亚马逊云科技

User Guide

Amazon Linux 2

Amazon Linux 2: User Guide

Table of Contents

What is Amazon Linux 2? 1
Amazon Linux availability 1
Deprecated functionality 2
`compat-` packages 2
Deprecated functionality discontinued in AL1, removed in AL2 2
32-bit x86 (i686) AMIs 3
`aws-apitools-*` replaced by Amazon CLI 3
`systemd` replaces `upstart` in AL2 4
Functionality deprecated in AL2 and removed in AL2023 4
32-bit x86 (i686) Packages 5
`aws-apitools-*` replaced by Amazon CLI 5
`amazon-cloudwatch-agent` replaces `awslogs` 6
`bzr` revision control system 6
cgroup v1 6
log4j hotpatch (`log4j-cve-2021-44228-hotpatch`) 6
`lsb_release` and the `system-lsb-core` package 7
`mcrypt` 7
OpenJDK 7 (`java-1.7.0-openjdk`) 8
Python 2.7 8
`rsyslog-openssl` replaces `rsyslog-gnutls` 8
Network Information Service (NIS) / `yp` 8
Multiple domain names in Amazon VPC `create-dhcp-options` 8
Sun RPC in `glibc` 9
OpenSSH key fingerprint in `audit` log 9
`ld.gold` Linker 9
ping6 10
Prepare your migration to AL2023 11
Review the list of changes in AL2023 11
Migrate to `systemd` timers from `cron` jobs 11
AL2 Limitations 12
yum cannot verify GPG signatures made with GPG subkeys 12
Compare AL1 and AL2 13
AL1 support and EOL 13
Support for Amazon Graviton processors 13

- `systemd` replaces `upstart` as `init` system 13
- Python 2.6 and 2.7 were replaced with Python 3 13
- AL1 and AL2 AMI comparison 14
- AL1 and AL2 container comparison 43

AL2 on Amazon EC2 51

- Launch Amazon EC2 instance with AL2 AMI 51
- Find the latest AL2 AMI using Systems Manager 51
- Connect to an Amazon EC2 instance 52
- AL2 AMI boot mode 53
- Package repository 53
  - Security updates 54
  - Repository configuration 56
- Using cloud-init on AL2 56
  - Supported user data formats 58
- Configure instances 59
  - Common configuration scenarios 59
  - Manage software 60
  - Processor state control 68
  - I/O scheduler 76
  - Change the hostname 78
  - Set up dynamic DNS 82
  - Configure network interfaces using ec2-net-utils 84
- User provided kernels 86
  - HVM AMIs (GRUB) 86
  - Paravirtual AMIs (PV-GRUB) 87
- AL2 AMI release notifications 93
- Configure the MATE desktop connection 96
  - Prerequisite 97
  - Configure the RDP connection 98
- AL2 Tutorials 100
  - Install LAMP on AL2 100
  - Configure SSL/TLS on AL2 113
  - Host a WordPress blog on AL2 131

AL2 outside Amazon EC2 143

- Run AL2 on premises 143
  - Step 1: Prepare the `seed.iso` boot image 143

Step 2: Download the AL2 VM image 146
Step 3: Boot and connect to your new VM 146
Identify AL2 instances 149
Identify Amazon Linux images 149
AL2 149
Amazon Linux AMI 150
Amazon integration in AL2 151
Amazon command line tools 151
Programming languages and runtimes 152
C/C++ and Fortran 152
Go in AL2 153
Java 153
Perl 154
Perl modules 154
PHP 154
Migrating from earlier PHP 8.x versions 154
Migrating from PHP 7.x versions 155
Python in AL2 155
Rust in AL2 155
AL2 kernel 157
AL2 supported kernels 157
Kernel Live Patching 158
Supported configurations and prerequisites 159
Work with Kernel Live Patching 161
Limitations 166
Frequently asked questions 167
AL2 Extras 168
List of Amazon Linux 2 Extras 169
AL2 Reserved Users and Groups 174
List of Amazon Linux 2 Reserved Users 174
List of Amazon Linux 2 Reserved Groups 184
AL2 Source Packages 200
Security and Compliance 201
Enable FIPS Mode on AL2 201

What is Amazon Linux 2?

Amazon Linux 2 (AL2) is a Linux operating system from Amazon Web Services (Amazon). AL2 is designed to provide a stable, secure, and high-performing environment for applications running on Amazon EC2. It also includes packages that enable efficient integration with Amazon, including launch configuration tools and many popular Amazon libraries and tools. Amazon provides ongoing security and maintenance updates for all instances running AL2. Many applications developed on CentOS, and similar distributions, run on AL2. AL2 is provided at no additional charge.

Note

AL2 is no longer the current version of Amazon Linux. AL2023 is the successor to AL2. For more information, see Comparing AL2 and AL2023 and the list of Package changes in AL2023 in the AL2023 User Guide.

Amazon Linux availability

Amazon provides AL2023, AL2, and Amazon Linux 1 (AL1, formerly Amazon Linux AMI). If you are migrating from another Linux distribution to Amazon Linux, we recommend that you migrate to AL2023.

Note

Standard support for AL1 ended on December 31, 2020. The AL1 maintenance support phase ended December 31, 2023. For more information about AL1 EOL and maintenance support, see the blog post Update on Amazon Linux AMI end-of-life.

For more information about Amazon Linux, see AL2023, AL2, and AL1.

For Amazon Linux container images, see Amazon Linux container image in the *Amazon Elastic Container Registry User Guide*.

Deprecated functionality in AL2

The following sections describe functionality supported in AL2 and not present in AL2023. This is functionality such as features and packages that are present in AL2, but not in AL2023 and will not be added to AL2023. See the AL2 documentation for how long this functionality is supported in AL2.

`compat-` packages

Any packages in AL2 with the prefix of `compat-` are provided for binary compatibility with older binaries that have not yet been rebuilt for modern versions of the package. Each new major version of Amazon Linux will not carry forward any `compat-` package from prior releases.

All `compat-` packages in a release of Amazon Linux (such as AL2) are discontinued, and not present in the subsequent version (such as AL2023). We strongly recommend that software is rebuilt against updated versions of the libraries.

Deprecated functionality discontinued in AL1, removed in AL2

This section describes functionality that is available in AL1, and is no longer available in AL2.

Note

As part of the maintenance support phase of AL1, some packages had an end-of-life (EOL) date earlier than the EOL of AL1. For more information, see AL1 Package support statements.

Note

Some AL1 functionality was discontinued in earlier releases. For information, see the AL1 Release Notes.

Topics

- 32-bit x86 (i686) AMIs

- aws-apitools-* replaced by Amazon CLI
- systemd replaces upstart in AL2

32-bit x86 (i686) AMIs

As part of the 2014.09 release of AL1, Amazon Linux announced that it would be the last release to produce 32-bit AMIs. Therefore, starting from the 2015.03 release of AL1, Amazon Linux no longer supports running the system in 32-bit mode. AL2 offers limited runtime support for 32-bit binaries on x86-64 hosts and does not provide development packages to enable the building of new 32-bit binaries. AL2023 no longer includes any 32-bit user space packages. We recommend that users complete their transition to 64-bit code before migrating to AL2023.

If you need to run 32-bit binaries on AL2023, it is possible to use the 32-bit userspace from AL2 inside an AL2 container running on top of AL2023.

`aws-apitools-*` replaced by Amazon CLI

Before the release of the Amazon CLI in September 2013, Amazon made a set of command line utilities available, implemented in Java, which allowed users to make Amazon EC2 API calls. These tools were discontinued in 2015, with the Amazon CLI becoming the preferred way to interact with Amazon EC2 APIs from the command line. The set of command line utilities includes the following `aws-apitools-*` packages.

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

Upstream support for the `aws-apitools-*` packages ended in March of 2017. Despite the lack of upstream support, Amazon Linux continued to ship some of these command line utilities, such as `aws-apitools-ec2`, to provide backward compatibility for users. The Amazon CLI is a more robust and complete tool than the `aws-apitools-*` packages as it is actively maintained and provides a means of using all Amazon APIs.

The aws-apitools-* packages were deprecated in March 2017 and will not be receiving further updates. All users of any of these packages should migrate to the Amazon CLI as soon as possible. These packages are not present in AL2023.

AL1 also provided the aws-apitools-iam and aws-apitools-rds packages, which were deprecated in AL1, and are not present in Amazon Linux from AL2 onward.

systemd replaces upstart in AL2

AL2 was the first Amazon Linux release to use the systemd init system, replacing upstart in AL1. Any upstart specific configuration must be changed as part of the migration from AL1 to a newer version of Amazon Linux. It is not possible to use systemd on AL1, so moving from upstart to systemd can only be done as part of moving to a more recent major version of Amazon Linux such as AL2 or AL2023.

Functionality deprecated in AL2 and removed in AL2023

This section describes functionality that is available in AL2, and no longer available in AL2023.

Topics

- 32-bit x86 (i686) Packages
- aws-apitools-* replaced by Amazon CLI
- awslogs deprecated in favor of unified Amazon CloudWatch Logs agent
- bzr revision control system
- cgroup v1
- log4j hotpatch (log4j-cve-2021-44228-hotpatch)
- lsb_release and the system-lsb-core package
- mcrypt
- OpenJDK 7 (java-1.7.0-openjdk)
- Python 2.7
- rsyslog-openssl replaces rsyslog-gnutls
- Network Information Service (NIS) / yp
- Multiple domain names in Amazon VPC create-dhcp-options

- Sun RPC in glibc
- OpenSSH key fingerprint in audit log
- ld.gold Linker
- ping6

32-bit x86 (i686) Packages

As part of the 2014.09 release of AL1, we announced that it would be the last release to produce 32-bit AMIs. Therefore, starting from the 2015.03 release of AL1, Amazon Linux no longer supports running the system in 32-bit mode. AL2 provides limited runtime support for 32-bit binaries on x86-64 hosts and doesn't provide development packages to enable the building of new 32-bit binaries. AL2023 no longer includes any 32-bit userspace packages. We recommend customers complete their transition to 64-bit code.

If you need to run 32-bit binaries on AL2023, it is possible to use the 32-bit userspace from AL2 inside an AL2 container running on top of AL2023.

`aws-apitools-*` replaced by Amazon CLI

Prior to release of the Amazon CLI in September 2013, Amazon made a set of command line utilities available, implemented in Java, which allowed customers to make Amazon EC2 API calls. These tools were deprecated in 2015, with the Amazon CLI becoming the preferred way to interact with Amazon EC2 APIs from the command line. This includes the following `aws-apitools-*` packages.

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

Upstream support for the `aws-apitools-*` packages ended in March of 2017. Despite the lack of upstream support, Amazon Linux continued to ship some of these command line utilities (such as `aws-apitools-ec2`) in order to provide backwards compatibility for customers. The Amazon

CLI is a more robust and complete tool than the `aws-apitools-*` packages as it is actively maintained and provides a means of using all Amazon APIs.

The `aws-apitools-*` packages were deprecated in March 2017 and will not be receiving further updates. All users of any of these packages should migrate to the Amazon CLI as soon as possible. These packages are not present in AL2023.

`awslogs` deprecated in favor of unified Amazon CloudWatch Logs agent

The `awslogs` package is deprecated in AL2 and is no longer present in AL2023. It is replaced by the unified CloudWatch Logs agent, available in the `amazon-cloudwatch-agent` package. For more information, see the Amazon CloudWatch Logs User Guide.

`bzr` revision control system

The GNU Bazaar (`bzr`) revision control system is discontinued in AL2 and no longer present in AL2023.

Users of `bzr` are advised to migrate their repositories to `git`.

cgroup v1

AL2023 moves to Unified Control Group hierarchy (cgroup v2), whereas AL2 uses cgroup v1. As AL2 doesn't support cgroup v2, this migration needs to be completed as part of moving to AL2023.

log4j hotpatch (`log4j-cve-2021-44228-hotpatch`)

Note

The `log4j-cve-2021-44228-hotpatch` package is deprecated in AL2 and removed in AL2023.

In response to CVE-2021-44228, Amazon Linux released an RPM packaged version of the Hotpatch for Apache Log4j for AL1 and AL2. In the announcement of the addition of the hotpatch to Amazon Linux , we noted that "Installing the hotpatch is not a replacement for updating to a log4j version that mitigates CVE-2021-44228 or CVE-2021-45046.".

The hotpatch was a mitigation to allow time to patch `log4j`. The first general availability release of AL2023 was 15 months after CVE-2021-44228, so AL2023 doesn't ship with the hotpatch (enabled or not).

Customers running their own `log4j` versions on Amazon Linux are advised to ensure they have updated to versions not affected by CVE-2021-44228 or CVE-2021-45046.

`lsb_release` and the `system-lsb-core` package

Historically, some software invoked the `lsb_release` command (provided in AL2 by the `system-lsb-core` package) to get information about the Linux distribution that it was being run on. The Linux Standards Base (LSB) introduced this command and Linux distributions adopted it. Linux distributions have evolved to use the simpler standard of holding this information in `/etc/os-release` and other related files.

The `os-release` standard comes out of `systemd`. For more information, see systemd os-release documentation.

AL2023 doesn't ship with the `lsb_release` command, and doesn't include the `system-lsb-core` package. Software should complete the transition to the `os-release` standard to maintain compatibility with Amazon Linux and other major Linux distributions.

`mcrypt`

The `mcrypt` library and associated PHP extension was deprecated in AL2, and is no longer present in AL2023.

Upstream PHP deprecated the `mcrypt` extension in PHP 7.1 which was first released in December 2016 and had its final release in October 2019.

The upstream `mcrypt` library last made a release in 2007, and has not made the migration from `cvs` revision control that SourceForge required for new commits in 2017, with the most recent commit (and only for 3 years prior) being from 2011 removing the mention of the project having a maintainer.

Any remaining users of `mcrypt` are advised to port their code to OpenSSL, as `mcrypt` will not be added to AL2023.

OpenJDK 7 (java-1.7.0-openjdk)

Note

AL2023 provides several versions of Amazon Corretto to support Java based workloads. The OpenJDK 7 packages are deprecated in AL2, and no longer present in AL2023. The oldest JDK available in AL2023 is provided by Corretto 8.

For more information about Java on Amazon Linux, see Java in AL2.

Python 2.7

Note

AL2023 removed Python 2.7, so any OS components requiring Python are written to work with Python 3. To continue to use a version of Python provided by and supported by Amazon Linux, convert Python 2 code to Python 3.

For more information about Python on Amazon Linux, see Python in AL2.

rsyslog-openssl replaces rsyslog-gnutls

The rsyslog-gnutls package is deprecated in AL2, and no longer present in AL2023. The rsyslog-openssl package should be a drop-in replacement for any usage of the rsyslog-gnutls package.

Network Information Service (NIS) / yp

The Network Information Service (NIS), originally called Yellow Pages or YP is deprecated in AL2, and no longer present in AL2023. This includes the following packages: ypbind, ypserv, and yp-tools. Other packages that integrate with NIS have this functionality removed in AL2023.

Multiple domain names in Amazon VPC create-dhcp-options

In Amazon Linux 2, it was possible to pass multiple domain names in the domain-name parameter to create-dhcp-options, which would result in /etc/resolv.conf containing something like search foo.example.com bar.example.com. The Amazon VPC DHCP server sends the list of provided domain names using DHCP option 15, which only supports a single domain name (see RFC 2132 section 3.17). Since AL2023 uses systemd-networkd for network configuration, which follows the RFC, this accidental feature in AL2 is not present on AL2023

The Amazon CLI and Amazon VPC documentation has this to say: "Some Linux operating systems accept multiple domain names separated by spaces. However, Windows and other Linux operating systems treat the value as a single domain, which results in unexpected behavior. If your DHCP option set is associated with a Amazon VPC that has instances running operating systems that treat the value as a single domain, specify only one domain name. "

On these systems, such as AL2023, specifying two domains using DHCP option 15 (which only allows one), and since the space character is invalid in domain names, this will result in the space character being encoded as 032, resulting in /etc/resolv.conf containing search foo.exmple.com032bar.example.com.

In order to support multiple domain names, a DHCP server should use DHCP Option 119 (see RFC 3397, section 2). See the Amazon VPC User Guide for when this is supported by the Amazon VPC DHCP server.

Sun RPC in glibc

The implementation of Sun RPC in glibc is deprecated in AL2 and removed in AL2023. Customers are advised to move to using the libtirpc library (available in AL2 and AL2023) if Sun RPC functionality is required. Adopting libtirpc also enables applications to support IPv6.

This change reflects the broader community's adoption of upstream glibc removing this functionality, for example the Removal of Sun RPC interfaces from glibc in Fedora and a similar change in Gentoo.

OpenSSH key fingerprint in audit log

Later in the lifecyle of AL2, a patch was added to the OpenSSH package to emit the key fingerprint used to authenticate. This functionality is not present in AL2023.

ld.gold Linker

The ld.gold linker is available in AL2, and is removed in AL2023. Customers building software that explicitly references the gold linker should migrate to the regular (ld.bfd) linker.

The upstream GNU Binutils release notes for version 2.44 (released Feb 2025) document the removal of `ld.gold`: "In a change to our previous practice, in this release the binutils-2.44.tar tarball does not contain the sources for the gold linker. This is because the gold linker is now deprecated and will eventually be removed unless volunteers step forward and offer to continue development and maintenance."

ping6

In AL2023, the regular `ping` utility natively supports IPv6, and the separate `/bin/ping6` is no longer required. In AL2023, `/usr/sbin/ping6` is a symlink to the `/usr/bin/ping` executable.

This change follows the broader community's adoption of newer `iputils` versions which provide this functionality, for example the Ping IPv6 change in Fedora.

Prepare your migration to AL2023

You can prepare your move to AL2023 while you continue to use AL2.

Topics

- Review the list of changes in AL2023
- Migrate to systemd timers from cron jobs

Review the list of changes in AL2023

The AL2023 documentation contains a detailed list of changes that have been implemented since AL2. This information is located in the Comparing AL2 and AL2023 section. There is also a comprehensive list of software package changes located in the Package changes in AL2023 section.

AL2023 doesn't include `amazon-linux-extras`. Instead, it provides namespaced packages where multiple versions are provided. Because many packages are updated in AL2023, the base versions in AL2023 might be later than the versions that you are getting from `amazon-linux-extras`.

Note

We recommend that you don't run `amazon-linux-extras`, because it is EOL.

After you review these sections in the documentation, you can determine if there are changes in AL2023 that might require you to adapt your environment for the migration. For example, you might need to finally migrate a Python 2.7 script to Python 3.

Migrate to `systemd` timers from `cron` jobs

By default, `cron` is not installed in AL2023. You can migrate your `cron` jobs to `systemd` timers in AL2 in preparation for migrating to AL2023. `systemd` has many advantages, such as more precise control over when timers are run and improved logging.

AL2 Limitations

The following topics cover various limitations of AL2, and if they have been resolved in a newer version of Amazon Linux.

Topics

- yum cannot verify GPG signatures made with GPG subkeys

yum cannot verify GPG signatures made with GPG subkeys

The version of the `rpm` package manager in AL2 is from before `rpm` added support for verifying package signatures made with GPG subkeys. If you are creating packages to be compatible with AL2, you will need to ensure you use GPG signing keys which are compatible with the `rpm` which is part of AL2

In order to ensure backwards compatibility for existing users, the version of `rpm` in AL2 receives only security backports.

The version of `rpm` in AL2023 includes support for verifying package signatures made with GPG subkeys.

Compare AL1 and AL2

The following topics describe key differences between AL1 and AL2. They also contain information about lifespan and support, and package changes.

Topics

- AL1 support and EOL
- Support for Amazon Graviton processors
- systemd replaces upstart as init system
- Python 2.6 and 2.7 were replaced with Python 3
- Comparing packages installed on AL1 and AL2 AMIs
- Comparing packages installed on AL1 and AL2 base container images

AL1 support and EOL

AL1 is now EOL. AL1 ended standard support as of December 31, 2020, and was in a maintenance support phase until December 31, 2023.

We recommend upgrading to the latest Amazon Linux version.

Support for Amazon Graviton processors

AL2 introduced support for Graviton processors. AL2023 is further optimized for Graviton processors.

`systemd` replaces `upstart` as `init` system

In AL2, `systemd` replaced `upstart` as the `init` system.

Python 2.6 and 2.7 were replaced with Python 3

Although AL1 marked Python 2.6 as EOL with the 2018.03 release, the packages were still in the repositories to install. AL2 shipped with Python 2.7 as the earliest supported Python version.

AL2023 completes the transition to Python 3, and no Python 2.x versions are included in the repositories.

Comparing packages installed on AL1 and AL2 AMIs

Package	AL1 AMI	AL2 AMI
GeoIP		1.5.0
PyYAML		3.10
acl	2.2.49	2.2.51
acpid	2.0.19	2.0.19
alsa-lib	1.0.22	
amazon-linux-extras		2.0.3
amazon-linux-extras-yum-plugin		2.0.3
amazon-ssm-agent	3.2.1705.0	3.2.1705.0
at	3.1.10	3.1.13
attr	2.4.46	2.4.46
audit	2.6.5	2.8.1
audit-libs	2.6.5	2.8.1
authconfig	6.2.8	6.2.8
aws-amitools-ec2	1.5.13	
aws-cfn-bootstrap	1.4	2.0
aws-cli	1.18.107	

Package	AL1 AMI	AL2 AMI
awscli		1.18.147
basesystem	10.0	10.0
bash	4.2.46	4.2.46
bash-completion		2.1
bc	1.06.95	1.06.95
bind-export-libs		9.11.4
bind-libs	9.8.2	9.11.4
bind-libs-lite		9.11.4
bind-license		9.11.4
bind-utils	9.8.2	9.11.4
binutils	2.27	2.29.1
blktrace		1.0.5
boost-date-time		1.53.0
boost-system		1.53.0
boost-thread		1.53.0
bridge-utils		1.5
bzip2	1.0.6	1.0.6
bzip2-libs	1.0.6	1.0.6
ca-certificates	2023.2.62	2023.2.62
checkpolicy	2.1.10	

Package	AL1 AMI	AL2 AMI
chkconfig	1.3.49.3	1.7.4
chrony		4.2
cloud-disk-utils	0.27	
cloud-init	0.7.6	19.3
cloud-utils-growpart		0.31
copy-jdk-configs	3.3	
coreutils	8.22	8.22
cpio	2.10	2.12
cracklib	2.8.16	2.9.0
cracklib-dicts	2.8.16	2.9.0
cronie	1.4.4	1.4.11
cronie-anacron	1.4.4	1.4.11
crontabs	1.10	1.11
cryptsetup	1.6.7	1.7.4
cryptsetup-libs	1.6.7	1.7.4
curl	7.61.1	8.3.0
cyrus-sasl	2.1.23	
cyrus-sasl-lib	2.1.23	2.1.26
cyrus-sasl-plain	2.1.23	2.1.26
dash	0.5.5.1	

Package	AL1 AMI	AL2 AMI
db4	4.7.25	
db4-utils	4.7.25	
dbus	1.6.12	1.10.24
dbus-libs	1.6.12	1.10.24
dejavu-fonts-common	2.33	
dejavu-sans-fonts	2.33	
dejavu-serif-fonts	2.33	
device-mapper	1.02.135	1.02.170
device-mapper-event	1.02.135	1.02.170
device-mapper-event-libs	1.02.135	1.02.170
device-mapper-libs	1.02.135	1.02.170
device-mapper-persistent-data	0.6.3	0.7.3
dhclient	4.1.1	4.2.5
dhcp-common	4.1.1	4.2.5
dhcp-libs		4.2.5
diffutils	3.3	3.3
dmidecode		3.2
dmraid	1.0.0.rc16	1.0.0.rc16
dmraid-events	1.0.0.rc16	1.0.0.rc16
dosfstools		3.0.20

Package	AL1 AMI	AL2 AMI
dracut	004	033
dracut-config-ec2		2.0
dracut-config-generic		033
dracut-modules-growroot	0.20	
dump	0.4	
dyninst		9.3.1
e2fsprogs	1.43.5	1.42.9
e2fsprogs-libs	1.43.5	1.42.9
ec2-hibinit-agent	1.0.0	1.0.2
ec2-instance-connect		1.1
ec2-instance-connect-selinux		1.1
ec2-net-utils	0.7	1.7.3
ec2-utils	0.7	1.2
ed	1.1	1.9
elfutils-default-yama-scope		0.176
elfutils-libelf	0.168	0.176
elfutils-libs		0.176
epel-release	6	
ethtool	3.15	4.8
expat	2.1.0	2.1.0

Package	AL1 AMI	AL2 AMI
file	5.37	5.11
file-libs	5.37	5.11
filesystem	2.4.30	3.2
findutils	4.4.2	4.5.11
fipscheck	1.3.1	1.4.1
fipscheck-lib	1.3.1	1.4.1
fontconfig	2.8.0	
fontpackages-filesystem	1.41	
freetype	2.3.11	2.8
fuse-libs	2.9.4	2.9.2
gawk	3.1.7	4.0.2
gdbm	1.8.0	1.13
gdisk	0.8.10	0.8.10
generic-logos	17.0.0	18.0.0
get_reference_source	1.2	
gettext		0.19.8.1
gettext-libs		0.19.8.1
giflib	4.1.6	
glib2	2.36.3	2.56.1
glibc	2.17	2.26

Package	AL1 AMI	AL2 AMI
glibc-all-langpacks		2.26
glibc-common	2.17	2.26
glibc-locale-source		2.26
glibc-minimal-langpack		2.26
gmp	6.0.0	6.0.0
gnupg2	2.0.28	2.0.22
gpgme	1.4.3	1.3.2
gpm-libs	1.20.6	1.20.7
grep	2.20	2.20
groff	1.22.2	
groff-base	1.22.2	1.22.2
grub	0.97	
grub2		2.06
grub2-common		2.06
grub2-efi-x64-ec2		2.06
grub2-pc		2.06
grub2-pc-modules		2.06
grub2-tools		2.06
grub2-tools-minimal		2.06
grubby	7.0.15	8.28

Package	AL1 AMI	AL2 AMI
gssproxy		0.7.0
gzip	1.5	1.5
hardlink		1.3
hesiod	3.1.0	
hibagent	1.0.0	1.1.0
hmaccalc	0.9.12	
hostname		3.13
hunspell		1.3.2
hunspell-en		0.20121024
hunspell-en-GB		0.20121024
hunspell-en-US		0.20121024
hwdata	0.233	0.252
info	5.1	5.1
initscripts	9.03.58	9.49.47
iproute	4.4.0	5.10.0
iptables	1.4.21	1.8.4
iptables-libs		1.8.4
iputils	20121221	20180629
irqbalance	1.5.0	1.7.0
jansson		2.10

Package	AL1 AMI	AL2 AMI
java-1.7.0-openjdk	1.7.0.321	
javapackages-tools	0.9.1	
jbigkit-libs		2.0
jpackage-utils	1.7.5	
json-c		0.11
kbd	1.15	1.15.5
kbd-legacy		1.15.5
kbd-misc	1.15	1.15.5
kernel	4.14.326	5.10.199
kernel-tools	4.14.326	5.10.199
keyutils	1.5.8	1.5.8
keyutils-libs	1.5.8	1.5.8
kmod	14	25
kmod-libs	14	25
kpartx	0.4.9	0.4.9
kpatch-runtime		0.9.4
krb5-libs	1.15.1	1.15.1
langtable		0.0.31
langtable-data		0.0.31
langtable-python		0.0.31

Package	AL1 AMI	AL2 AMI
lcms2	2.6	
less	436	458
libICE	1.0.6	
libSM	1.2.1	
libX11	1.6.0	
libX11-common	1.6.0	
libXau	1.0.6	
libXcomposite	0.4.3	
libXext	1.3.2	
libXfont	1.4.5	
libXi	1.7.2	
libXrender	0.9.8	
libXtst	1.2.2	
libacl	2.2.49	2.2.51
libaio	0.3.109	0.3.109
libassuan	2.0.3	2.1.0
libattr	2.4.46	2.4.46
libbasicobjects		0.1.1
libblkid	2.23.2	2.30.2
libcap	2.16	2.54

Package	AL1 AMI	AL2 AMI
libcap-ng	0.7.5	0.7.5
libcap54	2.54	
libcgroup	0.40.rc1	
libcollection		0.7.0
libcom_err	1.43.5	1.42.9
libconfig		1.4.9
libcroco		0.6.12
libcrypt		2.26
libcurl	7.61.1	8.3.0
libdaemon		0.14
libdb		5.3.21
libdb-utils		5.3.21
libdrm		2.4.97
libdwarf		20130207
libedit	2.11	3.0
libestr		0.1.9
libevent	2.0.21	2.0.21
libfastjson		0.99.4
libfdisk		2.30.2
libffi	3.0.13	3.0.13

Package	AL1 AMI	AL2 AMI
libfontenc	1.0.5	
libgcc		7.3.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.5.3
libgomp		7.3.1
libgpg-error	1.11	1.12
libgssglue	0.1	
libicu	50.2	50.2
libidn	1.18	1.28
libidn2	2.3.0	2.3.0
libini_config		1.3.1
libjpeg-turbo	1.2.90	2.0.90
libmetalink		0.1.3
libmnl	1.0.3	1.0.3
libmount	2.23.2	2.30.2
libnetfilter_conntrack	1.0.4	1.0.6
libnfnetlink	1.0.1	1.0.1
libnfsidmap	0.25	0.25
libnghttp2	1.33.0	1.41.0
libnih	1.0.1	

