



ユーザーガイド

# Amazon S3 on Outposts



API バージョン 2006-03-01

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

# Amazon S3 on Outposts: ユーザーガイド

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

Amazon の商標とトレードドレスは、Amazon 以外の製品またはサービスとの関連において、顧客に混乱を招いたり、Amazon の名誉または信用を毀損するような方法で使用することはできません。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

S3 on Outposts とは .....	1
S3 on Outposts のしくみ .....	1
Regions .....	2
バケット .....	2
オブジェクト .....	3
キー .....	3
S3 バージョニング .....	4
バージョン ID .....	4
ストレージクラスと暗号化 .....	4
バケットポリシー .....	4
S3 on Outposts アクセスポイント .....	5
S3 on Outposts の機能 .....	6
アクセス管理 .....	6
ストレージのログ記録とモニタリング .....	6
強力な整合性 .....	7
関連サービス .....	7
S3 on Outposts へのアクセス .....	8
AWS マネジメントコンソール .....	8
AWS Command Line Interface .....	8
AWS SDK .....	8
S3 on Outposts の支払い .....	9
次のステップ .....	9
Outpost の設定 .....	10
新しい Outpost をオーダーする .....	10
S3 on Outposts の違い .....	11
仕様 .....	11
サポートされている API オペレーション .....	12
S3 on Outposts でサポートされている Amazon S3 AWS CLI コマンド .....	12
サポートされていない Amazon S3 の機能 .....	12
ネットワーク制限 .....	13
S3 on Outposts の開始方法 .....	15
S3 コンソールの使用 .....	15
バケット、アクセスポイント、およびエンドポイントを作成する .....	16
次のステップ .....	18

AWS CLI および SDK for Java の使用 .....	19
ステップ 1: バケットを作成する .....	19
ステップ 2: アクセスポイントの作成 .....	20
ステップ 3: エンドポイントを作成する .....	21
ステップ 4: S3 on Outposts バケットにオブジェクトをアップロードする .....	22
S3 on Outposts のネットワーキング .....	23
ネットワークアクセスタイプの選択 .....	23
S3 on Outposts のバケットおよびオブジェクトにアクセスする .....	23
クロスアカウント Elastic Network Interface を使用した接続の管理 .....	24
S3 on Outposts バケットの操作 .....	25
バケット .....	25
アクセスポイント .....	25
エンドポイント .....	26
S3 on Outposts の API オペレーション .....	26
S3 on Outposts バケットを作成および管理する .....	28
バケットの作成 .....	28
タグの追加 .....	32
バケットポリシーの使用 .....	33
バケットポリシーの追加 .....	34
バケットポリシーの表示 .....	36
バケットポリシーの削除 .....	37
バケットポリシーの例 .....	38
バケットの一覧表示 .....	43
バケットを取得する .....	44
バケットの削除 .....	46
アクセスポイントの使用 .....	47
アクセスポイントの作成 .....	48
アクセスポイントでのバケット形式のエイリアスの使用 .....	49
アクセスポイント設定の表示 .....	54
アクセスポイントの一覧表示 .....	55
アクセスポイントの削除 .....	56
アクセスポイントポリシーの編集 .....	57
アクセスポイントポリシーの表示 .....	59
エンドポイントの使用 .....	61
エンドポイントの作成 .....	62
エンドポイントを一覧表示する .....	64

エンドポイントの削除 .....	65
S3 on Outposts オブジェクトの操作 .....	67
オブジェクトのアップロード .....	68
オブジェクトのコピー .....	69
AWS SDK for Java の使用 .....	70
オブジェクトの取得 .....	71
オブジェクトのリスト化 .....	74
オブジェクトの削除 .....	77
HeadBucket の使用 .....	82
マルチパートアップロードの実行 .....	84
S3 on Outposts バケット内のオブジェクトのマルチパートアップロードを実行する .....	84
S3 on Outposts バケット内のラージオブジェクトをマルチパートアップロードでコピーする .....	87
S3 on Outposts バケットにオブジェクトのパーツを一覧表示する .....	89
S3 on Outposts バケット内の進行中のマルチパートアップロードのリストを取得する .....	90
署名付き URL の使用 .....	91
署名付き URL 機能の制限 .....	92
署名付き URL を作成できるユーザー .....	94
S3 on Outposts は、どのタイミングで署名付き URL の有効期限切れの日時を確認しますか? .....	95
オブジェクトの共有 .....	95
オブジェクトのアップロード .....	100
ローカルの Amazon EMR を使用した Amazon S3 on Outposts .....	105
Amazon S3 on Outposts バケットを作成する .....	106
Amazon S3 on Outposts での Amazon EMR の開始方法 .....	107
認可と認証キャッシング .....	112
認可キャッシュと認証キャッシュの設定 .....	112
SigV4A 署名の検証 .....	113
セキュリティ .....	114
IAM の設定 .....	115
S3 on Outposts ポリシーのプリンシパル .....	117
S3 on Outposts の ARN .....	117
S3 on Outposts のポリシー例 .....	119
エンドポイントのアクセス許可 .....	119
S3 on Outposts でのサービスにリンクされたロール .....	122
データ暗号化 .....	122

S3 on Outposts の AWS PrivateLink .....	122
制約と制限 .....	124
S3 on Outposts インターフェイスエンドポイントへのアクセス .....	124
オンプレミスの DNS 設定の更新 .....	126
VPC エンドポイントの作成 .....	126
VPC エンドポイントポリシーとバケットポリシーの作成 .....	127
Signature Version 4 (SigV4) のポリシーキー .....	129
Signature Version 4 関連の条件キーを使用するバケットポリシーの例 .....	131
AWS マネージドポリシー .....	133
AWSS3OnOutpostsServiceRolePolicy .....	134
ポリシーの更新 .....	134
サービスにリンクされたロールの使用 .....	135
S3 on Outposts のサービスにリンクされたロールのアクセス許可 .....	135
S3 on Outposts でのサービスにリンクされたロールの作成 .....	138
S3 on Outposts でのサービスにリンクされたロールの編集 .....	139
S3 on Outposts でのサービスにリンクされたロールの削除 .....	139
S3 on Outposts で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 .....	140
S3 on Outposts ストレージの管理 .....	141
S3 バージョニングの管理 .....	141
ライフサイクル設定の作成と管理 .....	144
コンソールを使用する .....	144
AWS CLI および SDK for Java の使用 .....	149
S3 on Outposts のオブジェクトのレプリケート .....	152
レプリケーション設定 .....	153
S3 Replication on Outposts の要件 .....	154
レプリケーションの対象 .....	155
レプリケーションの対象外 .....	156
S3 Replication on Outposts でサポートされていないものは何ですか? .....	157
レプリケーションの設定 .....	157
レプリケーションの管理 .....	177
S3 on Outposts の共有 .....	186
前提条件 .....	186
手順 .....	187
使用例 .....	188
その他の サービス .....	191
S3 on Outposts のモニタリング .....	192

CloudWatch メトリクス .....	192
CloudWatch メトリクス .....	193
Amazon CloudWatch Events .....	195
CloudTrail ログ .....	196
S3 on Outposts オブジェクトの CloudTrail ログ記録の有効化 .....	197
Amazon S3 on Outposts AWS CloudTrail ログファイルのエントリ .....	199
S3 on Outposts での開発 .....	202
サポートされるリージョン .....	202
S3 on Outposts API .....	203
オブジェクト管理のための Amazon S3 API オペレーション .....	203
バケット管理のための Amazon S3 コントロール API オペレーション .....	204
Outposts を管理するための S3 on Outposts API オペレーション .....	205
S3コントロールクライアントの設定 .....	206
IPv6 を使用したリクエストの実行 .....	206
IPv6 の使用開始 .....	207
デュアルスタックのエンドポイントを使用したリクエストの実行 .....	208
IAM ポリシーでの IPv6 アドレスの使用 .....	208
IP アドレス互換性のテスト .....	210
AWS PrivateLink での IPv6 の使用 .....	210
デュアルスタックのエンドポイントの使用 .....	213

# Amazon S3 on Outposts とは

AWS Outposts は、同じ AWS インフラストラクチャ、AWS サービス、API、およびツールを実質的にあらゆるデータセンター、コロケーションスペース、またはオンプレミスの施設に提供するフルマネージドサービスであり、真に一貫性のあるハイブリッドエクスペリエンスを実現します。AWS Outposts は、オンプレミスシステムへの低レイテンシーアクセス、ローカルデータ処理、データレジデンシー、およびローカルシステムの相互依存性を持つアプリケーションの移行に必要なワークロードに最適です。詳細については、「AWS Outposts ユーザーガイド」の「[What is AWS Outposts?](#)」を参照してください。

Amazon S3 on Outposts を使用すると、S3 バケットを Outposts に作成して、オンプレミスでのオブジェクトの保存と取得を容易に行うことができます。S3 on Outposts は、OUTPOSTS という新しいストレージクラスを提供し、これは Amazon S3 API を使用し、Outposts の複数のデバイスとサーバー間でデータを永続的かつ冗長的に保存するように設計されています。仮想プライベートクラウド (VPC) を介したアクセスポイントとエンドポイント接続を使用して、Outposts バケットと通信します。

Outposts バケットでは、Amazon S3 と同じ API と機能 (アクセスポリシー、暗号化、タグ付けなど) を使用できます。AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDK、または REST API を使用して S3 on Outposts を使用できます。

- [S3 on Outposts のしくみ](#)
- [S3 on Outposts の機能](#)
- [関連サービス](#)
- [S3 on Outposts へのアクセス](#)
- [S3 on Outposts の支払い](#)
- [次のステップ](#)

## S3 on Outposts のしくみ

S3 on Outposts は、オブジェクトストレージサービスであり、データを Outpost 上のバケット内にオブジェクトとして保存します。オブジェクトとは、データファイルと、そのファイルを記述する任意のメタデータのことです。バケットとは、オブジェクトのコンテナのことです。

S3 on Outposts にデータを保存するには、まずバケットを作成します。バケットを作成するときには、バケット名とバケットを保持する Outpost を指定します。S3 on Outposts バケットにアクセス

してオブジェクト操作を実行するには、次にアクセスポイントを作成して設定します。アクセスポイントにリクエストをルーティングするエンドポイントも作成する必要があります。

アクセスポイントは、S3 にデータを保存するあらゆる AWS のサービス やお客様のアプリケーションのデータアクセスを簡素化します。アクセスポイントは、バケットにアタッチされた名前付きのネットワークエンドポイントであり、GetObject や PutObject などのオブジェクト操作を実行するために使用できます。各アクセスポイントには、個別の許可とネットワーク制御があります。

AWS マネジメントコンソール、AWS CLI、AWS SDK、または REST API を使用して、S3 on Outposts バケット、アクセスポイント、およびエンドポイントの作成と管理ができます。S3 on Outposts バケット内のオブジェクトをアップロードおよび管理するには、AWS CLI、AWS SDK、または REST API を使用できます。

## Regions

AWS Outposts プロビジョニング時、ユーザーまたは AWS は、バケット操作とテレメトリ用に、Outpost を選択した AWS リージョン または Outposts ホームリージョンに接続するサービスリンク接続を作成します。Outpost は、親 AWS リージョン への接続性に依存します。Outposts ラックは、切断された操作や接続がない環境向けに設計されていません。詳細については、「AWS Outposts ユーザーガイド」の「[AWS リージョン への Outpost の接続性](#)」を参照してください。

## バケット

バケットとは、S3 on Outposts に保存されるオブジェクトのコンテナです。バケットにはオブジェクトをいくつでも保存でき、1 つの Outpost には 1 つのアカウントにつき最大 100 個のバケットを保存できます。

バケットを作成するときには、バケット名を入力し、バケットが存在する Outpost を選択します。バケットの作成後は、バケット名を変更したり、バケットを別の Outpost に移動したりすることはできません。バケット名は、[Amazon S3 バケットの命名規則](#)に従う必要があります。S3 on Outposts では、バケット名は Outpost と AWS アカウント に対して一意です。S3 on Outposts バケットには、バケットを識別するための outpost-id、account-id、およびバケット名が必要です。

以下の例では、S3 on Outposts バケットの Amazon リソースネーム (ARN) 形式を示しています。ARN は、Outpost が属するリージョン、Outpost アカウント、Outpost ID、およびバケット名で構成されます。

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/bucket/bucket-name
```

すべてのオブジェクトはバケット内に保存されます。Outposts バケット内の任意のオブジェクトにアクセスするには、アクセスポイントを使用する必要があります。オブジェクト操作のためにバケットを指定するときには、アクセスポイント ARN またはアクセスポイントエイリアスを使用します。アクセスポイントエイリアスの詳細については、「[S3 on Outposts アクセスポイントでのバケット形式のエイリアスの使用](#)」を参照してください。

次の例は、outpost-id、account-id、アクセスポイント名を含む S3 on Outposts のアクセスポイント ARN 形式を示しています。

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

バケットの詳細については、「[S3 on Outposts バケットの操作](#)」を参照してください。

## オブジェクト

オブジェクトは、S3 on Outposts に保存される基本エンティティです。オブジェクトは、オブジェクトデータとメタデータで構成されます。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メタデータや、Content-Type などの標準 HTTP メタデータが含まれます。また、オブジェクトの保存時にカスタムメタデータを指定することもできます。オブジェクトは、キー (または名前) によってバケット内で一意に識別されます。

Amazon S3 on Outposts では、オブジェクトデータは常に Outpost に保存されます。AWS が Outpost ラックを設置すると、データの常駐要件を満たすために、データは Outpost にローカルに保たれます。オブジェクトが Outpost を離れたり、AWS リージョン 外に出たりすることはありません。AWS マネジメントコンソールはリージョン内でホストされるため、コンソールを使用して Outpost にオブジェクトをアップロードしたり、管理したりすることはできません。ただし REST API、AWS Command Line Interface (AWS CLI)、および AWS SDK を使用して、アクセスポイントを介してオブジェクトのアップロードと管理を行うことができます。

## キー

オブジェクトキー (または キー名) 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。バケットとオブジェクトキーの組み合わせで、各オブジェクトを一意的に識別します。

次の例は、S3 on Outposts オブジェクトの ARN 形式を示し、これは、Outpost が属するリージョンの AWS リージョン コード、AWS アカウント ID、Outpost ID、バケット名、およびオブジェクトキーを含みます。

```
arn:aws:s3-outposts:us-west-2:123456789012:outpost/ op-01ac5d28a6a232904/bucket/amzn-s3-demo-bucket1/object/myobject
```

オブジェクトキーの詳細については、「[S3 on Outposts オブジェクトの操作](#)」を参照してください。

## S3 バージョニング

Outposts バケットの S3 バージョニングを使用して、オブジェクトの複数のバリエーションを同じバケットに保持できます。S3 バージョニングを使用すると、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、復元することができます。S3 バージョニングによって、意図しないユーザーアクションやアプリケーション障害から復旧できます。

詳細については、「[S3 on Outposts バケットの S3 バージョニングの管理](#)」を参照してください。

### バージョン ID

バケットで S3 バージョニングを有効にすると、S3 on Outposts はバケットに追加されたすべてのオブジェクトに一意のバージョン ID を与えます。バージョニングを有効にした時点でバケットにすでに存在していたオブジェクトのバージョン ID は null です。これらの (またはその他の) オブジェクトを他のオペレーション ([PutObject](#)) で変更すると、新しいオブジェクトは一意のバージョン ID を取得します。

詳細については、「[S3 on Outposts バケットの S3 バージョニングの管理](#)」を参照してください。

## ストレージクラスと暗号化

S3 on Outposts は、新しいストレージクラスである S3 Outposts (OUTPOSTS) を提供します。S3 Outposts のストレージクラスは、AWS Outposts のバケットに保存されたオブジェクトに対してのみ使用できます。S3 on Outposts で他の S3 ストレージクラスを使用しようとすると、S3 on Outposts は InvalidStorageClass エラーを返します。

デフォルトでは、S3 Outposts (OUTPOSTS) ストレージクラスに保存されたオブジェクトは、Amazon S3 マネージド暗号化キーによるサーバー側の暗号化 (SSE-S3) を使用して暗号化されます。詳細については、「[S3 on Outposts のデータ暗号化](#)」を参照してください。

## バケットポリシー

バケットポリシーは、リソースベースの AWS Identity and Access Management (IAM) ポリシーを使用して、バケットとその中のオブジェクトへのアクセス許可を付与できます。バケット所有者のみ

が、ポリシーをバケットに関連付けることができます。バケットに添付された許可は、バケット所有者が所有するバケットのすべてのオブジェクトに適用されます。バケットポリシーのサイズは 20 KB に制限されています。

バケットポリシーは、AWS で標準である JSON ベースの IAM ポリシー言語を使用します。バケットポリシーを使用して、バケット内のオブジェクトに対する許可を追加または拒否できます。バケットポリシーは、ポリシーの要素に基づいて、リクエストを許可または拒否します。これらの要素には、リクエスト、S3 on Outposts アクション、リソース、およびリクエストの側面または条件 (リクエストの作成に使用された IP アドレスなど) が含まれます。例えば、バケット所有者がアップロードされるオブジェクトの完全コントロールを持っていることを確認しつつ、S3 on Outposts バケットにオブジェクトをアップロードするクロスアカウント許可を付与するバケットポリシーを作成できます。

バケットポリシーでは、ARN やその他の値でワイルドカード文字 (\*) を使用して、オブジェクトのサブセットに対する許可を付与できます。例えば、共通の[プレフィックス](#)で始まるか、.html などの特定の拡張子で終わるオブジェクトのグループへのアクセスをコントロールできます。

## S3 on Outposts アクセスポイント

S3 on Outposts アクセスポイントは、名前付きネットワークエンドポイントであり、そのエンドポイントを使用してデータにアクセスする方法を記述した専用のアクセスポリシーを持ちます。アクセスポイントは、S3 on Outposts の共有データセットへの大規模なデータアクセスの管理を簡素化します。アクセスポイントは、バケットにアタッチされ、それを使用して、GetObject や PutObject などの S3 オブジェクト操作を実行できます。

オブジェクト操作のためにバケットを指定するときには、アクセスポイント ARN またはアクセスポイントエイリアスを使用します。アクセスポイントエイリアスの詳細については、「[S3 on Outposts アクセスポイントでのバケット形式のエイリアスの使用](#)」を参照してください。

アクセスポイントには、そのアクセスポイントを介して行われるすべてのリクエストに S3 on Outposts が適用する個別の許可とネットワークコントロールがあります。各アクセスポイントは、基になるバケットにアタッチされたバケットポリシーと連動して機能するカスタマイズされたアクセスポイントポリシーを適用します。

詳細については、「[S3 on Outposts のバケットおよびオブジェクトにアクセスする](#)」を参照してください。

# S3 on Outposts の機能

## アクセス管理

S3 on Outposts には、バケットとオブジェクトへのアクセスを監査および管理する機能があります。デフォルトでは、S3 on Outposts バケットとそれらの中のオブジェクトはプライベートです。自分が作成した S3 on Outposts リソースにのみアクセスできます。

以下の機能を使用して、特定のユースケースをサポートする詳細なリソース許可を付与したり、S3 on Outposts リソースの許可を監査したりできます。

- [S3 ブロックパブリックアクセス](#) – 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します。Outposts のバケットの場合、デフォルトでは、[パブリックアクセスをブロック] は常に有効です。
- [AWS Identity and Access Management \(IAM\)](#) – IAM は、AWS リソース (S3 on Outposts リソースなど) へのアクセスを安全に管理するためのウェブサービスです。IAM を使用すると、ユーザーがアクセスできる AWS のリソースを制御するアクセス許可を集中管理できます。IAM を使用して、誰を認証 (サインイン) し、誰にリソースの使用を認可する (アクセス許可を付与する) かを制御します。
- [S3 on Outposts アクセスポイント](#) – S3 on Outposts の共有データセットへのデータアクセスを管理します。アクセスポイントは、専用アクセスポリシーを持つ名前付きネットワークエンドポイントです。アクセスポイントは、バケットにアタッチされ、GetObject や PutObject などのオブジェクト操作を実行するために使用できます。
- [バケットポリシー](#) – IAM ベースのポリシー言語を使用して、S3 バケットとその中のオブジェクトに対するリソースベースの許可を設定します。
- [AWS Resource Access Manager \(AWS RAM\)](#) – S3 on Outposts の容量を AWS アカウント 間、組織内、または AWS Organizations の組織単位 (OU) 内で安全に共有します。

## ストレージのログ記録とモニタリング

S3 on Outposts には、S3 on Outposts リソースの使用状況をモニタリングおよびコントロールするためのロギングおよびモニタリングツールが用意されています。詳細については、「[モニタリングツール](#)」を参照してください。

- [S3 on Outposts 用の Amazon CloudWatch メトリクス](#) – リソースの運用状態を追跡し、容量の可用性を把握します。

- [S3 on Outposts 用 Amazon CloudWatch Events イベント](#) — S3 on Outposts API イベントのルールを作成して、Amazon Simple Queue Service (Amazon SQS)、Amazon Simple Notification Service (Amazon SNS)、AWS Lambda など、サポートされているすべての CloudWatch Events ターゲットを通じて通知を受信します。
- [AWS CloudTrail S3 on Outposts のログ](#) — S3 on Outposts でユーザー、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を記録します。CloudTrail ログを使用すると、S3 バケットレベルおよびオブジェクトレベルのオペレーションの詳細な API 追跡が可能になります。

## 強力な整合性

S3 on Outposts は、すべての AWS リージョンの S3 on Outposts バケット内のオブジェクトの PUT リクエストと DELETE リクエストについて、書き込み後読み取りの強力な整合性を提供します。この動作は、新しいオブジェクトの書き込みと、既存のオブジェクトを上書きする PUT リクエストの両方に適用され、DELETE リクエストにも適用されます。さらに、S3 on Outposts オブジェクトタグとオブジェクトメタデータ (HEAD オブジェクトなど) には、強力な整合性があります。詳細については、「Amazon S3 ユーザーガイド」の「[Amazon S3 のデータ整合性モデル](#)」を参照してください。

## 関連サービス

S3 on Outposts にロードしたデータは、他の AWS のサービスで使用できます。よく使用すると思われるサービスは次のとおりです。

- [Amazon Elastic Compute Cloud \(Amazon EC2\)](#) – 安全でスケーラブルなコンピューティング容量を AWS クラウドで提供します。Amazon EC2 の使用により、ハードウェアに事前投資する必要がなくなり、アプリケーションをより速く開発およびデプロイできます。Amazon EC2 を使用すると、必要な数 (またはそれ以下) の仮想サーバーの起動、セキュリティおよびネットワーキングの構成、ストレージの管理ができます。
- [Outposts 上の Amazon Elastic Block Store](#) - Outposts 上で Amazon EBS ローカルスナップショットを使用して、S3 on Outposts に Outpost のボリュームのスナップショットをローカルに保存できます。
- [Outposts 上の Amazon Relational Database Service \(Amazon RDS\)](#) — Amazon RDS ローカルバックアップを使用して、Amazon RDS バックアップを Outpost にローカルに保存できます。
- [AWS DataSync](#) - Outposts と AWS リージョン 間のデータ転送を自動化して、転送する対象、転送するタイミング、使用するネットワーク帯域幅を選択できます。S3 on Outposts は、と統合されていますAWS DataSync 高スループットのローカル処理を必要とするオンプレミスアプリケー

シヨンの場合、S3 on Outposts は、データ転送とネットワーク変動からのバッファを最小限に抑えるオンプレミスのオブジェクトストレージだけでなく、Outposts と AWS リージョン 間でデータを簡単に転送する機能も提供します。

## S3 on Outposts へのアクセス

S3 on Outposts は、次のいずれかの方法で使用できます。

### AWS マネジメントコンソール

コンソールは、S3 on Outposts と AWS リソースのウェブベースのユーザーインターフェイスです。AWS アカウント にサインアップ済みの場合は、AWS マネジメントコンソール にサインインし、AWS マネジメントコンソール ホームページから [S3] を選択することで、S3 on Outposts にアクセスできます。次に、左のナビゲーションペインから [Outposts buckets] (Outposts バケット) を選択します。

### AWS Command Line Interface

AWS コマンドラインツールを使用して、コマンドを発行するか、システムのコマンドラインでスクリプトを作成して AWS (S3 を含む) タスクを実行します。

[AWS Command Line Interface \(AWS CLI\)](#) は、幅広い AWS のサービスのセットに対するコマンドを提供します。AWS CLI は、Windows、macOS、Linux でサポートされています。使用を開始するには、「[AWS Command Line Interfaceユーザーガイド](#)」を参照してください。S3 on Outposts で使用できるコマンドの詳細については、「AWS CLI コマンドリファレンス」で [s3api](#)、[s3control](#)、および [s3outposts](#) を参照してください。

### AWS SDK

AWS には、さまざまなプログラミング言語およびプラットフォーム (Java、Python、Ruby、.NET、iOS、Android など) のライブラリとサンプルコードで構成された SDK (ソフトウェア開発キット) が用意されています。AWS SDK は、S3 on Outposts と AWS へのプログラムによるアクセスを作成するのに役立ちます。S3 on Outposts は Amazon S3 と同じ SDK を使用するため、S3 on Outposts は同じ S3 API、自動化、およびツールを使用して、一貫したエクスペリエンスを提供します。

S3 on Outposts は REST サービスです。AWS SDK ライブラリを使用して S3 on Outposts にリクエストを送信できます。これは、基盤となる REST API をラップして、プログラミングタスクを簡素

化します。例えば、SDK は署名の計算、リクエストの暗号化による署名、エラーの管理、リクエストの自動再試行などのタスクを処理します。AWS SDK のダウンロードやインストールなどの詳細については、「[AWS での構築ツール](#)」を参照してください。

## S3 on Outposts の支払い

Amazon EC2 インスタンスタイプ、Amazon EBS 汎用ソリッドステートドライブ (SSD) ポリユーム (gp2)、および S3 on Outposts の組み合わせを特徴とするさまざまな AWS Outposts ラック構成を購入できます。価格設定には、配送、設置、インフラストラクチャサービス保守、およびソフトウェアパッチとアップグレードが含まれます。

詳細については、「[AWS Outposts ラックの料金](#)」を参照してください。

## 次のステップ

S3 on Outposts の使用の詳細については、以下のトピックを参照してください。

- [Outpost の設定](#)
- [Amazon S3 on Outposts と Amazon S3 の違い](#)
- [Amazon S3 on Outposts の開始方法](#)
- [S3 on Outposts のネットワーキング](#)
- [S3 on Outposts バケットの操作](#)
- [S3 on Outposts オブジェクトの操作](#)
- [S3 on Outposts のセキュリティ](#)
- [S3 on Outposts ストレージの管理](#)
- [Amazon S3 on Outposts での開発](#)

# Outpost の設定

Amazon S3 on Outposts の使用を開始するには、施設にデプロイされた Amazon S3 容量の Outpost が必要です。Outpost と S3 容量の注文オプションの詳細については、「[AWS Outposts](#)」を参照してください。Outposts に S3 容量があるかどうかを確認するには、[ListOutPostsWithS3](#) API コールを使用できます。仕様と、S3 on Outposts が Amazon S3 とどのように異なるかについては、「[Amazon S3 on Outposts と Amazon S3 の違い](#)」を参照してください。

詳細については、以下のトピックを参照してください。

## トピック

- [新しい Outpost をオーダーする](#)

## 新しい Outpost をオーダーする

S3 容量を持つ新しい Outpost を注文する必要がある場合は、[AWS Outposts ラックの料金](#)を参照して、Amazon Elastic Compute Cloud (Amazon EC2)、Amazon Elastic Block Store (Amazon EBS)、Amazon S3 の容量オプションについてご確認ください。

構成を選択したら、「AWS Outposts ユーザーガイド」の「[Outpost を作成して Outpost 容量を注文する](#)」の手順に従います。

# Amazon S3 on Outposts と Amazon S3 の違い

Amazon S3 on Outposts は、オンプレミスの AWS Outposts 環境にオブジェクトストレージを提供します。S3 on Outposts を使用すると、オンプレミスアプリケーションの近くにデータを維持することによって、ローカル処理、データの常駐性、および要求の厳しいパフォーマンスニーズを満たすのに役立ちます。Amazon S3 の API と機能を使用することで、S3 on Outposts は、Outposts へのデータの保存、保護、タグ付け、レポート作成、およびアクセスコントロールを容易にし、AWS インフラストラクチャをオンプレミス施設に拡張して、一貫したハイブリッドエクスペリエンスを実現します。

S3 on Outposts の独自性については、以下のトピックを参照してください。

## トピック

- [S3 on Outposts の仕様](#)
- [S3 on Outposts でサポートされている API のオペレーション](#)
- [S3 on Outposts でサポートされている Amazon S3 AWS CLI コマンド](#)
- [S3 on Outposts でサポートされていない Amazon S3 の機能](#)
- [S3 on Outposts のネットワーク要件](#)

## S3 on Outposts の仕様

- Outposts のバケットの最大サイズは 50 TB です。
- Outposts バケット内のオブジェクトの最大サイズは 5 TB です。
- AWS アカウントごとの Outposts のバケットの最大数は 100 です。
- Outposts のバケットには、アクセスポイントとエンドポイントを使用してのみアクセスできません。
- Outposts のバケットごとのアクセスポイントの最大数は 10 です。
- アクセスポイントのポリシーのサイズは 20 KB に制限されています。
- Outpost 所有者は、AWS Resource Access Manager を使用して AWS Organizations で組織内のアクセスを管理することができます。Outpost へのアクセスを必要とするすべてのアカウントは、AWS Organizations の所有者アカウントと同じ組織内になければなりません。
- S3 on Outposts のバケット所有者アカウントは、常にバケット内のすべてのオブジェクトの所有者です。

- バケットに対し操作ができるのは、S3 on Outpostsバケット所有者アカウントのみです。
- オブジェクトのサイズの制限は、Amazon S3 と同じです。
- S3 on Outposts に保存されるすべてのオブジェクトは、OUTPOSTS ストレージクラスに保存されます。
- デフォルトでは、OUTPOSTS ストレージクラスに保存されているすべてのオブジェクトは、Amazon S3 マネージド暗号化キー (SSE-S3) によるサーバー側の暗号化を使用して保存されます。また、ユーザーが用意した暗号化キー (SSE-C) で、サーバー側の暗号化を使用してオブジェクトを保存するように明示的に選択することもできます。
- Outpost にオブジェクトを保存する十分なスペースがない場合、API は容量不足の例外 (ICE) を返します。

## S3 on Outposts でサポートされている API のオペレーション

S3 on Outposts でサポートされている API オペレーションのリストについては、「[Amazon S3 on Outposts の API オペレーション](#)」を参照してください。

## S3 on Outposts でサポートされている Amazon S3 AWS CLI コマンド

以下の Amazon S3 AWS CLI コマンドは、現時点では Amazon S3 on Outposts でサポートされていません。詳細については、AWS CLI コマンドリファレンスの「[使用できるコマンド](#)」を参照してください。

- 同じ Outposts バケット内の [cp](#)、[mv](#)、[sync](#)、またはローカル環境と Outposts バケット間。
- [ls](#)
- [presign](#)
- [rm](#)

## S3 on Outposts でサポートされていない Amazon S3 の機能

Amazon S3 の以下の機能は、現時点では Amazon S3 on Outposts でサポートされていません。使用しようとしても、拒否されます。

- 条件付きリクエスト

- アクセスコントロールリスト (ACL)
- Cross-Origin Resource Sharing (CORS)
- S3 バッチオペレーション
- S3 インベントリレポート
- デフォルトのバケット暗号化の変更
- パブリックバケット
- 多要素認証 (MFA) Delete
- S3 ライフサイクルの移行 (オブジェクトの削除と不完全なマルチパートアップロードの停止を除く)
- S3 オブジェクトロックのリーガルホールド
- オブジェクトロックの保持
- AWS Key Management Service (AWS KMS) キー (SSE-KMS) によるサーバー側の暗号化
- S3 Replication Time Control (S3 RTC)
- Amazon CloudWatch のリクエストメトリクス
- メトリクスの構成
- Transfer Acceleration
- S3 イベント通知
- リクエスト支払いバケットに、
- S3 Select
- AWS Lambda のイベント
- サーバーアクセスのログ記録
- HTTP POST リクエスト
- SOAP
- ウェブサイトへのアクセス

## S3 on Outposts のネットワーク要件

- S3 on Outposts のアクセスポイントにリクエストをルーティングするには、S3 on Outposts エンドポイントを作成し設定する必要があります。S3 on Outposts のエンドポイントには、次の制限が適用されます。

- Outposts の各仮想プライベートクラウド (VPC) には、エンドポイントが 1 つ関連付けられており、Outpost 1 つにつき最大 100 のエンドポイントを設定できます。
- 複数のアクセスポイントを同じエンドポイントにマッピングできます。
- エンドポイントは、次の CIDR 範囲のサブスペースに CIDR ブロックがある VPC にのみ追加できます。
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16
- Outpost へのエンドポイントは、CIDR ブロックが重複しない VPC からのみ作成できます。
- エンドポイントは、その Outposts サブネット内からのみ作成できます。
- エンドポイントの作成に使用するサブネットには、S3 on Outposts で使用するための 4 つの IP アドレスが含まれている必要があります。
- ユーザー所有の IP アドレスプール (CoIP プール) を指定する場合は、S3 on Outposts で使用する 4 つの IP アドレスが含まれている必要があります。
- VPC ごとに作成できるエンドポイントは、1 つの Outpost あたり 1 つだけです。

# Amazon S3 on Outposts の開始方法

Amazon S3 on Outposts を使用すると、AWS Outposts で S3 バケットを作成し、ローカルデータアクセス、ローカルデータ処理、データレジデンシーを必要とするアプリケーション用に、オンプレミスのオブジェクトを簡単に保存および取得できます。S3 on Outposts は、新しいストレージクラス、S3 Outposts (OUTPOSTS) を提供し、これは Amazon S3 API を使用し、AWS Outposts 上の複数のデバイスやサーバー間でデータを永続的かつ冗長的に保存するように設計されています。仮想プライベートクラウド (VPC) を介したアクセスポイントとエンドポイント接続を使用して、Outposts バケットと通信します。Outposts バケットでは、Amazon S3 と同じ API と機能 (アクセスポリシー、暗号化、タグ付けなど) を使用できます。AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDK、または REST API を使用して S3 on Outposts を使用できます。

Amazon S3 on Outposts を使用すると、Amazon S3 と同様に、オブジェクトストレージ、アクセスポリシー、暗号化、タグ付けなどの Amazon S3 API と機能を AWS Outposts で利用できます。S3 on Outposts の詳細については、「[Amazon S3 on Outposts とは](#)」を参照してください。

## トピック

- [AWS マネジメントコンソール を使用した開始方法](#)
- [AWS CLI および SDK for Java の使用開始](#)

## AWS マネジメントコンソール を使用した開始方法

Amazon S3 on Outposts を使用すると、AWS Outposts で S3 バケットを作成し、ローカルデータアクセス、ローカルデータ処理、データレジデンシーを必要とするアプリケーション用に、オンプレミスのオブジェクトを簡単に保存および取得できます。S3 on Outposts は、新しいストレージクラス、S3 Outposts (OUTPOSTS) を提供し、これは Amazon S3 API を使用し、AWS Outposts 上の複数のデバイスやサーバー間でデータを永続的かつ冗長的に保存するように設計されています。仮想プライベートクラウド (VPC) を介したアクセスポイントとエンドポイント接続を使用して、Outposts バケットと通信します。Outposts バケットでは、Amazon S3 と同じ API と機能 (アクセスポリシー、暗号化、タグ付けなど) を使用できます。AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDK、または REST API を使用して S3 on Outposts を使用できます。詳細については、「[Amazon S3 on Outposts とは](#)」を参照してください。

コンソールを使用して S3 on Outposts の使用を開始するには、以下のトピックを参照してください。AWS CLI または AWS SDK for Java を使って開始するには、「[AWS CLI および SDK for Java の使用開始](#)」を参照してください。

## トピック

- [バケット、アクセスポイント、およびエンドポイントを作成する](#)
- [次のステップ](#)

## バケット、アクセスポイント、およびエンドポイントを作成する

以下の手順は、S3 on Outposts で最初のバケットを作成する方法を示しています。コンソールを使用してバケットを作成するときには、バケットに関連付けられたアクセスポイントとエンドポイントも作成します。これにより、バケットへのオブジェクトの保存をすぐに開始できます。

1. AWS マネジメントコンソール にサインインし、Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. [Outposts バケットの作成] を選択します。
4. [バケット名] に、バケットのドメインネームシステム (DNS) に準拠する名前を入力します。

バケット名には次の条件があります。

- AWS アカウント、Outpost、および Outpost が属する AWS リージョン 内で一意であること。
- 3~63 文字である必要があります。
- 大文字を含めないでください。
- 先頭の文字には小文字の英文字または数字を使用する。

バケットを作成したら、その名前を変更することはできません。バケット名の詳細については、「Amazon S3 ユーザーガイド」の「[バケットの名前付け](#)」を参照してください。

### Important

バケット名にアカウント番号などの機密情報を含めないでください。バケット名は、バケット内のオブジェクトを参照する URL に表示されます。

5. [Outpost] で、バケットを配置する Outpost を選択します。

6. [Bucket Versioning] (バケットのバージョンニング) で、S3 on Outposts バケットの S3 バージョニングの状態を次のいずれかのオプションに設定します。
  - [Disable] (無効化) (デフォルト) — バケットはバージョンニング無効のままです。
  - [Enable] (有効化) — バケット内のオブジェクトの S3 バージョニングを有効にします。バケットに追加されたすべてのオブジェクトは、一意のバージョン ID を受け取ります。

S3 バージョニングの詳細については、[S3 on Outposts バケットの S3 バージョニングの管理](#) を参照してください。

7. (オプション) Outposts バケットに関連付けるオプションのタグを追加します。タグを使用して、個々のプロジェクトまたはプロジェクトのグループの基準を追跡したり、コスト配分タグを使用してバケットにラベルを付けたりできます。

デフォルトでは、Outpost バケットに保存されているすべてのオブジェクトは、Amazon S3 マネージド暗号化キー (SSE-S3) によるサーバー側の暗号化を使用して保存されます。また、ユーザーが用意した暗号化キー (SSE-C) で、サーバー側の暗号化を使用してオブジェクトを保存するように明示的に選択することもできます。暗号化タイプを変更するには、REST API、AWS Command Line Interface (AWS CLI)、または AWS SDK を使用する必要があります。

8. [Outposts access point settings] (Outposts アクセスポイントの設定) セクションで、アクセスポイント名を入力します。

S3 on Outposts アクセスポイントは、S3 on Outposts の共有データセットへの大規模なデータアクセスの管理を簡素化します。アクセスポイントは、Outposts バケットにアタッチされた名前付きのネットワークエンドポイントで、S3 オブジェクトのオペレーションを実行するために使用できます。詳細については、「[アクセスポイント](#)」を参照してください。

アクセスポイント名は、このリージョンおよび Outpost のアカウント内で一意であり、「[アクセスポイントの制約と制限](#)」に準拠している必要があります。

9. この Amazon S3 on Outposts アクセスポイントの VPC を選択します。

VPC がない場合は、[Create VPC] (VPC を作成) を選択します。詳細については、「Amazon S3 ユーザーガイド」の「[仮想プライベートクラウド \(VPC\) に制限されたアクセスポイントの作成](#)」を参照してください。

仮想プライベートクラウド (VPC) を使用すると、定義した仮想ネットワークに AWS リソースを起動できます。仮想ネットワークは、お客様自身のデータセンターで運用されていた従来の

ネットワークによく似ていますが、のスケラブルなインフラストラクチャを使用できるというメリットがありますAWS

10. (既存の VPC の場合はオプション) エンドポイントの [Endpoint subnet] (エンドポイントサブネット) を選択します。

サブネットは、VPC の IP アドレスの範囲です。必要なサブネットがない場合は、[Create subnet] (サブネットの作成) を選択します。詳細については、「[S3 on Outposts のネットワーキング](#)」を参照してください。

11. (既存の VPC の場合はオプション) エンドポイントの [Endpoint security group] (エンドポイントセキュリティグループ) を選択します。

[セキュリティグループ](#)は、仮想ファイアウォールとして機能し、インバウンドトラフィックとアウトバウンドトラフィックをコントロールします。

12. (既存の VPC の場合はオプション) [Endpoint access type] (エンドポイントアクセスタイプ) を選択します。

- プライベート — VPC と共に使用します。
- お客様所有 IP – オンプレミスネットワーク内からユーザー所有の IP アドレスプール (CoIP プール) と共に使用します。

13. (オプション) [Outpost access point policy] (Outpost アクセスポイントポリシー) を指定します。コンソールに、アクセスポイントの Amazon リソースネーム (ARN) が自動的に表示されます。これをポリシーで使用できます。

14. [Outposts バケットの作成] を選択します。

#### Note

Outpost エンドポイントが作成され、バケットが使用できるようになるまでに最長で 5 分かかることがあります。追加のバケット設定を構成するには、[View details] (詳細の表示) を選択します。

## 次のステップ

Amazon S3 on Outposts では、オブジェクトデータは常に Outpost に保存されます。AWS が Outpost ラックを設置すると、データの常駐要件を満たすために、データは Outpost にローカルに保たれます。オブジェクトが Outpost を離れたり、AWS リージョン 外に出たりすることはありません。AWS マネジメントコンソールはリージョン内でホストされるため、コンソールを使用して

Outpost にオブジェクトをアップロードしたり、管理したりすることはできません。ただし REST API、AWS Command Line Interface (AWS CLI)、および AWS SDK を使用して、アクセスポイントを介してオブジェクトのアップロードと管理を行うことができます。

S3 on Outposts バケット、アクセスポイント、およびエンドポイントを作成した後、AWS CLI または SDK for Java を使用して、オブジェクトをバケットにアップロードできます。詳細については、「[S3 on Outposts バケットにオブジェクトをアップロードする](#)」を参照してください。

## AWS CLI および SDK for Java の使用開始

Amazon S3 on Outposts を使用すると、AWS Outposts で S3 バケットを作成し、ローカルデータアクセス、ローカルデータ処理、データレジデンシーを必要とするアプリケーション用に、オンプレミスのオブジェクトを簡単に保存および取得できます。S3 on Outposts は、新しいストレージクラス、S3 Outposts (OUTPOSTS) を提供し、これは Amazon S3 API を使用し、AWS Outposts 上の複数のデバイスやサーバー間でデータを永続的かつ冗長的に保存するように設計されています。仮想プライベートクラウド (VPC) を介したアクセスポイントとエンドポイント接続を使用して、Outposts バケットと通信します。Outposts バケットでは、Amazon S3 と同じ API と機能 (アクセスポリシー、暗号化、タグ付けなど) を使用できます。AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDK、または REST API を使用して S3 on Outposts を使用できます。詳細については、「[Amazon S3 on Outposts とは](#)」を参照してください。

S3 on Outposts の使用を開始するには、バケット、アクセスポイント、およびエンドポイントを作成する必要があります。その後、バケットにオブジェクトをアップロードできます。以下の例は、AWS CLI および SDK for Java を使用して、S3 on Outposts の使用を開始する方法を示しています。コンソールを使用して開始するには、「[AWS マネジメントコンソールを使用した開始方法](#)」を参照してください。

### トピック

- [ステップ 1: バケットを作成する](#)
- [ステップ 2: アクセスポイントの作成](#)
- [ステップ 3: エンドポイントを作成する](#)
- [ステップ 4: S3 on Outposts バケットにオブジェクトをアップロードする](#)

## ステップ 1: バケットを作成する

以下の AWS CLI および SDK for Java の例は、S3 on Outposts バケットを作成する方法を示しています。

## AWS CLI

### Example

次の例では、AWS CLI を使用して S3 on Outposts バケット (s3-outposts:CreateBucket) を作成します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control create-bucket --bucket example-outposts-bucket --outpost-id op-01ac5d28a6a232904
```

## SDK for Java

### Example

AWS SDK for Java を使用して S3 Outposts バケットを作成する方法の例については、「AWS SDK for Java 2.x のコード例」の「[CreateOutpostsBucket.java](#)」を参照してください。

## ステップ 2: アクセスポイントの作成

Amazon S3 on Outposts バケットにアクセスするには、アクセスポイントを作成して設定する必要があります。以下の例は、AWS CLI および SDK for Java を使用して、アクセスポイントを作成する方法を示しています。

アクセスポイントは、Amazon S3 の共有データセットへの大規模なデータアクセスの管理を簡素化します。アクセスポイントは、バケットにアタッチされた名前付きのネットワークエンドポイントで、Amazon S3 オブジェクトのオペレーション (GetObject や PutObject など) を実行するために使用できます。S3 on Outposts では、Outposts バケット内の任意のオブジェクトにアクセスするにはアクセスポイントを使用する必要があります。アクセスポイントでは、仮想ホスト形式のアドレス指定のみがサポートされます。

## AWS CLI

### Example

次の AWS CLI の例では、Outposts バケットのアクセスポイントを作成します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control create-access-point --account-id 123456789012 --name example-outposts-access-point --bucket "arn:aws:s3-
```

