



Manual do usuário

Amazon Linux 2023



Amazon Linux 2023: Manual do usuário

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

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é o Amazon Linux 2023?	1
Cadência de lançamento	1
Liberações principais e secundárias	2
Consumindo novos lançamentos	2
Política de suporte de longo prazo	2
Nomeação e controle de versão	3
Otimizações operacionais e de desempenho	4
Relacionamento com o Fedora	5
cloud-init personalizado	6
Atualizações e recursos de segurança	7
Gerenciar atualizações	8
Segurança na nuvem	8
Modos SELinux	8
Programa de conformidade	8
Padrão do servidor SSH	8
Principais características do OpenSSL 3	9
Serviço de redes	9
Pacotes principais do conjunto de ferramentas glibc, gcc, binutils	10
Ferramenta de gerenciamento de pacotes	11
Configuração do servidor SSH padrão	11
Funcionalidades descontinuadas	14
compat-Pacotes do	14
Funcionalidades obsoletas descontinuadas no AL1 e removidas do AL2	14
AMIs x86 (i686) de 32 bits	15
Substituição de aws-apitools-* por AWS CLI	15
systemd substitui upstart no AL2	16
Funcionalidade obsoleta e removida em AL2 AL2023	16
Pacotes x86 (i686) de 32 bits	17
aws-apitools-* substituído por AWS CLI	18
amazon-cloudwatch-agents substitui awslogs	18
Sistema de controle de revisão bzip	18
cgroup v1	19
Hotpatch de log4j (log4j-cve-2021-44228-hotpatch)	19
lsb_release e o pacote system-lsb-core	19

mcrypt	20
OpenJDK 7 (java-1.7.0-openjdk)	20
Python 2.7	20
rsyslog-opensslsubstitui rsyslog-gnutls	21
Serviço de Informações de Rede (NIS)/yp	21
Vários nomes de domínio em create-dhcp-options do Amazon VPC	21
Sun RPC no glibc	22
Impressão digital da chave OpenSSH no log audit	22
Vinculador ld.gold	22
ping6	22
Pacote ftp	23
Descontinuações no AL2023	24
Suporte ao runtime x86 (i686) de 32 bits	25
aspell	25
Berkeley DB (libdb)	25
cron	26
IMDSv1	26
pcre versão 1	26
System V init (sysvinit)	27
Descontinuação dos pacotes em fim da vida útil	27
Comparando AL2 e AL2 023	29
Pacotes adicionados, atualizados e removidos	30
Suporte para cada versão	30
Alterações de nomenclatura e controle de versão	30
Otimizações	30
Proveniente de vários upstreams	31
Serviço do sistema de rede	31
Gerenciador de pacote	31
Usar o cloud-init	31
Suporte gráfico para desktop	32
Compilador Triplet	32
Pacotes x86 (i686) de 32 bits	32
lsb_release e o pacote system-lsb-core	33
EPEL	33
axel- HTTP/FTP cliente	35
brotli e libbrotli: compressão	36

collectd: daemon de coleta de estatísticas	36
cpulimit	36
exim: agente de transferência de mensagens	36
fuse3: File System in Userspace (FUSE) v3	37
ganglia: sistema de monitoramento distribuído	37
git-lfs: controle de versão de arquivos grandes com o Git	37
haveged: uma fonte de entropia usando o algoritmo HAVEGE	37
inotify-tools: ferramentas de linha de comandos da inotify	37
iperf- Referência TCP/UDP de desempenho	38
jemalloc: implementação alternativa de malloc	38
libbsd: biblioteca de funções compatível com BSD	39
libserf: biblioteca de cliente HTTP	39
libzstd: biblioteca de compressão zstd	39
Servidor web de lighttpd	39
lshell: um shell restrito	39
monit: monitor de processos, arquivos, diretórios e dispositivos	40
nodejs	40
perl-Config-General	40
python2-lockfile: bloqueio de arquivos	41
python2-rsa: RSA pura em Python	41
python2-simplejson: rotinas JSON para Python 2	42
rkhunter: Rootkit Hunter	42
rssh: um shell restrito para uso com OpenSSH	42
sscg: gerador de certificados SSL autoassinados	42
stress: teste de estresse	43
stress-ng: teste de estresse	43
tmpwatch: remove arquivos com base na hora do último acesso	43
xmlstarlet: utilitários XML de linha de comandos	43
O Python 2.7 foi substituído pelo Python 3	43
Atualizações de segurança	44
SELinux	44
OpenSSL 3	45
IMDSv2	45
Remoção do hotpatch log4j (log4j-cve-2021-44228-hotpatch)	46
Atualizações determinísticas para estabilidade	46
gp3 como tipo de volume padrão do Amazon EBS	46

Hierarquia unificada de grupos de controle (cgroup v2)	47
Temporizadores systemd substituem cron	47
Melhoria da cadeia de ferramentas: gcc, binutils e glibc	48
Diário de systemd substitui rsyslog	48
Dependências de pacotes minimizadas	49
Alterações de pacotes para curl e libcurl	49
Guarda de Privacidade GNU (GNUPG)	50
Amazon Corretto como JVM padrão	50
AWS CLIV2	50
Inicialização preferencial e segura por UEFI	50
Alterações na configuração padrão do servidor SSH	51
O kernel muda de AL2023 AL2	51
IPv4 TTL	51
Alterações na configuração do kernel com foco na segurança	52
Outras alterações na configuração do kernel	56
Suporte ao sistema de arquivos do kernel	58
Alterações em /tmp	63
Alterações na AMI e na imagem de contêiner	63
Comparação da AMI do Amazon Linux 2 e AL2023	64
Comparação entre a AMI mínima do Amazon Linux 2 e o AL2023	97
Comparação de contêineres do Amazon Linux 2 e do AL2023	118
Comparação entre o AL1 e o AL2023	126
Suporte para cada versão	126
systemd substitui upstart como sistema init	127
Python 2.6 e 2.7 foram substituídos pelo Python 3	127
OpenJDK 8 como o JDK mais antigo	127
Mudanças no kernel em AL2 023 de AL1	127
Kernel Live Patching	127
Suporte ao sistema de arquivos do kernel	127
Alterações na configuração do kernel com foco na segurança	130
Outras alterações na configuração do kernel	133
Comparação entre AL1 e AL2023 AMI	133
Comparação entre a AMI do AL1 e do AL2023	168
Comparação de contêineres AL1 e AL2023	188
Requisitos do sistema	197
Requisitos de CPU para executar o AL2023	197

Requisitos de CPU ARM para AL2023	197
Requisitos da CPU x86-64 para AL2023	198
Requisitos de memória (RAM) para executar o AL2023	199
O kernel Linux da Amazon	200
Ciclo de vida do kernel	200
Atualizações do kernel	201
Sobreposição de versões do kernel entre distribuições	201
Manipulação de CVE	202
O que você deve fazer	202
Área de trabalho gráfica	203
Tópicos relacionados	203
Pacotes suplementares para Amazon Linux	204
O que são pacotes suplementares para Amazon Linux (ou SPAL)?	204
Benefícios	204
Support of SPAL packages	205
Relatando problemas relacionados ao pacote	205
Tópicos relacionados	206
Configurar o SPAL em 023 AL2	206
Pré-requisitos	207
Verificando os pré-requisitos	207
Instalando o SPAL em seu sistema	207
Instalando pacotes SPAL	209
Desinstalando o repositório SPAL do seu sistema	210
Tópicos relacionados	210
Executar aplicações	211
Controle de recursos com systemd	211
Controle de recursos com systemd-run para execução de comandos únicos	211
Controle de recursos em um serviço de systemd	214
Usar utilitários cgroups	218
Usando AL2023 em AWS	221
Começando com AWS	221
Inscreva-se para um Conta da AWS	221
Criar um usuário com acesso administrativo	222
Conceder acesso programático	223
AL2023 no Amazon EC2	225
Inicializar o AL2023 usando o console do Amazon EC2	226

Lançamento AL2023 usando o parâmetro SSM e AWS CLI	227
Iniciando a AL2023 AMI mais recente usando CloudFormation	228
Lançamento AL2023 usando uma ID de AMI específica	230
AL2023 Depreciação e ciclo de vida da AMI	230
Conexão com AL2023 instâncias	231
Comparando AL2023 padrão (padrão) e mínimo AMIs	231
AL2023 em contêineres	259
AL2023 Imagem do contêiner base	260
AL2023 Imagem mínima do contêiner	262
Criando imagens básicas de contêineres AL2023	264
AL2023 comparação da lista de pacotes de imagens de contêiner	268
AL2023 AMI mínima em comparação com imagens de contêiner	274
AL2023 no Elastic Beanstalk	291
AL2023 CloudShell	291
AL2023 para hosts de contêineres do Amazon ECS	292
Alterações relevantes para o Amazon ECS em relação ao AL2	292
AMI otimizadas para Amazon ECS	293
Amazon EFS no AL2023	293
amazon-efs-utils	294
Montar o sistema de arquivos do Amazon EFS	294
Amazon EMR ativado AL2023	295
AL2023 lançamentos baseados no Amazon EMR	295
AL2023 baseado no Amazon EMR no EKS	295
AL2023 em AWS Lambda	295
Runtime provided.al2023 do Lambda	295
AL2023 tempos de execução baseados	296
Tutoriais	297
Instale o LAMP em AL2023	297
Etapa 1: Preparar o servidor LAMP	298
Etapa 2: Testar o servidor LAMP	303
Etapa 3: Proteger o servidor do banco de dados	305
Etapa 4: Instalação (opcional) phpMyAdmin	306
Solução de problemas	309
Tópicos relacionados	310
Configurar SSL/TLS em AL2023	310
Pré-requisitos	312

Etapa 1: habilitar o TLS no servidor	313
Etapa 2: obter um certificado assinado por uma CA	316
Etapa 3: testar e intensificar a configuração de segurança	324
Solução de problemas	328
Hospede um WordPress blog no AL2023	329
Pré-requisitos	330
Instalar WordPress	330
Próximas etapas	340
Ajuda! Meu nome DNS público mudou e agora meu blog não está funcionando	342
Transição do Redis 6 para o Valkey em 023 AL2	343
Cronograma de suporte para o Redis 6	343
Introdução ao Valkey	343
Plano e cronograma de migração	344
Opções e etapas de migração	344
Tópicos relacionados	347
Instale o GNOME em AL2023	347
Pré-requisitos	348
Instalação	348
Tópicos relacionados	348
Configurar o VNC em AL2023	349
Pré-requisitos	349
Etapa 1: instalação	349
Etapa 2: configuração	350
Etapa 3: conectar-se usando um cliente VNC	351
(Opcional) Iniciar o serviço na inicialização	352
(Opcional) Desabilitar a tela de bloqueio de inatividade	353
Tópicos relacionados	353
Usando o Multi-Gen LRU (MGLRU) em kernels AL2023	353
Configuração e ajuste	354
AL2023 fora do Amazon EC2	356
Baixe imagens da VM AL2023	356
Configurações compatíveis	356
Requisitos de KVM	357
Requisitos na VMware	359
Requisitos no Hyper-V	361
Configuração da VM AL2023	363

Configuração baseada em NoCloud <code>seed.iso</code>	364
Configuração baseada guestinfo da VMware	368
Comparação da lista de pacotes do AL2023 para a AMI padrão e a imagem para KVM	370
Comparação da lista de pacotes do AL2023 para a AMI padrão e a imagem para VMware OVA	395
Comparação da lista de pacotes do AL2023 para a AMI padrão e a imagem para Hyper-V	420
Identificar versões do Amazon Linux	446
/etc/os-release	446
Principais diferenças	447
Tipos de campos	447
/etc/os-releaseExemplos da	449
Comparação com outras distribuições	450
Específicos para Amazon Linux	452
/etc/sytem-release	453
/etc/image-id	453
Exemplos específicos do Amazon Linux	454
Código de exemplo	456
Layout do sistema de arquivos	469
/	469
/boot	470
/boot/efi	470
/etc	470
/home	470
/root	471
/srv	471
/tmp	472
/run	473
/usr	473
/usr/bin	474
/usr/include	474
/usr/lib e da /usr/lib64	474
/usr/local	474
/usr/share	474
/var	475
/var/cache	475
/var/lib	475

/var/log	475
/var/spool	476
/var/tmp	476
Atualizando AL2023	477
Práticas recomendadas para implantar atualizações de forma segura	477
Preparar-se para atualizações secundárias	481
Preparar-se para atualizações principais	481
Receber notificações sobre novas atualizações	482
Atualizações determinísticas por meio de repositórios com versionamento	483
Controle as atualizações recebidas de versões principais e secundárias	483
Diferenças entre atualizações de versão menor e principal	484
Como saber quando há atualizações disponíveis	484
Controle as atualizações de pacotes disponíveis nos AL2023 repositórios	485
Substituição de instâncias	485
Atualizações determinísticas no local	486
Gerenciar atualizações	494
Verificar as atualizações de pacotes disponíveis	495
Aplicar atualizações de segurança usando DNF e versões do repositório	499
Reinício automático do serviço após atualizações (de segurança)	512
Quando é necessário reinicializar para aplicar as atualizações de segurança?	513
Lançamento de uma instância com a versão mais recente do repositório ativada	514
Obtendo informações de suporte do pacote	515
dnf check-release-update	515
Adicionar, habilitar ou desabilitar novos repositórios	519
Adicionando repositórios com cloud-init	522
Kernel Live Patching	523
Limitações	524
Configurações e pré-requisitos compatíveis	525
Trabalhar com o Kernel Live Patching	525
Atualizações do kernel	530
Versões do kernel Linux em AL2023	531
Atualizando AL2023 para uma versão mais recente do kernel	531
AL2023 kernels - Perguntas frequentes	536
Linguagens de programação e tempos de execução	538
C/C++ e Fortran	538
GCC 14	540

Comparação entre versões padrão da linguagem	541
Go	542
AL2023 Função Lambda: Go	542
Java	543
Node.js	40
Perl	544
Perl módulos	545
PHP	545
Migrando para novas versões PHP	545
Migrando de PHP 7.x	546
PHP módulos	546
Python	546
Python módulos	547
Ruby	547
Rust	549
AL2023 Função Lambda: Rust	550
TypeScript	550
Usuários e grupos reservados do AL2023	555
Lista de usuários reservados do AL2023	555
Lista de grupos reservados do AL2023	563
Codecs disponíveis em 023 AL2	576
Segurança e conformidade	579
Recomendações de segurança	580
Anúncios do ALAS	580
INFELIZMENTE FAQs	581
Recomendações do ALAS	581
Recomendações e repositórios RPM	581
Consultivo IDs	582
Data de lançamento da recomendação e data de atualização da recomendação	583
Tipos de recomendação	583
Gravidades de recomendações	583
Recomendações e pacotes	584
Avisos e CVEs	585
Texto da recomendação	586
Recomendações de patch ao vivo do kernel	586
Esquema updateinfo.xml	587

Listar recomendações aplicáveis	588
Atualizações no local	592
Aplicar as atualizações mencionadas em uma recomendação	592
Configurando modos do SELinux para o AL2023	596
Status e modos padrão do SELinux para o AL2023	596
Mudar para o modo enforcing	597
Opção para desativar o SELinux	598
Habilitar o modo FIPS no AL2023	600
Habilitar o modo FIPS em um contêiner do AL2023	602
Trocar de provedor de OpenSSL FIPS no AL2023	604
Endurecimento do kernel	605
Opções de fortalecimento do kernel (independente da arquitetura)	606
Opções de fortalecimento de kernel específicas do x86-64	622
Opções de endurecimento de kernel específicas do aarch64	625
UEFI Secure Boot no AL2023	627
Habilitar UEFI Secure Boot no AL2023	627
Inscrição de uma instância existente	628
Registrar imagem do instantâneo	629
Atualizações de revogação	630
Como funciona o UEFI Secure Boot no AL2023	630
Inscrevendo suas próprias chaves	631
.....	dcxxxii

O que é o Amazon Linux 2023?

O Amazon Linux 2023 (AL2023) é a próxima geração do Amazon Linux da Amazon Web Services (AWS). Com o AL2023, você pode desenvolver e executar aplicações em nuvem e corporativas em um ambiente de runtime seguro, estável e de alto desempenho. Você também obtém um ambiente de aplicação que oferece suporte a longo prazo com acesso às mais recentes inovações no Linux. O AL2023 é fornecido sem nenhum custo adicional.

O AL2023 é o sucessor do Amazon Linux 2 (AL2). Para obter informações sobre as diferenças entre o AL2023 e o AL2, consulte [Comparando AL2 e AL2 023](#) e [Package changes in AL2023](#).

Tópicos

- [Cadência de lançamento](#)
- [Nomeação e controle de versão](#)
- [Otimizações operacionais e de desempenho](#)
- [Relacionamento com o Fedora](#)
- [cloud-init personalizado](#)
- [Atualizações e recursos de segurança](#)
- [Serviço de redes](#)
- [Pacotes principais do conjunto de ferramentas glibc, gcc, binutils](#)
- [Ferramenta de gerenciamento de pacotes](#)
- [Configuração do servidor SSH padrão](#)

Cadência de lançamento

O Amazon Linux 2023 (AL2023) foi lançado em março de 2023 e receberá suporte até 30 de junho de 2029. Há duas fases de suporte:

- Suporte padrão: durante esta fase, a versão recebe atualizações trimestrais de versões secundárias. A fase de suporte padrão termina em 30 de junho de 2027.
- Manutenção: durante esta fase, a versão recebe somente atualizações de segurança e correções de bugs críticos. Essas atualizações são publicadas assim que estão disponíveis. A fase de manutenção termina em 30 de junho de 2029.

Liberações principais e secundárias

A cada lançamento do Amazon Linux (versão principal, versão secundária ou lançamento de segurança), lançamos uma nova imagem de máquina da Amazon (AMI) Linux.

- **Versão principal:** inclui novos recursos e melhorias em segurança e desempenho em toda a pilha. As melhorias podem incluir grandes mudanças no kernel, no conjunto de ferramentas, Glib C, OpenSSL e em quaisquer outras bibliotecas e utilitários do sistema. Os principais lançamentos do Amazon Linux são baseados em parte na versão atual da distribuição upstream do Fedora Linux. AWS pode adicionar ou substituir pacotes específicos de outros upstreams que não sejam do Fedora.
- **Liberação de uma versão secundária:** uma atualização trimestral que inclui atualizações de segurança, correções de bugs e novos recursos e pacotes. Cada versão secundária é uma lista cumulativa de atualizações que inclui correções de segurança e bugs, além de novos recursos e pacotes. Essas versões podem incluir os tempos de execução de idiomas mais recentes, como PHP. Eles também podem incluir outros pacotes de software populares, como Ansible e Docker.

Consumindo novos lançamentos

As atualizações são fornecidas por meio de uma combinação de novas versões de imagem de máquina da Amazon (AMI) e dos novos repositórios correspondentes. Por padrão, uma nova AMI e o repositório para o qual ela aponta são acoplados. No entanto, você pode direcionar suas instâncias do Amazon EC2 em execução para versões mais recentes do repositório ao longo do tempo para aplicar atualizações nas instâncias em execução. Você também pode atualizar lançando novas instâncias das AMIs mais recentes.

Política de suporte de longo prazo

O Amazon Linux fornece atualizações para todos os seus pacotes e mantém a compatibilidade em uma versão principal para seus aplicativos criados no Amazon Linux. Os pacotes principais, como a biblioteca glibc, OpenSSL, OpenSSH e o gerenciador de pacotes DNF, recebem suporte durante a vida útil da versão principal do AL2023. Pacotes que não fazem parte dos pacotes principais são compatíveis com base em suas fontes upstream específicas. Você pode ver o status de suporte específico e as datas de pacotes individuais executando o comando a seguir.

```
$ sudo dnf supportinfo --pkg packagename
```

Você pode obter informações sobre todos os pacotes atualmente instalados executando o comando a seguir.

```
$ sudo dnf supportinfo --show installed
```

A lista completa dos pacotes principais é finalizada durante a pré-visualização. Se você quiser ver mais pacotes incluídos como pacotes principais, conte-nos. Avaliamos à medida que coletamos feedback. O feedback sobre o AL2023 pode ser fornecido por meio de seu representante AWS designado ou registrando um problema no repositório [amazon-linux-2023](https://github.com/amazon-linux-2023) no GitHub.

Nomeação e controle de versão

O AL2023 fornece um lançamento secundário a cada três meses durante os dois anos de suporte padrão. Cada versão é identificada por um incremento de 0 a N. 0 se refere à versão principal original dessa iteração. Todos os lançamentos serão chamados de Amazon Linux 2023. Quando a próxima versão do Amazon Linux for lançada, o AL2023 entrará no suporte estendido e receberá atualizações de segurança e correções de bugs críticos.

Por exemplo, versões menores do AL2023 têm o seguinte formato:

- 2023.**0**.20230301
- 2023.**1**.20230601
- 2023.**2**.20230901

As AMIs AL2023 correspondentes têm o seguinte formato:

- al2023-ami-2023.**0**.20230301.0-kernel-6.1-x86_64
- al2023-ami-2023.**1**.20230601.0-kernel-6.1-x86_64
- al2023-ami-2023.**2**.20230901.0-kernel-6.1-x86_64

Em uma versão secundária específica, os lançamentos regulares da AMI ocorrem com um timestamp da data do lançamento da AMI.

- al2023-ami-2023.0.**20230301**.0-kernel-6.1-x86_64
- al2023-ami-2023.0.**20230410**.0-kernel-6.1-x86_64
- al2023-ami-2023.0.**20230520**.0-kernel-6.1-x86_64

O método recomendado para identificar uma instância do AL2 ou do AL2023 começa com a leitura da string de enumeração de plataforma comum (CPE) de `/etc/system-release-cpe`. Em seguida, divida a string em seus campos. Por fim, leia os valores da plataforma e da versão.

O AL2023 também introduz novos arquivos para identificação da plataforma:

- `/etc/amazon-linux-release` symlinks para `/etc/system-release`
- `/etc/amazon-linux-release-cpe` symlinks para `/etc/system-release-cpe`

Esses dois arquivos indicam que uma instância é Amazon Linux. Não é necessário ler um arquivo ou dividir a string em campos, a menos que você queira saber os valores específicos da plataforma e da versão.

Otimizações operacionais e de desempenho

Kernel Amazon Linux 6.1

- O AL2023 usa os drivers mais recentes para dispositivos Adaptador de Rede Elástica (ENA) e Elastic Fabric Adapter (EFA). O AL2023 enfoca backports de funcionalidade e desempenho para hardware na infraestrutura do Amazon EC2.
- O kernel live patching está disponível para os tipos de instância `x86_64` e `aarch64`. Isso reduz a necessidade de reinicializar com frequência.
- Todas as configurações de compilação e runtime do kernel incluem muitas das mesmas otimizações operacionais e de desempenho do AL2.

Seleção do conjunto de ferramentas básico e sinalizadores de construção padrão

- Os pacotes do AL2023 são desenvolvidos com otimizações de compilador (`-O2`) habilitadas por padrão.
- Os pacotes AL2023 são criados exigindo `x86-64v2` para sistemas `x86-64` (`-march=x86-64-v2`) e Graviton 2 ou superior para `aarch64` (`-march=armv8.2-a+crypto -mtune=neoverse-n1`).
- Os pacotes AL2023 são criados com a vetorização automática ativada (`-ftree-vectorize`).
- Os pacotes AL2023 são criados com o Link Time Optimization (LTO) ativado.
- O AL2023 usa as versões atualizadas de Rust, Clang/LLVM e Go.

Seleção e versões de pacotes

- Alguns backports para os principais componentes do sistema incluem várias melhorias de desempenho para execução na infraestrutura do Amazon EC2, especialmente nas instâncias do Graviton.
- O AL2023 é integrado com vários Serviços da AWS e recursos. Isso inclui a AWS CLI, SSM Agent, o Amazon Kinesis Agent e o CloudFormation.
- O AL2023 usa o Amazon Corretto como Java Development Kit (JDK).
- O AL2023 fornece mecanismos de banco de dados e atualizações de tempo de execução da linguagem de programação para versões mais recentes à medida que são lançadas por projetos upstream. Os tempos de execução da linguagem de programação com novas versões são adicionados quando são lançados.

Implantação em um ambiente de nuvem

- A AMI básica do AL2023 e as imagens de contêiner são atualizadas com frequência para oferecer suporte à substituição de instâncias de patches.
- As atualizações do kernel estão incluídas nas atualizações da AMI AL2023. Isso significa que você não precisa usar comandos como `yum update` e `reboot` para atualizar seu kernel.
- Além da AMI AL2023 padrão, uma AMI mínima e uma imagem de contêiner também estão disponíveis. Escolha a AMI mínima para executar um ambiente com o número mínimo de pacotes necessários para executar seu serviço.
- Por padrão, as AMIs e os contêineres do AL2023 estão bloqueados em uma versão específica dos repositórios de pacotes. Não há atualização automática quando eles são lançados. Isso significa que você está sempre no controle de quando ingere qualquer atualização de pacote. Você sempre pode testar em um ambiente beta/gama antes de começar a produção. Se houver algum problema, você pode usar o caminho de reversão pré-validado.

Relacionamento com o Fedora

O AL2023 mantém seus próprios ciclos de vida de lançamento e suporte independentes do Fedora. O AL2023 fornece versões atualizadas de software de código aberto, uma variedade maior de pacotes e lançamentos frequentes. Isso preserva os conhecidos sistemas operacionais baseados em RPM.

A versão Generally Available (GA) do AL2023 não é diretamente comparável a nenhuma versão específica do Fedora. A versão AL2023 GA inclui componentes do Fedora 34, 35 e 36. Alguns dos componentes são iguais aos do Fedora e alguns são modificados. Outros componentes se assemelham mais aos componentes do CentOS Stream 9 ou foram desenvolvidos de forma independente. O kernel Amazon Linux é originado das opções de suporte de longo prazo que estão no kernel.org, escolhidas independentemente do Fedora.

cloud-init personalizado

O pacote cloud-init é uma aplicação de código aberto que inicializa imagens Linux em um ambiente de computação em nuvem. Para obter mais informações, consulte a [documentação de cloud-init](#).

O AL2023 contém uma versão personalizada do cloud-init. Com cloud-init, você pode especificar o que ocorre em sua instância no momento da inicialização.

Ao inicializar uma instância, você poderá usar campos de dados de usuários para cloud-init. Isso significa que você pode usar imagens comuns da Amazon Machine (AMIS) para muitos casos de uso e configurá-las dinamicamente quando você inicia uma instância. O AL2023 também usa cloud-init para configurar a conta `ec2-user`.

AL2023 usa as ações cloud-init em `/etc/cloud/cloud.cfg.d` e `/etc/cloud/cloud.cfg`. Você pode criar seus próprios arquivos de ações cloud-init no diretório `/etc/cloud/cloud.cfg.d`. Cloud-init lê todos os arquivos desse diretório em ordem lexicográfica. Arquivos mais recentes substituem arquivos mais antigos. Quando cloud-init inicia uma instância, o pacote cloud-init executa as seguintes tarefas de configuração:

- Definir o local padrão
- Define o nome do host
- Analisa e manipula dados do usuário
- Gerenciar chaves SSH privadas de host.
- Adicionar as chaves SSH públicas de um usuário ao `.ssh/authorized_keys` para facilitar login e administração.
- Prepara os repositórios para gerenciamento de pacotes.
- Lidar com as ações de pacotes definidas nos dados do usuário.
- Executa scripts de usuário que estão nos dados do usuário
- Monta volumes de armazenamento de instâncias, se aplicável

- Por padrão, se o volume de armazenamento de instâncias `ephemeral0` estiver presente e contiver um sistema de arquivos válido, o volume de armazenamento de instâncias será montado em `/media/ephemeral0`. Caso contrário, ele não está montado.
- Por padrão, para os tipos de instância `m1.small` e `c1.medium`, todos os volumes de troca associados à instância são montados.
- É possível substituir a montagem do volume de armazenamento de instância padrão pela seguinte diretiva de `cloud-init`:

```
#cloud-config
mounts:
- [ ephemeral0 ]
```

Para obter mais informações sobre o controle sobre montagens, consulte [Montagens](#) na documentação do `cloud-init`.

- Quando uma instância é executada, os volumes de armazenamento de instâncias compatíveis com TRIM não são formatados. Antes de montar volumes de armazenamento de instâncias, você deve particionar e formatar volumes de armazenamento de instâncias.

Para obter mais informações, consulte o [Suporte a TRIM do volume de armazenamento de instâncias](#) no Guia do usuário do Amazon EC2.

- Ao iniciar suas instâncias, você pode usar o módulo `disk_setup` para particionar e formatar os volumes do seu armazenamento de instância.

Para obter mais informações, consulte [Configurações de Disco](#) na documentação do `cloud-init`.

Para obter informações sobre como usar o `cloud-init` com a SELinux, consulte [Use cloud-init para ativar o modo enforcing](#).

Para obter informações sobre formatos de dados do usuário `cloud-init`, consulte [Formatos de dados do usuário](#) na documentação `cloud-init`.

Atualizações e recursos de segurança

O AL2023 fornece muitas atualizações e soluções de segurança.

Tópicos

- [Gerenciar atualizações](#)

- [Segurança na nuvem](#)
- [Modos SELinux](#)
- [Programa de conformidade](#)
- [Padrão do servidor SSH](#)
- [Principais características do OpenSSL 3](#)

Gerenciar atualizações

Aplique atualizações de segurança usando DNF e as versões de repositórios. Para obter mais informações, consulte [Gerencie atualizações de pacotes e sistemas operacionais no AL2023](#).

Segurança na nuvem

A segurança é uma responsabilidade compartilhada entre a AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem. Para obter mais informações, consulte [Segurança e compatibilidade no Amazon Linux 2023](#).

Modos SELinux

Por padrão, o SELinux está habilitado e configurado para o modo permissivo no AL2023. No modo permissivo, as negações de permissão são registradas, mas não aplicadas.

As políticas do SELinux definem permissões para usuários, processos, programas, arquivos e dispositivos. Com o SELinux, é possível escolher uma das duas políticas. As políticas são direcionadas ou de segurança multinível (MLS).

Para obter mais informações sobre os modos e políticas do SELinux, consulte [Configurando modos do SELinux para o AL2023](#) e o [SELinux Project Wiki](#).

Programa de conformidade

Audidores independentes avaliam a segurança e a conformidade da AL2023 junto com vários programas de conformidade da AWS.

Padrão do servidor SSH

O AL2023 inclui o OpenSSH 8.7. Por padrão, o OpenSSH 8.7 `ssh-rsa` desativa o algoritmo de troca de chaves. Para obter mais informações, consulte [Configuração do servidor SSH padrão](#).

Principais características do OpenSSL 3

- O Certificate Management Protocol (CMP, RFC 4210) inclui CRMF (RFC 4211) e transferência HTTP (RFC 6712).
- Um cliente HTTPS ou HTTP em libcrypto é compatível com as ações GET e POST, redirecionamento, conteúdo simples e ASN.1 codificado, proxies e tempos limite.
- O EVP_KDF funciona com funções de derivação de chaves.
- As EVP_MAC API obras com MACs.
- Suporte TLS ao Linux Kernel.

Para obter mais informações, consulte o [Guia de migração do OpenSSL](#).

Serviço de redes

O projeto de código aberto `systemd-networkd` está amplamente disponível nas distribuições Linux modernas. O projeto usa uma linguagem de configuração declarativa semelhante ao resto da estrutura `systemd`. Seus principais tipos de arquivo de configuração são os arquivos `.network` e `.link`.

O `amazon-ec2-net-utils` pacote gera configurações específicas da interface no diretório `/run/systemd/network`. Essas configurações habilitam redes IPv4 e IPv6 em interfaces quando elas estão conectadas a uma instância. Essas configurações também instalam regras de roteamento de políticas que ajudam a garantir que o tráfego de origem local seja roteado para a rede por meio da interface de rede da instância correspondente. Essas regras garantem que o tráfego correto seja roteado pela interface de rede elástica (ENI) por meio dos endereços ou prefixos associados. Para obter mais informações sobre o uso da ENI, consulte [Using ENI](#) no Guia do usuário do Amazon EC2.

Você pode personalizar esse comportamento de rede colocando um arquivo de configuração personalizado no diretório `/etc/systemd/network` para substituir as configurações padrão contidas em `/run/systemd/network`.

A documentação do [systemd.network](#) descreve como o serviço `systemd-networkd` determina a configuração que se aplica a uma interface específica. Ele também gera nomes alternativos, conhecidos como `altnames`, para as interfaces compatíveis com ENI para refletir as propriedades de vários recursos da AWS. Essas propriedades de interface compatíveis com ENI são o campo

ENI ID e DeviceIndex do anexo ENI. Você pode se referir a essas interfaces usando suas propriedades ao usar várias ferramentas, como o comando `ip`.

Os nomes de interface das instâncias do AL2023 são gerados usando o esquema de nomenclatura de slots de `systemd`. Para obter mais informações, consulte [esquema nomenclatura do systemd.net](https://systemd.net/esquema-nomenclatura).

Além disso, o AL2023 usa o algoritmo de agendamento de transmissão de rede de gerenciamento ativo de filas `fq_codel` por padrão. Para mais informações, consulte a [visão geral do CoDel](#).

Pacotes principais do conjunto de ferramentas glibc, gcc, binutils

Um subconjunto de pacotes no Amazon Linux é designado como pacotes principais da cadeia de ferramentas. Como parte principal AL2023, os pacotes principais recebem cinco anos de suporte. Podemos alterar a versão de um pacote, mas o suporte de longo prazo se aplica ao pacote incluído na versão do Amazon Linux.

Esses três pacotes principais fornecem uma cadeia de ferramentas do sistema que é usada para criar a maioria dos softwares na distribuição Amazon Linux.

Pacote	Definição	Finalidade
glibc 2.34	Biblioteca C do sistema	Usado pela maioria dos programas binários que fornecem funções padrão e pela interface entre os programas e o kernel.
gcc 11.2	Suíte de compiladores gcc	Compila C, C++, Fortran.
binutils 2.35	Assembler e vinculador, além de outras ferramentas binárias	Manipula ou inspeciona programas binários.

Recomendamos que atualizações de bibliotecas glibc sejam reinicializadas após atualizações de bibliotecas. Para atualizações de pacotes que controlam serviços, pode ser suficiente reiniciar os serviços para obter as atualizações. No entanto, a reinicialização do sistema garante que todas as atualizações anteriores de pacotes e bibliotecas sejam concluídas.

Ferramenta de gerenciamento de pacotes

A ferramenta padrão de gerenciamento de pacotes de software no AL2023 é DNF. O DNF é o sucessor de YUM, a ferramenta de gerenciamento de pacotes no AL2.

DNF é semelhante ao YUM em seu uso. Muitos comandos e opções de comandos DNF são iguais aos comandos YUM. Em uma interface de linha de comandos (CLI), na maioria dos casos `dnf` substitui `yum`.

Por exemplo, para os seguintes comandos `yum` do AL2:

```
$ sudo yum install packagename
$ sudo yum search packagename
$ sudo yum remove packagename
```

No AL2023, eles se tornam os seguintes comandos:

```
$ sudo dnf install packagename
$ sudo dnf search packagename
$ sudo dnf remove packagename
```

No AL2023, o comando `yum` ainda está disponível, mas como um ponteiro para o comando `dnf`. Portanto, quando o comando `yum` é usado no shell ou em um script, todos os comandos e opções são iguais ao DNF CLI. Para obter mais informações sobre as diferenças entre o YUM CLI e o DNF CLI, consulte [Alterações em DNF CLI comparado com YUM](#).

Para obter uma referência completa dos comandos e opções do comando `dnf`, consulte a página do manual `man dnf`. Para obter mais informações, consulte [DNF Command Reference](#).

Configuração do servidor SSH padrão

Se você tem clientes SSH de vários anos atrás, talvez veja um erro ao se conectar a uma instância. Se o erro indicar que não foi encontrado nenhum tipo de chave de host correspondente, atualize sua chave de host SSH para solucionar esse problema.

Desativação padrão de assinaturas **ssh-rsa**

O AL2023 inclui uma configuração padrão que desativa o algoritmo de chave de host `ssh-rsa` legado e gera um conjunto reduzido de chaves de host. Os clientes devem oferecer suporte ao

algoritmo da chave de host `ssh-ed25519` ou `ecdsa-sha2-nistp256` ao algoritmo da chave do host.

A configuração padrão aceita qualquer um desses algoritmos de troca de chaves:

- `curve25519-sha256`
- `curve25519-sha256@libssh.org`
- `ecdh-sha2-nistp256`
- `ecdh-sha2-nistp384`
- `ecdh-sha2-nistp521`
- `diffie-hellman-group-exchange-sha256`
- `diffie-hellman-group14-sha256`
- `diffie-hellman-group16-sha512`
- `diffie-hellman-group18-sha512`

Por padrão, o AL2023 gera chaves de hospedagem `ed25519` e `ECDSA`. Os clientes oferecem suporte ao algoritmo da chave de host `ssh-ed25519` ou `ecdsa-sha2-nistp256`. Ao se conectar por SSH a uma instância, você deve usar um cliente que ofereça suporte a um algoritmo compatível, como `ssh-ed25519` ou `ecdsa-sha2-nistp256`. Se você precisar usar outros tipos de chave, substitua a lista de chaves geradas por um fragmento de `cloud-config` nos dados do usuário.

No exemplo a seguir, `cloud-config` gera uma chave de host `rsa` com as chaves `ecdsa` e `ed25519`.

```
#cloud-config
ssh_genkeytypes:
- ed25519
- ecdsa
- rsa
```

Se você usa um par de chaves RSA para autenticação de chave pública, seu cliente SSH deve oferecer suporte a uma assinatura `rsa-sha2-256` ou `rsa-sha2-512`. Se você estiver usando um cliente incompatível e não conseguir fazer o upgrade, reative o suporte de `ssh-rsa` na sua instância. Para reabilitar o suporte a `ssh-rsa`, ative a política de criptografia do sistema `LEGACY` usando os seguintes comandos.

```
$ sudo dnf install crypto-policies-scripts
```

```
$ sudo update-crypto-policies --set LEGACY
```

Para obter mais informações sobre o gerenciamento de chaves de host, consulte [Amazon Linux Host keys](#).

Funcionalidades descontinuadas no AL2023

As funcionalidades descontinuadas no AL2 e que não estão no AL2023 estão documentadas aqui. Essas funcionalidades incluem recursos e pacotes que estão no AL2, mas não no AL2023, e que não serão adicionadas ao AL2023. Para obter mais informações sobre por quanto tempo as funcionalidades receberão suporte no AL2, consulte [Deprecated functionality in AL2](#).

Há também funcionalidades no AL2023 que foram descontinuadas e serão removidas em um lançamento futuro. Este capítulo descreve quais funcionalidades são essas, quando deixarão de receber suporte e quando serão removidas do Amazon Linux. Conhecer as funcionalidades descontinuadas ajudará você a implantar o AL2023, bem como a se preparar para a próxima versão principal do Amazon Linux.

Tópicos

- [compat-Pacotes do](#)
- [Funcionalidades obsoletas descontinuadas no AL1 e removidas do AL2](#)
- [Funcionalidade obsoleta e removida em AL2 AL2023](#)
- [Descontinuações no AL2023](#)

compat-Pacotes do

Todos os pacotes no AL2 com o prefixo `compat-` são fornecidos para compatibilidade com binários mais antigos que ainda não foram recompilados para as versões modernas do pacote. Novas versões principais do Amazon Linux não transferirão nenhum pacote `compat-` de lançamentos anteriores.

Todos os pacotes `compat-` de um lançamento do Amazon Linux (por exemplo, AL2) são descontinuados e não são disponibilizados na versão subsequente (por exemplo, AL2023). É altamente recomendável que o software seja recompilado com base nas versões atualizadas das bibliotecas.

Funcionalidades obsoletas descontinuadas no AL1 e removidas do AL2

Esta seção descreve as funcionalidades que estão disponíveis no AL1 e que não estão mais disponíveis no AL2.

Note

Como parte da fase de suporte para manutenção do AL1, alguns pacotes tiveram uma data de fim da vida útil (EOL) anterior ao EOL do AL1. Para obter mais informações, consulte [AL1 Package support statements](#).

Note

Algumas funcionalidades do AL1 foram descontinuadas em lançamentos anteriores. Para obter informações, consulte [AL1 Release Notes](#).

Tópicos

- [AMIs x86 \(i686\) de 32 bits](#)
- [Substituição de `aws-apitools-\*` por AWS CLI](#)
- [systemd substitui upstart no AL2](#)

AMIs x86 (i686) de 32 bits

Como parte do [lançamento 2014.09 do AL1](#), o Amazon Linux anunciou que seria o último lançamento a produzir AMIs de 32 bits. Portanto, a partir do [lançamento 2015.03 do AL1](#), o Amazon Linux não é mais compatível com a execução do sistema no modo de 32 bits. O AL2 oferece suporte de runtime limitado para binários de 32 bits em hosts x86-64 e não fornece pacotes de desenvolvimento para permitir o desenvolvimento de novos binários de 32 bits. O AL2023 não inclui mais nenhum pacote de espaço do usuário de 32 bits. Recomendamos que os usuários concluam a transição para o código de 64 bits antes de migrar para o AL2023.

Se você precisar executar binários de 32 bits no AL2023, poderá usar o espaço do usuário de 32 bits do AL2 dentro de um contêiner do AL2 executado sobre o AL2023.

Substituição de **`aws-apitools-*`** por AWS CLI

Antes do lançamento da AWS CLI em setembro de 2013, a AWS disponibilizou um conjunto de utilitários de linha de comandos, implementados em Java, que permitiam aos usuários fazer chamadas de API do Amazon EC2. Essas ferramentas foram descontinuadas em 2015, com a

AWS CLI se tornando a forma preferida de interagir com as APIs do Amazon EC2 por meio da linha de comandos. O conjunto de utilitários de linha de comandos inclui os seguintes pacotes `aws-apitools-*`.

- `aws-apitools-as`
- `aws-apitools-cfn`
- `aws-apitools-common`
- `aws-apitools-ec2`
- `aws-apitools-elb`
- `aws-apitools-mon`

O suporte upstream aos pacotes `aws-apitools-*` terminou em março de 2017. Apesar da falta de suporte upstream, o Amazon Linux continuou a ser distribuído com alguns desses utilitários de linha de comandos, como `aws-apitools-ec2`, para fins de compatibilidade com versões anteriores aos usuários. A AWS CLI é uma ferramenta mais robusta e completa do que os pacotes `aws-apitools-*`, pois é mantida ativamente e fornece um meio de usar todas as APIs da AWS.

Os pacotes `aws-apitools-*` foram descontinuados em março de 2017 e não receberão mais atualizações. Todos os usuários de qualquer um desses pacotes devem migrar para a AWS CLI o quanto antes. Esses pacotes não estão presentes no AL2023.

O AL1 também forneceu os pacotes `aws-apitools-iam` e `aws-apitools-rds`, que foram descontinuados no AL1 e não estão presentes no Amazon Linux a partir do AL2.

systemd substitui upstart no AL2

O AL2 foi o primeiro lançamento do Amazon Linux a usar o sistema de inicialização `systemd`, substituindo `upstart` no AL1. Qualquer configuração específica de `upstart` deve ser alterada como parte da migração do AL1 para uma versão mais recente do Amazon Linux. Não é possível usar `systemd` no AL1, portanto, a mudança de `upstart` para `systemd` só pode ser feita como parte da migração para uma versão principal mais recente do Amazon Linux, como AL2 ou AL2023.

Funcionalidade obsoleta e removida em AL2 AL2023

Esta seção descreve a funcionalidade que está disponível AL2 e não está mais disponível no AL2023.

Tópicos

- [Pacotes x86 \(i686\) de 32 bits](#)
- [aws-apitools-*substituído por AWS CLI](#)
- [awslogsdescontinuado em favor do agente Amazon Logs unificado CloudWatch](#)
- [Sistema de controle de revisão bzd](#)
- [cgroup v1](#)
- [Hotpatch de log4j \(log4j-cve-2021-44228-hotpatch\)](#)
- [lsb_release e o pacote system-lsb-core](#)
- [mccrypt](#)
- [OpenJDK 7 \(java-1.7.0-openjdk\)](#)
- [Python 2.7](#)
- [rsyslog-opensslsubstitui rsyslog-gnutls](#)
- [Serviço de Informações de Rede \(NIS\)/yp](#)
- [Vários nomes de domínio em create-dhcp-options do Amazon VPC](#)
- [Sun RPC no glibc](#)
- [Impressão digital da chave OpenSSH no log audit](#)
- [Vinculador ld.gold](#)
- [ping6](#)
- [Pacote ftp](#)

Pacotes x86 (i686) de 32 bits

Como parte da [versão 2014.09 do AL1](#), anunciamos que seria a última versão a produzir 32 bits. Portanto, a partir da [versão 2015.03 do](#), o AL1 Amazon Linux não suporta mais a execução do sistema no modo de 32 bits. AL2 fornece suporte de tempo de execução limitado para binários de 32 bits em hosts x86-64 e não fornece pacotes de desenvolvimento para permitir a criação de novos binários de 32 bits. AL2023 não inclui mais nenhum pacote de espaço de usuário de 32 bits. Recomendamos que os clientes concluam a transição para o código de 64 bits.

Se você precisar executar binários de 32 bits AL2023, é possível usar o espaço de usuário de 32 bits de AL2 dentro de um AL2 contêiner executado em cima do. AL2023

aws-apitools-* substituído por AWS CLI

Antes do lançamento do AWS CLI em setembro de 2013, AWS disponibilizou um conjunto de utilitários de linha de comando, implementados em Java, que permitiam aos clientes fazer chamadas de API do Amazon EC2. Essas ferramentas foram descontinuadas em 2015, AWS CLI tornando-se a forma preferida de interagir com o Amazon EC2 APIs a partir da linha de comando. Isso inclui os pacotes `aws-apitools-*` a seguir.

- `aws-apitools-as`
- `aws-apitools-cfn`
- `aws-apitools-common`
- `aws-apitools-ec2`
- `aws-apitools-elb`
- `aws-apitools-mon`

O suporte upstream aos pacotes `aws-apitools-*` terminou em março de 2017. Apesar da falta de suporte upstream, o Amazon Linux continuou a fornecer alguns desses utilitários de linha de comandos (como `aws-apitools-ec2`) para fins de compatibilidade com versões anteriores para os clientes. AWS CLI É uma ferramenta mais robusta e completa do que os `aws-apitools-*` pacotes, pois é mantida ativamente e fornece um meio de usar tudo AWS APIs.

Os pacotes `aws-apitools-*` foram descontinuados em março de 2017 e não receberão mais atualizações. Todos os usuários de qualquer um desses pacotes devem migrar para o o AWS CLI assim que possível. Esses pacotes não estão presentes no AL2023.

awslogs descontinuado em favor do agente Amazon Logs unificado CloudWatch

O [awslogs](#) pacote está obsoleto AL2 e não está mais presente no. AL2023 Ele é substituído pelo [agente de CloudWatch registros unificado](#), disponível no `amazon-cloudwatch-agent` pacote. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Logs](#).

Sistema de controle de revisão **bzr**

O sistema de controle de revisão [GNU Bazaar](#) (`bzr`) foi descontinuado em AL2 e não está mais presente no AL2023.

Os usuários do `bzr` devem migrar seus repositórios para `git`.

cgroup v1

AL2023 passa para a hierarquia do Grupo de Controle Unificado (cgroup v2), enquanto AL2 usa cgroup v1. Como AL2 não é compatível com cgroup v2, essa migração precisa ser concluída como parte da mudança para o. AL2023

Hotpatch de log4j (**log4j-cve-2021-44228-hotpatch**)

Note

O `log4j-cve-2021-44228-hotpatch` pacote foi descontinuado AL2 e removido. AL2023

Em resposta ao [CVE-2021-44228](#), a Amazon Linux lançou uma versão empacotada em RPM do Hotpatch para Apache Log4j para [e](#). AL1 AL2 No [anúncio da adição do hotpatch ao Amazon Linux](#), observamos que “Instalar o hotpatch não substitui a atualização para uma versão de log4j que atenua a CVE-2021-44228 ou a CVE-2021-45046”.

O hotpatch foi uma mitigação para dar tempo de corrigir log4j. A primeira versão de disponibilidade geral do AL2023 foi 15 meses após o [CVE-2021-44228](#), portanto, AL2023 não vem com o hotpatch (ativado ou não).

Os clientes que executam suas próprias versões log4j no Amazon Linux são aconselhados a garantir que tenham atualizado para versões não afetadas pela [CVE-2021-44228](#) ou [CVE-2021-45046](#).

lsb_release e o pacote system-lsb-core

Historicamente, alguns softwares invocavam o `lsb_release` comando (fornecido AL2 pelo `system-lsb-core` pacote) para obter informações sobre a distribuição Linux na qual ele estava sendo executado. O Linux Standards Base (LSB) introduziu esse comando e as distribuições Linux o adotaram. As distribuições Linux evoluíram para usar o padrão mais simples de armazenar essas informações em `/etc/os-release` e outros arquivos relacionados.

O padrão `os-release` sai de `systemd`. Para obter mais informações, consulte a [documentação do systemd os-release](#).

AL2023 não vem com o `lsb_release` comando e não inclui o `system-lsb-core` pacote. O software deve concluir a transição para o padrão `os-release` para manter a compatibilidade com o Amazon Linux e outras grandes distribuições Linux.

mcrypt

A `mcrypt` biblioteca e a PHP extensão associada foram preteridas em AL2 e não estão mais presentes no. AL2023

O PHP upstream [descontinuou a extensão `mcrypt` no PHP 7.1](#), que foi lançado pela primeira vez em dezembro de 2016 e teve seu lançamento final em outubro de 2019.

A `mcrypt` biblioteca upstream foi [lançada pela última vez em 2007](#) e não fez a migração do controle de cvs revisão [SourceForge exigida para novos commits em 2017](#), com o commit mais recente (e apenas 3 anos antes) sendo de 2011, removendo a menção de que o projeto tinha um mantenedor.

Todos os usuários restantes do `mcrypt` são aconselhados a portar seu código para `OpenSSL`, pois não `mcrypt` será adicionado AL2023.

OpenJDK 7 (**java-1.7.0-openjdk**)

Note

AL2023 fornece várias versões do [Amazon Corretto para suportar cargas](#) de trabalho Java baseadas. Os pacotes do OpenJDK 7 estão obsoletos e não estão mais presentes AL2 no. AL2023 O JDK mais antigo disponível em AL2023 é fornecido pelo Corretto 8.

Para saber mais sobre Java no Amazon Linux, consulte [Java no AL2023](#).

Python 2.7

Note

AL2023 removeu o Python 2.7, portanto, todos os componentes do sistema operacional que exigem Python são escritos para funcionar com o Python 3. Para continuar usando uma versão do Python fornecida e compatível com o Amazon Linux, converta o código do Python 2 em Python 3.

Para saber mais sobre Python no Amazon Linux, consulte [Python no AL2023](#).

rsyslog-openssl substitui **rsyslog-gnutls**

O **rsyslog-gnutls** pacote está obsoleto em AL2 e não está mais presente no. AL2023 O pacote **rsyslog-openssl** deve ser um substituto imediato para qualquer uso do pacote **rsyslog-gnutls**.

Serviço de Informações de Rede (NIS)/yp

O Network Information Service (NIS), originalmente chamado de Páginas Amarelas ou YP está obsoleto e não está mais presente no AL2. AL2023 Isso inclui os seguintes pacotes: **ypbind**, **ypserv** e **yp-tools**. Outros pacotes que se integram ao NIS têm essa funcionalidade removida AL2023.

Vários nomes de domínio em **create-dhcp-options** do Amazon VPC

No Amazon Linux 2, era possível enviar vários nomes de domínio no parâmetro **domain-name** para [create-dhcp-options](#), o que fazia com que **/etc/resolv.conf** contivesse algo parecido com **search foo.example.com bar.example.com**. O servidor DHCP do Amazon VPC envia a lista de nomes de domínio fornecidos usando DHCP opção 15, que só oferece suporte a um único nome de domínio (consulte a [seção 3.17 da RFC 2132](#)). Como é AL2023 usado **systemd-networkd** para configuração de rede, que segue o RFC, esse recurso acidental em não AL2 está presente no AL2023

A [documentação do Amazon VPC](#) e da [AWS CLI](#) dizem o seguinte: “Alguns sistemas operacionais Linux aceitam vários nomes de domínio separados por espaços. No entanto, o Windows e outros sistemas operacionais Linux tratam o valor como um domínio único, o que resulta em um comportamento inesperado. Se o seu conjunto de opções DHCP estiver associado a uma Amazon VPC que contenha instâncias executando sistemas operacionais que tratam o valor como um domínio único, especifique somente um nome de domínio.

Nesses sistemas, por exemplo AL2023, especificar dois domínios usando a DHCP opção 15 (que permite apenas um), e como o [caractere de espaço é inválido em nomes de domínio](#), isso resultará na codificação do caractere de espaço como **032**, resultando em **conter. /etc/resolv.conf search foo.exmple.com032bar.example.com**

Para oferecer suporte a vários nomes de domínio, um servidor DHCP deve usar DHCP opção 119 (consulte a [seção 2 da RFC 3397](#)). Consulte o [Guia do usuário do Amazon VPC](#) para saber quando isso é compatível com o servidor DHCP do Amazon VPC.

Sun RPC no **glibc**

A implementação de Sun RPC in **glibc** foi descontinuada AL2 e removida em. AL2023 Recomenda-se que os clientes passem a usar a **libtirpc** biblioteca (disponível em AL2 e AL2023) se a Sun RPC funcionalidade for necessária. A adoção de **libtirpc** também permite que as aplicações ofereçam suporte a IPv6.

Esta alteração reflete a adoção pela comunidade em geral de **glibc** upstream, que remove esta funcionalidade, por exemplo, a [remoção das interfaces Sun RPC de **glibc** no Fedora](#) e uma [mudança semelhante no Gentoo](#).

Impressão digital da chave OpenSSH no log **audit**

Posteriormente no ciclo de vida do AL2, um patch foi adicionado ao pacote OpenSSH para emitir a impressão digital da chave usada para autenticar. Essa funcionalidade não está presente no AL2023.

Vinculador **ld.gold**

O **ld.gold** vinculador está disponível em AL2 e é removido em AL2023. Os clientes que desenvolvem software que faz referência explícita ao vinculador gold devem migrar para o vinculador regular (**ld.bfd**).

As [notas de lançamento da versão 2.44](#) do [GNU Binutils](#) upstream (lançada em fevereiro de 2025) documentam a remoção de **ld.gold**: “Em uma mudança em relação à nossa prática anterior, nesta versão, o tarball **binutils-2.44.tar** não contém as fontes para o vinculador gold. Isso ocorre porque o vinculador gold foi descontinuado e, eventualmente, será removido, a menos que voluntários se apresentem e se ofereçam para continuar o desenvolvimento e a manutenção.”

ping6

Em AL2023, o **ping** utilitário regular oferece suporte IPv6 nativo e o separado não **/bin/ping6** é mais necessário. In AL2023, **/usr/sbin/ping6** é um link simbólico para o **/usr/bin/ping** executável.

Essa mudança segue a adoção pela comunidade mais ampla de **iputils** versões mais recentes que fornecem essa funcionalidade, por exemplo, a [IPv6 mudança de Ping no Fedora](#).

Pacote **ftp**

O pacote `ftp` no AL2 não está mais disponível no Amazon Linux a partir do AL2023. Essa decisão foi tomada como parte de nosso compromisso contínuo com a segurança, a capacidade de manutenção e as práticas modernas de desenvolvimento de software. Como parte (ou antes) da migração para o AL2023, recomendamos migrar todo uso do pacote `ftp` legado para uma de suas alternativas.

Contexto

O pacote `ftp` legado não tem sido mantido ativamente upstream há muitos anos. A última atualização significativa do código-fonte ocorreu no início dos anos 2000, e o repositório original não está mais disponível. Embora algumas distribuições do Linux tenham aplicado patches para vulnerabilidades de segurança, a base de código permanece praticamente sem manutenção.

Alternativas recomendadas

O AL2023 fornece várias alternativas modernas e ativamente mantidas para a funcionalidade FTP:

`lftp` (disponível no AL2 e no AL2023)

Um sofisticado programa de transferência de arquivos que é compatível com FTP, HTTP, SFTP e outros protocolos. Ele oferece mais recursos do que o cliente `ftp` tradicional e é mantido ativamente.

Instale-o com o: `dnf install lftp`

`curl` (disponível no AL2 e no AL2023)

Uma versátil ferramenta de linha de comandos para transferência de dados com URLs, compatível com FTP, FTPS, HTTP, HTTPS e muitos outros protocolos.

Disponível por padrão no AL2023 por meio do pacote `curl-minimal`. Para obter um suporte mais amplo a protocolos, você pode atualizar para `curl-full` usando `dnf swap curl-minimal curl-full`.

`wget` (disponível no AL2 e no AL2023)

Um utilitário de linha de comandos não interativo para download de arquivos da web, compatível com os protocolos HTTP, HTTPS e FTP.

Instale com: `dnf install wget` (não instalado por padrão em todas as imagens do AL2023)

`sftp` (disponível no AL2 e no AL2023)

Um protocolo de transferência de arquivos seguro que opera por SSH, fornecendo transferências de arquivos criptografadas.

Disponível por padrão como parte do pacote OpenSSH.

Considerações sobre a migração

Se suas aplicações ou scripts dependerem do cliente `ftp` legado, considere as seguintes abordagens de migração:

1. Atualize os scripts para usar alternativas modernas: modifique seus scripts para usar `lftp`, `curl`, `wget` ou `sftp` em vez do cliente `ftp` legado.
2. Analise as dependências do pacote: algumas aplicações podem listar o pacote `ftp` como uma dependência nos metadados do pacote, mesmo que tenham migrado há muito tempo para o uso interno de protocolos modernos. Nesses casos, a aplicação pode funcionar corretamente no AL2023, apesar da falta de `/usr/bin/ftp` do pacote `ftp`. Analise os requisitos reais da sua aplicação em vez de confiar apenas nas dependências declaradas.
3. Atualize as dependências da aplicação: para aplicações que você mantém e que ainda declaram dependência em relação ao pacote `ftp`, mas não o usam de fato, atualize os metadados do pacote para remover essa dependência desnecessária.

Considerações sobre segurança

O protocolo FTP transmite dados, incluindo credenciais de autenticação, em texto simples. Para aplicações sensíveis à segurança, é altamente recomendável usar alternativas criptografadas, como SFTP ou HTTPS, que são compatíveis com as ferramentas alternativas recomendadas.

Descontinuações no AL2023

Esta seção descreve funcionalidades que existem no AL2023 e provavelmente serão removidas em uma versão futura do Amazon Linux. Cada seção descreverá a funcionalidade e quando se espera que ela seja removida do Amazon Linux.

 Note

Esta seção será atualizada ao longo do tempo, à medida que o ecossistema do Linux evoluir e as futuras versões principais do Amazon Linux estiverem mais próximas do lançamento.

Tópicos

- [Suporte ao runtime x86 \(i686\) de 32 bits](#)
- [aspell](#)
- [Berkeley DB \(libdb\)](#)
- [cron](#)
- [IMDSv1](#)
- [pcre versão 1](#)
- [System V init \(sysvinit\)](#)
- [Descontinuação dos pacotes em fim da vida útil](#)

Suporte ao runtime x86 (i686) de 32 bits

O AL2023 mantém a capacidade de executar binários x86 (i686) de 32 bits. É provável que a próxima versão principal do Amazon Linux não ofereça mais suporte à execução de binários de espaço de usuário de 32 bits.

aspell

Embora o AL2023 seja distribuído com o pacote `aspell`, ele foi descontinuado e será removido na próxima versão principal do Amazon Linux. Os clientes devem migrar para substitutos modernos, como `hunspell` ou `enchant2`.

A descontinuação de `aspell` no AL2023 segue uma mudança da comunidade em geral, como a [descontinuação de aspell no Fedora](#).

Berkeley DB (**libdb**)

O AL2023 é distribuído com a versão 5.3.28 da biblioteca Berkeley DB (`libdb`). Esta é a última versão do Berkeley DB antes da mudança da licença Sleepycat menos restritiva para a licença GNU Affero GPLv3 (AGPL).

Há alguns pacotes no AL2023 que permanecem dependentes do Berkeley DB (`libdb`), e a biblioteca será removida na próxima versão principal do Amazon Linux.

 Note

O gerenciador de pacotes `dnf` no AL2023 mantém suporte somente leitura para um banco de dados `rpm` no formato Berkeley DB (BDB). Esse suporte será removido na próxima versão principal do Amazon Linux.

A descontinuação de `libdb` segue uma mudança da comunidade em geral, como a [descontinuação de `libdb` no Fedora](#).

cron

O pacote `cronie` foi instalado por padrão na AMI AL2, fornecendo suporte para a forma `crontab` tradicional de programar tarefas periódicas. No AL2023, `cronie` não está incluído por padrão. Portanto, o suporte a `crontab` não é mais fornecido por padrão.

No AL2023, você também pode instalar o pacote `cronie` para usar trabalhos `cron` clássicos. Recomendamos que você migre para temporizadores `systemd` devido à funcionalidade adicional fornecida pelo `systemd`.

É possível que uma versão futura do Amazon Linux, possivelmente a próxima versão principal, não inclua mais suporte a trabalhos `cron` clássicos e conclua a transição para temporizadores `systemd`. Recomendamos que você deixe de usar `cron`.

IMDSv1

Por padrão, as AMIs do AL2023 são configuradas para serem inicializadas no modo somente IMDSv2, desabilitando o uso de IMDSv1. Ainda existe a opção de usar o AL2023 com o IMDSv1 habilitado. É provável que uma versão futura do Amazon Linux imponha somente IMDSv2.

Para obter mais informações sobre a configuração do IMDS para AMIs, consulte [Configurar a AMI](#) no Guia do usuário do Amazon EC2.

pcr versão 1

O pacote `pcr` legado foi descontinuado e será removido na próxima versão principal do Amazon Linux. O pacote `pcr2` é o sucessor. Embora as primeiras versões do AL2023 tenham sido

distribuídas com um número limitado de pacotes compilados em `pcre`, esses pacotes serão migrados para `pcre2` no AL2023. A biblioteca `pcre` descontinuada permanecerá disponível no AL2023.

Note

A versão descontinuada de `pcre` não receberá atualizações de segurança durante toda a vida útil do AL2023. Para obter mais informações sobre o ciclo de vida do suporte a `pcre` e a quantidade de tempo em que o pacote receberá atualizações de segurança, consulte as [declarações de suporte sobre o pacote `pcre`](#).

A descontinuação de `pcre` em favor de `pcre2` segue uma mudança da comunidade em geral nesse sentido, como a [descontinuação de `pcre` no Fedora](#).

System V init (**sysvinit**)

Embora o AL2023 mantenha a compatibilidade com versões anteriores dos scripts do serviço System V (init), o projeto de `systemd` upstream, como parte de seu [lançamento v254](#), anunciou a [descontinuação do suporte aos scripts do serviço System V](#) e indicou que o suporte será removido em uma versão futura do `systemd`. Para obter mais informações, consulte [systemd](#).

O AL2023 manterá a compatibilidade com versões anteriores dos scripts do serviço System V (init), mas os usuários são incentivados a migrar para o uso de arquivos de unidade nativos de `systemd` para estarem preparados para quando o suporte aos scripts do serviço System V (init) for removido do Amazon Linux, provavelmente na próxima versão principal.

Descontinuação dos pacotes em fim da vida útil

Cada pacote disponível no AL2023 tem uma [declaração de suporte](#) associada que abrange informações específicas do Amazon Linux. Essas declarações abrangem o núcleo do sistema operacional e sua vida útil, além de pacotes como [the section called “PHP”](#) e [the section called “Python”](#), em que o AL2023 é distribuído com diversas versões e cada uma recebe suporte pela duração do projeto de código aberto upstream.

No AL2023, você pode obter informações sobre o suporte aos pacotes usando o gerenciador de pacotes `dnf`. Para obter mais informações, consulte [Obtendo informações de suporte do pacote](#).

Quando um pacote deixa de receber suporte antes do final da versão principal do Amazon Linux, deve-se presumir que esse pacote foi descontinuado e não estará presente na próxima versão principal do Amazon Linux.

Para pacotes como [the section called “PHP”](#) e [the section called “Python”](#), em que cada versão principal do Amazon Linux foi distribuída com várias versões, cada uma com um ciclo de vida de suporte diferente, é provável que continuem presentes nas novas versões principais do Amazon Linux, embora com pouca ou nenhuma sobreposição das versões principais dos pacotes. É recomendável conhecer os cronogramas de suporte aos pacotes do Amazon Linux ao selecionar dependências.

Comparando AL2 e AL2 023

Os tópicos a seguir descrevem as principais diferenças entre AL2 e AL2 023.

Para obter mais informações sobre a funcionalidade obsoleta em AL1,, e AL2 AL2 023, consulte.

[Funcionalidades descontinuadas no AL2023](#)

Tópicos

- [Pacotes adicionados, atualizados e removidos](#)
- [Suporte para cada versão](#)
- [Alterações de nomenclatura e controle de versão](#)
- [Otimizações](#)
- [Proveniente de vários upstreams](#)
- [Serviço do sistema de rede](#)
- [Gerenciador de pacote](#)
- [Usar o cloud-init](#)
- [Suporte gráfico para desktop](#)
- [Compilador Triplet](#)
- [Pacotes x86 \(i686\) de 32 bits](#)
- [lsb_release e o pacote system-lsb-core](#)
- [Extra Packages for Enterprise Linux \(EPEL\)](#)
- [O Python 2.7 foi substituído pelo Python 3](#)
- [Atualizações de segurança](#)
- [Atualizações determinísticas para estabilidade](#)
- [gp3 como tipo de volume padrão do Amazon EBS](#)
- [Hierarquia unificada de grupos de controle \(cgroup v2\)](#)
- [Temporizadores systemd substituem cron](#)
- [Melhoria da cadeia de ferramentas: gcc, binutils e glibc](#)
- [Diário de systemd substitui rsyslog](#)
- [Dependências de pacotes minimizadas](#)
- [Amazon Corretto como JVM padrão](#)
- [AWS CLIV2](#)

- [Inicialização preferencial e segura por UEFI](#)
- [Alterações na configuração padrão do servidor SSH](#)
- [AL2023 o kernel muda de AL2](#)
- [/tmp é agora tmpfs](#)
- [Alterações na AMI e na imagem de contêiner](#)
- [Comparação de pacotes instalados nas AMIs Amazon Linux 2 e Amazon Linux 2023](#)
- [Comparação entre pacotes instalados no Amazon Linux 2 e nas AMIs do Amazon Linux 2023](#)
- [Compare de comparação de pacotes instalados em imagens de contêiner de base do Amazon Linux 2023 e do Amazon Linux 2023](#)

Pacotes adicionados, atualizados e removidos

AL2023 contém milhares de pacotes de software disponíveis para uso. Para obter uma lista completa de todos os pacotes adicionados, atualizados ou removidos em AL2 023 em comparação com as versões anteriores do Amazon Linux, consulte [Package changes in AL2 023](#).

Para solicitar que um pacote seja adicionado ou alterado em AL2 023, registre um problema no repositório [amazon-linux-2023](#) em. GitHub

Suporte para cada versão

Para AL2 2023, oferecemos cinco anos de suporte.

Para obter mais informações, consulte [Cadência de lançamento](#).

Alterações de nomenclatura e controle de versão

AL20 023 suporta os mesmos mecanismos que AL2 suportam a identificação da plataforma. AL20 023 também introduz novos arquivos para identificação da plataforma.

Para obter mais informações, consulte [Nomeação e controle de versão](#).

Otimizações

AL20 023 otimiza o tempo de inicialização para reduzir o tempo desde a inicialização da instância até a execução da carga de trabalho do cliente. Essas otimizações abrangem a configuração, as

cloud-init configurações e os recursos do kernel da EC2 instância Amazon que são incorporados aos pacotes no sistema operacional, como `e. kmod systemd`

Para obter mais informações sobre essas otimizações, consulte [Otimizações operacionais e de desempenho](#).

Proveniente de vários upstreams

AL20 023 é baseado em RPM e inclui componentes provenientes de várias versões do Fedora e de outras distribuições, como o CentOS 9 Stream. O kernel Amazon Linux é originado das versões de suporte de longo prazo (LTS) diretamente do kernel.org, escolhidas independentemente de outras distribuições.

Para obter mais informações, consulte [Relacionamento com o Fedora](#).

Serviço do sistema de rede

O serviço `systemd-networkd` do sistema gerencia as interfaces de rede em AL2 023. Esta é uma alteração de AL2, que usa ISC `dhclient` `oudhclient`.

Para obter mais informações, consulte [Serviço de redes](#).

Gerenciador de pacote

A ferramenta padrão de gerenciamento de pacotes de software no AL2 023 é DNF. DNF é a sucessora da ferramenta de gerenciamento de pacotes em AL2. YUM

Para obter mais informações, consulte [Ferramenta de gerenciamento de pacotes](#).

Usar o cloud-init

Em AL2 023, cloud-init gerencia o repositório de pacotes. Por padrão, em versões anteriores do Amazon Linux, cloud-init instalou atualizações de segurança. Esse não é o padrão para AL2 023. Os novos recursos determinísticos de atualização para atualização `releasetool` no lançamento descrevem a maneira AL2 023 de habilitar atualizações de pacotes no lançamento. Para obter mais informações, consulte [Gerencie atualizações de pacotes e sistemas operacionais no AL2023 e Atualizações determinísticas para estabilidade](#).

Com AL2 023, você pode usar cloud-init comSELinux. Para obter mais informações, consulte [Use cloud-init para ativar o modo enforcing](#).

Cloud-init carrega o conteúdo de configuração com cloud-init de locais remotos usando HTTP(S). Nas versões anteriores, o Amazon Linux não alertava você quando os recursos remotos não estão disponíveis. Em AL2 023, recursos remotos indisponíveis criam um erro fatal e falham na cloud-init execução. Essa mudança no comportamento de AL2, fornece um comportamento padrão mais seguro de “falha fechada”.

Para obter mais informações, consulte [cloud-init personalizado](#) e a [Documentação do cloud-init](#).

Suporte gráfico para desktop

AL20 023 apresenta um ambiente de desktop gráfico baseado em GNOME a partir da versão 2023.7, substituindo o desktop MATE usado em. AL2 Esta versão oferece aos usuários uma experiência de desktop diferente, mantendo o desempenho otimizado para nuvem do AL2 023. O ambiente de área de trabalho GNOME oferece várias opções de personalização, recursos de integração do sistema e um design de interface distinto, proporcionando aos usuários uma alternativa ao ambiente de área de trabalho MATE anterior. Consulte a página [AL2023 Desktop gráfico](#) para obter mais detalhes.

Compilador Triplet

AL2023 define o trigêmeo do compilador para GCC e LLVM para indicar que amazon é o fornecedor.

Assim, eles se AL2 aarch64-redhat-linux-gcc tornam aarch64-amazon-linux-gcc em AL2 023.

Isso deve ser completamente transparente para a maioria dos usuários e pode afetar apenas aqueles que estão criando compiladores no AL2 023.

Pacotes x86 (i686) de 32 bits

Como parte da [versão 2014.09 AL1](#), foi anunciado que seria a última versão a produzir 32 bits. AMIs Portanto, a partir da [versão 2015.03 do](#), o AL1 Amazon Linux não suportava mais a execução do sistema no modo de 32 bits. AL2 oferecia suporte de tempo de execução limitado para binários de 32 bits em hosts x86-64 e não fornecia pacotes de desenvolvimento para permitir a criação de novos binários de 32 bits. AL2023 não inclui mais nenhum pacote de espaço de usuário de 32 bits. Recomendamos que você conclua a transição para o código de 64 bits.

Se você precisar executar binários de 32 bits no AL2 023, é possível usar o espaço do usuário de 32 bits de AL2 dentro de um AL2 contêiner executado sobre o 023. AL2

lsb_release e o pacote system-lsb-core

Historicamente, alguns softwares invocavam o `lsb_release` comando (fornecido AL2 pelo `system-lsb-core` pacote) para obter informações sobre a distribuição Linux na qual ele estava sendo executado. O Linux Standards Base (LSB) introduziu esse comando e as distribuições Linux o adotaram. As distribuições Linux evoluíram para usar o padrão mais simples de armazenar essas informações em `/etc/os-release` e outros arquivos relacionados.

O padrão `os-release` sai de `systemd`. Para obter mais informações, consulte a [documentação do systemd os-release](#).

AL2023 não vem com o `lsb_release` comando e não inclui o `system-lsb-core` pacote. O software deve concluir a transição para o padrão `os-release` para manter a compatibilidade com o Amazon Linux e outras grandes distribuições Linux.

Extra Packages for Enterprise Linux (EPEL)

Warning

O AL2 epe1 Extra habilitou o EPEL7 repositório de terceiros. Desde 30/6/2024, o repositório EPEL7 de terceiros não é mais mantido.

Esse repositório de terceiros não receberá atualizações futuras. Isso significa que não haverá correções de segurança para pacotes no repositório EPEL.

Esta seção abordará as opções em AL2 023 para pacotes encontrados em EPEL.

Extra Packages for Enterprise Linux (EPEL) é um projeto na comunidade Fedora com o objetivo de criar uma grande variedade de pacotes para sistemas operacionais Linux de nível corporativo. O projeto foi produzido principalmente RHEL e CentOS embalado. AL2 apresenta um alto nível de compatibilidade com CentOS 7. Como resultado, muitos EPEL7 pacotes funcionam em AL2.

Não há EPEL versões binárias compatíveis com AL2 023. No entanto, os clientes que desejam usar seus EPEL7 pacotes no AL2 023 têm algumas opções. Alguns EPEL pacotes têm alternativas em AL2 023, enquanto outros são fornecidos como parte do [Pacotes suplementares para Amazon Linux](#).

 Warning

Adicione somente repositórios projetados para serem usados com AL2 023. Embora os repositórios projetados para outras distribuições possam funcionar atualmente, não há garantia de que continuarão funcionando com qualquer atualização de pacote no AL2 023 ou com o repositório não projetado para uso com o 023. AL2

Esta página fornece informações sobre os EPEL7 pacotes usados pelos clientes em AL2 e suas contrapartes AL2 023.

Para o restante dos pacotes, os clientes poderão usar Pacotes Suplementares para Amazon Linux (SPAL). O SPAL fornece milhares de EPEL9 pacotes, criados especificamente para o Amazon Linux 2023, mas esses pacotes não são cobertos pelos AWS Support Plans. Isso significa que CVEs os pacotes SPAL não estão sendo rastreados e os patches são fornecidos somente quando disponíveis no upstream.

 Important

Consulte a documentação [Pacotes suplementares para Amazon Linux](#) antes de usá-lo.

Tópicos

- [axel- HTTP/FTP cliente](#)
- [brotli e libbrotli: compressão](#)
- [collectd: daemon de coleta de estatísticas](#)
- [cpulimit: limitador de uso da CPU](#)
- [exim: agente de transferência de mensagens](#)
- [fuse3: File System in Userspace \(FUSE\) v3](#)
- [ganglia: sistema de monitoramento distribuído](#)
- [git-lfs: controle de versão de arquivos grandes com o Git](#)
- [haveged: uma fonte de entropia usando o algoritmo HAVEGE](#)
- [inotify-tools: ferramentas de linha de comandos da inotify](#)
- [iperf- Referência TCP/UDP de desempenho](#)

- [jemalloc](#): implementação alternativa de malloc
- [libbsd](#): biblioteca de funções compatível com BSD
- [libserf](#): biblioteca de cliente HTTP
- [libzstd](#): biblioteca de compressão zstd
- [Servidor web de lighttpd](#)
- [lshell](#): um shell restrito
- [monit](#): monitor de processos, arquivos, diretórios e dispositivos
- [nodejs](#)
- [perl-Config-General](#)
- [python2-lockfile](#): bloqueio de arquivos
- [python2-rsa](#): RSA pura em Python
- [python2-simplejson](#): rotinas JSON para Python 2
- [rkhunter](#): Rootkit Hunter
- [rssh](#): um shell restrito para uso com OpenSSH
- [sscg](#): gerador de certificados SSL autoassinados
- [stress](#): teste de estresse
- [stress-ng](#): teste de estresse
- [tmpwatch](#): remove arquivos com base na hora do último acesso
- [xmlstarlet](#): utilitários XML de linha de comandos

axel- HTTP/FTP cliente

O pacote axel estava em EPEL7 e nunca foi distribuído como parte do Amazon Linux. As alternativas disponíveis em AL2 023 são curl e wget

Warning

A opção -S para axel usa uma conexão http não criptografada para descobrir espelhos para um arquivo.

É altamente recomendável migrar qualquer uso de axel para curl ou wget.

brotli e **libbrotli**: compressão

Os **libbrotli** pacotes **brotli** e **libbrotli** estavam disponíveis em EPEL7, enquanto apenas o **brotli** pacote estava disponível no AL2 núcleo.

Tanto os **libbrotli** pacotes **brotli** quanto os **libbrotli** estão incluídos em AL2 023.

O **brotli** pacote pode ser instalado no AL2 023 com o seguinte comando:

```
[ec2-user ~]$ sudo dnf install brotli
```

O **libbrotli** pacote pode ser instalado no AL2 023 com o seguinte comando:

```
[ec2-user ~]$ sudo dnf install libbrotli
```

collectd: daemon de coleta de estatísticas

O **collect** pacote estava disponível e também estava disponível nos **collectd** **collectd-python3** AL2 Extras. EPEL7

O **collectd** pacote está incluído no AL2 023 e pode ser instalado executando o seguinte comando:

```
[ec2-user ~]$ sudo dnf install collectd
```

cpulimit: limitador de uso da CPU

No Amazon Linux 2023, **systemd** fornece funcionalidade para limitar o uso de processos ou grupos de processos pela CPU. Essa funcionalidade também é fácil de usar para qualquer serviço **systemd**.

O **systemd** contém mecanismos avançados de controle de recursos que podem ser usados para garantir que qualquer tarefa ou grupo de tarefas tenha um limite nos recursos que pode consumir. Para saber mais, consulte a documentação upstream de [systemd.resource-control](#) e [Limitar o uso de recursos do processo no AL2023 usando systemd](#).

exim: agente de transferência de mensagens

O **exim** pacote estava disponível em EPEL7 e estava disponível anteriormente em AL1. O Amazon Linux 2023 fornece tanto o Mail Transfer Agents postfix quanto o sendmail Mail Transfer Agents (MTAs).

fuse3: File System in Userspace (FUSE) v3

O pacote `fuse3` (incluindo `fuse3-libs` e `fuse3-devel`) estava em EPEL7. Esses pacotes fazem parte do AL2 023 e cada um pode ser instalado executando o seguinte comando relevante:

```
[ec2-user ~]$ sudo dnf install fuse3
```

```
[ec2-user ~]$ sudo dnf install fuse3-libs
```

```
[ec2-user ~]$ sudo dnf install fuse3-devel
```

ganglia: sistema de monitoramento distribuído

O `ganglia` pacote estava disponível EPEL7 e estava disponível anteriormente em AL1. Não foi enviado com AL2.

O projeto upstream teve um período de inatividade em que algumas vagas não CVEs estavam sendo atendidas. Embora tenha havido atividade recente no projeto upstream, não está planejado adicionar `ganglia` ao AL2 023.

git-lfs: controle de versão de arquivos grandes com o Git

O pacote `git-lfs` estava em EPEL7. No Amazon Linux 2023, o pacote `git-lfs` está incluído no repositório principal. Em AL2 023, `git-lfs` pode ser instalado executando o seguinte comando:

```
[ec2-user ~]$ sudo dnf install git-lfs
```

haveged: uma fonte de entropia usando o algoritmo HAVEGE

O pacote `haveged` estava em EPEL7. O Amazon Linux 2023 vem pré-configurado com fontes de entropia, não exigindo o uso de `haveged`.

inotify-tools: ferramentas de linha de comandos da inotify

O `inotify-tools` pacote chegou e está incluído no AL2 023. EPEL7

Note

Em AL2 023, `systemd` oferece suporte à ativação baseada em caminho, que pode ser usada para agir em eventos, como quando um caminho existe ou muda.

Grande parte do uso de `inotify-tools` já pode ser melhor realizado de maneira mais confiável usando a ativação de caminho de `systemd`. Para saber mais, consulte [systemd.path](#).

O `inotify-tools` pacote está incluído no AL2 023 e pode ser instalado executando o seguinte comando:

```
[ec2-user ~]$ sudo dnf install inotify-tools
```

iperf- Referência TCP/UDP de desempenho

O pacote da `iperf` versão 2 estava disponível e também estava disponível no testing AL2 Extra. e também estava disponível em EPEL7 AL1

Note

O pacote `iperf3` também está disponível, fornecendo a versão 3 de `iperf`.

O `iperf` pacote está incluído no AL2 023 e pode ser instalado executando o seguinte comando:

```
[ec2-user ~]$ sudo dnf install iperf
```

jemalloc: implementação alternativa de **malloc**

O `jemalloc` pacote chegou EPEL7 e estava disponível nos `lamp-mariadb10.2-php7.2` `mariadb10.5` AL2 Extras.

O `jemalloc` pacote está incluído no AL2 023 e pode ser instalado executando o seguinte comando:

```
[ec2-user ~]$ sudo dnf install jemalloc
```

libbsd: biblioteca de funções compatível com BSD

O `libbsd` pacote estava pronto EPEL7 e também estava disponível no testing AL2 Extra.

O `libbsd` pacote está incluído no AL2 023 e pode ser instalado executando o seguinte comando:

```
[ec2-user ~]$ sudo dnf install libbsd
```

Os arquivos de desenvolvimento de `libbsd` podem ser instalados executando o comando a seguir.

```
[ec2-user ~]$ sudo dnf install libbsd-devel
```

libserf: biblioteca de cliente HTTP

O pacote `libserf` estava em EPEL7. O pacote `libserf` é fornecido no Amazon Linux 2023. É possível instalá-lo executando o seguinte comando:

```
[ec2-user ~]$ sudo dnf install libserf
```

libzstd: biblioteca de compressão zstd

O `libzstd` pacote estava no AL2 núcleo, assim como noEPEL7. O `libzstd` pacote também faz parte do AL2 023.

```
[ec2-user ~]$ sudo dnf install libzstd
```

Servidor web de **lighttpd**

O `lighttpd` pacote estava disponível EPEL7 e estava disponível anteriormente em AL1. O Amazon Linux 2023 oferece os servidores web `httpd` e `nginx` do Apache.

lshell: um shell restrito

O pacote `lshell` nunca foi distribuído como parte do Amazon Linux. Ele estava disponível em EPEL6. O [repositório de pacotes do Fedora para lshell](#) explica [por que ele não virou pacote](#) em EPEL7 ou Fedora 30. Também foi [removido do Debian](#).

[O lshell projeto upstream não está mais sendo mantido ativamente e contém os conhecidos Críticos não corrigidos CVEs: CVE-2016-6902e CVE-2016-6903.](#)

A alternativa sugerida no bug do Debian, [rssh](#), também não é mantida pelo desenvolvedor original, com o autor citando problemas de segurança irremediáveis como motivo.

Por esses motivos, `lschell` a adição a AL2 023 não está planejada.

monit: monitor de processos, arquivos, diretórios e dispositivos

No Amazon Linux 2023, `systemd` oferece uma ampla variedade de funcionalidades para monitorar, iniciar, interromper e reiniciar serviços. Isso inclui reinicializações com limite de taxa, espera entre tentativas de reinicialização e inicialização de outro serviço em caso de falha. Para saber mais, consulte a documentação de [systemd.service](#).

No AL2 023, `systemd` também oferece suporte à ativação baseada em caminho, que pode ser usada para agir em eventos, como quando um caminho existe ou muda. Para saber mais, consulte [systemd.path](#).

Há opções de configuração comuns para unidades de `systemd` que permitem especificar dependências, condicionais e ações a serem tomadas em caso de sucesso ou falha. Para saber mais, consulte a documentação de [systemd.unit](#).

Há mecanismos avançados de controle de recursos fornecidos por `systemd` que podem ser usados para garantir que qualquer tarefa de monitoramento não faça uso excessivo de CPU ou memória. Para saber mais, consulte [systemd.resource-control](#).

nodejs

O pacote da `nodejs` versão 16 estava disponível EPEL7 e agora `nodejs` está incluído na versão AL2 023. No momento em que este artigo foi escrito, as `nodejs` versões 18 e 20 estavam disponíveis em AL2 023. Você pode instalar `nodejs` 18 em AL2 023 com o seguinte comando:

```
[ec2-user ~]$ sudo dnf install nodejs
```

Você pode instalar `nodejs` 20 em AL2 023 com o seguinte comando:

```
[ec2-user ~]$ sudo dnf install nodejs20
```

perl-Config-General

O `perl-Config-General` pacote chegou e agora está incluído no AL2 023. EPEL7 Você pode instalar o `perl-Config-General` pacote em AL2 023 com o seguinte comando:

```
[ec2-user ~]$ sudo dnf install perl-Config-General
```

Os módulos Perl também podem ser instalados solicitando que o DNF instale o pacote que fornece um módulo Perl específico. Com esse método, é possível usar o nome mais conhecido do módulo Perl em vez do nome do pacote do sistema operacional.

```
[ec2-user ~]$ sudo dnf install 'perl(Config:General)'
```

python2-lockfile: bloqueio de arquivos

O python2-lockfile pacote estava pronto EPEL7 e AL2 incluía um python-lockfile pacote. Em AL2 023 [O Python 2.7 foi substituído pelo Python 3](#), portanto, uma variante do Python 2 desse pacote não será adicionada ao AL2 023.

A versão Python 3 deste pacote está incluída na AL2 versão 023. Você pode instalar o python3-lockfile pacote no AL2 023 com um dos seguintes comandos:

```
[ec2-user ~]$ sudo dnf install python3-lockfile
```

Os módulos Python também podem ser instalados solicitando que o DNF instale o pacote que fornece um módulo Python específico.

```
[ec2-user ~]$ sudo dnf install 'python3dist(lockfile)'
```

python2-rsa: RSA pura em Python

O python2-rsa pacote estava pronto EPEL7 e AL2 incluía um python2-rsa pacote. Em AL2 023 [O Python 2.7 foi substituído pelo Python 3](#), portanto, uma variante do Python 2 desse pacote não será adicionada ao AL2 023.

A versão Python 3 deste pacote está incluída na AL2 versão 023. Você pode instalar o python3-rsa pacote no AL2 023 com um dos seguintes comandos:

```
[ec2-user ~]$ sudo dnf install python3-rsa
```

Os módulos Python também podem ser instalados solicitando que o DNF instale o pacote que fornece um módulo Python específico.

```
[ec2-user ~]$ sudo dnf install 'python3dist(rsa)'
```

python2-simplejson: rotinas JSON para Python 2

O pacote `python2-simplejson` estava em EPEL7. Em AL2 023 [O Python 2.7 foi substituído pelo Python 3](#), portanto, uma variante do Python 2 desse pacote não será adicionada ao AL2 023.

A versão Python 3 deste pacote está incluída na AL2 versão 023. Você pode instalar o `python3-simplejson` pacote em AL2 023 com o seguinte comando:

```
[ec2-user ~]$ sudo dnf install python3-simplejson
```

Os módulos Python também podem ser instalados solicitando que o DNF instale o pacote que fornece um módulo Python específico.

```
[ec2-user ~]$ sudo dnf install 'python3dist(simplejson)'
```

rkhunter: Rootkit Hunter

O `rkhunter` pacote está incluído em AL2 023 junto com `chkrootkit`.

```
[ec2-user ~]$ sudo dnf install rkhunter
```

```
[ec2-user ~]$ sudo dnf install chkrootkit
```

rssh: um shell restrito para uso com OpenSSH

O pacote `rssh` estava em EPEL7. O pacote [rssh](#) upstream não está sendo mantido, com o autor citando problemas de segurança irremediáveis como motivo.

Com o autor citando problemas de segurança não solucionáveis, a adição `rssh` a AL2 023 não está planejada.

sscg: gerador de certificados SSL autoassinados

O `sscg` pacote estava no AL2 núcleo, assim como no EPEL7. O `sscg` pacote também faz parte do AL2 023.

```
[ec2-user ~]$ sudo dnf install sscg
```

stress: teste de estresse

O **stress** pacote chegou EPEL7 e também estava disponível em AL1

O **stress** pacote está incluído no AL2 023 e pode ser instalado executando o seguinte comando:

```
[ec2-user ~]$ sudo dnf install stress
```

stress-ng: teste de estresse

O **stress-ng** pacote estava pronto EPEL7 e também estava disponível no `testing` AL2 Extra.

O **stress-ng** pacote está incluído no AL2 023 e pode ser instalado executando o seguinte comando:

```
[ec2-user ~]$ sudo dnf install stress-ng
```

tmpwatch: remove arquivos com base na hora do último acesso

No Amazon Linux 2023, essa funcionalidade é fornecida por [systemd-tmpfiles](#).

xmlstarlet: utilitários XML de linha de comandos

O **xmlstarlet** pacote estava disponível EPEL7 e não está disponível em AL2 023.

O pacote upstream não é atualizado há mais de 9 anos (última atualização em agosto de 2014). Durante os quatro anos anteriores (pelo menos desde julho de 2010), um pedido para um novo mantenedor ficou sem resposta. É por esse motivo que não está planejado **xmlstarlet** adicionar AL2 023.

O Python 2.7 foi substituído pelo Python 3

AL2 fornece patches de suporte e segurança para o Python 2.7 até junho de 2025, como parte do nosso compromisso de suporte de longo prazo (LTS) para AL2 pacotes principais. Esse suporte vai além da declaração da comunidade Python upstream do Python 2.7 de janeiro de 2020 end-of-life.

AL2 usa o gerenciador de yum pacotes, que tem uma forte dependência do Python 2.7. Em AL2 023, o gerenciador de dnf pacotes migrou para o Python 3 e não precisa mais do Python 2.7. AL2023 foi completamente movido para o Python 3.

Note

AL2023 removeu o Python 2.7, então todos os componentes do sistema operacional que exigem Python são escritos para funcionar com o Python 3. Para continuar usando uma versão do Python fornecida e compatível com o Amazon Linux, converta o código do Python 2 em Python 3.

Para obter mais informações sobre Python no Amazon Linux, consulte [Python no AL2023](#).

Atualizações de segurança

O Amazon Linux 2023 aprimora o endurecimento presente em AL2. Para obter mais informações, consulte [Segurança e compatibilidade no Amazon Linux 2023](#). Para obter mais informações sobre as alterações do fortalecimento do AL2 kernel, consulte [Alterações na configuração do kernel com foco na segurança](#).

Tópicos

- [SELinux](#)
- [OpenSSL 3](#)
- [IMDSv2](#)
- [Remoção do hotpatch log4j \(log4j-cve-2021-44228-hotpatch\)](#)

SELinux

Por padrão, Security Enhanced Linux (SELinux) para AL2 023 é enabled e definido como permissive modo. No modo permissive, as negações de permissão são registradas, mas não aplicadas.

SELinux é um recurso de segurança do kernel Amazon Linux, que estava disabled em AL2. SELinux é uma coleção de recursos e utilitários do kernel que fornece a arquitetura de controle de acesso (MAC) obrigatória nos principais subsistemas do kernel.

Para obter mais informações, consulte [Configurando modos do SELinux para o AL2023](#).

Para obter mais informações sobre SELinux repositórios, ferramentas e políticas, consulte [SELinux Notebook](#), [Tipos de SELinux política](#) e [SELinux projeto](#).

OpenSSL 3

AL2023 apresenta o kit de ferramentas de Open Secure Sockets Layer version 3 (OpenSSL 3) criptografia. AL2023 suportes TLS 1.3 e protocolos TLS 1.2 de rede.

Por padrão, AL2 vem com OpenSSL 1.0.2. Você pode criar aplicativos contra o OpenSSL 1.1.1.

Para mais informações sobre OpenSSL, consulte o [Guia de migração de OpenSSL](#).

Para obter mais informações sobre a segurança, consulte [Atualizações e recursos de segurança](#).

IMDSv2

Por padrão, todas as instâncias lançadas com a AMI AL2 023 requerem IMDSv2 -only e seu limite de salto padrão será definido como 2 para permitir o suporte à carga de trabalho em contêineres. Isso é feito definindo o parâmetro `imds-support` como `v2.0`. Para obter mais informações, consulte [Configurar a AMI](#) no Guia EC2 do usuário da Amazon.

Note

O tempo de validade do token de sessão pode estar entre 1 segundo e 6 horas. Os endereços para direcionar as solicitações de API para consultas IMDSv2 são os seguintes:

- IPv4: 169.254.169.254
- IPv6: fd00:ec2:254

Você pode substituir manualmente essas configurações e habilitar IMDSv1 usando as propriedades de inicialização da opção de metadados da instância. Você também pode usar os controles do IAM para aplicar configurações de IMDS diferentes. Para obter mais informações sobre como configurar e usar o Instance Metadata Service, consulte [Usar IMDSv2](#), [Configurar opções de metadados de instância para novas instâncias e Modificar opções de metadados de instância para instâncias existentes](#), no Amazon EC2 User Guide.

Remoção do hotpatch log4j (**log4j-cve-2021-44228-hotpatch**)

Note

AL2023 não é enviado com o `log4j-cve-2021-44228-hotpatch` pacote.

Em resposta ao [CVE-2021-44228](#), a Amazon Linux lançou uma versão empacotada em RPM do Hotpatch para Apache Log4j para [e. AL1 AL2](#) No [anúncio da adição do hotpatch ao Amazon Linux](#), observamos que “Instalar o hotpatch não substitui a atualização para uma versão log4j que atenua o CVE-2021-44228 ou o CVE-2021-45046”.

O hotpatch foi uma mitigação para dar tempo de corrigir log4j. A primeira versão de Disponibilidade Geral (GA) do AL2 023 foi 15 meses após o [CVE-2021-44228](#), portanto, o AL2 023 não vem com o hotpatch (ativado ou não).

Os usuários que executam suas próprias versões de log4j no Amazon Linux devem garantir que tenham atualizado para versões não afetadas pela [CVE-2021-44228](#) ou [CVE-2021-45046](#).

AL2O 023 fornece orientações [Atualizando AL2023](#) para que você possa se manter atualizado com os patches de segurança. Os avisos de segurança são publicados no [Amazon Linux Security Center](#).

Atualizações determinísticas para estabilidade

Com o recurso de atualizações determinísticas por meio de repositórios versionados, cada 023 AL2 AMI, por padrão, está bloqueada para uma versão específica do repositório. Você pode usar atualizações determinísticas para obter maior consistência entre as versões e atualizações do pacote. Cada versão, principal ou secundária, inclui uma versão específica do repositório.

Novo no AL2 023, a atualização determinística por padrão está ativada. Essa é uma melhoria em relação ao método manual e incremental de bloqueio usado em AL2 e em outras versões anteriores.

Para obter mais informações, consulte [Atualizações determinísticas por meio de repositórios versionados em AL2023](#).

gp3 como tipo de volume padrão do Amazon EBS

A AMI AL2 023 e AL2 ambas usam o sistema de XFS arquivos no sistema de arquivos raiz. Para o AL2 023, as `mkfs` opções do sistema de arquivos do dispositivo raiz são ainda mais otimizadas para

a Amazon EC2. AL20 023 também oferece suporte a vários outros sistemas de arquivos que você pode usar em outros volumes para atender aos seus requisitos específicos.

AL2023 AMIs usa gp3 volumes Amazon EBS por padrão, enquanto usa gp2 volumes AL2 AMIs Amazon EBS por padrão. É possível alterar o tipo de volume quando lançar uma instância.

Para obter mais informações sobre os tipos de volume Amazon EBS, consulte [Volumes de propósito do Amazon EBS](#).

Para obter mais informações sobre o lançamento de uma EC2 instância da Amazon, consulte [Iniciar uma instância](#) no Guia EC2 do usuário da Amazon.

Hierarquia unificada de grupos de controle (cgroup v2)

Um Grupo de Controle (cgroup) é um recurso do kernel Linux para organizar hierarquicamente os processos e distribuir os recursos do sistema entre eles. Os grupos de controle são usados extensivamente para implementar um tempo de execução de contêiner e por `systemd`.

AL2 suporta `cgroupv1` e AL2 023 suporta `cgroupv2`. Isso é notável ao executar workloads em contêineres, como quando [Usar AMIs do Amazon ECS baseadas no AL2023 para hospedar workloads em contêineres](#).

Embora o AL2 023 ainda inclua código que pode fazer com que o sistema seja executado usando `cgroupv1`, essa não é uma configuração recomendada ou suportada e será completamente removida em uma futura versão principal do Amazon Linux.

Há uma extensa documentação sobre as [interfaces de baixo nível de kernel do Linux](#), bem como a [documentação de delegação de `systemd` cgroup](#).

Um caso de uso comum fora dos contêineres é criar unidades de `systemd` que tenham limites nos recursos do sistema que podem usar. Para saber mais, consulte [systemd.resource-control](#).

Temporizadores **systemd** substituem **cron**

O `cronie` pacote foi instalado por padrão na AL2 AMI, fornecendo suporte para a `crontab` forma tradicional de programar tarefas periódicas. Em AL2 023, não `cronie` está incluído por padrão. Portanto, o suporte a `crontab` não é mais fornecido por padrão.

Opcionalmente, você pode instalar o pacote `cronie` para usar trabalhos tarefas clássicas `cron`. Recomendamos que você migre para temporizadores `systemd` devido à funcionalidade adicional fornecida pelo `systemd`.

Melhoria da cadeia de ferramentas: gcc, binutils e glibc

AL2O 023 inclui muitos dos mesmos pacotes principais do. AL2

Atualizamos os três pacotes principais do conjunto de ferramentas a seguir para AL2 023.

Nome do pacote	AL2	AL2023
glibc	2.26	2.34
gcc	7.3	11.3
binutils	2.29	2,39

Para obter mais informações, consulte [Pacotes principais do conjunto de ferramentas glibc, gcc, binutils](#).

Para saber mais sobre runtimes das linguagens C, C++ e Fortran, incluindo normas de linguagem padrão atualizadas, consulte [C, C++ e Fortran no AL2023](#).

Para obter mais informações sobre essas otimizações, consulte [Otimizações operacionais e de desempenho](#).

Diário de **systemd** substitui **rsyslog**

Em AL2 023, o pacote do sistema de registro foi alterado de AL2. AL2O 023 não é instalado `rsyslog` por padrão, portanto, os arquivos de log baseados em texto, como os `/var/log/messages` que estavam disponíveis em, AL2 não estão disponíveis por padrão. A configuração padrão para AL2 023 é `systemd-journal`, que pode ser examinada usando `journalctl`. Embora `rsyslog` seja um pacote opcional no AL2 023, recomendamos a nova `journalctl` interface `systemd` baseada e os pacotes relacionados. Para obter mais informações, consulte a página do manual [journalctl](#).

O `systemd journal` equivalente a alguns comandos de `syslog` comumente usados é abordado na tabela a seguir.

Comando AL2 syslog	AL2equivalente a 023 systemd journal
<code>[ec2-user ~]\$ cat /var/log/messages</code>	<code>[ec2-user ~]\$ journalctl</code>
<code>[ec2-user ~]\$ tail -f /var/log/messages</code>	<code>[ec2-user ~]\$ journalctl -f</code>
<code>[ec2-user ~]\$ grep foo /var/log/messages</code>	<code>[ec2-user ~]\$ journalctl grep foo</code>

Dependências de pacotes minimizadas

O Amazon Linux 2023 minimiza o grafo de dependências de muitos pacotes para fornecer uma pegada menor para as aplicações. As mudanças notáveis AL2 incluem os `gnupg-minimal` pacotes `curl-minimal` e, que reduzem significativamente o número de pacotes necessários, mantendo a funcionalidade comumente usada.

Tópicos

- [Alterações de pacotes para curl e libcurl](#)
- [Guarda de Privacidade GNU \(GNUPG\)](#)

Alterações de pacotes para **curl** e **libcurl**

AL2023 separa os protocolos e funcionalidades comuns dos `libcurl` pacotes `curl` e em `e. curl-minimal libcurl-minimal`. Isso reduz o espaço ocupado por disco, memória e dependência para a maioria dos usuários e é o pacote padrão para AL2 023 AMIs e contêineres.

