail ディストリビューションを設定するには、どうすればよいですか？

ディストリビューションを作成し、オリジンとして WordPress インスタンスを選択し、プランを決めたら設定が完了します。Lightsail ディストリビューションでは、ディストリビューション設定が自動的に構成され、ほとんどの WordPress 設定でパフォーマンスが最適化されます。

複数のオリジンをアタッチできますか?

複数のオリジンを Lightsail ディストリビューションにアタッチすることはできませんが、複数のインスタンスを Lightsail ロードバランサーにアタッチして、それをディストリビューションのオリジンとして指定することは可能です。

Lightsail ディストリビューションは証明書の作成をサポートしていますか？

はい。Lightsail ディストリビューションでは、ディストリビューションの管理ページから直接、証明書の作成、検証、アタッチが容易にできます。

証明書は必要ですか?

カスタムドメイン名をディストリビューションで使用する場合に、証明書が必要となります。すべての Lightsail ディストリビューションは、HTTPS が有効な 独自 Amazon CloudFront ドメイン名と共に作成されます。しかしカスタムドメインをディストリビューションで使用する場合は、カスタムドメインの証明書をディストリビューションにアタッチする必要があります。

作成できる証明書の数に制限はありますか?

はい。詳細については Lightsail Service Quotas を参照してください。

HTTP リクエストを HTTPS にリダイレクトするようにディストリビューションを設定するには、どうすればよいですか?

Lightsail ディストリビューションは自動的にすべての HTTP リクエストを HTTPS にリダイレクトするので、ユーザーのコンテンツは安全に配信されます。

apex ドメインが Lightsail ディストリビューションを指すように設定するには、どうすればよいですか？

apex ドメインを CDN ディストリビューションに向けるには、ディストリビューションのデフォルトドメインに apex ドメインをマッピングするドメインのドメインネームシステム (DNS) に、 エイリアスレコードを作成する必要があります。DNS ホスティングプロバイダーが エイリアスレコードをサポートしていない場合は、Lightsail DNS ゾーンを使用するとディストリビューションのドメインへの Apex ドメインの指定が、簡単に設定できます。

ディストリビューションと関連付いているプランを変更することはできますか?

はい。1 か月に 1 回ディストリビューションのプランを変更できます。2回目のプラン変更をご希望の場合は、翌月になるまでお待ちいただきます。

自分のディストリビューションが機能しているかどうか、どうすれば分かりますか?

Lightsail ディストリビューションには、ディストリビューションのパフォーマンスを追跡するさまざまなメトリクスが備わっています。ディストリビューションが受け取ったリクエストの合計数や、ディストリビューションがクライアントおよびオリジンに送信したデータ量、エラーになったリクエストの割合などが用確認ただけます。さらに、ディストリビューションメトリクスにリンクしたアラートも作成できます。

Lightsail ディストリビューションのキャッシュされたコンテンツを削除できますか？

キャッシュされたコンテンツはすべて削除できますが、削除できない特定のファイルやフォルダがあります。

どのような場合に、Amazon CloudFront ディストリビューションではなく Lightsail ディストリビューションを使うべきですか？

Lightsail ディストリビューションは、インスタンスやロードバランサーなどの Lightsail リソースでウェブサイトやウェブアプリケーションを運営するユーザー向けに設計されています。で別のサービスを使用してウェブサイトやアプリケーションを AWS ホストしている場合、複雑な設定ニーズがあ

る場合、または 1 秒あたりのリクエスト数が多いか、ビデオストリーミング量が多いワークロードがある場合は、Amazon CloudFront を使用することをお勧めします。

Lightsail コンテンツ配信ネットワークディストリビューションを Amazon CloudFront に移動することはできますか?

はい。同様に設定したディストリビューションを Amazon CloudFront に作成することで、Lightsail ディストリビューションを移動できます。Lightsail ディストリビューションで設定できるすべての設定は、CloudFront ディストリビューションでも設定できます。ディストリビューションを CloudFront に移動するには、以下のステップを実行してください。

Lightsail ディストリビューションを CloudFront に移動する方法

- ディストリビューションのオリジンとして設定されている Lightsail インスタンスのスナップショットを作成する。スナップショットを Amazon EC2 にエクスポートし、Amazon EC2 のスナップショットから新しいインスタンスを作成します。詳細については、「スナップショットを Amazon EC2 にエクスポートする」を参照してください。

  > **Note**
  >
  > ウェブサイトまたはウェブアプリケーションをロードバランスする必要がある場合は、Elastic Load Balancing に Application Load Balancer を作成します。詳細については、「Elastic Load Balancing ユーザーガイド」を参照してください。

- Lightsail ディストリビューションのカスタムドメインを無効にして、使用していた可能性のある証明書をデタッチします。詳細については、Amazon Lightsail ディストリビューションのカスタムドメインを無効にするを参照してください。
- AWS Command Line Interface (AWS CLI) を使用して get-distributions コマンドを実行し、Lightsailディストリビューションの設定のリストを取得します。詳細については、「AWS CLI リファレンス」の「get-distributions」を参照してください。
- CloudFront コンソールにサインインしてお客様の Lightsail ディストリビューションと同じ環境設定のディストリビューションを作成します。詳細については、「Amazon CloudFront デベロッパーガイド」の「ディストリビューションの作成」を参照してください。
- CloudFront ディストリビューションにアタッチする証明書を AWS Certificate Manager c (ACM) で作成します。詳細については、「ACM ユーザーガイド」の「パブリック証明書のリクエスト」を参照してください。

- CloudFront ディストリビューションを更新して、作成した ACM 証明書を使用します。詳細については、「CloudFront ユーザーガイド」の「CloudFront ディストリビューションの更新」を参照してください。

Lightsail CDN はどのような目的で使用されますか？

Lightsail CDN ディストリビューションは固定料金のデータ転送バンドルを使用して作成されるため、サービスの使用コストが単純かつ予測可能になります。ディストリビューションバンドルは、1 か月分相当の使用量をカバーするように設計されています。ディストリビューションバンドルを超過料金の発生を防ぐような方法 (バンドルの頻繁なアップグレードまたはダウングレード、または異常に多量なディストリビューションを一つのオリジンで使用するなどを含むが、これらに限らない方法) で使用することは、本来の使用目的の範囲を超えるため、許可されていません。さらに、1 秒あたりのリクエスト数が多いワークロードや、大量のビデオストリーミングを伴うワークロードは許可されません。これらの動作に従事すると、データサービスまたはアカウントがスロットリングされたり停止される可能性があります。詳細については、「データ転送の課金対象を教えてください。」を参照してください。

Lightsail CDN ディストリビューションは IPv6 をサポートしていますか?

すべての Lightsail CDN ディストリビューションでは、デフォルトで IPv6 が有効になっています。ディストリビューションのホスト名は IPv4 アドレスと IPv6 アドレスの両方に解決されます。IPv6 は、CDN の管理ページにあるネットワークタブのトグルから無効にすることができます。

Lightsail CDN ディストリビューションで動作させるには、オリジンを IPv6 有効にする必要がありますか?

いいえ。 CDN ディストリビューションは IPv6 と IPv4 の両方のトラフィックを受け入れ、バックエンドのオリジンと通信するときにはシームレスに IPv4 へ変換します。したがって、オリジンの背後にあるディストリビューションはデュアルスタックまたは IPv4 のみの、いずれも可能です。

データベース

Lightsail マネージド型データベースとは何ですか？

Lightsail マネージドデータベースは、ウェブサーバーやメールサーバーなどの他のワークロードではなく、データベースの実行のみに特化したインスタンスです。マネージドデータベースには、ユー

ザーが作成した複数のデータベースを含むことができます。このデータベースには、スタンドアロンデータベースで使用する同じツールやアプリケーションを使ってアクセスできます。Lightsail がデータベースの基盤となるインフラストラクチャやオペレーティングシステムのセキュリティとヘルスを管理するため、インフラストラクチャ管理の深い専門知識がなくてもデータベースを実行することができます。

通常の Lightsail インスタンスと同様に、Lightsail マネージドデータベースには必要に応じてスケールアップできる固定量のメモリ、処理能力、SSD ベースのストレージがプランに含まれています。Lightsail は作成後、選択したデータベースを自動的にインストールして設定します。

