



用户指南

Amazon ECR



API 版本 2015-09-21

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon ECR: 用户指南

Copyright © 2024 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 ECR	1
Amazon ECR 的组件	1
Amazon ECR 的功能	2
如何开始使用 Amazon ECR	2
Amazon ECR 定价	3
在图像的生命周期中移动图像	4
先决条件	4
安装 AWS CLI	4
安装 Docker	4
步骤 1：创建 Docker 镜像	5
步骤 2：向您的默认注册表验证身份	7
步骤 3：创建存储库	8
步骤 4：推送镜像到 Amazon ECR	8
步骤 5：从 Amazon ECR 提取镜像	10
步骤 6：删除镜像	10
步骤 7：删除存储库	11
优化性能	12
私有注册表	14
注册表概念	14
注册表身份验证	14
使用 Amazon ECR 凭证辅助程序	15
使用授权令牌	15
使用 HTTP API 身份验证	16
注册表设置	17
注册表权限	17
注册表策略示例	18
授予跨账户复制权限	20
授予读取缓存的权限	22
私有存储库	23
存储库概念	23
创建存储库来存储图像	24
后续步骤	25
查看存储库详细信息	25
删除存储库	26

存储库策略	26
存储库策略与 IAM policy	27
存储库策略示例	28
设置存储库策略声明	33
标记存储库	34
标签基本知识	34
标记资源以便于计费	35
添加标签	35
删除标签	37
私有映像	38
推送镜像	38
所需的 IAM 权限	39
推送 Docker 映像	40
推送多架构镜像	41
推送 Helm Chart	43
签署镜像	45
注意事项	45
先决条件	45
为 Notary 客户端配置身份验证	46
签署镜像	46
后续步骤	47
删除签名	47
查看镜像详细信息	48
提取镜像	48
正在提取亚马逊 Linux 容器镜像	50
删除镜像	51
重新为镜像添加标签	52
防止图像标签被覆盖	55
设置图像标签的可变性 (AWS Management Console)	55
设置图像标签的可变性 (AWS CLI)	55
容器镜像清单格式	56
Amazon ECR 镜像清单转换	57
将 Amazon ECR 映像与 Amazon ECS 结合使用	58
所需的 IAM 权限	58
在任务定义中指定 Amazon ECR 镜像	59
将 Amazon ECR 映像与 Amazon EKS 结合使用	60

所需的 IAM 权限	60
在亚马逊 EKS 集群上安装 Helm 图表	61
扫描图像中是否存在漏洞	63
存储库筛选器	63
筛选通配符	64
增强扫描	64
增强扫描的注意事项	65
所需的 IAM 权限	66
配置增强扫描	66
更改增强扫描持续时间	68
EventBridge 事件	69
检索调查结果	74
基本扫描	75
区域支持以改进基本扫描	76
操作系统支持基本扫描和改进的基本扫描	77
配置改进的基本扫描	78
配置基本扫描	78
手动扫描镜像	79
检索调查结果	80
图像扫描疑难解答	82
了解扫描状态 SCAN_ELIGIBILITY_EXPIRED	82
同步上游注册表	83
存储库创建模板	83
使用直通缓存规则的注意事项	83
所需的 IAM 权限	85
使用注册表权限	85
后续步骤	87
创建缓存提取规则	87
先决条件	87
使用 AWS Management Console	88
使用 AWS CLI	92
后续步骤	95
存储库创建模板	95
工作方式	95
所需的 IAM 权限	98
创建存储库创建模板	98

删除存储库创建模板	100
验证直通缓存规则	100
使用缓存提取规则来提取映像	101
存储上游存储库凭证	103
排查缓存提取问题	109
复制镜像	111
私有镜像复制的注意事项	111
复制示例	112
示例：配置跨区域复制到单个目标区域	112
示例：使用存储库筛选条件配置跨区域复制	113
示例：配置跨区域复制到多个目标区域	113
示例：配置跨账户复制	114
示例：在配置中指定多个规则	114
配置复制	115
自动清理图像	117
生命周期策略工作原理	117
生命周期策略评估规则	117
创建生命周期策略预览	118
创建生命周期策略	120
先决条件	120
生命周期策略的示例	121
生命周期策略模板	122
根据镜像使用期限筛选	122
根据镜像计数筛选	123
根据多个规则筛选	123
在单个规则中筛选多个标签	126
筛选所有镜像	128
生命周期策略属性	131
规则优先级	131
描述	131
标签状态	131
标签模式列表	132
标签前缀列表	132
计数类型	133
计数单位	133
计数	133

操作	134
安全性	135
Identity and Access Management	135
受众	136
使用身份进行身份验证	136
使用策略管理访问	139
Amazon Elastic Container Registry 如何与 IAM 结合使用	140
基于身份的策略示例	145
使用基于标签的访问控制	149
AWS 亚马逊 ECR 的托管策略	150
使用服务相关角色	157
故障排除	162
数据保护	164
静态加密	165
合规性验证	171
基础设施安全性	172
接口 VPC 端点 (AWS PrivateLink)	172
防止跨服务混淆座席	180
监控	182
可视化 Service Quotas 并设置警报	183
用量指标	184
用量报告	185
存储库指标	185
启用 CloudWatch 指标	185
可用指标和维度	186
使用查看指标 CloudWatch	186
活动和 EventBridge	187
来自 Amazon ECR 的示例事件	187
使用 记录 AWS CloudTrail操作	191
亚马逊 ECR 信息位于 CloudTrail	191
了解 Amazon ECR 日志文件条目	192
使用 AWS 软件开发工具包	203
代码示例	204
操作	204
DescribeRepositories	204
ListImages	206

服务限额	209
在 AWS Management Console 中管理您的 Amazon ECR 服务配额	213
创建 CloudWatch 警报以监控 API 用量指标	214
故障排除	215
Docker 故障排除	215
Docker 日志不包含预期的错误消息	215
从 Amazon ECR 存储库提取镜像时，出现错误：“Filesystem Verification Failed”(文件系统验证失败) 或“404: Image Not Found”(404：找不到镜像)	215
从 Amazon ECR 提取镜像时，出现错误：“Filesystem Layer Verification Failed”(文件系统分层验证失败)	216
推送到存储库时出现 HTTP 403 错误或“no basic auth credentials”(没有基础级验证凭证) 错误	217
排查 Amazon ECR 错误消息问题	217
HTTP 429：请求太多或 ThrottleException	218
HTTP 403：“User [arn] is not authorized to perform [operation]”(用户 [arn] 没有执行 [operation] 的权限)	218
HTTP 404：“Repository Does Not Exist”(存储库不存在) 错误	218
错误：无法从非 TTY 设备执行交互式登录	219
文档历史记录	220
.....	CCXXIV

什么是 Amazon Elastic Container Registry ?

Amazon Elastic Container Registry (Amazon ECR) AWS 是一项安全、可扩展且可靠的托管容器镜像注册服务。Amazon ECR 使用 AWS IAM 支持具有基于资源的权限的私有存储库。这样，指定用户或 Amazon EC2 实例可以访问您的容器存储库和映像。您可以使用首选 CLI 推送、提取和管理 Docker 映像、Open Container Initiative (OCI) 映像和 OCI 兼容构件。

Note

Amazon ECR 也支持公共容器映像存储库。有关更多信息，请参阅 Amazon ECR Public 用户指南中的 [什么是 Amazon ECR Public](#)。

AWS 容器服务团队维护着一个公开的路线图 GitHub。它包含有关团队正在做的事情的信息，并允许所有 AWS 客户直接提供反馈。有关更多信息，请参阅 [AWS 容器路线图](#)。

Amazon ECR 的组件

Amazon ECR 包括以下组件：

注册表

为每个 AWS 账户提供 Amazon ECR 私有注册表；您可以在注册表中创建一个或多个存储库，并在其中存储 Docker 镜像、开放容器计划 (OCI) 镜像和兼容 OCI 的项目。有关更多信息，请参阅 [Amazon ECR 私有注册表](#)。

授权令牌

您的客户端必须作为 AWS 用户对 Amazon ECR 私有注册表进行身份验证，然后才能推送和提取映像。有关更多信息，请参阅 [Amazon ECR 中的私有注册表身份验证](#)。

存储库

Amazon ECR 存储库包含您的 Docker 镜像、Open Container Initiative (OCI) 镜像和 OCI 兼容构件。有关更多信息，请参阅 [Amazon ECR 私有存储库](#)。

存储库策略

您可以通过存储库策略来控制对存储库及其中的内容的访问。有关更多信息，请参阅 [Amazon ECR 中的私有存储库策略](#)。

图像

您可以对存储库推送和提取容器映像。这些映像可以在开发系统中本地使用，也可以在 Amazon ECS 任务定义和 Amazon EKS Pod 规范中使用。有关更多信息，请参阅 [将 Amazon ECR 映像与 Amazon ECS 结合使用](#) 和 [将 Amazon ECR 映像与 Amazon EKS 结合使用](#)。

Amazon ECR 的功能

Amazon ECR 提供以下功能：

- 生命周期策略有助于管理存储库中映像的生命周期。您可以定义导致清理未使用映像的规则。您可以在将规则应用到存储库之前对其进行测试。有关更多信息，请参阅 [在 Amazon ECR 中使用生命周期策略自动清理图像](#)。
- 映像扫描有助于识别容器映像中的软件漏洞。每个存储库都可以配置为在推送时扫描。这可确保扫描推送到存储库的每个新映像。然后，您可以检索映像扫描的结果。有关更多信息，请参阅 [扫描图像以查找 Amazon ECR 中的软件漏洞](#)。
- 跨区域和跨账户复制使您可以更轻松地将映像放置在需要的位置。它配置为注册表设置，并基于每个区域。有关更多信息，请参阅 [Amazon ECR 中的私有注册表设置](#)。
- 缓存提取规则提供了在私有 Amazon ECR 注册表中缓存上游注册表中的存储库的方法。使用缓存提取规则时，Amazon ECR 将定期访问上游注册表，以确保 Amazon ECR 私有注册表中缓存的映像为最新状态。有关更多信息，请参阅 [将上游注册表与 Amazon ECR 私有注册表同步](#)。

如何开始使用 Amazon ECR

如果您使用的是亚马逊弹性容器服务 (Amazon ECS) 或亚马逊 Elastic Kubernetes Service (Amazon EKS)，请注意这两项服务的设置与亚马逊 ECR 的设置类似，因为亚马逊 ECR 是这两项服务的扩展。

AWS Command Line Interface 与 Amazon ECR 一起使用时，请使用支持最新 Amazon ECR 功能的版本。AWS CLI 如果您在中看不到对 Amazon ECR 功能的支持 AWS CLI，请升级到最新版本的 AWS CLI。有关安装最新版本的信息 AWS CLI，请参阅《AWS Command Line Interface 用户指南》AWS CLI 中的 [安装或更新到最新版本的](#)。

要了解如何使用 AWS CLI 和 Docker 将容器映像推送到私有 Amazon ECR 存储库，请参阅 [在 Amazon ECR 中移动图像的整个生命周期](#)

Amazon ECR 定价

使用 Amazon ECR，您只需为存储库中存储的数据量以及从映像推送和提取所传输的数据付费。有关更多信息，请参阅 [Amazon ECR 定价](#)。

在 Amazon ECR 中移动图像的整个生命周期

如果您是首次使用 Amazon ECR，请在 Docker CLI 和中使用以下步骤 AWS CLI 来创建示例映像、向默认注册表进行身份验证并创建私有存储库。然后将图像推送到私有存储库并从中提取图像。完成示例图像后，删除示例图像和存储库。

要使用代 AWS Management Console 替 AWS CLI，请参阅[the section called “创建存储库来存储图像”](#)。

[有关可用于管理 AWS 资源的其他工具（包括不同的 AWS 软件开发工具包、IDE 工具包和 Windows PowerShell 命令行工具）的更多信息，请参阅 <http://aws.amazon.com/tools/>。](#)

先决条件

如果您尚未安装最新版本 AWS CLI 和 Docker，也未准备就绪，请按照以下步骤安装这两个工具。

安装 AWS CLI

要将 Amazon ECR AWS CLI 与 Amazon ECR 一起使用，请安装最新 AWS CLI 版本。有关信息，请参阅《AWS Command Line Interface 用户指南》中的[安装 AWS Command Line Interface](#)。

安装 Docker

Docker 适用于许多不同的操作系统，包括大多数现代 Linux 分发版 (如 Ubuntu) 甚至 MacOS 和 Windows。有关如何在特定的操作系统上安装 Docker 的更多信息，请转到 [Docker 安装指南](#)。

您无需本地开发系统即可使用 Docker。如果您已在使用 Amazon EC2，则可启动 Amazon Linux 2023 实例并安装 Docker 以开始使用。

如果您已安装 Docker，请跳到[步骤 1：创建 Docker 镜像](#)。

使用 Amazon Linux 2023 AMI 在 Amazon EC2 实例上安装 Docker

1. 使用最新版 Amazon Linux 2023 AMI 启动实例。有关更多信息，请参阅 Amazon EC2 用户指南中的[启动实例](#)。
2. 连接到您的实例。有关更多信息，请参阅 Amazon EC2 用户指南中的“[连接到您的 Linux 实例](#)”。
3. 更新实例上已安装的程序包和程序包缓存。

```
sudo yum update -y
```

4. 安装最新的 Docker Community Edition 程序包。

```
sudo yum install docker
```

5. 启动 Docker 服务。

```
sudo service docker start
```

6. 将 `ec2-user` 添加到 `docker` 组，以便您能够执行 Docker 命令，而无需使用 `sudo`。

```
sudo usermod -a -G docker ec2-user
```

7. 退出，再重新登录以接受新的 `docker` 组权限。您可以关闭当前的 SSH 终端窗口并在新终端窗口中重新连接到实例，完成这一过程。您的新 SSH 会话将具有相应的 `docker` 组权限。
8. 验证 `ec2-user` 是否能在没有 `sudo` 的情况下运行 Docker 命令。

```
docker info
```

Note

在某些情况下，您可能需要重新启动实例，以便为 `ec2-user` 提供访问 Docker 进程守护程序的权限。如果您看到以下错误，请尝试重启您的实例：

```
Cannot connect to the Docker daemon. Is the docker daemon running on this host?
```

步骤 1：创建 Docker 镜像

在本步骤中，您将创建简单 Web 应用程序的 Docker 映像，并在本地系统或 Amazon EC2 实例上测试此映像。

创建简单 Web 应用程序的 Docker 镜像

1. 创建名为 `Dockerfile` 的文件。`Dockerfile` 是一个清单文件，描述了用于 Docker 镜像的基本镜像以及要安装的项目以及在此项目上运行的内容。有关 `Dockerfile` 的更多信息，请转到 [Dockerfile 参考](#)。

touch Dockerfile

2. 编辑您刚刚创建的 Dockerfile 并添加以下内容。

```
FROM public.ecr.aws/amazonlinux/amazonlinux:latest

# Install dependencies
RUN yum update -y && \
    yum install -y httpd

# Install apache and write hello world message
RUN echo 'Hello World!' > /var/www/html/index.html

# Configure apache
RUN echo 'mkdir -p /var/run/httpd' >> /root/run_apache.sh && \
    echo 'mkdir -p /var/lock/httpd' >> /root/run_apache.sh && \
    echo '/usr/sbin/httpd -D FOREGROUND' >> /root/run_apache.sh && \
    chmod 755 /root/run_apache.sh

EXPOSE 80

CMD /root/run_apache.sh
```

此 Dockerfile 使用 Amazon ECR Public 上托管的 Amazon Linux 2 公有映像。RUN 指令更新包缓存，安装一些适用于 Web 服务器的软件包，然后将“Hello World!”内容写入 Web 服务器的文档根目录。EXPOSE 指令在容器上公开端口 80，CMD 指令启动 Web 服务器。

3. 从您的 Dockerfile 生成 Docker 镜像。

Note

Docker 的某些版本可能需要在以下命令中使用 Dockerfile 完整路径，而不是所示的相对路径。

```
docker build -t hello-world .
```

4. 列出容器映像。

```
docker images --filter reference=hello-world
```

输出：

REPOSITORY SIZE	TAG	IMAGE ID	CREATED
hello-world 194MB	latest	e9ffedc8c286	4 minutes ago

5. 运行新构建的镜像。-p 80:80 选项将容器上公开的端口 80 映射到主机系统上的端口 80。有关 docker run 的更多信息，请转到 [Docker 运行参考](#)。

```
docker run -t -i -p 80:80 hello-world
```

Note

来自 Apache Web 服务器的输出将显示在终端窗口中。您可以忽略“Could not reliably determine the fully qualified domain name”消息。

6. 打开浏览器并指向正在运行 Docker 并托管您的容器的服务器。
 - 如果您使用的是 EC2 实例，这将是服务器的 Public DNS 值，此值与您用于通过 SSH 连接到实例的地址相同。确保实例的安全组允许端口 80 上的入站流量。
 - 如果您正在本地运行 Docker，可将您的浏览器指向 <http://localhost/>。
 - 如果你在 Windows 或 Mac 计算机 docker-machine 上使用，请使用 docker-machine ip 命令查找托管 Docker 的 VirtualBox 虚拟机的 IP 地址，用你正在使用的 docker 计算机的名称替换 **#名**。

```
docker-machine ip machine-name
```

您应看到一个显示“Hello World!”语句的网页。

7. 通过键入 Ctrl + c 来停止 Docker 容器。

步骤 2：向您的默认注册表验证身份

安装并配置完毕后 AWS CLI，请使用默认注册表对 Docker CLI 进行身份验证。这样一来，docker 命令可以通过 Amazon ECR 推送和提取镜像。AWS CLI 提供了用于简化身份验证过程的 get-login-password 命令。

要使用向 Amazon ECR 注册表对 Docker 进行身份验证 `get-login-password`，请运行命令。`aws ecr get-login-password` 将身份验证令牌传递给 `docker login` 命令时，将值 `AWS` 用作用户名，并指定要对其进行身份验证的 Amazon ECR 注册表 URI。如果对多个注册表进行身份验证，则必须针对每个注册表重复该命令。

Important

如果收到错误，请安装或更新到最新版本的 AWS CLI。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](#)。

- [get-login-password](#) (AWS CLI)

```
aws ecr get-login-password --region region | docker login --username AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

- [get-ecr \(\) LoginCommand](#) AWS Tools for Windows PowerShell

```
(Get-ECRLoginCommand).Password | docker login --username AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

步骤 3：创建存储库

现在您已拥有可推送到 Amazon ECR 的镜像，还必须创建一个存储库来保存它。在本示例中，您创建一个名称为 `hello-repository` 的存储库，稍后将推送 `hello-world:latest` 镜像到这里。要创建存储库，请运行以下命令：

```
aws ecr create-repository \  
  --repository-name hello-repository \  
  --region region
```

步骤 4：推送镜像到 Amazon ECR

现在您可以推送镜像到上一部分中创建的 Amazon ECR 存储库。满足以下先决条件后，使用 `docker CLI` 推送镜像：

- 安装的最低版本 `docker` 为：1.7。

- Amazon ECR 授权令牌配置为 `docker login`
- Amazon ECR 存储库存在且用户有向该存储库推送的权限。

在满足这些先决条件后，即可将镜像推送到您在帐户的默认注册表中新创建的存储库中。

标记镜像并推送到 Amazon ECR

1. 列出您存储在本地的镜像，以识别要标记和推送的镜像。

```
docker images
```

输出：

REPOSITORY	TAG	IMAGE ID	CREATED
VIRTUAL SIZE			
hello-world	latest	e9ffedc8c286	4 minutes ago
241MB			

2. 标记镜像并推送到存储库。

```
docker tag hello-world:latest aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository
```

3. 推送镜像。

```
docker push aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository
```

输出：

```
The push refers to a repository [aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository] (len: 1)
e9ae3c220b23: Pushed
a6785352b25c: Pushed
0998bf8fb9e9: Pushed
0a85502c06c9: Pushed
latest: digest: sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636EXAMPLE
size: 6774
```

步骤 5：从 Amazon ECR 提取镜像

将您的图像推送到您的 Amazon ECR 存储库后，您可以从其他位置提取它。满足以下先决条件后，使用 docker CLI 拉取镜像：

- 安装的最低版本docker为：1.7。
- Amazon ECR 授权令牌配置为。docker login
- Amazon ECR 存储库存在且用户有从该存储库提取的权限。

在满足这些先决条件后，即可提取您的镜像。要从 Amazon ECR 提取示例镜像，请运行以下命令：

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository:latest
```

输出：

```
latest: Pulling from hello-repository
0a85502c06c9: Pull complete
0998bf8fb9e9: Pull complete
a6785352b25c: Pull complete
e9ae3c220b23: Pull complete
Digest: sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636EXAMPLE
Status: Downloaded newer image for aws_account_id.dkr.region.amazonaws.com/hello-
repository:latest
```

步骤 6：删除镜像

如果您不再需要某个存储库中的图像，则可以删除该图像。要删除图像，请指定该图像所在的存储库以及该图像的imageTag或imageDigest值。以下示例删除hello-repository存储库中带有 image 标签的图像latest。要从存储库中删除您的示例图像，请运行以下命令：

```
aws ecr batch-delete-image \  
  --repository-name hello-repository \  
  --image-ids imageTag=latest \  
  --region region
```

步骤 7：删除存储库

如果您不再需要整个图像存储库，则可以删除该存储库。以下示例使用该`--force`标志删除包含图像的存储库。要删除包含映像的存储库 (及其中的所有映像)，请运行以下命令：

```
aws ecr delete-repository \  
  --repository-name hello-repository \  
  --force \  
  --region region
```

优化 Amazon ECR 的性能

使用 Amazon ECR 时，您可以使用以下有关设置和策略的建议来优化性能。

使用 Docker 1.10 及以上版本可利用同时层上传

Docker 镜像由层组成，是镜像的中间构建阶段。Dockerfile 的每一行都会创建新层。当使用 Docker 1.10 及以上版本时，Docker 在默认情况下会在上传至 Amazon ECR 的同时推送尽可能多的层，从而缩短上传时间。

使用较小基本镜像

通过 Docker Hub 提供的默认镜像，可能包含您的应用程序不需要的很多依赖项。请考虑使用其他人在 Docker 社区创建并维护的较小镜像，或使用 Docker 最小 Scratch 镜像构建您自己的基本镜像。有关更多信息，请参阅 Docker 文档中的[创建基本镜像](#)。

更早将更改最少的依赖性放入您的 Dockerfile

Docker 缓存层，可加速构建时间。如果从最后一次构建至今，某一层上没有任何更改，则 Docker 将使用缓存版本，而不重新构建层。但是，每层都依赖之前出现的层。如果层发生更改，则 Docker 不仅重新编译该层，也会重新编译该层之后出现的所有层。

为了尽量缩短重新构建 Dockerfile 并重新上传层所需的时间，可考虑早些时候将更改频率最低的依赖项放入 Dockerfile。将经常更改的依赖项 (如应用程序的源代码) 稍后放入堆栈。

链接命令以避免不必要文件的存储

在层中创建的中间文件会作为该层的一部分保留，即使该层在后续层中被删除。考虑以下示例：

```
WORKDIR /tmp
RUN wget http://example.com/software.tar.gz
RUN wget tar -xvf software.tar.gz
RUN mv software/binary /opt/bin/myapp
RUN rm software.tar.gz
```

在本示例中，第一个和第二个 RUN 命令创建的层包含原始 .tar.gz 文件及其所有解压内容。即使第四个 RUN 命令已删除 .tar.gz 文件。这些命令可以链接在一起，构成单独的运行语句，以确保最终 Docker 镜像中不包含不必要的文件。

```
WORKDIR /tmp
RUN wget http://example.com/software.tar.gz &&\
    wget tar -xvf software.tar.gz &&\
```

```
mv software/binary /opt/bin/myapp &&\nrm software.tar.gz
```

使用最近的区域终端节点

通过确保使用最靠近所运行应用程序的区域终端节点，可以减少从 Amazon ECR 提取镜像的延迟。如果应用程序在 Amazon EC2 实例上运行，可以使用以下 shell 代码从实例的可用区获取区域：

```
REGION=$(curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone\n|\nsed -n 's/^(\d*\)[a-zA-Z]*$/\1/p')
```

可以使用 `--region` 参数将该区域传递给 AWS CLI 命令，也可以使用命令将该区域设置为配置文件的默认区域。aws configure 您还可以在使用 AWS SDK 拨打电话时设置区域。有关更多信息，请参阅适用于特定编程语言的软件开发工具包文档。

Amazon ECR 私有注册表

Amazon ECR 私有注册表通过高度可用且可扩展的架构托管容器镜像。您可以使用私有注册表管理由 Docker 以及 Open Container Initiative(OCI) 镜像和构件组成的私有镜像存储库。每个 AWS 账户都提供有原定设置的私有 Amazon ECR 注册表。有关 Amazon ECR 公有注册表的更多信息，请参阅 Amazon Elastic Container Registry Public 用户指南中的[公有注册表](#)。

私有注册表概念

- 您的原定设置私有注册表的 URL 为 `https://aws_account_id.dkr.ecr.us-west-2.amazonaws.com`。
- 预设情况下，您的账户可以读取和写入私有注册表中的存储库。但是，用户需要权限才能调用 Amazon ECR API，以及向您的私有存储库推送或拉取映像。Amazon ECR 提供了多个托管策略来控制不同级别的用户访问。有关更多信息，请参阅 [Amazon Elastic Container Registry 基于身份的策略示例](#)。
- 您必须为 Docker 客户端授予注册表权限，以便使用 `docker push` 和 `docker pull` 命令对该私有注册表中的存储库执行推送和提取镜像操作。有关更多信息，请参阅 [Amazon ECR 中的私有注册表身份验证](#)。
- 可通过 用户访问策略及存储库策略对私有存储库加以控制。有关存储库策略的更多信息，请参阅 [Amazon ECR 中的私有存储库策略](#)。
- 通过配置私有注册表的复制，私有注册表中的存储库可以在自己的私有注册表中的区域之间进行复制，也可以跨单独的账户进行复制。有关更多信息，请参阅 [在 Amazon ECR 中复制私有镜像](#)。

Amazon ECR 中的私有注册表身份验证

您可以使用 AWS Management Console AWS CLI、或 AWS SDK 来创建和管理私有仓库。也可以使用这些方法对镜像执行某些操作，例如列出或删除镜像。这些客户端使用标准的 AWS 身份验证方法。尽管在技术上可以使用 Amazon ECR API 推送和提取镜像，但您更有可能使用 Docker CLI 或特定语言的 Dockerf 库。

Docker CLI 不支持本机 IAM 身份验证方法。必须执行其他步骤，以便 Amazon ECR 可以对 Docker 推送和提取请求进行身份验证和授权。

我们提供以下各节详细介绍的注册表身份验证方法。

使用 Amazon ECR 凭证辅助程序

Amazon ECR 提供了 Docker 凭证辅助程序，这使得在 Amazon ECR 中推送和提取镜像时更容易存储和使用 Docker 凭证。有关安装和配置步骤，请参阅 [Amazon ECR Docker 凭证辅助程序](#)。

Note

目前，Amazon ECR Docker 凭证助手不支持多重身份验证 (MFA)。

使用授权令牌

授权令牌的权限范围与用于检索身份验证令牌的 IAM 委托人的权限范围相匹配。身份验证令牌用于访问您的 IAM 委托人有权访问且有效期为 12 小时的任何 Amazon ECR 注册表。要获取授权令牌，必须使用 [GetAuthorizationToken](#) API 操作检索包含用户名 AWS 和编码密码的 base64 编码的授权令牌。该 AWS CLI `get-login-password` 命令通过检索和解码授权令牌来简化此操作，然后您可以将其传送到 `docker login` 命令中进行身份验证。

使用 `get-login` 针对 Amazon ECR 私有注册表验证 Docker

- 要使用向 Amazon ECR 注册表对 Docker 进行身份验证 `get-login-password`，请运行命令。 `aws ecr get-login-password` 将身份验证令牌传递给 `docker login` 命令时，将值 AWS 用作用户名，并指定要对其进行身份验证的 Amazon ECR 注册表 URI。如果对多个注册表进行身份验证，则必须针对每个注册表重复该命令。

Important

如果收到错误，请安装或更新到最新版本的 AWS CLI。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](#)。

- [get-login-password](#) (AWS CLI)

```
aws ecr get-login-password --region region | docker login --username AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

- [get-ecr \(\) LoginCommand](#) AWS Tools for Windows PowerShell

```
(Get-ECRLoginCommand).Password | docker login --username AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

使用 HTTP API 身份验证

Amazon ECR 支持 [Docker 注册表 HTTP API](#)。但是，由于 Amazon ECR 属于私有注册表，因此您必须为每个 HTTP 请求提供授权令牌。您可以使用 -H 选项添加 HTTP 授权标头，curl 然后传递 get-authorization-token AWS CLI 命令提供的授权令牌。

使用 Amazon ECR HTTP API 进行身份验证

1. 使用检索授权令牌 AWS CLI 并将其设置为环境变量。

```
TOKEN=$(aws ecr get-authorization-token --output text --query 'authorizationData[].authorizationToken')
```

2. 要向 API 进行身份验证，可将 \$TOKEN 变量传递到 curl 命令的 -H 选项。例如，以下命令会列出 Amazon ECR 存储库中的镜像标签。有关更多信息，请参阅 [Docker 注册表 HTTP API 参考文档](#)。

```
curl -i -H "Authorization: Basic $TOKEN" https://aws_account_id.dkr.ecr.region.amazonaws.com/v2/amazonlinux/tags/list
```

您可以在一个 (扩展) 代码行中执行所有这些操作：

```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8
Date: Thu, 04 Jan 2018 16:06:59 GMT
Docker-Distribution-Api-Version: registry/2.0
Content-Length: 50
Connection: keep-alive

{"name":"amazonlinux","tags":["2017.09","latest"]}
```


Amazon ECR 中的私有注册表设置

Amazon ECR 使用私有注册表设置在注册表级别配置功能。为每个区域分别配置私有注册表设置。您可以使用私有注册表设置来配置以下功能。

- 注册表权限 – 注册表权限策略允许控制复制和缓存提取权限。有关更多信息，请参阅 [Amazon ECR 中的私有注册表权限](#)。
- 缓存提取规则 – 缓存提取规则用于在 Amazon ECR 私有注册表中缓存来自上游注册表的映像。有关更多信息，请参阅 [将上游注册表与 Amazon ECR 私有注册表同步](#)。
- 复制配置 – 复制配置用于控制存储库是跨 AWS 区域复制，还是跨账户复制。有关更多信息，请参阅 [在 Amazon ECR 中复制私有镜像](#)。
- 存储库创建模板 – 存储库创建模板用于定义 Amazon ECR 代表您创建新存储库时要应用的标准设置。例如，通过缓存提取操作创建的存储库。有关更多信息，请参阅 [用于控制在缓存拉取操作期间创建的存储库的模板](#)。
- 扫描配置—默认情况下，为您的注册表启用基本扫描。您可以启用增强扫描功能，该功能提供一种自动持续的扫描模式，可扫描操作系统和编程语言包漏洞。有关更多信息，请参阅 [扫描图像以查找 Amazon ECR 中的软件漏洞](#)。

Amazon ECR 中的私有注册表权限

Amazon ECR 使用注册表策略在私有注册表级别向 AWS 主体授予权限。这些权限用于确定对复制和拉取缓存功能的访问权限范围。

Amazon ECR 仅在私有注册表级别强制执行以下权限。如果向注册表策略添加了任何其他操作，则将发生错误。

- `ecr:ReplicateImage` – 向其他账户（称为源注册表）授予将其镜像复制到您的注册表的权限。这仅用于跨账户复制。
- `ecr:BatchImportUpstreamImage` – 授权检索外部镜像并将其导入到您的私有注册表。
- `ecr:CreateRepository` – 授予在私有注册表中创建存储库的权限。如果该私有注册表中尚未存在用于存储复制镜像或缓存镜像的存储库，则需要此权限。

Note

虽然可以将 `ecr:*` 操作添加到私有注册表权限策略，但最佳实践是仅根据您使用的功能添加所需的特定操作，而不是使用通配符。

主题

- [Amazon ECR 的私有注册政策示例](#)
- [授予在 Amazon ECR 中进行跨账户复制的注册权限](#)
- [授予 Amazon ECR 中读取缓存的注册表权限](#)

Amazon ECR 的私有注册政策示例

以下示例显示了您可用于控制用户对 Amazon ECR 注册表所具备权限的注册表权限策略声明。

Note

在每个示例中，如果从您的注册表权限声明中删除 `ecr:CreateRepository` 操作，复制仍然可能发生。但是，要成功复制，您需要在账户中创建具有相同名称的存储库。

示例：允许源账户的根用户复制所有存储库

以下注册表权限策略允许源账户的 `root` 用户复制所有存储库。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReplicationAccessCrossAccount",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::source_account_id:root"
      },
      "Action": [
        "ecr:CreateRepository",
        "ecr:ReplicateImage"
      ]
    }
  ]
}
```

```

        "Resource": [
            "arn:aws:ecr:us-west-2:your_account_id:repository/*"
        ]
    }
]
}

```

示例：允许来自多个账户的 root 用户

以下注册表权限策略有两个声明。每条语句都允许源账户的 root 用户复制所有存储库。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReplicationAccessCrossAccount",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::source_account_id:root"
      },
      "Action": [
        "ecr:CreateRepository",
        "ecr:ReplicateImage"
      ],
      "Resource": [
        "arn:aws:ecr:us-west-2:your_account_id:repository/*"
      ]
    },
    {
      "Sid": "ReplicationAccessCrossAccount",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::source_account_id:root"
      },
      "Action": [
        "ecr:CreateRepository",
        "ecr:ReplicateImage"
      ],
      "Resource": [
        "arn:aws:ecr:us-west-2:your_account_id:repository/*"
      ]
    }
  ]
}

```

```
}
```

示例：允许源账户的根用户复制带有前缀 **prod-** 的所有存储库。

以下注册表权限策略允许源账户的 root 用户复制所有以开头的存储库prod-。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReplicationAccessCrossAccount",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::source_account_id:root"
      },
      "Action": [
        "ecr:CreateRepository",
        "ecr:ReplicateImage"
      ],
      "Resource": [
        "arn:aws:ecr:us-west-2:your_account_id:repository/prod-*"
      ]
    }
  ]
}
```

授予在 Amazon ECR 中进行跨账户复制的注册权限

跨账户策略类型被用于向 AWS 主体授予权限，允许将存储库从源注册表复制到您的注册表。预设情况下，您有权在自己的注册表中配置跨区域复制。如果您授予其他账户将内容复制到注册表的权限，则只需配置注册表策略。

注册表策略必须授予 `ecr:ReplicateImage` API 操作权限。此 API 是一个内部 Amazon ECR API，可在区域或账户之间复制镜像。您还可以授予 `ecr:CreateRepository` 权限，该权限允许 Amazon ECR 在您的注册表中创建存储库（如果存储库尚不存在）。如果未提供 `ecr:CreateRepository` 权限，则必须在注册表中手动创建与源存储库名称相同的存储库。如果两者均未完成，复制将失败。任何失败的操作 `CreateRepository` 或 `ReplicateImage` API 操作都会显示在中 CloudTrail。

要配置复制的权限策略 (AWS Management Console)

1. 打开位于 <https://console.aws.amazon.com/ecr/> 的 Amazon ECR 控制台。

2. 从导航栏中，选择区域以配置注册表策略。
3. 在导航窗格中，选择 Private registry (私有注册表)、Registry permissions (注册表权限)。
4. 在 Registry permissions (注册表权限) 页面上，选择 Generate statement (生成语句)。
5. 使用策略生成器完成以下步骤以定义策略声明。
 - a. 对于 Policy type (策略类型)，选择 Cross account policy (跨账户策略)。
 - b. 对于声明 ID，输入唯一的声明 ID。此字段在注册表策略上用作 Sid。
 - c. 对于账户，输入要向其授予权限的每个账户的账户 ID。当指定多个账户 ID 时，请将它们以逗号分隔。
6. 展开预览策略声明部分以查看注册表权限策略声明。
7. 确认策略声明后，选择添加到策略以将策略保存到您的注册表。

要配置复制的权限策略 (AWS CLI)

1. 创建名为 registry_policy.json 的文件并使用注册表策略填充它。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReplicationAccessCrossAccount",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::source_account_id:root"
      },
      "Action": [
        "ecr:CreateRepository",
        "ecr:ReplicateImage"
      ],
      "Resource": [
        "arn:aws:ecr:us-west-2:your_account_id:repository/*"
      ]
    }
  ]
}
```

2. 使用策略文件创建注册表策略。

```
aws ecr put-registry-policy \
```

```
--policy-text file://registry_policy.json \  
--region us-west-2
```

3. 检索要确认的注册表策略。

```
aws ecr get-registry-policy \  
--region us-west-2
```

授予 Amazon ECR 中读取缓存的注册表权限

Amazon ECR 私有注册表权限可用于限定各个 IAM 实体使用推送缓存的权限范围。如果 IAM policy 授予 IAM 实体的权限多于注册表权限策略授予的权限，则 IAM policy 优先。

要创建私有注册表的权限策略 (AWS Management Console)

1. 打开 Amazon ECR 控制台：<https://console.aws.amazon.com/ecr/>。
2. 从导航栏中，选择您在其中配置私有注册表权限语句的区域。
3. 在导航窗格中，选择 Private registry (私有注册表)、Registry permissions (注册表权限)。
4. 在 Registry permissions (注册表权限) 页面上，选择 Generate statement (生成语句)。
5. 对于要创建的每个缓存提取权限策略语句，请执行以下操作。
 - a. 对于 Policy type (策略类型)，请选择 Pull through cache policy (推送缓存策略)。
 - b. 对于 Statement id (语句 ID)，为推送缓存语句策略提供名称。
 - c. 对于 IAM entities (IAM 实体)，指定要包含在策略中的用户、组或角色。
 - d. 对于 Repository namespace (存储库命名空间)，选择要与策略关联的推送缓存规则。
 - e. 对于 Repository names (存储库名称)，指定要应用规则的存储库基本名称。例如，如果您想在 Amazon ECR Public 上指定 Amazon Linux 存储库，存储库名称将为 amazonlinux。

Amazon ECR 私有存储库

Amazon ECR 私有存储库包含您的 Docker 镜像、开放容器计划 (OCI) 镜像和兼容 OCI 的工件。您可以使用 Amazon ECR API 操作或 Amazon ECR 控制台的“存储库”部分创建、监控和删除镜像存储库，并设置权限来控制谁可以访问这些存储库。Amazon ECR 还与 Docker CLI 集成，因此您可以将映像从开发环境推送和拉取到存储库。

主题

- [私有存储库概念](#)
- [创建 Amazon ECR 私有存储库来存储图像](#)
- [在 Amazon ECR 中查看私有存储库的内容和详细信息](#)
- [删除 Amazon ECR 中的私有存储库](#)
- [Amazon ECR 中的私有存储库政策](#)
- [在 Amazon ECR 中标记私有存储库](#)

私有存储库概念

- 默认情况下，您的账户可以读取和写入默认注册表中的存储库 (`aws_account_id.dkr.ecr.region.amazonaws.com`)。但是，用户需拥有权限才能调用 Amazon ECR API 并从您的存储库中推送或提取镜像。Amazon ECR 提供了多个托管策略来控制不同级别的用户访问。有关更多信息，请参阅 [Amazon Elastic Container Registry 基于身份的策略示例](#)。
- 通过用户访问策略及个别存储库策略均可控制存储库。有关更多信息，请参阅 [Amazon ECR 中的私有存储库政策](#)。
- 存储库名称可支持命名空间，您可以使用命名空间分组相似的存储库。例如，如果多个团队使用相同的注册表，团队 A 可以使用 team-a 命名空间，团队 B 可以使用 team-b 命名空间。这样，每个团队都有自己的名为 web-app 的镜像，每个镜像都以团队命名空间开头。这种配置可在不产生干扰的情况下同时使用每个团队上的这些镜像。团队 A 的镜像是 team-a/web-app，团队 B 的镜像是 team-b/web-app。
- 您的镜像可以在自己的注册表中跨区域和跨账户复制到其他存储库。您可以通过在注册表设置中指定复制配置来执行此操作。有关更多信息，请参阅 [Amazon ECR 中的私有注册表设置](#)。

创建 Amazon ECR 私有存储库来存储图像

创建 Amazon ECR 私有存储库，然后使用该存储库存储您的容器映像。请按照以下步骤以使用 AWS Management Console 创建私有存储库。有关使用创建存储库的步骤 AWS CLI，请参阅 [步骤 3：创建存储库](#)。

创建存储库 (AWS Management Console)

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择您创建存储库的区域。
3. 在“存储库”页面上，选择“私有仓库”，然后选择“创建存储库”。
4. 对于 Visibility settings (可见性设置)，请验证已选择 Private (私有)。
5. 对于存储库名称，输入存储库的唯一名称。存储库名称可以自行指定 (例如 nginx-web-app)。或者，可以在其前面加上命名空间来将存储库分组到类别中 (例如 project-a/nginx-web-app)。

Note

存储库名称最多可以包含 256 个字符。名称必须以字母开头，并且只能包含小写字母、数字、连字符、下划线和正斜杠。不支持使用双连字符、双下划线或双正斜杠。

6. 对于标签不变性，选择存储库的标签可变性设置。配置有不可变标签的存储库会阻止覆盖镜像标签。有关更多信息，请参阅 [防止在 Amazon ECR 中覆盖图片标签](#)。
7. 对于 Scan on push (推送时扫描)，尽管您可以在存储库级别为基础扫描指定扫描设置，但最佳实践是在私有注册表级别指定扫描配置。通过在私有注册表级别指定扫描设置，您可以启用增强扫描或基本扫描，并定义用于指定扫描哪些存储库的筛选条件。有关更多信息，请参阅 [扫描图像以查找 Amazon ECR 中的软件漏洞](#)。
8. 对于 KMS 加密，请选择是否使用对存储库中的图像启用加密 AWS Key Management Service。默认情况下，启用 KMS 加密后，Amazon ECR 会使用带有别名 aws/ecr 的 AWS 托管式密钥 (KMS 密钥)。当您首次创建启用 KMS 加密的存储库时，将在您的账户中创建此密钥。有关更多信息，请参阅 [静态加密](#)。
9. 当启用 KMS 加密时，选择客户加密设置 (高级) 以选择自己的 KMS 密钥。该 KMS 密钥必须与集群同在一个区域中。选择“创建 AWS KMS 密钥”，导航到 AWS KMS 控制台以创建自己的密钥。
10. 选择创建存储库。

后续步骤

要查看将图像推送到存储库的步骤，请选择存储库并选择“查看推送命令”。要详细了解如何将镜像推送到存储库，请参阅[将图像推送到 Amazon ECR 私有存储库](#)。

在 Amazon ECR 中查看私有存储库的内容和详细信息

创建私有存储库后，您可以在以下位置查看有关该存储库的详细信息 AWS Management Console：

- 存储库中存储了哪些镜像
- 与存储库中存储的每个镜像相关的详细信息，包括每个镜像的大小和 SHA 摘要
- 为存储库内容指定的扫描频率
- 存储库是否具有与之关联的活动拉取缓存规则
- 存储库的加密设置

Note

从 Docker 版本 1.9 开始，在将镜像推送至 V2 版本的 Docker 注册表之前，Docker 客户端会压缩镜像的分层。docker images 命令的输出显示未压缩的镜像大小。因此，请记住，Docker 可能会返回比 AWS Management Console 中所示更大的镜像。

查看存储库信息 (AWS Management Console)

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择包含要查看的存储库的区域。
3. 在导航窗格中，选择存储库。
4. 在 Repositories (存储库) 页面上，选择 Private (私有) 选项卡，然后选择要查看的存储库。
5. 在存储库详细信息页面上，控制台默认为 Images (镜像) 视图。使用导航菜单查看其他存储库相关信息。
 - 选择 Summary (摘要) 查看存储库详细信息并拉取该存储库的计数数据。
 - 选择 Images (镜像) 以查看与存储库中的镜像相关的信息。要查看有关镜像的更多信息，请选择镜像标签。有关更多信息，请参阅 [在 Amazon ECR 中查看图片详情](#)。

如果您要删除的镜像没有标签，您可以选择要删除的存储库左侧的框，然后选择删除。有关更多信息，请参阅 [在 Amazon ECR 中删除图片](#)。

- 选择权限以查看适用于存储库的存储库策略。有关更多信息，请参阅 [Amazon ECR 中的私有存储库策略](#)。
- 选择生命周期策略以查看适用于存储库的生命周期策略规则。此处还可查看生命周期事件历史记录。有关更多信息，请参阅 [在 Amazon ECR 中使用生命周期策略自动清理图像](#)。
- 选择标签以查看适用于存储库的元数据标签。

删除 Amazon ECR 中的私有存储库

如果存储库已使用完毕，您可以删除它。在中删除存储库时 AWS Management Console，存储库中包含的所有图像也会被删除；此操作无法撤消。

Important

已删除存储库中的图像也会被删除。您无法撤消此操作。

删除存储库 (AWS Management Console)

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择包含要删除的存储库的区域。
3. 在导航窗格中，选择存储库。
4. 在 Repositories (存储库) 页面上，选择 Private (私有) 选项卡，然后选择要删除的存储库，并选择 Delete (删除)。
5. 在删除 *repository_name* 窗口中，验证是否应删除所选存储库，然后选择删除。

Amazon ECR 中的私有存储库政策

Amazon ECR 使用基于资源的权限控制对存储库的访问。基于资源的权限允许您指定哪些用户或角色有权访问存储库，以及他们可以对存储库执行哪些操作。默认情况下，只有创建存储库的 AWS 账户才有权访问存储库。您可以应用允许对仓库进行额外访问的存储库策略。

主题

- [存储库策略与 IAM policy](#)
- [Amazon ECR 中的私有存储库策略示例](#)

- [在 Amazon ECR 中设置私有仓库政策声明](#)

存储库策略与 IAM policy

Amazon ECR 存储库策略是 IAM policy 的一部分，这些策略专用于控制对单个 Amazon ECR 存储库的访问。IAM policy 通常用于应用针对整个 Amazon ECR 服务的权限，但也可用于控制对特定资源的访问。

在确定某个特定用户或角色可对存储库执行的操作时，将同时使用 Amazon ECR 存储库策略和 IAM policy。如果通过存储库策略允许某个用户或角色执行某个操作但通过 IAM policy 拒绝其执行该操作（或反过来），则将拒绝该操作。用户或角色只需通过存储库策略或 IAM policy 之一获得执行某个操作的许可，而不需要同时通过这两个策略来获得执行该操作的许可。

Important

Amazon ECR 要求用户有权通过 IAM policy 调用 `ecr:GetAuthorizationToken` API，然后才能对注册表进行身份验证并对任意 Amazon ECR 存储库推送或提取任意镜像。Amazon ECR 提供多个托管 IAM policy，用于控制不同级别下的用户访问；有关更多信息，请参阅 [Amazon Elastic Container Registry 基于身份的策略示例](#)。

您可以使用这两个策略类型之一来控制对您的存储库的访问，如以下示例中所示。

此示例显示了一个 Amazon ECR 存储库策略，该策略允许某个特定用户描述存储库及存储库内的镜像。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ECRRepositoryPolicy",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::account-id:user/username"},
      "Action": [
        "ecr:DescribeImages",
        "ecr:DescribeRepositories"
      ]
    }
  ]
}
```

