Amazon Linux 2023

User Guide

Amazon Linux 2023: User Guide

Table of Contents

What is Amazon Linux 2023?

Learn about Amazon Linux 2023 (AL2023), the next generation of Amazon Linux from Amazon Web Services. Develop and run cloud and enterprise applications in a secure, stable, and high-performance runtime environment. With AL2023, you get an application environment that offers long-term support with access to the latest innovations in Linux. AL2023 is provided at no additional charge.

Topics

Comparing Amazon Linux 2 and Amazon Linux 2023

The following topics outline key differences between Amazon Linux 2 and Amazon Linux 2023 (AL2023).

Topics

Support for each release

For Amazon Linux 2023, we offer five years of support.

For more information, see Release cadence (p. 5).

Naming and versioning changes

AL2023 supports the same mechanisms that Amazon Linux 2 supports for platform identification. AL2023 also introduces new files for platform identification.

For more information, see Naming and versioning (p. 6).

Optimizations

AL2023 optimizes boot time to reduce the time from instance launch to running the customer workload. These optimizations span the Amazon EC2 instance kernel configuration, `cloud-init` configurations, and features that are built into packages in the OS such askmod and `systemd`.

For more information about optimizations, see Performance and operational optimizations (p. 7).

Security updates

SELinux

By default, Security Enhanced Linux (SELinux) for AL2023 is `enabled` and set to `permissive` mode. In `permissive` mode, permission denials are logged but not enforced.

SELinux is a security feature of the Amazon Linux kernel, which was `disabled` in Amazon Linux 2. SELinux is a collection of kernel features and utilities that provides mandatory access control (MAC) architecture into the major subsystems of the kernel.

For more information, see Setting SELinux modes (p. 18).

For more information about SELinux repositories, tools, and policies, see SELinux Notebook, Types of SELinux policy, and SELinux Project .

OpenSSL 3

AL2023 features the Open Secure Sockets Layer version 3 (OpenSSL 3) cryptography toolkit. AL2023 uses the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) network protocols. It also uses the required cryptography standards.

By default, Amazon Linux 2 comes with OpenSSL 1.0.2. You can build applications against OpenSSL 1.1.1.

For more information about OpenSSL, see the OpenSSL migration guide.

For more information about security, see Security updates and features (p. 14).

IMDSv2

By default, any instances launched with the AL2023 AMI will require the use of IMDSv2-only and your default hop limit will be set to 2 to allow for containerized workload support. This is done by setting the `imds-support` parameter to `v2.0`. For more information, see Configure the AMI in the *Amazon EC2 User Guide for Linux Instances*.

Note
The session token's time of validity can be anywhere between 1 second and 6 hours. The addresses to direct the API requests for IMDSv2 queries are the following:

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

You can still manually override these settings and enable IMDSv1 using Instance Metadata option launch properties. You can also still use IAM controls to enforce different IMDS settings. For more information about setting up and using the Instance Metadata Service, see Use IMDSv2, Configure instance metadata options for new instances, and Modify instance metadata options for existing instances, in the *Amazon EC2 User Guide for Linux Instances*.

Deterministic upgrades for stability

With the deterministic upgrades through versioned repositories feature, every AL2023 Amazon Machine Image (AMI) by default is locked to a specific repository version. You can use deterministic upgrades to achieve greater consistency among package versions and updates. Each release, major or minor, includes a specific repository version.

New with AL2023, deterministic upgrading by default is enabled. This is an improvement over the manual, incremental method of locking that's used in Amazon Linux 2 and other earlier versions.

For more information, see Using Deterministic upgrades through versioned repository (p. 8).

Sourced from multiple upstreams

AL2023 is RPM-based and includes components sourced from multiple versions of Fedora and other distributions, such as CentOS 9 Stream. The Amazon Linux kernel is sourced from the long-term support (LTS) releases directly from kernel.org, chosen independently from other distributions.

For more information, see Relationship to Fedora (p. 8).

AMI root file system and default Amazon EBS volume type

The AL2023 AMI and Amazon Linux 2 both use the XFS file system on the root file system. For AL2023, the `mkfs`options for the root device file system are further optimized for Amazon EC2. AL2023 also supports a number of other file systems that you can use on other volumes to meet your specific requirements.

AL2023 AMIs use Amazon EBS `gp3` volumes by default, whereas Amazon Linux 2 AMIs use Amazon EBS `gp2` volumes by default. You can change the volume type when you launch an instance. For more information about Amazon EBS volume types, see Amazon EBS General Purpose Volumes. For more information about launching an Amazon EC2 instance, see Launch an instance in the *Amazon EC2 User Guide for Linux Instances*.

Networking system service

The `systemd-networkd` system service manages the network interfaces in AL2023. This is a change from Amazon Linux 2, which uses ISC dhclient or `dhclient`.

For more information, see Networking service (p. 16).

Packages for glibc, gcc, and binutils

AL2023 includes many of the same core packages as Amazon Linux 2.

We updated the following three core toolchain packages for AL2023.

Package name	Amazon Linux 2	AL2023
glibc	2.26	2.34
gcc	7.3	11.3
binutils	2.29	2.39

For more information, see Core toolchain packages glibc, gcc, binutils (p. 16).

Package manager

The default software package management tool on AL2023 is DNF. DNF is the successor to YUM, the package management tool in Amazon Linux 2.

For more information, see Package management tool (p. 17).

SSH server default configuration changes

For the AL2023 AMI, we changed the types of `sshd` host keys that we generate with the release. We also dropped some legacy key types to avoid generating them at launch time. Clients must support the `rsa-sha2-256` and `rsa-sha2-512` protocols or `ssh-ed25519` with use of an `ed25519` key. By default, `ssh-rsa` signatures are disabled.

Additionally, AL2023 configuration settings in the default `sshd_config` file contain `UseDNS=no`. This new setting means that DNS impairments are less likely to block your ability to establish `ssh` sessions with your instances. The tradeoff is that the "`from=`*`hostname.domain`*`,`*`hostname.domain`*" line entries in your `authorized_keys` files won't be resolved. Because `sshd` no longer attempts to resolve the DNS names, each comma separated *`hostname.domain`* value must be translated to a corresponding IP address.

For more information, see Default SSH server configuration (p. 17).

Extra Packages for Enterprise Linux (EPEL)

Extra Packages for Enterprise Linux (EPEL) is a project in the Fedora community with the objective of creating a large array of packages for enterprise-level Linux operating systems. The project has primarily produced RHEL and CentOS packages. Amazon Linux 2 features a high level of compatibility with CentOS 7. As a result, many EPEL7 packages work on Amazon Linux 2. However, AL2023 doesn't support EPEL or EPEL-like repositories.

Using cloud-init

In AL2023, cloud-init manages the package repository. By default, in earlier versions of Amazon Linux, cloud-init installed security updates. This isn't the default for AL2023. The new deterministic upgrading features for updating `releasever` at launch describe the AL2023 way to enable package updates at launch. For more information, see Managing packages and operating system updates (p. 34) and Deterministic upgrades for stability (p. 3).

With AL2023, you can use cloud-init with SELinux. For more information, see Use cloud-init to enable `enforcing` mode (p. 20).

Cloud-init loads configuration content with cloud-init from remote locations using HTTP(S). In earlier versions, Amazon Linux doesn't alert you when remote resources are unavailable. In AL2023, unavailable remote resources creates a fatal error and fails the cloud-init execution. This change in behavior from Amazon Linux 2, provides a safer "fail closed" default behavior.

For more information, see Customized cloud-init (p. 13) and the cloud-init Documentation.

Graphical desktop support

Amazon Linux 2023 is cloud-centered and optimized for Amazon EC2 usage and currently does not include a graphical or desktop environment. To provide feedback on GitHub, see https://github.com/amazonlinux/amazon-linux-2023/issues.

Release cadence

A new major version of Amazon Linux is released every two years and includes five years of support. Each release includes support in two phases. The standard support phase covers the first two years. Next, a maintenance phase continues support for an additional three years.

In the standard support phase, the release receives quarterly minor version updates. During the maintenance phase, a release receives only security updates and critical bug fixes that are published as soon as they're available.