```
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-  
bucket" --vpc-configuration VpcId=example-vpc-12345
```

## SDK for Java

### Example

AWS SDK for Java を使用して S3 Outposts バケットのアクセスポイントを作成する方法の例については、「AWS SDK for Java 2.x のコード例」の「[CreateOutpostsAccessPoint.java](#)」を参照してください。

## ステップ 3: エンドポイントを作成する

Amazon S3 on Outposts のアクセスポイントにリクエストをルーティングするには、S3 on Outposts エンドポイントを作成し設定する必要があります。エンドポイントを作成するには、Outposts のホームリージョンへのサービスリンクとのアクティブな接続が必要です。Outpost 上の各仮想プライベートクラウド (VPC) に 1 つのエンドポイントに関連付けることができます。エンドポイントクォータの詳細については、[S3 on Outposts のネットワーク要件](#) を参照してください。Outposts バケットにアクセスしてオブジェクトオペレーションを実行できるようにするには、エンドポイントを作成する必要があります。詳細については、「[エンドポイント](#)」を参照してください。

以下の例は、AWS CLI および SDK for Java を使用して、エンドポイントを作成する方法を示しています。エンドポイントの作成と管理に必要な許可の詳細については、「[S3 on Outposts エンドポイントの許可](#)」を参照してください。

### AWS CLI

#### Example

次の AWS CLI の例では、VPC リソースアクセスタイプを使用して、Outpost のエンドポイントを作成します。VPC はサブネットから派生します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id  
subnet-8c7a57c5 --security-group-id sg-ab19e0d1
```

次の AWS CLI の例では、アクセスタイプにユーザー所有の IP アドレスプール (CoIP プール) を使用して、Outpost のエンドポイントを作成します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id  
subnet-8c7a57c5 --security-group-id sg-ab19e0d1 --access-type CustomerOwnedIp --  
customer-owned-ipv4-pool ipv4pool-coip-12345678901234567
```

## SDK for Java

### Example

AWS SDK for Java を使用して S3 Outpost のエンドポイントを作成する方法の例については、「AWS SDK for Java 2.x のコード例」の「[CreateOutpostsEndPoint.java](#)」を参照してください。

## ステップ 4: S3 on Outposts バケットにオブジェクトをアップロードする

オブジェクトをアップロードするには、「[S3 on Outposts バケットにオブジェクトをアップロードする](#)」を参照してください。

# S3 on Outposts のネットワーキング

Amazon S3 on Outposts を使用して、ローカルのデータアクセス、データ処理、およびデータレジデンシーを必要とするアプリケーションのために、オンプレミスでオブジェクトの保存と取得ができます。このセクションでは、S3 on Outposts にアクセスするためのネットワーク要件について説明します。

## トピック

- [ネットワークアクセスタイプの選択](#)
- [S3 on Outposts のバケットおよびオブジェクトにアクセスする](#)
- [クロスアカウント Elastic Network Interface](#)

## ネットワークアクセスタイプの選択

Outposts の S3 には、VPC 内またはオンプレミスネットワークからアクセスできます。アクセスポイントとエンドポイント接続を使用して、Outposts バケットと通信します。この接続は、AWS ネットワーク内における VPC と S3 on Outposts バケット間のトラフィックを維持します。エンドポイントを作成するときには、エンドポイントアクセスタイプを Private (VPC ルーティングの場合) または CustomerOwnedIp (CoIP プールの場合) のどちらかに指定する必要があります。

- Private (VPC ルーティングの場合) - アクセスタイプを指定しなかった場合、S3 on Outposts はデフォルトで Private を使用します。Private アクセスタイプでは、VPC のインスタンスは、Outposts のリソースと通信するためにパブリック IP アドレスを必要としません。VPC 内から S3 on Outposts を操作できます。このタイプのエンドポイントは、直接的な VPC ルーティングを通じてオンプレミスネットワークからはアクセスできません。詳細については、「AWS Outposts ユーザーガイド」の「[ローカルゲートウェイテーブル](#)」を参照してください。
- CustomerOwnedIp (CoIP プールの場合) — Private アクセスタイプをデフォルト設定せず、CustomerOwnedIp を選択しなかった場合、IP アドレス範囲を指定する必要があります。このアクセスタイプを使用して、オンプレミスネットワークと VPC 内の両方から S3 on Outposts を操作できます。VPC 内の S3 on Outposts にアクセスする場合、トラフィックはローカルゲートウェイの帯域幅に制限されます。

## S3 on Outposts のバケットおよびオブジェクトにアクセスする

S3 on Outposts バケットおよびオブジェクトにアクセスするには、次のものがが必要です。

- VPC のアクセスポイント。
- 同じ VPC のエンドポイント。
- Outpost と AWS リージョン 間のアクティブな接続。Outpost をリージョンに接続する方法の詳細については、AWSOutposts ユーザーガイドの[AWS リージョンへの Outpost 接続](#)を参照してください。

S3 on Outposts 内のバケットおよびオブジェクトへのアクセスの詳細については、「[S3 on Outposts バケットの操作](#)」と「[S3 on Outposts オブジェクトの操作](#)」を参照してください。

## クロスアカウント Elastic Network Interface

S3 on Outposts エンドポイントは、Amazon リソースネーム (ARN) を持つ名前付きリソースです。これらのエンドポイントが作成されると、AWS Outposts は複数のクロスアカウント Elastic Network Interface を設定します。S3 on Outposts クロスアカウント Elastic Network Interface は、他のネットワークインターフェイスと同様ですが、1 つの例外があり、S3 on Outposts は、クロスアカウント Elastic Network Interface を Amazon EC2 インスタンスに関連付けます。

S3 on Outposts のドメインネームシステム (DNS) により、クロスアカウント Elastic Network Interface を介してリクエストのロードバランシングが行われます。S3 on Outposts は、クロスアカウント Elastic Network Interface を AWS アカウント内に作成します。これは、Amazon EC2 コンソールの [Network Interface] (ネットワークインターフェイス) ペインで表示できます。

CoIP プールアクセスタイプを使用するエンドポイントの場合、S3 on Outposts は、設定された CoIP プールから IP アドレスをクロスアカウント Elastic Network Interface に割り当てて関連付けます。

# S3 on Outposts バケットの操作

Amazon S3 on Outposts を使用すると、AWS Outposts に S3 バケットを作成して、ローカルデータアクセス、ローカルデータ処理、およびデータレジデンシーを必要とするアプリケーション用に、オブジェクトの保存と取得がオンプレミスで容易にできます。S3 on Outposts は、新しいストレージクラス、S3 Outposts (OUTPOSTS) を提供し、これは Amazon S3 API を使用し、AWS Outposts 上の複数のデバイスやサーバー間でデータを永続的かつ冗長的に保存するように設計されています。Outposts バケットでは、アクセスポリシー、暗号化、タグ付けなど、Amazon S3 と同じ API と機能を使用できます。詳細については、「[Amazon S3 on Outposts とは](#)」を参照してください。

仮想プライベートクラウド (VPC) を介したアクセスポイントおよびエンドポイント接続を使用して、Outposts バケットと通信します。S3 on Outposts バケットおよびオブジェクトにアクセスするには、VPC 用のアクセスポイントと、同じ VPC のエンドポイントが必要です。詳細については、「[S3 on Outposts のネットワーキング](#)」を参照してください。

## バケット

S3 on Outposts では、バケット名は Outpost で一意であり、バケットを識別するために、Outpost が属するリージョンの AWS リージョン コード、AWS アカウント ID、Outpost ID、およびバケット名が必要です。

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/bucket/bucket-name
```

詳細については、「[S3 on Outposts のリソース ARN](#)」を参照してください。

## アクセスポイント

Amazon S3 on Outposts は、Outposts バケットにアクセスする唯一の手段として、Virtual Private Cloud (VPC) のみのアクセスポイントをサポートします。

アクセスポイントは、Amazon S3 の共有データセットへの大規模なデータアクセスの管理を簡素化します。アクセスポイントは、バケットにアタッチされた名前付きのネットワークエンドポイントで、Amazon S3 オブジェクトのオペレーション (GetObject や PutObject など) を実行するために使用できます。S3 on Outposts では、Outposts バケット内の任意のオブジェクトにアクセスするにはアクセスポイントを使用する必要があります。アクセスポイントでは、仮想ホスト形式のアドレス指定のみがサポートされます。

以下の例は、S3 on Outposts アクセスポイントの ARN 形式を示しています。アクセスポイント ARN には、Outpost が属するリージョンの AWS リージョン コード、AWS アカウント ID、Outpost ID、およびアクセスポイント名が含まれます。

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

## エンドポイント

S3 on Outposts のアクセスポイントにリクエストをルーティングするには、S3 on Outposts エンドポイントを作成し設定する必要があります。S3 on Outposts エンドポイントを使用すると、VPC を Outposts バケットにプライベートに接続できます。Outposts エンドポイント上の S3 は、Outposts バケット上の S3 へのエン트리ポイントの仮想 Uniform Resource Identifier (URI) です。これらは水平にスケールされ、冗長で、可用性の高い VPC コンポーネントです。

Outposts の各仮想プライベートクラウド (VPC) には、エンドポイントが 1 つ関連付けられており、Outpost 1 つにつき最大 100 のエンドポイントを設定できます。Outposts バケットにアクセスしてオブジェクト操作を実行できるようにするには、これらのエンドポイントを作成する必要があります。これらのエンドポイントを作成することにより、S3 と S3 on Outposts で同じオペレーションを実行できるようにすることで、API モデルと動作を同じにすることもできます。

## S3 on Outposts の API オペレーション

Outposts バケット API オペレーションを管理するために、S3 on Outposts は、Amazon S3 エンドポイントとは別のエンドポイントをホストします。このエンドポイントは `s3-outposts.region.amazonaws.com` です。

Amazon S3 API オペレーションを使用するには、正しい ARN 形式を使用してバケットとオブジェクトに署名する必要があります。リクエストが Amazon S3 (`s3-control.region.amazonaws.com`) または S3 on Outposts (`s3-outposts.region.amazonaws.com`) のいずれに対するものであるかを Amazon S3 が判断できるように、API オペレーションに ARN を渡す必要があります。ARN 形式に基づき、その後、S3 はリクエストに署名し、適切にルーティングできます。

リクエストが Amazon S3 コントロールプレーンに送信されるたびに、SDK は ARN からコンポーネントを抽出し、ARN から抽出された `outpost-id` 値と共に追加のヘッダー `x-amz-outpost-id` を含めます。リクエストは、S3 on Outposts エンドポイントにルーティングされる前に、ARN からのサービス名を使用して署名されます。この動作は、`s3control` クライアントが扱うすべての API オペレーションに適用されます。

次の表は、Amazon S3 on Outposts の拡張 API オペレーションと、Amazon S3 に関する変更点の一覧です。

API	S3 on Outposts パラメータ値
CreateBucket	ARN、Outpost ID としてのバケット名
ListRegionalBuckets	Outpost ID
DeleteBucket	ARN としてのバケット名
DeleteBucketLifecycleConfiguration	ARN としてのバケット名
GetBucketLifecycleConfiguration	ARN としてのバケット名
PutBucketLifecycleConfiguration	ARN としてのバケット名
GetBucketPolicy	ARN としてのバケット名
PutBucketPolicy	ARN としてのバケット名
DeleteBucketPolicy	ARN としてのバケット名
GetBucketTagging	ARN としてのバケット名
PutBucketTagging	ARN としてのバケット名
DeleteBucketTagging	ARN としてのバケット名
CreateAccessPoint	ARN としてのアクセスポイント名
DeleteAccessPoint	ARN としてのアクセスポイント名
GetAccessPoint	ARN としてのアクセスポイント名
GetAccessPoint	ARN としてのアクセスポイント名

API	S3 on Outposts パラメータ値
ListAccessPoints	ARN としてのアクセスポイント名
PutAccessPointPolicy	ARN としてのアクセスポイント名
GetAccessPointPolicy	ARN としてのアクセスポイント名
DeleteAccessPointPolicy	ARN としてのアクセスポイント名

## S3 on Outposts バケットを作成および管理する

S3 on Outposts バケットの作成と管理の詳細については、以下のトピックを参照してください。

### S3 on Outposts バケットを作成する

Amazon S3 on Outposts を使用すると、AWS Outposts で S3 バケットを作成し、ローカルデータアクセス、ローカルデータ処理、データレジデンシーを必要とするアプリケーション用に、オンプレミスのオブジェクトを簡単に保存および取得できます。S3 on Outposts は、新しいストレージクラス、S3 Outposts (OUTPOSTS) を提供し、これは Amazon S3 API を使用し、AWS Outposts 上の複数のデバイスやサーバー間でデータを永続的かつ冗長的に保存するように設計されています。仮想プライベートクラウド (VPC) を介したアクセスポイントとエンドポイント接続を使用して、Outposts バケットと通信します。Outposts バケットでは、Amazon S3 と同じ API と機能 (アクセスポリシー、暗号化、タグ付けなど) を使用できます。AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDK、または REST API を使用して S3 on Outposts を使用できます。詳細については、「[Amazon S3 on Outposts とは](#)」を参照してください。

#### Note

バケットを作成する AWS アカウント がそのバケットを所有し、アクションをコミットできる唯一のアカウントです。バケットには、Outposts、タグ、デフォルトの暗号化、アクセスポイント設定などの設定プロパティがあります。アクセスポイント設定には、仮想プライベートクラウド (VPC)、バケット内のオブジェクトにアクセスするためのアクセスポイントポリシー、およびその他のメタデータが含まれます。詳細については、「[S3 on Outposts の仕様](#)」を参照してください。

AWS PrivateLink を使用して、仮想プライベートクラウド (VPC) のインターフェイス VPC エンドポイント経由でバケットやエンドポイントの管理アクセスを提供するバケットを作成する場合は、「[S3 on Outposts の AWS PrivateLink](#)」を参照してください。

次の例は、AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、および AWS SDK for Java を使用して、S3 on Outposts バケットを作成する方法を示しています。

## S3 コンソールの使用

1. AWS マネジメントコンソール にサインインし、Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. [Outposts バケットの作成] を選択します。
4. [バケット名] に、バケットのドメインネームシステム (DNS) に準拠する名前を入力します。

バケット名には次の条件があります。

- AWS アカウント、Outpost、および Outpost が属する AWS リージョン 内で一意であること。
- 3~63 文字である必要があります。
- 大文字を含めないでください。
- 先頭の文字には小文字の英文字または数字を使用する。

バケットを作成したら、その名前を変更することはできません。バケット名の詳細については、「Amazon S3 ユーザーガイド」の「[バケットの名前付け](#)」を参照してください。

### Important

バケット名にアカウント番号などの機密情報を含めないでください。バケット名は、バケット内のオブジェクトを参照する URL に表示されます。

5. [Outpost] で、バケットを配置する Outpost を選択します。
6. [Bucket Versioning] (バケットのバージョニング) で、S3 on Outposts バケットの S3 バージョニングの状態を次のいずれかのオプションに設定します。
  - [Disable] (無効化) (デフォルト) — バケットはバージョニング無効のままです。

- [Enable] (有効化) — バケット内のオブジェクトの S3 バージョニングを有効にします。バケットに追加されたすべてのオブジェクトは、一意のバージョン ID を受け取ります。

S3 バージョニングの詳細については、[S3 on Outposts バケットの S3 バージョニングの管理](#) を参照してください。

7. (オプション) Outposts バケットに関連付けるオプションのタグを追加します。タグを使用して、個々のプロジェクトまたはプロジェクトのグループの基準を追跡したり、コスト配分タグを使用してバケットにラベルを付けたりできます。

デフォルトでは、Outpost バケットに保存されているすべてのオブジェクトは、Amazon S3 マネージド暗号化キー (SSE-S3) によるサーバー側の暗号化を使用して保存されます。また、ユーザーが用意した暗号化キー (SSE-C) で、サーバー側の暗号化を使用してオブジェクトを保存するように明示的に選択することもできます。暗号化タイプを変更するには、REST API、AWS Command Line Interface (AWS CLI)、または AWS SDK を使用する必要があります。

8. [Outposts access point settings] (Outposts アクセスポイントの設定) セクションで、アクセスポイント名を入力します。

S3 on Outposts アクセスポイントは、S3 on Outposts の共有データセットへの大規模なデータアクセスの管理を簡素化します。アクセスポイントは、Outposts バケットにアタッチされた名前付きのネットワークエンドポイントで、S3 オブジェクトのオペレーションを実行するために使用できます。詳細については、「[アクセスポイント](#)」を参照してください。

アクセスポイント名は、このリージョンおよび Outpost のアカウント内で一意であり、「[アクセスポイントの制約と制限](#)」に準拠している必要があります。

9. この Amazon S3 on Outposts アクセスポイントの VPC を選択します。

VPC がない場合は、[Create VPC] (VPC を作成) を選択します。詳細については、「Amazon S3 ユーザーガイド」の「[仮想プライベートクラウド \(VPC\) に制限されたアクセスポイントの作成](#)」を参照してください。

仮想プライベートクラウド (VPC) を使用すると、定義した仮想ネットワークに AWS リソースを起動できます。仮想ネットワークは、お客様自身のデータセンターで運用されていた従来のネットワークによく似ていますが、のスケラブルなインフラストラクチャを使用できるというメリットがありますAWS

10. (既存の VPC の場合はオプション) エンドポイントの [Endpoint subnet] (エンドポイントサブネット) を選択します。

サブネットは、VPC の IP アドレスの範囲です。必要なサブネットがない場合は、[Create subnet] (サブネットの作成) を選択します。詳細については、「[S3 on Outposts のネットワーキング](#)」を参照してください。

11. (既存の VPC の場合はオプション) エンドポイントの [Endpoint security group] (エンドポイントセキュリティグループ) を選択します。

[セキュリティグループ](#)は、仮想ファイアウォールとして機能し、インバウンドトラフィックとアウトバウンドトラフィックをコントロールします。

12. (既存の VPC の場合はオプション) [Endpoint access type] (エンドポイントアクセスタイプ) を選択します。

- プライベート — VPC と共に使用します。
- お客様所有 IP – オンプレミスネットワーク内からユーザー所有の IP アドレスプール (CoIP プール) と共に使用します。

13. (オプション) [Outpost access point policy] (Outpost アクセスポイントポリシー) を指定します。コンソールに、アクセスポイントの Amazon リソースネーム (ARN) が自動的に表示されます。これをポリシーで使用できます。

14. [Outposts バケットの作成] を選択します。

#### Note

Outpost エンドポイントが作成され、バケットが使用できるようになるまでに最長で 5 分かかることがあります。追加のバケット設定を構成するには、[View details] (詳細の表示) を選択します。

## AWS CLI を使用する場合

### Example

次の例では、AWS CLI を使用して S3 on Outposts バケット (s3-outposts:CreateBucket) を作成します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control create-bucket --bucket example-outposts-bucket --outpost-id op-01ac5d28a6a232904
```

## AWS SDK for Java の使用

### Example

AWS SDK for Java を使用して S3 Outposts バケットを作成する方法の例については、「AWS SDK for Java 2.x のコード例」の「[CreateOutpostsBucket.java](#)」を参照してください。

## S3 on Outposts バケットのタグの追加

Amazon S3 on Outposts バケットにタグを追加して、個々のプロジェクトまたはプロジェクトのグループのストレージコストおよび他の基準を追跡できます。

### Note

バケットを作成する AWS アカウント は、バケットを所有しており、タグを変更できる唯一のアカウントです。

### S3 コンソールの使用

1. AWS マネジメントコンソール にサインインし、Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. タグを編集する Outposts バケットを選択します。
4. [プロパティ] タブを選択します。
5. タグで、[Edit] を選択します。
6. [Add new tag] (新しいタグの追加) を選択し、[Key] (キー) とオプションの [Value] (値) を入力します。

個々のプロジェクトまたはプロジェクトグループについて、その他の基準を追跡するために、Outposts バケットに関連付けるタグを追加します。

7. [Save changes] (変更の保存) をクリックします。

## AWS CLI の使用

次の AWS CLI の例では、タグを指定する JSON ドキュメント (*tagging.json*) を現在のフォルダーで使用して、S3 on Outposts バケットにタグ付け設定を適用しています。この例を実行するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。

```
aws s3control put-bucket-tagging --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket --tagging file://tagging.json
```

*tagging.json*

```
{
  "TagSet": [
    {
      "Key": "organization",
      "Value": "marketing"
    }
  ]
}
```

次の AWS CLI の例では、コマンドラインから直接、S3 on Outposts バケットにタグ付け設定を適用しています。

```
aws s3control put-bucket-tagging --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket --tagging 'TagSet=[{Key=organization,Value=marketing}]'
```

このコマンドの詳細については、「AWS CLI リファレンス」の「[put-bucket-tagging](#)」を参照してください。

## バケットポリシーを使用して Amazon S3 on Outposts バケットへのアクセスを管理する

バケットポリシーは、リソースベースの AWS Identity and Access Management (IAM) ポリシーを使用して、バケットとその中のオブジェクトへのアクセス許可を付与できます。バケット所有者のみが、ポリシーをバケットに関連付けることができます。バケットに添付された許可は、バケット所有者が所有するバケットのすべてのオブジェクトに適用されます。バケットポリシーのサイズは 20 KB に制限されています。詳細については、「[バケットポリシー](#)」を参照してください。

バケットポリシーを更新して、Amazon S3 on Outposts バケットへのアクセスを管理できます。詳細については、以下のトピックを参照してください。

## トピック

- [Amazon S3 on Outposts バケットのバケットポリシーを追加または編集する](#)
- [Amazon S3 on Outposts バケットのバケットポリシーを表示する](#)
- [Amazon S3 on Outposts バケットのバケットポリシーを削除する](#)
- [バケットポリシーの例](#)

## Amazon S3 on Outposts バケットのバケットポリシーを追加または編集する

バケットポリシーは、リソースベースの AWS Identity and Access Management (IAM) ポリシーを使用して、バケットとその中のオブジェクトへのアクセス許可を付与できます。バケット所有者のみが、ポリシーをバケットに関連付けることができます。バケットに添付された許可は、バケット所有者が所有するバケットのすべてのオブジェクトに適用されます。バケットポリシーのサイズは 20 KB に制限されています。詳細については、「[バケットポリシー](#)」を参照してください。

以下のトピックでは、AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、または AWS SDK for Java を使用して、Amazon S3 on Outposts バケットポリシーを更新する方法を示します。

### S3 コンソールの使用

バケットポリシーを作成または編集するには

1. AWS マネジメントコンソール にサインインし、Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. バケットポリシーを編集する Outposts バケットを選択します。
4. [アクセス許可] タブを選択します。
5. [Outposts bucket policy] (Outposts バケットポリシー) セクションで、新しいポリシーを作成または編集するには、[Edit] (編集) を選択します。

これで、S3 on Outposts バケットポリシーを追加または編集できます。詳細については、「[S3 on Outposts で IAM を設定する](#)」を参照してください。

## AWS CLI の使用

次の AWS CLI の例では、Outposts バケットにポリシーを配置します。

1. 以下のバケットポリシーを JSON ファイルに保存します。この例では、ファイル名は `policy1.json` です。*user input placeholders* を、ユーザー自身の情報に置き換えます。

JSON

```
{
  "Version": "2012-10-17",
  "Id": "testBucketPolicy",
  "Statement": [
    {
      "Sid": "st1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": [
        "s3-outposts:GetObject",
        "s3-outposts:PutObject",
        "s3-outposts:DeleteObject",
        "s3-outposts:ListBucket"
      ],
      "Resource": "arn:aws:s3-outposts:us-east-1:123456789012:outpost/op-01ac5d28a6a232904/bucket/amzn-s3-demo-bucket"
    }
  ]
}
```

2. `put-bucket-policy` CLI コマンドの一部として JSON ファイルを送信します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control put-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket --policy file://policy1.json
```

## AWS SDK for Java の使用

次の SDK for Java の例では、Outposts バケットにポリシーを配置します。

```
import com.amazonaws.services.s3control.model.*;

public void putBucketPolicy(String bucketArn) {

    String policy = "{\"Version\":\"2012-10-17\",\"Id\":\"testBucketPolicy\",
    \"Statement\": [{\"Sid\":\"st1\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"" +
    AccountId+ "\"},\"Action\":\"s3-outposts:*\",\"Resource\":\"" + bucketArn + "\"}]}";

    PutBucketPolicyRequest reqPutBucketPolicy = new PutBucketPolicyRequest()
        .withAccountId(AccountId)
        .withBucket(bucketArn)
        .withPolicy(policy);

    PutBucketPolicyResult respPutBucketPolicy =
    s3ControlClient.putBucketPolicy(reqPutBucketPolicy);
    System.out.printf("PutBucketPolicy Response: %s%n",
    respPutBucketPolicy.toString());
}
```

## Amazon S3 on Outposts バケットのバケットポリシーを表示する

バケットポリシーは、リソースベースの AWS Identity and Access Management (IAM) ポリシーを使用して、バケットとその中のオブジェクトへのアクセス許可を付与できます。バケット所有者のみが、ポリシーをバケットに関連付けることができます。バケットに添付された許可は、バケット所有者が所有するバケットのすべてのオブジェクトに適用されます。バケットポリシーのサイズは 20 KB に制限されています。詳細については、「[バケットポリシー](#)」を参照してください。

以下のトピックでは、AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、または AWS SDK for Java を使用して、Amazon S3 on Outposts バケットポリシーを表示する方法を示します。

### S3 コンソールの使用

バケットポリシーを作成または編集するには

1. AWS マネジメントコンソール にサインインし、Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。

2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. アクセス許可を編集する Outposts バケットを選択します。
4. [Permissions] タブを選択します。
5. [Outposts bucket policy] (Outposts バケットポリシー) セクションで、既存のバケットポリシーを確認できます。詳細については、「[S3 on Outposts で IAM を設定する](#)」を参照してください。

## AWS CLI の使用

次の AWS CLI の例では、Outposts バケットのポリシーを取得します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control get-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket
```

## AWS SDK for Java の使用

次の SDK for Java の例では、Outposts バケットのポリシーを取得します。

```
import com.amazonaws.services.s3control.model.*;

public void getBucketPolicy(String bucketArn) {

    GetBucketPolicyRequest reqGetBucketPolicy = new GetBucketPolicyRequest()
        .withAccountId(AccountId)
        .withBucket(bucketArn);

    GetBucketPolicyResult respGetBucketPolicy =
s3ControlClient.getBucketPolicy(reqGetBucketPolicy);
    System.out.printf("GetBucketPolicy Response: %s%n",
respGetBucketPolicy.toString());

}
```

## Amazon S3 on Outposts バケットのバケットポリシーを削除する

バケットポリシーは、リソースベースの AWS Identity and Access Management (IAM) ポリシーを使用して、バケットとその中のオブジェクトへのアクセス許可を付与できます。バケット所有者のみが、ポリシーをバケットに関連付けることができます。バケットに添付された許可は、バケット所有

者が所有するバケットのすべてのオブジェクトに適用されます。バケットポリシーのサイズは 20 KB に制限されています。詳細については、「[バケットポリシー](#)」を参照してください。

以下のトピックでは、AWS マネジメントコンソールまたは AWS Command Line Interface (AWS CLI) を使用して、Amazon S3 on Outposts バケットポリシーを表示する方法を示します。

## S3 コンソールの使用

バケットポリシーを削除するには

1. <https://console.aws.amazon.com/s3/> で Amazon S3 コンソールを開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. アクセス許可を編集する Outposts バケットを選択します。
4. [Permissions] タブを選択します。
5. [Outposts bucket policy] (Outposts バケットポリシー) セクションで、[Delete] (削除) を選択します。
6. 削除を確定します。

## AWS CLI の使用

次の例では、AWS CLI を使用して、S3 on Outposts バケット (s3-outposts:DeleteBucket) のバケットポリシーを削除します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control delete-bucket-policy --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket
```

## バケットポリシーの例

S3 on Outposts バケットポリシーを使用すると、S3 on Outposts バケット内のオブジェクトへのアクセスを保護して、適切な権限を持つユーザーだけがアクセスできるようにすることができます。適切な権限を持たない認証済みユーザーが S3 on Outposts リソースにアクセスできないようにすることもできます。

このセクションでは、S3 on Outposts バケットポリシーの一般的なユースケース例を紹介します。これらのポリシーをテストするには、*user input placeholders* をお客様の情報 (バケット名など) と置き換えます。

オブジェクトのセットに対するアクセス許可を付与または拒否するために、Amazon リソースネーム (ARN) やその他の値でワイルドカード文字 (\*) を使用できます。例えば、共通の[プレフィックス](#)で始まるか、.html などの特定の拡張子で終わるオブジェクトのグループへのアクセスをコントロールできます。

AWS Identity and Access Management (IAM) ポリシー言語については、「[S3 on Outposts で IAM を設定する](#)」を参照してください。

#### Note

Amazon S3 コンソールを使用して [s3outposts](#) のアクセス許可をテストするときには、コンソールに必要な `s3outposts:createendpoint` や `s3outposts:listendpoints` などのアクセス許可を追加で付与する必要があります。

バケットポリシーを作成するためのその他のリソース

- S3 on Outposts バケットポリシーの作成時に使用できる IAM ポリシーアクション、リソース、条件キーのリストについては、「[Amazon S3 on Outposts のアクション、リソース、条件キー](#)」を参照してください。
- S3 on Outposts ポリシーの作成に関するガイダンスについては、「[Amazon S3 on Outposts バケットのバケットポリシーを追加または編集する](#)」を参照してください。

トピック

- [特定の IP アドレスに基づく Amazon S3 on Outposts バケットへのアクセスの管理](#)

特定の IP アドレスに基づく Amazon S3 on Outposts バケットへのアクセスの管理

バケットポリシーは、リソースベースの AWS Identity and Access Management (IAM) ポリシーを使用して、バケットとその中のオブジェクトへのアクセス許可を付与できます。バケット所有者のみが、ポリシーをバケットに関連付けることができます。バケットに添付された許可は、バケット所有者が所有するバケットのすべてのオブジェクトに適用されます。バケットポリシーのサイズは 20 KB に制限されています。詳細については、「[バケットポリシー](#)」を参照してください。

## 特定の IP アドレスへのアクセスの制限

以下の例では、リクエストが指定した IP アドレス範囲から発信されたものでない限り、指定したバケット内のオブジェクトに対してすべてのユーザーが [S3 on Outposts のオペレーション](#) を実行できないようにします。

### Note

特定の IP アドレスへのアクセスを制限する場合は、S3 on Outposts バケットにアクセスできる VPC エンドポイント、VPC ソース IP アドレス、または外部 IP アドレスも必ず指定してください。指定しないと、適切な権限が設定されていない限り、すべてのユーザーが S3 on Outposts バケット内のオブジェクトに対して [s3outposts](#) オペレーションを実行することをポリシーで拒否されている場合、バケットにアクセスできなくなる可能性があります。

このポリシーの Condition ステートメントは、許可された IP バージョン 4 (IPv4) の IP アドレスの範囲として、**192.0.2.0/24** を識別します。

Condition ブロックでは、NotIpAddress 条件と aws:SourceIp 条件キー (AWS 全体をターゲットとする条件キー) を使用します。aws:SourceIp 条件キーは、パブリック IP アドレス範囲にのみ使用できます。これらの条件キーの詳細については、「[S3 on Outposts のアクション、リソース、条件キー](#)」を参照してください。aws:SourceIp IPv4 値は標準の CIDR 表記を使用します。詳細については、「IAM ユーザーガイド」の「[IAM JSON ポリシー要素のリファレンス](#)」を参照してください。

### Warning

この S3 on Outposts ポリシーを使用する前に、この例の **192.0.2.0/24** IP アドレス範囲をユースケースに適した値に置き換えてください。置き換えないと、バケットにアクセスできなくなります。

```
{
  "Version": "2012-10-17",
  "Id": "S3OutpostsPolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
```

```
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3-outposts:*",
    "Resource": [
      "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
accesspoint/EXAMPLE-ACCESS-POINT-NAME",
      "arn:aws:aws:s3-outposts:region:111122223333:outpost/OUTPOSTS-ID/
bucket/amzn-s3-demo-bucket"
    ],
    "Condition": {
      "NotIpAddress": {
        "aws:SourceIp": "192.0.2.0/24"
      }
    }
  }
]
```

## IPv4 アドレスと IPv6 アドレスの許可

IPv6 アドレスの使用を開始する場合は、既存の IPv4 アドレス範囲に IPv6 アドレス範囲を追加して組織のすべてのポリシーを更新することをお勧めします。こうすることで、IPv6 への移行後もポリシーが引き続き機能するようになります。

以下の S3 on Outposts バケットポリシーの例は、組織の有効な IP アドレスすべてを含めるために、IPv4 アドレス範囲と IPv6 アドレス範囲を混在させる方法を示しています。このポリシーの例では、サンプル IP アドレス ([192.0.2.1](#) および [2001:DB8:1234:5678::1](#)) へのアクセスを許可したり、アドレス [203.0.113.1](#) および [2001:DB8:1234:5678:ABCD::1](#) へのアクセスを拒否したりできます。

aws:SourceIp 条件キーは、パブリック IP アドレス範囲にのみ使用できます。aws:SourceIp の IPv6 の値は、標準の CIDR 形式で指定する必要があります。IPv6 では、0 の範囲を表すために :: の使用がサポートされています (例: [2001:DB8:1234:5678::/64](#))。詳細については、『IAM ユーザーガイド』の「[IP アドレス条件演算子](#)」を参照してください。

### Warning

この S3 on Outposts ポリシーを使用する前に、この例の IP アドレス範囲をユースケースに適した値に置き換えます。置き換えないと、バケットにアクセスできなくなる可能性があります。

## JSON

```
{
  "Id": "S3OutpostsPolicyId2",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowIPmix",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": [
        "s3-outposts:GetObject",
        "s3-outposts:PutObject",
        "s3-outposts:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/amzn-s3-demo-bucket",
        "arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/amzn-s3-demo-bucket/*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "192.0.2.0/24",
            "2001:DB8:1234:5678::/64"
          ]
        },
        "NotIpAddress": {
          "aws:SourceIp": [
            "203.0.113.0/24",
            "2001:DB8:1234:5678:ABCD::/80"
          ]
        }
      }
    }
  ]
}
```

# Amazon S3 on Outposts バケットの一覧表示

Amazon S3 on Outposts を使用すると、AWS Outposts で S3 バケットを作成し、ローカルデータアクセス、ローカルデータ処理、データレジデンシーを必要とするアプリケーション用に、オンプレミスのオブジェクトを簡単に保存および取得できます。S3 on Outposts は、新しいストレージクラス、S3 Outposts (OUTPOSTS) を提供し、これは Amazon S3 API を使用し、AWS Outposts 上の複数のデバイスやサーバー間でデータを永続的かつ冗長的に保存するように設計されています。仮想プライベートクラウド (VPC) を介したアクセスポイントとエンドポイント接続を使用して、Outposts バケットと通信します。Outposts バケットでは、Amazon S3 と同じ API と機能 (アクセスポリシー、暗号化、タグ付けなど) を使用できます。AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDK、または REST API を使用して S3 on Outposts を使用できます。詳細については、「[Amazon S3 on Outposts とは](#)」を参照してください。

S3 on Outposts 内のバケットの操作の詳細については、[S3 on Outposts バケットの操作](#) を参照してください。

次の例は、AWS マネジメントコンソール、AWS CLI、および AWS SDK for Java を使用して、S3 on Outposts バケットのリストを返す方法を示しています。

## S3 コンソールの使用

1. Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. [Outposts buckets] (Outposts バケット) で、S3 on Outposts バケットのリストを確認します。

## の使用AWS CLI

次の AWS CLI の例では、Outpost 内のバケットのリストを取得します。このコマンドを使用するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。このコマンドの詳細については、「AWS CLI リファレンス」の「[list-regional-buckets](#)」を参照してください。

```
aws s3control list-regional-buckets --account-id 123456789012 --outpost-id op-01ac5d28a6a232904
```

## AWS SDK for Java の使用

次の SDK for Java 例では、Outpost 内のバケットのリストを取得します。詳細については、「Amazon Simple Storage Service API リファレンス」の「[ListRegionalBuckets](#)」を参照してください。

```
import com.amazonaws.services.s3control.model.*;

public void listRegionalBuckets() {

    ListRegionalBucketsRequest reqListBuckets = new ListRegionalBucketsRequest()
        .withAccountId(AccountId)
        .withOutpostId(OutpostId);

    ListRegionalBucketsResult respListBuckets =
s3ControlClient.listRegionalBuckets(reqListBuckets);
    System.out.printf("ListRegionalBuckets Response: %s\n",
respListBuckets.toString());
}
```

## AWS CLI および SDK for Java を使用して、S3 on Outposts バケットを取得する

Amazon S3 on Outposts を使用すると、AWS Outposts で S3 バケットを作成し、ローカルデータアクセス、ローカルデータ処理、データレジデンシーを必要とするアプリケーション用に、オンプレミスのオブジェクトを簡単に保存および取得できます。S3 on Outposts は、新しいストレージクラス、S3 Outposts (OUTPOSTS) を提供し、これは Amazon S3 API を使用し、AWS Outposts 上の複数のデバイスやサーバー間でデータを永続的かつ冗長的に保存するように設計されています。仮想プライベートクラウド (VPC) を介したアクセスポイントとエンドポイント接続を使用して、Outposts バケットと通信します。Outposts バケットでは、Amazon S3 と同じ API と機能 (アクセスポリシー、暗号化、タグ付けなど) を使用できます。AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDK、または REST API を使用して S3 on Outposts を使用できます。詳細については、「[Amazon S3 on Outposts とは](#)」を参照してください。

以下の例は、AWS CLI と AWS SDK for Java を使用して、S3 on Outposts バケットを取得する方法を示しています。

**Note**

AWS CLI または AWS SDK を通じて Amazon S3 on Outposts を使用する場合、バケット名の代わりに Outposts のアクセスポイント ARN を提供します。アクセスポイント ARN は次の形式になります。*region* は Outpost が属するリージョンの AWS リージョン コードです。

```
arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

S3 on Outposts ARN の詳細については、「[S3 on Outposts のリソース ARN](#)」を参照してください。

## の使用AWS CLI

次の S3 on Outposts の例では、AWS CLI を使用してバケットを取得します。このコマンドを使用するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。詳細については、「AWS CLI リファレンス」の「[get-bucket](#)」を参照してください。

```
aws s3control get-bucket --account-id 123456789012 --bucket "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket"
```

## AWS SDK for Java の使用

次の S3 on Outposts の例では、SDK for Java を使用してバケットを取得します。詳細については、「Amazon Simple Storage Service API リファレンス」の「[GetBucket](#)」を参照してください。

```
import com.amazonaws.services.s3control.model.*;

public void getBucket(String bucketArn) {

    GetBucketRequest reqGetBucket = new GetBucketRequest()
        .withBucket(bucketArn)
        .withAccountId(AccountId);

    GetBucketResult respGetBucket = s3ControlClient.getBucket(reqGetBucket);
    System.out.printf("GetBucket Response: %s\n", respGetBucket.toString());

}
```

# Amazon S3 on Outposts バケットの削除

Amazon S3 on Outposts を使用すると、AWS Outposts で S3 バケットを作成し、ローカルデータアクセス、ローカルデータ処理、データレジデンシーを必要とするアプリケーション用に、オンプレミスのオブジェクトを簡単に保存および取得できます。S3 on Outposts は、新しいストレージクラス、S3 Outposts (OUTPOSTS) を提供し、これは Amazon S3 API を使用し、AWS Outposts 上の複数のデバイスやサーバー間でデータを永続的かつ冗長的に保存するように設計されています。仮想プライベートクラウド (VPC) を介したアクセスポイントとエンドポイント接続を使用して、Outposts バケットと通信します。Outposts バケットでは、Amazon S3 と同じ API と機能 (アクセスポリシー、暗号化、タグ付けなど) を使用できます。AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDK、または REST API を使用して S3 on Outposts を使用できます。詳細については、「[Amazon S3 on Outposts とは](#)」を参照してください。

S3 on Outposts 内のバケットの操作の詳細については、[S3 on Outposts バケットの操作](#) を参照してください。

バケットを作成する AWS アカウント がそのバケットを所有し、バケットを削除できる唯一のアカウントです。

## Note

- Outposts バケットは、削除する前に空にする必要があります。

Amazon S3 コンソールは、S3 on Outposts オブジェクトアクションをサポートしていません。S3 on Outposts バケット内のオブジェクトを削除するには、REST API、AWS CLI、または AWS SDK を使用する必要があります。

- Outposts バケットを削除する前に、バケットの Outposts アクセスポイントを削除する必要があります。詳細については、「[アクセスポイントの削除](#)」を参照してください。
- バケットが削除された後は、そのバケットを復元することはできません。

次の例は、AWS マネジメントコンソール と AWS Command Line Interface (AWS CLI) を使用して、S3 on Outposts バケットを削除する方法を示しています。

## S3 コンソールの使用

1. AWS マネジメントコンソール にサインインし、Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。

2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. 削除するバケットを選択して、[Delete] (削除) を選択します。
4. 削除を確定します。

## AWS CLI の使用

次の例では、AWS CLI を使用して S3 on Outposts バケット (s3-outposts>DeleteBucket) を削除します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control delete-bucket --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket
```

## Amazon S3 on Outposts アクセスポイントの操作

Amazon S3 on Outposts バケットにアクセスするには、アクセスポイントを作成して設定する必要があります。

アクセスポイントは、Amazon S3 の共有データセットへの大規模なデータアクセスの管理を簡素化します。アクセスポイントは、バケットにアタッチされた名前付きのネットワークエンドポイントで、Amazon S3 オブジェクトのオペレーション (GetObject や PutObject など) を実行するために使用できます。S3 on Outposts では、Outposts バケット内の任意のオブジェクトにアクセスするにはアクセスポイントを使用する必要があります。アクセスポイントでは、仮想ホスト形式のアドレス指定のみがサポートされます。

### Note

Outposts バケットを作成する AWS アカウントがそのバケットを所有し、そのバケットにアクセスポイントを割り当てることができる唯一のアカウントです。

以下のセクションでは、S3 on Outposts バケットを作成および管理する方法について説明します。

### トピック

- [S3 on Outposts アクセスポイントの作成](#)

- [S3 on Outposts アクセスポイントでのバケット形式のエイリアスの使用](#)
- [アクセスポイント設定に関する情報を表示する](#)
- [Amazon S3 on Outposts アクセスポイントのリストを表示する](#)
- [アクセスポイントの削除](#)
- [アクセスポイントポリシーを追加または編集するには](#)
- [S3 on Outposts アクセスポイントのアクセスポイントポリシーの表示](#)

## S3 on Outposts アクセスポイントの作成

Amazon S3 on Outposts バケットにアクセスするには、アクセスポイントを作成して設定する必要があります。

アクセスポイントは、Amazon S3 の共有データセットへの大規模なデータアクセスの管理を簡素化します。アクセスポイントは、バケットにアタッチされた名前付きのネットワークエンドポイントで、Amazon S3 オブジェクトのオペレーション (GetObject や PutObject など) を実行するために使用できます。S3 on Outposts では、Outposts バケット内の任意のオブジェクトにアクセスするにはアクセスポイントを使用する必要があります。アクセスポイントでは、仮想ホスト形式のアドレス指定のみがサポートされます。

以下の例は、AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、および AWS SDK for Java を使用して、S3 on Outposts アクセスポイントを作成する方法を示しています。

### Note

Outposts バケットを作成する AWS アカウント がそのバケットを所有し、そのバケットにアクセスポイントを割り当てることができる唯一のアカウントです。

### S3 コンソールの使用

1. Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. Outposts アクセスポイントを作成する Outposts バケットを選択します。
4. [Outposts アクセスポイント] タブを選択します。
5. [Outposts access points] (Outposts アクセスポイント) セクションで、[Create Outposts access point] (Outposts アクセスポイントの作成) を選択します。

- [Outposts access point settings] (Outposts アクセスポイントの設定) セクションで、アクセスポイントの名前を入力し、アクセスポイントの仮想プライベートクラウド (VPC) を選択します。
- アクセスポイントのポリシーを追加する場合は、[Outposts access point policy] (Outposts アクセスポイントポリシー) セクションに入力します。