此示例显示了一个 IAM policy，该策略可实现与上面相同的目标，方法是使用资源参数将策略范围限定为存储库 (由存储库的完整 ARN 指定)。有关 Amazon Resource Name (ARN) 格式的更多信息，请参阅 [资源](#)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowDescribeRepoImage",
      "Effect": "Allow",
      "Action": [
        "ecr:DescribeImages",
        "ecr:DescribeRepositories"
      ],
      "Resource": ["arn:aws:ecr:region:account-id:repository/repository-name"]
    }
  ]
}
```

Amazon ECR 中的私有存储库策略示例

Important

此页面上的存储库策略示例旨在应用于 Amazon ECR 私有存储库。如果直接与 IAM 主体结合使用，这些存储库策略将无法正常工作，除非进行修改，将 Amazon ECR 存储库指定为资源。有关如何设置存储库策略的更多信息，请参阅 [在 Amazon ECR 中设置私有仓库政策声明](#)。

Amazon ECR 存储库策略是 IAM policy 的一部分，这些策略专用于控制对单个 Amazon ECR 存储库的访问。IAM policy 通常用于应用针对整个 Amazon ECR 服务的权限，但也可用于控制对特定资源的访问。有关更多信息，请参阅 [存储库策略与 IAM policy](#)。

以下存储库策略示例显示了可用于控制对 Amazon ECR 私有存储库的访问权限的权限声明。

Important

Amazon ECR 要求用户有权通过 IAM policy 调用 `ecr:GetAuthorizationToken` API，然后才能对注册表进行身份验证并从任意 Amazon ECR 存储库推送或提取任意镜像。Amazon

ECR 提供多个托管 IAM policy，用于控制不同级别下的用户访问；有关更多信息，请参阅 [Amazon Elastic Container Registry 基于身份的策略示例](#)。

示例：允许一个或多个用户

以下存储库策略允许一个或多个用户向存储库推送和从存储库提取映像。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPushPull",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::account-id:user/push-pull-user-1",
          "arn:aws:iam::account-id:user/push-pull-user-2"
        ]
      },
      "Action": [
        "ecr:BatchGetImage",
        "ecr:BatchCheckLayerAvailability",
        "ecr:CompleteLayerUpload",
        "ecr:GetDownloadUrlForLayer",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
      ]
    }
  ]
}
```

示例：允许其他账户

以下存储库策略允许特定账户推送镜像。

Important

您授予权限的账户必须启用所创建存储库策略的区域，否则将发生错误。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCrossAccountPush",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-id:root"
      },
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:CompleteLayerUpload",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
      ]
    }
  ]
}
```

以下存储库策略允许部分用户拉取镜像 (*pull-user-1* 和 *pull-user-2*)，并为其他用户提供完全的访问权限 (*admin-user*)。

Note

对于目前不支持的更复杂的存储库策略 AWS Management Console，您可以使用 [set-repository-policy](#) AWS CLI 命令应用该策略。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPull",
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::account-id:user/pull-user-1",
          "arn:aws:iam::account-id:user/pull-user-2"
        ]
      },
    }
  ],
}
```

```

    "Action": [
      "ecr:BatchGetImage",
      "ecr:GetDownloadUrlForLayer"
    ]
  },
  {
    "Sid": "AllowAll",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::account-id:user/admin-user"
    },
    "Action": [
      "ecr:*"
    ]
  }
]
}

```

示例：拒绝所有

以下存储库策略拒绝所有账户中的所有用户提取镜像。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyPull",
      "Effect": "Deny",
      "Principal": "*",
      "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer"
      ]
    }
  ]
}

```

示例：限制对特定 IP 地址的访问权限

以下示例拒绝向任何用户授予在应用于来自特定地址范围的存储库时执行任何 Amazon ECR 操作的权限。

此语句中的条件确定允许的 Internet 协议版本 4 (IPv4) IP 地址范围为 54.240.143.*。

该Condition块使用NotIpAddress条件和aws:SourceIp条件键，后者是一个 AWS宽范围的条件键。有关这些条件键的更多信息，请参阅 [AWS 全局条件上下文键](#)。aws:sourceIp IPv4 值使用标准CIDR 表示法。有关更多信息，请参阅《IAM 用户指南》中的 [IP 地址条件运算符](#)。

```
{
  "Version": "2012-10-17",
  "Id": "ECRPolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "ecr:*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": "54.240.143.0/24"
        }
      }
    }
  ]
}
```

示例：允许某项 AWS 服务

以下存储库策略允许 AWS CodeBuild 访问与该服务集成所需的 Amazon ECR API 操作。使用以下示例时，您应该使用 aws:SourceArn 和 aws:SourceAccount 条件键来限定哪些资源可以使用这些权限。有关更多信息，请参阅AWS CodeBuild 用户指南 CodeBuild中的 [Amazon ECR 示例](#)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccess",
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer"
      ],
      "Condition": {
```



```
        "ArnLike":{
            "aws:SourceArn":"arn:aws:codebuild:region:123456789012:project/project-
name"
        },
        "StringEquals":{
            "aws:SourceAccount":"123456789012"
        }
    }
}
]
```

在 Amazon ECR 中设置私有仓库政策声明

您可以按照以下步骤向中的存储库添加访问策略声明。AWS Management Console 您可以为每个存储库添加多个策略声明。有关示例策略，请参阅 [Amazon ECR 中的私有存储库策略示例](#)。

Important

Amazon ECR 要求用户有权通过 IAM policy 调用 `ecr:GetAuthorizationToken` API，然后才能对注册表进行身份验证并从任意 Amazon ECR 存储库推送或提取任意镜像。Amazon ECR 提供多个托管 IAM 策略，用于控制不同级别下的用户访问；有关更多信息，请参阅 [Amazon Elastic Container Registry 基于身份的策略示例](#)。

设置存储库策略声明

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择包含要对其设置策略声明的存储库的区域。
3. 在导航窗格中，选择存储库。
4. 在存储库页面上，选择要对其设置策略声明的存储库，以查看存储库的内容。
5. 从存储库镜像列表视图的导航窗格中，选择权限、编辑。

Note

如果您未看到权限选项，请确保您处于存储库镜像列表视图中。

6. 在编辑权限页面上，选择添加声明。

7. 对于声明名称，输入声明的名称。
8. 对于效果，选择策略语句产生的结果是允许还是明确拒绝。
9. 对于委托人，选择策略声明应用到的范围。有关更多信息，请参阅 IAM 用户指南中的 [AWS JSON 策略元素：委托人](#)。
 - 通过选中 Everyone (*) 复选框，可以将该语句应用于所有经过身份验证的 AWS 用户。
 - 对于服务委托人，指定服务委托人名称 (例如 `ecs.amazonaws.com`) 以将声明应用到特定的服务。
 - 对于 AWS 账户 ID，请指定一个 AWS 账号 (例如 `111122223333`)，以将该账单应用于特定 AWS 账户下的所有用户。可以使用逗号分隔的列表指定多个账户。

Important

您授予权限的账户必须启用所创建存储库策略的区域，否则将发生错误。

- 对于 IAM 实体，请选择您的 AWS 账户下要应用声明的角色或用户。

Note

对于目前不支持的更复杂的存储库策略 AWS Management Console，您可以使用 [set-repository-policy](#) AWS CLI 命令应用该策略。

10. 对于操作，从各个 API 操作的列表中选择应当应用策略声明的 Amazon ECR API 操作的范围。
11. 完成后，选择保存设置策略。
12. 对要添加的每个存储库策略重复以上步骤。

在 Amazon ECR 中标记私有存储库

为了帮助您管理 Amazon ECR 存储库，您可以使用 AWS 资源标签将自己的元数据分配给新的或现有的 Amazon ECR 存储库。例如，您可以为账户的 Amazon ECR 存储库定义一组标签以帮助跟踪每个存储库的拥有者。

标签基本知识

标签对 Amazon ECR 没有任何语义意义，应严格按字符串进行解析。标签不会自动分配至资源。您可以修改标签的密钥和值，还可以随时删除资源的标签。您可以将标签的值设为空的字符串，但是不能将

其设为空值。如果您添加的标签的值与该实例上现有标签的值相同，新的值就会覆盖旧值。如果删除资源，资源的所有标签也会被删除。

您可以使用 Amazon ECR 控制台、AWS CLI、和 Amazon ECR API 来处理标签。

使用 AWS Identity and Access Management (IAM)，您可以控制 AWS 账户中哪些用户有权创建、编辑或删除标签。有关 IAM 策略中标签的信息，请参阅[the section called “使用基于标签的访问控制”](#)。

标记资源以便于计费

您为 Amazon ECR 存储库添加的标签在成本和使用率报告中启用标签后查看成本分配时非常有帮助。有关更多信息，请参阅[Amazon ECR 用量报告](#)。

如需查看组合资源的成本，请按具有相同标签键值的资源组织您的账单信息。例如，您可以将特定的应用程序名称用作几个资源的标签，然后组织账单信息，以查看在数个服务中的使用该应用程序的总成本。有关设置带有标签的成本分配报告的更多信息，请参阅 AWS Billing 用户指南中的[月度成本分配报告](#)。

Note

如果您已启用报告，则可以在 24 小时后查看当月的数据。

向 Amazon ECR 中的私有存储库添加标签

您可以向私有仓库添加标签。

有关标签名称和最佳做法的信息，请参阅[《标签 AWS 资源用户指南》中的标签命名限制和要求以及最佳实践](#)。

向存储库添加标签 (AWS Management Console)

1. 打开位于 <https://console.aws.amazon.com/ecr/> 的 Amazon ECR 控制台。
2. 从导航栏中，选择要使用的区域。
3. 在导航窗格中，选择存储库。
4. 在存储库页面上，选中要标记的存储库旁边的复选框。
5. 从操作菜单中，选择存储库标签。
6. 在存储库标签页面上，选择添加标签、添加标签。

7. 在编辑存储库标签页面上，为每个标签指定键和值，然后选择保存。

向存储库 (AWS CLI 或 API) 添加标签

您可以使用或 API 添加或覆盖一个 AWS CLI 或多个标签。

- AWS CLI - [标签资源](#)
- API 操作- [TagResource](#)

以下示例说明如何使用添加标签 AWS CLI。

示例 1：为存储库添加标签

以下命令为存储库添加标签。

```
aws ecr tag-resource \  
  --resource-arn arn:aws:ecr:region:account_id:repository/repository_name \  
  --tags Key=stack,Value=dev
```

示例 2：使用多个标签标记存储库

以下命令向存储库添加三个标签。

```
aws ecr tag-resource \  
  --resource-arn arn:aws:ecr:region:account_id:repository/repository_name \  
  --tags Key=key1,Value=value1 Key=key2,Value=value2 Key=key3,Value=value3
```

示例 3：列出存储库的标签

以下命令列出了与存储库关联的标签。

```
aws ecr list-tags-for-resource \  
  --resource-arn arn:aws:ecr:region:account_id:repository/repository_name
```

示例 4：创建仓库并添加标签

以下命令创建一个名为 test-repo 的存储库并添加键为 team、值为 devs 的标签。

```
aws ecr create-repository \  
  --repository-name test-repo \  
  --tags Key=team,Value=devs
```

```
--tags Key=team,Value=devs
```

从 Amazon ECR 的私有存储库中删除标签

您可以从私有仓库中删除标签。

从私有存储库中删除标签 (AWS Management Console)

1. 打开位于 <https://console.aws.amazon.com/ecr/> 的 Amazon ECR 控制台。
2. 从导航栏中，选择要使用的区域。
3. 在存储库页面上，选中要从中删除标签的存储库旁边的复选框。
4. 从操作菜单中，选择存储库标签。
5. 在存储库标签页面上，选择编辑。
6. 在编辑存储库标签页面上，选择要删除的每个标签对应的删除，然后选择保存。

从私有存储库中删除标签 (AWS CLI)

您可以使用或 API 删除一个 AWS CLI 或多个标签。

- AWS CLI -[取消标记资源](#)
- API 操作-[UntagResource](#)

以下示例说明如何使用从存储库中删除标签 AWS CLI。

```
aws ecr untag-resource \  
  --resource-arn arn:aws:ecr:region:account_id:repository/repository_name \  
  --tag-keys tag_key
```

亚马逊 ECR 中的私有图片

Amazon ECR 将 Docker 镜像、开放容器计划 (OCI) 镜像和兼容 OCI 的工件存储在私有存储库中。您可以使用 Docker CLI 或首选客户端从存储库推送和提取镜像。

主题

- [将图像推送到 Amazon ECR 私有存储库](#)
- [对存储在 Amazon ECR 私有存储库中的图像进行签名](#)
- [从 Amazon ECR 私有存储库中删除签名](#)
- [在 Amazon ECR 中查看图片详情](#)
- [从 Amazon ECR 私有存储库将映像拉取到您的本地环境](#)
- [正在提取亚马逊 Linux 容器镜像](#)
- [在 Amazon ECR 中删除图片](#)
- [在 Amazon ECR 中重新标记图片](#)
- [防止在 Amazon ECR 中覆盖图片标签](#)
- [Amazon ECR 中支持容器镜像清单格式](#)
- [将 Amazon ECR 映像与 Amazon ECS 结合使用](#)
- [将 Amazon ECR 映像与 Amazon EKS 结合使用](#)

将图像推送到 Amazon ECR 私有存储库

您可以将 Docker 镜像、清单列表和 Open Container Initiative (OCI) 镜像以及兼容的构件推送到您的私有存储库。

Amazon ECR 还提供了一种将您的映像复制到其他存储库的方法。通过在私有注册表设置中指定复制配置，您可以在自己的注册表中跨区域进行复制，也可以跨不同的账户进行复制。有关更多信息，请参阅 [Amazon ECR 中的私有注册表设置](#)。

主题

- [将图像推送到 Amazon ECR 私有存储库的 IAM 权限](#)
- [将 Docker 镜像推送到亚马逊 ECR 私有存储库](#)
- [将多架构映像推送到 Amazon ECR 私有存储库](#)
- [将 Helm 图表推送到 Amazon ECR 私有存储库](#)

将图像推送到 Amazon ECR 私有存储库的 IAM 权限

用户需要 IAM 权限才能将图像推送到 Amazon ECR 私有存储库。按照授予最低权限的最佳实践，您可以授予对特定存储库的访问权限。您也可以授予对所有存储库的访问权限。

用户必须通过请求授权令牌，向其推送镜像的每个 Amazon ECR 注册表进行身份验证。Amazon ECR 提供了多种 AWS 托管策略来控制不同级别的用户访问权限。有关更多信息，请参阅 [AWS Amazon 弹性容器注册表的托管策略](#)。

您也可以创建自己的 IAM 策略。以下 IAM 策略授予将图像推送到特定存储库所需的权限。必须将存储库指定为完整的 Amazon Resource Name (ARN)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:CompleteLayerUpload",
        "ecr:UploadLayerPart",
        "ecr:InitiateLayerUpload",
        "ecr:BatchCheckLayerAvailability",
        "ecr:PutImage"
      ],
      "Resource": "arn:aws:ecr:region:111122223333:repository/repository-name"
    },
    {
      "Effect": "Allow",
      "Action": "ecr:GetAuthorizationToken",
      "Resource": "*"
    }
  ]
}
```

以下 IAM 策略授予将镜像推送到所有存储库所需的权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "ecr:CompleteLayerUpload",
        "ecr:GetAuthorizationToken",
        "ecr:UploadLayerPart",
        "ecr:InitiateLayerUpload",
        "ecr:BatchCheckLayerAvailability",
        "ecr:PutImage"
    ],
    "Resource": "*"
}
]
```

将 Docker 镜像推送到亚马逊 ECR 私有存储库

您可以使用 `docker push` 命令将容器镜像推送到 Amazon ECR 存储库。

Amazon ECR 还支持创建和推送用于多架构映像的 Docker 清单列表。有关信息，请参阅 [将多架构映像推送到 Amazon ECR 私有存储库](#)。

推送 Docker 镜像到 Amazon ECR 存储库

在推送镜像之前，Amazon ECR 存储库必须存在。有关更多信息，请参阅 [the section called “创建存储库来存储图像”](#)。

1. 向要向其推送镜像的 Amazon ECR 注册表验证 Docker 客户端的身份。必须针对每个注册表获得授权令牌，令牌有效期为 12 小时。有关更多信息，请参阅 [Amazon ECR 中的私有注册表身份验证](#)。

要对 Amazon ECR 注册表验证 Docker，请运行 `aws ecr get-login-password` 命令。将身份验证令牌传递给 `docker login` 命令时，将值 `AWS` 用作用户名，并指定要对其进行身份验证的 Amazon ECR 注册表 URI。如果对多个注册表进行身份验证，则必须针对每个注册表重复该命令。

Important

如果收到错误，请安装或更新到最新版本的 AWS CLI。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 get-login-password --region region | docker login --username AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```


2. 如果在要推送的注册表中还没有您的镜像存储库，请创建它。有关更多信息，请参阅 [创建 Amazon ECR 私有存储库来存储图像](#)。
3. 识别要推送的本地镜像。运行 `docker images` 命令列出系统中的容器镜像。

```
docker images
```

在生成的命令输出中，可以通过 `repository:tag` 值或镜像 ID 识别镜像。

4. 通过要使用的 Amazon ECR 注册表、存储库和可选镜像标签名称组合标记您的镜像。注册表格式为 `aws_account_id.dkr.ecr.us-west-2.amazonaws.com`。存储库名称应与您为镜像创建的存储库一致。如果省略镜像标签，我们将假定标签为 `latest`。

以下示例使用 ID `e9ae3c220b23` 作为 `aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository:tag` 来标记本地镜像。

```
docker tag e9ae3c220b23 aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository:tag
```

5. 使用 `docker push` 命令推送镜像：

```
docker push aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository:tag
```

6. (可选) 通过重复 [Step 4](#) 和 [Step 5](#)，向镜像应用任何其他标签并将这些标签推送到 Amazon ECR。

将多架构映像推送到 Amazon ECR 私有存储库

您可以通过创建和推送 Docker 清单列表将多架构映像推送到 Amazon ECR 存储库。清单列表是通过指定一个或多个镜像名称创建的镜像列表。在大多数情况下，清单列表是根据具有相同功能但适用于不同操作系统或架构的图像创建的。清单列表不是必需项。有关更多信息，请参阅 [Docker 清单](#)。

清单列表可以像其他 Amazon ECR 镜像一样在 Amazon ECS 任务定义或 Amazon EKS Pod 规范中提取或引用。

先决条件

- 在你的 Docker CLI 中，开启实验性功能。有关实验性功能的信息，请参阅 Docker 文档中的 [实验性功能](#)。
- 在推送镜像之前，Amazon ECR 存储库必须存在。有关更多信息，请参阅 [the section called “创建存储库来存储图像”](#)。

- 在创建 Docker 清单之前，必须将镜像推送到您的存储库。有关如何推送镜像的信息，请参阅 [将 Docker 镜像推送到亚马逊 ECR 私有存储库](#)。

将多架构 Docker 镜像推送到 Amazon ECR 存储库

1. 向要向其推送镜像的 Amazon ECR 注册表验证 Docker 客户端的身份。必须针对每个注册表获得授权令牌，令牌有效期为 12 小时。有关更多信息，请参阅 [Amazon ECR 中的私有注册表身份验证](#)。

要对 Amazon ECR 注册表验证 Docker，请运行 `aws ecr get-login-password` 命令。将身份验证令牌传递给 `docker login` 命令时，将值 `AWS` 用作用户名，并指定要对其进行身份验证的 Amazon ECR 注册表 URI。如果对多个注册表进行身份验证，则必须针对每个注册表重复该命令。

Important

如果收到错误，请安装或更新到最新版本的 AWS CLI。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 get-login-password --region region | docker login --username AWS --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

2. 列出存储库中的镜像，确认镜像标签。

```
aws ecr describe-images --repository-name my-repository
```

3. 创建 Docker 清单列表。manifest create 命令验证引用的镜像是否已存在于您的存储库中，并在本地创建清单。

```
docker manifest create aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository:image_one_tag aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository:image_two
```

4. (可选) 检查 Docker 清单列表。这使您能够确认清单列表中引用的每个镜像清单的大小和摘要。

```
docker manifest inspect aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository
```

5. 将 Docker 清单列表推送到您的 Amazon ECR 存储库。

```
docker manifest push aws_account_id.dkr.ecr.us-west-2.amazonaws.com/my-repository
```

将 Helm 图表推送到 Amazon ECR 私有存储库

您可以将开放容器倡议 (OCI) 项目推送到 Amazon ECR 存储库。要查看此功能的示例，请使用以下步骤将 Helm 图表推送到 Amazon ECR。

有关在 Amazon EKS 上使用 Amazon ECR 托管的 Helm 图表的信息，请参阅[在亚马逊 EKS 集群上安装 Helm 图表](#)。

将 Helm Chart 推送到 Amazon ECR 存储库

1. 安装最新版本的 Helm 客户端。这些步骤是使用 Helm 版本 3.8.2 编写的。有关更多信息，请参阅[安装 Helm](#)。
2. 请按照以下步骤创建测试 Helm Chart。有关更多信息，请参阅[Helm 文档 - 入门](#)。
 - a. 创建名为 `helm-test-chart` 的 Helm Chart 并清除 `templates` 目录的内容。

```
helm create helm-test-chart  
rm -rf helm-test-chart/templates/*
```

- b. 在 `templates` 文件夹 `ConfigMap` 中创建。

```
cd helm-test-chart/templates  
cat <<EOF > configmap.yaml  
apiVersion: v1  
kind: ConfigMap  
metadata:  
  name: helm-test-chart-configmap  
data:  
  myvalue: "Hello World"  
EOF
```

3. 打包图表。输出将包含您在推送 Helm Chart 时使用的打包图表的文件名。

```
cd ../../  
helm package helm-test-chart
```

输出

```
Successfully packaged chart and saved it to: /Users/username/helm-test-chart-0.1.0.tgz
```

4. 创建存储库以存储 Helm Chart。存储库的名称应与步骤 2 中创建 Helm Chart 时使用的名称匹配。有关更多信息，请参阅 [创建 Amazon ECR 私有存储库来存储图像](#)。

```
aws ecr create-repository \  
  --repository-name helm-test-chart \  
  --region us-west-2
```

5. 对要向其推送 Helm Chart 的 Amazon ECR 注册表验证 Docker 客户端的身份。必须针对每个注册表获得授权令牌，令牌有效期为 12 小时。有关更多信息，请参阅 [Amazon ECR 中的私有注册表身份验证](#)。

```
aws ecr get-login-password \  
  --region us-west-2 | helm registry login \  
  --username AWS \  
  --password-stdin aws_account_id.dkr.ecr.us-west-2.amazonaws.com
```

6. 使用 `helm push` 命令推送 Helm Chart。输出应包括 Amazon ECR 存储库 URI 和 SHA 摘要。

```
helm push helm-test-chart-0.1.0.tgz oci://aws_account_id.dkr.ecr.us-west-2.amazonaws.com/
```

7. 描述您的 Helm Chart。

```
aws ecr describe-images \  
  --repository-name helm-test-chart \  
  --region us-west-2
```

在输出中，验证 `artifactMediaType` 参数指示正确的构件类型。

```
{  
  "imageDetails": [  
    {  
      "registryId": "aws_account_id",  
      "repositoryName": "helm-test-chart",
```

```
    "imageDigest":
      "sha256:dd8aebdda7df991a0ffe0b3d6c0cf315fd582cd26f9755a347a52adEXAMPLE",
      "imageTags": [
        "0.1.0"
      ],
      "imageSizeInBytes": 1620,
      "imagePushedAt": "2021-09-23T11:39:30-05:00",
      "imageManifestMediaType": "application/vnd.oci.image.manifest.v1+json",
      "artifactMediaType": "application/vnd.cncf.helm.config.v1+json"
    }
  ]
}
```

8. (可选) 有关其他步骤，请安装 Helm configmap 并开始使用 Amazon EKS。有关更多信息，请参阅 [在亚马逊 EKS 集群上安装 Helm 图表](#)。

对存储在 Amazon ECR 私有存储库中的图像进行签名

Amazon ECR AWS Signer 与集成，为您提供了一种签署容器映像的方法。您可以将容器映像和签名存储在私有存储库中。

注意事项

使用 Amazon ECR 镜像签名时应注意以下事项。

- 存储库中存储的签名计入每个存储库的最大镜像数量的服务限额。有关更多信息，请参阅 [Amazon ECR 服务配额](#)。
- 在使用 Amazon ECR 生命周期策略时，任何使 OCI 镜像索引过期或删除 OCI 镜像索引的规则操作都将导致 Amazon ECR 在 24 小时内删除该镜像索引引用的任何签名。

先决条件

在您开始之前，必须满足以下先决条件。

- 安装并配置最新版本的 AWS CLI。有关更多信息，请参阅《AWS Command Line Interface 用户指南》中的 [安装或更新最新版本的 AWS CLI](#)。
- 安装符号 CLI 和符号 AWS Signer 插件。有关更多信息，请参阅《AWS Signer 开发人员指南》中的 [对容器镜像进行签名的先决条件](#)。

- 签署存储在 Amazon ECR 私有存储库中的容器镜像。有关更多信息，请参阅 [将图像推送到 Amazon ECR 私有存储库](#)。

为 Notary 客户端配置身份验证

在使用 Notafication CLI 创建签名之前，必须先配置客户端，使其能够向 Amazon ECR 进行身份验证。如果您在安装 Notation 客户端的同一主机上安装了 Docker，那么 Notation 将重用您为 Docker 客户端使用的相同身份验证方法。Docker login 和 logout 命令将允许 Notation sign 和 verify 命令使用相同的凭证，并且您不必单独对 Notation 进行身份验证。有关配置 Notary 客户端进行身份验证的详细信息，请参阅 Notary Project 文档中的 [使用符合 OCI 的注册表进行身份验证](#)

如果您没有使用 Docker 或其他使用 Docker 凭证的工具，那么我们建议使用 Amazon ECR Docker 凭证助手作为您的凭证存储。有关如何安装和配置 Amazon ECR 凭证助手的更多信息，请参阅 [Amazon ECR Docker 凭证助手](#)。

签署镜像

可以使用以下步骤创建签署容器镜像所需的资源，并将签名存储到 Amazon ECR 私有存储库中。Notation 使用摘要对镜像进行签名。

要签署映像

1. 使用 AWS Signer 签名平台创建 Notation-OCI-SHA384-ECDSA 签名配置文件。您可以选择性地使用 `--signature-validity-period` 参数指定签名有效期。可以使用 DAYS、MONTHS 或 YEARS 指定此值。如果未指定有效期，则使用默认值 135 个月。

```
aws signer put-signing-profile --profile-name ecr_signing_profile --platform-id  
Notation-OCI-SHA384-ECDSA
```

Note

签名配置文件名称仅支持字母数字字符和下划线 (_)。

2. 根据您的默认注册表进行 Notation 客户端身份验证。以下示例使用对 Amazon ECR 私有注册表的 Notation CLI 进行身份验证。AWS CLI

```
aws ecr get-login-password --region region | notation login --username AWS --  
password-stdin 111122223333.dkr.ecr.region.amazonaws.com
```

3. 使用 Notation CLI 对映像进行签名，并且使用存储库名称和 SHA 摘要指定映像。这将创建签名并将其推送到正在签名的镜像所在的 Amazon ECR 私有存储库。

在以下示例中，我们正在对curl带 SHA 摘要的存储库sha256:ca78e5f730f9a789ef8c63bb55275ac12dfb9e8099e6EXAMPLE中的映像进行签名。

```
notation
sign 111122223333.dkr.ecr.region.amazonaws.com/
curl@sha256:ca78e5f730f9a789ef8c63bb55275ac12dfb9e8099e6EXAMPLE --plugin
"com.amazonaws.signer.notation.plugin" --id "arn:aws:signer:region:111122223333:/
signing-profiles/ecrSigningProfileName"
```

后续步骤

签署容器映像后，可以在本地验证签名。有关验证镜像的说明，请参阅《AWS Signer 开发者指南》中的[“登录后在本地验证镜像”](#)

从 Amazon ECR 私有存储库中删除签名

您可以从 Amazon ECR 私有存储库中删除签名。当您使用 Notation CLI 创建和推送签名时，还会在您的 Amazon ECR 存储库中创建 OCI 镜像索引。Amazon ECR API 不支持删除 OCI 镜像索引所引用的构件或镜像，因此以下是清理这些构件的可用选项。

- (推荐) 您可以使用 ORAS CLI 删除构件，ORAS 将处理镜像索引的更新或删除。
- 您可以使用 Amazon ECR API 或控制台先删除 OCI 镜像索引，然后删除引用的构件，例如签名。

使用 ORAS 客户端删除签名和其他引用类型构件时，ORAS 会管理 OCI 镜像索引。ORAS 将首先从索引中删除对构件的引用，然后删除清单。可以使用 `oras manifest delete` 命令，以引用签名构件的索引。

使用 ORAS CLI 删除签名

1. 安装和配置 ORAS 客户端。

有关安装和配置 ORAS 客户端的信息，请参阅 ORAS 文档中的[安装](#)。

2. 要使用 ORAS CLI 删除签名，请运行以下命令：

```
oras manifest
delete 111122223333.dkr.ecr.region.amazonaws.com/
repository_name@sha256:ca78e5f730f9a789ef8c63bb55275ac12dfb9e8099e6EXAMPLE
```

在 Amazon ECR 中查看图片详情

将图像推送到存储库后，您可以查看有关该图像的信息。所包括的详细信息如下：

- 镜像 URI
- 镜像标签
- 构件媒体类型
- 镜像清单类型
- 扫描状态
- 镜像的大小 (以 MiB 为单位)
- 镜像推送到存储库的时间
- 复制状态

查看镜像详细信息 (AWS Management Console)

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择包含镜像所在存储库的区域。
3. 在导航窗格中，选择存储库。
4. 在存储库页面上，选择要查看的存储库。
5. 在存储库：**repository_name**页面上，选择要查看详细信息的镜像。

从 Amazon ECR 私有存储库将映像拉取到您的本地环境

如果希望运行 Amazon ECR 中可用的 Docker 镜像，可以使用 `docker pull` 命令将其提取到本地环境。您可以从默认注册表或与其他 AWS 账户关联的注册表中执行此操作。

要在 Amazon ECS 任务定义中使用 Amazon ECR 镜像，请参阅 [将 Amazon ECR 映像与 Amazon ECS 结合使用](#)

⚠ Important

Amazon ECR 要求用户有权通过 IAM policy 调用 `ecr:GetAuthorizationToken` API，然后才能对注册表进行身份验证并从任意 Amazon ECR 存储库推送或提取任意镜像。Amazon ECR 提供了多种 AWS 托管策略来控制不同级别的用户访问权限。有关 Amazon ECR AWS 托管策略的信息，请参阅[AWS Amazon 弹性容器注册表的托管策略](#)。

从 Amazon ECR 存储库提取 Docker 镜像

1. 将您的 Docker 客户端验证到要从中提取镜像的 Amazon ECR 注册表。必须针对每个注册表获得授权令牌，令牌有效期为 12 小时。有关更多信息，请参阅[Amazon ECR 中的私有注册表身份验证](#)。
2. (可选) 识别要提取的镜像。
 - 可以使用 `aws ecr describe-repositories` 命令列出注册表中的存储库：

```
aws ecr describe-repositories
```

上述示例注册表包含一个名为 `amazonlinux` 的存储库。

- 可以使用 `aws ecr describe-images` 命令描述存储库中的镜像：

```
aws ecr describe-images --repository-name amazonlinux
```

上述示例存储库具有带标签 `latest` 和 `2016.09` 的镜像，并且镜像摘要为 `sha256:f1d4ae3f7261a72e98c6ebefe9985cf10a0ea5bd762585a43e0700ed99863807`。

3. 使用 `docker pull` 命令提取镜像。镜像名称格式应为 `registry/repository[:tag]` 以便按标签提取，或为 `registry/repository[@digest]` 以便按摘要提取。

```
docker pull aws_account_id.dkr.ecr.us-west-2.amazonaws.com/amazonlinux:latest
```

⚠ Important

如果您收到 `repository-url not found: does not exist or no pull access` 错误，您可能需要向 Amazon ECR 验证您的 Docker 客户端。有关更多信息，请参阅[Amazon ECR 中的私有注册表身份验证](#)。

正在提取亚马逊 Linux 容器镜像

构建 Amazon Linux 容器镜像的软件组件与 Amazon Linux AMI 中包含的软件组件相同。Amazon Linux 容器镜像可在任何环境中用作 Docker 工作负载的基础映像。如果您将 Amazon Linux AMI 用于亚马逊 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。

您可以在本地开发环境中使用 Amazon Linux 容器镜像，然后使用 Amazon ECS 将您的应用程序推送到 AWS 使用 Amazon ECS。有关更多信息，请参阅 [将 Amazon ECR 映像与 Amazon ECS 结合使用](#)。

Amazon Linux 容器镜像在 Amazon ECR Public 中和 [Docker Hub 上](#) 可用。要获得对 Amazon Linux 容器镜像的支持，请访问 [AWS 开发者论坛](#)。

从 Amazon ECR Public 中提取 Amazon Linux 容器镜像

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 ECR 中的私有注册表身份验证](#)。

Note

从版本 1.18.1.187 开始，AWS CLI 中提供有 `ecr-public` 命令，但是，我们建议使用最新版本 AWS CLI。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
```

您可以在一个 (扩展) 代码行中执行所有这些操作：

```
Login succeeded
```

2. 使用 `docker pull` 命令提取 Amazon Linux 容器镜像。要在 Amazon ECR Public Gallery 中查看 Amazon Linux 容器镜像，请参阅 [Amazon ECR Public Gallery - amazonlinux](#)。

```
docker pull public.ecr.aws/amazonlinux/amazonlinux:latest
```

3. (可选) 在本地运行容器。

```
docker run -it public.ecr.aws/amazonlinux/amazonlinux /bin/bash
```

从 Docker Hub 提取 Amazon Linux 容器镜像

1. 使用 `docker pull` 命令提取 Amazon Linux 容器镜像。

```
docker pull amazonlinux
```

2. (可选) 在本地运行容器。

```
docker run -it amazonlinux:latest /bin/bash
```

在 Amazon ECR 中删除图片

如果您结束使用镜像，可以从存储库中删除它。如果您结束使用存储库，可以删除整个存储库以及其中的所有镜像。有关更多信息，请参阅 [删除 Amazon ECR 中的私有存储库](#)。

作为手动删除镜像的替代方法，您可以创建存储库生命周期策略，以便更好地控制存储库中镜像的生命周期管理。生命周期策略自动执行此过程。有关更多信息，请参阅 [在 Amazon ECR 中使用生命周期策略自动清理图像](#)。

删除镜像 (AWS Management Console)

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择包含要删除的镜像的区域。
3. 在导航窗格中，选择存储库。
4. 在存储库页面上，选择包含要删除的镜像的存储库。
5. 在存储库：**repository_name** 页面上，选择要删除的镜像左侧的框，然后选择删除。
6. 在删除镜像对话框中，验证选定的镜像是否应被删除，然后选择删除。

删除镜像 (AWS CLI)

1. 列出存储库中的镜像。带标签的镜像将具有镜像摘要以及相关标签的列表。不带标签的镜像仅具有镜像摘要。

```
aws ecr list-images \  
  --repository-name my-repo
```

2. (可选) 通过指定要删除镜像的关联标签来删除镜像的任何不需要的标签。从镜像中删除最后一个标签后，也会删除该镜像。

```
aws ecr batch-delete-image \  
  --repository-name my-repo \  
  --image-ids imageTag=tag1 imageTag=tag2
```

3. 通过指定镜像摘要删除带标签或不带标签的镜像。在通过引用镜像摘要来删除镜像时，镜像及其所有标签都会被删除。

```
aws ecr batch-delete-image \  
  --repository-name my-repo \  
  --image-ids imageDigest=sha256:4f70ef7a4d29e8c0c302b13e25962d8f7a0bd304EXAMPLE
```

要删除多个镜像，您可以在请求中指定多个镜像标签或镜像摘要。

```
aws ecr batch-delete-image \  
  --repository-name my-repo \  
  --image-ids imageDigest=sha256:4f70ef7a4d29e8c0c302b13e25962d8f7a0bd304EXAMPLE  
  imageDigest=sha256:f5t0e245ssffc302b13e25962d8f7a0bd304EXAMPLE
```

在 Amazon ECR 中重新标记图片

借助 Docker Image Manifest V2 Schema 2 镜像，可以使用 `--image-tag` 命令的 `put-image` 选项重新为现有镜像添加标签。无需使用 Docker 提取或推送镜像，即可重新添加标签。对于大型镜像，此过程可大大节省重新为镜像添加标签所需的网络带宽和时间。

重新为镜像添加标签 (AWS CLI)

要使用重新标记图像 AWS CLI

1. 使用 `batch-get-image` 命令可获取要重新添加标签的镜像的镜像清单并将其写入文件中。在此示例中，标签为 *latest*，所在存储库为 *amazonlinux* 的镜像的清单被写入名为 *MANIFEST* 的环境变量中。

```
MANIFEST=$(aws ecr batch-get-image --repository-name amazonlinux --image-ids  
imageTag=latest --output text --query 'images[].imageManifest')
```

2. 使用 `put-image` 命令的 `--image-tag` 选项将镜像清单与新标签一起放置到 Amazon ECR 中。在此示例中，镜像的标签为 **2017.03**。

Note

如果该 `--image-tag` 选项在您的版本中不可用 AWS CLI，请升级到最新版本。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 put-image --repository-name amazonlinux --image-tag 2017.03 --image-  
manifest "$MANIFEST"
```

3. 验证您的新镜像标签是否已附加到您的镜像。在以下输出中，镜像具有标签 `latest` 和 `2017.03`。

```
aws ecr describe-images --repository-name amazonlinux
```

您可以在一个 (扩展) 代码行中执行所有这些操作：

```
{  
  "imageDetails": [  
    {  
      "imageSizeInBytes": 98755613,  
      "imageDigest":  
"sha256:8d00af8f076eb15a33019c2a3e7f1f655375681c4e5be157a26EXAMPLE",  
      "imageTags": [  
        "latest",  
        "2017.03"  
      ],  
      "registryId": "aws_account_id",  
      "repositoryName": "amazonlinux",  
      "imagePushedAt": 1499287667.0  
    }  
  ]  
}
```

重新为镜像添加标签 (AWS Tools for Windows PowerShell)

要使用重新标记图像 AWS Tools for Windows PowerShell

1. 使用 `Get-ECRImageBatch` cmdlet 获取要重新添加标签的镜像的描述，并将该镜像写入到环境变量。在此示例中，标签为 `latest`、所在存储库为 `amazonlinux` 的镜像被写入环境变量 `$Image` 中。

Note

如果您的系统上没有可用的 `Get-ECRImageBatch` cmdlet，请参阅 [AWS Tools for Windows PowerShell 用户指南中的设置 AWS Tools for Windows PowerShell](#)。

```
$Image = Get-ECRImageBatch -ImageId @{ imageTag="latest" } -  
RepositoryName amazonlinux
```

2. 将该镜像的清单写入到 `$Manifest` 环境变量。

```
$Manifest = $Image.Images[0].ImageManifest
```

3. 使用 `Write-ECRImage` cmdlet 的 `-ImageTag` 选项将镜像清单与新标签一起放置到 Amazon ECR 中。在此示例中，镜像的标签为 `2017.09`。

```
Write-ECRImage -RepositoryName amazonlinux -ImageManifest $Manifest -  
ImageTag 2017.09
```

4. 验证您的新镜像标签是否已附加到您的镜像。在以下输出中，镜像具有标签 `latest` 和 `2017.09`。

```
Get-ECRImage -RepositoryName amazonlinux
```

您可以在一个 (扩展) 代码行中执行所有这些操作：

```
ImageDigest                                                    ImageTag  
-----                                                    -  
sha256:359b948ea8866817e94765822787cd482279eed0c17bc674a7707f4256d5d497 latest  
sha256:359b948ea8866817e94765822787cd482279eed0c17bc674a7707f4256d5d497 2017.09
```

防止在 Amazon ECR 中覆盖图片标签

您可以通过在存储库中开启标签不可变性来防止图像标签被覆盖。启用标签不可变性后，如果您推送的图像的标签已存在于存储库中，则会返回 `ImageTagAlreadyExistsException` 错误。标签不可变性会影响所有标签。你不能使某些标签不可变，而其他标签则不是。

您可以使用 AWS Management Console 和 AWS CLI 工具为新存储库或现有存储库设置图像标签的可变性。要使用控制台步骤创建存储库，请参阅 [创建 Amazon ECR 私有存储库来存储图像](#)。

设置图像标签的可变性 ()AWS Management Console

设置图像标签的可变性

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择包含要编辑的存储库的区域。
3. 在导航窗格中，选择存储库。
4. 在 Repositories (存储库) 页面上，选择 Private (私有) 选项卡，然后选择要编辑的存储库，并选择 Edit (编辑)。
5. 对于标签不变性，选择存储库的标签可变性设置。配置有不可变标签的存储库会阻止覆盖镜像标签。有关更多信息，请参阅 [防止在 Amazon ECR 中覆盖图片标签](#)。
6. 对于 Image scan settings (镜像扫描设置)，尽管您可以在存储库级别为基础扫描指定扫描设置，但最佳实践是在私有注册表级别指定扫描配置。通过在私有注册表级别指定扫描设置，您可以启用增强扫描或基本扫描，并定义用于指定扫描哪些存储库的筛选条件。有关更多信息，请参阅 [扫描图像以查找 Amazon ECR 中的软件漏洞](#)。
7. 对于 Encryption settings (加密设置)，此字段仅供查看，因为存储库的加密设置在存储库创建完成之后无法更改。
8. 选择保存以更新存储库设置。

设置图像标签的可变性 ()AWS CLI

创建配置有不可变标签的存储库

使用以下命令之一创建配置有不可变标签的新镜像存储库。

- [create-repository](#) (AWS CLI)

```
aws ecr create-repository --repository-name name --image-tag-mutability IMMUTABLE --region us-east-2
```

- [New-ECRRepository](#) (AWS Tools for Windows PowerShell)

```
New-ECRRepository -RepositoryName name -ImageTagMutability IMMUTABLE -Region us-east-2 -Force
```

更新存储库的图像标签可变性设置

使用以下命令之一更新现有存储库的镜像标签可变性设置。

- [put-image-tag-mutability](#) (AWS CLI)

```
aws ecr put-image-tag-mutability --repository-name name --image-tag-mutability IMMUTABLE --region us-east-2
```

- [write-ecr 可变性 \(\) ImageTag](#) AWS Tools for Windows PowerShell

```
Write-ECRImageTagMutability -RepositoryName name -ImageTagMutability IMMUTABLE -Region us-east-2 -Force
```

Amazon ECR 中支持容器镜像清单格式

Amazon ECR 支持以下容器镜像清单格式：

- Docker Image Manifest V2 Schema 1 (与 Docker 版本 1.9 和更早版本配合使用)
- Docker Image Manifest V2 Schema 2 (与 Docker 版本 1.10 和更新版本配合使用)
- Open Container Initiative (OCI) 规范 (v1.0 和更高版本)