Year	Amazon Linux 2023	Amazon Linux 2025	Amazon Linux 2027	Amazon Linux 2029
2023	✓ Standard support			
2024	✓ Standard support			
2025	Maintenance	✓ Standard support		
2026	Maintenance	✓ Standard support		
2027	Maintenance	Maintenance	✓ Standard support	
2028	✗ EOL	Maintenance	✓ Standard support	
2029	✗ EOL	Maintenance	Maintenance	✓ Standard support
2030	✗ EOL	✗ EOL	Maintenance	✓ Standard support
2031	✗ EOL	✗ EOL	Maintenance	Maintenance

Major and minor releases

With every Amazon Linux release (major version, minor version, or a security release), we release a new Linux Amazon Machine Image (AMI).

- **Major version release**— Includes new features and improvements in security and performance across the stack. The improvements might include major changes to the kernel, toolchain, Glib C, OpenSSL,

and any other system libraries and utilities. Major releases of Amazon Linux are based in part on the current version of the upstream Fedora Linux distribution. Amazon might add or replace specific packages from other non-Fedora upstreams.

- **Minor version release**— A quarterly update that includes security updates, bug fixes, and new features and packages. Each minor version is a cumulative list of updates that includes security and bug fixes in addition to new features and packages. These releases might include latest language runtimes, such as PHP. They might also include other popular software packages such as Ansible and Docker.

Consuming new releases

Updates are provided through a combination of new Amazon Machine Image (AMI) releases and corresponding new repositories. By default, a new AMI and the repository that it points to are coupled. However, you can point your running Amazon EC2 instances to newer repository versions over time to apply updates on the running instances. You can also update by launching new instances of the latest AMIs.

Long-term support policy

Amazon Linux provides updates for all of your packages and maintains compatibility within a major version for your applications that are built on Amazon Linux. Core packages such as the glibc library, OpenSSL, OpenSSH, and the DNF package manager receive support for the lifetime of the major AL2023 release. Packages that aren't part of the core packages are supported based on their specific upstream sources. You can see the specific support status and dates of individual packages by running the following command.

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

You can get information on all currently installed packages by running the following command.

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

The full list of core packages is finalized during the preview. If you want to see more packages included as core packages, tell us. We evaluate as we are collecting feedback. Feedback on AL2023 can be provided through your designated Amazon representative or by filing an issue in the amazon-linux-2023 repo on GitHub.

Naming and versioning

Amazon Linux 2023 (AL2023) provides a minor release every three months during the two years of standard support. Each release is identified by an increment from 0 to N. 0 refers to the original major release for that iteration. All releases will be called Amazon Linux 2023. When Amazon Linux 2025 is released, AL2023 will enter extended support and receive updates for security updates and critical bug fixes.

For example, minor releases of AL2023 have the following format:

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

The corresponding AL2023 AMIs have the following format:

- `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`

Within a specific minor version, regular AMI releases occur with a timestamp of the date of the AMI release.

- `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`

The recommended method for identifying an Amazon Linux 2 or AL2023 instance starts with reading the Common Platform Enumeration (CPE) string from `/etc/system-release-cpe`. Then, split the string into its fields. Last, read the platform and version values.

AL2023 also introduces new files for platform identification:

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

These two files indicate that an instance is Amazon Linux. There's no need to read a file or split the string into fields, unless you want to know the specific platform and version values.

Performance and operational optimizations

Amazon Linux 6.1 kernel

- Amazon Linux 2023 (AL2023) uses the latest drivers for Elastic Network Adapter (ENA) and Elastic Fabric Adapter (EFA) devices. AL2023 has a focus on performance and functionality backports for hardware in Amazon EC2 infrastructure.
- Kernel live patching is available for the `x86_64` and `aarch64` instance types. This reduces the need for frequent reboots.

  Note
  Kernel live patching is not enabled in AL2023 RC0. It will be enabled before AL2023 RC1.
- All kernel build and runtime configurations include many of the same performance and operational optimizations of Amazon Linux 2.

Base toolchain selection and default build flags

- The compiler optimization paths are optimized for performance on specific target architectures on EC2 hardware. Graviton 2 instances in particular are optimized for performance.
- AL2023 uses the updated versions of Rust, Clang/LLVM, and Go.

Package selection and versions

- Select backports to major system components include several performance improvements for running on Amazon EC2 infrastructure, especially Graviton instances.
- AL2023 is integrated with several Amazon Web Services and features. This includes the Amazon CLI, SSM Agent, Amazon Kinesis Agent, and CloudFormation.

- AL2023 uses Amazon Corretto as the Java Development Kit (JDK).
- AL2023 provides database engines and programming language runtime updates to newer versions as they're released by upstream projects. Programming language runtimes with new versions are added when they're released.

Deployment in a cloud environment

- The base AL2023 AMI and container images are frequently updated to support patching instance replacement.
- Kernel updates are included in AL2023 AMI updates. This means that you don't need to use commands such as `yum update` and `reboot` to update your kernel.
- In addition to the standard AL2023 AMI, a minimal AMI and container image is also available. Choose the minimal AMI to run an environment with the minimal number of packages that's required to run your service.
- By default, AL2023 AMIs and containers are locked to a specific version of the package repositories. There's no auto-update when they're launched. This means that you're always in control of when you ingest any package update. You can always test in a beta/gamma environment before rolling out to production. If there's a problem, you can use the pre-validated rollback path.

Relationship to Fedora

Amazon Linux 2023 (AL2023) maintains its own release and support life cycles independent of Fedora. AL2023 provides updated versions of open-source software, a larger variety of packages, and frequent releases. This preserves the familiar RPM-based operating systems.

The Generally Available (GA) version of AL2023 isn't directly comparable to any specific Fedora release. The AL2023 GA version includes components from Fedora 34, 35, and 36. Some of the components are the same as the components in Fedora and some are modified. Other components more closely resemble the components in CentOS 9 Streams or were developed independently. The Amazon Linux kernel is sourced from the long-term support options that are on kernel.org, chosen independently from Fedora.

Using Deterministic upgrades through versioned repository

Note
By default, your Amazon Linux 2023 (AL2023) instance doesn't automatically receive additional critical and important security updates at launch. Your instance initially contains the updates that were available in the version of AL2023 and the chosen AMI.

Control the updates received from major and minor releases

With AL2023, you can ensure consistency between package versions and updates across your environment. You can also ensure consistency for multiple instances of the same Amazon Machine Image (AMI). With the deterministic upgrades through versioned repositories feature, which is turned on by default, you can apply updates based on a schedule that meets your specific needs.

Whenever we release new package updates, there's a new version to lock to, and new AMIs that lock to that version.

AL2023 locks to a specific version of your repository. This is supported for both major or minor versions. The AL2023 AMI, exposed through our SSM parameters, is always the latest version. It has the most up-to-date packages and updates, including critical and important security updates.

If you launch an instance from an existing AMI, updates aren't automatically applied. Any additional packages that are installed as part of your provisioning map to the repository version of the existing AMI.

With this feature, you're in charge of ensuring consistency among package versions and updates across your environment. This is particularly the case if you're launching multiple instances from the same AMI. You can apply updates based on a schedule that meets your needs. You can also apply a specific set of updates on launch because these can also be locked to a specific repository version.

Differences between minor and major version upgrades

Major version releases of AL2023 include large-scale updates and might add, delete, or update packages. To ensure compatibility, upgrade your instance to a new major version only after you test your application on that version.

Minor version releases of AL2023 include feature and security updates, but don't include package changes. This ensures that Linux features and the system library API stay available on new versions. Testing your application before updating isn't necessary.

Control the package updates available from the Amazon Linux 2023 repositories

When we publish a new version of the Amazon Linux 2023 (AL2023) repositories, all previous versions are still available. By default, the plugin for managing repository versions locks to the same version that was used to build the AMI. If you want to control package updates, follow these steps.

1. Discover available repository versions by running the following command.

```
$ sudo dnf check-release-update
```

2. Select a version by running the following command.

```
$ sudo dnf --releasever=version update
```

This command starts an update using `dnf` from your current Amazon Linux release version to the release version that's specified in the command line. A list of the package updates is presented by `dnf`. Before the update is processed, you must confirm the update. After the update is complete, the new release version becomes the default release version that `dnf` uses for all future activities.

For more information, see Managing packages and operating system updates (p. 34).

Deterministic upgrades through versioned repositories usage

Topics

Using a deterministic upgraded system

When you run the `dnf upgrade` command, the system checks for upgrades in the repository that the `releasever` variable specifies. A valid `releasever` is either *`latest`* or a date-stamped version such as *`2023.0.20230210`*.

