Amazon ECR

用户指南

API 版本 2015-09-21

Amazon ECR: 用户指南

Table of Contents

什么是 Amazon Elastic Container Registry?

Amazon Elastic Container Registry (Amazon ECR) 是一种 AWS 托管容器映像注册表服务，安全、可扩展且可靠。Amazon ECR 使用 AWS 支持具有基于资源的权限的私有容器映像存储库。IAM这样，指定用户或 Amazon EC2 实例就可以访问您的容器存储库和映像。您可以使用首选 CLI 来推送、拉取和管理 Docker 映像、开放容器计划（OCI）映像和兼容 OCI 的构件。

> Note
>
> Amazon ECR 还支持公有容器映像存储库。有关更多信息，请参阅 Amazon ECR 公共用户指南 中的什么是 Amazon ECR 公共功能。

AWS 容器服务团队在 GitHub 上维护着公有路线图。该路线图包含有关团队工作的信息，并允许所有 AWS 客户提供直接反馈。有关更多信息，请参阅 AWS 容器路线图.

的组成部分Amazon ECR

Amazon ECR 包含以下组件：

注册表

我们为每个 Amazon ECR 账户均提供了一个 AWS 镜像仓库；您可以在镜像仓库中创建镜像存储库，并在其中存储镜像。有关更多信息，请参阅 Amazon ECR 私有注册表 (p. 13).

授权令牌

您的客户端必须作为 Amazon ECR 用户向 AWS 注册表进行身份验证，然后才能推送和拉取映像。有关更多信息，请参阅 私有注册表身份验证 (p. 13).

存储库

映像存储库包含您的 Docker 映像、开放容器计划（OCI）映像和与 OCI 兼容的构件。Amazon ECR有关更多信息，请参阅 Amazon ECR 私有存储库 (p. 20).

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映像的访问。有关更多信息，请参阅 存储库策略 (p. 23).

映像

您可以对存储库推送和拉取容器映像。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 在 Amazon ECS 中使用 Amazon ECR 映像 (p. 58) 和 在 Amazon EKS 中使用 Amazon ECR 映像 (p. 59).

的功能Amazon ECR

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 生命周期策略 (p. 42).

- 映像扫描有助于识别容器映像中的软件漏洞。每个存储库均可配置为 scan on push （推送时扫描）。这可确保将每个新映像推送到存储库。然后，您可以检索映像扫描的结果。有关更多信息，请参阅 映像扫描 (p. 53).
- 通过跨区域和跨账户复制，您可以更轻松地将映像置于需要它们的位置。这配置为注册表设置，并且基于每个区域。有关更多信息，请参阅 私有注册表设置 (p. 15).

如何开始使用 Amazon ECR

要使用 Amazon ECR，必须设置以安装 AWS Command Line Interface 和 Docker。有关更多信息，请参阅 使用 进行设置Amazon ECR (p. 3) 和 将 Amazon ECR 与 AWS CLI 结合使用 (p. 7).

定价Amazon ECR

使用 Amazon ECR，您只需为存储库中存储的数据量以及映像推送和提取的数据传输付费。有关更多信息，请参阅 Amazon ECR 定价.

使用 进行设置Amazon ECR

如果已注册 AWS 并已在使用 Amazon Elastic Container Service (Amazon ECS) 或 Amazon Elastic Kubernetes Service (Amazon EKS)，那么您与使用 Amazon ECR. 已近在咫尺。这两种服务的设置过程相似，因为 Amazon ECR 是这些服务的扩展。要在 AWS CLI 中使用 Amazon ECR，必须使用支持最新 AWS CLI 功能的 Amazon ECR 版本。如果在 Amazon ECR 中没有看到对 AWS CLI 功能的支持，可以升级到最新版本。有关更多信息，请参阅 http://www.amazonaws.cn/cli/.

按照以下任务完成设置，以便首次将容器映像推送到 Amazon ECR 如果您已完成以下任何步骤，可以跳过这些步骤并继续执行下一步。

Sign up for AWS

当您注册 AWS 时，您的 AWS 账户会自动注册所有服务，包括 Amazon ECR. 您只需为使用的服务付费。

如果您已有 AWS 账户，请跳到下一个任务。如果您还没有 AWS 账户，请使用以下步骤创建。

创建 AWS 账户

1. 打开 https://portal.amazonaws.cn/billing/signup。
2. 按照屏幕上的说明进行操作。

   在注册时，您将接到一通电话，要求您使用电话键盘输入一个验证码。

请记下您的 AWS 账号，因为在下一个任务中您会用到它。

创建 IAM 用户

AWS 中的服务（例如 Amazon ECR）要求您在访问时提供凭证，以便服务可以确定您是否有权限访问其资源。控制台要求您的密码。您可以为您的 AWS 账户创建访问密钥以访问命令行界面或 API。但是，我们不建议您使用 AWS 账户的凭证访问 AWS，而建议您改用 AWS Identity and Access ManagementIAM 创建 IAM 用户，然后将该用户添加到具有管理权限的 IAM 组或授予此用户管理权限。然后，您就可以使用专门的 URL 和该 AWS 用户的凭证来访问 IAM

如果您已注册 AWS 但尚未为自己创建一个 IAM 用户，则可以使用 IAM 控制台自行创建。

自行创建管理员用户并将该用户添加到管理员组（控制台）

1. 通过选择 根用户，然后输入您的 AWS 账户的电子邮件地址，以账户拥有者身份登录到 IAM 控制台。在下一页上，输入您的密码。

   Note

   强烈建议您遵守以下使用 **`Administrator`** IAM 用户的最佳实践，妥善保存根用户凭证。只在执行少数账户和服务管理任务时才作为根用户登录。
2. 在导航窗格中，选择用户，然后选择添加用户。
3. 对于 User name (用户名)，输入 **`Administrator`**。
4. 选中 AWS 管理控制台 访问 旁边的复选框。然后选择自定义密码，并在文本框中输入新密码。

5. （可选）默认情况下，AWS 要求新用户在首次登录时创建新密码。您可以清除 User must create a new password at next sign-in (用户必须在下次登录时创建新密码) 旁边的复选框以允许新用户在登录后重置其密码。
6. 选择下一步: 权限。
7. 在设置权限下，选择将用户添加到组。
8. 选择创建组。
9. 在 Create group (创建组) 对话框中，对于 Group name (组名称)，输入 **`Administrators`**。
10. 选择 Filter policies (筛选策略)，然后选择 AWS managed-job function (AWS 托管的工作职能) 以筛选表内容。
11. 在策略列表中，选中 AdministratorAccess 的复选框。然后选择 Create group (创建组)。

    Note

    您必须先激活 IAM 用户和角色对账单的访问权限，然后才能使用 `AdministratorAccess` 权限访问 AWS Billing and Cost Management 控制台。为此，请按照"向账单控制台委派访问权限"教程第 1 步中的说明进行操作。
12. 返回到组列表中，选中您的新组所对应的复选框。如有必要，选择 Refresh 以在列表中查看该组。
13. 选择下一步: 标签。
14. （可选）通过以键值对的形式附加标签来向用户添加元数据。有关在 IAM 中使用标签的更多信息，请参阅 IAM 用户指南 中的标记 IAM 实体。
15. 选择 Next: Review (下一步: 审核) 以查看要添加到新用户的组成员资格的列表。如果您已准备好继续，请选择 Create user。

您可使用此相同的流程创建更多的组和用户，并允许您的用户访问 AWS 账户资源。要了解有关使用策略限制用户对特定 AWS 资源的权限的信息，请参阅访问管理和示例策略。

要以该新 IAM 用户的身份登录，请从 AWS 控制台注销，然后使用以下 URL，其中 your_aws_account_id 是您不带连字符的 AWS 账号（例如，如果您的 AWS 账号是 `1234-5678-9012`，则您的 AWS 账户 ID 是 `123456789012`):

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

输入您刚创建的 IAM 用户名和密码。登录后，导航栏显示 your_user_name @ your_aws_account_id".

如果您不希望您的登录页面 URL 包含 AWS 账户 ID，可以创建账户别名。从 IAM 控制面板中，选择 Customize（自定义），然后输入 Account Alias（账户别名），例如您的公司名称。有关更多信息，请参阅中的 AWS 账户 ID 及其别名。IAM 用户指南

要在创建账户别名后登录，请使用以下 URL：

```
https://your_account_alias.signin.aws.amazon.com/console/
```

要为您的账户验证 IAM 用户的登录链接，请打开 IAM 控制台并在控制面板的 IAM users sign-in link（IAM 用户登录链接）下进行检查。

有关 IAM 的更多信息，请参阅 AWS Identity and Access Management 用户指南.

通过 Amazon ECR 开始使用 AWS 管理控制台

通过在 Amazon ECR 控制台中创建存储库开始使用 Amazon ECR Amazon ECR 控制台可以引导您完成开始创建第一个存储库的过程。

开始之前，请确保您已完成中的步骤。使用 进行设置Amazon ECR (p. 3).

创建映像存储库

存储库是您存储 中 Docker 或 Open Container Initiative (OCI) 映像的地方。Amazon ECR. 当您每次在 Amazon ECR 中推送或拉取映像时，您将指定存储库和注册表位置，以告知将映像推送到哪个位置或从哪个位置拉取映像。

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/。
2. 选择开始使用.
3. 对于 Tag imutability （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映像标签。有关更多信息，请参阅 映像标签可变性 (p. 53).
4. 对于 Scan on push （推送扫描），选择存储库的映像扫描设置。配置为在推送时进行扫描的存储库将在每次推送映像时启动映像扫描，否则需要手动启动映像扫描。有关更多信息，请参阅 映像扫描 (p. 53).
5. 选择 Create repository.

构建、标记和推送 Docker 映像

在向导的此部分中，您使用 Docker CLI 标记现有本地映像 (您从 Dockerfile 构建或从另一个注册表中拉取的映像，例如 Docker Hub)，然后将标记的映像推送到 Amazon ECR 注册表。有关使用 Docker CLI 的更多详细步骤，请参阅将 Amazon ECR 与 AWS CLI 结合使用 (p. 7).

1. 选择您创建的存储库，然后选择 View push commands （查看推送命令）以查看将映像推送到新存储库的步骤。
2. 检索 docker login 命令，此命令可用于通过将控制台中的 aws ecr get-login 命令粘贴到终端窗口中来对注册表验证 Docker 客户端的身份。

   Note

   AWS CLI 从版本 1.9.15 开始提供 get-login 命令；但对于较新的 Docker 版本（17.06 或更高版本），我们建议使用 1.11.91 版或更高版本。可以使用 aws --version 命令查看 AWS CLI 版本。如果您使用的是 Docker 17.06 或更高版本，请在 `get-login` 后包含 `--no-include-email` 选项。如果收到 `Unknown options: --no-include-email` 错误，请安装最新版本的 AWS CLI。有关更多信息，请参阅 AWS Command Line Interface 用户指南 中的 安装 AWS 命令行界面。
3. 运行上一步中返回的 docker login 命令。此命令提供一个在 12 小时内有效的授权令牌。

   Important

   在您执行此 docker login 命令时，系统上的其他用户可以在进程列表 (ps -e) 显示中看到该命令字符串。由于 docker login 命令包含身份验证凭证，因此系统上的其他用户可按此方式查看凭证会带来风险。他们可能会使用凭证获取对您的存储库的推送和拉取访问权限。如果您所在的系统不安全，则应考虑此风险，并通过省略 `-p` *`password`* 选项并在系统提示时输入密码来以交互方式登录。

4. （可选）如果您有要让映像推送的 Dockerfile，请为新存储库构建并标记映像。将控制台中的 docker build 命令粘贴到终端窗口。确定您与您的 Dockerfile 位于同一目录中。
5. 通过将控制台中的 docker tag 命令粘贴到终端窗口中来为 ECR 注册表和新存储库标记映像。此控制台命令假设您的映像是通过上一步中的 Dockerfile 构建得来的。如果您未通过 Dockerfile 构建映像，请将 *repository*:latest 的第一个实例更换为要推送的本地映像的映像 ID 或映像名称。
6. 通过将 docker push 命令粘贴到终端窗口中来将新标记的映像推送到 ECR 存储库。
7. 选择 Close。

将 Amazon ECR 与 AWS CLI 结合使用

以下步骤将指导您完成首次使用 Docker CLI 和 将容器映像推送到私有Amazon ECR存储库所需的步骤AWS CLI。

有关可用于管理 AWS 资源的其他工具（包括不同的 AWS SDKs、IDE 工具包和 Windows PowerShell 命令行工具）的更多信息，请参阅 www.amazonaws.cnhttp：///tools/。

Prerequisites

开始之前，请确保您已完成中的步骤。使用 进行设置Amazon ECR (p. 3).

如果您尚未安装最新 AWS CLI 和 Docker 并且未准备好使用，请使用以下步骤来安装这两个工具。

安装 AWS CLI

可以使用 AWS 命令行工具，在系统的命令行中发出命令来执行 Amazon ECR 和其他 AWS 任务。与使用控制台相比，此方法更快、更方便。命令行工具也非常适用于构建执行 AWS 任务的脚本。

要在 AWS CLI 中使用 Amazon ECR，请安装最新的 AWS CLI 版本（Amazon ECR 中从 AWS CLI 版本开始提供 1.9.15). 功能）。可以使用 AWS CLI 命令查看 aws --version 版本。有关安装 AWS CLI 或升级到最新版本的信息，请参阅 中的https://docs.amazonaws.cn/cli/latest/userguide/install-cliv2.html安装 AWS CLI 版本 AWS Command Line Interface 用户指南2。

安装 Docker

Docker 可用于许多不同的操作系统，包括大多数现代 Linux 发行版（如 Ubuntu），甚至包括 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。如果您已使用 Amazon EC2，则可启动 Amazon Linux 2 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 映像 (p. 8).

在 Amazon EC2 实例上安装 Docker

1. 使用 Amazon Linux 2 AMI. 启动实例。有关更多信息，请参阅 https://docs.amazonaws.cn/AWSEC2/latest/UserGuide/launching-instance.html 中的Amazon EC2 用户指南（适用于 Linux 实例）启动实例.
2. 连接到您的 实例。有关更多信息，请参阅 https://docs.amazonaws.cn/AWSEC2/latest/UserGuide/AccessingInstances.html 中的Amazon EC2 用户指南（适用于 Linux 实例）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 amazon-linux-extras 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 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 映像

在本节中，您将创建简单 Web 应用程序的 Docker 映像，并在本地系统或 EC2 实例上测试此映像，然后将此映像推送至容器注册表（如 Amazon ECR 或 Docker Hub），以便能够在 ECS 任务定义中使用它。

创建简单 Web 应用程序的 Docker 映像

1. 创建一个名为 的文件Dockerfile。 Dockerfile 是一个清单，描述要用于 Docker 映像的基本映像以及要在其中安装并运行的内容。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 ubuntu:18.04

# Install dependencies
RUN apt-get update && \
 apt-get -y install apache2

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

# Configure apache
RUN echo '. /etc/apache2/envvars' > /root/run_apache.sh && \
 echo 'mkdir -p /var/run/apache2' >> /root/run_apache.sh && \
 echo 'mkdir -p /var/lock/apache2' >> /root/run_apache.sh && \
 echo '/usr/sbin/apache2 -D FOREGROUND' >> /root/run_apache.sh && \
 chmod 755 /root/run_apache.sh

EXPOSE 80
```

```
CMD /root/run_apache.sh
```

此 Dockerfile 使用 Ubuntu 18.04 映像。RUN 说明将更新程序包缓存，安装一些适用于 Web 服务器的软件包，然后编写“Hello World!” 内容发送到 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 映像。

   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. 运行 docker images 以验证是否已正确创建映像。

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

输出：

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

5. 运行新构建的映像。-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 server's 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，则这是服务器的 Public DNS（公有 DNS）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/.
   - 如果您在 Windows 或docker-machine计算机上使用 macOS ，请使用 VirtualBox 命令查找托管 Docker 的 docker-machine ip VM 的 IP 地址，替换 *machine-name* 替换为您正在使用的 Docker 计算机的名称。

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

   您应看到一个包含“Hello World!”的网页。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 AWS CLI 后，向默认注册表验证 Docker CLI 的身份。这样一来，docker 命令可以通过 Amazon ECR. 推送和拉取镜像。AWS CLI 提供 get-login-password 命令来简化身份验证过程。

要使用 get-login-password 针对 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 命令行界面。

- 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-ECRLoginCommand (适用于 Windows PowerShell 的 AWS 工具)

```
(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-world` 的存储库，稍后将推送 `hello-world:latest` 映像到这里。要创建存储库，请运行以下命令：

```
aws ecr create-repository \
    --repository-name hello-world \
    --image-scanning-configuration scanOnPush=true \
    --region us-east-1
```

步骤 4：推送映像到 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。您使用 docker CLI 推送映像，但必须满足一些先决条件才能正常工作：

- 安装最低版本的 docker1.7
- 已使用 Amazon ECR 配置 docker login. 授权令牌。
- 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，即可将映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映像，以识别要标记和推送的映像。

   ```
   docker images
   ```

   输出：

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

2. 标记映像并推送到存储库。

```
docker tag hello-world:latest aws_account_id.dkr.ecr.us-east-1.amazonaws.com/hello-world:latest
```

3. 推送映像。

```
docker push aws_account_id.dkr.ecr.us-east-1.amazonaws.com/hello-world:latest
```

输出：

```
The push refers to a repository [aws_account_id.dkr.ecr.us-east-1.amazonaws.com/hello-world] (len: 1)
e9ae3c220b23: Pushed
a6785352b25c: Pushed
0998bf8fb9e9: Pushed
0a85502c06c9: Pushed
latest: digest: sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b size: 6774
```