詳細については、「[S3 on Outposts で IAM を設定する](#)」を参照してください。

## AWS CLI の使用

### Example

次の AWS CLI の例では、Outposts バケットのアクセスポイントを作成します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control create-access-point --account-id 123456789012
--name example-outposts-access-point --bucket "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket" --vpc-configuration VpcId=example-vpc-12345
```

## AWS SDK for Java の使用

### Example

AWS SDK for Java を使用して S3 Outposts バケットのアクセスポイントを作成する方法の例については、「AWS SDK for Java 2.x のコード例」の「[CreateOutpostsAccessPoint.java](#)」を参照してください。

## S3 on Outposts アクセスポイントでのバケット形式のエイリアスの使用

S3 on Outposts では、Outposts バケット内の任意のオブジェクトにアクセスするにはアクセスポイントを使用する必要があります。バケットのアクセスポイントを作成するたびに、S3 on Outposts によってアクセスポイントのエイリアスが自動的に生成されます。このアクセスポイントエイリアスは、あらゆるデータプレーンオペレーションにおいて、アクセスポイント ARN の代わりに使用できます。例えば、アクセスポイントエイリアスを使用して、PUT、GET、LIST などのオブジェクトレベルの操作を実行できます。これらのオペレーションのリストについては、「[オブジェクト管理のための Amazon S3 API オペレーション](#)」を参照してください。

以下は、*my-access-point* という名前のアクセスポイントの ARN とアクセスポイントのエイリアスの例です。

- アクセスポイント ARN – `arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/my-access-point`
- アクセスポイントエイリアス – `my-access-po-o01ac5d28a6a232904e8xz5w8ijx1qzlb3i3kuse10--op-s3`

ARN の詳細については、「AWS 全般のリファレンス」の「[Amazon リソースネーム \(ARN\)](#)」を参照してください。

アクセスポイントエイリア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

- [アクセスポイントエイリアス](#)
- [S3 on Outposts オブジェクトオペレーションでのアクセスポイントエイリアスの使用](#)
- [制限事項](#)

## アクセスポイントエイリアス

アクセスポイントエイリアスは、S3 on Outposts バケットと同じ名前空間内に作成されます。アクセスポイントを作成すると、S3 on Outposts はアクセスポイントエイリアスを自動的に生成し、変更することはできません。アクセスポイントエイリアスは、有効な S3 on Outposts バケットの名前のすべての要件を満たしており、次の部分で構成されています。

`access point name prefix-metadata--op-s3`

### Note

--op-s3 サフィックスは、アクセスポイントエイリアス用に予約されているため、バケット名やアクセスポイント名には使用しないことをお勧めします。S3 on Outposts バケット命名規則の詳細については、「[S3 on Outposts バケットの操作](#)」を参照してください。

## アクセスポイントエイリアスの検索

以下の例は、Amazon S3 コンソールと AWS CLI を使用して、アクセスポイントエイリアスを検索する方法を示しています。

## Example: Amazon S3 コンソールでアクセスポイントエイリアスを検索してコピーする

コンソールでアクセスポイントを作成すると、[Access Points] (アクセスポイント) リストの [Access Point alias] (アクセスポイントエイリアス) 列からアクセスポイントエイリアスを取得できます。

アクセスポイントエイリアスをコピーするには

1. Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts access points] (Outposts アクセスポイント) を選択します。
3. アクセスポイントエイリアスをコピーするには、次のいずれかの操作を行います。
  - [Access Points] (アクセスポイント) リストで、アクセスポイント名の横にあるオプションボタンを選択し、次に [Copy Access Point alias] (アクセスポイントエイリアスのコピー) を選択します。
  - アクセスポイント名を選択します。次に、[Outposts access point overview] (Outposts アクセスポイントの概要) で、アクセスポイントエイリアスをコピーします。

## Example: AWS CLI を使用してアクセスポイントを作成し、レスポンスでのアクセスポイントエイリアスを検索する

次の `create-access-point` コマンドにおける AWS CLI の例は、アクセスポイントを作成し、自動的に生成されたアクセスポイントエイリアスを返します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control create-access-point --bucket example-outposts-bucket --name example-outposts-access-point --account-id 123456789012
```

```
{
  "AccessPointArn":
    "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/
    accesspoint/example-outposts-access-point",
  "Alias": "example-outp-o01ac5d28a6a232904e8xz5w8ijx1qzlbp3i3kuse10--op-s3"
}
```

## Example: AWS CLI を使用してアクセスポイントのエイリアスを取得する

次の `get-access-point` コマンドにおける AWS CLI の例は、指定されたアクセスポイントに関する情報を返します。この情報には、アクセスポイントエイリアスが含まれます。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control get-access-point --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket --name example-outposts-access-point --account-id 123456789012

{
  "Name": "example-outposts-access-point",
  "Bucket": "example-outposts-bucket",
  "NetworkOrigin": "Vpc",
  "VpcConfiguration": {
    "VpcId": "vpc-01234567890abcdef"
  },
  "PublicAccessBlockConfiguration": {
    "BlockPublicAcls": true,
    "IgnorePublicAcls": true,
    "BlockPublicPolicy": true,
    "RestrictPublicBuckets": true
  },
  "CreationDate": "2022-09-18T17:49:15.584000+00:00",
  "Alias": "example-outp-o0b1d075431d83bebde8xz5w8ijx1qzlbp3i3kuse10--op-s3"
}
```

## Example: アクセスポイントを一覧表示して、AWS CLI を使用してアクセスポイントエイリアスを検索する

次の `list-access-points` コマンドにおける AWS CLI の例は、指定されたアクセスポイントに関する情報を取得します。この情報には、アクセスポイントエイリアスが含まれます。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control list-access-points --account-id 123456789012 --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-
bucket

{
  "AccessPointList": [
    {
      "Name": "example-outposts-access-point",
```

```
    "NetworkOrigin": "Vpc",
    "VpcConfiguration": {
      "VpcId": "vpc-01234567890abcdef"
    },
    "Bucket": "example-outposts-bucket",
    "AccessPointArn": "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point",
    "Alias": "example-outp-o0b1d075431d83bebde8xz5w8ijx1qz1bp3i3kuse10--op-s3"
  }
]
}
```

## S3 on Outposts オブジェクトオペレーションでのアクセスポイントエイリアスの使用

アクセスポイントを採用する場合、大幅なコード変更を必要とせずに、アクセスポイントエイリアスを使用できます。

この AWS CLI の例は、S3 on Outposts バケットの `get-object` オペレーションを示しています。この例では、フルアクセスポイント ARN の代わりに、アクセスポイントエイリアスを `--bucket` の値として使用します。

```
aws s3api get-object --bucket my-access-po-
o0b1d075431d83bebde8xz5w8ijx1qz1bp3i3kuse10--op-s3 --key testkey sample-object.rtf

{
  "AcceptRanges": "bytes",
  "LastModified": "2020-01-08T22:16:28+00:00",
  "ContentLength": 910,
  "ETag": "\"00751974dc146b76404bb7290f8f51bb\"",
  "VersionId": "null",
  "ContentType": "text/rtf",
  "Metadata": {}
}
```

## 制限事項

- お客様はエイリアスを設定できません。
- アクセスポイントでは、エイリアスの削除、変更、無効化はできません。

- アクセスポイントエイリアスは S3 on Outposts コントロールプレーンオペレーションに使用することはできません。S3 on Outposts コントロールプレーンオペレーションのリストについては、「[バケット管理のための Amazon S3 コントロール API オペレーション](#)」を参照してください。
- エイリアスは AWS Identity and Access Management (IAM ポリシー) では使用できません。

## アクセスポイント設定に関する情報を表示する

アクセスポイントは、Amazon S3 の共有データセットへの大規模なデータアクセスの管理を簡素化します。アクセスポイントは、バケットにアタッチされた名前付きのネットワークエンドポイントで、Amazon S3 オブジェクトのオペレーション (GetObject や PutObject など) を実行するために使用できます。S3 on Outposts では、Outposts バケット内の任意のオブジェクトにアクセスするにはアクセスポイントを使用する必要があります。アクセスポイントでは、仮想ホスト形式のアドレス指定のみがサポートされます。

以下のトピックでは、AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、および AWS SDK for Java を使用して S3 on Outposts アクセスポイントの設定情報を返す方法について説明します。

### S3 コンソールの使用

1. Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts access points] (Outposts アクセスポイント) を選択します。
3. 設定の詳細を表示する Outposts アクセスポイントを選択します。
4. [Outposts access point overview] (Outposts アクセスポイントの概要) で、アクセスポイント設定の詳細を確認します。

### の使用AWS CLI

次の AWS CLI の例では、Outposts バケットのアクセスポイントを取得します。*user input placeholders* を、ユーザー自身の情報に置き換えます。

```
aws s3control get-access-point --account-id 123456789012 --name arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

## AWS SDK for Java の使用

次の SDK for Java の例では、Outposts バケットのアクセスポイントを取得します。

```
import com.amazonaws.services.s3control.model.*;

public void getAccessPoint(String accessPointArn) {

    GetAccessPointRequest reqGetAP = new GetAccessPointRequest()
        .withAccountId(AccountId)
        .withName(accessPointArn);

    GetAccessPointResult respGetAP = s3ControlClient.getAccessPoint(reqGetAP);
    System.out.printf("GetAccessPoint Response: %s%n", respGetAP.toString());

}
```

## Amazon S3 on Outposts アクセスポイントのリストを表示する

アクセスポイントは、Amazon S3 の共有データセットへの大規模なデータアクセスの管理を簡素化します。アクセスポイントは、バケットにアタッチされた名前付きのネットワークエンドポイントで、Amazon S3 オブジェクトのオペレーション (GetObject や PutObject など) を実行するために使用できます。S3 on Outposts では、Outposts バケット内の任意のオブジェクトにアクセスするにはアクセスポイントを使用する必要があります。アクセスポイントでは、仮想ホスト形式のアドレス指定のみがサポートされます。

以下のトピックでは、AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、および AWS SDK for Java を使用して S3 on Outposts アクセスポイントのリストを返す方法を示します。

### S3 コンソールの使用

1. Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts access points] (Outposts アクセスポイント) を選択します。
3. [Outposts access points] (Outposts アクセスポイント) で、S3 on Outposts アクセスポイントのリストを確認します。

## の使用AWS CLI

次の AWS CLI の例では、Outposts バケットのアクセスポイントを一覧表示します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control list-access-points --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket
```

## AWS SDK for Java の使用

次の SDK for Java の例では、Outposts バケットのアクセスポイントを一覧表示します。

```
import com.amazonaws.services.s3control.model.*;

public void listAccessPoints(String bucketArn) {

    ListAccessPointsRequest reqListAPs = new ListAccessPointsRequest()
        .withAccountId(AccountId)
        .withBucket(bucketArn);

    ListAccessPointsResult respListAPs = s3ControlClient.listAccessPoints(reqListAPs);
    System.out.printf("ListAccessPoints Response: %s\n", respListAPs.toString());

}
```

## アクセスポイントの削除

アクセスポイントは、Amazon S3 の共有データセットへの大規模なデータアクセスの管理を簡素化します。アクセスポイントは、バケットにアタッチされた名前付きのネットワークエンドポイントで、Amazon S3 オブジェクトのオペレーション (GetObject や PutObject など) を実行するために使用できます。S3 on Outposts では、Outposts バケット内の任意のオブジェクトにアクセスするにはアクセスポイントを使用する必要があります。アクセスポイントでは、仮想ホスト形式のアドレス指定のみがサポートされます。

以下の例は、AWS マネジメントコンソール と AWS Command Line Interface (AWS CLI) を使用して、アクセスポイントを削除する方法を示しています。

### S3 コンソールの使用

1. <https://console.aws.amazon.com/s3/> で Amazon S3 コンソールを開きます。

2. 左のナビゲーションペインで、[Outposts access points] (Outposts アクセスポイント) を選択します。
3. [Outposts access points] (Outposts アクセスポイント) セクションで、削除する Outposts アクセスポイントを選択します。
4. [削除] を選択します。
5. 削除を確定します。

## AWS CLI の使用

次の AWS CLI の例では、Outposts アクセスポイントを削除します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control delete-access-point --account-id 123456789012 --name arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

## アクセスポイントポリシーを追加または編集するには

アクセスポイントには、そのアクセスポイントを介して行われるすべてのリクエストに Amazon S3 on Outposts が適用する個別の許可とネットワークコントロールがあります。各アクセスポイントは、基になるバケットにアタッチされたバケットポリシーと連動して機能するカスタマイズされたアクセスポイントポリシーを適用します。詳細については、「[アクセスポイント](#)」を参照してください。

次のトピックでは、AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、および AWS SDK for Java を使用して、S3 on Outposts アクセスポイントのアクセスポイントポリシーを追加または編集する方法について説明します。

### S3 コンソールの使用

1. Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. アクセスポイントポリシーを編集する Outposts バケットを選択します。
4. [Outposts アクセスポイント] タブを選択します。
5. [Outposts access points] (Outposts アクセスポイント) セクションで、ポリシーを編集するアクセスポイントを選択し、[Edit policy] (ポリシーの編集) を選択します。

6. [Outposts access point policy] (Outposts アクセスポイントポリシー) セクションで、ポリシーを追加または編集します。詳細については、「[S3 on Outposts で IAM を設定する](#)」を参照してください。

## AWS CLI の使用

次の AWS CLI の例では、Outposts アクセスポイントにポリシーを配置します。

1. 次のアクセスポイントポリシーを JSON ファイルに保存します。この例では、ファイル名は `appolicy1.json` です。*user input placeholders* を、ユーザー自身の情報に置き換えます。

```
{
  "Version": "2012-10-17",
  "Id": "exampleAccessPointPolicy",
  "Statement": [
    {
      "Sid": "st1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "123456789012"
      },
      "Action": "s3-outposts:*",
      "Resource": "arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point"
    }
  ]
}
```

2. `put-access-point-policy` CLI コマンドの一部として JSON ファイルを送信します。*user input placeholders* を、ユーザー自身の情報に置き換えます。

```
aws s3control put-access-point-policy --account-id 123456789012 --name arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point --policy file://appolicy1.json
```

## AWS SDK for Java の使用

次の SDK for Java の例では、Outposts アクセスポイントにポリシーを配置します。

```
import com.amazonaws.services.s3control.model.*;

public void putAccessPointPolicy(String accessPointArn) {

    String policy = "{\"Version\":\"2012-10-17\",\"Id\":\"testAccessPointPolicy\",
    \"Statement\": [{\"Sid\":\"st1\",\"Effect\":\"Allow\",\"Principal\":{\"AWS\":\"" +
    AccountId + "\"},\"Action\":\"s3-outposts:*\",\"Resource\":\"" + accessPointArn +
    "\"}]}";

    PutAccessPointPolicyRequest reqPutAccessPointPolicy = new
    PutAccessPointPolicyRequest()
        .withAccountId(AccountId)
        .withName(accessPointArn)
        .withPolicy(policy);

    PutAccessPointPolicyResult respPutAccessPointPolicy =
    s3ControlClient.putAccessPointPolicy(reqPutAccessPointPolicy);
    System.out.printf("PutAccessPointPolicy Response: %s\n",
    respPutAccessPointPolicy.toString());
    printWriter.printf("PutAccessPointPolicy Response: %s\n",
    respPutAccessPointPolicy.toString());

}
```

## S3 on Outposts アクセスポイントのアクセスポイントポリシーの表示

アクセスポイントには、そのアクセスポイントを介して行われるすべてのリクエストに Amazon S3 on Outposts が適用する個別の許可とネットワークコントロールがあります。各アクセスポイントは、基になるバケットにアタッチされたバケットポリシーと連動して機能するカスタマイズされたアクセスポイントポリシーを適用します。詳細については、「[アクセスポイント](#)」を参照してください。

S3 on Outposts でのアクセスポイントの操作の詳細については、「[S3 on Outposts バケットの操作](#)」を参照してください。

以下のトピックでは、AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、および AWS SDK for Java を使用して、S3 on Outposts アクセスポイントポリシーを表示する方法について説明します。

### S3 コンソールの使用

1. <https://console.aws.amazon.com/s3/> で Amazon S3 コンソールを開きます。

2. 左のナビゲーションペインで、[Outposts access points] (Outposts アクセスポイント) を選択します。
3. ポリシーを表示する Outposts アクセスポイントを選択します。
4. [Permissions] (許可) タブで、S3 on Outposts アクセスポイントポリシーを確認します。
5. アクセスポイントポリシーを編集するには、[「アクセスポイントポリシーを追加または編集するには」](#)を参照してください。

## AWS CLI の使用

次の AWS CLI の例では、Outposts アクセスポイントのポリシーを取得します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control get-access-point-policy --account-id 123456789012 --name arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

## AWS SDK for Java の使用

次の SDK for Java の例では、Outposts アクセスポイントのポリシーを取得します。

```
import com.amazonaws.services.s3control.model.*;

public void getAccessPointPolicy(String accessPointArn) {

    GetAccessPointPolicyRequest reqGetAccessPointPolicy = new
    GetAccessPointPolicyRequest()
        .withAccountId(AccountId)
        .withName(accessPointArn);

    GetAccessPointPolicyResult respGetAccessPointPolicy =
    s3ControlClient.getAccessPointPolicy(reqGetAccessPointPolicy);
    System.out.printf("GetAccessPointPolicy Response: %s%n",
    respGetAccessPointPolicy.toString());
    printWriter.printf("GetAccessPointPolicy Response: %s%n",
    respGetAccessPointPolicy.toString());

}
```

## Amazon S3 on Outposts エンドポイントの使用

Amazon S3 on Outposts のアクセスポイントにリクエストをルーティングするには、S3 on Outposts エンドポイントを作成し設定する必要があります。エンドポイントを作成するには、Outposts のホームリージョンへのサービスリンクとのアクティブな接続が必要です。Outpost 上の各仮想プライベートクラウド (VPC) に 1 つのエンドポイントに関連付けることができます。エンドポイントクォータの詳細については、[S3 on Outposts のネットワーク要件](#) を参照してください。Outposts バケットにアクセスしてオブジェクトオペレーションを実行できるようにするには、エンドポイントを作成する必要があります。詳細については、「[エンドポイント](#)」を参照してください。

エンドポイントを作成したら、[ステータス] フィールドを使用してエンドポイントの状態を把握できます。Outposts がオフラインの場合は、CREATE\_FAILED が返されます。サービスリンク接続を確認し、エンドポイントを削除して、接続が再開された後に作成オペレーションを再試行できます。その他のエラーコードのリストについては、以下を参照してください。詳細については、「[エンドポイント](#)」を参照してください。

API	ステータス	失敗理由エラーコード	メッセージ - 失敗理由
CreateEndpoint	Create_Failed	OutpostNotReachable	Outposts ホームリージョンへのサービスリンク接続が停止しているため、エンドポイントを作成できませんでした。接続の詳細を確認し、エンドポイントを削除して、もう一度お試しください。
CreateEndpoint	Create_Failed	InternalError	内部エラーのため、エンドポイントを作成できませんでした。エンドポイントを削除して、もう一度作成してください。
DeleteEndpoint	Delete_Failed	OutpostNotReachable	Outposts ホームリージョンへのサービスリンク接続が停止しているため、エンドポイントを削除できませんでした。接続を確認して、もう一度お試しください。

API	ステータス	失敗理由エラーコード	メッセージ - 失敗理由
DeleteEndpoint	Delete_Failed	InternalError	内部エラーのため、エンドポイントを削除できませんでした。もう一度試してください。

S3 on Outposts のバケットの操作の詳細については、[S3 on Outposts バケットの操作](#) を参照してください。

以下のセクションでは、S3 on Outposts のエンドポイントを作成して管理する方法を説明します。

## トピック

- [Outpost でエンドポイントを作成する](#)
- [Amazon S3 on Outposts エンドポイントのリストを表示する](#)
- [Amazon S3 on Outposts エンドポイントの削除](#)

## Outpost でエンドポイントを作成する

Amazon S3 on Outposts のアクセスポイントにリクエストをルーティングするには、S3 on Outposts エンドポイントを作成し設定する必要があります。エンドポイントを作成するには、Outposts のホームリージョンへのサービスリンクとのアクティブな接続が必要です。Outpost 上の各仮想プライベートクラウド (VPC) に 1 つのエンドポイントを関連付けることができます。エンドポイントクォータの詳細については、[S3 on Outposts のネットワーク要件](#) を参照してください。Outposts バケットにアクセスしてオブジェクトオペレーションを実行できるようにするには、エンドポイントを作成する必要があります。詳細については、「[エンドポイント](#)」を参照してください。

### アクセス許可

エンドポイントの作成に必要な許可の詳細については、「[S3 on Outposts エンドポイントの許可](#)」を参照してください。

また、エンドポイントの作成時に、S3 on Outposts は AWS アカウントでサービスにリンクされたロールを作成します。詳細については、「[Amazon S3 on Outposts でのサービスにリンクされたロールの使用](#)」を参照してください。

以下の例は、AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI) および AWS SDK for Java を使用して、S3 on Outposts エンドポイントを作成する方法を示しています。

## S3 コンソールの使用

1. AWS マネジメントコンソール にサインインし、Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts access points] (Outposts アクセスポイント) を選択します。
3. [Outposts endpoints] (Outposts エンドポイント) タブを選択します。
4. [Create Outposts endpoint] (Outposts エンドポイントの作成) を選択します。
5. [Outpost] で、このエンドポイントを作成する Outpost を選択します。
6. [VPC] で、まだエンドポイントがなく、Outposts エンドポイントのルールにも準拠している VPC を選択します。

仮想プライベートクラウド (VPC) を使用すると、定義した仮想ネットワークに AWS リソースを起動できます。仮想ネットワークは、お客様自身のデータセンターで運用されていた従来のネットワークによく似ていますが、のスケラブルなインフラストラクチャを使用できるというメリットがありますAWS

VPC がない場合は、[Create VPC] (VPC を作成) を選択します。詳細については、「Amazon S3 ユーザーガイド」の「[仮想プライベートクラウド \(VPC\) に制限されたアクセスポイントの作成](#)」を参照してください。

7. [Create Outposts endpoint] (Outposts エンドポイントの作成) を選択します。

## AWS CLI を使用する場合

### Example

次の AWS CLI の例では、VPC リソースアクセスタイプを使用して、Outpost のエンドポイントを作成します。VPC はサブネットから派生します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id  
subnet-8c7a57c5 --security-group-id sg-ab19e0d1
```

次の AWS CLI の例では、アクセスタイプにユーザー所有の IP アドレスプール (CoIP プール) を使用して、Outpost のエンドポイントを作成します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3outposts create-endpoint --outpost-id op-01ac5d28a6a232904 --subnet-id
  subnet-8c7a57c5 --security-group-id sg-ab19e0d1 --access-type CustomerOwnedIp --
  customer-owned-ipv4-pool ipv4pool-coip-12345678901234567
```

## AWS SDK for Java の使用

### Example

AWS SDK for Java を使用して S3 Outpost のエンドポイントを作成する方法の例については、「AWS SDK for Java 2.x のコード例」の「[CreateOutpostsEndPoint.java](#)」を参照してください。

## Amazon S3 on Outposts エンドポイントのリストを表示する

Amazon S3 on Outposts のアクセスポイントにリクエストをルーティングするには、S3 on Outposts エンドポイントを作成し設定する必要があります。エンドポイントを作成するには、Outposts のホームリージョンへのサービスリンクとのアクティブな接続が必要です。Outpost 上の各仮想プライベートクラウド (VPC) に 1 つのエンドポイントに関連付けることができます。エンドポイントクォータの詳細については、[S3 on Outposts のネットワーク要件](#) を参照してください。Outposts バケットにアクセスしてオブジェクトオペレーションを実行できるようにするには、エンドポイントを作成する必要があります。詳細については、「[エンドポイント](#)」を参照してください。

次の例は、AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、および AWS SDK for Java を使用して、S3 on Outposts エンドポイントのリストを返す方法を示しています。

### S3 コンソールの使用

1. Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts access points] (Outposts アクセスポイント) を選択します。
3. [Outposts access points](Outposts アクセスポイント) ページで、[Outposts endpoints] (Outposts エンドポイント) タブを選択します。
4. [Outposts endpoints] (Outposts エンドポイント) で、S3 on Outposts エンドポイントのリストを表示できます。

### の使用AWS CLI

次の AWS CLI の例では、お客様のアカウントに関連付けられた AWS Outposts のリソースのエンドポイントを一覧表示します。このコマンドの詳細については、「AWS CLI リファレンス」の「[list-endpoints](#)」を参照してください。

```
aws s3outposts list-endpoints
```

## AWS SDK for Java の使用

次のSDK for Java の例では、Outposts のエンドポイントを一覧表示します。詳細については、「Amazon Simple Storage Service API リファレンス」の「[ListEndpoints](#)」を参照してください。

```
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.ListEndpointsRequest;
import com.amazonaws.services.s3outposts.model.ListEndpointsResult;

public void listEndpoints() {
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
        .standard().build();

    ListEndpointsRequest listEndpointsRequest = new ListEndpointsRequest();
    ListEndpointsResult listEndpointsResult =
s3OutpostsClient.listEndpoints(listEndpointsRequest);
    System.out.println("List endpoints result is " + listEndpointsResult);
}
```

## Amazon S3 on Outposts エンドポイントの削除

Amazon S3 on Outposts のアクセスポイントにリクエストをルーティングするには、S3 on Outposts エンドポイントを作成し設定する必要があります。エンドポイントを作成するには、Outposts のホームリージョンへのサービスリンクとのアクティブな接続が必要です。Outpost 上の各仮想プライベートクラウド (VPC) に 1 つのエンドポイントに関連付けることができます。エンドポイントクォータの詳細については、[S3 on Outposts のネットワーク要件](#) を参照してください。Outposts バケットにアクセスしてオブジェクトオペレーションを実行できるようにするには、エンドポイントを作成する必要があります。詳細については、「[エンドポイント](#)」を参照してください。

次の例は、AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、および AWS SDK for Java を使用して、S3 on Outposts エンドポイントを削除する方法を示しています。

### S3 コンソールの使用

1. Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts access points] (Outposts アクセスポイント) を選択します。

3. [Outposts access points](Outposts アクセスポイント) ページで、[Outposts endpoints] (Outposts エンドポイント) タブを選択します。
4. [Outposts endpoints] (Outposts エンドポイント) で、削除するエンドポイントを選択し、[Delete] (削除) を選択します。

## の使用AWS CLI

次の AWS CLI の例では、Outposts のエンドポイントを削除します。このコマンドを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3outposts delete-endpoint --endpoint-id example-endpoint-id --outpost-id op-01ac5d28a6a232904
```

## AWS SDK for Java の使用

次の SDK for Java の例では、Outposts のエンドポイントを削除します。この例を実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
import com.amazonaws.arn.Arn;
import com.amazonaws.services.s3outposts.AmazonS3Outposts;
import com.amazonaws.services.s3outposts.AmazonS3OutpostsClientBuilder;
import com.amazonaws.services.s3outposts.model.DeleteEndpointRequest;

public void deleteEndpoint(String endpointArnInput) {
    String outpostId = "op-01ac5d28a6a232904";
    AmazonS3Outposts s3OutpostsClient = AmazonS3OutpostsClientBuilder
        .standard().build();

    Arn endpointArn = Arn.fromString(endpointArnInput);
    String[] resourceParts = endpointArn.getResource().getResource().split("/");
    String endpointId = resourceParts[resourceParts.length - 1];
    DeleteEndpointRequest deleteEndpointRequest = new DeleteEndpointRequest()
        .withEndpointId(endpointId)
        .withOutpostId(outpostId);
    s3OutpostsClient.deleteEndpoint(deleteEndpointRequest);
    System.out.println("Endpoint with id " + endpointId + " is deleted.");
}
```

## S3 on Outposts オブジェクトの操作

Amazon S3 on Outposts を使用すると、AWS Outposts で S3 バケットを作成し、ローカルデータアクセス、ローカルデータ処理、データレジデンシーを必要とするアプリケーション用に、オンプレミスのオブジェクトを簡単に保存および取得できます。S3 on Outposts は、新しいストレージクラス、S3 Outposts (OUTPOSTS) を提供し、これは Amazon S3 API を使用し、AWS Outposts 上の複数のデバイスやサーバー間でデータを永続的かつ冗長的に保存するように設計されています。仮想プライベートクラウド (VPC) を介したアクセスポイントとエンドポイント接続を使用して、Outposts バケットと通信します。Outposts バケットでは、Amazon S3 と同じ API と機能 (アクセスポリシー、暗号化、タグ付けなど) を使用できます。AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDK、または REST API を使用して S3 on Outposts を使用できます。

オブジェクトは、Amazon S3 on Outposts に保存される基本エンティティです。すべてのオブジェクトはバケット内に保存されます。Outpost バケット内の任意のオブジェクトにアクセスするには、アクセスポイントを使用する必要があります。オブジェクト操作のためにバケットを指定するときには、アクセスポイントの Amazon リソースネーム (ARN) またはアクセスポイントエイリアスを使用します。アクセスポイントエイリアスの詳細については、「[S3 on Outposts アクセスポイントでのバケット形式のエイリアスの使用](#)」を参照してください。

次の例は、S3 on Outposts アクセスポイントの ARN 形式を示し、これは、Outpost が属するリージョンの AWS リージョン コード、AWS アカウント ID、Outpost ID、アクセスポイント名を含みます。

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

S3 on Outposts ARN の詳細については、「[S3 on Outposts のリソース ARN](#)」を参照してください。

オブジェクト ARN は次の形式を使用し、Outpost が属する AWS リージョン、AWS アカウント ID、Outpost ID、バケット名、およびオブジェクトキーが含まれます。

```
arn:aws:s3-outposts:us-west-2:123456789012:outpost/op-01ac5d28a6a232904/bucket/amzn-s3-demo-bucket1/object/myobject
```

Amazon S3 on Outposts では、オブジェクトデータは常に Outpost に保存されます。AWS が Outpost ラックを設置すると、データの常駐要件を満たすために、データは Outpost にローカルに保たれます。オブジェクトが Outpost を離れたり、AWS リージョン 外に出たりすることはありま

せん。AWS マネジメントコンソールはリージョン内でホストされるため、コンソールを使用して Outpost にオブジェクトをアップロードしたり、管理したりすることはできません。ただし REST API、AWS Command Line Interface (AWS CLI)、および AWS SDK を使用して、アクセスポイントを介してオブジェクトのアップロードと管理を行うことができます。

## トピック

- [S3 on Outposts バケットにオブジェクトをアップロードする](#)
- [AWS SDK for Java を使用して Amazon S3 on Outposts バケットにオブジェクトをコピーする](#)
- [Amazon S3 on Outposts バケットからオブジェクトを取得する](#)
- [Amazon S3 on Outposts バケット内のオブジェクトを一覧表示する](#)
- [Amazon S3 on Outposts バケット内のオブジェクトを削除する](#)
- [HeadBucket を使用して、S3 on Outposts バケットが存在し、アクセス許可があるかどうかを調べる](#)
- [SDK for Java でのマルチパートアップロードの実行と管理](#)
- [S3 on Outposts での署名付き URL の使用](#)
- [ローカルの Amazon EMR on Outposts を使用した Amazon S3 on Outposts](#)
- [認可と認証キャッシング](#)

## S3 on Outposts バケットにオブジェクトをアップロードする

オブジェクトは、Amazon S3 on Outposts に保存される基本エンティティです。すべてのオブジェクトはバケット内に保存されます。Outpost バケット内の任意のオブジェクトにアクセスするには、アクセスポイントを使用する必要があります。オブジェクト操作のためにバケットを指定するときには、アクセスポイントの Amazon リソースネーム (ARN) またはアクセスポイントエイリアスを使用します。アクセスポイントエイリアスの詳細については、「[S3 on Outposts アクセスポイントでのバケット形式のエイリアスの使用](#)」を参照してください。

次の例は、S3 on Outposts アクセスポイントの ARN 形式を示し、これは、Outpost が属するリージョンの AWS リージョン コード、AWS アカウント ID、Outpost ID、アクセスポイント名を含みます。

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

S3 on Outposts ARN の詳細については、「[S3 on Outposts のリソース ARN](#)」を参照してください。

Amazon S3 on Outposts では、オブジェクトデータは常に Outpost に保存されます。AWS が Outpost ラックを設置すると、データの常駐要件を満たすために、データは Outpost にローカルに保たれます。オブジェクトが Outpost を離れたり、AWS リージョン 外に出たりすることはありません。AWS マネジメントコンソールはリージョン内でホストされるため、コンソールを使用して Outpost にオブジェクトをアップロードしたり、管理したりすることはできません。ただし REST API、AWS Command Line Interface (AWS CLI)、および AWS SDK を使用して、アクセスポイントを介してオブジェクトのアップロードと管理を行うことができます。

以下の AWS CLI および AWS SDK for Java の例は、アクセスポイントを使用して S3 on Outposts バケットにオブジェクトをアップロードする方法を示しています。

## AWS CLI

### Example

次の例では、S3 on Outposts バケット (s3-outposts:PutObject) に、AWS CLI を使って sample-object.xml という名前のオブジェクトを配置します。このコマンドを使用するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。このコマンドの詳細については、「AWS CLI リファレンス」の「[put-object](#)」を参照してください。

```
aws s3api put-object --bucket arn:aws:s3-outposts:Region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point --key sample-object.xml --body sample-object.xml
```

## SDK for Java

### Example

AWS SDK for Java を使用して S3 Outposts バケットにオブジェクトをアップロードする方法の例については、「AWS SDK for Java 2.x のコード例」の「[PutObjectOnOutpost.java](#)」を参照してください。

## AWS SDK for Java を使用して Amazon S3 on Outposts バケットにオブジェクトをコピーする

オブジェクトは、Amazon S3 on Outposts に保存される基本エンティティです。すべてのオブジェクトはバケット内に保存されます。Outpost バケット内の任意のオブジェクトにアクセスするには、アクセスポイントを使用する必要があります。オブジェクト操作のためにバケットを指定するときには、アクセスポイントの Amazon リソースネーム (ARN) またはアクセスポイントエイリアスを使用

します。アクセスポイントエイリアスの詳細については、「[S3 on Outposts アクセスポイントでのバケット形式のエイリアスの使用](#)」を参照してください。

次の例は、S3 on Outposts アクセスポイントの ARN 形式を示し、これは、Outpost が属するリージョンの AWS リージョン コード、AWS アカウント ID、Outpost ID、アクセスポイント名を含みます。

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

S3 on Outposts ARN の詳細については、「[S3 on Outposts のリソース ARN](#)」を参照してください。

Amazon S3 on Outposts では、オブジェクトデータは常に Outpost に保存されます。AWS が Outpost ラックを設置すると、データの常駐要件を満たすために、データは Outpost にローカルに保たれます。オブジェクトが Outpost を離れたり、AWS リージョン 外に出たりすることはありません。AWS マネジメントコンソールはリージョン内でホストされるため、コンソールを使用して Outpost にオブジェクトをアップロードしたり、管理したりすることはできません。ただし REST API、AWS Command Line Interface (AWS CLI)、および AWS SDK を使用して、アクセスポイントを介してオブジェクトのアップロードと管理を行うことができます。

次の例は、AWS SDK for Java を使用して、S3 on Outposts バケット内のオブジェクトをコピーする方法を示しています。

## AWS SDK for Java の使用

次の S3 on Outposts の例では、SDK for Java を使用してオブジェクトを同じバケットの新しいオブジェクトにコピーします。この例を実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.CopyObjectRequest;

public class CopyObject {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String sourceKey = "*** Source object key ***";
        String destinationKey = "*** Destination object key ***";
```

```
try {
    // This code expects that you have AWS credentials set up per:
    // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
    AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
        .enableUseArnRegion()
        .build();

    // Copy the object into a new object in the same bucket.
    CopyObjectRequest copyObjectRequest = new CopyObjectRequest(accessPointArn,
sourceKey, accessPointArn, destinationKey);
    s3Client.copyObject(copyObjectRequest);
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

## Amazon S3 on Outposts バケットからオブジェクトを取得する

オブジェクトは、Amazon S3 on Outposts に保存される基本エンティティです。すべてのオブジェクトはバケット内に保存されます。Outpost バケット内の任意のオブジェクトにアクセスするには、アクセスポイントを使用する必要があります。オブジェクト操作のためにバケットを指定するときには、アクセスポイントの Amazon リソースネーム (ARN) またはアクセスポイントエイリアスを使用します。アクセスポイントエイリアスの詳細については、「[S3 on Outposts アクセスポイントでのバケット形式のエイリアスの使用](#)」を参照してください。

次の例は、S3 on Outposts アクセスポイントの ARN 形式を示し、これは、Outpost が属するリージョンの AWS リージョン コード、AWS アカウント ID、Outpost ID、アクセスポイント名を含みます。

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

S3 on Outposts ARN の詳細については、「[S3 on Outposts のリソース ARN](#)」を参照してください。

Amazon S3 on Outposts では、オブジェクトデータは常に Outpost に保存されます。AWS が Outpost ラックを設置すると、データの常駐要件を満たすために、データは Outpost にローカルに保たれます。オブジェクトが Outpost を離れたり、AWS リージョン 外に出たりすることはありません。AWS マネジメントコンソールはリージョン内でホストされるため、コンソールを使用して Outpost にオブジェクトをアップロードしたり、管理したりすることはできません。ただし REST API、AWS Command Line Interface (AWS CLI)、および AWS SDK を使用して、アクセスポイントを介してオブジェクトのアップロードと管理を行うことができます。

以下の例は、AWS Command Line Interface(AWS CLI) と AWS SDK for Java を使用してオブジェクトをダウンロード (取得) する方法を示しています。

## の使用AWS CLI

次の例では、S3 on Outposts バケット (s3-outposts:GetObject) から、AWS CLI を使って `sample-object.xml` という名前のオブジェクトを取得します。このコマンドを使用するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。このコマンドの詳細については、「AWS CLI リファレンス」の「[get-object](#)」を参照してください。

```
aws s3api get-object --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point --key testkey sample-object.xml
```

## AWS SDK for Java の使用

次の S3 on Outposts の例では、SDK for Java を使用してオブジェクトを取得します。この例を実行するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。詳細については、[Amazon Simple Storage Service API リファレンス](#)の `GetObject` を参照してください。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GetObjectRequest;
import com.amazonaws.services.s3.model.ResponseHeaderOverrides;
import com.amazonaws.services.s3.model.S3Object;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
```

```
public class GetObject {
    public static void main(String[] args) throws IOException {
        String accessPointArn = "*** access point ARN ***";
        String key = "*** Object key ***";

        S3Object fullObject = null, objectPortion = null, headerOverrideObject = null;
        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            // Get an object and print its contents.
            System.out.println("Downloading an object");
            fullObject = s3Client.getObject(new GetObjectRequest(accessPointArn, key));
            System.out.println("Content-Type: " +
fullObject.getObjectMetadata().getContentType());
            System.out.println("Content: ");
            displayTextInputStream(fullObject.getObjectContent());

            // Get a range of bytes from an object and print the bytes.
            GetObjectRequest rangeObjectRequest = new GetObjectRequest(accessPointArn,
key)
                .withRange(0, 9);
            objectPortion = s3Client.getObject(rangeObjectRequest);
            System.out.println("Printing bytes retrieved.");
            displayTextInputStream(objectPortion.getObjectContent());

            // Get an entire object, overriding the specified response headers, and
            print the object's content.
            ResponseHeaderOverrides headerOverrides = new ResponseHeaderOverrides()
                .withCacheControl("No-cache")
                .withContentDisposition("attachment; filename=example.txt");
            GetObjectRequest getObjectRequestHeaderOverride = new
GetObjectRequest(accessPointArn, key)
                .withResponseHeaders(headerOverrides);
            headerOverrideObject = s3Client.getObject(getObjectRequestHeaderOverride);
            displayTextInputStream(headerOverrideObject.getObjectContent());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
    }
}
```

```
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    } finally {
        // To ensure that the network connection doesn't remain open, close any
open input streams.
        if (fullObject != null) {
            fullObject.close();
        }
        if (objectPortion != null) {
            objectPortion.close();
        }
        if (headerOverrideObject != null) {
            headerOverrideObject.close();
        }
    }
}

private static void displayTextInputStream(InputStream input) throws IOException {
    // Read the text input stream one line at a time and display each line.
    BufferedReader reader = new BufferedReader(new InputStreamReader(input));
    String line = null;
    while ((line = reader.readLine()) != null) {
        System.out.println(line);
    }
    System.out.println();
}
}
```

## Amazon S3 on Outposts バケット内のオブジェクトを一覧表示する

オブジェクトは、Amazon S3 on Outposts に保存される基本エンティティです。すべてのオブジェクトはバケット内に保存されます。Outpost バケット内の任意のオブジェクトにアクセスするには、アクセスポイントを使用する必要があります。オブジェクト操作のためにバケットを指定するときには、アクセスポイントの Amazon リソースネーム (ARN) またはアクセスポイントエイリアスを使用します。アクセスポイントエイリアスの詳細については、「[S3 on Outposts アクセスポイントでのバケット形式のエイリアスの使用](#)」を参照してください。

次の例は、S3 on Outposts アクセスポイントの ARN 形式を示し、これは、Outpost が属するリージョンの AWS リージョン コード、AWS アカウント ID、Outpost ID、アクセスポイント名を含みます。

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

S3 on Outposts ARN の詳細については、「[S3 on Outposts のリソース ARN](#)」を参照してください。

#### Note

Amazon S3 on Outposts では、オブジェクトデータは常に Outpost に保存されます。AWS が Outpost ラックを設置すると、データの常駐要件を満たすために、データは Outpost にローカルに保たれます。オブジェクトが Outpost を離れたり、AWS リージョン 外に出たりすることはありません。AWS マネジメントコンソールはリージョン内でホストされるため、コンソールを使用して Outpost にオブジェクトをアップロードしたり、管理したりすることはできません。ただし REST API、AWS Command Line Interface (AWS CLI)、および AWS SDK を使用して、アクセスポイントを介してオブジェクトのアップロードと管理を行うことができます。

次の例は、AWS CLI および AWS SDK for Java を使用して、S3 on Outposts バケット内のオブジェクトを一覧表示する方法を示しています。

## AWS CLI の使用

次の例では、S3 on Outposts バケット (s3-outposts:ListObjectsV2) 内のオブジェクトを、AWS CLI を使用して一覧表示します。このコマンドを使用するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。このコマンドの詳細については、「AWS CLI リファレンス」の「[list-objects-v2](#)」を参照してください。

```
aws s3api list-objects-v2 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point
```

#### Note

AWS SDK を通じて Amazon S3 on Outposts でこのアクションを使用する場合、バケット名の代わりに Outposts のアクセスポイント ARN を以下の形式で提供します:

arn:aws:s3-outposts:*region*:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/*example-Outposts-Access-Point*。S3 on Outposts ARN の詳細については、「[S3 on Outposts のリソース ARN](#)」を参照してください。

## AWS SDK for Java の使用

次の S3 on Outposts の例では、SDK for Java を使用してバケット内のオブジェクトを一覧表示します。この例を実行するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。

### Important

この例では、ListObjectsAPI オペレーションの最新リビジョンである [ListObjectsV2](#) を使用します。この改訂版の API オペレーションをアプリケーション開発に使用することをお勧めします。下位互換性のために、Amazon S3 はこの API の以前のバージョンであるオペレーションを引き続きサポートします。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListObjectsV2Request;
import com.amazonaws.services.s3.model.ListObjectsV2Result;
import com.amazonaws.services.s3.model.S3ObjectSummary;

public class ListObjectsV2 {

    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();
```

```
System.out.println("Listing objects");

// maxKeys is set to 2 to demonstrate the use of
// ListObjectsV2Result.getNextContinuationToken()
ListObjectsV2Request req = new
ListObjectsV2Request().withBucketName(accessPointArn).withMaxKeys(2);
ListObjectsV2Result result;

do {
    result = s3Client.listObjectsV2(req);

    for (S3ObjectSummary objectSummary : result.getObjectSummaries()) {
        System.out.printf(" - %s (size: %d)\n", objectSummary.getKey(),
objectSummary.getSize());
    }
    // If there are more than maxKeys keys in the bucket, get a
continuation token
    // and list the next objects.
    String token = result.getNextContinuationToken();
    System.out.println("Next Continuation Token: " + token);
    req.setContinuationToken(token);
} while (result.isTruncated());
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
// it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
// couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

## Amazon S3 on Outposts バケット内のオブジェクトを削除する

オブジェクトは、Amazon S3 on Outposts に保存される基本エンティティです。すべてのオブジェクトはバケット内に保存されます。Outpost バケット内の任意のオブジェクトにアクセスするには、アクセスポイントを使用する必要があります。オブジェクト操作のためにバケットを指定するときには、アクセスポイントの Amazon リソースネーム (ARN) またはアクセスポイントエイリアスを使用します。アクセスポイントエイリアスの詳細については、「[S3 on Outposts アクセスポイントでのバケット形式のエイリアスの使用](#)」を参照してください。

次の例は、S3 on Outposts アクセスポイントの ARN 形式を示し、これは、Outpost が属するリージョンの AWS リージョン コード、AWS アカウント ID、Outpost ID、アクセスポイント名を含みます。

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

S3 on Outposts ARN の詳細については、「[S3 on Outposts のリソース ARN](#)」を参照してください。

Amazon S3 on Outposts では、オブジェクトデータは常に Outpost に保存されます。AWS が Outpost ラックを設置すると、データの常駐要件を満たすために、データは Outpost にローカルに保たれます。オブジェクトが Outpost を離れたり、AWS リージョン 外に出たりすることはありません。AWS マネジメントコンソールはリージョン内でホストされるため、コンソールを使用して Outpost にオブジェクトをアップロードしたり、管理したりすることはできません。ただし REST API、AWS Command Line Interface (AWS CLI)、および AWS SDK を使用して、アクセスポイントを介してオブジェクトのアップロードと管理を行うことができます。

次の例は、AWS Command Line Interface (AWS CLI) と AWS SDK for Java を使用して、S3 on Outposts バケット内の単一のオブジェクトまたは複数のオブジェクトを削除する方法を示しています。

## AWS CLI の使用

以下の例は、S3 on Outposts バケットから単一のオブジェクトまたは複数のオブジェクトを削除する方法を示しています。

### delete-object

次の例では、`sample-object.xml` を使用して、S3 on Outposts バケット (`s3-outposts>DeleteObject`) から AWS CLI という名前のオブジェクトを削除します。このコマンドを使用するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。このコマンドの詳細については、「[AWS CLI リファレンス](#)」の「delete-object」を参照してください。

```
aws s3api delete-object --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-access-point --key sample-object.xml
```

## delete-objects

以下の例では、AWS CLI を使用して、S3 on Outposts バケット (s3-outposts>DeleteObject) から sample-object.xml と test1.text という名前の 2 つのオブジェクトを削除します。このコマンドを使用するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。このコマンドの詳細については、「[AWS CLI リファレンス](#)」の「delete-objects」を参照してください。

```
aws s3api delete-objects --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point --delete file://delete.json
```