You can change the value of `releasever` using one of the following methods. These methods are listed in descending system priority. This means that method 1 overrides methods 2 and 3, and method 2 overrides method 3.

1. The value in the command line flag, `--releasever=`*`latest`*, if it's used.
2. The value that's specified in the override variable file, `/etc/dnf/vars/releasever`, if it's set.
3. The currently installed version of the `system-release` package.

In the following example, the version is *`2023.0.20230210`*:

```
$ rpm -q system-release
system-release-2023.0.20230210-0.amzn2023.noarch
```

In a newly installed system, the override variable is not present. No upgrades are available because the system is locked to the installed version of `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!
```

You can get packages of a specific version by using the `releasever` flag to provide the version that you want.

```
$ 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
```

Because the `--releasever` option overrides both `system-release` and `/etc/dnf/vars/releasever`, the result of this upgrade is the following:

1. The upgrade replaces all installed packages that changed between the previous and new versions.
2. The upgrade locks the system to the repository for the new version of `system-release`.

Selective update of a deterministic upgraded system

You might want to install selected packages from a recent release, while leaving the system locked to the original release version.

You can use `dnf check-update` to identify the packages that you want to upgrade.

```
$ 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
```

Install the packages that you want to upgrade. Use `sudo dnf upgrade --releasever=latest` and the package names to ensure that the `system-release` package remains unchanged.

```
$ 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
Using `sudo dnf upgrade --releasever=latest` updates all packages, including `system-release`. Then, the version remains locked to the new `system-release` unless you set the persistent override.

Using persistent override with deterministic upgrade

Instead of adding `--releasever=latest`, you can use persistent override to *unlock* the system by setting the variable value to *`latest`*.

```
$ 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-chrony-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
If you used the override variable `/etc/dnf/vars/releasever`, use the following command to restore the default locking behavior by erasing the override value.

```
$ sudo rm /etc/dnf/vars/releasever
```

Customized cloud-init

The cloud-init package is an open-source application that bootstraps Linux images in a cloud computing environment. For more information, see the cloud-init documentation.

AL2023 contains a customized version of cloud-init. With cloud-init, you can specify what occurs to your instance at boot time.

When you launch an instance, you can use user-data fields to pass actions to cloud-init. This means that you can use common Amazon Machine Images (AMIs) for many use cases and configure them dynamically when you start an instance. AL2023 also uses cloud-init to configure the `ec2-user` account.

AL2023 uses the cloud-init actions in `/etc/cloud/cloud.cfg.d` and `/etc/cloud/cloud.cfg`. You can create your own cloud-init action files in the `/etc/cloud/cloud.cfg.d` directory. Cloud-init reads all the files in this directory in lexicographical order. Later files overwrite values in earlier files. When cloud-init launches an instance, the cloud-init package does the following configuration tasks:

- Sets the default locale
- Sets the hostname
- Parses and handles user-data
- Generates host private SSH keys
- Adds a user's public SSH keys to `.ssh/authorized_keys` for easy login and administration
- Prepares the repositories for package management
- Handles package actions that are defined in user-data
- Runs user scripts that are in user-data
- Mounts instance store volumes, if applicable
  - By default, if the `ephemeral0` instance store volume is present and contains a valid file system, the instance store volume is mounted at `/media/ephemeral0`. Otherwise, it's not mounted.
  - By default, for the `m1.small` and `c1.medium` instance types, all swap volumes that are associated with the instance are mounted.
  - You can override the default instance store volume mount with the following cloud-init directive:

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

    For more control over mounts, see Mounts in the cloud-init documentation.
  - When an instance launches, instance store volumes that support TRIM aren't formatted. Before you can mount instance store volumes, you must partition and format instance store volumes.

    For more information, see Instance store volume TRIM support in the *Amazon EC2 User Guide for Linux Instances*.
  - When you launch your instances, you can use the `disk_setup` module to partition and format your instance store volumes.

    For more information, see Disk Setup in the cloud-init documentation.

For information about using cloud-init with SELinux, see Use cloud-init to enable `enforcing` mode (p. 20).

For information about cloud-init user-data formats, see User-Data Formats in the cloud-init documentation.

For more information about cloud-init, see the cloud-init Documentation.

Security updates and features

Amazon Linux 2023 (AL2023) provides many security updates and solutions.

Topics

Manage updates

Apply security updates using DNF and repository versions. For more information, see Managing packages and operating system updates (p. 34).

Security in the cloud

Understand how to apply the shared responsibility model for security in the cloud and of the cloud when using AL2023. For more information, see Security in Amazon Linux 2023 (p. 45).

SELinux modes

By default, SELinux is enabled and set to permissive mode in AL2023. In permissive mode, permission denials are logged but not enforced.

The SELinux policies define permissions for users, processes, programs, files, and devices. With SELinux, you can choose one of two policies. The policies are targeted or multi-level security (MLS).

For more information about SELinux modes and policy, see Setting SELinux modes (p. 18) and the SELinux Project Wiki.

Compliance program

Independent auditors assess the security and compliance of AL2023 along with many Amazon compliance programs.

SSH server default

AL2023 includes OpenSSH 8.7. OpenSSH 8.7 by default disables the `ssh-rsa` key exchange algorithm. For more information, see Default SSH server configuration (p. 17).

Major features of OpenSSL 3

- The Certificate Management Protocol (CMP, RFC 4210) includes both CRMF (RFC 4211) and HTTP transfer (RFC 6712).
- A HTTP or HTTPS client in libcrypto supports GET and POST actions, redirection, plain and ASN.1-encoded content, proxies, and timeouts.
- The EVP_KDF works with Key Derivation Functions.
- The EVP_MAC API works with MACs.
- Linux Kernel TLS support.

For more information, see the OpenSSL migration guide.

For more information about compliance and security in the cloud, see Security in Amazon Linux 2023 (p. 45).

Networking service

The open-source project `systemd-networkd` is widely available in modern Linux distributions. The project uses a declarative configuration language that's similar to the rest of the `systemd` framework. Its primary configuration file types are `.network` and `.link` files.

The `amazon-ec2-net-utils` package generates interface-specific configurations in the `/run/systemd/network` directory. These configurations enable both IPv4 and IPv6 networking on interfaces when they're attached to an instance. These configurations also install policy routing rules that help ensure that locally sourced traffic is routed to the network through the corresponding instance's network interface. These rules do this by ensuring that the right traffic is routed through the Elastic Network Interface (ENI) from the associated addresses or prefixes. For more information about using ENI, see Using ENI in the *Amazon EC2 User Guide for Linux Instances*.

You can customize this networking behavior by placing a custom configuration file in the `/etc/systemd/network` directory to override the default configuration settings contained in `/run/systemd/network`.

The systemd.network documentation describes how the `systemd-networkd` service determines the configuration that applies to a specific interface. It also generates alternative names, known as altnames, for the ENI-backed interfaces to reflect the properties of various Amazon resources. These ENI-backed interface properties are the `ENI ID` and the `DeviceIndex` field of the ENI attachment. You can refer to these interfaces using their properties when using various tools, such as the `ip` command.

Amazon Linux 2023 (AL2023) instance interface names are generated using the `systemd` slot naming scheme. For more information, see systemd.net naming scheme.

Additionally, AL2023 uses the `fq_codel` active queue management network transmission scheduling algorithm by default. For more information, see CoDel overview.

Core toolchain packages glibc, gcc, binutils

A subset of packages in Amazon Linux is designated as core toolchain packages. As a major part of Amazon Linux 2023 (AL2023), core packages receive five years of support. We might change the version of a package, but long-term support applies to the package included in the Amazon Linux release.

These three core packages provide a system toolchain that's used to build most software in the Amazon Linux distribution.

Package	Definition	Purpose
glibc 2.34	System C library	Used by most binary programs that provide standard functions and by the interface between programs and the kernel.
gcc 11.2	gcc compiler suite	Compiles C, C++, Fortran.
binutils 2.35	Assembler and linker plus other binary tools	Manipulates or inspects binary programs.

We recommend that updates to any glibc libraries are followed by a reboot. For updates to packages that control services, it might be sufficient to restart the services to pick up the updates. However, a system reboot ensures that all previous package and library updates are complete.

Package management tool

In Amazon Linux 2023 (AL2023), the default software package management tool is DNF. DNF is the successor to YUM, the package management tool in Amazon Linux 2.

DNF is similar to YUM in its usage. Many DNF commands are the same and with the same options as YUM commands. In a Command Line Interface (CLI) command, in most cases `dnf` replaces `yum`.

For example, for the following Amazon Linux 2 `yum` commands:

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

In AL2023, they become these commands:

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

Many of the `dnf` commands and `yum` commands are the same. In AL2023 the `yum` command is still available, but as a pointer to the `dnf` command. So, when the `yum` command is used in the shell or in a script, all commands and options are the same as the DNF CLI. For more information about the differences between the YUM CLI and the DNF CLI, see Changes in DNF CLI compared to YUM.

For a complete reference of commands and options for the `dnf` command, refer to the man page `man dnf`. For more information, see DNF Command Reference

Default SSH server configuration

If you have SSH clients from several years ago, you might see an error when you connect to an instance. If the error tells you there's no matching host key type found, update your SSH host key to troubleshoot this issue.

Default disabling of `ssh-rsa` signatures

Amazon Linux 2023 (AL2023) includes a default configuration that disables the legacy `ssh-rsa` host key algorithm and generates a reduced set of host keys. Clients must support the `ssh-ed25519` or the `ecdsa-sha2-nistp256` host key algorithm.

The default configuration accepts any of these key exchange algorithms:

- `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`

By default, AL2023 generates ed25519 and ECDSA host keys. Clients support either the ssh-ed25519 or the ecdsa-sha2-nistp256 host key algorithm. When you connect by SSH to an instance, you must use a client that supports a compatible algorithm, such as ssh-ed25519 or ecdsa-sha2-nistp256. If you need to use other key types, override the list of generated keys with a cloud-config fragment in user-data.

In the following example, cloud-config generates a rsa host key with the ecdsa and ed25519 keys.

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

If you use an RSA key pair for public key authentication, your SSH client must support a rsa-sha2-256 or rsa-sha2-512 signature. If you're using an incompatible client and can't upgrade, re-enable ssh-rsa support on your instance. Do this by activating the LEGACY system crypto policy with the following commands.