步骤 5：从 拉取映像Amazon ECR

在推送镜像到 Amazon ECR 存储库后，可以从其他位置拉取该镜像。可使用 docker CLI 拉取映像，但必须满足以下几个先决条件才能正常使用：

- 安装最低版本的 docker1.7
- 已使用 Amazon ECR 配置 docker login. 授权令牌。
- Amazon ECR 存储库存在且用户有从该存储库拉取的权限。

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

```
docker pull aws_account_id.dkr.ecr.us-east-1.amazonaws.com/hello-world:latest
```

输出：

```
latest: Pulling from hello-world
0a85502c06c9: Pull complete
0998bf8fb9e9: Pull complete
a6785352b25c: Pull complete
e9ae3c220b23: Pull complete
Digest: sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b
Status: Downloaded newer image for aws_account_id.dkr.ecr.us-east-1.amazonaws.com/hello-world:latest
```

步骤 6：删除映像

如果您不再需要一个存储库中的某个映像，则可以使用 batch-delete-image 命令将其删除。要删除映像，您必须指定它所在的存储库，并指定映像的 `imageTag` 或 `imageDigest` 值。以下示例删除 `hello-world` 存储库中映像标签为 `latest`. 的映像。

```
aws ecr batch-delete-image \
      --repository-name hello-world \
```

```
      --image-ids imageTag=latest
```

输出：

```
{
    "failures": [],
    "imageIds": [
        {
            "imageTag": "latest",
            "imageDigest":
 "sha256:215d7e4121b30157d8839e81c4e0912606fca105775bb0636b95aed25f52c89b"
        }
    ]
}
```

步骤 7：删除存储库

如果您不再需要整个存储库中的所有映像，您可以删除存储库。默认情况下，您不能删除包含映像的存储库；但是，`--force` 标记允许此操作。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-world \
      --force
```

Amazon ECR 私有注册表

Amazon ECR 私有注册表在高度可用和可扩展的架构中托管您的容器映像。您可以使用私有注册表管理由 Docker 和开放容器计划 （OCI） 映像和构件组成的私有映像存储库。每个AWS账户都有一个默认私有Amazon ECR注册表。有关 Amazon ECR 公共注册表的更多信息，请参阅 公共用户指南 中的 Amazon Elastic Container Registry 公共注册表。

私有注册表概念

- 默认私有注册表的 URL 是 `https://aws_account_id.dkr.ecr.region.amazonaws.com`。
- 默认情况下，您的账户对私有注册表中的存储库具有读写访问权限。但是， IAM 用户需要调用 Amazon ECR APIs 的权限，以及向您的私有存储库推送或拉取映像的权限。 Amazon ECR 提供了多个托管策略来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 (p. 72).
- 您必须向私有注册表验证 Docker 客户端，以便使用 docker push 和 docker pull 命令在该注册表中的存储库之间推送和拉取映像。有关更多信息，请参阅私有注册表身份验证 (p. 13).
- 可以使用IAM用户访问策略和存储库策略控制私有存储库。有关存储库策略的更多信息，请参阅存储库策略 (p. 23).
- 您的私有注册表中的存储库可以跨您自己的私有注册表中的区域复制，也可以通过为您的私有注册表配置复制来跨单独的账户复制。有关更多信息，请参阅私有映像复制 (p. 39).

私有注册表身份验证

您可以使用 AWS 管理控制台、 AWS CLI或 AWS SDKs 创建和管理私有存储库。也可以使用这些方法对映像执行某些操作，例如列出或删除映像。这些客户端使用标准 AWS 身份验证方法。即使您可以使用 Amazon ECR API 推送和拉取映像，您也更有可能使用 Docker CLI 或特定于语言的 Docker 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 推送和拉取请求进行身份验证和授权。

以下部分详细介绍的注册表身份验证方法可供使用。

使用Amazon ECR凭证辅助程序

Amazon ECR 提供了一个 Docker 凭证辅助程序，该辅助程序使在将映像推送和拉取到 时存储和使用 Docker 凭证变得更加轻松Amazon ECR。有关安装和配置步骤，请参阅 Amazon ECR Docker 凭据辅助程序.

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委托人的权限范围匹配。身份验证令牌用于访问您的Amazon ECR委托人有权访问且有效期为 12 小时的任何IAM注册表。要获取授权令牌，您必须使用 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` 命令可以通过检索和解码授权令牌来简化此操作，然后您可以将授权令牌传送到 docker login 命令中进行身份验证。

使用 get-login-password 向 Amazon ECR 私有注册表验证 Docker

要使用 get-login-password 针对 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 命令行界面。

- 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-ECRLoginCommand (适用于 Windows PowerShell 的 AWS 工具)

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

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

使用 1.17.10 之前的 AWS CLI 版本时，get-login 命令可用于对您的 Amazon ECR 注册表进行身份验证。可以使用 aws --version 命令查看 AWS CLI 版本。

1. 运行 aws ecr get-login 命令。以下示例适用于与创建请求的账户关联的默认注册表。要访问其他账户注册表，请使用 `--registry-ids` *aws_account_id* 选项。有关更多信息，请参阅 AWS CLI Command Reference 中的 get-login。

   ```
   aws ecr get-login --region region --no-include-email
   ```

   结果输出是 docker login 命令，此命令可用于对 Amazon ECR 注册表验证 Docker 客户端。

   ```
   docker login -u AWS -p password https://aws_account_id.dkr.ecr.region.amazonaws.com
   ```

2. 将 docker login 命令复制并粘贴到终端，授权您的 Docker CLI 访问注册表。此命令提供一个授权令牌，此令牌在 12 小时内对指定注册表有效。

   > **Note**
   >
   > 如果使用的是 Windows PowerShell，复制并粘贴这样的长字符串将不起作用。请使用以下命令。

   ```
   Invoke-Expression -Command (Get-ECRLoginCommand -Region region).Command
   ```

   > **Important**
   >
   > 在您执行此 docker login 命令时，系统上的其他用户可以在进程列表 (ps -e) 显示中看到该命令字符串。由于 docker login 命令包含身份验证凭证，因此系统上的其他用户可按此方式查看凭证会带来风险。他们可能会使用凭证获取对您的存储库的推送和拉取访问权限。如果您所在的系统不安全，则应考虑此风险，并通过省略 `-p` *password* 选项并在系统提示时输入密码来以交互方式登录。

使用 HTTP API 身份验证

Amazon ECR 支持 Docker 注册表 HTTP API. 但是，由于 Amazon ECR 属于私有镜像仓库，因此您必须为每个 HTTP 请求提供授权令牌。您可以通过使用 `-H` 的 curl 选项来添加 HTTP 授权标头，以传递由 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 AWS CLI 检索授权令牌并将其设置为环境变量。

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

2. 要对 API 进行身份验证，请将 `$TOKEN` 变量传递到 的 `-H` 选项curl。 例如，以下命令列出 Amazon ECR 存储库中的映像标签。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"]}
```

私有注册表设置

Amazon ECR 使用注册表设置在注册表级别配置功能。私有注册表设置是为每个区域单独配置的。目前，唯一的注册表设置是 复制设置，用于配置存储库中映像的跨区域和跨账户复制。有关更多信息，请参阅私有映像复制 (p. 39).

私有注册表权限

Amazon ECR 使用注册表策略向 AWS 委托人授予权限，从而允许将存储库从源注册表复制到您的注册表。默认情况下，您有权在自己的注册表中配置跨区域复制。只有在向另一个账户授予将内容复制到您的注册表的权限时，您才需要配置注册表策略。

注册表策略必须授予 `ecr:ReplicateImage` API 操作的权限。此 API 是一个可在区域或账户之间复制映像的内部 Amazon ECR API。您还可以授予 权限，该`ecr:CreateRepository`权限允许 在注册表中Amazon ECR创建存储库（如果它们不存在）。如果未`ecr:CreateRepository`提供权限，则必须在注册表中手动创建与源存储库同名的存储库。如果两者都未完成，则复制将失败。任何失败CreateRepository的或 ReplicateImage API 操作都会显示在 中CloudTrail。

主题

- 私有注册表策略示例 (p. 17)

设置私有注册表权限语句

您可以使用以下步骤为注册表添加或更新权限策略。您可以为每个注册表添加多个策略语句。有关示例策略，请参阅 私有注册表策略示例 (p. 17).

为私有注册表配置权限策略（AWS 管理控制台）

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/。
2. 从导航栏中，选择要在其中配置注册表策略的区域。
3. 在导航窗格中，选择注册。
4. 在 Registrys（注册表）页面上，选择您的 Private registry（私有注册表），然后选择 Permissions（权限）。
5. 在 Private registry permissions（私有注册表权限）页面上，选择 Generate statement（生成语句）。
6. 完成以下步骤以使用策略生成器定义策略声明。
   a. 对于 Policy type（策略类型），选择 Cross-account policy（跨账户策略）。
   b. 对于 Statement ID（语句 ID），输入唯一的语句 ID。此字段用作注册表策略`Sid`上的 。
   c. 对于 Accounts（账户IDs），输入要向其授予权限的每个账户的账户。指定多个账户 时IDs，请使用逗号将它们隔开。
7. 展开 Preview policy statement（预览策略语句）部分以查看注册表权限策略语句。
8. 确认策略语句后，选择 Add to policy（添加到策略）以将策略保存到您的注册表。

为私有注册表配置权限策略（AWS CLI）

1. 创建一个名为 的文件`registry_policy.json`并使用注册表策略填充该文件。

```
{
    "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. 检索注册表的策略以确认。

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

删除私有注册表权限语句

您可以使用以下步骤删除注册表的所有权限策略语句。

删除私有注册表的权限策略（AWS 管理控制台）

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/。
2. 从导航栏中，选择要在其中配置注册表权限策略的区域。
3. 在导航窗格中，选择注册。
4. 在 Registrys（注册表）页面上，选择您的 Private registry（私有注册表），然后选择 Permissions（权限）。
5. 在 Private registry permissions（私有注册表权限）页面上，选择 Delete（删除）。
6. 在 Delete registry policy 确认屏幕上，选择 Delete policy。

删除私有注册表的权限策略（AWS CLI）

1. 删除注册表策略。

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

2. 检索注册表的策略以确认。

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

私有注册表策略示例

以下示例显示了注册表权限策略语句，您可以使用这些语句来控制用户对您的Amazon ECR注册表具有的权限。

示例：允许源账户的根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：允许多个账户

```
{
    "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以 开头的所有存储库 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-*"
            ]
        }
    ]
}
```

示例：允许源账户的根用户复制以 开头的所有存储库 prod-

如果从注册表权限语句中删除了 ecr:CreateRepository 操作，则可以复制存储库。但是，要成功复制，您需要在您的账户中创建具有相同名称的存储库。

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

Amazon ECR 私有存储库

Amazon Elastic Container Registry (Amazon ECR) 提供了 API 操作来创建、监控和删除镜像存储库，并设置权限以管理谁可以访问存储库。您可以在 控制台的 Repositories（存储库）Amazon ECR 部分中执行相同的操作。Amazon ECR 还与 Docker CLI 集成，以便将镜像从开发环境推送和提取到您的存储库。

主题

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，IAM 用户需要调用 Amazon ECR API 的权限以及在存储库中推送或拉取映像的权限。Amazon ECR 提供了多个托管策略来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 (p. 72).
- 可以通过 IAM 用户访问策略和各个存储库策略来控制存储库。有关更多信息，请参阅 存储库策略 (p. 23).
- 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。通过这样做，每个团队都有自己的名为 的映像，并且每个映像均带有团队命名空间。`web-app`此配置允许每个团队中的这些图像同时使用，而不会造成干扰。团队 A 的映像为 `team-a/web-app`，团队 B 的映像为 `team-b/web-app`。
- 您的映像可以复制到您自己的注册表中和跨账户的 区域中的其他存储库。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 私有注册表设置 (p. 15).

创建存储库

在将 Docker 镜像推送到 Amazon ECR 之前，必须先创建用于存储镜像的存储库。您可以使用 Amazon ECR、AWS 管理控制台 和 AWS CLI 开发工具包来创建 AWS 存储库。

创建存储库

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/repositories。
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 Repositories.
4. 在存储库页面上，选择创建存储库.
5. 对于 Repository name（存储库名称），输入存储库的唯一名称。可以自行指定存储库名称（例如 `nginx-web-app`）。或者，也可以在它前面加上命名空间以将存储库分组到类别中（例如，`project-a/nginx-web-app`）。

Note

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（/）。

6. 对于 Tag imutability（标签不变性），选择存储库的标签可变性设置。配置了不可变标签的存储库可防止覆盖映像标签。有关更多信息，请参阅 映像标签可变性 (p. 53).
7. 对于 Scan on push（推送扫描），选择存储库的映像扫描设置。配置为在推送时扫描的存储库会在推送映像时启动映像扫描。如果要在不同的时间启动映像扫描，您需要手动启动映像扫描。有关更多信息，请参阅 映像扫描 (p. 53).
8. 对于 KMS encryption（KMS 加密），选择是否使用 AWS Key Management Service 启用对存储库中映像的加密。默认情况下，启用 KMS 加密后，Amazon ECR 使用别名为 AWS 的 `aws/ecr` 托管客户主密钥（CMK）。首次创建启用了 KMS 加密的存储库时，将在您的账户中创建此主密钥。有关更多信息，请参阅 静态加密 (p. 77).
9. 启用 KMS 加密后，选择 Customer encryption settings（advanced）（客户加密设置（高级））以选择您自己的 CMK。CMK 必须与集群位于同一区域。选择创建 AWS KMS 密钥以导航到 AWS KMS 控制台来创建您自己的密钥。
10. 选择 Create repository.
11. （可选）选择您创建的存储库，然后选择 View push commands（查看推送命令）以查看将映像推送到新存储库的步骤。

    a. 检索 docker login 命令，此命令可用于通过将控制台中的 aws ecr get-login 命令粘贴到终端窗口中来对注册表验证 Docker 客户端的身份。

       Note

       AWS CLI 从版本 1.9.15 开始提供 get-login 命令；但对于较新的 Docker 版本（17.06 或更高版本），我们建议使用 1.11.91 版或更高版本。可以使用 aws --version 命令查看 AWS CLI 版本。如果您使用的是 Docker 17.06 或更高版本，请在 `get-login` 后包含 `--no-include-email` 选项。如果收到 `Unknown options: --no-include-email` 错误，请安装最新版本的 AWS CLI。有关更多信息，请参阅 AWS Command Line Interface 用户指南 中的 安装 AWS 命令行界面。

    b. 运行上一步中返回的 docker login 命令。此命令提供一个在 12 小时内有效的授权令牌。

       Important

       在您执行此 docker login 命令时，系统上的其他用户可以在进程列表 (ps -e) 显示中看到该命令字符串。由于 docker login 命令包含身份验证凭证，因此系统上的其他用户可按此方式查看凭证会带来风险。他们可能会使用凭证获取对您的存储库的推送和拉取访问权限。如果您所在的系统不安全，则应考虑此风险，并通过省略 `-p` *`password`* 选项并在系统提示时输入密码来以交互方式登录。

    c. （可选）如果您有要让映像推送的 Dockerfile，请为新存储库构建并标记映像。将控制台中的 docker build 命令粘贴到终端窗口。确定您与您的 Dockerfile 位于同一目录中。

    d. 通过将控制台中的 docker tag 命令粘贴到终端窗口中来为 ECR 注册表和新存储库标记映像。此控制台命令假设您的映像是通过上一步中的 Dockerfile 构建得来的。如果您未通过 Dockerfile 构建映像，请将 *`repository`*`:latest` 的第一个实例更换为要推送的本地映像的映像 ID 或映像名称。

    e. 通过将 docker push 命令粘贴到终端窗口中来将新标记的映像推送到 ECR 存储库。

    f. 选择 Close。

查看存储库信息

创建存储库后，您可以在 AWS 管理控制台中查看其信息：

- 存储库中存储了哪些映像
- 映像是否有标签

- 映像的标签
- 映像的 SHA 摘要
- 映像的大小（以 MiB 为单位）
- 映像推送到存储库的时间

> **Note**
>
> 从 Docker 版本 1.9 开始，Docker 客户端将压缩映像层，然后再将其推送到 V2 Docker 注册表。命令的输出显示未压缩的映像大小。docker images因此，请记住，Docker 返回的映像可能比 AWS 管理控制台 中显示的映像大。

查看存储库信息（AWS 管理控制台）

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/repositories。
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 Repositories.
4. 在 Repositories（存储库）页面上，选择要查看的存储库。
5. 在 Repositories（存储库）上：***repository_name*** 页面上，使用导航栏查看有关映像的信息。
   - 选择 Images（映像）以查看有关存储库中的映像的信息。要查看有关映像的更多信息，请选择映像。有关更多信息，请参阅 查看映像详细信息 (p. 36).

     如果要删除未标记的映像，您可以选择要删除的存储库左侧的框，然后选择 Delete（删除）。有关更多信息，请参阅 删除映像 (p. 37).
   - 选择 Permissions（权限）以查看应用于存储库的存储库策略。有关更多信息，请参阅 存储库策略 (p. 23).
   - 选择 Lifecycle Policy（生命周期策略）以查看应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 生命周期策略 (p. 42).
   - 选择 Tags（标签）以查看应用于存储库的元数据标签。

编辑存储库

可以编辑现有存储库以更改其映像标签可变性和映像扫描设置。

编辑存储库

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/repositories。
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 Repositories.
4. 在 Repositories (存储库) 页面上，选择要编辑的存储库，然后选择 Edit (编辑).
5. 对于 Tag imutability（标签不变性），选择存储库的标签可变性设置。配置了不可变标签的存储库可防止覆盖映像标签。有关更多信息，请参阅 映像标签可变性 (p. 53).
6. 对于 Scan on push（推送扫描），选择存储库的映像扫描设置。配置为在推送时扫描的存储库会在推送时启动映像扫描。如果您希望映像扫描在不同的时间启动，则需要手动启动它们。有关更多信息，请参阅 映像扫描 (p. 53).
7. 选择 Save（保存）以更新存储库设置。

删除存储库

如果您不再使用某个存储库，可将其删除。当您在 AWS 管理控制台 中删除存储库时，该存储库中包含的所有镜像也将被删除；此操作无法撤消。

删除存储库

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/repositories。
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 Repositories.
4. 在 Repositories (存储库) 页面上，选择要删除的存储库，然后选择 Delete (删除).
5. 在 Delete （删除） 中 ***repository_name*** 窗口，确认应删除选定的存储库，然后选择 Delete （删除）。

   Important

   选定存储库中的所有映像也会被删除。

存储库策略

Amazon ECR 使用基于资源的权限控制对存储库的访问。基于资源的权限让您可以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，只有存储库所有者有权访问存储库。您可以应用策略文档来允许针对您的存储库的其他权限。

存储库策略与 IAM 政策

Amazon ECR 存储库策略是 IAM 针对个人控制访问权限的范围范围内的政策 Amazon ECR 存储库。 IAM 政策通常用于应用整个权限 Amazon ECR 还可以用于控制对特定资源的访问。

两者 Amazon ECR 存储库策略和 IAM 确定特定行动时使用政策 IAM 用户或角色可能在存储库上执行。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 策略拒绝其执行该操作（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 策略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 要求用户先通过 IAM 策略允许对 `ecr:GetAuthorizationToken` API 的权限，然后才能对镜像仓库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 策略，在不同级别控制用户访问权限。有关更多信息，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 72)。

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 Amazon ECR 存储库策略，允许特定的 IAM 用户可以描述存储库内的存储库和图像。

```
{
  "Version": "2008-10-17",
  "Statement": [{
    "Sid": "ECR Repository Policy",
    "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 策略，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 资源名称 (ARN) 的更多信息，请参阅Resources (p. 66).

