Amazon Linux 2022

User Guide

Amazon Linux 2022: User Guide

Table of Contents

What is Amazon Linux 2022?

Amazon Linux 2022, the next generation of Amazon Linux from AWS. It provides a secure, stable, and high-performance runtime environment where you can develop and run cloud and enterprise applications. With Amazon Linux 2022, you get an application environment that offers long-term support with access to the latest innovations in Linux. Amazon Linux 2022 is provided at no additional charge.

Topics

Amazon Linux release cadence

A new major version of Amazon Linux is released every two years and includes five years of long-term support. Each release consists of two phases. A standard support phase lasts two years and is followed by a maintenance phase, which lasts 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.

Major version	2022	2023	2024	2025	2026	2027	2028	2029	2030
Amazon Linux 2022	Standard Support		Maintenance			EOL			
Amazon Linux 2024			Standard Support		Maintenance			EOL	
Amazon Linux 2026					Standard Support		Maintenance		
Amazon Linux 2028							Standard Support		Maintenance

Major and minor releases

- **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, glibc, 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. AWS 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.

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

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 it points to are coupled. However, you can point your running Amazon EC2 instances to newer repository versions over time to apply updates on running instances. Customers can also update by launching new instances of the latest AMIs.

Long-term support policy

Amazon Linux provides updates for all packages and maintains compatibility within a major version for customer applications that are built on Amazon Linux. Core packages, such as glibc, openssl, openssh, and the dnf package manager, receive support for the lifetime of the major Amazon Linux 2022 release. Packages that aren't part of the core packages receive support defined by their upstream sources. You can see the specific support status and dates of individual packages by running the following command.

```
# dnf supportinfo --pkg packagename
```

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

```
# dnf supportinfo --show installed
```

The full list of core packages will be finalized during the preview. If you want to see more packages included as core packages, tell us. We will evaluate as we are collecting feedback. Feedback on Amazon Linux 2022 can be provided through your designated AWS representative or Amazon Linux Discussion Forums.

Naming and versioning

There will be a minor release every three months. Each is identified by an increment from 0 to N. 0 refers to the original major release for that iteration.

For example, minor releases of Amazon Linux 2022 will have the following format.

- `2022.`**`0`**`.20220301`
- `2022.`**`1`**`.20220601`
- `2022.`**`2`**`.20220901`

The corresponding Amazon Linux 2022 AMIs will have the following format.

- `al2022-ami-2022.`**`0`**`.20220301.0-kernel-5.15-x86_64`
- `al2022-ami-2022.`**`1`**`.20220301.0-kernel-5.15-x86_64`
- `al2022-ami-2022.`**`2`**`.20220301.0-kernel-5.15-x86_64`

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

- `al2022-ami-2022.0.`**`20220301`**`.0-kernel-5.15-x86_64`
- `al2022-ami-2022.0.`**`20220410`**`.0-kernel-5.15-x86_64`
- `al2022-ami-2022.0.`**`20220520`**`.0-kernel-5.15-x86_64`

Relationship to Fedora

Major releases of Amazon Linux will be based in part on the current version of the upstream Fedora Linux distribution to use the package variety and updates that Fedora provides, though Amazon may add or replace specific packages from other non-Fedora upstreams. Linux kernel will be sourced from

kernel.org's long-term support choices and will be picked independently from Fedora. Preview version of Amazon Linux 2022 will be based on Fedora 34, and the generally available version of Amazon Linux 2022 will be based on Fedora 35.

Version locking

Note
In the default configuration, your Amazon Linux 2022 instance doesn't receive critical and important security updates automatically at launch.

Controlling the updates received from major and minor releases

With Amazon Linux 2022, you can ensure consistency of package versions and updates across your environment. You can ensure consistency among multiple instances from the same AMI. With the version locking feature, you can apply updates based on the schedule that works for you.

Amazon Linux 2022 locks to a specific version of your repository. This can be any major or minor version. The Amazon Linux 2022 Amazon Machine Image (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 older AMI, updates aren't automatically applied. Any additional packages that are installed as part of your provisioning map to the repository version that the older AMI was built from. You can ensure 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 the schedule that works for you. You can apply a specific set of desired updates on launch because these can be locked to a specific repository version.