Package	AL1 AMI	AL2 AMI
libnl	1.1.4	
libnl3		3.2.28
libnl3-cli		3.2.28
libpath_utils		0.2.1
libpcap		1.5.3
libpciaccess		0.14
libpipeline	1.2.3	1.2.3
libpng	1.2.49	1.5.13
libpsl	0.6.2	
libpwquality	1.2.3	1.2.3
libref_array		0.1.5
libseccomp		2.4.1
libselinux	2.1.10	2.5
libselinux-utils	2.1.10	2.5
libsemanage	2.1.6	2.5
libsepol	2.1.7	2.5
libsmartcols	2.23.2	2.30.2
libss	1.43.5	1.42.9
libssh2	1.4.2	1.4.3
libsss_idmap		1.16.5

Package	AL1 AMI	AL2 AMI
libsss_nss_idmap		1.16.5
libstdc++		7.3.1
libstdc++72	7.2.1	
libstoragemgmt		1.6.1
libstoragemgmt-python		1.6.1
libstoragemgmt-python-clibs		1.6.1
libsysfs	2.1.0	2.1.0
libtasn1	2.3	4.10
libteam		1.27
libtiff		4.0.3
libtirpc	0.2.4	0.2.4
libudev	173	
libunistring	0.9.3	0.9.3
libuser	0.60	0.60
libutempter	1.1.5	1.1.6
libuuid	2.23.2	2.30.2
libverto	0.2.5	0.2.5
libverto-libevent		0.2.5
libwebp		0.3.0
libxcb	1.11	

Package	AL1 AMI	AL2 AMI
libxml2	2.9.1	2.9.1
libxml2-python		2.9.1
libxml2-python27	2.9.1	
libxslt	1.1.28	
libyaml	0.1.6	0.1.4
lm_sensors-libs		3.4.0
log4j-cve-2021-44228-hotpatch	1.3	
logrotate	3.7.8	3.8.6
lsof	4.82	4.87
lua	5.1.4	5.1.4
lvm2	2.02.166	2.02.187
lvm2-libs	2.02.166	2.02.187
lz4		1.7.5
mailcap	2.1.31	
make	3.82	3.82
man-db	2.6.3	2.6.3
man-pages	4.10	3.53
man-pages-overrides		7.5.2
mariadb-libs		5.5.68
mdadm	3.2.6	4.0

Package	AL1 AMI	AL2 AMI
microcode_ctl	2.1	2.1
mingetty	1.08	
mlocate		0.26
mtr		0.92
nano	2.5.3	2.9.8
nc	1.84	
ncurses	5.7	6.0
ncurses-base	5.7	6.0
ncurses-libs	5.7	6.0
net-tools	1.60	2.0
nettle		2.7.1
newt	0.52.11	0.52.15
newt-python		0.52.15
newt-python27	0.52.11	
nfs-utils	1.3.0	1.3.0
nspr	4.25.0	4.35.0
nss	3.53.1	3.90.0
nss-pem	1.0.3	1.0.3
nss-softokn	3.53.1	3.90.0
nss-softokn-freebl	3.53.1	3.90.0

Package	AL1 AMI	AL2 AMI
nss-sysinit	3.53.1	3.90.0
nss-tools	3.53.1	3.90.0
nss-util	3.53.1	3.90.0
ntp	4.2.8p15	
ntpdate	4.2.8p15	
ntsysv	1.3.49.3	1.7.4
numactl	2.0.7	
numactl-libs		2.0.9
openldap	2.4.40	2.4.44
openssh	7.4p1	7.4p1
openssh-clients	7.4p1	7.4p1
openssh-server	7.4p1	7.4p1
openssl	1.0.2k	1.0.2k
openssl-libs		1.0.2k
os-prober		1.58
p11-kit	0.18.5	0.23.22
p11-kit-trust	0.18.5	0.23.22
pam	1.1.8	1.1.8
pam_ccreds	10	
pam_krb5	2.3.11	

Package	AL1 AMI	AL2 AMI
pam_passwdqc	1.0.5	
parted	2.1	3.1
passwd	0.79	0.79
pciutils	3.1.10	3.5.1
pciutils-libs	3.1.10	3.5.1
pcre	8.21	8.32
pcre2		10.23
perl	5.16.3	5.16.3
perl-Carp	1.26	1.26
perl-Digest	1.17	
perl-Digest-HMAC	1.03	
perl-Digest-MD5	2.52	
perl-Digest-SHA	5.85	
perl-Encode	2.51	2.51
perl-Exporter	5.68	5.68
perl-File-Path	2.09	2.09
perl-File-Temp	0.23.01	0.23.01
perl-Filter	1.49	1.49
perl-Getopt-Long	2.40	2.40
perl-HTTP-Tiny	0.033	0.033

Package	AL1 AMI	AL2 AMI
perl-PathTools	3.40	3.40
perl-Pod-Escapes	1.04	1.04
perl-Pod-Perldoc	3.20	3.20
perl-Pod-Simple	3.28	3.28
perl-Pod-Usage	1.63	1.63
perl-Scalar-List-Utils	1.27	1.27
perl-Socket	2.010	2.010
perl-Storable	2.45	2.45
perl-Text-ParseWords	3.29	3.29
perl-Time-HiRes	1.9725	1.9725
perl-Time-Local	1.2300	1.2300
perl-constant	1.27	1.27
perl-libs	5.16.3	5.16.3
perl-macros	5.16.3	5.16.3
perl-parent	0.225	0.225
perl-podlators	2.5.1	2.5.1
perl-threads	1.87	1.87
perl-threads-shared	1.43	1.43
pinentry	0.7.6	0.8.1
pkgconfig	0.27.1	0.27.1

Package	AL1 AMI	AL2 AMI
plymouth		0.8.9
plymouth-core-libs		0.8.9
plymouth-scripts		0.8.9
pm-utils	1.4.1	1.4.1
policycoreutils	2.1.12	2.5
popt	1.13	1.13
postfix		2.10.1
procmail	3.22	
procps	3.2.8	
procps-ng		3.3.10
psacct	6.3.2	6.6.1
psmisc	22.20	22.20
pth	2.0.7	2.0.7
pygpgme		0.3
pyliblzma		0.5.3
pystache		0.5.3
python		2.7.18
python-babel		0.9.6
python-backports		1.0
python-backports-ssl_match_hostname		3.5.0.1

Package	AL1 AMI	AL2 AMI
python-cffi		1.6.0
python-chardet		2.2.1
python-configobj		4.7.2
python-daemon		1.6
python-devel		2.7.18
python-docutils		0.12
python-enum34		1.0.4
python-idna		2.4
python-iniparse		0.4
python-ipaddress		1.0.16
python-jinja2		2.7.2
python-jsonpatch		1.2
python-jsonpointer		1.9
python-jwcrypto		0.4.2
python-kitchen		1.1.1
python-libs		2.7.18
python-lockfile		0.9.1
python-markupsafe		0.11
python-pillow		2.0.0
python-ply		3.4

Package	AL1 AMI	AL2 AMI
python-pycparser		2.14
python-pycurl		7.19.0
python-repoze-lru		0.4
python-requests		2.6.0
python-simplejson		3.2.0
python-urlgrabber		3.10
python-urllib3		1.25.9
python2-botocore		1.18.6
python2-colorama		0.3.9
python2-cryptography		1.7.2
python2-dateutil		2.6.1
python2-futures		3.0.5
python2-jmespath		0.9.3
python2-jsonschema		2.5.1
python2-oauthlib		2.0.1
python2-pyasn1		0.1.9
python2-rpm		4.11.3
python2-rsa		3.4.1
python2-s3transfer		0.3.3
python2-setuptools		41.2.0

Package	AL1 AMI	AL2 AMI
python2-six		1.11.0
python27	2.7.18	
python27-PyYAML	3.10	
python27-babel	0.9.4	
python27-backports	1.0	
python27-backports-ssl_match_hostname	3.4.0.2	
python27-boto	2.48.0	
python27-botocore	1.17.31	
python27-chardet	2.0.1	
python27-colorama	0.4.1	
python27-configobj	4.7.2	
python27-crypto	2.6.1	
python27-daemon	1.5.2	
python27-dateutil	2.1	
python27-devel	2.7.18	
python27-docutils	0.11	
python27-ecdsa	0.11	
python27-futures	3.0.3	
python27-imaging	1.1.6	
python27-iniparse	0.3.1	

Package	AL1 AMI	AL2 AMI
python27-jinja2	2.7.2	
python27-jmespath	0.9.2	
python27-jsonpatch	1.2	
python27-jsonpointer	1.0	
python27-kitchen	1.1.1	
python27-libs	2.7.18	
python27-lockfile	0.8	
python27-markupsafe	0.11	
python27-paramiko	1.15.1	
python27-pip	9.0.3	
python27-ply	3.4	
python27-pyasn1	0.1.7	
python27-pycurl	7.19.0	
python27-pygpgme	0.3	
python27-pyliblzma	0.5.3	
python27-pystache	0.5.3	
python27-pyxattr	0.5.0	
python27-requests	1.2.3	
python27-rsa	3.4.1	
python27-setuptools	36.2.7	

Package	AL1 AMI	AL2 AMI
python27-simplejson	3.6.5	
python27-six	1.8.0	
python27-urlgrabber	3.10	
python27-urllib3	1.24.3	
python27-virtualenv	15.1.0	
python3		3.7.16
python3-daemon		2.2.3
python3-docutils		0.14
python3-libs		3.7.16
python3-lockfile		0.11.0
python3-pip		20.2.2
python3-pystache		0.5.4
python3-setuptools		49.1.3
python3-simplejson		3.2.0
pyxattr		0.5.1
qrencode-libs		3.4.1
quota	4.00	4.01
quota-nls	4.00	4.01
rdate		1.4
readline	6.2	6.2

Package	AL1 AMI	AL2 AMI
rmt	0.4	
rng-tools	5	6.8
rootfiles	8.1	8.1
rpcbind	0.2.0	0.2.0
rpm	4.11.3	4.11.3
rpm-build-libs	4.11.3	4.11.3
rpm-libs	4.11.3	4.11.3
rpm-plugin-systemd-inhibit		4.11.3
rpm-python27	4.11.3	
rsync	3.0.6	3.1.2
rsyslog	5.8.10	8.24.0
ruby	2.0	
ruby20	2.0.0.648	
ruby20-irb	2.0.0.648	
ruby20-libs	2.0.0.648	
rubygem20-bigdecimal	1.2.0	
rubygem20-json	1.8.3	
rubygem20-psych	2.0.0	
rubygem20-rdoc	4.2.2	
rubygems20	2.0.14.1	

Package	AL1 AMI	AL2 AMI
scl-utils		20130529
screen	4.0.3	4.1.0
sed	4.2.1	4.2.2
selinux-policy		3.13.1
selinux-policy-targeted		3.13.1
sendmail	8.14.4	
setserial	2.17	2.17
setup	2.8.14	2.8.71
setuptool		1.19.11
sgpio	1.2.0.10	1.2.0.10
shadow-utils	4.1.4.2	4.1.5.1
shared-mime-info	1.1	1.8
slang	2.2.1	2.2.4
sqlite	3.7.17	3.7.17
sssd-client		1.16.5
strace		4.26
sudo	1.8.23	1.8.23
sysctl-defaults	1.0	1.0
sysfsutils	2.1.0	
sysstat		10.1.5

Package	AL1 AMI	AL2 AMI
system-release	2018.03	2
systemd		219
systemd-libs		219
systemd-sysv		219
systemtap-runtime		4.5
sysvinit	2.87	
sysvinit-tools		2.88
tar	1.26	1.26
tcp_wrappers	7.6	7.6
tcp_wrappers-libs	7.6	7.6
tcpdump		4.9.2
tcsh		6.18.01
teamd		1.27
time	1.7	1.7
tmpwatch	2.9.16	
traceroute	2.0.14	2.0.22
ttmkfdir	3.0.9	
tzdata	2023c	2023c
tzdata-java	2023c	
udev	173	

Package	AL1 AMI	AL2 AMI
unzip	6.0	6.0
update-motd	1.0.1	1.1.2
upstart	0.6.5	
usermode		1.111
ustr	1.0.4	1.0.4
util-linux	2.23.2	2.30.2
vim-common	9.0.1712	9.0.2081
vim-data	9.0.1712	9.0.2081
vim-enhanced	9.0.1712	9.0.2081
vim-filesystem	9.0.1712	9.0.2081
vim-minimal	9.0.1712	9.0.2081
virt-what		1.18
wget	1.18	1.14
which	2.19	2.20
words	3.0	3.0
xfsdump		3.1.8
xfsprogs		5.0.0
xorg-x11-font-utils	7.2	
xorg-x11-fonts-Type1	7.2	
xxd	9.0.1712	9.0.2081

Package	AL1 AMI	AL2 AMI
xz	5.2.2	5.2.2
xz-libs	5.2.2	5.2.2
yajl		2.0.4
yum	3.4.3	3.4.3
yum-langpacks		0.4.2
yum-metadata-parser	1.1.4	1.1.4
yum-plugin-priorities	1.1.31	1.1.31
yum-plugin-upgrade-helper	1.1.31	
yum-utils	1.1.31	1.1.31
zip	3.0	3.0
zlib	1.2.8	1.2.7

Comparing packages installed on AL1 and AL2 base container images

Package	AL1 Container	AL2 Container
amazon-linux-extras		2.0.3
basesystem	10.0	10.0
bash	4.2.46	4.2.46
bzip2-libs	1.0.6	1.0.6
ca-certificates	2023.2.62	2023.2.62

Package	AL1 Container	AL2 Container
chkconfig	1.3.49.3	1.7.4
coreutils	8.22	8.22
cpio		2.12
curl	7.61.1	8.3.0
cyrus-sasl-lib	2.1.23	2.1.26
db4	4.7.25	
db4-utils	4.7.25	
diffutils		3.3
elfutils-libelf	0.168	0.176
expat	2.1.0	2.1.0
file-libs	5.37	5.11
filesystem	2.4.30	3.2
findutils		4.5.11
gawk	3.1.7	4.0.2
gdbm	1.8.0	1.13
glib2	2.36.3	2.56.1
glibc	2.17	2.26
glibc-common	2.17	2.26
glibc-langpack-en		2.26
glibc-minimal-langpack		2.26

Package	AL1 Container	AL2 Container
gmp	6.0.0	6.0.0
gnupg2	2.0.28	2.0.22
gpgme	1.4.3	1.3.2
grep	2.20	2.20
gzip	1.5	
info	5.1	5.1
keyutils-libs	1.5.8	1.5.8
krb5-libs	1.15.1	1.15.1
libacl	2.2.49	2.2.51
libassuan	2.0.3	2.1.0
libattr	2.4.46	2.4.46
libblkid		2.30.2
libcap	2.16	2.54
libcom_err	1.43.5	1.42.9
libcrypt		2.26
libcurl	7.61.1	8.3.0
libdb		5.3.21
libdb-utils		5.3.21
libffi	3.0.13	3.0.13
libgcc		7.3.1

Package	AL1 Container	AL2 Container
libgcc72	7.2.1	
libgcrypt	1.5.3	1.5.3
libgpg-error	1.11	1.12
libicu	50.2	
libidn2	2.3.0	2.3.0
libmetalink		0.1.3
libmount		2.30.2
libnghttp2	1.33.0	1.41.0
libpsl	0.6.2	
libselinux	2.1.10	2.5
libsepol	2.1.7	2.5
libssh2	1.4.2	1.4.3
libstdc++		7.3.1
libstdc++72	7.2.1	
libtasn1	2.3	4.10
libunistring	0.9.3	0.9.3
libuuid		2.30.2
libverto	0.2.5	0.2.5
libxml2	2.9.1	2.9.1
libxml2-python27	2.9.1	

Package	AL1 Container	AL2 Container
lua	5.1.4	5.1.4
make	3.82	
ncurses	5.7	6.0
ncurses-base	5.7	6.0
ncurses-libs	5.7	6.0
nspr	4.25.0	4.35.0
nss	3.53.1	3.90.0
nss-pem	1.0.3	1.0.3
nss-softokn	3.53.1	3.90.0
nss-softokn-freebl	3.53.1	3.90.0
nss-sysinit	3.53.1	3.90.0
nss-tools	3.53.1	3.90.0
nss-util	3.53.1	3.90.0
openldap	2.4.40	2.4.44
openssl	1.0.2k	
openssl-libs		1.0.2k
p11-kit	0.18.5	0.23.22
p11-kit-trust	0.18.5	0.23.22
pcre	8.21	8.32
pinentry	0.7.6	0.8.1

Package	AL1 Container	AL2 Container
pkgconfig	0.27.1	
popt	1.13	1.13
pth	2.0.7	2.0.7
pygpgme		0.3
pyliblzma		0.5.3
python		2.7.18
python-iniparse		0.4
python-libs		2.7.18
python-pycurl		7.19.0
python-urlgrabber		3.10
python2-rpm		4.11.3
python27	2.7.18	
python27-chardet	2.0.1	
python27-iniparse	0.3.1	
python27-kitchen	1.1.1	
python27-libs	2.7.18	
python27-pycurl	7.19.0	
python27-pygpgme	0.3	
python27-pyliblzma	0.5.3	
python27-pyxattr	0.5.0	

Package	AL1 Container	AL2 Container
python27-urlgrabber	3.10	
pyxattr		0.5.1
readline	6.2	6.2
rpm	4.11.3	4.11.3
rpm-build-libs	4.11.3	4.11.3
rpm-libs	4.11.3	4.11.3
rpm-python27	4.11.3	
sed	4.2.1	4.2.2
setup	2.8.14	2.8.71
shared-mime-info	1.1	1.8
sqlite	3.7.17	3.7.17
sysctl-defaults	1.0	
system-release	2018.03	2
tar	1.26	
tzdata	2023c	2023c
vim-data		9.0.2081
vim-minimal		9.0.2081
xz-libs	5.2.2	5.2.2
yum	3.4.3	3.4.3
yum-metadata-parser	1.1.4	1.1.4

Package	AL1 Container	AL2 Container
yum-plugin-ovl	1.1.31	1.1.31
yum-plugin-priorities	1.1.31	1.1.31
yum-utils	1.1.31	
zlib	1.2.8	1.2.7

AL2 on Amazon EC2

Note

AL2 is no longer the current version of Amazon Linux. AL2023 is the successor to AL2. For more information, see Comparing AL2 and AL2023 and the list of Package changes in AL2023 in the AL2023 User Guide.

Topics

- Launch Amazon EC2 instance with AL2 AMI
- Find the latest AL2 AMI using Systems Manager
- Connect to an Amazon EC2 instance
- AL2 AMI boot mode
- Package repository
- Using cloud-init on AL2
- Configure AL2 instances
- User provided kernels
- AL2 AMI release notifications
- Configure the AL2 MATE desktop connection
- AL2 Tutorials

Launch Amazon EC2 instance with AL2 AMI

You can launch an Amazon EC2 instance with the AL2 AMI. For more information, see Step 1: Launch an instance.

Find the latest AL2 AMI using Systems Manager

Amazon EC2 provides Amazon Systems Manager public parameters for public AMIs maintained by Amazon that you can use when launching instances. For example, the EC2-provided parameter /

aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2 is available in all Regions and always points to the latest version of the AL2 AMI in a given Region.

To find the latest AL2023 AMI using Amazon Systems Manager, see Get started with AL2023.

The Amazon EC2 AMI public parameters are available from the following path:

/aws/service/ami-amazon-linux-latest

You can view a list of all Amazon Linux AMIs in the current Amazon Region by running the following Amazon CLI command.

```
aws ssm get-parameters-by-path --path /aws/service/ami-amazon-linux-latest --query
 "Parameters[].Name"
```

To launch an instance using a public parameter

The following example uses the EC2-provided public parameter to launch an m5.xlarge instance using the latest AL2 AMI.

To specify the parameter in the command, use the following syntax: resolve:ssm:*public-parameter*, where resolve:ssm is the standard prefix and public-parameter is the path and name of the public parameter.

In this example, the --count and --security-group parameters are not included. For --count, the default is 1. If you have a default VPC and a default security group, they are used.

```
aws ec2 run-instances
    --image-id resolve:ssm:/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-
gp2
    --instance-type m5.xlarge
    --key-name MyKeyPair
```

For more information, see Using public parameters in the *Amazon Systems Manager User Guide* and Query for the latest Amazon Linux AMI IDs Using Amazon Systems Manager Parameter Store.

Connect to an Amazon EC2 instance

There are several ways to connect to your Amazon Linux instance, including SSH, Amazon Systems Manager Session Manager, and EC2 Instance Connect. For more information, see Connect to your Linux instance in the *Amazon EC2 User Guide*.

SSH users and sudo

Amazon Linux does not allow remote `root` secure shell (SSH) by default. Also, password authentication is disabled to prevent brute force attacks. To enable SSH logins to an Amazon Linux instance, you must provide your key pair to the instance at launch. You must also set the security group used to launch your instance to allow SSH access. By default, the only account that can log in remotely using SSH is `ec2-user`. This account also has **sudo** privileges. If you enable remote `root` login, be aware that it is less secure than relying on key pairs and a secondary user.

AL2 AMI boot mode

AL2 AMIs don't have a boot mode parameter set. Instances launched from AL2 AMIs follow the default boot mode value of the instance type. For more information, see Boot modes in the *Amazon EC2 User Guide*.

Package repository

This information applies to AL2. For information about AL2023, see Manage packages and operating system updates in AL2023 in the *Amazon Linux 2023 User Guide*.

AL2 and AL1 are designed to be used with online package repositories hosted in each Amazon EC2 Amazon Region. The repositories are available in all Regions and are accessed using **yum** update tools. Hosting repositories in each Region enables us to deploy updates quickly and without any data transfer charges.

> ⚠ **Important**
>
> The last version of AL1 reached EOL on December 31, 2023 and will not receive any security updates or bug fixes starting January 1, 2024. For more information, see Amazon Linux AMI end-of-life.

If you don't need to preserve data or customizations for your instances, you can launch new instances using the current AL2 AMI. If you do need to preserve data or customizations for your instances, you can maintain those instances through the Amazon Linux package repositories. These repositories contain all the updated packages. You can choose to apply these updates to your running instances. Earlier versions of the AMI and update packages continue to be available for use, even as new versions are released.

Note

To update and install packages without internet access on an Amazon EC2 instance, see How can I update yum or install packages without internet access on my Amazon EC2 instances running AL1, AL2, or AL2023?

To install packages, use the following command:

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

If you find that Amazon Linux doesn't contain an application that you need, you can install the application directly on your Amazon Linux instance. Amazon Linux uses RPMs and yum for package management, and that is likely the most direct way to install new applications. You should check to see if an application is available in our central Amazon Linux repository first, because many applications are available there. From there, you can add these applications to your Amazon Linux instance.

To upload your applications onto a running Amazon Linux instance, use **scp** or **sftp** and then configure the application by logging in to your instance. Your applications can also be uploaded during the instance launch by using the **PACKAGE_SETUP** action from the built-in cloud-init package. For more information, see Using cloud-init on AL2.

Security updates

Security updates are provided using the package repositories. Both security updates and updated AMI security alerts are published in the Amazon Linux Security Center. For more information about Amazon security policies or to report a security problem, see Amazon Cloud Security.

AL1 and AL2 are configured to download and install critical or important security updates at launch time. Kernel updates are not included in this configuration.

In AL2023, this configuration has changed compared to AL1 and AL2. For more information about security updates for AL2023, see Security updates and features in the *Amazon Linux 2023 User Guide*.

We recommend that you make the necessary updates for your use case after launch. For example, you might want to apply all updates (not just security updates) at launch, or evaluate each update and apply only the ones applicable to your system. This is controlled using the following cloud-init setting: `repo_upgrade`. The following snippet of cloud-init configuration shows how you can change the settings in the user data text you pass to your instance initialization:

```
#cloud-config
repo_upgrade: security
```

The possible values for `repo_upgrade` are as follows:

`critical`

Apply outstanding critical security updates.

`important`

Apply outstanding critical and important security updates.

`medium`

Apply outstanding critical, important, and medium security updates.

`low`

Apply all outstanding security updates, including low-severity security updates.

`security`

Apply outstanding critical or important updates that Amazon marks as security updates.

`bugfix`

Apply updates that Amazon marks as bug fixes. Bug fixes are a larger set of updates, which include security updates and fixes for various other minor bugs.

`all`

Apply all applicable available updates, regardless of their classification.

`none`

Don't apply any updates to the instance on start up.

> **Note**
>
> Amazon Linux does not mark any updates as `bugfix`. To apply non-security related updates from Amazon Linux use `repo_upgrade: all`.

The default setting for `repo_upgrade` is security. That is, if you don't specify a different value in your user data, by default, Amazon Linux performs the security upgrades at launch for any packages installed at that time. Amazon Linux also notifies you of any updates to the installed packages by listing the number of available updates upon login using the `/etc/motd` file. To install these updates, you need to run **sudo yum upgrade** on the instance.

Repository configuration

For AL1 and AL2, AMIs are a snapshot of the packages available at the time the AMI was created, with the exception of security updates. Any packages not on the original AMI, but installed at runtime, will be the latest version available. To get the latest packages available for AL2, run **yum update -y**.

> **Troubleshooting tip**
>
> If you get a `cannot allocate memory` error running **yum update** on nano instance types, such as `t3.nano`, you might need to allocate swap space to enable the update.

For AL2023, the repository configuration has changed compared to AL1 and AL2. For more information about the AL2023 repository, see Managing packages and operating system updates.

Versions up to AL2023 were configured to deliver a continuous flow of updates to roll from one minor version of Amazon Linux to the next version, also called *rolling releases*. As a best practice, we recommend you update your AMI to the latest available AMI rather than launching old AMIs and applying updates.

In-place upgrades are not supported between major Amazon Linux versions, such as from AL1 to AL2 or from AL2 to AL2023. For more information, see Amazon Linux availability.

Using cloud-init on AL2

The cloud-init package is an open-source application built by Canonical that is used to bootstrap Linux images in a cloud computing environment, such as Amazon EC2. Amazon Linux contains a customized version of cloud-init. This allows you to specify actions that should happen to your instance at boot time. You can pass desired actions to cloud-init through the user data fields when launching an instance. This means you can use common AMIs for many use cases and configure them dynamically at startup. Amazon Linux also uses cloud-init to perform initial configuration of the ec2-user account.

For more information, see the cloud-init documentation.

Amazon Linux uses the cloud-init actions found in `/etc/cloud/cloud.cfg.d` and `/etc/cloud/cloud.cfg`. You can create your own cloud-init action files in `/etc/cloud/cloud.cfg.d`. All files in this directory are read by cloud-init. They are read in lexical order, and later files overwrite values in earlier files.

The cloud-init package performs these (and other) common configuration tasks for instances at boot:

- Set the default locale.
- Set the hostname.
- Parse and handle user data.
- Generate host private SSH keys.
- Add a user's public SSH keys to `.ssh/authorized_keys` for easy login and administration.
- Prepare the repositories for package management.
- Handle package actions defined in user data.
- Run user scripts found in user data.
- Mount instance store volumes, if applicable.
  - By default, the `ephemeral0` instance store volume is mounted at `/media/ephemeral0` if it is present and contains a valid file system; otherwise, it is not mounted.
  - By default, any swap volumes associated with the instance are mounted (only for `m1.small` and `c1.medium` instance types).
  - 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.
  - Instance store volumes that support TRIM are not formatted when an instance launches, so you must partition and format them before you can mount them. For more information, see Instance store volume TRIM support. You can use the `disk_setup` module to partition and

format your instance store volumes at boot. For more information, see Disk Setup in the cloud-init documentation.

Supported user data formats

The cloud-init package supports user data handling of a variety of formats:

- Gzip
  - If user data is gzip compressed, cloud-init decompresses the data and handles it appropriately.
- MIME multipart
  - Using a MIME multipart file, you can specify more than one type of data. For example, you could specify both a user data script and a cloud config type. Each part of the multipart file can be handled by cloud-init if it is one of the supported formats.
- Base64 decoding
  - If user data is base64-encoded, cloud-init determines if it can understand the decoded data as one of the supported types. If it understands the decoded data, it decodes the data and handles it appropriately. If not, it returns the base64 data intact.
- User data script
  - Begins with `#!` or `Content-Type: text/x-shellscript`.
  - The script is run by `/etc/init.d/cloud-init-user-scripts` during the first boot cycle. This occurs late in the boot process (after the initial configuration actions are performed).
- Include file
  - Begins with `#include` or `Content-Type: text/x-include-url`.
  - This content is an include file. The file contains a list of URLs, one per line. Each of the URLs is read, and their content passed through this same set of rules. The content read from the URL can be gzip compressed, MIME-multi-part, or plaintext.
- Cloud config data
  - Begins with `#cloud-config` or `Content-Type: text/cloud-config`.
  - This content is cloud config data.
- Upstart job (not supported on AL2)
  - Begins with `#upstart-job` or `Content-Type: text/upstart-job`.
  - This content is stored in a file in `/etc/init`, and upstart consumes the content as it does with other upstart jobs.