```
$ sudo dnf install crypto-policies-scripts
$ sudo update-crypto-policies --set LEGACY
```

For more information about managing host keys, see Amazon Linux Host keys

Setting SELinux modes

By default, Security Enhanced Linux (SELinux) is enabled and set to permissive mode for Amazon Linux 2023 (AL2023). In permissive mode, permission denials are logged but not enforced. SELinux is a collection of kernel features and utilities to provide a strong, flexible, mandatory access control (MAC) architecture to the major subsystems of the kernel.

SELinux provides an enhanced mechanism to enforce the separation of information based on confidentiality and integrity requirements. This separation of information reduces threats of tampering and bypassing of application security mechanisms. It also confines damage that can be caused by malicious or flawed applications.

SELinux includes a set of sample security policy configuration files that's designed to meet everyday security goals.

For more information about SELinux features and functionality, see SELinux Notebook and Policy Languages".

Topics

Default SELinux status and modes

For Amazon Linux 2023 (AL2023), SELinux by default is enabled and set to permissive mode. In permissive mode, permission denials are logged but not enforced.

The **getenforce** or **sestatus** commands tell you the current SELinux status, policy, and mode.

With the default status set to enabled and permissive, the **getenforce** command returns permissive.

The **sestatus** command returns the SELinux status and the current SELinux policy:

```
$ 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
```

When you run SELinux in permissive mode, users can sometimes label files incorrectly. When you run SELinux in the disabled status, files aren't labeled. Both incorrect or unlabeled files can cause problems when you change to enforcing mode.

SELinux automatically relabels files to avoid this problem. SELinux prevents labeling problems with automatic relabeling when you change the status to enabled.

Change to enforcing mode

When you run SELinux in enforcing mode, the SELinux utility is enforcing the configured policy. SELinux governs the capabilities of select applications by allowing or denying access based on the policy's rules.

You can find the current SELinux mode with the getenforce command.

```
getenforce
Permissive
```

Edit config file to enable enforcing mode

You can use the following steps to change the mode to enforcing.

1. Edit the /etc/selinux/config file to change to enforcing mode. The SELINUX setting should look like the example.

   ```
   SELINUX=enforcing
   ```

2. Restart your system to complete the change to enforcing mode.

   ```
   $ sudo reboot
   ```

On the next boot, SELinux relabels all files and directories in the system. SELinux also adds the SELinux context for files and directories that were created when SELinux was disabled.

After changing to enforcing mode, SELinux might deny some actions because of incorrect or missing SELinux policy rules. You can view the actions that SELinux denies with the following command.

```
$ sudo ausearch -m AVC,USER_AVC,SELINUX_ERR,USER_SELINUX_ERR -ts recent
```

Use cloud-init to enable `enforcing` mode

As an alternative, when you launch your instance, pass the following `cloud-config` as user-data to enable `enforcing` mode.

```
#cloud-config
selinux:
  mode: enforcing
```

By default, this setting causes the instance to reboot. For greater stability, we recommend rebooting your instance. However, if you prefer, you can skip the reboot by providing the following `cloud-config`.

```
#cloud-config
selinux:
  mode: enforcing
  selinux_no_reboot: 1
```

Option to disable SELinux

When you disable SELinux, SELinux policy isn't loaded or enforced. Access Vector Cache (AVC) messages aren't logged. You lose all benefits of running SELinux.

Instead of disabling SELinux, we recommend using `permissive` mode. It costs only a little more to run in `permissive` mode than it does to disable SELinux completely. Transitioning from `permissive` mode to `enforcing` mode requires much less configuration adjustment than transitioning back to `enforcing` mode after disabling SELinux. You can label files, and the system can track and log actions that the active policy might have denied.

Change SELinux to `permissive` mode

When you run SELinux in `permissive` mode, SELinux policy isn't enforced. In `permissive` mode, SELinux logs AVC messages but doesn't deny operations. You can use these AVC messages for troubleshooting, debugging, and SELinux policy improvements.

1. Edit the `/etc/selinux/config` file to change to `permissive` mode. The `SELINUX` value should look like the example.

   ```
   SELINUX=permissive
   ```

2. Restart your system to complete the change to `permissive` mode.

   ```
   sudo reboot
   ```

Disable SELinux

When you disable SELinux, SELinux policy isn't loaded or enforced, and AVC messages aren't logged. You lose all benefits of running SELinux.

1. Make sure that the `grubby` package is installed.

   ```
   rpm -q grubby
   grubby-version
   ```

2. Configure your bootloader to add `selinux=0` to the kernel command line.

```
sudo grubby --update-kernel ALL --args selinux=0
```

3. Restart your system.

```
sudo reboot
```

4. Run the `getenforce` command to confirm that SELinux is `Disabled`.

```
$ getenforce
Disabled
```

For more information about SELinux, see the SELinux Notebook and SELinux configuration.

Comparing Amazon Linux 2023 standard (default) and minimal AMIs

You can choose to use a standard or minimal Amazon Machine Image (AMI) of Amazon Linux 2023 (AL2023).

The standard AL2023 AMI is the default Amazon Machine Image (AMI) that you create. This version comes installed with all of the most commonly used applications and tools. We recommend the standard AMI if you want to get started quickly and aren't interested in customizing the AMI.

The minimal AL2023 AMI is the basic, streamlined version that contains only the most basic tools and utilities necessary to run the OS. We recommend the minimal AMI if you want to have the smallest OS footprint possible. The minimal AMI offers slightly reduced disk space utilization and better long-term cost efficiency. The minimal AMI is suitable if you want a smaller OS and don't mind manually installing tools and applications.

For instructions on how to create an Amazon EC2 instance of the standard or minimal AMI type, see Get started with Amazon Linux 2023 (p. 28).

Kernel Live Patching on Amazon Linux 2023

You can use Kernel Live Patching for Amazon Linux 2023 (AL2023) to apply security vulnerability and critical bug patches to a running Linux kernel without rebooting or disrupting running applications. In addition, Kernel Live Patching can help improve your application's availability while also keeping your infrastructure secure and up to date.

Amazon Web Services releases two types of kernel live patches for AL2023:

- **Security updates** – Include updates for Linux common vulnerabilities and exposures (CVE). These updates are typically rated as *important* or *critical* using the Amazon Linux Security Advisory ratings. They generally map to a Common Vulnerability Scoring System (CVSS) score of 7 and higher. In some cases, Amazon might provide updates before a CVE is assigned. In these cases, the patches might appear as bug fixes.
- **Bug fixes** – Include fixes for critical bugs and stability issues that aren't associated with CVEs.

Amazon Web Services provides kernel live patches for an AL2023 kernel version for up to 3 months after its release. After this period, you must update to a later kernel version to continue to receive kernel live patches.

AL2023 kernel live patches are made available as signed RPM packages in the existing AL2023 repositories. The patches can be installed on individual instances using existing **DNF package manager** workflows. Or, they can be installed on a group of managed instances using Amazon Systems Manager.

Kernel Live Patching on AL2023 is provided at no additional cost.

Topics

Supported configurations and prerequisites

Kernel Live Patching is supported on Amazon EC2 instances and on-premises virtual machines that run AL2023.

To use Kernel Live Patching on AL2023, you must use the following:

- A 64-bit `x86_64` or `ARM64` architecture
- Kernel version `6.1`

Work with Kernel Live Patching

You can enable and use Kernel Live Patching on individual instances using the command line on the instance itself. Alternatively, you can enable and use Kernel Live Patching on a group of managed instances using Amazon Systems Manager.

The following sections explain how to enable and use Kernel Live Patching on individual instances using the command line.

For more information about enabling and using Kernel Live Patching on a group of managed instances, see Use Kernel Live Patching on AL2023 instances in the *Amazon Systems Manager User Guide*.

Topics

Enable Kernel Live Patching

Kernel Live Patching is disabled by default on AL2023. To use live patching, you must install the **DNF** plugin for Kernel Live Patching and enable the live patching functionality.

To enable Kernel Live Patching

1. Kernel live patches are available for AL2023 with kernel version `6.1`. To check your kernel version, run the following command.