Controlling the package updates available from the Amazon Linux 2022 repositories

When we publish a new version of the Amazon Linux 2022 repositories, all previous versions are still available. By default, the plugin for managing repository versions lock 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.

```
# dnf check-release-update
```

2. Select a version by running the following command.

```
# dnf --releasever=version update
```

At that point, `dnf install` or `dnf upgrade` only chooses packages from the selected repository version. If you don't need to control package updates, you can select the latest version. This always points to the most recent version of the Amazon Linux 2022 repositories. This restores the legacy behavior for package updates that you and existing patch workflows might expect.

For more information, see Managing packages and operating system updates (p. 14).

Security features

SELinux as a default

By default, Amazon Linux 2022 has SELinux enabled and is in Enforcing mode. SELinux is a security module that provides access control policies and is now the standard in the RPM based distributions such as Fedora, RHEL, and CentOS. It's widely used in the industry to lock down Linux servers and to help protect against malicious activity. We enforce this by default because it raises the security bar for our service. By running SELinux in Permissive mode rather than Enforcing mode, you can develop a SELinux policy for your application. We're providing a toggle to disable the configuration if you don't want to enforce SELinux.

Turning off SELinux configuration

You can change SELinux settings to permissive mode using the following command.

```
# setenforce 0
```

You can also change SELinux configuration using `cloud-init`.

Installing Amazon Linux

Topics

Setting up Amazon Linux for use with AWS services

You can set up Amazon Linux for use with your other AWS services. For example, you can choose an Amazon Linux image when you launch an Amazon Elastic Compute Cloud (Amazon EC2) instance. You can also use Amazon Linux in a container with Amazon Elastic Container Service (Amazon ECS) or Amazon Elastic Kubernetes Service (Amazon EKS). In all of these cases, there are some prerequisites that you need to complete. We describe them here.

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

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

Create an AWS account

When you sign up with AWS, you get an account number that has access to all of the services that AWS offers, including those you can use Amazon Linux with.

If you already have an AWS account, skip to the next prerequisite.

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Create an IAM user

To access an AWS service, you provide credentials. These credentials determine *authentication* (who you are) and *authorization* (which permissions you have to perform actions on AWS resources).

When you first create an Amazon Web Services (AWS) account, you begin with a single sign-in identity. That identity has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user*. When you sign in, enter the email address and password that you used to create the account.

Important
We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks. To view the tasks that require you to sign in as the root user, see Tasks that require root user credentials.

For more information about the root user and IAM user credentials, see AWS account root user credentials and IAM user credentials in the *AWS General Reference*.

To create an administrator user for yourself and add the user to an administrators group (console)

1. Sign in to the IAM console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   Note
   We strongly recommend that you adhere to the best practice of using the **`Administrator`** IAM user that follows and securely lock away the root user credentials. Sign in as the root user only to perform a few account and service management tasks.
2. In the navigation pane, choose **Users** and then choose **Add users**.
3. For **User name**, enter **`Administrator`**.
4. Select the check box next to **AWS Management Console access**. Then select **Custom password**, and then enter your new password in the text box.
5. (Optional) By default, AWS requires the new user to create a new password when first signing in. You can clear the check box next to **User must create a new password at next sign-in** to allow the new user to reset their password after they sign in.
6. Choose **Next: Permissions**.
7. Under **Set permissions**, choose **Add user to group**.
8. Choose **Create group**.
9. In the **Create group** dialog box, for **Group name** enter **`Administrators`**.
10. Choose **Filter policies**, and then select **AWS managed - job function** to filter the table contents.
11. In the policy list, select the check box for **AdministratorAccess**. Then choose **Create group**.

    Note
    You must activate IAM user and role access to Billing before you can use the `AdministratorAccess` permissions to access the AWS Billing and Cost Management console. To do this, follow the instructions in step 1 of the tutorial about delegating access to the billing console.
12. Back in the list of groups, select the check box for your new group. Choose **Refresh** if necessary to see the group in the list.
13. Choose **Next: Tags**.
14. (Optional) Add metadata to the user by attaching tags as key-value pairs. For more information about using tags in IAM, see Tagging IAM entities in the *IAM User Guide*.
15. Choose **Next: Review** to see the list of group memberships to be added to the new user. When you are ready to proceed, choose **Create user**.