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "ECR Repository Policy",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::account-id:user/username"
    },
    "Action": [
      "ecr:DescribeImages",
      "ecr:DescribeRepositories"
    ],
    "Resource": [
      "arn:aws:ecr:region:account-id:repository/repository-name"
    ]
    }]
}
```

主题

设置存储库策略声明

您可以通过以下步骤在 AWS 管理控制台中向存储库添加访问策略声明。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 存储库策略示例 (p. 25).

> **Important**
>
> Amazon ECR 要求用户先通过 IAM 策略允许对 `ecr:GetAuthorizationToken` API 的权限，然后才能对镜像仓库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 策略，在不同级别控制用户访问权限。有关更多信息，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 72)。

设置存储库策略声明

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/repositories。
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 资料库.
4. 在 资料库 页面，选择要在上设置策略声明的存储库。
5. 在导航窗格中，选择 权限， 编辑.
6. 在 编辑权限 页面，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，选择策略声明是否会导致允许或显式拒绝。
9. 对于 本金，选择要将策略声明应用到的范围。有关详细信息，请参阅 AWSJSON政策要素: 本金 在 IAM 用户指南.

   - 您可以将该语句应用到所有已验证的 AWS 用户选择 每个人(*) 复选框。
   - 对于 服务负责人，指定服务主体名称（例如， `ecs.amazonaws.com`)将声明应用到特定服务。
   - 对于 AWS帐户ID，指定 AWS 账号（例如， `111122223333`)将声明应用到特定的所有用户 AWS 账户。可以使用逗号分隔的列表指定多个账户。
   - 对于 IAM 实体，选择您的 AWS 账户将该声明应用到。

Note

对于 AWS 管理控制台 中当前不支持的较复杂的存储库策略，您可以使用 set-repository-policy AWS CLI 命令应用此策略。

10. 对于 操作，选择 Amazon ECR 策略声明应适用于单个API操作列表的API操作。
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。

删除存储库策略声明

如果不再希望将一个现有的策略声明应用至存储库，您可以删除它。

删除存储库策略声明

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/repositories。
2. 从导航栏中，选择包含要从其中删除策略声明的存储库的区域。
3. 在导航窗格中，选择 资料库.
4. 在 资料库 页面，选择要从中删除策略声明的存储库。
5. 在导航窗格中，选择 权限， 编辑.
6. 在 编辑权限 页面，选择 删除.

存储库策略示例

以下示例显示了可用于控制用户对 Amazon ECR 存储库的权限的策略声明。

Important

Amazon ECR 要求用户先通过 IAM 策略允许对 `ecr:GetAuthorizationToken` API 的权限，然后才能对镜像仓库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 策略，在不同级别控制用户访问权限。有关更多信息，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 72)。

示例：允许 IAM 您帐户内的用户

以下存储库策略允许您的账户中的 IAM 用户推送和拉取映像。

```
{
    "Version": "2008-10-17",
    "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:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability",
                "ecr:PutImage",
```

```
                "ecr:InitiateLayerUpload",
                "ecr:UploadLayerPart",
                "ecr:CompleteLayerUpload"
            ]
        }
    ]
}
```

示例：允许另一个帐户

以下存储库策略允许特定账户推送映像。

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "AllowCrossAccountPush",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::account-id:root"
            },
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchCheckLayerAvailability",
                "ecr:PutImage",
                "ecr:InitiateLayerUpload",
                "ecr:UploadLayerPart",
                "ecr:CompleteLayerUpload"
            ]
        }
    ]
}
```

以下存储库策略允许某些 IAM 用户可以拉动图像（*pull-user-1* $and *pull-user-2*），同时提供对另一个(*admin-user*).

> **Note**
>
> 对于 AWS 管理控制台 中当前不支持的较复杂的存储库策略，您可以使用 set-repository-policy AWS CLI 命令应用此策略。

```
{
    "Version": "2008-10-17",
    "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:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ]
        },
        {
            "Sid": "AllowAll",
            "Effect": "Allow",
            "Principal": {
```

```
                "AWS": "arn:aws:iam::account-id:user/admin-user"
            },
            "Action": [
                "ecr:*"
            ]
        }
    ]
}
```

示例：允许全部 AWS 要拉动图像的帐户

以下存储库策略允许所有 AWS 账户拉取映像。

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "AllowPull",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ]
        }
    ]
}
```

示例：拒绝全部

以下存储库策略拒绝所有用户拉取映像。

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "DenyPull",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ]
        }
    ]
}
```

示例：限制对特定IP地址的访问

以下示例向任何用户授予在应用于存储库时执行任何 Amazon ECR 操作的权限。但是，请求必须来自条件中指定的 IP 地址范围。

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.*`，只有一个例外：。`54.240.143.188`.

TheThethe `Condition` Block使用 `IpAddress` 和 `NotIpAddress` 条件和 `aws:SourceIp` 条件键，这是 AWS-widecondition键。有关这些条件密钥的详细信息，请参阅 AWS 全局条件上下文关键字. TheThetheaws:sourceIp Ipv4值使用标准CIDR标记。有关详细信息，请参阅 IP地址条件操作符 在 IAM 用户指南.

```
{
    "Version": "2012-10-17",
    "Id": "ECRPolicyId1",
    "Statement": [
        {
            "Sid": "IPAllow",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "ecr:*",
            "Condition": {
                "NotIpAddress": {
                    "aws:SourceIp": "54.240.143.188/32"
                },
                "IpAddress": {
                    "aws:SourceIp": "54.240.143.0/24"
                }
            }
        }
    ]
}
```

示例：服务相关角色

以下存储库策略允许 AWS CodeBuild 访问 Amazon ECR 与该服务集成所需的API操作。有关详细信息，请参阅 Amazon ECR 样本 CodeBuild 在 AWS CodeBuild 用户指南.

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

标记 Amazon ECR 存储库

为了帮助您管理您的 Amazon ECR 存储库，您可以选择通过标签的形式为每个存储库分配您自己的元数据。本主题介绍标签并说明如何创建标签。

内容

有关标签的基本知识

标签是为 AWS 资源分配的标记。每个标签都包含您定义的一个键 和一个可选值。

标签可让您按各种标准（例如用途、所有者或环境）对 AWS 资源进行分类。这在您具有相同类型的许多资源时会很有用 — 您可以根据分配给资源的标签快速识别特定资源。例如，您可以为账户的 Amazon ECR 存储库定义一组标签以帮助您跟踪每个存储库的拥有者。

我们建议您设计一组满足您的需求的标签键。使用一组连续的标签键，管理资源时会更加轻松。您可以根据添加的标签搜索和筛选资源。

标签对 Amazon ECR 没有任何语义意义，严格按字符串进行解析。同时，标签不会自动分配至您的资源。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。

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

如果您使用的是 AWS Identity and Access Management (IAM)，则可以控制 AWS 账户中的哪些用户拥有创建、编辑或删除标签的权限。

标记 资源

您可以标记新的或现有的 Amazon ECR 存储库。

如果您使用的是 Amazon ECR 控制台，则可以在创建新资源时对其应用标签，或随时在导航窗格上使用 Tags (标签) 选项对现有资源应用标签。

如果您使用的是 Amazon ECR API、AWS CLI 或 AWS 开发工具包,则可以使用 `tags` API 操作上的 CreateRepository 参数向新存储库应用标签,或使用 `TagResource` API 操作向现有资源应用标签。有关更多信息,请参阅 TagResource。

此外，如果无法在存储库创建期间应用标签，则系统将回滚存储库创建过程。这样可确保创建带有标签的存储库，或根本不创建存储库，以及确保任何时候都不创建未标记的存储库。通过在创建时标记存储库，您不需要在存储库创建后运行自定义标记脚本。

标签限制

下面是适用于标签的基本限制：

- 每个存储库的最大标签数 – 50
- 对于每个存储库，每个标签键都必须是唯一的，每个标签键只能有一个值。
- 最大键长度 – 128 个 Unicode 字符（采用 UTF-8 格式）
- 最大值长度 – 256 个 Unicode 字符（采用 UTF-8 格式）
- 如果您的标记方案针对多个服务和资源使用，请记得其他服务可能对允许使用的字符有限制。通常允许使用的字符包括：可用 UTF-8 格式表示的字母、数字和空格，以及以下字符：+ - = . _ : / @。
- 标签键和值区分大小写。
- 请不要对键或值使用 `aws:` 前缀；它保留供 AWS 使用。您无法编辑或删除带此前缀的标签键或值。具有此前缀的标签不计入每个资源的标签数限制。

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 使用率报告 (p. 90)。

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 https://docs.amazonaws.cn/awsaccountbilling/latest/aboutv2/configurecostallocreport.html 中的月度成本分配报告AWS Billing and Cost Management 用户指南。

Note

如果您已启用报告，则可以在 24 小时后查看当月的数据。

通过控制台使用标签

通过使用 Amazon ECR 控制台，您可以管理与新的或现有的存储库关联的标签。

当您在 Amazon ECR 控制台中选择特定存储库时，可通过在导航窗格中选择 Tags (标签) 来查看标签。

为存储库添加标签

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/。
2. 从导航栏中，选择要使用的区域。
3. 在导航窗格中，选择 Repositories。
4. 在 Repositories 页面上，选择要查看的存储库。
5. 在 Repositories (存储库) 上: ***repository_name*** 页面上,从导航窗格中选择 Tags (标签)。
6. 在 Tags (标签) 页面上，选择 Add tags (添加标签)、Add tag (添加标签)。
7. 在 Edit Tags (编辑标签) 页面上，为每个标签指定键和值，然后选择 Save (保存)。

删除单个资源的标签

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/。
2. 从导航栏中，选择要使用的区域。
3. 在 Repositories 页面上，选择要查看的存储库。
4. 在 Repositories (存储库) 上: ***repository_name*** 页面上,从导航窗格中选择 Tags (标签)。
5. 在 Tags (标签) 页面上，选择 Edit (编辑)。
6. 在 Edit Tags (编辑标签) 页面上，选择要删除的每个标签对应的 Remove (删除)，然后选择 Save (保存)。

通过 AWS CLI 或 API 使用标签

使用以下命令添加、更新、列出和删除资源标签。相应文档提供了示例。

Amazon ECR 资源标记支持

任务	AWS CLI	API 操作
添加或覆盖一个或多个标签。	tag-resource	TagResource
删除一个或多个标签。	untag-resource	UntagResource

以下示例演示如何使用 AWS CLI 管理标签。

示例 1：标记现有存储库

以下命令标记现有存储库。

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

示例 2：使用多个标签标记现有存储库

以下命令标记现有存储库。

```
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: 取消标记现有存储库

以下命令删除现有存储库的标签。

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

示例 4: 列出存储库的标签

以下命令列出与现有存储库关联的标签。

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

示例 5: 创建存储库并应用标签

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
```

私有映像

Amazon Elastic Container Registry (Amazon ECR) 将 Docker 映像、开放容器计划（OCI）映像和 OCI 兼容构件存储在存储库中。您可以使用 Docker CLI 或首选客户端在存储库中推送和拉取映像。

> **Important**
>
> Amazon ECR 要求用户先通过 IAM 策略允许对 `ecr:GetAuthorizationToken` API 的权限，然后才能对镜像仓库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 策略，在不同级别控制用户访问权限。有关更多信息，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 72)。

主题

推送映像

您可以将 Docker 映像、清单列表和 Open Container Initiative（OCI）映像和兼容的构件推送到您的存储库。以下页面将更详细地介绍这些内容。

> **Note**
>
> 通过在注册表设置中指定复制配置，您的映像可以跨自己的注册表中的区域和跨账户复制到其他存储库。有关更多信息，请参阅 私有注册表设置 (p. 15).

主题

推送 Docker 映像

您可以使用 Amazon ECR 命令将 Docker 映像推送到 docker push 存储库。

Important

Amazon ECR 要求用户先通过 IAM 策略允许对 `ecr:GetAuthorizationToken` API 的权限，然后才能对镜像仓库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 策略，在不同级别控制用户访问权限。有关更多信息，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 72)。

Amazon ECR 还支持创建和推送用于多架构映像的 Docker 清单列表。清单列表中引用的每个映像都必须已经被推送到您的存储库。有关更多信息，请参阅 推送多架构映像 (p. 33).

推送 Docker 镜像到 Amazon ECR 存储库

1. 向要向其推送镜像的 Amazon ECR 镜像仓库验证 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 私有注册表身份验证 (p. 13).
2. 如果您的映像存储库在要推送到的注册表中尚不存在，请创建它。有关更多信息，请参阅 创建存储库 (p. 20).
3. 识别要推送的映像。运行 docker images 命令列出系统中的映像。

```
docker images
```

   您可以使用 *repository:tag* 生成的命令输出中的 值或映像 ID。
4. 通过要使用的 Amazon ECR 镜像仓库、存储库和可选镜像标签名称组合标记您的镜像。镜像仓库格式为 *aws_account_id*.dkr.ecr.*region*.amazonaws.com. 存储库名称应与您为映像创建的存储库一致。如果省略映像标签，我们将假定标签为 `latest`.

   以下示例使用 ID 标记映像 *e9ae3c220b23* 作为 *aws_account_id*.dkr.ecr.*region*.amazonaws.com/my-web-app。

```
docker tag e9ae3c220b23 aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app
```

5. 使用 docker push 命令推送映像：

```
docker push aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app
```

6. （可选）通过重复Amazon ECR和Step 4 (p. 33)，向镜像应用任何其他标签并将这些标签推送到 Step 5 (p. 33). 您最多可以为 Amazon ECR 中的每个映像应用 100 个标签。

推送多架构映像

Amazon ECR 支持创建和推送用于多架构映像的 Docker 清单列表。清单列表 是通过指定一个或多个映像名称创建的映像列表。在大多数情况下，清单列表是从提供相同功能但适用于不同操作系统或架构的映像创建的。清单列表不是必需的。有关更多信息，请参阅 Docker 清单.

Important

您的 Docker CLI 必须启用实验功能才能使用此功能。有关详细信息，请参阅实验功能.

清单列表可以像其他 Amazon ECS 映像一样在 Amazon EKS 任务定义或 Amazon ECR Pod 规范中被拉取或引用。

可以使用以下步骤创建 Docker 清单列表并将其推送到 Amazon ECR 存储库。您必须已将映像推送到您的存储库，才能在 Docker 清单中引用。有关如何推送映像的信息，请参阅推送 Docker 映像 (p. 32)。

将多架构 Docker 映像推送到 Amazon ECR 存储库

1. 向要将映像推送到的 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 私有注册表身份验证 (p. 13).