```
$ sudo dnf list kernel
```

2. Install the **DNF** plugin for Kernel Live Patching.

```
$ sudo dnf install -y kpatch-dnf
```

3. Enable the **DNF** plugin for Kernel Live Patching.

```
$ sudo dnf kernel-livepatch -y auto
```

   This command also installs the latest version of the kernel live patch RPM from the configured repositories.

4. To confirm that the **DNF** plugin for kernel live patching installed successfully, run the following command.

   When you enable Kernel Live Patching, an empty kernel live patch RPM is automatically applied. If Kernel Live Patching was successfully enabled, this command returns a list that includes the initial empty kernel live patch RPM.

```
$ sudo rpm -qa | grep kernel-livepatch
dnf-plugin-kernel-livepatch-1.0-0.11.amzn2023.noarch
kernel-livepatch-6.1.12-17.42-1.0-0.amzn2023.x86_64
```

5. Install the **kpatch** package.

```
$ sudo dnf install -y kpatch-runtime
```

6. Update the **kpatch** service if it was previously installed.

```
$ sudo dnf update kpatch-runtime
```

7. Start the **kpatch** service. This service loads all of the kernel live patches upon initialization or at boot.

```
$ sudo systemctl enable kpatch.service && sudo systemctl start kpatch.service
```

View the available kernel live patches

Amazon Linux security alerts are published to the Amazon Linux Security Center. For more information about the AL2023 security alerts, including alerts for kernel live patches, see the Amazon Linux Security Center. Kernel live patches are prefixed with `ALASLIVEPATCH`. The Amazon Linux Security Center might not list kernel live patches that address bugs.

You can also discover the available kernel live patches for advisories and CVEs using the command line.

To list all available kernel live patches for advisories

Use the following command.

```
$ 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
```

To list all available kernel live patches for CVEs

Use the following command.

```
$ 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
```

Apply kernel live patches

You apply kernel live patches using the **DNF** package manager in the same way that you apply regular updates. The **DNF** plugin for Kernel Live Patching manages the kernel live patches that you apply and eliminates the need to reboot.

> **Tip**
> We recommend that you update your kernel regularly using Kernel Live Patching to achieve that it remains secure and up to date.

You can choose to apply a specific kernel live patch, or to apply any available kernel live patches along with your regular security updates.

To apply a specific kernel live patch

1. Get the kernel live patch version using one of the commands described in View the available kernel live patches (p. 23).
2. Apply the kernel live patch for your AL2023 kernel.

   ```
   $ sudo dnf install kernel-livepatch-kernel_version-package_version.amzn2023.x86_64
   ```

   For example, the following command applies a kernel live patch for AL2023 kernel version `6.1.12-17.42`

   ```
   $ sudo dnf install kernel-livepatch-6.1.12-17.42-1.0-4.amzn2023.x86_64
   ```

To apply any available kernel live patches along with your regular security updates

Use the following command.

```
$ sudo dnf update --security
```

Omit the `--security` option to include bug fixes.

> **Important**
>
> - The kernel version isn't updated after applying kernel live patches. The version is only updated to the new version after the instance is rebooted.
> - An AL2023 kernel receives kernel live patches for 3 months. After this period, no new kernel live patches are released for that kernel version.
> - To continue to receive kernel live patches after 3 months, you must reboot the instance to move to the new kernel version. The instance continues to receive kernel live patches for the next 3 months after you update it.
> - To check the support window for your kernel version, run the following command:
>
>   ```
>   $ sudo dnf kernel-livepatch support
>   ```

View the applied kernel live patches

To view the applied kernel live patches

Use the following command.

```
$ 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)
```

The command returns a list of the loaded and installed security update kernel live patches. The following is example output.

> **Note**
> A single kernel live patch can include and install multiple live patches.

Disable Kernel Live Patching

If you no longer need to use Kernel Live Patching, you can disable it at any time.

- Disable the use of livepatches:
  1. Disable the plugin:

     ```
     $ sudo dnf kernel-livepatch manual
     ```

  2. Disable the kpatch service:

     ```
     $ sudo systemctl disable --now kpatch.service
     ```

- Fully remove the livepatch tools:
  1. Remove the plugin:

     ```
     $ sudo dnf remove kpatch-dnf
     ```

  2. Remove kpatch-runtime:

     ```
     $ sudo dnf remove kpatch-runtime
     ```

  3. Remove any installed livepatches:

     ```
     $ sudo dnf remove kernel-livepatch\*
     ```

Limitations

Kernel Live Patching has the following limitations:

- While applying a kernel live patch, you can't perform hibernation, use advanced debugging tools (such as `SystemTap`, `kprobes`, and eBPF-based tools), or access `ftrace` output files used by the Kernel Live Patching infrastructure.

Using Amazon Linux 2023 on Amazon

You can set up Amazon Linux 2023 (AL2023) for use with your other Amazon Web Services. For example, you can choose an Amazon Linux image when you launch an Amazon Elastic Compute Cloud (Amazon EC2) instance. Complete the following tasks to get set up to use AL2023 with your other Amazon Web Services.

For these setup procedures, you use the Amazon Identity and Access Management (IAM) service. For complete information about IAM, see the following reference materials:

- Amazon Identity and Access Management (IAM)
- IAM User Guide

Sign up for an Amazon Web Services account

If you do not have an Amazon Web Services account, use the following procedure to create one.

To sign up for Amazon Web Services

1. Open http://www.amazonaws.cn/ and choose **Sign Up**.
2. Follow the on-screen instructions.

Amazon sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to http://www.amazonaws.cn/ and choosing **My Account**.

Secure IAM users

After you sign up for an Amazon Web Services account, safeguard your administrative user by turning on multi-factor authentication (MFA). For instructions, see Enable a virtual MFA device for an IAM user (console) in the *IAM User Guide*.

To give other users access to your Amazon Web Services account resources, create IAM users. To secure your IAM users, turn on MFA and only give the IAM users the permissions needed to perform their tasks.

For more information about creating and securing IAM users, see the following topics in the *IAM User Guide*:

- Creating an IAM user in your Amazon Web Services account
- Access management for Amazon resources
- Example IAM identity-based policies

Granting programmatic access

Users need programmatic access if they want to interact with Amazon outside of the Amazon Web Services Management Console. The Amazon APIs and the Amazon Command Line Interface require

access keys. Whenever possible, create temporary credentials that consist of an access key ID, a secret access key, and a security token that indicates when the credentials expire.

To grant users programmatic access, choose one of the following options.

Which user needs programmatic access?	To	By
IAM	Use short-term credentials to sign programmatic requests to the Amazon CLI or Amazon APIs (directly or by using the Amazon SDKs).	Following the instructions in Using temporary credentials with Amazon resources in the *IAM User Guide*.
IAM	(Not recommended) Use long-term credentials to sign programmatic requests to the Amazon CLI or Amazon APIs (directly or by using the Amazon SDKs).	Following the instructions in Managing access keys for IAM users in the *IAM User Guide*.

Get started with Amazon Linux 2023

Launching Amazon Linux 2023 using the SSM parameter and Amazon CLI

In the Amazon CLI, you can use an AMI's SSM parameter value to launch a new instance of Amazon Linux 2023 (AL2023). More specifically, use one of the dynamic SSM parameter values from the following list, and add `/aws/service/ami-amazon-linux-latest/` before the SSM parameter value/. You use this to launch the instance in the Amazon CLI.

- `al2023-ami-kernel-default-arm64` for arm64 architecture
- `al2023-ami-minimal-kernel-default-arm64` for arm64 architecture (minimal AMI)
- `al2023-ami-kernel-default-x86_64` for x86_64 architecture
- `al2023-ami-minimal-kernel-default-x86_64` for x86_64 architecture (minimal AMI)

> **Note**
> Each of the *italic* items is an example parameter. Replace them with your own information.