对 Docker Image Manifest V2 Schema 2 的支持可提供以下功能：

- 能够为单个镜像使用多个标签。
- 支持存储 Windows 容器镜像。

Amazon ECR 镜像清单转换

在 Amazon ECR 中推送和提取镜像时，您的容器引擎客户端（例如 Docker）将与注册表进行通信以就客户端了解的清单格式以及要用于镜像的注册表达成一致。

在使用 Docker 版本 1.9 或更旧版本将镜像推送到 Amazon ECR 时，镜像清单格式将存储为 Docker Image Manifest V2 Schema 1。在使用 Docker 版本 1.10 或更新版本将镜像推送到 Amazon ECR 时，镜像清单格式将存储为 Docker Image Manifest V2 Schema 2。

在从 Amazon ECR 按标签提取镜像时，Amazon ECR 将返回存储在存储库中的镜像清单格式。仅当客户端理解该格式时，才会将其返回。如果客户端不理解所存储的镜像清单格式，则 Amazon ECR 会将镜像清单转换为客户端能够理解的格式。例如，如果 Docker 1.9 客户端请求的镜像清单存储格式为 Docker Image Manifest V2 Schema 2，那么 Amazon ECR 将以 Docker Image Manifest V2 Schema 1 格式返回该清单。下表介绍了按标签提取镜像时 Amazon ECR 支持的可用转换：

客户端请求的架构	作为 V2 Schema 1 推送到 ECR	作为 V2 Schema 2 推送到 ECR	作为 OCI 推送到 ECR
V2 Schema 1	无需转换	已转换为 V2 Schema 1	已转换为 V2 Schema 1
V2 Schema 2	无可用的转换，客户端将回退到 V2 Schema 1	无需转换	已转换为 V2 Schema 2
OCI	无可用的转换	已转换为 OCI	无需转换

Important

如果按摘要提取镜像，则没有可用的转换。您的客户端必须了解存储在 Amazon ECR 中的镜像清单格式。如果您在 Docker 1.9 或更旧版本的客户端上按摘要请求 Docker Image Manifest V2 Schema 2 镜像，则无法提取镜像。有关更多信息，请参阅 Docker 文档中的[注册表兼容性](#)。

在此示例中，如果按标签请求同一镜像，Amazon ECR 会将镜像清单转换为客户端能够理解的格式。镜像提取成功。

将 Amazon ECR 映像与 Amazon ECS 结合使用

您可以使用 Amazon ECR 私有存储库来托管容器镜像和构件，您的 Amazon ECS 任务可能会从中提取这些镜像和构件。为此，Amazon ECS 或 Fargate 容器代理必须有进行 `ecr:BatchGetImage`、`ecr:GetDownloadUrlForLayer` 或 `ecr:GetAuthorizationToken` API 调用的权限。

所需的 IAM 权限

下表显示了每种启动类型要使用的 IAM 角色，该角色为您的任务从 Amazon ECR 私有存储库提取提供了所需的权限。Amazon ECS 提供包括所需权限的托管 IAM 策略。

启动类型	IAM 角色	AWS 托管 IAM 策略
Amazon EC2 实例上的 Amazon ECS	使用容器实例 IAM 角色，该角色与注册到 Amazon ECS 集群的 Amazon EC2 实例相关联。有关更多信息，请参阅 Amazon Elastic Container Service 开发人员指南中的 容器实例 IAM 角色 。	AmazonEC2ContainerServiceforEC2Role 有关更多信息，请参阅 Amazon Elastic Container Service 开发人员指南中的 AmazonEC2ContainerServiceforEC2Role
Fargate 上的 Amazon ECS	使用您在 Amazon ECS 任务定义中引用的任务执行 IAM 角色。有关更多信息，请参阅 Amazon Elastic Container Service 开发人员指南中的 任务执行 IAM 角色 。	AmazonECSTaskExecutionRolePolicy 有关更多信息，请参阅 Amazon Elastic Container Service 开发人员指南中的 AmazonECSTaskExecutionRolePolicy 。
外部实例上的 Amazon ECS	使用容器实例 IAM 角色，该角色与注册到 Amazon ECS 集群的本地服务器或虚拟机 (VM) 相关联。有关更多信息，请参阅 Amazon Elastic Container	AmazonEC2ContainerServiceforEC2Role 有关更多信息，请参阅 Amazon Elastic Container Service 开发人员指南中

启动类型	IAM 角色	AWS 托管 IAM 策略
	Service 开发人员指南中的容器实例 Amazon ECS 角色。	的 AmazonEC2ContainerServiceforEC2Role 。

Important

AWS 托管 IAM 策略包含您可能不需要的额外权限即可使用。在这种情况下，这些是从 Amazon ECR 私有存储库提取所需的最低权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer",
        "ecr:GetAuthorizationToken"
      ],
      "Resource": "*"
    }
  ]
}
```

在 Amazon ECS 任务定义中指定 Amazon ECR 镜像

创建 Amazon ECS 任务定义时，您可以指定在 Amazon ECR 私有存储库中托管的容器镜像。在任务定义中，确保对您的 Amazon ECR 镜像使用完整的 `registry/repository:tag` 命名。例如：`aws_account_id.dkr.ecr.region.amazonaws.com/my-repository:latest`。

以下任务定义代码段显示了用于指定在 Amazon ECS 任务定义中的 Amazon ECR 中托管的容器映像的语法。

```
{
  "family": "task-definition-name",
  ...
  "containerDefinitions": [
```

```
{
  "name": "container-name",
  "image": "aws_account_id.dkr.ecr.region.amazonaws.com/my-
repository:latest",
  ...
},
...
}
```

将 Amazon ECR 映像与 Amazon EKS 结合使用

您可以将您的亚马逊 ECR 镜像与亚马逊 EKS 一起使用。

从 Amazon ECR 中引用映像时，您必须为映像使用完整的 registry/repository:tag 命名。例如，`aws_account_id.dkr.ecr.region.amazonaws.com/my-repository:latest`。

所需的 IAM 权限

如果您将 Amazon EKS 工作负载托管在托管节点、自我管理节点或上 AWS Fargate，请查看以下内容：

- 托管在托管或自行管理节点上的 Amazon EKS 工作负载：需要使用 Amazon EKS 工作节点 IAM 角色 (NodeInstanceRole)。Amazon EKS Worker 节点 IAM 角色必须包含以下适用于 Amazon ECR 的 IAM policy 权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer",
        "ecr:GetAuthorizationToken"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

如果您使用 `eksctl` 或 [Amazon EKS 入门](#) 中的 AWS CloudFormation 模板来创建集群和工作节点组，则默认情况下，这些 IAM 权限将应用于您的工作节点 IAM 角色。

- 托管于 Amazon EKS 工作负载 AWS Fargate：使用 Fargate 容器执行角色，该角色允许你的 pod 从私有 Amazon ECR 存储库中提取映像。有关更多信息，请参阅 [创建 Fargate Pod 执行角色](#)。

在亚马逊 EKS 集群上安装 Helm 图表

Amazon ECR 中托管的 Helm 图表可以安装在你的 Amazon EKS 集群上。

先决条件

- 安装最新版本的 Helm 客户端。这些步骤是使用 Helm 版本 3.9.0 编写的。有关更多信息，请参阅 [安装 Helm](#)。
- 您至少已在计算机上安装了 AWS CLI 的版本 1.23.9 或 2.6.3。有关更多信息，请参阅 [安装或更新 AWS CLI 的最新版本](#)。
- 您已将 Helm Chart 推送到您的 Amazon ECR 存储库。有关更多信息，请参阅 [将 Helm 图表推送到 Amazon ECR 私有存储库](#)。
- 您已配置 `kubectl` 以使用 Amazon EKS。有关更多信息，请参阅 Amazon EKS 用户指南中的 [为 Amazon EKS 创建 kubeconfig](#)。如果集群的以下命令成功，说明您已正确配置。

```
kubectl get svc
```

在 Amazon EKS 集群上安装 Helm 图表

1. 向托管 Helm Chart 的 Amazon ECR 注册表验证您的 Helm 客户端。必须针对每个注册表获得授权令牌，令牌有效期为 12 小时。有关更多信息，请参阅 [Amazon ECR 中的私有注册表身份验证](#)。

```
aws ecr get-login-password \
  --region us-west-2 | helm registry login \
  --username AWS \
  --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

2. 安装图表。`helm-test-chart` 用你的存储库替换，用 Helm 图表的标签替换 `0.1.0`。

```
helm install ecr-chart-demo oci://aws_account_id.dkr.ecr.region.amazonaws.com/helm-test-chart --version 0.1.0
```

输出应如下所示：

```
NAME: ecr-chart-demo
LAST DEPLOYED: Tue May 31 17:38:56 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
```

3. 验证图表安装。

```
helm list -n default
```

输出示例：

NAME	NAMESPACE	REVISION	UPDATED
STATUS	CHART	APP VERSION	
ecr-chart-demo	default	1	2022-06-01 15:56:40.128669157 +0000
UTC deployed	helm-test-chart-0.1.0	1.16.0	

4. (可选) 查看已安装的 Helm 图表 ConfigMap。

```
kubectl describe configmap helm-test-chart-configmap
```

5. 完成后，您可以从集群中删除图表版本。

```
helm uninstall ecr-chart-demo
```

扫描图像以查找 Amazon ECR 中的软件漏洞

改进后的基本扫描功能在 Amazon ECR 的预览版中，可能会发生变化。在此公开预览期间，您只能使用选择 AWS Management Console 加入改进版基本扫描版本。

Amazon ECR 图像扫描有助于识别容器映像中的软件漏洞。提供以下扫描类型。

Important

在增强扫描、基本扫描和改进的基本扫描版本之间切换将导致先前建立的扫描不再可用。你必须重新设置扫描。但是，如果您切换回以前的扫描版本，则已建立的扫描将可用。

- **增强扫描**—Amazon ECR 与 Amazon Inspector 集成以提供对您的存储库的自动连续扫描。扫描容器镜像是否存在操作系统和编程语言包漏洞。当出现新的漏洞时，扫描结果会更新，Amazon Inspector 会发出一个事件 EventBridge 来通知您。增强型扫描提供以下功能：
 - 操作系统和编程语言包漏洞。
 - 两种扫描频率：按下扫描和连续扫描。
- **基本扫描**— Amazon ECR 提供了两个使用常见漏洞和暴露 (CVE) 数据库的基本扫描版本：使用开源 Clair 项目的当前 GA 版本和使用我们原生技术的新改进版基本扫描（预览版）。AWS 通过基本扫描，您可以将存储库配置为在推送时扫描，也可以执行手动扫描，而 Amazon ECR 会提供扫描结果列表。基本扫描提供以下内容：
 - 操作系统扫描。
 - 两种扫描频率：手动和按下扫描。

Important

DescribeImagesAPI 中不支持imageScanFindingsSummary新版本的基本扫描。imageScanStatus要查看这些内容，请使用 DescribeImageScanFindings API。

用于选择在 Amazon ECR 中扫描哪些存储库的筛选器

为私有注册表配置图像扫描时，您可以使用筛选器来选择要扫描哪些存储库。

使用 basic (基本) 扫描时，您可以指定按推送筛选条件进行扫描，以指定将哪些存储库设置为在推送新镜像时进行镜像扫描。对于任何不符合基本扫描的推送筛选条件的存储库，都将设置为 manual (手动) 扫描频率，这意味着必须手动触发扫描才能执行扫描。

使用 enhanced (增强) 扫描时，您可以为推送扫描和连续扫描分别指定筛选条件。任何不符合增强扫描筛选条件的存储库都将禁用扫描。如果您使用增强扫描并为推送扫描和连续扫描分别指定了筛选条件，并且同一存储库符合多个筛选条件，则 Amazon ECR 会强制执行该存储库的连续扫描筛选条件，而不是推送扫描筛选条件。

筛选通配符

指定筛选条件后，没有通配符的筛选条件将匹配包含该筛选条件的所有存储库名称。带通配符 (*) 的筛选条件匹配任何存储库名称，通配符会在其中替换存储库名称中的零个或多个字符。

下表提供了示例，其中存储库名称在水平轴上表示，示例筛选条件在垂直轴上指定。

	prod	repo-prod	prod-repo	repo-prod-repo	prodrepo
prod	是	是	是	是	是
*prod	是	是	否	否	否
prod*	是	否	是	否	是
prod	是	是	是	是	是
prod*repo	否	否	是	否	是

扫描图像以查找 Amazon ECR 中的操作系统和编程语言包漏洞

Amazon ECR 增强扫描是与 Amazon Inspector 的集成，它为您的容器镜像提供漏洞扫描。将会扫描容器映像是否存在操作系统和编程语言包漏洞。您可以使用 Amazon ECR 和 Amazon Inspector 直接查看扫描结果。有关 Amazon Inspector 的更多信息，请参阅 Amazon Inspector 用户指南中的[使用 Amazon Inspector 扫描容器镜像](#)。

借助增强扫描功能，您可以选择配置哪些存储库进行自动连续扫描，配置哪些存储库在推送时扫描。此操作通过设置扫描筛选条件完成。

增强扫描的注意事项

在启用 Amazon ECR 增强型扫描之前，请考虑以下几点。

- 使用此功能不会收取 Amazon ECR 的额外费用，但是 Amazon Inspector 会收取扫描映像的费用。有关更多信息，请参阅 [Amazon Inspector 定价](#)。
- 以下区域不支持增强扫描：
 - 中东 (阿联酋) (me-central-1)
 - 亚太地区 (海得拉巴) (ap-south-2)
 - 以色列 (特拉维夫) (il-central-1)
 - 亚太地区 (墨尔本) (ap-southeast-4)
 - 欧洲 (西班牙) (eu-south-2)
- Amazon Inspector 支持扫描特定操作系统。获取完整列表，请参阅 Amazon Inspector 用户指南中的 [支持的操作系统 - Amazon ECR 扫描](#)。
- Amazon Inspector 使用服务相关 IAM 角色，该角色提供了为存储库提供增强扫描所需的权限。为私有注册表开启增强扫描后，Amazon Inspector 会自动创建服务相关 IAM 角色。有关更多信息，请参阅 Amazon Inspector 用户指南中的 [将服务相关角色用于 Amazon Inspector](#)。
- 当您最初为私有注册表开启增强扫描功能时，Amazon Inspector 只能根据图片推送时间戳或过去 90 天内提取的图像识别过去 30 天内推送到 Amazon ECR 的图像。更早映像的扫描状态将为 SCAN_ELIGIBILITY_EXPIRED。如果您需要让 Amazon Inspector 扫描这些映像，则应将其重新推送到您的存储库。
- 开启增强扫描后推送到 Amazon ECR 的所有映像都会在配置的持续时间内持续扫描。默认情况下，持续时间为生命周期。此设置可通过 Amazon Inspector 控制台来配置。有关更多信息，请参阅 [在 Amazon Inspector 中更改图像的增强扫描时长](#)。
- 如果您的 Amazon ECR 私有注册表开启了增强扫描，则只能使用增强扫描功能来扫描符合扫描筛选条件的所有存储库。不符合任何筛选条件的任何存储库将使用 Off 扫描频率，并且不会被扫描。不支持使用增强扫描进行手动扫描。有关更多信息，请参阅 [用于选择在 Amazon ECR 中扫描哪些存储库的筛选器](#)。
- 如果您为推送扫描和连续扫描分别指定了筛选条件，并且同一存储库符合多个筛选条件，则 Amazon ECR 会强制执行该存储库的连续扫描筛选条件，而不是推送扫描筛选条件。
- 启用增强扫描后，当存储库的扫描频率发生变化 EventBridge 时，Amazon ECR 会向发送一个事件。EventBridge 当初始扫描完成以及创建、更新或关闭图像扫描结果时，Amazon Inspector 会发出事件。

在 Amazon ECR 中进行增强扫描所需的 IAM 权限

Amazon ECR 增强扫描需要 Amazon Inspector 服务相关 IAM 角色，且要求启用和使用增强扫描的 IAM 主体有权调用扫描所需的 Amazon Inspector API。为私有注册表开启增强扫描后，Amazon Inspector 会自动创建 Amazon Inspector 服务相关 IAM 角色。有关更多信息，请参阅 Amazon Inspector 用户指南中的[将服务相关角色用于 Amazon Inspector](#)。

以下 IAM policy 授予启用和使用增强扫描所需的权限。它包括 Amazon Inspector 创建服务相关 IAM 角色所需的权限，以及开启和关闭增强扫描和检索扫描结果所需的 Amazon Inspector API 权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "inspector2:Enable",
        "inspector2:Disable",
        "inspector2:ListFindings",
        "inspector2:ListAccountPermissions",
        "inspector2:ListCoverage"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": [
            "inspector2.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

在 Amazon ECR 中为图像配置增强扫描

为您的私有注册表配置每个区域的增强扫描。

确认您具有配置增强扫描的正确 IAM 权限。有关信息，请参阅 [在 Amazon ECR 中进行增强扫描所需的 IAM 权限](#)。

AWS Management Console

为私有注册表开启增强扫描功能

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择要为其设置扫描配置的区域。
3. 在导航窗格中，选择私有注册表、设置、扫描。
4. 在 Scanning configuration (扫描配置) 页面中，为 Scan type (扫描类型) 选择 Enhanced scanning (增强扫描)。

默认情况下，如果选择“增强扫描”，则会持续扫描您的所有存储库。

5. 要选择要持续扫描的特定存储库，请清除“持续扫描所有存储库”复选框，然后定义筛选器：

Important

没有通配符的筛选条件将匹配包含该筛选条件的所有存储库名称。带通配符 (*) 的筛选条件匹配存储库名称，通配符会替换存储库名称中的零个或多个字符。要查看过滤器行为的示例，请参阅 [the section called “筛选通配符”](#)。

- a. 输入基于存储库名称的筛选器，然后选择“添加筛选器”。
 - b. 决定推送图像时要扫描哪些存储库：
 - 要在推送时扫描所有存储库，请选择“推送所有仓库时扫描”。
 - 要选择要在推送时扫描的特定存储库，请输入基于存储库名称的筛选器，然后选择“添加筛选器”。
6. 选择保存。
 7. 在您要开启增强扫描的每个区域中重复这些步骤。

AWS CLI

使用以下 AWS CLI 命令开启对私有注册表的增强扫描 AWS CLI。您可以使用 rules 对象指定扫描筛选条件。

- [put-registry-scanning-configuration](#) (AWS CLI)

以下示例将为您私有注册表开启增强扫描。默认情况下，如果没有指定 `rules`，Amazon ECR 会将扫描配置设置为对所有存储库进行持续扫描。

```
aws ecr put-registry-scanning-configuration \  
  --scan-type ENHANCED \  
  --region us-east-2
```

以下示例将为您私有注册表开启增强扫描并指定扫描筛选条件。示例中的扫描筛选条件将对名称中带有 `prod` 的所有存储库开启持续扫描。

```
aws ecr put-registry-scanning-configuration \  
  --scan-type ENHANCED \  
  --rules '[{"repositoryFilters" : [{"filter": "prod", "filterType" :  
  "WILDCARD"}], "scanFrequency" : "CONTINUOUS_SCAN"}]' \  
  --region us-east-2
```

以下示例将为您私有注册表开启增强扫描并指定多个扫描筛选条件。示例中的扫描筛选条件将对名称中带有 `prod` 的所有存储库开启持续扫描，并且对所有其他存储库开启仅在推送时扫描。

```
aws ecr put-registry-scanning-configuration \  
  --scan-type ENHANCED \  
  --rules '[{"repositoryFilters" : [{"filter": "prod", "filterType" :  
  "WILDCARD"}], "scanFrequency" : "CONTINUOUS_SCAN"}, {"repositoryFilters" :  
  [{"filter": "*", "filterType" : "WILDCARD"}], "scanFrequency" : "SCAN_ON_PUSH"}]' \  
  --region us-west-2
```

在 Amazon Inspector 中更改图像的增强扫描时长

您可以更改 Amazon Inspector 持续扫描您的 Amazon ECR 私有存储库中图像的天数。默认情况下，当您的 Amazon ECR 私有注册表开启增强扫描时，Amazon Inspector 服务会持续监控您的存储库，直到删除映像或禁用增强扫描。可以使用 Amazon Inspector 设置更改 Amazon Inspector 扫描图像的持续时间。可用扫描持续时间为生命周期（默认）、180 天和 30 天。当存储库的扫描持续时间已过时，在列出扫描漏洞时，将会显示 `SCAN_ELIGIBILITY_EXPIRED` 的扫描状态。有关更多信息，请参阅 Amazon Inspector 用户指南中的 [更改 Amazon ECR 自动重新扫描持续时间](#)。

更改增强扫描持续时间设置

1. 打开 Amazon Inspector 控制台，[网址为 https://console.aws.amazon.com/inspector/v2/home](https://console.aws.amazon.com/inspector/v2/home)。
2. 在左侧导航中，展开 Settings (设置)，然后选择 General (通用)。
3. 在 Settings (设置) 页面上的 ECR re-scan duration (ECR 重新扫描持续时间) 下，选择一个设置，然后选择 Save (保存)。

EventBridge 为了在 Amazon ECR 中进行增强扫描而发送的事件

启用增强扫描后，当存储库的扫描频率发生变化 EventBridge 时，Amazon ECR 会向发送一个事件。EventBridge 当初始扫描完成以及创建、更新或关闭图像扫描结果时，Amazon Inspector 会向其发送事件。

存储库扫描频率更改事件

为注册表开启增强扫描后，当开启了增强扫描的资源发生更改时，Amazon ECR 将发送以下事件。这包括正在创建新存储库、正在更改存储库的扫描频率，或者在开启了增强扫描功能的存储库中创建或删除映像。有关更多信息，请参阅 [扫描图像以查找 Amazon ECR 中的软件漏洞](#)。

```
{
  "version": "0",
  "id": "0c18352a-a4d4-6853-ef53-0abEXAMPLE",
  "detail-type": "ECR Scan Resource Change",
  "source": "aws.ecr",
  "account": "123456789012",
  "time": "2021-10-14T20:53:46Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "action-type": "SCAN_FREQUENCY_CHANGE",
    "repositories": [{
      "repository-name": "repository-1",
      "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-1",
      "scan-frequency": "SCAN_ON_PUSH",
      "previous-scan-frequency": "MANUAL"
    },
    {
      "repository-name": "repository-2",
      "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-2",
      "scan-frequency": "CONTINUOUS_SCAN",
      "previous-scan-frequency": "SCAN_ON_PUSH"
    }
  ]
}
```

```
  },
  {
    "repository-name": "repository-3",
    "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-3",
    "scan-frequency": "CONTINUOUS_SCAN",
    "previous-scan-frequency": "SCAN_ON_PUSH"
  }
],
"resource-type": "REPOSITORY",
"scan-type": "ENHANCED"
}
}
```

初始镜像扫描的事件 (增强扫描)

为注册表开启增强扫描后，当初始映像扫描完成时，Amazon Inspector 会发送以下事件。finding-severity-counts 参数仅返回严重性级别的值 (如果存在)。例如，如果镜像不包含任何 CRITICAL 级别的结果，则不会返回任何关键计数。有关更多信息，请参阅 [扫描图像以查找 Amazon ECR 中的操作系统和编程语言包漏洞](#)。

事件模式：

```
{
  "source": ["aws.inspector2"],
  "detail-type": ["Inspector2 Scan"]
}
```

输出示例：

```
{
  "version": "0",
  "id": "739c0d3c-4f02-85c7-5a88-94a9EXAMPLE",
  "detail-type": "Inspector2 Scan",
  "source": "aws.inspector2",
  "account": "123456789012",
  "time": "2021-12-03T18:03:16Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ecr:us-east-2:123456789012:repository/amazon/amazon-ecs-sample"
  ],
  "detail": {
    "scan-status": "INITIAL_SCAN_COMPLETE",
```

```

    "repository-name": "arn:aws:ecr:us-east-2:123456789012:repository/amazon/
amazon-ecs-sample",
    "finding-severity-counts": {
      "CRITICAL": 7,
      "HIGH": 61,
      "MEDIUM": 62,
      "TOTAL": 158
    },
    "image-digest":
"sha256:36c7b282abd0186e01419f2e58743e1bf635808231049bbc9d77e5EXAMPLE",
    "image-tags": [
      "latest"
    ]
  }
}

```

镜像扫描结果更新的事件 (增强扫描)

为注册表开启增强扫描后，在创建、更新或关闭映像扫描结果时，Amazon Inspector 会发送以下事件。有关更多信息，请参阅 [扫描图像以查找 Amazon ECR 中的操作系统和编程语言包漏洞](#)。

事件模式：

```

{
  "source": ["aws.inspector2"],
  "detail-type": ["Inspector2 Finding"]
}

```

输出示例：

```

{
  "version": "0",
  "id": "42dbea55-45ad-b2b4-87a8-afaEXAMPLE",
  "detail-type": "Inspector2 Finding",
  "source": "aws.inspector2",
  "account": "123456789012",
  "time": "2021-12-03T18:02:30Z",
  "region": "us-east-2",
  "resources": [
    "arn:aws:ecr:us-east-2:123456789012:repository/amazon/amazon-ecs-sample/
sha256:36c7b282abd0186e01419f2e58743e1bf635808231049bbc9d77eEXAMPLE"
  ],
  "detail": {

```

```
"awsAccountId": "123456789012",
  "description": "In libssh2 v1.9.0 and earlier versions, the SSH_MSG_DISCONNECT
  logic in packet.c has an integer overflow in a bounds check, enabling an attacker to
  specify an arbitrary (out-of-bounds) offset for a subsequent memory read. A crafted
  SSH server may be able to disclose sensitive information or cause a denial of service
  condition on the client system when a user connects to the server.",
  "findingArn": "arn:aws:inspector2:us-east-2:123456789012:finding/
be674aadd0f75ac632055EXAMPLE",
  "firstObservedAt": "Dec 3, 2021, 6:02:30 PM",
  "inspectorScore": 6.5,
  "inspectorScoreDetails": {
    "adjustedCvss": {
      "adjustments": [],
      "cvssSource": "REDHAT_CVE",
      "score": 6.5,
      "scoreSource": "REDHAT_CVE",
      "scoringVector": "CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N",
      "version": "3.0"
    }
  },
  "lastObservedAt": "Dec 3, 2021, 6:02:30 PM",
  "packageVulnerabilityDetails": {
    "cvss": [
      {
        "baseScore": 6.5,
        "scoringVector": "CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N",
        "source": "REDHAT_CVE",
        "version": "3.0"
      },
      {
        "baseScore": 5.8,
        "scoringVector": "AV:N/AC:M/Au:N/C:P/I:N/A:P",
        "source": "NVD",
        "version": "2.0"
      },
      {
        "baseScore": 8.1,
        "scoringVector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:H",
        "source": "NVD",
        "version": "3.1"
      }
    ],
    "referenceUrls": [
      "https://access.redhat.com/errata/RHSA-2020:3915"
    ]
  }
}
```



```

    ],
    "source": "REDHAT_CVE",
    "sourceUrl": "https://access.redhat.com/security/cve/CVE-2019-17498",
    "vendorCreatedAt": "Oct 16, 2019, 12:00:00 AM",
    "vendorSeverity": "Moderate",
    "vulnerabilityId": "CVE-2019-17498",
    "vulnerablePackages": [
      {
        "arch": "X86_64",
        "epoch": 0,
        "name": "libssh2",
        "packageManager": "OS",
        "release": "12.amzn2.2",
        "sourceLayerHash":
"sha256:72d97abdfae3b3c933ff41e39779cc72853d7bd9dc1e4800c5294dEXAMPLE",
        "version": "1.4.3"
      }
    ]
  },
  "remediation": {
    "recommendation": {
      "text": "Update all packages in the vulnerable packages section to
their latest versions."
    }
  },
  "resources": [
    {
      "details": {
        "awsEcrContainerImage": {
          "architecture": "amd64",
          "imageHash":
"sha256:36c7b282abd0186e01419f2e58743e1bf635808231049bbc9d77e5EXAMPLE",
          "imageTags": [
            "latest"
          ],
          "platform": "AMAZON_LINUX_2",
          "pushedAt": "Dec 3, 2021, 6:02:13 PM",
          "registry": "123456789012",
          "repositoryName": "amazon/amazon-ecs-sample"
        }
      },
      "id": "arn:aws:ecr:us-east-2:123456789012:repository/amazon/amazon-ecs-
sample/sha256:36c7b282abd0186e01419f2e58743e1bf635808231049bbc9d77EXAMPLE",
      "partition": "N/A",

```

```
        "region": "N/A",
        "type": "AWS_ECR_CONTAINER_IMAGE"
    }
],
"severity": "MEDIUM",
"status": "ACTIVE",
"title": "CVE-2019-17498 - libssh2",
"type": "PACKAGE_VULNERABILITY",
"updatedAt": "Dec 3, 2021, 6:02:30 PM"
}
}
```

在 Amazon ECR 中检索结果以进行增强扫描

您可以检索上次完成的增强图像扫描的扫描结果，然后在 Amazon Inspector 中打开扫描结果以查看更多细节。根据常见漏洞和暴露 (CVE) 数据库，按严重程度列出了发现的软件漏洞。

有关扫描镜像时的常见问题的排查详细信息，请参阅 [对 Amazon ECR 中的图像扫描进行故障排除](#)。

AWS Management Console

通过 AWS Management Console 使用以下步骤检索镜像扫描结果。

检索图像扫描结果

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择您的存储库所在的区域。
3. 在导航窗格中，选择存储库。
4. 在存储库页面上，选择包含要扫描检索结果的镜像的存储库。
5. 在 Images (镜像) 页面的 Vulnerabilities (漏洞) 下，为要检索其扫描结果的镜像选择 See findings (查看详细信息) 。
6. 要在 Amazon Inspector 控制台中查看更多详细信息，请在“名称”列中选择漏洞名称。

AWS CLI

使用以下 AWS CLI 命令使用检索图像扫描结果 AWS CLI。您可以使用 `imageTag` 或 `imageDigest` 指定镜像，这两者都可以使用 [list-images](#) CLI 命令获取。

- [describe-image-scan-findings](#) (AWS CLI)

以下示例使用镜像标签。

```
aws ecr describe-image-scan-findings \  
  --repository-name name \  
  --image-id imageTag=tag_name \  
  --region us-east-2
```

以下示例使用镜像摘要。

```
aws ecr describe-image-scan-findings \  
  --repository-name name \  
  --image-id imageDigest=sha256_hash \  
  --region us-east-2
```

扫描图像以查找 Amazon ECR 中的操作系统漏洞

改进后的基本扫描功能在 Amazon ECR 的预览版中，可能会发生变化。在此公开预览期间，您只能使用选择 AWS Management Console 加入改进版基本扫描版本。

Amazon ECR 提供了两个版本的基本扫描，它们使用常见漏洞和暴露 (CVE) 数据库：

- 使用开源 Clair 项目的当前 GA 版本。有关 Clair 的更多信息，请参阅上的 [Clair](#)。GitHub
- 使用 AWS 原生技术的全新改进版基本扫描（预览版）。

Amazon ECR 使用来自上游分发源的 CVE 的严重性（如果有）。否则，将使用通用漏洞评分系统 (CVSS) 分数。CVSS 评分可用于获取 NVD 漏洞严重性评级。有关更多信息，请参阅 [NVD 漏洞严重性评级](#)。

两个版本的 Amazon ECR 基本扫描都支持筛选器，用于指定推送时要扫描哪些存储库。任何与推送扫描过滤器不匹配的存储库都将设置为手动扫描频率，这意味着您必须手动开始扫描。每隔 24 小时可以扫描一次图像。24 小时包括推送时的初始扫描（如果已配置）和任何手动扫描。

可以为每个镜像检索上次完成的镜像扫描结果。图像扫描完成后，Amazon ECR 会向亚马逊 EventBridge 发送一个事件。有关更多信息，请参阅 [亚马逊 ECR 事件和 EventBridge](#)。

区域支持以改进基本扫描

以下区域支持基本扫描的改进版：

- 亚太地区 (香港) (ap-east-1)
- 欧洲 (斯德哥尔摩) (eu-north-1)
- 中东 (巴林) (me-south-1)
- 亚太地区 (孟买) (ap-south-1)
- 欧洲 (巴黎) (eu-west-3)
- AWS GovCloud (美国东部) (us-gov-east-1)
- 非洲 (开普敦) (af-south-1)
- 亚太地区 (雅加达) (ap-southeast-3)
- 欧洲地区 (法兰克福) (eu-central-1)
- 欧洲地区 (爱尔兰) (eu-west-1)
- 南美洲 (圣保罗) (sa-east-1)
- 美国东部 (俄亥俄州) (us-east-2)
- AWS GovCloud (美国西部) (us-gov-west-1)
- 亚太地区 (东京) (ap-northeast-1)
- 亚太地区 (首尔) (ap-northeast-2)
- 亚太地区 (大阪) (ap-northeast-3)
- 欧洲 (米兰) (eu-south-1)
- 欧洲 (伦敦) (eu-west-2)
- 美国东部 (弗吉尼亚州北部) (us-east-1)
- 亚太地区 (新加坡) (ap-southeast-1)
- 亚太地区 (悉尼) (ap-southeast-2)
- 加拿大 (中部) (ca-central-1)
- 美国西部 (加利福尼亚北部) (us-west-1)
- 美国西部 (俄勒冈州) (us-west-2)
- 欧洲 (苏黎世) (eu-central-2)

操作系统支持基本扫描和改进的基本扫描

作为安全最佳实践，为了持续提供保障，我们建议您继续使用支持的操作系统版本。根据供应商政策，已停产的操作系统将不再使用补丁进行更新，而且在许多情况下，不再为它们发布新的安全公告。此外，当受影响的操作系统的支持期结束时，一些供应商会从他们的信息源中删除现有的安全公告和检测。在发行版失去供应商的支持后，Amazon ECR 可能不再支持对其进行漏洞扫描。Amazon ECR 针对已停产的操作系统得出的任何调查结果都应仅用于提供信息。下面列出了当前支持的操作系统和版本。

操作系统	版本
Alpine Linux (Alpine)	3.19
Alpine Linux (Alpine)	3.18
Alpine Linux (Alpine)	3.17
Alpine Linux (Alpine)	3.16
Amazon Linux 2 (AL2)	AL2
亚马逊 Linux 2023 (AL2023)	AL2023
CentOS Linux (CentOS)	7
Debian 服务器 (Bookworm)	12
Debian 服务器 (Bullseye)	11
Debian 服务器 (Buster)	10
Oracle Linux (Oracle)	9
Oracle Linux (Oracle)	8
Oracle Linux (Oracle)	7
Ubuntu (农历)	23.04
Ubuntu (Jammy)	22.04 (LTS)

操作系统	版本
Ubuntu (Focal)	20.04 (LTS)
Ubuntu (Bionic)	18.04 (ESM)
Ubuntu (Xenial)	16.04 (ESM)
Ubuntu (Trusty)	14.04 (ESM)
Red Hat Enterprise Linux (RHEL)	7
Red Hat Enterprise Linux (RHEL)	8
Red Hat Enterprise Linux (RHEL)	9

在 Amazon ECR 中为图像配置改进的基本扫描功能

Amazon ECR 基本扫描的改进版本现已提供预览版。使用 AWS 原生技术改进了基本扫描。

为您的私有仓库配置改进的每个区域的基本扫描。有关支持改进基本扫描的区域列表，请参阅[区域支持以改进基本扫描](#)。

为私有注册表开启改进版基本扫描

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择要为其设置扫描配置的区域。
3. 在导航窗格中，选择 Private registry (私有注册表)、Scanning (扫描)。
4. 在扫描配置页面上，对于扫描类型，选择改进的基本扫描 (预览中)-新增。
5. 默认情况下，您的所有存储库都设置为手动扫描。您可以选择通过指定推送时扫描筛选条件来开启推送时扫描。您可以为所有存储库或单个存储库设置推送时扫描。有关更多信息，请参阅[用于选择在 Amazon ECR 中扫描哪些存储库的筛选器](#)。

在 Amazon ECR 中为图像配置基本扫描

默认情况下，Amazon ECR 会开启对所有私有注册表的基本扫描。因此，除非您更改了私人注册表上的扫描设置，否则无需开启基本扫描。基本扫描使用开源 Clair 项目。

您可以使用以下步骤在推送过滤器上定义一个或多个扫描。

为私有注册表开启基本扫描

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择要为其设置扫描配置的区域。
3. 在导航窗格中，选择 Private registry (私有注册表)、Scanning (扫描)。
4. 在 Scanning configuration (扫描配置) 页面中，为 Scan type (扫描类型) 选择 Basic scanning (基本扫描)。
5. 默认情况下，您的所有存储库都设置为手动扫描。您可以选择通过指定推送时扫描筛选条件来开启推送时扫描。您可以为所有存储库或单个存储库设置推送时扫描。有关更多信息，请参阅 [用于选择在 Amazon ECR 中扫描哪些存储库的筛选器](#)。

手动扫描映像以查找 Amazon ECR 中的操作系统漏洞

如果您的仓库未配置为推送时扫描，则可以手动启动图像扫描。每隔 24 小时可以扫描一次图像。24 小时包括推送时的初始扫描 (如果已配置) 和任何手动扫描。

有关扫描镜像时的常见问题的排查详细信息，请参阅 [对 Amazon ECR 中的图像扫描进行故障排除](#)。

AWS Management Console

通过 AWS Management Console 使用以下步骤开始手动镜像扫描。

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择您创建存储库的区域。
3. 在导航窗格中，选择存储库。
4. 在存储库页面上，选择包含要扫描的镜像的存储库。
5. 在镜像页面上，选择要扫描的镜像，然后选择扫描。

AWS CLI

- [start-image-scan](#) (AWS CLI)

以下示例使用镜像标签。

```
aws ecr start-image-scan --repository-name name --image-id imageTag=tag_name --  
region us-east-2
```

以下示例使用镜像摘要。

```
aws ecr start-image-scan --repository-name name --image-id imageDigest=sha256_hash  
--region us-east-2
```

AWS Tools for Windows PowerShell

- [Get-ecr 调查结果 \(\) ImageScan](#) AWS Tools for Windows PowerShell

以下示例使用镜像标签。

```
Start-ECRImageScan -RepositoryName name -ImageId_ImageTag tag_name -Region us-  
east-2 -Force
```

以下示例使用镜像摘要。

```
Start-ECRImageScan -RepositoryName name -ImageId_ImageDigest sha256_hash -  
Region us-east-2 -Force
```

在 Amazon ECR 中检索基本扫描的结果

您可以检索上次完成的基本图像扫描的扫描结果。根据常见漏洞和暴露 (CVE) 数据库，按严重程度列出了发现的软件漏洞。

有关扫描镜像时的常见问题的排查详细信息，请参阅 [对 Amazon ECR 中的图像扫描进行故障排除](#)。

AWS Management Console

通过 AWS Management Console 使用以下步骤检索镜像扫描结果。

检索图像扫描结果

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择您创建存储库的区域。

3. 在导航窗格中，选择存储库。
4. 在存储库页面上，选择包含要扫描检索结果的镜像的存储库。
5. 在镜像页面的漏洞列下，选择要检索其扫描结果的镜像的详细信息。

AWS CLI

使用以下 AWS CLI 命令使用检索图像扫描结果 AWS CLI。您可以使用 `imageTag` 或 `imageDigest` 指定镜像，这两者都可以使用 [list-images](#) CLI 命令获取。

- [describe-image-scan-findings](#) (AWS CLI)

以下示例使用镜像标签。

```
aws ecr describe-image-scan-findings --repository-name name --image-id  
imageTag=tag_name --region us-east-2
```

以下示例使用镜像摘要。

```
aws ecr describe-image-scan-findings --repository-name name --image-id  
imageDigest=sha256_hash --region us-east-2
```

AWS Tools for Windows PowerShell

- [Get-ecr 调查结果 \(\) ImageScan](#) AWS Tools for Windows PowerShell

以下示例使用镜像标签。

```
Get-ECRImageScanFinding -RepositoryName name -ImageId_ImageTag tag_name -  
Region us-east-2
```

以下示例使用镜像摘要。

```
Get-ECRImageScanFinding -RepositoryName name -ImageId_ImageDigest sha256_hash -  
Region us-east-2
```

对 Amazon ECR 中的图像扫描进行故障排除

以下是常见的镜像扫描失败。您可以在 Amazon ECR 控制台中显示图片详情、通过 API 或使用 API AWS CLI 来查看此类错误。DescribeImageScanFindings

UnsupportedImage 错误

尝试对使用 Amazon ECR 不支持其基础镜像扫描的操作系统构建的镜像进行基础扫描时，您可能会得到 `UnsupportedImageError` 错误。Amazon ECR 支持 Amazon Elastic Container Storage Storage Simple Storage、Amazon Linux 2、Debian、Ubuntu、CentOS、Oracle Linux、Alpine 和 RHEL Linux 发行版的主要版本。一旦分发失去其供应商的支持，Amazon ECR 可能不再支持扫描它是否存在漏洞。Amazon ECR 不支持扫描从 [Docker scratch](#) 镜像构建的镜像。

Important

使用增强型扫描时，Amazon Inspector 支持扫描特定操作系统和媒体类型。有关完整列表，请参阅 Amazon Inspector 用户指南中的[支持的操作系统和媒体类型](#)。

返回 UNDEFINED 严重性级别

您可能会收到严重性级别为 UNDEFINED 的扫描结果。造成这种情况的常见原因如下：

- CVE 源未向该漏洞分配优先级。
- 该漏洞被分配了一个 Amazon ECR 无法识别的优先级。

要确定漏洞的严重性和描述，您可以直接从源查看 CVE。

了解扫描状态 `SCAN_ELIGIBILITY_EXPIRED`

如果您的私有注册表启用了使用 Amazon Inspector 的增强扫描，并且您正在查看扫描漏洞，则可能会看到扫描状态为 `SCAN_ELIGIBILITY_EXPIRED`。造成这种情况的常见原因如下。

- 当您为私有注册表初始开启增强扫描时，Amazon Inspector 将仅根据映像推送时间戳识别过去 30 天内推送到 Amazon ECR 的映像。更早映像的扫描状态将为 `SCAN_ELIGIBILITY_EXPIRED`。如果您需要让 Amazon Inspector 扫描这些映像，则应将其重新推送到您的存储库。
- 如果在 Amazon Inspector 控制台中更改了 ECR 重新扫描持续时间并且该时间已过，则镜像的扫描状态将会更改为 `inactive` 并显示原因代码 `expired`，并且将会计划关闭该镜像的所有关联调查结果。这会导致 Amazon ECR 控制台将扫描状态列为 `SCAN_ELIGIBILITY_EXPIRED`。

将上游注册表与 Amazon ECR 私有注册表同步

使用拉取缓存规则，您可以将上游注册表的内容与 Amazon ECR 私有注册表同步。

Amazon ECR 目前支持为以下上游注册表创建缓存提取规则。

- Docker Hub、微软 Azure 容器注册表、GitHub 容器注册表和 GitLab 容器注册表 (需要身份验证)
- Amazon ECR Public、Kubernetes 容器映像注册表和 Quay (不需要身份验证)

对于 GitLab 容器注册表，Amazon ECR 仅支持通过 GitLab software-as-a-service 提供的产品 GitLab .com 提取缓存。

对于需要身份验证的上游注册表，您必须将凭据存储在 AWS Secrets Manager 密钥中。通过 Amazon ECR 控制台，您可以轻松地每个经过身份验证的上游注册表创建 Secrets Manager 密钥。有关使用 Secrets Manager 控制台创建 Secrets Manager 密钥的更多信息，请参阅[将您的上游存储库凭证存储在 AWS Secrets Manager 密钥中](#)。

为上游注册表创建缓存提取规则后，只需使用 Amazon ECR 私有注册表 URI 从该上游注册表中提取映像即可。然后，Amazon ECR 会创建一个存储库，在私有注册表中缓存该映像。在您随后对带有给定标签的缓存图像的拉取请求中，Amazon ECR 会检查上游注册表，以查看是否有带有该特定标签的镜像的新版本，并尝试至少每 24 小时更新一次私有注册表中的镜像。

存储库创建模板

Amazon ECR 已支持存储库创建模板 (目前处于预览阶段)。这样的话，在为 Amazon ECR 使用缓存提取规则代表您创建的新存储库指定初始配置时，您有控制权。每个模板都包含存储库命名空间前缀，用于将新存储库与特定模板匹配。模板可以指定所有存储库设置的配置，包括基于资源的访问策略、标签不变性、加密和生命周期策略。存储库创建模板中的设置仅在存储库创建期间应用，对现有存储库或使用任何其他方法创建的存储库没有任何影响。有关更多信息，请参阅[用于控制在缓存拉取操作期间创建的存储库的模板](#)。

使用直通缓存规则的注意事项

使用 Amazon ECR 提取缓存规则时，请考虑以下几点。

- 以下区域不支持创建缓存提取规则。
 - 中国 (北京) (cn-north-1)