- Cloud boothook
  - Begins with `#cloud-boothook` or `Content-Type: text/cloud-boothook`.
  - This content is boothook data. It is stored in a file under `/var/lib/cloud` and then runs immediately.
  - This is the earliest *hook* available. There is no mechanism provided for running it only one time. The boothook must take care of this itself. It is provided with the instance ID in the environment variable `INSTANCE_ID`. Use this variable to provide a once-per-instance set of boothook data.

Configure AL2 instances

After you have successfully launched and logged into your AL2 instance, you can make changes to it. There are many different ways you can configure an instance to meet the needs of a specific application. The following are some common tasks to help get you started.

Contents

- Common configuration scenarios
- Manage software on your AL2 instance
- Processor state control for your Amazon EC2 AL2 instance
- I/O scheduler for AL2
- Change the hostname of your AL2 instance
- Set up dynamic DNS on your AL2 instance
- Configure your network interface using ec2-net-utils for AL2

Common configuration scenarios

The base distribution of Amazon Linux contains the software packages and utilities that are required for basic server operations. However, many more software packages are available in various software repositories, and even more packages are available for you to build from source code. For more information on installing and building software from these locations, see Manage software on your AL2 instance.

Amazon Linux instances come pre-configured with an `ec2-user`, but you may want to add other users that do not have super-user privileges. For more information on adding and removing users, see Manage users on your Linux instance in the *Amazon EC2 User Guide*.

If you have your own network with a domain name registered to it, you can change the hostname of an instance to identify itself as part of that domain. You can also change the system prompt to show a more meaningful name without changing the hostname settings. For more information, see Change the hostname of your AL2 instance. You can configure an instance to use a dynamic DNS service provider. For more information, see Set up dynamic DNS on your AL2 instance.

When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common configuration tasks and even run scripts after the instance starts. You can pass two types of user data to Amazon EC2: cloud-init directives and shell scripts. For more information, see Run commands on your Linux instance at launch in the *Amazon EC2 User Guide*.

Manage software on your AL2 instance

The base distribution of Amazon Linux contains the software packages and utilities that are required for basic server operations.

This information applies to AL2. For information about AL2023, see Manage packages and operating system updates in AL2023 in the *Amazon Linux 2023 User Guide*.

It is important to keep software up to date. Many packages in a Linux distribution are updated frequently to fix bugs, add features, and protect against security exploits. For more information, see Update instance software on your AL2 instance.

By default, AL2 instances launch with the following repositories enabled:

- `amzn2-core`
- `amzn2extra-docker`

While there are many packages available in these repositories that are updated by Amazon, there might be a package that you want to install that is contained in another repository. For more information, see Add repositories on an AL2 instance. For help finding and installing packages in enabled repositories, see Find and install software packages on an AL2 instance.

Not all software is available in software packages stored in repositories; some software must be compiled on an instance from its source code. For more information, see Prepare to compile software on an AL2 instance.

AL2 instances manage their software using the yum package manager. The yum package manager can install, remove, and update software, as well as manage all of the dependencies for each package.

Contents

- Update instance software on your AL2 instance
- Add repositories on an AL2 instance
- Find and install software packages on an AL2 instance
- Prepare to compile software on an AL2 instance

Update instance software on your AL2 instance

It is important to keep software up to date. Packages in a Linux distribution are updated frequently to fix bugs, add features, and protect against security exploits. When you first launch and connect to an Amazon Linux instance, you might see a message asking you to update software packages for security purposes. This section shows how to update an entire system, or just a single package.

This information applies to AL2. For information about AL2023, see Manage packages and operating system updates in AL2023 in the *Amazon Linux 2023 User Guide*.

For information about changes and updates to AL2, see AL2 release notes.

For information about changes and updates to AL2023, see AL2023 release notes.

> ⚠ **Important**
>
> If you launched an EC2 instance that uses an Amazon Linux 2 AMI into an IPv6-only subnet, you must connect to the instance and run `sudo amazon-linux-https disable`. This lets your AL2 instance connect to the yum repository in S3 over IPv6 using the http patch service.

To update all packages on an AL2 instance

1. (Optional) Start a **screen** session in your shell window. Sometimes you might experience a network interruption that can disconnect the SSH connection to your instance. If this happens during a long software update, it can leave the instance in a recoverable, although confused

state. A **screen** session allows you to continue running the update even if your connection is interrupted, and you can reconnect to the session later without problems.

a. Execute the **screen** command to begin the session.

```
[ec2-user ~]$ screen
```

b. If your session is disconnected, log back into your instance and list the available screens.

```
[ec2-user ~]$ screen -ls
There is a screen on:
 17793.pts-0.ip-12-34-56-78 (Detached)
1 Socket in /var/run/screen/S-ec2-user.
```

c. Reconnect to the screen using the **screen -r** command and the process ID from the previous command.

```
[ec2-user ~]$ screen -r 17793
```

d. When you are finished using **screen**, use the **exit** command to close the session.

```
[ec2-user ~]$ exit
[screen is terminating]
```

2. Run the **yum update** command. Optionally, you can add the `--security` flag to apply only security updates.

```
[ec2-user ~]$ sudo yum update
```

3. Review the packages listed, enter **y**, and press Enter to accept the updates. Updating all of the packages on a system can take several minutes. The **yum** output shows the status of the update while it is running.

4. (Optional) Reboot your instance to ensure that you are using the latest packages and libraries from your update; kernel updates are not loaded until a reboot occurs. Updates to any `glibc` libraries should also be followed by a reboot. For updates to packages that control services, it might be sufficient to restart the services to pick up the updates, but a system reboot ensures that all previous package and library updates are complete.

To update a single package on an AL2 instance

Use this procedure to update a single package (and its dependencies) and not the entire system.

1. Run the **yum update** command with the name of the package to update.

   ```
   [ec2-user ~]$ sudo yum update openssl
   ```

2. Review the package information listed, enter **y**, and press Enter to accept the update or updates. Sometimes there will be more than one package listed if there are package dependencies that must be resolved. The **yum** output shows the status of the update while it is running.
3. (Optional) Reboot your instance to ensure that you are using the latest packages and libraries from your update; kernel updates are not loaded until a reboot occurs. Updates to any `glibc` libraries should also be followed by a reboot. For updates to packages that control services, it might be sufficient to restart the services to pick up the updates, but a system reboot ensures that all previous package and library updates are complete.

Add repositories on an AL2 instance

This information applies to AL2. For information about AL2023, see Deterministic upgrades through versioned repositories on AL2023 in the *Amazon Linux 2023 User Guide*.

By default, AL2 instances launch with the following repositories enabled:

- `amzn2-core`
- `amzn2extra-docker`

While there are many packages available in these repositories that are updated by Amazon Web Services, there might be a package that you want to install that is contained in another repository.

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

To determine what yum repositories are already installed

List the installed yum repositories with the following command:

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

The resulting output lists the installed repositories and reports the status of each. Enabled repositories display the number of packages they contain.

To add a yum repository to /etc/yum.repos.d

1. Find the location of the `.repo` file. This will vary depending on the repository you are adding. In this example, the `.repo` file is at `https://www.`*`example`*`.com/`*`repository`*`.repo`.
2. Add the repository with the **yum-config-manager** command.

   ```
   [ec2-user ~]$ sudo yum-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 **yum-config-manager** command with the `--enable` *`repository`* flag. The following command enables the Extra Packages for Enterprise Linux (EPEL) repository from the Fedora project. By default, this repository is present in `/etc/yum.repos.d` on Amazon Linux AMI instances, but it is not enabled.

```
[ec2-user ~]$ sudo yum-config-manager --enable epel
```

For more information, and to download the latest version of this package, see https://fedoraproject.org/wiki/EPEL.

Find and install software packages on an AL2 instance

You can use a package management tool to find and install software packages. In Amazon Linux 2, the default software package management tool is YUM. In AL2023, the default software package management tool is DNF. For more information, see Package management tool in the *Amazon Linux 2023 User Guide*.

Find software packages on an AL2 instance

You can use the **yum search** command to search the descriptions of packages that are available in your configured repositories. This is especially helpful if you don't know the exact name of the package you want to install. Simply append the keyword search to the command; for multiple word searches, wrap the search query with quotation marks.

```
[ec2-user ~]$ yum search "find"
```

The following is example output.

```
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
============================== N/S matched: find ===============================
findutils.x86_64 : The GNU versions of find utilities (find and xargs)
gedit-plugin-findinfiles.x86_64 : gedit findinfiles plugin
ocaml-findlib-devel.x86_64 : Development files for ocaml-findlib
perl-File-Find-Rule.noarch : Perl module implementing an alternative interface to
 File::Find
robotfindskitten.x86_64 : A game/zen simulation. You are robot. Your job is to find
 kitten.
mlocate.x86_64 : An utility for finding files by name
ocaml-findlib.x86_64 : Objective CAML package manager and build helper
perl-Devel-Cycle.noarch : Find memory cycles in objects
perl-Devel-EnforceEncapsulation.noarch : Find access violations to blessed objects
perl-File-Find-Rule-Perl.noarch : Common rules for searching for Perl things
perl-File-HomeDir.noarch : Find your home and other directories on any platform
perl-IPC-Cmd.noarch : Finding and running system commands made easy
perl-Perl-MinimumVersion.noarch : Find a minimum required version of perl for Perl code
texlive-xesearch.noarch : A string finder for XeTeX
valgrind.x86_64 : Tool for finding memory management bugs in programs
valgrind.i686 : Tool for finding memory management bugs in programs
```

Multiple word search queries in quotation marks only return results that match the exact query. If you don't see the expected package, simplify your search to one keyword and then scan the results. You can also try keyword synonyms to broaden your search.

For more information about packages for AL2, see the following:

- AL2 Extras Library

- Package repository

Install software packages on an AL2 instance

In AL2, the yum package management tool searches all of your enabled repositories for different software packages and handles any dependencies in the software installation process. For information about installing software packages in AL2023, see Managing packages and operating system updates in the *Amazon Linux 2023 User Guide*.

To install a package from a repository

Use the **yum install *package*** command, replacing *package* with the name of the software to install. For example, to install the **links** text-based web browser, enter the following command.

```
[ec2-user ~]$ sudo yum install links
```

To install RPM package files that you have downloaded

You can also use **yum install** to install RPM package files that you have downloaded from the internet. To do this, append the path name of an RPM file to the installation command instead of a repository package name.

```
[ec2-user ~]$ sudo yum install my-package.rpm
```

To list installed packages

To view a list of installed packages on your instance, use the following command.

```
[ec2-user ~]$ yum list installed
```

Prepare to compile software on an AL2 instance

Open-source software is available on the internet that has not been pre-compiled and made available for download from a package repository. You might eventually discover a software package that you need to compile yourself, from its source code. For your system to be able to compile software in AL2 and Amazon Linux, you need to install several development tools, such as **make**, **gcc**, and **autoconf**.

Because software compilation is not a task that every Amazon EC2 instance requires, these tools are not installed by default, but they are available in a package group called "Development Tools" that is easily added to an instance with the **yum groupinstall** command.

```
[ec2-user ~]$ sudo yum groupinstall "Development Tools"
```

Software source code packages are often available for download (from websites such as https://github.com/ and http://sourceforge.net/) as a compressed archive file, called a tarball. These tarballs will usually have the `.tar.gz` file extension. You can decompress these archives with the **tar** command.

```
[ec2-user ~]$ tar -xzf software.tar.gz
```

After you have decompressed and unarchived the source code package, you should look for a `README` or `INSTALL` file in the source code directory that can provide you with further instructions for compiling and installing the source code.

To retrieve source code for Amazon Linux packages

Amazon Web Services provides the source code for maintained packages. You can download the source code for any installed packages with the **yumdownloader --source** command.

Run the **yumdownloader --source *package*** command to download the source code for *package*. For example, to download the source code for the `htop` package, enter the following command.

```
[ec2-user ~]$ yumdownloader --source htop

Loaded plugins: priorities, update-motd, upgrade-helper
Enabling amzn-updates-source repository
Enabling amzn-main-source repository
amzn-main-source
                       | 1.9 kB  00:00:00
amzn-updates-source
                       | 1.9 kB  00:00:00
(1/2): amzn-updates-source/latest/primary_db
                       |  52 kB  00:00:00
(2/2): amzn-main-source/latest/primary_db
                       | 734 kB  00:00:00
htop-1.0.1-2.3.amzn1.src.rpm
```

The location of the source RPM is in the directory from which you ran the command.

Processor state control for your Amazon EC2 AL2 instance

C-states control the sleep levels that a core can enter when it is idle. C-states are numbered starting with C0 (the shallowest state where the core is totally awake and executing instructions) and go to C6 (the deepest idle state where a core is powered off).

P-states control the desired performance (in CPU frequency) from a core. P-states are numbered starting from P0 (the highest performance setting where the core is allowed to use Intel Turbo Boost Technology to increase frequency if possible), and they go from P1 (the P-state that requests the maximum baseline frequency) to P15 (the lowest possible frequency).

You might want to change the C-state or P-state settings to increase processor performance consistency, reduce latency, or tune your instance for a specific workload. The default C-state and P-state settings provide maximum performance, which is optimal for most workloads. However, if your application would benefit from reduced latency at the cost of higher single- or dual-core frequencies, or from consistent performance at lower frequencies as opposed to bursty Turbo Boost frequencies, consider experimenting with the C-state or P-state settings that are available to these instances.

For information about Amazon EC2 instance types that provide the ability for the operating system to control processor C-states and P-states, see Processor state control for your Amazon EC2 instance in the *Amazon EC2 User Guide*.

The following sections describe the different processor state configurations and how to monitor the effects of your configuration. These procedures were written for, and apply to Amazon Linux; however, they might also work for other Linux distributions with a Linux kernel version of 3.9 or newer.

> **Note**
>
> The examples on this page use the following:
>
> - The **turbostat** utility to display processor frequency and C-state information. The **turbostat** utility is available on Amazon Linux by default.
> - The **stress** command to simulate a workload. To install **stress**, first enable the EPEL repository by running **sudo amazon-linux-extras install epel**, and then run **sudo yum install -y stress**.

If the output does not display the C-state information, include the **--debug** option in the command (**sudo turbostat --debug stress *<options>***).

Contents

- Highest performance with maximum Turbo Boost frequency
- High performance and low latency by limiting deeper C-states
- Baseline performance with the lowest variability

Highest performance with maximum Turbo Boost frequency

This is the default processor state control configuration for the Amazon Linux AMI, and it is recommended for most workloads. This configuration provides the highest performance with lower variability. Allowing inactive cores to enter deeper sleep states provides the thermal headroom required for single or dual core processes to reach their maximum Turbo Boost potential.

The following example shows a `c4.8xlarge` instance with two cores actively performing work reaching their maximum processor Turbo Boost frequency.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [30680] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [30680] successful run completed in 10s
pk cor CPU    %c0  GHz  TSC SMI    %c1    %c3    %c6    %c7   %pc2   %pc3   %pc6   %pc7
  Pkg_W RAM_W PKG_% RAM_%
             5.54 3.44 2.90   0   9.18   0.00  85.28   0.00   0.00   0.00   0.00   0.00
  94.04 32.70 54.18  0.00
 0   0   0   0.12 3.26 2.90   0   3.61   0.00  96.27   0.00   0.00   0.00   0.00   0.00
  48.12 18.88 26.02  0.00
 0   0  18   0.12 3.26 2.90   0   3.61
 0   1   1   0.12 3.26 2.90   0   4.11   0.00  95.77   0.00
 0   1  19   0.13 3.27 2.90   0   4.11
 0   2   2   0.13 3.28 2.90   0   4.45   0.00  95.42   0.00
 0   2  20   0.11 3.27 2.90   0   4.47
 0   3   3   0.05 3.42 2.90   0  99.91   0.00   0.05   0.00
 0   3  21  97.84 3.45 2.90   0   2.11
...
 1   1  10   0.06 3.33 2.90   0  99.88   0.01   0.06   0.00
 1   1  28  97.61 3.44 2.90   0   2.32
```

```
...
10.002556 sec
```

In this example, vCPUs 21 and 28 are running at their maximum Turbo Boost frequency because the other cores have entered the C6 sleep state to save power and provide both power and thermal headroom for the working cores. vCPUs 3 and 10 (each sharing a processor core with vCPUs 21 and 28) are in the C1 state, waiting for instruction.

In the following example, all 18 cores are actively performing work, so there is no headroom for maximum Turbo Boost, but they are all running at the "all core Turbo Boost" speed of 3.2 GHz.

```
[ec2-user ~]$ sudo turbostat stress -c 36 -t 10
stress: info: [30685] dispatching hogs: 36 cpu, 0 io, 0 vm, 0 hdd
stress: info: [30685] successful run completed in 10s
pk cor CPU    %c0  GHz  TSC SMI    %c1    %c3    %c6    %c7   %pc2   %pc3   %pc6   %pc7
  Pkg_W RAM_W PKG_% RAM_%
            99.27 3.20 2.90   0   0.26   0.00   0.47   0.00   0.00   0.00   0.00   0.00
 228.59 31.33 199.26  0.00
 0   0   0  99.08 3.20 2.90   0   0.27   0.01   0.64   0.00   0.00   0.00   0.00   0.00
 114.69 18.55 99.32  0.00
 0   0  18  98.74 3.20 2.90   0   0.62
 0   1   1  99.14 3.20 2.90   0   0.09   0.00   0.76   0.00
 0   1  19  98.75 3.20 2.90   0   0.49
 0   2   2  99.07 3.20 2.90   0   0.10   0.02   0.81   0.00
 0   2  20  98.73 3.20 2.90   0   0.44
 0   3   3  99.02 3.20 2.90   0   0.24   0.00   0.74   0.00
 0   3  21  99.13 3.20 2.90   0   0.13
 0   4   4  99.26 3.20 2.90   0   0.09   0.00   0.65   0.00
 0   4  22  98.68 3.20 2.90   0   0.67
 0   5   5  99.19 3.20 2.90   0   0.08   0.00   0.73   0.00
 0   5  23  98.58 3.20 2.90   0   0.69
 0   6   6  99.01 3.20 2.90   0   0.11   0.00   0.89   0.00
 0   6  24  98.72 3.20 2.90   0   0.39
...
```

High performance and low latency by limiting deeper C-states

C-states control the sleep levels that a core may enter when it is inactive. You may want to control C-states to tune your system for latency versus performance. Putting cores to sleep takes time, and although a sleeping core allows more headroom for another core to boost to a higher frequency, it takes time for that sleeping core to wake back up and perform work. For example, if a core that is assigned to handle network packet interrupts is asleep, there may be a delay in servicing that interrupt. You can configure the system to not use deeper C-states, which reduces the processor reaction latency, but that in turn also reduces the headroom available to other cores for Turbo Boost.

A common scenario for disabling deeper sleep states is a Redis database application, which stores the database in system memory for the fastest possible query response time.

To limit deeper sleep states on AL2

1. Open the `/etc/default/grub` file with your editor of choice.

   ```
   [ec2-user ~]$ sudo vim /etc/default/grub
   ```

2. Edit the `GRUB_CMDLINE_LINUX_DEFAULT` line and add the `intel_idle.max_cstate=1` and `processor.max_cstate=1` options to set C1 as the deepest C-state for idle cores.

   ```
   GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
    biosdevname=0 nvme_core.io_timeout=4294967295 intel_idle.max_cstate=1
    processor.max_cstate=1"
   GRUB_TIMEOUT=0
   ```

   The `intel_idle.max_cstate=1` option configures the C-state limit for Intel-based instances, and the `processor.max_cstate=1` option configures the C-state limit for AMD-based instances. It is safe to add both options to your configuration. This allows a single configuration to set the desired behavior on both Intel and AMD.

3. Save the file and exit your editor.

4. Run the following command to rebuild the boot configuration.

   ```
   [ec2-user ~]$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

5. Reboot your instance to enable the new kernel option.

   ```
   [ec2-user ~]$ sudo reboot
   ```

To limit deeper sleep states on Amazon Linux AMI

1. Open the `/boot/grub/grub.conf` file with your editor of choice.

```
[ec2-user ~]$ sudo vim /boot/grub/grub.conf
```

2. Edit the `kernel` line of the first entry and add the `intel_idle.max_cstate=1` and `processor.max_cstate=1` options to set C1 as the deepest C-state for idle cores.

   ```
   # created by imagebuilder
   default=0
   timeout=1
   hiddenmenu

   title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)
   root (hd0,0)
   kernel /boot/vmlinuz-3.14.26-24.46.amzn1.x86_64 root=LABEL=/ console=ttyS0
    intel_idle.max_cstate=1  processor.max_cstate=1
   initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
   ```

   The `intel_idle.max_cstate=1` option configures the C-state limit for Intel-based instances, and the `processor.max_cstate=1` option configures the C-state limit for AMD-based instances. It is safe to add both options to your configuration. This allows a single configuration to set the desired behavior on both Intel and AMD.

3. Save the file and exit your editor.

4. Reboot your instance to enable the new kernel option.

   ```
   [ec2-user ~]$ sudo reboot
   ```

The following example shows a `c4.8xlarge` instance with two cores actively performing work at the "all core Turbo Boost" core frequency.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [5322] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [5322] successful run completed in 10s
pk cor CPU    %c0  GHz  TSC SMI    %c1    %c3    %c6    %c7   %pc2   %pc3   %pc6   %pc7
  Pkg_W RAM_W PKG_% RAM_%
             5.56 3.20 2.90   0  94.44   0.00   0.00   0.00   0.00   0.00   0.00   0.00
 131.90 31.11 199.47  0.00
 0   0   0   0.03 2.08 2.90   0  99.97   0.00   0.00   0.00   0.00   0.00   0.00   0.00
  67.23 17.11 99.76  0.00
 0   0  18   0.01 1.93 2.90   0  99.99
```

```
 0   1   1   0.02 1.96 2.90    0   99.98   0.00   0.00   0.00
 0   1  19  99.70 3.20 2.90    0    0.30
...
 1   1  10   0.02 1.97 2.90    0   99.98   0.00   0.00   0.00
 1   1  28  99.67 3.20 2.90    0    0.33
 1   2  11   0.04 2.63 2.90    0   99.96   0.00   0.00   0.00
 1   2  29   0.02 2.11 2.90    0   99.98
...
```

In this example, the cores for vCPUs 19 and 28 are running at 3.2 GHz, and the other cores are in the C1 C-state, awaiting instruction. Although the working cores are not reaching their maximum Turbo Boost frequency, the inactive cores will be much faster to respond to new requests than they would be in the deeper C6 C-state.

Baseline performance with the lowest variability

You can reduce the variability of processor frequency with P-states. P-states control the desired performance (in CPU frequency) from a core. Most workloads perform better in P0, which requests Turbo Boost. But you may want to tune your system for consistent performance rather than bursty performance that can happen when Turbo Boost frequencies are enabled.

Intel Advanced Vector Extensions (AVX or AVX2) workloads can perform well at lower frequencies, and AVX instructions can use more power. Running the processor at a lower frequency, by disabling Turbo Boost, can reduce the amount of power used and keep the speed more consistent. For more information about optimizing your instance configuration and workload for AVX, see the Intel website .

CPU idle drivers control P-state. Newer CPU generations require updated CPU idle drivers that correspond to the kernel level as follows:

- Linux kernel versions 5.6 and higher (for example, m6i) – Supports Intel Icelake.
- Linux kernel versions 5.10 and higher (for example, m6a) – Supports AMD Milan.

To detect if a running system's kernel recognizes the CPU, run the following command.

```
if [ -d /sys/devices/system/cpu/cpu0/cpuidle ]; then echo "C-state control enabled";
 else echo "Kernel cpuidle driver does not recognize this CPU generation"; fi
```

If the output of this command indicates a lack of support, we recommend that you upgrade the kernel.

This section describes how to limit deeper sleep states and disable Turbo Boost (by requesting the P1 P-state) to provide low-latency and the lowest processor speed variability for these types of workloads.

To limit deeper sleep states and disable Turbo Boost on AL2

1. Open the `/etc/default/grub` file with your editor of choice.

   ```
   [ec2-user ~]$ sudo vim /etc/default/grub
   ```

2. Edit the `GRUB_CMDLINE_LINUX_DEFAULT` line and add the `intel_idle.max_cstate=1` and `processor.max_cstate=1` options to set C1 as the deepest C-state for idle cores.

   ```
   GRUB_CMDLINE_LINUX_DEFAULT="console=tty0 console=ttyS0,115200n8 net.ifnames=0
    biosdevname=0 nvme_core.io_timeout=4294967295 intel_idle.max_cstate=1
    processor.max_cstate=1"
   GRUB_TIMEOUT=0
   ```

   The `intel_idle.max_cstate=1` option configures the C-state limit for Intel-based instances, and the `processor.max_cstate=1` option configures the C-state limit for AMD-based instances. It is safe to add both options to your configuration. This allows a single configuration to set the desired behavior on both Intel and AMD.

3. Save the file and exit your editor.

4. Run the following command to rebuild the boot configuration.

   ```
   [ec2-user ~]$ grub2-mkconfig -o /boot/grub2/grub.cfg
   ```

5. Reboot your instance to enable the new kernel option.

   ```
   [ec2-user ~]$ sudo reboot
   ```

6. When you need the low processor speed variability that the P1 P-state provides, run the following command to disable Turbo Boost.

   ```
   [ec2-user ~]$ sudo sh -c "echo 1 > /sys/devices/system/cpu/intel_pstate/no_turbo"
   ```

7. When your workload is finished, you can re-enable Turbo Boost with the following command.

   ```
   [ec2-user ~]$ sudo sh -c "echo 0 > /sys/devices/system/cpu/intel_pstate/no_turbo"
   ```

To limit deeper sleep states and disable Turbo Boost on Amazon Linux AMI

1. Open the `/boot/grub/grub.conf` file with your editor of choice.

   ```
   [ec2-user ~]$ sudo vim /boot/grub/grub.conf
   ```

2. Edit the `kernel` line of the first entry and add the `intel_idle.max_cstate=1` and `processor.max_cstate=1` options to set C1 as the deepest C-state for idle cores.

   ```
   # created by imagebuilder
   default=0
   timeout=1
   hiddenmenu

   title Amazon Linux 2014.09 (3.14.26-24.46.amzn1.x86_64)
   root (hd0,0)
   kernel /boot/vmlinuz-3.14.26-24.46.amzn1.x86_64 root=LABEL=/ console=ttyS0
    intel_idle.max_cstate=1 processor.max_cstate=1
   initrd /boot/initramfs-3.14.26-24.46.amzn1.x86_64.img
   ```

   The `intel_idle.max_cstate=1` option configures the C-state limit for Intel-based instances, and the `processor.max_cstate=1` option configures the C-state limit for AMD-based instances. It is safe to add both options to your configuration. This allows a single configuration to set the desired behavior on both Intel and AMD.

3. Save the file and exit your editor.

4. Reboot your instance to enable the new kernel option.

   ```
   [ec2-user ~]$ sudo reboot
   ```

5. When you need the low processor speed variability that the P1 P-state provides, run the following command to disable Turbo Boost.

   ```
   [ec2-user ~]$ sudo sh -c "echo 1 > /sys/devices/system/cpu/intel_pstate/no_turbo"
   ```

6. When your workload is finished, you can re-enable Turbo Boost with the following command.

   ```
   [ec2-user ~]$ sudo sh -c "echo 0 > /sys/devices/system/cpu/intel_pstate/no_turbo"
   ```

The following example shows a `c4.8xlarge` instance with two vCPUs actively performing work at the baseline core frequency, with no Turbo Boost.

```
[ec2-user ~]$ sudo turbostat stress -c 2 -t 10
stress: info: [5389] dispatching hogs: 2 cpu, 0 io, 0 vm, 0 hdd
stress: info: [5389] successful run completed in 10s
pk cor CPU    %c0  GHz  TSC SMI    %c1    %c3    %c6    %c7   %pc2   %pc3   %pc6   %pc7
  Pkg_W RAM_W PKG_% RAM_%
             5.59 2.90 2.90   0  94.41   0.00   0.00   0.00   0.00   0.00   0.00   0.00
 128.48 33.54 200.00  0.00
 0   0   0   0.04 2.90 2.90   0  99.96   0.00   0.00   0.00   0.00   0.00   0.00   0.00
  65.33 19.02 100.00  0.00
 0   0  18   0.04 2.90 2.90   0  99.96
 0   1   1   0.05 2.90 2.90   0  99.95   0.00   0.00   0.00
 0   1  19   0.04 2.90 2.90   0  99.96
 0   2   2   0.04 2.90 2.90   0  99.96   0.00   0.00   0.00
 0   2  20   0.04 2.90 2.90   0  99.96
 0   3   3   0.05 2.90 2.90   0  99.95   0.00   0.00   0.00
 0   3  21  99.95 2.90 2.90   0   0.05
...
 1   1  28  99.92 2.90 2.90   0   0.08
 1   2  11   0.06 2.90 2.90   0  99.94   0.00   0.00   0.00
 1   2  29   0.05 2.90 2.90   0  99.95
```

The cores for vCPUs 21 and 28 are actively performing work at the baseline processor speed of 2.9 GHz, and all inactive cores are also running at the baseline speed in the C1 C-state, ready to accept instructions.