Lightsail マネージド型データベースでできることは何ですか？

Lightsail マネージドデータベースは、手のかからない簡単な方法でデータをクラウドに保存することができます。マネージドデータベースは新しいデータベースとして実行できまし、既存のオンプレミスやホストされたデータベースから Lightsail に移行して実行することもできます。

マネージドデータベースはアプリケーションをスケールすることもできます。データベースをハードウェア専有インスタンスに分離することで、より多くのトラフィックと激しい負荷の受け入れが可能になります。Lightsail マネージドデータベースは特に、単体のインスタンスを超えたスケールの際にデータの同期が必要となるステートフル アプリケーション (WordPress やい一般的な CMC など) に有効です。マネージドデータベースは Lightsail ロードバランサー、そして2つ以上の Lightsail インスタンスと組み合わせることができるので、強力なスケールされたアプリケーションが作成できます。Lightsail の高可用性マネージドデータベースプランを使用すると、データベースに冗長性を加えることもできるので、アプリケーションの高いアップタイムが確保できます。

Lightsail の管理対象を教えてください。

Lightsail はマネージドデータベースとその基盤となるインフラストラクチャに対し、さまざまなメンテナンス作業とセキュリテ管理を行います。Lightsail はデータベースを自動バックアップします。データベース復元ツールを使うと過去 7 日間の特定の時点の復元ができるので、データ損失やコンポーネントの障害から保護してくれます。またLightsail はセキュリティ強化のため、保管中のデータや移動中のデータを自動暗号化します。データベースのパスワードは保存されるので、簡単かつ安全にデータベースに接続できます。メンテナンスにおいては、Lightsail はユーザーが設定したメンテナンスウィンドウの期間でデータベースのメンテナンスを実行します。このメンテナンスには、最新のマイナーデータベースバージョンへの自動アップグレードと、基盤となるインフラストラクチャおよびオペレーティングシステムの全面的な管理が含まれます。

Lightsail がサポートしているデータベースの種類とバージョンを教えてください。

Lightsail マネージドデータベースは、MySQL および PostgreSQL の最新のメジャーバージョンをサポートしています。現在、これらのバージョンは MySQL 5.7、MySQL 8.0、PostgreSQL 9、PostgreSQL 10、PostgreSQL 11、および PostgreSQL 12 です。Lightsail は、各メジャーバージョンのオプションのために、最新のマイナーバージョンのみをサポートしています。

Lightsail はどのようなマネージド型データベースプランを提供していますか？

Lightsail は、標準および高可用性プランで 6 サイズのマネージドデータベースを提供します。各プランには固定のストレージ容量と月間データ転送許容枠が付いています。しばらくしてから必要に応じてより大きなプランにスケールアップしたり、スタンダードプランと高可用性プランを切り替えたりすることもできます。高可用性プランにはスタンダードプランと同じリソースが含まれるほかに、プライマリデータベースとは別のアベイラビリティーゾーンで実行されるスタンバイデータベースが含まれているので、冗長性に富んでいます。データ転送費用の詳細については、「データ転送の課金対象を教えてください。」を参照してください。

高可用性プランとは何ですか?

Lightsail マネージドデータベースは、スタンダードプランおよび高可用性プランでお求めいただけます。スタンダードプランと高可用性プランには、メモリやストレージ、データ転送許容枠など、同じプランリソースが含まれています。高可用性プランでは、データベースに冗長性と耐久性が追加されます。プライマリデータベースとは別のアベイラビリティーゾーンに自動的にスタンバイデータベースが作成され、同時にスタンバイデータベースにはデータがレプリケートされます。そのためインフラストラクチャ障害やメンテナンスの際、スタンバイデータベースにフェイルオーバーされます。これにより、Lightsail が自動的にデータベースのアップグレードやメンテナンスを開始した際でも、アップタイムが確保されます。高可用性プランは、高いアップタイムが要求されるプロダクション用のアプリケーションやソフトウェアを実行する場合に使用します。

Lightsail マネージド型データベースをスケールアップおよびスケールダウンするには、どうすればよいですか？

Lightsail マネージドデータベースをスケールアップするには、スナップショットを作成し、そのスナップショットからより大きい新規データベースプランを作成します。または、緊急復元機能を使用してより大きな新規データベースを作成します。また、これらのいずれかの方法でスタンダードプラ

ンと高可用性プランを切り替えることも可能です。データベースをスケールダウンすることはできません。詳細については、Lightsail でスナップショットからデータベースを作成するを参照してください。

Lightsail マネージド型データベースをバックアップするには、どうすればよいですか？

Lightsail は自動的にデータをバックアップするので、過去の特定の時点のデータから新しいデータベースを復元できます。自動バックアップはデータベースの無料サービスですが、過去 7 日分のデータしか保存されません。データベースを削除すると、すべての自動バックアップレコードが削除されるので、特定の時点への復元ができなくなります。データベース削除後にデータのバックアップを保持したり、過去7日以前のバックアップを保持したい場合は、手動スナップショットを使用します。

Lightsail マネージド データベースの手動スナップショットは、データベース管理ページから作成できます。手動スナップショットにはデータベース内のすべてのデータが含まれるので、永続的に保存したいデータのバックアップとして使用できます。手動スナップショットを使用して、より大きな新規データベースを作成したり、スタンダートプランと高可用性プランを切り替えたりすることもできます。手動スナップショットは削除するまで保存され、0.05 USD/GB (毎月) が請求されます。

Lightsail マネージド型データベースを削除すると、データはどうなりますか？

Lightsail マネージドデータベースを削除すると、データベースとすべての自動バックアップの両方が削除されます。データベースを削除する前に手動スナップショットを作成した場合を除き、このデータを復元する方法はありません。データベースを削除する際に、Lightsail は手動スナップショットを作成するためのワンクリックオプションを表示するので、必要に応じて偶発的なデータの損失を防ぐことができます。削除前の手動スナップショット作成は任意となりますが、強くお勧めします。手動スナップショットは、保存したデータが不要になった時点で削除できます。

異なるアベイラビリティーゾーン AWS リージョン または異なるアベイラビリティーゾーンで実行されているLightsailマネージドデータベースにインスタンス (複数可) を接続することはできますか?

Lightsail マネージド型データベースは、異なる AWS リージョンで実行されているインスタンスとは使用できません。ただし、ユーザーのインスタンスとは異なるアベイラビリティーゾーンのデータベースは使用いただけます。

Lightsail マネージド型データベースにデータをロードするには、どうすればよいですか？

Lightsail マネージドデータベースにデータをロードするには、まずデータのインポートモードを有効にする必要があります。データのインポートモードを有効にすると、お好みのデータベースクライアントを使用してデータを手動でアップロードできます。データのロードが完了したら、必ずデータのインポートモードをオフにし、データベースの自動バックアップとログ記録が再開されるようにしてください。詳細については、「MySQL データベースにデータをインポートする」および「PostgreSQL データベースにデータをインポートする」を参照してください。

Lightsail マネージド型データベースのデータにアクセスするには、どうすればよいですか？

一般的な SQL クライアントアプリケーションを使用してデータベースに接続し、データをクエリできます。GUI ベースの管理とクエリには MySQL Workbench をお勧めします。エンドポイント URL や DNS 名などの接続データは、データベース内のデータベース管理画面で確認できます。詳細については、「MySQL データベースに接続する」または「Amazon Lightsail で PostgreSQL データベースに接続する」を参照してください。

Lightsail マネージド型データベースは Lightsail インスタンスとどのように連携しますか?

作成した Lightsail マネージドデータベースは、アプリケーションですぐに使い始めることができます。Lightsail インスタンスは、ユーザーのアプリのウェブサーバーまたは専用ワークロードとして使用されます。Lightsail インスタンスをデータベースに接続するには、データベースエンドポイントを使用し、安全に保存されているパスワードを参照して、データベースをデータストアとしてアプリケーションコード内に設定します。接続データはデータベース管理画面で確認できます。データベース設定ファイルのファイル名とロケーションはアプリケーションによって異なります。なお、同じテーブルまたは別のテーブルを使用して、複数のインスタンスを 1 つのデータベースに接続することが可能です。

AWS アカウントで実行されている EC2 インスタンスにLightsailマネージドデータベースを接続するにはどうすればよいですか?