- 中国 (宁夏) (cn-northwest-1)
- AWS GovCloud (美国东部) (us-gov-east-1)
- AWS GovCloud (美国西部) (us-gov-west-1)
- AWS Lambda 不支持使用拉取缓存规则从 Amazon ECR 提取容器映像。
- 如果使用提取缓存提取镜像，首次提取镜像时不支持 Amazon ECR FIPS 服务终端节点。但是，使用 Amazon ECR FIPS 服务终端节点可以处理后续提取。
- 当通过 Amazon ECR 私有注册表 URI 提取缓存图像时，图像提取将由 AWS IP 地址启动。这可以确保映像提取不被计入上游注册表所实施的任何提取率配额。
- 当通过 Amazon ECR 私有注册表 URI 提取缓存的映像时，Amazon ECR 至少每 24 小时检查一次上游存储库，以验证缓存的映像是否为最新版本。如果上游注册表中有较新的映像，Amazon ECR 会尝试更新缓存的映像。此计时器基于缓存镜像的最后一次提取。
- 如果 Amazon ECR 出于任何原因无法从上游注册表更新映像，并且映像已提取，则系统仍会提取最后缓存的映像。
- 在创建包含上游注册表凭证的 Secrets Manager 密钥时，密钥名称必须使用 `ecr-pullthroughcache/` 前缀。密钥还必须与缓存提取规则位于相同的账户和区域。
- 当使用缓存提取规则提取多架构镜像时，清单列表和清单列表中引用的每个镜像都会被提取到 Amazon ECR 存储库中。如果您只想提取特定架构，则可以使用与架构关联的镜像摘要或标签，而不是与清单列表关联的标签来提取镜像。
- Amazon ECR 使用服务相关的 IAM 角色，为 Amazon ECR 提供所需的权限，以创建存储库、检索用于身份验证的 Secrets Manager 密钥，以及代表您推送缓存映像。服务相关 IAM 角色在创建缓存提取规则时自动创建。有关更多信息，请参阅 [用于缓存提取的 Amazon ECR 服务相关角色](#)。
- 默认情况下，提取缓存映像的 IAM 主体具有通过其 IAM policy 授予的权限。您可以使用 Amazon ECR 私有注册表权限策略进一步限定 IAM 实体的权限范围。有关更多信息，请参阅 [使用注册表权限](#)。
- 使用缓存提取工作流创建的 Amazon ECR 存储库与任何其他 Amazon ECR 存储库受到同等对待。支持所有存储库功能，例如复制和镜像扫描。
- 当 Amazon ECR 使用提取缓存操作代表您创建新存储库时，除非存在匹配的存储库创建模板，否则以下默认设置将应用于存储库。您可以使用存储库创建模板，定义 Amazon ECR 代表您创建的存储库所应用的设置。有关更多信息，请参阅 [用于控制在缓存拉取操作期间创建的存储库的模板](#)。
 - 标签不变性 – 已关闭，标签可变、可被覆盖。
 - 加密 – 使用默认 AES256 加密。
 - 存储库权限 – 已忽略，不应用任何存储库权限策略。
 - 生命周期策略 – 已忽略，不应用任何生命周期策略。

- 资源标签 – 已忽略，不应用任何资源标签。
- 使用缓存提取规则为存储库启用映像标签不变性，这将阻止 Amazon ECR 使用相同的标签更新映像。
- 首次使用直通缓存规则拉取图像时，可能需要通往互联网的路由。在某些情况下，需要一条通往互联网的路由，因此最好设置一条路由，以免出现任何故障。因此，如果您已将 Amazon ECR 配置为使用接口 VPC 终端节点，AWS PrivateLink 则需要确保第一次拉取具有通往互联网的路由。一种方法是在同一 VPC 中创建一个带有 Internet 网关的公有子网，然后将所有出站流量从私有子网路由到公有子网。后续使用拉取缓存规则拉取图像不需要这样做。有关更多信息，请参阅《Amazon Virtual Private Cloud 用户指南》中的[示例路由选项](#)。

将上游注册表与 Amazon ECR 私有注册表同步所需的 IAM 权限

除了对私有注册表进行身份验证以及推送和提取映像时所需的 Amazon ECR API 权限之外，还需要以下其他权限才能有效使用缓存提取规则。

- `ecr:CreatePullThroughCacheRule` – 授予创建拉取缓存规则的权限。此权限必须通过基于身份的 IAM policy 授予。
- `ecr:BatchImportUpstreamImage` – 授权检索外部镜像并将其导入到您的私有注册表。可以通过使用私有注册表权限策略、基于身份的 IAM policy 或通过使用基于资源的存储库权限策略授予此权限。有关使用存储库权限的更多信息，请参阅[Amazon ECR 中的私有存储库政策](#)。
- `ecr:CreateRepository` – 授予在私有注册表中创建存储库的权限。如果存储缓存图像的存储库不存在，则需要此权限。可以通过基于身份的 IAM policy 或私有注册表权限策略授予此权限。
- `ecr:TagResource` – 授权将元数据标签添加到 Amazon RDS 资源。仅当以下情况时需要此权限：您提取的映像采用缓存提取规则，该规则具有关联的存储库创建模板，而该模板配置为向存储库添加资源标签。此权限必须通过基于身份的 IAM policy 授予。

使用注册表权限

Amazon ECR 私有注册表权限可用于限定各个 IAM 实体使用缓存提取的权限范围。如果 IAM policy 授予 IAM 实体的权限多于注册表权限策略授予的权限，则 IAM policy 优先。例如，如果用户已授予 `ecr:*` 权限，则无需额外的注册表级别权限。

要创建私有注册表的权限策略 (AWS Management Console)

1. 打开 Amazon ECR 控制台：<https://console.aws.amazon.com/ecr/>。
2. 从导航栏中，选择您在其中配置私有注册表权限语句的区域。

3. 在导航窗格中，选择 Private registry (私有注册表)、Registry permissions (注册表权限)。
4. 在 Registry permissions (注册表权限) 页面上，选择 Generate statement (生成语句)。
5. 对于要创建的每个缓存提取权限策略语句，请执行以下操作。
 - a. 对于 Policy type (策略类型)，请选择 Pull through cache policy (推送缓存策略)。
 - b. 对于 Statement id (语句 ID)，为推送缓存语句策略提供名称。
 - c. 对于 IAM entities (IAM 实体)，指定要包含在策略中的用户、组或角色。
 - d. 对于 Repository namespace (存储库命名空间)，选择要与策略关联的推送缓存规则。
 - e. 对于 Repository names (存储库名称)，指定要应用规则的存储库基本名称。例如，如果您想在 Amazon ECR Public 上指定 Amazon Linux 存储库，存储库名称将为 amazonlinux。

要创建私有注册表的权限策略 (AWS CLI)

使用以下 AWS CLI 命令通过指定私有注册表权限 AWS CLI。

1. 创建名为 `ptc-registry-policy.json` 的本地文件，其中包含注册表策略的内容。以下示例授予创建存储库并从 Amazon ECR Public 中拉取镜像的 `ecr-pull-through-cache-user` 权限，Amazon ECR Public 是与之前创建的拉取缓存规则相关联的上游源。

```
{
  "Sid": "PullThroughCacheFromReadOnlyRole",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/ecr-pull-through-cache-user"
  },
  "Action": [
    "ecr:CreateRepository",
    "ecr:BatchImportUpstreamImage"
  ],
  "Resource": "arn:aws:ecr:us-east-1:111122223333:repository/ecr-public/*"
}
```

Important

仅当存储缓存镜像的存储库不存在时才需要 `ecr:CreateRepository` 权限。例如，如果存储库创建操作和镜像拉取操作是由单独的 IAM 主体（例如管理员和开发人员）完成。

2. 使用 [put-registry-policy](#) 命令设置注册表策略。

```
aws ecr put-registry-policy \  
  --policy-text file://ptc-registry.policy.json
```

后续步骤

准备好开始使用缓存提取规则后，请执行以下后续步骤。

- 创建缓存提取规则。有关更多信息，请参阅 [在 Amazon ECR 中创建直通缓存规则](#)。
- 创建存储库创建模板。有了存储库创建模板，在为 Amazon ECR 在缓存提取操作期间代表您创建的新存储库定义设置时，您能拥有控制权。有关更多信息，请参阅 [用于控制在缓存拉取操作期间创建的存储库的模板](#)。

在 Amazon ECR 中创建直通缓存规则

对于每个包含要缓存在您的 Amazon ECR 私有注册表中的图像的 upstream 注册表，您必须创建一条直通缓存规则。

对于需要身份验证的 upstream 注册表，您必须将凭据存储在 Secrets Manager 密钥中。您可以使用现有的密钥或创建一个新密钥。您可以在 Amazon ECR 控制台或 Secrets Manager 控制台中创建 Secrets Manager 密钥。要使用 Secrets Manager 控制台而不是 Amazon ECR 控制台创建 Secrets Manager 密钥，请参阅 [将您的上游存储库凭证存储在 AWS Secrets Manager 密钥中](#)。

先决条件

- 确认您拥有创建直通缓存规则的适当的 IAM 权限。有关信息，请参阅 [将上游注册表与 Amazon ECR 私有注册表同步所需的 IAM 权限](#)。
- 对于需要身份验证的 upstream 注册表：如果要使用现有密钥，请验证 Secrets Manager 密钥是否满足以下要求：
 - 秘密的名称以开头 `ecr-pullthroughcache/`。AWS Management Console 仅显示带 `ecr-pullthroughcache/` 前缀的 Secrets Manager 密钥。
 - 密钥所在的账户和区域必须与拉取缓存规则所在的账户和区域相匹配。

要创建缓存提取规则 (AWS Management Console)

以下步骤演示如何使用 Amazon ECR 控制台创建缓存提取规则和 Secrets Manager 密钥。要使用 Secrets Manager 控制台创建密钥，请参阅[将您的上游存储库凭证存储在 AWS Secrets Manager 密钥中](#)。

对于 Amazon ECR Public、Kubernetes 容器注册表或 Quay

1. 打开 Amazon ECR 控制台：<https://console.aws.amazon.com/ecr/>。
2. 从导航栏中，选择要配置私有注册表设置的区域。
3. 在导航窗格中，选择 Private registry (私有注册表)、Pull through cache (缓存提取)。
4. 在 Pull through cache configuration (缓存提取配置) 页面上，选择 Add rule (添加规则)。
5. 在步骤 1：指定来源页面上，对于注册表，请从上游注册表的列表中选择 Amazon ECR Public、Kubernetes 或 Quay，然后选择下一步。
6. 在步骤 2：指定目标页面上，对于 Amazon ECR 存储库前缀，请指定在缓存从源公有注册表中提取的映像时要使用的存储库命名空间前缀，然后选择下一步。默认情况下，已填充命名空间，但也可以指定自定义命名空间。
7. 在步骤 3：审核并创建页面上，审核缓存提取规则配置，然后选择创建。
8. 对要创建的每个缓存提取重复上一步骤。单独为每个区域创建了缓存提取规则。

对于 Docker Hub

1. 打开 Amazon ECR 控制台：<https://console.aws.amazon.com/ecr/>。
2. 从导航栏中，选择要配置私有注册表设置的区域。
3. 在导航窗格中，选择 Private registry (私有注册表)、Pull through cache (缓存提取)。
4. 在 Pull through cache configuration (缓存提取配置) 页面上，选择 Add rule (添加规则)。
5. 在步骤 1：指定来源页面上，对于注册表，选择 Docker Hub，然后选择下一步。
6. 在步骤 2：配置身份验证页面上，对于上游凭证，您必须以 AWS Secrets Manager 密钥存储 Docker Hub 的身份验证凭证。您可以指定现有密钥，也可以使用 Amazon ECR 控制台创建新密钥。
 - a. 要使用现有密钥，请选择使用现有 AWS 密钥。对于密钥名称，使用下拉列表选择现有的密钥，然后选择下一步。

Note

AWS Management Console 仅显示名称使用 `ecr-pullthroughcache/` 前缀的 Secrets Manager 密钥。密钥还必须与缓存提取规则位于相同的账户和区域。


- b. 要创建新密钥，请选择创建 AWS 密钥，执行以下操作，然后选择下一步。
 - i. 在密钥名称中，请为密钥指定一个描述性名称。密钥名称必须包含 1-512 个 Unicode 字符。
 - ii. 在 Docker Hub 用户名中，请指定 Docker Hub 用户名。
 - iii. 在 Docker Hub 访问令牌中，请指定 Docker Hub 访问令牌。有关如何创建 Docker Hub 访问令牌的更多信息，请参阅 Docker 文档中的 [Create and manage access tokens](#)。
7. 在步骤 3：指定目标页面上，对于 Amazon ECR 存储库前缀，请指定在缓存从源公有注册表中提取的映像时要使用的存储库命名空间，然后选择下一步。

默认情况下，已填充命名空间，但也可以指定自定义命名空间。

8. 在步骤 4：审核并创建页面上，审核缓存提取规则配置，然后选择创建。
9. 对要创建的每个缓存提取重复上一步骤。单独为每个区域创建了缓存提取规则。

对于 GitHub 容器注册表

1. 打开 Amazon ECR 控制台：<https://console.aws.amazon.com/ecr/>。
2. 从导航栏中，选择要配置私有注册表设置的区域。
3. 在导航窗格中，选择 Private registry (私有注册表)、Pull through cache (缓存提取)。
4. 在 Pull through cache configuration (缓存提取配置) 页面上，选择 Add rule (添加规则)。
5. 在“步骤 1：指定来源”页面上，对于“注册表”，选择“GitHub 容器注册表”，然后选择“下一步”。
6. 在“步骤 2：配置身份验证”页面上，对于上游凭证，您必须将 GitHub 容器注册表的身份验证凭据存储在 AWS Secrets Manager 密钥中。您可以指定现有密钥，也可以使用 Amazon ECR 控制台创建新密钥。
 - a. 要使用现有密钥，请选择使用现有 AWS 密钥。对于密钥名称，使用下拉列表选择现有的密钥，然后选择下一步。

 Note

AWS Management Console 仅显示名称使用 `ecr-pullthroughcache/` 前缀的 Secrets Manager 密钥。密钥还必须与缓存提取规则位于相同的账户和区域。


- b. 要创建新密钥，请选择创建 AWS 密钥，执行以下操作，然后选择下一步。
 - i. 在密钥名称中，请为密钥指定一个描述性名称。密钥名称必须包含 1-512 个 Unicode 字符。
 - ii. 对于 GitHub 容器注册表用户名，请指定您的 GitHub 容器注册表用户名。
 - iii. 对于 GitHub 容器注册表访问令牌，请指定您的 GitHub 容器注册表访问令牌。有关创建 GitHub 访问令牌的更多信息，请参阅 GitHub 文档中的[管理您的个人访问令牌](#)。
7. 在步骤 3：指定目标页面上，对于 Amazon ECR 存储库前缀，请指定在缓存从源公有注册表中提取的映像时要使用的存储库命名空间，然后选择下一步。

默认情况下，已填充命名空间，但也可以指定自定义命名空间。


8. 在步骤 4：审核并创建页面上，审核缓存提取规则配置，然后选择创建。
9. 对要创建的每个缓存提取重复上一步骤。单独为每个区域创建了缓存提取规则。

对于 Microsoft Azure 容器注册表

1. 打开 Amazon ECR 控制台：<https://console.aws.amazon.com/ecr/>。
2. 从导航栏中，选择要配置私有注册表设置的区域。
3. 在导航窗格中，选择 Private registry (私有注册表)、Pull through cache (缓存提取)。
4. 在 Pull through cache configuration (缓存提取配置) 页面上，选择 Add rule (添加规则)。
5. 在步骤 1：指定来源页面上，执行以下操作。
 - a. 在注册表中，请选择 Microsoft Azure 容器注册表
 - b. 在源注册表 URL 中，请指定 Microsoft Azure 容器注册表的名称，然后选择下一步。

 Important

您只需要指定前缀，因为系统已代表您填充 `.azurecr.io` 后缀。

6. 在步骤 2：配置身份验证页面上，对于上游凭证，您必须以 AWS Secrets Manager 密钥存储 Microsoft Azure 容器注册表的身份验证凭证。您可以指定现有密钥，也可以使用 Amazon ECR 控制台创建新密钥。
 - a. 要使用现有密钥，请选择使用现有 AWS 密钥。对于密钥名称，使用下拉列表选择现有的密钥，然后选择下一步。
-  **Note**

AWS Management Console 仅显示名称使用 `ecr-pullthroughcache/` 前缀的 Secrets Manager 密钥。密钥还必须与缓存提取规则位于相同的账户和区域。
- b. 要创建新密钥，请选择创建 AWS 密钥，执行以下操作，然后选择下一步。
 - i. 在密钥名称中，请为密钥指定一个描述性名称。密钥名称必须包含 1-512 个 Unicode 字符。
 - ii. 在 Microsoft Azure 容器注册表用户名中，请指定 Microsoft Azure 容器注册表用户名。
 - iii. 在 Microsoft Azure 容器注册表访问令牌中，请指定 Microsoft Azure 容器注册表访问令牌。有关如何创建 Microsoft Azure 容器注册表访问令牌的更多信息，请参阅 Microsoft Azure 文档中的 [创建令牌 - 门户](#)。
 7. 在步骤 3：指定目标页面上，对于 Amazon ECR 存储库前缀，请指定在缓存从源公有注册表中提取的映像时要使用的存储库命名空间，然后选择下一步。


默认情况下，已填充命名空间，但也可以指定自定义命名空间。

8. 在步骤 4：审核并创建页面上，审核缓存提取规则配置，然后选择创建。
9. 对要创建的每个缓存提取重复上一步骤。单独为每个区域创建了缓存提取规则。

对于 GitLab 容器注册表

1. 打开 Amazon ECR 控制台：<https://console.aws.amazon.com/ecr/>。
2. 从导航栏中，选择要配置私有注册表设置的区域。
3. 在导航窗格中，选择 Private registry (私有注册表)、Pull through cache (缓存提取)。
4. 在 Pull through cache configuration (缓存提取配置) 页面上，选择 Add rule (添加规则)。
5. 在“步骤 1：指定来源”页面上，对于“注册表”，选择“GitLab 容器注册表”，然后选择“下一步”。

6. 在“步骤 2：配置身份验证”页面上，对于上游凭证，您必须将 GitLab 容器注册表的身份验证凭据存储在 AWS Secrets Manager 密钥中。您可以指定现有密钥，也可以使用 Amazon ECR 控制台创建新密钥。
 - a. 要使用现有密钥，请选择使用现有 AWS 密钥。对于密钥名称，使用下拉列表选择现有的密钥，然后选择下一步。有关使用 Secrets Manager 控制台创建 Secrets Manager 密钥的更多信息，请参阅[将您的上游存储库凭证存储在 AWS Secrets Manager 密钥中](#)。

 Note

AWS Management Console 仅显示名称使用 `ecr-pullthroughcache/` 前缀的 Secrets Manager 密钥。密钥还必须与缓存提取规则位于相同的账户和区域。

- b. 要创建新密钥，请选择创建 AWS 密钥，执行以下操作，然后选择下一步。
 - i. 在密钥名称中，请为密钥指定一个描述性名称。密钥名称必须包含 1-512 个 Unicode 字符。
 - ii. 对于 GitLab 容器注册表用户名，请指定您的 GitLab 容器注册表用户名。
 - iii. 对于 GitLab 容器注册表访问令牌，请指定您的 GitLab 容器注册表访问令牌。有关创建 GitLab 容器注册表访问令牌的更多信息，请参阅 GitLab 文档中的[个人访问令牌、群组访问令牌或项目访问令牌](#)。
7. 在步骤 3：指定目标页面上，对于 Amazon ECR 存储库前缀，请指定在缓存从源公有注册表中提取的映像时要使用的存储库命名空间，然后选择下一步。

默认情况下，已填充命名空间，但也可以指定自定义命名空间。

8. 在步骤 4：审核并创建页面上，审核缓存提取规则配置，然后选择创建。
9. 对要创建的每个缓存提取重复上一步骤。单独为每个区域创建了缓存提取规则。

要创建缓存提取规则 (AWS CLI)

使用 [create-pull-through-cache-rule AWS CLI 命令为 Amazon ECR 私有注册表创建穿过缓存规则](#)。对于需要身份验证的上游注册表，您必须以 Secrets Manager 密钥存储凭证。要使用 Secrets Manager 控制台创建密钥，请参阅[将您的上游存储库凭证存储在 AWS Secrets Manager 密钥中](#)。

针对每个支持的上游注册表提供以下示例。

对于 Amazon ECR Public

以下示例为 Amazon ECR 公有注册表创建一个缓存提取规则。它指定了存储库前缀 `ecr-public`，这导致使用缓存提取规则创建的每个存储库都具有 `ecr-public/upstream-repository-name` 命名方案。

```
aws ecr create-pull-through-cache-rule \  
  --ecr-repository-prefix ecr-public \  
  --upstream-registry-url public.ecr.aws \  
  --region us-east-2
```

对于 Kubernetes 容器注册表

以下示例为 Kubernetes 公有注册表创建了一个缓存提取规则。它指定了存储库前缀 `kubernetes`，这导致使用缓存提取规则创建的每个存储库都具有 `kubernetes/upstream-repository-name` 命名方案。

```
aws ecr create-pull-through-cache-rule \  
  --ecr-repository-prefix kubernetes \  
  --upstream-registry-url registry.k8s.io \  
  --region us-east-2
```

对于 Quay

以下示例为 Quay 公有注册表创建了一个缓存提取规则。它指定了存储库前缀 `quay`，这导致使用推送缓存规则创建的每个存储库都具有命名方案 `quay/upstream-repository-name`。

```
aws ecr create-pull-through-cache-rule \  
  --ecr-repository-prefix quay \  
  --upstream-registry-url quay.io \  
  --region us-east-2
```

对于 Docker Hub

以下示例为 Docker Hub 注册表创建了一个缓存提取规则。它指定了存储库前缀 `docker-hub`，这导致使用缓存提取规则创建的每个存储库都具有 `docker-hub/upstream-repository-name` 命名方案。您必须指定包含 Docker Hub 凭证的密钥的完整 Amazon 资源名称 (ARN)。

```
aws ecr create-pull-through-cache-rule \  
  --ecr-repository-prefix docker-hub \  
  --upstream-registry-url registry-1.docker.io \  
  --region us-east-2
```

```
--credential-arn arn:aws:secretsmanager:us-east-2:111122223333:secret:ecr-  
pullthroughcache/example1234 \  
--region us-east-2
```

对于 GitHub 容器注册表

以下示例为 GitHub 容器注册表创建了通过缓存规则。它指定了存储库前缀 `docker-hub`，这导致使用缓存提取规则创建的每个存储库都具有 `github/upstream-repository-name` 命名方案。您必须指定包含您的 GitHub 容器注册凭证的密钥的完整 Amazon 资源名称 (ARN)。

```
aws ecr create-pull-through-cache-rule \  
--ecr-repository-prefix github \  
--upstream-registry-url ghcr.io \  
--credential-arn arn:aws:secretsmanager:us-east-2:111122223333:secret:ecr-  
pullthroughcache/example1234 \  
--region us-east-2
```

对于 Microsoft Azure 容器注册表

以下示例为 Microsoft Azure 容器注册表创建了拉取缓存规则。它指定了存储库前缀 `azure`，这导致使用缓存提取规则创建的每个存储库都具有 `azure/upstream-repository-name` 命名方案。您必须指定包含 Microsoft Azure 容器注册表凭证的密钥的完整 Amazon 资源名称 (ARN)。

```
aws ecr create-pull-through-cache-rule \  
--ecr-repository-prefix azure \  
--upstream-registry-url myregistry.azurecr.io \  
--credential-arn arn:aws:secretsmanager:us-east-2:111122223333:secret:ecr-  
pullthroughcache/example1234 \  
--region us-east-2
```

对于 GitLab 容器注册表

以下示例为 GitLab 容器注册表创建了通过缓存规则。它指定了存储库前缀 `gitlab`，这导致使用缓存提取规则创建的每个存储库都具有 `gitlab/upstream-repository-name` 命名方案。您必须指定包含您的 GitLab 容器注册凭证的密钥的完整 Amazon 资源名称 (ARN)。

```
aws ecr create-pull-through-cache-rule \  
--ecr-repository-prefix gitlab \  
--upstream-registry-url registry.gitlab.com \  
--credential-arn arn:aws:secretsmanager:us-east-2:111122223333:secret:ecr-  
pullthroughcache/example1234 \  
--region us-east-2
```

```
--region us-east-2
```

后续步骤

创建直通缓存规则后，接下来的步骤如下：

- 创建存储库创建模板。有了存储库创建模板，在为 Amazon ECR 在缓存提取操作期间代表您创建的新存储库定义设置时，您能拥有控制权。有关更多信息，请参阅 [用于控制在缓存拉取操作期间创建的存储库的模板](#)。
- 验证缓存提取规则。在验证缓存提取规则时，Amazon ECR 会与上游注册表建立网络连接，验证它是否可以访问包含上游注册表凭证的 Secrets Manager 密钥，以及身份验证是否成功。有关更多信息，请参阅 [验证 Amazon ECR 中的直通缓存规则](#)。
- 开始使用缓存提取规则。有关更多信息，请参阅 [在 Amazon ECR 中使用拉取缓存规则提取图像](#)。

用于控制在缓存拉取操作期间创建的存储库的模板

存储库创建模板功能是 Amazon ECR 的预览版，可能会发生变化。在此公开预览期间，AWS Management Console 只能使用来管理您的仓库创建模板。

使用 Amazon ECR 存储库创建模板来定义 Amazon ECR 在拉取缓存操作期间代表您创建的存储库的设置。存储库创建模板中的设置仅在存储库创建期间应用，对现有存储库或使用任何其他方法创建的存储库没有任何影响。

以下区域不支持存储库创建模板：

- 中国 (北京) (cn-north-1)
- 中国 (宁夏) (cn-northwest-1)
- AWS GovCloud (美国东部) (us-gov-east-1)
- AWS GovCloud (美国西部) (us-gov-west-1)

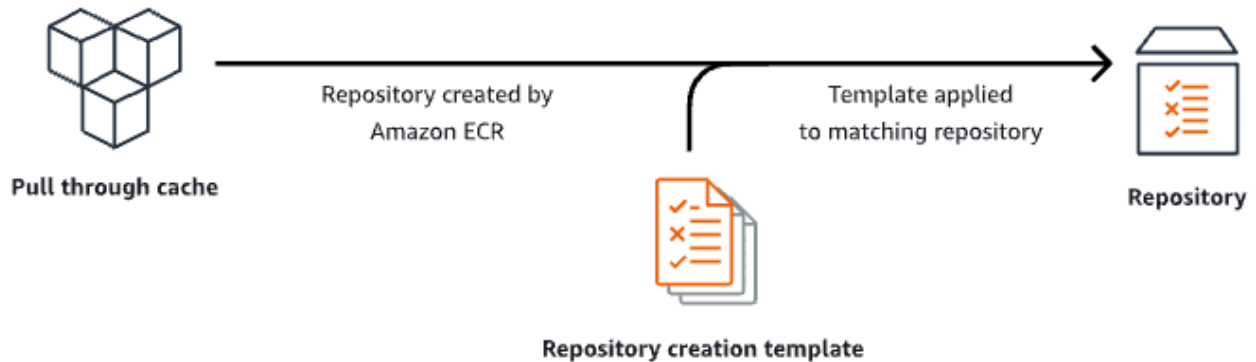
存储库创建模板的工作原理

有时，Amazon ECR 需要代表您创建新的私有存储库。例如，您首次使用缓存提取规则来检索上游存储库的内容并将其存储于 Amazon ECR 私有注册表时。当没有与您的缓存提取规则匹配的存储库

创建模板时，Amazon ECR 会为新存储库使用默认设置。这些默认设置包括关闭标签不变性、使用 AES-256 加密，以及不应用任何存储库或生命周期策略。

使用前缀与缓存提取规则匹配的存储库创建模板，您可以定义 Amazon ECR 为通过缓存提取操作创建的新存储库应用的设置。您可以为新存储库定义标签不变性、加密配置、存储库权限、生命周期策略和资源标签。

下图演示了在使用存储库创建模板时 Amazon ECR 使用的工作流程。



以下内容详细描述了存储库创建模板中的每个参数。

Prefix

Prefix 是与模板关联的存储库命名空间前缀。使用此前缀创建的所有存储库都将应用此模板中定义的设置。例如，前缀 `prod` 将应用于以 `prod/` 开头的所有存储库。同理，前缀 `prod/team` 将应用于以 `prod/team/` 开头的所有存储库。

要将某模板应用于注册表中没有关联创建模板的所有存储库，您可以使用 `ROOT` 作为前缀。

⚠ Important

前缀末尾始终应用假定的 `/`。如果您指定 `ecr-public` 为前缀，Amazon ECR 会将其视为 `ecr-public/`。使用缓存提取规则时，您在创建规则时指定的存储库前缀也应指定为存储库创建模板前缀。

描述

此模板描述可选，用于描述存储库创建模板的用途。

模板版本

要使用的存储库创建模板版本。目前，仅支持 TV1 模板版本。

配置版本

要使用的模板的存储库配置版本。每个模板都必须包括存储库配置。默认配置版本是 CV1，包含映像标签可变性、存储库策略和生命周期策略设置。

镜像标签可变性

使用模板创建的存储库要使用的标签可变性设置。如果省略此参数，将使用 MUTABLE 的默认设置，该设置将允许覆盖映像标签。对于通过缓存提取操作创建的存储库所使用的模板，建议使用此设置。这样可以确保当标签相同时，Amazon ECR 可以更新缓存的映像。

如果指定 IMMUTABLE，则存储库中的所有映像标签都将不可变，从而防止这些标签被覆盖。

加密配置

使用模板创建的存储库要使用的加密配置。

如果您使用 KMS 加密类型，则使用具有 AWS KMS 中存储的 AWS Key Management Service 密钥的服务器端加密来加密存储库的内容。当您使用 AWS KMS 加密数据时，可以使用 Amazon ECR 的默认 AWS 托管 AWS KMS 密钥，也可以指定您自己的 AWS KMS 密钥，该密钥是您已经创建的。有关更多信息，请参阅《亚马逊简单存储服务用户指南》中的[使用服务器端加密使用存储在 AWS Key Management Service \(SSE-KMS\) 中的密 AWS Key Management Service 钥保护数据](#)。

如果您使用 AES256 加密类型，Amazon ECR 将使用具有 Amazon S3 托管加密密钥的服务器端加密，从而使用 AES-256 加密算法对存储库中的映像进行加密。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的[使用采用 Amazon S3 托管加密密钥的服务器端加密 \(SSE-S3\) 保护数据](#)。

存储库权限

使用模板创建的存储库要应用的存储库策略。存储库策略使用基于资源的权限来控制对存储库的访问权限。基于资源的权限让您指定能够访问存储库的 IAM 用户或角色，以及这些用户或角色可以对该存储库执行的操作。默认情况下，只有创建仓库的 AWS 账户才有权访问仓库。您可以应用策略文档来授予或拒绝针对您的存储库的其他权限。有关更多信息，请参阅[Amazon ECR 中的私有存储库政策](#)。

存储库生命周期策略

使用模板创建的存储库要使用的生命周期策略。生命周期策略提供了对私有存储库中映像的生命周期管理的更多控制。生命周期策略是一组规则，其中的每个规则为 Amazon ECR 定义一个操作。

这提供了一种自动清理容器镜像的方法，例如根据使用期限或计数过期的镜像。有关更多信息，请参阅 [在 Amazon ECR 中使用生命周期策略自动清理图像](#)。

资源标签

资源标签是应用于存储库的元数据，旨在帮助您对其进行分类和整理。每个标签都包含定义的一个密钥和一个可选值。

创建存储库创建模板的 IAM 权限

IAM 主体需要具有以下权限，才能管理存储库创建模板。此这些权限必须通过基于身份的 IAM policy 授予。

- `ecr:CreateRepositoryCreationTemplate` – 授权创建存储库创建模板。
- `ecr>DeleteRepositoryCreationTemplate` – 授权删除存储库创建模板。
- `ecr:PutLifecyclePolicy` – 授权创建生命周期策略并将其应用于存储库。仅当存储库创建模板包含生命周期策略时，才需要此权限。
- `ecr:SetRepositoryPolicy` – 授权创建存储库的权限策略。仅当存储库创建模板包含存储库策略时，才需要此权限。
- `ecr:TagResource` – 授权将元数据标签添加到资源。仅当存储库创建模板包含资源标签时，才需要此权限。

在 Amazon ECR 中创建存储库创建模板

您可以创建存储库创建模板，定义 Amazon ECR 在缓存提取操作期间代表您创建的存储库要使用的设置。创建存储库创建模板后，在拉取缓存操作期间创建的所有新存储库都将应用这些设置。这不会对之前创建的存储库产生任何影响。

创建存储库创建模板 (AWS Management Console)

1. 打开 Amazon ECR 控制台：<https://console.aws.amazon.com/ecr/>。
2. 在导航栏中，选择存储库创建模板的创建区域。
3. 在导航窗格中，依次选择私有注册表、存储库创建模板。
4. 在存储库创建模板页面上，选择创建模板。
5. 在步骤 1：定义模板页面上，对于模板详细信息，选择特定前缀，将模板应用于特定的存储库命名空间前缀，或者选择 ECR 注册表中的任意前缀，将模板应用于与该区域中任何其他模板都不匹配的所有存储库。

- a. 如果选择特定前缀，则在前缀中指定要应用模板的存储库命名空间前缀。前缀末尾始终应用假定的 /。例如，前缀 prod 将应用于以 prod/ 开头的的所有存储库。同理，前缀 prod/team 将应用于以 prod/team/ 开头的的所有存储库。
 - b. 如果选择 ECR 注册表中的任意前缀，则前缀将设置为 ROOT。
6. 在模板描述中，为模板指定可选描述，然后选择下一步。
 7. 在步骤 2：添加存储库创建配置页面上，指定使用模板创建的存储库要应用的存储库设置配置。
 - a. 在映像标签可变性中，选择要使用的标签可变性设置。有关更多信息，请参阅 [防止在 Amazon ECR 中覆盖图片标签](#)。


选择可变后，映像标签可以被覆盖。对于通过缓存提取操作创建的存储库所使用的模板，建议使用此设置。这样可以确保当标签相同时，Amazon ECR 可以更新缓存的映像。

选择不可变后，将防止映像标签被覆盖。为存储库配置了不可变标签后，如果尝试推送某个映像但其标签在存储库中已存在，系统将返回 ImageTagAlreadyExistsException 错误。当存储库开启标签不可变性时，这会影响所有标签，您不能将某些标签配置为不可变，而将其其他标签配置为可变。

- b. 在加密配置中，选择要使用的加密设置。有关更多信息，请参阅 [静态加密](#)。

选择 AES-256 时，Amazon ECR 使用具有 Amazon Simple Storage Service 托管加密密钥的服务器端加密，从而使用 AES-256 加密算法对静态数据进行加密。此服务不会产生额外费用。

选择 AWS KMS 后，Amazon ECR 将使用服务器端加密，密钥存储在 AWS Key Management Service ()AWS KMS 中。当您使用 AWS KMS 加密数据时，您可以使用默认的 AWS 托管密钥（由 Amazon ECR 管理），也可以指定自己的 AWS KMS 密钥（称为客户托管密钥）。

 Note

存储库创建之后，存储库的加密设置无法更改。

- c. 在存储库权限中，请指定使用此模板创建的存储库要应用的存储库权限策略。您可以选择使用下拉菜单，为最常见的使用案例选择一个 JSON 示例。有关更多信息，请参阅 [Amazon ECR 中的私有存储库政策](#)。

- d. 在存储库生命周期策略中，指定使用此模板创建的存储库要应用的存储库生命周期策略。您可以选择使用下拉菜单，为最常见的使用案例选择一个 JSON 示例。有关更多信息，请参阅 [在 Amazon ECR 中使用生命周期策略自动清理图像](#)。
 - e. 对于存储库 AWS 标签，以键值对的形式指定要与使用此模板创建的存储库关联的元数据，然后选择下一步。有关更多信息，请参阅 [在 Amazon ECR 中标记私有存储库](#)。
8. 在步骤 3：审核并创建页面上，审核您为存储库创建模板指定的设置。选择编辑选项以进行更改。完成后，选择创建。

在 Amazon ECR 中删除存储库创建模板

使用完存储库创建模板后，您可以将其删除。删除存储库创建模板后，在缓存拉取操作期间创建的任何存储库都将应用默认设置。

删除存储库创建模板 (AWS Management Console)

1. 打开 Amazon ECR 控制台：<https://console.aws.amazon.com/ecr/>。
2. 在导航栏中，选择要删除的存储库创建模板所在的区域。
3. 在导航窗格中，依次选择私有注册表、存储库创建模板。
4. 在存储库创建模板页面上，选择要删除的存储库创建模板。
5. 在操作下拉菜单中，选择删除。

验证 Amazon ECR 中的直通缓存规则

创建直通缓存规则后，对于需要身份验证的上游注册表，您可以验证该规则是否正常运行。在验证拉取缓存规则时，Amazon ECR 会与上游注册表建立网络连接，验证它是否可以访问包含上游注册表证书的 Secrets Manager 密钥，并验证身份验证是否成功。

在开始使用直通缓存规则之前，请确认您拥有适当的 IAM 权限。有关更多信息，请参阅 [将上游注册表与 Amazon ECR 私有注册表同步所需的 IAM 权限](#)。

删除缓存提取规则 (AWS Management Console)

以下步骤演示如何使用 Amazon ECR 控制台验证缓存提取规则。

1. 打开 Amazon ECR 控制台：<https://console.aws.amazon.com/ecr/>。
2. 在导航栏中，选择包含要验证的缓存提取规则的区域。

3. 在导航窗格中，选择 Private registry (私有注册表)、Pull through cache (缓存提取)。
4. 在缓存提取配置页面上，选择要验证的缓存提取规则。然后，使用操作下拉菜单并选择查看详细信息。
5. 在缓存提取规则详细信息页面上，使用操作下拉菜单并选择验证身份验证。Amazon ECR 将显示带结果的横幅。
6. 对要验证的每个缓存提取规则重复上述步骤。

删除缓存提取规则 (AWS CLI)

[validate-pull-through-cache-rule](#) AWS CLI 命令用于验证 Amazon ECR 私有注册表的拉取缓存规则。

以下示例使用 `ecr-public` 命名空间前缀。将该值替换为要验证的缓存提取规则的前缀值。

```
aws ecr validate-pull-through-cache-rule \  
  --ecr-repository-prefix ecr-public \  
  --region us-east-2
```

在响应中，`isValid` 参数指示验证是否成功。如果为 `true`，则表示 Amazon ECR 能够访问上游注册表，并且身份验证成功。如果为 `false`，则表示存在问题且验证失败。`failure` 参数指示原因。

在 Amazon ECR 中使用拉取缓存规则提取图像

以下示例演示在使用缓存提取规则来提取映像时要使用的命令语法。如果您在使用拉取缓存规则拉取上游镜像时收到错误，请参阅 [对 Amazon ECR 中的直通缓存问题进行故障排除](#)，以查看最常见的错误以及如何解决这些错误。

在开始使用直通缓存规则之前，请确认您拥有适当的 IAM 权限。有关更多信息，请参阅 [将上游注册表与 Amazon ECR 私有注册表同步所需的 IAM 权限](#)。

Note

以下示例使用使用的默认 Amazon ECR 存储库命名空间值。AWS Management Console 确保您使用已配置的 Amazon ECR 私有存储库 URI。

对于 Amazon ECR Public

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/ecr-public/repository_name/  
image_name:tag
```

Kubernetes 容器注册表

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/kubernetes/repository_name/  
image_name:tag
```

Quay

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/quay/repository_name/  
image_name:tag
```

Docker Hub

对于 Docker Hub 官方映像：

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/docker-hub/  
library/image_name:tag
```

Note

对于 Docker Hub 官方映像，必须包含 `/library` 前缀。对于所有其他 Docker Hub 存储库，应省略 `/library` 前缀。

对于所有其他 Docker Hub 映像：

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/docker-hub/repository_name/  
image_name:tag
```

GitHub 容器注册表

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/github/repository_name/  
image_name:tag
```

Microsoft Azure 容器注册表

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/azure/repository_name/  
image_name:tag
```

GitLab 容器注册表