I/O scheduler for AL2

The I/O scheduler is a part of the Linux operating system that sorts and merges I/O requests and determines the order in which they are processed.

I/O schedulers are particularly beneficial for devices such as magnetic hard drives, where seek time can be expensive and where it is optimal to merge co-located requests. I/O schedulers have less of an effect with solid state devices and virtualized environments. This is because for solid state devices, sequential and random access don't differ, and for virtualized environments, the host provides its own layer of scheduling.

This topic discusses the Amazon Linux I/O scheduler. For more information about the I/O scheduler used by other Linux distributions, refer to their respective documentation.

Topics

- Supported schedulers
- Default scheduler
- Change the scheduler

Supported schedulers

Amazon Linux supports the following I/O schedulers:

- `deadline` — The *Deadline* I/O scheduler sorts I/O requests and handles them in the most efficient order. It guarantees a start time for each I/O request. It also gives I/O requests that have been pending for too long a higher priority.
- `cfq` — The *Completely Fair Queueing* (CFQ) I/O scheduler attempts to fairly allocate I/O resources between processes. It sorts and inserts I/O requests into per-process queues.
- `noop` — The *No Operation* (noop) I/O scheduler inserts all I/O requests into a FIFO queue and then merges them into a single request. This scheduler does not do any request sorting.

Default scheduler

No Operation (noop) is the default I/O scheduler for Amazon Linux. This scheduler is used for the following reasons:

- Many instance types use virtualized devices where the underlying host performs scheduling for the instance.
- Solid state devices are used in many instance types where the benefits of an I/O scheduler have less effect.
- It is the least invasive I/O scheduler, and it can be customized if needed.

Change the scheduler

Changing the I/O scheduler can increase or decrease performance based on whether the scheduler results in more or fewer I/O requests being completed in a given time. This is largely dependent on your workload, the generation of the instance type that's being used, and the type of device being accessed. If you change the I/O scheduler being used, we recommend that you use a tool, such as **iotop**, to measure I/O performance and to determine whether the change is beneficial for your use case.

You can view the I/O scheduler for a device using the following command, which uses `nvme0n1` as an example. Replace `nvme0n1` in the following command with the device listed in `/sys/block` on your instance.

```
$  cat /sys/block/nvme0n1/queue/scheduler
```

To set the I/O scheduler for the device, use the following command.

```
$  echo cfq|deadline|noop > /sys/block/nvme0n1/queue/scheduler
```

For example, to set the I/O scheduler for an *`xvda`* device from noop to `cfq`, use the following command.

```
$  echo cfq > /sys/block/xvda/queue/scheduler
```

Change the hostname of your AL2 instance

When you launch an instance into a private VPC, Amazon EC2 assigns a guest OS hostname. The type of hostname that Amazon EC2 assigns depends on your subnet settings. For more information about EC2 hostnames, see Amazon EC2 instance hostname types in the *Amazon EC2 User Guide*.

A typical Amazon EC2 private DNS name for an EC2 instance configured to use IP-based naming with an IPv4 address looks something like this: `ip-12-34-56-78.us-west-2.compute.internal`, where the name consists of the internal domain, the service (in this case, `compute`), the region, and a form of the private IPv4 address. Part of this hostname is displayed at the shell prompt when you log into your instance (for example, `ip-12-34-56-78`). Each time you stop and restart your Amazon EC2 instance (unless you are using an Elastic IP address), the public IPv4 address changes, and so does your public DNS name, system hostname, and shell prompt.

> ⚠ **Important**
>
> This information applies to Amazon Linux. For information about other distributions, see their specific documentation.

Change the system hostname

If you have a public DNS name registered for the IP address of your instance (such as webserver.mydomain.com), you can set the system hostname so your instance identifies itself as a part of that domain. This also changes the shell prompt so that it displays the first portion of this name instead of the hostname supplied by Amazon (for example, ip-12-34-56-78). If you do not have a public DNS name registered, you can still change the hostname, but the process is a little different.

In order for your hostname update to persist, you must verify that the preserve_hostname cloud-init setting is set to true. You can run the following command to edit or add this setting:

```
sudo vi /etc/cloud/cloud.cfg
```

If the preserve_hostname setting is not listed, add the following line of text to the end of the file:

```
preserve_hostname: true
```

To change the system hostname to a public DNS name

Follow this procedure if you already have a public DNS name registered.

1. - For AL2: Use the **hostnamectl** command to set your hostname to reflect the fully qualified domain name (such as **webserver.mydomain.com**).

     ```
     [ec2-user ~]$ sudo hostnamectl set-hostname webserver.mydomain.com
     ```

   - For Amazon Linux AMI: On your instance, open the /etc/sysconfig/network configuration file in your favorite text editor and change the HOSTNAME entry to reflect the fully qualified domain name (such as **webserver.mydomain.com**).

     ```
     HOSTNAME=webserver.mydomain.com
     ```

2. Reboot the instance to pick up the new hostname.

   ```
   [ec2-user ~]$ sudo reboot
   ```

Alternatively, you can reboot using the Amazon EC2 console (on the **Instances** page, select the instance and choose **Instance state**, **Reboot instance**).

3. Log into your instance and verify that the hostname has been updated. Your prompt should show the new hostname (up to the first ".") and the **hostname** command should show the fully-qualified domain name.

```
[ec2-user@webserver ~]$ hostname
webserver.mydomain.com
```

To change the system hostname without a public DNS name

1. - For AL2: Use the **hostnamectl** command to set your hostname to reflect the desired system hostname (such as **`webserver`**).

   ```
   [ec2-user ~]$ sudo hostnamectl set-hostname webserver.localdomain
   ```

   - For Amazon Linux AMI: On your instance, open the `/etc/sysconfig/network` configuration file in your favorite text editor and change the `HOSTNAME` entry to reflect the desired system hostname (such as **`webserver`**).

   ```
   HOSTNAME=webserver.localdomain
   ```

2. Open the `/etc/hosts` file in your favorite text editor and change the entry beginning with **`127.0.0.1`** to match the example below, substituting your own hostname.

```
127.0.0.1 webserver.localdomain webserver localhost4 localhost4.localdomain4
```

3. Reboot the instance to pick up the new hostname.

```
[ec2-user ~]$ sudo reboot
```

Alternatively, you can reboot using the Amazon EC2 console (on the **Instances** page, select the instance and choose **Instance state**, **Reboot instance**).

4. Log into your instance and verify that the hostname has been updated. Your prompt should show the new hostname (up to the first ".") and the **hostname** command should show the fully-qualified domain name.

```
[ec2-user@webserver ~]$ hostname
webserver.localdomain
```

You can also implement more programmatic solutions, such as specifying user data to configure your instance. If your instance is part of an Auto Scaling group, you can use lifecycle hooks to define user data. For more information, see Run commands on your Linux instance at launch and Lifecycle hook for instance launch in the *Amazon CloudFormation User Guide*.

Change the shell prompt without affecting the hostname

If you do not want to modify the hostname for your instance, but you would like to have a more useful system name (such as **webserver**) displayed than the private name supplied by Amazon (for example, `ip-12-34-56-78`), you can edit the shell prompt configuration files to display your system nickname instead of the hostname.

To change the shell prompt to a host nickname

1. Create a file in `/etc/profile.d` that sets the environment variable called `NICKNAME` to the value you want in the shell prompt. For example, to set the system nickname to **webserver**, run the following command.

```
[ec2-user ~]$ sudo sh -c 'echo "export NICKNAME=webserver" > /etc/profile.d/
prompt.sh'
```

2. Open the `/etc/bashrc` (Red Hat) or `/etc/bash.bashrc` (Debian/Ubuntu) file in your favorite text editor (such as **vim** or **nano**). You need to use **sudo** with the editor command because `/etc/bashrc` and `/etc/bash.bashrc` are owned by `root`.

3. Edit the file and change the shell prompt variable (PS1) to display your nickname instead of the hostname. Find the following line that sets the shell prompt in `/etc/bashrc` or `/etc/bash.bashrc` (several surrounding lines are shown below for context; look for the line that starts with `[ "$PS1"`):

```
  # Turn on checkwinsize
  shopt -s checkwinsize
  [ "$PS1" = "\\s-\\v\\\$ " ] && PS1="[\u@\h \W]\\$ "
  # You might want to have e.g. tty in prompt (e.g. more virtual machines)
  # and console windows
```

Change the \h (the symbol for hostname) in that line to the value of the NICKNAME variable.

```
# Turn on checkwinsize
shopt -s checkwinsize
[ "$PS1" = "\\s-\\v\\\$ " ] && PS1="[\u@$NICKNAME \W]\\$ "
# You might want to have e.g. tty in prompt (e.g. more virtual machines)
# and console windows
```

4. (Optional) To set the title on shell windows to the new nickname, complete the following steps.

   a. Create a file named /etc/sysconfig/bash-prompt-xterm.

   ```
   [ec2-user ~]$ sudo touch /etc/sysconfig/bash-prompt-xterm
   ```

   b. Make the file executable using the following command.

   ```
   [ec2-user ~]$ sudo chmod +x /etc/sysconfig/bash-prompt-xterm
   ```

   c. Open the /etc/sysconfig/bash-prompt-xterm file in your favorite text editor (such as **vim** or **nano**). You need to use **sudo** with the editor command because /etc/sysconfig/bash-prompt-xterm is owned by root.

   d. Add the following line to the file.

   ```
   echo -ne "\033]0;${USER}@${NICKNAME}:${PWD/#$HOME/~}\007"
   ```

5. Log out and then log back in to pick up the new nickname value.

Change the hostname on other Linux distributions

The procedures on this page are intended for use with Amazon Linux only. For more information about other Linux distributions, see their specific documentation and the following articles:

- How do I assign a static hostname to a private Amazon EC2 instance running RHEL 7 or Centos 7?

Set up dynamic DNS on your AL2 instance

When you launch an EC2 instance, it is assigned a public IP address and a public Domain Name System (DNS) name that you can use to reach it from the internet. Because there are so many hosts in the Amazon Web Services domain, these public names must be quite long for each name to remain unique. A typical Amazon EC2 public DNS name looks something like this: `ec2-12-34-56-78.us-west-2.compute.amazonaws.com`, where the name consists of the Amazon Web Services domain, the service (in this case, `compute`), the Amazon Web Services Region, and a form of the public IP address.

Dynamic DNS services provide custom DNS host names within their domain area that can be easy to remember and that can also be more relevant to your host's use case. Some of these services are also free of charge. You can use a dynamic DNS provider with Amazon EC2 and configure the instance to update the IP address associated with a public DNS name each time the instance starts. There are many different providers to choose from, and the specific details of choosing a provider and registering a name with them are outside the scope of this guide.

To use dynamic DNS with Amazon EC2

1. Sign up with a dynamic DNS service provider and register a public DNS name with their service. This procedure uses the free service from noip.com/free as an example.
2. Configure the dynamic DNS update client. After you have a dynamic DNS service provider and a public DNS name registered with their service, point the DNS name to the IP address for your instance. Many providers (including noip.com) allow you to do this manually from your account page on their website, but many also support software update clients. If an update client is running on your EC2 instance, your dynamic DNS record is updated each time the IP address changes, as happens after a shutdown and restart. In this example, you install the noip2 client, which works with the service provided by noip.com.

    a. Enable the Extra Packages for Enterprise Linux (EPEL) repository to gain access to the `noip2` client.

    > **Note**
    >
    > AL2 instances have the GPG keys and repository information for the EPEL repository installed by default. For more information, and to download the latest version of this package, see https://fedoraproject.org/wiki/EPEL.

    ```
    [ec2-user ~]$ sudo amazon-linux-extras install epel -y
    ```

    b. Install the `noip` package.

```
[ec2-user ~]$ sudo yum install -y noip
```

c. Create the configuration file. Enter the login and password information when prompted and answer the subsequent questions to configure the client.

```
[ec2-user ~]$ sudo noip2 -C
```

3. Enable the noip service.

```
[ec2-user ~]$ sudo systemctl enable noip.service
```

4. Start the noip service.

```
[ec2-user ~]$ sudo systemctl start noip.service
```

   This command starts the client, which reads the configuration file (`/etc/no-ip2.conf`) that you created earlier and updates the IP address for the public DNS name that you chose.

5. Verify that the update client has set the correct IP address for your dynamic DNS name. Allow a few minutes for the DNS records to update, and then try to connect to your instance using SSH with the public DNS name that you configured in this procedure.

Configure your network interface using ec2-net-utils for AL2

Amazon Linux 2 AMIs may contain additional scripts installed by Amazon, known as ec2-net-utils. These scripts optionally automate the configuration of your network interfaces. These scripts are available for AL2 only.

> **Note**
>
> For Amazon Linux 2023, the `amazon-ec2-net-utils` package generates interface-specific configurations in the `/run/systemd/network` directory. For more information, see Networking service in the *Amazon Linux 2023 User Guide*.

Use the following command to install the package on AL2 if it's not already installed, or update it if it's installed and additional updates are available:

```
$ yum install ec2-net-utils
```

The following components are part of ec2-net-utils:

udev rules (`/etc/udev/rules.d`)

Identifies network interfaces when they are attached, detached, or reattached to a running instance, and ensures that the hotplug script runs (`53-ec2-network-interfaces.rules`). Maps the MAC address to a device name (`75-persistent-net-generator.rules`, which generates `70-persistent-net.rules`).

hotplug script

Generates an interface configuration file suitable for use with DHCP (`/etc/sysconfig/network-scripts/ifcfg-eth`*N*). Also generates a route configuration file (`/etc/sysconfig/network-scripts/route-eth`*N*).

DHCP script

Whenever the network interface receives a new DHCP lease, this script queries the instance metadata for Elastic IP addresses. For each Elastic IP address, it adds a rule to the routing policy database to ensure that outbound traffic from that address uses the correct network interface. It also adds each private IP address to the network interface as a secondary address.

ec2ifup eth*N* (`/usr/sbin/`)

Extends the functionality of the standard **ifup**. After this script rewrites the configuration files `ifcfg-eth`*N* and `route-eth`*N*, it runs **ifup**.

ec2ifdown eth*N* (`/usr/sbin/`)

Extends the functionality of the standard **ifdown**. After this script removes any rules for the network interface from the routing policy database, it runs **ifdown**.

ec2ifscan (`/usr/sbin/`)

Checks for network interfaces that have not been configured and configures them.

This script isn't available in the initial release of ec2-net-utils.

To list any configuration files that were generated by ec2-net-utils, use the following command:

```
$ ls -l /etc/sysconfig/network-scripts/*-eth?
```

To disable the automation, you can add `EC2SYNC=no` to the corresponding `ifcfg-ethN` file. For example, use the following command to disable the automation for the eth1 interface:

```
$ sed -i -e 's/^EC2SYNC=yes/EC2SYNC=no/' /etc/sysconfig/network-scripts/ifcfg-eth1
```

To disable the automation completely, you can remove the package using the following command:

```
$ yum remove ec2-net-utils
```

User provided kernels

If you need a custom kernel on your Amazon EC2 instances, you can start with an AMI that is close to what you want, compile the custom kernel on your instance, and update the bootloader to point to the new kernel. This process varies depending on the virtualization type that your AMI uses. For more information, see Linux AMI virtualization types in the *Amazon EC2 User Guide*.

Contents

- HVM AMIs (GRUB)
- Paravirtual AMIs (PV-GRUB)

HVM AMIs (GRUB)

HVM instance volumes are treated like actual physical disks. The boot process is similar to that of a bare metal operating system with a partitioned disk and bootloader, which enables it to work with all currently supported Linux distributions. The most common bootloader is GRUB or GRUB2.

By default, GRUB does not send its output to the instance console because it creates an extra boot delay. For more information, see Instance console output in the *Amazon EC2 User Guide*. If you are installing a custom kernel, you should consider enabling GRUB output.

You don't need to specify a fallback kernel, but we recommend that you have a fallback when you test a new kernel. GRUB can fall back to another kernel in the event that the new kernel fails. Having a fallback kernel enables the instance to boot even if the new kernel isn't found.

The legacy GRUB for Amazon Linux uses `/boot/grub/menu.lst`. GRUB2 for AL2 uses `/etc/default/grub`. For more information about updating the default kernel in the bootloader, see the documentation for your Linux distribution.

Paravirtual AMIs (PV-GRUB)

AMIs that use paravirtual (PV) virtualization use a system called *PV-GRUB* during the boot process. PV-GRUB is a paravirtual bootloader that runs a patched version of GNU GRUB 0.97. When you start an instance, PV-GRUB starts the boot process and then chain loads the kernel specified by your image's `menu.lst` file.

PV-GRUB understands standard `grub.conf` or `menu.lst` commands, which allows it to work with all currently supported Linux distributions. Older distributions such as Ubuntu 10.04 LTS, Oracle Enterprise Linux, or CentOS 5.x require a special "ec2" or "xen" kernel package, while newer distributions include the required drivers in the default kernel package.

Most modern paravirtual AMIs use a PV-GRUB AKI by default (including all of the paravirtual Linux AMIs available in the Amazon EC2 Launch Wizard Quick Start menu), so there are no additional steps that you need to take to use a different kernel on your instance, provided that the kernel you want to use is compatible with your distribution. The best way to run a custom kernel on your instance is to start with an AMI that is close to what you want and then to compile the custom kernel on your instance and modify the `menu.lst` file to boot with that kernel.

You can verify that the kernel image for an AMI is a PV-GRUB AKI. Run the following describe-images command (substituting your kernel image ID) and check whether the `Name` field starts with `pv-grub`:

```
aws ec2 describe-images --filters Name=image-id,Values=aki-880531cd
```

Contents

- Limitations of PV-GRUB
- Configure GRUB for paravirtual AMIs
- Amazon PV-GRUB Kernel Image IDs
- Update PV-GRUB

Limitations of PV-GRUB

PV-GRUB has the following limitations:

- You can't use the 64-bit version of PV-GRUB to start a 32-bit kernel or vice versa.
- You can't specify an Amazon ramdisk image (ARI) when using a PV-GRUB AKI.

- Amazon has tested and verified that PV-GRUB works with these file system formats: EXT2, EXT3, EXT4, JFS, XFS, and ReiserFS. Other file system formats might not work.
- PV-GRUB can boot kernels compressed using the gzip, bzip2, lzo, and xz compression formats.
- Cluster AMIs don't support or need PV-GRUB, because they use full hardware virtualization (HVM). While paravirtual instances use PV-GRUB to boot, HVM instance volumes are treated like actual disks, and the boot process is similar to the boot process of a bare metal operating system with a partitioned disk and bootloader.
- PV-GRUB versions 1.03 and earlier don't support GPT partitioning; they support MBR partitioning only.
- If you plan to use a logical volume manager (LVM) with Amazon Elastic Block Store (Amazon EBS) volumes, you need a separate boot partition outside of the LVM. Then you can create logical volumes with the LVM.

Configure GRUB for paravirtual AMIs

To boot PV-GRUB, a GRUB `menu.lst` file must exist in the image; the most common location for this file is `/boot/grub/menu.lst`.

The following is an example of a `menu.lst` configuration file for booting an AMI with a PV-GRUB AKI. In this example, there are two kernel entries to choose from: Amazon Linux 2018.03 (the original kernel for this AMI), and Vanilla Linux 4.16.4 (a newer version of the Vanilla Linux kernel from https://www.kernel.org/). The Vanilla entry was copied from the original entry for this AMI, and the `kernel` and `initrd` paths were updated to the new locations. The `default 0` parameter points the bootloader to the first entry it sees (in this case, the Vanilla entry), and the `fallback 1` parameter points the bootloader to the next entry if there is a problem booting the first.

```
default 0
fallback 1
timeout 0
hiddenmenu

title Vanilla Linux 4.16.4
root (hd0)
kernel /boot/vmlinuz-4.16.4 root=LABEL=/ console=hvc0
initrd /boot/initrd.img-4.16.4

title Amazon Linux 2018.03 (4.14.26-46.32.amzn1.x86_64)
root (hd0)
```

```
kernel /boot/vmlinuz-4.14.26-46.32.amzn1.x86_64 root=LABEL=/ console=hvc0
initrd /boot/initramfs-4.14.26-46.32.amzn1.x86_64.img
```

You don't need to specify a fallback kernel in your `menu.lst` file, but we recommend that you have a fallback when you test a new kernel. PV-GRUB can fall back to another kernel in the event that the new kernel fails. Having a fallback kernel allows the instance to boot even if the new kernel isn't found.

PV-GRUB checks the following locations for `menu.lst`, using the first one it finds:

- `(hd0)/boot/grub`
- `(hd0,0)/boot/grub`
- `(hd0,0)/grub`
- `(hd0,1)/boot/grub`
- `(hd0,1)/grub`
- `(hd0,2)/boot/grub`
- `(hd0,2)/grub`
- `(hd0,3)/boot/grub`
- `(hd0,3)/grub`

Note that PV-GRUB 1.03 and earlier only check one of the first two locations in this list.

Amazon PV-GRUB Kernel Image IDs

PV-GRUB AKIs are available in all Amazon EC2 regions, excluding Asia Pacific (Osaka). There are AKIs for both 32-bit and 64-bit architecture types. Most modern AMIs use a PV-GRUB AKI by default.

We recommend that you always use the latest version of the PV-GRUB AKI, as not all versions of the PV-GRUB AKI are compatible with all instance types. Use the following describe-images command to get a list of the PV-GRUB AKIs for the current region:

```
aws ec2 describe-images --owners amazon --filters Name=name,Values=pv-grub-*.gz
```

PV-GRUB is the only AKI available in the `ap-southeast-2` Region. You should verify that any AMI you want to copy to this Region is using a version of PV-GRUB that is available in this Region.

The following are the current AKI IDs for each Region. Register new AMIs using an hd0 AKI.

Note

We continue to provide hd00 AKIs for backward compatibility in Regions where they were previously available.

ap-northeast-1, Asia Pacific (Tokyo)

Image ID	Image Name
aki-f975a998	pv-grub-hd0_1.05-i386.gz
aki-7077ab11	pv-grub-hd0_1.05-x86_64.gz

ap-southeast-1, Asia Pacific (Singapore) Region

Image ID	Image Name
aki-17a40074	pv-grub-hd0_1.05-i386.gz
aki-73a50110	pv-grub-hd0_1.05-x86_64.gz

ap-southeast-2, Asia Pacific (Sydney)

Image ID	Image Name
aki-ba5665d9	pv-grub-hd0_1.05-i386.gz
aki-66506305	pv-grub-hd0_1.05-x86_64.gz

eu-central-1, Europe (Frankfurt)

Image ID	Image Name
aki-1419e57b	pv-grub-hd0_1.05-i386.gz
aki-931fe3fc	pv-grub-hd0_1.05-x86_64.gz

eu-west-1, Europe (Ireland)

Image ID	Image Name
aki-1c9fd86f	pv-grub-hd0_1.05-i386.gz
aki-dc9ed9af	pv-grub-hd0_1.05-x86_64.gz

sa-east-1, South America (São Paulo)

Image ID	Image Name
aki-7cd34110	pv-grub-hd0_1.05-i386.gz
aki-912fbcfd	pv-grub-hd0_1.05-x86_64.gz

us-east-1, US East (N. Virginia)

Image ID	Image Name
aki-04206613	pv-grub-hd0_1.05-i386.gz
aki-5c21674b	pv-grub-hd0_1.05-x86_64.gz

us-gov-west-1, Amazon GovCloud (US-West)

Image ID	Image Name
aki-5ee9573f	pv-grub-hd0_1.05-i386.gz
aki-9ee55bff	pv-grub-hd0_1.05-x86_64.gz

us-west-1, US West (N. California)

Image ID	Image Name
aki-43cf8123	pv-grub-hd0_1.05-i386.gz

Image ID	Image Name
aki-59cc8239	pv-grub-hd0_1.05-x86_64.gz

us-west-2, US West (Oregon)

Image ID	Image Name
aki-7a69931a	pv-grub-hd0_1.05-i386.gz
aki-70cb0e10	pv-grub-hd0_1.05-x86_64.gz

Update PV-GRUB

We recommend that you always use the latest version of the PV-GRUB AKI, as not all versions of the PV-GRUB AKI are compatible with all instance types. Also, older versions of PV-GRUB are not available in all regions, so if you copy an AMI that uses an older version to a Region that does not support that version, you will be unable to boot instances launched from that AMI until you update the kernel image. Use the following procedures to check your instance's version of PV-GRUB and update it if necessary.

To check your PV-GRUB version

1. Find the kernel ID for your instance.

   ```
   aws ec2 describe-instance-attribute --instance-id instance_id --attribute kernel --
   region region

   {
       "InstanceId": "instance_id",
       "KernelId": "aki-70cb0e10"
   }
   ```

   The kernel ID for this instance is `aki-70cb0e10`.

2. View the version information of that kernel ID.

   ```
   aws ec2 describe-images --image-ids aki-70cb0e10 --region region
   ```

```
{
    "Images": [
        {
            "VirtualizationType": "paravirtual",
            "Name": "pv-grub-hd0_1.05-x86_64.gz",
            ...
            "Description": "PV-GRUB release 1.05, 64-bit"
        }
    ]
}
```

This kernel image is PV-GRUB 1.05. If your PV-GRUB version is not the newest version (as shown in Amazon PV-GRUB Kernel Image IDs), you should update it using the following procedure.

To update your PV-GRUB version

If your instance is using an older version of PV-GRUB, you should update it to the latest version.

1. Identify the latest PV-GRUB AKI for your Region and processor architecture from Amazon PV-GRUB Kernel Image IDs.
2. Stop your instance. Your instance must be stopped to modify the kernel image used.

   ```
   aws ec2 stop-instances --instance-ids instance_id --region region
   ```

3. Modify the kernel image used for your instance.

   ```
   aws ec2 modify-instance-attribute --instance-id instance_id --kernel kernel_id --
   region region
   ```

4. Restart your instance.

   ```
   aws ec2 start-instances --instance-ids instance_id --region region
   ```

AL2 AMI release notifications

To be notified when new Amazon Linux AMIs are released, you can subscribe using Amazon SNS.

For information about subscribing to notifications for AL2023, see Receiving notifications on new updates in the *Amazon Linux 2023 User Guide*.

Note

Standard support for AL1 ended on December 31, 2020. The AL1 maintenance support phase ended December 31, 2023. For more information about the AL1 EOL and maintenance support, see the blog post Update on Amazon Linux AMI end-of-life.

To subscribe to Amazon Linux notifications

1. Open the Amazon SNS console at https://console.amazonaws.cn/sns/v3/home.
2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must select the Region in which the SNS notification that you are subscribing to was created.
3. In the navigation pane, choose **Subscriptions**, **Create subscription**.
4. For the **Create subscription** dialog box, do the following:
   a. [AL2] For **Topic ARN**, copy and paste the following Amazon Resource Name (ARN): **`arn:aws:sns:us-east-1:137112412989:amazon-linux-2-ami-updates`**.
   b. [Amazon Linux] For **Topic ARN**, copy and paste the following Amazon Resource Name (ARN): **`arn:aws:sns:us-east-1:137112412989:amazon-linux-ami-updates`**.
   c. For **Protocol**, choose **Email**.
   d. For **Endpoint**, enter an email address that you can use to receive the notifications.
   e. 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.

Whenever AMIs are released, we send notifications to the subscribers of the corresponding topic. To stop receiving these notifications, use the following procedure to unsubscribe.

To unsubscribe from Amazon Linux notifications

1. Open the Amazon SNS console at https://console.amazonaws.cn/sns/v3/home.

2. In the navigation bar, change the Region to **US East (N. Virginia)**, if necessary. You must use the Region in which the SNS notification was created.
3. In the navigation pane, choose **Subscriptions**, select the subscription, and choose **Actions**, **Delete subscriptions**.
4. When prompted for confirmation, choose **Delete**.

Amazon Linux AMI SNS message format

The schema for the SNS message is as follows.

```
{
    "description": "Validates output from AMI Release SNS message",
    "type": "object",
    "properties": {
        "v1": {
            "type": "object",
            "properties": {
                "ReleaseVersion": {
                    "description": "Major release (ex. 2018.03)",
                    "type": "string"
                },
                "ImageVersion": {
                    "description": "Full release (ex. 2018.03.0.20180412)",
                    "type": "string"
                },
                "ReleaseNotes": {
                    "description": "Human-readable string with extra information",
                    "type": "string"
                },
                "Regions": {
                    "type": "object",
                    "description": "Each key will be a region name (ex. us-east-1)",
                    "additionalProperties": {
                        "type": "array",
                        "items": {
                            "type": "object",
                            "properties": {
                                "Name": {
                                    "description": "AMI Name (ex. amzn-ami-
hvm-2018.03.0.20180412-x86_64-gp2)",
                                    "type": "string"
                                },
                                "ImageId": {
                                    "description": "AMI Name (ex.ami-467ca739)",
                                    "type": "string"
                                }
                            },
                            "required": [
                                "Name",
                                "ImageId"
                            ]
                        }
                    }
                }
            },
            "required": [
                "ReleaseVersion",
                "ImageVersion",
                "ReleaseNotes",
                "Regions"
            ]
        }
    },
    "required": [
        "v1"
    ]
}
```

Configure the AL2 MATE desktop connection

The MATE desktop environment is pre-installed and pre-configured in AMIs with the following description:

".NET Core *x.x*, Mono *x.xx*, PowerShell *x.x*, and MATE DE pre-installed to run your .NET applications on Amazon Linux 2 with Long Term Support (LTS)."