Se a funcionalidade completa do `curl` for necessária, por exemplo, para suporte de `gopher://`, execute os seguintes comandos para instalar os `curl-full` e `libcurl-full`.

```
$ dnf swap libcurl-minimal libcurl-full
```

```
$ dnf swap curl-minimal curl-full
```

Guarda de Privacidade GNU (GNUPG)

AL2023 separa a funcionalidade mínima e completa do gnupg2 pacote em gnupg2-minimal pacotes. gnupg2-full Por padrão, apenas o pacote gnupg2-minimal está instalado. Isso fornece a funcionalidade mínima necessária para verificar as assinaturas digitais nos pacotes rpm.

Para obter mais funcionalidades de gnupg2, como a capacidade de baixar chaves de um servidor de chaves, verifique se o pacote gnupg2-full está instalado. Execute o seguinte comando para trocar gnupg2-minimal por gnupg2-full.

```
$ dnf swap gnupg2-minimal gnupg2-full
```

Amazon Corretto como JVM padrão

AL2023 é fornecido com o [Amazon Corretto](#) como o Java Development Kit (JDK) padrão (e único). Todos os pacotes Java baseados em AL2023 são todos construídos com Amazon Corretto 17.

Se você estiver migrando de AL2, poderá fazer a transição sem problemas da OpenJDK versão equivalente para o AL2 Amazon Corretto

AWS CLIV2

AL2023 vem com a AWS CLI versão 2, enquanto que AL2 vem com a versão 1 do AWS CLI.

Inicialização preferencial e segura por UEFI

Por padrão, todas as instâncias iniciadas com a AMI AL2023 em tipos de instância compatíveis com o firmware UEFI serão iniciadas no modo UEFI. Isso é feito definindo o parâmetro AMI do modo de inicialização como uefi-preferred. Para obter mais informações, consulte [Modos de inicialização](#) no Guia EC2 do usuário da Amazon.

Nos tipos de EC2 instância da Amazon que oferecem suporte ao UEFI Secure Boot, é possível habilitar o Secure Boot no Amazon Linux 2023. Para obter mais informações, consulte [UEFI Secure Boot no AL2023](#).

Alterações na configuração padrão do servidor SSH

Para a AMI AL2 023, alteramos os tipos de chaves de `sshd` host que geramos com a versão. Também eliminamos alguns tipos de chaves legadas para evitar gerá-las no momento do lançamento. Os clientes devem oferecer suporte aos protocolos `rsa-sha2-256` e `rsa-sha2-512` ou `ssh-ed25519` com o uso de uma chave `ed25519`. Por padrão, as assinaturas `ssh-rsa` estão desabilitadas.

Além disso, AL2 023 configurações no `sshd_config` arquivo padrão contém `UseDNS=no`. Essa nova configuração significa que é menos provável que deficiências de DNS bloqueiem sua capacidade de estabelecer sessões `ssh` com suas instâncias. A desvantagem é que as entradas de linha `from=hostname.domain,hostname.domain` em seus arquivos `authorized_keys` não serão resolvidas. Como `sshd` não tenta mais resolver os nomes DNS, cada valor de `hostname.domain` separado por vírgula deve ser traduzido para um IP address correspondente.

Para obter mais informações, consulte [Configuração do servidor SSH padrão](#).

AL2023 o kernel muda de AL2

AL2023 traz o kernel 6.1, bem como muitas mudanças de configuração para otimizar ainda mais o Amazon Linux para a nuvem. Para a maioria dos usuários, essas mudanças devem ser totalmente transparentes.

IPv4 TTL

O TTL para IPv4 é configurado via `sysctl`, com os valores padrão presentes em `/etc/sysctl.d/00-defaults.conf`. Esse valor pode ser personalizado por meio dos métodos `sysctl` usuais. Para saber mais, consulte a página `sysctl` man.

AL2 define o `net.ipv4.ip_default_ttl` valor como 255, enquanto o AL2023 define como 127. Isso alinha os padrões do Amazon Linux com outras grandes distribuições do Linux. Não é recomendável alterar esse padrão sem uma necessidade comprovada.

Alterações na configuração do kernel com foco na segurança

Opção do CONFIG	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_IKDBG_ON_DATA_CORRUPTION	n	y	n	y	y	y	y	y
CONFIG_IKDBG_FAULT_MIN_ARRAY	4096	4096	4096	4096	65536	65536	65536	65536
CONFIG_IKDBG_VMEM	n	y	n	y	n	n	n	n
CONFIG_IKDBG_VPORT	n	y	n	y	n	n	n	n
CONFIG_IKDBG_RTIFY_SOURCE	n	y	n	y	y	y	y	y
CONFIG_IKDBG_RDENED_IKDBG_ERCOPY_IKDBG_LLBACK	N/D	N/D	y	y	N/D	N/D	N/D	N/D
CONFIG_IKDBG_IT_ON_AIOCC_DEFAULT_ON	N/D	N/D	n	n	n	n	n	n
CONFIG_IKDBG_IT_ON_FI	N/D	N/D	n	n	n	n	n	n

Opção do CONFIG	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
E_DEFAULT_ON								
CONFIG_MMU_DEFAULT_DMA_REQUIRED	N/D	N/D	N/D	N/D	n	n	n	n
CONFIG_ISCSI_AUTOLOAD	y	y	y	y	n	n	n	n
CONFIG_HED_CORE	N/D	N/D	N/D	N/D	N/D	y	N/D	y
CONFIG_HED_STACK_END_CHECK	n	y	n	y	y	y	y	y
CONFIG_CURITY_ESG_RESTRICT	n	n	n	n	y	y	y	y
CONFIG_CURITY_LINUX_DEFAULTABLE	y	y	y	y	n	n	N/D	N/D
CONFIG_UFFLE_PAGE_ALLOCATION	N/D	N/D	y	y	y	y	y	y

Opção do CONFIG	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_AB_FREEZE_HARDDISK	n	y	y	y	y	y	y	y
CONFIG_AB_FREEZE_RANDOM	n	n	y	y	y	y	y	y

Alterações específicas na configuração do kernel com foco na segurança x86-64

Opção do CONFIG	AL2/4.14/x86_64	AL2/5.10/x86_64	AL2023/6.1/ x86_64	AL2023/6.12/ x86_64
<u>CONFIG_AMD_IOMMU</u>	y	y	y	y
<u>CONFIG_AMD_IOMMU_V2</u>	m	m	y	N/D
<u>CONFIG_RANDOMIZE_MEMORY</u>	N/D	y	y	y

Alterações específicas na configuração do kernel com foco na segurança aarch64 (ARM/Graviton)

Opção do CONFIG	AL2/4.14/ aarch64	AL2/5.10/ aarch64	AL2023/6.1/ aarch64	AL2023/6.12/ aarch64
<u>CONFIG_ARM64_PTR_AUTH</u>	N/D	y	y	y
<u>CONFIG_ARM64_PTR_AUTH_KERNEL</u>	N/D	N/D	y	y
<u>CONFIG_ARM64_SW_TTBR0_PAN</u>	y	y	y	y

/dev/mem, /dev/kmem e /dev/port

O Amazon Linux 2023 desativa /dev/mem, CONFIG_DEVMEM e /dev/port (e CONFIG_DEVPORT) completamente, com base nas restrições já existentes. AL2

O /dev/kmem código foi completamente removido do Linux no kernel 5.13 e, embora tenha sido desativado AL2, agora não é aplicável a. AL2023

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

FORTIFY_SOURCE

AL2023 habilitado CONFIG_FORTIFY_SOURCE em todas as arquiteturas suportadas. Esse recurso é um recurso de fortalecimento da segurança. Onde o compilador pode determinar e validar os tamanhos do buffer, esse recurso pode detectar estouros de buffer em funções comuns de string e memória.

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Carregamento automático de disciplinas de linha (**CONFIG_LDISC_AUToload**)

O AL2023 kernel não carregará automaticamente disciplinas de linha, como por exemplo, por software usando o. `TIOCSETDioct1`, a menos que a solicitação venha de um processo com as `CAP_SYS_MODULE` permissões.

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Acesso a **dmesg** para usuários sem privilégios (**CONFIG_SECURITY_DMESG_RESTRICT**)

Por padrão, AL2023 não permite que usuários sem privilégios acessem o `dmesg`

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

SELinux **selinuxfs**desabilitar

AL2023 desativa a opção obsoleta do CONFIG_SECURITY_SELINUX_DISABLE kernel, que habilitou um método de tempo de execução de desativação antes do carregamento da política SELinux .

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Outras alterações na configuração do kernel

Opção do CONFIG	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_1	100	250	100	250	100	100	100	100
CONFIG_1_CPUS	4096	8192	4096	8192	4096	8192	4096	8192
CONFIG_1_NIC_ON_CPS	y	n	y	n	y	y	y	y

Opção do CONFIG	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_NIC_ON_OFF PS_VALUES	1	0	1	0	1	1	1	1
CONFIG_IP	m	m	m	m	n	n	n	n
CONFIG_IP	m	m	m	m	n	n	n	n
CONFIG_N_PV	N/D	y	N/D	n	N/D	n	N/D	n

CONFIG_HZ

AL2023 define CONFIG_HZ como 100 em ambas x86-64 as aarch64 plataformas.

CONFIG_NR_CPUS

AL2023 define CONFIG_NR_CPUS um número mais próximo do número máximo de núcleos de CPU encontrados no Amazon EC2.

Pânico no OOPS

O AL2023 kernel entrará em pânico quando se opuser. Esse recurso é equivalente à inicialização com `oops=panic` na linha de comando do kernel.

Um kernel oops é onde o kernel detectou um erro interno que pode afetar a confiabilidade adicional do sistema.

Suporte para PPP e SLIP

AL2023 não suporta os protocolos PPP ou SLIP.

Suporte para convidados do Xen PV

AL2023 não suporta a execução como convidado do Xen PV.

Suporte ao sistema de arquivos do kernel

Houve várias mudanças nos sistemas de arquivos nos quais o kernel AL2 suportará a montagem, junto com mudanças nos esquemas de particionamento que o kernel analisará.

Opção do CONFIG	AL2/4.14/ aarch64	AL2/4.14/ x86_64	AL2/5.10/ aarch64	AL2/5.10/ x86_64	AL2023/6 1/ aarch64	AL2023/6 1/ x86_64	AL2023/6 12/ aarch64	AL2023/6 12/ x86_64
CONFIG_ S_FS	n	m	n	m	n	n	n	n
CONFIG_ _RXRPC	n	m	n	m	n	n	n	n
CONFIG_ D_DISKL EL	y	y	y	y	n	n	n	n
CONFIG_ AMFS	m	m	m	m	n	n	n	n
CONFIG_ AMFS_BLK KDEV	N/D	N/D	y	n	N/D	N/D	N/D	N/D
CONFIG_ _CLONE	N/D	N/D	n	n	n	n	n	n
CONFIG_ _ERA	m	n	m	n	n	n	n	n
CONFIG_ _INTEGR Y	n	m	n	m	m	m	m	m

Opção do CONFIG	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_ILOG_WRED	n	n	m	m	m	m	m	m
CONFIG_I_SWITCH	m	n	m	n	n	n	n	n
CONFIG_I_VERIFY	m	n	m	n	m	m	m	m
CONFIG_I_RYPT_FS	n	m	n	m	n	n	n	n
CONFIG_I_FAT_FS	N/D	N/D	m	m	m	m	m	m
CONFIG_I_T2_FS	n	m	n	m	n	n	n	n
CONFIG_I_T3_FS	n	m	n	m	n	n	n	n
CONFIG_I_S2_FS	m	m	m	m	n	n	n	n
CONFIG_I_SPLUS_FS	n	m	n	m	n	n	n	n
CONFIG_I_S_FS	n	m	n	m	n	n	n	n
CONFIG_I_S_FS	n	n	n	n	n	n	n	n

Opção do CONFIG	AL2/4.14/aarch64	AL2/4.14/x86_64	AL2/5.10/aarch64	AL2/5.10/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_M_PARTITION	n	y	n	y	n	n	n	n
CONFIG_C_PARTITION	n	y	n	y	n	n	n	n
CONFIG_S_V2	n	m	n	m	n	n	n	n
CONFIG_FS_FS	n	m	n	n	n	n	n	n
CONFIG_MFS_FS	n	m	n	m	n	n	n	n
CONFIG_XLARISS_PARTITION	n	y	n	y	n	n	n	n
CONFIG_UASHFS_MOUNTD	n	y	n	y	y	y	y	y
CONFIG_N_PARTITION	n	y	n	y	n	n	n	n

Suporte ao sistema de arquivos Andrew (AFS)

O kernel não é mais construído com suporte para o sistema de arquivos AFS. AL2 não foi fornecido com suporte de espaço de usuário para AFS.

suporte cramfs

O kernel não é mais construído com suporte para o sistema de arquivos cramfs. O sucessor AL2023 é o sistema de squashfs arquivos.

Suporte a etiquetas de disco BSD

O kernel não é mais construído com suporte para rótulos de disco BSD. Se for necessário ler volumes com rótulos de disco BSD, vários BSDs podem ser iniciados.

Alterações no Device Mapper

Houve várias alterações nos alvos do Device Mapper configurados no AL2023 kernel.

eCryptFs apoio

O sistema de arquivos ecryptfs foi descontinuado no Amazon Linux. Os componentes do espaço de usuário do ecryptfs estavam presentes AL1, foram removidos e AL2023 não constroem AL2 mais o kernel com suporte. ecryptfs

exFAT

Support para o sistema de exFAT arquivos foi adicionado ao kernel 5.10 em. AL2 Ele não estava presente no AL2 lançamento com um kernel 4.14. AL2023 continua oferecendo suporte ao sistema exFAT de arquivos.

Os sistemas de arquivos ext2, ext3 e ext4

AL2023 vem com a CONFIG_EXT4_USE_FOR_EXT2 opção, o que significa que o código do sistema de ext4 arquivos será usado para ler sistemas de ext2 arquivos legados.

CONFIG__FS GFS2

O kernel não é mais construído com CONFIG__FSGFS2.

Suporte estendido ao sistema de arquivos Apple HFS (HFS+)

Em AL2, somente os x86-64 kernels foram construídos com o suporte do sistema de hfsplus arquivos. O kernel AL2 5.15 não inclui hfsplus suporte em nenhuma arquitetura. Em AL2023, concluímos a descontinuação do suporte hfsplus no Amazon Linux.

Suporte a sistemas de arquivos HFS

Em AL2, somente os x86-64 kernels foram construídos com o suporte do sistema de hfs arquivos. O kernel AL2 5.15 não inclui hfs suporte em nenhuma arquitetura. Em AL2023, concluímos a descontinuação do suporte hfs no Amazon Linux.

Suporte a sistemas de arquivos JFS

Os AL2 x86-64 kernels mais antigos foram construídos com suporte ao sistema jfs de arquivos. O kernel AL2 5.15 não inclui jfs suporte em nenhuma arquitetura. Nenhum deles AL1 ou AL2 fornecido com o espaço de usuário do JFS. Em AL2023, concluímos a descontinuação do suporte jfs no Amazon Linux.

O kernel do Linux upstream está [considerando a remoção de JFS](#). Portanto, se você tiver dados em um sistema de arquivos JFS, deverá migrá-los para outro sistema de arquivos. Em 2024, JFS foi removido de todos os kernels atuais do Amazon Linux.

Suporte ao gerenciador de disco lógico do Windows (disco dinâmico) (CONFIG_LDM_PARTITION)

AL2023 não oferece mais suporte Windows 2000 a Windows XP discos Windows Vista dinâmicos com partições de MS-DOS estilo. Esse código nunca ofereceu suporte aos discos dinâmicos baseados em GPT mais recentes introduzidos com o Windows Vista.

Suporte ao mapa de partições do Macintosh

AL2023 não suporta mais o mapa de partições clássico do Macintosh. As versões modernas do macOS criarão tabelas de partições GPT modernas por padrão sobre esse tipo mais antigo.

NFSv2 apoio

AL2023 não suporta mais NFSv2, mas continua a oferecer suporte a NFSv3 NFSv4, NFSv4 .1 e NFSv4 .2. Recomendamos que você migre para NFSv3 ou para uma versão mais recente.

NTFS (CONFIG_NTFS_FS)

O ntfs3 código foi substituído ntfs para acessar sistemas de arquivos NTFS no Amazon Linux a partir do kernel 5.10 em. AL2 AL2023 não inclui mais o ntfs código e depende exclusivamente do ntfs3 código para acessar sistemas de arquivos NTFS.

romfs file system

O sistema de squashfs arquivos é o sucessor do sistema de romfs arquivos no Amazon Linux, e o AL2023 kernel não é mais construído com suporte para o. romfs

Formato de partição de disco rígido Solaris x86

AL2023 não oferece mais suporte ao formato de partição de disco rígido x86 do Solaris.

squashfszstdcompressão

AL2023 adiciona suporte para sistemas de squashfs arquivos zstd compactados em todas as arquiteturas suportadas.

Suporte para tabela de partição Sun

AL2023 não inclui mais suporte para o formato de tabela de partições Sun (CONFIG_SUN_PARTITION).

/tmp é agora tmpfs

O Amazon Linux 2023 introduz mudanças na forma como /tmp se comporta em comparação com o Amazon Linux 2. A configuração padrão para AL2 era que ambos /tmp /var/tmp estivessem no sistema de arquivos raiz. O Amazon Linux 2023 usa por padrão tmpfs para /tmp com um limite de 50% da RAM e um máximo de um milhão de inodes. Essas mudanças alinham o Amazon Linux com o comportamento de outras distribuições do Linux.

Para obter detalhes completos sobre o layout do sistema de arquivos de AL2 023, consulte [/tmp](#) e [/var/tmp](#) na [Layout do sistema de arquivos](#) seção.

Alterações na AMI e na imagem de contêiner

Houve algumas mudanças nos pacotes incluídos AMIs e nos contêineres.

O Amazon Linux 2023 apresenta uma [the section called “AL2023 Imagem mínima do contêiner”](#) e suporte à criação de [the section called “Criando imagens básicas de contêineres AL2023”](#). Para obter mais informações, consulte [Uso AL2023 em contêineres](#).

Comparação de pacotes instalados nas AMIs Amazon Linux 2 e Amazon Linux 2023

Uma comparação dos RPMs presentes nas AMIs padrão do Amazon Linux 2 e do AL2023.

Pacote	AMI do AL2	AMI do AL2023
acl	2.2.51	2.3.1
acpid	2.0.19	2.0.32
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.5.1
amazon-linux-extras	2.0.3	
amazon-linux-extras-yum-plugin	2.0.3	
amazon-linux-repo-s3		2023.6.20241031
amazon-linux-sb-keys		2023.1
amazon-rpm-config		228
amazon-ssm-agent	3.3.987.0	3.3.987.0
amd-ucode-firmware	20200421 (noarch)	20210208 (noarch)
at	3.1.13	3.1.23
attr	2.4.46	2.5.1
audit	2.8.1	3.0.6
audit-libs	2.8.1	3.0.6

Pacote	AMI do AL2	AMI do AL2023
authconfig	6.2.8	
aws-cfn-bootstrap	2.0	2.0
awscli	1.18.147	
awscli-2		2.15.30
basesystem	10.0	11
bash	4.2.46	5.2.15
bash-completion	2.1	2.11
bc	1.06.95	1.07.1
bind-export-libs	9.11.4	
bind-libs	9.11.4	9.18.28
bind-libs-lite	9.11.4	
bind-license	9.11.4	9.18.28
bind-utils	9.11.4	9.18.28
binutils	2.29.1	2.39
blktrace	1.0.5	
boost-date-time	1.53.0 (x86_64)	
boost-filesystem		1.75.0
boost-system	1.53.0 (x86_64)	1.75.0
boost-thread	1.53.0 (x86_64)	1.75.0
bridge-utils	1.5	

Pacote	AMI do AL2	AMI do AL2023
bzip2	1.0.6	1.0.8
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.68	2023.2.68
c-ares		1.19.1
checkpolicy		3.4
chkconfig	1.7.4	1.15
chrony	4.2	4.3
cloud-init	19.3	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart	0.31	0.31
coreutils	8.22	8.32
coreutils-common		8.32
cpio	2.12	2.13
cracklib	2.9.0	2.9.6
cracklib-dicts	2.9.0	2.9.6
cronie	1.4.11	
cronie-anacron	1.4.11	
crontabs	1.11	1.11
crypto-policies		20220428
crypto-policies-scripts		20220428

Pacote	AMI do AL2	AMI do AL2023
cryptsetup	1.7.4	2.6.1
cryptsetup-libs	1.7.4	2.6.1
curl	8.3.0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	2.1.27
cyrus-sasl-plain	2.1.26	2.1.27
dbus	1.10.24	1.12.28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.10.24	1.12.28
device-mapper	1.02.170	1.02.185
device-mapper-event	1.02.170	
device-mapper-event-libs	1.02.170	
device-mapper-libs	1.02.170	1.02.185
device-mapper-persistent-data	0.7.3	
dhclient	4.2.5	
dhcp-common	4.2.5	
dhcp-libs	4.2.5	
diffutils	3.3	3.8

Pacote	AMI do AL2	AMI do AL2023
dmidecode	3.2	
dmraid	1.0.0.rc16	
dmraid-events	1.0.0.rc16	
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0
dnf-plugin-support-info		1.2
dnf-utils		4.3.0
dosfstools	3.0.20	4.2
dracut	033	055
dracut-config-ec2	2.0	3.0
dracut-config-generic	033	055
dwz		0.14
dyninst	9.3.1 (x86_64)	10.2.1
e2fsprogs	1.42.9	1.46.5
e2fsprogs-libs	1.42.9	1.46.5
ec2-hibinit-agent	1.0.8	1.0.8

Pacote	AMI do AL2	AMI do AL2023
<u>ec2-instance-connect</u>	1.1	1.1
ec2-instance-connect-selinux	1.1	1.1
ec2-net-utils	1.7.3	
ec2-utils	1.2	2.2.0
ed	1.9	1.14.2
efibootmgr	15 (aarch64)	
efi-filesystem		5
efi-srpm-macros		5
efivar		38
efivar-libs	31 (aarch64)	38
elfutils-debuginfod-client		0.188
elfutils-default-yama-scope	0.176	0.188
elfutils-libelf	0.176	0.188
elfutils-libs	0.176	0.188
ethtool	4.8	5.15
expat	2.1.0	2.5.0
file	5.11	5.39
file-libs	5.11	5.39

Pacote	AMI do AL2	AMI do AL2023
filesystem	3.2	3.14
findutils	4.5.11	4.8.0
fipscheck	1.4.1	
fipscheck-lib	1.4.1	
fonts-srpm-macros		2.0.5
freetype	2.8	
fstrm		0.6.1
fuse-libs	2.9.2	2.9.9
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1.19
gdisk	0.8.10	1.0.8
generic-logos	18.0.0	
GeoIP	1.5.0	
gettext	0.19.8.1	0.21
gettext-libs	0.19.8.1	0.21
ghc-srpm-macros		1.5.0
glib2	2.56.1	2.74.7
glibc	2.26	2.34
glibc-all-langpacks	2.26	2.34

Pacote	AMI do AL2	AMI do AL2023
glibc-common	2.26	2.34
glibc-gconv-extra		2.34
glibc-locale-source	2.26	2.34
glibc-minimal-lang pack	2.26	
gmp	6.0.0	6.2.1
gnupg2	2.0.22	
gnupg2-minimal		2.3.7
gnutls		3.8.0
go-srpm-macros		3.2.0
gpgme	1.3.2	1.15.1
gpm-libs	1.20.7	1.20.7
grep	2.20	3.8
groff-base	1.22.2	1.22.4
grub2	2.06	
grub2-common	2.06	2.06
grub2-efi-aa64	2.06 (aarch64)	
grub2-efi-aa64-ec2	2.06 (aarch64)	2.06 (aarch64)
grub2-efi-aa64-mod ules	2.06 (noarch)	
grub2-efi-x64-ec2	2.06 (x86_64)	2.06 (x86_64)

Pacote	AMI do AL2	AMI do AL2023
grub2-pc	2.06 (x86_64)	
grub2-pc-modules	2.06 (noarch)	2.06
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.28	8.40
gssproxy	0.7.0	0.8.4
gzip	1.5	1.12
hardlink	1.3	
hibagent	1.1.0	
hostname	3.13	3.23
hunspell	1.3.2	1.7.0
hunspell-en	0.20121024	0.20140811.1
hunspell-en-GB	0.20121024	0.20140811.1
hunspell-en-US	0.20121024	0.20140811.1
hunspell-filesystem		1.7.0
hwdata	0.252	0.384
info	5.1	6.7
inih		49
initscripts	9.49.47	10.09
iproute	5.10.0	6.10.0

Pacote	AMI do AL2	AMI do AL2023
iptables	1.8.4	
iptables-libs	1.8.4	
iputils	20180629	20210202
irqbalance	1.7.0	1.9.0
jansson	(2.10)	2.14
jbigkit-libs	2.0	
jemalloc		5.2.1
jitterentropy		3.4.1
jq		1.7.1
json-c	0.11	0.14
kbd	1.15.5	2.4.0
kbd-legacy	1.15.5	
kbd-misc	1.15.5	2.4.0
kernel	5.10.228	6.1.112
kernel-libbpf		6.1.112
kernel-livepatch-r epo-s3		2023.6.20241031
kernel-srpm-macros		1,0
kernel-tools	5.10.228	6.1.112
keyutils	1.5.8	1.6.3
keyutils-libs	1.5.8	1.6.3

Pacote	AMI do AL2	AMI do AL2023
kmod	25	29
kmod-libs	25	29
kpartx	0.4.9	
kpatch-runtime	0.9.4	0.9.7
krb5-libs	1.15.1	1.21.3
langtable	0.0.31	
langtable-data	0.0.31	
langtable-python	0.0.31	
less	458	608
libacl	2.2.51	2.3.1
libaio	0.3.109	0.3.111
libarchive		3.7.4
libargon2		20171227
libassuan	2.1.0	2.5.5
libattr	2.4.46	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2.30.2	2.37.4
libcap	2.54	2.48
libcap-ng	0.7.5	0.8.2
libcbor		0.7.0

Pacote	AMI do AL2	AMI do AL2023
libcollection	0.7.0	0.7.0
libcom_err	1.42.9	1.46.5
libcomps		0.1.20
libconfig	1.4.9	1.7.2
libcroco	0.6.12	
libcrypt	2.26	
libcurl	8.3.0	
libcurl-minimal		8.5.0
libdaemon	0.14	
libdb	5.3.21	5.3.28
libdb-utils	5.3.21	
libdhash		0.5.0
libdnf		0.69.0
libdrm	2.4.97	
libdwarf	20130207 (x86_64)	
libeconf		0.4.0
libedit	3.0	3.1
libestr	0.1.9	
libev		4.33
libevent	2.0.21	2.1.12

Pacote	AMI do AL2	AMI do AL2023
libfastjson	0.99.4	
libfdisk	2.30.2	2.37.4
libffi	3.0.13	3.4.4
libfido2		1.10.0
libgcc	7.3.1	11.4.1
libgcrypt	1.5.3	1.10.2
libgomp	7.3.1	11.4.1
libgpg-error	1.12	1.42
libibverbs		48.0
libicu	50.2	
libidn	1.28	
libidn2	2.3.0	2.3.2
libini_config	1.3.1	1.3.1
libjpeg-turbo	2.0.90	
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libldb		2.6.2
libmaxminddb		1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.3	1.0.4

Pacote	AMI do AL2	AMI do AL2023
libmodulemd		2.13.0
libmount	2.30.2	2.37.4
libnetfilter_conntrack	1.0.6	
libnfnetworking	1.0.1	
libnfsidmap	0.25	2.5.4
libnhttp2	1.41.0	1.59.0
libnl3	3.2.28	3.5.0
libnl3-cli	3.2.28	
libpath_utils	0.2.1	0.2.1
libpcap	1.5.3	1.10.1
libpciaccess	0.14 (x86_64)	
libpipeline	1.2.3	1.5.3
libpkgconf		1.8.0
libpng	1.5.13	
libpsl	0.21.5	0.21.1
libpwquality	1.2.3	1.4.4
libref_array	0.1.5	0.1.5
librepo		1.14.5
libreport-fs		2.15.2
libseccomp	2.5.2	2.5.3

Pacote	AMI do AL2	AMI do AL2023
libselinux	2,5	3.4
libselinux-utils	2,5	3.4
libsemanage	2,5	3.4
libsepol	2,5	3.4
libsigsegv		2.13
libsmartcols	2.30.2	2.37.4
libsolv		0.7.22
libss	1.42.9	1.46.5
libssh2	1.4.3	
libsss_certmap		2.9.4
libsss_idmap	1.16.5	2.9.4
libsss_nss_idmap	1.16.5	2.9.4
libsss_sudo		2.9.4
libstdc++	7.3.1	11.4.1
libstoragemgmt	1.6.1	1.9.4
libstoragemgmt-python	1.6.1	
libstoragemgmt-python-clibs	1.6.1	
libsysfs	2.1.0	
libtalloc		2.3.4

Pacote	AMI do AL2	AMI do AL2023
libtasn1	4.10	4.19.0
libtdb		1.4.7
libteam	1.27	
libtevent		0.13.0
libtextstyle		0.21
libtiff	4.0.3	
libtirpc	0.2.4	1.3.3
libunistring	0.9.3	0.9.10
libuser	0.60	0.63
libutempter	1.1.6	1.2.1
libuuid	2.30.2	2.37.4
libuv		1.47.0
libverto	0.2.5	0.3.2
libverto-libev		0.3.2
libverto-libevent	0.2.5	
libwebp	0.3.0	
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libxml2-python	2.9.1	
libyaml	0.1.4	0.2.5

Pacote	AMI do AL2	AMI do AL2023
libzstd		1.5.5
linux-firmware-whe nce		20210208 (noarch)
lm_sensors-libs	3.4.0	3.6.0
lmdb-libs		0.9.29
logrotate	3.8.6	3.20.1
lsof	4.87	4.94.0
lua	5.1.4	
lua-libs		5.4.4
lua-srpm-macros		1
lvm2	2.02.187	
lvm2-libs	2.02.187	
lz4	1.7.5	
lz4-libs		1.9.4
make	3.82	
man-db	2.6.3	2.9.3
man-pages	3.53	5.10
man-pages-overrides	7.5.2	
mariadb-libs	5.5.68	
mdadm	4,0	
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)

Pacote	AMI do AL2	AMI do AL2023
mlocate	0.26	
mpfr		4.1.0
mtr	0.92	
nano	2.9.8	5,8
ncurses	6.0	6.2
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
nettle	2.7.1	3.8
net-tools	2.0	2.0
newt	0.52.15	0.52.21
newt-python	0.52.15	
nfs-utils	1.3.0	2.5.4
npth		1.6
nspr	4.35.0	4.35.0
nss	3.90.0	3.90.0
nss-pem	1.0.3	
nss-softokn	3.90.0	3.90.0
nss-softokn-freebl	3.90.0	3.90.0
nss-sysinit	3.90.0	3.90.0
nss-tools	3.90.0	

Pacote	AMI do AL2	AMI do AL2023
nss-util	3.90.0	3.90.0
ntsysv	1.7.4	1.15
numactl-libs	2.0.9	2.0.14
ocaml-srpm-macros		6
oniguruma		6.9.7.1
openblas-srpm-macros		2
openldap	2.4.44	2.4.57
openssh	7.4p1	8.7p1
openssh-clients	7.4p1	8.7p1
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-libs	1.0.2k	3.0.8
openssl-pkcs11		0.4.12
os-prober	1,58	1.77
p11-kit	0.23.22	0.24.1
p11-kit-trust	0.23.22	0.24.1
package-notes-srpm-macros		0.4
pam	1.1.8	1.5.1
parted	3.1	3.4
passwd	0.79	0.80

Pacote	AMI do AL2	AMI do AL2023
pciutils	3.5.1	3.7.0
pciutils-libs	3.5.1	3.7.0
pcre	8.32	
pcre2	10.23	10.40
pcre2-syntax		10.40
perl	5.16.3	
perl-Carp	1.26	1.50
perl-Class-Struct		0.66
perl-constant	1.27	1.33
perl-DynaLoader		1.47
perl-Encode	2.51	3.15
perl-Errno		1.30
perl-Exporter	5.68	5.74
perl-Fcntl		1.13
perl-File-Basename		2.85
perl-File-Path	2.09	2.18
perl-File-stat		1.09
perl-File-Temp	0.23.01	0.231.100
perl-Filter	1.49	
perl-Getopt-Long	2.40	2.52

Pacote	AMI do AL2	AMI do AL2023
perl-Getopt-Std		1.12
perl-HTTP-Tiny	0.033	0.078
perl-if		0.60.800
perl-interpreter		5.32.1
perl-I0		1.43
perl-IPC-Open3		1.21
perl-libs	5.16.3	5.32.1
perl-macros	5.16.3	
perl-MIME-Base64		3.16
perl-mro		1.23
perl-overload		1.31
perl-overloading		0.02
perl-parent	0.225	0.238
perl-PathTools	3.40	3.78
perl-Pod-Escapes	1.04	1.07
perl-podlators	2.5.1	4.14
perl-Pod-Perldoc	3.20	3.28.01
perl-Pod-Simple	3.28	3.42
perl-Pod-Usage	1.63	2.01
perl-POSIX		1.94

Pacote	AMI do AL2	AMI do AL2023
perl-Scalar-List-Utils	1.27	1.56
perl-SelectSaver		1.02
perl-Socket	2.010	2.032
perl-srpm-macros		1
perl-Storable	2.45	3.21
perl-subs		1.03
perl-Symbol		1.08
perl-Term-ANSIColor		5.01
perl-Term-Cap		1.17
perl-Text-ParseWords	3.29	3.30
perl-Text-Tabs+Wrap		2021.0726
perl-threads	1.87	
perl-threads-shared	1.43	
perl-Time-HiRes	1.9725	
perl-Time-Local	1.2300	1.300
perl-vars		1.05
pinentry	0.8.1	
pkgconf		1.8.0
pkgconfig	0.27.1	
pkgconf-m4		1.8.0

Pacote	AMI do AL2	AMI do AL2023
pkgconf-pkg-config		1.8.0
plymouth	0.8.9	
plymouth-core-libs	0.8.9	
plymouth-scripts	0.8.9	
pm-utils	1.4.1	
policycoreutils	2,5	3.4
policycoreutils-python-utils		3.4
popt	1.13	1.18
postfix	2.10.1	
procps-ng	3.3.10	3.3.17
protobuf-c		1.4.1
psacct	6.6.1	6.6.4
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpgme	0.3	
pyliblzma	0.5.3	
pystache	0.5.3	
python	2.7.18	

Pacote	AMI do AL2	AMI do AL2023
python2-botocore	1.18.6	
python2-colorama	0.3.9	
python2-cryptography	1.7.2	
python2-dateutil	2.6.1	
python2-futures	3.0.5	
python2-jmespath	0.9.3	
python2-jsonschema	2.5.1	
python2-oauthlib	2.0.1	
python2-pyasn1	0.1.9	
python2-rpm	4.11.3	
python2-rsa	3.4.1	
python2-s3transfer	0.3.3	
python2-setuptools	41.2.0	
python2-six	1.11.0	
python3	3.7.16	3.9.16
python3-attrs		20.3.0
python3-audit		3.0.6
python3-awscrt		0.19.19
python3-babel		2.9.1
python3-cffi		1.14.5

Pacote	AMI do AL2	AMI do AL2023
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-daemon	2.2.3	2.3.0
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils	0.14	0.16
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-idna		(2.10)
python3-jinja2		2.11.3
python3-jmespath		0.10.0
python3-jsonpatch		1.21
python3-jsonpointer		2.0
python3-jsonschema		3.2.0
python3-libcomps		0.1.20

Pacote	AMI do AL2	AMI do AL2023
python3-libdnf		0.69.0
python3-libs	3.7.16	3.9.16
python3-libselinux		3.4
python3-libsemanage		3.4
python3-libstorage mgmt		1.9.4
python3-lockfile	0.11.0	0.12.2
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip	20.2.2	
python3-pip-wheel		21.3.1
python3-ply		3.11
python3-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0.24
python3-pycparser		2.20
python3-pyrsistent		0.17.3
python3-pyserial		3.4

Pacote	AMI do AL2	AMI do AL2023
python3-pysocks		1.7.1
python3-pystache	0.5.4	
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2.25.1
python3-rpm		4.16.1.3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools	49.1.3	59.6.0
python3-setuptools-wheel		59.6.0
python3-simplejson	3.2.0	
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5
python-babel	0.9.6	
python-backports	1,0	

Pacote	AMI do AL2	AMI do AL2023
python-backports-s sl_match_hostname	3.5.0.1	
python-cffi	1.6.0	
python-chardet	2.2.1	
python-chevron		0.13.1
python-configobj	4.7.2	
python-daemon	1.6	
python-devel	2.7.18	
python-docutils	0.12	
python-enum34	1.0.4	
python-idna	2.4	
python-iniparse	0.4	
python-ipaddress	1.0.16	
python-jinja2	2.7.2	
python-jsonpatch	1.2	
python-jsonpointer	1.9	
python-jwcrypto	0.4.2	
python-kitchen	1.1.1	
python-libs	2.7.18	
python-lockfile	0.9.1	
python-markupsafe	0.11	

Pacote	AMI do AL2	AMI do AL2023
python-pillow	2.0.0	
python-ply	3.4	
python-pycparser	2.14	
python-pycurl	7.19.0	
python-repoze-lru	0.4	
python-requests	2.6.0	
python-simplejson	3.2.0	
python-srpm-macros		3.9
python-urlgrabber	3.10	
python-urllib3	1.25.9	
pyxattr	0.5.1	
PyYAML	3.10	
qrencode-libs	3.4.1	
quota	4.01	4.06
quota-nls	4.01	4.06
rdate	1.4	
readline	6.2	8.1
rng-tools	6.8	6.14
rootfiles	8.1	8.1
rpcbind	0.2.0	1.2.6

Pacote	AMI do AL2	AMI do AL2023
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
rsync	3.1.2	3.2.6
rsyslog	8.24.0	
rust-srpm-macros		21
sbsigntools		0.9.4
scl-utils	20130529	
screen	4.1.0	4.8.0
sed	4.2.2	4.8
selinux-policy	3.13.1	38.1.45
selinux-policy-targeted	3.13.1	38.1.45
setserial	2.17	
setup	2.8.71	2.13.7
setuptools	1.19.11	
sgpio	1.2.0.10	

Pacote	AMI do AL2	AMI do AL2023
shadow-utils	4.1.5.1	4,9
shared-mime-info	1.8	
slang	2.2.4	2.3.2
sqlite	3.7.17	
sqlite-libs		3.40.0
sssd-client	1.16.5	2.9.4
sssd-common		2.9.4
sssd-kcm		2.9.4
sssd-nfs-idmap		2.9.4
strace	4.26	6.8
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1,0
sysstat	10.1.5	12.5.6
systemd	219	252.23
systemd-libs	219	252.23
systemd-networkd		252.23
systemd-pam		252.23
systemd-resolved		252.23
systemd-sysv	219	
systemd-udev		252.23

Pacote	AMI do AL2	AMI do AL2023
system-release	2	2023.6.20241031
systemtap-runtime	4.5	4.8
sysvinit-tools	2.88	
tar	1.26	1.34
tbb		2020.3
tcp_wrappers	7.6	
tcp_wrappers-libs	7.6	
tcpdump	4.9.2	4.99.1
tcsh	6.18.01	6.24.07
teamd	1.27	
time	1,7	1.9
traceroute	2.0.22	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	1.1.2	2.2
usermode	1.111	
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2.30.2	2.37.4
util-linux-core		2.37.4

Pacote	AMI do AL2	AMI do AL2023
vim-common	9.0.2153	9.0.2153
vim-data	9.0.2153	9.0.2153
vim-enhanced	9.0.2153	9.0.2153
vim-filesystem	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153
virt-what	1.18	
wget	1.14	1.21.3
which	2.20	2.21
words	3.0	3.0
xfsdump	3.1.8	3.1.11
xfsprogs	5.0.0	5.18.0
xxd	9.0.2153	9.0.2153
xxhash-libs		0.8.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yajl	2.0.4	
yum	3.4.3	4.14.0
yum-langpacks	0.4.2	
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	

Pacote	AMI do AL2	AMI do AL2023
yum-utils	1.1.31	
zip	3.0	3.0
zlib	1.2.7	1.2.11
zram-generator		1.1.2
zram-generator-defaults		1.1.2
zstd		1.5.5

Comparação entre pacotes instalados no Amazon Linux 2 e nas AMIs do Amazon Linux 2023

Uma comparação dos RPMs presentes nas AMIs mínimas do Amazon Linux 2 e do AL2023.

Pacote	Mínima do AL2	Mínima do AL2023
acl	2.2.51	
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.5.1
amazon-linux-extras	2.0.3	
amazon-linux-repo-s3		2023.6.20241031
amazon-linux-sb-keys		2023.1
amd-ucode-firmware	20200421 (noarch)	20210208 (noarch)
audit	2.8.1	3.0.6

Pacote	Mínima do AL2	Mínima do AL2023
audit-libs	2.8.1	3.0.6
authconfig	6.2.8	
awscli-2		2.15.30
basesystem	10.0	11
bash	4.2.46	5.2.15
bind-export-libs	9.11.4	
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.68	2023.2.68
checkpolicy		3.4
chkconfig	1.7.4	
chrony	4.2	4.3
cloud-init	19.3	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart	0.31	0.31
coreutils	8.22	8.32
coreutils-common		8.32
cpio	2.12	2.13
cracklib	2.9.0	2.9.6
cracklib-dicts	2.9.0	2.9.6
cronie	1.4.11	

Pacote	Mínima do AL2	Mínima do AL2023
cronie-anacron	1.4.11	
crontabs	1.11	
crypto-policies		20220428
cryptsetup-libs	1.7.4	2.6.1
curl	8.3.0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	2.1.27
dbus	1.10.24	1.12.28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.10.24	1.12.28
device-mapper	1.02.170	1.02.185
device-mapper-libs	1.02.170	1.02.185
dhclient	4.2.5	
dhcp-common	4.2.5	
dhcp-libs	4.2.5	
diffutils	3.3	3.8
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2

Pacote	Mínima do AL2	Mínima do AL2023
dnf-plugins-core		4.3.0
dnf-plugin-support-info		1.2
dracut	033	055
dracut-config-ec2	2.0	3.0
dracut-config-generic	033	055
e2fsprogs	1.42.9	1.46.5
e2fsprogs-libs	1.42.9	1.46.5
ec2-utils	1.2	2.2.0
efibootmgr	15 (aarch64)	
efi-filesystem		5
efivar		38
efivar-libs	31 (aarch64)	38
elfutils-default-yama-scope	0.176	0.188
elfutils-libelf	0.176	0.188
elfutils-libs	0.176	0.188
expat	2.1.0	2.5.0
file	5.11	5.39
file-libs	5.11	5.39

Pacote	Mínima do AL2	Mínima do AL2023
filesystem	3.2	3.14
findutils	4.5.11	4.8.0
fipscheck	1.4.1	
fipscheck-lib	1.4.1	
freetype	2.8	
fuse-libs	2.9.2	2.9.9
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1.19
gdisk	0.8.10	1.0.8
gettext	0.19.8.1	0.21
gettext-libs	0.19.8.1	0.21
glib2	2.56.1	2.74.7
glibc	2.26	2.34
glibc-all-langpacks	2.26	2.34
glibc-common	2.26	2.34
glibc-locale-source	2.26	2.34
glibc-minimal-lang pack	2.26	
gmp	6.0.0	6.2.1
gnupg2	2.0.22	

Pacote	Mínima do AL2	Mínima do AL2023
<u>gnupg2-minimal</u>		2.3.7
gnutls		3.8.0
gpgme	1.3.2	1.15.1
grep	2.20	3.8
groff-base	1.22.2	1.22.4
grub2	2.06	
grub2-common	2.06	2.06
grub2-efi-aa64	2.06 (aarch64)	
grub2-efi-aa64-ec2	2.06 (aarch64)	2.06 (aarch64)
grub2-efi-aa64-modules	2.06 (noarch)	
grub2-efi-x64-ec2	2.06 (x86_64)	2.06 (x86_64)
grub2-pc	2.06 (x86_64)	
grub2-pc-modules	2.06 (noarch)	2.06
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.28	8.40
gzip	1.5	1.12
hardlink	1.3	
hostname	3.13	3.23
hwdata		0.384

Pacote	Mínima do AL2	Mínima do AL2023
info	5.1	
inih		49
initscripts	9.49.47	10.09
iproute	5.10.0	6.10.0
iptables	1.8.4	
iptables-lib	1.8.4	
iputils	20180629	20210202
irqbalance	1.7.0	1.9.0
jansson		2.14
jitterentropy		3.4.1
jq		1.7.1
json-c		0.14
kbd		2.4.0
kbd-misc		2.4.0
kernel	4.14.355	6.1.112
kernel-libbpf		6.1.112
kernel-livepatch-r epo-s3		2023.6.20241031
keyutils-lib	1.5.8	1.6.3
kmod	25	29
kmod-lib	25	29

Pacote	Mínima do AL2	Mínima do AL2023
kpartx	0.4.9	
krb5-libs	1.15.1	1.21.3
less	458	608
libacl	2.2.51	2.3.1
libarchive		3.7.4
libargon2		20171227
libassuan	2.1.0	2.5.5
libattr	2.4.46	2.5.1
libblkid	2.30.2	2.37.4
libcap	2.54	2.48
libcap-ng	0.7.5	0.8.2
libcbor		0.7.0
libcom_err	1.42.9	1.46.5
libcomps		0.1.20
libcroco	0.6.12	
libcrypt	2.26	
libcurl	8.3.0	
libcurl-minimal		8.5.0
libdb	5.3.21	5.3.28
libdb-utils	5.3.21	

Pacote	Mínima do AL2	Mínima do AL2023
libdnf		0.69.0
libeconf		0.4.0
libedit	3.0	3.1
libestr	0.1.9	
libfastjson	0.99.4	
libfdisk	2.30.2	2.37.4
libffi	3.0.13	3.4.4
libfido2		1.10.0
libgcc	7.3.1	11.4.1
libgcrypt	1.5.3	1.10.2
libgomp	7.3.1	11.4.1
libgpg-error	1.12	1.42
libicu	50.2	
libidn	1.28	
libidn2	2.3.0	2.3.2
libkcapi		1.4.0
libkcapi-hmacalc		1.4.0
libmetalink	0.1.3	
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0

Pacote	Mínima do AL2	Mínima do AL2023
libmount	2.30.2	2.37.4
libnetfilter_conntrack	1.0.6	
libnfnetlink	1.0.1	
libnhttp2	1.41.0	1.59.0
libpcap	1.5.3	
libpipeline	1.2.3	1.5.3
libpng	1.5.13	
libpsl	0.21.5	0.21.1
libpwquality	1.2.3	1.4.4
librepo		1.14.5
libreport-filesystem		2.15.2
libseccomp	2.5.2	2.5.3
libselinux	2,5	3.4
libselinux-utils	2,5	3.4
libsemanage	2,5	3.4
libsepol	2,5	3.4
libsigsegv		2.13
libsmartcols	2.30.2	2.37.4
libsolv		0.7.22
libss	1.42.9	1.46.5

Pacote	Mínima do AL2	Mínima do AL2023
libssh2	1.4.3	
libstdc++	7.3.1	11.4.1
libsysfs	2.1.0	
libtasn1	4.10	4.19.0
libtextstyle		0.21
libunistring	0.9.3	0.9.10
libuser	0.60	0.63
libutempter	1.1.6	1.2.1
libuuid	2.30.2	2.37.4
libverto	0.2.5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml	0.1.4	0.2.5
libzstd		1.5.5
linux-firmware-whe nce		20210208 (noarch)
logrotate	3.8.6	3.20.1
lua	5.1.4	
lua-libs		5.4.4
lz4	1.7.5	
lz4-libs		1.9.4

Pacote	Mínima do AL2	Mínima do AL2023
make	3.82	
man-db	2.6.3	2.9.3
mariadb-libs	5.5.68	
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mpfr		4.1.0
ncurses	6.0	6.2
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
nettle	2.7.1	3.8
net-tools	2.0	2.0
newt	0.52.15	
newt-python	0.52.15	
npth		1.6
nspr	4.35.0	
nss	3.90.0	
nss-pem	1.0.3	
nss-softokn	3.90.0	
nss-softokn-freebl	3.90.0	
nss-sysinit	3.90.0	
nss-tools	3.90.0	

Pacote	Mínima do AL2	Mínima do AL2023
nss-util	3.90.0	
numactl-libs	2.0.9	2.0.14
oniguruma		6.9.7.1
openldap	2.4.44	2.4.57
openssh	7.4p1	8.7p1
openssh-clients	7.4p1	8.7p1
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-libs	1.0.2k	3.0.8
openssl-pkcs11		0.4.12
os-prober	1,58	1.77
p11-kit	0.23.22	0.24.1
p11-kit-trust	0.23.22	0.24.1
pam	1.1.8	1.5.1
passwd	0.79	0.80
pciutils		3.7.0
pciutils-libs		3.7.0
pcre	8.32	
pcre2	10.23	10.40
pcre2-syntax		10.40

Pacote	Mínima do AL2	Mínima do AL2023
pinentry	0.8.1	
pkgconfig	0.27.1	
polycoreutils	2.5	3.4
popt	1.13	1.18
postfix	2.10.1	
procps-ng	3.3.10	3.3.17
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpgme	0.3	
pyliblzma	0.5.3	
python	2.7.18	
python2-cryptography	1.7.2	
python2-jsonschema	2.5.1	
python2-oauthlib	2.0.1	
python2-pyasn1	0.1.9	
python2-rpm	4.11.3	
python2-setuptools	41.2.0	
python2-six	1.11.0	
python3		3.9.16

Pacote	Mínima do AL2	Mínima do AL2023
python3-attrs		20.3.0
python3-audit		3.0.6
python3-awscrt		0.19.19
python3-babel		2.9.1
python3-cffi		1.14.5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils		0.16
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-idna		(2.10)
python3-jinja2		2.11.3
python3-jmespath		0.10.0

Pacote	Mínima do AL2	Mínima do AL2023
python3-jsonpatch		1.21
python3-jsonpointer		2.0
python3-jsonschema		3.2.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs		3.9.16
python3-libselenium		3.4
python3-libsemanage		3.4
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip-wheel		21.3.1
python3-ply		3.11
python3-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0.24
python3-pycparser		2.20
python3-pyrsistent		0.17.3
python3-pyserial		3.4

Pacote	Mínima do AL2	Mínima do AL2023
python3-pysocks		1.7.1
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2.25.1
python3-rpm		4.16.1.3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59.6.0
python3-setuptools-wheel		59.6.0
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5
python-babel	0.9.6	
python-backports	1,0	
python-backports-ssl-match_hostname	3.5.0.1	
python-cffi	1.6.0	

Pacote	Mínima do AL2	Mínima do AL2023
python-chardet	2.2.1	
python-configobj	4.7.2	
python-devel	2.7.18	
python-enum34	1.0.4	
python-idna	2.4	
python-iniparse	0.4	
python-ipaddress	1.0.16	
python-jinja2	2.7.2	
python-jsonpatch	1.2	
python-jsonpointer	1.9	
python-jwcrypto	0.4.2	
python-libs	2.7.18	
python-markupsafe	0.11	
python-ply	3.4	
python-pycparser	2.14	
python-pycurl	7.19.0	
python-repoze-lru	0.4	
python-requests	2.6.0	
python-urlgrabber	3.10	
python-urllib3	1.25.9	

Pacote	Mínima do AL2	Mínima do AL2023
pyxattr	0.5.1	
PyYAML	3.10	
qrencode-libs	3.4.1	
readline	6.2	8.1
rng-tools	6.8	6.14
rootfiles	8.1	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
rsyslog	8.24.0	
sbsigntools		0.9.4
sed	4.2.2	4.8
selinux-policy	3.13.1	38.1.45
selinux-policy-targeted	3.13.1	38.1.45
setup	2.8.71	2.13.7
shadow-utils	4.1.5.1	4,9

Pacote	Mínima do AL2	Mínima do AL2023
shared-mime-info	1.8	
slang	2.2.4	
sqlite	3.7.17	
sqlite-libs		3.40.0
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1,0
systemd	219	252.23
systemd-libs	219	252.23
systemd-networkd		252.23
systemd-pam		252.23
systemd-resolved		252.23
systemd-sysv	219	
systemd-udev		252.23
system-release	2	2023.6.20241031
sysvinit-tools	2.88	
tar	1.26	1.34
tcp_wrappers-libs	7.6	
tzdata	2024a	2024a
update-motd	1.1.2	2.2
userspace-rcu		0.12.1

Pacote	Mínima do AL2	Mínima do AL2023
ustr	1.0.4	
util-linux	2.30.2	2.37.4
util-linux-core		2.37.4
vim-data	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153
which	2.20	2.21
xfsprogs	5.0.0	5.18.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	
zlib	1.2.7	1.2.11
zram-generator		1.1.2
zram-generator-defaults		1.1.2
zstd		1.5.5

Compare de comparação de pacotes instalados em imagens de contêiner de base do Amazon Linux 2023 e do Amazon Linux 2023

Uma comparação dos RPMs presentes nas imagens básicas de contêiner do Amazon Linux 2 e do AL2023.

Pacote	Contêiner do AL2	Contêiner do AL2023
alternatives		1.15
amazon-linux-extras	2.0.3	
amazon-linux-repo-cdn		2023.6.20241031
audit-libs		3.0.6
basesystem	10.0	11
bash	4.2.46	5.2.15
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.68	2023.2.68
chkconfig	1.7.4	
coreutils	8.22	
coreutils-single		8.32
cpio	2.12	
crypto-policies		20220428
curl	8.3.0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	

Pacote	Contêiner do AL2	Contêiner do AL2023
diffutils	3.3	
dnf		4.14.0
dnf-data		4.14.0
elfutils-default-yama-scope		0.188
elfutils-libelf	0.176	0.188
elfutils-libs		0.188
expat	2.1.0	2.5.0
file-libs	5.11	5.39
filesystem	3.2	3.14
findutils	4.5.11	
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1.19
glib2	2.56.1	2.74.7
glibc	2.26	2.34
glibc-common	2.26	2.34
glibc-langpack-en	2.26	
glibc-minimal-langpack	2.26	2.34
gmp	6.0.0	6.2.1

Pacote	Contêiner do AL2	Contêiner do AL2023
gnupg2	2.0.22	
gnupg2-minimal		2.3.7
gpgme	1.3.2	1.15.1
grep	2.20	3.8
info	5.1	
json-c		0.14
keyutils-libs	1.5.8	1.6.3
krb5-libs	1.15.1	1.21.3
libacl	2.2.51	2.3.1
libarchive		3.7.4
libassuan	2.1.0	2.5.5
libattr	2.4.46	2.5.1
libblkid	2.30.2	2.37.4
libcap	2.54	2.48
libcap-ng		0.8.2
libcom_err	1.42.9	1.46.5
libcomps		0.1.20
libcrypt	2.26	
libcurl	8.3.0	
libcurl-minimal		8.5.0

Pacote	Contêiner do AL2	Contêiner do AL2023
libdb	5.3.21	
libdb-utils	5.3.21	
libdnf		0.69.0
libffi	3.0.13	3.4.4
libgcc	7.3.1	11.4.1
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.12	1.42
libidn2	2.3.0	2.3.2
libmetalink	0.1.3	
libmodulemd		2.13.0
libmount	2.30.2	2.37.4
libnghttp2	1.41.0	1.59.0
libpsl	0.21.5	0.21.1
librepo		1.14.5
libreport-filessystem		2.15.2
libselinux	2,5	3.4
libsepol	2,5	3.4
libsigsegv		2.13
libsmartcols		2.37.4

Pacote	Contêiner do AL2	Contêiner do AL2023
libsolv		0.7.22
libssh2	1.4.3	
libstdc++	7.3.1	11.4.1
libtasn1	4.10	4.19.0
libunistring	0.9.3	0.9.10
libuuid	2.30.2	2.37.4
libverto	0.2.5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml		0.2.5
libzstd		1.5.5
lua	5.1.4	
lua-libs		5.4.4
lz4-libs		1.9.4
mpfr		4.1.0
ncurses	6.0	
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
npth		1.6
nspr	4.35.0	

Pacote	Contêiner do AL2	Contêiner do AL2023
nss	3.90.0	
nss-pem	1.0.3	
nss-softokn	3.90.0	
nss-softokn-freebl	3.90.0	
nss-sysinit	3.90.0	
nss-tools	3.90.0	
nss-util	3.90.0	
openldap	2.4.44	
openssl-lib	1.0.2k	3.0.8
p11-kit	0.23.22	0.24.1
p11-kit-trust	0.23.22	0.24.1
pcre	8.32	
pcre2		10.40
pcre2-syntax		10.40
pinentry	0.8.1	
popt	1.13	1.18
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpgme	0.3	
pyliblzma	0.5.3	

Pacote	Contêiner do AL2	Contêiner do AL2023
python	2.7.18	
python2-rpm	4.11.3	
python3		3.9.16
python3-dnf		4.14.0
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs		3.9.16
python3-pip-wheel		21.3.1
python3-rpm		4.16.1.3
python3-setuptools-wheel		59.6.0
python-iniparse	0.4	
python-libs	2.7.18	
python-pycurl	7.19.0	
python-urlgrabber	3.10	
pyxattr	0.5.1	
readline	6.2	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3

Pacote	Contêiner do AL2	Contêiner do AL2023
rpm-libs	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
sed	4.2.2	4.8
setup	2.8.71	2.13.7
shared-mime-info	1.8	
sqlite	3.7.17	
sqlite-libs		3.40.0
system-release	2	2023.6.20241031
tzdata	2024a	2024a
vim-data	9.0.2153	
vim-minimal	9.0.2153	
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-ovl	1.1.31	
yum-plugin-priorities	1.1.31	
zlib	1.2.7	1.2.11

Comparação entre o AL1 e o AL2023

Os tópicos a seguir descrevem as principais diferenças entre o AL1 e o AL2023 que ainda não foram abordadas pela [comparação com o AL2](#).

Note

O AL1 chegou ao fim de sua vida útil em 31 de dezembro de 2023 e não receberá mais atualizações de segurança ou correções de bugs a partir de 1.º de janeiro de 2024. Para obter mais informações sobre o fim da vida útil e o suporte para manutenção do AL1, consulte a postagem do blog [Update on Amazon Linux AMI end-of-life](#). Recomendamos que você atualize as aplicações para o AL2023, o que inclui suporte de longo prazo até 2028.

Tópicos

- [Suporte para cada versão](#)
- [systemd substitui upstart como sistema init](#)
- [Python 2.6 e 2.7 foram substituídos pelo Python 3](#)
- [OpenJDK 8 como o JDK mais antigo](#)
- [AL2023 alterações no kernel do Amazon Linux \(1\) AL1](#)
- [Comparar pacotes instalados nas AMIs do Amazon Linux 1 \(AL1\) e do Amazon Linux 2023](#)
- [Comparação entre pacotes instalados nas AMIs mínimas do Amazon Linux 1 \(AL1\) e do Amazon Linux 2023](#)
- [Comparar pacotes instalados nas imagens de contêiner base do Amazon Linux 1 \(AL1\) e do Amazon Linux 2023](#)

Suporte para cada versão

Para o AL2023, oferecemos cinco anos de suporte a partir da data de lançamento. O AL1 encerrou o suporte padrão em 31 de dezembro de 2020 e encerrou o suporte para manutenção em 31 de dezembro de 2023.

Para obter mais informações, consulte [Cadência de lançamento](#).

systemd substitui upstart como sistema init

No AL2, upstart foi substituído por systemd como sistema init. O AL2023 também usa systemd como sistema init, adotando ainda mais novos recursos e funcionalidades do systemd.

Python 2.6 e 2.7 foram substituídos pelo Python 3

Embora o AL1 tenha marcado o Python 2.6 como EOL com o lançamento 2018.03, os pacotes ainda estavam disponíveis nos repositórios para instalação. O AL2 foi distribuído com o Python 2.7 como versão mais antiga compatível do Python, e o AL2023 conclui a transição para o Python 3. Nenhuma versão 2.x do Python está incluída nos repositórios do AL2023.

Para obter mais informações sobre Python no Amazon Linux, consulte [Python no AL2023](#).

OpenJDK 8 como o JDK mais antigo

O AL2023 é fornecido com o [Amazon Corretto](#) como o Java Development Kit (JDK) padrão (e único). Todos os pacotes baseados em Java no AL2023 são desenvolvidos com Amazon Corretto 17.

No AL1, o OpenJDK 1.6.0 (java-1.6.0-openjdk) chegou ao fim da vida útil com o primeiro lançamento 2018.03, e o OpenJDK 1.7.0 (java-1.7.0-openjdk) chegou ao fim da vida útil em meados de 2020, embora ambos estivessem disponíveis nos repositórios do AL1. A versão mais antiga do OpenJDK disponível no AL2023 é o OpenJDK 8, fornecido pelo Amazon Corretto 8.

AL2023 alterações no kernel do Amazon Linux (1) AL1

Kernel Live Patching

Tanto o AL2 023 quanto o AL2 adicionam suporte para a funcionalidade de correção ao vivo do kernel. Isso permite corrigir vulnerabilidades de segurança críticas e importantes no kernel do Linux sem reinicialização ou tempo de inatividade. Para obter mais informações, consulte [Aplicação de patches do kernel em tempo real no AL2023](#).

Suporte ao sistema de arquivos do kernel

Houve várias mudanças nos sistemas de arquivos nos quais o kernel AL1 suportará a montagem, junto com mudanças nos esquemas de particionamento que o kernel analisará.

Opção do CONFIG	AL1/4.14/ x86_64	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_AFS_FS</u>	m	n	n	n	n
<u>CONFIG_AFS_RRPC</u>	m	n	n	n	n
<u>CONFIG_BSD_DISKLABEL</u>	y	n	n	n	n
<u>CONFIG_CRAMFS</u>	m	n	n	n	n
<u>CONFIG_CRAMFS_BLOCKDEV</u>	N/D	N/D	N/D	N/D	N/D
<u>CONFIG_DM_CLONE</u>	N/D	n	n	n	n
<u>CONFIG_DM_ERA</u>	n	n	n	n	n
<u>CONFIG_DM_INTEGRITY</u>	m	m	m	m	m
<u>CONFIG_DM_LOG_WRITES</u>	n	m	m	m	m
<u>CONFIG_DM_SWITCH</u>	n	n	n	n	n
<u>CONFIG_DM_VERITY</u>	n	m	m	m	m

Opção do CONFIG	AL1/4.14/ x86_64	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_ECRYPT_FS</u>	m	n	n	n	n
<u>CONFIG_EXFAT_FS</u>	N/D	m	m	m	m
<u>CONFIG_EXT2_FS</u>	m	n	n	n	n
<u>CONFIG_EXT3_FS</u>	m	n	n	n	n
<u>CONFIG_GFS2_FS</u>	n	n	n	n	n
<u>CONFIG_HFSPLUS_FS</u>	m	n	n	n	n
<u>CONFIG_HFS_FS</u>	m	n	n	n	n
<u>CONFIG_JFS_FS</u>	n	n	n	n	n
<u>CONFIG_LDM_PARTITION</u>	y	n	n	n	n
<u>CONFIG_MAC_PARTITION</u>	y	n	n	n	n
<u>CONFIG_NFS_V2S</u>	m	n	n	n	n

Opção do CONFIG	AL1/4.14/ x86_64	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_NTFS_FS</u>	m	n	n	n	n
<u>CONFIG_ROMFS_FS</u>	m	n	n	n	n
<u>CONFIG_SOLARIS_X86_PARTITION</u>	y	n	n	n	n
<u>CONFIG_SQUASHFS_ZSTD</u>	y	y	y	y	y
<u>CONFIG_SUN_PARTITION</u>	y	n	n	n	n

Alterações na configuração do kernel com foco na segurança

Opção do CONFIG	AL1/4.14/ x86_64	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_BUG_ON_DATA_CORRUPTION</u>	y	y	y	y	y
<u>CONFIG_DEFAULT_FAULT_MMAP_MIN_ADDR</u>	4096	65536	65536	65536	65536

Opção do CONFIG	AL1/4.14/ x86_64	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_DE VMEM</u>	y	n	n	n	n
<u>CONFIG_DE VPORT</u>	y	n	n	n	n
<u>CONFIG_FO RTIFY_SOU RCE</u>	y	y	y	y	y
<u>CONFIG_HA RDENED_US ERCOPY_FA LLBACK</u>	N/D	N/D	N/D	N/D	N/D
<u>CONFIG_IN IT_ON_ALL OC_DEFAULT T_ON</u>	N/D	n	n	n	n
<u>CONFIG_IN IT_ON_FRE E_DEFAULT _ON</u>	N/D	n	n	n	n
<u>CONFIG_IO MMU_DEFAU LT_DMA_ST RICT</u>	N/D	n	n	n	n
<u>CONFIG_LD ISC_AUTOL OAD</u>	y	n	n	n	n

Opção do CONFIG	AL1/4.14/ x86_64	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_SC HED_CORE</u>	N/D	N/D	y	N/D	y
<u>CONFIG_SC HED_STACK _END_CHEC K</u>	y	y	y	y	y
<u>CONFIG_SE CURITY_DM ESG_RESTR ICT</u>	n	y	y	y	y
<u>CONFIG_SE CURITY_SE LINUX_DIS ABLE</u>	y	n	n	N/D	N/D
<u>CONFIG_SH UFFLE_PAG E_ALLOCAT OR</u>	N/D	y	y	y	y
<u>CONFIG_SL AB_FREELI ST_HARDEN ED</u>	y	y	y	y	y
<u>CONFIG_SL AB_FREELI ST_RANDOM</u>	n	y	y	y	y

Outras alterações na configuração do kernel

Opção do CONFIG	AL1/4.14/ x86_64	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_HZ</u>	250	100	100	100	100
<u>CONFIG_NR_CPUS</u>	8192	4096	8192	4096	8192
<u>CONFIG_PANIC_ON_OOPS</u>	n	y	y	y	y
<u>CONFIG_PANIC_ON_OOPS_VALUE</u>	0	1	1	1	1
<u>CONFIG_PREEMPT</u>	m	n	n	n	n
<u>CONFIG_SMP</u>	m	n	n	n	n
<u>CONFIG_XEN_PV</u>	y	N/D	n	N/D	n

Comparar pacotes instalados nas AMIs do Amazon Linux 1 (AL1) e do Amazon Linux 2023

Uma comparação dos RPMs presentes nas AMIs padrão do AL1 e do AL2023.

Pacote	AMI do AL1	AMI do AL2023
acl	2.2.49	2.3.1
acpid	2.0.19	2.0.32

Pacote	AMI do AL1	AMI do AL2023
alsa-lib	1.0.22	
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.5.1
amazon-linux-repo-s3		2023.6.20241031
amazon-linux-sb-keys		2023.1
amazon-rpm-config		228
amazon-ssm-agent	3.2.2222.0	3.3.987.0
amd-ucode-firmware		20210208
at	3.1.10	3.1.23
attr	2.4.46	2.5.1
audit	2.6.5	3.0.6
audit-libs	2.6.5	3.0.6
authconfig	6.2.8	
aws-amitools-ec2	1.5.13	
aws-cfn-bootstrap	1.4	2.0
aws-cli	1.18.107	
awscli-2		2.15.30
basesystem	10.0	11
bash	4.2.46	5.2.15

Pacote	AMI do AL1	AMI do AL2023
bash-completion		2.11
bc	1.06.95	1.07.1
bind-libs	9.8.2	9.18.28
bind-license		9.18.28
bind-utils	9.8.2	9.18.28
binutils	2.27	2.39
boost-filesystem		1.75.0
boost-system		1.75.0
boost-thread		1.75.0
bzip2	1.0.6	1.0.8
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.62	2023.2.68
c-ares		1.19.1
checkpolicy	2.1.10	3.4
chkconfig	1.3.49.3	1.15
chrony		4.3
cloud-disk-utils	0.27	
cloud-init	0.7.6	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart		0.31

Pacote	AMI do AL1	AMI do AL2023
copy-jdk-configs	3.3	
coreutils	8.22	8.32
coreutils-common		8.32
cpio	(2.10)	2.13
cracklib	2.8.16	2.9.6
cracklib-dicts	2.8.16	2.9.6
cronie	1.4.4	
cronie-anacron	1.4.4	
crontabs	1.10	1.11
crypto-policies		20220428
crypto-policies-scripts		20220428
cryptsetup	1.6.7	2.6.1
cryptsetup-libs	1.6.7	2.6.1
curl	7.61.1	
curl-minimal		8.5.0
cyrus-sasl	2.1.23	
cyrus-sasl-lib	2.1.23	2.1.27
cyrus-sasl-plain	2.1.23	2.1.27
dash	0.5.5.1	
db4	4.7.25	

Pacote	AMI do AL1	AMI do AL2023
db4-utils	4.7.25	
dbus	1.6.12	1.12.28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.6.12	1.12.28
dejavu-fonts-common	2.33	
dejavu-sans-fonts	2.33	
dejavu-serif-fonts	2.33	
device-mapper	1.02.135	1.02.185
device-mapper-event	1.02.135	
device-mapper-event-libs	1.02.135	
device-mapper-libs	1.02.135	1.02.185
device-mapper-persistent-data	0.6.3	
dhclient	4.1.1	
dhcp-common	4.1.1	
diffutils	3.3	3.8
dmraid	1.0.0.rc16	
dmraid-events	1.0.0.rc16	
dnf		4.14.0

Pacote	AMI do AL1	AMI do AL2023
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0
dnf-plugin-support-info		1.2
dnf-utils		4.3.0
dosfstools		4.2
dracut	004	055
dracut-config-ec2		3.0
dracut-config-generic		055
dracut-modules-growroot	0.20	
dump	0.4	
dwz		0.14
dyninst		10.2.1
e2fsprogs	1.43.5	1.46.5
e2fsprogs-libs	1.43.5	1.46.5
ec2-hibinit-agent	1.0.0	1.0.8
ec2-instance-connect		1.1

Pacote	AMI do AL1	AMI do AL2023
ec2-instance-connect-selinux		1.1
ec2-net-utils	0.7	
ec2-utils	0.7	2.2.0
ed	1.1	1.14.2
efi-filesystem		5
efi-srpm-macros		5
efivar		38
efivar-libs		38
elfutils-debuginfod-client		0.188
elfutils-default-yama-scope		0.188
elfutils-libelf	0.168	0.188
elfutils-libs		0.188
epel-release	6	
ethtool	3.15	5.15
expat	2.1.0	2.5.0
file	5.37	5.39
file-libs	5.37	5.39
filesystem	2.4.30	3.14

Pacote	AMI do AL1	AMI do AL2023
findutils	4.4.2	4.8.0
fipscheck	1.3.1	
fipscheck-lib	1.3.1	
fontconfig	2.8.0	
fontpackages-files ystem	1.41	
fonts-srpm-macros		2.0.5
freetype	2.3.11	
fstrm		0.6.1
fuse-libs	2.9.4	2.9.9
gawk	3.1.7	5.1.0
gdbm	1.8.0	
gdbm-libs		1.19
gdisk	0.8.10	1.0.8
generic-logos	17.0.0	
get_reference_source	1.2	
gettext		0.21
gettext-libs		0.21
ghc-srpm-macros		1.5.0
giflib	4.1.6	
glib2	2.36.3	2.74.7

Pacote	AMI do AL1	AMI do AL2023
glibc	2.17	2.34
glibc-all-langpacks		2.34
glibc-common	2.17	2.34
glibc-gconv-extra		2.34
glibc-locale-source		2.34
gmp	6.0.0	6.2.1
gnupg2	2.0.28	
gnupg2-minimal		2.3.7
gnutls		3.8.0
go-srpm-macros		3.2.0
gpgme	1.4.3	1.15.1
gpm-libs	1.20.6	1.20.7
grep	2.20	3.8
groff	1.22.2	
groff-base	1.22.2	1.22.4
grub	0.97	
grub2-common		2.06
grub2-efi-x64-ec2		2.06
grub2-pc-modules		2.06
grub2-tools		2.06

Pacote	AMI do AL1	AMI do AL2023
grub2-tools-minimal		2.06
grubby	7.0.15	8.40
gssproxy		0.8.4
gzip	1.5	1.12
hesiod	3.1.0	
hibagent	1.0.0	
hmaccalc	0.9.12	
hostname		3.23
hunspell		1.7.0
hunspell-en		0.20140811.1
hunspell-en-GB		0.20140811.1
hunspell-en-US		0.20140811.1
hunspell-filesystem		1.7.0
hwdata	0.233	0.384
info	5.1	6.7
inih		49
initscripts	9.03.58	10.09
iproute	4.4.0	6.10.0
iptables	1.4.21	
iputils	20121221	20210202

Pacote	AMI do AL1	AMI do AL2023
irqbalance	1.5.0	1.9.0
jansson		2.14
java-1.7.0-openjdk	1.7.0.321	
javapackages-tools	0.9.1	
jemalloc		5.2.1
jitterentropy		3.4.1
jpackage-utils	1.7.5	
jq		1.7.1
json-c		0.14
kbd	1.15	2.4.0
kbd-misc	1.15	2.4.0
kernel	4.14.336	6.1.112
kernel-libbpf		6.1.112
kernel-livepatch-r epo-s3		2023.6.20241031
kernel-srpm-macros		1,0
kernel-tools	4.14.336	6.1.112
keyutils	1.5.8	1.6.3
keyutils-libs	1.5.8	1.6.3
kmod	14	29
kmod-libs	14	29

Pacote	AMI do AL1	AMI do AL2023
kpartx	0.4.9	
kpatch-runtime		0.9.7
krb5-libs	1.15.1	1.21.3
lcms2	2.6	
less	436	608
libacl	2.2.49	2.3.1
libaio	0.3.109	0.3.111
libarchive		3.7.4
libargon2		20171227
libassuan	2.0.3	2.5.5
libattr	2.4.46	2.5.1
libbasicobjects		0.1.1
libblkid	2.23.2	2.37.4
libcap	2.16	2.48
libcap54	2.54	
libcap-ng	0.7.5	0.8.2
libcbor		0.7.0
libcgroup	0.40.rc1	
libcollection		0.7.0
libcom_err	1.43.5	1.46.5

Pacote	AMI do AL1	AMI do AL2023
libcomps		0.1.20
libconfig		1.7.2
libcurl	7.61.1	
libcurl-minimal		8.5.0
libdb		5.3.28
libdhash		0.5.0
libdnf		0.69.0
libeconf		0.4.0
libedit	2.11	3.1
libev		4.33
libevent	2.0.21	2.1.12
libfdisk		2.37.4
libffi	3.0.13	3.4.4
libfido2		1.10.0
libfontenc	1.0.5	
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.11	1.42

Pacote	AMI do AL1	AMI do AL2023
libgssglue	0.1	
libibverbs		48.0
libICE	1.0.6	
libicu	50.2	
libidn	1.18	
libidn2	2.3.0	2.3.2
libini_config		1.3.1
libjpeg-turbo	1.2.90	
libkcapi		1.4.0
libkcapi-hmacalc		1.4.0
libldb		2.6.2
libmaxminddb		1.5.2
libmetalink		0.1.3
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0
libmount	2.23.2	2.37.4
libnetfilter_connt rack	1.0.4	
libnfnetlink	1.0.1	
libnfsidmap	0.25	2.5.4
libnhttp2	1.33.0	1.59.0

Pacote	AMI do AL1	AMI do AL2023
libnih	1.0.1	
libnl	1.1.4	
libnl3		3.5.0
libpath_utils		0.2.1
libpcap		1.10.1
libpipeline	1.2.3	1.5.3
libpkgconf		1.8.0
libpng	1.2.49	
libpsl	0.6.2	0.21.1
libpwquality	1.2.3	1.4.4
libref_array		0.1.5
librepo		1.14.5
libreport-filesystem		2.15.2
libseccomp		2.5.3
libselinux	2.1.10	3.4
libselinux-utils	2.1.10	3.4
libsemanage	2.1.6	3.4
libsepol	2.1.7	3.4
libsigsegv		2.13
libSM	1.2.1	

Pacote	AMI do AL1	AMI do AL2023
libsmartcols	2.23.2	2.37.4
libsolv		0.7.22
libss	1.43.5	1.46.5
libssh2	1.4.2	
libsss_certmap		2.9.4
libsss_idmap		2.9.4
libsss_nss_idmap		2.9.4
libsss_sudo		2.9.4
libstdc++		11.4.1
libstdc++72	7.2.1	
libstoragegmt		1.9.4
libsysfs	2.1.0	
libtalloc		2.3.4
libtasn1	2.3	4.19.0
libtdb		1.4.7
libtevent		0.13.0
libtextstyle		0.21
libtirpc	0.2.4	1.3.3
libudev	173	
libunistring	0.9.3	0.9.10

Pacote	AMI do AL1	AMI do AL2023
libuser	0.60	0.63
libutempter	1.1.5	1.2.1
libuuid	2.23.2	2.37.4
libuv		1.47.0
libverto	0.2.5	0.3.2
libverto-libev		0.3.2
libX11	1.6.0	
libX11-common	1.6.0	
libXau	1.0.6	
libxcb	1.11	
libXcomposite	0.4.3	
libxcrypt		4.4.33
libXext	1.3.2	
libXfont	1.4.5	
libXi	1.7.2	
libxml2	2.9.1	2.10.4
libxml2-python27	2.9.1	
libXrender	0.9.8	
libxslt	1.1.28	
libXtst	1.2.2	

Pacote	AMI do AL1	AMI do AL2023
libyam1	0.1.6	0.2.5
libzstd		1.5.5
linux-firmware-whe nce		20210208
lm_sensors-libs		3.6.0
lmdb-libs		0.9.29
log4j-cve-2021-44228-hotpatch	1.3	
logrotate	3.7.8	3.20.1
lsof	4.82	4.94.0
lua	5.1.4	
lua-libs		5.4.4
lua-srpm-macros		1
lvm2	2.02.166	
lvm2-libs	2.02.166	
lz4-libs		1.9.4
mailcap	2.1.31	
make	3.82	
man-db	2.6.3	2.9.3
man-pages	4.10	5.10
mdadm	3.2.6	

Pacote	AMI do AL1	AMI do AL2023
microcode_ctl	2.1	2.1
mingetty	1.08	
mpfr		4.1.0
nano	2.5.3	5,8
nc	1.84	
ncurses	5.7	6.2
ncurses-base	5.7	6.2
ncurses-libs	5.7	6.2
nettle		3.8
net-tools	1.60	2.0
newt	0.52.11	0.52.21
newt-python27	0.52.11	
nfs-utils	1.3.0	2.5.4
npth		1.6
nspr	4.25.0	4.35.0
nss	3.53.1	3.90.0
nss-pem	1.0.3	
nss-softokn	3.53.1	3.90.0
nss-softokn-freebl	3.53.1	3.90.0
nss-sysinit	3.53.1	3.90.0

Pacote	AMI do AL1	AMI do AL2023
nss-tools	3.53.1	
nss-util	3.53.1	3.90.0
ntp	4.2.8p15	
ntpd	4.2.8p15	
ntsysv	1.3.49.3	1.15
numactl	2.0.7	
numactl-libs		2.0.14
ocaml-srpm-macros		6
oniguruma		6.9.7.1
openblas-srpm-macros		2
openldap	2.4.40	2.4.57
openssh	7.4p1	8.7p1
openssh-clients	7.4p1	8.7p1
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-libs		3.0.8
openssl-pkcs11		0.4.12
os-prober		1.77
p11-kit	0.18.5	0.24.1
p11-kit-trust	0.18.5	0.24.1

Pacote	AMI do AL1	AMI do AL2023
package-notes-srpm-macros		0.4
pam	1.1.8	1.5.1
pam_ccreds	10	
pam_krb5	2.3.11	
pam_passwdqc	1.0.5	
parted	2.1	3.4
passwd	0.79	0.80
pciutils	3.1.10	3.7.0
pciutils-libs	3.1.10	3.7.0
pcre	8.21	
pcre2		10.40
pcre2-syntax		10.40
perl	5.16.3	
perl-Carp	1.26	1.50
perl-Class-Struct		0.66
perl-constant	1.27	1.33
perl-Digest	1.17	
perl-Digest-HMAC	1.03	
perl-Digest-MD5	2.52	
perl-Digest-SHA	5.85	

Pacote	AMI do AL1	AMI do AL2023
perl-DynaLoader		1.47
perl-Encode	2.51	3.15
perl-Errno		1.30
perl-Exporter	5.68	5.74
perl-Fcntl		1.13
perl-File-Basename		2.85
perl-File-Path	2.09	2.18
perl-File-stat		1.09
perl-File-Temp	0.23.01	0.231.100
perl-Filter	1.49	
perl-Getopt-Long	2.40	2.52
perl-Getopt-Std		1.12
perl-HTTP-Tiny	0.033	0.078
perl-if		0.60.800
perl-interpreter		5.32.1
perl-IO		1.43
perl-IPC-Open3		1.21
perl-libs	5.16.3	5.32.1
perl-macros	5.16.3	
perl-MIME-Base64		3.16

Pacote	AMI do AL1	AMI do AL2023
perl-mro		1.23
perl-overload		1.31
perl-overloading		0.02
perl-parent	0.225	0.238
perl-PathTools	3.40	3.78
perl-Pod-Escapes	1.04	1.07
perl-podlators	2.5.1	4.14
perl-Pod-Perldoc	3.20	3.28.01
perl-Pod-Simple	3.28	3.42
perl-Pod-Usage	1.63	2.01
perl-POSIX		1.94
perl-Scalar-List-Utils	1.27	1.56
perl-SelectSaver		1.02
perl-Socket	2.010	2.032
perl-srpm-macros		1
perl-Storable	2.45	3.21
perl-subs		1.03
perl-Symbol		1.08
perl-Term-ANSIColor		5.01
perl-Term-Cap		1.17

Pacote	AMI do AL1	AMI do AL2023
perl-Text-ParseWords	3.29	3.30
perl-Text-Tabs+Wrap		2021.0726
perl-threads	1.87	
perl-threads-shared	1.43	
perl-Time-HiRes	1.9725	
perl-Time-Local	1.2300	1.300
perl-vars		1.05
pinentry	0.7.6	
pkgconf		1.8.0
pkgconfig	0.27.1	
pkgconf-m4		1.8.0
pkgconf-pkg-config		1.8.0
pm-utils	1.4.1	
policycoreutils	2.1.12	3.4
policycoreutils-python-utils		3.4
popt	1.13	1.18
procmail	3.22	
procps	3.2.8	
procps-ng		3.3.17
protobuf-c		1.4.1

Pacote	AMI do AL1	AMI do AL2023
psacct	6.3.2	6.6.4
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa		20240212
python27	2.7.18	
python27-babel	0.9.4	
python27-backports	1,0	
python27-backports-ssl_match_hostname	3.4.0.2	
python27-boto	2.48.0	
python27-botocore	1.17.31	
python27-chardet	2.0.1	
python27-colorama	0.4.1	
python27-configobj	4.7.2	
python27-crypto	2.6.1	
python27-daemon	1.5.2	
python27-dateutil	2.1	
python27-devel	2.7.18	
python27-docutils	0.11	
python27-ecdsa	0.11	

Pacote	AMI do AL1	AMI do AL2023
python27-futures	3.0.3	
python27-imaging	1.1.6	
python27-iniparse	0.3.1	
python27-jinja2	2.7.2	
python27-jmespath	0.9.2	
python27-jsonpatch	1.2	
python27-jsonpointer	1,0	
python27-kitchen	1.1.1	
python27-libs	2.7.18	
python27-lockfile	0.8	
python27-markupsafe	0.11	
python27-paramiko	1.15.1	
python27-pip	9.0.3	
python27-ply	3.4	
python27-pyasnl	0.1.7	
python27-pycurl	7.19.0	
python27-pygpme	0.3	
python27-pyliblzma	0.5.3	
python27-pystache	0.5.3	
python27-pyxdattr	0.5.0	

Pacote	AMI do AL1	AMI do AL2023
python27-PyYAML	3.10	
python27-requests	1.2.3	
python27-rsa	3.4.1	
python27-setuptools	36.2.7	
python27-simplejson	3.6.5	
python27-six	1.8.0	
python27-urlgrabber	3.10	
python27-urllib3	1.24.3	
python27-virtualenv	15.1.0	
python3		3.9.16
python3-attrs		20.3.0
python3-audit		3.0.6
python3-awscrt		0.19.19
python3-babel		2.9.1
python3-cffi		1.14.5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-daemon		2.3.0

Pacote	AMI do AL1	AMI do AL2023
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils		0.16
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-idna		(2.10)
python3-jinja2		2.11.3
python3-jmespath		0.10.0
python3-jsonpatch		1.21
python3-jsonpointer		2.0
python3-jjsonschema		3.2.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs		3.9.16
python3-libselinux		3.4
python3-libsemanage		3.4

Pacote	AMI do AL1	AMI do AL2023
python3-libstorage mgmt		1.9.4
python3-lockfile		0.12.2
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip-wheel		21.3.1
python3-ply		3.11
python3-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0.24
python3-pycparser		2.20
python3-pyrsistent		0.17.3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2.25.1
python3-rpm		4.16.1.3

Pacote	AMI do AL1	AMI do AL2023
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setuptools		4.4.1
python3-setuptools		59.6.0
python3-setuptools-wheel		59.6.0
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5
python-chevron		0.13.1
python-srpm-macros		3.9
quota	4.00	4.06
quota-nls	4.00	4.06
readline	6.2	8.1
rmt	0.4	
rng-tools	5	6.14
rootfiles	8.1	8.1
rpcbind	0.2.0	1.2.6
rpm	4.11.3	4.16.1.3

Pacote	AMI do AL1	AMI do AL2023
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit		4.16.1.3
rpm-python27	4.11.3	
rpm-sign-libs		4.16.1.3
rsync	3.0.6	3.2.6
rsyslog	5.8.10	
ruby	2.0	
ruby20	2.0.0.648	
ruby20-irb	2.0.0.648	
ruby20-libs	2.0.0.648	
rubygem20-bigdecimal	1.2.0	
rubygem20-json	1.8.3	
rubygem20-psych	2.0.0	
rubygem20-rdoc	4.2.2	
rubygems20	2.0.14.1	
rust-srpm-macros		21
sbsigntools		0.9.4
screen	4.0.3	4.8.0

Pacote	AMI do AL1	AMI do AL2023
sed	4.2.1	4.8
selinux-policy		38.1.45
selinux-policy-targeted		38.1.45
sendmail	8.14.4	
setserial	2.17	
setup	2.8.14	2.13.7
sgpio	1.2.0.10	
shadow-utils	4.1.4.2	4,9
shared-mime-info	1.1	
slang	2.2.1	2.3.2
sqlite	3.7.17	
sqlite-libs		3.40.0
sssd-client		2.9.4
sssd-common		2.9.4
sssd-kcm		2.9.4
sssd-nfs-idmap		2.9.4
strace		6.8
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1,0
sysfsutils	2.1.0	

Pacote	AMI do AL1	AMI do AL2023
sysstat		12.5.6
systemd		252.23
systemd-libs		252.23
systemd-networkd		252.23
systemd-pam		252.23
systemd-resolved		252.23
systemd-udev		252.23
system-release	2018.03	2023.6.20241031
systemtap-runtime		4.8
sysvinit	2.87	
tar	1.26	1.34
tbb		2020.3
tcp_wrappers	7.6	
tcp_wrappers-libs	7.6	
tcpdump		4.99.1
tcsh		6.24.07
time	1,7	1.9
tmpwatch	2.9.16	
traceroute	2.0.14	2.1.3
ttmkfdir	3.0.9	

Pacote	AMI do AL1	AMI do AL2023
tzdata	2023c	2024a
tzdata-java	2023c	
udev	173	
unzip	6.0	6.0
update-motd	1.0.1	2.2
upstart	0.6.5	
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2.23.2	2.37.4
util-linux-core		2.37.4
vim-common	9.0.2120	9.0.2153
vim-data	9.0.2120	9.0.2153
vim-enhanced	9.0.2120	9.0.2153
vim-filesystem	9.0.2120	9.0.2153
vim-minimal	9.0.2120	9.0.2153
wget	1.18	1.21.3
which	2.19	2.21
words	3.0	3.0
xfsdump		3.1.11
xfstools		5.18.0

Pacote	AMI do AL1	AMI do AL2023
xorg-x11-fonts-Type1	7.2	
xorg-x11-font-utils	7.2	
xxd	9.0.2120	9.0.2153
xxhash-libs		0.8.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	
yum-plugin-upgrade-helper	1.1.31	
yum-utils	1.1.31	
zip	3.0	3.0
zlib	1.2.8	1.2.11
zram-generator		1.1.2
zram-generator-defaults		1.1.2
zstd		1.5.5

Comparação entre pacotes instalados nas AMIs mínimas do Amazon Linux 1 (AL1) e do Amazon Linux 2023

Uma comparação dos RPMs presentes nas AMIs mínimas do AL1 e do AL2023.

Pacote	Mínima do AL1	Mínima do AL2023
acpid	2.0.19	
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.5.1
amazon-linux-repo-s3		2023.6.20241031
amazon-linux-sb-keys		2023.1
amd-ucode-firmware		20210208
audit	2.6.5	3.0.6
audit-libs	2.6.5	3.0.6
authconfig	6.2.8	
awscli-2		2.15.30
basesystem	10.0	11
bash	4.2.46	5.2.15
binutils	2.27	
bzip2	1.0.6	
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.62	2023.2.68

Pacote	Mínima do AL1	Mínima do AL2023
checkpolicy	2.1.10	3.4
chkconfig	1.3.49.3	
chrony		4.3
cloud-disk-utils	0.27	
cloud-init	0.7.6	22.2.2
cloud-init-cfg-ec2		22.2.2
cloud-utils-growpart		0.31
coreutils	8.22	8.32
coreutils-common		8.32
cpio	(2.10)	2.13
cracklib	2.8.16	2.9.6
cracklib-dicts	2.8.16	2.9.6
cronie	1.4.4	
cronie-anacron	1.4.4	
crontabs	1.10	
crypto-policies		20220428
cryptsetup-libs		2.6.1
curl	7.61.1	
curl-minimal		8.5.0
cyrus-sasl	2.1.23	

Pacote	Mínima do AL1	Mínima do AL2023
cyrus-sasl-lib	2.1.23	2.1.27
dash	0.5.5.1	
db4	4.7.25	
db4-utils	4.7.25	
dbus		1.12.28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.6.12	1.12.28
device-mapper		1.02.185
device-mapper-libs		1.02.185
dhclient	4.1.1	
dhcp-common	4.1.1	
diffutils	3.3	3.8
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0
dnf-plugin-support-info		1.2
dracut	004	055

Pacote	Mínima do AL1	Mínima do AL2023
dracut-config-ec2		3.0
dracut-config-generic		055
dracut-modules-growroot	0.20	
e2fsprogs	1.43.5	1.46.5
e2fsprogs-libs	1.43.5	1.46.5
ec2-utils	0.7	2.2.0
ed	1.1	
efi-filesystem		5
efivar		38
efivar-libs		38
elfutils-default-yama-scope		0.188
elfutils-libelf	0.168	0.188
elfutils-libs		0.188
ethtool	3.15	
expat	2.1.0	2.5.0
file	5.37	5.39
file-libs	5.37	5.39
filesystem	2.4.30	3.14

Pacote	Mínima do AL1	Mínima do AL2023
findutils	4.4.2	4.8.0
fipscheck	1.3.1	
fipscheck-lib	1.3.1	
fuse-libs	2.9.4	2.9.9
gawk	3.1.7	5.1.0
gdbm	1.8.0	
gdbm-libs		1.19
gdisk	0.8.10	1.0.8
generic-logos	17.0.0	
get_reference_source	1.2	
gettext		0.21
gettext-libs		0.21
glib2	2.36.3	2.74.7
glibc	2.17	2.34
glibc-all-langpacks		2.34
glibc-common	2.17	2.34
glibc-locale-source		2.34
gmp	6.0.0	6.2.1
gnupg2	2.0.28	
gnupg2-minimal		2.3.7

Pacote	Mínima do AL1	Mínima do AL2023
gnutls		3.8.0
gpgme	1.4.3	1.15.1
grep	2.20	3.8
groff	1.22.2	
groff-base	1.22.2	1.22.4
grub	0.97	
grub2-common		2.06
grub2-efi-x64-ec2		2.06
grub2-pc-modules		2.06
grub2-tools		2.06
grub2-tools-minimal		2.06
grubby	7.0.15	8.40
gzip	1.5	1.12
hesiod	3.1.0	
hmaccalc	0.9.12	
hostname		3.23
hwdata	0.233	0.384
info	5.1	
inih		49
initscripts	9.03.58	10.09

Pacote	Mínima do AL1	Mínima do AL2023
iproute	4.4.0	6.10.0
iptables	1.4.21	
iputils	20121221	20210202
irqbalance		1.9.0
jansson		2.14
jitterentropy		3.4.1
jq		1.7.1
json-c		0.14
kbd	1.15	2.4.0
kbd-misc	1.15	2.4.0
kernel	4.14.336	6.1.112
kernel-libbpf		6.1.112
kernel-livepatch-r epo-s3		2023.6.20241031
keyutils-libs	1.5.8	1.6.3
kmod	14	29
kmod-libs	14	29
krb5-libs	1.15.1	1.21.3
less	436	608
libacl	2.2.49	2.3.1
libarchive		3.7.4

Pacote	Mínima do AL1	Mínima do AL2023
libargon2		20171227
libassuan	2.0.3	2.5.5
libattr	2.4.46	2.5.1
libblkid	2.23.2	2.37.4
libcap	2.16	2.48
libcap54	2.54	
libcap-ng	0.7.5	0.8.2
libcbor		0.7.0
libcgroup	0.40.rc1	
libcom_err	1.43.5	1.46.5
libcomps		0.1.20
libcurl	7.61.1	
libcurl-minimal		8.5.0
libdb		5.3.28
libdnf		0.69.0
libeconf		0.4.0
libedit	2.11	3.1
libfdisk		2.37.4
libffi	3.0.13	3.4.4
libfido2		1.10.0

Pacote	Mínima do AL1	Mínima do AL2023
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.11	1.42
libicu	50.2	
libidn	1.18	
libidn2	2.3.0	2.3.2
libkcapi		1.4.0
libkcapi-hmacalc		1.4.0
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0
libmount	2.23.2	2.37.4
libnetfilter_connt rack	1.0.4	
libnfnetlink	1.0.1	
libnghttp2	1.33.0	1.59.0
libnih	1.0.1	
libpipeline		1.5.3
libpsl	0.6.2	0.21.1
libpwquality	1.2.3	1.4.4

Pacote	Mínima do AL1	Mínima do AL2023
librepo		1.14.5
libreport-filessystem		2.15.2
libseccomp		2.5.3
libselinux	2.1.10	3.4
libselinux-utils	2.1.10	3.4
libsemanage	2.1.6	3.4
libsepol	2.1.7	3.4
libsigsegv		2.13
libsmartcols	2.23.2	2.37.4
libsolv		0.7.22
libss	1.43.5	1.46.5
libssh2	1.4.2	
libstdc++		11.4.1
libstdc++72	7.2.1	
libsysfs	2.1.0	
libtasn1	2.3	4.19.0
libtextstyle		0.21
libudev	173	
libunistring	0.9.3	0.9.10
libuser	0.60	0.63

Pacote	Mínima do AL1	Mínima do AL2023
libutempter	1.1.5	1.2.1
libuuid	2.23.2	2.37.4
libverto	0.2.5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml	0.1.6	0.2.5
libzstd		1.5.5
linux-firmware-whe nce		20210208
logrotate	3.7.8	3.20.1
lua	5.1.4	
lua-libs		5.4.4
lz4-libs		1.9.4
make	3.82	
man-db		2.9.3
microcode_ctl	2.1	2.1
mingetty	1.08	
mpfr		4.1.0
ncurses	5.7	6.2
ncurses-base	5.7	6.2
ncurses-libs	5.7	6.2

Pacote	Mínima do AL1	Mínima do AL2023
nettle		3.8
net-tools	1.60	2.0
newt	0.52.11	
newt-python27	0.52.11	
npth		1.6
nspr	4.25.0	
nss	3.53.1	
nss-pem	1.0.3	
nss-softokn	3.53.1	
nss-softokn-freebl	3.53.1	
nss-sysinit	3.53.1	
nss-tools	3.53.1	
nss-util	3.53.1	
ntp	4.2.8p15	
ntpdate	4.2.8p15	
numactl-libs		2.0.14
oniguruma		6.9.7.1
openldap	2.4.40	2.4.57
openssh	7.4p1	8.7p1
openssh-clients		8.7p1

Pacote	Mínima do AL1	Mínima do AL2023
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-lib		3.0.8
openssl-pkcs11		0.4.12
os-prober		1.77
p11-kit	0.18.5	0.24.1
p11-kit-trust	0.18.5	0.24.1
pam	1.1.8	1.5.1
passwd	0.79	0.80
pciutils	3.1.10	3.7.0
pciutils-lib	3.1.10	3.7.0
pcre	8.21	
pcre2		10.40
pcre2-syntax		10.40
pinentry	0.7.6	
pkgconfig	0.27.1	
policycoreutils	2.1.12	3.4
popt	1.13	1.18
procmail	3.22	
procps	3.2.8	

Pacote	Mínima do AL1	Mínima do AL2023
procps-ng		3.3.17
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa		20240212
python27	2.7.18	
python27-babel	0.9.4	
python27-backports	1,0	
python27-backports-ssl_match_hostname	3.4.0.2	
python27-chardet	2.0.1	
python27-configobj	4.7.2	
python27-iniparse	0.3.1	
python27-jinja2	2.7.2	
python27-jsonpatch	1.2	
python27-jsonpointer	1,0	
python27-libs	2.7.18	
python27-markupsafe	0.11	
python27-pycurl	7.19.0	
python27-pygpme	0.3	
python27-pyliblzma	0.5.3	

Pacote	Mínima do AL1	Mínima do AL2023
python27-pyxattr	0.5.0	
python27-PyYAML	3.10	
python27-requests	1.2.3	
python27-setuptools	36.2.7	
python27-six	1.8.0	
python27-urlgrabber	3.10	
python27-urllib3	1.24.3	
python3		3.9.16
python3-attrs		20.3.0
python3-audit		3.0.6
python3-awscrt		0.19.19
python3-babel		2.9.1
python3-cffi		1.14.5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0

Pacote	Mínima do AL1	Mínima do AL2023
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils		0.16
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-idna		(2.10)
python3-jinja2		2.11.3
python3-jmespath		0.10.0
python3-jsonpatch		1.21
python3-jsonpointer		2.0
python3-jsonschema		3.2.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs		3.9.16
python3-libselenium		3.4
python3-libsemanage		3.4
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip-wheel		21.3.1

Pacote	Mínima do AL1	Mínima do AL2023
python3-ply		3.11
python3-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0.24
python3-pycparser		2.20
python3-pyrsistent		0.17.3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2.25.1
python3-rpm		4.16.1.3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml- clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59.6.0
python3-setuptools- wheel		59.6.0
python3-six		1.15.0

Pacote	Mínima do AL1	Mínima do AL2023
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5
readline	6.2	8.1
rng-tools		6.14
rootfiles	8.1	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit		4.16.1.3
rpm-python27	4.11.3	
rpm-sign-libs		4.16.1.3
rsyslog	5.8.10	
sbsigntools		0.9.4
sed	4.2.1	4.8
selinux-policy		38.1.45
selinux-policy-targeted		38.1.45
sendmail	8.14.4	

Pacote	Mínima do AL1	Mínima do AL2023
setserial	2.17	
setup	2.8.14	2.13.7
shadow-utils	4.1.4.2	4,9
shared-mime-info	1.1	
slang	2.2.1	
sqlite	3.7.17	
sqlite-libs		3.40.0
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1,0
sysfsutils	2.1.0	
systemd		252.23
systemd-libs		252.23
systemd-networkd		252.23
systemd-pam		252.23
systemd-resolved		252.23
systemd-udev		252.23
system-release	2018.03	2023.6.20241031
sysvinit	2.87	
tar	1.26	1.34
tcp_wrappers-libs	7.6	

Pacote	Mínima do AL1	Mínima do AL2023
tzdata	2023c	2024a
udev	173	
update-motd	1.0.1	2.2
upstart	0.6.5	
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2.23.2	2.37.4
util-linux-core		2.37.4
vim-data	9.0.2120	9.0.2153
vim-minimal	9.0.2120	9.0.2153
which	2.19	2.21
xfspgrog		5.18.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-priorities	1.1.31	
yum-plugin-upgrade-helper	1.1.31	
zlib	1.2.8	1.2.11

Pacote	Mínima do AL1	Mínima do AL2023
zram-generator		1.1.2
zram-generator-def aults		1.1.2
zstd		1.5.5

Comparar pacotes instalados nas imagens de contêiner base do Amazon Linux 1 (AL1) e do Amazon Linux 2023

Uma comparação dos RPMs presentes nas imagens básicas de contêiner do AL1 e do AL2023.