You can use this same process to create more groups and users and to give your users access to your AWS account resources. To learn about using policies that restrict user permissions to specific AWS resources, see Access management and Example policies.

Important
Protect your AWS account. Never send or share your credentials with anyone outside of your organization. No one who legitimately represents Amazon will ever ask you for your credentials.

After you've created your IAM user, use its credentials to sign in to the AWS Management Console. For more information, see How IAM users sign in to your AWS account in the *IAM User Guide*.

Create an access key for your IAM user

When you use AWS programmatically, either directly or by using tools such as the AWS Command Line Interface (AWS CLI), you provide your AWS access keys so that AWS can verify your identity in programmatic calls.

Access keys consist of an access key ID and secret access key, which are used to sign programmatic requests that you make to AWS. If you don't have access keys, you can create them from the AWS Management Console. As a best practice, do not use the AWS account root user access keys for any task where it's not required. Instead, create a new administrator IAM user with access keys for yourself.

The only time that you can view or download the secret access key is when you create the keys. You cannot recover them later. However, you can create new access keys at any time. You must also have permissions to perform the required IAM actions. For more information, see Permissions required to access IAM resources in the *IAM User Guide*.

To create access keys for an IAM user

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**.
3. Choose the name of the user whose access keys you want to create, and then choose the **Security credentials** tab.
4. In the **Access keys** section, choose **Create access key**.
5. To view the new access key pair, choose **Show**. You will not have access to the secret access key again after this dialog box closes. Your credentials will look something like this:

   - Access key ID: AKIAIOSFODNN7EXAMPLE
   - Secret access key: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
6. To download the key pair, choose **Download .csv file**. Store the keys in a secure location. You will not have access to the secret access key again after this dialog box closes.

   Keep the keys confidential in order to protect your AWS account and never email them. Do not share them outside your organization, even if an inquiry appears to come from AWS or Amazon.com. No one who legitimately represents Amazon will ever ask you for your secret key.
7. After you download the `.csv` file, choose **Close**. When you create an access key, the key pair is active by default, and you can use the pair right away.

Related topics

- What is IAM? in the *IAM User Guide*
- AWS security credentials in *AWS General Reference*

Get started with Amazon Linux

Launching Amazon Linux 2022 using Amazon EC2 console

Use the Amazon EC2 console to launch Amazon Linux 2022.