The environment provides an intuitive graphical user interface for administering AL2 instances with minimal use of the command line. The interface uses graphical representations, such as icons, windows, toolbars, folders, wallpapers, and desktop widgets. Built-in, GUI-based tools are available to perform common tasks. For example, there are tools for adding and removing software, applying updates, organizing files, launching programs, and monitoring system health.

Important

xrdp is the remote desktop software bundled in the AMI. By default, xrdp uses a self-signed TLS certificate to encrypt remote desktop sessions. Neither Amazon nor the xrdp maintainers recommend using self-signed certificates in production. Instead, obtain a certificate from an appropriate certificate authority (CA) and install it on your instances. For more information about TLS configuration, see TLS security layer on the xrdp wiki.

Note

If you prefer to use a virtual network computing (VNC) service instead of xrdp, see the How do I install a GUI on my Amazon EC2 instance running AL2 Amazon Knowledge Center article.

Prerequisite

To run the commands shown in this topic, you must install the Amazon Command Line Interface (Amazon CLI) or Amazon Tools for Windows PowerShell, and configure your Amazon profile.

Options

1. Install the Amazon CLI – For more information, see Installing the Amazon CLI and Configuration basics in the *Amazon Command Line Interface User Guide*.
2. Install the Tools for Windows PowerShell – For more information, see Installing the Amazon Tools for Windows PowerShell and Shared credentials in the *Amazon Tools for PowerShell User Guide*.

Tip

As an alternative to doing a full installation of the Amazon CLI, you can use Amazon CloudShell for a browser-based, pre-authenticated shell that launches directly from the Amazon Web Services Management Console. Check supported Amazon Web Services Regions, to make sure it's available in the region you are working in.

Configure the RDP connection

Follow these steps to set up a Remote Desktop Protocol (RDP) connection from your local machine to an AL2 instance running the MATE desktop environment.

1. To get the ID of the AMI for AL2 that includes MATE in the AMI name, you can use the describe-images command from your local command line tool. If you have not installed the command line tools, you can perform the following query directly from an Amazon CloudShell session. For information about how to launch a shell session from CloudShell, see Getting started with Amazon CloudShell. From the Amazon EC2 console, you can find the MATE-included AMI by launching an instance, and then entering `MATE` in the AMI search bar. The AL2 Quick Start with MATE pre-installed will appear in the search results.

```
aws ec2 describe-images --filters "Name=name,Values=amzn2*MATE*" --query
 "Images[*].[ImageId,Name,Description]"
[
    [
        "ami-0123example0abc12",
        "amzn2-x86_64-MATEDE_DOTNET-2020.12.04",
        ".NET Core 5.0, Mono 6.12, PowerShell 7.1, and MATE DE pre-installed to run
 your .NET applications on Amazon Linux 2 with Long Term Support (LTS)."
    ],
    [
        "ami-0456example0def34",
        "amzn2-x86_64-MATEDE_DOTNET-2020.04.14",
        "Amazon Linux 2 with .Net Core, PowerShell, Mono, and MATE Desktop
 Environment"
    ]
]
```

   Choose the AMI that is appropriate for your use.

2. Launch an EC2 instance with the AMI that you located in the previous step. Configure the security group to allow for inbound TCP traffic to port 3389. For more information about configuring security groups, see Security groups for your VPC. This configuration enables you to use an RDP client to connect to the instance.
3. Connect to the instance using SSH.
4. Update the software and kernel on the instance.

```
[ec2-user ~]$ sudo yum update
```

After the update completes, reboot the instance to ensure that it is using the latest packages and libraries from the update; kernel updates are not loaded until a reboot occurs.

```
[ec2-user ~]$ sudo reboot
```

5. Reconnect to the instance and run the following command on your Linux instance to set the password for `ec2-user`.

```
[ec2-user ~]$ sudo passwd ec2-user
```

6. Install the certificate and key.

   If you already have a certificate and key, copy them to the `/etc/xrdp/` directory as follows:

   - Certificate — `/etc/xrdp/cert.pem`
   - Key — `/etc/xrdp/key.pem`

   If you do not have a certificate and key, use the following command to generate them in the `/etc/xrdp` directory.

```
$ sudo openssl req -x509 -sha384 -newkey rsa:3072 -nodes -keyout /etc/xrdp/key.pem
 -out /etc/xrdp/cert.pem -days 365
```

> **Note**
>
> This command generates a certificate that is valid for 365 days.

7. Open an RDP client on the computer from which you will connect to the instance (for example, Remote Desktop Connection on a computer running Microsoft Windows). Enter `ec2-user` as the user name and enter the password that you set in the previous step.

To disable `xrdp` on your Amazon EC2 instance

You can disable `xrdp` at any time by running one of the following commands on your Linux instance. The following commands do not impact your ability to use MATE using an X11 server.

```
[ec2-user ~]$ sudo systemctl disable xrdp
```

```
[ec2-user ~]$ sudo systemctl stop xrdp
```

To enable `xrdp` on your Amazon EC2 instance

To re-enable `xrdp` so that you can connect to your AL2 instance running the MATE desktop environment, run one of the following commands on your Linux instance.

```
[ec2-user ~]$ sudo systemctl enable xrdp
```

```
[ec2-user ~]$ sudo systemctl start xrdp
```

AL2 Tutorials

The following tutorials show you how to perform common tasks using Amazon EC2 instances running AL2. For video tutorials, see Amazon Instructional videos and labs.

For AL2023 instructions, see Tutorials in the *Amazon Linux 2023 User Guide*.

Tutorials

- Tutorial: Install a LAMP server on AL2
- Tutorial: Configure SSL/TLS on AL2
- Tutorial: Host a WordPress blog on AL2

Tutorial: Install a LAMP server on AL2

The following procedures help you install an Apache web server with PHP and MariaDB (a community-developed fork of MySQL) support on your AL2 instance (sometimes called a LAMP web server or LAMP stack). You can use this server to host a static website or deploy a dynamic PHP application that reads and writes information to a database.

> ⚠ **Important**
>
> If you are trying to set up a LAMP web server on a different distribution, such as Ubuntu or Red Hat Enterprise Linux, this tutorial will not work. For AL2023, see Install a LAMP server on AL2023. For Ubuntu, see the following Ubuntu community documentation: ApacheMySQLPHP. For other distributions, see their specific documentation.

Option: Complete this tutorial using automation

To complete this tutorial using Amazon Systems Manager Automation instead of the following tasks, run the AmazonDocs-InstallALAMPServer-AL2 Automation document.

Tasks

- Step 1: Prepare the LAMP server
- Step 2: Test your LAMP server
- Step 3: Secure the database server
- Step 4: (Optional) Install phpMyAdmin
- Troubleshoot
- Related topics

Step 1: Prepare the LAMP server

Prerequisites

- This tutorial assumes that you have already launched a new instance using AL2, with a public DNS name that is reachable from the internet. For more information, see Launch an instance in the *Amazon EC2 User Guide*. You must also have configured your security group to allow SSH (port 22), HTTP (port 80), and HTTPS (port 443) connections. For more information about these prerequisites, see Security group rules in the *Amazon EC2 User Guide*.
- The following procedure installs the latest PHP version available on AL2, currently `php8.2`. If you plan to use PHP applications other than those described in this tutorial, you should check their compatibility with `php8.2`.

To prepare the LAMP server

1. Connect to your instance.
2. To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process may take a few minutes, but it is important to make sure that you have the latest security updates and bug fixes.

   The `-y` option installs the updates without asking for confirmation. If you would like to examine the updates before installing, you can omit this option.

```
[ec2-user ~]$ sudo yum update -y
```

3. Install the `mariadb10.5` Amazon Linux Extras repositories to get the latest version of the MariaDB package.

   ```
   [ec2-user ~]$ sudo amazon-linux-extras install mariadb10.5
   ```

   If you receive an error stating `sudo: amazon-linux-extras: command not found`, then your instance was not launched with an Amazon Linux 2 AMI (perhaps you are using the Amazon Linux AMI instead). You can view your version of Amazon Linux using the following command.

   ```
   cat /etc/system-release
   ```

4. Install the php`8.2` Amazon Linux Extras repositories to get the latest version of the PHP package for AL2.

   ```
   [ec2-user ~]$ sudo amazon-linux-extras install php8.2
   ```

5. Now that your instance is current, you can install the Apache web server, MariaDB, and PHP software packages. Use the yum install command to install multiple software packages and all related dependencies at the same time

   ```
   [ec2-user ~]$ sudo yum install -y httpd
   ```

   You can view the current versions of these packages using the following command:

   ```
   yum info package_name
   ```

6. Start the Apache web server.

   ```
   [ec2-user ~]$ sudo systemctl start httpd
   ```

7. Use the **systemctl** command to configure the Apache web server to start at each system boot.

   ```
   [ec2-user ~]$ sudo systemctl enable httpd
   ```

   You can verify that **httpd** is on by running the following command:

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

8. Add a security rule to allow inbound HTTP (port 80) connections to your instance if you have not already done so. By default, a **launch-wizard-*N*** security group was set up for your instance during initialization. This group contains a single rule to allow SSH connections.

   a. Open the Amazon EC2 console at https://console.amazonaws.cn/ec2/.

   b. Choose **Instances** and select your instance.

   c. On the **Security** tab, view the inbound rules. You should see the following rule:

   ```
   Port range    Protocol     Source
   22            tcp          0.0.0.0/0
   ```

   > ⚠ **Warning**
   >
   > Using `0.0.0.0/0` allows all IPv4 addresses to access your instance using SSH. This is acceptable for a short time in a test environment, but it's unsafe for production environments. In production, you authorize only a specific IP address or range of addresses to access your instance.

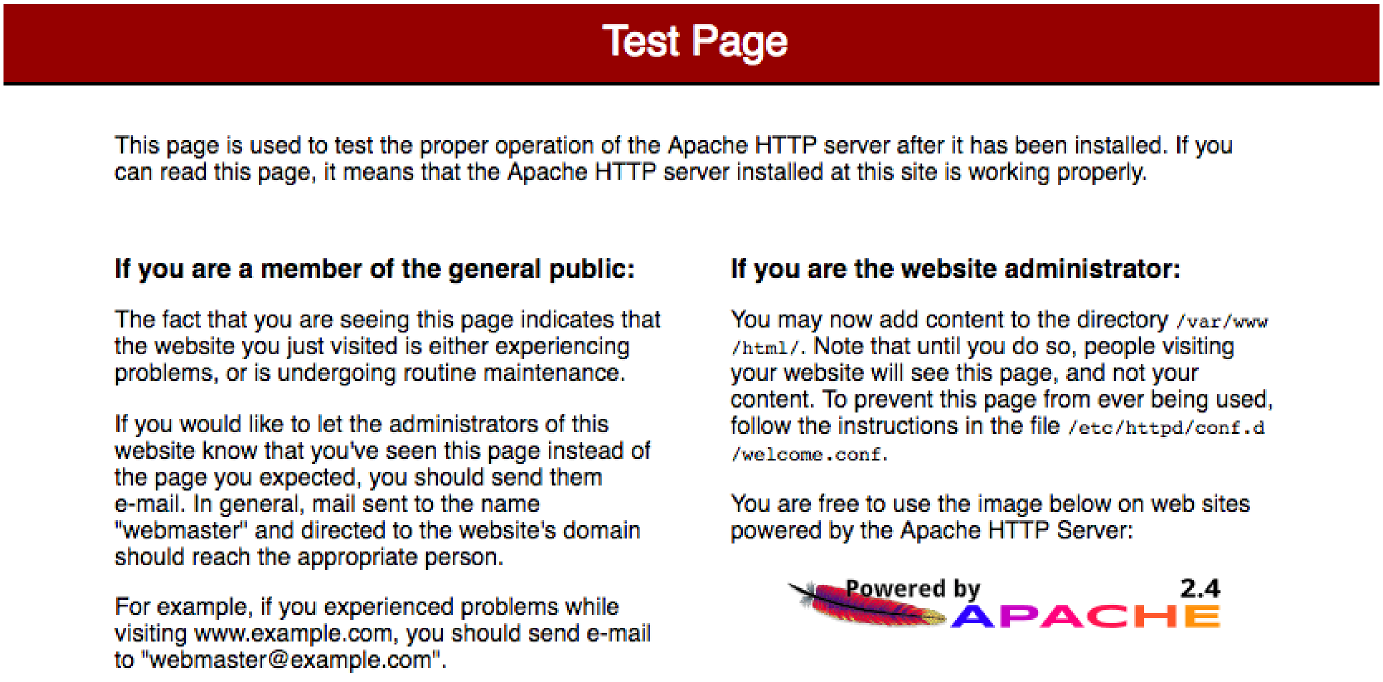
   d. Choose the link for the security group. Using the procedures in [Add rules to a security group](), add a new inbound security rule with the following values:

      - **Type**: HTTP
      - **Protocol**: TCP
      - **Port Range**: 80
      - **Source**: Custom

9. Test your web server. In a web browser, type the public DNS address (or the public IP address) of your instance. If there is no content in `/var/www/html`, you should see the Apache test page. You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS** column; if this column is hidden, choose **Show/Hide Columns** (the gear-shaped icon) and choose **Public DNS**).

   Verify that the security group for the instance contains a rule to allow HTTP traffic on port 80. For more information, see [Add rules to security group]().

> **Important**
>
> If you are not using Amazon Linux, you may also need to configure the firewall on your instance to allow these connections. For more information about how to configure the firewall, see the documentation for your specific distribution.



Test Page

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

If you are a member of the general public:

The fact that you are seeing this page indicates that the website you just visited is either experiencing problems, or is undergoing routine maintenance.

If you would like to let the administrators of this website know that you've seen this page instead of the page you expected, you should send them e-mail. In general, mail sent to the name "webmaster" and directed to the website's domain should reach the appropriate person.

For example, if you experienced problems while visiting www.example.com, you should send e-mail to "webmaster@example.com".

If you are the website administrator:

You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

You are free to use the image below on web sites powered by the Apache HTTP Server:

Apache **httpd** serves files that are kept in a directory called the Apache document root. The Amazon Linux Apache document root is `/var/www/html`, which by default is owned by root.

To allow the `ec2-user` account to manipulate files in this directory, you must modify the ownership and permissions of the directory. There are many ways to accomplish this task. In this tutorial, you add `ec2-user` to the `apache` group, to give the `apache` group ownership of the `/var/www` directory and assign write permissions to the group.

To set file permissions

1. Add your user (in this case, `ec2-user`) to the `apache` group.

```
[ec2-user ~]$ sudo usermod -a -G apache ec2-user
```

2. Log out and then log back in again to pick up the new group, and then verify your membership.

    a. Log out (use the **exit** command or close the terminal window):

    ```
    [ec2-user ~]$ exit
    ```

    b. To verify your membership in the `apache` group, reconnect to your instance, and then run the following command:

    ```
    [ec2-user ~]$ groups
    ec2-user adm wheel apache systemd-journal
    ```

3. Change the group ownership of `/var/www` and its contents to the `apache` group.

    ```
    [ec2-user ~]$ sudo chown -R ec2-user:apache /var/www
    ```

4. To add group write permissions and to set the group ID on future subdirectories, change the directory permissions of `/var/www` and its subdirectories.

    ```
    [ec2-user ~]$ sudo chmod 2775 /var/www && find /var/www -type d -exec sudo chmod
     2775 {} \;
    ```

5. To add group write permissions, recursively change the file permissions of `/var/www` and its subdirectories:

    ```
    [ec2-user ~]$ find /var/www -type f -exec sudo chmod 0664 {} \;
    ```

Now, `ec2-user` (and any future members of the `apache` group) can add, delete, and edit files in the Apache document root, enabling you to add content, such as a static website or a PHP application.

To secure your web server (Optional)

A web server running the HTTP protocol provides no transport security for the data that it sends or receives. When you connect to an HTTP server using a web browser, the URLs that you visit, the content of webpages that you receive, and the contents (including passwords) of any HTML forms that you submit are all visible to eavesdroppers anywhere along the network pathway. The best practice for securing your web server is to install support for HTTPS (HTTP Secure), which protects your data with SSL/TLS encryption.

For information about enabling HTTPS on your server, see Tutorial: Configure SSL/TLS on AL2.

Step 2: Test your LAMP server

If your server is installed and running, and your file permissions are set correctly, your `ec2-user` account should be able to create a PHP file in the `/var/www/html` directory that is available from the internet.

To test your LAMP server

1. Create a PHP file in the Apache document root.

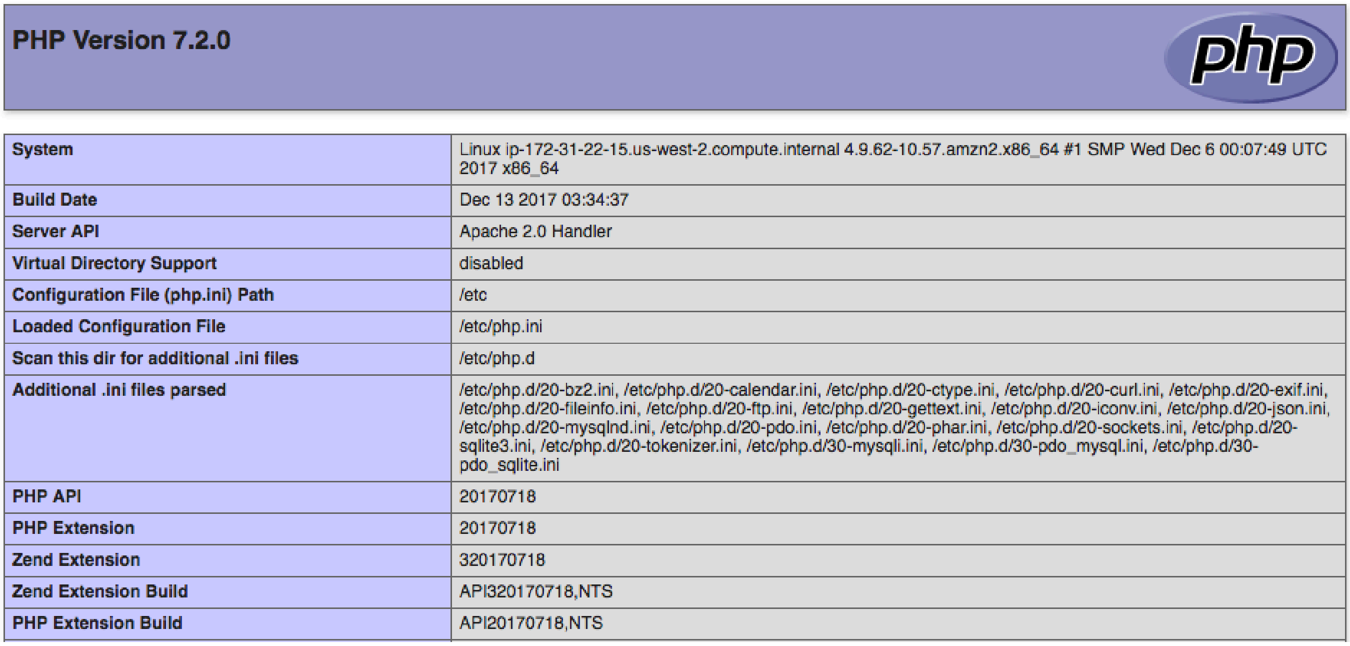
   ```
   [ec2-user ~]$ echo "<?php phpinfo(); ?>" > /var/www/html/phpinfo.php
   ```

   If you get a "Permission denied" error when trying to run this command, try logging out and logging back in again to pick up the proper group permissions that you configured in To set file permissions.

2. In a web browser, type the URL of the file that you just created. This URL is the public DNS address of your instance followed by a forward slash and the file name. For example:

   ```
   http://my.public.dns.amazonaws.com/phpinfo.php
   ```

   You should see the PHP information page:



PHP Version 7.2.0

System	Linux ip-172-31-22-15.us-west-2.compute.internal 4.9.62-10.57.amzn2.x86_64 #1 SMP Wed Dec 6 00:07:49 UTC 2017 x86_64
Build Date	Dec 13 2017 03:34:37
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-json.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS

If you do not see this page, verify that the `/var/www/html/phpinfo.php` file was created properly in the previous step. You can also verify that all of the required packages were installed with the following command.

```
[ec2-user ~]$ sudo yum list installed httpd mariadb-server php-mysqlnd
```

If any of the required packages are not listed in your output, install them with the **sudo yum install *package*** command. Also verify that the `php7.2` and `lamp-mariadb10.2-php7.2` extras are enabled in the output of the **amazon-linux-extras** command.

3. Delete the `phpinfo.php` file. Although this can be useful information, it should not be broadcast to the internet for security reasons.

```
[ec2-user ~]$ rm /var/www/html/phpinfo.php
```

You should now have a fully functional LAMP web server. If you add content to the Apache document root at `/var/www/html`, you should be able to view that content at the public DNS address for your instance.

Step 3: Secure the database server

The default installation of the MariaDB server has several features that are great for testing and development, but they should be disabled or removed for production servers. The

mysql_secure_installation command walks you through the process of setting a root password and removing the insecure features from your installation. Even if you are not planning on using the MariaDB server, we recommend performing this procedure.

To secure the MariaDB server

1. Start the MariaDB server.

   ```
   [ec2-user ~]$ sudo systemctl start mariadb
   ```

2. Run **mysql_secure_installation**.

   ```
   [ec2-user ~]$ sudo mysql_secure_installation
   ```

   a. When prompted, type a password for the root account.

      i. Type the current root password. By default, the root account does not have a password set. Press Enter.

      ii. Type **Y** to set a password, and type a secure password twice. For more information about creating a secure password, see https://identitysafe.norton.com/password-generator/. Make sure to store this password in a safe place.

      Setting a root password for MariaDB is only the most basic measure for securing your database. When you build or install a database-driven application, you typically create a database service user for that application and avoid using the root account for anything but database administration.

   b. Type **Y** to remove the anonymous user accounts.

   c. Type **Y** to disable the remote root login.

   d. Type **Y** to remove the test database.

   e. Type **Y** to reload the privilege tables and save your changes.

3. (Optional) If you do not plan to use the MariaDB server right away, stop it. You can restart it when you need it again.

   ```
   [ec2-user ~]$ sudo systemctl stop mariadb
   ```

4. (Optional) If you want the MariaDB server to start at every boot, type the following command.

```
[ec2-user ~]$ sudo systemctl enable mariadb
```

Step 4: (Optional) Install phpMyAdmin

phpMyAdmin is a web-based database management tool that you can use to view and edit the MySQL databases on your EC2 instance. Follow the steps below to install and configure `phpMyAdmin` on your Amazon Linux instance.

> **Important**
>
> We do not recommend using `phpMyAdmin` to access a LAMP server unless you have enabled SSL/TLS in Apache; otherwise, your database administrator password and other data are transmitted insecurely across the internet. For security recommendations from the developers, see Securing your phpMyAdmin installation. For general information about securing a web server on an EC2 instance, see Tutorial: Configure SSL/TLS on AL2.

To install phpMyAdmin

1. Install the required dependencies.

   ```
   [ec2-user ~]$ sudo yum install php-mbstring php-xml -y
   ```

2. Restart Apache.

   ```
   [ec2-user ~]$ sudo systemctl restart httpd
   ```

3. Restart `php-fpm`.

   ```
   [ec2-user ~]$ sudo systemctl restart php-fpm
   ```

4. Navigate to the Apache document root at `/var/www/html`.

   ```
   [ec2-user ~]$ cd /var/www/html
   ```

5. Select a source package for the latest phpMyAdmin release from https://www.phpmyadmin.net/downloads. To download the file directly to your instance, copy the link and paste it into a **wget** command, as in this example:

```
[ec2-user html]$ wget https://www.phpmyadmin.net/downloads/phpMyAdmin-latest-all-languages.tar.gz
```

6. Create a `phpMyAdmin` folder and extract the package into it with the following command.

```
[ec2-user html]$ mkdir phpMyAdmin && tar -xvzf phpMyAdmin-latest-all-languages.tar.gz -C phpMyAdmin --strip-components 1
```

7. Delete the *`phpMyAdmin-latest-all-languages.tar.gz`* tarball.

```
[ec2-user html]$ rm phpMyAdmin-latest-all-languages.tar.gz
```

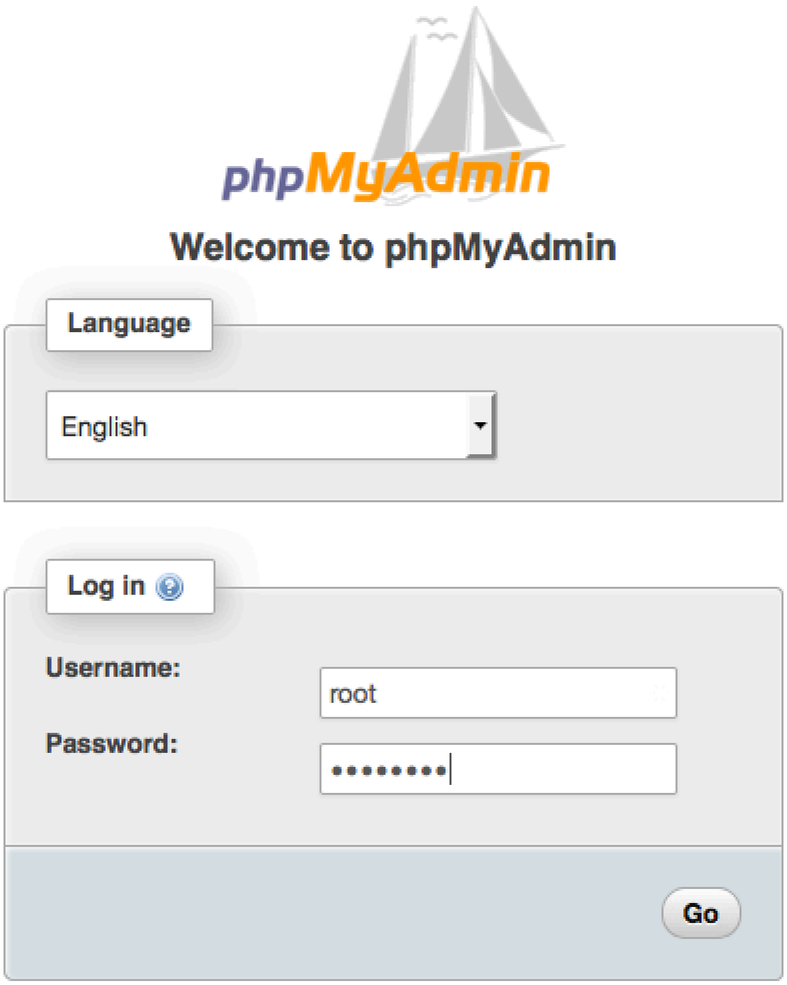
8. (Optional) If the MySQL server is not running, start it now.

```
[ec2-user ~]$ sudo systemctl start mariadb
```

9. In a web browser, type the URL of your phpMyAdmin installation. This URL is the public DNS address (or the public IP address) of your instance followed by a forward slash and the name of your installation directory. For example:

```
http://my.public.dns.amazonaws.com/phpMyAdmin
```

You should see the phpMyAdmin login page:



Welcome to phpMyAdmin

Language: English

Username: root

Password: •••••••••

10. Log in to your phpMyAdmin installation with the `root` user name and the MySQL root password you created earlier.

    Your installation must still be configured before you put it into service. We suggest that you begin by manually creating the configuration file, as follows:

    a. To start with a minimal configuration file, use your favorite text editor to create a new file, and then copy the contents of `config.sample.inc.php` into it.
    b. Save the file as `config.inc.php` in the phpMyAdmin directory that contains `index.php`.
    c. Refer to post-file creation instructions in the Using the Setup script section of the phpMyAdmin installation instructions for any additional setup.

For information about using phpMyAdmin, see the phpMyAdmin User Guide.

Troubleshoot

This section offers suggestions for resolving common problems you may encounter while setting up a new LAMP server.

I can't connect to my server using a web browser

Perform the following checks to see if your Apache web server is running and accessible.

- **Is the web server running?**

  You can verify that **httpd** is on by running the following command:

  ```
  [ec2-user ~]$ sudo systemctl is-enabled httpd
  ```

  If the **httpd** process is not running, repeat the steps described in To prepare the LAMP server.
- **Is the firewall correctly configured?**

  Verify that the security group for the instance contains a rule to allow HTTP traffic on port 80. For more information, see Add rules to security group.

I can't connect to my server using HTTPS

Perform the following checks to see if your Apache web server is configured to support HTTPS.

- **Is the web server correctly configured?**

  After you install Apache, the server is configured for HTTP traffic. To support HTTPS, enable TLS on the server and install an SSL certificate. For information, see Tutorial: Configure SSL/TLS on AL2.
- **Is the firewall correctly configured?**

  Verify that the security group for the instance contains a rule to allow HTTPS traffic on port 443. For more information, see Add rules to a security group.