```
docker pull aws_account_id.dkr.ecr.region.amazonaws.com/gitlab/repository_name/  
image_name:tag
```

将您的上游存储库凭证存储在 AWS Secrets Manager 密钥中

为需要身份验证的上游存储库创建缓存提取规则时，您必须以 Secrets Manager 密钥存储凭证。使用 Secrets Manager 密钥可能会产生费用。有关更多信息，请参阅[AWS Secrets Manager 定价](#)。

以下过程引导您了解如何为每个支持的上游存储库创建 Secrets Manager 密钥。您可以选择在 Amazon ECR 控制台使用“创建缓存提取规则”工作流程来创建密钥，而不是使用 Secrets Manager 控制台来创建密钥。有关更多信息，请参阅 [在 Amazon ECR 中创建直通缓存规则](#)。

Docker Hub

为 Docker Hub 凭证创建 Secrets Manager 密钥 (AWS Management Console)

1. 在 <https://console.aws.amazon.com/secretsmanager/> 打开 Secrets Manager 控制台。
2. 选择 存储新密钥。
3. 在选择密钥类型页面上，执行以下操作：
 - a. 对于密钥类型，请选择其他密钥类型。
 - b. 在键/值对中，为 Docker Hub 凭证创建两行键/值对。您可以在密钥中存储最多 65536 个字节。
 - i. 对于第一个键/值对，指定 username 为键，指定 Docker Hub 用户名为值。
 - ii. 对于第二个键/值对，指定 accessToken 为键，指定 Docker Hub 访问令牌为值。有关如何创建 Docker Hub 访问令牌的更多信息，请参阅 Docker 文档中的 [Create and manage access tokens](#)。
 - c. 对于加密密钥，保留默认的 aws/secretsmanager AWS KMS key 值，然后选择下一步。使用此密钥不产生任何费用。有关更多信息，请参阅《AWS Secrets Manager 用户指南》中的 [Secrets Manager 中的密钥加密和解密](#)。

⚠ Important

您必须使用默认 `aws/secretsmanager` 加密密钥来加密密钥。Amazon ECR 不支持为此密钥使用客户自主管理型密钥 (CMK)。

4. 在配置密钥页面上，执行以下操作：

- a. 输入一个描述性的 Secret name (密钥名称) 和 Description (说明)。密钥名称必须包含 1-512 个 Unicode 字符，并且以 `ecr-pullthroughcache/` 为前缀。

⚠ Important

Amazon ECR AWS Management Console 仅显示名称使用 `ecr-pullthroughcache/` 前缀的 Secrets Manager 机密。

- b. (可选) 在标签部分中，在您的密钥中添加一个或多个标签。有关标记策略，请参阅《AWS Secrets Manager 用户指南》中的 [标记 Secrets Manager 密钥](#)。请不要将敏感信息存储在标签中，因为它们未加密。
 - c. (可选) 在资源权限，要将资源策略添加到您的密钥中，请选择编辑权限。有关更多信息，请参阅《AWS Secrets Manager 用户指南》中的 [将权限策略附加到 Secrets Manager 密钥](#)。
 - d. (可选) 在复制密钥中，要将您的密钥复制到另一个密钥 AWS 区域，请选择复制密钥。您可以现在复制密钥，也可以回头再复制。有关更多信息，请参阅《AWS Secrets Manager 用户指南》中的 [将密钥复制到其他区域](#)。
 - e. 选择下一步。
5. (可选) 在 Configure rotation (配置轮换) 页面上，您可以启用自动轮换。您也可以现在保持关闭轮换，然后稍后将其打开。有关更多信息，请参阅《AWS Secrets Manager 用户指南》中的 [轮换 Secrets Manager 密钥](#)。选择下一步。
 6. 在 Review (审核) 页上，审核您的密钥详细信息，然后选择 Store (存储)。

Secrets Manager 将返回到密钥列表。如果您的新密钥未显示，请选择 Refresh (刷新) 按钮。

GitHub Container Registry

为你的 GitHub 容器注册表凭证创建 Secrets Manager 密钥 (AWS Management Console)

1. 在 <https://console.aws.amazon.com/secretsmanager/> 打开 Secrets Manager 控制台。
2. 选择 存储新密钥。
3. 在选择密钥类型页面上，执行以下操作：
 - a. 对于密钥类型，请选择其他密钥类型。
 - b. 在键/值对中，为您的 GitHub 凭据创建两行。您可以在密钥中存储最多 65536 个字节。
 - i. 对于第一个键/值对，指定username为密钥，将您的 GitHub用户名指定为值。
 - ii. 对于第二个密钥/值对，指定accessToken为密钥，将 GitHub 访问令牌指定为值。
有关创建 GitHub 访问令牌的更多信息，请参阅 GitHub 文档中的[管理您的个人访问令牌](#)。
 - c. 对于加密密钥，保留默认的 aws/secretsmanager AWS KMS key 值，然后选择下一步。
使用此密钥不产生任何费用。有关更多信息，请参阅《AWS Secrets Manager 用户指南》中的 [Secrets Manager 中的密钥加密和解密](#)。

Important

您必须使用默认 aws/secretsmanager 加密密钥来加密密钥。Amazon ECR 不支持为此密钥使用客户自主管理型密钥 (CMK)。

4. 在 Configure secret (配置密钥) 页面上，执行以下操作：
 - a. 输入一个描述性的 Secret name (密钥名称) 和 Description (说明)。密钥名称必须包含 1-512 个 Unicode 字符，并且以 ecr-pullthroughcache/ 为前缀。

Important

Amazon ECR AWS Management Console 仅显示名称使用 ecr-pullthroughcache/ 前缀的 Secrets Manager 机密。

- b. (可选) 在标签部分中，在您的密钥中添加一个或多个标签。有关标记策略，请参阅《AWS Secrets Manager 用户指南》中的[标记 Secrets Manager 密钥](#)。请不要将敏感信息存储在标签中，因为它们未加密。

- c. (可选) 在资源权限, 要将资源策略添加到您的密钥中, 请选择编辑权限。有关更多信息, 请参阅《AWS Secrets Manager 用户指南》中的[将权限策略附加到 Secrets Manager 密钥](#)。
 - d. (可选) 在复制密钥中, 要将您的密钥复制到另一个密钥 AWS 区域, 请选择复制密钥。您可以现在复制密钥, 也可以回头再复制。有关更多信息, 请参阅《AWS Secrets Manager 用户指南》中的[将密钥复制到其他区域](#)。
 - e. 选择下一步。
5. (可选) 在 Configure rotation (配置轮换) 页面上, 您可以启用自动轮换。您也可以现在保持关闭轮换, 然后稍后将其打开。有关更多信息, 请参阅《AWS Secrets Manager 用户指南》中的[轮换 Secrets Manager 密钥](#)。选择下一步。
 6. 在 Review (审核) 页上, 审核您的密钥详细信息, 然后选择 Store (存储)。

Secrets Manager 将返回到密钥列表。如果您的新密钥未显示, 请选择 Refresh (刷新) 按钮。

Microsoft Azure Container Registry

为 Microsoft Azure 容器注册表凭证创建 Secrets Manager 密钥 (AWS Management Console)

1. 在 <https://console.aws.amazon.com/secretsmanager/> 打开 Secrets Manager 控制台。
2. 选择 存储新密钥。
3. 在选择密钥类型页面上, 执行以下操作:
 - a. 对于密钥类型, 请选择其他密钥类型。
 - b. 在键/值对中, 为 Microsoft Azure 凭证创建两行键/值对。您可以在密钥中存储最多 65536 个字节。
 - i. 对于第一个键/值对, 指定 username 为键, 指定 Microsoft Azure 容器注册表用户名为值。
 - ii. 对于第二个键/值对, 指定 accessToken 为键, Microsoft Azure 容器注册表访问令牌为值。有关如何创建 Microsoft Azure 访问令牌的更多信息, 请参阅 Microsoft Azure 文档中的[创建令牌 - 门户](#)。
 - c. 对于加密密钥, 保留默认的 aws/secretsmanager AWS KMS key 值, 然后选择下一步。使用此密钥不产生任何费用。有关更多信息, 请参阅《AWS Secrets Manager 用户指南》中的[Secrets Manager 中的密钥加密和解密](#)。

⚠ Important

您必须使用默认 `aws/secretsmanager` 加密密钥来加密密钥。Amazon ECR 不支持为此密钥使用客户自主管理型密钥 (CMK)。

4. 在 Configure secret (配置密钥) 页面上, 执行以下操作:
 - a. 输入一个描述性的 Secret name (密钥名称) 和 Description (说明)。密钥名称必须包含 1-512 个 Unicode 字符, 并且以 `ecr-pullthroughcache/` 为前缀。

⚠ Important

Amazon ECR AWS Management Console 仅显示名称使用 `ecr-pullthroughcache/` 前缀的 Secrets Manager 机密。


- b. (可选) 在标签部分中, 在您的密钥中添加一个或多个标签。有关标记策略, 请参阅《AWS Secrets Manager 用户指南》中的[标记 Secrets Manager 密钥](#)。请不要将敏感信息存储在标签中, 因为它们未加密。
 - c. (可选) 在资源权限, 要将资源策略添加到您的密钥中, 请选择编辑权限。有关更多信息, 请参阅《AWS Secrets Manager 用户指南》中的[将权限策略附加到 Secrets Manager 密钥](#)。
 - d. (可选) 在复制密钥中, 要将您的密钥复制到另一个密钥 AWS 区域, 请选择复制密钥。您可以现在复制密钥, 也可以回头再复制。有关更多信息, 请参阅《AWS Secrets Manager 用户指南》中的[将密钥复制到其他区域](#)。
 - e. 选择下一步。
 5. (可选) 在 Configure rotation (配置轮换) 页面上, 您可以启用自动轮换。您也可以现在保持关闭轮换, 然后稍后将其打开。有关更多信息, 请参阅《AWS Secrets Manager 用户指南》中的[轮换 Secrets Manager 密钥](#)。选择下一步。
6. 在 Review (审核) 页上, 审核您的密钥详细信息, 然后选择 Store (存储)。

Secrets Manager 将返回到密钥列表。如果您的新密钥未显示, 请选择 Refresh (刷新) 按钮。

GitLab Container Registry


为你的 GitLab 容器注册表凭证创建 Secrets Manager 密钥 (AWS Management Console)

1. 在 <https://console.aws.amazon.com/secretsmanager/> 打开 Secrets Manager 控制台。
2. 选择 存储新密钥。
3. 在选择密钥类型页面上，执行以下操作：
 - a. 对于密钥类型，请选择其他密钥类型。
 - b. 在键/值对中，为您的 GitLab 凭据创建两行。您可以在密钥中存储最多 65536 个字节。
 - i. 对于第一个键/值对，指定username为密钥，将您的 GitLab 容器注册表用户名指定为值。
 - ii. 对于第二个密钥/值对，指定accessToken为密钥，将您的 GitLab 容器注册表访问令牌指定为值。有关创建 GitLab 容器注册表访问令牌的更多信息，请参阅 GitLab 文档中的[个人访问令牌、群组访问令牌或项目访问令牌](#)。
 - c. 对于加密密钥，保留默认的 aws/secretsmanager AWS KMS key 值，然后选择下一步。使用此密钥不产生任何费用。有关更多信息，请参阅《AWS Secrets Manager 用户指南》中的 [Secrets Manager 中的密钥加密和解密](#)。

 Important

您必须使用默认 aws/secretsmanager 加密密钥来加密密钥。Amazon ECR 不支持为此密钥使用客户自主管理型密钥 (CMK)。

4. 在 Configure secret (配置密钥) 页面上，执行以下操作：
 - a. 输入一个描述性的 Secret name (密钥名称) 和 Description (说明)。密钥名称必须包含 1-512 个 Unicode 字符，并且以 ecr-pullthroughcache/ 为前缀。

 Important

Amazon ECR AWS Management Console 仅显示名称使用 ecr-pullthroughcache/ 前缀的 Secrets Manager 机密。

- b. (可选) 在标签部分中，在您的密钥中添加一个或多个标签。有关标记策略，请参阅《AWS Secrets Manager 用户指南》中的[标记 Secrets Manager 密钥](#)。请不要将敏感信息存储在标签中，因为它们未加密。

- c. (可选) 在资源权限, 要将资源策略添加到您的密钥中, 请选择编辑权限。有关更多信息, 请参阅《AWS Secrets Manager 用户指南》中的[将权限策略附加到 Secrets Manager 密钥](#)。
 - d. (可选) 在复制密钥中, 要将您的密钥复制到另一个密钥 AWS 区域, 请选择复制密钥。您可以现在复制密钥, 也可以回头再复制。有关更多信息, 请参阅《AWS Secrets Manager 用户指南》中的[将密钥复制到其他区域](#)。
 - e. 选择下一步。
5. (可选) 在 Configure rotation (配置轮换) 页面上, 您可以启用自动轮换。您也可以现在保持关闭轮换, 然后稍后将其打开。有关更多信息, 请参阅《AWS Secrets Manager 用户指南》中的[轮换 Secrets Manager 密钥](#)。选择下一步。
 6. 在 Review (审核) 页上, 审核您的密钥详细信息, 然后选择 Store (存储)。

Secrets Manager 将返回到密钥列表。如果您的新密钥未显示, 请选择 Refresh (刷新) 按钮。

对 Amazon ECR 中的直通缓存问题进行故障排除

使用拉取缓存规则拉取上游镜像时, 您可能会收到以下常见错误。

存储库不存在

指示存储库不存在的错误通常是由该存储库不存在于 Amazon ECR 私有注册表中或者未向拉取上游镜像的 IAM 主体授予 `ecr:CreateRepository` 权限所致。要解决此错误, 您应该验证拉取命令中的存储库 URI 正确, 已向拉取上游镜像的 IAM 主体授予所需的 IAM 权限, 或者在执行上游镜像拉取之前, 已在您的 Amazon ECR 私有注册表中创建要向其推送上游镜像的存储库。有关所需 IAM 权限的更多信息, 请参阅[将上游注册表与 Amazon ECR 私有注册表同步所需的 IAM 权限](#)。

以下是此错误的示例。

```
Error response from daemon: repository 111122223333.dkr.ecr.us-east-1.amazonaws.com/
ecr-public/amazonlinux/amazonlinux not found: name unknown: The repository with
name 'ecr-public/amazonlinux/amazonlinux' does not exist in the registry with id
'111122223333'
```

未找到请求的图片

指示找不到镜像的错误通常是由该镜像不存在于上游注册表中或者未向拉取上游镜像的 IAM 主体授予 `ecr:BatchImportUpstreamImage` 权限但已在 Amazon ECR 私有注册表中创建存储库所

致。要解决此错误，您应验证上游镜像和镜像标签名称正确且存在，并且已向拉取上游镜像的 IAM 主体授予所需的 IAM 权限。有关所需 IAM 权限的更多信息，请参阅 [将上游注册表与 Amazon ECR 私有注册表同步所需的 IAM 权限](#)。

以下是此错误的示例。

```
Error response from daemon: manifest for 111122223333.dkr.ecr.us-east-1.amazonaws.com/ecr-public/amazonlinux/amazonlinux:latest not found: manifest unknown: Requested image not found
```

403 从 Docker Hub 存储库中提取数据时禁止使用

从标记为 Docker 官方映像的 Docker Hub 存储库中提取时，您必须在使用的 URI 中包含 `/library/`。例如，`aws_account_id.dkr.ecr.region.amazonaws.com/docker-hub/library/image_name:tag`。如果您省略 Docker Hub 官方映像的 `/library/`，则当您尝试使用缓存提取规则来提取映像时，系统将返回 403 Forbidden 错误。有关更多信息，请参阅 [在 Amazon ECR 中使用拉取缓存规则提取图像](#)。

以下是此错误的示例。

```
Error response from daemon: failed to resolve reference "111122223333.dkr.ecr.us-west-2.amazonaws.com/docker-hub/amazonlinux:2023": pulling from host 111122223333.dkr.ecr.us-west-2.amazonaws.com failed with status code [manifests 2023]: 403 Forbidden
```

在 Amazon ECR 中复制私有镜像

您可以配置 Amazon ECR 私有注册表以支持存储库的复制。Amazon ECR 同时适用于跨区域和跨账户复制。要进行跨账户复制，目标账户必须配置注册表权限策略，以允许从源注册表进行复制。有关更多信息，请参阅 [Amazon ECR 中的私有注册表权限](#)。

主题

- [私有镜像复制的注意事项](#)
- [Amazon ECR 的私有镜像复制示例](#)
- [在 Amazon ECR 中配置私有镜像复制](#)

私有镜像复制的注意事项

使用私有镜像复制时应注意以下事项。

- 只有在配置复制之后，推送到存储库的内容才会被复制。存储库中任何先前存在的内容都不会复制。为存储库配置复制后，Amazon ECR 将保持目标和源同步。
- 复制完成后，跨区域和账户的存储库名称将保持不变。Amazon ECR 不支持在复制过程中更改存储库名称。
- 首次配置私有注册表以进行复制时，Amazon ECR 会代表您创建服务相关 IAM 角色。服务相关 IAM 角色授予 Amazon ECR 复制服务在注册表中创建存储库和复制镜像所需的权限。有关更多信息，请参阅 [对 Amazon ECR 使用服务相关角色](#)。
- 要进行跨账户复制，私有注册表目标必须授予允许源注册表复制其镜像的权限。通过设置私有注册表权限策略来完成此授权。有关更多信息，请参阅 [Amazon ECR 中的私有注册表权限](#)。
- 如果更改私有注册表的权限策略以删除权限，则以前授予权限的任何进行中复制都可能完成。
- 要进行跨区域复制，在该区域内或向该区域进行任何复制操作之前，源账户和目标账户都必须选择加入该区域。有关更多信息，请参阅《Amazon Web Services 一般参考》中的 [管理 AWS 区域](#)。
- 不支持在 AWS 分区之间进行跨区域复制。例如，us-west-2 中的存储库无法复制到 cn-north-1。有关 AWS 分区的更多信息，请参阅《AWS 一般参考》中的 [ARN 格式](#)。
- 私有注册表的复制配置最多可以包含 25 个跨所有规则的唯一目标，最多共有 10 个规则。每个规则最多可包含 100 个筛选条件。这允许为包含用于生产和测试的镜像的存储库指定单独的规则。
- 复制配置支持通过指定存储库前缀来筛选私有注册表中复制的存储库。有关示例，请参阅 [示例：使用存储库筛选条件配置跨区域复制](#)。

- 每次镜像推送只会执行一次复制操作。例如，如果您配置了从 us-west-2 到 us-east-1 以及从 us-east-1 到 us-east-2 的跨区域复制，则推送到 us-west-2 的镜像仅复制到 us-east-1，它不会再复制到 us-east-2。此行为同时适用于跨区域和跨账户复制。
- 大多数映像会在不到 30 分钟的时间内复制，但在极少数情况下，复制可能需要更长的时间。
- 注册表复制不执行任何删除操作。复制镜像和存储库不再使用时，可以手动删除它们。
- 存储库策略 (包括 IAM policy) 和生命周期策略不会被复制，而且除了对为其定义的存储库之外，不会产生任何影响。
- 不会复制存储库设置。预设情况下，在由于复制操作而创建的所有存储库上，标签不变性、镜像扫描和 KMS 加密设置都处于禁用状态。可以在创建存储库后更改标签不变性和镜像扫描设置。但是，该设置仅适用于设置更改后推送的镜像。
- 如果在存储库上启用了标签不变性，并且复制了与现有镜像使用相同标签的镜像，则该镜像将被复制，但不包含重复的标签。这可能会形成未标记的镜像。

Amazon ECR 的私有镜像复制示例

以下各示例演示了私有映像复制的常见应用场景。如果您使用配置复制 AWS CLI，则可以在创建 JSON 文件时使用 JSON 示例作为起点。如果您使用配置复制 AWS Management Console，则在查看并提交页面上查看复制规则时，您将看到类似的 JSON。

示例：配置跨区域复制到单个目标区域

下面显示了在单个注册表中配置跨区域复制的示例。此示例假定您的账户 ID 为 111122223333，并且您正在区域 (而不是 us-west-2) 中指定此复制配置。

```
{
  "rules": [
    {
      "destinations": [
        {
          "region": "us-west-2",
          "registryId": "111122223333"
        }
      ]
    }
  ]
}
```


示例：使用存储库筛选条件配置跨区域复制

下面显示了为与前缀名称值匹配的存储库配置跨区域复制的示例。此示例假定您的账户 ID 为 111122223333，您正在区域 (而不是 us-west-1) 中指定此复制配置，并且具有前缀为 prod 的存储库。

```
{
  "rules": [{
    "destinations": [{
      "region": "us-west-1",
      "registryId": "111122223333"
    }],
    "repositoryFilters": [{
      "filter": "prod",
      "filterType": "PREFIX_MATCH"
    }]
  }]
}
```

示例：配置跨区域复制到多个目标区域

下面显示了在单个注册表中配置跨区域复制的示例。此示例假定您的账户 ID 为 111122223333，并且您正在区域 (而不是 us-west-1 或 us-west-2) 中指定此复制配置。

```
{
  "rules": [
    {
      "destinations": [
        {
          "region": "us-west-1",
          "registryId": "111122223333"
        },
        {
          "region": "us-west-2",
          "registryId": "111122223333"
        }
      ]
    }
  ]
}
```

示例：配置跨账户复制

下面显示了为注册表配置跨账户复制的示例。此示例将配置复制到 444455556666 账户和 us-west-2 区域。

Important

要进行跨账户复制，目标账户必须配置注册表权限策略，以允许进行复制。有关更多信息，请参阅 [Amazon ECR 中的私有注册表权限](#)。

```
{
  "rules": [
    {
      "destinations": [
        {
          "region": "us-west-2",
          "registryId": "444455556666"
        }
      ]
    }
  ]
}
```

示例：在配置中指定多个规则

以下显示了配置注册表的多个复制规则的示例。此示例配置 111122223333 账户的复制，其具备一个规则，即将复制前缀为 prod 的存储库复制到 us-west-2 区域，并将带有前缀 test 的存储库复制到 us-east-2 区域。复制配置最多可以包含 10 个规则，每个规则最多指定 25 个目标。

```
{
  "rules": [{
    "destinations": [{
      "region": "us-west-2",
      "registryId": "111122223333"
    }],
    "repositoryFilters": [{
      "filter": "prod",
      "filterType": "PREFIX_MATCH"
    }]
  }]
```

```
},
{
  "destinations": [{
    "region": "us-east-2",
    "registryId": "111122223333"
  }],
  "repositoryFilters": [{
    "filter": "test",
    "filterType": "PREFIX_MATCH"
  }]
}
]
```

在 Amazon ECR 中配置私有镜像复制

为您的私有注册表配置每个区域的复制。您可以配置跨区域复制或跨账户复制。

有关常用复制的示例，请参阅 [Amazon ECR 的私有镜像复制示例](#)。

配置注册表复制设置 (AWS Management Console)

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择要配置注册表复制设置的区域。
3. 在导航窗格中，选择私有注册表。
4. 在私有注册表页面上的复制部分，选择编辑。
5. 在复制页面上，选择添加复制规则。
6. 在目标类型页面上，选择是启用跨区域复制、跨账户复制还是两者，然后选择下一步。
7. 如果启用了跨区域复制，则在配置目标区域中，选择一个或多个目标区域，然后选择下一步。
8. 如果启用了跨账户复制，则在跨账户复制中，选择注册表的跨账户复制设置。对于目标帐户，输入目标帐户的帐户 ID 以及复制到其中的一个或多个目标区域。选择目标帐户 + 以将其他帐户配置为复制目标。

Important

要进行跨账户复制，目标帐户必须配置注册表权限策略，以允许执行复制。有关更多信息，请参阅 [Amazon ECR 中的私有注册表权限](#)。

9. (可选) 在添加筛选条件页面上，为复制规则指定一个或多个筛选条件，然后选择添加。对要与复制操作相关联的每个筛选条件重复此步骤。必须将筛选条件指定为存储库名称前缀。如果未添加筛选条件，则复制所有存储库的内容。添加所有筛选条件后，选择下一步。
10. 在存储库的审核和提交页面上，查看复制规则配置，然后选择提交规则。

配置注册表复制设置 (AWS CLI)

1. 创建包含要为注册表定义的复制规则的 JSON 文件。复制配置最多可以包含 10 个规则，所有规则最多包含 25 个唯一目标，每个规则最多包含 100 个筛选条件。要在自己的账户中配置跨区域复制，请指定自己的账户 ID。有关更多示例，请参阅[Amazon ECR 的私有镜像复制示例](#)。

```
{
  "rules": [{
    "destinations": [{
      "region": "destination_region",
      "registryId": "destination_accountId"
    }],
    "repositoryFilters": [{
      "filter": "repository_prefix_name",
      "filterType": "PREFIX_MATCH"
    }]
  }]
}
```

2. 创建注册表的复制配置。

```
aws ecr put-replication-configuration \
  --replication-configuration file://replication-settings.json \
  --region us-west-2
```

3. 确认您的注册表设置。

```
aws ecr describe-registry \
  --region us-west-2
```

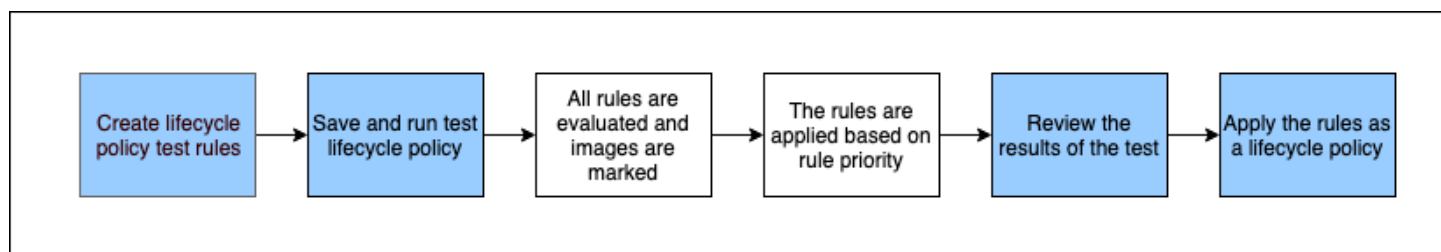
在 Amazon ECR 中使用生命周期策略自动清理图像

Amazon ECR 生命周期策略提供了对私有存储库中镜像的生命周期管理的更多控制。生命周期策略包含一条或多条规则，每条规则都定义了 Amazon ECR 的操作。根据生命周期策略中的过期标准，映像将根据年龄或计数在 24 小时内过期。当 Amazon ECR 根据生命周期策略执行操作时，此操作将作为事件捕获。AWS CloudTrail 有关更多信息，请参阅 [使用记录 Amazon ECR 操作 AWS CloudTrail](#)。

生命周期策略工作原理

生命周期策略由一条或多条规则组成，这些规则确定存储库中的哪些镜像应过期。在考虑使用生命周期策略时，务必使用生命周期策略预览来确认生命周期策略视为过期的镜像，然后再将其应用到存储库。将生命周期策略应用到存储库后，镜像将在达到到期标准后 24 小时内过期。当 Amazon ECR 基于生命周期策略执行操作时，这将在 AWS CloudTrail 中记录为一个事件。有关更多信息，请参阅 [使用记录 Amazon ECR 操作 AWS CloudTrail](#)。

下图显示了生命周期策略工作流程。



1. 创建一个或多个测试规则。
2. 保存测试规则并运行预览。
3. 生命周期策略评估程序遍历所有规则，并标记每个规则影响的镜像。
4. 然后，生命周期策略评估程序根据规则优先级应用规则，并显示存储库中的哪些镜像设置为过期。
5. 查看测试结果，确保标记为过期的镜像符合您预期的要求。
6. 将测试规则应用为存储库的生命周期策略。
7. 创建生命周期策略后，镜像将在达到过期标准后 24 小时内过期。

生命周期策略评估规则

生命周期策略评估器负责解析生命周期策略的明文 JSON，评估所有规则，然后根据规则优先级将这些规则应用于存储库中的镜像。下文更详细地解释了生命周期策略评估器采用的逻辑。有关示例，请参阅 [Amazon ECR 中的生命周期策略示例](#)。

- 无论规则优先级如何，评估器都会同时评估所有规则。评估所有规则后，将根据规则优先级应用这些规则。
- 镜像由一条或零条规则设为过期。
- 与规则的标记要求匹配的镜像不能被优先级较低的规则设为过期。
- 规则永远不能标记已由较高优先级规则标记的镜像，但仍然可以将其识别为未过期。
- 规则集必须包含一组唯一的标签前缀。
- 只允许一个规则选择未标记的镜像。
- 如果清单列表引用了映像，则在未先删除清单列表的情况下，该映像无法过期。
- 过期始终按 `pushed_at_time` 排序，并且始终是较早的镜像在较新的镜像之前过期。
- 生命周期策略规则可以指定 `tagPatternList` 或 `tagPrefixList`，但不能同时指定这两个选项。但是，一个生命周期策略可能包含多个规则，而不同的规则可能同时使用模式和前缀列表。
- 仅在 `tagStatus` 为 `tagged` 时才能使用 `tagPatternList` 或 `tagPrefixList` 参数。
- 使用时 `tagPatternList`，只要与通配符筛选条件匹配，即表示该映像成功匹配。例如，假设应用的筛选条件为 `prod*`，则将匹配名称以 `prod` 开头的存储库，例如 `prod`、`prod1` 或 `production-team1`。同样，如果应用的筛选条件为 `*prod*`，则将匹配名称包含 `prod` 的存储库，例如 `repo-production` 或 `prod-team`。

Important

每个字符串最多可以使用四个通配符 (*)。例如，`["*test*1*2*3", "test*1*2*3*"]` 有效，而 `["test*1*2*3*4*5*6"]` 无效。

- 使用 `tagPrefixList` 时，如果 `tagPrefixList` 值中的全部标签均与任何映像的标签匹配，则表示该映像成功匹配。
- 仅当 `countType` 为 `sinceImagePushed` 时，才能使用 `countUnit` 参数。
- 使用 `countType = imageCountMoreThan`，镜像基于 `pushed_at_time` 从最新到早排序，然后所有大于指定计数的镜像都将过期。
- 使用 `countType = sinceImagePushed`，其 `pushed_at_time` 早于指定天数 (基于 `countNumber`) 的所有镜像均会过期。

在 Amazon ECR 中创建生命周期策略预览

在应用生命周期策略之前，您可以使用生命周期策略预览来查看其对镜像存储库的影响。在将生命周期策略应用到存储库之前进行预览被认为是一种最佳实践。

Note

如果您使用 Amazon ECR 复制功能在不同的区域或账户中复制存储库，请注意，生命周期策略只能对创建该存储库的区域中的存储库执行操作。因此，如果您开启了复制功能，则建议在要将存储库复制到的每个区域和账户中创建一个生命周期策略。

创建生命周期策略预览 (AWS Management Console)

1. 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
2. 从导航栏中，选择包含要对其执行生命周期策略预览的存储库的区域。
3. 在导航窗格中的私有注册表下，选择存储库。
4. 在私有存储库页面上选择一个存储库，然后使用操作下拉列表选择生命周期策略。
5. 在该存储库的生命周期策略页面上，选择编辑测试规则、创建规则。
6. 为每个测试生命周期策略规则指定以下详细信息。
 - a. 对于规则优先级，输入该规则优先级的编号。规则优先级决定了生命周期策略规则的应用顺序。
 - b. 对于规则描述，输入该生命周期策略规则的说明。
 - c. 对于映像状态，选择已标记（通配符匹配）、已标记（前缀匹配）、未标记或任意。
 - d. 如果映像状态选择的是已标记（通配符匹配），则对于指定用于通配符匹配的标签，您可以指定带有通配符（*）的映像标签列表，以便根据您的生命周期策略执行操作。例如，假设您的映像标记为 prod、prod1、prod2 等，则可以指定 prod* 来对所有这些映像执行操作。如果指定多个标签，则仅选择具有所有指定标签的镜像。
7. 重复第 5-7 步以创建其他测试生命周期策略规则。

Important

每个字符串最多可以使用四个通配符（*）。例如，["*test*1*2*3", "test*1*2*3*"] 有效，而 ["test*1*2*3*4*5*6"] 无效。

- e. 如果映像状态选择的是已标记（前缀匹配），则对于指定用于前缀匹配的标签，您可以指定一个映像标签列表，以便根据您的生命周期策略执行操作。
- f. 对于匹配条件，选择自从推送映像以来或映像计数超过，然后指定一个值。
- g. 选择保存。

- 要运行生命周期策略预览，请选择保存并运行测试。
- 在测试生命周期规则的镜像匹配下方，查看生命周期策略预览的影响。
- 如果对预览结果满意，请选择应用为生命周期策略以创建具有指定规则的生命周期策略。应用生命周期策略后，受影响的镜像会在 24 小时内过期。
- 如果您对预览结果不满意，可以删除一个或多个测试生命周期规则，创建一个或多个规则来替换它们，然后再重复测试。

在 Amazon ECR 中为存储库创建生命周期策略

使用生命周期策略创建一组规则，使未使用的存储库映像过期。创建生命周期策略后，受影响的映像将在 24 小时内过期。

Note

如果您使用 Amazon ECR 复制功能在不同的区域或账户中复制存储库，请注意，生命周期策略只能对创建该存储库的区域中的存储库执行操作。因此，如果您开启了复制功能，则建议在要将存储库复制到的每个区域和账户中创建一个生命周期策略。

先决条件

最佳实践：创建生命周期策略预览，以验证生命周期策略规则过期的图像是否符合您的预期。有关说明，请参阅[在 Amazon ECR 中创建生命周期策略预览](#)。

创建生命周期策略 (AWS Management Console)

- 从 <https://console.aws.amazon.com/ecr/repositories> 打开 Amazon ECR 控制台。
- 从导航栏中，选择包含要对其创建生命周期策略的存储库的区域。
- 在导航窗格中的私有注册表下，选择存储库。
- 在私有存储库页面上选择一个存储库，然后使用操作下拉列表选择生命周期策略。
- 在该存储库的生命周期策略页面上，选择创建规则。
- 输入生命周期策略规则的以下详细信息。
 - 对于规则优先级，输入该规则优先级的编号。规则优先级决定了生命周期策略规则的应用顺序。

- b. 对于规则描述，输入该生命周期策略规则的说明。
- c. 对于映像状态，选择已标记（通配符匹配）、已标记（前缀匹配）、未标记或任意。
- d. 如果映像状态选择的是已标记（通配符匹配），则对于指定用于通配符匹配的标签，您可以指定带有通配符（*）的映像标签列表，以便根据您的生命周期策略执行操作。例如，假设您的映像标记为 prod、prod1、prod2 等，则可以指定 prod* 来对所有这些映像执行操作。如果指定多个标签，则仅选择具有所有指定标签的镜像。

⚠ Important

每个字符串最多可以使用四个通配符（*）。例如，["*test*1*2*3", "test*1*2*3*"] 有效，而 ["test*1*2*3*4*5*6"] 无效。

- e. 如果映像状态选择的是已标记（前缀匹配），则对于指定用于前缀匹配的标签，您可以指定一个映像标签列表，以便根据您的生命周期策略执行操作。
 - f. 对于匹配条件，选择自从推送映像以来或映像计数超过，然后指定一个值。
 - g. 选择保存。
7. 重复步骤 5-7 以创建其他生命周期策略规则。

创建生命周期策略 (AWS CLI)

1. 获取要为其创建生命周期策略的存储库的名称。

```
aws ecr describe-repositories
```

2. 创建名为 policy.json 的本地文件，其中包含生命周期策略的内容。有关生命周期策略示例，请参阅 [Amazon ECR 中的生命周期策略示例](#)。
3. 通过指定存储库名称并引用创建的生命周期策略 JSON 文件来创建生命周期策略。

```
aws ecr put-lifecycle-policy \  
  --repository-name repository-name \  
  --lifecycle-policy-text file://policy.json
```

Amazon ECR 中的生命周期策略示例

以下是显示语法的生命周期策略示例。

要查看有关策略属性的更多信息，请参阅[Amazon ECR 中的生命周期策略属性](#)。有关使用创建生命周期策略的说明 AWS CLI，请参阅[创建生命周期策略 \(AWS CLI\)](#)。

生命周期策略模板

生命周期策略的内容将在与存储库关联之前进行评估。以下是生命周期策略的 JSON 语法模板。

```
{
  "rules": [
    {
      "rulePriority": integer,
      "description": "string",
      "selection": {
        "tagStatus": "tagged"|"untagged"|"any",
        "tagPatternList": list<string>,
        "tagPrefixList": list<string>,
        "countType": "imageCountMoreThan"|"sinceImagePushed",
        "countUnit": "string",
        "countNumber": integer
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}
```

根据镜像使用期限筛选

以下示例演示了一个策略的生命周期策略语法，该策略使用 prod* 的 tagPatternList 来确保具有以 prod 开头的标签并且存在时间超过 14 天的映像会过期。

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Expire images older than 14 days",
      "selection": {
        "tagStatus": "tagged",
        "tagPatternList": ["prod*"],
        "countType": "sinceImagePushed",
        "countUnit": "days",

```

```
        "countNumber": 14
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}
```

根据镜像计数筛选

以下示例演示了一个策略的生命周期策略语法，该策略将仅保留一个未标记的映像并使所有其他映像过期。

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Keep only one untagged image, expire all others",
      "selection": {
        "tagStatus": "untagged",
        "countType": "imageCountMoreThan",
        "countNumber": 1
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}
```

根据多个规则筛选

以下示例在生命周期策略中使用多条规则。在此提供了一个示例存储库和生命周期策略以及结果的说明。

示例 A

存储库内容：

- 镜像 A，标签列表：["beta-1", "prod-1"]，已推送：10 天前

- 镜像 B，标签列表：["beta-2", "prod-2"]，已推送：9 天前
- 镜像 C，标签列表：["beta-3"]，已推送：8 天前

生命周期策略文本：

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Rule 1",
      "selection": {
        "tagStatus": "tagged",
        "tagPatternList": ["prod*"],
        "countType": "imageCountMoreThan",
        "countNumber": 1
      },
      "action": {
        "type": "expire"
      }
    },
    {
      "rulePriority": 2,
      "description": "Rule 2",
      "selection": {
        "tagStatus": "tagged",
        "tagPatternList": ["beta*"],
        "countType": "imageCountMoreThan",
        "countNumber": 1
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}
```

此生命周期策略的逻辑是：

- 规则 1 标识带有前缀 prod 标签的镜像。它应该从最早的镜像开始标记镜像，直到剩下一个或更少的匹配镜像。它将镜像 A 标记为过期。

- 规则 2 标识带有前缀 beta 标签的镜像。它应该从最早的镜像开始标记镜像，直到剩下一个或更少的匹配镜像。它将镜像 A 和镜像 B 标记为过期。但是，规则 1 已标记镜像 A，如果镜像 B 已过期，则会违反规则 1，因此跳过。
- 结果：镜像 A 已过期。

示例 B

这与前面的示例相同，但规则优先级顺序会更改以展示结果。

存储库内容：

- 镜像 A，标签列表：["beta-1", "prod-1"]，已推送：10 天前
- 镜像 B，标签列表：["beta-2", "prod-2"]，已推送：9 天前
- 镜像 C，标签列表：["beta-3"]，已推送：8 天前

生命周期策略文本：

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Rule 1",
      "selection": {
        "tagStatus": "tagged",
        "tagPatternList": ["beta*"],
        "countType": "imageCountMoreThan",
        "countNumber": 1
      },
      "action": {
        "type": "expire"
      }
    },
    {
      "rulePriority": 2,
      "description": "Rule 2",
      "selection": {
        "tagStatus": "tagged",
        "tagPatternList": ["prod*"],
        "countType": "imageCountMoreThan",
        "countNumber": 1
      }
    }
  ]
}
```

```
    },
    "action": {
      "type": "expire"
    }
  ]
}
```

此生命周期策略的逻辑是：

- 规则 1 标识带有前缀 beta 标签的镜像。它应该从最早的镜像开始标记镜像，直到剩下一个或更少的匹配镜像。它可以看到所有三个镜像，并将镜像 A 和镜像 B 标记为过期。
- 规则 2 标识带有前缀 prod 标签的镜像。它应该从最早的镜像开始标记镜像，直到剩下一个或更少的匹配镜像。它不会看到任何镜像，因为所有可用镜像已经被规则 1 标记，因此不会标记其他镜像。
- 结果：镜像 A 和 B 已过期。

在单个规则中筛选多个标签

以下示例为单个规则中的多个标签模式指定了生命周期策略语法。在此提供了一个示例存储库和生命周期策略以及结果的说明。

示例 A

在单个规则上指定多个标签模式时，映像必须与所有列出的标签模式匹配。

存储库内容：

- 镜像 A，标签列表：["alpha-1"]，已推送：12 天前
- 镜像 B，标签列表：["beta-1"]，已推送：11 天前
- 镜像 C，标签列表：["alpha-2", "beta-2"]，已推送：10 天前
- 镜像 D，标签列表：["alpha-3"]，已推送：4 天前
- 镜像 E，标签列表：["beta-3"]，已推送：3 天前
- 镜像 F，标签列表：["alpha-4", "beta-4"]，已推送：2 天前

```
{
  "rules": [
```

```

    {
      "rulePriority": 1,
      "description": "Rule 1",
      "selection": {
        "tagStatus": "tagged",
        "tagPatternList": ["alpha*", "beta*"],
        "countType": "sinceImagePushed",
        "countNumber": 5,
        "countUnit": "days"
      },
      "action": {
        "type": "expire"
      }
    }
  ]
}

```

此生命周期策略的逻辑是：

- 规则 1 用于识别标记有前缀 alpha 和 beta 的映像。它可以看到镜像 C 和 F。它应该标记超过五天的镜像，镜像 C 符合此条件。
- 结果：镜像 C 已过期。

示例 B

以下示例说明标签不是独占的。

存储库内容：

- 镜像 A，标签列表：["alpha-1", "beta-1", "gamma-1"]，已推送：10 天前
- 镜像 B，标签列表：["alpha-2", "beta-2"]，已推送：9 天前
- 镜像 C，标签列表：["alpha-3", "beta-3", "gamma-2"]，已推送：8 天前

```

{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Rule 1",
      "selection": {
        "tagStatus": "tagged",

```

```
        "tagPatternList": ["alpha*", "beta*"],
        "countType": "imageCountMoreThan",
        "countNumber": 1
    },
    "action": {
        "type": "expire"
    }
}
]
```

此生命周期策略的逻辑是：

- 规则 1 用于识别标记有前缀 alpha 和 beta 的映像。它可以看到所有的镜像。它应该从最早的镜像开始标记镜像，直到剩下一个或更少的匹配镜像。它将镜像 D 标记为过期。
- 结果：镜像 A 和 B 已过期。

筛选所有镜像

以下生命周期策略示例指定具有不同筛选条件的所有镜像。此处提供了一个示例存储库和生命周期策略以及结果的说明。

示例 A

下面显示了应用于所有规则但仅保留一个镜像并使所有其他镜像过期的策略的生命周期策略语法。

存储库内容：

- 镜像 A，标签列表：["alpha-1"]，已推送：4 天前
- 镜像 B，标签列表：["beta-1"]，已推送：3 天前
- 镜像 C，标签列表：[]，已推送：2 天前
- 镜像 D，标签列表：["alpha-2"]，已推送：1 天前

```
{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Rule 1",
      "selection": {
```



```

        "tagStatus": "any",
        "countType": "imageCountMoreThan",
        "countNumber": 1
    },
    "action": {
        "type": "expire"
    }
}
]
}

```

此生命周期策略的逻辑是：

- 规则 1 标识所有镜像。它可以看到镜像 A、B、C 和 D。它应该使最新镜像之外的所有其他镜像过期。它将镜像 A、B 和 C 标记为过期。
- 结果：镜像 A、B 和 C 已过期。

示例 B

以下示例说明了将所有规则类型组合在一个策略中的生命周期策略。

存储库内容：

- 镜像 A，标签列表：["alpha-1", "beta-1"]，已推送：4 天前
- 图片 B，标签列表：[]，已推送：3 天前
- 镜像 C，标签列表：["alpha-2"]，已推送：2 天前
- 镜像 D，标签列表：["git hash"]，已推送：1 天前
- 镜像 E，标签列表：[]，已推送：1 天前

```

{
  "rules": [
    {
      "rulePriority": 1,
      "description": "Rule 1",
      "selection": {
        "tagStatus": "tagged",
        "tagPatternList": ["alpha"],
        "countType": "imageCountMoreThan",
        "countNumber": 1
      }
    }
  ]
}

```