Pacote	Contêiner do AL1	Contêiner do AL2023
alternatives		1.15
amazon-linux-repo- cdn		2023.6.20241031
audit-libs		3.0.6
basesystem	10.0	11
bash	4.2.46	5.2.15
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.62	2023.2.68
chkconfig	1.3.49.3	
coreutils	8.22	
coreutils-single		8.32
crypto-policies		20220428

Pacote	Contêiner do AL1	Contêiner do AL2023
curl	7.61.1	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.23	
db4	4.7.25	
db4-utils	4.7.25	
dnf		4.14.0
dnf-data		4.14.0
elfutils-default-y ama-scope		0.188
elfutils-libelf	0.168	0.188
elfutils-libs		0.188
expat	2.1.0	2.5.0
file-libs	5.37	5.39
filesystem	2.4.30	3.14
gawk	3.1.7	5.1.0
gdbm	1.8.0	
gdbm-libs		1.19
glib2	2.36.3	2.74.7
glibc	2.17	2.34
glibc-common	2.17	2.34

Pacote	Contêiner do AL1	Contêiner do AL2023
glibc-minimal-lang pack		2.34
gmp	6.0.0	6.2.1
gnupg2	2.0.28	
gnupg2-minimal		2.3.7
gpgme	1.4.3	1.15.1
grep	2.20	3.8
gzip	1.5	
info	5.1	
json-c		0.14
keyutils-libs	1.5.8	1.6.3
krb5-libs	1.15.1	1.21.3
libacl	2.2.49	2.3.1
libarchive		3.7.4
libassuan	2.0.3	2.5.5
libattr	2.4.46	2.5.1
libblkid		2.37.4
libcap	2.16	2.48
libcap-ng		0.8.2
libcom_err	1.43.5	1.46.5
libcomps		0.1.20

Pacote	Contêiner do AL1	Contêiner do AL2023
<u>libcurl</u>	7.61.1	
<u>libcurl-minimal</u>		8.5.0
libdnf		0.69.0
libffi	3.0.13	3.4.4
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.11	1.42
libicu	50.2	
libidn2	2.3.0	2.3.2
libmodulemd		2.13.0
libmount		2.37.4
libnghttp2	1.33.0	1.59.0
libpsl	0.6.2	0.21.1
librepo		1.14.5
libreport-filessystem		2.15.2
libselinux	2.1.10	3.4
libsepol	2.1.7	3.4
libsigsegv		2.13

Pacote	Contêiner do AL1	Contêiner do AL2023
libsmartcols		2.37.4
libsolv		0.7.22
libssh2	1.4.2	
libstdc++		11.4.1
libstdc++72	7.2.1	
libtasn1	2.3	4.19.0
libunistring	0.9.3	0.9.10
libuuid		2.37.4
libverto	0.2.5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libxml2-python27	2.9.1	
libyaml		0.2.5
libzstd		1.5.5
lua	5.1.4	
lua-libs		5.4.4
lz4-libs		1.9.4
make	3.82	
mpfr		4.1.0
ncurses	5.7	

Pacote	Contêiner do AL1	Contêiner do AL2023
ncurses-base	5.7	6.2
ncurses-libs	5.7	6.2
npth		1.6
nspr	4.25.0	
nss	3.53.1	
nss-pem	1.0.3	
nss-softokn	3.53.1	
nss-softokn-freebl	3.53.1	
nss-sysinit	3.53.1	
nss-tools	3.53.1	
nss-util	3.53.1	
openldap	2.4.40	
openssl	1.0.2k	
openssl-libs		3.0.8
p11-kit	0.18.5	0.24.1
p11-kit-trust	0.18.5	0.24.1
pcre	8.21	
pcre2		10.40
pcre2-syntax		10.40
pinentry	0.7.6	

Pacote	Contêiner do AL1	Contêiner do AL2023
pkgconfig	0.27.1	
popt	1.13	1.18
pth	2.0.7	
publicsuffix-list-dafsa		20240212
python27	2.7.18	
python27-chardet	2.0.1	
python27-iniparse	0.3.1	
python27-kitchen	1.1.1	
python27-libs	2.7.18	
python27-pycurl	7.19.0	
python27-pygpme	0.3	
python27-pyliblzma	0.5.3	
python27-pyattr	0.5.0	
python27-urlgrabber	3.10	
python3		3.9.16
python3-dnf		4.14.0
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0

Pacote	Contêiner do AL1	Contêiner do AL2023
python3-libs		3.9.16
python3-pip-wheel		21.3.1
python3-rpm		4.16.1.3
python3-setuptools-wheel		59.6.0
readline	6.2	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-python27	4.11.3	
rpm-sign-libs		4.16.1.3
sed	4.2.1	4.8
setup	2.8.14	2.13.7
shared-mime-info	1.1	
sqlite	3.7.17	
sqlite-libs		3.40.0
sysctl-defaults	1,0	
system-release	2018.03	2023.6.20241031
tar	1.26	
tzdata	2023c	2024a
xz-libs	5.2.2	5.2.5

Pacote	Contêiner do AL1	Contêiner do AL2023
yum	3.4.3	4.14.0
yum-metadata-parser	1.1.4	
yum-plugin-ovl	1.1.31	
yum-plugin-priorities	1.1.31	
yum-utils	1.1.31	
zlib	1.2.8	1.2.11

Requisitos do sistema do AL2023

Esta seção descreve os requisitos do sistema para usar o AL2023.

Tópicos

- [Requisitos de CPU para executar o AL2023](#)
- [Requisitos de memória \(RAM\) para executar o AL2023](#)

Requisitos de CPU para executar o AL2023

Para executar qualquer código do AL2023, o processador usado precisa atender a determinados requisitos mínimos. Tentativas de executar o AL2023 em CPUs que não atendam a esses requisitos podem resultar em erros de instrução ilegal logo no início da execução do código.

Os requisitos mínimos se aplicam a [AL2023 no Amazon EC2](#), [AL2023 em contêineres](#) e [AL2023 fora do Amazon EC2](#).

Requisitos de CPU ARM para AL2023

Todos os binários aarch64 (ARM) do AL2023 são compilados para 64 bits. Não há binários ARM de 32 bits disponíveis, portanto, é necessária uma CPU ARM de 64 bits.

Note

Para instâncias baseadas em ARM, o AL2023 é compatível apenas com tipos de instância que usam processadores Graviton2 ou posteriores. O AL2023 não oferece suporte a instâncias A1.

O AL2023 requer um processador compatível com ARMv8.2 com a extensão de criptografia (ARMv8.2+crypto). Todos os pacotes do AL2023 para aarch64 são compilados com o sinalizador do compilador `-march=armv8.2-a+crypto`. Embora tentemos imprimir mensagens de erro elegantes quando se tenta executar o código do AL2023 em processadores ARM mais antigos, é possível que a primeira mensagem de erro seja um erro de instrução ilegal.

 Note

Devido aos requisitos básicos de CPU `aarch64` do AL2023, todos os sistemas Raspberry Pi anteriores ao Raspberry Pi 5 não atendem aos requisitos mínimos de CPU.

Requisitos da CPU x86-64 para AL2023

Todos os binários x86-64 do AL2023 são compilados para a revisão x86-64v2 da arquitetura x86-64, enviando `-march=x86-64-v2` para o compilador.

A revisão x86-64v2 da arquitetura adiciona os seguintes recursos de CPU à arquitetura x86-64 de base:

- `CMPXCHG16B`
- `LAHF-SAHF`
- `POPCNT`
- `SSE3`
- `SSE4_1`
- `SSE4_2`
- `SSSE3`

Isso corresponde aproximadamente aos processadores x86-64 lançados em 2009 ou posteriormente. Exemplos incluem Intel Nehalem, AMD Jaguar e Atom Silvermont, junto com as microarquiteturas Eden C e VIA Nano.

No Amazon EC2, todos os tipos de instâncias x86-64 oferecem suporte x86-64v2, incluindo famílias de instâncias M1, C1 e M2.

Nenhum binário x86 (i686) de 32 bits do AL2023 foi criado. Embora o AL2023 mantenha o suporte para executar binários de espaço de usuário de 32 bits, essa funcionalidade está obsoleta e pode ser removida em uma futura versão principal do Amazon Linux. Para obter mais informações, consulte [Pacotes x86 \(i686\) de 32 bits](#).

Requisitos de memória (RAM) para executar o AL2023

A família de tipos de instância `.nano` do Amazon EC2 (`t2.nano`, `t3.nano`, `t3a.nano` e `t4g.nano`) tem 512 MB de RAM, e esse é o requisito mínimo para o AL2023.

Note

Embora 512 MB seja o requisito mínimo, esses tipos de instância têm restrição de memória e a funcionalidade e o desempenho podem ser limitados.

As imagens do AL2023 não foram testadas em sistemas com menos de 512 MB de RAM. A execução de imagens de contêiner baseadas no AL2023 em menos de 512 MB de RAM dependerá da workload em contêineres.

Algumas workloads, como `dnf upgrade` entre alguns lançamentos do AL2023, podem exigir mais de 512 MB de RAM. Por esse motivo, o lançamento [AL2023.3](#) introduziu a habilitação de `zram` por padrão para instâncias com menos de 800 MB de RAM. Para workloads em contêineres, isso significa que algumas workloads podem funcionar bem em instâncias do AL2023 com essa quantidade de memória, mas falhar quando executadas em um contêiner restrito a essa quantidade de uso de memória.

Para tipos de exemplo com menos de 800 MB de RAM, o AL2023 (a partir do [AL2023.3](#) ou mais recente) habilitará a troca baseada em `zram` por padrão. Exemplos de tipos de instância do Amazon EC2 com menos de 800 MB de memória incluem `t4g.nano`, `t3a.nano`, `t3.nano`, `t2.nano` e `t1.micro`. Isso significa menos cenários de falta de memória para esses tipos de instância, porque o AL2023 compactará e descompactará páginas de memória sob demanda. Isso permite workloads que, de outra forma, exigiriam um tipo de instância com mais memória, às custas do uso da CPU necessário para fazer a compactação.

O kernel Linux da Amazon

O Amazon Linux 2023 (AL2023) lança um novo kernel Long-Term Support (LTS) no primeiro trimestre de cada ano, com base no kernel LTS anual da comunidade Linux upstream. Cada novo kernel LTS traz as mais recentes correções de segurança, melhorias de desempenho, suporte de hardware e recursos do kernel Linux upstream.

Ciclo de vida do kernel

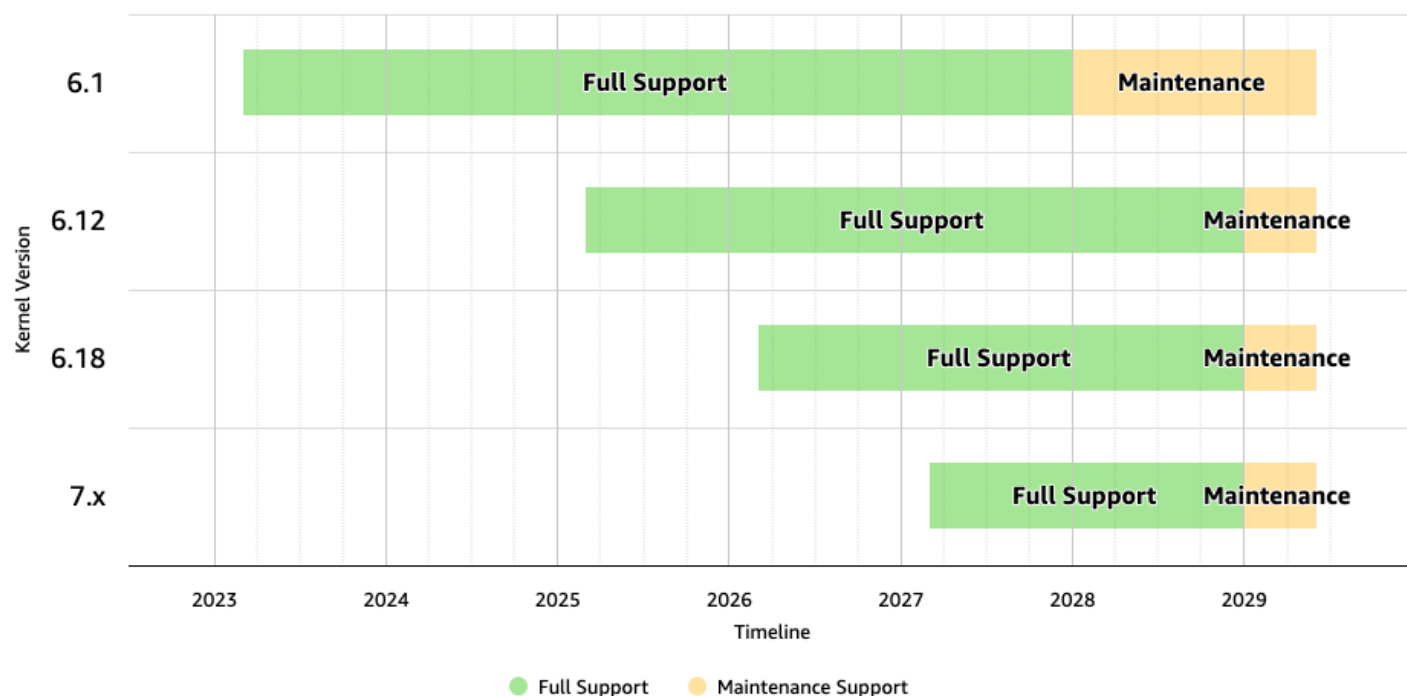
Cada kernel AL2023 LTS é suportado por quatro anos em duas fases:

1. Suporte total (anos 1 a 2) — O kernel recebe correções para todas as severidades do CVE por meio de rebases regulares no kernel LTS upstream. Essa fase se alinha com a janela de suporte LTS da comunidade Linux upstream. Se o upstream estender a janela de suporte do LTS, o Amazon Linux seguirá o exemplo.
2. Suporte de manutenção (anos 3 a 4) — Após o término do período de suporte LTS upstream, a equipe do Amazon Linux continua fazendo o backport principalmente de correções para vulnerabilidades críticas e importantes CVEs (pontuação CVSS 7.0 ou superior), bem como para vulnerabilidades exploradas conhecidas. A severidade baixa e média CVEs não é revertida durante essa fase.

Depois de quatro anos, o kernel chega ao fim da vida útil e não recebe mais atualizações de segurança. Os kernels mais antigos permanecem disponíveis por meio de repositórios e você pode continuar usando-os. Você não deve esperar mais patches ou correções. Recomendamos a atualização para um kernel compatível antes dessa data. O kernel específico não AMIs será mais atualizado após a data de fim do suporte de um kernel.

O gráfico a seguir mostra o cronograma de suporte para os kernels LTS atuais do Amazon Linux:

Amazon Linux 2023 Kernel Release Timeline



Atualizações do kernel

A partir de junho de 2026, AL2023 atualizará o kernel padrão anualmente. O [al2023-ami-kernel-default](#) conjunto de AMIs será atualizado para o kernel LTS mais recente, para que as instâncias recém-lançadas criem a nova versão do kernel.

As instâncias em execução não são atualizadas automaticamente para um novo kernel. Para atualizar o kernel em uma instância existente, você deve instalar explicitamente o novo pacote do kernel e reinicializar. Para obter detalhes, consulte [Atualizando o kernel Linux em AL2023](#).

É altamente recomendável adotar novos kernels imediatamente para se beneficiar das mais recentes melhorias de segurança e desempenho. Os kernels mais antigos recebem menos e mais lentas correções de segurança ao longo do tempo. Incorpore as atualizações do kernel em seus pipelines de teste e implantação existentes para validar a compatibilidade antes de lançá-las para produção.

Sobreposição de versões do kernel entre distribuições

Para simplificar as migrações entre as distribuições Amazon Linux, pelo menos uma versão do kernel estará disponível na distribuição atual e na próxima distribuição Amazon Linux. Isso permite que

você primeiro atualize para um novo kernel em sua distribuição existente, valide suas cargas de trabalho e depois migre para a nova distribuição com a certeza de que a mesma versão do kernel está disponível lá.

Manipulação de CVE

AL2023 aborda o kernel da CVEs seguinte forma:

- Vulnerabilidades críticas e importantes CVEs (CVSS 7.0 e superior) e conhecidas exploradas são transferidas para todos os kernels suportados.
- Baixo e médio CVEs (CVSS abaixo de 7.0) são abordados por meio de rebases regulares de LTS upstream durante a fase de suporte completo. Durante a fase de suporte de manutenção, eles não CVEs são retroportados. Se um CVE baixo ou médio não for resolvido por um rebase LTS upstream dentro de 60 dias, ele será marcado como “Nenhuma correção planejada” no Centro de Segurança Linux da [Amazon](#).

Executar o kernel mais recente disponível é a melhor maneira de obter atualizações recentes de segurança, desempenho e funcionalidade.

O que você deve fazer

1. Implemente a Integração Contínua (CI) para seus aplicativos — Antes de fazer qualquer alteração na configuração de produção, verifique se ela foi validada adequadamente em um estágio de teste. Inclua atualizações do kernel em sua validação.
2. Mantenha-se atualizado com o kernel mais recente — Adote cada novo kernel LTS anual assim que ele estiver disponível. Os kernels mais novos recebem correções de segurança mais rapidamente. Consuma a [al2023-ami-kernel-default](#) série de AMIs para estar automaticamente na versão do kernel recomendada pela equipe do Amazon Linux.
3. Automatize as atualizações — Use ferramentas como o [AWS Systems Manager](#) para gerenciar as atualizações do kernel em toda a sua frota.
4. Plano para reinicializações — Cada atualização do kernel requer uma reinicialização. Inclua janelas de reinicialização em seus cronogramas de manutenção.

AL2023 Desktop gráfico

O Amazon Linux 2023 fornece uma interface gráfica opcional, leve e otimizada para nuvem, baseada no GNOME a partir do lançamento 2023.7. Esse ambiente de área de trabalho moderno oferece recursos aprimorados de produtividade com ferramentas integradas, como o Firefox, para navegação segura, enquanto mantém o suporte ao Amazon DCV e VNC para acesso remoto.

Note

AL2023 acompanha de perto a versão upstream do Firefox Extended Support Release (ESR) e é atualizado para o próximo ESR assim que disponível. Para obter mais informações, consulte o [calendário de lançamentos do Firefox ESR](#) e as [notas de lançamento do Firefox](#).

Tópicos relacionados

Para saber mais sobre como instalar o ambiente gráfico de área de trabalho, consulte a seguinte documentação:

- [Tutorial: Instale o ambiente de trabalho GNOME em AL2023](#)

Pacotes suplementares para Amazon Linux

Esta seção apresenta pacotes suplementares para Amazon Linux (SPAL) e descreve seus benefícios e limitações, além de diretrizes para relatar problemas relacionados a pacotes.

Tópicos

- [O que são pacotes suplementares para Amazon Linux \(ou SPAL\)?](#)
- [Benefícios](#)
- [Support of SPAL packages](#)
- [Relatando problemas relacionados ao pacote](#)
- [Tópicos relacionados](#)
- [Tutorial: Configurar o repositório SPAL em 023 AL2](#)

O que são pacotes suplementares para Amazon Linux (ou SPAL)?

Pacotes suplementares para Amazon Linux (SPAL) é um repositório de pacotes dedicado que fornece acesso a milhares de pacotes adicionais derivados de [pacotes extras para Enterprise Linux 9](#) (). EPEL9 Esses pacotes complementam o software existente disponível no núcleo do Amazon Linux 2023.

O SPAL simplifica a implantação de software fornecendo pacotes pré-criados que são compatíveis com AL2 023, eliminando a necessidade de os clientes criarem pacotes a partir do próprio código-fonte. Isso economiza tempo e esforço no processo de instalação do software.

Note

O SPAL está disponível em todas as regiões AWS comerciais, incluindo as regiões AWS GovCloud (EUA) e a China, para AL2 023 instâncias com uma versão de lançamento igual 2023.9.20251117 ou posterior.

Benefícios

O SPAL oferece várias vantagens importantes para os usuários do Amazon Linux 2023:

- Expansão dos casos de uso do AL2 023 — O acesso a pacotes adicionais populares além do repositório principal do AL2 023, como, `GDAL` or `pandocdrbd-utils`, permite que os clientes atendam a várias necessidades comerciais e de desenvolvimento.
- Simplificando o gerenciamento de pacotes AL2 023 — Ao fornecer pacotes pré-construídos para AL2 023, o processo de criação de pacotes adicionais a partir do código-fonte é eliminado, economizando tempo e reduzindo o risco de erros de compilação.
- Simplificação AL2 para a migração AL2 023 — O repositório SPAL permite a migração perfeita de cargas de trabalho de AL2 0 AL2 para 023, incluindo cargas de trabalho que dependem de pacotes. EPEL7

Support of SPAL packages

Os pacotes SPAL não são cobertos pelo mesmo nível de suporte dos pacotes principais AL2 023, que recebem suporte durante toda a vida útil do Amazon Linux 2023.

Important

Antes de usar o SPAL, os clientes devem avaliar cuidadosamente as seguintes considerações:

- Os pacotes SPAL NÃO são cobertos pelos AWS Support Plans.
- Os pacotes SPAL são fornecidos “no estado em que se encontram” a partir do upstream. EPEL9
- Os pacotes SPAL NÃO receberão rastreamento de segurança AWS CVE.
- Os pacotes SPAL recebem patches de segurança e correções de bugs exclusivamente do upstream, EPEL9 quando disponíveis.

Relatando problemas relacionados ao pacote

Se você encontrar um problema com um pacote SPAL, recomendamos primeiro verificar se o mesmo problema ocorre com o pacote correspondente no repositório upstream EPEL9 . Para isso, consulte a [lista de problemas no repositório](#) EPEL upstream.

Se o problema não estiver presente EPEL9, crie um problema no [GitHubrepositório Amazon Linux 2023](#), pois isso indica que o problema é específico da compilação ou configuração do pacote SPAL.

Essa abordagem garante que os problemas sejam resolvidos pelos mantenedores apropriados e contribua para a qualidade geral dos pacotes SPAL e EPEL upstream.

 Note

Os problemas relatados serão tratados com base no melhor esforço.

Tópicos relacionados

Para obter informações sobre como configurar o SPAL em seu sistema, consulte a seguinte página de documentação:

- [Tutorial: Configurar o repositório SPAL em 023 AL2](#)

Tutorial: Configurar o repositório SPAL em 023 AL2

Os Pacotes Suplementares para Amazon Linux (SPAL) são um repositório de pacotes adicional para o AL2 023 que fornece aos clientes acesso a milhares de pacotes de código aberto.

O tutorial a seguir ajuda você a configurar o repositório SPAL na sua instância AL2 023. Ao instalar o repositório, você terá acesso a todos os pacotes RPM disponíveis no SPAL. Depois de instalado, você pode usar seu gerenciador de pacotes para instalar e usar esses pacotes em seu sistema.

Conteúdos

- [Pré-requisitos](#)
- [Verificando os pré-requisitos](#)
- [Instalando o SPAL em seu sistema](#)
- [Instalando pacotes SPAL](#)
- [Desinstalando o repositório SPAL do seu sistema](#)
- [Tópicos relacionados](#)

Pré-requisitos

Este tutorial pressupõe que você já tenha executado uma instância usando a versão de lançamento AL2023.9.20251117 ou posterior. Para saber mais, consulte as páginas [AL2023 no Amazon EC2](#) e [Atualizando AL2023](#).

Verificando os pré-requisitos

- Para verificar se sua instância atende aos pré-requisitos, você pode verificar a versão `system-release` instalada em seu sistema.

Para verificar a versão do pacote, você pode usar o comando a seguir.

```
[ec2-user ~]$ rpm -qi system-release
```

O comando exibirá informações sobre o pacote, incluindo a versão principal.

```
Name       : system-release
Version    : 2023.9.20251117
...
```

Note

Certifique-se de ter a versão mais recente do `system-release` instalado. Você pode executar `sudo dnf upgrade` para atualizar para a versão mais recente.

Instalando o SPAL em seu sistema

1. Instale o `spal-release` pacote no seu sistema. Isso adiciona o arquivo `.repo` de configuração e as chaves GPG ao seu sistema.

```
[ec2-user ~]$ sudo dnf install spal-release
```

Note

Durante a instalação, a declaração de suporte será exibida. A declaração explica o escopo de suporte e as limitações do SPAL. Reserve um tempo para revisar essas informações com cuidado.

2. Verifique se a configuração do repositório SPAL foi adicionada com sucesso ao seu sistema.

```
[ec2-user ~]$ cat /etc/yum.repos.d/amazonlinux-spal.repo
```

Você deve ver os dois repositórios configurados em seu sistema: `amazonlinux-spal` e `amazonlinux-spal-source`

Você também pode verificar a lista de repositórios configurados executando `dnf repolist`.

```
[ec2-user ~]$ dnf repolist --all
```

Note

O `--all` sinalizador é obrigatório para ver os repositórios habilitados e desativados.

Ambos os repositórios SPAL devem estar disponíveis. Observe que o repositório Amazon Linux 2023 SPAL - repositório de pacotes de origem está desativado por padrão.

repo id	repo name
status	
amazonlinux-spal	Amazon Linux 2023 SPAL repository
enabled	
amazonlinux-spal-source	Amazon Linux 2023 SPAL repository - Source packages
disabled	

3. (Opcional) Ative o repositório de origem.

Note

Os repositórios de origem RPM (SRPM) geralmente são desativados por padrão porque são usados principalmente por desenvolvedores para criar pacotes, não por usuários

finais para instalação de software. O DNF ativa automaticamente os repositórios de origem quando você usa comandos que exigem pacotes de origem, como. `dnf download --source package`

Você não precisa habilitar manualmente o repositório de origem para operações únicas de pacotes de origem. Siga esta etapa somente se quiser reconstruir a SRPMs partir do SPAL em seu sistema.

Para habilitar permanentemente o repositório SPAL do Amazon Linux 2023 - repositório de pacotes de origem em seu sistema, execute o seguinte comando:

```
[ec2-user ~]$ sudo dnf config-manager --enable amazonlinux-spal-source
```

Instalando pacotes SPAL

- Instale pacotes SPAL em seu sistema executando o `dnf install` comando.

```
[ec2-user ~]$ sudo dnf install package
```

Note

Você pode usar `dnf list` para ver uma lista completa de pacotes SPAL.

```
[ec2-user ~]$ dnf list --repo=amazonlinux-spal
```

Note

O SPAL é um repositório versionado. Certifique-se de ter a versão mais recente do `system-release` instalado para ver a lista mais recente de pacotes.

Para obter mais informações sobre atualizações determinísticas, você pode verificar [Atualizações determinísticas por meio de repositórios versionados em AL2023](#)

Desinstalando o repositório SPAL do seu sistema

1. Remova a configuração do repositório SPAL usando o `dnf remove` comando.

```
[ec2-user ~]$ sudo dnf remove spal-release
```

2. Verifique se o repositório foi removido executando o `dnf repolist` comando.

```
[ec2-user ~]$ dnf repolist
```

Important

A remoção da configuração do repositório SPAL do seu sistema não remove nenhum pacote SPAL instalado no sistema.

Tópicos relacionados

Para obter mais informações sobre os pacotes suplementares para o repositório Amazon Linux, consulte a seguinte documentação:

- [Pacotes suplementares para Amazon Linux](#)

Executar aplicações no AL2023

Esta seção aborda métodos para executar aplicações no Amazon Linux 2023 (AL2023), incluindo o gerenciamento de quando elas são iniciadas (e reiniciadas) e o controle do uso de recursos.

Tópicos

- [Limitar o uso de recursos do processo no AL2023 usando systemd](#)
- [Limitar o uso de recursos do processo no AL2023 usando cgroups](#)

Limitar o uso de recursos do processo no AL2023 usando systemd

No Amazon Linux 2023 (AL2023), recomendamos o uso de `systemd` para controlar quais recursos podem ser usados por processos ou grupos de processos. O uso de `systemd` é um substituto poderoso e fácil de usar para manipular `cgroups` manualmente ou usar utilitários como [cpulimit](#), que antes só estavam disponíveis para o Amazon Linux no repositório [EPEL](#) de terceiros.

Para obter informações abrangentes, consulte a documentação do `systemd` upstream para [systemd.resource-control](#) ou a página `man` para `systemd.resource-control` em uma instância do AL2023.

Os exemplos abaixo usarão o teste de estresse da CPU `stress-ng` (do pacote `stress-ng`) para simular uma aplicação com uso intenso da CPU e `memcached` para simular uma aplicação com uso intenso da memória.

Os exemplos abaixo abrangem a aplicação de um limite de CPU em um comando único e um limite de memória em um serviço. A maioria das restrições de recursos que `systemd` oferece podem ser usadas em qualquer lugar que `systemd` execute um processo, e é possível usar várias ao mesmo tempo. Os exemplos abaixo são limitados a uma única restrição para fins ilustrativos.

Controle de recursos com **systemd-run** para execução de comandos únicos

Embora comumente associado aos serviços do sistema, `systemd` também pode ser usado por usuários que não são raiz para executar serviços, agendar temporizadores ou executar processos pontuais. No exemplo a seguir, vamos usar `stress-ng` como a aplicação. No primeiro exemplo, vamos executá-la usando `systemd-run` na conta padrão `ec2-user` e, no segundo exemplo, vamos aplicar limites quanto ao uso da CPU.

Example: usar `systemd-run` na linha de comandos para executar um processo, sem limitar o uso de recursos

1. Garanta que o pacote `stress-ng` esteja instalado, pois vamos usá-lo no exemplo.

```
[ec2-user ~]$ sudo dnf install -y stress-ng
```

2. Use `systemd-run` para executar um teste de estresse da CPU de 10 segundos sem limitar a quantidade de CPU que ele pode usar.

```
[ec2-user ~]$ systemd-run --user --tty --wait --property=CPUAccounting=1 stress-ng
--cpu 1 --timeout 10
Running as unit: run-u6.service
Press ^] three times within 1s to disconnect TTY.
stress-ng: info: [339368] setting to a 10 second run per stressor
stress-ng: info: [339368] dispatching hogs: 1 cpu
stress-ng: info: [339368] successful run completed in 10.00s
Finished with result: success
Main processes terminated with: code=exited/status=0
Service runtime: 10.068s
CPU time consumed: 9.060s
```

A opção `--user` instrui `systemd-run` a executar o comando como o usuário conectado, a opção `--tty` significa que há um TTY anexado, `--wait` indica para esperar até que o serviço seja concluído e a opção `--property=CPUAccounting=1` instrui `systemd-run` a registrar quanto tempo de CPU é usado na execução do processo. A opção da linha de comandos `--property` pode ser usada para enviar as configurações de `systemd-run` que podem estar definidas em um arquivo de configuração `systemd.unit`.

Quando instruído a colocar carga na CPU, o programa `stress-ng` usará todo o tempo de CPU disponível para realizar o teste durante o período solicitado. Para uma aplicação do mundo real, pode ser desejável limitar o tempo total de execução de um processo. No exemplo abaixo, solicitaremos que `stress-ng` seja executado por um tempo maior do que a restrição de duração máxima que foi imposta usando `systemd-run`.

Example: usar `systemd-run` na linha de comandos para executar um processo, limitando o uso da CPU a 1 segundo

1. Verifique se `stress-ng` está instalado para executar este exemplo.

2. A propriedade `LimitCPU` é equivalente a `ulimit -t`, que limitará o tempo máximo que esse processo poderá usar a CPU. Nesse caso, como estamos solicitando uma execução de estresse de 10 segundos e limitando o uso da CPU a 1 segundo, o comando receberá um sinal `SIGXCPU` e falhará.

```
[ec2-user ~]$ systemd-run --user --tty --wait --property=CPUAccounting=1 --  
property=LimitCPU=1 stress-ng --cpu 1 --timeout 10  
Running as unit: run-u12.service  
Press ^] three times within 1s to disconnect TTY.  
stress-ng: info: [340349] setting to a 10 second run per stressor  
stress-ng: info: [340349] dispatching hogs: 1 cpu  
stress-ng: fail: [340349] cpu instance 0 corrupted bogo-ops counter, 1370 vs 0  
stress-ng: fail: [340349] cpu instance 0 hash error in bogo-ops counter and run  
flag, 3250129726 vs 0  
stress-ng: fail: [340349] metrics-check: stressor metrics corrupted, data is  
compromised  
stress-ng: info: [340349] unsuccessful run completed in 1.14s  
Finished with result: exit-code  
Main processes terminated with: code=exited/status=2  
Service runtime: 1.201s  
CPU time consumed: 1.008s
```

Mais comumente, talvez você queira restringir a porcentagem do tempo da CPU que pode ser consumido por um processo específico. No exemplo abaixo, restringiremos a porcentagem do tempo da CPU que pode ser consumido por `stress-ng`. Para um serviço do mundo real, pode ser desejável limitar a porcentagem máxima do tempo da CPU que um processo em segundo plano pode consumir para deixar os recursos livres para o processo que atende às solicitações do usuário.

Example: usar `systemd-run` para limitar um processo a 10% do tempo de uma CPU

1. Verifique se `stress-ng` está instalado para executar este exemplo.
2. Vamos usar a propriedade `CPUQuota` para instruir `systemd-run` a restringir o uso da CPU pelo comando que vamos executar. Não vamos limitar a quantidade de tempo que o processo pode permanecer em execução, apenas a quantidade de CPU que ele pode usar.

```
[ec2-user ~]$ systemd-run --user --tty --wait --property=CPUAccounting=1 --  
property=CPUQuota=10% stress-ng --cpu 1 --timeout 10  
Running as unit: run-u13.service  
Press ^] three times within 1s to disconnect TTY.
```

```
stress-ng: info:  [340664] setting to a 10 second run per stressor
stress-ng: info:  [340664] dispatching hogs: 1 cpu
stress-ng: info:  [340664] successful run completed in 10.08s
Finished with result: success
Main processes terminated with: code=exited/status=0
Service runtime: 10.140s
CPU time consumed: 1.014s
```

Observe como a contabilidade da CPU nos informa que, embora o serviço tenha sido executado por 10 segundos, ele consumiu apenas 1 segundo de tempo real da CPU.

Há várias maneiras de configurar `systemd` para limitar o uso de recursos de CPU, memória, rede e E/S. Consulte a documentação do `systemd` upstream para [systemd.resource-control](#) ou a página man para `systemd.resource-control` em uma instância do AL2023 para obter informações abrangentes.

Nos bastidores, `systemd` usa recursos de kernel do Linux, como `cgroups`, para implementar esses limites, evitando a necessidade de configurá-los manualmente. A [documentação de kernel do Linux para cgroup-v2](#) contém muitos detalhes sobre como `cgroups` funcionam.

Controle de recursos em um serviço de **systemd**

Há vários parâmetros que podem ser adicionados à seção `[Service]` dos serviços de `systemd` para controlar o uso dos recursos do sistema. Isso inclui limites rígidos e flexíveis. Para conhecer o comportamento exato de cada opção, consulte a documentação de `systemd` upstream para [systemd.resource-control](#) ou a página man para `systemd.resource-control` em uma instância do AL2023.

Os limites comumente usados são `MemoryHigh` para especificar um limite de controle de utilização para o uso de memória, `MemoryMax` para definir um limite superior rígido (uma vez atingido, o OOM Killer é invocado) e `CPUQuota` (conforme ilustrado na seção anterior). Também é possível configurar pesos e prioridades em vez de números fixos.

Example: usar `systemd` para definir limites de uso de memória para os serviços

Neste exemplo, definiremos um limite rígido de uso de memória para `memcached`, um cache simples de valores-chave, e mostraremos como o OOM Killer é invocado para esse serviço e não para todo o sistema.

1. Primeiro, precisamos instalar os pacotes necessários para este exemplo.

```
[ec2-user ~]$ sudo dnf install -y memcached libmemcached-awesome-tools
```

2. Habilite `memcached.service`, depois inicie o serviço para que `memcached` esteja em execução.

```
[ec2-user ~]$ sudo systemctl enable memcached.service
Created symlink /etc/systemd/system/multi-user.target.wants/memcached.service # /usr/lib/systemd/system/memcached.service.
[ec2-user ~]$ sudo systemctl start memcached.service
```

3. Verifique se `memcached.service` está em execução.

```
[ec2-user ~]$ sudo systemctl status memcached.service
# memcached.service - memcached daemon
   Loaded: loaded (/usr/lib/systemd/system/memcached.service; enabled; preset: disabled)
   Active: active (running) since Fri 2025-01-31 22:36:42 UTC; 1s ago
     Main PID: 356294 (memcached)
        Tasks: 10 (limit: 18907)
       Memory: 1.8M
          CPU: 20ms
      CGroup: /system.slice/memcached.service
             ##356294 /usr/bin/memcached -p 11211 -u memcached -m 64 -c 1024 -l 127.0.0.1,::1

Jan 31 22:35:36 ip-1-2-3-4.us-west-2.compute.internal systemd[1]: Started memcached.service - memcached daemon.
```

4. Agora que `memcached` está instalado e em execução, podemos observar que ele funciona inserindo alguns dados aleatórios no cache.

Em `/etc/sysconfig/memcached`, a variável `CACHESIZE` é definida como 64 por padrão, o que significa 64 megabytes. Ao inserir mais dados do que o tamanho máximo do cache, podemos ver que preenchemos o cache e alguns itens são removidos usando `memcached-tool`, e que `memcached.service` está usando cerca de 64 MB de memória.

```
[ec2-user ~]$ for i in $(seq 1 150); do dd if=/dev/random of=$i bs=512k count=1; memcp -s localhost $i; done
[ec2-user ~]$ memcached-tool localhost display
# Item_Size Max_age Pages Count Full? Evicted Evict_Time OOM
```

```

 2      120B      0s      1      0      no      0      0      0
39    512.0K      4s     63     126     yes     24      2      0
[ec2-user ~]$ sudo systemctl status memcached.service
# memcached.service - memcached daemon
   Loaded: loaded (/usr/lib/systemd/system/memcached.service; enabled; preset:
disabled)
   Active: active (running) since Fri 2025-01-31 22:36:42 UTC; 7min ago
 Main PID: 356294 (memcached)
    Tasks: 10 (limit: 18907)
   Memory: 66.7M
      CPU: 203ms
   CGroup: /system.slice/memcached.service
           ##356294 /usr/bin/memcached -p 11211 -u memcached -m 64 -c 1024 -l
127.0.0.1,::1

Jan 31 22:36:42 ip-1-2-3-4.us-west-2.compute.internal systemd[1]: Started
memcached.service - memcached daemon.

```

5. Use a propriedade `MemoryMax` para definir um limite rígido para `memcached.service`. Se esse limite atingido, o OOM Killer será invocado. Opções adicionais podem ser definidas para o serviço adicionando-as a um arquivo de substituição. Isso pode ser feito editando diretamente o arquivo `/etc/systemd/system/memcached.service.d/override.conf` ou de forma interativa, usando o comando `edit` de `systemctl`.

```
[ec2-user ~]$ sudo systemctl edit memcached.service
```

Adicione o seguinte à substituição para definir um limite rígido de 32 MB de memória para o serviço.

```
[Service]
MemoryMax=32M
```

6. Instrua `systemd` a recarregar sua configuração.

```
[ec2-user ~]$ sudo systemctl daemon-reload
```

7. Observe que `memcached.service` agora está sendo executado com um limite de memória de 32 MB.

```
[ec2-user ~]$ sudo systemctl status memcached.service
# memcached.service - memcached daemon
   Loaded: loaded (/usr/lib/systemd/system/memcached.service; enabled; preset:
disabled)
   Drop-In: /etc/systemd/system/memcached.service.d
            ##override.conf
   Active: active (running) since Fri 2025-01-31 23:09:13 UTC; 49s ago
 Main PID: 358423 (memcached)
    Tasks: 10 (limit: 18907)
   Memory: 1.8M (max: 32.0M available: 30.1M)
      CPU: 25ms
   CGroup: /system.slice/memcached.service
           ##358423 /usr/bin/memcached -p 11211 -u memcached -m 64 -c 1024 -l
127.0.0.1,::1

Jan 31 23:09:13 ip-1-2-3-4.us-west-2.compute.internal systemd[1]: Started
memcached.service - memcached daemon.
```

8. O serviço funcionará normalmente usando menos de 32 MB de memória, o que podemos confirmar ao carregar menos de 32 MB de dados aleatórios no cache e verificar o status do serviço.

```
[ec2-user ~]$ for i in $(seq 1 30); do dd if=/dev/random of=$i bs=512k count=1;
memcp -s localhost $i; done
```

```
[ec2-user ~]$ sudo systemctl status memcached.service
# memcached.service - memcached daemon
   Loaded: loaded (/usr/lib/systemd/system/memcached.service; enabled; preset:
disabled)
   Drop-In: /etc/systemd/system/memcached.service.d
            ##override.conf
   Active: active (running) since Fri 2025-01-31 23:14:48 UTC; 3s ago
 Main PID: 359492 (memcached)
    Tasks: 10 (limit: 18907)
   Memory: 18.2M (max: 32.0M available: 13.7M)
      CPU: 42ms
   CGroup: /system.slice/memcached.service
           ##359492 /usr/bin/memcached -p 11211 -u memcached -m 64 -c 1024 -l
127.0.0.1,::1
```

```
Jan 31 23:14:48 ip-1-2-3-4.us-west-2.compute.internal systemd[1]: Started
memcached.service - memcached daemon.
```

9. Agora podemos fazer com que memcached use mais de 32 MB de memória, tentando usar os 64 MB completos de cache que são a configuração padrão de memcached.

```
[ec2-user ~]$ for i in $(seq 1 150); do dd if=/dev/random of=$i bs=512k count=1;
memcp -s localhost $i; done
```

Você observará que em algum momento durante o comando acima, há erros de conexão com o servidor de memcached. Isso ocorre porque o OOM Killer encerrou o processo devido à restrição que impusemos a ele. O restante do sistema funcionará normalmente e nenhum outro processo será considerado pelo OOM Killer, já que restringimos apenas `memcached.service`.

```
[ec2-user ~]$ sudo systemctl status memcached.service
# memcached.service - memcached daemon
   Loaded: loaded (/usr/lib/systemd/system/memcached.service; enabled; preset:
disabled)
   Drop-In: /etc/systemd/system/memcached.service.d
           ##override.conf
   Active: failed (Result: oom-kill) since Fri 2025-01-31 23:20:28 UTC; 2s ago
 Duration: 2.901s
   Process: 360130 ExecStart=/usr/bin/memcached -p ${PORT} -u ${USER} -m
${CACHE_SIZE} -c ${MAXCONN} $OPTIONS (code=killed, signal=KILL)
  Main PID: 360130 (code=killed, signal=KILL)
    CPU: 94ms
```

```
Jan 31 23:20:25 ip-1-2-3-4.us-west-2.compute.internal systemd[1]: Started
memcached.service - memcached daemon.
Jan 31 23:20:28 ip-1-2-3-4.us-west-2.compute.internal systemd[1]:
memcached.service: A process of this unit has been killed by the OOM killer.
Jan 31 23:20:28 ip-1-2-3-4.us-west-2.compute.internal systemd[1]:
memcached.service: Main process exited, code=killed, status=9/KILL
Jan 31 23:20:28 ip-1-2-3-4.us-west-2.compute.internal systemd[1]:
memcached.service: Failed with result 'oom-kill'.
```

Limitar o uso de recursos do processo no AL2023 usando cgroups

Embora seja recomendável usar [Controle de recursos com systemd](#), esta seção aborda o uso básico dos utilitários `libcgroup-tools` básicos para limitar o uso da CPU e da memória pelos processos.

Ambos os métodos são alternativas ao uso do utilitário [cpulimit](#), encontrado anteriormente em [EPEL](#).

O exemplo abaixo abrange a execução do teste de estresse `stress-ng` (do pacote `stress-ng`) enquanto limita o uso de CPU e memória usando utilitários do pacote `libcgroup-tools` e os ajustáveis em `sysfs`.

Usar **libcgroup-tools** na linha de comandos para limitar o uso de recursos

1. Instale o pacote `libcgroup-tools`.

```
[ec2-user ~]$ sudo dnf install libcgroup-tools
```

2. Crie um cgroup com os controladores `memory` e `cpu`, e dê a ele um nome (`our-example-limits`). Use as opções `-a` e `-t` para permitir que o usuário `ec2-user` controle os ajustáveis do cgroup.

```
[ec2-user ~]$ sudo cgcreate -a ec2-user -t ec2-user -g memory,cpu:our-example-limits
```

Agora existe um diretório `/sys/fs/cgroup/our-example-limits/` que contém arquivos que podem ser usados para controlar cada ajustável.

Note

O Amazon Linux 2 usa `cgroup-v1` em vez de `cgroup-v2`, que é usado no AL2023. No AL2, os caminhos `sysfs` são diferentes, e haverá diretórios `/sys/fs/cgroup/memory/our-example-limits` e `/sys/fs/cgroup/cpu/our-example-limits` pertencentes a `ec2-user` que conterão arquivos que podem ser usados para controlar os limites de cgroup.

3. Limite o uso de memória por todos os processos em cgroup a 100 milhões de bytes.

```
[ec2-user ~]$ echo 100000000 > /sys/fs/cgroup/our-example-limits/memory.max
```


 Note

O Amazon Linux 2 usa cgroup-v1 em vez de cgroup-v2, que é usado pelo Amazon Linux 2023. Isso significa que alguns ajustáveis são diferentes. Para limitar o uso de memória no AL2, o ajustável abaixo é usado em vez disso.

```
[ec2-user ~]$ echo 10000000 > /sys/fs/cgroup/memory/our-example-limits/  
memory.limit_in_bytes
```

4. Limite o uso de CPU por todos os processos em cgroup a 10%. O formato do arquivo `cpu.max` é `$MAX $PERIOD`, limitando o grupo a consumir `$MAX` para cada `$PERIOD`.

```
[ec2-user ~]$ echo 10000 100000 > /sys/fs/cgroup/our-example-limits/cpu.max
```

O Amazon Linux 2 usa cgroup-v1 em vez de cgroup-v2, que é usado pelo Amazon Linux 2023. Isso significa que alguns ajustáveis são diferentes, incluindo como limitar o uso da CPU.

5. O exemplo abaixo executa `stress-ng` (que pode ser instalado executando `dnf install -y stress-ng`) em `our-example-limits` cgroup. Enquanto o comando `stress-ng` está em execução, você pode usar `top` para observar a limitação em 10% do tempo da CPU.

```
[ec2-user ~]$ sudo cgexec -g memory,cpu:our-example-limits stress-ng --cpu 1
```

6. Remova cgroup.

```
[ec2-user ~]$ sudo cgdelete -g memory,cpu:our-example-limits
```

A [documentação de kernel do Linux para cgroup-v2](#) contém muitos detalhes sobre como esse mecanismo funciona. A documentação dos controladores [cpu](#) e [memory](#) abrange os detalhes de como usar cada opção ajustável.

Usando AL2023 em AWS

Você pode configurar AL2023 para uso com outros Serviços da AWS. Por exemplo, você pode escolher uma AL2023 AMI ao iniciar uma instância do [Amazon Elastic Compute Cloud](#) (Amazon EC2).

Para esses procedimentos de configuração, você usa o serviço AWS Identity and Access Management (IAM). Para obter mais informações sobre o IAM, consulte os seguintes materiais de referência:

- [AWS Identity and Access Management \(IAM\)](#)
- [Guia do usuário do IAM](#)

Tópicos

- [Começando com AWS](#)
- [AL2023 no Amazon EC2](#)
- [Uso AL2023 em contêineres](#)
- [AL2023 no AWS Elastic Beanstalk](#)
- [Usar o AL2023 no AWS CloudShell](#)
- [Usar AMIs do Amazon ECS baseadas no AL2023 para hospedar workloads em contêineres](#)
- [Usar o Amazon Elastic File System no AL2023](#)
- [Usando o Amazon EMR baseado em AL2023](#)
- [Usando AL2023 em AWS Lambda](#)

Começando com AWS

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a <https://portal.aws.amazon.com/billing/inscrição>.

2. Siga as instruções online.

Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando <https://aws.amazon.com/e> escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS Centro de Identidade do AWS IAM, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login [Console de gerenciamento da AWS](#) como proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

Para obter ajuda ao fazer login usando o usuário-raiz, consulte [Fazer login como usuário-raiz](#) no Guia do usuário do Início de Sessão da AWS .

2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

Para obter instruções, consulte [Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz \(console\) no Guia](#) do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

Para obter instruções, consulte [Habilitar o Centro de Identidade do AWS IAM](#) no Guia do usuário do Centro de Identidade do AWS IAM .

2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM](#) no Guia Centro de Identidade do AWS IAM do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte [Como fazer login no portal de AWS acesso](#) no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

Para obter instruções, consulte [Criar um conjunto de permissões](#) no Guia do usuário do Centro de Identidade do AWS IAM .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de logon único ao grupo.

Para obter instruções, consulte [Adicionar grupos](#) no Guia do usuário do Centro de Identidade do AWS IAM .

Conceder acesso programático

Os usuários precisam de acesso programático se quiserem interagir com pessoas AWS fora do Console de gerenciamento da AWS. A forma de conceder acesso programático depende do tipo de usuário que está acessando AWS.

Para conceder acesso programático aos usuários, selecione uma das seguintes opções:

Qual usuário precisa de acesso programático?	Para	Por
IAM	(Recomendado) Use as credenciais do console como	Siga as instruções da interface que deseja utilizar.

Qual usuário precisa de acesso programático?	Para	Por
	credenciais temporárias para assinar solicitações programáticas para o AWS CLI, AWS SDKs ou. AWS APIs	• Para o AWS CLI, consulte Login para desenvolvimento AWS local no Guia AWS Command Line Interface do usuário. • Para AWS SDKs isso, consulte Login para desenvolvimento AWS local no Guia de referência de ferramentas AWS SDKs e ferramentas.
Identidade da força de trabalho (Usuários gerenciados no Centro de Identidade do IAM)	Use credenciais temporárias para assinar solicitações programáticas para o AWS CLI AWS SDKs, ou. AWS APIs	Siga as instruções da interface que deseja utilizar. • Para o AWS CLI, consulte Configurando o AWS CLI para uso Centro de Identidade do AWS IAM no Guia do AWS Command Line Interface usuário. • Para AWS SDKs, ferramentas e AWS APIs, consulte a autenticação do IAM Identity Center no Guia de referência de ferramentas AWS SDKs e ferramentas.
IAM	Use credenciais temporárias para assinar solicitações programáticas para o AWS CLI AWS SDKs, ou. AWS APIs	Siga as instruções em Como usar credenciais temporárias com AWS recursos no Guia do usuário do IAM.

Qual usuário precisa de acesso programático?	Para	Por
IAM	(Não recomendado) Use credenciais de longo prazo para assinar solicitações programáticas para o AWS CLI, AWS SDKs, ou. AWS APIs	Siga as instruções da interface que deseja utilizar. • Para isso AWS CLI, consulte Autenticação usando credenciais de usuário do IAM no Guia do AWS Command Line Interface usuário. • Para ferramentas AWS SDKs e ferramentas, consulte Autenticar usando credenciais de longo prazo no Guia de referência de ferramentas AWS SDKs e ferramentas. • Para isso AWS APIs, consulte Gerenciamento de chaves de acesso para usuários do IAM no Guia do usuário do IAM.

AL2023 no Amazon EC2

Use um dos procedimentos a seguir para iniciar uma instância do Amazon EC2 com uma AMI AL2023 . Você pode escolher a AMI padrão ou a AMI mínima. Para obter mais informações sobre as diferenças entre a AMI padrão e a AMI mínima, consulte [Comparando AL2023 padrão \(padrão\) e mínimo AMIs](#).

Tópicos

- [Inicializar o AL2023 usando o console do Amazon EC2](#)
- [Lançamento AL2023 usando o parâmetro SSM e AWS CLI](#)

- [Iniciando a AL2023 AMI mais recente usando CloudFormation](#)
- [Lançamento AL2023 usando uma ID de AMI específica](#)
- [AL2023 Depreciação e ciclo de vida da AMI](#)
- [Conexão com AL2023 instâncias](#)
- [Comparando o AL2023 padrão e o mínimo AMIs](#)

Inicializar o AL2023 usando o console do Amazon EC2

Use o console do Amazon EC2 para iniciar uma AMI AL2023 .

Note

Para instâncias baseadas em ARM, AL2023 só é compatível com tipos de instância que usam processadores Graviton2 ou posteriores. AL2023 não oferece suporte a instâncias A1.

Usar as seguintes etapas. Para iniciar uma instância do Amazon EC2 com uma AMI AL2023 a partir do console do Amazon EC2.

Para iniciar uma instância do EC2 com uma AMI AL2023

1. Abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
2. No painel de navegação, escolha AMIs.
3. No menu suspenso de filtros, escolha Imagens públicas.
4. No campo de pesquisa, digite **al2023-ami**.

Note

Certifique-se de que a Amazon apareça na coluna Alias do proprietário.

5. Selecione uma imagem da lista. Em Origem, você pode determinar se a AMI é padrão ou mínima. Um nome de AL2023 AMI pode ser interpretado usando este formato:

```
'al2023-[ami || ami-minimal]-2023.0.[release build date].[build number]-kernel-[version number]-[arm64 || x86_64]'
```

6. A imagem a seguir mostra uma lista parcial de AL2023 AMIs.

Capacity Reservations	Amazon Machine Images (AMIs) (4) Info Public images Q Search al2023-ami Clear filters Recycle Bin EC2 Image Builder Actions Launch instance from AMI					
▼ Images						
AMIs						
AMI Catalog						
▼ Elastic Block Store						
Volumes						
Snapshots						
Lifecycle Manager						
	☐	Name	AMI ID	AMI name	Source	Owner
	☐	-	ami-000a4d9c6067d5d0d	al2023-ami-2023.0.20230222.1...	amazon/al2023-ami-2023.0.20230222.1-kernel-6.1-arm64	137112412989 amazon
	☐	-	ami-0a409f3927bd2662f	al2023-ami-2023.0.20230222.1...	amazon/al2023-ami-2023.0.20230222.1-kernel-6.1-x86_64	137112412989 amazon
	☐	-	ami-043e11d11db3d437e	al2023-ami-minimal-2023.0.20...	amazon/al2023-ami-minimal-2023.0.20230222.1-kernel-6.1-ar...	137112412989 amazon
	☐	-	ami-0d19aa82c9a61ef2c	al2023-ami-minimal-2023.0.20...	amazon/al2023-ami-minimal-2023.0.20230222.1-kernel-6.1-x8...	137112412989 amazon

Para saber mais sobre como inicializar instâncias do Amazon EC2, consulte [Get started with Amazon EC2 Linux instances](#) no Guia do usuário do Amazon EC2.

Lançamento AL2023 usando o parâmetro SSM e AWS CLI

No AWS CLI, você pode usar o valor do parâmetro SSM da AMI para iniciar uma nova instância do AL2023. Mais especificamente, use um dos valores dinâmicos do parâmetro SSM da lista a seguir e adicione `/aws/service/ami-amazon-linux-latest/` antes do valor do parâmetro SSM/. É possível usar uma instância usando a AWS CLI.

- `al2023-ami-kernel-default-arm64` para a arquitetura arm64
- `al2023-ami-minimal-kernel-default-arm64` para arquitetura arm64 (AMI mínima)
- `al2023-ami-kernel-default-x86_64` para a arquitetura x86_64
- `al2023-ami-minimal-kernel-default-x86_64` para a arquitetura x86_64 (AMI mínima)

Note

Cada um dos *italic* itens é um exemplo de parâmetro. Substitua-os por suas próprias informações.

```
$ aws ec2 run-instances \
  --image-id \
    resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64 \
  --instance-type m5.xlarge \
  --region us-east-1 \
  --key-name aws-key-us-east-1 \
  --security-group-ids sg-004a7650
```

O sinalizador `--image-id` especifica o valor do parâmetro SSM.

O sinalizador `--instance-type` especifica o tipo e o tamanho da instância. Esse sinalizador deve ser compatível com o tipo de AMI selecionado.

A `--region` sinalização especifica Região da AWS onde você cria sua instância.

A `--key-name` sinalização especifica Região da AWS a chave s usada para se conectar à instância. Se você não fornecer uma chave que exista na região em que você criou a instância, não poderá se conectar à instância usando SSH.

O sinalizador `--security-group-ids` especifica o grupo de segurança que determina as permissões de acesso para tráfego de rede de entrada e saída.

Important

Isso AWS CLI exige que você especifique um grupo de segurança existente que permita acesso à instância de sua máquina remota pela portaTCP:22. Sem um grupo de segurança especificado, sua nova instância é colocada em um grupo de segurança padrão. Em um grupo de segurança padrão, sua instância só pode se conectar às outras instâncias dentro da sua VPC.

Para obter mais informações, consulte o [Lançamento, listagem e encerramento de instâncias do Amazon EC2](#) no Guia do usuário da AWS Command Line Interface .

Iniciando a AL2023 AMI mais recente usando CloudFormation

Para iniciar uma AL2023 AMI usando CloudFormation, use um dos modelos a seguir.

Note

Arm64 AMIs Cada um deles exige tipos de instância diferentes. x86_64 Para obter mais informações, consulte [Tipos de instâncias do Amazon EC2](#)

Modelo JSON:

```
{
  "Parameters": {
```

```

    "LatestAmiId": {
      "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
      "Default": "/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-
default-x86_64"
    },
    "Resources": {
      "MyEC2Instance": {
        "Type": "AWS::EC2::Instance",
        "Properties": {
          "InstanceType": "t2.large",
          "ImageId": {
            "Ref": "LatestAmiId"
          }
        }
      }
    }
  }
}

```

Modelo YAML:

```

Parameters:
  LatestAmiId:
    Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
    Default: '/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-default-
x86_64'

Resources:
  Instance:
    Type: 'AWS::EC2::Instance'
    Properties:
      InstanceType: 't2.large'
      ImageId: !Ref LatestAmiId

```

Certifique-se de substituir o parâmetro AMI no final da seção “Padrão”, se necessário. Os seguintes valores de parâmetros estão disponíveis:

- al2023-ami-kernel-6.1-arm64 para a arquitetura arm64
- al2023-ami-minimal-kernel-6.1-arm64 para arquitetura arm64 (AMI mínima)
- al2023-ami-kernel-6.1-x86_64 para a arquitetura x86_64
- al2023-ami-minimal-kernel-6.1-x86_64 para a arquitetura x86_64 (AMI mínima)

A seguir estão as especificações dinâmicas do kernel. A versão padrão do kernel muda automaticamente com cada atualização da versão principal do kernel.

- `al2023-ami-kernel-default-arm64` para a arquitetura `arm64`
- `al2023-ami-minimal-kernel-default-arm64` para arquitetura `arm64` (AMI mínima)
- `al2023-ami-kernel-default-x86_64` para a arquitetura `x86_64`
- `al2023-ami-minimal-kernel-default-x86_64` para a arquitetura `x86_64` (AMI mínima)

Lançamento AL2023 usando uma ID de AMI específica

Você pode iniciar uma AMI AL2023 específica usando a ID da AMI. Você pode determinar qual ID da AMI AL2023 é necessária consultando a lista de AMI no console do Amazon EC2. Ou você pode usar AWS Systems Manager. Se você estiver usando o Systems Manager, certifique-se de selecionar o alias da AMI dentre os listados na seção anterior. Para obter mais informações, consulte [Consultar a AMI mais recente do Amazon Linux IDs usando o AWS Systems Manager Parameter Store](#).

AL2023 Depreciação e ciclo de vida da AMI

Cada nova AL2023 versão inclui uma nova AMI. Quando a AMI é registrada, ela é marcada com uma data de suspensão de uso. A data de suspensão de uso de cada AL2023 AMI é de 90 dias a partir do momento em que foi lançada, de acordo com o período oferecido para cada lançamento individual do kernel. [Aplicação de patches do kernel em tempo real no AL2023](#)

Note

A data de suspensão de uso de 90 dias se refere a uma AMI individual e não ao período de suporte do AL2023 [Cadência de lançamento](#) produto.

Para saber mais sobre descontinuação de AMI, consulte [Descontinuar uma AMI](#) no Guia do usuário do Amazon EC2.

O uso regular de uma AMI atualizada para executar uma instância garante que a instância comece com as atualizações de segurança mais recentes, incluindo um kernel atualizado. Se você iniciar uma versão anterior de uma AMI e aplicar atualizações, haverá um período em que a instância não terá as atualizações de segurança mais recentes. Para garantir que você esteja usando a AMI mais recente, recomendamos usar os parâmetros do SSM.

Para obter mais informações sobre como usar os parâmetros do SSM para iniciar uma instância, consulte:

- [Lançamento AL2023 usando o parâmetro SSM e AWS CLI](#)
- [Iniciando a AL2023 AMI mais recente usando CloudFormation](#)

Conexão com AL2023 instâncias

Use SSH ou AWS Systems Manager para se conectar à sua AL2023 instância.

Conectar a sua instância usando SSH

Para obter instruções sobre como usar SSH para se conectar a uma instância, consulte [Conectar-se à instância do Linux usando um cliente SSH](#) no Guia do usuário do Amazon EC2.

Conecte-se à sua instância usando AWS Systems Manager

Para obter instruções sobre como usar AWS Systems Manager para se conectar a uma AL2023 instância, consulte [Conecte-se à sua instância Linux usando o Session Manager](#) no Guia do usuário do Amazon EC2.

Usar o Amazon EC2 Instance Connect

A AMI do AL2023, exceto a AMI mínima, vem com o agente do EC2 Instance Connect instalado por padrão. Para usar o EC2 Instance Connect com uma AL2023 instância executada a partir da AMI mínima, você deve instalar o `ec2-instance-connect` pacote. Para obter instruções sobre como usar o EC2 Instance Connect, consulte [Conecte-se à sua instância Linux usando um endereço IP público e o EC2 Instance Connect](#) no Guia do usuário do Amazon EC2.

Comparando o AL2023 padrão e o mínimo AMIs

Você pode inicializar uma instância do Amazon EC2 com uma AMI padrão (predefinição) ou mínima do AL2023. Para obter instruções sobre como inicializar uma instância do Amazon EC2 com o tipo de AMI padrão ou mínima, consulte [AL2023 no Amazon EC2](#).

A AL2023 AMI padrão vem com todos os aplicativos e ferramentas mais usados instalados. Recomendamos a AMI padrão se você quiser começar rapidamente e não estiver interessado em personalizar a AMI.

A AL2023 AMI mínima é a versão básica e simplificada que contém somente as ferramentas e os utilitários mais básicos necessários para executar o sistema operacional (SO). Recomendamos a

AMI mínima se você quiser ter o menor espaço de sistema operacional possível. A AMI mínima oferece uma utilização ligeiramente reduzida do espaço em disco e melhor eficiência de custos a longo prazo. A AMI mínima é adequada se você deseja um sistema operacional menor e não se importa em instalar ferramentas e aplicativos manualmente.

A imagem do contêiner está mais próxima da AMI AL2023 mínima no conjunto de pacotes.

Comparar pacotes instalados em imagens Amazon Linux 2023

Uma comparação do RPMs presente nas imagens da AL2023 AMI, da AMI mínima e do contêiner.

Pacote	AMI	AMI mínima	Contêiner
acl	2.3.1		
acpid	2.0.32		
alternatives	1.15	1.15	1.15
amazon-chrony-config	4.3	4.3	
amazon-ec2-net-utils	2.5.1	2.5.1	
amazon-linux-repo-cdn			2023.6.20241031
amazon-linux-repo-s3	2023.6.20241031	2023.6.20241031	
amazon-linux-sb-keys	2023.1	2023.1	
amazon-rpm-config	228		
amazon-ssm-agent	3.3.987.0		

Pacote	AMI	AMI mínima	Contêiner
amd-ucode-firmware	20210208 (noarch)	20210208 (noarch)	
at	3.1.23		
attr	2.5.1		
audit	3.0.6	3.0.6	
audit-libs	3.0.6	3.0.6	3.0.6
aws-cfn-bootstrap	2,0		
awscli-2	2.15.30	2.15.30	
basesystem	11	11	11
bash	5.2.15	5.2.15	5.2.15
bash-completion	2.11		
bc	1.07.1		
bind-libs	9.18.28		
bind-license	9.18.28		
bind-utils	9.18.28		
binutils	2,39		
boost-fil esystem	1.75.0		
boost-system	1.75.0		
boost-thread	1.75.0		

Pacote	AMI	AMI mínima	Contêiner
bzip2	1.0.8		
bzip2-libs	1.0.8	1.0.8	1.0.8
ca-certificates	2023.2.68	2023.2.68	2023.2.68
c-ares	1.19.1		
checkpolicy	3.4	3.4	
chkconfig	1.15		
chrony	4.3	4.3	
cloud-init	22.2.2	22.2.2	
cloud-init-cfg-ec2	22.2.2	22.2.2	
cloud-utils-growpart	0,31	0,31	
coreutils	8,32	8,32	
coreutils-common	8,32	8,32	
coreutils-single			8,32
cpio	2.13	2.13	
cracklib	2.9.6	2.9.6	
cracklib-dicts	2.9.6	2.9.6	
crontabs	1.11		
crypto-policies	2020428	2020428	2020428

Pacote	AMI	AMI mínima	Contêiner
crypto-policies-scripts	2020428		
cryptsetup	2.6.1		
cryptsetup-libs	2.6.1	2.6.1	
curl-minimal	8.5.0	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27	
cyrus-sasl-plain	2.1.27		
dbus	1.12.28	1.12.28	
dbus-broker	32	32	
dbus-common	1.12.28	1.12.28	
dbus-libs	1.12.28	1.12.28	
device-mapper	1.02.185	1.02.185	
device-mapper-libs	1.02.185	1.02.185	
diffutils	3.8	3.8	
dnf	4.14.0	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2	
dnf-plugins-core	4.3.0	4.3.0	

Pacote	AMI	AMI mínima	Contêiner
dnf-plugin-support-info	1.2	1.2	
dnf-utils	4.3.0		
dosfstools	4.2		
dracut	055	055	
dracut-config-ec2	3.0	3.0	
dracut-config-generic	055	055	
dwz	0,14		
dyninst	10.2.1		
e2fsprogs	1.46,5	1.46,5	
e2fsprogs-libs	1.46,5	1.46,5	
ec2-hibinit-agent	1.0.8		
ec2-instance-connect	1.1		
ec2-instance-connect-selinux	1.1		
ec2-utils	2.2.0	2.2.0	
ed	1.14.2		
efi-filesystem	5	5	

Pacote	AMI	AMI mínima	Contêiner
efi-srpm-macros	5		
efivar	38	38	
efivar-libs	38	38	
elfutils- debuginfod- client	0.188		
elfutils- default-yama- scope	0.188	0.188	0.188
elfutils-libelf	0.188	0.188	0.188
elfutils-libs	0.188	0.188	0.188
ethtool	5.15		
expat	2.5.0	2.5.0	2.5.0
file	5,39	5,39	
file-libs	5,39	5,39	5,39
filesystem	3,14	3,14	3,14
findutils	4.8.0	4.8.0	
fonts-srpm- macros	2.0.5		
fstrm	0.6.1		
fuse-libs	2.9.9	2.9.9	
gawk	5.1.0	5.1.0	5.1.0

Pacote	AMI	AMI mínima	Contêiner
gdbm-libs	1,19	1,19	1,19
gdisk	1.0.8	1.0.8	
gettext	0,21	0,21	
gettext-libs	0,21	0,21	
ghc-srpm-macros	1.5.0		
glib2	2.74.7	2.74.7	2.74.7
glibc	2.34	2.34	2.34
glibc-all-langpacks	2.34	2.34	
glibc-common	2.34	2.34	2.34
glibc-gconv-extra	2.34		
glibc-locale-source	2.34	2.34	
glibc-minimal-langpack			2.34
gmp	6.2.1	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7	2.3.7
gnutls	3.8.0	3.8.0	
go-srpm-macros	3.2.0		
gpgme	1.15.1	1.15.1	1.15.1
gpm-libs	1.20.7		

Pacote	AMI	AMI mínima	Contêiner
grep	3.8	3.8	3.8
groff-base	1.22.4	1.22.4	
grub2-common	2.06	2.06	
grub2-efi-aa64-ec2	2.06 (aarch64)	2.06 (aarch64)	
grub2-efi-x64-ec2	2.06 (x86_64)	2.06 (x86_64)	
grub2-pc-modules	2.06	2.06	
grub2-tools	2.06	2.06	
grub2-tools-minimal	2.06	2.06	
grubby	8,40	8,40	
gssproxy	0.8.4		
gzip	1.12	1.12	
hostname	3.23	3.23	
hunspell	1.7.0		
hunspell-en	0.20140811.1		
hunspell-en-GB	0.20140811.1		
hunspell-en-US	0.20140811.1		
hunspell-filesystem	1.7.0		

Pacote	AMI	AMI mínima	Contêiner
hwdata	0,384	0,384	
info	6.7		
inih	49	49	
initscripts	10.09	10.09	
iproute	6.10.0	6.10.0	
iputils	20210202	20210202	
irqbalance	1.9.0	1.9.0	
jansson	2.14	2.14	
jemalloc	5.2.1		
jitterentropy	3.4.1	3.4.1	
jq	1.7.1	1.7.1	
json-c	0,14	0,14	0,14
kbd	2.4.0	2.4.0	
kbd-misc	2.4.0	2.4.0	
kernel	6.1.112	6.1.112	
kernel-libbpf	6.1.112	6.1.112	
kernel- livepatch-repo- s3	2023.6.20241031	2023.6.20241031	
kernel-srpm- macros	1,0		

Pacote	AMI	AMI mínima	Contêiner
kernel-tools	6.1.112		
keyutils	1.6.3		
keyutils-libs	1.6.3	1.6.3	1.6.3
kmod	29	29	
kmod-libs	29	29	
kpatch-runtime	0.9.7		
krb5-libs	1.21.3	1.21.3	1.21.3
less	608	608	
libacl	2.3.1	2.3.1	2.3.1
libaio	0.3.111		
libarchive	3.7.4	3.7.4	3.7.4
libargon2	27 de dezembro de 2017	27 de dezembro de 2017	
libassuan	2.5.5	2.5.5	2.5.5
libattr	2.5.1	2.5.1	2.5.1
libbasicobjects	0.1.1		
libblkid	2.37.4	2.37.4	2.37.4
libcap	2,48	2,48	2,48
libcap-ng	0.8.2	0.8.2	0.8.2
libcbor	0.7.0	0.7.0	
libcollection	0.7.0		

Pacote	AMI	AMI mínima	Contêiner
libcom_err	1.46,5	1.46,5	1.46,5
libcomps	0.1.20	0.1.20	0.1.20
libconfig	1.7.2		
libcurl-minimal	8.5.0	8.5.0	8.5.0
libdb	5.3.28	5.3.28	
libdhash	0.5.0		
libdnf	0.69,0	0.69,0	0.69,0
libeconf	0.4.0	0.4.0	
libedit	3.1	3.1	
libev	4,33		
libevent	2.1.12		
libfdisk	2.37.4	2.37.4	
libffi	3.4.4	3.4.4	3.4.4
libfido2	1.10.0	1.10.0	
libgcc	11.4.1	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2	1.10.2
libgomp	11.4.1	11.4.1	11.4.1
libgpg-error	1,42	1,42	1,42
libibverbs	48,0		
libidn2	2.3.2	2.3.2	2.3.2

Pacote	AMI	AMI mínima	Contêiner
libini_config	1.3.1		
libkcapi	1.4.0	1.4.0	
libkcapi-hmaccalc	1.4.0	1.4.0	
libldb	2.6.2		
libmaxminddb	1.5.2		
libmetalink	0.1.3		
libmnl	1.0.4	1.0.4	
libmodulemd	2.13.0	2.13.0	2.13.0
libmount	2.37.4	2.37.4	2.37.4
libnfsidmap	2.5.4		
libnghttp2	1.59.0	1.59.0	1.59.0
libnl3	3.5.0		
libpath_utils	0.2.1		
libpcap	1.10.1		
libpipeline	1.5.3	1.5.3	
libpkgconf	1.8.0		
libpsl	0,21,1	0,21,1	0,21,1
libpwquality	1.4.4	1.4.4	
libref_array	0.1.5		
librepo	1.14.5	1.14.5	1.14.5

Pacote	AMI	AMI mínima	Contêiner
libreport-filesystem	2.15.2	2.15.2	2.15.2
libseccomp	2.5.3	2.5.3	
libselinux	3.4	3.4	3.4
libselinux-utils	3.4	3.4	
libsemanage	3.4	3.4	
libsepol	3.4	3.4	3.4
libsigsegv	2.13	2.13	2.13
libsmartcols	2.37.4	2.37.4	2.37.4
libsolv	0.7.22	0.7.22	0.7.22
libss	1.46,5	1.46,5	
libsss_certmap	2.9.4		
libsss_idmap	2.9.4		
libsss_nss_idmap	2.9.4		
libsss_sudo	2.9.4		
libstdc++	11.4.1	11.4.1	11.4.1
libstoragemgmt	1.9.4		
libtalloc	2.3.4		
libtasn1	4.19.0	4.19.0	4.19.0

Pacote	AMI	AMI mínima	Contêiner
libtdb	1.4.7		
libtevent	0.13.0		
libtextstyle	0,21	0,21	
libtirpc	1.3.3		
libunistring	0.9.10	0.9.10	0.9.10
libuser	0,63	0,63	
libutempter	1.2.1	1.2.1	
libuuid	2.37.4	2.37.4	2.37.4
libuv	1.47.0		
libverto	0.3.2	0.3.2	0.3.2
libverto-libev	0.3.2		
libxcrypt	4.4.3	4.4.3	4.4.3
libxml2	2.10.4	2.10.4	2.10.4
libyaml	0.2.5	0.2.5	0.2.5
libzstd	1.5.5	1.5.5	1.5.5
linux-firmware-whence	20210208 (noarch)	20210208 (noarch)	
lm_sensors-libs	3.6.0		
lmdb-libs	0.9.29		
logrotate	3.20.1	3.20.1	
lsf	4.94.0		

Pacote	AMI	AMI mínima	Contêiner
lua-libs	5.4.4	5.4.4	5.4.4
lua-srpm-macros	1		
lz4-libs	1.9.4	1.9.4	1.9.4
man-db	2.9.3	2.9.3	
man-pages	5.10		
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)	
mpfr	4.1.0	4.1.0	4.1.0
nano	5,8		
ncurses	6.2	6.2	
ncurses-base	6.2	6.2	6.2
ncurses-libs	6.2	6.2	6.2
nettle	3.8	3.8	
net-tools	2,0	2.0	
newt	0,52,21		
nfs-utils	2.5.4		
npth	1,6	1,6	1.6
nspr	4.35.0		
nss	3.90,0		
nss-softokn	3.90,0		
nss-softokn-freebl	3.90,0		

Pacote	AMI	AMI mínima	Contêiner
nss-sysinit	3.90,0		
nss-util	3.90,0		
ntsysv	1.15		
numactl-libs	2.0.14	2.0.14	
ocaml-srpm-macros	6		
oniguruma	6.9.7.1	6.9.7.1	
openblas-srpm-macros	2		
openldap	2.4.57	2.4.57	
openssh	8.7p1	8.7p1	
openssh-clients	8.7p1	8.7p1	
openssh-server	8.7p1	8.7p1	
openssl	3.0.8	3.0.8	
openssl-libs	3.0.8	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12	
os-prober	1,7	1,7	
p11-kit	0.24.1	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1	0.24.1
package-notes-srpm-macros	0.4		

Pacote	AMI	AMI mínima	Contêiner
pam	1.5.1	1.5.1	
parted	3.4		
passwd	0,80	0,80	
pciutils	3.7.0	3.7.0	
pciutils-libs	3.7.0	3.7.0	
pcre2	10h40	10h40	10h40
pcre2-syntax	10h40	10h40	10h40
perl-Carp	1,50		
perl-Class-Struct	0,66		
perl-constant	1,33		
perl-DynaLoader	1,47		
perl-Encode	3,15		
perl-Errno	1,30		
perl-Exporter	5,74		
perl-Fcntl	1.13		
perl-File-Basename	2,85		
perl-File-Path	2,18		
perl-File-stat	1,09		
perl-File-Temp	0.231.100		

Pacote	AMI	AMI mínima	Contêiner
perl-Getopt-Long	2,52		
perl-Getopt-Std	1.12		
perl-HTTP-Tiny	0,078		
perl-if	0.60.800		
perl-interpreter	5.32.1		
perl-IO	1,43		
perl-IPC-Open3	1,21		
perl-libs	5.32.1		
perl-MIME-Base64	3,16		
perl-mro	1,23		
perl-overload	1,31		
perl-overloading	0,02		
perl-parent	0,238		
perl-PathTools	3,78		
perl-Pod-Escapes	1,07		
perl-podlators	4.14		
perl-Pod-Perldoc	3.28.01		

Pacote	AMI	AMI mínima	Contêiner
perl-Pod-Simple	3,42		
perl-Pod-Usage	2.01		
perl-POSIX	1,94		
perl-Scalar-List-Utils	1,56		
perl-SelectSaver	1.02		
perl-Socket	2.032		
perl-srpm-macros	1		
perl-Storable	3.21		
perl-subs	1,03		
perl-Symbol	1,08		
perl-Term-ANSIColor	5.01		
perl-Term-Cap	1.17		
perl-Text-ParseWords	3,30		
perl-Text-Tabs+Wrap	2021.07.26		
perl-Time-Local	1.300		
perl-vars	1,05		
pkgconf	1.8.0		

Pacote	AMI	AMI mínima	Contêiner
pkgconf-m4	1.8.0		
pkgconf-pkg-config	1.8.0		
polycoreutils	3.4	3.4	
polycoreutils-python-utils	3.4		
popt	1,18	1,18	1,18
procps-ng	3.3.17	3.3.17	
protobuf-c	1.4.1		
psacct	6.6.4		
psmisc	23,4	23,4	
publicsuffix-list-dafsa	20240212	20240212	20240212
python3	3.9.16	3.9.16	3.9.16
python3-attrs	20.3.0	20.3.0	
python3-audit	3.0.6	3.0.6	
python3-awscli	0.19.19	0.19.19	
python3-babel	2.9.1	2.9.1	
python3-cffi	1.14.5	1.14.5	
python3-chardet	4.0.0	4.0.0	

Pacote	AMI	AMI mínima	Contêiner
python3-c olorama	0.4.4	0.4.4	
python3-c onfigobj	5.0.6	5.0.6	
python3-c ryptography	36.0.1	36.0.1	
python3-daemon	2.3.0		
python3-d ateutil	2.8.1	2.8.1	
python3-dbus	1.2.18	1.2.18	
python3-distro	1.5.0	1.5.0	
python3-dnf	4.14.0	4.14.0	4.14.0
python3-dnf- plugins-core	4.3.0	4.3.0	
python3-d ocutils	0,16	0,16	
python3-gpg	1.15.1	1.15.1	1.15.1
python3-hawkey	0.69,0	0.69,0	0.69,0
python3-idna	(2.10)	(2.10)	
python3-jinja2	2.11.3	2.11.3	
python3-j mespath	0.10.0	0.10.0	

Pacote	AMI	AMI mínima	Contêiner
python3-j sonpatch	1,21	1,21	
python3-j sonpointer	2,0	2.0	
python3-j sonschema	3.2.0	3.2.0	
python3-l ibcomps	0.1.20	0.1.20	0.1.20
python3-libdnf	0.69,0	0.69,0	0.69,0
python3-libs	3.9.16	3.9.16	3.9.16
python3-l ibselinux	3.4	3.4	
python3-l ibsemanage	3.4	3.4	
python3-l ibstoragemgmt	1.9.4		
python3-l ockfile	0.12.2		
python3-m arkupsafe	1.1.1	1.1.1	
python3-n etifaces	0.10.6	0.10.6	
python3-o authlib	3.0.2	3.0.2	

Pacote	AMI	AMI mínima	Contêiner
python3-pip-wheel	21.3.1	21.3.1	21.3.1
python3-ply	3.11	3.11	
python3-policycoreutils	3.4	3.4	
python3-prettytable	0.7.2	0.7.2	
python3-prompt-toolkit	3.0.24	3.0.24	
python3-pyparser	2.20	2.20	
python3-pyrsistent	0.17.3	0.17.3	
python3-pyserial	3.4	3.4	
python3-pysocks	1.7.1	1.7.1	
python3-pytz	2022.7.1	2022.7.1	
python3-pyyaml	5.4.1	5.4.1	
python3-requests	2.25.1	2.25.1	
python3-rpm	4.16.1.3	4.16.1.3	4.16.1.3
python3-ruamel-yaml	0.16.6	0.16.6	

Pacote	AMI	AMI mínima	Contêiner
python3-ruamel-yaml-clib	0.1.2	0.1.2	
python3-setools	4.4.1	4.4.1	
python3-setuptools	59.6.0	59.6.0	
python3-setuptools-wheel	59.6.0	59.6.0	59.6.0
python3-six	1.15.0	1.15.0	
python3-systemd	235	235	
python3-urllib3	1.25.10	1.25.10	
python3-wcwidth	0.2.5	0.2.5	
python-chevron	0.13.1		
python-srpm-macros	3.9		
quota	4.06		
quota-nls	4.06		
readline	8.1	8.1	8.1
rng-tools	6.14	6.14	
rootfiles	8.1	8.1	
rpcbind	1.2.6		
rpm	4.16.1.3	4.16.1.3	4.16.1.3

Pacote	AMI	AMI mínima	Contêiner
rpm-build-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3	
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3	
rpm-sign-libs	4.16.1.3	4.16.1.3	4.16.1.3
rsync	3.2.6		
rust-srpm-macros	21		
sbsigntools	0.9.4	0.9.4	
screen	4.8.0		
sed	4.8	4.8	4.8
selinux-policy	38.1.45	38.1.45	
selinux-policy-targeted	38.1.45	38.1.45	
setup	2.13.7	2.13.7	2.13.7
shadow-utils	4,9	4,9	
slang	2.3.2		
sqlite-libs	3.40,0	3.40,0	3.40,0
sssd-client	2.9.4		
sssd-common	2.9.4		

Pacote	AMI	AMI mínima	Contêiner
sssd-kcm	2.9.4		
sssd-nfs-idmap	2.9.4		
strace	6.8		
sudo	1.9.15	1.9.15	
sysctl-defaults	1.0	1,0	
sysstat	12.5.6		
systemd	252,23	252,23	
systemd-libs	252,23	252,23	
systemd-networkd	252,23	252,23	
systemd-pam	252,23	252,23	
systemd-resolved	252,23	252,23	
systemd-udev	252,23	252,23	
system-release	2023.6.20241031	2023.6.20241031	2023.6.20241031
systemtap-runtime	4.8		
tar	1,34	1,34	
tbb	2020.3		
tcpdump	4.99.1		
tcsh	24.6.07		

Pacote	AMI	AMI mínima	Contêiner
time	1.9		
traceroute	2.1.3		
tzdata	2024a	2024a	2024a
unzip	6.0		
update-motd	2.2	2.2	
userspace-rcu	0.12.1	0.12.1	
util-linux	2.37.4	2.37.4	
util-linux-core	2.37.4	2.37.4	
vim-common	9.0.2153		
vim-data	9.0.2153	9.0.2153	
vim-enhanced	9.0.2153		
vim-filessystem	9.0.2153		
vim-minimal	9.0.2153	9.0.2153	
wget	1.21.3		
which	2.21	2.21	
words	3.0		
xfsdump	3.1.11		
xfspgrog	5.18.0	5.18.0	
xxd	9.0.2153		
xxhash-libs	0.8.0		

Pacote	AMI	AMI mínima	Contêiner
xz	5.2.5	5.2.5	
xz-libs	5.2.5	5.2.5	5.2.5
yum	4.14.0	4.14.0	4.14.0
zip	3.0		
zlib	1.2.11	1.2.11	1.2.11
zram-generator	1.1.2	1.1.2	
zram-generator-defaults	1.1.2	1.1.2	
zstd	1.5.5	1.5.5	

Uso AL2023 em contêineres

Note

Para obter mais informações sobre como usar para AL2023 hospedar cargas de trabalho em contêineres no Amazon ECS, consulte. [AL2023 para hosts de contêineres do Amazon ECS](#)

Há várias maneiras que AL2023 podem ser usadas dentro de contêineres, dependendo do caso de uso. [AL2023 Imagem do contêiner base](#) É mais semelhante a uma imagem de contêiner do Amazon Linux 2 e à AMI AL2023 mínima.

Para usuários avançados, oferecemos uma imagem mínima de contêiner, introduzida na versão AL2023 2.2, junto com a documentação que descreve como criar contêineres [básicos](#).

AL2023 também pode ser usado para hospedar cargas de trabalho em contêineres, com base em imagens de contêiner ou AL2023 em contêineres baseados em outras distribuições Linux. Você pode usar [AL2023 para hosts de contêineres do Amazon ECS](#) ou usar diretamente os pacotes de tempo de execução do contêiner fornecidos. Os `nerdctl` pacotes `dockercontainerd`, e estão disponíveis para instalação e uso AL2023.

Tópicos

- [Usando a imagem do contêiner AL2023 base](#)
- [AL2023 Imagem mínima do contêiner](#)
- [Criando imagens básicas de contêineres AL2023](#)
- [Comparar pacotes instalados em imagens de contêiner do Amazon Linux 2023](#)
- [Comparar pacotes instalados em imagens de contêiner do Amazon Linux 2023](#)

Usando a imagem do contêiner AL2023 base

A imagem do AL2023 contêiner é criada a partir dos mesmos componentes de software incluídos na AL2023 AMI. Está disponível para uso em qualquer ambiente como imagem base para workloads do Docker. Se estiver usando o Amazon Linux AMI para aplicativos no [Amazon Elastic Compute Cloud](#) (Amazon EC2), você poderá colocar seus aplicativos em contêineres com a imagem de contêiner do Amazon Linux.

Use a imagem do contêiner Amazon Linux em seu ambiente de desenvolvimento local e, em seguida, envie seu aplicativo para AWS usar o [Amazon Elastic Container Service](#) (Amazon ECS). Para obter mais informações, consulte [Usar imagens do Amazon ECR com o Amazon ECS](#) no Guia do usuário do Amazon Elastic Container Registry.

A imagem de contêiner do Amazon Linux está disponível no Amazon ECR Public. Você pode fornecer feedback por AL2023 meio de seu AWS representante designado ou registrando um problema no repositório [amazon-linux-2023](#) em. GitHub

Para extrair a imagem de contêiner do Amazon Linux do Amazon ECR Public

1. Autentique o cliente do Docker para seu registro do Amazon Linux Public. Os tokens de autenticação são válidos por 12 horas. Para obter mais informações, consulte [Autenticação de registro privado](#) no Guia do Usuário do Registro de Contêineres da Amazon Elastic.

Note

O get-login-password comando é suportado usando a versão mais recente da AWS CLI versão 2. Para obter mais informações, consulte [Instalar a AWS Command Line Interface](#) no Guia do usuário da AWS Command Line Interface .

```
$ aws ecr-public get-login-password --region us-east-1 | docker login --username  
AWS --password-stdin public.ecr.aws
```

A saída é a seguinte:

```
Login succeeded
```

2. Extraia a imagem do contêiner do Amazon Linux usando o comando `docker pull`. Para visualizar a imagem do contêiner do Amazon Linux na Galeria Pública do Amazon ECR, consulte [Galeria pública do Amazon ECR - amazonlinux](#).

Note

Ao extrair a imagem do AL2023 Docker contêiner, você pode usar as tags em um dos seguintes formatos:

- Para obter a versão mais recente da imagem do AL2023 contêiner, use a `:2023` tag.
- Para obter uma versão específica do AL2023, você pode usar o seguinte formato:
 - `:2023.[0-7 release quarter].[release date].[build number]`

Os exemplos a seguir usam a tag `:2023` e extraem a imagem de contêiner mais recente disponível de AL2023.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023
```

3. (Opcional) Execute o contêiner localmente.

```
$ docker run -it --security-opt seccomp=unconfined public.ecr.aws/amazonlinux/  
amazonlinux:2023 /bin/bash
```

Para extrair a imagem do AL2023 contêiner do Docker Hub

1. Puxe a imagem do AL2023 contêiner usando o `docker pull` comando.

```
$ docker pull amazonlinux:2023
```

2. (Opcional) Execute o contêiner localmente.

```
$ docker run -it amazonlinux:2023 /bin/bash
```

Note

A imagem do contêiner AL2023 usa somente o gerenciador de `dnf` pacotes para instalar pacotes de software. Isso significa que não há nenhum comando `amazon-linux-extras` ou um comando equivalente a ser usado para software adicional.

AL2023 Imagem mínima do contêiner

Note

As imagens de AL2023 contêiner padrão são adequadas para a maioria dos casos de uso, e a adaptação à imagem mínima do contêiner provavelmente será mais trabalhosa do que a adaptação à imagem AL2023 básica do contêiner.

A imagem AL2023 mínima do contêiner, introduzida em AL2023 .2, difere da imagem básica do contêiner porque contém somente os pacotes mínimos necessários para instalar outros pacotes. A imagem mínima de contêiner foi projetada para ser um conjunto mínimo de pacotes, não um conjunto conveniente de pacotes.

A imagem AL2023 mínima do contêiner é criada a partir de componentes de software já disponíveis no AL2023. A principal diferença na imagem mínima de contêiner é o uso de `microdnf` para fornecer o gerenciador de pacotes `dnf`, em vez do `dnf` completo baseado em Python. Isso permite que a imagem mínima do contêiner seja menor, com a desvantagem de não ter o conjunto completo de recursos do gerenciador de `dnf` pacotes que está incluído na imagem básica do AL2023 AMIs contêiner.

A imagem AL2023 mínima do contêiner forma a base do ambiente de execução do `provided.al2023` AWS Lambda.

Para obter uma lista detalhada dos pacotes incluídos na imagem mínima de contêiner, consulte [Comparar pacotes instalados em imagens de contêiner do Amazon Linux 2023](#).

Tamanho mínimo da imagem do contêiner

Como a imagem AL2023 mínima do contêiner contém menos pacotes do que a imagem AL2023 básica do contêiner, ela também é significativamente menor. A tabela a seguir compara as opções de imagem de contêiner das versões atuais e anteriores do Amazon Linux.

Note

O tamanho da imagem é mostrado no [Amazon Linux na Galeria Pública do Amazon ECR](#).

Imagem	Versão	Tamanho da imagem	Observação
Amazon Linux (1AL1)	2018.03.0.20230918.0	62,3 MB	Somente x86-64
Amazon Linux 2	2.0.20230926.0	64,2 MB	aaarch64 é 1,6 MB maior que x86-64
Imagem de contêiner base do Amazon Linux 2023	2023.2.20231002.0	52,4 MB	
Imagem de contêiner mínimo do Amazon Linux	2023.2.20231002.0-minimal	35,2 MB	

Usando a imagem AL2023 Minimal Container

A imagem AL2023 mínima do contêiner está disponível em ECR e a `2023-minimal` tag sempre apontará para a imagem de contêiner mínima AL2023 baseada mais recente, enquanto a `minimal` tag pode ser atualizada para uma versão mais recente do Amazon Linux do que AL2023.

Você pode extrair essas tags usando `docker` com o seguinte exemplo:

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:minimal
```

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023-minimal
```

O exemplo a seguir mostra um Dockerfile que extrai a imagem mínima de contêiner e instala o GCC sobre ela:

```
FROM public.ecr.aws/amazonlinux/amazonlinux:2023-minimal
RUN dnf install -y gcc && dnf clean all
```

Criando imagens básicas de contêineres AL2023

A imagem de contêiner do AL2023 é criada com base nos mesmos componentes de software que são incluídos na AMI do AL2023. Essa tag inclui um software que possibilita que a camada de base do contêiner se comporte de forma semelhante à execução em uma instância do Amazon EC2, como o gerenciador de pacotes dnf. Esta seção explica como você pode construir um contêiner do zero que inclua apenas as dependências mínimas necessárias para uma aplicação.

Note

As imagens de AL2023 contêiner padrão são adequadas para a maioria dos casos de uso. O uso da imagem de contêiner padrão facilita a criação em cima da imagem. Uma imagem de contêiner básico dificulta a criação sobre a sua imagem.

Para criar um contêiner com dependências mínimas para um aplicativo

1. Determine suas dependências de tempo de execução. Isso variará de acordo com sua inscrição.
2. Construa um Dockerfile / Containerfile que constrói FROM scratch. O exemplo a seguir de Dockerfile pode ser usado para criar um contêiner que contém somente bash shell e suas dependências.

```
FROM public.ecr.aws/amazonlinux/amazonlinux:2023 as build
RUN mkdir /sysroot
RUN dnf --releasever=$(rpm -q system-release --qf '%{VERSION}') \
```

```
--installroot /sysroot \  
-y \  
--setopt=install_weak_deps=False \  
install bash  
  
FROM scratch  
COPY --from=build /sysroot /  
WORKDIR /  
ENTRYPOINT ["/bin/bash"]
```

- Este Dockerfile funciona ao:

1. Iniciando um AL2023 contêiner chamado build. Esse contêiner será usado para inicializar o contêiner básico. Esse contêiner não é implantado sozinho, mas gera o contêiner a ser implantado.
2. Criar o diretório /sysroot. Esse diretório será onde o contêiner build instalará as dependências necessárias para o contêiner de barebones. Em uma etapa subsequente, o /sysroot caminho será empacotado para ser o diretório raiz de nossa imagem básica.

Usar a `--installroot` opção de dessa dnf maneira é como criamos as outras AL2023 imagens. Trata-se de um recurso de dnf que permite que instaladores e ferramentas de criação de imagens funcionem.

3. Invocar dnf para instalar pacotes em /sysroot.

O comando `rpm -q system-release --qf '%{VERSION}'` consulta (-q) o pacote `%{VERSION}`, definindo o formato da consulta (-qf) para imprimir a versão do pacote que está sendo consultado (a variável `system-release` é a variável rpm da versão do RPM).

Ao definir o argumento `--releasever` de dnf para a versão de `system-release` no contêiner build, o Dockerfile pode ser usado para reconstruir o contêiner básico sempre que uma imagem base de contêiner atualizada do Amazon Linux for lançada.

É possível definir o `--releasever` para qualquer versão do Amazon Linux 2023, como 2023.10.20260216. Isso significaria que o build contêiner seria executado como a AL2023 versão mais recente, mas construiria o contêiner básico a partir de 2023.10.20260216, independentemente da versão atual. AL2023

A opção `--setopt=install_weak_deps=False` de configuração diz `dnf` para instalar somente as dependências necessárias, em vez de recomendadas ou sugeridas.

4. Copiar o sistema instalado na raiz de um contêiner vazio (FROM scratch).
5. Definindo `ENTRYPOINT` o como o binário desejado, neste caso `/bin/bash`.
3. Crie um diretório vazio e adicione o conteúdo do exemplo na Etapa 2 a um arquivo chamado `Dockerfile`.

```
$ mkdir al2023-barebones-bash-example
$ cd al2023-barebones-bash-example
$ cat > Dockerfile <<EOF
FROM public.ecr.aws/amazonlinux/amazonlinux:2023 as build
RUN mkdir /sysroot
RUN dnf --releasever=$(rpm -q system-release --qf '%{VERSION}') \
  --installroot /sysroot \
  -y \
  --setopt=install_weak_deps=False \
  install bash && dnf --installroot /sysroot clean all

FROM scratch
COPY --from=build /sysroot /
WORKDIR /
ENTRYPOINT ["/bin/bash"]
EOF
```

4. Crie o contêiner executando o comando a seguir.

```
$ docker build -t al2023-barebones-bash-example
```

5. Execute o contêiner usando o comando a seguir para ver o quão mínimo é um contêiner de somente `bash`.

```
$ docker run -it --rm al2023-barebones-bash-example
bash-5.2# rpm
bash: rpm: command not found
bash-5.2# du -sh /usr/
bash: du: command not found
bash-5.2# ls
bash: ls: command not found
bash-5.2# echo /bin/*
```

```
/bin/alias /bin/bash /bin/bashbug /bin/bashbug-64 /bin/bg /bin/catchsegv /bin/cd /
bin/command /bin/fc /bin/fg /bin/gencat /bin/getconf /bin/getent /bin/getopts /
bin/hash /bin/iconv /bin/jobs /bin/ld.so /bin/ldd /bin/locale /bin/localedef /
bin/pldd /bin/read /bin/sh /bin/sotrust /bin/sprof /bin/type /bin/tzselect /bin/
ulimit /bin/umask /bin/unalias /bin/wait /bin/zdump
```

Para um exemplo mais prático, o procedimento a seguir cria um contêiner para um aplicativo C que exibe Hello World!.

1. Crie um diretório vazio e adicione o código-fonte C e Dockerfile.

```
$ mkdir al2023-barebones-c-hello-world-example
$ cd al2023-barebones-c-hello-world-example
$ cat > hello-world.c <<EOF
#include <stdio.h>
int main(void)
{
    printf("Hello World!\n");
    return 0;
}
EOF

$ cat > Dockerfile <<EOF
FROM public.ecr.aws/amazonlinux/amazonlinux:2023 as build
COPY hello-world.c /
RUN dnf -y install gcc
RUN gcc -o hello-world hello-world.c
RUN mkdir /sysroot
RUN mv hello-world /sysroot/
RUN dnf --releasever=$(rpm -q system-release --qf '%{VERSION}') \
    --installroot /sysroot \
    -y \
    --setopt=install_weak_deps=False \
    install glibc && dnf --installroot /sysroot clean all

FROM scratch
COPY --from=build /sysroot /
WORKDIR /
ENTRYPOINT ["/hello-world"]
EOF
```


2. Execute o contêiner usando o seguinte comando.

```
$ docker build -t al2023-barebones-c-hello-world-example .
```

3. Execute o contêiner usando o seguinte comando.