パブリックインターネットを介して接続することで、Lightsail マネージドデータベースを EC2 インスタンスに接続できます。すべての AWS サービスへの接続では、データベースのデータ転送許容量が使用され、データ転送許容量を超える AWS サービスへのパブリックインターネット経由でのデー

タ出力には超過料金が発生することに注意してください。Lightsail マネージドデータベースと EC2 インスタンスの間で VPC ピアリングは使用できません。データ転送費用の詳細については、「データ転送の課金対象を教えてください。」を参照してください。

Lightsail マネージド型データベースのパブリックモードとプライベートモードの違いは何ですか？

デフォルトでは、Lightsail マネージドデータベースはプライベートモードで作成されます。Lightsail インスタンスからのアクセスのみが許可されるため、保護された状態です。パブリックインターネットを介してソフトウェアやサービスに接続する必要がある場合は、データベースをパブリックモードに設定します。データの安全性を維持するため、パブリックモードを長期的に有効にしておくことはお勧めしません。パブリックモードとプライベートモードは、データベース管理画面からいつでも切り替えることができます。

Lightsail マネージド型データベースが使用するポートを管理できますか？

できません。Lightsail はセキュリティ上の理由でポートを自動的に管理します。パブリックモードのすべての Lightsail マネージドデータベースに対して MySQL 用のポート 3306 を開放します。データベースがプライベートモードの場合は、 Lightsail アカウントで実行されているリソースに対してのみ、内部ネットワーク経由でデータベースが開放されます。

Lightsail マネージドデータベースサービスは IPv6 をサポートしていますか?

Lightsail マネージドデータベースサービスは IPv6 をサポートしていません。

ドメイン

Lightsail ドメインでは何ができますか？

Lightsail ドメインを使用すると、ウェブサイトまたはアプリケーションのドメインを登録および管理できます。他のプロバイダーに登録されているドメインがある場合は、それらのドメインの管理を Lightsail に移管できます。これらのドメインを Lightsail リソースにポイントさせることもできます。

どの最上位ドメイン (TLD) を使用できますか?

Lightsail は、Amazon Route 53 と同じ汎用 TLD を使用します。地理的ドメインを登録する場合は、Route 53 コンソールを使用することをお勧めします。地理的ドメインは、Route 53 を使用して

登録してから Lightsail コンソールで使用できるようになります。Lightsail がサポートしている TLD の詳細については、Amazon Route 53 デベロッパーガイドの「Amazon Route 53 に登録できるドメイン」を参照してください。

Lightsail を既存ドメインの DNS サービスにできますか？

別の DNS サービスプロバイダーを使用して登録したドメインの DNS 管理を Lightsail に移管できます。詳細については、「DNS ゾーンを作成してドメインの DNS レコードを管理する」を参照してください。

Lightsail のドメイン登録の使用を開始するには、どうすればよいですか？

Lightsail にログインすると、Lightsail コンソールを使用してドメインを作成および管理できます。詳細については、「ドメインの登録」を参照してください。

どのような場合に Route 53 ではなく Lightsail にドメインを登録すればよいですか？

ドメインの登録、DNS ゾーンの作成、ドメインのトラフィックの Lightsail リソースへのルーティングなどのタスクは Lightsail で実行されます。ドメイン登録の延長、トラフィックポリシーを含むドメインの移管、プライベートホストゾーンの作成などの高度なタスクには Route 53 を使用することをお勧めします。

ドメインを Lightsail に移管できますか？

ドメインは Route 53 に移管できます。ドメイン移管が完了すると、ドメインは Lightsail コンソールで使用できるようになります。詳細については、「Amazon Route 53 での Lightsail ドメインの管理」を参照してください。

ドメインではどの Lightsail リソースを使用できますか？

Lightsail にドメインを登録すると、ドメインが Lightsail インスタンス、コンテナ、ロードバランサー、静的 IP、またはコンテンツ配信ネットワーク (CDN) を指すようにできます。

Amazon Elastic Compute Cloud (Amazon EC2) に Lightsail リソースをエクスポートする

Amazon EC2 へのエクスポートとは

Amazon EC2 へのエクスポートとは、Amazon EC2 で Lightsail のインスタンスのコピーを作成する機能です。Amazon EC2 にエクスポートすると、Amazon EC2 が提供する様々なインスタンスタイプ、環境設定、および料金モデルから選択できます。ネットワーキングやストレージ、コンピューティング環境をより細かく調整することが可能になります。

Amazon EC2 にエクスポートする理由は何ですか?

Lightsail では、バンドルされた予測可能な低料金で様々なクラウドベースのアプリケーションを簡単に実行、スケールできます。また Lightsail では、ネットワーキングやアクセス管理などのクラウド環境設定を自動的にセットアップできます。

Amazon EC2 にエクスポートすることで、様々なインスタンスタイプでのアプリケーションの実行が可能になります。より強力な CPU パワーやメモリ、またネットワーキング機能を備えた仮想マシンから、FPGA や GPU を備えた特殊化および高速化されたインスタンスまで、多岐にわたります。さらに、Amazon EC2 では自動で実行される管理やセットアップが少ないため、VPC などのクラウド環境の設定をより細かく調整できます。

Amazon EC2 へのエクスポートの仕組みを教えてください。

まず、Lightsail インスタンスまたはブロックストレージディスクの手動スナップショットをエクスポートする必要があります。Amazon EC2 の使用に慣れているユーザーであれば、Amazon EC2 作成ウィザードまたは API を使用して、新しい Amazon EC2 インスタンスまたは Amazon EBS ボリュームを作成できます。これらは、既存の EC2 AMI や EBS ボリュームから作成されます。また Lightsail はガイド付きの Lightsail コンソール エクスペリエンスも提供しており、新しい EC2 インスタンスを作成するのに便利です。

> **Note**
>
> cPanel & WHM (CentOS 7) インスタンスのスナップショットは Amazon EC2 にエクスポートできません。

どのように請求されますか?

Amazon EC2 へのエクスポート機能の使用は無料です。手動スナップショットを Amazon EC2 にエクスポートすると、Lightsail 手動スナップショットに加えて、Amazon EC2 イメージに対する料金が個別に請求されます。また新しい Amazon EC2 インスタンスを起動した場合にも、Amazon EC2 によって課金されます。これには、インスタンスの Amazon EBS ストレージボリュームやデータ転送が含まれます。新しいインスタンスおよびリソースの料金の詳細については、「Amazon EC2 の料金ページ」を参照してください。Lightsail アカウントで引き続き実行される Lightsail リソースは、削除されるまで通常料金が継続的に課金されます。

マネージドデータベースやディスクのスナップショットはエクスポートできますか?

エクスポート機能では、Lightsail ディスクの手動スナップショットはエクスポートできますが、マネージドデータベースの手動スナップショットのエクスポートは現在サポートされていません。ディスクスナップショットは、Amazon EC2 コンソールまたは API から、Amazon EBS ボリュームとして復元できます。

どのような Lightsail リソースをエクスポートできますか ?

Lightsail の Amazon EC2 へのエクスポート機能は、Linux および Windows インスタンスのスナップショットを Amazon EC2 にエクスポートすることをサポートするための設計を施されています。またこれは、Amazon EBS へのブロックストレージディスクのスナップショットのエクスポートもサポートしています。データベース、コンテナサービス、コンテンツ配信ネットワーク (CDN) ディストリビューション、ロードバランサー、静的 IP、および DNS レコードのエクスポートは現在サポートされていません。さらに、現時点では Django、 Ghost、および cPanel と WHM の インスタンススナップショットは Amazon EC2 にエクスポートすることはできません。

インスタンス

Lightsail インスタンスとは何ですか?

Lightsail インスタンスは、 AWS クラウド内に存在する仮想プライベートサーバー (VPS) です。Lightsail インスタンスを使用して、データの保管やコードの実行、ウェブベースのアプリケーションやウェブサイトを構築したりできます。インスタンスは、パブリック (インターネット) とプライベート (VPC) のネットワーキングを通じて相互にも、他の AWS リソースにも接続できます。Lightsail コンソールからインスタンスの作成、管理、接続が簡単に行えます。

Lightsail プランとは何ですか?