```
    },
    "action": {
      "type": "expire"
    }
  },
  {
    "rulePriority": 2,
    "description": "Rule 2",
    "selection": {
      "tagStatus": "untagged",
      "countType": "sinceImagePushed",
      "countUnit": "days",
      "countNumber": 1
    },
    "action": {
      "type": "expire"
    }
  },
  {
    "rulePriority": 3,
    "description": "Rule 3",
    "selection": {
      "tagStatus": "any",
      "countType": "imageCountMoreThan",
      "countNumber": 1
    },
    "action": {
      "type": "expire"
    }
  }
]
}
```

此生命周期策略的逻辑是：

- 规则 1 标识带有前缀 alpha 标签的镜像。它识别镜像 A 和 C。它应该保留最新镜像并将其余镜像标记为过期。它将镜像 A 标记为过期。
- 规则 2 标识未标记的镜像。它可以识别镜像 B 和 E。它应该将所有超过一天的镜像标记为过期。它将镜像 B 标记为过期。
- 规则 3 标识所有镜像。它识别镜像 A、B、C、D 和 E。它应该保留最新镜像并将其余镜像标记为过期。但是，它无法标记镜像 A、B、C 或 E，因为它们由优先级更高的规则识别。它将镜像 D 标记为过期。

- 结果：镜像 A、B 和 D 已过期。

Amazon ECR 中的生命周期策略属性

生命周期策略具有以下属性。

要查看生命周期策略的示例，请参阅[Amazon ECR 中的生命周期策略示例](#)。有关使用创建生命周期策略的说明 AWS CLI，请参阅[创建生命周期策略 \(AWS CLI\)](#)。

规则优先级

rulePriority

类型：整数

必需：是

设置应用规则的顺序，从低到高。首先应用优先级1为的生命周期策略规则，然后应用优先级2为的规则，依此类推。当您向某个生命周期策略添加规则时，必须为每个规则赋予一个唯一的 rulePriority 值。值不必在策略中的规则之间保持顺序。具有 tagStatus 值 any 的规则必须具有最大的 rulePriority 值并且最后被评估。

描述

description

类型：字符串

必需：否

(可选) 描述生命周期策略中规则的用途。

标签状态

tagStatus

类型：字符串

必需：是

确定要添加的生命周期策略规则是否为镜像指定标签。可接受的选项包括 `tagged`、`untagged` 或 `any`。如果您指定 `any`，则所有镜像都会根据它们评估规则。如果指定 `tagged`，还必须指定 `tagPrefixList` 值。如果指定 `untagged`，那么必须省略 `tagPrefixList`。

标签模式列表

`tagPatternList`

类型：`list[string]`

必填项：`tagStatus` 设置为“已标记”且未指定 `tagPrefixList` 时，是必填项

为已标记的映像创建生命周期策略时，最佳实践是使用 `tagPatternList` 来指定要过期的标签。您必须指定以逗号分隔的可能包含通配符（*）的映像标签模式列表，以便根据此列表执行生命周期策略操作。例如，假设映像标记为 `prod`、`prod1`、`prod2` 等，则可以使用标签模式列表 `prod*` 来指定所有这些映像。如果指定多个标签，则仅选择具有所有指定标签的镜像。

Important

每个字符串最多可以使用四个通配符（*）。例如，`["*test*1*2*3"`，`"test*1*2*3*"]` 有效，而 `["test*1*2*3*4*5*6"]` 无效。

标签前缀列表

`tagPrefixList`

类型：`list[string]`

必填项：`tagStatus` 设置为“已标记”且未指定 `tagPatternList` 时，是必填项

仅在您指定了 `"tagStatus": "tagged"` 但未指定 `tagPatternList` 时才使用。您必须指定以逗号分隔的镜像标签前缀列表，以便根据此列表执行生命周期策略操作。例如，如果您的镜像被标记为 `prod`、`prod1`、`prod2` 等，则可以使用标签前缀 `prod` 以指定所有这些标签。如果指定多个标签，则仅选择具有所有指定标签的镜像。

计数类型

countType

类型：字符串

必需：是

指定要应用于镜像的计数类型。

如果 `countType` 设置为 `imageCountMoreThan`，您还可以指定 `countNumber` 以创建一个规则，用于设置存储库中存在的镜像数量限制。如果 `countType` 设置为 `sinceImagePushed`，您还可以指定 `countUnit` 和 `countNumber`，以指定存储库中存在的镜像的时间限制。

计数单位

countUnit

类型：字符串

必需：是，仅当 `countType` 设置为 `sinceImagePushed` 时

指定计数单位 `days` 作为时间单位，除此之外，还指定 `countNumber` 表示天数。

只有在 `countType` 为 `sinceImagePushed` 时才能指定；如果您在 `countType` 是任何其他值时指定计数单位，将发生错误。

计数

countNumber

类型：整数

必需：是

指定计数数量。可接受的值为正整数 (0 不是可接受的值)。

如果使用的 `countType` 是 `imageCountMoreThan`，则该值为您希望在存储库中保留的镜像的最大数量。如果使用的 `countType` 是 `sinceImagePushed`，则该值为镜像的最大使用期限。

操作

type

类型：字符串。

必需：是

指定操作类型。支持的值为 `expire`。

Amazon Elastic Container Registry 中的安全性

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 AWS 的共同责任。[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 合规性计划](#) 的一部分，第三方审核人员将定期测试和验证安全性的有效性。要了解适用于 Amazon ECR 的合规性计划，请参阅 [合规性计划范围内的 AWS 服务](#)。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您的公司的要求以及适用的法律法规。

该文档帮助您了解如何在使用 Amazon ECR 时应用责任共担模式。以下主题说明如何配置 Amazon ECR 以实现您的安全性和合规性目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的 Amazon ECR 资源。

主题

- [适用于 Amazon Elastic Container Registry 的 Identity and Access Management](#)
- [Amazon ECR 中的数据保护](#)
- [Amazon Elastic Container Registry 的合规性验证](#)
- [Amazon Elastic Registry 中的基础设施安全性](#)
- [防止跨服务混淆座席](#)

适用于 Amazon Elastic Container Registry 的 Identity and Access Management

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以通过身份验证（登录）和授权（具有权限）使用 Amazon ECR 资源。您可以使用 IAM AWS 服务，无需支付额外费用。

主题

- [受众](#)
- [使用身份进行身份验证](#)

- [使用策略管理访问](#)
- [Amazon Elastic Container Registry 如何与 IAM 结合使用](#)
- [Amazon Elastic Container Registry 基于身份的策略示例](#)
- [使用基于标签的访问控制](#)
- [AWS Amazon 弹性容器注册表的托管策略](#)
- [对 Amazon ECR 使用服务相关角色](#)
- [排查 Amazon Elastic Container Registry 的身份和访问权限问题](#)

受众

您的使用方式 AWS Identity and Access Management (IAM) 会有所不同，具体取决于您在 Amazon ECR 中所做的工作。

服务用户 - 如果您使用 Amazon ECR 服务来完成工作，您的管理员会为您提供所需的凭证和权限。随着您使用更多 Amazon ECR 功能来完成工作，您可能需要额外权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法访问 Amazon ECR 中的功能，请参阅[排查 Amazon Elastic Container Registry 的身份和访问权限问题](#)。

服务管理员 - 如果您在公司负责管理 Amazon ECR 资源，您可能对 Amazon ECR 具有完全访问权限。您有责任确定您的服务用户应访问哪些 Amazon ECR 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要了解有关您的公司如何将 IAM 与 Amazon ECR 搭配使用的更多信息，请参阅[Amazon Elastic Container Registry 如何与 IAM 结合使用](#)。

IAM 管理员 - 如果您是 IAM 管理员，您可能希望了解如何编写策略以管理对 Amazon ECR 的访问的详细信息。要查看您可在 IAM 中使用的 Amazon ECR 基于身份的策略示例，请参阅[Amazon Elastic Container Registry 基于身份的策略示例](#)。

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份或通过担任 AWS 账户根用户担任 IAM 角色进行身份验证（登录 AWS）。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。AWS IAM Identity Center（IAM Identity Center）用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，您的管理员以前使用 IAM 角色设置了身份联合验证。当你使用联合访问 AWS 时，你就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅《AWS 登录 用户指南》中的[如何登录到您 AWS 账户](#)的。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用推荐的方法自行签署请求的更多信息，请参阅 IAM 用户指南中的[签署 AWS API 请求](#)。

无论使用何种身份验证方法，您可能需要提供其他安全信息。例如，AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《AWS IAM Identity Center 用户指南》中的[多重身份验证](#)和《IAM 用户指南》中的[在 AWS 中使用多重身份验证 \(MFA\)](#)。

AWS 账户 root 用户

创建时 AWS 账户，首先要有一个登录身份，该身份可以完全访问账户中的所有资源 AWS 服务和资源。此身份被称为 AWS 账户 root 用户，使用您创建帐户时使用的电子邮件地址和密码登录即可访问该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关需要您以根用户身份登录的任务的完整列表，请参阅 IAM 用户指南 中的[需要根用户凭证的任务](#)。

IAM 用户和群组

[IAM 用户](#)是您 AWS 账户 内部对个人或应用程序具有特定权限的身份。在可能的情况下，我们建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果您有一些特定的使用场景需要长期凭证以及 IAM 用户，建议您轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的[对于需要长期凭证的使用场景定期轮换访问密钥](#)。

[IAM 组](#)是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可能具有一个名为 IAMAdmins 的组，并为该组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅《IAM 用户指南》中的[何时创建 IAM 用户（而不是角色）](#)。

IAM 角色

[IAM 角色](#)是您内部具有特定权限 AWS 账户 的身份。它类似于 IAM 用户，但与特定人员不关联。您可以 AWS Management Console 通过[切换角色在中临时担任 IAM 角色](#)。您可以通过调用 AWS CLI 或 AWS API 操作或使用自定义 URL 来代入角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的[使用 IAM 角色](#)。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问 – 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关联合身份验证的角色的信息，请参阅《IAM 用户指南》中的 [为第三方身份提供商创建角色](#)。如果您使用 IAM Identity Center，则需要配置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的 [权限集](#)。
- 临时 IAM 用户权限 – IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取 – 您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解用于跨账户访问的角色和基于资源的策略之间的差别，请参阅《IAM 用户指南》中的 [IAM 角色与基于资源的策略有何不同](#)。
- 跨服务访问 — 有些 AWS 服务使用其他 AWS 服务服务中的功能。例如，当您在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Amazon S3 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
- 转发访问会话 (FAS) — 当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅[转发访问会话](#)。
- 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 [IAM 角色](#)。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的[创建向 AWS 服务委派权限的角色](#)。
- 服务相关角色-服务相关角色是一种与服务相关联的服务角色。AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 A@@ mazon EC2 上运行的应用程序 — 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这优先于在 EC2 实例中存储访问密钥。要向 EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建附加到该实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅《IAM 用户指南》中的 [使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限](#)。

要了解是使用 IAM 角色还是 IAM 用户，请参阅《IAM 用户指南》中的[何时创建 IAM 角色（而不是用户）](#)。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略是其中的一个对象 AWS ，当与身份或资源关联时，它会定义其权限。AWS 在委托人（用户、root 用户或角色会话）发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的结构和内容的更多信息，请参阅《IAM 用户指南》中的 [JSON 策略概览](#)。

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

IAM 策略定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或 AWS API 获取角色信息。

基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的 [创建 IAM 策略](#)。

基于身份的策略可以进一步归类为内联策略或托管式策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管式策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的 [在托管式策略与内联策略之间进行选择](#)。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Simple Storage Service（Amazon S3）存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中 [指定主体](#)。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

其他策略类型

AWS 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- **权限边界** - 权限边界是一个高级功能，用于设置基于身份的策略可以为 IAM 实体 (IAM 用户或角色) 授予的最大权限。您可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 Principal 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅《IAM 用户指南》中的 [IAM 实体的权限边界](#)。
- **服务控制策略 (SCP)**-SCP 是 JSON 策略，用于指定组织或组织单位 (OU) 的最大权限。AWS Organizations AWS Organizations 是一项用于对您的企业拥有的多 AWS 账户 项进行分组和集中管理的 服务。如果在组织内启用了所有功能，则可对任意或全部账户应用服务控制策略 (SCP)。SCP 限制成员账户中的实体 (包括每个 AWS 账户根用户实体) 的权限。有关 Organizations 和 SCP 的更多信息，请参阅《AWS Organizations 用户指南》中的 [SCP 的工作原理](#)。
- **会话策略** - 会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅《IAM 用户指南》中的 [会话策略](#)。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的 [策略评估逻辑](#)。

Amazon Elastic Container Registry 如何与 IAM 结合使用

在使用 IAM 管理对 Amazon ECR 的访问权限之前，您应该了解哪些 IAM 功能可用于 Amazon ECR。要全面了解 Amazon ECR 和其他 AWS 服务如何与 IAM 配合使用，请参阅 IAM 用户指南中的与 IAM 配合使用的AWS [服务](#)。

主题

- [Amazon ECR 基于身份的策略](#)
- [Amazon ECR 基于资源的策略](#)
- [基于 Amazon ECR 标签的授权](#)
- [Amazon ECR IAM 角色](#)

Amazon ECR 基于身份的策略

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。Amazon ECR 支持特定的操作、资源和条件键。要了解在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的 [IAM JSON 策略元素参考](#)。

操作

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 Action 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

Amazon ECR 中的策略操作在操作前面使用以下前缀：`ecr:`。例如，要授予某人使用 Amazon ECR `CreateRepository` API 操作创建 Amazon ECR 存储库的权限，您应将 `ecr:CreateRepository` 操作纳入其策略中。策略语句必须包含 Action 或 NotAction 元素。Amazon ECR 定义了一组自己的操作，以描述您可以使用该服务执行的任务。

要在单个语句中指定多项操作，请使用逗号将它们隔开，如下所示：

```
"Action": [  
    "ecr:action1",  
    "ecr:action2"
```

您也可以使用通配符 (*) 指定多个操作。例如，要指定以单词 `Describe` 开头的所有操作，包括以下操作：

```
"Action": "ecr:Describe*"
```

要查看 Amazon ECR 操作的列表，请参阅 IAM 用户指南中的 [Amazon Elastic Container Registry 的操作、资源和条件键](#)。

资源

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 Resource 或 NotResource 元素。作为最佳实践，请使用其 [Amazon 资源名称 \(ARN \)](#) 指定资源。对于支持特定资源类型 (称为资源级权限) 的操作，您可以执行此操作。

对于不支持资源级权限的操作 (如列出操作) ，请使用通配符 (*) 指示语句应用于所有资源。

```
"Resource": "*"
```

Amazon ECR 存储库资源具有以下 ARN ：

```
arn:${Partition}:ecr:${Region}:${Account}:repository/${Repository-name}
```

有关 ARN 格式的更多信息，请参阅 [Amazon 资源名称 \(ARN\)](#) 和 [AWS 服务命名空间](#)。

例如，要在语句中指定 us-east-1 区域中的 my-repo 存储库，请使用以下 ARN ：

```
"Resource": "arn:aws:ecr:us-east-1:123456789012:repository/my-repo"
```

要指定属于特定账户的所有存储库，请使用通配符 (*) ：

```
"Resource": "arn:aws:ecr:us-east-1:123456789012:repository/*"
```

要在单个语句中指定多个资源，请使用逗号分隔 ARN。

```
"Resource": [  
    "resource1",  
    "resource2"
```

要查看 Amazon ECR 资源类型及其 ARN 的列表，请参阅 IAM 用户指南中的 [Amazon Elastic Container Registry 定义的资源](#)。要了解您可以使用哪些操作指定每个资源的 ARN，请参阅 [Amazon Elastic Container Registry 定义的操作](#)。

条件键

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 Condition 元素 (或 Condition 块) 中，可以指定语句生效的条件。Condition 元素是可选的。您可以创建使用 [条件运算符](#) (例如，等于或小于) 的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则使用逻辑 OR 运算来 AWS 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的 [IAM 策略元素：变量和标签](#)。

AWS 支持全局条件密钥和特定于服务的条件密钥。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的 [AWS 全局条件上下文密钥](#)。

Amazon ECR 定义了自己的一组条件键，还支持使用一些全局条件键。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的 [AWS 全局条件上下文密钥](#)。

大多数 Amazon ECR 操作都支持 `aws:ResourceTag` 和 `ecr:ResourceTag` 条件键。有关更多信息，请参阅 [使用基于标签的访问控制](#)。

要查看 Amazon ECR 条件键的列表，请参阅 IAM 用户指南中的 [Amazon Elastic Container Registry 定义的条件键](#)。要了解您可以对哪些操作和资源使用条件键，请参阅 [Amazon Elastic Container Registry 定义的操作](#)。

示例

要查看 Amazon ECR 基于身份的策略的示例，请参阅 [Amazon Elastic Container Registry 基于身份的策略示例](#)。

Amazon ECR 基于资源的策略

基于资源的策略是 JSON 策略文档，它们指定了指定委托人可在 Amazon ECR 资源上执行的操作以及在什么条件下可执行。Amazon ECR 支持针对 Amazon ECR 存储库的基于资源的权限策略。基于资源的策略允许您基于资源向其他账户授予使用权限。您也可以使用基于资源的策略以允许 AWS 服务访问您的 Amazon ECR 存储库。

要启用跨账户访问，您可以将整个账户或其他账户中的 IAM 实体指定为 [基于资源的策略中的委托人](#)。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当委托人和资源位于不同的 AWS 账户中时，您还必须向委托人实体授予访问资源的权限。通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅《IAM 用户指南》中的 [IAM 角色与基于资源的策略有何不同](#)。

Amazon ECR 服务仅支持一种类型的基于资源的策略（称为容器策略），这种策略附加到存储库。这个策略定义哪些委托人实体（账户、用户、角色和联合身份用户）可以在存储库上执行操作。要了解如何将基于资源的策略附加到存储库，请参阅[Amazon ECR 中的私有存储库政策](#)。

Note

在 Amazon ECR 存储库策略中，策略元素 Sid 支持 IAM policy 所不支持的其他字符和间距。

示例

要查看 Amazon ECR 基于资源的策略的示例，请参阅 [Amazon ECR 中的私有存储库策略示例](#)，

基于 Amazon ECR 标签的授权

您可以将标签附加到 Amazon ECR 资源，或者在请求中将标签传递给 Amazon ECR。要基于标签控制访问，您需要使用 `ecr:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的[条件元素](#)中提供标签信息。有关标记 Amazon ECR 资源的更多信息，请参阅 [在 Amazon ECR 中标记私有存储库](#)。

要查看基于身份的策略（用于根据资源上的标签来限制对该资源的访问）的示例，请参阅[使用基于标签的访问控制](#)。

Amazon ECR IAM 角色

[IAM 角色](#) 是您的 AWS 账户中具有特定权限的实体。

将临时凭证用于 Amazon ECR

可以使用临时凭证进行联合身份验证登录，分派 IAM 角色或分派跨账户角色。您可以通过调用 AWS STS API 操作（例如[AssumeRole](#)或[GetFederation令牌](#)）来获取临时安全证书。

Amazon ECR 支持使用临时凭证。

服务相关角色

[服务相关角色](#) 允许 AWS 服务访问其他服务中的资源以代表您完成操作。服务相关角色显示在 IAM 账户中，并归该服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

Amazon ECR 支持服务相关角色。有关更多信息，请参阅 [对 Amazon ECR 使用服务相关角色](#)。

Amazon Elastic Container Registry 基于身份的策略示例

默认情况下，用户和角色没有创建或修改 Amazon ECR 资源的权限。他们也无法使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS API 执行任务。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅 IAM 用户指南中的 [创建 IAM 策略](#)。

有关 Amazon ECR 定义的操作和资源类型的详细信息，包括每种资源类型的 ARN 格式，请参阅服务授权参考中的 [Amazon Elastic Container Registry 的操作、资源和条件键](#)。

要了解如何使用这些示例 JSON 策略文档创建 IAM 基于身份的策略，请参阅《IAM 用户指南》中的 [在 JSON 选项卡上创建策略](#)。

主题

- [策略最佳实践](#)
- [使用 Amazon ECR 控制台](#)
- [允许用户查看他们自己的权限](#)
- [访问一个 Amazon ECR 存储库](#)

策略最佳实践

基于身份的策略确定某个人是否可以创建、访问或删除您账户中的 Amazon ECR 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下准则和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的 AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 [AWS 托管策略](#) 或 [工作职能的 AWS 托管策略](#)。
- 应用最低权限 – 在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的策略和权限](#)。
- 使用 IAM 策略中的条件进一步限制访问权限 – 您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使

用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 AWS CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 [IAM JSON 策略元素：条件](#)。

- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言 (JSON) 和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的 [IAM Access Analyzer 策略验证](#)。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的 [配置受 MFA 保护的 API 访问](#)。

有关 IAM 中的最佳实践的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的安全最佳实践](#)。

使用 Amazon ECR 控制台

要访问 Amazon Elastic Container Registry 控制台，您必须具有一组最低的权限。这些权限必须允许您列出和查看有关您 AWS 账户中的 Amazon ECR 资源的详细信息。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

为确保这些实体仍然可以使用 Amazon ECR 控制台，请向实体添加 AmazonEC2ContainerRegistryReadOnly AWS 托管策略。有关更多信息，请参阅 IAM 用户指南中的 [为用户添加权限](#)：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:GetRepositoryPolicy",
        "ecr:DescribeRepositories",
        "ecr:ListImages",
        "ecr:DescribeImages",
        "ecr:BatchGetImage",
        "ecr:GetLifecyclePolicy",
        "ecr:GetLifecyclePolicyPreview",
        "ecr:ListTagsForResource",
        "ecr:DescribeImageScanFindings"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "*"
  }
]
}

```

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与您尝试执行的 API 操作相匹配的操作。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}

```

```
    }  
  ]  
}
```

访问一个 Amazon ECR 存储库

在本示例中，您想向 AWS 账户中的用户授予访问您的 Amazon ECR 存储库的权限。my-repo 您还希望允许用户推送、提取和列出镜像。

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "ListImagesInRepository",  
      "Effect": "Allow",  
      "Action": [  
        "ecr:ListImages"  
      ],  
      "Resource": "arn:aws:ecr:us-east-1:123456789012:repository/my-repo"  
    },  
    {  
      "Sid": "GetAuthorizationToken",  
      "Effect": "Allow",  
      "Action": [  
        "ecr:GetAuthorizationToken"  
      ],  
      "Resource": "*"   
    },  
    {  
      "Sid": "ManageRepositoryContents",  
      "Effect": "Allow",  
      "Action": [  
        "ecr:BatchCheckLayerAvailability",  
        "ecr:GetDownloadUrlForLayer",  
        "ecr:GetRepositoryPolicy",  
        "ecr:DescribeRepositories",  
        "ecr:ListImages",  
        "ecr:DescribeImages",  
        "ecr:BatchGetImage",  
        "ecr:InitiateLayerUpload",  
        "ecr:UploadLayerPart",  
        "ecr:CompleteLayerUpload",  
        "ecr:PutImage"  
      ]  
    }  
  ]  
}
```

```
    ],
    "Resource": "arn:aws:ecr:us-east-1:123456789012:repository/my-repo"
  }
]
}
```

使用基于标签的访问控制

Amazon ECR CreateRepository API 操作允许您在创建存储库时指定标签。有关更多信息，请参阅 [在 Amazon ECR 中标记私有存储库](#)。

要使用户能够在创建存储桶时标记存储桶，用户必须有权使用创建资源的操作（例如，`ecr:CreateRepository`）。如果在资源创建操作中指定了标签，则 Amazon 会对 `ecr:CreateRepository` 操作执行额外的授权，以验证用户是否具备创建标签的权限。

您可以通过 IAM policy 来使用基于标签的访问控制。示例如下。

以下策略仅允许用户创建存储库或将其标记为 `key=environment,value=dev`。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateTaggedRepository",
      "Effect": "Allow",
      "Action": [
        "ecr:CreateRepository"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/environment": "dev"
        }
      }
    },
    {
      "Sid": "AllowTagRepository",
      "Effect": "Allow",
      "Action": [
        "ecr:TagResource"
      ],
      "Resource": "*",
      "Condition": {
```

```

        "StringEquals": {
            "aws:RequestTag/environment": "dev"
        }
    }
}
]
}

```

以下策略允许用户访问所有存储库（除非这些存储库标记为 `key=environment,value=prod`）。

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "ecr:*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": "ecr:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ecr:ResourceTag/environment": "prod"
        }
      }
    }
  ]
}

```

AWS Amazon 弹性容器注册表的托管策略

AWS 托管策略是由创建和管理的独立策略 AWS。AWS 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住，AWS 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 AWS 客户使用。我们建议通过定义特定于您的使用场景的[客户管理型策略](#)来进一步减少权限。

您无法更改 AWS 托管策略中定义的权限。如果 AWS 更新 AWS 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。AWS 最有可能在启动新的 API 或现有服务可以使用新 AWS 服务的 API 操作时更新 AWS 托管策略。

有关更多信息，请参阅《IAM 用户指南》中的 [AWS 托管策略](#)。

Amazon ECR 提供了多个托管策略，您可以将这些策略附加到 IAM 身份或 Amazon EC2 实例。这些托管策略允许对访问 Amazon ECR 资源和 API 操作进行不同级别的控制。有关这些策略中提到的每个 API 操作的更多信息，请参阅 Amazon Elastic Container Registry API 参考中的 [操作](#)。

主题

- [AmazonEC2ContainerRegistryFullAccess](#)
- [AmazonEC2ContainerRegistryPowerUser](#)
- [AmazonEC2ContainerRegistryReadOnly](#)
- [AWSECRPullThroughCache_ServiceRolePolicy](#)
- [ECRReplicationServiceRolePolicy](#)
- [Amazon ECR 更新了托 AWS 管政策](#)

AmazonEC2ContainerRegistryFullAccess

您可以将 AmazonEC2ContainerRegistryFullAccess 策略附加到 IAM 身份。

您可以使用此托管策略作为起点，根据您的具体要求创建自己的 IAM policy。例如，您可以创建一个策略，专门为用户或角色提供完全管理员访问权限，以管理 Amazon ECR 的使用。借助 [Amazon ECR 生命周期策略](#) 功能，您可以指定存储库中镜像的生命周期管理。生命周期策略事件作为 CloudTrail 事件报告。Amazon ECR 已与集成 AWS CloudTrail，因此它可以直接在 Amazon ECR 控制台中显示您的生命周期策略事件。AmazonEC2ContainerRegistryFullAccess 托管 IAM policy 包含促进此行为的 `cloudtrail:LookupEvents` 权限。

权限详细信息

此策略包含以下权限：

- `ecr` - 允许委托人完全访问所有 Amazon ECR API。
- `cloudtrail`— 允许委托人查找由 CloudTrail 捕获的管理事件或 AWS CloudTrail Insights 事件。

AmazonEC2ContainerRegistryFullAccess 策略如下所示。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```

    "Effect": "Allow",
    "Action": [
        "ecr:*",
        "cloudtrail:LookupEvents"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceLinkedRole"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "iam:AWSServiceName": [
                "replication.ecr.amazonaws.com"
            ]
        }
    }
}
]
}

```

AmazonEC2ContainerRegistryPowerUser

您可以将 AmazonEC2ContainerRegistryPowerUser 策略附加到 IAM 身份。

此策略授予管理权限，以允许 IAM 用户读写存储库，但不允许他们删除存储库或更改应用于存储库的策略文档。

权限详细信息

此策略包含以下权限：

- ecr - 允许委托人读取和写入存储库，以及读取生命周期策略。委托人不会被授予删除存储库或更改应用于它们的生命周期策略的权限。

AmazonEC2ContainerRegistryPowerUser 策略如下所示。

```

{
    "Version": "2012-10-17",

```



```
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ecr:GetAuthorizationToken",
      "ecr:BatchCheckLayerAvailability",
      "ecr:GetDownloadUrlForLayer",
      "ecr:GetRepositoryPolicy",
      "ecr:DescribeRepositories",
      "ecr:ListImages",
      "ecr:DescribeImages",
      "ecr:BatchGetImage",
      "ecr:GetLifecyclePolicy",
      "ecr:GetLifecyclePolicyPreview",
      "ecr:ListTagsForResource",
      "ecr:DescribeImageScanFindings",
      "ecr:InitiateLayerUpload",
      "ecr:UploadLayerPart",
      "ecr:CompleteLayerUpload",
      "ecr:PutImage"
    ],
    "Resource": "*"
  }
]
```

AmazonEC2ContainerRegistryReadOnly

您可以将 AmazonEC2ContainerRegistryReadOnly 策略附加到 IAM 身份。

此策略授予 Amazon ECR 的只读权限。这包括列出存储库及其中镜像的功能。它还包括使用 Docker CLI 从 Amazon ECR 提取镜像的功能。

权限详细信息

此策略包含以下权限：

- ecr - 允许委托人读取存储库及其各自的生命周期策略。

AmazonEC2ContainerRegistryReadOnly 策略如下所示。

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ecr:GetAuthorizationToken",
      "ecr:BatchCheckLayerAvailability",
      "ecr:GetDownloadUrlForLayer",
      "ecr:GetRepositoryPolicy",
      "ecr:DescribeRepositories",
      "ecr:ListImages",
      "ecr:DescribeImages",
      "ecr:BatchGetImage",
      "ecr:GetLifecyclePolicy",
      "ecr:GetLifecyclePolicyPreview",
      "ecr:ListTagsForResource",
      "ecr:DescribeImageScanFindings"
    ],
    "Resource": "*"
  }
]
```

AWSECRPullThroughCache_ServiceRolePolicy

您不能将 AWSECRPullThroughCache_ServiceRolePolicy 托管的 IAM policy 附加到您的 IAM 实体。此策略附加到服务相关角色，该角色允许 Amazon ECR 通过缓存提取 workflow 将镜像推送到存储库。有关更多信息，请参阅 [用于缓存提取的 Amazon ECR 服务相关角色](#)。

ECRReplicationServiceRolePolicy

您不能将 ECRReplicationServiceRolePolicy 托管的 IAM policy 附加到您的 IAM 实体。此附加到服务相关角色的策略允许 Amazon ECR 代表您执行操作。有关更多信息，请参阅 [对 Amazon ECR 使用服务相关角色](#)。

Amazon ECR 更新了托 AWS 管政策

查看自该服务开始跟踪这些更改以来对 Amazon ECR AWS 托管政策更新的详细信息。有关此页面更改的自动提示，请订阅 Amazon ECR 文档历史记录页面上的 RSS 源。

更改	描述	日期
AWSECRPullThroughCache_ServiceRolePolicy – 对现有策略的更新	Amazon ECR 将新权限添加到 AWSECRPullThroughCache_ServiceRolePolicy 策略。这些权限允许 Amazon ECR 检索 Secrets Manager 密钥的加密内容。当使用缓存提取规则来缓存需要身份验证的上游注册表中的映像时，需要此类权限。	2023 年 11 月 15 日
AWSECRPullThroughCache_ServiceRolePolicy : 新策略	Amazon ECR 添加了新策略。此策略与用于缓存提取功能的 AWSServiceRoleForECRPullThroughCache 服务相关角色相关。	2021 年 11 月 29 日
ECR ReplicationServiceRolePolicy — 新政策	Amazon ECR 添加了新策略。此策略与用于复制功能的 AWSServiceRoleForECRReplication 服务相关角色相关。	2020 年 12 月 4 日
AmazonEC2 Container Registry FullAccess -更新现有政策	Amazon ECR 将新权限添加到 AmazonEC2ContainerRegistryFullAccess 策略。这些权限允许委托人创建 Amazon ECR 服务相关角色。	2020 年 12 月 4 日
AmazonEC2 Container Registry ReadOnly -更新现有政策	Amazon ECR 将新权限添加到 AmazonEC2ContainerRegistryReadOnly 策略，该策略允许委托人读取生命周期策略、列出标签以及描述镜像的扫描结果。	2019 年 12 月 10 日

更改	描述	日期
AmazonEC2 Container Registry PowerUser -更新现有政策	Amazon ECR 将新权限添加到 AmazonEC2ContainerRegistryPowerUser 策略。这些权限允许委托人读取生命周期策略、列出标签以及描述镜像的扫描结果。	2019 年 12 月 10 日
AmazonEC2 Container Registry FullAccess -更新现有政策	Amazon ECR 将新权限添加到 AmazonEC2ContainerRegistryFullAccess 策略。它们允许校长查找由 CloudTrail 捕获的管理事件或 AWS CloudTrail Insights 事件。	2017 年 11 月 10 日
AmazonEC2 Container Registry ReadOnly -更新现有政策	Amazon ECR 将新权限添加到 AmazonEC2ContainerRegistryReadOnly 策略。这些权限允许委托人描述 Amazon ECR 镜像。	2016 年 10 月 11 日
AmazonEC2 Container Registry PowerUser -更新现有政策	Amazon ECR 将新权限添加到 AmazonEC2ContainerRegistryPowerUser 策略。这些权限允许委托人描述 Amazon ECR 镜像。	2016 年 10 月 11 日
亚马逊 EC2 ContainerRegistry ReadOnly -新政策	Amazon ECR 添加了一个新策略，用于向 Amazon ECR 授予只读权限。这些权限包括列出存储库及其中镜像的功能。它们还包括使用 Docker CLI 从 Amazon ECR 提取镜像的功能。	2015 年 12 月 21 日

更改	描述	日期
亚马逊 EC2 ContainerRegistry PowerUser-新政策	Amazon ECR 添加了一项新策略，该策略授予管理权限，从而允许用户读写存储库，但不允许他们删除存储库或更改应用于存储库的策略文档。	2015 年 12 月 21 日
亚马逊 EC2 ContainerRegistry FullAccess-新政策	Amazon ECR 添加了新策略。此策略授予 Amazon ECR 的完全访问权限。	2015 年 12 月 21 日
Amazon ECR 开始跟踪更改	Amazon ECR 已开始跟踪 AWS 托管策略的更改。	2021 年 6 月 24 日

对 Amazon ECR 使用服务相关角色

Amazon Elastic Container Registry (Amazon ECR) [AWS Identity and Access Management 使用 \(IAM\) 服务相关](#) 角色来提供使用复制和提取缓存功能所需的权限。服务相关角色是一种独特类型的 IAM 角色，它与 Amazon ECR 直接相关。服务相关角色由 Amazon ECR 预定义。它包含该服务支持私有注册表的复制和缓存提取功能所需的所有权限。为注册表配置复制或缓存提取之后，系统会自动创建服务相关角色。有关更多信息，请参阅 [Amazon ECR 中的私有注册表设置](#)。

服务相关角色可让您更轻松地通过 Amazon ECR 设置复制和缓存提取。这是因为，使用该角色，您就不必手动添加所有必要的权限。Amazon ECR 定义其服务相关角色的权限，除非另外定义，否则只有 Amazon ECR 可以代入该角色。定义的权限包括信任策略和权限策略。不能将该权限策略附加到任何其他 IAM 实体。

只有先禁用注册表上的复制或缓存提取之后，才能删除相应的服务相关角色。这可以确保您不会无意中删除 Amazon ECR 对这些功能所需的权限。

有关支持服务相关角色的其他服务的信息，请参阅[与 IAM 配合使用的 AWS 服务](#)。在此链接页面上，查找在服务相关角色列中具有是值的服务。选择是与查看该服务的服务相关角色文档的链接。

主题

- [Amazon ECR 服务相关角色支持的区域](#)
- [用于复制的 Amazon ECR 服务相关角色](#)
- [用于缓存提取的 Amazon ECR 服务相关角色](#)

Amazon ECR 服务相关角色支持的区域

Amazon ECR 在提供 Amazon ECR 服务的所有区域支持使用服务相关角色。有关 Amazon ECR 区域可用性的更多信息，请参阅 [AWS 区域和端点](#)。

用于复制的 Amazon ECR 服务相关角色

Amazon ECR 使用名为的服务相关角色 `AWSServiceRoleForECRReplication`，该角色允许 Amazon ECR 跨多个账户复制映像。

Amazon ECR 的服务相关角色权限

`AWSServiceRoleForECRReplication` 服务相关角色信任以下服务来代入该角色：

- `replication.ecr.amazonaws.com`

以下 `ECRReplicationServiceRolePolicy` 角色权限策略允许 Amazon ECR 对资源使用以下操作：

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:CreateRepository",
        "ecr:ReplicateImage"
      ],
      "Resource": "*"
    }
  ]
}
```

Note

`ReplicateImage` 是 Amazon ECR 用于复制的内部 API，不能直接调用。

必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅 IAM 用户指南中的 [服务相关角色权限](#)。

为 Amazon ECR 创建服务相关角色

无需手动创建 Amazon ECR 服务相关角色。当您在 AWS Management Console、或 AWS API 中为注册表配置复制设置时 AWS CLI，Amazon ECR 会为您创建服务相关角色。

如果您删除了此服务相关角色然后需要再次创建它，则可以使用相同的流程在您的账户中重新创建此角色。当您为注册表配置复制设置时，Amazon ECR 将再次为您创建服务相关角色。

为 Amazon ECR 编辑服务相关角色

Amazon ECR 不允许手动编辑 AWSServiceRoleForECRReplication 服务相关角色。在创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。不过，您可以使用 IAM 编辑角色的说明。有关更多信息，请参阅《IAM 用户指南》中的[编辑服务相关角色](#)。

为 Amazon ECR 删除服务相关角色

如果不再需要使用某个需要服务相关角色的功能或服务，我们建议您删除该角色。这样您就没有未被主动监控或维护的未使用实体。但是，您必须先删除每个区域中的注册表复制配置，才能手动删除服务相关角色。

Note

如果您尝试删除资源，而 Amazon ECR 服务仍在使用该角色，则删除操作可能会失败。如果发生这种情况，请等待几分钟，然后重试。

删除使用的 Amazon ECR 资源 AWSServiceRoleForECRReplication

1. 打开位于 <https://console.aws.amazon.com/ecr/> 的 Amazon ECR 控制台。
2. 从导航栏中，选择为其设置复制配置的区域。
3. 在导航窗格中，选择私有注册表。
4. 在 Private registry (私有注册表) 页面上的 Replication configuration (复制配置) 部分，选择 Edit (编辑)。
5. 要删除所有复制规则，请选择 Delete all (全部删除)。此步骤需要确认。

使用 IAM 手动删除服务相关角色

使用 IAM 控制台 AWS CLI、或 AWS API 删除 AWSServiceRoleForECRReplication 服务相关角色。有关更多信息，请参见《IAM 用户指南》中的[删除服务相关角色](#)。

用于缓存提取的 Amazon ECR 服务相关角色

Amazon ECR 使用名为的服务相关角色 `AWSServiceRoleForECRPullThroughCache`，该角色允许 Amazon ECR 代表您执行操作以完成缓存提取操作。有关缓存提取的更多信息，请参阅[将上游注册表与 Amazon ECR 私有注册表同步](#)。

Amazon ECR 的服务相关角色权限

`AWSServiceRoleForECRPullThroughCache` 服务相关角色信任以下服务来代入该角色。

- `pullthroughcache.ecr.amazonaws.com`

权限详细信息

`AWSECRPullThroughCache_ServiceRolePolicy` 权限策略附加到服务相关角色。此托管策略授权 Amazon ECR 执行以下操作。有关更多信息，请参阅[AWSECRPullThroughCache_ServiceRolePolicy](#)。

- `ecr` – 允许 Amazon ECR 服务将映像推送到私有存储库。
- `secretsmanager:GetSecretValue`— 允许 Amazon ECR 服务检索 AWS Secrets Manager 密钥的加密内容。当使用缓存提取规则在私有注册表中缓存需要身份验证的上游注册表中的映像时，需要此类权限。此权限仅适用于带 `ecr-pullthroughcache/` 名称前缀的密钥。

`AWSECRPullThroughCache_ServiceRolePolicy` 策略包含以下 JSON。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ECR",
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:InitiateLayerUpload",
        "ecr:UploadLayerPart",
        "ecr:CompleteLayerUpload",
        "ecr:PutImage"
      ],
      "Resource": "*"
    }
  ],
  "Resource": "*"
}
```



```
    },
    {
      "Sid": "SecretsManager",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": "arn:aws:secretsmanager:*:*:secret:ecr-pullthroughcache/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceAccount": "${aws:PrincipalAccount}"
        }
      }
    }
  ]
}
```

必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的[服务相关角色权限](#)。

为 Amazon ECR 创建服务相关角色

无需手动为缓存提取创建 Amazon ECR 服务相关角色。当您在 AWS Management Console、或 AWS API 中为私有注册表创建直通缓存规则时 AWS CLI，Amazon ECR 会为您创建服务相关角色。

如果您删除了此服务相关角色然后需要再次创建它，则可以使用相同的流程在您的账户中重新创建此角色。当您为私有注册表创建缓存提取规则时，如果服务相关角色尚不存在，则 Amazon ECR 将为您再次创建该角色。

为 Amazon ECR 编辑服务相关角色

Amazon ECR 不允许手动编辑 `AWSServiceRoleForECRPullThroughCache` 服务相关角色。在创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。不过，您可以使用 IAM 编辑角色的说明。有关更多信息，请参阅《IAM 用户指南》中的[编辑服务相关角色](#)。

为 Amazon ECR 删除服务相关角色

如果不再需要使用某个需要服务相关角色的功能或服务，我们建议您删除该角色。这样您就没有未被主动监控或维护的未使用实体。但是，您必须先删除每个区域中的注册表缓存提取规则，才能手动删除服务相关角色。

Note

如果您尝试删除资源，而 Amazon ECR 服务仍在使用该角色，则删除操作可能会失败。如果发生这种情况，请等待几分钟，然后重试。

要删除 `AWSServiceRoleForECRPullThroughCache` 服务相关角色所使用的 Amazon ECR 资源

1. 打开 Amazon ECR 控制台：<https://console.aws.amazon.com/ecr/>。
2. 从导航栏中，选择创建缓存提取规则所在的区域。
3. 在导航窗格中，选择私有注册表。
4. 在 Private registry (私有注册表) 页面上的 Pull through cache configuration (缓存提取配置) 部分，选择 Edit (编辑)。
5. 对于您创建的每个缓存提取规则，选择该规则，然后选择 Delete rule (删除规则)。

使用 IAM 手动删除服务相关角色

使用 IAM 控制台 AWS CLI、或 AWS API 删除 `AWSServiceRoleForECRPullThroughCache` 服务相关角色。有关更多信息，请参见《IAM 用户指南》中的[删除服务相关角色](#)。

排查 Amazon Elastic Container Registry 的身份和访问权限问题

可以使用以下信息，以帮助您诊断和修复在使用 Amazon ECR 和 IAM 时可能遇到的常见问题。

主题

- [我无权在 Amazon ECR 中执行操作](#)
- [我无权执行 iam : PassRole](#)
- [我想允许我以外的人访问我 AWS 账户的 Amazon ECR 资源](#)

我无权在 Amazon ECR 中执行操作

如果您收到错误提示，表明您无权执行某个操作，则您必须更新策略以允许执行该操作。

当 mateojackson IAM 用户尝试使用控制台查看有关虚构 `my-example-widget` 资源的详细信息，但不拥有虚构 `ecr:GetWidget` 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
  ecr:GetWidget on resource: my-example-widget
```

在此情况下，必须更新 mateojackson 用户的策略，以允许使用 `ecr:GetWidget` 操作访问 `my-example-widget` 资源。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我无权执行 iam : PassRole

如果您收到一个错误，表明您无权执行 `iam:PassRole` 操作，则必须更新策略以允许您将角色传递给 Amazon ECR。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 marymajor 的 IAM 用户尝试使用控制台在 Amazon ECR 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
  iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 `iam:PassRole` 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

我想允许我以外的人访问我 AWS 账户 的 Amazon ECR 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以担任角色。对于支持基于资源的策略或访问控制列表 (ACL) 的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解 Amazon ECR 是否支持这些功能，请参阅 [Amazon Elastic Container Registry 如何与 IAM 结合使用](#)。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户，请参阅 [IAM 用户指南中的向您拥有 AWS 账户 的另一个 IAM 用户提供访问](#) 权限。

- 要了解如何向第三方提供对您的资源的访问[权限 AWS 账户](#)，请参阅 [IAM 用户指南中的向第三方提供访问权限](#)。AWS 账户
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的[为经过外部身份验证的用户（身份联合验证）提供访问权限](#)。
- 要了解使用角色和基于资源的策略进行跨账户存取之间的差别，请参阅《IAM 用户指南》中的[IAM 角色与基于资源的策略有何不同](#)。

Amazon ECR 中的数据保护

[责任 AWS 共担模式](#)适用于亚马逊弹性容器服务中的数据保护。如本模型所述 AWS，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础设施上的内容的控制。您还负责您所使用的 AWS 服务的安全配置和管理任务。有关数据隐私的更多信息，请参阅[数据隐私常见问题](#)。有关欧洲数据保护的信息，请参阅 AWS 安全性博客上的[AWS 责任共担模式和 GDPR](#) 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。我们还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证 (MFA)。
- 使用 SSL/TLS 与资源通信。AWS 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅[《美国联邦信息处理标准 \(FIPS\) 第 140-2 版》](#)。

我们强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您 AWS 服务使用控制台、API 或 AWS 软件开发工具包与 Amazon ECS 或其他服务器合作的情况。AWS CLI 在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供网址，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

主题

- [静态加密](#)

静态加密

Amazon ECR 将镜像存储在 Amazon ECR 管理的 Amazon S3 存储桶中。默认情况下，Amazon ECR 使用具有 Amazon S3 托管加密密钥的服务器端加密，从而使用 AES-256 加密算法对静态数据进行加密。这不需要您采取任何行动，且不会另外收取费用。有关更多信息，请参阅 Amazon Simple Storage Service 用户指南中的[借助使用 Amazon S3 托管式加密密钥的服务器端加密 \(SSE-S3\) 保护数据](#)。

为了更好地控制 Amazon ECR 存储库的加密，您可以使用服务器端加密，并使用存储在 AWS Key Management Service (AWS KMS) 中的 KMS 密钥。在使用 AWS KMS 加密数据时，您可以使用由 Amazon ECR 管理的默认 AWS 托管式密钥，也可以指定自己的 KMS 密钥（称为客户托管密钥）。有关更多信息，请参阅《[亚马逊简单存储服务用户指南](#)》中的[使用服务器端加密和存储在 AWS KMS \(SSE-KMS\) 中的 KMS 密钥保护数据](#)。

每个 Amazon ECR 存储库都有一个加密配置，该配置在创建存储库时进行设置。您可以在每个存储库上使用不同的加密配置。有关更多信息，请参阅[创建 Amazon ECR 私有存储库来存储图像](#)。

在启用 AWS KMS 加密的情况下创建存储库时，将使用 KMS 密钥对存储库的内容进行加密。此外，Amazon ECR 还会向 KMS 密钥添加 AWS KMS 授权，并将 Amazon ECR 存储库作为被授权者委托人。

以下内容提供了对 Amazon ECR 如何与 AWS KMS 集成以加密和解密存储库的高层级理解：

1. 创建存储库时，Amazon ECR 会向发送 [DescribeKey](#) 调用，AWS KMS 以验证和检索加密配置中指定的 KMS 密钥的亚马逊资源名称 (ARN)。
2. Amazon ECR 向发送了两个 [CreateGrant](#) 请求 AWS KMS，要求在 KMS 密钥上创建授权，以允许 Amazon ECR 使用数据密钥加密和解密数据。
3. 推送图像时，会向其发出 [GenerateDataKey](#) 请求 AWS KMS，指定用于加密图像层和清单的 KMS 密钥。
4. AWS KMS 生成新的数据密钥，使用指定的 KMS 密钥对其进行加密，然后发送要与图像层元数据和图像清单一起存储的加密数据密钥。
5. 拉取图像时，会向发出 [Decrypt](#) 请求 AWS KMS，指定加密的数据密钥。
6. AWS KMS 解密加密的数据密钥并将解密后的数据密钥发送到 Amazon S3。
7. 数据密钥用于在提取镜像层之前对其进行解密。

8. 删除存储库后，Amazon ECR 会向发送两个 [RetireGrant](#) 请求，要求停 AWS KMS 用为该存储库创建的授权。

注意事项

在 Amazon ECR 中使用 AWS KMS 加密功能时，应考虑以下几点。

- 如果您创建采用 KMS 加密的 Amazon ECR 存储库，但未指定 KMS 密钥，则 Amazon ECR 会默认使用 AWS 托管式密钥 带有别名 `aws/ecr` 的。首次创建启用 KMS 加密的存储库时，在您的账户中创建此 KMS 密钥。
- 当您将 KMS 加密与自己的 KMS 密钥结合使用时，密钥必须与您的存储库位于同一个区域中。
- Amazon ECR 代表您创建的授权不应被撤销。如果您撤销授予 Amazon ECR 使用您账户中 AWS KMS 密钥的权限，Amazon ECR 将无法访问这些数据、加密推送到存储库的新映像，也无法在提取时对其进行解密。当您撤销 Amazon ECR 授权时，更改将立即生效。要撤销访问权限，则应删除存储库，而不是撤销该授权。删除存储库后，Amazon ECR 会代表您停用授权。
- 使用 AWS KMS 密钥需要付费。有关更多信息，请参阅 [AWS Key Management Service 定价](#)。

所需的 IAM 权限

创建或删除使用 AWS KMS 进行服务器端加密的 Amazon ECR 存储库时，所需的权限取决于您正在使用的特定 KMS 密钥。

使用适用于 Amazon ECR 时需要 AWS 托管式密钥 的 IAM 权限

默认情况下，如果为 Amazon ECR 存储库启用了 AWS KMS 加密，但未指定 KMS 密钥，则使用 AWS 托管式密钥 适用于 Amazon ECR 的。当使用 Amazon ECR 的 AWS 托管 KMS 密钥加密存储库时，任何有权创建存储库的委托人也可以在存储库上启用 AWS KMS 加密。但是，删除存储库的 IAM 委托人必须具有 `kms:RetireGrant` 权限。这样可以停用创建存储库时添加到 AWS KMS 密钥中的授权。

以下示例 IAM policy 可作为内联策略添加到用户，以确保用户具有删除启用加密的存储库所需的最低权限。可以使用资源参数指定用于加密存储库的 KMS 密钥。

```
{
  "Version": "2012-10-17",
  "Id": "ecr-kms-permissions",
  "Statement": [
    {
```

```

        "Sid": "AllowAccessToRetireTheGrantsAssociatedWithTheKey",
        "Effect": "Allow",
        "Action": [
            "kms:RetireGrant"
        ],
        "Resource": "arn:aws:kms:us-
west-2:111122223333:key/b8d9ae76-080c-4043-92EXAMPLE"
    }
]
}

```

使用客户托管密钥时所需的 IAM 权限

使用客户托管密钥创建启用 AWS KMS 加密功能的存储库时，创建存储库的用户或角色必须拥有 KMS 密钥策略和 IAM 策略的权限。

创建自己的 KMS 密钥时，您可以使用原定设置密钥策略 AWS KMS 创建，也可以指定自己的密钥策略。为确保账户所有者可以管理客户托管的密钥，KMS 密钥的密钥策略应允许账户的根用户 AWS KMS 执行所有操作。可以向密钥策略添加额外的作用域权限，但至少应向根用户授予管理 KMS 密钥的权限。要允许 KMS 密钥仅用于源自 Amazon ECR 的请求，您可以将 kms: [ViaService 条件密钥与值](#) 一起使用。ecr.<region>.amazonaws.com

以下示例密钥策略为拥有 KMS 密钥的 AWS 账户（根用户）提供了对 KMS 密钥的完全访问权限。有关此示例密钥策略的更多信息，请参阅 AWS Key Management Service 开发人员指南 [中的允许访问 AWS 账户和启用 IAM 策略](#)。