```
$ docker run -it --rm al2023-barebones-c-hello-world-example  
Hello World!
```

Comparar pacotes instalados em imagens de contêiner do Amazon Linux 2023

Uma comparação do RPMs presente na imagem do contêiner AL2023 base em comparação com o RPMs presente na imagem AL2023 mínima do contêiner.

Pacote	Contêiner	Contêiner mínimo
alternatives	1.15	1.15
amazon-linux-repo-cdn	2023.6.20241031	2023.6.20241031
audit-libs	3.0.6	3.0.6
basesystem	11	11
bash	5.2.15	5.2.15
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023.2.68	2023.2.68
coreutils-single	8,32	8,32
crypto-policies	2020428	2020428
curl-minimal	8.5.0	8.5.0

Pacote	Contêiner	Contêiner mínimo
dnf	4.14.0	
dnf-data	4.14.0	4.14.0
elfutils-default-yama-scope	0.188	
elfutils-libelf	0.188	
elfutils-libs	0.188	
expat	2.5.0	
file-libs	5,39	5,39
filesystem	3,14	3,14
gawk	5.1.0	5.1.0
gdbm-libs	1,19	
glib2	2.74.7	2.74.7
glibc	2.34	2.34
glibc-common	2.34	2.34
glibc-minimal-langpack	2.34	2.34
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gobject-introspection		1.73.0
gpgme	1.15.1	1.15.1

Pacote	Contêiner	Contêiner mínimo
grep	3.8	3.8
json-c	0,14	0,14
keyutils-libs	1.6.3	1.6.3
krb5-libs	1.21.3	1.21.3
libacl	2.3.1	2.3.1
libarchive	3.7.4	3.7.4
libassuan	2.5.5	2.5.5
libattr	2.5.1	2.5.1
libblkid	2.37.4	2.37.4
libcap	2,48	2,48
libcap-ng	0.8.2	0.8.2
libcom_err	1.46,5	1.46,5
libcomps	0.1.20	
<u>libcurl-minimal</u>	8.5.0	8.5.0
libdnf	0.69,0	0.69,0
libffi	3.4.4	3.4.4
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	
libgpg-error	1,42	1,42

Pacote	Contêiner	Contêiner mínimo
libidn2	2.3.2	2.3.2
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libnghttp2	1.59.0	1.59.0
libpeas		1.32.0
libpsl	0,21,1	0,21,1
librepo	1.14.5	1.14.5
libreport-filessystem	2.15.2	2.15.2
libselinux	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2.13	2.13
libsmartcols	2.37.4	2.37.4
libsolv	0.7.22	0.7.22
libstdc++	11.4.1	11.4.1
libtasn1	4.19.0	4.19.0
libunistring	0.9.10	0.9.10
libuuid	2.37.4	2.37.4
libverto	0.3.2	0.3.2
libxcrypt	4.4.3	
libxml2	2.10.4	2.10.4

Pacote	Contêiner	Contêiner mínimo
libyam1	0.2.5	0.2.5
libzstd	1.5.5	1.5.5
lua-libs	5.4.4	5.4.4
lz4-libs	1.9.4	1.9.4
microdnf		3.10.0
microdnf-dnf		3.10.0
mpfr	4.1.0	4.1.0
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
npth	1,6	1.6
openssl-libs	3.0.8	3.0.8
p11-kit	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1
pcre2	10h40	10h40
pcre2-syntax	10h40	10h40
popt	1,18	1,18
publicsuffix-list-dafsa	2024/02/12	2024/02/12
python3	3.9.16	
python3-dnf	4.14.0	
python3-gpg	1.15.1	

Pacote	Contêiner	Contêiner mínimo
python3-hawkey	0.69,0	
python3-libcomps	0.1.20	
python3-libdnf	0.69,0	
python3-libs	3.9.16	
python3-pip-wheel	21.3.1	
python3-rpm	4.16.1.3	
python3-setuptools-wheel	59.6.0	
readline	8.1	8.1
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	
rpm-libs	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	
sed	4.8	4.8
setup	2.13.7	2.13.7
sqlite-libs	3.40,0	3.40,0
system-release	2023.6.20241031	2023.6.20241031
tzdata	2024a	
xz-libs	5.2.5	5.2.5
yum	4.14.0	
zlib	1.2.11	1.2.11

Comparar pacotes instalados em imagens de contêiner do Amazon Linux 2023

Uma comparação do RPMs presente na AMI AL2023 minimal com o RPMs presente nas imagens AL2023 básicas e mínimas do contêiner.

Pacote	AMI mínima	Contêiner	Contêiner mínimo
alternatives	1.15	1.15	1.15
amazon-chrondy-config	4.3		
amazon-ec2-net-utils	2.5.1		
amazon-linux-repo-cdn		2023.6.20241031	2023.6.20241031
amazon-linux-repo-s3	2023.6.20241031		
amazon-linux-sb-keys	2023.1		
amd-ucode-firmware	20210208 (noarch)		
audit	3.0.6		
audit-libs	3.0.6	3.0.6	3.0.6
awscli-2	2.15.30		
basesystem	11	11	11
bash	5.2.15	5.2.15	5.2.15
bzip2-libs	1.0.8	1.0.8	1.0.8

Pacote	AMI mínima	Contêiner	Contêiner mínimo
ca-certificates	2023.2.68	2023.2.68	2023.2.68
checkpolicy	3.4		
chrony	4.3		
cloud-init	22.2.2		
cloud-init-cfg-ec2	22.2.2		
cloud-utils-growpart	0,31		
coreutils	8,32		
coreutils-common	8,32		
coreutils-single		8,32	8,32
cpio	2.13		
cracklib	2.9.6		
cracklib-dicts	2.9.6		
crypto-policies	2020428	2020428	2020428
cryptsetup-libs	2.6.1		
curl-minimal	8.5.0	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27		
dbus	1.12.28		
dbus-broker	32		

Pacote	AMI mínima	Contêiner	Contêiner mínimo
dbus-common	1.12.28		
dbus-libs	1.12.28		
device-mapper	1.02.185		
device-mapper-libs	1.02.185		
diffutils	3.8		
dnf	4.14.0	4.14.0	
dnf-data	4.14.0	4.14.0	4.14.0
dnf-plugin-release-notification	1.2		
dnf-plugins-core	4.3.0		
dnf-plugin-support-info	1.2		
dracut	055		
dracut-config-ec2	3.0		
dracut-config-generic	055		
e2fsprogs	1.46,5		
e2fsprogs-libs	1.46,5		
ec2-utils	2.2.0		

Pacote	AMI mínima	Contêiner	Contêiner mínimo
efi-filesystem	5		
efivar	38		
efivar-libs	38		
elfutils- default-yama- scope	0.188	0.188	
elfutils-libelf	0.188	0.188	
elfutils-libs	0.188	0.188	
expat	2.5.0	2.5.0	
file	5,39		
file-libs	5,39	5,39	5,39
filesystem	3,14	3,14	3,14
findutils	4.8.0		
fuse-libs	2.9.9		
gawk	5.1.0	5.1.0	5.1.0
gdbm-libs	1,19	1,19	
gdisk	1.0.8		
gettext	0,21		
gettext-libs	0,21		
glib2	2.74.7	2.74.7	2.74.7
glibc	2.34	2.34	2.34

Pacote	AMI mínima	Contêiner	Contêiner mínimo
glibc-all-langpacks	2.34		
glibc-common	2.34	2.34	2.34
glibc-locale-source	2.34		
glibc-minimal-langpack		2.34	2.34
gmp	6.2.1	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7	2.3.7
gnutls	3.8.0		
gobject-introspection			1.73.0
gpgme	1.15.1	1.15.1	1.15.1
grep	3.8	3.8	3.8
groff-base	1.22.4		
grub2-common	2.06		
grub2-efi-aa64-ec2	2.06 (aarch64)		
grub2-efi-x64-ec2	2.06 (x86_64)		
grub2-pc-modules	2.06		
grub2-tools	2.06		

Pacote	AMI mínima	Contêiner	Contêiner mínimo
grub2-tools-minimal	2.06		
grubby	8,40		
gzip	1.12		
hostname	3.23		
hwdata	0,384		
inih	49		
initscripts	10.09		
iproute	6.10.0		
iputils	20210202		
irqbalance	1.9.0		
jansson	2.14		
jitterentropy	3.4.1		
jq	1.7.1		
json-c	0,14	0,14	0,14
kbd	2.4.0		
kbd-misc	2.4.0		
kernel	6.1.112		
kernel-libbpf	6.1.112		

Pacote	AMI mínima	Contêiner	Contêiner mínimo
kernel-li vepatch-repo- s3	2023.6.20241031		
keyutils-libs	1.6.3	1.6.3	1.6.3
kmod	29		
kmod-libs	29		
krb5-libs	1.21.3	1.21.3	1.21.3
less	608		
libacl	2.3.1	2.3.1	2.3.1
libarchive	3.7.4	3.7.4	3.7.4
libargon2	27 de dezembro de 2017		
libassuan	2.5.5	2.5.5	2.5.5
libattr	2.5.1	2.5.1	2.5.1
libblkid	2.37.4	2.37.4	2.37.4
libcap	2,48	2,48	2,48
libcap-ng	0.8.2	0.8.2	0.8.2
libcbor	0.7.0		
libcom_err	1.46,5	1.46,5	1.46,5
libcomps	0.1.20	0.1.20	
libcurl-minimal	8.5.0	8.5.0	8.5.0

Pacote	AMI mínima	Contêiner	Contêiner mínimo
libdb	5.3.28		
libdnf	0.69,0	0.69,0	0.69,0
libeconf	0.4.0		
libedit	3.1		
libfdisk	2.37.4		
libffi	3.4.4	3.4.4	3.4.4
libfido2	1.10.0		
libgcc	11.4.1	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2	1.10.2
libgomp	11.4.1	11.4.1	
libgpg-error	1,42	1,42	1,42
libidn2	2.3.2	2.3.2	2.3.2
libkcapi	1.4.0		
libkcapi-hmacalc	1.4.0		
libmnl	1.0.4		
libmodulemd	2.13.0	2.13.0	2.13.0
libmount	2.37.4	2.37.4	2.37.4
libnghttp2	1.59.0	1.59.0	1.59.0
libpeas			1.32.0
libpipeline	1.5.3		

Pacote	AMI mínima	Contêiner	Contêiner mínimo
libpsl	0,21,1	0,21,1	0,21,1
libpwquality	1.4.4		
librepo	1.14.5	1.14.5	1.14.5
libreport-filesystem	2.15.2	2.15.2	2.15.2
libseccomp	2.5.3		
libselinux	3.4	3.4	3.4
libselinux-utils	3.4		
libsemanage	3.4		
libsepol	3.4	3.4	3.4
libsigsegv	2.13	2.13	2.13
libsmartcols	2.37.4	2.37.4	2.37.4
libsolv	0.7.22	0.7.22	0.7.22
libss	1.46,5		
libstdc++	11.4.1	11.4.1	11.4.1
libtasn1	4.19.0	4.19.0	4.19.0
libtextstyle	0,21		
libunistring	0.9.10	0.9.10	0.9.10
libuser	0,63		
libutempter	1.2.1		

Pacote	AMI mínima	Contêiner	Contêiner mínimo
libuuid	2.37.4	2.37.4	2.37.4
libverto	0.3.2	0.3.2	0.3.2
libxcrypt	4.4.3	4.4.3	
libxml2	2.10.4	2.10.4	2.10.4
libyaml	0.2.5	0.2.5	0.2.5
libzstd	1.5.5	1.5.5	1.5.5
linux-firmware-whence	20210208 (noarch)		
logrotate	3.20.1		
lua-libs	5.4.4	5.4.4	5.4.4
lz4-libs	1.9.4	1.9.4	1.9.4
man-db	2.9.3		
microcode_ctl	2.1 (x86_64)		
microdnf			3.10.0
microdnf-dnf			3.10.0
mpfr	4.1.0	4.1.0	4.1.0
ncurses	6.2		
ncurses-base	6.2	6.2	6.2
ncurses-libs	6.2	6.2	6.2
nettle	3.8		
net-tools	2.0		

Pacote	AMI mínima	Contêiner	Contêiner mínimo
npth	1,6	1,6	1.6
numactl-libs	2.0.14		
oniguruma	6.9.7.1		
openldap	2.4.57		
openssh	8.7p1		
openssh-clients	8.7p1		
openssh-server	8.7p1		
openssl	3.0.8		
openssl-libs	3.0.8	3.0.8	3.0.8
openssl-pkcs11	0.4.12		
os-prober	1,7		
p11-kit	0.24.1	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1	0.24.1
pam	1.5.1		
passwd	0,80		
pciutils	3.7.0		
pciutils-libs	3.7.0		
pcre2	10,40	10,40	10,40
pcre2-syntax	10,40	10,40	10,40
policycoreutils	3.4		

Pacote	AMI mínima	Contêiner	Contêiner mínimo
popt	1,18	1,18	1,18
procps-ng	3.3.17		
psmisc	23,4		
publicsuffix- list-dafsa	2024/02/12	2024/02/12	2024/02/12
python3	3.9.16	3.9.16	
python3-attrs	20.3.0		
python3-audit	3.0.6		
python3-awscrt	0.19.19		
python3-babel	2.9.1		
python3-cffi	1.14.5		
python3-chardet	4.0.0		
python3-c olorama	0.4.4		
python3-c onfigobj	5.0.6		
python3-c ryptography	36.0.1		
python3-d ateutil	2.8.1		
python3-dbus	1.2.18		
python3-distro	1.5.0		

Pacote	AMI mínima	Contêiner	Contêiner mínimo
python3-dnf	4.14.0	4.14.0	
python3-dnf-plugins-core	4.3.0		
python3-d ocutils	0,16		
python3-gpg	1.15.1	1.15.1	
python3-hawkey	0.69,0	0.69,0	
python3-idna	(2.10)		
python3-jinja2	2.11.3		
python3-j mespath	0.10.0		
python3-j sonpatch	1,21		
python3-j sonpointer	2,0		
python3-j sonschema	3.2.0		
python3-l ibcomps	0.1.20	0.1.20	
python3-libdnf	0.69,0	0.69,0	
python3-libs	3.9.16	3.9.16	
python3-l ibselinux	3.4		

Pacote	AMI mínima	Contêiner	Contêiner mínimo
python3-l ibsemanage	3.4		
python3-m arkupsafe	1.1.1		
python3-n etifaces	0.10.6		
python3-o authlib	3.0.2		
python3-pip- wheel	21.3.1	21.3.1	
python3-ply	3.11		
python3-p olicycoreutils	3.4		
python3-p rettytable	0.7.2		
python3-prompt- toolkit	3.0.24		
python3-p yparser	2.20		
python3-p ysistent	0.17.3		
python3-p yserial	3.4		
python3-pysocks	1.7.1		
python3-pytz	2022.7.1		

Pacote	AMI mínima	Contêiner	Contêiner mínimo
python3-pyyaml	5.4.1		
python3-requests	2.25.1		
python3-rpm	4.16.1.3	4.16.1.3	
python3-ruamel-yaml	0.16.6		
python3-ruamel-yaml-clib	0.1.2		
python3-setuptools	4.4.1		
python3-setuptools	59.6.0		
python3-setuptools-wheel	59.6.0	59.6.0	
python3-six	1.15.0		
python3-systemd	235		
python3-urllib3	1.25.10		
python3-wcwidth	0.2.5		
readline	8.1	8.1	8.1
rng-tools	6.14		
rootfiles	8.1		
rpm	4.16.1.3	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3	

Pacote	AMI mínima	Contêiner	Contêiner mínimo
rpm-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3		
rpm-plugin-systemd-inhibit	4.16.1.3		
rpm-sign-libs	4.16.1.3	4.16.1.3	
sbsigntools	0.9.4		
sed	4.8	4.8	4.8
selinux-policy	38.1.45		
selinux-policy-targeted	38.1.45		
setup	2.13.7	2.13.7	2.13.7
shadow-utils	4,9		
sqlite-libs	3.40,0	3.40,0	3.40,0
sudo	1.9.15		
sysctl-defaults	1,0		
systemd	252,23		
systemd-libs	252,23		
systemd-networkd	252,23		
systemd-pam	252,23		

Pacote	AMI mínima	Contêiner	Contêiner mínimo
systemd-resolved	252,23		
systemd-udev	252,23		
system-release	2023.6.20241031	2023.6.20241031	2023.6.20241031
tar	1,34		
tzdata	2024a	2024a	
update-motd	2.2		
userspace-rcu	0.12.1		
util-linux	2.37.4		
util-linux-core	2.37.4		
vim-data	9.0.2153		
vim-minimal	9.0.2153		
which	2.21		
xfspgrog	5.18.0		
xz	5.2.5		
xz-libs	5.2.5	5.2.5	5.2.5
yum	4.14.0	4.14.0	
zlib	1.2.11	1.2.11	1.2.11
zram-generator	1.1.2		
zram-generator-defaults	1.1.2		

Pacote	AMI mínima	Contêiner	Contêiner mínimo
zstd	1.5.5		

AL2023 no AWS Elastic Beanstalk

AWS Elastic Beanstalk é um serviço para implantar e escalar aplicativos e serviços da web. Carregue seu código e o Elastic Beanstalk lida automaticamente na implantação, desde o provisionamento de capacidade, balanceamento de carga e ajuste de escala automático ao monitoramento da saúde do aplicativo. Para mais informações, consulte [AWS Elastic Beanstalk](#).

Para usar o Elastic Beanstalk, crie uma aplicação, faça upload de uma versão dela na forma de um pacote de origem (por exemplo, arquivo Java .war) no Elastic Beanstalk e forneça algumas informações sobre a aplicação. O Elastic Beanstalk inicia automaticamente um ambiente e cria e configura os recursos da AWS necessários para executar o código. Para obter mais informações, consulte o [Guia do desenvolvedor do AWS Elastic Beanstalk](#).

As plataformas Linux do Elastic Beanstalk usam instâncias do Amazon EC2, e essas instâncias executam o Amazon Linux. A partir de 4 de agosto de 2023, o Elastic Beanstalk oferece as seguintes ramificações de plataforma com base no Amazon Linux 2023: Docker, Tomcat, Java SE, Node.js, PHP e Python. O Elastic Beanstalk está trabalhando para lançar o suporte ao AL2023 para mais plataformas do Elastic Beanstalk.

A lista completa do suporte à plataforma Elastic Beanstalk e das plataformas atuais criadas com base no AL2023 pode ser encontrada na seção de [plataformas Linux do Elastic Beanstalk](#) do Guia do [Desenvolvedor do Elastic Beanstalk](#).

Você pode encontrar as notas de lançamento das novas plataformas do Elastic Beanstalk e versões das plataformas existentes nas [Notas de lançamento do Elastic Beanstalk](#).

Usar o AL2023 no AWS CloudShell

AWS CloudShell O é um shell pré-autenticado baseado em navegador que você pode iniciar diretamente do Console de gerenciamento da AWS. Você pode navegar até o CloudShell a partir do Console de gerenciamento da AWS de maneiras diferentes. Para obter mais informações, consulte [Como começar a usar o AWS CloudShell?](#)

O AWS CloudShell, que atualmente é baseado no Amazon Linux 2, migrará para o AL2023. A migração para o AL2023 começará a ser implementada em todas Regiões da AWS, a partir de 4 de dezembro de 2023. Para obter mais informações sobre a migração do CloudShell para o AL2023, consulte [Migração de AWS CloudShell do Amazon Linux 2 para o Amazon Linux 2023](#).

Usar AMIs do Amazon ECS baseadas no AL2023 para hospedar workloads em contêineres

Note

Para obter mais informações sobre como usar o AL2023 dentro de um contêiner, consulte [AL2023 em contêineres](#).

O Amazon Elastic Container Service (Amazon ECS) é um serviço totalmente gerenciado de orquestração de contêineres ajuda a implantar, gerenciar e dimensionar facilmente aplicações containerizadas. Como um serviço totalmente gerenciado, o Amazon ECS vem com práticas recomendadas operacionais e de configuração da AWS incorporadas. Ele é integrado à AWS e a ferramentas de terceiros, como o Amazon Elastic Container Registry (Amazon ECR) e o Docker. Essa integração torna mais fácil para as equipes se concentrarem na criação das aplicações, não no ambiente. Você pode executar e dimensionar suas workloads de contêiner em Regiões AWS na nuvem, sem a complexidade de gerenciar um ambiente de gerenciamento ou nós.

Você pode hospedar workloads em contêineres no AL2023 usando a AMI do AL2023 otimizada para Amazon ECS. Para obter mais informações, consulte [AMIs do Linux otimizadas para o Amazon ECS](#).

Alterações no AL2023 para o Amazon ECS em comparação com o AL2

Assim como no AL2, o AL2023 fornece os pacotes básicos necessários para execução como uma instância Linux do Amazon ECS. No AL2, os pacotes `containerd`, `docker` e `ecs-init` estavam disponíveis por meio de `amazon-linux-extras`, enquanto o AL2023 inclui esses pacotes nos repositórios principais.

Com o recurso de atualizações determinísticas por meio de repositórios com versionamento, cada AMI do AL2023, por padrão, é bloqueada para uma versão específica do repositório. Isso também vale para a AMI do AL2023 otimizada para Amazon ECS. Todas as atualizações do seu ambiente podem ser cuidadosamente gerenciadas e testadas antes da implantação, além de fornecer uma maneira fácil de voltar ao conteúdo de uma AMI anterior em caso de problema. Para obter mais

informações sobre esse recurso do AL2023, consulte [Atualizações determinísticas por meio de repositórios versionados em AL2023](#).

O AL2023 muda para cgroup v2, em vez de usar a interface cgroup v1 compatível com o AL2. Para obter mais informações, consulte [Hierarquia unificada de grupos de controle \(cgroup v2\)](#).

Note

As versões do AL2023 anteriores à [2023.2.20230920](#) (o primeiro lançamento do AL2023.2) continham um bug em systemd para manipulação de falta de memória (OOM) dentro de um cgroup. Todos os processos no cgroup sempre eram interrompidos, em vez de o OOM-Killer escolher um processo por vez, que é o comportamento esperado. Isso foi uma regressão em comparação com o comportamento do AL2, e foi corrigido no lançamento 2023.2.20230920 do AL2023.

O código para compilar a AMI otimizada para Amazon ECS está disponível no [projeto amazon-ecs-ami no GitHub](#). As [notas de lançamento](#) descrevem o mapeamento entre versões do AL2023 e da AMI do Amazon ECS.

Personalização da AMI otimizada para Amazon ECS baseada no AL2023

Important

Recomendamos que você use a AMI do AL2023 otimizada para Amazon ECS. Para obter mais informações, consulte [AMIs do Linux otimizadas para o Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Você pode usar os mesmos scripts de construção que o Amazon ECS usa para criar AMIs personalizadas. Para obter mais informações, consulte [Script de compilação da AMI do Linux otimizada para o Amazon ECS](#).

Usar o Amazon Elastic File System no AL2023

O Amazon Elastic File System (Amazon EFS) fornece armazenamento de arquivos sem servidor e com elasticidade total para você compartilhar dados de arquivos sem provisionar ou gerenciar a capacidade e o desempenho do armazenamento. O Amazon EFS foi criado para escalar sob

demanda para petabytes sem interromper os aplicativos, podendo aumentar ou diminuir à medida que arquivos são adicionados e removidos. O Amazon EFS possui uma interface de serviços da web que permite criar e configurar sistemas de arquivos de maneira rápida e fácil. O serviço gerencia toda a infraestrutura de armazenamento de arquivos para você, para que você evite a complexidade de implantar, corrigir e manter configurações complexas de sistemas de arquivos.

O Amazon EFS é compatível com o protocolo Network File System versão 4 (NFSv4.1 e NFSv4.0), de forma que os aplicativos e ferramentas usados atualmente funcionam perfeitamente com o Amazon EFS. Várias instâncias computacionais, incluindo o Amazon EC2, Amazon ECS e o AWS Lambda podem acessar um sistema de arquivos Amazon EFS ao mesmo tempo. Portanto, um sistema de arquivos EFS pode fornecer uma fonte de dados comum para workloads e aplicativos em execução em mais de uma instância de computação ou servidor.

Instalando **amazon-efs-utils** no AL2023

O pacote **amazon-efs-utils** está disponível nos repositórios do AL2023 para ser instalado e usado para acessar os sistemas de arquivos do Amazon EFS.

Instale o pacote **amazon-efs-utils** no AL2023

- Instale **amazon-efs-utils** usando o comando a seguir.

```
$ dnf -y install amazon-efs-utils
```

Montar um sistema de arquivos do Amazon EFS no AL2023

Depois de instalar **amazon-efs-utils**, você pode montar um sistema de arquivos do Amazon EFS em sua instância do AL2023.

Montar um sistema de arquivos do Amazon EFS no AL2023

- Para montar usando o ID do sistema de arquivos, use o comando a seguir.

```
sudo mount -t efs file-system-id efs-mount-point/
```

Você também pode montar o sistema de arquivos para que os dados em trânsito sejam criptografados usando TLS ou usando o nome DNS ou o IP de destino de montagem em vez do ID

do sistema de arquivos. Para obter mais informações, consulte [Montagem em instâncias Amazon Linux usando o auxiliar de montagem EFS](#).

Usando o Amazon EMR baseado em AL2023

O Amazon EMR é um serviço da web que facilita o processamento de grandes quantidades de dados de maneira eficiente usando o Apache Hadoop e os serviços oferecidos pela AWS.

AL2023 lançamentos baseados no Amazon EMR

A versão 7.0.0 do Amazon EMR foi a primeira versão baseada em AL2023. Com esta versão, AL2023 é o sistema operacional básico do Amazon EMR, trazendo todas as vantagens do AL2023 Amazon EMR. Para saber mais, consulte [Amazon EMR 7.0.0 release notes](#).

AL2023 baseado no Amazon EMR no EKS

O Amazon EMR no EKS 6.13 foi a primeira versão introduzida AL2023 como opção. Com essa versão, você pode iniciar o Spark AL2023 como sistema operacional, junto com o tempo de execução do Java 17. Para saber mais, consulte [Amazon EMR on EKS 6.13 release notes](#) e [Amazon EMR on EKS release notes](#).

Usando AL2023 em AWS Lambda

Com AWS Lambda, você pode executar código sem provisionar ou gerenciar servidores. Você paga somente pelo tempo de computação consumido, não havendo cobrança quando seu código não está em execução. Você pode executar código para praticamente qualquer tipo de aplicação ou serviço de backend, tudo sem nenhuma administração. Carregar seu código e o Lambda cuidará de tudo que for necessário para executar e escalar seu código com alta disponibilidade.

AL2023 **provided.al2023** tempo de execução gerenciado e imagem de contêiner

O tempo de execução `provided.al2023` básico é baseado na [imagem AL2023 mínima do contêiner](#) e fornece um tempo de execução gerenciado pelo Lambda AL2023 baseado e uma imagem [base do contêiner](#). Como o `provided.al2023` tempo de execução é baseado na imagem AL2023 mínima do contêiner, ele é substancialmente menor, com menos de 40 MB, do que o `provided.al2` tempo de execução, com cerca de 109 MB.

Para saber mais, consulte [Runtimes do Lambda](#) e [Criar uma função do Lambda usando uma imagem de contêiner](#).

AL2023 tempos de execução Lambda baseados em Lambda

Versões futuras de tempos de execução de linguagem gerenciada, como Node.js 20, Python 3.12, Java 21 e .NET 8, são baseadas AL2023 e serão usadas `provided.al2023` como imagem base, conforme descrito no [anúncio de tempos de execução AL2023 baseados](#).

AL2023 funções Lambda baseadas em

- [AL2023 Funções Lambda escritas em Go](#)
- [AL2023 Funções Lambda escritas em Rust](#)

Para obter mais informações, consulte [Cotas do Lambda](#) no AWS Lambda Guia do desenvolvedor do e.

Tutoriais

Os tutoriais a seguir mostram como executar tarefas comuns usando instâncias do Amazon EC2 que executam o Amazon Linux 2023 (AL2023). Para acessar tutoriais em vídeo, consulte os [vídeos instrutivos e laboratórios da AWS](#).

Para obter instruções sobre o AL2, consulte [Tutorials for Amazon EC2 instances running Linux](#) no Guia do usuário do Amazon EC2.

Tutoriais

- [Tutorial: Instalar um servidor LAMP em AL2023](#)
- [Tutorial: Configurar SSL/TLS em AL2023](#)
- [Tutorial: hospede um WordPress blog no AL2023](#)
- [Tutorial: Transição do Redis 6 para o Valkey em 023 AL2](#)
- [Tutorial: Instale o ambiente de trabalho GNOME em AL2023](#)
- [Tutorial: Configurar o servidor TigerVNC em AL2023](#)
- [Usando o Multi-Gen LRU \(MGLRU\) em kernels AL2023](#)

Tutorial: Instalar um servidor LAMP em AL2023

Os procedimentos a seguir ajudam você a instalar um servidor web Apache com suporte a PHP e [MariaDB](#) (uma bifurcação do MySQL desenvolvida pela comunidade) em AL2023 sua instância (às vezes chamada de servidor web LAMP ou pilha LAMP). Você pode usar esse servidor para hospedar um site estático ou para implantar um aplicativo PHP dinâmico que lê e grava informações em um banco de dados.

Important

Esses procedimentos são destinados ao uso com AL2023. Se você estiver tentando configurar um servidor web LAMP em uma distribuição diferente, como Ubuntu ou Red Hat Enterprise Linux, este tutorial não funcionará. Para o Ubuntu, consulte a seguinte documentação da comunidade Ubuntu: [ApacheMySQLPHP](#). Para outras distribuições, consulte a documentação específica.

Tarefas

- [Etapa 1: Preparar o servidor LAMP](#)
- [Etapa 2: Testar o servidor LAMP](#)
- [Etapa 3: Proteger o servidor do banco de dados](#)
- [Etapa 4: Instalação \(opcional\) phpMyAdmin](#)
- [Solução de problemas](#)
- [Tópicos relacionados](#)

Etapa 1: Preparar o servidor LAMP

Pré-requisitos

- Este tutorial pressupõe que você já tenha iniciado uma nova instância usando AL2023, com um nome DNS público que pode ser acessado pela Internet. Para obter mais informações, consulte [AL2023 no Amazon EC2](#). Você também precisa ter configurado o security group para permitir conexões SSH (porta 22), HTTP (porta 80) e HTTPS (porta 443). Para saber mais sobre esses pré-requisitos, consulte [Authorize inbound traffic for your Linux instances](#) no Guia do usuário do Amazon EC2.
- O procedimento a seguir instala a versão mais recente do PHP disponível no AL2023, atualmente 8.1. Se você planeja usar aplicações PHP diferentes daquelas descritas neste tutorial, você deve verificar a compatibilidade com a versão 8.1.

Para preparar o servidor LAMP

1. Conecte-se à sua instância. Para obter mais informações, consulte [Conexão com AL2023 instâncias](#).
2. Para garantir que todos os pacotes de software estejam atualizados, execute uma atualização rápida de software em sua instância. Esse processo poderá levar alguns minutos, mas é importante ter certeza de que você tem as atualizações de segurança e correções de bug mais recentes.

A opção `-y` instala as atualizações sem solicitar confirmação. Para examinar as atualizações antes da instalação, você pode omitir essa opção.

```
[ec2-user ~]$ sudo dnf upgrade -y
```

3. Instale as versões mais recentes do servidor web Apache e dos pacotes PHP para AL2023.

```
[ec2-user ~]$ sudo dnf install -y httpd wget php-fpm php-mysql php-json php php-devel
```

4. Instale os pacotes de software do MariaDB. Use o comando `dnf install` para instalar os vários pacotes de software e todas as dependências relacionadas ao mesmo tempo.

```
[ec2-user ~]$ sudo dnf install mariadb105-server
```

Você pode visualizar as versões atuais desses pacotes usando o comando a seguir:

```
[ec2-user ~]$ sudo dnf info package_name
```

Exemplo:

```
[root@ip-172-31-25-170 ec2-user]# dnf info mariadb105
Last metadata expiration check: 0:00:16 ago on Tue Feb 14 21:35:13 2023.
Installed Packages
Name           : mariadb105
Epoch         : 3
Version        : 10.5.16
Release        : 1.amzn2023.0.6
Architecture   : x86_64
Size           : 18 M
Source         : mariadb105-10.5.16-1.amzn2023.0.6.src.rpm
Repository     : @System
From repo      : amazonlinux
Summary        : A very fast and robust SQL database server
URL            : http://mariadb.org
License        : GPLv2 and LGPLv2
Description    : MariaDB is a community developed fork from MySQL - a multi-user,
                  multi-threaded
                  : SQL database server. It is a client/server implementation consisting
of
                  : a server daemon (mariadb) and many different client programs and
libraries.
                  : The base package contains the standard MariaDB/MySQL client programs
and
                  : utilities.
```

5. Inicie o servidor web Apache.


```
[ec2-user ~]$ sudo systemctl start httpd
```

6. Use o comando `systemctl` para configurar o servidor web Apache para iniciar em cada inicialização do sistema.

```
[ec2-user ~]$ sudo systemctl enable httpd
```

Você pode verificar se `httpd` está ativo executando o seguinte comando:

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

7. Adicione uma regra de segurança para permitir conexões HTTP de entrada (porta 80) na instância caso você ainda não tenha feito isso. Por padrão, um grupo de **N** segurança do assistente de inicialização foi criado para sua instância durante a execução. Se você não acrescentou regras adicionais de grupo de segurança, esse grupo contém apenas uma única regra para permitir conexões SSH.
- Abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
 - No navegador à esquerda, selecione Instances (Instâncias) e selecione sua instância.
 - Na guia Security (Segurança), exiba as regras de entrada. Você deve ver a seguinte regra:

Port range	Protocol	Source
22	tcp	0.0.0.0/0

 Warning

Usar `0.0.0.0/0` permite que todos os IPv4 endereços acessem sua instância usando SSH. Isso é aceitável para um período curto em um ambiente de teste, mas não é seguro em ambientes de produção. Na produção, você autorizará somente um endereço IP específico ou intervalo de endereços para acessar a instância.

- Se não houver uma regra de entrada para permitir conexões HTTP (porta 80), será necessário adicionar a regra agora. Escolha o link do grupo de segurança. Usando os procedimentos contidos em [Authorize inbound traffic for your Linux instances](#), adicione uma nova regra de segurança de entrada com os seguintes valores:

- Tipo: HTTP

- Protocolo: TCP
- Port Range: 80
- Source (Origem): personalizado

8. Teste o servidor web. Em um navegador, digite o endereço DNS público (ou o endereço IP público) de sua instância. Se não houver conteúdo em `/var/www/html`, você deverá verificar a página de teste do Apache, que exibirá a mensagem “It works!” (Funciona!).

Você pode obter o DNS público para sua instância usando o console do Amazon EC2 (verifique a coluna DNS IPv4 público; se essa coluna estiver oculta, escolha Preferências (o ícone em forma de engrenagem) e ative DNS público). IPv4

Verifique se o grupo de segurança da instância contém uma regra para permitir o tráfego HTTP na porta 80. Para saber mais, consulte [Add rules to security group](#).

 Important

Se você não estiver usando o Amazon Linux, talvez seja necessário configurar o firewall na instância para permitir essas conexões. Para obter mais informações sobre como configurar o firewall, consulte a documentação de sua distribuição específica.

O `httpd` do Apache é usado para os arquivos que são mantidos em um diretório chamado raiz de documentos do Apache. O diretório raiz de documentos Apache do Amazon Linux é `/var/www/html`, que, por padrão, é de propriedade da raiz.

Para permitir que a conta do `ec2-user` manipule arquivos nesse diretório, você deve modificar a propriedade e as permissões do diretório. Existem diversas maneiras de realizar essa tarefa. Neste tutorial, você adiciona `ec2-user` ao grupo `apache` para dar ao grupo `apache` a propriedade do diretório `/var/www` e atribuir permissões de gravação ao grupo.

Para definir permissões de arquivo

1. Adicione o usuário (neste caso, o `ec2-user`) ao grupo do `apache`.

```
[ec2-user ~]$ sudo usermod -a -G apache ec2-user
```

2. Faça logout e login novamente para selecionar o novo grupo verifique sua associação.
 - a. Faça logout (use o comando `exit` ou feche a janela do terminal):

```
[ec2-user ~]$ exit
```

- b. Para verificar sua associação no grupo apache, reconecte-se à instância e execute o comando a seguir:

```
[ec2-user ~]$ groups  
ec2-user adm wheel apache systemd-journal
```

3. Altere a propriedade do grupo do `/var/www` e seu conteúdo para o grupo do apache.

```
[ec2-user ~]$ sudo chown -R ec2-user:apache /var/www
```

4. Para adicionar as permissões de gravação do grupo e definir o ID do grupo nos subdiretórios futuros, altere as permissões de diretório de `/var/www` e de seus subdiretórios.

```
[ec2-user ~]$ sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod  
2775 {} \;
```

5. Para adicionar permissões de gravação do grupo, altere recursivamente as permissões de arquivo de `/var/www` e de seus subdiretórios:

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
```

Agora, `ec2-user` (e outros todos os futuros do grupo apache) poderão adicionar, excluir e editar arquivos na raiz do documento Apache, permitindo que você adicione conteúdo, como um site estático ou uma aplicação PHP.

Para proteger o servidor web (opcional)

Um servidor web que executa o protocolo HTTP não fornece nenhuma segurança de transporte para os dados que envia ou recebe. Quando você se conecta a um servidor HTTP usando um navegador da Web, o conteúdo URLs que você visita, o conteúdo das páginas da Web que você recebe e o conteúdo (incluindo senhas) de qualquer formulário HTML que você envia são todos visíveis para espiões em qualquer lugar ao longo do caminho da rede. A melhor prática para proteger o servidor web é instalar suporte para HTTPS (HTTP seguro), que protege os dados por meio de criptografia SSL/TLS.

Para obter informações sobre como habilitar o HTTPS no servidor, consulte [Tutorial: Configurar SSL/TLS em AL2023](#).

Etapa 2: Testar o servidor LAMP

Se o servidor estiver instalado e em execução, e suas permissões de arquivo estiverem definidas corretamente, a conta do `ec2-user` poderá criar um arquivo PHP no diretório `/var/www/html` disponível na Internet.

Para testar o servidor do LAMP

1. Crie um arquivo PHP no diretório base do Apache.

```
[ec2-user ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
```

Se você receber o erro "Permissão negada" ao tentar executar esse comando, tente fazer logout e login novamente para obter as permissões corretas do grupo que você configurou em [Para definir permissões de arquivo](#).

2. Em um navegador da web, digite a URL do arquivo que você acabou de criar. Essa URL é o endereço DNS público da instância seguido por uma barra e o nome do arquivo. Por exemplo:

```
http://my.public.dns.amazonaws.com/phpinfo.php
```

Você deve ver a página de informações do PHP:

PHP Version 8.1.7



System	Linux ip-172-31-16-77.ec2.internal 5.15.57-28.127.amzn2022.aarch64 #1 SMP Thu Aug 4 17:06:57 UTC 2022 aarch64
Build Date	Jun 7 2022 18:21:38
Build System	Linux
Build Provider	Amazon Linux
Compiler	gcc (GCC) 11.3.1 20220421 (Red Hat 11.3.1-2)
Architecture	aarch64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/10-opcache.ini, /etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gd.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-mysqld.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-simplexml.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlwriter.ini, /etc/php.d/20-xsl.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/30-xmldrader.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902,NTS
PHP Extension Build	API20210902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, bzip2.*, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
 Zend Engine v4.1.7, Copyright (c) Zend Technologies
 with Zend OPcache v8.1.7, Copyright (c), by Zend Technologies

Se você não vir essa página, verifique se o arquivo `/var/www/html/phpinfo.php` foi criado corretamente na etapa anterior. Você também pode verificar se todos os pacotes necessários foram instalados com o comando a seguir.

```
[ec2-user ~]$ sudo dnf list installed httpd mariadb-server php-mysqld
```

Se alguns dos pacotes necessários não estiverem listados na saída, instale-os com o comando `sudo yum install package`.

3. Exclua o arquivo `phpinfo.php`. Embora essas informações possam ser úteis, elas não devem ser transmitidas pela Internet por motivos de segurança.

```
[ec2-user ~]$ rm /var/www/html/phpinfo.php
```

Agora você deve ter um servidor web do LAMP totalmente funcional. Se adicionar conteúdo ao diretório base do Apache em `/var/www/html`, você deverá poder visualizar esse conteúdo no endereço DNS público de sua instância.

Etapa 3: Proteger o servidor do banco de dados

A instalação padrão do servidor MariaDB tem vários recursos que são bons para teste e desenvolvimento, mas devem ser desabilitados ou removidos em servidores de produção. O comando `mysql_secure_installation` orienta você durante o processo de configuração de uma senha raiz e da remoção de recursos não seguros da instalação. Mesmo que você não esteja planejando usar o servidor MariaDB é recomendável executar este procedimento.

Para proteger o servidor MariaDB

1. Inicie o servidor MariaDB.

```
[ec2-user ~]$ sudo systemctl start mariadb
```

2. Executar `mysql_secure_installation`.

```
[ec2-user ~]$ sudo mysql_secure_installation
```

- a. Quando solicitado, digite uma senha para a conta raiz.
 - i. Digite a senha raiz atual. Por padrão, a conta raiz não tem uma senha definida. Pressione Enter.
 - ii. Digite **Y** para definir uma senha e digite uma senha segura duas vezes. Para obter mais informações sobre como criar uma senha segura, consulte <https://identitysafe.norton.com/password-generator/>. Armazene essa senha em um lugar seguro.

A configuração de uma senha raiz para o MariaDB é somente a medida mais básica para proteger seu banco de dados. Ao criar ou instalar um aplicativo controlado por

banco de dados, geralmente, você cria um usuário de serviço de banco para esse aplicativo e evita usar a conta raiz para qualquer coisa que não seja a administração do banco de dados.

- b. Digite **Y** para remover as contas de usuários anônimos.
 - c. Digite **Y** para desabilitar o recurso de login remoto da raiz.
 - d. Digite **Y** para remover o banco de dados de teste.
 - e. Digite **Y** para recarregar as tabelas de privilégios e salvar suas alterações.
3. (Opcional) Se você não pretende usar o servidor MariaDB imediatamente, interrompa-o. Você poderá reiniciá-lo quando precisar dele novamente.

```
[ec2-user ~]$ sudo systemctl stop mariadb
```

4. (Opcional) Se você quiser que o servidor MariaDB seja iniciado a cada inicialização, digite o comando a seguir.

```
[ec2-user ~]$ sudo systemctl enable mariadb
```

Etapa 4: Instalação (opcional) phpMyAdmin

[phpMyAdmin](#) é uma ferramenta de gerenciamento de banco de dados baseada na web que você pode usar para visualizar e editar os bancos de dados MySQL na sua instância do EC2. Siga as etapas a seguir para instalar e configurar o phpMyAdmin em sua instância do Amazon Linux.

Important

Não recomendamos o uso phpMyAdmin para acessar um servidor LAMP, a menos que você tenha habilitado SSL/TLS o Apache; caso contrário, sua senha de administrador de banco de dados e outros dados serão transmitidos de forma insegura pela Internet. Para recomendações de segurança dos desenvolvedores, consulte [Protegendo sua phpMyAdmin instalação](#). Para obter informações gerais sobre como proteger um servidor web em uma instância do EC2, consulte [Tutorial: Configurar SSL/TLS em AL2023](#).

Para instalar phpMyAdmin

1. Instale as dependências necessárias.

```
[ec2-user ~]$ sudo dnf install php-mbstring php-xml -y
```

2. Reinicie o Apache.

```
[ec2-user ~]$ sudo systemctl restart httpd
```

3. Reinicie o php-fpm.

```
[ec2-user ~]$ sudo systemctl restart php-fpm
```

4. Navegue até o diretório base do Apache em `/var/www/html`.

```
[ec2-user ~]$ cd /var/www/html
```

5. Selecione um pacote de origem para a phpMyAdmin versão mais recente em <https://www.phpmyadmin.net/downloads>. Para fazer download do arquivo diretamente para a instância, copie o link e cole-o em um comando `wget`, como neste exemplo:

```
[ec2-user html]$ wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz
```

6. Crie uma pasta `phpMyAdmin` e extraia o pacote dela com o comando a seguir.

```
[ec2-user html]$ mkdir phpMyAdmin && tar -xvzf phpMyAdmin-latest-all-languages.tar.gz -C phpMyAdmin --strip-components 1
```

7. Exclua o `phpMyAdmin-latest-all-languages.tar.gz` tarball.

```
[ec2-user html]$ rm phpMyAdmin-latest-all-languages.tar.gz
```

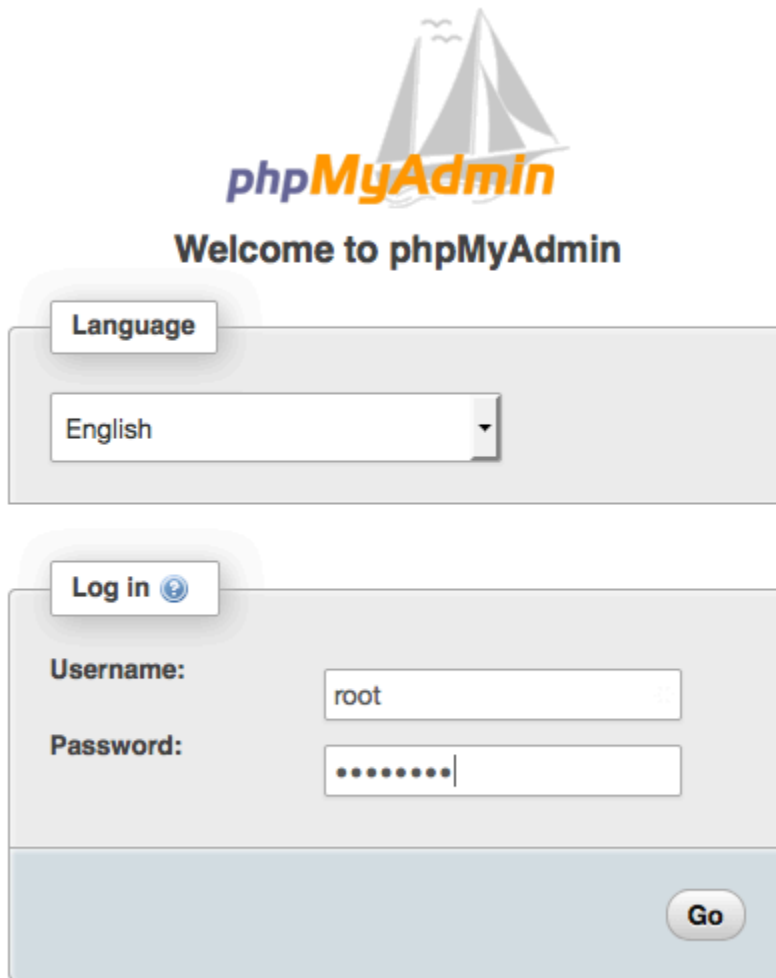
8. (Opcional) Se o servidor MySQL não estiver em execução, inicie-o agora.

```
[ec2-user ~]$ sudo systemctl start mariadb
```

9. Em um navegador da Web, digite a URL da sua phpMyAdmin instalação. Essa URL é o endereço DNS público (ou o endereço IP público) da instância seguido por uma barra e o nome do diretório de instalação. Por exemplo:

```
http://my.public.dns.amazonaws.com/phpMyAdmin
```


Você deve ver a página de phpMyAdmin login:



phpMyAdmin

Welcome to phpMyAdmin

Language

English

Log in ?

Username: root

Password:

Go

10. Faça login na sua phpMyAdmin instalação com o nome de `root` usuário e a senha raiz do MySQL que você criou anteriormente.

A instalação ainda deve ser configurada antes que você a coloque em serviço. Sugerimos que você comece criando manualmente o arquivo de configuração, da seguinte maneira:

- a. Para começar com um arquivo de configuração mínima, use seu editor de texto favorito para criar um novo arquivo e, em seguida, copie o conteúdo de `config.sample.inc.php` para ele.
- b. Salve o arquivo como `config.inc.php` no phpMyAdmin diretório que contém `index.php`.
- c. Consulte as instruções de criação pós-arquivo na seção [Usando o script](#) de phpMyAdmin instalação das instruções de instalação para qualquer configuração adicional.

Para obter informações sobre o uso phpMyAdmin, consulte o [Guia phpMyAdmin do usuário](#).

Solução de problemas

Esta seção oferece sugestões para resolver problemas comuns que podem surgir durante a configuração de um novo servidor do LAMP.

Não consigo me conectar ao servidor usando um navegador da web

Execute as seguintes verificações para ver se o servidor da web do Apache está em execução e acessível.

- O servidor web está em execução?

Você pode verificar se httpd está ativo executando o seguinte comando:

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

Se o processo httpd não estiver em execução, repita as etapas descritas em [Para preparar o servidor LAMP](#).

- O firewall está configurado corretamente?

Verifique se o grupo de segurança da instância contém uma regra para permitir o tráfego HTTP na porta 80. Para saber mais, consulte [Add rules to security group](#).

Não consigo me conectar ao meu servidor usando HTTPS

Execute as seguintes verificações para ver se o servidor da web do Apache está configurado para dar suporte a HTTPS.

- O servidor Web está configurado corretamente?

Depois de instalar o Apache, o servidor é configurado para tráfego HTTP. Para suportar HTTPS, ative o TLS no servidor e instale um certificado SSL. Para mais informações, consulte [Tutorial: Configurar SSL/TLS em AL2023](#).

- O firewall está configurado corretamente?

Verifique se o grupo de segurança da instância contém uma regra para permitir o tráfego HTTPS na porta 443. Para saber mais, consulte [Authorize inbound traffic for your Linux instances](#).

Tópicos relacionados

Para obter mais informações sobre como transferir arquivos para sua instância ou instalar um WordPress blog em seu servidor web, consulte a documentação a seguir:

- [Transferir arquivos para uma instância do Linux usando WinSCP](#) no Guia do usuário do Amazon EC2.
- [Transferir arquivos para instâncias do Linux usando um cliente SCP](#) no Guia do usuário do Amazon EC2.
- [Tutorial: hospede um WordPress blog no AL2023](#)

Para obter mais informações sobre os comandos e o software usados neste tutorial, consulte as seguintes páginas da web:

- Servidor web Apache: <http://httpd.apache.org/>
- Servidor de banco de dados MariaDB: <https://mariadb.org/>
- Linguagem de programação PHP: <http://php.net/>

Para obter mais informações sobre como registrar um nome de domínio para o servidor web ou transferir um nome de domínio existente para este host, consulte [Como criar e migrar domínios e subdomínios para o Amazon Route 53](#) no Guia do desenvolvedor do Amazon Route 53.

Tutorial: Configurar SSL/TLS em AL2023

Secure Layer/Transport Sockets Layer Security (SSL/TLS) creates an encrypted channel between a web server and web client that protects data in transit from being eavesdropped on. This tutorial explains how to add support manually for SSL/TLS em uma instância EC2 com AL2023 um servidor web Apache). Este tutorial pressupõe que você não esteja usando um balanceador de carga. Se você estiver usando Elastic Load Balancing, poderá optar por configurar o descarregamento do SSL no balanceador de carga, usando, em vez disso, um certificado do [AWS Certificate Manager](#).

Por motivos históricos, a criptografia na Web é conhecida simplesmente como SSL. Embora navegadores da web ainda ofereçam suporte a SSL, o protocolo sucessor TLS é menos vulnerável a ataques. O AL2023 desabilita o suporte do servidor para todas as versões do SSL por padrão. [Órgãos de normas de segurança](#) consideram o TLS 1.0 não seguro. O TLS 1.0 e TLS 1.1 foram formalmente [preteridos](#) em março de 2021. Este tutorial contém orientações baseadas exclusivamente na ativação do TLS 1.2. O TLS 1.3 foi finalizado em 2018 e está disponível AL2 desde que a biblioteca TLS subjacente (OpenSSL neste tutorial) seja suportada e habilitada. [Os clientes devem ser compatíveis com o TLS 1.2 ou posterior até 28 de junho de 2023](#). Para obter mais informações sobre os padrões de criptografia atualizados, consulte [RFC 7568](#) e [RFC 8446](#).

Este tutorial refere-se à criptografia da Web moderna simplesmente como TLS.

Important

Esses procedimentos são destinados ao AL2023. Se você estiver tentando configurar uma instância do EC2 executando uma distribuição diferente ou uma instância executando uma versão antiga do Amazon Linux, alguns procedimentos deste tutorial poderão não funcionar. Para o Ubuntu, consulte a seguinte documentação da comunidade do Ubuntu: [Open SSL on Ubuntu](#) (Open SSL no Ubuntu). Para o Red Hat Enterprise Linux, consulte: [Como configurar o Servidor Web Apache HTTP](#). Para outras distribuições, consulte a documentação específica.

Note

Como alternativa, você pode usar o AWS Certificate Manager (ACM) para enclaves AWS Nitro, que é um aplicativo de enclave que permite usar SSL/TLS certificados públicos e privados com seus aplicativos e servidores web em execução em instâncias do Amazon EC2 com o Nitro Enclaves. AWS O Nitro Enclaves é um recurso do Amazon EC2 que permite a criação de ambientes computacionais isolados para proteger e processar com segurança dados altamente confidenciais, como certificados e chaves privadas. SSL/TLS O ACM for Nitro Enclaves funciona com nginx em execução em sua instância do Linux do Amazon EC2 para criar chaves privadas, distribuir certificados e chaves privadas, além de gerenciar renovações de certificados. Para usar o ACM for Nitro Enclaves, é necessário usar uma instância do Linux habilitada para enclave.

Para obter mais informações, consulte [O que são AWS Nitro Enclaves?](#) e [AWS Certificate Manager para Nitro Enclaves](#) no Guia do usuário do AWS Nitro Enclaves.

Conteúdos

- [Pré-requisitos](#)
- [Etapa 1: habilitar o TLS no servidor](#)
- [Etapa 2: obter um certificado assinado por uma CA](#)
- [Etapa 3: testar e intensificar a configuração de segurança](#)
- [Solução de problemas](#)

Pré-requisitos

Antes de começar este tutorial, conclua as seguintes etapas:

- Execute uma instância com suporte do EBS AL2023 . Para obter mais informações, consulte [AL2023 no Amazon EC2](#).
- Configure seus grupos de segurança para permitir que sua instância aceite conexões nas seguintes portas TCP:
 - SSH (porta 22)
 - HTTP (porta 80)
 - HTTPS (porta 443)

Para saber mais, consulte [Authorize inbound traffic for your Linux instances](#) no Guia do usuário do Amazon EC2.

- Instale o servidor Web Apache. Para step-by-step obter instruções, consulte [Tutorial: Instalar um servidor LAMP em AL2023](#). Somente o pacote httpd e suas dependências são necessários e, portanto, você pode ignorar as instruções que envolvem PHP e MariaDB.
- Para identificar e autenticar sites, a infraestrutura de chave pública (PKI) do TLS depende do Sistema de Nomes de Domínio (DNS). Para usar sua instância do EC2 para hospedar um site público, você precisará registrar um nome de domínio para seu servidor da Web ou transferir um nome de domínio existente para o host do Amazon EC2. Há vários serviços de registro de domínio e de hospedagem DNS de terceiros disponíveis para isso, ou você pode usar o [Amazon Route 53](#).

Etapa 1: habilitar o TLS no servidor

Esse procedimento conduz você pelo processo de configuração do TLS AL2023 com um certificado digital autoassinado.

Note

Um certificado autoassinado é aceitável para testes, mas não para produção. Quando você expõe seu certificado autoassinado na Internet, os visitantes de seu site recebem avisos de segurança.

Para habilitar o TLS em um servidor

1. Conecte-se à sua instância e confirme se o Apache está em execução. Para obter mais informações, consulte [Conexão com AL2023 instâncias](#).

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

Se o valor retornado não for "habilitado", inicie o Apache e configure-o para iniciar sempre que o sistema for inicializado.

```
[ec2-user ~]$ sudo systemctl start httpd && sudo systemctl enable httpd
```

2. Para garantir que todos os pacotes de software estejam atualizados, execute uma atualização rápida de software em sua instância. Esse processo pode levar alguns minutos, mas é importante ter certeza de que você tem as atualizações de segurança e correções de bug mais recentes.

Note

A opção `-y` instala as atualizações sem solicitar confirmação. Para examinar as atualizações antes da instalação, você pode omitir essa opção.

```
[ec2-user ~]$ sudo dnf install openssl mod_ssl
```

3. Depois de inserir o seguinte comando, você será levado a um prompt onde poderá inserir informações sobre seu site.

```
[ec2-user ~]$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/pki/tls/private/apache-selfsigned.key -out /etc/pki/tls/certs/apache-selfsigned.crt
```

Isso gera dois arquivos:

- /etc/pki/tls/private/apache-selfsigned.key(a chave privada)
- /etc/pki/tls/certs/apache-selfsigned.crt(o certificado autoassinado)

O nome do arquivo do certificado corresponde ao padrão atribuído na SSLCertificateFile diretiva em/etc/httpd/conf.d/ssl.conf.

Sua instância agora possui os seguintes arquivos que você usará para configurar seu servidor seguro e criar um certificado para teste:

- /etc/httpd/conf.d/ssl.conf

O arquivo de configuração para mod_ssl. Contém as diretrizes que informam ao Apache onde encontrar chaves de criptografia e certificados, as versões do protocolo TLS a serem permitidas e as cifras de criptografia a serem aceitas. Este será o seu arquivo de certificado local:

- /etc/pki/tls/certs/apache-selfsigned.crt
- /etc/pki/tls/private/apache-selfsigned.key

O .crt arquivo contém o certificado autoassinado e o .key arquivo contém a chave privada. O Apache requer que o certificado e a chave estejam no formato PEM, que consiste em caracteres ASCII codificados em Base64 enquadrados pelas linhas "BEGIN" e "END", como neste exemplo.

```
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBAgwgaggSkAgEAAoIBAQD2KKx/8Zk94m1q
3gQMZF9ZN66Ls19+3tHAgQ5Fpo9KJDhzLj00CI8u1PTcGmAah5kEitCEc0wzmNeo
BC10wYR6G0rGaKtK9Dn7CuIjvubtUysVyQoMVPQ97ldeakHWeRMiEJFXg6kZZ0vr
GvwnKoMh3D1K44D9dX7IDua2P1Yx5+eroA+1Lqf32ZSaA00bBIMIYTHigwbHMZoT
...
56tE7THvH7v0Ef4/iU0sIrEzaMaJ0mqkmY1A70qQGQKBgBF3H1qNRNHuyMcPODFs
```

```

27hDzPDinrquSEvoZlggkDMlh2irTiipJ/GhkvTpoQlv0fK/VXw8vSgeaBuhwJvS
LXU9HvYq0U604FgD3nAyB9hI0BE13r1HjUvbjT7moH+RhNz6eqqdsccCS09VtRA0
4QQvAq0a8UheYeoXLdWcHaLP
-----END PRIVATE KEY-----

-----BEGIN CERTIFICATE-----
MIIEAzCCA10gAwIBAgICWxQwDQYJKoZIhvcNAQELBQAwbGExCzAJBgNVBAYTAi0t
MRIwEAYDVQQIDAlTb211U3RhdGUxETAPBgNVBACMFNvbWVkaXR5MRkwFwYDVQQK
DBBTb211T3JnYW5pemF0aW9uMR8wHQYDVQQLDBZTb211T3JnYW5pemF0aW9uYXV
bm10MRkwFwYDVQQDDDBBpcC0xNzItMzEtMjAtMjMMSQwIgYJKoZIhvcNAQkBFhVy
...
z5rRUE/XzxRLBZ0oWZpNWTXJkQ3uFYH6s/sBwtHpKKZMz0vDedREjNKAvk4ws6F0
CuIjvubtUysVyQoMVPQ971deakHWeRMiEJFXg6kZZ0vrGvwnKoMh3D1K44D9d1U3
WanXWehT6FiSZvB4sTEXXJN2jdw8g+sHGnZ8zC0sc1knYhHrCVD2vnB1ZJKSZvak
3ZazhBxtQSukFM0nWPP2a0DMMFGYUHOd0BQE8sBJxg==
-----END CERTIFICATE-----

```

Os nomes de arquivos e as extensões são uma conveniência e não têm efeito na função. Por exemplo, você pode chamar um certificado de `cert.crt`, `cert.pem` ou de um outro nome de arquivo qualquer, desde que a diretiva relacionada no arquivo `ssl.conf` use o mesmo nome.

Note

Ao substituir os arquivos TLS padrão por seus próprios arquivos personalizados, verifique se eles estão no formato PEM.

4. Reinicie o Apache.

```
[ec2-user ~]$ sudo systemctl restart httpd
```

Note

Verifique se a porta TCP 443 está acessível em sua instância do EC2, conforme descrito anteriormente.

5. Seu servidor da Web do Apache agora deve oferecer suporte a HTTPS (HTTP seguro) por meio da porta 443. Teste-o digitando o endereço IP ou o nome do domínio totalmente qualificado de sua instância do EC2 em uma barra de URL de um navegador com o prefixo **https://**.

Como você está se conectando a um site com um certificado de host autoassinado não confiável, o navegador poderá exibir uma série de avisos de segurança. Ignore os avisos e continue para o site.

Se a página de teste padrão do Apache for aberta, a configuração do TLS no servidor estará correta. Todos os dados que passam entre o navegador e o servidor agora estão criptografados.

 Note

Para impedir que os visitantes do site encontrem telas de avisos, você precisa obter um certificado assinado por uma CA confiável que, além de criptografar, também autentique você publicamente como o proprietário do site.

Etapa 2: obter um certificado assinado por uma CA

Você pode seguir este processo para obter um certificado assinado por uma CA:

- Gere uma solicitação de assinatura de certificado (CSR) a partir de uma chave privada
- Enviar a CSR para uma autoridade de certificação (CA)
- Obtenha um certificado de host assinado
- Configure o Apache para usá-lo

Um certificado de host TLS X.509 autoassinado é idêntico em termos criptológicos a um certificado assinado por uma CA. A diferença é social, não matemática. Uma CA promete validar, no mínimo, a propriedade de um domínio antes de emitir um certificado para um candidato. Cada navegador da Web contém uma lista de CAs informações confiáveis do fornecedor do navegador para fazer isso. Primariamente, um certificado X.509 consiste em uma chave pública, que corresponde à chave privada do servidor, e uma assinatura pela CA que é vinculada criptograficamente à chave pública. Quando um navegador se conecta a um servidor web via HTTPS, o servidor apresenta um certificado para o navegador verificar sua lista de confiáveis CAs. Se o assinante estiver na lista ou for acessível por meio de uma cadeia de confiança que consiste em outros assinantes confiáveis, o navegador negociará um canal rápido de dados criptografados com o servidor e carregará a página.

Geralmente, os certificados são caros devido ao trabalho envolvido na validação das solicitações, portanto, vale a pena comparar os preços. Alguns CAs oferecem certificados de nível básico

gratuitamente. O mais notável deles CAs é o projeto [Let's Encrypt](#), que também suporta a automação do processo de criação e renovação de certificados. Para obter mais informações sobre como usar um certificado Let's Encrypt, consulte [Obtenção do Certbot](#).

Se você planeja oferecer serviços de nível comercial, o [AWS Certificate Manager](#) é uma boa opção.

É importante ter um certificado de host subjacente. Desde 2019, grupos [governamentais](#) e do [setor](#) recomendam usar um tamanho de chave (módulo) mínimo de 2.048 bits para chaves de RSA para a proteção de documentos até 2030. O tamanho do módulo padrão gerado pelo OpenSSL AL2023 in é de 2048 bits, o que é adequado para uso em um certificado assinado pela CA. No procedimento a seguir, uma etapa opcional é fornecida para aqueles que desejam uma chave personalizada, por exemplo, uma com módulo maior ou que usa um algoritmo diferente de criptografia.

 Important

As instruções para adquirir certificados de host assinados pela CA não funcionarão, a menos que você possua um domínio DNS registrado e hospedado.

Para obter um certificado assinado por uma CA

1. Conecte-se à sua instância e navegue até/etc/pki/tls/private/. Este é o diretório onde você armazenará a chave privada do servidor para TLS. Se você preferir usar uma chave de host existente para gerar a CSR, vá para a Etapa 3. Para obter mais informações sobre como se conectar à sua instância, consulte [Conexão com AL2023 instâncias](#)
2. (Opcional) Gerar uma nova chave privada. Estes são alguns exemplos de configurações de chave. Qualquer uma das chaves resultantes funciona com seu servidor Web, mas elas variam no grau e no tipo de segurança que elas implementam.
 - Exemplo 1: criar uma chave host de RSA padrão. O arquivo resultante, **custom.key**, é uma chave privada de RSA de 2048 bits.

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key
```

- Exemplo 2: criar uma chave de RSA mais forte com um módulo maior. O arquivo resultante, **custom.key**, é uma chave privada de RSA de 4096 bits.

```
[ec2-user ~]$ sudo openssl genrsa -out custom.key 4096
```

- Exemplo 3: criar uma chave de RSA de 4096 bits criptografada com proteção por senha. O arquivo resultante, **custom.key**, é uma chave privada de RSA de 4096 bits criptografada com a cifra AES-128.

 Important

A criptografia da chave fornece maior segurança, mas como uma chave criptografada requer uma senha, os serviços que dependem dela não podem ser iniciados automaticamente. Sempre que usar essa chave, você precisará fornecer a senha (no exemplo anterior, "abcde12345") por meio de uma conexão SSH.

```
[ec2-user ~]$ sudo openssl genrsa -aes128 -passout pass:abcde12345 -out  
custom.key 4096
```

- Exemplo 4: criar uma chave usando uma cifra não RSA. A criptografia RSA pode ser relativamente devagar devido ao tamanho de suas chaves públicas, que são baseadas no produto de dois números primos grandes. No entanto, é possível criar chaves para TLS que usam códigos não RSA. As chaves baseadas em matemática de curvas elípticas são menores e computacionalmente mais rápidas para fornecer um nível de segurança equivalente.

```
[ec2-user ~]$ sudo openssl ecparam -name prime256v1 -out custom.key -genkey
```

O resultado é uma chave privada de curva elíptica de 256 bits que usa prime256v1, uma "curva nomeada" compatível com OpenSSL. A força de criptografia é um pouco maior que uma chave de RSA de 2048 bits, [de acordo com o NIST](#).

 Note

Nem todos os CAs oferecem o mesmo nível de suporte para elliptic-curve-based chaves que para chaves RSA.

Certifique-se de que a nova chave privada tenha propriedade e permissões altamente restritivas (owner=root, group=root, somente para o proprietário). read/write O comando será o mostrado no exemplo a seguir.

```
[ec2-user ~]$ sudo chown root:root custom.key
[ec2-user ~]$ sudo chmod 600 custom.key
[ec2-user ~]$ ls -al custom.key
```

Os comandos anteriores produzem o resultado a seguir.

```
-rw----- root root custom.key
```

Depois de criar e configurar uma chave satisfatória, você pode criar uma CSR.

3. Crie uma CSR usando sua chave preferida. O exemplo a seguir usa **custom.key**.

```
[ec2-user ~]$ sudo openssl req -new -key custom.key -out csr.pem
```

A OpenSSL abre uma caixa de diálogo e solicita a informação exibida na tabela a seguir. Todos os campos, exceto Common Name (Nome comum), são opcionais para um certificado de host básico validado por domínio.

Nome	Descrição	Exemplo
Nome do país	A abreviação ISO de duas letras para seu país.	US (=Estados Unidos)
Nome do estado ou província	O nome do estado ou província onde sua organização está localizada. Este nome não pode ser abreviado.	Washington
Nome da localidade	A localização de sua organização, como uma cidade.	Seattle
Nome da organização	A razão social completa da sua organização. Não abrevie o nome de sua organização.	Corporação de exemplo
Nome da unidade organizacional	Informações organizacionais adicionais, se houver.	Departamento de exemplo

Nome	Descrição	Exemplo
Nome comum	Esse valor deve corresponder exatamente ao endereço Web que você espera que os usuários digitem em um navegador. Geralmente, isso significa um nome de domínio com um nome de host ou alias prefixados na forma www.example.com . Em testes com um certificado autoassinado e sem resolução de DNS, o nome comum pode consistir apenas no nome do host. CAs também oferecem certificados mais caros que aceitam nomes curingas, como. *.example.com	www.exemplo.com
Endereço de e-mail	O endereço de e-mail do administrador do servidor.	someone@example.com

Finalmente, a OpenSSL solicita uma senha de desafio opcional. Essa senha se aplica somente à CSR e às transações entre você e sua CA, portanto, siga as recomendações da CA sobre este e o outro campo opcional, nome da empresa opcional. A senha de desafio da CSR não tem nenhum efeito sobre a operação do servidor.

O arquivo resultante **csr.pem** contém sua chave pública, a assinatura digital de sua chave pública e os metadados que você inseriu.

- Envie a CSR a uma CA. Geralmente, isso consiste em abrir seu arquivo de CSR em um editor de texto e copiar o conteúdo em um formulário da Web. No momento, você pode ser solicitado a fornecer um ou mais nomes alternativos de assunto (SANs) para serem colocados no certificado. Se **www.example.com** for o nome comum, **example.com** seria um bom SAN e vice-versa. Um visitante de seu site que digitar qualquer um desses nomes verá uma conexão livre de erros. Se o formulário web da CA permitir, inclua o nome comum na lista de SANs. Alguns o CAs incluem automaticamente.

Depois que sua solicitação é aprovada, você recebe um novo certificado de host assinado pela CA. Você também pode receber uma instrução para fazer download de um arquivo de certificado

intermediário que contém os certificados adicionais necessários para concluir a cadeia de confiança da CA.

Note

Sua CA pode enviar a você arquivos em vários formatos com várias finalidades. Para este tutorial, você deve usar apenas um arquivo de certificado em formato PEM, que geralmente (mas nem sempre) é identificado por uma extensão de arquivo `.pem` ou `.crt`. Se você não tiver certeza sobre qual arquivo usar, abra os arquivos com um editor de texto e localize um que contenha um ou mais blocos com a linha a seguir.

```
- - - - -BEGIN CERTIFICATE - - - - -
```

O arquivo também deve terminar com a linha a seguir.

```
- - - - -END CERTIFICATE - - - - -
```

Você também pode testar um arquivo na linha de comando da forma a seguir.

```
[ec2-user certs]$ openssl x509 -in certificate.crt -text
```

Verifique se as linhas aparecem no arquivo. Não use os arquivos que terminam com `.p7b`, `.p7c` ou extensões de arquivo semelhantes.

5. Coloque o novo certificado assinado pela CA e quaisquer certificados intermediários no diretório `/etc/pki/tls/certs`.

Note

Há várias maneiras para fazer upload do novo certificado para a instância do EC2, mas a maneira mais simples e informativa é abrir um editor de texto (`vi`, `nano`, bloco de notas etc.) no seu computador local e na sua instância e, em seguida, copiar e colar os conteúdos do arquivo entre eles. Você precisa de permissões raiz [`sudo`] ao realizar essas operações na instância do EC2. Dessa forma, você vê imediatamente se há algum problema de permissão ou de caminho. No entanto, tenha cuidado para não adicionar mais linhas ao copiar o conteúdo ou ao alterá-lo de alguma maneira.

De dentro do `/etc/pki/tls/certs` diretório, verifique se as configurações de propriedade, grupo e permissão do arquivo correspondem aos AL2023 padrões altamente restritivos (owner=root, group=root, somente para proprietário). read/write O exemplo a seguir mostra os comandos a serem usados.

```
[ec2-user certs]$ sudo chown root:root custom.crt
[ec2-user certs]$ sudo chmod 600 custom.crt
[ec2-user certs]$ ls -al custom.crt
```

Esses comandos devem produzir o resultado a seguir.

```
-rw----- root root custom.crt
```

As permissões para o arquivo de certificado intermediário são menos estritas (owner=root, group=root, proprietário pode gravar, grupo pode ler, mundo pode ler). O exemplo a seguir mostra os comandos a serem usados.

```
[ec2-user certs]$ sudo chown root:root intermediate.crt
[ec2-user certs]$ sudo chmod 644 intermediate.crt
[ec2-user certs]$ ls -al intermediate.crt
```

Esses comandos devem produzir o resultado a seguir.

```
-rw-r--r-- root root intermediate.crt
```

6. Coloque a chave privada que você usou para criar o CSR no diretório `/etc/pki/tls/private/`.

Note

Há várias maneiras para fazer upload da chave personalizada para a instância do EC2, mas a maneira mais simples e informativa é abrir um editor de texto (vi, nano, bloco de notas etc.) no seu computador local e na sua instância e, em seguida, copiar e colar os conteúdos do arquivo entre eles. Você precisa de permissões raiz [sudo] ao realizar essas operações na instância do EC2. Dessa forma, você vê imediatamente se há

algum problema de permissão ou de caminho. No entanto, tenha cuidado para não adicionar mais linhas ao copiar o conteúdo ou ao alterá-lo de alguma maneira.

De dentro do `/etc/pki/tls/private` diretório, use os comandos a seguir para verificar se as configurações de propriedade, grupo e permissão do arquivo correspondem aos AL2023 padrões altamente restritivos (owner=root, group=root, somente para proprietário). read/write

```
[ec2-user private]$ sudo chown root:root custom.key
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ ls -al custom.key
```

Esses comandos devem produzir o resultado a seguir.

```
-rw----- root root custom.key
```

7. Edite `/etc/httpd/conf.d/ssl.conf` para refletir seu novo certificado e arquivos de chave.

 Note

Certifique-se de que `SSLEngine on` esteja definido no seu VirtualHost bloco.

- a. Forneça o caminho e o nome do arquivo do certificado de host assinado por CA na diretiva `SSLCertificateFile` do Apache:

```
SSLCertificateFile /etc/pki/tls/certs/custom.crt
```

- b. Se você receber um arquivo de certificado intermediário (`intermediate.crt` neste exemplo), forneça o caminho e o nome do arquivo usando a diretiva `SSLCACertificateFile` do Apache:

```
SSLCACertificateFile /etc/pki/tls/certs/intermediate.crt
```


 Note

Alguns CAs combinam o certificado do host e os certificados intermediários em um único arquivo, tornando a `SSLCACertificateFile` diretiva desnecessária. Consulte as instruções fornecidas pela CA.

- c. Forneça o caminho e o nome do arquivo da chave privada (`custom.key` neste exemplo) na diretiva `SSLCertificateKeyFile` do Apache:

```
SSLCertificateKeyFile /etc/pki/tls/private/custom.key
```

8. Salve o `/etc/httpd/conf.d/ssl.conf` e reinicie o Apache.

```
[ec2-user ~]$ sudo systemctl restart httpd
```

9. Teste seu servidor inserindo seu nome de domínio em uma barra de URL do navegador com o prefixo `https://`. Seu navegador deve carregar a página de teste via HTTPS sem gerar erros.

Etapa 3: testar e intensificar a configuração de segurança

Depois que o SSL/TLS estiver operacional e exposto ao público, você precisará testar se ele é realmente seguro. É fácil fazer isso usando serviços online, como o [Qualys SSL Labs](#) que executa uma análise completa e gratuita de sua configuração de segurança. Com base nos resultados, você pode decidir intensificar a configuração de segurança padrão controlando quais protocolos você aceita, quais cifras você prefere e quais você exclui. Para obter mais informações, consulte [como a Qualys formula suas pontuações](#).

 Important

Os testes no mundo real são cruciais para a segurança do servidor. Pequenos erros de configuração podem resultar em rupturas de segurança sérias e em perda de dados. Como as práticas de segurança recomendadas são alteradas constantemente em resposta a pesquisas e a ameaças emergentes, auditorias periódicas da segurança são essenciais para uma boa administração do servidor.

No site [Qualys SSL Labs](https://www.qualys.com/ssllabs), digite o nome do domínio totalmente qualificado de seu servidor no formato **www.example.com**. Depois de dois minutos, você recebe uma classificação (de A a F) para seu site e um detalhamento dos resultados. A tabela a seguir resume o relatório de um domínio com configurações idênticas à configuração padrão do Apache ativada e com um certificado Certbot padrão. AL2023

Classificação geral	B
Certificado	100%
Suporte ao protocolo	95%
Troca de chaves	70%
Intensidade da cifra	90%

Embora a visão geral mostre que a configuração é mais sólida, o relatório detalhado sinaliza vários possíveis problemas, listados aqui em ordem de gravidade:

✗ A RC4 cifra é compatível com o uso de alguns navegadores mais antigos. Uma cifra é o núcleo matemático de um algoritmo de criptografia. RC4, [uma cifra rápida usada para criptografar fluxos de dados TLS, é conhecida por ter várias fraquezas graves](#). A menos que você tenha boas razões para oferecer suporte a navegadores legados, você deve desabilitar isso.

✗ Versões antigas do TLS são compatíveis. A configuração é compatível com o TLS 1.0 (já obsoleto) e o TLS 1.1 (em um caminho para a reprovação). Apenas o TLS 1.2 é recomendado desde 2018.

✗ O sigilo de encaminhamento não é totalmente compatível. O [sigilo encaminhado](#) é um recurso de algoritmos que criptografam usando chaves de sessão temporárias (efêmeras) derivadas da chave privada. Na prática, isso significa que os atacantes não podem descriptografar dados HTTPS mesmo que tenham a chave privada de longo prazo de um servidor Web.

Para corrigir e preparar futuramente a configuração do TLS

1. Abra o arquivo de configuração `/etc/httpd/conf.d/ssl.conf` em um editor de texto e comente as seguintes linhas digitando “#” no início delas.

```
#SSLProtocol all -SSLv3
```

2. Adicione a seguinte diretiva:

```
#SSLProtocol all -SSLv3
SSLProtocol -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 +TLSv1.2
```

Essa diretiva desabilita explicitamente as versões 2 e 3 do SSL, bem como as versões 1.0 e 1.1 do TLS. O servidor agora se recusa a aceitar conexões criptografadas com clientes que não estejam usando o TLS 1.2. A expressão detalhada na diretiva transmite mais claramente, para um leitor humano, para que o servidor está configurado.

Note

Desabilitar as versões 1.0 e 1.1 do TLS dessa forma bloqueia o acesso ao seu site de uma pequena porcentagem de navegadores da Web desatualizados.

Para modificar a lista de cifras permitidas

1. No arquivo de configuração `/etc/httpd/conf.d/ssl.conf`, localize a seção com a diretiva **SSLCipherSuite** e comente a linha existente ao inserir `"#"` no início dela.

```
#SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5
```

2. Especifique conjuntos de criptografia explícitos e uma ordem de cifra que priorize o sigilo antecipado e evite cifras inseguras. A diretiva `SSLCipherSuite` usada aqui é baseada na saída do [gerador de configuração SSL do Mozilla](#), que adapta uma configuração TLS ao software específico em execução no seu servidor. Primeiro, determine suas versões do Apache e do OpenSSL usando os comandos a seguir.

```
[ec2-user ~]$ yum list installed | grep httpd

[ec2-user ~]$ yum list installed | grep openssl
```

Por exemplo, se a informação exibida for Apache 2.4.34 e OpenSSL 1.0.2, insira esses valores no gerador. Se você escolher o modelo de compatibilidade "moderno", isso criará uma diretiva `SSLCipherSuite` que impõe a segurança de forma agressiva, mas ainda funciona para a maioria dos navegadores. Se o software não oferecer suporte à configuração moderna, você poderá atualizá-lo ou escolher a configuração "intermediária".

```
SSLCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-  
ECDSA-CHACHA20-POLY1305:  
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-  
SHA256:  
ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-  
RSA-AES128-SHA256
```

As cifras selecionadas têm ECDHE em seus nomes, o que significa Elliptic Curve Diffie-Hellman Ephemeral (Curva elíptica de Diffie-Hellman efêmera). O termo ephemeral (efêmera) indica forward secrecy. Como subproduto, essas cifras não são compatíveis. RC4

Recomendamos que você use uma lista explícita de cifras em vez de confiar em padrões ou em diretrizes concisas cujo conteúdo não é visível.

Copie a diretiva gerada em `/etc/httpd/conf.d/ssl.conf`.

Note

Embora sejam mostradas em várias linhas aqui para facilitar a leitura, a diretiva deve estar em uma única linha quando copiada para `/etc/httpd/conf.d/ssl.conf` com apenas dois pontos (sem espaços) entre os nomes das cifras.

3. Por fim, remova o comentário da linha a seguir, excluindo o `"#"` no início dela.

```
#SSLHonorCipherOrder on
```

Essa diretiva força o servidor a preferir cifras de alta classificação incluindo (neste caso) aquelas que oferecem suporte a forward secrecy. Com essa diretiva ativada, o servidor tenta estabelecer uma conexão altamente segura antes de voltar a usar cifras permitidas com menos segurança.

Depois de concluir esses dois procedimentos, salve as alterações em `/etc/httpd/conf.d/ssl.conf` e reinicie o Apache.

Se você testar o domínio novamente no [Qualys SSL Labs](#), verá que a RC4 vulnerabilidade e outros avisos desapareceram e que o resumo se parece com o seguinte.

Classificação geral

A

Certificado	100%
Suporte ao protocolo	100%
Troca de chaves	90%
Intensidade da cifra	90%

Cada atualização do OpenSSL apresenta novas cifras e retira o suporte às cifras antigas. Mantenha sua AL2023 instância EC2 up-to-date, fique atento aos anúncios de segurança do [OpenSSL](#) e fique atento às denúncias de novas falhas de segurança na imprensa técnica.

Solução de problemas

- Meu servidor da web do Apache não inicia, a menos que eu digite uma senha.

Esse é comportamento esperado se você tiver instalado uma chave privada de servidor criptografada e protegida por senha.

Você pode remover a criptografia e a solicitação de senha da chave. Supondo que você tenha uma chave de RSA privada criptografada chamada `custom.key` no diretório padrão, e que a senha seja **abcde12345**, execute os comandos a seguir na sua instância do EC2 para gerar uma versão descriptografada da chave:

```
[ec2-user ~]$ cd /etc/pki/tls/private/
[ec2-user private]$ sudo cp custom.key custom.key.bak
[ec2-user private]$ sudo openssl rsa -in custom.key -passin pass:abcde12345 -out
custom.key.nocrypt
[ec2-user private]$ sudo mv custom.key.nocrypt custom.key
[ec2-user private]$ sudo chown root:root custom.key
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ sudo systemctl restart httpd
```

O Apache agora deve iniciar sem solicitar uma senha a você.

- Obtenho erros ao executar `sudo dnf install -y mod_ssl`.

Quando estiver instalando os pacotes necessários para SSL, você verá erros como os exibidos a seguir.

```
Error: httpd24-tools conflicts with httpd-tools-2.2.34-1.16.amzn1.x86_64
Error: httpd24 conflicts with httpd-2.2.34-1.16.amzn1.x86_64
```

Isso normalmente significa que sua instância do EC2 não está em execução AL2023. Este tutorial comporta somente instâncias recentemente criadas em uma AMI oficial do AL2023.

Tutorial: hospede um WordPress blog no AL2023

Os procedimentos a seguir ajudarão você a instalar, configurar e proteger um WordPress blog na sua instância do AL2023. Este tutorial é uma boa introdução ao uso do Amazon EC2, pois você tem controle total sobre um servidor web que hospeda seu WordPress blog, o que não é típico de um serviço de hospedagem tradicional.

Você é responsável para atualizar os pacotes de software e manter os patches de segurança para seu servidor. Para uma WordPress instalação mais automatizada que não exija interação direta com a configuração do servidor web, o CloudFormation serviço fornece um WordPress modelo que também pode ajudar você a começar rapidamente. Para mais informações, consulte [Get started](#) (Conceitos básicos) no AWS CloudFormation User Guide (Guia do usuário do). Se você precisar de uma solução de alta disponibilidade com um banco de dados desacoplado, consulte [Implantação de um WordPress site de alta disponibilidade](#) no Guia do desenvolvedor.AWS Elastic Beanstalk

Important

Esses procedimentos são destinados ao uso com AL2023. Para obter informações sobre outras distribuições, consulte a documentação específica. Muitas etapas deste tutorial não funcionam em instâncias Ubuntu. Para obter ajuda WordPress na instalação em uma instância do Ubuntu, consulte [WordPress](#) documentação do Ubuntu. Você também pode usar [CodeDeploy](#) para realizar essa tarefa nos sistemas Amazon Linux, macOS ou Unix.

Tópicos

- [Pré-requisitos](#)
- [Instalar WordPress](#)
- [Próximas etapas](#)
- [Ajuda! Meu nome DNS público mudou e agora meu blog não está funcionando](#)

Pré-requisitos

É altamente recomendável que você associe um endereço IP elástico (EIP) à instância que você está usando para hospedar um WordPress blog. Isso impede que o endereço DNS público da sua instância mude e quebre sua instalação. Se você tiver um nome de domínio e quiser usá-lo para o blog, pode atualizar o registro DNS do nome de domínio para indicar ao seu endereço EIP (para obter ajuda com isso, contate seu registrador de nome de domínio). Você pode ter um endereço EIP associado a uma instância em execução, gratuitamente. Para obter mais informações, consulte [Endereços IP elásticos](#) no Guia do usuário do Amazon EC2. O tutorial [Tutorial: Instalar um servidor LAMP em AL2023](#) apresenta etapas para configurar um grupo de segurança para permitir tráfego de HTTP e HTTPS, bem como várias etapas para garantir que as permissões de arquivos sejam definidas corretamente para seu servidor da Web. Para obter informações sobre como adicionar regras aos grupos de segurança, consulte [Add rules to a security group](#).

Se você ainda não tiver um nome de domínio para seu blog, pode registrar um nome de domínio com o Route 53 e associar o endereço EIP de sua instância com seu nome de domínio. Para obter mais informações, consulte [Registrar nomes de domínio usando o Amazon Route 53](#) no Guia do desenvolvedor do Amazon Route 53.

Instalar WordPress

Conecte-se à sua instância e baixe o pacote WordPress de instalação. Para obter mais informações sobre como se conectar à sua instância, consulte [Conexão com AL2023 instâncias](#).

1. Baixe e instale esses pacotes usando o comando a seguir.

```
dnf install wget php-mysqlnd httpd php-fpm php-mysqli mariadb105-server php-json  
php php-devel -y
```

2. Você pode notar um aviso exibido com verbiagem semelhante na saída (as versões podem variar com o tempo):

```
WARNING:  
  A newer release of "Amazon Linux" is available.  
  
  Available Versions:  
  
dnf upgrade --releasever=2023.0.20230202  
  
  Release notes:
```

```
https://aws.amazon.com
```

Version 2023.0.20230204:

Run the following command to update to 2023.0.20230204:

```
dnf upgrade --releasever=2023.0.20230204 ... etc
```

Como prática recomendada, recomendamos manter o sistema operacional o mais up-to-date possível, mas talvez você queira repetir cada versão para garantir que não haja conflitos em seu ambiente. Se a instalação dos pacotes anteriores anotados na etapa 1 falhar, talvez seja necessário atualizar para uma das versões mais recentes listadas e tentar novamente.

3. Baixe o pacote WordPress de instalação mais recente com o `wget` comando. O comando a seguir sempre deve baixar a versão mais recente.

```
[ec2-user ~]$ wget https://wordpress.org/latest.tar.gz
```

4. Descompacte e desarchive o pacote de instalação. A pasta de instalação é descompactada para uma pasta chamada `wordpress`.

```
[ec2-user ~]$ tar -xzf latest.tar.gz
```

Para criar um usuário de banco de dados e um banco de dados para sua WordPress instalação

Sua WordPress instalação precisa armazenar informações, como postagens de blog e comentários de usuários, em um banco de dados. Esse procedimento ajuda você a criar um banco de dados para seu blog e um usuário autorizado a ler e salvar as informações.

1. Inicie o servidor do banco de dados e da Web.

```
[ec2-user ~]$ sudo systemctl start mariadb httpd
```

2. Faça login no servidor do banco de dados como usuário `root`. Insira a senha de `root` do banco de dados quando solicitado; ela poderá ser diferente da sua senha do sistema de `root` ou poderá até estar vazia, se você não tiver protegido seu servidor do banco de dados.

Se ainda não tiver protegido seu servidor do banco de dados, é muito importante que você faça isso. Para obter mais informações, consulte [Etapa 3: Proteger o servidor do banco de dados \(AL2023\)](#).


```
[ec2-user ~]$ mysql -u root -p
```

3. Crie um usuário e uma senha para seu banco de dados do MySQL. Sua WordPress instalação usa esses valores para se comunicar com seu banco de dados MySQL. Digite o comando a seguir, substituindo um nome de usuário e uma senha exclusivos.

```
CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'your_strong_password';
```

Crie uma senha forte para seu usuário. Não use o caractere de aspa única (') na sua senha, pois isso quebrará o comando anterior. Não reutilize uma senha existente e armazene essa senha em um lugar seguro.

4. Crie seu banco de dados. Dê ao seu banco de dados um nome descritivo e significativo, como `wordpress-db`.

Note

As marcas de pontuação que cercam o nome do banco de dados no comando abaixo são chamados backticks. A chave de backtick (`) costuma estar localizada acima da chave Tab de um teclado padrão. Backticks nem sempre são necessários, mas permitem que você use caracteres de outra forma ilegais, como hífen, no nome dos bancos de dados.

```
CREATE DATABASE `wordpress-db`;
```

5. Conceda privilégios totais do seu banco de dados ao WordPress usuário que você criou anteriormente.

```
GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpress-user"@"localhost";
```

6. Limpe os privilégios do banco de dados para receber todas as suas alterações.

```
FLUSH PRIVILEGES;
```

7. Saia do cliente `mysql`.

```
exit
```

Para criar e editar o arquivo wp-config.php

A pasta WordPress de instalação contém um exemplo de arquivo de configuração chamado wp-config-sample.php. Nesse procedimento, você copia esse arquivo e o edita para caber na sua configuração específica.

1. Copie o arquivo wp-config-sample.php para um arquivo chamado wp-config.php. Isso cria um novo arquivo de configuração e mantém o arquivo de exemplo original intacto como um backup.

```
[ec2-user ~]$ cp wordpress/wp-config-sample.php wordpress/wp-config.php
```

2. Edite o arquivo wp-config.php com seu editor de texto favorito (como o nano ou o vim) e insira os valores da instalação. Se você não tiver um editor de texto favorito, o nano é ideal para iniciantes.

```
[ec2-user ~]$ nano wordpress/wp-config.php
```

- a. Encontre a linha que define DB_NAME e altere database_name_here para o nome do banco de dados criado em [Step 4](#) de [Para criar um usuário de banco de dados e um banco de dados para sua WordPress instalação](#).

```
define('DB_NAME', 'wordpress-db');
```

- b. Encontre a linha que define DB_USER e altere username_here para o usuário do banco de dados que você criou [Step 3](#) de [Para criar um usuário de banco de dados e um banco de dados para sua WordPress instalação](#).

```
define('DB_USER', 'wordpress-user');
```

- c. Encontre a linha que define DB_PASSWORD e altere password_here para a senha mais forte que você criou em [Step 3](#) de [Para criar um usuário de banco de dados e um banco de dados para sua WordPress instalação](#).

```
define('DB_PASSWORD', 'your_strong_password');
```

- d. Encontre a seção chamada Authentication Unique Keys and Salts. Esses SALT valores KEY e esses fornecem uma camada de criptografia aos cookies do navegador que WordPress os usuários armazenam em suas máquinas locais. Basicamente, adicionar

valores longos e aleatórios aqui deixa seu site mais seguro. Visite <https://api.wordpress.org/secret-key/1.1/salt/> para gerar aleatoriamente um conjunto de valores-chave que você pode copiar e colar em seu `wp-config.php` arquivo. Para colar texto em um terminal do PuTTY, coloque o cursor onde deseja colar texto e clique com o botão direito do mouse dentro do terminal do PuTTY.

Para obter mais informações sobre chaves de segurança, acesse <https://wordpress.org/support/article/editing-wp-config-php/#security-keys>.

Note

Os valores abaixo são somente para fins de exemplo; não use esses valores para a instalação.

```
define('AUTH_KEY',          ' #U$${[RXN8:b^-L 0(WU_+ c+WFkI~c]o]-bHw+)}Aj[wTwSiZ<Qb[mghEXcRh-');
define('SECURE_AUTH_KEY',  'Zsz._P=l/|y.Lq)XjlkW51y5NJ76E6EJ.AV0pCKZZB,*~*r ?60P$eJT@;+(ndLg');
define('LOGGED_IN_KEY',    'ju}qwre3V*+8f_z0Wf?{LlGsQ]Ye@2Jh^,8x>)Y |;(^[Iw]Pi+LG#A4R?7N`YB3');
define('NONCE_KEY',        'P(g62HeZxEes|LnI^i=H,[XwK9I&[2s|: ?0N}VJM?;v2v]v+;+^9eXUahg@::Cj');
define('AUTH_SALT',        'C$DpB4Hj[JK: ?{ql`sRVa: { :7yShy(9A@5wg+`JJVb1fk%-Bx*M4(qc[Qg%JT!h');
define('SECURE_AUTH_SALT', 'd!uRu#}+q#{f$Z?Z9uFPG.$ {+S{n~1M&%@~gL>U>NV<zpD-@2-Es7Q10-bp28EKv');
define('LOGGED_IN_SALT',   ';j{00P*owZf)kVD+FVLn-~ >.|Y%Ug4#I^*LVd9QeZ^&XmK|e(76miC+&W&+^0P/');
define('NONCE_SALT',       '-97r*V/cgxLmp?Zy4zUU4r99QQ_rGs2LTd%P;|_e1tS)8_B/, .6[=UK<J_y9?JWG');
```

e. Salve o arquivo e saia do seu editor de texto.

Para instalar seus WordPress arquivos na raiz do documento Apache

- Agora que você descompactou a pasta de instalação, criou um banco de dados e um usuário MySQL e personalizou o arquivo de WordPress configuração, você está pronto para copiar os arquivos de instalação para a raiz do documento do servidor web para poder executar o script de instalação que conclui a instalação. A localização desses arquivos depende se

you want your WordPress blog to be available at the real root of your web server (for example, `my.public.dns.amazonaws.com`) or in a subdirectory or folder below the root (for example, `my.public.dns.amazonaws.com/blog`).

- If you want WordPress to run at the root of your document, copy the contents of the WordPress installation directory (but not the directory itself) in the following form:

```
[ec2-user ~]$ cp -r wordpress/* /var/www/html/
```

- If you want WordPress to run in an alternative directory at the root of the document, first create that directory and, next, copy the files into it. In this example, WordPress will be executed from the `blog` directory:

```
[ec2-user ~]$ mkdir /var/www/html/blog  
[ec2-user ~]$ cp -r wordpress/* /var/www/html/blog/
```

Important

For security reasons, if you are not following the next procedure immediately, stop the Apache Web Server (`httpd`) now. After moving your installation to the root of the Apache document, the WordPress installation script becomes unprotected and an intruder can obtain access to your blog if the Apache web server is running. To stop the Apache web server, enter the command `sudo service httpd stop`. If you are moving to the next procedure, you do not need to stop the Apache Web Server.

Para permitir o uso WordPress de links permanentes

WordPress os permalinks precisam usar `.htaccess` arquivos Apache para funcionarem corretamente, mas isso não está habilitado por padrão no Amazon Linux. Use o procedimento a seguir para permitir todas as substituições na raiz de documentos do Apache.

1. Abra o arquivo `httpd.conf` com seu editor de texto de preferência (como `nano` ou `vim`). Se você não tiver um editor de texto favorito, o `nano` é ideal para iniciantes.

```
[ec2-user ~]$ sudo vim /etc/httpd/conf/httpd.conf
```

2. Encontre a seção que começa com `<Directory "/var/www/html">`.

```
<Directory "/var/www/html">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
AllowOverride None

#
# Controls who can get stuff from this server.
#
Require all granted
</Directory>
```

3. Altere a linha `AllowOverride None` na seção acima para `AllowOverride All`.

 Note

Há múltiplas linhas `AllowOverride` nesse arquivo; altere a linha na seção `<Directory "/var/www/html">`.

```
AllowOverride All
```

4. Salve o arquivo e saia do seu editor de texto.

Para instalar a biblioteca de desenhos gráficos PHP em AL2023

A biblioteca de desenhos gráficos para PHP permite modificar imagens. Instale esta biblioteca caso você precise cortar a imagem do cabeçalho do blog. A versão phpMyAdmin que você instala pode exigir uma versão mínima específica dessa biblioteca (por exemplo, versão 8.1).

Use o comando a seguir para instalar a biblioteca de desenhos gráficos PHP em AL2023. Por exemplo, se você instalou o php8.1 da origem como parte da instalação da pilha LAMP, este comando instalará a versão 8.1 da biblioteca de desenhos gráficos PHP.

```
[ec2-user ~]$ sudo dnf install php-gd
```

Para verificar a versão instalada, use o seguinte comando:

```
[ec2-user ~]$ sudo dnf list installed | grep php-gd
```

A seguir está um exemplo de saída:

php-gd.x86_64	8.1.30-1.amzn2	@amazonlinux

Para instalar a biblioteca de desenhos gráficos PHP no Amazon Linux AMI

A biblioteca de desenhos gráficos para PHP permite modificar imagens. Instale esta biblioteca caso você precise cortar a imagem do cabeçalho do blog. A versão phpMyAdmin que você instala pode exigir uma versão mínima específica dessa biblioteca (por exemplo, versão 8.1).

Para verificar quais versões estão disponíveis, use o seguinte comando:

```
[ec2-user ~]$ dnf list | grep php
```

A seguir são mostradas linhas de exemplo da saída para a biblioteca de desenhos gráficos PHP (versão 8.1):

php8.1.aarch64	8.1.7-1.amzn2023.0.1
@amazonlinux	
php8.1-cli.aarch64	8.1.7-1.amzn2023.0.1
@amazonlinux	
php8.1-common.aarch64	8.1.7-1.amzn2023.0.1
@amazonlinux	

php8.1-devel.aarch64	8.1.7-1.amzn2023.0.1
@amazonlinux	
php8.1-fpm.aarch64	8.1.7-1.amzn2023.0.1
@amazonlinux	
php8.1-gd.aarch64	8.1.7-1.amzn2023.0.1
@amazonlinux	

Use o comando a seguir para instalar uma versão específica da biblioteca de desenhos gráficos PHP (por exemplo, php8.1) no Amazon Linux AMI:

```
[ec2-user ~]$ sudo dnf install -y php8.1-gd
```

Para corrigir as permissões de arquivos para o Apache Web Server

Alguns dos recursos disponíveis WordPress exigem acesso de gravação à raiz do documento Apache (como o upload de mídia pelas telas de administração). Se você não tiver feito isso, aplique as associações e permissões de grupo a seguir (conforme descrito em mais detalhes no [tutorial do servidor web LAMP](#)).

1. Conceda a propriedade do arquivo de `/var/www` e seu conteúdo para o usuário apache.

```
[ec2-user ~]$ sudo chown -R apache /var/www
```

2. Conceda a propriedade do grupo do `/var/www` e seu conteúdo para o grupo do apache.

```
[ec2-user ~]$ sudo chgrp -R apache /var/www
```

3. Altere as permissões do diretório do `/var/www` e de seus subdiretórios para adicionar permissões de gravação do grupo e definir o ID do grupo em subdiretórios futuros.

```
[ec2-user ~]$ sudo chmod 2775 /var/www
[ec2-user ~]$ find /var/www -type d -exec sudo chmod 2775 {} \;
```

4. Altere recursivamente as permissões de arquivo de `/var/www` e de seus subdiretórios.

```
[ec2-user ~]$ find /var/www -type f -exec sudo chmod 0644 {} \;
```

 Note

Se você também pretende usar WordPress como servidor FTP, precisará de configurações de grupo mais permissivas aqui. Revise as [etapas recomendadas e as configurações de segurança WordPress](#) para fazer isso.

5. Reinicie o Apache Web Server para pegar o grupo e as permissões novos.

```
[ec2-user ~]$ sudo systemctl restart httpd
```

Para executar o script WordPress de instalação com AL2023

Você está pronto para instalar WordPress. Os comandos usados por você dependem do sistema operacional. Os comandos neste procedimento são para uso com AL2023. Use o procedimento que segue este com a AL2023 AMI.

1. Use o comando `systemctl` para garantir que `httpd` e os serviços do banco de dados sejam iniciados a cada inicialização do sistema.

```
[ec2-user ~]$ sudo systemctl enable httpd && sudo systemctl enable mariadb
```

2. Verifique se o servidor do banco de dados está em execução.

```
[ec2-user ~]$ sudo systemctl status mariadb
```

Se o serviço do banco de dados não está em execução, inicie-o.

```
[ec2-user ~]$ sudo systemctl start mariadb
```

3. Verifique se o Apache Web Server (`httpd`) está sendo executado.

```
[ec2-user ~]$ sudo systemctl status httpd
```

Se o serviço `httpd` não estiver sendo executado, inicie-o.

```
[ec2-user ~]$ sudo systemctl start httpd
```


4. Em um navegador da Web, digite a URL do seu WordPress blog (o endereço DNS público da sua instância ou o endereço seguido pela `blog` pasta). Você deve ver o script WordPress de instalação. Forneça as informações exigidas pela WordPress instalação. Escolha Instalar WordPress para concluir a instalação. Para obter mais informações, consulte [Etapa 5: Executar o script de instalação](#) no WordPress site.

Para executar o script WordPress de instalação com a AL2023 AMI

1. Use o comando `chkconfig` para garantir que `httpd` e os serviços do banco de dados sejam iniciados a cada inicialização do sistema.

```
[ec2-user ~]$ sudo chkconfig httpd on && sudo chkconfig mariadb on
```

2. Verifique se o servidor do banco de dados está em execução.

```
[ec2-user ~]$ sudo service mariadb status
```

Se o serviço do banco de dados não está em execução, inicie-o.

```
[ec2-user ~]$ sudo service mariadb start
```

3. Verifique se o Apache Web Server (`httpd`) está sendo executado.

```
[ec2-user ~]$ sudo service httpd status
```

Se o serviço `httpd` não estiver sendo executado, inicie-o.