バンドルとも呼ばれている Lightsail プランは、固定量のメモリ (RAM)、コンピューティング (vCPUs)、SSD ベースのストレージ (ディスク)、および無料データ転送枠を備えた仮想サーバーを含んでいます。また Lightsail プランは静的IPv4アドレスとDNS 管理を提供します。Lightsail プランは1時間ごとのオンデマンドで課金されるため、プランの使用分に対してのみお支払いいただきます。データ転送費用の詳細については、「データ転送の課金対象を教えてください。」を参照してください。

インスタンスでは何のソフトウェアが実行できますか?

Lightsail には多様なオペレーティングシステムおよびアプリケーションテンプレートが用意されており、新しい Lightsail インスタンスを作成した際、自動的にインストールされます。アプリケーションテンプレートはWordPress、WordPress Multisite、cPanel & WHM、PrestaShop、Django、Drupal、Ghost、Joomla!、Magento、Redmine、LAMP、Nginx (LEMP)、MEAN、Node.js などを含んでいます。

ブラウザ内の SSH または独自の SSH クライアントを使用して、インスタンスに追加のソフトウェアをインストールすることが可能です。

Lightsail は、どのオペレーティングシステムで使用できますか?

Lightsail は現在 7 つの Linux または Unix-like ディストリビューション (AlmaLinux OS 9、Amazon Linux 2、Amazon Linux 2023、CentOS、Debian、FreeBSD、OpenSUSE と Ubuntu) および 3 つの Windows Server のバージョン (2016、2019 と 2022) をサポートしています。

Lightsail インスタンスを使用するには自分のライセンスを持ち込む必要がありますか ?

Lightsail で利用可能なすべてのインスタンスブループリントには、cPanel と WHM ブループリントを除くライセンスが含まれています。そのブループリントには 15 日間の試用ライセンスが含まれています。詳細については、「クイックスタートガイド: Amazon Lightsail 上の cPanel と WHM」を参照してください。他のすべてのインスタンスブループリントでは、Bring-Your-Own-License (BYOL) は必要ありません。

Lightsail インスタンスを作成するには、どうすればよいですか?

Lightsail にログイン後、Lightsail コンソール、コマンドラインインターフェイス (CLI)、または API を使用してインスタンスの作成および管理ができます。

コンソールへの初回ログインの際に、インスタンスの作成を選択します。インスタンスの作成ページでは、ソフトウェア、ロケーション、およびインスタンスの名前が選択できます。作成 を選択すると、数分以内に新しいインスタンスが自動的にスピンアップします。

Lightsail インスタンスは、どのように動作しますか?

Lightsail インスタンスは、ウェブサーバー、開発者環境、小規模なデータベースのユースケース AWS 向けに によって特別に設計されています。このようなワークロードは、常時または継続してCPU をフルに使用するわけではありませんが、時としてパフォーマンスバーストを必要とします。Lightsail はバーストパフォーマンスインスタンスを使用しているため、ベースラインレベルの CPU パフォーマンスに加え、ベースラインレベルを超えたバースト能力も提供します。この設計により、必要なときに必要なパフォーマンスを得ることが可能です。その一方で、他環境のオーバーサブスクリプションによって引き起こされがちなパフォーマンスの変動やそのほかの副作用からユーザーを保護します。

動画エンコーディングや HPC アプリケーションなどのアプリケーションのため、一貫して高い CPU パフォーマンスを有する高度な環境設定とインスタンスが必要な場合は、Amazon EC2 を使用することをお勧めします。

インスタンスがバーストしているかどうか、どうやって確認できますか?

インスタンスの CPU 使用率メトリクスグラフに、持続可能領域とバースト領域が表示されます。Lightsail 持続可能領域内だと、インスタンスはシステムのオペレーションに影響を与えることなく無制限で作動することができます。負荷が高い場合、インスタンスはバースト領域で作動し始める可能性があります。バースト領域で作動している間、インスタンスは大量の CPU サイクルを消費しています。したがって、この領域では限られた期間しか作動できません。詳細については、「Amazon Lightsail のインスタンスメトリクスの表示」を参照してください。

メトリクスアラームを追加して、インスタンスの CPU 使用率が持続可能領域からバースト領域に移動した際に通知を受けます。詳細については、「Amazon Lightsail でのインスタンスメトリクスアラームの作成」を参照してください 。

Lightsail インスタンスへ接続するには、どうすればよいですか?

Lightsail では、ブラウザからインスタンスのターミナルにワンクリックで安全に接続できます。Linux/Unix ベースのインスタンスへの SSH アクセスおよび Windows ベースのインスタンスへの RDP アクセスがサポートされています。ワンクリック接続を利用するには、インスタンス管理画面を起動し、SSH を使用して接続 または RDP を使用して接続 を選択します。新しいブラウザウィンドウが開き、インスタンスに自動接続されます。

独自のクライアントを使用して Linux/Unix ベースのインスタンスに接続する場合は、Lightsail によってSSH キーの格納および管理作業が実行されます。SSH クライアントで使用できる安全なキーが提供されます。

インスタンスをバックアップするには、どうすればよいですか?

データをバックアップする場合は、Lightsail コンソールまたは API を使用してインスタンスの手動スナップショットを作成するか、自動スナップショットを有効にして Lightsail にスナップショットを毎日作成させることができます。障害やコードのデプロイに不具合が発生した場合は、インスタンスのスナップショットを後から使用して、新しいインスタンスを作成することができます。詳細については、「スナップショット」を参照してください。

プランをアップグレードできますか?

はい。インスタンスのスナップショットを使用して、より大きいサイズの新しいインスタンスを作成できます。詳細については、「スナップショット」を参照してください。

AWS アカウントの他のリソースにLightsailインスタンスを接続するにはどうすればよいですか?

VPC ピアリングを使用して、Lightsailインスタンスを AWS アカウントの Amazon VPC リソースにプライベートに接続できます。Lightsail アカウントページでVPC ピア接続を有効にするを選択するだけで、Lightsail が接続を実行します。VPC ピアリングを有効にすると、プライベート IPs を使用してデフォルトの Amazon VPC 内の他の AWS リソースに対処できます。こちらから手順を確認いただけます。

> **Note**
>
> 2013 年 12 月より前に AWS 作成されたアカウントにはデフォルトの VPC がないため、 との VPC ピアリングを機能Lightsailさせるには、 AWS アカウントにデフォルトの Amazon VPC を設定する必要があります。デフォルト VPC のセットアップに関する詳細は、こちら。

インスタンスの停止と削除の違いは何ですか?

インスタンスを停止すると、現在の状態で電源がオフになり、いつでも再開することができます。インスタンスを停止すると、そのパブリック IPv4 は解放されるため、停止して再開した後も同じIP を

保持する必要のあるインスタンスには、静的 IP を使用することをお勧めします。インスタンスにアタッチされているパブリック IPv6 アドレスは、インスタンスを停止して再開しても変更されないので注意してください。

インスタンスの削除は、破壊的なアクションです。インスタンスのスナップショットを作成していない限り、すべてのインスタンスデータが失われ、復元できなくなります。自動スナップショットも、手動スナップショットとしてコピーして保持しない限り、インスタンスと共に削除されます。インスタンスのパブリック IP とプライベート IP アドレスも解放されます。インスタンスで静的 IPv4 アドレス を使用していた場合、その静的 IP アドレス はデタッチされますが、アカウントには残ります。

ロードバランサー

Lightsail ロードバランサーでどのようなことができますか?

Lightsail ロードバランサーは、可用性が高いウェブサイトとアプリケーションの構築を可能にします。異なるアベイラビリティーゾーン内のインスタンスにトラフィックを分散し、正常なターゲットインスタンスにのみトラフィックを送ることで、Lightsail ロードバランサーはインスタンスの不具合やデータセンターの停止などによってアプリケーションがダウンするリスクを軽減します。Lightsail ロードバランサーとマルチターゲットインスタンスを使用することで、お客様のウェブサイトやアプリケーションはウェブトラフィックの増加に対応することが可能となり、ロード時間がピークのときでも訪問者に良好なパフォーマンスを提供することができます。

さらに、Lightsail ロードバランサーは安全性の高いアプリケーションを構築し、HTTPS トラフィックの受信を可能にするのに役立ちます。Lightsail は SSL/TLS 証明書のリクエストやプロビジョニング、維持などの手間を軽減します。ビルトインの証明書管理は、ユーザーの代わりに証明書をリクエストおよび更新し、証明書をロードバランサーに自動的に追加します。

異なるアベイラビリティーゾーン AWS リージョン または異なるアベイラビリティーゾーンのインスタンスでロードバランサーを使用できますか?