```

{
  "Version": "2012-10-17",
  "Id": "ecr-key-policy",
  "Statement": [
    {
      "Sid": "EnableIAMUserPermissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    }
  ]
}

```

除了必要的 Amazon ECR 权限外，创建存储库的 IAM 用户 `kms:CreateGrant`、`kms:RetireGrant`、IAM 角色或 AWS 账户还必须拥有、和 `kms:DescribeKey` 权限。

Note

`kms:RetireGrant` 权限必须添加到创建存储库的用户或角色的 IAM policy 中。`kms:CreateGrant` 和 `kms:DescribeKey` 权限可以添加到 KMS 密钥的密钥策略或创建存储库的用户或角色的 IAM policy 中。有关 AWS KMS 权限工作原理的更多信息，请参阅《AWS Key Management Service 开发者指南》中的 [AWS KMS API 权限：操作和资源参考](#)。

以下示例 IAM policy 可作为内联策略添加到用户，以确保用户拥有创建启用加密的存储库所需的最低权限，并在完成存储库时删除存储库。可以使用资源参数指定用于加密存储库的 AWS KMS key。

```
{
  "Version": "2012-10-17",
  "Id": "ecr-kms-permissions",
  "Statement": [
    {
      "Sid":
"AllowAccessToCreateAndRetireTheGrantsAssociatedWithTheKeyAsWellAsDescribeTheKey",
      "Effect": "Allow",
      "Action": [
        "kms:CreateGrant",
        "kms:RetireGrant",
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:us-
west-2:111122223333:key/b8d9ae76-080c-4043-92EXAMPLE"
    }
  ]
}
```

创建存储库时，允许用户在控制台中列出 KMS 密钥

使用 Amazon ECR 控制台创建存储库时，您可以授予权限，允许用户在启用存储库加密时于区域中列出客户托管的 KMS 密钥。以下 IAM policy 示例显示了使用控制台时列出 KMS 密钥和别名所需的权限。

```
{
```



```

"Version": "2012-10-17",
"Statement": {
  "Effect": "Allow",
  "Action": [
    "kms:ListKeys",
    "kms:ListAliases",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
}

```

监控 Amazon ECR 与 AWS KMS 的集成

您可以使用 AWS CloudTrail 来跟踪 Amazon ECR 代表您发送 AWS KMS 的请求。日志中的 CloudTrail 日志条目包含加密上下文密钥，便于识别。

Amazon ECR 加密上下文

加密上下文 是一组包含任意非机密数据的键值对。当您在加密数据的请求中包含加密上下文时，会以加密 AWS KMS 方式将加密上下文绑定到加密数据。要解密数据，您必须传入相同的加密上下文。

在对的“[加密GenerateDataKey](#)”和“[解密](#)”请求中，AWS KMS Amazon ECR 使用具有两个名称值对的加密上下文，用于标识存储库和正在使用的 Amazon S3 存储桶。如以下示例所示。名称不会变化，但与其组合的加密上下文会因每个值而异。

```

"encryptionContext": {
  "aws:s3:arn": "arn:aws:s3::us-west-2-starport-manifest-bucket/EXAMPLE1-90ab-cdef-fedc-ba987BUCKET1/sha256:a7766145a775d39e53a713c75b6fd6d318740e70327aaa3ed5d09e0ef33fc3df",
  "aws:ecr:arn": "arn:aws:ecr:us-west-2:111122223333:repository/repository-name"
}

```

您可以使用加密上下文在审计记录和日志（例如和 Amazon CloudWatch Logs）中识别这些加密操作，并作为策略和授权中的授权条件。[AWS CloudTrail](#)

Amazon ECR 加密上下文包含两个名称-值对。

- `aws:s3:arn` – 第一个名称 - 值对标识存储桶。键是 `aws:s3:arn`。值是 Amazon S3 存储桶的 Amazon Resource Name (ARN)。

```
"aws:s3:arn": "ARN of an Amazon S3 bucket"
```

例如，如果存储桶的 ARN 是 `arn:aws:s3:::us-west-2-starport-manifest-bucket/EXAMPLE1-90ab-cdef-fedc-ba987BUCKET1/sha256:a7766145a775d39e53a713c75b6fd6d318740e70327aaa3ed5d09e0ef33fc3df`，加密上下文将包括以下对。

```
"arn:aws:s3:::us-west-2-starport-manifest-bucket/EXAMPLE1-90ab-cdef-fedc-ba987BUCKET1/sha256:a7766145a775d39e53a713c75b6fd6d318740e70327aaa3ed5d09e0ef33fc3df"
```

- `aws:ecr:arn` – 第二个名称 - 值对标识存储库的 Amazon Resource Name (ARN)。键是 `aws:ecr:arn`。值是存储库的 ARN。

```
"aws:ecr:arn": "ARN of an Amazon ECR repository"
```

例如，如果存储库的 ARN 是 `arn:aws:ecr:us-west-2:111122223333:repository/repository-name`，加密上下文将包括以下对。

```
"aws:ecr:arn": "arn:aws:ecr:us-west-2:111122223333:repository/repository-name"
```

故障排除

使用控制台删除 Amazon ECR 存储库时，如果存储库已成功删除，但 Amazon ECR 无法停用添加到存储库 KMS 密钥的授权，您将收到以下错误消息。

```
The repository [{repository-name}] has been deleted successfully but the grants created by the kmsKey [{kms_key}] failed to be retired
```

发生这种情况时，您可以自己取消对存储库的 AWS KMS 授权。

手动取消对存储库的 AWS KMS 授权

1. 列出用于存储库的 AWS KMS 密钥的授权。key-id 值包含在您从控制台收到的错误中。您还可以使用 `list-keys` 命令列出账户中特定区域中的客户托管的 KMS 密钥、AWS 托管式密钥和客户托管的 KMS 密钥。

```
aws kms list-grants \
  --key-id b8d9ae76-080c-4043-9237-c815bfc21dfc
  --region us-west-2
```

输出包括 EncryptionContextSubset 以及存储库的 Amazon Resource Name (ARN)。这可用于确定添加到密钥中的哪个授权是您想要停用的授权。GrantId 值将在下一步中停用授权时使用。

- 取消为存储库添加的 AWS KMS 密钥的每项授权。将的值替换为 *GrantId* 上一步输出中的授权 ID。

```
aws kms retire-grant \  
  --key-id b8d9ae76-080c-4043-9237-c815bfc21dfc \  
  --grant-id GrantId \  
  --region us-west-2
```

Amazon Elastic Container Registry 的合规性验证

要了解是否属于特定合规计划的范围，请参阅 AWS 服务 [“按合规计划划分的范围”](#)，然后选择您感兴趣的合规计划。AWS 服务 有关一般信息，请参阅 [AWS 合规计划](#) [AWS](#)。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 [“下载报告”](#) 中的 [“AWS Artifact”](#)。

您在使用 AWS 服务 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。AWS 提供了以下资源来帮助实现合规性：

- [安全与合规性快速入门指南](#) — 这些部署指南讨论了架构注意事项，并提供了部署以安全性和合规性为重点 AWS 的基准环境的步骤。
- 在 [Amazon Web Services 上构建 HIPAA 安全与合规性](#) — 本白皮书描述了各公司如何使用 AWS 来创建符合 HIPAA 资格的应用程序。

Note

并非所有 AWS 服务 人都符合 HIPAA 资格。有关更多信息，请参阅 [符合 HIPAA 要求的服务参考](#)。

- [AWS 合规资源](#) [AWS](#) — 此工作簿和指南集可能适用于您所在的行业和所在地区。
- [AWS 客户合规指南](#) — 从合规角度了解责任共担模式。这些指南总结了保护的最佳实践，AWS 服务 并将指南映射到跨多个框架（包括美国国家标准与技术研究院 (NIST)、支付卡行业安全标准委员会 (PCI) 和国际标准化组织 (ISO)) 的安全控制。

- [使用AWS Config 开发人员指南中的规则评估资源](#) — 该 AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- [AWS Security Hub](#)— 这 AWS 服务 提供了您内部安全状态的全面视图 AWS。Security Hub 通过安全控件评估您的 AWS 资源并检查其是否符合安全行业标准和最佳实践。有关受支持服务及控件的列表，请参阅 [Security Hub 控件参考](#)。
- [Amazon GuardDuty](#) — 它通过监控您的 AWS 账户环境中是否存在可疑和恶意活动，来 AWS 服务检测您的工作负载、容器和数据面临的潜在威胁。GuardDuty 通过满足某些合规性框架规定的入侵检测要求，可以帮助您满足各种合规性要求，例如 PCI DSS。
- [AWS Audit Manager](#)— 这 AWS 服务 可以帮助您持续审计 AWS 使用情况，从而简化风险管理以及对法规和行业标准的合规性。

Amazon Elastic Registry 中的基础设施安全性

作为一项托管服务，Amazon Elastic Container Registry 受 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅[AWS 云安全](#)。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS ecurity Pillar Well-Architected Fram ework 中的[基础设施保护](#)。

您可以使用 AWS 已发布的 API 调用通过网络访问 Amazon ECR。客户端必须支持以下内容：

- 传输层安全性协议 (TLS)。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密 (PFS) 的密码套件，例如 DHE (临时 Diffie-Hellman) 或 ECDHE (临时椭圆曲线 Diffie-Hellman)。大多数现代系统 (如 Java 7 及更高版本) 都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 委托人关联的秘密访问密钥来对请求进行签名。或者，您可以使用 [AWS Security Token Service](#) (AWS STS) 生成临时安全凭证来对请求进行签名。

您可以从任何网络位置调用这些 API 操作，但 Amazon ECR 不支持基于资源的访问策略，其中可以包含基于源 IP 地址的限制。您还可以使用 Amazon ECR 策略来控制来自特定 Amazon Virtual Private Cloud (Amazon VPC) 终端节点或特定 VPC 的访问。实际上，这可以将对给定 Amazon ECR 资源的网络访问与网络中的特定 VPC 隔离开来。AWS 有关更多信息，请参阅 [Amazon ECR 接口 VPC 终端节点 \(\)AWS PrivateLink](#)。

Amazon ECR 接口 VPC 终端节点 ()AWS PrivateLink

您可以将 Amazon ECR 配置为使用接口 VPC 终端节点以改善 VPC 的安全状况。VPC 终端节点由一项技术提供支持 AWS PrivateLink，该技术使您能够通过私有 IP 地址私密访问 Amazon ECR API。

AWS PrivateLink 将您的 VPC 和 Amazon ECR 之间的所有网络流量限制到亚马逊网络。您无需互联网网关、NAT 设备或虚拟私有网关。

有关 AWS PrivateLink 和 VPC 终端节点的更多信息，请参阅 Amazon VPC 用户指南中的 [VPC 终端节点](#)。

Amazon ECR VPC 终端节点的注意事项

在为 Amazon ECR 配置 VPC 终端节点之前，请注意以下事项：

- 要允许 Amazon EC2 实例上托管的 Amazon ECR 任务从 Amazon ECR 中提取私有镜像，请确保您也为 Amazon ECS 创建接口 VPC 终端节点。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 [接口 VPC 终端节点 \(AWS PrivateLink\)](#)。

Important

在 Fargate 上托管的 Amazon ECS 任务不需要 Amazon ECS 接口 VPC 终端节点。

- 使用 Linux 平台版本 1.3.0 或更早版本在 Fargate 上托管的 Amazon ECS 任务只需要 `com.amazonaws.region.ecr.dkr` Amazon ECR VPC 终端节点和 Amazon S3 网关终端节点，即可利用此功能。
- 使用 Linux 平台版本 1.4.0 或更早版本在 Fargate 上托管的 Amazon ECS 任务需要 `com.amazonaws.region.ecr.dkr` 和 `com.amazonaws.region.ecr.api` Amazon ECR VPC 端点以及 Amazon S3 网关端点，才能利用此功能。
- 使用 Windows 平台版本 1.0.0 或更早版本在 Fargate 上托管的 Amazon ECS 任务需要 `com.amazonaws.region.ecr.dkr` 和 `com.amazonaws.region.ecr.api` Amazon ECR VPC 端点以及 Amazon S3 网关端点，才能利用此功能。
- 从 Amazon ECR 提取容器镜像的 Amazon ECS 任务（在 Fargate 上托管）可以通过向其任务的任务执行 IAM 角色添加条件键，限制对其任务使用的特定 VPC 和服务使用的 VPC 终端节点的访问。有关更多信息，请参阅 Amazon Elastic Container Service 开发人员指南中的 [用于通过接口终端节点提取 Amazon ECR 镜像的 Fargate 任务的可选 IAM 权限](#)。
- 托管在 Fargate 上的 Amazon ECS 任务需要使用日志 CloudWatch VPC 终端节点才能从 Amazon ECR 提取容器镜像，这些镜像也使用 `awslogs` 日志驱动程序向 CloudWatch 日志发送日志信息。有关更多信息，请参阅 [创建 CloudWatch 日志端点](#)。
- 附加到 VPC 端点的安全组必须允许端口 443 上来自 VPC 的私有子网的传入连接。
- VPC 端点当前不支持跨区域请求。确保在计划向 Amazon ECR 发出 API 调用的同一区域中创建 VPC 终端节点。

- VPC 端点目前不支持 Amazon ECR 公有存储库。考虑使用缓存提取规则将公有映像托管在与 VPC 端点位于同一区域的私有存储库中。有关更多信息，请参阅 [将上游注册表与 Amazon ECR 私有注册表同步](#)。
- VPC 终端节点仅支持通过 Amazon Route 53 AWS 提供的 DNS。如果您希望使用自己的 DNS，可以使用条件 DNS 转发。有关更多信息，请参阅 Amazon VPC 用户指南中的 [DHCP 选项集](#)。
- 如果您的容器具有与 Amazon S3 的现有连接，则在添加 Amazon S3 网关端点时，其连接可能会短暂中断。要避免此中断，请创建一个使用 Amazon S3 网关端点的新 VPC，然后将 Amazon ECS 集群及其容器迁移到该新 VPC。
- 首次使用缓存提取规则提取镜像时，如果您已将 Amazon ECR 配置为通过 AWS PrivateLink 使用接口 VPC 终端节点，您需要在同一个 VPC 中创建一个带有 NAT 网关的公有子网，然后将至互联网的所有出站流量从私有子网路由到 NAT 网关，才能使提取操作正常工作。后续的镜像提取不需要此操作。有关更多信息，请参阅 Amazon Virtual Private Cloud 用户指南中的 [场景：从私有子网访问互联网](#)。

Windows 镜像注意事项

基于 Windows 操作系统的镜像包括受许可证限制而无法分发的构件。预设情况下，当您 Windows 镜像推送到 Amazon ECR 存储库时，包含这些构件的层不会被推送，因为它们被视为外部层。当构件由 Microsoft 提供时，将从 Microsoft Azure 基础设施中检索外部层。因此，要使容器能够从 Azure 中提取这些外部层，除了创建 VPC 终端节点之外，还需要执行其他步骤。

当您 Windows 镜像推送到 Amazon ECR 时，可以通过使用 Docker 守护进程中的 `--allow-nondistributable-artifacts` 标记覆盖此行为。启用后，此标记会将许可层推送到 Amazon ECR，从而使这些镜像可以通过 VPC 终端节点从 Amazon ECR 提取，而无需额外访问 Azure。

Important

使用 `--allow-nondistributable-artifacts` 标记不会排除您遵守 Windows 容器基础镜像许可证条款的义务；您不能发布 Windows 内容以供公共或第三方重新分发。允许在您自己的环境中使用。

要在 Docker 安装中启用此标记，您必须修改 Docker 守护进程配置文件，根据您的 Docker 安装，该配置文件通常可以在 Docker Engine 部分下的设置或首选项菜单中配置，也可以通过直接编辑 `C:\ProgramData\docker\config\daemon.json` 文件配置。

以下是所需配置的示例。将值替换为要将镜像推送到其中的存储库 URI。

```
{
  "allow-nondistributable-artifacts": [
    "111122223333.dkr.ecr.us-west-2.amazonaws.com"
  ]
}
```

修改 Docker 守护进程配置文件后，您必须在尝试推送镜像之前重新启动 Docker 守护进程。通过验证基础层是否推送到您的存储库，确认推送工作运转正常。

Note

Windows 镜像的基础层很庞大。层太大将导致延长推送时间，并在 Amazon ECR 中增加这些镜像的存储成本。出于这些原因，我们建议仅在严格要求此选项以减少构建时间和持续存储成本时使用此选项。例如，在 Amazon ECR 中压缩之后，`mcr.microsoft.com/windows/servercore` 镜像的大小约为 1.7 GiB。

为 Amazon ECR 创建 VPC 终端节点

要为 Amazon ECR 服务创建 VPC 终端节点，请使用 Amazon VPC 用户指南中的[创建接口终端节点过程](#)。

在 Amazon EC2 实例上托管的 Amazon ECS 任务需要 Amazon ECR 终端节点和 Amazon S3 网关终端节点。

使用平台版本 1.4.0 或更高版本在 Fargate 上托管的 Amazon ECS 任务需要 Amazon ECR VPC 终端节点和 Amazon S3 网关终端节点。

使用平台版本 1.3.0 或更早版本在 Fargate 上托管的 Amazon ECS 任务只需要 `com.amazonaws.region.ecr.dkr` Amazon ECR VPC 终端节点和 Amazon S3 网关终端节点。

Note


创建终端节点的顺序无关紧要。

`com.amazonaws.region.ecr.dkr`

此终端节点用于 Docker Registry API。诸如 `push` 和 `pull` 这样的 Docker 客户端命令使用此终端节点。

在创建此终端节点时，您必须启用私有 DNS 主机名。为此，请确保在创建 VPC 终端节点时，在 Amazon VPC 控制台中选择启用私有 DNS 名称选项。

com.amazonaws.**region**.ecr.api

 Note

指定的##表示 Amazon ECR 支持的 AWS 区域（例如us-east-2美国东部（俄亥俄州）地区的区域标识符。

此终端节点用于对 Amazon ECR API 执行的调用。API 操作（如 DescribeImages 和 CreateRepository）转到此终端节点。

在创建此终端节点时，您可以选择启用私有 DNS 主机名。在创建 VPC 终端节点时，通过在 VPC 控制台中选择启用私有 DNS 名称，可启用此设置。如果您为 VPC 终端节点启用私有 DNS 主机名，请将您的 SDK 或更新 AWS CLI 到最新版本，以便在使用 SDK 或 AWS CLI 时无需指定终端节点 URL。

如果您启用私有 DNS 主机名并且使用的是 2019 年 1 月 24 日之前发布的 SDK 或 AWS CLI 版本，则必须使用--endpoint-url参数来指定接口终端节点。以下示例显示了终端节点 URL 的格式。

```
aws ecr create-repository --repository-name name --endpoint-url https://  
api.ecr.region.amazonaws.com
```

如果您不为 VPC 终端节点启用私有 DNS 主机名，则必须使用 --endpoint-url 参数并指定接口终端节点的 VPC 终端节点 ID。以下示例显示了终端节点 URL 的格式。

```
aws ecr create-repository --repository-name name --endpoint-url  
https://VPC_endpoint_ID.api.ecr.region.vpce.amazonaws.com
```

创建 Amazon S3 网关终端节点

对于从 Amazon ECR 提取私有镜像的 Amazon ECS 任务，您必须为 Amazon S3 创建网关终端节点。必须创建网关终端节点，因为 Amazon ECR 使用 Amazon S3 来存储您的镜像层。当容器从 Amazon ECR 下载镜像时，它们必须访问 Amazon ECR 才能获取镜像清单，然后 Amazon S3 才能下载实际镜像层。以下是包含每个 Docker 镜像层的 Amazon S3 存储桶的 Amazon Resource Name (ARN)。

```
arn:aws:s3:::prod-region-starport-layer-bucket/*
```


使用 Amazon VPC 用户指南中的[创建网关终端节点](#)过程为 Amazon ECR 创建以下 Amazon S3 网关终端节点。在创建终端节点时，请务必为您的 VPC 选择路由表。

com.amazonaws.*region*.s3

Amazon S3 网关终端节点使用 IAM policy 文档来限制对服务的访问。可以使用完全访问权限策略，因为您在任务 IAM 角色或其他 IAM 用户策略中设置的任何限制仍然适用于此策略。如果要限制 Amazon S3 存储桶访问权限为使用 Amazon ECR 所需的最低权限，请参阅[Amazon ECR 的 Amazon S3 存储桶最低权限](#)。

Amazon ECR 的 Amazon S3 存储桶最低权限

Amazon S3 网关终端节点使用 IAM policy 文档来限制对服务的访问。要仅允许 Amazon ECR 的 Amazon S3 存储桶最低权限，请在为终端节点创建 IAM policy 文档时，限制对 Amazon ECR 使用的 Amazon S3 存储桶的访问权限。

下表描述了 Amazon ECR 所需的 Amazon S3 存储桶策略权限。

权限	描述
arn:aws:s3:::prod- <i>region</i> -starport-layer-bucket/*	提供对包含每个 Docker 镜像层的 Amazon S3 存储桶的访问权限。表示 Amazon ECR 支持的 AWS 区域的区域标识符，例如美国东部（俄亥俄）区域的 us-east-2。

示例

以下示例说明如何提供对 Amazon ECR 操作所需的 Amazon S3 存储桶的访问权限。

```
{
  "Statement": [
    {
      "Sid": "Access-to-specific-bucket-only",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::prod-region-starport-layer-bucket/*"]
    }
  ]
}
```

```

    }
  ]
}

```

创建 CloudWatch 日志端点

使用 Fargate 启动类型的 Amazon ECS 任务需要你创建 `com.amazonaws`，这些任务使用没有互联网网关的 VPC，也使用 `awslogs` CloudWatch 日志驱动程序向日志发送日志信息。 `regi@@ on .logs` 接口日志的 VPC 终端节点。 CloudWatch 有关更多信息，请参阅 Amazon CloudWatch Logs 用户指南中的 [将 CloudWatch 日志与接口 VPC 终端节点配合使用](#)。

为 Amazon ECR VPC 端点创建终端节点策略

VPC 端点策略是一种 IAM 资源策略，您在创建或修改端点时可将它附加到端点。如果您在创建终端节点时未附加策略，AWS 则会为您附加允许完全访问服务的默认策略。端点策略不会覆盖或替换用户策略或服务特定的策略。这是一个单独的策略，用于控制从端点中对指定服务进行的访问。端点策略必须采用 JSON 格式编写。有关更多信息，请参阅《Amazon VPC 用户指南》中的 [使用 VPC 端点控制对服务的访问权限](#)。

我们建议您创建一个 IAM 资源策略，并将该策略同时附加到两个 Amazon ECR VPC 终端节点。

下面是用于 Amazon ECR 的端点策略示例。此策略允许特定 IAM 角色从 Amazon ECR 中提取镜像。

```

{
  "Statement": [{
    "Sid": "AllowPull",
    "Principal": {
      "AWS": "arn:aws:iam::1234567890:role/role_name"
    },
    "Action": [
      "ecr:BatchGetImage",
      "ecr:GetDownloadUrlForLayer",
      "ecr:GetAuthorizationToken"
    ],
    "Effect": "Allow",
    "Resource": "*"
  }]
}

```

下面的终端节点策略示例阻止删除指定的存储库。

```
{
  "Statement": [{
    "Sid": "AllowAll",
    "Principal": "*",
    "Action": "*",
    "Effect": "Allow",
    "Resource": "*"
  },
  {
    "Sid": "PreventDelete",
    "Principal": "*",
    "Action": "ecr:DeleteRepository",
    "Effect": "Deny",
    "Resource": "arn:aws:ecr:region:1234567890:repository/repository_name"
  }
  ]
}
```

下面的终端节点策略示例将前面的两个示例组合到一个策略中。

```
{
  "Statement": [{
    "Sid": "AllowAll",
    "Effect": "Allow",
    "Principal": "*",
    "Action": "*",
    "Resource": "*"
  },
  {
    "Sid": "PreventDelete",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "ecr:DeleteRepository",
    "Resource": "arn:aws:ecr:region:1234567890:repository/repository_name"
  },
  {
    "Sid": "AllowPull",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::1234567890:role/role_name"
    },
    "Action": [
      "ecr:BatchGetImage",

```

```
    "ecr:GetDownloadUrlForLayer",
        "ecr:GetAuthorizationToken"
    ],
    "Resource": "*"
}
]
```

修改 Amazon ECR 的 VPC 终端节点策略

1. 通过以下网址打开 Amazon VPC 控制台：<https://console.aws.amazon.com/vpc/>。
2. 在导航窗格中，选择端点。
3. 如果您没有为 Amazon ECR 创建 VPC 终端节点，请参阅 [为 Amazon ECR 创建 VPC 终端节点](#)。
4. 选择要将策略添加到的 Amazon ECR VPC 终端节点，然后在屏幕的下半部分中选择策略选项卡。
5. 选择编辑策略并对策略进行更改。
6. 选择保存以保存策略。

共享子网

您无法在与您共享的子网中创建、描述、修改或删除 VPC 端点。但是，您可以在与您共享的子网中使用 VPC 端点。

防止跨服务混淆座席

混淆代理问题是一个安全性问题，即不具有操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在中 AWS，跨服务模仿可能会导致混乱的副手问题。一个服务（呼叫服务）调用另一项服务（所谓的“服务”）时，可能会发生跨服务模拟。可以操纵调用服务，使用其权限以在其他情况下该服务不应有访问权限的方式对另一个客户的资源进行操作。为防止这种情况，AWS 提供可帮助您保护所有服务的数据的工具，而这些服务中的服务主体有权限访问账户中的资源。

我们建议在资源策略中使用 [aws:SourceArn](#) 或 [aws:SourceAccount](#) 全局条件上下文键，以限制 Amazon ECR 为其他服务提供的资源访问权限。如果您只希望将一个资源与跨服务访问相关联，请使用 `aws:SourceArn`。如果您想允许该账户中的任何资源与跨服务使用操作相关联，请使用 `aws:SourceAccount`。

防范混淆代理问题最有效的方法是使用 `aws:SourceArn` 全局条件上下文键和资源的完整 ARN。如果不知道资源的完整 ARN，或者正在指定多个资源，请针对

ARN 未知部分使用带有通配符字符 (*) 的 `aws:SourceArn` 全局上下文条件键。例如，`arn:aws:service:region:123456789012:*`。

如果 `aws:SourceArn` 值不包含账户 ID，例如 Amazon S3 存储桶 ARN，您必须使用两个全局条件上下文键来限制权限。

`aws:SourceArn` 的值必须为 `ResourceDescription`。

以下示例展示了如何使用 Amazon ECR 存储库策略中的 `aws:SourceArn` 和 `aws:SourceAccount` 全局条件上下文密钥来允许 AWS CodeBuild 访问与该服务集成所必需的 Amazon ECR API 操作，同时还可以防止出现混乱的代理问题。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeBuildAccess",
      "Effect": "Allow",
      "Principal": {
        "Service": "codebuild.amazonaws.com"
      },
      "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer"
      ],
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:codebuild:region:123456789012:project/project-  
name"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

Amazon ECR 监控

您可以通过亚马逊监控您的 Amazon ECR API 使用情况 CloudWatch，亚马逊会收集来自亚马逊 ECR 的原始数据并将其处理为可读的、近乎实时的指标。这些统计数据会记录两周，以便您可以访问历史信息并深入了解自己的 API 使用情况。Amazon ECR 指标数据将在一分钟 CloudWatch 内自动发送到。有关的更多信息 CloudWatch，请参阅 [Amazon CloudWatch 用户指南](#)。

Amazon ECR 基于您的 API 用量提供指标，这些指标适用于授权、镜像推送和镜像提取操作。

监控是维护 Amazon ECR 和您的 AWS 解决方案的可靠性、可用性和性能的重要组成部分。我们建议您从构成 AWS 解决方案的资源中收集监控数据，以便在出现多点故障时可以更轻松地进行调试。不过，在开始监控 Amazon ECR 之前，您应制定一个监控计划并在计划中回答下列问题：

- 监控目的是什么？
- 您将监控哪些资源？
- 监控这些资源的频率如何？
- 您将使用哪些监控工具？
- 谁负责执行监控任务？
- 出现错误时应通知谁？

下一步，通过在不同时间和不同负载条件下测量性能，在您的环境中建立正常 Amazon ECR 性能的基准。在监控 Amazon ECR 时，存储历史监控数据，以便将此数据与新的性能数据进行比较，确定正常性能模式和性能异常，并设计解决问题的方法。

主题

- [可视化 Service Quotas 并设置警报](#)
- [Amazon ECR 用量指标](#)
- [Amazon ECR 用量报告](#)
- [Amazon ECR 存储库指标](#)
- [亚马逊 ECR 事件和 EventBridge](#)
- [使用记录 Amazon ECR 操作 AWS CloudTrail](#)

可视化 Service Quotas 并设置警报

您可以使用 CloudWatch 控制台直观显示您的服务配额，并查看您的当前使用量与服务配额的对比情况。还可以设置警报，从而在接近配额时向您发送通知。

可视化服务配额并选择性地设置警报

1. 打开 CloudWatch 控制台，[网址为 https://console.aws.amazon.com/cloudwatch/](https://console.aws.amazon.com/cloudwatch/)。
2. 在导航窗格中，选择指标。
3. 在所有指标选项卡上，选择用量，然后选择按 AWS 资源。

这将显示服务配额用量指标的列表。

4. 选中其中一个指标旁边的复选框。

该图表显示您当前对该 AWS 资源的使用情况。

5. 要将服务配额添加到图表，请执行以下操作：

- a. 选择 Graphed metrics (绘制的指标) 选项卡。
- b. 选择数学表达式、从空表达式开始。然后在新行中，在详细信息下，输入 **SERVICE_QUOTA(m1)**。

这将向图表中添加一个新行，并显示指标中表示的资源的服务配额。

6. 要以配额百分比的形式查看您的当前用量，请添加新的表达式或更改当前 SERVICE_QUOTA 表达式。对于新的表达式，请使用 **m1/60/SERVICE_QUOTA(m1)*100**。
7. (可选) 要设置一个警报，以便在接近服务配额时向您发送通知，请执行以下操作：

- a. 在 **m1/60/SERVICE_QUOTA(m1)*100** 行上的操作下，选择警报图标。该图标看起来像一个铃铛。

这将显示警报创建页面。

- b. 在条件下，确保阈值类型为静态，并将当 Expression1 为设置为大于。在多于下，输入 **80**。这将创建一个警报，当用量超过配额的 80% 时，该警报将进入 ALARM 状态。
- c. 选择下一步。
- d. 在下一页上，选择一个 Amazon SNS 主题或创建一个新主题。当警报进入 ALARM 状态时，会向此主题发送通知。然后选择下一步。
- e. 在下一页上，输入警报的名称和描述，然后选择下一步。
- f. 选择创建警报。

Amazon ECR 用量指标

您可以使用 CloudWatch 使用量指标来了解您的账户的资源使用情况。使用这些指标在 CloudWatch 图表和仪表板上可视化您当前的服务使用情况。

Amazon ECR 使用率指标与 AWS 服务配额相对应。您可以配置警报，以在用量接近服务配额时向您发出警报。有关 Amazon ECR 默认服务配额的更多信息，请参阅 [Amazon ECR 服务配额](#)。

Amazon ECR 在 AWS/Usage 命名空间中发布以下指标。

指标	描述
CallCount	<p>从您的账户调用 API 操作的次数。资源由与指标关联的维度定义。</p> <p>此指标最有用的统计数据是 SUM，表示定义时段内来自所有贡献者的值的总和。</p>

以下维度用于优化由 Amazon ECR 发布的用量指标。

维度	描述
Service	包含资源的 AWS 服务的名称。对于 Amazon ECR 用量指标，此维度的值为 ECR。
Type	正在报告的实体的类型。目前，Amazon ECR 用量指标的唯一有效值为 API。
Resource	<p>正在运行的资源的类型。目前，Amazon ECR 会返回有关以下 API 操作的 API 用量的信息。</p> <ul style="list-style-type: none">GetAuthorizationTokenBatchCheckLayerAvailabilityInitiateLayerUploadUploadLayerPartCompleteLayerUploadPutImage

维度	描述
	<ul style="list-style-type: none">• BatchGetImage• GetDownloadUrlForLayer
Class	所跟踪的资源的类。Amazon ECR 目前不使用类维度。

Amazon ECR 用量报告

AWS 提供名为 Cost Explorer 的免费报告工具，使您能够分析 Amazon ECR 资源的成本和使用情况。

使用 Cost Explorer 查看用量和成本的图表。您可以查看之前 13 个月的数据，并预测您在接下来三个月内可能产生的费用。您可以使用 Cost Explorer 查看有关您一段时间内在 AWS 资源方面的费用的模式、确定需要进一步查询的方面以及查看可用于了解您的成本的趋势。您还可以指定数据的时间范围，并按天或按月查看时间数据。

成本和用量报告中的计量数据显示跨所有 Amazon ECR 存储库的用量。有关更多信息，请参阅 [标记资源以便于计费](#)。

有关创建 AWS 成本和使用情况报告的更多信息，请参阅《AWS Billing 用户指南》中的 [AWS 成本和使用情况报告](#)。

Amazon ECR 存储库指标

Amazon ECR 向亚马逊 CloudWatch 发送存储库提取计数指标。Amazon ECR 指标数据会 CloudWatch 在 1 分钟内自动发送到。有关的更多信息 CloudWatch，请参阅 [Amazon CloudWatch 用户指南](#)。

主题

- [启用 CloudWatch 指标](#)
- [可用指标和维度](#)
- [使用控制台查看 Amazon ECR 指标 CloudWatch](#)

启用 CloudWatch 指标

Amazon ECR 自动发送所有存储库的存储库指标。无需执行任何手动步骤。

可用指标和维度

以下部分列出了 Amazon ECR 发送给亚马逊 CloudWatch 的指标和维度。

Amazon ECR 指标

Amazon ECR 提供了可用于监控存储库的指标。您可以测量拉取计数。

AWS/ECR 命名空间包括以下指标。

RepositoryPullCount

对存储库中映像的总拉取次数。

有效维度：RepositoryName。

有效统计数据：平均值、最小值、最大值、总计和样本数。最有用的统计指标是和。

单位：整数。

Amazon ECR 指标的维度

Amazon ECR 指标使用 AWS/ECR 命名空间并提供以下维度的指标。

RepositoryName

此维度将筛选您为指定存储库中所有容器映像请求的数据。

使用控制台查看 Amazon ECR 指标 CloudWatch

您可以在 CloudWatch 控制台上查看 Amazon ECR 存储库指标。CloudWatch 控制台可对您的资源进行精细且可自定义的显示。有关更多信息，请参阅 [Amazon CloudWatch 用户指南](#)。

在 CloudWatch 控制台中查看指标

1. 打开 CloudWatch 控制台，[网址为 https://console.aws.amazon.com/cloudwatch/](https://console.aws.amazon.com/cloudwatch/)。
2. 在左侧导航中，依次选择 Metrics (指标)、All metrics (所有指标)。
3. 在 Browse (浏览) 选项卡的 AWS Namespaces (命名空间) 下，选择 ECR。
4. 选择要查看的指标。存储库指标的范围为 ECR > Repository Metrics (ECR > 存储库指标)。

亚马逊 ECR 事件和 EventBridge

Amazon EventBridge 使您能够实现 AWS 服务自动化，并自动响应系统事件，例如应用程序可用性问题或资源更改。来自 AWS 服务的事件几乎实时 EventBridge 地传送到。您可以编写简单规则来指示您关注的事件，并包括要在事件匹配规则时执行的自动化操作。可自动触发的操作包括：

- 将事件添加到日志中的 CloudWatch 日志组
- 调用函数 AWS Lambda
- 调用 Amazon EC2 Run Command
- 将事件中继到 Amazon Kinesis Data Streams
- 激活 AWS Step Functions 状态机
- 通知 Amazon SNS 主题或 Amazon SQS 队列

有关更多信息，请参阅 [《亚马逊 EventBridge 用户指南》EventBridge 中的“亚马逊入门”](#)。

来自 Amazon ECR 的示例事件

以下是来自 Amazon ECR 的示例事件。尽最大努力发出事件。

已完成镜像推送的事件

每个镜像推送完成后，将发送以下事件。有关更多信息，请参阅 [将 Docker 镜像推送到亚马逊 ECR 私有存储库](#)。

```
{
  "version": "0",
  "id": "13cde686-328b-6117-af20-0e5566167482",
  "detail-type": "ECR Image Action",
  "source": "aws.ecr",
  "account": "123456789012",
  "time": "2019-11-16T01:54:34Z",
  "region": "us-west-2",
  "resources": [],
  "detail": {
    "result": "SUCCESS",
    "repository-name": "my-repository-name",
    "image-digest":
      "sha256:7f5b2640fe6fb4f46592dfd3410c4a79dac4f89e4782432e0378abcd1234",
    "action-type": "PUSH",
    "image-tag": "latest"
  }
}
```

```
}  
}
```

缓存提取操作的事件

尝试缓存提取操作时，将发送以下事件。有关更多信息，请参阅 [将上游注册表与 Amazon ECR 私有注册表同步](#)。

```
{  
  "version": "0",  
  "id": "85fc3613-e913-7fc4-a80c-a3753e4aa9ae",  
  "detail-type": "ECR Pull Through Cache Action",  
  "source": "aws.ecr",  
  "account": "123456789012",  
  "time": "2023-02-29T02:36:48Z",  
  "region": "us-west-2",  
  "resources": [  
    "arn:aws:ecr:us-west-2:123456789012:repository/docker-hub/alpine"  
  ],  
  "detail": {  
    "rule-version": "1",  
    "sync-status": "SUCCESS",  
    "ecr-repository-prefix": "docker-hub",  
    "repository-name": "docker-hub/alpine",  
    "upstream-registry-url": "public.ecr.aws",  
    "image-tag": "3.17.2",  
    "image-digest":  
    "sha256:4aa08ef415aecc80814cb42fa41b658480779d80c77ab15EXAMPLE",  
  }  
}
```

已完成镜像扫描的事件 (基本扫描)

为注册表启用基本扫描后，当每个镜像扫描完成时，会发送以下事件。finding-severity-counts 参数仅返回严重性级别的值 (如果存在)。例如，如果镜像不包含任何 CRITICAL 级别的结果，则不会返回任何关键计数。有关更多信息，请参阅 [扫描图像以查找 Amazon ECR 中的操作系统漏洞](#)。

Note

有关启用增强扫描后 Amazon Inspector 发出的事件的详细信息，请参阅 [EventBridge 为了在 Amazon ECR 中进行增强扫描而发送的事件](#)。

```
{
  "version": "0",
  "id": "85fc3613-e913-7fc4-a80c-a3753e4aa9ae",
  "detail-type": "ECR Image Scan",
  "source": "aws.ecr",
  "account": "123456789012",
  "time": "2019-10-29T02:36:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ecr:us-east-1:123456789012:repository/my-repository-name"
  ],
  "detail": {
    "scan-status": "COMPLETE",
    "repository-name": "my-repository-name",
    "finding-severity-counts": {
      "CRITICAL": 10,
      "MEDIUM": 9
    },
    "image-digest":
      "sha256:7f5b2640fe6fb4f46592dfd3410c4a79dac4f89e4782432e0378abcd1234",
    "image-tags": []
  }
}
```

启用了增强扫描的资源的变更通知事件 (增强扫描)

为注册表启用增强扫描后，当启用了增强扫描的资源发生更改时，Amazon ECR 将发送以下事件。这包括正在创建的新存储库、正在更改的存储库的扫描频率，或者在启用了增强扫描功能的存储库中创建或删除镜像的时间。有关更多信息，请参阅 [扫描图像以查找 Amazon ECR 中的软件漏洞](#)。

```
{
  "version": "0",
  "id": "0c18352a-a4d4-6853-ef53-0ab8638973bf",
  "detail-type": "ECR Scan Resource Change",
  "source": "aws.ecr",
  "account": "123456789012",
  "time": "2021-10-14T20:53:46Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "action-type": "SCAN_FREQUENCY_CHANGE",
    "repositories": [{
      "repository-name": "repository-1",
```

```

"repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-1",
"scan-frequency": "SCAN_ON_PUSH",
"previous-scan-frequency": "MANUAL"
},
{
"repository-name": "repository-2",
"repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-2",
"scan-frequency": "CONTINUOUS_SCAN",
"previous-scan-frequency": "SCAN_ON_PUSH"
},
{
"repository-name": "repository-3",
"repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-3",
"scan-frequency": "CONTINUOUS_SCAN",
"previous-scan-frequency": "SCAN_ON_PUSH"
}
],
"resource-type": "REPOSITORY",
"scan-type": "ENHANCED"
}
}

```

镜像删除的事件

删除镜像时将发送以下事件。有关更多信息，请参阅 [在 Amazon ECR 中删除图片](#)。

```

{
"version": "0",
"id": "dd3b46cb-2c74-f49e-393b-28286b67279d",
"detail-type": "ECR Image Action",
"source": "aws.ecr",
"account": "123456789012",
"time": "2019-11-16T02:01:05Z",
"region": "us-west-2",
"resources": [],
"detail": {
"result": "SUCCESS",
"repository-name": "my-repository-name",
"image-digest":
"sha256:7f5b2640fe6fb4f46592dfd3410c4a79dac4f89e4782432e0378abcd1234",
"action-type": "DELETE",
"image-tag": "latest"
}
}

```

}

使用记录 Amazon ECR 操作 AWS CloudTrail

Amazon ECR 与 AWS CloudTrail 一项服务集成，该服务提供用户、角色或 AWS 服务在 Amazon ECR 中采取的操作的记录。CloudTrail 将以下 Amazon ECR 操作捕获为事件：

- 所有 API 调用，包括来自 Amazon ECR 控制台的调用
- 由于存储库上的加密设置而采取的所有操作
- 由于生命周期策略规则而采取的所有操作，包括成功和不成功的操作

Important

由于单个 CloudTrail 事件的大小限制，对于 10 张或更多图像过期的生命周期策略操作，Amazon ECR 会向发送多个事件。CloudTrail 此外，Amazon ECR 中每个镜像最多可包含 100 个标签。

创建跟踪后，您可以允许将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括 Amazon ECR 的事件。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的“事件历史记录”中查看最新的事件。使用此信息，您可以确定向 Amazon ECR 发出了什么请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间以及其他详细信息。

有关更多信息，请参阅 [《AWS CloudTrail 用户指南》](#)。

亚马逊 ECR 信息位于 CloudTrail

CloudTrail 在您创建 AWS 账户时已在您的账户上启用。当 Amazon ECR 中发生活动时，该活动会与其他 AWS 服务 CloudTrail 事件一起记录在事件历史记录中。您可以在自己的 AWS 账户中查看、搜索和下载最近发生的事件。有关更多信息，请参阅 [使用事件历史记录查看 CloudTrail 事件](#)。

要持续记录您的 AWS 账户中的事件，包括 Amazon ECR 的事件，请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。在控制台中创建跟踪时，您可以将跟踪应用到单个区域或所有区域。跟踪记录 AWS 分区中的事件，并将日志文件传送到您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务来分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅：

- [为您的 AWS 账户创建跟踪](#)
- [AWS 与日志的服务集成 CloudTrail](#)

- [配置 Amazon SNS 通知 CloudTrail](#)
- [接收来自多个区域的 CloudTrail 日志文件](#)和[接收来自多个账户的 CloudTrail 日志文件](#)

所有 Amazon ECR API 操作均由《[亚马逊弹性容器注册表 API 参考](#)》记录 CloudTrail 并记录在案。执行常见任务时，会在 CloudTrail 日志文件中为该任务中的每个 API 操作生成部分。例如，当您创建存储库时 `GetAuthorizationTokenCreateRepository`，会在 CloudTrail 日志文件中生成 `SetRepositoryPolicy` 章节。当您将其某个镜像推送到存储库中时，则将生成 `InitiateLayerUpload`、`UploadLayerPart`、`CompleteLayerUpload` 和 `PutImage` 部分。当您提取镜像时，则将生成 `GetDownloadUrlForLayer` 和 `BatchGetImage` 部分。有关这些常见任务的示例，请参阅 [CloudTrail 日志条目示例](#)。

每个事件或日记账条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根用户凭证还是 用户凭证发出的
- 请求是使用角色还是联合用户的临时安全凭证发出的
- 请求是否由其他 AWS 服务发出

有关更多信息，请参阅[CloudTrailuserIdentity元素](#)。

了解 Amazon ECR 日志文件条目

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关所请求操作的信息、操作的日期和时间、请求参数以及其他信息。CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序出现。

CloudTrail 日志条目示例

以下是一些常见 Amazon ECR 任务的 CloudTrail 日志条目示例。

Note

为提高可读性，这些示例已进行格式化处理。在 CloudTrail 日志文件中，所有条目和事件都连接成一行。此外，该示例限于一个 Amazon ECR 条目。在真实的 CloudTrail 日志文件中，您可以看到来自多个 AWS 服务的条目和事件。

主题

- [示例：创建存储库操作](#)
- [示例：创建 Amazon ECR 存储库时的 AWS KMS CreateGrant API 操作](#)
- [示例：镜像推送操作](#)
- [示例：镜像提取操作](#)
- [示例：镜像生命周期策略操作](#)

示例：创建存储库操作

以下示例显示了演示该CreateRepository操作的 CloudTrail 日志条目。