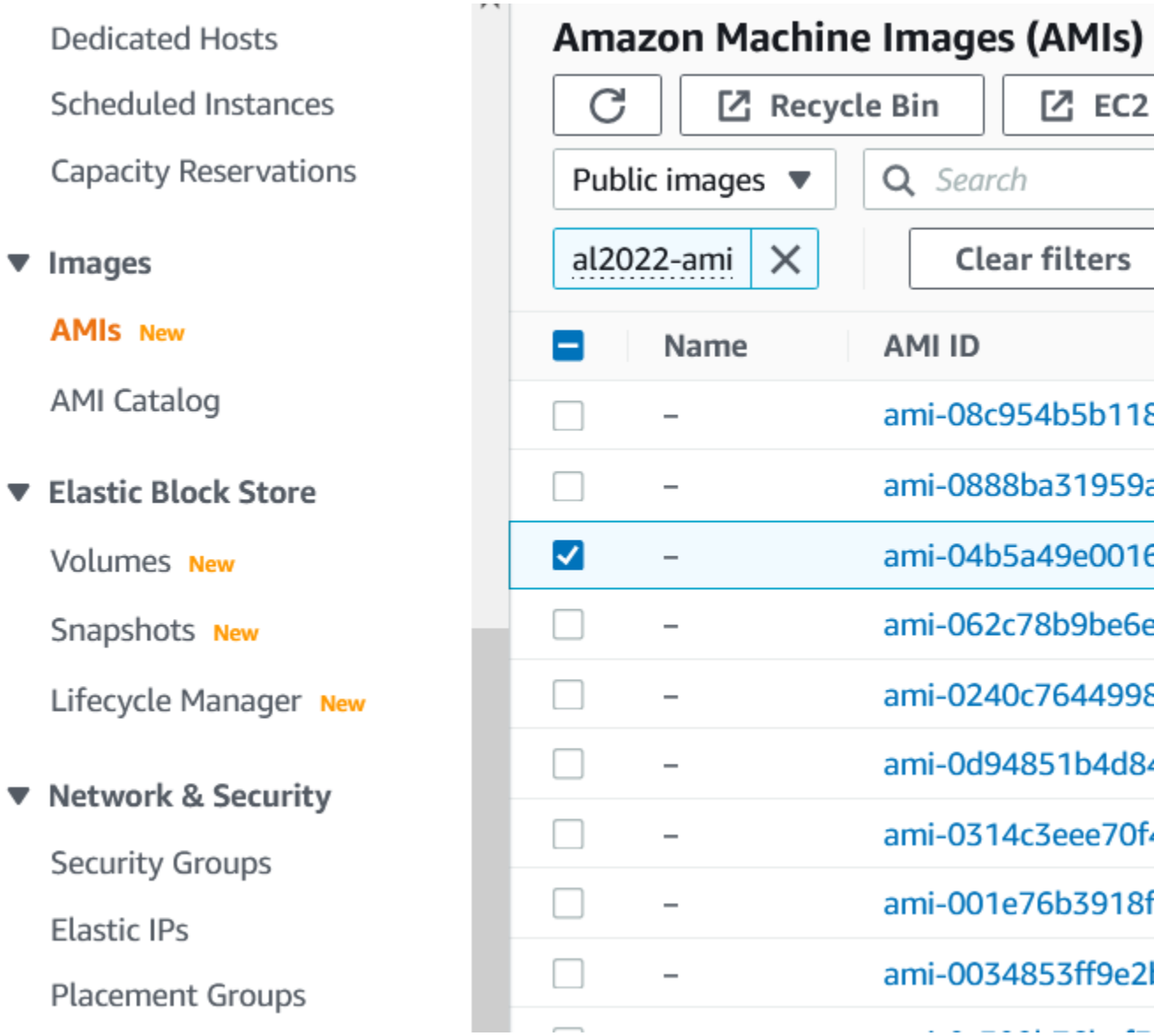
> **Note**
> Amazon Linux 2022 does not support A1 instances. Only instances based on Graviton2 and later generation processors are supported.

To launch an Amazon Linux 2022 instance from the Amazon EC2 console follow these instructions.

1. Open EC2 Dashboard, Images, AMIs.
2. Select **Public images**.
3. Search for `al2022-ami`.

   The list will include Amazon Linux 2022 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.

   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 2022 AMI using AWS CloudFormation

To launch an Amazon Linux 2022 AMI using AWS CloudFormation, you can use the following template.

```
Parameters:
  LatestAmiId:
    Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
    Default: '/aws/service/ami-amazon-linux-latest/al2022-ami-kernel-5.10-arm64'

Resources:
 Instance:
    Type: 'AWS::EC2::Instance'
    Properties:
```

```
    ImageId: !Ref LatestAmiId
```

Make sure to replace the AMI alias section if needed. The following aliases are available.

- `al2022-ami-kernel-5.10-arm64` for arm64 architecture
- `al2022-ami-minimal-kernel-5.10-arm64` for arm64 architecture (minimal AMI)
- `al2022-ami-kernel-5.10-x86_64` for x86_64 architecture
- `al2022-ami-minimal-kernel-5.10-x86_64` for x86_64 architecture (minimal AMI)

Launching Amazon Linux 2022 using a specific AMI ID

You can launch specific Amazon Linux 2022 AMI using its AMI ID. You can determine the needed Amazon Linux 2022 AMI ID by looking at the AMI list in the Amazon EC2 console or by using AWS Systems Manager. If you are using Systems Manager, make sure to indicate the AMI alias that are listed in the previous section. For more information, see Query for the latest Amazon Linux AMI IDs using AWS Systems Manager Parameter Store.

Connecting to instances

Use SSH to connect to your Amazon Linux 2022 instance.

Note
SSH is currently the only supported method for connecting to Amazon Linux 2022 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 container image

The Amazon Linux container image is built from the same software components that are included in the Amazon Linux 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 AWS 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 on Amazon ECR Public. Support for the Amazon Linux container image can be found by visiting the AWS developer forums.