異なる AWS リージョンで実行されているインスタンスではロードバランサーを使用できません。ただし、異なるアベイラビリティーゾーンのターゲットインスタンスでは、ロードバランサーを使用できます。そのため、ターゲットインスタンスを複数のアベイラビリティーゾーンに分散して、アプリケーションの可用性を最大化することをお勧めしています。

Lightsail ロードバランサーはトラフィックの急増にどのように対応しますか?

Lightsail ロードバランサーは自動的にスケーリングしてアプリケーションへのトラフィックの急増を処理するため、手動で調整する必要はありません。アプリケーションへのトラフィックの急増が一時的なものである場合、Lightsail ロードバランサーは自動的にスケールして、効率的な方法でLightsail インスタンスにトラフィックを継続して送ります。Lightsail ロードバランサーはトラフィックの急増を容易に処理するよう設計されていますが、大量のトラフィックが一貫して発生するアプリケーションではパフォーマンスの低下とスロットリングが起きる可能性があります。5 GB/時間を超える継続的なデータ処理や、大量接続 (新規接続 40 万/時間以上、アクティブな同時接続 1.5 万以上) がお客様のアプリケーションに発生すると予想される場合、Application Load Balancerを備えた Amazon EC2 の使用をお勧めします。

Lightsail ロードバランサーはターゲットインスタンスにどのようにトラフィックをルーティングしますか?

Lightsail ロードバランサーはラウンドロビンアルゴリズムに基づいて、正常なターゲットインスタンスにトラフィックを送信します。

Lightsail ロードバランサーはターゲットインスタンスが正常かどうかをどのように判別しますか?

ロードバランサーを作成してインスタンスにアタッチした際、Lightsail はヘルスチェックリクエストをウェブアプリケーションのルートに送信します。Lightsail が ping を実行する経路 (共通ファイルまたはウェブページの URL) を指定することで、ロケーションをカスタマイズできます。この経路を使用してターゲットインスタンスに到達できた場合、Lightsail はそこにトラフィックをルーティングします。いずれかのターゲットインスタンスが応答しない場合、ヘルスチェックは失敗し、Lightsail はそのインスタンスにはトラフィックをルーティングしません。ヘルスチェックの詳細

ロードバランサーにアタッチできるインスタンスの数を教えてください。

ターゲットインスタンスはお好きなだけロードバランサーに追加できます。(お客様の Lightsail アカウントインスタンスクォータに達するまで)

1 つのインスタンスを複数のロードバランサーに割り当てることはできますか?

はい。Lightsail では、必要に応じてインスタンスをターゲットインスタンスとして複数のロードバランサーに追加することができます。

ロードバランサーを削除すると、ターゲットインスタンスはどうなりますか?

ロードバランサーを削除しても、アタッチされたターゲットインスタンスは引き続き正常に実行され、通常の Lightsail インスタンスとしてLightsail コンソールに表示されます。ロードバランサーを削除した際は、 DNS レコードを更新して過去のターゲットインスタンスのいずれかにトラフィックを送信する必要がある場合が多いので、ご注意ください。

セッション永続性とは何ですか?

セッション永続性を使用すると、ロードバランサーは特定のターゲットインスタンスに訪問者のセッションをバインドすることができます。これにより、セッション中にそのユーザーから受けたリクエストはすべて、同じターゲットインスタンスに送信されます。Lightsail はデータの一貫性を保つため、訪問者が同じターゲットインスタンスをヒットする必要のあるアプリケーションの、セッション永続性をサポートします。例えば、ユーザー認証を必要とする多くのアプリケーションは、セッション永続性を使用する利点があります。ロードバランサーの作成後、ロードバランサー管理画面より指定したロードバランサーに対するセッション永続性が有効にできます。詳細については、「ロードバランサーのセッション永続性を有効にする」を参照してください。

Lightsail ロードバランサーではどのような種類の接続がサポートされますか?

Lightsail ロードバランサーは HTTP および HTTPS 接続をサポートしています。

Lightsail ロードバランサーは IPv6 をサポートしていますか?

Lightsail2021 年 1 月 12 日以降に作成されたロードバランサーは、デフォルトでデュアルスタックモードで動作します (つまり、IPv4 プロトコルと IPv6 プロトコルの両方でクライアントトラフィックを受け入れます)。この日付より前に作成されたロードバランサーは、ロードバランサー管理ページの [ネットワーク] タブの切り替えスイッチによって IPv6 を有効にできます。このトグルを使用して、ロードバランサーの IPv6 を無効にすることも可能です。

IPv6 が有効になっているロードバランサーを使うには、そのロードバランサーの背後にあるインスタンスの IPv6 も有効になっている必要がありますか?

いいえ。ロードバランサーは IPv4 と IPv6 両方のトラフィックを受け入れ、バックエンドのインスタンスと通信する際は、シームレスに IPv4 へ変換します。したがって、ロードバランサーの背後にあるインスタンスはデュアルスタックまたは IPv4 のみの、いずれも可能です。

手動および自動スナップショット

スナップショットとは何ですか?

スナップショットは、インスタンスまたはデータベース、およびブロックストレージディスクの特定の時点のバックアップです。リソースのスナップショットはいつでも作成できます。または、インスタンスおよびディスクの自動スナップショットを有効にすると、ユーザーの代わりに Lightsail がスナップショットを作成します。スナップショットをベースラインとして使用して、新しいリソースを作成したりデータをバックアップすることが可能です。スナップショットには、リソースの復元に必要なすべてのデータ (スナップショットが作成された時点のデータ) が含まれます。スナップショットからリソースを作成して復元すると、その新しいリソースはスナップショットの作成に使用された元のリソースの正確なレプリカとして始まります。

Lightsail インスタンス、ディスク、データベースのスナップショットを手動で作成できます。また自動スナップショットを使用すると、 Lightsail にインスタンスとディスクのスナップショットを自動的に毎日作成するように、指示できます。詳細については、「スナップショット」を参照してください。

自動スナップショットとは何ですか?

自動スナップショットは、Amazon Lightsail で Linux/Unix インスタンスの毎日のスナップショットをスケジュールする方法の 1 つです。ユーザーは時間を指定することが可能で、Lightsail は指定した時間に自動的にスナップショットを作成し、最新の自動スナップショットを 7 つ、常に保持します。スナップショットの有効化は無料です。スナップショットで使われた実際のストレージの料金のみをお支払いいただきます。

手動スナップショットと自動スナップショットの違いは何ですか?

自動スナップショットはタグ付けしたり、Amazon EC2 に直接エクスポートしたりすることができません。ただし、自動スナップショットはコピーして手動のスナップショットに変換することができ

ます。自動スナップショットを手動スナップショットにコピーするには、自動スナップショットのコンテキストメニューから [Keep] を選択して、手動スナップショットとしてコピーします。

どのようなリソースがスナップショットをサポートしていますか?

インスタンス、データベース、ディスクの手動スナップショットが作成できます。

自動スナップショットは、Lightsailコンソール、LightsailAPI、または を使用して Linux または Unix インスタンスで有効にし AWS CLI、LightsailAPI のみを使用してディスクで有効にすることができます AWS CLI。自動スナップショットは現在、Windows インスタンスまたはマネージドデータベースではサポートされていません。

スナップショットはどれくらいの期間保存できますか?

手動スナップショットはユーザーが削除することを選択するまで保存されます。詳細については、Amazon Lightsail でスナップショットを削除するを参照してください。

自動スナップショットは、新しい自動スナップショットに置き換えられるまで保存されます。Lightsail は、最新の 7 つの自動スナップショットを保存してから、最も古いスナップショットを削除して最新のスナップショットに置き換えます。ただし、手動スナップショットとしてコピーすることで、特定の自動スナップショットを保持できます。詳細については、Amazon Lightsail でのインスタンスまたはディスクの自動スナップショットの保持を参照してください。アカウントに保存されている自動スナップショットに対してスナップショットストレージ料金が請求されます。

自動スナップショットを有効にするには、どうすればよいですか?