```
{
  "eventVersion": "1.04",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:account_name",
    "arn": "arn:aws:sts::123456789012:user/Mary_Major",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-07-11T21:54:07Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::123456789012:role/Admin",
        "accountId": "123456789012",
        "userName": "Admin"
      }
    }
  },
  "eventTime": "2018-07-11T22:17:43Z",
  "eventSource": "ecr.amazonaws.com",
  "eventName": "CreateRepository",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "203.0.113.12",
  "userAgent": "console.amazonaws.com",
  "requestParameters": {
    "repositoryName": "testrepo"
  },
}
```

```
"responseElements": {
  "repository": {
    "repositoryArn": "arn:aws:ecr:us-east-2:123456789012:repository/testrepo",
    "repositoryName": "testrepo",
    "repositoryUri": "123456789012.dkr.ecr.us-east-2.amazonaws.com/testrepo",
    "createdAt": "Jul 11, 2018 10:17:44 PM",
    "registryId": "123456789012"
  }
},
"requestID": "cb8c167e-EXAMPLE",
"eventID": "e3c6f4ce-EXAMPLE",
"resources": [
  {
    "ARN": "arn:aws:ecr:us-east-2:123456789012:repository/testrepo",
    "accountId": "123456789012"
  }
],
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

示例：创建 Amazon ECR 存储库时的 AWS KMS CreateGrant API 操作

以下示例显示了一个 CloudTrail 日志条目，该条目演示了在启用了 KMS 加密的情况下创建 Amazon ECR 存储库时的 AWS KMS CreateGrant 操作。对于每个启用 KMS 加密创建的存储库，您应该会在中看到两个 CreateGrant 日志条目 CloudTrail。

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAIIEP6W46J43IG7LXAQ",
    "arn": "arn:aws:iam::123456789012:user/Mary_Major",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major",
    "sessionContext": {
      "sessionIssuer": {
        },
      "webIdFederationData": {
        },
      }
    }
  }
}
```

```
    "attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2020-06-10T19:22:10Z"
    }
  },
  "invokedBy": "AWS Internal"
},
"eventTime": "2020-06-10T19:22:10Z",
"eventSource": "kms.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-west-2",
"sourceIPAddress": "203.0.113.12",
"userAgent": "console.amazonaws.com",
"requestParameters": {
  "keyId": "4b55e5bf-39c8-41ad-b589-18464af7758a",
  "granteePrincipal": "ecr.us-west-2.amazonaws.com",
  "operations": [
    "GenerateDataKey",
    "Decrypt"
  ],
  "retiringPrincipal": "ecr.us-west-2.amazonaws.com",
  "constraints": {
    "encryptionContextSubset": {
      "aws:ecr:arn": "arn:aws:ecr:us-west-2:123456789012:repository/testrepo"
    }
  }
},
"responseElements": {
  "grantId": "3636af9adfee1accb67b83941087dcd45e7fadc4e74ff0103bb338422b5055f3"
},
"requestID": "047b7dea-b56b-4013-87e9-a089f0f6602b",
"eventID": "af4c9573-c56a-4886-baca-a77526544469",
"readOnly": false,
"resources": [
  {
    "accountId": "123456789012",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-west-2:123456789012:key/4b55e5bf-39c8-41ad-
b589-18464af7758a"
  }
],
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
```

```
}

```

示例：镜像推送操作

以下示例显示了一个 CloudTrail 日志条目，该条目演示了使用该PutImage操作的图像推送。

Note

推送图像时，您还将在 CloudTrail 日志中看到InitiateLayerUploadUploadLayerPart、和CompleteLayerUpload引用。

```
{
  "eventVersion": "1.04",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:account_name",
    "arn": "arn:aws:sts::123456789012:user/Mary_Major",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2019-04-15T16:42:14Z"
      }
    }
  },
  "eventTime": "2019-04-15T16:45:00Z",
  "eventSource": "ecr.amazonaws.com",
  "eventName": "PutImage",
  "awsRegion": "us-east-2",
  "sourceIPAddress": "203.0.113.12",
  "userAgent": "console.amazonaws.com",
  "requestParameters": {
    "repositoryName": "testrepo",
    "imageTag": "latest",
    "registryId": "123456789012",
    "imageManifest": "{\n  \"schemaVersion\": 2,\n  \"mediaType\": \"application/\n  vnd.docker.distribution.manifest.v2+json\",\n  \"config\": {\n    \"mediaType\":\n  \"application/vnd.docker.container.image.v1+json\",\n    \"size\": 5543,\n    \"digest\": \"sha256:000b9b805af1cdb60628898c9f411996301a1c13afd3dbef1d8a16ac6dbf503a"
  }
}
```

```

{"layers": [{"mediaType": "application/
vnd.docker.image.rootfs.diff.tar.gzip", "size": 43252507, "digest": "sha256:3b37166ec61459e76e33282dda08f2a9cd698ca7e3d6bc44e6a6e7580cdeff8e"}, {"mediaType": "application/
vnd.docker.image.rootfs.diff.tar.gzip", "size": 846, "digest": "sha256:504facff238fde83f1ca8f9f54520b4219c5b8f80be9616ddc52d31448a044bd"}, {"mediaType": "application/
vnd.docker.image.rootfs.diff.tar.gzip", "size": 615, "digest": "sha256:ebbcacd28e101968415b0c812b2d2dc60f969e36b0b08c073bf796e12b1bb449"}, {"mediaType": "application/
vnd.docker.image.rootfs.diff.tar.gzip", "size": 850, "digest": "sha256:c7fb3351ecad291a88b92b600037e2435c84a347683d540042086fe72c902b8a"}, {"mediaType": "application/
vnd.docker.image.rootfs.diff.tar.gzip", "size": 168, "digest": "sha256:2e3debadcbf7e542e2aefbce1b64a358b1931fb403b3e4aeca27cb4d809d56c2"}, {"mediaType": "application/vnd.docker.image.rootfs.diff.tar.gzip", "size": 37720774, "digest": "sha256:f8c9f51ad524d8ae9bf4db69cd3e720ba92373ec265f5c390ffb21bb0c277941"}, {"mediaType": "application/
vnd.docker.image.rootfs.diff.tar.gzip", "size": 30432107, "digest": "sha256:813a50b13f61cf1f8d25f19fa96ad3aa5b552896c83e86ce413b48b091d7f01b"}, {"mediaType": "application/
vnd.docker.image.rootfs.diff.tar.gzip", "size": 197, "digest": "sha256:7ab043301a6187ea3293d80b30ba06c7bf1a0c3cd4c43d10353b31bc0cecf7d"}, {"mediaType": "application/
vnd.docker.image.rootfs.diff.tar.gzip", "size": 154, "digest": "sha256:67012cca8f31dc3b8ee2305e7762fee20c250513effdedb38a1c37784a5a2e71"}, {"mediaType": "application/
vnd.docker.image.rootfs.diff.tar.gzip", "size": 176, "digest": "sha256:3bc892145603fffc9b1c97c94e2985b4cb19ca508750b15845a5d97becbd1a0e"}, {"mediaType": "application/
vnd.docker.image.rootfs.diff.tar.gzip", "size": 183, "digest": "sha256:6f1c79518f18251d35977e7e46bfa6c6b9cf50df2a79d4194941d95c54258d18"}, {"mediaType": "application/
vnd.docker.image.rootfs.diff.tar.gzip", "size": 212, "digest": "sha256:b7bcfbc2e2888afebede4dd1cd5eebf029bb6315feeaf0b56e425e11a50afe42"}, {"mediaType": "application/
vnd.docker.image.rootfs.diff.tar.gzip", "size": 212, "digest": "sha256:2b220f8b0f32b7c2ed8eaafe1c802633bbd94849b9ab73926f0ba46cdae91629"}],
"responseElements": {
  "image": {
    "repositoryName": "testrepo",

```

```

    "imageManifest": "{\n  \"schemaVersion\": 2,\n  \"mediaType\": \"application/  
vnd.docker.distribution.manifest.v2+json\", \n  \"config\": {\n    \"mediaType\":  
    \"application/vnd.docker.container.image.v1+json\", \n    \"size\": 5543, \n  
    \"digest\": \"sha256:000b9b805af1cdb60628898c9f411996301a1c13afd3dbef1d8a16ac6dbf503a  
    \"\n  }, \n  \"layers\": [\n    {\n      \"mediaType\": \"application/  
vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 43252507, \n  
      \"digest\": \"sha256:3b37166ec61459e76e33282dda08f2a9cd698ca7e3d6bc44e6a6e7580cdeff8e  
    \"\n    }, \n    {\n      \"mediaType\": \"application/  
vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 846, \n      \"digest  
\": \"sha256:504facff238fde83f1ca8f9f54520b4219c5b8f80be9616ddc52d31448a044bd  
    \"\n    }, \n    {\n      \"mediaType\": \"application/  
vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 615, \n      \"digest  
\": \"sha256:ebbcacd28e101968415b0c812b2d2dc60f969e36b0b08c073bf796e12b1bb449\" \n  
    }, \n    {\n      \"mediaType\": \"application/  
vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 850, \n      \"digest  
\": \"sha256:c7fb3351ecad291a88b92b600037e2435c84a347683d540042086fe72c902b8a  
    \"\n    }, \n    {\n      \"mediaType\": \"application/  
vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 168, \n      \"digest\":  
    \"sha256:2e3debadcbf7e542e2aefbce1b64a358b1931fb403b3e4aeca27cb4d809d56c2\" \n    }, \n    {\n      \"mediaType\": \"application/vnd.docker.image.rootfs.diff.tar.gzip  
\", \n      \"size\": 37720774, \n      \"digest\":  
    \"sha256:f8c9f51ad524d8ae9bf4db69cd3e720ba92373ec265f5c390ffb21bb0c277941\" \n  
    }, \n    {\n      \"mediaType\": \"application/  
vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 30432107, \n  
      \"digest\": \"sha256:813a50b13f61cf1f8d25f19fa96ad3aa5b552896c83e86ce413b48b091d7f01b  
    \"\n    }, \n    {\n      \"mediaType\": \"application/  
vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 197, \n      \"digest  
\": \"sha256:7ab043301a6187ea3293d80b30ba06c7bf1a0c3cd4c43d10353b31bc0cecf7d  
    \"\n    }, \n    {\n      \"mediaType\": \"application/  
vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 154, \n      \"digest  
\": \"sha256:67012cca8f31dc3b8ee2305e7762fee20c250513effdedb38a1c37784a5a2e71\" \n  
    }, \n    {\n      \"mediaType\": \"application/  
vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 176, \n      \"digest  
\": \"sha256:3bc892145603fffc9b1c97c94e2985b4cb19ca508750b15845a5d97becbd1a0e  
    \"\n    }, \n    {\n      \"mediaType\": \"application/  
vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 183, \n      \"digest  
\": \"sha256:6f1c79518f18251d35977e7e46bfa6c6b9cf50df2a79d4194941d95c54258d18\" \n  
    }, \n    {\n      \"mediaType\": \"application/  
vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 212, \n      \"digest  
\": \"sha256:b7bcfbc2e2888afebede4dd1cd5eebf029bb6315feeaf0b56e425e11a50afe42\" \n  
    }, \n    {\n      \"mediaType\": \"application/  
vnd.docker.image.rootfs.diff.tar.gzip\", \n      \"size\": 212, \n      \"digest\":  
    \"sha256:2b220f8b0f32b7c2ed8eaafe1c802633bbd94849b9ab73926f0ba46cdae91629\" \n    } \n  ] \n}"

```

```
"registryId": "123456789012",
"imageId": {
  "imageDigest":
"sha256:98c8b060c21d9adbb6b8c41b916e95e6307102786973ab93a41e8b86d1fc6d3e",
  "imageTag": "latest"
}
},
"requestID": "cf044b7d-5f9d-11e9-9b2a-95983139cc57",
"eventID": "2bfd4ee2-2178-4a82-a27d-b12939923f0f",
"resources": [{
  "ARN": "arn:aws:ecr:us-east-2:123456789012:repository/testrepo",
  "accountId": "123456789012"
}],
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

示例：镜像提取操作

以下示例显示了一个 CloudTrail 日志条目，该条目演示了使用该BatchGetImage操作的图像拉取。

Note

拉取图像时，如果您在本地还没有该图像，则还会在 CloudTrail 日志中看到GetDownloadUrlForLayer引用。

```
{
  "eventVersion": "1.04",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:account_name",
    "arn": "arn:aws:sts::123456789012:user/Mary_Major",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Mary_Major",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2019-04-15T16:42:14Z"
      }
    }
  }
}
```

```
}
},
"eventTime": "2019-04-15T17:23:20Z",
"eventSource": "ecr.amazonaws.com",
"eventName": "BatchGetImage",
"awsRegion": "us-east-2",
"sourceIPAddress": "203.0.113.12",
"userAgent": "console.amazonaws.com",
"requestParameters": {
  "imageIds": [{
    "imageTag": "latest"
  }],
  "acceptedMediaTypes": [
    "application/json",
    "application/vnd.oci.image.manifest.v1+json",
    "application/vnd.oci.image.index.v1+json",
    "application/vnd.docker.distribution.manifest.v2+json",
    "application/vnd.docker.distribution.manifest.list.v2+json",
    "application/vnd.docker.distribution.manifest.v1+prettyjws"
  ],
  "repositoryName": "testrepo",
  "registryId": "123456789012"
},
"responseElements": null,
"requestID": "2a1b97ee-5fa3-11e9-a8cd-cd2391aeda93",
"eventID": "c84f5880-c2f9-4585-9757-28fa5c1065df",
"resources": [{
  "ARN": "arn:aws:ecr:us-east-2:123456789012:repository/testrepo",
  "accountId": "123456789012"
}],
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

示例：镜像生命周期策略操作

以下示例显示了一个 CloudTrail 日志条目，该条目演示了图像何时因生命周期策略规则而过期。可通过筛选事件名称字段的 PolicyExecutionEvent 来定位此事件类型。

⚠ Important

由于单个 CloudTrail 事件的大小限制，对于 10 张或更多图像过期的生命周期策略操作，Amazon ECR 会向发送多个事件。CloudTrail 此外，Amazon ECR 中每个镜像最多可包含 100 个标签。

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "accountId": "123456789012",
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2020-03-12T20:22:12Z",
  "eventSource": "ecr.amazonaws.com",
  "eventName": "PolicyExecutionEvent",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "AWS Internal",
  "userAgent": "AWS Internal",
  "requestParameters": null,
  "responseElements": null,
  "eventID": "9354dd7f-9aac-4e9d-956d-12561a4923aa",
  "readOnly": true,
  "resources": [
    {
      "ARN": "arn:aws:ecr:us-west-2:123456789012:repository/testrepo",
      "accountId": "123456789012",
      "type": "AWS::ECR::Repository"
    }
  ],
  "eventType": "AwsServiceEvent",
  "recipientAccountId": "123456789012",
  "serviceEventDetails": {
    "repositoryName": "testrepo",
    "lifecycleEventPolicy": {
      "lifecycleEventRules": [
        {
          "rulePriority": 1,
          "description": "remove all images > 2",
          "lifecycleEventSelection": {
            "tagStatus": "Any",
            "tagPrefixList": [],

```

```
        "countType": "Image count more than",
        "countNumber": 2
    },
    "action": "expire"
}
],
"lastEvaluatedAt": 0,
"policyVersion": 1,
"policyId": "ceb86829-58e7-9498-920c-aa042e33037b"
},
"lifecycleEventImageActions": [
{
    "lifecycleEventImage": {
        "digest":
"sha256:ddba4d27a7ffc3f86dd6c2f92041af252a1f23a8e742c90e6e1297bfa1bc0c45",
        "tagStatus": "Tagged",
        "tagList": [
            "alpine"
        ],
        "pushedAt": 1584042813000
    },
    "rulePriority": 1
},
{
    "lifecycleEventImage": {
        "digest":
"sha256:6ab380c5a5acf71c1b6660d645d2cd79cc8ce91b38e0352cbf9561e050427baf",
        "tagStatus": "Tagged",
        "tagList": [
            "centos"
        ],
        "pushedAt": 1584042842000
    },
    "rulePriority": 1
}
]
}
}
```

将 Amazon ECR 与软件开发工具包一起使用 AWS

AWS 软件开发套件 (SDK) 可用于许多流行的编程语言。每个软件开发工具包都提供 API、代码示例和文档，使开发人员能够更轻松地了解其首选语言构建应用程序。

SDK 文档	代码示例
AWS SDK for C++	AWS SDK for C++ 代码示例
AWS CLI	AWS CLI 代码示例
AWS SDK for Go	AWS SDK for Go 代码示例
AWS SDK for Java	AWS SDK for Java 代码示例
AWS SDK for JavaScript	AWS SDK for JavaScript 代码示例
AWS SDK for Kotlin	AWS SDK for Kotlin 代码示例
AWS SDK for .NET	AWS SDK for .NET 代码示例
AWS SDK for PHP	AWS SDK for PHP 代码示例
AWS Tools for PowerShell	PowerShell 代码示例工具
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) 代码示例
AWS SDK for Ruby	AWS SDK for Ruby 代码示例
AWS SDK for Rust	AWS SDK for Rust 代码示例
适用于 SAP ABAP 的 AWS SDK	适用于 SAP ABAP 的 AWS SDK 代码示例
AWS SDK for Swift	AWS SDK for Swift 代码示例

示例可用性

找不到所需的内容？通过使用此页面底部的提供反馈链接请求代码示例。

使用 AWS 软件开发工具包的 Amazon ECR 的代码示例

以下代码示例展示了如何将 Amazon ECR 与 AWS 软件开发套件 (SDK) 一起使用。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景和跨服务示例的上下文查看操作。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅[将 Amazon ECR 与软件开发工具包一起使用 AWS](#)。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

代码示例

- [使用 AWS 软件开发工具包对 Amazon ECR 执行的操作](#)
 - [DescribeRepositories与 AWS SDK 或 CLI 配合使用](#)
 - [ListImages与 AWS SDK 或 CLI 配合使用](#)

使用 AWS 软件开发工具包对 Amazon ECR 执行的操作

以下代码示例演示了如何使用 AWS 软件开发工具包执行单个 Amazon ECR 操作。这些摘录调用 Amazon ECR API，是必须在上下文中运行的大型程序的代码摘录。每个示例都包含一个指向的链接 GitHub，您可以在其中找到有关设置和运行代码的说明。

以下示例仅包括最常用的操作。如需完整列表，请参阅[亚马逊弹性容器注册表 \(Amazon ECR\) Container Registry API 参考](#)。

示例

- [DescribeRepositories与 AWS SDK 或 CLI 配合使用](#)
- [ListImages与 AWS SDK 或 CLI 配合使用](#)

DescribeRepositories与 AWS SDK 或 CLI 配合使用

以下代码示例演示如何使用 DescribeRepositories。

CLI

AWS CLI

在注册表中描述存储库

此示例将描述账户默认注册表中的存储库。

命令:

```
aws ecr describe-repositories
```

输出:

```
{
  "repositories": [
    {
      "registryId": "012345678910",
      "repositoryName": "ubuntu",
      "repositoryArn": "arn:aws:ecr:us-west-2:012345678910:repository/
ubuntu"
    },
    {
      "registryId": "012345678910",
      "repositoryName": "test",
      "repositoryArn": "arn:aws:ecr:us-west-2:012345678910:repository/test"
    }
  ]
}
```

- 有关 API 的详细信息，请参阅AWS CLI 命令参考[DescribeRepositories](#)中的。

Rust

适用于 Rust 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
async fn show_repos(client: &aws_sdk_ecr::Client) -> Result<(),
aws_sdk_ecr::Error> {
    let rsp = client.describe_repositories().send().await?;
```

```
let repos = rsp.repositories();

println!("Found {} repositories:", repos.len());

for repo in repos {
    println!("  ARN: {}", repo.repository_arn().unwrap());
    println!("  Name: {}", repo.repository_name().unwrap());
}

Ok(())
}
```

- 有关 API 的详细信息，请参阅适用[DescribeRepositories](#)于 Rust 的 AWS SDK API 参考。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅[将 Amazon ECR 与软件开发工具包一起使用 AWS](#)。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

ListImages 与 AWS SDK 或 CLI 配合使用

以下代码示例演示如何使用 ListImages。

CLI

AWS CLI

列出存储库的映像

以下 list-images 示例将显示 cluster-autoscaler 存储库的映像列表。

```
aws ecr list-images \
  --repository-name cluster-autoscaler
```

输出：

```
{
  "imageIds": [
    {
      "imageDigest":
"sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",
      "imageTag": "v1.13.8"
    },
  ],
}
```

```
{
  "imageDigest":
"sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",
  "imageTag": "v1.13.7"
},
{
  "imageDigest":
"sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
  "imageTag": "v1.13.6"
}
]
```

- 有关 API 的详细信息，请参阅AWS CLI 命令参考[ListImages](#)中的。

Rust

适用于 Rust 的 SDK

Note

还有更多相关信息 GitHub。在 [AWS 代码示例存储库](#) 中查找完整示例，了解如何进行设置和运行。

```
async fn show_images(
    client: &aws_sdk_ecr::Client,
    repository: &str,
) -> Result<(), aws_sdk_ecr::Error> {
    let rsp = client
        .list_images()
        .repository_name(repository)
        .send()
        .await?;

    let images = rsp.image_ids();

    println!("found {} images", images.len());

    for image in images {
        println!(
            "image: {}:{}",

```

```
        image.image_tag().unwrap(),
        image.image_digest().unwrap()
    );
}

ok(())
}
```

- 有关 API 的详细信息，请参阅适用[ListImages](#)于 Rust 的AWS SDK API 参考。

有关 S AWS DK 开发者指南和代码示例的完整列表，请参阅[将 Amazon ECR 与软件开发工具包一起使用 AWS](#)。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

Amazon ECR 服务配额

下表提供了 Amazon Elastic Container Registry (Amazon ECR) 的默认服务配额。

姓名	默认值	是否可调整	说明
复制配置中每条规则的筛选器	每个受支持的区域：100 个	不可以	复制配置中每条规则的筛选器的最大数量。
每个存储库的镜像数	每个支持的区域：10,000 个	可以	每个存储库的最大镜像数。
层分段	每个支持的区域：4,200 个	不可以	层分段数上限。仅当您使用 Amazon ECR API 操作直接启动镜像推送操作的分段上传时才适用。
生命周期策略长度	每个支持的区域：30,720 个	不可以	生命周期策略中的最大字符数。
层分段大小上限	每个受支持的区域：10 个	不可以	层分段的最大大小 (MiB)。仅当您使用 Amazon ECR API 操作直接启动镜像推送操作的分段上传时才适用。
层大小上限	每个支持的区域：52000 个	不可以	层的最大大小 (MiB)。

姓名	默认值	是否可调整	说明
层分段大小下限	每个受支持的区域：5 个	不可以	层分段的最小大小 (MiB)。仅当您使用 Amazon ECR API 操作直接启动镜像推送操作的分段上传时才适用。
每个注册表的缓存提取规则	每个受支持的区域：50 个	不可以	缓存提取规则的最大数量。
BatchCheckLayerAvailability 请求的速率	每个受支持的区域：每秒 1000 个	可以	您在当前区域中每秒可以发出的 BatchCheckLayerAvailability 请求的最大数量。将镜像推送到存储库时，会检查每个镜像层以验证之前是否已上传它。如果已上传，则会跳过镜像层。
BatchGetImage 请求的速率	每个支持的区域：每秒 2,000 个	可以	您在当前区域中每秒可以发出的 BatchGetImage 请求的最大数量。提取镜像时，会调用 BatchGetImage API 一次以检索镜像清单。如果您请求提高此 API 的配额，则还请查看 GetDownloadUrlForLayer 的使用情况。

姓名	默认值	是否可调整	说明
CompleteLayerUpload 请求的速率	每个支持的区域： 每秒 100 个	可以	您在当前区域中每秒可以发出的 CompleteLayerUpload 请求的最大数量。推送镜像时，对于每个新的镜像层，都会调用一次 CompleteLayerUpload API，以验证上传是否已完成。
GetAuthorizationToken 请求的速率	每个支持的区域： 每秒 500 个	可以	您在当前区域中每秒可以发出的 GetAuthorizationToken 请求的最大数量。
GetDownloadUrlForLayer 请求的速率	每个支持的区域： 每秒 3,000 个	可以	您在当前区域中每秒可以发出的 GetDownloadUrlForLayer 请求的最大数量。提取镜像时，对于每个尚未缓存的镜像层调用一次 GetDownloadUrlForLayer API。如果您请求提高此 API 的配额，则还请查看 BatchGetImage 的使用情况。

姓名	默认值	是否可调整	说明
InitiateLayerUpload 请求的速率	每个支持的区域： 每秒 100 个	可以	您在当前区域中每秒可以发出的 InitiateLayerUpload 请求的最大数量。推送镜像时，对于每个尚未上传的镜像层，会调用一次 InitiateLayerUpload API。是否已上传镜像层由 BatchCheckLayerAvailability API 操作确定。
PutImage 请求的速率	每个支持的区域： 每秒 10 个	可以	您在当前区域中每秒可以发出的 PutImage 请求的最大数量。推送镜像并上传所有新的镜像层后，将调用一次 PutImage API，以创建或更新镜像清单以及与该镜像关联的标签。
UploadLayerPart 请求的速率	每个支持的区域： 每秒 500 个	可以	您在当前区域中每秒可以发出的 UploadLayerPart 请求的最大数量。推送映像时，每个新映像层均会分段上传，每个新映像层部分都会调用一次 UploadLayerPart API。
镜像扫描速率	每个支持的区域： 1 个	不可以	每 24 小时每个镜像的最大镜像扫描次数。

姓名	默认值	是否可调整	说明
已注册的存储库	每个支持的区域： 10,000 个	可以	您可以在当前区域中的此账户中创建的存储库的最大数量。
每个生命周期策略的规则数	每个受支持的区域： 50 个	不可以	生命周期策略中的最大规则数量
每个复制配置的规则数	每个受支持的区域： 10 个	不可以	复制配置中的最大规则数。
每个镜像的标签数	每个支持的区域： 1,000 个	不可以	每个镜像的最大标签数。
复制配置中所有规则的唯一目标	每个受支持的区域： 25 个	不可以	复制配置中所有规则的最大唯一目标数

在 AWS Management Console 中管理您的 Amazon ECR 服务配额

Amazon ECR 已与 Service Quotas 集成，后者是一项 AWS 服务，可让您从中心位置查看和管理您的配额。有关更多信息，请参阅 Service Quotas 用户指南中的[什么是 Service Quotas？](#)。

可使用 Service Quotas 轻松查找所有 Amazon ECR Service Quotas 的值。

查看 Amazon ECR Service Quotas (AWS Management Console)

1. 访问 <https://console.aws.amazon.com/servicequotas/>，打开 Service Quotas 控制台。
2. 在导航窗格中，选择 AWS 服务。
3. 从 AWS 服务列表中，搜索并选择 Amazon Elastic Container Registry (Amazon ECR)。

在 Service Quotas 列表中，您可以查看服务配额名称、应用的值 (如果该值可用)、AWS 原定设置配额以及配额值是否可调整。

4. 要查看有关服务限额的其他信息 (如描述)，请选择限额名称。

要请求提高限额，请参阅《服务限额用户指南》中的[请求提高限额](#)。

创建 CloudWatch 警报以监控 API 用量指标

Amazon ECR 提供了 CloudWatch 用量指标，这些指标与注册表身份验证、镜像推送和镜像提取操作所涉及的每个 API 的 AWS 服务配额相对应。在 Service Quotas 控制台中，您可以在图表上可视化您的用量，并配置警报以便在您的用量接近服务配额时提醒您。有关更多信息，请参阅[Amazon ECR 用量指标](#)。

使用以下步骤根据其中一个 Amazon ECR API 用量指标创建 CloudWatch 警报。

根据您的 Amazon ECR 用量配额创建警报 (AWS Management Console)

1. 访问 <https://console.aws.amazon.com/servicequotas/>，打开服务限额控制台。
2. 在导航窗格中，选择 AWS 服务。
3. 从 AWS 服务列表中，搜索并选择 Amazon Elastic Container Registry (Amazon ECR)。
4. 在 Service Quotas 列表中，选择要为其创建警报的 Amazon ECR 用量配额。
5. 在 Amazon CloudWatch Events 警报部分中，选择创建。
6. 对于警报阈值，选择要设置为警报值的适用配额值的百分比。
7. 对于警报名称，输入警报名称，然后选择创建。

Amazon ECR 故障排除

本章帮助您查找 Amazon ECR 的诊断信息，并提供常见问题和错误消息的故障排除步骤。

主题

- [对使用 Amazon ECR 时的 Docker 命令和问题进行故障排除](#)
- [排查 Amazon ECR 错误消息问题](#)

对使用 Amazon ECR 时的 Docker 命令和问题进行故障排除

在某些情况下，对 Amazon ECR 运行 Docker 命令可能会导致错误消息。一些常见错误消息和可能的解决办法解释如下。

主题

- [Docker 日志不包含预期的错误消息](#)
- [从 Amazon ECR 存储库提取镜像时，出现错误：“Filesystem Verification Failed”\(文件系统验证失败\)或“404: Image Not Found”\(404：找不到镜像\)](#)
- [从 Amazon ECR 提取镜像时，出现错误：“Filesystem Layer Verification Failed”\(文件系统分层验证失败\)](#)
- [推送到存储库时出现 HTTP 403 错误或“no basic auth credentials”\(没有基础级验证凭证\) 错误](#)

Docker 日志不包含预期的错误消息

要开始调试任何与 Docker 相关的问题，请先在主机实例上运行的 Docker 守护程序上打开 Docker 调试输出。如果您在 Amazon ECS 容器实例上使用从 Amazon ECR 提取的映像，请参阅亚马逊弹性容器服务开发人员指南中的[配置来自 Docker 守护程序的详细输出](#)。

从 Amazon ECR 存储库提取镜像时，出现错误：“Filesystem Verification Failed”(文件系统验证失败) 或“404: Image Not Found”(404：找不到镜像)

在 Docker 1.9 或更高版本中使用 `docker pull` 命令从 Amazon ECR 存储库提取镜像时，可能会收到错误 `Filesystem verification failed`。如果使用的是 1.9 之前的 Docker 版本，则可能收到错误 `404: Image not found`。

以下为一些可能的原因及它们的解释。

本地磁盘已满

如果运行 `docker pull` 命令的本地磁盘已满，那么对本地文件计算的 SHA-1 哈希值可能与 Amazon ECR 计算的 SHA-1 哈希值不同。确保本地磁盘有足够的剩余空间可存储所提取的 Docker 镜像。为腾出空间存储新镜像，可以删除旧镜像。使用 `docker images` 命令可查看所有已下载到本地的 Docker 镜像的列表及这些镜像的大小。

由于网络错误，客户端无法连接到远程存储库

调用 Amazon ECR 存储库需要 Internet 连接正常。验证网络设置，然后验证其他工具和应用程序是否可以访问 Internet 上的资源。如果在私有子网中对 Amazon EC2 实例运行 `docker pull`，请验证该子网是否具有连接至 Internet 的路由。可使用网络地址转换 (NAT) 服务器或托管的 NAT 网关。

目前，对 Amazon ECR 存储库的调用还要求通过您的公司防火墙访问 Amazon Simple Storage Service (Amazon S3)。如果贵企业或组织使用的是允许服务终端节点的防火墙软件或 NAT 设备，请确保当前区域的 Amazon S3 服务终端节点在允许范围内。

如果您通过 HTTP 代理使用 Docker，可以对 Docker 进行相应的代理设置。有关更多信息，请参阅 Docker 文档中的 [HTTP 代理](#)。

从 Amazon ECR 提取镜像时，出现错误：“Filesystem Layer Verification Failed”(文件系统分层验证失败)

您可能在使用 `image image-name not found` 命令提取镜像时收到错误 `docker pull`。如果检查 Docker 日志，可能会看到与下面类似的错误：

```
filesystem layer verification failed for digest sha256:2b96f...
```

此错误表示镜像的一个或多个层下载失败。以下为一些可能的原因及它们的解释。

您正在使用旧版本的 Docker

在使用低于 1.10 的 Docker 版本时，有少数情况会出现此错误。请将您的 Docker 客户端升级至 1.10 或更高版本。

您的客户端遇到网络错误或磁盘错误

如前文对 `Filesystem verification failed` 消息的讨论中所述，磁盘已满或网络问题可能会导致一个或多个层无法下载。请遵循上述建议确保您的文件系统未滿，并且您在网络中有对 Amazon S3 的访问权限。

推送到存储库时出现 HTTP 403 错误或“no basic auth credentials”(没有基础级验证凭证) 错误

有时，即使您已使用 `aws ecr get-login-password` 命令成功通过 Docker 身份验证，也可能会从 `docker push` 或 `docker pull` 命令收到 HTTP 403 (Forbidden) 错误或者错误消息 `no basic auth credentials`。以下是此问题的一些已知的原因：

您已验证到其他区域

身份验证请求与特定的区域相关联，不能跨区域使用。例如，如果您从美国西部 (俄勒冈) 获得授权令牌，不能使用它对您在东部 (弗吉尼亚北部) 的存储库进行身份验证。要解决此问题，请确保您已从存储库所在的同一区域检索了身份验证令牌。有关更多信息，请参阅 [the section called “注册表身份验证”](#)。

您已进行身份验证以推送到您没有权限的存储库

您没有必要的权限来推送到存储库。有关更多信息，请参阅 [Amazon ECR 中的私有存储库政策](#)。

您的令牌已过期。

对于使用 `GetAuthorizationToken` 操作获取的令牌，默认授权令牌有效期为 12 小时。

wincred 凭证管理器中的错误

某些版本的适用于 Windows 的 Docker 使用名为 wincred 的凭证管理器，但它无法正确处理由 `aws ecr get-login-password` 生成的 Docker 登录命令 (有关更多信息，请参阅 <https://github.com/docker/docker/issues/22910>)。可以运行作为输出的 Docker 登录命令，但如果尝试推送或提取镜像，这些命令会失败。修复这个错误的方法是，对于从 `aws ecr get-login-password` 输出的 Docker 登录命令，删除其注册表参数中的 `https://` 方案。如下所示为不带 HTTPS 方案的 Docker 登录命令示例。

```
docker login -u AWS -p <password> <aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

排查 Amazon ECR 错误消息问题

在某些情况下，您通过 Amazon ECR 控制台发起的 API 调用或 AWS CLI 退出时会显示错误消息。一些常见错误消息和可能的解决办法解释如下。

HTTP 429：请求太多或 ThrottleException

您可能会收到一个429: Too Many Requests或多个 Amazon ECR 操作或 API 调用的 ThrottleException 错误或错误。这表示由于您在短时间内重复调用 Amazon ECR 中的单个终端节点，您的请求已受限制。单个用户在一段时间内，调用单个终端节点的次数超过特定阈值时，就会产生限制。

Amazon ECR 中的每个 API 操作都有一个与之相关的速率限制。例如，[GetAuthorizationToken](#) 操作的限制为每秒 20 个事务 (TPS)，允许高达 200 TPS 的突增。在每个区域，每个账户会收到一个可存储多达 200 点 GetAuthorizationToken 积分的存储桶。这些积分以每秒 20 点的速度补充。如果您的存储桶有 200 点积分，则可实现每秒 200 个 GetAuthorizationToken API 事务 (持续一秒)，然后无限期地维持每秒 20 个事务。有关 Amazon ECR API 速率限制的更多信息，请参阅 [Amazon ECR 服务配额](#)。

要处理限制错误，请在代码中实施增量退避重试函数。有关更多信息，请参阅 AWS SDK 和工具参考指南中的 [重试行为](#)。另一种选择是请求提高速率限制，您可以使用 Service Quotas 控制台执行此操作。有关更多信息，请参阅在 [AWS Management Console](#) 中管理您的 [Amazon ECR 服务配额](#)。

HTTP 403：“User [arn] is not authorized to perform [operation]”(用户 [arn] 没有执行 [operation] 的权限)

尝试通过 Amazon ECR 执行操作时，您可能会收到以下错误：

```
$ aws ecr get-login-password
A client error (AccessDeniedException) occurred when calling the GetAuthorizationToken
operation:
    User: arn:aws:iam::account-number:user/username is not authorized to perform:
    ecr:GetAuthorizationToken on resource: *
```

这表示您的用户没有获得使用 Amazon ECR 的权限，或者这些权限设置不正确。尤其是在对 Amazon ECR 执行操作时，请验证是否已授予用户访问该存储库的权限。有关创建和验证 Amazon ECR 权限的更多信息，请参阅 [适用于 Amazon Elastic Container Registry 的 Identity and Access Management](#)。

HTTP 404：“Repository Does Not Exist”(存储库不存在) 错误

如果您指定了当前不存在的 Docker Hub 存储库，Docker Hub 会自动创建存储库。但在使用 Amazon ECR 时，新存储库必须在使用前显式创建。这会防止意外创建新存储库 (例如，由于输入错误)，也可确保为所有新存储库明确分配适当的安全访问策略。有关创建存储库的更多信息，请参阅 [Amazon ECR 私有存储库](#)。

错误：无法从非 TTY 设备执行交互式登录

如果您收到错误 `Cannot perform an interactive login from a non TTY device`，则以下故障排除步骤应该会有所帮助。

- 确认您使用的是 AWS CLI 版本 2，并且您的系统上没有 AWS CLI 版本 1 的冲突版本。有关更多信息，请参阅[安装或更新 AWS CLI 的最新版本](#)。
- 确认您已 AWS CLI 使用有效的凭据配置了您的证书。有关更多信息，请参阅[安装或更新 AWS CLI 的最新版本](#)。
- 验证您的 AWS CLI 命令语法是否正确。

文档历史记录

下表列出了自 Amazon ECR 上一次发布以来对文档所做的重要更改。我们还经常更新文档来处理发送给我们的反馈意见。

更改	描述	日期
向中国区域添加了跨区域和跨账户复制	Amazon ECR 为中国区域增加了筛选复制哪些存储库的支持。	2024 年 5 月 15 日
添加了 GitLab 容器注册表以查看缓存规则	Amazon ECR 增加了对为 GitLab 容器注册表创建拉取缓存规则的支持。	2024 年 5 月 8 日
Amazon ECR 生命周期策略更新，增加了对使用通配符的支持	Amazon ECR 增加了对在生命周期策略中使用通配符的支持，方法是在生命周期策略规则中使用 <code>tagPatternList</code> 参数。有关更多信息，请参阅 在 Amazon ECR 中使用生命周期策略自动清理图像 。	2023 年 12 月 18 日
Amazon ECR 存储库创建模板	Amazon ECR 增加了对存储库创建模板的支持。有关更多信息，请参阅 用于控制在缓存拉取操作期间创建的存储库的模板 。	2023 年 11 月 15 日
Amazon ECR 缓存提取添加了针对经过身份验证的上游注册表的支持	Amazon ECR 增加了对使用需要对拉取缓存规则进行身份验证的上游注册表的支持。有关更多信息，请参阅 将上游注册表与 Amazon ECR 私有注册表同步 。	2023 年 11 月 15 日
AWSECRPullThroughCache_ServiceRolePolicy - 更新了现有策略	Amazon ECR 将新权限添加到 <code>AWSECRPullThroughCache_ServiceRolePolicy</code> 策略。这些权限允许 Amazon ECR 检索 Secrets Manager 密钥的加密内容。当使用缓存提取规则来缓存需要身份验证的上游注册表中的映像时，需要此类权限。	2023 年 11 月 15 日
Amazon ECR 镜像签名	Amazon ECR，并 AWS Signer 增加了对使用公证客户端创建和推送容器镜像签名的支持。有关更多信息，请参阅 对存储在 Amazon ECR 私有存储库中的图像进行签名 。	2023 年 6 月 6 日

更改	描述	日期
增加了 Kubernetes 容器注册表以提取缓存规则	Amazon ECR 增加了对创建 Kubernetes 容器注册表的缓存提取规则的支持。有关更多信息，请参阅 将上游注册表与 Amazon ECR 私有注册表同步 。	2023 年 6 月 1 日
Amazon ECR 增强扫描持续时间支持	Amazon Inspector 增加了对启用增强扫描时监控存储库的持续时间设置支持。有关更多信息，请参阅 在 Amazon Inspector 中更改图像的增强扫描时长 。	2022 年 6 月 28 日
Amazon ECR 向亚马逊发送存储库提取计数指标 CloudWatch	Amazon ECR 向亚马逊 CloudWatch 发送存储库提取计数指标。有关更多信息，请参阅 Amazon ECR 存储库指标 。	2022 年 1 月 6 日
扩展了复制支持	Amazon ECR 增加了要复制的存储库进行筛选的支持。有关更多信息，请参阅 在 Amazon ECR 中复制私有镜像 。	2021 年 9 月 21 日
AWS Amazon ECR 的托管策略	Amazon ECR 增加了 AWS 托管策略的文档。有关更多信息，请参阅 AWS Amazon 弹性容器注册表的托管策略 。	2021 年 6 月 24 日
跨区域和跨账户复制	Amazon ECR 增加了对配置私有注册表复制设置的支持。有关更多信息，请参阅 Amazon ECR 中的私有注册表设置 。	2020 年 12 月 8 日
OCI 构件支持	Amazon ECR 增加了对推送和提取 Open Container Initiative (OCI) 构件的支持。新的参数 <code>artifactMediaTypes</code> 添加到 <code>DescribeImages</code> API 响应中以指示工件类型。 有关更多信息，请参阅 将 Helm 图表推送到 Amazon ECR 私有存储库 。	2020 年 8 月 24 日
静态加密	Amazon ECR 增加了对结合使用服务器端加密和 AWS Key Management Service (AWS KMS) 中所存储客户托管密钥配置加密的支持。 有关更多信息，请参阅 静态加密 。	2020 年 7 月 29 日

更改	描述	日期
多架构镜像	<p>Amazon ECR 添加了对创建和推送用于多架构镜像的 Docker 清单列表的支持。</p> <p>有关更多信息，请参阅 将多架构映像推送到 Amazon ECR 私有存储库。</p>	2020 年 4 月 28 日
Amazon ECR 使用情况指标	<p>Amazon ECR 添加了 CloudWatch 使用量指标，可让您了解账户的资源使用情况。您还可以从 Service Quotas 控制台 CloudWatch 和 Service Quotas 控制台创建 CloudWatch 警报，以便在使用量接近已申请的服务配额时收到提醒。</p> <p>有关更多信息，请参阅 Amazon ECR 用量指标。</p>	2020 年 2 月 28 日
更新了 Amazon ECR 服务配额	<p>更新了 Amazon ECR 服务配额，以包含每个 API 的配额。</p> <p>有关更多信息，请参阅 Amazon ECR 服务配额。</p>	2020 年 2 月 19 日
已添加 get-login-password 命令	<p>增加了对 get-login-password 的支持，它提供了一个简单而安全的方法来检索授权令牌。</p> <p>有关更多信息，请参阅 使用授权令牌。</p>	2020 年 2 月 4 日
镜像扫描	<p>增加了对镜像扫描的支持，这有助于识别容器镜像中的软件漏洞。Amazon ECR 使用开源 CoreOS Clair 项目中的常见漏洞和披露 (CVE) 数据库，并为您提供扫描发现结果的列表。</p> <p>有关更多信息，请参阅 扫描图像以查找 Amazon ECR 中的软件漏洞。</p>	2019 年 10 月 24 日
VPC 终端节点策略	<p>增加了对在 Amazon ECR 接口 VPC 终端节点上设置 IAM policy 的支持。</p> <p>有关更多信息，请参阅 为 Amazon ECR VPC 端点创建终端节点策略。</p>	2019 年 9 月 26 日

更改	描述	日期
镜像标签可变性	<p>增加了对将存储库配置为不可变的支持，以防止覆盖镜像标签。</p> <p>有关更多信息，请参阅 防止在 Amazon ECR 中覆盖图片标签。</p>	2019 年 7 月 25 日
接口 VPC 端点 (AWS PrivateLink)	<p>增加了对配置由提供支持的接口 VPC 终端节点的支持 AWS PrivateLink。这能让您在您的 VPC 和 Amazon ECR 之间创建私有连接，而无需通过 Internet、NAT 实例、VPN 连接或 AWS Direct Connect 进行访问。</p> <p>有关更多信息，请参阅 Amazon ECR 接口 VPC 终端节点 ()AWS PrivateLink。</p>	2019 年 1 月 25 日
为资源加标签	<p>Amazon ECR 增加了对为存储库添加元数据标签的支持。</p> <p>有关更多信息，请参阅 在 Amazon ECR 中标记私有存储库。</p>	2018 年 12 月 18 日
Amazon ECR 名称更改	<p>重命名 Amazon Elastic Container Registry (以前名为 Amazon EC2 Container Registry)。</p>	2017 年 11 月 21 日
生命周期策略	<p>Amazon ECR 生命周期策略使您能够指定存储库中镜像的生命周期管理。</p> <p>有关更多信息，请参阅 在 Amazon ECR 中使用生命周期策略自动清理图像。</p>	2017 年 10 月 11 日
Amazon ECR 支持 Docker Image Manifest 2、Schema 2	<p>Amazon ECR 现已支持 Docker Image Manifest V2 Schema 2 (与 Docker 版本 1.10 和更高版本配合使用)</p> <p>有关更多信息，请参阅 Amazon ECR 中支持容器镜像清单格式。</p>	2017 年 1 月 27 日
Amazon ECR 正式发布	<p>Amazon Elastic Container Registry (Amazon ECR) 是一项安全、可扩展和可靠的 AWS 托管 Docker 注册服务。</p>	2015 年 12 月 21 日

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。