Related topics

For more information about transferring files to your instance or installing a WordPress blog on your web server, see the following documentation:

- Transfer files to your Linux instance using WinSCP.
- Transfer files to Linux instances using an SCP client.
- Tutorial: Host a WordPress blog on AL2

For more information about the commands and software used in this tutorial, see the following webpages:

- Apache web server: http://httpd.apache.org/
- MariaDB database server: https://mariadb.org/
- PHP programming language: http://php.net/
- The `chmod` command: https://en.wikipedia.org/wiki/Chmod
- The `chown` command: https://en.wikipedia.org/wiki/Chown

For more information about registering a domain name for your web server, or transferring an existing domain name to this host, see Creating and Migrating Domains and Subdomains to Amazon Route 53 in the *Amazon Route 53 Developer Guide*.

Tutorial: Configure SSL/TLS on AL2

Secure Sockets Layer/Transport Layer Security (SSL/TLS) creates an encrypted channel between a web server and web client that protects data in transit from being eavesdropped on. This tutorial explains how to add support manually for SSL/TLS on an EC2 instance with AL2 and Apache web server. This tutorial assumes that you are not using a load balancer. If you are using Elastic Load Balancing, you can choose to configure SSL offload on the load balancer, using a certificate from Amazon Certificate Manager instead.

For historical reasons, web encryption is often referred to simply as SSL. While web browsers still support SSL, its successor protocol TLS is less vulnerable to attack. AL2 disables server-side support for all versions of SSL by default. Security standards bodies consider TLS 1.0 to be unsafe. TLS 1.0 and TLS 1.1 were formally deprecated in March 2021. This tutorial contains guidance based exclusively on enabling TLS 1.2. TLS 1.3 was finalized in 2018 and is available in AL2 as long as the underlying TLS library (OpenSSL in this tutorial) is supported and enabled. Clients must support TLS 1.2 or later by June 28, 2023. For more information about the updated encryption standards, see RFC 7568 and RFC 8446.

This tutorial refers to modern web encryption simply as TLS.

Important

These procedures are intended for use with AL2. We also assume that you are starting with a new Amazon EC2 instance. If you are trying to set up an EC2 instance running a different distribution, or an instance running an old version of AL2, some procedures in this tutorial might not work. For Ubuntu, see the following community documentation: Open SSL on Ubuntu. For Red Hat Enterprise Linux, see the following: Setting up the Apache HTTP Web Server. For other distributions, see their specific documentation.

Note

Alternatively, you can use Amazon Certificate Manager (ACM) for Amazon Nitro enclaves, which is an enclave application that allows you to use public and private SSL/TLS certificates with your web applications and servers running on Amazon EC2 instances with Amazon Nitro Enclaves. Nitro Enclaves is an Amazon EC2 capability that enables creation of isolated compute environments to protect and securely process highly sensitive data, such as SSL/TLS certificates and private keys.

ACM for Nitro Enclaves works with **nginx** running on your Amazon EC2 Linux instance to create private keys, to distribute certificates and private keys, and to manage certificate renewals.

To use ACM for Nitro Enclaves, you must use an enclave-enabled Linux instance.

For more information, see What is Amazon Nitro Enclaves? and Amazon Certificate Manager for Nitro Enclaves in the *Amazon Nitro Enclaves User Guide*.

Contents

- Prerequisites
- Step 1: Enable TLS on the server
- Step 2: Obtain a CA-signed certificate
- Step 3: Test and harden the security configuration
- Troubleshoot

Prerequisites

Before you begin this tutorial, complete the following steps:

- Launch an Amazon EBS backed AL2 instance. For more information, see Launch an instance in the *Amazon EC2 User Guide*.
- Configure your security groups to allow your instance to accept connections on the following TCP ports:
  - SSH (port 22)
  - HTTP (port 80)
  - HTTPS (port 443)

  For more information, see Security group rules in the *Amazon EC2 User Guide*.
- Install the Apache web server. For step-by-step instructions, see Tutorial: Install a LAMP Web Server on AL2. Only the httpd package and its dependencies are needed, so you can ignore the instructions involving PHP and MariaDB.
- To identify and authenticate websites, the TLS public key infrastructure (PKI) relies on the Domain Name System (DNS). To use your EC2 instance to host a public website, you need to register a domain name for your web server or transfer an existing domain name to your Amazon EC2 host. Numerous third-party domain registration and DNS hosting services are available for this, or you can use Amazon Route 53.

Step 1: Enable TLS on the server

Option: Complete this tutorial using automation

To complete this tutorial using Amazon Systems Manager Automation instead of the following tasks, run the automation document.

This procedure takes you through the process of setting up TLS on AL2 with a self-signed digital certificate.

Note

A self-signed certificate is acceptable for testing but not production. If you expose your self-signed certificate to the internet, visitors to your site are greeted by security warnings.

To enable TLS on a server

1. Connect to your instance and confirm that Apache is running.

```
[ec2-user ~]$ sudo systemctl is-enabled httpd
```

If the returned value is not "enabled," start Apache and set it to start each time the system boots.

```
[ec2-user ~]$ sudo systemctl start httpd && sudo systemctl enable httpd
```

2. To ensure that all of your software packages are up to date, perform a quick software update on your instance. This process may take a few minutes, but it is important to make sure that you have the latest security updates and bug fixes.

> **Note**
>
> The `-y` option installs the updates without asking for confirmation. If you would like to examine the updates before installing, you can omit this option.

```
[ec2-user ~]$ sudo yum update -y
```

3. Now that your instance is current, add TLS support by installing the Apache module `mod_ssl`.

```
[ec2-user ~]$ sudo yum install -y mod_ssl
```

Your instance now has the following files that you use to configure your secure server and create a certificate for testing:

- `/etc/httpd/conf.d/ssl.conf`

  The configuration file for mod_ssl. It contains *directives* telling Apache where to find encryption keys and certificates, the TLS protocol versions to allow, and the encryption ciphers to accept.

- `/etc/pki/tls/certs/make-dummy-cert`

  A script to generate a self-signed X.509 certificate and private key for your server host. This certificate is useful for testing that Apache is properly set up to use TLS. Because it offers no proof of identity, it should not be used in production. If used in production, it triggers warnings in Web browsers.

4. Run the script to generate a self-signed dummy certificate and key for testing.

   ```
   [ec2-user ~]$ cd /etc/pki/tls/certs
   sudo ./make-dummy-cert localhost.crt
   ```

   This generates a new file `localhost.crt` in the `/etc/pki/tls/certs/` directory. The specified file name matches the default that is assigned in the **SSLCertificateFile** directive in `/etc/httpd/conf.d/ssl.conf`.

   This file contains both a self-signed certificate and the certificate's private key. Apache requires the certificate and key to be in PEM format, which consists of Base64-encoded ASCII characters framed by "BEGIN" and "END" lines, as in the following abbreviated example.

   ```
   -----BEGIN PRIVATE KEY-----
   MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQD2KKx/8Zk94m1q
   3gQMZF9ZN66Ls19+3tHAgQ5Fpo9KJDhzLjOOCI8u1PTcGmAah5kEitCEc0wzmNeo
   BCl0wYR6G0rGaKtK9Dn7CuIjvubtUysVyQoMVPQ97ldeakHWeRMiEJFXg6kZZ0vr
   GvwnKoMh3DlK44D9dX7IDua2PlYx5+eroA+1Lqf32ZSaAO0bBIMIYTHigwbHMZoT
   ...
   56tE7THvH7vOEf4/iUOsIrEzaMaJ0mqkmY1A70qQGQKBgBF3H1qNRNHuyMcPODFs
   27hDzPDinrquSEvoZIggkDMlh2irTiipJ/GhkvTpoQlv0fK/VXw8vSgeaBuhwJvS
   LXU9HvYq0U6O4FgD3nAyB9hI0BE13r1HjUvbjT7moH+RhnNz6eqqdscCS09VtRAo
   4QQvAqOa8UheYeoXLdWcHaLP
   -----END PRIVATE KEY-----

   -----BEGIN CERTIFICATE-----
   MIIEazCCA1OgAwIBAgICWxQwDQYJKoZIhvcNAQELBQAwgbExCzAJBgNVBAYTAi0t
   MRIwEAYDVQQIDAlTb21lU3RhdGUxETAPBgNVBAcMCFNvbWVDaXR5MRkwFwYDVQQK
   DBBTb21lT3JnYW5pemF0aW9uMR8wHQYDVQQLDBZTb21lT3JnYW5pemF0aW9uYWxV
   bml0MRkwFwYDVQQDDBBpcC0xNzItMzEtMjAtMjM2MSQwIgYJKoZIhvcNAQkBFhVy
   ...
   z5rRUE/XzxRLBZOoWZpNWTXJkQ3uFYH6s/sBwtHpKKZMzOvDedREjNKAvk4ws6F0
   CuIjvubtUysVyQoMVPQ97ldeakHWeRMiEJFXg6kZZ0vrGvwnKoMh3DlK44D9dlU3
   WanXWehT6FiSZvB4sTEXXJN2jdw8g+sHGnZ8zCOsclknYhHrCVD2vnBlZJKSZvak
   3ZazhBxtQSukFMOnWPP2a0DMMFGYUHOd0BQE8sBJxg==
   -----END CERTIFICATE-----
   ```

   The file names and extensions are a convenience and have no effect on function. For example, you can call a certificate `cert.crt`, `cert.pem`, or any other file name, so long as the related directive in the `ssl.conf` file uses the same name.

> **Note**
>
> When you replace the default TLS files with your own customized files, be sure that they are in PEM format.

5. Open the `/etc/httpd/conf.d/ssl.conf` file using your favorite text editor (such as **vim** or **nano**) as root user and comment out the following line, because the self-signed dummy certificate also contains the key. If you do not comment out this line before you complete the next step, the Apache service fails to start.

   ```
   SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
   ```

6. Restart Apache.

   ```
   [ec2-user ~]$ sudo systemctl restart httpd
   ```

   > **Note**
   >
   > Make sure that TCP port 443 is accessible on your EC2 instance, as previously described.

7. Your Apache web server should now support HTTPS (secure HTTP) over port 443. Test it by entering the IP address or fully qualified domain name of your EC2 instance into a browser URL bar with the prefix **`https://`**.

   Because you are connecting to a site with a self-signed, untrusted host certificate, your browser may display a series of security warnings. Override the warnings and proceed to the site.

   If the default Apache test page opens, it means that you have successfully configured TLS on your server. All data passing between the browser and server is now encrypted.

   > **Note**
   >
   > To prevent site visitors from encountering warning screens, you must obtain a trusted, CA-signed certificate that not only encrypts, but also publicly authenticates you as the owner of the site.

Step 2: Obtain a CA-signed certificate

You can use the following process to obtain a CA-signed certificate:

- Generate a certificate signing request (CSR) from a private key
- Submit the CSR to a certificate authority (CA)
- Obtain a signed host certificate
- Configure Apache to use the certificate

A self-signed TLS X.509 host certificate is cryptologically identical to a CA-signed certificate. The difference is social, not mathematical. A CA promises, at a minimum, to validate a domain's ownership before issuing a certificate to an applicant. Each web browser contains a list of CAs trusted by the browser vendor to do this. An X.509 certificate consists primarily of a public key that corresponds to your private server key, and a signature by the CA that is cryptographically tied to the public key. When a browser connects to a web server over HTTPS, the server presents a certificate for the browser to check against its list of trusted CAs. If the signer is on the list, or accessible through a *chain of trust* consisting of other trusted signers, the browser negotiates a fast encrypted data channel with the server and loads the page.

Certificates generally cost money because of the labor involved in validating the requests, so it pays to shop around. A few CAs offer basic-level certificates free of charge. The most notable of these CAs is the Let's Encrypt project, which also supports the automation of the certificate creation and renewal process. For more information about using a Let's Encrypt certificate, see Get Certbot.

If you plan to offer commercial-grade services, Amazon Certificate Manager is a good option.

Underlying the host certificate is the key. As of 2019, government and industry groups recommend using a minimum key (modulus) size of 2048 bits for RSA keys intended to protect documents, through 2030. The default modulus size generated by OpenSSL in AL2 is 2048 bits, which is suitable for use in a CA-signed certificate. In the following procedure, an optional step provided for those who want a customized key, for example, one with a larger modulus or using a different encryption algorithm.

⚠ **Important**

These instructions for acquiring a CA-signed host certificate do not work unless you own a registered and hosted DNS domain.

To obtain a CA-signed certificate

1. Connect to your instance and navigate to /etc/pki/tls/private/. This is the directory where you store the server's private key for TLS. If you prefer to use an existing host key to generate the CSR, skip to Step 3.
2. (Optional) Generate a new private key. Here are some examples of key configurations. Any of the resulting keys works with your web server, but they vary in the degree and type of security that they implement.

   - **Example 1:** Create a default RSA host key. The resulting file, **`custom.key`**, is a 2048-bit RSA private key.

     ```
     [ec2-user ~]$ sudo openssl genrsa -out custom.key
     ```

   - **Example 2:** Create a stronger RSA key with a bigger modulus. The resulting file, **`custom.key`**, is a 4096-bit RSA private key.

     ```
     [ec2-user ~]$ sudo openssl genrsa -out custom.key 4096
     ```

   - **Example 3:** Create a 4096-bit encrypted RSA key with password protection. The resulting file, **`custom.key`**, is a 4096-bit RSA private key encrypted with the AES-128 cipher.

     ⚠ **Important**

     Encrypting the key provides greater security, but because an encrypted key requires a password, services depending on it cannot be auto-started. Each time you use this key, you must supply the password (in the preceding example, "abcde12345") over an SSH connection.

     ```
     [ec2-user ~]$ sudo openssl genrsa -aes128 -passout pass:abcde12345 -out
      custom.key 4096
     ```

- **Example 4:** Create a key using a non-RSA cipher. RSA cryptography can be relatively slow because of the size of its public keys, which are based on the product of two large prime numbers. However, it is possible to create keys for TLS that use non-RSA ciphers. Keys based on the mathematics of elliptic curves are smaller and computationally faster when delivering an equivalent level of security.

  ```
  [ec2-user ~]$ sudo openssl ecparam -name prime256v1 -out custom.key -genkey
  ```

  The result is a 256-bit elliptic curve private key using prime256v1, a "named curve" that OpenSSL supports. Its cryptographic strength is slightly greater than a 2048-bit RSA key, according to NIST.

  > **Note**
  >
  > Not all CAs provide the same level of support for elliptic-curve-based keys as for RSA keys.

Make sure that the new private key has highly restrictive ownership and permissions (owner=root, group=root, read/write for owner only). The commands would be as shown in the following example.

```
[ec2-user ~]$ sudo chown root:root custom.key
[ec2-user ~]$ sudo chmod 600 custom.key
[ec2-user ~]$ ls -al custom.key
```

The preceding commands yield the following result.

```
-rw------- root root custom.key
```

After you have created and configured a satisfactory key, you can create a CSR.

3. Create a CSR using your preferred key. The following example uses **`custom.key`**.

```
[ec2-user ~]$ sudo openssl req -new -key custom.key -out csr.pem
```

OpenSSL opens a dialog and prompts you for the information shown in the following table. All of the fields except **Common Name** are optional for a basic, domain-validated host certificate.

Name	Description	Example
Country Name	The two-letter ISO abbreviation for your country.	US (=United States)
State or Province Name	The name of the state or province where your organization is located. This name cannot be abbreviated.	Washington
Locality Name	The location of your organization, such as a city.	Seattle
Organization Name	The full legal name of your organization. Do not abbreviate your organization name.	Example Corporation
Organizational Unit Name	Additional organizational information, if any.	Example Dept
Common Name	This value must exactly match the web address that you expect users to enter into a browser. Usually, this means a domain name with a prefixed hostname or alias in the form **`www.example.com`** . In testing with a self-signed certificate and no DNS resolution, the common name may consist of the hostname alone. CAs also offer more expensive certificates that accept wild-card names such as **`*.example.com`** .	www.example.com
Email Address	The server administrator's email address.	someone@example.com

Finally, OpenSSL prompts you for an optional challenge password. This password applies only to the CSR and to transactions between you and your CA, so follow the CA's recommendations about this and the other optional field, optional company name. The CSR challenge password has no effect on server operation.

The resulting file **`csr.pem`** contains your public key, your digital signature of your public key, and the metadata that you entered.

4. Submit the CSR to a CA. This usually consists of opening your CSR file in a text editor and copying the contents into a web form. At this time, you may be asked to supply one or more subject alternate names (SANs) to be placed on the certificate. If **`www.example.com`** is the common name, then **`example.com`** would be a good SAN, and vice versa. A visitor to your site entering either of these names would see an error-free connection. If your CA web form allows it, include the common name in the list of SANs. Some CAs include it automatically.

   After your request has been approved, you receive a new host certificate signed by the CA. You might also be instructed to download an *intermediate certificate* file that contains additional certificates needed to complete the CA's chain of trust.

   Note

   Your CA might send you files in multiple formats intended for various purposes. For this tutorial, you should only use a certificate file in PEM format, which is usually (but not always) marked with a `.pem` or `.crt` file extension. If you are uncertain which file to use, open the files with a text editor and find the one containing one or more blocks beginning with the following line.

   ```
   - - - - -BEGIN CERTIFICATE - - - - -
   ```

   The file should also end with the following line.

   ```
   - - - -END CERTIFICATE - - - - -
   ```

   You can also test the file at the command line as shown in the following.

   ```
   [ec2-user certs]$ openssl x509 -in certificate.crt -text
   ```

   Verify that these lines appear in the file. Do not use files ending with `.p7b`, `.p7c`, or similar file extensions.

5. Place the new CA-signed certificate and any intermediate certificates in the `/etc/pki/tls/certs` directory.

   > **Note**
   >
   > There are several ways to upload your new certificate to your EC2 instance, but the most straightforward and informative way is to open a text editor (for example, vi, nano, or notepad) on both your local computer and your instance, and then copy and paste the file contents between them. You need root [sudo] permissions when performing these operations on the EC2 instance. This way, you can see immediately if there are any permission or path problems. Be careful, however, not to add any additional lines while copying the contents, or to change them in any way.

   From inside the `/etc/pki/tls/certs` directory, check that the file ownership, group, and permission settings match the highly restrictive AL2 defaults (owner=root, group=root, read/write for owner only). The following example shows the commands to use.

   ```
   [ec2-user certs]$ sudo chown root:root custom.crt
   [ec2-user certs]$ sudo chmod 600 custom.crt
   [ec2-user certs]$ ls -al custom.crt
   ```

   These commands should yield the following result.

   ```
   -rw------- root root custom.crt
   ```

   The permissions for the intermediate certificate file are less stringent (owner=root, group=root, owner can write, group can read, world can read). The following example shows the commands to use.

   ```
   [ec2-user certs]$ sudo chown root:root intermediate.crt
   [ec2-user certs]$ sudo chmod 644 intermediate.crt
   [ec2-user certs]$ ls -al intermediate.crt
   ```

   These commands should yield the following result.

   ```
   -rw-r--r-- root root intermediate.crt
   ```

6. Place the private key that you used to create the CSR in the `/etc/pki/tls/private/` directory.

> **Note**
>
> There are several ways to upload your custom key to your EC2 instance, but the most straightforward and informative way is to open a text editor (for example, vi, nano, or notepad) on both your local computer and your instance, and then copy and paste the file contents between them. You need root [sudo] permissions when performing these operations on the EC2 instance. This way, you can see immediately if there are any permission or path problems. Be careful, however, not to add any additional lines while copying the contents, or to change them in any way.

   From inside the `/etc/pki/tls/private` directory, use the following commands to verify that the file ownership, group, and permission settings match the highly restrictive AL2 defaults (owner=root, group=root, read/write for owner only).

```
[ec2-user private]$ sudo chown root:root custom.key
[ec2-user private]$ sudo chmod 600 custom.key
[ec2-user private]$ ls -al custom.key
```

   These commands should yield the following result.

```
-rw------- root root custom.key
```

7. Edit `/etc/httpd/conf.d/ssl.conf` to reflect your new certificate and key files.

   a. Provide the path and file name of the CA-signed host certificate in Apache's `SSLCertificateFile` directive:

```
SSLCertificateFile /etc/pki/tls/certs/custom.crt
```

   b. If you received an intermediate certificate file (`intermediate.crt` in this example), provide its path and file name using Apache's `SSLCACertificateFile` directive:

```
SSLCACertificateFile /etc/pki/tls/certs/intermediate.crt
```

> **Note**
>
> Some CAs combine the host certificate and the intermediate certificates in a single file, making the `SSLCACertificateFile` directive unnecessary. Consult the instructions provided by your CA.

c. Provide the path and file name of the private key (`custom.key` in this example) in Apache's `SSLCertificateKeyFile` directive:

```
SSLCertificateKeyFile /etc/pki/tls/private/custom.key
```

8. Save `/etc/httpd/conf.d/ssl.conf` and restart Apache.

```
[ec2-user ~]$ sudo systemctl restart httpd
```

9. Test your server by entering your domain name into a browser URL bar with the prefix `https://`. Your browser should load the test page over HTTPS without generating errors.

Step 3: Test and harden the security configuration

After your TLS is operational and exposed to the public, you should test how secure it really is. This is easy to do using online services such as Qualys SSL Labs, which performs a free and thorough analysis of your security setup. Based on the results, you may decide to harden the default security configuration by controlling which protocols you accept, which ciphers you prefer, and which you exclude. For more information, see how Qualys formulates its scores.

> **Important**
>
> Real-world testing is crucial to the security of your server. Small configuration errors may lead to serious security breaches and loss of data. Because recommended security practices change constantly in response to research and emerging threats, periodic security audits are essential to good server administration.

On the Qualys SSL Labs site, enter the fully qualified domain name of your server, in the form **`www.example.com`**. After about two minutes, you receive a grade (from A to F) for your site and a detailed breakdown of the findings. The following table summarizes the report for a domain with settings identical to the default Apache configuration on AL2, and with a default Certbot certificate.

Overall rating	B
Certificate	100%
Protocol support	95%
Key exchange	70%
Cipher strength	90%

Though the overview shows that the configuration is mostly sound, the detailed report flags several potential problems, listed here in order of severity:

✗ **The RC4 cipher is supported for use by certain older browsers.** A cipher is the mathematical core of an encryption algorithm. RC4, a fast cipher used to encrypt TLS data-streams, is known to have several serious weaknesses. Unless you have very good reasons to support legacy browsers, you should disable this.

✗ **Old TLS versions are supported.** The configuration supports TLS 1.0 (already deprecated) and TLS 1.1 (on a path to deprecation). Only TLS 1.2 has been recommended since 2018.

✗ **Forward secrecy is not fully supported.** Forward secrecy is a feature of algorithms that encrypt using temporary (ephemeral) session keys derived from the private key. This means in practice that attackers cannot decrypt HTTPS data even if they possess a web server's long-term private key.

To correct and future-proof the TLS configuration

1. Open the configuration file `/etc/httpd/conf.d/ssl.conf` in a text editor and comment out the following line by entering "#" at the beginning of the line.

   ```
   #SSLProtocol all -SSLv3
   ```

2. Add the following directive:

   ```
   #SSLProtocol all -SSLv3
   SSLProtocol -SSLv2 -SSLv3 -TLSv1 -TLSv1.1 +TLSv1.2
   ```

This directive explicitly disables SSL versions 2 and 3, as well as TLS versions 1.0 and 1.1. The server now refuses to accept encrypted connections with clients using anything except TLS 1.2. The verbose wording in the directive conveys more clearly, to a human reader, what the server is configured to do.

> **Note**
>
> Disabling TLS versions 1.0 and 1.1 in this manner blocks a small percentage of outdated web browsers from accessing your site.

To modify the list of allowed ciphers

1. In the configuration file `/etc/httpd/conf.d/ssl.conf`, find the section with the **`SSLCipherSuite`** directive and comment out the existing line by entering "#" at the beginning of the line.

   ```
   #SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5
   ```

2. Specify explicit cipher suites and a cipher order that prioritizes forward secrecy and avoids insecure ciphers. The `SSLCipherSuite` directive used here is based on output from the Mozilla SSL Configuration Generator, which tailors a TLS configuration to the specific software running on your server. First determine your Apache and OpenSSL versions by using the output from the following commands.

   ```
   [ec2-user ~]$ yum list installed | grep httpd

   [ec2-user ~]$ yum list installed | grep openssl
   ```

   For example, if the returned information is Apache 2.4.34 and OpenSSL 1.0.2, we enter this into the generator. If you choose the "modern" compatibility model, this creates an `SSLCipherSuite` directive that aggressively enforces security but still works for most browsers. If your software doesn't support the modern configuration, you can update your software or choose the "intermediate" configuration instead.

   ```
   SSLCipherSuite ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-
   ECDSA-CHACHA20-POLY1305:
   ```

```
ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
SHA256:
ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-
RSA-AES128-SHA256
```

The selected ciphers have *ECDHE* in their names, an abbreviation for *Elliptic Curve Diffie-Hellman Ephemeral* . The term *ephemeral* indicates forward secrecy. As a by-product, these ciphers do not support RC4.

We recommend that you use an explicit list of ciphers instead of relying on defaults or terse directives whose content isn't visible.

Copy the generated directive into `/etc/httpd/conf.d/ssl.conf`.

> **Note**
>
> Though shown here on several lines for readability, the directive must be on a single line when copied to `/etc/httpd/conf.d/ssl.conf`, with only a colon (no spaces) between cipher names.

3. Finally, uncomment the following line by removing the "#" at the beginning of the line.

```
#SSLHonorCipherOrder on
```

This directive forces the server to prefer high-ranking ciphers, including (in this case) those that support forward secrecy. With this directive turned on, the server tries to establish a strong secure connection before falling back to allowed ciphers with lesser security.

After completing both of these procedures, save the changes to `/etc/httpd/conf.d/ssl.conf` and restart Apache.

If you test the domain again on [Qualys SSL Labs](), you should see that the RC4 vulnerability and other warnings are gone and the summary looks something like the following.

Overall rating	A
Certificate	100%

Protocol support	100%
Key exchange	90%
Cipher strength	90%

Each update to OpenSSL introduces new ciphers and removes support for old ones. Keep your EC2 AL2 instance up-to-date, watch for security announcements from OpenSSL, and be alert to reports of new security exploits in the technical press.

Troubleshoot

- **My Apache webserver doesn't start unless I enter a password**

  This is expected behavior if you installed an encrypted, password-protected, private server key.

  You can remove the encryption and password requirement from the key. Assuming that you have a private encrypted RSA key called `custom.key` in the default directory, and that the password on it is **abcde12345**, run the following commands on your EC2 instance to generate an unencrypted version of the key.

  ```
  [ec2-user ~]$ cd /etc/pki/tls/private/
  [ec2-user private]$ sudo cp custom.key custom.key.bak
  [ec2-user private]$ sudo openssl rsa -in custom.key -passin pass:abcde12345 -out custom.key.nocrypt
  [ec2-user private]$ sudo mv custom.key.nocrypt custom.key
  [ec2-user private]$ sudo chown root:root custom.key
  [ec2-user private]$ sudo chmod 600 custom.key
  [ec2-user private]$ sudo systemctl restart httpd
  ```

  Apache should now start without prompting you for a password.

- **I get errors when I run sudo yum install -y mod_ssl.**

  When you are installing the required packages for SSL, you may see errors similar to the following.

  ```
  Error: httpd24-tools conflicts with httpd-tools-2.2.34-1.16.amzn1.x86_64
  Error: httpd24 conflicts with httpd-2.2.34-1.16.amzn1.x86_64
  ```

This typically means that your EC2 instance is not running AL2. This tutorial only supports instances freshly created from an official AL2 AMI.

Tutorial: Host a WordPress blog on AL2

The following procedures will help you install, configure, and secure a WordPress blog on your AL2 instance. This tutorial is a good introduction to using Amazon EC2 in that you have full control over a web server that hosts your WordPress blog, which is not typical with a traditional hosting service.

You are responsible for updating the software packages and maintaining security patches for your server. For a more automated WordPress installation that does not require direct interaction with the web server configuration, the Amazon CloudFormation service provides a WordPress template that can also get you started quickly. For more information, see Get started in the *Amazon CloudFormation User Guide*. If you need a high-availability solution with a decoupled database, see Deploying a high-availability WordPress website in the *Amazon Elastic Beanstalk Developer Guide*.

> ⚠ **Important**
>
> These procedures are intended for use with AL2. For more information about other distributions, see their specific documentation. Many steps in this tutorial do not work on Ubuntu instances. For help installing WordPress on an Ubuntu instance, see WordPress in the Ubuntu documentation. You can also use CodeDeploy to accomplish this task on Amazon Linux, macOS, or Unix systems.

Topics

- Prerequisites
- Install WordPress
- Next steps
- Help! My public DNS name changed and now my blog is broken

Prerequisites

This tutorial assumes that you have launched an AL2 instance with a functional web server with PHP and database (either MySQL or MariaDB) support by following all of the steps in Tutorial:

Install a LAMP server on AL2. This tutorial also has steps for configuring a security group to allow HTTP and HTTPS traffic, as well as several steps to ensure that file permissions are set properly for your web server. For information about adding rules to your security group, see Add rules to a security group.

We strongly recommend that you associate an Elastic IP address (EIP) to the instance you are using to host a WordPress blog. This prevents the public DNS address for your instance from changing and breaking your installation. If you own a domain name and you want to use it for your blog, you can update the DNS record for the domain name to point to your EIP address (for help with this, contact your domain name registrar). You can have one EIP address associated with a running instance at no charge. For more information, see Elastic IP addresses in the *Amazon EC2 User Guide*.

If you don't already have a domain name for your blog, you can register a domain name with Route 53 and associate your instance's EIP address with your domain name. For more information, see Registering domain names using Amazon Route 53 in the *Amazon Route 53 Developer Guide*.

Install WordPress

Option: Complete this tutorial using automation

To complete this tutorial using Amazon Systems Manager Automation instead of the following tasks, run the automation document.

Connect to your instance, and download the WordPress installation package.

To download and unzip the WordPress installation package

1. Download the latest WordPress installation package with the **wget** command. The following command should always download the latest release.

```
[ec2-user ~]$ wget https://wordpress.org/latest.tar.gz
```

2. Unzip and unarchive the installation package. The installation folder is unzipped to a folder called `wordpress`.

```
[ec2-user ~]$ tar -xzf latest.tar.gz
```

To create a database user and database for your WordPress installation

Your WordPress installation needs to store information, such as blog posts and user comments, in a database. This procedure helps you create your blog's database and a user that is authorized to read and save information to it.

1. Start the database server.

   -
     ```
     [ec2-user ~]$ sudo systemctl start mariadb
     ```

2. Log in to the database server as the `root` user. Enter your database `root` password when prompted; this may be different than your `root` system password, or it might even be empty if you have not secured your database server.

   If you have not secured your database server yet, it is important that you do so. For more information, see To secure the MariaDB server (AL2).

   ```
   [ec2-user ~]$ mysql -u root -p
   ```

3. Create a user and password for your MySQL database. Your WordPress installation uses these values to communicate with your MySQL database.

   Make sure that you create a strong password for your user. Do not use the single quote character (') in your password, because this will break the preceding command. Do not reuse an existing password, and make sure to store this password in a safe place.

   Enter the following command, substituting a unique user name and password.

   ```
   CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'your_strong_password';
   ```

4. Create your database. Give your database a descriptive, meaningful name, such as `wordpress-db`.

   > **Note**
   >
   > The punctuation marks surrounding the database name in the command below are called backticks. The backtick (`) key is usually located above the `Tab` key on a standard keyboard. Backticks are not always required, but they allow you to use otherwise illegal characters, such as hyphens, in database names.

```
CREATE DATABASE `wordpress-db`;
```

5. Grant full privileges for your database to the WordPress user that you created earlier.

   ```
   GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpress-user"@"localhost";
   ```

6. Flush the database privileges to pick up all of your changes.

   ```
   FLUSH PRIVILEGES;
   ```

7. Exit the `mysql` client.

   ```
   exit
   ```

To create and edit the wp-config.php file

The WordPress installation folder contains a sample configuration file called `wp-config-sample.php`. In this procedure, you copy this file and edit it to fit your specific configuration.

1. Copy the `wp-config-sample.php` file to a file called `wp-config.php`. This creates a new configuration file and keeps the original sample file intact as a backup.

   ```
   [ec2-user ~]$ cp wordpress/wp-config-sample.php wordpress/wp-config.php
   ```

2. Edit the `wp-config.php` file with your favorite text editor (such as **nano** or **vim**) and enter values for your installation. If you do not have a favorite text editor, `nano` is suitable for beginners.

   ```
   [ec2-user ~]$ nano wordpress/wp-config.php
   ```

   a. Find the line that defines `DB_NAME` and change `database_name_here` to the database name that you created in Step 4 of To create a database user and database for your WordPress installation.

      ```
      define('DB_NAME', 'wordpress-db');
      ```

b. Find the line that defines `DB_USER` and change `username_here` to the database user that you created in Step 3 of To create a database user and database for your WordPress installation.

```
define('DB_USER', 'wordpress-user');
```

c. Find the line that defines `DB_PASSWORD` and change `password_here` to the strong password that you created in Step 3 of To create a database user and database for your WordPress installation.

```
define('DB_PASSWORD', 'your_strong_password');
```

d. Find the section called `Authentication Unique Keys and Salts`. These `KEY` and `SALT` values provide a layer of encryption to the browser cookies that WordPress users store on their local machines. Basically, adding long, random values here makes your site more secure. Visit https://api.wordpress.org/secret-key/1.1/salt/ to randomly generate a set of key values that you can copy and paste into your `wp-config.php` file. To paste text into a PuTTY terminal, place the cursor where you want to paste the text and right-click your mouse inside the PuTTY terminal.

For more information about security keys, go to https://wordpress.org/support/article/editing-wp-config-php/#security-keys.

Note

The values below are for example purposes only; do not use these values for your installation.

```
define('AUTH_KEY',          ' #U$$+[RXN8:b^-L 0(WU_+ c+WFkI~c]o]-bHw+)/
Aj[wTwSiZ<Qb[mghEXcRh-');
define('SECURE_AUTH_KEY',  'Zsz._P=l/|y.Lq)XjlkwS1y5NJ76E6EJ.AV0pCKZZB,*~*r ?
6OP$eJT@;+(ndLg');
define('LOGGED_IN_KEY',    'ju}qwre3V*+8f_zOWf?{LlGsQ]Ye@2Jh^,8x>)Y |;(^[Iw]Pi
+LG#A4R?7N`YB3');
define('NONCE_KEY',        'P(g62HeZxEes|LnI^i=H,[XwK9I&[2s|:?0N}VJM%?;v2v]v+;
+^9eXUahg@::Cj');
define('AUTH_SALT',        'C$DpB4Hj[JK:?{ql`sRVa:{:7yShy(9A@5wg+`JJVb1fk%_-
Bx*M4(qc[Qg%JT!h');
```

```
define('SECURE_AUTH_SALT', 'd!uRu#}+q#{f$Z?Z9uFPG.${+S{n~1M&%@~gL>U>NV<zpD-@2-
Es7Q1O-bp28EKv');
define('LOGGED_IN_SALT',   ';j{00P*owZf)kVD+FVLn-~ >.|Y%Ug4#I^*LVd9QeZ^&XmK|
e(76miC+&W&+^0P/');
define('NONCE_SALT',       '-97r*V/cgxLmp?Zy4zUU4r99QQ_rGs2LTd%P;|
_e1tS)8_B/,.6[=UK<J_y9?JWG');
```

e. Save the file and exit your text editor.

To install your WordPress files under the Apache document root

- Now that you've unzipped the installation folder, created a MySQL database and user, and customized the WordPress configuration file, you are ready to copy your installation files to your web server document root so you can run the installation script that completes your installation. The location of these files depends on whether you want your WordPress blog to be available at the actual root of your web server (for example, *my.public.dns.amazonaws.com*) or in a subdirectory or folder under the root (for example, *my.public.dns.amazonaws.com/blog*).
  - If you want WordPress to run at your document root, copy the contents of the wordpress installation directory (but not the directory itself) as follows:

    ```
    [ec2-user ~]$ cp -r wordpress/* /var/www/html/
    ```

  - If you want WordPress to run in an alternative directory under the document root, first create that directory, and then copy the files to it. In this example, WordPress will run from the directory `blog`:

    ```
    [ec2-user ~]$ mkdir /var/www/html/blog
    [ec2-user ~]$ cp -r wordpress/* /var/www/html/blog/
    ```

> ⚠ **Important**
>
> For security purposes, if you are not moving on to the next procedure immediately, stop the Apache web server (`httpd`) now. After you move your installation under the Apache document root, the WordPress installation script is unprotected and an attacker could gain access to your blog if the Apache web server were running. To stop the Apache web

server, enter the command **sudo systemctl stop httpd**. If you are moving on to the next procedure, you do not need to stop the Apache web server.

To allow WordPress to use permalinks

WordPress permalinks need to use Apache `.htaccess` files to work properly, but this is not enabled by default on Amazon Linux. Use this procedure to allow all overrides in the Apache document root.

1. Open the `httpd.conf` file with your favorite text editor (such as **nano** or **vim**). If you do not have a favorite text editor, `nano` is suitable for beginners.

   ```
   [ec2-user ~]$ sudo vim /etc/httpd/conf/httpd.conf
   ```

2. Find the section that starts with `<Directory "/var/www/html">`.

   ```
   <Directory "/var/www/html">
       #
       # Possible values for the Options directive are "None", "All",
       # or any combination of:
       #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
       #
       # Note that "MultiViews" must be named *explicitly* --- "Options All"
       # doesn't give it to you.
       #
       # The Options directive is both complicated and important.  Please see
       # http://httpd.apache.org/docs/2.4/mod/core.html#options
       # for more information.
       #
       Options Indexes FollowSymLinks

       #
       # AllowOverride controls what directives may be placed in .htaccess files.
       # It can be "All", "None", or any combination of the keywords:
       #   Options FileInfo AuthConfig Limit
       #
       AllowOverride None

       #
       # Controls who can get stuff from this server.
       #
   ```

```
    Require all granted
</Directory>
```

3. Change the `AllowOverride None` line in the above section to read `AllowOverride` *`All`*.

> **Note**
>
> There are multiple `AllowOverride` lines in this file; be sure you change the line in the `<Directory "/var/www/html">` section.

```
AllowOverride All
```

4. Save the file and exit your text editor.

To install the PHP graphics drawing library on AL2

The GD library for PHP enables you to modify images. Install this library if you need to crop the header image for your blog. The version of phpMyAdmin that you install might require a specific minimum version of this library (for example, version 7.2).

Use the following command to install the PHP graphics drawing library on AL2. For example, if you installed php7.2 from amazon-linux-extras as part of installing the LAMP stack, this command installs version 7.2 of the PHP graphics drawing library.

```
[ec2-user ~]$ sudo yum install php-gd
```

To verify the installed version, use the following command:

```
[ec2-user ~]$ sudo yum list installed php-gd
```

The following is example output:

```
php-gd.x86_64                        7.2.30-1.amzn2              @amzn2extra-php7.2
```

To fix file permissions for the Apache web server

Some of the available features in WordPress require write access to the Apache document root (such as uploading media though the Administration screens). If you have not already done so, apply the following group memberships and permissions (as described in greater detail in the Tutorial: Install a LAMP server on AL2).

1. Grant file ownership of `/var/www` and its contents to the `apache` user.

   ```
   [ec2-user ~]$ sudo chown -R apache /var/www
   ```

2. Grant group ownership of `/var/www` and its contents to the `apache` group.

   ```
   [ec2-user ~]$ sudo chgrp -R apache /var/www
   ```

3. Change the directory permissions of `/var/www` and its subdirectories to add group write permissions and to set the group ID on future subdirectories.

   ```
   [ec2-user ~]$ sudo chmod 2775 /var/www
   [ec2-user ~]$ find /var/www -type d -exec sudo chmod 2775 {} \;
   ```

4. Recursively change the file permissions of `/var/www` and its subdirectories.

   ```
   [ec2-user ~]$ find /var/www -type f -exec sudo chmod 0644 {} \;
   ```

   > **Note**
   >
   > If you intend to also use WordPress as an FTP server, you'll need more permissive Group settings here. Please review the recommended steps and security settings in WordPress to accomplish this.

5. Restart the Apache web server to pick up the new group and permissions.

   -
     ```
     [ec2-user ~]$ sudo systemctl restart httpd
     ```

Run the WordPress installation script with AL2

You are ready to install WordPress. The commands that you use depend on the operating system. The commands in this procedure are for use with AL2.

1. Use the **systemctl** command to ensure that the `httpd` and database services start at every system boot.

```
[ec2-user ~]$ sudo systemctl enable httpd && sudo systemctl enable mariadb
```

2. Verify that the database server is running.

   ```
   [ec2-user ~]$ sudo systemctl status mariadb
   ```

   If the database service is not running, start it.

   ```
   [ec2-user ~]$ sudo systemctl start mariadb
   ```

3. Verify that your Apache web server (`httpd`) is running.

   ```
   [ec2-user ~]$ sudo systemctl status httpd
   ```

   If the `httpd` service is not running, start it.

   ```
   [ec2-user ~]$ sudo systemctl start httpd
   ```

4. In a web browser, type the URL of your WordPress blog (either the public DNS address for your instance, or that address followed by the `blog` folder). You should see the WordPress installation script. Provide the information required by the WordPress installation. Choose **Install WordPress** to complete the installation. For more information, see Step 5: Run the Install Script on the WordPress website.

Next steps

After you have tested your WordPress blog, consider updating its configuration.

Use a custom domain name

If you have a domain name associated with your EC2 instance's EIP address, you can configure your blog to use that name instead of the EC2 public DNS address. For more information, see Changing The Site URL on the WordPress website.

Configure your blog

You can configure your blog to use different themes and plugins to offer a more personalized experience for your readers. However, sometimes the installation process can backfire, causing you to lose your entire blog. We strongly recommend that you create a backup Amazon Machine Image

(AMI) of your instance before attempting to install any themes or plugins so you can restore your blog if anything goes wrong during installation. For more information, see Create your own AMI.

Increase capacity

If your WordPress blog becomes popular and you need more compute power or storage, consider the following steps:

- Expand the storage space on your instance. For more information, see Amazon EBS Elastic Volumes in the *Amazon EBS User Guide*.
- Move your MySQL database to Amazon RDS to take advantage of the service's ability to scale easily.

Improve network performance of your internet traffic

If you expect your blog to drive traffic from users located around the world, consider Amazon Global Accelerator. Global Accelerator helps you achieve lower latency by improving internet traffic performance between your users' client devices and your WordPress application running on Amazon. Global Accelerator uses the Amazon global network to direct traffic to a healthy application endpoint in the Amazon Region that is closest to the client.

Learn more about WordPress

For information about WordPress, see the WordPress Codex help documentation at http://codex.wordpress.org/.

For more information about troubleshooting your installation, see Common installation problems.

For information about making your WordPress blog more secure, see Hardening WordPress.

For information about keeping your WordPress blog up-to-date, see Updating WordPress.

Help! My public DNS name changed and now my blog is broken

Your WordPress installation is automatically configured using the public DNS address for your EC2 instance. If you stop and restart the instance, the public DNS address changes (unless it is associated with an Elastic IP address) and your blog will not work anymore because it references resources at an address that no longer exists (or is assigned to another EC2 instance). A more detailed description of the problem and several possible solutions are outlined in Changing the Site URL.

If this has happened to your WordPress installation, you might be able to recover your blog with the procedure below, which uses the **wp-cli** command line interface for WordPress.

To change your WordPress site URL with the wp-cli

1. Connect to your EC2 instance with SSH.
2. Note the old site URL and the new site URL for your instance. The old site URL is likely the public DNS name for your EC2 instance when you installed WordPress. The new site URL is the current public DNS name for your EC2 instance. If you are not sure of your old site URL, you can use **curl** to find it with the following command.

   ```
   [ec2-user ~]$ curl localhost | grep wp-content
   ```

   You should see references to your old public DNS name in the output, which will look like this (old site URL in red):

   ```
   <script type='text/javascript' src='http://ec2-52-8-139-223.us-
   west-1.compute.amazonaws.com/wp-content/themes/twentyfifteen/js/functions.js?
   ver=20150330'></script>
   ```

3. Download the **wp-cli** with the following command.

   ```
   [ec2-user ~]$ curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/
   phar/wp-cli.phar
   ```

4. Search and replace the old site URL in your WordPress installation with the following command. Substitute the old and new site URLs for your EC2 instance and the path to your WordPress installation (usually `/var/www/html` or `/var/www/html/blog`).

   ```
   [ec2-user ~]$ php wp-cli.phar search-replace 'old_site_url' 'new_site_url' --path=/
   path/to/wordpress/installation --skip-columns=guid
   ```

5. In a web browser, enter the new site URL of your WordPress blog to verify that the site is working properly again. If it is not, see Changing the Site URL and Common installation problems for more information.

Using Amazon Linux 2 outside of Amazon EC2

The AL2 container images can be run in compatible container runtime environments.

AL2 can also be run as a virtualized guest outside of directly being run on Amazon EC2.

> **Note**
>
> The configuration of AL2 images differs from AL2023.
> When migrating to AL2023, ensure that you review Using Amazon Linux 2023 outside of Amazon EC2 and adapt your configuration to be compatible with AL2023.

Run AL2 as a virtual machine on premises

Use the AL2 virtual machine (VM) images for on-premises development and testing. We offer a different AL2 VM image for each of the supported virtualization platforms. You can view the list of supported platforms on the Amazon Linux 2 virtual machine images page.

To use the AL2 virtual machine images with one of the supported virtualization platforms, do the following:

- Step 1: Prepare the seed.iso boot image
- Step 2: Download the AL2 VM image
- Step 3: Boot and connect to your new VM

Step 1: Prepare the `seed.iso` boot image

The `seed.iso` boot image includes the initial configuration information that is needed to boot your new VM, such as the network configuration, host name, and user data.

> **Note**
>
> The `seed.iso` boot image includes only the configuration information required to boot the VM. It does not include the AL2 operating system files.

To generate the `seed.iso` boot image, you need two configuration files:

- `meta-data` – This file includes the hostname and static network settings for the VM.
- `user-data` – This file configures user accounts, and specifies their passwords, key pairs, and access mechanisms. By default, the AL2 VM image creates an `ec2-user` user account. You use the `user-data` configuration file to set the password for the default user account.

To create the `seed.iso` boot disc

1. Create a new folder named `seedconfig` and navigate into it.
2. Create the `meta-data` configuration file.

   a. Create a new file named `meta-data`.

   b. Open the `meta-data` file using your preferred editor and add the following.

   ```
   local-hostname: vm_hostname
   # eth0 is the default network interface enabled in the image. You can configure
    static network settings with an entry like the following.
   network-interfaces: |
     auto eth0
     iface eth0 inet static
     address 192.168.1.10
     network 192.168.1.0
     netmask 255.255.255.0
     broadcast 192.168.1.255
     gateway 192.168.1.254
   ```

   Replace *`vm_hostname`* with a VM host name of your choice, and configure the network settings as required.

   c. Save and close the `meta-data` configuration file.

   For an example `meta-data` configuration file that specifies a VM hostname (`amazonlinux.onprem`), configures the default network interface (`eth0`), and specifies static IP addresses for the necessary network devices, see the sample Seed.iso file.
3. Create the `user-data` configuration file.

   a. Create a new file named `user-data`.

   b. Open the `user-data` file using your preferred editor and add the following.

```
#cloud-config
#vim:syntax=yaml
users:
# A user by the name `ec2-user` is created in the image by default.
  - default
chpasswd:
  list: |
    ec2-user:plain_text_password
# In the above line, do not add any spaces after 'ec2-user:'.
```

Replace *`plain_text_password`* with a password of your choice for the default `ec2-user` user account.

c. (Optional) By default, cloud-init applies network settings each time the VM boots. Add the following to prevent cloud-init from applying network settings at each boot, and to retain the network settings applied during the first boot.

```
# NOTE: Cloud-init applies network settings on every boot by default. To retain
 network settings
# from first boot, add the following ‘write_files’ section:
write_files:
  - path: /etc/cloud/cloud.cfg.d/80_disable_network_after_firstboot.cfg
    content: |
      # Disable network configuration after first boot
      network:
        config: disabled
```

d. Save and close the `user-data` configuration file.

You can also create additional user accounts and specify their access mechanisms, passwords, and key pairs. For more information about the supported directives, see Module reference. For an example `user-data` file that creates three additional users and specifies a custom password for the default `ec2-user` user account, see the sample Seed.iso file.

4. Create the `seed.iso` boot image using the `meta-data` and `user-data` configuration files.

   For Linux, use a tool such as **genisoimage**. Navigate into the `seedconfig` folder, and run the following command.

```
$ genisoimage -output seed.iso -volid cidata -joliet -rock user-data meta-data
```

For macOS, use a tool such as **hdiutil**. Navigate one level up from the `seedconfig` folder, and run the following command.

```
$ hdiutil makehybrid -o seed.iso -hfs -joliet -iso -default-volume-name cidata
 seedconfig/
```

Step 2: Download the AL2 VM image

We offer a different AL2 VM image for each of the supported virtualization platforms. You can view the list of supported platforms and download the correct VM image for your chosen platform from the Amazon Linux 2 virtual machine images page.

Step 3: Boot and connect to your new VM

To boot and connect to your new VM, you must have the `seed.iso` boot image (created in Step 1) and an AL2 VM image (downloaded in Step 2). The steps vary depending on your chosen VM platform.

VMware vSphere

The VM image for VMware is made available in the OVF format.

To boot the VM using VMware vSphere

1. Create a new datastore for the `seed.iso` file, or add it to an existing datastore.
2. Deploy the OVF template, but do not start the VM yet.
3. In the **Navigator** panel, right-click the new virtual machine and choose **Edit Settings**.
4. On the **Virtual Hardware** tab, for **New device**, choose **CD/DVD Drive**, and then choose **Add**.
5. For **New CD/DVD Drive**, choose **Datastore ISO File**. Select the datastore to which you added the `seed.iso` file, browse to and select the `seed.iso` file, and then choose **OK**.
6. For **New CD/DVD Drive**, select **Connect**, and then choose **OK**.

After you have associated the datastore with the VM, you should be able to boot it.

KVM

To boot the VM using KVM

1. Open the **Create new VM** wizard.
2. For Step 1, choose **Import existing disk image**.
3. For Step 2, browse to and select the VM image. For **OS type** and **Version**, choose **Linux** and **Red Hat Enterprise Linux 7.0** respectively.
4. For Step 3, specify the amount of RAM and the number of CPUs to use.
5. For Step 4, enter a name for the new VM and select **Customize configuration before install**, and choose **Finish**.
6. In the Configuration window for the VM, choose **Add Hardware**.
7. In the **Add New Virtual Hardware** window, choose **Storage**.
8. In the Storage configuration, choose **Select or create custom storage**. For **Device type**, choose **CDROM device**. Choose **Manage**, **Browse Local**, and then navigate to and select the `seed.iso` file. Choose **Finish**.
9. Choose **Begin Installation**.

Oracle VirtualBox

To boot the VM using Oracle VirtualBox

1. Open Oracle VirtualBox and choose **New**.
2. For **Name**, enter a descriptive name for the virtual machine, and for **Type** and **Version**, select **Linux** and **Red Hat (64-bit)** respectively. Choose **Continue**.
3. For **Memory size**, specify the amount of memory to allocate to the virtual machine, and then choose **Continue**.
4. For **Hard disk**, choose **Use an existing virtual hard disk file**, browse to and open the VM image, and then choose **Create**.
5. Before you start the VM, you must load the `seed.iso` file in the virtual machine's virtual optical drive:

   a. Select the new VM, choose **Settings**, and then choose **Storage**.

   b. In the **Storage Devices** list, under **Controller: IDE**, choose the *Empty* optical drive.

c. In the **Attributes** section for the optical drive, choose the browse button, select **Choose Virtual Optical Disk File**, and then select the `seed.iso` file. Choose **OK** to apply the changes and close the Settings.

After you have added the `seed.iso` file to the virtual optical drive, you should be able to start the VM.

Microsoft Hyper-V

The VM image for Microsoft Hyper-V is compressed into a zip file. You must extract the contents of the zip file.

To boot the VM using Microsoft Hyper-V

1. Open the **New Virtual Machine Wizard**.
2. When prompted to select a generation, select **Generation 1**.
3. When prompted to configure the network adapter, for **Connection** choose **External**.
4. When prompted to connect a virtual hard disk, choose **Use an existing virtual hard disk**, choose **Browse**, and then navigate to and select the VM image. Choose **Finish** to create the VM.
5. Right-click the new VM and choose **Settings**. In the **Settings** window, under **IDE Controller 1**, choose **DVD Drive**.
6. For the DVD drive, choose **Image file** and then browse to and select the `seed.iso` file.
7. Apply the changes and start the VM.

After the VM has booted, log in using one of the user accounts that is defined in the `user-data` configuration file. After you have logged in for the first time, you can then disconnect the `seed.iso` boot image from the VM.

Identify AL2 instances

The following information describes how to identify an AL2 instance from another Amazon Linux version or other Linux distribution.

Identify Amazon Linux images

Each image contains a unique `/etc/image-id` file that identifies it. This file contains the following information about the image:

- `image_name`, `image_version`, `image_arch` – Values from the build recipe that Amazon used to construct the image.
- `image_stamp` – A unique, random hex value generated during image creation.
- `image_date` – The UTC time of image creation, in *YYYYMMDDhhmmss* format.
- `recipe_name`, `recipe_id` – The name and ID of the build recipe Amazon used to construct the image.

Amazon Linux contains an `/etc/system-release` file that specifies the current release that is installed. This file is updated using **yum** and is part of the `system-release` RPM Package Manager (RPM).

Amazon Linux also contains a machine-readable version of `/etc/system-release` that follows the Common Platform Enumeration (CPE) specification; see `/etc/system-release-cpe`.

AL2

The following is an example of `/etc/image-id` for the current version of AL2.

```
[ec2-user ~]$ cat /etc/image-id
      image_name="amzn2-ami-hvm"
image_version="2"
image_arch="x86_64"
image_file="amzn2-ami-hvm-2.0.20180810-x86_64.xfs.gpt"
image_stamp="8008-2abd"
image_date="20180811020321"
recipe_name="amzn2 ami"
recipe_id="c652686a-2415-9819-65fb-4dee-9792-289d-1e2846bd"
```

The following is an example of `/etc/system-release` for the current version of AL2.

```
[ec2-user ~]$ cat /etc/system-release
      AL2
```

The following is an example of `/etc/os-release` for AL2.

```
[ec2-user ~]$ cat /etc/os-release
NAME="Amazon Linux"
VERSION="2"
ID="amzn"
ID_LIKE="centos rhel fedora"
VERSION_ID="2"
PRETTY_NAME="Amazon Linux 2"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2"
HOME_URL="https://amazonlinux.com/"
```

Amazon Linux AMI

The following is an example of `/etc/image-id` for the current Amazon Linux AMI.

```
[ec2-user ~]$ cat /etc/image-id
      image_name="amzn-ami-hvm"
image_version="2018.03"
image_arch="x86_64"
image_file="amzn-ami-hvm-2018.03.0.20180811-x86_64.ext4.gpt"
image_stamp="cc81-f2f3"
image_date="20180811012746"
recipe_name="amzn ami"
recipe_id="5b283820-dc60-a7ea-d436-39fa-439f-02ea-5c802dbd"
```

The following is an example of `/etc/system-release` for the current Amazon Linux AMI.

```
[ec2-user ~]$ cat /etc/system-release
      Amazon Linux AMI release 2018.03
```

Amazon integration in AL2

Amazon command line tools

The Amazon Command Line Interface (Amazon CLI) is an open source tool that provides a consistent interface to interact with Amazon Web Services services using commands in your command-line shell. For more information, see What is the Amazon Command Line Interface? in the *Amazon Command Line Interface User Guide*.

AL2 and AL1 have version 1 of the Amazon CLI preinstalled. The current release of Amazon Linux, AL2023, has version 2 of the Amazon CLI preinstalled. For more information about using the Amazon CLI on AL2023, see Get started with AL2023 in the *Amazon Linux 2023 User Guide*.

Getting started with programming runtimes

AL2 provides different versions of certain language runtimes. We work with upstream projects, such as PHP, that support multiple versions at the same time. To find information about how to install and manage these name-versioned packages, use the `yum` command to search and install these packages. For more information, see Package repository.

The following topics describe how each language runtime functions in AL2.

Topics

- C, C++, and Fortran in AL2
- Go in AL2
- Java in AL2
- Perl in AL2
- PHP in AL2
- Python in AL2
- Rust in AL2

C, C++, and Fortran in AL2

AL2 includes both the GNU Compiler Collection (GCC) and the Clang frontend for LLVM.

The major version of GCC will remain constant throughout the lifetime of AL2. Bug and security fixes might be backported to the major version of GCC that ships in AL2.

By default, AL2 includes version 7.3 of GCC which builds almost all packages. The `gcc10` package makes GCC 10 available to a limited extent, but we don't recommend using GCC 10 to build packages.

The default compiler flags that build AL2 RPMs include some optimization and hardening flags. We recommend that you include some optimization and hardening flags if you are building your own code with GCC.

The default compiler and optimization flags in AL2023 improve upon what is present in AL2.

Go in AL2

You might want to build your own code written in Go on Amazon Linux using a toolchain provided with AL2.

The Go toolchain will be updated throughout the life of AL2. This might be in response to any CVE in the toolchain we ship, or as a prerequisite of addressing a CVE in another package.

Go is a relatively fast moving programming language. There might be a situation where existing applications written in Go have to adapt to new versions of the Go toolchain. For more information about Go, see Go 1 and the Future of Go Programs.

Although AL2 will incorporate new versions of the Go toolchain during its life, this will not be in lockstep with the upstream Go releases. Therefore, using the Go toolchain provided in AL2 might not be suitable if you want to build Go code using cutting-edge features of the Go language and standard library.

During the lifetime of AL2, earlier package versions are not removed from the repositories. If an earlier Go toolchain is required, you can choose to forgo bug and security fixes of newer Go toolchains and install an earlier version from the repositories using the same mechanisms available for any RPM.

If you want to build your own Go code on AL2 you can use the Go toolchain included in AL2 with the knowledge that this toolchain might move forward through the lifetime of AL2.

Java in AL2

AL2 provides several versions of Amazon Corretto to support Java based workloads, as well as some OpenJDK versions. We recommend that you migrate to Amazon Corretto in preparation for migrating to AL2023.

Corretto is a build of the Open Java Development Kit (OpenJDK) with long-term support from Amazon. Corretto is certified using the Java Technical Compatibility Kit (TCK) to ensure it meets the Java SE standard and is available on Linux, Windows, and macOS.

An Amazon Corretto package is available for each of Corretto 1.8.0, Corretto 11, and Corretto 17.

Each Corretto version in AL2 is supported for the same period of time as the Corretto version is, or until the end of life of AL2, whichever is sooner. For more information, see the Amazon Corretto FAQs.

Perl in AL2

AL2 provides version 5.16 of the Perl programming language.

Perl modules in AL2

Various Perl modules are packaged as RPMs in AL2. Although there are many Perl modules available as RPMs, Amazon Linux does not try to package every possible Perl module. Modules packaged as RPMs might be relied upon by other operating system RPM packages, so Amazon Linux will prioritize ensuring they are security patched over pure feature updates.

AL2 also includes CPAN so that Perl developers can use the idiomatic package manager for Perl modules.

PHP in AL2

AL2 currently provides two fully supported versions of the PHP programming language as part of AL2 Extras Library. Each PHP version is supported for the same time frame as upstream PHP as listed under deprecated date in List of Amazon Linux 2 Extras.

For information about how to use AL2 Extras to install application and software updates on your instances, see AL2 Extras Library.

To assist migration to AL2023, both PHP 8.1 and 8.2 are available on AL2 and AL2023.

> **Note**
>
> AL2 includes PHP 7.1, 7.2, 7.3, and 7.4 in `amazon-linux-extras`. All of these Extras are EOL and are not guaranteed to get any additional security updates.
> To find out when each version of PHP is deprecated in AL2, see the List of Amazon Linux 2 Extras.

Migrating from earlier PHP 8.x versions

The upstream PHP community put together comprehensive migration documentation for moving to PHP 8.2 from PHP 8.1. Documentation also exists for migrating from PHP 8.0 to 8.1.

AL2 includes PHP 8.0, 8.1, and 8.2 in `amazon-linux-extras` that enables an efficient upgrade path to AL2023. To find out when each version of PHP is deprecated in AL2, see the List of Amazon Linux 2 Extras.

Migrating from PHP 7.x versions

The upstream PHP community put together comprehensive migration documentation for moving to PHP 8.0 from PHP 7.4. Combined with the documentation referenced in the previous section on migrating to PHP 8.1, and PHP 8.2, you have all of the steps needed to migrate your PHP based application to modern PHP.

The PHP project maintains a list and schedule of supported versions, along with a list of unsupported branches.

> **Note**
>
> When AL2023 was released, all 7.x and 5.x versions of PHP were not supported by the PHP community, and were not included as options in AL2023.

Python in AL2

AL2 provides support and security patches for Python 2.7 until June 2026, as part of our long-term support commitment for AL2 core packages. This support extends beyond the upstream Python community declaration of Python 2.7 EOL of January 2020.

> **Note**
>
> AL2023 completely removed Python 2.7. Any components requiring Python are now written to work with Python 3.

AL2 uses the `yum` package manager that has a hard dependency on Python 2.7. In AL2023, the `dnf` package manager has migrated to Python 3, and no longer requires Python 2.7. AL2023 has completely moved to Python 3. We recommend that you complete your migration to Python 3.

Rust in AL2

You might want to build your own code written in Rust on AL2 using a toolchain provided with AL2.

The Rust toolchain will be updated throughout the life of AL2. This might be in response to a CVE in the toolchain we ship, or as prerequisite for a CVE update in another package.

Rust is a relatively fast moving language, with new releases at approximately a six-week cadence. The new releases might add new language or standard library features. Although AL2 will incorporate new versions of the Rust toolchain during its life, this will not be in lockstep with the upstream Rust releases. Therefore, using the Rust toolchain provided in AL2 might not be suitable if you want to build Rust code using cutting-edge features of the Rust language.

During the lifetime of AL2, previous package versions are not removed from the repositories. If a previous Rust toolchain is required, you can choose to forgo bug and security fixes of newer Rust toolchains and install a previous version from the repositories using the same processes available for any RPM.

To build your own Rust code on AL2, use the Rust toolchain included in AL2 with the knowledge that this toolchain might move forward throughout the lifetime of AL2.

AL2 kernel

AL2 originally shipped with a 4.14 kernel, with version 5.10 as the current default. If you are still using a 4.14 kernel, you are encouraged to migrate to the 5.10 kernel.

Kernel live patching is supported on AL2.

Topics

- AL2 supported kernels
- Kernel Live Patching on AL2

AL2 supported kernels

Supported kernel versions

Currently, AL2 AMIs are available with kernel versions 4.14 and 5.10, with version 5.10 as the default. We recommend that you use an AL2 AMI with kernel 5.10.

AL2023 AMIs are available with kernel version 6.1. For more information, see AL2023 Kernel changes from AL2 in the *Amazon Linux 2023 User Guide*.

Support Timeframe

The 5.10 kernel available on AL2 will be supported until the AL2 AMI reaches the end of standard support.

Live patching support

AL2 kernel version	Kernel live patching supported
4.14	Yes
5.10	Yes
5.15	No

Kernel Live Patching on AL2

Kernel Live Patching for AL2 allows you to apply specific security vulnerability and critical bug patches to a running Linux kernel, without reboots or disruptions to running applications. This allows you to benefit from improved service and application availability, while applying these fixes until the system can be rebooted.

For information about Kernel Live Patching for AL2023, see Kernel Live Patching on AL2023 in the *Amazon Linux 2023 User Guide*.

Amazon releases two types of kernel live patches for AL2:

- **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 are not associated with CVEs.

Amazon provides kernel live patches for an AL2 kernel version for up to 3 months after its release. After the 3-month period, you must update to a later kernel version to continue to receive kernel live patches.

AL2 kernel live patches are made available as signed RPM packages in the existing AL2 repositories. The patches can be installed on individual instances using existing **yum** workflows, or they can be installed on a group of managed instances using Amazon Systems Manager.

Kernel Live Patching on AL2 is provided at no additional cost.

Topics

- Supported configurations and prerequisites
- Work with Kernel Live Patching
- Limitations
- Frequently asked questions

Supported configurations and prerequisites

Kernel Live Patching is supported on Amazon EC2 instances and on-premises virtual machines running AL2.

To use Kernel Live Patching on AL2, you must use:

- Kernel version `4.14` or `5.10` on the `x86_64` architecture
- Kernel version `5.10` on the `ARM64` architecture

Policy Requirements

To download packages from Amazon Linux repositories, Amazon EC2 needs access to service-owned Amazon S3 buckets. If you are using a Amazon Virtual Private Cloud (VPC) endpoint for Amazon S3 in your environment, you need to ensure that your VPC endpoint policy allows access to those public buckets.

The table describes each of the Amazon S3 buckets that EC2 might need to access for Kernel Live Patching.

S3 bucket ARN	Description
arn:aws:s3:::packages.*region*.amazonaws.com/*	Amazon S3 bucket containing Amazon Linux AMI packages
arn:aws:s3:::repo.*region*.amazonaws.com/*	Amazon S3 bucket containing Amazon Linux AMI repositories
arn:aws:s3:::amazonlinux.*region*.amazonaws.com/*	Amazon S3 bucket containing AL2 repositories
arn:aws:s3:::amazonlinux-2-repos-*region*/*	Amazon S3 bucket containing AL2 repositories

The following policy illustrates how to restrict access to identities and resources that belong to your organization and provide access to the Amazon S3 buckets required for Kernel Live Patching. Replace *region*, *principal-org-id* and *resource-org-id* with your organization's values.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowRequestsByOrgsIdentitiesToOrgsResources",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "principal-org-id",
          "aws:ResourceOrgID": "resource-org-id"
        }
      }
    },
    {
      "Sid": "AllowAccessToAmazonLinuxAMIRepositories",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::packages.region.amazonaws.com/*",
        "arn:aws:s3:::repo.region.amazonaws.com/*",
        "arn:aws:s3:::amazonlinux.region.amazonaws.com/*",
        "arn:aws:s3:::amazonlinux-2-repos-region/*"
      ]
    }
  ]
}
```

Work with Kernel Live Patching

You can enable and use Kernel Live Patching on individual instances using the command line on the instance itself, or 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 AL2 instances in the *Amazon Systems Manager User Guide*.

Topics

- Enable Kernel Live Patching
- View the available kernel live patches
- Apply kernel live patches
- View the applied kernel live patches
- Disable Kernel Live Patching

Enable Kernel Live Patching

Kernel Live Patching is disabled by default on AL2. To use live patching, you must install the **yum** plugin for Kernel Live Patching and enable the live patching functionality.

Prerequisites

Kernel Live Patching requires `binutils`. If you do not have `binutils` installed, install it using the following command:

```
$ sudo yum install binutils
```

To enable Kernel Live Patching

1. Kernel live patches are available for the following AL2 kernel versions:

   - Kernel version `4.14` or `5.10` on the `x86_64` architecture
   - Kernel version `5.10` on the `ARM64` architecture

To check your kernel version, run the following command.

```
$ sudo yum list kernel
```

2. If you already have a supported kernel version, skip this step. If you do not have a supported kernel version, run the following commands to update the kernel to the latest version and to reboot the instance.

```
$ sudo yum install -y kernel
```

```
$ sudo reboot
```

3. Install the **yum** plugin for Kernel Live Patching.

```
$ sudo yum install -y yum-plugin-kernel-livepatch
```

4. Enable the **yum** plugin for Kernel Live Patching.

```
$ sudo yum kernel-livepatch enable -y
```

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

5. To confirm that the **yum** plugin for kernel live patching has installed successfully, run the following command.

```
$ rpm -qa | grep kernel-livepatch
```

   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. The following is example output.

```
yum-plugin-kernel-livepatch-1.0-0.11.amzn2.noarch
kernel-livepatch-5.10.102-99.473-1.0-0.amzn2.x86_64
```

6. Install the **kpatch** package.

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

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

   ```
   $ sudo yum update kpatch-runtime
   ```

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

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

9. Enable the Kernel Live Patching topic in the AL2 Extras Library. This topic contains the kernel live patches.

   ```
   $ sudo amazon-linux-extras enable livepatch
   ```

View the available kernel live patches

Amazon Linux security alerts are published to the Amazon Linux Security Center. For more information about the AL2 security alerts, which include 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.

```
$ yum updateinfo list
```

The following shows example output.

```
Loaded plugins: extras_suggestions, kernel-livepatch, langpacks, priorities, update-
motd
ALAS2LIVEPATCH-2020-002 important/Sec. kernel-
livepatch-5.10.102-99.473-1.0-3.amzn2.x86_64
ALAS2LIVEPATCH-2020-005 medium/Sec. kernel-livepatch-5.10.102-99.473-1.0-4.amzn2.x86_64
updateinfo list done
```

To list all available kernel live patches for CVEs

Use the following command.

```
$ yum updateinfo list cves
```

The following shows example output.

```
Loaded plugins: extras_suggestions, kernel-livepatch, langpacks, priorities, update-
motdamzn2-core/2/x86_64 | 2.4 kB 00:00:00
CVE-2019-15918 important/Sec. kernel-livepatch-5.10.102-99.473-1.0-3.amzn2.x86_64
CVE-2019-20096 important/Sec. kernel-livepatch-5.10.102-99.473-1.0-3.amzn2.x86_64
CVE-2020-8648 medium/Sec. kernel-livepatch-5.10.102-99.473-1.0-4.amzn2.x86_64
updateinfo list done
```

Apply kernel live patches

You apply kernel live patches using the **yum** package manager in the same way that you would apply regular updates. The **yum** plugin for Kernel Live Patching manages the kernel live patches that are available to be applied.

> **Tip**
>
> We recommend that you update your kernel regularly using Kernel Live Patching to ensure that it receives specific important and critical security fixes until the system can be rebooted. Please also check if additional fixes have been made available to the native kernel package that cannot be deployed as live patches and update and reboot into the kernel update for those cases.

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.
2. Apply the kernel live patch for your AL2 kernel.

   ```
   $ sudo yum install kernel-livepatch-kernel_version.x86_64
   ```

For example, the following command applies a kernel live patch for AL2 kernel version `5.10.102-99.473`.

```
$ sudo yum install kernel-livepatch-5.10.102-99.473-1.0-4.amzn2.x86_64
```

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

Use the following command.

```
$ sudo yum update --security
```

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

Important

- The kernel version is not updated after applying kernel live patches. The version is only updated to the new version after the instance is rebooted.
- An AL2 kernel receives kernel live patches for a period of three months. After the three month period has lapsed, no new kernel live patches are released for that kernel version. To continue to receive kernel live patches after the three-month period, you must reboot the instance to move to the new kernel version, which will then continue receiving kernel live patches for the next three months. To check the support window for your kernel version, run `yum kernel-livepatch supported`.

View the applied kernel live patches

To view the applied kernel live patches

Use the following command.

```
$ kpatch list
```

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

```
Loaded patch modules:
```

```
livepatch_cifs_lease_buffer_len [enabled]
livepatch_CVE_2019_20096 [enabled]
livepatch_CVE_2020_8648 [enabled]

Installed patch modules:
livepatch_cifs_lease_buffer_len (5.10.102-99.473.amzn2.x86_64)
livepatch_CVE_2019_20096 (5.10.102-99.473.amzn2.x86_64)
livepatch_CVE_2020_8648 (5.10.102-99.473.amzn2.x86_64)
```

> **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.

To disable Kernel Live Patching

1. Remove the RPM packages for the applied kernel live patches.

   ```
   $ sudo yum kernel-livepatch disable
   ```

2. Uninstall the **yum** plugin for Kernel Live Patching.

   ```
   $ sudo yum remove yum-plugin-kernel-livepatch
   ```

3. Reboot the instance.

   ```
   $ sudo reboot
   ```

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.

- **Note**

  Due to technical limitations, some issues cannot be addressed with live patching. Because of that, these fixes will not be shipped in the kernel live patch package but only in the native kernel package update. You can install the native kernel package update and reboot the system to activate the patches as usual.

Frequently asked questions

For frequently asked questions about Kernel Live Patching for AL2, see the Amazon Linux 2 Kernel Live Patching FAQ.

AL2 Extras Library

Warning

The epel Extra enables the third party EPEL7 repository. As of 2024-06-30 the third-party EPEL7 repository is *no longer being maintained*.
This third-party repository will have *no future updates*. This means there will be *no security fixes* for packages in the *EPEL* repository.
See the EPEL section of the Amazon Linux 2023 User Guide for options for some EPEL packages.

With AL2, you can use the Extras Library to install application and software updates on your instances. These software updates are known as *topics*. You can install a specific version of a topic or omit the version information to use the most recent version. Extras help alleviate having to compromise between the stability of an operating system and the freshness of available software.

The contents of Extras topics are exempt from the Amazon Linux policy on long-term support and binary compatibility. Extras topics provide access to a curated list of packages. The versions of the packages might be updated frequently or might not be supported for the same amount of time as AL2.

Note

Individual Extras topics might be deprecated before AL2 reaches EOL.

To list the available topics, use the following command.

```
[ec2-user ~]$ amazon-linux-extras list
```

To enable a topic and install the latest version of its package to ensure freshness, use the following command.

```
[ec2-user ~]$ sudo amazon-linux-extras install topic
```

To enable topics and install specific versions of their packages to ensure stability, use the following command.

```
[ec2-user ~]$ sudo amazon-linux-extras install topic=version topic=version
```

To remove a package installed from a topic, use the following command.

```
[ec2-user ~]$ sudo yum remove $(yum list installed | grep amzn2extra-topic | awk
 '{ print $1 }')
```

Note

This command does not remove packages that were installed as dependencies of the Extra.

To disable a topic and make the packages inaccessible to the yum package manager, use the following command.

```
[ec2-user ~]$ sudo amazon-linux-extras disable topic
```

Important

This command is intended for advanced users. Improper usage of this command could cause package compatibility conflicts.

List of Amazon Linux 2 Extras

Extra name	Deprecated date
BCC	
GraphicsMagick1.3	
R3.4	
R4	
ansible2	2023-09-30
aws-nitro-enclaves-cli	

Extra name	Deprecated date
awscli1	
collectd	
collectd-python3	
corretto8	2025-06-30
dnsmasq	
dnsmasq2.85	
docker	2025-06-30
ecs	
emacs	2018-11-14
epel	2024-06-30
firecracker	2022-11-08
firefox	
gimp	2018-11-14
golang1.11	2023-08-01
golang1.19	2023-09-30
golang1.9	2018-12-14
haproxy2	
httpd_modules	
java-openjdk11	2024-09-30
kernel-5.10	

Extra name	Deprecated date
kernel-5.15	
kernel-5.4	
kernel-ng	2022-08-08
lamp-mariadb10.2-php7.2	2020-11-30
libreoffice	
livepatch	
lustre	
lustre2.10	
lynis	
mariadb10.5	2025-06-24
mate-desktop1.x	
memcached1.5	
mock	
mock2	
mono	
nano	2018-11-14
nginx1	
nginx1.12	2019-09-20
nginx1.22.1	
php7.1	2020-01-15

Extra name	Deprecated date
php7.2	2020-11-30
php7.3	2021-12-06
php7.4	2022-11-03
php8.0	2023-11-26
php8.1	2025-12-31
php8.2	2026-12-31
postgresql10	2023-09-30
postgresql11	2023-11-09
postgresql12	2024-11-14
postgresql13	2025-06-30
postgresql14	2025-06-30
postgresql9.6	2022-08-09
python3	2018-08-22
python3.8	2024-10-14
redis4.0	2021-05-25
redis6	
ruby2.4	2020-08-27
ruby2.6	2023-03-31
ruby3.0	2024-03-31
rust1	

Extra name	Deprecated date
selinux-ng	
squid4	2023-09-30
testing	
tomcat8.5	2024-03-31
tomcat9	
unbound1.13	
unbound1.17	
vim	2018-11-14

AL2 Reserved Users and Groups

AL2 pre-allocates certain users and groups during both the provisioning of the image and during the installation of certain packages. The users, groups, and their associated UIDs and GIDs are listed here to prevent conflicts.

Topics

- List of Amazon Linux 2 Reserved Users
- List of Amazon Linux 2 Reserved Groups

List of Amazon Linux 2 Reserved Users

Listed by UID

User name	UID
root	0
bin	1
daemon	2
adm	3
lp	4
sync	5
shutdown	6
halt	7
mail	8
uucp	10
operator	11
games	12

User name	UID
ftp	14
oprofile	16
pkiuser	17
squid	23
named	25
postgres	26
mysql	27
nscd	28
nscd	28
rpcuser	29
rpc	32
amandabackup	33
ntp	38
mailman	41
gdm	42
mailnull	47
apache	48
smmsp	51
tomcat	53
ldap	55

User name	UID
tss	59
nslcd	65
pegasus	66
avahi	70
tcpdump	72
sshd	74
radvd	75
cyrus	76
arpwatch	77
fax	78
dbus	81
postfix	89
quagga	92
radiusd	95
radiusd	95
hsqldb	96
dovecot	97
ident	98
nobody	99
qemu	107

User name	UID
usbmuxd	113
stap-server	155
avahi-autoipd	170
pulse	171
rtkit	172
dhcpd	177
sanlock	179
haproxy	188
hacluster	189
systemd-journal-gateway	191
systemd-network	192
systemd-resolve	193
uuidd	357
tang	358
stapdev	359
stapsys	360
stapusr	361
systemd-journal-upload	362
systemd-journal-remote	363
saned	364

User name	UID
pesign	365
pcpqa	366
pcp	367
memcached	368
ipsilon	369
ipaapi	370
kdcproxy	371
ods	372
sssd	373
gluster	374
fedfs	375
dovenull	376
coroqnetd	377
clevis	378
clamscan	379
clamilt	380
clamupdate	381
colord	382
geoclue	383
aws-kinesis-agent-user	384

User name	UID
cwagent	385
unbound	386
polkitd	387
saslauth	388
dirsrv	389
chrony	996
ec2-instance-connect	997
rngd	998
libstoragemgmt	999
ec2-user	1000
nfsnobody	65534

Listed by Name

User name	UID
adm	3
amandabackup	33
apache	48
arpwatch	77
avahi	70
avahi-autoipd	170

User name	UID
aws-kinesis-agent-user	384
bin	1
chrony	996
clamilt	380
clamscan	379
clamupdate	381
clevis	378
colord	382
coroqnetd	377
cwagent	385
cyrus	76
daemon	2
dbus	81
dhcpd	177
dirsrv	389
dovecot	97
dovenull	376
ec2-instance-connect	997
ec2-user	1000
fax	78

User name	UID
fedfs	375
ftp	14
games	12
gdm	42
geoclue	383
gluster	374
hacluster	189
halt	7
haproxy	188
hsqldb	96
ident	98
ipaapi	370
ipsilon	369
kdcproxy	371
ldap	55
libstoragemgmt	999
lp	4
mail	8
mailman	41
mailnull	47

User name	UID
memcached	368
mysql	27
named	25
nfsnobody	65534
nobody	99
nscd	28
nscd	28
nslcd	65
ntp	38
ods	372
operator	11
oprofile	16
pcp	367
pcpqa	366
pegasus	66
pesign	365
pkiuser	17
polkitd	387
postfix	89
postgres	26

User name	UID
pulse	171
qemu	107
quagga	92
radiusd	95
radiusd	95
radvd	75
rngd	998
root	0
rpc	32
rpcuser	29
rtkit	172
saned	364
sanlock	179
saslauth	388
shutdown	6
smmsp	51
squid	23
sshd	74
sssd	373
stap-server	155

User name	UID
stapdev	359
stapsys	360
stapusr	361
sync	5
systemd-journal-gateway	191
systemd-journal-remote	363
systemd-journal-upload	362
systemd-network	192
systemd-resolve	193
tang	358
tcpdump	72
tomcat	53
tss	59
unbound	386
usbmuxd	113
uucp	10
uuidd	357

List of Amazon Linux 2 Reserved Groups

Listed by GID

Group name	GID
root	0
bin	1
daemon	2
sys	3
adm	4
tty	5
disk	6
disk	6
lp	7
mem	8
kmem	9
wheel	10
cdrom	11
mail	12
uucp	14
man	15
oprofile	16
pkiuser	17
dialout	18
floppy	19

Group name	GID
games	20
slocate	21
utmp	22
squid	23
named	25
postgres	26
mysql	27
nscd	28
nscd	28
rpcuser	29
rpc	32
tape	33
tape	33
utempter	35
kvm	36
ntp	38
video	39
dip	40
mailman	41
gdm	42

Group name	GID
mailnull	47
apache	48
ftp	50
smmsp	51
tomcat	53
lock	54
ldap	55
tss	59
audio	63
pegasus	65
avahi	70
tcpdump	72
sshd	74
radvd	75
saslauth	76
saslauth	76
arpwatch	77
fax	78
dbus	81
screen	84

Group name	GID
quaggavt	85
wbpriv	88
wbpriv	88
postfix	89
postdrop	90
quagga	92
radiusd	95
radiusd	95
hsqldb	96
dovecot	97
ident	98
nobody	99
users	100
qemu	107
usbmuxd	113
stap-server	155
stapusr	156
stapusr	156
stapsys	157
stapdev	158

Group name	GID
avahi-autoipd	170
pulse	171
rtkit	172
dhcpd	177
sanlock	179
haproxy	188
haclient	189
systemd-journal	190
systemd-journal	190
systemd-journal-gateway	191
systemd-network	192
systemd-resolve	193
usbmon	351
wireshark	352
uuidd	353
tang	354
systemd-journal-upload	355
sfcb	356
systemd-journal-remote	356
saned	357

Group name	GID
pesign	358
pcpqa	359
pcp	360
memcached	361
virtlogin	362
ipsilon	363
pkcs11	364
ipaapi	365
kdcproxy	366
ods	367
sssd	368
libvirt	369
gluster	370
fedfs	371
dovenull	372
docker	373
coroqnetd	374
clevis	375
clamscan	376
clamilt	377

Group name	GID
virusgroup	378
virusgroup	378
virusgroup	378
clamupdate	379
colord	380
geoclue	381
printadmin	382
aws-kinesis-agent-user	383
cwagent	384
pulse-rt	385
pulse-access	386
unbound	387
polkitd	388
dirsrv	389
cgred	993
chrony	994
ec2-instance-connect	995
rngd	996
libstoragemgmt	997
ssh_keys	998

Group name	GID
input	999
ec2-user	1000
nfsnobody	65534

Listed by Name

Group name	GID
adm	4
apache	48
arpwatch	77
audio	63
avahi	70
avahi-autoipd	170
aws-kinesis-agent-user	383
bin	1
cdrom	11
cgred	993
chrony	994
clamilt	377
clamscan	376
clamupdate	379

Group name	GID
clevis	375
colord	380
coroqnetd	374
cwagent	384
daemon	2
dbus	81
dhcpd	177
dialout	18
dip	40
dirsrv	389
disk	6
disk	6
docker	373
dovecot	97
dovenull	372
ec2-instance-connect	995
ec2-user	1000
fax	78
fedfs	371
floppy	19

Group name	GID
ftp	50
games	20
gdm	42
geoclue	381
gluster	370
haclient	189
haproxy	188
hsqldb	96
ident	98
input	999
ipaapi	365
ipsilon	363
kdcproxy	366
kmem	9
kvm	36
ldap	55
libstoragemgmt	997
libvirt	369
lock	54
lp	7

Group name	GID
mail	12
mailman	41
mailnull	47
man	15
mem	8
memcached	361
mysql	27
named	25
nfsnobody	65534
nobody	99
nscd	28
nscd	28
ntp	38
ods	367
oprofile	16
pcp	360
pcpqa	359
pegasus	65
pesign	358
pkcs11	364

Group name	GID
pkiuser	17
polkitd	388
postdrop	90
postfix	89
postgres	26
printadmin	382
pulse	171
pulse-access	386
pulse-rt	385
qemu	107
quagga	92
quaggavt	85
radiusd	95
radiusd	95
radvd	75
rngd	996
root	0
rpc	32
rpcuser	29
rtkit	172

Group name	GID
saned	357
sanlock	179
saslauth	76
saslauth	76
screen	84
sfcb	356
slocate	21
smmsp	51
squid	23
ssh_keys	998
sshd	74
sssd	368
stap-server	155
stapdev	158
stapsys	157
stapusr	156
stapusr	156
sys	3
systemd-journal	190
systemd-journal	190

Group name	GID
systemd-journal-gateway	191
systemd-journal-remote	356
systemd-journal-upload	355
systemd-network	192
systemd-resolve	193
tang	354
tape	33
tape	33
tcpdump	72
tomcat	53
tss	59
tty	5
unbound	387
usbmon	351
usbmuxd	113
users	100
utempter	35
utmp	22
uucp	14
uuidd	353

Group name	GID
video	39
virtlogin	362
virusgroup	378
virusgroup	378
virusgroup	378
wbpriv	88
wbpriv	88
wheel	10
wireshark	352

AL2 Source Packages

You can view the source of packages you have installed on your instance for reference purposes by using tools provided in Amazon Linux. Source packages are available for all of the packages included in Amazon Linux and the online package repository. Determine the package name for the source package you want to install and use the **yumdownloader --source** command to view source within your running instance. For example:

```
[ec2-user ~]$ yumdownloader --source bash
```

The source RPM can be unpacked and, for reference, you can view the source tree using standard RPM tools. After you finish debugging, the package is available for use.

Security and Compliance in AL2

Cloud security at Amazon is the highest priority. 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.

Enable FIPS Mode on AL2

This section explains how to enable Federal Information Processing Standards (FIPS) on AL2. For more information about FIPS, see:

- Federal Information Processing Standard (FIPS)
- Compliance FAQs: Federal Information Processing Standards

Prerequisites

- An existing AL2 Amazon EC2 instance with access to the internet to download required packages. For more information about launching an AL2 Amazon EC2 instance, see AL2 on Amazon EC2.
- You must connect to your Amazon EC2 instance using SSH or Amazon Systems Manager. .

> ⚠ **Important**
>
> ED25519 SSH user keys aren't supported in FIPS mode. If you launched your Amazon EC2 instance using an ED25519 SSH key pair, you must generate new keys using another

algorithm (such as RSA) or you may lose access to your instance after enabling FIPS mode. For more information see Create key pairs in the *Amazon EC2 User Guide*.

Enable FIPS Mode

1. Connect to your AL2 instance using SSH or Amazon Systems Manager.
2. Ensure the system is up to date. For more information, see Package repository.
3. Install and enable the `dracut-fips` module by running the following commmands.

```
sudo yum -y install dracut-fips
sudo dracut -f
```

4. Enable FIPS mode on the Linux kernel command-line using the following command. This will enable FIPS mode system-wide for the modules listed in the AL2 FAQ

```
sudo /sbin/grubby --update-kernel=ALL --args="fips=1"
```

5. Reboot your AL2 instance.

```
sudo reboot
```

6. To verify that FIPS mode is enabled, reconnect to your instance and run the following command.

```
sysctl crypto.fips_enabled
```

You should see the following output:

```
crypto.fips_enabled = 1
```

You can also verify that OpenSSH is in FIPS mode by running the following command:

```
ssh localhost 2>&1 | grep FIPS
```

You should see the following output:

```
FIPS mode initialized
```