```
delete.json
{
  "Objects": [
    {
      "Key": "test1.txt"
    },
    {
      "Key": "sample-object.xml"
    }
  ],
  "Quiet": false
}
```

## AWS SDK for Java の使用

以下の例は、S3 on Outposts バケットから単一のオブジェクトまたは複数のオブジェクトを削除する方法を示しています。

### DeleteObject

次の S3 on Outposts の例では、SDK for Java を使用してバケット内のオブジェクトを削除します。この例を使用するには、Outpost のアクセスポイント ARN と、削除するオブジェクトのキー名を指定します。詳細については、Amazon Simple Storage Service API リファレンスの「[DeleteObject](#)」を参照してください。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
```

```
import com.amazonaws.services.s3.model.DeleteObjectRequest;

public class DeleteObject {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String keyName = "*** key name ****";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            s3Client.deleteObject(new DeleteObjectRequest(accessPointArn, keyName));
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        } catch (SdkClientException e) {
            // Amazon S3 couldn't be contacted for a response, or the client
            // couldn't parse the response from Amazon S3.
            e.printStackTrace();
        }
    }
}
```

## DeleteObjects

次の S3 on Outposts の例では、SDK for Java を使用してバケット内のオブジェクトをアップロードして、削除します。この例を使用するには、Outpost のアクセスポイント ARN を指定します。詳細については、「Amazon Simple Storage Service API リファレンス」の「[DeleteObject](#)」を参照してください。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.DeleteObjectsRequest;
import com.amazonaws.services.s3.model.DeleteObjectsRequest.KeyVersion;
import com.amazonaws.services.s3.model.DeleteObjectsResult;
```

```
import java.util.ArrayList;

public class DeleteObjects {

    public static void main(String[] args) {
        String accessPointArn = "arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-
outposts-access-point";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            // Upload three sample objects.
            ArrayList<KeyVersion> keys = new ArrayList<KeyVersion>();
            for (int i = 0; i < 3; i++) {
                String keyName = "delete object example " + i;
                s3Client.putObject(accessPointArn, keyName, "Object number " + i + "
to be deleted.");
                keys.add(new KeyVersion(keyName));
            }
            System.out.println(keys.size() + " objects successfully created.");

            // Delete the sample objects.
            DeleteObjectsRequest multiObjectDeleteRequest = new
DeleteObjectsRequest(accessPointArn)
                .withKeys(keys)
                .withQuiet(false);

            // Verify that the objects were deleted successfully.
            DeleteObjectsResult delObjRes =
s3Client.deleteObjects(multiObjectDeleteRequest);
            int successfulDeletes = delObjRes.getDeletedObjects().size();
            System.out.println(successfulDeletes + " objects successfully
deleted.");
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
    }
}
```

```
    } catch (SdkClientException e) {  
        // Amazon S3 couldn't be contacted for a response, or the client  
        // couldn't parse the response from Amazon S3.  
        e.printStackTrace();  
    }  
}  
}
```

## HeadBucket を使用して、S3 on Outposts バケットが存在し、アクセス許可があるかどうかを調べる

オブジェクトは、Amazon S3 on Outposts に保存される基本エンティティです。すべてのオブジェクトはバケット内に保存されます。Outpost バケット内の任意のオブジェクトにアクセスするには、アクセスポイントを使用する必要があります。オブジェクト操作のためにバケットを指定するときには、アクセスポイントの Amazon リソースネーム (ARN) またはアクセスポイントエイリアスを使用します。アクセスポイントエイリアスの詳細については、「[S3 on Outposts アクセスポイントでのバケット形式のエイリアスの使用](#)」を参照してください。

次の例は、S3 on Outposts アクセスポイントの ARN 形式を示し、これは、Outpost が属するリージョンの AWS リージョン コード、AWS アカウント ID、Outpost ID、アクセスポイント名を含みます。

```
arn:aws:s3-outposts:region:account-id:outpost/outpost-id/accesspoint/accesspoint-name
```

S3 on Outposts ARN の詳細については、「[S3 on Outposts のリソース ARN](#)」を参照してください。

### Note

Amazon S3 on Outposts では、オブジェクトデータは常に Outpost に保存されます。AWS が Outpost ラックを設置すると、データの常駐要件を満たすために、データは Outpost にローカルに保たれます。オブジェクトが Outpost を離れたり、AWS リージョン 外に出たりすることはありません。AWS マネジメントコンソールはリージョン内でホストされるため、コンソールを使用して Outpost にオブジェクトをアップロードしたり、管理したりすることはできません。ただし REST API、AWS Command Line Interface (AWS CLI)、および AWS SDK を使用して、アクセスポイントを介してオブジェクトのアップロードと管理を行うことができます。

以下の AWS Command Line Interface (AWS CLI) と AWS SDK for Java の例は、HeadBucket API オペレーションを使用して、Amazon S3 on Outposts バケットが存在し、そのバケットにアクセスする許可があるかどうかを調べる方法を示しています。詳細については、「Amazon Simple Storage Service API リファレンス」の「[HeadBucket](#)」を参照してください。

## の使用AWS CLI

次の S3 on Outposts AWS CLI の例では、`head-bucket` コマンドを使用して、バケットが存在し、そのバケットにアクセスする許可があるかどうかを調べます。このコマンドを使用するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。詳細については、「AWS CLI リファレンス」の「[head-bucket](#)」を参照してください。

```
aws s3api head-bucket --bucket arn:aws:s3-
outposts:region:123456789012:outpost/op-01ac5d28a6a232904/accesspoint/example-outposts-
access-point
```

## AWS SDK for Java の使用

次の S3 on Outposts の例は、バケットが存在するかどうか、また、そのバケットにアクセスする許可があるかどうかを調べる方法を示しています。この例を使用するには、Outpost のアクセスポイント ARN を指定します。詳細については、「Amazon Simple Storage Service API リファレンス」の「[HeadBucket](#)」を参照してください。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.HeadBucketRequest;

public class HeadBucket {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
            credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();
```

```
s3Client.headBucket(new HeadBucketRequest(accessPointArn));
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

## SDK for Java でのマルチパートアップロードの実行と管理

Amazon S3 on Outposts を使用すると、AWS Outposts リソースで S3 バケットを作成し、ローカルデータアクセス、ローカルデータ処理、データレジデンシーを必要とするアプリケーション用に、オンプレミスのオブジェクトを保存および取得できます。AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDK、または REST API を使用して S3 on Outposts を使用できます。詳細については、「[Amazon S3 on Outposts とは](#)」を参照してください。

以下の例は、AWS SDK for Java で S3 on Outposts を使用して、マルチパートアップロードを実行し、管理する方法を示しています。

### トピック

- [S3 on Outposts バケット内のオブジェクトのマルチパートアップロードを実行する](#)
- [S3 on Outposts バケット内のラージオブジェクトをマルチパートアップロードでコピーする](#)
- [S3 on Outposts バケットにオブジェクトのパーツを一覧表示する](#)
- [S3 on Outposts バケット内の進行中のマルチパートアップロードのリストを取得する](#)

## S3 on Outposts バケット内のオブジェクトのマルチパートアップロードを実行する

次の S3 on Outposts の例では、SDK for Java を使用して、バケットへオブジェクトのマルチパートアップロードを開始、アップロード、完了します。この例を実行するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。詳細については、「Amazon Simple Storage Service ユーザーガイド」の「[マルチパートアップロードを使用したオブジェクトのアップロード](#)」を参照してください。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.ArrayList;
import java.util.List;

public class MultipartUploadCopy {
    public static void main(String[] args) {
        String accessPointArn = "*** Source access point ARN ***";
        String sourceObjectKey = "*** Source object key ***";
        String destObjectKey = "*** Target object key ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
InitiateMultipartUploadRequest(accessPointArn, destObjectKey);
            InitiateMultipartUploadResult initResult =
s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
GetObjectMetadataRequest(accessPointArn, sourceObjectKey);
            ObjectMetadata metadataResult =
s3Client.getObjectMetadata(metadataRequest);
            long objectSize = metadataResult.getContentLength();

            // Copy the object using 5 MB parts.
            long partSize = 5 * 1024 * 1024;
            long bytePosition = 0;
            int partNum = 1;
            List<CopyPartResult> copyResponses = new ArrayList<CopyPartResult>();
            while (bytePosition < objectSize) {
                // The last part might be smaller than partSize, so check to make sure
```

```
// that lastByte isn't beyond the end of the object.
long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

// Copy this part.
CopyPartRequest copyRequest = new CopyPartRequest()
    .withSourceBucketName(accessPointArn)
    .withSourceKey(sourceObjectKey)
    .withDestinationBucketName(accessPointArn)
    .withDestinationKey(destObjectKey)
    .withUploadId(initResult.getUploadId())
    .withFirstByte(bytePosition)
    .withLastByte(lastByte)
    .withPartNumber(partNum++);
copyResponses.add(s3Client.copyPart(copyRequest));
bytePosition += partSize;
}

// Complete the upload request to concatenate all uploaded parts and make
the copied object available.
CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest(
    accessPointArn,
    destObjectKey,
    initResult.getUploadId(),
    getETags(copyResponses));
s3Client.completeMultipartUpload(completeRequest);
System.out.println("Multipart copy complete.");
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}

// This is a helper function to construct a list of ETags.
private static List<PartETag> getETags(List<CopyPartResult> responses) {
    List<PartETag> etags = new ArrayList<PartETag>();
    for (CopyPartResult response : responses) {
        etags.add(new PartETag(response.getPartNumber(), response.getETag()));
    }
}
```

```
    return etags;
}
```

## S3 on Outposts バケット内のラージオブジェクトをマルチパートアップロードでコピーする

次の S3 on Outposts の例では、SDK for Java を使用してバケットにオブジェクトをコピーします。この例を実行するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.ArrayList;
import java.util.List;

public class MultipartUploadCopy {
    public static void main(String[] args) {
        String accessPointArn = "*** Source access point ARN ***";
        String sourceObjectKey = "*** Source object key ***";
        String destObjectKey = "*** Target object key ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();

            // Initiate the multipart upload.
            InitiateMultipartUploadRequest initRequest = new
            InitiateMultipartUploadRequest(accessPointArn, destObjectKey);
            InitiateMultipartUploadResult initResult =
            s3Client.initiateMultipartUpload(initRequest);

            // Get the object size to track the end of the copy operation.
            GetObjectMetadataRequest metadataRequest = new
            GetObjectMetadataRequest(accessPointArn, sourceObjectKey);
```

```
ObjectMetadata metadataResult =
s3Client.getObjectMetadata(metadataRequest);
long objectSize = metadataResult.getContentLength();

// Copy the object using 5 MB parts.
long partSize = 5 * 1024 * 1024;
long bytePosition = 0;
int partNum = 1;
List<CopyPartResult> copyResponses = new ArrayList<CopyPartResult>();
while (bytePosition < objectSize) {
    // The last part might be smaller than partSize, so check to make sure
    // that lastByte isn't beyond the end of the object.
    long lastByte = Math.min(bytePosition + partSize - 1, objectSize - 1);

    // Copy this part.
    CopyPartRequest copyRequest = new CopyPartRequest()
        .withSourceBucketName(accessPointArn)
        .withSourceKey(sourceObjectKey)
        .withDestinationBucketName(accessPointArn)
        .withDestinationKey(destObjectKey)
        .withUploadId(initResult.getUploadId())
        .withFirstByte(bytePosition)
        .withLastByte(lastByte)
        .withPartNumber(partNum++);
    copyResponses.add(s3Client.copyPart(copyRequest));
    bytePosition += partSize;
}

// Complete the upload request to concatenate all uploaded parts and make
the copied object available.
CompleteMultipartUploadRequest completeRequest = new
CompleteMultipartUploadRequest(
    accessPointArn,
    destObjectKey,
    initResult.getUploadId(),
    getETags(copyResponses));
s3Client.completeMultipartUpload(completeRequest);
System.out.println("Multipart copy complete.");
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
```

```
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}

// This is a helper function to construct a list of ETags.
private static List<PartETag> getETags(List<CopyPartResult> responses) {
    List<PartETag> etags = new ArrayList<PartETag>();
    for (CopyPartResult response : responses) {
        etags.add(new PartETag(response.getPartNumber(), response.getETag()));
    }
    return etags;
}
}
```

## S3 on Outposts バケットにオブジェクトのパーツを一覧表示する

次の S3 on Outposts の例では、SDK for Java を使用してバケット内のオブジェクトのパーツを一覧表示します。この例を実行するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.*;

import java.util.List;

public class ListParts {
    public static void main(String[] args) {
        String accessPointArn = "*** access point ARN ***";
        String keyName = "*** Key name ***";
        String uploadId = "*** Upload ID ***";

        try {
            // This code expects that you have AWS credentials set up per:
            // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-credentials.html
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .enableUseArnRegion()
                .build();
```

```
        ListPartsRequest listPartsRequest = new ListPartsRequest(accessPointArn,
keyName, uploadId);
        PartListing partListing = s3Client.listParts(listPartsRequest);
        List<PartSummary> partSummaries = partListing.getParts();

        System.out.println(partSummaries.size() + " multipart upload parts");
        for (PartSummary p : partSummaries) {
            System.out.println("Upload part: Part number = \"" + p.getPartNumber()
+ "\", ETag = " + p.getETag());
        }

    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

## S3 on Outposts バケット内の進行中のマルチパートアップロードのリストを取得する

次の S3 on Outposts の例では、Outposts バケットから SDK for Java を使用して、進行中のマルチパートアップロードのリストを取得する方法を示します。この例を実行するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.ListMultipartUploadsRequest;
import com.amazonaws.services.s3.model.MultipartUpload;
import com.amazonaws.services.s3.model.MultipartUploadListing;

import java.util.List;

public class ListMultipartUploads {
    public static void main(String[] args) {
```

```
String accessPointArn = "*** access point ARN ***";

try {
    // This code expects that you have AWS credentials set up per:
    // https://docs.aws.amazon.com/sdk-for-java/v1/developer-guide/setup-
credentials.html
    AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
        .enableUseArnRegion()
        .build();

    // Retrieve a list of all in-progress multipart uploads.
    ListMultipartUploadsRequest allMultipartUploadsRequest = new
ListMultipartUploadsRequest(accessPointArn);
    MultipartUploadListing multipartUploadListing =
s3Client.listMultipartUploads(allMultipartUploadsRequest);
    List<MultipartUpload> uploads =
multipartUploadListing.getMultipartUploads();

    // Display information about all in-progress multipart uploads.
    System.out.println(uploads.size() + " multipart upload(s) in progress.");
    for (MultipartUpload u : uploads) {
        System.out.println("Upload in progress: Key = \"" + u.getKey() + "\",
id = " + u.getUploadId());
    }
} catch (AmazonServiceException e) {
    // The call was transmitted successfully, but Amazon S3 couldn't process
    // it, so it returned an error response.
    e.printStackTrace();
} catch (SdkClientException e) {
    // Amazon S3 couldn't be contacted for a response, or the client
    // couldn't parse the response from Amazon S3.
    e.printStackTrace();
}
}
```

## S3 on Outposts での署名付き URL の使用

バケットポリシーを更新せずに、Outpost にローカルに保存されているオブジェクトへの時間制限付きのアクセス許可を付与するには、事前署名付きの URL を使用します。署名付き URL を使用すると、バケット所有者は仮想プライベートクラウド (VPC) 内の個人とオブジェクトを共有したり、オブジェクトをアップロードまたは削除する権限を、これらのユーザーに付与したりできます。

AWS SDK または AWS Command Line Interface(AWS CLI) を使用して署名付き URL を作成する際には、その URL に対し特定のアクションを関連付けます。また、1 秒から 7 日まで指定可能なカスタム有効期限を選択して、署名付き URL に対し期間制限付きのアクセス許可を付与できます。署名付き URL を共有すると、VPC 内のユーザーは署名元のユーザーと同じように、URL に埋め込まれたアクションを実行できるようになります。この URL は有効期限が切れると失効し、以後は機能しなくなります。

## 署名付き URL 機能の制限

署名付き URL の機能は、それを作成したユーザーの許可によって制限されます。本質的に署名付き URL は、それらを保有しているユーザーに対しアクセスを許可するためのペアトークンです。そのため、適切に保護することをお勧めします。

### AWS Signature Version 4 (SigV4)

署名済み URL リクエストが AWS Signature Version 4 (SigV4) により認証される際に実行する特定の動作を適用するには、バケットポリシーとアクセスポイントポリシーで条件キーを使用します。例えば、`s3-outposts:signatureAge` 条件を使用するバケットポリシーを作成し、署名が作成されてから 10 分以上経過している場合に、`example-outpost-bucket` バケット内のオブジェクトに対する Amazon S3 on Outposts の署名付き URL リクエストを、すべて拒否することができます。この例を実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

### JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Deny a presigned URL request if the signature is more than 10 minutes old",
      "Effect": "Deny",
      "Principal": {"AWS": "444455556666"},
      "Action": "s3-outposts:*",
      "Resource": "arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/*",
      "Condition": {
        "NumericGreaterThan": {"s3-outposts:signatureAge": 600000},
        "StringEquals": {"s3-outposts:authType": "REST-QUERY-STRING"}
      }
    }
  ]
}
```

```
    }  
  ]  
}
```

Signature Version 4 を使用している署名付き URL リクエストで、その認証時に実行する特定の動作を適用するための条件キー、および他のポリシー例のリストについては、「[AWS Signature Version 4 \(SigV4\) での認証固有ポリシーキー](#)」を参照してください。

## ネットワークパスでの制限

署名付き URL の使用と、S3 on Outposts アクセスをすべて特定のネットワークパスに制限する場合は、その特定のネットワークパスを使用しながら (IAM) ポリシーを記述できます。呼び出しを行う IAM プリンシパルに対する制限を設定するには、ID ベースの AWS Identity and Access Management (IAM) ポリシー (ユーザー、グループ、ロールポリシーなど) を使用できます。S3 on Outposts リソースでの制限を設定する場合は、リソースベースのポリシー (バケットポリシーやアクセスポイントポリシーなど) を使用します。

IAM プリンシパルでのネットワークパスの制限では、これらの認証情報のユーザーは、指定したネットワークからリクエストを送信する必要があります。バケットまたはアクセスポイントの制限により、そのリソースに対するすべてのリクエストは、指定したネットワークから発信される必要があります。これらの制限は、署名付き URL のシナリオ以外でも適用されます。

使用する IAM グローバル条件は、エンドポイントのタイプによって異なります。S3 on Outposts でパブリックエンドポイントを使用している場合は、aws:SourceIp を使用します。S3 on Outposts の VPC エンドポイントを使用している場合は、aws:SourceVpc または aws:SourceVpce を使用します。

次の IAM ポリシーステートメントでは、プリンシパルは、指定されたネットワーク範囲からのみ AWS にアクセスする必要があります。このポリシーステートメントでは、すべてのアクセスがその範囲から発信される必要があります。これは、S3 on Outposts の署名付き URL を使用するユーザーに対しても当てはまります。この例を実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
{  
  "Sid": "NetworkRestrictionForIAMPrincipal",  
  "Effect": "Deny",  
  "Action": "*",  
  "Resource": "*",  
  "Condition": {
```

```
"NotIpAddressIfExists": {"aws:SourceIp": "IP-address-range"},
"BoolIfExists": {"aws:ViaAWSService": "false"}
}
}
```

S3 on Outposts バケットへのアクセスを特定のネットワーク範囲に制限するため、aws:SourceIP AWS グローバル条件キーを使用するバケットポリシーの例については、「[S3 on Outposts で IAM を設定する](#)」を参照してください。

## 署名付き URL を作成できるユーザー

有効なセキュリティ認証情報を持つすべてのユーザーが、署名付き URL を作成できます。しかし、VPC 内のユーザーがオブジェクトに正常にアクセスするには、署名付き URL をベースにするオペレーションの実行許可を持っているユーザーにより、この署名付き URL が作成されている必要があります。

以下の認証情報は、署名付き URL の作成に使用することができます。

- IAM インスタンスプロファイル – 最大 6 時間有効。
- AWS Security Token Service – AWS アカウント ルートユーザーや IAM ユーザーの認証情報など、永続的な認証情報を使用して署名されている場合は、最大 36 時間まで有効。
- IAM ユーザー – AWS Signature Version 4 を使用している場合は、最大 7 日間まで有効。

最大 7 日間有効の署名付き URL を作成するには、まず、使用する SDK への IAM ユーザー認証情報 (アクセスキーとシークレットキー) の委任を行います。次に、AWS Signature Version 4 を使用して署名付き URL を生成します。

### Note

- 一時トークンを使用して作成した署名付き URL において、URL の有効期限の終了より先にそのトークンが有効期限切れになった場合は、URL も失効します。
- 署名付き URL は、その URL を保有しているすべてのユーザーに S3 on Outposts バケットへのアクセスを許可するため、適切な保護を行うことをお勧めします。署名付き URL の保護の詳細については、「[署名付き URL 機能の制限](#)」を参照してください。

## S3 on Outposts は、どのタイミングで署名付き URL の有効期限切れの日時を確認しますか？

S3 on Outposts は、HTTP リクエスト時に署名付き URL の有効期限日時を確認します。例えば、有効期限が切れる時刻の直前にクライアントが大きなファイルのダウンロードを開始した場合は、ダウンロード中に有効期限時刻が経過しても、そのダウンロードは継続されます。しかし、接続が中断し、クライアントがダウンロードを再開しようとした時点で有効期限切れの時刻が経過している場合には、そのダウンロードは失敗します。

署名付き URL を使用してオブジェクトを共有またはアップロードする方法の詳細については、次のトピックを参照してください。

### トピック

- [署名付き URL を使用したオブジェクトの共有](#)
- [S3 on Outposts バケットにオブジェクトをアップロードするための署名付き URL の生成](#)

## 署名付き URL を使用したオブジェクトの共有

バケットポリシーを更新せずに、Outpost にローカルに保存されているオブジェクトへの時間制限付きのアクセス許可を付与するには、事前署名付きの URL を使用します。署名付き URL を使用すると、バケット所有者は仮想プライベートクラウド (VPC) 内の個人とオブジェクトを共有したり、オブジェクトをアップロードまたは削除する権限を、これらのユーザーに付与したりできます。

AWS SDK または AWS Command Line Interface(AWS CLI) を使用して署名付き URL を作成する際には、その URL に対し特定のアクションを関連付けます。また、1 秒から 7 日まで指定可能なカスタム有効期限を選択して、署名付き URL に対し期間制限付きのアクセス許可を付与できます。署名付き URL を共有すると、VPC 内のユーザーは署名元のユーザーと同じように、URL に埋め込まれたアクションを実行できるようになります。この URL は有効期限が切れると失効し、以後は機能しなくなります。

署名付き URL を作成する場合には、ご自身のセキュリティ認証情報を設定し、さらに次の情報を指定する必要があります。

- Outposts バケット上の Amazon S3 アクセスポイントの Amazon リソースネーム (ARN)
- オブジェクトキー
- HTTP メソッド (オブジェクトをダウンロードするための GET)

- 有効期限の終了日時

署名付き URL は、指定した期間内でのみ有効です。つまり、有効期限が切れる日時の前に、URL で許可されているアクションを開始する必要があります。有効期限が切れる日時までは、署名付き URL を複数回使用できます。一時トークンを使用して署名付き URL を作成した場合、トークンが有効期限切れになると、URL の有効期限より前であってもその URL は失効します。

署名付き URL へのアクセス権を持つ仮想プライベートクラウド (VPC) のユーザーは、このオブジェクトにアクセスできます。例えば、プライベートのバケット内にプライベートの動画を格納している場合は、署名付き URL を生成することで、その動画を他ユーザーと共有できます。署名付き URL は、その URL を保有するすべてのユーザーに S3 on Outposts バケットへのアクセス許可を付与するため、この URL には適切な保護を行うことをお勧めします。署名付き URL の保護の詳細については、[署名付き URL 機能の制限](#) を参照してください。

有効なセキュリティ認証情報を持つすべてのユーザーが、署名付き URL を作成できます。ただし、署名付き URL の基となるオペレーションを実行するアクセス許可を持つユーザーが、その署名付き URL を作成する必要があります。詳細については、「[署名付き URL を作成できるユーザー](#)」を参照してください。

AWS SDK と AWS CLI を使用すると、S3 on Outposts バケット内にあるオブジェクトを共有するための署名付き URL を生成できます。詳細については、以下の例を参照してください。

## AWS SDK の使用

AWS SDK を使用して署名付き URL を生成します。この URL を他ユーザーに配布すると、そのユーザーがオブジェクトを取得できるようになります。

### Note

署名付き URL の生成に AWS SDK を使用した場合、その署名付き URL の最大有効期限は、作成時点から 7 日間となります。

## Java

### Example

次のサンプルコードは、署名付き URL を生成します。この URL は、Outposts バケット上の S3 からオブジェクトを取得可能にするために他のユーザーに配布できます。詳細については、「[S3](#)

[on Outposts での署名付き URL の使用](#)」を参照してください。この例を実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.HttpMethod;
import com.amazonaws.SdkClientException;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.s3.AmazonS3;
import com.amazonaws.services.s3.AmazonS3ClientBuilder;
import com.amazonaws.services.s3.model.GeneratePresignedUrlRequest;

import java.io.IOException;
import java.net.URL;
import java.time.Instant;

public class GeneratePresignedURL {

    public static void main(String[] args) throws IOException {
        Regions clientRegion = Regions.DEFAULT_REGION;
        String accessPointArn = "*** access point ARN ***";
        String objectKey = "*** object key ***";

        try {
            AmazonS3 s3Client = AmazonS3ClientBuilder.standard()
                .withRegion(clientRegion)
                .withCredentials(new ProfileCredentialsProvider())
                .build();

            // Set the presigned URL to expire after one hour.
            java.util.Date expiration = new java.util.Date();
            long expTimeMillis = Instant.now().toEpochMilli();
            expTimeMillis += 1000 * 60 * 60;
            expiration.setTime(expTimeMillis);

            // Generate the presigned URL.
            System.out.println("Generating pre-signed URL.");
            GeneratePresignedUrlRequest generatePresignedUrlRequest =
                new GeneratePresignedUrlRequest(accessPointArn, objectKey)
                    .withMethod(HttpMethod.GET)
                    .withExpiration(expiration);
            URL url = s3Client.generatePresignedUrl(generatePresignedUrlRequest);
```

```
        System.out.println("Pre-Signed URL: " + url.toString());
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 couldn't
process
        // it, so it returned an error response.
        e.printStackTrace();
    } catch (SdkClientException e) {
        // Amazon S3 couldn't be contacted for a response, or the client
        // couldn't parse the response from Amazon S3.
        e.printStackTrace();
    }
}
}
```

## .NET

### Example

次のサンプルコードは、署名付き URL を生成します。この URL は、Outposts バケット上の S3 からオブジェクトを取得可能にするために他のユーザーに配布できます。詳細については、「[S3 on Outposts での署名付き URL の使用](#)」を参照してください。この例を実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
using Amazon;
using Amazon.S3;
using Amazon.S3.Model;
using System;

namespace Amazon.DocSamples.S3
{
    class GenPresignedURLTest
    {
        private const string accessPointArn = "*** access point ARN ***";
        private const string objectKey = "*** object key ***";
        // Specify how long the presigned URL lasts, in hours.
        private const double timeoutDuration = 12;
        // Specify your bucket Region (an example Region is shown).
        private static readonly RegionEndpoint bucketRegion =
RegionEndpoint.USWest2;
        private static IAmazonS3 s3Client;

        public static void Main()
        {
```

```
s3Client = new AmazonS3Client(bucketRegion);
string urlString = GeneratePreSignedURL(timeoutDuration);
}
static string GeneratePreSignedURL(double duration)
{
    string urlString = "";
    try
    {
        GetPreSignedUrlRequest request1 = new GetPreSignedUrlRequest
        {
            BucketName = accessPointArn,
            Key = objectKey,
            Expires = DateTime.UtcNow.AddHours(duration)
        };
        urlString = s3Client.GetPreSignedURL(request1);
    }
    catch (AmazonS3Exception e)
    {
        Console.WriteLine("Error encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine("Unknown encountered on server. Message:'{0}' when
writing an object", e.Message);
    }
    return urlString;
}
}
}
```

## Python

次の例では、SDK for Python (Boto3) を使用して、オブジェクトを共有するための署名付き URL を生成しています。例えば、Boto3 クライアントおよび `generate_presigned_url` 関数を使用して、オブジェクトを GET する署名付き URL を生成します。

```
import boto3
url = boto3.client('s3').generate_presigned_url(
    ClientMethod='get_object',
    Params={'Bucket': 'ACCESS_POINT_ARN', 'Key': 'OBJECT_KEY'},
    ExpiresIn=3600)
```

SDK for Python (Boto3) を使用して署名付き URL を生成する方法については、「AWS SDK for Python (Boto) API リファレンス」の「[Python](#)」を参照してください。

## の使用AWS CLI

次の例では、AWS CLI コマンドにより、S3 on Outposts バケットのための署名付き URL を生成します。この例を実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

### Note

署名付き URL の作成に AWS CLI を使用した場合、その署名付き URL の最大有効期限は、作成時点から 7 日間になります。

```
aws s3 presign s3://arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/example-outpost-access-point/mydoc.txt --expires-in 604800
```

詳細については、「AWS CLI コマンドリファレンス」の「[presign](#)」を参照してください。

## S3 on Outposts バケットにオブジェクトをアップロードするための署名付き URL の生成

バケットポリシーを更新せずに、Outpost にローカルに保存されているオブジェクトへの時間制限付きのアクセス許可を付与するには、事前署名付きの URL を使用します。署名付き URL を使用すると、バケット所有者は仮想プライベートクラウド (VPC) 内の個人とオブジェクトを共有したり、オブジェクトをアップロードまたは削除する権限を、これらのユーザーに付与したりできます。

AWS SDK または AWS Command Line Interface(AWS CLI) を使用して署名付き URL を作成する際には、その URL に対し特定のアクションを関連付けます。また、1 秒から 7 日まで指定可能なカスタム有効期限を選択して、署名付き URL に対し期間制限付きのアクセス許可を付与できます。署名付き URL を共有すると、VPC 内のユーザーは署名元のユーザーと同じように、URL に埋め込まれたアクションを実行できるようになります。この URL は有効期限が切れると失効し、以後は機能しなくなります。

署名付き URL を作成する場合には、ご自身のセキュリティ認証情報を設定し、さらに次の情報を指定する必要があります。

- Outposts バケット上の Amazon S3 アクセスポイントの Amazon リソースネーム (ARN)
- オブジェクトキー
- HTTP メソッド (オブジェクトをアップロードするための PUT)
- 有効期限の終了日時

署名付き URL は、指定した期間内でのみ有効です。つまり、有効期限が切れる日時の前に、URL で許可されているアクションを開始する必要があります。有効期限が切れる日時までは、署名付き URL を複数回使用できます。一時トークンを使用して署名付き URL を作成した場合、トークンが有効期限切れになると、URL の有効期限より前であってもその URL は失効します。

マルチパートアップロードなど、複数のステップで構成されているアクションでは、すべてのステップを有効期限が切れる前に開始する必要があります。有効期限が切れた URL を使用して、S3 on Outposts がステップの開始を試みた場合は、エラーが発生します。

署名付き URL へのアクセス権を持つ仮想プライベートクラウド (VPC) 内のユーザーは、オブジェクトをアップロードすることが可能です。例えば、署名付き URL にアクセスできる VPC 内のユーザーは、お客様のバケットへのオブジェクトのアップロードが可能です。署名付き URL は、その URL を保有するすべてのユーザーに対し、S3 on Outposts バケットへのアクセスを許可するため、これらの URL には適切な保護を行うことをお勧めします。署名付き URL の保護の詳細については、[署名付き URL 機能の制限](#) を参照してください。

有効なセキュリティ認証情報を持つすべてのユーザーが、署名付き URL を作成できます。ただし、署名付き URL の基となるオペレーションを実行するアクセス許可を持つユーザーが、その署名付き URL を作成する必要があります。詳細については、「[署名付き URL を作成できるユーザー](#)」を参照してください。

## AWS SDK を使用した S3 on Outposts オブジェクトオペレーションへの署名付き URL の生成

### Java

#### SDK for Java 2.x

この例では、限られた時間内に S3 on Outposts バケットにオブジェクトをアップロードするために使用できる、署名付き URL を生成する方法を説明します。詳細については、「[S3 on Outposts での署名付き URL の使用](#)」を参照してください。

```
public static void signBucket(S3Presigner presigner, String
outpostAccessPointArn, String keyName) {
```

```
try {
    PutObjectRequest objectRequest = PutObjectRequest.builder()
        .bucket(accessPointArn)
        .key(keyName)
        .contentType("text/plain")
        .build();

    PutObjectPresignRequest presignRequest =
PutObjectPresignRequest.builder()
        .signatureDuration(Duration.ofMinutes(10))
        .putObjectRequest(objectRequest)
        .build();

    PresignedPutObjectRequest presignedRequest =
presigner.presignPutObject(presignRequest);

    String myURL = presignedRequest.url().toString();
    System.out.println("Presigned URL to upload a file to: " +myURL);
    System.out.println("Which HTTP method must be used when uploading a
file: " +
        presignedRequest.httpRequest().method());

    // Upload content to the S3 on Outposts bucket by using this URL.
    URL url = presignedRequest.url();

    // Create the connection and use it to upload the new object by using
the presigned URL.
    HttpURLConnection connection = (HttpURLConnection)
url.openConnection();
    connection.setDoOutput(true);
    connection.setRequestProperty("Content-Type", "text/plain");
    connection.setRequestMethod("PUT");
    OutputStreamWriter out = new
OutputStreamWriter(connection.getOutputStream());
    out.write("This text was uploaded as an object by using a presigned
URL.");
    out.close();

    connection.getResponseCode();
    System.out.println("HTTP response code is " +
connection.getResponseCode());
}
```

```
    } catch (S3Exception e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

## Python

### SDK for Python (Boto3)

S3 on Outposts アクションを期間限定で実行できる署名付き URL の生成方法を示します。詳細については、「[S3 on Outposts での署名付き URL の使用](#)」を参照してください。この URL でリクエストを行うには、Requests パッケージを使用します。

```
import argparse
import logging
import boto3
from botocore.exceptions import ClientError
import requests

logger = logging.getLogger(__name__)

def generate_presigned_url(s3_client, client_method, method_parameters,
    expires_in):
    """
    Generate a presigned S3 on Outposts URL that can be used to perform an
    action.

    :param s3_client: A Boto3 Amazon S3 client.
    :param client_method: The name of the client method that the URL performs.
    :param method_parameters: The parameters of the specified client method.
    :param expires_in: The number of seconds that the presigned URL is valid for.
    :return: The presigned URL.
    """
    try:
        url = s3_client.generate_presigned_url(
            ClientMethod=client_method,
            Params=method_parameters,
            ExpiresIn=expires_in
        )
```

```
        logger.info("Got presigned URL: %s", url)
    except ClientError:
        logger.exception(
            "Couldn't get a presigned URL for client method '%s'.",
client_method)
        raise
    return url

def usage_demo():
    logging.basicConfig(level=logging.INFO, format='%(levelname)s: %(message)s')

    print('-'*88)
    print("Welcome to the Amazon S3 on Outposts presigned URL demo.")
    print('-'*88)

    parser = argparse.ArgumentParser()
    parser.add_argument('accessPointArn', help="The name of the S3 on Outposts
access point ARN.")
    parser.add_argument(
        'key', help="For a GET operation, the key of the object in S3 on
Outposts. For a "
            "PUT operation, the name of a file to upload.")
    parser.add_argument(
        'action', choices=('get', 'put'), help="The action to perform.")
    args = parser.parse_args()

    s3_client = boto3.client('s3')
    client_action = 'get_object' if args.action == 'get' else 'put_object'
    url = generate_presigned_url(
        s3_client, client_action, {'Bucket': args.accessPointArn, 'Key':
args.key}, 1000)

    print("Using the Requests package to send a request to the URL.")
    response = None
    if args.action == 'get':
        response = requests.get(url)
    elif args.action == 'put':
        print("Putting data to the URL.")
        try:
            with open(args.key, 'r') as object_file:
                object_text = object_file.read()
            response = requests.put(url, data=object_text)
        except FileNotFoundError:
```

```
        print(f"Couldn't find {args.key}. For a PUT operation, the key must
be the "
            f"name of a file that exists on your computer.")

    if response is not None:
        print("Got response:")
        print(f"Status: {response.status_code}")
        print(response.text)

    print('-'*88)

if __name__ == '__main__':
    usage_demo()
```

## ローカルの Amazon EMR on Outposts を使用した Amazon S3 on Outposts

Amazon EMR は、Apache Hadoop や Apache Spark などのビッグデータフレームワークを AWS で簡単に実行して、膨大な量のデータを処理および分析できるマネージドクラスタープラットフォームです。これらのフレームワークと、関連するオープンソースプロジェクトを使用することで、分析用のデータやビジネスインテリジェンスワークロードを処理できます。Amazon EMR は、大量のデータを変換したり、他の AWS データストアやデータベースとの間でも出し入れしたりできるようにします。また、Amazon S3 on Outposts をサポートしています。Amazon EMR の詳細については、「Amazon EMR 管理ガイド」の「[Outposts での Amazon EMR](#)」を参照してください。

Amazon S3 on Outposts については、Amazon EMR はバージョン 7.0.0 で Apache Hadoop S3A コネクタのサポートを開始しました。以前のバージョンの Amazon EMR はローカルの S3 on Outposts をサポートしておらず、EMR ファイルシステム (EMRFS) はサポートされていません。

### サポートされているアプリケーション

Amazon S3 on Outposts を使用した Amazon EMR は、次のアプリケーションをサポートしています。

- Hadoop
- Spark
- Hue
- Hive

- Sqoop
- Pig
- Hudi
- Flink

詳細については、「[Amazon EMR リリースガイド](#)」を参照してください。

## Amazon S3 on Outposts バケットを作成して設定する

Amazon EMR では、Amazon S3 on Outposts で AWS SDK for Java を使用して、入力データと出力データを保存します。Amazon EMR ログファイルは、選択したリージョンの Amazon S3 ロケーションに保存され、Outpost にローカルに保存されることはありません。詳細については、「Amazon EMR 管理ガイド」の「[Amazon EMR ログ](#)」を参照してください。

Amazon S3 と DNS の要件に従うには、S3 on Outposts バケットに命名に関する制約と制限があります。詳細については、「[S3 on Outposts バケットを作成する](#)」を参照してください。

Amazon EMR バージョン 7.0.0 以降では、S3 on Outposts と S3A ファイルシステムで Amazon EMR を使用できます。

### 前提条件

S3 on Outposts アクセス許可 – Amazon EMR インスタンスプロファイルを作成する場合、ロールには S3 on Outposts の AWS Identity and Access Management (IAM) 名前空間が含まれている必要があります。S3 on Outposts には独自の名前空間、`s3-outposts*` があります。この名前空間を使用するポリシーの例については、「[S3 on Outposts で IAM を設定する](#)」を参照してください。

S3A コネクタ – Amazon S3 on Outposts バケットのデータにアクセスするように EMR クラスターを設定するには、Apache Hadoop S3A コネクタを使用する必要があります。コネクタを使用するには、すべての S3 URI が `s3a` スキームを使用していることを確認してください。そうでない場合は、S3 URI が S3A コネクタと連携するように EMR クラスターに使用するファイルシステムの実装を設定できます。

S3A コネクタと連携するようにファイルシステムの実装を設定するには、EMR クラスターの `fs.file_scheme.impl` および `fs.AbstractFileSystem.file_scheme.impl` 設定プロパティを使用します。*file\_scheme* は使用している S3 URI のタイプに対応します。次の例を使用するには、*user input placeholders* をユーザー自身の情報に置き換えます。例えば、`s3` スキームを使用する S3 URI ファイルシステムの実装を変更するには、次のクラスター設定プロパティを指定します。

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "fs.s3.impl": "org.apache.hadoop.fs.s3a.S3AFileSystem",
      "fs.AbstractFileSystem.s3.impl": "org.apache.hadoop.fs.s3a.S3A"
    }
  }
]
```

S3A を使用するには、`fs.file_scheme.impl` 設定プロパティを `org.apache.hadoop.fs.s3a.S3AFileSystem` に設定し、`fs.AbstractFileSystem.file_scheme.impl` プロパティを `org.apache.hadoop.fs.s3a.S3A` に設定します。

例えば、パス `s3a://bucket/...` にアクセスする場合は、`fs.s3a.impl` プロパティを `org.apache.hadoop.fs.s3a.S3AFileSystem` に設定し、`fs.AbstractFileSystem.s3a.impl` プロパティを `org.apache.hadoop.fs.s3a.S3A` に設定します。

## Amazon S3 on Outposts での Amazon EMR の開始方法

以下のトピックでは、Amazon S3 on Outposts で Amazon EMR の使用を開始する方法について説明します。

### トピック

- [許可ポリシーを作成する](#)
- [クラスターを作成および設定する](#)
- [設定の概要](#)
- [考慮事項](#)

### 許可ポリシーを作成する

Amazon S3 on Outposts を使用する EMR クラスターを作成する前に、クラスターの Amazon EC2 インスタンスプロファイルにアタッチする IAM ポリシーを作成する必要があります。ポリシーには、S3 on Outposts アクセスポイントの Amazon リソースネーム (ARN) にアクセスするためのア

アクセス許可が必要です。S3 on Outposts の IAM ポリシーの作成に関する詳細については、「[S3 on Outposts で IAM を設定する](#)」を参照してください。

次のポリシー例は、必要なアクセス許可を付与する方法を示します。ポリシーを作成したら、[the section called “クラスターを作成および設定する”](#) セクションで説明されているように、お使いの EMR クラスターの作成に使用するインスタンスプロファイルロールにポリシーをアタッチします。この例を実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Resource": "arn:aws:s3-outposts:us-  
west-2:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/access-point-name",
      "Action": [
        "s3-outposts:*"
      ]
    }
  ]
}
```

## クラスターを作成および設定する

S3 on Outposts で Spark を実行するクラスターを作成するには、コンソールで次の手順を実行します。

S3 on Outposts で Spark を実行するクラスターを作成するには

1. Amazon EMR コンソール (<https://console.aws.amazon.com/elasticmapreduce/>) を開きます。
2. 左のナビゲーションペインで [クラスター] を選択します。
3. [クラスターを作成] を選択します。
4. [Amazon EMR リリース] で、emr-7.0.0 以降を選択します。
5. アプリケーションバンドルで、[Spark インタラクティブ] を選択します。次に、クラスターに含める他のサポートされているアプリケーションを選択します。
6. Amazon S3 on Outposts を有効にするには、構成設定を入力します。

### 構成設定の例

次の構成設定の例を使用するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "fs.s3a.bucket.DOC-EXAMPLE-BUCKET.accesspoint.arn": "arn:aws:s3-outposts:us-
west-2:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/access-point-name"
      "fs.s3a.committer.name": "magic",
      "fs.s3a.select.enabled": "false"
    }
  },
  {
    "Classification": "hadoop-env",
    "Configurations": [
      {
        "Classification": "export",
        "Properties": {
          "JAVA_HOME": "/usr/lib/jvm/java-11-amazon-corretto.x86_64"
        }
      }
    ],
    "Properties": {}
  },
  {
    "Classification": "spark-env",
    "Configurations": [
      {
        "Classification": "export",
        "Properties": {
          "JAVA_HOME": "/usr/lib/jvm/java-11-amazon-corretto.x86_64"
        }
      }
    ],
    "Properties": {}
  },
  {
    "Classification": "spark-defaults",
    "Properties": {
      "spark.executorEnv.JAVA_HOME": "/usr/lib/jvm/java-11-amazon-
corretto.x86_64",
```

```

        "spark.sql.sources.fastS3PartitionDiscovery.enabled": "false"
    }
}
]

```

7. [ネットワーク] セクションで、AWS Outposts ラック上にある仮想プライベートクラウド (VPC) とサブネットを選択します。Amazon EMR on Outposts の詳細については、「Amazon EMR 管理ガイド」の「[AWS Outposts での EMR クラスター](#)」を参照してください。
8. [Amazon EMR の EC2 インスタンスプロファイル] セクションで、[前に作成したアクセス許可ポリシー](#)がアタッチされている IAM ロールを選択します。
9. 残りのクラスター設定を行い、[クラスターの作成] を選択します。

## 設定の概要

次の表は、Amazon EMR で S3 on Outposts を使用するクラスターをセットアップするときに、S3A 設定とそのパラメータに指定する値を示しています。

パラメータ	デフォルト値	S3 on Outposts に必要な値	説明
<code>fs.s3a.aws.credentials.provider</code>	指定しない場合、S3A は Outposts のバケット名を持つリージョンバケットで S3 を検索します。	S3 on Outposts バケットのアクセスポイント ARN	Amazon S3 on Outposts は、Outposts バケットにアクセスする唯一の手段として、Virtual Private Cloud (VPC) のみのアクセスポイントをサポートします。
<code>fs.s3a.committer.name</code>	<code>file</code>	<code>magic</code>	マジックコミッターは、S3 on Outposts でサポートされている唯一のコミッターです。

パラメータ	デフォルト値	S3 on Outposts に必要な値	説明
<code>fs.s3a.select.enabled</code>	TRUE	FALSE	S3 Select は Outposts ではサポートされていません。
<code>JAVA_HOME</code>	<code>/usr/lib/jvm/java-8</code>	<code>/usr/lib/jvm/java-11-amazon-corretto.x86_64</code>	S3A 上の S3 on Outposts には、Java バージョン 11 が必要です。

次の表は、Amazon EMR で S3 on Outposts を使用するクラスターをセットアップするときに、Spark 設定とそのパラメータに指定する値を示しています。