```
$ 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
```

The `--image-id` flag specifies the SSM parameter value.

The `--instance-type` flag specifies the type and size of the instance. This flag must be compatible with the AMI type that you selected.

The `--region` flag specifies the Amazon Web Services Region where you create your instance.

The `--key-name` flag specifies the Amazon Web Services Region's key that's used to connect to the instance. If you don't provide a key that exists in the Region where you create the instance, you can't connect to the instance using SSH.

The `--security-group-ids` flag specifies the security group that determines the access permissions for inbound and outbound network traffic.

> **Important**
> The Amazon CLI requires that you specify an existing security group that allows access to the instance from your remote machine over port TCP:22. Without a specified security group, your new instance are placed in a default security group. In a default security group, your instance can only connect with the other instances within your VPC.

For more information, see Launching, listing, and terminating Amazon EC2 instances in the *Amazon Command Line Interface User Guide*.

Launching Amazon Linux 2023 (AL2023) using the Amazon EC2 console

Use the Amazon EC2 console to launch Amazon Linux 2023 (AL2023).

Note
For Arm-based instances, AL2023 only supports instance types that use Graviton2 or later processors. AL2023 doesn't support A1 instances.

To launch an AL2023 instance from the Amazon EC2 console, follow these instructions.

1. Open EC2 Dashboard, Images, AMIs.
2. Select **Public images**.
3. Search for *al2023-ami*.

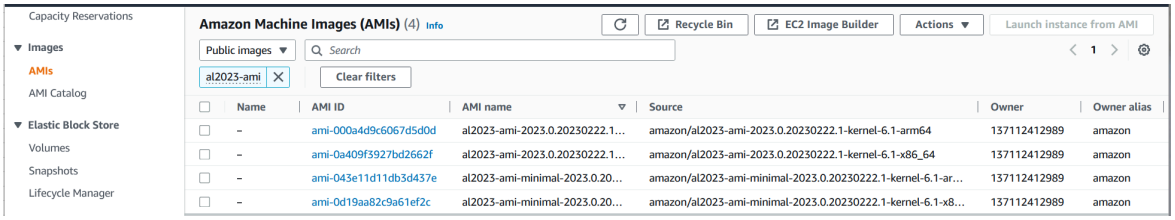
   The list includes AL2023 AMIs. Make sure that **amazon** appears in the **Owner alias** column.
4. Select an image from the list.
5. Select **Launch instance from image**, and follow the instructions to complete the launch.

An AL2023 AMI name can be interpreted by using this format:

```
'al2023-[ami || minimal-ami]-2023.0.[release build date].[build number]-kernel-
[version number]-[arm64 || x86_64]'
```

The following is an example of an AMI ID in this format.

```
'al2023-ami-2023.0.20230210.1-kernel-6.1-arm64'
```



For more information about launching Amazon EC2 instances, see Get started with Amazon EC2 Linux instances in the *Amazon EC2 User Guide for Linux Instances*.

Launching the latest Amazon Linux 2023 AMI using Amazon CloudFormation

To launch an Amazon Linux 2023 (AL2023) AMI using Amazon CloudFormation, you can use either of the following templates.

Note
The `x86_64` and `Arm64` AMIs each require different instance types. For more information, see Amazon EC2 Instance Types

`JSON` template:

```
{
  "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"
        }
      }
    }
  }
}
```

YAML template:

```
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
```

Make sure to replace the AMI parameter at the end of the "Default" section, if needed. The following parameter values are available:

- `al2023-ami-kernel-6.1-arm64` for arm64 architecture
- `al2023-ami-minimal-kernel-6.1-arm64` for arm64 architecture (minimal AMI)
- `al2023-ami-kernel-6.1-x86_64` for x86_64 architecture
- `al2023-ami-minimal-kernel-6.1-x86_64` for x86_64 architecture (minimal AMI)

The following are dynamic kernel specifications. The default kernel version automatically changes with each major kernel version update.

- `al2023-ami-kernel-default-arm64` for arm64 architecture
- `al2023-ami-minimal-kernel-default-arm64` for arm64 architecture (minimal AMI)
- `al2023-ami-kernel-default-x86_64` for x86_64 architecture
- `al2023-ami-minimal-kernel-default-x86_64` for x86_64 architecture (minimal AMI)

Launching Amazon Linux 2023 using a specific AMI ID

You can launch specific Amazon Linux 2023 (AL2023) AMI using the AMI ID. You can determine which AL2023 AMI ID is needed by looking at the AMI list in the Amazon EC2 console. Or, you can do this using

Amazon Systems Manager. If you're using Systems Manager, make sure to select the AMI alias from those that are listed in the previous section. For more information, see Query for the latest Amazon Linux AMI IDs using Amazon Systems Manager Parameter Store.

Connecting to instances

Use SSH to connect to your Amazon Linux 2023 (AL2023) instance.

Note
SSH is currently the only supported method for connecting to AL2023 instances.

Connecting via SSH

For instructions on how to use SSH to connect to an instance, see Connect to your Linux instance using SSH.

Using the Amazon Linux 2023 container image

The Amazon Linux 2023 (AL2023) container image is built from the same software components that are included in the AL2023 AMI. It's available for use in any environment as a base image for Docker workloads. If you're using the Amazon Linux AMI for applications in Amazon Elastic Compute Cloud (Amazon EC2), you can containerize your applications with the Amazon Linux container image.

Use the Amazon Linux container image in your local development environment and then push your application to Amazon using Amazon Elastic Container Service (Amazon ECS). For more information, see Using Amazon ECR images with Amazon ECS in the *Amazon Elastic Container Registry User Guide*.

The Amazon Linux container image is available . You can provide feedback for AL2023 through your designated Amazon representative or by filing an issue in the amazon-linux-2023 repo on GitHub.

To pull the AL2023 container image from Docker Hub

1. Pull the AL2023 container image using the **docker pull** command.

```
$ docker pull amazonlinux:2023
```

2. (Optional) Run the container locally.

```
$ docker run -it amazonlinux:2023 /bin/bash
```

Note
The container image of AL2023 uses only the `dnf` package manager to install software packages. This means that there's no `amazon-linux-extras` or equivalent command to use for additional software.

Managing packages and operating system updates

Unlike previous versions of Amazon Linux, Amazon Linux 2023 (AL2023) Amazon Machine Images (AMIs) are locked to a specific version of the Amazon Linux repository. To apply both security and bug fixes to an AL2023 instance, update the DNF configuration. Alternatively, launch a newer AL2023 instance. This section describes how to manage DNF packages and repositories on a running instance. It also describes how to configure DNF from a user data script to enable the latest available Amazon Linux repository at launch time. For more information, see DNF Command Reference.

Checking for available package updates

You can use the `dnf check-update` command to check for any updates for your system. For AL2023, we recommend that you add the `--releasever=`*`version-number`* option to the command.

When you add this option, DNF also checks for updates for a later version of the repository. For example, after you run the `dnf check-release-update` command, use the latest returned version as the value for the *`version-number`*.

If the instance is updated to use the latest version of the repository, a list of all the packages to update is included in the output.

Note
If you don't specify the release version with the optional flag to the `dnf check-update` command, only the currently configured repository version is checked. This means that packages in the later version of the repository aren't checked.