To pull the Amazon Linux container image from Amazon ECR Public

1. Authenticate your Docker client to the Amazon Linux Public registry. Authentication tokens are valid for 12 hours. For more information, see Private registry authentication in the *Amazon Elastic Container Registry User Guide*.

   Note
   The **get-login-password** command is supported using the latest version of AWS CLI version 2. For more information, see Installing the AWS Command Line Interface in the *AWS Command Line Interface User Guide*.

   ```
   $ aws ecr-public get-login-password --region us-east-1 | docker login --username AWS --
   password-stdin public.ecr.aws
   ```

   The output is as follows:

   ```
   Login succeeded
   ```

2. Pull the Amazon Linux container image using the **docker pull** command. To view the Amazon Linux container image on the Amazon ECR Public Gallery, see Amazon ECR Public Gallery - amazonlinux.

   Note
   To get the latest version of the container image of Amazon Linux 2022, use the tag **:2022**. To get a specific version of the container image, you need to use the tag listed in the Amazon ECR Public Gallery -amazonlinux, for example **:2022.0.20211222.0**. The following examples use the tag **:2022** and pull the most recent available container image of Amazon Linux 2022.

   ```
   # docker pull public.ecr.aws/amazonlinux/amazonlinux:2022
   ```

3. (Optional) Run the container locally.

   ```
   # docker run -it --security-opt seccomp=unconfined public.ecr.aws/amazonlinux/
   amazonlinux:2022 /bin/bash
   ```

To pull the Amazon Linux container image from Docker Hub

1. Pull the Amazon Linux container image using the **docker pull** command.

```
# docker pull amazonlinux:2022
```

2. (Optional) Run the container locally.

```
# docker run -it amazonlinux:2022 /bin/bash
```

Managing packages and operating system updates

Unlike previous versions of Amazon Linux, Amazon Linux 2022 Amazon Machine Images (AMIs) are locked to a specific version of the Amazon Linux repository. To apply both security and bug fixes to an Amazon Linux 2022 instance, make sure to update the DNF configuration. Alternatively, launch a newer Amazon Linux 2022 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. Security in Amazon Linux (p. 21) || Checking for available package updates (p. 14)

Checking for available package updates

You can use the `dnf check-update` command to check for any updates for your system. If there's a newer version of the repository, the command returns a list of packages with available updates. For Amazon Linux 2022, we recommend that you add the `--releasever` option to the `dnf check-update` 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 `releasever`. If the instance is updated to use the latest version of the repository, a list of all the packages that need to be updated is included in the output.

```
$ sudo dnf check-update --releasever=2022.0.20220308
Last metadata expiration check: 0:01:04 ago on Thu 07 Apr 2022 11:34:13 PM UTC.

cyrus-sasl-lib.x86_64                              2.1.27-9.amzn2022
         amazonlinux
cyrus-sasl-plain.x86_64                            2.1.27-9.amzn2022
         amazonlinux
ec2-utils.noarch                                   2.0.1-1.amzn2022
         amazonlinux
expat.x86_64                                       2.4.6-1.amzn2022
         amazonlinux
freetype.x86_64                                    2.11.0-3.amzn2022
         amazonlinux
harfbuzz.x86_64                                    2.9.1-1.amzn2022
         amazonlinux
kernel.x86_64                                      5.15.25-14.106.amzn2022
         amazonlinux
kernel-headers.x86_64                              5.15.25-14.106.amzn2022
         amazonlinux
kernel-tools.x86_64                                5.15.25-14.106.amzn2022 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 that need to be updated. You can add the option for a list of the security updates only.

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 Amazon Linux2022 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=2022.0.20220315
Last metadata expiration check: 1:25:53 ago on Thu 07 Apr 2022 11:55:59 PM UTC.
Dependencies resolved.
================================================================================
 Package                    Architecture Version
 Repository         Size
================================================================================
Upgrading:
 bind-libs                  x86_64       32:9.16.22-1.amzn2022.0.1
 amazonlinux       1.2 M
 bind-license               noarch       32:9.16.22-1.amzn2022.0.1
 amazonlinux        17 k
 bind-utils                 x86_64       32:9.16.22-1.amzn2022.0.1
 amazonlinux       204 k
 coreutils                  x86_64       8.32-30.amzn2022.0.1
 amazonlinux       1.1 M
 coreutils-common           x86_64       8.32-30.amzn2022.0.1
 amazonlinux       2.0 M
 cryptsetup                 x86_64       2.3.6-1.amzn2022.0.1
 amazonlinux       186 k
 cryptsetup-libs            x86_64       2.3.6-1.amzn2022.0.1
 amazonlinux       477 k
...
```