```
[ec2-user ~]$ sudo service httpd start
```

4. Em um navegador da Web, digite a URL do seu WordPress blog (o endereço DNS público da sua instância ou o endereço seguido pela `blog` pasta). Você deve ver o script WordPress de instalação. Forneça as informações exigidas pela WordPress instalação. Escolha Instalar WordPress para concluir a instalação. Para obter mais informações, consulte [Etapa 5: Executar o script de instalação](#) no WordPress site.

Próximas etapas

Depois de testar seu WordPress blog, considere atualizar sua configuração.

Usar um nome de domínio personalizado

Se você tiver um nome de domínio associado ao endereço EIP da sua instância do EC2, pode configurar o blog para usar esse nome em vez do endereço DNS público do EC2. Para obter mais informações, consulte [Alterando a URL do WordPress site](#) no site.

Configurar seu blog

Você pode configurar seu blog para usar diferentes [temas](#) e [plug-ins](#) e oferecer uma experiência mais personalizada para seus leitores. Contudo, às vezes o processo de instalação pode dar errado, fazendo com que você perca o blog inteiro. Recomendamos veementemente que você crie um backup da imagem de máquina da Amazon (AMI) de sua instância antes de tentar instalar quaisquer temas ou plug-ins, de forma que consiga restaurar o blog se algo der errado durante a instalação. Para saber mais, consulte [Create your own AMI](#) no Guia do usuário do Amazon EC2.

Aumentar a capacidade

Se seu WordPress blog se tornar popular e você precisar de mais capacidade computacional ou armazenamento, considere as seguintes etapas:

- Expanda o espaço de armazenamento na sua instância. Para saber mais, consulte [Volumes Elásticos do Amazon EBS](#).
- Mova o banco de dados MySQL para o [Amazon RDS](#) para aproveitar a capacidade de dimensionamento que o serviço oferece.

Melhore a performance de rede do tráfego da Internet

Se você espera que seu blog gere tráfego de usuários localizados em todo o mundo, considere o [AWS Global Accelerator](#). O Global Accelerator ajuda você a obter menor latência melhorando o desempenho do tráfego da Internet entre os dispositivos cliente de seus usuários e seu WordPress aplicativo em execução. AWS O Global Accelerator usa a [rede AWS global](#) para direcionar o tráfego para um endpoint de aplicativo saudável na AWS região mais próxima do cliente.

Saiba mais sobre WordPress

Os links a seguir contêm mais informações sobre WordPress.

- Para obter informações sobre WordPress, consulte a documentação de ajuda do WordPress Codex no [Codex](#).

- Para saber mais sobre como solucionar problemas de instalação, acesse [Common installatino problems](#).
- Para obter informações sobre como tornar seu WordPress blog mais seguro, acesse [Fortalecimento WordPress](#).
- Para obter informações sobre como manter seu WordPress blog up-to-date, acesse [Atualizar WordPress](#).

Ajuda! Meu nome DNS público mudou e agora meu blog não está funcionando

Sua WordPress instalação é configurada automaticamente usando o endereço DNS público da sua instância EC2. Se você parar e reiniciar a instância, as alterações no endereço DNS público (a menos que estejam associadas a um endereço IP elástico) e seu blog não funcionarão mais, pois ele faz referência a recursos em um endereço que não existe mais (ou é atribuído a outra instância do EC2). Uma descrição mais detalhada do problema e várias soluções possíveis estão descritas em <https://wordpress.org/support/article/changing-the-site-url/>.

Se isso aconteceu com sua WordPress instalação, talvez você consiga recuperar seu blog com o procedimento abaixo, que usa a interface de linha de wp-cli comando para WordPress.

Para alterar o URL WordPress do seu site com o wp-cli

1. Conecte-se à sua instância do EC2 com SSH.
2. Anote o URL do site antigo e do site novo para sua instância. O URL antigo do site provavelmente é o nome DNS público da sua instância do EC2 quando você instalou. WordPress O URL do novo site é o nome DNS público atual da sua instância do EC2. Se você não tiver certeza da URL do site antigo, pode usar o curl para encontrá-la com o seguinte comando.

```
[ec2-user ~]$ curl localhost | grep wp-content
```

Você deve visualizar referências ao nome DNS público antigo na saída, que terá a seguinte aparência (URL do site antigo em vermelho):

```
<script type='text/javascript' src='http://ec2-52-8-139-223.us-west-1.compute.amazonaws.com/wp-content/themes/twentyfifteen/js/functions.js?ver=20150330'></script>
```

3. Faça download do wp-cli com o seguinte comando.

```
[ec2-user ~]$ curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
```

4. Pesquise e substitua o URL antigo do site em sua WordPress instalação pelo comando a seguir. Substitua os URLs antigos e novos do site para sua instância do EC2 e o caminho para sua WordPress instalação (geralmente `/var/www/html` ou `/var/www/html/blog`).

```
[ec2-user ~]$ php wp-cli.phar search-replace 'old_site_url' 'new_site_url' --path=/path/to/wordpress/installation --skip-columns=guid
```

5. Em um navegador da Web, insira a nova URL do seu WordPress blog para verificar se o site está funcionando corretamente novamente. Se não estiver, consulte [Changing the site URL](#) e [Common installation problems](#) para obter mais informações.

Tutorial: Transição do Redis 6 para o Valkey em 023 AL2

A documentação a seguir descreve os principais aspectos da transição do Redis 6 para o Valkey em AL2 023.

Cronograma de suporte para o Redis 6

O Redis 6 chega ao fim da vida útil (EOL) em 31 de janeiro de 2027. Após essa data, o Redis 6 não receberá mais atualizações nem patches de segurança do projeto Redis. É altamente recomendável que os usuários migrem para o Valkey antes de janeiro de 2027 para garantir suporte contínuo e atualizações de segurança.

Para obter mais informações sobre os cronogramas de suporte da versão do Redis, consulte a documentação do [Redis End-Of-Life](#) Schedule.

Introdução ao Valkey

O Valkey é um fork de código aberto do Redis 7, mantido pela The Linux Foundation. É totalmente compatível com as versões 2.x a 7.2.x do Redis Open Source Software (OSS). O Valkey mantém a API e a funcionalidade conhecidas do Redis, oferecendo vários aprimoramentos:

- Desempenho aprimorado por meio de multisegmentação.
- Maior eficiência de memória, especialmente no modo de cluster.

- Replicação de canal duplo para melhor consistência de dados.

Plano e cronograma de migração

É altamente recomendável que os usuários migrem do Redis 6 para o Valkey antes de 31 de janeiro de 2027, quando o Redis 6 chegar ao fim da vida útil (EOL). Essa migração requer intervenção manual e não é automática.

O Amazon Linux recomenda essa migração para garantir atualizações contínuas de funcionalidade, suporte e segurança para aplicações dependentes do Redis.

Opções e etapas de migração

Propomos três caminhos de migração para o Valkey com base nos seus requisitos de implantação e necessidades operacionais.

Opção 1: instalação de nova instância

Para novas implantações ou quando a migração de dados não for necessária:

1. Instale o Valkey:

```
[ec2-user ~]$ sudo dnf install valkey
```

2. Inicie o Valkey:

```
[ec2-user ~]$ sudo systemctl start valkey
```

3. (Opcional) Habilite o Valkey na inicialização:

```
[ec2-user ~]$ sudo systemctl enable valkey
```

4. Verifique a instalação:

```
[ec2-user ~]$ valkey-cli info server  
[ec2-user ~]$ valkey-cli ping
```

Opção 2: substituição no local

Para instâncias existentes em que a persistência de dados não é necessária:

1. Interrompa o Redis 6:

```
[ec2-user ~]$ sudo systemctl stop redis6
```

2. Instale o Valkey:

```
[ec2-user ~]$ sudo dnf install valkey
```

3. (Opcional) Use a configuração do Redis 6 no Valkey:

```
[ec2-user ~]$ sudo cp /etc/redis6/redis6.conf /etc/valkey/valkey.conf  
[ec2-user ~]$ sudo cp /etc/valkey/valkey.conf /etc/valkey/valkey.conf.backup  
[ec2-user ~]$ sudo chown valkey:root /etc/valkey/valkey.conf  
[ec2-user ~]$ sudo sed -i 's|^dir\s.*|dir /var/lib/valkey|g' /etc/valkey/  
valkey.conf
```

4. (Opcional) Use o arquivo de configuração sentinela do Redis 6 no Valkey:

```
[ec2-user ~]$ sudo cp /etc/redis6/sentinel.conf /etc/valkey/sentinel.conf  
[ec2-user ~]$ sudo chown valkey:root /etc/valkey/sentinel.conf
```

5. Inicie o Valkey:

```
[ec2-user ~]$ sudo systemctl start valkey
```

6. (Opcional) Habilite o Valkey na inicialização:

```
[ec2-user ~]$ sudo systemctl enable valkey
```

7. Verifique a instalação do Valkey:

```
[ec2-user ~]$ valkey-cli info server  
[ec2-user ~]$ valkey-cli ping
```

8. Remova o Redis 6:

```
[ec2-user ~]$ sudo dnf remove redis6
```

Opção 3: migração de dados

Essa opção permite que você execute o Redis 6 e o Valkey simultaneamente.

1. Instale o Valkey sem remover o Redis 6:

```
[ec2-user ~]$ sudo dnf install valkey
```

2. (Opcional) Use a configuração do Redis 6 no Valkey:

```
[ec2-user ~]$ sudo cp /etc/redis6/redis6.conf /etc/valkey/valkey.conf
[ec2-user ~]$ sudo cp /etc/valkey/valkey.conf /etc/valkey/valkey.conf.backup
[ec2-user ~]$ sudo chown valkey:root /etc/valkey/valkey.conf
[ec2-user ~]$ sudo sed -i 's|^dir\s.*|dir /var/lib/valkey|g' /etc/valkey/
valkey.conf
```

3. (Opcional) Use o arquivo de configuração sentinela do Redis 6 no Valkey:

```
[ec2-user ~]$ sudo cp /etc/redis6/sentinel.conf /etc/valkey/sentinel.conf
[ec2-user ~]$ sudo chown valkey:root /etc/valkey/sentinel.conf
```

4. Modifique a configuração do Valkey:

Edite `/etc/valkey/valkey.conf` e defina a diretiva “port” com um valor diferente (por exemplo, 6380) para evitar conflitos com o Redis 6.

5. Inicie o Valkey:

```
[ec2-user ~]$ sudo systemctl start valkey
```

6. (Opcional) Habilite o Valkey na inicialização:

```
[ec2-user ~]$ sudo systemctl enable valkey
```

7. Verifique a instalação do Valkey:

```
[ec2-user ~]$ valkey-cli -p port info server
[ec2-user ~]$ valkey-cli -p port ping
```

Note

port Substitua pelo número da porta configurada.

8. Migre os dados:

Agora você pode migrar dados do Redis 6 para o Valkey usando métodos de replicação ou transferência manual de dados.

9. Atualize as configurações das aplicações:

Atualize gradualmente as aplicações para usar a porta do Valkey.

10. Remova o Redis 6:

Depois que todos os dados e aplicações tiverem sido migrados, você poderá interromper e remover o Redis 6.

```
[ec2-user ~]$ sudo systemctl stop redis6  
[ec2-user ~]$ sudo dnf remove redis6
```

 Note

É altamente recomendável validar o processo de migração em um ambiente de teste antes de implementar mudanças nos sistemas de produção.

Tópicos relacionados

Para saber mais sobre o Valkey:

- Valkey: <https://valkey.io/>
- Migração do Vale: <https://valkey.io/topics/migration/>

Tutorial: Instale o ambiente de trabalho GNOME em AL2023

O [ambiente de trabalho GNOME](#) está disponível como uma interface gráfica de usuário opcional a AL2023 partir da versão 2023.7 ou posterior.

Os procedimentos a seguir ajudam você a instalar o ambiente de desktop GNOME na sua AL2023 instância. Você pode usar essa interface gráfica para interagir com o sistema Linux usando um ambiente de área de trabalho familiar em vez de apenas a interface de linha de comandos.

Conteúdos

- [Pré-requisitos](#)
- [Instalação](#)
- [Tópicos relacionados](#)

Pré-requisitos

- O ambiente de área de trabalho requer pelo menos 2,4 GB de memória. Portanto, uma instância do tipo `t2.medium` ou melhor é recomendada para garantir um desempenho adequado. Exemplos de tipos de instância com memória insuficiente incluem `t2.nano`, `t2.micro` e `t2.small`. Essa restrição também se aplica a instâncias `t3` e `t4` desse tamanho, bem como a qualquer outro tipo de instância que não atenda aos requisitos de memória.
- Este tutorial pressupõe que você já tenha AL2023 executado uma instância usando a versão 2023.7 ou posterior. Para saber mais, consulte as páginas [AL2023 no Amazon EC2](#) e [Atualizando AL2023](#).

Instalação

- Instale o ambiente de área de trabalho GNOME e os pacotes relacionados.

```
[ec2-user ~]$ sudo dnf groupinstall "Desktop" -y
```

Note

Para acessar o ambiente gráfico de área de trabalho, você precisará instalar e configurar software adicional, como o Amazon DCV ou VNC. Essas ferramentas permitem que você se conecte e interaja com a interface gráfica de usuário na rede.

Tópicos relacionados

Para saber mais sobre o ambiente gráfico de área de trabalho, consulte a seguinte documentação:

- [What Is Amazon DCV?](#) no Guia do administrador do Amazon DCV
- [Tutorial: Configurar o servidor TigerVNC em AL2023](#)

Tutorial: Configurar o servidor TigerVNC em AL2023

Os procedimentos a seguir ajudam você a configurar o servidor VNC na sua AL2023 instância. A VNC permite que você acesse e interaja remotamente com o ambiente gráfico de área de trabalho por meio de uma conexão de rede segura.

Conteúdos

- [Pré-requisitos](#)
- [Etapa 1: instalação](#)
- [Etapa 2: configuração](#)
- [Etapa 3: conectar-se usando um cliente VNC](#)
- [\(Opcional\) Iniciar o serviço na inicialização](#)
- [\(Opcional\) Desabilitar a tela de bloqueio de inatividade](#)
- [Tópicos relacionados](#)

Pré-requisitos

- Este tutorial pressupõe que você já tenha instalado o ambiente de desktop GNOME na sua AL2023 instância. Para obter mais informações, consulte a página do [Tutorial: Instale o ambiente de trabalho GNOME em AL2023](#).
- Este tutorial usa o encaminhamento de portas SSH para acessar o servidor de VNC. Para saber mais sobre como configurar um par de chaves, consulte [Conectar-se à instância do Linux usando SSH](#) no Guia do usuário do Amazon EC2.
- O procedimento a seguir não apresenta o processo de instalação de um cliente VNC. Você deve ter um cliente VNC instalado em sua máquina local para poder se conectar e interagir com o ambiente de área de trabalho.

Etapa 1: instalação

1. Conecte-se à sua instância. Para obter mais informações, consulte [Conexão com AL2023 instâncias](#).
2. Instale o pacote do servidor TigerVNC para AL2023

A opção `-y` instala o pacote sem solicitar confirmação. Para examinar o pacote antes da instalação, você pode omitir essa opção.

```
[ec2-user ~]$ sudo dnf install -y tigervnc-server
```

Etapa 2: configuração

1. Verifique se o usuário configurou uma senha de VNC.

```
[ec2-user ~]$ vncpasswd
```

2. Atribua um número de exibição ao usuário.

```
[ec2-user ~]$ sudo vi /etc/tigervnc/vncserver.users
```

Adicione a seguinte configuração:

```
:1=ec2-user
```

Note

Você pode atribuir qualquer número de exibição ao usuário. Usaremos `:1` para os fins deste exemplo.

3. Edite o arquivo de configuração do servidor de VNC.

```
[ec2-user ~]$ sudo vi /etc/tigervnc/vncserver-config-defaults
```

Adicione a seguinte configuração:

```
session=gnome
securitytypes=vncauth,tlsvnc
geometry=1920x1080
localhost
alwaysshared
```

 Note

Você pode alterar a resolução da tela usando o parâmetro `geometry`. Usaremos `1920x1080` para os fins deste exemplo.

4. Inicie o servidor de VNC. Esse processo precisará ser repetido toda vez que você reiniciar a instância. Se quiser automatizar o processo de inicialização desse serviço, consulte a seção opcional abaixo.

```
[ec2-user ~]$ sudo systemctl start vncserver@:1
```

 Important

Ao iniciar o serviço `vncserver`, a parcela depois de `@` deve corresponder ao número de exibição definido para o usuário no arquivo `/etc/tigervnc/vncserver.users`.

Depois de executar essa etapa, você pode criar o túnel SSH por meio da sua máquina local e se conectar usando seu cliente VNC.

Etapa 3: conectar-se usando um cliente VNC

O servidor de VNC expõe um soquete TCP para conexões de clientes. Embora você possa expor a porta de VNC diretamente por meio do seu grupo de segurança, este tutorial demonstra o uso da criação de túneis SSH como uma abordagem mais segura, criptografando a conexão entre a máquina local e a instância do EC2. Depois de se conectar pelo túnel, você se autenticará no servidor de VNC usando a senha que configurou na etapa anterior. Para saber mais sobre grupos de segurança, consulte [Alterar os grupos de segurança da instância do Amazon EC2](#) no Guia do usuário do Amazon EC2.

1. Crie um túnel SSH por meio da sua máquina local.

```
$ ssh -i <keypair> -L 5901:localhost:5901 ec2-user@<address>
```

 Note

Substitua `<keypair>` pelo caminho para a sua chave SSH e `<address>` pelo IP público ou nome DNS da sua instância. A porta muda com base no número de exibição usado para iniciar o `vncserver`. Por exemplo, `:1` usa a porta 5901, `:2` usa a porta 5902 etc.

2. Use o cliente VNC para se conectar a `localhost:5901` ou `127.0.0.1:5901` com a senha de VNC definida anteriormente.

 Important

Mantenha o túnel SSH aberto ao usar a VNC. Se o túnel SSH não estiver aberto, não será possível usar o cliente VNC para visualizar e interagir com o ambiente de área de trabalho.

(Opcional) Iniciar o serviço na inicialização

Se você planeja usar a VNC regularmente, talvez queira configurar o servidor de VNC para ser iniciado automaticamente quando a instância for inicializada. Isso elimina a necessidade de iniciar manualmente o servidor de VNC sempre que você reiniciar a instância. Essa configuração garante que o ambiente gráfico de área de trabalho esteja pronto e acessível assim que a instância concluir o processo de inicialização.

- Configure o serviço para ser iniciado no momento da inicialização.

```
[ec2-user ~]$ sudo systemctl enable vncserver@:1
```

 Important

Ao habilitar o serviço `vncserver`, a parcela depois de `@` deve corresponder ao número de exibição definido para o usuário no arquivo `/etc/tigervnc/vncserver.users`. Além disso, você pode enviar o argumento `--now` depois de `enable` para iniciar o serviço imediatamente.

Depois de realizar essa etapa, você não precisará mais iniciar o `vncserver` toda vez que reinicializar a instância.

(Opcional) Desabilitar a tela de bloqueio de inatividade

- Defina o atraso de inatividade como zero para desabilitar a tela de bloqueio quando o usuário ficar inativo por um longo período.

```
[ec2-user ~]$ gsettings set org.gnome.desktop.session idle-delay 0
```

Tópicos relacionados

Para saber mais sobre o ambiente gráfico de área de trabalho, consulte a seguinte documentação:

- [Tutorial: Instale o ambiente de trabalho GNOME em AL2023](#)
- [What Is Amazon DCV?](#) no Guia do administrador do Amazon DCV

Usando o Multi-Gen LRU (MGLRU) em kernels AL2023

O [LRU multigeracional](#) é um algoritmo moderno de recuperação de páginas no kernel do Linux, projetado para melhorar o desempenho do gerenciamento de memória sob pressão da memória. Ele substitui o mecanismo tradicional de LRU (menos acessado recentemente) usado para determinar quais páginas da memória devem ser recuperadas quando o sistema está com pouca memória.

O mecanismo tradicional de LRU usa um modelo de duas listas (ativa e inativa) para rastrear o uso de páginas, o que pode se tornar ineficiente em workloads modernas com grandes conjuntos de trabalho. O MGLRU substitui esse método por várias “gerações” de páginas, permitindo que o kernel tome decisões mais inteligentes com base em informações mais refinadas sobre o envelhecimento.

Os benefícios do MGLRU incluem:

- Melhores decisões de recuperação: identificação mais precisa de páginas frias (não utilizadas).
- Menor latência e melhor throughput: especialmente para workloads com grandes espaços de endereço ou muitos processos simultâneos.

- Retenção de cache aprimorada: páginas usadas recentemente têm menos probabilidade de serem removidas prematuramente.
- Design escalável e eficiente em termos de bloqueio: funciona melhor em máquinas com muitos CPUs

Configuração e ajuste

A configuração do kernel `CONFIG_LRU_GEN` está habilitada nos AL2023 kernels. Ela é compilada no MGLRU, mas não o habilita por padrão.

O MGLRU pode ser habilitado e ajustado usando o arquivo `/sys/kernel/mm/lru_gen/enabled`. O valor é uma máscara de bits. Recomenda-se ativar todos os componentes, a menos que alguns deles tenham efeitos colaterais indesejáveis.

Bit	Componentes
0	O interruptor principal para o LRU multigera cional.
1	Limando o bit acessado nas entradas da tabela de páginas folhas em grandes lotes, quando o MMU o definir (por exemplo, em x86). Teoricamente, esse comportamento pode piorar a contenção de bloqueios (<code>mmap_lock</code>). Se estiver desabilitado, o LRU multigera cional sofrerá uma pequena degradação de desempenho para workloads que mapeiam páginas ativas de forma contígua, cujos bits acessados podem ser eliminados por menos lotes maiores.
2	Limando também o bit acessado nas entradas da tabela de páginas não folhas, quando o MMU o definir (por exemplo, em x86). Esse comportamento não foi verificado em variações x86 além da Intel e da AMD. Se estiver desabilitado, o LRU multigeracional sofrerá

Bit	Componentes
	uma degradação de desempenho insignificante.
[yYnN]	Enable/disable todos os componentes acima.

Um exemplo de como o MGLRU pode ser habilitado:

```
[ec2-user ~]$ echo y >/sys/kernel/mm/lru_gen/enabled
```

Isto habilita todos os componentes:

```
[ec2-user ~]$ cat /sys/kernel/mm/lru_gen/enabled
0x0007
```


Usando o Amazon Linux 2023 fora do Amazon EC2

As imagens do contêiner Amazon Linux 2023 podem ser executadas em ambientes de tempo de execução de contêineres compatíveis. Para obter mais informações sobre como usar o Amazon Linux 2023 dentro de um contêiner, consulte [AL2023 em contêineres](#).

O Amazon Linux 2023 (AL2023) também pode ser executado como um convidado virtualizado, além de ser executado diretamente no Amazon EC2. No momento, há imagens para KVM (qcow2), VMware (OVA) e Hyper-V (vhdx) disponíveis.

Note

A configuração das imagens do Amazon Linux 2023 é diferente da do Amazon Linux 2. Se você estiver [executando o Amazon Linux 2 como uma máquina virtual no local](#), precisará adaptar sua configuração para ser compatível com o AL2023.

Fazer download de imagens do Amazon Linux 2023 para usar com KVM, VMware e Hyper-V

As imagens de disco do Amazon Linux 2023 para uso com KVM, VMware e Hyper-V podem ser baixadas em cdn.amazonlinux.com.

Configurações compatíveis do Amazon Linux 2023 para uso em ambientes virtualizados não pertencentes ao Amazon EC2

Esta seção aborda os requisitos para executar o Amazon Linux 2023 em ambientes virtualizados que não sejam do Amazon EC2, como na KVM, na VMware ou no Hyper-V.

A base [Requisitos do sistema do AL2023](#) se aplica a todos os ambientes virtualizados que não são do Amazon EC2. Uma lista dos modelos de dispositivos compatíveis é detalhada para cada ambiente de hipervisor nos tópicos a seguir.

A KVM, a VMware e o Hyper-V oferecem muitas opções de configuração, e é preciso tomar cuidado para configurá-las de acordo com suas necessidades de segurança, desempenho e confiabilidade. Para obter mais informações, consulte a documentação fornecida pelo seu hipervisor.

Tópicos

- [Requisitos para execução AL2023 em KVM](#)
- [Requisitos para executar o AL2023 na VMware](#)
- [Requisitos para executar o Amazon Linux 2023 no Hyper-V](#)

Requisitos para execução AL2023 em KVM

Esta seção descreve os requisitos para execução AL2023 no KVM. As imagens KVM do AL2023 estão disponíveis para ambas as `aarch64` arquiteturas `x86-64`. Esses requisitos são adicionais aos [Requisitos do sistema do AL2023](#) básicos para as imagens para KVM.

Tópicos

- [Requisitos de host KVM para execução em AL2023 KVM](#)
- [Suporte de dispositivos para um AL2023 KVM](#)
- [Modo de inicialização \(UEFI/BIOS\) suporte para um AL2023 KVM](#)
- [Limitações em execução AL2023 no KVM](#)

Requisitos de host KVM para execução em AL2023 KVM

As imagens para KVM estão atualmente qualificadas em um host executando o Ubuntu 22.04.3 LTS com `qemu` versão 6.2+dfsg-2ubuntu6.15 fornecido por esta versão do Ubuntu, usando um tipo de máquina `q35` para `x86-64` e um tipo de máquina `virt` para `aarch64`.

Suporte de dispositivos para um AL2023 KVM

Os modelos de **qemu** dispositivos testados para uso com imagens AL2023 KVM (ambos **aarch64** e **x86-64**) são:

- `virtio-blk` (dispositivo de bloco `virtio`)
- `virtio-scsi` (Controlador `virtio` SCSI com dispositivo de disco)
- `virtio-net` (dispositivo `virtio` de rede)
- `ahci` (para uso com a unidade de CD-ROM virtual)
- `usb-storage` (sobre `xhci`)

Modelos de **qemu** dispositivos adicionais habilitados na qualificação de imagem AL2023 KVM, mas não muito exercitados, são:

- VGA (qemu VGA) somente no x86-64
- `virtio-rng` (gerador virtual de números aleatórios)
- Dispositivos antigos de teclado AT e mouse PS/2
- dispositivo serial antigo

Modo de inicialização (UEFI/BIOS) suporte para um AL2023 KVM

A imagem x86-64 é testada com os modos antigos BIOS e UEFI de inicialização. As imagens aarch64 são testadas com o modo de inicialização UEFI.

Note

Por padrão, ao usar o modo de inicialização UEFI, alguns gerenciadores de máquinas virtuais provisionarão a VM com as chaves Microsoft Secure Boot, o que habilita o Secure Boot. Essa configuração não inicializará AL2023.

Como o carregador de AL2023 inicialização não é assinado pela Microsoft, a VM deve ser provisionada sem as chaves UEFI ou com as chaves do AL2023 Secure Boot.

Important

O suporte ao Secure Boot a imagens para KVM ainda não foi validado.

Limitações em execução AL2023 no KVM

Existem algumas limitações conhecidas na execução AL2023 no KVM.

Note

O código que implementa algumas das funcionalidades não suportadas listadas pode existir AL2023 e funcionar corretamente. A lista de funcionalidades não compatíveis existe para que você possa tomar decisões informadas sobre em que confiar para trabalhar hoje e o que a equipe do Amazon Linux qualificará como parte de futuras atualizações.

Limitações conhecidas com a execução AL2023 no KVM

- O agente convidado KVM não está atualmente empacotado ou não é compatível.
- Não há suporte para conexão e desconexão automática de CPU, memória ou qualquer outro tipo de dispositivo.
- Não há suporte à hibernação da VM.
- Não há suporte à migração da VM.
- A passagem de qualquer dispositivo, como PCI Passthrough ou USB Passthrough, não é compatível.

Requisitos para executar o AL2023 na VMware

Esta seção descreve os requisitos para executar o AL2023 na VMware. As imagens do AL2023 para VMware estão disponíveis apenas para a arquitetura x86-64. As imagens do VMware para aarch64 não estão disponíveis nem são compatíveis. Esses requisitos são adicionais aos [Requisitos do sistema do AL2023](#) básicos para as imagens para VMware.

Tópicos

- [Requisitos de host da VMware para executar o AL2023 na VMware](#)
- [Suporte a dispositivos para o AL2023 na VMware](#)
- [Suporte a modos de inicialização \(UEFI e BIOS\) para o AL2023 na VMware](#)
- [Limitações ao executar o AL2023 na VMware](#)

Requisitos de host da VMware para executar o AL2023 na VMware

As imagens do AL2023 para VMware OVA estão atualmente qualificadas no seguinte:

- VMware Workstation 17.5.0 em execução em hosts usando um processador Intel(R) Xeon(R) Platinum 8124M
- VMware vSphere 8.0 usando um processador Intel(R) Xeon(R) Platinum 8275CL

As imagens do AL2023 para VMware OVA especificam uma versão de hardware da máquina de 13.

A versão 13 de hardware da máquina da VMware é compatível com:

- ESXi 6.5 ou posterior

- VMware Workstation 14 ou posterior

Suporte a dispositivos para o AL2023 na VMware

Os seguintes modelos de dispositivos da VMware foram testados para uso com imagens do AL2023 para VMware OVA (somente **x86-64**):

- `vmw_pvscsi` (controlador SCSI paravirtualizado da VMware)
- `vmxnet3` (dispositivo de rede paravirtualizado da VMware)
- `ata_piix` (IDE antigo para uso somente com a unidade de CD-ROM virtual)

Modelos adicionais de dispositivos da VMware habilitados na qualificação da imagem do AL2023 para VMware, mas não amplamente utilizados:

- `vmw_vmci` e a interface `vsock` relacionada (transporte de soquete virtual para o agente convidado da VMware)
- Dispositivo de balão de memória `vmw_balloon`
- Controlador SVGA da VMware
- Dispositivos antigos de teclado AT e mouse PS/2

O pacote de agente convidado da VMware (`open-vm-tools`) está disponível e instalado por padrão nas imagens do AL2023 para VMware OVA.

Suporte a modos de inicialização (UEFI e BIOS) para o AL2023 na VMware

A partir do lançamento 2023.3.20231211, a imagem do AL2023 para VMware OVA foi validada nos modos de inicialização legados BIOS e UEFI. A configuração padrão do OVA ainda é o BIOS legado, mas pode ser alterada pelo usuário.

Important

O suporte ao Secure Boot requer UEFI, que não foi validado para o AL2023 em execução na VMware.

Limitações ao executar o AL2023 na VMware

Há algumas limitações conhecidas relacionadas à execução do AL2023 na VMware.

Note

O código que implementa algumas das funcionalidades não compatíveis listadas pode existir no AL2023 e funcionar corretamente. A lista de funcionalidades não compatíveis existe para que os clientes possam tomar decisões informadas sobre em que confiar para trabalhar hoje e o que a equipe do Amazon Linux qualificará como parte de futuras atualizações.

Limitações conhecidas relacionadas à execução do AL2023 na VMware

- O UEFI Secure Boot ainda não foi validado com o AL2023 na VMware.
- Não há suporte para conexão e desconexão automática de CPU, memória ou qualquer outro tipo de dispositivo.
- Não há suporte à hibernação da VM.
- Não há suporte à migração da VM.
- A passagem de qualquer dispositivo, como PCI Passthrough ou USB Passthrough, não é compatível.

Requisitos para executar o Amazon Linux 2023 no Hyper-V

Esta seção aborda os requisitos para executar o Amazon Linux 2023 no Hyper-V. As imagens do AL2023 para Hyper-V estão disponíveis apenas para a arquitetura x86-64. As imagens para Hyper-V para aarch64 não estão disponíveis nem são compatíveis no momento.

Esta seção aborda os requisitos adicionais além dos [Requisitos do sistema do AL2023](#) básicos para as imagens para KVM.

Tópicos

- [Requisitos de host do Hyper-V para executar o Amazon Linux 2023 no Hyper-V](#)
- [Suporte a dispositivos para o Amazon Linux 2023 no Hyper-V](#)
- [Limitações ao executar o Amazon Linux 2023 no Hyper-V](#)

Requisitos de host do Hyper-V para executar o Amazon Linux 2023 no Hyper-V

A principal qualificação do Amazon Linux 2023 no Hyper-V acontece no Windows Server 2022 executado em uma instância `c5.metal` do EC2.

Suporte a dispositivos para o Amazon Linux 2023 no Hyper-V

O Amazon Linux 2023 é testado em máquinas virtuais Hyper-V da 1ª geração e 2ª geração com o seguinte conjunto de hardware virtualizado:

- VM da 1ª geração (inicialização legada do BIOS)
- VM da 2ª geração (inicialização no modo UEFI, sem inicialização segura)
- Os seguintes modelos de dispositivos foram testados para uso com imagens do AL2023 para Hyper-V:
 - Armazenamento virtual do Hyper-V `hv_storvsc` para o disco-raiz e a unidade CD-ROM emulada em VMs da 2ª geração.
 - IDE PIIX emulado `ata_piix` para a unidade CD-ROM virtual em VMs da 1ª geração.
 - Ethernet virtual do Hyper-V `hv_netvsc`.
- Os seguintes modelos de dispositivos estão habilitados, mas apenas levemente testados:
 - Modo de texto VGA legado em VMs da 1ª geração.
 - Framebuffer baseado em firmware UEFI `simplifiedmfb` em VMs da 2ª geração.
 - Balão do Hyper-V `hv_balloon`.
 - HID/mouse do Hyper-V `hid_hyperv`.
- Os seguintes modos de dispositivo ainda não estão habilitados no AL2023:
 - Passagem PCI do Hyper-V.
 - Gráficos do Hyper-V DRM.

Important

Para máquinas virtuais da 2ª geração, não há suporte ao Secure Boot e ele deve ser desabilitado antes de inicializar a máquina virtual para garantir uma inicialização bem-sucedida do Amazon Linux 2023. No momento, o Hyper-V só oferece suporte ao Secure Boot com componentes de software assinados por chaves próprias da Microsoft, enquanto o

carregador de inicialização do Amazon Linux é assinado por uma chave privada da Amazon. O Hyper-V ainda não oferece suporte à importação de chaves de terceiros.

Limitações ao executar o Amazon Linux 2023 no Hyper-V

A seguir estão algumas limitações conhecidas relacionadas à execução do Amazon Linux 2023 no Hyper-V:

Note

O código que implementa algumas das funcionalidades não compatíveis listadas pode existir no AL2023 e funcionar corretamente. A lista de funcionalidades não compatíveis existe para que os clientes possam tomar decisões informadas sobre em que confiar para trabalhar hoje e o que a equipe do Amazon Linux qualificará como parte de futuras atualizações.

Limitações conhecidas relacionadas à execução do AL2023 no Hyper-V

- O modo UEFI Secure Boot ainda não é compatível nem funciona com o AL2023 no Hyper-V.
- Não há suporte para conexão e desconexão automática de CPU, memória ou qualquer outro tipo de dispositivo.
- A hibernação da máquina virtual (VM) não é compatível.
- A migração de máquina virtual (VM) não é compatível.
- A passagem de qualquer dispositivo, como PCI Passthrough ou USB Passthrough, não é compatível.

Instalação do Amazon Linux 2023 e configuração do **cloud-init** quando usado fora do Amazon EC2

Esta seção aborda como instalar e configurar uma máquina virtual do Amazon Linux 2023 quando não é executada diretamente no Amazon EC2, como na KVM, na VMware ou no Hyper-V.

Por padrão, as imagens de uma máquina virtual Amazon Linux 2023 não vêm provisionadas com nenhuma senha de usuário ou chave ssh e obterão sua configuração de rede por meio de DHCP na primeira interface de rede descoberta. Isso significa que, por padrão, sem configuração adicional, não há como se conectar à máquina virtual resultante.

Portanto, alguma forma de configuração precisa ser fornecida à máquina virtual. O mecanismo padrão para fazer isso no Amazon Linux é por meio de fontes de dados `cloud-init`.

O Amazon Linux 2023 foi qualificado com as seguintes fontes de dados:

NoCloud

Esse é o método tradicional de configuração de imagens locais por meio de um CD-ROM virtual contendo uma imagem inicial ISO9660 com arquivos de configuração `cloud-init`.

VMware

Além disso, o Amazon Linux 2023 oferece suporte à configuração de imagens VMware executadas no vSphere por meio da fonte de dados específica da VMware via VMware `guestinfo.userdata` e `guestinfo.metadata`.

Note

A configuração das fontes de dados pode ser diferente da do Amazon Linux 2. Mais especificamente, o Amazon Linux 2023 usa `systemd-networkd` para sua configuração e exige o uso do `cloud-init` "Networking Config Version 2", conforme documentado na [documentação de configuração de cloud-init rede](#).

A documentação completa dos mecanismos de `cloud-init` configuração da versão `cloud-init` empacotada no Amazon Linux 2023 pode ser encontrada na [documentação upstream cloud-init](#).

Configuração NoCloud (**seed.iso**) **cloud-init** para Amazon Linux 2023 em KVM e VMware

Esta seção aborda como criar e usar uma imagem `seed.iso` para configurar o Amazon Linux 2023 em execução na KVM ou na VMware. Como os ambientes da KVM e da VMware não têm o [serviço de metadados de instância do Amazon EC2 \(IMDS\)](#), é necessário um método alternativo de configuração do Amazon Linux 2023, e fornecer uma imagem `seed.iso` é um desses métodos.

A imagem de `seed.iso` inclui as informações de configuração inicial necessárias para inicializar e configurar sua nova máquina virtual, como a configuração de rede, o nome do host e os dados do usuário.

 Note

A imagem de `seed.iso` inclui somente as informações de configuração necessárias para inicializar a VM. Ela não inclui os arquivos do sistema operacional do Amazon Linux 2023.

Para gerar a imagem de `seed.iso`, você precisa de pelo menos dois arquivos de configuração:

meta-data

Esse arquivo normalmente inclui o nome do host da máquina virtual.

user-data

Esse arquivo normalmente configura contas de usuário, senhas, pares de chaves ssh e/ou mecanismos de acesso. Por padrão, as imagens do VMware do Amazon Linux 2023 KVM criam uma conta de usuário `ec2-user`. Você pode usar o arquivo de configuração `user-data` para definir a senha e/ou as teclas SSH para esta conta de usuário padrão.

network-config (opcional)

Esse arquivo normalmente fornece uma configuração de rede para a máquina virtual que substituirá a padrão. A configuração padrão é para usar DHCP na primeira interface de rede disponível.

Crie a imagem de disco **seed.iso**

1. Em um computador Linux ou macOS, crie uma nova pasta chamada `seedconfig` e navegue até ela.

 Note

É possível usar o Windows ou outro sistema operacional para concluir essas etapas, mas você precisará encontrar a ferramenta equivalente a `mkisofs` para concluir a criação da imagem `seed.iso`.

2. Crie o arquivo de configuração `meta-data`.
 - a. Crie um novo arquivo chamado `meta-data`.

- b. Abra o arquivo meta-data usando o editor de texto de sua preferência e adicione o seguinte: substituindo *vm-hostname* pelo nome do host da VM:

```
#cloud-config
local-hostname: vm-hostname
```

- c. Salve e feche o arquivo de configuração meta-data.
3. Crie o arquivo de configuração user-data.
 - a. Crie um novo arquivo chamado user-data.
 - b. Abra o arquivo user-data usando o editor de texto de sua preferência e adicione o seguinte, fazendo as substituições necessárias:

```
#cloud-config
#vim:syntax=yaml
users:
# A user by the name 'ec2-user' is created in the image by default.
- default
- name: ec2-user
ssh_authorized_keys:
- ssh-rsa ssh-key
# In the above line, replace ssh key with the content of your ssh public key.
```

- c. Você também pode adicionar mais contas de usuário ao arquivo de configuração user-data.

Também é possível especificar contas de usuário adicionais, seus mecanismos de acesso, senhas e pares de chave. Para obter mais informações sobre as diretivas compatíveis, consulte a [Documentação upstream de cloud-init](#).

- d. Salve e feche o arquivo de configuração user-data.
4. (Opcional) Crie o arquivo de configuração network-config.
 - a. Crie um novo arquivo chamado network-config.
 - b. Abra o arquivo network-config usando o editor de texto de sua preferência e adicione o seguinte: substitua os vários endereços IP pelos apropriados para sua configuração.

```
#cloud-config
version: 2
```

```
ethernets:
  enp1s0:
    addresses:
      - 192.168.122.161/24
    gateway4: 192.168.122.1
    nameservers:
      addresses: 192.168.122.1
```

 Note

A configuração de rede `cloud-init` fornece mecanismos para comparar com o endereço MAC da interface em vez de especificar o nome da interface, que pode mudar dependendo da configuração da VM. Esses (e mais) recursos `cloud-init` para configuração de rede são descritos com mais detalhes na [documentação upstream do cloud-init Network Config Versão 2](#).

- c. Salve e feche o arquivo de configuração `network-config`.
5. Crie a imagem do disco `seed.iso` usando os arquivos de configuração `meta-data`, `user-data` e `network-config` opcionais criados nas etapas anteriores.

Siga um destes procedimentos, dependendo do sistema operacional no qual você está criando a imagem do disco `seed.iso`.

- Em sistemas Linux, use uma ferramenta como **mkisofs** ou **genisoimage** para criar o arquivo completo `seed.iso`. Navegue até a pasta `seedconfig` e execute o comando a seguir:

```
$ mkisofs -output seed.iso -volid cidata -joliet -rock user-data meta-data
```

- Se você usar a `network-config`, inclua-a na invocação de **mkisofs**:

```
$ mkisofs -output seed.iso -volid cidata -joliet -rock user-data meta-data
network-config
```

- Nos sistemas macOS, você pode usar uma ferramenta como **hdiutil** para gerar o arquivo `seed.iso` finalizado. Como **hdiutil** leva um nome de caminho em vez de uma lista de arquivos, a mesma invocação pode ser usada, independentemente de um arquivo de configuração `network-config` ter sido criado ou não.

```
$ hdiutil makehybrid -o seed.iso -hfs -joliet -iso -default-volume-name cidata  
seedconfig/
```

6. O arquivo `seed.iso` resultante agora pode ser anexado à sua nova máquina virtual do Amazon Linux 2023 usando uma unidade CD-ROM virtual para `cloud-init` a fim de encontrar a configuração ao sistema na primeira inicialização e aplicá-la.

VMwarecloud-initconfiguração guestinfo para on AL2023 VMware

VMwareos ambientes não têm o [Amazon EC2 Instance Meta Data Service \(IMDS\)](#), portanto, é necessário um método alternativo de configuração. AL2023 Esta seção descreve como usar um mecanismo de configuração alternativo para a unidade CD-ROM virtual `seed.iso` que está disponível na VMware vSphere.

Esse método de configuração usa o mecanismo `extraconfig` da VMware para fornecer dados de configuração a `cloud-init`. Para cada uma das chaves a seguir, uma propriedade **keyname.encoding** correspondente deve ser fornecida.

As seguintes chaves podem ser fornecidas ao mecanismo `extraconfig` da VMware.

guestinfo.metadata

JSON ou YAML contendo metadados de `cloud-init`

guestinfo.userdata

Um documento YAML contendo dados do usuário `cloud-init` no formato `cloud-config`.

guestinfo.vendordata (opcional)

YAML contendo dados do fornecedor de `cloud-init`

As propriedades de codificação correspondentes (`guestinfo.metadata.encoding`, `guestinfo.userdata.encoding` e `guestinfo.vendordata.encoding`) podem conter:

base64

O conteúdo da propriedade é codificado em `base64`.

gzip+base64

O conteúdo da propriedade é compactado com `gzip` após a codificação de `base64`.

Note

O método `seed.iso` é compatível com um arquivo de configuração `network-config` separado (opcional). O `guestinfo` da VMware difere na forma como a configuração de rede é fornecida. Informações adicionais são fornecidas na próxima seção.

Se uma configuração de rede explícita for desejada, ela deverá ser incorporada em metadata na forma de duas propriedades YAML ou JSON:

network

Contém a configuração de rede codificada em formato JSON ou YAML.

network.encoding

Contém a codificação dos dados de configuração de rede acima. As codificações `cloud-init` compatíveis são as mesmas dos dados de `guestinfo`: `base64` e `gzip+base64`.

Example Usar a ferramenta de CLI **govc** da VMware vSphere para enviar a configuração com **guestinfo**

1. Prepare os arquivos de configuração meta-data, user-data e, opcionalmente, network-config, conforme descrito em [Configuração NoCloud \(seed.iso\) cloud-init para Amazon Linux 2023 em KVM e VMware](#).
2. Converta os arquivos de configuração em formatos utilizáveis por `guestinfo` da VMware.

```
# 'meta-data', `user-data` and `network-config` are the configuration
# files in the same format that would be used by a NoCloud (seed.iso)
# data source, read-them and convert them to VMware guestinfo
#
# The VM_NAME variable is assumed to be set to the name of the VM
# It is assumed that the necessary govc environment (credentials etc...) are
# already set

metadata=$(cat "meta-data")
userdata=$(cat "user-data")
if [ -e "network-config" ] ; then
    # We need to embed the network config inside the meta-data
    netconf=$(base64 -w0 "network-config")
```

```
    metadata=$(printf "%s\nnetwork: %s\nnetwork.encoding: base64" "$metadata"
"$netconf")
fi
metadata=$(base64 -w0 <<< "$metadata")
govc vm.change -vm "$VM_NAME" \
    -e guestinfo.metadata="$metadata" \
    -e guestinfo.metadata.encoding="base64"
userdata=$(base64 -w0 <<< "$userdata")
govc vm.change -vm "$VM_NAME" \
    -e guestinfo.userdata="$userdata" \
    -e guestinfo.userdata.encoding="base64"
```

Comparar pacotes instalados na AMI padrão do Amazon Linux 2023 com a imagem do AL2023 para KVM

Uma comparação dos RPMs presentes na AMI padrão do AL2023 com os RPMs presentes na imagem do AL2023 para KVM.

Pacote	AMI	KVM
acl	2.3.1	2.3.1
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chrony-config	4.3	
amazon-ec2-net-utils	2.5.1	
amazon-linux-onprem		1.2
amazon-linux-repo-cdn		2023.6.20241031
amazon-linux-repo-s3	2023.6.20241031	
amazon-linux-sb-keys	2023.1	2023.1

Pacote	AMI	KVM
amazon-onprem-netw ork		1.2
amazon-rpm-config	228	228
amazon-ssm-agent	3.3.987.0	3.3.987.0
amd-ucode-firmware	20210208 (noarch)	20210208 (noarch)
at	3.1.23	3.1.23
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2.15.30	2.15.30
basesystem	11	11
bash	5.2.15	5.2.15
bash-completion	2.11	2.11
bc	1.07.1	1.07.1
bind-libs	9.18.28	9.18.28
bind-license	9.18.28	9.18.28
bind-utils	9.18.28	9.18.28
binutils	2.39	2.39
boost-filesystem	1.75.0	1.75.0
boost-system	1.75.0	1.75.0

Pacote	AMI	KVM
boost-thread	1.75.0	1.75.0
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023.2.68	2023.2.68
c-ares	1.19.1	
checkpolicy	3.4	3.4
chkconfig	1.15	1.15
chrony	4.3	4.3
cloud-init	22.2.2	22.2.2
cloud-init-cfg-ec2	22.2.2	
cloud-init-cfg-onpre		22.2.2
cloud-utils-growpart	0.31	0.31
coreutils	8.32	8.32
coreutils-common	8.32	8.32
cpio	2.13	2.13
cracklib	2.9.6	2.9.6
cracklib-dicts	2.9.6	2.9.6
crontabs	1.11	1.11
crypto-policies	20220428	20220428

Pacote	AMI	KVM
crypto-policies-scripts	20220428	20220428
cryptsetup	2.6.1	2.6.1
cryptsetup-libs	2.6.1	2.6.1
curl-minimal	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27
cyrus-sasl-plain	2.1.27	2.1.27
dbus	1.12.28	1.12.28
dbus-broker	32	32
dbus-common	1.12.28	1.12.28
dbus-libs	1.12.28	1.12.28
device-mapper	1.02.185	1.02.185
device-mapper-libs	1.02.185	1.02.185
diffutils	3.8	3.8
dnf	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2
dnf-plugins-core	4.3.0	4.3.0
dnf-plugin-support-info	1.2	1.2

Pacote	AMI	KVM
dnf-utils	4.3.0	4.3.0
dosfstools	4.2	4.2
dracut	055	055
dracut-config-ec2	3.0	
dracut-config-generic	055	055
dwz	0.14	0.14
dyninst	10.2.1	10.2.1
e2fsprogs	1.46.5	1.46.5
e2fsprogs-libs	1.46.5	1.46.5
ec2-hibinit-agent	1.0.8	
ec2-instance-connect	1.1	
ec2-instance-connect-selinux	1.1	
ec2-utils	2.2.0	
ed	1.14.2	1.14.2
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38
efivar-libs	38	38

Pacote	AMI	KVM
elfutils-debuginfod-client	0.188	0.188
elfutils-default-yama-scope	0.188	0.188
elfutils-libelf	0.188	0.188
elfutils-libs	0.188	0.188
ethtool	5.15	5.15
expat	2.5.0	2.5.0
file	5.39	5.39
file-libs	5.39	5.39
filesystem	3.14	3.14
findutils	4.8.0	4.8.0
fonts-srpm-macros	2.0.5	2.0.5
fstrm	0.6.1	0.6.1
fuse-libs	2.9.9	2.9.9
gawk	5.1.0	5.1.0
gdbm-libs	1.19	1.19
gdisk	1.0.8	1.0.8
gettext	0.21	0.21
gettext-libs	0.21	0.21
ghc-srpm-macros	1.5.0	1.5.0

Pacote	AMI	KVM
glib2	2.74.7	2.74.7
glibc	2.34	2.34
glibc-all-langpacks	2.34	2.34
glibc-common	2.34	2.34
glibc-gconv-extra	2.34	2.34
glibc-locale-source	2.34	2.34
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gnutls	3.8.0	3.8.0
go-srpm-macros	3.2.0	3.2.0
gpgme	1.15.1	1.15.1
gpm-libs	1.20.7	1.20.7
grep	3.8	3.8
groff-base	1.22.4	1.22.4
grub2-common	2.06	2.06
grub2-efi-aa64-ec2	2.06 (aarch64)	2.06 (aarch64)
grub2-efi-x64-ec2	2.06 (x86_64)	2.06 (x86_64)
grub2-pc		2.06 (x86_64)
grub2-pc-modules	2.06	2.06 (noarch)
grub2-tools	2.06	2.06

Pacote	AMI	KVM
grub2-tools-minimal	2.06	2.06
grubby	8.40	8.40
gssproxy	0.8.4	0.8.4
gzip	1.12	1.12
hostname	3.23	3.23
hunspell	1.7.0	1.7.0
hunspell-en	0.20140811.1	0.20140811.1
hunspell-en-GB	0.20140811.1	0.20140811.1
hunspell-en-US	0.20140811.1	0.20140811.1
hunspell-filesystem	1.7.0	1.7.0
hwdata	0.384	0.384
info	6.7	6.7
inih	49	49
initscripts	10.09	10.09
iproute	6.10.0	6.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jemalloc	5.2.1	5.2.1
jitterentropy	3.4.1	3.4.1

Pacote	AMI	KVM
jq	1.7.1	
json-c	0.14	0.14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0
kernel	6.1.112	6.1.112
kernel-libbpf	6.1.112	6.1.112
kernel-livepatch-r epo-cdn		2023.6.20241031
kernel-livepatch-r epo-s3	2023.6.20241031	
kernel-modules-extra		6.1.112
kernel-modules-ext ra-common		6.1.112
kernel-srpm-macros	1.0	1,0
kernel-tools	6.1.112	6.1.112
keyutils	1.6.3	1.6.3
keyutils-libs	1.6.3	1.6.3
kmod	29	29
kmod-libs	29	29
kpatch-runtime	0.9.7	0.9.7
krb5-libs	1.21.3	1.21.3

Pacote	AMI	KVM
less	608	608
libacl	2.3.1	2.3.1
libaio	0.3.111	0.3.111
libarchive	3.7.4	3.7.4
libargon2	20171227	20171227
libassuan	2.5.5	2.5.5
libattr	2.5.1	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2.37.4	2.37.4
libcap	2.48	2.48
libcap-ng	0.8.2	0.8.2
libcbor	0.7.0	0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1.46.5	1.46.5
libcomps	0.1.20	0.1.20
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0.69.0	0.69.0

Pacote	AMI	KVM
libeconf	0.4.0	0.4.0
libedit	3.1	3.1
libev	4.33	4.33
libevent	2.1.12	2.1.12
libfdisk	2.37.4	2.37.4
libffi	3.4.4	3.4.4
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1
libgpg-error	1.42	1.42
libibverbs	48.0	48.0
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1
libkcapi	1.4.0	1.4.0
libkcapi-hmacalc	1.4.0	1.4.0
libldb	2.6.2	
libmaxminddb	1.5.2	1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.4	1.0.4

Pacote	AMI	KVM
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libnfsidmap	2.5.4	2.5.4
libnhttp2	1.59.0	1.59.0
libnl3	3.5.0	3.5.0
libpath_utils	0.2.1	0.2.1
libpcap	1.10.1	1.10.1
libpipeline	1.5.3	1.5.3
libpkgconf	1.8.0	1.8.0
libpsl	0.21.1	0.21.1
libpwquality	1.4.4	1.4.4
libref_array	0.1.5	0.1.5
librepo	1.14.5	1.14.5
libreport-filessystem	2.15.2	2.15.2
libseccomp	2.5.3	2.5.3
libselinux	3.4	3.4
libselinux-utils	3.4	3.4
libsemanage	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2.13	2.13

Pacote	AMI	KVM
libsmartcols	2.37.4	2.37.4
libsolv	0.7.22	0.7.22
libss	1.46.5	1.46.5
libsss_certmap	2.9.4	
libsss_idmap	2.9.4	2.9.4
libsss_nss_idmap	2.9.4	2.9.4
libsss_sudo	2.9.4	
libstdc++	11.4.1	11.4.1
libstoragegmt	1.9.4	1.9.4
libtalloc	2.3.4	
libtasn1	4.19.0	4.19.0
libtdb	1.4.7	
libtevent	0.13.0	
libtextstyle	0.21	0.21
libtirpc	1.3.3	1.3.3
libunistring	0.9.10	0.9.10
libuser	0.63	0.63
libutempter	1.2.1	1.2.1
libuuid	2.37.4	2.37.4
libuv	1.47.0	1.47.0

Pacote	AMI	KVM
libverto	0.3.2	0.3.2
libverto-libev	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33
libxml2	2.10.4	2.10.4
libyaml	0.2.5	0.2.5
libzstd	1.5.5	1.5.5
linux-firmware-whence	20210208 (noarch)	20210208 (noarch)
lm_sensors-libs	3.6.0	3.6.0
lmdb-libs	0.9.29	0.9.29
logrotate	3.20.1	3.20.1
lsof	4.94.0	4.94.0
lua-libs	5.4.4	5.4.4
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4
man-db	2.9.3	2.9.3
man-pages	5.10	5.10
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mpfr	4.1.0	4.1.0
nano	5,8	5,8
ncurses	6.2	6.2

Pacote	AMI	KVM
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
nettle	3.8	3.8
net-tools	2.0	2.0
newt	0.52.21	0.52.21
nfs-utils	2.5.4	2.5.4
npth	1,6	1.6
nspr	4.35.0	4.35.0
nss	3.90.0	3.90.0
nss-softokn	3.90.0	3.90.0
nss-softokn-freebl	3.90.0	3.90.0
nss-sysinit	3.90.0	3.90.0
nss-util	3.90.0	3.90.0
ntsysv	1.15	1.15
numactl-libs	2.0.14	2.0.14
ocaml-srpm-macros	6	6
oniguruma	6.9.7.1	
openblas-srpm-macros	2	2
openldap	2.4.57	2.4.57
openssh	8.7p1	8.7p1

Pacote	AMI	KVM
openssh-clients	8.7p1	8.7p1
openssh-server	8.7p1	8.7p1
openssl	3.0.8	3.0.8
openssl-lib	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12
os-prober	1.77	1.77
p11-kit	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1
package-notes-srpm-macros	0.4	0.4
pam	1.5.1	1.5.1
parted	3.4	3.4
passwd	0.80	0.80
pciutils	3.7.0	3.7.0
pciutils-lib	3.7.0	3.7.0
pcre2	10.40	10.40
pcre2-syntax	10.40	10.40
perl-Carp	1.50	1.50
perl-Class-Struct	0.66	0.66
perl-constant	1.33	1.33
perl-DynaLoader	1.47	1.47

Pacote	AMI	KVM
perl-Encode	3.15	3.15
perl-Errno	1.30	1.30
perl-Exporter	5.74	5.74
perl-Fcntl	1.13	1.13
perl-File-Basename	2.85	2.85
perl-File-Path	2.18	2.18
perl-File-stat	1.09	1.09
perl-File-Temp	0.231.100	0.231.100
perl-Getopt-Long	2.52	2.52
perl-Getopt-Std	1.12	1.12
perl-HTTP-Tiny	0.078	0.078
perl-if	0.60.800	0.60.800
perl-interpreter	5.32.1	5.32.1
perl-IO	1.43	1.43
perl-IPC-Open3	1.21	1.21
perl-libs	5.32.1	5.32.1
perl-MIME-Base64	3.16	3.16
perl-mro	1.23	1.23
perl-overload	1.31	1.31
perl-overloading	0.02	0.02

Pacote	AMI	KVM
perl-parent	0.238	0.238
perl-PathTools	3.78	3.78
perl-Pod-Escapes	1.07	1.07
perl-podlators	4.14	4.14
perl-Pod-Perldoc	3.28.01	3.28.01
perl-Pod-Simple	3.42	3.42
perl-Pod-Usage	2.01	2.01
perl-POSIX	1.94	1.94
perl-Scalar-List-Utils	1.56	1.56
perl-SelectSaver	1.02	1.02
perl-Socket	2.032	2.032
perl-srpm-macros	1	1
perl-Storable	3.21	3.21
perl-subs	1.03	1.03
perl-Symbol	1.08	1.08
perl-Term-ANSIColor	5.01	5.01
perl-Term-Cap	1.17	1.17
perl-Text-ParseWords	3.30	3.30
perl-Text-Tabs+Wrap	2021.0726	2021.0726
perl-Time-Local	1.300	1.300

Pacote	AMI	KVM
perl-vars	1.05	1.05
pkgconf	1.8.0	1.8.0
pkgconf-m4	1.8.0	1.8.0
pkgconf-pkg-config	1.8.0	1.8.0
polycoreutils	3.4	3.4
polycoreutils-python-utils	3.4	
popt	1.18	1.18
procps-ng	3.3.17	3.3.17
protobuf-c	1.4.1	1.4.1
psacct	6.6.4	6.6.4
psmisc	23.4	23.4
publicsuffix-list-dafsa	20240212	20240212
python3	3.9.16	3.9.16
python3-attrs	20.3.0	20.3.0
python3-audit	3.0.6	3.0.6
python3-awscrt	0.19.19	0.19.19
python3-babel	2.9.1	2.9.1
python3-cffi	1.14.5	1.14.5
python3-chardet	4.0.0	4.0.0

Pacote	AMI	KVM
python3-colorama	0.4.4	0.4.4
python3-configobj	5.0.6	5.0.6
python3-cryptography	36.0.1	36.0.1
python3-daemon	2.3.0	
python3-dateutil	2.8.1	2.8.1
python3-dbus	1.2.18	1.2.18
python3-distro	1.5.0	1.5.0
python3-dnf	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0
python3-docutils	0.16	0.16
python3-gpg	1.15.1	1.15.1
python3-hawkey	0.69.0	0.69.0
python3-idna	(2.10)	(2.10)
python3-jinja2	2.11.3	2.11.3
python3-jmespath	0.10.0	0.10.0
python3-jsonpatch	1.21	1.21
python3-jsonpointer	2.0	2.0
python3-jjsonschema	3.2.0	3.2.0
python3-libcomps	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0

Pacote	AMI	KVM
python3-libs	3.9.16	3.9.16
python3-libselinux	3.4	3.4
python3-libsemanage	3.4	3.4
python3-libstorage mgmt	1.9.4	1.9.4
python3-lockfile	0.12.2	
python3-markupsafe	1.1.1	1.1.1
python3-netifaces	0.10.6	0.10.6
python3-oauthlib	3.0.2	3.0.2
python3-pip-wheel	21.3.1	21.3.1
python3-ply	3.11	3.11
python3-policycore utils	3.4	3.4
python3-prettytable	0.7.2	0.7.2
python3-prompt-too lkit	3.0.24	3.0.24
python3-pycparser	2.20	2.20
python3-pyrsistent	0.17.3	0.17.3
python3-pyserial	3.4	3.4
python3-pysocks	1.7.1	1.7.1
python3-pytz	2022.7.1	2022.7.1

Pacote	AMI	KVM
python3-pyyaml	5.4.1	5.4.1
python3-requests	2.25.1	2.25.1
python3-rpm	4.16.1.3	4.16.1.3
python3-ruamel-yaml	0.16.6	0.16.6
python3-ruamel-yaml-clib	0.1.2	0.1.2
python3-setools	4.4.1	4.4.1
python3-setuptools	59.6.0	59.6.0
python3-setuptools-wheel	59.6.0	59.6.0
python3-six	1.15.0	1.15.0
python3-systemd	235	235
python3-urllib3	1.25.10	1.25.10
python3-wcwidth	0.2.5	0.2.5
python-chevron	0.13.1	
python-srpm-macros	3.9	3.9
quota	4.06	4.06
quota-nls	4.06	4.06
readline	8.1	8.1
rng-tools	6.14	6.14
rootfiles	8.1	8.1

Pacote	AMI	KVM
rpcbind	1.2.6	1.2.6
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	4.16.1.3
rsync	3.2.6	3.2.6
rust-srpm-macros	21	21
sbsigntools	0.9.4	0.9.4
screen	4.8.0	4.8.0
sed	4.8	4.8
selinux-policy	38.1.45	38.1.45
selinux-policy-targeted	38.1.45	38.1.45
setup	2.13.7	2.13.7
shadow-utils	4,9	4,9
slang	2.3.2	2.3.2
sqlite-libs	3.40.0	3.40.0
sssd-client	2.9.4	2.9.4

Pacote	AMI	KVM
sssd-common	2.9.4	
sssd-kcm	2.9.4	
sssd-nfs-idmap	2.9.4	
strace	6.8	6.8
sudo	1.9.15	1.9.15
sysctl-defaults	1.0	1,0
sysstat	12.5.6	12.5.6
systemd	252.23	252.23
systemd-libs	252.23	252.23
systemd-networkd	252.23	252.23
systemd-pam	252.23	252.23
systemd-resolved	252.23	252.23
systemd-udev	252.23	252.23
system-release	2023.6.20241031	2023.6.20241031
systemtap-runtime	4.8	4.8
tar	1.34	1.34
tbb	2020.3	2020.3
tcpdump	4.99.1	4.99.1
tcsh	6.24.07	6.24.07
time	1.9	1.9

Pacote	AMI	KVM
traceroute	2.1.3	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	2.2	2.2
userspace-rcu	0.12.1	0.12.1
util-linux	2.37.4	2.37.4
util-linux-core	2.37.4	2.37.4
vim-common	9.0.2153	9.0.2153
vim-data	9.0.2153	9.0.2153
vim-enhanced	9.0.2153	9.0.2153
vim-filesystem	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153
wget	1.21.3	1.21.3
which	2.21	2.21
words	3.0	3.0
xfsdump	3.1.11	3.1.11
xfspgrog	5.18.0	5.18.0
xxd	9.0.2153	9.0.2153
xxhash-libs	0.8.0	0.8.0
xz	5.2.5	5.2.5

Pacote	AMI	KVM
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0
zlib	1.2.11	1.2.11
zram-generator	1.1.2	
zram-generator-def aults	1.1.2	
zstd	1.5.5	1.5.5

Comparar pacotes instalados na AMI padrão do Amazon Linux 2023 com a imagem do AL2023 para VMware OVA

Uma comparação dos RPMs presentes na AMI padrão do AL2023 com os RPMs presentes na imagem do AL2023 para VMware OVA.

Pacote	AMI	VMware OVA
acl	2.3.1	2.3.1
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chrony-config	4.3	
amazon-ec2-net-utils	2.5.1	
amazon-linux-onprem		1.2
amazon-linux-repo- cdn		2023.6.20241031

Pacote	AMI	VMware OVA
amazon-linux-repo-s3	2023.6.20241031	
amazon-linux-sb-keys	2023.1	2023.1
amazon-onprem-netw ork		1.2
amazon-rpm-config	228	228
amazon-ssm-agent	3.3.987.0	3.3.987.0
amd-ucode-firmware	20210208	20210208
at	3.1.23	3.1.23
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2.15.30	2.15.30
basesystem	11	11
bash	5.2.15	5.2.15
bash-completion	2.11	2.11
bc	1.07.1	1.07.1
bind-libs	9.18.28	9.18.28
bind-license	9.18.28	9.18.28
bind-utils	9.18.28	9.18.28
binutils	2.39	2.39

Pacote	AMI	VMware OVA
boost-filesystem	1.75.0	1.75.0
boost-system	1.75.0	1.75.0
boost-thread	1.75.0	1.75.0
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023.2.68	2023.2.68
c-ares	1.19.1	
checkpolicy	3.4	3.4
chkconfig	1.15	1.15
chrony	4.3	4.3
cloud-init	22.2.2	22.2.2
cloud-init-cfg-ec2	22.2.2	
cloud-init-cfg-onpre		22.2.2
cloud-utils-growpart	0.31	0.31
coreutils	8.32	8.32
coreutils-common	8.32	8.32
cpio	2.13	2.13
cracklib	2.9.6	2.9.6
cracklib-dicts	2.9.6	2.9.6
crontabs	1.11	1.11

Pacote	AMI	VMware OVA
crypto-policies	20220428	20220428
crypto-policies-scripts	20220428	20220428
cryptsetup	2.6.1	2.6.1
cryptsetup-libs	2.6.1	2.6.1
curl-minimal	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27
cyrus-sasl-plain	2.1.27	2.1.27
dbus	1.12.28	1.12.28
dbus-broker	32	32
dbus-common	1.12.28	1.12.28
dbus-libs	1.12.28	1.12.28
device-mapper	1.02.185	1.02.185
device-mapper-libs	1.02.185	1.02.185
diffutils	3.8	3.8
dnf	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2
dnf-plugins-core	4.3.0	4.3.0

Pacote	AMI	VMware OVA
dnf-plugin-support-info	1.2	1.2
dnf-utils	4.3.0	4.3.0
dosfstools	4.2	4.2
dracut	055	055
dracut-config-ec2	3.0	
dracut-config-generic	055	055
dwz	0.14	0.14
dyninst	10.2.1	10.2.1
e2fsprogs	1.46.5	1.46.5
e2fsprogs-libs	1.46.5	1.46.5
ec2-hibinit-agent	1.0.8	
ec2-instance-connect	1.1	
ec2-instance-connect-selinux	1.1	
ec2-utils	2.2.0	
ed	1.14.2	1.14.2
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38

Pacote	AMI	VMware OVA
efivar-libs	38	38
elfutils-debuginfod-client	0.188	0.188
elfutils-default-yama-scope	0.188	0.188
elfutils-libelf	0.188	0.188
elfutils-libs	0.188	0.188
ethtool	5.15	5.15
expat	2.5.0	2.5.0
file	5.39	5.39
file-libs	5.39	5.39
filesystem	3.14	3.14
findutils	4.8.0	4.8.0
fonts-srpm-macros	2.0.5	2.0.5
fstrm	0.6.1	0.6.1
fuse3		3.10.4
fuse3-libs		3.10.4
fuse-common		3.10.4
fuse-libs	2.9.9	2.9.9
gawk	5.1.0	5.1.0
gdbm-libs	1.19	1.19

Pacote	AMI	VMware OVA
gdisk	1.0.8	1.0.8
gettext	0.21	0.21
gettext-libs	0.21	0.21
ghc-srpm-macros	1.5.0	1.5.0
glib2	2.74.7	2.74.7
glibc	2.34	2.34
glibc-all-langpacks	2.34	2.34
glibc-common	2.34	2.34
glibc-gconv-extra	2.34	2.34
glibc-locale-source	2.34	2.34
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gnutls	3.8.0	3.8.0
go-srpm-macros	3.2.0	3.2.0
gpgme	1.15.1	1.15.1
gpm-libs	1.20.7	1.20.7
grep	3.8	3.8
groff-base	1.22.4	1.22.4
grub2-common	2.06	2.06
grub2-efi-x64-ec2	2.06	2.06

Pacote	AMI	VMware OVA
grub2-pc		2.06
grub2-pc-modules	2.06	2.06
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.40	8.40
gssproxy	0.8.4	0.8.4
gzip	1.12	1.12
hostname	3.23	3.23
hunspell	1.7.0	1.7.0
hunspell-en	0.20140811.1	0.20140811.1
hunspell-en-GB	0.20140811.1	0.20140811.1
hunspell-en-US	0.20140811.1	0.20140811.1
hunspell-filesystem	1.7.0	1.7.0
hwdata	0.384	0.384
info	6.7	6.7
inih	49	49
initscripts	10.09	10.09
iproute	6.10.0	6.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0

Pacote	AMI	VMware OVA
jansson	2.14	2.14
jemalloc	5.2.1	5.2.1
jitterentropy	3.4.1	3.4.1
jq	1.7.1	
json-c	0.14	0.14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0
kernel	6.1.112	6.1.112
kernel-libbpf	6.1.112	6.1.112
kernel-livepatch-r epo-cdn		2023.6.20241031
kernel-livepatch-r epo-s3	2023.6.20241031	
kernel-modules-extra		6.1.112
kernel-modules-ext ra-common		6.1.112
kernel-srpm-macros	1.0	1,0
kernel-tools	6.1.112	6.1.112
keyutils	1.6.3	1.6.3
keyutils-libs	1.6.3	1.6.3
kmod	29	29

Pacote	AMI	VMware OVA
kmod-libs	29	29
kpatch-runtime	0.9.7	0.9.7
krb5-libs	1.21.3	1.21.3
less	608	608
libacl	2.3.1	2.3.1
libaio	0.3.111	0.3.111
libarchive	3.7.4	3.7.4
libargon2	20171227	20171227
libassuan	2.5.5	2.5.5
libattr	2.5.1	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2.37.4	2.37.4
libcap	2.48	2.48
libcap-ng	0.8.2	0.8.2
libcbor	0.7.0	0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1.46.5	1.46.5
libcomps	0.1.20	0.1.20
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0

Pacote	AMI	VMware OVA
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0.69.0	0.69.0
libeconf	0.4.0	0.4.0
libedit	3.1	3.1
libev	4.33	4.33
libevent	2.1.12	2.1.12
libfdisk	2.37.4	2.37.4
libffi	3.4.4	3.4.4
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1
libgpg-error	1.42	1.42
libibverbs	48.0	48.0
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1
libkcapi	1.4.0	1.4.0
libkcapi-hmacalc	1.4.0	1.4.0
libldb	2.6.2	

Pacote	AMI	VMware OVA
libmaxminddb	1.5.2	1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.4	1.0.4
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libmspack		0.10.1
libnfsidmap	2.5.4	2.5.4
libnghttp2	1.59.0	1.59.0
libnl3	3.5.0	3.5.0
libpath_utils	0.2.1	0.2.1
libpcap	1.10.1	1.10.1
libpipeline	1.5.3	1.5.3
libpkgconf	1.8.0	1.8.0
libpsl	0.21.1	0.21.1
libpwquality	1.4.4	1.4.4
libref_array	0.1.5	0.1.5
librepo	1.14.5	1.14.5
libreport-filessystem	2.15.2	2.15.2
libseccomp	2.5.3	2.5.3
libselinux	3.4	3.4

Pacote	AMI	VMware OVA
libselinux-utils	3.4	3.4
libsemanage	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2.13	2.13
libsmartcols	2.37.4	2.37.4
libsolv	0.7.22	0.7.22
libss	1.46.5	1.46.5
libsss_certmap	2.9.4	
libsss_idmap	2.9.4	2.9.4
libsss_nss_idmap	2.9.4	2.9.4
libsss_sudo	2.9.4	
libstdc++	11.4.1	11.4.1
libstoragegmt	1.9.4	1.9.4
libtalloc	2.3.4	
libtasn1	4.19.0	4.19.0
libtdb	1.4.7	
libtevent	0.13.0	
libtextstyle	0.21	0.21
libtirpc	1.3.3	1.3.3
libtool-ltdl		2.4.7

Pacote	AMI	VMware OVA
libunistring	0.9.10	0.9.10
libuser	0.63	0.63
libutempter	1.2.1	1.2.1
libuuid	2.37.4	2.37.4
libuv	1.47.0	1.47.0
libverto	0.3.2	0.3.2
libverto-libev	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33
libxml2	2.10.4	2.10.4
libxslt		1.1.34
libyaml	0.2.5	0.2.5
libzstd	1.5.5	1.5.5
linux-firmware-whence	20210208	20210208
lm_sensors-libs	3.6.0	3.6.0
lmdb-libs	0.9.29	0.9.29
logrotate	3.20.1	3.20.1
lsof	4.94.0	4.94.0
lua-libs	5.4.4	5.4.4
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4

Pacote	AMI	VMware OVA
man-db	2.9.3	2.9.3
man-pages	5.10	5.10
microcode_ctl	2.1	2.1
mpfr	4.1.0	4.1.0
nano	5,8	5,8
ncurses	6.2	6.2
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
nettle	3.8	3.8
net-tools	2.0	2.0
newt	0.52.21	0.52.21
nfs-utils	2.5.4	2.5.4
npth	1,6	1.6
nspr	4.35.0	4.35.0
nss	3.90.0	3.90.0
nss-softokn	3.90.0	3.90.0
nss-softokn-freebl	3.90.0	3.90.0
nss-sysinit	3.90.0	3.90.0
nss-util	3.90.0	3.90.0
ntsysv	1.15	1.15

Pacote	AMI	VMware OVA
numactl-libs	2.0.14	2.0.14
ocaml-srpm-macros	6	6
oniguruma	6.9.7.1	
openblas-srpm-macros	2	2
openldap	2.4.57	2.4.57
openssh	8.7p1	8.7p1
openssh-clients	8.7p1	8.7p1
openssh-server	8.7p1	8.7p1
openssl	3.0.8	3.0.8
openssl-libs	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12
open-vm-tools		12.3.0
os-prober	1.77	1.77
p11-kit	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1
package-notes-srpm-macros	0.4	0.4
pam	1.5.1	1.5.1
parted	3.4	3.4
passwd	0.80	0.80
pciutils	3.7.0	3.7.0

Pacote	AMI	VMware OVA
pciutils-libs	3.7.0	3.7.0
pcre2	10.40	10.40
pcre2-syntax	10.40	10.40
perl-Carp	1.50	1.50
perl-Class-Struct	0.66	0.66
perl-constant	1.33	1.33
perl-DynaLoader	1.47	1.47
perl-Encode	3.15	3.15
perl-Errno	1.30	1.30
perl-Exporter	5.74	5.74
perl-Fcntl	1.13	1.13
perl-File-Basename	2.85	2.85
perl-File-Path	2.18	2.18
perl-File-stat	1.09	1.09
perl-File-Temp	0.231.100	0.231.100
perl-Getopt-Long	2.52	2.52
perl-Getopt-Std	1.12	1.12
perl-HTTP-Tiny	0.078	0.078
perl-if	0.60.800	0.60.800
perl-interpreter	5.32.1	5.32.1

Pacote	AMI	VMware OVA
perl-I0	1.43	1.43
perl-IPC-Open3	1.21	1.21
perl-libs	5.32.1	5.32.1
perl-MIME-Base64	3.16	3.16
perl-mro	1.23	1.23
perl-overload	1.31	1.31
perl-overloading	0.02	0.02
perl-parent	0.238	0.238
perl-PathTools	3.78	3.78
perl-Pod-Escapes	1.07	1.07
perl-podlators	4.14	4.14
perl-Pod-Perldoc	3.28.01	3.28.01
perl-Pod-Simple	3.42	3.42
perl-Pod-Usage	2.01	2.01
perl-POSIX	1.94	1.94
perl-Scalar-List-Utils	1.56	1.56
perl-SelectSaver	1.02	1.02
perl-Socket	2.032	2.032
perl-srpm-macros	1	1
perl-Storable	3.21	3.21

Pacote	AMI	VMware OVA
perl-subs	1.03	1.03
perl-Symbol	1.08	1.08
perl-Term-ANSIColor	5.01	5.01
perl-Term-Cap	1.17	1.17
perl-Text-ParseWords	3.30	3.30
perl-Text-Tabs+Wrap	2021.0726	2021.0726
perl-Time-Local	1.300	1.300
perl-vars	1.05	1.05
pkgconf	1.8.0	1.8.0
pkgconf-m4	1.8.0	1.8.0
pkgconf-pkg-config	1.8.0	1.8.0
polycoreutils	3.4	3.4
polycoreutils-python-utils	3.4	
popt	1.18	1.18
procps-ng	3.3.17	3.3.17
protobuf-c	1.4.1	1.4.1
psacct	6.6.4	6.6.4
psmisc	23.4	23.4
publicsuffix-list-dafsa	20240212	20240212

Pacote	AMI	VMware OVA
python3	3.9.16	3.9.16
python3-attrs	20.3.0	20.3.0
python3-audit	3.0.6	3.0.6
python3-awscrt	0.19.19	0.19.19
python3-babel	2.9.1	2.9.1
python3-cffi	1.14.5	1.14.5
python3-chardet	4.0.0	4.0.0
python3-colorama	0.4.4	0.4.4
python3-configobj	5.0.6	5.0.6
python3-cryptography	36.0.1	36.0.1
python3-daemon	2.3.0	
python3-dateutil	2.8.1	2.8.1
python3-dbus	1.2.18	1.2.18
python3-distro	1.5.0	1.5.0
python3-dnf	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0
python3-docutils	0.16	0.16
python3-gpg	1.15.1	1.15.1
python3-hawkey	0.69.0	0.69.0
python3-idna	(2.10)	(2.10)

Pacote	AMI	VMware OVA
python3-jinja2	2.11.3	2.11.3
python3-jmespath	0.10.0	0.10.0
python3-jsonpatch	1.21	1.21
python3-jsonpointer	2.0	2.0
python3-jsonschema	3.2.0	3.2.0
python3-libcomps	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0
python3-libs	3.9.16	3.9.16
python3-libselenium	3.4	3.4
python3-libsemanage	3.4	3.4
python3-libstorage mgmt	1.9.4	1.9.4
python3-lockfile	0.12.2	
python3-markupsafe	1.1.1	1.1.1
python3-netifaces	0.10.6	0.10.6
python3-oauthlib	3.0.2	3.0.2
python3-pip-wheel	21.3.1	21.3.1
python3-ply	3.11	3.11
python3-policycore utils	3.4	3.4
python3-prettytable	0.7.2	0.7.2

Pacote	AMI	VMware OVA
python3-prompt-toolkit	3.0.24	3.0.24
python3-pycparser	2.20	2.20
python3-pyrsistent	0.17.3	0.17.3
python3-pyserial	3.4	3.4
python3-pysocks	1.7.1	1.7.1
python3-pytz	2022.7.1	2022.7.1
python3-pyyaml	5.4.1	5.4.1
python3-requests	2.25.1	2.25.1
python3-rpm	4.16.1.3	4.16.1.3
python3-ruamel-yaml	0.16.6	0.16.6
python3-ruamel-yaml-clib	0.1.2	0.1.2
python3-setuptools	4.4.1	4.4.1
python3-setuptools	59.6.0	59.6.0
python3-setuptools-wheel	59.6.0	59.6.0
python3-six	1.15.0	1.15.0
python3-systemd	235	235
python3-urllib3	1.25.10	1.25.10
python3-wcwidth	0.2.5	0.2.5

Pacote	AMI	VMware OVA
python-chevron	0.13.1	
python-srpm-macros	3.9	3.9
quota	4.06	4.06
quota-nls	4.06	4.06
readline	8.1	8.1
rng-tools	6.14	6.14
rootfiles	8.1	8.1
rpcbind	1.2.6	1.2.6
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	4.16.1.3
rsync	3.2.6	3.2.6
rust-srpm-macros	21	21
sbsigntools	0.9.4	0.9.4
screen	4.8.0	4.8.0
sed	4.8	4.8
selinux-policy	38.1.45	38.1.45

Pacote	AMI	VMware OVA
selinux-policy-targeted	38.1.45	38.1.45
setup	2.13.7	2.13.7
shadow-utils	4,9	4,9
slang	2.3.2	2.3.2
sqlite-libs	3.40.0	3.40.0
sssd-client	2.9.4	2.9.4
sssd-common	2.9.4	
sssd-kcm	2.9.4	
sssd-nfs-idmap	2.9.4	
strace	6.8	6.8
sudo	1.9.15	1.9.15
sysctl-defaults	1.0	1,0
sysstat	12.5.6	12.5.6
systemd	252.23	252.23
systemd-libs	252.23	252.23
systemd-networkd	252.23	252.23
systemd-pam	252.23	252.23
systemd-resolved	252.23	252.23
systemd-udev	252.23	252.23
system-release	2023.6.20241031	2023.6.20241031

Pacote	AMI	VMware OVA
systemtap-runtime	4.8	4.8
tar	1.34	1.34
tbb	2020.3	2020.3
tcpdump	4.99.1	4.99.1
tcsh	6.24.07	6.24.07
time	1.9	1.9
traceroute	2.1.3	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	2.2	2.2
userspace-rcu	0.12.1	0.12.1
util-linux	2.37.4	2.37.4
util-linux-core	2.37.4	2.37.4
vim-common	9.0.2153	9.0.2153
vim-data	9.0.2153	9.0.2153
vim-enhanced	9.0.2153	9.0.2153
vim-filesystem	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153
wget	1.21.3	1.21.3
which	2.21	2.21

Pacote	AMI	VMware OVA
words	3.0	3.0
xfsdump	3.1.11	3.1.11
xfspgrog	5.18.0	5.18.0
xmlsec1		1.2.33
xmlsec1-openssl		1.2.33
xxd	9.0.2153	9.0.2153
xxhash-libs	0.8.0	0.8.0
xz	5.2.5	5.2.5
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0
zlib	1.2.11	1.2.11
zram-generator	1.1.2	
zram-generator-def aults	1.1.2	
zstd	1.5.5	1.5.5

Comparar pacotes instalados na AMI padrão do Amazon Linux 2023 com a imagem do AL2023 para Hyper-V

Uma comparação dos RPMs presentes na AMI padrão do AL2023 com os RPMs presentes na imagem do AL2023 para Hyper-V.

Pacote	AMI	Hyper-V VHDX
acl	2.3.1	2.3.1
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chrony-config	4.3	
amazon-ec2-net-utils	2.4.1	
amazon-linux-onprem		1.2
amazon-linux-repo-cdn		2023.4.20240319
amazon-linux-repo-s3	2023.4.20240319	
amazon-linux-sb-keys	2023.1	2023.1
amazon-onprem-netw ork		1.2
amazon-rpm-config	228	228
amazon-ssm-agent	3.2.2303.0	3.2.2303.0
at	3.1.23	3.1.23
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2.14.5	2.14.5
basesystem	11	11

Pacote	AMI	Hyper-V VHDX
bash	5.2.15	5.2.15
bash-completion	2.11	2.11
bc	1.07.1	1.07.1
bind-libs	9.16.48	9.16.48
bind-license	9.16.48	9.16.48
bind-utils	9.16.48	9.16.48
binutils	2.39	2.39
boost-filesystem	1.75.0	1.75.0
boost-system	1.75.0	1.75.0
boost-thread	1.75.0	1.75.0
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023.2.64	2023.2.64
c-ares	1.19.0	
checkpolicy	3.4	3.4
chkconfig	1.15	1.15
chrony	4.3	4.3
cloud-init	22.2.2	22.2.2
cloud-init-cfg-ec2	22.2.2	
cloud-init-cfg-onp rem		22.2.2

Pacote	AMI	Hyper-V VHDX
cloud-utils-growpart	0.31	0.31
coreutils	8.32	8.32
coreutils-common	8.32	8.32
cpio	2.13	2.13
cracklib	2.9.6	2.9.6
cracklib-dicts	2.9.6	2.9.6
crontabs	1.11	1.11
crypto-policies	20220428	20220428
crypto-policies-scripts	20220428	20220428
cryptsetup	2.6.1	2.6.1
cryptsetup-libs	2.6.1	2.6.1
curl-minimal	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27
cyrus-sasl-plain	2.1.27	2.1.27
dbus	1.12.28	1.12.28
dbus-broker	32	32
dbus-common	1.12.28	1.12.28
dbus-libs	1.12.28	1.12.28
device-mapper	1.02.185	1.02.185
device-mapper-libs	1.02.185	1.02.185

Pacote	AMI	Hyper-V VHDX
diffutils	3.8	3.8
dnf	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2
dnf-plugins-core	4.3.0	4.3.0
dnf-plugin-support-info	1.2	1.2
dnf-utils	4.3.0	4.3.0
dosfstools	4.2	4.2
dracut	055	055
dracut-config-ec2	3.0	
dracut-config-generic	055	055
dwz	0.14	0.14
dyninst	10.2.1	10.2.1
e2fsprogs	1.46.5	1.46.5
e2fsprogs-libs	1.46.5	1.46.5
ec2-hibinit-agent	1.0.8	
ec2-instance-connect	1.1	
ec2-instance-connect-selinux	1.1	

Pacote	AMI	Hyper-V VHDX
ec2-utils	2.2.0	
ed	1.14.2	1.14.2
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38
efivar-libs	38	38
elfutils-debuginfod-client	0.188	0.188
elfutils-default-yama-scope	0.188	0.188
elfutils-libelf	0.188	0.188
elfutils-libs	0.188	0.188
ethtool	5.15	5.15
expat	2.5.0	2.5.0
file	5.39	5.39
file-libs	5.39	5.39
filesystem	3.14	3.14
findutils	4.8.0	4.8.0
fonts-srpm-macros	2.0.5	2.0.5
fstrm	0.6.1	0.6.1
fuse-libs	2.9.9	2.9.9

Pacote	AMI	Hyper-V VHDX
gawk	5.1.0	5.1.0
gdbm-libs	1.19	1.19
gdisk	1.0.8	1.0.8
gettext	0.21	0.21
gettext-libs	0.21	0.21
ghc-srpm-macros	1.5.0	1.5.0
glib2	2.74.7	2.74.7
glibc	2.34	2.34
glibc-all-langpacks	2.34	2.34
glibc-common	2.34	2.34
glibc-gconv-extra	2.34	2.34
glibc-locale-source	2.34	2.34
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gnutls	3.8.0	3.8.0
go-srpm-macros	3.2.0	3.2.0
gpgme	1.15.1	1.15.1
gpm-libs	1.20.7	1.20.7
grep	3.8	3.8
groff-base	1.22.4	1.22.4

Pacote	AMI	Hyper-V VHDX
grub2-common	2.06	2.06
grub2-efi-x64-ec2	2.06	2.06
grub2-pc		2.06
grub2-pc-modules	2.06	2.06
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.40	8.40
gssproxy	0.8.4	0.8.4
gzip	1.12	1.12
hostname	3.23	3.23
hunspell	1.7.0	1.7.0
hunspell-en	0.20140811.1	0.20140811.1
hunspell-en-GB	0.20140811.1	0.20140811.1
hunspell-en-US	0.20140811.1	0.20140811.1
hunspell-filesystem	1.7.0	1.7.0
hwdata	0.353	0.353
hyperv-daemons		0
hyperv-daemons-lic ense		0
hypervfcopyd		0
hypervkvpd		0

Pacote	AMI	Hyper-V VHDX
hyperv-tools		0
hypervvssd		0
info	6.7	6.7
inih	49	49
initscripts	10.09	10.09
iproute	5.10.0	5.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jitterentropy	3.4.1	3.4.1
jq	1.7.1	1.7.1
json-c	0.14	0.14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0
kernel	6.1.79	6.1.79
kernel-livepatch-r epo-cdn		2023.4.20240319
kernel-livepatch-r epo-s3	2023.4.20240319	
kernel-modules-extra		6.1.79

Pacote	AMI	Hyper-V VHDX
kernel-modules-extra-common		6.1.79
kernel-srpm-macros	1.0	1,0
kernel-tools	6.1.79	6.1.79
keyutils	1.6.3	1.6.3
keyutils-libs	1.6.3	1.6.3
kmod	29	29
kmod-libs	29	29
kpatch-runtime	0.9.7	0.9.7
krb5-libs	1.21	1.21
less	608	608
libacl	2.3.1	2.3.1
libaio	0.3.111	0.3.111
libarchive	3.5.3	3.5.3
libargon2	20171227	20171227
libassuan	2.5.5	2.5.5
libattr	2.5.1	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2.37.4	2.37.4
libcap	2.48	2.48
libcap-ng	0.8.2	0.8.2

Pacote	AMI	Hyper-V VHDX
libcbor	0.7.0	0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1.46.5	1.46.5
libcomps	0.1.20	0.1.20
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0.69.0	0.69.0
libeconf	0.4.0	0.4.0
libedit	3.1	3.1
libev	4.33	4.33
libevent	2.1.12	2.1.12
libfdisk	2.37.4	2.37.4
libffi	3.4.4	3.4.4
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1
libgpg-error	1.42	1.42

Pacote	AMI	Hyper-V VHDX
libibverbs	48.0	48.0
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1
libkcapi	1.4.0	1.4.0
libkcapi-hmaccalc	1.4.0	1.4.0
libldb	2.6.2	
libmaxminddb	1.5.2	1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.4	1.0.4
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libnfsidmap	2.5.4	2.5.4
libnghttp2	1.57.0	1.57.0
libnl3	3.5.0	3.5.0
libpath_utils	0.2.1	0.2.1
libpcap	1.10.1	1.10.1
libpipeline	1.5.3	1.5.3
libpkgconf	1.8.0	1.8.0
libpsl	0.21.1	0.21.1
libpwquality	1.4.4	1.4.4

Pacote	AMI	Hyper-V VHDX
libref_array	0.1.5	0.1.5
librepo	1.14.5	1.14.5
libreport-filesystem	2.15.2	2.15.2
libseccomp	2.5.3	2.5.3
libselinux	3.4	3.4
libselinux-utils	3.4	3.4
libsemanage	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2.13	2.13
libsmartcols	2.37.4	2.37.4
libsolv	0.7.22	0.7.22
libss	1.46.5	1.46.5
libsss_certmap	2.9.4	
libsss_idmap	2.9.4	2.9.4
libsss_nss_idmap	2.9.4	2.9.4
libsss_sudo	2.9.4	
libstdc++	11.4.1	11.4.1
libstoragegmt	1.9.4	1.9.4
libtalloc	2.3.4	
libtasn1	4.19.0	4.19.0

Pacote	AMI	Hyper-V VHDX
libtdb	1.4.7	
libtevent	0.13.0	
libtextstyle	0.21	0.21
libtirpc	1.3.3	1.3.3
libunistring	0.9.10	0.9.10
libuser	0.63	0.63
libutempter	1.2.1	1.2.1
libuuid	2.37.4	2.37.4
libuv	1.47.0	1.47.0
libverto	0.3.2	0.3.2
libverto-libev	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33
libxml2	2.10.4	2.10.4
libyaml	0.2.5	0.2.5
libzstd	1.5.5	1.5.5
lm_sensors-libs	3.6.0	3.6.0
lmdb-libs	0.9.29	0.9.29
logrotate	3.20.1	3.20.1
lsof	4.94.0	4.94.0
lua-libs	5.4.4	5.4.4

Pacote	AMI	Hyper-V VHDX
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4
man-db	2.9.3	2.9.3
man-pages	5.10	5.10
microcode_ctl	2.1	2.1
mpfr	4.1.0	4.1.0
nano	5,8	5,8
ncurses	6.2	6.2
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
nettle	3.8	3.8
net-tools	2.0	2.0
newt	0.52.21	0.52.21
nfs-utils	2.5.4	2.5.4
npth	1,6	1.6
nspr	4.35.0	4.35.0
nss	3.90.0	3.90.0
nss-softokn	3.90.0	3.90.0
nss-softokn-freebl	3.90.0	3.90.0
nss-sysinit	3.90.0	3.90.0

Pacote	AMI	Hyper-V VHDX
nss-util	3.90.0	3.90.0
ntsysv	1.15	1.15
numactl-libs	2.0.14	2.0.14
ocaml-srpm-macros	6	6
oniguruma	6.9.7.1	6.9.7.1
openblas-srpm-macros	2	2
openldap	2.4.57	2.4.57
openssh	8.7p1	8.7p1
openssh-clients	8.7p1	8.7p1
openssh-server	8.7p1	8.7p1
openssl	3.0.8	3.0.8
openssl-libs	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12
os-prober	1.77	1.77
p11-kit	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1
package-notes-srpm-macros	0.4	0.4
pam	1.5.1	1.5.1
parted	3.4	3.4
passwd	0.80	0.80

Pacote	AMI	Hyper-V VHDX
pciutils	3.7.0	3.7.0
pciutils-libs	3.7.0	3.7.0
pcre2	10.40	10.40
pcre2-syntax	10.40	10.40
perl-Carp	1.50	1.50
perl-Class-Struct	0.66	0.66
perl-constant	1.33	1.33
perl-DynaLoader	1.47	1.47
perl-Encode	3.15	3.15
perl-Errno	1.30	1.30
perl-Exporter	5.74	5.74
perl-Fcntl	1.13	1.13
perl-File-Basename	2.85	2.85
perl-File-Path	2.18	2.18
perl-File-stat	1.09	1.09
perl-File-Temp	0.231.100	0.231.100
perl-Getopt-Long	2.52	2.52
perl-Getopt-Std	1.12	1.12
perl-HTTP-Tiny	0.078	0.078
perl-if	0.60.800	0.60.800

Pacote	AMI	Hyper-V VHDX
perl-interpreter	5.32.1	5.32.1
perl-IO	1.43	1.43
perl-IPC-Open3	1.21	1.21
perl-libs	5.32.1	5.32.1
perl-MIME-Base64	3.16	3.16
perl-mro	1.23	1.23
perl-overload	1.31	1.31
perl-overloading	0.02	0.02
perl-parent	0.238	0.238
perl-PathTools	3.78	3.78
perl-Pod-Escapes	1.07	1.07
perl-podlators	4.14	4.14
perl-Pod-Perldoc	3.28.01	3.28.01
perl-Pod-Simple	3.42	3.42
perl-Pod-Usage	2.01	2.01
perl-POSIX	1.94	1.94
perl-Scalar-List-Utils	1.56	1.56
perl-SelectSaver	1.02	1.02
perl-Socket	2.032	2.032
perl-srpm-macros	1	1

Pacote	AMI	Hyper-V VHDX
perl-Storable	3.21	3.21
perl-subvars	1.03	1.03
perl-Symbol	1.08	1.08
perl-Term-ANSIColor	5.01	5.01
perl-Term-Cap	1.17	1.17
perl-Text-ParseWords	3.30	3.30
perl-Text-Tabs+Wrap	2021.0726	2021.0726
perl-Time-Local	1.300	1.300
perl-vars	1.05	1.05
pkgconf	1.8.0	1.8.0
pkgconf-m4	1.8.0	1.8.0
pkgconf-pkg-config	1.8.0	1.8.0
policycoreutils	3.4	3.4
policycoreutils-python-utils	3.4	
popt	1.18	1.18
procps-ng	3.3.17	3.3.17
protobuf-c	1.4.1	1.4.1
psacct	6.6.4	6.6.4
psmisc	23.4	23.4

Pacote	AMI	Hyper-V VHDX
publicsuffix-list-dafsa	20240212	20240212
python3	3.9.16	3.9.16
python3-attrs	20.3.0	20.3.0
python3-audit	3.0.6	3.0.6
python3-awscli	0.19.19	0.19.19
python3-babel	2.9.1	2.9.1
python3-cffi	1.14.5	1.14.5
python3-chardet	4.0.0	4.0.0
python3-colorama	0.4.4	0.4.4
python3-configobj	5.0.6	5.0.6
python3-cryptography	36.0.1	36.0.1
python3-daemon	2.3.0	
python3-dateutil	2.8.1	2.8.1
python3-dbus	1.2.18	1.2.18
python3-distro	1.5.0	1.5.0
python3-dnf	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0
python3-docutils	0.16	0.16
python3-gpg	1.15.1	1.15.1

Pacote	AMI	Hyper-V VHDX
python3-hawkey	0.69.0	0.69.0
python3-idna	(2.10)	(2.10)
python3-jinja2	2.11.3	2.11.3
python3-jmespath	0.10.0	0.10.0
python3-jsonpatch	1.21	1.21
python3-jsonpointer	2.0	2.0
python3-jsonschema	3.2.0	3.2.0
python3-libcomps	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0
python3-libs	3.9.16	3.9.16
python3-libselenium	3.4	3.4
python3-libsemanage	3.4	3.4
python3-libstorage mgmt	1.9.4	1.9.4
python3-lockfile	0.12.2	
python3-markupsafe	1.1.1	1.1.1
python3-netifaces	0.10.6	0.10.6
python3-oauthlib	3.0.2	3.0.2
python3-pip-wheel	21.3.1	21.3.1
python3-ply	3.11	3.11

Pacote	AMI	Hyper-V VHDX
python3-policycore utils	3.4	3.4
python3-prettytable	0.7.2	0.7.2
python3-prompt-too lkit	3.0.24	3.0.24
python3-pycparser	2.20	2.20
python3-pyrsistent	0.17.3	0.17.3
python3-pyserial	3.4	3.4
python3-pysocks	1.7.1	1.7.1
python3-pytz	2022.7.1	2022.7.1
python3-pyyaml	5.4.1	5.4.1
python3-requests	2.25.1	2.25.1
python3-rpm	4.16.1.3	4.16.1.3
python3-ruamel-yaml	0.16.6	0.16.6
python3-ruamel-yaml- clib	0.1.2	0.1.2
python3-setools	4.4.1	4.4.1
python3-setuptools	59.6.0	59.6.0
python3-setuptools- wheel	59.6.0	59.6.0
python3-six	1.15.0	1.15.0
python3-systemd	235	235

Pacote	AMI	Hyper-V VHDX
python3-urllib3	1.25.10	1.25.10
python3-wcwidth	0.2.5	0.2.5
python-chevron	0.13.1	
python-srpm-macros	3.9	3.9
quota	4.06	4.06
quota-nls	4.06	4.06
readline	8.1	8.1
rng-tools	6.14	6.14
rootfiles	8.1	8.1
rpcbind	1.2.6	1.2.6
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	4.16.1.3
rsync	3.2.6	3.2.6
rust-srpm-macros	21	21
sbsigntools	0.9.4	0.9.4
screen	4.8.0	4.8.0

Pacote	AMI	Hyper-V VHDX
sed	4.8	4.8
selinux-policy	37.22	37.22
selinux-policy-targeted	37.22	37.22
setup	2.13.7	2.13.7
shadow-utils	4,9	4,9
slang	2.3.2	2.3.2
sqlite-libs	3.40.0	3.40.0
sssd-client	2.9.4	2.9.4
sssd-common	2.9.4	
sssd-kcm	2.9.4	
sssd-nfs-idmap	2.9.4	
strace	5.16	5.16
sudo	1.9.14	1.9.14
sysctl-defaults	1.0	1,0
sysstat	12.5.6	12.5.6
systemd	252.16	252.16
systemd-libs	252.16	252.16
systemd-networkd	252.16	252.16
systemd-pam	252.16	252.16
systemd-resolved	252.16	252.16

Pacote	AMI	Hyper-V VHDX
systemd-udev	252.16	252.16
system-release	2023.4.20240319	2023.4.20240319
systemtap-runtime	4.8	4.8
tar	1.34	1.34
tbb	2020.3	2020.3
tcpdump	4.99.1	4.99.1
tcsh	6.24.07	6.24.07
time	1.9	1.9
traceroute	2.1.3	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	2.2	2.2
userspace-rcu	0.12.1	0.12.1
util-linux	2.37.4	2.37.4
util-linux-core	2.37.4	2.37.4
vim-common	9.0.2153	9.0.2153
vim-data	9.0.2153	9.0.2153
vim-enhanced	9.0.2153	9.0.2153
vim-filesystem	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153

Pacote	AMI	Hyper-V VHDX
wget	1.21.3	1.21.3
which	2.21	2.21
words	3.0	3.0
xfsdump	3.1.11	3.1.11
xfsprogs	5.18.0	5.18.0
xxd	9.0.2153	9.0.2153
xxhash-libs	0.8.0	0.8.0
xz	5.2.5	5.2.5
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0
zlib	1.2.11	1.2.11
zram-generator	1.1.2	
zram-generator-def aults	1.1.2	
zstd	1.5.5	1.5.5

Identificar instâncias e versões do Amazon Linux

Pode ser importante determinar a qual distribuição do Linux e a qual versão dessa distribuição uma imagem ou instância do sistema operacional pertence. O Amazon Linux fornece mecanismos para identificar o Amazon Linux em relação a outras distribuições do Linux, bem como para identificar a qual versão do Amazon Linux a imagem pertence.

Esta seção abordará os diferentes métodos que podem ser usados e as respectivas limitações, além de apresentar alguns exemplos de uso.

Tópicos

- [Usar o padrão de os-release](#)
- [Específicos para Amazon Linux](#)
- [Código de exemplo para detecção do sistema operacional](#)

Usar o padrão de **os-release**

O Amazon Linux está em conformidade com o [padrão de os-release](#) para a identificação de distribuições do Linux. Esse arquivo fornece informações legíveis por máquina sobre a identificação do sistema operacional e as informações da versão.

Note

O padrão determina que `/etc/os-release` seja analisado primeiro, seguido por `/usr/lib/os-release`. Deve-se tomar cuidado para seguir o padrão em relação aos nomes e caminhos dos arquivos.

Tópicos

- [Principais diferenças de identificação](#)
- [Tipos de campos: legíveis por máquina versus legíveis por humanos](#)
- [/etc/os-releaseExemplos da](#)
- [Comparação com outras distribuições](#)

Principais diferenças de identificação

O `os-release` é encontrado em `/etc/os-release` ou, se não estiver presente, em `/usr/lib/os-release`. Consulte o [padrão de os-release](#) para obter informações completas.

A maneira mais confiável de determinar se uma instância está executando o Amazon Linux é verificar o campo `ID` em `os-release`.

A maneira mais confiável de determinar a distinção entre as versões é verificar o campo `VERSION_ID` em `os-release`:

- AMI do Amazon Linux: `VERSION_ID` contém uma versão baseada em data (por exemplo, `2018.03`)
- AL2: `VERSION_ID="2"`
- AL2023: `VERSION_ID="2023"`

Note

Lembre-se de que `VERSION_ID` é um campo legível por máquina destinado ao uso programático, enquanto `Pretty_Name` é projetado para exibição aos usuários. Consulte [the section called “Tipos de campos”](#) para obter mais informações sobre os tipos de campos.

Tipos de campos: legíveis por máquina versus legíveis por humanos

O arquivo `/etc/os-release` (ou `/usr/lib/os-release` se `/etc/os-release` não existir) contém dois tipos de campos: campos legíveis por máquina, destinados ao uso programático, e campos legíveis por humanos, destinados à apresentação aos usuários.

Campos legíveis por máquina

Esses campos usam formatos padronizados e são destinados ao processamento por scripts, gerenciadores de pacotes e outras ferramentas automatizadas. Eles contêm apenas letras minúsculas, números e pontuação limitada (pontos, sublinhados e hifens).

- `ID`: identificador do sistema operacional. O Amazon Linux usa `amzn` em todas as versões, diferenciando-o de outras distribuições, como Debian (`debian`), Ubuntu (`ubuntu`) ou Fedora (`fedora`).

- `VERSION_ID`: versão do sistema operacional para uso programático (por exemplo, `2023`).
- `ID_LIKE`: lista separada por espaços de distribuições relacionadas (por exemplo, `fedora`).
- `VERSION_CODENAME`: codinome de lançamento para scripts (por exemplo, `karoo`).
- `VARIANT_ID`: identificador da variante para decisões programáticas.
- `BUILD_ID`: identificador da compilação para imagens do sistema.
- `IMAGE_ID`: identificador da imagem para ambientes em contêineres.
- `PLATFORM_ID`: identificador da plataforma (por exemplo, `platform:a12023`).

Campos legíveis por humanos

Esses campos são destinados à exibição para os usuários e podem conter espaços, letras maiúsculas e minúsculas e texto descritivo. Eles devem ser usados ao apresentar informações do sistema operacional nas interfaces de usuário.

- `NAME`: nome do sistema operacional para exibição (por exemplo, `Amazon Linux`).
- `PRETTY_NAME`: nome completo do sistema operacional com versão para exibição (por exemplo, `Amazon Linux 2023.8.20250721`).
- `VERSION`: informações da versão adequadas para apresentação ao usuário.
- `VARIANT`: nome da variante ou edição para exibição (por exemplo, `Server Edition`).

Outros campos de informações

Esses campos fornecem metadados adicionais sobre o sistema operacional:

- `HOME_URL`: URL da página inicial do projeto.
- `DOCUMENTATION_URL`: URL da documentação.
- `SUPPORT_URL`: URL de informações de suporte.
- `BUG_REPORT_URL`: URL de relatórios de bugs.
- `VENDOR_NAME`: nome do fornecedor.
- `VENDOR_URL`: URL do fornecedor.
- `SUPPORT_END`: data do fim do suporte no formato `AAAA-MM-DD`.
- `CPE_NAME`: identificador de enumeração de plataforma comum.

- `ANSI_COLOR`: código de cores ANSI para exibição do terminal.

Ao escrever scripts ou aplicações que precisam identificar o Amazon Linux de forma programática, use os campos legíveis por máquina, como `ID` e `VERSION_ID`. Ao exibir informações do sistema operacional aos usuários, use os campos legíveis por humanos, como `PRETTY_NAME`.

`/etc/os-release` Exemplos da

O conteúdo do arquivo `/etc/os-release` varia entre as versões do Amazon Linux:

AL2023