```
$ sudo dnf check-update --releasever=2023.0.20230210
Last metadata expiration check: 0:06:13 ago on Mon 13 Feb 2023 10:39:32 PM UTC.

bind-libs.x86_64                      32:9.16.27-1.amzn2023          amazonlinux
bind-license.noarch                   32:9.16.27-1.amzn2023          amazonlinux
bind-utils.x86_64                     32:9.16.27-1.amzn2023          amazonlinux
cloud-init.noarch                     22.2.2-1.amzn2023.1.4          amazonlinux
dnf.noarch                            4.12.0-2.amzn2023.0.1          amazonlinux
dnf-data.noarch                       4.12.0-2.amzn2023.0.1          amazonlinux
dracut.x86_64                         055-6.amzn2023.0.4             amazonlinux
dracut-config-generic.x86_64          055-6.amzn2023.0.4             amazonlinux
glib2.x86_64                          2.73.2-678.amzn2023            amazonlinux
gmp.x86_64                            1:6.2.1-2.amzn2023             amazonlinux
grep.x86_64                           3.8-1.amzn2023.0.1             amazonlinux
kpatch-runtime.noarch                 0.9.4-7.amzn2023               amazonlinux
libgcc.x86_64                         11.3.1-2.amzn2023.0.6          amazonlinux
libgomp.x86_64                        11.3.1-2.amzn2023.0.6          amazonlinux
libpkgconf.x86_64                     1.7.3-7.amzn2023.0.1           amazonlinux
libstdc++.x86_64                      11.3.1-2.amzn2023.0.6          amazonlinux
lz4-libs.x86_64                       1.9.4-1.amzn2023               amazonlinux
pkgconf.x86_64                        1.7.3-7.amzn2023.0.1           amazonlinux
pkgconf-m4.noarch                     1.7.3-7.amzn2023.0.1           amazonlinux
```

```
pkgconf-pkg-config.x86_64              1.7.3-7.amzn2023.0.1           amazonlinux
python3-dnf.noarch                     4.12.0-2.amzn2023.0.1          amazonlinux
python3-rpm.x86_64                     4.16.1.3-12.amzn2023.0.2       amazonlinux
rpm.x86_64                             4.16.1.3-12.amzn2023.0.2       amazonlinux
rpm-build-libs.x86_64                  4.16.1.3-12.amzn2023.0.2       amazonlinux
rpm-libs.x86_64                        4.16.1.3-12.amzn2023.0.2       amazonlinux
rpm-plugin-selinux.x86_64              4.16.1.3-12.amzn2023.0.2       amazonlinux
rpm-plugin-systemd-inhibit.x86_64      4.16.1.3-12.amzn2023.0.2       amazonlinux
rpm-sign-libs.x86_64                   4.16.1.3-12.amzn2023.0.2       amazonlinux
slang.x86_64                           2.3.2-9.amzn2023.0.1           amazonlinux
system-release.noarch                  2023.0.20230210-0.amzn2023     amazonlinux
systemd.x86_64                         250.8-1.amzn2023.0.1           amazonlinux
systemd-libs.x86_64                    250.8-1.amzn2023.0.1           amazonlinux
systemd-networkd.x86_64                250.8-1.amzn2023.0.1           amazonlinux
systemd-pam.x86_64                     250.8-1.amzn2023.0.1           amazonlinux
systemd-resolved.x86_64                250.8-1.amzn2023.0.1           amazonlinux
systemd-udev.x86_64                    250.8-1.amzn2023.0.1           amazonlinux
vim-common.x86_64                      2:9.0.327-1.amzn2023.0.1       amazonlinux
vim-data.noarch                        2:9.0.327-1.amzn2023.0.1       amazonlinux
vim-enhanced.x86_64                    2:9.0.327-1.amzn2023.0.1       amazonlinux
vim-filesystem.noarch                  2:9.0.327-1.amzn2023.0.1       amazonlinux
vim-minimal.x86_64                     2:9.0.327-1.amzn2023.0.1       amazonlinux
wget.x86_64                            1.21.3-1.amzn2023              amazonlinux
yum.noarch                             4.12.0-2.amzn2023.0.1          amazonlinux
```

For this command, if there are newer packages available, the return code is 100. If there aren't any newer packages available, the return code is 0. In addition, the output also lists all the packages to update.

Applying security updates using DNF and repository versions

New package updates and security updates are made available to new repository versions only. For instances that you launched from earlier AL2023 AMI versions, you must update the repository version before you can install security updates. The `dnf check-release-update` command includes an example update command that updates all the packages that are installed on the system to versions in a newer repository.

```
$ sudo dnf update --releasever=2023.0.20230210
Last metadata expiration check: 0:01:40 ago on Mon 13 Feb 2023 10:39:32 PM UTC.
Dependencies resolved.
================================================================================
 Package                    Arch   Version                    Repository   Size
================================================================================
Upgrading:
 bind-libs                  x86_64 32:9.16.27-1.amzn2023      amazonlinux 1.2 M
 bind-license               noarch 32:9.16.27-1.amzn2023      amazonlinux  16 k
 bind-utils                 x86_64 32:9.16.27-1.amzn2023      amazonlinux 202 k
 cloud-init                 noarch 22.2.2-1.amzn2023.1.4      amazonlinux 1.1 M
 dnf                        noarch 4.12.0-2.amzn2023.0.1      amazonlinux 454 k
 dnf-data                   noarch 4.12.0-2.amzn2023.0.1      amazonlinux  42 k
 dracut                     x86_64 055-6.amzn2023.0.4         amazonlinux 345 k
 dracut-config-generic      x86_64 055-6.amzn2023.0.4         amazonlinux 8.5 k
 glib2                      x86_64 2.73.2-678.amzn2023        amazonlinux 2.7 M
 gmp                        x86_64 1:6.2.1-2.amzn2023         amazonlinux 324 k
 grep                       x86_64 3.8-1.amzn2023.0.1         amazonlinux 316 k
 kpatch-runtime             noarch 0.9.4-7.amzn2023           amazonlinux  30 k
 libgcc                     x86_64 11.3.1-2.amzn2023.0.6      amazonlinux 121 k
 libgomp                    x86_64 11.3.1-2.amzn2023.0.6      amazonlinux 296 k
 libpkgconf                 x86_64 1.7.3-7.amzn2023.0.1       amazonlinux  37 k
```

```
 libstdc++                   x86_64 11.3.1-2.amzn2023.0.6      amazonlinux 758 k
 lz4-libs                    x86_64 1.9.4-1.amzn2023           amazonlinux  81 k
 pkgconf                     x86_64 1.7.3-7.amzn2023.0.1       amazonlinux  41 k
 pkgconf-m4                  noarch 1.7.3-7.amzn2023.0.1       amazonlinux  15 k
 pkgconf-pkg-config          x86_64 1.7.3-7.amzn2023.0.1       amazonlinux  11 k
 python3-dnf                 noarch 4.12.0-2.amzn2023.0.1      amazonlinux 415 k
 python3-rpm                 x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux  89 k
 rpm                         x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux 487 k
 rpm-build-libs              x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux  92 k
 rpm-libs                    x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux 311 k
 rpm-plugin-selinux          x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux  18 k
 rpm-plugin-systemd-inhibit x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux  19 k
 rpm-sign-libs               x86_64 4.16.1.3-12.amzn2023.0.2   amazonlinux  22 k
 slang                       x86_64 2.3.2-9.amzn2023.0.1       amazonlinux 410 k
 system-release              noarch 2023.0.20230210-0.amzn2023 amazonlinux  25 k
 systemd                     x86_64 250.8-1.amzn2023.0.1       amazonlinux 4.2 M
 systemd-libs                x86_64 250.8-1.amzn2023.0.1       amazonlinux 615 k
 systemd-networkd            x86_64 250.8-1.amzn2023.0.1       amazonlinux 614 k
 systemd-pam                 x86_64 250.8-1.amzn2023.0.1       amazonlinux 335 k
 systemd-resolved            x86_64 250.8-1.amzn2023.0.1       amazonlinux 277 k
 systemd-udev                x86_64 250.8-1.amzn2023.0.1       amazonlinux 1.9 M
 vim-common                  x86_64 2:9.0.327-1.amzn2023.0.1   amazonlinux 7.2 M
 vim-data                    noarch 2:9.0.327-1.amzn2023.0.1   amazonlinux  27 k
 vim-enhanced                x86_64 2:9.0.327-1.amzn2023.0.1   amazonlinux 1.8 M
 vim-filesystem              noarch 2:9.0.327-1.amzn2023.0.1   amazonlinux  21 k
 vim-minimal                 x86_64 2:9.0.327-1.amzn2023.0.1   amazonlinux 764 k
 wget                        x86_64 1.21.3-1.amzn2023          amazonlinux 813 k
 yum                         noarch 4.12.0-2.amzn2023.0.1      amazonlinux  39 k

Transaction Summary
================================================================================
Upgrade  43 Packages
...
```

You can add the `--security` option to update the packages with security features only.