パラメータ	デフォルト値	S3 on Outposts に必要な値	説明
<code>spark.sql.sources.fastS3PartitionDiscovery.enabled</code>	TRUE	FALSE	S3 on Outposts は高速パーティションをサポートしていません。
<code>spark.executorEnv.JAVA_HOME</code>	<code>/usr/lib/jvm/java-8</code>	<code>/usr/lib/jvm/java-11-amazon-corretto.x86_64</code>	S3A 上の S3 on Outposts には、Java バージョン 11 が必要です。

## 考慮事項

Amazon EMR を S3 on Outposts バケットと統合するときは、次の点を考慮してください。

- Amazon S3 on Outposts は、Amazon EMR バージョン 7.0.0 以降でサポートされています。

- Amazon EMR で S3 on Outposts を使用するには、S3A コネクタが必要です。S3 on Outposts バケットを操作するために必要な機能を備えているのは S3A のみです。S3A コネクタのセットアップ情報については、「[前提条件](#)」を参照してください。
- Amazon S3 on Outposts は、Amazon EMR での Amazon S3 マネージドキーによるサーバー側の暗号化 (SSE-S3) のみをサポートします。詳細については、「[the section called “データ暗号化”](#)」を参照してください。
- Amazon S3 on Outposts は、S3A FileOutputCommitter を使用した書き込みをサポートしていません。S3 on Outposts バケットに S3A FileOutputCommitter を使用して書き込むと、次のエラーが発生します。InvalidStorageClass: The storage class you specified is not valid
- Amazon S3 on Outposts は、Amazon EMR Serverless または Amazon EMR on EKS ではサポートされていません。
- Amazon EMR ログは、選択したリージョンの Amazon S3 ロケーションに保存され、S3 on Outposts バケットにローカルに保存されることはありません。

## 認可と認証キャッシング

S3 on Outposts は、Outposts ラック上のローカルで認証および認可データを安全にキャッシュします。キャッシュは、リクエストごとに親 AWS リージョン へのラウンドトリップを削除します。これにより、ネットワークラウンドトリップに伴う変動を排除できます。S3 on Outposts の認証キャッシュと認可キャッシュを使用すると、Outposts と AWS リージョン間の接続のレイテンシーとは無関係の一貫したレイテンシーが得られます。

S3 on Outposts API リクエストを行うと、認証および認可データは安全にキャッシュされます。その後、キャッシュされたデータは、後続の S3 オブジェクト API リクエストを認証するために使用されます。S3 on Outposts は、リクエストが Signature Version 4A (SigV4A) を使用して署名された場合にのみ、認証および認可データをキャッシュします。キャッシュは、S3 on Outposts サービス内の Outposts にローカルに保存されます。S3 API リクエストを行うと、非同期的に更新されます。キャッシュは暗号化され、プレーンテキストの暗号化キーは Outposts に保存されません。

キャッシュは、Outpost が AWS リージョン に接続されてから最大 10 分間有効です。S3 on Outposts API リクエストを行うと、最新のポリシーが使用されるように非同期的に更新されます。Outpost が AWS リージョン から切断されている場合、キャッシュは最大 12 時間有効です。

## 認可キャッシュと認証キャッシングの設定

S3 on Outposts は、SigV4A アルゴリズムで署名されたリクエストの認証および認可データを自動的にキャッシュします。詳細については、「AWS Identity and Access Management ユーザーガイド」

の「[AWS API リクエストの署名](#)」を参照してください。SigV4A アルゴリズムは、最新バージョンの AWS SDK で使用できます。これは、[AWSCommon Runtime \(CRT\) ライブラリ](#)への依存関係を通じて取得できます。

最新バージョンの AWS SDK を使用し、最新バージョンの CRT をインストールする必要があります。例えば、`pip install awscrt` を実行して Boto3 で CRT の最新バージョンを取得できます。

S3 on Outposts は、SigV4 アルゴリズムで署名されたリクエストの認証および認可データをキャッシュしません。

## SigV4A 署名の検証

AWS CloudTrail を使用してリクエストが SigV4A で署名されたことを確認できます。S3 on Outposts の CloudTrail の設定についての詳細は、「[AWS CloudTrail ログで S3 on Outposts をモニタリングする](#)」を参照してください。

CloudTrail を設定したら、CloudTrail ログの `SignatureVersion` フィールドでリクエストがどのように署名されたかを確認できます。SigV4A で署名されたリクエストでは、`SignatureVersion` が `AWS4-ECDSA-P256-SHA256` に設定されます。SigV4 で署名されたリクエストでは、`SignatureVersion` が `AWS4-HMAC-SHA256` に設定されます。

# S3 on Outposts のセキュリティ

AWS でのクラウドセキュリティは最優先事項です。AWS のユーザーは、セキュリティを最も重視する組織の要件を満たすように構築されたデータセンターとネットワークアーキテクチャを利用できます。

セキュリティは AWS とお客様の間の共有責任です。[責任共有モデル](#)では、これを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説明しています。

- **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、お客様が使用するサービスを安全に提供します。[AWS コンプライアンスプログラム](#)の一環として、サードパーティーの監査が定期的にセキュリティの有効性をテストおよび検証しています。Amazon S3 on Outposts に適用するコンプライアンスプログラムの詳細については、「[コンプライアンスプログラムによる対象範囲の AWS 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** — お客様の責任は、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、お客様のデータの機密性、企業の要件、および適用可能な法律および規制などの他の要因についても責任を担います。

このドキュメントは、S3 on Outposts 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、セキュリティとコンプライアンスの目標を満たすように S3 on Outposts を設定する方法について説明します。また、S3 on Outposts リソースのモニタリングと安全確保に役立つその他の AWS のサービスの使用方法も説明します。

## トピック

- [S3 on Outposts で IAM を設定する](#)
- [S3 on Outposts のデータ暗号化](#)
- [S3 on Outposts の AWS PrivateLink](#)
- [AWS Signature Version 4 \(SigV4\) での認証固有ポリシーキー](#)
- [Amazon S3 on Outposts 用の AWS マネージドポリシー](#)
- [Amazon S3 on Outposts でのサービスにリンクされたロールの使用](#)

## S3 on Outposts で IAM を設定する

AWS Identity and Access Management (IAM) は、管理者が AWS リソースへのアクセスを安全に制御するために役立つ AWS のサービスです。IAM 管理者は、誰を認証 (サインイン) し、誰に Amazon S3 on Outposts リソースの使用を承認する (アクセス許可を付与する) かを制御します。IAM は、追加費用なしで使用できる AWS のサービスです。デフォルトでは、ユーザーは S3 on Outposts リソースとオペレーションへのアクセス許可を持っていません。S3 on Outposts リソースと API オペレーションへのアクセス許可を付与するには、IAM を使用して [ユーザー](#)、[グループ](#)、または [ロール](#) を作成し、アクセス許可をアタッチできます。

アクセス権限を付与するにはユーザー、グループ、またはロールにアクセス許可を追加します。

- AWS IAM アイデンティティセンターのユーザーとグループ:

アクセス許可セットを作成します。「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」](#) の手順に従ってください。

- IAM 内で、ID プロバイダーによって管理されているユーザー:

ID フェデレーションのロールを作成します。詳細については 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 ユーザー:

- ユーザーが担当できるロールを作成します。手順については IAM ユーザーガイドの [IAM ユーザーのロールの作成](#) を参照してください。
- (お奨めできない方法) ポリシーをユーザーに直接アタッチするか、ユーザーをユーザーグループに追加します。IAM ユーザーガイドの [ユーザー \(コンソール\) へのアクセス許可の追加](#) の指示に従います。

S3 on Outposts では、IAM ID ベースのポリシーだけでなく、バケットポリシーとアクセスポイントポリシーの両方もサポートしています。バケットポリシーとアクセスポイントポリシーは、S3 on Outposts リソースにアタッチされた、[リソースベースのポリシー](#) です。

- バケットポリシーはバケットにアタッチされ、そのバケットとバケット内のオブジェクトへのリクエストを、ポリシー内の要素に基づいて許可または拒否します。
- 対照的に、アクセスポイントポリシーはアクセスポイントにアタッチされるもので、そのアクセスポイントへのリクエストを許可または拒否します。

アクセスポイントポリシーは、基盤 S3 on Outposts バケットにアタッチされているバケットポリシーと連動します。アプリケーションやユーザーが、S3 on Outposts アクセスポイントを通じて S3 on Outposts にあるオブジェクトにアクセスするためには、アクセスポイントポリシーとバケットポリシーの両方でリクエストを許可している必要があります。

アクセスポイントポリシーに含めた制限は、そのアクセスポイントを介したリクエストにのみ適用されます。例えば、アクセスポイントがバケットにアタッチされている場合、そのバケットに直接送られリクエストを、アクセスポイントポリシーを使用して許可または拒否することはできません。ただし、バケットポリシーで制限を適用することで、バケットに対して直接送られる、またはアクセスポイントを介して送られるリクエストの両方を、許可または拒否することが可能です。

IAM ポリシーまたはリソースベースのポリシーで、どの S3 on Outposts アクションを許可または拒否するかを定義します。S3 on Outposts での各アクションは、S3 on Outposts の特定の API オペレーションに対応しています。S3 on Outposts では、s3-outposts: 名前空間が使用されます。AWS リージョンでの S3 on Outposts コントロール API へのリクエストと、Outpost 上のオブジェクト API エンドポイントへのリクエストに対しては、IAM を使用した認証が行われ、s3-outposts: 名前空間のプレフィックスが承認されます。S3 on Outposts を使用するには、IAM ユーザーを設定し、s3-outposts: IAM 名前空間に対して認可します。

詳細については、「サービス認可リファレンス」の「[S3 on Outposts のアクション、リソース、および条件キー](#)」を参照してください。

#### Note

- S3 on Outposts では、アクセスコントロールリスト (ACL) はサポートされていません。
- S3 on Outposts は、バケットの所有者がオブジェクトにアクセスしたり削除したりができなくなることを防ぐために、デフォルトでバケット所有者をオブジェクト所有者として設定します。
- S3 on Outposts では、オブジェクトがパブリックアクセスできないようにするために、常に S3 ブロックパブリックアクセスが有効になっています。

S3 on Outposts 用の IAM の設定については、以下のトピックを参照してください。

#### トピック

- [S3 on Outposts ポリシーのプリンシパル](#)
- [S3 on Outposts のリソース ARN](#)

- [S3 on Outposts のポリシー例](#)
- [S3 on Outposts エンドポイントの許可](#)
- [S3 on Outposts でのサービスにリンクされたロール](#)

## S3 on Outposts ポリシーのプリンシパル

リソースベースのポリシーを作成して、S3 on Outposts バケットへのアクセスを許可するには、Principal 要素を使用して、そのリソースでのアクションまたはオペレーションに対してリクエストが可能な、ユーザーまたはアプリケーションを指定する必要があります。S3 on Outposts ポリシーには、以下のプリンシパルのいずれかが使用できます。

- AWS アカウント
- IAM ユーザー
- IAM ロール
- アクセスを特定の IP 範囲に制限するために Condition 要素を使用するポリシー内でワイルドカード文字 (\*) を指定することによる、すべてのプリンシパル

### Important

ポリシーに、アクセスを特定の IP アドレス範囲に制限するための Condition を含めない限り、Principal 要素内でワイルドカード文字 (\*) を使用して S3 on Outposts バケットのポリシーを記述することはできません。この制限により、S3 on Outposts バケットへのパブリックアクセスを、確実に防止することができます。例については、[S3 on Outposts のポリシー例](#)を参照してください。

Principal 要素の詳細については、「IAM ユーザーガイド」の「[AWS JSON ポリシーの要素: Principal](#)」を参照してください。

## S3 on Outposts のリソース ARN

S3 on Outposts の Amazon リソースネーム (ARN) には、Outpost が属する AWS リージョン、AWS アカウント ID、およびリソース名に加えて、Outposts ID が含まれます。Outposts バケットとオブジェクトにアクセスしてアクションを実行するには、次の表に示すいずれかの ARN 形式を使用する必要があります。

ARN の *partition* 値は AWS リージョングループのことを指します。それぞれの AWS アカウントは 1 つのパーティションです。サポートされているパーティションは以下のとおりです。

- aws – AWS リージョン
- aws-us-gov - AWS GovCloud (US) リージョン

次の表は、S3 on Outposts ARN 形式を示しています。

Amazon S3 on Outposts の ARN	ARN 形式	例
バケット ARN	arn: <i>partition</i> :s3-outposts: <i>region</i> : <i>account_id</i> :outpost / <i>outpost_id</i> / bucket/ <i>bucket_name</i>	arn:aws:s3-outposts: <i>us-west-2</i> :123456789012 :outpost/ <i>op-01ac5d28a6a232904</i> / bucket/ <i>amzn-s3-demo-bucket1</i>
アクセスポイント ARN	arn: <i>partition</i> :s3-outposts: <i>region</i> : <i>account_id</i> :outpost / <i>outpost_id</i> /accesspoint/ <i>accesspoint_name</i>	arn:aws:s3-outposts: <i>us-west-2</i> :123456789012 :outpost/ <i>op-01ac5d28a6a232904</i> /accesspoint/ <i>access-point-name</i>
オブジェクト ARN	arn: <i>partition</i> :s3-outposts: <i>region</i> : <i>account_id</i> :outpost / <i>outpost_id</i> / bucket/ <i>bucket_name</i> / object/ <i>object_key</i>	arn:aws:s3-outposts: <i>us-west-2</i> :123456789012 :outpost/ <i>op-01ac5d28a6a232904</i> / bucket/ <i>amzn-s3-demo-bucket1</i> /object/ <i>myobject</i>

Amazon S3 on Outposts の ARN	ARN 形式	例
S3 on Outposts アクセスポイント オブジェクトの ARN (ポリシーで使用)	arn: <i>partition</i> :s3-outposts: <i>region</i> : <i>account_id</i> :outpost / <i>outpost_id</i> /accesspoint/ <i>accesspoint_name</i> / object/ <i>object_key</i>	arn: <i>aws</i> :s3-outposts: <i>us-west-2</i> : <i>123456789012</i> :outpost/ <i>op-01ac5d28a6a232904</i> /accesspoint/ <i>access-point-name/object/myobject</i>
S3 on Outposts の ARN	arn: <i>partition</i> :s3-outposts: <i>region</i> : <i>account_id</i> :outpost / <i>outpost_id</i>	arn: <i>aws</i> :s3-outposts: <i>us-west-2</i> : <i>123456789012</i> :outpost/ <i>op-01ac5d28a6a232904</i>

## S3 on Outposts のポリシー例

Example: AWS アカウント のプリンシパルを使用する S3 on Outposts バケットポリシー

次のバケットポリシーは、AWS アカウント プリンシパルを使用して、S3 on Outposts バケットへのアクセスを許可します。このバケットポリシーを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

Example: ワイルドカードプリンシパル (\*) とアクセスを特定の IP アドレス範囲に制限する条件キーを使用する、S3 on Outposts バケットポリシー

次のバケットポリシーは、ワイルドカードプリンシパル (\*) を、アクセスを特定の IP アドレス範囲に制限する 条件とともに使用しています。このバケットポリシーを実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

## S3 on Outposts エンドポイントの許可

S3 on Outposts には、S3 on Outposts のエンドポイントアクションを管理するために、IAM に独自の許可が必要です。


**Note**

- お客様所有の IP アドレスプール (CoIP プール) アクセスタイプを使用するエンドポイントの場合、次の表示に示す通り、CoIP プールから IP アドレスで機能するアクセス権限を保有する必要があります。
- AWS Resource Access Manager を使って S3 on Outposts にアクセスする共有アカウントの場合では、共有アカウントのユーザーは共有サブネット上に独自のエンドポイントを作成できません。共有アカウントのユーザーが独自のエンドポイントを管理する場合、共有アカウントは Outpost に独自のサブネットを作成する必要があります。詳細については、「[the section called “S3 on Outposts の共有”](#)」を参照してください。

次の表は、S3 on Outposts エンドポイントに関連する S3 の IAM 許可を示しています。

Action	IAM アクセス許可
CreateEndpoint	s3-outposts:CreateEndpoint ec2:CreateNetworkInterface ec2:DescribeNetworkInterfaces ec2:DescribeVpcs ec2:DescribeSecurityGroups ec2:DescribeSubnets ec2:CreateTags iam:CreateServiceLinkedRole  オンプレミスのお客様所有の IP アドレスプール (CoIP プール) アクセスタイプを使用しているエンドポイントの場合、次の追加のアクセス許可が必要です。  s3-outposts:CreateEndpoint

Action	IAM アクセス許可
	ec2:DescribeCoipPools ec2:GetCoipPoolUsage ec2:AllocateAddress ec2:AssociateAddress ec2:DescribeAddresses ec2:DescribeLocalGatewayRouteTableVpcAssociations
DeleteEndpoint	s3-outposts:DeleteEndpoint ec2>DeleteNetworkInterface ec2:DescribeNetworkInterfaces オンプレミスのお客様所有の IP アドレスプール (CoIP プール) アクセスタイプを使用しているエンドポイントの場合、次の追加のアクセス許可が必要です。 s3-outposts:DeleteEndpoint ec2:DisassociateAddress ec2:DescribeAddresses ec2:ReleaseAddress
ListEndpoints	s3-outposts:ListEndpoints

 Note

IAM ポリシーでリソースタグを使用すると、アクセス許可を管理できます。

## S3 on Outposts でのサービスにリンクされたロール

S3 on Outposts は IAM サービスにリンクされたロールを使用して、ユーザーに代わっていくつかのネットワークリソースを作成します。詳細については、「[Amazon S3 on Outposts でのサービスにリンクされたロールの使用](#)」を参照してください。

## S3 on Outposts のデータ暗号化

デフォルトでは、Amazon S3 on Outposts に保存されるすべてのデータは、Amazon S3 マネージド暗号化キーによるサーバー側の暗号化 (SSE-S3) を使用して暗号化されます。詳細については、「Amazon S3 ユーザーガイド」の「[Amazon S3 マネージドキーによるサーバー側の暗号化 \(SSE-S3\) の使用](#)」を参照してください。

オプションで、ユーザーが用意した暗号化キーによるサーバー側の暗号化 (SSE-C) を使用できます。SSE-C を使用するには、オブジェクト API リクエストの一部として暗号化キーを指定します。サーバー側の暗号化では、オブジェクトのメタデータではなく、オブジェクトデータのみが暗号化されます。詳細については、「Amazon S3 ユーザーガイド」の「[お客様が指定したキーによるサーバー側の暗号化の使用](#)」をご参照ください。

### Note

S3 on Outposts は、AWS Key Management Service (AWS KMS) キー (SSE-KMS) を使用したサーバー側暗号化をサポートしていません。

## S3 on Outposts の AWS PrivateLink

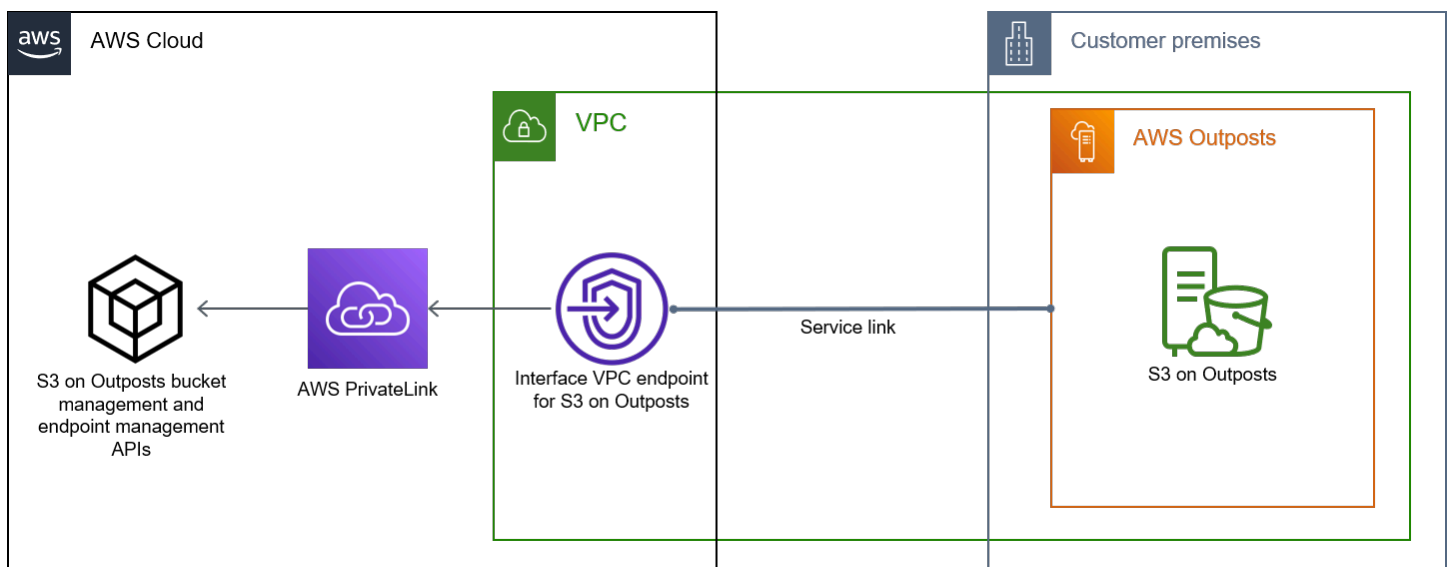
S3 on Outposts は AWS PrivateLink をサポートしているため、仮想プライベートネットワーク内のプライベートエンドポイント経由で S3 on Outposts ストレージに直接管理アクセスを提供します。これにより、仮想プライベートクラウド (VPC) のプライベート IP アドレスを使用して、内部ネットワークアーキテクチャを簡素化し、Outposts オブジェクトストレージに対して管理オペレーションを実行できます。AWS PrivateLink を使用することで、パブリック IP アドレスやプロキシサーバーを使用する必要がなくなります。

Amazon S3 on Outposts の AWS PrivateLink では、インターフェイス VPC エンドポイントを仮想プライベートクラウド (VPC) でプロビジョニングし、S3 on Outposts の[バケット管理](#)および[エンドポイント管理](#) API にアクセスできます。インターフェイス VPC エンドポイントには、VPC またはオンプレミスにデプロイしたアプリケーションから、仮想プライベートネットワーク (VPN)

や AWS Direct Connect 経由で直接アクセスできます。バケット管理 API とエンドポイント管理 API には、AWS PrivateLink を介してアクセスできます。AWS PrivateLink は、[データ転送](#) API オペレーション (GET、PUT、同様の API など) をサポートしていません。これらのオペレーションは、S3 on Outposts エンドポイントとアクセスポイントの設定を通じて既にプライベートに転送されています。詳細については、「[S3 on Outposts のネットワーキング](#)」を参照してください。

インターフェイスエンドポイントは、VPC 内のサブネットからプライベート IP アドレスが割り当てられた 1 つ以上の Elastic Network Interface (ENI) で表されます。S3 on Outposts のインターフェイスエンドポイントに対するリクエストは、AWS ネットワーク上の S3 on Outposts バケット管理およびエンドポイント管理 API に自動的にルーティングされます。AWS Direct Connect または AWS Virtual Private Network (Site-to-Site VPN) を介して、オンプレミスのアプリケーションから VPC 内のインターフェイスエンドポイントにアクセスすることもできます。VPC をオンプレミスネットワークに接続する方法の詳細については、[Direct Connect ユーザーガイド](#)および [AWS Site-to-Site VPN ユーザーガイド](#)を参照してください。

次の図に示すように、インターフェイスエンドポイントは、AWS ネットワークおよび AWS PrivateLink を介してリクエストを S3 on Outposts のバケット管理およびエンドポイント管理 API にルーティングします。



インターフェイスエンドポイントの一般的な情報については、AWS PrivateLink ガイドの[インターフェイス VPC エンドポイント \(AWS PrivateLink\)](#)を参照してください。

## トピック

- [制約と制限](#)
- [S3 on Outposts インターフェイスエンドポイントへのアクセス](#)

- [オンプレミスの DNS 設定の更新](#)
- [S3 on Outposts 用の VPC エンドポイントの作成](#)
- [S3 on Outposts のバケットポリシーと VPC エンドポイントポリシーの作成](#)

## 制約と制限

AWS PrivateLink を介して S3 on Outposts のバケット管理およびエンドポイント管理 API にアクセスする場合は、VPC の制限が適用されます。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のプロパティと制限](#) と [AWS PrivateLink クォータ](#) を参照してください。

また、AWS PrivateLink では、以下の機能はサポートされていません。

- [連邦情報処理規格 \(FIPS\) エンドポイント](#)
- [S3 on Outposts データ転送 API](#)、GET、PUT、および類似のオブジェクト API オペレーションなど。
- プライベート DNS

## S3 on Outposts インターフェイスエンドポイントへのアクセス

AWS PrivateLink を使用して S3 on Outposts バケット管理およびエンドポイント管理 API にアクセスするには、エンドポイント固有の DNS 名を使用するようにアプリケーションを更新する必要があります。インターフェイスエンドポイントを作成すると、AWS PrivateLink は、Amazon S3 はエンドポイント固有の 2 つのタイプの S3 on Outposts 名 (Regional および zonal) を生成します。

- Regional DNS 名 – 一意の VPC エンドポイント ID、サービス識別子、AWS リージョン、および `vpce.amazonaws.com`、`vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com` などが含まれます。
- Zonal DNS 名 – 一意の VPC エンドポイント ID、アベイラビリティーゾーン、サービス識別子、AWS リージョン、および `vpce.amazonaws.com` たとえば `vpce-1a2b3c4d-5e6f-us-east-1a.s3-outposts.us-east-1.vpce.amazonaws.com` が含まれます。このオプションは、アーキテクチャがアベイラビリティーゾーンを分離する場合に使用できます。例えば、障害を隔離し、リージョン間のデータ転送コストを削減するために Zonal DNS 名を使用できます。

### ⚠ Important

S3 on Outposts のインターフェイスエンドポイントは、パブリック DNS ドメインから解決されます。S3 on Outposts はプライベート DNS をサポートしていません。すべてのバケット管理およびエンドポイント管理 API には `--endpoint-url` パラメータを使用します。

## AWS CLI の例

`--region` および `--endpoint-url` パラメータを使用して S3 on Outposts インターフェイスエンドポイントを介してバケット管理およびエンドポイント管理 API にアクセスします。

Example: エンドポイント URL を使用して S3 コントロール API のバケットをリスト化します。

次の例では、リージョン `us-east-1`、VPC エンドポイント URL `vpce-1a2b3c4d-5e6f.s3.us-east-1.vpce.amazonaws.com`、およびアカウント ID `111122223333` を適切な情報に置き換えます。

```
aws s3control list-regional-buckets --region us-east-1 --endpoint-url
https://vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com --account-
id 111122223333
```

## AWS SDK の例

S3 on Outposts インターフェイスエンドポイントの S3 コントロール API にアクセスするように、SDK を最新バージョンに更新して、エンドポイント URL を使用するようにクライアントを設定します。

### SDK for Python (Boto3)

Example: エンドポイント URL を使用して S3 コントロール API にアクセスする

次の例では、リージョン `us-east-1` と VPC エンドポイント URL `vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com` を適切な情報に置き換えます。

```
control_client = session.client(
    service_name='s3control',
    region_name='us-east-1',
```

```
endpoint_url='https://vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com'  
)
```

詳細については、Boto3 デベロッパーガイドの「[Amazon S3 の AWS PrivateLink](#)」を参照してください。

## SDK for Java 2.x

Example: エンドポイント URL を使用して S3 コントロール API にアクセスする

次の例では、VPC エンドポイント URL `vpce-1a2b3c4d-5e6f.s3-outposts.us-east-1.vpce.amazonaws.com` とリージョン `Region.US_EAST_1` を適切な情報に置き換えます。

```
// control client  
Region region = Region.US_EAST_1;  
S3ControlClient = S3ControlClient.builder().region(region)  
  
    .endpointOverride(URI.create("https://vpce-1a2b3c4d-5e6f.s3-outposts.us-  
east-1.vpce.amazonaws.com"))  
    .build()
```

詳細については、「AWS SDK for Java API リファレンス」の「[S3ControlClient](#)」を参照してください。

## オンプレミスの DNS 設定の更新

エンドポイント固有の DNS 名を使用して S3 on Outposts バケット管理およびエンドポイント管理 API にアクセスする場合、オンプレミス DNS リゾルバーを更新する必要はありません。パブリック S3 on Outposts DNS ドメインからのインターフェイスエンドポイントのプライベート IP アドレスを使用して、エンドポイント固有の DNS 名を解決できます。

## S3 on Outposts 用の VPC エンドポイントの作成

S3 on Outposts の VPC インターフェイスエンドポイントを作成するには、「AWS PrivateLink ガイド」の「[VPC エンドポイントの作成](#)」を参照してください。

## S3 on Outposts のバケットポリシーと VPC エンドポイントポリシーの作成

VPC エンドポイントに S3 on Outposts へのアクセスをコントロールするエンドポイントポリシーをアタッチできます。また、S3 on Outposts バケットポリシーの `aws:sourceVpce` 条件を使用して、特定の VPC エンドポイントからの特定のバケットへのアクセスを制限することもできます。VPC エンドポイントポリシーを使用すると、S3 on Outposts バケット管理 API およびエンドポイント管理 API へのアクセスを制御できます。バケットポリシーを使用すると、S3 on Outposts バケット管理 API へのアクセスを制御できます。ただし、`aws:sourceVpce` を使用して S3 on Outposts のオブジェクトアクションへのアクセスを管理することはできません。

S3 on Outposts のアクセスポリシーでは、以下の情報が指定されています。

- アクションを許可または拒否する AWS Identity and Access Management (IAM) プリンシパル。
- 許可または拒否される S3 コントロールアクション。
- アクションが許可または拒否される S3 on Outposts リソース。

次の例は、バケットまたはエンドポイントへのアクセスを制限するポリシーを示しています。VPC 接続の詳細については、[ホワイトペーパー Amazon Virtual Private Cloud 接続オプション AWS のネットワークから VPC への接続オプション](#) を参照してください。

### Important

- このセクションで説明している VPC エンドポイントに例のポリシーを適用すると、バケットへのアクセスが意図せずにブロックされる場合があります。VPC エンドポイントからの接続に対するバケットのアクセスを制限するバケットのアクセス許可により、バケットへのすべての接続がブロックされる場合があります。この問題を修正する方法については、[バケットポリシーの VPC または VPC エンドポイント ID が間違っています。「ポリシーを修正してバケットにアクセスできるようにするにはどうすれば良いですか?」](#) (サポート ナレッジセンター) を参照してください。
- 次のバケットポリシーの例を使用する前に、VPC エンドポイントの ID をユースケースに応じた値に置き換えてください。そうしないと、バケットにアクセスできません。
- ポリシーで、特定の VPC エンドポイントからの S3 on Outposts バケットへのアクセスだけが許可されている場合、コンソールリクエストは指定の VPC エンドポイントから送信されないため、そのバケットのコンソールアクセスは無効になります。

## トピック

- [例: VPC エンドポイントから特定のバケットへのアクセスの制限](#)
- [例:S3 on Outposts バケットポリシーの特定の VPC エンドポイントからのアクセスを拒否する](#)

### 例: VPC エンドポイントから特定のバケットへのアクセスの制限

特定の S3 on Outposts バケットへのアクセスのみを制限するエンドポイントポリシーを作成できます。次のポリシーは、GetBucketPolicy アクションのアクセスを *example-outpost-bucket* のみに制限します。このポリシーを使用するには、例の値を独自の値に置き換えます。

#### JSON

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909151",
  "Statement": [
    {
      "Sid": "Access-to-specific-bucket-only",
      "Principal": {
        "AWS": "111122223333"
      },
      "Action": "s3-outposts:GetBucketPolicy",
      "Effect": "Allow",
      "Resource": "arn:aws:s3-outposts:us-east-1:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket"
    }
  ]
}
```

### 例:S3 on Outposts バケットポリシーの特定の VPC エンドポイントからのアクセスを拒否する

次の S3 on Outposts バケットポリシーは、*vpce-1a2b3c4d* VPC エンドポイントを使用する *example-outpost-bucket* バケットの GetBucketPolicy へのアクセスを拒否します。

aws:sourceVpce 条件はエンドポイントを指定し、VPC エンドポイントリソースの Amazon リソースネーム (ARN) を必要とせず、エンドポイント ID のみを指定します。このポリシーを使用するには、例の値を独自の値に置き換えます。

## JSON

```
{
  "Version": "2012-10-17",
  "Id": "Policy1415115909152",
  "Statement": [
    {
      "Sid": "Deny-access-to-specific-VPCE",
      "Principal": {
        "AWS": "111122223333"
      },
      "Action": "s3-outposts:GetBucketPolicy",
      "Effect": "Deny",
      "Resource": "arn:aws:s3-outposts:us-east-1:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket",
      "Condition": {
        "StringEquals": {
          "aws:sourceVpce": "vpce-1a2b3c4d"
        }
      }
    }
  ]
}
```

## AWS Signature Version 4 (SigV4) での認証固有ポリシーキー

次の表に、Amazon S3 on Outposts で使用する AWS Signature Version 4 (SigV4) 認証に関連する条件キーを示します。これらの条件をバケットポリシーに追加して、Signature Version 4 を使用してリクエストが認証された際の特定の動作を適用できます。エンドポイントポリシーの例については、「[Signature Version 4 関連の条件キーを使用するバケットポリシーの例](#)」を参照してください。Signature Version 4 によるリクエストの認証の詳細については、「Amazon Simple Storage Service API リファレンス」の「[Authenticating requests \(AWS Signature Version 4\)](#)」(リクエストの認証 (Signature Version 4)) を参照してください。

適用できるキー	説明
s3-outposts:authType	S3 on Outposts では、さまざまな認証方式をサポートしています。受信するリクエストが特定の認証方法を使用するように制限するには、このオプションの条件キーを使用します。例えば、この条件キーを使用する

適用できるキー	説明
	<p>と、リクエスト認証のために HTTP Authorization ヘッダーのみの使用を指定できます。</p> <p>有効な値:</p> <p>REST-HEADER</p> <p>REST-QUERY-STRING</p>
s3-outposts:signatureAge	<p>認証されたリクエストの中で署名が有効である時間長 (ミリ秒単位)。</p> <p>この条件は、署名付き URL でのみ機能します。</p> <p>Signature Version 4 では、署名キーは最大 7 日間有効です。したがって、署名の有効期間も最大 7 日間となります。詳細については、「Amazon Simple Storage Service API リファレンス」の「<a href="#">リクエスト署名の導入</a>」を参照してください。この条件を使用すると、署名の有効期間をさらに制限できます。</p> <p>値の例: 600000</p>

適用できるキー	説明
s3-outposts:x-amz-content-sha256	<p>この条件キーを使用すると、バケット内にある署名されていないコンテンツを許可しません。</p> <p>Authorization ヘッダを使用するリクエストに対し Signature Version 4 を使用する場合、署名計算に x-amz-content-sha256 ヘッダーを追加した後、その値をハッシュペイロードに設定します。</p> <p>この条件キーをバケットポリシーで使用すると、ペイロードに署名がないアップロードを、すべて拒否することができます。例:</p> <ul style="list-style-type: none"> <li>リクエスト認証において、Authorization ヘッダーを使用するアップロードを拒否し、ペイロードには署名しません。詳細については、「Amazon Simple Storage Service API リファレンス」の「<a href="#">ペイロードを1つのチャンクで転送する</a>」を参照してください。</li> <li>署名付き URL を使用するアップロードを拒否します。署名済み URL は常に UNSIGNED_PAYLOAD を持ちます。詳細については、「Amazon Simple Storage Service API リファレンス」の「<a href="#">認証リクエスト</a>」および「<a href="#">認証メソッド</a>」を参照してください。</li> </ul> <p>有効な値: UNSIGNED-PAYLOAD</p>

## Signature Version 4 関連の条件キーを使用するバケットポリシーの例

次の例を実行するには、*user input placeholders* をユーザー自身の情報に置き換えます。

Example: s3-outposts:signatureAge

次のバケットポリシーは、署名が作成されてから 10分以上経過している場合、example-outpost-bucket 内のオブジェクトに対する S3 on Outposts の署名付き URL リクエストを拒否します。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

        "Sid": "Deny a presigned URL request if the signature is more than 10
minutes old",
        "Effect": "Deny",
        "Principal": {"AWS": "444455556666"},
        "Action": "s3-outposts:*",
        "Resource": "arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/
object/*",
        "Condition": {
            "NumericGreaterThan": {"s3-outposts:signatureAge": 600000},
            "StringEquals": {"s3-outposts:authType": "REST-QUERY-STRING"}
        }
    }
]
}

```

#### Example: s3-outposts:authType

次のバケットポリシーでは、リクエスト認証に Authorization ヘッダーを使用しているリクエストのみを許可します。署名付き URL リクエストは、クエリパラメーターを使用してリクエストと認証情報を指定しているため、すべての署名付き URL リクエストは拒否されます。詳細については、「Amazon Simple Storage Service API リファレンス」の「[認証メソッド](#)」を参照してください。

#### JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow only requests that use the Authorization header for
request authentication. Deny presigned URL requests.",
      "Effect": "Deny",
      "Principal": {"AWS": "111122223333"},
      "Action": "s3-outposts:*",
      "Resource": "arn:aws:s3-outposts:us-
east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/
object/*",
      "Condition": {
        "StringNotEquals": {
          "s3-outposts:authType": "REST-HEADER"
        }
      }
    }
  ]
}

```

```

    }
  }
]
}

```

Example: s3-outposts:x-amz-content-sha256

次のバケットポリシーは、署名付き URL を使用するアップロードなど、未署名のペイロードを含むすべてのアップロードを拒否します。詳細については、「Amazon Simple Storage Service API リファレンス」の「[Authenticating Requests](#)」(認証リクエスト)「[Authentication methods](#)」(認証メソッド)を参照してください。

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Deny uploads with unsigned payloads.",
      "Effect": "Deny",
      "Principal": {"AWS": "111122223333"},
      "Action": "s3-outposts:*",
      "Resource": "arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/bucket/example-outpost-bucket/object/*",
      "Condition": {
        "StringEquals": {
          "s3-outposts:x-amz-content-sha256": "UNSIGNED-PAYLOAD"
        }
      }
    }
  ]
}

```

## Amazon S3 on Outposts 用の AWS マネージドポリシー

AWS マネージドポリシーは、AWS が作成および管理するスタンドアロンポリシーです。AWS マネージドポリシーは、多くの一般的なユースケースに対してアクセス許可を提供するように設計されているため、ユーザー、グループ、ロールへのアクセス許可の割り当てを開始できます。

AWS マネージドポリシーは、ご利用の特定のユースケースに対して最小特権のアクセス許可を付与しない場合があることにご注意ください。これは、すべての AWS ユーザーが使用できるようになるのを避けるためです。ユースケースに固有の [カスタマー管理ポリシー](#) を定義して、アクセス許可を絞り込むことをお勧めします。

AWS マネージドポリシーで定義されたアクセス許可は変更できません。AWS が AWS マネージドポリシーに定義されているアクセス許可を更新すると、更新はポリシーがアタッチされているすべてのプリンシパルアイデンティティ (ユーザー、グループ、ロール) に影響します。新しい AWS のサービスを起動するか、既存のサービスで新しい API オペレーションが使用可能になると、AWS が AWS マネージドポリシーを更新する可能性が最も高くなります。

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」](#) を参照してください。

## AWS マネージドポリシー: AWSS3OnOutpostsServiceRolePolicy

サービスにリンクされたロール `AWSServiceRoleForS3OnOutposts` の一部として、ネットワークリソースを管理するのに役立ちます。

このポリシーの許可を確認するには、「[AWSS3OnOutpostsServiceRolePolicy](#)」を参照してください。

## AWS マネージドポリシーに対する Amazon S3 on Outposts の更新

S3 on Outposts の AWS マネージドポリシーの更新に関する詳細を、このサービスがこれらの変更の追跡を開始した以降の分について表示します。

変更	説明	日付
S3 on Outposts で <code>AWSS3OnOutpostsServiceRolePolicy</code> を追加	S3 on Outposts では、サービスにリンクされたロール <code>AWSServiceRoleForS3OnOutposts</code> の一部として <code>AWSS3OnOutpostsServiceRolePolicy</code> が追加されました。これは、ネットワークリソースを管理するのに役立ちます。	2023 年 10 月 3 日

変更	説明	日付
S3 on Outposts で変更の追跡を開始	S3 on Outposts で AWS マネージドポリシーの変更の追跡を開始しました。	2023 年 10 月 3 日

## Amazon S3 on Outposts でのサービスにリンクされたロールの使用

Amazon S3 on Outposts 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。サービスにリンクされたロールは、S3 on Outposts 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、S3 on Outposts によって事前定義されており、ユーザーの代わりにサービスから他の AWS のサービスを呼び出す必要のあるアクセス許可がすべて含まれています。

サービスにリンクされたロールを使用することで、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、S3 on Outposts の設定が簡単になります。S3 on Outposts は、サービスにリンクされたロールのアクセス許可を定義します。特に定義されている場合を除き、S3 on Outposts のみがそのロールを引き受けることができます。定義される許可は信頼ポリシーと許可ポリシーに含まれており、その許可ポリシーを他の IAM エンティティにアタッチすることはできません。

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、リソースへのアクセス許可を誤って削除できないようになるため、S3 on Outposts リソースが保護されます。

サービスにリンクされたロールをサポートする他のサービスについては、「[IAM と連動する AWS のサービス](#)」を参照し、Service-linked roles (サービスにリンクされたロール) の列内ではいと表記されたサービスを確認してください。サービスリンクロールに関するドキュメントをサービスで表示するには、リンクで [はい] を選択します。

### S3 on Outposts のサービスにリンクされたロールのアクセス許可

S3 on Outposts は、AWSServiceRoleForS3OnOutposts という名前のロールを使用して、ユーザーに代わってネットワークリソースを管理します。

AWSServiceRoleForS3OnOutposts サービスリンクロールは、以下のサービスを信頼してロールを引き受けます。

- `s3-outposts.amazonaws.com`

AWSS3onOutpostsServiceRolePolicy というロールアクセス許可ポリシーは、S3 on Outposts に、指定されたリソースで次のアクションを完了することを許可します。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcs",
        "ec2:DescribeCoipPools",
        "ec2:GetCoipPoolUsage",
        "ec2:DescribeAddresses",
        "ec2:DescribeLocalGatewayRouteTableVpcAssociations"
      ],
      "Resource": "*",
      "Sid": "DescribeVpcResources"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:security-group/*"
      ],
      "Sid": "CreateNetworkInterface"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:network-interface/*"
      ]
    }
  ]
}
```

```
    ],
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/CreatedBy": "S3 On Outposts"
      }
    },
    "Sid": "CreateTagsForCreateNetworkInterface"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:AllocateAddress"
    ],
    "Resource": [
      "arn:aws:ec2:*:*:ipv4pool-ec2/*"
    ],
    "Sid": "AllocateIpAddress"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:AllocateAddress"
    ],
    "Resource": [
      "arn:aws:ec2:*:*:elastic-ip/*"
    ],
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/CreatedBy": "S3 On Outposts"
      }
    },
    "Sid": "CreateTagsForAllocateIpAddress"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:ModifyNetworkInterfaceAttribute",
      "ec2:CreateNetworkInterfacePermission",
      "ec2>DeleteNetworkInterface",
      "ec2>DeleteNetworkInterfacePermission",
      "ec2:DisassociateAddress",
      "ec2:ReleaseAddress",
      "ec2:AssociateAddress"
    ]
  },
]
```