自動スナップショットは、Linux または Unix インスタンスを作成するとき、またはインスタンスが実行中になった後に、Lightsail コンソール、Lightsail API、または AWS CLI を使用して有効にすることができます。

自動スナップショットは、ディスクに対してその作成時または作成後に有効にすることもできます。ただし、これは Lightsail API または AWS CLIを使用してのみ可能です。

詳細については、「Amazon Lightsail でのインスタンスおよびディスクの自動スナップショットの有効化と無効化」を参照してください。

自動スナップショットはいつ作成されますか?

自動スナップショットを有効にすると、リソースが配置されている AWS リージョン に基づいてデフォルト時間が設定されます。自動スナップショットの時間は 1 時間単位で希望の時刻に変更でき

ます。詳細については、Amazon Lightsail でのインスタンスまたはディスクの自動スナップショットの時間変更 を参照してください。

保存できるスナップショットの数を教えてください。

手動スナップショットは必要な数だけ保存できます。ただし、自動スナップショット は最新の 7 つのみが保存され、最も古いスナップショットは最新のスナップショットに置き換えられます。

スナップショットはどのように課金されますか?

Lightsail アカウントに保存されているスナップショットに対してのみお支払いいただきます。Lightsail スナップショット (手動および自動) の保存にかかるコストは 1 か月あたり 0.05 USD/GB です。

自動スナップショットを無効にすると、スナップショットは失われますか?

いいえ。自動スナップショットを無効にすると、Lightsail による毎日のスナップショット作成は停止され、既存の自動スナップショットは保持されます。自動スナップショットを再度有効にすると、Lightsail による毎日のスナップショット作成は再開され、最も古いスナップショットは削除されて、最新のスナップショットに置き換えられます。

自動スナップショットが置き換えられないようにする場合は、どうすればよいですか?

特定の自動スナップショットを、手動スナップショットとしてコピーすることで保持できます。詳細については、Amazon Lightsail でのインスタンスまたはディスクの自動スナップショットの保持を参照してください。

自動スナップショットは削除できますか?

自動スナップショットのコンテキストメニューから [Delete] (削除) を選択することで、いつでも自動スナップショットを削除できます。詳細については、「自動インスタンスのスナップショットを削除する」を参照してください。

スナップショットはどのように使用できますか?

スナップショットは、元のリソースに不具合が発生した場合などに、ベースラインとして使用したり、新しいリソースの作成に使用できます。詳細については、「スナップショット」を参照してください。

スナップショットを Amazon EC2 にエクスポートして、そのサービス内に新しいリソースを作成することもできます。詳細については、「スナップショットを Amazon EC2 にエクスポートする」を参照してください。

リソースのヘルスメトリクスとアラーム

メトリクスとは何ですか?

Lightsail は、インスタンス、データベース、ロードバランサーのメトリクスデータをレポートします。一部のメトリクスには、インスタンスの CPU 使用率 (%)、インバウンドおよびアウトバウンドネットワークトラフィックの量、システムおよびインスタンスのエラー数、データベースディスクキューの深さ、データベース空きストレージ容量、ロードバランサーのエラー数、ロードバランサーの応答時間などがあります。メトリクスを使用すると、リソースの信頼性、可用性、パフォーマンスを監視および維持することができます。リソースから定期的にメトリクスデータをモニタリングして収集し、マルチポイント障害が発生した場合に、より簡単にデバッグできるようにします。詳細については、「リソースのメトリクス」を参照してください。

アラームとは何ですか?

Lightsail では、インスタンス、データベース、またはロードバランサーのメトリクスを監視するアラームを作成することができます。アラームは、指定したしきい値を基準にしたメトリクスの値に基づいて通知するように設定できます。詳細については、「アラーム」を参照してください。

通知は、Lightsail コンソールに表示されるバナー、メールアドレスに送信される メール、携帯電話番号に送信される SMS テキストメッセージです。通知の詳細については、「通知」を参照してください。

アラームはいくつ追加できますか?

インスタンス、データベース、ロードバランサーに使用できるメトリクスごとに 2 つのアラームを設定できます。詳細については、「アラーム」を参照してください。

ネットワーク

Lightsail で IP アドレスをどのように使用しますか ?

すべての Lightsail インスタンスでは、プライベート IPv4 アドレス、パブリック IPv4 アドレス、またはパブリック IPv6 アドレスが自動的に取得されます (2021 年 1 月 12 日以前に作成されたイン

スタンスでは IPv6 を手動で有効にする必要があります)。プライベート IP を使用して、Lightsailインスタンスと AWS リソース間でデータをプライベートに無料で送信できます。パブリック IP を使用すると、登録済みのドメイン名や SSH またはローカルコンピュータからの RDP 接続などを介して、インターネットからインスタンスに接続することができます。また、パブリック IPv4 アドレスの代わりに、インスタンスが停止および開始されても IPv4 アドレスが変化しない静的 IPv4 アドレスをインスタンスに接続することもできます。インスタンスに割り当てられた IPv6 アドレスは、インスタンスが削除されるか、インスタンス上で IPv6 を無効にして IPv6 アドレスを手動で解放するまで、変更されません。

Lightsail は IPv6 のみのインスタンスをサポートしていますか?

はい。Lightsail インスタンスは、デュアルスタック (IPv4 と IPv6) および IPv6 専用環境をサポートしています。

静的 IP とは何ですか?

静的 IP は、Lightsail ユーザーアカウント専用の固定パブリック IP アドレスです。静的 IPv4 アドレスをインスタンスに割り当て、パブリック IPv4 と置き換えることができます。インスタンスを別のインスタンスに置き換える場合は、静的 IP を新しいインスタンスに再割り当てすることができます。この方法により、インスタンスを置き換えるたびに外部システム (DNS レコードなど) を新しい IP に紐付けるための設定をする必要がなくなります。Lightsailは現在、IPv4 の静的 IP のみをサポートしています。静的 IPv6 アドレスは使用できません。ただし、インスタンスに割り当てられた IPv6 アドレスは、インスタンスが削除されるか、インスタンス上で IPv6 を無効にして IPv6 アドレスを手動で解放するまで、変更されません。

インスタンスには静的 IP をいくつアタッチできますか?

1 つのインスタンスに同時アタッチできる静的 IP は 1 つのみです。

DNS レコードとは何ですか?

DNS は世界中に分散して存在するサービスで、`www.example.com` などの人が読むことができる名前を、コンピュータが互いに接続する際に使用される `192.0.2.1` などの英数字の IP アドレスに変換します。Lightsail では、`photos.example.com` などの登録されたドメイン名を、Lightsail インスタンスのパブリック IP に簡単にマッピングできます。これによって、`example.com` などの人が読むことができる名前がブラウザに入力されると、そのアドレスはユーザーがアクセスすべきインスタンスの IP に Lightsail によって自動的に変換されます。これらの変換は、それぞれ DNS クエリと呼ばれます。

Lightsail でドメインを使用するには、必ず先に登録しておく必要があるのでご注意ください。ドメインは Lightsail または選択した DNS レジストラを使用して登録できます。

インスタンスのファイアウォール設定を管理することはできますか?

はい。Lightsail ファイアウォールを使用することで、インスタンスのデータトラフィックを制御できます。Lightsail コンソールから、複数のタイプのトラフィックでパブリックアクセス可能なインスタンスのポートに関するルールを設定することができます。

オブジェクトストレージとバケット

Lightsail オブジェクトストレージではどのようなことができますか?

ユーザーは画像、動画、HTML ファイルなどの静的コンテンツを Lightsail オブジェクトストレージサービスのバケットに保存することができます。バケットに保存されているオブジェクトは、ウェブサイトやアプリケーションで使用することができます。Lightsail オブジェクトストレージは、数回クリックするだけで Lightsail CDN ディストリビューションに関連付けることが可能で、世界中のユーザーへのコンテンツ配信を迅速かつ容易に高速化できるようになります。また、低コストで安全なバックアップ ソリューションとしても使用できます。詳細については、「オブジェクトストレージ」を参照してください。

Lightsail オブジェクトストレージのコストについて教えてください。

Lightsail オブジェクトストレージには、 が利用可能なすべての に 3 AWS リージョンつの異なる固定価格バンドルLightsailがあります。最初のバンドルは 1 USD/月で最初の12カ月間は無料です。このバンドルには 5 GB のストレージ容量と 25 GB 分のデータ転送が含まれます。2 つ目のバンドルは毎月 3 USD で、100 GB のストレージ容量と 250 GB 分のデータ転送が含まれています。最後に、3 番目のバンドルは毎月 5 USD で、250 GB のストレージ容量と 500 GB 分のデータ転送が含まれています。バンドルされたデータ転送許容量はバケットからのデータ転送以外には使用されないため、Lightsail オブジェクトストレージにはバケットへの無制限のデータ転送が含まれます。

Lightsail バケットに関連付けられているプランを変更することはできますか?