```
$ sudo dnf update --releasever=2023.0.20230928 --security
Amazon Linux 2023 repository                    18 MB/s |  11 MB     00:00
Last metadata expiration check: 0:00:02 ago on Mon 13 Feb 2023 10:39:32 PM UTC.
Dependencies resolved.
================================================================================
 Package           Arch      Version                        Repository      Size
================================================================================
Upgrading:
 bind-libs         x86_64    32:9.16.27-1.amzn2023          amazonlinux    1.2 M
 bind-license      noarch    32:9.16.27-1.amzn2023          amazonlinux     16 k
 bind-utils        x86_64    32:9.16.27-1.amzn2023          amazonlinux    202 k
 gmp               x86_64    1:6.2.1-2.amzn2023             amazonlinux    324 k
 lz4-libs          x86_64    1.9.4-1.amzn2023               amazonlinux     81 k
 vim-common        x86_64    2:9.0.327-1.amzn2023.0.1       amazonlinux    7.2 M
 vim-data          noarch    2:9.0.327-1.amzn2023.0.1       amazonlinux     27 k
 vim-enhanced      x86_64    2:9.0.327-1.amzn2023.0.1       amazonlinux    1.8 M
 vim-filesystem    noarch    2:9.0.327-1.amzn2023.0.1       amazonlinux     21 k
 vim-minimal       x86_64    2:9.0.327-1.amzn2023.0.1       amazonlinux    764 k
 wget              x86_64    1.21.3-1.amzn2023              amazonlinux    813 k

Transaction Summary
================================================================================
Upgrade  11 Packages
...
```

To discover AL2023 package versions, do one or more of the following:

- Run the `dnf check-update` command.

- Subscribe to the Amazon Linux repository update SNS topic (`arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates`). For more information, see Subscribing to an Amazon SNS topic in the *Amazon Simple Notification Service Developer Guide*.
- Regularly refer to the AL2023 release notes.

When applying security updates to a running instance, it's important to make sure that DNF is pointing at the latest repository version.

Launching an instance with the latest repository version enabled

You can add DNF commands to a user-data script to control what RPM packages are installed on an Amazon Linux AMI when it's launched. In the following example, a user-data script is used to make sure that any instance launched with the user-data script has the same package updates installed.

```
#!/bin/bash
dnf update --releasever=2023.0.20230928
# Additional setup and install commands below
dnf install httpd php74 mysql80
```

You must run this script as superuser (root). To do this, run the following command.

```
$ sudo sh -c "bash nameofscript.sh"
```

For more information, see User data and shell scripts in the *Amazon EC2 User Guide for Linux Instances*.

Note
Instead of using a user-data script, launch the latest Amazon Linux AMI or a custom AMI that's based on the Amazon Linux AMI. The latest Amazon Linux AMI has all the necessary updates installed and is configured to point at a particular repository version.

Getting package support information

AL2023 incorporates many different open-source software projects. Each of these projects is managed independently from Amazon Linux and have different release and end-of-support schedules. To provide you with Amazon Linux specific information about these different packages, the DNF `supportinfo` plugin provides metadata about a package. In the following example, the **`dnf supportinfo`** command returns metadata for the `glibc` package.

```
$ 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.
```

Checking for newer repository versions

In an AL2023 instance, you can use the DNF utility to manage repositories and apply updated RPM packages. These packages are available in the Amazon Linux repositories. You can use the DNF command `dnf check-release-update` to check for new versions of the DNF repository.

```
$ 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 update --releasever=2023.0.20230210

    Release notes:
     https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes.html
```

This returns a full list of all the newer versions of the DNF repositories that are available. If nothing's returned, this means that DNF is currently configured to use the latest available version. The version of the currently installed `system-release` package sets the `releasever` DNF variable. To check the current repository version, run the following command.

```
$ rpm -q system-release --qf "%{VERSION}\n"
```

When you run DNF package transactions (such as install, update, or remove commands), a warning message notifies you of any new repository versions. For example, if you install the `httpd` package on an instance that was launched from an older version of AL2023, the following output is returned.

```
$ 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
                    noarch 18.0.0-12.amzn2023.0.3 amazonlinux  19 k
 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!
```

Adding, enabling, or disabling new repositories

To install a package from a different repository with the DNF package management system, add the repository information to the `/etc/dnf/dnf.conf` file or to its own *`repository.repo`* file in the `/etc/yum.repos.d` directory. You can do this manually. However, most DNF repositories provide their own *`repository.repo`* file at their repository URL.

> **Note**
> At this time, there are no additional repositories that can be added to AL2023. This might change in the future. Also, you can write your own packages, and make those packages available to your AL2023 enterprise environment. Before you can use the packages, you must add and enable the repository where the packages are stored.

To find out what repositories are currently enabled, you can run the following command:

```
$ 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
If you don't add the `--verbose` option flag, the output only includes the `Repo-id`, `Repo-name`, and `Repo-status` information.

To add a yum repository to `/etc/yum.repos.d` directory:

1. Find the location of the `.repo` file. In this example, the `.repo` file is at *https://www.example.com/repository.repo*.
2. Add the repository with the `dnf config-manager` command.

```
$ 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
```

After you install a repository, you must enable it as described in the next procedure.

To enable a `yum` repository in `/etc/yum.repos.d`, use the `dnf config-manager` command with the `--enable` flag and *repository* name.

```
$ sudo dnf config-manager --enable repository
```

Note
To disable a repository, use the same command syntax, but replace `--enable` with `--disable` in the command.

Adding repositories with cloud-init

In addition to adding a repository using the previous method, you can also add a new repository using the `cloud-init` framework.

To add a new package repository, we recommend the use of the following template. Consider saving this file locally.

```
#cloud-config
yum_repos:
  repository.repo:
    baseurl: https://www.example.com/
```

```
    enabled: true
    failovermethod: priority
    gpgcheck: true
    gpgkey: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EXAMPLE
    name: Example Repository
```

Note

One advantage to using `cloud-init` is that you can add a `packages:` section to your configuration file. In this section, you can include the names of the packages that you want to install. You can install packages from either the default repository or the new repository that you added in the `cloud-config` file.

For more specific information about the structure of the YAML file, see Adding a YUM repository in the *`cloud-init` documentation*.

After you set up the YAML format file, you can run it in the `cloud-init` framework in the Amazon CLI. Make sure to include the `--userdata` option and the name of the `.yml` file to call the desired operations.

```
$ 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.yml
```

Receiving notifications on new updates

You can receive notifications whenever a new Amazon Linux 2023 (AL2023) AMI is released. Notifications are published with Amazon SNS using the following topic.

- Messages are posted here when a new AL2023 AMI is published. The version of the AMI will be included in the message.

```
arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates
```

These messages can be received using several different methods. We recommend you follow this method.

1. Open the Amazon SNS console.
2. In the navigation bar, change the Amazon Web Services Region to **US East (N. Virginia)**, if necessary. You must select the Region where the SNS notification that you're subscribing to was created.
3. In the navigation pane, choose **Subscriptions**, **Create subscription**.
4. For the **Create subscription** dialog box, take the following steps.
   a. For **Topic ARN**, copy and paste the **Amazon Resource Name** (ARN).
   b. For **Protocol**, choose **Email**.
   c. For **Endpoint**, enter an email address that you can use to receive the notifications.
   d. Choose **Create subscription**.
5. You receive a confirmation email with the subject line "Amazon Notification - Subscription Confirmation". Open the email and choose **Confirm subscription** to complete your subscription.

Getting started with programming runtimes

Amazon Linux 2023 (AL2023) provides different versions of some language runtimes. We work with upstream projects who support multiple versions at the same time. Find information about how to install and manage these name-versioned packages using the `dnf` command to search and install these packages.

Security in Amazon Linux 2023

As an Amazon customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between Amazon and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – Amazon is responsible for protecting the infrastructure that runs Amazon services in the Amazon Cloud. Amazon also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the Amazon Compliance Programs. To learn about the compliance programs that apply to AL2023, see Amazon Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the Amazon service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations