用户指南

Amazon Linux 2023

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon Linux 2023: 用户指南

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

什么是 Amazon Linux 2023？ 1
发布频率 1
主要和次要发布 2
使用新版本 2
长期支持政策 2
命名和版本控制 3
性能和操作优化 4
与 Fedora 的关系 5
定制 cloud-init 5
安全更新和功能 6
管理更新 7
云中的安全性 7
SELinux 模式 7
合规性计划 7
SSH 服务器默认 7
OpenSSL 3 的主要功能 7
网络服务 8
核心工具链包 glibc, gcc, binutils 8
软件包管理工具 9
默认 SSH 服务器配置 10
已弃用的功能 12
`compat-`软件包 12
已弃用的功能已在中停用 AL1，已在中删除 AL2 12
32 位 x86 (i686) AMIs 13
`aws-apitools-*`取而代之的是 AWS CLI 13
`systemd`替换`upstart`在 AL2 14
在 023 年中弃用 AL2 并移除的 AL2功能 14
32 位 x86 (i686) 软件包 15
`aws-apitools-*`替换为 AWS CLI 15
`amazon-cloudwatch-agent`取代 `awslogs` 16
`bzr`版本控制系统 16
cgroup v1 16
log4j热补丁 () `log4j-cve-2021-44228-hotpatch` 16
`lsb_release` 和 `system-lsb-core` 软件包 16

mcrypt 17
OpenJDK 7 () java-1.7.0-openjdk 17
Python 2.7 17
rsyslog-openssl取代 rsyslog-gnutls 18
网络信息服务 (NIS)/yp 18
亚马逊 VPC 中有多个域名 create-dhcp-options 18
Sun RPC中的glibc 18
日志中有 OpenSSH 密钥指纹 audit 19
ld.goldLinker 19
ping6 19
2023 年 AL2已被弃用 19
32 位 x86 (i686) 运行时支持 20
aspell 20
伯克利数据库 (libdb) 20
cron 20
IMDSv1 21
pcre 版本 1 21
System V init (sysvinit) 21
已弃用 EOL 软件包 21
比较 AL2 和 AL2 023 23
添加、升级和删除了软件包 24
针对每个版本的支持 24
命名和版本控制更改 24
优化 24
来自多个上游 25
网络系统服务 25
软件包管理器 25
使用 cloud-init 25
图形桌面支持 26
编译器三元组 26
32 位 x86 (i686) 软件包 26
lsb_release 和 system-lsb-core 软件包 26
EPEL 27
axel-HTTP/FTP 客户端 28
brotli和 libbrotli-压缩 29
collectd-统计数据收集守护程序 29

cpulimit ... 29
exim-邮件传输代理 ... 30
fuse3-用户空间中的文件系统 (FUSE) v3 ... 30
ganglia-分布式监控系统 ... 30
git-lfs-使用 Git 控制大文件的版本 ... 30
haveged-使用熵源 HAVEGE algorithm ... 30
inotify-tools-inotify 命令行工具 ... 30
iperf-TCP/UDP 性能基准测试 ... 31
jemalloc-替代malloc实现 ... 31
libbsd-兼容 BSD 的函数库 ... 31
libserf-HTTP 客户端库 ... 32
libzstd-zstd 压缩库 ... 32
lighttpd网络服务器 ... 32
lshell-受限炮弹 ... 32
monit-进程、文件、目录和设备监视器 ... 33
nodejs ... 33
perl-Config-General ... 33
python2-lockfile-文件锁定 ... 34
python2-rsa-纯粹的 Python RSA ... 34
python2-simplejson-适用于 Python 2 的 JSON 例程 ... 34
rkhunter-rootkit Hunter ... 35
rssh-适用于 OpenSSH 的受限外壳 ... 35
sscg-自签名 SSL 证书生成器 ... 35
stress-压力测试 ... 35
stress-ng-压力测试 ... 35
tmpwatch-根据上次访问时间删除文件 ... 36
xmlstarlet-命令行 XML 实用工具 ... 36
Python 2.7 已被 Python 3 所取代 ... 36
安全更新 ... 36
SELinux ... 37
OpenSSL 3 ... 37
IMDSv2 ... 37
移除 log4j 热补丁 () log4j-cve-2021-44228-hotpatch ... 38
确定性升级用于提高稳定性 ... 38
gp3作为默认 Amazon EBS 卷类型 ... 39
统一控制组层次结构 (cgroup v2) ... 39

systemd计时器替换 cron 39
改进的工具链：gcc, binutils，以及 glibc 40
systemd日记替换 rsyslog 40
最大限度地减少软件包依赖性 41
curl 和 libcurl 的软件包变更 41
GNU Privacy Guard (GNUPG) 41
Amazon Corretto 作为默认 JVM 41
AWS CLI v2 42
UEFI 首选和安全启动 42
SSH 服务器默认配置更改 42
AL2023 中的内核更改从 AL2 42
IPv4 TTL 42
注重安全的内核配置变化 43
其他内核配置变化 47
内核文件系统支持 48
/tmp改变 53
AMI 和容器镜像变更 54
亚马逊 Linux 2 和 AL2 023 AMI 对比 54
亚马逊 Linux 2 和 AL2 023 最小 AMI 比较 87
亚马逊 Linux 2 和 AL2 023 容器对比 108
比较 AL1 和 AL2 023 116
针对每个版本的支持 116
systemd 取代 upstart 作为 init 系统 116
Python 2.6 和 2.7 已被 Python 3 所取代 117
OpenJDK 8 是最旧的 JDK 117
AL2023 中的内核更改从 AL1 117
内核实时修补 117
内核文件系统支持 117
注重安全的内核配置变化 120
其他内核配置变化 122
AL1 和 AL2 023 AMI 的比较 123
AL1 和 AL2 023 最小 AMI 对比 157
AL1 和 AL2 023 集装箱对比 177
系统要求 186
运行 AL2 023 时的 CPU 要求 186
AL22023 年的 ARM CPU 要求 186

x86-64 023 的 CPU 要求 AL2 187
运行 AL2 023 所需的内存 (RAM) 187
图形桌面 189
相关 主题 189
正在运行的应用程序 190
使用资源控制 systemd 190
systemd-run用于运行一次性命令的资源控制 190
systemd服务中的资源控制 193
使用 cgroups utilities 197
在 on 上使用 AL2 023 AWS 199
入门 AWS 199
注册获取 AWS 账户 199
创建具有管理访问权限的用户 200
授权以编程方式访问 201
AL2亚马逊上的 023 EC2 202
使用亚马逊 EC2 控制台启动 AL2 023 203
使用 SSM 参数启动 AL2 023 和 AWS CLI 204
使用启动最新的 AL2 023 AMI AWS CloudFormation 205
使用特定的 AMI ID 启动 AL2 023 206
AL2023 AMI 的弃用和生命周期 207
正在连接到 AL2 023 个实例 207
比较 AL2 023 标准（默认）和最小值 AMIs 208
AL2023 装在容器里 235
AL2023 基础容器镜像 236
AL2023 最小容器镜像 238
构建基本系统 AL2 023 容器镜像 239
AL2023 容器镜像包列表对比 243
AL2023 与容器镜像相比 AMI 最小 249
AL2023 在 Elastic Beanstalk 上 266
AL2023 CloudShell 266
AL2023 适用于亚马逊 ECS 容器主机 267
此后 Amazon ECS 的相关变更 AL2 267
自定义 Amazon ECS 优化的 AMI 268
AL22023 年的亚马逊 EFS 268
amazon-efs-utils 268
装载 Amazon EFS 文件系统 269

2023 年的亚马逊 EMR AL2 269
AL2基于 023 的亚马逊 EMR 发布 269
AL2基于 023 的 EKS 上的 Amazon EMR 269
AL2023 开启 AWS Lambda 269
`provided.al2023`Lambda 运行时 270
AL2基于 023 的运行时 270
教程 271
在 AL2 023 上安装 LAMP 271
步骤 1：准备 LAMP 服务器 272
步骤 2：测试 LAMP 服务器 276
步骤 3：确保数据库服务器的安全 278
步骤 4：（可选）安装 phpMyAdmin 279
故障排除 282
相关 主题 282
在 023 上配置 SSL/TLS AL2 283
前提条件 284
步骤 1：在服务器上启用 TLS 285
步骤 2：获取 CA 签名的证书 287
步骤 3：测试和强化安全配置 294
故障排除 297
在 AL2 023 上 WordPress 发布一篇博客 298
先决条件 299
安装 WordPress 299
后续步骤 308
帮助！我的公有 DNS 名称发生更改导致我的博客瘫痪 309
023 年 Redis 6 到 Valkey 过渡 AL2 310
Redis 6 的 Support 时间表 310
Valkey 简介 311
迁移计划和时间表 311
迁移选项和步骤 311
相关 主题 314
在 023 上 AL2安装 GNOME 314
前提条件 315
安装 315
相关 主题 315
在 023 上 AL2配置 VNC 315

前提条件 316
步骤 1：安装 316
步骤 2：配置 316
步骤 3：使用 VNC 客户端进行连接 318
（可选）启动时启动服务 318
（可选）禁用空闲锁屏 319
相关 主题 319
AL2亚马逊以外的 023 EC2 320
下载 AL2 023 虚拟机镜像 320
支持的配置 320
KVM 要求 321
VMware 要求 323
Hyper-V 要求 325
AL2023 虚拟机配置 326
基于 NoCloud `seed.iso` 的配置 327
VMware 基于访客信息的配置 330
AL2023 标准 AMI 和 KVM 镜像的软件包清单对比 332
AL2023 标准 AMI 和 VMware OVA 镜像的软件包清单对比 357
AL2023 标准 AMI 和 Hyper-V 映像的软件包清单对比 383
文件系统布局 409
`/` 409
`/boot` 409
`/boot/efi` 410
`/etc` 410
`/home` 410
`/root` 411
`/srv` 411
`/tmp` 411
`/run` 412
`/usr` 412
`/usr/bin` 413
`/usr/include` 413
`/usr/lib` 和 `/usr/lib64` 413
`/usr/local` 413
`/usr/share` 413
`/var` 413

/var/cache ... 414
/var/lib ... 414
/var/log ... 414
/var/spool ... 414
/var/tmp ... 414
正在更新 AL2 023 ... 416
安全部署更新的最佳实践 ... 416
为次要更新做准备 ... 418
为重大更新做准备 ... 419
接收有关新更新的通知 ... 419
通过版本控制的存储库进行确定性升级 ... 420
控制从主要版本和次要版本收到的更新 ... 420
主要版本升级和次要版本升级之间的区别 ... 421
知道何时有更新 ... 421
控制 AL2 023 存储库中可用的软件包更新 ... 421
实例替换 ... 422
就地确定性升级 ... 422
管理更新 ... 429
查看可用的软件包更新 ... 430
使用应用安全更新 DNF 和存储库版本 ... 434
（安全）更新后自动重启服务 ... 447
何时需要重启才能应用安全更新？ ... 448
启动已启用最新存储库版本的实例 ... 448
获取软件包支持信息 ... 449
dnf check-release-update ... 449
添加、启用或禁用新存储库 ... 453
使用 cloud-init 添加存储库 ... 455
内核实时修补 ... 456
限制 ... 457
支持的配置和先决条件 ... 457
使用内核实时修补 ... 458
内核更新 ... 463
AL2023 上的 Linux 内核版本 ... 463
将 AL2 023 更新到内核 6.12 ... 463
AL2023 内核-常见问题解答 ... 465
编程语言和运行时 ... 467

C/C++ 还有 Fortran 467
Go 468
AL2023 Lambda 函数：Go 468
Java 469
NodeJS 33
Perl 470
Perl modules 470
PHP 470
迁移到新版本 PHP 版本 471
迁移来源 PHP 7.x 471
PHP modules 471
Python 471
Python modules 472
Rust 472
AL2023 Lambda 函数：Rust 473
AL2023 预留用户和群组 474
AL2023 个预留用户名单 474
AL2023 预留组名单 482
023 年推出的编解码器 AL2 495
安全性与合规性 496
安全公告 497
ALAS 通告 497
唉 FAQs 497
ALAS 公告 497
公告和 RPM 存储库 498
咨询 IDs 498
公告发布日期和公告更新日期 499
咨询类型 499
咨询严重性 499
公告和套餐 500
公告和 CVEs 500
咨询文本 501
内核实时补丁公告 501
`updateinfo.xml` 架构 502
列出适用的公告 502
就地更新 506

应用公告中提及的更新 506
为 AL2 0 SELinux 23 设置模式 509
AL2023 的默认 SELinux 状态和模式 510
改为 enforcing 模式 511
禁用选项 SELinux 512
在 023 上 AL2启用 FIPS 模式 513
在 AL2 023 容器中启用 FIPS 模式 514
在 023 上交换 OpenSSL FIPS 提供商 AL2 516
内核强化 518
内核强化选项（与架构无关） 518
x86-64 特定的内核强化选项 533
aarch64 特定的内核强化选项 536
UEFI 在 023 上安全启动 AL2 538
在 023 上启用 UEFI 安全启动 AL2 538
注册现有实例 539
注册快照映像 539
撤消更新 540
UEFI Secure Boot 在 023 上是如何工作的 AL2 540
注册您自己的密钥 541
..... dxlii

什么是 Amazon Linux 2023？

亚马逊 Linux 2023 (AL2023) 是亚马逊网络服务 (AWS) 推出的下一代亚马逊 Linux。借 AL2助 023，您可以在安全、稳定和高性能的运行时环境中开发和运行云和企业应用程序。此外，您还可以获得一个提供长期支持的应用程序环境，可以访问 Linux 中的最新创新。 AL2023 不收取额外费用。

AL2023 是亚马逊 Linux 2 (AL2) 的继任者。有关 023 和之间区别的信息 AL2，请参阅比较 AL2 和 AL2 023和 AL2 023 中的 AL2 Package 更改。

主题

- 发布频率
- 命名和版本控制
- 性能和操作优化
- 与 Fedora 的关系
- 定制 cloud-init
- 安全更新和功能
- 网络服务
- 核心工具链包 glibc, gcc, binutils
- 软件包管理工具
- 默认 SSH 服务器配置

发布频率

亚马逊 Linux 2023 (AL2023) 于 2023 年 3 月发布，支持时间将持续到 2029 年 6 月 30 日。支持分为两个阶段：

- 标准支持 — 在此阶段，该版本每季度都会收到次要版本更新。标准支持阶段将于 2027 年 6 月 30 日结束。
- 维护 — 在此阶段，该版本仅收到安全更新和关键错误修复。这些更新一经发布就会立即发布。维护阶段将于 2029 年 6 月 30 日结束。

主要和次要发布

伴随每个 Amazon Linux 发布（主要版本、次要版本或安全发布），我们都会发布一个新的 Linux Amazon 机器映像 (AMI)。

- 主要版本发布 — 包括整个堆栈的安全和性能的新功能和改进。这些改进可能包括对内核、工具链的重大更改，Glib C, OpenSSL，以及任何其他系统库和实用程序。亚马逊 Linux 的主要版本部分基于上游 Fedora Linux 发行版的当前版本。 AWS 可能会添加或替换来自其他非 Fedora 上游的特定软件包。
- 次要版本发布 — 季度更新，其中包括安全更新、错误修复以及新功能和软件包。每个次要版本都是更新的一个累积列表，其中除了包括新功能和软件包，还包括安全和错误修复。这些版本可能包括最新的语言运行时，例如 PHP。 它们还可能包括其他流行的软件包，例如Ansible和Docker。

使用新版本

更新是通过新的 Amazon 机器映像 (AMI) 版本与相应的新存储库的组合提供的。默认情况下，新的 AMI 与它所指向的存储库耦合在一起。但是，随着时间的推移，您可以将正在运行的 Amazon EC2 实例指向更新的存储库版本，以便在正在运行的实例上应用更新。您也可以通过启动最新版本的新实例进行更新 AMIs。

长期支持政策

Amazon Linux 会在一个主要版本中为您基于 Amazon Linux 构建的应用程序提供所有软件包更新并保持兼容性。核心软件包，例如 glibc 图书馆，OpenSSL, OpenSSH，还有 DNF 软件包管理器在 AL2 023 主版本的生命周期内获得支持。不属于核心软件包的软件包将基于其特定的上游来源获得支持。您可以通过运行以下命令查看各个软件包的具体支持状态和日期。

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

您可以通过运行以下命令了解当前已安装的所有软件包的信息。

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

核心软件包的完整列表在预览期间最终确定。如果您想了解更多的核心软件包，请告诉我们。我们会在收集反馈时作出评估。有关 AL2 023 的反馈可以通过您的指定 AWS 代表提供，也可以通过在亚马逊 linux- 2023 存储库中提交问题来提供。 GitHub

命名和版本控制

AL2在两年的标准支持期间，023 每三个月提供一次次要版本。每个版本都用从 0 到 N 的一个增量标识。0 指该迭代的最早的主要版本。所有版本都称为 Amazon Linux 2023。下一版 Amazon Linux 发布时， AL2023 将进入扩展支持，并接收有关安全更新和关键错误修复的更新。

例如， AL2023 的次要版本采用以下格式：

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

相应的 AL2 023 AMIs 具有以下格式：

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

在一个特定的次要版本中，常规 AMI 发布使用 AMI 发布日期的时间戳。

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

识别 AL2 或 AL2 023 实例的推荐方法是从中读取通用平台枚举 (CPE) 字符串。`/etc/system-release-cpe`然后，将该字符串拆分为多个字段。最后，阅读平台和版本值。

AL2023 还引入了用于平台识别的新文件：

- `/etc/amazon-linux-release` 符号链接到 `/etc/system-release`
- `/etc/amazon-linux-release-cpe` 符号链接到 `/etc/system-release-cpe`

这两个文件表明实例是 Amazon Linux。除非您想知道特定的平台和版本值，否则无需读取文件或将字符串拆分为多个字段。

性能和操作优化

Amazon Linux 6.1 内核

- AL2023 使用弹性网络适配器 (ENA) 和弹性结构适配器 (EFA) 设备的最新驱动程序。 AL2023 侧重于 Ama EC2 zon 基础设施中硬件的性能和功能向后移植。
- 内核实时修补适用于 `x86_64` 和 `aarch64` 实例类型。这减少了频繁重启的需求。
- 所有内核编译和运行时配置都包含许多相同的性能和操作优化。 AL2

基本工具链选择和默认构建标志

- AL2023 包是在默认启用编译器优化 (`-O2`) 的情况下构建的
- AL2023 软件包需要`x86-64v2``x86-64`系统 (`-march=x86-64-v2`)，Graviton 2 或更高版本需要系统 ()。`aarch64` `-march=armv8.2-a+crypto -mtune=neoverse-n1`
- AL2023 个包是在启用自动矢量化功能的情况下构建的 ()。`-ftree-vectorize`
- AL2023 个包是在启用链接时间优化 (LTO) 的情况下构建的。
- AL2023 使用更新版本的 Rust, Clang/LLVM，以及 Go.

软件包选择和版本

- 主要系统组件的精选反向移植包括在 Amazon EC2 基础设施（尤其是 Graviton 实例）上运行的多项性能改进。
- AL2023 集成了多个 AWS 服务 和功能。这包括 AWS CLI、SSM 代理、亚马逊 Kinesis 代理和。CloudFormation
- AL2023 使用 Amazon Corretto 作为 Java 开发套件 (JDK)。
- AL2023 为上游项目发布的较新版本提供数据库引擎和编程语言运行时更新。当发布新版本时，将添加有关新版本的编程语言运行时。

云环境中的部署

- 基本 AL2 023 AMI 和容器镜像经常更新以支持修补实例替换。
- AL2023 AMI 更新中包含内核更新。这意味着您无需使用像 `yum update` 和 `reboot` 这样的命令就能更新内核。
- 除了标准的 AL2 023 AMI 之外，还提供最低限度的 AMI 和容器镜像。如果选择AMI 最低版本，则使用运行服务所需的最少量软件包来运行环境。

- 默认情况下， AL2023 AMIs 和容器被锁定到软件包存储库的特定版本。当它们启动时，不会自动更新。这意味着您始终掌控着何时提取任何软件包更新。在投入生产之前，您总是可以先在 beta/gamma 环境中测试一下。如果出现问题，可以使用预先验证的回滚路径。

与 Fedora 的关系

AL2023 维护自己的发行版并支持独立于 Fedora 的生命周期。 AL2023 提供开源软件的更新版本、更多种类的软件包和频繁的版本。这保留了熟悉的基于 RPM 的操作系统。

AL2023 的通用版本 (GA) 无法直接与任何特定的 Fedora 版本相提并论。 AL2023 GA 版本包括 Fedora 34、35 和 36 中的组件。有些组件与 Fedora 中的组件相同，有些则经过修改。其他组件与 CentOS Stream 9 中的组件更为相似，或者是独立开发的。Amazon Linux 内核源自 kernel.org 上长期支持的且独立于 Fedora 选择的选项。

定制 cloud-init

这些区域有：cloud-init package 是一个开源应用程序，可在云计算环境中引导 Linux 映像。有关更多信息，请参阅 cloud-init 文档。

AL2023 包含的自定义版本为 cloud-init。 随着 cloud-init，您可以指定实例在启动时会发生什么。

启动实例时，您可以使用用户数据字段将操作传递给 cloud-init。 这意味着您可以在许多用例中使用常见的 Amazon 系统映像 (AMIs)，并在启动实例时对其进行动态配置。 AL2023 还使用 cloud-init 来配置`ec2-user`账户。

AL2023 使用 cloud-init `/etc/cloud/cloud.cfg.d`和中的操作`/etc/cloud/cloud.cfg`。你可以自己创作 cloud-init `/etc/cloud/cloud.cfg.d`目录中的操作文件。Cloud-init 按字典顺序读取此目录中的所有文件。时间较晚的文件覆盖时间较早的文件中的值。时间 cloud-init 启动一个实例，cloud-init 软件包执行以下配置任务：

- 设置默认区域
- 设置主机名
- 解析和处理用户数据
- 生成主机私有 SSH 密钥
- 将用户的公有 SSH 密钥添加到 `.ssh/authorized_keys` 以便于登录和管理
- 准备存储库以进行软件包管理

- 处理用户数据中定义的软件包操作
- 运行用户数据中的用户脚本
- 装载实例存储卷（如果适用）
  - 默认情况下，如果 `ephemeral0` 实例存储卷存在且包含一个有效文件系统，则说明实例存储卷已装载在 `/media/ephemeral0`。否则，说明未装载。
  - 默认情况下，对于 `m1.small` 和 `c1.medium` 实例类型，将装载与实例关联的所有交换卷。
  - 您可以使用以下方法覆盖默认的实例存储卷装载 cloud-init 指令：

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

    要进一步控制坐骑，请参阅中的坐骑 cloud-init 文档中）。
  - 当一个实例启动时，支持 TRIM 的实例存储卷不会被格式化。要装载实例存储卷，您必须对实例存储卷进行分区和格式化。

    有关更多信息，请参阅 Amazon EC2 用户指南中的实例存储卷 TRIM 支持。
  - 当您启动实例时，可以使用 `disk_setup` 模块对实例存储卷进行分区和格式化。

    有关更多信息，请参阅中的 “磁盘设置” cloud-init 文档中）。

有关使用的信息 cloud-init wit SELinux h，请参阅使用 cloud-init 启用`enforcing`模式。

有关信息 cloud-init 用户数据格式，请参阅中的用户数据格式 cloud-init 文档中）。

安全更新和功能

AL2023 提供了许多安全更新和解决方案。

主题

- 管理更新
- 云中的安全性
- SELinux 模式
- 合规性计划
- SSH 服务器默认

- OpenSSL 3 的主要功能

管理更新

使用应用安全更新 DNF 和存储库版本。有关更多信息，请参阅 在 AL2 023 中管理软件包和操作系统更新。

云中的安全性

安全是双方共同承担 AWS 的责任。责任共担模型将其描述为云的安全和云中的安全。有关更多信息，请参阅 Amazon Linux 2023 中的安全性与合规性。

SELinux 模式

默认情况下，在 AL2 023 中 SELinux 处于启用状态并设置为宽容模式。在许可模式下，记录但不强制执行权限拒绝。

这些 SELinux 策略定义了用户、进程、程序、文件和设备的权限。使用 SELinux，您可以从两个策略中选择一个。这些策略有针对性或为多级安全 (MLS)。

有关 SELinux 模式和策略的更多信息，请参阅为 AL2 0 SELinux 23 设置模式和 P SELinux roject Wiki。

合规性计划

独立审计师评估 AL2 023的安全性和合规性以及许多合 AWS 规计划。

SSH 服务器默认

AL2023 包括 OpenSSH 8.7。OpenSSH 8.7 默认禁用 `ssh-rsa` 密钥交换算法。有关更多信息，请参阅 默认 SSH 服务器配置。

OpenSSL 3 的主要功能

- 证书管理协议（CMP，RFC 4210）同时包括 CRMF (RFC 4211) 和 HTTP 传输 (RFC 6712)。
- A HTTP 或 HTTPS libcrypto 中的客户端支持 GET 以及 POST 操作、重定向、普通和 ASN.1-编码的内容、代理和超时。
- 这些区域有：EVP_KDF 适用于密钥派生函数。

- 这些区域有：EVP_MAC API 与之合作 MACs.
- Linux 内 TLS 支持。

有关更多信息，请参阅 OpenSSL 迁移指南。

网络服务

开源项目 `systemd-networkd` 在现代 Linux 发行版中随处可见。该项目使用的声明式配置语言与其余的 `systemd` 框架类似。它的主要配置文件类型是 `.network` 和 `.link` 文件。

`amazon-ec2-net-utils` 软件包在 `/run/systemd/network` 目录中生成接口特定的配置。当接口连接到实例时 IPv4 ，这些配置可以同时在接口上启用 IPv6 网络连接。这些配置还安装策略路由规则，有助于确保本地来源的流量通过相应实例的网络接口路由到网络。这些规则可确保从关联的地址或前缀通过弹性网络接口 (ENI) 路由正确的流量。有关使用 ENI 的更多信息，请参阅 Amazon EC2 用户指南中的使用 ENI。

您可以通过在 `/etc/systemd/network` 目录中放置一个自定义配置文件以覆盖 `/run/systemd/network` 中包含的默认配置设置，来自定义此网络行为。

systemd.network 文档描述了 `systemd-networkd` 服务如何确定适用于特定接口的配置。它还会生成替代名称，称为 altnames，让 ENI 支持的接口能够反映各种 AWS 资源的属性。这些 ENI 支持的接口属性是 ENI 连接的 `ENI ID` 和 `DeviceIndex` 字段。您在使用各种工具（例如 `ip` 命令）时，可以使用这些接口的属性来引用这些接口。

AL2023 实例接口名称是使用`systemd`插槽命名方案生成的。有关更多信息，请参阅 systemd.net 命名方案。

此外，默认情况下， AL2023 使用`fq_codel`主动队列管理网络传输调度算法。有关更多信息，请参阅CoDel概述。

核心工具链包 glibc, gcc, binutils

Amazon Linux 中的一部分软件包被指定为核心工具链软件包。作为 AL2 023 的重要组成部分，核心软件包将获得五年的支持。我们可能更改某个软件包的版本，但长期支持适用于 Amazon Linux 版本中包含的该软件包。

这三个核心软件包提供用于构建 Amazon Linux 发行版中大多数软件的系统工具链。

软件包	定义	用途
glibc 2.34	系统 C 库	供大多数提供标准函数的二进制程序以及程序和内核之间的接口使用。
gcc 11.2	gcc 编译器套件	编译 C, C++, Fortran.
binutils 2.35	汇编器和链接器以及其他二进制工具	操作或检查二进制程序。

我们建议更新任何 glibc 库之后会重新启动。要更新控制服务的软件包，重新启动服务便足以实现更新。不过，系统重启可以确保之前的所有软件包和库更新都能完成。

软件包管理工具

AL2023 中的默认软件包管理工具是 DNF. DNF 是的继任者 YUM，中的软件包管理工具 AL2。

DNF 类似于 YUM 在其用法中。很多 DNF 命令和命令选项相同 YUM 命令。在命令行界面 (CLI) 命令中，大多数情况下用 `dnf` 替换 `yum`。

例如，对于以下 AL2 `yum`命令：

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

在 AL2 023 中，它们变成了以下命令：

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

在 AL2 023 中，该`yum`命令仍然可用，但可以作为指向`dnf`命令的指针。因此，在 `yum` shell 或脚本中使用该命令时，所有命令和选项都与 DNF CLI。 有关两者之间区别的更多信息 YUM CLI 还有 DNF CLI，请参阅中的更改 DNF CLI 相比之下 YUM.

有关 `dnf` 命令的命令和选项的完整参考，请参阅手册页 `man dnf`。有关更多信息，请参阅 DNF 命令参考。

默认 SSH 服务器配置

如果您有几年前的 SSH 客户端，可能在连接到实例时会看到一个错误。如果该错误告诉您未找到匹配的主机密钥类型，请更新您的 SSH 主机密钥来解决此问题。

默认禁用 **ssh-rsa** 签名

AL2023 包括默认配置，该配置禁用旧版`ssh-rsa`主机密钥算法并生成一组简化的主机密钥。客户端必须支持 `ssh-ed25519` 或 `ecdsa-sha2-nistp256` 主机密钥算法。

该默认配置接受以下任一密钥交换算法：

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

默认情况下， AL2023 生成`ed25519`和`ECDSA`主机密钥。客户端支持 `ssh-ed25519` 或 `ecdsa-sha2-nistp256` 主机密钥算法。当您通过 SSH 连接到实例时，必须使用支持兼容算法（例如 `ssh-ed25519` 或 `ecdsa-sha2-nistp256`）的客户端。如果您需要使用其他密钥类型，请使用用户数据中的一个 `cloud-config` 片段覆盖生成的密钥列表。

在以下示例中，`cloud-config` 使用 `ecdsa` 和 `ed25519` 密钥生成一个 `rsa` 主机密钥。

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

如果您使用 RSA 密钥对进行公钥身份验证，则您的 SSH 客户端必须支持 `rsa-sha2-256` 或 `rsa-sha2-512` 签名。如果您使用的是不兼容的客户端并且无法升级，请在您的实例上重新启用 `ssh-rsa` 支持。要重新启用`ssh-rsa`支持，请使用以下命令激活LEGACY系统加密策略。

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

有关管理主机密钥的更多信息，请参阅 Amazon Linux 主机密钥。

2023 年 AL2已弃用的功能

此处记录了 023 中 AL2 已弃用但在 AL2 023 中不存在的功能。这是诸如功能和软件包之类的功能，它们存在于 023 中 AL2，但不存在于 AL2 023 中，也不会添加到 AL2 023 中。有关该功能的支持期限的更多信息 AL2，请参阅中的已弃用功能。 AL2

AL2023 中还有一些功能已被弃用，并将在未来的版本中移除。本章介绍该功能是什么、何时不再受支持，以及何时将其从 Amazon Linux 中删除。了解已弃用的功能将有助于您部署 AL2 023，并为下一个主要版本的 Amazon Linux 做好准备。

主题

- compat-软件包
- 已弃用的功能已在中停用 AL1，已在中删除 AL2
- 在 023 年中弃用 AL2 并移除的 AL2功能
- 2023 年 AL2已被弃用

`compat-`软件包

提供前缀为的所有软件包`compat-`都是为了 AL2 与尚未针对该软件包的现代版本重建的旧二进制文件进行二进制兼容。每个新的主要版本的 Amazon Linux 都不会继承先前版本中的任何`compat-`软件包。

Amazon Linux 发行版（例如 AL2）中的所有`compat-`软件包都已弃用，并且在后续版本（例如 AL2 023）中不存在。我们强烈建议针对库的更新版本重新构建软件。

已弃用的功能已在中停用 AL1，已在中删除 AL2

本节介绍中提供的功能 AL1以及中不再提供的功能 AL2。

> **Note**
>
> 作为维护支持阶段的一部分 AL1，某些软件包的 end-of-life (EOL) 日期早于 EOL。 AL1有关更多信息，请参阅 Pac AL1 kage 支持声明。

> **Note**
>
> 某些 AL1 功能在早期版本中已停用。有关信息，请参阅AL1 发行说明。

主题

- 32 位 x86 (i686) AMIs
- aws-apitools-*取而代之的是 AWS CLI
- systemd替换upstart在 AL2

32 位 x86 (i686) AMIs

作为 2014.09 版本的一部分， AL1亚马逊 Linux 宣布这将是最后一个产生 32 位的版本。 AMIs因此，从 2015.03 版本开始， AL1Amazon Linux 不再支持在 32 位模式下运行系统。 AL2 为 x86-64 主机上的 32 位二进制文件提供了有限的运行时支持，并且不提供开发包来支持构建新的 32 位二进制文件。 AL2023 不再包含任何 32 位用户空间包。我们建议用户在迁移到 AL2 023 之前完成向 64 位代码的过渡。

如果您需要在 023 上运行 32 位二进制文件，则可以从 AL2 023 之上运行的 AL2 容器 AL2 内部使用 32 位用户空间。 AL2

`aws-apitools-*`取而代之的是 AWS CLI

在 2013 AWS CLI 年 9 月发布之前，提供 AWS 了一组命令行实用程序，实现于 Java，它允许用户调用亚马逊 EC2 API。这些工具已于 2015 年停产， AWS CLI 成为通过命令行与 Amazon 进行交互 EC2 APIs 的首选方式。这组命令行实用程序包括以下`aws-apitools-*`软件包。

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

对这些`aws-apitools-*`软件包的上游支持已于 2017 年 3 月结束。尽管缺乏上游支持，但 Amazon Linux 继续提供其中一些命令行实用程序，例如`aws-apitools-ec2`，为用户提供向后兼容性。该

AWS CLI 工具比`aws-apitools-*`软件包更强大、更完整，因为它得到了积极维护，并且提供了一种使用所有软件包的方法 AWS APIs。

这些`aws-apitools-*`软件包已于 2017 年 3 月被弃用，不会再收到更多更新。其中任何一个软件包的所有用户都应 AWS CLI 尽快迁移到。这些软件包在 AL2 023 中不存在。

AL1 还提供了`aws-apitools-iam`和`aws-apitools-rds`软件包，这些软件包已在中弃用 AL1，从 AL2 此以后不存在于 Amazon Linux 中。

`systemd`替换`upstart`在 AL2

AL2 是第一个使用初`systemd`始化系统的 Amazon Linux 版本，取而代`upstart`之 AL1的是。在从较新版本的 Amazon Linux 迁移 AL1 到更新版本的过程中，必须更改任何`upstart`特定的配置。它不可能`systemd`在上使用 AL1，因此`systemd`只能在迁移`upstart`到最新的 Amazon Linux 主要版本（例如 AL2 或 AL2 023）时才能完成。

在 023 年中弃用 AL2 并移除的 AL2功能

本节介绍在 023 中可用但在 AL2 AL2 023 中不再可用的功能。

主题

- 32 位 x86 (i686) 软件包
- aws-apitools-*替换为 AWS CLI
- awslogs已弃用，转而使用统一的 Amazon CloudWatch Logs 代理
- bzr版本控制系统
- cgroup v1
- log4j热补丁 () log4j-cve-2021-44228-hotpatch
- lsb_release 和 system-lsb-core 软件包
- mcrypt
- OpenJDK 7 () java-1.7.0-openjdk
- Python 2.7
- rsyslog-openssl取代 rsyslog-gnutls
- 网络信息服务 (NIS)/yp
- 亚马逊 VPC 中有多个域名 create-dhcp-options

- Sun RPC中的glibc
- 日志中有 OpenSSH 密钥指纹 audit
- ld.goldLinker
- ping6

32 位 x86 (i686) 软件包

作为 2014.09 版本的一部分 AL1，我们宣布这将是最后一个产生 32 位的版本。 AMIs因此，从 2015.03 版本开始， AL1Amazon Linux 不再支持在 32 位模式下运行系统。 AL2 为 x86-64 主机上的 32 位二进制文件提供了有限的运行时支持，并且不提供开发包来支持构建新的 32 位二进制文件。 AL2023 不再包含任何 32 位用户空间软件包。我们建议客户完成向 64 位代码的过渡。

如果您需要在 023 上运行 32 位二进制文件，则可以从 AL2 023 之上运行的 AL2 容器 AL2 内部使用 32 位用户空间。 AL2

`aws-apitools-*`替换为 AWS CLI

在 2013 AWS CLI 年 9 月发布之前，提供 AWS 了一组命令行实用程序，这些实用程序已在中实现 Java，允许客户调用 Amazon EC2 API。这些工具已于 2015 年被弃用， AWS CLI 成为通过命令行与 Amazon 进行交互 EC2 APIs 的首选方式。这包括以下`aws-apitools-*`软件包。

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

对这些`aws-apitools-*`软件包的上游支持已于 2017 年 3 月结束。尽管缺乏上游支持，但 Amazon Linux 继续提供其中一些命令行实用程序（例如`aws-apitools-ec2`），以便为客户提供向后兼容性。该 AWS CLI 工具比`aws-apitools-*`软件包更强大、更完整，因为它得到了积极维护，并且提供了一种使用所有软件包的方法 AWS APIs。

这些`aws-apitools-*`软件包已于 2017 年 3 月被弃用，不会再收到更多更新。其中任何一个软件包的所有用户都应 AWS CLI 尽快迁移到。这些软件包在 AL2 023 中不存在。

awslogs已弃用，转而使用统一的 Amazon CloudWatch Logs 代理

该awslogs软件包已在中弃用 AL2 ，在 AL2 023 中不再存在。它已被amazon-cloudwatch-agent软件包中提供的统一 CloudWatch 日志代理所取代。有关更多信息，请参阅 Amazon CloudWatch 日志用户指南。

bzr版本控制系统

GNU Bazaar(bzr) 版本控制系统已于 AL2 023 年 AL2停产，不再存在。

建议bzr的用户将其存储库迁移到git。

cgroup v1

AL2023 移至统一控制组层次结构（cgroup v2），而 AL2 使用 cgroup v1。由于 AL2 不支持 cgroup v2，因此需要在迁移到 023 的过程中完成此迁移。 AL2

log4j热补丁 () log4j-cve-2021-44228-hotpatch

> **Note**
>
> 该log4j-cve-2021-44228-hotpatch软件包已在 023 中弃用 AL2 并于 AL2 023 年删除。

为了回应 CVE-2021-44228，亚马逊 Linux 发布了适用于 Ap ache Log4j 的 Hotpatch 的 RPM 打包版本，适用于和。 AL1 AL2在宣布在 Amazon Linux 中添加热补丁时，我们指出："安装热补丁并不能取代更新到缓解 CVE-2021-44228 或 CVE-2021-45046 的 log4j 版本。"

热补丁是一种缓解措施，可以留出时间来修补 log4j。 AL2023 的第一个正式发布版本是 CVE-2021-44228 发布的 15 个月后，因此 AL2 023 不附带该热补丁（无论是否启用）。

建议在 Amazon Linux 上运行自己的 log4j 版本的客户务必更新到未受 CVE-2021-44228 或 CVE-2021-45046 影响的版本。

lsb_release 和 system-lsb-core 软件包

过去，一些软件会调用该lsb_release命令（ AL2 由软件system-lsb-core包提供）来获取有关其正在运行的 Linux 发行版的信息。Linux Standards Base (LSB) 引入了此命令，Linux 发行版采用了此命令。Linux 发行版已经演变为使用更简单的标准，将这些信息保存在 /etc/os-release 和其他相关文件中。

os-release 标准来自 systemd。有关更多信息，请参阅 systemd os-release 文档。

AL2023 不附带该lsb_release命令，也不包括system-lsb-core软件包。软件应完成向 os-release 标准的过渡，以保持与 Amazon Linux 和其他主要 Linux 发行版的兼容性。

mcrypt

该mcrypt库和相关PHP扩展已在中弃用 AL2，在 AL2 023 中不再存在。

Upstream 在 PHP 7.1 中PHP弃用了该扩展，该mcrypt扩展于 2016 年 12 月首次发布，最终版本于 2019 年 10 月发布。

上游mcrypt库最后一次发布是在 2007 年，并未在 2017 年完成新提交SourceForge 所需的cvs版本控制迁移，最近一次提交（之前只有 3 年）是从 2011 年开始的，删除了该项目有维护者的提法。

建议所有剩余mcrypt的用户将其代码移植到OpenSSL，因为mcrypt不会添加到 AL2 023。

OpenJDK 7 () java-1.7.0-openjdk

> **Note**
>
> AL2023 提供了多个版本的 Amazon Corretto 来支持基于的工作负载。JavaOpenJDK 7 软件包已 AL2在中弃用，在 023 中不再存在。 AL2Corretto 8 提供了 AL2 023 年最古老的 JDK。

有关亚马逊 Linux 上的 Java 的更多信息，请参阅Java在 AL2 023。

Python 2.7

> **Note**
>
> AL2023 删除了 Python 2.7，因此任何需要 Python 的操作系统组件都是为了与 Python 3 配合使用而编写的。要继续使用 Amazon Linux 提供并支持的 Python 版本，请将 Python 2 代码转换为 Python 3。

有关亚马逊 Linux 上的 Python 的更多信息，请参阅Python 在 AL2 023。

rsyslog-openssl取代 rsyslog-gnutls

该rsyslog-gnutls软件包已在中弃用 AL2，在 AL2 023 中不再存在。该rsyslog-openssl软件包应可直接替换任何用途。rsyslog-gnutls

网络信息服务 (NIS)/yp

网络信息服务 (NIS)，最初名YP为黄页或已在中弃用 AL2，在 AL2 023 年不再存在。这包括以下软件包：ypbindypserv、和yp-tools。与之集成的其他软件包NIS已在 AL2 023 中删除了此功能。

亚马逊 VPC 中有多个域名 create-dhcp-options

在 Amazon Linux 2 中，可以在domain-name参数中将多个域名传递给 create-dhcp-options，这将导致/etc/resolv.conf包含类似的内容search foo.example.com bar.example.com。Amazon VPC DHCP 服务器使用DHCP选项 15 发送所提供的域名列表，该选项仅支持单个域名（参见 RFC 2132 第 3.17 节）。由systemd-networkd于 AL2 023 使用网络配置（紧随其后）RFC，因此 023 中不 AL2 存在此意外功能 AL2

AWS CLI和 Amazon VPC 文档这样说："某些 Linux 操作系统接受多个用空格分隔的域名。但是 Windows，其他 Linux 操作系统将该值视为单个域，这会导致意外行为。如果您的DHCP选项集与一个 Amazon VPC 相关联，而该实例运行的操作系统将该值视为单个域，则仅指定一个域名。"

在这些系统（例如 AL2 023）上，使用DHCP选项 15（仅允许一个）指定两个域，并且由于域名中的空格字符无效，这将导致空格字符被编码为032，从而导致包含。/etc/resolv.conf search foo.exmple.com032bar.example.com

为了支持多个域名，DHCP服务器应使用DHCP选项 119（参见 RFC 3397，第 2 节）。有关亚马逊 VPC DHCP 服务器何时支持此功能，请参阅亚马逊 VPC 用户指南。

Sun RPC中的glibc

in 的实现已Sun RPC在 023 中glibc被弃用 AL2 并在 AL2 023 中移除。如果需要Sun RPC功能，建议客户转而使用该libtirpc库（在 AL2 和 AL2 023 中提供）。采用libtirpc还可以支持应用程序 IPv6。

这一变化反映了更广泛的社区对上游glibc删除此功能的采用，例如，从 Fedora glibc 中删除了Sun RPC接口，Gentoo 也进行了类似的更改。

日志中有 OpenSSH 密钥指纹 `audit`

在生命周期的后期 AL2，在 OpenSSH 包中添加了一个补丁，用于发出用于身份验证的密钥指纹。AL2023 中不存在此功能。

`ld.gold`Linker

该`ld.gold`链接器在中可用 AL2，并于 AL2 023 年移除。开发明确引用gold链接器的软件的客户应迁移到常规 (`ld.bfd`) 链接器。

上游的 GNU Binutils 版本2.44（2025年2月发布）的发行说明记录了以下内容的删除`ld.gold`：“与之前的做法不同，在本版本中，binutils-2.44.tar压缩包不包含金链接器的来源。这是因为金链接器现已被弃用，除非志愿者挺身而出并主动提出继续开发和维护，否则最终将被删除。”

ping6

在 AL2 023 中，常规`ping`实用程序原生支持 IPv6，不再需要`/bin/ping6`分开。在 AL2 023 中，`/usr/sbin/ping6`是指向可执行`/usr/bin/ping`文件的符号链接。

这一变化是在更广泛的社区采用提供此功能的较新`iputils`版本之后进行的，例如 Fedora 中的 Ping IPv6 更改。

2023 年 AL2已被弃用

本节介绍了 AL2 023 中存在的功能，并且很可能会在未来版本的 Amazon Linux 中移除这些功能。每个部分都将描述该功能是什么，以及预计何时将其从 Amazon Linux 中删除。

> **Note**
>
> 随着 Linux 生态系统的发展以及 Amazon Linux 的未来主要版本即将发布，本节将随着时间的推移而更新。

主题

- 32 位 x86 (i686) 运行时支持
- aspell
- 伯克利数据库 (libdb)

- cron
- IMDSv1
- pcre 版本 1
- System V init (sysvinit)
- 已弃用 EOL 软件包

32 位 x86 (i686) 运行时支持

AL2023 保留了运行 32 位 x86 (i686) 二进制文件的功能。Amazon Linux 的下一个主要版本很可能不再支持运行 32 位用户空间二进制文件。

aspell

虽然 AL2 023 随`aspell`软件包一起提供，但它已被弃用，并将在下一个主要版本的 Amazon Linux 中删除。建议客户迁移到现代替代品，例如`hunspell`或`enchant2`。

AL2023 年的弃用`aspell`是在更广泛的社区转变之后发生的，例如中的`aspell弃用`。Fedora

伯克利数据库 (libdb)

AL2023 附带了 Berkeley DB () `libdb` 库的 5.3.28 版本。这是伯克利数据库在许可证从限制较少的 Sleepycat 许可证更改为 GNU Affero GPLv3 (AGPL) 许可证之前的最后一个版本。

AL2023 年很少有软件包仍然依赖于 Berkeley DB (`libdb`)，该库将在下一个主要版本的 Amazon Linux 中删除。

> **Note**
>
> AL2023 中的`dnf`包管理器保留了对 Berkeley DB (BDB) 格式`rpm`数据库的只读支持。此支持将在下一个主要版本的 Amazon Linux 中删除。

的弃用`libdb`是在更广泛的社区偏离它之后发生的，例如中的`libdb弃用`。Fedora

cron

该`cronie`软件包默认安装在 AL2 AMI 上，为计划定期任务的传统`crontab`方式提供支持。在 AL2 023 中，`cronie`默认不包括在内。因此，默认`crontab`情况下不再提供对的支持。

在 AL2 023 中，您可以选择安装cronie软件包以使用经典cron作业。由于 systemd 提供附加功能，建议您迁移到 systemd 计时器。

未来版本的 Amazon Linux（可能是下一个主要版本）可能会不再支持经典cron作业，而是完成向systemd计时器的过渡。我们建议您停止使用cron。

IMDSv1

默认情况下， AL2023 配置 AMIs 为在IMDSv2仅限模式下启动，从而禁用。IMDSv1仍然可以选择在 IMDSv1 启用后使用 AL2 023。未来版本的 Amazon Linux 可能IMDSv2只强制执行。

有关 IMDS 配置的更多信息 AMIs，请参阅 Amazon EC2 用户指南中的配置 AMI。

pcre 版本 1

旧版pcre软件包已被弃用，并将在 Amazon Linux 的下一个主要版本中移除。pcre2 软件包是后继版本。尽管 AL2 023 的第一个版本附带的软件包数量有限pcre，但这些软件包将迁移到 AL2 023 以pcre2内。已弃用的pcre库将在 AL2 2023 年继续可用。

> **Note**
>
> 已弃用版本在 AL2 0 pcre 23 的整个生命周期内都不会收到安全更新。有关pcre支持生命周期和软件包接收安全更新的时间长度的更多信息，请参阅软件包上的软件pcre包支持声明。

不赞成使用pcre2是在更广泛的社区朝着这个方向转变之后发生的，例如pcre弃用。pcre Fedora

System V init (sysvinit)

尽管 AL2 023 保留了与 s System V ervice (init) 脚本的向后兼容性，但作为其 v254 版本的一部分，上游systemd项目宣布不再支持 System V 服务脚本，并表示将在的未来版本中取消支持。systemd有关更多信息，请参阅 systemd。

AL2023 将保持与 s System V ervice (init) 脚本的向后兼容性，但我们鼓励用户迁移到使用原生systemd单元文件，以便为可能在下一个主要版本中从 Amazon Linux 中删除对System V服务 (init) 脚本的支持做好准备。

已弃用 EOL 软件包

AL2023 中提供的每个软件包都有一个相关的支持声明，其中涵盖了 Amazon Linux 的特定信息。这些陈述涵盖了操作系统的核心及其生命周期，以及诸如the section called “PHP”和之类的软件包the

section called “Python”，其中 AL2 023 提供了多个版本，并且在上游开源项目的持续时间内支持每个版本。

在 AL2 023 中，您可以使用软件包管理器获取dnf软件包支持信息。有关更多信息，请参阅 获取软件包支持信息。

如果在 Amazon Linux 主版本结束之前不再支持某个软件包，则应假设该软件包已被弃用，并且不会出现在下一个主要版本的 Amazon Linux 中。

对于诸如the section called “PHP”和之类的软件包the section called “Python”，每个主要的 Amazon Linux 版本都提供了多个版本，每个版本的支持生命周期都不同，因此它们很可能会继续出现在新的主要版本的 Amazon Linux 中，尽管与主要版本的软件包几乎没有重叠。建议在选择依赖项时记住 Amazon Linux 软件包的支持时间表。

比较 AL2 和 AL2 023

以下主题描述了 AL2 和 AL2 023 之间的主要区别。

有关、和 AL2 023 中已弃用的功能的更多信息 AL1 AL2，请参阅。2023 年 AL2已弃用的功能

主题

- 添加、升级和删除了软件包
- 针对每个版本的支持
- 命名和版本控制更改
- 优化
- 来自多个上游
- 网络系统服务
- 软件包管理器
- 使用 cloud-init
- 图形桌面支持
- 编译器三元组
- 32 位 x86 (i686) 软件包
- lsb_release 和 system-lsb-core 软件包
- Extra Packages for Enterprise Linux (EPEL)
- Python 2.7 已被 Python 3 所取代
- 安全更新
- 确定性升级用于提高稳定性
- gp3作为默认 Amazon EBS 卷类型
- 统一控制组层次结构 (cgroup v2)
- systemd计时器替换 cron
- 改进的工具链：gcc, binutils，以及 glibc
- systemd日记替换 rsyslog
- 最大限度地减少软件包依赖性
- Amazon Corretto 作为默认 JVM

- AWS CLI v2
- UEFI 首选和安全启动
- SSH 服务器默认配置更改
- AL2023 个内核变更自 AL2
- /tmp 现在是 tmpfs
- AMI 和容器镜像变更
- 比较亚马逊 Linux 2 和亚马逊 Linux 2023 上安装的软件包 AMIs
- 比较亚马逊 Linux 2 和亚马逊 Linux 2023 上安装的软件包 Minimal AMIs
- 比较 Amazon Linux 2 和 Amazon Linux 2023 基础容器映像上安装的软件包

添加、升级和删除了软件包

AL2023 包含数千个可供使用的软件包。有关 023 年新增、升级或删除的所有软件包与之前的 Amazon Linux 版本对比的完整列表，请参阅 AL2 023 年 AL2的 Package 变更。

要申请在 AL2 023 年添加或更改软件包，请在亚马逊 linux- 2023 存储库中提交问题。 GitHub

针对每个版本的支持

对于 AL2 023，我们提供五年的支持。

有关更多信息，请参阅 发布频率。

命名和版本控制更改

AL2023 支持的机制与支持平台识别的 AL2 机制相同。 AL2023 还引入了用于平台识别的新文件。

有关更多信息，请参阅 命名和版本控制。

优化

AL2023 优化了启动时间，以缩短从实例启动到运行客户工作负载的时间。这些优化涵盖了 Amazon EC2 实例内核`cloud-init`配置、配置和内置于操作系统软件包（如`kmod`和`systemd`）中的功能。

有关优化的更多信息，请参阅性能和操作优化。

来自多个上游

AL2023 基于 RPM，包括来自多个版本的 Fedora 和其他发行版（例如 CentOS 9 Stream）的组件。Amazon Linux 内核直接源自来自 kernel.org 的长期支持 (LTS) 版本，该版本是独立于其他发行版选择的。

有关更多信息，请参阅 与 Fedora 的关系。

网络系统服务

`systemd-networkd`系统服务在 AL2 023 中管理网络接口。这是改为 AL2，后者使用 ISC dhclient 或 `dhclient`) 的请求。

有关更多信息，请参阅 网络服务。

软件包管理器

AL2023 上的默认软件包管理工具是 DNF. DNF 是的继任者 YUM，中的软件包管理工具 AL2。

有关更多信息，请参阅 软件包管理工具。

使用 cloud-init

在 AL2 023 年，cloud-init 管理软件包存储库。默认情况下，在早期版本的亚马逊 Linux 中，cloud-init 已安装安全更新。这不是 AL2 023 的默认设置。用于在启动时更新的确定性升级功能描述了 AL2 023 `releasever` 在启动时启用软件包更新的方法。有关更多信息，请参阅在 AL2 023 中管理软件包和操作系统更新和确定性升级用于提高稳定性。

使用 AL2 023，你可以使用 cloud-init 替换为 SELinux。 有关更多信息，请参阅使用 cloud-init 启用`enforcing`模式。

Cloud-init 使用加载配置内容 cloud-init 从远程位置使用 HTTP(S)。 在早期版本中，当远程资源不可用时，Amazon Linux 不会提醒您。在 AL2 023 中，不可用的远程资源会导致严重错误并失败 cloud-init 处决。此行为更改从 AL2，提供了更安全的 “失效关闭” 默认行为。

有关更多信息，请参阅定制 cloud-init和 cloud-init 文档。

图形桌面支持

AL2自 2023.7 版本起，023 采用基于 GNOME 的图形桌面环境，取代了中使用的 MATE 桌面。 AL2 此版本为用户提供了不同的桌面体验，同时保持了 AL2 023 的云优化性能。GNOME 桌面环境提供了各种自定义选项、系统集成功能和独特的用户界面设计，为用户提供了以前的 MATE 桌面环境的替代方案。有关更多详细信息，请参阅AL2023 图形化桌面页面。

编译器三元组

AL2023 将编译器三元组设置为 GCC `LLVM`并表明那`amazon`是供应商。

因此， AL2 `aarch64-redhat-linux-gcc`变成 `aarch64-amazon-linux-gcc` AL2 023。

对于大多数用户来说，这应该是完全透明的，并且可能只会影响那些在 AL2 023 上构建编译器的用户。

32 位 x86 (i686) 软件包

作为 2014.09版本的一部分， AL1它宣布这将是最后一个生产32位的版本。 AMIs因此，从 2015.03 版本起， AL1Amazon Linux 不再支持在 32 位模式下运行系统。 AL2 为 x86-64 主机上的 32 位二进制文件提供了有限的运行时支持，并且没有提供支持构建新 32 位二进制文件的开发包。 AL2023 不再包含任何 32 位用户空间软件包。我们建议您完成向 64 位代码的过渡。

如果您需要在 023 上运行 32 位二进制文件，则可以从 AL2 023 之上运行的 AL2 容器 AL2 内部使用 32 位用户空间。 AL2

`lsb_release` 和 `system-lsb-core` 软件包

过去，一些软件会调用该`lsb_release`命令（ AL2 由软件`system-lsb-core`包提供）来获取有关其运行的 Linux 发行版的信息。Linux Standards Base (LSB) 引入了此命令，Linux 发行版采用了此命令。Linux 发行版已经演变为使用更简单的标准，将这些信息保存在 `/etc/os-release` 和其他相关文件中。

`os-release` 标准来自 `systemd`。有关更多信息，请参阅 systemd os-release 文档。

AL2023 不附带该`lsb_release`命令，也不包括`system-lsb-core`软件包。软件应完成向 `os-release` 标准的过渡，以保持与 Amazon Linux 和其他主要 Linux 发行版的兼容性。

Extra Packages for Enterprise Linux (EPEL)

> **Warning**
>
> E AL2 `epel xtra` 启用了第三方 EPEL7 存储库。截至 2024 年 6 月 30 日，第三方 EPEL7 存储库已不再维护。
> 此第三方存储库将不会有 future 更新。这意味着 EPEL 存储库中的软件包不会有安全补丁。
> 本节将介绍 AL2 023 中某些软件包的选项 EPEL.

Extra Packages for Enterprise Linux (EPEL) 是中的一个项目 Fedora 社区，目标是为企业级 Linux 操作系统创建大量软件包。该项目主要产生了 RHEL 以及 CentOS 包裹。 AL2 具有高度的兼容性 CentOS 7。 结果，很多 EPEL7 软件包可以正常工作 AL2。

目前没有 EPEL 或 EPEL类似于 AL2 023 的存储库。但是，里面有许多包裹 EPEL7 客户使用过的要么在 023 年上 AL2 市，要么在 AL2 023 年 AL2有替代品。本节将介绍其中一些软件包以及 AL2 023 中的选项。

> **Warning**
>
> 仅添加设计用于 AL2 023 的存储库。
> 虽然为其他发行版设计的存储库现在可以使用，但不能保证它们会继续使用 023 中的任何软件包更新或不是为 AL2 023 设计的存储库。 AL2

还有一些软件包可以从中安装 EPEL 不会 AL2 将其添加到 AL2 023 中。造成这种情况的常见原因是诸如上游项目不再维护或无法修复之类的问题 CVEs。本节还将介绍其中一些软件包以及存在哪些替代方案。

主题

- axel-HTTP/FTP 客户端
- brotli和 libbrotli-压缩
- collectd-统计数据收集守护程序
- cpulimit-CPU 使用限制器
- exim-邮件传输代理

- fuse3-用户空间中的文件系统 (FUSE) v3
- ganglia-分布式监控系统
- git-lfs-使用 Git 控制大文件的版本
- haveged-使用熵源 HAVEGE algorithm
- inotify-tools-inotify 命令行工具
- iperf-TCP/UDP 性能基准测试
- jemalloc-替代malloc实现
- libbsd-兼容 BSD 的函数库
- libserf-HTTP 客户端库
- libzstd-zstd 压缩库
- lighttpd网络服务器
- lshell-受限炮弹
- monit-进程、文件、目录和设备监视器
- nodejs
- perl-Config-General
- python2-lockfile-文件锁定
- python2-rsa-纯粹的 Python RSA
- python2-simplejson-适用于 Python 2 的 JSON 例程
- rkhunter-rootkit Hunter
- rssh-适用于 OpenSSH 的受限外壳
- sscg-自签名 SSL 证书生成器
- stress-压力测试
- stress-ng-压力测试
- tmpwatch-根据上次访问时间删除文件
- xmlstarlet-命令行 XML 实用工具

`axel`-HTTP/FTP 客户端

`axel`包裹在里面 EPEL7，而且从未作为亚马逊 Linux 的一部分发货。 AL2023 中可用的替代方案是`curl`和。`wget`

> **Warning**
>
> axel使用未加密的-S选项 http 连接以发现文件的镜像。

强烈建议将 over 的任何用法迁axel移到curl或wget。

brotli和 libbrotli-压缩

brotli和libbrotli包裹在里面 EPEL7，而 AL2 核心版本中只有该brotli软件包可用。

brotli和libbrotli软件包都包含在 AL2 023 中。

可以使用以下命令在 AL2 023 上安装该brotli软件包：

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

可以使用以下命令在 AL2 023 上安装该libbrotli软件包：

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

collectd-统计数据收集守护程序

collect包裹在里面 EPEL7，并且还在 E collectd-python3 AL2 xtras collectd 中提供。

该collectd软件包包含在 AL2 023 中，可以通过运行以下命令进行安装：

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

cpulimit-CPU 使用限制器

在 Amazon Linux 2023 中，systemd提供了限制进程或进程组的 CPU 使用率的功能。此功能也易于用于任何systemd服务。

提供了强大的资源控制工具，可以systemd用来确保任何任务或一组任务在消耗的资源上受到限制。有关更多信息，请参阅上游 systemd.resource-control 文档以及。使用限制 AL2 023 中的进程资源使用量 systemd

exim-邮件传输代理

exim包裹在里面 EPEL7，之前已在 AL1。亚马逊 Linux 2023 同时提供postfix和sendmail邮件传输代理 (MTAs)。

fuse3-用户空间中的文件系统 (FUSE) v3

fuse3包裹（包括fuse3-libs和fuse3-devel）在里面 EPEL7。 这些软件包是 AL2 023 的一部分，每个软件包都可以通过运行以下相关命令进行安装：

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

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

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

ganglia-分布式监控系统

ganglia包裹在里面 EPEL7，之前已在 AL1。它不是随附的 AL2。

上游项目有一段不活跃的时期，有些未解决的问题没有 CVEs得到解决。尽管上游项目最近有一些活动，但不计划将其增加ganglia到 AL2 023。

git-lfs-使用 Git 控制大文件的版本

git-lfs包裹在里面 EPEL7。 在亚马逊 Linux 2023 中，该git-lfs软件包包含在核心存储库中。在 AL2 023 上，git-lfs可以通过运行以下命令进行安装：

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

haveged-使用熵源 HAVEGE algorithm

haveged包裹在里面 EPEL7。 亚马逊 Linux 2023 预先配置了熵源，不需要使用。haveged

inotify-tools-inotify 命令行工具

inotify-tools包裹在里面 EPEL7，并包含在 AL2 023 中。

> **Note**
>
> 在 AL2 023 中，systemd支持基于路径的激活，可用于对事件（例如路径存在或更改时）采取行动。
> 现在inotify-tools，使用systemd路径激活可以以更可靠的方式更好地完成大部分用途。有关更多信息，请参阅 systemd.path。

该inotify-tools软件包包含在 AL2 023 中，可以通过运行以下命令进行安装：

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

iperf-TCP/UDP 性能基准测试

iperf版本 2 软件包已在 EPEL7，还在 Extra testing AL2 . 中可用，也有 AL1

> **Note**
>
> 该iperf3软件包也可用，提供的版本为 3 iperf。

该iperf软件包包含在 AL2 023 中，可以通过运行以下命令进行安装：

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

jemalloc-替代malloc实现

jemalloc包裹在里面 EPEL7，并在 Ext mariadb10.5 AL2 ras lamp-mariadb10.2-php7.2 中可用。

该jemalloc软件包包含在 AL2 023 中，可以通过运行以下命令进行安装：

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

libbsd-兼容 BSD 的函数库

libbsd包裹在里面 EPEL7，并且还在 E testing AL2 xtra 中可用。

该libbsd软件包包含在 AL2 023 中，可以通过运行以下命令进行安装：

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

的开发文件libbsd可以通过运行以下命令来安装。

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

libserf-HTTP 客户端库

libserf包裹在里面 EPEL7。 该libserf软件包在亚马逊 Linux 2023 中提供。可以通过运行以下命令进行安装：

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

libzstd-zstd 压缩库

该libzstd软件包在 AL2 核心中，也在 EPEL7。 该libzstd软件包也是 AL2 023 的一部分。

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

lighttpd网络服务器

lighttpd包裹在里面 EPEL7，之前已在 AL1。亚马逊 Linux 2023 同时提供 Apache httpd 和nginx网络服务器。

lshell-受限炮弹

该lshell包裹从未作为亚马逊 Linux 的一部分发货。它可用于 EPEL6。 Fedora 打包存储库，lshell其中包含了为什么它没有被打包进去 EPEL7 或者 Fedora 30。它也已从 Debian 中删除。

上游lshell项目已停止积极维护，并且包含已知的未修补的 Critical CVEs：CVE-2016-6902 和 CVE-2016-6903。

Debian 错误中建议的替代方案rssh也未在上游维护，作者列举了无法修复的安全问题作为原因。

出于这些原因，没有计划添加lshell到 AL2 023。

monit-进程、文件、目录和设备监视器

在 Amazon Linux 2023 中，systemd提供了一系列用于监控、启动、停止和重启服务的功能。这包括限制重启的速率、在重启尝试之间等待，以及在失败时启动其他服务。有关更多信息，请参阅 systemd.service 文档。

在 AL2 023 中，systemd还支持基于路径的激活，可用于对事件（例如路径存在或更改时）采取行动。有关更多信息，请参阅 systemd.path。

systemd单元有常见的配置选项，允许指定依赖关系、条件以及成功或失败时要采取的操作。有关更多信息，请参阅 systemd.unit 文档。

提供了强大的资源控制工具，可用来systemd确保任何监视任务都不会占用过多的 CPU 或内存。有关更多信息，请参阅 system.r esource-control。

nodejs

nodejs版本 16 软件包已在 EPEL7，nodejs现在已包含在 AL2 023 中。在撰写本文时，nodejs版本 18 和 20 都已在 AL2 023 年推出。你可以使用以下命令在 AL2 023 上安装 nodejs 18：

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

你可以使用以下命令在 AL2 023 上安装 nodejs 20：

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

perl-Config-General

perl-Config-General包裹在里面 EPEL7，现在已包含在 AL2 023 中。你可以使用以下命令在 AL2 023 中安装perl-Config-General软件包：

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

也可以通过询问来安装 Perl 模块 DNF 安装提供特定 Perl 模块的软件包。使用此方法，您可以使用更熟悉的 Perl 模块名称而不是操作系统包名称。

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

python2-lockfile-文件锁定

python2-lockfile包裹在里面 EPEL7，并 AL2 包括一个python-lockfile包裹。在 AL2 023 中Python 2.7 已被 Python 3 所取代，因此该软件包的 Python 2 变体不会添加到 AL2 023 中。

此软件包的 Python 3 版本包含在 AL2 023 中。您可以使用以下命令之一在 AL2 023 中安装该python3-lockfile软件包：

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

也可以通过询问来安装 Python 模块 DNF 来安装提供特定 Python 模块的软件包。

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

python2-rsa-纯粹的 Python RSA

python2-rsa包裹在里面 EPEL7，并 AL2 包括一个python2-rsa包裹。在 AL2 023 中Python 2.7 已被 Python 3 所取代，因此该软件包的 Python 2 变体不会添加到 AL2 023 中。

此软件包的 Python 3 版本包含在 AL2 023 中。您可以使用以下命令之一在 AL2 023 中安装该python3-rsa软件包：

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

也可以通过询问来安装 Python 模块 DNF 来安装提供特定 Python 模块的软件包。

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

python2-simplejson-适用于 Python 2 的 JSON 例程

python2-simplejson包裹在里面 EPEL7。 在 AL2 023 中Python 2.7 已被 Python 3 所取代，因此该软件包的 Python 2 变体不会添加到 AL2 023 中。

此软件包的 Python 3 版本包含在 AL2 023 中。你可以使用以下命令在 AL2 023 中安装python3-simplejson软件包：

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

也可以通过询问来安装 Python 模块 DNF 来安装提供特定 Python 模块的软件包。

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

rkhunter-rootkit Hunter

该rkhunter软件包随附在 AL2 023 中。chkrootkit

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

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

rssh-适用于 OpenSSH 的受限外壳

rssh包裹在里面 EPEL7。 上游rssh软件包无法维护，作者列举了无法修复的安全问题作为原因。

由于作者列举了无法修复的安全问题，因此没有计划添加rssh到 AL2 023。

sscg-自签名 SSL 证书生成器

该sscg软件包在 AL2 核心中，也在 EPEL7。 该sscg软件包也是 AL2 023 的一部分。

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

stress-压力测试

stress包裹在里面 EPEL7，并且还提供了 AL1

该stress软件包包含在 AL2 023 中，可以通过运行以下命令进行安装：

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

stress-ng-压力测试

stress-ng包裹在里面 EPEL7，并且还在 E testing AL2 xtra 中可用。

该stress-ng软件包包含在 AL2 023 中，可以通过运行以下命令进行安装：

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

tmpwatch-根据上次访问时间删除文件

在亚马逊 Linux 2023 中，此功能由提供systemd-tmpfiles。

xmlstarlet-命令行 XML 实用工具

xmlstarlet包裹在里面 EPEL7，并且在 AL2 023 中不可用。

上游软件包已经有9年多没有被触及了（最后一次触及是在2014年8月）。在过去的四年中（至少从 2010 年 7 月开始），对新维护者的请求一直没有得到答复。正是出于这个原因，不打算将其添加xmlstarlet到 AL2 023。

Python 2.7 已被 Python 3 所取代

AL2 在 2025 年 6 月之前为 Python 2.7 提供支持和安全补丁，这是我们对 AL2 核心包的长期支持 (LTS) 承诺的一部分。这种支持不仅限于上游 Python 社区在 2020 年 1 月发布的 Python 2.7 end-of-life 声明。

AL2 使用yum包管理器，它对 Python 2.7 有硬依赖性。在 AL2 023 中，dnf包管理器已迁移到 Python 3，不再需要 Python 2.7。 AL2023 已完全移至 Python 3。

> **Note**
>
> AL2023 删除了 Python 2.7，因此任何需要 Python 的操作系统组件都是为了与 Python 3 配合使用而编写的。要继续使用 Amazon Linux 提供并支持的 Python 版本，请将 Python 2 代码转换为 Python 3。

有关 Amazon Linux 上的 Python 的更多信息，请参阅 Python 在 AL2 023。

安全更新

亚马逊 Linux 2023 在中存在的强化基础上进行了改进。 AL2有关更多信息，请参阅 Amazon Linux 2023 中的安全性与合规性。有关中内核强化更改的更多信息 AL2，请参阅注重安全的内核配置变化。

主题

- SELinux
- OpenSSL 3
- IMDSv2
- 移除 log4j 热补丁 () log4j-cve-2021-44228-hotpatch

SELinux

默认情况下，Security Enhanced Linux (SELinux) fo AL2 r 023 是`enabled`并设置为`permissive`模式。在 `permissive` 模式下，记录但不强制执行权限拒绝。

SELinux 是 Amazon Linux 内核的一项安全功能，该内核曾`disabled`在 AL2。SELinux 是提供强制访问控制的内核功能和实用程序的集合 (MAC) 架构融入内核的主要子系统。

有关更多信息，请参阅 为 AL2 0 SELinux 23 设置模式。

有关 SELinux 存储库、工具和策略，请参阅SELinux 笔记本、 SELinux 策略类型和SELinux 项目。

OpenSSL 3

AL2023 的特点是 Open Secure Sockets Layer version 3 (OpenSSL 3) 密码学工具包。 AL2023 支持 TLS 1.3 以及 TLS 1.2 网络协议。

默认情况下， AL2 附带 OpenSSL 1.0.2。 您可以针对构建应用程序 OpenSSL 1.1.1.

有关 OpenSSL，请参阅 OpenSSL 迁移指南。

有关安全性的更多信息，请参阅安全更新和功能。

IMDSv2

默认情况下，任何使用 AL2 023 AMI 启动的实例都需要 IMDSv2-only，您的默认跳数限制将设置为 2，以支持容器化工作负载。这可通过将 `imds-support` 参数设置为 `v2.0` 来完成。有关更多信息，请参阅 Amazon EC2 用户指南中的配置 AMI。

> **Note**
>
> 会话令牌的有效时间可以介于 1 秒到 6 小时之间。指挥的地址 API 请求 IMDSv2 查询如下：

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

您可以手动覆盖这些设置并启用 IMDSv1 使用实例元数据选项启动属性。您还可以使用 IAM 控制来强制执行不同的 IMDS 设置。有关设置和使用实例元数据服务的更多信息，请参阅使用 IMDSv2、Amazon EC2 用户指南中的为新实例配置实例元数据选项和修改现有实例的实例元数据选项。

移除 log4j 热补丁 () `log4j-cve-2021-44228-hotpatch`

> **Note**
>
> AL2023 不随`log4j-cve-2021-44228-hotpatch`包裹一起发货。

为了回应 CVE-2021-44228，亚马逊 Linux 发布了适用于 Ap ache Log4j 的 Hotpatch 的 RPM 打包版本，适用于和。 AL1 AL2在宣布向 Amazon Linux 添加热补丁时，我们指出："安装热补丁并不能取代更新到可缓解 CVE-2021-44228 或 CVE-2021-45046 的 log4j 版本。"

热补丁是一种缓解措施，可以留出时间来修补 `log4j`。 AL2023 的第一个通用可用性 (GA) 版本是 CVE-2021-44228 发布的 15 个月后，因此 AL2 023 不附带热补丁（无论是否启用）。

在 Amazon Linux 上运行自己的`log4j`版本的用户应确保已更新到未受 CVE-2021-44228 或 CVE-2021-45046 影响的版本。

AL2023 提供了相关指导，正在更新 AL2 023以便您可以及时了解最新的安全补丁。安全公告发布在 Amazon Linux 安全中心上。

确定性升级用于提高稳定性

借助通过版本控制存储库进行确定性升级功能，默认情况下，每个 AL2 023 AMI 都锁定到特定的存储库版本。您可以使用确定性升级来提高软件包版本和更新之间的一致性。每个版本，无论是主要版本还是次要版本，都包含一个特定的存储库版本。

AL2023 的新增功能，默认启用确定性升级。这是对和其他早期版本中 AL2 使用的手动增量锁定方法的改进。

有关更多信息，请参阅 023 年通过版本化存储库进行确定性升级 AL2。

gp3作为默认 Amazon EBS 卷类型

AL2023 AMI 和 AL2 两者都使用 XFS 根文件系统上的文件系统。对于 AL2 023，根设备文件系统的`mkfs`选项已针对 Amazon EC2 进行了进一步优化。 AL2023 还支持许多其他文件系统，您可以在其他卷上使用这些文件系统来满足您的特定要求。

AL2023 默认 AMIs 使用亚马逊 EBS gp3 卷，而默认使用 AL2 AMIs 亚马逊 EBS gp2 卷。您可以在启动实例时更改卷类型。

有关 Amazon EBS 卷类型的更多信息，请参阅 Amazon EBS 通用型卷。

有关启动亚马逊 EC2 实例的更多信息，请参阅亚马逊 EC2 用户指南中的启动实例。

统一控制组层次结构 (cgroup v2)

对照组 (cgroup) 是一项 Linux 内核功能，用于分层组织进程并在进程之间分配系统资源。控制组被广泛用于通过 `systemd` 实现容器运行时系统。

AL2 支持 cgroupv1，还有 AL2 023 支持 cgroupv2。 如果运行容器化工作负载（例如何时），这一点值得注意。使用基于 AL2 023 的 Amazon ECS AMIs 托管容器化工作负载

尽管 AL2 023 仍然包含可以让系统运行的代码 cgroupv1，这不是推荐或支持的配置，并且将在未来的 Amazon Linux 主要版本中完全删除。

有大量关于低级 Linux 内核接口的文档以及 systemd cgroup 委派文档。

容器之外的一个常见用例是创建对可以使用的系统资源有限制的`systemd`单元。有关更多信息，请参阅 system.r esource-control。

`systemd`计时器替换 `cron`

默认情况下，该`cronie`软件包安装在 AL2 AMI 上，为定期任务的传统计划`crontab`方式提供支持。在 AL2 023 中，`cronie`默认不包括在内。因此，默认`crontab`情况下不再提供对的支持。

您可以选择安装 `cronie` 软件包以使用经典的 `cron` 作业。由于 `systemd` 提供附加功能，建议您迁移到 `systemd` 计时器。

改进的工具链：gcc, binutils，以及 glibc

AL2023 包含许多与. 相同的核心软件包。 AL2

我们为 AL2 023 更新了以下三个核心工具链包。

程序包名称	AL2	AL2023
glibc	2.26	2.34
gcc	7.3	11.3
binutils	2.29	2.39

有关更多信息，请参阅 核心工具链包 glibc, gcc, binutils。

有关优化的更多信息，请参阅性能和操作优化。

systemd日记替换 rsyslog

在 AL2 023 中，日志系统包已更改为。 AL2 AL2默认情况下，023 不会安装，因此rsyslog默认情况下，基于文本/var/log/messages的日志文件（例如中提供的文件） AL2 不可用。 AL2023 的默认配置是systemd-journal，可以使用进行检查。journalctl尽管在 AL2 023 中rsyslog是可选的软件包，但我们建议使用systemd基于新的journalctl接口和相关软件包。有关更多信息，请参阅 journalctl 手册页面。

这些区域有：systmed journal 相当于一些常用的 syslog 下表涵盖了命令。

AL2 syslog 命令	AL2023 systemd journal 相等物
`[ec2-user ~]$ cat /var/log/messages`	`[ec2-user ~]$ journalctl`
`[ec2-user ~]$ tail -f /var/log/messages`	`[ec2-user ~]$ journalctl -f`
`[ec2-user ~]$ grep foo /var/log/messages`	`[ec2-user ~]$ journalctl | grep foo`

最大限度地减少软件包依赖性

Amazon Linux 2023 最大限度地减少了许多软件包的依赖关系图，从而减少了应用程序的占用空间。与之相比，值得注意的变化 AL2 包括`curl-minimal`和`gnupg-minimal`软件包，它们在保留常用功能的同时，显著减少了所需软件包的数量。

主题

- curl 和 libcurl 的软件包变更
- GNU Privacy Guard (GNUPG)

`curl` 和 `libcurl` 的软件包变更

AL2023 将和的常用协议和功能分离出来，`libcurl`打包成`curl``curl-minimal`和。`libcurl-minimal`这减少了大多数用户的磁盘、内存和依赖占用空间，并且是 AL2 023 AMIs 和容器的默认软件包。

如果需要 `curl` 的全部功能（例如为了获得 `gopher://` 支持），请运行以下命令来安装 `curl-full` 和 `libcurl-full` 软件包。

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

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

GNU Privacy Guard (GNUPG)

AL2023 将软件包的最小和完整功能分离到`gnupg2-minimal`和`gnupg2`打包中。`gnupg2-full`默认情况下仅安装 `gnupg2-minimal` 软件包。这提供了验证 `rpm` 软件包上的数字签名所需的最低功能。

要从 `gnupg2` 获得更多功能，例如从密钥服务器下载密钥，请确保已安装 `gnupg2-full` 软件包。运行以下命令，将 `gnupg2-minimal` 交换为 `gnupg2-full`。

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

Amazon Corretto 作为默认 JVM

AL2023 附带亚马逊 Corretto 作为默认（也是唯一的）Java 开发套件 (JDK)。全部 Java 023 中基于 AL2 023 的软件包都是用构建的 Amazon Corretto 17.

如果要从中迁移 AL2，则可以顺利地从等效`OpenJDK`版本过渡 AL2 到 Amazon Corretto.

AWS CLI v2

AL2023 附带 AWS CLI 版本 2，而 AL2 随版本 1 一起提供。 AWS CLI

UEFI 首选和安全启动

默认情况下，在支持 UEFI 固件的实例类型上使用 AL2 023 AMI 启动的任何实例都将在 UEFI 模式下启动。这是通过将启动模式 AMI 参数设置为 `uefi-preferred` 来完成的。有关更多信息，请参阅 Amazon EC2 用户指南中的启动模式。

在支持 UEFI 安全启动的亚马逊 EC2 实例类型上，可以在亚马逊 Linux 2023 中启用安全启动。有关更多信息，请参阅 UEFI 在 023 上安全启动 AL2。

SSH 服务器默认配置更改

对于 AL2 023 AMI，我们更改了随版本生成的`sshd`主机密钥的类型。我们还删除了一些旧密钥类型，以避免在启动时生成它们。客户端必须通过使用 `ed25519` 密钥支持 `rsa-sha2-256` 和 `rsa-sha2-512` 协议或 `ssh-ed25519`。默认情况下，`ssh-rsa` 签名处于禁用状态。

此外，默认`sshd_config`文件中的 AL2 023 配置设置包含`UseDNS=no`。这个新设置意味着 DNS 损伤不太可能阻碍您与实例建立`ssh`会话。但代价在于，`authorized_keys` 文件中的 `from=`*`hostname.domain`*`,`*`hostname.domain`* 行条目将无法解析。由于`sshd`不再尝试解析 DNS 名称，因此必须将每个以逗号分隔的*`hostname.domain`*值转换为相应的 IP address.

有关更多信息，请参阅 默认 SSH 服务器配置。

AL2023 个内核变更自 AL2

AL2023 引入了 6.1 内核以及许多配置更改，以进一步优化 Amazon Linux 的云端应用。对于大多数用户来说，这些更改应该是完全透明的。

IPv4 TTL

的 TTL IPv4 是通过配置的`sysctl`，默认值显示在中。`/etc/sysctl.d/00-defaults.conf`该值可以通过常用`sysctl`方法进行自定义。有关更多信息，请参阅`sysctl`man页面。

AL2 将该`net.ipv4.ip_default_ttl`值设置为 255，而 AL2 023 将其设置为 127。这使亚马逊 Linux 的默认设置与其他主要的 Linux 发行版保持一致。如果没有证明有必要，不建议更改此默认值。

注重安全的内核配置变化

CONFIG 选项	AL2/4.14/ aarch64	AL2/4.14/ x86_64	AL2/5.10/ aarch64	AL2/5.10/ x86_64	AL2023/6. 1/ aarch64	AL2023/6. 1/ x86_64	AL2023/6. 12/ aarch64	AL2023/6. 12/ x86_64
CONFIG_I G_ON_DA _CORRUP ON	n	y	n	y	y	y	y	y
CONFIG_I FAULT_MI P_MIN_AI R	4096	4096	4096	4096	65536	65536	65536	65536
CONFIG_I VMEM	n	y	n	y	n	n	n	n
CONFIG_I VPORT	n	y	n	y	n	n	n	n
CONFIG_I RTIFY_S RCE	n	y	n	y	y	y	y	y
CONFIG_I RDENED_ ERCOPY_ LLBACK	不适用	不适用	y	y	不适用	不适用	不适用	不适用
CONFIG_ IT_ON_AI OC_DEFA T_ON	不适用	不适用	n	n	n	n	n	n

CONFIG 选项	AL2/4.14/ aarch64	AL2/4.14/ x86_64	AL2/5.10/ aarch64	AL2/5.10/ x86_64	AL2023/6. 1/ aarch64	AL2023/6. 1/ x86_64	AL2023/6. 12/ aarch64	AL2023/6. 12/ x86_64
CONFIG_ IT_ON_FI E_DEFAUI _ON	不适用	不适用	n	n	n	n	n	n
CONFIG_ MMU_DEF/ LT_DMA_ RICT	不适用	不适用	不适用	不适用	n	n	n	n
CONFIG_ ISC_AUT OAD	y	y	y	y	n	n	n	n
CONFIG_ HED_CORI	不适用	不适用	不适用	不适用	不适用	y	不适用	y
CONFIG_ HED_STA _END_CHI K	n	y	n	y	y	y	y	y
CONFIG_ CURITY_ ESG_RES ICT	n	n	n	n	y	y	y	y
CONFIG_ CURITY_ LINUX_D ABLE	y	y	y	y	n	n	不适用	不适用

CONFIG 选项	AL2/4.14/ aarch64	AL2/4.14/ x86_64	AL2/5.10/ aarch64	AL2/5.10/ x86_64	AL2023/6. 1/ aarch64	AL2023/6 1/ x86_64	AL2023/6 12/ aarch64	AL2023/6. 12/ x86_64
CONFIG_S UFFLE_PA E_ALLOCA OR	不适用	不适用	y	y	y	y	y	y
CONFIG_S AB_FREEL ST_HARDE ED	n	y	y	y	y	y	y	y
CONFIG_S AB_FREEL ST_RANDO	n	n	y	y	y	y	y	y

x86-64 以特定安全为重点的内核配置更改

CONFIG 选项	AL2/4.14/x86_64	AL2/5.10/x86_64	AL2023/6.1/ x86_64	AL2023/6.12/ x86_64
CONFIG_AM D_IOMMU	y	y	y	y
CONFIG_AM D_IOMMU_V2	m	m	y	不适用
CONFIG_RA NDOMIZE_M EMORY	不适用	y	y	y

aarch64 (arm/Graviton) 以安全为重点的内核配置更改

CONFIG 选项	AL2/4.14/ aarch64	AL2/5.10/ aarch64	AL2023/6.1/ aarch64	AL2023/6.12/ aarch64
CONFIG_ARM64_PTR_AUTH	不适用	y	y	y
CONFIG_ARM64_PTR_AUTH_KERNEL	不适用	不适用	y	y
CONFIG_ARM64_SW_TTBR0_PAN	y	y	y	y

/dev/mem、/dev/kmem 和 /dev/port

Amazon Linux 2023 在已有的限制的基础上完全禁用/dev/memCONFIG_DEVMEM和/dev/port（而且CONFIG_DEVPORT）。 AL2

在 5.13 内核中，该/dev/kmem代码已从 Linux 中完全删除，虽然它在中被禁用 AL2，但现在不适用于 AL2 023。

此选项是内核自我保护项目推荐设置之一。

FORTIFY_SOURCE

AL2023 CONFIG_FORTIFY_SOURCE 在所有支持的架构上启用。此功能是一项安全强化功能。在编译器可以确定和验证缓冲区大小的情况下，此功能可以检测常见字符串和内存函数中的缓冲区溢出。

此选项是内核自我保护项目推荐设置之一。

线路纪律自动加载 () CONFIG_LDISC_AUTOLOAD

除非请求来自具有CAP_SYS_MODULE权限的进程，否则 AL2 023 内核不会自动加载线路规范TIOCSETDioctl，例如通过使用的软件进行加载。

此选项是内核自我保护项目推荐设置之一。

dmesg非特权用户的访问权限 () CONFIG_SECURITY_DMESG_RESTRICT

默认情况下，AL2023 不允许非特权用户访问。dmesg

此选项是内核自我保护项目推荐设置之一。

SELinux selinuxfs禁用

AL2023 禁用已弃用的CONFIG_SECURITY_SELINUX_DISABLE内核选项，该选项启用了一种在加载策略 SELinux 之前禁用的运行时方法。

此选项是内核自我保护项目推荐设置之一。

其他内核配置变化

CONFIG 选项	AL2/4.14/ aarch64	AL2/4.14/ x86_64	AL2/5.10/ aarch64	AL2/5.10/ x86_64	AL2023/6. 1/ aarch64	AL2023/6 1/ x86_64	AL2023/6. 12/ aarch64	AL2023/6. 12/ x86_64
CONFIG_I	100	250	100	250	100	100	100	100
CONFIG_I _CPUS	4096	8192	4096	8192	4096	8192	4096	8192
CONFIG_I NIC_ON_(PS	y	n	y	n	y	y	y	y
CONFIG_I NIC_ON_(PS_VALUI	1	0	1	0	1	1	1	1
CONFIG_I P	m	m	m	m	n	n	n	n
CONFIG_S IP	m	m	m	m	n	n	n	n

CONFIG 选项	AL2/4.14/ aarch64	AL2/4.14/ x86_64	AL2/5.10/ aarch64	AL2/5.10/ x86_64	AL2023/6. 1/ aarch64	AL2023/6 1/ x86_64	AL2023/6. 12/ aarch64	AL2023/6. 12/ x86_64
CONFIG_ N_PV	不适用	y	不适用	n	不适用	n	不适用	n

CONFIG_HZ

AL2在两个aarch64平台上CONFIG_HZ，023 都设置为 100 x86-64。

CONFIG_NR_CPUS

AL2023 设置CONFIG_NR_CPUS为一个更接近亚马逊 EC2上最大 CPU 内核数的数字。

内核错误处理

AL2023 内核出现故障时会死机。此功能等同于在内核命令行上使用 oops=panic 引导。

内核错误是指内核检测到可能影响系统的进一步可靠性的内部错误。

PPP 和 SLIP 支持

AL2023 不支持 PPP 或 SLIP 协议。

Xen PV 访客支持

AL2023 不支持以 Xen PV 访客身份运行。

内核文件系统支持

内核 AL2 将支持挂载的文件系统发生了几处变化，内核将要解析的分区方案也发生了变化。

CONFIG 选项	AL2/4.14/ aarch64	AL2/4.14/ x86_64	AL2/5.10/ aarch64	AL2/5.10/ x86_64	AL2023/6. 1/ aarch64	AL2023/6 1/ x86_64	AL2023/6. 12/ aarch64	AL2023/6. 12/ x86_64
CONFIG_ S_FS	n	m	n	m	n	n	n	n

CONFIG 选项	AL2/4.14/ aarch64	AL2/4.14/ x86_64	AL2/5.10/ aarch64	AL2/5.10/ x86_64	AL2023/6. 1/ aarch64	AL2023/6. 1/ x86_64	AL2023/6. 12/ aarch64	AL2023/6. 12/ x86_64
CONFIG_ _RXRPC	n	m	n	m	n	n	n	n
CONFIG_ D_DISKL EL	y	y	y	y	n	n	n	n
CONFIG_ AMFS	m	m	m	m	n	n	n	n
CONFIG_ AMFS_BL KDEV	不适用	不适用	y	n	不适用	不适用	不适用	不适用
CONFIG_ _CLONE	不适用	不适用	n	n	n	n	n	n
CONFIG_ _ERA	m	n	m	n	n	n	n	n
CONFIG_ _INTEGR Y	n	m	n	m	m	m	m	m
CONFIG_ _LOG_WR ES	n	n	m	m	m	m	m	m
CONFIG_ _SWITCH	m	n	m	n	n	n	n	n
CONFIG_ _VERITY	m	n	m	n	n	n	n	n

CONFIG 选项	AL2/4.14/ aarch64	AL2/4.14/ x86_64	AL2/5.10/ aarch64	AL2/5.10/ x86_64	AL2023/6. 1/ aarch64	AL2023/6 1/ x86_64	AL2023/6 12/ aarch64	AL2023/6. 12/ x86_64
CONFIG_I RYPT_FS	n	m	n	m	n	n	n	n
CONFIG_I FAT_FS	不适用	不适用	m	m	m	m	m	m
CONFIG_I T2_FS	n	m	n	m	n	n	n	n
CONFIG_I T3_FS	n	m	n	m	n	n	n	n
CONFIG_(S2_FS	m	m	m	m	n	n	n	n
CONFIG_I SPLUS_F	n	m	n	m	n	n	n	n
CONFIG_I S_FS	n	m	n	m	n	n	n	n
CONFIG_ S_FS	n	n	n	n	n	n	n	n
CONFIG_I M_PARTI ON	n	y	n	y	n	n	n	n
CONFIG_I C_PARTI ON	n	y	n	y	n	n	n	n
CONFIG_I S_V2	n	m	n	m	n	n	n	n

CONFIG 选项	AL2/4.14/ aarch64	AL2/4.14/ x86_64	AL2/5.10/ aarch64	AL2/5.10/ x86_64	AL2023/6. 1/ aarch64	AL2023/6 1/ x86_64	AL2023/6. 12/ aarch64	AL2023/6. 12/ x86_64
CONFIG_I FS_FS	n	m	n	n	n	n	n	n
CONFIG_I MFS_FS	n	m	n	m	n	n	n	n
CONFIG_S LARIS_X8 _PARTIT N	n	y	n	y	n	n	n	n
CONFIG_S UASHFS_Z TD	n	y	n	y	y	y	y	y
CONFIG_S N_PARTI ON	n	y	n	y	n	n	n	n

Andrew 文件系统支持 (AFS)

内核在构建时不再支持`afs`文件系统。 AL2 未附带的用户空间支持。`afs`

cramfs 支持

内核不再支持 `cramfs` 文件系统。 AL2023 中的继任者是`squashfs`文件系统。

BSD 磁盘标签支持

内核不再支持 BSD 磁盘标签。如果需要读取带有 BSD 磁盘标签的卷，则 BSDs可以启动各种卷标。

设备映射器变化

在 AL2 023 内核中配置的设备映射器目标进行了几处更改。

eCryptFs 支持

Amazon Linux 中已弃用了 `ecryptfs` 文件系统。的用户空间组件`ecryptfs`已存在于中并在中删除 AL1 AL2， AL2023 不再使用`ecryptfs`支持构建内核。

exFAT

在 5.10 内核中 AL2添加了对`exFAT`文件系统的支持。它在 4.14 内核 AL2 发布时不存在。 AL2023 继续支持`exFAT`文件系统。

ext2、ext3 和 ext4 文件系统

AL2023 附带`CONFIG_EXT4_USE_FOR_EXT2`选项，这意味着`ext4`文件系统代码将用于读取传统`ext2`文件系统。

CONFIG_ _FS GFS2

该内核不再是使用 CONFIG_ GFS2 _FS 构建的。

Apple Extended HFS 文件系统支持 (HFS+)

在中 AL2，只有内`x86-64`核是在支持`hfsplus`文件系统的情况下构建的。 AL2 5.15 内核不`hfsplus`支持任何架构。 AL22023 年，我们完成了对亚马逊 Linux `hfsplus` 支持的弃用。

HFS 文件系统支持

在中 AL2，只有内`x86-64`核是在支持`hfs`文件系统的情况下构建的。 AL2 5.15 内核不`hfs`支持任何架构。 AL22023 年，我们完成了对亚马逊 Linux `hfs` 支持的弃用。

JFS 文件系统支持

较旧的 AL2 `x86-64`内核是在支持`jfs`文件系统的情况下构建的。 AL2 5.15 内核不`jfs`支持任何架构。两者都不是， AL1 AL2 或者与 JFS 用户空间一起提供。 AL22023 年，我们完成了对亚马逊 Linux `jfs` 支持的弃用。

上游 Linux 内核正在考虑删除`JFS`。因此，如果您在`JFS`文件系统上有数据，则应将其迁移到另一个文件系统。2024 `JFS` 年，已从所有当前的亚马逊 Linux 内核中删除。

Windows逻辑磁盘管理器（动态磁盘）支持 (**`CONFIG_LDM_PARTITION`**)

AL2023 不再支持Windows 2000Windows XP、或带有MS-DOS样式分区的Windows Vista动态磁盘。此代码从来不支持中引入的基于 GPT 的较新的动态磁盘。Windows Vista

Macintosh 分区映射支持

AL2023 不再支持经典的 Macintosh 分区图。默认情况下，现代 macOS 版本会创建现代 GPT 分区表，以取代这种较旧的类型。

NFSv2 支持

AL2023 不再支持 NFSv2，但继续支持 NFSv3、 NFSv4、 NFSv4 .1 和 NFSv4 .2。我们建议您迁移到 NFSv3 或更新版本。

NTFS (**`CONFIG_NTFS_FS`**)

从5.10内核`ntfs`开始，在亚马逊 Linux 上访问 NTFS 文件系统的`ntfs3`代码已被替换。 AL2 AL2023 不再包含该`ntfs`代码，而是完全依赖该`ntfs3`代码来访问 NTFS 文件系统。

romfs 文件系统

在 Amazon Linux 中，`squashfs``romfs`文件系统是文件系统的继任者，而且 AL2 023 内核的构建不再支持`romfs`了。

Solaris x86 硬盘分区格式

AL2023 不再支持 Solaris x86 硬盘分区格式。

`squashfs`zstd 压缩

AL2023 在所有支持的架构上增加了对zstd压缩`squashfs`文件系统的支持。

Sun 分区表支持

AL2023 不再支持 Sun 分区表格式 (`CONFIG_SUN_PARTITION`)。

`/tmp` 现在是 **`tmpfs`**

与亚马逊 Linux 2 相比，亚马逊 Linux 2023 的`/tmp`行为方式发生了变化。的默认配置 AL2 是，`/tmp`和`/var/tmp`都在根文件系统上。亚马逊 Linux 2023 默认为使用 `tmpfs` for，内存限制为 50%，最大值为 100 万 `/tmp` inodes。 这些变化使亚马逊 Linux 与其他 Linux 发行版的行为保持一致。

有关 AL2 023 文件系统布局的完整详细信息，请参阅文件系统布局部分`/var/tmp`中的`/tmp`和。

AMI 和容器镜像变更

AMIs 和容器中包含的包裹已发生一些变化。

亚马逊 Linux 2023 推出了the section called “AL2023 最小容器镜像”，并支持构建the section called “构建基本系统 AL2 023 容器镜像”。有关更多信息，请参阅 在容器中使用 AL2 023。

比较亚马逊 Linux 2 和亚马逊 Linux 2023 上安装的软件包 AMIs

对 RPMs 目前亚马逊 Linux 2 和 AL2 023 标准 AMIs的比较。

软件包	AL2 AMI	AL2023 AMI
acl	2.2.51	2.3.1
acpid	2.0.19	2.0.32
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.5.1
amazon-linux-extras	2.0.3	
amazon-linux-extras-yum-plugin	2.0.3	
amazon-linux-repo-s3		2023.6.20241031
amazon-linux-sb-keys		2023.1
amazon-rpm-config		228
amazon-ssm-agent	3.3.987.0	3.3.987.0
amd-ucode-firmware	2020421（没有 arch）	20210208（noarch）
at	3.1.13	3.1.23
attr	2.4.46	2.5.1

软件包	AL2 AMI	AL2023 AMI
audit	2.8.1	3.0.6
audit-libs	2.8.1	3.0.6
authconfig	6.2.8	
aws-cfn-bootstrap	2.0	2.0
awscli	1.18.147	
awscli-2		2.15.30
basesystem	10.0	11
bash	4.2.46	5.2.15
bash-completion	2.1	2.1.1
bc	1.06.95	1.07.1
bind-export-libs	9.11.4	
bind-libs	9.11.4	9.18.28
bind-libs-lite	9.11.4	
bind-license	9.11.4	9.18.28
bind-utils	9.11.4	9.18.28
binutils	2.29.1	2.39
blktrace	1.0.5	
boost-date-time	1.53.0 (x86_64)	
boost-filesystem		1.75.0
boost-system	1.53.0 (x86_64)	1.75.0

软件包	AL2 AMI	AL2023 AMI
boost-thread	1.53.0 (x86_64)	1.75.0
bridge-utils	1.5	
bzip2	1.0.6	1.0.8
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.68	2023.2.68
c-ares		1.19.1
checkpolicy		3.4
chkconfig	1.7.4	1.15
chrony	4.2	4.3
cloud-init	19.3	22.2
cloud-init-cfg-ec2		22.2
cloud-utils-growpart	0.31	0.31
coreutils	8.22	8.32
coreutils-common		8.32
cpio	2.12	2.13
cracklib	2.9.0	2.9.6
cracklib-dicts	2.9.0	2.9.6
cronie	1.4.11	
cronie-anacron	1.4.11	
crontabs	1.11	1.11

软件包	AL2 AMI	AL2023 AMI
crypto-policies		20220428
crypto-policies-scripts		20220428
cryptsetup	1.7.4	2.6.1
cryptsetup-libs	1.7.4	2.6.1
curl	8.3.0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	2.1.27
cyrus-sasl-plain	2.1.26	2.1.27
dbus	1.10.24	1.12.28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.10.24	1.12.28
device-mapper	1.02.170	1.02.185
device-mapper-event	1.02.170	
device-mapper-event-libs	1.02.170	
device-mapper-libs	1.02.170	1.02.185
device-mapper-persistent-data	0.7.3	
dhclient	4.2.5	

软件包	AL2 AMI	AL2023 AMI
dhcp-common	4.2.5	
dhcp-libs	4.2.5	
diffutils	3.3	3.8
dmidecode	3.2	
dmraid	1.0.0.rc16	
dmraid-events	1.0.0.rc16	
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0
dnf-plugin-support-info		1.2
dnf-utils		4.3.0
dosfstools	3.0.20	4.2
dracut	033	055
dracut-config-ec2	2.0	3.0
dracut-config-generic	033	055
dwz		0.14
dyninst	9.3.1 (x86_64)	10.2.1

软件包	AL2 AMI	AL2023 AMI
e2fsprogs	1.42.9	1.46.5
e2fsprogs-libs	1.42.9	1.46.5
ec2-hibinit-agent	1.0.8	1.0.8
ec2-instance-connect	1.1	1.1
ec2-instance-connect-selinux	1.1	1.1
ec2-net-utils	1.7.3	
ec2-utils	1.2	2.2.0
ed	1.9	1.14.2
efibootmgr	15 (aarch64)	
efi-filesystem		5
efi-srpm-macros		5
efivar		38
efivar-libs	31 (aarch64)	38
elfutils-debuginfod-client		0.188
elfutils-default-yama-scope	0.176	0.188
elfutils-libelf	0.176	0.188
elfutils-libs	0.176	0.188
ethtool	4.8	5.15

软件包	AL2 AMI	AL2023 AMI
expat	2.1.0	2.5.0
file	5.11	5.39
file-libs	5.11	5.39
filesystem	3.2	3.14
findutils	4.5.11	4.8.0
fipscheck	1.4.1	
fipscheck-lib	1.4.1	
fonts-srpm-macros		2.0.5
freetype	2.8	
fstrm		0.6.1
fuse-libs	2.9.2	2.9.9
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1.19
gdisk	0.8.10	1.0.8
generic-logos	18.0.0	
GeoIP	1.5.0	
gettext	0.19.8.1	0.21
gettext-libs	0.19.8.1	0.21
ghc-srpm-macros		1.5.0

软件包	AL2 AMI	AL2023 AMI
glib2	2.56.1	2.74.7
glibc	2.26	2.34
glibc-all-langpacks	2.26	2.34
glibc-common	2.26	2.34
glibc-gconv-extra		2.34
glibc-locale-source	2.26	2.34
glibc-minimal-langpack	2.26	
gmp	6.0.0	6.2.1
gnupg2	2.0.22	
gnupg2-minimal		2.3.7
gnutls		3.8.0
go-srpm-macros		3.2.0
gpgme	1.3.2	1.15.1
gpm-libs	1.20.7	1.20.7
grep	2.20	3.8
groff-base	1.22.2	1.22.4
grub2	2.06	
grub2-common	2.06	2.06
grub2-efi-aa64	2.06 (aarch64)	
grub2-efi-aa64-ec2	2.06 (aarch64)	2.06 (aarch64)

软件包	AL2 AMI	AL2023 AMI
grub2-efi-aa64-modules	2.06 (noarch)	
grub2-efi-x64-ec2	2.06 (x86_64)	2.06 (x86_64)
grub2-pc	2.06 (x86_64)	
grub2-pc-modules	2.06 (noarch)	2.06
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.28	8.40
gssproxy	0.7.0	0.8.4
gzip	1.5	1.12
hardlink	1.3	
hibagent	1.1.0	
hostname	3.13	3.23
hunspell	1.3.2	1.7.0
hunspell-en	0.20121024	0.20140811.1
hunspell-en-GB	0.20121024	0.20140811.1
hunspell-en-US	0.20121024	0.20140811.1
hunspell-filesystem		1.7.0
hwdata	0.252	0.384
info	5.1	6.7
inih		49

软件包	AL2 AMI	AL2023 AMI
initscripts	9.49.47	10.09
iproute	5.10.0	6.10.0
iptables	1.8.4	
iptables-libs	1.8.4	
iputils	20180629	20210202
irqbalance	1.7.0	1.9.0
jansson	2.10	2.14
jbigkit-libs	2.0	
jemalloc		5.2.1
jitterentropy		3.4.1
jq		1.7.1
json-c	0.11	0.14
kbd	1.15.5	2.4.0
kbd-legacy	1.15.5	
kbd-misc	1.15.5	2.4.0
kernel	5.10.228	6.1.112
kernel-libbpf		6.1.112
kernel-livepatch-repo-s3		2023.6.20241031
kernel-srpm-macros		1.0
kernel-tools	5.10.228	6.1.112

软件包	AL2 AMI	AL2023 AMI
keyutils	1.5.8	1.6.3
keyutils-libs	1.5.8	1.6.3
kmod	25	29
kmod-libs	25	29
kpartx	0.4.9	
kpatch-runtime	0.9.4	0.9.7
krb5-libs	1.15.1	1.21.3
langtable	0.0.31	
langtable-data	0.0.31	
langtable-python	0.0.31	
less	458	608
libacl	2.2.51	2.3.1
libaio	0.3.109	0.3.111
libarchive		3.7.4
libargon2		20171227
libassuan	2.1.0	2.5.5
libattr	2.4.46	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2.30.2	2.37.4
libcap	2.54	2.48

软件包	AL2 AMI	AL2023 AMI
libcap-ng	0.7.5	0.8.2
libcbor		0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1.42.9	1.46.5
libcomps		0.1.20
libconfig	1.4.9	1.7.2
libcroco	0.6.12	
libcrypt	2.26	
libcurl	8.3.0	
libcurl-minimal		8.5.0
libdaemon	0.14	
libdb	5.3.21	5.3.28
libdb-utils	5.3.21	
libdhash		0.5.0
libdnf		0.69.0
libdrm	2.4.97	
libdwarf	20130207 (x86_64)	
libeconf		0.4.0
libedit	3.0	3.1
libestr	0.1.9	

软件包	AL2 AMI	AL2023 AMI
libev		4.33
libevent	2.0.21	2.1.12
libfastjson	0.99.4	
libfdisk	2.30.2	2.37.4
libffi	3.0.13	3.4.4
libfido2		1.10.0
libgcc	7.3.1	11.4.1
libgcrypt	1.5.3	1.10.2
libgomp	7.3.1	11.4.1
libgpg-error	1.12	1.42
libibverbs		48.0
libicu	50.2	
libidn	1.28	
libidn2	2.3.0	2.3.2
libini_config	1.3.1	1.3.1
libjpeg-turbo	2.0.90	
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libldb		2.6.2
libmaxminddb		1.5.2

软件包	AL2 AMI	AL2023 AMI
libmetalink	0.1.3	0.1.3
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0
libmount	2.30.2	2.37.4
libnetfilter_conntrack	1.0.6	
libnfnetlink	1.0.1	
libnfsidmap	0.25	2.5.4
libnghttp2	1.41.0	1.59.0
libnl3	3.2.28	3.5.0
libnl3-cli	3.2.28	
libpath_utils	0.2.1	0.2.1
libpcap	1.5.3	1.10.1
libpciaccess	0.14 (x86_64)	
libpipeline	1.2.3	1.5.3
libpkgconf		1.8.0
libpng	1.5.13	
libpsl	0.21.5	0.21.1
libpwquality	1.2.3	1.4.4
libref_array	0.1.5	0.1.5
librepo		1.14.5

软件包	AL2 AMI	AL2023 AMI
libreport-filesystem		2.15.2
libseccomp	2.5.2	2.5.3
libselinux	2.5	3.4
libselinux-utils	2.5	3.4
libsemanage	2.5	3.4
libsepol	2.5	3.4
libsigsegv		2.13
libsmartcols	2.30.2	2.37.4
libsolv		0.7.22
libss	1.42.9	1.46.5
libssh2	1.4.3	
libsss_certmap		2.9.4
libsss_idmap	1.16.5	2.9.4
libsss_nss_idmap	1.16.5	2.9.4
libsss_sudo		2.9.4
libstdc++	7.3.1	11.4.1
libstoragemgmt	1.6.1	1.9.4
libstoragemgmt-python	1.6.1	
libstoragemgmt-python-clibs	1.6.1	

软件包	AL2 AMI	AL2023 AMI
libsysfs	2.1.0	
libtalloc		2.3.4
libtasn1	4.10	4.19.0
libtdb		1.4.7
libteam	1.27	
libtevent		0.13.0
libtextstyle		0.21
libtiff	4.0.3	
libtirpc	0.2.4	1.3.3
libunistring	0.9.3	0.9.10
libuser	0.60	0.63
libutempter	1.1.6	1.2.1
libuuid	2.30.2	2.37.4
libuv		1.47.0
libverto	0.2.5	0.3.2
libverto-libev		0.3.2
libverto-libevent	0.2.5	
libwebp	0.3.0	
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4

软件包	AL2 AMI	AL2023 AMI
libxml2-python	2.9.1	
libyaml	0.1.4	0.2.5
libzstd		1.5.5
linux-firmware-whence		20210208（noarch）
lm_sensors-libs	3.4.0	3.6.0
lmdb-libs		0.9.29
logrotate	3.8.6	3.20.1
lsof	4.87	4.94.0
lua	5.1.4	
lua-libs		5.4.4
lua-srpm-macros		1
lvm2	2.02.187	
lvm2-libs	2.02.187	
lz4	1.7.5	
lz4-libs		1.9.4
make	3.82	
man-db	2.6.3	2.9.3
man-pages	3.53	5.10
man-pages-overrides	7.5.2	
mariadb-libs	5.5.68	

软件包	AL2 AMI	AL2023 AMI
mdadm	4.0	
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mlocate	0.26	
mpfr		4.1.0
mtr	0.92	
nano	2.9.8	5.8
ncurses	6.0	6.2
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
nettle	2.7.1	3.8
net-tools	2.0	2.0
newt	0.52.15	0.52.21
newt-python	0.52.15	
nfs-utils	1.3.0	2.5.4
npth		1.6
nspr	4.35.0	4.35.0
nss	3.90.0	3.90.0
nss-pem	1.0.3	
nss-softokn	3.90.0	3.90.0
nss-softokn-freebl	3.90.0	3.90.0

软件包	AL2 AMI	AL2023 AMI
nss-sysinit	3.90.0	3.90.0
nss-tools	3.90.0	
nss-util	3.90.0	3.90.0
ntsysv	1.7.4	1.15
numactl-libs	2.0.9	2.0.14
ocaml-srpm-macros		6
oniguruma		6.9.7.1
openblas-srpm-macros		2
openldap	2.4.44	2.4.57
openssh	7.4p1	8.7p1
openssh-clients	7.4p1	8.7p1
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-libs	1.0.2k	3.0.8
openssl-pkcs11		0.4.12
os-prober	1.58	1.77
p11-kit	0.23.22	0.24.1
p11-kit-trust	0.23.22	0.24.1
package-notes-srpm-macros		0.4
pam	1.1.8	1.5.1

软件包	AL2 AMI	AL2023 AMI
parted	3.1	3.4
passwd	0.79	0.80
pciutils	3.5.1	3.7.0
pciutils-libs	3.5.1	3.7.0
pcre	8.32	
pcre2	10.23	10.40
pcre2-syntax		10.40
perl	5.16.3	
perl-Carp	1.26	1.50
perl-Class-Struct		0.66
perl-constant	1.27	1.33
perl-DynaLoader		1.47
perl-Encode	2.51	3.15
perl-Errno		1.30
perl-Exporter	5.68	5.74
perl-Fcntl		1.13
perl-File-Basename		2.85
perl-File-Path	2.09	2.18
perl-File-stat		1.09
perl-File-Temp	0.23.01	0.231.100

软件包	AL2 AMI	AL2023 AMI
perl-Filter	1.49	
perl-Getopt-Long	2.40	2.52
perl-Getopt-Std		1.12
perl-HTTP-Tiny	0.033	0.078
perl-if		0.60.800
perl-interpreter		5.32.1
perl-IO		1.43
perl-IPC-Open3		1.21
perl-libs	5.16.3	5.32.1
perl-macros	5.16.3	
perl-MIME-Base64		3.16
perl-mro		1.23
perl-overload		1.31
perl-overloading		0.02
perl-parent	0.225	0.238
perl-PathTools	3.40	3.78
perl-Pod-Escapes	1.04	1.07
perl-podlators	2.5.1	4.14
perl-Pod-Perldoc	3.20	3.28.01
perl-Pod-Simple	3.28	3.42

软件包	AL2 AMI	AL2023 AMI
perl-Pod-Usage	1.63	2.01
perl-POSIX		1.94
perl-Scalar-List-Utils	1.27	1.56
perl-SelectSaver		1.02
perl-Socket	2.010	2.032
perl-srpm-macros		1
perl-Storable	2.45	3.21
perl-subs		1.03
perl-Symbol		1.08
perl-Term-ANSIColor		5.01
perl-Term-Cap		1.17
perl-Text-ParseWords	3.29	3.30
perl-Text-Tabs+Wrap		2021.0726
perl-threads	1.87	
perl-threads-shared	1.43	
perl-Time-HiRes	1.9725	
perl-Time-Local	1.2300	1.300
perl-vars		1.05
pinentry	0.8.1	
pkgconf		1.8.0

软件包	AL2 AMI	AL2023 AMI
pkgconfig	0.27.1	
pkgconf-m4		1.8.0
pkgconf-pkg-config		1.8.0
plymouth	0.8.9	
plymouth-core-libs	0.8.9	
plymouth-scripts	0.8.9	
pm-utils	1.4.1	
policycoreutils	2.5	3.4
policycoreutils-python-utils		3.4
popt	1.13	1.18
postfix	2.10.1	
procps-ng	3.3.10	3.3.17
protobuf-c		1.4.1
psacct	6.6.1	6.6.4
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpgme	0.3	
pyliblzma	0.5.3	

软件包	AL2 AMI	AL2023 AMI
pystache	0.5.3	
python	2.7.18	
python2-botocore	1.18.6	
python2-colorama	0.3.9	
python2-cryptography	1.7.2	
python2-dateutil	2.6.1	
python2-futures	3.0.5	
python2-jmespath	0.9.3	
python2-jsonschema	2.5.1	
python2-oauthlib	2.0.1	
python2-pyasn1	0.1.9	
python2-rpm	4.11.3	
python2-rsa	3.4.1	
python2-s3transfer	0.3.3	
python2-setuptools	41.2.0	
python2-six	1.11.0	
python3	3.7.16	3.9.16
python3-attrs		20.3.0
python3-audit		3.0.6
python3-awscrt		0.19.19

软件包	AL2 AMI	AL2023 AMI
python3-babel		2.9.1
python3-cffi		1.14.5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-daemon	2.2.3	2.3.0
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils	0.14	0.16
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-idna		2.10
python3-jinja2		2.11.3
python3-jmespath		0.10.0
python3-jsonpatch		1.21
python3-jsonpointer		2.0

软件包	AL2 AMI	AL2023 AMI
python3-jsonschema		3.2.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs	3.7.16	3.9.16
python3-libselinux		3.4
python3-libsemanage		3.4
python3-libstorage mgmt		1.9.4
python3-lockfile	0.11.0	0.12.2
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip	20.2.2	
python3-pip-wheel		21.3.1
python3-ply		3.11
python3-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0.24
python3-pycparser		2.20

软件包	AL2 AMI	AL2023 AMI
python3-pyrsistent		0.17.3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pystache	0.5.4	
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2.25.1
python3-rpm		4.16.1.3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools	49.1.3	59.6.0
python3-setuptools-wheel		59.6.0
python3-simplejson	3.2.0	
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5
python-babel	0.9.6	

软件包	AL2 AMI	AL2023 AMI
python-backports	1.0	
python-backports-ssl_match_hostname	3.5.0.1	
python-cffi	1.6.0	
python-chardet	2.2.1	
python-chevron		0.13.1
python-configobj	4.7.2	
python-daemon	1.6	
python-devel	2.7.18	
python-docutils	0.12	
python-enum34	1.0.4	
python-idna	2.4	
python-iniparse	0.4	
python-ipaddress	1.0.16	
python-jinja2	2.7.2	
python-jsonpatch	1.2	
python-jsonpointer	1.9	
python-jwcrypto	0.4.2	
python-kitchen	1.1.1	
python-libs	2.7.18	
python-lockfile	0.9.1	

软件包	AL2 AMI	AL2023 AMI
python-markupsafe	0.11	
python-pillow	2.0.0	
python-ply	3.4	
python-pycparser	2.14	
python-pycurl	7.19.0	
python-repoze-lru	0.4	
python-requests	2.6.0	
python-simplejson	3.2.0	
python-srpm-macros		3.9
python-urlgrabber	3.10	
python-urllib3	1.25.9	
pyxattr	0.5.1	
PyYAML	3.10	
qrencode-libs	3.4.1	
quota	4.01	4.06
quota-nls	4.01	4.06
rdate	1.4	
readline	6.2	8.1
rng-tools	6.8	6.14
rootfiles	8.1	8.1

软件包	AL2 AMI	AL2023 AMI
rpcbind	0.2.0	1.2.6
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
rsync	3.1.2	3.2.6
rsyslog	8.24.0	
rust-srpm-macros		21
sbsigntools		0.9.4
scl-utils	20130529	
screen	4.1.0	4.8.0
sed	4.2.2	4.8
selinux-policy	3.13.1	38.1.45
selinux-policy-targeted	3.13.1	38.1.45
setserial	2.17	
setup	2.8.71	2.13.7
setuptool	1.19.11	

软件包	AL2 AMI	AL2023 AMI
sgpio	1.2.0.10	
shadow-utils	4.1.5.1	4.9
shared-mime-info	1.8	
slang	2.2.4	2.3.2
sqlite	3.7.17	
sqlite-libs		3.40.0
sssd-client	1.16.5	2.9.4
sssd-common		2.9.4
sssd-kcm		2.9.4
sssd-nfs-idmap		2.9.4
strace	4.26	6.8
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1.0
sysstat	10.1.5	12.5.6
systemd	219	252.23
systemd-libs	219	252.23
systemd-networkd		252.23
systemd-pam		252.23
systemd-resolved		252.23
systemd-sysv	219	

软件包	AL2 AMI	AL2023 AMI
systemd-udev		252.23
system-release	2	2023.6.20241031
systemtap-runtime	4.5	4.8
sysvinit-tools	2.88	
tar	1.26	1.34
tbb		2020.3
tcp_wrappers	7.6	
tcp_wrappers-libs	7.6	
tcpdump	4.9.2	4.99.1
tcsh	6.18.01	6.24.07
teamd	1.27	
time	1.7	1.9
traceroute	2.0.22	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	1.1.2	2.2
usermode	1.111	
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2.30.2	2.37.4

软件包	AL2 AMI	AL2023 AMI
util-linux-core		2.37.4
vim-common	9.0.2153	9.0.2153
vim-data	9.0.2153	9.0.2153
vim-enhanced	9.0.2153	9.0.2153
vim-filesystem	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153
virt-what	1.18	
wget	1.14	1.21.3
which	2.20	2.21
words	3.0	3.0
xfsdump	3.1.8	3.1.11
xfsprogs	5.0.0	5.18.0
xxd	9.0.2153	9.0.2153
xxhash-libs		0.8.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yajl	2.0.4	
yum	3.4.3	4.14.0
yum-langpacks	0.4.2	
yum-metadata-parser	flink-client	

软件包	AL2 AMI	AL2023 AMI
yum-plugin-priorities	1.1.31	
yum-utils	1.1.31	
zip	3.0	3.0
zlib	1.2.7	1.2.11
zram-generator		1.1.2
zram-generator-defaults		1.1.2
zstd		1.5.5

比较亚马逊 Linux 2 和亚马逊 Linux 2023 上安装的软件包 Minimal AMIs

亚马逊 Linux 2 和 AL2 023 Minimal 上 RPMs 现在的比较。 AMIs

软件包	AL2 最小	AL2023 Minimal
acl	2.2.51	
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.5.1
amazon-linux-extras	2.0.3	
amazon-linux-repo-s3		2023.6.20241031
amazon-linux-sb-keys		2023.1

软件包	AL2 最小	AL2023 Minimal
amd-ucode-firmware	2020421（没有 arch）	20210208（noarch）
audit	2.8.1	3.0.6
audit-libs	2.8.1	3.0.6
authconfig	6.2.8	
awscli-2		2.15.30
basesystem	10.0	11
bash	4.2.46	5.2.15
bind-export-libs	9.11.4	
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.68	2023.2.68
checkpolicy		3.4
chkconfig	1.7.4	
chrony	4.2	4.3
cloud-init	19.3	22.2
cloud-init-cfg-ec2		22.2
cloud-utils-growpart	0.31	0.31
coreutils	8.22	8.32
coreutils-common		8.32
cpio	2.12	2.13
cracklib	2.9.0	2.9.6

软件包	AL2 最小	AL2023 Minimal
cracklib-dicts	2.9.0	2.9.6
cronie	1.4.11	
cronie-anacron	1.4.11	
crontabs	1.11	
crypto-policies		20220428
cryptsetup-libs	1.7.4	2.6.1
curl	8.3.0	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.26	2.1.27
dbus	1.10.24	1.12.28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.10.24	1.12.28
device-mapper	1.02.170	1.02.185
device-mapper-libs	1.02.170	1.02.185
dhclient	4.2.5	
dhcp-common	4.2.5	
dhcp-libs	4.2.5	
diffutils	3.3	3.8
dnf		4.14.0

软件包	AL2 最小	AL2023 Minimal
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0
dnf-plugin-support-info		1.2
dracut	033	055
dracut-config-ec2	2.0	3.0
dracut-config-generic	033	055
e2fsprogs	1.42.9	1.46.5
e2fsprogs-libs	1.42.9	1.46.5
ec2-utils	1.2	2.2.0
efibootmgr	15 (aarch64)	
efi-filesystem		5
efivar		38
efivar-libs	31 (aarch64)	38
elfutils-default-yama-scope	0.176	0.188
elfutils-libelf	0.176	0.188
elfutils-libs	0.176	0.188
expat	2.1.0	2.5.0

软件包	AL2 最小	AL2023 Minimal
file	5.11	5.39
file-libs	5.11	5.39
filesystem	3.2	3.14
findutils	4.5.11	4.8.0
fipscheck	1.4.1	
fipscheck-lib	1.4.1	
freetype	2.8	
fuse-libs	2.9.2	2.9.9
gawk	4.0.2	5.1.0
gdbm	1.13	
gdbm-libs		1.19
gdisk	0.8.10	1.0.8
gettext	0.19.8.1	0.21
gettext-libs	0.19.8.1	0.21
glib2	2.56.1	2.74.7
glibc	2.26	2.34
glibc-all-langpacks	2.26	2.34
glibc-common	2.26	2.34
glibc-locale-source	2.26	2.34
glibc-minimal-langpack	2.26	

软件包	AL2 最小	AL2023 Minimal
gmp	6.0.0	6.2.1
gnupg2	2.0.22	
gnupg2-minimal		2.3.7
gnutls		3.8.0
gpgme	1.3.2	1.15.1
grep	2.20	3.8
groff-base	1.22.2	1.22.4
grub2	2.06	
grub2-common	2.06	2.06
grub2-efi-aa64	2.06 (aarch64)	
grub2-efi-aa64-ec2	2.06 (aarch64)	2.06 (aarch64)
grub2-efi-aa64-modules	2.06 (noarch)	
grub2-efi-x64-ec2	2.06 (x86_64)	2.06 (x86_64)
grub2-pc	2.06 (x86_64)	
grub2-pc-modules	2.06 (noarch)	2.06
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.28	8.40
gzip	1.5	1.12
hardlink	1.3	

软件包	AL2 最小	AL2023 Minimal
hostname	3.13	3.23
hwdata		0.384
info	5.1	
inih		49
initscripts	9.49.47	10.09
iproute	5.10.0	6.10.0
iptables	1.8.4	
iptables-libs	1.8.4	
iputils	20180629	20210202
irqbalance	1.7.0	1.9.0
jansson		2.14
jitterentropy		3.4.1
jq		1.7.1
json-c		0.14
kbd		2.4.0
kbd-misc		2.4.0
kernel	4.14.355	6.1.112
kernel-libbpf		6.1.112
kernel-livepatch-repo-s3		2023.6.20241031
keyutils-libs	1.5.8	1.6.3

软件包	AL2 最小	AL2023 Minimal
kmod	25	29
kmod-libs	25	29
kpartx	0.4.9	
krb5-libs	1.15.1	1.21.3
less	458	608
libacl	2.2.51	2.3.1
libarchive		3.7.4
libargon2		20171227
libassuan	2.1.0	2.5.5
libattr	2.4.46	2.5.1
libblkid	2.30.2	2.37.4
libcap	2.54	2.48
libcap-ng	0.7.5	0.8.2
libcbor		0.7.0
libcom_err	1.42.9	1.46.5
libcomps		0.1.20
libcroco	0.6.12	
libcrypt	2.26	
libcurl	8.3.0	
libcurl-minimal		8.5.0

软件包	AL2 最小	AL2023 Minimal
libdb	5.3.21	5.3.28
libdb-utils	5.3.21	
libdnf		0.69.0
libeconf		0.4.0
libedit	3.0	3.1
libestr	0.1.9	
libfastjson	0.99.4	
libfdisk	2.30.2	2.37.4
libffi	3.0.13	3.4.4
libfido2		1.10.0
libgcc	7.3.1	11.4.1
libgcrypt	1.5.3	1.10.2
libgomp	7.3.1	11.4.1
libgpg-error	1.12	1.42
libicu	50.2	
libidn	1.28	
libidn2	2.3.0	2.3.2
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libmetalink	0.1.3	

软件包	AL2 最小	AL2023 Minimal
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0
libmount	2.30.2	2.37.4
libnetfilter_conntrack	1.0.6	
libnfnetlink	1.0.1	
libnghttp2	1.41.0	1.59.0
libpcap	1.5.3	
libpipeline	1.2.3	1.5.3
libpng	1.5.13	
libpsl	0.21.5	0.21.1
libpwquality	1.2.3	1.4.4
librepo		1.14.5
libreport-filesystem		2.15.2
libseccomp	2.5.2	2.5.3
libselinux	2.5	3.4
libselinux-utils	2.5	3.4
libsemanage	2.5	3.4
libsepol	2.5	3.4
libsigsegv		2.13
libsmartcols	2.30.2	2.37.4

软件包	AL2 最小	AL2023 Minimal
libsolv		0.7.22
libss	1.42.9	1.46.5
libssh2	1.4.3	
libstdc++	7.3.1	11.4.1
libsysfs	2.1.0	
libtasn1	4.10	4.19.0
libtextstyle		0.21
libunistring	0.9.3	0.9.10
libuser	0.60	0.63
libutempter	1.1.6	1.2.1
libuuid	2.30.2	2.37.4
libverto	0.2.5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml	0.1.4	0.2.5
libzstd		1.5.5
linux-firmware-whence		20210208（noarch）
logrotate	3.8.6	3.20.1
lua	5.1.4	
lua-libs		5.4.4

软件包	AL2 最小	AL2023 Minimal
lz4	1.7.5	
lz4-libs		1.9.4
make	3.82	
man-db	2.6.3	2.9.3
mariadb-libs	5.5.68	
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mpfr		4.1.0
ncurses	6.0	6.2
ncurses-base	6.0	6.2
ncurses-libs	6.0	6.2
nettle	2.7.1	3.8
net-tools	2.0	2.0
newt	0.52.15	
newt-python	0.52.15	
npth		1.6
nspr	4.35.0	
nss	3.90.0	
nss-pem	1.0.3	
nss-softokn	3.90.0	
nss-softokn-freebl	3.90.0	

软件包	AL2 最小	AL2023 Minimal
nss-sysinit	3.90.0	
nss-tools	3.90.0	
nss-util	3.90.0	
numactl-libs	2.0.9	2.0.14
oniguruma		6.9.7.1
openldap	2.4.44	2.4.57
openssh	7.4p1	8.7p1
openssh-clients	7.4p1	8.7p1
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-libs	1.0.2k	3.0.8
openssl-pkcs11		0.4.12
os-prober	1.58	1.77
p11-kit	0.23.22	0.24.1
p11-kit-trust	0.23.22	0.24.1
pam	1.1.8	1.5.1
passwd	0.79	0.80
pciutils		3.7.0
pciutils-libs		3.7.0
pcre	8.32	

软件包	AL2 最小	AL2023 Minimal
pcre2	10.23	10.40
pcre2-syntax		10.40
pinentry	0.8.1	
pkgconfig	0.27.1	
policycoreutils	2.5	3.4
popt	1.13	1.18
postfix	2.10.1	
procps-ng	3.3.10	3.3.17
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa	20240208	20240212
pygpgme	0.3	
pyliblzma	0.5.3	
python	2.7.18	
python2-cryptography	1.7.2	
python2-jsonschema	2.5.1	
python2-oauthlib	2.0.1	
python2-pyasn1	0.1.9	
python2-rpm	4.11.3	
python2-setuptools	41.2.0	

软件包	AL2 最小	AL2023 Minimal
python2-six	1.11.0	
python3		3.9.16
python3-attrs		20.3.0
python3-audit		3.0.6
python3-awscrt		0.19.19
python3-babel		2.9.1
python3-cffi		1.14.5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils		0.16
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-idna		2.10

软件包	AL2 最小	AL2023 Minimal
python3-jinja2		2.11.3
python3-jmespath		0.10.0
python3-jsonpatch		1.21
python3-jsonpointer		2.0
python3-jsonschema		3.2.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs		3.9.16
python3-libselinux		3.4
python3-libsemanage		3.4
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip-wheel		21.3.1
python3-ply		3.11
python3-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0.24
python3-pycparser		2.20

软件包	AL2 最小	AL2023 Minimal
python3-pyrsistent		0.17.3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2.25.1
python3-rpm		4.16.1.3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59.6.0
python3-setuptools-wheel		59.6.0
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5
python-babel	0.9.6	
python-backports	1.0	

软件包	AL2 最小	AL2023 Minimal
python-backports-ssl_match_hostname	3.5.0.1	
python-cffi	1.6.0	
python-chardet	2.2.1	
python-configobj	4.7.2	
python-devel	2.7.18	
python-enum34	1.0.4	
python-idna	2.4	
python-iniparse	0.4	
python-ipaddress	1.0.16	
python-jinja2	2.7.2	
python-jsonpatch	1.2	
python-jsonpointer	1.9	
python-jwcrypto	0.4.2	
python-libs	2.7.18	
python-markupsafe	0.11	
python-ply	3.4	
python-pycparser	2.14	
python-pycurl	7.19.0	
python-repoze-lru	0.4	
python-requests	2.6.0	

软件包	AL2 最小	AL2023 Minimal
python-urlgrabber	3.10	
python-urllib3	1.25.9	
pyxattr	0.5.1	
PyYAML	3.10	
qrencode-libs	3.4.1	
readline	6.2	8.1
rng-tools	6.8	6.14
rootfiles	8.1	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
rsyslog	8.24.0	
sbsigntools		0.9.4
sed	4.2.2	4.8
selinux-policy	3.13.1	38.1.45
selinux-policy-targeted	3.13.1	38.1.45

软件包	AL2 最小	AL2023 Minimal
setup	2.8.71	2.13.7
shadow-utils	4.1.5.1	4.9
shared-mime-info	1.8	
slang	2.2.4	
sqlite	3.7.17	
sqlite-libs		3.40.0
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1.0
systemd	219	252.23
systemd-libs	219	252.23
systemd-networkd		252.23
systemd-pam		252.23
systemd-resolved		252.23
systemd-sysv	219	
systemd-udev		252.23
system-release	2	2023.6.20241031
sysvinit-tools	2.88	
tar	1.26	1.34
tcp_wrappers-libs	7.6	
tzdata	2024a	2024a

软件包	AL2 最小	AL2023 Minimal
update-motd	1.1.2	2.2
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2.30.2	2.37.4
util-linux-core		2.37.4
vim-data	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153
which	2.20	2.21
xfsprogs	5.0.0	5.18.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	flink-client	
yum-plugin-priorit ies	1.1.31	
zlib	1.2.7	1.2.11
zram-generator		1.1.2
zram-generator-def aults		1.1.2
zstd		1.5.5

比较 Amazon Linux 2 和 Amazon Linux 2023 基础容器映像上安装的软件包

亚马逊 Linux 2 和 AL2 023 基础容器镜像上的 RPMs 当前镜像的比较。

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

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

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

软件包	AL2 容器	AL2023 集装箱
libdb	5.3.21	
libdb-utils	5.3.21	
libdnf		0.69.0
libffi	3.0.13	3.4.4
libgcc	7.3.1	11.4.1
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.12	1.42
libidn2	2.3.0	2.3.2
libmetalink	0.1.3	
libmodulemd		2.13.0
libmount	2.30.2	2.37.4
libnghttp2	1.41.0	1.59.0
libpsl	0.21.5	0.21.1
librepo		1.14.5
libreport-filesystem		2.15.2
libselinux	2.5	3.4
libsepol	2.5	3.4
libsigsegv		2.13
libsmartcols		2.37.4

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

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

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

软件包	AL2 容器	AL2023 集装箱
rpm-libs	4.11.3	4.16.1.3
rpm-sign-libs		4.16.1.3
sed	4.2.2	4.8
setup	2.8.71	2.13.7
shared-mime-info	1.8	
sqlite	3.7.17	
sqlite-libs		3.40.0
system-release	2	2023.6.20241031
tzdata	2024a	2024a
vim-data	9.0.2153	
vim-minimal	9.0.2153	
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	flink-client	
yum-plugin-ovl	1.1.31	
yum-plugin-priorit ies	1.1.31	
zlib	1.2.7	1.2.11

比较 AL1 和 AL2 023

以下主题描述了与 AL2 023 AL1 之间的主要区别，这些差异尚未包含在与 AL2的比较中。

> **Note**
>
> AL1 已于 2023 年 12 月 31 日达到其 end-of-life (EOL)，自 2024 年 1 月 1 日起将不会收到任何安全更新或错误修复。有关 AL1 EOL 和维护支持的更多信息，请参阅博客文章 Amazon Linux AMI end-of-life 上的更新。我们建议您将应用程序升级到 AL2 023，其中包括直到 2028 年的长期支持。

主题

- 针对每个版本的支持
- systemd 取代 upstart 作为 init 系统
- Python 2.6 和 2.7 已被 Python 3 所取代
- OpenJDK 8 是最旧的 JDK
- AL2亚马逊 Linux 上的 023 个内核更改 1 () AL1
- 比较亚马逊 Linux 1 (AL1) 和亚马逊 Linux 2023 上安装的软件包 AMIs
- 比较亚马逊 Linux 1 (AL1) 和亚马逊 Linux 2023 Minimal 上安装的软件包 AMIs
- 比较亚马逊 Linux 1 (AL1) 和亚马逊 Linux 2023 基础容器镜像上安装的软件包

针对每个版本的支持

对于 AL2 023，我们自发布之日起提供五年的支持。 AL1 自 2020 年 12 月 31 日起终止标准支持，并于 2023 年 12 月 31 日终止维护支持。

有关更多信息，请参阅 发布频率。

systemd 取代 upstart 作为 init 系统

In 已 AL2 `upstart`被 `systemd` as `init` 系统所取代。 AL2023 还使用`systemd`作为其`init`系统，进一步采用了的新特性和功能。`systemd`

Python 2.6 和 2.7 已被 Python 3 所取代

尽管在 2018.03 版本中将 Python 2.6 AL1 标记为 EOL，但这些软件包仍可在存储库中进行安装。 AL2 Python 2.7 是最早支持的 Python 版本， AL2023 完成了向 Python 3 的过渡。 AL2023 存储库中不包含 Python 2.x 版本。

有关 Amazon Linux 上的 Python 的更多信息，请参阅 Python 在 AL2 023。

OpenJDK 8 是最旧的 JDK

AL2023 附带亚马逊 Corretto 作为默认（也是唯一的）Java 开发套件 (JDK)。全部 Java 023 中基于 AL2 023 的软件包是用 Amazon Corretto 17.

在中 AL1，OpenJDK 1.6.0 `java-1.6.0-openjdk` () 在2018.03的第一个版本中停产，OpenJDK 1.7.0 `java-1.7.0-openjdk` () 在2020年中期停产，尽管两个版本都在存储库中可用。 AL1 AL2023 年最早推出的 OpenJDK 版本是 OpenJDK 8，由 Amazon Corretto 8.

AL2亚马逊 Linux 上的 023 个内核更改 1 () AL1

内核实时修补

AL2023 和都 AL2 增加了对内核实时补丁功能的支持。这使您无需重启或停机即可修补 Linux 内核中的关键和重要安全漏洞。有关更多信息，请参阅 023 上的 Kernel Live AL2 补丁。

内核文件系统支持

内核支持挂载的文件系统发生了几处变化，内核 AL1 将要解析的分区方案也发生了变化。

CONFIG 选项	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_AFS_FS	m	n	n	n	n
CONFIG_AF_RXRPC	m	n	n	n	n

CONFIG 选项	AL1/4.14/x86_64	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_BSD_DISKLABEL	y	n	n	n	n
CONFIG_CRAMFS	m	n	n	n	n
CONFIG_CRAMFS_BLOCKDEV	不适用	不适用	不适用	不适用	不适用
CONFIG_DM_CLONE	不适用	n	n	n	n
CONFIG_DM_ERA	n	n	n	n	n
CONFIG_DM_INTEGRITY	m	m	m	m	m
CONFIG_DM_LOG_WRITES	n	m	m	m	m
CONFIG_DM_SWITCH	n	n	n	n	n
CONFIG_DM_VERITY	n	n	n	n	n
CONFIG_ECRYPT_FS	m	n	n	n	n
CONFIG_EXFAT_FS	不适用	m	m	m	m

CONFIG 选项	AL1/4.14/ x86_64	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_EXT2_FS	m	n	n	n	n
CONFIG_EXT3_FS	m	n	n	n	n
CONFIG_GFS2_FS	n	n	n	n	n
CONFIG_HFSPLUS_FS	m	n	n	n	n
CONFIG_HFS_FS	m	n	n	n	n
CONFIG_JFS_FS	n	n	n	n	n
CONFIG_LDM_PARTITION	y	n	n	n	n
CONFIG_MAC_PARTITION	y	n	n	n	n
CONFIG_NFS_V2	m	n	n	n	n
CONFIG_NTFS_FS	m	n	n	n	n
CONFIG_ROMFS_FS	m	n	n	n	n

CONFIG 选项	AL1/4.14/ x86_64	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_SOLARIS_X86_PARTITION	y	n	n	n	n
CONFIG_SQUASHFS_ZSTD	y	y	y	y	y
CONFIG_SUN_PARTITION	y	n	n	n	n

注重安全的内核配置变化

CONFIG 选项	AL1/4.14/ x86_64	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_BUG_ON_DATA_CORRUPTION	y	y	y	y	y
CONFIG_DEFAULT_MMAP_MIN_ADDR	4096	65536	65536	65536	65536
CONFIG_DEVMEM	y	n	n	n	n
CONFIG_DEVPORT	y	n	n	n	n

CONFIG 选项	AL1/4.14/ x86_64	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_FORTIFY_SOURCE	y	y	y	y	y
CONFIG_HARDENED_USERCOPY_FALLBACK	不适用	不适用	不适用	不适用	不适用
CONFIG_INIT_ON_ALLOC_DEFAULT_ON	不适用	n	n	n	n
CONFIG_INIT_ON_FREE_DEFAULT_ON	不适用	n	n	n	n
CONFIG_IOMMU_DEFAULT_DMA_STRICT	不适用	n	n	n	n
CONFIG_LDISC_AUTOLOAD	y	n	n	n	n
CONFIG_SCHED_CORE	不适用	不适用	y	不适用	y
CONFIG_SCHED_STACK_END_CHECK	y	y	y	y	y

CONFIG 选项	AL1/4.14/ x86_64	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_SECURITY_DMESG_RESTRICT	n	y	y	y	y
CONFIG_SECURITY_SELINUX_DISABLE	y	n	n	不适用	不适用
CONFIG_SHUFFLE_PAGE_ALLOCATOR	不适用	y	y	y	y
CONFIG_SLAB_FREELIST_HARDENED	y	y	y	y	y
CONFIG_SLAB_FREELIST_RANDOM	n	y	y	y	y

其他内核配置变化

CONFIG 选项	AL1/4.14/ x86_64	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_HZ	250	100	100	100	100
CONFIG_NR_CPUS	8192	4096	8192	4096	8192

CONFIG 选项	AL1/4.14/ x86_64	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_PANIC_ON_OOPS	n	y	y	y	y
CONFIG_PANIC_ON_OOPS_VALUE	0	1	1	1	1
CONFIG_PPP	m	n	n	n	n
CONFIG_SLIP	m	n	n	n	n
CONFIG_XEN_PV	y	不适用	n	不适用	n

比较亚马逊 Linux 1 (AL1) 和亚马逊 Linux 2023 上安装的软件包 AMIs

与 AL2 023标准 AMIs上的 RPMs 现在 AL1 的比较。

软件包	AL1 AMI	AL2023 AMI
acl	2.2.49	2.3.1
acpid	2.0.19	2.0.32
alsa-lib	1.0.22	
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.5.1

软件包	AL1 AMI	AL2023 AMI
amazon-linux-repo-s3		2023.6.20241031
amazon-linux-sb-keys		2023.1
amazon-rpm-config		228
amazon-ssm-agent	3.2.2222.0	3.3.987.0
amd-ucode-firmware		20210208
at	3.1.10	3.1.23
attr	2.4.46	2.5.1
audit	2.6.5	3.0.6
audit-libs	2.6.5	3.0.6
authconfig	6.2.8	
aws-amitools-ec2	1.5.13	
aws-cfn-bootstrap	1.4	2.0
aws-cli	1.18.107	
awscli-2		2.15.30
basesystem	10.0	11
bash	4.2.46	5.2.15
bash-completion		2.1.1
bc	1.06.95	1.07.1
bind-libs	9.8.2	9.18.28
bind-license		9.18.28

软件包	AL1 AMI	AL2023 AMI
bind-utils	9.8.2	9.18.28
binutils	2.27	2.39
boost-filesystem		1.75.0
boost-system		1.75.0
boost-thread		1.75.0
bzip2	1.0.6	1.0.8
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.62	2023.2.68
c-ares		1.19.1
checkpolicy	2.1.10	3.4
chkconfig	1.3.49.3	1.15
chrony		4.3
cloud-disk-utils	0.27	
cloud-init	0.7.6	22.2
cloud-init-cfg-ec2		22.2
cloud-utils-growpart		0.31
copy-jdk-configs	3.3	
coreutils	8.22	8.32
coreutils-common		8.32
cpio	2.10	2.13

软件包	AL1 AMI	AL2023 AMI
cracklib	2.8.16	2.9.6
cracklib-dicts	2.8.16	2.9.6
cronie	1.4.4	
cronie-anacron	1.4.4	
crontabs	1.10	1.11
crypto-policies		20220428
crypto-policies-scripts		20220428
cryptsetup	1.6.7	2.6.1
cryptsetup-libs	1.6.7	2.6.1
curl	7.61.1	
curl-minimal		8.5.0
cyrus-sasl	2.1.23	
cyrus-sasl-lib	2.1.23	2.1.27
cyrus-sasl-plain	2.1.23	2.1.27
dash	0.5.5.1	
db4	4.7.25	
db4-utils	4.7.25	
dbus	1.6.12	1.12.28
dbus-broker		32
dbus-common		1.12.28

软件包	AL1 AMI	AL2023 AMI
dbus-libs	1.6.12	1.12.28
dejavu-fonts-common	2.33	
dejavu-sans-fonts	2.33	
dejavu-serif-fonts	2.33	
device-mapper	1.02.135	1.02.185
device-mapper-event	1.02.135	
device-mapper-event-libs	1.02.135	
device-mapper-libs	1.02.135	1.02.185
device-mapper-persistent-data	0.6.3	
dhclient	4.1.1	
dhcp-common	4.1.1	
diffutils	3.3	3.8
dmraid	1.0.0.rc16	
dmraid-events	1.0.0.rc16	
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0

软件包	AL1 AMI	AL2023 AMI
dnf-plugin-support-info		1.2
dnf-utils		4.3.0
dosfstools		4.2
dracut	004	055
dracut-config-ec2		3.0
dracut-config-generic		055
dracut-modules-growroot	0.20	
dump	0.4	
dwz		0.14
dyninst		10.2.1
e2fsprogs	1.43.5	1.46.5
e2fsprogs-libs	1.43.5	1.46.5
ec2-hibinit-agent	1.0.0	1.0.8
ec2-instance-connect		1.1
ec2-instance-connect-selinux		1.1
ec2-net-utils	0.7	
ec2-utils	0.7	2.2.0
ed	1.1	1.14.2

软件包	AL1 AMI	AL2023 AMI
efi-filesystem		5
efi-srpm-macros		5
efivar		38
efivar-libs		38
elfutils-debuginfod-client		0.188
elfutils-default-yama-scope		0.188
elfutils-libelf	0.168	0.188
elfutils-libs		0.188
epel-release	6	
ethtool	3.15	5.15
expat	2.1.0	2.5.0
file	5.37	5.39
file-libs	5.37	5.39
filesystem	2.4.30	3.14
findutils	4.4.2	4.8.0
fipscheck	1.3.1	
fipscheck-lib	1.3.1	
fontconfig	2.8.0	

软件包	AL1 AMI	AL2023 AMI
fontpackages-filesystem	1.41	
fonts-srpm-macros		2.0.5
freetype	2.3.11	
fstrm		0.6.1
fuse-libs	2.9.4	2.9.9
gawk	3.1.7	5.1.0
gdbm	1.8.0	
gdbm-libs		1.19
gdisk	0.8.10	1.0.8
generic-logos	17.0.0	
get_reference_source	1.2	
gettext		0.21
gettext-libs		0.21
ghc-srpm-macros		1.5.0
giflib	4.1.6	
glib2	2.36.3	2.74.7
glibc	2.17	2.34
glibc-all-langpacks		2.34
glibc-common	2.17	2.34
glibc-gconv-extra		2.34

软件包	AL1 AMI	AL2023 AMI
glibc-locale-source		2.34
gmp	6.0.0	6.2.1
gnupg2	2.0.28	
gnupg2-minimal		2.3.7
gnutls		3.8.0
go-srpm-macros		3.2.0
gpgme	1.4.3	1.15.1
gpm-libs	1.20.6	1.20.7
grep	2.20	3.8
groff	1.22.2	
groff-base	1.22.2	1.22.4
grub	0.97	
grub2-common		2.06
grub2-efi-x64-ec2		2.06
grub2-pc-modules		2.06
grub2-tools		2.06
grub2-tools-minimal		2.06
grubby	7.0.15	8.40
gssproxy		0.8.4
gzip	1.5	1.12

软件包	AL1 AMI	AL2023 AMI
hesiod	3.1.0	
hibagent	1.0.0	
hmaccalc	0.9.12	
hostname		3.23
hunspell		1.7.0
hunspell-en		0.20140811.1
hunspell-en-GB		0.20140811.1
hunspell-en-US		0.20140811.1
hunspell-filesystem		1.7.0
hwdata	0.233	0.384
info	5.1	6.7
inih		49
initscripts	9.03.58	10.09
iproute	4.4.0	6.10.0
iptables	1.4.21	
iputils	20121221	20210202
irqbalance	1.5.0	1.9.0
jansson		2.14
java-1.7.0-openjdk	1.7.0.321	
javapackages-tools	0.9.1	

软件包	AL1 AMI	AL2023 AMI
jemalloc		5.2.1
jitterentropy		3.4.1
jpackage-utils	1.7.5	
jq		1.7.1
json-c		0.14
kbd	1.15	2.4.0
kbd-misc	1.15	2.4.0
kernel	4.14.336	6.1.112
kernel-libbpf		6.1.112
kernel-livepatch-repo-s3		2023.6.20241031
kernel-srpm-macros		1.0
kernel-tools	4.14.336	6.1.112
keyutils	1.5.8	1.6.3
keyutils-libs	1.5.8	1.6.3
kmod	14	29
kmod-libs	14	29
kpartx	0.4.9	
kpatch-runtime		0.9.7
krb5-libs	1.15.1	1.21.3
lcms2	2.6	

软件包	AL1 AMI	AL2023 AMI
less	436	608
libacl	2.2.49	2.3.1
libaio	0.3.109	0.3.111
libarchive		3.7.4
libargon2		20171227
libassuan	2.0.3	2.5.5
libattr	2.4.46	2.5.1
libbasicobjects		0.1.1
libblkid	2.23.2	2.37.4
libcap	2.16	2.48
libcap54	2.54	
libcap-ng	0.7.5	0.8.2
libcbor		0.7.0
libcgroup	0.40.rc1	
libcollection		0.7.0
libcom_err	1.43.5	1.46.5
libcomps		0.1.20
libconfig		1.7.2
libcurl	7.61.1	
libcurl-minimal		8.5.0

软件包	AL1 AMI	AL2023 AMI
libdb		5.3.28
libdhash		0.5.0
libdnf		0.69.0
libeconf		0.4.0
libedit	2.1.1	3.1
libev		4.33
libevent	2.0.21	2.1.12
libfdisk		2.37.4
libffi	3.0.13	3.4.4
libfido2		1.10.0
libfontenc	1.0.5	
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.11	1.42
libgssglue	0.1	
libibverbs		48.0
libICE	1.0.6	
libicu	50.2	

软件包	AL1 AMI	AL2023 AMI
libidn	1.18	
libidn2	2.3.0	2.3.2
libini_config		1.3.1
libjpeg-turbo	1.2.90	
libkcapi		1.4.0
libkcapi-hmaccalc		1.4.0
libldb		2.6.2
libmaxminddb		1.5.2
libmetalink		0.1.3
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0
libmount	2.23.2	2.37.4
libnetfilter_conntrack	1.0.4	
libnfnetlink	1.0.1	
libnfsidmap	0.25	2.5.4
libnghttp2	1.33.0	1.59.0
libnih	1.0.1	
libnl	flink-client	
libnl3		3.5.0
libpath_utils		0.2.1

软件包	AL1 AMI	AL2023 AMI
libpcap		1.10.1
libpipeline	1.2.3	1.5.3
libpkgconf		1.8.0
libpng	1.2.49	
libpsl	0.6.2	0.21.1
libpwquality	1.2.3	1.4.4
libref_array		0.1.5
librepo		1.14.5
libreport-filesystem		2.15.2
libseccomp		2.5.3
libselinux	2.1.10	3.4
libselinux-utils	2.1.10	3.4
libsemanage	2.1.6	3.4
libsepol	2.1.7	3.4
libsigsegv		2.13
libSM	1.2.1	
libsmartcols	2.23.2	2.37.4
libsolv		0.7.22
libss	1.43.5	1.46.5
libssh2	1.4.2	

软件包	AL1 AMI	AL2023 AMI
libsss_certmap		2.9.4
libsss_idmap		2.9.4
libsss_nss_idmap		2.9.4
libsss_sudo		2.9.4
libstdc++		11.4.1
libstdc++72	7.2.1	
libstoragemgmt		1.9.4
libsysfs	2.1.0	
libtalloc		2.3.4
libtasn1	2.3	4.19.0
libtdb		1.4.7
libtevent		0.13.0
libtextstyle		0.21
libtirpc	0.2.4	1.3.3
libudev	173	
libunistring	0.9.3	0.9.10
libuser	0.60	0.63
libutempter	1.1.5	1.2.1
libuuid	2.23.2	2.37.4
libuv		1.47.0

软件包	AL1 AMI	AL2023 AMI
libverto	0.2.5	0.3.2
libverto-libev		0.3.2
libX11	1.6.0	
libX11-common	1.6.0	
libXau	1.0.6	
libxcb	1.11	
libXcomposite	0.4.3	
libxcrypt		4.4.33
libXext	1.3.2	
libXfont	1.4.5	
libXi	1.7.2	
libxml2	2.9.1	2.10.4
libxml2-python27	2.9.1	
libXrender	0.9.8	
libxslt	1.1.28	
libXtst	1.2.2	
libyaml	0.1.6	0.2.5
libzstd		1.5.5
linux-firmware-whence		20210208
lm_sensors-libs		3.6.0

软件包	AL1 AMI	AL2023 AMI
lmdb-libs		0.9.29
log4j-cve-2021-44228-hotpatch	1.3	
logrotate	3.7.8	3.20.1
lsof	4.82	4.94.0
lua	5.1.4	
lua-libs		5.4.4
lua-srpm-macros		1
lvm2	2.02.166	
lvm2-libs	2.02.166	
lz4-libs		1.9.4
mailcap	2.1.31	
make	3.82	
man-db	2.6.3	2.9.3
man-pages	4.10	5.10
mdadm	3.2.6	
microcode_ctl	2.1	2.1
mingetty	1.08	
mpfr		4.1.0
nano	2.5.3	5.8
nc	1.84	

软件包	AL1 AMI	AL2023 AMI
ncurses	5.7	6.2
ncurses-base	5.7	6.2
ncurses-libs	5.7	6.2
nettle		3.8
net-tools	1.60	2.0
newt	0.52.11	0.52.21
newt-python27	0.52.11	
nfs-utils	1.3.0	2.5.4
npth		1.6
nspr	4.25.0	4.35.0
nss	3.53.1	3.90.0
nss-pem	1.0.3	
nss-softokn	3.53.1	3.90.0
nss-softokn-freebl	3.53.1	3.90.0
nss-sysinit	3.53.1	3.90.0
nss-tools	3.53.1	
nss-util	3.53.1	3.90.0
ntp	4.2.8p15	
ntpdate	4.2.8p15	
ntsysv	1.3.49.3	1.15

软件包	AL1 AMI	AL2023 AMI
numactl	2.0.7	
numactl-libs		2.0.14
ocaml-srpm-macros		6
oniguruma		6.9.7.1
openblas-srpm-macros		2
openldap	2.4.40	2.4.57
openssh	7.4p1	8.7p1
openssh-clients	7.4p1	8.7p1
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-libs		3.0.8
openssl-pkcs11		0.4.12
os-prober		1.77
p11-kit	0.18.5	0.24.1
p11-kit-trust	0.18.5	0.24.1
package-notes-srpm-macros		0.4
pam	1.1.8	1.5.1
pam_ccreds	10	
pam_krb5	2.3.11	
pam_passwdqc	1.0.5	

软件包	AL1 AMI	AL2023 AMI
parted	2.1	3.4
passwd	0.79	0.80
pciutils	3.1.10	3.7.0
pciutils-libs	3.1.10	3.7.0
pcre	8.21	
pcre2		10.40
pcre2-syntax		10.40
perl	5.16.3	
perl-Carp	1.26	1.50
perl-Class-Struct		0.66
perl-constant	1.27	1.33
perl-Digest	1.17	
perl-Digest-HMAC	1.03	
perl-Digest-MD5	2.52	
perl-Digest-SHA	5.85	
perl-DynaLoader		1.47
perl-Encode	2.51	3.15
perl-Errno		1.30
perl-Exporter	5.68	5.74
perl-Fcntl		1.13

软件包	AL1 AMI	AL2023 AMI
perl-File-Basename		2.85
perl-File-Path	2.09	2.18
perl-File-stat		1.09
perl-File-Temp	0.23.01	0.231.100
perl-Filter	1.49	
perl-Getopt-Long	2.40	2.52
perl-Getopt-Std		1.12
perl-HTTP-Tiny	0.033	0.078
perl-if		0.60.800
perl-interpreter		5.32.1
perl-IO		1.43
perl-IPC-Open3		1.21
perl-libs	5.16.3	5.32.1
perl-macros	5.16.3	
perl-MIME-Base64		3.16
perl-mro		1.23
perl-overload		1.31
perl-overloading		0.02
perl-parent	0.225	0.238
perl-PathTools	3.40	3.78

软件包	AL1 AMI	AL2023 AMI
perl-Pod-Escapes	1.04	1.07
perl-podlators	2.5.1	4.14
perl-Pod-Perldoc	3.20	3.28.01
perl-Pod-Simple	3.28	3.42
perl-Pod-Usage	1.63	2.01
perl-POSIX		1.94
perl-Scalar-List-Utils	1.27	1.56
perl-SelectSaver		1.02
perl-Socket	2.010	2.032
perl-srpm-macros		1
perl-Storable	2.45	3.21
perl-subs		1.03
perl-Symbol		1.08
perl-Term-ANSIColor		5.01
perl-Term-Cap		1.17
perl-Text-ParseWords	3.29	3.30
perl-Text-Tabs+Wrap		2021.0726
perl-threads	1.87	
perl-threads-shared	1.43	
perl-Time-HiRes	1.9725	

软件包	AL1 AMI	AL2023 AMI
perl-Time-Local	1.2300	1.300
perl-vars		1.05
pinentry	0.7.6	
pkgconf		1.8.0
pkgconfig	0.27.1	
pkgconf-m4		1.8.0
pkgconf-pkg-config		1.8.0
pm-utils	1.4.1	
policycoreutils	2.1.12	3.4
policycoreutils-python-utils		3.4
popt	1.13	1.18
procmail	3.22	
procps	3.2.8	
procps-ng		3.3.17
protobuf-c		1.4.1
psacct	6.3.2	6.6.4
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa		20240212

软件包	AL1 AMI	AL2023 AMI
python27	2.7.18	
python27-babel	0.9.4	
python27-backports	1.0	
python27-backports-ssl_match_hostname	3.4.0.2	
python27-boto	2.48.0	
python27-botocore	1.17.31	
python27-chardet	2.0.1	
python27-colorama	0.4.1	
python27-configobj	4.7.2	
python27-crypto	2.6.1	
python27-daemon	1.5.2	
python27-dateutil	2.1	
python27-devel	2.7.18	
python27-docutils	0.11	
python27-ecdsa	0.11	
python27-futures	3.0.3	
python27-imaging	1.1.6	
python27-iniparse	0.3.1	
python27-jinja2	2.7.2	
python27-jmespath	0.9.2	

软件包	AL1 AMI	AL2023 AMI
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-PyYAML	3.10	
python27-requests	1.2.3	
python27-rsa	3.4.1	
python27-setuptools	36.2.7	
python27-simplejson	3.6.5	

软件包	AL1 AMI	AL2023 AMI
python27-six	1.8.0	
python27-urlgrabber	3.10	
python27-urllib3	1.24.3	
python27-virtualenv	15.1.0	
python3		3.9.16
python3-attrs		20.3.0
python3-audit		3.0.6
python3-awscrt		0.19.19
python3-babel		2.9.1
python3-cffi		1.14.5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-daemon		2.3.0
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0

软件包	AL1 AMI	AL2023 AMI
python3-docutils		0.16
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-idna		2.10
python3-jinja2		2.11.3
python3-jmespath		0.10.0
python3-jsonpatch		1.21
python3-jsonpointer		2.0
python3-jsonschema		3.2.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs		3.9.16
python3-libselinux		3.4
python3-libsemanage		3.4
python3-libstorage mgmt		1.9.4
python3-lockfile		0.12.2
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip-wheel		21.3.1

软件包	AL1 AMI	AL2023 AMI
python3-ply		3.11
python3-policycoreutils		3.4
python3-prettytable		0.7.2
python3-prompt-toolkit		3.0.24
python3-pycparser		2.20
python3-pyrsistent		0.17.3
python3-pyserial		3.4
python3-pysocks		1.7.1
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2.25.1
python3-rpm		4.16.1.3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59.6.0
python3-setuptools-wheel		59.6.0
python3-six		1.15.0

软件包	AL1 AMI	AL2023 AMI
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5
python-chevron		0.13.1
python-srpm-macros		3.9
quota	4.00	4.06
quota-nls	4.00	4.06
readline	6.2	8.1
rmt	0.4	
rng-tools	5	6.14
rootfiles	8.1	8.1
rpcbind	0.2.0	1.2.6
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit		4.16.1.3
rpm-python27	4.11.3	
rpm-sign-libs		4.16.1.3
rsync	3.0.6	3.2.6

软件包	AL1 AMI	AL2023 AMI
rsyslog	5.8.10	
ruby	2.0	
ruby20	2.0.0.648	
ruby20-irb	2.0.0.648	
ruby20-libs	2.0.0.648	
rubygem20-bigdecimal	1.2.0	
rubygem20-json	1.8.3	
rubygem20-psych	2.0.0	
rubygem20-rdoc	4.2.2	
rubygems20	2.0.14.1	
rust-srpm-macros		21
sbsigntools		0.9.4
screen	4.0.3	4.8.0
sed	4.2.1	4.8
selinux-policy		38.1.45
selinux-policy-targeted		38.1.45
sendmail	8.14.4	
setserial	2.17	
setup	2.8.14	2.13.7
sgpio	1.2.0.10	

软件包	AL1 AMI	AL2023 AMI
shadow-utils	4.1.4.2	4.9
shared-mime-info	1.1	
slang	2.2.1	2.3.2
sqlite	3.7.17	
sqlite-libs		3.40.0
sssd-client		2.9.4
sssd-common		2.9.4
sssd-kcm		2.9.4
sssd-nfs-idmap		2.9.4
strace		6.8
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1.0
sysfsutils	2.1.0	
sysstat		12.5.6
systemd		252.23
systemd-libs		252.23
systemd-networkd		252.23
systemd-pam		252.23
systemd-resolved		252.23
systemd-udev		252.23

软件包	AL1 AMI	AL2023 AMI
system-release	2018.03	2023.6.20241031
systemtap-runtime		4.8
sysvinit	2.87	
tar	1.26	1.34
tbb		2020.3
tcp_wrappers	7.6	
tcp_wrappers-libs	7.6	
tcpdump		4.99.1
tcsh		6.24.07
time	1.7	1.9
tmpwatch	2.9.16	
traceroute	2.0.14	2.1.3
ttmkfdir	3.0.9	
tzdata	2023c	2024a
tzdata-java	2023c	
udev	173	
unzip	6.0	6.0
update-motd	1.0.1	2.2
upstart	0.6.5	
userspace-rcu		0.12.1

软件包	AL1 AMI	AL2023 AMI
ustr	1.0.4	
util-linux	2.23.2	2.37.4
util-linux-core		2.37.4
vim-common	9.0.2120	9.0.2153
vim-data	9.0.2120	9.0.2153
vim-enhanced	9.0.2120	9.0.2153
vim-filesystem	9.0.2120	9.0.2153
vim-minimal	9.0.2120	9.0.2153
wget	1.18	1.21.3
which	2.19	2.21
words	3.0	3.0
xfsdump		3.1.11
xfsprogs		5.18.0
xorg-x11-fonts-Type1	7.2	
xorg-x11-font-utils	7.2	
xxd	9.0.2120	9.0.2153
xxhash-libs		0.8.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0

软件包	AL1 AMI	AL2023 AMI
yum-metadata-parser	flink-client	
yum-plugin-priorit ies	1.1.31	
yum-plugin-upgrade- helper	1.1.31	
yum-utils	1.1.31	
zip	3.0	3.0
zlib	1.2.8	1.2.11
zram-generator		1.1.2
zram-generator-def aults		1.1.2
zstd		1.5.5

比较亚马逊 Linux 1 (AL1) 和亚马逊 Linux 2023 Minimal 上安装的软件包 AMIs

RPMs 现在 AL1 和 AL2 023 Minimal 的比较。 AMIs

软件包	AL1 极简主义	AL2023 Minimal
acpid	2.0.19	
alternatives		1.15
amazon-chrony-config		4.3
amazon-ec2-net-utils		2.5.1
amazon-linux-repo-s3		2023.6.20241031

软件包	AL1 极简主义	AL2023 Minimal
amazon-linux-sb-keys		2023.1
amd-ucode-firmware		20210208
audit	2.6.5	3.0.6
audit-libs	2.6.5	3.0.6
authconfig	6.2.8	
awscli-2		2.15.30
basesystem	10.0	11
bash	4.2.46	5.2.15
binutils	2.27	
bzip2	1.0.6	
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.62	2023.2.68
checkpolicy	2.1.10	3.4
chkconfig	1.3.49.3	
chrony		4.3
cloud-disk-utils	0.27	
cloud-init	0.7.6	22.2
cloud-init-cfg-ec2		22.2
cloud-utils-growpart		0.31
coreutils	8.22	8.32

软件包	AL1 极简主义	AL2023 Minimal
coreutils-common		8.32
cpio	2.10	2.13
cracklib	2.8.16	2.9.6
cracklib-dicts	2.8.16	2.9.6
cronie	1.4.4	
cronie-anacron	1.4.4	
crontabs	1.10	
crypto-policies		20220428
cryptsetup-libs		2.6.1
curl	7.61.1	
curl-minimal		8.5.0
cyrus-sasl	2.1.23	
cyrus-sasl-lib	2.1.23	2.1.27
dash	0.5.5.1	
db4	4.7.25	
db4-utils	4.7.25	
dbus		1.12.28
dbus-broker		32
dbus-common		1.12.28
dbus-libs	1.6.12	1.12.28

软件包	AL1 极简主义	AL2023 Minimal
device-mapper		1.02.185
device-mapper-libs		1.02.185
dhclient	4.1.1	
dhcp-common	4.1.1	
diffutils	3.3	3.8
dnf		4.14.0
dnf-data		4.14.0
dnf-plugin-release-notification		1.2
dnf-plugins-core		4.3.0
dnf-plugin-support-info		1.2
dracut	004	055
dracut-config-ec2		3.0
dracut-config-generic		055
dracut-modules-growroot	0.20	
e2fsprogs	1.43.5	1.46.5
e2fsprogs-libs	1.43.5	1.46.5
ec2-utils	0.7	2.2.0
ed	1.1	

软件包	AL1 极简主义	AL2023 Minimal
efi-filesystem		5
efivar		38
efivar-libs		38
elfutils-default-yama-scope		0.188
elfutils-libelf	0.168	0.188
elfutils-libs		0.188
ethtool	3.15	
expat	2.1.0	2.5.0
file	5.37	5.39
file-libs	5.37	5.39
filesystem	2.4.30	3.14
findutils	4.4.2	4.8.0
fipscheck	1.3.1	
fipscheck-lib	1.3.1	
fuse-libs	2.9.4	2.9.9
gawk	3.1.7	5.1.0
gdbm	1.8.0	
gdbm-libs		1.19
gdisk	0.8.10	1.0.8
generic-logos	17.0.0	

软件包	AL1 极简主义	AL2023 Minimal
get_reference_source	1.2	
gettext		0.21
gettext-libs		0.21
glib2	2.36.3	2.74.7
glibc	2.17	2.34
glibc-all-langpacks		2.34
glibc-common	2.17	2.34
glibc-locale-source		2.34
gmp	6.0.0	6.2.1
gnupg2	2.0.28	
gnupg2-minimal		2.3.7
gnutls		3.8.0
gpgme	1.4.3	1.15.1
grep	2.20	3.8
groff	1.22.2	
groff-base	1.22.2	1.22.4
grub	0.97	
grub2-common		2.06
grub2-efi-x64-ec2		2.06
grub2-pc-modules		2.06

软件包	AL1 极简主义	AL2023 Minimal
grub2-tools		2.06
grub2-tools-minimal		2.06
grubby	7.0.15	8.40
gzip	1.5	1.12
hesiod	3.1.0	
hmaccalc	0.9.12	
hostname		3.23
hwdata	0.233	0.384
info	5.1	
inih		49
initscripts	9.03.58	10.09
iproute	4.4.0	6.10.0
iptables	1.4.21	
iputils	20121221	20210202
irqbalance		1.9.0
jansson		2.14
jitterentropy		3.4.1
jq		1.7.1
json-c		0.14
kbd	1.15	2.4.0

软件包	AL1 极简主义	AL2023 Minimal
kbd-misc	1.15	2.4.0
kernel	4.14.336	6.1.112
kernel-libbpf		6.1.112
kernel-livepatch-repo-s3		2023.6.20241031
keyutils-libs	1.5.8	1.6.3
kmod	14	29
kmod-libs	14	29
krb5-libs	1.15.1	1.21.3
less	436	608
libacl	2.2.49	2.3.1
libarchive		3.7.4
libargon2		20171227
libassuan	2.0.3	2.5.5
libattr	2.4.46	2.5.1
libblkid	2.23.2	2.37.4
libcap	2.16	2.48
libcap54	2.54	
libcap-ng	0.7.5	0.8.2
libcbor		0.7.0
libcgroup	0.40.rc1	

软件包	AL1 极简主义	AL2023 Minimal
libcom_err	1.43.5	1.46.5
libcomps		0.1.20
libcurl	7.61.1	
libcurl-minimal		8.5.0
libdb		5.3.28
libdnf		0.69.0
libeconf		0.4.0
libedit	2.1.1	3.1
libfdisk		2.37.4
libffi	3.0.13	3.4.4
libfido2		1.10.0
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.11	1.42
libicu	50.2	
libidn	1.18	
libidn2	2.3.0	2.3.2
libkcapi		1.4.0

软件包	AL1 极简主义	AL2023 Minimal
libkcapi-hmaccalc		1.4.0
libmnl	1.0.3	1.0.4
libmodulemd		2.13.0
libmount	2.23.2	2.37.4
libnetfilter_conntrack	1.0.4	
libnfnetlink	1.0.1	
libnghttp2	1.33.0	1.59.0
libnih	1.0.1	
libpipeline		1.5.3
libpsl	0.6.2	0.21.1
libpwquality	1.2.3	1.4.4
librepo		1.14.5
libreport-filesystem		2.15.2
libseccomp		2.5.3
libselinux	2.1.10	3.4
libselinux-utils	2.1.10	3.4
libsemanage	2.1.6	3.4
libsepol	2.1.7	3.4
libsigsegv		2.13
libsmartcols	2.23.2	2.37.4

软件包	AL1 极简主义	AL2023 Minimal
libsolv		0.7.22
libss	1.43.5	1.46.5
libssh2	1.4.2	
libstdc++		11.4.1
libstdc++72	7.2.1	
libsysfs	2.1.0	
libtasn1	2.3	4.19.0
libtextstyle		0.21
libudev	173	
libunistring	0.9.3	0.9.10
libuser	0.60	0.63
libutempter	1.1.5	1.2.1
libuuid	2.23.2	2.37.4
libverto	0.2.5	0.3.2
libxcrypt		4.4.33
libxml2	2.9.1	2.10.4
libyaml	0.1.6	0.2.5
libzstd		1.5.5
linux-firmware-whence		20210208
logrotate	3.7.8	3.20.1

软件包	AL1 极简主义	AL2023 Minimal
lua	5.1.4	
lua-libs		5.4.4
lz4-libs		1.9.4
make	3.82	
man-db		2.9.3
microcode_ctl	2.1	2.1
mingetty	1.08	
mpfr		4.1.0
ncurses	5.7	6.2
ncurses-base	5.7	6.2
ncurses-libs	5.7	6.2
nettle		3.8
net-tools	1.60	2.0
newt	0.52.11	
newt-python27	0.52.11	
npth		1.6
nspr	4.25.0	
nss	3.53.1	
nss-pem	1.0.3	
nss-softokn	3.53.1	

软件包	AL1 极简主义	AL2023 Minimal
nss-softokn-freebl	3.53.1	
nss-sysinit	3.53.1	
nss-tools	3.53.1	
nss-util	3.53.1	
ntp	4.2.8p15	
ntpdate	4.2.8p15	
numactl-libs		2.0.14
oniguruma		6.9.7.1
openldap	2.4.40	2.4.57
openssh	7.4p1	8.7p1
openssh-clients		8.7p1
openssh-server	7.4p1	8.7p1
openssl	1.0.2k	3.0.8
openssl-libs		3.0.8
openssl-pkcs11		0.4.12
os-prober		1.77
p11-kit	0.18.5	0.24.1
p11-kit-trust	0.18.5	0.24.1
pam	1.1.8	1.5.1
passwd	0.79	0.80

软件包	AL1 极简主义	AL2023 Minimal
pciutils	3.1.10	3.7.0
pciutils-libs	3.1.10	3.7.0
pcre	8.21	
pcre2		10.40
pcre2-syntax		10.40
pinentry	0.7.6	
pkgconfig	0.27.1	
policycoreutils	2.1.12	3.4
popt	1.13	1.18
procmail	3.22	
procps	3.2.8	
procps-ng		3.3.17
psmisc	22.20	23.4
pth	2.0.7	
publicsuffix-list-dafsa		20240212
python27	2.7.18	
python27-babel	0.9.4	
python27-backports	1.0	
python27-backports-ssl_match_hostname	3.4.0.2	

软件包	AL1 极简主义	AL2023 Minimal
python27-chardet	2.0.1	
python27-configobj	4.7.2	
python27-iniparse	0.3.1	
python27-jinja2	2.7.2	
python27-jsonpatch	1.2	
python27-jsonpointer	1.0	
python27-libs	2.7.18	
python27-markupsafe	0.11	
python27-pycurl	7.19.0	
python27-pygpgme	0.3	
python27-pyliblzma	0.5.3	
python27-pyxattr	0.5.0	
python27-PyYAML	3.10	
python27-requests	1.2.3	
python27-setuptools	36.2.7	
python27-six	1.8.0	
python27-urlgrabber	3.10	
python27-urllib3	1.24.3	
python3		3.9.16
python3-attrs		20.3.0

软件包	AL1 极简主义	AL2023 Minimal
python3-audit		3.0.6
python3-awscrt		0.19.19
python3-babel		2.9.1
python3-cffi		1.14.5
python3-chardet		4.0.0
python3-colorama		0.4.4
python3-configobj		5.0.6
python3-cryptography		36.0.1
python3-dateutil		2.8.1
python3-dbus		1.2.18
python3-distro		1.5.0
python3-dnf		4.14.0
python3-dnf-plugins-core		4.3.0
python3-docutils		0.16
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-idna		2.10
python3-jinja2		2.11.3
python3-jmespath		0.10.0
python3-jsonpatch		1.21

软件包	AL1 极简主义	AL2023 Minimal
python3-jsonpointer		2.0
python3-jsonschema		3.2.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs		3.9.16
python3-libselinux		3.4
python3-libsemanage		3.4
python3-markupsafe		1.1.1
python3-netifaces		0.10.6
python3-oauthlib		3.0.2
python3-pip-wheel		21.3.1
python3-ply		3.11
python3-policycore utils		3.4
python3-prettytable		0.7.2
python3-prompt-too lkit		3.0.24
python3-pycparser		2.20
python3-pyrsistent		0.17.3
python3-pyserial		3.4
python3-pysocks		1.7.1

软件包	AL1 极简主义	AL2023 Minimal
python3-pytz		2022.7.1
python3-pyyaml		5.4.1
python3-requests		2.25.1
python3-rpm		4.16.1.3
python3-ruamel-yaml		0.16.6
python3-ruamel-yaml-clib		0.1.2
python3-setools		4.4.1
python3-setuptools		59.6.0
python3-setuptools-wheel		59.6.0
python3-six		1.15.0
python3-systemd		235
python3-urllib3		1.25.10
python3-wcwidth		0.2.5
readline	6.2	8.1
rng-tools		6.14
rootfiles	8.1	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3

软件包	AL1 极简主义	AL2023 Minimal
rpm-plugin-selinux		4.16.1.3
rpm-plugin-systemd-inhibit		4.16.1.3
rpm-python27	4.11.3	
rpm-sign-libs		4.16.1.3
rsyslog	5.8.10	
sbsigntools		0.9.4
sed	4.2.1	4.8
selinux-policy		38.1.45
selinux-policy-targeted		38.1.45
sendmail	8.14.4	
setserial	2.17	
setup	2.8.14	2.13.7
shadow-utils	4.1.4.2	4.9
shared-mime-info	1.1	
slang	2.2.1	
sqlite	3.7.17	
sqlite-libs		3.40.0
sudo	1.8.23	1.9.15
sysctl-defaults	1.0	1.0

软件包	AL1 极简主义	AL2023 Minimal
sysfsutils	2.1.0	
systemd		252.23
systemd-libs		252.23
systemd-networkd		252.23
systemd-pam		252.23
systemd-resolved		252.23
systemd-udev		252.23
system-release	2018.03	2023.6.20241031
sysvinit	2.87	
tar	1.26	1.34
tcp_wrappers-libs	7.6	
tzdata	2023c	2024a
udev	173	
update-motd	1.0.1	2.2
upstart	0.6.5	
userspace-rcu		0.12.1
ustr	1.0.4	
util-linux	2.23.2	2.37.4
util-linux-core		2.37.4
vim-data	9.0.2120	9.0.2153

软件包	AL1 极简主义	AL2023 Minimal
vim-minimal	9.0.2120	9.0.2153
which	2.19	2.21
xfsprogs		5.18.0
xz	5.2.2	5.2.5
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	flink-client	
yum-plugin-priorit ies	1.1.31	
yum-plugin-upgrade- helper	1.1.31	
zlib	1.2.8	1.2.11
zram-generator		1.1.2
zram-generator-def aults		1.1.2
zstd		1.5.5

比较亚马逊 Linux 1 (AL1) 和亚马逊 Linux 2023 基础容器镜像上安装的软件包

基础容器镜像 AL1 和 AL2 023 基础容器镜像的比较。 RPMs

软件包	AL1 容器	AL2023 集装箱
alternatives		1.15

软件包	AL1 容器	AL2023 集装箱
amazon-linux-repo-cdn		2023.6.20241031
audit-libs		3.0.6
basesystem	10.0	11
bash	4.2.46	5.2.15
bzip2-libs	1.0.6	1.0.8
ca-certificates	2023.2.62	2023.2.68
chkconfig	1.3.49.3	
coreutils	8.22	
coreutils-single		8.32
crypto-policies		20220428
curl	7.61.1	
curl-minimal		8.5.0
cyrus-sasl-lib	2.1.23	
db4	4.7.25	
db4-utils	4.7.25	
dnf		4.14.0
dnf-data		4.14.0
elfutils-default-yama-scope		0.188
elfutils-libelf	0.168	0.188

软件包	AL1 容器	AL2023 集装箱
elfutils-libs		0.188
expat	2.1.0	2.5.0
file-libs	5.37	5.39
filesystem	2.4.30	3.14
gawk	3.1.7	5.1.0
gdbm	1.8.0	
gdbm-libs		1.19
glib2	2.36.3	2.74.7
glibc	2.17	2.34
glibc-common	2.17	2.34
glibc-minimal-langpack		2.34
gmp	6.0.0	6.2.1
gnupg2	2.0.28	
gnupg2-minimal		2.3.7
gpgme	1.4.3	1.15.1
grep	2.20	3.8
gzip	1.5	
info	5.1	
json-c		0.14
keyutils-libs	1.5.8	1.6.3

软件包	AL1 容器	AL2023 集装箱
krb5-libs	1.15.1	1.21.3
libacl	2.2.49	2.3.1
libarchive		3.7.4
libassuan	2.0.3	2.5.5
libattr	2.4.46	2.5.1
libblkid		2.37.4
libcap	2.16	2.48
libcap-ng		0.8.2
libcom_err	1.43.5	1.46.5
libcomps		0.1.20
libcurl	7.61.1	
libcurl-minimal		8.5.0
libdnf		0.69.0
libffi	3.0.13	3.4.4
libgcc		11.4.1
libgcc72	7.2.1	
libgcrypt	1.5.3	1.10.2
libgomp		11.4.1
libgpg-error	1.11	1.42
libicu	50.2	

软件包	AL1 容器	AL2023 集装箱
libidn2	2.3.0	2.3.2
libmodulemd		2.13.0
libmount		2.37.4
libnghttp2	1.33.0	1.59.0
libpsl	0.6.2	0.21.1
librepo		1.14.5
libreport-filesystem		2.15.2
libselinux	2.1.10	3.4
libsepol	2.1.7	3.4
libsigsegv		2.13
libsmartcols		2.37.4
libsolv		0.7.22
libssh2	1.4.2	
libstdc++		11.4.1
libstdc++72	7.2.1	
libtasn1	2.3	4.19.0
libunistring	0.9.3	0.9.10
libuuid		2.37.4
libverto	0.2.5	0.3.2
libxcrypt		4.4.33

软件包	AL1 容器	AL2023 集装箱
libxml2	2.9.1	2.10.4
libxml2-python27	2.9.1	
libyaml		0.2.5
libzstd		1.5.5
lua	5.1.4	
lua-libs		5.4.4
lz4-libs		1.9.4
make	3.82	
mpfr		4.1.0
ncurses	5.7	
ncurses-base	5.7	6.2
ncurses-libs	5.7	6.2
npth		1.6
nspr	4.25.0	
nss	3.53.1	
nss-pem	1.0.3	
nss-softokn	3.53.1	
nss-softokn-freebl	3.53.1	
nss-sysinit	3.53.1	
nss-tools	3.53.1	

软件包	AL1 容器	AL2023 集装箱
nss-util	3.53.1	
openldap	2.4.40	
openssl	1.0.2k	
openssl-libs		3.0.8
p11-kit	0.18.5	0.24.1
p11-kit-trust	0.18.5	0.24.1
pcre	8.21	
pcre2		10.40
pcre2-syntax		10.40
pinentry	0.7.6	
pkgconfig	0.27.1	
popt	1.13	1.18
pth	2.0.7	
publicsuffix-list-dafsa		20240212
python27	2.7.18	
python27-chardet	2.0.1	
python27-iniparse	0.3.1	
python27-kitchen	1.1.1	
python27-libs	2.7.18	
python27-pycurl	7.19.0	

软件包	AL1 容器	AL2023 集装箱
python27-pygpgme	0.3	
python27-pyliblzma	0.5.3	
python27-pyxattr	0.5.0	
python27-urlgrabber	3.10	
python3		3.9.16
python3-dnf		4.14.0
python3-gpg		1.15.1
python3-hawkey		0.69.0
python3-libcomps		0.1.20
python3-libdnf		0.69.0
python3-libs		3.9.16
python3-pip-wheel		21.3.1
python3-rpm		4.16.1.3
python3-setuptools-wheel		59.6.0
readline	6.2	8.1
rpm	4.11.3	4.16.1.3
rpm-build-libs	4.11.3	4.16.1.3
rpm-libs	4.11.3	4.16.1.3
rpm-python27	4.11.3	
rpm-sign-libs		4.16.1.3

软件包	AL1 容器	AL2023 集装箱
sed	4.2.1	4.8
setup	2.8.14	2.13.7
shared-mime-info	1.1	
sqlite	3.7.17	
sqlite-libs		3.40.0
sysctl-defaults	1.0	
system-release	2018.03	2023.6.20241031
tar	1.26	
tzdata	2023c	2024a
xz-libs	5.2.2	5.2.5
yum	3.4.3	4.14.0
yum-metadata-parser	flink-client	
yum-plugin-ovl	1.1.31	
yum-plugin-priorit ies	1.1.31	
yum-utils	1.1.31	
zlib	1.2.8	1.2.11

AL2023 系统要求

本节介绍使用 AL2 023 的系统要求。

主题

- 运行 AL2 023 时的 CPU 要求
- 运行 AL2 023 所需的内存 (RAM)

运行 AL2 023 时的 CPU 要求

要运行任何 AL2 023 代码，使用的处理器需要满足某些最低要求。尝试在不符合这些要求 CPUs 的情况下运行 AL2 023 可能会导致代码执行初期出现非法指令错误。

最低要求适用于AL2亚马逊上的 023 EC2AL2023 装在容器里、和AL2亚马逊以外的 023 EC2。

AL22023 年的 ARM CPU 要求

全部 AL2 023 (`aarch64`ARM) 二进制文件是为 64 位构建的。没有 32 位 ARM 二进制文件可用，因此 64 位 ARM 需要使用中央处理器。

> **Note**
>
> 对于基于 ARM 的实例， AL2023 仅支持使用 Graviton2 或更高版本处理器的实例类型。AL2023 不支持 A1 实例。

AL2023 需要兼容 ARMv8 .2 的处理器以及加密扩展 () `ARMv8.2+crypto`。所有 AL2 023 软件包`aarch64`均使用`-march=armv8.2-a+crypto`编译器标志构建。尽管当尝试在较旧版本上运行 AL2 023 代码时，我们会尝试打印优雅的错误消息 ARM 处理器，第一条错误消息可能是非法指令错误。

> **Note**
>
> 由于 AL2 023 的`aarch64`基本 CPU 要求，所有 Raspberry Pi 之前的系统 Raspberry Pi 5 不符合最低的 CPU 要求。

x86-64 023 的 CPU 要求 AL2

所有 AL2 023 x86-64 二进制文件都是通过传递给编译器-march=x86-64-v2来为x86-64架构的x86-64v2修订而构建的。

该架构的x86-64v2修订版在基准x86-64架构的基础上增加了以下 CPU 功能：

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

这大致对应于 2009 年或之后发布的x86-64处理器。示例包括 Intel Nehalem, AMD Jaguar, Atom Silvermont，以及 VIA Nano 以及 Eden C 微架构。

在 Amazon 中 EC2，所有x86-64实例类型都支持 x86-64v2M1，包括C1、和M2实例系列。

没有 32 位 x86 (i686) AL2 023 二进制文件已构建。尽管 AL2 023 保留了对运行 32 位用户空间二进制文件的支持，但此功能已被弃用，并且可能会在未来的主版本的 Amazon Linux 中删除。有关更多信息，请参阅 32 位 x86 (i686) 软件包。

运行 AL2 023 所需的内存 (RAM)

Amazon EC2 .nano 系列的实例类型（t2.nano、t3.nanot3a.nano、和t4g.nano）具有 512 MB 的内存，这是 AL2 023 的最低要求。

> **Note**
>
> 尽管 512 MB 是最低要求，但这些实例类型受内存限制，功能和性能可能会受到限制。

AL2023 张映像尚未在内存小于 512 MB 的系统上进行测试。在小于 512 MB 的 RAM 中运行基于 AL2 023 的容器映像将取决于容器化工作负载。

某些工作负载（例如某些 AL2 023 版本`dnf upgrade`之间）可能需要超过 512 MB 的 RAM。出于这个原因，AL2023.3 版本引入了对内存小于 800 MB 的实例的`zram`默认启用功能。对于容器化工作负载，这意味着某些工作负载可能在具有此内存量的 AL2 023 实例上运行良好，但在内存使用量限制的容器中运行时会失败。

对于内存小于 800 MB 的实例类型，默认情况下， AL2023（自 AL2023.3 或更高版本起）将启用`zram`基于交换的功能。内存小于 800 MB 的 Amazon EC2 实例类型的示例包括`t4g.nano``t3a.nano`、`t3.nano`、`t2.nano`、和`t1.micro`。这意味着这些实例类型的内存不足情况更少，因为 AL2 023 将按需压缩和解压缩内存页面。这可以支持原本需要具有更多内存的实例类型的工作负载，但会以增加压缩所需的 CPU 使用量为代价。

AL2023 图形化桌面

截至 2023.7 版，亚马逊 Linux 2023 提供了基于 GNOME 的可选、轻量级、云端优化的图形界面。这种现代桌面环境通过 Firefox 等内置工具提供更高的生产力功能，用于安全浏览，同时保持 Amazon DCV 和 VNC 对远程访问的支持。

相关 主题

有关安装图形桌面环境的更多信息，请参阅以下文档：

- 教程：在 023 上 AL2安装 GNOME 桌面环境

在 AL2 023 上运行应用程序

本节介绍在 Amazon Linux 2023 (AL2023) 上运行应用程序的方法，包括管理应用程序何时启动（和重启）以及控制资源使用情况。

主题

- 使用限制 AL2 023 中的进程资源使用量 systemd
- 使用限制 AL2 023 中的进程资源使用量 cgroups

使用限制 AL2 023 中的进程资源使用量 systemd

在 Amazon Linux 2023 (AL2023) 上，我们建议使用systemd来控制进程或进程组可以使用的资源。使用systemd是一种功能强大且易于使用的替代品，既可以cgroups手动操作，也可以使用诸如cpulimit以前在第三方EPEL存储库中仅适用于Amazon Linux的实用程序。

如需全面信息，请参阅 systemd.resource- control 的上游systemd文档，或者 man 在 AL2 023 实例systemd.resource-control上显示的页面。

以下示例将使用 stress-ng CPU 压力测试（来自stress-ng软件包）来模拟 CPU 密集型应用程序，memcached并模拟内存密集型应用程序。

以下示例介绍了对一次性命令设置 CPU 限制和对服务设置内存限制。systemd提供的大多数资源限制都可以在任何运行流程的地方使用，并且可以同时使用多个资源限制。systemd出于说明的目的，以下示例仅限于单个约束。

systemd-run用于运行一次性命令的资源控制

虽然通常与系统服务相关联，但非 root 用户systemd也可以使用它来运行服务、安排计时器或运行一次性进程。在以下示例中，我们将使用stress-ng作为示例应用程序。在第一个示例中，我们将使用systemd-runec2-user默认帐户运行它，在第二个示例中，我们将限制其 CPU 使用率。

Example 在命令行**systemd-run**上使用来运行进程，但不限制资源使用

1. 确保stress-ng软件包已安装，因为我们将在示例中使用它。

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

2. systemd-run用于执行 10 秒的 CPU 压力测试，而不限制其可使用的 CPU 量。

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

   该--user选项告诉systemd-run要以我们登录的用户身份执行命令，该--tty选项的意思是 TTY 已连接，--wait表示等到服务完成，该--property=CPUAccounting=1选项指示systemd-run记录运行该进程使用了多少 CPU 时间。--property命令行选项可用于传递可在配置文件中systemd.unit配置的systemd-run设置。

当被指示对 CPU 施加负载时，stress-ng程序将使用所有可用的 CPU 时间在你要求运行的持续时间内执行测试。对于现实世界的应用程序，可能需要限制进程的总运行时间。在下面的示例中，我们将stress-ng要求运行的时间长于我们使用的最大持续时间限制systemd-run。

Example 在命令行**systemd-run**上使用来运行进程，将 CPU 使用率限制在 1 秒以内

1. 确保stress-ng已安装以运行此示例。
2. 该LimitCPU属性等效于ulimit -t该属性，将限制允许此进程在 CPU 上使用的最大时间。在这种情况下，由于我们要求进行 10 秒的 stress 运行，并且我们将 CPU 使用率限制为 1 秒，因此该命令将收到SIGXCPU信号并失败。

```
[ec2-user ~]$ systemd-run --user --tty --wait --property=CPUAccounting=1 --
property=LimitCPU=1 stress-ng --cpu 1 --timeout 10
Running as unit: run-u12.service
Press ^] three times within 1s to disconnect TTY.
stress-ng: info:  [340349] setting to a 10 second run per stressor
stress-ng: info:  [340349] dispatching hogs: 1 cpu
stress-ng: fail:  [340349] cpu instance 0 corrupted bogo-ops counter, 1370 vs 0
stress-ng: fail:  [340349] cpu instance 0 hash error in bogo-ops counter and run
 flag, 3250129726 vs 0
```

```
stress-ng: fail:  [340349] metrics-check: stressor metrics corrupted, data is
 compromised
stress-ng: info:  [340349] unsuccessful run completed in 1.14s
Finished with result: exit-code
Main processes terminated with: code=exited/status=2
Service runtime: 1.201s
CPU time consumed: 1.008s
```

更常见的是，您可能需要限制特定进程可以消耗的 CPU 时间百分比。在下面的示例中，我们将限制可以消耗的 CPU 时间百分比`stress-ng`。对于现实世界的服务，可能需要限制后台进程可以消耗的最大 CPU 时间百分比，以便为处理用户请求的进程腾出资源。

Example 用于**`systemd-run`**将一个进程限制在一个 CPU 上占用 CPU 时间的 10%

1. 确保`stress-ng`已安装以运行此示例。
2. 我们将使用该`CPUQuota`属性来告诉我们`systemd-run`要运行的命令限制CPU使用率。我们没有限制该进程可以运行的时间，只是在限制它可以使用多少CPU。

```
[ec2-user ~]$ systemd-run --user --tty --wait --property=CPUAccounting=1 --
property=CPUQuota=10% stress-ng --cpu 1 --timeout 10
Running as unit: run-u13.service
Press ^] three times within 1s to disconnect TTY.
stress-ng: info:  [340664] setting to a 10 second run per stressor
stress-ng: info:  [340664] dispatching hogs: 1 cpu
stress-ng: info:  [340664] successful run completed in 10.08s
Finished with result: success
Main processes terminated with: code=exited/status=0
Service runtime: 10.140s
CPU time consumed: 1.014s
```

注意是怎么回事 CPU 记账告诉我们，虽然该服务运行了 10 秒，但它只消耗了 1 秒的实际 CPU 时间。

可以通过多种方式进行配置，`systemd`以限制 CPU、内存、网络和 IO 的资源使用量。请参阅 systemd.resource- control 的上游`systemd`文档，或者 man 在 AL2 023 实例`systemd.resource-control`上的页面以获取全面的文档。

幕后，systemd是使用 Linux 内核的功能（例如cgroups实现这些限制），同时无需手动配置它们。的 Linux 内核文档cgroup-v2包含有关cgroups工作的大量细节。

systemd服务中的资源控制

可以将多个参数添加到systemd服务[Service]部分以控制系统资源使用情况。其中包括硬限制和软限制。有关每个选项的确切行为，请参阅 systemd.resource- control 的上游systemd文档，或者 man 在 AL2 023 实例systemd.resource-control上显示的页面。

常用的限制是MemoryHigh指定内存使用量的限制限制、MemoryMax设置硬性上限（达到该上限后，就会调用 OOM Killer）和CPUQuota（如上一节所示）。也可以配置权重和优先级，而不是固定数字。

Example **systemd**用于设置服务的内存使用限制

在此示例中，我们将为其设置硬内存使用限制memcached，即简单的键值缓存，并显示如何为该服务而不是整个系统调用 OOM Killer。

1. 首先，我们需要安装此示例所需的软件包。

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

2. 启用，memcached.service然后启动该服务，使其memcached正常运行。

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

3. 检查是否memcached.service正在运行。

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

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

4. 现在它已安装并正在运行，我们可以通过在缓存中插入一些随机数据来观察它是否起作用memcached

   默认情况下/etc/sysconfig/memcached，该CACHESIZE变量设置为 64，即 64 兆字节。通过向缓存中插入的数据超过最大缓存大小，我们可以看到我们填满了缓存，有些项目被使用逐出memcached-tool，并且使用了大约 64MB 的内存。memcached.service

```
[ec2-user ~]$ for i in $(seq 1 150); do dd if=/dev/random of=$i bs=512k count=1;
 memcp -s localhost $i; done
[ec2-user ~]$ memcached-tool localhost display
  #  Item_Size  Max_age   Pages   Count   Full?  Evicted Evict_Time OOM
  2     120B         0s       1       0      no        0        0    0
 39   512.0K         4s      63     126     yes       24        2    0
[ec2-user ~]$ sudo systemctl status memcached.service
# memcached.service - memcached daemon
     Loaded: loaded (/usr/lib/systemd/system/memcached.service; enabled; preset:
 disabled)
     Active: active (running) since Fri 2025-01-31 22:36:42 UTC; 7min ago
   Main PID: 356294 (memcached)
      Tasks: 10 (limit: 18907)
     Memory: 66.7M
        CPU: 203ms
     CGroup: /system.slice/memcached.service
             ##356294 /usr/bin/memcached -p 11211 -u memcached -m 64 -c 1024 -l
 127.0.0.1,::1

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

5. 使用该MemoryMax属性为如果被击中，OOM Killer 将在memcached.service何处被调用，设置硬性限制。通过将其他选项添加到覆盖文件中，可以为服务设置其他选项。这可以通过直接编辑/etc/systemd/system/memcached.service.d/override.conf文件来完成，也可以使用edit命令以交互方式完成。systemctl

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

将以下内容添加到替代项中，将该服务的内存硬限制设置为 32MB。

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

6. 告知systemd要重新加载其配置

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

7. 请注意，现在memcached.service正在运行，内存限制为 32MB。

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

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

8. 该服务将在占用少于32MB的内存时正常运行，我们可以通过将少于32MB的随机数据加载到缓存中，然后检查服务的状态来检查这一点。

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

```
[ec2-user ~]$ sudo systemctl status memcached.service
# memcached.service - memcached daemon
     Loaded: loaded (/usr/lib/systemd/system/memcached.service; enabled; preset:
 disabled)
    Drop-In: /etc/systemd/system/memcached.service.d
             ##override.conf
```

```
     Active: active (running) since Fri 2025-01-31 23:14:48 UTC; 3s ago
   Main PID: 359492 (memcached)
      Tasks: 10 (limit: 18907)
     Memory: 18.2M (max: 32.0M available: 13.7M)
        CPU: 42ms
     CGroup: /system.slice/memcached.service
             ##359492 /usr/bin/memcached -p 11211 -u memcached -m 64 -c 1024 -l
 127.0.0.1,::1

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

9. 现在，通过memcached尝试使用默认memcached配置的完整 64MB 缓存，我们可以使用超过 32MB 的内存。

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

你会发现，在执行上述命令的某个时候，memcached服务器会出现连接错误。这是因为 OOM Killer 由于我们对其施加的限制而终止了该进程。系统的其余部分将正常运行，OOM Killer 不会考虑其他进程，因为只有我们限制了memcached.service这些进程。

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

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

```
Jan 31 23:20:28 ip-1-2-3-4.us-west-2.compute.internal systemd[1]:
 memcached.service: Failed with result 'oom-kill'.
```

使用限制 AL2 023 中的进程资源使用量 cgroups

虽然建议使用使用资源控制 systemd，但本节介绍基本libcgroup-tools实用程序的基本用法，以限制进程的 CPU 和内存使用量。这两种方法都是使用该cpulimit实用程序的替代方法，之前可在中找到EPEL。

以下示例介绍运行stress-ng压力测试（来自stress-ng软件包），同时使用软件libcgroup-tools包中的实用程序和中的sysfs可调参数限制其 CPU 和内存使用量。

在命令行**libcgroup-tools**上使用以限制资源使用

1. 安装 libcgroup-tools 软件包。

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

2. cgroup使用memory和cpu控制器创建 a，并为其命名 (our-example-limits)。使用-a和-t选项允许ec2-user用户控制的可调参数 cgroup

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

现在有一个/sys/fs/cgroup/our-example-limits/目录，其中包含可用于控制每个可调对象的文件。

> **Note**
>
> 亚马逊 Linux 2 使用cgroup-v1cgroup-v2的是在 AL2 023 上使用的。开 AL2启后，sysfs路径不同，会有/sys/fs/cgroup/memory/our-example-limits和/sys/fs/cgroup/cpu/our-example-limits目录ec2-user所拥有的目录包含可用于控制限制的文件cgroup。

3. 将所有进程的内存使用量限制在 1 亿字节cgroup以内。

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

> **Note**
>
> 亚马逊 Linux 2 使用的`cgroup-v1`不是亚马逊 Linux 2023 所`cgroup-v2`使用的。这意味着有些可调参数是不同的。要限制内存使用量 AL2，请改用以下可调参数。
>
> ```
> [ec2-user ~]$ echo 10000000 > /sys/fs/cgroup/memory/our-example-limits/memory.limit_in_bytes
> ```

4. 将我们所有进程的 CPU 使用率限制在 10% `cgroup` 以内。该`cpu.max`文件的格式是`$MAX $PERIOD`，将该组限制`$MAX`为每人消费`$PERIOD`。

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

   亚马逊 Linux 2 使用的`cgroup-v1`不是亚马逊 Linux 2023 所`cgroup-v2`使用的。这意味着有些可调参数是不同的，包括如何限制 CPU 使用率。

5. 下面的示例在中运行`stress-ng`（可以通过运行来安装`dnf install -y stress-ng`）`our-example-limits` cgroup。 在`stress-ng`命令运行时，你可以观察到`top`，它被限制在 10% 以内 CPU 时间。

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

6. 通过删除 cgroup 进行清理

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

的 Linux 内核文档`cgroup-v2`包含有关其工作原理的大量详细信息。CPU 和内存控制器的文档详细介绍了如何使用每个可调选项。

在 on 上使用 AL2 023 AWS

您可以将 AL2 023 设置为与其他 AWS 服务人一起使用。例如，您可以在启动亚马逊弹性计算云 (Amazon EC2) 实例时选择 AL2 023 AMI。

对于这些设置过程，您可以使用 AWS Identity and Access Management (IAM) 服务。有关 IAM 的完整信息，请参阅以下参考材料：

- AWS Identity and Access Management (IAM)
- IAM 用户指南

主题

- 入门 AWS
- AL2亚马逊上的 023 EC2
- 在容器中使用 AL2 023
- AL2023 开启 AWS Elastic Beanstalk
- 在中使用 AL2 023 AWS CloudShell
- 使用基于 AL2 023 的 Amazon ECS AMIs 托管容器化工作负载
- 在 AL2 023 上使用亚马逊 Elastic File System
- 使用基于 023 的亚马逊 EMR AL2
- 在中使用 AL2 023 AWS Lambda

入门 AWS

注册获取 AWS 账户

如果您没有 AWS 账户，请完成以下步骤来创建一个。

要注册 AWS 账户

1. 打开https://portal.aws.amazon.com/billing/注册。
2. 按照屏幕上的说明操作。

   在注册时，将接到电话或收到短信，要求使用电话键盘输入一个验证码。

当您注册时 AWS 账户，就会创建AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务 和资源。作为最佳安全实践，请为用户分配管理访问权限，并且只使用根用户来执行需要根用户访问权限的任务。

AWS 注册过程完成后会向您发送一封确认电子邮件。您可以随时前往 https://aws.amazon.com/并选择“我的账户”，查看您当前的账户活动并管理您的账户。

创建具有管理访问权限的用户

注册后，请保护您的安全 AWS 账户 AWS 账户根用户 AWS IAM Identity Center，启用并创建管理用户，这样您就不会使用 root 用户执行日常任务。

保护你的 AWS 账户根用户

1. 选择 Root 用户并输入您的 AWS 账户 电子邮件地址，以账户所有者的身份登录。AWS Management Console在下一页上，输入您的密码。

   要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的 Signing in as the root user。
2. 为您的根用户启用多重身份验证（MFA）。

   有关说明，请参阅 I A M 用户指南中的为 AWS 账户 根用户启用虚拟 MFA 设备（控制台）。

创建具有管理访问权限的用户

1. 启用 IAM Identity Center。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 Enabling AWS IAM Identity Center。
2. 在 IAM Identity Center 中，为用户授予管理访问权限。

   有关使用 IAM Identity Center 目录 作为身份源的教程，请参阅《用户指南》 IAM Identity Center 目录中的使用默认设置配置AWS IAM Identity Center 用户访问权限。

以具有管理访问权限的用户身份登录

- 要使用您的 IAM Identity Center 用户身份登录，请使用您在创建 IAM Identity Center 用户时发送到您的电子邮件地址的登录网址。

有关使用 IAM Identity Center 用户登录的帮助，请参阅AWS 登录 用户指南中的登录 AWS 访问门户。

将访问权限分配给其他用户

1. 在 IAM Identity Center 中，创建一个权限集，该权限集遵循应用最低权限的最佳做法。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 Create a permission set。

2. 将用户分配到一个组，然后为该组分配单点登录访问权限。

   有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 Add groups。

授权以编程方式访问

如果用户想在 AWS 外部进行交互，则需要编程访问权限 AWS Management Console。授予编程访问权限的方式取决于正在访问的用户类型 AWS。

要向用户授予编程式访问权限，请选择以下选项之一。

哪个用户需要编程式访问权限？	目的	方式
人力身份 （在 IAM Identity Center 中管理的用户）	使用临时证书签署向 AWS CLI AWS SDKs、或发出的编程请求 AWS APIs。	按照您希望使用的界面的说明进行操作。 • 有关的 AWS CLI，请参阅《AWS Command Line Interface 用户指南》 AWS IAM Identity Center中的 “配置 AWS CLI 要使用”。 • 有关工具和 AWS SDKs AWS APIs，请参阅《工具参考指南》中的 IAM 身份中心身份验证AWSSDKs 和工具参考指南。

哪个用户需要编程式访问权限？	目的	方式
IAM	使用临时证书签署向 AWS CLI AWS SDKs、或发出的编程请求 AWS APIs。	按照 IAM 用户指南中的将临时证书与 AWS 资源配合使用中的说明进行操作。
IAM	（不推荐使用） 使用长期凭证签署向 AWS CLI AWS SDKs、或发出的编程请求 AWS APIs。	按照您希望使用的界面的说明进行操作。 • 有关信息 AWS CLI，请参阅用户指南中的使用 IAM 用户证书进行身份验证。AWS Command Line Interface • 有关 AWS SDKs 和工具，请参阅《工具参考指南》AWS SDKs 和《工具参考指南》中的使用长期凭证进行身份验证。 • 有关信息 AWS APIs，请参阅 IAM 用户指南中的管理 IAM 用户的访问密钥。

AL2亚马逊上的 023 EC2

使用以下过程之一启动带有 AL2 023 AMI 的 Amazon EC2 实例。您可以选择标准 AMI，也可以选择 AMI 最低版本。有关标准 AMI 与AMI 最低版本 之间区别的更多信息，请参阅比较 AL2 023 标准（默认）和最小值 AMIs。

主题

- 使用亚马逊 EC2 控制台启动 AL2 023
- 使用 SSM 参数启动 AL2 023 和 AWS CLI
- 使用启动最新的 AL2 023 AMI AWS CloudFormation
- 使用特定的 AMI ID 启动 AL2 023
- AL2023 AMI 的弃用和生命周期

- 正在连接到 AL2 023 个实例
- 比较 AL2 023 标准版和最小值 AMIs

使用亚马逊 EC2 控制台启动 AL2 023

使用亚马逊 EC2 控制台启动 AL2 023 AMI。

> **Note**
>
> 对于基于 ARM 的实例，AL2023 仅支持使用 Graviton2 或更高版本处理器的实例类型。AL2023 不支持 A1 实例。

使用以下步骤从亚马逊 EC2 控制台启动带有 AL2 023 AMI 的亚马逊 EC2 实例。

启动带有 AL2 023 AMI 的 EC2 实例

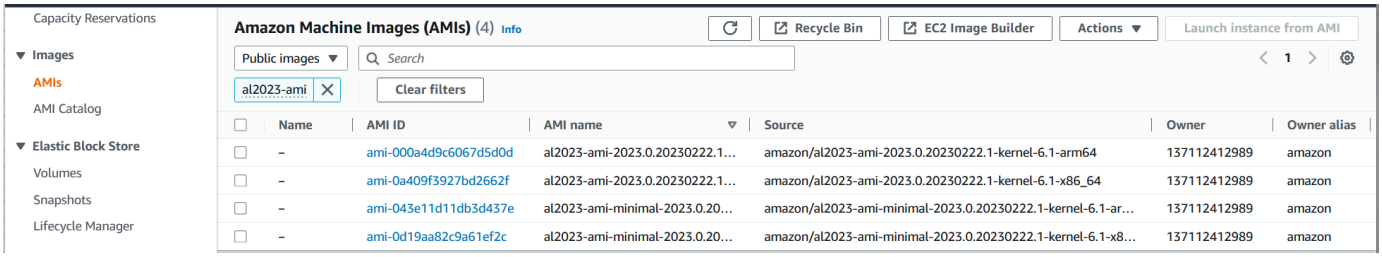
1. 打开 Amazon EC2 控制台，网址为https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，请选择 AMIs。
3. 从下拉菜单中选择公有映像。
4. 在搜索字段中输入 **`al2023-ami`**。

   > **Note**
   >
   > 确保 amazon 显示在拥有者别名列中。

5. 从列表中选择一个映像。在来源下，您可以确定使用标准 AMI 还是AMI 最低版本。可以使用以下格式解释 AL2 023 AMI 名称：

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

6. 下图显示了 AL2 023 AMIs 的部分列表。



有关启动亚马逊 EC2 实例的更多信息，请参阅亚马逊 EC2 用户指南中的亚马逊 EC2 Linux 实例入门。

使用 SSM 参数启动 AL2 023 和 AWS CLI

在中 AWS CLI，您可以使用 AMI 的 SSM 参数值启动新实例 AL2 023。更具体地说，使用以下列表中的一个动态 SSM 参数值，然后在 SSM 参数 value/ 之前添加 `/aws/service/ami-amazon-linux-latest/`。您可以使用以下参数在 AWS CLI中启动实例。

- `al2023-ami-kernel-default-arm64`，适用于 arm64 架构
- `al2023-ami-minimal-kernel-default-arm64`，适用于 arm64 架构（AMI 最低版本）
- `al2023-ami-kernel-default-x86_64`，适用于 x86_64 架构
- `al2023-ami-minimal-kernel-default-x86_64`，适用于 x86_64 架构（AMI 最低版本）

> **Note**
>
> 每个*italic*项目都是一个示例参数。将它们替换为您自己的信息。

```
$ aws ec2 run-instances \
  --image-id \
    resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64 \
  --instance-type m5.xlarge \
  --region us-east-1 \
  --key-name aws-key-us-east-1 \
  --security-group-ids sg-004a7650
```

`--image-id` 标志指定 SSM 参数值。

`--instance-type` 标志指定实例的类型和大小。该标志必须与您选择的 AMI 类型兼容。

该`--region`标志指定您在 AWS 区域 哪里创建实例。

该`--key-name`标志指定 AWS 区域了用于连接到实例的密钥。如果您不提供创建实例的区域中存在的密钥，则无法使用 SSH 连接到该实例。

`--security-group-ids` 标志指定可确定对入站和出站网络流量的访问权限的安全组。

> **Important**
>
> AWS CLI 要求您指定允许通过端口从远程计算机访问实例的现有安全组TCP:22。如果没有指定的安全组，您的新实例将被置于默认安全组中。在默认安全组中，您的实例只能与您的 VPC 中的其他实例连接。

有关更多信息，请参阅AWS Command Line Interface 用户指南中的启动、列出和终止 Amazon EC2 实例。

使用启动最新的 AL2 023 AMI AWS CloudFormation

要使用启动 AL2 023 AMI AWS CloudFormation，请使用以下模板之一。

> **Note**
>
> x86_64和Arm64 AMIs 都需要不同的实例类型。有关更多信息，请参阅 Amazon EC2 实例类型

JSON 模板：

```
{
  "Parameters": {
    "LatestAmiId": {
      "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
      "Default": "/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-
default-x86_64"
    }
  },
  "Resources": {
    "MyEC2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "InstanceType": "t2.large",
        "ImageId": {
          "Ref": "LatestAmiId"
        }
      }
    }
  }
```

```
}
```

YAML 模板：

```
Parameters:
  LatestAmiId:
    Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'
    Default: '/aws/service/ami-amazon-linux-latest/al2023-ami-minimal-kernel-default-
x86_64'

Resources:
  Instance:
    Type: 'AWS::EC2::Instance'
    Properties:
      InstanceType: 't2.large'
      ImageId: !Ref LatestAmiId
```

如果需要，请务必替换“Default”部分末尾的 AMI 参数。可以使用以下参数值：

- `al2023-ami-kernel-6.1-arm64`，适用于 arm64 架构
- `al2023-ami-minimal-kernel-6.1-arm64`，适用于 arm64 架构（AMI 最低版本）
- `al2023-ami-kernel-6.1-x86_64`，适用于 x86_64 架构
- `al2023-ami-minimal-kernel-6.1-x86_64`，适用于 x86_64 架构（AMI 最低版本）

以下是动态内核规范。每次主要内核版本更新时，默认内核版本都会自动更改。

- `al2023-ami-kernel-default-arm64`，适用于 arm64 架构
- `al2023-ami-minimal-kernel-default-arm64`，适用于 arm64 架构（AMI 最低版本）
- `al2023-ami-kernel-default-x86_64`，适用于 x86_64 架构
- `al2023-ami-minimal-kernel-default-x86_64`，适用于 x86_64 架构（AMI 最低版本）

使用特定的 AMI ID 启动 AL2 023

您可以使用 AMI ID 启动特定的 AL2 023 AMI。您可以通过查看亚马逊 EC2 控制台中的 AMI 列表来确定需要哪个 AL2 023 AMI ID。或者，你可以使用 AWS Systems Manager。如果您使用的是 Systems Manager，请务必从上一节中列出的别名中选择 AMI 别名。有关更多信息，请参阅 IDs 使用 AWS Systems Manager 参数存储查询最新的 Amazon Linux AMI。

AL2023 AMI 的弃用和生命周期

每个新的 AL2 023 版本都包含一个新的 AMI。AMI 注册后，会标记有弃用日期。每个 AL2 023 AMI 的弃用日期为自发布之日起 90 天，以匹配为每个内核版本提供的时间段。023 上的 Kernel Live AL2 补丁

> **Note**
>
> 90 天弃用日期是指单个 AMI，并不指 AL2 023 发布频率 或产品支持期。

有关弃用 AMI 的更多信息，请参阅亚马逊 EC2 用户指南中的弃用 AMI。

定期使用更新的 AMI 启动实例，可确保实例启动时有最新的安全更新，包括更新的内核。如果您启动先前版本的 AMI 并应用更新，则该实例在一段时间内会没有最新的安全更新。为确保您使用的是最新的 AMI，建议您使用 SSM 参数。

有关如何使用 SSM 参数启动实例的更多信息，请参阅：

- 使用 SSM 参数启动 AL2 023 和 AWS CLI
- 使用启动最新的 AL2 023 AMI AWS CloudFormation

正在连接到 AL2 023 个实例

使用 SSH 或 AWS Systems Manager 连接到您的 AL2 023 实例。

使用 SSH 连接到您的实例

有关如何使用 SSH 连接实例的说明，请参阅 Amazon EC2 用户指南中的使用 SSH 连接您的 Linux 实例。

使用 Connect 连接到您的实例 AWS Systems Manager

有关如何使用连接 AWS Systems Manager 到 AL2 023 实例的说明，请参阅亚马逊 EC2 用户指南中的使用会话管理器连接您的 Linux 实例。

使用 Amazon EC2 实例 Connect

AL2023 AMI（不包括最小 AMI）默认安装了 Instance EC2 Connect 代理。要将 Instance Connect 与从最小 AMI 启动的 AL2 023 实例一起使用 EC2 ，您必须安装该`ec2-instance-connect`软件包。

有关使用 EC2 实例连接的说明，请参阅亚马逊 EC2 用户指南中的使用 EC2 实例连接连接您的 Linux 实例。

比较 AL2 023 标准版和最小值 AMIs

您可以使用标准（默认）或最低 AL2 023 AMI 启动亚马逊 EC2 实例。有关如何启动标准或最小 AMI 类型的 Amazon EC2 实例的说明，请参阅AL2亚马逊上的 023 EC2。

标准 AL2 023 AMI 安装了所有最常用的应用程序和工具。如果您想快速入门并且对自定义 AMI 不感兴趣，建议您使用标准 AMI。

最小的 AL2 023 AMI 是简化的基本版本，仅包含运行操作系统 (OS) 所需的最基本的工具和实用程序。如果您希望操作系统占用空间尽可能小，建议您使用AMI 最低版本。AMI 最低版本 可以稍微降低磁盘空间占用，提高长期成本效益。如果您想要更小的操作系统并且不介意手动安装工具和应用程序，则 AMI 最低版本 是合适的选项。

容器镜像更接近包集中的 AL2 023 最小 AMI。

比较 Amazon Linux 2023 映像上安装的软件包

AL2023 AMI、Minimal AMI 和容器镜像上的 RPMs 当前镜像的比较。

软件包	AMI	AMI 最低版本	容器
`acl`	2.3.1		
`acpid`	2.0.32		
`alternatives`	1.15	1.15	1.15
`amazon-chrony-config`	4.3	4.3	
`amazon-ec2-net-utils`	2.5.1	2.5.1	
`amazon-linux-repo-cdn`			2023.6.20241031
`amazon-linux-repo-s3`	2023.6.20241031	2023.6.20241031	

软件包	AMI	AMI 最低版本	容器
amazon-linux-sb-keys	2023.1	2023.1	
amazon-rpm-config	228		
amazon-ssm-agent	3.3.987.0		
amd-ucode-firmware	20210208（noarch）	20210208（noarch）	
at	3.1.23		
attr	2.5.1		
audit	3.0.6	3.0.6	
audit-libs	3.0.6	3.0.6	3.0.6
aws-cfn-bootstrap	2.0		
awscli-2	2.15.30	2.15.30	
basesystem	11	11	11
bash	5.2.15	5.2.15	5.2.15
bash-completion	2.1.1		
bc	1.07.1		
bind-libs	9.18.28		
bind-license	9.18.28		
bind-utils	9.18.28		

软件包	AMI	AMI 最低版本	容器
binutils	2.39		
boost-filesystem	1.75.0		
boost-system	1.75.0		
boost-thread	1.75.0		
bzip2	1.0.8		
bzip2-libs	1.0.8	1.0.8	1.0.8
ca-certificates	2023.2.68	2023.2.68	2023.2.68
c-ares	1.19.1		
checkpolicy	3.4	3.4	
chkconfig	1.15		
chrony	4.3	4.3	
cloud-init	22.2	22.2	
cloud-init-cfg-ec2	22.2	22.2	
cloud-utils-growpart	0.31	0.31	
coreutils	8.32	8.32	
coreutils-common	8.32	8.32	
coreutils-single			8.32

软件包	AMI	AMI 最低版本	容器
cpio	2.13	2.13	
cracklib	2.9.6	2.9.6	
cracklib-dicts	2.9.6	2.9.6	
crontabs	1.11		
crypto-policies	20220428	20220428	20220428
crypto-policies-scripts	20220428		
cryptsetup	2.6.1		
cryptsetup-libs	2.6.1	2.6.1	
curl-minimal	8.5.0	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27	
cyrus-sasl-plain	2.1.27		
dbus	1.12.28	1.12.28	
dbus-broker	32	32	
dbus-common	1.12.28	1.12.28	
dbus-libs	1.12.28	1.12.28	
device-mapper	1.02.185	1.02.185	
device-mapper-libs	1.02.185	1.02.185	
diffutils	3.8	3.8	

软件包	AMI	AMI 最低版本	容器
dnf	4.14.0	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2	
dnf-plugins-core	4.3.0	4.3.0	
dnf-plugin-support-info	1.2	1.2	
dnf-utils	4.3.0		
dosfstools	4.2		
dracut	055	055	
dracut-config-ec2	3.0	3.0	
dracut-config-generic	055	055	
dwz	0.14		
dyninst	10.2.1		
e2fsprogs	1.46.5	1.46.5	
e2fsprogs-libs	1.46.5	1.46.5	
ec2-hibinit-agent	1.0.8		

软件包	AMI	AMI 最低版本	容器
ec2-instance-connect	1.1		
ec2-instance-connect-selinux	1.1		
ec2-utils	2.2.0	2.2.0	
ed	1.14.2		
efi-filesystem	5	5	
efi-srpm-macros	5		
efivar	38	38	
efivar-libs	38	38	
elfutils-debuginfod-client	0.188		
elfutils-default-yama-scope	0.188	0.188	0.188
elfutils-libelf	0.188	0.188	0.188
elfutils-libs	0.188	0.188	0.188
ethtool	5.15		
expat	2.5.0	2.5.0	2.5.0
file	5.39	5.39	
file-libs	5.39	5.39	5.39

软件包	AMI	AMI 最低版本	容器
filesystem	3.14	3.14	3.14
findutils	4.8.0	4.8.0	
fonts-srpm-macros	2.0.5		
fstrm	0.6.1		
fuse-libs	2.9.9	2.9.9	
gawk	5.1.0	5.1.0	5.1.0
gdbm-libs	1.19	1.19	1.19
gdisk	1.0.8	1.0.8	
gettext	0.21	0.21	
gettext-libs	0.21	0.21	
ghc-srpm-macros	1.5.0		
glib2	2.74.7	2.74.7	2.74.7
glibc	2.34	2.34	2.34
glibc-all-langpacks	2.34	2.34	
glibc-common	2.34	2.34	2.34
glibc-gconv-extra	2.34		
glibc-locale-source	2.34	2.34	

软件包	AMI	AMI 最低版本	容器
glibc-minimal-langpack			2.34
gmp	6.2.1	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7	2.3.7
gnutls	3.8.0	3.8.0	
go-srpm-macros	3.2.0		
gpgme	1.15.1	1.15.1	1.15.1
gpm-libs	1.20.7		
grep	3.8	3.8	3.8
groff-base	1.22.4	1.22.4	
grub2-common	2.06	2.06	
grub2-efi-aa64-ec2	2.06 (aarch64)	2.06 (aarch64)	
grub2-efi-x64-ec2	2.06 (x86_64)	2.06 (x86_64)	
grub2-pc-modules	2.06	2.06	
grub2-tools	2.06	2.06	
grub2-tools-minimal	2.06	2.06	
grubby	8.40	8.40	
gssproxy	0.8.4		

软件包	AMI	AMI 最低版本	容器
gzip	1.12	1.12	
hostname	3.23	3.23	
hunspell	1.7.0		
hunspell-en	0.20140811.1		
hunspell-en-GB	0.20140811.1		
hunspell-en-US	0.20140811.1		
hunspell-filesystem	1.7.0		
hwdata	0.384	0.384	
info	6.7		
inih	49	49	
initscripts	10.09	10.09	
iproute	6.10.0	6.10.0	
iputils	20210202	20210202	
irqbalance	1.9.0	1.9.0	
jansson	2.14	2.14	
jemalloc	5.2.1		
jitterentropy	3.4.1	3.4.1	
jq	1.7.1	1.7.1	
json-c	0.14	0.14	0.14
kbd	2.4.0	2.4.0	

软件包	AMI	AMI 最低版本	容器
kbd-misc	2.4.0	2.4.0	
kernel	6.1.112	6.1.112	
kernel-libbpf	6.1.112	6.1.112	
kernel-livepatch-repo-s3	2023.6.20241031	2023.6.20241031	
kernel-srpm-macros	1.0		
kernel-tools	6.1.112		
keyutils	1.6.3		
keyutils-libs	1.6.3	1.6.3	1.6.3
kmod	29	29	
kmod-libs	29	29	
kpatch-runtime	0.9.7		
krb5-libs	1.21.3	1.21.3	1.21.3
less	608	608	
libacl	2.3.1	2.3.1	2.3.1
libaio	0.3.111		
libarchive	3.7.4	3.7.4	3.7.4
libargon2	20171227	20171227	
libassuan	2.5.5	2.5.5	2.5.5

软件包	AMI	AMI 最低版本	容器
libattr	2.5.1	2.5.1	2.5.1
libbasicobjects	0.1.1		
libblkid	2.37.4	2.37.4	2.37.4
libcap	2.48	2.48	2.48
libcap-ng	0.8.2	0.8.2	0.8.2
libcbor	0.7.0	0.7.0	
libcollection	0.7.0		
libcom_err	1.46.5	1.46.5	1.46.5
libcomps	0.1.20	0.1.20	0.1.20
libconfig	1.7.2		
libcurl-minimal	8.5.0	8.5.0	8.5.0
libdb	5.3.28	5.3.28	
libdhash	0.5.0		
libdnf	0.69.0	0.69.0	0.69.0
libeconf	0.4.0	0.4.0	
libedit	3.1	3.1	
libev	4.33		
libevent	2.1.12		
libfdisk	2.37.4	2.37.4	
libffi	3.4.4	3.4.4	3.4.4

软件包	AMI	AMI 最低版本	容器
libfido2	1.10.0	1.10.0	
libgcc	11.4.1	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2	1.10.2
libgomp	11.4.1	11.4.1	11.4.1
libgpg-error	1.42	1.42	1.42
libibverbs	48.0		
libidn2	2.3.2	2.3.2	2.3.2
libini_config	1.3.1		
libkcapi	1.4.0	1.4.0	
libkcapi-hmaccalc	1.4.0	1.4.0	
libldb	2.6.2		
libmaxminddb	1.5.2		
libmetalink	0.1.3		
libmnl	1.0.4	1.0.4	
libmodulemd	2.13.0	2.13.0	2.13.0
libmount	2.37.4	2.37.4	2.37.4
libnfsidmap	2.5.4		
libnghttp2	1.59.0	1.59.0	1.59.0
libnl3	3.5.0		
libpath_utils	0.2.1		

软件包	AMI	AMI 最低版本	容器
libpcap	1.10.1		
libpipeline	1.5.3	1.5.3	
libpkgconf	1.8.0		
libpsl	0.21.1	0.21.1	0.21.1
libpwquality	1.4.4	1.4.4	
libref_array	0.1.5		
librepo	1.14.5	1.14.5	1.14.5
libreport-filesystem	2.15.2	2.15.2	2.15.2
libseccomp	2.5.3	2.5.3	
libselinux	3.4	3.4	3.4
libselinux-utils	3.4	3.4	
libsemanage	3.4	3.4	
libsepol	3.4	3.4	3.4
libsigsegv	2.13	2.13	2.13
libsmartcols	2.37.4	2.37.4	2.37.4
libsolv	0.7.22	0.7.22	0.7.22
libss	1.46.5	1.46.5	
libsss_certmap	2.9.4		
libsss_idmap	2.9.4		

软件包	AMI	AMI 最低版本	容器
libsss_nss_idmap	2.9.4		
libsss_sudo	2.9.4		
libstdc++	11.4.1	11.4.1	11.4.1
libstoragemgmt	1.9.4		
libtalloc	2.3.4		
libtasn1	4.19.0	4.19.0	4.19.0
libtdb	1.4.7		
libtevent	0.13.0		
libtextstyle	0.21	0.21	
libtirpc	1.3.3		
libunistring	0.9.10	0.9.10	0.9.10
libuser	0.63	0.63	
libutempter	1.2.1	1.2.1	
libuuid	2.37.4	2.37.4	2.37.4
libuv	1.47.0		
libverto	0.3.2	0.3.2	0.3.2
libverto-libev	0.3.2		
libxcrypt	4.4.33	4.4.33	4.4.33
libxml2	2.10.4	2.10.4	2.10.4
libyaml	0.2.5	0.2.5	0.2.5

软件包	AMI	AMI 最低版本	容器
libzstd	1.5.5	1.5.5	1.5.5
linux-firmware-whence	20210208（noarch）	20210208（noarch）	
lm_sensors-libs	3.6.0		
lmdb-libs	0.9.29		
logrotate	3.20.1	3.20.1	
lsof	4.94.0		
lua-libs	5.4.4	5.4.4	5.4.4
lua-srpm-macros	1		
lz4-libs	1.9.4	1.9.4	1.9.4
man-db	2.9.3	2.9.3	
man-pages	5.10		
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)	
mpfr	4.1.0	4.1.0	4.1.0
nano	5.8		
ncurses	6.2	6.2	
ncurses-base	6.2	6.2	6.2
ncurses-libs	6.2	6.2	6.2
nettle	3.8	3.8	
net-tools	2.0	2.0	
newt	0.52.21		

软件包	AMI	AMI 最低版本	容器
nfs-utils	2.5.4		
npth	1.6	1.6	1.6
nspr	4.35.0		
nss	3.90.0		
nss-softokn	3.90.0		
nss-softokn-freebl	3.90.0		
nss-sysinit	3.90.0		
nss-util	3.90.0		
ntsysv	1.15		
numactl-libs	2.0.14	2.0.14	
ocaml-srpm-macros	6		
oniguruma	6.9.7.1	6.9.7.1	
openblas-srpm-macros	2		
openldap	2.4.57	2.4.57	
openssh	8.7p1	8.7p1	
openssh-clients	8.7p1	8.7p1	
openssh-server	8.7p1	8.7p1	
openssl	3.0.8	3.0.8	

软件包	AMI	AMI 最低版本	容器
openssl-libs	3.0.8	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12	
os-prober	1.77	1.77	
p11-kit	0.24.1	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1	0.24.1
package-notes-srpm-macros	0.4		
pam	1.5.1	1.5.1	
parted	3.4		
passwd	0.80	0.80	
pciutils	3.7.0	3.7.0	
pciutils-libs	3.7.0	3.7.0	
pcre2	10.40	10.40	10.40
pcre2-syntax	10.40	10.40	10.40
perl-Carp	1.50		
perl-Class-Struct	0.66		
perl-constant	1.33		
perl-DynaLoader	1.47		
perl-Encode	3.15		
perl-Errno	1.30		

软件包	AMI	AMI 最低版本	容器
perl-Exporter	5.74		
perl-Fcntl	1.13		
perl-File-Basename	2.85		
perl-File-Path	2.18		
perl-File-stat	1.09		
perl-File-Temp	0.231.100		
perl-Getopt-Long	2.52		
perl-Getopt-Std	1.12		
perl-HTTP-Tiny	0.078		
perl-if	0.60.800		
perl-interpreter	5.32.1		
perl-IO	1.43		
perl-IPC-Open3	1.21		
perl-libs	5.32.1		
perl-MIME-Base64	3.16		
perl-mro	1.23		
perl-overload	1.31		

软件包	AMI	AMI 最低版本	容器
perl-overloading	0.02		
perl-parent	0.238		
perl-PathTools	3.78		
perl-Pod-Escapes	1.07		
perl-podlators	4.14		
perl-Pod-Perldoc	3.28.01		
perl-Pod-Simple	3.42		
perl-Pod-Usage	2.01		
perl-POSIX	1.94		
perl-Scalar-List-Utils	1.56		
perl-SelectSaver	1.02		
perl-Socket	2.032		
perl-srpm-macros	1		
perl-Storable	3.21		
perl-subs	1.03		
perl-Symbol	1.08		

软件包	AMI	AMI 最低版本	容器
perl-Term-ANSIColor	5.01		
perl-Term-Cap	1.17		
perl-Text-ParseWords	3.30		
perl-Text-Tabs+Wrap	2021.0726		
perl-Time-Local	1.300		
perl-vars	1.05		
pkgconf	1.8.0		
pkgconf-m4	1.8.0		
pkgconf-pkg-config	1.8.0		
policycoreutils	3.4	3.4	
policycoreutils-python-utils	3.4		
popt	1.18	1.18	1.18
procps-ng	3.3.17	3.3.17	
protobuf-c	1.4.1		
psacct	6.6.4		
psmisc	23.4	23.4	

软件包	AMI	AMI 最低版本	容器
publicsuffix-list-dafsa	20240212	20240212	20240212
python3	3.9.16	3.9.16	3.9.16
python3-attrs	20.3.0	20.3.0	
python3-audit	3.0.6	3.0.6	
python3-awscrt	0.19.19	0.19.19	
python3-babel	2.9.1	2.9.1	
python3-cffi	1.14.5	1.14.5	
python3-chardet	4.0.0	4.0.0	
python3-colorama	0.4.4	0.4.4	
python3-configobj	5.0.6	5.0.6	
python3-cryptography	36.0.1	36.0.1	
python3-daemon	2.3.0		
python3-dateutil	2.8.1	2.8.1	
python3-dbus	1.2.18	1.2.18	
python3-distro	1.5.0	1.5.0	
python3-dnf	4.14.0	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0	

软件包	AMI	AMI 最低版本	容器
python3-docutils	0.16	0.16	
python3-gpg	1.15.1	1.15.1	1.15.1
python3-hawkey	0.69.0	0.69.0	0.69.0
python3-idna	2.10	2.10	
python3-jinja2	2.11.3	2.11.3	
python3-jmespath	0.10.0	0.10.0	
python3-jsonpatch	1.21	1.21	
python3-jsonpointer	2.0	2.0	
python3-jsonschema	3.2.0	3.2.0	
python3-libcomps	0.1.20	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0	0.69.0
python3-libs	3.9.16	3.9.16	3.9.16
python3-libselinux	3.4	3.4	
python3-libsemanage	3.4	3.4	
python3-libstoragemgmt	1.9.4		

软件包	AMI	AMI 最低版本	容器
python3-lockfile	0.12.2		
python3-markupsafe	1.1.1	1.1.1	
python3-netifaces	0.10.6	0.10.6	
python3-oauthlib	3.0.2	3.0.2	
python3-pip-wheel	21.3.1	21.3.1	21.3.1
python3-ply	3.11	3.11	
python3-policycoreutils	3.4	3.4	
python3-prettytable	0.7.2	0.7.2	
python3-prompt-toolkit	3.0.24	3.0.24	
python3-pycparser	2.20	2.20	
python3-pyrsistent	0.17.3	0.17.3	
python3-pyserial	3.4	3.4	
python3-pysocks	1.7.1	1.7.1	
python3-pytz	2022.7.1	2022.7.1	

软件包	AMI	AMI 最低版本	容器
python3-pyyaml	5.4.1	5.4.1	
python3-requests	2.25.1	2.25.1	
python3-rpm	4.16.1.3	4.16.1.3	4.16.1.3
python3-ruamel-yaml	0.16.6	0.16.6	
python3-ruamel-yaml-clib	0.1.2	0.1.2	
python3-setools	4.4.1	4.4.1	
python3-setuptools	59.6.0	59.6.0	
python3-setuptools-wheel	59.6.0	59.6.0	59.6.0
python3-six	1.15.0	1.15.0	
python3-systemd	235	235	
python3-urllib3	1.25.10	1.25.10	
python3-wcwidth	0.2.5	0.2.5	
python-chevron	0.13.1		
python-srpm-macros	3.9		
quota	4.06		
quota-nls	4.06		

软件包	AMI	AMI 最低版本	容器
readline	8.1	8.1	8.1
rng-tools	6.14	6.14	
rootfiles	8.1	8.1	
rpcbind	1.2.6		
rpm	4.16.1.3	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3	
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3	
rpm-sign-libs	4.16.1.3	4.16.1.3	4.16.1.3
rsync	3.2.6		
rust-srpm-macros	21		
sbsigntools	0.9.4	0.9.4	
screen	4.8.0		
sed	4.8	4.8	4.8
selinux-policy	38.1.45	38.1.45	
selinux-policy-targeted	38.1.45	38.1.45	
setup	2.13.7	2.13.7	2.13.7

软件包	AMI	AMI 最低版本	容器
shadow-utils	4.9	4.9	
slang	2.3.2		
sqlite-libs	3.40.0	3.40.0	3.40.0
sssd-client	2.9.4		
sssd-common	2.9.4		
sssd-kcm	2.9.4		
sssd-nfs-idmap	2.9.4		
strace	6.8		
sudo	1.9.15	1.9.15	
sysctl-defaults	1.0	1.0	
sysstat	12.5.6		
systemd	252.23	252.23	
systemd-libs	252.23	252.23	
systemd-networkd	252.23	252.23	
systemd-pam	252.23	252.23	
systemd-resolved	252.23	252.23	
systemd-udev	252.23	252.23	
system-release	2023.6.20241031	2023.6.20241031	2023.6.20241031

软件包	AMI	AMI 最低版本	容器
systemtap-runtime	4.8		
tar	1.34	1.34	
tbb	2020.3		
tcpdump	4.99.1		
tcsh	6.24.07		
time	1.9		
traceroute	2.1.3		
tzdata	2024a	2024a	2024a
unzip	6.0		
update-motd	2.2	2.2	
userspace-rcu	0.12.1	0.12.1	
util-linux	2.37.4	2.37.4	
util-linux-core	2.37.4	2.37.4	
vim-common	9.0.2153		
vim-data	9.0.2153	9.0.2153	
vim-enhanced	9.0.2153		
vim-filesystem	9.0.2153		
vim-minimal	9.0.2153	9.0.2153	
wget	1.21.3		
which	2.21	2.21	

软件包	AMI	AMI 最低版本	容器
words	3.0		
xfsdump	3.1.11		
xfsprogs	5.18.0	5.18.0	
xxd	9.0.2153		
xxhash-libs	0.8.0		
xz	5.2.5	5.2.5	
xz-libs	5.2.5	5.2.5	5.2.5
yum	4.14.0	4.14.0	4.14.0
zip	3.0		
zlib	1.2.11	1.2.11	1.2.11
zram-generator	1.1.2	1.1.2	
zram-generator-defaults	1.1.2	1.1.2	
zstd	1.5.5	1.5.5	

在容器中使用 AL2 023

> **Note**
>
> 有关如何使用 AL2 023 在 Amazon ECS 上托管容器化工作负载的更多信息，请参阅。AL2023 适用于亚马逊 ECS 容器主机

根据用例，有几种方法可以在容器内使用 AL2 023。与AL2023 基础容器镜像亚马逊 Linux 2 容器镜像和 AL2 023 最小 AMI 最为相似。

对于高级用户，我们提供了 AL2 023.2 版本中引入的最小容器镜像，以及描述如何构建基本容器的文档。

AL2023 还可用于托管容器化工作负载，可以是基于 AL2 023 的容器映像，也可以是基于其他 Linux 发行版的容器。您可以使用 AL2023 适用于亚马逊 ECS 容器主机，也可以直接使用提供的容器运行时包。`dockercontainerd`、和`nerdctl`软件包可在 AL2 023 上安装和使用。

主题

- 使用 AL2 023 基础容器镜像
- AL2023 最小容器镜像
- 构建基本系统 AL2 023 容器镜像
- 比较 Amazon Linux 2023 容器映像上安装的软件包
- 比较 Amazon Linux 2023 AMI 最低版本 和容器映像上安装的软件包

使用 AL2 023 基础容器镜像

AL2023 容器镜像由 AL2 023 AMI 中包含的相同软件组件构建。作为 Docker 工作负载的基本映像，它可用在任何环境中。如果您将亚马逊 Linux AMI 用于亚马逊弹性计算云（亚马逊 EC2）中的应用程序，则可以使用亚马逊 Linux 容器镜像对应用程序进行容器化。

在本地开发环境中使用 Amazon Linux 容器镜像，然后使用亚马逊弹性容器服务 (Amazon ECS) 将您的应用程序推送到 AWS 使用亚马逊弹性容器服务 (Amazon ECS)。有关更多信息，请参阅《Amazon Elastic Container Registry 用户指南》中的将 Amazon ECR 映像与 Amazon ECS 结合使用。

Amazon Linux 容器映像在 Amazon ECR Public 上可用。您可以通过您的指定 AWS 代表或通过在亚马逊 linux- 2023 存储库中提交问题来提供有关 AL2 023 的反馈。 GitHub

从 Amazon ECR Public 中提取 Amazon Linux 容器映像

1. 对您的 Amazon Linux Public 注册表进行 Docker 客户端身份验证。验证令牌的有效期为 12 小时。有关更多信息，请参阅《Amazon Elastic Container Registry 用户指南》中的私有注册表身份验证。

   > **Note**
   >
   > 最新版本的 AWS CLI 版本 2 支持该`get-login-password`命令。有关更多信息，请参阅AWS Command Line Interface 《用户指南》中的安装 AWS Command Line Interface。

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

输出如下所示。

```
Login succeeded
```

2. 运行 docker pull 命令以拉取 Amazon Linux 容器映像。要在 Amazon ECR 公开映像浏览馆中查看 Amazon Linux 容器映像，请参阅 Amazon ECR 公开映像浏览馆 - amazonlinux。

> **Note**
>
> 拉取 AL2 023 Docker 容器镜像时，可以使用以下格式之一的标签：
>
> - 要获取最新版本的 AL2 023 容器镜像，请使用`:2023`标签。
> - 要获取 AL2 023 的特定版本，您可以使用以下格式：
> - `:2023.`*`[0-7 release quarter]`*`.`*`[release date]`*`.`*`[build number]`*
>
> 以下示例使用标签`:2023`并提取最新的可用容器镜像 AL2 023。

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023
```

3. （可选）在本地运行容器。

```
$ docker run -it --security-opt seccomp=unconfined public.ecr.aws/amazonlinux/
amazonlinux:2023 /bin/bash
```

从 Docker Hub 中提取 AL2 023 容器镜像

1. 使用`docker pull`命令提取 AL2 023 容器镜像。

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

2. （可选）在本地运行容器。

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

> **Note**
>
> AL2023 的容器镜像仅使用软件dnf包管理器来安装软件包。这意味着没有 `amazon-linux-extras` 或等效的命令可以用于其他软件。

AL2023 最小容器镜像

> **Note**
>
> 标准的 AL2 023 容器镜像适用于大多数用例，适应最小的容器镜像可能比适应 AL2 023 基础容器镜像要花更多的精力。

AL20 AL2 23.2 中引入的 023 最小容器镜像与基础容器镜像不同，因为它只包含安装其他软件包所需的最低限度的软件包。最小容器镜像被设计成一组最小的软件包，而不是一组方便的软件包。

AL2023 最小容器镜像由 AL2 023 中已有的软件组件构建。最小容器镜像的主要区别在于使用`microdnf`dnf软件包管理器而不是功能Python齐全的软件包管理器dnf。这样可以缩小最小容器映像，但要权衡一下没有 AL2 023 AMIs 和基础容器镜像中包含的dnf软件包管理器的完整功能集。

AL2023 最小容器镜像构成了 Lambda 运行时`provided.al2023` AWS 环境的基础。

有关最小容器镜像中包含的软件包的详细列表，请参阅比较 Amazon Linux 2023 容器映像上安装的软件包。

最小容器映像大小

由于 AL2 023 最小容器镜像包含的包比 AL2 023 基础容器镜像少，因此它也要小得多。下表比较了当前和过去版本的 Amazon Linux 的容器镜像选项。

> **Note**
>
> 映像大小如 Amazon ECR 公开映像浏览馆上的 Amazon Linux 所示。

图像	版本	映像大小	注意
亚马逊 Linux (1AL1)	2018.03.0.20230918.0	62.3MB	仅限 x86-64
Amazon Linux 2	2.0.20230926.0	64.2MB	aarch64 比 x86-64 大 1.6MB
Amazon Linux 2023 基本容器映像	2023.2.20231002.0	52.4MB	
Amazon Linux 2023 最小容器映像	2023.2.20231002.0-minimal	35.2MB	

使用 AL2 023 最小容器镜像

AL2023 最小容器镜像在上ECR可用，2023-minimal标签将始终指向基于 AL2 023 的最小容器镜像，而minimal标签可能会更新为比 AL2 023 更新的亚马逊 Linux 版本。

您可以使用以下示例提取这些标签：docker

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:minimal
```

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023-minimal
```

以下示例显示了Dockerfile获取最小容器镜像并在其上安装 GCC 的：

```
FROM public.ecr.aws/amazonlinux/amazonlinux:2023-minimal
RUN dnf install -y gcc && dnf clean all
```

构建基本系统 AL2 023 容器镜像

AL2023 容器镜像由 AL2 023 AMI 中包含的相同软件组件构建。它包括一个软件，使基础容器层的行为与在 Amazon EC2 实例上运行的行为类似，例如包管理器dnf。本节介绍如何从头开始构建仅包含应用程序所需的最低依赖项的容器。

> **Note**
>
> 标准 AL2 023 容器镜像适用于大多数用例。使用标准容器映像可以轻松地在映像之上进行构建。基本容器镜像使得在镜像之上构建变得更加困难。

针对应用程序创建具有最少依赖项的容器

1. 确定运行时依赖项。这将因您的应用程序而异。
2. 构造一个构建 `FROM scratch` 的 `Dockerfile/Containerfile`。`Dockerfile` 的以下示例可用于构建仅包含 `bash` shell 及其依赖项的容器。

```
FROM public.ecr.aws/amazonlinux/amazonlinux:2023 as build
RUN mkdir /sysroot
RUN dnf --releasever=$(rpm -q system-release --qf '%{VERSION}') \
  --installroot /sysroot \
  -y \
  --setopt=install_weak_deps=False \
  install bash

FROM scratch
COPY --from=build /sysroot /
WORKDIR /
ENTRYPOINT ["/bin/bash"]
```

- `Dockerfile` 的运行方式是：
  1. 启动一个名`build`为的 AL2 023 容器。该容器将用于引导基本容器，该容器本身不会部署，而会生成要部署的容器。
  2. 创建 `/sysroot` 目录。该目录将是 `build` 容器安装基本容器所需的依赖项的地方。在接下来的步骤中，`/sysroot` 路径将被打包为我们的基本映像的根目录。

     以这种`dnf`方式使用`--installroot`选项就是我们创建其他 AL2 023 图像的方式。`dnf` 的一项功能使得安装程序和映像创建工具能够正常工作。
  3. 调用 `dnf` 以将软件包安装到 `/sysroot`。

`rpm -q system-release --qf '%{VERSION}'` 命令会查询 (`-q`) `system-release` 软件包，设置查询格式 (`--qf`) 以打印出所查询的软件包的版本（`%{VERSION}` 变量是 RPM 版本的 `rpm` 变量）。

通过将 `dnf` 的 `--releasever` 参数设置为 `build` 容器中的 `system-release` 版本，每当发布更新的 Amazon Linux 容器基本映像时，都可以使用 `Dockerfile` 来重建基本容器。

可以将其设置为任何亚马逊 Linux 2023 版本，例如 2023.7.20250331。`--releasever`这样做意味着`build`容器将作为最新的 023 版本运行，但无论当前的 AL2 023 版本是什么，都要从 2023.7.20250331 开始构建准系统容器。 AL2

`--setopt=install_weak_deps=False` 配置选项可告诉 `dnf` 只安装必需 的依赖项，而不是推荐或建议的依赖项。

4. 将已安装的系统复制到空白 (`FROM scratch`) 容器的根目录中。
5. 在本例 `/bin/bash` 中，将 `ENTRYPOINT` 设置为所需的二进制文件。

3. 创建一个空目录，将步骤 2 中示例的内容添加到名为 `Dockerfile` 的文件中。

```
$ mkdir al2023-barebones-bash-example
 $ cd al2023-barebones-bash-example
 $ cat > Dockerfile <<EOF
FROM public.ecr.aws/amazonlinux/amazonlinux:2023 as build
RUN mkdir /sysroot
RUN dnf --releasever=$(rpm -q system-release --qf '%{VERSION}') \
  --installroot /sysroot \
  -y \
  --setopt=install_weak_deps=False \
  install bash && dnf --installroot /sysroot clean all

FROM scratch
COPY --from=build /sysroot /
WORKDIR /
ENTRYPOINT ["/bin/bash"]
EOF
```

4. 通过运行以下命令构建容器。

```
$ docker build -t al2023-barebones-bash-example
```

5. 使用以下命令运行容器，了解仅限 bash 的最小容器的运行方式。

```
$ docker run -it --rm al2023-barebones-bash-example
bash-5.2# rpm
bash: rpm: command not found
bash-5.2# du -sh /usr/
bash: du: command not found
bash-5.2# ls
bash: ls: command not found
bash-5.2# echo /bin/*
/bin/alias /bin/bash /bin/bashbug /bin/bashbug-64 /bin/bg /bin/catchsegv /bin/cd /
bin/command /bin/fc /bin/fg /bin/gencat /bin/getconf /bin/getent /bin/getopts /
bin/hash /bin/iconv /bin/jobs /bin/ld.so /bin/ldd /bin/locale /bin/localedef /
bin/pldd /bin/read /bin/sh /bin/sotruss /bin/sprof /bin/type /bin/tzselect /bin/
ulimit /bin/umask /bin/unalias /bin/wait /bin/zdump
```

举一个更实际的例子，以下过程可为显示 Hello World! 的一个 C 应用程序构建一个容器。

1. 创建一个空目录并添加 C 源代码和 Dockerfile。

```
$ mkdir al2023-barebones-c-hello-world-example
$ cd al2023-barebones-c-hello-world-example
$ cat > hello-world.c <<EOF
#include <stdio.h>
int main(void)
{
  printf("Hello World!\n");
  return 0;
}
EOF

$ cat > Dockerfile <<EOF
FROM public.ecr.aws/amazonlinux/amazonlinux:2023 as build
COPY hello-world.c /
RUN dnf -y install gcc
RUN gcc -o hello-world hello-world.c
RUN mkdir /sysroot
RUN mv hello-world /sysroot/
RUN dnf --releasever=$(rpm -q system-release --qf '%{VERSION}') \
  --installroot /sysroot \
  -y \
  --setopt=install_weak_deps=False \
```

```
  install glibc && dnf --installroot /sysroot clean all

FROM scratch
COPY --from=build /sysroot /
WORKDIR /
ENTRYPOINT ["/hello-world"]
EOF
```

2. 使用以下命令构建容器。

```
$ docker build -t al2023-barebones-c-hello-world-example .
```

3. 使用以下命令运行容器。

```
$ docker run -it --rm al2023-barebones-c-hello-world-example
Hello World!
```

比较 Amazon Linux 2023 容器映像上安装的软件包

将 AL2 023 基础容器映像上的 RPMs 当前图像与 AL2 023 最小容器镜像上的 RPMs 当前图像进行比较。

软件包	容器	最小容器
alternatives	1.15	1.15
amazon-linux-repo-cdn	2023.6.20241031	2023.6.20241031
audit-libs	3.0.6	3.0.6
basesystem	11	11
bash	5.2.15	5.2.15
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023.2.68	2023.2.68

软件包	容器	最小容器
coreutils-single	8.32	8.32
crypto-policies	20220428	20220428
curl-minimal	8.5.0	8.5.0
dnf	4.14.0	
dnf-data	4.14.0	4.14.0
elfutils-default-yama-scope	0.188	
elfutils-libelf	0.188	
elfutils-libs	0.188	
expat	2.5.0	
file-libs	5.39	5.39
filesystem	3.14	3.14
gawk	5.1.0	5.1.0
gdbm-libs	1.19	
glib2	2.74.7	2.74.7
glibc	2.34	2.34
glibc-common	2.34	2.34
glibc-minimal-langpack	2.34	2.34
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7

软件包	容器	最小容器
gobject-introspection		1.73.0
gpgme	1.15.1	1.15.1
grep	3.8	3.8
json-c	0.14	0.14
keyutils-libs	1.6.3	1.6.3
krb5-libs	1.21.3	1.21.3
libacl	2.3.1	2.3.1
libarchive	3.7.4	3.7.4
libassuan	2.5.5	2.5.5
libattr	2.5.1	2.5.1
libblkid	2.37.4	2.37.4
libcap	2.48	2.48
libcap-ng	0.8.2	0.8.2
libcom_err	1.46.5	1.46.5
libcomps	0.1.20	
libcurl-minimal	8.5.0	8.5.0
libdnf	0.69.0	0.69.0
libffi	3.4.4	3.4.4
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2

软件包	容器	最小容器
libgomp	11.4.1	
libgpg-error	1.42	1.42
libidn2	2.3.2	2.3.2
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libnghttp2	1.59.0	1.59.0
libpeas		1.32.0
libpsl	0.21.1	0.21.1
librepo	1.14.5	1.14.5
libreport-filesystem	2.15.2	2.15.2
libselinux	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2.13	2.13
libsmartcols	2.37.4	2.37.4
libsolv	0.7.22	0.7.22
libstdc++	11.4.1	11.4.1
libtasn1	4.19.0	4.19.0
libunistring	0.9.10	0.9.10
libuuid	2.37.4	2.37.4
libverto	0.3.2	0.3.2

软件包	容器	最小容器
libxcrypt	4.4.33	
libxml2	2.10.4	2.10.4
libyaml	0.2.5	0.2.5
libzstd	1.5.5	1.5.5
lua-libs	5.4.4	5.4.4
lz4-libs	1.9.4	1.9.4
microdnf		3.10.0
microdnf-dnf		3.10.0
mpfr	4.1.0	4.1.0
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
npth	1.6	1.6
openssl-libs	3.0.8	3.0.8
p11-kit	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1
pcre2	10.40	10.40
pcre2-syntax	10.40	10.40
popt	1.18	1.18
publicsuffix-list-dafsa	20240212	20240212
python3	3.9.16	

软件包	容器	最小容器
python3-dnf	4.14.0	
python3-gpg	1.15.1	
python3-hawkey	0.69.0	
python3-libcomps	0.1.20	
python3-libdnf	0.69.0	
python3-libs	3.9.16	
python3-pip-wheel	21.3.1	
python3-rpm	4.16.1.3	
python3-setuptools-wheel	59.6.0	
readline	8.1	8.1
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	
rpm-libs	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	
sed	4.8	4.8
setup	2.13.7	2.13.7
sqlite-libs	3.40.0	3.40.0
system-release	2023.6.20241031	2023.6.20241031
tzdata	2024a	
xz-libs	5.2.5	5.2.5

软件包	容器	最小容器
yum	4.14.0	
zlib	1.2.11	1.2.11

比较 Amazon Linux 2023 AMI 最低版本 和容器映像上安装的软件包

将 023 Minimial AMI 上的 RPMs 当前 AL2图像与 AL2 023 基础镜像和最小 RPMs 容器镜像上的当前图像进行比较。

软件包	AMI 最低版本	容器	最小容器
alternatives	1.15	1.15	1.15
amazon-chrony-config	4.3		
amazon-ec2-net-utils	2.5.1		
amazon-linux-repo-cdn		2023.6.20241031	2023.6.20241031
amazon-linux-repo-s3	2023.6.20241031		
amazon-linux-sb-keys	2023.1		
amd-ucode-firmware	20210208（noarch）		
audit	3.0.6		
audit-libs	3.0.6	3.0.6	3.0.6
awscli-2	2.15.30		

软件包	AMI 最低版本	容器	最小容器
basesystem	11	11	11
bash	5.2.15	5.2.15	5.2.15
bzip2-libs	1.0.8	1.0.8	1.0.8
ca-certificates	2023.2.68	2023.2.68	2023.2.68
checkpolicy	3.4		
chrony	4.3		
cloud-init	22.2		
cloud-init-cfg-ec2	22.2		
cloud-utils-growpart	0.31		
coreutils	8.32		
coreutils-common	8.32		
coreutils-single		8.32	8.32
cpio	2.13		
cracklib	2.9.6		
cracklib-dicts	2.9.6		
crypto-policies	20220428	20220428	20220428
cryptsetup-libs	2.6.1		
curl-minimal	8.5.0	8.5.0	8.5.0

软件包	AMI 最低版本	容器	最小容器
cyrus-sasl-lib	2.1.27		
dbus	1.12.28		
dbus-broker	32		
dbus-common	1.12.28		
dbus-libs	1.12.28		
device-mapper	1.02.185		
device-mapper-libs	1.02.185		
diffutils	3.8		
dnf	4.14.0	4.14.0	
dnf-data	4.14.0	4.14.0	4.14.0
dnf-plugin-release-notification	1.2		
dnf-plugins-core	4.3.0		
dnf-plugin-support-info	1.2		
dracut	055		
dracut-config-ec2	3.0		
dracut-config-generic	055		

软件包	AMI 最低版本	容器	最小容器
e2fsprogs	1.46.5		
e2fsprogs-libs	1.46.5		
ec2-utils	2.2.0		
efi-filesystem	5		
efivar	38		
efivar-libs	38		
elfutils-default-yama-scope	0.188	0.188	
elfutils-libelf	0.188	0.188	
elfutils-libs	0.188	0.188	
expat	2.5.0	2.5.0	
file	5.39		
file-libs	5.39	5.39	5.39
filesystem	3.14	3.14	3.14
findutils	4.8.0		
fuse-libs	2.9.9		
gawk	5.1.0	5.1.0	5.1.0
gdbm-libs	1.19	1.19	
gdisk	1.0.8		
gettext	0.21		

软件包	AMI 最低版本	容器	最小容器
gettext-libs	0.21		
glib2	2.74.7	2.74.7	2.74.7
glibc	2.34	2.34	2.34
glibc-all-langpacks	2.34		
glibc-common	2.34	2.34	2.34
glibc-locale-source	2.34		
glibc-minimal-langpack		2.34	2.34
gmp	6.2.1	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7	2.3.7
gnutls	3.8.0		
gobject-introspection			1.73.0
gpgme	1.15.1	1.15.1	1.15.1
grep	3.8	3.8	3.8
groff-base	1.22.4		
grub2-common	2.06		
grub2-efi-aa64-ec2	2.06 (aarch64)		
grub2-efi-x64-ec2	2.06 (x86_64)		

软件包	AMI 最低版本	容器	最小容器
grub2-pc-modules	2.06		
grub2-tools	2.06		
grub2-tools-minimal	2.06		
grubby	8.40		
gzip	1.12		
hostname	3.23		
hwdata	0.384		
inih	49		
initscripts	10.09		
iproute	6.10.0		
iputils	20210202		
irqbalance	1.9.0		
jansson	2.14		
jitterentropy	3.4.1		
jq	1.7.1		
json-c	0.14	0.14	0.14
kbd	2.4.0		
kbd-misc	2.4.0		
kernel	6.1.112		

软件包	AMI 最低版本	容器	最小容器
kernel-libbpf	6.1.112		
kernel-livepatch-repo-s3	2023.6.20241031		
keyutils-libs	1.6.3	1.6.3	1.6.3
kmod	29		
kmod-libs	29		
krb5-libs	1.21.3	1.21.3	1.21.3
less	608		
libacl	2.3.1	2.3.1	2.3.1
libarchive	3.7.4	3.7.4	3.7.4
libargon2	20171227		
libassuan	2.5.5	2.5.5	2.5.5
libattr	2.5.1	2.5.1	2.5.1
libblkid	2.37.4	2.37.4	2.37.4
libcap	2.48	2.48	2.48
libcap-ng	0.8.2	0.8.2	0.8.2
libcbor	0.7.0		
libcom_err	1.46.5	1.46.5	1.46.5
libcomps	0.1.20	0.1.20	
libcurl-minimal	8.5.0	8.5.0	8.5.0

软件包	AMI 最低版本	容器	最小容器
libdb	5.3.28		
libdnf	0.69.0	0.69.0	0.69.0
libeconf	0.4.0		
libedit	3.1		
libfdisk	2.37.4		
libffi	3.4.4	3.4.4	3.4.4
libfido2	1.10.0		
libgcc	11.4.1	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2	1.10.2
libgomp	11.4.1	11.4.1	
libgpg-error	1.42	1.42	1.42
libidn2	2.3.2	2.3.2	2.3.2
libkcapi	1.4.0		
libkcapi-hmaccalc	1.4.0		
libmnl	1.0.4		
libmodulemd	2.13.0	2.13.0	2.13.0
libmount	2.37.4	2.37.4	2.37.4
libnghttp2	1.59.0	1.59.0	1.59.0
libpeas			1.32.0
libpipeline	1.5.3		

软件包	AMI 最低版本	容器	最小容器
libpsl	0.21.1	0.21.1	0.21.1
libpwquality	1.4.4		
librepo	1.14.5	1.14.5	1.14.5
libreport-filesystem	2.15.2	2.15.2	2.15.2
libseccomp	2.5.3		
libselinux	3.4	3.4	3.4
libselinux-utils	3.4		
libsemanage	3.4		
libsepol	3.4	3.4	3.4
libsigsegv	2.13	2.13	2.13
libsmartcols	2.37.4	2.37.4	2.37.4
libsolv	0.7.22	0.7.22	0.7.22
libss	1.46.5		
libstdc++	11.4.1	11.4.1	11.4.1
libtasn1	4.19.0	4.19.0	4.19.0
libtextstyle	0.21		
libunistring	0.9.10	0.9.10	0.9.10
libuser	0.63		
libutempter	1.2.1		

软件包	AMI 最低版本	容器	最小容器
libuuid	2.37.4	2.37.4	2.37.4
libverto	0.3.2	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33	
libxml2	2.10.4	2.10.4	2.10.4
libyaml	0.2.5	0.2.5	0.2.5
libzstd	1.5.5	1.5.5	1.5.5
linux-firmware-whence	20210208 (noarch)		
logrotate	3.20.1		
lua-libs	5.4.4	5.4.4	5.4.4
lz4-libs	1.9.4	1.9.4	1.9.4
man-db	2.9.3		
microcode_ctl	2.1 (x86_64)		
microdnf			3.10.0
microdnf-dnf			3.10.0
mpfr	4.1.0	4.1.0	4.1.0
ncurses	6.2		
ncurses-base	6.2	6.2	6.2
ncurses-libs	6.2	6.2	6.2
nettle	3.8		
net-tools	2.0		

软件包	AMI 最低版本	容器	最小容器
npth	1.6	1.6	1.6
numactl-libs	2.0.14		
oniguruma	6.9.7.1		
openldap	2.4.57		
openssh	8.7p1		
openssh-clients	8.7p1		
openssh-server	8.7p1		
openssl	3.0.8		
openssl-libs	3.0.8	3.0.8	3.0.8
openssl-pkcs11	0.4.12		
os-prober	1.77		
p11-kit	0.24.1	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1	0.24.1
pam	1.5.1		
passwd	0.80		
pciutils	3.7.0		
pciutils-libs	3.7.0		
pcre2	10.40	10.40	10.40
pcre2-syntax	10.40	10.40	10.40
policycoreutils	3.4		

软件包	AMI 最低版本	容器	最小容器
popt	1.18	1.18	1.18
procps-ng	3.3.17		
psmisc	23.4		
publicsuffix-list-dafsa	20240212	20240212	20240212
python3	3.9.16	3.9.16	
python3-attrs	20.3.0		
python3-audit	3.0.6		
python3-awscrt	0.19.19		
python3-babel	2.9.1		
python3-cffi	1.14.5		
python3-chardet	4.0.0		
python3-colorama	0.4.4		
python3-configobj	5.0.6		
python3-cryptography	36.0.1		
python3-dateutil	2.8.1		
python3-dbus	1.2.18		
python3-distro	1.5.0		

软件包	AMI 最低版本	容器	最小容器
python3-dnf	4.14.0	4.14.0	
python3-dnf-plugins-core	4.3.0		
python3-d ocutils	0.16		
python3-gpg	1.15.1	1.15.1	
python3-hawkey	0.69.0	0.69.0	
python3-idna	2.10		
python3-jinja2	2.11.3		
python3-j mespath	0.10.0		
python3-j sonpatch	1.21		
python3-j sonpointer	2.0		
python3-j sonschema	3.2.0		
python3-l ibcomps	0.1.20	0.1.20	
python3-libdnf	0.69.0	0.69.0	
python3-libs	3.9.16	3.9.16	
python3-l ibselinux	3.4		

软件包	AMI 最低版本	容器	最小容器
python3-libsemanage	3.4		
python3-markupsafe	1.1.1		
python3-netifaces	0.10.6		
python3-oauthlib	3.0.2		
python3-pip-wheel	21.3.1	21.3.1	
python3-ply	3.11		
python3-policycoreutils	3.4		
python3-prettytable	0.7.2		
python3-prompt-toolkit	3.0.24		
python3-pycparser	2.20		
python3-pyrsistent	0.17.3		
python3-pyserial	3.4		
python3-pysocks	1.7.1		
python3-pytz	2022.7.1		

软件包	AMI 最低版本	容器	最小容器
python3-pyyaml	5.4.1		
python3-requests	2.25.1		
python3-rpm	4.16.1.3	4.16.1.3	
python3-ruamel-yaml	0.16.6		
python3-ruamel-yaml-clib	0.1.2		
python3-setools	4.4.1		
python3-setuptools	59.6.0		
python3-setuptools-wheel	59.6.0	59.6.0	
python3-six	1.15.0		
python3-systemd	235		
python3-urllib3	1.25.10		
python3-wcwidth	0.2.5		
readline	8.1	8.1	8.1
rng-tools	6.14		
rootfiles	8.1		
rpm	4.16.1.3	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3	

软件包	AMI 最低版本	容器	最小容器
rpm-libs	4.16.1.3	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3		
rpm-plugin-systemd-inhibit	4.16.1.3		
rpm-sign-libs	4.16.1.3	4.16.1.3	
sbsigntools	0.9.4		
sed	4.8	4.8	4.8
selinux-policy	38.1.45		
selinux-policy-targeted	38.1.45		
setup	2.13.7	2.13.7	2.13.7
shadow-utils	4.9		
sqlite-libs	3.40.0	3.40.0	3.40.0
sudo	1.9.15		
sysctl-defaults	1.0		
systemd	252.23		
systemd-libs	252.23		
systemd-networkd	252.23		
systemd-pam	252.23		

软件包	AMI 最低版本	容器	最小容器
systemd-resolved	252.23		
systemd-udev	252.23		
system-release	2023.6.20241031	2023.6.20241031	2023.6.20241031
tar	1.34		
tzdata	2024a	2024a	
update-motd	2.2		
userspace-rcu	0.12.1		
util-linux	2.37.4		
util-linux-core	2.37.4		
vim-data	9.0.2153		
vim-minimal	9.0.2153		
which	2.21		
xfsprogs	5.18.0		
xz	5.2.5		
xz-libs	5.2.5	5.2.5	5.2.5
yum	4.14.0	4.14.0	
zlib	1.2.11	1.2.11	1.2.11
zram-generator	1.1.2		
zram-generator-defaults	1.1.2		

软件包	AMI 最低版本	容器	最小容器
zstd	1.5.5		

AL2023 开启 AWS Elastic Beanstalk

AWS Elastic Beanstalk 是一项用于部署和扩展 Web 应用程序和服务的服务。您只需上传代码，Elastic Beanstalk 会自动处理部署 — 从容量配置、负载平衡、自动扩展到应用程序运行状况监控。有关更多信息，请参阅 AWS Elastic Beanstalk。

要使用 Elastic Beanstalk，您需要创建一个应用程序，将应用程序版本以应用程序源包的形式（如 Java .war 文件）上传到 Elastic Beanstalk，然后提供一些有关该应用程序的信息。Elastic Beanstalk 会自动启动环境，并创建和 AWS 配置运行代码所需的资源。有关更多信息，请参见AWS Elastic Beanstalk 开发人员指南。

Elastic Beanstalk Linux 平台 EC2 使用亚马逊实例，这些实例运行亚马逊 Linux。截至 2023 年 8 月 4 日，Elastic Beanstalk 提供以下基于 Amazon Linux 2023 的平台分支：Docker、Tomcat、Java SE、Node.js、PHP 和 Python。Elastic Beanstalk 正在努力向更多 Elastic Beanstalk 平台发布 AL2对 023 的支持。

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/WelcomeElastic Beanstalk 平台支持和当前在 023 之 AL2上构建的平台的完整列表可在 Elastic Beanstalk 开发者指南的 Elastic Beanstal k Linux 平台部分找到。

有关新的 Elastic Beanstalk 平台和现有平台版本的发布说明，可在 Elastic Beanstalk 发布说明中找到。

在中使用 AL2 023 AWS CloudShell

AWS CloudShell 是一个基于浏览器、经过预先验证的 shell，您可以直接从启动。 AWS Management Console您可以通过 AWS Management Console 几种不同的方式导航到。 CloudShell 有关更多信息，请参阅如何开始 AWS CloudShell？

AWS CloudShell目前基于亚马逊 Linux 2，将迁移到 AL2 023。从2023年12月4日开始，向 AL2 023 的迁移将 AWS 区域 开始全面推出。有关 CloudShell 迁移到 AL2 023 的更多信息，请参阅从亚马逊 Linux 2 AWS CloudShell 迁移到亚马逊 Linux 2023。

使用基于 AL2 023 的 Amazon ECS AMIs 托管容器化工作负载

> **Note**
>
> 有关如何在容器内使用 AL2 023 的更多信息，请参阅AL2023 装在容器里。

Amazon Elastic Container Service (Amazon ECS) 是一种完全托管式的容器编排服务，可以帮助您轻松部署、管理和扩展容器化应用程序。作为一项完全托管的服务，Amazon ECS 内置了 AWS 配置和操作最佳实践。它与亚马逊弹性容器注册表 (Amazon ECR) AWS 和 Docker 等第三方工具集成。这种集成使团队更容易专注于构建应用程序而不是环境。您可以在云端跨 AWS 区域运行和扩展容器工作负载，省去了管理控制面板的繁琐。

您可以使用基于 023 的 Amazon ECS 优化型 AMI 在 AL2 023 上托管容器化工作负载。 AL2有关更多信息，请参阅 Amazon ECS 优化的 AMI

与之相比，亚马逊 ECS 在 AL2 023 年的变化 AL2

与一样 AL2， AL2023 提供了作为亚马逊 ECS Linux 实例运行所需的基本软件包。 AL2 在`containerd``docker`、和`ecs-init`软件包中，可通过获得`amazon-linux-extras`，而 AL2 023 则将这些软件包包含在核心存储库中。

借助通过版本控制存储库进行确定性升级功能，默认情况下，每个 AL2 023 AMI 都锁定到特定的存储库版本。 AL2023 Amazon ECS 优化的 AMI 也是如此。在部署之前，可以仔细管理和测试您的环境的所有更新，并提供一种在出现问题时恢复到先前 AMI 内容的简便方法。有关此 AL2 023 功能的更多信息，请参阅023 年通过版本化存储库进行确定性升级 AL2。

AL2023 通过中支持的 cgroup v1 接口切换到 cgroup v2。 AL2有关更多信息，请参阅 统一控制组层次结构 (cgroup v2)。

> **Note**
>
> AL22023.2.20230920 之前的 023 个版本（第一个 AL2 023. 2 版本）在 cgroup 内的 for (OOM) 处理中`systemd`包含一个错误。 Out-of-Memorycgroup 中的所有进程总是被杀死，而不是 OOM 杀手一次选择一个进程，这是预期的行为。
> 与 AL2 行为相比，这是一种回归，自2023年2月20230920版本起已修复。 AL2

该项目中提供了用于构建 Amazon ECS 优化的 AMI 的amazon-ecs-ami GitHub 代码。发行说明描述了哪个 AL2 023 版本与哪个 Amazon ECS AMI 版本对应。

自定义基于 AL2 023 的亚马逊 ECS 优化版 AMI

> **Important**
>
> 我们建议您使用经过 Amazon ECS 优化的 AL2 023 AMI。有关更多信息，请参阅《亚马逊弹性容器服务开发者指南》中的亚马逊 ECS 优化版 AMI。

您可以使用与 Amazon ECS 创建自定义脚本相同的构建脚本 AMIs。有关更多信息，请参阅亚马逊 ECS 优化的 Linux AMI 构建脚本。

在 AL2 023 上使用亚马逊 Elastic File System

Amazon Elastic File System (Amazon EFS) 提供无服务器的完全弹性文件存储，这使您无需预置或管理存储容量和性能即可共享文件数据。Amazon EFS 可在不中断应用程序的情况下按需扩展到 PB 级，并可在您添加和移除文件时自动扩涨或收缩。Amazon EFS 具有简单的 Web 服务界面，可让您快速方便地创建和配置文件系统。该服务为您管理所有文件存储基础设施，这意味着您可以避免部署、修补和维护复杂文件系统配置的复杂性。

Amazon EFS 支持网络文件系统版本 4（NFSv4.1 和 NFSv4 .0）协议，因此您目前使用的应用程序和工具可与 Amazon EFS 无缝协作。包括亚马逊 EC2、亚马逊 ECS 和 AWS Lambda在内的多个计算实例可以同时访问一个 Amazon EFS 文件系统。因此，一个 EFS 文件系统可以为多个计算实例或服务器上运行的工作负载和应用程序提供通用数据源。

在 AL2 0 **`amazon-efs-utils`** 23 上安装

该`amazon-efs-utils`软件包可在待安装的 AL2 023 存储库中找到，用于访问 Amazon EFS 文件系统。

在 AL2 023 上安装**`amazon-efs-utils`**软件包

- `amazon-efs-utils`使用以下命令进行安装。

```
$ dnf -y install amazon-efs-utils
```

在 AL2 023 挂载亚马逊 EFS 文件系统

安装完成后`amazon-efs-utils`，您可以在 AL2 023 实例上挂载 Amazon EFS 文件系统。

在 AL2 023 挂载亚马逊 EFS 文件系统

- 要使用文件系统 ID 进行装载，请使用以下命令。

```
sudo mount -t efs file-system-id efs-mount-point/
```

您也可以装载该文件系统，以便使用 TLS 或 DNS 名称或装载目标 IP 而非文件系统 id 来加密传输中的数据。有关更多信息，请参阅使用 EFS 装载帮助程序在 Amazon Linux 实例上装载文件系统。

使用基于 023 的亚马逊 EMR AL2

Amazon EMR 是一种 Web 服务，它使您可以轻松地利用 Apache Hadoop 和 AWS所提供的服务高效地处理海量数据。

AL2基于 023 的亚马逊 EMR 发布

亚马逊 EMR 版本 7.0.0 是在 023 基础上构建的第一个版本。 AL2在此版本中， AL2023 成为亚马逊 EMR 的基础操作系统，将 023 的所有优势 AL2带到了亚马逊 EMR 中。有关更多信息，请参阅 Amazon EMR 7.0.0 发行说明。

AL2基于 023 的 EKS 上的 Amazon EMR

EKS 6.13 上的 Amazon EMR 是第一个引入 AL2 023 作为选项的版本。在这个版本中，你可以启动以 AL2 023 为操作系统的 Spark，同时启动 Java 17 运行时。有关更多信息，请参阅 E KS 6.13 上的 Amazon EMR 发行说明，以及 EKS 上的所有亚马逊 EMR 发行说明。

在中使用 AL2 023 AWS Lambda

使用 AWS Lambda，您无需预置或管理服务器即可运行代码。您只需为所消耗的计算时间付费，当您的代码未运行时，不收取任何费用。您几乎可以为任何类型的应用程序或后端服务运行代码，所有这些都无需管理。只需上传您的代码，Lambda 就会处理以高可用性运行和扩展您的代码所需的一切。

AL2023 **provided.al2023** 托管运行时和容器镜像

provided.al2023基本运行时基于 AL2023 最小容器镜像，并提供基于 AL2 023 的 Lambda 托管运行时和容器基础镜像。由于provided.al2023运行时间基于 AL2 023 最小容器映像，因此小于 40 MB 的运行时间要比大约 109 MB 的provided.al2运行时间小得多。

有关更多信息，请参阅 Lambda 运行时和使用 Lambd a 容器镜像。

AL2基于 023 的 Lambda 运行时

托管语言运行时的未来版本，例如 Node.js 20,Python 3.12，Java 21，而且 .NET 8，基于 AL2 023，将provided.al2023用作基于 AL2 023 的运行时公告中所述的基本映像。

AL2基于 023 的 Lambda 函数

- AL2023 写入的 Lambda 函数 Go
- AL2023 写入的 Lambda 函数 Rust

有关更多信息，请参阅《AWS Lambda 开发人员指南》中的 Lambda 运行时系统。

教程

以下教程向您展示如何使用运行亚马逊 Linux 2023 (AL2023) 的亚马逊 EC2 实例执行常见任务。有关视频教程，请参阅AWS 教学视频和实验。

有关 AL2 说明，请参阅亚马逊 EC2 用户指南中的运行 Linux 的亚马逊 EC2 实例教程。

教程

- 教程：在 AL2 023 上安装 LAMP 服务器
- 教程：在 023 上配置 SSL/TLS AL2
- 教程：在 AL2 023 上主持 WordPress 博客
- 教程：023 年 Redis 6 到 Valkey 过渡 AL2
- 教程：在 023 上 AL2安装 GNOME 桌面环境
- 教程：在 023 上配置 TigerVNC 服务器 AL2

教程：在 AL2 023 上安装 LAMP 服务器

以下过程可帮助你在 023 实例（有时称为 LAMP 网络服务器或 LAMP 堆栈）上安装支持 PHP 和 M ar iaDB（社区开发 AL2的 MySQL 分支）的 Apache 网络服务器。您可以使用此服务器来托管静态网站或部署能对数据库中的信息执行读写操作的动态 PHP 应用程序。

> **Important**
>
> 这些程序旨在与 AL2 023 一起使用。如果您尝试在其他发行版（例如 Ubuntu 或红帽企业 Linux）上设置 LAMP Web 服务器，则本教程不适合。对于 Ubuntu，请参阅以下 Ubuntu 社区文档：SQLPHP。ApacheMy有关其他发布版本，请参阅特定于该版本的文档。

任务

- 步骤 1：准备 LAMP 服务器
- 步骤 2：测试 LAMP 服务器
- 步骤 3：确保数据库服务器的安全
- 步骤 4：（可选）安装 phpMyAdmin
- 故障排除

- 相关 主题

步骤 1：准备 LAMP 服务器

先决条件

- 本教程假设您已经使用 AL2 023 启动了一个新实例，其公有 DNS 名称可以从互联网访问。有关更多信息，请参阅 AL2亚马逊上的 023 EC2。您还必须配置安全组，以便允许 SSH (端口 22)、HTTP (端口 80) 和 HTTPS (端口 443) 连接。有关这些先决条件的更多信息，请参阅 Amazon EC2 用户指南中的授权 Linux 实例的入站流量。
- 以下过程安装 AL2 023 上可用的最新 PHP 版本，当前为 8.1。如果您计划使用本教程中所述的 PHP 应用程序之外的 PHP 应用程序，则应检查其与 8.1 的兼容性。

准备 LAMP 服务器

1. 连接到您的 实例。有关更多信息，请参阅 正在连接到 AL2 023 个实例。
2. 为确保您的所有软件包都处于最新状态，请对您的实例执行快速软件更新。此过程可能需要几分钟的时间，但必须确保您拥有最新的安全更新和缺陷修复。

   -y 选项安装更新时不提示确认。如果您希望在安装前检查更新，则可以忽略该选项。

   ```
   [ec2-user ~]$ sudo dnf upgrade -y
   ```

3. 安装最新版本的 Apache Web 服务器和 AL2 023 的 PHP 软件包。

   ```
   [ec2-user ~]$ sudo dnf install -y httpd wget php-fpm php-mysqli php-json php php-
   devel
   ```

4. 安装 MariaDB 软件包。使用 dnf install 命令可同时安装多个软件包和所有相关依赖项。

   ```
   [ec2-user ~]$ sudo dnf install mariadb105-server
   ```

   您可以使用以下命令查看这些程序包的当前版本：

   ```
   [ec2-user ~]$ sudo dnf info package_name
   ```

   示例：

```
[root@ip-172-31-25-170 ec2-user]# dnf info mariadb105
Last metadata expiration check: 0:00:16 ago on Tue Feb 14 21:35:13 2023.
Installed Packages
Name         : mariadb105
Epoch        : 3
Version      : 10.5.16
Release      : 1.amzn2023.0.6
Architecture : x86_64
Size         : 18 M
Source       : mariadb105-10.5.16-1.amzn2023.0.6.src.rpm
Repository   : @System
From repo    : amazonlinux
Summary      : A very fast and robust SQL database server
URL          : http://mariadb.org
License      : GPLv2 and LGPLv2
Description  : MariaDB is a community developed fork from MySQL - a multi-user,
 multi-threaded
             : SQL database server. It is a client/server implementation consisting
 of
             : a server daemon (mariadbd) and many different client programs and
 libraries.
             : The base package contains the standard MariaDB/MySQL client programs
 and
             : utilities.
```

5. 启动 Apache Web 服务器。

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

6. 使用 systemctl 命令配置 Apache Web 服务器，使其在每次系统启动时启动。

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

您可以通过运行以下命令验证 httpd 是否已启用：

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

7. 如果您尚未这样做，请添加安全规则以允许与您的实例的入站 HTTP (端口 80) 连接。默认情况下，启动时会为您的实例创建启动向导***N***安全组。如果您没有添加其他安全组规则，此组仅包含允许 SSH 连接的一条规则。

a. 打开 Amazon EC2 控制台，网址为https://console.aws.amazon.com/ec2/。

b. 在左侧的导航中，选择 Instances（实例），然后选择您的实例。

c. 在安全选项卡上，查看入站规则。您应看到以下规则：

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

> **Warning**
>
> 使用`0.0.0.0/0`允许所有 IPv4 地址通过 SSH 访问您的实例。这在测试环境中可以接受一小段时间，但是在生产环境中并不安全。在生产环境中，您仅授权特定 IP 地址或地址范围访问您的实例。

d. 如果没有允许 HTTP（端口 80）连接的入站规则，则必须立即添加规则。选择安全组的链接。使用授权您的 Linux 实例的入站流量中的步骤，添加具有以下值的新入站安全规则：

- 类型：HTTP
- 协议：TCP
- Port Range：80
- Source：Custom

8. 测试您的 Web 服务器。在 Web 浏览器中，键入您的实例的公有 DNS 地址 (或公有 IP 地址)。如果 `/var/www/html` 中没有内容，您应该会看到 Apache 测试页面，该页面将显示消息 It works!（它工作正常！）。

   您可以使用 Amazon EC2 控制台获取实例的公有 DNS（查看公有 IPv4 DNS 列；如果此列处于隐藏状态，请选择首选项（齿轮形图标）并打开公 IPv4 有 DNS）。

   验证实例的安全组是否包含允许端口 80 上的 HTTP 流量的规则。有关更多信息，请参阅向安全组添加规则。

> **Important**
>
> 如果您使用的不是 Amazon Linux，则还可能需要在实例上配置防火墙才能允许这些连接。有关如何配置防火墙的更多信息，请参阅适用于特定分配的文档。

Apache httpd 提供的文件保存在名为 Apache 文档根目录的目录中。Amazon Linux Apache 文档根目录为 /var/www/html，默认情况下归根用户所有。

要允许 ec2-user 账户操作此目录中的文件，必须修改其所有权和权限。有多种方式可以完成此任务。在本教程中，可将 ec2-user 添加到 apache 组，将 /var/www 目录的所有权授予 apache 组，并为该组指定写入权限。

设置文件权限

1. 将您的用户 (这里指 ec2-user) 添加到 apache。

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

2. 先退出再重新登录以选取新组，然后验证您的成员资格。

   a. 退出（使用 exit 命令或关闭终端窗口）：

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

   b. 要验证您是否为 apache 组的成员，请重新连接到实例，然后运行以下命令：

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

3. 将 /var/www 及其内容的组所有权更改到 apache 组。

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

4. 要添加组写入权限以及设置未来子目录上的组 ID，请更改 /var/www 及其子目录的目录权限。

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

5. 要添加组写入权限，请递归地更改 /var/www 及其子目录的文件权限：

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

这样，ec2-user (和 apache 组的任何未来成员) 可以添加、删除和编辑 Apache 文档根目录中的文件，允许您添加内容，如静态网站或 PHP 应用程序。

保护您的 Web 服务器 (可选)

运行 HTTP 协议的 Web 服务器不为其发送或接收的数据提供传输安全。当您使用 Web 浏览器连接到 HTTP 服务器时，网络路径上任何地方的窃听者都可以看到您访问的内容、收到的网页内容以及您提交的任何 HTML 表单的内容（包括密码）。 URLs 保护您的 Web 服务器的最佳实践是安装 HTTPS (HTTP Secure) 支持，它将使用 SSL/TLS 加密保护您的数据。

有关在服务器上启用 HTTPS 的信息，请参阅 教程：在 023 上配置 SSL/TLS AL2。

步骤 2：测试 LAMP 服务器

如果服务器已安装并运行，且文件权限设置正确，则 `ec2-user` 账户应该能够在 `/var/www/html` 目录 (可从 Internet 访问) 中创建 PHP 文件。

测试您的 LAMP 服务器

1. 在 Apache 文档根目录中创建一个 PHP 文件。

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

   尝试运行该命令时，如果出现“Permission denied (权限被拒绝)”错误，请尝试先注销，再重新登录，以获取您在 设置文件权限 中配置的适当组权限。

2. 在 Web 浏览器中，键入您刚刚创建的文件的 URL。此 URL 是实例的公用 DNS 地址，后接正斜杠和文件名。例如：

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

   您应该会看到 PHP 信息页面：

PHP Version 8.1.7

System	Linux ip-172-31-16-77.ec2.internal 5.15.57-28.127.amzn2022.aarch64 #1 SMP Thu Aug 4 17:06:57 UTC 2022 aarch64
Build Date	Jun 7 2022 18:21:38
Build System	Linux
Build Provider	Amazon Linux
Compiler	gcc (GCC) 11.3.1 20220421 (Red Hat 11.3.1-2)
Architecture	aarch64
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/10-opcache.ini, /etc/php.d/20-bz2.ini, /etc/php.d/20-calendar.ini, /etc/php.d/20-ctype.ini, /etc/php.d/20-curl.ini, /etc/php.d/20-dom.ini, /etc/php.d/20-exif.ini, /etc/php.d/20-fileinfo.ini, /etc/php.d/20-ftp.ini, /etc/php.d/20-gd.ini, /etc/php.d/20-gettext.ini, /etc/php.d/20-iconv.ini, /etc/php.d/20-mbstring.ini, /etc/php.d/20-mysqlnd.ini, /etc/php.d/20-pdo.ini, /etc/php.d/20-phar.ini, /etc/php.d/20-simplexml.ini, /etc/php.d/20-sockets.ini, /etc/php.d/20-sqlite3.ini, /etc/php.d/20-tokenizer.ini, /etc/php.d/20-xml.ini, /etc/php.d/20-xmlwriter.ini, /etc/php.d/20-xsl.ini, /etc/php.d/30-mysqli.ini, /etc/php.d/30-pdo_mysql.ini, /etc/php.d/30-pdo_sqlite.ini, /etc/php.d/30-xmlreader.ini
PHP API	20210902
PHP Extension	20210902
Zend Extension	420210902
Zend Extension Build	API420210902,NTS
PHP Extension Build	API20210902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, compress.bzip2, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, bzip2.*, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.1.7, Copyright (c) Zend Technologies
with Zend OPcache v8.1.7, Copyright (c), by Zend Technologies

如果您未看到此页面，请验证上一步中是否已正确创建 `/var/www/html/phpinfo.php` 文件。您还可以使用以下命令验证已经安装了所有必需的程序包。

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

如果输出中未列出任何必需的程序包，请使用 sudo yum install ***package*** 命令安装它们。

3. 删除 `phpinfo.php` 文件。尽管此信息可能很有用，但出于安全考虑，不应将其传播到 Internet。

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

现在，您应该有了一个功能完善的 LAMP Web 服务器。如果您将内容添加到 Apache 文档根目录 (位于 `/var/www/html`)，您应该能够在您的实例的公有 DNS 地址中看到该内容。

步骤 3：确保数据库服务器的安全

MariaDB 服务器的默认安装提供有多种功能，这些功能对于测试和开发都很有帮助，但对于产品服务器，应禁用或删除这些功能。mysql_secure_installation 命令可引导您设置根密码并删除安装中的不安全功能。即使您不打算使用 MariaDB 服务器，我们也建议执行此步骤。

保护 MariaDB 服务器

1. 启动 MariaDB 服务器。

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

2. 运行 mysql_secure_installation。

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

   a. 在提示时，键入根账户的密码。

      i. 键入当前根密码。默认情况下，根账户没有设置密码。按 Enter。

      ii. 键入 **Y** 设置密码，然后键入两次安全密码。有关创建安全密码的更多信息，请参阅https://identitysafe.norton.com/password-generator/。确保将此密码存储在安全位置。

      设置 MariaDB 根密码仅是保护数据库的最基本措施。在您构建或安装数据库驱动的应用程序时，您通常可以为该应用程序创建数据库服务用户，并避免使用根账户执行除数据库管理以外的操作。

   b. 键入 **Y** 删除匿名用户账户。

   c. 键入 **Y** 禁用远程根登录。

   d. 键入 **Y** 删除测试数据库。

   e. 键入 **Y** 重新加载权限表并保存您的更改。

3. (可选) 如果您不打算立即使用 MariaDB 服务器，请停止它。您可以在需要时再次重新启动。

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

4. (可选) 如果您希望每次启动时 MariaDB 服务器都启动，请键入以下命令。

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

步骤 4：（可选）安装 phpMyAdmin

phpMyAdmin是一款基于 Web 的数据库管理工具，可用于查看和编辑 EC2实例上的 MySQL 数据库。按照下述步骤操作，在您的 Amazon Linux 实例上安装和配置 phpMyAdmin。

> **Important**
>
> 除非您在 Apache 中启用了 SSL/TLS，否则我们不建议您使用 phpMyAdmin 访问 LAMP 服务器；如果您使用 phpMyAdmin，您的数据库管理员密码和其他数据将无法安全地通过 Internet 传输。有关开发人员提出的安全建议，请参阅保护您的 phpMyAdmin 安装。有关保护 EC2 实例上的 Web 服务器的一般信息，请参阅教程：在 023 上配置 SSL/TLS AL2。

要安装 phpMyAdmin

1. 安装所需的依赖项。

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

2. 重启 Apache。

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

3. 重启 php-fpm。

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

4. 导航到位于 /var/www/html 的 Apache 文档根。

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

5. 从 https://www.phpmyadmin.net/downloads 选择最新 phpMyAdmin 版本的源软件包。要将文件直接下载到您的实例，请复制链接并将其粘贴到 wget 命令，如本示例中所述：

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

6. 使用以下命令创建 phpMyAdmin 文件夹并将程序包提取到其中。

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

7. 删除*phpMyAdmin-latest-all-languages.tar.gz*压缩包。

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

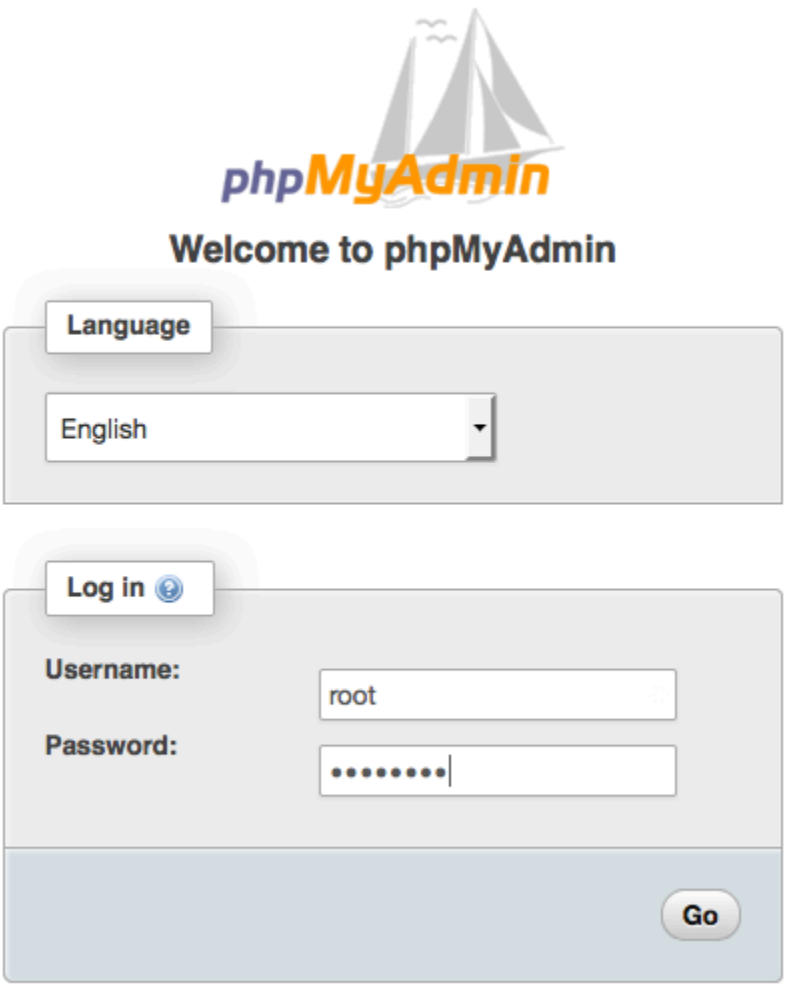
8. (可选) 如果 MySQL 服务器未运行，请立即启动它。

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

9. 在 Web 浏览器中，键入 phpMyAdmin 安装的 URL。此 URL 是实例的公有 DNS 地址 (或公有 IP 地址)，后接正斜杠和您安装目录的名称。例如：

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

   你应该会看到 phpMyAdmin 登录页面：



10. 使用您之前创建的`root`用户名和 MySQL 根密码登录您的 phpMyAdmin 安装。

    您的安装仍需进行配置，然后才能投入使用。我们建议您首先手动创建配置文件，如下所示：

    a. 要从最小的配置文件开始，请使用您常用的文本编辑器创建一个新文件，然后将`config.sample.inc.php` 的内容复制到该文件中。
    b. 将文件另存为`config.inc.php`包含的 phpMyAdmin 目录中`index.php`。
    c. 有关任何其他设置，请参阅安装说明的 “使用 phpMyAdmin 安装脚本” 部分中的文件创建后说明。

    有关使用的信息 phpMyAdmin，请参阅《phpMyAdmin 用户指南》。

故障排除

本部分提供了解决在设置新 LAMP 服务器时可能遇到的常见问题的建议。

我无法使用 Web 浏览器连接到我的服务器

执行以下检查以查看您的 Apache Web 服务器是否正在运行且可以访问。

- Web 服务器正在运行吗？

  您可以通过运行以下命令验证 httpd 是否已启用：

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

  如果 httpd 进程未运行，请重复准备 LAMP 服务器中描述的步骤。

- 防火墙是否配置正确？

  验证实例的安全组是否包含允许端口 80 上的 HTTP 流量的规则。有关更多信息，请参阅向安全组添加规则。

我无法使用 HTTPS 连接到我的服务器

执行以下检查以查看 Apache Web 服务器是否配置为支持 HTTPS。

- Web 服务器配置是否正确？

  安装 Apache 后，服务器将针对 HTTP 流量进行配置。要支持 HTTPS，请在服务器上启用 TLS 并安装 SSL 证书。有关信息，请参阅教程：在 023 上配置 SSL/TLS AL2。

- 防火墙是否配置正确？

  验证实例的安全组是否包含允许端口 443 上的 HTTPS 流量的规则。有关更多信息，请参阅授权 Linux 实例的入站流量。

相关 主题

有关将文件传输到您的实例或在 Web 服务器上安装 WordPress 博客的更多信息，请参阅以下文档：

- 使用亚马逊 EC2 用户指南中的 WinSCP 将文件传输到您的 Linux 实例。

- 使用《亚马逊 EC2 用户指南》中的 SCP 客户端将文件传输到 Linux 实例。
- 教程：在 AL2 023 上主持 WordPress 博客

有关本教程中使用的命令和软件的更多信息，请参阅以下网页：

- Apache Web 服务器：http://httpd.apache.org/
- MariaDB 数据库服务器：https://mariadb.org/
- PHP 编程语言：http://php.net/

有关注册 Web 服务器域名或将现有域名转移到此主机的更多信息，请参阅 Amazon Route 53 开发人员指南中的创建域和子域并将其迁移到 Amazon Route 53。

教程：在 023 上配置 SSL/TLS AL2

在装有 AL2 023 和 Apache Web 服务器的 EC2 实例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上使用安全套接字。本教程假定您未使用负载均衡器。如果您正在使用 Elastic Load Balancing，则可以选择使用来自 AWS Certificate Manager 的证书在负载均衡器上配置 SSL 卸载。

由于历史原因，Web 加密通常简称为 SSL。虽然 Web 浏览器仍然支持 SSL，但其后续协议 TLS 不太容易受到攻击。 AL2默认情况下，023 禁用服务器端对所有 SSL 版本的支持。安全标准机构认为 TLS 1.0 不安全。TLS 1.0 和 TLS 1.1 已于 2021 年 3 月正式弃用。本教程仅包含有关启用 TLS 1.2 的指导。TLS 1.3 已于 2018 年完成， AL2 只要支持和启用底层 TLS 库（本教程中的 OpenSSL），即可使用。客户端必须在 2023 年 6 月 28 日之前支持 TLS 1.2 或更高版本。有关更新的加密标准的更多信息，请参阅 RFC 7568 和 RFC 8446。

在本教程中，将现代 Web 加密简称为 TLS。

> **Important**
>
> 这些程序旨在与 AL2 023 一起使用。如果您正在尝试设置运行其他发行版的 EC2 实例，或者设置运行旧版本的 Amazon Linux 的实例，则本教程中的某些步骤可能不起作用。对于 Ubuntu，请参阅以下 Ubuntu 社区文档：Open SSL on Ubuntu。有关 Red Hat Enterprise Linux 的信息，请参阅以下：设置 Apache HTTP Web 服务器。有关其他发布版本，请参阅特定于该版本的文档。

> **Note**
>
> 或者，您可以将 AWS Certificate Manager (ACM) 用于 AWS Nitro 安全区，这是一种安全区应用程序，允许您在带有 Nitro Enclaves 的 Amazon 实例上运行的 Web 应用程序和服务器上使用公有和私有 SSL/TLS 证书。 EC2 AWS Nitro Enclaves 是 Amazon 的一项 EC2 功能，它允许创建隔离的计算环境，以保护和安全处理高度敏感的数据，例如 SSL/TLS 证书和私钥。
> ACM for Nitro Enclaves 可与在 Amazon Linux 实例上运行的 ng EC2 inx 配合使用，以创建私钥、分发证书和私钥以及管理证书续订。
> 要使用适用于 Nitro Enclaves 的 ACM，必须使用启用了 Enclave 的 Linux 实例。
> 有关更多信息，请参阅什么是 AWS 硝基飞地？ 以及AWS Certificate Manager 《Nitro Enclaves 用户指南》中的 AWS Nitro Enclaves。

内容

- 前提条件
- 步骤 1：在服务器上启用 TLS
- 步骤 2：获取 CA 签名的证书
- 步骤 3：测试和强化安全配置
- 故障排除

前提条件

在开始本教程之前，请完成以下步骤：

- 启动 EBS 支持的 AL2 023 实例。有关更多信息，请参阅 AL2亚马逊上的 023 EC2。
- 配置安全组以允许您的实例接受以下 TCP 端口上的连接：
  - SSH (端口 22)
  - HTTP (端口 80)
  - HTTPS (端口 443)

  有关更多信息，请参阅 Amazon EC2 用户指南中的授权 Linux 实例的入站流量。
- 安装 Apache Web 服务器。有关 step-by-step说明，请参阅教程：在 AL2 023 上安装 LAMP 服务器。仅需要 httpd 包及其依赖项，因此可以忽略涉及 PHP 和 MariaDB 的说明。

- 要识别和验证网站，TLS 公有密钥基础设施 (PKI) 依赖于域名系统 (DNS)。要使用您的 EC2 实例托管公共网站，您需要为 Web 服务器注册域名或将现有域名转移到您的 Amazon EC2 主机。可通过很多第三方域注册和 DNS 托管服务来执行此操作，也可以使用 Amazon Route 53 执行此操作。

步骤 1：在服务器上启用 TLS

此过程将引导您完成使用自签名数字证书在 AL2 023 上设置 TLS 的过程。

> **Note**
>
> 自签名证书对于测试是可接受的，但对于生产不是。如果您将自签名证书公开到 Internet，您网站的访客将会看到安全警告。

在服务器上启用 TLS

1. 连接到您的实例并确认 Apache 正在运行。有关更多信息，请参阅 正在连接到 AL2 023 个实例。

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

   如果返回的值不是“启用”，则启动 Apache 并将它设置为每次随系统一起启动。

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

2. 为确保您的所有软件包都处于最新状态，请对您的实例执行快速软件更新。此过程可能需要几分钟的时间，但必须确保您拥有最新的安全更新和缺陷修复。

   > **Note**
   >
   > -y 选项安装更新时不提示确认。如果您希望在安装前检查更新，则可以忽略该选项。

   ```
   [ec2-user ~]$ sudo dnf install openssl mod_ssl
   ```

3. 输入以下命令后，系统将向您显示提示框，您可以在其中输入有关站点的信息。

```
[ec2-user ~]$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /
etc/pki/tls/private/apache-selfsigned.key -out /etc/pki/tls/certs/apache-
selfsigned.crt
```

这会在 `/etc/pki/tls/certs/` 目录中生成一个新文件 `apache-selfsigned.crt`。指定的文件名与 `/etc/httpd/conf.d/ssl.conf` 中的 SSLCertificateFile 指令指定的默认值匹配。

您的实例现在具有以下文件，可使用这些文件配置安全服务器并创建证书以进行测试：

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

  mod_ssl 的配置文件。它包含一些指令 以指示 Apache 在何处查找以下信息：加密密钥和证书、要允许的 TLS 协议版本以及要接受的加密密码。这将是您的本地证书文件：

- `/etc/pki/tls/certs/apache-selfsigned.crt`

该文件包含自签名证书以及证书的私有密钥。Apache 要求证书和密钥采用 PEM 格式，其中包含 Base64 编码的 ASCII 字符，并用“BEGIN”和“END”行框起来，如以下简短示例所示。


```
-----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-----
```

文件名和扩展名只是为了提供便利，对功能没有影响。例如，只要 `ssl.conf` 文件中的相关指令使用相同的名称，您就可以将证书命名为 `cert.crt`、`cert.pem` 或任何其他文件名。

> **Note**
> 在使用您自己的自定义文件替换默认 TLS 文件时，请确保它们采用 PEM 格式。

4. 重启 Apache。

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

> **Note**
> 如前所述，确保您的 EC2 实例上可以访问 TCP 端口 443。

5. 现在，您的 Apache Web 服务器应通过端口 443 支持 HTTPS (安全 HTTP)。通过在带有前缀的浏览器 URL 栏中输入 EC2实例的 IP 地址或完全限定域名来对其进行测试**`https://`**。

   由于您正在使用自签名的不可信主机证书连接到站点，因此您的浏览器可能会显示一系列安全警告。忽视这些警告并继续连接站点。

   如果默认 Apache 测试页面打开，这意味着您已成功在服务器上配置 TLS。在浏览器和服务器之间传输的所有数据现在都已加密。

> **Note**
> 为了防止站点访问者遇到警告屏幕，您必须获取一个可信 CA 签名证书，该证书不仅进行加密，而且还公开验证您是否为站点拥有者。

步骤 2：获取 CA 签名的证书

您可以使用以下过程获取 CA 签名证书：

- 从私有密钥生成证书签名请求 (CSR)
- 将 CSR 提交给证书颁发机构 (CA)

- 获取签名的主机证书
- 配置 Apache 以使用证书

从加密角度看，自签名 TLS X.509 主机证书与 CA 签名证书完全相同。二者之间的区别在于社交层面，而非数学层面。CA 承诺，在向申请者颁发证书之前，至少验证域的所有权。每个 Web 浏览器都包含浏览器供应商 CAs 信任的列表。X.509 证书主要包含一个与您的私有服务器密钥对应的公有密钥和一个以加密方式与该公有密钥关联的 CA 的签名。当浏览器通过 HTTPS 连接到 Web 服务器时，服务器会提供一份证书，供浏览器根据其可信列表进行核对 CAs。如果签署人位于列表上，或可通过由其他可信签署人组成的一系列信任访问，则浏览器将与服务器协商一个快速加密数据通道并加载页面。

由于验证请求需要投入人力，证书通常会产生费用，因此应货比三家。一些公司免费 CAs 提供基本级别的证书。其中最引人注目的是 CAs L et's Encrypt 项目，该项目还支持证书创建和续订过程的自动化。有关使用 Let's Encrypt 证书的更多信息，请参阅获取 Certbot。

如果您打算提供商业级服务，AWS Certificate Manager 是一个不错的选择。

主机证书的基础是密钥。从 2019 年开始，政府和行业群体建议 RSA 密钥使用 2048 位的最小密钥（模数）大小，旨在将文档一直保护到 2030 年。OpenSSL AL2 在 023 中生成的默认模数大小为 2048 位，适用于在 CA 签名的证书中使用。在以下过程中，为需要自定义密钥的人员提供了一个可选步骤，例如，具有较大模数或使用不同加密算法的步骤。

> **Important**
>
> 除非您拥有注册并托管的 DNS 域，否则，有关获取 CA 签名主机证书的这些说明不适用。

获取 CA 签名的证书

1. Connect 连接到您的实例并导航to /etc/pki/tls/private/。这是存储 TLS 的服务器私有密钥的目录。如果您希望使用现有的主机密钥生成 CSR，请跳到步骤 3。有关连接到您的实例的更多信息，请参阅 正在连接到 AL2 023 个实例
2. (可选) 生成新的私有密钥。下面是一些密钥配置示例。任何生成的密钥都可用于您的 Web 服务器，但它们实施安全的程度和类型有所不同。

   - 示例 1：创建默认 RSA 主机密钥。生成的文件 **`custom.key`** 是一个 2048 位 RSA 私有密钥。

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

- 示例 2：创建具有更大模数的更严格的 RSA 密钥。生成的文件 **custom.key** 是一个 4096 位 RSA 私有密钥。

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

- 示例 3：创建具有密码保护的 4096 位加密的 RSA 密钥。生成的文件 **custom.key** 是一个已使用 AES-128 密码加密的 4096 位 RSA 私有密钥。

> **Important**
>
> 对密钥进行加密可增强安全性，但由于加密的密钥需要密码，因此依赖于加密密钥的服务无法自动启动。每当您使用此密钥时，都必须通过 SSH 连接提供密码（在上一示例中为“abcde12345”）。

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

- 示例 4：使用非 RSA 密码创建密钥。RSA 加密可能相对较慢，因为其公有密钥的大小基于两个大素数的乘积。不过，可以为 TLS 创建使用非 RSA 密码的密钥。在交付同等级别的安全性时，基于椭圆曲线的数学运算的密钥更小，计算起来更快。

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

  结果为一个使用 prime256v1（OpenSSL 支持的“命名曲线”）的 256 位椭圆曲线私有密钥。根据 NIST，其加密强度略高于 2048 位 RSA 密钥。

> **Note**
>
> 并非所有密钥都 CAs 提供与 RSA elliptic-curve-based 密钥相同的支持级别。

请确保新的私有密钥具有高度限制的所有权和权限 (所有者=根、组=根、仅面向所有者的读取/写入权限)。命令将如以下示例所示。

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

上述命令生成以下结果。

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

在创建并配置满意的密钥后，可以创建 CSR。

3. 使用您首选的密钥创建 CSR。下面的示例使用了 **custom.key**。

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

OpenSSL 将打开一个对话框，并提示您输入下表中显示的信息。对于基本的经域验证的主机证书来说，除 Common Name 以外的所有字段都是可选字段。

名称	描述	示例
国家/地区名称	代表国家/地区的两个字母 ISO 缩写。	US (=美国)
州或省名称	组织所在州或省的名称。此名称不可使用缩写。	Washington
所在地名称	您的组织所在的位置，例如城市。	Seattle
组织名称	组织的法定全称。请勿缩写组织名称。	Example Corporation
组织部门名称	额外的组织信息 (如果有)。	示例部门
公用名	此值必须与您希望用户输入浏览器中的 Web 地址完全匹配。通常，这表示以主机名称为前缀的域名或采用 **www.example.com** 格式的别名。在使用自签名证书且没有 DNS 解析进行测试时，公用名可能仅包含主机名。 CAs 还提供更昂贵的证书，这些证书接受通配符名称，例如. ***.example.com**	www.example.com
电子邮件地址	服务器管理员的电子邮件地址。	someone@example.com

最后，OpenSSL 将提示您输入可选的质询密码。此密码仅适用于 CSR 和您与 CA 之间的事务，因此请遵循 CA 提供的有关此密码以及其他可选字段、可选公司名的建议。CSR 质询密码不会影响服务器操作。

生成的文件 **csr.pem** 包含您的公有密钥、您的公有密钥的数字签名以及您输入的元数据。

4. 将 CSR 提交给 CA。这通常包括在文本编辑器中打开 CSR 文件并将内容复制到 Web 表格中。此时，可能会要求您提供一个或多个主题备用名称 (SANs) 以放在证书上。如果 **www.example.com** 是公用名，则 **example.com** 将是一个很好的 SAN，反之亦然。您网站的访客如果输入这两个名称的任何一个，便可看到一个没有错误的连接。如果您的 CA Web 表单允许，请在列表中包含常用名 SANs。有些是自动 CAs 包含的。

   在您的请求获得批准后，您将收到一个由 CA 签署的新主机证书。此外，系统可能会指示您下载中间证书文件，该文件包含完成 CA 的信任链所需的其他证书。

   > **Note**
   >
   > 您的 CA 可能会针对各种用途发送多种格式的文件。在本教程中，您应只使用 PEM 格式的证书文件，此格式通常会（但不总是）标有 `.pem` 或 `.crt` 文件扩展名。如果您不确定要使用哪个文件，请使用文本编辑器打开这些文件，并查找一个包含一个或多个以下面的行开始的块的文件。
   >
   > ```
   > - - - - -BEGIN CERTIFICATE - - - - -
   > ```
   >
   > 该文件还应以下面的行结束。
   >
   > ```
   > - - - -END CERTIFICATE - - - - -
   > ```
   >
   > 您还可以在命令行上测试文件，如下所示。
   >
   > ```
   > [ec2-user certs]$ openssl x509 -in certificate.crt -text
   > ```
   >
   > 验证这些行是否显示在文件中。请勿使用结尾为 `.p7b`、`.p7c` 或类似文件扩展名的文件。

5. 将新的 CA 签名证书和任何中间证书放在 `/etc/pki/tls/certs` 目录中。

> **Note**
>
> 有几种方法可以将新证书上传到您的 EC2实例，但最直接和最有用的方法是在本地计算机和实例上打开文本编辑器（例如 vi、nano 或记事本），然后在它们之间复制并粘贴文件内容。在 EC2实例上执行这些操作时，你需要 root [sudo] 权限。这样，一旦有任何权限或路径问题，您可以立即看到。但请小心操作，不要在复制内容时添加任何多余的行或以任何方式更改内容。

在`/etc/pki/tls/certs`目录内部，检查文件所有权、组和权限设置是否与严格限制的 AL2 023 默认值相匹配（owner=root、group=root、仅限所有者读/写）。以下示例显示了要使用的命令。

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

这些命令应生成以下结果。

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

中间证书文件的权限并不严格 (所有者=根、组=根、所有者可以写入、组可以读取、任何人均可读取)。以下示例显示了要使用的命令。

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

这些命令应生成以下结果。

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

6. 将用于创建 CSR 的私有密钥放在 `/etc/pki/tls/private/` 目录中。

> **Note**
>
> 有多种方法可以将自定义密钥上传到您的 EC2 实例，但最直接和最有用的方法是在本地计算机和实例上打开文本编辑器（例如 vi、nano 或记事本），然后在它们之间复制并粘贴文

件内容。在 EC2实例上执行这些操作时，你需要 root [sudo] 权限。这样，一旦有任何权限或路径问题，您可以立即看到。但请小心操作，不要在复制内容时添加任何多余的行或以任何方式更改内容。

在`/etc/pki/tls/private`目录内部，使用以下命令验证文件所有权、组和权限设置是否符合严格限制的 AL2 023 默认值（owner=root、group=root、仅限所有者的读/写）。

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

这些命令应生成以下结果。

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

7. 编辑 `/etc/httpd/conf.d/ssl.conf` 以反映您的新证书和密钥文件。

   a. 在 Apache 的 `SSLCertificateFile` 指令中提供 CA 签名主机证书的路径和文件名：

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

   b. 如果您收到一个中间证书文件（此示例中为 `intermediate.crt`），请使用 Apache 的 `SSLCACertificateFile` 指令提供其路径和文件名：

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

   > **Note**
   >
   > 有些证书将主机证书和中间证书 CAs 合并到一个文件中，因此没有必要使用该`SSLCACertificateFile`指令。请查询您的 CA 提供的说明。

   c. 在 Apache 的 `SSLCertificateKeyFile` 指令中提供私有密钥的路径和文件名（在该示例中为 `custom.key`）：

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

8. 保存 `/etc/httpd/conf.d/ssl.conf` 并重启 Apache。

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

9. 通过在浏览器 URL 栏中输入带有 `https://` 前缀的域名来测试您的服务器。您的浏览器应通过 HTTPS 加载测试页面而不会产生错误。

步骤 3：测试和强化安全配置

在 TLS 可操作且公开发布后，应测试其实际安全性。使用在线服务（例如 Qualys SSL Labs，该服务可对您的安全设置执行免费的全面分析）可轻松执行此操作。根据结果，您可以决定通过控制接受的协议、首选的密码和排除的密码来强化默认安全配置。有关更多信息，请参阅 Qualys 如何用公式表示其分数。

> **Important**
>
> 实际测试对服务器的安全性非常重要。少量配置错误可能导致严重的安全漏洞和数据丢失。由于建议的安全实践会不断变化以响应调查和新兴威胁，因此定期安全审核对于良好的服务器管理来说是必不可少的。

在 Qualys SSL Labs 站点上，使用 **`www.example.com`** 格式输入服务器的完全限定域名。约两分钟后，您将收到您站点的评级 (从 A 到 F) 和结果的详细信息。下表汇总了设置与 AL2 023 上默认 Apache 配置相同且具有默认 Certbot 证书的域的报告。

总评	B
证书	100%
协议支持	95%
密钥交换	70%
密码强度	90%

虽然概述信息显示配置基本正确，但详细报告标记了几个潜在的问题（在此处按严重性顺序列出）：

✗ 某些较旧的浏览器支持使用该 RC4 密码。密码是加密算法的数学核心。 RC4是一种用于加密 TLS 数据流的快速密码，众所周知，它有几个严重的弱点。除非您有充分理由支持旧版浏览器，否则，应禁用该密码。

✗ 支持旧 TLS 版本。该配置支持 TLS 1.0（已弃用）和 TLS 1.1（即将弃用）。从 2018 年开始，仅建议使用 TLS 1.2。

✗ 不完全支持向前保密性。向前保密性是一种算法功能，它使用从私有密钥派生的临时会话密钥进行加密。这意味着，在实践中，攻击者无法解密 HTTPS 数据，即使他们拥有 Web 服务器的长期私有密钥。

纠正 TLS 配置并供将来使用

1. 在文本编辑器中打开 `/etc/httpd/conf.d/ssl.conf` 配置文件，并在以下行的开头输入“#”以注释掉该行。

```
#SSLProtocol all -SSLv3
```

2. 添加以下指令：

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

该指令显式禁用 SSL 版本 2 和 3 以及 TLS 版本 1.0 和 1.1。现在，服务器拒绝接受与使用 TLS 1.2 以外的任何协议的客户端之间的加密连接。指令中的冗长文字更清楚地向人类读者阐述为服务器配置的用途。

> **Note**
>
> 以此方式禁用 TLS 1.0 和 1.1 版可阻止一小部分过时的 Web 浏览器访问您的网站。

修改允许的密码列表

1. 在 `/etc/httpd/conf.d/ssl.conf` 配置文件中，找到包含 **`SSLCipherSuite`** 指令的部分，并通过在现有行的开头输入“#”来注释掉该行。

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

2. 指定显式的密码套件，并指定密码顺序以优先使用向前保密性并避免不安全的密码。此处使用的 `SSLCipherSuite` 指令基于 Mozilla SSL 配置生成器的输出，该生成器根据服务器上运行的特定软件定制 TLS 配置。首先，通过使用以下命令的输出确定 Apache 和 OpenSSL 版本。

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

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

例如，如果返回的信息是 Apache 2.4.34 和 OpenSSL 1.0.2，我们将其输入到生成器中。如果您选择“现代”兼容性模型，这将创建一条 `SSLCipherSuite` 指令，虽然该指令积极实施安全性，但仍适用于大多数浏览器。如果您的软件不支持现代配置，则可以更新软件或改为选择“中间”配置。

```
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
```

选定的密码名称中包含 ECDHE，它是 Elliptic Curve Diffie-Hellman Ephemeral 的缩写。术语 ephemeral 表示向前保密性。顺便说一句，这些密码不支持。 RC4

建议您使用密码的明确列表，而不是依赖于内容不可见的默认值或简短指令。

将生成的指令复制到 `/etc/httpd/conf.d/ssl.conf` 中。

> **Note**
>
> 此处为方便阅读将指令显示为几行，但在复制到 `/etc/httpd/conf.d/ssl.conf` 时，该指令必须位于一行中，并且密码名称之间只有一个冒号（无空格）。

3. 最后，通过删除以下行开头的“#”来取消对该行的注释。

```
#SSLHonorCipherOrder on
```

该指令强制服务器优先使用排名较高的密码，包括（在该示例中）支持向前保密性的密码。启用此指令后，服务器会在回滚到允许的安全性较低的密码之前尝试建立高度安全的连接。

在完成这两个过程后，将更改保存到 `/etc/httpd/conf.d/ssl.conf` 并重新启动 Apache。

如果您在 Qualys SSL Labs 上再次测试该域名，您应该会看到 RC4 漏洞和其他警告已消失，摘要如下所示。

总评	A
证书	100%
协议支持	100%
密钥交换	90%
密码强度	90%

在每次更新 OpenSSL 时，将引入新的密码并删除对旧密码的支持。保留您的 EC2 AL2 023 实例 up-to-date，留意来自 Open SSL 的安全公告，并警惕技术媒体上有关新安全漏洞的报道。

故障排除

- 除非我输入密码，否则我的 Apache Web 服务器不会启动

  如果您安装了受密码保护的加密的私有服务器密钥，这是预期行为。

  您可以从密钥中删除加密和密码要求。假设您在默认目录`custom.key`中有一个名为的私有加密 RSA 密钥，并且其密码是**`abcde12345`**，则在您的 EC2实例上运行以下命令以生成该密钥的未加密版本。

  ```
  [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 现在启动时应该不会提示您提供密码。

- 我在运行 sudo dnf install -y mod_ssl 时收到了错误。

在为 SSL 安装所需的程序包时，您可能会看到与以下内容类似的错误。

```
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
```

这通常意味着您的 EC2 实例未运行 AL2 023。本教程仅支持从官方 AL2 023 AMI 新创建的实例。

教程：在 AL2 023 上主持 WordPress 博客

以下过程将帮助您在 AL2 023 实例上安装、配置和保护 WordPress 博客。本教程很好地介绍了如何使用Amazon，因为您可以完全控制托管 WordPress 博客的网络服务器，这 EC2 在传统托管服务中并不常见。

您负责更新软件包并为您的服务器维护安全补丁。对于不需要与 Web 服务器配置直接交互的自动化程度更高的 WordPress 安装，该 AWS CloudFormation 服务提供了一个可以帮助您快速入门的 WordPress模板。有关更多信息，请参阅AWS CloudFormation 用户指南中的入门。如果您需要具有分离数据库的高可用性解决方案，请参阅开发人员指南中的部署高可用性 WordPress 网站。AWS Elastic Beanstalk

> **Important**
>
> 这些程序旨在与 AL2 023 一起使用。有关其他发布版本的信息，请参阅特定于该版本的文档。本教程中的很多步骤对 Ubuntu 实例并不适用。有关在 Ubuntu 实例 WordPress 上安装的帮助，请参阅 Ubuntu 文档WordPress中的。你也可以使用CodeDeploy在亚马逊 Linux、macOS 或 Unix 系统上完成此任务。

主题

- 先决条件
- 安装 WordPress
- 后续步骤
- 帮助！我的公有 DNS 名称发生更改导致我的博客瘫痪

先决条件

我们强烈建议您将弹性 IP 地址 (EIP) 关联到用于托管 WordPress 博客的实例。这将防止您的实例的公有 DNS 地址更改和中断您的安装。如果您有一个域名且打算将其用于您的博客，则可更新该域名的 DNS 记录，使其指向您的 EIP 地址 (如需帮助，请联系您的域名注册商)。您可以免费将一个 EIP 地址与正在运行的实例相关联。有关更多信息，请参阅 Amazon EC2 用户指南中的弹性 IP 地址。本 教程：在 AL2 023 上安装 LAMP 服务器 教程还介绍了配置安全组以允许 HTTP 和 HTTPS 流量的步骤，以及用于确保为 Web 服务器正确设置文件权限的几个步骤。有关向安全组添加规则的信息，请参阅向安全组添加规则。

如果您的博客还没有域名，则可使用 Route 53 注册一个域名并将您的实例的 EIP 地址与您的域名相关联。有关更多信息，请参阅 Amazon Route 53 开发人员指南中的使用 Amazon Route 53 注册域名。

安装 WordPress

连接到您的实例，然后下载 WordPress 安装包。有关连接到实例的更多信息，请参阅正在连接到 AL2 023 个实例。

1. 使用以下命令下载并安装这些软件包。

```
dnf install wget php-mysqlnd httpd php-fpm php-mysqli mariadb105-server php-json
 php php-devel -y
```

2. 您可能会发现显示一条与输出中措辞类似的警告（版本可能随着时间变化）：

```
WARNING:
  A newer release of "Amazon Linux" is available.

  Available Versions:

dnf upgrade --releasever=2023.0.20230202

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

  Version 2023.0.20230204:
    Run the following command to update to 2023.0.20230204:

      dnf upgrade --releasever=2023.0.20230204 ... etc
```

作为最佳实践，我们建议 up-to-date尽可能保留操作系统，但您可能需要遍历每个版本，以确保您的环境中没有冲突。如果步骤 1 中提及的软件包安装失败，则可能需要更新到所列出的某个较新版本，然后重试。

3. 使用wget命令下载最新的 WordPress 安装包。以下命令始终会下载最新版本。

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

4. 解压并解档安装包。安装文件夹解压到名为 `wordpress` 的文件夹。

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

为您的 WordPress 安装创建数据库用户和数据库

您的 WordPress 安装需要将博客文章和用户评论等信息存储在数据库中。此过程帮助您创建自己的博客数据库，并创建一个有权读取该数据库的信息并将信息保存到该数据库的用户。

1. 启动数据库和 Web 服务器。

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

2. 以 `root` 用户身份登录数据库服务器。在系统提示时输入您的数据库 `root` 密码，它可能与您的 `root` 系统密码不同；如果您尚未给您的数据库服务器加密，它甚至可能是空的。

   如果您尚未给您的数据库服务器加密，则必须执行这项操作。有关更多信息，请参阅 步骤 3：确保数据库服务器的安全 (AL2023)。

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

3. 为您的 MySQL 数据库创建用户和密码。您的 WordPress安装使用这些值与您的 MySQL 数据库进行通信。输入以下命令，以替换唯一的用户名和密码。

```
CREATE USER 'wordpress-user'@'localhost' IDENTIFIED BY 'your_strong_password';
```

   确保为您的用户创建强密码。请勿在您的密码中使用单引号字符 (')，因为这将中断前面的命令。请勿重复使用现有密码，并确保将密码保存在安全的位置。

4. 创建数据库。为数据库提供一个有意义的描述性名称，例如 `wordpress-db`。

> **Note**
>
> 以下命令中数据库名称两边的标点符号称为反引号。在标准键盘上，反引号 (`) 键通常位于 Tab 键的上方。并不总是需要反引号，但是它们允许您在数据库名称中使用其他的非法字符，例如连字符。

```
CREATE DATABASE `wordpress-db`;
```

5. 向之前创建的 WordPress 用户授予数据库的完全权限。

```
GRANT ALL PRIVILEGES ON `wordpress-db`.* TO "wordpress-user"@"localhost";
```

6. 刷新数据库权限以接受您的所有更改。

```
FLUSH PRIVILEGES;
```

7. 退出 `mysql` 客户端。

```
exit
```

创建和编辑 wp-config.php 文件

WordPress 安装文件夹包含一个名为的示例配置文件`wp-config-sample.php`。在本步骤中，您将复制此文件并进行编辑以适合您的具体配置。

1. 将 `wp-config-sample.php` 文件复制为一个名为 `wp-config.php` 的文件。这样做会创建新的配置文件并将原先的示例配置文件原样保留作为备份。

```
[ec2-user ~]$ cp wordpress/wp-config-sample.php wordpress/wp-config.php
```

2. 用您喜欢的文本编辑器（例如 nano 或 vim）编辑 `wp-config.php` 文件并输入适用于您的安装的值。如果没有常用的文本编辑器，nano 比较适合初学者使用。

```
[ec2-user ~]$ nano wordpress/wp-config.php
```

a. 查找定义 DB_NAME 的行并将 database_name_here 更改为您在 Step 4 的 为您的 WordPress 安装创建数据库用户和数据库 中创建的数据库名称。

```
define('DB_NAME', 'wordpress-db');
```

b. 查找定义 DB_USER 的行并将 username_here 更改为您在 Step 3 的 为您的 WordPress 安装创建数据库用户和数据库 中创建的数据库用户。

```
define('DB_USER', 'wordpress-user');
```

c. 查找定义 DB_PASSWORD 的行并将 password_here 更改为您在 Step 3 的 为您的 WordPress 安装创建数据库用户和数据库 中创建的强密码。

```
define('DB_PASSWORD', 'your_strong_password');
```

d. 查找名为 Authentication Unique Keys and Salts 的一节。这些KEY和SALT值为 WordPress 用户存储在本地计算机上的浏览器 Cookie 提供了一层加密。总而言之，添加长的随机值将使您的站点更安全。访问 https://api.wordpress。 org/secret-key/1.1/salt/以随机生成一组密钥值，您可以将其复制并粘贴到wp-config.php文件中。要粘贴文本到 PuTTY 终端，请将光标放在您要粘贴文本的地方，并在 PuTTY 终端内部右键单击鼠标。

有关安全密钥的更多信息，请访问 https://wordpress。 org/support/article/editing-wp-config-php/#security-keys。

> **Note**
>
> 以下值仅用作示例；请勿使用以下值进行安装。

```
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. 保存文件并退出文本编辑器。

将 WordPress 文件安装在 Apache 文档根目录下

- 现在，您已经解压缩了安装文件夹，创建了 MySQL 数据库和用户，并自定义了 WordPress 配置文件，接下来就可以将安装文件复制到 Web 服务器文档根目录了，这样就可以运行完成安装的安装脚本了。这些文件的位置取决于您是希望 WordPress 博客在 Web 服务器的实际根目录（例如*my.public.dns.amazonaws.com*）中可用，还是在根目录下的子目录或文件夹（例如*my.public.dns.amazonaws.com/blog*）中可用。
  - 如果 WordPress 要在文档根目录下运行，请按如下方式复制 wordpress 安装目录的内容（但不是目录本身）：

    ```
    [ec2-user ~]$ cp -r wordpress/* /var/www/html/
    ```

  - 如果 WordPress 要在文档根目录下的备用目录中运行，请先创建该目录，然后将文件复制到该目录。在此示例中， WordPress将从以下目录运行`blog`：

    ```
    [ec2-user ~]$ mkdir /var/www/html/blog
    [ec2-user ~]$ cp -r wordpress/* /var/www/html/blog/
    ```

> **Important**
>
> 出于安全原因，如果您不打算立即进入到下一个过程，请立即停止 Apache Web 服务器 (`httpd`)。将安装移至 Apache 文档根目录下后， WordPress 安装脚本将不受保护，如果 Apache Web 服务器正在运行，攻击者可能会访问您的博客。要终止 Apache Web 服务器，请输入命令 sudo service httpd stop。如果您即将继续到下一个步骤，则不需要终止 Apache Web 服务器。

允许 WordPress 使用永久链接

WordPress 永久链接需要使用 Apache `.htaccess` 文件才能正常工作，但是 Amazon Linux 上默认不启用此功能。使用此过程可允许 Apache 文档根目录中的所有覆盖。

1. 使用您常用的文本编辑器（如 vim 或 nano）打开 `httpd.conf` 文件。如果没有常用的文本编辑器，nano 比较适合初学者使用。

```
[ec2-user ~]$ sudo vim /etc/httpd/conf/httpd.conf
```

2. 找到以 `<Directory "/var/www/html">` 开头的部分。

```
<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. 在以上部分中将 `AllowOverride None` 行改为读取 `AllowOverride All`。

> **Note**
>
> 此文件中有多个 `AllowOverride` 行；请确保更改 `<Directory "/var/www/html">` 部分中的行。

```
AllowOverride All
```

4. 保存文件并退出文本编辑器。

在 AL2 023 上安装 PHP 图形绘图库

PHP 的 GD 库允许您修改图像。如果您需要裁剪博客的标题图像，请安装此库。您安装 phpMyAdmin 的版本可能需要此库的特定最低版本（例如 8.1 版）。

使用以下命令在 AL2 023 上安装 PHP 图形绘图库。例如，如果您在安装 LAMP 堆栈的过程中从源安装了 php8.1，则此命令将安装 8.1 版的 PHP 图形绘图库。

```
[ec2-user ~]$ sudo dnf install php-gd
```

要验证安装的版本，请使用以下命令：

```
[ec2-user ~]$ sudo dnf list installed | grep php-gd
```

下面是示例输出：

```
php-gd.x86_64                    8.1.30-1.amzn2            @amazonlinux
```

在 Amazon Linux AMI 上安装 PHP 图形绘图库

PHP 的 GD 库允许您修改图像。如果您需要裁剪博客的标题图像，请安装此库。您安装 phpMyAdmin 的版本可能需要此库的特定最低版本（例如 8.1 版）。

要验证哪些版本可用，请使用以下命令：

```
[ec2-user ~]$ dnf list | grep php
```

以下是 PHP 图形绘图库（8.1 版）的输出中的示例行：

```
php8.1.aarch64                                                    8.1.7-1.amzn2023.0.1
                       @amazonlinux
php8.1-cli.aarch64                                                8.1.7-1.amzn2023.0.1
                       @amazonlinux
php8.1-common.aarch64                                             8.1.7-1.amzn2023.0.1
                       @amazonlinux
php8.1-devel.aarch64                                              8.1.7-1.amzn2023.0.1
                       @amazonlinux
php8.1-fpm.aarch64                                                8.1.7-1.amzn2023.0.1
                       @amazonlinux
php8.1-gd.aarch64                                                 8.1.7-1.amzn2023.0.1
                       @amazonlinux
```

使用以下命令在 Amazon Linux AMI 上安装特定版本的 PHP 图形绘图库（例如 php 8.1 版）：

```
[ec2-user ~]$ sudo dnf install -y php8.1-gd
```

修复 Apache Web 服务器的文件权限

中的某些可用功能 WordPress 需要对 Apache 文档根目录具有写入权限（例如通过 “管理” 屏幕上传媒体）。如果您尚未进行此操作，请应用以下组成员关系和权限 (在 LAMP Web 服务器教程中有更为详细的描述)。

1. 将 `/var/www` 及其内容的文件所有权授予 `apache` 用户。

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

2. 将 `/var/www` 及其内容的组所有权授予 `apache` 组。

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

3. 更改 `/var/www` 及其子目录的目录权限，以添加组写入权限及设置未来子目录上的组 ID。

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

4. 递归地更改 `/var/www` 及其子目录的文件权限。

   ```
   [ec2-user ~]$ find /var/www -type f -exec sudo chmod 0644 {} \;
   ```

> **Note**
>
> 如果您还打算用 WordPress 作 FTP 服务器，则需要在此处进行更宽松的群组设置。请查看中的建议步骤和安全设置 WordPress以完成此操作。

5. 重启 Apache Web 服务器，让新组和权限生效。

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

使用 AL2 023 运行 WordPress 安装脚本

您已准备好进行安装 WordPress。您使用的命令取决于操作系统。此过程中的命令适用于 AL2 023。在 AL2 023 AMI 中使用此步骤之后的步骤。

1. 使用 systemctl 命令确保 `httpd` 和数据库服务在每次系统启动时启动。

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

2. 验证数据库服务器是否正在运行。

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

   如果数据库服务未运行，请启动。

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

3. 验证您的 Apache Web 服务器 (`httpd`) 正在运行。

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

   如果 `httpd` 服务未运行，请启动。

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

4. 在 Web 浏览器中，键入 WordPress 博客的 URL（要么是您的实例的公有 DNS 地址，要么是该地址后跟`blog`文件夹）。您应该会看到 WordPress 安装脚本。提供 WordPress 安装所需的信息。选择 “安装” WordPress 以完成安装。有关更多信息，请参阅 WordPress 网站上的步骤 5：运行安装脚本。

使用 AL2 023 AMI 运行 WordPress 安装脚本

1. 使用 chkconfig 命令确保 `httpd` 和数据库服务在每次系统启动时启动。

```
[ec2-user ~]$ sudo chkconfig httpd on && sudo chkconfig mariadb on
```

2. 验证数据库服务器是否正在运行。

```
[ec2-user ~]$ sudo service mariadb status
```

   如果数据库服务未运行，请启动。

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

3. 验证您的 Apache Web 服务器 (`httpd`) 正在运行。

```
[ec2-user ~]$ sudo service httpd status
```

   如果 `httpd` 服务未运行，请启动。

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

4. 在 Web 浏览器中，键入 WordPress 博客的 URL（要么是您的实例的公有 DNS 地址，要么是该地址后跟`blog`文件夹）。您应该会看到 WordPress 安装脚本。提供 WordPress 安装所需的信息。选择 “安装” WordPress 以完成安装。有关更多信息，请参阅 WordPress 网站上的步骤 5：运行安装脚本。

后续步骤

测试完 WordPress 博客后，可以考虑更新其配置。

使用自定义域名

如果您有与 EC2 实例的 EIP 地址关联的域名，则可以将博客配置为使用该名称而不是 EC2 公有 DNS 地址。有关更多信息，请参阅更改网站上的站点 URL。 WordPress

配置您的博客

您可以将您的博客配置为使用不同的主题和插件，从而向您的读者提供更具个性化的体验。但是，有时安装过程可能事与愿违，从而导致您丢失您的整个博客。强烈建议您在尝试安装任何主题或插件之前，

为您的实例创建一个备份 Amazon Machine Image (AMI)，以便在安装过程中出现任何问题时，您还可以还原您的博客。有关更多信息，请参阅《亚马逊 EC2 用户指南》中的创建您自己的 AMI。

增加容量

如果您的 WordPress 博客越来越受欢迎，并且您需要更多的计算能力或存储空间，请考虑以下步骤：

- 对实例扩展存储空间。有关更多信息，请参阅 Amazon EBS 弹性卷。
- 将您的 MySQL 数据库移动到 Amazon RDS 以利用服务的轻松扩展功能。

提高互联网流量的网络性能

如果希望您的博客吸引世界各地用户的流量，请考虑 AWS Global Accelerator。Global Accelerator 通过改善用户的客户端设备和运行的 WordPress 应用程序之间的互联网流量性能，帮助您降低延迟 AWS。Global Accelerator 使用AWS 全球网络将流量引导到离客户端最近的 AWS 区域中运行良好的应用程序终端节点。

了解更多关于 WordPress

以下链接包含有关的更多信息 WordPress。

- 有关信息 WordPress，请参阅 WordPress Codex 上的 Codex 帮助文档。
- 有关安装疑难解答的更多信息，请转到常见安装问题。
- 有关提高 WordPress 博客安全性的信息，请访问强化 WordPress。
- 有关保留 WordPress 博客的信息 up-to-date，请前往更新 WordPress。

帮助！我的公有 DNS 名称发生更改导致我的博客瘫痪

您的 WordPress 安装是使用您的 EC2 实例的公有 DNS 地址自动配置的。如果您停止并重启实例，则公有 DNS 地址会发生变化（除非它与弹性 IP 地址相关联），并且您的博客将无法运行，因为它引用了不再存在（或已分配给其他 EC2 实例）的地址上的资源。https://wordpress 中概述了对该问题的更详细描述和几种可能的解决方案。 org/support/article/changing-the-site-url/。

如果 WordPress 安装时发生了这种情况，则可以通过以下步骤恢复博客，该过程使用wp-cli命令行界面 WordPress。

要使用更改您的 WordPress 网站网址 wp-cli

1. 使用 SSH 连接到您的 EC2 实例。

2. 请记下您的实例的旧站点 URL 和新站点 URL。安装时，旧站点 URL 很可能是您的 EC2 实例的公有 DNS 名称 WordPress。新的站点 URL 是您的 EC2实例的当前公有 DNS 名称。如果您不确定旧站点 URL 是什么，则可通过以下命令使用 curl 来查找它。

```
[ec2-user ~]$ curl localhost | grep wp-content
```

您应该会在输出中看到对您的旧公有 DNS 名称的引用，如下所示 (旧站点 URL 用红色表示)：

```
<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. 使用以下命令下载 wp-cli。

```
[ec2-user ~]$ curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/
phar/wp-cli.phar
```

4. 使用以下命令搜索并替换 WordPress 安装中的旧站点 URL。用新旧站点代 URLs 替您的 EC2 实例和 WordPress 安装路径（通常为 o `/var/www/html` r`/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. 在 Web 浏览器中，输入 WordPress 博客的新网站 URL，以验证该网站是否恢复正常运行。如果不是，请参阅更改站点 URL 和常见安装问题了解更多信息。

教程：023 年 Redis 6 到 Valkey 过渡 AL2

以下文档描述了 023 年从 Redis 6 过渡到 Valkey 的关键方面 AL2。

Redis 6 的 Support 时间表

Redis 6 将于 2025 年 8 月 31 日达到使用寿命终结 (EOL)。在此日期之后，Redis 6 将不再收到来自 Redis 项目的更新或安全补丁。我们强烈建议用户在 2025 年 8 月之前迁移到 Valkey，以确保持续的支持和安全更新。

有关 Redis 版本支持时间表的更多信息，请参阅 Redis End-Of-Life 计划文档。

Valkey 简介

Valkey 是 Redis 7 的开源分支，由 Linux 基金会维护。它与 Redis 开源软件 (OSS) 版本 2.x 到 7.2.x 完全兼容。Valkey 保留了熟悉的 Redis API 和功能，同时提供了多项增强功能：

- 通过多线程增强性能。
- 提高了内存效率，尤其是在集群模式下。
- 双通道复制可提高数据一致性。

迁移计划和时间表

强烈建议用户在2025年8月31日Redis 6进入生命周期（EOL）之前从Redis 6迁移到Valkey。这种迁移需要手动干预，而且不是自动的。

Amazon Linux 建议进行此迁移，以确保依赖于 Redis 的应用程序持续提供功能、支持和安全更新。

迁移选项和步骤

根据您的部署要求和运营需求，我们建议三种迁移到 Valkey 的途径。

选项 1：安装新实例

对于新部署或不需要数据迁移时：

1. 安装 Valkey：

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

2. 启动 Valkey：

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

3. （可选）在启动时启用 Valkey：

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

4. 验证安装：

```
[ec2-user ~]$ valkey-cli info server
```

```
[ec2-user ~]$ valkey-cli ping
```

选项 2：就地更换

对于不需要数据持久性的现有实例：

1. Stop Redis 6：

```
[ec2-user ~]$ sudo systemctl stop redis6
```

2. 安装 Valkey：

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

3. （可选）在 Valkey 中使用 Redis 6 配置：

```
[ec2-user ~]$ sudo cp /etc/redis6/redis6.conf /etc/valkey/valkey.conf
[ec2-user ~]$ sudo cp /etc/valkey/valkey.conf /etc/valkey/valkey.conf.backup
[ec2-user ~]$ sudo chown valkey:root /etc/valkey/valkey.conf
[ec2-user ~]$ sudo sed -i 's|^dir\s.*|dir /var/lib/valkey|g' /etc/valkey/
valkey.conf
```

4. （可选）在 Valkey 中使用 Redis 6 哨兵配置文件：

```
[ec2-user ~]$ sudo cp /etc/redis6/sentinel.conf /etc/valkey/sentinel.conf
[ec2-user ~]$ sudo chown valkey:root /etc/valkey/sentinel.conf
```

5. 启动 Valkey：

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

6. （可选）在启动时启用 Valkey：

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

7. 验证 Valkey 的安装情况：

```
[ec2-user ~]$ valkey-cli info server
[ec2-user ~]$ valkey-cli ping
```

8. 移除 Redis 6：

```
[ec2-user ~]$ sudo dnf remove redis6
```

选项 3：数据迁移

此选项允许你同时运行 Redis 6 和 Valkey。

1. 在不移除 Redis 6 的情况下安装 Valkey：

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

2. （可选）在 Valkey 中使用 Redis 6 配置：

```
[ec2-user ~]$ sudo cp /etc/redis6/redis6.conf /etc/valkey/valkey.conf
[ec2-user ~]$ sudo cp /etc/valkey/valkey.conf /etc/valkey/valkey.conf.backup
[ec2-user ~]$ sudo chown valkey:root /etc/valkey/valkey.conf
[ec2-user ~]$ sudo sed -i 's|^dir\s.*|dir /var/lib/valkey|g' /etc/valkey/
valkey.conf
```

3. （可选）在 Valkey 中使用 Redis 6 哨兵配置文件：

```
[ec2-user ~]$ sudo cp /etc/redis6/sentinel.conf /etc/valkey/sentinel.conf
[ec2-user ~]$ sudo chown valkey:root /etc/valkey/sentinel.conf
```

4. 修改 Valkey 配置：

   编辑 “端口” 指令`/etc/valkey/valkey.conf`并将其设置为不同的值（例如 6380），以避免与 Redis 6 发生冲突。

5. 启动 Valkey：

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

6. （可选）在启动时启用 Valkey：

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

7. 验证 Valkey 的安装情况：

```
[ec2-user ~]$ valkey-cli -p port info server
[ec2-user ~]$ valkey-cli -p port ping
```

> **Note**
>
> *port*替换为配置的端口号。

8. 迁移数据：

   现在，您可以使用复制或手动数据传输方法将数据从 Redis 6 迁移到 Valkey。

9. 更新应用程序配置：

   逐渐更新您的应用程序以使用 Valkey 端口。

10. 移除 Redis 6：

    迁移完所有数据和应用程序后，您可以停止并删除 Redis 6。

```
[ec2-user ~]$ sudo systemctl stop redis6
[ec2-user ~]$ sudo dnf remove redis6
```

> **Note**
>
> 强烈建议在生产系统中实施更改之前，先在测试环境中验证迁移过程。

相关 主题

有关 Valkey 的更多信息，请执行以下操作：

- Valkey：https://valkey.io/
- Valkey 迁移：https://valkey。 io/topics/migration/

教程：在 023 上 AL2安装 GNOME 桌面环境

从 2023.7 版或更高版本起，GNOME 桌面环境可作为 AL2 023 的可选图形用户界面提供。

以下过程可帮助您在 AL2 023 实例上安装 GNOME 桌面环境。您可以使用此图形界面，使用熟悉的桌面环境与您的 Linux 系统进行交互，而不仅仅是命令行界面。

内容

- 前提条件
- 安装
- 相关 主题

前提条件

- 桌面环境至少需要 2.4 GB 的内存。因此，建议使用类型`t2.medium`或更好的实例以确保足够的性能。内存不足的实例类型的示例包括`t2.nano``t2.micro`、和`t2.small`。此限制也适用于`t3`这种大小的`t4`实例，以及任何其他不符合内存要求的实例类型。
- 本教程假设您已经使用运行版本 2023.7 或更高版本的 AL2 023 启动了实例。有关更多信息，请参阅AL2亚马逊上的 023 EC2和正在更新 AL2 023页面。

安装

- 安装 GNOME 桌面环境和相关软件包。

```
[ec2-user ~]$ sudo dnf groupinstall "Desktop" -y
```

> **Note**
>
> 要访问图形桌面环境，您需要安装和配置其他软件，例如 Amazon DCV 或 VNC。这些工具允许您通过网络连接图形用户界面并与之交互。

相关 主题

有关图形桌面环境的更多信息，请参阅以下文档：

- 什么是亚马逊 DCV？ 在 Amazon DCV 管理员指南中
- 教程：在 023 上配置 TigerVNC 服务器 AL2

教程：在 023 上配置 TigerVNC 服务器 AL2

以下过程可帮助您在 AL2 023 实例上设置 VNC 服务器。VNC 允许您通过安全的网络连接远程访问图形桌面环境并与之交互。

内容

- 前提条件
- 步骤 1：安装
- 步骤 2：配置
- 步骤 3：使用 VNC 客户端进行连接
- （可选）启动时启动服务
- （可选）禁用空闲锁屏
- 相关 主题

前提条件

- 本教程假设您已经在 AL2 023 实例上安装了 GNOME 桌面环境。有关更多信息，请参阅教程：在 023 上 AL2安装 GNOME 桌面环境页面。
- 本教程使用 SSH 端口转发来访问 VNC 服务器。有关设置密钥对的更多信息，请参阅亚马逊 EC2 用户指南中的使用 SSH 连接您的 Linux 实例。
- 以下步骤不会指导您完成安装 VNC 客户端的过程。您必须在本地计算机上安装 VNC 客户端，才能连接到桌面环境并与之交互。

步骤 1：安装

1. 连接到您的实例。有关更多信息，请参阅 正在连接到 AL2 023 个实例。
2. 安装 023 版的 TigerVNC 服务器软件包。 AL2

   该-y选项无需确认即可安装软件包。如果您想在安装之前检查软件包，可以省略此选项。

   ```
   [ec2-user ~]$ sudo dnf install -y tigervnc-server
   ```

步骤 2：配置

1. 确保用户已配置 VNC 密码。

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

2. 为用户分配显示号码。

```
[ec2-user ~]$ sudo vi /etc/tigervnc/vncserver.users
```

添加以下配置：

```
:1=ec2-user
```

> **Note**
>
> 您可以为用户分配任何显示号码。为了这个例子，我们使用显示器 `:1`。

3. 编辑 VNC 服务器配置文件。

```
[ec2-user ~]$ sudo vi /etc/tigervnc/vncserver-config-defaults
```

添加以下配置：

```
session=gnome
securitytypes=vncauth,tlsvnc
geometry=1920x1080
localhost
alwaysshared
```

> **Note**
>
> 您可以使用`geometry`参数更改显示屏的分辨率。为了这个例子，我们使用`1920x1080`了。

4. 启动 VNC 服务器。每次重启实例时，都需要重复此过程。如果您想自动启动此服务，请参阅下面的可选部分。

```
[ec2-user ~]$ sudo systemctl start vncserver@:1
```

> **Important**
>
> 启动`vncserver`服务时，后面的部分 `@`必须与 `/etc/tigervnc/vncserver.users`文件中为用户设置的显示编号相匹配。

执行此步骤后，您可以从本地计算机创建 SSH 隧道并使用 VNC 客户端进行连接。

步骤 3：使用 VNC 客户端进行连接

VNC 服务器公开了一个用于客户端连接的 TCP 套接字。虽然您可以直接通过安全组公开 VNC 端口，但本教程演示了通过加密本地计算机与实例之间的连接来使用 SSH 隧道作为一种更安全的方法。 EC2 通过隧道连接后，您将使用在上一步中配置的密码向 VNC 服务器进行身份验证。有关安全组的更多信息，请参阅亚马逊 EC2 用户指南中的更改亚马逊 EC2 实例的安全组。

1. 从本地计算机创建 SSH 隧道。

```
$ ssh -i <keypair> -L 5901:localhost:5901 ec2-user@<address>
```

> **Note**
>
> `<keypair>`替换为 SSH 密钥的路径以及实例 `<address>`的公有 IP 或 DNS 名称。端口会根据用于启动的显示编号而变化 `vncserver`。例如，显示器`:1`使用端口`5901`，显示器`:2`使用端口`5902`等。

2. 使用您的 VNC 客户端连接`localhost:5901`或使用先前设置的 VNC 密码 `127.0.0.1:5901`进行连接。

> **Important**
>
> 使用 VNC 时保持 SSH 隧道处于打开状态。如果 SSH 隧道未打开，您将无法使用 VNC 客户端查看桌面环境并与之交互。

（可选）启动时启动服务

如果您计划定期使用 VNC，则可能需要将 VNC 服务器配置为在实例启动时自动启动。这样就无需在每次重启实例时手动启动 VNC 服务器。此配置可确保您的图形桌面环境在实例完成启动过程后立即准备就绪并可以访问。

- 将服务配置为在启动时启动。

```
[ec2-user ~]$ sudo systemctl enable vncserver@:1
```

> **Important**
>
> 启用该vncserver服务时，后面的部分 @必须与 /etc/tigervnc/vncserver.users文件中为用户设置的显示编号相匹配。此外，你可以在之后传递 --now参数enable以立即启动服务。

执行此步骤后，您将不再需要在 vncserver每次重启实例时都启动。

（可选）禁用空闲锁屏

- 将空闲延迟设置为零，以便在用户长时间处于非活动状态时禁用锁屏。

```
[ec2-user ~]$ gsettings set org.gnome.desktop.session idle-delay 0
```

相关 主题

有关图形桌面环境的更多信息，请参阅以下文档：

- 教程：在 023 上 AL2安装 GNOME 桌面环境
- 什么是亚马逊 DCV？ 在 Amazon DCV 管理员指南中

在亚马逊之外使用亚马逊 Linux 2023 EC2

Amazon Linux 2023 容器映像可以在兼容的容器运行时环境中运行。有关如何在容器内使用 Amazon Linux 2023 的更多信息，请参阅AL2023 装在容器里。

除了直接在亚马逊上运行之外，亚马逊 Linux 2023 (AL2023) 也可以作为虚拟客户机运行。 EC2目前有 KVM (qcow2)、 VMware (OVA) 和 Hyper-V (vhdx) 图像可用。

> **Note**
>
> Amazon Linux 2023 的映像配置不同于 Amazon Linux 2。
> 如果您在本地以虚拟机形式运行 Amazon Linux 2，则需要调整配置以与 AL2 023 兼容。

下载亚马逊 Linux 2023 镜像，用于 KVM VMware、和 Hyper-V

用于 KVM 和 Hyper-V 的亚马逊 Linux 2023 磁盘映像可以从 VMware cdn.amazonlinux.com 下载。

支持在非亚马逊 EC2 虚拟化环境中使用的亚马逊 Linux 2023 配置

本节介绍在非亚马逊 EC2 虚拟化环境（例如 KVM、 VMware或和 Hyper-V）中运行亚马逊 Linux 2023 的要求。

该基础AL2023 系统要求适用于所有非亚马逊 EC2 虚拟化环境。以下主题详细列出对于每个虚拟机监控器环境所支持的设备型号列表。

KVM VMware、和 Hyper-V 提供了许多配置选项，需要谨慎配置它们以满足您的安全、性能和可靠性需求。有关更多信息，请查看虚拟机监控器提供的文档。

主题

- 在 KVM 上运行 AL2 023 的要求
- 在上运行 AL2 023 的要求 VMware
- 在 Hyper-V 上运行亚马逊 Linux 2023 的要求

在 KVM 上运行 AL2 023 的要求

本节介绍在 KVM 上运行 AL2 023 的要求。 AL2023 的 KVM 镜像适用于两种架构`aarch64`。`x86-64`这些要求是对 KVM 映像基础AL2023 系统要求要求的补充。

主题

- 在 KVM 上运行 AL2 023 的 KVM 主机要求
- KVM 上支持 AL2 023 的设备
- 启动模式 (UEFI 以及 BIOS) 在 KVM 上支持 AL2 023
- 在 KVM 上运行 AL2 023 的限制

在 KVM 上运行 AL2 023 的 KVM 主机要求

KVM 镜像目前在运行 Ubuntu 22.04.3 LTS 且版本的主机上已通过认证 qemu 6.2+dfsg-2ubuntu6.15，由此 Ubuntu 版本提供，使用`q35`计算机类型为`x86-64`，`virt`计算机类型为。`aarch64`

KVM 上支持 AL2 023 的设备

经测试可与 AL2 023 个 KVM 映像（两者**`aarch64`**兼有**`x86-64`**）一起使用的**`qemu`**设备型号为：

- `virtio-blk`（`virtio` 块设备）
- `virtio-scsi` (`virtio` SCSI 带磁盘设备的控制器）
- `virtio-net`（`virtio` 网络设备）
- `ahci`（用于虚拟 CD-ROM 驱动器）
- `usb-storage`（通过 `xhci`）

在 AL2 023 KVM 映像认证中启用但未大量使用的其他**`qemu`**设备型号有：

- VGA (qemu VGA) `x86-64` 仅开启
- `virtio-rng`（虚拟随机数生成器）
- 遗留 AT 键盘和 PS/2 鼠标设备
- 传统串行设备

启动模式 (UEFI 以及 BIOS) 在 KVM 上支持 AL2 023

该`x86-64`图像已使用两个旧版进行测试 BIOS 以及 UEFI 启动模式。`aarch64`图像经过测试 UEFI 启动模式。

> **Note**
>
> 默认情况下，使用时 UEFI 启动模式下，一些虚拟机管理器会为虚拟机配置支持安全启动的 Microsoft 安全启动密钥。此配置无法启动 AL2 023。
> 由于 AL2 023 启动加载程序不是由 Microsoft 签名的，因此必须配置虚拟机时要么没有 UEFI 密钥，要么使用 AL2 023 密钥进行安全启动。

> **Important**
>
> 安全启动支持 KVM图像尚未经过验证。

在 KVM 上运行 AL2 023 的限制

在 KVM 上运行 AL2 023 有一些已知的限制。

> **Note**
>
> 实现某些列出的不支持的功能的代码可能存在于 AL2 023 中，并且可以正常运行。存在不支持的功能列表，以便您可以做出明智的决定，决定今天要依赖什么来工作，以及 Amazon Linux 团队在未来更新中将有资格使用哪些功能。

在 KVM 上运行 AL2 023 的已知限制

- 目前未打包或不支持 KVM 客户机代理。
- 不支持热插拔 CPU、内存或任何其他设备类型。
- 不支持虚拟机休眠。
- 不支持虚拟机迁移。
- 不支持任何设备的透传，例如通过 PCI 透传或 USB 透传都不被支持。

在上运行 AL2 023 的要求 VMware

本节介绍在上运行 AL2 023 的要求 VMware。 的 VMware AL2023 的图像仅适用于该x86-64架构。VMware 的图片aarch64不可用或不支持。这些要求是对基本AL2023 系统要求要求的补充 VMware 图片。

主题

- VMware 在上运行 AL2 023 的主机要求 VMware
- 设备支持 AL2 023 开启 VMware
- 启动模式 (UEFI 以及 BIOS) 在 AL2 023 上支持 VMware
- 在 AL2 023 上运行的限制 VMware

VMware 在上运行 AL2 023 的主机要求 VMware

AL2023 VMware OVA 映像目前符合以下条件：

- VMware 在使用英特尔 (R) 至强 (R) 铂金 8124M 处理器的主机上运行的工作站 17.5.0
- VMware vSphere 8.0 使用英特尔 (R) 至强 (R) 铂金 8275CL 处理器

AL2023 VMware OVA 映像将计算机硬件版本指定为 13。

VMware 计算机硬件版本 13 受以下设备支持：

- ESXi 6.5 或更高版本
- VMware 工作站 14 或更高版本

设备支持 AL2 023 开启 VMware

以下 VMware 设备型号经过测试，可与 AL2 023 配合使用 VMware OVA 图片（**x86-64**仅限）：

- `vmw_pvscsi` (VMware 半虚拟化 SCSI 控制器）
- `vmxnet3` (VMware 半虚拟化网络设备）
- `ata_piix`（遗产 IDE 仅适用于虚拟 CD-ROM 驱动器）

额外 VMware AL2023 年启用的设备型号 VMware 图像资格，但没有大量使用：

- `vmw_vmci`和相关的`vsock`接口（虚拟套接字传输） VMware 客座代理）
- `vmw_balloon` 内存气球设备
- VMware `SVGA`控制器
- 遗留 AT 键盘和 PS/2 鼠标设备

这些区域有：VMware 客户机代理软件包 (`open-vm-tools`) 可用并默认安装在 AL2 023 中 VMware OVA 图像。

启动模式 (UEFI 以及 BIOS) 在 AL2 023 上支持 VMware

截至 2023.3.20231211 版本发布时，023 AL2 VMware OVA 映像已在两个旧版中都经过验证 BIOS 以及 UEFI 启动模式。OVA 默认配置仍然是传统配置 BIOS 但可以由用户更改。

> **Important**
>
> 需要安全启动支持 UEFI，尚未在 AL2 023 上进行验证 VMware.

在 AL2 023 上运行的限制 VMware

在上运行 AL2 023 有一些已知的限制 VMware.

> **Note**
>
> 实现某些列出的不支持的功能的代码可能存在于 AL2 023 中，并且可以正常运行。不被支持的功能列表之所以存在，是为了让客户能够就什么依赖当前工作，以及 Amazon Linux 团队将把什么视为未来更新的一部分工作做出明智决定。

运行 AL2 023 的已知限制 VMware

- UEFI 在 AL2 023 开启的情况下，当前未验证安全启动 VMware.
- 不支持热插拔 CPU、内存或任何其他设备类型。
- 不支持虚拟机休眠。
- 不支持虚拟机迁移。
- 不支持任何设备的透传，例如通过 PCI 透传或 USB 透传都不被支持。

在 Hyper-V 上运行亚马逊 Linux 2023 的要求

本节介绍在 Hyper-V 上运行亚马逊 Linux 2023 的要求。 AL2023 的 Hyper-V 映像仅适用于该架构。`x86-64``aarch64`目前尚不提供或不支持的 Hyper-V 图像。

除了 Hyper-V 映像的基础AL2023 系统要求外，本节还涵盖了其他要求。

主题

- 在 Hyper-V 上运行 Amazon Linux 2023 的 Hyper-V 主机要求
- 在 Hyper-V 上为亚马逊 Linux 2023 提供设备支持
- 在 Hyper-V 上运行亚马逊 Linux 2023 的限制

在 Hyper-V 上运行 Amazon Linux 2023 的 Hyper-V 主机要求

Hyper-V 上的 Amazon Linux 2023 的主要资格是在实例上运行的 Windows Server 2022 上进行 EC2 `c5.metal`的。

在 Hyper-V 上为亚马逊 Linux 2023 提供设备支持

Amazon Linux 2023 在第 1 代和第 2 代 Hyper-V 虚拟机上进行了测试，其中包含以下虚拟化硬件：

- 第 1 代（旧版 BIOS 启动）虚拟机
- 第 2 代（UEFI 启动-无安全启动）虚拟机
- 以下设备型号经过测试，可与 AL2 023 个 Hyper-V 映像配合使用：
  - 第 2 代中根磁盘和模拟 CD-ROM 驱动器的 Hyper-V 虚拟存储 *`hv_storvsc`* VMs
  - `ata_piix`为第 1 代的虚拟 CD-ROM 驱动器模拟了 PIIX IDE VMs
  - Hyper-V 虚拟以太网 `hv_netvsc`
- 以下设备型号已启用，但经过了轻微测试：
  - 第 1 代的传统 VGA 文本模式 VMs
  - 第 2 代基于 UEFI 固件的帧缓冲区 *`simpledrmfb`* VMs
  - Hyper-V 气球 `hv_balloon`
  - Hyper-V 气球 `hv_balloon`
  - Hyper-V Hid/Mouse `hid_hyperv`
- 目前未在 AL2 023 中启用以下设备模式：

- Hyper-V PCI 直通
- Hyper-V DRM 显卡

> **Important**
>
> 对于第 2 代虚拟机，不支持安全启动，必须在启动虚拟机之前将其禁用，才能成功启动 Amazon Linux 2023。Hyper-V 目前仅支持使用由微软自己的密钥签名的软件组件进行安全启动，而亚马逊 Linux 引导加载程序则由亚马逊私钥签名。Hyper-V 目前不支持导入第三方密钥。

在 Hyper-V 上运行亚马逊 Linux 2023 的限制

以下是在 Hyper-V 上运行 Amazon Linux 2023 时的一些已知限制：

> **Note**
>
> 实现某些列出的不支持的功能的代码可能存在于 AL2 023 中，并且可以正常运行。不被支持的功能列表之所以存在，是为了让客户能够就什么依赖当前工作，以及 Amazon Linux 团队将把什么视为未来更新的一部分工作做出明智决定。

在 Hyper-V 上运行 AL2 023 的已知限制

- Hyper-V 上的 AL2 023 目前不支持 UEFI 安全启动模式也不起作用
- 不支持热插拔 CPU、内存或任何其他设备类型。
- 不支持虚拟机 (VM) 休眠。
- 不支持虚拟机 (VM) 迁移。
- 不支持任何设备的透传，例如通过 PCI 透传或 USB 透传都不被支持。

亚马逊 Linux 2023 在亚马逊以外使用时的设置和`cloud-init`配置 EC2

本节介绍如何在不直接在亚马逊上运行时（例如在 KVM 或 Hyper-V 上运行）时设置和配置亚马逊 EC2 Linux 2023 虚拟机。 VMware

默认情况下，Amazon Linux 2023 虚拟机映像不提供任何用户密码或 ssh 密钥，将通过以下方式获取其网络配置 DHCP 在第一个发现的网络接口上。这意味着在默认情况下，没有其他配置，无法连接到生成的虚拟机。

因此，需要向虚拟机提供某种形式的配置。对于 Amazon Linux 来说，完成这项工作的标准机制是通过 `cloud-init` 数据源。

Amazon Linux 2023 已通过以下数据源的认证：

NoCloud

这是通过包含种子的虚拟 CD-ROM 配置本地映像的传统方法 ISO9660 带有`cloud-init`配置文件的图像。

VMware

Amazon Linux 2023 还支持通过和配置通过 VMware 特定数据源在 vSphere 上运行的 VMware 映像。 VMware `guestinfo.userdata` `guestinfo.metadata`

> **Note**
>
> 数据源的配置与 Amazon Linux 2 不同。更具体地说，Amazon Linux 2023 使用 `systemd-networkd` 来进行配置，并且要求使用 `cloud-init`“网络配置版本 2”，如 `cloud-init` 网络配置文档所述。

有关在 Amazon Linux 2023 中打包的 `cloud-init` 版本的 `cloud-init` 配置机制的完整文档，可从上游 `cloud-init` 文档中找到。

NoCloud (`seed.iso`) `cloud-init` 在 KVM 上对亚马逊 Linux 2023 进行配置以及 VMware

本节介绍如何创建和使用`seed.iso`映像来配置在上运行的亚马逊 Linux 2023 KVM 或 VMware。 因为 KVM 以及 VMware 环境没有 Amazon EC2 实例元数据服务 (IMDS)，则需要另一种配置 Amazon Linux 2023 的方法，而提供`seed.iso`图像就是其中一种方法。

`seed.iso` 启动映像包括启动和配置新的虚拟机所需的初始配置信息，例如网络配置、主机名和用户数据。

> **Note**
>
> `seed.iso` 映像仅包括启动虚拟机所需的配置信息，它不包括 Amazon Linux 2023 操作系统文件。

要生成 `seed.iso` 映像，您至少需要两个配置文件，有时则需要三个：

`meta-data`

此文件通常包含虚拟机的主机名。

`user-data`

此文件通常配置用户帐户及其密码，ssh 密钥对和/或访问机制。默认情况下，亚马逊 Linux 2023 KVM 和 VMware 镜像会创建一个`ec2-user`用户账户。您可以使用 `user-data` 配置文件为此默认用户账户设置密码和/或 ssh 密钥。

`network-config`（可选）

此文件通常为虚拟机提供网络配置，该配置将覆盖默认配置。默认配置是使用 DHCP 在第一个可用的网络接口上。

创建 **`seed.iso`** 磁盘映像

1. 在 Linux 或 MacOS 计算机上，创建一个名为 `seedconfig` 的新文件夹并导航到该文件夹。

   > **Note**
   >
   > 虽然可以使用 Windows 或其他操作系统完成这些步骤，但您必须找到 `mkisofs` 的等效工具才能完成创建 `seed.iso` 映像。

2. 创建 `meta-data` 配置文件。

   a. 创建名为 `meta-data` 的新文件。

   b. 使用您的首选编辑器打开`meta-data`文件并添加以下内容，*`vm-hostname`*替换为 VM 的主机名：

   ```
   #cloud-config
   local-hostname: vm-hostname
   ```

c. 保存并关闭 meta-data 配置文件。

3. 创建 user-data 配置文件。

a. 创建名为 user-data 的新文件。

b. 使用首选编辑器打开 user-data 文件并添加以下内容，进行必要的替换：

```
#cloud-config
#vim:syntax=yaml
users:
# A user by the name 'ec2-user' is created in the image by default.
  - default
  - name: ec2-user
ssh_authorized_keys:
  - ssh-rsa ssh-key
# In the above line, replace ssh key with the content of your ssh public key.
```

c. 您可以选择向user-data配置文件中添加更多用户帐户。

您可以指定其他用户账户、用户访问机制、密码和密钥对。有关支持的指令的更多信息，请参阅上游 cloud-init 文档。

d. 保存并关闭 user-data 配置文件。

4. （可选）创建 network-config 配置文件。

a. 创建名为 network-config 的新文件。

b. 使用首选编辑器打开 network-config 文件并添加以下内容，将各个 IP 地址替换为适合您的设置的地址。

```
#cloud-config
version: 2
ethernets:
  enp1s0:
    addresses:
      - 192.168.122.161/24
    gateway4: 192.168.122.1
    nameservers:
      addresses: 192.168.122.1
```

> **Note**
>
> cloud-init网络配置提供了与之匹配的机制 MAC 接口的地址，而不是指定接口名称，接口名称可能会根据虚拟机配置而变化。上游 cloud-init 网络配置版本 2 文档中详细描述了网络配置的这个（以及更多）cloud-init 功能。

c. 保存并关闭 network-config 配置文件。

5. 使用在前面步骤中创建的 meta-data、user-data 和可选的 network-config 配置文件创建 seed.iso 磁盘映像。

   根据要在其上创建 seed.iso 磁盘映像的操作系统，执行以下操作之一。

   - 在 Linux 系统上，使用 **mkisofs** 或 **genisoimage** 之类的工具来创建已完成的 seed.iso 文件。导航到 seedconfig 文件夹，并运行以下命令：

     ```
     $ mkisofs -output seed.iso -volid cidata -joliet -rock user-data meta-data
     ```

   - 如果您使用 network-config，则将其包含在 **mkisofs** 调用中：

     ```
     $ mkisofs -output seed.iso -volid cidata -joliet -rock user-data meta-data
      network-config
     ```

   - 在 macOS 系统上，您可以使用 **hdiutil** 之类的工具来生成已完成的 seed.iso 文件。由于 **hdiutil** 采用路径名而不是文件列表，因此无论是否创建了 network-config 配置文件，都可以使用相同的调用。

     ```
     $ hdiutil makehybrid -o seed.iso -hfs -joliet -iso -default-volume-name cidata
      seedconfig/
     ```

6. 现在，可以使用虚拟 CD-ROM 驱动器将生成的seed.iso文件附加到您的新 Amazon Linux 2023 虚拟机上，cloud-init以便在首次启动时查找并将配置应用于系统。

VMware AL2023 上的 guestinfo **cloud-init** 配置 VMware

VMware 环境没有 Amazon EC2 实例元数据服务 (IMDS)，因此需要另一种配置 AL2 023 的方法。本节介绍如何使用seed.iso虚拟CD-ROM驱动器的替代配置机制，该机制可在中找到 VMware vSphere。

此配置方法使用 VMware `extraconfig`向其提供配置数据的机制`cloud-init`。对于以下每个密钥，都必须提供相应的***keyname*.encoding**属性。

可以将以下密钥提供给 VMware `extraconfig`机制。

guestinfo.metadata

JSON 或 YAML 包含`cloud-init`元数据

guestinfo.userdata

A YAML 包含`cloud-config`格式的`cloud-init`用户数据的文档。

guestinfo.vendordata（可选）

YAML 包含`cloud-init`供应商数据

相应的编码属性（`guestinfo.metadata.encoding`、`guestinfo.userdata.encoding` 和 `guestinfo.vendordata.encoding`）可以包含：

base64

属性的内容用 `base64` 编码。

gzip+base64

属性的内容先用 `base64` 编码再用 `gzip` 压缩。

> **Note**
>
> 该`seed.iso`方法支持单独的（可选）`network-config`配置文件。VMware `guestinfo`网络配置的提供方式有所不同。其他信息将在下一节中提供。

如果需要明确的网络配置，则应将其以两种形式嵌入 `metadata` YAML 或 JSON 属性:

network

包含 JSON 或 YAML 格式的编码网络配置。

network.encoding

包含上述网络配置数据的编码。对于 `guestinfo` 数据，`cloud-init` 支持的编码是相同的：`base64` 和 `gzip+base64`。

Example 使用 VMware 用于传递配置的 vSphere **govc** CLI 工具 **guestinfo**

1. 按中所述准备`meta-data``user-data`、和可选`network-config`配置文件[NoCloud (`seed.iso`) `cloud-init` 在 KVM 上对亚马逊 Linux 2023 进行配置以及 VMware]。
2. 将配置文件转换为可用的格式 VMware `guestinfo`.

```
# 'meta-data', `user-data` and `network-config` are the configuration
# files in the same format that would be used by a NoCloud (seed.iso)
# data source, read-them and convert them to VMware guestinfo
#
# The VM_NAME variable is assumed to be set to the name of the VM
# It is assumed that the necessary govc environment (credentials etc...) are
 already set

metadata=$(cat "meta-data")
userdata=$(cat "user-data")
if [ -e "network-config" ] ; then
    # We need to embed the network config inside the meta-data
    netconf=$(base64 -w0 "network-config")
    metadata=$(printf "%s\nnetwork: %s\nnetwork.encoding: base64" "$metadata"
 "$netconf")
fi
metadata=$(base64 -w0 <<< "$metadata")
govc vm.change -vm "$VM_NAME" \
    -e guestinfo.metadata="$metadata" \
    -e guestinfo.metadata.encoding="base64"
userdata=$(base64 -w0 <<< "$userdata")
govc vm.change -vm "$VM_NAME" \
    -e guestinfo.userdata="$userdata" \
    -e guestinfo.userdata.encoding="base64"
```

将亚马逊 Linux 2023 标准 AMI 上安装的软件包与 AL2 023 KVM 镜像进行比较

将 AL2 023 标准 AMI 上的 RPMs 当前内容与 AL2 023 KVM 镜像上的 RPMs 当前图像进行比较。

软件包	AMI	KVM
`acl`	2.3.1	2.3.1

软件包	AMI	KVM
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chrony-config	4.3	
amazon-ec2-net-utils	2.5.1	
amazon-linux-onprem		1.2
amazon-linux-repo-cdn		2023.6.20241031
amazon-linux-repo-s3	2023.6.20241031	
amazon-linux-sb-keys	2023.1	2023.1
amazon-onprem-network		1.2
amazon-rpm-config	228	228
amazon-ssm-agent	3.3.987.0	3.3.987.0
amd-ucode-firmware	20210208 (noarch)	20210208 (noarch)
at	3.1.23	3.1.23
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2.15.30	2.15.30
basesystem	11	11

软件包	AMI	KVM
bash	5.2.15	5.2.15
bash-completion	2.1.1	2.1.1
bc	1.07.1	1.07.1
bind-libs	9.18.28	9.18.28
bind-license	9.18.28	9.18.28
bind-utils	9.18.28	9.18.28
binutils	2.39	2.39
boost-filesystem	1.75.0	1.75.0
boost-system	1.75.0	1.75.0
boost-thread	1.75.0	1.75.0
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023.2.68	2023.2.68
c-ares	1.19.1	
checkpolicy	3.4	3.4
chkconfig	1.15	1.15
chrony	4.3	4.3
cloud-init	22.2	22.2
cloud-init-cfg-ec2	22.2	
cloud-init-cfg-onprem		22.2

软件包	AMI	KVM
cloud-utils-growpart	0.31	0.31
coreutils	8.32	8.32
coreutils-common	8.32	8.32
cpio	2.13	2.13
cracklib	2.9.6	2.9.6
cracklib-dicts	2.9.6	2.9.6
crontabs	1.11	1.11
crypto-policies	20220428	20220428
crypto-policies-scripts	20220428	20220428
cryptsetup	2.6.1	2.6.1
cryptsetup-libs	2.6.1	2.6.1
curl-minimal	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27
cyrus-sasl-plain	2.1.27	2.1.27
dbus	1.12.28	1.12.28
dbus-broker	32	32
dbus-common	1.12.28	1.12.28
dbus-libs	1.12.28	1.12.28
device-mapper	1.02.185	1.02.185
device-mapper-libs	1.02.185	1.02.185

软件包	AMI	KVM
diffutils	3.8	3.8
dnf	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2
dnf-plugins-core	4.3.0	4.3.0
dnf-plugin-support-info	1.2	1.2
dnf-utils	4.3.0	4.3.0
dosfstools	4.2	4.2
dracut	055	055
dracut-config-ec2	3.0	
dracut-config-generic	055	055
dwz	0.14	0.14
dyninst	10.2.1	10.2.1
e2fsprogs	1.46.5	1.46.5
e2fsprogs-libs	1.46.5	1.46.5
ec2-hibinit-agent	1.0.8	
ec2-instance-connect	1.1	
ec2-instance-connect-selinux	1.1	

软件包	AMI	KVM
ec2-utils	2.2.0	
ed	1.14.2	1.14.2
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38
efivar-libs	38	38
elfutils-debuginfod-client	0.188	0.188
elfutils-default-yama-scope	0.188	0.188
elfutils-libelf	0.188	0.188
elfutils-libs	0.188	0.188
ethtool	5.15	5.15
expat	2.5.0	2.5.0
file	5.39	5.39
file-libs	5.39	5.39
filesystem	3.14	3.14
findutils	4.8.0	4.8.0
fonts-srpm-macros	2.0.5	2.0.5
fstrm	0.6.1	0.6.1
fuse-libs	2.9.9	2.9.9

软件包	AMI	KVM
gawk	5.1.0	5.1.0
gdbm-libs	1.19	1.19
gdisk	1.0.8	1.0.8
gettext	0.21	0.21
gettext-libs	0.21	0.21
ghc-srpm-macros	1.5.0	1.5.0
glib2	2.74.7	2.74.7
glibc	2.34	2.34
glibc-all-langpacks	2.34	2.34
glibc-common	2.34	2.34
glibc-gconv-extra	2.34	2.34
glibc-locale-source	2.34	2.34
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gnutls	3.8.0	3.8.0
go-srpm-macros	3.2.0	3.2.0
gpgme	1.15.1	1.15.1
gpm-libs	1.20.7	1.20.7
grep	3.8	3.8
groff-base	1.22.4	1.22.4

软件包	AMI	KVM
grub2-common	2.06	2.06
grub2-efi-aa64-ec2	2.06 (aarch64)	2.06 (aarch64)
grub2-efi-x64-ec2	2.06 (x86_64)	2.06 (x86_64)
grub2-pc		2.06 (x86_64)
grub2-pc-modules	2.06	2.06 (noarch)
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.40	8.40
gssproxy	0.8.4	0.8.4
gzip	1.12	1.12
hostname	3.23	3.23
hunspell	1.7.0	1.7.0
hunspell-en	0.20140811.1	0.20140811.1
hunspell-en-GB	0.20140811.1	0.20140811.1
hunspell-en-US	0.20140811.1	0.20140811.1
hunspell-filesystem	1.7.0	1.7.0
hwdata	0.384	0.384
info	6.7	6.7
inih	49	49
initscripts	10.09	10.09

软件包	AMI	KVM
iproute	6.10.0	6.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jemalloc	5.2.1	5.2.1
jitterentropy	3.4.1	3.4.1
jq	1.7.1	
json-c	0.14	0.14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0
kernel	6.1.112	6.1.112
kernel-libbpf	6.1.112	6.1.112
kernel-livepatch-repo-cdn		2023.6.20241031
kernel-livepatch-repo-s3	2023.6.20241031	
kernel-modules-extra		6.1.112
kernel-modules-extra-common		6.1.112
kernel-srpm-macros	1.0	1.0
kernel-tools	6.1.112	6.1.112

软件包	AMI	KVM
keyutils	1.6.3	1.6.3
keyutils-libs	1.6.3	1.6.3
kmod	29	29
kmod-libs	29	29
kpatch-runtime	0.9.7	0.9.7
krb5-libs	1.21.3	1.21.3
less	608	608
libacl	2.3.1	2.3.1
libaio	0.3.111	0.3.111
libarchive	3.7.4	3.7.4
libargon2	20171227	20171227
libassuan	2.5.5	2.5.5
libattr	2.5.1	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2.37.4	2.37.4
libcap	2.48	2.48
libcap-ng	0.8.2	0.8.2
libcbor	0.7.0	0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1.46.5	1.46.5

软件包	AMI	KVM
libcomps	0.1.20	0.1.20
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0.69.0	0.69.0
libeconf	0.4.0	0.4.0
libedit	3.1	3.1
libev	4.33	4.33
libevent	2.1.12	2.1.12
libfdisk	2.37.4	2.37.4
libffi	3.4.4	3.4.4
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1
libgpg-error	1.42	1.42
libibverbs	48.0	48.0
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1

软件包	AMI	KVM
libkcapi	1.4.0	1.4.0
libkcapi-hmaccalc	1.4.0	1.4.0
libldb	2.6.2	
libmaxminddb	1.5.2	1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.4	1.0.4
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libnfsidmap	2.5.4	2.5.4
libnghttp2	1.59.0	1.59.0
libnl3	3.5.0	3.5.0
libpath_utils	0.2.1	0.2.1
libpcap	1.10.1	1.10.1
libpipeline	1.5.3	1.5.3
libpkgconf	1.8.0	1.8.0
libpsl	0.21.1	0.21.1
libpwquality	1.4.4	1.4.4
libref_array	0.1.5	0.1.5
librepo	1.14.5	1.14.5
libreport-filesystem	2.15.2	2.15.2

软件包	AMI	KVM
libseccomp	2.5.3	2.5.3
libselinux	3.4	3.4
libselinux-utils	3.4	3.4
libsemanage	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2.13	2.13
libsmartcols	2.37.4	2.37.4
libsolv	0.7.22	0.7.22
libss	1.46.5	1.46.5
libsss_certmap	2.9.4	
libsss_idmap	2.9.4	2.9.4
libsss_nss_idmap	2.9.4	2.9.4
libsss_sudo	2.9.4	
libstdc++	11.4.1	11.4.1
libstoragemgmt	1.9.4	1.9.4
libtalloc	2.3.4	
libtasn1	4.19.0	4.19.0
libtdb	1.4.7	
libtevent	0.13.0	
libtextstyle	0.21	0.21

软件包	AMI	KVM
libtirpc	1.3.3	1.3.3
libunistring	0.9.10	0.9.10
libuser	0.63	0.63
libutempter	1.2.1	1.2.1
libuuid	2.37.4	2.37.4
libuv	1.47.0	1.47.0
libverto	0.3.2	0.3.2
libverto-libev	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33
libxml2	2.10.4	2.10.4
libyaml	0.2.5	0.2.5
libzstd	1.5.5	1.5.5
linux-firmware-whence	20210208 (noarch)	20210208 (noarch)
lm_sensors-libs	3.6.0	3.6.0
lmdb-libs	0.9.29	0.9.29
logrotate	3.20.1	3.20.1
lsof	4.94.0	4.94.0
lua-libs	5.4.4	5.4.4
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4

软件包	AMI	KVM
man-db	2.9.3	2.9.3
man-pages	5.10	5.10
microcode_ctl	2.1 (x86_64)	2.1 (x86_64)
mpfr	4.1.0	4.1.0
nano	5.8	5.8
ncurses	6.2	6.2
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
nettle	3.8	3.8
net-tools	2.0	2.0
newt	0.52.21	0.52.21
nfs-utils	2.5.4	2.5.4
npth	1.6	1.6
nspr	4.35.0	4.35.0
nss	3.90.0	3.90.0
nss-softokn	3.90.0	3.90.0
nss-softokn-freebl	3.90.0	3.90.0
nss-sysinit	3.90.0	3.90.0
nss-util	3.90.0	3.90.0
ntsysv	1.15	1.15

软件包	AMI	KVM
numactl-libs	2.0.14	2.0.14
ocaml-srpm-macros	6	6
oniguruma	6.9.7.1	
openblas-srpm-macros	2	2
openldap	2.4.57	2.4.57
openssh	8.7p1	8.7p1
openssh-clients	8.7p1	8.7p1
openssh-server	8.7p1	8.7p1
openssl	3.0.8	3.0.8
openssl-libs	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12
os-prober	1.77	1.77
p11-kit	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1
package-notes-srpm-macros	0.4	0.4
pam	1.5.1	1.5.1
parted	3.4	3.4
passwd	0.80	0.80
pciutils	3.7.0	3.7.0
pciutils-libs	3.7.0	3.7.0

软件包	AMI	KVM
pcre2	10.40	10.40
pcre2-syntax	10.40	10.40
perl-Carp	1.50	1.50
perl-Class-Struct	0.66	0.66
perl-constant	1.33	1.33
perl-DynaLoader	1.47	1.47
perl-Encode	3.15	3.15
perl-Errno	1.30	1.30
perl-Exporter	5.74	5.74
perl-Fcntl	1.13	1.13
perl-File-Basename	2.85	2.85
perl-File-Path	2.18	2.18
perl-File-stat	1.09	1.09
perl-File-Temp	0.231.100	0.231.100
perl-Getopt-Long	2.52	2.52
perl-Getopt-Std	1.12	1.12
perl-HTTP-Tiny	0.078	0.078
perl-if	0.60.800	0.60.800
perl-interpreter	5.32.1	5.32.1
perl-IO	1.43	1.43

软件包	AMI	KVM
perl-IPC-Open3	1.21	1.21
perl-libs	5.32.1	5.32.1
perl-MIME-Base64	3.16	3.16
perl-mro	1.23	1.23
perl-overload	1.31	1.31
perl-overloading	0.02	0.02
perl-parent	0.238	0.238
perl-PathTools	3.78	3.78
perl-Pod-Escapes	1.07	1.07
perl-podlators	4.14	4.14
perl-Pod-Perldoc	3.28.01	3.28.01
perl-Pod-Simple	3.42	3.42
perl-Pod-Usage	2.01	2.01
perl-POSIX	1.94	1.94
perl-Scalar-List-Utils	1.56	1.56
perl-SelectSaver	1.02	1.02
perl-Socket	2.032	2.032
perl-srpm-macros	1	1
perl-Storable	3.21	3.21
perl-subs	1.03	1.03

软件包	AMI	KVM
perl-Symbol	1.08	1.08
perl-Term-ANSIColor	5.01	5.01
perl-Term-Cap	1.17	1.17
perl-Text-ParseWords	3.30	3.30
perl-Text-Tabs+Wrap	2021.0726	2021.0726
perl-Time-Local	1.300	1.300
perl-vars	1.05	1.05
pkgconf	1.8.0	1.8.0
pkgconf-m4	1.8.0	1.8.0
pkgconf-pkg-config	1.8.0	1.8.0
policycoreutils	3.4	3.4
policycoreutils-python-utils	3.4	
popt	1.18	1.18
procps-ng	3.3.17	3.3.17
protobuf-c	1.4.1	1.4.1
psacct	6.6.4	6.6.4
psmisc	23.4	23.4
publicsuffix-list-dafsa	20240212	20240212
python3	3.9.16	3.9.16

软件包	AMI	KVM
python3-attrs	20.3.0	20.3.0
python3-audit	3.0.6	3.0.6
python3-awscrt	0.19.19	0.19.19
python3-babel	2.9.1	2.9.1
python3-cffi	1.14.5	1.14.5
python3-chardet	4.0.0	4.0.0
python3-colorama	0.4.4	0.4.4
python3-configobj	5.0.6	5.0.6
python3-cryptography	36.0.1	36.0.1
python3-daemon	2.3.0	
python3-dateutil	2.8.1	2.8.1
python3-dbus	1.2.18	1.2.18
python3-distro	1.5.0	1.5.0
python3-dnf	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0
python3-docutils	0.16	0.16
python3-gpg	1.15.1	1.15.1
python3-hawkey	0.69.0	0.69.0
python3-idna	2.10	2.10
python3-jinja2	2.11.3	2.11.3

软件包	AMI	KVM
python3-jmespath	0.10.0	0.10.0
python3-jsonpatch	1.21	1.21
python3-jsonpointer	2.0	2.0
python3-jsonschema	3.2.0	3.2.0
python3-libcomps	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0
python3-libs	3.9.16	3.9.16
python3-libselinux	3.4	3.4
python3-libsemanage	3.4	3.4
python3-libstorage mgmt	1.9.4	1.9.4
python3-lockfile	0.12.2	
python3-markupsafe	1.1.1	1.1.1
python3-netifaces	0.10.6	0.10.6
python3-oauthlib	3.0.2	3.0.2
python3-pip-wheel	21.3.1	21.3.1
python3-ply	3.11	3.11
python3-policycore utils	3.4	3.4
python3-prettytable	0.7.2	0.7.2

软件包	AMI	KVM
python3-prompt-toolkit	3.0.24	3.0.24
python3-pycparser	2.20	2.20
python3-pyrsistent	0.17.3	0.17.3
python3-pyserial	3.4	3.4
python3-pysocks	1.7.1	1.7.1
python3-pytz	2022.7.1	2022.7.1
python3-pyyaml	5.4.1	5.4.1
python3-requests	2.25.1	2.25.1
python3-rpm	4.16.1.3	4.16.1.3
python3-ruamel-yaml	0.16.6	0.16.6
python3-ruamel-yaml-clib	0.1.2	0.1.2
python3-setools	4.4.1	4.4.1
python3-setuptools	59.6.0	59.6.0
python3-setuptools-wheel	59.6.0	59.6.0
python3-six	1.15.0	1.15.0
python3-systemd	235	235
python3-urllib3	1.25.10	1.25.10
python3-wcwidth	0.2.5	0.2.5

软件包	AMI	KVM
python-chevron	0.13.1	
python-srpm-macros	3.9	3.9
quota	4.06	4.06
quota-nls	4.06	4.06
readline	8.1	8.1
rng-tools	6.14	6.14
rootfiles	8.1	8.1
rpcbind	1.2.6	1.2.6
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	4.16.1.3
rsync	3.2.6	3.2.6
rust-srpm-macros	21	21
sbsigntools	0.9.4	0.9.4
screen	4.8.0	4.8.0
sed	4.8	4.8
selinux-policy	38.1.45	38.1.45

软件包	AMI	KVM
selinux-policy-targeted	38.1.45	38.1.45
setup	2.13.7	2.13.7
shadow-utils	4.9	4.9
slang	2.3.2	2.3.2
sqlite-libs	3.40.0	3.40.0
sssd-client	2.9.4	2.9.4
sssd-common	2.9.4	
sssd-kcm	2.9.4	
sssd-nfs-idmap	2.9.4	
strace	6.8	6.8
sudo	1.9.15	1.9.15
sysctl-defaults	1.0	1.0
sysstat	12.5.6	12.5.6
systemd	252.23	252.23
systemd-libs	252.23	252.23
systemd-networkd	252.23	252.23
systemd-pam	252.23	252.23
systemd-resolved	252.23	252.23
systemd-udev	252.23	252.23
system-release	2023.6.20241031	2023.6.20241031

软件包	AMI	KVM
systemtap-runtime	4.8	4.8
tar	1.34	1.34
tbb	2020.3	2020.3
tcpdump	4.99.1	4.99.1
tcsh	6.24.07	6.24.07
time	1.9	1.9
traceroute	2.1.3	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	2.2	2.2
userspace-rcu	0.12.1	0.12.1
util-linux	2.37.4	2.37.4
util-linux-core	2.37.4	2.37.4
vim-common	9.0.2153	9.0.2153
vim-data	9.0.2153	9.0.2153
vim-enhanced	9.0.2153	9.0.2153
vim-filesystem	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153
wget	1.21.3	1.21.3
which	2.21	2.21

软件包	AMI	KVM
words	3.0	3.0
xfsdump	3.1.11	3.1.11
xfsprogs	5.18.0	5.18.0
xxd	9.0.2153	9.0.2153
xxhash-libs	0.8.0	0.8.0
xz	5.2.5	5.2.5
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0
zlib	1.2.11	1.2.11
zram-generator	1.1.2	
zram-generator-defaults	1.1.2	
zstd	1.5.5	1.5.5

将亚马逊 Linux 2023 标准 AMI 上安装的软件包与 AL2 023 VMware OVA 镜像进行比较

将 AL2 023 标准 AMI 上的 RPMs 当前图像与 AL2 023 VMware OVA 图像上的 RPMs 当前图像进行比较。

软件包	AMI	VMware OVA
acl	2.3.1	2.3.1

软件包	AMI	VMware OVA
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chrony-config	4.3	
amazon-ec2-net-utils	2.5.1	
amazon-linux-onprem		1.2
amazon-linux-repo-cdn		2023.6.20241031
amazon-linux-repo-s3	2023.6.20241031	
amazon-linux-sb-keys	2023.1	2023.1
amazon-onprem-network		1.2
amazon-rpm-config	228	228
amazon-ssm-agent	3.3.987.0	3.3.987.0
amd-ucode-firmware	20210208	20210208
at	3.1.23	3.1.23
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2.15.30	2.15.30
basesystem	11	11

软件包	AMI	VMware OVA
bash	5.2.15	5.2.15
bash-completion	2.1.1	2.1.1
bc	1.07.1	1.07.1
bind-libs	9.18.28	9.18.28
bind-license	9.18.28	9.18.28
bind-utils	9.18.28	9.18.28
binutils	2.39	2.39
boost-filesystem	1.75.0	1.75.0
boost-system	1.75.0	1.75.0
boost-thread	1.75.0	1.75.0
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023.2.68	2023.2.68
c-ares	1.19.1	
checkpolicy	3.4	3.4
chkconfig	1.15	1.15
chrony	4.3	4.3
cloud-init	22.2	22.2
cloud-init-cfg-ec2	22.2	
cloud-init-cfg-onprem		22.2

软件包	AMI	VMware OVA
cloud-utils-growpart	0.31	0.31
coreutils	8.32	8.32
coreutils-common	8.32	8.32
cpio	2.13	2.13
cracklib	2.9.6	2.9.6
cracklib-dicts	2.9.6	2.9.6
crontabs	1.11	1.11
crypto-policies	20220428	20220428
crypto-policies-scripts	20220428	20220428
cryptsetup	2.6.1	2.6.1
cryptsetup-libs	2.6.1	2.6.1
curl-minimal	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27
cyrus-sasl-plain	2.1.27	2.1.27
dbus	1.12.28	1.12.28
dbus-broker	32	32
dbus-common	1.12.28	1.12.28
dbus-libs	1.12.28	1.12.28
device-mapper	1.02.185	1.02.185
device-mapper-libs	1.02.185	1.02.185

软件包	AMI	VMware OVA
diffutils	3.8	3.8
dnf	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2
dnf-plugins-core	4.3.0	4.3.0
dnf-plugin-support-info	1.2	1.2
dnf-utils	4.3.0	4.3.0
dosfstools	4.2	4.2
dracut	055	055
dracut-config-ec2	3.0	
dracut-config-generic	055	055
dwz	0.14	0.14
dyninst	10.2.1	10.2.1
e2fsprogs	1.46.5	1.46.5
e2fsprogs-libs	1.46.5	1.46.5
ec2-hibinit-agent	1.0.8	
ec2-instance-connect	1.1	
ec2-instance-connect-selinux	1.1	

软件包	AMI	VMware OVA
ec2-utils	2.2.0	
ed	1.14.2	1.14.2
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38
efivar-libs	38	38
elfutils-debuginfod-client	0.188	0.188
elfutils-default-yama-scope	0.188	0.188
elfutils-libelf	0.188	0.188
elfutils-libs	0.188	0.188
ethtool	5.15	5.15
expat	2.5.0	2.5.0
file	5.39	5.39
file-libs	5.39	5.39
filesystem	3.14	3.14
findutils	4.8.0	4.8.0
fonts-srpm-macros	2.0.5	2.0.5
fstrm	0.6.1	0.6.1
fuse3		3.10.4

软件包	AMI	VMware OVA
fuse3-libs		3.10.4
fuse-common		3.10.4
fuse-libs	2.9.9	2.9.9
gawk	5.1.0	5.1.0
gdbm-libs	1.19	1.19
gdisk	1.0.8	1.0.8
gettext	0.21	0.21
gettext-libs	0.21	0.21
ghc-srpm-macros	1.5.0	1.5.0
glib2	2.74.7	2.74.7
glibc	2.34	2.34
glibc-all-langpacks	2.34	2.34
glibc-common	2.34	2.34
glibc-gconv-extra	2.34	2.34
glibc-locale-source	2.34	2.34
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gnutls	3.8.0	3.8.0
go-srpm-macros	3.2.0	3.2.0
gpgme	1.15.1	1.15.1

软件包	AMI	VMware OVA
gpm-libs	1.20.7	1.20.7
grep	3.8	3.8
groff-base	1.22.4	1.22.4
grub2-common	2.06	2.06
grub2-efi-x64-ec2	2.06	2.06
grub2-pc		2.06
grub2-pc-modules	2.06	2.06
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.40	8.40
gssproxy	0.8.4	0.8.4
gzip	1.12	1.12
hostname	3.23	3.23
hunspell	1.7.0	1.7.0
hunspell-en	0.20140811.1	0.20140811.1
hunspell-en-GB	0.20140811.1	0.20140811.1
hunspell-en-US	0.20140811.1	0.20140811.1
hunspell-filesystem	1.7.0	1.7.0
hwdata	0.384	0.384
info	6.7	6.7

软件包	AMI	VMware OVA
inih	49	49
initscripts	10.09	10.09
iproute	6.10.0	6.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jemalloc	5.2.1	5.2.1
jitterentropy	3.4.1	3.4.1
jq	1.7.1	
json-c	0.14	0.14
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0
kernel	6.1.112	6.1.112
kernel-libbpf	6.1.112	6.1.112
kernel-livepatch-repo-cdn		2023.6.20241031
kernel-livepatch-repo-s3	2023.6.20241031	
kernel-modules-extra		6.1.112
kernel-modules-extra-common		6.1.112

软件包	AMI	VMware OVA
kernel-srpm-macros	1.0	1.0
kernel-tools	6.1.112	6.1.112
keyutils	1.6.3	1.6.3
keyutils-libs	1.6.3	1.6.3
kmod	29	29
kmod-libs	29	29
kpatch-runtime	0.9.7	0.9.7
krb5-libs	1.21.3	1.21.3
less	608	608
libacl	2.3.1	2.3.1
libaio	0.3.111	0.3.111
libarchive	3.7.4	3.7.4
libargon2	20171227	20171227
libassuan	2.5.5	2.5.5
libattr	2.5.1	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2.37.4	2.37.4
libcap	2.48	2.48
libcap-ng	0.8.2	0.8.2
libcbor	0.7.0	0.7.0

软件包	AMI	VMware OVA
libcollection	0.7.0	0.7.0
libcom_err	1.46.5	1.46.5
libcomps	0.1.20	0.1.20
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0.69.0	0.69.0
libeconf	0.4.0	0.4.0
libedit	3.1	3.1
libev	4.33	4.33
libevent	2.1.12	2.1.12
libfdisk	2.37.4	2.37.4
libffi	3.4.4	3.4.4
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1
libgpg-error	1.42	1.42
libibverbs	48.0	48.0

软件包	AMI	VMware OVA
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1
libkcapi	1.4.0	1.4.0
libkcapi-hmaccalc	1.4.0	1.4.0
libldb	2.6.2	
libmaxminddb	1.5.2	1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.4	1.0.4
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libmspack		0.10.1
libnfsidmap	2.5.4	2.5.4
libnghttp2	1.59.0	1.59.0
libnl3	3.5.0	3.5.0
libpath_utils	0.2.1	0.2.1
libpcap	1.10.1	1.10.1
libpipeline	1.5.3	1.5.3
libpkgconf	1.8.0	1.8.0
libpsl	0.21.1	0.21.1
libpwquality	1.4.4	1.4.4

软件包	AMI	VMware OVA
libref_array	0.1.5	0.1.5
librepo	1.14.5	1.14.5
libreport-filesystem	2.15.2	2.15.2
libseccomp	2.5.3	2.5.3
libselinux	3.4	3.4
libselinux-utils	3.4	3.4
libsemanage	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2.13	2.13
libsmartcols	2.37.4	2.37.4
libsolv	0.7.22	0.7.22
libss	1.46.5	1.46.5
libsss_certmap	2.9.4	
libsss_idmap	2.9.4	2.9.4
libsss_nss_idmap	2.9.4	2.9.4
libsss_sudo	2.9.4	
libstdc++	11.4.1	11.4.1
libstoragemgmt	1.9.4	1.9.4
libtalloc	2.3.4	
libtasn1	4.19.0	4.19.0

软件包	AMI	VMware OVA
libtdb	1.4.7	
libtevent	0.13.0	
libtextstyle	0.21	0.21
libtirpc	1.3.3	1.3.3
libtool-ltdl		2.4.7
libunistring	0.9.10	0.9.10
libuser	0.63	0.63
libutempter	1.2.1	1.2.1
libuuid	2.37.4	2.37.4
libuv	1.47.0	1.47.0
libverto	0.3.2	0.3.2
libverto-libev	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33
libxml2	2.10.4	2.10.4
libxslt		1.1.34
libyaml	0.2.5	0.2.5
libzstd	1.5.5	1.5.5
linux-firmware-whence	20210208	20210208
lm_sensors-libs	3.6.0	3.6.0
lmdb-libs	0.9.29	0.9.29

软件包	AMI	VMware OVA
logrotate	3.20.1	3.20.1
lsof	4.94.0	4.94.0
lua-libs	5.4.4	5.4.4
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4
man-db	2.9.3	2.9.3
man-pages	5.10	5.10
microcode_ctl	2.1	2.1
mpfr	4.1.0	4.1.0
nano	5.8	5.8
ncurses	6.2	6.2
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
nettle	3.8	3.8
net-tools	2.0	2.0
newt	0.52.21	0.52.21
nfs-utils	2.5.4	2.5.4
npth	1.6	1.6
nspr	4.35.0	4.35.0
nss	3.90.0	3.90.0

软件包	AMI	VMware OVA
nss-softokn	3.90.0	3.90.0
nss-softokn-freebl	3.90.0	3.90.0
nss-sysinit	3.90.0	3.90.0
nss-util	3.90.0	3.90.0
ntsysv	1.15	1.15
numactl-libs	2.0.14	2.0.14
ocaml-srpm-macros	6	6
oniguruma	6.9.7.1	
openblas-srpm-macros	2	2
openldap	2.4.57	2.4.57
openssh	8.7p1	8.7p1
openssh-clients	8.7p1	8.7p1
openssh-server	8.7p1	8.7p1
openssl	3.0.8	3.0.8
openssl-libs	3.0.8	3.0.8
openssl-pkcs11	0.4.12	0.4.12
open-vm-tools		12.3.0
os-prober	1.77	1.77
p11-kit	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1

软件包	AMI	VMware OVA
package-notes-srpm-macros	0.4	0.4
pam	1.5.1	1.5.1
parted	3.4	3.4
passwd	0.80	0.80
pciutils	3.7.0	3.7.0
pciutils-libs	3.7.0	3.7.0
pcre2	10.40	10.40
pcre2-syntax	10.40	10.40
perl-Carp	1.50	1.50
perl-Class-Struct	0.66	0.66
perl-constant	1.33	1.33
perl-DynaLoader	1.47	1.47
perl-Encode	3.15	3.15
perl-Errno	1.30	1.30
perl-Exporter	5.74	5.74
perl-Fcntl	1.13	1.13
perl-File-Basename	2.85	2.85
perl-File-Path	2.18	2.18
perl-File-stat	1.09	1.09
perl-File-Temp	0.231.100	0.231.100

软件包	AMI	VMware OVA
perl-Getopt-Long	2.52	2.52
perl-Getopt-Std	1.12	1.12
perl-HTTP-Tiny	0.078	0.078
perl-if	0.60.800	0.60.800
perl-interpreter	5.32.1	5.32.1
perl-IO	1.43	1.43
perl-IPC-Open3	1.21	1.21
perl-libs	5.32.1	5.32.1
perl-MIME-Base64	3.16	3.16
perl-mro	1.23	1.23
perl-overload	1.31	1.31
perl-overloading	0.02	0.02
perl-parent	0.238	0.238
perl-PathTools	3.78	3.78
perl-Pod-Escapes	1.07	1.07
perl-podlators	4.14	4.14
perl-Pod-Perldoc	3.28.01	3.28.01
perl-Pod-Simple	3.42	3.42
perl-Pod-Usage	2.01	2.01
perl-POSIX	1.94	1.94

软件包	AMI	VMware OVA
perl-Scalar-List-Utils	1.56	1.56
perl-SelectSaver	1.02	1.02
perl-Socket	2.032	2.032
perl-srpm-macros	1	1
perl-Storable	3.21	3.21
perl-subs	1.03	1.03
perl-Symbol	1.08	1.08
perl-Term-ANSIColor	5.01	5.01
perl-Term-Cap	1.17	1.17
perl-Text-ParseWords	3.30	3.30
perl-Text-Tabs+Wrap	2021.0726	2021.0726
perl-Time-Local	1.300	1.300
perl-vars	1.05	1.05
pkgconf	1.8.0	1.8.0
pkgconf-m4	1.8.0	1.8.0
pkgconf-pkg-config	1.8.0	1.8.0
policycoreutils	3.4	3.4
policycoreutils-python-utils	3.4	
popt	1.18	1.18

软件包	AMI	VMware OVA
procps-ng	3.3.17	3.3.17
protobuf-c	1.4.1	1.4.1
psacct	6.6.4	6.6.4
psmisc	23.4	23.4
publicsuffix-list-dafsa	20240212	20240212
python3	3.9.16	3.9.16
python3-attrs	20.3.0	20.3.0
python3-audit	3.0.6	3.0.6
python3-awscrt	0.19.19	0.19.19
python3-babel	2.9.1	2.9.1
python3-cffi	1.14.5	1.14.5
python3-chardet	4.0.0	4.0.0
python3-colorama	0.4.4	0.4.4
python3-configobj	5.0.6	5.0.6
python3-cryptography	36.0.1	36.0.1
python3-daemon	2.3.0	
python3-dateutil	2.8.1	2.8.1
python3-dbus	1.2.18	1.2.18
python3-distro	1.5.0	1.5.0
python3-dnf	4.14.0	4.14.0

软件包	AMI	VMware OVA
python3-dnf-plugins-core	4.3.0	4.3.0
python3-docutils	0.16	0.16
python3-gpg	1.15.1	1.15.1
python3-hawkey	0.69.0	0.69.0
python3-idna	2.10	2.10
python3-jinja2	2.11.3	2.11.3
python3-jmespath	0.10.0	0.10.0
python3-jsonpatch	1.21	1.21
python3-jsonpointer	2.0	2.0
python3-jsonschema	3.2.0	3.2.0
python3-libcomps	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0
python3-libs	3.9.16	3.9.16
python3-libselinux	3.4	3.4
python3-libsemanage	3.4	3.4
python3-libstorage mgmt	1.9.4	1.9.4
python3-lockfile	0.12.2	
python3-markupsafe	1.1.1	1.1.1
python3-netifaces	0.10.6	0.10.6

软件包	AMI	VMware OVA
python3-oauthlib	3.0.2	3.0.2
python3-pip-wheel	21.3.1	21.3.1
python3-ply	3.11	3.11
python3-policycoreutils	3.4	3.4
python3-prettytable	0.7.2	0.7.2
python3-prompt-toolkit	3.0.24	3.0.24
python3-pycparser	2.20	2.20
python3-pyrsistent	0.17.3	0.17.3
python3-pyserial	3.4	3.4
python3-pysocks	1.7.1	1.7.1
python3-pytz	2022.7.1	2022.7.1
python3-pyyaml	5.4.1	5.4.1
python3-requests	2.25.1	2.25.1
python3-rpm	4.16.1.3	4.16.1.3
python3-ruamel-yaml	0.16.6	0.16.6
python3-ruamel-yaml-clib	0.1.2	0.1.2
python3-setools	4.4.1	4.4.1
python3-setuptools	59.6.0	59.6.0

软件包	AMI	VMware OVA
python3-setuptools-wheel	59.6.0	59.6.0
python3-six	1.15.0	1.15.0
python3-systemd	235	235
python3-urllib3	1.25.10	1.25.10
python3-wcwidth	0.2.5	0.2.5
python-chevron	0.13.1	
python-srpm-macros	3.9	3.9
quota	4.06	4.06
quota-nls	4.06	4.06
readline	8.1	8.1
rng-tools	6.14	6.14
rootfiles	8.1	8.1
rpcbind	1.2.6	1.2.6
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3
rpm-libs	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	4.16.1.3

软件包	AMI	VMware OVA
rsync	3.2.6	3.2.6
rust-srpm-macros	21	21
sbsigntools	0.9.4	0.9.4
screen	4.8.0	4.8.0
sed	4.8	4.8
selinux-policy	38.1.45	38.1.45
selinux-policy-targeted	38.1.45	38.1.45
setup	2.13.7	2.13.7
shadow-utils	4.9	4.9
slang	2.3.2	2.3.2
sqlite-libs	3.40.0	3.40.0
sssd-client	2.9.4	2.9.4
sssd-common	2.9.4	
sssd-kcm	2.9.4	
sssd-nfs-idmap	2.9.4	
strace	6.8	6.8
sudo	1.9.15	1.9.15
sysctl-defaults	1.0	1.0
sysstat	12.5.6	12.5.6
systemd	252.23	252.23

软件包	AMI	VMware OVA
systemd-libs	252.23	252.23
systemd-networkd	252.23	252.23
systemd-pam	252.23	252.23
systemd-resolved	252.23	252.23
systemd-udev	252.23	252.23
system-release	2023.6.20241031	2023.6.20241031
systemtap-runtime	4.8	4.8
tar	1.34	1.34
tbb	2020.3	2020.3
tcpdump	4.99.1	4.99.1
tcsh	6.24.07	6.24.07
time	1.9	1.9
traceroute	2.1.3	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0
update-motd	2.2	2.2
userspace-rcu	0.12.1	0.12.1
util-linux	2.37.4	2.37.4
util-linux-core	2.37.4	2.37.4
vim-common	9.0.2153	9.0.2153

软件包	AMI	VMware OVA
vim-data	9.0.2153	9.0.2153
vim-enhanced	9.0.2153	9.0.2153
vim-filesystem	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153
wget	1.21.3	1.21.3
which	2.21	2.21
words	3.0	3.0
xfsdump	3.1.11	3.1.11
xfsprogs	5.18.0	5.18.0
xmlsec1		1.2.33
xmlsec1-openssl		1.2.33
xxd	9.0.2153	9.0.2153
xxhash-libs	0.8.0	0.8.0
xz	5.2.5	5.2.5
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0
zlib	1.2.11	1.2.11
zram-generator	1.1.2	
zram-generator-defaults	1.1.2	

软件包	AMI	VMware OVA
zstd	1.5.5	1.5.5

将亚马逊 Linux 2023 标准 AMI 上安装的软件包与 AL2 023 Hyper-V 镜像进行比较

将 AL2 023 标准 AMI 上的 RPMs 当前图像与 AL2 023 Hyper-V RPMs 图像上的当前图像进行比较。

软件包	AMI	Hyper-V VHDX
acl	2.3.1	2.3.1
acpid	2.0.32	
alternatives	1.15	1.15
amazon-chrony-config	4.3	
amazon-ec2-net-utils	2.4.1	
amazon-linux-onprem		1.2
amazon-linux-repo-cdn		2023.4.20240319
amazon-linux-repo-s3	2023.4.20240319	
amazon-linux-sb-keys	2023.1	2023.1
amazon-onprem-network		1.2
amazon-rpm-config	228	228
amazon-ssm-agent	3.2.2303.0	3.2.2303.0
at	3.1.23	3.1.23

软件包	AMI	Hyper-V VHDX
attr	2.5.1	2.5.1
audit	3.0.6	3.0.6
audit-libs	3.0.6	3.0.6
aws-cfn-bootstrap	2.0	
awscli-2	2.14.5	2.14.5
basesystem	11	11
bash	5.2.15	5.2.15
bash-completion	2.1.1	2.1.1
bc	1.07.1	1.07.1
bind-libs	9.16.48	9.16.48
bind-license	9.16.48	9.16.48
bind-utils	9.16.48	9.16.48
binutils	2.39	2.39
boost-filesystem	1.75.0	1.75.0
boost-system	1.75.0	1.75.0
boost-thread	1.75.0	1.75.0
bzip2	1.0.8	1.0.8
bzip2-libs	1.0.8	1.0.8
ca-certificates	2023.2.64	2023.2.64
c-ares	1.19.0	

软件包	AMI	Hyper-V VHDX
checkpolicy	3.4	3.4
chkconfig	1.15	1.15
chrony	4.3	4.3
cloud-init	22.2	22.2
cloud-init-cfg-ec2	22.2	
cloud-init-cfg-onprem		22.2
cloud-utils-growpart	0.31	0.31
coreutils	8.32	8.32
coreutils-common	8.32	8.32
cpio	2.13	2.13
cracklib	2.9.6	2.9.6
cracklib-dicts	2.9.6	2.9.6
crontabs	1.11	1.11
crypto-policies	20220428	20220428
crypto-policies-scripts	20220428	20220428
cryptsetup	2.6.1	2.6.1
cryptsetup-libs	2.6.1	2.6.1
curl-minimal	8.5.0	8.5.0
cyrus-sasl-lib	2.1.27	2.1.27

软件包	AMI	Hyper-V VHDX
cyrus-sasl-plain	2.1.27	2.1.27
dbus	1.12.28	1.12.28
dbus-broker	32	32
dbus-common	1.12.28	1.12.28
dbus-libs	1.12.28	1.12.28
device-mapper	1.02.185	1.02.185
device-mapper-libs	1.02.185	1.02.185
diffutils	3.8	3.8
dnf	4.14.0	4.14.0
dnf-data	4.14.0	4.14.0
dnf-plugin-release-notification	1.2	1.2
dnf-plugins-core	4.3.0	4.3.0
dnf-plugin-support-info	1.2	1.2
dnf-utils	4.3.0	4.3.0
dosfstools	4.2	4.2
dracut	055	055
dracut-config-ec2	3.0	
dracut-config-generic	055	055

软件包	AMI	Hyper-V VHDX
dwz	0.14	0.14
dyninst	10.2.1	10.2.1
e2fsprogs	1.46.5	1.46.5
e2fsprogs-libs	1.46.5	1.46.5
ec2-hibinit-agent	1.0.8	
ec2-instance-connect	1.1	
ec2-instance-connect-selinux	1.1	
ec2-utils	2.2.0	
ed	1.14.2	1.14.2
efi-filesystem	5	5
efi-srpm-macros	5	5
efivar	38	38
efivar-libs	38	38
elfutils-debuginfod-client	0.188	0.188
elfutils-default-yama-scope	0.188	0.188
elfutils-libelf	0.188	0.188
elfutils-libs	0.188	0.188
ethtool	5.15	5.15

软件包	AMI	Hyper-V VHDX
expat	2.5.0	2.5.0
file	5.39	5.39
file-libs	5.39	5.39
filesystem	3.14	3.14
findutils	4.8.0	4.8.0
fonts-srpm-macros	2.0.5	2.0.5
fstrm	0.6.1	0.6.1
fuse-libs	2.9.9	2.9.9
gawk	5.1.0	5.1.0
gdbm-libs	1.19	1.19
gdisk	1.0.8	1.0.8
gettext	0.21	0.21
gettext-libs	0.21	0.21
ghc-srpm-macros	1.5.0	1.5.0
glib2	2.74.7	2.74.7
glibc	2.34	2.34
glibc-all-langpacks	2.34	2.34
glibc-common	2.34	2.34
glibc-gconv-extra	2.34	2.34
glibc-locale-source	2.34	2.34

软件包	AMI	Hyper-V VHDX
gmp	6.2.1	6.2.1
gnupg2-minimal	2.3.7	2.3.7
gnutls	3.8.0	3.8.0
go-srpm-macros	3.2.0	3.2.0
gpgme	1.15.1	1.15.1
gpm-libs	1.20.7	1.20.7
grep	3.8	3.8
groff-base	1.22.4	1.22.4
grub2-common	2.06	2.06
grub2-efi-x64-ec2	2.06	2.06
grub2-pc		2.06
grub2-pc-modules	2.06	2.06
grub2-tools	2.06	2.06
grub2-tools-minimal	2.06	2.06
grubby	8.40	8.40
gssproxy	0.8.4	0.8.4
gzip	1.12	1.12
hostname	3.23	3.23
hunspell	1.7.0	1.7.0
hunspell-en	0.20140811.1	0.20140811.1

软件包	AMI	Hyper-V VHDX
hunspell-en-GB	0.20140811.1	0.20140811.1
hunspell-en-US	0.20140811.1	0.20140811.1
hunspell-filesystem	1.7.0	1.7.0
hwdata	0.353	0.353
hyperv-daemons		0
hyperv-daemons-license		0
hypervfcopyd		0
hypervkvpd		0
hyperv-tools		0
hypervvssd		0
info	6.7	6.7
inih	49	49
initscripts	10.09	10.09
iproute	5.10.0	5.10.0
iputils	20210202	20210202
irqbalance	1.9.0	1.9.0
jansson	2.14	2.14
jitterentropy	3.4.1	3.4.1
jq	1.7.1	1.7.1
json-c	0.14	0.14

软件包	AMI	Hyper-V VHDX
kbd	2.4.0	2.4.0
kbd-misc	2.4.0	2.4.0
kernel	6.1.79	6.1.79
kernel-livepatch-repo-cdn		2023.4.20240319
kernel-livepatch-repo-s3	2023.4.20240319	
kernel-modules-extra		6.1.79
kernel-modules-extra-common		6.1.79
kernel-srpm-macros	1.0	1.0
kernel-tools	6.1.79	6.1.79
keyutils	1.6.3	1.6.3
keyutils-libs	1.6.3	1.6.3
kmod	29	29
kmod-libs	29	29
kpatch-runtime	0.9.7	0.9.7
krb5-libs	1.21	1.21
less	608	608
libacl	2.3.1	2.3.1
libaio	0.3.111	0.3.111

软件包	AMI	Hyper-V VHDX
libarchive	3.5.3	3.5.3
libargon2	20171227	20171227
libassuan	2.5.5	2.5.5
libattr	2.5.1	2.5.1
libbasicobjects	0.1.1	0.1.1
libblkid	2.37.4	2.37.4
libcap	2.48	2.48
libcap-ng	0.8.2	0.8.2
libcbor	0.7.0	0.7.0
libcollection	0.7.0	0.7.0
libcom_err	1.46.5	1.46.5
libcomps	0.1.20	0.1.20
libconfig	1.7.2	1.7.2
libcurl-minimal	8.5.0	8.5.0
libdb	5.3.28	5.3.28
libdhash	0.5.0	
libdnf	0.69.0	0.69.0
libeconf	0.4.0	0.4.0
libedit	3.1	3.1
libev	4.33	4.33

软件包	AMI	Hyper-V VHDX
libevent	2.1.12	2.1.12
libfdisk	2.37.4	2.37.4
libffi	3.4.4	3.4.4
libfido2	1.10.0	1.10.0
libgcc	11.4.1	11.4.1
libgcrypt	1.10.2	1.10.2
libgomp	11.4.1	11.4.1
libgpg-error	1.42	1.42
libibverbs	48.0	48.0
libidn2	2.3.2	2.3.2
libini_config	1.3.1	1.3.1
libkcapi	1.4.0	1.4.0
libkcapi-hmaccalc	1.4.0	1.4.0
libldb	2.6.2	
libmaxminddb	1.5.2	1.5.2
libmetalink	0.1.3	0.1.3
libmnl	1.0.4	1.0.4
libmodulemd	2.13.0	2.13.0
libmount	2.37.4	2.37.4
libnfsidmap	2.5.4	2.5.4

软件包	AMI	Hyper-V VHDX
libnghttp2	1.57.0	1.57.0
libnl3	3.5.0	3.5.0
libpath_utils	0.2.1	0.2.1
libpcap	1.10.1	1.10.1
libpipeline	1.5.3	1.5.3
libpkgconf	1.8.0	1.8.0
libpsl	0.21.1	0.21.1
libpwquality	1.4.4	1.4.4
libref_array	0.1.5	0.1.5
librepo	1.14.5	1.14.5
libreport-filesystem	2.15.2	2.15.2
libseccomp	2.5.3	2.5.3
libselinux	3.4	3.4
libselinux-utils	3.4	3.4
libsemanage	3.4	3.4
libsepol	3.4	3.4
libsigsegv	2.13	2.13
libsmartcols	2.37.4	2.37.4
libsolv	0.7.22	0.7.22
libss	1.46.5	1.46.5

软件包	AMI	Hyper-V VHDX
libsss_certmap	2.9.4	
libsss_idmap	2.9.4	2.9.4
libsss_nss_idmap	2.9.4	2.9.4
libsss_sudo	2.9.4	
libstdc++	11.4.1	11.4.1
libstoragemgmt	1.9.4	1.9.4
libtalloc	2.3.4	
libtasn1	4.19.0	4.19.0
libtdb	1.4.7	
libtevent	0.13.0	
libtextstyle	0.21	0.21
libtirpc	1.3.3	1.3.3
libunistring	0.9.10	0.9.10
libuser	0.63	0.63
libutempter	1.2.1	1.2.1
libuuid	2.37.4	2.37.4
libuv	1.47.0	1.47.0
libverto	0.3.2	0.3.2
libverto-libev	0.3.2	0.3.2
libxcrypt	4.4.33	4.4.33

软件包	AMI	Hyper-V VHDX
libxml2	2.10.4	2.10.4
libyaml	0.2.5	0.2.5
libzstd	1.5.5	1.5.5
lm_sensors-libs	3.6.0	3.6.0
lmdb-libs	0.9.29	0.9.29
logrotate	3.20.1	3.20.1
lsof	4.94.0	4.94.0
lua-libs	5.4.4	5.4.4
lua-srpm-macros	1	1
lz4-libs	1.9.4	1.9.4
man-db	2.9.3	2.9.3
man-pages	5.10	5.10
microcode_ctl	2.1	2.1
mpfr	4.1.0	4.1.0
nano	5.8	5.8
ncurses	6.2	6.2
ncurses-base	6.2	6.2
ncurses-libs	6.2	6.2
nettle	3.8	3.8
net-tools	2.0	2.0

软件包	AMI	Hyper-V VHDX
newt	0.52.21	0.52.21
nfs-utils	2.5.4	2.5.4
npth	1.6	1.6
nspr	4.35.0	4.35.0
nss	3.90.0	3.90.0
nss-softokn	3.90.0	3.90.0
nss-softokn-freebl	3.90.0	3.90.0
nss-sysinit	3.90.0	3.90.0
nss-util	3.90.0	3.90.0
ntsysv	1.15	1.15
numactl-libs	2.0.14	2.0.14
ocaml-srpm-macros	6	6
oniguruma	6.9.7.1	6.9.7.1
openblas-srpm-macros	2	2
openldap	2.4.57	2.4.57
openssh	8.7p1	8.7p1
openssh-clients	8.7p1	8.7p1
openssh-server	8.7p1	8.7p1
openssl	3.0.8	3.0.8
openssl-libs	3.0.8	3.0.8

软件包	AMI	Hyper-V VHDX
openssl-pkcs11	0.4.12	0.4.12
os-prober	1.77	1.77
p11-kit	0.24.1	0.24.1
p11-kit-trust	0.24.1	0.24.1
package-notes-srpm-macros	0.4	0.4
pam	1.5.1	1.5.1
parted	3.4	3.4
passwd	0.80	0.80
pciutils	3.7.0	3.7.0
pciutils-libs	3.7.0	3.7.0
pcre2	10.40	10.40
pcre2-syntax	10.40	10.40
perl-Carp	1.50	1.50
perl-Class-Struct	0.66	0.66
perl-constant	1.33	1.33
perl-DynaLoader	1.47	1.47
perl-Encode	3.15	3.15
perl-Errno	1.30	1.30
perl-Exporter	5.74	5.74
perl-Fcntl	1.13	1.13

软件包	AMI	Hyper-V VHDX
perl-File-Basename	2.85	2.85
perl-File-Path	2.18	2.18
perl-File-stat	1.09	1.09
perl-File-Temp	0.231.100	0.231.100
perl-Getopt-Long	2.52	2.52
perl-Getopt-Std	1.12	1.12
perl-HTTP-Tiny	0.078	0.078
perl-if	0.60.800	0.60.800
perl-interpreter	5.32.1	5.32.1
perl-IO	1.43	1.43
perl-IPC-Open3	1.21	1.21
perl-libs	5.32.1	5.32.1
perl-MIME-Base64	3.16	3.16
perl-mro	1.23	1.23
perl-overload	1.31	1.31
perl-overloading	0.02	0.02
perl-parent	0.238	0.238
perl-PathTools	3.78	3.78
perl-Pod-Escapes	1.07	1.07
perl-podlators	4.14	4.14

软件包	AMI	Hyper-V VHDX
perl-Pod-Perldoc	3.28.01	3.28.01
perl-Pod-Simple	3.42	3.42
perl-Pod-Usage	2.01	2.01
perl-POSIX	1.94	1.94
perl-Scalar-List-Utils	1.56	1.56
perl-SelectSaver	1.02	1.02
perl-Socket	2.032	2.032
perl-srpm-macros	1	1
perl-Storable	3.21	3.21
perl-subs	1.03	1.03
perl-Symbol	1.08	1.08
perl-Term-ANSIColor	5.01	5.01
perl-Term-Cap	1.17	1.17
perl-Text-ParseWords	3.30	3.30
perl-Text-Tabs+Wrap	2021.0726	2021.0726
perl-Time-Local	1.300	1.300
perl-vars	1.05	1.05
pkgconf	1.8.0	1.8.0
pkgconf-m4	1.8.0	1.8.0
pkgconf-pkg-config	1.8.0	1.8.0

软件包	AMI	Hyper-V VHDX
policycoreutils	3.4	3.4
policycoreutils-python-utils	3.4	
popt	1.18	1.18
procps-ng	3.3.17	3.3.17
protobuf-c	1.4.1	1.4.1
psacct	6.6.4	6.6.4
psmisc	23.4	23.4
publicsuffix-list-dafsa	20240212	20240212
python3	3.9.16	3.9.16
python3-attrs	20.3.0	20.3.0
python3-audit	3.0.6	3.0.6
python3-awscrt	0.19.19	0.19.19
python3-babel	2.9.1	2.9.1
python3-cffi	1.14.5	1.14.5
python3-chardet	4.0.0	4.0.0
python3-colorama	0.4.4	0.4.4
python3-configobj	5.0.6	5.0.6
python3-cryptography	36.0.1	36.0.1
python3-daemon	2.3.0	

软件包	AMI	Hyper-V VHDX
python3-dateutil	2.8.1	2.8.1
python3-dbus	1.2.18	1.2.18
python3-distro	1.5.0	1.5.0
python3-dnf	4.14.0	4.14.0
python3-dnf-plugins-core	4.3.0	4.3.0
python3-docutils	0.16	0.16
python3-gpg	1.15.1	1.15.1
python3-hawkey	0.69.0	0.69.0
python3-idna	2.10	2.10
python3-jinja2	2.11.3	2.11.3
python3-jmespath	0.10.0	0.10.0
python3-jsonpatch	1.21	1.21
python3-jsonpointer	2.0	2.0
python3-jsonschema	3.2.0	3.2.0
python3-libcomps	0.1.20	0.1.20
python3-libdnf	0.69.0	0.69.0
python3-libs	3.9.16	3.9.16
python3-libselinux	3.4	3.4
python3-libsemanage	3.4	3.4

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

软件包	AMI	Hyper-V VHDX
python3-ruamel-yaml	0.16.6	0.16.6
python3-ruamel-yaml-clib	0.1.2	0.1.2
python3-setools	4.4.1	4.4.1
python3-setuptools	59.6.0	59.6.0
python3-setuptools-wheel	59.6.0	59.6.0
python3-six	1.15.0	1.15.0
python3-systemd	235	235
python3-urllib3	1.25.10	1.25.10
python3-wcwidth	0.2.5	0.2.5
python-chevron	0.13.1	
python-srpm-macros	3.9	3.9
quota	4.06	4.06
quota-nls	4.06	4.06
readline	8.1	8.1
rng-tools	6.14	6.14
rootfiles	8.1	8.1
rpcbind	1.2.6	1.2.6
rpm	4.16.1.3	4.16.1.3
rpm-build-libs	4.16.1.3	4.16.1.3

软件包	AMI	Hyper-V VHDX
rpm-libs	4.16.1.3	4.16.1.3
rpm-plugin-selinux	4.16.1.3	4.16.1.3
rpm-plugin-systemd-inhibit	4.16.1.3	4.16.1.3
rpm-sign-libs	4.16.1.3	4.16.1.3
rsync	3.2.6	3.2.6
rust-srpm-macros	21	21
sbsigntools	0.9.4	0.9.4
screen	4.8.0	4.8.0
sed	4.8	4.8
selinux-policy	37.22	37.22
selinux-policy-targeted	37.22	37.22
setup	2.13.7	2.13.7
shadow-utils	4.9	4.9
slang	2.3.2	2.3.2
sqlite-libs	3.40.0	3.40.0
sssd-client	2.9.4	2.9.4
sssd-common	2.9.4	
sssd-kcm	2.9.4	
sssd-nfs-idmap	2.9.4	

软件包	AMI	Hyper-V VHDX
strace	5.16	5.16
sudo	1.9.14	1.9.14
sysctl-defaults	1.0	1.0
sysstat	12.5.6	12.5.6
systemd	252.16	252.16
systemd-libs	252.16	252.16
systemd-networkd	252.16	252.16
systemd-pam	252.16	252.16
systemd-resolved	252.16	252.16
systemd-udev	252.16	252.16
system-release	2023.4.20240319	2023.4.20240319
systemtap-runtime	4.8	4.8
tar	1.34	1.34
tbb	2020.3	2020.3
tcpdump	4.99.1	4.99.1
tcsh	6.24.07	6.24.07
time	1.9	1.9
traceroute	2.1.3	2.1.3
tzdata	2024a	2024a
unzip	6.0	6.0

软件包	AMI	Hyper-V VHDX
update-motd	2.2	2.2
userspace-rcu	0.12.1	0.12.1
util-linux	2.37.4	2.37.4
util-linux-core	2.37.4	2.37.4
vim-common	9.0.2153	9.0.2153
vim-data	9.0.2153	9.0.2153
vim-enhanced	9.0.2153	9.0.2153
vim-filesystem	9.0.2153	9.0.2153
vim-minimal	9.0.2153	9.0.2153
wget	1.21.3	1.21.3
which	2.21	2.21
words	3.0	3.0
xfsdump	3.1.11	3.1.11
xfsprogs	5.18.0	5.18.0
xxd	9.0.2153	9.0.2153
xxhash-libs	0.8.0	0.8.0
xz	5.2.5	5.2.5
xz-libs	5.2.5	5.2.5
yum	4.14.0	4.14.0
zip	3.0	3.0

软件包	AMI	Hyper-V VHDX
`zlib`	1.2.11	1.2.11
`zram-generator`	1.1.2	
`zram-generator-defaults`	1.1.2	
`zstd`	1.5.5	1.5.5

文件系统布局

本节介绍 AL2 023 系统的文件系统布局，包括可能特定于实例或基于 AL2 023 的容器的详细信息。请参阅`file-hierarchy(7)` man 页面以获取更多信息。

主题

- / (根目录)
- /boot (内核，initramfs，等等)
- /etc (系统配置)
- /home (用户主目录)
- /root (root 用户主目录)
- /srv (服务器负载)
- /tmp (小型临时文件)
- /run (运行时数据)
- /usr (系统资源)
- /var (永久变量系统数据)

/ (根目录)

默认情况下， AL2023 个映像配置为可写镜像/，允许特权用户创建新的文件和目录。

可以将`systemd`服务配置为使用与该服务不同的路径或图像显示，也可以对任何路径设置访问限制。/

> **Note**
>
> 最佳做法是将`systemd`服务配置为限制该服务可以访问的内容。这可能包括使用该`ReadOnlyPaths=/`指令，该指令将该服务设为只/读。
> 有关使用限制服务`systemd`对系统的访问权限的更多信息，请参阅 `systemd.exec(5)` man 页面。

/boot (内核，initramfs，等等)

默认情况下，可启动的 AL2 023 映像配置/boot为 root 文件系统。该/boot路径仅与可启动映像相关，因此在 AL2 023 容器映像中未使用。

此目录是 AL2 023 启动所需的文件所在地，例如 Linux 内核和 initramfs。 只能使用操作系统提供的工具来操作此目录的内容。

/boot/efi（EFI 系统分区）

默认情况下，可启动的 AL2 023 映像配置为 EFI System 分区正在挂载到/boot/efi。此文件系统由操作系统管理，包含对启动系统至关重要的代码和配置。

此路径与容器镜像无关。

/etc（系统配置）

AL2023 上的/etc目录包含系统特定的配置。默认情况下，/etc上面有 AL2 023 张图片 root 文件系统，可由特权用户写入。

> **Note**
>
> 应用程序（包括systemd）通常会保留默认配置，/usr（系统资源）在该配置下可以通过将配置放入来覆盖。/etc（系统配置）
> 对于这些应用程序，更改中的文件/usr（系统资源）而不是覆盖中的默认配置，很/etc可能会导致更新软件包时所做的更改被覆盖。

/home（用户主目录）

普通用户的主目录位于下方/home，但是软件应始终查找每个用户的$HOME环境变量，而不是依赖诸如/home/$USER之类的模式。

默认情况下，有 AL2 023 张图片/home在 root 文件系统，但软件不应依赖于此。将操作系统配置为/home单独的文件系统是完全有效的，该文件系统将在启动后期装载，或者只有在用户向系统进行身份验证后才会装载。

这些区域有：root 用户主目录不在中/root (root 用户主目录)，/home而是为了在无法挂载/home文件系统时可以使用。

> **Note**
>
> 对于不需要写入权限的systemd服务，最好使用该ProtectHome=read-only指令进行配置。/home使用此选项，/home/root、和将设置/run/user为该服务的只读状态。

使用指令配置不需要任何访问权限的服务也是最佳实践，该ProtectHome=tmpfs指令将在沙箱中运行该服务/home，其中/root、和/run/user是空的只读tmpfs文件系统。/home 有关使用限制服务systemd对系统的访问权限的更多信息，请参阅 systemd.exec(5) man 页面。

/root (root 用户主目录)

的主目录 root user 是/root目录，故意与之分开，/home（用户主目录）这样当文件系统不可用时，/home（用户主目录）它就会出现。

配置systemd服务的最佳做法与配置服务的最佳/root实践相同/home（用户主目录）。

/srv（服务器负载）

该/srv目录由系统管理员管理，Amazon Linux 2023 对该目录的组织方式没有任何限制。

可以将该/srv目录配置为位于单独的文件系统上，因此它只能在稍后启动时才可用。

/tmp（小型临时文件）

> **Note**
>
> 亚马逊 Linux 2023 与亚马逊 Linux 2 的不同之处在于，默认情况下/tmp是现在，tmpfs而不是前进的道路 root 文件系统。

> **Note**
>
> 在容器中运行时，通常由您的容器运行时配置来决定是否/tmp是tmpfs，或者磁盘上的路径，以及是否有正在运行的清理过程。

该/tmp目录用于存放大小有限的小临时文件。默认情况下， AL2023 将其配置为tmpfs文件系统，其大小限制为 RAM 的 50%，最大为 100 万 inodes.

应用程序应该更喜欢$TMPDIR环境变量中的路径/tmp。然后，用户可以将$TMPDIR环境变量设置为覆盖应用程序应使用的路径 /tmp

对于较大的临时文件，/var/tmp应改用。

> **Warning**
>
> 由于/tmp是共享的，因此使用安全的方法来创建临时文件非常重要。有关详细信息，请参阅有关使用/tmp和/var/tmp安全的上游systemd文档。

> **Note**
>
> 最佳做法是将PrivateTmp=指令设置为yes或在不与主机或其他disconnected服务共享/tmp/var/tmp且不与主机或其他服务共享的沙箱中运行服务。systemd
> 有关更多信息，包括如何配置两个服务以共享相同的私有临时目录，请参阅systemd.exec(5) man 页面。

的内容/tmp通常会在启动时清理，未使用的文件会定期清理。默认情况下，清理过程会在启动后不久运行，然后每天运行。有关如何配置临时文件清理的信息，请参阅和 tmpfiles.d(5) systemd-tmpfiles(8) man 手册页。

/tmp和/var/tmp路径密切相关，存在的目的也不同。

/run（运行时数据）

系统包使用该/run目录来存储少量运行时数据（例如套接字文件）。它是一个tmpfs文件系统，只能由特权程序写入。

无论是在写入文件系统之前，还是在/var/log文件系统可用之前，系统组件都可以使用该/run/log目录来/var/log存储日志。

该/run/user/路径包含每个用户的运行时目录。默认情况下，这些tmpfs文件系统将在用户登录systemd时挂载，并在用户不再登录时删除。根据 XDG Base Directory 规范，不应直接引用这些路径，而应通过$XDG_RUNTIME_DIR环境变量引用。

/usr（系统资源）

该/usr层次结构适用于供应商提供的操作系统资源。除了/usr/local层次结构之外，除了操作系统包管理器/usr之外，其他任何内容都不应被修改。

软件应用程序必须假设它/usr可以是只读的。/usr层次结构不得用于易失性数据。除此之外/usr/local，不得像操作系统包管理器那样将/usr层次结构用于在软件包安装/删除之外添加或更改的任何数据。操作系统包管理器可能会假设所有/usr层次结构（除外/usr/local）都是相同的挂载点。

安装在操作系统包管理器之外的软件不应在中存储数据，/usr因为这可能会阻碍将来对操作系统包管理器的任何调用。/usr/local层次结构是个例外，它是为操作系统包管理器之外的软件保留的。

/usr/bin（可执行文件）

可执行文件，这些文件应出现在标准搜索中$PATH，并且对于从 shell 调用很有用。从 shell 调用没有用处的守护程序和可执行文件则存在于或中。/usr/lib /usr/libexec

/usr/include（C/C++ 头文件）

该/usr/include目录包含 C 和 C++ 头文件，通常包含在带有-devel后缀的软件包中。

/usr/lib和/usr/lib64（共享库）

在 Amazon Linux 2023 上，该/usr/lib64路径用于 64 位共享库和依赖架构的包数据。由于 AL2023 不提供任何 32 位用户空间支持，因此只有 64 位共享库可用。

该/usr/lib路径用于来自与所有架构兼容的操作系统包中的静态数据。这可能包括通常不从 shell 调用的可执行文件，也可以在中找到。/usr/libexec共享库位于/usr/lib64而不是中/usr/lib。

/usr/local（系统管理员安装的软件）

在 Amazon Linux 2023 上，系统管理员可以使用该/usr/local路径在其中安装软件，这些软件不归操作系统所有，操作系统也不会触及。默认/usr/local层次结构反映了/层次结构。

/usr/share（共享资源）

共享资源，例如文档、字体和时区数据/usr/share。各种规格通常会规定数据存储在此目录中的确切位置和格式。

/usr/share/doc（文档）

软件包附带的文档将存储在/usr/share/doc。

/var（永久变量系统数据）

/var/cache（缓存）

相比之下/var/lib，擦除中的数据/var/cache不会导致数据丢失，因为应用程序需要能够从其他来源重建/var/cache数据。

/var/lib（永久系统数据）

该/var/lib目录用于存储永久性系统数据。各种系统组件将在此处放置该组件专用的数据。相比之下/var/cache，擦除中的数据/var/lib会导致数据丢失。

例如，默认情况下，PostgreSQL 数据库服务器将在中存储数据库数据。/var/lib/pgsql这些数据的布局和文件格式对PostgreSQL来说是私有的，而且它是永久性数据，就像被删除一样，用户会遇到数据丢失的情况。

/var/log（永久日志）

此目录用于存储永久性日志。建议软件使用syslog(3)或 sd_journal_print(3) API 调用，而不是直接在下存储日志文件/var/log。

> **Note**
>
> 在 AL2 023 中systemd日记替换 rsyslog，这与默认的 Amazon Linux 2 配置有显著的区别。

有关使用读取日志的更多信息journalctl，请参阅journalctl手册页。

许多应用程序使用自己的机制来写入中找到的日志文件，有时甚至是轮换日志文件/var/log。有关如何配置其日志文件的信息，请参阅这些应用程序的文档。

/var/spool（邮件和打印机队列）

此目录用于存储诸如邮件或打印机队列之类的永久性数据。

/var/tmp（较大的临时文件）

对于大小有限的小临时文件，可能/tmp应该改用。

虽然/tmp默认配置为tmpfs卷，/var/tmp但默认配置为根文件系统上的路径，因此是存放更大、更持久的临时文件的地方。默认情况下，会定期运行清理作业，它会删除最近未访问过的文件。

有关如何配置临时文件清理的信息，请参阅和 `tmpfiles.d(5)` `systemd-tmpfiles(8)` man 手册页。

与之类似/tmp，应用程序应优先使用$TMPDIR环境变量中指定的路径/var/tmp。然后，用户可以将$TMPDIR环境变量设置为覆盖应用程序应使用的路径/var/tmp。

> **Warning**
>
> 由于/var/tmp是共享的（按原样）/tmp，因此使用安全的方法来创建临时文件非常重要。有关详细信息，请参阅有关使用/tmp和/var/tmp安全的上游systemd文档。

> **Note**
>
> 最佳做法是将PrivateTmp=指令设置为yes或在不与主机或其他disconnected服务共享/tmp/var/tmp且不与主机或其他服务共享的沙箱中运行服务。systemd
> 有关更多信息，包括如何配置两个服务以共享相同的私有临时目录，请参阅 `systemd.exec(5)` man 页面。

/tmp和/var/tmp路径密切相关，存在的目的也不同。

正在更新 AL2 023

请务必及时了解最新的 AL2 023 版本，这样您就可以从安全更新和新功能中受益。使用 AL2 023，您可以通过023 年通过版本化存储库进行确定性升级 AL2以下方式确保整个环境中的软件包版本和更新之间的一致性。

> **Warning**
>
> 运行`dnf --releasever=latest update`不是最佳实践，很可能会导致操作系统更新首先在生产环境中进行测试。
>
> 与其使用`latest`，不如使用特定的 AL2 023 发行版。这样可以确保您在生产实例中部署的更改与之前测试的相同。例如，`dnf --releasever=2023.7.20250331 update`将始终更新到 2023.7.20250331 版本。
>
> 有关更多信息，请参阅《AL2023 用户指南》中的 “更新 AL2 023” 部分。

主题

- 安全部署更新的最佳实践
- 接收有关新更新的通知
- 023 年通过版本化存储库进行确定性升级 AL2
- 在 AL2 023 中管理软件包和操作系统更新
- 023 上的 Kernel Live AL2 补丁
- 在 AL2 023 上更新 Linux 内核

安全部署更新的最佳实践

Amazon Linux 2023 (AL2023) 具有多项功能，旨在帮助安全地将更新部署到操作系统，并能够知道两次更新之间发生了哪些变化，并在必要时轻松恢复到旧版本。本节探讨了 AWS 从十多年内部和外部使用 Amazon Linux 中吸取的经验教训。

> **Warning**
>
> 运行`dnf --releasever=latest update`不是最佳实践，很可能会导致操作系统更新首先在生产环境中进行测试。

与其使用`latest`，不如使用特定的 AL2 023 发行版。这样可以确保您在生产实例中部署的更改与之前测试的相同。例如，`dnf --releasever=2023.7.20250331 update`将始终更新到 2023.7.20250331 版本。
有关更多信息，请参阅《AL2023 用户指南》中的 “更新 AL2 023” 部分。

如果不对操作系统更新的部署安全性进行规划，则应用程序/服务与操作系统更新之间意外负面交互的影响可能会大得多，包括完全中断。与任何软件问题一样，越早发现问题，对最终用户的影响就越小。

重要的是不要陷入相信两件根本不正确的事情的陷阱：

1. 操作系统供应商在更新操作系统时绝不会犯错误。
2. 您所依赖的操作系统的具体行为或接口与操作系统供应商认为值得依赖的行为和接口相匹配。

   也就是说，操作系统供应商和您都同意更新存在问题。

不要依赖良好的意图，应将系统部署到位，以确保部署安全包括对操作系统的任何更新。

不建议通过部署到生产环境来测试新的操作系统更新。最佳做法是将操作系统视为部署的另一部分，并考虑应用您认为适用于生产环境的任何其他更改的相同部署安全机制。

在部署到生产系统之前，最好先测试所有操作系统更新。部署时，建议将分阶段部署与良好的监控相结合。分阶段部署可以确保在出现问题（即使不是立即出现的）时，影响也仅限于队列的子集，并且可以停止更新的进一步部署，同时可以进行进一步的调查和缓解措施。

减轻更新操作系统带来的任何负面影响通常是第一要务，其次是解决问题，无论问题在哪里。如果引入操作系统更新与负面影响相关，那么恢复到先前已知良好的操作系统版本的能力是一个强大的工具。

Amazon Linux 2023 推出通过版本控制的存储库进行确定性升级了一项强大的新功能，可确保操作系统版本（或单个软件包）的任何更改都是可重复的。因此，如果在从一个操作系统版本转移到下一个操作系统版本时遇到问题，则在研究如何解决问题的同时，有一些易于使用的机制可以坚持使用已知有效的操作系统版本。

在 AL2 023 中，每当我们发布新的软件包更新时，都会有一个新版本需要锁定，而新 AMIs 版本则锁定到该版本。AL2023 发行说明涵盖了每个版本中的更改，并2023 年的亚马逊 Linux 安全公 AL2告涵盖了软件包更新中解决的安全问题。

例如，如果您受到 2023.6. 20241028 版本中存在的问题的影响，则可以立即恢复使用先前版本 2023.6.20241010 的 AMIs 和容器镜像。在这种情况下，软件包中存在一个错误，该错误已在随后的

2023.6.20241031 版本中修复，但是通过版本控制的存储库进行确定性升级对于任何受影响的人，都可以立即采取简单的措施来缓解：只需使用之前的图像即可。

通过版本控制的存储库进行确定性升级还可确保任何正在进行的操作系统更新部署，无论是就地部署还是通过启动新的映像 AMIs 或容器映像，都不会受到随后发布的操作系统更新的影响。

在我们的第一个例子中，舰队 A 是一个庞大的舰队，在 2023.6.20241028 版本发布时，从 2023.5.20241001 部署到 2023.6.20241010 版本的更新已经完成了一半。 通过版本控制的存储库进行确定性升级意味着舰队 A 的部署仍在继续，所应用的更新没有任何变化。

基于波浪或分阶段的部署策略（例如首先部署到舰队的1%，然后部署到5%、10%、20%、40%，直到达到100%）的目的是能够在扩大范围之前以有限的方式测试变更。这种部署策略通常被认为是部署任何生产变更的最佳实践。

由于采用了基于波浪的部署策略，而且 2023.6.20241010 的舰队 A 更新处于同时部署到许多主机的阶段，因此 2023.6.20241028 的发布对正在进行的部署没有影响，这要归功于使用。 通过版本控制的存储库进行确定性升级

如果舰队 B 运行的是旧版本，比如 2023.5.20240708，并已开始将更新部署到 2023.6.20241028，而舰队 B 受到该版本中问题的影响，则在部署初期就会注意到这一点。届时，可以决定是否在该问题得到修复之前暂停任何部署，或者在此期间是否开始部署队列A正在运行的相同版本，即 2023.6.20241010，以便舰队B获得2023.5. 20240708和2023.6.20241010 之间的所有更新。

请务必注意，不及时更新操作系统可能会导致问题。新的更新可能包含可能与您的环境相关的错误和安全补丁。有关更多信息，请参阅Amazon Linux 2023 中的安全性与合规性和在 AL2 023 中管理软件包和操作系统更新。

重要的是要配置您的部署系统，使其能够轻松获取新的操作系统更新，在部署到生产环境之前对其进行测试，并使用基于波浪的部署等机制来最大限度地减少任何负面影响。为了能够减轻操作系统更新的任何负面影响，重要的是要知道如何使部署系统指向先前已知良好的操作系统版本，问题得到解决后，不再锁定到已知良好的旧版本，而是移至已知良好的新版本。

为次要更新做准备

为操作系统的较小更新（例如 AL2 023的新单点版本）做准备工作将仅限于零努力。请务必阅读 AL2023 发行说明，了解即将发生的任何变化。

即将结束的软件包的支持期可能涉及迁移到较新版本的语言运行时（例如 withPHP 在 AL2 023）。最佳做法是提前做好准备，在支持期结束之前轻松迁移到新的语言运行时版本。

对于诸如此类的软件包pcre 版本 1，还有机会提前计划并将您的任何代码迁移到其替代品（在本例中为pcre版本 2）。最好的做法是尽快这样做，以便有时间应对任何挫折。

如果没有直接替换，例如使用伯克利数据库 (libdb)，则可能需要根据自己的用例做出选择。

为重大更新做准备

人们几乎普遍认为，更新到新的主要操作系统版本需要规划、努力适应已更改或已弃用的功能，还需要在部署之前进行测试。能够更逐步地为 Amazon Linux 2023 的下一个主要版本做好准备的情况并不少见，例如在继续迁移到下一个主要版本之前，解决对已弃用或已删除功能的任何使用问题。

例如，从023移动 AL2 到 AL2 023时，阅读该在 023 年中弃用 AL2 并移除的 AL2功能部分可能会导致许多安全而小的步骤，这些步骤可能会在使用时发生， AL2 以便为 AL2 023做准备。例如，任何Python 2.7 已被 Python 3 所取代用法（操作系统使用之外，例如在yum程序包管理器中）都可以迁移到 Python 3，为使用做准备Python 在 AL2 023。如果使用 PHP， AL2 （通过 PHP 8.2 E AL2 x tra）和 AL2 023 都提供 PHP 8.2，因此 PHP 版本迁移和操作系统迁移不必同时进行。

在使用 AL2 023 的同时，也可以为今天的亚马逊 Linux 2023 的下一个主要版本做准备，同时使用 AL2 023。本2023 年 AL2已被弃用节介绍了 AL2 023 年已弃用且即将移除的功能和软件包。

例如，将任何剩余的System V init (sysvinit)用途（例如init脚本）迁移到其systemd等效项将使您为未来做好准备，并允许您使用全套systemd功能来监视服务、如何以及是否重启服务、它需要哪些其他服务，以及是否应应用任何资源或权限限制。

对于 32 位支持等功能，弃用可能跨越操作系统的多个主要版本。对于 32 位，亚马逊 Linux 1 (AL1) 已弃用，32 位 x86 (i686) AMIs亚马逊 Linux 2 已弃用，亚马逊 Linux 32 位 x86 (i686) 软件包 2023 已弃用。32 位 x86 (i686) 运行时支持从该操作系统的过渡IMDSv1还跨越了操作系统的多个主要版本。对于这些类型的更改，据了解，有些客户需要更长的时间来适应这些更改，因此，在 Amazon Linux 2023 中不再提供该功能之前，还有很大的回旋余地。

已弃用功能的列表会在操作系统的生命周期内更新，建议及时了解其更改。

接收有关新更新的通知

每当新的 AL2 023 AMI 发布时，您都可以收到通知。通过 Amazon SNS 使用以下主题发布通知。

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

发布新的 AL2 023 AMI 时，将在此处发布消息。AMI 的版本将包含在消息中。

可以使用几种不同的方法接收这些消息。我们建议您使用以下方法。

1. 打开 Amazon SNS 控制台。
2. 如有必要，在导航栏中更改 AWS 区域 为美国东部（弗吉尼亚北部）。必须选择创建您订阅的 SNS 通知的区域。
3. 在导航窗格中，依次选择订阅、创建订阅。
4. 对于 Create subscription 对话框，执行以下操作：
   a. 对于主题 ARN，请复制并粘贴以下亚马逊资源名称 (ARN)：。**`arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates`**
   b. 对于协议，选择电子邮件。
   c. 对于端点，输入可用来接收通知的电子邮件地址。
   d. 选择创建订阅。
5. 您会收到一封主题为 “AWS 通知-订阅确认” 的确认电子邮件。打开该电子邮件，选择确认订阅来完成订阅。

023 年通过版本化存储库进行确定性升级 AL2

> **Note**
> 默认情况下，您的 AL2 023 实例在启动时不会自动收到其他关键和重要的安全更新。您的实例最初包含 AL2 023 版本中可用的更新和所选 AMI。

控制从主要版本和次要版本收到的更新

使用 AL2 023，您可以确保整个环境中的软件包版本和更新之间的一致性。您还可以确保同一个 Amazon 机器映像 (AMI) 的多个实例的一致性。利用通过版本控制的存储库的确定性升级功能（默认开启），您可以根据满足您特定需求的时间表来应用更新。

每当我们发布新的软件包更新时，都会有一个新版本需要锁定，而新 AMIs 版本则锁定到该版本。

AL2023 锁定存储库的特定版本。主要版本和次要版本都支持此功能。通过我们的 SSM 参数公开的 AL2 023 AMI 始终是最新版本。它包含最多的 up-to-date软件包和更新，包括关键和重要的安全更新。

如果您从一个现有 AMI 启动实例，则不会自动应用更新。作为配置的一部分安装的任何其他软件包映射到现有 AMI 的存储库版本。

使用此功能，您可以负责确保整个环境中软件包版本和更新之间的一致性。当您从同一个 AMI 启动多个实例时，则功能尤其有用。您可以根据满足您需求的时间表应用更新。您还可以在启动时应用一组特定的更新，因为这些更新也可以锁定到特定的存储库版本。

主要版本升级和次要版本升级之间的区别

AL2023 的主要版本包括大规模更新，可能会添加、删除或更新软件包。在将实例升级到某个新的主要版本时，为确保兼容性，应先在该版本上测试了您的应用程序后再升级。

AL2023 的次要版本包括功能和安全更新，但不包括软件包更改。这确保了 Linux 功能和系统库 API 在新版本中保持可用。无需在更新之前测试应用程序。

知道何时有更新

要应用更新，您需要知道更新是否可用，然后知道如何部署更新。

对于新 AL2 023 AMIs 发布 AMIs 时派生的建筑物，EC2 Image Builder 可以自动构建、修补和测试 AMIs。要触发自己的 AMI 构建管道或使用基础 AMIs，您可以接收有关新更新的通知。

要进行就地修补，您可以使用诸如 Pat AWS Systems Manager ch Manager 之类的工具来协调在队列中应用更新。

对于其他 AMIs 基于 AL2 023的公众，其提供者 AMIs 可能有自己的发布时间表和通知方法。使用派生镜像 AMIs 或容器镜像时，请查看发行商提供的文档，了解何时发布更新。

每个版本的更改都记录在 AL2023 版本说明中。安全更新发布在亚马逊 Linux 安全中心 (ALAS) 上。

控制 AL2 023 存储库中可用的软件包更新

当我们发布 AL2 023 存储库的新版本时，所有以前的版本仍然可用。默认情况下，用于管理存储库版本的插件锁定到用于构建 AMI 的相同版本。如果要控制软件包更新，请按照下列步骤操作。

1. 通过运行以下命令发现可用的存储库版本。

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

2. 您可以通过运行以下命令选择一个版本。

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

该命令使用 dnf 启动一个更新，从您当前的 Amazon Linux 发行版本更新为命令行中指定的发行版本。dnf 将显示一个软件包更新列表。在处理更新之前，您必须先确认更新。更新完成后，新的发行版本变成 dnf 用于所有未来活动的默认发行版本。

有关更多信息，请参阅 在 AL2 023 中管理软件包和操作系统更新。

通过实例替换进行确定性更新

Amazon Linux 2023 的023 年通过版本化存储库进行确定性升级 AL2功能使实例替换成为确定性、安全地推出 023 更新版本的 AL2简便方法。确定性更新意味着，随着新版本的逐步推出，如果发现任何问题，在确定问题原因的同时，可以很容易地恢复到以前的 AMI。

使用实例替换而不是就地修补意味着更新更具确定性和可预测性，因为启动新容量可能是一个经过充分测试的代码路径，具有清晰的 A 和 B 状态。部署开始之前，可以在 CI/CD 系统中对每种之前和之后的状态进行很好的测试。

在进行就地修补时，在应用更新之前和之后之间存在许多中间状态，因此很难测试所有状态组合。

使用具有确定性更新的实例替换的操作系统更新策略非常适合蓝/绿、波浪和基于阶段的部署模型。

通过版本控制的存储库使用确定性升级

主题

- 使用确定性升级系统
- 选择性更新确定性升级系统
- 在确定性升级中使用持久取代

使用确定性升级系统

> **Note**
>
> 包管理器的默认行为已从改为 AL2。

确定性升级是一种强有力的方法，可以确保在广泛部署之前对生产环境的所有更改进行全面测试。每个新的 AL2 023 AMI 都锁定到特定版本的 AL2 023。这提供了启动特定 AMI 时安装了哪些版本的操作系统包的确定性行为。可以对特定的发行版本进行就地更新，从而确保队列中的行为具有确定性。当你迁

移到新的 AMIs 或就地更新版本时，你可以测试你的 CI/CD 管道中的每个版本，在部署到生产环境之前发现任何潜在的问题。

您可以使用诸如 AWS Systems Manager Patch Manager 之类的工具来协调在队列中应用更新。对于 AMIs 在新 AL2 023 AMIs 发布时派生的构建，EC2 Image Builder 可以自动构建 AMIs、修补和测试，或者你可以接收有关新更新的通知知道新基础何时可 AMIs 用，或者触发自己的 AMI 构建管道。

有关限制对来自特定通告的更新的信息，请参阅 就地应用安全更新

要进行就地修补，您可以使用dnf软件包管理器。但您运行 `dnf upgrade` 命令时，系统会检查 `releasever` 变量指定的存储库中的升级。有效版本`releasever`是其中一个*`latest`*或一个带有日期戳的版本，例如. *`2023.7.20250331`*

您可以使用以下任一方法更改 `releasever` 的值。这些方法按系统优先级降序列出。这意味着方法 1 优先于方法 2 和 3，方法 2 优先于方法 3。

1. 命令行标志中的值，`--releasever=`*`latest`*（如果使用）。
2. 替代变量文件中指定的值，`/etc/dnf/vars/releasever`（如果设置）。
3. 当前安装的 `system-release` 软件包版本。

在以下示例中，版本为*`2023.0.20230210`*：

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

在新安装的系统中，替代变量不存在。由于系统锁定到 `system-release` 的已安装版本，所以无升级可用。

```
$ cat /etc/dnf/vars/releasever
cat: /etc/dnf/vars/releasever: No such file or directory
```

```
$ sudo dnf upgrade
Last metadata expiration check: 0:00:02 ago on Wed 15 Feb 2023 06:14:12 PM UTC.
Dependencies resolved.
Nothing to do.
Complete!
```

要获取某个特定版本的软件包，您可以通过使用 `releasever` 标志来提供所需的版本。

```
$ rpm -q system-release
system-release-2023.0.20230222-0.amzn2023.noarch
```

```
$ sudo dnf upgrade --releasever=2023.0.20230329
Amazon Linux 2023 repository                      26 MB/s |  12 MB     00:00
Dependencies resolved.
================================================================================
 Package                 Arch    Version                      Repository   Size
================================================================================
Installing:
 kernel                   aarch64 6.1.21-1.45.amzn2023        amazonlinux  26 M
Upgrading:
 amazon-linux-repo-s3     noarch  2023.0.20230329-0.amzn2023  amazonlinux  18 k
 ca-certificates          noarch  2023.2.60-1.0.amzn2023.0.1  amazonlinux 828 k
 cloud-init               noarch  22.2.2-1.amzn2023.1.7       amazonlinux 1.1 M

         ... [ list edited for clarity ]

 system-release           noarch  2023.0.20230329-0.amzn2023  amazonlinux  29 k

         ... [ list edited for clarity ]

 vim-data                 noarch  2:9.0.1403-1.amzn2023.0.1   amazonlinux  25 k
 vim-minimal              aarch64 2:9.0.1403-1.amzn2023.0.1   amazonlinux 753 k

Transaction Summary
================================================================================
Install    1 Package
Upgrade   42 Packages

Total download size: 56 M
```

由于 `--releasever` 选项优先于 `system-release` 和 `/etc/dnf/vars/releasever`，所以此升级的结果如下：

1. 升级将替换在先前版本和新版本之间有所变化的所有已安装的软件包。
2. 升级会将系统锁定到 `system-release` 的新版本的存储库。

通过始终指定要更新到什么`releasever`（即 AL2 023 版本），您可以对整个舰队进行一组确定性的更改。您启动了版本*A*，更新为*B*，然后更新为*C*。

选择性更新确定性升级系统

> **Note**
>
> 我们建议安装新版本中的所有更新，而不是选择特定的更新。仅将部分更新应用于操作系统应该是进行整个更新的标准做法的例外。

您或许想要安装来自最近版本的选定软件包，同时仍将系统锁定到原始发行版本。

您可以使用 `dnf check-update` 来确认要升级的软件包。

```
$ sudo dnf check-update --releasever=latest --security
Amazon Linux 2023 repository                      13 MB/s |  10 MB     00:00
Last metadata expiration check: 0:00:02 ago on Wed 15 Feb 2023 02:52:21 AM UTC.

bind-libs.aarch64                  32:9.16.27-1.amzn2023.0.1         amazonlinux
bind-license.noarch                32:9.16.27-1.amzn2023.0.1         amazonlinux
bind-utils.aarch64                 32:9.16.27-1.amzn2023.0.1         amazonlinux
cryptsetup.aarch64                 2.4.3-2.amzn2023.0.1              amazonlinux
cryptsetup-libs.aarch64            2.4.3-2.amzn2023.0.1              amazonlinux
curl-minimal.aarch64               7.85.0-1.amzn2023.0.1             amazonlinux
glibc.aarch64                      2.34-40.amzn2023.0.2              amazonlinux
glibc-all-langpacks.aarch64        2.34-40.amzn2023.0.2              amazonlinux
glibc-common.aarch64               2.34-40.amzn2023.0.2              amazonlinux
glibc-locale-source.aarch64        2.34-40.amzn2023.0.2              amazonlinux
gmp.aarch64                        1:6.2.1-2.amzn2023.0.1            amazonlinux
gnupg2-minimal.aarch64             2.3.7-1.amzn2023.0.2              amazonlinux
gzip.aarch64                       1.10-5.amzn2023.0.1               amazonlinux
kernel.aarch64                     6.1.12-17.42.amzn2023             amazonlinux
kernel-tools.aarch64               6.1.12-17.42.amzn2023             amazonlinux
libarchive.aarch64                 3.5.3-2.amzn2023.0.1              amazonlinux
libcurl-minimal.aarch64            7.85.0-1.amzn2023.0.1             amazonlinux
libsepol.aarch64                   3.4-3.amzn2023.0.2                amazonlinux
libsolv.aarch64                    0.7.22-1.amzn2023.0.1             amazonlinux
libxml2.aarch64                    2.9.14-1.amzn2023.0.1             amazonlinux
logrotate.aarch64                  3.20.1-2.amzn2023.0.2             amazonlinux
lua-libs.aarch64                   5.4.4-3.amzn2023.0.1              amazonlinux
lz4-libs.aarch64                   1.9.4-1.amzn2023.0.1              amazonlinux
openssl.aarch64                    1:3.0.5-1.amzn2023.0.3            amazonlinux
openssl-libs.aarch64               1:3.0.5-1.amzn2023.0.3            amazonlinux
pcre2.aarch64                      10.40-1.amzn2023.0.1              amazonlinux
```

```
pcre2-syntax.noarch                 10.40-1.amzn2023.0.1              amazonlinux
rsync.aarch64                       3.2.6-1.amzn2023.0.2              amazonlinux
vim-common.aarch64                  2:9.0.475-1.amzn2023.0.1          amazonlinux
vim-data.noarch                     2:9.0.475-1.amzn2023.0.1          amazonlinux
vim-enhanced.aarch64                2:9.0.475-1.amzn2023.0.1          amazonlinux
vim-filesystem.noarch               2:9.0.475-1.amzn2023.0.1          amazonlinux
vim-minimal.aarch64                 2:9.0.475-1.amzn2023.0.1          amazonlinux
xz.aarch64                          5.2.5-9.amzn2023.0.1              amazonlinux
xz-libs.aarch64                     5.2.5-9.amzn2023.0.1              amazonlinux
zlib.aarch64                        1.2.11-32.amzn2023.0.3            amazonlinux
```

安装要升级的软件包。使用 `sudo dnf upgrade --releasever=latest` 和软件包名称来确保 `system-release` 软件包保持不变。

```
$ sudo dnf upgrade --releasever=latest openssl openssl-libs
Last metadata expiration check: 0:01:28 ago on Wed 15 Feb 2023 02:52:21 AM UTC.
Dependencies resolved.
================================================================================
 Package          Arch        Version                     Repository       Size
================================================================================
Upgrading:
 openssl          aarch64     1:3.0.5-1.amzn2023.0.3      amazonlinux     1.1 M
 openssl-libs     aarch64     1:3.0.5-1.amzn2023.0.3      amazonlinux     2.1 M

Transaction Summary
================================================================================
Upgrade  2 Packages

Total download size: 3.2 M
```

> **Note**
>
> 使用 `sudo dnf upgrade --releasever=latest` 更新所有软件包，包括 `system-release`。然后，该版本仍锁定到新的 `system-release`，除非您设置了持久取代。

在确定性升级中使用持久取代

> **Note**
>
> 通过确定性更新，您可以将操作系统更改集成到您的 CI/CD 管道中。禁用确定性更新将无法在部署之前进行测试。

您可以通过将变量值设置为`--releasever=latest`，使用永久覆盖来解锁系统，而不必添加*`latest`*。通过始终使用`latest`，这会将 AL2 023 的行为还原为 AL2 更新模型，在这种模型中，对包管理器的任何调用都将始终查看最新版本，并且不会锁定到任何特定版本的操作系统。

> **Warning**
>
> 通过使用永久覆盖确定性更新来解锁软件包管理器，您冒着发现应用程序与生产环境中的操作系统更新之间可能存在的不兼容性的风险。
> 虽然不兼容的情况很少见，但是通过操作系统更新，您正在将新的代码更改集成到您的环境中，但是集成测试可以防止部署对生产环境产生负面影响的代码更改。

```
$ echo latest | sudo tee /etc/dnf/vars/releasever
latest
```

```
$ sudo dnf upgrade
Last metadata expiration check: 0:03:36 ago on Wed 15 Feb 2023 02:52:21 AM UTC.
Dependencies resolved.
================================================================================
 Package                 Arch    Version                      Repository   Size
================================================================================
Installing:
 kernel                  aarch64 6.1.73-45.135.amzn2023       amazonlinux  24 M
Upgrading:
 acl                     aarch64 2.3.1-2.amzn2023.0.1         amazonlinux  72 k
 alternatives            aarch64 1.15-2.amzn2023.0.1          amazonlinux  36 k
 amazon-ec2-net-utils    noarch  2.3.0-1.amzn2023.0.1         amazonlinux  16 k
 at                      aarch64 3.1.23-6.amzn2023.0.1        amazonlinux  60 k
 attr                    aarch64 2.5.1-3.amzn2023.0.1         amazonlinux  59 k
 audit                   aarch64 3.0.6-1.amzn2023.0.1         amazonlinux 249 k
 audit-libs              aarch64 3.0.6-1.amzn2023.0.1         amazonlinux 116 k
 aws-c-auth-libs         aarch64 0.6.5-6.amzn2023.0.2         amazonlinux  79 k
```

```
 aws-c-cal-libs          aarch64 0.5.12-7.amzn2023.0.2        amazonlinux  34 k
 aws-c-common-libs       aarch64 0.6.14-6.amzn2023.0.2        amazonlinux 119 k
 aws-c-compression-libs  aarch64 0.2.14-5.amzn2023.0.2        amazonlinux  22 k
 aws-c-event-stream-libs aarch64 0.2.7-5.amzn2023.0.2         amazonlinux  47 k
 aws-c-http-libs         aarch64 0.6.8-6.amzn2023.0.2         amazonlinux 147 k
 aws-c-io-libs           aarch64 0.10.12-5.amzn2023.0.6       amazonlinux 109 k
 aws-c-mqtt-libs         aarch64 0.7.8-7.amzn2023.0.2         amazonlinux  61 k
 aws-c-s3-libs           aarch64 0.1.27-5.amzn2023.0.3        amazonlinux  54 k
 aws-c-sdkutils-libs     aarch64 0.1.1-5.amzn2023.0.2         amazonlinux  26 k
 aws-checksums-libs      aarch64 0.1.12-5.amzn2023.0.2        amazonlinux  50 k
 awscli-2                noarch  2.7.8-1.amzn2023.0.4         amazonlinux 7.3 M
 basesystem              noarch  11-11.amzn2023.0.1           amazonlinux 7.8 k
 bash                    aarch64 5.1.8-2.amzn2023.0.1         amazonlinux 1.6 M
 bash-completion         noarch  1:2.11-2.amzn2023.0.1        amazonlinux 292 k
 bc                      aarch64 1.07.1-14.amzn2023.0.1       amazonlinux 120 k
 bind-libs               aarch64 32:9.16.27-1.amzn2023.0.1    amazonlinux 1.2 M
 bind-license            noarch  32:9.16.27-1.amzn2023.0.1    amazonlinux  14 k
 bind-utils              aarch64 32:9.16.27-1.amzn2023.0.1    amazonlinux 206 k
 binutils                aarch64 2.38-20.amzn2023.0.3         amazonlinux 4.6 M
 boost-filesystem        aarch64 1.75.0-4.amzn2023.0.1        amazonlinux  55 k
 boost-system            aarch64 1.75.0-4.amzn2023.0.1        amazonlinux  14 k
 boost-thread            aarch64 1.75.0-4.amzn2023.0.1        amazonlinux  54 k
 bzip2                   aarch64 1.0.8-6.amzn2023.0.1         amazonlinux  53 k
 bzip2-libs              aarch64 1.0.8-6.amzn2023.0.1         amazonlinux  44 k
 c-ares                  aarch64 1.17.2-1.amzn2023.0.1        amazonlinux 107 k
 ca-certificates         noarch  2021.2.50-1.0.amzn2023.0.3   amazonlinux 343 k
 checkpolicy             aarch64 3.4-3.amzn2023.0.1           amazonlinux 345 k
 chkconfig               aarch64 1.15-2.amzn2023.0.1          amazonlinux 162 k
 chrony                  aarch64 4.2-7.amzn2023.0.4           amazonlinux 314 k
 cloud-init              noarch  22.2.2-1.amzn2023.1.7        amazonlinux 1.1 M
 cloud-utils-growpart    aarch64 0.31-8.amzn2023.0.2          amazonlinux  31 k
 coreutils               aarch64 8.32-30.amzn2023.0.2         amazonlinux 1.1 M
 coreutils-common        aarch64 8.32-30.amzn2023.0.2         amazonlinux 2.0 M
 cpio                    aarch64 2.13-10.amzn2023.0.1         amazonlinux 269 k
 cracklib                aarch64 2.9.6-27.amzn2023.0.1        amazonlinux  83 k
 cracklib-dicts          aarch64 2.9.6-27.amzn2023.0.1        amazonlinux 3.6 M
 crontabs                noarch  1.11-24.20190603git.amzn2023.0.1
                                                              amazonlinux  19 k
 crypto-policies         noarch  20230128-1.gitdfb10ea.amzn2023.0.1
                                                              amazonlinux  61 k
 crypto-policies-scripts noarch  20230128-1.gitdfb10ea.amzn2023.0.1
                                                              amazonlinux  81 k
...
Installing dependencies:
```

```
 amazon-linux-repo-cdn   noarch  2023.0.20230210-0.amzn2023   amazonlinux  16 k
 xxhash-libs             aarch64 0.8.0-3.amzn2023.0.1         amazonlinux  32 k
Installing weak dependencies:
 amazon-chrony-config    noarch  4.2-7.amzn2023.0.4           amazonlinux  14 k
 gawk-all-langpacks      aarch64 5.1.0-3.amzn2023.0.1         amazonlinux 207 k

Transaction Summary
================================================================================
Install    5 Packages
Upgrade  413 Packages

Total download size: 199 M
```

> **Note**
>
> 如果您使用了取代变量 `/etc/dnf/vars/releasever`，请使用以下命令，通过擦除取代值来恢复默认锁定行为。
>
> ```
> $ sudo rm /etc/dnf/vars/releasever
> ```

使用永久替换使用`latest`而不是特定版本与的默认行为类似。 AL2有些服务是 AMIs基于这些服务来构建的， AL2 它们会禁用此行为，并锁定到特定的软件包版本，就像你在 AL2 023 上默认获得的那样。

与其禁用确定性更新，不如将实例替换为从新 AMI 启动的实例。如果无法选择更换实例，我们建议使用诸如 AWS Systems Manager Patch Manager 之类的工具来协调在队列中应用更新。EC2 Image Builder 还可以自动构建、修补和测试您自己从 AL2 023 个基础映像中 AMIs 派生的映像。您也可以使用接收有关新更新的通知它来触发自己的 AMI 构建管道。

`latest`在预生产环境中使用，然后使用部署到生产环境并`latest`不能防止操作系统更新和应用程序之间出现任何问题。新的 AL2 023 版本可能在任何时间点发布，因此生产`latest`中的所有使用都存在风险。

在 AL2 023 中管理软件包和操作系统更新

与之前版本的亚马逊 Linux 不同， AL2023 AMIs 被锁定到特定版本的亚马逊 Linux 存储库。要对 AL2 023 实例同时应用安全修复和错误修复，请更新 DNF 配置到最新的可用发行版本。或者，启动一个较新的 AL2 023 实例。

本节介绍如何管理 DNF 正在运行的实例上的软件包和存储库。它还描述了如何配置 DNF 通过用户数据脚本在启动时启用最新可用的 Amazon Linux 存储库。有关更多信息，请参阅 DNF 命令参考。

建议应用新 AL2 023 版本中的所有可用更新。只选择安全更新，或者只选择特定的更新应该是例外而不是规则。有关与安全公告特定实例相关的列表，请参阅列出适用的公告。有关仅安装与特定公告相关的更新的信息，请参阅就地应用安全更新。

> **Important**
>
> 如果您想报告漏洞或对 AWS 云服务或开源项目有安全疑虑，请使用 “漏洞报告” 页面与 AWS 安全部门联系

主题

- 查看可用的软件包更新
- 使用应用安全更新 DNF 和存储库版本
- （安全）更新后自动重启服务
- 何时需要重启才能应用安全更新？
- 启动已启用最新存储库版本的实例
- 获取软件包支持信息
- 使用检查较新的存储库版本 dnf check-release-update
- 添加、启用或禁用新存储库
- 使用 cloud-init 添加存储库

查看可用的软件包更新

您可以使用 `dnf check-update` 命令查看有关您的系统的任何更新。对于 AL2 023，我们建议您在命令中添加`--releasever=`*`version-number`*选项。

添加此选项时，DNF 还会检查更高版本的存储库是否有更新。例如，当您运行 `dnf check-update` 命令后，使用最新返回的版本作为 *`version-number`* 的值。

如果将实例更新为使用最新版本的存储库，则输出将包括所有要更新的软件包的列表。

> **Note**
>
> 如果您没有在 `dnf check-update` 命令中使用可选标志来指定发布版本，则只查看当前配置的存储库版本。这意味着不会查看存储库的更高版本中的软件包。

Updates in a specific version

在这个例子中，我们将看看如果我们启动了 2023.0.20230315 版本的容器，那么 2023.1.20230628 版本中有哪些可用的更新。

> **Note**
>
> 此示例使用 2023.0.20230315 和 2023.1.20230628 版本，但这些版本不是 023 的最新版本。有关包含最新安全更新的最新版本，请参阅 AL2 023 发行说明。AL2

在此示例中，我们将从 2023.0.202 30315 版本的容器镜像开始。

首先，我们从容器注册表中获取此容器镜像。末尾的表示特定版本的图像版本；此图像版本通常为零。`.0`

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
   2023.0.20230315.0: Pulling from amazonlinux/amazonlinux
b76f3b09316a: Pull complete
Digest: sha256:94e7183b0739140dbd5b639fb7600f0a2299cec5df8780c26d9cb409da5315a9
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.0.20230315.0
public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
```

我们现在可以在容器内生成一个外壳，我们将从中检查是否有更新。

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
   bash-5.2#
```

该`dnf check-update`命令现在用于检查 2023.1.202 30628 版本中可用的更新。

> **Note**
>
> 应用软件包更新是一项特权操作。尽管在容器中运行时通常不需要提升权限，但如果在非容器化环境（例如 Amazon EC2 实例）中运行，则无需提升权限即可检查更新。

```
$ dnf check-update --releasever=2023.1.20230628
Amazon Linux 2023 repository                      60 MB/s |  15 MB     00:00
Last metadata expiration check: 0:00:02 ago on Mon Jul 22 17:25:34 2024.

amazon-linux-repo-cdn.noarch        2023.1.20230628-0.amzn2023        amazonlinux
ca-certificates.noarch              2023.2.60-1.0.amzn2023.0.2        amazonlinux
curl-minimal.x86_64                 8.0.1-1.amzn2023                  amazonlinux
glib2.x86_64                        2.74.7-688.amzn2023.0.1           amazonlinux
glibc.x86_64                        2.34-52.amzn2023.0.3              amazonlinux
glibc-common.x86_64                 2.34-52.amzn2023.0.3              amazonlinux
glibc-minimal-langpack.x86_64       2.34-52.amzn2023.0.3              amazonlinux
gnupg2-minimal.x86_64               2.3.7-1.amzn2023.0.4              amazonlinux
keyutils-libs.x86_64                1.6.3-1.amzn2023                  amazonlinux
libcap.x86_64                       2.48-2.amzn2023.0.3               amazonlinux
libcurl-minimal.x86_64              8.0.1-1.amzn2023                  amazonlinux
libgcc.x86_64                       11.3.1-4.amzn2023.0.3             amazonlinux
libgomp.x86_64                      11.3.1-4.amzn2023.0.3             amazonlinux
libstdc++.x86_64                    11.3.1-4.amzn2023.0.3             amazonlinux
libxml2.x86_64                      2.10.4-1.amzn2023.0.1             amazonlinux
ncurses-base.noarch                 6.2-4.20200222.amzn2023.0.4       amazonlinux
ncurses-libs.x86_64                 6.2-4.20200222.amzn2023.0.4       amazonlinux
openssl-libs.x86_64                 1:3.0.8-1.amzn2023.0.3            amazonlinux
python3-rpm.x86_64                  4.16.1.3-12.amzn2023.0.6          amazonlinux
rpm.x86_64                          4.16.1.3-12.amzn2023.0.6          amazonlinux
rpm-build-libs.x86_64               4.16.1.3-12.amzn2023.0.6          amazonlinux
rpm-libs.x86_64                     4.16.1.3-12.amzn2023.0.6          amazonlinux
rpm-sign-libs.x86_64                4.16.1.3-12.amzn2023.0.6          amazonlinux
system-release.noarch               2023.1.20230628-0.amzn2023        amazonlinux
tzdata.noarch                       2023c-1.amzn2023.0.1              amazonlinux
bash-5.2#
```

`system-release`软件包的版本显示了`dnf upgrade`命令将更新到的版本，即命令中请求的 2023.1.20230628 版本。`dnf check-update --releasever=`*`2023.1.20230628`*

Updates in the latest version

在这个例子中，我们将看看如果我们启动了 202 3.4.20 240319 `latest` 版本的容器，那么在 AL2 023 版本中有哪些可用的更新。在撰写本文时，`latest`版本为 2023.5.20240708，因此本示例中列出的更新将以该版本为准。

> **Note**
>
> 此示例使用 2023.4.20240319 和 2023.5.20240 708 版本，后者是撰写本文时的最新版本。有关最新版本的更多信息，请参阅 AL2023 版本说明。

在此示例中，我们将从 2023.4.20 240319 版本的容器镜像开始。

首先，我们从容器注册表中获取此容器镜像。末尾的表示特定版本的图像版本。`.1`虽然图像版本通常为零，但此示例使用的是图像版本为一个的版本。

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
   2023.4.20240319.1: Pulling from amazonlinux/amazonlinux
6de065fda9a2: Pull complete
Digest: sha256:b4838c4cc9211d966b6ea158dacc9eda7433a16ba94436508c2d9f01f7658b4e
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.4.20240319.1
public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
```

我们现在可以在容器内生成一个外壳，我们将从中检查是否有更新。

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
   bash-5.2#
```

该`dnf check-update`命令现在用于检查该版本中可用的更新，在撰写本文时，该*latest*版本是 2023.5. 20 240708。

> **Note**
>
> 应用软件包更新是一项特权操作。尽管在容器中运行时通常不需要提升权限，但如果在非容器化环境（例如 Amazon EC2 实例）中运行，则无需提升权限即可检查更新。

```
$ dnf --releasever=latest check-update
```

```
Amazon Linux 2023 repository                       78 MB/s |  25 MB     00:00
Last metadata expiration check: 0:00:04 ago on Mon Jul 22 17:39:13 2024.

amazon-linux-repo-cdn.noarch        2023.5.20240708-1.amzn2023        amazonlinux
curl-minimal.x86_64                 8.5.0-1.amzn2023.0.4              amazonlinux
dnf.noarch                          4.14.0-1.amzn2023.0.5             amazonlinux
dnf-data.noarch                     4.14.0-1.amzn2023.0.5             amazonlinux
expat.x86_64                        2.5.0-1.amzn2023.0.4              amazonlinux
glibc.x86_64                        2.34-52.amzn2023.0.10             amazonlinux
glibc-common.x86_64                 2.34-52.amzn2023.0.10             amazonlinux
glibc-minimal-langpack.x86_64       2.34-52.amzn2023.0.10             amazonlinux
krb5-libs.x86_64                    1.21-3.amzn2023.0.4               amazonlinux
libblkid.x86_64                     2.37.4-1.amzn2023.0.4             amazonlinux
libcurl-minimal.x86_64              8.5.0-1.amzn2023.0.4              amazonlinux
libmount.x86_64                     2.37.4-1.amzn2023.0.4             amazonlinux
libnghttp2.x86_64                   1.59.0-3.amzn2023.0.1             amazonlinux
libsmartcols.x86_64                 2.37.4-1.amzn2023.0.4             amazonlinux
libuuid.x86_64                      2.37.4-1.amzn2023.0.4             amazonlinux
openssl-libs.x86_64                 1:3.0.8-1.amzn2023.0.12           amazonlinux
python3.x86_64                      3.9.16-1.amzn2023.0.8             amazonlinux
python3-dnf.noarch                  4.14.0-1.amzn2023.0.5             amazonlinux
python3-libs.x86_64                 3.9.16-1.amzn2023.0.8             amazonlinux
system-release.noarch               2023.5.20240708-1.amzn2023        amazonlinux
yum.noarch                          4.14.0-1.amzn2023.0.5             amazonlinux
bash-5.2#
```

`system-release`软件包的版本显示了`dnf upgrade`命令将更新到的版本。

对于此命令，如果有更新的软件包可用，则返回码为 100。如果没有更新的软件包可用，则返回码为 0。此外，输出中还会列出所有要更新的软件包。

使用应用安全更新 DNF 和存储库版本

新的软件包更新和安全更新仅适用于新的存储库版本。对于从较早的 AL2 023 AMI 版本启动的实例，必须先更新存储库版本，然后才能安装安全更新。`dnf check-release-update` 命令包括一个示例更新命令，可将系统上安装的所有软件包更新为较新存储库中的版本。

> **Note**
>
> 如果您没有在 `dnf check-update` 命令中使用可选标志来指定发布版本，则只查看当前配置的存储库版本。这意味着对任何更高版本的存储库中存在的已安装软件包的任何更新都不会被应用。

本节介绍推荐的升级路径，即应用所有可用更新，而不是选择单个更新或仅选择标记为安全更新的更新。通过应用所有更新，现有实例将移至与启动更新后的 AMI 相同的程序包集。这种一致性减少了队列中软件包版本的差异。有关应用特定更新的更多信息，请参阅就地应用安全更新。

Applying updates in a specific version

在此示例中，如果我们启动了 2023.0.20230315 版本的容器，我们将应用 2023.1.20230628 版本中可用的更新。

> **Note**
>
> 此示例使用 2023.0.20230315 和 2023.1.20230628 版本，但这些版本不是 023 的最新版本。有关包含最新安全更新的最新版本，请参阅 AL2 023 发行说明。AL2

在此示例中，我们将从 2023.0.202 30315 版本的容器镜像开始。

首先，我们从容器注册表中获取此容器镜像。末尾的表示特定版本的图像版本；此图像版本通常为零。`.0`

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
   2023.0.20230315.0: Pulling from amazonlinux/amazonlinux
b76f3b09316a: Pull complete
Digest: sha256:94e7183b0739140dbd5b639fb7600f0a2299cec5df8780c26d9cb409da5315a9
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.0.20230315.0
public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
```

我们现在可以在容器内生成一个外壳，然后从中应用更新。

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
   bash-5.2#
```

该`dnf upgrade`命令现在用于应用 2023.1.202 30628 版本中存在的所有更新。

> **Note**
>
> 应用软件包更新是一项特权操作。尽管在容器中运行时通常不需要提升权限，但如果在 Amazon EC2 实例等非容器化环境中运行，则需要以用户身份运行`dnf upgrade`命令。`root`这可以使用`sudo`或`su`命令来完成。

```
$ dnf upgrade --releasever=2023.1.20230628
Amazon Linux 2023 repository                      38 MB/s |  15 MB     00:00
Last metadata expiration check: 0:00:02 ago on Mon Jul 22 17:49:08 2024.
Dependencies resolved.
================================================================================
 Package                 Arch    Version                       Repository    Size
================================================================================
Upgrading:
 amazon-linux-repo-cdn   noarch  2023.1.20230628-0.amzn2023    amazonlinux   18 k
 ca-certificates         noarch  2023.2.60-1.0.amzn2023.0.2    amazonlinux  829 k
 curl-minimal            x86_64  8.0.1-1.amzn2023              amazonlinux  150 k
 glib2                   x86_64  2.74.7-688.amzn2023.0.1       amazonlinux  2.7 M
 glibc                   x86_64  2.34-52.amzn2023.0.3          amazonlinux  1.9 M
 glibc-common            x86_64  2.34-52.amzn2023.0.3          amazonlinux  307 k
 glibc-minimal-langpack  x86_64  2.34-52.amzn2023.0.3          amazonlinux   35 k
 gnupg2-minimal          x86_64  2.3.7-1.amzn2023.0.4          amazonlinux  421 k
 keyutils-libs           x86_64  1.6.3-1.amzn2023               amazonlinux   33 k
 libcap                  x86_64  2.48-2.amzn2023.0.3            amazonlinux   67 k
 libcurl-minimal         x86_64  8.0.1-1.amzn2023               amazonlinux  249 k
 libgcc                  x86_64  11.3.1-4.amzn2023.0.3          amazonlinux  105 k
 libgomp                 x86_64  11.3.1-4.amzn2023.0.3          amazonlinux  280 k
 libstdc++               x86_64  11.3.1-4.amzn2023.0.3          amazonlinux  744 k
 libxml2                 x86_64  2.10.4-1.amzn2023.0.1          amazonlinux  706 k
 ncurses-base            noarch  6.2-4.20200222.amzn2023.0.4    amazonlinux   60 k
 ncurses-libs            x86_64  6.2-4.20200222.amzn2023.0.4    amazonlinux  328 k
 openssl-libs            x86_64  1:3.0.8-1.amzn2023.0.3         amazonlinux  2.2 M
 python3-rpm             x86_64  4.16.1.3-12.amzn2023.0.6       amazonlinux   88 k
 rpm                     x86_64  4.16.1.3-12.amzn2023.0.6       amazonlinux  486 k
 rpm-build-libs          x86_64  4.16.1.3-12.amzn2023.0.6       amazonlinux   90 k
 rpm-libs                x86_64  4.16.1.3-12.amzn2023.0.6       amazonlinux  309 k
 rpm-sign-libs           x86_64  4.16.1.3-12.amzn2023.0.6       amazonlinux   21 k
 system-release          noarch  2023.1.20230628-0.amzn2023     amazonlinux   29 k
 tzdata                  noarch  2023c-1.amzn2023.0.1           amazonlinux  433 k
```

```
Transaction Summary
================================================================================
Upgrade  25 Packages

Total download size: 12 M
Is this ok [y/N]:
```

system-release软件包的版本显示了dnf upgrade命令将更新到的版本，即命令中请求的2023.1.20230628 版本。dnf upgrade --releasever=*2023.1.20230628*

默认情况下，dnf会要求您确认是否要应用更新。您可以通过使用-y标志来绕过此提示dnf。在本示例中，该dnf upgrade -y --releasever=*2023.1.20230628*命令在应用更新之前不会要求进行确认。这在脚本或其他自动化环境中很有用。

确认要应用更新后，即可dnf应用更新。

```
Is this ok [y/N]:y
   Downloading Packages:
(1/25): libcap-2.48-2.amzn2023.0.3.x86_64.rpm    1.5 MB/s |  67 kB     00:00
(2/25): python3-rpm-4.16.1.3-12.amzn2023.0.6.x86 2.1 MB/s |  88 kB     00:00
(3/25): libcurl-minimal-8.0.1-1.amzn2023.x86_64. 2.6 MB/s | 249 kB     00:00
(4/25): glib2-2.74.7-688.amzn2023.0.1.x86_64.rpm  26 MB/s | 2.7 MB     00:00
(5/25): glibc-minimal-langpack-2.34-52.amzn2023. 1.3 MB/s |  35 kB     00:00
(6/25): rpm-build-libs-4.16.1.3-12.amzn2023.0.6. 2.8 MB/s |  90 kB     00:00
(7/25): rpm-libs-4.16.1.3-12.amzn2023.0.6.x86_64 6.6 MB/s | 309 kB     00:00
(8/25): libgcc-11.3.1-4.amzn2023.0.3.x86_64.rpm  3.9 MB/s | 105 kB     00:00
(9/25): glibc-common-2.34-52.amzn2023.0.3.x86_64  11 MB/s | 307 kB     00:00
(10/25): glibc-2.34-52.amzn2023.0.3.x86_64.rpm    31 MB/s | 1.9 MB     00:00
(11/25): rpm-sign-libs-4.16.1.3-12.amzn2023.0.6. 877 kB/s |  21 kB     00:00
(12/25): gnupg2-minimal-2.3.7-1.amzn2023.0.4.x86  15 MB/s | 421 kB     00:00
(13/25): openssl-libs-3.0.8-1.amzn2023.0.3.x86_6  35 MB/s | 2.2 MB     00:00
(14/25): libxml2-2.10.4-1.amzn2023.0.1.x86_64.rp  14 MB/s | 706 kB     00:00
(15/25): curl-minimal-8.0.1-1.amzn2023.x86_64.rp 4.2 MB/s | 150 kB     00:00
(16/25): rpm-4.16.1.3-12.amzn2023.0.6.x86_64.rpm  11 MB/s | 486 kB     00:00
(17/25): libgomp-11.3.1-4.amzn2023.0.3.x86_64.rp 7.0 MB/s | 280 kB     00:00
(18/25): libstdc++-11.3.1-4.amzn2023.0.3.x86_64.  14 MB/s | 744 kB     00:00
(19/25): keyutils-libs-1.6.3-1.amzn2023.x86_64.r 1.6 MB/s |  33 kB     00:00
(20/25): ncurses-libs-6.2-4.20200222.amzn2023.0.  10 MB/s | 328 kB     00:00
(21/25): tzdata-2023c-1.amzn2023.0.1.noarch.rpm   11 MB/s | 433 kB     00:00
(22/25): amazon-linux-repo-cdn-2023.1.20230628-0 781 kB/s |  18 kB     00:00
(23/25): ca-certificates-2023.2.60-1.0.amzn2023.  16 MB/s | 829 kB     00:00
(24/25): system-release-2023.1.20230628-0.amzn20 1.5 MB/s |  29 kB     00:00
```

```
(25/25): ncurses-base-6.2-4.20200222.amzn2023.0. 3.1 MB/s |  60 kB     00:00
--------------------------------------------------------------------------------
Total                                            28 MB/s |  12 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Upgrading        : libgcc-11.3.1-4.amzn2023.0.3.x86_64                   1/50
  Running scriptlet: libgcc-11.3.1-4.amzn2023.0.3.x86_64                   1/50
  Upgrading        : system-release-2023.1.20230628-0.amzn2023.noarch       2/50
  Upgrading        : amazon-linux-repo-cdn-2023.1.20230628-0.amzn2023.no    3/50
  Upgrading        : ncurses-base-6.2-4.20200222.amzn2023.0.4.noarch        4/50
  Upgrading        : tzdata-2023c-1.amzn2023.0.1.noarch                     5/50
  Upgrading        : glibc-common-2.34-52.amzn2023.0.3.x86_64               6/50
  Running scriptlet: glibc-2.34-52.amzn2023.0.3.x86_64                      7/50
  Upgrading        : glibc-2.34-52.amzn2023.0.3.x86_64                      7/50
  Running scriptlet: glibc-2.34-52.amzn2023.0.3.x86_64                      7/50
  Upgrading        : glibc-minimal-langpack-2.34-52.amzn2023.0.3.x86_64     8/50
  Upgrading        : libcap-2.48-2.amzn2023.0.3.x86_64                      9/50
  Upgrading        : gnupg2-minimal-2.3.7-1.amzn2023.0.4.x86_64            10/50
  Upgrading        : libgomp-11.3.1-4.amzn2023.0.3.x86_64                  11/50
  Running scriptlet: ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch     12/50
  Upgrading        : ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch     12/50
  Running scriptlet: ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch     12/50
  Upgrading        : openssl-libs-1:3.0.8-1.amzn2023.0.3.x86_64            13/50
  Upgrading        : libcurl-minimal-8.0.1-1.amzn2023.x86_64               14/50
  Upgrading        : curl-minimal-8.0.1-1.amzn2023.x86_64                  15/50
  Upgrading        : rpm-libs-4.16.1.3-12.amzn2023.0.6.x86_64              16/50
  Upgrading        : rpm-4.16.1.3-12.amzn2023.0.6.x86_64                   17/50
  Upgrading        : rpm-build-libs-4.16.1.3-12.amzn2023.0.6.x86_64        18/50
  Upgrading        : rpm-sign-libs-4.16.1.3-12.amzn2023.0.6.x86_64         19/50
  Upgrading        : python3-rpm-4.16.1.3-12.amzn2023.0.6.x86_64           20/50
  Upgrading        : glib2-2.74.7-688.amzn2023.0.1.x86_64                  21/50
  Upgrading        : libxml2-2.10.4-1.amzn2023.0.1.x86_64                  22/50
  Upgrading        : libstdc++-11.3.1-4.amzn2023.0.3.x86_64                23/50
  Upgrading        : keyutils-libs-1.6.3-1.amzn2023.x86_64                 24/50
  Upgrading        : ncurses-libs-6.2-4.20200222.amzn2023.0.4.x86_64       25/50
  Cleanup          : glib2-2.73.2-680.amzn2023.0.3.x86_64                  26/50
  Cleanup          : libstdc++-11.3.1-4.amzn2023.0.2.x86_64                27/50
  Cleanup          : libxml2-2.10.3-2.amzn2023.0.1.x86_64                  28/50
  Cleanup          : python3-rpm-4.16.1.3-12.amzn2023.0.5.x86_64           29/50
  Cleanup          : rpm-build-libs-4.16.1.3-12.amzn2023.0.5.x86_64        30/50
```

```
  Cleanup          : rpm-sign-libs-4.16.1.3-12.amzn2023.0.5.x86_64         31/50
  Cleanup          : rpm-libs-4.16.1.3-12.amzn2023.0.5.x86_64              32/50
  Cleanup          : libcap-2.48-2.amzn2023.0.2.x86_64                     33/50
  Cleanup          : gnupg2-minimal-2.3.7-1.amzn2023.0.3.x86_64            34/50
  Cleanup          : ncurses-libs-6.2-4.20200222.amzn2023.0.3.x86_64       35/50
  Cleanup          : libgomp-11.3.1-4.amzn2023.0.2.x86_64                  36/50
  Cleanup          : rpm-4.16.1.3-12.amzn2023.0.5.x86_64                   37/50
  Cleanup          : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64             38/50
  Cleanup          : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64          39/50
  Cleanup          : openssl-libs-1:3.0.8-1.amzn2023.0.1.x86_64            40/50
  Cleanup          : keyutils-libs-1.6.1-2.amzn2023.0.2.x86_64             41/50
  Cleanup          : amazon-linux-repo-cdn-2023.0.20230315-1.amzn2023.no   42/50
  Cleanup          : system-release-2023.0.20230315-1.amzn2023.noarch      43/50
  Cleanup          : ca-certificates-2023.2.60-1.0.amzn2023.0.1.noarch     44/50
  Cleanup          : ncurses-base-6.2-4.20200222.amzn2023.0.3.noarch       45/50
  Cleanup          : glibc-minimal-langpack-2.34-52.amzn2023.0.2.x86_64    46/50
  Cleanup          : glibc-2.34-52.amzn2023.0.2.x86_64                     47/50
  Cleanup          : glibc-common-2.34-52.amzn2023.0.2.x86_64              48/50
  Cleanup          : tzdata-2022g-1.amzn2023.0.1.noarch                    49/50
  Cleanup          : libgcc-11.3.1-4.amzn2023.0.2.x86_64                   50/50
  Running scriptlet: libgcc-11.3.1-4.amzn2023.0.2.x86_64                   50/50
  Running scriptlet: ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch     50/50
  Running scriptlet: rpm-4.16.1.3-12.amzn2023.0.6.x86_64                   50/50
  Running scriptlet: libgcc-11.3.1-4.amzn2023.0.2.x86_64                   50/50
  Verifying        : libcurl-minimal-8.0.1-1.amzn2023.x86_64                1/50
  Verifying        : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64           2/50
  Verifying        : libcap-2.48-2.amzn2023.0.3.x86_64                      3/50
  Verifying        : libcap-2.48-2.amzn2023.0.2.x86_64                      4/50
  Verifying        : glib2-2.74.7-688.amzn2023.0.1.x86_64                   5/50
  Verifying        : glib2-2.73.2-680.amzn2023.0.3.x86_64                   6/50
  Verifying        : python3-rpm-4.16.1.3-12.amzn2023.0.6.x86_64            7/50
  Verifying        : python3-rpm-4.16.1.3-12.amzn2023.0.5.x86_64            8/50
  Verifying        : glibc-minimal-langpack-2.34-52.amzn2023.0.3.x86_64     9/50
  Verifying        : glibc-minimal-langpack-2.34-52.amzn2023.0.2.x86_64    10/50
  Verifying        : rpm-libs-4.16.1.3-12.amzn2023.0.6.x86_64              11/50
  Verifying        : rpm-libs-4.16.1.3-12.amzn2023.0.5.x86_64              12/50
  Verifying        : rpm-build-libs-4.16.1.3-12.amzn2023.0.6.x86_64        13/50
  Verifying        : rpm-build-libs-4.16.1.3-12.amzn2023.0.5.x86_64        14/50
  Verifying        : glibc-2.34-52.amzn2023.0.3.x86_64                     15/50
  Verifying        : glibc-2.34-52.amzn2023.0.2.x86_64                     16/50
  Verifying        : libgcc-11.3.1-4.amzn2023.0.3.x86_64                   17/50
  Verifying        : libgcc-11.3.1-4.amzn2023.0.2.x86_64                   18/50
  Verifying        : glibc-common-2.34-52.amzn2023.0.3.x86_64              19/50
  Verifying        : glibc-common-2.34-52.amzn2023.0.2.x86_64              20/50
```

```
  Verifying        : rpm-sign-libs-4.16.1.3-12.amzn2023.0.6.x86_64         21/50
  Verifying        : rpm-sign-libs-4.16.1.3-12.amzn2023.0.5.x86_64         22/50
  Verifying        : openssl-libs-1:3.0.8-1.amzn2023.0.3.x86_64            23/50
  Verifying        : openssl-libs-1:3.0.8-1.amzn2023.0.1.x86_64            24/50
  Verifying        : gnupg2-minimal-2.3.7-1.amzn2023.0.4.x86_64            25/50
  Verifying        : gnupg2-minimal-2.3.7-1.amzn2023.0.3.x86_64            26/50
  Verifying        : libxml2-2.10.4-1.amzn2023.0.1.x86_64                  27/50
  Verifying        : libxml2-2.10.3-2.amzn2023.0.1.x86_64                  28/50
  Verifying        : curl-minimal-8.0.1-1.amzn2023.x86_64                  29/50
  Verifying        : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64             30/50
  Verifying        : rpm-4.16.1.3-12.amzn2023.0.6.x86_64                   31/50
  Verifying        : rpm-4.16.1.3-12.amzn2023.0.5.x86_64                   32/50
  Verifying        : libstdc++-11.3.1-4.amzn2023.0.3.x86_64                33/50
  Verifying        : libstdc++-11.3.1-4.amzn2023.0.2.x86_64                34/50
  Verifying        : libgomp-11.3.1-4.amzn2023.0.3.x86_64                  35/50
  Verifying        : libgomp-11.3.1-4.amzn2023.0.2.x86_64                  36/50
  Verifying        : keyutils-libs-1.6.3-1.amzn2023.x86_64                 37/50
  Verifying        : keyutils-libs-1.6.1-2.amzn2023.0.2.x86_64             38/50
  Verifying        : ncurses-libs-6.2-4.20200222.amzn2023.0.4.x86_64       39/50
  Verifying        : ncurses-libs-6.2-4.20200222.amzn2023.0.3.x86_64       40/50
  Verifying        : ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch     41/50
  Verifying        : ca-certificates-2023.2.60-1.0.amzn2023.0.1.noarch     42/50
  Verifying        : tzdata-2023c-1.amzn2023.0.1.noarch                    43/50
  Verifying        : tzdata-2022g-1.amzn2023.0.1.noarch                    44/50
  Verifying        : amazon-linux-repo-cdn-2023.1.20230628-0.amzn2023.no   45/50
  Verifying        : amazon-linux-repo-cdn-2023.0.20230315-1.amzn2023.no   46/50
  Verifying        : system-release-2023.1.20230628-0.amzn2023.noarch      47/50
  Verifying        : system-release-2023.0.20230315-1.amzn2023.noarch      48/50
  Verifying        : ncurses-base-6.2-4.20200222.amzn2023.0.4.noarch       49/50
  Verifying        : ncurses-base-6.2-4.20200222.amzn2023.0.3.noarch       50/50

Upgraded:
  amazon-linux-repo-cdn-2023.1.20230628-0.amzn2023.noarch
  ca-certificates-2023.2.60-1.0.amzn2023.0.2.noarch
  curl-minimal-8.0.1-1.amzn2023.x86_64
  glib2-2.74.7-688.amzn2023.0.1.x86_64
  glibc-2.34-52.amzn2023.0.3.x86_64
  glibc-common-2.34-52.amzn2023.0.3.x86_64
  glibc-minimal-langpack-2.34-52.amzn2023.0.3.x86_64
  gnupg2-minimal-2.3.7-1.amzn2023.0.4.x86_64
  keyutils-libs-1.6.3-1.amzn2023.x86_64
  libcap-2.48-2.amzn2023.0.3.x86_64
  libcurl-minimal-8.0.1-1.amzn2023.x86_64
  libgcc-11.3.1-4.amzn2023.0.3.x86_64
```

```
  libgomp-11.3.1-4.amzn2023.0.3.x86_64
  libstdc++-11.3.1-4.amzn2023.0.3.x86_64
  libxml2-2.10.4-1.amzn2023.0.1.x86_64
  ncurses-base-6.2-4.20200222.amzn2023.0.4.noarch
  ncurses-libs-6.2-4.20200222.amzn2023.0.4.x86_64
  openssl-libs-1:3.0.8-1.amzn2023.0.3.x86_64
  python3-rpm-4.16.1.3-12.amzn2023.0.6.x86_64
  rpm-4.16.1.3-12.amzn2023.0.6.x86_64
  rpm-build-libs-4.16.1.3-12.amzn2023.0.6.x86_64
  rpm-libs-4.16.1.3-12.amzn2023.0.6.x86_64
  rpm-sign-libs-4.16.1.3-12.amzn2023.0.6.x86_64
  system-release-2023.1.20230628-0.amzn2023.noarch
  tzdata-2023c-1.amzn2023.0.1.noarch

  Complete!
bash-5.2#
```

Updates in the latest version

在此示例中，如果我们启动了 202 3.4.20 24 AL2 03 `latest` 19 版本的容器，我们将应用 023 版本中可用的更新。在撰写本文时，`latest`版本为 2023.5.20240708，因此本示例中列出的更新将以该版本为准。

> **Note**
>
> 此示例使用 2023.4.20240319 和 2023.5.20240 708 版本，后者是撰写本文时的最新版本。有关最新版本的更多信息，请参阅 AL2023 版本说明。

在此示例中，我们将从 2023.4.20 240319 版本的容器镜像开始。

首先，我们从容器注册表中获取此容器镜像。末尾的表示特定版本的图像版本。`.1`虽然图像版本通常为零，但此示例使用的是图像版本为一个的版本。

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
   2023.4.20240319.1: Pulling from amazonlinux/amazonlinux
6de065fda9a2: Pull complete
Digest: sha256:b4838c4cc9211d966b6ea158dacc9eda7433a16ba94436508c2d9f01f7658b4e
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.4.20240319.1
public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
```

我们现在可以在容器内生成一个外壳，然后从中应用更新。

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
   bash-5.2#
```

该dnf upgrade命令现在用于应用该版本中可用的更新，在撰写本文时，该*latest*版本是2023.5. 20 240708。

> **Note**
>
> 应用软件包更新是一项特权操作。尽管在容器中运行时通常不需要提升权限，但如果在 Amazon EC2 实例等非容器化环境中运行，则需要以用户身份运行dnf upgrade命令。root这可以使用sudo或su命令来完成。

默认情况下，dnf会要求您确认是否要应用更新。在这个例子中，我们通过使用-y标志来dnf绕过这个提示。

```
$ dnf -y --releasever=latest update
Amazon Linux 2023 repository                       75 MB/s |  25 MB     00:00
Last metadata expiration check: 0:00:04 ago on Mon Jul 22 18:00:10 2024.
Dependencies resolved.
================================================================================
 Package                 Arch    Version                      Repository    Size
================================================================================
Upgrading:
 amazon-linux-repo-cdn   noarch  2023.5.20240708-1.amzn2023   amazonlinux   17 k
 curl-minimal            x86_64  8.5.0-1.amzn2023.0.4         amazonlinux  160 k
 dnf                     noarch  4.14.0-1.amzn2023.0.5        amazonlinux  460 k
 dnf-data                noarch  4.14.0-1.amzn2023.0.5        amazonlinux   34 k
 expat                   x86_64  2.5.0-1.amzn2023.0.4         amazonlinux  117 k
 glibc                   x86_64  2.34-52.amzn2023.0.10        amazonlinux  1.9 M
 glibc-common            x86_64  2.34-52.amzn2023.0.10        amazonlinux  295 k
 glibc-minimal-langpack  x86_64  2.34-52.amzn2023.0.10        amazonlinux   23 k
 krb5-libs               x86_64  1.21-3.amzn2023.0.4          amazonlinux  758 k
 libblkid                x86_64  2.37.4-1.amzn2023.0.4        amazonlinux  105 k
 libcurl-minimal         x86_64  8.5.0-1.amzn2023.0.4         amazonlinux  275 k
 libmount                x86_64  2.37.4-1.amzn2023.0.4        amazonlinux  132 k
 libnghttp2              x86_64  1.59.0-3.amzn2023.0.1        amazonlinux   79 k
 libsmartcols            x86_64  2.37.4-1.amzn2023.0.4        amazonlinux   62 k
 libuuid                 x86_64  2.37.4-1.amzn2023.0.4        amazonlinux   26 k
 openssl-libs            x86_64  1:3.0.8-1.amzn2023.0.12      amazonlinux  2.2 M
```

```
 python3                 x86_64  3.9.16-1.amzn2023.0.8        amazonlinux   27 k
 python3-dnf             noarch  4.14.0-1.amzn2023.0.5        amazonlinux  409 k
 python3-libs            x86_64  3.9.16-1.amzn2023.0.8        amazonlinux  7.3 M
 system-release          noarch  2023.5.20240708-1.amzn2023   amazonlinux   28 k
 yum                     noarch  4.14.0-1.amzn2023.0.5        amazonlinux   32 k

 Transaction Summary
================================================================================
Upgrade  21 Packages

Total download size: 14 M
Downloading Packages:
(1/21): amazon-linux-repo-cdn-2023.5.20240708-1. 345 kB/s |  17 kB     00:00
(2/21): dnf-4.14.0-1.amzn2023.0.5.noarch.rpm     6.8 MB/s | 460 kB     00:00
(3/21): dnf-data-4.14.0-1.amzn2023.0.5.noarch.rp 1.6 MB/s |  34 kB     00:00
(4/21): expat-2.5.0-1.amzn2023.0.4.x86_64.rpm    4.6 MB/s | 117 kB     00:00
(5/21): glibc-2.34-52.amzn2023.0.10.x86_64.rpm    38 MB/s | 1.9 MB     00:00
(6/21): glibc-common-2.34-52.amzn2023.0.10.x86_6 8.8 MB/s | 295 kB     00:00
(7/21): glibc-minimal-langpack-2.34-52.amzn2023. 1.7 MB/s |  23 kB     00:00
(8/21): curl-minimal-8.5.0-1.amzn2023.0.4.x86_64 998 kB/s | 160 kB     00:00
(9/21): libblkid-2.37.4-1.amzn2023.0.4.x86_64.rp 4.1 MB/s | 105 kB     00:00
(10/21): krb5-libs-1.21-3.amzn2023.0.4.x86_64.rp  16 MB/s | 758 kB     00:00
(11/21): libmount-2.37.4-1.amzn2023.0.4.x86_64.r 7.9 MB/s | 132 kB     00:00
(12/21): libnghttp2-1.59.0-3.amzn2023.0.1.x86_64 5.6 MB/s |  79 kB     00:00
(13/21): libsmartcols-2.37.4-1.amzn2023.0.4.x86_ 4.4 MB/s |  62 kB     00:00
(14/21): libcurl-minimal-8.5.0-1.amzn2023.0.4.x8 7.1 MB/s | 275 kB     00:00
(15/21): libuuid-2.37.4-1.amzn2023.0.4.x86_64.rp 1.1 MB/s |  26 kB     00:00
(16/21): python3-3.9.16-1.amzn2023.0.8.x86_64.rp 1.5 MB/s |  27 kB     00:00
(17/21): python3-dnf-4.14.0-1.amzn2023.0.5.noarc  19 MB/s | 409 kB     00:00
(18/21): system-release-2023.5.20240708-1.amzn20 1.9 MB/s |  28 kB     00:00
(19/21): yum-4.14.0-1.amzn2023.0.5.noarch.rpm    1.6 MB/s |  32 kB     00:00
(20/21): openssl-libs-3.0.8-1.amzn2023.0.12.x86_  26 MB/s | 2.2 MB     00:00
(21/21): python3-libs-3.9.16-1.amzn2023.0.8.x86_  59 MB/s | 7.3 MB     00:00
--------------------------------------------------------------------------------
Total                                             34 MB/s |  14 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Upgrading        : glibc-common-2.34-52.amzn2023.0.10.x86_64             1/42
  Upgrading        : glibc-minimal-langpack-2.34-52.amzn2023.0.10.x86_64   2/42
  Running scriptlet: glibc-2.34-52.amzn2023.0.10.x86_64                    3/42
```

```
  Upgrading        : glibc-2.34-52.amzn2023.0.10.x86_64                      3/42
  Running scriptlet: glibc-2.34-52.amzn2023.0.10.x86_64                      3/42
  Upgrading        : libuuid-2.37.4-1.amzn2023.0.4.x86_64                    4/42
  Upgrading        : openssl-libs-1:3.0.8-1.amzn2023.0.12.x86_64             5/42
  Upgrading        : krb5-libs-1.21-3.amzn2023.0.4.x86_64                    6/42
  Upgrading        : libblkid-2.37.4-1.amzn2023.0.4.x86_64                   7/42
  Running scriptlet: libblkid-2.37.4-1.amzn2023.0.4.x86_64                   7/42
  Upgrading        : expat-2.5.0-1.amzn2023.0.4.x86_64                       8/42
  Upgrading        : python3-3.9.16-1.amzn2023.0.8.x86_64                    9/42
  Upgrading        : python3-libs-3.9.16-1.amzn2023.0.8.x86_64              10/42
  Upgrading        : libnghttp2-1.59.0-3.amzn2023.0.1.x86_64                11/42
  Upgrading        : libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64            12/42
  Upgrading        : system-release-2023.5.20240708-1.amzn2023.noarch       13/42
  Upgrading        : amazon-linux-repo-cdn-2023.5.20240708-1.amzn2023.no    14/42
  Upgrading        : dnf-data-4.14.0-1.amzn2023.0.5.noarch                  15/42
  Upgrading        : python3-dnf-4.14.0-1.amzn2023.0.5.noarch               16/42
  Upgrading        : dnf-4.14.0-1.amzn2023.0.5.noarch                       17/42
  Running scriptlet: dnf-4.14.0-1.amzn2023.0.5.noarch                       17/42
  Upgrading        : yum-4.14.0-1.amzn2023.0.5.noarch                       18/42
  Upgrading        : curl-minimal-8.5.0-1.amzn2023.0.4.x86_64               19/42
  Upgrading        : libmount-2.37.4-1.amzn2023.0.4.x86_64                  20/42
  Upgrading        : libsmartcols-2.37.4-1.amzn2023.0.4.x86_64              21/42
  Cleanup          : yum-4.14.0-1.amzn2023.0.4.noarch                       22/42
  Running scriptlet: dnf-4.14.0-1.amzn2023.0.4.noarch                       23/42
  Cleanup          : dnf-4.14.0-1.amzn2023.0.4.noarch                       23/42
  Running scriptlet: dnf-4.14.0-1.amzn2023.0.4.noarch                       23/42
  Cleanup          : python3-dnf-4.14.0-1.amzn2023.0.4.noarch               24/42
  Cleanup          : amazon-linux-repo-cdn-2023.4.20240319-1.amzn2023.no    25/42
  Cleanup          : libmount-2.37.4-1.amzn2023.0.3.x86_64                  26/42
  Cleanup          : curl-minimal-8.5.0-1.amzn2023.0.2.x86_64               27/42
  Cleanup          : libcurl-minimal-8.5.0-1.amzn2023.0.2.x86_64            28/42
  Cleanup          : krb5-libs-1.21-3.amzn2023.0.3.x86_64                   29/42
  Cleanup          : libblkid-2.37.4-1.amzn2023.0.3.x86_64                  30/42
  Cleanup          : libnghttp2-1.57.0-1.amzn2023.0.1.x86_64                31/42
  Cleanup          : libsmartcols-2.37.4-1.amzn2023.0.3.x86_64              32/42
  Cleanup          : system-release-2023.4.20240319-1.amzn2023.noarch       33/42
  Cleanup          : dnf-data-4.14.0-1.amzn2023.0.4.noarch                  34/42
  Cleanup          : python3-3.9.16-1.amzn2023.0.6.x86_64                   35/42
  Cleanup          : python3-libs-3.9.16-1.amzn2023.0.6.x86_64              36/42
  Cleanup          : openssl-libs-1:3.0.8-1.amzn2023.0.11.x86_64            37/42
  Cleanup          : libuuid-2.37.4-1.amzn2023.0.3.x86_64                   38/42
  Cleanup          : expat-2.5.0-1.amzn2023.0.3.x86_64                      39/42
  Cleanup          : glibc-2.34-52.amzn2023.0.8.x86_64                      40/42
  Cleanup          : glibc-minimal-langpack-2.34-52.amzn2023.0.8.x86_64     41/42
```

```
  Cleanup           : glibc-common-2.34-52.amzn2023.0.8.x86_64              42/42
  Running scriptlet: glibc-common-2.34-52.amzn2023.0.8.x86_64              42/42
  Verifying         : amazon-linux-repo-cdn-2023.5.20240708-1.amzn2023.no    1/42
  Verifying         : amazon-linux-repo-cdn-2023.4.20240319-1.amzn2023.no    2/42
  Verifying         : curl-minimal-8.5.0-1.amzn2023.0.4.x86_64               3/42
  Verifying         : curl-minimal-8.5.0-1.amzn2023.0.2.x86_64               4/42
  Verifying         : dnf-4.14.0-1.amzn2023.0.5.noarch                       5/42
  Verifying         : dnf-4.14.0-1.amzn2023.0.4.noarch                       6/42
  Verifying         : dnf-data-4.14.0-1.amzn2023.0.5.noarch                  7/42
  Verifying         : dnf-data-4.14.0-1.amzn2023.0.4.noarch                  8/42
  Verifying         : expat-2.5.0-1.amzn2023.0.4.x86_64                      9/42
  Verifying         : expat-2.5.0-1.amzn2023.0.3.x86_64                     10/42
  Verifying         : glibc-2.34-52.amzn2023.0.10.x86_64                    11/42
  Verifying         : glibc-2.34-52.amzn2023.0.8.x86_64                     12/42
  Verifying         : glibc-common-2.34-52.amzn2023.0.10.x86_64             13/42
  Verifying         : glibc-common-2.34-52.amzn2023.0.8.x86_64              14/42
  Verifying         : glibc-minimal-langpack-2.34-52.amzn2023.0.10.x86_64   15/42
  Verifying         : glibc-minimal-langpack-2.34-52.amzn2023.0.8.x86_64    16/42
  Verifying         : krb5-libs-1.21-3.amzn2023.0.4.x86_64                  17/42
  Verifying         : krb5-libs-1.21-3.amzn2023.0.3.x86_64                  18/42
  Verifying         : libblkid-2.37.4-1.amzn2023.0.4.x86_64                 19/42
  Verifying         : libblkid-2.37.4-1.amzn2023.0.3.x86_64                 20/42
  Verifying         : libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64           21/42
  Verifying         : libcurl-minimal-8.5.0-1.amzn2023.0.2.x86_64           22/42
  Verifying         : libmount-2.37.4-1.amzn2023.0.4.x86_64                 23/42
  Verifying         : libmount-2.37.4-1.amzn2023.0.3.x86_64                 24/42
  Verifying         : libnghttp2-1.59.0-3.amzn2023.0.1.x86_64               25/42
  Verifying         : libnghttp2-1.57.0-1.amzn2023.0.1.x86_64               26/42
  Verifying         : libsmartcols-2.37.4-1.amzn2023.0.4.x86_64             27/42
  Verifying         : libsmartcols-2.37.4-1.amzn2023.0.3.x86_64             28/42
  Verifying         : libuuid-2.37.4-1.amzn2023.0.4.x86_64                  29/42
  Verifying         : libuuid-2.37.4-1.amzn2023.0.3.x86_64                  30/42
  Verifying         : openssl-libs-1:3.0.8-1.amzn2023.0.12.x86_64           31/42
  Verifying         : openssl-libs-1:3.0.8-1.amzn2023.0.11.x86_64           32/42
  Verifying         : python3-3.9.16-1.amzn2023.0.8.x86_64                  33/42
  Verifying         : python3-3.9.16-1.amzn2023.0.6.x86_64                  34/42
  Verifying         : python3-dnf-4.14.0-1.amzn2023.0.5.noarch              35/42
  Verifying         : python3-dnf-4.14.0-1.amzn2023.0.4.noarch              36/42
  Verifying         : python3-libs-3.9.16-1.amzn2023.0.8.x86_64             37/42
  Verifying         : python3-libs-3.9.16-1.amzn2023.0.6.x86_64             38/42
  Verifying         : system-release-2023.5.20240708-1.amzn2023.noarch      39/42
  Verifying         : system-release-2023.4.20240319-1.amzn2023.noarch      40/42
  Verifying         : yum-4.14.0-1.amzn2023.0.5.noarch                      41/42
  Verifying         : yum-4.14.0-1.amzn2023.0.4.noarch                      42/42
```

```
Upgraded:
  amazon-linux-repo-cdn-2023.5.20240708-1.amzn2023.noarch
  curl-minimal-8.5.0-1.amzn2023.0.4.x86_64
  dnf-4.14.0-1.amzn2023.0.5.noarch
  dnf-data-4.14.0-1.amzn2023.0.5.noarch
  expat-2.5.0-1.amzn2023.0.4.x86_64
  glibc-2.34-52.amzn2023.0.10.x86_64
  glibc-common-2.34-52.amzn2023.0.10.x86_64
  glibc-minimal-langpack-2.34-52.amzn2023.0.10.x86_64
  krb5-libs-1.21-3.amzn2023.0.4.x86_64
  libblkid-2.37.4-1.amzn2023.0.4.x86_64
  libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64
  libmount-2.37.4-1.amzn2023.0.4.x86_64
  libnghttp2-1.59.0-3.amzn2023.0.1.x86_64
  libsmartcols-2.37.4-1.amzn2023.0.4.x86_64
  libuuid-2.37.4-1.amzn2023.0.4.x86_64
  openssl-libs-1:3.0.8-1.amzn2023.0.12.x86_64
  python3-3.9.16-1.amzn2023.0.8.x86_64
  python3-dnf-4.14.0-1.amzn2023.0.5.noarch
  python3-libs-3.9.16-1.amzn2023.0.8.x86_64
  system-release-2023.5.20240708-1.amzn2023.noarch
  yum-4.14.0-1.amzn2023.0.5.noarch

Complete!
bash-5.2#
```

要发现 AL2 023 更新，请执行以下一项或多项操作：

- 运行 `dnf check-update` 命令。这将检查您锁定的 Amazon Linux 版本中是否有任何未应用的更新。如果您仅更新了`system-release`软件包，移动了实例锁定的存储库版本，但未应用其中任何可用的更新，则可能会显示更新。
- 订阅 Amazon Linux 存储库更新 SNS 主题 (`arn:aws:sns:us-east-1:137112412989:amazon-linux-2023-ami-updates`)。有关更多信息，请参阅《Amazon Simple Notification Service 开发人员指南》中的订阅 Amazon SNS 主题。
- 定期参阅 AL2023 发行说明。
- 通过以下方式发现新版本使用检查较新的存储库版本 `dnf check-release-update`.

> **Important**
>
> 包含安全更新的 AL2 023 新版本经常发布。请务必及时更新相关的安全补丁。

（安全）更新后自动重启服务

亚马逊 Linux 现在附带了智能重启套餐。 `Smart-restart`每当使用系统包管理器安装或删除软件包时，都会在系统更新时重新启动 systemd 服务。每当执行时都会`dnf （update|upgrade|downgrade)`发生这种情况。

`Smart-restart`使用来自的`needs-restarting`软件包`dnf-utils`和自定义的拒绝列表机制来确定需要重新启动哪些服务以及是否建议重新启动系统。如果建议重新启动系统，则会生成重启提示标记文件 (`/run/smart-restart/reboot-hint-marker`)。

要安装 **`smart-restart`**，请执行以下操作

运行以下命令 DNF 命令（就像你对待任何其他软件包一样）。

```
$ sudo dnf install smart-restart
```

安装完成后，后续事务将触发`smart-restart`逻辑。

拒绝名单

`Smart-restart`可以指示阻止某些服务重新启动。被封锁的服务不会影响是否需要重启的决定。要阻止其他服务，请添加一个带有后缀`-denylist`的文件`/etc/smart-restart-conf.d/`，如以下示例所示。

```
$ cat /etc/smart-restart-conf.d/custom-denylist
# Some comments
myservice.service
```

> **Note**
>
> 在决定是否需要重新启动时，会读取和评估所有`*-denylist`文件。

自定义挂钩

除了 denylisting 之外，还`smart-restart`提供了一种在尝试重启服务之前和之后运行自定义脚本的机制。自定义脚本可用于手动执行准备步骤或将剩余或已完成的重启通知其他组件。

所有`/etc/smart-restart-conf.d/`带有`-pre-restart`或后缀的脚本`-post-restart`都会被执行。如果顺序很重要，请在所有脚本前面加上一个数字以确保执行顺序，如以下示例所示。

```
$ ls /etc/smart-restart-conf.d/*-pre-restart
001-my-script-pre-restart
002-some-other-script-pre-restart
```

何时需要重启才能应用安全更新？

在某些情况下，Amazon Linux 需要重启才能应用更新：

- Linux 内核软件包的更新需要重新启动才能激活包含最新安全更新的新内核。内核livepatching可能允许您在有限的时间内推迟安全更新。详情请咨询023 上的 Kernel Live AL2 补丁。
- 在 M EC2 etal 实例上，Amazon Linux 提供微码更新（通过英特尔的`microcode_ctl`软件包 CPUs 和 AMD 的`amd-ucode-firmware`软件包） CPUs。这些微码更新将仅在后续实例重启时激活。对于虚拟化 EC2 实例，底层 AWS Nitro 系统会为您处理微码更新。
- 某些正在运行的 systemd 服务只有在系统完全重启后才能正常运行。该`smart-restart`机制将通过留下重启提示来通知您此类情况。请参阅（安全）更新后自动重启服务。

启动已启用最新存储库版本的实例

你可以添加 DNF 向用户数据脚本发出的命令来控制什么 RPM 当亚马逊 Linux AMI 启动时，软件包就会安装在它上。在以下示例中，使用用户数据脚本来确保通过其启动的任何实例都安装了同样的软件包更新。

```
#!/bin/bash
dnf upgrade --releasever=2023.0.20230210
# Additional setup and install commands below
dnf install httpd php7.4 mysql80
```

您必须以超级用户（根用户）的身份来运行此脚本。为此，请运行以下命令。

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

有关更多信息，请参阅 Amazon EC2 用户指南中的用户数据和 shell 脚本。

> **Note**
>
> 您也可以不使用用户数据脚本，而是启动最新的 Amazon Linux AMI 或基于 Amazon Linux AMI 的自定义 AMI。最新的 Amazon Linux AMI 已安装了所有必要的更新，并且被配置为指向特定的存储库版本。

获取软件包支持信息

AL2023 包含许多不同的开源软件项目。这些项目中的每一个都独立于 Amazon Linux 进行管理，并且有不同的发布和 end-of-support时间表。为了向您提供有关这些不同软件包的 Amazon Linux 特定信息，DNF `supportinfo`插件提供有关软件包的元数据。在以下示例中，**`dnf supportinfo`** 命令返回 `glibc` 软件包的元数据。

```
$ sudo dnf supportinfo --pkg glibc
Last metadata expiration check: 0:07:56 ago on Wed Mar  1 23:21:49 2023.
Name                 : glibc
Version              : 2.34-52.amzn2023.0.2
State                : installed
Support Status       : supported
Support Periods      : from 2023-03-15      : supported
                     : from 2028-03-15      : unsupported
Support Statement    : Amazon Linux 2023 End Of Life
Link                 : https://aws.amazon.com/amazon-linux-ami/faqs/
Other Info           : This is the support statement for AL2023. The
     ...: end of life of Amazon Linux 2023 would be March 2028.
     ...: From this point, the Amazon Linux 2023 packages (listed
  ...: below) will no longer, receive any updates from AWS.
```

Package 支持信息也可在 AL2023 发行说明的支持声明部分中找到。

使用检查较新的存储库版本 `dnf check-release-update`

在 AL2 023 实例中，你可以使用 DNF 用于管理存储库并应用更新的实用程序 RPM 包裹。可以从 Amazon Linux 存储库获得这些软件包。您可以使用 DNF 命令`dnf check-release-update`来检查是否有新版本的 DNF 存储库。

> **Note**
>
> AL2默认情况下，023 容器镜像不包含该`dnf check-release-update`命令。

```
$ dnf check-release-update
No such command: check-release-update. Please use /usr/bin/dnf --help
It could be a DNF plugin command, try: "dnf install 'dnf-command(check-release-
update)'"
```

运行`dnf install 'dnf-command(check-release-update)'`时，dnf将安装提供`check-release-update`命令的软件包，即`dnf-plugin-release-notification`软件包。在下面的示例中，给出了`-q`参数`dnf`以使其具有安静的输出。

```
$ dnf -y -q install 'dnf-command(check-release-update)'
Installed:
  dnf-plugin-release-notification-1.2-1.amzn2023.0.2.noarch
```

在 Amazon EC2 实例等非容器化环境中，默认包含该`check-release-update`命令。

```
$ sudo dnf check-release-update
WARNING:
  A newer release of "Amazon Linux" is available.

  Available Versions:

  Version 2023.0.20230210:
    Run the following command to update to 2023.0.20230210:

      dnf upgrade --releasever=2023.0.20230210

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

这将返回所有较新版本的完整列表 DNF 可用的存储库。如果没有返回任何内容，这意味着 DNF 当前配置为使用最新的可用版本。当前安装的`system-release`软件包的版本设置了 `releasever` DNF 变量。要查看当前存储库版本，请运行以下命令。

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

当你跑步的时候 DNF 软件包事务（例如安装、更新或删除命令），则会显示一条警告消息，通知您任何新的存储库版本。例如，如果您将`httpd`软件包安装在从旧版本 AL2 023 启动的实例上，则会返回以下输出。

```
$ sudo dnf install httpd -y
Last metadata expiration check: 0:16:52 ago on Wed Mar  1 23:21:49 2023.
Dependencies resolved.
=================================================================
 Package            Arch   Version                Repository   Size
=================================================================
Installing:
 httpd              x86_64 2.4.54-3.amzn2023.0.4  amazonlinux  46 k
Installing dependencies:
 apr                x86_64 1.7.2-2.amzn2023.0.2   amazonlinux 129 k
 apr-util           x86_64 1.6.3-1.amzn2023.0.1   amazonlinux  98 k
 generic-logos-httpd
                    noarch 18.0.0-12.amzn2023.0.3 amazonlinux  19 k
 httpd-core         x86_64 2.4.54-3.amzn2023.0.4  amazonlinux 1.3 M
 httpd-filesystem   noarch 2.4.54-3.amzn2023.0.4  amazonlinux  13 k
 httpd-tools        x86_64 2.4.54-3.amzn2023.0.4  amazonlinux  80 k
 libbrotli          x86_64 1.0.9-4.amzn2023.0.2   amazonlinux 315 k
 mailcap            noarch 2.1.49-3.amzn2023.0.3  amazonlinux  33 k
Installing weak dependencies:
 apr-util-openssl   x86_64 1.6.3-1.amzn2023.0.1   amazonlinux  17 k
 mod_http2          x86_64 1.15.24-1.amzn2023.0.3 amazonlinux 152 k
 mod_lua            x86_64 2.4.54-3.amzn2023.0.4  amazonlinux  60 k

Transaction Summary
=================================================================
Install  12 Packages

Total download size: 2.3 M
Installed size: 6.8 M
Downloading Packages:
(1/12): apr-util-openssl-1.6.3-1.am 212 kB/s |  17 kB     00:00
(2/12): apr-1.7.2-2.amzn2023.0.2.x8 1.1 MB/s | 129 kB     00:00
(3/12): httpd-core-2.4.54-3.amzn202 8.9 MB/s | 1.3 MB     00:00
(4/12): mod_http2-1.15.24-1.amzn202 1.9 MB/s | 152 kB     00:00
(5/12): apr-util-1.6.3-1.amzn2023.0 1.7 MB/s |  98 kB     00:00
(6/12): mod_lua-2.4.54-3.amzn2023.0 1.4 MB/s |  60 kB     00:00
(7/12): httpd-2.4.54-3.amzn2023.0.4 1.5 MB/s |  46 kB     00:00
(8/12): libbrotli-1.0.9-4.amzn2023. 4.4 MB/s | 315 kB     00:00
(9/12): mailcap-2.1.49-3.amzn2023.0 753 kB/s |  33 kB     00:00
(10/12): httpd-tools-2.4.54-3.amzn2 978 kB/s |  80 kB     00:00
(11/12): httpd-filesystem-2.4.54-3. 210 kB/s |  13 kB     00:00
(12/12): generic-logos-httpd-18.0.0 439 kB/s |  19 kB     00:00
-----------------------------------------------------------------
```

```
Total                                   6.6 MB/s | 2.3 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                1/1
  Installing       : apr-1.7.2-2.amzn2023.0.2.x86_64               1/12
  Installing       : apr-util-openssl-1.6.3-1.amzn2023.0.1.        2/12
  Installing       : apr-util-1.6.3-1.amzn2023.0.1.x86_64          3/12
  Installing       : mailcap-2.1.49-3.amzn2023.0.3.noarch          4/12
  Installing       : httpd-tools-2.4.54-3.amzn2023.0.4.x86_        5/12
  Installing       : generic-logos-httpd-18.0.0-12.amzn2023        6/12
  Running scriptlet: httpd-filesystem-2.4.54-3.amzn2023.0.4        7/12
  Installing       : httpd-filesystem-2.4.54-3.amzn2023.0.4        7/12
  Installing       : httpd-core-2.4.54-3.amzn2023.0.4.x86_6        8/12
  Installing       : mod_http2-1.15.24-1.amzn2023.0.3.x86_6        9/12
  Installing       : libbrotli-1.0.9-4.amzn2023.0.2.x86_64        10/12
  Installing       : mod_lua-2.4.54-3.amzn2023.0.4.x86_64         11/12
  Installing       : httpd-2.4.54-3.amzn2023.0.4.x86_64           12/12
  Running scriptlet: httpd-2.4.54-3.amzn2023.0.4.x86_64           12/12
  Verifying        : apr-1.7.2-2.amzn2023.0.2.x86_64                1/12
  Verifying        : apr-util-openssl-1.6.3-1.amzn2023.0.1.        2/12
  Verifying        : httpd-core-2.4.54-3.amzn2023.0.4.x86_6        3/12
  Verifying        : mod_http2-1.15.24-1.amzn2023.0.3.x86_6        4/12
  Verifying        : apr-util-1.6.3-1.amzn2023.0.1.x86_64          5/12
  Verifying        : mod_lua-2.4.54-3.amzn2023.0.4.x86_64          6/12
  Verifying        : libbrotli-1.0.9-4.amzn2023.0.2.x86_64         7/12
  Verifying        : httpd-2.4.54-3.amzn2023.0.4.x86_64            8/12
  Verifying        : httpd-tools-2.4.54-3.amzn2023.0.4.x86_        9/12
  Verifying        : mailcap-2.1.49-3.amzn2023.0.3.noarch         10/12
  Verifying        : httpd-filesystem-2.4.54-3.amzn2023.0.4       11/12
  Verifying        : generic-logos-httpd-18.0.0-12.amzn2023       12/12

Installed:
  apr-1.7.2-2.amzn2023.0.2.x86_64
  apr-util-1.6.3-1.amzn2023.0.1.x86_64
  apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
  httpd-2.4.54-3.amzn2023.0.4.x86_64
  httpd-core-2.4.54-3.amzn2023.0.4.x86_64
  httpd-filesystem-2.4.54-3.amzn2023.0.4.noarch
  httpd-tools-2.4.54-3.amzn2023.0.4.x86_64
  libbrotli-1.0.9-4.amzn2023.0.2.x86_64
```

```
  mailcap-2.1.49-3.amzn2023.0.3.noarch
  mod_http2-1.15.24-1.amzn2023.0.3.x86_64
  mod_lua-2.4.54-3.amzn2023.0.4.x86_64

Complete!
```

添加、启用或禁用新存储库

> **Warning**
>
> 仅添加设计用于 AL2 023 的存储库。
> 虽然为其他发行版设计的存储库现在可以使用，但不能保证它们会继续使用 023 中的任何软件包更新或不是为在 AL2 023 中使用而设计的存储库。 AL2

要从与默认 Amazon Linux 存储库不同的存储库中安装软件包，您需要配置DNF包管理系统以了解存储库的位置

要dnf了解软件包存储库，请将存储库信息添加到/etc/yum.repos.d/目录中该存储库的配置文件中。许多第三方存储库要么提供配置文件内容，要么提供包含配置文件的可安装软件包。

> **Note**
>
> 虽然可以直接在/etc/dnf/dnf.conf文件中配置存储库，但不建议这样做。建议在中将每个存储库配置在自己的文件中/etc/yum.repos.d/。

要了解当前启用了哪些存储库，可以运行以下命令：

```
$ dnf repolist all --verbose
Loaded plugins: builddep, changelog, config-manager, copr, debug, debuginfo-install,
 download, generate_completion_cache, groups-manager, needs-restarting, playground,
 release-notification, repoclosure, repodiff, repograph, repomanage, reposync,
 supportinfo
DNF version: 4.12.0
cachedir: /var/cache/dnf
Last metadata expiration check: 0:00:02 ago on Wed Mar  1 23:40:15 2023.
Repo-id            : amazonlinux
Repo-name          : Amazon Linux 2023 repository
Repo-status        : enabled
Repo-revision      : 1677203368
```

```
Repo-updated       : Fri Feb 24 01:49:28 2023
Repo-pkgs          : 12632
Repo-available-pkgs: 12632
Repo-size          : 12 G
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/x86_64/mirror.list
Repo-baseurl       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/guids/
cf9296325a6c46ff40c775a8e2d632c4c3fd9d9164014ce3304715d61b90ca8e/x86_64/
                   : (0 more)
Repo-expire        : 172800 second(s) (last: Wed Mar  1 23:40:15
                   : 2023)
Repo-filename      : /etc/yum.repos.d/amazonlinux.repo

Repo-id            : amazonlinux-debuginfo
Repo-name          : Amazon Linux 2023 repository - Debug
Repo-status        : disabled
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/debuginfo/x86_64/mirror.list
Repo-expire        : 21600 second(s) (last: unknown)
Repo-filename      : /etc/yum.repos.d/amazonlinux.repo

Repo-id            : amazonlinux-source
Repo-name          : Amazon Linux 2023 repository - Source packages
Repo-status        : disabled
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/core/mirrors/2023.0.20230222/SRPMS/mirror.list
Repo-expire        : 21600 second(s) (last: unknown)
Repo-filename      : /etc/yum.repos.d/amazonlinux.repo

Repo-id            : kernel-livepatch
Repo-name          : Amazon Linux 2023 Kernel Livepatch repository
Repo-status        : disabled
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/kernel-livepatch/mirrors/al2023/x86_64/mirror.list
Repo-expire        : 172800 second(s) (last: unknown)
Repo-filename      : /etc/yum.repos.d/kernel-livepatch.repo

Repo-id            : kernel-livepatch-source
Repo-name          : Amazon Linux 2023 Kernel Livepatch repository -
                   : Source packages
Repo-status        : disabled
Repo-mirrors       : https://al2023-repos-us-west-2-de612dc2.s3.dualstack.us-
west-2.amazonaws.com/kernel-livepatch/mirrors/al2023/SRPMS/mirror.list
```

```
Repo-expire        : 21600 second(s) (last: unknown)
Repo-filename      : /etc/yum.repos.d/kernel-livepatch.repo
Total packages: 12632
```

> **Note**
>
> 如果不添加 `--verbose` 选项标志，则输出仅包含 `Repo-id`、`Repo-name` 和 `Repo-status` 信息。

将 `yum` 存储库添加到 `/etc/yum.repos.d` 目录

1. 查找 `.repo` 文件的位置。在本示例中，`.repo` 文件位于 *https://www.example.com/repository.repo*。
2. 使用 `dnf config-manager` 命令添加存储库。

```
$ sudo dnf config-manager --add-repo https://www.example.com/repository.repo
Loaded plugins: priorities, update-motd, upgrade-helper
adding repo from: https://www.example.com/repository.repo
grabbing file https://www.example.com/repository.repo to /etc/
yum.repos.d/repository.repo
repository.repo                                          | 4.0 kB     00:00
repo saved to /etc/yum.repos.d/repository.repo
```

安装存储库后，必须按照以下过程启用存储库。

要在中启用yum存储库`/etc/yum.repos.d`，请使用带有`--enable`标志和*repository*名称的`dnf config-manager`命令。

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

> **Note**
>
> 要禁用存储库，请使用同样的命令语法，但在命令中将 `--enable` 替换为 `--disable`。

使用 cloud-init 添加存储库

除了使用前面的方法添加存储库，您还可以使用 `cloud-init` 框架添加新的存储库。

要添加新的软件包存储库，建议您使用以下模板。考虑将此文件保存在本地。

```
#cloud-config
yum_repos:
  repository.repo:
    baseurl: https://www.example.com/
    enabled: true
    gpgcheck: true
    gpgkey: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EXAMPLE
    name: Example Repository
```

> **Note**
>
> 使用 `cloud-init` 的一个好处是，您可以在配置文件中添加一个 `packages:` 部分。在该部分中，您可以包括要安装的软件包的名称。您可以安装来自默认存储库或来自您在 `cloud-config` 文件中添加的新存储库的软件包。
> 有关 YAML 文件结构的更多具体信息，请参阅《`cloud-init` 文档》中的添加 YUM 存储库。

设置 YAML 格式文件后，可以在 AWS CLI中在 `cloud-init` 框架中运行该文件。确保包含 `--userdata` 选项和 `.yml` 文件的名称，以便调用所需操作。

```
$ aws ec2 run-instances \
  --image-id \
    resolve:ssm:/aws/service/ami-amazon-linux-latest/al2023-ami-kernel-default-x86_64 \
  --instance-type m5.xlarge \
  --region us-east-1 \
  --key-name aws-key-us-east-1 \
  --security-group-ids sg-004a7650 \
  --user-data file://cloud-config.yml
```

023 上的 Kernel Live AL2 补丁

您可以使用适用于 AL2 023 的 Kernel Live Patching 将特定的安全漏洞和严重错误补丁应用于正在运行的 Linux 内核，而无需重新启动或中断正在运行的应用程序。此外，Kernel Live Patching 可以帮助提高应用程序的可用性，同时应用这些修复程序，直到系统可以重启为止。

AWS 为 AL2 023 发布了两种类型的内核实时补丁：

- 安全更新 - 包括 Linux 常见漏洞和风险（CVE）的更新。通常使用 Amazon Linux 安全公告评级将这些更新评为重要 或关键。它们通常对应于通用漏洞评分系统 (CVSS) 的 7 分或更高。在某些情况下， AWS 可能会在分配 CVE 之前提供更新。在这些情况下，补丁可能会显示为错误修复。
- 错误修复-包括针对与之无关的关键错误和稳定性问题的修复 CVEs。

AWS 在 AL2 023 内核版本发布后，为其提供最长 3 个月的内核实时补丁。三个月之后，您必须更新为更高的内核版本才能继续收到内核实时补丁。

AL2023 内核实时补丁以已签名的 RPM 包的形式在现有 AL2 023 存储库中提供。可以使用现有 DNF 软件包管理器工作流在单个实例上安装补丁。或者，也可以使用 S AWS ystems Manager 将它们安装在一组托管实例上。

AL2023 上提供的内核实时补丁无需支付额外费用。

主题

- 限制
- 支持的配置和先决条件
- 使用内核实时修补

限制

在应用内核实时补丁时，无法执行休眠以及使用高级调试工具（例如 `SystemTap`、`kprobes` 和基于 eBPF 的工具），或者访问内核实时修补基础设施使用的 `ftrace` 输出文件。

> **Note**
>
> 由于技术限制，某些问题无法通过实时修补来解决。因此，这些修复程序不会在内核实时补丁包中提供，而只能在原生内核包更新中提供。您可以像往常一样安装本机内核软件包并更新和重启系统以激活补丁。

支持的配置和先决条件

运行 AL2 023 的 Amazon EC2 实例和本地虚拟机支持内核实时补丁。

要在 AL2 023 上使用 Kernel Live Patching，必须使用以下内容：

- 64 位 x86_64 或 ARM64 架构

- 内核版本 6.1

策略要求

要从 AL2 023 存储库下载软件包，亚马逊 EC2 需要访问服务拥有的 Amazon S3 存储桶。如果您在环境中使用适用于 Amazon S3 的亚马逊虚拟私有云 (VPC) 终端节点，请确保您的 VPC 终端节点策略允许访问这些公有存储桶。下表描述了亚马逊 EC2 可能需要访问的 Amazon S3 存储桶，才能进行内核实时补丁。

S3 存储桶 ARN	描述
arn: aws: s3::: al2023-repos--de612dc2/* *region*	包含 AL2 023 个存储库的亚马逊 S3 存储桶

使用内核实时修补

要在单独的实例上启用和使用内核实时修补，可在该实例本身上使用命令行。要在一组托管实例上启用和使用内核实时修补，可以使用 AWS Systems Manager。

以下部分介绍如何在单独的实例上使用命令行来启用和使用内核实时修补。

有关在一组托管实例上启用和使用内核实时补丁的更多信息，请参阅《用户指南》中的 “在 AL2 023 实例上使用内核实时补丁”AWS Systems Manager 。

主题

- 启用内核实时修补
- 查看可用的内核实时补丁
- 应用内核实时补丁
- 查看应用的内核实时补丁
- 禁用内核实时修补

启用内核实时修补

默认情况下，内核实时补丁在 AL2 023 处于禁用状态。要使用内核实时修补，必须为其安装 DNF 插件，并启用实时修补功能。

启用内核实时修补

1. 内核实时补丁适用于内核版本6.1的 AL2 023。要检查内核版本，请运行以下命令。

```
$ sudo dnf list kernel
```

2. 为内核实时修补安装 DNF 插件。

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

3. 为内核实时修补启用 DNF 插件。

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

   此命令还会安装来自配置的存储库的最新版本的内核实时补丁 RPM。

4. 要确认内核实时修补的 DNF 插件是否安装成功，请运行以下命令。

   当您启用内核实时修补时，会自动应用空的内核实时补丁 RPM。如果成功启用了 Kernel Live Patching，则此命令将返回一个列表，其中包括初始的空内核实时补丁 RPM（以及另一个 RPM 设置包含 livepatch 的 DNF 存储库）。

```
$ sudo rpm -qa | grep kernel-livepatch
kernel-livepatch-repo-s3-2023.7.20250428-0.amzn2023.noarch
kernel-livepatch-6.1.134-150.224-1.0-0.amzn2023.x86_64
```

5. 安装 kpatch 软件包。

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

6. 如果之前安装过 kpatch 服务，请更新它。

```
$ sudo dnf upgrade kpatch-runtime
```

7. 启动 kpatch 服务。此服务在初始化或启动时会加载所有内核实时补丁。

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

查看可用的内核实时补丁

Amazon Linux 安全警报会发布到 Amazon Linux 安全中心。有关 AL2 023 安全警报的更多信息，包括内核实时补丁警报，请参阅 Amazon Linux 安全中心。内核实时补丁的前缀为 `ALASLIVEPATCH`。Amazon Linux 安全中心可能不会列出解决错误的内核实时补丁。

您还可以 CVEs 使用命令行查找可用的内核实时补丁以获取公告。

列出所有可用的内核实时补丁以获取公告

使用以下命令。

```
$ sudo dnf updateinfo list
Last metadata expiration check: 1:06:23 ago on Mon 13 Feb 2023 09:28:19 PM UTC.
ALAS2LIVEPATCH-2021-123   important/Sec. kernel-
livepatch-6.1.12-17.42-1.0-4.amzn2023.x86_64
ALAS2LIVEPATCH-2022-124   important/Sec. kernel-
livepatch-6.1.12-17.42-1.0-3.amzn2023.x86_64
```

列出所有可用的内核实时补丁 CVEs

使用以下命令。

```
$ sudo dnf updateinfo list cves
Last metadata expiration check: 1:07:26 ago on Mon 13 Feb 2023 09:28:19 PM UTC.
CVE-2022-0123    important/Sec. kernel-livepatch-6.1.12-17.42-1.0-4.amzn2023.x86_64
CVE-2022-3210    important/Sec. kernel-livepatch-6.1.12-17.42-1.0-3.amzn2023.x86_64
```

应用内核实时补丁

您可以使用 DNF 软件包管理器，就像应用常规更新那样来应用内核实时补丁。内核实时补丁的 DNF 插件管理可供应用的内核实时补丁。

> **Tip**
>
> 我们建议您使用 Kernel Live Patching 定期更新内核，以确保在系统重新启动之前，内核收到特定的重要和关键安全补丁。另请检查是否为无法作为实时补丁部署的本机内核软件包提供了其他修复程序，并在这些情况下更新并重启内核更新。

您可以选择应用特定的内核实时补丁，或者应用任何可用的内核实时补丁以及定期安全更新。

应用特定内核实时补丁

1. 使用 查看可用的内核实时补丁 中描述的命令之一获取内核实时补丁版本。
2. 为您的 AL2 023 内核应用内核实时补丁。

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

例如，以下命令应用了 AL2 0.23 内核版本的内核实时补丁 `6.1.12-17.42`

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

应用任何可用的内核实时补丁以及定期安全更新

使用以下命令。

```
$ sudo dnf upgrade --security
```

省略 `--security` 选项将包含错误修复。

> **Important**
>
> - 应用内核实时补丁后，内核版本不会更新。仅当重启实例后，版本才会更新为新版本。
> - AL2023 内核会接收 3 个月的内核实时补丁。之后，不会为该内核版本发布新的内核实时补丁。
> - 要想在三个月后继续收到内核实时补丁，您必须重启实例以移至新的内核版本。更新后，实例将在接下来的三个月内继续收到内核实时补丁。
> - 要查看内核版本的支持窗口，请运行以下命令。
>
> ```
> $ sudo dnf kernel-livepatch support
> The current version of the Linux kernel you are running will no longer receive
 live patches after 2025-07-22.
> ```

查看应用的内核实时补丁

查看应用的内核实时补丁

使用以下命令。

```
$ sudo kpatch list
Loaded patch modules:
livepatch_CVE_2022_36946 [enabled]

Installed patch modules:
livepatch_CVE_2022_36946 (6.1.57-29.131.amzn2023.x86_64)
livepatch_CVE_2022_36946 (6.1.57-30.131.amzn2023.x86_64)
```

该命令返回已加载和安装的安全更新内核实时补丁的列表。下面是示例输出。

> **Note**
>
> 单个内核实时补丁可以包含和安装多个实时补丁。

禁用内核实时修补

如果您不再需要使用内核实时修补，可以随时禁用它。

- 禁用使用 livepatches:
  1. 禁用插件：

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

  2. 禁用 kpatch 服务:

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

- 完全移除 livepatch 工具：
  1. 移除插件：

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

  2. 删除 kpatch-runtime:

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

  3. 移除所有已安装的内容 livepatches:

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

在 AL2 023 上更新 Linux 内核

主题

- AL2023 上的 Linux 内核版本
- 将 AL2 023 更新到内核 6.12
- AL2023 内核-常见问题解答

AL2023 上的 Linux 内核版本

AL2023 定期包含基于 Linux 内核的长期支持 (LTS) 版本的新内核版本。

AL2023 最初于 2023 年 3 月发布，内核为 6.1。

2025 年 4 月， AL2023 增加了对 Linux 内核 6.12 的支持。该内核增加了新功能，包括 EEVDF 调度、FUSE 直通 I/O 支持、新的 Futex API 以及 eBPF 的改进。内核 6.12 还允许用户空间程序在运行时使用用户空间阴影堆栈和内存封存来保护自己的安全。

将 AL2 023 更新到内核 6.12

您可以通过选择预装内核 6.12 的 AMI 或升级现有 AL2 023 实例，在内核 6.12 中运行 023。 AL2 EC2

正在运行 AL2 023 内核 6.12 AMI

您可以选择通过 AWS 控制台或查询 SSM 以获取特定参数来运行预装内核 6.12 的 AL2 023 AMI。要查询的 SSM 密钥以其开头，`/aws/service/ami-amazon-linux-latest/`后面是其中一个

- `al2023-ami-kernel-6.12-arm64`，适用于 arm64 架构
- `al2023-ami-minimal-kernel-6.12-arm64`，适用于 arm64 架构（AMI 最低版本）
- `al2023-ami-kernel-6.12-x86_64`，适用于 x86_64 架构
- `al2023-ami-minimal-kernel-6.12-x86_64`，适用于 x86_64 架构（AMI 最低版本）

有关选择 AL2 023 使用 SSM 参数启动 AL2 023 和 AWS CLI AMIs 的详细信息，请参阅。

将 AL2 023 实例更新到内核 6.12

您可以通过以下步骤将正在运行的 AL2 023 实例就地升级到内核 6.12：

1. 安装 `kernel6.12` 包：

```
$ sudo dnf install -y kernel6.12
```

2. 获取最新版本的`kernel6.12`软件包：

```
$ version=$(rpm -q --qf '%{version}-%{release}.%{arch}\n' kernel6.12 | sort -V |
 tail -1)
```

3. 将新`kernel6.12`内核设为默认内核：

```
$ sudo grubby --set-default "/boot/vmlinuz-$version"
```

4. 重启系统：

```
$ sudo reboot
```

5. （可选）卸载内核 6.1：

```
$ sudo dnf remove -y kernel
```

从内核 6.12 降级到内核 6.1

如果您在任何时候需要降级回内核 6.1，请使用以下步骤：

1. 请务必安装`kernel`软件包：

```
$ sudo dnf install -y kernel
```

2. 获取最新版本的`kernel`软件包：

```
$ version=$(rpm -q --qf '%{version}-%{release}.%{arch}\n' kernel | sort -V | tail
 -1)
```

3. 将内核 6.1 设置为默认内核：

```
$ sudo grubby --set-default "/boot/vmlinuz-$version"
```

4. 重启系统：

```
$ sudo reboot
```

5. （可选）卸载内核 6.12：

```
$ sudo dnf remove -y kernel6.12
```

AL2023 内核-常见问题解答

1. 内核更新后我需要重启吗？

对正在运行的内核进行每一次更改都需要重新启动。

2. 如何 up-to-date跨多个实例保留内核？

Amazon Linux 不提供管理实例队列的工具。我们建议您使用诸如 S AWS ystems Manager 之类的工具修补大型机群。

3. 如何检查我现在运行的是哪个内核版本？

在你的 AL2 023 实例上执行以下命令：

```
$ uname -r
```

4. 如何为内核 6.12 安装内核头文件、开发包和其他模块？

请运行：

```
$ sudo dnf install -y kernel6.12-modules-extra-$(uname -r) kernel-headers-$(uname -r)
 kernel-devel-$(uname -r)
```

5. 如何`perf`为我的内核选择正确的版本？

`perf`的功能与你正在运行的内核版本紧密相关。我们`perf`为内核 6.1 和`perf6.12`内核 6.12 提供了软件包。如果您已经`perf`安装并想切换到内核 6.12 版本，请执行：

```
$ dnf -y swap perf perf6.12
```

开始在 023 上 AL2编程运行时

AL2023 提供了某些语言运行时的不同版本。我们与同时支持多个版本的上游项目合作。查找有关如何使用 `dnf` 命令搜索、安装和管理这些名称受版本控制的软件包的信息。

以下主题概述了每种语言生态系统在 AL2 023 年是如何存在的。

主题

- C, C++，以及 Fortran 在 AL2 023
- Go 在 AL2 023
- Java在 AL2 023
- NodeJS 在 AL2 023
- Perl 在 AL2 023
- PHP 在 AL2 023
- Python 在 AL2 023
- Rust 在 AL2 023

C, C++，以及 Fortran 在 AL2 023

AL2023 包括两个 GNU 编译器集合 (GCC) 还有 Clang 前端 LLVM（低级虚拟机）。

的主要版本 GCC 将在 AL2 023 年的整个生命周期中保持不变。次要版本修复了错误，可能包含在 AL2 023 版本中。其他错误、性能和安全修复可能会向后移植到的主要版本中 GCC 那是 AL2 023 年发货的。

AL2023 包括的版本 11 GCC 使用 C (`gcc`)、C++ (g++) 和 Fortran (`gfortran`) 前端。

AL2023 不启用 Ada (`gnat`), Go (`gcc-go`)、Objective-C 或 Objective-C++ 前端。

构建 AL2 023 RPMs 时使用的默认编译器标志包括优化和强化标志。要使用 GCC 构建自己的代码，我们建议您添加优化和强化标志。

> **Note**
>
> 调用 `gcc --version` 时，会显示一个版本字符串，例如 `gcc (GCC) 11.3.1 20221121 (Red Hat 11.3.1-4)`。`Red Hat` 指的是 Amazon Linux GCC 软件包所基于的 GCC 供应商

分支。根据显示的错误报告网址`gcc --help`，所有错误报告和支持请求都应发送至 Amazon Linux。
有关该供应商分支中一些长期变更（例如`__GNUC_RH_RELEASE__`宏）的更多信息，请参阅 Fedora 软件包来源。

有关核心工具链的更多信息，请参阅核心工具链包 glibc, gcc, binutils。

有关 AL2 023 及其与其他 Linux 发行版关系的更多信息，请参阅与 Fedora 的关系。

有关 AL2 023 年编译器三元组与之相比的变化的更多信息， AL2请参阅。编译器三元组

Go 在 AL2 023

你可能想自己编写代码 Go在亚马逊 Linux 上，可能想使用 AL2 023 附带的工具链。与之类似 AL2，AL2023 将更新 Go 工具链贯穿整个操作系统的生命周期。这或许是为了回应我们发布的工具链中的任何 CVE，或许是季度发布的一部分。

Go 是一种移动速度相对较快的语言。可能存在写入现有应用程序的情况 Go 必须适应新版本的 Go 工具链。有关 Go，请参阅 Go 1 和《的未来》 Go 节目。

尽管 AL2 023 将包含新版本的 Go 工具链在其生命周期中，这不会与上游同步进行 Go 发布。因此，使用 Go 如果你想构建， AL2023 中提供的工具链可能不适合 Go 使用尖端功能编写代码 Go 语言和标准库。

在 AL2 023 的生命周期内，之前的软件包版本不会从存储库中删除。如果是以前的 Go 工具链是必需的，你可以选择放弃更新的错误和安全修复 Go 工具链，并使用适用于任何 RPM 的相同机制从存储库中安装先前版本。

如果你想自己建造 Go AL2023 上的代码，你可以使用 Go 工具链包含在 AL2 023 中，因为它知道该工具链可能会在 023 的生命周期中向前发展。 AL2

AL2023 写入的 Lambda 函数 Go

如 Go 编译为原生代码，Lambda 会处理 Go 作为自定义运行时。您可以使用`provided.al2023`运行时进行部署 Go 在 AL2 023 上函数到 Lambda。

有关更多信息，请参阅使用构建 Lambda 函数 Go（在 AWS Lambda 开发人员指南中）。

Java在 AL2 023

AL2023 提供了多个版本的 Amazon Corretto 来支持基于的工作负载。Java AL2023 中包含的所有Java基础软件包均使用构建。Amazon Corretto 17

Corretto 是开放式 Java 开发套件 (OpenJDK) 的版本，得到了长期支持。AmazonCorretto 已通过 Java 技术兼容性套件 (TCK) 认证，可确保其符合 Java SE 标准并且可在 Linux、和。Windows macOS

Corretto 1.8.0、Corretto 11 和 Corretto 17 都有一个可用的 Amazon Corretto 软件包。

023 AL2 中的每个 Corretto 版本的支持时间与 Corretto 版本相同，或者直到 023 的生命周期结束，以较早者为准。 AL2有关更多信息，请参阅亚马逊 Linux 软件包支持声明和亚马逊 Corretto。 FAQs

NodeJS 在 AL2 023

NodeJS在 AL2 023 中，由 18、20 和 22 版本表示。它们是命名空间的，可以同时安装在同一个系统上。NodeJS 以多个包的形式分发，其中包括节点、与其兼容版本的 npm 工具、文档、库等。例如，对于 NodeJS 18、节点和 npm 由`nodejs`和`nodejs-npm`软件包提供。但是，以下所有版本的 NodeJS 都有以开头的命名空间包名。`nodejs{MAJOR_VERSION}`例如，NodeJS 20，随附的 n ode 和 npm `nodejs20-npm` 分别打包为`nodejs20`和。

允许同时安装不同的主要版本 NodeJS，软件包附带命名空间的可执行文件、模块和其他文件，以避免重叠和文件系统冲突。例如，节点可执行文件被命名`/usr/bin/node-{MAJOR_VERSION}`，npm 可执行文件被命名`/usr/bin/npm-{MAJOR_VERSION}`。但是，在运行的系统`/usr/bin/npm`上只能有一个`/usr/bin/node`和一个。这些可执行文件是虚拟名称（符号链接），指向当前活动版本的 NodeJS 的实际可执行文件。它是通过使用替代系统来实现的。

使用备选方案可以让你使用单个命令来选择哪个 NodeJS 使用版本的配置文件、二进制文件（例如`node`和`npm`）和全局安装的模块。默认情况下，备选方案配置为 auto 模式，该模式使用优先级来选择当前处于活动状态的 NodeJS。 但是，您可以随时通过运行在已安装的版本之间切换`alternatives --config node`。当前，所有支持的 NodeJS 版本都具有同等优先级。

一些有用的替代命令：

1. 检查备选方案中配置了什么

```
alternatives --list
```

2. 检查节点的当前配置

```
alternatives --display node
```

3. 以交互方式更改 NodeJS version

```
alternatives --config node
```

4. 切换到手动模式并选择特定版本

```
alternatives --set node /usr/bin/node-{MAJOR_VERSION}
```

5. 切换回 auto 版本选择模式

```
alternatives --auto node
```

Perl 在 AL2 023

AL2023 提供了 5.32 版本的 Perl编程语言。

虽然 Perl 作为其中的一部分，提供了高度的语言兼容性 Perl 在过去的几十年中，有5个版本，预计 Amazon Linux不会改变 Perl 在 AL2 023 版本中为 5.32。亚马逊 Linux 将继续发布安全补丁 Perl 根据我们的套餐支持声明，生命周期 AL2为 023。

Perl AL2023 年的模块

各种 Perl 模块的封装 RPMs 方式与 AL2 023 相同。虽然有很多 Perl 可用模块 RPMs，因为 Amazon Linux 的目标并不是要打包所有可能的东西 Perl 模块。模块打包为其他操作系统 RPM 包 RPMs 可能依赖的模块，因此 Amazon Linux 将优先考虑这些安全补丁而不是纯粹的功能更新。

AL2023 还包括CPAN这样 Perl 开发人员可以使用惯用包管理器来 Perl 模块。

PHP 在 AL2 023

AL2023 目前提供了 PHP编程语言，版本 8.1、8.2、8.3 和 8.4。每个版本的支持时间与上游版本相同 PHP。 有关更多信息，请参阅 Pac kage 支持声明。

从较旧版本迁移 PHP 版本

上游 PHP 社区整理了用于搬家的全面迁移文档：

- 来自 PHP 8.3.x 到 PHP 8.4.x
- 来自 PHP 8.2.x 到 PHP 8.3.x
- 来自 PHP 8.1.x 到 PHP 8.2.x
- 来自 PHP 8.0.x 到 PHP 8.1.x

AL2 包括 PHP 8.0、8.1 和 8.2 提供了轻松`amazon-linux-extras`升级到 AL2 023 的途径。

迁移来源 PHP 7.x 版本

> **Note**
>
> 这些区域有：PHPproject 维护支持的版本列表和时间表，以及不支持的分支的列表。AL2023 发布时，所有 7.x 和 5.x 版本的 PHP没有得到支持 PHP 社区，并未在 AL2 023 年作为选项包括在内。

上游 PHP 社区整理了全面的迁移文档，以便迁移到 PHP 8.0 来自 PHP 7.4 。结合上一节中提及的有关迁移到的文档 PHP 8.1 和 PHP 8.2，你可以迁移你的 PHP 基于现代的应用程序 PHP.

> **Note**
>
> AL2 包括 PHP 7.1、7.2、7.3 和 7. `amazon-linux-extras` 4 英寸 值得注意的是，所有这些额外功能都 end-of-life保证会获得任何进一步的安全更新。

PHP AL2023 年的模块

AL2023 包括许多 PHP 包含在中的模块 PHP 核心。 AL2023 的目标不是将所有软件包都包含在 PHP 扩展社区图书馆 (PECL) 。

Python 在 AL2 023

AL2023 已移除 Python 2.7 以及任何需要的组件 Python 现在是为使用而写的 Python 3.

AL2023 makes Python 3 可用`/usr/bin/python3`来保持与客户代码以及 AL2 023 附带的 Python 代码的兼容性，这将保持不变 Python 3.9 在 AL2 023 年的生命中。

`/usr/bin/python3`指向的 python 版本被认为是系统 Python，对于 AL2 023 来说，这是 Python 3.9。

的较新版本 Python 之外的压缩算法（例如 Python 3.11 在 AL2 023 中作为软件包提供，并在上游版本的生命周期内受支持。有关支持 Python 3.11 多长时间的信息，请参阅 Python 3. 11。

的多个版本 Python 可以在 AL2 023 上同时安装。虽然`/usr/bin/python3`永远都是 Python 3.9，Python 的每个版本都有命名空间，可以通过其版本号找到。例如，如果安装了 `python3.11`，则 `/usr/bin/python3.11` 将与 `/usr/bin/python3.9` 并存，而 `/usr/bin/python3` 符号链接指向 `/usr/bin/python3.9`。

> **Note**
>
> 不要更改`/usr/bin/python3`符号链接指向的内容，因为这可能会破坏 AL2 023 的核心功能。

Python AL2023 年的模块

各种 Python 模块的封装 RPMs 方式与 AL2 023 相同。通常， RPMs 对于 Python 将仅针对以下系统版本构建模块 Python.

Rust 在 AL2 023

你可能想用他们写的代码来编写 Rust在亚马逊 Linux 上，可能想使用 AL2 023 附带的工具链。

与之类似 AL2， AL2023 将更新 Rust 工具链贯穿整个操作系统的生命周期。这或许是为了回应我们发布的工具链中的任何 CVE，或许是季度发布的一部分。

Rust 是一种发展速度相对较快的语言，大概每六周发布一个新版本。这些版本中可能添加了新的语言或标准库功能。尽管 AL2 023 将加入新版本的 Rust 工具链在其生命周期中，这不会与上游同步进行 Rust 发布。因此，使用 Rust 如果你想构建， AL2023 中提供的工具链可能不适合 Rust 使用尖端功能编写代码 Rust 语言。

在 AL2 023 的生命周期内，不会从存储库中删除旧的软件包版本。如果年龄较大 Rust 工具链是必需的，你可以选择放弃更新的错误和安全修复 Rust 工具链，并使用适用于任何 RPM 的相同机制从存储库中安装旧版本。

如果你想自己建造 Rust AL2023 上的代码，你可以使用 Rust 工具链包含在 AL2 023 中，因为它知道该工具链可能会在 023 的生命周期中向前发展。 AL2

AL2023 写入的 Lambda 函数 Rust

因为 Rust 编译为原生代码，Lambda 会处理 Rust 作为自定义运行时。您可以使用`provided.al2023`运行时进行部署 Rust 在 AL2 023 上函数到 Lambda。

有关更多信息，请参阅使用构建 Lambda 函数 Rust（在 AWS Lambda 开发人员指南中）。

AL2023 预留用户和群组

AL2023 在映像配置期间和安装某些软件包期间都会预先分配某些用户和组。为了防止发生冲突，此处 GIDs 列出了用户、组 UIDs 及其关联的。

主题

- AL2023 个预留用户名单
- AL2023 预留组名单

AL2023 个预留用户名单

用户名称	UID
root	0
bin	1
守护进程	2
adm	3
lp	4
同步	5
shutdown	6
停止	7
邮件	8
operator	11
游戏	12
ftp	14
鱿鱼	23

用户名称	UID
被命名	25
postgres	26
mysql	27
nscd	28
nscd	28
rpcuser	29
rpc	32
mailnull	47
Apache	48
smmsp	51
Tomcat	53
ldap	55
tss	59
nslcd	65
avahi	70
tcpdump	72
sshd	74
radvd	75
dbus	81
postfix	89

用户名称	UID
dovecot	97
stapusr	156
stapsys	157
stapdev	158
avahi-autoipd	170
脉冲	171
rtkit	172
sanlock	179
系统网络	192
系统解析	193
uuidd	961
Stap-server	962
systemd-journal-remote	963
redis6	970
pesign	971
smtpq	972
smtpd	973
nginx	974
munge	975
memcached	976

用户名称	UID
sphinx	977
haproxy	978
latpak	979
debuginfo	980
dovenull	981
dnsmasq	982
未绑定	983
clamsan	984
大声疾呼	985
clamup	986
colord	987
ods	988
aws-kinesis-agent-user	989
saslauth	990
cwagent	991
polkitd	992
ec2-instance-connect	993
chrony	994
系统时间同步	995
系统核心转储	996

用户名称	UID
libstoragemgmt	997
系统房	999
ec2 用户	1000
没有人	65534

按名称列出

用户名称	UID
adm	3
Apache	48
avahi	70
avahi-autoipd	170
aws-kinesis-agent-user	989
bin	1
chrony	994
大声疾呼	985
clamsan	984
clamup	986
colord	987
cwagent	991
守护进程	2

用户名称	UID
dbus	81
debuginfo	980
dnsmasq	982
dovecot	97
dovenull	981
ec2-instance-connect	993
ec2 用户	1000
latpak	979
ftp	14
游戏	12
停止	7
haproxy	978
ldap	55
libstoragemgmt	997
lp	4
邮件	8
mailnull	47
memcached	976
munge	975
mysql	27

用户名称	UID
被命名	25
nginx	974
没有人	65534
nscd	28
nscd	28
nslcd	65
ods	988
operator	11
pesign	971
polkitd	992
postfix	89
postgres	26
脉冲	171
radvd	75
redis6	970
root	0
rpc	32
rpcuser	29
rtkit	172
sanlock	179

用户名称	UID
saslauth	990
shutdown	6
smmsp	51
smtpd	973
smtpq	972
sphinx	977
鱿鱼	23
sshd	74
Stap-server	962
stapdev	158
stapsys	157
stapusr	156
同步	5
系统核心转储	996
systemd-journal-remote	963
系统网络	192
系统房	999
系统解析	193
系统时间同步	995
tcpdump	72

用户名称	UID
Tomcat	53
tss	59
未绑定	983
uuidd	961

AL2023 预留组名单

组名	GID
root	0
bin	1
守护进程	2
sys	3
adm	4
tty	5
磁盘	6
磁盘	6
lp	7
mem	8
kmem	9
wheel	10
cdrom	11

组名	GID
邮件	12
邮件	12
man	15
拨号	18
软盘	19
游戏	20
slocate	21
utmp	22
鱿鱼	23
被命名	25
postgres	26
mysql	27
nscd	28
nscd	28
rpcuser	29
rpc	32
磁带	33
utempter	35
kvm	36
视频	39

组名	GID
mailnull	47
Apache	48
ftp	50
smmsp	51
Tomcat	53
锁定	54
ldap	55
tss	59
audio	63
avahi	70
tcpdump	72
sshd	74
radvd	75
saslauth	76
dbus	81
screen	84
wbpriv	88
postfix	89
后期投放	90
dovecot	97

组名	GID
用户	100
input	104
给予	105
sgx	106
嘲笑	135
stapusr	156
stapusr	156
stapsys	157
stapsys	157
stapdev	158
stapdev	158
avahi-autoipd	170
脉冲	171
rtkit	172
sanlock	179
系统日志	190
系统网络	192
系统解析	193
usbmon	959
钢丝鲨	960

组名	GID
uuidd	961
Stap-server	962
systemd-journal-remote	963
用户共享	964
redis6	965
pesign	966
smtpq	967
smtpd	968
nginx	969
munge	970
memcached	971
sphinx	972
跟踪	973
haproxy	974
latpak	975
debuginfo	976
dovenull	977
dnsmasq	978
未绑定	979
clamsan	980

组名	GID
大声疾呼	981
病毒组	982
病毒组	982
病毒组	982
clamup	983
printadmin	984
colord	985
ods	986
Docker	987
aws-kinesis-agent-user	988
cwagent	989
pulse-rt	990
脉冲接入	991
ec2-instance-connect	993
chrony	994
系统时间同步	995
系统核心转储	996
libstoragemgmt	997
ssh_keys	998
系统房	999

组名	GID
ec2 用户	1000
没有	1001
polkitd	9920
没有人	65534

按名称列出

组名	GID
adm	4
Apache	48
audio	63
avahi	70
avahi-autoipd	170
aws-kinesis-agent-user	988
bin	1
cdrom	11
chrony	994
大声疾呼	981
clamsan	980
clamup	983
colord	985

组名	GID
cwagent	989
守护进程	2
dbus	81
debuginfo	976
拨号	18
磁盘	6
磁盘	6
dnsmasq	978
Docker	987
dovecot	97
dovenull	977
ec2-instance-connect	993
ec2 用户	1000
latpak	975
软盘	19
ftp	50
游戏	20
haproxy	974
input	104
kmem	9

组名	GID
kvm	36
ldap	55
libstoragemgmt	997
锁定	54
lp	7
邮件	12
邮件	12
mailnull	47
man	15
mem	8
memcached	971
嘲笑	135
munge	970
mysql	27
被命名	25
没有	1001
nginx	969
没有人	65534
nscd	28
nscd	28

组名	GID
ods	986
pesign	966
polkitd	9920
后期投放	90
postfix	89
postgres	26
printadmin	984
脉冲	171
脉冲接入	991
pulse-rt	990
radvd	75
redis6	965
给予	105
root	0
rpc	32
rpcuser	29
rtkit	172
sanlock	179
saslauth	76
screen	84

组名	GID
sgx	106
slocate	21
smmsp	51
smtpd	968
smtpq	967
sphinx	972
鱿鱼	23
ssh_keys	998
sshd	74
Stap-server	962
stapdev	158
stapdev	158
stapsys	157
stapsys	157
stapusr	156
stapusr	156
sys	3
系统核心转储	996
系统日志	190
systemd-journal-remote	963

组名	GID
系统网络	192
系统房	999
系统解析	193
系统时间同步	995
磁带	33
tcpdump	72
Tomcat	53
跟踪	973
tss	59
tty	5
未绑定	979
usbmon	959
用户	100
用户共享	964
utempter	35
utmp	22
uuidd	961
视频	39
病毒组	982
病毒组	982

组名	GID
病毒组	982
wbpriv	88
wheel	10
钢丝鲨	960

2023 年可用的编解码器列表 AL2

AL2023 通过其标准存储库提供了一系列多媒体编解码器。本页概述了编解码器及其典型用例。

> **Important**
>
> 使用和分发 Amazon Linux 中包含的编解码器可能需要您从第三方获得许可权，包括某些第三方音频和视频格式的所有者或许可人。您全权负责获得这些许可并支付任何必要的特许权使用费或费用。

编解码器	描述
flac	一款免费的开源无损音频编解码器，可在不丢失任何数据或质量的情况下压缩音频，通常用于高质量的音频存储
fdk-aac-free	AAC（高级音频编解码器）标准的开源实现，为流媒体或文件存储等 MP3 替代方案提供高质量的音频压缩
webrtc-audio-processing	WebRTC（网络实时通信）中使用的音频处理库，提供噪声抑制、回声消除和增益控制等功能
opus	一款功能多样、高效的音频编解码器，专为实时流媒体而设计，具有低延迟，支持各种音频应用，包括VoIP和音乐流
libsndfile	用于读取和写入各种格式（例如 WAV、AIFF 和 FLAC）的音频文件的库，通常用于音频处理和处理工具

Amazon Linux 2023 中的安全性与合规性

> **Important**
>
> 如果您想报告漏洞或对 AWS 云服务或开源项目有安全疑虑，请使用 “漏洞报告” 页面与 AWS 安全部门联系

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在 AWS 云中运行 AWS 服务的基础架构。 AWS 还为您提供可以安全使用的服务。作为AWS 合规计划合规计划合规计划合的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用于 AL2 023 的合规计划，请参阅按合规计划划分的范围内的AWS 服务按合规计划。
- 云中的安全性：您的责任由您使用的 AWS 服务决定。您还需要对其它因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

主题

- 2023 年的亚马逊 Linux 安全公 AL2告
- 列出适用的公告
- 就地应用安全更新
- 为 AL2 0 SELinux 23 设置模式
- 在 023 上 AL2启用 FIPS 模式
- 在 AL2 023 容器中启用 FIPS 模式
- 在 023 上交换 OpenSSL FIPS 提供商 AL2
- AL2023 内核强化
- UEFI 在 023 上安全启动 AL2

2023 年的亚马逊 Linux 安全公 AL2告

尽管我们努力确保 Amazon Linux 的安全，但有时会有一些安全问题需要修复。当有可用的修复程序时，就会发布公告。我们发布公告的主要位置是 Amazon Linux 安全中心 (ALAS)。有关更多信息，请参阅 Amazon Linux 安全中心。

> **Important**
>
> 如果您想报告漏洞或对 AWS 云服务或开源项目有安全疑虑，请使用 “漏洞报告” 页面与 AWS 安全部门联系

有关影响 AL2 023 的问题和相关更新的信息由 Amazon Linux 团队在多个地点发布。安全工具通常会从这些主要源代码中获取信息并将结果呈现给您。因此，您可能不会直接与 Amazon Linux 发布的主要来源进行交互，而是与您的首选工具（例如 Amaz on Inspec tor）提供的界面进行交互。

亚马逊 Linux 安全中心公告

Amazon Linux 的公告是针对不符合公告内容的商品提供的。本节包含有关ALAS本身的公告，以及不适合公告的信息。有关更多信息，请参阅亚马逊 Linux 安全中心 (ALAS) 公告。

例如，适用于A pache Log4j的2021-001-Amazon Linux热补丁公告属于公告而不是公告。在本公告中，Amazon Linux 添加了一个软件包，以帮助客户缓解不属于亚马逊 Linux 的软件中的安全问题。

Amazon Linux 安全中心 CVE Explorer 也在 ALAS 公告中发布。欲了解更多信息，请参阅新网站 CVEs。

亚马逊 Linux 安全中心常见问题解答

有关 ALAS 和 Amazon Linux 如何评估的一些常见问题的答案 CVEs，请参阅亚马逊 Linux 安全中心 (ALAS) 常见问题解答 ()。FAQs

ALAS 公告

亚马逊 Linux 公告包含与亚马逊 Linux 用户相关的重要信息，通常是有关安全更新的信息。A mazon Linux 安全中心是网络上可以看到公告的地方。咨询信息也是 RPM 软件包存储库元数据的一部分。

公告和 RPM 存储库

Amazon Linux 2023 软件包存储库可能包含描述零个或多个更新的元数据。该`dnf updateinfo`命令以包含此信息的存储库元数据文件名命名`updateinfo.xml`。虽然命令名为`updateinfo`，而元数据文件指的是`update`，但这些都是指作为公告一部分的软件包更新。

Amazon Linux Advisories 发布在亚马逊 Linux 安全中心网站上，`dnf`软件包管理器引用的 RPM 存储库元数据中也存在信息。网站和存储库的元数据最终是一致的，并且网站上的信息和存储库元数据可能存在不一致之处。这通常发生在 AL2 023 的新版本正在发布时，在最新的 AL2 023 版本之后公告已有更新。

虽然通常会与解决该问题的软件包更新一起发布新的公告，但情况并非总是如此。可以为已发布的软件包中已解决的新问题创建公告。现有公告也可以使用新公告进行更新 CVEs ，现有更新将解决这些问题。

Amazon Linux 2023 的023 年通过版本化存储库进行确定性升级 AL2功能意味着特定 AL2 023 版本的 RPM 存储库包含截至该版本的 RPM 存储库元数据的快照。这包括描述安全更新的元数据。特定 AL2 023 版本的 RPM 存储库在发布后不会更新。查看旧版本的 AL2 023 RPM 存储库时，新的或更新的安全公告将不可见。有关如何使用`dnf`软件包管理器查看`latest`存储库版本或特定的 AL2 023 版本的信息，请参阅该列出适用的公告部分。

咨询 IDs

每份公告都由一个人引用`id`。目前，这是亚马逊 Linux 的一个怪癖，亚马逊 L inux 安全中心网站将公ALAS2告列为 ALAS-2024-581，而`dnf`软件包管理器会将该公告列为 023-2024-581。如果指的是特定的公告，则需要使用软件包管理器 ID 时就地应用安全更新。

对于 Amazon Linux，每个主要版本的操作系统都有自己的咨询命名空间 IDs。不应对 Amazon Linux 公告的格式做出任何假设 IDs。从历史上看，Amazon Linux Advisory IDs 一直遵循以下模式`NAMESPACE-YEAR-NUMBER`。的全部可能`NAMESPACE`值尚未定义，但已包括`ALAS`、`ALASCORRETTO8`、`ALAS2023`、`ALAS2ALASPYTHON3.8`、和`ALASUNBOUND-1.17`。这是创建通告的年份，并且`NUMBER`是命名空间中的唯一整数。`YEAR`

尽管 IDs Ad visory 通常是按更新发布顺序排列的，但有许多原因不可能出现这种情况，因此不应假设这种情况。

将建议 ID 视为一个不透明的字符串，该字符串在 Amazon Linux 的每个主要版本中都是唯一的。

在 Amazon Linux 2 中，每个 Extra 都位于单独的 RPM 存储库中，咨询元数据仅包含在与之相关的存储库中。一个存储库的公告不适用于另一个存储库。在 Amazon Linux 安全中心网站上，目前每个主要的 Amazon Linux 版本都有一个公告列表，并且没有按存储库列表分列。

由 AL2于 023 不使用 Extras 机制来打包其他版本的软件包，因此目前只有两个 RPM 存储库，每个存储库都有咨询、`core`存储库和存储库。`livepatchlivepatch`存储库适用于023 上的 Kernel Live AL2 补丁。

公告发布日期和公告更新日期

Amazon Linux Advisories 的公告发布日期表示安全更新首次在 RPM 存储库中公开发布的时间。在 RPM 存储库提供可安装的修复程序后，会立即在 Amazon Linux 安全中心网站上发布公告。

公告更新日期表示在公告上次发布后何时向其添加新信息。

在 AL2 023 版本号（例如 2023.6.20241031）和与该版本一起发布的公告公告发布日期之间不应做任何假设。

咨询类型

RPM 存储库元数据支持不同类型的咨询。虽然 Amazon Linux 几乎只发布了安全更新的公告，但不应假设这一点。可能会发布诸如错误修复、增强功能和新软件包之类的事件的公告，并将该公告标记为包含此类更新。

咨询严重性

每份公告都有自己的严重性，因为每个问题都是单独评估的。在单个通告中 CVEs 可以解决多个问题，每个 CVE 可能有不同的评估，但该通告本身具有一个严重性。可以有多个公告提及单个软件包更新，因此一个特定的软件包更新可能有多个严重性（每个公告一个）。

按严重性降序排列，Amazon Linux 使用了 “严重”、“重要”、“中” 和 “低” 来表示公告的严重性。Amazon Linux Advisories 也可能没有严重性，尽管这种情况极为罕见。

Amazon Linux 是基于 RPM 的 Linux 发行版之一，它使用 “中等” 一词，而其他一些基于 RPM 的 Linux 发行版则使用等效的术语 “中”。Amazon Linux 软件包管理器将这两个术语视为等效，第三方软件包存储库可能使用 “中” 一词。

随着对公告中解决的相关问题的了解越来越多，Amazon Linux Advisories 的严重性可能会随着时间的推移而发生变化。

通告的严重性通常会跟踪该公告所 CVEs 引用的 Amazon Linux 评估的最高 CVSS 分数。在某些情况下，情况可能并非如此。一个例子是存在未分配 CVE 但已解决的问题。

有关 Amazon Linux 如何使用咨询严重性评级的更多信息，请参阅 ALAS 常见问题解答。

公告和套餐

单个软件包可能有许多公告，但并非所有软件包都会为其发布公告。可以在多个公告中引用特定的软件包版本，每个公告都有自己的严重性和。 CVEs

在一个新的 AL2 023 版本中，可以同时发布多个针对同一个软件包更新的公告，也可以快速连续发布。

像其他 Linux 发行版一样，可以从同一个源代码包中构建一到多个不同的二进制包。例如，ALAS-2024-698 是亚马逊 Linux 安全中心网站 AL2 023 部分列出的适用于该`mariadb105`软件包的公告。这是源包的名称，咨询本身指的是源包旁边的二进制包。在本例中，有十几个二进制包是从同一个`mariadb105`源代码包中构建的。虽然存在与源包同名的二进制包极为常见，但这并不普遍。

虽然 Amazon Linux Advisories 通常会列出所有根据更新的源代码包构建的二进制包，但不应假设如此。包管理器和 RPM 存储库元数据格式允许公告列出已更新的二进制包的子集。

特定的公告也可能仅适用于特定的 CPU 架构。有些软件包可能不是为所有架构构建的，也可能存在不影响所有架构的问题。如果某个软件包在所有架构上都可用，但问题仅适用于一个架构，那么 Amazon Linux 通常不会发布仅提及受影响架构的公告，尽管不应假设这一点。

由于软件包依赖关系的性质，公告通常会引用一个软件包，但是安装该更新需要其他软件包更新，包括未在公告中列出的软件包。`dnf`软件包管理器将负责安装所需的依赖项。

公告和 CVEs

一份公告可能涉及零个或多个 CVEs，并且可能有多个咨询引用同一 CVE。

建议何时可能引用零的一个例子 CVEs 是尚未为问题分配 CVE（或从来没有）。

例如，当 CVE 适用于多个软件包时，多个公告可能会引用同一 CVE 的示例。例如，CVE-2024-21208 适用于 Corretto 8、11、17 和 21。这些 Corretto 版本 AL2在 023 年都是一个单独的软件包，每个软件包都有公告：ALAS-2024-754 适用于 Corretto 8，ALAS-2024-753 适用于 Corretto 11，ALAS-2024-752 适用于 Corretto 17，ALAS-2024-752 适用于 Corretto 21，适用于 Corretto 21。虽然这些 Corretto 版本都有相同的列表，但不应 CVEs假设这一点。

对于不同的软件包，可以对特定 CVE 进行不同的评估。例如，如果在严重性为 “重要” 的通告中引用了特定 CVE，则可能会发布另一份通告，其中引用了具有不同严重性的相同 CVE。

RPM 存储库元数据允许列出每份公告的参考文献。虽然 Amazon Linux 通常只引用 CVEs，但元数据格式确实允许其他参考类型。

RPM 软件包存储库元数据将仅在 CVEs 有可用的修复程序时引用。亚马逊 Linux 安全中心网站的 “浏览” 部分 CVEs 包含有关亚马逊 Linux 评估过的信息。此评估可能会得出各种 Amazon Linux 版本和软件包的 CVSS 基本分数、严重程度和状态。特定 Amazon Linux 版本或软件包的 CVE 状态可能为 “未受影响”、“待修复” 或 “未计划修复”。在公告发布之前，的状态和评估 CVEs 可能会多次以任何方式发生变化。这包括重新评估 CVE 对亚马逊 Linux 的适用性。

该公告首次发布后，该公 CVEs 告所引用的列表可能会发生变化。

咨询文本

公告还将包含描述创建咨询的原因的一个或多个问题的文本。通常，此文本将是未经修改的 CVE 文本。此文本可能指有可用的修复程序的上游版本号，这些版本号与 Amazon Linux 已应用修复的软件包版本不同。Amazon Linux 通常会从较新的上游版本中反向移植修复程序。如果公告文本中提到的上游版本与亚马逊 Linux 版本中提供的版本不同，则公告中的亚马逊 Linux 软件包版本对于亚马逊 Linux 来说将是准确的。

RPM 存储库元数据中的咨询文本可以是占位符文本，详情请访问 Amazon Linux 安全中心网站。

内核实时补丁公告

实时补丁公告的独特之处在于，它们所指的软件包（Linux 内核）与公告所针对的软件包（例如`kernel-livepatch-6.1.15-28.43`）不同。

内核动态补丁公告将引用特定的 Live Patch 包可以针对该实时补丁包所适用的特定内核版本解决的问题（例如 CVEs）。

每个上线补丁都适用于特定的内核版本。要为 CVE 应用实时补丁，需要安装适用于您的内核版本的正确实时补丁包，并应用实时补丁。

例如，可以为 AL2 023 内核版本对 CVE-2023-6111 进行实时修补`6.1.56-82.1256.1.59-84.139`，和。`6.1.61-85.141`还发布了包含此 CVE 修复程序的新内核版本，并有单独的公告。为了在 AL2 023 上解决 CVE-2023-6111 问题，需要运行等于或高于 0 ALAS223-2023-461 指定版本的内核版本，或者需要在应用适用的 livepatch 的情况下运行带有此 CVE 实时补丁的内核版本。

当特定内核版本有新的实时补丁可用并且已经有可用的实时补丁时，就会发布该`kernel-livepatch-KERNEL_VERSION`软件包的新版本。例如，发布

公ALASLIVEPATCH-2023-003告的kernel-livepatch-6.1.15-28.43-1.0-1.amzn2023软件包中包含了涵盖三个6.1.15-28.43内核的实时补丁 CVEs。后来，该kernel-livepatch-6.1.15-28.43-1.0-2.amzn2023软件包发布了公ALASLIVEPATCH-2023-009告；这是对先前6.1.15-28.43内核的实时补丁包的更新，其中包含另外三个的实时补丁 CVEs。其他内核版本还存在其他实时补丁建议问题，其中包含针对这些特定内核版本的实时补丁的软件包。

有关内核实时补丁的更多信息，请参阅023 上的 Kernel Live AL2 补丁。

对于任何围绕安全公告开发工具的人，还建议查看该咨询的 XML 架构和 updateinfo.xml部分以获取更多信息。

咨询的 XML 架构和 updateinfo.xml

该updateinfo.xml文件是软件包存储库格式的一部分。它是dnf包管理器为实现列出适用的公告和就地应用安全更新等功能而解析的元数据。

我们建议使用dnf包管理器的 API，而不是编写自定义代码来解析存储库元数据格式。 AL2023 dnf 中的版本可以解析 AL2 023 和 AL2 存储库格式，因此 API 可用于检查任一操作系统版本的咨询信息。

RPM 软件管理项目在 rpm 元数据存储库中记录了 RPM 元数据格式。 GitHub

对于那些直接解析updateinfo.xml元数据的开发工具，强烈建议您仔细阅读 rpm-metadata 文档。该文档涵盖了在野外看到的情况，其中包括许多例外情况，您可以合理地解释为元数据格式的规则。

上的 raw-historical-rpm-repository-examples 存储库中还有越来越多的真实updateinfo.xml文件示例。 GitHub

如果文档中有任何不清楚的地方，你可以在 GitHub 项目上打开一个问题，这样我们就可以回答问题并适当地更新文档。作为开源项目，也欢迎拉取请求更新文档。

列出适用的公告

dnf软件包管理器可以访问元数据，这些元数据描述了哪些软件包版本中修复了哪些公告。因此，它可以列出适用于实例或容器镜像的公告。

> **Note**
>
> 诸如之类的工具AWS Systems Manager可以使用此功能来显示与队列相关的更新，而不仅仅是单个实例。

列出更新时，您可以指示dnf查看特定 AL2 023 版本的元数据或最新版本的元数据。

> **Note**
>
> 一旦 AL2 023 版本发布，它就是不可变的。因此，Amazon Linux 安全中心上新的或更新的公告只会添加到 023 新版本的 AL2元数据中

现在，我们将举例说明哪些公告适用于某些 AL2 023 容器镜像。这些命令都适用于非容器化环境，例如实例。 EC2

Listing advisories in a specific version

在此示例中，我们将看看 2023.1.20230628 版本中的哪些公告与 2023.0.20230315 版本的容器镜像相关。

> **Note**
>
> 此示例使用 2023.0.20230315 和 2023.1.20230628 版本，但这些版本不是 023 的最新版本。有关包含最新安全更新的最新版本，请参阅 AL2 023 发行说明。AL2

在此示例中，我们将从 2023.0.202 30315 版本的容器镜像开始。

首先，我们从容器注册表中获取此容器镜像。末尾的表示特定版本的图像版本；此图像版本通常为零。`.0`

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
   2023.0.20230315.0: Pulling from amazonlinux/amazonlinux
b76f3b09316a: Pull complete
Digest: sha256:94e7183b0739140dbd5b639fb7600f0a2299cec5df8780c26d9cb409da5315a9
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.0.20230315.0
public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
```

我们现在可以在容器内生成一个 shell，我们将要求从中列dnf出哪些公告与容器中安装的软件包有关。

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.0.20230315.0
```

```
bash-5.2#
```

现在，该dnf updateinfo命令用于显示 2023.1.20230628 版本中的哪些公告与我们已安装的软件包相关的摘要。

```
$ dnf updateinfo --releasever=2023.1.20230628
Amazon Linux 2023 repository                    42 MB/s |  15 MB     00:00
Last metadata expiration check: 0:00:02 ago on Mon Jul 22 20:24:24 2024.
Updates Information Summary: available
    8 Security notice(s)
        1 Important Security notice(s)
        5 Medium Security notice(s)
        2 Low Security notice(s)
```

要获取公告清单，可以--list选择给。dnf updateinfo

```
$ dnf updateinfo --releasever=2023.1.20230628 --list
Last metadata expiration check: 0:01:22 ago on Mon Jul 22 20:24:24 2024.
ALAS2023-2023-193 Medium/Sec.    curl-minimal-8.0.1-1.amzn2023.x86_64
ALAS2023-2023-225 Medium/Sec.    glib2-2.74.7-688.amzn2023.0.1.x86_64
ALAS2023-2023-195 Low/Sec.       libcap-2.48-2.amzn2023.0.3.x86_64
ALAS2023-2023-193 Medium/Sec.    libcurl-minimal-8.0.1-1.amzn2023.x86_64
ALAS2023-2023-145 Low/Sec.       libgcc-11.3.1-4.amzn2023.0.3.x86_64
ALAS2023-2023-145 Low/Sec.       libgomp-11.3.1-4.amzn2023.0.3.x86_64
ALAS2023-2023-145 Low/Sec.       libstdc++-11.3.1-4.amzn2023.0.3.x86_64
ALAS2023-2023-163 Medium/Sec.    libxml2-2.10.4-1.amzn2023.0.1.x86_64
ALAS2023-2023-220 Important/Sec. ncurses-base-6.2-4.20200222.amzn2023.0.4.noarch
ALAS2023-2023-220 Important/Sec. ncurses-libs-6.2-4.20200222.amzn2023.0.4.x86_64
ALAS2023-2023-181 Medium/Sec.    openssl-libs-1:3.0.8-1.amzn2023.0.2.x86_64
ALAS2023-2023-222 Medium/Sec.    openssl-libs-1:3.0.8-1.amzn2023.0.3.x86_64
```

Listing advisories in the latest version

在这个例子中，我们将看看如果我们启动了 202 3.4.20 240319 latest 版本的容器，那么在 AL2 023 版本中有哪些可用的更新。在撰写本文时，latest版本为 2023.5.20240708，因此本示例中列出的更新将以该版本为准。

> **Note**
>
> 此示例使用 2023.4.20240319 和 2023.5.20240 708 版本，后者是撰写本文时的最新版本。有关最新版本的更多信息，请参阅 AL2023 版本说明。

在此示例中，我们将从 2023.4.20 240319 版本的容器镜像开始。

首先，我们从容器注册表中获取此容器镜像。末尾的表示特定版本的图像版本。.1虽然图像版本通常为零，但此示例使用的是图像版本为一个的版本。

```
$ docker pull public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
   2023.4.20240319.1: Pulling from amazonlinux/amazonlinux
6de065fda9a2: Pull complete
Digest: sha256:b4838c4cc9211d966b6ea158dacc9eda7433a16ba94436508c2d9f01f7658b4e
Status: Downloaded newer image for public.ecr.aws/amazonlinux/
amazonlinux:2023.4.20240319.1
public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
```

我们现在可以在容器内生成一个外壳，我们将从中检查更新。

```
$ docker run -it public.ecr.aws/amazonlinux/amazonlinux:2023.4.20240319.1
   bash-5.2#
```

现在，该dnf updateinfo命令用于显示最新版本中哪些公告与我们已安装的软件包相关的摘要。在撰写本文时，2023.1.20230628是最新版本。

```
$ dnf --releasever=latest updateinfo
Amazon Linux 2023 repository                      76 MB/s |  25 MB     00:00
Last metadata expiration check: 0:00:04 ago on Mon Jul 22 20:59:54 2024.
Updates Information Summary: available
    9 Security notice(s)
        4 Important Security notice(s)
        4 Medium Security notice(s)
        1 Low Security notice(s)
```

要获取公告清单，可以--list选择给。dnf updateinfo

```
$ dnf updateinfo --releasever=latest --list
Last metadata expiration check: 0:00:58 ago on Mon Jul 22 20:59:54 2024.
ALAS2023-2024-581 Low/Sec.       curl-minimal-8.5.0-1.amzn2023.0.3.x86_64
ALAS2023-2024-596 Medium/Sec.    curl-minimal-8.5.0-1.amzn2023.0.4.x86_64
ALAS2023-2024-576 Important/Sec. expat-2.5.0-1.amzn2023.0.4.x86_64
ALAS2023-2024-589 Important/Sec. glibc-2.34-52.amzn2023.0.10.x86_64
ALAS2023-2024-589 Important/Sec. glibc-common-2.34-52.amzn2023.0.10.x86_64
ALAS2023-2024-589 Important/Sec. glibc-minimal-langpack-2.34-52.amzn2023.0.10.x86_64
ALAS2023-2024-586 Medium/Sec.    krb5-libs-1.21-3.amzn2023.0.4.x86_64
ALAS2023-2024-581 Low/Sec.       libcurl-minimal-8.5.0-1.amzn2023.0.3.x86_64
```

```
ALAS2023-2024-596 Medium/Sec.    libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64
ALAS2023-2024-592 Important/Sec. libnghttp2-1.59.0-3.amzn2023.0.1.x86_64
ALAS2023-2024-640 Medium/Sec.    openssl-libs-1:3.0.8-1.amzn2023.0.12.x86_64
ALAS2023-2024-605 Medium/Sec.    python3-3.9.16-1.amzn2023.0.7.x86_64
ALAS2023-2024-616 Important/Sec. python3-3.9.16-1.amzn2023.0.8.x86_64
ALAS2023-2024-605 Medium/Sec.    python3-libs-3.9.16-1.amzn2023.0.7.x86_64
ALAS2023-2024-616 Important/Sec. python3-libs-3.9.16-1.amzn2023.0.8.x86_64
```

就地应用安全更新

有关应用更新的概述，请参阅使用应用安全更新 DNF 和存储库版本。`--security`选项会`dnf upgrade`将软件包更新限制为仅包含建议的软件包。本节的其余部分将介绍如何仅安装特定的安全更新。

> **Note**
>
> 建议应用新 AL2 023 版本中的所有可用更新。只选择安全更新，或者只选择特定的更新应该是例外情况，而不是规则。

应用公告中提及的更新

输出第一列中的建议标识符`dnf upgradeinfo`可用于为公告中提及的软件包应用更新。可以指示`dnf`软件包管理器将公告中的软件包更新到最新的可用版本，或者仅更新到公告中提及的版本。如果已经安装了更新，则更新命令不起作用。

要仅在公告中提及的版本之前对受影响的软件包应用更新，请在使用`--advisory`选项指定公告的同时使用`dnf upgrade-minimal`命令。以下示例`dnf upgrade-minimal`在 AL2 023 版本 2023.0.20230315 容器中运行。

```
$ dnf upgrade-minimal -y --releasever=2023.1.20230628 --advisory ALAS2023-2023-193
Amazon Linux 2023 repository                     46 MB/s |  15 MB     00:00
Last metadata expiration check: 0:00:03 ago on Mon Jul 22 20:36:13 2024.
Dependencies resolved.
================================================================================
 Package              Arch        Version                Repository        Size
================================================================================
Upgrading:
 curl-minimal         x86_64      8.0.1-1.amzn2023       amazonlinux      150 k
```

```
 libcurl-minimal      x86_64      8.0.1-1.amzn2023       amazonlinux      249 k

Transaction Summary
================================================================================
Upgrade  2 Packages

Total download size: 399 k
Downloading Packages:
(1/2): curl-minimal-8.0.1-1.amzn2023.x86_64.rpm 2.7 MB/s | 150 kB     00:00
(2/2): libcurl-minimal-8.0.1-1.amzn2023.x86_64. 3.8 MB/s | 249 kB     00:00
--------------------------------------------------------------------------------
Total                                           2.5 MB/s | 399 kB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Upgrading        : libcurl-minimal-8.0.1-1.amzn2023.x86_64                1/4
  Upgrading        : curl-minimal-8.0.1-1.amzn2023.x86_64                   2/4
  Cleanup          : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64              3/4
  Cleanup          : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64           4/4
  Running scriptlet: libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64           4/4
  Verifying        : libcurl-minimal-8.0.1-1.amzn2023.x86_64                1/4
  Verifying        : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64           2/4
  Verifying        : curl-minimal-8.0.1-1.amzn2023.x86_64                   3/4
  Verifying        : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64              4/4

Upgraded:
  curl-minimal-8.0.1-1.amzn2023.x86_64  libcurl-minimal-8.0.1-1.amzn2023.x86_64

Complete!
```

即使使用相同的软件包版本`--releasever=latest`是`dnf`为了进行解决公告所需的最低限度更新，也会进行更新。

使用带`--advisory`选项的常规`dnf upgrade`命令会将公告中提及的相关软件包更新到可用的最新版本，该版本可能比公告中提到的版本要新。

> **Note**
>
> 除非更新`system-release`软件包，否则锁定到`dnf`的 AL2 023 存储库的版本不会更改。

> **Warning**
>
> 在不更改锁定到的存储库版本的情况下安装不同版本 AL2 023 的更新时，必须注意后续的任何变dnf更操作。dnf例如，在安装或更新软件包时，由于软件包依赖关系在新版本中可能已更改，因此您保留的旧版本可能无法满足这些新的依赖关系。

以下示例在 AL2 023 版本 2023.0.20230315 容器中运行，该容器引用了 023 的最新版本，撰写时为 2023.5.20 24 AL2 0708。请注意，两个更新到curl的版本都比更新到的版本update-minimal更新，但是这个较新的版本带来了新的依赖关系。

```
$ dnf upgrade -y --releasever=latest --advisory ALAS2023-2023-193
Amazon Linux 2023 repository                     80 MB/s |  25 MB     00:00
Last metadata expiration check: 0:00:04 ago on Mon Jul 22 20:48:38 2024.
Dependencies resolved.
================================================================================
 Package                   Arch     Version                 Repository     Size
================================================================================
Upgrading:
 curl-minimal              x86_64   8.5.0-1.amzn2023.0.4    amazonlinux   160 k
 libcurl-minimal           x86_64   8.5.0-1.amzn2023.0.4    amazonlinux   275 k
 libnghttp2                x86_64   1.59.0-3.amzn2023.0.1   amazonlinux    79 k
Installing dependencies:
 libpsl                    x86_64   0.21.1-3.amzn2023.0.2   amazonlinux    61 k
 publicsuffix-list-dafsa   noarch   20240212-61.amzn2023    amazonlinux    59 k

Transaction Summary
================================================================================
Install  2 Packages
Upgrade  3 Packages

Total download size: 634 k
Downloading Packages:
(1/5): publicsuffix-list-dafsa-20240212-61.amzn 1.1 MB/s |  59 kB     00:00
(2/5): curl-minimal-8.5.0-1.amzn2023.0.4.x86_64 2.6 MB/s | 160 kB     00:00
(3/5): libpsl-0.21.1-3.amzn2023.0.2.x86_64.rpm  949 kB/s |  61 kB     00:00
(4/5): libnghttp2-1.59.0-3.amzn2023.0.1.x86_64. 3.7 MB/s |  79 kB     00:00
(5/5): libcurl-minimal-8.5.0-1.amzn2023.0.4.x86 6.7 MB/s | 275 kB     00:00
--------------------------------------------------------------------------------
Total                                           3.5 MB/s | 634 kB     00:00
Running transaction check
Transaction check succeeded.
```

```
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Upgrading        : libnghttp2-1.59.0-3.amzn2023.0.1.x86_64                1/8
  Installing       : publicsuffix-list-dafsa-20240212-61.amzn2023.noarch    2/8
  Installing       : libpsl-0.21.1-3.amzn2023.0.2.x86_64                    3/8
  Upgrading        : libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64            4/8
  Upgrading        : curl-minimal-8.5.0-1.amzn2023.0.4.x86_64               5/8
  Cleanup          : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64              6/8
  Cleanup          : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64           7/8
  Cleanup          : libnghttp2-1.51.0-1.amzn2023.x86_64                    8/8
  Running scriptlet: libnghttp2-1.51.0-1.amzn2023.x86_64                    8/8
  Verifying        : libpsl-0.21.1-3.amzn2023.0.2.x86_64                    1/8
  Verifying        : publicsuffix-list-dafsa-20240212-61.amzn2023.noarch    2/8
  Verifying        : curl-minimal-8.5.0-1.amzn2023.0.4.x86_64               3/8
  Verifying        : curl-minimal-7.88.1-1.amzn2023.0.1.x86_64              4/8
  Verifying        : libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64            5/8
  Verifying        : libcurl-minimal-7.88.1-1.amzn2023.0.1.x86_64           6/8
  Verifying        : libnghttp2-1.59.0-3.amzn2023.0.1.x86_64                7/8
  Verifying        : libnghttp2-1.51.0-1.amzn2023.x86_64                    8/8

Upgraded:
  curl-minimal-8.5.0-1.amzn2023.0.4.x86_64
  libcurl-minimal-8.5.0-1.amzn2023.0.4.x86_64
  libnghttp2-1.59.0-3.amzn2023.0.1.x86_64
Installed:
  libpsl-0.21.1-3.amzn2023.0.2.x86_64
  publicsuffix-list-dafsa-20240212-61.amzn2023.noarch

Complete!
```

为 AL2 0 SELinux 23 设置模式

默认情况下，安全增强型 Linux (SELinux) 为`enabled`并设置为 AL2 023 的`permissive`模式。在宽容模式下，权限拒绝会被记录下来，但不会强制执行。 SELinux 是内核功能和实用程序的集合，用于为内核的主要子系统提供强大、灵活的强制访问控制 (MAC) 架构。

SELinux 提供了一种增强的机制，用于根据保密和完整性要求强制信息分离。这种信息分离减少了篡改和绕过应用程序安全机制的威胁，也限制了恶意或有缺陷的应用程序可能导致的损害。

SELinux 包括一组安全策略配置文件示例，旨在满足日常安全目标。

有关 SELinux 特性和功能的更多信息，请参阅 “SELinux 笔记本和策略语言”。

主题

- AL2023 的默认 SELinux 状态和模式
- 改为 enforcing 模式
- 为 AL2 023 禁 SELinux 用的选项

AL2023 的默认 SELinux 状态和模式

对于 AL2 023 SELinux ，默认为enabled并设置为permissive模式。在 permissive 模式下，记录但不强制执行权限拒绝。

getenforce或**sestatus**命令告诉你当前的 SELinux状态、策略和模式。

当默认状态设置为 enabled 和 permissive 时，**getenforce** 命令返回 permissive。

该**sestatus**命令返回 SELinux 状态和当前 SELinux 策略，如以下示例所示：

```
$ sestatus
SELinux status:                 enabled
  SELinuxfs mount:                /sys/fs/selinux
  SELinux root directory:         /etc/selinux
  Loaded policy name:             targeted
  Current mode:                   permissive
  Mode from config file:          permissive
  Policy MLS status:              enabled
  Policy deny_unknown status:     allowed
  Memory protection checking:     actual (secure)
  Max kernel policy version:      33
```

当您在permissive模式 SELinux 下运行时，用户可能会错误地标记文件。当你 SELinux 以disabled状态运行时，文件不会被标记。当您改为 enforcing 模式时，不正确或未标记的文件都可能导致问题。

SELinux 自动重新标记文件以避免出现此问题。 SELinux 当您将状态更改为时，自动重新贴标签可防止出现标签问题。enabled

改为 enforcing 模式

当你跑步的时候 SELinux 在enforcing模式下，SELinux 实用程序enforcing是配置的策略。SELinux 通过根据策略的规则允许或拒绝访问来控制选定应用程序的功能。

要查找当前 SELinux 模式，运行getenforce命令。

```
getenforce
Permissive
```

编辑配置文件以启用 enforcing 模式

要将模式更改为enforcing，请执行以下步骤。

1. 编辑 /etc/selinux/config 文件以改为 enforcing 模式。该SELINUX设置应类似于以下示例。

   ```
   SELINUX=enforcing
   ```

2. 重新启动系统以完成改为 enforcing 模式。

   ```
   $ sudo reboot
   ```

下次开机时，SELinux 重新标记系统中的所有文件和目录。SELinux 还添加了 SELinux 创建时创建的文件和目录的上下文 SELinux 是disabled。

切换到enforcing模式后，SELinux 可能会因为不正确或缺失而拒绝某些操作 SELinux 政策规则。您可以查看以下操作 SELinux 使用以下命令拒绝。

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

使用 cloud-init 启用enforcing模式

或者，当您启动实例时，请将以下 cloud-config 作为用户数据传递以启用 enforcing 模式。

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

默认情况下，此设置会导致实例重启。为了提高稳定性，建议您重启实例。但是，如果您愿意，可以通过提供以下 `cloud-config` 来跳过重新启动。

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

为 AL2 023 禁 SELinux 用的选项

当你禁用时 SELinux, SELinux 未加载或强制执行策略，也未记录访问矢量缓存 (AVC) 消息。你失去了跑步的所有好处 SELinux.

而不是禁用 SELinux，我们建议使用`permissive`模式。在模式下运行的成本仅比禁用`permissive`模式高一点 SELinux 完全。与禁用后过渡回`permissiveenforcing`模式相比，从模式过渡到`enforcing`模式所需的配置调整要少得多 SELinux。 您可以标记文件，系统可以跟踪和记录活动策略可能拒绝的操作。

更改 SELinux 进入**`permissive`**模式

当你跑步的时候 SELinux 在`permissive`模式下，SELinux 策略未得到执行。在`permissive`模式下，SELinux 记录 AVC 消息但不拒绝操作。您可以使用这些 AVC 消息进行故障排除、调试和 SELinux 政策改进。

要改变 SELinux 要进入宽容模式，请使用以下步骤。

1. 编辑 `/etc/selinux/config` 文件以改为 `permissive` 模式。该`SELINUX`值应类似于以下示例。

   ```
   SELINUX=permissive
   ```

2. 重新启动系统以完成改为 `permissive` 模式。

   ```
   sudo reboot
   ```

禁用 SELinux

当你禁用时 SELinux, SELinux 未加载或强制执行策略，也不会记录 AVC 消息。你失去了跑步的所有好处 SELinux.

要禁用 SELinux，请使用以下步骤。

1. 确保grubby软件包已安装。

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

2. 配置您的引导加载程序以将 selinux=0 添加到内核命令行。

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

3. 重新启动系统。

```
sudo reboot
```

4. 运行getenforce 命令进行确认 SELinux 是Disabled。

```
$ getenforce
Disabled
```

有关 SELinux，请参阅 SELinux 笔记本和 SELinux 配置。

在 023 上 AL2启用 FIPS 模式

本节介绍如何在 AL2 023 上启用《联邦信息处理标准》(FIPS)。有关 FIPS 的更多信息，请参阅：

- 美国联邦信息处理标准 (FIPS)
- 合规性 FAQs：联邦信息处理标准

> **Note**
>
> 本节记录了如何启用 FIPS AL2023 中的模式，它不涵盖 AL2 023 加密模块的认证状态。

先决条件

- 现有的 AL2 023（AL2023.2 或更高版本）Amazon EC2 实例，可以访问互联网下载所需软件包。有关启动 AL2 023 Amazon EC2 实例的更多信息，请参阅使用亚马逊 EC2 控制台启动 AL2 023。
- 您必须使用 SSH 或连接到您的 Amazon EC2 实例 AWS Systems Manager。有关更多信息，请参阅正在连接到 AL2 023 个实例。

> **Important**
>
> ED25519 FIPS 模式下不支持 SSH 用户密钥。如果您使用 ED25519 SSH 密钥对启动 Amazon EC2 实例，则必须使用其他算法（例如 RSA）生成新密钥，否则在启用 FIPS 模式后您可能会失去对实例的访问权限。有关更多信息，请参阅 Amazon EC2 用户指南中的创建密钥对。

启用 FIPS 模式

1. 使用 SSH 或 AWS Systems Manager Connect 连接到您的 AL2 023 实例。
2. 确保系统是最新版本。有关更多信息，请参阅 在 AL2 023 中管理软件包和操作系统更新。
3. 确保安装了`crypto-policies`实用程序，然后 up-to-date。

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

4. 通过运行以下命令来启用 FIPS 模式。这将为 AL20 23 常见问题解答中列出的模块在系统范围内启用 FIPS 模式

```
sudo fips-mode-setup --enable
```

5. 使用以下命令重新引导实例。

```
sudo reboot
```

6. 要验证是否已启用 FIPS 模式，请重新连接到实例并运行以下命令。

```
sudo fips-mode-setup --check
```

以下示例输出显示已启用 FIPS 模式：

```
FIPS mode is enabled.
```

在 AL2 023 容器中启用 FIPS 模式

本节介绍如何在 AL2 023 容器中启用联邦信息处理标准 (FIPS)。有关 FIPS 的更多信息，请参阅：

- 美国联邦信息处理标准 (FIPS)

- 合规性 FAQs：联邦信息处理标准

> **Note**
>
> 本节记录了如何启用 FIPS AL2023 容器中的模式。它不包括 AL2 023 加密模块的认证状态。

先决条件

- 现有的 AL2 023（AL2023.2 或更高版本）Amazon EC2 实例，可以访问互联网下载所需软件包。有关启动 AL2 023 Amazon EC2 实例的更多信息，请参阅使用亚马逊 EC2 控制台启动 AL2 023。
- 您必须使用 SSH 或连接到您的 Amazon EC2 实例 AWS Systems Manager。有关更多信息，请参阅正在连接到 AL2 023 个实例。

> **Important**
>
> 该`fips-mode-setup`命令将无法在容器内正常运行。请阅读以下步骤，在 AL2 023 容器中正确配置 FIPS 模式。

在 AL2 023 容器中启用 FIPS 模式

1. 必须先在 AL2 023 容器主机上启用 FIPS 模式。按照中的说明在主机在 023 上 AL2启用 FIPS 模式上启用 FIPS 模式。
2. 使用 SSH 或 AWS Systems Manager Connect 连接到您的 AL2 023 容器主机实例。
3. 如果 AL2 023 主机处于 FIPS 模式并且可以从容器内部访问，则 AL2 023 容器中将自动启用 FIPS 模式。`/proc/sys/crypto/fips_enabled`如果的内容`/proc/sys/crypto/fips_enabled`为`0`，则不启用 FIPS，值为`1`表示已启用 FIPS 模式。

   您可以通过在 AL2 023 主机和容器上运行以下命令来验证 FIPS 是否已启用：

   ```
   cat /proc/sys/crypto/fips_enabled
   ```

4. 接下来，在容器内启用 FIPS crypto-policies。有几种方法可以实现此目的，如以下选项所述。使用最适合您的环境的选项。

   a. 使用以下命令在容器内手动启用 FIPS crypto-policies：`update-crypto-policies`

```
# Run these commands inside the container
dnf install -y crypto-policies-scripts
update-crypto-policies --set FIPS
```

b. 在 AL2 023 容器中创建bind挂载（这与其他发行版中的podman工作方式类似）：

```
# Run these commands inside the container
mount --bind /usr/share/crypto-policies/back-ends/FIPS /etc/crypto-policies/back-
ends
echo "FIPS" > /usr/share/crypto-policies/default-fips-config
mount --bind /usr/share/crypto-policies/default-fips-config /etc/crypto-policies/
config
```

c. 也可以创建绑定挂载，以便 AL2 023 容器与 AL2 023 主机的加密策略相匹配。以下仅作为示例提供。如果容器和主机之间的 crypto-policies 和包版本存在不兼容的差异，则此配置可能会导致问题：

```
sudo docker pull amazonlinux:2023
sudo docker run --mount type=bind,readonly,src=/etc/crypto-policies,dst=/etc/
crypto-policies -it amazonlinux:2023
```

5. 执行上述步骤后，您可以使用以下命令再次验证容器中是否已启用 FIPS：

```
$ cat /etc/crypto-policies/config
FIPS

$ cat /proc/sys/crypto/fips_enabled
1
```

在 023 上交换 OpenSSL FIPS 提供商 AL2

本节介绍如何在 023 上在 OpenSSL FIPS 提供程序`latest`和 `certified` OpenSSL FIPS 提供程序之间切换。 AL2

有关 FIPS 的更多信息，请参阅：

- 美国联邦信息处理标准 (FIPS)
- 合规性 FAQs：联邦信息处理标准

- 关于加密模块选择和使用的 FedRAMP 政策

> **Important**
>
> 在 AL2 023.7 及更高版本中，默认的 OpenSSL FIPS 提供程序是`openssl-fips-provider-latest`软件包，它会定期接收错误修复和安全更新。
> 以下说明仅适用于想要固定在`openssl-fips-provider-certified`包裹上的客户。此版本的 FIPS 提供程序将与 NIST 证书上的校验和相匹配，并且可能没有最新的更新。
> 有关 FIPS 认证模块和软件包版本的更多信息，请参阅 AL2023 常见问题解答。

先决条件

- 现有的 AL2 023（AL2023.7 或更高版本）Amazon EC2 实例，可以访问互联网下载所需软件包。有关启动 AL2 023 Amazon EC2 实例的更多信息，请参阅使用亚马逊 EC2 控制台启动 AL2 023。
- 您必须使用 SSH 或连接到您的 Amazon EC2 实例 AWS Systems Manager。有关更多信息，请参阅正在连接到 AL2 023 个实例。
- 要在 AL2 023 上启用 FIPS 模式，请按照中的说明进行操作。在 023 上 AL2启用 FIPS 模式

在`openssl-fips-provider-latest`和之间切换 `openssl-fips-provider-certified`

1. 用于切换 `dnf` OpenSSL FIPS 提供程序：

```
sudo dnf -y swap openssl-fips-provider-latest openssl-fips-provider-certified
```

2. 检查您是否使用的是经过认证的 OpenSSL FIPS 提供商。在 AL2 023 处于 FIPS 模式下，运行以下命令：

```
openssl list -providers
```

   您应看到以下输出：

```
Providers:
  base
    name: OpenSSL Base Provider
    version: 3.2.2
    status: active
```

```
  default
    name: OpenSSL Default Provider
    version: 3.2.2
    status: active
  fips
    name: Amazon Linux 2023 - OpenSSL FIPS Provider
    version: 3.0.8-d694bfa693b76001
    status: active
```

AL2023 内核强化

AL2023 中的 6.1 Linux 内核配置和构建时包含多个强化选项和功能。

内核强化选项（与架构无关）

CONFIG 选项	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_ACPI_CUSTOM_METHOD	n	n	不适用	不适用
CONFIG_BINFMT_MISC	m	m	m	m
CONFIG_BUG	y	y	y	y
CONFIG_BUG_ON_DATA_CORRUPTION	y	y	y	y
CONFIG_CFI_CLANG	不适用	不适用	不适用	不适用
CONFIG_CFI_PERMISSIVE	不适用	不适用	不适用	不适用

CONFIG 选项	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_COMPAT	y	y	y	y
CONFIG_COMPAT_BRK	n	n	n	n
CONFIG_COMPAT_VDSO	不适用	n	不适用	n
CONFIG_DEBUG_CREDENTIALS	n	n	不适用	不适用
CONFIG_DEBUG_LIST	y	y	y	y
CONFIG_DEBUG_NOTIFIERS	n	n	n	n
CONFIG_DEBUG_SG	n	n	n	n
CONFIG_DEBUG_VIRTUAL	n	n	n	n
CONFIG_DEBUG_WX	n	n	n	n
CONFIG_DEFAULT_MMAP_MIN_ADDR	65536	65536	65536	65536
CONFIG_DEVKMEM	不适用	不适用	不适用	不适用

CONFIG 选项	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_DEVMEM	n	n	n	n
CONFIG_EFI_DISABLE_PCI_DMA	n	n	n	n
CONFIG_FORTIFY_SOURCE	y	y	y	y
CONFIG_HARDENED_USERCOPY	y	y	y	y
CONFIG_HARDENED_USERCOPY_FALLBACK	不适用	不适用	不适用	不适用
CONFIG_HARDENED_USERCOPY_PAGESPAN	不适用	不适用	不适用	不适用
CONFIG_HIBERNATION	y	y	y	y
CONFIG_HW_RANDOM_TPM	不适用	不适用	不适用	不适用
CONFIG_INET_DIAG	m	m	m	m

CONFIG 选项	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_INIT_ON_ALLOC_DEFAULT_ON	n	n	n	n
CONFIG_INIT_ON_FREE_DEFAULT_ON	n	n	n	n
CONFIG_INIT_STACK_ALL_ZERO	不适用	不适用	不适用	不适用
CONFIG_IOMMU_DEFAULT_DMA_STRICT	n	n	n	n
CONFIG_IOMMU_SUPPORT	y	y	y	y
CONFIG_IO_STRICT_DEVMEM	不适用	不适用	不适用	不适用
CONFIG_KEXEC	y	y	y	y
CONFIG_KFENCE	n	n	n	n

CONFIG 选项	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_LDISC_AUTOLOAD	n	n	n	n
CONFIG_LEGACY_PTYS	n	n	n	n
CONFIG_LOCK_DOWN_KERNEL_FORCE_CONFIDENTIALITY	n	n	n	n
CONFIG_MODULES	y	y	y	y
CONFIG_MODULE_SIG	y	y	y	y
CONFIG_MODULE_SIG_ALL	y	y	y	y
CONFIG_MODULE_SIG_FORCE	n	n	n	n
CONFIG_MODULE_SIG_HASH	sha512	sha512	sha512	sha512
CONFIG_MODULE_SIG_KEY	certs/signing_key.pem	certs/signing_key.pem	certs/signing_key.pem	certs/signing_key.pem

CONFIG 选项	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_MODULE_SIG_SHA512	y	y	y	y
CONFIG_PAGE_POISONING	n	n	n	n
CONFIG_PAGE_POISONING_NO_SANITY	不适用	不适用	不适用	不适用
CONFIG_PAGE_POISONING_ZERO	不适用	不适用	不适用	不适用
CONFIG_PANIC_ON_OOPS	y	y	y	y
CONFIG_PANIC_TIMEOUT	0	0	0	0
CONFIG_PROC_KCORE	y	y	y	y
CONFIG_RANDOMIZE_KSTACK_OFFSET_DEFAULT	n	n	n	n

CONFIG 选项	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_RANDOM_TRUST_BOOTLOADER	y	y	不适用	不适用
CONFIG_RANDOM_TRUST_CPU	y	y	不适用	不适用
CONFIG_REFCOUNT_FULL	不适用	不适用	不适用	不适用
CONFIG_SCHED_CORE	不适用	y	不适用	y
CONFIG_SCHED_STACK_END_CHECK	y	y	y	y
CONFIG_SECCOMP	y	y	y	y
CONFIG_SECCOMP_FILTER	y	y	y	y
CONFIG_SECURITY	y	y	y	y
CONFIG_SECURITY_DMESG_RESTRICT	y	y	y	y

CONFIG 选项	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_SECURITY_LANDLOCK	n	n	n	n
CONFIG_SECURITY_LOCKDOWN_LSM	y	y	y	y
CONFIG_SECURITY_LOCKDOWN_LSM_EARLY	y	y	y	y
CONFIG_SECURITY_SELINUX_BOOTPARAM	y	y	y	y
CONFIG_SECURITY_SELINUX_DEVELOP	y	y	y	y
CONFIG_SECURITY_SELINUX_DISABLE	n	n	不适用	不适用
CONFIG_SECURITY_WRITABLE_HOOKS	不适用	不适用	不适用	不适用

CONFIG 选项	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_SECURITY_YAMA	y	y	y	y
CONFIG_SHUFFLE_PAGE_ALLOCATOR	y	y	y	y
CONFIG_SLAB_FREELIST_HARDENED	y	y	y	y
CONFIG_SLAB_FREELIST_RANDOM	y	y	y	y
CONFIG_SLUB_DEBUG	y	y	y	y
CONFIG_STACKPROTECTOR	y	y	y	y
CONFIG_STACKPROTECTOR_STRONG	y	y	y	y
CONFIG_STATIC_USERMODEHELPER	n	n	n	n

CONFIG 选项	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_STRICT_DEVMEM	n	n	n	n
CONFIG_STRICT_KERNEL_RWX	y	y	y	y
CONFIG_STRICT_MODULE_RWX	y	y	y	y
CONFIG_SYN_COOKIES	y	y	y	y
CONFIG_VMAP_STACK	y	y	y	y
CONFIG_WERROR	n	n	n	n
CONFIG_ZERO_CALL_USED_REGS	n	n	n	n

允许在运行时插入/替换 ACPI 方法 (CONFIG_ACPI_CUSTOM_METHOD)

Amazon Linux 禁用了此选项，因为它允许 `root` 用户写入任意内核内存。

此选项是内核自我保护项目推荐设置之一。

其他二进制格式 (**binfmt_misc**)

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。在 AL2 023 中，此功能是可选的，是作为内核模块构建的。

BUG() 支持

此选项是内核自我保护项目推荐设置之一。

如果内核在检查内核内存结构的有效性时遇到数据损坏，则执行 BUG()

Linux 内核的某些部分会检查数据结构的内部一致性，并在检测到数据损坏执行 BUG()。

此选项是内核自我保护项目推荐设置之一。

COMPAT_BRK

此选项禁用后（这就是 Amazon Linux 配置内核的方式），则 randomize_va_space sysctl 设置会默认为 2，这也会在 mmap 基础地址、堆栈和 VDSO 页面随机化之上启用随机掩码。

此选项存在于内核中，目的是提供与 1996 年及更早版本的一些古老 libc.so.5 二进制文件的兼容性。

此选项是内核自我保护项目推荐设置之一。

COMPAT_VDSO

此配置选项与 x86-64 有关，与 aarch64 无关。将此设置为 n，Amazon Linux 内核不会在可预测的地址上显示 32 位虚拟动态共享对象 (VDSO)。自 2004 年以来，已知因该选项设置为 n 而被损坏的最新 glibc 版本是 glibc 2.3.3。

此选项是内核自我保护项目推荐设置之一。

CONFIG_DEBUG 门控强化

受限的 Linux 内核配置选项CONFIG_DEBUG通常设计用于为调试问题而构建的内核，性能等问题不是优先事项。 AL2023 启用CONFIG_DEBUG_LIST强化选项。

在配置 IOMMU 之前，禁用 EFI 存根中 PCI 设备的 DMA

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。

用于在内核和用户空间之间复制内存的强化功能

当内核需要将内存复制到用户空间或从用户空间复制内存时，此选项会启用一些检查，可防范某些类别的堆溢出问题。

CONFIG_HARDENED_USERCOPY_FALLBACK 选项存在于内核 4.16 到 5.15 中，可帮助内核开发人员通过 WARN( ) 发现任何缺失的允许列表条目。由于 AL2 023 附带了 6.1 内核，因此此选项与 AL2 023 不再相关。

该CONFIG_HARDENED_USERCOPY_PAGESPAN选项主要作为开发人员的调试选项存在于内核中，在 AL2 023 年不再适用于 6.1 内核。

此选项是内核自我保护项目推荐设置之一。

休眠支持

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。需要启用此选项才能支持将休眠按需型实例并支持休眠中断的竞价型实例

随机数生成

AL2023 内核配置为确保有足够的熵可供内部使用。 EC2

CONFIG_INET_DIAG

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。在 AL2 023 中，此功能是可选的，是作为内核模块构建的。

在分配和取消分配时，将所有内核页和 slab 分配器内存设置为零

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。由于默认启用此功能可能会影响性能，因此这些选项在 AL2 023 中处于禁用状态。CONFIG_INIT_ON_ALLOC_DEFAULT_ON 行为可以通过将 init_on_alloc=1 添加到内核命令行来启用，CONFIG_INIT_ON_FREE_DEFAULT_ON 行为可以通过添加 init_on_free=1 来启用。

将所有堆栈变量初始化为零 (CONFIG_INIT_STACK_ALL_ZERO)

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。此选项需要 GCC 12 或更高，而 AL2 023 则搭载海湾合作委员会 11。

内核模块签名

AL2023 对内核模块的签名进行签名和验证。为了保持构建第三方模块的用户的兼容性，尚未启用要求模块具有有效签名的 CONFIG_MODULE_SIG_FORCE 选项。对于想要确保所有内核模块都经过签名的用户，可以配置 锁定 Linux 安全模块 (LSM) 以强制执行此操作。

kexec

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。启用此选项是为了便于使用 kdump 功能。

IOMMU Support

AL2023 启用 IOMMU 支持。默认情况下，CONFIG_IOMMU_DEFAULT_DMA_STRICT 选项未启用，但可以通过将 iommu.passthrough=0 iommu.strict=1 添加到内核命令行来配置此功能。

kfence

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。

传统 pty 支持

AL2023 使用现代 PTY 接口 (devpts)。

此选项是内核自我保护项目推荐设置之一。

锁定 Linux 安全模块 (LSM)

AL2023 构建 lockdown LSM，它将在使用安全启动时自动锁定内核。

CONFIG_LOCK_DOWN_KERNEL_FORCE_CONFIDENTIALITY 选项未启用。尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。不使用安全启动时，可以启用锁定 LSM 并根据需要进行配置。

页面中毒

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。同样 在分配和取消分配时，将所有内核页和 slab 分配器内存设置为零 ，由于这可能会影响性能，因此在 AL2 023 内核中禁用了该功能。

堆栈保护器

AL2023 内核是使用堆栈保护器功能构建的 GCC 使用-fstack-protector-strong选项启用。

此选项是内核自我保护项目推荐设置之一。

seccomp BPF API

这些区域有：seccomp 诸如systemd和容器运行时之类的软件使用强化功能来强化用户空间应用程序。

此选项是内核自我保护项目推荐设置之一。

panic() 超时

AL2023 内核配置此值设置为0，这意味着内核在死机后不会重新启动。尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。这可以通过 sysctl、/proc/sys/kernel/panic 和在内核命令行上进行配置。

安全模型

AL2默认情况下，023 SELinux 在允许模式下启用。有关更多信息，请参阅 为 AL2 0 SELinux 23 设置模式。

锁定 Linux 安全模块 (LSM) 和 yama 模块也已启用。

/proc/kcore

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。

在系统调用进入时进行内核栈偏移随机化

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。这可以通过在内核命令行上设置 randomize_kstack_offset=on 来启用。

引用计数检查 (CONFIG_REFCOUNT_FULL)

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。由于此选项可能对性能产生影响，因此目前未启用。

调度程序意识到 SMT 核心 (CONFIG_SCHED_CORE)

AL2023 内核是用它构建的CONFIG_SCHED_CORE，允许用户空间应用程序使用。prctl(PR_SCHED_CORE)此选项是内核自我保护项目推荐设置之一。

在调用 **schedule()** 时检查是否存在栈损坏 (**CONFIG_SCHED_STACK_END_CHECK**)

AL2023 内核是在CONFIG_SCHED_STACK_END_CHECK启用状态下构建的。此选项是内核自我保护项目推荐设置之一。

内存分配器强化

AL2023 内核允许使用CONFIG_SHUFFLE_PAGE_ALLOCATOR、CONFIG_SLAB_FREELIST_HARDENED和选项对内核内存分配器进行强化。CONFIG_SLAB_FREELIST_RANDOM此选项是内核自我保护项目推荐设置之一。

SLUB 调试支持

AL2023 内核启用，CONFIG_SLUB_DEBUG因为此选项为分配器启用了可选的调试功能，这些功能可以在内核命令行上启用。此选项是内核自我保护项目推荐设置之一。

CONFIG_STATIC_USERMODEHELPER

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。这是因为 CONFIG_STATIC_USERMODEHELPER 需要该发行版的特殊支持，而 Amazon Linux 中目前没有这种支持。

只读内核文本和 rodata（**CONFIG_STRICT_KERNEL_RWX**和**CONFIG_STRICT_MODULE_RWX**）

AL2023 内核配置为标记内核和内核模块文本，以及 rodata 内存为只读，非文本内存标记为不可执行。此选项是内核自我保护项目推荐设置之一。

TCP syncookie 支持 () **CONFIG_SYN_COOKIES**

AL2023 内核构建时支持 TCP 同步 cookie。此选项是内核自我保护项目推荐设置之一。

带有保护页面的虚拟映射栈 (**CONFIG_VMAP_STACK**)

AL2023 内核是用构建的CONFIG_VMAP_STACK，支持带有保护页的虚拟映射内核堆栈。此选项是内核自我保护项目推荐设置之一。

使用编译器警告作为错误进行构建 (**CONFIG_WERROR**)

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。

在函数退出时进行寄存器清零 (CONFIG_ZERO_CALL_USED_REGS)

尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。

用户空间分配的最小地址

此强化选项可以帮助减少内核 NULL 指针错误的影响。此选项是内核自我保护项目推荐设置之一。

clang 特定的强化选项

AL2023 内核是用以下方式构建的 GCC 而不是 clang，因此无法启用CONFIG_CFI_CLANG强化选项，这也使之CONFIG_CFI_PERMISSIVE不适用。尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。

x86-64 特定的内核强化选项

CONFIG 选项	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_AMD_IOMMU	不适用	y	不适用	y
CONFIG_AMD_IOMMU_V2	不适用	y	不适用	不适用
CONFIG_IA32_EMULATION	不适用	y	不适用	y
CONFIG_INTEL_IOMMU	不适用	y	不适用	y
CONFIG_INTEL_IOMMU_DEFAULT_ON	不适用	n	不适用	n

CONFIG 选项	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_INTEL_IOMMU_SVM	不适用	n	不适用	n
CONFIG_LEGACY_VSYSCALL_NONE	不适用	n	不适用	n
CONFIG_MODIFY_LDT_SYSCALL	不适用	n	不适用	n
CONFIG_PAGE_TABLE_ISOLATION	不适用	y	不适用	不适用
CONFIG_RANDOMIZE_MEMORY	不适用	y	不适用	y
CONFIG_X86_64	不适用	y	不适用	y
CONFIG_X86_MSR	不适用	y	不适用	y
CONFIG_X86_VSYSCALL_EMULATION	不适用	y	不适用	y
CONFIG_X86_X32	不适用	不适用	不适用	不适用
CONFIG_X86_X32_ABI	不适用	n	不适用	n

x86-64 支持

基本 x86-64 支持包括物理地址扩展 (PAE) 和不执行 (NX) 位支持。此选项是内核自我保护项目推荐设置之一。

AMD 和 Intel IOMMU 支持

AL2023 内核构建时支持 AMD 和英特尔 IOMMUs。 此选项是内核自我保护项目推荐设置之一。

`CONFIG_INTEL_IOMMU_DEFAULT_ON` 选项未设置，但可以通过将 `intel_iommu=on` 传递到内核命令行来启用。尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。

该`CONFIG_INTEL_IOMMU_SVM`选项目前未在 AL2 023 中启用。尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。

支持 32 位用户空间

> **Important**
>
> 对 32 位 x86 用户空间的支持已弃用，在未来的主版本 Amazon Linux 中，可能会取消对运行 32 位用户空间二进制文件的支持。

> **Note**
>
> 虽然 AL2 023 不再包含任何 32 位软件包，但内核仍将支持运行 32 位用户空间。请参阅32 位 x86 (i686) 软件包了解更多信息。

为了支持运行 32 位用户空间应用程序， AL2023 不启用该`CONFIG_X86_VSYSCALL_EMULATION`选项，而是启用`CONFIG_IA32_EMULATIONCONFIG_COMPAT`、和选项。`CONFIG_X86_VSYSCALL_EMULATION`尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。

这些区域有：x32 未启用 64 位处理器的本机 32 位 ABI（`CONFIG_X86_X32`和）。`CONFIG_X86_X32_ABI`此选项是内核自我保护项目推荐设置之一。

x86 型号专用寄存器 (MSR) 支持

启用 `CONFIG_X86_MSR` 选项是为了支持 `turbostat`。尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。

`modify_ldt` 系统调用

AL2023 不允许用户程序使用 syscall 修改 x86 本地描述符表 (LDT)。`modify_ldt`此调用是运行 16 位或分段代码所必需的，缺少此调用可能会破坏软件`dosemu`，例如在下方运行某些程序 WINE，还有一些非常古老的线程库。此选项是内核自我保护项目推荐设置之一。

在用户模式下移除内核映射

AL2023 配置内核，使大多数内核地址不会映射到用户空间。此选项是内核自我保护项目推荐设置之一。

随机化内核内存部分

AL2023 将内核配置为随机化内核内存部分的基本虚拟地址。此选项是内核自我保护项目推荐设置之一。

aarch64 特定的内核强化选项

CONFIG 选项	AL2023/6.1/aarch64	AL2023/6.1/x86_64	AL2023/6.12/aarch64	AL2023/6.12/x86_64
CONFIG_ARM64_BTI	y	不适用	y	不适用
CONFIG_ARM64_BTI_KERNEL	不适用	不适用	不适用	不适用
CONFIG_ARM64_PTR_AUTH	y	不适用	y	不适用
CONFIG_ARM64_PTR_AUTH_KERNEL	y	不适用	y	不适用

CONFIG 选项	AL2023/6.1/ aarch64	AL2023/6.1/ x86_64	AL2023/6.12/ aarch64	AL2023/6.12/ x86_64
CONFIG_ARM64_SW_TTBR0_PAN	y	不适用	y	不适用
CONFIG_UNMAP_KERNEL_AT_EL0	y	不适用	y	不适用

分支目标识别

AL2023 内核支持分支目标识别 (CONFIG_ARM64_BTI)。此选项是内核自我保护项目推荐设置之一。

该CONFIG_ARM64_BTI_KERNEL选项在 AL2 023 中未启用，因为它是用构建的 GCC，并且由于 gcc 错误，上游内核中目前已禁用使用此选项构建内核的支持。尽管此选项是内核自我保护项目 (KSPP) 推荐设置之一，但 AL2 023 并未将此配置选项设置为 KSPP 建议的选项。

指针身份验证 (CONFIG_ARM64_PTR_AUTH)

AL2023 内核构建时支持指针身份验证扩展（ ARMv8.3 扩展的一部分），该扩展可用于帮助缓解面向返回的编程 (ROP) 技术。Graviton 3 引入了在 Graviton 上进行指针身份验证所需的硬件支持。

CONFIG_ARM64_PTR_AUTH 选项已启用，并支持针对用户空间的指针身份验证。由于该CONFIG_ARM64_PTR_AUTH_KERNEL选项也已启用，因此 AL2 023 内核可以自行使用返回地址保护。

此选项是内核自我保护项目推荐设置之一。

使用 TTBR0_EL1 切换的 Emulate Privileged Access Never

此选项可防止内核直接访问用户空间内存，TTBR0_EL1 只由用户访问例程临时设置为有效值。

此选项是内核自我保护项目推荐设置之一。

在用户空间中运行时取消内核映射

AL2023 内核配置为在用户空间 () 中运行时取消内核映射。CONFIG_UNMAP_KERNEL_AT_EL0此选项是内核自我保护项目推荐设置之一。

UEFI 在 023 上安全启动 AL2

AL2从 2023.1 版本开始，023 支持 UEFI 安全启动。您必须将 AL2 023 用于同时支持 UEFI 和 UEFI 安全启动的 Amazon EC2 实例。有关更多信息，请参阅《亚马逊 EC2 用户指南》中的 “在 UEFI 启动模式下启动亚马逊 EC2 实例的要求”。

AL2023 启用 UEFI 安全启动的实例仅接受由签名的内核级别代码，包括 Linux 内核和模块 Amazon 因此，您可以确保您的实例仅运行由签名的内核级代码 AWS。

有关亚马逊 EC2 实例和 UEFI 安全启动的更多信息，请参阅亚马逊用户指南 EC2 中的适用于亚马逊亚马逊实例的 UEFI 安全启动。 EC2

先决条件

- 您必须使用版本为 AL2 023 2023.1 或更高版本的 AMI。
- 实例类型必须支持 UEFI 安全启动。有关更多信息，请参阅《亚马逊 EC2 用户指南》中的 “在 UEFI 启动模式下启动亚马逊 EC2 实例的要求”。

在 023 上启用 UEFI 安全启动 AL2

标准 AL2 023 AMIs 包含引导加载程序和由我们的密钥签名的内核。您可以通过注册现有实例来启用 UEFI 安全启动，也可以通过注册快照中的映像在预启用 UEFI 安全启动的情况下进行创建 AMIs 。默认情况下，标准 AL2 023 不启用 UEFI 安全启动。 AMIs

如果实例类型支持 UEFI`uefi-preferred`， AMIs 则启动模式设置为 AL2 023，可确保使用这些启动的实例 AMIs 将使用 UEFI 固件。如果实例类型不支持 UEFI，则使用传统 BIOS 固件启动实例。在传统 BIOS 模式下启动实例时，不会强制执行 UEFI 安全启动。

有关亚马逊 EC2 实例上的 AMI 启动模式的更多信息，请参阅亚马逊 EC2 用户指南中的亚马逊 EC2 启动模式下的实例启动行为。

主题

- 注册现有实例
- 注册快照映像
- 撤消更新
- UEFI Secure Boot 在 023 上是如何工作的 AL2
- 注册您自己的密钥

注册现有实例

要注册一个现有实例，请使用一组密钥填充特定的 UEFI 固件变量，这组密钥使固件能够验证引导加载程序，而引导加载程序又能在下次启动时验证内核。

1. Amazon Linux 提供一个简化注册流程的工具。请运行以下命令以使用一组必要的密钥和证书预置实例。

```
sudo amazon-linux-sb enroll
```

2. 运行以下命令来重新启动实例。实例重启后，将启用 UEFI 安全启动。

```
sudo reboot
```

> **Note**
>
> 亚马逊 Linux AMIs 目前不支持 Nitro 可信平台模块 (NitroTPM)。如果您除了需要 UEFI 安全启动还需要 NitroTPM ，请参阅下一节的信息。

注册快照映像

使用亚马逊 EC2 `register-image` API 从亚马逊 EBS 根卷的快照注册 AMI 时，您可以使用包含 UEFI 变量存储状态的二进制 blob 来配置 AMI。通过提供 AL2 023`UefiData`，即可启用 UEFI 安全启动，无需执行上一节中的步骤。

有关创建和使用二进制 blob 的更多信息，请参阅 A ma EC2 zon 用户指南中的创建包含预先填充的变量存储的二进制 blob。

AL2023 提供了一个预先构建的二进制 blob，可以直接在 Amazon 实例上使用。 EC2 该二进制 blob 位于正在运行的实例上的 `/usr/share/amazon-linux-sb-keys/uefi.vars` 中。此 blob 由 `amazon-linux-sb-keys` RPM 软件包提供，从 2023.1 版本 AMIs 开始，该软件包默认安装在 AL2 023 上。

> **Note**
>
> 为确保您使用的是最新版本的密钥和撤销，请使用与创建 AMI 相同的版本 AL2 023 中的 blob。

当您注册映像时，建议您使用设置为 `uefi` 的 `RegisterImage` API 参数 `BootMode`。这使得您可以通过将 `TpmSupport` 参数设置为 `v2.0` 来启用 NitroTPM。此外，`BootMode` 设置为 `uefi` 可以确保启用 UEFI 安全启动，并且在切换到不支持 UEFI 的实例类型时无法意外禁用该模式。

有关 NitrotPM 的更多信息，请参阅亚马逊用户指南中的适用于亚马逊 EC2 亚马逊实例的 NitrotPM。EC2

撤消更新

Amazon Linux 可能需要发布使用更新的密钥签名的新版引导加载程序 `grub2` 或 Linux 内核。在这种情况下，就需要撤消旧密钥，以防止以前版本的引导加载程序中的可利用漏洞绕过 UEFI 安全启动验证流程。

`grub2` 或 `kernel` 软件包的更新始终会自动将撤消列表更新到正在运行的实例的 UEFI 变量存储中。这意味着，在 UEFI 安全启动启用的情况下，您在为软件包安装安全更新后，将无法再运行该软件包的旧版本。

UEFI Secure Boot 在 023 上是如何工作的 AL2

与其他 Linux 发行版不同，Amazon Linux 不提供一个名为 shim 的额外组件来充当第一阶段引导加载程序。shim 通常使用 Microsoft 密钥签名。例如，在提供 shim 的 Linux 发行版上，shim 会加载 `grub2` 引导加载程序，该引导加载程序使用 shim 自己的代码来验证 Linux 内核。此外，shim 在机器所有者密钥 (MOK) 数据库中维护自己的一组密钥和撤消，该数据库位于 UEFI 变量存储中并使用 `mokutil` 工具进行控制。

Amazon Linux 不提供 shim。因为 AMI 所有者控制 UEFI 变量，所以不需要此中间步骤，而此中间步骤会对启动和引导时间产生不利影响。此外，我们默认选择不包括对任何供应商密钥的信任，这减少了执行不想要的二进制文件的机会。当然，如果客户选择信任，则可以包括想要的二进制文件。

使用 Amazon Linux，UEFI 可以直接加载和验证我们的 `grub2` 引导加载程序。`grub2` 引导加载程序已修改为在加载 Linux 内核后，使用 UEFI 对其进行验证。因此，使用存储在通常 UEFI db 变量（授权密钥数据库）中的相同证书验证 Linux 内核，并根据与引导加载程序和其他 UEFI 二进制文件相同的 dbx 变量（撤消数据库）测试 Linux 内核。因为我们提供自己的 PK 和 KEK 密钥来控制对 db 数据库和 dbx 数据库的访问，所以我们可以根据需要分发签名的更新和撤消，而无需像 shim 这样的中介。

有关 UEFI 安全启动的更多信息，请参阅亚马逊用户指南 EC2 中的 UEFI 安全启动如何与亚马逊实例配合使用。 EC2

注册您自己的密钥

如上一节所述，亚马逊 Linux 不要求在亚马逊上使用 UEFI 安全启动。`shim` EC2当你阅读其他 Linux 发行版的文档时，你可能会找到使用管理机器所有者密钥 (MOK) 数据库的文档`mokutil`， AL2023 上不存在该文档。`shim`和 MOK 环境绕过了 UEFI 固件中密钥注册的一些限制，这些限制不适用于亚马逊 EC2 实施 UEFI 安全启动的方式。借助 Amazon， EC2 有一些机制可以轻松地直接操作 UEFI 变量存储中的密钥。

如果要注册自己的密钥，则可以通过操作现有实例中的变量存储（请参阅从实例内部向变量存储中添加密钥）或通过构造预填充的二进制 blob（请参阅创建包含预填变量存储的二进制 blob）来实现。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。