はい。個々のLightsailバケットのストレージプランは、毎月の AWS 請求サイクル内で 1 回変更できます。

Lightsail オブジェクトストレージから Amazon S3 にオブジェクトをコピーできますか?

はい。Lightsail オブジェクトストレージから Amazon S3 へのコピーはサポートされています。詳細については、「AWS Premium Support ナレッジセンター」の「Amazon S3 バケットから別のバケットにすべてのオブジェクトをコピーする方法とは?」を参照してください。

Lightsail オブジェクトストレージの使用を開始するにはどうすればよいですか?

Lightsail オブジェクトストレージを使用するには、まずデータを保存するために使用するバケットを作成する必要があります。詳細については、「バケットの作成」を参照してください。バケットが実行状態になったら、Lightsail コンソールを使用してファイルをアップロードするか、ログまたはその他アプリケーションデータといったコンテンツをバケットに追加するようにアプリケーションを設定することによって、バケットへのオブジェクトの追加を開始できます。または、 AWS Command Line Interface (AWS CLI) を使用して Lightsail オブジェクトストレージの使用を開始することもできます。

バケットにオブジェクトをアップロードするにはどうすればよいですか?

画像やそのほかの静的ファイルなどのオブジェクトをバケットにアップロードする場合、トップナビゲーションタブ [Objects] (オブジェクト) から [Upload] (アップロード) を選択し、コンピュータから正しいファイルまたはディレクトリを選択します。または、デスクトップから Lightsail オブジェクトストレージコンソール内のマークされた領域にファイルやディレクトリをドラッグアンドドロップします。

バケットへのパブリックアクセスをブロックできますか?

Lightsail バケットとオブジェクトは、デフォルトでプライベートに設定されています。つまり、適切な許可を持つユーザーのみがバケットとオブジェクトにアクセスできます。ユーザーは、このデフォルト設定を変更し、個々のオブジェクトをプライベートバケットで公開して読み取り専用にするか、バケット全体を公開して読み取り専用にするかを選択できます。ユーザーがバケットまたはオブジェクトを公開すると、世界中のすべての人がそのコンテンツを読むことができます。詳細については、「バケットのアクセス許可」を参照してください。

バケットにプログラムによるアクセスを追加するにはどうすればよいですか?

バケットへのプログラムによるアクセスは、アクセスキーまたはロールのいずれかを使用します。まず、Lightsail コンソールでプログラム的に接続するバケットを選択します。次に、[Permissions] (許可) タブでアクセスキーを作成、または Lightsail インスタンスにロールを割り当ててから、バケットを使用するようにウェブサイトまたはアプリケーションコードを設定します。ウェブサイトまたはアプリケーションでオブジェクトストレージをどのように使うかの用途に応じて、この動作は異なる場合があります。詳細については、「バケットのアクセス許可」を参照してください。

バケットを他の AWS アカウントと共有するにはどうすればよいですか?

Lightsail では、バケット管理ページのクロスアカウントアクセスセクションで指定した AWS アカウント ID を使用してバケットへのアクセスを共有できるため、クロスアカウント共有が容易になります。 AWS アカウント ID を指定すると、そのアカウントはバケットへの読み取り専用アクセス権を持ちます。詳細については、「バケットのアクセス許可」を参照してください。

バージョニングとは何ですか?

バージョニングは、バケット内の全てのオブジェクトストレージのすべてのバージョンを保存、取得、復元することができます。これは、偶発的な上書きや削除からの更なる保護となります。詳細については、「バケットのオブジェクトのバージョニングを有効化または一時停止する」を参照してください。

Lightsail バケットを Lightsail CDN ディストリビューションに関連付けるにはどうすればよいですか?

Lightsail オブジェクトストレージは、数回クリックするだけで Lightsail CDN ディストリビューションに関連付けることが可能で、世界中のユーザーへのコンテンツ配信を迅速かつ容易に高速化できるようになります。これを行うには、Lightsail CDN ディストリビューションを作成し、Lightsail CDN ディストリビューションのオリジンとして Lightsail バケットを選択します。詳細については、「Lightsail コンテンツ配信ネットワークディストリビューションでの Amazon Lightsail バケットの使用」を参照してください。

Lightsail オブジェクトストレージサービスにはどのような制限がありますか?

Lightsail オブジェクトストレージサービスでは、アカウントごとに最大 20 のバケットを作成できます。バケットに保存できるオブジェクト数に制限はありません。すべてのオブジェクトを 1 つのバケットに保存したり、複数のバケットに分けて整理することも可能です。

Lightsail オブジェクトストレージはモニタリングとアラートをサポートしていますか?

Lightsail オブジェクトストレージでは、お客様がバケット内の合計使用済み容量とバケット内のオブジェクト数に関するメトリクスを簡単に表示できます。これらのメトリクスに基づいたアラートもサポートされています。詳細については、「Amazon Lightsail でバケットのメトリクスを表示する」および「バケットメトリクスアラームを作成する」を参照してください。

Lightsail のタグ

タグとは

タグとは、Lightsail リソースに割り当てるラベルです。タグはそれぞれ、1 つのキーと 1 つの値で構成されており、どちらもお客様側が定義します。タグ値はオプションであるため、「キーのみ」のタグを作成して Lightsail コンソールでリソースをフィルタリングすることを選択できます。

Lightsail でタグを使用するには、どうすればよいですか ?

タグを使用することで。Lightsail コンソールおよび API でリソースのグループ化やフィルタリングをしたり、請求書の料金を追跡して整理したり、アクセス管理ルールを通じて誰がリソースを表示または変更できるかを調整したりできます。リソースにタグを付けることで、以下のことができます。

- 整理 – Lightsail コンソールおよび API のフィルタを使用し、割り当てたタグに基づいてリソースを表示および管理します。これは、同じタイプのリソースが多数ある場合に役立ちます。割り当てたタグに基づいて特定のリソースをすばやく識別できます。
- コスト配分 – リソースにタグを付けて請求コンソールで「コスト配分タグ」を作成することで、さまざまなプロジェクトやユーザー間のコストの追跡や配分を行います。例えば、請求書を分割してプロジェクト別またはクライアント別にコストを確認することができます。
- アクセスの管理 – AWS アカウントにアクセスできるユーザーが AWS Identity and Access Management ポリシーを使用してLightsailリソースを編集、作成、削除する方法を制御します。こ

れにより、他者とのコラボレーションが容易になります。Lightsail リソースに対するフルアクセスを他者に付与する必要はありません。

Lightsail でのタグの使用の詳細については、「タグ」を参照してください。

タグ付けできるのはどのようなリソースですか?

Lightsail は現在、以下のリソースのタグ付けをサポートしています。

- インスタンス (Linux および Windows)
- コンテナサービス
- ブロックストレージディスク
- ロードバランサー
- データベース
- DNS ゾーン
- インスタンス、ディスク、データベースの手動スナップショット

手動スナップショットはタグをサポートしますが、スナップショットにタグ AWS CLI を付けるには Lightsail API または を使用する必要があります。Lightsail コンソールを使用して、タグ付きのインスタンス、ディスク、またはデータベースの手動スナップショットを作成する場合、手動スナップショットにはソースリソースと同じタグが自動的に付けられます。Lightsail コンソールを使用して、タグ付きの手動スナップショットから新しいリソースを作成するときに、これらのタグを編集できます。

自動スナップショットにタグを付けることはできません。

Lightsail スナップショットにタグを付けるには、どうすればよいですか?

Lightsail コンソールでは、手動スナップショットにソースリソースと同じタグが自動的に付けられます。Lightsail API または を使用してスナップショット AWS CLI を作成する場合は、スナップショットのタグを自分で選択できます。

> **Important**
>
> データベースの手動スナップショットのタグ (コスト配分タグ) は現在、請求レポートに含まれません。

キー値タグとキーのみタグの違いは何ですか?

Lightsail タグはキーと値のペアであり、さまざまなカテゴリ (project:Blog、project:Game、project:Test など) をまたいでインスタンスなどのリソースを整理できます。これにより、リソースの整理、請求レポート、アクセス管理などのすべてのユースケースを全面的に管理できます。Lightsail コンソールでキーのみのタグをリソースに付けて、コンソールで迅速なフィルタリングを行うこともできます。

Amazon Lightsail のドキュメント履歴

以下の表は、Lightsail のドキュメントの変更と更新をまとめたものです。このテーブルが新しいエントリを取得したときに通知を受け取るには、RSS リーダーを使用して次の URL をサブスクライブします。

```
https://docs.aws.amazon.com/lightsail/latest/userguide/document_history.rss
```

変更	説明	日付
Plesk クイックスタートガイドの更新	クイックスタートガイドを更新して、Lightsail で Plesk インスタンスを作成しました。	2025 年 10 月 31 日
新しいインスタンスバンドル	Lightsail で Linux/Unix インスタンスと Windows インスタンスの両方により大きなインスタンスバンドルを追加しました。	2025 年 10 月 31 日
インスタンスバンドルページ	このドキュメントにインスタンスバンドル情報を詳述する新しいページを追加しました。	2025 年 10 月 31 日
Lightsail が CORS サポートを追加	Lightsail に Lightsail バケットの CORS サポートが追加されました。	2025 年 10 月 15 日
ブループリントクイックスタートガイドの更新	複数のクイックスタートガイドを更新し、スナップショットと静的 IP 関連のコンテンツを改訂しました。	2025 年 10 月 2 日
Lightsail AWS リージョン 拡張	Lightsail が、アジアパシフィック (ジャカルタ) リージョン	2025 年 7 月 31 日

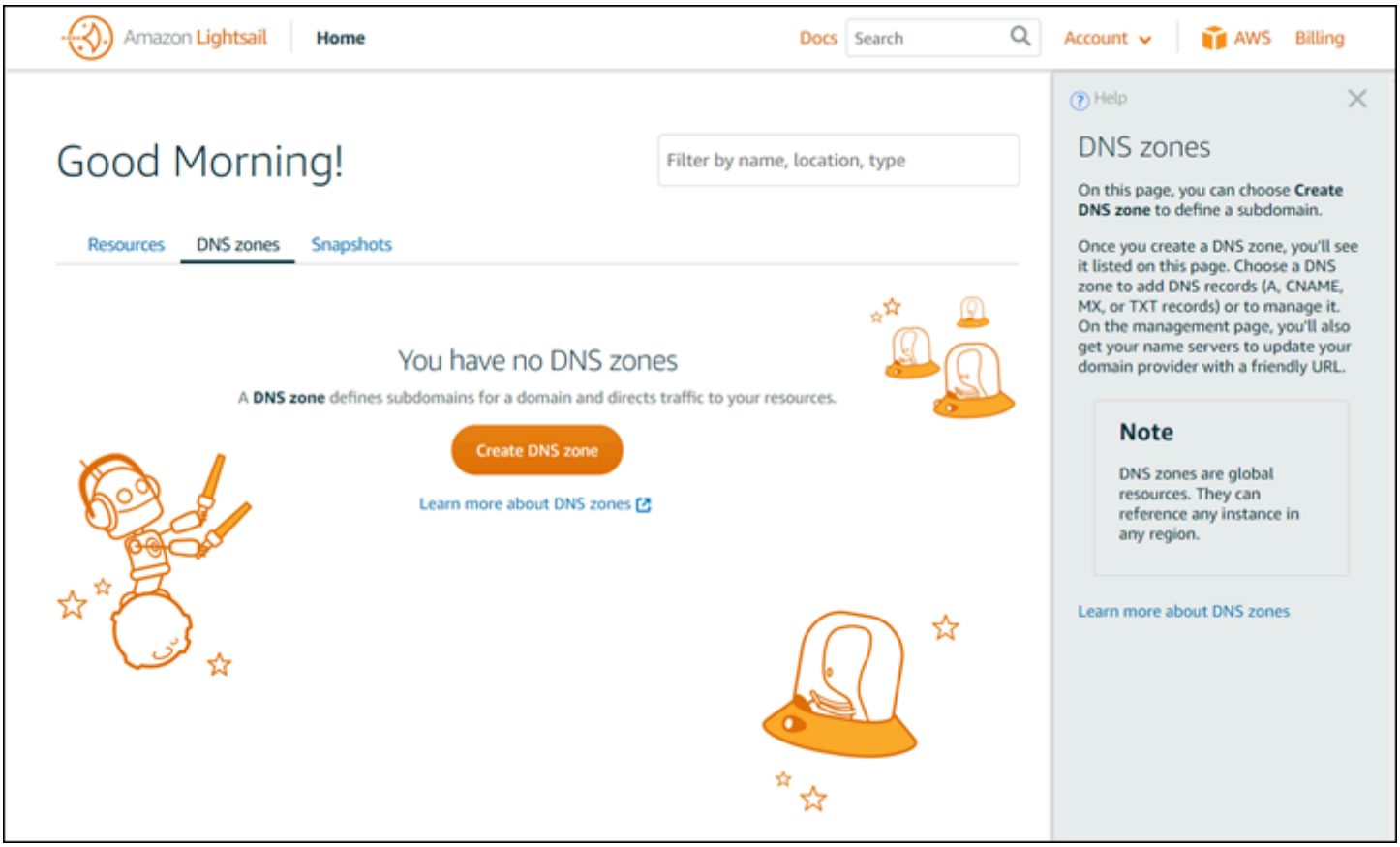
	で利用できるようになりました。	
Lightsail がドキュメントの更新の追跡を開始しました	このドキュメント履歴ページの初回公開。	2025 年 7 月 30 日

Lightsail に役立つリソースを検索する

Amazon Lightsail では、いくつかの方法でヘルプを見つけられます。

コンテキスト依存のヘルプパネル

Lightsail には、コンソールの各ページにコンテキスト依存の [ヘルプ] パネルがあり、現在のページに固有の追加のヒントと情報が表示されます。現在のページについて何か質問がある場合はいつでもヘルプパネルを開き、終わったらヘルプパネルを閉じます。ヘルプパネルを開くには、ページにある [ヘルプ] を選択するか、ユーザーインターフェイスの随所にある小さい疑問符を選択します。



ユーザーガイドについて

Amazon Lightsail ユーザーガイドには、Lightsail での作業に役立つ操作方法のトピックやコンセプトの説明が記載されています。たとえば、インスタンスの作成、インスタンスへの接続、ドメインの管理を行うことができます。

検索の使用

Lightsail の任意のページで、各ページの上部にある検索ボックスを使用して、ドキュメントのトピックを検索できます。ドキュメントの検索ページでもう一度検索すると、検索を絞り込むことができます。

探していたものが見つからなかった場合。当社にフィードバックを送信していただければ対処いたします。Lightsail の各ページで、[フィードバックの提供] とフィードバックの送信を選択して提案を行うことができます。

Lightsail の CLI および API の使用

AWS Command Line Interface (AWS CLI) または Lightsail REST API を使用して、Lightsailリソースを作成、読み取り、更新、削除できます。REST API に加えて、Java、Ruby、JavaScript (Node.js)、Go、PHP、Python、.NET (C#)、C++ などの複数の言語の SDK も提供しています。Lightsail API に関する詳細については、「Lightsail API レファレンス」を参照してください。

> **Note**
>
> LightsailAPI を使用するには、アクセスキーを生成する必要があります。Lightsail API を使用するためのアクセスキーのセットアップ方法を確認してください。

AWS CLI は、 Lightsailリソースを使用する際に役立ちます。AWS で AWS CLI、「」と入力`aws lightsail help`するだけで、使用可能なコマンドについて学習できます。特定の CLI コマンドのヘルプを表示する場合は、コマンド名の後に `help` と入力すると、そのパラメーターと例外の詳細が表示されます。詳細については、「Lightsail CLI リファレンス」を参照してください。

AWS フォーラムおよびその他のコミュニティリソース

AWS ディスカッションフォーラム「AWS フォーラム」に質問を投稿することもできます。

翻訳は機械翻訳により提供されています。提供された翻訳内容と英語版の間で齟齬、不一致または矛盾がある場合、英語版が優先します。