```
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/CreatedBy": "S3 On Outposts"
      }
    },
    "Sid": "ReleaseVpcResources"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ec2:CreateTags"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "ec2:CreateAction": [
          "CreateNetworkInterface",
          "AllocateAddress"
        ],
        "aws:RequestTag/CreatedBy": [
          "S3 On Outposts"
        ]
      }
    },
    "Sid": "CreateTags"
  }
]
```

サービスにリンクされたロールの作成、編集、削除を IAM エンティティ (ロールなど) に許可するには、アクセス許可を設定する必要があります。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役割のアクセス許可](#)」を参照してください。

## S3 on Outposts で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。AWS マネジメントコンソール、AWS CLI、または AWS API で S3 on Outposts エンドポイントを作成すると、S3 on Outposts によってサービスにリンクされたロールが作成されます。

このサービスリンクロールを削除した後で再度作成する必要がある場合は同じ方法でアカウントにロールを再作成できます。S3 on Outposts エンドポイントを作成すると、S3 on Outposts によってサービスにリンクされたロールが再び作成されます。

IAM コンソールを使用して、S3 on Outposts ユースケースでサービスにリンクされたロールを作成することもできます。AWS CLI または AWS API では、`s3-outposts.amazonaws.com` サービス名を使用してサービスリンクロールを作成します。詳細については、IAM ユーザーガイドの「[サービスリンクロールの作成](#)」を参照してください。このサービスリンクロールを削除しても、同じ方法でロールを再作成できます。

## S3 on Outposts でのサービスにリンクされたロールの編集

S3 on Outposts では、サービスにリンクされたロール `AWSServiceRoleForS3onOutposts` を編集することはできません。さまざまなエンティティが参照する可能性があるため、これにはロール名が含まれます。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「[サービスリンクロールの編集](#)」を参照してください。

## S3 on Outposts で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を必要とする機能やサービスが不要になった場合は、ロールを削除することをお勧めします。そうすることで、積極的にモニタリングまたは保守さ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、手動で削除する前に、サービスリンクロールのリソースをクリーンアップする必要があります。

### Note

リソースを削除する際に、S3 on Outposts サービスでそのロールが使用されている場合、削除は失敗することがあります。失敗した場合は数分待ってから操作を再試行してください。

`AWSServiceRoleForS3OnOutposts` が使用する S3 on Outposts リソースを削除するには

1. すべての AWS リージョンの AWS アカウントで、[Outposts エンドポイントの S3 を削除します](#)。
2. IAM を使用して、サービスにリンクされたロールを削除します。

IAM コンソール、AWS CLI、または AWS API を使用して、`AWSServiceRoleForS3onOutposts` サービスリンク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。

## S3 on Outposts で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

S3 on Outposts は、そのサービスを利用できるすべての AWS 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。詳細については、「[S3 on Outposts のリージョンとエンドポイント](#)」を参照してください。

# S3 on Outposts ストレージの管理

Amazon S3 on Outposts を使用すると、AWS Outposts で S3 バケットを作成し、ローカルデータアクセス、ローカルデータ処理、データレジデンシーを必要とするアプリケーション用に、オンプレミスのオブジェクトを簡単に保存および取得できます。S3 on Outposts は、新しいストレージクラス、S3 Outposts (OUTPOSTS) を提供し、これは Amazon S3 API を使用し、AWS Outposts 上の複数のデバイスやサーバー間でデータを永続的かつ冗長的に保存するように設計されています。仮想プライベートクラウド (VPC) を介したアクセスポイントとエンドポイント接続を使用して、Outposts バケットと通信します。Outposts バケットでは、Amazon S3 と同じ API と機能 (アクセスポリシー、暗号化、タグ付けなど) を使用できます。AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDK、または REST API を使用して S3 on Outposts を使用できます。詳細については、[Amazon S3 on Outposts とは](#) を参照してください。

Amazon S3 on Outposts ストレージ容量の管理および共有の詳細については、以下のトピックを参照してください。

## トピック

- [S3 on Outposts バケットの S3 バージョニングの管理](#)
- [Amazon S3 on Outposts バケットのライフサイクル設定を作成および管理する](#)
- [S3 on Outposts のオブジェクトのレプリケート](#)
- [AWS RAM を使用して S3 on Outposts を共有する](#)
- [S3 on Outposts を使用するその他の AWS のサービス](#)

## S3 on Outposts バケットの S3 バージョニングの管理

有効にすると、S3 バージョニングは、同じバケット内にあるオブジェクトの複数の異なるコピーを保存します。S3 バージョニングを使用すると、Outposts 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、復元できます。S3 バージョニングによって、意図しないユーザーアクションやアプリケーション障害から復旧できます。

Amazon S3 on Outposts バケットには、次の 3 つのバージョニング状態があります。

- バージョニングが無効 — バケットの S3 バージョニングを有効化または停止したことがない場合は、バージョニングが無効であり、S3 バージョニングの状態は返されません。S3 バージョニングの詳細については、[S3 on Outposts バケットの S3 バージョニングの管理](#) を参照してください。

- 有効化 — バケット内のオブジェクトの S3 バージョニングを有効にします。バケットに追加されたすべてのオブジェクトは、一意のバージョン ID を受け取ります。バージョニングを有効にした時点でバケットにすでに存在していたオブジェクトのバージョン ID は null です。これらの (またはその他の) オブジェクトを他のオペレーション ([PutObject](#)) で変更すると、新しいオブジェクトは一意のバージョン ID を取得します。
- 停止 — バケット内のオブジェクトの S3 バージョニングを停止します。バージョニングが停止された後に、バケットに追加されたすべてのオブジェクトは、バージョン ID null を受け取ります。詳細については、「Amazon S3 ユーザーガイド」の「[バージョニングが停止されたバケットへのオブジェクトの追加](#)」を参照してください。

S3 on Outposts バケットの S3 バージョニングを有効にすると、バージョニング無効の状態に戻すことはできません。ただし、バージョニングを停止することはできます。S3 バージョニングの詳細については、[S3 on Outposts バケットの S3 バージョニングの管理](#) を参照してください。

バケット内の各オブジェクトには、最新のバージョンと最新でないバージョンが 0 個以上存在します。ストレージコストを削減するには、バケットの S3 ライフサイクルルールを設定して、指定した期間の後に最新でないバージョンを期限切れにすることができます。詳細については、「[Amazon S3 on Outposts バケットのライフサイクル設定を作成および管理する](#)」を参照してください。

次の例は、AWS マネジメントコンソールと AWS Command Line Interface (AWS CLI) を使用して、既存の S3 on Outposts バケットのバージョニングを有効化または停止する方法を示しています。S3 バージョニングを有効にしてバケットを作成するには、「[S3 on Outposts バケットを作成する](#)」を参照してください。

#### Note

バケットを作成する AWS アカウント がそのバケットを所有し、アクションをコミットできる唯一のアカウントです。バケットには、Outposts、タグ、デフォルトの暗号化、アクセスポイント設定などの設定プロパティがあります。アクセスポイント設定には、仮想プライベートクラウド (VPC)、バケット内のオブジェクトにアクセスするためのアクセスポイントポリシー、およびその他のメタデータが含まれます。詳細については、「[S3 on Outposts の仕様](#)」を参照してください。

## S3 コンソールの使用

バケットの S3 バージョニング設定を編集するには

1. AWS マネジメントコンソールにサインインし、Amazon S3 コンソール <https://console.aws.amazon.com/s3/> を開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. S3 バージョニングを有効にする Outposts バケットを選択します。
4. [プロパティ] タブを選択します。
5. [バケットのバージョニング] で [編集] を選択します。
6. 以下のいずれかのオプションを選択して、バケットの S3 バージョニング設定を編集します。
  - S3 バージョニングを一時停止し、新しいオブジェクトバージョンの作成を停止するには、[Suspend] (停止) を選択します。
  - S3 バージョニングを有効にして、各オブジェクトの複数の異なるコピーを保存するには、[Enable] (有効化) を選択します。
7. [Save changes] (変更の保存) をクリックします。

## の使用AWS CLI

AWS CLI を使用してバケットの S3 バージョニングを有効化または停止するには、次の例に示すように `put-bucket-versioning` コマンドを使用します。これらの例を実行するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。

詳細については、「AWS CLI リファレンス」の「[put-bucket-versioning](#)」を参照してください。

Example: S3 バージョニングを有効にするには

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket --versioning-configuration Status=Enabled
```

Example: S3 バージョニングを停止するには

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/bucket/example-outposts-bucket --versioning-configuration Status=Suspended
```

# Amazon S3 on Outposts バケットのライフサイクル設定を作成および管理する

S3 ライフサイクルを使用して、Amazon S3 on Outposts のストレージ容量を最適化することができます。ライフサイクルルールを作成して、オブジェクトが古くなったり、新しいバージョンに置き換えられたりしたときに期限切れにすることができます。ライフサイクルルールを作成、有効化、無効化、または削除できます。

S3 ライフサイクルの詳細については、「[Amazon S3 on Outposts バケットのライフサイクル設定を作成および管理する](#)」を参照してください。

## Note

バケットを作成する AWS アカウント がそのバケットを所有し、ライフサイクルルールを作成、有効化、無効化、または削除できる唯一のアカウントです。

S3 on Outposts バケットのライフサイクル設定を作成および管理するには、以下のトピックを参照してください。

## トピック

- [AWS マネジメントコンソール を使用したライフサイクルルールの作成と管理](#)
- [AWS CLI および SDK for Java を使用したライフサイクル設定の作成と管理](#)

## AWS マネジメントコンソール を使用したライフサイクルルールの作成と管理

S3 ライフサイクルを使用して、Amazon S3 on Outposts のストレージ容量を最適化することができます。ライフサイクルルールを作成して、オブジェクトが古くなったり、新しいバージョンに置き換えられたりしたときに期限切れにすることができます。ライフサイクルルールを作成、有効化、無効化、または削除できます。

S3 ライフサイクルの詳細については、「[Amazon S3 on Outposts バケットのライフサイクル設定を作成および管理する](#)」を参照してください。

**Note**

バケットを作成する AWS アカウント がそのバケットを所有し、ライフサイクルルールを作成、有効化、無効化、または削除できる唯一のアカウントです。

AWS マネジメントコンソール を使用して S3 on Outposts のライフサイクルルールを作成および管理するには、以下のトピックを参照してください。

## トピック

- [ライフサイクルルールの作成](#)
- [ライフサイクルルールの有効化](#)
- [ライフサイクルルールの編集](#)
- [ライフサイクルルールの削除](#)

## ライフサイクルルールの作成

1. AWS マネジメントコンソール にサインインし、Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. ライフサイクルルールを作成する Outposts バケットを選択します。
4. [Management] (管理) タブを選択して、[Create Lifecycle rule] (ライフサイクルルールを作成する) を選択します。
5. [Lifecycle rule name] (ライフサイクルルール名) の値を入力します。
6. [Rule scope] (ルールスコープ) で、次のいずれかのオプションを選択します。
  - 特定のフィルターの適用範囲を制限するには、[Limit the scope of this rule using one or more filters] (1 つまたは複数のフィルターを使用してこのルールの適用範囲を制限) を選択します。次に、プレフィックスフィルターまたはオブジェクトサイズを追加します。
  - このルールをバケット内のすべてのオブジェクトに適用するには、[Apply to all objects in the bucket] (バケット内のすべてのオブジェクトに適用) を選択します。
7. [Lifecycle rule actions] (ライフサイクルルールのアクション) で、次のいずれかのオプションを選択します。

- [Expire current versions of objects] (オブジェクトの現行バージョンを期限切れにする) — バージョニングが有効なバケットの場合、S3 on Outposts は削除マーカを追加し、オブジェクトを最新でないバージョンとして保持します。S3 バージョニングを使用しないバケットの場合、S3 on Outposts はオブジェクトを完全に削除します。
- [Permanently delete noncurrent versions of objects] (オブジェクトの最新ではないバージョンを完全に削除する) — S3 on Outposts は、オブジェクトの最新ではないバージョンを完全に削除します。
- [Delete expired object delete markers or incomplete multipart uploads] (期限切れのオブジェクト削除マーカまたは未完了のマルチパートアップロードを削除する) — S3 on Outposts は、期限切れのオブジェクト削除マーカまたは未完了のマルチパートアップロードを完全に削除します。

オブジェクトタグを使用してライフサイクルルールの範囲を制限する場合、[Delete expired object delete markers] (期限切れのオブジェクト削除マーカを削除する) を選択することはできません。また、[Expire current object versions] (オブジェクトの現行バージョンを期限切れにする) を選択した場合は、[Delete expired object delete markers] (期限切れのオブジェクト削除マーカを削除する) を選択することもできません。

#### Note

サイズベースのフィルターは、削除マーカや未完了のマルチパートアップロードには使用できません。

8. [Expire current versions of objects] (オブジェクトの現行バージョンを期限切れにする) または [Permanently delete noncurrent versions of objects] (オブジェクトの最新ではないバージョンを完全に削除する) を選択した場合は、特定の日付またはオブジェクトの経過時間に基づいてルールトリガーを設定します。
9. [Delete expired object delete markers] (期限切れのオブジェクト削除マーカを削除する) を選択した場合、期限切れのオブジェクト削除マーカの削除を確認するには、[Delete expired object delete markers] (期限切れのオブジェクト削除マーカを削除する) を選択します。
10. [Timeline Summary] (タイムラインの概要) で、ライフサイクルルールを確認し、[Create rule] (ルールの作成) を選択します。

## ライフサイクルルールの有効化

バケットのライフサイクルルールを有効または無効にするには

1. Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. ライフサイクルルールを有効または無効にする Outposts バケットを選択します。
4. [Management] (管理) タブを選択し、[Lifecycle rule] (ライフサイクルルール) で、有効または無効にするルールを選択します。
5. [Action] (アクション) で、[Enable or disable rule] (ルールを有効化または無効化) を選択します。

## ライフサイクルルールの編集

1. Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. ライフサイクルルールを編集する Outposts バケットを選択します。
4. [Management] (管理) タブを選択し、編集するライフサイクルルールを選択します。
5. (オプション) [Lifecycle rule name] (ライフサイクルルール名) の値を更新します。
6. [Rule scope] (ルールスコープ) で、必要に応じてスコープを編集します。
  - 特定のフィルターの適用範囲を制限するには、[Limit the scope of this rule using one or more filters] (1 つまたは複数のフィルターを使用してこのルールの適用範囲を制限) を選択します。次に、プレフィックスフィルターまたはオブジェクトサイズを追加します。
  - このルールをバケット内のすべてのオブジェクトに適用するには、[Apply to all objects in the bucket] (バケット内のすべてのオブジェクトに適用) を選択します。
7. [Lifecycle rule actions] (ライフサイクルルールのアクション) で、次のいずれかのオプションを選択します。
  - [Expire current versions of objects] (オブジェクトの現行バージョンを期限切れにする) — バージョニングが有効なバケットの場合、S3 on Outposts は削除マーカーを追加し、オブジェクトを最新でないバージョンとして保持します。S3 バージョニングを使用しないバケットの場合、S3 on Outposts はオブジェクトを完全に削除します。
  - [Permanently delete noncurrent versions of objects] (オブジェクトの最新ではないバージョンを完全に削除する) — S3 on Outposts は、オブジェクトの最新ではないバージョンを完全に削除します。

- [Delete expired object delete markers or incomplete multipart uploads] (期限切れのオブジェクト削除マーカ―または未完了のマルチパートアップロードを削除する) — S3 on Outposts は、期限切れのオブジェクト削除マーカ―または未完了のマルチパートアップロードを完全に削除します。

オブジェクトタグを使用してライフサイクルルールの範囲を制限する場合、[Delete expired object delete markers] (期限切れのオブジェクト削除マーカ―を削除する) を選択することはできません。また、[Expire current object versions] (オブジェクトの現行バージョンを期限切れにする) を選択した場合は、[Delete expired object delete markers] (期限切れのオブジェクト削除マーカ―を削除する) を選択することもできません。

#### Note

サイズベースのフィルターは、削除マーカ―や未完了のマルチパートアップロードには使用できません。

8. [Expire current versions of objects] (オブジェクトの現行バージョンを期限切れにする) または [Permanently delete noncurrent versions of objects] (オブジェクトの最新ではないバージョンを完全に削除する) を選択した場合は、特定の日付またはオブジェクトの経過時間に基づいてルールトリガーを設定します。
9. [Delete expired object delete markers] (期限切れのオブジェクト削除マーカ―を削除する) を選択した場合、期限切れのオブジェクト削除マーカ―の削除を確認するには、[Delete expired object delete markers] (期限切れのオブジェクト削除マーカ―を削除する) を選択します。
10. [保存] を選択します。

## ライフサイクルルールの削除

1. Amazon S3 コンソール (<https://console.aws.amazon.com/s3/>) を開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. ライフサイクルルールを削除する Outposts バケットを選択します。
4. [Management] (管理) タブを選択し、[Lifecycle rule] (ライフサイクルルール) で、削除するルールを選択します。
5. [Delete] (削除) をクリックします。

# AWS CLI および SDK for Java を使用したライフサイクル設定の作成と管理

S3 ライフサイクルを使用して、Amazon S3 on Outposts のストレージ容量を最適化することができます。ライフサイクルルールを作成して、オブジェクトが古くなったり、新しいバージョンに置き換えられたりしたときに期限切れにすることができます。ライフサイクルルールを作成、有効化、無効化、または削除できます。

S3 ライフサイクルの詳細については、「[Amazon S3 on Outposts バケットのライフサイクル設定を作成および管理する](#)」を参照してください。

## Note

バケットを作成する AWS アカウント がそのバケットを所有し、ライフサイクルルールを作成、有効化、無効化、または削除できる唯一のアカウントです。

AWS Command Line Interface (AWS CLI) と AWS SDK for Java を使用して S3 on Outposts バケツ用のライフサイクル設定を作成および管理するには、以下の例を参照してください。

## トピック

- [ライフサイクル設定を PUT する](#)
- [S3 on Outposts バケットのライフサイクル設定を GET する](#)

## ライフサイクル設定を PUT する

### AWS CLI

次の AWS CLI の例では、Outposts バケツにライフサイクル設定ポリシーを配置します。このポリシーでは、フラグ付きのプレフィックス (*myprefix*) とタグを持つすべてのオブジェクトが 10 日後に期限切れになることを指定します。この例を実行するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。

1. ライフサイクル設定ポリシーを JSON ファイルに保存します。この例では、ファイル名は `lifecycle1.json` です。

```
{
  "Rules": [
```

```
{
  "ID": "id-1",
  "Filter": {
    "And": {
      "Prefix": "myprefix",
      "Tags": [
        {
          "Value": "mytagvalue1",
          "Key": "mytagkey1"
        },
        {
          "Value": "mytagvalue2",
          "Key": "mytagkey2"
        }
      ],
      "ObjectSizeGreaterThan": 1000,
      "ObjectSizeLessThan": 5000
    }
  },
  "Status": "Enabled",
  "Expiration": {
    "Days": 10
  }
}
]
```

2. `put-bucket-lifecycle-configuration` CLI コマンドの一部として JSON ファイルを送信します。このコマンドを使用するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。このコマンドの詳細については、「AWS CLI リファレンス」の「[put-bucket-lifecycle-configuration](#)」を参照してください。

```
aws s3control put-bucket-lifecycle-configuration --account-id 123456789012 --
bucket arn:aws:s3-outposts:region:123456789012:outpost/op-01ac5d28a6a232904/
bucket/example-outposts-bucket --lifecycle-configuration file://lifecycle1.json
```

## SDK for Java

次の SDK for Java の例では、Outposts バケットにライフサイクル設定を配置します。このライフサイクル設定では、フラグ付きのプレフィックス (*myprefix*) とタグを持つすべてのオブジェクトが 10 日後に期限切れになることを指定します。この例を実行するには、それぞれの *user*

*input placeholder* をユーザー自身の情報に置き換えます。詳細については、「Amazon S3 API リファレンス」の「[PutBucketLifecycleConfiguration](#)」を参照してください。

```
import com.amazonaws.services.s3control.model.*;

public void putBucketLifecycleConfiguration(String bucketArn) {

    S3Tag tag1 = new S3Tag().withKey("mytagkey1").withValue("mytagkey1");
    S3Tag tag2 = new S3Tag().withKey("mytagkey2").withValue("mytagkey2");

    LifecycleRuleFilter lifecycleRuleFilter = new LifecycleRuleFilter()
        .withAnd(new LifecycleRuleAndOperator()
            .withPrefix("myprefix")
            .withTags(tag1, tag2))
            .withObjectSizeGreaterThan(1000)
            .withObjectSizeLessThan(5000);

    LifecycleExpiration lifecycleExpiration = new LifecycleExpiration()
        .withExpiredObjectDeleteMarker(false)
        .withDays(10);

    LifecycleRule lifecycleRule = new LifecycleRule()
        .withStatus("Enabled")
        .withFilter(lifecycleRuleFilter)
        .withExpiration(lifecycleExpiration)
        .withID("id-1");

    LifecycleConfiguration lifecycleConfiguration = new LifecycleConfiguration()
        .withRules(lifecycleRule);

    PutBucketLifecycleConfigurationRequest reqPutBucketLifecycle = new
    PutBucketLifecycleConfigurationRequest()
        .withAccountId(AccountId)
        .withBucket(bucketArn)
        .withLifecycleConfiguration(lifecycleConfiguration);

    PutBucketLifecycleConfigurationResult respPutBucketLifecycle =
    s3ControlClient.putBucketLifecycleConfiguration(reqPutBucketLifecycle);
    System.out.printf("PutBucketLifecycleConfiguration Response: %s\n",
    respPutBucketLifecycle.toString());
}
```

## S3 on Outposts バケットのライフサイクル設定を GET する

### AWS CLI

次の AWS CLI の例では、Outposts バケットにライフサイクル設定を取得します。このコマンドを使用するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。このコマンドの詳細については、「AWS CLI リファレンス」の「[put-bucket-lifecycle-configuration](#)」を参照してください。

```
aws s3control get-bucket-lifecycle-configuration --account-id 123456789012 --bucket
arn:aws:s3-outposts:<your-region>:123456789012:outpost/op-01ac5d28a6a232904/
bucket/example-outposts-bucket
```

### SDK for Java

次の SDK for Java の例では、Outposts バケットにライフサイクル設定を取得します。詳細については、「Amazon Simple Storage Service API リファレンス」の「[GetBucketLifecycleConfiguration](#)」を参照してください。

```
import com.amazonaws.services.s3control.model.*;

public void getBucketLifecycleConfiguration(String bucketArn) {

    GetBucketLifecycleConfigurationRequest reqGetBucketLifecycle = new
    GetBucketLifecycleConfigurationRequest()
        .withAccountId(AccountId)
        .withBucket(bucketArn);

    GetBucketLifecycleConfigurationResult respGetBucketLifecycle =
    s3ControlClient.getBucketLifecycleConfiguration(reqGetBucketLifecycle);
    System.out.printf("GetBucketLifecycleConfiguration Response: %s%n",
    respGetBucketLifecycle.toString());

}
```

## S3 on Outposts のオブジェクトのレプリケート

AWS Outposts で S3 レプリケーションを有効にすると、Amazon S3 on Outposts を設定して、S3 オブジェクトを異なる Outposts 間、または同じ Outpost のバケット間で自動的にレプリケートできます。S3 Replication on Outposts を使用すると、同じまたは異なる Outposts、または異なるア

カウント間でデータの複数のレプリカを管理できるため、データの常駐要件を満たすことができます。S3 Replication on Outposts は、準拠したストレージのニーズとアカウント間のデータ共有を強化するのに役立ちます。レプリカがソースデータと同一であることを確認する必要がある場合は、S3 Replication on Outposts を使用して、元のオブジェクトの作成時間、タグ、バージョン ID などのすべてのメタデータを保持するオブジェクトのレプリカを作成できます。

S3 Replication on Outposts では、バケット間のオブジェクトレプリケーションのステータスをモニタリングするための詳細なメトリクスと通知も提供されます。Amazon CloudWatch を使用して、レプリケーション保留中のバイト数、レプリケーション保留中のオペレーション、およびソースバケットとターゲットバケット間のレプリケーションレイテンシーを追跡することで、レプリケーションの進行状況を監視できます。設定の問題をすばやく診断して修正するには、レプリケーションオブジェクトの障害に関する通知を受け取るように Amazon EventBridge を設定することもできます。詳細については[レプリケーションの管理](#)を参照してください。

## トピック

- [レプリケーション設定](#)
- [S3 Replication on Outposts の要件](#)
- [レプリケーションの対象](#)
- [レプリケーションの対象外](#)
- [S3 Replication on Outposts でサポートされていないものは何ですか？](#)
- [レプリケーションの設定](#)
- [レプリケーションの管理](#)

## レプリケーション設定

S3 on Outposts はレプリケーション設定を XML 形式で保存します。レプリケーション設定 XML ファイルで、AWS Identity and Access Management (IAM) ロールと 1 つ以上のルールを指定します。

```
<ReplicationConfiguration>
  <Role>IAM-role-ARN</Role>
  <Rule>
    ...
  </Rule>
  <Rule>
    ...
</ReplicationConfiguration>
```

```
</Rule>
...
</ReplicationConfiguration>
```

S3 on Outposts はユーザーの許可なしにオブジェクトをレプリケートすることはできません。レプリケーション設定で指定した IAM ロールを使用して S3 on Outposts アクセス許可を付与します。S3 on Outposts は、ユーザーに代わってオブジェクトをレプリケートするための IAM ロールを引き受けます。レプリケーションを開始する前に、必要なアクセス許可を IAM ロールに付与する必要があります。S3 on Outposts に対するこれらのアクセス許可の詳細については、「[IAM ロールの作成](#)」を参照してください。

次のシナリオでは、レプリケーション設定にルールを 1 つ追加します。

- すべてのオブジェクトをレプリケートします。
- オブジェクトのサブセットをレプリケートします。ルールにフィルターを追加して、オブジェクトのサブセットを特定します。フィルターでは、ルールが適用されるオブジェクトのサブセットを特定するために、オブジェクトキープレフィックス、タグ、またはその両方の組み合わせを指定します。

オブジェクトの異なるサブセットをレプリケートする場合は、レプリケーション設定に複数のルールを追加します。各ルールでは、オブジェクトの異なるサブセットを選択するフィルターを指定します。例えば、tax/ または document/ のいずれかのキープレフィックスを持つオブジェクトをレプリケートするとします。このためには、tax/ キープレフィックスフィルターを指定するルールと、document/ キープレフィックスを指定するもう 1 つのルールの 2 つのルールを追加します。

S3 on Outposts のレプリケーション設定とレプリケーションルールの詳細については、Amazon Simple Storage Service API リファレンスの「[ReplicationConfiguration](#)」を参照してください。

## S3 Replication on Outposts の要件

レプリケーションには以下が必要です。

- 送信先の Outpost CIDR 範囲は、ソースの Outpost サブネットテーブルに関連付けられている必要があります。詳細については、「[レプリケーションルールの作成の前提条件](#)」を参照してください。
- レプリケート元とレプリケート先の両方のバケットで、S3 バージョニングを有効にする必要があります。バージョニングの詳細については、「[S3 on Outposts バケットの S3 バージョニングの管理](#)」を参照してください。

- Amazon S3 on Outposts には、ユーザーに代わってレプリケート元バケットのオブジェクトをレプリケート先バケットにレプリケートするアクセス許可が必要です。つまり、GET および PUT のアクセス許可を S3 on Outposts に委任するには、サービスロールを作成する必要があります。
  1. サービスロールを作成する前に、レプリケート元バケットに対する GET アクセス許可とレプリケート先バケットに対する PUT アクセス許可が必要です。
  2. S3 on Outposts にアクセス許可を委任するサービスロールを作成するには、まず IAM エンティティ (ユーザーまたはロール) が `iam:CreateRole` および `iam:PassRole` アクションを実行できるようにアクセス許可を設定する必要があります。次に、IAM エンティティがサービスロールを作成できるようにします。S3 on Outposts がユーザーに代わってサービスロールを引き受け、GET および PUT のアクセス許可を S3 on Outposts に委任するには、必要な信頼とアクセス許可のポリシーをロールに割り当てる必要があります。S3 on Outposts に対するこれらのアクセス許可の詳細については、「[IAM ロールの作成](#)」を参照してください。サービスロールの作成の詳細については、「[サービスのロールの作成](#)」を参照してください。

## レプリケーションの対象

デフォルトで、S3 on Outposts は以下をレプリケートします。

- レプリケーション設定の追加後に作成されたオブジェクト。
- レプリケート元オブジェクトからレプリカへのオブジェクトメタデータ。レプリカからレプリケート元オブジェクトへのメタデータのレプリケーションについては、「[Outposts の Amazon S3 レプリカ変更の同期が有効になっている場合のレプリケーションステータス](#)」を参照してください。
- オブジェクトタグ、存在する場合。

## 削除オペレーションがレプリケーションに与える影響

レプリケート元バケットからオブジェクトを削除すると、デフォルトで次のアクションが実行されます。

- オブジェクトバージョン ID を指定せずに DELETE リクエストを行った場合、S3 on Outposts は削除マーカールールを追加します。S3 on Outposts では、削除マーカールールを次のように扱います。
  - S3 on Outposts は、デフォルトでは削除マーカールールをレプリケートしません。
  - ただし、タグベース以外のルールには削除マーカールールレプリケーションを追加できます。レプリケーション設定で削除マーカールールレプリケーションを有効にする場合は、「[S3 コンソールの使用](#)」を参照してください。

- DELETE リクエストで削除するオブジェクトバージョン ID を指定した場合、S3 on Outposts はレプリケート元バケット内のそのオブジェクトバージョンを完全に削除します。しかし、レプリケート先バケット内でその削除をレプリケートすることはありません。これは、レプリケート先バケットからは、同じオブジェクトバージョンを削除しないことを意味します。これは悪意のある削除からデータを保護します。

## レプリケーションの対象外

デフォルトで、S3 on Outposts は以下をレプリケートしません。

- 別のレプリケーションルールによって作成されたレプリカである、レプリケート元バケットのオブジェクト。例えば、バケット A がレプリケート元でバケット B がレプリケート先であるレプリケーションを設定するとします。ここで、バケット B をレプリケート元、バケット C をレプリケート先とする別のレプリケーション設定を追加したとします。この場合、バケット A のオブジェクトのレプリカであるバケット B のオブジェクトは、バケット C にレプリケートされません。
- 既に別のレプリケート先にレプリケートされている、レプリケート元バケット内のオブジェクト。例えば、既存のレプリケーション設定でレプリケート先バケットを変更した場合、S3 on Outposts がそのオブジェクトを再度レプリケートすることはありません。
- お客様が用意した暗号化キー (SSE-C) を使用し、サーバー側の暗号化で作成されたオブジェクト。
- バケットレベルのサブリソースの更新

たとえば、ライフサイクル設定を変更したり、レプリケート元バケットに通知設定を追加した場合、これらの変更はレプリケート先バケットには適用されません。この機能により、レプリケート元バケットとレプリケート先バケットで異なる設定を指定できます。

- ライフサイクル設定によって実行されたアクション。

例えば、ライフサイクル設定がレプリケート元バケットでのみ有効で、有効期限アクションを設定した場合、S3 on Outposts はレプリケート元バケットの有効期限切れになったオブジェクトの削除マーカーを作成しますが、それらのマーカーはレプリケート先バケットにレプリケートされません。同じライフサイクル設定をレプリケート元バケットとレプリケート先バケットの両方に適用する場合は、両方で同じライフサイクル設定を有効にします。ライフサイクル設定についての詳細は、[Amazon S3 on Outposts バケットのライフサイクル設定を作成および管理する](#) を参照してください。

## S3 Replication on Outposts でサポートされていないものは何ですか？

S3 レプリケーションの以下の機能は、現時点では S3 on Outposts でサポートされていません。

- S3 Replication Time Control (S3 RTC) S3 Replication on Outposts のオブジェクトトラフィックは オンプレミスネットワーク (ローカルゲートウェイ) を経由するため、S3 RTC はサポートされていません。ローカルゲートウェイの詳細については、AWS Outposts ユーザーガイドの「[ローカルゲートウェイの操作](#)」を参照してください。
- バッチオペレーションの S3 レプリケーション

## レプリケーションの設定

### Note

レプリケーションルールをセットアップする前にバケットに存在していたオブジェクトは、自動的にレプリケートされません。つまり、Amazon S3 on Outposts はさかのぼってオブジェクトをレプリケートしません。レプリケーション設定の前に作成されたオブジェクトをレプリケートするには、CopyObject API オペレーションを使用して、それらを同じバケットにコピーします。オブジェクトがコピーされると、バケットに「新しい」オブジェクトとして表示され、レプリケーション設定が適用されます。オブジェクトのコピーの詳細については、Amazon Simple Storage Service API リファレンスの「[AWS SDK for Java を使用して Amazon S3 on Outposts バケットにオブジェクトをコピーする](#)」および「[CopyObject](#)」を参照してください。

S3 Replication on Outposts を有効にするには、レプリケート元 Outposts バケットにレプリケーションルールを追加します。レプリケーションルールは、指定されたとおりにオブジェクトをレプリケートするように S3 on Outposts に指示します。レプリケーションルールでは、以下の項目を指定する必要があります。

- レプリケート元 Outposts バケットアクセスポイント – アクセスポイント Amazon Resource Name (ARN) または S3 on Outposts でオブジェクトをレプリケートするバケットのアクセスポイントエイリアス。アクセスポイントのエイリアスの使用の詳細については、「[Using a bucket-style alias for your S3 on Outposts bucket access point](#)」(S3 on Outposts バケットアクセスポイントにバケット形式のエイリアスを使用する)を参照してください。

- レプリケートするオブジェクト – レプリケート元 Outposts バケットまたはサブセット内のすべてのオブジェクトをレプリケートできます。サブセットを特定するには、[キー名のプレフィックス](#)、1 つ以上のオブジェクトタグ、またはその両方を設定で指定します。

例えば、キー名のプレフィックス Tax/ のオブジェクトのみをレプリケートするようにレプリケーションルールを設定した場合、S3 on Outposts は Tax/doc1 や Tax/doc2 などのキーを持つオブジェクトをレプリケートします。しかし、Legal/doc3 というキーを持つオブジェクトはレプリケートしません。プレフィックスと 1 つ以上のタグの両方を指定した場合、S3 on Outposts は特定のキープレフィックスとタグを持つオブジェクトのみをレプリケートします。

- レプリケート先の Outposts バケット – S3 on Outposts でオブジェクトをレプリケートするバケットの ARN またはアクセスポイントエイリアス。

REST API、AWS SDK、AWS Command Line Interface (AWS CLI)、または Amazon S3 コンソールを使用してレプリケーションルールを設定できます。

S3 on Outposts は、レプリケーションルールの設定をサポートする API も提供します。詳細については、Amazon Simple Storage Service API リファレンスの次のトピックを参照してください。

- [PutBucketReplication](#)
- [GetBucketReplication](#)
- [DeleteBucketReplication](#)

## トピック

- [レプリケーションルールの作成の前提条件](#)
- [Outposts でのレプリケーションルールの作成](#)

## レプリケーションルールの作成の前提条件

### トピック

- [レプリケート元とレプリケート先の Outpost サブネットの接続](#)
- [IAM ロールの作成](#)

## レプリケート元とレプリケート先の Outpost サブネットの接続

レプリケーショントラフィックをレプリケート元 Outpost からレプリケート先 Outpost にローカルゲートウェイ経由で送信するには、新しいルートを追加してネットワークを設定する必要があります。アクセスポイントの Classless Inter-Domain Routing (CIDR) ネットワーク範囲を相互に接続する必要があります。アクセスポイントのペアごとに、この接続を一度だけセットアップする必要があります。

接続を設定する手順は、アクセスポイントに関連付けられている Outposts エンドポイントのアクセスタイプによって異なります。エンドポイントのアクセスタイプは、[プライベート] (AWS Outposts のルーティング用直接仮想プライベートクラウド (VPC)) または [カスタマー所有の IP] (オンプレミスネットワーク内のカスタマー所有の IP アドレスプール [ColP プール]) のいずれかです。

### ステップ 1: レプリケート元 Outposts エンドポイントの CIDR 範囲を確認する

レプリケート元アクセスポイントに関連付けられているレプリケート元エンドポイントの CIDR 範囲を確認するには

1. AWS マネジメントコンソールにサインインし、Amazon S3 コンソール <https://console.aws.amazon.com/s3/> を開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. [Outposts バケット] リストで、レプリケーションに必要なレプリケート元バケットを選択します。
4. [Outposts アクセスポイント] タブを選択し、レプリケーションルールのレプリケート元バケットの Outposts アクセスポイントを選択します。
5. Outposts エンドポイントを選択します。
6. [ステップ 5](#) で使用するサブネット ID をコピーします。
7. レプリケート元 Outposts エンドポイントの CIDR 範囲を見つけるために使用する方法は、エンドポイントのアクセスタイプによって異なります。

Outposts エンドポイントの概要セクションで、「アクセスタイプ」を参照してください。

- アクセスタイプが [プライベート] の場合は、Classless Inter-Domain Routing (CIDR) 値をコピーして [ステップ 6](#) で使用します。
- アクセスタイプが [カスタマー所有の IP] の場合は、次の操作を行います。
  1. お客様所有の IPv4 プール値をコピーして、後でアドレスプールの ID として使用します。

2. AWS Outposts コンソールの <https://console.aws.amazon.com/outposts/> を開いてください。
3. ナビゲーションペインで [ローカルゲートウェイのルートテーブル] をクリックします。
4. レプリケート元 Outpost のローカルゲートウェイルートテーブル ID 値を選択します。
5. 詳細ペインの [CoIP プール] タブを選択します。以前にコピーした CoIP プール ID の値を検索ボックスに貼り付けます。
6. 一致した CoIP プールについては、[ステップ 6](#) で使用するレプリケート元 Outposts エンドポイントの対応する CIDR 値をコピーします。

## ステップ 2: レプリケート先の Outposts エンドポイントのサブネット ID と CIDR 範囲を確認する

レプリケート先アクセスポイントに関連付けられているレプリケート先エンドポイントのサブネット ID と CIDR 範囲を確認するには、[ステップ 1](#) と同じ手順に従い、それらのサブステップを適用するときに、レプリケート元 Outposts エンドポイントをレプリケート先 Outposts エンドポイントに変更します。[ステップ 6](#) で使用するために、レプリケート先 Outposts エンドポイントのサブネット ID 値をコピーします。[ステップ 5](#) で使用するために、レプリケート先 Outposts エンドポイントの CIDR 値をコピーします。

## ステップ 3: レプリケート元 Outpost のローカルゲートウェイ ID を確認する

レプリケート元 Outpost のローカルゲートウェイ ID を確認するには

1. AWS Outposts コンソールの <https://console.aws.amazon.com/outposts/> を開いてください。
2. 左のナビゲーションペインで [ローカルゲートウェイ] を選択します。
3. [ローカルゲートウェイ] ページで、レプリケーションに使用するレプリケート元 Outpost ID を確認します。
4. [ステップ 5](#) で使用するために、レプリケート元 Outpost のローカルゲートウェイ ID 値をコピーします。

ローカルゲートウェイの詳細については、AWS Outposts ユーザーガイドの「[Local gateway](#)」(ローカルゲートウェイ) を参照してください。

## ステップ 4: レプリケート先 Outpost のローカルゲートウェイ ID を確認する

レプリケート先 Outpost のローカルゲートウェイ ID を確認するには、レプリケート先の Outpost ID を検索する点を除き、[ステップ 3](#) と同じサブステップに従います。[ステップ 6](#) で使用するために、レプリケート先 Outpost のローカルゲートウェイ ID 値をコピーします。

## ステップ 5: レプリケート元 Outpost サブネットからレプリケート先 Outpost サブネットへの接続をセットアップする

レプリケート元 Outpost サブネットからレプリケート先 Outpost サブネットに接続するには

1. AWS マネジメントコンソール にサインインして、VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
2. 左側のナビゲーションペインで [Subnets] (サブネット) を選択します。
3. 検索ボックスに、[ステップ 1](#) で確認したレプリケート元 Outposts エンドポイントのサブネット ID を入力します。一致するサブネット ID のサブネットを選択します。
4. 一致するサブネット項目には、このサブネットのルートテーブル値を選択します。
5. ルートテーブルが選択されたページで、[アクション] を選択し、次に [ルートの編集] を選択します。
6. [ルートの編集] タブで、[ルートの追加] を選択します。
7. [送信先] に、[ステップ 2](#) で確認したレプリケート先 Outposts エンドポイントの CIDR 範囲を入力します。
8. [ターゲット] で [Outpost Local Gateway] を選択し、[ステップ 3](#) で見つけたレプリケート元 Outpost のローカルゲートウェイ ID を入力します。
9. [Save changes] (変更の保存) をクリックします。
10. ルートのステータスが [アクティブ] であることを確認します。

## ステップ 6: レプリケート先 Outpost サブネットからレプリケート元 Outpost サブネットへの接続をセットアップする

1. AWS マネジメントコンソール にサインインして、VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
2. 左側のナビゲーションペインで [Subnets] (サブネット) を選択します。
3. 検索ボックスに、[ステップ 2](#) で確認したレプリケート先 Outposts エンドポイントのサブネット ID を入力します。一致するサブネット ID のサブネットを選択します。
4. 一致するサブネット項目には、このサブネットのルートテーブル値を選択します。
5. ルートテーブルが選択されたページで、[アクション] を選択し、次に [ルートの編集] を選択します。
6. [ルートの編集] タブで、[ルートの追加] を選択します。

7. [送信先] に、[ステップ 1](#) で確認したレプリケート元 Outposts エンドポイントの CIDR 範囲を入力します。
8. [ターゲット] で [Outpost Local Gateway] を選択し、[ステップ 4](#) で見つけたレプリケート先 Outpost のローカルゲートウェイ ID を入力します。
9. [Save changes] (変更の保存) をクリックします。
10. ルートのステータスが [アクティブ] であることを確認します。

レプリケート元とレプリケート先のアクセスポイントの CIDR ネットワーク範囲を接続したら、AWS Identity and Access Management (IAM) ロールを作成する必要があります。

### IAM ロールの作成

デフォルトで、すべての S3 on Outposts リソース (バケット、オブジェクト、関連するサブリソース) はプライベートであり、リソース所有者のみがリソースにアクセスできます。S3 on Outposts には、ソース Outposts バケットからオブジェクトを読み取って、レプリケートするアクセス許可が必要です。IAM サービスロールを作成してこれらの許可を付与し、その後、レプリケーション設定でそのロールを指定します。

このセクションでは、信頼ポリシーと最低限必要なアクセス許可ポリシーについて説明します。チュートリアル例では、IAM ロールを作成するための手順をステップバイステップで説明しています。詳細については、「[Outposts でのレプリケーションルールの作成](#)」を参照してください。IAM ロールの詳細については、IAM ユーザーガイドの [IAM ロール](#) を参照してください。

- 以下の例は、信頼ポリシーを示しています。ここでは、このロールを引き受けることができるサービスプリンシパルとして S3 on Outposts を特定します。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3-outposts.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

```
}
```

- 以下の例は、アクセスポリシーを示しています。ここでは、ユーザーに代わってレプリケーションタスクを実行する権許可をロールに付与します。S3 on Outposts がこのロールを引き受ける場合、このポリシーで指定されたアクセス許可を持つこととなります。このポリシーを使用するには、*user input placeholders* をユーザー自身の情報に置き換えます。レプリケート元とレプリケート先 Outposts の Outposts ID およびレプリケート元とレプリケート先 Outposts バケットのバケット名とアクセスポイント名に置き換えてください。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3-outposts:GetObjectVersionForReplication",
        "s3-outposts:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3-outposts:us-east-1:123456789012:outpost/SOURCE-OUTPOST-ID/bucket/SOURCE-OUTPOSTS-BUCKET/object/*",
        "arn:aws:s3-outposts:us-east-1:123456789012:outpost/SOURCE-OUTPOST-ID/accesspoint/SOURCE-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3-outposts:ReplicateObject",
        "s3-outposts:ReplicateDelete"
      ],
      "Resource": [
        "arn:aws:s3-outposts:us-east-1:123456789012:outpost/DESTINATION-OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET/object/*",
        "arn:aws:s3-outposts:us-east-1:123456789012:outpost/DESTINATION-OUTPOST-ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
      ]
    }
  ]
}
```

}

アクセスポリシーは、以下のアクションに対するアクセス許可を付与します。

- `s3-outposts:GetObjectVersionForReplication` — このアクションのためのアクセス許可はすべてのオブジェクトに付与され、各オブジェクトに関連する特定のオブジェクトバージョンを S3 on Outposts が取得できるようにします。
- `s3-outposts:GetObjectVersionTagging` — *SOURCE-OUTPOSTS-BUCKET* バケット (レプリケート元バケット) のオブジェクトに対するこのアクションのアクセス許可により、S3 on Outposts はレプリケーションのためにオブジェクトタグを読み取ることができるようになります。詳細については、「[S3 on Outposts バケットのタグの追加](#)」を参照してください。S3 on Outposts にこのアクセス許可がない場合は、オブジェクトはレプリケートされますが、オブジェクトタグはレプリケートされません。
- `s3-outposts:ReplicateObject` および `s3-outposts:ReplicateDelete` — *DESTINATION-OUTPOSTS-BUCKET* バケットのオブジェクトに属すこれらのアクションの許可により、S3 on Outposts はレプリケート先 Outposts バケットにオブジェクトまたは削除マーカをレプリケートできます。削除マーカの詳細については、「[削除オペレーションがレプリケーションに与える影響](#)」を参照してください。

#### Note

- *DESTINATION-OUTPOSTS-BUCKET* バケット (レプリケート先バケット) に対する `s3-outposts:ReplicateObject` アクションのアクセス権限により、オブジェクトタグのレプリケーションも許可されます。したがって、`s3-outposts:ReplicateTags` アクションのアクセス許可を明示的に付与する必要はありません。
- クロスアカウントレプリケーションの場合、レプリケート先 Outposts バケットの所有者がバケットポリシーを更新して、*DESTINATION-OUTPOSTS-BUCKET* での `s3-outposts:ReplicateObject` アクションに対するアクセス許可を付与する必要があります。`s3-outposts:ReplicateObject` アクションにより、S3 on Outposts はレプリケート先 Outposts バケットにオブジェクトおよびオブジェクトタグをレプリケートできます。

S3 on Outposts アクションのリストについては、「[S3 on Outposts によって定義されたアクション](#)」を参照してください。

**⚠ Important**

IAM ロールを所有する AWS アカウントは、IAM ロールに付与するアクションの許可を持っている必要があります。

例えば、レプリケート元 Outposts バケットに別の AWS アカウントが所有するオブジェクトが含まれていたとします。オブジェクトの所有者は、IAM ロールを所有する AWS アカウントに、バケットポリシーとアクセスポイントポリシーを介して必要なアクセス許可を明示的に付与する必要があります。そうでない場合、S3 on Outposts はオブジェクトにアクセスできず、オブジェクトのレプリケーションは失敗します。

ここで説明されているアクセス許可は、最小のレプリケーション設定に関連しています。オプションのレプリケーション設定を追加する場合は、追加のアクセス許可を S3 on Outposts に付与する必要があります。

レプリケーション元とレプリケーション先の Outposts バケットが異なる AWS アカウントによって所有されている場合のアクセス許可の付与

レプリケーション元とレプリケーション先の Outposts バケットが同じアカウントによって所有されていない場合、レプリケート先バケットの所有者は、バケットとレプリケート先バケットのアクセスポイントポリシーを更新する必要があります。これらのポリシーでは、次のポリシー例に示すように、レプリケート元 Outposts バケットの所有者と IAM サービスロールにレプリケーションアクションを実行するアクセス許可を付与する必要があります。そうしないと、レプリケーションは失敗します。このポリシーの例では、***DESTINATION-OUTPOSTS-BUCKET*** はレプリケーション先バケットです。これらのポリシーの例を実行するには、***user input placeholders*** をユーザー自身の情報に置き換えます。

IAM サービスロールを手動で作成する場合は、次のポリシー例に示すように、ロールパスを `role/service-role/` として設定します。詳細については、IAM ユーザーガイドの「[IAM の ARN](#)」をご参照ください。

JSON

```
{
  "Version": "2012-10-17",
  "Id": "PolicyForDestinationBucket",
  "Statement": [
    {
```

```

        "Sid": "Permissions on objects",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:role/service-role/source-account-IAM-role"
        },
        "Action": [
            "s3-outposts:ReplicateDelete",
            "s3-outposts:ReplicateObject"
        ],
        "Resource": [
            "arn:aws:s3-outposts:us-east-1:444455556666:outpost/DESTINATION-OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET/object/*"
        ]
    }
]
}

```

## JSON

```

{
    "Version": "2012-10-17",
    "Id": "PolicyForDestinationAccessPoint",
    "Statement": [
        {
            "Sid": "Permissions on objects",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/service-role/source-account-IAM-role"
            },
            "Action": [
                "s3-outposts:ReplicateDelete",
                "s3-outposts:ReplicateObject"
            ],
            "Resource": [
                "arn:aws:s3-outposts:us-east-1:111122223333:outpost/DESTINATION-OUTPOST-ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
            ]
        }
    ]
}

```

```
}
```

### Note

レプリケート元 Outposts バケットのオブジェクトがタグ付きの場合は、以下の点に注意してください。

レプリケート元の Outposts バケット所有者が (IAM ロールによって) オブジェクトタグをレプリケートするための `s3-outposts:GetObjectVersionTagging` および `s3-outposts:ReplicateTags` アクションのアクセス許可を S3 on Outposts に付与した場合、Amazon S3 はオブジェクトと共にタグをレプリケートします。IAM ロールに関する詳細は、「[IAM ロールの作成](#)」を参照してください。

## Outposts でのレプリケーションルールの作成

S3 Replication on Outposts は、同一または異なる AWS Outposts にあるバケット間でのオブジェクトの自動的で非同期のレプリケーションです。レプリケーションでは、新しく作成されたオブジェクトおよびオブジェクトの更新が、レプリケート元 Outposts バケットからレプリケート先 Outposts バケットにコピーされます。詳細については、「[S3 on Outposts のオブジェクトのレプリケート](#)」を参照してください。

### Note

レプリケーションルールをセットアップする前にレプリケート元 Outposts バケットに存在したオブジェクト。つまり、S3 on Outposts はさかのぼってオブジェクトをレプリケートしません。レプリケーション設定の前に作成されたオブジェクトをレプリケートするには、CopyObject API オペレーションを使用して、それらを同じバケットにコピーします。オブジェクトがコピーされると、バケットに「新しい」オブジェクトとして表示され、レプリケーション設定が適用されます。オブジェクトのコピーの詳細については、Amazon Simple Storage Service API リファレンスの「[AWS SDK for Java を使用して Amazon S3 on Outposts バケットにオブジェクトをコピーする](#)」および「[CopyObject](#)」を参照してください。

レプリケーションを設定するときは、レプリケート元 Outposts バケットにレプリケーションルールを追加します。レプリケーションルールにより、レプリケート元 Outposts バケットオブジェクトと、レプリケートされたオブジェクトが保存されるレプリケート先 Outposts バケットが定義され

まず、ルールを作成して、バケット内のすべてのオブジェクト、または特定のキー名のプレフィックス、1つ以上のオブジェクトタグ、あるいはその両方を持つオブジェクトのサブセットをレプリケートできます。レプリケート先 Outposts バケツはレプリケート元 Outposts バケツと同じ Outposts にあっても、別の Outposts にあってもかまいません。

S3 on Outposts レプリケーションルールでは、レプリケート元とレプリケート先のバケツ名の代わりに、レプリケート元 Outposts バケツのアクセスポイント Amazon リソースネーム (ARN) とレプリケート先の Outposts バケツのアクセスポイント ARN の両方を指定する必要があります。

削除するオブジェクトバージョンの ID を指定した場合、S3 on Outposts はレプリケート元 Outposts バケツ内のそのオブジェクトバージョンを削除します。しかし、レプリケート先 Outposts バケツ内でその削除をレプリケートすることはありません。つまり、レプリケート先 Outposts バケツから同じオブジェクトバージョンを削除しません。この動作は悪意のある削除からデータを保護します。

Outposts バケツにレプリケーションルールを追加すると、そのルールはデフォルトで有効になるため、保存するとすぐに動作を開始します。

この例では、異なる Outposts にあり、同じ AWS アカウントが所有する、レプリケート元 Outposts とレプリケート先 Outposts のバケツのレプリケーションのセットアップを行います。Amazon S3 コンソールを使用する例については、AWS Command Line Interface (AWS CLI)、AWS SDK for Java、および AWS SDK for .NET を参照してください。クロスアカウントの S3 Replication on Outposts アクセス許可の詳細については、[「レプリケーション元とレプリケーション先の Outposts バケツが異なる AWS アカウントによって所有されている場合のアクセス許可の付与」](#)を参照してください。

S3 on Outposts レプリケーションルールを設定するための前提条件については、[「レプリケーションルールの作成の前提条件」](#)を参照してください。

## S3 コンソールの使用

以下の手順を実行して、レプリケート元 Outposts バケツと異なる Outposts にレプリケート先 Amazon S3 on Outposts バケツがある場合のレプリケーションルールを設定します。

レプリケート先 Outposts バケツがレプリケート元 Outposts バケツとは別のアカウントにある場合は、レプリケート先 Outposts バケツにバケツポリシーを追加して、レプリケート元 Outposts バケツアカウントの所有者に、レプリケート先 Outposts バケツのオブジェクトをレプリケートするアクセス許可を付与する必要があります。

## レプリケーションルールを作成するには

1. AWS マネジメントコンソール にサインインし、Amazon S3 コンソール <https://console.aws.amazon.com/s3/> を開きます。
2. [Outposts バケット] リストで、レプリケート元バケットとして使用するバケットの名前を選択します。
3. [管理] タブを選択し、[レプリケーションルール] までスクロールして、[レプリケーションルールを作成] を選択します。
4. [レプリケーションルール名] で、後でルールを識別しやすいようにルールの名前を入力します。ルール名は必須であり、バケット内で一意である必要があります。
5. [ステータス] では、デフォルトで [有効] が選択されています。有効にされたルールは、保存するとすぐに機能し始めます。ルールを後から有効にする場合は、[無効] を選択します。
6. [優先度] では、ルールの優先度値によって、重複するルールがある場合に適用するルールが決まります。オブジェクトが複数のレプリケーションルールの範囲に含まれる場合、S3 on Outposts はこれらの優先度の値を使用して競合を回避します。デフォルトでは、新しいルールが最優先でレプリケーション設定に追加されます。数値が大きいほど、優先度が高くなります。

ルールの優先度を変更するには、ルールを保存した後、レプリケーションルールリストからルール名を選択し、[アクション] を選択して、[優先度の編集] を選択します。

7. [ソースバケット] には、レプリケーションソースを設定するための次のオプションがあります。
  - バケット全体をレプリケートするには、[バケット内のすべてのオブジェクトに適用] を選択します。
  - プレフィックスまたはタグフィルタリングをレプリケーションソースに適用するには、[1 つまたは複数のフィルターを使用してこのルールの適用範囲を制限します] を選択します。プレフィックスとタグを組み合わせたことができます。
  - 同じプレフィックスを持つすべてのオブジェクトを複製するには、[プレフィックス] で、ボックスにプレフィックスを入力します。[プレフィックス] フィルターを使用すると、同じ文字列で始まる名前 (pictures など) を持つすべてのオブジェクトへのレプリケーションを制限します。


フォルダの名前をプレフィックスとして入力する場合は、最後の文字に / (スラッシュ) を使用する必要があります (例: pictures/)。

- 1 つまたは複数のオブジェクトタグを持つすべてのオブジェクトをレプリケートするには、[タグを追加] を選択し、ボックスにキーと値のペアを入力します。別のタグを追加する

には、この手順を繰り返します。オブジェクトタグの詳細については、[S3 on Outposts バケットのタグの追加](#)を参照してください。

8. S3 on Outposts のレプリケート元バケットにアクセスしてレプリケーションを行うには、[ソースアクセスポイント名] で、レプリケート元バケットに接続されているアクセスポイントを選択します。
9. [送信先] で、S3 on Outposts でオブジェクトをレプリケートするレプリケート先 Outposts バケットのアクセスポイント ARN を選択します。レプリケート先 Outposts バケットは、レプリケート元 Outposts バケットと同じまたは異なる AWS アカウント に配置することができます。

レプリケート先バケットがレプリケート元 Outposts バケットとは別のアカウントにある場合は、レプリケート先 Outposts バケットにバケットポリシーを追加して、レプリケート元 Outposts バケットアカウントの所有者に、レプリケート先 Outposts バケットのオブジェクトをレプリケートするアクセス許可を付与する必要があります。詳細については、「[レプリケーション元とレプリケーション先の Outposts バケットが異なる AWS アカウントによって所有されている場合のアクセス許可の付与](#)」を参照してください。

 Note

レプリケート先 Outposts バケットでバージョニングが有効になっていない場合は、[バージョニングの有効化] ボタンを含む警告が表示されます。バケットでバージョニングを有効にするには、このボタンを選択します。

10. S3 on Outposts がユーザーのためにオブジェクトをレプリケートできる AWS Identity and Access Management (IAM) サービスロールを設定します。

IAM ロールをセットアップするには、[IAM ロール] で、次のいずれかの操作を行います。

- S3 on Outposts でレプリケーション設定用の新しい IAM ロールを作成するには、[既存の IAM ロールから選択する] を選択し、[新しいロールの作成] を選択します。ルールを保存すると、選択したレプリケート元 Outposts バケットとレプリケート先 Outposts バケットに一致する IAM ロールに対して新しいポリシーが生成されます。[新しいロールを作成] を選択することを推奨します。
- 既存の IAM ロールの使用も選択できます。その場合は、レプリケーションに必要なアクセス許可を S3 on Outposts に付与するロールを選択する必要があります。このロールによってレプリケーションルールに従うための十分なアクセス許可が S3 on Outposts に付与されない場合、レプリケーションは失敗します。

既存のロールを選択するには、[既存の IAM ロールから選択] を選択してから、ドロップダウンメニューでロールを選択します。[IAM ロール ARN を入力する] を選択してから、IAM ロールの Amazon リソースネーム (ARN) を入力することもできます。

**⚠ Important**

S3 on Outposts バケットにレプリケーションルールを追加する場合は、S3 on Outposts レプリケーションアクセス許可を付与する IAM ロールを作成して渡すことができる `iam:CreateRole` と `iam:PassRole` アクセス許可を持つ必要があります。詳細については、IAM ユーザーガイドの「[AWS のサービスのサービスにロールを渡すアクセス権限をユーザーに付与する](#)」を参照してください。

- Outposts バケットのすべてのオブジェクトは、デフォルトで暗号化されます。S3 on Outposts の暗号化の詳細については、「[S3 on Outposts のデータ暗号化](#)」を参照してください。Amazon S3 マネージドキー (SSE-S3) によるサーバー側の暗号化を使用して暗号化されたオブジェクトのみがレプリケートできます。AWS Key Management Service(AWS KMS) キーを使用したサーバー側の暗号化 (SSE-KMS) またはお客様が用意した暗号化キーを使用したサーバー側の暗号化 (SSE-C) を使用して暗号化されたオブジェクトのレプリケーションはサポートされていません。
- 必要に応じて、レプリケーションルール設定を設定する際は、次の追加オプションを有効にします。
  - レプリケーション設定で S3 on Outposts レプリケーションメトリクスを有効にするには、[レプリケーションメトリクス] を選択します。詳細については、「[レプリケーションメトリクスによる進行状況のモニタリング](#)」を参照してください。
  - レプリケーション設定で削除マーカーレプリケーションを有効にする場合は、[削除マーカーのレプリケーション] を選択します。詳細については、「[削除オペレーションがレプリケーションに与える影響](#)」を参照してください。
  - レプリカに加えられたメタデータの変更をレプリケート元オブジェクトに複製する場合は、[レプリカ変更の同期] を選択します。詳細については、「[Outposts の Amazon S3 レプリカ変更の同期が有効になっている場合のレプリケーションステータス](#)」を参照してください。
- [ロールを作成] を選択して終了します。

ルールを保存したら、ルールを編集、有効化、無効化、または削除できます。そのためには、レプリケート元 Outposts バケットの [管理] タブに移動し、[レプリケーションルール] セクションまでスクロールして、ルールを選択し、[ルールの編集] を選択します。

## の使用AWS CLI

レプリケート元 Outposts バケットとレプリケート先 Outposts バケットが同じ AWS アカウント によって所有されている場合に、レプリケーションの設定に AWS CLI を使用するには

- レプリケート元 Outposts バケットとレプリケート先 Outposts バケットを作成する
- 両方のバケットでバージョニングを有効化する
- オブジェクトをレプリケートするための S3 on Outposts 許可を付与する IAM ロールを作成します。
- レプリケート元 Outposts バケットにレプリケーション設定を追加します。

設定を確認するには、テストします。

レプリケート元 Outposts バケットとレプリケート先 Outposts バケットが同じ AWS アカウント によって所有されている場合にレプリケーションをセットアップするには

1. AWS CLI の認証情報プロファイルを設定します。この例では、プロファイル名 `acctA` を使用します。認証情報プロファイルの設定については、AWS Command Line Interface ユーザーガイドの「[名前付きプロファイル](#)」を参照してください。

### Important

この演習に使用するプロファイルは、必要なアクセス権限を持っている必要があります。例えば、レプリケーション設定で、S3 on Outposts が引き受けることができる IAM ロールを指定します。使用するプロファイルに `iam:CreateRole` および `iam:PassRole` アクセス許可がある場合のみ実行できます。詳細については、IAM ユーザーガイドの「[AWS のサービスのサービスにロールを渡すアクセス権限をユーザーに付与する](#)」を参照してください。管理者の認証情報を使用して名前付きプロファイルを作成すると、指定されたプロファイルにはすべてのタスクを実行するために必要なアクセス許可が付与されます。

2. レプリケート元バケットを作成してバージョニングを有効にします。次の `create-bucket` コマンドは、米国東部 (バージニア北部) (`us-east-1`) リージョンに `SOURCE-OUTPOSTS-BUCKET` バケットを作成します。このコマンドを使用するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control create-bucket --bucket SOURCE-OUTPOSTS-BUCKET --outpost-id SOURCE-OUTPOST-ID --profile acctA --region us-east-1
```

次の `put-bucket-versioning` コマンドは、*SOURCE-OUTPOSTS-BUCKET* バケッでバージョンングを有効にします。このコマンドを使用するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/bucket/SOURCE-OUTPOSTS-BUCKET --versioning-configuration Status=Enabled --profile acctA
```

3. レプリケート先バケッを作成してバージョンングを有効にします。次の `create-bucket` コマンドは、米国西部 (オレゴン) (*us-west-2*) リージョンに *DESTINATION-OUTPOSTS-BUCKET* バケッを作成します。このコマンドを使用するには、*user input placeholders* をユーザー自身の情報に置き換えます。

#### Note

レプリケーション元とレプリケーション先の Outposts バケッの両方が同じ AWS アカウントにある場合、レプリケーション設定をセットアップするには、同じ名前付きプロファイルを使用します。この例では *acctA* を使用します。異なる AWS アカウントによってバケッが所有されている場合、レプリケーション設定をテストするには、それぞれのバケッに異なるプロファイルを指定します。

```
aws s3control create-bucket --bucket DESTINATION-OUTPOSTS-BUCKET --create-bucket-configuration LocationConstraint=us-west-2 --outpost-id DESTINATION-OUTPOST-ID --profile acctA --region us-west-2
```

次の `put-bucket-versioning` コマンドは、*DESTINATION-OUTPOSTS-BUCKET* バケッでバージョンングを有効にします。このコマンドを使用するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control put-bucket-versioning --account-id 123456789012 --bucket arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-ID/bucket/DESTINATION-OUTPOSTS-BUCKET --versioning-configuration Status=Enabled --profile acctA
```

4. [IAM サービスロールを作成します]。レプリケーション設定では、後で **SOURCE-OUTPOSTS-BUCKET** バケットにこのサービスロールを追加します。S3 on Outposts は、ユーザーに代わってオブジェクトをレプリケートするこのロールを引き受けます。IAM ロールは 2 つのステップで作成します。
  - a. IAM ロールを作成します。
    - i. 次の信頼ポリシーをコピーして、ローカルコンピュータの現在のディレクトリにある `s3-on-outposts-role-trust-policy.json` という名前のファイルに保存します。このポリシーは、サービスロールを引き受けるアクセス許可を S3 on Outposts サービスプリンシパルに付与します。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "s3-outposts.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- ii. 次のコマンドを実行してロールを作成します。*user input placeholders* を、ユーザー自身の情報に置き換えます。

```
aws iam create-role --role-name replicationRole --assume-role-policy-document file://s3-on-outposts-role-trust-policy.json --profile acctA
```

- b. アクセス許可ポリシーをサービスロールにアタッチします。
      - i. 次のアクセス権限ポリシーをコピーして、ローカルコンピュータの現在のディレクトリにある `s3-on-outposts-role-permissions-policy.json` という名前のファイルに保存します。このポリシーは、さまざまな S3 on Outposts バケットとオブジェクトアクションに対するアクセス許可を付与します。このポリシーを使用するには、*user input placeholders* をユーザー自身の情報に置き換えます。

## JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3-outposts:GetObjectVersionForReplication",
        "s3-outposts:GetObjectVersionTagging"
      ],
      "Resource": [
        "arn:aws:s3-outposts:us-  
east-1:123456789012:outpost/SOURCE-OUTPOST-ID/bucket/SOURCE-OUTPOSTS-  
BUCKET/object/*",
        "arn:aws:s3-outposts:us-  
east-1:123456789012:outpost/SOURCE-OUTPOST-ID/accesspoint/SOURCE-  
OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3-outposts:ReplicateObject",
        "s3-outposts:ReplicateDelete"
      ],
      "Resource": [
        "arn:aws:s3-outposts:us-  
east-1:123456789012:outpost/DESTINATION-OUTPOST-ID/  
bucket/DESTINATION-OUTPOSTS-BUCKET/object/*",
        "arn:aws:s3-outposts:us-  
east-1:123456789012:outpost/DESTINATION-OUTPOST-ID/  
accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT/object/*"
      ]
    }
  ]
}
```

- ii. ポリシーを作成してロールにアタッチするには、次のコマンドを実行します。 *user input placeholders* を、ユーザー自身の情報に置き換えます。

```
aws iam put-role-policy --role-name replicationRole --policy-  
document file://s3-on-outposts-role-permissions-policy.json --policy-  
name replicationRolePolicy --profile acctA
```

5. **SOURCE-OUTPOSTS-BUCKET** バケットにレプリケーション設定を追加します。

- a. S3 on Outposts API は XML 形式でのレプリケーション設定を要求しますが、AWS CLI ではレプリケーション設定を JSON 形式で指定する必要があります。以下の JSON を、コンピュータのローカルディレクトリの `replication.json` というファイルに保存します。この設定を使用するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
{  
  "Role": "IAM-role-ARN",  
  "Rules": [  
    {  
      "Status": "Enabled",  
      "Priority": 1,  
      "DeleteMarkerReplication": { "Status": "Disabled" },  
      "Filter" : { "Prefix": "Tax"},  
      "Destination": {  
        "Bucket":  
          "arn:aws:s3-outposts:region:123456789012:outpost/DESTINATION-OUTPOST-  
ID/accesspoint/DESTINATION-OUTPOSTS-BUCKET-ACCESS-POINT"  
        }  
      }  
    ]  
  }  
}
```

- b. 次の `put-bucket-replication` コマンドを実行して、レプリケート元 Outposts バケットにレプリケーション設定を追加します。このコマンドを使用するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control put-bucket-replication --account-id 123456789012 --  
bucket arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-  
ID/bucket/SOURCE-OUTPOSTS-BUCKET --replication-configuration file://  
replication.json --profile acctA
```

- c. レプリケーション設定を取得するには、`get-bucket-replication` コマンドを使用します。このコマンドを使用するには、*user input placeholders* をユーザー自身の情報に置き換えます。

```
aws s3control get-bucket-replication --account-id 123456789012 --bucket
arn:aws:s3-outposts:region:123456789012:outpost/SOURCE-OUTPOST-ID/
bucket/SOURCE-OUTPOSTS-BUCKET --profile acctA
```

6. Amazon S3 コンソールでセットアップをテストします。
  - a. AWS マネジメントコンソール にサインインし、Amazon S3 コンソール <https://console.aws.amazon.com/s3/> を開きます。
  - b. **SOURCE-OUTPOSTS-BUCKET** バケツに Tax という名前のフォルダを作成します。
  - c. **SOURCE-OUTPOSTS-BUCKET** バケツ内の Tax フォルダにサンプルオブジェクトを追加します。
  - d. **DESTINATION-OUTPOSTS-BUCKET** バケツで、以下の点を確認します。
    - S3 on Outposts がオブジェクトをレプリケートしました。

#### Note

S3 on Outposts がオブジェクトをレプリケートするのにかかる時間は、オブジェクトのサイズによって異なります。レプリケーションのステータスを確認する方法については、「[レプリケーションステータス情報の取得](#)」を参照してください。

- オブジェクトの [プロパティ] タブで、[レプリケーションステータス] が [レプリカ] に設定されています (これをレプリカオブジェクトとして識別)。

## レプリケーションの管理

このセクションでは、S3 on Outposts で使用可能な追加のレプリケーション設定オプション、レプリケーションステータスの確認方法、レプリケーションのトラブルシューティング方法について説明します 主要なレプリケーション設定については、「[レプリケーションの設定](#)」を参照してください。

### トピック

- [レプリケーションメトリクスによる進行状況のモニタリング](#)
- [レプリケーションステータス情報の取得](#)
- [レプリケーションのトラブルシューティング](#)

## • [S3 Replication on Outposts に EventBridge を使用する](#)

### レプリケーションメトリクスによる進行状況のモニタリング

S3 Replication on Outposts は、レプリケーション設定のレプリケーションルールの詳細なメトリクスを提供します。レプリケーションメトリクスを使用してレプリケーションの進行状況を 5 分間隔でモニタリングするには、保留中のレプリケーションのバイト数、レプリケーションのレイテンシー、レプリケーション、保留中のレプリケーションの操作を追跡します。設定の問題のトラブルシューティングを支援するために、Amazon EventBridge を設定して、レプリケーションの失敗に関する通知を受け取ることもできます。

レプリケーションメトリクスを有効にすると、S3 Replication on Outposts は次のメトリクスを Amazon CloudWatch に発行します。

- レプリケーションを保留しているバイト – 特定のレプリケーションルールのレプリケーションを保留しているオブジェクトの合計バイト数。
- レプリケーションレイテンシー – 特定のレプリケーションルールで、レプリケート先バケットがレプリケート元バケットの背後にある最大秒数。
- レプリケーションを保留している操作 – 特定のレプリケーションルールでレプリケーションを保留している操作の数。操作には、オブジェクト、削除マーカー、タグが含まれます

#### Note

S3 Replication on Outposts は、CloudWatch カスタムメトリクスと同じ料金レートで請求されます。詳細については、「[CloudWatch 料金表](#)」を参照してください。

### レプリケーションステータス情報の取得

レプリケーションステータスは、Amazon S3 on Outposts によってレプリケートされるオブジェクトの現在の状態を判断するのに役立ちます。レプリケート元オブジェクトのレプリケーションステータスは、PENDING、COMPLETEDまたはFAILEDのいずれかを返します。レプリカのレプリケーションステータスがREPLICAに返されます。

#### レプリケーションステータスの概要

レプリケーションシナリオでは、レプリケーションを設定するレプリケート元バケットと、S3 on Outposts がオブジェクトをレプリケートするレプリケート先バケットを使用します。これらのバ

ケットからオブジェクト (GetObject を使用) またはオブジェクトメタデータ (HeadObject を使用) をリクエストすると、S3 on Outposts はレスポンスとして以下のように `x-amz-replication-status` ヘッダーを返します。

- レプリケート元バケットのオブジェクトをリクエストする場合、リクエストしたオブジェクトがレプリケーション対象であると、S3 on Outposts は `x-amz-replication-status` ヘッダーを返します。

例えば、レプリケーション設定でオブジェクトプレフィックス `TaxDocs` を指定して、キー名のプレフィックス `TaxDocs` が付いたオブジェクトのみをレプリケートするように S3 on Outposts に指示しているとします。このキー名のプレフィックスを持つ、アップロードしたすべてのオブジェクト (`TaxDocs/document1.pdf` など) がレプリケートされます。このキー名のプレフィックスが付いたオブジェクトのリクエストでは、S3 on Outposts が、オブジェクトのレプリケーション状態が `PENDING`、`COMPLETED`、または `FAILED` の値のいずれかの `x-amz-replication-status` ヘッダーを返します。

#### Note

オブジェクトをアップロードした後で、オブジェクトのレプリケーションに失敗した場合、レプリケーションを再試行できません。もう一度オブジェクトをアップロードする必要があります。レプリケーションロールの許可またはバケットの許可がないなどの問題がある場合、オブジェクトは `FAILED` の状態に移行します。バケットや Outpost が使用できないなどの一時的な障害が発生した場合、レプリケーションのステータスは `FAILED` にはならず、`PENDING` のままになります。リソースがオンラインに戻ると、S3 on Outposts はこれらのオブジェクトのレプリケーションを再開します。

- レプリケート先バケットからオブジェクトをリクエストした場合、リクエストされたオブジェクトが S3 on Outposts によって作成されたレプリカであるときに、S3 on Outposts は値が `REPLICA` である `x-amz-replication-status` ヘッダーを返します。

#### Note

レプリケーションが有効になっているレプリケート元バケットからオブジェクトを削除する前に、削除する前にオブジェクトのレプリケーションステータスをチェックして、そのオブジェクトがレプリケートされていることを確認します。

Outposts の Amazon S3 レプリカ変更の同期が有効になっている場合のレプリケーションステータス

レプリケーションルールが S3 on Outposts レプリカの変更を有効にすると、レプリカは REPLICIA 以外のステータスをレポートできます。メタデータの変更がレプリケート中の場合は、レプリカの x-amz-replication-status ヘッダーは PENDING を返します。レプリカ変更の同期がメタデータのレプリケートに失敗した場合、レプリカのヘッダーは FAILED を返します。メタデータが正しくレプリケートされると、レプリカのヘッダーは値 REPLICIA を返します。

## レプリケーションのトラブルシューティング

レプリケーションを設定した後にオブジェクトレプリカがレプリケート先 Amazon S3 on Outposts バケットに表示されない場合は、これらのトラブルシューティングのヒントを使用して問題を特定し、修正してください。

- S3 on Outposts がオブジェクトをレプリケートするために要する時間はさまざまな要因に依存します (レプリケート元とレプリケート先の Outposts、オブジェクトのサイズなど)。

ソースオブジェクトのレプリケーションステータスを確認することもできます。オブジェクトのレプリケーションステータスが PENDING の場合は、S3 on Outposts がレプリケーションを完了していません。オブジェクトのレプリケーションステータスが FAILED の場合は、レプリケート元バケットのレプリケーション設定を確認してください。

- レプリケート元バケットのレプリケーション設定について、以下を確認します。
  - レプリケート先バケットのアクセスポイント Amazon リソースネーム (ARN) が正しい。
  - キー名のプレフィックスが正しい。たとえば、Tax というプレフィックスが付いたオブジェクトをレプリケートする設定にした場合、Tax/document1 や Tax/document2 のようなキー名のオブジェクトのみがレプリケートされます。キー名が "document3" のオブジェクトはレプリケートされません。
  - ステータスは Enabled です。
- どのバケットでもバージョニングが一時停止されていないことを確認します。レプリケート元とレプリケート先の両方のバケットで、バージョニングを有効にする必要があります。
- レプリケート先バケットが他の AWS アカウントによって所有されている場合、バケット所有者が、レプリケート元バケットの所有者に対してオブジェクトのレプリケーションを許可するバケットポリシーをレプリケート先バケットに設定していることを確認します。例については、[「レプリケーション元とレプリケーション先の Outposts バケットが異なる AWS アカウントによって所有されている場合のアクセス許可の付与」](#)を参照してください。
- オブジェクトのレプリカがレプリケート先バケットに表示されない場合、以下がレプリケーションを妨げている可能性があります。

- レプリケート元バケットのオブジェクト自体が別のレプリケーション設定によって作成されたレプリカである場合、S3 on Outposts はそのオブジェクトをレプリケートしません。例えば、バケット A からバケット B へ、バケット B からバケット C へのレプリケーション設定を設定した場合、S3 on Outposts はバケット B にあるオブジェクトレプリカをバケット C へレプリケートしません。

バケット A のオブジェクトをバケット B とバケット C にレプリケートする場合は、ソースバケットのレプリケーション設定の異なるレプリケーションルールで複数のバケット宛先を設定します。例えば、ソースバケット A に 2 つのレプリケーションルールを作成します。1 つのルールは宛先バケット B にレプリケートし、もう 1 つのルールは宛先バケット C にレプリケートします。

- レプリケート元バケット所有者は、オブジェクトをアップロードするための許可を他の AWS アカウントに付与できます。デフォルトでは、レプリケート元バケット所有者は、他のアカウントによって作成されたオブジェクトに対するアクセス許可を持ちません。レプリケーション設定では、レプリケート元バケット所有者がアクセス許可を持つオブジェクトのみがレプリケートされます。レプリケーションの問題を回避するために、レプリケート元バケット所有者は、条件付きでオブジェクトを作成するための他の AWS アカウント アクセス許可を付与し、それらのオブジェクトに対する明示的なアクセス許可を要求することができます。
- レプリケーション設定に、特定のタグを持つオブジェクトのサブセットをレプリケートするためのルールを追加するとします。この場合、S3 on Outposts がオブジェクトをレプリケートするには、オブジェクトの作成時に特定のタグキーと値を割り当てる必要があります。まずオブジェクトを作成して、それから既存のオブジェクトにタグを追加した場合、S3 on Outposts はそのオブジェクトをレプリケートしません。
- バケットポリシーで次のいずれかのアクションでレプリケーションのロールへのアクセスを拒否した場合、レプリケーションは失敗します。

### レプリケート元のバケット

```
"s3-outposts:GetObjectVersionForReplication",  
"s3-outposts:GetObjectVersionTagging"
```

### レプリケート先バケット:

```
"s3-outposts:ReplicateObject",  
"s3-outposts:ReplicateDelete",  
"s3-outposts:ReplicateTags"
```

- EventBridge は、オブジェクトがレプリケート先の Outposts にレプリケートされない場合に通知できます。詳細については、「[S3 Replication on Outposts に EventBridge を使用する](#)」を参照してください。

## S3 Replication on Outposts に EventBridge を使用する

Amazon S3 on Outposts は、Amazon EventBridge と統合されており、s3-outposts ネームスペースを使用します。EventBridge は、アプリケーションをさまざまなソースからのデータに接続するために使用できるサーバーレスのイベントバスサービスです。詳細については、「Amazon EventBridge ユーザーガイド」の「[Amazon EventBridge とは](#)」を参照してください。

レプリケーション設定の問題のトラブルシューティングを支援するために、Amazon EventBridge を設定して、レプリケーションの失敗イベントに関する通知を受け取ることもできます。EventBridge は、オブジェクトがデステイネーション Outposts にレプリケートされない場合に通知できます。レプリケートされるオブジェクトの現在の状態の詳細については、「[レプリケーションステータスの概要](#)」を参照してください。

Outposts バケット内で特定のイベントが発生するたびに S3 on Outposts は EventBridge にイベントを送信できます。他の宛先とは異なり、配信するイベントタイプを選択する必要はありません。また、EventBridge ルールを使用すると、イベントを追加のターゲットにルートできます。EventBridge を有効にすると、S3 on Outposts は次のすべてのイベントを EventBridge に送信します。

イベントタイプ	説明	名前空間
Operation FailedReplication	レプリケーションルール内のオブジェクトのレプリケーションが失敗しました。S3 Replication on Outposts の詳細については、「 <a href="#">EventBridge を使用して S3 Replication on Outposts の障害理由を表示する</a> 」を参照してください。	s3-outposts

### EventBridge を使用して S3 Replication on Outposts の障害理由を表示する

次の表は、Outposts の S3 レプリケーションが失敗した理由を示しています。EventBridge ルールを設定して、Amazon Simple Queue Service (Amazon SQS)、Amazon Simple Notification Service (Amazon SNS)、AWS Lambda、または Amazon CloudWatch Logs を介して失敗の理由を公開して

表示できます。EventBridge でこれらのリソースを使用するために必要なアクセス許可の詳細については、「[EventBridge のリソースベースのポリシーを使用する](#)」を参照してください。

レプリケーションの失敗の理由	説明
AssumeRoleNotPermitted	S3 on Outposts は、レプリケーション設定で指定されている AWS Identity and Access Management (IAM) ロールを引き継ぐことはできません。
DstBucketNotFound	S3 on Outposts は、レプリケーション設定で指定されたレプリケート先バケットを見つけることができません。
DstBucketUnversioned	Outposts レプリケート先バケットでバージョンングが有効になっていません。S3 Replication on Outposts でオブジェクトをレプリケートするには、レプリケート先バケットのバージョンングを有効にします。
DstDelObjNotPermitted	S3 on Outposts は、レプリケート先バケットに削除をレプリケートできません。レプリケート先バケットに対する <code>s3-outposts:ReplicateDelete</code> 権限がない可能性があります。
DstMultipartCompleteNotPermitted	S3 on Outposts は、レプリケート先バケットでオブジェクトのマルチパートアップロードを完了できません。レプリケート先バケットに対する <code>s3-outposts:ReplicateObject</code> 権限がない可能性があります。
DstMultipartInitNotPermitted	S3 on Outposts は、レプリケート先バケットへのオブジェクトのマルチパートアップロードを開始できません。レプリケート先バケットに対する <code>s3-outposts:Replic</code>

レプリケーションの失敗の理由	説明
	<code>ateObject</code> 権限がない可能性があります。
<code>DstMultipartPartUploadNotPermitted</code>	S3 on Outposts は、レプリケート先バケットにマルチパートアップロードオブジェクトをアップロードできません。レプリケート先バケットに対する <code>s3-outposts:ReplicateObject</code> 権限がない可能性があります。
<code>DstOutOfCapacity</code>	Outpost に S3 ストレージ容量が不足しているため、S3 on Outposts は、デステイネーションアウトポストにレプリケートできません。
<code>DstPutObjNotPermitted</code>	S3 on Outposts は、レプリケート先バケットにオブジェクトをレプリケートできません。レプリケート先バケットに対する <code>s3-outposts:ReplicateObject</code> 権限がない可能性があります。
<code>DstPutTaggingNotPermitted</code>	S3 on Outposts は、レプリケート先バケットにオブジェクトタグをレプリケートできません。レプリケート先バケットに対する <code>s3-outposts:ReplicateObject</code> 権限がない可能性があります。
<code>DstVersionNotFound</code>	S3 on Outposts は、そのオブジェクトバージョンのメタデータをレプリケートするためにレプリケート先バケットで必要なオブジェクトバージョンを見つけることができません。

レプリケーションの失敗の理由	説明
SrcBucketReplicationConfigMissing	S3 on Outposts は、ソースの Outposts バケットに関連付けられているアクセスポイントのレプリケーション設定を見つけることができません。
SrcGetObjectNotPermitted	S3 on Outposts は、レプリケーションのレプリケート元バケットにあるオブジェクトにアクセスできません。レプリケート元バケットに対する s3-outposts:GetObjectVersionForReplication 権限がない可能性があります。
SrcGetTaggingNotPermitted	S3 on Outposts は、レプリケート元バケットからオブジェクトタグ情報にアクセスできません。レプリケート元バケットに対する s3-outposts:GetObjectVersionTagging 権限がない可能性があります。
SrcHeadObjectNotPermitted	S3 on Outposts は、レプリケート元バケットからオブジェクトメタデータを取得できません。レプリケート元バケットに対する s3-outposts:GetObjectVersionForReplication 権限がない可能性があります。
SrcObjectNotEligible	このオブジェクトはレプリケーションの対象外です。オブジェクトまたはそのオブジェクトタグがレプリケーション設定と一致しません。

レプリケーションのトラブルシューティングの詳細については、以下のトピックを参照してください。

- [IAM ロールの作成](#)
- [レプリケーションのトラブルシューティング](#)

## CloudWatch での EventBridge のモニタリング

モニタリングには、Amazon EventBridge が Amazon CloudWatch と統合します。EventBridge は自動的に CloudWatch に毎分メトリクスを送信します。これらのメトリクスには、[ルール](#)に一致した[イベント](#)の数と、ルールによって[ターゲット](#)が呼び出された回数が含まれます。Eventbridge でルールが実行されると、このルールに関連付けられているすべてのターゲットが呼び出されます。EventBridge の動作は、CloudWatch を通じて次の方法でモニタリングできます。

- EventBridge ルールで利用可能な [EventBridge メトリクス](#)は、CloudWatch ダッシュボードからモニタリングできます。次に、CloudWatch アラームなどの CloudWatch 機能を使用して、特定のメトリクスにアラームを設定できます。これらのメトリクスがアラームで指定したカスタムしきい値に達すると、通知が届き、それに応じてアクションを実行できます。
- Amazon CloudWatch Logs を EventBridge ルールのターゲットとして設定できます。次に、EventBridge がログストリームを作成し、CloudWatch Logs がログエントリとしてイベントからテキストを保存します。詳細については、「[EventBridge の CloudWatch Logs のアクセス許可](#)」を参照してください。

EventBridge イベント配信のデバッグとイベントのアーカイブの詳細については、次のトピックを参照してください。

- [イベントの再試行ポリシーとデッドレターキューの使用](#)
- [EventBridge イベントのアーカイブ](#)

## AWS RAM を使用して S3 on Outposts を共有する

Simple Storage Service (Amazon S3) on Outposts は、AWS Resource Access Manager ([AWS RAM](#)) を使用して、組織内の複数のアカウント間で S3 容量の共有をサポートしています。。S3 on Outposts 共有を使用すると、他のユーザーが Outpost でバケット、エンドポイント、およびアクセスポイントを作成および管理できるようになります。

このトピックでは、AWS RAM を使用して S3 on Outposts と関連するリソースを、お客様の AWS 組織の別の AWS アカウントと共有する方法をご紹介します。

### 前提条件

- Outpost 所有者アカウントには AWS Organizations で設定された組織が存在します。詳細については、「AWS Organizations ユーザーガイド」で「[組織を作成する](#)」を参照してください。

- 組織には S3 on Outposts の容量を共有する AWS アカウント が含まれます。詳細については、「AWS Organizations ユーザーガイド」の「[AWS アカウント への招待の送信](#)」を参照してください。
- 共有する以下のオプションのいずれかを選択します。エンドポイントにもアクセスできるように、2 番目のリソース (サブネットまたは Outposts) を選択する必要があります。エンドポイントは、S3 on Outposts に格納されたデータにアクセスするためのネットワーク要件です。

オプション 1	オプション 2
S3 on Outposts	S3 on Outposts
ユーザーが Outposts とアクセスポイントにバケットを作成し、それらのバケットにオブジェクトを追加できるようにします。	ユーザーが Outposts とアクセスポイントにバケットを作成し、それらのバケットにオブジェクトを追加できるようにします。
サブネット	Outposts
ユーザーが仮想プライベートクラウド (VPC) とサブネットに関連付けられているエンドポイントを使用できるようにします。	ユーザーが S3 容量チャートと AWS Outposts コンソールのホームページを参照できるようにします。また、共有の Outposts にサブネットを作成し、エンドポイントを作成することもできます。

## 手順

1. Outpost を所有する AWS アカウント を使用して AWS マネジメントコンソール にサインインし、<https://console.aws.amazon.com/ram/home> で AWS RAM コンソールを開きます。
2. AWS RAM で AWS Organizations との共有が有効になっていることを確認します。詳細については、「AWS RAM ユーザーガイド」の「[AWS Organizations 内のリソース共有を有効化する](#)」を参照してください。
3. [前提条件](#)で、オプション 1 またはオプション 2 を使用してリソース共有を作成します。複数の S3 on Outposts リソースがある場合は、共有するリソースの Amazon リソース名前 (ARN) を選択します。エンドポイントを有効にするには、サブネットまたは Outpost を共有します。

リソース共有の作成の詳細については、「AWS RAM ユーザーガイド」の「[リソース共有を作成する](#)」を参照してください。

4. リソースを共有した AWS アカウント は、S3 on Outposts を使用できるようになります。[前提条件](#)で選択したオプションに応じて、アカウントユーザーに次の情報を提供します。

オプション 1	オプション 2
Outpost ID	Outpost ID
VPC ID。	
サブネット ID	
セキュリティグループの ID	

#### Note

ユーザーは、AWS RAM コンソール、AWS Command Line Interface (AWS CLI)、AWS SDK、または REST API を使用して、リソースが共有されていることを確認できます。ユーザーは、[get-resource-shares](#) CLI コマンドを使用して、既存のリソース共有を表示できます。

## 使用例

S3 on Outposts リソースを別のアカウントと共有した後、そのアカウントは Outpost のバケットとオブジェクトを管理できます。[Subnets] (サブネット) リソース共有した場合、そのアカウントは、作成したエンドポイントを使用できます。次の例では、ユーザーが AWS CLI を使用し、これらのリソースを共有した後、Outpost とやり取るする方法についてご紹介します。

Example: バケットの作成

次の例では、Outpost `op-01ac5d28a6a232904` で `amzn-s3-demo-bucket1` という名前のバケットを作成します。このコマンドを使用する前に、それぞれの *user input placeholder* をユースケースに適した値に置き換えてください。

```
aws s3control create-bucket --bucket amzn-s3-demo-bucket1 --outpost-id op-01ac5d28a6a232904
```

このコマンドの詳細については、「AWS CLI リファレンス」の「[create-bucket](#)」を参照してください。

#### Example: アクセスポイントの作成

次の例では、次の表に示すパラメータ例を使用して、Outpost 上にアクセスポイントを作成します。このコマンドを使用する前に、これらの *user input placeholder* 値と AWS リージョンコードをユースケースに適した値に置き換えてください。

パラメータ	値
アカウント ID	<i>111122223333</i>
アクセスポイント名	<i>example-outpost-access-point</i>
Outpost ID	<i>op-01ac5d28a6a232904</i>
Outpost バケット名	<i>amzn-s3-demo-bucket1</i>
VPC ID	<i>vpc-1a2b3c4d5e6f7g8h9</i>

#### Note

アカウント ID パラメータは、AWS アカウントバケット所有者 (共有ユーザー) の ID である必要があります。

```
aws s3control create-access-point --account-id 111122223333 --name example-outpost-access-point \  
--bucket arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/  
bucket/amzn-s3-demo-bucket1 \  
--vpc-configuration VpcId=vpc-1a2b3c4d5e6f7g8h9
```

このコマンドの使用に関する詳細については、「AWS CLI リファレンス」の「[create-access-point](#)」を参照してください。

## Example: オブジェクトのアップロード

次の例では、ユーザーのローカルファイルシステムから、AWS アカウント `111122223333` が所有しているアクセスポイント `example-outpost-access-point` on the Outpost `op-01ac5d28a6a232904` を経由し、`images/my_image.jpg` という名前のオブジェクトへ、ファイル `my_image.jpg` をアップロードします。このコマンドを使用する前に、これらの `user input placeholder` 値と AWS リージョンコードをユースケースに適した値に置き換えてください。

```
aws s3api put-object --bucket arn:aws:s3-outposts:us-east-1:111122223333:outpost/op-01ac5d28a6a232904/accesspoint/example-outpost-access-point \
--body my_image.jpg --key images/my_image.jpg
```

このコマンドの詳細については、「AWS CLI リファレンス」の「[put-object](#)」を参照してください。

### Note

このオペレーションの結果、[Resource not found] (リソースが見つからない) エラーまたは応答がない場合、VPC に共有エンドポイントがない可能性があります。

共有エンドポイントがあるかどうかを確認するには、[list-shared-endpoints](#) AWS CLI コマンドを使用します。共有エンドポイントがない場合は、Outpost の所有者と連携して作成します。詳細については、「Amazon Simple Storage Service API リファレンス」の「[ListSharedEndpoints](#)」を参照してください。

## Example: エンドポイントの作成

次の例では、共有の Outposts にエンドポイントを作成します。このコマンドを使用する前に、Outpost ID、サブネット ID、およびセキュリティグループ ID の `user input placeholder` 値をユースケースに適した値に置き換えます。

### Note

ユーザーがこの操作を実行できるのは、リソース共有に Outposts リソースが含まれている場合に限りです。

```
aws s3outposts create-endpoint --outposts-id op-01ac5d28a6a232904 --subnet-id XXXXXX --  
security-group-id XXXXXX
```