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

```
$ sudo dnf update --releasever=2022.0.20220315 --security
Last metadata expiration check: 1:25:53 ago on Thu 07 Apr 2022 11:55:59 PM UTC.
Dependencies resolved.


cyrus-sasl-lib.x86_64                          2.1.27-9.amzn2022
         amazonlinux
cyrus-sasl-plain.x86_64                        2.1.27-9.amzn2022
         amazonlinux
expat.x86_64                                   2.4.6-1.amzn2022
         amazonlinux
freetype.x86_64                                2.11.0-3.amzn2022
         amazonlinux
harfbuzz.x86_64                                2.9.1-1.amzn2022
         amazonlinux
kernel.x86_64                                  5.15.25-14.106.amzn2022
         amazonlinux
kernel-headers.x86_64                          5.15.25-14.106.amzn2022
         amazonlinux
kernel-tools.x86_64                            5.15.25-14.106.amzn2022
         amazonlinux
libsepol.x86_64                                3.3-2.amzn2022
         amazonlinux
lua-libs.x86_64                                5.4.4-1.amzn2022
         amazonlinux
```

```
openssl-libs.x86_64                                 1:3.0.0-1.amzn2022.0.1
         amazonlinux
systemd.x86_64                                      248.10-1.amzn2022.0.1
         amazonlinux
...
```

To discover Amazon Linux package versions, you should do one or more of the following:

- Run the `dnf check-update` command.
- Subscribe to the Amazon Linux repository.
- Update SNS topic.
- Regularly refer to the Amazon Linux 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 simple user-data script is used to make sure any instance launched with the user-data script has the same package updates installed.

```
#!/bin/bash
dnf update --releasever=2022.7.240505.0
#Additional setup and install commands below
dnf install httpd php74 mysql80
```

You must run this script as super user. To do this, run `$ sudo sh -c "bash [nameofscript].sh"`.

For more information about user data, see User data and shell scripts in the Amazon Linux2022 User Guide.

> **Note**
> As an alternative to 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

Amazon Linux 2022 consists of many different open-source software projects. Each of these projects are 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

Name                 : glibc
Version              : glibc-0:2.34-7.amzn2022.x86_64
State                : installed
Support Status       : supported
```

```
Support Periods      :
from 2022-06-30      : supported
from 2027-06-30      : unsupported
Support Statement    : Amazon Linux 2022 End Of Life
Link                 : https://aws.amazon.com/amazon-linux-ami/faqs/
```

Checking for newer repository versions

In an Amazon Linux 2022 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 2022.0.20220222:
    Run the following command to update to 2022.0.20220222:

      dnf update --releasever=2022.0.20220222

    Release notes:
     https://aws.amazon.com

  Version 2022.0.20220302:
    Run the following command to update to 2022.0.20220302:

      dnf update --releasever=2022.0.20220302

    Release notes:
     https://aws.amazon.com
```

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.

```
$ sudo 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 Amazon Linux2022, the following output is returned.

```
$ sudo dnf install httpd

Last metadata expiration check: 1:56:42 ago on Sat 16 Apr 2022 12:37:08 AM UTC.
Dependencies resolved.
================================================================================
 Package                Arch      Version                   Repository      Size
================================================================================
Installing:
 httpd                  x86_64    2.4.52-1.amzn2022.0.1     amazonlinux    1.4 M
Installing dependencies:
 apr                    x86_64    1.7.0-9.amzn2022          amazonlinux    121 k
```

```
 apr-util               x86_64    1.6.1-16.amzn2022.0.1    amazonlinux     97 k
 generic-logos-httpd    noarch    18.0.0-12.amzn2022       amazonlinux     19 k
 httpd-filesystem       noarch    2.4.52-1.amzn2022.0.1    amazonlinux     13 k
 httpd-tools            x86_64    2.4.52-1.amzn2022.0.1    amazonlinux     82 k
 mailcap                noarch    2.1.49-3.amzn2022        amazonlinux     34 k

Transaction Summary
================================================================================
Install  7 Packages

Total download size: 1.7 M
Installed size: 5.5 M
Is this ok [y/N]: y
Downloading Packages:
(1/7): apr-1.7.0-9.amzn2022.x86_64.rpm          366 kB/s | 121 kB     00:00
(2/7): apr-util-1.6.1-16.amzn2022.0.1.x86_64.rp 287 kB/s |  97 kB     00:00
(3/7): httpd-2.4.52-1.amzn2022.0.1.x86_64.rpm   3.8 MB/s | 1.4 MB     00:00
(4/7): mailcap-2.1.49-3.amzn2022.noarch.rpm     530 kB/s |  34 kB     00:00
(5/7): generic-logos-httpd-18.0.0-12.amzn2022.n 473 kB/s |  19 kB     00:00
(6/7): httpd-tools-2.4.52-1.amzn2022.0.1.x86_64 753 kB/s |  82 kB     00:00
(7/7): httpd-filesystem-2.4.52-1.amzn2022.0.1.n 276 kB/s |  13 kB     00:00
--------------------------------------------------------------------------------
Total                                           3.1 MB/s | 1.7 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : apr-1.7.0-9.amzn2022.x86_64                            1/7
  Installing       : apr-util-1.6.1-16.amzn2022.0.1.x86_64                  2/7
  Installing       : httpd-tools-2.4.52-1.amzn2022.0.1.x86_64               3/7
  Running scriptlet: httpd-filesystem-2.4.52-1.amzn2022.0.1.noarch          4/7
  Installing       : httpd-filesystem-2.4.52-1.amzn2022.0.1.noarch          4/7
  Installing       : generic-logos-httpd-18.0.0-12.amzn2022.noarch          5/7
  Installing       : mailcap-2.1.49-3.amzn2022.noarch                       6/7
  Installing       : httpd-2.4.52-1.amzn2022.0.1.x86_64                     7/7
  Running scriptlet: httpd-2.4.52-1.amzn2022.0.1.x86_64                     7/7
  Verifying        : apr-util-1.6.1-16.amzn2022.0.1.x86_64                  1/7
  Verifying        : httpd-2.4.52-1.amzn2022.0.1.x86_64                     2/7
  Verifying        : apr-1.7.0-9.amzn2022.x86_64                            3/7
  Verifying        : httpd-tools-2.4.52-1.amzn2022.0.1.x86_64               4/7
  Verifying        : mailcap-2.1.49-3.amzn2022.noarch                       5/7
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2022.noarch          6/7
  Verifying        : httpd-filesystem-2.4.52-1.amzn2022.0.1.noarch          7/7

Installed:
  apr-1.7.0-9.amzn2022.x86_64
  apr-util-1.6.1-16.amzn2022.0.1.x86_64
  generic-logos-httpd-18.0.0-12.amzn2022.noarch
  httpd-2.4.52-1.amzn2022.0.1.x86_64
  httpd-filesystem-2.4.52-1.amzn2022.0.1.noarch
  httpd-tools-2.4.52-1.amzn2022.0.1.x86_64
  mailcap-2.1.49-3.amzn2022.noarch

Complete!
```

Receiving notifications on new updates

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

- Messages are posted here when a new Amazon Linux 2022 AMI is published. The version of the AMI will be included in the message.

```
arn:aws:sns:us-east-1:137112412989:amazon-linux-2022-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 AWS 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 "AWS Notification - Subscription Confirmation". Open the email and choose `Confirm subscription` to complete your subscription.

Getting started with programming runtimes

Amazon Linux 2022 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

As an AWS 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 AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS 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 AWS Compliance Programs. To learn about the compliance programs that apply to Amazon Linux, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS 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

This documentation helps you understand how to apply the shared responsibility model when using Amazon Linux. It shows you how to configure Amazon Linux to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Amazon Linux resources.

Contents

Compliance validation for Amazon Linux

Third-party auditors assess the security and compliance of Amazon Linux as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see AWS Services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using Amazon Linux is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- Architecting for HIPAA Security and Compliance Whitepaper – This whitepaper describes how companies can use AWS to create HIPAA-compliant applications.
- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – AWS Config; assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.