```
[ec2-user ~]$ cat /etc/os-release
```

```
NAME="Amazon Linux"
VERSION="2023"
ID="amzn"
ID_LIKE="fedora"
VERSION_ID="2023"
PLATFORM_ID="platform:al2023"
PRETTY_NAME="Amazon Linux 2023.8.20250721"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2023"
HOME_URL="https://aws.amazon.com/linux/amazon-linux-2023/"
DOCUMENTATION_URL="https://docs.aws.amazon.com/linux/"
SUPPORT_URL="https://aws.amazon.com/premiumsupport/"
BUG_REPORT_URL="https://github.com/amazonlinux/amazon-linux-2023"
VENDOR_NAME="AWS"
VENDOR_URL="https://aws.amazon.com/"
SUPPORT_END="2029-06-30"
```

AL2

```
[ec2-user ~]$ cat /etc/os-release
```

```
NAME="Amazon Linux"
VERSION="2"
ID="amzn"
ID_LIKE="centos rhel fedora"
VERSION_ID="2"
```

```
PRETTY_NAME="Amazon Linux 2"  
ANSI_COLOR="0;33"  
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2"  
HOME_URL="https://amazonlinux.com/"  
SUPPORT_END="2026-06-30"
```

Amazon Linux AMI

```
[ec2-user ~]$ cat /etc/os-release
```

```
NAME="Amazon Linux AMI"  
VERSION="2018.03"  
ID="amzn"  
ID_LIKE="rhel fedora"  
VERSION_ID="2018.03"  
PRETTY_NAME="Amazon Linux AMI 2018.03"  
ANSI_COLOR="0;33"  
CPE_NAME="cpe:/o:amazon:linux:2018.03:ga"  
HOME_URL="http://aws.amazon.com/amazon-linux-ami/"
```

Comparação com outras distribuições

Para entender como o Amazon Linux se encaixa no ecossistema mais amplo do Linux, compare o formato de `/etc/os-release` com outras distribuições principais:

Fedora

```
[ec2-user ~]$ cat /etc/os-release
```

```
NAME="Fedora Linux"  
VERSION="42 (Container Image)"  
RELEASE_TYPE=stable  
ID=fedora  
VERSION_ID=42  
VERSION_CODENAME=""  
PLATFORM_ID="platform:f42"  
PRETTY_NAME="Fedora Linux 42 (Container Image)"  
ANSI_COLOR="0;38;2;60;110;180"  
LOGO=fedora-logo-icon  
CPE_NAME="cpe:/o:fedoraproject:fedora:42"
```

```
DEFAULT_HOSTNAME="fedora"
HOME_URL="https://fedoraproject.org/"
DOCUMENTATION_URL="https://docs.fedoraproject.org/en-US/fedora/f42/system-
administrators-guide/"
SUPPORT_URL="https://ask.fedoraproject.org/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"
REDHAT_BUGZILLA_PRODUCT="Fedora"
REDHAT_BUGZILLA_PRODUCT_VERSION=42
REDHAT_SUPPORT_PRODUCT="Fedora"
REDHAT_SUPPORT_PRODUCT_VERSION=42
SUPPORT_END=2026-05-13
VARIANT="Container Image"
VARIANT_ID=container
```

Debian

```
[ec2-user ~]$ cat /etc/os-release
```

```
PRETTY_NAME="Debian GNU/Linux 12 (bookworm)"
NAME="Debian GNU/Linux"
VERSION_ID="12"
VERSION="12 (bookworm)"
VERSION_CODENAME=bookworm
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
```

Ubuntu

```
[ec2-user ~]$ cat /etc/os-release
```

```
PRETTY_NAME="Ubuntu 24.04.2 LTS"
NAME="Ubuntu"
VERSION_ID="24.04"
VERSION="24.04.2 LTS (Noble Numbat)"
VERSION_CODENAME=noble
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
```



```
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=noble
LOGO=ubuntu-logo
```

Observe como os campos legíveis por máquina fornecem identificação consistente em todas as distribuições:

- ID: identifica o sistema operacional de maneira exclusiva (amzn para Amazon Linux, fedora para Fedora, debian para Debian, ubuntu para Ubuntu).
- ID_LIKE: mostra as relações das distribuições (o Amazon Linux usa fedora (AL2023) ou centos rhel fedora (AL2), enquanto o Ubuntu mostra debian para indicar sua herança do Debian).
- VERSION_ID: fornece informações de versão analisáveis por máquina (2023 para AL2023, 42 para Fedora, 12 para Debian, 24.04 para Ubuntu).

Por outro lado, os campos legíveis por humanos foram projetados para serem exibidos aos usuários:

- NAME: nome amigável do sistema operacional (Amazon Linux, Fedora Linux, Debian GNU/Linux, Ubuntu).
- PRETTY_NAME: nome de exibição completo com a versão (Amazon Linux 2023.8.20250721, Fedora Linux 42 (Container Image), Debian GNU/Linux 12 (bookworm), Ubuntu 24.04.2 LTS).
- VERSION: versão legível por humanos com contexto adicional, como codinomes ou tipos de lançamento.

Ao escrever scripts multiplataforma, sempre use os campos legíveis por máquina (ID, VERSION_ID, ID_LIKE) para lógica e decisões, e use os campos legíveis por humanos (PRETTY_NAME, NAME) somente para exibir informações aos usuários.

Específicos para Amazon Linux

Há alguns arquivos específicos do Amazon Linux que podem ser usados para identificar o Amazon Linux e sua versão. O novo código deve usar o padrão de [/etc/os-release](#) para ser compatível entre distribuições. O uso de qualquer arquivo específico do Amazon Linux é desencorajado.

Tópicos

- [O arquivo `/etc/system-release`](#)
- [Arquivo de identificação de imagem](#)
- [Exemplos de arquivos específicos do Amazon Linux](#)

O arquivo `/etc/system-release`

O Amazon Linux contém um arquivo `/etc/system-release` que especifica a versão atual que está instalada. Esse arquivo é atualizado usando gerenciadores de pacotes e, no Amazon Linux, faz parte do pacote `system-release`. Embora algumas outras distribuições, como o Fedora, também tenham esse arquivo, ele não está presente nas distribuições baseadas em Debian, como o Ubuntu.

Note

O arquivo `/etc/system-release` contém uma string legível por humanos e não deve ser usado de forma programática para identificar um sistema operacional ou um lançamento. Em vez disso, use os campos legíveis por máquina em `/etc/os-release` (ou `/usr/lib/os-release` se `/etc/os-release` não existir).

O Amazon Linux também contém uma versão legível por máquina de `/etc/system-release` que acompanha a especificação de enumeração de plataforma comum (CPE) no arquivo `/etc/system-release-cpe`.

Arquivo de identificação de imagem

Cada imagem do Amazon Linux contém um arquivo `/etc/image-id` exclusivo que fornece informações adicionais sobre a imagem original, conforme gerada pela equipe do Amazon Linux. Esse arquivo é específico do Amazon Linux e não é encontrado em outras distribuições do Linux, como Debian, Ubuntu ou Fedora. Esse arquivo contém as seguintes informações sobre a imagem:

- `image_name`, `image_version`, `image_arch`: valores da fórmula de compilação usada para criar a imagem.
- `image_stamp`: valor hexadecimal aleatório exclusivo gerado durante a criação da imagem.
- `image_date`: horário UTC da criação da imagem, no formato AAAAMMDDhhmmss.
- `recipe_name`, `recipe_id`: o nome e o ID da fórmula de compilação usados para criar a imagem.

Exemplos de arquivos específicos do Amazon Linux

As seções a seguir fornecem exemplos dos arquivos de identificação específicos do Amazon Linux para cada versão principal do Amazon Linux.

Note

Em qualquer código do mundo real, `/usr/lib/os-release` deve ser usado se o arquivo `/etc/os-release` não existir.

AL2023

Os exemplos a seguir mostram os arquivos de identificação do AL2023.

Exemplo de `/etc/image-id` para o AL2023:

```
[ec2-user ~]$ cat /etc/image-id
```

```
image_name="al2023-container"  
image_version="2023"  
image_arch="x86_64"  
image_file="al2023-container-2023.8.20250721.2-x86_64"  
image_stamp="822b-1a9e"  
image_date="20250719211531"  
recipe_name="al2023 container"  
recipe_id="89b25f7b-be82-2215-a8eb-6e63-0830-94ea-658d41c4"
```

Exemplo de `/etc/system-release` para o AL2023:

```
[ec2-user ~]$ cat /etc/system-release
```

```
Amazon Linux release 2023.8.20250721 (Amazon Linux)
```

AL2

Os exemplos a seguir mostram os arquivos de identificação do AL2.

Exemplo de `/etc/image-id` para o AL2:

```
[ec2-user ~]$ cat /etc/image-id
```

```
image_name="amzn2-container-raw"  
image_version="2"  
image_arch="x86_64"  
image_file="amzn2-container-raw-2.0.20250721.2-x86_64"  
image_stamp="4126-16ad"  
image_date="20250721225801"  
recipe_name="amzn2 container"  
recipe_id="948422df-a4e6-5fc8-ba89-ef2e-0e1f-e1bb-16f84087"
```

Exemplo de /etc/system-release para o AL2:

```
[ec2-user ~]$ cat /etc/system-release
```

```
Amazon Linux release 2 (Karoo)
```

AMI do Amazon Linux

Os exemplos a seguir mostram os arquivos de identificação para a AMI do Amazon Linux.

Exemplo de /etc/image-id para a AMI do Amazon Linux:

```
[ec2-user ~]$ cat /etc/image-id
```

```
image_name="amzn-container-minimal"  
image_version="2018.03"  
image_arch="x86_64"  
image_file="amzn-container-minimal-2018.03.0.20231218.0-x86_64"  
image_stamp="407d-5ef3"  
image_date="20231218203210"  
recipe_name="amzn container"  
recipe_id="b1e7635e-14e3-dd57-b1ab-7351-edd0-d9e0-ca6852ea"
```

Exemplo de /etc/system-release para a AMI do Amazon Linux:

```
[ec2-user ~]$ cat /etc/system-release
```

Código de exemplo para detecção do sistema operacional

Os exemplos a seguir demonstram como detectar o sistema operacional e a versão de forma programática usando o arquivo `/etc/os-release` (ou `/usr/lib/os-release` se `/etc/os-release` não existir). Esses exemplos mostram como distinguir entre o Amazon Linux e outras distribuições, bem como usar o campo `ID_LIKE` para determinar famílias de distribuições.

O script abaixo é implementado em várias linguagens de programação diferentes, e cada implementação produzirá a mesma saída.

Shell

```
#!/bin/bash

# Function to get a specific field from os-release file
get_os_release_field() {
    local field="$1"
    local os_release_file

    # Find the os-release file
    if [ -f /etc/os-release ]; then
        os_release_file='/etc/os-release'
    elif [ -f /usr/lib/os-release ]; then
        os_release_file='/usr/lib/os-release'
    else
        echo "Error: os-release file not found" >&2
        return 1
    fi

    # Source the file in a subshell and return the requested field.
    #
    # A subshell means that variables from os-release are only available
    # within the subshell, and the main script environment remains clean.
    (
        . "$os_release_file"
        eval "echo \"\${$field}\""
    )
}
```

```
is_amazon_linux() {
    [ "$(get_os_release_field ID)" = "amzn" ]
}

is_fedora() {
    [ "$(get_os_release_field ID)" = "fedora" ]
}

is_ubuntu() {
    [ "$(get_os_release_field ID)" = "ubuntu" ]
}

is_debian() {
    [ "$(get_os_release_field ID)" = "debian" ]
}

# Function to check if this is like Fedora (includes Amazon Linux, CentOS, RHEL,
# etc.)
is_like_fedora() {
    local id="$(get_os_release_field ID)"
    local id_like="$(get_os_release_field ID_LIKE)"
    [ "$id" = "fedora" ] || [[ "$id_like" == *"fedora"* ]]
}

# Function to check if this is like Debian (includes Ubuntu and derivatives)
is_like_debian() {
    local id="$(get_os_release_field ID)"
    local id_like="$(get_os_release_field ID_LIKE)"
    [ "$id" = "debian" ] || [[ "$id_like" == *"debian"* ]]
}

# Get the main fields we'll use multiple times
ID="$(get_os_release_field ID)"
VERSION_ID="$(get_os_release_field VERSION_ID)"
PRETTY_NAME="$(get_os_release_field PRETTY_NAME)"
ID_LIKE="$(get_os_release_field ID_LIKE)"

echo "Operating System Detection Results:"
echo "====="
echo "Is Amazon Linux: $(is_amazon_linux && echo YES || echo NO)"
echo "Is Fedora: $(is_fedora && echo YES || echo NO)"
echo "Is Ubuntu: $(is_ubuntu && echo YES || echo NO)"
echo "Is Debian: $(is_debian && echo YES || echo NO)"
echo "Is like Fedora: $(is_like_fedora && echo YES || echo NO)"
```

```

echo "Is like Debian: $(is_like_debian && echo YES || echo NO)"
echo
echo "Detailed OS Information:"
echo "======"
echo "ID: $ID"
echo "VERSION_ID: $VERSION_ID"
echo "PRETTY_NAME: $PRETTY_NAME"
[ -n "$ID_LIKE" ] && echo "ID_LIKE: $ID_LIKE"

# Amazon Linux specific information
if is_amazon_linux; then
    echo ""
    echo "Amazon Linux Version Details:"
    echo "======"
    case "$VERSION_ID" in
        2018.03)
            echo "Amazon Linux AMI (version 1)"
            ;;
        2)
            echo "Amazon Linux 2"
            ;;
        2023)
            echo "Amazon Linux 2023"
            ;;
        *)
            echo "Unknown Amazon Linux version: $VERSION_ID"
            ;;
    esac

    # Check for Amazon Linux specific files
    [ -f /etc/image-id ] && echo "Amazon Linux image-id file present"
fi

```

Python 3.7-3.9

```

#!/usr/bin/env python3

import os
import sys

def parse_os_release():
    """Parse the os-release file and return a dictionary of key-value pairs."""
    os_release_data = {}

```

```
# Try /etc/os-release first, then /usr/lib/os-release
for path in ['/etc/os-release', '/usr/lib/os-release']:
    if os.path.exists(path):
        try:
            with open(path, 'r') as f:
                for line in f:
                    line = line.strip()
                    if line and not line.startswith('#') and '=' in line:
                        key, value = line.split('=', 1)
                        # Remove quotes if present
                        value = value.strip('"\'')
                        os_release_data[key] = value
            return os_release_data
        except IOError:
            continue

print("Error: os-release file not found")
sys.exit(1)

def is_amazon_linux(os_data):
    """Check if this is Amazon Linux."""
    return os_data.get('ID') == 'amzn'

def is_fedora(os_data):
    """Check if this is Fedora."""
    return os_data.get('ID') == 'fedora'

def is_ubuntu(os_data):
    """Check if this is Ubuntu."""
    return os_data.get('ID') == 'ubuntu'

def is_debian(os_data):
    """Check if this is Debian."""
    return os_data.get('ID') == 'debian'

def is_like_fedora(os_data):
    """Check if this is like Fedora (includes Amazon Linux, CentOS, RHEL, etc.)."""
    if os_data.get('ID') == 'fedora':
        return True
    id_like = os_data.get('ID_LIKE', '')
    return 'fedora' in id_like

def is_like_debian(os_data):
```



```

    """Check if this is like Debian (includes Ubuntu and derivatives)."""
    if os_data.get('ID') == 'debian':
        return True
    id_like = os_data.get('ID_LIKE', '')
    return 'debian' in id_like

def main():
    # Parse os-release file
    os_data = parse_os_release()

    # Display results
    print("Operating System Detection Results:")
    print("=====")
    print(f"Is Amazon Linux: {'YES' if is_amazon_linux(os_data) else 'NO'}")
    print(f"Is Fedora: {'YES' if is_fedora(os_data) else 'NO'}")
    print(f"Is Ubuntu: {'YES' if is_ubuntu(os_data) else 'NO'}")
    print(f"Is Debian: {'YES' if is_debian(os_data) else 'NO'}")
    print(f"Is like Fedora: {'YES' if is_like_fedora(os_data) else 'NO'}")
    print(f"Is like Debian: {'YES' if is_like_debian(os_data) else 'NO'}")

    # Additional information
    print()
    print("Detailed OS Information:")
    print("=====")
    print(f"ID: {os_data.get('ID', '')}")
    print(f"VERSION_ID: {os_data.get('VERSION_ID', '')}")
    print(f"PRETTY_NAME: {os_data.get('PRETTY_NAME', '')}")
    if os_data.get('ID_LIKE'):
        print(f"ID_LIKE: {os_data.get('ID_LIKE')}")

    # Amazon Linux specific information
    if is_amazon_linux(os_data):
        print()
        print("Amazon Linux Version Details:")
        print("=====")
        version_id = os_data.get('VERSION_ID', '')
        if version_id == '2018.03':
            print("Amazon Linux AMI (version 1)")
        elif version_id == '2':
            print("Amazon Linux 2")
        elif version_id == '2023':
            print("Amazon Linux 2023")
        else:
            print(f"Unknown Amazon Linux version: {version_id}")

```

```
# Check for Amazon Linux specific files
if os.path.exists('/etc/image-id'):
    print("Amazon Linux image-id file present")

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

Python 3.10+

```
#!/usr/bin/env python3

import os
import sys
import platform

def is_amazon_linux(os_data):
    """Check if this is Amazon Linux."""
    return os_data.get('ID') == 'amzn'

def is_fedora(os_data):
    """Check if this is Fedora."""
    return os_data.get('ID') == 'fedora'

def is_ubuntu(os_data):
    """Check if this is Ubuntu."""
    return os_data.get('ID') == 'ubuntu'

def is_debian(os_data):
    """Check if this is Debian."""
    return os_data.get('ID') == 'debian'

def is_like_fedora(os_data):
    """Check if this is like Fedora (includes Amazon Linux, CentOS, RHEL, etc.)."""
    if os_data.get('ID') == 'fedora':
        return True
    id_like = os_data.get('ID_LIKE', '')
    return 'fedora' in id_like

def is_like_debian(os_data):
    """Check if this is like Debian (includes Ubuntu and derivatives)."""
    if os_data.get('ID') == 'debian':
        return True
```

```

    id_like = os_data.get('ID_LIKE', '')
    return 'debian' in id_like

def main():
    # Parse os-release file using the standard library function (Python 3.10+)
    try:
        os_data = platform.freedesktop_os_release()
    except OSError:
        print("Error: os-release file not found")
        sys.exit(1)

    # Display results
    print("Operating System Detection Results:")
    print("=====")
    print(f"Is Amazon Linux: {'YES' if is_amazon_linux(os_data) else 'NO'}")
    print(f"Is Fedora: {'YES' if is_fedora(os_data) else 'NO'}")
    print(f"Is Ubuntu: {'YES' if is_ubuntu(os_data) else 'NO'}")
    print(f"Is Debian: {'YES' if is_debian(os_data) else 'NO'}")
    print(f"Is like Fedora: {'YES' if is_like_fedora(os_data) else 'NO'}")
    print(f"Is like Debian: {'YES' if is_like_debian(os_data) else 'NO'}")

    # Additional information
    print()
    print("Detailed OS Information:")
    print("=====")
    print(f"ID: {os_data.get('ID', '')}")
    print(f"VERSION_ID: {os_data.get('VERSION_ID', '')}")
    print(f"PRETTY_NAME: {os_data.get('PRETTY_NAME', '')}")
    if os_data.get('ID_LIKE'):
        print(f"ID_LIKE: {os_data.get('ID_LIKE')}")

    # Amazon Linux specific information
    if is_amazon_linux(os_data):
        print()
        print("Amazon Linux Version Details:")
        print("=====")
        version_id = os_data.get('VERSION_ID', '')
        if version_id == '2018.03':
            print("Amazon Linux AMI (version 1)")
        elif version_id == '2':
            print("Amazon Linux 2")
        elif version_id == '2023':
            print("Amazon Linux 2023")
        else:

```

```

        print(f"Unknown Amazon Linux version: {version_id}")

    # Check for Amazon Linux specific files
    if os.path.exists('/etc/image-id'):
        print("Amazon Linux image-id file present")

if __name__ == '__main__':
    main()

```

Perl

```

#!/usr/bin/env perl

use strict;
use warnings;

# Function to parse the os-release file and return a hash of key-value pairs
sub parse_os_release {
    my %os_release_data;

    # Try /etc/os-release first, then /usr/lib/os-release
    my @paths = ('/etc/os-release', '/usr/lib/os-release');

    for my $path (@paths) {
        if (-f $path) {
            if (open(my $fh, '<', $path)) {
                while (my $line = <$fh>) {
                    chomp $line;
                    next if $line =~ /\s*$/ || $line =~ /\s*#/;

                    if ($line =~ /^(([^\s]+)=(.*)$)/) {
                        my ($key, $value) = ($1, $2);
                        # Remove quotes if present
                        $value =~ s/^[\'"]|[\']$//g;
                        $os_release_data{$key} = $value;
                    }
                }
                close($fh);
                return %os_release_data;
            }
        }
    }
}

```

```
    die "Error: os-release file not found\n";
}

# Function to check if this is Amazon Linux
sub is_amazon_linux {
    my %os_data = @_;
    return ($os_data[ID] // '') eq 'amzn';
}

# Function to check if this is Fedora
sub is_fedora {
    my %os_data = @_;
    return ($os_data[ID] // '') eq 'fedora';
}

# Function to check if this is Ubuntu
sub is_ubuntu {
    my %os_data = @_;
    return ($os_data[ID] // '') eq 'ubuntu';
}

# Function to check if this is Debian
sub is_debian {
    my %os_data = @_;
    return ($os_data[ID] // '') eq 'debian';
}

# Function to check if this is like Fedora (includes Amazon Linux, CentOS, RHEL,
etc.)
sub is_like_fedora {
    my %os_data = @_;
    return 1 if ($os_data[ID] // '') eq 'fedora';
    my $id_like = $os_data[ID_LIKE] // '';
    return $id_like =~ /fedora/;
}

# Function to check if this is like Debian (includes Ubuntu and derivatives)
sub is_like_debian {
    my %os_data = @_;
    return 1 if ($os_data[ID] // '') eq 'debian';
    my $id_like = $os_data[ID_LIKE] // '';
    return $id_like =~ /debian/;
}
```

```

# Main execution
my %os_data = parse_os_release();

# Display results
print "Operating System Detection Results:\n";
print "=====\n";
print "Is Amazon Linux: " . (is_amazon_linux(%os_data) ? "YES" : "NO") . "\n";
print "Is Fedora: " . (is_fedora(%os_data) ? "YES" : "NO") . "\n";
print "Is Ubuntu: " . (is_ubuntu(%os_data) ? "YES" : "NO") . "\n";
print "Is Debian: " . (is_debian(%os_data) ? "YES" : "NO") . "\n";
print "Is like Fedora: " . (is_like_fedora(%os_data) ? "YES" : "NO") . "\n";
print "Is like Debian: " . (is_like_debian(%os_data) ? "YES" : "NO") . "\n";
print "\n";

# Additional information
print "Detailed OS Information:\n";
print "=====\n";
print "ID: " . ($os_data{ID} // '') . "\n";
print "VERSION_ID: " . ($os_data{VERSION_ID} // '') . "\n";
print "PRETTY_NAME: " . ($os_data{PRETTY_NAME} // '') . "\n";
print "ID_LIKE: " . ($os_data{ID_LIKE} // '') . "\n" if $os_data{ID_LIKE};

# Amazon Linux specific information
if (is_amazon_linux(%os_data)) {
    print "\n";
    print "Amazon Linux Version Details:\n";
    print "=====\n";
    my $version_id = $os_data{VERSION_ID} // '';

    if ($version_id eq '2018.03') {
        print "Amazon Linux AMI (version 1)\n";
    } elsif ($version_id eq '2') {
        print "Amazon Linux 2\n";
    } elsif ($version_id eq '2023') {
        print "Amazon Linux 2023\n";
    } else {
        print "Unknown Amazon Linux version: $version_id\n";
    }

    # Check for Amazon Linux specific files
    if (-f '/etc/image-id') {
        print "Amazon Linux image-id file present\n";
    }
}

```

```
}
```

Quando executado em sistemas diferentes, o script produzirá a seguinte saída:

AL2023

```
Operating System Detection Results:
=====
Is Amazon Linux: YES
Is Fedora: NO
Is Ubuntu: NO
Is Debian: NO
Is like Fedora: YES
Is like Debian: NO

Detailed OS Information:
=====
ID: amzn
VERSION_ID: 2023
PRETTY_NAME: Amazon Linux 2023.8.20250721
ID_LIKE: fedora

Amazon Linux Version Details:
=====
Amazon Linux 2023
Amazon Linux image-id file present
```

AL2

```
Operating System Detection Results:
=====
Is Amazon Linux: YES
Is Fedora: NO
Is Ubuntu: NO
Is Debian: NO
Is like Fedora: YES
Is like Debian: NO

Detailed OS Information:
=====
ID: amzn
VERSION_ID: 2
```

```
PRETTY_NAME: Amazon Linux 2
ID_LIKE: centos rhel fedora

Amazon Linux Version Details:
=====
Amazon Linux 2
Amazon Linux image-id file present
```

Amazon Linux AMI

```
Operating System Detection Results:
=====
Is Amazon Linux: YES
Is Fedora: NO
Is Ubuntu: NO
Is Debian: NO
Is like Fedora: YES
Is like Debian: NO

Detailed OS Information:
=====
ID: amzn
VERSION_ID: 2018.03
PRETTY_NAME: Amazon Linux AMI 2018.03
ID_LIKE: rhel fedora

Amazon Linux Version Details:
=====
Amazon Linux AMI (version 1)
Amazon Linux image-id file present
```

Ubuntu

```
Operating System Detection Results:
=====
Is Amazon Linux: NO
Is Fedora: NO
Is Ubuntu: YES
Is Debian: NO
Is like Fedora: NO
Is like Debian: YES

Detailed OS Information:
```



```
=====
ID: ubuntu
VERSION_ID: 24.04
PRETTY_NAME: Ubuntu 24.04.2 LTS
ID_LIKE: debian
```

Debian

```
Operating System Detection Results:
=====
Is Amazon Linux: NO
Is Fedora: NO
Is Ubuntu: NO
Is Debian: YES
Is like Fedora: NO
Is like Debian: YES

Detailed OS Information:
=====
ID: debian
VERSION_ID: 12
PRETTY_NAME: Debian GNU/Linux 12 (bookworm)
```

Fedora

```
Operating System Detection Results:
=====
Is Amazon Linux: NO
Is Fedora: YES
Is Ubuntu: NO
Is Debian: NO
Is like Fedora: YES
Is like Debian: NO

Detailed OS Information:
=====
ID: fedora
VERSION_ID: 42
PRETTY_NAME: Fedora Linux 42 (Container Image)
```

Layout do sistema de arquivos

Esta seção aborda o layout do sistema de arquivos de um sistema do AL2023, incluindo detalhes que podem ser específicos de instâncias ou de contêineres baseados no AL2023. Para obter mais informações, consulte a página `file-hierarchy(7)` man.

Tópicos

- [/ \(o diretório-raiz\)](#)
- [/boot \(kernel, initramfs etc.\)](#)
- [/etc \(configuração do sistema\)](#)
- [/home \(diretórios iniciais de usuários\)](#)
- [/root \(diretório inicial do usuário root\)](#)
- [/srv \(carga útil do servidor\)](#)
- [/tmp \(pequenos arquivos temporários\)](#)
- [/run \(dados de runtime\)](#)
- [/usr \(recursos do sistema\)](#)
- [/var \(dados do sistema de variáveis persistentes\)](#)

/ (o diretório-raiz)

Por padrão, as imagens do AL2023 são configuradas com um / gravável, permitindo que usuários com privilégios criem arquivos e diretórios.

É possível configurar os serviços de `systemd` para usar um caminho ou uma imagem diferente para aparecer como / para esse serviço, bem como colocar restrições de acesso em qualquer caminho.

Note

É uma prática recomendada que os serviços de `systemd` sejam configurados para restringir a que o serviço tem acesso. Isso pode incluir o uso da diretiva `ReadOnlyPaths=/` que torna / somente leitura para esse serviço.

Para obter mais informações sobre como usar `systemd` para restringir o acesso que um serviço tem ao sistema, consulte a página `systemd.exec(5)` man.

/boot (kernel, initramfs etc.)

Por padrão, as imagens inicializáveis do AL2023 são configuradas com `/boot` no sistema de arquivos root. O caminho `/boot` só é relevante para imagens inicializáveis, portanto, não é usado em imagens de contêiner do AL2023.

Esse diretório abriga os arquivos necessários para a inicialização do AL2023, como o kernel do Linux e initramfs. O conteúdo desse diretório só deve ser manipulado usando as ferramentas fornecidas com o sistema operacional.

/boot/efi (partição do sistema EFI)

Por padrão, as imagens inicializáveis do AL2023 são configuradas com a partição EFI System montada em `/boot/efi`. Esse sistema de arquivos é gerenciado pelo sistema operacional e contém código e configuração essenciais para inicializar o sistema.

Esse caminho não é relevante para imagens de contêiner.

/etc (configuração do sistema)

O diretório `/etc` no AL2023 contém a configuração específica do sistema. Por padrão, as imagens do AL2023 vêm com `/etc` no sistema de arquivos root e podem ser gravadas por usuários com privilégios.

Note

É comum que as aplicações (inclusive `systemd`) mantenham a configuração padrão em [/usr \(recursos do sistema\)](#), podendo ser substituída ao inserir a configuração em [/etc \(configuração do sistema\)](#).

Para essas aplicações, alterar os arquivos em [/usr \(recursos do sistema\)](#) em vez de substituir a configuração padrão em `/etc` provavelmente resultará na substituição das alterações quando o pacote for atualizado.

/home (diretórios iniciais de usuários)

Usuários normais têm seus diretórios iniciais em `/home`, mas o software deve sempre procurar a variável de ambiente `$HOME` de cada usuário, em vez de confiar em um padrão, como `/home/$USER`.

Por padrão, as imagens do AL2023 têm `/home` no sistema de arquivos `root`, mas o software não deve depender disso. É perfeitamente válido que o sistema operacional seja configurado para que `/home` seja um sistema de arquivos separado, que é montado posteriormente durante a inicialização ou somente após a autenticação do usuário no sistema.

O diretório inicial do usuário `root` não está em `/home`, mas sim em [/root \(diretório inicial do usuário root\)](#) para que esteja disponível caso o sistema de arquivos `/home` não possa ser montado.

Note

É a prática recomendada para serviços de `systemd` que não precisam de acesso de gravação em `/home` para serem configurados com a diretiva `ProtectHome=read-only`. Com essa opção, `/home`, `/root` e `/run/user` se tornam somente leitura para o serviço. Também é uma prática recomendada para serviços que não precisam de nenhum acesso a `/home` para serem configurados com a diretiva `ProtectHome=tmpfs`, que executará o serviço em um ambiente de sandbox onde `/home`, `/root` e `/run/user` são sistemas de arquivos `tmpfs` vazios e somente leitura.

Para obter mais informações sobre como usar `systemd` para restringir o acesso que um serviço tem ao sistema, consulte a página `systemd.exec(5)` man.

/root (diretório inicial do usuário root)

O diretório inicial do usuário `root` é o diretório `/root`, propositadamente separado de [/home \(diretórios iniciais de usuários\)](#) para que esteja presente caso [/home \(diretórios iniciais de usuários\)](#) esteja em um sistema de arquivos indisponível.

A prática recomendada para configurar serviços de `systemd` é a mesma para `/root` e para [/home \(diretórios iniciais de usuários\)](#).

/srv (carga útil do servidor)

O diretório `/srv` é gerenciado pelo administrador do sistema, e o Amazon Linux 2023 não impõe restrições sobre como esse diretório é organizado.

É possível configurar o diretório `/srv` para estar em um sistema de arquivos separado, para que ele só fique disponível posteriormente na inicialização.

/tmp (pequenos arquivos temporários)

Note

O Amazon Linux 2023 é diferente do Amazon Linux 2, pois, por padrão, /tmp agora é tmpfs em vez de um caminho no sistema de arquivos root.

Note

Quando executado em um contêiner, normalmente é a configuração de runtime do contêiner que determina se /tmp é tmpfs, ou um caminho em disco, e se há ou não um processo de limpeza em execução.

O diretório /tmp é para arquivos temporários pequenos e limitados por tamanho. Por padrão, o AL2023 o configura para ser um sistema de arquivos tmpfs com um limite de tamanho de 50% da RAM e um máximo de um milhão de inodes.

As aplicações devem preferir o caminho na variável de ambiente \$TMPDIR em vez de /tmp. Depois, os usuários podem definir a variável de ambiente \$TMPDIR para substituir o caminho que uma aplicação deve usar para /tmp.

Para arquivos temporários maiores, [/var/tmp](#) deve ser usado em vez disso.

Warning

Como /tmp é compartilhado, é importante usar métodos seguros para a criação de arquivos temporários. Para obter detalhes, consulte a documentação de systemd upstream em [Using /tmp and /var/tmp Safely](#).

Note

É uma prática recomendada que os serviços de systemd sejam configurados com a diretiva PrivateTmp= definida como yes ou disconnected, o que executa o serviço em um ambiente de sandbox onde /tmp e [/var/tmp](#) não são compartilhados com o host ou outros serviços.

Para obter mais informações, incluindo como configurar dois serviços para compartilhar os mesmos diretórios temporários privados, consulte a página `systemd.exec(5) man`.

Normalmente, o conteúdo de `/tmp` é limpo no momento da inicialização e os arquivos não utilizados são limpos regularmente. Por padrão, o processo de limpeza é executado logo após a inicialização e, depois, diariamente. Para obter informações sobre como configurar a limpeza de arquivos temporários, consulte as páginas de manual `tmpfiles.d(5)` e `systemd-tmpfiles(8) man`.

Os caminhos `/tmp` e `/var/tmp` estão intimamente relacionados e existem para propósitos diferentes.

/run (dados de runtime)

O diretório `/run` é usado por pacotes do sistema para armazenar pequenas quantidades de dados de runtime (como arquivos de soquete). É um sistema de arquivos `tmpfs` e só pode ser gravado por programas com privilégios.

O diretório `/run/log` pode ser usado pelos componentes do sistema para armazenar logs antes de serem gravados em `/var/log` ou antes que o sistema de arquivos `/var/log` esteja disponível.

O caminho `/run/user/` contém diretórios de runtime para cada usuário. Por padrão, tratam-se de sistemas de arquivos `tmpfs` individuais montados por `systemd` quando o usuário faz login e apagados quando o usuário se desconecta. De acordo com [XDG Base Directory Specification](#), não se deve fazer referência direta a esses caminhos, mas sim por meio da variável de ambiente `$XDG_RUNTIME_DIR`.

/usr (recursos do sistema)

A hierarquia `/usr` se destina a recursos do sistema operacional fornecidos pelo fornecedor. Exceto pela hierarquia `/usr/local`, somente o gerenciador de pacotes do sistema operacional deve poder fazer modificações em `/usr`.

As aplicações de software devem presumir que `/usr` pode ser somente leitura. A hierarquia `/usr` não deve ser usada para dados voláteis. Exceto `/usr/local`, a hierarquia `/usr` não deve ser usada para nenhum dado adicionado ou alterado fora da instalação/remoção de pacotes, conforme feito pelo gerenciador de pacotes do sistema operacional. O gerenciador de pacotes do sistema operacional pode presumir que toda a hierarquia `/usr` (exceto `/usr/local`) é o mesmo ponto de montagem.

Software instalado fora do gerenciador de pacotes do sistema operacional não deve armazenar dados em `/usr` pois isso pode impedir invocações futuras do gerenciador de pacotes do sistema operacional. A hierarquia [`/usr/local`](#) é a exceção e é reservada para software fora do gerenciador de pacotes do sistema operacional.

`/usr/bin` (executáveis)

Arquivos executáveis que devem aparecer na pesquisa padrão de `$PATH` e que são úteis para invocar por meio de um shell. Daemons e executáveis que não são úteis para invocar por meio de um shell residem em `/usr/lib` ou `/usr/libexec`.

`/usr/include` (cabeçalhos C/C++)

O diretório `/usr/include` contém arquivos de cabeçalho C e C++, geralmente contidos em pacotes com o sufixo `-devel`.

`/usr/lib` e `/usr/lib64` (bibliotecas compartilhadas)

No Amazon Linux 2023, o caminho `/usr/lib64` é usado para bibliotecas compartilhadas de 64 bits e dados de pacotes que dependem da arquitetura. Como o AL2023 não é distribuído com nenhum suporte a espaços de usuário de 32 bits, há apenas bibliotecas compartilhadas de 64 bits disponíveis.

O caminho `/usr/lib` é para dados estáticos de pacotes do sistema operacional que são compatíveis com todas as arquiteturas. Isso pode incluir executáveis que geralmente não são invocados por meio de um shell, que também podem ser encontrados em `/usr/libexec`. As bibliotecas compartilhadas são encontradas em `/usr/lib64` em vez de `/usr/lib`.

`/usr/local` (software instalado pelo administrador do sistema)

No Amazon Linux 2023, o caminho `/usr/local` está disponível para o administrador do sistema instalar software, software que não pertence ao sistema operacional e que não será afetado pelo sistema operacional. A hierarquia `/usr/local` padrão reflete a hierarquia `/`.

`/usr/share` (recursos compartilhados)

Recursos compartilhados, como documentação, fontes e dados de fuso horário, residem em `/usr/share`. É comum que várias especificações ditem exatamente onde e em qual formato os dados são armazenados nesse diretório.

/usr/share/doc (documentação)

A documentação que acompanha os pacotes será armazenada em `/usr/share/doc`.

/var (dados do sistema de variáveis persistentes)

/var/cache (cache)

Ao contrário de [/var/lib](#), apagar dados em `/var/cache` não resultará em perda de dados, pois as aplicações precisam ser capazes de recriar seus dados de `/var/cache` de outras fontes.

/var/lib (dados do sistema persistentes)

O diretório `/var/lib` é usado para dados do sistema persistentes. Vários componentes do sistema colocarão aqui dados que são privados de cada componente. Ao contrário de [/var/cache](#), apagar dados em `/var/lib` resultará em perda de dados.

Por exemplo, o servidor de banco de dados PostgreSQL armazenará os dados do banco de dados em `/var/lib/pgsql` por padrão. O layout e os formatos de arquivo desses dados são privados do PostgreSQL, e são dados persistentes, pois, se forem apagados, o usuário sofrerá perda de dados.

/var/log (logs persistentes)

Esse diretório é usado para armazenar logs persistentes. É recomendável que o software use as chamadas de API `syslog(3)` ou `sd_journal_print(3)` em vez de armazenar arquivos de log diretamente em `/var/log`.

Note

No AL2023, o [Diário de systemd substitui rsyslog](#), o que é uma diferença notável em relação à configuração padrão do Amazon Linux 2.

Para obter mais informações sobre como ler logs usando `journalctl`, consulte a página do manual [journalctl](#).

Muitas aplicações usam mecanismos próprios para gravar e, às vezes, alternar os arquivos de log encontrados em `/var/log`. Consulte a documentação dessas aplicações sobre como configurar os respectivos arquivos de log.

/var/spool (filas de mensagens e impressoras)

Esse diretório é usado para dados persistentes, como filas de mensagens ou impressoras.

/var/tmp (arquivos temporários maiores)

Para arquivos temporários pequenos e limitados por tamanho, talvez seja melhor usar [/tmp](#).

Embora [/tmp](#) esteja configurado por padrão para ser um volume `tmpfs`, `/var/tmp` é configurado por padrão para ser um caminho no sistema de arquivos raiz e, portanto, é o local para arquivos temporários maiores e mais persistentes. Por padrão, um trabalho de limpeza é executado regularmente para remover arquivos não acessados recentemente.

Para obter informações sobre como configurar a limpeza de arquivos temporários, consulte as páginas de manual `tmpfiles.d(5)` e `systemd-tmpfiles(8)` man.

Assim como [/tmp](#), as aplicações devem preferir o caminho especificado na variável de ambiente `$TMPDIR` em vez de `/var/tmp`. Depois, os usuários podem definir a variável de ambiente `$TMPDIR` para substituir o caminho que uma aplicação deve usar para `/var/tmp`.

Warning

Como `/var/tmp` é compartilhado (assim como [/tmp](#)), é importante usar métodos seguros para a criação de arquivos temporários. Para obter detalhes, consulte a documentação de `systemd` upstream em [Using /tmp and /var/tmp Safely](#).

Note

É uma prática recomendada que os serviços de `systemd` sejam configurados com a diretiva `PrivateTmp=` definida como `yes` ou `disconnected`, o que executa o serviço em um ambiente de sandbox onde [/tmp](#) e [/var/tmp](#) não são compartilhados com o host ou outros serviços.

Para obter mais informações, incluindo como configurar dois serviços para compartilhar os mesmos diretórios temporários privados, consulte a página `systemd.exec(5)` man.

Os caminhos [/tmp](#) e [/var/tmp](#) estão intimamente relacionados e existem para propósitos diferentes.

Atualizando AL2023

É importante manter-se atualizado com os AL2023 lançamentos para que você possa se beneficiar das atualizações de segurança e dos novos recursos. Com AL2023, você pode garantir a consistência entre as versões e atualizações do pacote em seu ambiente por meio de [Atualizações determinísticas por meio de repositórios versionados em AL2023](#).

Warning

A execução de `dnf --releasever=latest update` não é uma prática recomendada e provavelmente resultará no primeiro teste de uma atualização do sistema operacional em produção.

Em vez de usar `latest`, use uma versão AL2023 de lançamento específica. Isso garante que você esteja implantando as mesmas alterações em todas as instâncias de produção que testou anteriormente. Por exemplo, sempre `dnf --releasever=2023.10.20260216 update` atualizará para a versão 2023.10.20260216.

Para obter mais informações, consulte a AL2023 seção [Atualização](#) no [Guia AL2023 do usuário](#).

Tópicos

- [Práticas recomendadas para implantar atualizações de forma segura](#)
- [Receber notificações sobre novas atualizações](#)
- [Atualizações determinísticas por meio de repositórios versionados em AL2023](#)
- [Gerencie atualizações de pacotes e sistemas operacionais no AL2023](#)
- [Aplicação de patches do kernel em tempo real no AL2023](#)
- [Atualizando o kernel Linux em AL2023](#)

Práticas recomendadas para implantar atualizações de forma segura

O Amazon Linux 2023 (AL2023) tem vários recursos projetados para ajudar na implantação segura de atualizações no sistema operacional e na capacidade de saber o que mudou entre as

atualizações e, se necessário, reverter facilmente para a versão mais antiga. Esta seção explora as lições aprendidas em mais AWS de uma década de uso interno e externo do Amazon Linux.

 Warning

A execução de `dnf --releasever=latest update` não é uma prática recomendada e provavelmente resultará no primeiro teste de uma atualização do sistema operacional em produção.

Em vez de usar `latest`, use uma versão AL2023 de lançamento específica. Isso garante que você esteja implantando as mesmas alterações em todas as instâncias de produção que testou anteriormente. Por exemplo, sempre `dnf --releasever=2023.10.20260216 update` e atualizará para a versão 2023.10.20260216.

Para obter mais informações, consulte a AL2023 seção [Atualização](#) no [Guia AL2023 do usuário](#).

Sem planejar a segurança da implantação das atualizações do sistema operacional, o impacto de uma interação negativa inesperada entre você application/service e uma atualização do sistema operacional pode ser significativamente maior, incluindo uma interrupção total. Como acontece com qualquer problema de software, quanto mais cedo o problema for detectado, menor será o impacto que ele poderá ter sobre os usuários finais.

É importante não cair na armadilha de acreditar em duas coisas que fundamentalmente não são verdadeiras:

1. O fornecedor do sistema operacional nunca cometerá um erro em uma atualização do sistema operacional.
2. O comportamento específico ou a interface com o sistema operacional em que você confia corresponde ao comportamento e às interfaces que o fornecedor do sistema operacional consideraria confiáveis.

Ou seja, tanto o fornecedor do sistema operacional quanto você concordariam que houve um problema com a atualização.

Não confie em boas intenções, implemente sistemas para garantir que a segurança da implantação inclua qualquer atualização do sistema operacional.

Não é recomendável testar novas atualizações do sistema operacional implantando-as em ambientes de produção. É uma prática recomendada considerar o sistema operacional como outra parte de sua implantação e pensar em aplicar os mesmos mecanismos de segurança de implantação que você considera adequados para qualquer outra alteração em um ambiente de produção.

É uma prática recomendada testar toda e qualquer atualização do sistema operacional antes de implantar nos sistemas de produção. Durante a implantação, são recomendadas implementações graduais combinadas com um bom monitoramento. As implementações em etapas podem garantir que, se ocorrer um problema, mesmo que não seja imediato, o impacto seja restrito a um subconjunto da frota, e a implantação da atualização possa ser interrompida enquanto se realiza uma investigação mais aprofundada e se tomam medidas de mitigação.

A mitigação de qualquer impacto negativo causado pela atualização do sistema operacional costuma ser a primeira prioridade, seguida pela resolução do problema, onde quer que ele esteja. Quando a introdução de uma atualização do sistema operacional está correlacionada a um impacto negativo, a capacidade de reverter para a versão anterior do sistema operacional, que se sabe ser boa, é uma ferramenta poderosa.

O Amazon Linux 2023 apresenta [Atualizações determinísticas por meio de repositórios com versionamento](#), um novo recurso poderoso para garantir que qualquer alteração na versão do sistema operacional (ou pacotes individuais) seja repetível. Assim, se ocorrer algum problema ao mudar de uma versão do sistema operacional para a seguinte, existem mecanismos simples de usar disponíveis para manter a versão do sistema operacional que funciona corretamente enquanto se resolve o problema.

Com isso AL2023, sempre que lançamos novas atualizações de pacotes, há uma nova versão para bloquear e um novo AMIs bloqueio para essa versão. As [notas AL2023 de lançamento](#) abordam as mudanças em cada versão e [Consultorias de segurança do Amazon Linux para AL2023](#) abordam os problemas de segurança abordados nas atualizações do pacote.

[Por exemplo, se você foi afetado pelo problema presente na versão 2023.6.20241028, você poderia imediatamente voltar a usar as imagens de contêiner da versão anterior, 2023.6.20241010. AMIs](#) Nesse caso, havia um bug em um pacote que foi corrigido no lançamento subsequente [2023.6.20241031](#), mas com [Atualizações determinísticas por meio de repositórios com versionamento](#) qualquer pessoa afetada poderia tomar imediatamente uma medida simples para mitigar o problema: bastava usar as imagens anteriores.

[Atualizações determinísticas por meio de repositórios com versionamento](#) também garante que qualquer implantação em andamento de uma atualização do sistema operacional, seja no local

ou por meio do lançamento de imagens novas AMIs ou de contêiner, não seja afetada pelas atualizações do sistema operacional lançadas posteriormente.

No nosso primeiro exemplo, a frota A é uma grande frota que está no meio da implantação da atualização de [2023.5.20241001](#) para [2023.6.20241010](#) quando [2023.6.20241028](#) é lançada. [Atualizações determinísticas por meio de repositórios com versionamento](#) significam que a implementação para a frota A continua sem qualquer alteração nas atualizações que está aplicando.

O objetivo das estratégias de implantação baseadas em ondas ou em fases, como implantar primeiro em 1% de uma frota, depois em 5%, 10%, 20% e 40%, até atingir 100%, é poder testar uma mudança de forma limitada antes de implementá-la mais amplamente. Esse tipo de estratégia de implantação geralmente é considerado a prática recomendada para implantar qualquer alteração em produção.

Com uma estratégia de implantação baseada em ondas e a atualização da frota A para [2023.6.20241010](#) em um estágio em que está sendo implantada em muitos hosts ao mesmo tempo, o fato de [2023.6.20241028](#) ter sido lançada não tem impacto na implantação em andamento graças ao uso de [Atualizações determinísticas por meio de repositórios com versionamento](#).

Se a frota B estivesse executando uma versão mais antiga, digamos [2023.5.20240708](#), e tivesse começado a implantar a atualização para [2023.6.20241028](#), e a frota B fosse afetada pelo problema nessa versão, isso seria percebido no início da implantação. Nesse momento, pode-se decidir se é necessário pausar qualquer implementação até que uma correção para esse problema esteja disponível ou se, entretanto, é necessário iniciar uma implantação da mesma versão que a frota A estava executando, [2023.6.20241010](#), para que a frota B receba todas as atualizações entre [2023.5.20240708](#) e [2023.6.20241010](#).

É importante observar que não instalar as atualizações do sistema operacional prontamente pode causar problemas. Novas atualizações provavelmente contêm correções de bugs e segurança que podem ser relevantes para o seu ambiente. Para obter mais informações, consulte [Segurança e compatibilidade no Amazon Linux 2023](#) e [Gerencie atualizações de pacotes e sistemas operacionais no AL2023](#).

É importante configurar seus sistemas de implantação para instalar facilmente novas atualizações do sistema operacional, testá-las antes da implantação em produção e usar mecanismos, como implantações baseadas em ondas, para minimizar qualquer impacto negativo. Para poder mitigar qualquer impacto negativo de uma atualização do sistema operacional, é importante saber como fazer com que seus sistemas de implantação apontem para uma versão anterior em boas condições

do sistema operacional e, depois que o problema for resolvido, não ficar preso a essa versão antiga, mas sim mudar para uma nova versão em boas condições.

Preparar-se para atualizações secundárias

A preparação para atualizações menores do sistema operacional, como uma nova versão pontual do AL2023, deve ser limitada a zero esforço. Não deixe de ler as [notas AL2023 de lançamento](#) para ver as próximas alterações.

O fim do [período de suporte de um pacote](#) pode envolver a mudança para uma versão mais recente do runtime da linguagem (como no caso de [PHPem AL2 023](#)). É uma prática recomendada se preparar para isso com antecedência, migrando para novas versões de runtime da linguagem com bastante antecedência do término do período de suporte.

Para pacotes como [pcr e versão 1](#), também há a oportunidade de planejar com antecedência e migrar qualquer código para seu substituto, que neste caso é pcr e versão 2. É recomendável fazer isso o quanto antes, para ter tempo suficiente para lidar com eventuais contratempos.

Onde não há substituição direta, como no caso de [Berkeley DB \(libdb\)](#), talvez seja necessário fazer uma escolha com base no seu caso de uso.

Preparar-se para atualizações principais

A atualização para uma nova versão principal de um sistema operacional é quase universalmente vista como algo que requer planejamento, trabalho para se adaptar a funcionalidades alteradas ou descontinuadas e também testes antes da implantação. Não é incomum poder se preparar para a próxima versão principal do Amazon Linux 2023 de forma mais incremental, como resolver qualquer uso de funcionalidades descontinuadas ou removidas antes de prosseguir com a migração para a próxima versão principal.

Por exemplo, ao passar de AL2 para AL2023, a leitura da [Funcionalidade obsoleta e removida em AL2 AL2023](#) seção pode resultar em várias etapas pequenas e seguras que podem ocorrer enquanto ainda são usadas AL2 para se preparar AL2023. Por exemplo, qualquer uso de [O Python 2.7 foi substituído pelo Python 3](#) (fora do uso do sistema operacional, como no gerenciador de pacotes yum) pode ser migrado para o Python 3 em preparação para o uso de [Python no AL2023](#). Se estiver usando [PHP](#), ambos AL2 (por meio do PHP 8.2 [AL2 Extra](#)) e AL2023 lançam o PHP 8.2 e, portanto, a migração da versão do PHP e a migração do sistema operacional não precisam ocorrer simultaneamente.

Ao usar AL2023, também é possível se preparar para a próxima versão principal do Amazon Linux 2023 hoje, enquanto estiver usando AL2023. A [Descontinuações no AL2023](#) seção aborda recursos e pacotes que estão obsoletos AL2023 e devem ser removidos.

Por exemplo, migrar qualquer uso restante de [System V init \(sysvinit\)](#), como scripts `init`, para seus equivalentes de `systemd` preparará você para o futuro, além de permitir que use o conjunto completo de recursos de `systemd` para monitorar o serviço, como e se deve reiniciá-lo, quais outros serviços ele precisa e se alguma restrição de recurso ou permissão deve ser aplicada.

Para recursos como suporte a 32 bits, a descontinuação pode abranger várias versões principais do sistema operacional. Para 32 bits, o Amazon Linux 1 (AL1) está obsoleto [AMIs x86 \(i686\) de 32 bits](#), o Amazon Linux 2 está obsoleto e o Amazon Linux 2023 está [Pacotes x86 \(i686\) de 32 bits](#) obsoleto. [Suporte ao runtime x86 \(i686\) de 32 bits](#) A transição do [IMDSv1](#) também abrange várias versões principais do sistema operacional. Para esses tipos de mudanças, entende-se que alguns clientes precisam de mais tempo para se adaptar a elas, portanto, há uma grande margem de manobra antes que a funcionalidade não esteja mais disponível no Amazon Linux 2023.

A lista de funcionalidades descontinuadas é atualizada durante a vida útil do sistema operacional, e é recomendável acompanhar as alterações feitas nela.

Receber notificações sobre novas atualizações

Você pode receber notificações sempre que uma nova AMI do AL2023 for lançada. As notificações são publicadas com o [Amazon SNS](#) usando o tópico a seguir.

```
arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates
```

As mensagens são publicadas aqui quando uma nova AMI AL2023 é publicada. A versão da AMI será incluída na mensagem.

Essas mensagens podem ser recebidas usando vários métodos diferentes. Recomendamos que você use o método a seguir.

1. Abra o console do [Amazon SNS](#).
2. Na barra de navegação, altere a Região da AWS para Leste dos EUA (N. da Virgínia), se necessário. É necessário selecionar esta região onde a notificação do SNS que está assinando foi criada nesta região.
3. No painel de navegação, escolha Assinaturas, Criar assinatura.

4. Na caixa de diálogo Create subscription, faça o seguinte:
 - a. Em ARN do tópico, copie e cole o seguinte nome do recurso da Amazon (ARN):
arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates.
 - b. Em Protocol (Protocolo), escolha Email.
 - c. Em Endpoint, insira um endereço de e-mail que possa ser usado para receber notificações.
 - d. Selecione Create subscription.
5. Você receberá um e-mail de confirmação com o assunto "Notificação da AWS – confirmação de assinatura". Abra o e-mail e escolha Confirm subscription para concluir a assinatura.

Atualizações determinísticas por meio de repositórios versionados em AL2023

Note

Por padrão, sua AL2023 instância não recebe automaticamente atualizações de segurança adicionais críticas e importantes na inicialização. Sua instância contém inicialmente as atualizações que estavam disponíveis na versão AL2023 e na AMI escolhida.

Controle as atualizações recebidas de versões principais e secundárias

Com AL2023, você pode garantir a consistência entre as versões e atualizações do pacote em seu ambiente. Você também pode garantir a consistência de várias instâncias da mesma Imagem de máquina da Amazon (AMI). Com as atualizações determinísticas através do recurso de repositórios de versão, que é ativado por padrão, você pode aplicar atualizações com base em um cronograma que atenda às suas necessidades específicas.

Sempre que lançamos novas atualizações de pacotes, há uma nova versão para bloquear e um novo AMIs bloqueio para essa versão.

AL2023 bloqueia uma versão específica do seu repositório. Isso é compatível com versões principais ou secundárias. A AL2023 AMI, exposta por meio de nossos parâmetros SSM, é sempre a versão mais recente. Ele tem a maioria dos up-to-date pacotes e atualizações, incluindo atualizações de segurança críticas e importantes.

Se você iniciar uma instância em uma AMI existente, as atualizações não são aplicadas automaticamente. Todos os pacotes adicionais instalados como parte do seu provisionamento são mapeados para a versão do repositório da AMI existente.

Com esse recurso, você é responsável por garantir a consistência entre as versões e atualizações do pacote em seu ambiente. Esse é particularmente o caso se você estiver executando várias instâncias da mesma AMI. É possível aplicar atualizações baseadas em um cronograma que atenda às suas necessidades. Você também pode aplicar um conjunto específico de atualizações no lançamento, pois elas também podem ser bloqueadas em uma versão específica do repositório.

Diferenças entre atualizações de versão menor e principal

As versões principais do AL2023 incluem atualizações em grande escala e podem adicionar, excluir ou atualizar pacotes. Para garantir a compatibilidade, atualize sua instância para uma nova versão principal somente depois de testar seu aplicativo nessa versão.

As versões secundárias do AL2023 incluem atualizações de recursos e segurança, mas não incluem alterações de pacote. Isso garante que os recursos do Linux e a API da biblioteca do sistema permaneçam disponíveis em novas versões. Não é necessário testar o aplicativo antes da atualização.

Como saber quando há atualizações disponíveis

Para aplicar uma atualização, é necessário saber se há uma disponível e saber como implantá-la.

Para compilações derivadas AMIs quando novas AL2023 AMIs são lançadas, o [EC2 Image Builder](#) pode criar, corrigir e AMIs testar automaticamente. Para acionar seus próprios pipelines de construção de AMI ou usar a base AMIs, você pode [Receber notificações sobre novas atualizações](#).

Para a aplicação de patches no local, você pode usar ferramentas, como o [Gerenciador de Patches do AWS Systems Manager](#), para orquestrar a aplicação de atualizações em uma frota.

Para outros públicos, AMIs com base em AL2023, os fornecedores desses AMIs podem ter seu próprio cronograma de lançamentos e métodos de notificação. Ao usar imagens derivadas AMIs ou de contêiner, verifique a documentação do editor para saber quando as atualizações serão lançadas.

As mudanças em cada versão estão documentadas nas [notas AL2023 de lançamento](#). As atualizações de segurança são publicadas no [Amazon Linux Security Center \(ALAS\)](#).

Controle as atualizações de pacotes disponíveis nos AL2023 repositórios

Quando publicamos uma nova versão dos AL2023 repositórios, todas as versões anteriores ainda estão disponíveis. Por padrão, o plug-in para gerenciar versões do repositório se fixa na mesma versão usada para criar a AMI. Se pretende controlar as atualizações do pacote, siga estas etapas.

1. Descubra as versões disponíveis do repositório ao executar o comando a seguir.

```
$ sudo dnf check-release-update
```

2. O comando a seguir pode ser executado para verificar.

```
$ sudo dnf upgrade --releasever=version
```

Esse comando inicia uma atualização usando dnf a partir da versão atual de lançamento Amazon Linux para a versão de lançamento que é especificada na linha de comando. Uma lista das atualizações do pacote é apresentada por dnf. Antes que a atualização seja processada, você deve confirmar a atualização. Depois que a atualização for concluída, a nova versão de lançamento se tornará a versão de lançamento padrão que dnf usa para todas as atividades futuras.

Para obter mais informações, consulte [Gerencie atualizações de pacotes e sistemas operacionais no AL2023](#).

Atualizações determinísticas por meio da substituição de instâncias

O recurso de [Atualizações determinísticas por meio de repositórios versionados em AL2023](#) do Amazon Linux 2023 torna a substituição de instâncias uma maneira fácil de implementar versões atualizadas do AL2023 de forma determinística e segura. As atualizações determinísticas significam que, à medida que uma nova versão é lançada de forma progressiva, se algum problema for encontrado, será simples reverter para a AMI anterior e determinar a causa do problema.

O uso da substituição de instâncias em vez da aplicação de patches no local garante atualizações mais determinísticas e previsíveis, pois o lançamento de novas capacidades pode ser um caminho de código bem testado com estados A e B claros. Cada um dos estados antes e depois pode ser bem testado em um sistema de CI/CD antes do início da implantação.

Ao aplicar patches no local, há muitos estados intermediários entre antes e depois da aplicação das atualizações, o que é mais difícil de testar para todas as combinações de estados.

Uma estratégia de atualização do sistema operacional que usa a substituição de instâncias com atualizações determinísticas é ideal para modelos de implantação azul/verde ou de implantação baseada em ondas e fases.

Usar atualizações determinísticas por meio de repositórios com versionamento

Tópicos

- [Usando um sistema determinístico atualizado](#)
- [Atualização seletiva de um sistema determinístico atualizado](#)
- [Usando a substituição persistente com atualização determinística](#)

Usando um sistema determinístico atualizado

Note

O comportamento padrão do gerenciador de pacotes mudou de AL2.

As atualizações determinísticas são uma forma poderosa de garantir que todas as mudanças nos ambientes de produção possam ser totalmente testadas antes de uma ampla implantação. Cada nova AL2023 AMI está bloqueada para uma versão específica do AL2023. Isso fornece um comportamento determinístico de quais versões dos pacotes de sistema operacional são instaladas ao inicializar a AMI específica. As atualizações no local podem ser para uma versão de lançamento específica, garantindo um comportamento determinístico em toda a frota. Ao migrar para versões de atualização novas AMIs ou in-loco, você pode testar cada uma em seu CI/CD pipeline, detectando possíveis problemas antes de implantá-la em ambientes de produção.

Você pode usar ferramentas, como o [Gerenciador de Patches do AWS Systems Manager](#), para orquestrar a aplicação de atualizações em uma frota. Para compilações derivadas AMIs quando novas AL2023 AMIs são lançadas, o [EC2 Image Builder](#) pode criar, corrigir e AMIs testar automaticamente, ou você [Receber notificações sobre novas atualizações](#) pode saber quando uma nova AMIs base está disponível ou acionar seus próprios pipelines de criação de AMI.

Para obter informações sobre como restringir as atualizações às de uma recomendação específica, consulte [Aplicar atualizações de segurança no local](#).

Para aplicar patches no local, você pode usar o gerenciador de pacotes `dnf`. Quando você executa o comando `dnf upgrade`, o sistema verifica se há atualizações no repositório que a variável `releasever` especifica. Uma versão válida `releasever` é uma *latest* ou uma versão com carimbo de data, como. *2023.10.20260216*

É possível alterar o valor de `releasever` usando um dos métodos a seguir. Esses métodos estão listados em prioridade decrescente do sistema. Isso significa que o método 1 substitui os métodos 2 e 3, e o método 2 substitui o método 3.

1. O valor no sinalizador da linha de comando, `--releasever=latest`, se for usado.
2. O valor especificado no arquivo da variável de substituição, `/etc/dnf/vars/releasever`, se estiver definido.
3. A versão atualmente instalada do pacote `system-release`.

No exemplo a seguir, a versão é *2023.0.20230210*:

```
$ rpm -q system-release
system-release-2023.0.20230210-0.amzn2023.noarch
```

Em um sistema recém-instalado, a variável de substituição não está presente. Nenhuma atualização está disponível porque o sistema está bloqueado para a versão instalada do `system-release`.

```
$ cat /etc/dnf/vars/releasever
cat: /etc/dnf/vars/releasever: No such file or directory
```

```
$ sudo dnf upgrade
Last metadata expiration check: 0:00:02 ago on Wed 15 Feb 2023 06:14:12 PM UTC.
Dependencies resolved.
Nothing to do.
Complete!
```

Você pode obter pacotes de uma versão específica usando o sinalizador `releasever` para fornecer a versão desejada.

```
$ rpm -q system-release
system-release-2023.0.20230222-0.amzn2023.noarch
```

```
$ sudo dnf upgrade --releasever=2023.0.20230329
```

```

Amazon Linux 2023 repository                26 MB/s | 12 MB    00:00
Dependencies resolved.
=====
Package                Arch    Version                                Repository    Size
=====
Installing:
kernel                  aarch64 6.1.21-1.45.amzn2023                  amazonlinux   26 M
Upgrading:
amazon-linux-repo-s3    noarch  2023.0.20230329-0.amzn2023            amazonlinux   18 k
ca-certificates         noarch  2023.2.60-1.0.amzn2023.0.1            amazonlinux   828 k
cloud-init              noarch  22.2.2-1.amzn2023.1.7                  amazonlinux   1.1 M

... [ list edited for clarity ]

system-release          noarch  2023.0.20230329-0.amzn2023            amazonlinux   29 k

... [ list edited for clarity ]

vim-data                noarch  2:9.0.1403-1.amzn2023.0.1             amazonlinux   25 k
vim-minimal             aarch64 2:9.0.1403-1.amzn2023.0.1             amazonlinux   753 k

Transaction Summary
=====
Install    1 Package
Upgrade   42 Packages

Total download size: 56 M

```

Como a opção `--releasever` substitui ambas `system-release` e `/etc/dnf/vars/releasever`, o resultado dessa atualização é o seguinte:

1. A atualização substitui todos os pacotes instalados que foram alterados entre a versão anterior e a nova.
2. A atualização bloqueia o sistema no repositório da nova versão do `system-release`.

Ao sempre especificar para qual `releasever` (ou seja, AL2023 versão) atualizar, você tem um conjunto determinístico de mudanças em uma frota. Você lançou a versão **A**, atualizou para **B** e depois atualizou para **C**.

Atualização seletiva de um sistema determinístico atualizado

Note

Recomendamos instalar todas as atualizações em um novo lançamento, em vez de selecionar atualizações específicas. Aplicar apenas parte de uma atualização ao sistema operacional deve ser uma exceção à prática padrão de realizar a atualização inteira.

Talvez você queira instalar pacotes selecionados de uma versão recente, deixando o sistema bloqueado para a versão original.

É possível usar `dnf check-update` para identificar os pacotes que você deseja atualizar.

```
$ sudo dnf check-update --releasever=latest --security
Amazon Linux 2023 repository                13 MB/s | 10 MB      00:00
Last metadata expiration check: 0:00:02 ago on Wed 15 Feb 2023 02:52:21 AM UTC.

bind-libs.aarch64                32:9.16.27-1.amzn2023.0.1      amazonlinux
bind-license.noarch              32:9.16.27-1.amzn2023.0.1      amazonlinux
bind-utils.aarch64              32:9.16.27-1.amzn2023.0.1      amazonlinux
cryptsetup.aarch64              2.4.3-2.amzn2023.0.1           amazonlinux
cryptsetup-libs.aarch64         2.4.3-2.amzn2023.0.1           amazonlinux
curl-minimal.aarch64            7.85.0-1.amzn2023.0.1          amazonlinux
glibc.aarch64                   2.34-40.amzn2023.0.2           amazonlinux
glibc-all-langpacks.aarch64     2.34-40.amzn2023.0.2           amazonlinux
glibc-common.aarch64            2.34-40.amzn2023.0.2           amazonlinux
glibc-locale-source.aarch64     2.34-40.amzn2023.0.2           amazonlinux
gmp.aarch64                     1:6.2.1-2.amzn2023.0.1         amazonlinux
gnupg2-minimal.aarch64          2.3.7-1.amzn2023.0.2           amazonlinux
gzip.aarch64                    1.10-5.amzn2023.0.1            amazonlinux
kernel.aarch64                  6.1.12-17.42.amzn2023          amazonlinux
kernel-tools.aarch64            6.1.12-17.42.amzn2023          amazonlinux
libarchive.aarch64              3.5.3-2.amzn2023.0.1           amazonlinux
libcurl-minimal.aarch64         7.85.0-1.amzn2023.0.1          amazonlinux
libsepol.aarch64                3.4-3.amzn2023.0.2             amazonlinux
libsolv.aarch64                 0.7.22-1.amzn2023.0.1          amazonlinux
libxml2.aarch64                 2.9.14-1.amzn2023.0.1          amazonlinux
logrotate.aarch64               3.20.1-2.amzn2023.0.2          amazonlinux
lua-libs.aarch64                5.4.4-3.amzn2023.0.1           amazonlinux
lz4-libs.aarch64                1.9.4-1.amzn2023.0.1           amazonlinux
openssl.aarch64                 1:3.0.5-1.amzn2023.0.3         amazonlinux
```

openssl-libs.aarch64	1:3.0.5-1.amzn2023.0.3	amazonlinux
pcre2.aarch64	10.40-1.amzn2023.0.1	amazonlinux
pcre2-syntax.noarch	10.40-1.amzn2023.0.1	amazonlinux
rsync.aarch64	3.2.6-1.amzn2023.0.2	amazonlinux
vim-common.aarch64	2:9.0.475-1.amzn2023.0.1	amazonlinux
vim-data.noarch	2:9.0.475-1.amzn2023.0.1	amazonlinux
vim-enhanced.aarch64	2:9.0.475-1.amzn2023.0.1	amazonlinux
vim-filesystem.noarch	2:9.0.475-1.amzn2023.0.1	amazonlinux
vim-minimal.aarch64	2:9.0.475-1.amzn2023.0.1	amazonlinux
xz.aarch64	5.2.5-9.amzn2023.0.1	amazonlinux
xz-libs.aarch64	5.2.5-9.amzn2023.0.1	amazonlinux
zlib.aarch64	1.2.11-32.amzn2023.0.3	amazonlinux

Instale os pacotes que você deseja atualizar. Use `sudo dnf upgrade --releasever=latest` e os nomes dos pacotes para garantir que o pacote `system-release` permaneça inalterado.

```
$ sudo dnf upgrade --releasever=latest openssl openssl-libs
```

```
Last metadata expiration check: 0:01:28 ago on Wed 15 Feb 2023 02:52:21 AM UTC.
Dependencies resolved.
```

```
=====
Package           Arch      Version                               Repository      Size
=====
Upgrading:
openssl           aarch64   1:3.0.5-1.amzn2023.0.3               amazonlinux      1.1 M
openssl-libs      aarch64   1:3.0.5-1.amzn2023.0.3               amazonlinux      2.1 M
```

```
Transaction Summary
```

```
=====
Upgrade 2 Packages
```

```
Total download size: 3.2 M
```

Note

O uso de `sudo dnf upgrade --releasever=latest` atualiza todos os pacotes, inclusive `system-release`. Em seguida, a versão permanece bloqueada para o novo `system-release`, a menos que você defina a substituição persistente.

Usando a substituição persistente com atualização determinística

Note

Com atualizações determinísticas, você pode integrar as alterações do sistema operacional ao seu CI/CD pipeline. A desativação das atualizações determinísticas elimina a capacidade de testar antes da implantação.

Em vez de adicionar `--releasever=latest`, você pode usar a substituição persistente para desbloquear o sistema definindo o valor da variável como `latest`. Ao sempre usar `latest`, isso reverte o comportamento do AL2023 modelo de AL2 atualização, em que qualquer chamada para o gerenciador de pacotes sempre examinará a versão mais recente e não estará bloqueada em nenhuma versão específica do sistema operacional.

Warning

Ao desvincular o gerenciador de pacotes usando uma substituição persistente de atualizações determinísticas, você corre o risco de descobrir possíveis incompatibilidades entre sua aplicação e uma atualização do sistema operacional em produção.

Embora as incompatibilidades sejam raras, uma atualização do sistema operacional integra novas alterações de código em seu ambiente, e os testes de integração podem impedir a implantação de alterações de código que tenham um impacto negativo nos ambientes de produção.

```
$ echo latest | sudo tee /etc/dnf/vars/releasever
latest
```

```
$ sudo dnf upgrade
```

```
Last metadata expiration check: 0:03:36 ago on Wed 15 Feb 2023 02:52:21 AM UTC.
Dependencies resolved.
```

```
=====
Package                Arch    Version                               Repository    Size
=====
Installing:
kernel                  aarch64 6.1.73-45.135.amzn2023               amazonlinux   24 M
Upgrading:
acl                     aarch64 2.3.1-2.amzn2023.0.1                 amazonlinux   72 k
```


alternatives	aarch64	1.15-2.amzn2023.0.1	amazonlinux	36 k
amazon-ec2-net-utils	noarch	2.3.0-1.amzn2023.0.1	amazonlinux	16 k
at	aarch64	3.1.23-6.amzn2023.0.1	amazonlinux	60 k
attr	aarch64	2.5.1-3.amzn2023.0.1	amazonlinux	59 k
audit	aarch64	3.0.6-1.amzn2023.0.1	amazonlinux	249 k
audit-libs	aarch64	3.0.6-1.amzn2023.0.1	amazonlinux	116 k
aws-c-auth-libs	aarch64	0.6.5-6.amzn2023.0.2	amazonlinux	79 k
aws-c-cal-libs	aarch64	0.5.12-7.amzn2023.0.2	amazonlinux	34 k
aws-c-common-libs	aarch64	0.6.14-6.amzn2023.0.2	amazonlinux	119 k
aws-c-compression-libs	aarch64	0.2.14-5.amzn2023.0.2	amazonlinux	22 k
aws-c-event-stream-libs	aarch64	0.2.7-5.amzn2023.0.2	amazonlinux	47 k
aws-c-http-libs	aarch64	0.6.8-6.amzn2023.0.2	amazonlinux	147 k
aws-c-io-libs	aarch64	0.10.12-5.amzn2023.0.6	amazonlinux	109 k
aws-c-mqtt-libs	aarch64	0.7.8-7.amzn2023.0.2	amazonlinux	61 k
aws-c-s3-libs	aarch64	0.1.27-5.amzn2023.0.3	amazonlinux	54 k
aws-c-sdkutils-libs	aarch64	0.1.1-5.amzn2023.0.2	amazonlinux	26 k
aws-checksums-libs	aarch64	0.1.12-5.amzn2023.0.2	amazonlinux	50 k
awscli-2	noarch	2.7.8-1.amzn2023.0.4	amazonlinux	7.3 M
basesystem	noarch	11-11.amzn2023.0.1	amazonlinux	7.8 k
bash	aarch64	5.1.8-2.amzn2023.0.1	amazonlinux	1.6 M
bash-completion	noarch	1:2.11-2.amzn2023.0.1	amazonlinux	292 k
bc	aarch64	1.07.1-14.amzn2023.0.1	amazonlinux	120 k
bind-libs	aarch64	32:9.16.27-1.amzn2023.0.1	amazonlinux	1.2 M
bind-license	noarch	32:9.16.27-1.amzn2023.0.1	amazonlinux	14 k
bind-utils	aarch64	32:9.16.27-1.amzn2023.0.1	amazonlinux	206 k
binutils	aarch64	2.38-20.amzn2023.0.3	amazonlinux	4.6 M
boost-filesystem	aarch64	1.75.0-4.amzn2023.0.1	amazonlinux	55 k
boost-system	aarch64	1.75.0-4.amzn2023.0.1	amazonlinux	14 k
boost-thread	aarch64	1.75.0-4.amzn2023.0.1	amazonlinux	54 k
bzip2	aarch64	1.0.8-6.amzn2023.0.1	amazonlinux	53 k
bzip2-libs	aarch64	1.0.8-6.amzn2023.0.1	amazonlinux	44 k
c-ares	aarch64	1.17.2-1.amzn2023.0.1	amazonlinux	107 k
ca-certificates	noarch	2021.2.50-1.0.amzn2023.0.3	amazonlinux	343 k
checkpolicy	aarch64	3.4-3.amzn2023.0.1	amazonlinux	345 k
chkconfig	aarch64	1.15-2.amzn2023.0.1	amazonlinux	162 k
chrony	aarch64	4.2-7.amzn2023.0.4	amazonlinux	314 k
cloud-init	noarch	22.2.2-1.amzn2023.1.7	amazonlinux	1.1 M
cloud-utils-growpart	aarch64	0.31-8.amzn2023.0.2	amazonlinux	31 k
coreutils	aarch64	8.32-30.amzn2023.0.2	amazonlinux	1.1 M
coreutils-common	aarch64	8.32-30.amzn2023.0.2	amazonlinux	2.0 M
cpio	aarch64	2.13-10.amzn2023.0.1	amazonlinux	269 k
cracklib	aarch64	2.9.6-27.amzn2023.0.1	amazonlinux	83 k
cracklib-dicts	aarch64	2.9.6-27.amzn2023.0.1	amazonlinux	3.6 M
crontabs	noarch	1.11-24.20190603git.amzn2023.0.1		

```

                                amazonlinux 19 k
crypto-policies                noarch 20230128-1.gitdfb10ea.amzn2023.0.1
                                amazonlinux 61 k
crypto-policies-scripts       noarch 20230128-1.gitdfb10ea.amzn2023.0.1
                                amazonlinux 81 k
...
Installing dependencies:
amazon-linux-repo-cdn         noarch 2023.0.20230210-0.amzn2023  amazonlinux 16 k
xxhash-libs                  aarch64 0.8.0-3.amzn2023.0.1      amazonlinux 32 k
Installing weak dependencies:
amazon-chroney-config        noarch 4.2-7.amzn2023.0.4              amazonlinux 14 k
gawk-all-langpacks          aarch64 5.1.0-3.amzn2023.0.1          amazonlinux 207 k

Transaction Summary
=====
Install      5 Packages
Upgrade    413 Packages

Total download size: 199 M

```

Note

Se você usou a variável de substituição `/etc/dnf/vars/releasever`, use o comando a seguir para restaurar o comportamento de bloqueio padrão apagando o valor de substituição.

```
$ sudo rm /etc/dnf/vars/releasever
```

O uso de uma substituição persistente ao uso, `latest` em vez de uma versão específica, é semelhante ao comportamento padrão do. AL2 Existem serviços criados AMIs com base nos AL2 quais desabilitam esse comportamento e bloqueiam versões de pacotes específicas, como as que você usa por padrão AL2023.

Em vez de desabilitar as atualizações determinísticas, recomendamos substituir as instâncias por aquelas inicializadas de uma nova AMI. Se a substituição de instâncias não for uma opção, recomendamos o uso de ferramentas, como o [Gerenciador de Patches do AWS Systems Manager](#), para orquestrar a aplicação de atualizações em uma frota. [O EC2 Image Builder](#) também pode criar, corrigir e testar automaticamente suas AMIs próprias imagens AL2023 derivadas de imagens básicas. Você também pode [Receber notificações sobre novas atualizações](#), o que pode ser usado para acionar seus próprios pipelines de desenvolvimento de AMIs.

O uso de `latest` em um ambiente de pré-produção e, depois, a implantação em produção usando `latest` não oferece proteção contra nenhum problema entre uma atualização do sistema operacional e sua aplicação. Uma nova AL2023 versão pode ocorrer a qualquer momento e, portanto, todos os usos `latest` na produção acarretam riscos.

Gerencie atualizações de pacotes e sistemas operacionais no AL2023

Diferentemente das versões anteriores do Amazon Linux, AL2023 AMIs estão bloqueadas em uma versão específica do repositório Amazon Linux. Para aplicar correções de segurança e de erros a uma AL2023 instância, atualize a DNF configuração para a versão mais recente disponível. Como alternativa, execute uma AL2023 instância mais nova.

Esta seção descreve como gerenciar pacotes e repositórios de DNF em uma instância em execução. Também descreve como configurar DNF a partir de um script de dados do usuário para habilitar o repositório Amazon Linux mais recente disponível no momento do lançamento. Para obter mais informações, consulte [Referência de comandos da DNF](#).

É recomendável aplicar todas as atualizações disponíveis em uma nova AL2023 versão. Escolher apenas atualizações de segurança ou apenas atualizações específicas deve ser a exceção, e não a regra. Para listar quais [Recomendações de segurança](#) são relevantes para uma instância específica, consulte [Listar recomendações aplicáveis](#). Para obter informações sobre como instalar somente atualizações relevantes para uma [recomendação](#) específica, consulte [Aplicar atualizações de segurança no local](#).

Important

Se você quiser denunciar uma vulnerabilidade ou tiver uma preocupação de segurança em relação a serviços em AWS nuvem ou projetos de código aberto, entre em contato com a AWS Segurança usando a [página Relatórios de vulnerabilidades](#)

Tópicos

- [Verificar as atualizações de pacotes disponíveis](#)
- [Aplicar atualizações de segurança usando DNF e versões do repositório](#)
- [Reinício automático do serviço após atualizações \(de segurança\)](#)

- [Quando é necessário reinicializar para aplicar as atualizações de segurança?](#)
- [Lançamento de uma instância com a versão mais recente do repositório ativada](#)
- [Obtendo informações de suporte do pacote](#)
- [Verificar versões mais recentes do repositório com `dnf check-release-update`](#)
- [Adicionar, habilitar ou desabilitar novos repositórios](#)
- [Adicionando repositórios com `cloud-init`](#)

Verificar as atualizações de pacotes disponíveis

Você pode usar o comando `dnf check-update` para verificar se há atualizações no seu sistema. Para isso AL2023, recomendamos que você adicione a `--releasever=version-number` opção ao comando.

Ao adicionar essa opção, DNF verifica também se há atualizações para uma versão posterior do repositório. Por exemplo, depois de executar o comando `dnf check-update`, use a última versão retornada como o valor para o `version-number`.

Se a instância for atualizada para usar a versão mais recente do repositório, a saída incluirá uma lista de todos os pacotes a serem atualizados.

Note

Se você não especificar a versão de lançamento com o sinalizador opcional no comando `dnf check-update`, somente a versão do repositório atualmente configurada será verificada. Isso significa que os pacotes na versão posterior do repositório não são verificados.

Updates in a specific version

Neste exemplo, veremos quais atualizações estão disponíveis no lançamento [2023.1.20230628](#) ao inicializar um contêiner do lançamento [2023.0.20230315](#).

Note

Este exemplo usa as versões [2023.0.20230315](#) e [2023.1.20230628](#), e essas não são as versões mais recentes do AL2023. Consulte as [notas](#) de lançamento das versões mais recentes, que contêm as AL2023 atualizações de segurança mais recentes.

Neste exemplo, começaremos com uma imagem de contêiner para o lançamento

[2023.0.20230315](#).

Primeiro, buscamos essa imagem de contêiner no registro de contêineres. O trecho `.0` no final indica a versão da imagem para um lançamento específico, e essa versão da imagem geralmente é 0.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
2023.0.20230315.0: Pulling from amazonlinux/amazonlinux
b76f3b09316a: Pull complete
Digest: sha256:94e7183b0739140dbd5b639fb7600f0a2299cec5df8780c26d9cb409da5315a9
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.0.20230315.0
public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
```

Agora podemos gerar um shell dentro do contêiner, por meio do qual verificaremos se há atualizações.

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
bash-5.2#
```

O comando `dnf check-update` agora é usado para verificar as atualizações disponíveis no lançamento [2023.1.20230628](#).

Note

A aplicação de atualizações de pacotes é uma operação privilegiada. Embora normalmente não seja necessário elevar privilégios ao executar em um contêiner, se estiver executando em um ambiente sem contêineres, como uma instância do Amazon EC2, você poderá verificar se há atualizações sem elevar os privilégios.

```
$ dnf check-update --releasever=2023.1.20230628
Amazon Linux 2023 repository           60 MB/s | 15 MB      00:00
Last metadata expiration check: 0:00:02 ago on Mon Jul 22 17:25:34 2024.

amazon-linux-repo-cdn.noarch           2023.1.20230628-0.amzn2023      amazonlinux
ca-certificates.noarch                 2023.2.60-1.0.amzn2023.0.2     amazonlinux
curl-minimal.x86_64                   8.0.1-1.amzn2023               amazonlinux
```

glib2.x86_64	2.74.7-688.amzn2023.0.1	amazonlinux
glibc.x86_64	2.34-52.amzn2023.0.3	amazonlinux
glibc-common.x86_64	2.34-52.amzn2023.0.3	amazonlinux
glibc-minimal-langpack.x86_64	2.34-52.amzn2023.0.3	amazonlinux
gnupg2-minimal.x86_64	2.3.7-1.amzn2023.0.4	amazonlinux
keyutils-libs.x86_64	1.6.3-1.amzn2023	amazonlinux
libcap.x86_64	2.48-2.amzn2023.0.3	amazonlinux
libcurl-minimal.x86_64	8.0.1-1.amzn2023	amazonlinux
libgcc.x86_64	11.3.1-4.amzn2023.0.3	amazonlinux
libgomp.x86_64	11.3.1-4.amzn2023.0.3	amazonlinux
libstdc++.x86_64	11.3.1-4.amzn2023.0.3	amazonlinux
libxml2.x86_64	2.10.4-1.amzn2023.0.1	amazonlinux
ncurses-base.noarch	6.2-4.20200222.amzn2023.0.4	amazonlinux
ncurses-libs.x86_64	6.2-4.20200222.amzn2023.0.4	amazonlinux
openssl-libs.x86_64	1:3.0.8-1.amzn2023.0.3	amazonlinux
python3-rpm.x86_64	4.16.1.3-12.amzn2023.0.6	amazonlinux
rpm.x86_64	4.16.1.3-12.amzn2023.0.6	amazonlinux
rpm-build-libs.x86_64	4.16.1.3-12.amzn2023.0.6	amazonlinux
rpm-libs.x86_64	4.16.1.3-12.amzn2023.0.6	amazonlinux
rpm-sign-libs.x86_64	4.16.1.3-12.amzn2023.0.6	amazonlinux
system-release.noarch	2023.1.20230628-0.amzn2023	amazonlinux
tzdata.noarch	2023c-1.amzn2023.0.1	amazonlinux
bash-5.2#		

A versão do pacote `system-release` mostra o lançamento para o qual um comando `dnf upgrade` faria a atualização, que é o lançamento [2023.1.20230628](#) solicitado no comando `dnf check-update --releasever=2023.1.20230628`.

Updates in the latest version

Neste exemplo, veremos quais atualizações estão disponíveis na versão de AL2023 se lançarmos um contêiner da latest versão [2023.4.20240319](#). No momento em que este artigo foi escrito, o lançamento latest era [2023.5.20240708](#), portanto, as atualizações listadas neste exemplo serão as desse lançamento.

Note

Este exemplo usa os lançamentos [2023.4.20240319](#) e [2023.5.20240708](#), sendo que este é o lançamento mais recente no momento em que este artigo foi escrito. Para obter mais informações sobre as versões mais recentes, consulte as [Notas AL2023 de versão](#).

Neste exemplo, começaremos com uma imagem de contêiner para o lançamento

[2023.4.20240319](#).

Primeiro, buscamos essa imagem de contêiner no registro de contêineres. O trecho `.1` no final indica a versão da imagem para um lançamento específico. Embora a versão da imagem geralmente seja 0, este exemplo usa um lançamento em que a versão da imagem é 1.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
2023.4.20240319.1: Pulling from amazonlinux/amazonlinux
6de065fda9a2: Pull complete
Digest: sha256:b4838c4cc9211d966b6ea158dacc9eda7433a16ba94436508c2d9f01f7658b4e
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.4.20240319.1
public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
```

Agora podemos gerar um shell dentro do contêiner, por meio do qual verificaremos se há atualizações.

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
bash-5.2#
```

O comando `dnf check-update` agora é usado para verificar as atualizações disponíveis no lançamento `latest` que, no momento em que este artigo foi escrito, era [2023.5.20240708](#).

Note

A aplicação de atualizações de pacotes é uma operação privilegiada. Embora normalmente não seja necessário elevar privilégios ao executar em um contêiner, se estiver executando em um ambiente sem contêineres, como uma instância do Amazon EC2, você poderá verificar se há atualizações sem elevar os privilégios.

```
$ dnf --releasever=latest check-update
Amazon Linux 2023 repository              78 MB/s | 25 MB      00:00
Last metadata expiration check: 0:00:04 ago on Mon Jul 22 17:39:13 2024.

amazon-linux-repo-cdn.noarch              2023.5.20240708-1.amzn2023      amazonlinux
curl-minimal.x86_64                      8.5.0-1.amzn2023.0.4           amazonlinux
dnf.noarch                                4.14.0-1.amzn2023.0.5          amazonlinux
dnf-data.noarch                           4.14.0-1.amzn2023.0.5          amazonlinux
```

expat.x86_64	2.5.0-1.amzn2023.0.4	amazonlinux
glibc.x86_64	2.34-52.amzn2023.0.10	amazonlinux
glibc-common.x86_64	2.34-52.amzn2023.0.10	amazonlinux
glibc-minimal-langpack.x86_64	2.34-52.amzn2023.0.10	amazonlinux
krb5-libs.x86_64	1.21-3.amzn2023.0.4	amazonlinux
libblkid.x86_64	2.37.4-1.amzn2023.0.4	amazonlinux
libcurl-minimal.x86_64	8.5.0-1.amzn2023.0.4	amazonlinux
libmount.x86_64	2.37.4-1.amzn2023.0.4	amazonlinux
libnghttp2.x86_64	1.59.0-3.amzn2023.0.1	amazonlinux
libsmartcols.x86_64	2.37.4-1.amzn2023.0.4	amazonlinux
libuuid.x86_64	2.37.4-1.amzn2023.0.4	amazonlinux
openssl-libs.x86_64	1:3.0.8-1.amzn2023.0.12	amazonlinux
python3.x86_64	3.9.16-1.amzn2023.0.8	amazonlinux
python3-dnf.noarch	4.14.0-1.amzn2023.0.5	amazonlinux
python3-libs.x86_64	3.9.16-1.amzn2023.0.8	amazonlinux
system-release.noarch	2023.5.20240708-1.amzn2023	amazonlinux
yum.noarch	4.14.0-1.amzn2023.0.5	amazonlinux
bash-5.2#		

A versão do pacote `system-release` mostra o lançamento para o qual um comando `dnf upgrade` seria atualizado.

Para esse comando, se houver pacotes mais novos disponíveis, o código de retorno será 100. Se não houver pacotes mais novos disponíveis, o código de retorno será 0. Além disso, a saída também lista todos os pacotes a serem atualizados.

Aplicar atualizações de segurança usando DNF e versões do repositório

Novas atualizações de pacotes e atualizações de segurança são disponibilizadas somente para novas versões do repositório. Para instâncias que você executou a partir de versões anteriores da AL2023 AMI, você deve atualizar a versão do repositório antes de poder instalar as atualizações de segurança. O comando `dnf check-release-update` inclui um exemplo de comando `update` que atualiza todos os pacotes instalados no sistema para versões em um repositório mais novo.

Note

Se você não especificar a versão de lançamento com o sinalizador opcional no comando `dnf check-update`, somente a versão do repositório atualmente configurada será verificada. Isso significa que nenhuma atualização dos pacotes instalados presentes em qualquer versão posterior do repositório será aplicada.

Esta seção aborda o caminho de atualização recomendado para aplicar todas as atualizações disponíveis em vez de escolher atualizações individuais ou somente aquelas marcadas como atualizações de segurança. Ao aplicar todas as atualizações, as instâncias existentes são movidas para o mesmo conjunto de pacotes que a inicialização de uma AMI atualizada. Essa consistência reduz a variação das versões de pacotes em uma frota. Para saber mais sobre como aplicar atualizações específicas, consulte [Aplicar atualizações de segurança no local](#).

Applying updates in a specific version

Neste exemplo, aplicaremos as atualizações disponíveis no lançamento [2023.1.20230628](#) ao inicializar um contêiner do lançamento [2023.0.20230315](#).

Note

Este exemplo usa as versões [2023.0.20230315](#) e [2023.1.20230628](#), e essas não são as versões mais recentes do AL2023. Consulte as [notas](#) de lançamento das versões mais recentes, que contêm as AL2023 atualizações de segurança mais recentes.

Neste exemplo, começaremos com uma imagem de contêiner para o lançamento [2023.0.20230315](#).

Primeiro, buscamos essa imagem de contêiner no registro de contêineres. O trecho `.0` no final indica a versão da imagem para um lançamento específico, e essa versão da imagem geralmente é 0.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
2023.0.20230315.0: Pulling from amazonlinux/amazonlinux
b76f3b09316a: Pull complete
Digest: sha256:94e7183b0739140dbd5b639fb7600f0a2299cec5df8780c26d9cb409da5315a9
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.0.20230315.0
public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
```

Agora podemos gerar um shell dentro do contêiner, por meio do qual aplicaremos as atualizações.

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
bash-5.2#
```

O comando `dnf upgrade` agora é usado para aplicar todas as atualizações presentes no lançamento [2023.1.20230628](#).

Note

A aplicação de atualizações de pacotes é uma operação privilegiada. Embora normalmente não seja necessário elevar privilégios ao executar em um contêiner, se estiver executando em um ambiente sem contêineres, como uma instância do Amazon EC2, você precisará executar o comando `dnf upgrade` como usuário `root`. Isso pode ser feito utilizando os comandos `sudo` ou `su`.

```
$ dnf upgrade --releasever=2023.1.20230628
```

```
Amazon Linux 2023 repository                      38 MB/s | 15 MB      00:00
```

```
Last metadata expiration check: 0:00:02 ago on Mon Jul 22 17:49:08 2024.
```

```
Dependencies resolved.
```

Package	Arch	Version	Repository	Size
Upgrading:				
amazon-linux-repo-cdn	noarch	2023.1.20230628-0.amzn2023	amazonlinux	18 k
ca-certificates	noarch	2023.2.60-1.0.amzn2023.0.2	amazonlinux	829 k
curl-minimal	x86_64	8.0.1-1.amzn2023	amazonlinux	150 k
glib2	x86_64	2.74.7-688.amzn2023.0.1	amazonlinux	2.7 M
glibc	x86_64	2.34-52.amzn2023.0.3	amazonlinux	1.9 M
glibc-common	x86_64	2.34-52.amzn2023.0.3	amazonlinux	307 k
glibc-minimal-langpack	x86_64	2.34-52.amzn2023.0.3	amazonlinux	35 k
gnupg2-minimal	x86_64	2.3.7-1.amzn2023.0.4	amazonlinux	421 k
keyutils-libs	x86_64	1.6.3-1.amzn2023	amazonlinux	33 k
libcap	x86_64	2.48-2.amzn2023.0.3	amazonlinux	67 k
libcurl-minimal	x86_64	8.0.1-1.amzn2023	amazonlinux	249 k
libgcc	x86_64	11.3.1-4.amzn2023.0.3	amazonlinux	105 k
libgomp	x86_64	11.3.1-4.amzn2023.0.3	amazonlinux	280 k
libstdc++	x86_64	11.3.1-4.amzn2023.0.3	amazonlinux	744 k
libxml2	x86_64	2.10.4-1.amzn2023.0.1	amazonlinux	706 k
ncurses-base	noarch	6.2-4.20200222.amzn2023.0.4	amazonlinux	60 k
ncurses-libs	x86_64	6.2-4.20200222.amzn2023.0.4	amazonlinux	328 k
openssl-libs	x86_64	1:3.0.8-1.amzn2023.0.3	amazonlinux	2.2 M
python3-rpm	x86_64	4.16.1.3-12.amzn2023.0.6	amazonlinux	88 k
rpm	x86_64	4.16.1.3-12.amzn2023.0.6	amazonlinux	486 k
rpm-build-libs	x86_64	4.16.1.3-12.amzn2023.0.6	amazonlinux	90 k
rpm-libs	x86_64	4.16.1.3-12.amzn2023.0.6	amazonlinux	309 k

```

rpm-sign-libs          x86_64  4.16.1.3-12.amzn2023.0.6  amazonlinux  21 k
system-release         noarch  2023.1.20230628-0.amzn2023  amazonlinux  29 k
tzdata                 noarch  2023c-1.amzn2023.0.1      amazonlinux  433 k

```

Transaction Summary

```
=====
```

```
Upgrade  25 Packages
```

```
Total download size: 12 M
```

```
Is this ok [y/N]:
```

A versão do pacote `system-release` mostra o lançamento para o qual um comando `dnf upgrade` faria a atualização, que é o lançamento [2023.1.20230628](#) solicitado no comando `dnf upgrade --releasever=2023.1.20230628`.

Por padrão, `dnf` solicitará que você confirme se deseja aplicar as atualizações. Você pode ignorar esse prompt usando o sinalizador `-y` para `dnf`. Neste exemplo, o comando `dnf upgrade -y --releasever=2023.1.20230628` não solicitará confirmação antes de aplicar as atualizações. Isso é útil em scripts ou outros ambientes de automação.

Depois de confirmar que deseja aplicar as atualizações, `dnf` as aplica.

```
Is this ok [y/N]:y
```

```
Downloading Packages:
```

```

(1/25): libcap-2.48-2.amzn2023.0.3.x86_64.rpm    1.5 MB/s | 67 kB    00:00
(2/25): python3-rpm-4.16.1.3-12.amzn2023.0.6.x86 2.1 MB/s | 88 kB    00:00
(3/25): libcurl-minimal-8.0.1-1.amzn2023.x86_64. 2.6 MB/s | 249 kB    00:00
(4/25): glib2-2.74.7-688.amzn2023.0.1.x86_64.rpm 26 MB/s | 2.7 MB    00:00
(5/25): glibc-minimal-langpack-2.34-52.amzn2023. 1.3 MB/s | 35 kB    00:00
(6/25): rpm-build-libs-4.16.1.3-12.amzn2023.0.6. 2.8 MB/s | 90 kB    00:00
(7/25): rpm-libs-4.16.1.3-12.amzn2023.0.6.x86_64 6.6 MB/s | 309 kB    00:00
(8/25): libgcc-11.3.1-4.amzn2023.0.3.x86_64.rpm 3.9 MB/s | 105 kB    00:00
(9/25): glibc-common-2.34-52.amzn2023.0.3.x86_64 11 MB/s | 307 kB    00:00
(10/25): glibc-2.34-52.amzn2023.0.3.x86_64.rpm   31 MB/s | 1.9 MB    00:00
(11/25): rpm-sign-libs-4.16.1.3-12.amzn2023.0.6. 877 kB/s | 21 kB    00:00
(12/25): gnupg2-minimal-2.3.7-1.amzn2023.0.4.x86 15 MB/s | 421 kB    00:00
(13/25): openssl-libs-3.0.8-1.amzn2023.0.3.x86_6 35 MB/s | 2.2 MB    00:00
(14/25): libxml2-2.10.4-1.amzn2023.0.1.x86_64.rp 14 MB/s | 706 kB    00:00
(15/25): curl-minimal-8.0.1-1.amzn2023.x86_64.rp 4.2 MB/s | 150 kB    00:00
(16/25): rpm-4.16.1.3-12.amzn2023.0.6.x86_64.rpm 11 MB/s | 486 kB    00:00
(17/25): libgomp-11.3.1-4.amzn2023.0.3.x86_64.rp 7.0 MB/s | 280 kB    00:00
(18/25): libstdc++-11.3.1-4.amzn2023.0.3.x86_64. 14 MB/s | 744 kB    00:00
(19/25): keyutils-libs-1.6.3-1.amzn2023.x86_64.r 1.6 MB/s | 33 kB    00:00

```

```
(20/25): ncurses-libs-6.2-4.20200222.amzn2023.0. 10 MB/s | 328 kB 00:00
(21/25): tzdata-2023c-1.amzn2023.0.1.noarch.rpm 11 MB/s | 433 kB 00:00
(22/25): amazon-linux-repo-cdn-2023.1.20230628-0 781 kB/s | 18 kB 00:00
(23/25): ca-certificates-2023.2.60-1.0.amzn2023. 16 MB/s | 829 kB 00:00
(24/25): system-release-2023.1.20230628-0.amzn20 1.5 MB/s | 29 kB 00:00
(25/25): ncurses-base-6.2-4.20200222.amzn2023.0. 3.1 MB/s | 60 kB 00:00
```

```
-----
Total                               28 MB/s | 12 MB 00:00
```

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

```
Preparing      : 1/1
Upgrading      : libgcc-11.3.1-4.amzn2023.0.3.x86_64 1/50
Running scriptlet: libgcc-11.3.1-4.amzn2023.0.3.x86_64 1/50
Upgrading      : system-release-2023.1.20230628-0.amzn2023.noarch 2/50
Upgrading      : amazon-linux-repo-cdn-2023.1.20230628-0.amzn2023.no 3/50
Upgrading      : ncurses-base-6.2-4.20200222.amzn2023.0.4.noarch 4/50
Upgrading      : tzdata-2023c-1.amzn2023.0.1.noarch 5/50
Upgrading      : glibc-common-2.34-52.amzn2023.0.3.x86_64 6/50
Running scriptlet: glibc-2.34-52.amzn2023.0.3.x86_64 7/50
Upgrading      : glibc-2.34-52.amzn2023.0.3.x86_64 7/50
Running scriptlet: glibc-2.34-52.amzn2023.0.3.x86_64 7/50
Upgrading      : glibc-minimal-langpack-2.34-52.amzn2023.0.3.x86_64 8/50
Upgrading      : libcap-2.48-2.amzn2023.0.3.x86_64 9/50
Upgrading      : gnupg2-minimal-2.3.7-1.amzn2023.0.4.x86_64 10/50
Upgrading      : libgomp-11.3.1-4.amzn2023.0.3.x86_64 11/50
Running scriptlet: ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch 12/50
Upgrading      : ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch 12/50
Running scriptlet: ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch 12/50
Upgrading      : openssl-libs-1:3.0.8-1.amzn2023.0.3.x86_64 13/50
Upgrading      : libcurl-minimal-8.0.1-1.amzn2023.x86_64 14/50
Upgrading      : curl-minimal-8.0.1-1.amzn2023.x86_64 15/50
Upgrading      : rpm-libs-4.16.1.3-12.amzn2023.0.6.x86_64 16/50
Upgrading      : rpm-4.16.1.3-12.amzn2023.0.6.x86_64 17/50
Upgrading      : rpm-build-libs-4.16.1.3-12.amzn2023.0.6.x86_64 18/50
Upgrading      : rpm-sign-libs-4.16.1.3-12.amzn2023.0.6.x86_64 19/50
Upgrading      : python3-rpm-4.16.1.3-12.amzn2023.0.6.x86_64 20/50
Upgrading      : glib2-2.74.7-688.amzn2023.0.1.x86_64 21/50
Upgrading      : libxml2-2.10.4-1.amzn2023.0.1.x86_64 22/50
Upgrading      : libstdc++-11.3.1-4.amzn2023.0.3.x86_64 23/50
Upgrading      : keyutils-libs-1.6.3-1.amzn2023.x86_64 24/50
Upgrading      : ncurses-libs-6.2-4.20200222.amzn2023.0.4.x86_64 25/50
```

Cleanup	: glib2-2.73.2-680.amzn2023.0.3.x86_64	26/50
Cleanup	: libstdc++-11.3.1-4.amzn2023.0.2.x86_64	27/50
Cleanup	: libxml2-2.10.3-2.amzn2023.0.1.x86_64	28/50
Cleanup	: python3-rpm-4.16.1.3-12.amzn2023.0.5.x86_64	29/50
Cleanup	: rpm-build-libs-4.16.1.3-12.amzn2023.0.5.x86_64	30/50
Cleanup	: rpm-sign-libs-4.16.1.3-12.amzn2023.0.5.x86_64	31/50
Cleanup	: rpm-libs-4.16.1.3-12.amzn2023.0.5.x86_64	32/50
Cleanup	: libcap-2.48-2.amzn2023.0.2.x86_64	33/50
Cleanup	: gnupg2-minimal-2.3.7-1.amzn2023.0.3.x86_64	34/50
Cleanup	: ncurses-libs-6.2-4.20200222.amzn2023.0.3.x86_64	35/50
Cleanup	: libgomp-11.3.1-4.amzn2023.0.2.x86_64	36/50
Cleanup	: rpm-4.16.1.3-12.amzn2023.0.5.x86_64	37/50
Cleanup	: curl-minimal-7.88.1-1.amzn2023.0.1.x86_64	38/50
Cleanup	: libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64	39/50
Cleanup	: openssl-libs-1:3.0.8-1.amzn2023.0.1.x86_64	40/50
Cleanup	: keyutils-libs-1.6.1-2.amzn2023.0.2.x86_64	41/50
Cleanup	: amazon-linux-repo-cdn-2023.0.20230315-1.amzn2023.no	42/50
Cleanup	: system-release-2023.0.20230315-1.amzn2023.noarch	43/50
Cleanup	: ca-certificates-2023.2.60-1.0.amzn2023.0.1.noarch	44/50
Cleanup	: ncurses-base-6.2-4.20200222.amzn2023.0.3.noarch	45/50
Cleanup	: glibc-minimal-langpack-2.34-52.amzn2023.0.2.x86_64	46/50
Cleanup	: glibc-2.34-52.amzn2023.0.2.x86_64	47/50
Cleanup	: glibc-common-2.34-52.amzn2023.0.2.x86_64	48/50
Cleanup	: tzdata-2022g-1.amzn2023.0.1.noarch	49/50
Cleanup	: libgcc-11.3.1-4.amzn2023.0.2.x86_64	50/50
Running scriptlet:	libgcc-11.3.1-4.amzn2023.0.2.x86_64	50/50
Running scriptlet:	ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch	50/50
Running scriptlet:	rpm-4.16.1.3-12.amzn2023.0.6.x86_64	50/50
Running scriptlet:	libgcc-11.3.1-4.amzn2023.0.2.x86_64	50/50
Verifying	: libcurl-minimal-8.0.1-1.amzn2023.x86_64	1/50
Verifying	: libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64	2/50
Verifying	: libcap-2.48-2.amzn2023.0.3.x86_64	3/50
Verifying	: libcap-2.48-2.amzn2023.0.2.x86_64	4/50
Verifying	: glib2-2.74.7-688.amzn2023.0.1.x86_64	5/50
Verifying	: glib2-2.73.2-680.amzn2023.0.3.x86_64	6/50
Verifying	: python3-rpm-4.16.1.3-12.amzn2023.0.6.x86_64	7/50
Verifying	: python3-rpm-4.16.1.3-12.amzn2023.0.5.x86_64	8/50
Verifying	: glibc-minimal-langpack-2.34-52.amzn2023.0.3.x86_64	9/50
Verifying	: glibc-minimal-langpack-2.34-52.amzn2023.0.2.x86_64	10/50
Verifying	: rpm-libs-4.16.1.3-12.amzn2023.0.6.x86_64	11/50
Verifying	: rpm-libs-4.16.1.3-12.amzn2023.0.5.x86_64	12/50
Verifying	: rpm-build-libs-4.16.1.3-12.amzn2023.0.6.x86_64	13/50
Verifying	: rpm-build-libs-4.16.1.3-12.amzn2023.0.5.x86_64	14/50
Verifying	: glibc-2.34-52.amzn2023.0.3.x86_64	15/50

Verifying	: glibc-2.34-52.amzn2023.0.2.x86_64	16/50
Verifying	: libgcc-11.3.1-4.amzn2023.0.3.x86_64	17/50
Verifying	: libgcc-11.3.1-4.amzn2023.0.2.x86_64	18/50
Verifying	: glibc-common-2.34-52.amzn2023.0.3.x86_64	19/50
Verifying	: glibc-common-2.34-52.amzn2023.0.2.x86_64	20/50
Verifying	: rpm-sign-libs-4.16.1.3-12.amzn2023.0.6.x86_64	21/50
Verifying	: rpm-sign-libs-4.16.1.3-12.amzn2023.0.5.x86_64	22/50
Verifying	: openssl-libs-1:3.0.8-1.amzn2023.0.3.x86_64	23/50
Verifying	: openssl-libs-1:3.0.8-1.amzn2023.0.1.x86_64	24/50
Verifying	: gnupg2-minimal-2.3.7-1.amzn2023.0.4.x86_64	25/50
Verifying	: gnupg2-minimal-2.3.7-1.amzn2023.0.3.x86_64	26/50
Verifying	: libxml2-2.10.4-1.amzn2023.0.1.x86_64	27/50
Verifying	: libxml2-2.10.3-2.amzn2023.0.1.x86_64	28/50
Verifying	: curl-minimal-8.0.1-1.amzn2023.x86_64	29/50
Verifying	: curl-minimal-7.88.1-1.amzn2023.0.1.x86_64	30/50
Verifying	: rpm-4.16.1.3-12.amzn2023.0.6.x86_64	31/50
Verifying	: rpm-4.16.1.3-12.amzn2023.0.5.x86_64	32/50
Verifying	: libstdc++-11.3.1-4.amzn2023.0.3.x86_64	33/50
Verifying	: libstdc++-11.3.1-4.amzn2023.0.2.x86_64	34/50
Verifying	: libgomp-11.3.1-4.amzn2023.0.3.x86_64	35/50
Verifying	: libgomp-11.3.1-4.amzn2023.0.2.x86_64	36/50
Verifying	: keyutils-libs-1.6.3-1.amzn2023.x86_64	37/50
Verifying	: keyutils-libs-1.6.1-2.amzn2023.0.2.x86_64	38/50
Verifying	: ncurses-libs-6.2-4.20200222.amzn2023.0.4.x86_64	39/50
Verifying	: ncurses-libs-6.2-4.20200222.amzn2023.0.3.x86_64	40/50
Verifying	: ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch	41/50
Verifying	: ca-certificates-2023.2.60-1.0.amzn2023.0.1.noarch	42/50
Verifying	: tzdata-2023c-1.amzn2023.0.1.noarch	43/50
Verifying	: tzdata-2022g-1.amzn2023.0.1.noarch	44/50
Verifying	: amazon-linux-repo-cdn-2023.1.20230628-0.amzn2023.no	45/50
Verifying	: amazon-linux-repo-cdn-2023.0.20230315-1.amzn2023.no	46/50
Verifying	: system-release-2023.1.20230628-0.amzn2023.noarch	47/50
Verifying	: system-release-2023.0.20230315-1.amzn2023.noarch	48/50
Verifying	: ncurses-base-6.2-4.20200222.amzn2023.0.4.noarch	49/50
Verifying	: ncurses-base-6.2-4.20200222.amzn2023.0.3.noarch	50/50

Upgraded:

```
amazon-linux-repo-cdn-2023.1.20230628-0.amzn2023.noarch
ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch
curl-minimal-8.0.1-1.amzn2023.x86_64
glib2-2.74.7-688.amzn2023.0.1.x86_64
glibc-2.34-52.amzn2023.0.3.x86_64
glibc-common-2.34-52.amzn2023.0.3.x86_64
glibc-minimal-langpack-2.34-52.amzn2023.0.3.x86_64
```

```
gnupg2-minimal-2.3.7-1.amzn2023.0.4.x86_64
keyutils-libs-1.6.3-1.amzn2023.x86_64
libcap-2.48-2.amzn2023.0.3.x86_64
libcurl-minimal-8.0.1-1.amzn2023.x86_64
libgcc-11.3.1-4.amzn2023.0.3.x86_64
libgomp-11.3.1-4.amzn2023.0.3.x86_64
libstdc++-11.3.1-4.amzn2023.0.3.x86_64
libxml2-2.10.4-1.amzn2023.0.1.x86_64
ncurses-base-6.2-4.20200222.amzn2023.0.4.noarch
ncurses-libs-6.2-4.20200222.amzn2023.0.4.x86_64
openssl-libs-1:3.0.8-1.amzn2023.0.3.x86_64
python3-rpm-4.16.1.3-12.amzn2023.0.6.x86_64
rpm-4.16.1.3-12.amzn2023.0.6.x86_64
rpm-build-libs-4.16.1.3-12.amzn2023.0.6.x86_64
rpm-libs-4.16.1.3-12.amzn2023.0.6.x86_64
rpm-sign-libs-4.16.1.3-12.amzn2023.0.6.x86_64
system-release-2023.1.20230628-0.amzn2023.noarch
tzdata-2023c-1.amzn2023.0.1.noarch
```

Complete!

bash-5.2#

Updates in the latest version

Neste exemplo, aplicaremos as atualizações disponíveis na versão de AL2023 se lançarmos um contêiner da latest versão [2023.4.20240319](#). No momento em que este artigo foi escrito, o lançamento latest era [2023.5.20240708](#), portanto, as atualizações listadas neste exemplo serão as desse lançamento.

Note

Este exemplo usa os lançamentos [2023.4.20240319](#) e [2023.5.20240708](#), sendo que este é o lançamento mais recente no momento em que este artigo foi escrito. Para obter mais informações sobre as versões mais recentes, consulte as [Notas AL2023 de versão](#).

Neste exemplo, começaremos com uma imagem de contêiner para o lançamento [2023.4.20240319](#).

Primeiro, buscamos essa imagem de contêiner no registro de contêineres. O trecho `.1` no final indica a versão da imagem para um lançamento específico. Embora a versão da imagem geralmente seja 0, este exemplo usa um lançamento em que a versão da imagem é 1.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
2023.4.20240319.1: Pulling from amazonlinux/amazonlinux
6de065fda9a2: Pull complete
Digest: sha256:b4838c4cc9211d966b6ea158dacc9eda7433a16ba94436508c2d9f01f7658b4e
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.4.20240319.1
public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
```

Agora podemos gerar um shell dentro do contêiner, por meio do qual aplicaremos as atualizações.

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
bash-5.2#
```

O comando `dnf upgrade` agora é usado para aplicar as atualizações disponíveis no lançamento `latest` que, no momento em que este artigo foi escrito, era [2023.5.20240708](#).

Note

A aplicação de atualizações de pacotes é uma operação privilegiada. Embora normalmente não seja necessário elevar privilégios ao executar em um contêiner, se estiver executando em um ambiente sem contêineres, como uma instância do Amazon EC2, você precisará executar o comando `dnf upgrade` como usuário `root`. Isso pode ser feito utilizando os comandos `sudo` ou `su`.

Por padrão, `dnf` solicitará que você confirme se deseja aplicar as atualizações. Neste exemplo, ignoramos esse prompt usando o sinalizador `-y` para `dnf`.

```
$ dnf -y --releasever=latest update
Amazon Linux 2023 repository              75 MB/s | 25 MB      00:00
Last metadata expiration check: 0:00:04 ago on Mon Jul 22 18:00:10 2024.
Dependencies resolved.
=====
Package                                Arch      Version                                Repository      Size
=====
Upgrading:
amazon-linux-repo-cdn                  noarch    2023.5.20240708-1.amzn2023            amazonlinux     17 k
curl-minimal                           x86_64    8.5.0-1.amzn2023.0.4                  amazonlinux     160 k
dnf                                     noarch    4.14.0-1.amzn2023.0.5                  amazonlinux     460 k
```


dnf-data	noarch	4.14.0-1.amzn2023.0.5	amazonlinux	34 k
expat	x86_64	2.5.0-1.amzn2023.0.4	amazonlinux	117 k
glibc	x86_64	2.34-52.amzn2023.0.10	amazonlinux	1.9 M
glibc-common	x86_64	2.34-52.amzn2023.0.10	amazonlinux	295 k
glibc-minimal-langpack	x86_64	2.34-52.amzn2023.0.10	amazonlinux	23 k
krb5-libs	x86_64	1.21-3.amzn2023.0.4	amazonlinux	758 k
libblkid	x86_64	2.37.4-1.amzn2023.0.4	amazonlinux	105 k
libcurl-minimal	x86_64	8.5.0-1.amzn2023.0.4	amazonlinux	275 k
libmount	x86_64	2.37.4-1.amzn2023.0.4	amazonlinux	132 k
libnghttp2	x86_64	1.59.0-3.amzn2023.0.1	amazonlinux	79 k
libsmartcols	x86_64	2.37.4-1.amzn2023.0.4	amazonlinux	62 k
libuuid	x86_64	2.37.4-1.amzn2023.0.4	amazonlinux	26 k
openssl-libs	x86_64	1:3.0.8-1.amzn2023.0.12	amazonlinux	2.2 M
python3	x86_64	3.9.16-1.amzn2023.0.8	amazonlinux	27 k
python3-dnf	noarch	4.14.0-1.amzn2023.0.5	amazonlinux	409 k
python3-libs	x86_64	3.9.16-1.amzn2023.0.8	amazonlinux	7.3 M
system-release	noarch	2023.5.20240708-1.amzn2023	amazonlinux	28 k
yum	noarch	4.14.0-1.amzn2023.0.5	amazonlinux	32 k

Transaction Summary

=====

Upgrade 21 Packages

Total download size: 14 M

Downloading Packages:

(1/21): amazon-linux-repo-cdn-2023.5.20240708-1.	345 kB/s 17 kB	00:00
(2/21): dnf-4.14.0-1.amzn2023.0.5.noarch.rpm	6.8 MB/s 460 kB	00:00
(3/21): dnf-data-4.14.0-1.amzn2023.0.5.noarch.rp	1.6 MB/s 34 kB	00:00
(4/21): expat-2.5.0-1.amzn2023.0.4.x86_64.rpm	4.6 MB/s 117 kB	00:00
(5/21): glibc-2.34-52.amzn2023.0.10.x86_64.rpm	38 MB/s 1.9 MB	00:00
(6/21): glibc-common-2.34-52.amzn2023.0.10.x86_6	8.8 MB/s 295 kB	00:00
(7/21): glibc-minimal-langpack-2.34-52.amzn2023.	1.7 MB/s 23 kB	00:00
(8/21): curl-minimal-8.5.0-1.amzn2023.0.4.x86_64	998 kB/s 160 kB	00:00
(9/21): libblkid-2.37.4-1.amzn2023.0.4.x86_64.rp	4.1 MB/s 105 kB	00:00
(10/21): krb5-libs-1.21-3.amzn2023.0.4.x86_64.rp	16 MB/s 758 kB	00:00
(11/21): libmount-2.37.4-1.amzn2023.0.4.x86_64.r	7.9 MB/s 132 kB	00:00
(12/21): libnghttp2-1.59.0-3.amzn2023.0.1.x86_64	5.6 MB/s 79 kB	00:00
(13/21): libsmartcols-2.37.4-1.amzn2023.0.4.x86_	4.4 MB/s 62 kB	00:00
(14/21): libcurl-minimal-8.5.0-1.amzn2023.0.4.x8	7.1 MB/s 275 kB	00:00
(15/21): libuuid-2.37.4-1.amzn2023.0.4.x86_64.rp	1.1 MB/s 26 kB	00:00
(16/21): python3-3.9.16-1.amzn2023.0.8.x86_64.rp	1.5 MB/s 27 kB	00:00
(17/21): python3-dnf-4.14.0-1.amzn2023.0.5.noarc	19 MB/s 409 kB	00:00
(18/21): system-release-2023.5.20240708-1.amzn20	1.9 MB/s 28 kB	00:00
(19/21): yum-4.14.0-1.amzn2023.0.5.noarch.rpm	1.6 MB/s 32 kB	00:00

```

(20/21): openssl-libs-3.0.8-1.amzn2023.0.12.x86_ 26 MB/s | 2.2 MB      00:00
(21/21): python3-libs-3.9.16-1.amzn2023.0.8.x86_ 59 MB/s | 7.3 MB      00:00
-----
Total                                          34 MB/s | 14 MB      00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                                : 1/1
  Upgrading                                : glibc-common-2.34-52.amzn2023.0.10.x86_64 1/42
  Upgrading                                : glibc-minimal-langpack-2.34-52.amzn2023.0.10.x86_64 2/42
  Running scriptlet: glibc-2.34-52.amzn2023.0.10.x86_64 3/42
  Upgrading                                : glibc-2.34-52.amzn2023.0.10.x86_64 3/42
  Running scriptlet: glibc-2.34-52.amzn2023.0.10.x86_64 3/42
  Upgrading                                : libuuid-2.37.4-1.amzn2023.0.4.x86_64 4/42
  Upgrading                                : openssl-libs-1:3.0.8-1.amzn2023.0.12.x86_64 5/42
  Upgrading                                : krb5-libs-1.21-3.amzn2023.0.4.x86_64 6/42
  Upgrading                                : libblkid-2.37.4-1.amzn2023.0.4.x86_64 7/42
  Running scriptlet: libblkid-2.37.4-1.amzn2023.0.4.x86_64 7/42
  Upgrading                                : expat-2.5.0-1.amzn2023.0.4.x86_64 8/42
  Upgrading                                : python3-3.9.16-1.amzn2023.0.8.x86_64 9/42
  Upgrading                                : python3-libs-3.9.16-1.amzn2023.0.8.x86_64 10/42
  Upgrading                                : libnghttp2-1.59.0-3.amzn2023.0.1.x86_64 11/42
  Upgrading                                : libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64 12/42
  Upgrading                                : system-release-2023.5.20240708-1.amzn2023.noarch 13/42
  Upgrading                                : amazon-linux-repo-cdn-2023.5.20240708-1.amzn2023.no 14/42
  Upgrading                                : dnf-data-4.14.0-1.amzn2023.0.5.noarch 15/42
  Upgrading                                : python3-dnf-4.14.0-1.amzn2023.0.5.noarch 16/42
  Upgrading                                : dnf-4.14.0-1.amzn2023.0.5.noarch 17/42
  Running scriptlet: dnf-4.14.0-1.amzn2023.0.5.noarch 17/42
  Upgrading                                : yum-4.14.0-1.amzn2023.0.5.noarch 18/42
  Upgrading                                : curl-minimal-8.5.0-1.amzn2023.0.4.x86_64 19/42
  Upgrading                                : libmount-2.37.4-1.amzn2023.0.4.x86_64 20/42
  Upgrading                                : libsmartcols-2.37.4-1.amzn2023.0.4.x86_64 21/42
  Cleanup                                  : yum-4.14.0-1.amzn2023.0.4.noarch 22/42
  Running scriptlet: dnf-4.14.0-1.amzn2023.0.4.noarch 23/42
  Cleanup                                  : dnf-4.14.0-1.amzn2023.0.4.noarch 23/42
  Running scriptlet: dnf-4.14.0-1.amzn2023.0.4.noarch 23/42
  Cleanup                                  : python3-dnf-4.14.0-1.amzn2023.0.4.noarch 24/42
  Cleanup                                  : amazon-linux-repo-cdn-2023.4.20240319-1.amzn2023.no 25/42
  Cleanup                                  : libmount-2.37.4-1.amzn2023.0.3.x86_64 26/42
  Cleanup                                  : curl-minimal-8.5.0-1.amzn2023.0.2.x86_64 27/42
  Cleanup                                  : libcurl-minimal-8.5.0-1.amzn2023.0.2.x86_64 28/42

```

Cleanup	: krb5-libs-1.21-3.amzn2023.0.3.x86_64	29/42
Cleanup	: libblkid-2.37.4-1.amzn2023.0.3.x86_64	30/42
Cleanup	: libnghttp2-1.57.0-1.amzn2023.0.1.x86_64	31/42
Cleanup	: libsmartcols-2.37.4-1.amzn2023.0.3.x86_64	32/42
Cleanup	: system-release-2023.4.20240319-1.amzn2023.noarch	33/42
Cleanup	: dnf-data-4.14.0-1.amzn2023.0.4.noarch	34/42
Cleanup	: python3-3.9.16-1.amzn2023.0.6.x86_64	35/42
Cleanup	: python3-libs-3.9.16-1.amzn2023.0.6.x86_64	36/42
Cleanup	: openssl-libs-1:3.0.8-1.amzn2023.0.11.x86_64	37/42
Cleanup	: libuuid-2.37.4-1.amzn2023.0.3.x86_64	38/42
Cleanup	: expat-2.5.0-1.amzn2023.0.3.x86_64	39/42
Cleanup	: glibc-2.34-52.amzn2023.0.8.x86_64	40/42
Cleanup	: glibc-minimal-langpack-2.34-52.amzn2023.0.8.x86_64	41/42
Cleanup	: glibc-common-2.34-52.amzn2023.0.8.x86_64	42/42
Running scriptlet:	glibc-common-2.34-52.amzn2023.0.8.x86_64	42/42
Verifying	: amazon-linux-repo-cdn-2023.5.20240708-1.amzn2023.no	1/42
Verifying	: amazon-linux-repo-cdn-2023.4.20240319-1.amzn2023.no	2/42
Verifying	: curl-minimal-8.5.0-1.amzn2023.0.4.x86_64	3/42
Verifying	: curl-minimal-8.5.0-1.amzn2023.0.2.x86_64	4/42
Verifying	: dnf-4.14.0-1.amzn2023.0.5.noarch	5/42
Verifying	: dnf-4.14.0-1.amzn2023.0.4.noarch	6/42
Verifying	: dnf-data-4.14.0-1.amzn2023.0.5.noarch	7/42
Verifying	: dnf-data-4.14.0-1.amzn2023.0.4.noarch	8/42
Verifying	: expat-2.5.0-1.amzn2023.0.4.x86_64	9/42
Verifying	: expat-2.5.0-1.amzn2023.0.3.x86_64	10/42
Verifying	: glibc-2.34-52.amzn2023.0.10.x86_64	11/42
Verifying	: glibc-2.34-52.amzn2023.0.8.x86_64	12/42
Verifying	: glibc-common-2.34-52.amzn2023.0.10.x86_64	13/42
Verifying	: glibc-common-2.34-52.amzn2023.0.8.x86_64	14/42
Verifying	: glibc-minimal-langpack-2.34-52.amzn2023.0.10.x86_64	15/42
Verifying	: glibc-minimal-langpack-2.34-52.amzn2023.0.8.x86_64	16/42
Verifying	: krb5-libs-1.21-3.amzn2023.0.4.x86_64	17/42
Verifying	: krb5-libs-1.21-3.amzn2023.0.3.x86_64	18/42
Verifying	: libblkid-2.37.4-1.amzn2023.0.4.x86_64	19/42
Verifying	: libblkid-2.37.4-1.amzn2023.0.3.x86_64	20/42
Verifying	: libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64	21/42
Verifying	: libcurl-minimal-8.5.0-1.amzn2023.0.2.x86_64	22/42
Verifying	: libmount-2.37.4-1.amzn2023.0.4.x86_64	23/42
Verifying	: libmount-2.37.4-1.amzn2023.0.3.x86_64	24/42
Verifying	: libnghttp2-1.59.0-3.amzn2023.0.1.x86_64	25/42
Verifying	: libnghttp2-1.57.0-1.amzn2023.0.1.x86_64	26/42
Verifying	: libsmartcols-2.37.4-1.amzn2023.0.4.x86_64	27/42
Verifying	: libsmartcols-2.37.4-1.amzn2023.0.3.x86_64	28/42
Verifying	: libuuid-2.37.4-1.amzn2023.0.4.x86_64	29/42

```

Verifying      : libuuid-2.37.4-1.amzn2023.0.3.x86_64      30/42
Verifying      : openssl-libs-1:3.0.8-1.amzn2023.0.12.x86_64 31/42
Verifying      : openssl-libs-1:3.0.8-1.amzn2023.0.11.x86_64 32/42
Verifying      : python3-3.9.16-1.amzn2023.0.8.x86_64     33/42
Verifying      : python3-3.9.16-1.amzn2023.0.6.x86_64     34/42
Verifying      : python3-dnf-4.14.0-1.amzn2023.0.5.noarch  35/42
Verifying      : python3-dnf-4.14.0-1.amzn2023.0.4.noarch  36/42
Verifying      : python3-libs-3.9.16-1.amzn2023.0.8.x86_64 37/42
Verifying      : python3-libs-3.9.16-1.amzn2023.0.6.x86_64 38/42
Verifying      : system-release-2023.5.20240708-1.amzn2023.noarch 39/42
Verifying      : system-release-2023.4.20240319-1.amzn2023.noarch 40/42
Verifying      : yum-4.14.0-1.amzn2023.0.5.noarch          41/42
Verifying      : yum-4.14.0-1.amzn2023.0.4.noarch          42/42

```

Upgraded:

```

amazon-linux-repo-cdn-2023.5.20240708-1.amzn2023.noarch
curl-minimal-8.5.0-1.amzn2023.0.4.x86_64
dnf-4.14.0-1.amzn2023.0.5.noarch
dnf-data-4.14.0-1.amzn2023.0.5.noarch
expat-2.5.0-1.amzn2023.0.4.x86_64
glibc-2.34-52.amzn2023.0.10.x86_64
glibc-common-2.34-52.amzn2023.0.10.x86_64
glibc-minimal-langpack-2.34-52.amzn2023.0.10.x86_64
krb5-libs-1.21-3.amzn2023.0.4.x86_64
libblkid-2.37.4-1.amzn2023.0.4.x86_64
libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64
libmount-2.37.4-1.amzn2023.0.4.x86_64
libnghttp2-1.59.0-3.amzn2023.0.1.x86_64
libsmartcols-2.37.4-1.amzn2023.0.4.x86_64
libuuid-2.37.4-1.amzn2023.0.4.x86_64
openssl-libs-1:3.0.8-1.amzn2023.0.12.x86_64
python3-3.9.16-1.amzn2023.0.8.x86_64
python3-dnf-4.14.0-1.amzn2023.0.5.noarch
python3-libs-3.9.16-1.amzn2023.0.8.x86_64
system-release-2023.5.20240708-1.amzn2023.noarch
yum-4.14.0-1.amzn2023.0.5.noarch

```

Complete!

bash-5.2#

Para descobrir AL2023 atualizações, faça um ou mais dos seguintes:

- Execute o comando `dnf check-update`. Isso verifica se há atualizações não aplicadas na versão do Amazon Linux com a qual você tem vínculo. Isso pode mostrar atualizações se você atualizou apenas o pacote `system-release`, movendo a versão dos repositórios à qual a instância está vinculada, mas sem aplicar nenhuma das atualizações disponíveis nela.
- Inscreva-se no tópico SNS de atualização do repositório Amazon Linux (`arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates`). Para obter instruções, consulte [Assinatura de um tópico do Amazon SNS](#) no Guia do desenvolvedor do Amazon Simple Notification Service.
- Consulte regularmente as [notas AL2023 de lançamento](#).
- Descubra novas versões ao [Verificar versões mais recentes do repositório com `dnf check-release-update`](#).

 Important

Novas versões das atualizações de segurança AL2023 que contêm são lançadas com frequência. Fique por dentro dos patches de segurança relevantes.

Reinício automático do serviço após atualizações (de segurança)

O Amazon Linux agora é distribuído com o pacote [smart-restart](#). `Smart-restart` reinicia os serviços de `systemd` nas atualizações do sistema sempre que um pacote é instalado ou excluído usando o gerenciador de pacotes do sistema. Isso ocorre sempre que `dnf (update|upgrade|downgrade)` é executado.

`Smart-restart` usa o pacote `needs-restarting` de `dnf-utils` e um mecanismo personalizado de lista de negações para determinar quais serviços precisam ser reiniciados e se a reinicialização do sistema é recomendada. Se uma reinicialização do sistema for recomendada, um arquivo marcador de dica de reinicialização será gerado (`/run/smart-restart/reboot-hint-marker`).

Para instalar o **smart-restart**

Execute o comando de DNF a seguir (da mesma forma que você executaria em outros pacotes).

```
$ sudo dnf install smart-restart
```

Após a instalação, as transações subsequentes acionarão a lógica `smart-restart`.

Lista de negações

`Smart-restart` pode ser instruído a impedir que determinados serviços sejam reiniciados. Os serviços bloqueados não contribuirão para a decisão de necessidade de reinicialização. Para bloquear serviços adicionais, adicione um arquivo com o sufixo `-denylist` em `/etc/smart-restart-conf.d/`, conforme mostrado no exemplo a seguir.

```
$ cat /etc/smart-restart-conf.d/custom-denylist
# Some comments
myservice.service
```

Note

Todos os arquivos `*-denylist` são lidos e avaliados ao decidir se uma reinicialização é necessária.

Hooks personalizados

Além da lista de negações, `smart-restart` fornece um mecanismo para executar scripts personalizados antes e depois das tentativas de reiniciar o serviço. Os scripts personalizados podem ser usados para executar manualmente as etapas de preparação ou para informar outros componentes sobre uma reinicialização pendente ou concluída.

Todos os scripts em `/etc/smart-restart-conf.d/` com o sufixo `-pre-restart` ou `-post-restart` são executados. Se a ordem for importante, coloque um número antes de todos os scripts para garantir a ordem de execução, conforme mostrado no exemplo a seguir.

```
$ ls /etc/smart-restart-conf.d/*-pre-restart
001-my-script-pre-restart
002-some-other-script-pre-restart
```

Quando é necessário reinicializar para aplicar as atualizações de segurança?

Em algumas situações, o Amazon Linux exige uma reinicialização para aplicar as atualizações:

- As atualizações do pacote de kernel do Linux exigem uma reinicialização para ativar o novo kernel com as atualizações de segurança mais recentes. A aplicação de patches do kernel em tempo real pode permitir que você adie as atualizações de segurança por um período limitado. Para obter detalhes, consulte [Aplicação de patches do kernel em tempo real no AL2023](#).
- Nas instâncias EC2 Metal, o Amazon Linux fornece atualizações de microcódigo (por meio do `microcode_ctl` pacote para Intel CPUs e do `amd-ucode-firmware` pacote para CPUs AMD). Essas atualizações de microcódigo só serão ativadas nas reinicializações subsequentes da instância. Para instâncias do EC2 virtualizadas, o [AWS Nitro System](#) lida com as atualizações de microcódigo para você.
- Alguns serviços de `systemd` em execução só funcionarão corretamente após a reinicialização completa do sistema. O mecanismo `smart-restart` informará você sobre essas situações apresentando dicas de reinicialização. Consulte [Reinício automático do serviço após atualizações \(de segurança\)](#).

Lançamento de uma instância com a versão mais recente do repositório ativada

Você pode adicionar comandos DNF a um script de dados do usuário para controlar quais pacotes RPM são instalados em um Amazon Linux AMI quando ele é lançado. No exemplo a seguir, um script de dados de usuário é usado para garantir que qualquer instância iniciada com o script de dados de usuário tenha as mesmas atualizações de pacote instaladas.

```
#!/bin/bash
dnf upgrade --releasever=2023.0.20230210
# Additional setup and install commands below
dnf install httpd php7.4 mysql80
```

Você deve executar esse script como superusuário (raiz). Para fazer isso, execute o comando a seguir.

```
$ sudo sh -c "bash nameofscript.sh"
```

Para saber mais, consulte [Dados de usuário e scripts de shell](#) no Guia do usuário do Amazon EC2.

 Note

Em vez de usar um script de dados do usuário, inicie a Amazon Linux AMI mais recente ou uma AMI personalizada baseada na Amazon Linux AMI. O Amazon Linux AMI mais recente tem todas as atualizações necessárias instaladas e está configurado para apontar para uma versão específica do repositório.

Obtendo informações de suporte do pacote

AL2023 incorpora muitos projetos diferentes de software de código aberto. Cada um desses projetos é gerenciado de forma independente do Amazon Linux e tem end-of-support lançamentos e cronogramas diferentes. Para fornecer informações específicas do Amazon Linux sobre esses diferentes pacotes, o plug-in `dnf supportinfo` fornece metadados sobre um pacote. No exemplo a seguir, o comando **`dnf supportinfo`** retorna metadados para o pacote `glibc`.

```
$ sudo dnf supportinfo --pkg glibc
Last metadata expiration check: 0:07:56 ago on Wed Mar 1 23:21:49 2023.
Name                               : glibc
Version                           : 2.34-52.amzn2023.0.2
State                             : installed
Support Status                    : supported
Support Periods                   : from 2023-03-15      : supported
                                   : from 2028-03-15      : unsupported
Support Statement                 : Amazon Linux 2023 End Of Life
Link                             : https://aws.amazon.com/amazon-linux-ami/faqs/
Other Info                        : This is the support statement for AL2023. The
    ...: end of life of Amazon Linux 2023 would be March 2028.
    ...: From this point, the Amazon Linux 2023 packages (listed
    ...: below) will no longer, receive any updates from AWS.
```

As informações de suporte do Package também estão disponíveis na seção de [declarações de suporte](#) das [Notas de AL2023 Lançamento](#).

Verificar versões mais recentes do repositório com **`dnf check-release-update`**

Em uma AL2023 instância, você pode usar o DNF utilitário para gerenciar repositórios e aplicar RPM pacotes atualizados. Esses pacotes estão disponíveis nos repositórios do Amazon Linux. Você pode

usar o comando `dnf check-release-update` do DNF para verificar novas versões do repositório DNF.

Note

AL2023 Por padrão, as imagens de contêiner não incluem o `dnf check-release-update` comando.

```
$ dnf check-release-update
```

```
No such command: check-release-update. Please use /usr/bin/dnf --help  
It could be a DNF plugin command, try: "dnf install 'dnf-command(check-release-  
update)'"
```

Quando `dnf install 'dnf-command(check-release-update)'` for executado, `dnf` instalará o pacote que fornece o comando `check-release-update`, que é o pacote `dnf-plugin-release-notification`. No exemplo abaixo, o argumento `-q` é fornecido a `dnf` para que ele tenha uma saída silenciosa.

```
$ dnf -y -q install 'dnf-command(check-release-update)'
```

```
Installed:  
dnf-plugin-release-notification-1.2-1.amzn2023.0.2.noarch
```

Em ambientes sem contêineres, como uma instância do Amazon EC2, o comando `check-release-update` é incluído por padrão.

```
$ sudo dnf check-release-update
```

```
WARNING:
```

```
A newer release of "Amazon Linux" is available.
```

```
Available Versions:
```

```
Version 2023.0.20230210:
```

```
Run the following command to update to 2023.0.20230210:
```

```
dnf upgrade --releasever=2023.0.20230210
```

```
Release notes:
```

```
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes.html
```

Isso retorna uma lista completa de todas as versões mais recentes dos repositórios DNF que estão disponíveis. Se nada for retornado, isso significa que DNF está atualmente configurado para usar a versão mais recente disponível. A versão do pacote `system-release` instalado atualmente define a variável `releasever` DNF. Para verificar a versão atual do repositório, execute o comando a seguir.

```
$ rpm -q system-release --qf "%{VERSION}\n"
```

Quando você executa transações de pacotes DNF (como comandos de instalação, atualização ou remoção), uma mensagem de aviso o notifica sobre qualquer nova versão do repositório. Por exemplo, se você instalar o `httpd` pacote em uma instância que foi executada a partir de uma versão mais antiga do AL2023, a saída a seguir será retornada.

```
$ sudo dnf install httpd -y
Last metadata expiration check: 0:16:52 ago on Wed Mar 1 23:21:49 2023.
Dependencies resolved.
=====
Package                Arch    Version                               Repository    Size
=====
Installing:
httpd                   x86_64  2.4.54-3.amzn2023.0.4               amazonlinux   46 k
Installing dependencies:
apr                     x86_64  1.7.2-2.amzn2023.0.2               amazonlinux   129 k
apr-util                x86_64  1.6.3-1.amzn2023.0.1               amazonlinux   98 k
generic-logos-httpd
httpd-core              x86_64  2.4.54-3.amzn2023.0.4               amazonlinux   1.3 M
httpd-filesystem        noarch  2.4.54-3.amzn2023.0.4               amazonlinux   13 k
httpd-tools             x86_64  2.4.54-3.amzn2023.0.4               amazonlinux   80 k
libbrotli               x86_64  1.0.9-4.amzn2023.0.2               amazonlinux   315 k
mailcap                 noarch  2.1.49-3.amzn2023.0.3               amazonlinux   33 k
Installing weak dependencies:
apr-util-openssl        x86_64  1.6.3-1.amzn2023.0.1               amazonlinux   17 k
mod_http2               x86_64  1.15.24-1.amzn2023.0.3             amazonlinux   152 k
mod_lua                 x86_64  2.4.54-3.amzn2023.0.4               amazonlinux   60 k

Transaction Summary
=====
Install 12 Packages

Total download size: 2.3 M
Installed size: 6.8 M
Downloading Packages:
```

```
(1/12): apr-util-openssl-1.6.3-1.am 212 kB/s | 17 kB      00:00
(2/12): apr-1.7.2-2.amzn2023.0.2.x8 1.1 MB/s | 129 kB     00:00
(3/12): httpd-core-2.4.54-3.amzn202 8.9 MB/s | 1.3 MB      00:00
(4/12): mod_http2-1.15.24-1.amzn202 1.9 MB/s | 152 kB     00:00
(5/12): apr-util-1.6.3-1.amzn2023.0 1.7 MB/s | 98 kB      00:00
(6/12): mod_lua-2.4.54-3.amzn2023.0 1.4 MB/s | 60 kB      00:00
(7/12): httpd-2.4.54-3.amzn2023.0.4 1.5 MB/s | 46 kB      00:00
(8/12): libbrotli-1.0.9-4.amzn2023. 4.4 MB/s | 315 kB     00:00
(9/12): mailcap-2.1.49-3.amzn2023.0 753 kB/s | 33 kB      00:00
(10/12): httpd-tools-2.4.54-3.amzn2 978 kB/s | 80 kB      00:00
(11/12): httpd-filesystem-2.4.54-3. 210 kB/s | 13 kB      00:00
(12/12): generic-logos-httpd-18.0.0 439 kB/s | 19 kB      00:00
```

```
-----
Total                               6.6 MB/s | 2.3 MB      00:00
```

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

```
Preparing      :                               1/1
Installing     : apr-1.7.2-2.amzn2023.0.2.x86_64 1/12
Installing     : apr-util-openssl-1.6.3-1.amzn2023.0.1. 2/12
Installing     : apr-util-1.6.3-1.amzn2023.0.1.x86_64 3/12
Installing     : mailcap-2.1.49-3.amzn2023.0.3.noarch 4/12
Installing     : httpd-tools-2.4.54-3.amzn2023.0.4.x86_ 5/12
Installing     : generic-logos-httpd-18.0.0-12.amzn2023 6/12
Running scriptlet: httpd-filesystem-2.4.54-3.amzn2023.0.4 7/12
Installing     : httpd-filesystem-2.4.54-3.amzn2023.0.4 7/12
Installing     : httpd-core-2.4.54-3.amzn2023.0.4.x86_6 8/12
Installing     : mod_http2-1.15.24-1.amzn2023.0.3.x86_6 9/12
Installing     : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 10/12
Installing     : mod_lua-2.4.54-3.amzn2023.0.4.x86_64 11/12
Installing     : httpd-2.4.54-3.amzn2023.0.4.x86_64 12/12
Running scriptlet: httpd-2.4.54-3.amzn2023.0.4.x86_64 12/12
Verifying      : apr-1.7.2-2.amzn2023.0.2.x86_64 1/12
Verifying      : apr-util-openssl-1.6.3-1.amzn2023.0.1. 2/12
Verifying      : httpd-core-2.4.54-3.amzn2023.0.4.x86_6 3/12
Verifying      : mod_http2-1.15.24-1.amzn2023.0.3.x86_6 4/12
Verifying      : apr-util-1.6.3-1.amzn2023.0.1.x86_64 5/12
Verifying      : mod_lua-2.4.54-3.amzn2023.0.4.x86_64 6/12
Verifying      : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 7/12
Verifying      : httpd-2.4.54-3.amzn2023.0.4.x86_64 8/12
Verifying      : httpd-tools-2.4.54-3.amzn2023.0.4.x86_ 9/12
Verifying      : mailcap-2.1.49-3.amzn2023.0.3.noarch 10/12
```

```
Verifying      : httpd-filesystem-2.4.54-3.amzn2023.0.4 11/12
Verifying      : generic-logos-httpd-18.0.0-12.amzn2023 12/12
```

Installed:

```
apr-1.7.2-2.amzn2023.0.2.x86_64
apr-util-1.6.3-1.amzn2023.0.1.x86_64
apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
httpd-2.4.54-3.amzn2023.0.4.x86_64
httpd-core-2.4.54-3.amzn2023.0.4.x86_64
httpd-filesystem-2.4.54-3.amzn2023.0.4.noarch
httpd-tools-2.4.54-3.amzn2023.0.4.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_http2-1.15.24-1.amzn2023.0.3.x86_64
mod_lua-2.4.54-3.amzn2023.0.4.x86_64
```

Complete!

Adicionar, habilitar ou desabilitar novos repositórios

Warning

Adicione somente repositórios projetados para serem usados com AL2023. Embora os repositórios projetados para outras distribuições possam funcionar atualmente, não há garantia de que continuarão funcionando com qualquer atualização de pacote AL2023 ou com o repositório não projetado para uso. AL2023

Para instalar um pacote de um repositório diferente dos repositórios padrão do Amazon Linux, você precisará configurar o sistema de gerenciamento de pacotes DNF para saber onde está o repositório.

Para informar ao dnf sobre um repositório de pacotes, adicione as informações do repositório a um arquivo de configuração para esse repositório no diretório `/etc/yum.repos.d/`. Muitos repositórios de terceiros fornecem o conteúdo do arquivo de configuração ou um pacote instalável que inclui o arquivo de configuração.

Note

Embora os repositórios possam ser configurados diretamente no arquivo `/etc/dnf/dnf.conf`, isso não é recomendado. É recomendável que cada repositório seja configurado em seu próprio arquivo em `/etc/yum.repos.d/`.

Para descobrir quais repositórios estão atualmente habilitados, você pode executar o seguinte comando:

```
$ dnf repolist all --verbose
```

```
Loaded plugins: builddep, changelog, config-manager, copr, debug, debuginfo-install,
download, generate_completion_cache, groups-manager, needs-restarting, playground,
release-notification, repoclosure, repodiff, repograph, repomanage, reposync,
supportinfo
```

```
DNF version: 4.12.0
```

```
cachedir: /var/cache/dnf
```

```
Last metadata expiration check: 0:00:02 ago on Wed Mar 1 23:40:15 2023.
```

```
Repo-id           : amazonlinux
Repo-name         : Amazon Linux 2023 repository
Repo-status      : enabled
Repo-revision     : 1677203368
Repo-updated      : Fri Feb 24 01:49:28 2023
Repo-pkgs        : 12632
Repo-available-pkgs: 12632
Repo-size        : 12 G
Repo-mirrors      : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/x86_64/mirror.list
Repo-baseurl      : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/guids/
cf9296325a6c46ff40c775a8e2d632c4c3fd9d9164014ce3304715d61b90ca8e/x86_64/
                  : (0 more)
Repo-expire       : 172800 second(s) (last: Wed Mar 1 23:40:15
                  : 2023)
Repo-filename     : /etc/yum.repos.d/amazonlinux.repo
```

```
Repo-id           : amazonlinux-debuginfo
Repo-name         : Amazon Linux 2023 repository - Debug
Repo-status      : disabled
Repo-mirrors      : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/debuginfo/x86_64/mirror.list
Repo-expire       : 21600 second(s) (last: unknown)
```

```
Repo-filename      : /etc/yum.repos.d/amazonlinux.repo

Repo-id            : amazonlinux-source
Repo-name          : Amazon Linux 2023 repository - Source packages
Repo-status        : disabled
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/SRPMS/mirror.list
Repo-expire        : 21600 second(s) (last: unknown)
Repo-filename      : /etc/yum.repos.d/amazonlinux.repo

Repo-id            : kernel-livepatch
Repo-name          : Amazon Linux 2023 Kernel Livepatch repository
Repo-status        : disabled
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/kernel-livepatch/mirrors/al2023/x86_64/mirror.list
Repo-expire        : 172800 second(s) (last: unknown)
Repo-filename      : /etc/yum.repos.d/kernel-livepatch.repo

Repo-id            : kernel-livepatch-source
Repo-name          : Amazon Linux 2023 Kernel Livepatch repository -
                   : Source packages
Repo-status        : disabled
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/kernel-livepatch/mirrors/al2023/SRPMS/mirror.list
Repo-expire        : 21600 second(s) (last: unknown)
Repo-filename      : /etc/yum.repos.d/kernel-livepatch.repo
Total packages: 12632
```

Note

Se você não adicionar o sinalizador de opção `--verbose`, a saída incluirá somente as informações de `Repo-id`, `Repo-name` e `Repo-status`.

Para adicionar um repositório **yum** ao diretório **/etc/yum.repos.d**:

1. Encontre a localização do arquivo `.repo`. Neste exemplo, o arquivo `.repo` está em <https://www.example.com/repository.repo>.
2. Adicione um repositório com o comando `dnf config-manager`.

```
$ sudo dnf config-manager --add-repo https://www.example.com/repository.repo
```

```
Loaded plugins: priorities, update-motd, upgrade-helper
adding repo from: https://www.example.com/repository.repo
grabbing file https://www.example.com/repository.repo to /etc/
yum.repos.d/repository.repo
repository.repo | 4.0 kB 00:00
repo saved to /etc/yum.repos.d/repository.repo
```

Após instalar um repositório, é necessário habilitá-lo como descrito no próximo procedimento.

Para habilitar um yum repositório/etc/yum.repos.d, use o `dnf config-manager` comando com a `--enable` bandeira e o *repository* nome.

```
$ sudo dnf config-manager --enable repository
```

Note

Para desativar um repositório, use a mesma sintaxe de comando, mas substitua `--enable` por `--disable` no comando.

Adicionando repositórios com cloud-init

Além de adicionar um repositório usando o método anterior, você também pode adicionar um novo repositório usando a estrutura de `cloud-init`.

Para adicionar um novo repositório de pacotes, recomendamos o uso do modelo a seguir. Considere salvar esse arquivo localmente.

```
#cloud-config
yum_repos:
  repository.repo:
    baseurl: https://www.example.com/
    enabled: true
    gpgcheck: true
    gpgkey: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EXAMPLE
    name: Example Repository
```

 Note

Uma vantagem de usar `cloud-init` é que você pode adicionar uma seção de `packages` ao seu arquivo de configuração. Nesta seção, você pode incluir os nomes dos pacotes que você deseja instalar. Você pode instalar pacotes do repositório padrão ou do novo repositório que você adicionou ao arquivo `cloud-config`.

Para obter informações mais específicas sobre a estrutura do arquivo YAML, consulte [Adicionar um repositório YUM](#) na Documentação do `cloud-init`.

Depois de configurar o arquivo no formato YAML, você pode executá-lo na estrutura do `cloud-init` na AWS CLI. Certifique-se de incluir a opção `--userdata` e o nome do arquivo `.yaml` para chamar as operações desejadas.

```
$ aws ec2 run-instances \
  --image-id \
    resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64 \
  --instance-type m5.xlarge \
  --region us-east-1 \
  --key-name aws-key-us-east-1 \
  --security-group-ids sg-004a7650 \
  --user-data file://cloud-config.yaml
```

Aplicação de patches do kernel em tempo real no AL2023

Você pode usar a aplicação de patches do kernel em tempo real para o AL2023 a fim de aplicar patches específicos para vulnerabilidades de segurança e bugs críticos a um kernel do Linux em execução, sem reiniciar ou interromper as aplicações que estão em execução. Além disso, a aplicação de patches do kernel em tempo real pode ajudar a melhorar a disponibilidade da aplicação, ao mesmo tempo em que aplica essas correções até que o sistema possa ser reinicializado.

A AWS disponibiliza dois tipos de patches do kernel em tempo real para o AL2023:

- Security updates (Atualizações de segurança): contêm atualizações para vulnerabilidades e exposições comuns (CVEs) do Linux. Normalmente, essas atualizações são classificadas como importantes ou críticas de acordo com as classificações do Boletim de segurança do Amazon Linux. Geralmente, elas são mapeadas com uma pontuação 7 ou maior do Common Vulnerability Scoring System (CVSS – Sistema de pontuação de vulnerabilidades comuns). Em alguns casos,

a AWS pode fornecer atualizações antes da atribuição de CVE. Nesses casos, os patches podem aparecer como correções de erros.

- Correções de erros: contêm correções de erros críticos e problemas de estabilidade que não estão associados às CVEs.

A AWS oferece patches do kernel em tempo real para uma versão do kernel do AL2023 por até 3 meses depois de seu lançamento. Após esse período, é necessário fazer a atualização para uma versão posterior do kernel para continuar a receber patches ao vivo do kernel.

Os patches ao vivo do kernel para o AL2023 são disponibilizados como pacotes RPM assinados nos repositórios existentes do AL2023. Os patches podem ser instalados em instâncias individuais usando fluxos de trabalho de gerenciamento de pacotes DNF existentes. Ou podem ser instalados em um grupo de instâncias gerenciadas usando o AWS Systems Manager.

O Kernel Live Patching no AL2023 é fornecido sem custo adicional.

Tópicos

- [Limitações](#)
- [Configurações e pré-requisitos compatíveis](#)
- [Trabalhar com o Kernel Live Patching](#)

Limitações

Ao aplicar um patch ao vivo no kernel, você não pode executar a hibernação, usar ferramentas avançadas de depuração (como SystemTap, kprobes e ferramentas baseadas em eBPF) ou acessar arquivos de saída `fttrace` usados pela infraestrutura do Kernel Live Patching.

Note

Devido a limitações técnicas, alguns problemas não podem ser resolvidos com a aplicação de patches em tempo real. Por causa disso, essas correções não serão distribuídas no pacote de patch do kernel em tempo real, mas somente na atualização do pacote nativo do kernel. Você pode instalar o pacote nativo do kernel e [atualizar e reinicializar](#) o sistema para ativar os patches normalmente.

Configurações e pré-requisitos compatíveis

O Kernel Live Patching pode ser usado nas instâncias do Amazon EC2 e nas máquinas virtuais locais que executam o AL2023.

Para usar o Kernel Live Patching no AL2023, você deve usar o seguinte:

- Um x86_64 de 64 bits ou arquitetura ARM64
- Versão 6.1 ou 6.12 do kernel

Requisitos de política

Para baixar pacotes dos repositórios do AL2023, o Amazon EC2 precisa acessar os buckets do Amazon S3 pertencentes ao serviço. Se estiver usando um endpoint do Amazon Virtual Private Cloud (VPC) para o Amazon S3 em seu ambiente, garanta que sua política de endpoint da VPC permita acesso a esses buckets públicos. A tabela a seguir descreve o bucket do Amazon S3 que o Amazon EC2 pode precisar acessar para a aplicação de patches do kernel em tempo real.

ARN do bucket do S3	Descrição
arn:aws:s3:::al2023-repos- <i>region</i> -de612dc2/*	Bucket do Amazon S3 contendo repositórios do AL2023

Trabalhar com o Kernel Live Patching

Você pode habilitar e usar o Kernel Live Patching em instâncias individuais usando a linha de comando na própria instância. Como alternativa, você pode habilitar e usar o Kernel Live Patching em um grupo de instâncias gerenciadas usando o AWS Systems Manager.

As seções a seguir explicam como habilitar e usar o Kernel Live Patching em instâncias individuais usando a linha de comando.

Para obter mais informações sobre como habilitar e usar o Kernel Live Patching em um grupo de instâncias gerenciadas, consulte [Use Kernel Live Patching on AL2023 instances](#) no Guia do usuário do AWS Systems Manager.

Tópicos

- [Habilitar o Kernel Live Patching](#)
- [Visualizar os patches ao vivo do kernel disponíveis](#)
- [Aplicar patches ao vivo do kernel](#)
- [Visualizar os patches ao vivo do kernel aplicados](#)
- [Desabilitar o Kernel Live Patching](#)

Habilitar o Kernel Live Patching

O Kernel Live Patching está desabilitado por padrão no AL2023. Para usar patches ao vivo, você deve instalar o plug-in DNF para patches ao vivo do kernel e ativar a funcionalidade de patching ao vivo.

Como habilitar o Kernel Live Patching

1. Os patches ao vivo do kernel estão disponíveis para AL2023 com a versão 6.1 do kernel. Para verificar a versão do kernel, execute o comando a seguir.

```
$ sudo dnf list kernel
```

2. Instale o plug-in DNF para o Kernel Live Patching.

```
$ sudo dnf install -y kpatch-dnf
```

3. Habilite o plug-in DNF para o Kernel Live Patching.

```
$ sudo dnf kernel-livepatch -y auto
```

Este comando também instala a versão mais recente de RPM do patch ao vivo do kernel a partir dos repositórios configurados.

4. Para confirmar se o plug-in DNF para a aplicação de patches ao vivo no kernel foi instalado com êxito, execute o comando a seguir.

Quando você habilita o Kernel Live Patching, um RPM de patch ao vivo do kernel vazio é aplicado automaticamente. Se a aplicação de patches do kernel em tempo real tiver sido habilitada com êxito, este comando retornará uma lista que inclui o RPM de patch do kernel em tempo real vazio inicial (e outro RPM que configura o repositório DNF contendo os patches em tempo real).

```
$ sudo rpm -qa | grep kernel-livepatch
kernel-livepatch-repo-s3-2023.7.20250428-0.amzn2023.noarch
kernel-livepatch-6.1.134-150.224-1.0-0.amzn2023.x86_64
```

5. Instale o pacote kpatch.

```
$ sudo dnf install -y kpatch-runtime
```

6. Atualize o serviço kpatch, caso tenha sido instalado anteriormente.

```
$ sudo dnf upgrade kpatch-runtime
```

7. Inicie o serviço kpatch. Este serviço carrega todos os patches ao vivo do kernel durante ou após a inicialização.

```
$ sudo systemctl enable kpatch.service && sudo systemctl start kpatch.service
```

Visualizar os patches ao vivo do kernel disponíveis

Os alertas de segurança do Amazon Linux são publicados no Centro de segurança do Amazon Linux. Para obter mais informações sobre os alertas de segurança do AL2023, que incluem alertas para patches ao vivo do kernel, consulte o [Centro de segurança do Amazon Linux](#). Os patches ao vivo do kernel são prefixados com ALASLIVEPATCH. O Centro de segurança do Amazon Linux pode não listar patches ao vivo do kernel que resolvam erros.

Também é possível descobrir os patches ao vivo do kernel disponíveis para recomendações e CVEs usando a linha de comando.

Como listar todos os patches ao vivo do kernel disponíveis para recomendações

Use o seguinte comando.

```
$ sudo dnf updateinfo list
Last metadata expiration check: 1:06:23 ago on Mon 13 Feb 2023 09:28:19 PM UTC.
ALAS2LIVEPATCH-2021-123    important/Sec. kernel-
livepatch-6.1.12-17.42-1.0-4.amzn2023.x86_64
ALAS2LIVEPATCH-2022-124    important/Sec. kernel-
livepatch-6.1.12-17.42-1.0-3.amzn2023.x86_64
```

Como listar todos os patches ao vivo do kernel disponíveis para CVEs

Use o seguinte comando.

```
$ sudo dnf updateinfo list cves
Last metadata expiration check: 1:07:26 ago on Mon 13 Feb 2023 09:28:19 PM UTC.
CVE-2022-0123      important/Sec. kernel-livepatch-6.1.12-17.42-1.0-4.amzn2023.x86_64
CVE-2022-3210      important/Sec. kernel-livepatch-6.1.12-17.42-1.0-3.amzn2023.x86_64
```

Aplicar patches ao vivo do kernel

Aplique patches ao vivo do kernel usando o gerenciador de pacotes DNF da mesma maneira que você aplicaria atualizações regulares. O plug-in DNF para a aplicação de patches do kernel em tempo real gerencia os patches do kernel em tempo real que estão disponíveis para aplicação.

Tip

Recomendamos que você atualize seu kernel regularmente usando a aplicação de patches do kernel em tempo real para garantir que ele receba correções de segurança específicas importantes e críticas até que o sistema possa ser reinicializado. Verifique também se foram disponibilizadas correções adicionais para o pacote do kernel nativo que não podem ser implementadas como patches em tempo real e, nesses casos, [atualize e reinicie](#) o kernel.

É possível optar por aplicar um patch ao vivo do kernel específico, ou aplicar qualquer patch ao vivo do kernel disponível com suas atualizações de segurança regulares.

Como aplicar um patch ao vivo do kernel específico

1. Obtenha a versão do patch ao vivo do kernel usando um dos comandos descritos em [Visualizar os patches ao vivo do kernel disponíveis](#).
2. Aplique o patch ao vivo do kernel no kernel do AL2023.

```
$ sudo dnf install kernel-livepatch-kernel_version-package_version.amzn2023.x86_64
```

Por exemplo, o comando a seguir aplica um patch ao vivo do kernel para a versão 6.1.12-17.42 AL2023 do kernel .

```
$ sudo dnf install kernel-livepatch-6.1.12-17.42-1.0-4.amzn2023.x86_64
```

Como aplicar patches ao vivo do kernel disponíveis com as atualizações de segurança regulares

Use o seguinte comando.

```
$ sudo dnf upgrade --security
```

Omita a opção `--security` para incluir correções de erros.

Important

- A versão do kernel não é atualizada após a aplicação de patches ao vivo do kernel. A versão só é atualizada para a nova versão depois da reinicialização da instância.
- Um kernel AL2023 recebe patches de kernel ao vivo por 3 meses. Após esse período, nenhum novo patch ativo do kernel será lançado para essa versão do kernel.
- Para continuar a receber patches ao vivo do kernel após 3 meses, você deve reinicializar a instância para migrar para a nova versão do kernel. A instância continua recebendo patches ativos do kernel pelos próximos 3 meses após a atualização.
- Para verificar a janela de suporte para a versão do kernel, execute o seguinte comando.

```
$ sudo dnf kernel-livepatch support
```

```
The current version of the Linux kernel you are running will no longer receive  
live patches after 2025-07-22.
```

Visualizar os patches ao vivo do kernel aplicados

Como visualizar os patches ao vivo do kernel aplicados

Use o seguinte comando.

```
$ sudo kpatch list
```

```
Loaded patch modules:
```

```
livepatch_CVE_2022_36946 [enabled]
```

```
Installed patch modules:
```

```
livepatch_CVE_2022_36946 (6.1.57-29.131.amzn2023.x86_64)
```

```
livepatch_CVE_2022_36946 (6.1.57-30.131.amzn2023.x86_64)
```

O comando retornará uma lista dos patches ao vivo do kernel de atualização de segurança carregados e instalados. A seguir está um exemplo de saída.

 Note

Um único patch ao vivo do kernel pode incluir e instalar vários patches ao vivo.

Desabilitar o Kernel Live Patching

Se não precisar mais usar o Kernel Live Patching, é possível desabilitá-lo a qualquer momento.

- Desabilite o uso de livepatches:

1. Desabilite o plug-in:

```
$ sudo dnf kernel-livepatch manual
```

2. Desabilite o serviço kpatch:

```
$ sudo systemctl disable --now kpatch.service
```

- Remova totalmente as ferramentas livepatch:

1. Remover o plug-in de:

```
$ sudo dnf remove kpatch-dnf
```

2. Remover kpatch-runtime:

```
$ sudo dnf remove kpatch-runtime
```

3. Remova qualquer livepatches instalado:

```
$ sudo dnf remove kernel-livepatch\*
```

Atualizando o kernel Linux em AL2023

Tópicos

- [Versões do kernel Linux em AL2023](#)

- [Atualizando AL2023 para uma versão mais recente do kernel](#)
- [AL2023 kernels - Perguntas frequentes](#)

Versões do kernel Linux em AL2023

AL2023 inclui regularmente novas versões do kernel com base nas versões Long-Term Support (LTS) do kernel Linux.

AL2023 foi lançado originalmente em março de 2023 com o kernel 6.1.

Em abril de 2025, AL2023 adicionou suporte para o kernel Linux 6.12. Esse kernel adicionou novos recursos, incluindo agendamento EEVDF, I/O suporte à passagem do FUSE, uma nova API Futex e melhorias no eBPF. O kernel 6.12 também permite que um programa de espaço de usuário se proteja no runtime usando pilhas de sombras do espaço de usuário e vedação de memória.

Em março de 2026, AL2023 adicionou suporte para o kernel Linux 6.18. O kernel 6.18 atualizado traz melhorias adicionais no suporte, virtualização, segurança e desempenho do processador. Os recursos notáveis incluem recursos aprimorados de IOMMU em todas as arquiteturas e controles vetoriais de ataque para gerenciar as mitigações de vulnerabilidades da CPU. Os aprimoramentos de desempenho vêm por meio de otimizações de criptografia com operações mais rápidas do FSCRYPT, melhorias no gerenciamento de memória e a introdução do Sheaves como uma nova camada de cache opcional baseada em array por CPU.

Atualizando AL2023 para uma versão mais recente do kernel

A partir de junho de 2026, AL2023 atualizará o kernel padrão anualmente. O [al2023-ami-kernel-default](#) conjunto de AMIs será atualizado para o kernel LTS mais recente, para que as instâncias recém-lançadas apareçam automaticamente com a nova versão do kernel. Essa é a maneira mais simples de se manter atualizado com as últimas correções de segurança e melhorias de desempenho.

Se preferir escolher uma versão específica do kernel, você pode executar o AL2023 com o kernel 6.12 ou 6.18 selecionando uma AMI com o kernel desejado pré-instalado ou atualizando uma instância EC2 existente do AL2023.

Executando uma AL2023 AMI com uma versão específica do kernel

Você pode optar por executar uma AL2023 AMI com um kernel específico pré-instalado por meio do AWS Console ou consultando o SSM para obter parâmetros específicos. As chaves do SSM

para consulta começam com `/aws/service/ami-amazon-linux-latest/` seguido por um dos seguintes:

Para o kernel 6.12

- `al2023-ami-kernel-6.12-arm64` para a arquitetura arm64
- `al2023-ami-minimal-kernel-6.12-arm64` para arquitetura arm64 (AMI mínima)
- `al2023-ami-kernel-6.12-x86_64` para a arquitetura x86_64
- `al2023-ami-minimal-kernel-6.12-x86_64` para a arquitetura x86_64 (AMI mínima)

Para o kernel 6.18

- `al2023-ami-kernel-6.18-arm64` para a arquitetura arm64
- `al2023-ami-minimal-kernel-6.18-arm64` para arquitetura arm64 (AMI mínima)
- `al2023-ami-kernel-6.18-x86_64` para a arquitetura x86_64
- `al2023-ami-minimal-kernel-6.18-x86_64` para a arquitetura x86_64 (AMI mínima)

Consulte [Lançamento AL2023 usando o parâmetro SSM e AWS CLI](#) para obter detalhes sobre a seleção AL2023 AMIs.

Atualização de uma AL2023 instância para um kernel mais novo

Você pode atualizar localmente uma AL2023 instância em execução para o kernel 6.12 ou 6.18 com as seguintes etapas:

1. Detecte o kernel atual e defina a versão de destino:

```
# Automatically detect current kernel version BEFORE upgrade
$ CURRENT_KERNEL=$(uname -r)
$ SOURCE_VERSION=""

$ if [[ $CURRENT_KERNEL == *"6.12"* ]]; then
    SOURCE_VERSION="6.12"
else
    SOURCE_VERSION=""
fi

# Save the source version to a persistent location for use after reboot
```

```
$ echo "${SOURCE_VERSION}" | sudo tee /var/lib/source_kernel_version > /dev/null
```

```
# Set your target version (change this to your desired kernel: 6.12 or 6.18)
$ TARGET_VERSION="6.12"
```

```
$ echo "Current kernel: ${SOURCE_VERSION:-6.1}"
$ echo "Upgrading to kernel ${TARGET_VERSION}"
```

2. Instale o pacote do kernel de destino:

```
$ sudo dnf install -y kernel${TARGET_VERSION}
```

3. Obtenha a versão mais recente do pacote do kernel de destino:

```
$ version=$(rpm -q --qf '%{version}-%{release}.%{arch}\n' kernel${TARGET_VERSION} |
sort -V | tail -1)
```

4. Torne o novo kernel seu kernel padrão:

```
$ sudo grubby --set-default "/boot/vmlinuz-$version"
```

5. Reinicialize o sistema:

```
$ sudo reboot
```

6. Desinstale o kernel anterior:

```
# Read the source kernel version from the saved file
$ SOURCE_VERSION=$(sudo cat /var/lib/source_kernel_version)

# Uninstall the source kernel
$ sudo dnf remove -y kernel${SOURCE_VERSION}
```

7. Substitua pacotes extras de kernel por seus equivalentes de kernel de destino:

```
# Set your target version (change this to your desired kernel: 6.12 or 6.18)
$ TARGET_VERSION="6.12"
```

```
$ declare -A pkgs
$ pkgs=(
```

```

[bpftool${SOURCE_VERSION}]=bpftool${TARGET_VERSION}
[kernel${SOURCE_VERSION}-debuginfo]=kernel${TARGET_VERSION}-debuginfo
[kernel${SOURCE_VERSION}-debuginfo-common]=kernel${TARGET_VERSION}-debuginfo-common
[kernel${SOURCE_VERSION}-headers]=kernel${TARGET_VERSION}-headers
[kernel${SOURCE_VERSION}-libbpf]=kernel${TARGET_VERSION}-libbpf
[kernel${SOURCE_VERSION}-libbpf-devel]=kernel${TARGET_VERSION}-libbpf-devel
[kernel${SOURCE_VERSION}-libbpf-static]=kernel${TARGET_VERSION}-libbpf-static
[kernel${SOURCE_VERSION}-modules-extra-common]=kernel${TARGET_VERSION}-modules-extra-common
[kernel${SOURCE_VERSION}-tools]=kernel${TARGET_VERSION}-tools
[kernel${SOURCE_VERSION}-tools-devel]=kernel${TARGET_VERSION}-tools-devel
[perf${SOURCE_VERSION}]=perf${TARGET_VERSION}
[python3-perf${SOURCE_VERSION}]=python3-perf${TARGET_VERSION}
)
$ for pkg in "${!pkgs[@]}"; do
    rpm -q $pkg && sudo dnf -y swap $pkg "${pkgs["$pkg"]}" ;
done

```

8. (Opcional) Desinstale o kernel-devel da versão anterior do kernel:

```

$ rpm -q kernel${SOURCE_VERSION}-devel && sudo dnf remove -y kernel
${SOURCE_VERSION}-devel

```

Fazendo o downgrade para uma versão anterior do kernel

Se, em algum momento, você precisar fazer o downgrade para uma versão anterior do kernel, use as seguintes etapas:

1. Detecte o kernel atual e defina a versão de destino:

```

# Automatically detect current kernel version BEFORE downgrade
$ CURRENT_KERNEL=$(uname -r)
$ SOURCE_VERSION=""

$ if [[ $CURRENT_KERNEL == *"6.12"* ]]; then
    SOURCE_VERSION="6.12"
elif [[ $CURRENT_KERNEL == *"6.18"* ]]; then
    SOURCE_VERSION="6.18"
fi

# Save the source version to a persistent location for use after reboot

```

```
$ echo "${SOURCE_VERSION}" | sudo tee /var/lib/source_kernel_version > /dev/null
```

```
# Set your target version (change this to your desired kernel)
# Use "" for kernel 6.1, "6.12" for kernel 6.12
$ TARGET_VERSION=""
```

```
$ echo "Downgrading from kernel ${SOURCE_VERSION:-6.1} to kernel
${TARGET_VERSION:-6.1}"
```

2. Substitua pacotes extras de kernel por seus equivalentes de kernel de destino:

```
$ declare -A pkgs
$ pkgs=(
  [bpftool${TARGET_VERSION}]=bpftool${SOURCE_VERSION}
  [kernel${TARGET_VERSION}-debuginfo]=kernel${SOURCE_VERSION}-debuginfo
  [kernel${TARGET_VERSION}-debuginfo-common]=kernel${SOURCE_VERSION}-debuginfo-common
  [kernel${TARGET_VERSION}-headers]=kernel${SOURCE_VERSION}-headers
  [kernel${TARGET_VERSION}-libbpf]=kernel${SOURCE_VERSION}-libbpf
  [kernel${TARGET_VERSION}-libbpf-devel]=kernel${SOURCE_VERSION}-libbpf-devel
  [kernel${TARGET_VERSION}-libbpf-static]=kernel${SOURCE_VERSION}-libbpf-static
  [kernel${TARGET_VERSION}-modules-extra-common]=kernel${SOURCE_VERSION}-modules-
  extra-common
  [kernel${TARGET_VERSION}-tools]=kernel${SOURCE_VERSION}-tools
  [kernel${TARGET_VERSION}-tools-devel]=kernel${SOURCE_VERSION}-tools-devel
  [perf${TARGET_VERSION}]=perf${SOURCE_VERSION}
  [python3-perf${TARGET_VERSION}]=python3-perf${SOURCE_VERSION}
)
$ for pkg in "${!pkgs[@]}"; do
  rpm -q "${pkgs["$pkg"]}" && sudo dnf -y swap "${pkgs["$pkg"]}" $pkg ;
done
```

3. Instale o pacote do kernel de destino:

```
$ sudo dnf install -y kernel${TARGET_VERSION}
```

4. Obtenha a versão mais recente do pacote do kernel de destino:

```
$ version=$(rpm -q --qf '%{version}-%{release}.%{arch}\n' kernel${TARGET_VERSION} |
sort -V | tail -1)
```

5. Torne o kernel de destino seu kernel padrão:

```
$ sudo grubby --set-default "/boot/vmlinuz-$version"
```

6. Reinicialize o sistema:

```
$ sudo reboot
```

7. Desinstale o kernel de origem:

```
# Read the source kernel version from the saved file
$ SOURCE_VERSION=$(sudo cat /var/lib/source_kernel_version)

# Uninstall the source kernel
$ sudo dnf remove -y kernel${SOURCE_VERSION}
```

AL2023 kernels - Perguntas frequentes

1. Preciso reinicializar após uma atualização do kernel?

Toda alteração no kernel em execução requer uma reinicialização.

2. Como faço para manter os kernels up-to-date em várias instâncias?

O Amazon Linux não fornece recursos para gerenciar frotas de instâncias. Recomendamos que você aplique patches em grandes frotas usando ferramentas como o [AWS Systems Manager](#).

3. Como faço para verificar qual versão do kernel está em execução no momento?

Execute esse comando na sua AL2023 instância:

```
$ uname -r
```

4. Qual kernel me AL2023 recomenda usar?

É recomendável atualizar para o AL2023 kernel 6.18 mais recente enquanto todos os outros AL2023 kernels ainda são suportados. Recomenda-se que os clientes testem suas cargas de trabalho antes da atualização.

5. Meus aplicativos existentes funcionarão com qualquer AL2023 kernel?

AL2023 suporta um kernel mais novo (6.12 ou 6.18) da mesma forma que o kernel 6.1. Os aplicativos funcionarão e as melhorias estão acontecendo nos bastidores. Em qualquer caso, os clientes devem testar suas cargas de trabalho específicas antes de migrar para um kernel mais novo.

6. Como faço para instalar cabeçalhos do kernel, pacotes de desenvolvimento e módulos extras para o kernel 6.12 ou 6.18?

Execute o seguinte:

```
$ version=$(uname -r | grep -oP '^\\d+\\.\\d+')  
$ sudo dnf install -y kernel${version}-modules-extra-$(uname -r) kernel${version}-  
headers-$(uname -r) kernel${version}-devel-$(uname -r)
```

7. Por quanto tempo os kernels 6.12 e 6.18 serão suportados?

Os kernel 6.12 e 6.18 terão suporte até o final planejado da vida útil do Amazon Linux 2023, que é 30/06/2029.

Introdução aos tempos de execução de programação em AL2023

AL2023 fornece versões diferentes de alguns tempos de execução de linguagem. Trabalhamos com projetos upstream que oferecem suporte a várias versões ao mesmo tempo. Encontre informações sobre como instalar e gerenciar esses pacotes com versão por nome usando o comando `dnf` para pesquisar e instalar esses pacotes.

Os tópicos a seguir descrevem como cada ecossistema de linguagem existe em AL2023.

Tópicos

- [C, C++ e Fortran no AL2023](#)
- [Goem AL2023](#)
- [Java no AL2023](#)
- [Node.js em AL2 023](#)
- [Perl no AL2023](#)
- [PHPem AL2 023](#)
- [Python no AL2023](#)
- [Rubyem AL2023](#)
- [Rustem AL2023](#)
- [TypeScript no AL2023](#)

C, C++ e Fortran no AL2023

O AL2023 inclui a coleção de compiladores GNU (GCC) e o frontend Clang para LLVM (máquina virtual de baixo nível).

A versão principal do GCC permanecerá constante durante toda a vida útil do AL2023. Versões menores trazem correções de bugs e podem ser incluídas nas versões do AL2023. Outras correções de bugs, desempenho e segurança podem ser transferidas para a versão principal de GCC que vem no AL2023.

O AL2023 inclui a versão 11 de GCC como compilador padrão com os frontends C (gcc), C++ (g++) e Fortran (gfortran). Além disso, o AL2023 fornece GCC versão 14 como um compilador alternativo opcional que pode ser instalado com a versão padrão.

O AL2023 não habilita os frontends Ada (gnat), Go (gcc-go), Objective-C ou Objective-C++.

Os sinalizadores padrão do compilador com os quais os RPMs do AL2023 são criados incluem sinalizadores de otimização e fortalecimento. Para criar seu próprio código com GCC, recomendamos que você inclua sinalizadores de otimização e hardening.

Note

Quando `gcc --version` é invocado, uma string de versão como `gcc (GCC) 11.3.1 20221121 (Red Hat 11.3.1-4)` é exibida. Red Hat refere-se à [filial do fornecedor do GCC](#) na qual o pacote Amazon Linux GCC se baseia. De acordo com o URL de relatórios de bugs mostrado por `gcc --help`, todos os relatórios de bugs e solicitações de suporte devem ser direcionados para o Amazon Linux.

Para obter mais informações sobre algumas das mudanças de longo prazo nessa ramificação de fornecedor, como a macro `__GNUC_RH_RELEASE__`, consulte [Fedora package sources](#).

Para obter mais informações sobre a cadeia de ferramentas principal, consulte [Pacotes principais do conjunto de ferramentas glibc, gcc, binutils](#).

Para obter mais informações sobre o AL2023 e sua relação com outras distribuições do Linux, consulte [Relacionamento com o Fedora](#).

Para obter mais informações sobre a alteração do triplete do compilador no AL2023 em relação ao AL2, consulte [Compilador Triplet](#).

Tópicos

- [GCC 14](#)
- [Comparação entre versões padrão da linguagem](#)

GCC 14

O AL2023 fornece o GCC 14 como um compilador opcional que pode ser instalado com o GCC 11 padrão. O GCC 14 inclui os recursos e as otimizações de linguagem mais recentes, tornando-o adequado para projetos que exigem suporte aos padrões de C, C++ ou Fortran mais recentes.

Para instalar o GCC 14, use o seguinte comando:

```
sudo dnf install gcc14 gcc14-c++ gcc14-gfortran
```

Os compiladores GCC 14 são instalados com nomes de comando específicos da versão para evitar conflitos com o GCC 11 padrão:

- `gcc14-gcc`: compilador C
- `gcc14-g++`: compilador C++
- `gcc14-gfortran`: compilador Fortran

Exemplo de uso:

```
gcc14-gcc -o myprogram myprogram.c  
gcc14-g++ -o mycppprogram mycppprogram.cpp  
gcc14-gfortran -o myfortranprogram myfortranprogram.f90
```

Você pode verificar a versão instalada executando o seguinte:

```
gcc14-gcc --version
```

Isso exibirá informações de versão semelhantes a: `gcc14-gcc (GCC) 14.2.1 20250110 (Red Hat 14.2.1-7)`

Note

Tanto o GCC 11 quanto o GCC 14 podem ser instalados simultaneamente no mesmo sistema. Os comandos padrão `gcc`, `g++` e `gfortran` continuarão usando o GCC 11, enquanto o GCC 14 é acessado por meio dos comandos específicos da versão.

Comparação entre versões padrão da linguagem

A tabela a seguir compara as versões padrão da linguagem padrão em diferentes versões do Amazon Linux e versões do compilador GCC:

Versão Amazon Linux	Padrão C (padrão)	Padrão C++ (padrão)	Padrão Fortran
AL2 com GCC 7 (padrão)	C11 (201112L)	C++14 (201402L)	Fortran 2008
AL2 com GCC 10 (opcional)	C17/C18 (201710L)	C++14 (201402L)	Fortran 2008
AL2023 com GCC 11 (padrão)	C17/C18 (201710L)	C++17 (201703L)	Fortran 2008
AL2023 com GCC 14 (opcional)	C17/C18 (201710L)	C++17 (201703L)	Fortran 2008

Principais melhorias por versão do GCC:

- GCC 10 versus GCC 7: atualização do padrão C de C11 para C17/C18, adição de suporte aos recursos do C++20 e melhoria dos recursos de otimização.
- GCC 11 versus GCC 10: atualização do padrão C++ de C++14 para C++17, melhoria do suporte ao C++20 e adição de recursos experimentais do C++23.
- GCC 14 versus GCC 11: adição de suporte completo ao padrão C23, melhoria dos recursos do C++23, aprimoramento da otimização e melhoria da conformidade com os padrões.

Padrões de linguagem compatíveis:

- Padrões C: todas as versões são compatíveis com C90, C99, C11 e C17/C18. O GCC 10+ é compatível com C2x (rascunho de C23), enquanto o GCC 14 oferece suporte completo ao C23.
- Padrões C++: todas as versões são compatíveis com C++98, C++03, C++11, C++14, C++17 e C++20. O GCC 11+ oferece suporte experimental ao C++23, com o GCC 14 oferecendo recursos aprimorados do C++23.
- Padrões Fortran: todas as versões oferecem suporte ao Fortran 2008, com níveis variados de recursos do Fortran 2018 dependendo da versão do GCC.

 Note

Embora os padrões permaneçam consistentes entre o GCC 11 e 14, o GCC 14 oferece suporte significativamente melhor a recursos de linguagem, melhor otimização, diagnósticos aprimorados e implementação mais completa de padrões mais novos quando solicitados explicitamente usando sinalizadores `-std=`.

Go em AL2023

Talvez você queira criar seu próprio código escrito [Go](#) no Amazon Linux e talvez queira usar uma cadeia de ferramentas fornecida com AL2023. Semelhante a AL2, AL2023 atualizará o Go conjunto de ferramentas durante toda a vida útil do sistema operacional. Isso pode ser em resposta a qualquer CVE no conjunto de ferramentas que enviamos ou como parte de uma versão trimestral.

Go é uma linguagem que evolui relativamente rápido. Pode ocorrer uma situação em que as aplicações existentes escritas em Go precisem se adaptar às novas versões da cadeia de ferramentas do Go. Para saber mais sobre o Go, consulte [Go 1 and the Future of Go Programs](#).

Embora AL2023 incorpore novas versões do Go conjunto de ferramentas durante sua vida útil, isso não estará em sintonia com as versões anteriores. Go Portanto, usar o Go conjunto de ferramentas fornecido em AL2023 pode não ser adequado se você quiser criar Go código usando recursos de ponta da Go linguagem e da biblioteca padrão.

Durante a vida útil do AL2023, as versões anteriores do pacote não são removidas dos repositórios. Se for necessária uma cadeia de ferramentas do Go anterior, você poderá optar por renunciar às correções de bugs e segurança das cadeias de ferramentas do Go mais recentes e instalar uma versão anterior dos repositórios usando os mesmos mecanismos disponíveis para qualquer RPM.

Se quiser criar seu próprio Go código AL2023, você pode usar o Go conjunto de ferramentas incluído AL2023 com o conhecimento de que esse conjunto de ferramentas pode avançar durante toda a vida útil do. AL2023

AL2023 Funções Lambda escritas em Go

Como o Go compila em código nativo, o Lambda trata o Go como um runtime personalizado. Você pode usar o `provided.al2023` tempo de execução para implantar Go funções no AL2023 Lambda.

Para saber mais, consulte [Criar funções do Lambda com Go](#) no Guia do desenvolvedor do AWS Lambda .

Java no AL2023

O AL2023 fornece várias versões do [Amazon Corretto](#) para compatibilidade com workloads baseadas em Java. Todos os pacotes baseados em Java incluídos no AL2023 são desenvolvidos com Amazon Corretto 17.

Corretto é uma compilação do Open Java Development Kit (OpenJDK) com suporte de longo prazo da Amazon. O Corretto é certificado usando o Java Technical Compatibility Kit (TCK) para garantir que ele atenda ao padrão Java SE e esteja disponível no Linux, Windows e macOS.

Há um pacote [Amazon Corretto](#) disponível para cada Corretto 1.8.0, Corretto 11 e Corretto 17.

Cada versão do Corretto no AL2023 é compatível pelo mesmo período de tempo que a versão do Corretto, ou até o final da vida útil do AL2023, o que ocorrer primeiro. Para obter mais informações, consulte [Amazon Linux package support statements](#) e [Amazon Corretto FAQs](#).

Node.js em AL2 023

[Node.js](#) em AL2 023 é representado pelas versões 20, 22 e 24. O Amazon Linux segue o [cronograma de suporte](#) upstream e o status de suporte de qualquer versão do Node.js sempre pode ser verificado na página de [status de suporte do Package](#). Todas as versões do Node.js compatíveis têm namespace e podem ser instaladas no mesmo sistema simultaneamente. A implantação de namespaces garante que cada instalação do Node.js seja exclusiva no sistema de arquivos. Isso é feito renomeando os principais diretórios e arquivos com base na versão de runtime. Os nomes reais dos executáveis serão parecidos com `node-{MAJOR_VERSION}` ou `npm-{MAJOR_VERSION}`. No entanto, apenas uma versão do Node.js pode estar ativa por vez. Essa versão ativa fornece os diretórios e nomes de arquivos padrão, como `node`, `npm` ou `/usr/lib/node_modules`, apontando-os para o tempo de execução atualmente ativo.

Isso é alcançado por meio dos recursos da ferramenta `alternatives`. É importante lembrar que os nomes dos executáveis padrão são virtuais e podem mudar a qualquer momento ao apontar para outra versão do Node.js instalada. Essa flexibilidade permite que o software que usa `node` no shebang selecione a versão desejada quando invocado. No entanto, quando uma versão específica do Node.js é necessária, a persistência da versão pode ser alcançada chamando o executável com namespace (por exemplo, `node-20` ou `node-22`), que sempre usará a versão especificada do tempo de execução. Além disso, os executáveis com namespace da ferramenta `npm`, como `npm-20` ou

npm-22, estão sempre associados à versão correspondente do Node.js, independentemente do runtime ativo no momento.

Node.js é distribuído como vários pacotes com namespace que começam com "".

`nodejs{MAJOR_VERSION}` Esses pacotes fornecem o node, uma versão compatível da ferramenta npm, documentação, bibliotecas e muito mais. Por exemplo, o node e o npm dos Node.js 22 são fornecidos pelos `nodejs22-npm` pacotes `nodejs22` e, respectivamente.

A ferramenta `alternatives` fornece um comando único para alternar entre versões do Node.js. Por padrão, `alternatives` é configurada para estar no modo automático, que usa prioridades para determinar a versão ativa do Node.js. No entanto, você pode ativar qualquer versão instalada quando quiser. No momento, todas as versões compatíveis do Node.js têm a mesma prioridade, o que significa que a primeira versão instalada será ativada automaticamente.

Exemplos úteis do uso de `alternatives`

1. Verifique para que `alternatives` está configurada.

```
alternatives --list
```

2. Verifique a configuração atual de node.

```
alternatives --display node
```

3. Altere a versão do Node.js de forma interativa.

```
alternatives --config node
```

4. Mude para o modo manual e selecione uma versão específica.

```
alternatives --set node /usr/bin/node-{MAJOR_VERSION}
```

5. Volte para o modo de seleção automática de versão.

```
alternatives --auto node
```

Perl no AL2023

O AL2023 fornece a versão 5.32 da linguagem de programação [Perl](#).

Embora o Perl tenha fornecido um alto grau de compatibilidade de linguagem como parte dos lançamentos do Perl 5 nas últimas décadas, não se espera que o Amazon Linux mude do Perl 5.32 durante o lançamento do AL2023. O Amazon Linux continuará aplicando patches de segurança no Perl durante a vida útil do AL2023, de acordo com nossas [declarações de suporte de pacotes](#).

Perl módulos no AL2023

Vários módulos do Perl são empacotados como RPMs no AL2023. Embora existam muitos módulos do Perl disponíveis como RPMs, o Amazon Linux não tem como objetivo empacotar todos os módulos do Perl possíveis. Módulos empacotados como RPMs podem ser usados por outros pacotes RPM do sistema operacional, portanto, o Amazon Linux priorizará esses patches de segurança em vez de atualizações puras de recursos.

O AL2023 também inclui CPAN para que os desenvolvedores de Perl possam usar o gerenciador de pacotes idiomáticos para módulos do Perl.

PHPem AL2 023

AL2Atualmente, o 023 fornece a linguagem de [PHP](#) programação, versões 8.1, 8.2, 8.3, 8.4 e 8.5. Cada versão receberá suporte pelo mesmo período que o PHP upstream. Para saber mais, consulte [Package support statements](#).

Migrar de versões mais antigas de PHP

A comunidade do PHP upstream reuniu uma documentação abrangente de migração para:

- [de PHP 8.4.x a 8.5.x PHP](#)
- [Migrar do PHP 8.3.x para o PHP 8.4.x](#)
- [Migrar do PHP 8.2.x para o PHP 8.3.x](#)
- [Migrar do PHP 8.1.x para o PHP 8.2.x](#)
- [Migrar do PHP 8.0.x para o PHP 8.1.x](#)

AL2 inclui PHP 8.0, 8.1 e 8.2 para `amazon-linux-extras` permitir um caminho fácil de atualização para AL2 023.

Migrando de PHP para a versão 7.x

Note

O projeto do [PHP](#) mantém uma lista e um cronograma de [versões compatíveis](#), bem como uma lista de [ramificações incompatíveis](#).

Quando o AL2 023 foi lançado, todas as versões 7.x e 5.x do não [PHP](#) eram suportadas pela PHP comunidade e não foram incluídas como opções no 023. AL2

A comunidade do PHP upstream reuniu uma [documentação abrangente para migração do PHP 7.4 para o PHP 8.0](#). Combinada com a documentação mencionada na seção anterior sobre migração para o PHP 8.1 e PHP 8.2, você pode migrar sua aplicação baseada em PHP para o PHP moderno.

Note

AL2 inclui PHP 7,1, 7,2, 7,3 e 7,4 pol. amazon-linux-extras. É importante observar que todos esses extras têm end-of-life ou não garantia de receber mais atualizações de segurança.

PHPmódulos em AL2 023

AL2O 023 inclui muitos PHP módulos incluídos no PHP Core. AL2O 023 não visa incluir todos os pacotes na [Biblioteca Comunitária de PHP Extensão \(PECL\)](#).

Python no AL2023

O AL2023 removeu o Python 2.7, e todos os componentes que exigem Python já foram escritos para funcionar com o Python 3.

O AL2023 disponibiliza o Python 3 como `/usr/bin/python3` para manter a compatibilidade com o código do cliente, bem como com o código Python distribuído com o AL2023. Isso permanecerá como Python 3.9 durante toda a vida útil do AL2023.

A versão do Python para a qual `/usr/bin/python3` aponta é considerada o sistema Python e, para o AL2023, é o Python 3.9.

Versões mais recentes do Python, como a Python 3.11, são disponibilizadas na forma de pacotes no AL2023 e receberão suporte durante toda a vida útil das versões upstream. Para obter informações sobre o tempo de suporte ao Python 3.11, consulte [Python 3.11](#).

Várias versões do Python podem ser instaladas simultaneamente no AL2023. Embora `/usr/bin/python3` seja sempre Python 3.9, cada versão do Python tem namespace e pode ser encontrada pelo número da versão. Por exemplo, se `python3.11` estiver instalado, `/usr/bin/python3.11` existirá ao lado `/usr/bin/python3.9` e o symlink `/usr/bin/python3` apontará para `/usr/bin/python3.9`.

Note

Não altere o destino do link simbólico `/usr/bin/python3`, pois isso pode comprometer a funcionalidade principal do AL2023.

Módulos do Python no AL2023

Vários módulos do Python são empacotados como RPMs no AL2023. Normalmente, os RPMs para módulos Python serão criados visando somente a versão do sistema do Python.

Rubyem AL2023

[Ruby](#) em AL2023 é representado pelas versões 3.2 e 3.4. O Amazon Linux segue o cronograma de suporte upstream e o status de suporte de qualquer Ruby versão sempre pode ser verificado no site do [Ruby](#). Todas as versões do Ruby compatíveis têm namespace e podem ser instaladas no mesmo sistema simultaneamente. A implantação de namespaces garante que cada instalação do Ruby seja exclusiva no sistema de arquivos. Isso é feito renomeando os principais diretórios e arquivos com base na versão de runtime. Os nomes reais dos executáveis serão parecidos com `ruby {MAJOR.MINOR}` (por exemplo, `ou. ruby3.2 ruby3.4 Ruby3.4` também fornece o binário com namespace MRI (Matz's Ruby Interpreter)`ruby3.4-mri`, que se refere à implementação de referência padrão baseada em C do Ruby. No entanto, apenas uma versão do Ruby pode estar ativa por vez. Essa versão ativa fornece os diretórios e nomes de arquivos padrão, como `ruby`, `gem` ou `bundle`, apontando-os para o tempo de execução atualmente ativo.

Isso é alcançado por meio dos recursos da ferramenta `alternatives`. É importante lembrar que os nomes dos executáveis padrão são virtuais e podem mudar a qualquer momento ao apontar para outra versão do Ruby instalada. Essa flexibilidade permite que o software que usa ruby in the

shebang selecione a versão desejada quando invocado. No entanto, quando uma versão específica do Ruby é necessária, a persistência da versão pode ser alcançada chamando o executável com namespace (por exemplo, `ruby3.2` ou `ruby3.4`), que sempre usará a versão especificada do tempo de execução. Além disso, os executáveis com namespace das ferramentas `gem` e `bundler`, como,., ou `ruby3.2-gem` `ruby3.4-gem` `ruby3.2-bundler`, estão sempre associados à Ruby versão correspondente `ruby3.4-bundler`, independentemente do tempo de execução ativo no momento.

Ruby é distribuído como vários pacotes com namespace que começam com "".

`ruby{MAJOR.MINOR}` Esses pacotes fornecem versões ruby compatíveis das ferramentas `gem` e `bundler`, documentação, bibliotecas e muito mais. Por exemplo, o tempo de execução principal do Ruby 3.2 é fornecido pelo `ruby3.2` pacote, que extrai `ruby3.2-rubygems` (fornecendo `gem`) e `ruby3.2-rubygem-bundler` (fornecendo pacote e empacotador) como dependências.

Depois de instalar uma Ruby versão, as entradas das ferramentas complementares podem aparecer como nulas na configuração das alternativas. Isso pode ser verificado executando `alternatives --display ruby`. Se as entradas aparecerem como nulas, elas deverão ser registradas manualmente usando `alternatives --install`. Por exemplo, para registrar todas as ferramentas complementares para a Ruby versão 3.4:

```
sudo alternatives --install /usr/bin/gem gem /usr/bin/ruby3.4-gem 34
sudo alternatives --install /usr/bin/bundle bundle /usr/bin/ruby3.4-bundle 34
sudo alternatives --install /usr/bin/bundler bundler /usr/bin/ruby3.4-bundler 34
sudo alternatives --install /usr/bin/erb erb /usr/bin/ruby3.4-erb 34
sudo alternatives --install /usr/bin/racc racc /usr/bin/ruby3.4-racc 34
sudo alternatives --install /usr/bin/rdoc rdoc /usr/bin/ruby3.4-rdoc 34
sudo alternatives --install /usr/bin/ri ri /usr/bin/ruby3.4-ri 34
```

O valor da prioridade (por exemplo, 34 para Ruby 3,4, 32 para Ruby 3,2) deve corresponder à prioridade usada na entrada alternativa principal do ruby. Depois de registradas, as ferramentas complementares serão gerenciadas automaticamente junto com a alternativa ruby.

A ferramenta `alternatives` fornece um comando único para alternar entre versões do Ruby. Por padrão, `alternatives` é configurada para estar no modo automático, que usa prioridades para determinar a versão ativa do Ruby. No entanto, você pode ativar qualquer versão instalada quando quiser. No momento, todas as versões compatíveis do Ruby têm a mesma prioridade, o que significa que a primeira versão instalada será ativada automaticamente.

Exemplos úteis do uso de alternatives

1. Verifique para que `alternatives` está configurada.

```
alternatives --list
```

2. Verifique a configuração atual do ruby

```
alternatives --display ruby
```

3. Altere a versão do Ruby de forma interativa.

```
alternatives --config ruby
```

4. Mude para o modo manual e selecione uma versão específica.

```
alternatives --set ruby /usr/bin/ruby{MAJOR.MINOR}
```

5. Volte para o modo de seleção automática de versão.

```
alternatives --auto ruby
```

Rustem AL2023

Talvez você queira criar código escrito [Rust](#) no Amazon Linux e talvez queira usar uma cadeia de ferramentas fornecida com AL2023.

Semelhante a AL2, AL2023 atualizará o Rust conjunto de ferramentas durante toda a vida útil do sistema operacional. Isso pode ser em resposta a qualquer CVE no conjunto de ferramentas que enviamos ou como parte de uma versão trimestral.

[Rust](#) é uma linguagem relativamente rápida, com novos lançamentos em uma cadência de aproximadamente seis semanas. Essas versões podem adicionar um novo idioma ou recursos de biblioteca padrão. Embora AL2023 incorpore novas versões do Rust conjunto de ferramentas durante sua vida útil, isso não estará em sintonia com as versões anteriores. Rust Portanto, usar o Rust conjunto de ferramentas fornecido em AL2023 pode não ser adequado se você quiser criar Rust código usando recursos de ponta da linguagem. Rust

Durante a vida útil do AL2023, as versões antigas do pacote não são removidas dos repositórios. Se for necessária uma cadeia de ferramentas do Rust anterior, você poderá optar por renunciar às correções de bugs e segurança das cadeias de ferramentas do Rust mais recentes e instalar uma versão anterior dos repositórios usando os mesmos mecanismos disponíveis para qualquer RPM.

Se quiser criar seu próprio Rust código AL2023, você pode usar o Rust conjunto de ferramentas incluído AL2023 com o conhecimento de que esse conjunto de ferramentas pode avançar durante toda a vida útil do. AL2023

AL2023 Funções Lambda escritas em Rust

Como o Rust compila em código nativo, o Lambda trata o Rust como um runtime personalizado. Você pode usar o `provided.al2023` tempo de execução para implantar Rust funções no AL2023 Lambda.

Para saber mais, consulte [Criar funções do Lambda com Rust](#) no Guia do desenvolvedor do AWS Lambda .

TypeScript no AL2023

Note

Este documento fornece as informações essenciais sobre o TypeScript e seu ambiente de execução baseado em Node.js. Ele também aborda um fluxo de trabalho de desenvolvimento típico e explica como o TypeScript é empacotado no AL2023 para oferecer um ambiente de desenvolvimento consistente e reproduzível.

[TypeScript](#) (TS) é uma linguagem de programação baseada em JavaScript (JS) que oferece todos os recursos do JS e também o [estende com um sistema de tipos](#). Em um cenário típico, os programas escritos em TS são traduzidos primeiro para o código JS, depois são executados como qualquer outro programa JS normal pelo Node.js. Na terminologia específica do TS, esse processo de tradução é chamado de [“compilação” e é executado por um “compilador”](#), chamado tsc. O compilador tsc em si é escrito em JS, portanto, para ser executado, ele também precisa de um ambiente de runtime do JS, como Node.js. Ao contrário de alguns outros ambientes de runtime do JS, Node.js só tem suporte experimental e leve ao TS no momento. Um suporte completo ao TS, incluindo verificação de tipos, ainda requer o uso de pacotes de terceiros, como o [typescript](#). A maneira esperada de obter o tsc (compilador do TS) para o ambiente de runtime do Node.js é instalar o módulo de nós typescript. Isso pode ser feito usando um dos gerenciadores de pacotes, normalmente o npm. Há duas formas de instalar o compilador do TS usando npm: globalmente e em um projeto. [O método oficialmente recomendado é instalar](#) o compilador do TS por projeto, o que garante consistência e reprodutibilidade a longo prazo para os projetos. No entanto, instalar o

compilador do TS globalmente ainda pode ser útil, pois fornece a mesma versão para todo o host e seu runtime do JS e, portanto, para projetos que não têm um compilador do TS instalado localmente. É exatamente assim que os pacotes RPM disponíveis no Amazon Linux, como `nodejs20-typescript` ou `nodejs22-typescript`, instalam um compilador do TS: globalmente no nível do sistema e separadamente para cada versão do Node.js compatível.

O tsc não depende diretamente de nenhuma versão do Node.js. O compilador espera determinado nível de recursos de runtime, que são definidos em um arquivo especial (`tsconfig.json`) por meio de opções como [target](#) e [lib](#). Os valores dessas opções representam uma versão do padrão [ECMAScript](#) (ES), que pode (ou não) ser compatível com o ambiente de runtime do JS. Versões diferentes do Node.js oferecem suporte a diferentes versões do padrão ES. Quanto mais recente for a versão do Node.js, maior e mais completa será a versão do padrão ES compatível. Se `tsconfig.json` não existir no diretório-raiz de um projeto, será usado o conjunto padrão de opções de configuração. A tabela de compatibilidade com as diferentes versões do Node.js e os recursos compatíveis de várias versões do padrão ES está disponível em [node.green](#). O tsc tem mais de 100 opções diferentes, que podem ser definidas em `tsconfig.json`. Também há suporte ao encadeamento de configurações quando algumas opções de configuração são definidas em outro arquivo e, depois, incluídas no arquivo principal. Essa abordagem permite instalar uma [configuração básica do TS](#) compatível com determinada versão do Node.js, depois estendê-lo com opções específicas do projeto. Felizmente, as configurações básicas do TS para o Node.js estão disponíveis na forma de módulos de nós que podem ser instalados em uma pasta de projeto usando npm. Aqui está o código-fonte da configuração para o Node.js versão [18](#), [20](#) e [22](#).

O design de runtime baseado em Node.js tem certa fraqueza: ele só oferece suporte a uma versão do runtime em um host e requer reprodutibilidade e consistência de todas as dependências no nível do projeto. Isso levou à seguinte abordagem comum de uso do TypeScript: a instalação local, dentro de um projeto, do compilador do TS, da configuração básica do TS para a versão do Node.js atual e de todas as dependências de software. Embora se espere que os módulos de nós instalados globalmente sejam apenas ferramentas de CLI, como npm, o tsc, que também é uma ferramenta de CLI, raramente é instalado globalmente. Felizmente, as instalações globais (em todo o sistema) e locais (dentro de um projeto) do tsc podem coexistir sem problemas e também podem ser versões diferentes que são usadas de forma independente. Observe que um tsc instalado localmente deve ser executado usando a ferramenta npx, que é instalada com o npm. Assim, mesmo com um compilador do TS do sistema, os usuários têm a oportunidade de escolher versões dos componentes do runtime, como Node.js (alternando a versão ativa por meio da ferramenta `alternatives`), um compilador do TS (instalando-o local ou globalmente, e também alternando a versão ativa por meio da ferramenta `alternatives`), e configurar para as necessidades específicas.

O Amazon Linux empacota um compilador do TS da mesma forma que outros módulos de nós instalados globalmente, como npm, com base na versão do Node.js. Pacotes e binários têm namespace e contêm a versão principal do Node.js como parte de seus nomes. O nome do executável padrão do compilador, tsc, é gerenciado no runtime pela ferramenta alternatives e aponta para a versão atualmente ativa do Node.js para a qual ele foi instalado e será executado. Essa seleção não depende da versão atual do runtime do Node.js. É possível ter o executável node apontando para o Node.js 20 e o tsc configurado para ser interpretado pelo Node.js 22. Também é possível usar os nomes com namespace de um compilador do TS, por exemplo, tsc-{MAJOR_VERSION}, independentemente do nome tsc padrão configurado.

Comandos úteis para gerenciar a versão ativa de um compilador do TS

1. Verifique para que alternatives está configurada.

```
alternatives --list
```

2. Verifique a configuração atual do tsc.

```
alternatives --display tsc
```

3. Altere a versão do tsc de forma interativa.

```
alternatives --config tsc
```

4. Mude para o modo manual e selecione uma versão específica.

```
alternatives --set tsc /usr/bin/tsc-{MAJOR_VERSION}
```

5. Volte para o modo de seleção automática de versão.

```
alternatives --auto tsc
```

Um exemplo de instalação e uso de várias versões do Node e de um compilador do TS no mesmo sistema:

```
# Check the AL2023 release
$ cat /etc/amazon-linux-release
Amazon Linux release 2023.9.20250929 (Amazon Linux)
```

```
# Install a TypeScript compiler for Node.js 20 and 22
# Node.js 20 and 22 will be installed automatically
$ sudo dnf install -qy nodejs20-typescript nodejs22-typescript

# Check what was installed
$ rpm -q nodejs20 nodejs20-typescript nodejs22 nodejs22-typescript
nodejs20-20.19.5-1.amzn2023.0.1.x86_64
nodejs20-typescript-5.9.2-1.amzn2023.0.1.noarch
nodejs22-22.19.0-1.amzn2023.0.1.x86_64
nodejs22-typescript-5.9.2-1.amzn2023.0.1.noarch

# Check the active version of Node.js - it is version 20
$ alternatives --display node
node - status is auto.
  link currently points to /usr/bin/node-20
/usr/bin/node-20 - priority 100
  slave npmrc: /usr/lib/nodejs20/lib/node_modules/npm/npmrc
  slave npm: /usr/bin/npm-20
  slave npx: /usr/bin/npx-20
  slave node_modules: /usr/lib/nodejs20/lib/node_modules
/usr/bin/node-22 - priority 100
  slave npmrc: /usr/lib/nodejs22/lib/node_modules/npm/npmrc
  slave npm: /usr/bin/npm-22
  slave npx: /usr/bin/npx-22
  slave node_modules: /usr/lib/nodejs22/lib/node_modules
Current 'best' version is /usr/bin/node-20.

# Check the active JS runtime version for TypeScript
# Currently, the tsc compiler will be executed by Node.js 22
$ alternatives --display tsc
tsc - status is auto.
  link currently points to /usr/bin/tsc-22
/usr/bin/tsc-22 - priority 100
  slave tsserver: /usr/bin/tsserver-22
/usr/bin/tsc-20 - priority 100
  slave tsserver: /usr/bin/tsserver-20
Current 'best' version is /usr/bin/tsc-22.

# Check versions printed by executables
$ node -v
v20.19.5

$ tsc -v
```

```
Version 5.9.2
```

```
# while the node is 20, tsc is executed by node 22 anyway
```

```
$ head -1 /usr/bin/tsc
```

```
#!/usr/bin/node-22
```

```
# However, instead of default executable names, e.g. node or tsc,
```

```
# we can use namespaced names to target any installed version
```

```
$ node-20 -v
```

```
v20.19.5
```

```
$ node-22 -v
```

```
v22.19.0
```

```
$ tsc-20 -v
```

```
Version 5.9.2
```

```
$ tsc-22 -v
```

```
Version 5.9.2
```

```
$ head -1 /usr/bin/tsc-20
```

```
#!/usr/bin/node-20
```

```
$ head -1 /usr/bin/tsc-22
```

```
#!/usr/bin/node-22
```

Usuários e grupos reservados do AL2023

O AL2023 pré-aloca determinados usuários e grupos durante o provisionamento da imagem e durante a instalação de determinados pacotes. Os usuários, grupos e seus UIDs e GIDs associados estão listados aqui para evitar conflitos.

Tópicos

- [Lista de usuários reservados do AL2023](#)
- [Lista de grupos reservados do AL2023](#)

Lista de usuários reservados do AL2023

Nome de usuário	UID
raiz	0
bin	1
daemon	2
adm	3
lp	4
sincronização	5
shutdown	6
parar	7
correio	8
operador	11
jogos	12
ftp	14

Nome de usuário	UID
squid	23
named	25
postgres	26
mysql	27
nscd	28
nscd	28
rpcuser	29
rpc	32
mailnull	47
apache	48
smmsp	51
tomcat	53
ldap	55
tss	59
nsld	65
avahi	70
tcpdump	72
sshd	74
radvd	75
dbus	81

Nome de usuário	UID
postfix	89
dovecot	97
stapusr	156
stapsys	157
stapdev	158
avahi-autoipd	170
pulse	171
rtkit	172
sanlock	179
systemd-network	192
systemd-resolve	193
uidd	961
stap-server	962
systemd-journal-remote	963
redis6	970
pesign	971
smtpq	972
smtpd	973
nginx	974
munge	975

Nome de usuário	UID
memcached	976
sphinx	977
haproxy	978
flatpak	979
debuginfod	980
dovnull	981
dnsmasq	982
unbound	983
clamscan	984
clamilt	985
clamupdate	986
colord	987
ods	988
aws-kinesis-agent-user	989
saslauth	990
cwagent	991
polkitd	992
ec2-instance-connect	993
chrony	994
systemd-timesync	995

Nome de usuário	UID
systemd-coredump	996
libstoragemgmt	997
systemd-oom	999
ec2-user	1000
nobody	65534

Listagem por nome

Nome de usuário	UID
adm	3
apache	48
avahi	70
avahi-autoipd	170
aws-kinesis-agent-user	989
bin	1
chrony	994
clamilt	985
clamscan	984
clamupdate	986
colord	987
cwagent	991

Nome de usuário	UID
daemon	2
dbus	81
debuginfod	980
dnsmasq	982
dovecot	97
dovnull	981
ec2-instance-connect	993
ec2-user	1000
flatpak	979
ftp	14
jogos	12
parar	7
haproxy	978
ldap	55
libstoragemgmt	997
lp	4
correio	8
mailnull	47
memcached	976
munge	975

Nome de usuário	UID
mysql	27
named	25
nginx	974
nobody	65534
nscd	28
nscd	28
nslcd	65
ods	988
operador	11
pesign	971
polkitd	992
postfix	89
postgres	26
pulse	171
radvd	75
redis6	970
raiz	0
rpc	32
rpcuser	29
rtkit	172

Nome de usuário	UID
sanlock	179
saslauth	990
shutdown	6
smmsp	51
smtpd	973
smtpq	972
sphinx	977
squid	23
sshd	74
stap-server	962
stapdev	158
stapsys	157
stapusr	156
sincronização	5
systemd-coredump	996
systemd-journal-remote	963
systemd-network	192
systemd-oom	999
systemd-resolve	193
systemd-timesync	995

Nome de usuário	UID
tcpdump	72
tomcat	53
tss	59
unbound	983
uidd	961

Lista de grupos reservados do AL2023

Group name	GID
raiz	0
bin	1
daemon	2
sys	3
adm	4
tty	5
disco	6
disco	6
lp	7
mem	8
kmem	9
wheel	10

Group name	GID
cdrom	11
correio	12
correio	12
man	15
dialout	18
floppy	19
jogos	20
slocate	21
utmp	22
squid	23
named	25
postgres	26
mysql	27
nscd	28
nscd	28
rpcuser	29
rpc	32
fita	33
utempter	35
kvm	36

Group name	GID
video	39
mailnull	47
apache	48
ftp	50
smmsp	51
tomcat	53
bloquear	54
ldap	55
tss	59
áudio	63
avahi	70
tcpdump	72
sshd	74
radvd	75
saslauth	76
dbus	81
screen	84
wbpriv	88
postfix	89
postdrop	90

Group name	GID
dovecot	97
usuários	100
input	104
render	105
sgx	106
mock	135
stapusr	156
stapusr	156
stapsys	157
stapsys	157
stapdev	158
stapdev	158
avahi-autoipd	170
pulse	171
rtkit	172
sanlock	179
systemd-journal	190
systemd-network	192
systemd-resolve	193
usbmon	959

Group name	GID
wireshark	960
uidd	961
stap-server	962
systemd-journal-remote	963
usershares	964
redis6	965
pesign	966
smtpq	967
smtpd	968
nginx	969
munge	970
memcached	971
sphinx	972
rastreamento	973
haproxy	974
flatpak	975
debuginfod	976
dovnull	977
dnsmasq	978
unbound	979

Group name	GID
clamscan	980
clamilt	981
virusgroup	982
virusgroup	982
virusgroup	982
clamupdate	983
printadmin	984
colord	985
ods	986
docker	987
aws-kinesis-agent-user	988
cwagent	989
pulse-rt	990
pulse-access	991
ec2-instance-connect	993
chrony	994
systemd-timesync	995
systemd-coredump	996
libstoragemgmt	997
ssh_keys	998

Group name	GID
systemd-oom	999
ec2-user	1000
ne	1001
polkitd	9920
nobody	65534

Listagem por nome

Group name	GID
adm	4
apache	48
áudio	63
avahi	70
avahi-autoipd	170
aws-kinesis-agent-user	988
bin	1
cdrom	11
chrony	994
clamilt	981
clamscan	980
clamupdate	983

Group name	GID
colord	985
cwagent	989
daemon	2
dbus	81
debuginfod	976
dialout	18
disco	6
disco	6
dnsmasq	978
docker	987
dovecot	97
dovnull	977
ec2-instance-connect	993
ec2-user	1000
flatpak	975
floppy	19
ftp	50
jogos	20
haproxy	974
input	104

Group name	GID
kmem	9
kvm	36
ldap	55
libstoragemgmt	997
bloquear	54
lp	7
correio	12
correio	12
mailnull	47
man	15
mem	8
memcached	971
mock	135
munge	970
mysql	27
named	25
ne	1001
nginx	969
nobody	65534
nscd	28

Group name	GID
nscd	28
ods	986
pesign	966
polkitd	9920
postdrop	90
postfix	89
postgres	26
printadmin	984
pulse	171
pulse-access	991
pulse-rt	990
radvd	75
redis6	965
render	105
raiz	0
rpc	32
rpcuser	29
rtkit	172
sanlock	179
saslauth	76

Group name	GID
screen	84
sgx	106
slocate	21
smmsp	51
smtpd	968
smtpq	967
sphinx	972
squid	23
ssh_keys	998
sshd	74
stap-server	962
stapdev	158
stapdev	158
stapsys	157
stapsys	157
stapusr	156
stapusr	156
sys	3
systemd-coredump	996
systemd-journal	190

Group name	GID
systemd-journal-remote	963
systemd-network	192
systemd-oom	999
systemd-resolve	193
systemd-timesync	995
fita	33
tcpdump	72
tomcat	53
rastreamento	973
tss	59
tty	5
unbound	979
usbmon	959
usuários	100
usershares	964
utempter	35
utmp	22
uudd	961
video	39
virusgroup	982

Group name	GID
virusgroup	982
virusgroup	982
wbpriv	88
wheel	10
wireshark	960

Lista de codecs disponíveis em 023 AL2

AL2O 023 fornece uma seleção de codecs multimídia por meio de seus repositórios padrão. Esta página fornece uma visão geral dos codecs e os respectivos casos de uso típicos.

Important

O uso e a distribuição de codecs incluídos no Amazon Linux podem exigir que você obtenha direitos de licença de terceiros, incluindo proprietários ou licenciadores de determinados formatos de áudio e vídeo de terceiros. Você é exclusivamente responsável por obter essas licenças e pagar quaisquer royalties ou taxas necessárias.

Codec	Description
flac	Um codec de áudio sem perdas gratuito e de código aberto que comprime o áudio sem perder nenhum dado ou qualidade, comumente usado para armazenamento de áudio de alta qualidade.
fdk-aac-free	Uma implementação de código aberto do padrão AAC (Advanced Audio Codec), fornecendo compressão de áudio de alta qualidade para MP3 alternativas como streaming ou armazenamento de arquivos
webrtc-audio-processing	Uma biblioteca para processamento de áudio usada em WebRTC (Web Real-Time Communication), oferecendo recursos como supressão de ruído, cancelamento de eco e controle de ganho.
opus	Um codec de áudio altamente versátil e eficiente projetado para streaming em tempo real, oferecendo baixa latência e suporte a

Codec	Description
	uma ampla variedade de aplicações de áudio, incluindo VoIP e streaming de música.
libsndfile	Uma biblioteca para leitura e gravação de arquivos de áudio em vários formatos (como WAV, AIFF e FLAC), comumente usada em ferramentas de processamento e manipulação de áudio.
openh264	Uma implementação de código aberto do codec de vídeo H.264. Ele fornece suporte de codificação e decodificação para vídeo H.264 usado em aplicativos de streaming, conferência e multimídia.
svt-av1	Uma implementação de código aberto e de alto desempenho do codec de AV1 vídeo. Ele fornece codificação e decodificação escaláveis para AV1 vídeo, otimizadas para processamento moderno e CPUs paralelo.
dav1d	Uma implementação de código aberto do decodificador de AV1 vídeo focada em velocidade, eficiência e portabilidade. Ele fornece AV1 decodificação de alto desempenho para uma ampla variedade de plataformas e aplicativos.
libde265	Uma implementação de código aberto do decodificador de vídeo H.265/HEVC. Ele fornece decodificação eficiente e portátil de fluxos de vídeo HEVC para uso em aplicativos e estruturas multimídia.

Codec	Description
libheif	Uma biblioteca de código aberto para leitura e gravação de arquivos de imagem HEIF e AVIF. Ele fornece codificação, decodificação e conversão eficientes de imagens e sequências de imagens usando formatos de compressão modernos, como HEVC e AV1
mpg123	Um decodificador de alto desempenho para fluxos de áudio MPEG, incluindo arquivos. MP3 Ele fornece decodificação de áudio rápida e precisa para aplicativos de reprodução, streaming e processamento de áudio.
x265	O objetivo principal do x265 é se tornar o melhor codificador H.265/HEVC disponível em qualquer lugar, oferecendo a maior eficiência de compressão e o mais alto desempenho em uma ampla variedade de plataformas de hardware. Este pacote contém o codificador de linha de comando.

Segurança e compatibilidade no Amazon Linux 2023

Important

Se quiser relatar uma vulnerabilidade ou tiver uma preocupação de segurança em relação aos serviços de nuvem da AWS ou projetos de código aberto, entre em contato com a AWS Security por meio da [página Vulnerability Reporting](#)

A segurança da nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você contará com um data center e uma arquitetura de rede criados para atender aos requisitos das organizações com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem: a AWS é responsável pela proteção da infraestrutura que executa serviços AWS na Nuvem AWS. A AWS também fornece serviços que podem ser usados com segurança. Auditores terceirizados testam e verificam regularmente a eficácia da nossa segurança como parte dos [Programas de conformidade da AWS](#). Para saber mais sobre os programas de conformidade que se aplicam ao AL2023, consulte [Serviços da AWS no escopo por programa de conformidade](#).
- Segurança na nuvem - sua responsabilidade é determinada pelo serviço AWS que você usa. Você também é responsável por outros fatores, inclusive a sensibilidade de seus dados, os requisitos da sua empresa, leis e regulamentos aplicáveis.

Tópicos

- [Consultorias de segurança do Amazon Linux para AL2023](#)
- [Listar recomendações aplicáveis](#)
- [Aplicar atualizações de segurança no local](#)
- [Configurando modos do SELinux para o AL2023](#)
- [Habilitar o modo FIPS no AL2023](#)
- [Habilitar o modo FIPS em um contêiner do AL2023](#)
- [Trocar de provedor de OpenSSL FIPS no AL2023](#)
- [AL2023 Endurecimento do kernel](#)

- [UEFI Secure Boot no AL2023](#)

Consultorias de segurança do Amazon Linux para AL2023

Embora trabalhem duro para tornar o Amazon Linux seguro, às vezes haverá problemas de segurança que precisam ser corrigidos. Uma recomendação é emitida quando uma correção está disponível. O principal local onde publicamos recomendações é o Amazon Linux Security Center (ALAS). Para obter informações, consulte o [Amazon Linux Security Center](#).

Important

Se você quiser denunciar uma vulnerabilidade ou tiver uma preocupação de segurança em relação a serviços em AWS nuvem ou projetos de código aberto, entre em contato com a AWS Segurança usando a [página Relatório de vulnerabilidades](#)

As informações sobre os problemas e as atualizações relevantes que afetam AL2023 são publicadas pela equipe do Amazon Linux em vários locais. É comum que as ferramentas de segurança busquem informações dessas fontes primárias e apresentem os resultados para você. Dessa forma, talvez você não interaja diretamente com as fontes primárias que o Amazon Linux publica, mas sim com a interface fornecida pelas suas ferramentas preferidas, como o [Amazon Inspector](#).

Anúncios do Amazon Linux Security Center

Os anúncios do Amazon Linux são fornecidos para itens que não se enquadram em uma recomendação. Esta seção contém anúncios sobre o próprio ALAS, junto com informações que não se enquadram em uma recomendação. Para saber mais, consulte [Amazon Linux Security Center \(ALAS\) Announcements](#).

Por exemplo, o item [2021-001 - Amazon Linux Hotpatch Announcement for Apache Log4j](#) se encaixa em um anúncio, e não em uma recomendação. Nesse anúncio, o Amazon Linux adicionou um pacote para ajudar os clientes a mitigar um problema de segurança em software que não fazia parte do Amazon Linux.

O [Amazon Linux Security Center CVE Explorer](#) também foi anunciado nos anúncios do ALAS. Para obter mais informações, consulte [Novo site para CVEs](#).

Perguntas frequentes do Amazon Linux Security Center

Para obter respostas a algumas perguntas frequentes sobre o ALAS e como o Amazon Linux avalia CVEs, consulte [Perguntas frequentes do Amazon Linux Security Center \(ALAS\)](#) (). FAQs

Recomendações do ALAS

Uma recomendação do Amazon Linux contém informações importantes e relevantes para os usuários do Amazon Linux, geralmente informações sobre atualizações de segurança. O [Amazon Linux Security Center](#) é onde os avisos podem ser vistos na web. As informações de recomendações também fazem parte dos metadados do repositório de pacotes RPM.

Recomendações e repositórios RPM

Um repositório de pacotes do Amazon Linux 2023 pode conter metadados que descrevem zero ou mais atualizações. O comando `dnf updateinfo` tem o nome do arquivo de metadados do repositório que contém essas informações, `updateinfo.xml`. Embora o comando tenha o nome `updateinfo` e o arquivo de metadados se refira a uma `update`, todos eles se referem a atualizações de pacotes que fazem parte de uma recomendação.

As recomendações do Amazon Linux são publicadas no site do [Amazon Linux Security Center](#), junto com as informações presentes nos metadados do repositório RPM aos quais o gerenciador de pacotes `dnf` se refere. O site e os metadados do repositório são eventualmente consistentes, e pode haver inconsistências nas informações do site e nos metadados do repositório. Isso normalmente ocorre quando uma nova versão do AL2023 está em processo de lançamento. Há uma atualização de um Aviso após a AL2023 versão mais recente.

Embora seja comum que uma nova recomendação seja emitida junto com a atualização do pacote que resolve o problema, esse nem sempre é o caso. Uma recomendação pode ser criada para um novo problema que é tratado em pacotes já lançados. Um Aviso existente também pode ser atualizado com novos CVEs , que são abordados pela atualização existente.

O [Atualizações determinísticas por meio de repositórios versionados em AL2023](#) recurso do Amazon Linux 2023 significa que o repositório RPM de uma AL2023 versão específica contém um instantâneo dos metadados do repositório RPM a partir dessa versão. Isso inclui os metadados que descrevem as atualizações de segurança. O repositório RPM de uma AL2023 versão específica não é atualizado após o lançamento. Avisos de segurança novos ou atualizados não estarão visíveis ao examinar uma versão mais antiga dos repositórios AL2023 RPM. Consulte a [Listar recomendações](#)

[aplicáveis](#) seção para saber como usar o gerenciador de `dnf` pacotes para examinar a versão do `latest` repositório ou uma AL2023 versão específica.

Consultivo IDs

Cada recomendação é identificada por um `id`. Atualmente, é uma peculiaridade do Amazon Linux, onde o site do [Amazon Linux Security Center](#) listará um aviso como [ALAS-2024-581](#), enquanto o gerenciador de `dnf` pacotes [listará esse aviso](#) como tendo o ID de `-2024-581`. ALAS2023 Ao [Aplicar atualizações de segurança no local](#), o ID do gerenciador de pacotes precisa ser usado para se referir a uma recomendação específica.

Para o Amazon Linux, cada versão principal do sistema operacional tem seu próprio namespace de Advisory. IDs Não se deve fazer suposições quanto ao formato do Amazon Linux Advisory. IDs Historicamente, a Amazon Linux Advisory IDs seguiu o padrão `namespace-year-number`. O intervalo completo de valores possíveis para `namespace` não está definido, mas inclui ALAS, ALASCORRETT08, ALAS2023, ALAS2, ALASPYTHON3.8 e ALASUNBOUND-1.17. `year` foi o ano em que a recomendação foi criada e `number` é um número inteiro exclusivo no namespace.

Embora o Advisory normalmente IDs seja sequencial e na ordem em que as atualizações sejam lançadas, há muitos motivos pelos quais esse não pode ser o caso, portanto, isso não deve ser assumido.

Trate o ID da recomendação como uma string opaca que é exclusiva para cada versão principal do Amazon Linux.

No Amazon Linux 2, cada Extra fica em um repositório RPM separado, e os metadados da recomendação ficam contidos somente no repositório ao qual são relevantes. Uma recomendação para um repositório não se aplica a outro repositório. No site do [Amazon Linux Security Center](#), há uma lista de recomendações para cada versão principal do Amazon Linux, e ela não está separada em listas por repositório.

Como AL2023 não usa o mecanismo Extras para empacotar versões alternativas de pacotes, atualmente existem apenas dois repositórios RPM, cada um com recomendações, o repositório e o core repositório. `livepatch` O repositório `livepatch` é para [Aplicação de patches do kernel em tempo real no AL2023](#).

Data de lançamento da recomendação e data de atualização da recomendação

A data de lançamento de recomendações do Amazon Linux indica quando a atualização de segurança foi disponibilizada ao público pela primeira vez no repositório RPM. As recomendações são publicadas no site do [Amazon Linux Security Center](#) assim que as correções são disponibilizadas para instalação por meio do repositório RPM.

A data de atualização da recomendação indica quando novas informações foram adicionadas a uma recomendação após sua última publicação.

Não deve haver nenhuma suposição entre o número da AL2023 versão (por exemplo, 2023.6.20241031) e a data de lançamento do comunicado dos avisos publicados junto com esse lançamento.

Tipos de recomendação

Os metadados do repositório RPM oferecem suporte a recomendações de diferentes tipos. Embora o Amazon Linux tenha emitido quase universalmente apenas recomendações que são atualizações de segurança, não se deve presumir que esse seja sempre o caso. É possível que sejam emitidas recomendações para eventos como correções de bugs, aprimoramentos e novos pacotes, e a recomendação seja marcada como contendo esse tipo de atualização.

Gravidades de recomendações

Cada recomendação tem sua própria gravidade, pois cada problema é avaliado separadamente. Várias CVEs podem ser tratadas em uma única Consultoria, e cada CVE pode ter uma avaliação diferente, mas a Consultoria em si tem uma Severidade. Pode haver várias recomendações referentes a uma única atualização de pacote. Portanto, pode haver várias gravidades para uma atualização de pacote específica (uma por recomendação).

Em ordem decrescente de gravidade, o Amazon Linux usa Crítica, Importante, Moderada e Baixa para indicar a gravidade de uma recomendação. As recomendações do Amazon Linux também podem não ter uma gravidade, embora isso seja extremamente raro.

O Amazon Linux é uma das distribuições do Linux baseadas em RPM que usa o termo Moderada, enquanto outras distribuições do Linux baseadas em RPM usam o termo equivalente Média. O gerenciador de pacotes do Amazon Linux trata ambos os termos como equivalentes, e repositórios de pacotes de terceiros podem usar o termo Média.

As recomendações do Amazon Linux podem mudar de gravidade ao longo do tempo, à medida que se aprende mais sobre as questões relevantes abordadas na recomendação.

A severidade de um aviso normalmente rastreia a pontuação CVSS mais alta avaliada pelo Amazon Linux para aqueles CVEs referenciados pelo aviso. Pode haver casos em que isso não acontece. Um exemplo é quando há um problema resolvido para o qual não há uma CVE atribuída.

Consulte as [perguntas frequentes do ALAS](#) para obter mais informações sobre como o Amazon Linux usa as classificações de gravidade das recomendações.

Recomendações e pacotes

Pode haver muitas recomendações para um único pacote, e nem todos os pacotes terão uma recomendação publicada para eles. Uma versão específica do pacote pode ser referenciada em vários avisos, cada um com sua própria severidade e CVEs.

É possível que vários avisos para a mesma atualização de pacote sejam emitidos simultaneamente em uma nova AL2023 versão ou em rápida sucessão.

Como em outras distribuições do Linux, pode haver um ou vários pacotes binários diferentes criados com base no mesmo pacote de origem. Por exemplo, o [ALAS-2024-698](#) é um aviso listado na seção [AL2023 do site do Amazon Linux Security Center](#) como aplicável ao pacote `mariadb105`. Esse é o nome do pacote de origem, e a recomendação em si se refere aos pacotes binários junto com o pacote de origem. Nesse caso, mais de uma dúzia de pacotes binários são criados com base em um único pacote de origem do `mariadb105`. Embora seja extremamente comum haver um pacote binário com o mesmo nome do pacote de origem, isso nem sempre acontece.

Embora as recomendações do Amazon Linux normalmente listem todos os pacotes binários criados com base no pacote de origem atualizado, não se deve presumir que esse seja sempre o caso. O gerenciador de pacotes e o formato de metadados do repositório RPM permitem recomendações que listam um subconjunto dos pacotes binários atualizados.

Uma recomendação específica também pode se aplicar somente a uma arquitetura de CPU específica. Pode haver pacotes que não foram criados para todas as arquiteturas ou problemas que não afetam todas as arquiteturas. No caso em que um pacote está disponível em todas as arquiteturas, mas um problema se aplica somente a uma, o Amazon Linux normalmente não emite uma recomendação que faz referência a apenas a arquitetura afetada, embora não se deva presumir que esse seja sempre o caso.

Devido à natureza das dependências do pacote, é comum que uma recomendação faça referência a um pacote, mas a instalação dessa atualização exija outras atualizações de pacotes, incluindo pacotes não listados na recomendação. O gerenciador de pacotes dnf cuidará da instalação das dependências necessárias.

Avisos e CVEs

Um Aviso pode abordar zero ou mais CVEs, e pode haver vários Avisos referenciando o mesmo CVE.

Um exemplo de quando uma Consultoria pode fazer referência a zero CVEs é quando um CVE ainda não foi (ou nunca) foi atribuído ao problema.

Um exemplo de onde várias recomendações podem fazer referência à mesma CVE é quando a CVE se aplica a vários pacotes. Por exemplo, a [CVE-2024-21208](#) se aplica ao Corretto 8, 11, 17 e 21. [Cada uma dessas versões do Corretto é um pacote separado e há um aviso para cada um desses pacotes: ALAS-2024-754 para o Corretto 8 AL2023, ALAS-2024-753 para o Corretto 11, ALAS-2024-752 para o Corretto 17 e ALAS-2024-752 para o Corretto 21.](#) Embora todos esses lançamentos do Corretto tenham a mesma lista de CVEs, isso não deve ser assumido.

Uma CVE específica pode ser avaliada de forma diferente para pacotes diferentes. Por exemplo, se uma CVE específica é referenciada em uma recomendação com uma gravidade Importante, é possível que outra recomendação seja emitida referenciando a mesma CVE com uma gravidade diferente.

Os metadados do repositório RPM permitem uma lista de referências para cada recomendação. Embora o Amazon Linux normalmente só faça referências CVEs, o formato de metadados permite outros tipos de referência.

Os metadados do repositório de pacotes RPM só se referirão CVEs com uma correção disponível. A [seção Explore do site do Amazon Linux Security Center](#) contém informações sobre o CVEs que o Amazon Linux avaliou. Essa avaliação pode resultar em uma pontuação básica, gravidade e status do CVSS para várias versões e pacotes do Amazon Linux. O status de uma CVE para uma versão ou um pacote específico do Amazon Linux pode ser Não afetado, Correção pendente ou Nenhuma correção planejada. O status e a avaliação do CVEs podem mudar várias vezes e de qualquer forma antes da emissão de um Aviso. Isso inclui a reavaliação da aplicabilidade de uma CVE ao Amazon Linux.

A lista de CVEs pessoas referenciadas por um Aviso pode mudar após a publicação inicial desse Aviso.

Texto da recomendação

Uma recomendação também conterá um texto descritivo dos problemas que foram o motivo da criação da recomendação. É comum que esse texto seja o texto não modificado da CVE. Esse texto pode se referir aos números de versão upstream em que uma correção está disponível e que são diferentes da versão do pacote à qual o Amazon Linux aplicou uma correção. É comum que o Amazon Linux faça a portabilidade retroativa das correções de versões upstream mais recentes. Caso o texto da recomendação mencione uma versão upstream diferente da versão distribuída em uma versão do Amazon Linux, as versões de pacote do Amazon Linux na recomendação serão precisas para o Amazon Linux.

É possível que o texto da recomendação nos metadados do repositório RPM seja um texto de espaço reservado que apenas faz referência ao site do [Amazon Linux Security Center](https://aws.amazon.com/security/linux-security-center/) para mais detalhes.

Recomendações de patch ao vivo do kernel

As recomendações para patches ao vivo são exclusivas, pois se referem a um pacote diferente (o kernel do Linux) do pacote ao qual a recomendação se refere (por exemplo, `kernel-livepatch-6.1.15-28.43`).

Um aviso para um [Kernel Live Patch](#) fará referência aos problemas (como CVEs) que o pacote específico do Live Patch pode resolver para a versão específica do kernel à qual o pacote de patch ativo se aplica.

Cada patch ao vivo é destinado a uma versão específica do kernel. Para aplicar um patch ao vivo a uma CVE, o pacote de patch ao vivo correto para a versão do kernel precisa ser instalado e o patch ao vivo aplicado.

Por exemplo, o [CVE-2023-6111](#) pode ser corrigido ao vivo para as versões do kernel e. AL2023 6.1.56-82.125 6.1.59-84.139 6.1.61-85.141 Uma nova versão do kernel com uma correção para essa CVE também foi lançada e tem [uma recomendação separada](#). Para que o [CVE-2023-6111](#) seja endereçado em AL2023 uma versão do kernel igual ou posterior à especificada por [ALAS2023-2023-461](#), precisa estar em execução, ou uma das versões do kernel com um patch ativo para esse CVE precisa estar em execução com o livepatch aplicável aplicado.

Quando há novos patches ao vivo disponíveis para uma versão específica do kernel que já tem um patch ao vivo disponível, uma nova versão do pacote `kernel-livepatch-KERNEL_VERSION` é lançada. Por exemplo, o [ALASLIVEPATCH-2023-003](#) Aviso foi emitido com o `kernel-`

livepatch-6.1.15-28.43-1.0-1.amzn2023 pacote que continha patches ativos para o 6.1.15-28.43 kernel, abrangendo três CVEs. Posteriormente, o [ALASLIVEPATCH-2023-009](#) Aviso foi lançado com o kernel-livepatch-6.1.15-28.43-1.0-2.amzn2023 pacote; uma atualização do pacote de patch ativo anterior para o 6.1.15-28.43 kernel contendo patches ativos para outros três CVEs. Também havia outros problemas de recomendações com patch ao vivo para outras versões do kernel, com pacotes contendo patches ao vivo para essas versões específicas do kernel.

Para saber mais sobre a aplicação de patches ao vivo do kernel, consulte [Aplicação de patches do kernel em tempo real no AL2023](#).

Para todos que desenvolvem ferramentas relacionadas a recomendações de segurança, também é recomendável consultar a seção [Esquema XML para recomendações e updateinfo.xml](#) para obter mais informações.

Esquema XML para recomendações e **updateinfo.xml**

O arquivo `updateinfo.xml` faz parte do formato do repositório de pacotes. São os metadados que o gerenciador de pacotes `dnf` analisa para implementar funcionalidades, como [Listar recomendações aplicáveis](#) e [Aplicar atualizações de segurança no local](#).

Recomendamos o uso da API do gerenciador de pacotes `dnf` em vez de escrever código personalizado para analisar os formatos de metadados do repositório. A versão do `dnf` in AL2023 pode analisar os formatos e o do AL2 repositório AL2023 e, portanto, a API pode ser usada para examinar informações consultivas para qualquer versão do sistema operacional.

O projeto [RPM Software Management](#) documenta os formatos de metadados RPM no repositório [rpm-metadata](#) em. GitHub

Para todos que desenvolvem ferramentas para análise direta dos metadados de `updateinfo.xml`, é altamente recomendável prestar muita atenção à [documentação de rpm-metadata](#). A documentação aborda o que já foi visto no mundo real, o que inclui muitas exceções ao que você pode razoavelmente interpretar como uma regra para o formato de metadados.

Também há um conjunto crescente de exemplos reais de `updateinfo.xml` arquivos no repositório [raw-historical-rpm-repository-examples](#) em. GitHub

Caso algo não esteja claro na documentação, você pode abrir um problema no GitHub projeto para que possamos responder à pergunta e atualizar a documentação adequadamente. Como se tratam

de projetos de código aberto, solicitações pull para atualizar a documentação também são bem-vindas.

Listar recomendações aplicáveis

O gerenciador de pacotes `dnf` tem acesso aos metadados que descrevem quais recomendações foram corrigidas em quais versões de pacotes. Assim, ele pode listar quais recomendações são aplicáveis a uma instância ou imagem de contêiner.

Note

Ferramentas como o [AWS Systems Manager](#) podem usar essa funcionalidade para mostrar quais atualizações são relevantes em uma frota, em vez de apenas em uma única instância.

Ao listar as atualizações, você pode instruir o `dnf` a examinar os metadados de um lançamento específico do AL2023 ou os metadados do lançamento mais recente.

Note

Depois que um lançamento do AL2023 é distribuído, ele é imutável. Assim, recomendações novas ou atualizadas no [Amazon Linux Security Center](#) são adicionadas somente aos metadados de novos lançamentos do AL2023.

Agora, veremos exemplos de como conferir quais recomendações se aplicam a algumas imagens de contêiner do AL2023. Todos esses comandos funcionam em ambientes sem contêineres, como instâncias do EC2.

Listing advisories in a specific version

Neste exemplo, veremos quais recomendações do lançamento [2023.1.20230628](#) são relevantes para uma imagem de contêiner do lançamento [2023.0.20230315](#).

Note

Este exemplo usa os lançamentos [2023.0.20230315](#) e [2023.1.20230628](#), e esses não são os lançamentos mais recentes do AL2023. Consulte [AL2023 Release Notes](#) para

conferir as versões mais recentes, que contêm as atualizações de segurança mais recentes.

Neste exemplo, começaremos com uma imagem de contêiner para o lançamento [2023.0.20230315](#).

Primeiro, buscamos essa imagem de contêiner no registro de contêineres. O trecho `.0` no final indica a versão da imagem para um lançamento específico, e essa versão da imagem geralmente é 0.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
2023.0.20230315.0: Pulling from amazonlinux/amazonlinux
b76f3b09316a: Pull complete
Digest: sha256:94e7183b0739140dbd5b639fb7600f0a2299cec5df8780c26d9cb409da5315a9
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.0.20230315.0
public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
```

Agora podemos gerar um shell dentro do contêiner, do qual solicitaremos que `dnf` liste as recomendações relevantes para os pacotes instalados no contêiner.

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
bash-5.2#
```

O comando `dnf updateinfo` agora é usado para exibir um resumo de quais recomendações do lançamento [2023.1.20230628](#) são relevantes para nossos pacotes instalados.

```
$ dnf updateinfo --releasever=2023.1.20230628
Amazon Linux 2023 repository          42 MB/s | 15 MB      00:00
Last metadata expiration check: 0:00:02 ago on Mon Jul 22 20:24:24 2024.
Updates Information Summary: available
  8 Security notice(s)
    1 Important Security notice(s)
    5 Medium Security notice(s)
    2 Low Security notice(s)
```

Para obter uma lista de recomendações, a opção `--list` pode ser concedida a `dnf updateinfo`.

```
$ dnf updateinfo --releasever=2023.1.20230628 --list
Last metadata expiration check: 0:01:22 ago on Mon Jul 22 20:24:24 2024.
ALAS2023-2023-193 Medium/Sec.      curl-minimal-8.0.1-1.amzn2023.x86_64
ALAS2023-2023-225 Medium/Sec.      glib2-2.74.7-688.amzn2023.0.1.x86_64
ALAS2023-2023-195 Low/Sec.         libcap-2.48-2.amzn2023.0.3.x86_64
ALAS2023-2023-193 Medium/Sec.      libcurl-minimal-8.0.1-1.amzn2023.x86_64
ALAS2023-2023-145 Low/Sec.         libgcc-11.3.1-4.amzn2023.0.3.x86_64
ALAS2023-2023-145 Low/Sec.         libgomp-11.3.1-4.amzn2023.0.3.x86_64
ALAS2023-2023-145 Low/Sec.         libstdc++-11.3.1-4.amzn2023.0.3.x86_64
ALAS2023-2023-163 Medium/Sec.      libxml2-2.10.4-1.amzn2023.0.1.x86_64
ALAS2023-2023-220 Important/Sec.    ncurses-base-6.2-4.20200222.amzn2023.0.4.noarch
ALAS2023-2023-220 Important/Sec.    ncurses-libs-6.2-4.20200222.amzn2023.0.4.x86_64
ALAS2023-2023-181 Medium/Sec.      openssl-libs-1:3.0.8-1.amzn2023.0.2.x86_64
ALAS2023-2023-222 Medium/Sec.      openssl-libs-1:3.0.8-1.amzn2023.0.3.x86_64
```

Listing advisories in the latest version

Neste exemplo, veremos quais atualizações estão disponíveis na versão latest do AL2023 ao inicializar um contêiner do lançamento [2023.4.20240319](#). No momento em que este artigo foi escrito, o lançamento latest era [2023.5.20240708](#), portanto, as atualizações listadas neste exemplo serão as desse lançamento.

Note

Este exemplo usa os lançamentos [2023.4.20240319](#) e [2023.5.20240708](#), sendo que este é o lançamento mais recente no momento em que este artigo foi escrito. Para obter mais informações sobre os lançamentos mais recentes, consulte [AL2023 Release Notes](#).

Neste exemplo, começaremos com uma imagem de contêiner para o lançamento [2023.4.20240319](#).

Primeiro, buscamos essa imagem de contêiner no registro de contêineres. O trecho `.1` no final indica a versão da imagem para um lançamento específico. Embora a versão da imagem geralmente seja 0, este exemplo usa um lançamento em que a versão da imagem é 1.

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
2023.4.20240319.1: Pulling from amazonlinux/amazonlinux
6de065fda9a2: Pull complete
Digest: sha256:b4838c4cc9211d966b6ea158dacc9eda7433a16ba94436508c2d9f01f7658b4e
```

```
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.4.20240319.1
public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
```

Agora podemos gerar um shell dentro do contêiner, por meio do qual verificaremos se há atualizações.

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
bash-5.2#
```

O comando `dnf updateinfo` agora é usado para exibir um resumo de quais recomendações do lançamento mais recente são relevantes para nossos pacotes instalados. No momento em que este artigo foi escrito, [2023.1.20230628](#) era o lançamento mais recente.

```
$ dnf --releasever=latest updateinfo
Amazon Linux 2023 repository                76 MB/s | 25 MB    00:00
Last metadata expiration check: 0:00:04 ago on Mon Jul 22 20:59:54 2024.
Updates Information Summary: available
  9 Security notice(s)
    4 Important Security notice(s)
    4 Medium Security notice(s)
    1 Low Security notice(s)
```

Para obter uma lista de recomendações, a opção `--list` pode ser concedida a `dnf updateinfo`.

```
$ dnf updateinfo --releasever=latest --list
Last metadata expiration check: 0:00:58 ago on Mon Jul 22 20:59:54 2024.
ALAS2023-2024-581 Low/Sec.      curl-minimal-8.5.0-1.amzn2023.0.3.x86_64
ALAS2023-2024-596 Medium/Sec.  curl-minimal-8.5.0-1.amzn2023.0.4.x86_64
ALAS2023-2024-576 Important/Sec. expat-2.5.0-1.amzn2023.0.4.x86_64
ALAS2023-2024-589 Important/Sec. glibc-2.34-52.amzn2023.0.10.x86_64
ALAS2023-2024-589 Important/Sec. glibc-common-2.34-52.amzn2023.0.10.x86_64
ALAS2023-2024-589 Important/Sec. glibc-minimal-langpack-2.34-52.amzn2023.0.10.x86_64
ALAS2023-2024-586 Medium/Sec.  krb5-libs-1.21-3.amzn2023.0.4.x86_64
ALAS2023-2024-581 Low/Sec.      libcurl-minimal-8.5.0-1.amzn2023.0.3.x86_64
ALAS2023-2024-596 Medium/Sec.  libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64
ALAS2023-2024-592 Important/Sec. libnghttp2-1.59.0-3.amzn2023.0.1.x86_64
ALAS2023-2024-640 Medium/Sec.  openssl-libs-1:3.0.8-1.amzn2023.0.12.x86_64
ALAS2023-2024-605 Medium/Sec.  python3-3.9.16-1.amzn2023.0.7.x86_64
ALAS2023-2024-616 Important/Sec. python3-3.9.16-1.amzn2023.0.8.x86_64
```

ALAS2023-2024-605	Medium/Sec.	python3-libs-3.9.16-1.amzn2023.0.7.x86_64
ALAS2023-2024-616	Important/Sec.	python3-libs-3.9.16-1.amzn2023.0.8.x86_64

Aplicar atualizações de segurança no local

Para obter uma visão geral da aplicação de atualizações, consulte [Aplicar atualizações de segurança usando DNF e versões do repositório](#). A opção `--security` para `dnf upgrade` restringirá as atualizações de pacotes somente àquelas que tiverem uma recomendação. O restante desta seção abordará como instalar somente atualizações de segurança específicas.

 Note

É recomendável aplicar todas as atualizações disponíveis em um novo lançamento do AL2023. Escolher apenas atualizações de segurança ou apenas atualizações específicas deve ser a exceção, e não a regra.

Aplicar as atualizações mencionadas em uma recomendação

Os identificadores de recomendação na primeira coluna da saída de `dnf upgradeinfo` podem ser usados para aplicar atualizações para os pacotes mencionados na recomendação. O gerenciador de pacotes `dnf` pode ser instruído a atualizar os pacotes na recomendação para as versões mais recentes disponíveis ou somente até as versões mencionadas na recomendação. Se as atualizações já estiverem instaladas, o comando de atualização não funcionará.

Para aplicar as atualizações aos pacotes afetados somente até a versão mencionada na recomendação, use o comando `dnf upgrade-minimal` enquanto usa a opção `--advisory` para especificar a recomendação. O exemplo a seguir está executado `dnf upgrade-minimal` em um contêiner do AL2023 versão [2023.0.20230315](#).

```
$ dnf upgrade-minimal -y --releasever=2023.1.20230628 --advisory ALAS2023-2023-193
Amazon Linux 2023 repository                               46 MB/s | 15 MB      00:00
Last metadata expiration check: 0:00:03 ago on Mon Jul 22 20:36:13 2024.
Dependencies resolved.
=====
Package                Arch      Version                Repository             Size
=====
Upgrading:
```

```
curl-minimal      x86_64      8.0.1-1.amzn2023      amazonlinux      150 k
libcurl-minimal   x86_64      8.0.1-1.amzn2023      amazonlinux      249 k

Transaction Summary
=====
Upgrade 2 Packages

Total download size: 399 k
Downloading Packages:
(1/2): curl-minimal-8.0.1-1.amzn2023.x86_64.rpm 2.7 MB/s | 150 kB      00:00
(2/2): libcurl-minimal-8.0.1-1.amzn2023.x86_64. 3.8 MB/s | 249 kB      00:00
-----
Total                               2.5 MB/s | 399 kB      00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :                               1/1
  Upgrading      : libcurl-minimal-8.0.1-1.amzn2023.x86_64 1/4
  Upgrading      : curl-minimal-8.0.1-1.amzn2023.x86_64    2/4
  Cleanup        : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64 3/4
  Cleanup        : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64 4/4
  Running scriptlet: libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64 4/4
  Verifying      : libcurl-minimal-8.0.1-1.amzn2023.x86_64 1/4
  Verifying      : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64 2/4
  Verifying      : curl-minimal-8.0.1-1.amzn2023.x86_64    3/4
  Verifying      : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64 4/4

Upgraded:
  curl-minimal-8.0.1-1.amzn2023.x86_64  libcurl-minimal-8.0.1-1.amzn2023.x86_64

Complete!
```

As mesmas versões do pacote são atualizadas mesmo que `--releasever=latest` seja usado, pois a solicitação é para que `dnf` faça a atualização mínima necessária para resolver a recomendação.

Usar o comando `dnf upgrade` regular com a opção `--advisory` atualizará os pacotes relevantes mencionados na recomendação para a versão mais recente disponível, que pode ser mais recente do que a versão mencionada na recomendação.

 Note

A menos que o pacote `system-release` esteja atualizado, a versão dos repositórios do AL2023 à qual `dnf` está vinculado não muda.

 Warning

Ao instalar atualizações de uma versão diferente do AL2023 sem alterar a versão do repositório à qual `dnf` está vinculado, deve-se tomar cuidado com qualquer operação de mutação do `dnf` subsequente. Por exemplo, ao instalar ou atualizar pacotes, como as dependências do pacote podem ter sido alteradas no lançamento mais recente, o lançamento mais antigo que você continua usando pode não ser capaz de satisfazer essas novas dependências.

O exemplo a seguir é executado em um contêiner do AL2023 versão [2023.0.20230315](#) referente à versão mais recente do AL2023 que, no momento em que este artigo foi escrito, era [2023.5.20240708](#). Observe que a versão de `curl` para a qual está sendo atualizado é mais recente do que a versão de `update-minimal` atualizada, mas que essa versão mais recente traz novas dependências.

```
$ dnf upgrade -y --releasever=latest --advisory ALAS2023-2023-193
Amazon Linux 2023 repository                80 MB/s | 25 MB      00:00
Last metadata expiration check: 0:00:04 ago on Mon Jul 22 20:48:38 2024.
Dependencies resolved.
=====
Package                                Arch      Version                                Repository      Size
=====
Upgrading:
curl-minimal                           x86_64    8.5.0-1.amzn2023.0.4                  amazonlinux     160 k
libcurl-minimal                         x86_64    8.5.0-1.amzn2023.0.4                  amazonlinux     275 k
libnghttp2                              x86_64    1.59.0-3.amzn2023.0.1                 amazonlinux      79 k
Installing dependencies:
libpsl                                  x86_64    0.21.1-3.amzn2023.0.2                 amazonlinux      61 k
publicsuffix-list-dafsa                 noarch    20240212-61.amzn2023                  amazonlinux      59 k

Transaction Summary
=====
Install 2 Packages
```

Upgrade 3 Packages

Total download size: 634 k

Downloading Packages:

```
(1/5): publicsuffix-list-dafsa-20240212-61.amzn 1.1 MB/s | 59 kB      00:00
(2/5): curl-minimal-8.5.0-1.amzn2023.0.4.x86_64 2.6 MB/s | 160 kB    00:00
(3/5): libpsl-0.21.1-3.amzn2023.0.2.x86_64.rpm 949 kB/s | 61 kB      00:00
(4/5): libnghttp2-1.59.0-3.amzn2023.0.1.x86_64. 3.7 MB/s | 79 kB      00:00
(5/5): libcurl-minimal-8.5.0-1.amzn2023.0.4.x86 6.7 MB/s | 275 kB    00:00
```

```
-----
Total                               3.5 MB/s | 634 kB      00:00
```

Running transaction check

Transaction check succeeded.

Running transaction test

Transaction test succeeded.

Running transaction

```
Preparing      :                               1/1
Upgrading      : libnghttp2-1.59.0-3.amzn2023.0.1.x86_64      1/8
Installing     : publicsuffix-list-dafsa-20240212-61.amzn2023.noarch 2/8
Installing     : libpsl-0.21.1-3.amzn2023.0.2.x86_64        3/8
Upgrading      : libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64 4/8
Upgrading      : curl-minimal-8.5.0-1.amzn2023.0.4.x86_64    5/8
Cleanup        : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64   6/8
Cleanup        : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64 7/8
Cleanup        : libnghttp2-1.51.0-1.amzn2023.x86_64         8/8
Running scriptlet: libnghttp2-1.51.0-1.amzn2023.x86_64       8/8
Verifying      : libpsl-0.21.1-3.amzn2023.0.2.x86_64        1/8
Verifying      : publicsuffix-list-dafsa-20240212-61.amzn2023.noarch 2/8
Verifying      : curl-minimal-8.5.0-1.amzn2023.0.4.x86_64    3/8
Verifying      : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64   4/8
Verifying      : libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64 5/8
Verifying      : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64 6/8
Verifying      : libnghttp2-1.59.0-3.amzn2023.0.1.x86_64     7/8
Verifying      : libnghttp2-1.51.0-1.amzn2023.x86_64        8/8
```

Upgraded:

```
curl-minimal-8.5.0-1.amzn2023.0.4.x86_64
libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64
libnghttp2-1.59.0-3.amzn2023.0.1.x86_64
```

Installed:

```
libpsl-0.21.1-3.amzn2023.0.2.x86_64
publicsuffix-list-dafsa-20240212-61.amzn2023.noarch
```


Complete!

Configurando modos do SELinux para o AL2023

Por padrão, o Security Enhanced Linux (SELinux) está `enabled` e configurado para o modo `permissive` para o AL2023. No modo `permissive`, as negações de permissão são registradas, mas não aplicadas. O SELinux é uma coleção de recursos e utilitários do kernel para fornecer uma arquitetura de controle de acesso (MAC) forte, flexível e obrigatória aos principais subsistemas do kernel.

O SELinux fornece um mecanismo aprimorado para impor a separação de informações com base nos requisitos de confidencialidade e integridade. Essa separação de informações reduz as ameaças de adulteração e desvio dos mecanismos de segurança do aplicativo. Também limita os danos que podem ser causados por aplicativos maliciosos ou defeituosos.

O SELinux inclui um conjunto de exemplos de arquivos de configuração de políticas de segurança projetados para atender às metas diárias de segurança.

Para obter mais informações sobre os recursos e funcionalidades do SELinux, consulte [Cadernos do SELinux](#) e [Idiomas da Política](#).

Tópicos

- [Status e modos padrão do SELinux para o AL2023](#)
- [Mudar para o modo enforcing](#)
- [Opção para desabilitar o SELinux para o AL2023](#)

Status e modos padrão do SELinux para o AL2023

Para o AL2023, o SELinux vem por padrão como `enabled` e configurado no modo `permissive`. No modo `permissive`, as negações de permissão são registradas, mas não aplicadas.

Os comandos **getenforce** ou **sestatus** lhe dizem o atual status, política e modo do SELinux.

Com o status padrão definido como `enabled` e `permissive`, o comando **getenforce** retorna `permissive`.

O comando **sestatus** retorna o status do SELinux e a política atual do SELinux, conforme mostrado no seguinte exemplo:

```
$ sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            targeted
Current mode:                  permissive
Mode from config file:         permissive
Policy MLS status:             enabled
Policy deny_unknown status:    allowed
Memory protection checking:    actual (secure)
Max kernel policy version:     33
```

Quando você executa o SELinux no modo **permissive**, os usuários podem rotular os arquivos incorretamente. Quando você executa o SELinux no status **disabled**, os arquivos não são rotulados. Arquivos incorretos ou não identificados podem causar problemas quando você muda para o modo **enforcing**.

O SELinux renomeia automaticamente os arquivos para evitar esse problema. O SELinux evita problemas de etiquetagem com a nova rotulagem automática quando você altera o status para **enabled**.

Mudar para o modo **enforcing**

Quando você executa SELinux no modo **enforcing**, o utilitário SELinux está **enforcing** a política configurada. O SELinux controla os recursos de aplicações selecionadas, permitindo ou negando acesso com base nas regras da política.

Para encontrar o modo atual do SELinux, execute o comando `getenforce`.

```
getenforce
Permissive
```

Edite o arquivo de configuração para ativar o modo **enforcing**

Para alterar o modo para **enforcing**, siga as etapas a seguir.

1. Edite o arquivo **enforcing** para mudar para o modo `/etc/selinux/config`. A configuração de SELINUX deve ser semelhante ao exemplo a seguir.

```
SELINUX=enforcing
```

2. Reinicie o sistema para concluir a mudança para o modo enforcing.

```
$ sudo reboot
```

Na próxima inicialização, o SELinux renomeará todos os arquivos e diretórios no sistema. O SELinux também adiciona o contexto do SELinux para arquivos e diretórios que foram criados quando o SELinux estava disabled.

Depois de mudar para o modo enforcing, o SELinux pode negar algumas ações devido a regras de política do SELinux incorretas ou ausentes. Você pode visualizar as ações que o SELinux nega com o comando a seguir.

```
$ sudo ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR -ts recent
```

Use cloud-init para ativar o modo **enforcing**.

Como alternativa, ao iniciar sua instância, transmita o seguinte cloud-config como dados do usuário para ativar o modo enforcing.

```
#cloud-config
selinux:
  mode: enforcing
```

Por padrão, essa configuração faz com que a instância seja reinicializada. Para maior estabilidade, recomendamos reinicializar sua instância. Entretanto, se você preferir, poderá pular a reinicialização fornecendo o cloud-config a seguir.

```
#cloud-config
selinux:
  mode: enforcing
  selinux_no_reboot: 1
```

Opção para desabilitar o SELinux para o AL2023

Quando você desabilita o SELinux, a política do SELinux não é carregada nem imposta e as mensagens do Access Vector Cache (AVC) não são registradas em log. Você perde todos os benefícios de executar o SELinux.

Em vez de desabilitar o SELinux, recomendamos usar o modo **permissive**. Custa apenas um pouco mais executar no modo **permissive** do que desabilitar completamente o SELinux. A transição do modo **permissive** para o modo **enforcing** requer muito menos ajustes de configuração do que a transição de volta ao modo **enforcing** após desabilitar o SELinux. Você pode rotular arquivos e o sistema pode rastrear e registrar ações que a política ativa possa ter negado.

Mudar o SELinux para o modo **permissive**

Quando você executa o SELinux no modo **permissive**, a política do SELinux não é imposta. No modo **permissive**, o SELinux registra as mensagens de AVC em log, mas não nega as operações. Você pode usar essas mensagens de AVC para solucionar problemas, depurar e aprimorar a política do SELinux.

Para mudar o SELinux para o modo permissivo, siga as etapas a seguir.

1. Edite o arquivo **permissive** para mudar para o modo `/etc/selinux/config`. O valor de **SELINUX** deve ser como o exemplo a seguir.

```
SELINUX=permissive
```

2. Reinicie o sistema para concluir a mudança para o modo **permissive**.

```
sudo reboot
```

Desabilitar SELinux

Ao desabilitar o SELinux, a política do SELinux não é carregada nem imposta, e as mensagens de AVC não são registradas em log. Você perde todos os benefícios de executar o SELinux.

Para desabilitar o SELinux, siga as etapas a seguir.

1. Garanta que o pacote **grubby** esteja instalado.

```
rpm -q grubby  
grubby-version
```

2. Configure seu bootloader para adicionar `selinux=0` à linha de comando do kernel.

```
sudo grubby --update-kernel ALL --args selinux=0
```

3. Reinicie o sistema.

```
sudo reboot
```

4. Execute o comando `getenforce` para confirmar se o SELinux é Disabled.

```
$ getenforce
Disabled
```

Para obter mais informações sobre o SELinux, consulte [SELinux Notebook](#) e [SELinux configuration](#).

Habilitar o modo FIPS no AL2023

Esta seção explica como habilitar o Federal Information Processing Standards (FIPS) no AL2023. Para obter mais informações sobre FIPS, consulte:

- [Federal Information Processing Standard \(FIPS\)](#)
- [Perguntas frequentes sobre conformidade: Federal Information Processing Standards](#)

Note

Esta seção documenta como ativar o modo FIPS no AL2023, mas não abrange o status da certificação dos módulos criptográficos do AL2023.

Pré-requisitos

- Uma instância AL2023 (AL2023.2 ou superior) existente do Amazon EC2 com acesso à Internet para baixar os pacotes necessários. Para obter mais informações sobre como iniciar uma instância de AL2023 Amazon EC2, consulte [Inicializar o AL2023 usando o console do Amazon EC2](#).
- Você deve se conectar à sua instância do Amazon EC2 usando SSH ou AWS Systems Manager. Para obter mais informações, consulte [Conexão com AL2023 instâncias](#).

⚠ Important

As chaves de usuário SSH ED25519 não são compatíveis no modo FIPS. Se você lançou sua instância do Amazon EC2 usando um par de chaves SSH ED25519, deverá gerar novas chaves usando outro algoritmo (como RSA) ou poderá perder o acesso à sua instância após ativar o modo FIPS. Para obter mais informações, consulte [Criar um par de chaves](#) no Guia do usuário do Amazon EC2.

Habilitar o modo FIPS

1. Conecte-se à sua instância do AL2023 usando SSH ou AWS Systems Manager.
2. Verifique se o sistema está atualizado. Para obter mais informações, consulte [Gerencie atualizações de pacotes e sistemas operacionais no AL2023](#).
3. Verifique se os utilitários `crypto-policies` estão instalados e atualizados.

```
sudo dnf -y install crypto-policies crypto-policies-scripts
```

4. Ativar modo FIPS executando o seguinte comando. Isso habilitará o modo FIPS em todo o sistema para os módulos listados em [AL2023 FAQ](#).

```
sudo fips-mode-setup --enable
```

5. Execute a instância usando o seguinte comando.

```
sudo reboot
```

6. Para verificar se o modo do FIPS está habilitado, reconecte-se à sua instância e execute o comando a seguir.

```
sudo fips-mode-setup --check
```

O exemplo de saída a seguir mostra que o modo do FIPS está habilitado:

```
FIPS mode is enabled.
```

Habilitar o modo FIPS em um contêiner do AL2023

Esta seção explica como habilitar o Federal Information Processing Standards (FIPS) em um contêiner do AL2023. Para obter mais informações sobre FIPS, consulte:

- [Federal Information Processing Standard \(FIPS\)](#)
- [Perguntas frequentes sobre conformidade: Federal Information Processing Standards](#)

Note

Esta seção documenta como habilitar o modo FIPS em um contêiner do AL2023. Ela não abrange o status de certificação dos módulos criptográficos do AL2023.

Pré-requisitos

- Uma instância AL2023 (AL2023.2 ou superior) existente do Amazon EC2 com acesso à Internet para baixar os pacotes necessários. Para obter mais informações sobre como iniciar uma instância de AL2023 Amazon EC2, consulte [Inicializar o AL2023 usando o console do Amazon EC2](#).
- Você deve se conectar à sua instância do Amazon EC2 usando SSH ou AWS Systems Manager. Para obter mais informações, consulte [Conexão com AL2023 instâncias](#).

Important

O comando `fips-mode-setup` não funcionará corretamente de dentro do contêiner. Leia as etapas abaixo para configurar corretamente o modo FIPS em um contêiner do AL2023.

Habilitar o modo FIPS em um contêiner do AL2023

1. O modo FIPS deve primeiro ser habilitado no host do contêiner do AL2023. Siga as instruções em [Habilitar o modo FIPS no AL2023](#) para habilitar o modo FIPS no host.
2. Conecte-se à instância de host do contêiner do AL2023 usando SSH ou AWS Systems Manager.
3. O modo FIPS será habilitado automaticamente em um contêiner do AL2023 se o host do AL2023 estiver no modo FIPS e `/proc/sys/crypto/fips_enabled` puder ser acessado de dentro

do contêiner. Se o conteúdo de `/proc/sys/crypto/fips_enabled` é `0`, o FIPS não está habilitado, enquanto um valor de `1` indica que o modo FIPS está habilitado.

Você pode verificar se o FIPS está habilitado executando o seguinte comando no host do AL2023 e no contêiner:

```
cat /proc/sys/crypto/fips_enabled
```

4. Depois, habilite as políticas de criptografia do FIPS dentro do contêiner. Há várias maneiras de fazer isso, descritas nas opções abaixo. Use a opção mais adequada para o seu ambiente.

a. Habilite as políticas de criptografia do FIPS manualmente dentro do contêiner usando o comando `update-crypto-policies`:

```
# Run these commands inside the container
dnf install -y crypto-policies-scripts
update-crypto-policies --set FIPS
```

b. Crie montagens `bind` no contêiner do AL2023 (isso é semelhante ao funcionamento de `podman` em outras distribuições):

```
# Run these commands inside the container
mount --bind /usr/share/crypto-policies/back-ends/FIPS /etc/crypto-policies/back-ends
echo "FIPS" > /usr/share/crypto-policies/default-fips-config
mount --bind /usr/share/crypto-policies/default-fips-config /etc/crypto-policies/config
```

c. Também é possível criar uma montagem vinculada para que o contêiner do AL2023 corresponda às políticas de criptografia do host do AL2023. O conteúdo a seguir é apenas um exemplo. Essa configuração pode causar problemas se houver diferenças incompatíveis nas políticas de criptografia e nas versões de pacotes entre o contêiner e o host:

```
sudo docker pull amazonlinux:2023
sudo docker run --mount type=bind,readonly,src=/etc/crypto-policies,dst=/etc/crypto-policies -it amazonlinux:2023
```

5. Depois de executar as etapas acima, você pode verificar novamente se o FIPS está habilitado no contêiner com os seguintes comandos:


```
$ cat /etc/crypto-policies/config
FIPS

$ cat /proc/sys/crypto/fips_enabled
1
```

Trocar de provedor de OpenSSL FIPS no AL2023

Esta seção explica como alternar entre provedores de OpenSSL FIPS latest e certified no AL2023.

Para obter mais informações sobre FIPS, consulte:

- [Federal Information Processing Standard \(FIPS\)](#)
- [Perguntas frequentes sobre conformidade: Federal Information Processing Standards](#)
- [FedRAMP Policy for Cryptographic Module Selection and Use](#)

Important

No AL2023.7 e posterior, o provedor de OpenSSL FIPS padrão é o pacote `openssl-fips-provider-latest`, que recebe correções de bugs e atualizações de segurança regulares. As instruções abaixo são apenas para clientes que desejam fixar o pacote `openssl-fips-provider-certified`. Essa versão do provedor de FIPS corresponderá à soma de verificação do certificado NIST e pode não ter as atualizações mais recentes.

Consulte [AL2023 FAQ](#) para obter mais informações sobre módulos certificados para FIPS e versões de pacotes.

Pré-requisitos

- Uma instância AL2023 (AL2023.7 ou superior) existente do Amazon EC2 com acesso à Internet para baixar os pacotes necessários. Para obter mais informações sobre como iniciar uma instância de AL2023 Amazon EC2, consulte [Inicializar o AL2023 usando o console do Amazon EC2](#).
- Você deve se conectar à sua instância do Amazon EC2 usando SSH ou AWS Systems Manager. Para obter mais informações, consulte [Conexão com AL2023 instâncias](#).
- Para habilitar o modo FIPS no AL2023, siga as instruções em [Habilitar o modo FIPS no AL2023](#).

Alternar entre **openssl-fips-provider-latest** e **openssl-fips-provider-certified**

1. Use `dnf` para trocar o provedor de OpenSSL FIPS:

```
sudo dnf -y swap openssl-fips-provider-latest openssl-fips-provider-certified
```

2. Verifique se você está usando o provedor de OpenSSL FIPS certificado. Com o AL2023 no modo FIPS, execute o seguinte comando:

```
openssl list -providers
```

A seguinte saída deverá ser mostrada:

```
Providers:
  base
    name: OpenSSL Base Provider
    version: 3.2.2
    status: active
  default
    name: OpenSSL Default Provider
    version: 3.2.2
    status: active
  fips
    name: Amazon Linux 2023 - OpenSSL FIPS Provider
    version: 3.0.8-d694bfa693b76001
    status: active
```

AL2023 Endurecimento do kernel

O kernel Linux 6.1 AL2023 é configurado e construído com várias opções e recursos de fortalecimento.

Opções de fortalecimento do kernel (independente da arquitetura)

Opção do CONFIG	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_ACPI_CUSTOM_METHOD</u>	n	n	N/D	N/D
<u>CONFIG_BINFMT_MISC</u>	m	m	m	m
<u>CONFIG_BUG</u>	y	y	y	y
<u>CONFIG_BUG_ON_DATA_CORRUPTION</u>	y	y	y	y
<u>CONFIG_CLANG_I</u>	N/D	N/D	N/D	N/D
<u>CONFIG_CLANG_I_PERMISSIVE</u>	N/D	N/D	N/D	N/D
<u>CONFIG_COMPAT</u>	y	y	y	y
<u>CONFIG_COMPAT_BRK</u>	n	n	n	n
<u>CONFIG_COMPAT_VDSO</u>	N/D	n	N/D	n
<u>CONFIG_DEBUG_CREDENTIALS</u>	n	n	N/D	N/D

Opção do CONFIG	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_DE BUG_LIST</u>	y	y	y	y
<u>CONFIG_DE BUG_NOTIF IERS</u>	n	n	n	n
<u>CONFIG_DE BUG_SG</u>	n	n	n	n
<u>CONFIG_DE BUG_VIRTU AL</u>	n	n	n	n
<u>CONFIG_DE BUG_WX</u>	n	n	n	n
<u>CONFIG_DE FAULT_MMA P_MIN_ADDR</u>	65536	65536	65536	65536
<u>CONFIG_DE VKMEM</u>	N/D	N/D	N/D	N/D
<u>CONFIG_DE VMEM</u>	n	n	n	n
<u>CONFIG_EF I_DISABLE _PCI_DMA</u>	n	n	n	n
<u>CONFIG_F0 RTIFY_SOU RCE</u>	y	y	y	y

Opção do CONFIG	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_HARDENED_USERCOPY</u>	y	y	y	y
<u>CONFIG_HARDENED_USERCOPY_FALLBACK</u>	N/D	N/D	N/D	N/D
<u>CONFIG_HARDENED_USERCOPY_PAGESPAN</u>	N/D	N/D	N/D	N/D
<u>CONFIG_HIBERNATION</u>	y	y	y	y
<u>CONFIG_HW_RANDOM_TPM</u>	N/D	N/D	N/D	N/D
<u>CONFIG_INET_DIAG</u>	m	m	m	m
<u>CONFIG_INET_ON_ALL_OSC_DEFAULT_ON</u>	n	n	n	n
<u>CONFIG_INET_ON_FREE_DEFAULT_ON</u>	n	n	n	n

Opção do CONFIG	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_IN IT_STACK_ ALL_ZERO</u>	N/D	N/D	N/D	N/D
<u>CONFIG_IO MMU_DEFAU LT_DMA_ST RICT</u>	n	n	n	n
<u>CONFIG_IO MMU_SUPPO RT</u>	y	y	y	y
<u>CONFIG_IO _STRICT_D EVMEM</u>	N/D	N/D	N/D	N/D
<u>CONFIG_KE XEC</u>	y	y	y	y
<u>CONFIG_KF ENCE</u>	n	n	n	n
<u>CONFIG_LD ISC_AUTOL OAD</u>	n	n	n	n
<u>CONFIG_LE GACY_PTYS</u>	n	n	n	n
<u>CONFIG_LO CK_DOWN_K ERNEL_FOR CE_CONFID ENTIALITY</u>	n	n	n	n

Opção do CONFIG	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_MODULES</u>	y	y	y	y
<u>CONFIG_MODULE_SIG</u>	y	y	y	y
<u>CONFIG_MODULE_SIG_ALL</u>	y	y	y	y
<u>CONFIG_MODULE_SIG_FORCE</u>	n	n	n	n
<u>CONFIG_MODULE_SIG_HASH</u>	sha512	sha512	sha512	sha512
<u>CONFIG_MODULE_SIG_KEY</u>	certs/signing_key.pem	certs/signing_key.pem	certs/signing_key.pem	certs/signing_key.pem
<u>CONFIG_MODULE_SIG_SHA512</u>	y	y	y	y
<u>CONFIG_PAGE_POISONING</u>	n	n	n	n
<u>CONFIG_PAGE_POISONING_NO_SANITY</u>	N/D	N/D	N/D	N/D

Opção do CONFIG	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_PAGE_POISONING_ZERO</u>	N/D	N/D	N/D	N/D
<u>CONFIG_PANIC_ON_OOPS</u>	y	y	y	y
<u>CONFIG_PANIC_TIMEOUT</u>	0	0	0	0
<u>CONFIG_PRINTK_CORE</u>	y	y	y	y
<u>CONFIG_RANDOMIZE_KERNEL_STACK_OFFSET_DEFAULT</u>	n	n	n	n
<u>CONFIG_RANDOMIZE_TRUST_BOOTLOADER</u>	y	y	N/D	N/D
<u>CONFIG_RANDOMIZE_TRUST_CPU</u>	y	y	N/D	N/D
<u>CONFIG_REFCOUNT_FULL</u>	N/D	N/D	N/D	N/D

Opção do CONFIG	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_SC HED_CORE</u>	N/D	y	N/D	y
<u>CONFIG_SC HED_STACK _END_CHECK</u>	y	y	y	y
<u>CONFIG_SE CCOMP</u>	y	y	y	y
<u>CONFIG_SE CCOMP_FIL TER</u>	y	y	y	y
<u>CONFIG_SE CURITY</u>	y	y	y	y
<u>CONFIG_SE CURITY_DM ESG_RESTR ICT</u>	y	y	y	y
<u>CONFIG_SE CURITY_LA NDLOCK</u>	y	y	y	y
<u>CONFIG_SE CURITY_LO CKDOWN_LSM</u>	y	y	y	y
<u>CONFIG_SE CURITY_LO CKDOWN_LS M_EARLY</u>	y	y	y	y

Opção do CONFIG	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_SECURITY_SELINUX_BOOTPARAM</u>	y	y	y	y
<u>CONFIG_SECURITY_SELINUX_DEVELOP</u>	y	y	y	y
<u>CONFIG_SECURITY_SELINUX_DISABLE</u>	n	n	N/D	N/D
<u>CONFIG_SECURITY_SELINUX_ITABLE_HOOKS</u>	N/D	N/D	N/D	N/D
<u>CONFIG_SECURITY_SELINUX_YAMA</u>	y	y	y	y
<u>CONFIG_SHUFFLE_PAGE_ALLOCATOR</u>	y	y	y	y
<u>CONFIG_SLAB_FREELIST_HARDENED</u>	y	y	y	y

Opção do CONFIG	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_SL AB_FREELI ST_RANDOM</u>	y	y	y	y
<u>CONFIG_SL UB_DEBUG</u>	y	y	y	y
<u>CONFIG_ST ACKPROTEC TOR</u>	y	y	y	y
<u>CONFIG_ST ACKPROTEC TOR_STRONG</u>	y	y	y	y
<u>CONFIG_ST ATIC_USER MODEHELPER</u>	n	n	n	n
<u>CONFIG_ST RICT_DEVM EM</u>	n	n	n	n
<u>CONFIG_ST RICT_KERN EL_RWX</u>	y	y	y	y
<u>CONFIG_ST RICT_MODU LE_RWX</u>	y	y	y	y
<u>CONFIG_SY N_COOKIES</u>	y	y	y	y
<u>CONFIG_VM AP_STACK</u>	y	y	y	y

Opção do CONFIG	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_WERROR	n	n	n	n
CONFIG_ZERO_CALL_USED_REGS	n	n	n	n

Permitir que os métodos ACPI estejam inserted/replaced em tempo de execução (CONFIG_ACPI_CUSTOM_METHOD)

O Amazon Linux desativa essa opção, pois permite que os usuários `root` gravem na memória arbitrária do kernel.

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Formatos binários diversos (**binfmt_misc**)

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), [AL2023 não define essa opção de configuração conforme recomendado](#) pelo KSPP. Em AL2023, esse recurso é opcional e é construído como um módulo do kernel.

Suporte a **BUG()**

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

BUG() se o kernel encontrar corrupção de dados ao verificar a validade das estruturas de memória do kernel

Algumas partes do kernel Linux verificarão a consistência interna das estruturas de dados e podem `BUG()` quando detectarem dados corrompidos.

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

COMPAT_BRK

Com essa opção desativada (que é como o Amazon Linux configura o kernel), a configuração de `randomize_va_space sysctl` é retornada para 2, o que também permite a randomização de heap sobre o topo da base mmap, pilha e randomização da página VDSO.

Essa opção existe no kernel para fornecer compatibilidade com alguns binários `libc.so.5` antigos de 1996 e anteriores.

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

COMPAT_VDSO

Essa opção de configuração é relevante para x86-64 o aarch64. Ao definir isso como n, o kernel do Amazon Linux não torna um objeto compartilhado dinâmico (VDSO) virtual de 32 bits visível em um endereço previsível. A mais recente glibc conhecida por ser quebrada por essa opção sendo definida por n é glibc 2.3.3, de 2004.

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

CONFIG_DEBUGendurecimento fechado

As opções de configuração do kernel Linux gated by CONFIG_DEBUG são normalmente projetadas para uso em kernels criados para problemas de depuração, e coisas como desempenho não são uma prioridade. AL2023 ativa a opção de CONFIG_DEBUG_LIST endurecimento.

Desative o DMA para dispositivos PCI no stub EFI antes de configurar o IOMMU

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP.

Fortalecimento para copiar memória entre o kernel e o espaço do usuário

Quando o kernel precisa copiar a memória para ou do espaço do usuário, essa opção ativa algumas verificações que podem proteger contra algumas classes de problemas de estouro de pilha.

A opção CONFIG_HARDENED_USERCOPY_FALLBACK existia nos kernels 4.16 a 5.15 para ajudar os desenvolvedores do kernel a descobrir quaisquer entradas ausentes da lista de permissões por meio de uma `WARN()`. Como AL2023 vem com um kernel 6.1, essa opção não é mais relevante para o. AL2023

A `CONFIG_HARDENED_USERCOPY_PAGESPAN` opção existia nos kernels principalmente como uma opção de depuração para desenvolvedores e não se aplica mais ao kernel 6.1 em. AL2023

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Suporte de hibernação

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP.

Essa opção precisa ser ativada para oferecer suporte à capacidade de [hibernar sua instância sob demanda](#) e para oferecer suporte à capacidade de [hibernar instâncias spot interrompidas](#).

Geração de números aleatórios

O kernel do AL2023 é configurado para garantir que a entropia adequada esteja disponível para uso no EC2.

CONFIG_INET_DIAG

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP. Em AL2023, esse recurso é opcional e é construído como um módulo do kernel.

Zere toda a memória do alocador de páginas e placas do kernel na alocação e desalocação

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP. Essas opções estão desativadas AL2023 devido ao possível impacto no desempenho da ativação dessa funcionalidade por padrão. O comportamento `CONFIG_INIT_ON_ALLOC_DEFAULT_ON` pode ser ativado adicionando `init_on_alloc=1` à linha de comando do kernel e o comportamento `CONFIG_INIT_ON_FREE_DEFAULT_ON` pode ser ativado adicionando `init_on_free=1`.

Inicialize todas as variáveis da pilha como zero (**CONFIG_INIT_STACK_ALL_ZERO**)

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP. Esta opção requer GCC 12 ou mais, embora seja AL2023 fornecida com o GCC 11.

Assinatura do módulo Kernel

AL2023 assina e valida as assinaturas dos módulos do kernel. A opção `CONFIG_MODULE_SIG_FORCE`, que exigiria que os módulos tivessem uma assinatura válida, não está habilitada para preservar a compatibilidade dos usuários que criam módulos de terceiros. Para usuários que desejam garantir que todos os módulos do kernel sejam assinados, [Módulo de segurança Linux \(LSM\) Lockdown](#) pode ser configurado para impor isso.

kexec

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP. Essa opção está ativada para que a funcionalidade `kdump` possa ser usada.

IOMMUSupport

AL2023 ativa o suporte ao IOMMU. A opção `CONFIG_IOMMU_DEFAULT_DMA_STRICT` não está habilitada por padrão, mas essa funcionalidade pode ser configurada adicionando `iommu.passthrough=0 iommu.strict=1` à linha de comando do kernel.

kfence

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP.

Suporte de **pty** antigo

AL2023 usa a PTY interface moderna (`devpts`).

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Módulo de segurança Linux (LSM) Lockdown

AL2023 constrói o lockdown LSM, que bloqueará automaticamente o kernel ao usar o Secure Boot.

A opção `CONFIG_LOCK_DOWN_KERNEL_FORCE_CONFIDENTIALITY` não está ativada. Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP. Quando não estiver usando o Secure Boot, é possível habilitar o LSM de bloqueio e configurá-lo conforme desejado.

Envenenamento de páginas

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP.

Da mesma forma [Zere toda a memória do alocador de páginas e placas do kernel na alocação e desalocação](#), isso está desativado no AL2023 kernel devido ao possível impacto no desempenho.

Protetor de pilha

O AL2023 kernel é construído com o recurso de proteção de pilha GCC ativado com a opção. - `fstack-protector-strong`

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

seccomp BPF API

O recurso de fortalecimento de seccomp é usado por softwares, como `systemd` e tempos de execução de contêineres, para fortalecer os aplicativos do espaço do usuário.

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

tempo esgotado para **panic()**

O AL2023 kernel é configurado com esse valor definido como `0`, o que significa que o kernel não será reinicializado após entrar em pânico. Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP. Isso é configurável por meio de `sysctl`, `/proc/sys/kernel/panic` e na linha de comando do kernel.

Modelos de segurança

AL2023 ativa SELinux no modo Permissivo por padrão. Para obter mais informações, consulte [Configurando modos do SELinux para o AL2023](#).

Os módulos [Módulo de segurança Linux \(LSM\) Lockdown](#) e `yama` também estão habilitados.

/proc/kcore

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP.

Randomização do deslocamento da pilha do kernel na entrada do syscall

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP. Isso pode ser ativado configurando `randomize_kstack_offset=on` na linha de comando do kernel.

Verificações de contagem de referência (**CONFIG_REFCOUNT_FULL**)

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP. No momento, essa opção não está habilitada devido ao possível impacto no desempenho.

Conhecimento do núcleos SMT pelo programador (**CONFIG_SCHED_CORE**)

O AL2023 kernel é construído com `CONFIG_SCHED_CORE`, o que permite o uso de aplicativos de espaço de usuário. `prctl(PR_SCHED_CORE)` Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Verifique se há corrupção na pilha em chamadas para **schedule()** (**CONFIG_SCHED_STACK_END_CHECK**)

O AL2023 kernel é construído com `CONFIG_SCHED_STACK_END_CHECK` enabled. Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Fortalecimento do alocador de memória

O AL2023 kernel permite o fortalecimento do alocador de memória do kernel com as opções, e. `CONFIG_SHUFFLE_PAGE_ALLOCATOR` `CONFIG_SLAB_FREELIST_HARDENED` `CONFIG_SLAB_FREELIST_RANDOM` Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

suporte de depuração SLUB

O AL2023 kernel é ativado `CONFIG_SLUB_DEBUG`, pois essa opção ativa recursos opcionais de depuração para o alocador que podem ser ativados na linha de comando do kernel. Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

CONFIG_STATIC_USERMODEHELPER

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP. Isso

ocorre porque `CONFIG_STATIC_USERMODEHELPER` requer suporte especial da distribuição que atualmente não está presente no Amazon Linux.

Texto do kernel somente para leitura e rodata (`CONFIG_STRICT_KERNEL_RWX` e `CONFIG_STRICT_MODULE_RWX`)

O AL2023 kernel é configurado para marcar o texto e a memória do kernel e do módulo do kernel como somente leitura, e a rodata memória não textual marcada como não executável. Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

TCP suporte syncookie (`CONFIG_SYN_COOKIES`)

O AL2023 kernel é construído com suporte para cookies de sincronização TCP. Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Pilha virtualmente mapeada com páginas de proteção (`CONFIG_VMAP_STACK`)

O AL2023 kernel é construído com `CONFIG_VMAP_STACK`, permitindo pilhas de kernel mapeadas virtualmente com páginas de proteção. Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Crie com avisos do compilador como erros (`CONFIG_WERROR`)

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP.

Registre a zeragem na função exit (`CONFIG_ZERO_CALL_USED_REGS`)

Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP.

Endereço mínimo para alocação de espaço de usuário

Essa opção de fortalecimento pode ajudar a reduzir o impacto dos bugs do ponteiro NULL do kernel. Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

clang opções específicas de endurecimento

O AL2023 kernel é construído com GCC em vez de clang, portanto, a opção de `CONFIG_CFI_CLANG` fortalecimento não pode ser ativada, o que também torna `CONFIG_CFI_PERMISSIVE` inaplicável. Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme recomendado pelo KSPP.

Opções de fortalecimento de kernel específicas do x86-64

Opção do CONFIG	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
<u>CONFIG_AMD_IOMMU</u>	N/D	y	N/D	y
<u>CONFIG_AMD_IOMMU_V2</u>	N/D	y	N/D	N/D
<u>CONFIG_IA32_EMULATION</u>	N/D	y	N/D	y
<u>CONFIG_INTEL_IOMMU</u>	N/D	y	N/D	y
<u>CONFIG_INTEL_IOMMU_DEFAULT_ON</u>	N/D	n	N/D	n
<u>CONFIG_INTEL_IOMMU_SVM</u>	N/D	n	N/D	n
<u>CONFIG_LEGACY_VSYSCALL_NONE</u>	N/D	n	N/D	n
<u>CONFIG_MODIFY_LDT_SYSCALL</u>	N/D	n	N/D	n
<u>CONFIG_PAGE_TABLE_ISOLATION</u>	N/D	y	N/D	N/D

Opção do CONFIG	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_RAMDOMIZE_MEMORY	N/D	y	N/D	y
CONFIG_X86_64	N/D	y	N/D	y
CONFIG_X86_64_MSR	N/D	y	N/D	y
CONFIG_X86_64_VSYSCALL_EMULATION	N/D	y	N/D	y
CONFIG_X86_64_X32	N/D	N/D	N/D	N/D
CONFIG_X86_64_X32_ABI	N/D	n	N/D	n

Suporte para x86-64

O suporte básico para x86-64 inclui o suporte a bits Physical Address Extension (PAE) e no-execute (NX). Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Suporte para AMD e Intel IOMMU

O AL2023 kernel é construído com suporte para AMD e Intel. IOMMUs Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

A opção CONFIG_INTEL_IOMMU_DEFAULT_ON não está definida, mas pode ser ativada passando intel_iommu=on para a linha de comando do kernel. Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa [opção de configuração conforme recomendado](#) pelo KSPP.

A CONFIG_INTEL_IOMMU_SVM opção não está habilitada no momento em AL2023. Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 [não define essa opção de configuração conforme recomendado](#) pelo KSPP.

Support para espaço de usuário de 32 bits

Important

O suporte para espaço de usuário x86 de 32 bits está obsoleto e o suporte para execução de binários de espaço de usuário de 32 bits pode ser removido em uma futura versão principal do Amazon Linux.

Note

Embora AL2023 não inclua mais pacotes de 32 bits, o kernel ainda suportará a execução de espaço de usuário de 32 bits. Consulte [Pacotes x86 \(i686\) de 32 bits](#) para obter mais informações.

Para oferecer suporte à execução de aplicativos de espaço de usuário de 32 bits, AL2023 não ativa a CONFIG_X86_VSYSCALL_EMULATION opção e ativa as opções CONFIG_IA32_EMULATIONCONFIG_COMPAT, e. CONFIG_X86_VSYSCALL_EMULATION Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 [não define essa opção de configuração conforme recomendado](#) pelo KSPP.

A ABI x32 nativa de 32 bits para processadores de 64 bits não está habilitada (CONFIG_X86_X32 e CONFIG_X86_X32_ABI). Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Suporte ao registro específico do modelo x86 (MSR)

A opção CONFIG_X86_MSR está ativada para oferecer suporte turbostat. Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 [não define essa opção de configuração conforme recomendado](#) pelo KSPP.

modify_ldt syscall

AL2023 não permite que programas de usuário modifiquem a Tabela Descritora Local (LDT) x86 com o syscall. `modify_ldt` Essa chamada é necessária para executar código segmentado ou de 16 bits e sua ausência pode interromper softwares do `seemu`, como a execução de alguns programas em WINE e algumas bibliotecas de threading muito antigas. Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Remover o mapeamento do kernel no modo de usuário

AL2023 configura o kernel para que a maioria dos endereços do kernel não seja mapeada no espaço do usuário. Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Randomize seções de memória do kernel

AL2023 configura o kernel para randomizar os endereços virtuais básicos das seções de memória do kernel. Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Opções de endurecimento de kernel específicas do aarch64

Opção do CONFIG	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_ARM64_BTI	y	N/D	y	N/D
CONFIG_ARM64_BTI_KERNEL	N/D	N/D	N/D	N/D
CONFIG_ARM64_PTR_AUTH	y	N/D	y	N/D
CONFIG_ARM64_PTR_AUTH_KERNEL	y	N/D	y	N/D

Opção do CONFIG	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_ARM64_SW_TTBR0_PAN	y	N/D	y	N/D
CONFIG_UNMAP_KERNEL_AT_EL0	y	N/D	y	N/D

Identificação do alvo da filial

O AL2023 kernel habilita o suporte para Branch Target Identification (CONFIG_ARM64_BTI). Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

A CONFIG_ARM64_BTI_KERNEL opção não está habilitada porque é construída com GCC, e o suporte para compilar o kernel com essa opção está atualmente desabilitado no kernel upstream devido a um bug do gcc. AL2023 Embora essa opção seja uma das [configurações recomendadas do Kernel Self Protection Project \(KSPP\)](#), AL2023 não define essa opção de configuração conforme [recomendado](#) pelo KSPP.

Autenticação de ponteiro (CONFIG_ARM64_PTR_AUTH)

O AL2023 kernel é construído com suporte para a extensão Pointer Authentication (parte das extensões ARMv8 .3), que pode ser usada para ajudar a mitigar as técnicas de Programação Orientada ao Retorno (ROP). O suporte de hardware necessário para autenticação de ponteiro no [Graviton](#) foi introduzido com o Graviton 3.

A opção CONFIG_ARM64_PTR_AUTH está habilitada e comporta autenticação de ponteiro no espaço do usuário. Como a CONFIG_ARM64_PTR_AUTH_KERNEL opção também está ativada, o AL2023 kernel pode usar a proteção do endereço de retorno para si mesmo.

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Emule o acesso privilegiado, nunca usando a comutação **TTBR0_EL1**

Essa opção impede que o kernel acesse diretamente a memória do espaço do usuário, com **TTBR0_EL1** sendo definido apenas temporariamente como um valor válido pelas rotinas de acesso do usuário.

Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

Desmapear o kernel ao executar no espaço do usuário

O AL2023 kernel é configurado para desmapear o kernel ao ser executado em userspace ().

CONFIG_UNMAP_KERNEL_AT_EL0 Essa opção é uma das [Configurações recomendadas do Kernel Self Protection Project](#).

UEFI Secure Boot no AL2023

O AL2023 oferece suporte ao UEFI Secure Boot a partir do lançamento 2023.1. Você deve usar o AL2023 com instâncias do Amazon EC2 que oferecem suporte a UEFI e UEFI Secure Boot. Para obter mais informações, consulte [Requisitos para executar uma instância do EC2 no modo de inicialização da UEFI](#) no Guia do usuário do Amazon EC2.

As instâncias do AL2023 com UEFI Secure Boot habilitado aceitam apenas códigos no nível do kernel, incluindo o kernel do Linux e módulos, que são assinados por Amazon, para que você possa garantir que sua instância execute apenas códigos no nível do kernel assinados por AWS.

Para obter mais informações sobre instâncias do Amazon EC2 e UEFI Secure Boot, consulte [UEFI Secure Boot para instâncias do Amazon EC2](#) no Guia do usuário do Amazon EC2.

Pré-requisitos

- Você deve usar uma AMI com AL2023 versão 2023.1 ou superior.
- O tipo de instância deve permitir a inicialização segura do UEFI. Para obter mais informações, consulte [Requisitos para executar uma instância do EC2 no modo de inicialização da UEFI](#) no Guia do usuário do Amazon EC2.

Habilitar UEFI Secure Boot no AL2023

As AMIs padrão do AL2023 incorporam um bootloader e um kernel assinados por nossas chaves. Você pode ativar o UEFI Secure Boot inscrevendo instâncias existentes ou criando AMIs com o UEFI

Secure Boot pré-ativado registrando uma imagem de um snapshot. O UEFI Secure Boot não está habilitado por padrão nas AMIs padrão do AL2023.

O modo de inicialização das AMIs AL2023 é definido para `uefi-preferred` que garanta que as instâncias executadas com essas AMIs usem o firmware UEFI, se o tipo de instância for compatível com UEFI. Se o tipo de instância não for compatível com UEFI, a instância será iniciada com firmware de BIOS antigo. Quando uma instância é executada no modo BIOS antigo, o UEFI Secure Boot não é aplicado.

Para obter mais informações sobre os modos de inicialização AMI em instâncias Amazon EC2, consulte [Comportamento de execução da instância com modos de inicialização do Amazon EC2](#) no Guia do usuário do Amazon EC2.

Tópicos

- [Inscrição de uma instância existente](#)
- [Registrar imagem do instantâneo](#)
- [Atualizações de revogação](#)
- [Como funciona o UEFI Secure Boot no AL2023](#)
- [Inscrevendo suas próprias chaves](#)

Inscrição de uma instância existente

Para registrar uma instância existente, preencha as variáveis específicas do firmware UEFI com um conjunto de chaves que permitem que o firmware verifique o carregador de inicialização e o carregador de inicialização verifique o kernel na próxima inicialização.

1. O Amazon Linux fornece uma ferramenta para simplificar o processo de inscrição. Execute o comando a seguir para provisionar a instância com o conjunto necessário de chaves e certificados.

```
sudo amazon-linux-sb enroll
```

2. Execute o seguinte comando para reiniciar a instância do . Depois que a instância for reinicializada, o UEFI Secure Boot será ativado.

```
sudo reboot
```

Note

Atualmente, as AMIs do Amazon Linux não oferecem suporte ao Nitro Trusted Platform Module (NitroTPM). Se você precisar do NitroTPM além do UEFI Secure Boot, use as informações na seção a seguir.

Registrar imagem do instantâneo

Ao registrar uma AMI a partir de um snapshot de um volume raiz do Amazon EBS usando a `register-image` API do Amazon EC2, você pode provisionar a AMI com um blob binário que contém o estado do armazenamento de variáveis UEFI. Ao fornecer o `AL2023 UefiData`, você ativa o UEFI Secure Boot e não precisa seguir as etapas na seção anterior.

Para obter mais informações sobre como criar e usar um blob binário, consulte [Criar um blob binário contendo um armazenamento de variáveis pré-preenchido](#) no Guia do usuário do Amazon EC2.

O AL2023 fornece um blob binário pré-criado que pode ser usado diretamente nas instâncias do Amazon EC2. O blob binário está localizado em `/usr/share/amazon-linux-sb-keys/uefi.vars` em uma instância em execução. Esse blob é fornecido pelo pacote `amazon-linux-sb-keys RPM`, que é instalado por padrão nas AMIs AL2023 a partir da versão 2023.1.

Note

Para garantir que você esteja usando a versão mais recente das chaves e revogações, use o blob da mesma versão do AL2023 que você usa para criar a AMI.

Ao registrar uma imagem, recomendamos usar o parâmetro `BootMode` da API [RegisterImage](#) definida como `uefi`. Isso permite que você ative o NitroTPM definindo o parâmetro `TpmSupport` como `v2.0`. Além disso, definir `BootMode` para `uefi` garante que o UEFI Secure Boot esteja habilitado e não possa ser desativado acidentalmente ao mudar para um tipo de instância que não seja compatível com UEFI.

Para obter mais informações sobre o NitroTPM, consulte [NitroTPM para instâncias do Amazon EC2](#) no Guia do usuário do Amazon EC2.

Atualizações de revogação

Talvez seja necessário que o Amazon Linux distribua uma nova versão do bootloader `grub2` ou do kernel Linux assinado com chaves atualizadas. Nesse caso, talvez seja necessário revogar a chave antiga para evitar a chance de permitir que bugs exploráveis de versões anteriores do bootloader ignorem o processo de verificação do UEFI Secure Boot.

As atualizações de pacotes do `grub2` ou `kernel` sempre atualizam automaticamente a lista de revogações no armazenamento de variáveis UEFI da instância em execução. Isso significa que, com o UEFI Secure Boot ativado, você não pode mais executar a versão antiga de um pacote depois de instalar uma atualização de segurança para o pacote.

Como funciona o UEFI Secure Boot no AL2023

Ao contrário de outras distribuições Linux, o Amazon Linux não fornece um componente adicional, chamado shim, para atuar como o bootloader de primeiro estágio. O calço geralmente é assinado com chaves da Microsoft. Por exemplo, em distribuições Linux com o shim, o shim carrega o bootloader `grub2` que usa o próprio código do shim para verificar o kernel Linux. Além disso, o shim mantém seu próprio conjunto de chaves e revogações no banco de dados da Machine Owner Key (MOK) localizado no armazenamento de variáveis UEFI e controlado com a ferramenta `mokutil`.

O Amazon Linux não oferece nada. Como o proprietário da AMI controla as variáveis da UEFI, essa etapa intermediária não é necessária e afetaria negativamente os tempos de lançamento e inicialização. Além disso, optamos por não incluir a confiança em nenhuma chave de fornecedor por padrão, para reduzir a chance de binários indesejados serem executados. Como sempre, os clientes podem incluir binários se quiserem fazer isso.

Com o Amazon Linux, o UEFI carrega e verifica diretamente nosso `grub2` bootloader. O bootloader `grub2` foi modificado para usar UEFI para verificar o kernel Linux após carregá-lo. Assim, o Kernel Linux é verificado usando os mesmos certificados armazenados na variável UEFI normal `db` (banco de dados de chaves autorizado) e testado com a mesma variável `dbx` (banco de dados de revogações) do bootloader e outros binários UEFI. Como fornecemos nossas próprias chaves PK e KEK, que controlam o acesso ao banco de dados `db` e ao banco de dados `dbx`, podemos distribuir atualizações e revogações assinadas conforme necessário, sem um intermediário como o shim.

Para obter mais informações sobre o UEFI Secure Boot, consulte [Como o UEFI Secure Boot funciona com instâncias do Amazon EC2](#) no Guia do usuário do Amazon EC2.

Inscrevendo suas próprias chaves

Conforme documentado na seção anterior, o Amazon Linux não exige uma inicialização segura shim para UEFI no Amazon EC2. Ao ler a documentação de outras distribuições Linux, você pode encontrar documentação para gerenciar o banco de dados Machine Owner Key (MOK) usando `mokutil`, que não está presente no AL2023. Os ambientes shim e MOK contornam algumas limitações de registro de chaves no firmware UEFI que não são aplicáveis à forma como o Amazon EC2 implementa o UEFI Secure Boot. Com o Amazon EC2, existem mecanismos para manipular facilmente e diretamente as chaves no armazenamento de variáveis UEFI.

Se quiser inscrever suas próprias chaves, você pode manipular o armazenamento de variáveis em uma instância existente (consulte [Adicionar chaves ao armazenamento de variáveis de dentro da instância](#)) ou criar um blob binário pré-preenchido (consulte [Criar um blob binário contendo um armazenamento de variáveis pré-preenchido](#)).

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.