このコマンドの詳細については、「AWS CLI リファレンス」の「[create-endpoint](#)」を参照してください。

## S3 on Outposts を使用するその他の AWS のサービス

AWS Outposts に対してローカルで実行するその他の AWS のサービスも、Amazon S3 on Outposts の容量を使用できます。Amazon CloudWatch では、S3Outposts 名前空間には、S3 on Outposts 内のバケットの詳細なメトリクスが表示されますが、これらのメトリクスには、他 AWS のサービスの使用は含まれません。他の AWS のサービスによって使用される S3 on Outposts の容量を管理するには、次の表の情報を参照してください。

AWS のサービス	説明	詳細情報
Amazon S3	すべての直接的な S3 on Outposts の使用量には、一致するアカウントおよびバケットの CloudWatch メトリクスがあります。	<a href="#">メトリクスを参照</a>
Amazon Elastic Block Store (Amazon EBS)	Outposts 上の Amazon EBS の場合、スナップショットのデステイネーションとして AWS Outpost を選択して、S3 on Outpost にローカルに保存できます。	<a href="#">詳細情報</a>
Amazon Relational Database Service (Amazon RDS)	Amazon RDS ローカルバックアップを使用して、RDS バックアップを Outpost にローカルに保存できます。	<a href="#">詳細情報</a>

# S3 on Outposts のモニタリング

Amazon S3 on Outposts を使用すると、AWS Outposts で S3 バケットを作成し、ローカルデータアクセス、ローカルデータ処理、データレジデンシーを必要とするアプリケーション用に、オンプレミスのオブジェクトを簡単に保存および取得できます。S3 on Outposts は、新しいストレージクラス、S3 Outposts (OUTPOSTS) を提供し、これは Amazon S3 API を使用し、AWS Outposts 上の複数のデバイスやサーバー間でデータを永続的かつ冗長的に保存するように設計されています。仮想プライベートクラウド (VPC) を介したアクセスポイントとエンドポイント接続を使用して、Outposts バケットと通信します。Outposts バケットでは、Amazon S3 と同じ API と機能 (アクセスポリシー、暗号化、タグ付けなど) を使用できます。AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDK、または REST API を使用して S3 on Outposts を使用できます。詳細については、[Amazon S3 on Outposts とは](#) を参照してください。

Amazon S3 on Outposts ストレージ容量のモニタリングの詳細については、以下のトピックを参照してください。

## トピック

- [Amazon CloudWatch メトリクスを使用した S3 on Outposts 容量の管理](#)
- [Amazon CloudWatch Events を使用した S3 on Outposts イベント通知の受信](#)
- [AWS CloudTrail ログで S3 on Outposts をモニタリングする](#)

## Amazon CloudWatch メトリクスを使用した S3 on Outposts 容量の管理

S3 の容量を管理するために、ストレージ使用率が特定のしきい値を超えたときにその旨を知らせる CloudWatch アラートを作成するようお勧めします。S3 on Outposts の CloudWatch メトリクスの詳細については、「[CloudWatch メトリクス](#)」を参照してください。Outpost にオブジェクトを保存するための十分なスペースがない場合、API は容量不足の例外 (ICE) を返します。容量を空けるには、明示的なデータ削除をトリガーする CloudWatch アラームを作成するか、ライフサイクルの有効期限ポリシーを使用してオブジェクトを期限切れにすることができます。削除前にデータを保存するには、AWS DataSync を使用して Amazon S3 on Outposts のバケットから AWS リージョンにある S3 バケットにデータをコピーできます。DataSync の使用の詳細については、「AWS DataSync ユーザーガイド」の「[AWS DataSync の開始方法](#)」を参照してください。

## CloudWatch メトリクス

S3Outposts 名前空間には、Amazon S3 on Outposts バケットのための以下のメトリクスが含まれます。プロビジョニングされた S3 on Outposts バイトの合計数、オブジェクトで使用可能な合計空きバイト数、および特定のバケットのすべてのオブジェクトの合計サイズをモニタリングできます。バケットまたはアカウント関連のメトリクスは、S3 を直接使用する場合すべてに存在しています。Amazon Elastic Block Store のローカルスナップショットや Amazon Relational Database Service バックアップを Outpost に保存するなど、S3 を間接的に使用することで S3 の容量が消費されますが、バケットやアカウント関連のメトリクスには含まれません。Amazon EBS スナップショットの詳細については、「[Amazon EBS Local Snapshots on Outposts](#)」(Outposts の Amazon EBS ローカルスナップショット) を参照してください。Amazon EBS コストレポートを確認するには、<https://console.aws.amazon.com/costmanagement/> をご覧ください。

### Note

S3 on Outposts は、次のメトリクスのみをサポートし、他の Amazon S3 メトリクスはサポートしません。

S3 on Outposts の容量は限られているため、ストレージ使用率が特定のしきい値を超えたときに警告する CloudWatch アラートを作成するようお勧めします。

メトリクス	説明	期間	単位	タイプ
OutpostTotalByte	Outpost のプロビジョニングされた合計キャパシティー (バイト単位)。	5 分	バイト	S3 on Outposts
OutpostFreeBytes	お客様のデータを保存するために Outpost で使用可能な空きバイト数。	5 分	バイト	S3 on Outposts
BucketUsedBytes	指定されたバケットのすべてのオブジェクトの合計サイズ。	5 分	バイト	S3 on Outposts。S3 での直接利用のみ。

メトリクス	説明	期間	単位	タイプ
AccountTotalBytes	指定された Outposts アカウントのすべてのオブジェクトの合計サイズ。	5 分	バイト	S3 on Outposts。S3 での直接利用のみ。
BytesPerReplication	特定のレプリケーションルールについて、レプリケーションが保留中のオブジェクトの合計バイト数。レプリケーションメトリクスを有効にする方法の詳細については、「 <a href="#">Outposts 間のレプリケーションルールの作成</a> 」を参照してください。	5 分	バイト	オプション。Outposts の S3 レプリケーションの場合。
OperationsPending	特定のレプリケーションルールについて、レプリケーションが保留中のオペレーションの総数。レプリケーションメトリクスを有効にする方法の詳細については、「 <a href="#">Outposts 間のレプリケーションルールの作成</a> 」を参照してください。	5 分	カウント	オプション。Outposts の S3 レプリケーションの場合。
ReplicationLatency	特定のレプリケーションルールで、レプリケーション先バケットがレプリケート元バケットよりも遅れている現在の遅延秒数。レプリケーションメトリクスを有効にする方法の詳細については、「 <a href="#">Outposts 間のレプリケーションルールの作成</a> 」を参照してください。	5 分	秒	オプション。Outposts の S3 レプリケーションの場合。

# Amazon CloudWatch Events を使用した S3 on Outposts イベント通知の受信

CloudWatch Events を使用して、Amazon S3 on Outposts API イベントのルールを作成できます。ルールを作成するときに、Amazon Simple Queue Service (Amazon SQS)、Amazon Simple Notification Service (Amazon SNS)、および AWS Lambda を含む、サポートされているすべての CloudWatch ターゲットを通じて通知を受け取ることを選択できます。詳細については、「Amazon CloudWatch Events ユーザーガイド」の[CloudWatch Events のターゲットとなるAWS サービスのリスト](#)を参照してください。S3 on Outposts で操作するターゲットサービスを選択するには、「Amazon CloudWatch Events ユーザーガイド」の「[AWS CloudTrail を使用して AWS API コールでトリガーする CloudWatch Events ルールの作成](#)」を参照してください。

## Note

S3 on Outposts オブジェクトオペレーションでは、CloudTrail によって送信される AWS API コールイベントは、それらのイベントを受信するように証跡 (オプションでイベントセレクターが付いています) を設定している場合にのみ、ルールに一致します。詳細については、AWS CloudTrail ユーザーガイドの[CloudTrail ログファイルの操作](#)を参照してください。

## Example

以下に、DeleteObject オペレーションのサンプルルールを示します。このサンプルルールを使用するには、*amzn-s3-demo-bucket1* を S3 on Outposts バケットの名前と置き換えます。

```
{
  "source": [
    "aws.s3-outposts"
  ],
  "detail-type": [
    "AWS API call through CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "s3-outposts.amazonaws.com"
    ],
    "eventName": [
      "DeleteObject"
    ],
  ],
}
```

```
"requestParameters": {  
  "bucketName": [  
    "amzn-s3-demo-bucket1"  
  ]  
}
```

## AWS CloudTrail ログで S3 on Outposts をモニタリングする

Amazon S3 on Outposts は AWS CloudTrail と統合されています。このサービスは、ユーザーやロール、または S3 on Outposts の AWS のサービス によって実行されたアクションを記録するサービスです。AWS CloudTrail を使用して、S3 on Outposts のバケットレベルおよびオブジェクトレベルのリクエストに関する情報を取得し、S3 on Outposts イベントアクティビティを監査してログに記録できます。

すべての Outposts バケットまたは特定の Outposts バケットのリストの CloudTrail データイベントを有効にするには、[CloudTrail で手動によって証跡を作成する](#) 必要があります。CloudTrail ログファイルエントリの詳細については、「[S3 on Outposts ログファイルエントリ](#)」を参照してください。

S3 on Outposts の CloudTrail データイベントの完全なリストについては、「Amazon S3 ユーザーガイド」の「[CloudTrail の Amazon S3 データイベント](#)」を参照してください。

### Note

- ベストプラクティスとして、AWS CloudTrail データイベント Outposts バケットに対してライフサイクルポリシーを作成することをお勧めします。ログファイルを監査する必要がある期間が経過したらログファイルを定期的に削除するように、ライフサイクルポリシーを設定します。これにより、各クエリで Amazon Athena が分析するデータの量が減ります。詳細については、「[Amazon S3 on Outposts バケットのライフサイクル設定を作成および管理する](#)」を参照してください。
- CloudTrail ログをクエリする方法の例については、AWS ビッグデータブログの記事 [Analyze Security, Compliance, and Operational Activity Using AWS CloudTrail and Amazon Athena](#) を参照してください。

## S3 on Outposts バケットでオブジェクトの CloudTrail ログ記録を有効にするには

Amazon S3 コンソールを使用し、AWS CloudTrail 証跡を設定して、Amazon S3 on Outposts バケット内のオブジェクトのデータイベントをログに記録できます。CloudTrail では、GetObject、DeleteObject、PutObject など、S3 on Outposts オブジェクトレベルの API オペレーションのログ記録がサポートされます。これらのイベントは、データイベントと呼ばれます。

デフォルトでは、CloudTrail 証跡はデータイベントを記録しません。ただし、指定した S3 on Outposts バケットのデータイベントを記録するか、AWS アカウントの S3 on Outposts バケット上のすべてのデータイベントを記録するように証跡を設定できます。

CloudTrail では、CloudTrail イベント履歴にデータイベントが設定されません。さらに、すべての S3 on Outposts バケットレベルの API オペレーションが CloudTrail イベント履歴に入力されるわけではありません。CloudTrail ログのクエリ方法の詳細については、AWS ナレッジセンターの「[Amazon CloudWatch Logs のフィルターパターンと Amazon Athena を使用して CloudTrail ログをクエリする](#)」を参照してください。

S3 on Outposts バケットのデータイベントをログに記録するように証跡を設定する場合、AWS CloudTrail コンソールまたは Amazon S3 コンソールのいずれかを使用できます。AWS アカウント内のすべての S3 on Outposts バケットのデータイベントを記録するように証跡を設定する場合は、CloudTrail コンソールを使用する方が簡単です。CloudTrail コンソールを使用して S3 on Outposts データイベントを記録するように証跡を設定する方法については、AWS CloudTrail ユーザーガイドの「[データイベント](#)」を参照してください。

### Important

追加の変更がイベントデータに適用されます。詳細については、[AWS CloudTrail 料金表](#)を参照してください。

以下の手順では、Amazon S3 コンソールを使用して、CloudTrail の証跡で S3 on Outposts バケットのデータイベントの記録を設定する方法を示します。

**Note**

バケットを作成する AWS アカウントが、そのバケットを所有し、AWS CloudTrail に送信する S3 on Outposts データイベントを設定できる唯一のアカウントです。

S3 on Outposts バケットでオブジェクトの CloudTrail データイベントの記録を有効にするには

1. AWS マネジメントコンソール にサインインし、Amazon S3 コンソール <https://console.aws.amazon.com/s3/> を開きます。
2. 左のナビゲーションペインで、[Outposts buckets] (Outposts バケット) を選択します。
3. CloudTrail を使用してログを記録するデータイベントがある Outposts バケットの名前を選択します。
4. [プロパティ] を選択します。
5. [AWS CloudTrail データイベント] セクションに進み、[CloudTrail で設定] を選択します。

AWS CloudTrail コンソールが開きます。

新しい CloudTrail の証跡を作成するか、既存の証跡を再利用して、証跡に記録するように S3 on Outposts データイベントを設定できます。

6. CloudTrail コンソールの[ダッシュボード] ページで、[証跡の作成] を選択します。
7. [ステップ 1 証跡の属性の選択] ページで、証跡の名前を入力し、証跡ログを保存する S3 バケットを選択し、その他の必要な設定を指定して、[次へ] を選択します。
8. [ステップ 2 の ログイベントの選択] ページの [イベントタイプ] で、[データイベント] を選択します。

[データイベントタイプ] では、[S3 Outposts] を選択します。[次へ] を選択します。

**Note**

- トレイルを作成して S3 on Outposts のデータイベントのログ記録を設定する場合、データイベントタイプを正しく指定する必要があります。
- CloudTrail コンソールを使用する場合は、[データイベントタイプ] で [S3 Outposts] を選択します。CloudTrail コンソールで証跡を作成する方法については、[AWS CloudTrail ユーザーガイド](#)のコンソールで証跡を作成および更新するを参照してください。CloudTrail コンソールで S3 on Outposts データイベントのログ記録を設定

する方法については、AWS CloudTrail ユーザーガイドの「[Amazon S3 オブジェクトのデータイベントをログに記録する](#)」を参照してください。

- AWS Command Line Interface (AWS CLI) または AWS SDK を使用する場合は、resources.type フィールドを AWS::S3Outposts::Object に設定します。AWS CLI を使用して S3 on Outposts データイベントをログに記録する方法の詳細については、AWS CloudTrail ユーザーガイドの「[S3 on Outposts イベントのログ記録](#)」を参照してください。
- S3 on Outposts バケットのデータイベントをログに記録するための証跡の設定に CloudTrail コンソールまたは Amazon S3 コンソールを使用する場合、そのバケットに対してオブジェクトレベルのロギングが有効化されていることが Amazon S3 コンソールに表示されます。

9. [ステップ 3 の確認と作成] ページで、設定した証跡属性とログイベントを確認します。次に、[証跡の作成]を選択します。

S3 on Outposts バケットでオブジェクトの CloudTrail データイベントの記録を無効にするには

1. AWS マネジメントコンソールにサインインし、<https://console.aws.amazon.com/cloudtrail/>で CloudTrail コンソールを開きます。
2. 左のナビゲーションペインで、[証跡] を選択します。
3. S3 on Outposts バケットのイベントを記録するために作成した証跡の名前を選択します。
4. 証跡の詳細ページで、右上隅にある [ログインを停止] を選択します。
5. 表示されたダイアログボックスで [ログインを停止] を選択します。

## Amazon S3 on Outposts AWS CloudTrail ログファイルのエントリ

Amazon S3 on Outposts 管理イベントは、AWS CloudTrail を通じて利用できます。さらに、必要に応じて[AWS CloudTrail のデータイベントのログ記録を有効にする](#)こともできます。

証跡は、リージョン内の指定した 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。Outposts バケットの CloudTrail ログには新しいフィールドの edgeDeviceDetails が含まれており、指定されたバケットが配置されている Outposts を識別できます。

追加のログフィールドには、要求されたアクション、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。

次の例では、s3-outposts での [PutObject](#) アクションを示す CloudTrail ログエントリについて説明します。

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "111122223333",
    "arn": "arn:aws:iam::111122223333:user/yourUserName",
    "accountId": "222222222222",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "yourUserName"
  },
  "eventTime": "2020-11-30T15:44:33Z",
  "eventSource": "s3-outposts.amazonaws.com",
  "eventName": "PutObject",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "26.29.66.20",
  "userAgent": "aws-cli/1.18.39 Python/3.4.10 Darwin/18.7.0 botocore/1.15.39",
  "requestParameters": {
    "expires": "Wed, 21 Oct 2020 07:28:00 GMT",
    "Content-Language": "english",
    "x-amz-server-side-encryption-customer-key-MD5": "wJa1rXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY",
    "ObjectCannedACL": "BucketOwnerFullControl",
    "x-amz-server-side-encryption": "Aes256",
    "Content-Encoding": "gzip",
    "Content-Length": "10",
    "Cache-Control": "no-cache",
    "Content-Type": "text/html; charset=UTF-8",
    "Content-Disposition": "attachment",
    "Content-MD5": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
    "x-amz-storage-class": "Outposts",
    "x-amz-server-side-encryption-customer-algorithm": "Aes256",
    "bucketName": "amzn-s3-demo-bucket1",
    "Key": "path/upload.sh"
  },
  "responseElements": {
    "x-amz-server-side-encryption-customer-key-MD5": "wJa1rXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY",
    "x-amz-server-side-encryption": "Aes256",
    "x-amz-version-id": "001",
    "x-amz-server-side-encryption-customer-algorithm": "Aes256",
```

```
    "ETag": "d41d8cd98f00b204e9800998ecf8427f"
  },
  "additionalEventData": {
    "CipherSuite": "ECDHE-RSA-AES128-SHA",
    "bytesTransferredIn": 10,
    "x-amz-id-2": "29xXQBV20
+x0HKItvzY1suLv1i6A52E0z0X159fpfsItYd58JhXwKxXAXI4IQkp6",
    "SignatureVersion": "SigV4",
    "bytesTransferredOut": 20,
    "AuthenticationMethod": "AuthHeader"
  },
  "requestID": "8E96D972160306FA",
  "eventID": "ee3b4e0c-ab12-459b-9998-0a5a6f2e4015",
  "readOnly": false,
  "resources": [
    {
      "accountId": "222222222222",
      "type": "AWS::S3Outposts::Object",
      "ARN": "arn:aws:s3-outposts:us-east-1:YYY:outpost/op-01ac5d28a6a232904/
bucket/path/upload.sh"
    },
    {
      "accountId": "222222222222",
      "type": "AWS::S3Outposts::Bucket",
      "ARN": "arn:aws:s3-outposts:us-east-1:YYY:outpost/op-01ac5d28a6a232904/
bucket/"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": false,
  "recipientAccountId": "444455556666",
  "sharedEventID": "02759a4c-c040-4758-b84b-7cbaaf17747a",
  "edgeDeviceDetails": {
    "type": "outposts",
    "deviceId": "op-01ac5d28a6a232904"
  },
  "eventCategory": "Data"
}
```

# Amazon S3 on Outposts での開発

Amazon S3 on Outposts を使用すると、AWS Outposts で S3 バケットを作成し、ローカルデータアクセス、ローカルデータ処理、データレジデンシーを必要とするアプリケーション用に、オンプレミスのオブジェクトを簡単に保存および取得できます。S3 on Outposts は、新しいストレージクラス、S3 Outposts (OUTPOSTS) を提供し、これは Amazon S3 API を使用し、AWS Outposts 上の複数のデバイスやサーバー間でデータを永続的かつ冗長的に保存するように設計されています。仮想プライベートクラウド (VPC) を介したアクセスポイントとエンドポイント接続を使用して、Outposts バケットと通信します。Outposts バケットでは、Amazon S3 と同じ API と機能 (アクセスポリシー、暗号化、タグ付けなど) を使用できます。AWS マネジメントコンソール、AWS Command Line Interface (AWS CLI)、AWS SDK、または REST API を使用して S3 on Outposts を使用できます。詳細については、「[Amazon S3 on Outposts とは](#)」を参照してください。

以下のトピックでは、S3 on Outposts での開発について説明します。

## トピック

- [S3 on Outposts でサポートされているリージョン](#)
- [Amazon S3 on Outposts の API オペレーション](#)
- [SDK for Java を使用して、S3 on Outposts の S3 コントロールクライアントを設定する](#)
- [IPv6 経由で S3 on Outposts へのリクエストを行う](#)

## S3 on Outposts でサポートされているリージョン

S3 on Outposts は、次の AWS リージョンでサポートされています。

- 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 (北カリフォルニア) (us-west-1)
- 米国西部 (オレゴン) (us-west-2)
- アフリカ (ケープタウン) (af-south-1)
- 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 (大阪) (ap-northeast-3)
- 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)
- ヨーロッパ (フランクフルト) (eu-central-1)
- 欧州 (アイルランド) (eu-west-1)
- ヨーロッパ (ロンドン) (eu-west-2)
- 欧州 (ミラノ) (eu-south-1)
- 欧州 (パリ) (eu-west-3)
- 欧州 (ストックホルム) (eu-north-1)
- イスラエル (テルアビブ) (il-central-1)
- 中東 (バーレーン) (me-south-1)
- 南米 (サンパウロ) (sa-east-1)
- AWS GovCloud (米国東部) (us-gov-east-1)
- AWS GovCloud (米国西部) (us-gov-west-1)

## Amazon S3 on Outposts の API オペレーション

このトピックでは、Amazon S3 Outposts で使用できる Amazon S3、Amazon S3 コントロール、および Amazon S3 on Outposts API オペレーションの一覧を示します。

### トピック

- [オブジェクト管理のための Amazon S3 API オペレーション](#)
- [バケット管理のための Amazon S3 コントロール API オペレーション](#)
- [Outposts を管理するための S3 on Outposts API オペレーション](#)

## オブジェクト管理のための Amazon S3 API オペレーション

S3 on Outposts は、Amazon S3 と同じオブジェクト API オペレーションを使用するように設計されています。Outpost バケット内の任意のオブジェクトにアクセスするには、アクセスポイントを使用する必要があります。S3 on Outposts でオブジェクト API オペレーションを使用するときは、Outposts アクセスポイントの Amazon リソースネーム (ARN) またはアクセスポイントエイリア

スのいずれかを指定します。アクセスポイントエイリアスの詳細については、「[S3 on Outposts アクセスポイントでのバケット形式のエイリアスの使用](#)」を参照してください。

Amazon S3 on Outposts では、以下の Amazon S3 API のオペレーションがサポートされています。

- [AbortMultipartUpload](#)
- [CompleteMultipartUpload](#)
- [CopyObject](#)
- [CreateMultipartUpload](#)
- [DeleteObject](#)
- [DeleteObjects](#)
- [DeleteObjectTagging](#)
- [GetObject](#)
- [GetObjectTagging](#)
- [HeadBucket](#)
- [HeadObject](#)
- [ListMultipartUploads](#)
- [ListObjects](#)
- [ListObjectsV2](#)
- [ListObjectVersions](#)
- [ListParts](#)
- [PutObject](#)
- [PutObjectTagging](#)
- [UploadPart](#)
- [UploadPartCopy](#)

## バケット管理のための Amazon S3 コントロール API オペレーション

S3 on Outposts では、バケット操作のために以下の Amazon S3 コントロール API オペレーションがサポートされています。

- [CreateAccessPoint](#)
- [CreateBucket](#)

- [DeleteAccessPoint](#)
- [DeleteAccessPointPolicy](#)
- [DeleteBucket](#)
- [DeleteBucketLifecycleConfiguration](#)
- [DeleteBucketPolicy](#)
- [DeleteBucketReplication](#)
- [DeleteBucketTagging](#)
- [GetAccessPoint](#)
- [GetAccessPointPolicy](#)
- [GetBucket](#)
- [GetBucketLifecycleConfiguration](#)
- [GetBucketPolicy](#)
- [GetBucketReplication](#)
- [GetBucketTagging](#)
- [GetBucketVersioning](#)
- [ListAccessPoints](#)
- [ListRegionalBuckets](#)
- [PutAccessPointPolicy](#)
- [PutBucketLifecycleConfiguration](#)
- [PutBucketPolicy](#)
- [PutBucketReplication](#)
- [PutBucketTagging](#)
- [PutBucketVersioning](#)

## Outposts を管理するための S3 on Outposts API オペレーション

S3 on Outposts では、エンドポイントを管理するために、以下の Amazon S3 on Outposts API オペレーションがサポートされています。

- [CreateEndpoint](#)
- [DeleteEndpoint](#)
- [ListEndpoints](#)

- [ListOutpostsWithS3](#)
- [ListSharedEndpoints](#)

## SDK for Java を使用して、S3 on Outposts の S3 コントロールクライアントを設定する

次の例では、AWS SDK for Java を使用して、Amazon S3 on Outposts の Amazon S3 コントロールクライアントを設定します。この例を実行するには、それぞれの *user input placeholder* をユーザー自身の情報に置き換えます。

```
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.services.s3control.AWSS3Control;
import com.amazonaws.services.s3control.AWSS3ControlClient;

public AWSS3Control createS3ControlClient() {

    String accessKey = AWSAccessKey;
    String secretKey = SecretAccessKey;
    BasicAWSCredentials awsCreds = new BasicAWSCredentials(accessKey, secretKey);

    return AWSS3ControlClient.builder().enableUseArnRegion()
        .withCredentials(new AWSStaticCredentialsProvider(awsCreds))
        .build();
}
```

## IPv6 経由で S3 on Outposts へのリクエストを行う

Amazon S3 on Outposts デュアルスタックエンドポイントは、IPv6 プロトコルまたは IPv4 プロトコルを使用した S3 on Outposts バケットへのリクエストをサポートしています。S3 on Outposts の IPv6 サポートを使用すると、IPv6 ネットワーク上の S3 on Outposts API 経由でバケットとコントロールプレーンリソースにアクセスし、運用できます。

### Note

[S3 on Outposts オブジェクトアクション](#) (PutObject や GetObject など) は、IPv6 ネットワークではサポートされていません。

IPv6 ネットワーク上で S3 on Outposts にアクセスする場合、追加料金はかかりません。S3 on Outposts の詳細については、「[S3 on Outposts の料金](#)」を参照してください。

## トピック

- [IPv6 の使用開始](#)
- [デュアルスタックのエンドポイントを使用して IPv6 ネットワーク経由でリクエストを実行する](#)
- [IAM ポリシーでの IPv6 アドレスの使用](#)
- [IP アドレス互換性のテスト](#)
- [AWS PrivateLink での IPv6 の使用](#)
- [S3 on Outposts デュアルスタックのエンドポイントの使用](#)

## IPv6 の使用開始

IPv6 経由で S3 on Outposts バケットにリクエストを実行するには、デュアルスタックのエンドポイントを使用する必要があります。次のセクションでは、デュアルスタックのエンドポイントを使用した IPv6 でのリクエストの実行方法について説明します。

以下は、IPv6 経由で S3 on Outposts バケットへのアクセスを試行する際の重要な考慮事項です。

- 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。
- 仮想ホスト形式およびパス形式のリクエストは、IPv6 アクセスをサポートしています。詳細については、「[S3 on Outposts デュアルスタックのエンドポイントの使用](#)」を参照してください。
- AWS Identity and Access Management (IAM) ユーザーまたは S3 on Outposts バケットポリシーで、ソース IP アドレスによるフィルタリングを使用する場合、IPv6 アドレス範囲を含めるようポリシーを更新する必要があります。

### Note

この要件は、IPv6 ネットワーク上の S3 on Outposts バケットオペレーションとコントロールプレーンリソースにのみ適用されます。[Amazon S3 on Outposts オブジェクトアクション](#)は、IPv6 ネットワークではサポートされていません。

- IPv6 を使用する場合、サーバーのアクセスログファイルは IPv6 形式の IP アドレスを出力します。IPv6 形式のリモート IP アドレスを解析できるように、S3 on Outposts ログファイルの解析に使用する既存のツール、スクリプト、ソフトウェアを更新する必要があります。その後、更新

されたツール、スクリプト、ソフトウェアは IPv6 形式のリモート IP アドレスを正しく解析しません。

## デュアルスタックのエンドポイントを使用して IPv6 ネットワーク経由でリクエストを実行する

IPv6 上で S3 on Outposts API コールを使用してリクエストを行うには、AWS CLI または AWS SDK 経由でデュアルスタックのエンドポイントを使用できます。[Amazon S3 コントロール API オペレーション](#)と [S3 on Outposts API オペレーション](#)は、IPv6 プロトコルまたは IPv4 プロトコル経由で S3 on Outposts にアクセスしているかどうかに関係なく、同じように機能します。ただし、[S3 on Outposts オブジェクトアクション](#) (PutObject や GetObject など) は、IPv6 ネットワークではサポートされていません。

AWS Command Line Interface (AWS CLI) や AWS SDK を使用する場合、パラメータまたはフラグを使ってデュアルスタックのエンドポイントに変更できます。設定ファイルの S3 on Outposts エンドポイントに上書きしてデュアルスタックのエンドポイントを直接指定することもできます。

デュアルスタックのエンドポイントを使用して、次のいずれかから IPv6 経由で S3 on Outposts バケットにアクセスできます。

- AWS CLI については、「[AWS CLI からのデュアルスタックのエンドポイントの使用](#)」を参照してください。
- AWS SDK については、[AWS SDK から S3 on Outposts デュアルスタックのエンドポイントを使用する](#)を参照してください。

## IAM ポリシーでの IPv6 アドレスの使用

IPv6 プロトコルを使用して S3 on Outposts バケットへのアクセスを試行する前に、IP アドレスによるフィルタリングに使用される IAM ユーザーまたは S3 on Outposts バケットポリシーが IPv6 アドレス範囲を含むように更新されているか確認する必要があります。IPv6 アドレスを処理するように IP アドレスフィルタリングポリシーが更新されていない場合、IPv6 プロトコルの使用を試行しているときに S3 on Outposts バケットにアクセスできなくなる可能性があります。

IP アドレスをフィルタリングする IAM ポリシーは、[IP アドレス条件演算子](#)を使用します。次の S3 on Outposts バケットポリシーは、IP アドレス条件演算子を使用して 54.240.143.\* の許可される IP 範囲の IPv4 アドレスを識別します。この範囲外のすべての IP アドレスは S3 on Outposts バケット

へのアクセスを拒否されます (DOC-EXAMPLE-BUCKET)。すべての IPv6 アドレスは許可範囲外であるため、このポリシーは IPv6 アドレスの DOC-EXAMPLE-BUCKET へのアクセスをブロックします。

## JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3-outposts:*",
      "Resource": "arn:aws:s3-outposts:us-east-1:111122223333:outpost/OUTPOSTS-ID/bucket/DOC-EXAMPLE-BUCKET/*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "54.240.143.0/24"
        }
      }
    }
  ]
}
```

次の例のように、S3 on Outposts バケットポリシーの Condition エlementを変更して、IPv4 (54.240.143.0/24) および IPv6 (2001:DB8:1234:5678::/64) アドレス範囲の両方を許可できます。例に示すように、IAM ユーザーとバケットポリシーの両方を更新するために同じタイプの Condition ブロックを使用できます。

```
"Condition": {
  "IpAddress": {
    "aws:SourceIp": [
      "54.240.143.0/24",
      "2001:DB8:1234:5678::/64"
    ]
  }
}
```

IPv6 を使用する前に、IPv6 アドレス範囲を許可する IP アドレスのフィルタリングに使用するすべての関連 IAM ユーザーとバケットポリシーを更新しなければなりません。既存の IPv4 アドレス範囲

に加えて、IAM ポリシーと組織の IPv6 アドレス範囲を更新することをお勧めします。IPv6 および IPv4; でのアクセスを許可するバケットポリシーの例については、「[特定の IP アドレスへのアクセスの制限](#)」を参照してください。

IAM ユーザーポリシーは、<https://console.aws.amazon.com/iam/> の IAM コンソールを使用して確認できます。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。S3 on Outposts バケットポリシーの編集の詳細については、「[Amazon S3 on Outposts バケットのバケットポリシーを追加または編集する](#)」を参照してください。

## IP アドレス互換性のテスト

Linux、Unix インスタンス、または macOS X プラットフォームを使用している場合は、IPv6 経由でデュアルスタックのエンドポイントへのアクセスをテストできます。例えば、IPv6 経由で Amazon S3 on Outposts エンドポイントへの接続をテストするには、dig コマンドを使用します。

```
dig s3-outposts.us-west-2.api.aws AAAA +short
```

IPv6 ネットワーク上にデュアルスタックのエンドポイントが適切に設定されている場合、dig コマンドは接続されている IPv6 アドレスを返します。例えば、次のようになります。

```
dig s3-outposts.us-west-2.api.aws AAAA +short

2600:1f14:2588:4800:b3a9:1460:159f:ebce

2600:1f14:2588:4802:6df6:c1fd:ef8a:fc76

2600:1f14:2588:4801:d802:8ccf:4e04:817
```

## AWS PrivateLink での IPv6 の使用

S3 on Outposts は、AWS PrivateLink サービスとエンドポイントの IPv6 プロトコルをサポートしています。IPv6 プロトコルの AWS PrivateLink のサポートにより、オンプレミスまたは他のプライベート接続から、IPv6 ネットワーク経由で VPC 内のサービスエンドポイントに接続できます。[S3 on Outposts の AWS PrivateLink](#) の IPv6 サポートにより、AWS PrivateLink をデュアルスタックのエンドポイントと統合することもできます。AWS PrivateLink で IPv6 を有効にする手順については、「[Expedite your IPv6 adoption with AWS PrivateLink services and endpoints](#)」を参照してください。

**Note**

サポートされている IP アドレスタイプを IPv4 から IPv6 に更新するには、「AWS PrivateLink ユーザーガイド」の「[サポートされている IP アドレスのタイプを変更する](#)」を参照してください。

## AWS PrivateLink での IPv6 の使用

IPv6 で AWS PrivateLink を使用している場合は、IPv6 またはデュアルスタックの VPC インターフェイスエンドポイントを作成する必要があります。AWS マネジメントコンソールを使用して VPC エンドポイントを作成する一般的な手順については、「AWS PrivateLink ユーザーガイド」の「[インターフェイス VPC エンドポイントを使用して AWS のサービスにアクセスする](#)」を参照してください。

### AWS マネジメントコンソール

次の手順を使用して、S3 on Outposts に接続するインターフェイス VPC エンドポイントを作成します。

1. AWS マネジメントコンソール にサインインして、VPC コンソール (<https://console.aws.amazon.com/vpc/>) を開きます。
2. ナビゲーションペインで、[エンドポイント] を選択します。
3. [エンドポイントの作成] を選択します。
4. [Service category] (サービスカテゴリ) で、[AWS services] (のサービス) を選択します。
5. [サービス名] で、S3 on Outposts サービス (com.amazonaws.us-east-1.s3-outposts) を選択します。
6. [VPC] で、S3 on Outposts にアクセスする VPC を選択します。
7. [サブネット] で、S3 on Outposts にアクセスするアベイラビリティゾーンごとにサブネットを 1 つ選択します。同じアベイラビリティゾーンから複数のサブネットを選択することはできません。選択したサブネットごとに、新しいエンドポイントネットワークインターフェイスが作成されます。デフォルトでは、サブネットの IP アドレス範囲の IP アドレスがエンドポイントネットワークインターフェイスに割り当てられます。エンドポイントネットワークインターフェイスの IP アドレスを指定するには、[IP アドレスの指定] を選択し、サブネットアドレス範囲の IPv6 アドレスを入力します。
8. [IP アドレスタイプ] で、[デュアルスタック] を選択します。IPv4 と IPv6 の両方のアドレスをエンドポイントのネットワークインターフェイスに割り当てます。このオプションは、選択

したすべてのサブネットに IPv4 と IPv6 両方のアドレス範囲がある場合にのみサポートされます。

9. [セキュリティグループ] で、VPC エンドポイントのエンドポイントネットワークインターフェイスに関連付けるセキュリティグループを選択します。デフォルトでは、デフォルトのセキュリティグループが VPC に関連付けられます。
10. [ポリシー] では、[フルアクセス] を選択して、VPC エンドポイントのすべてのリソースに対するすべてのプリンシパルによるすべてのオペレーションを許可します。それ以外の場合は、[カスタム] を選択して、プリンシパルが VPC エンドポイントを介してリソースに対してアクションを実行するために必要なアクセス許可を制御する VPC エンドポイントポリシーをアタッチします。このオプションは、サービスが VPC エンドポイントポリシーをサポートしている場合にのみ使用できます。詳細については、「[エンドポイントポリシー](#)」を参照してください。
11. (オプション) タグを追加するには、[新しいタグを追加] を選択し、そのタグのキーと値を入力してください。
12. エンドポイントの作成 を選択します。

#### Example– S3 on Outposts バケットポリシー

S3 on Outposts が VPC エンドポイントとやり取りできるようにするには、S3 on Outposts ポリシーを次のように更新します。

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3-outposts:*",
      "Resource": "*",
      "Principal": "*"
    }
  ]
}
```

#### AWS CLI

##### Note

VPC エンドポイントで IPv6 ネットワークを有効にするには、S3 on Outposts の SupportedIpAddressType フィルターに IPv6 を設定する必要があります。

次の例では、`create-vpc-endpoint` コマンドを使用して新しいデュアルスタックインターフェイスエンドポイントを作成します。

```
aws ec2 create-vpc-endpoint \  
--vpc-id vpc-12345678 \  
--vpc-endpoint-type Interface \  
--service-name com.amazonaws.us-east-1.s3-outposts \  
--subnet-id subnet-12345678 \  
--security-group-id sg-12345678 \  
--ip-address-type dualstack \  
--dns-options "DnsRecordIpType=dualstack"
```

AWS PrivateLink サービス設定によっては、新しく作成されたエンドポイント接続を使用する前に、VPC エンドポイントサービスプロバイダーによる承認が必要になる場合があります。詳細については、「AWS PrivateLink ユーザーガイド」の「[エンドポイントの接続リクエストを承諾または拒否する](#)」を参照してください。

次の例では、`modify-vpc-endpoint` コマンドを使用して、IPv 専用 VPC エンドポイントをデュアルスタックのエンドポイントに更新します。デュアルスタックのエンドポイントにより、IPv4 ネットワークと IPv6 ネットワークの両方にアクセスできるようになります。

```
aws ec2 modify-vpc-endpoint \  
--vpc-endpoint-id vpce-12345678 \  
--add-subnet-ids subnet-12345678 \  
--remove-subnet-ids subnet-12345678 \  
--ip-address-type dualstack \  
--dns-options "DnsRecordIpType=dualstack"
```

AWS PrivateLink で IPv6 を有効にする方法の詳細については、「[Expedite your IPv6 adoption with AWS PrivateLink services and endpoints](#)」を参照してください。

## S3 on Outposts デュアルスタックのエンドポイントの使用

S3 on Outposts デュアルスタックエンドポイントは、IPv6 および IPv4 を使用した S3 on Outposts バケットへのリクエストをサポートしています。このセクションは、S3 on Outposts デュアルスタックのエンドポイントを使用する方法を説明します。

### トピック

- [S3 on Outposts デュアルスタックのエンドポイント](#)

- [AWS CLI からのデュアルスタックのエンドポイントの使用](#)
- [AWS SDK から S3 on Outposts デュアルスタックのエンドポイントを使用する](#)

## S3 on Outposts デュアルスタックのエンドポイント

デュアルスタックのエンドポイントにリクエストを行うと、S3 on Outposts バケット URL は IPv6 または IPv4 アドレスに解決されます。IPv6 での S3 on Outposts バケットへのアクセスに関する詳細は、「[IPv6 経由で S3 on Outposts へのリクエストを行う](#)」を参照してください。

デュアルスタックのエンドポイントを介して S3 on Outposts バケットにアクセスするには、パス形式のエンドポイント名を使用します。S3 on Outposts はリージョンのデュアルスタックエンドポイント名のみをサポートしており、名前の一部としてリージョンを指定する必要があります。

デュアルスタックパス形式の FIPS エンドポイントの場合は、次の命名規則を使用します。

```
s3-outposts-fips.region.api.aws
```

デュアルスタックの非 FIPS エンドポイントの場合は、次の命名規則を使用します。

```
s3-outposts.region.api.aws
```

### Note

仮想ホスト形式のエンドポイント名は、S3 on Outposts ではサポートされていません。

## AWS CLI からのデュアルスタックのエンドポイントの使用

このセクションでは、デュアルスタックのエンドポイントへリクエストするのに使用される AWS CLI コマンドの例を示します。AWS CLI をセットアップする手順については、「[AWS CLI および SDK for Java の使用開始](#)」を参照してください。

AWS Config ファイルのプロファイル内で設定値 `use_dualstack_endpoint` を `true` に設定すると、`s3` と `s3api` AWS CLI コマンドによるすべての Amazon S3 リクエストが、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転送されます。--region オプションを使用して設定ファイルまたはコマンドでリージョンを指定します。

AWS CLI でデュアルスタックのエンドポイントを使用する場合、`path` のアドレス形式のみがサポートされます。設定ファイルで設定されたアドレス形式により、バケット名がホスト名の一部か

URL の一部が決まります。詳細については、AWS CLI ユーザーガイドの [s3outposts](#) を参照してください。

AWS CLI 経由でデュアルスタックのエンドポイントを使用するには、`s3control` コマンドまたは `s3outposts` コマンドの `http://s3.dualstack.region.amazonaws.com` または `https://s3-outposts-fips.region.api.aws` のエンドポイントを指定して、`--endpoint-url` パラメータを使用します。

例えば、次のようになります。

```
$ aws s3control list-regional-buckets --endpoint-url https://s3-outposts.region.api.aws
```

## AWS SDK から S3 on Outposts デュアルスタックのエンドポイントを使用する

このセクションでは、AWS SDK を使用してデュアルスタックのエンドポイントにアクセスする方法の例を示します。

### AWS SDK for Java 2.x のデュアルスタックのエンドポイントの例

次の例では、AWS SDK for Java 2.x を使用して S3 on Outposts クライアントを作成する際に、`S3ControlClient` クラスと `S3OutpostsClient` クラスを使用してデュアルスタックエンドポイントを有効化する方法を示しています。Amazon S3 on Outposts の動作する Java の例の作成とテストに関する手順については、「[AWS CLI および SDK for Java の使用開始](#)」を参照してください。

### Example– デュアルスタックのエンドポイントを有効にして S3ControlClient クラスを作成する

```
import com.amazonaws.AmazonServiceException;
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3control.S3ControlClient;
import software.amazon.awssdk.services.s3control.model.ListRegionalBucketsRequest;
import software.amazon.awssdk.services.s3control.model.ListRegionalBucketsResponse;
import software.amazon.awssdk.services.s3control.model.S3ControlException;

public class DualStackEndpointsExample1 {

    public static void main(String[] args) {
        Region clientRegion = Region.of("us-east-1");
```

```
String accountId = "111122223333";
String navyId = "9876543210";

try {
    // Create an S3ControlClient with dual-stack endpoints enabled.
    S3ControlClient s3ControlClient = S3ControlClient.builder()
                                                    .region(clientRegion)
                                                    .dualstackEnabled(true)
                                                    .build();

    ListRegionalBucketsRequest listRegionalBucketsRequest =
ListRegionalBucketsRequest.builder()

        .accountId(accountId)

        .outpostId(navyId)

        .build();

    ListRegionalBucketsResponse listBuckets =
s3ControlClient.listRegionalBuckets(listRegionalBucketsRequest);
    System.out.printf("ListRegionalBuckets Response: %s\n",
listBuckets.toString());
    } catch (AmazonServiceException e) {
        // The call was transmitted successfully, but Amazon S3 on Outposts
couldn't process
        // it, so it returned an error response.
        e.printStackTrace();
    }
    catch (S3ControlException e) {
        // Unknown exceptions will be thrown as an instance of this type.
        e.printStackTrace();
    }
    catch (SdkClientException e) {
        // Amazon S3 on Outposts couldn't be contacted for a response, or the
client
        // couldn't parse the response from Amazon S3 on Outposts.
        e.printStackTrace();
    }
}
}
```

### Example– デュアルスタックのエンドポイントを有効にして S3OutpostsClient を作成する

```
import com.amazonaws.AmazonServiceException;
```

```
import com.amazonaws.SdkClientException;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.s3outposts.S3OutpostsClient;
import software.amazon.awssdk.services.s3outposts.model.ListEndpointsRequest;
import software.amazon.awssdk.services.s3outposts.model.ListEndpointsResponse;
import software.amazon.awssdk.services.s3outposts.model.S3OutpostsException;

public class DualStackEndpointsExample2 {

    public static void main(String[] args) {
        Region clientRegion = Region.of("us-east-1");

        try {
            // Create an S3OutpostsClient with dual-stack endpoints enabled.
            S3OutpostsClient s3OutpostsClient = S3OutpostsClient.builder()
                .region(clientRegion)
                .dualstackEnabled(true)
                .build();

            ListEndpointsRequest listEndpointsRequest =
                ListEndpointsRequest.builder().build();

            ListEndpointsResponse listEndpoints =
                s3OutpostsClient.listEndpoints(listEndpointsRequest);
            System.out.printf("ListEndpoints Response: %s\n",
                listEndpoints.toString());
        } catch (AmazonServiceException e) {
            // The call was transmitted successfully, but Amazon S3 on Outposts
            // couldn't process
            // it, so it returned an error response.
            e.printStackTrace();
        }
        catch (S3OutpostsException e) {
            // Unknown exceptions will be thrown as an instance of this type.
            e.printStackTrace();
        }
        catch (SdkClientException e) {
            // Amazon S3 on Outposts couldn't be contacted for a response, or the
            // client
            // couldn't parse the response from Amazon S3 on Outposts.
            e.printStackTrace();
        }
    }
}
```

Windows で AWS SDK for Java 2.x を使用している場合は、必要に応じて、次の Java 仮想マシン (JVM) のプロパティを設定します。

```
java.net.preferIPv6Addresses=true
```