2. 列出存储库中的映像，确认映像标签。

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

3. 创建 Docker 清单列表。`manifest create` 命令验证引用的映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.region.amazonaws.com/my-
web-app aws_account_id.dkr.ecr.region.amazonaws.com/my-web-
app:image_one_tag aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app: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映像清单的大小和摘要。

```
docker manifest inspect aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app
```

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.region.amazonaws.com/my-web-app
```

推送 Helm 图表

Amazon ECR 支持将 Open Container Initiative（OCI）构件推送到您的存储库。要显示此功能，请使用以下步骤将 Helm 图表推送到 Amazon ECR。

有关将 Amazon ECR 托管的 Helm 图表与 Amazon EKS 一起使用的更多信息，请参阅安装托管在上的Helm 图表 Amazon ECR 配 Amazon EKS (p. 59)。

将 Helm 图表推送到 Amazon ECR 存储库

1. 安装 Helm 客户端版本 3。有关更多信息，请参阅安装 Helm。
2. 在 Helm 3 客户端中启用 OCI 支持。

```
export HELM_EXPERIMENTAL_OCI=1
```

3. 创建用于存储 Helm 图表的存储库。有关更多信息，请参阅 创建存储库 (p. 20).

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

4. 向要将 Helm 图表推送到的 Amazon ECR 注册表验证 Helm 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 私有注册表身份验证 (p. 13).

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

5. 使用以下步骤创建测试 Helm 图表。有关更多信息，请参阅 Helm 文档 - 入门。

   a. 创建一个名为 `helm-tutorial` 的目录以在其中工作。

```
mkdir helm-tutorial
cd helm-tutorial
```

b. 创建名为 mychart 的 Helm 图表并清除 templates 目录的内容。

```
helm create mychart
rm -rf ./mychart/templates/*
```

c. 在 templates 文件夹中创建 ConfigMap。

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

6. 在本地保存图表并使用注册表 URI 为图表创建别名。

```
cd ..
helm chart save . mychart
helm chart save . aws_account_id.dkr.ecr.us-west-2.amazonaws.com/artifact-test:mychart
```

7. 确定要推送的 Helm 图表。运行 helm chart list 命令以列出系统上的 Helm 图表。

```
helm chart list
```

输出应类似于以下内容：

```
REF                                                             NAME    VERSION DIGEST
 SIZE    CREATED
aws_account_id.dkr.ecr.us-west-2.amazonaws.com/artifact-tes..   mychart 0.1.0   30e0a03
 3.6 KiB 14 seconds
mychart                                                         mychart 0.1.0   ba3e62a 3.6
 KiB About a minute
```

8. 使用 helm chart push 命令推送 Helm 图表：

```
helm chart push aws_account_id.dkr.ecr.region.amazonaws.com/artifact-test:mychart
```

9. 描述您的 Helm 图表。

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

在输出中，验证 artifactMediaType 参数是否指示正确的构件类型。

```
{
    "imageDetails": [
        {
            "registryId": "aws_account_id",
            "repositoryName": "artifact-test",
            "imageDigest":
 "sha256:f23ab9dc0fda33175e465bd694a5f4cade93eaf62715fa9390d9fEXAMPLE",
            "imageTags": [
                "mychart"
            ],
            "imageSizeInBytes": 3714,
```

```
            "imagePushedAt": 1597433021.0,
            "imageManifestMediaType": "application/vnd.oci.image.manifest.v1+json",
            "artifactMediaType": "application/vnd.cncf.helm.config.v1+json"
        }
    ]
}
```

查看映像详细信息

将映像推送到存储库后，您可以在 AWS 管理控制台中查看其信息。包含的详细信息如下所示：

- 映像 URI
- 映像标签
- Artifact 媒体类型
- 映像清单类型
- 正在扫描状态
- 映像的大小（以 MB 为单位）
- 映像推送到存储库的时间
- 复制状态

查看映像详细信息（AWS 管理控制台）

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/repositories。
2. 从导航栏中，选择包含您的映像的存储库的区域。
3. 在导航窗格中，选择 Repositories.
4. 在 Repositories（存储库）页面上，选择要查看的存储库。
5. 在 Repositories（存储库）上：***repository_name*** 页面，选择要查看其详细信息的映像。

拉取映像

如果要运行 Amazon ECR 中可用的 Docker 映像，可以使用 docker pull 命令将其拉取到本地环境中。可以从默认镜像仓库或与其他 AWS 账户关联的镜像仓库执行此操作。要在 Amazon ECR 任务定义中使用 Amazon ECS 映像，请参阅 在 Amazon ECS 中使用 Amazon ECR 映像 (p. 58).

> **Important**
>
> Amazon ECR 要求用户先通过 IAM 策略允许对 `ecr:GetAuthorizationToken` API 的权限，然后才能对镜像仓库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 策略，在不同级别控制用户访问权限。有关更多信息，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 72)。

从 Amazon ECR 存储库拉取 Docker 镜像

1. 将您的 Docker 客户端验证到要从中拉取镜像的 Amazon ECR 镜像仓库。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 私有注册表身份验证 (p. 13).
2. (可选)识别要拉取的映像。
   - 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 的存储库。amazonlinux.

- 可以使用 aws ecr describe-images 命令描述存储库中的映像：

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

上述示例存储库具有带标签 latest 和 2016.09 的映像，并且映像摘要为 sha256:f1d4ae3f7261a72e98c6ebefe9985cf10a0ea5bd762585a43e0700ed99863807.

3. 使用 docker pull 命令拉取映像。映像名称格式应为 *registry*/*repository*[:*tag*] 以便按标签拉取，或为 *registry*/*repository*[@*digest*] 以便按摘要拉取。

```
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，则可能需要向 Amazon ECR 验证 Docker 客户端。有关更多信息，请参阅 私有注册表身份验证 (p. 13).

删除映像

如果您不再使用某个映像，可以从存储库中将其删除。您可以使用 AWS 管理控制台或 AWS CLI 删除映像。

Note

如果您不再使用存储库，可以删除整个存储库及其中的所有映像。有关更多信息，请参阅 删除存储库 (p. 22).

使用 删除镜像AWS 管理控制台

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/repositories。
2. 从导航栏中，选择包含要删除的映像的区域。
3. 在导航窗格中，选择 Repositories.
4. 在 Repositories （存储库）页面上，选择包含要删除的映像的存储库。
5. 在 Repositories （存储库）上：***repository_name*** 页面，选择要删除的映像左侧的框并选择 Delete（删除）。
6. 在 Delete image(s) 对话框中，验证选定的映像是否应被删除，然后选择 Delete.

使用 删除镜像AWS CLI

1. 列出存储库中的映像，以便按映像标签或摘要标识映像。

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

2. (可选) 通过指定要删除的映像标签来删除映像的任何不需要的标签。

Note

删除映像的最后一个标签后，将删除映像。

```
aws ecr batch-delete-image --repository-name my-repo --image-ids imageTag=latest
```

3. 通过指定要删除的映像的摘要来删除映像。

Note

在通过引用映像摘要来删除映像时，映像及其所有标签都会被删除。

```
aws ecr batch-delete-image --repository-name my-repo --image-ids
 imageDigest=sha256:4f70ef7a4d29e8c0c302b13e25962d8f7a0bd304c7c2c1a9d6fa3e9de6bf552d
```

重新为映像添加标签

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

重新为映像添加标签 (AWS CLI)

使用 重新为映像添加标签AWS CLI

1. 使用 batch-get-image 命令可获取要重新添加标签的映像的映像清单并将其写入环境变量。在此示例中，标签为 的映像的清单。*latest*，在存储库中，*amazonlinux*，写入环境变量，*MANIFEST*.

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

2. 使用 `--image-tag` 命令的 put-image 选项将镜像清单与新标签一起放置到 Amazon ECR 中。在本示例中，映像标记为 *2017.03*.

   Note

   如果 `--image-tag` 选项在您的 AWS CLI 版本中不可用，请升级到最新版本。有关更多信息，请参阅 https://docs.amazonaws.cn/cli/latest/userguide/ 中的AWS Command Line Interface 用户指南安装 AWS 命令行界面.

```
aws ecr put-image --repository-name amazonlinux --image-tag 2017.03 --image-manifest
 "$MANIFEST"
```

3. 验证您的新映像标签是否已附加到您的映像。在以下输出中，映像具有标签 `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":
 "sha256:8d00af8f076eb15a33019c2a3e7f1f655375681c4e5be157a2685dfe6f247227",
            "imageTags": [
                "latest",
                "2017.03"
            ],
            "registryId": "aws_account_id",
            "repositoryName": "amazonlinux",
            "imagePushedAt": 1499287667.0
        }
    ]
```

```
}
```

重新为映像添加标签 (适用于 Windows PowerShell 的 AWS 工具)

使用 重新为映像添加标签适用于 Windows PowerShell 的 AWS 工具

1. 使用 Get-ECRImageBatch cmdlet 获取要重新添加标签的映像的描述并将其写入环境变量。在此示例中，具有标签的图像。*latest*，在存储库中，*amazonlinux*，写入环境变量，*$Image*.

   Note

   如果您的系统上没有可用的 Get-ECRImageBatch cmdlet，请参阅 中的适用于 Windows PowerShell 的 AWS 工具设置 。适用于 Windows PowerShell 的 AWS 工具 用户指南

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

2. 将映像的清单写入 *$Manifest* 环境变量。

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

3. 使用 `-ImageTag` cmdlet 的 Write-ECRImage 选项将镜像清单与新标签一起放置到 Amazon ECR 中。在本示例中，映像标记为 *2017.09*.

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

4. 验证您的新映像标签是否已附加到您的映像。在以下输出中，映像具有标签 `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
```

私有映像复制

Amazon ECR 使用注册表设置在注册表级别配置私有映像复制。可以为跨区域或跨账户复制配置 Amazon ECR 私有注册表。为每个区域的私有注册表单独配置复制。下面更详细地介绍了支持的复制方法。

跨区域复制

为注册表启用跨区域复制会在一个或多个目标区域中复制存储库。仅复制在配置跨区域复制后推送到存储库的映像。

跨账户复制

为注册表启用跨账户复制会在您指定的目标账户和区域中复制存储库。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注册表进行复制。有关更多信息，请参阅 私有注册表权限 (p. 15).

主题

- 私有映像复制的注意事项 (p. 40)

私有映像复制的注意事项

使用私有映像复制时应考虑以下因素。

- 首次配置私有注册表进行复制时，Amazon ECR 会代表您创建一个服务相关角色。该服务相关角色向 Amazon ECR 复制服务授予在注册表中创建存储库和复制映像所需的权限。有关更多信息，请参阅 对 使用服务相关角色Amazon ECR (p. 70).
- 要进行跨账户复制，目标私有注册表必须授予权限以允许源注册表复制其图像。有关更多信息，请参阅 私有注册表权限 (p. 15).
- 如果更改注册表的权限以删除权限，则以前授予的任何正在进行的复制都将完成。
- 每次推送映像时，复制操作仅发生一次。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，推送到 `us-west-2` 的映像将仅复制到 `us-east-1`，则不会再次复制到 `us-east-2`。此行为适用于跨区域和跨账户复制。
- 必须在账户执行任何复制操作之前为该账户启用区域，然后才会在该区域中执行任何复制操作。有关更多信息，请参阅 https://docs.amazonaws.cn/general/latest/gr/rande-manage.html 中的管理 AWS 区域 Amazon Web Services 一般参考。
- 注册表复制不执行任何删除操作。在不再使用复制的映像和存储库时，可以手动删除它们。
- 生命周期策略不会复制，并且除了定义这些策略的存储库之外没有任何影响。
- 不复制存储库设置。默认情况下，在由于复制操作而创建的所有存储库上禁用标签不可变性、映像扫描和 KMS 加密设置。创建存储库后，可以更改标签不可变性和映像扫描设置。但是，该设置仅适用于该设置更改后推送的图像。
- 如果在存储库上启用了标签不可变性，并且复制了使用与现有映像相同的标签的映像，则将复制映像，但不会包含重复的标签。这可能会导致未标记映像。

配置私有映像复制

将为每个区域单独配置复制设置。使用以下步骤为您的私有注册表配置复制。

配置注册表复制设置（AWS 管理控制台）

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/repositories。
2. 从导航栏中，选择要为其配置注册表复制设置的区域。
3. 在导航窗格中，选择注册表。
4. 在 Registries（注册表）页面上，选择您的 Private（私有）注册表，然后选择 Edit（编辑）。
5. 在 Edit registry（编辑注册表）页面上，执行以下操作。
   a. 对于 Cross-Region replication（跨区域复制），选择注册表的跨区域复制设置。如果设置为 Enabled（已启用），请选择一个或多个 Destination regions（目标区域）。
   b. 对于 Cross-account replication（跨账户复制），选择注册表的跨账户复制设置。如果设置为 Enabled（已启用），请输入目标账户的账户 ID 以及要复制到的一个或多个 Destination regions（目标区域）。

      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 私有注册表权限 (p. 15).
6. 选择 Save.

配置注册表复制设置（AWS CLI）

1. 创建一个 JSON 文件，其中包含要为您的注册表定义的复制配置设置。其中可能包含一个或多个规则，每个规则包含一个目标区域和账户。如果要在区域之间复制自己的注册表中的映像，请指定您自己的账户 ID。有关更多示例，请参阅 私有映像复制示例 (p. 41).

```
{
    "rules": [
        {
            "destinations": [
                {
                    "region": "destination_region",
                    "registryId": "destination_accountId"
                }
            ]
        }
    ]
}
```

2. 为注册表创建复制配置。

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

3. 确认注册表设置。

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

私有映像复制示例

以下示例说明如何使用私有映像复制。

示例：配置到单个目标区域的跨区域复制

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`，并且您在 `us-west-2` 之外的区域中指定此复制配置。

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

示例：配置到多个目标区域的跨区域复制

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`，并且您将在 `us-west-1` 或 `us-west-2` 以外的区域中指定此复制配置。

```
{
    "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。此示例配置到 `444455556666` 账户和 `us-west-2` 区域的复制。

> **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 私有注册表权限 (p. 15).

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

生命周期策略

Amazon ECR 生命周期策略使您能够指定存储库中映像的生命周期管理。生命周期策略是一组规则，其中的每个规则为 定义一个操作。Amazon ECR. 这些操作适用于包含用给定字符串预固定标签的图像。这允许自动清理未使用的图像，例如基于年龄或计数的到期图像。您应该预计在创建生命周期策略后，受影响的影像在24小时内到期。

主题

生命周期策略模板

在与存储库关联之前，评估生命周期策略的内容。以下是生命周期策略的JSON语法模板。有关生命周期策略示例，请参阅 生命周期策略示例 (p. 47).

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

Note

TheThethe `tagPrefixList` 参数只能使用 `tagStatus` 是 `tagged`...TheThethe `countUnit` 参数只能使用 `countType` 是 `sinceImagePushed`...TheThethe `countNumber` 参数只能使用 `countType` 设置为 `imageCountMoreThan`.

生命周期策略参数

生命周期策略分为以下部分:

主题

规则优先级

`rulePriority`

类型：整数

必需：是

设置评估规则的顺序，从低到高。生命周期策略规则的优先级 `1` 将首先采取行动，一个规则优先于 `2` 将会接下来的，所以。在将规则添加到生命周期策略时，您必须为其提供每个 `rulePriority`...策略中的规则不需要连续的值。一个规则 `tagStatus` 价值 `any` 必须具有最高值 `rulePriority` 并按照最后一次评估。

Description

`description`

类型：字符串

必需：否

（可选）描述生命周期策略中规则的目的。

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映像指定标记。可接受的选项是 `tagged`，`untagged`，或 `any`...如果您指定 `any`，然后所有图像都应用于它们。如果您指定 `tagged`，然后您还必须指定 `tagPrefixList` 值。如果您指定 `untagged`然后你必须忽略 `tagPrefixList`.

标签前缀列表

`tagPrefixList`

类型:列表[字符串]

要求:是，只有 `tagStatus` 设置为标记

仅在您指定时使用 `"tagStatus": "tagged"`...您必须指定一个以逗号分隔的图像标签前缀列表，该列表将采取与生命周期策略的操作。例如，如果您的图像被标记为 `prod`，`prod1`，`prod2`，这样就会使用标签前缀 `prod` 要指定所有。如果指定多个标记，则只选择带有所有指定标记的图像。

计数类型

`countType`

类型：字符串

必需：是

指定应用于图像的计数类型。

IFIFIF `countType` 设置为 `imageCountMoreThan`，您还指定 `countNumber` 创建一个规则，对存储库中存在的图像数量设置限制。IFIFIF `countType` 设置为 `sinceImagePushed`，您还指定 `countUnit` 和 `countNumber` 要指定存储库中存在的图像的时间限制。

计数单位

`countUnit`

类型：字符串

要求:是，只有 `countType` 设置为 `sinceImagePushed`

指定计数单位 `days` 以此表示作为时间单位，除此之外 `countNumber`，这是天数。

只有在这种情况下 countType 是 sinceImagePushed；如果您指定计数单位，将发生错误 countType 是任何其他值。

计数

countNumber

类型：整数

必需：是

指定计数数量。可接受值为正整数（0 不是已接受值）。

如果 countType 已使用 imageCountMoreThan，然后值是您要保留在存储库中的最大图像数。如果 countType 已使用 sinceImagePushed，然后值为图像的最大年龄限制。

Action

type

类型：字符串

必需：是

指定操作类型。支持的值为 expire.

生命周期策略评估规则

生命周期策略评估员负责解析原文JSON并将其应用到指定存储库中的图像。创建生命周期策略时应注意以下规则:

- 图像已按一个或零个规则过期。
- 符合规则标签要求的图像不能按优先级较低的规则过期。
- 规则永远不会标记由更高优先级规则标记的图像，但是仍然可以将其标识为如同尚未过期的那样。
- 一组规则必须包含一组唯一的标签前缀。
- 只允许一个规则选择未标签的图像。
- 有效期始终为 pushed_at_time，并且在更新的图像之前始终使旧图像过期。
- 使用 tagPrefixList，图像已成功匹配IF 全部 标记 tagPrefixList 值与任何图像标签匹配。
- 带有 countType = imageCountMoreThan，图像的排序范围为基于 pushed_at_time 然后，所有大于指定计数的图像都已过期。
- 带有 countType = sinceImagePushed，所有图像 pushed_at_time 早于指定天数，基于 countNumber 已过期。

创建生命周期策略预览

生命周期策略预览允许您在执行图像存储库之前查看生命周期策略的影响。以下步骤向您展示如何创建生命周期策略预览。

如何使用控制台创建生命周期策略预览

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/repositories。
2. 从导航栏中选择包含要执行生命周期策略预览的存储库的区域。

3. 在导航窗格中，选择 资料库 并选择一个库。
4. 在 资料库: ***repository_name*** 页面，在导航窗格中选择 生命周期政策.
5. 在 资料库: ***repository_name***:生命周期策略 页面，选择 编辑测试规则， 创建规则.
6. 输入您的生命周期策略规则的以下详细信息:

   a. 对于 规则优先级，键入规则优先级的数字。
   b. 对于 规则描述，键入LifecyclePolicy规则的说明。
   c. 对于 图像状态，选择 标记， 未标记，或 任何.
   d. 如果您指定 `Tagged` 对于 图像状态，然后 标签前缀，您可以选择性地指定要与生命周期策略采取操作的图像标签列表。如果您指定 `Untagged`，此字段必须为空。
   e. 对于 匹配标准，选择值 由于图像推送 或 图像计数超过 （ 如适用 ）。
7. 选择 保存.
8. 通过重复步骤5–7创建其他生命周期策略规则。
9. 要运行生命周期策略预览，请选择 保存并运行测试.
10. 下方 测试生命周期规则的图像匹配项，回顾生命周期策略预览的影响。
11. 如果您对预览结果感到满意，请选择 应用作生命周期策略 创建具有指定规则的生命周期策略。

    Note

    您应该预计在创建生命周期策略后，受影响的影像在24小时内过期。

创建生命周期策略

生命周期策略允许您创建一组使未使用的存储库映像过期的规则。以下步骤向您展示如何创建生命周期策略。您应该预计在创建生命周期策略后，受影响的影像在24小时内过期。

创建生命周期策略AWS CLI)

使用 AWS CLI

1. 获取要创建生命周期策略的存储库的ID:

```
aws ecr describe-repositories
```

2. 创建生命周期策略

```
aws ecr put-lifecycle-policy [--registry-id <string>] --repository-name <string> --
lifecycle-policy-text <string>
```

创建生命周期策略AWS 管理控制台)

如何使用控制台创建生命周期策略

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/repositories。
2. 从导航栏中选择包含要创建生命周期策略的存储库的区域。
3. 在导航窗格中，选择 资料库 并选择一个库。
4. 在 资料库: ***repository_name*** 页面，在导航窗格中选择 生命周期政策.
5. 在 资料库: ***repository_name***:生命周期策略 页面，选择 创建规则.
6. 输入您的生命周期策略规则的以下详细信息:

   a. 对于 规则优先级，键入规则优先级的数字。

b. 对于 规则描述，键入LifecyclePolicy规则的说明。
c. 对于 图像状态，选择 标记， 未标记，或 任何.
d. 如果您指定 `Tagged` 对于 图像状态，然后 标签前缀，您可以选择性地指定要与生命周期策略采取操作的图像标签列表。如果您指定 `Untagged`，此字段必须为空。
e. 对于 匹配标准，选择值 由于图像推送 或 图像计数超过 （ 如适用 ）。

7. 选择 保存.

生命周期策略示例

以下是生命周期策略示例，显示语法。

主题

图像年龄过滤

以下示例显示了一个策略的生命周期策略语法，该策略将过期未标签的图像。 `14` 天数:

```
{
    "rules": [
        {
            "rulePriority": 1,
            "description": "Expire images older than 14 days",
            "selection": {
                "tagStatus": "untagged",
                "countType": "sinceImagePushed",
                "countUnit": "days",
                "countNumber": 14
            },
            "action": {
                "type": "expire"
            }
        }
    ]
}
```

图像计数过滤

以下示例显示了一个策略的生命周期策略语法，该策略仅保留一个未标签的图像，并将其过期到所有其他:

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

```
                "type": "expire"
            }
        }
    ]
}
```

筛选多个规则

以下示例使用生命周期策略中的多个规则。示例资料库和生命周期政策与结果的解释一同提供。

示例A

存储库内容:

- 图片A，标语列表:["beta-1","prod-1"]，推送:10天前
- 图片B，标语列表:["beta-2","prod-2"]，推送:9天前
- 图片C，标语列表:["beta-3"]，按下:8天前

生命周期策略文本:

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

此生命周期策略的逻辑将是:

- 规则1标识标有前缀的图像 `prod`...它应该标记图像以最旧的方式开始，直到剩余的一个或多个图像匹配。它标记到期的图像A。
- 规则2标识标有前缀的图像 `beta`...它应该标记图像以最旧的方式开始，直到剩余的一个或多个图像匹配。它将图像A标记为有效期的图像B。但是，规则1已经看到图像A，如果图像B已过期，则会违反规则1，因此跳过。
- 结果：图像A已过期。

示例B

这与上一个示例的存储库相同，但规则优先顺序更改为说明结果。

存储库内容:

- 图片A，标语列表:["beta-1","prod-1"]，推送:10天前
- 图片B，标语列表:["beta-2","prod-2"]，推送:9天前
- 图片C，标语列表:["beta-3"]，按下:8天前

生命周期策略文本:

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

此生命周期策略的逻辑将是:

- 规则1标识标有 `beta`...它应该标记图像以最旧的方式开始，直到剩余的一个或多个图像匹配。它会看到所有三个图像，并将图像A标记为有效期。
- 规则2标识标有 `prod`...它应该标记图像以最旧的方式开始，直到剩余的一个或多个图像匹配。它将看不到图像，因为所有可用图像都已经按规则1看到，因此标记没有其他图像。
- 结果：图像A和B已过期。

在单个规则中筛选多个标记

以下示例指定了单个规则中多个标记前缀的生命周期策略语法。示例资料库和生命周期政策与结果的解释一同提供。

示例A

在单个规则中指定多个标签前缀时，图像必须匹配所有列出的标签前缀。

存储库内容:

- 图片A，标语列表:["alpha-1"]，按下:12天前
- 图片B，标语列表:["beta-1"]，按下:11天前
- 图片C，标语列表:["alpha-2","beta-2"]，按下:10天前
- 图像D，标语列表:["alpha-3"]，推送:4天前
- 图像E，标语列表:["beta-3"]，推送:3天前
- 图片F，标语列表:["alpha-4","beta-4"]，按下:2天前

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

此生命周期策略的逻辑将是:

- 规则1标识标有 `alpha` 和 `beta`...它看到图像C和F。它应该标记超过5天的图像，这将是图像C。
- 结果：图像C已过期。

示例B

以下示例说明标签不属于唯一标签。

存储库内容:

- 图片A，标语列表:["alpha-1","beta-1","gamma-1"]，推送:10天前
- 图片B，标语列表:["alpha-2","beta-2"]，按下:9天前
- 图片C，标语列表:["alpha-3","beta-3","gamma-2"]，推送:8天前

```
{
    "rules": [
        {
            "rulePriority": 1,
            "description": "Rule 1",
            "selection": {
                "tagStatus": "tagged",
                "tagPrefixList": ["alpha", "beta"],
                "countType": "imageCountMoreThan",
                "countNumber": 1
            },
            "action": {
```

```
                "type": "expire"
            }
        }
    ]
}
```

此生命周期策略的逻辑将是:

- 规则1标识标有 `alpha` 和 `beta`...它会看到所有图像。它应该标记图像以最旧的方式开始，直到剩余的一个或多个图像匹配。它标记了到期的图像A和B。
- 结果：图像A和B已过期。

筛选所有图像

以下生命周期策略示例指定带有不同筛选器的所有图像。示例资料库和生命周期政策与结果的解释一同提供。

示例A

下面显示了适用于所有规则但仅保留一个映像并使所有其他规则失效的策略的生命周期策略语法。

存储库内容:

- 图片A，标语列表:["alpha-1"]，推送:4天前
- 图片B，标语列表:["beta-1"]，推送:3天前
- 图片C，标签列表: []，按下:2天前
- 图像D，标语列表:["alpha-2"]，按下: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将是:

- 规则1标识所有图像。它会看到图像A、B、C和D。它应该会过期除最新一个图像之外的所有图像。它标记到期的图像A、B和C。
- 结果：图像A、B和C已过期。

示例B

以下示例说明了一个在单个策略中结合所有规则类型的生命周期策略。

存储库内容:

- 图片A，标语列表:["alpha-1","beta-1"]，推送:4天前
- 图片B，标签列表: []，按下:3天前
- 图片C，标语列表:["alpha-2"]，按下:2天前
- 图像D，标语列表:["GITHASH"]，按下:1天前
- 图片E，标语列表: []，按下:1天前

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

- 规则1标识标有 `alpha`...它标识图像A和C。应保留最新图像并标记到期的剩余时间。它标记到期的图像A。
- 规则2标识未标签的图像。它标识图像B和E。应该标记到期时间超过1天的所有图像。它标记到期的图像B。
- 规则3标识所有图像。它识别映像 A、B、C、D 和 E。它应该保留最新映像并将其余映像标记为过期。但是，它不能标记图像A、B、C或E，因为它们被更高优先级规则识别。它将映像 D 标记为过期。

- 结果：图像A、B和D已过期。

映像标签可变性

您可以将存储库配置为不可变，以防止覆盖映像标签。在为存储库配置了不可变标签后，如果您尝试推送具有存储库中已存在的标签的映像，则会返回 `ImageTagAlreadyExistsException` 错误。

使用 AWS 管理控制台和 AWS CLI 工具，您可以在新存储库的创建期间或者随时为现有存储库设置映像标签的可变性。对于控制台步骤，请参阅创建存储库 (p. 20)和编辑存储库 (p. 22).

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映像存储库。

- create-repository (AWS CLI)

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

- New-ECRRepository (适用于 Windows PowerShell 的 AWS 工具)

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

更新现有存储库的映像标签可变性设置

使用以下命令之一更新现有存储库的映像标签可变性设置。

- put-image-tag-mutability (AWS CLI)

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

- Write-ECRImageTagMutability (适用于 Windows PowerShell 的 AWS 工具)

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

映像扫描

Amazon ECR 映像扫描有助于识别容器映像中的软件漏洞。Amazon ECR 使用开源 Clair 项目中的常见漏洞和风险（CVE）数据库，并提供扫描发现结果的列表。您可以查看扫描结果以获取有关正在部署的容器映像的安全性的信息。有关 Clair 的更多信息，请参阅 GitHub 上的 Clair。

Amazon ECR 使用上游分配源中的 CVE 的严重性（如果可用），否则我们使用通用漏洞评分系统 (CVSS) 评分。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.

您可以手动扫描存储在 Amazon ECR 中的容器映像。或者，您也可以将存储库配置为在将映像推送到存储库时扫描映像。可以为每个映像检索上次完成的映像扫描结果。映像扫描完成后，Amazon ECR 向 Amazon EventBridge（以前称为 CloudWatch Events）发送事件。有关更多信息，请参阅 Amazon ECR 事件和 EventBridge (p. 90).

有关扫描映像时的常见问题的排查详细信息，请参阅 排查映像扫描问题 (p. 108).

主题

配置存储库以在推送时扫描

您可以在创建过程中为新存储库或为现有存储库配置映像扫描设置。在启用 scan on push （推送时扫描）时，将在推送到存储库后扫描映像。如果在存储库上禁用了 scan on push （推送时扫描），则必须手动启动每个映像扫描以获取扫描结果。

主题

创建一个新的存储库以在推送时进行扫描

当新存储库配置为 scan on push （推送时扫描）时，将扫描推送到该存储库的所有新映像。然后，可以检索上次完成的映像扫描的结果。有关更多信息，请参阅 检索映像扫描查找结果 (p. 56).

要查看 AWS 管理控制台 步骤，请参阅创建存储库 (p. 20).

创建配置用于推送时扫描的存储库 (AWS CLI)

使用以下命令创建一个新存储库并配置了映像 scan on push （推送时扫描）。

- create-repository (AWS CLI)

```
aws ecr create-repository --repository-name name --image-scanning-configuration
 scanOnPush=true --region us-east-2
```

创建配置用于推送时扫描的存储库 (适用于 Windows PowerShell 的 AWS 工具)

使用以下命令创建一个新存储库并配置了映像 scan on push （推送时扫描）。

- New-ECRRepository (适用于 Windows PowerShell 的 AWS 工具)

```
New-ECRRepository -RepositoryName name -ImageScanningConfiguration_ScanOnPush true -
Region us-east-2 -Force
```

配置现有存储库以在推送时扫描

您可以将现有存储库配置为在将映像推送到存储库时对其进行扫描。此设置将应用于未来的映像推送。然后，可以检索上次完成的映像扫描的结果。有关更多信息，请参阅 检索映像扫描查找结果 (p. 56).

要查看 AWS 管理控制台 步骤，请参阅编辑存储库 (p. 22).

编辑现有存储库的设置 (AWS CLI)

使用以下命令编辑现有存储库的映像扫描设置。

- put-image-scanning-configuration (AWS CLI)

```
aws ecr put-image-scanning-configuration --repository-name name --image-scanning-
configuration scanOnPush=true --region us-east-2
```

Note

要为存储库禁用映像 scan on push (推送时扫描)，请指定 `scanOnPush=false`.

编辑现有存储库的设置 (适用于 Windows PowerShell 的 AWS 工具)

使用以下命令编辑现有存储库的映像扫描设置。

- New-ECRRepository (适用于 Windows PowerShell 的 AWS 工具)

```
Write-ECRImageScanningConfiguration -RepositoryName name -
ImageScanningConfiguration_ScanOnPush true -Region us-east-2 -Force
```

手动扫描映像

当您要扫描存储库中未配置为 scan on push （推送时扫描）的映像时，可以手动启动映像扫描。每天只能扫描一次映像。此限制包括初始 scan on push （推送时扫描）（如果启用）以及任何手动扫描。

有关扫描映像时的常见问题的排查详细信息，请参阅 排查映像扫描问题 (p. 108).

开始手动扫描映像（控制台）

通过 使用以下步骤开始手动映像扫描。AWS 管理控制台.

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/repositories。
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 Repositories.
4. 在 Repositories （存储库）页面上，选择包含要扫描的映像的存储库。
5. 在 Images (映像) 页面上，选择要扫描的映像，然后选择 Scan (扫描).

开始手动扫描映像 (AWS CLI)

使用以下 AWS CLI 命令启动映像的手动扫描。您可以使用 `imageTag` 或 `imageDigest` 指定映像，这两者都可以使用 list-images CLI 命令获取。

- start-image-scan (AWS CLI)

以下示例使用映像标签。

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

以下示例使用映像摘要。

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

开始手动扫描映像 (适用于 Windows PowerShell 的 AWS 工具)

使用以下 适用于 Windows PowerShell 的 AWS 工具 命令启动映像的手动扫描。您可以使用 `ImageId_ImageTag` 或 `ImageId_ImageDigest` 指定映像，这两者都可以使用 Get-ECRImage CLI 命令获取。

- Get-ECRImageScanFinding (适用于 Windows PowerShell 的 AWS 工具)

  以下示例使用映像标签。

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

  以下示例使用映像摘要。

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

检索映像扫描查找结果

您可以检索上次完成的映像扫描的扫描结果。扫描结果根据常见漏洞和披露 (CVE) 数据库按严重性列出发现的软件漏洞。

有关扫描映像时的常见问题的排查详细信息，请参阅 排查映像扫描问题 (p. 108).

检索映像扫描结果（控制台）

通过 使用以下步骤检索映像扫描结果。AWS 管理控制台.

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/repositories。
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 Repositories.
4. 在 Repositories（存储库）页面上，选择包含要检索其扫描结果的映像的存储库。
5. 在 Images（映像）页面上的 Vulnerabilities（漏洞）列下，选择要检索其扫描结果的映像的 Details（详细信息）。

检索映像扫描结果 (AWS CLI)

通过 AWS CLI 使用以下 AWS CLI. 命令检索映像扫描结果。您可以使用 `imageTag` 或 `imageDigest` 指定映像，这两者都可以使用 list-images CLI 命令获取。

- describe-image-scan-findings (AWS CLI)

  以下示例使用映像标签。

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

  以下示例使用映像摘要。

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

检索映像扫描结果 (适用于 Windows PowerShell 的 AWS 工具)

使用以下 适用于 Windows PowerShell 的 AWS 工具 命令检索映像扫描结果。您可以使用 `ImageId_ImageTag` 或 `ImageId_ImageDigest` 指定映像，这两者都可以使用 Get-ECRImage CLI 命令获取。

- Get-ECRImageScanFinding (适用于 Windows PowerShell 的 AWS 工具)

  以下示例使用映像标签。

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

  以下示例使用映像摘要。

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

容器映像清单格式

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：

- 能够为单个映像使用多个标签。
- 支持存储 Windows 容器映像。有关更多信息，请参阅 Amazon ECR 中的将 Windows 映像推送到 Amazon Elastic Container Service Developer Guide。

Amazon ECR 映像清单转换

在 Amazon ECR 中推送和拉取镜像时，您的容器引擎客户端 (例如 Docker) 将与镜像仓库进行通信以就客户端了解的清单格式以及要用于镜像的镜像仓库达成一致。

当您使用 Docker 版本 1.9 或更早版本将映像推送到 Amazon ECR 时，映像清单格式将存储为 Docker Image Manifest V2 Schema 1。当您使用 Docker 版本 1.10 或更高版本将映像推送到 Amazon ECR 时，映像清单格式将存储为 Docker Image Manifest V2 Schema 2。

在从 Amazon ECR 按标签 拉取映像时， 将返回存储在存储库中的映像清单格式。Amazon ECR仅当客户端理解该格式时，才会返回该格式。如果客户端不理解存储的图像清单格式，则 Amazon ECR 会将图像清单转换为可理解的格式。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。下表描述了在 Amazon ECR按标签 拉取映像时， 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

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 2	无可用转换，客户端将回退到 V2 Schema 1	无需转换	已转换为 V2 Schema 2
OCI	无可用转换	已转换为 OCI	无需转换

Important

如果您按摘要 拉取映像，则没有可用的转换。您的客户端必须了解存储在 Amazon ECR 中的映像清单格式。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 映像，则无法拉取映像。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，如果您按标签 请求相同的映像，则 Amazon ECR 会将映像清单转换为客户端可以理解的格式。映像拉取成功。

在 Amazon ECS 中使用 Amazon ECR 映像

您可以在 Amazon ECS 任务定义中使用 Amazon ECR 中托管的容器映像，但需要满足以下先决条件。

- 为 Amazon ECS 任务使用 EC2 启动类型时，容器实例必须至少使用 1.7.0 版本的 Amazon ECS 容器代理。最新版本的经 Amazon ECS 优化的 AMI 在任务定义中支持 Amazon ECR 映像。有关更多信息（包括最新的经 Amazon ECS 优化的 AMI ID），请参阅 Amazon Elastic Container Service Developer Guide 中的 Amazon ECS 优化的代理版本。
- 您使用的 Amazon ECS 容器实例 IAM 角色 (`ecsInstanceRole`) 必须包含以下适用于 Amazon ECR 的 IAM 策略权限。

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

如果您使用AmazonEC2ContainerServiceforEC2Role 托管策略，则您的容器实例 IAM 角色具有适当的权限。要查看您的角色是否支持 Amazon ECR，请参阅 Amazon Elastic Container Service Developer Guide 中的 Amazon ECS 容器实例 IAM 角色。

- 在 Amazon ECS 任务定义中，确保对您的 Amazon ECR 映像使用完整的 `registry/repository:tag` 命名。例如，*`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`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-web-app:latest",
            ...
        }
    ],
    ...
}
```

在 Amazon EKS 中使用 Amazon ECR 映像

您可以将 Amazon ECR 映像与 Amazon EKS 结合使用，但需要满足以下先决条件。

- 与您的工作线程节点一起使用的 Amazon EKS 工作线程节点 IAM 角色 (`NodeInstanceRole`) 必须对 Amazon ECR 具有以下 IAM 策略权限。

```
{
    "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 入门指南中的 AWS CloudFormation 模板来创建集群和工作线程节点组，则默认情况下，这些 IAM 权限将应用于您的工作线程节点 IAM 角色。

- 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-web-app*`:`*latest*。

安装托管在上的Helm图表 Amazon ECR 配 Amazon EKS

您的Helm图表托管于 Amazon ECR 可以安装在您的 Amazon EKS 群集。以下步骤演示了此。

Prerequisites

开始之前,请确保已完成以下步骤。

- 安装Helm客户端版本3。有关更多信息,请参阅 安装头盔.
- 您已将Helm图表推送给您的 Amazon ECR 存储库。有关更多信息，请参阅推送 Helm 图表 (p. 34)。
- 您已配置 `kubectl` 与 Amazon EKS. 有关更多信息,请参阅 创建 `kubeconfig` 为 Amazon EKS 在 Amazon EKS 用户指南. 如果以下命令成功用于您的群集,则表示您已正确配置。

```
kubectl get svc
```

安装 Amazon ECR 将Helmchart托管到 Amazon EKS 群集

1. 在Helm3客户端中启用OCI支持。

```
export HELM_EXPERIMENTAL_OCI=1
```

2. 将Helm客户端验证为 Amazon ECR 注册您的Helm图表托管。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，请参阅私有注册表身份验证 (p. 13)。

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

3. 将Helm图表拉到本地缓存中。

```
helm chart pull aws_account_id.dkr.ecr.region.amazonaws.com/repository-name:mychart
```

4. 将图表导出到本地目录。在本例中,我们使用名为的目录 `charts`.

```
helm chart export aws_account_id.dkr.ecr.region.amazonaws.com/repository-name:mychart
 --destination ./charts
```

5. 安装图表。

```
helm install ecr-chart-demo ./mychart
```

输出应类似于此:

```
NAME: ecr-chart-demo
LAST DEPLOYED: Wed Sep  2 14:32:07 2020
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
```

6. 验证图表的安装。输出将是由图表部署的Kubernetes资源的YAML表现形式。

```
helm get manifest ecr-chart-demo
```

7. (可选)查看您的Helm图表在 Amazon EKS pod.

```
kubectl get pods --all-namespaces
```

8. 完成后,您可以从群集中移除图表版本。

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

Amazon Linux 容器映像

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。它可用作任何环境的 Docker 工作负载的基本映像。如果您在 Amazon Linux 中为应用程序使用 Amazon EC2 AMI，则可以使用 Amazon Linux 容器映像容器化应用程序。

可以在本地开发环境中使用 Amazon Linux 容器镜像，然后使用 AWS 将应用程序推送到 Amazon ECS. 云。有关更多信息，请参阅 在 Amazon ECS 中使用 Amazon ECR 映像 (p. 58).

Amazon Linux 容器镜像在 和 Docker Hub. 上可用。访问 AWS 开发人员论坛.可获得针对 Amazon Linux 容器镜像的支持。

从 Docker Hub 拉取 Amazon Linux 容器镜像

1. 使用 Amazon Linux 命令拉取 docker pull 容器镜像。

```
docker pull amazonlinux
```

2. （可选）在本地运行容器。

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

中的安全性Amazon Elastic Container Registry

云安全性 AWS 是最高优先级。作为 AWS 客户，您将从专为满足大多数安全敏感型组织的要求而打造的数据中心和网络架构中受益。

安全性是 AWS 和您的共同责任。TheThethe 共同责任模式 将这种情况描述为安全性 的 云和安全 在 云:

- 云安全 – AWS 负责保护运行的基础设施 AWS 服务 AWS 云。 AWS 还为您提供可安全使用的服务。第三方审计师定期测试并验证我们的安全有效性 AWS 合规计划. 了解适用于 Amazon ECR，参见 AWS 按合规计划的范围内的服务.
- 云中的安全性 – 您的责任由 AWS 您使用的服务。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

该文档帮助您了解如何在使用 时应用责任共担模型。Amazon ECR. 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。您还将了解如何使用其他 AWS 服务来帮助您监控和保护 Amazon ECR 资源。

主题

适用于 Amazon Elastic Container Registry 的 Identity and Access Management

AWS Identity and Access Management (IAM) 是一项 AWS 服务，可帮助管理员安全地控制对 AWS 资源的访问。IAM 管理员控制谁可以通过身份验证 （登录）和授权 （具有权限）以使用 Amazon ECR 资源。IAM 是一项无需额外费用即可使用的 AWS 服务。

主题

- 排查 Amazon Elastic Container Registry 身份和访问的问题 (p. 75)

Audience

如何使用 AWS Identity and Access Management (IAM) 因您可以在 Amazon ECR 中执行的操作而异。

服务用户 – 如果您使用 Amazon ECR 服务来完成工作，则您的管理员会为您提供所需的凭证和权限。当您使用更多 Amazon ECR 功能来完成工作时，您可能需要额外权限。了解如何管理访问权限可帮助您向管理员请求适合的权限。如果您无法访问 Amazon ECR 中的一项功能，请参阅排查 Amazon Elastic Container Registry 身份和访问的问题 (p. 75)。

服务管理员 – 如果您在公司负责管理 Amazon ECR 资源，则您可能具有 Amazon ECR 的完全访问权限。您有责任确定您的员工应访问哪些 Amazon ECR 功能和资源。然后，您必须向 IAM 管理员提交请求以更改您的服务用户的权限。检查此页上的信息，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 协同工作 (p. 66)。

IAM 管理员 – 如果您是 IAM 管理员，您可能希望了解有关您可以如何编写策略以管理 Amazon ECR 访问权限的详细信息。要查看您可在 IAM 中使用的基于身份的 Amazon ECR 示例策略，请参阅Amazon Elastic Container Registry 基于身份的策略示例 (p. 72)。

使用身份进行身份验证

身份验证是您使用身份凭证登录 AWS 的方法。有关使用 AWS 管理控制台 登录的更多信息，请参阅 IAM 用户指南 中的 以 IAM 用户或 根用户身份登录 AWS 管理控制台。

您必须以 AWS 账户根用户、IAM 用户身份或通过代入 IAM 角色进行身份验证（登录到 AWS）。您还可以使用公司的单一登录身份验证方法，甚至使用 Google 或 Facebook 登录。在这些案例中，您的管理员以前使用 IAM 角色设置了联合身份验证。在您使用来自其他公司的凭证访问 AWS 时，您间接地代入了角色。

要直接登录到 AWS 管理控制台，请使用您的密码和 根用户 电子邮件地址或 IAM 用户名。您可以使用 根用户 或 IAM 用户访问密钥以编程方式访问 AWS。AWS 提供了开发工具包和命令行工具，可使用您的凭证对您的请求进行加密签名。如果您不使用 AWS 工具，则必须自行对请求签名。使用签名版本 4（用于对入站 API 请求进行验证的协议）完成此操作。有关身份验证请求的更多信息，请参阅 AWS General Reference 中的签名版本 4 签名流程。

无论使用何种身份验证方法，您可能还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 IAM 用户指南 中的在 AWS 中使用 Multi-Factor Authentication (MFA)。

AWS 账户根用户

当您首次创建 AWS 账户时，最初使用的是一个对账户中所有 AWS 服务和资源有完全访问权限的单点登录身份。此身份称为 AWS 账户 根用户，可使用您创建账户时所用的电子邮件地址和密码登录来获得此身份。强烈建议您不使用 根用户 执行日常任务，即使是管理任务。请遵守仅将 根用户 用于创建首个 IAM 用户的最佳实践。然后请妥善保存 根用户 凭证，仅用它们执行少数账户和服务管理任务。

IAM 用户和群组

IAM 用户是 AWS 账户内对某个人员或应用程序具有特定权限的一个身份。IAM 用户可以拥有长期凭证，例如用户名和密码或一组访问密钥。要了解如何生成访问密钥，请参阅 IAM 用户指南 中的管理 IAM 用户的访问密钥。为 IAM 用户生成访问密钥时，请确保查看并安全保存密钥对。您以后无法找回秘密访问密钥，而是必须生成新的访问密钥对。

IAM 组 是指定一个 IAM 用户集合的身份。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，您有一个名为 IAMAdmins 的组并为该组授予管理 IAM 资源的权限。

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 AWS 账户中具有特定权限的实体。它类似于 IAM 用户，但未与特定人员关联。您可以通过切换角色，在 AWS 管理控制台中暂时代入 IAM 角色。您可以调用 AWS CLI 或 AWS API 操作或使用自定义 URL 以代入角色。有关使用角色方法的更多信息，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：

- 临时 IAM 用户权限 – IAM 用户可代入 IAM 角色，暂时获得针对特定任务的不同权限。
- 联合身份用户访问 – 您也可以不创建 IAM 用户，而是使用来自 AWS Directory Service、您的企业用户目录或 Web 身份提供商的现有身份。这些用户被称为联合身份用户。在通过身份提供商请求访问权限时，AWS 将为联合身份用户分配角色。有关联合身份用户的更多信息，请参阅 IAM 用户指南 中的联合身份用户和角色。
- 跨账户访问 – 您可以使用 IAM 角色允许其他账户中的某个人（可信任委托人）访问您账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些 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 – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - 委托人权限 – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see Actions, Resources, and Condition Keys for Amazon Elastic Container Registry in the Service Authorization Reference.
  - 服务角色 – 服务角色是服务代入以代表您执行操作的 IAM 角色。服务角色只在您的账户内提供访问权限，不能用于为访问其他账户中的服务授权。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 中的创建角色以向 AWS 服务委派权限。
  - 服务相关角色 – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- 在 Amazon EC2 上运行的应用程序–对于在 EC2 实例上运行、并发出 AWS CLI 或 AWS API 请求的应用程序，您可以使用 IAM 角色管理它们的临时凭证。这优先于在 EC2 实例中存储访问密钥。要将 AWS 角色分配给 EC2 实例并使其对该实例的所有应用程序可用，您可以创建一个附加到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 角色向在 Amazon EC2 实例上运行的应用程序授予权限。

要了解是否使用 IAM 角色或 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

您将创建策略并将其附加到 IAM 身份或 AWS 资源，以便控制 AWS 中的访问。策略是 AWS中的对象；在与标识或资源相关联时，策略定义它们的权限。您可以通过 根用户 用户或 IAM 用户身份登录，也可以代入 IAM 角色。随后，当您提出请求时，AWS 会评估相关的基于身份或基于资源的策略。策略中的权限确定是允许还是拒绝请求。大多数策略在 AWS 中存储为 JSON 文档。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 策略概述。

Administrators can use AWS JSON policies to specify who has access to what. That is, which principal can perform actions on what resources, and under what conditions.

每个 IAM 实体（用户或角色）在一开始都没有权限。换言之，默认情况下，用户什么都不能做，甚至不能更改他们自己的密码。要为用户授予执行某些操作的权限，管理员必须将权限策略附加到用户。或者，管理员可以将用户添加到具有预期权限的组中。当管理员为某个组授予访问权限时，该组内的全部用户都会获得这些访问权限。

IAM 策略定义操作的权限，无论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。具有该策略的用户可以从 AWS 管理控制台、AWS CLI 或 AWS API 获取角色信息。

基于身份的策略

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the IAM 用户指南.

基于身份的策略可以进一步归类为内联策略或托管策略。内联策略直接嵌入单个用户、组或角色中。托管策略是可以附加到 AWS 账户中的多个用户、组和角色的独立策略。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管策略或内联策略之间进行选择，请参阅 IAM 用户指南 中的在托管策略与内联策略之间进行选择 。

基于资源的策略

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM role trust policies and Amazon S3 bucket policies. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

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来自 IAM 的 AWS 托管策略。

其他策略类型

AWS 支持额外的、不太常用的策略类型。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 – 权限边界是一项高级功能，借助该功能，您可以设置基于身份的策略可以授予 IAM 实体的最大权限（IAM 用户或角色）。您可为实体设置权限边界。这些结果权限是实体的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，指定了组织或组织单位 (OU) 在 AWS Organizations 中的最大权限。AWS Organizations 是一项服务，用于分组和集中管理您的企业拥有的多个 AWS 账户。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 限制成员账户中实体（包括每个 AWS 账户根用户）的权限。有关 组织 和 SCP 的更多信息，请参阅 AWS Organizations 用户指南 中的 SCP 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身份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 AWS 如何确定在涉及多个策略类型时是否允许请求，请参阅 IAM 用户指南 中的策略评估逻辑。

Amazon Elastic Container Registry 如何与 IAM 协同工作

在使用 IAM 管理对 Amazon ECR 的访问之前，您应了解哪些 IAM 功能可与 Amazon ECR 协同工作。要概括了解 Amazon ECR 及其他 AWS 服务如何与 IAM 协同工作，请参阅 AWS 中的IAM可与 协同工作的 IAM 用户指南 服务。

主题

Amazon ECR 基于身份的策略

使用 IAM 基于身份的策略，您可以指定允许或拒绝操作和资源，以及指定在什么条件下允许或拒绝操作。Amazon ECR 支持特定操作、资源和条件键。要了解您在 JSON 策略中使用的所有元素，请参阅 IAM 中的 JSON 策略元素参考。IAM 用户指南

Actions

Administrators can use AWS JSON policies to specify who has access to what. That is, which principal can perform actions on what resources, and under what conditions.

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相关操作的权限。

中的策略操作在操作前使用以下前缀：Amazon ECR`ecr:`。 例如，要授予某人使用 Amazon ECR Amazon ECR API 操作创建 `CreateRepository` 存储库的权限，您应将 `ecr:CreateRepository` 操作纳入其策略中。策略语句必须包括 `Action` 或 `NotAction` 元素。Amazon ECR 定义了自己的一组操作，这些操作描述了可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，请包括以下操作：

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

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

Resources

Administrators can use AWS JSON policies to specify who has access to what. That is, which principal can perform actions on what resources, and under what conditions.

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。作为最佳做法，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 ARNs 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 ARNs。

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

要查看 Amazon ECR 资源类型及其 ARNs 的列表，请参阅 中的 Amazon Elastic Container Registry 定义的资源。IAM 用户指南要了解您可以在哪些操作中指定每个资源的 ARN，请参阅 Amazon Elastic Container Registry 定义的操作。

条件键

Administrators can use AWS JSON policies to specify who has access to what. That is, which principal can perform actions on what resources, and under what conditions.

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 AWS 使用逻辑 OR 运算来评估条件。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，仅当 IAM 用户使用其 IAM 用户名进行标记时，您才可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 支持全局条件键和特定于服务的条件键。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南 中的 AWS 全局条件上下文键。

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 中的 AWS 全局条件上下文键。IAM 用户指南

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 (p. 74)。

要查看 Amazon ECR 条件键的列表，请参阅 中的 Amazon Elastic Container Registry 定义的条件键。IAM 用户指南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。

Examples

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 (p. 72)。

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 对于 Amazon ECR 存储库支持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 。将跨账户委托人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，还必须授予委托人实体对资源的访问权限。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委托人授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅 中的 IAM 角色与基于资源的策略有何不同。IAM 用户指南

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，请参阅存储库策略 (p. 23)。

Examples

要查看 Amazon ECR 基于资源的策略的示例，请参阅存储库策略示例 (p. 25)。

基于 Amazon ECR 标签的授权

您可以将标签附加到 Amazon ECR 资源或将请求中的标签传递到 Amazon ECR。要基于标签控制访问，您需要使用 、 或 `ecr:ResourceTag/key-name` 条件键在策略的`aws:RequestTag/key-name`条件元素`aws:TagKeys`中提供标签信息。有关标记 Amazon ECR 资源的更多信息，请参阅标记 Amazon ECR 存储库 (p. 28)。

要查看基于身份的策略（用于基于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 (p. 74)。

Amazon ECR IAM 角色

IAM 角色是 AWS 账户中具有特定权限的实体。

将临时凭证用于 Amazon ECR

您可以使用临时凭证进行联合身份登录，代入 IAM 角色或代入跨账户角色。您可以通过调用 AWS STS API 操作（如 AssumeRole 或 GetFederationToken）获取临时安全凭证。

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您的 IAM 账户中，并由该服务拥有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

Amazon ECR 不支持服务相关角色。

Amazon ECR 托管策略

Amazon ECR 提供了一些托管策略，您可以将它们附加到 IAM 用户或 EC2 实例，以实现对 Amazon ECR 资源和 API 操作的不同级别的控制。您可以直接应用这些策略，或者也可以使用它们作为自行创建策略的起点。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 Reference 中的操作。

主题

AmazonEC2ContainerRegistryFullAccess

对于希望为 IAM 用户或角色提供完全管理员访问权限以管理其使用 Amazon ECR 的客户，此托管策略是其起点。Amazon ECR 生命周期策略功能使客户可以指定存储库中映像的生命周期管理。生命周期策略事件作为 CloudTrail 事件报告，并且 Amazon ECR 与 AWS CloudTrail 集成以直接在 Amazon ECR 控制台中显示客户的生命周期策略事件。`AmazonEC2ContainerRegistryFullAccess` 托管 IAM 策略包含促进此行为的 `cloudtrail:LookupEvents` 权限。

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

AmazonEC2ContainerRegistryPowerUser

此托管策略授予对 Amazon ECR 的高级用户访问权限，从而允许对存储库进行读写访问，但不允许用户删除存储库或更改应用于存储库的策略文档。

```
{
 "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:InitiateLayerUpload",
   "ecr:UploadLayerPart",
   "ecr:CompleteLayerUpload",
   "ecr:PutImage"
  ],
```

```
  "Resource": "*"
 }]
}
```

AmazonEC2ContainerRegistryReadOnly

此托管策略授予对 Amazon ECR 的只读访问权限，例如，能够列出存储库和存储库中的映像，还能通过 Docker CLI 从 Amazon ECR 中拉取映像。

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

对 使用服务相关角色Amazon ECR

Amazon Elastic Container Registry （ Amazon ECR ） 使用 AWS Identity and Access Management （ IAM ） 服务相关角色提供对复制资源的访问权限。服务相关角色是一种与 IAM 直接关联的独特类型的 Amazon ECR. 角色。服务相关角色由 预定义Amazon ECR。它包括该服务为注册表支持跨区域和跨账户映像复制所需的所有权限。为注册表配置复制后，系统会代表您自动创建 服务相关角色。有关更多信息，请参阅私有注册表设置 (p. 15)。 />。

服务相关角色使使用 设置复制Amazon ECR变得更轻松。这是因为，通过使用它，您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，除非另行定义，否则仅 Amazon ECR 可以代入其角色。定义的权限包括信任策略和权限策略。权限策略不能附加到任何其他 IAM 实体。

只有在您的注册表上禁用复制后，才能删除服务相关角色。这可确保您不会无意中删除 Amazon ECR 的映像复制权限。

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. 在此链接到页面上，查找 Service-linked role （ 服务相关角色 ） 列中具有 Yes （ 是 ） 的服务。选择包含链接的 Yes （ 是 ） 以查看该服务的相关服务相关角色文档。

的服务相关角色权限Amazon ECR

Amazon ECR 使用名为 的服务相关角色AWSServiceRoleForECRReplication–Allows Amazon ECR to replicate images across multiple accounts.。

AWSServiceRoleForECRReplication 服务相关角色信任以下服务代入该角色：

- `replication.ecr.amazonaws.com`

角色权限策略Amazon ECR允许 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。

您必须配置权限以允许 IAM 实体（例如，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 创建服务相关角色Amazon ECR

无需手动创建 Amazon ECR 服务相关角色。当您在 AWS 管理控制台、 AWS CLI或 AWS API 中为您的注册表配置复制设置时， 会为您Amazon ECR创建服务相关角色。

如果您删除此服务相关角色并需要再次创建它，则可以使用相同的过程在您的账户中重新创建该角色。当您为注册表配置复制设置时， 会再次为您Amazon ECR创建服务相关角色。

编辑 的服务相关角色Amazon ECR

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。但是，您可以使用 编辑角色的说明。IAM. 有关更多信息，请参阅 IAM 用户指南 中的编辑服务相关角色.

删除 的服务相关角色 Amazon ECR

如果您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注册表的复制配置，然后才能手动删除服务相关角色。

Note

如果您在Amazon ECR服务仍在使用角色时尝试删除资源，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 Amazon ECR 所用的 AWSServiceRoleForECRReplication 资源

1. 通过以下网址打开 Amazon ECR 控制台：https://console.amazonaws.cn/ecr/。
2. 从导航栏中，选择设置复制配置的区域。
3. 在导航窗格中，选择 Registry settings （注册表设置）。
4. 同时选择 Cross-Region replication （跨区域复制） 和 Cross-account replication settings （跨账户复制设置）。
5. 选择 Save.

使用 手动删除服务相关角色 IAM

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，请参阅 IAM 用户指南 中的删除服务相关角色.

Amazon ECR 服务相关角色的受支持区域

Amazon ECR 支持在服务可用的所有区域中使用服务相关角色。有关更多信息，请参阅AWS Regions and Endpoints.

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

默认情况下，IAM 用户和角色没有创建或修改 Amazon ECR 资源的权限。它们还无法使用 AWS 管理控制台、AWS CLI 或 AWS API 执行任务。IAM 管理员必须创建 IAM 策略，为用户和角色授予权限，以便对他们所需的指定资源执行特定的 API 操作。然后，管理员必须将这些策略附加到需要这些权限的 IAM 用户或组。

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 中的在 JSON 选项卡上创建策略。IAM 用户指南

主题

策略最佳实践

基于身份的策略非常强大。它们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 – 要快速开始使用 Amazon ECR，请使用 AWS 托管策略，为您的员工授予他们所需的权限。这些策略已在您的账户中提供，并由 AWS 维护和更新。有关更多信息，请参阅 IAM 用户指南 中的利用 AWS 托管策略开始使用权限。
- 授予最低权限 – 创建自定义策略时，仅授予执行任务所需的许可。最开始只授予最低权限，然后根据需要授予其他权限。这样做比起一开始就授予过于宽松的权限而后再尝试收紧权限来说更为安全。有关更多信息，请参阅 IAM 用户指南 中的授予最小权限。
- 为敏感操作启用 MFA – 为增强安全性，要求 IAM 用户使用多重身份验证 (MFA) 来访问敏感资源或 API 操作。有关更多信息，请参阅 IAM 用户指南 中的在 AWS 中使用多重身份验证 (MFA)。
- 使用策略条件来增强安全性 – 在切实可行的范围内，定义基于身份的策略在哪些情况下允许访问资源。例如，您可编写条件来指定请求必须来自允许的 IP 地址范围。您也可以编写条件，以便仅允许指定日期或时间范围内的请求，或者要求使用 SSL 或 MFA。有关更多信息，请参阅 IAM 用户指南 中的 IAM JSON 策略元素：Condition。

使用 Amazon ECR 控制台

要访问 Amazon Elastic Container Registry 控制台，您必须拥有一组最低的权限。这些权限必须允许您列出和查看有关您的 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（IAM 用户或角色），控制台将无法按预期正常运行。

要确保这些实体仍然可以使用 Amazon ECR 控制台，请将 `AmazonEC2ContainerRegistryReadOnly`AWS 托管策略附加到这些实体。有关更多信息，请参阅 https://docs.amazonaws.cn/IAM/latest/UserGuide/id_users_change-permissions.html#users_change_permissions-add-console 中的向用户添加权限IAM 用户指南：

```
{
```

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

对于只需要调用 AWS CLI 或 AWS API 的用户，您无需为其提供最低控制台权限。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

此示例显示您可以如何创建策略，以便允许 IAM 用户查看附加到其用户身份的内联和托管策略。此策略包括在控制台上完成此操作或者以编程方式使用 AWS CLI 或 AWS API 所需的权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws-cn: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，您希望授予 IAM 账户中的 AWS 用户访问其中一个 Amazon ECR 存储库（`my-repo`）的权限。 您还想要允许用户推送、拉取和列出映像。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"ListImagesInRepository",
         "Effect":"Allow",
         "Action":[
            "ecr:ListImages"
         ],
         "Resource":"arn:aws-cn: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-cn:ecr:us-east-1:123456789012:repository/my-repo"
      }
   ]
}
```

使用基于标签的访问控制

利用 Amazon ECR CreateRepository API 操作，您可以在创建存储库时指定标签。有关更多信息，请参阅标记 Amazon ECR 存储库 (p. 28)。

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 策略使用基于标签的访问控制。示例如下。

以下策略仅允许 IAM 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 IAM 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"
                }
            }
        }
    ]
}
```

排查 Amazon Elastic Container Registry 身份和访问的问题

使用以下信息可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

如果 AWS 管理控制台 告诉您，您无权执行某个操作，则必须联系您的管理员寻求帮助。您的管理员是指为您提供用户名和密码的那个人。

当 `mateojackson` IAM 用户尝试使用控制台查看有关存储库的详细信息，但不具有 `ecr:DescribeRepositories` 权限时，会发生以下示例错误。

```
User: arn:aws-cn:iam::123456789012:user/mateojackson is not authorized to perform:
 ecr:DescribeRepositories on resource: my-repo
```

在这种情况下，Mateo 请求管理员更新其策略，以允许他使用 `ecr:DescribeRepositories` 操作访问 `my-repo` 资源。

我无权执行 iam:PassRole

如果您收到错误消息，提示您无权执行 `iam:PassRole` 操作，则必须联系您的管理员寻求帮助。您的管理员是指为您提供用户名和密码的那个人。请求那个人更新您的策略，以便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到该服务，而不是创建新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操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws-cn:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在这种情况下，Mary 请求她的管理员来更新其策略，以允许她执行 `iam:PassRole` 操作。

我想要查看我的访问密钥

创建 IAM 用户访问密钥之后，您可以随时查看您的访问密钥 ID。但是，您无法再查看您的秘密访问密钥。如果您丢失了私有密钥，则必须创建一个新的访问密钥对。

访问密钥包含两部分：访问密钥 ID（例如 `AKIAIOSFODNN7EXAMPLE`）和秘密访问密钥（例如 `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`）。与用户名和密码一样，您必须同时使用访问密钥 ID 和秘密访问密钥对请求执行身份验证。像对用户名和密码一样，安全地管理访问密钥。

> **Important**
>
> 请不要向第三方提供访问密钥，甚至为了帮助找到您的规范用户 ID 也不能提供。如果您这样做，可能会向某人提供对您的账户的永久访问权限。

当您创建访问密钥对时，系统会提示您将访问密钥 ID 和秘密访问密钥保存在一个安全位置。秘密访问密钥仅在您创建它时可用。如果您丢失了秘密访问密钥，则必须向您的 IAM 用户添加新的访问密钥。您最多可拥有两个访问密钥。如果您已有两个密钥，则必须删除一个密钥对，然后再创建新的密钥。要查看说明，请参阅 IAM 用户指南 中的管理访问密钥。

我是管理员并希望允许其他人访问 Amazon ECR

要允许其他人访问 Amazon ECR，您必须为需要访问权限的人员或应用程序创建 IAM 实体（用户或角色）。他们（它们）将使用该实体的凭证访问 AWS。然后，您必须将策略附加到实体，以便在 Amazon ECR 中为他们（它们）授予正确的权限。

要立即开始使用，请参阅 IAM 用户指南 中的创建您的第一个 IAM 委托用户和组。

我想要允许我的 AWS 账户之外的用户访问我的 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，可以代入角色。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 协同工作 (p. 66)。
- 要了解如何向您拥有的 AWS 账户中的资源提供访问权限，请参阅 IAM 用户指南 中的对您拥有的 AWS 账户中的 IAM 用户提供访问权限。
- 要了解如何向第三方 AWS 账户提供对您的资源的访问权限，请参阅 IAM 用户指南 中的向第三方拥有的 AWS 账户提供访问权限。
- 要了解如何通过联合身份验证提供访问权限，请参阅 IAM 用户指南 中的向经过外部身份验证的用户（联合身份验证）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访问之间的差别，请参阅 IAM 用户指南 中的 IAM 角色和基于资源的策略有何不同。

中的数据保护Amazon ECR

AWS 责任共担模式适用于 Amazon Elastic Container Service 中的数据保护。如该模式中所述，AWS 负责保护运行所有 AWS 云的全球基础设施。您负责维护对托管在此基础设施上的内容的控制。此内容包括您所使用的 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。

出于数据保护目的，我们建议您保护 AWS 账户凭证并使用 AWS Identity and Access Management (IAM) 设置单独的用户账户。这仅向每个用户授予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您的数据：

- 对每个账户使用 Multi-Factor Authentication (MFA)。
- 使用 SSL/TLS 与 AWS 资源进行通信。建议使用 TLS 1.2 或更高版本。
- 使用 AWS CloudTrail 设置 API 和用户活动日志记录。
- 使用 AWS 加密解决方案以及 AWS 服务中的所有默认安全控制。
- 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 中的个人数据。
- 如果在通过命令行界面或 API 访问 AWS 时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 终端节点。有关可用的 FIPS 终端节点的更多信息，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敏感的可识别信息（例如您客户的账号）放入自由格式字段（例如 Name (名称) 字段）。这包括使用控制台、API、AWS CLI 或 AWS 开发工具包处理 Amazon ECS 或其他 AWS 服务时。您输入到 Amazon ECS 或其他服务中的任何数据都可能被选取以包含在诊断日志中。当您向外部服务器提供 URL 时，请勿在 URL 中包含凭证信息来验证您对该服务器的请求。

主题

静态加密

Amazon ECR 将图像存储在 Amazon S3 管理的 Amazon ECR 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，该加密密钥使用 AES-256 加密算法对静态数据进行加密。

这不需要您执行任何操作，并且无需额外付费。有关更多信息，请参阅 中的使用具有 Amazon S3 托管加密密钥的服务器端加密 （SSE-S3 Amazon Simple Storage Service 开发人员指南保护数据。

要更好地控制Amazon ECR存储库的加密，您可以使用具有 AWS Key Management Service （AWS KMS）中存储的客户主密钥 （CMKs） 的服务器端加密。使用 AWS KMS 加密数据时，您可以使用由 管理的默认AWS托管 Amazon ECRCMK，也可以指定您自己的 CMK（称为客户托管 CMK）。有关更多信息，请参阅中的使用具有 AWS Key Management Service 中存储的服务器端加密 （SSE-KMS）CMKs Amazon Simple Storage Service 开发人员指南保护数据。

每个Amazon ECR存储库都有一个加密配置，在创建存储库时设置。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，请参阅创建存储库 (p. 20).

在创建存储库时启用AWS KMS加密，使用 CMK 加密存储库的内容。此外， 向 CMK Amazon ECR 添加AWS KMS授权，并将Amazon ECR存储库作为被授权委托人。

以下内容高度了解 如何与 Amazon ECR 集成AWS KMS以加密和解密您的存储库：

1. 创建存储库时， 会向 Amazon ECR 发送DescribeKey调用AWS KMS以验证和检索加密配置中指定的 CMK 的 Amazon 资源名称 （ARN）。
2. Amazon ECR 向 发送两个CreateGrant请求AWS KMS以在 CMK 上创建授权Amazon ECR，以允许 使用数据密钥加密和解密数据。
3. 在推送映像时，会向 发出GenerateDataKey请求AWS KMS，以指定用于加密映像层和清单的 CMK。
4. AWS KMS 生成新的数据密钥，使用指定的 CMK 对其进行加密，并发送加密的数据密钥以与映像层元数据和映像清单一起存储。
5. 拉取映像时，会向 发出 Decrypt 请求AWS KMS，并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的数据密钥发送到 Amazon S3。
7. 中的数据密钥，用于在拉取图像层之前解密图像层。
8. 删除存储库后， 向 Amazon ECR 发送两个RetireGrant请求AWS KMS以停用为存储库创建的授权。

Considerations

在将AWS KMS加密与 结合使用时，应考虑以下几点Amazon ECR。

- 如果您使用 KMS 加密创建Amazon ECR存储库，并且未指定 CMK， Amazon ECR 默认AWS使用具有别名的 `aws/ecr`托管 CMK。当您首次创建启用了 KMS 加密的存储库时，将在您的账户中创建此 CMK。
- 当您将 KMS 加密与您自己的 CMK 结合使用时，密钥必须与您的 存储库位于同一区域。
- AWS KMS 强制实施每个 CMK 500 个授权的限制。因此，每个 CMK 的Amazon ECR存储库数量限制为 500 个。
- 不应撤销 代表您Amazon ECR创建的授权。如果您撤销授予 使用您账户中的Amazon ECR密钥AWS KMS的权限的授权，则 Amazon ECR 无法访问此数据，对推送到存储库的新映像进行加密，或在推送时解密它们。当您撤销 的授权时Amazon ECR，更改将立即生效。要撤销访问权限，您应删除存储库，而不是撤销授权。删除存储库后， 将代表您Amazon ECR停用授权。
- 使用 AWS KMS 密钥会产生关联的成本。有关更多信息，请参阅 AWS Key Management Service 定价.

所需的 IAM 权限

使用 创建或删除具有服务器端加密的 Amazon ECR 存储库时AWS KMS，所需的权限取决于您使用的特定客户主密钥 （CMK）。

使用适用于 的 IAM 托管 CMK 时所需的AWS权限 Amazon ECR

默认情况下，当为 AWS KMS 存储库启用Amazon ECR加密但未指定 CMK 时AWS，将使用适用于 的Amazon ECR托管 CMK。当 的 AWS托管 CMK Amazon ECR 用于加密存储库时，有权创建存储库的任何委

托人也可以对存储库启用AWS KMS加密。但是，删除存储库的IAM委托人必须具有 kms:RetireGrant 权限。这样可以停用在创建存储库时添加到AWS KMS密钥的授权。

以下示例IAM策略可作为内联策略添加到用户，以确保他们具有删除启用了加密的存储库所需的最低权限。用于加密存储库的 AWS KMS 密钥可以使用 资源参数指定。

```
{
    "Version": "2012-10-17",
    "Id": "ecr-kms-permissions",
    "Statement": [
        {
            "Sid": "Allow access to retire the grants associated with the key",
            "Effect": "Allow",
            "Action": [
                "kms:RetireGrant"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/b8d9ae76-080c-4043-92EXAMPLE"
        }
    ]
}
```

使用客户托管 CMK 时所需的IAM权限

在创建使用客户托管 CMK 启用了AWS KMS加密的存储库时，CMK 密钥策略和创建存储库的用户或角色的IAM策略都有必需的权限。

在创建您自己的 CMK 时，您可以使用 AWS KMS 创建的默认密钥策略，也可以指定您自己的密钥策略。要确保账户所有者可管理客户托管 CMK，该 CMK 的密钥策略应允许账户的根用户执行所有 AWS KMS 操作。其他范围限定的权限可以添加到密钥策略中，但至少应该向根用户授予管理 CMK 的权限。要仅允许将 CMK 用于源自 的请求Amazon ECR，您可以将 kms：ViaService 条件键与 ecr.<region>.amazonaws.com 值结合使用。

以下示例密钥策略向拥有 CMK 的 AWS 账户（根用户）授予对 CMK 的完全访问权限。有关此示例密钥策略的更多信息，请参阅 中的允许访问 AWS 账户并启用 IAM AWS Key Management Service Developer Guide策略。

```
{
    "Version": "2012-10-17",
    "Id": "ecr-key-policy",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        }
    ]
}
```

创建存储库IAM的用户IAM、角色或AWS账户必须具有 kms:CreateGrant、 kms:RetireGrant和 kms:DescribeKey 权限以及必要的Amazon ECR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策略中。kms:CreateGrant 和 kms:DescribeKey 权限可以添加到 CMK 的密钥策略中，也可以添加到创建存储库的用户或角色的 IAM 策略中。有关AWS KMS权限工作原理的更多信息，请参阅 中的 AWS KMS API AWS Key Management Service Developer Guide权限：操作和资源参考。

以下示例IAM策略可作为内联策略添加到用户，以确保他们具有创建已启用加密的存储库以及在存储库完成后删除存储库所需的最低权限。用于加密存储库的 AWS KMS 密钥可以使用 资源参数指定。

```
{
    "Version": "2012-10-17",
    "Id": "ecr-kms-permissions",
    "Statement": [
        {
            "Sid": "Allow access to create and retire the grants associated with the key as
 well as describe the key",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:RetireGrant",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/b8d9ae76-080c-4043-92EXAMPLE"
        }
    ]
}
```

允许用户在创建存储库时在控制台CMKs中列出

使用 Amazon ECR 控制台创建存储库时，您可以授予 权限，使用户能够在为存储库启用加密时列出CMKs区域中托管的客户。以下IAM策略示例显示了使用 控制台时列出 CMKs 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 的交互

您可以使用 AWS CloudTrail 跟踪 Amazon ECR 代表您AWS KMS发送到 的请求。CloudTrail 日志中的日志条目包含一个加密上下文键，以便更轻松地识别它们。

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。在请求中包含加密上下文以加密数据时，AWS KMS 以加密方式将加密上下文绑定到加密的数据。要解密数据，您必须传入相同的加密上下文。

在发送到 的 GenerateDataKey 和 Decrypt 请求中AWS KMS， Amazon ECR 使用具有两个用于标识正在使用的存储库和–存储桶的名称Amazon S3值对的加密上下文。如下例所示。名称不会发生变化，但组合的加密上下文值对于每个值将不同。

```
"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核记录和日志中标识这些加密操作（如 AWS CloudTrail 和 Amazon CloudWatch Logs），并将加密上下文用作在策略和授权中进行授权的条件。

Amazon ECR 加密上下文包含两个名称–值对。

- s3 – 第一个名称–值对标识存储桶。密钥为 `aws:s3:arn`。 该值是Amazon S3存储桶的 Amazon 资源名称（ARN）。

```
"aws:s3:arn": "ARN of an Amazon S3 bucket"
```

例如，如果存储桶的 ARN 为 `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 资源名称（ARN）。密钥为 `aws:ecr:arn`。该值是存储库的 ARN。

```
"aws:ecr:arn": "ARN of an Amazon ECR repository"
```

例如，如果存储库的 ARN 为 `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"
```

Troubleshooting

使用 控制台删除 Amazon ECR 存储库时，如果已成功删除存储库，但 Amazon ECR 无法停用添加到存储库 CMK 的授权，您将收到以下错误。

```
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，您可以自行停用存储库AWS KMS的授权。

手动停用存储库AWS KMS的授权

1. 列出用于存储库的AWS KMS密钥的授权。值`key-id`包含在您从 控制台收到的错误中。您还可以使用 `list-keys` 命令列出在您的账户的特定区域中管理CMKs的 AWS 托管账户CMKs和客户。

```
aws kms list-grants \
     --key-id b8d9ae76-080c-4043-9237-c815bfc21dfc
     --region us-west-2
```

输出包含一个 `EncryptionContextSubset` ，其中包含存储库的 Amazon 资源名称（ARN）。这可用于确定添加到密钥中的哪个授权是您要停用的授权。在下一步中停用授权时，将使用 `GrantId` 值。

2. 停用为存储库添加的AWS KMS密钥的每个授权。替换 的值 *GrantId* 替换为上一步输出中的授权的 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

作为多个 AWS 合规性计划的一部分，第三方审计员将评估 Amazon Elastic Container Registry 的安全性和合规性。其中包括 SOC、PCI、HIPAA 等。

有关特定合规性计划范围内的 AWS 服务列表，请参阅合规性计划范围内的 AWS 服务。有关常规信息，请参阅 AWS 合规性计划。

您可以使用 AWS Artifact 下载第三方审计报告。有关更多信息，请参阅下载 AWS 构件中的报告。

您在使用 Amazon ECR 时的合规性责任由您数据的敏感性、您公司的合规性目标以及适用的法律法规决定。AWS 提供以下资源来帮助满足合规性：

- 安全性与合规性快速入门指南安全性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，并提供了在 AWS 上部署基于安全性和合规性的基准环境的步骤。
- 《设计符合 HIPAA 安全性和合规性要求的架构》白皮书 – 此白皮书介绍公司如何使用 AWS 创建符合 HIPAA 标准的应用程序。
- AWS 合规性资源 – 此业务手册和指南集合可能适用于您的行业和位置。
- AWS Config 开发人员指南中的使用规则评估资源 – 此 AWS Config 服务评估您的资源配置对内部实践、行业指南和法规的遵循情况。
- AWS Security Hub – 此 AWS 服务提供了 AWS 中安全状态的全面视图，可帮助您检查是否符合安全行业标准和最佳实践。

Amazon Elastic Container Registry 中的基础设施安全性

作为一项托管服务，Amazon Elastic Container Registry 由 Amazon Web Services：安全流程概述白皮书中所述的 AWS 全球网络安全程序提供保护。

您可以使用 AWS 发布的 API 调用通过网络访问 Amazon ECR。客户端必须支持传输层安全性 (TLS) 1.0 或更高版本。建议使用 TLS 1.2 或更高版本。客户端还必须支持具有完全向前保密 (PFS) 的密码套件，例如 Ephemeral Diffie-Hellman (DHE) 或 Elliptic Curve Ephemeral Diffie-Hellman (ECDHE)。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 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。事实上，这隔离了在 AWS 网络中仅从特定 VPC 到给定 Amazon ECR 资源的网络访问。有关更多信息，请参阅 Amazon ECR 接口 VPC 终端节点（AWS PrivateLink）(p. 82)。

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 提供支持AWS，您可以使用该技术PrivateLinkAmazon ECR通过私有 IP 地址私下访问 APIs 。 AWSPrivateLink 将

VPC 和 之间的所有网络流量限制Amazon ECR在 Amazon 网络以内。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 中的 VPC Amazon VPC 用户指南 终端节点。

关于 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。

- 要允许使用 Amazon ECS 启动类型的 EC2 任务从 Amazon ECR 中提取私有镜像，请确保也为 Amazon ECS. 创建接口 VPC 终端节点。有关更多信息，请参阅 中的AWS接口 VPC PrivateLink 终端节点（Amazon Elastic Container Service Developer Guide）。

  Important

  使用 Amazon ECS 启动类型的 Fargate 任务不需要 Amazon ECS 接口 VPC 终端节点。
- Amazon ECS 使用 Fargate 启动类型和平台版本 1.3.0 或更早版本的 任务只需要 com.amazonaws。***region***.ecr.dkr Amazon ECR VPC 终端节点和Amazon S3网关终端节点以利用此功能。
- Amazon ECS 使用 Fargate 启动类型和平台版本 1.4.0 或更高版本的 任务需要 com.amazonaws。***region***.ecr.dkr 和 com.amazonaws。***region***.ecr.api Amazon ECR VPC 终端节点以及用于利用此功能的Amazon S3网关终端节点。
- 使用 Amazon ECS 启动类型从 Fargate 拉取容器映像的 Amazon ECR 任务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 中的通过接口终端节点拉取 Amazon ECR 映像的 Fargate 任务的可选 IAM Amazon Elastic Container Service Developer Guide 权限。
- 使用 Amazon ECS 启动类型的 Fargate 任务（从 Amazon ECR 中拉取容器映像，同时使用 `awslogs` 日志驱动程序将日志信息发送到 CloudWatch Logs），需要 CloudWatch Logs VPC 终端节点。有关更多信息，请参阅创建 CloudWatch Logs 终端节点 (p. 86).
- 附加到 VPC 终端节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 终端节点当前不支持跨区域请求。确保在计划向 发出 API 调用的同一区域中创建 VPC 终端节点。Amazon ECR.
- VPC 终端节点仅通过 支持 Amazon 提供的 DNS。Amazon Route 53. 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https://docs.amazonaws.cn/AmazonVPC/latest/UserGuide/VPC_DHCP_Options.html中的 Amazon VPC 用户指南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 网关终端节点时，其连接可能会短暂中断。如果要避免此中断，请创建一个使用 Amazon S3 网关终端节点的新 VPC，然后将 Amazon ECS 集群及其容器迁移到该新 VPC。

Windows 映像的注意事项

基于 Windows 操作系统的图像包括受许可证限制的构件，无法分发。默认情况下，当您将 Windows 映像推送到 Amazon ECR 存储库时，不会推送包含这些构件的层，因为它们被视为外部层。当 Microsoft 提供构件时，外部层是从 Microsoft Azure 基础设施中检索的。因此，除了创建 VPC 终端节点之外，还需要执行其他步骤，以使容器能够从 Azure 中提取这些外部层。

在 Docker 守护程序中使用 Amazon ECR `--allow-nondistributable-artifacts` 标志将 Windows 映像推送到 时，可以覆盖此行为。启用后，此标记将推送许可层Amazon ECR，允许Amazon ECR通过 VPC 终端节点从 中拉取这些映像，而无需对 Azure 进行额外访问。

Important

使用 `--allow-nondistributable-artifacts` 标志并不妨碍您遵守 Windows 容器基本映像许可证的条款；您无法发布 Windows 内容以进行公开或第三方重新分配。您自己的环境中允许使用 。

要启用此标志用于 Docker 安装，您必须修改 Docker 守护程序配置文件，该文件通常根据 Docker 安装在 Docker Engine 部分下的设置或首选项菜单中进行配置，或者通过直接编辑 `C:\ProgramData\docker\config\daemon.json` 文件来配置。

以下是所需配置的示例。将 值替换为您要将映像推送到的存储库 URI。

```
{
    "allow-nondistributable-artifacts": [
        "111122223333.dkr.ecr.us-west-2.amazonaws.com"
    ]
}
```

修改 Docker 守护程序配置文件后，您必须先重新启动 Docker 守护程序，然后再尝试推送您的映像。通过验证基本层是否已推送到您的存储库，确认推送正常工作。

Note

Windows 映像的基本层很大。层大小将使 中的Amazon ECR这些映像的推送时间更长，并产生额外的存储成本。出于这些原因，我们建议仅在严格需要使用此选项来减少构建时间和持续存储成本时使用此选项。例如`mcr.microsoft.com/windows/servercore`，图像GiB在压缩时的大小大约为 Amazon ECR1.7。

为 创建 VPC 终端节点Amazon ECR

要为服务创建 VPC 终端节点Amazon ECR，请使用 中的https://docs.amazonaws.cn/vpc/latest/userguide/vpce-interface.html#create-interface-endpoint创建接口终端节点过程Amazon VPC 用户指南。

使用 Amazon ECS 启动类型的 EC2 任务需要Amazon ECR 终端节点和 Amazon S3 网关终端节点。

Amazon ECS 使用 Fargate 启动类型和平台版本 1.3.0 或更早版本的 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。

Amazon ECS 使用 Fargate 启动类型和平台版本 1.4.0 或更高版本的 任务需要 com.amazonaws。***region***.ecr.dkr 和 com.amazonaws。***region***.ecr.api 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 注册表 APIs。 Docker 客户端命令（如 `push` 和 `pull` ）使用此终端节点。

当您创建 com.amazonaws 时。***region***.ecr.dkr 终端节点，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 VPC 控制台中选择了 Enable Private DNS Name （启用私有 DNS 名称）选项。

com.amazonaws.***region***.ecr.api

Note

指定的 *region* 表示 支持的 AWS 区域的区域标识符Amazon ECR，例如 `us-east-2` 的 美国东部（俄亥俄）区域。

此终端节点用于对 Amazon ECR API 执行的调用。API 操作（如 DescribeImages 和 CreateRepositories）转到此终端节点。

当 com.amazonaws 时。***region***已创建 .ecr.api 终端节点，您可以选择启用私有 DNS 主机名。通过在创建 VPC 终端节点时在 VPC 控制台中选择 Enable Private DNS Name （启用私有 DNS 名称）来启用此设置。如果您为 VPC 终端节点启用私有 DNS 主机名，请将开发工具包或 AWS CLI 更新到最新版本，以便在使用开发工具包或 AWS CLI 时无需指定终端节点 URL。

如果您启用私有 DNS 主机名并使用 2019 年 1 月 24 日版之前发布的开发工具包或 AWS CLI 版本，则必须使用 `--endpoint-url` 参数指定接口终端节点。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 ECS 拉取私有映像的 Amazon ECR 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 来存储您的映像层。当容器从 Amazon ECR 下载映像时，它们必须访问 Amazon ECR 才能获取映像清单，然后 Amazon S3 才能下载实际映像层。以下是包含每个 Docker 映像层的 Amazon S3 存储桶的 Amazon 资源名称 (ARN)：

```
arn:aws:s3:::prod-region-starport-layer-bucket/*
```

使用 中的创建网关终端节点过程为 Amazon VPC 用户指南 创建以下Amazon S3网关终端节点Amazon ECR。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 策略文档来限制对服务的访问。可使用 Full Access 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适用于此策略。如果要将 Amazon S3 存储桶访问权限限制为使用 Amazon ECR 所需的最低权限，请参阅Amazon S3 的最低 Amazon ECR 存储桶权限 (p. 85).

Amazon S3 的最低 Amazon ECR 存储桶权限

Amazon S3 网关终端节点使用 IAM 策略文档来限制对服务的访问。要仅允许 Amazon S3 的最低 Amazon ECR 存储桶权限，请在为终端节点创建 Amazon S3 策略文档时，限制对 Amazon ECR 使用的 IAM 存储桶的访问权限。

下表描述了 Amazon S3 所需的 Amazon ECR. 存储桶策略权限。

许可	描述
`arn:aws:s3:::prod-region-starport-layer-bucket/*`	提供对包含每个 Docker 映像层的 Amazon S3 存储桶的访问权限。表示 AWS 支持的 Amazon ECR 区域的区域标识符，如`us-east-2`的 美国东部（俄亥俄）区域.

Example

以下示例说明如何提供对 Amazon S3 操作所需的 Amazon ECR 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 Logs 终端节点

Amazon ECS 使用 Fargate 启动类型的 任务，这些任务在没有 Internet 网关的情况下使用 VPC，还使用 awslogs 日志驱动程序发送日志信息以CloudWatch Logs要求您创建 com.amazonaws。***region***的 .logs 接口 VPC 终端节点CloudWatch Logs。有关更多信息，请参阅 中的https://docs.amazonaws.cn/AmazonCloudWatch/latest/logs/cloudwatch-logs-and-interface-VPC.html创建网关终端节点Amazon CloudWatch Logs User Guide。

为 Amazon ECR VPC 终端节点创建终端节点策略

VPC 终端节点策略是一种 IAM 资源策略，该策略在您创建或修改终端节点时可附加到该终端节点。如果您在创建终端节点时未附加策略，AWS 会为您附加一个默认策略，该策略允许对服务的完全访问。终端节点策略不会覆盖或替换 IAM 用户策略或服务特定的策略。这是一个单独的策略，用于控制从终端节点中对指定服务进行的访问。终端节点策略必须采用 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 终端节点控制对服务的访问.

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 的终端节点策略示例。Amazon ECR. 此策略允许特定 IAM 角色从 Amazon ECR. 中拉取映像。

```
{
 "Statement": [{
  "Sid": "AllowPull",
  "Principal": {
   "AWS": "arn:aws:iam::1234567890:role/role_name"
  },
  "Action": [
   "ecr:BatchGetImage",
   "ecr:GetDownloadUrlForLayer"
  ],
  "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"
   ],
   "Resource": "*"
  }
 ]
}
```

修改 的 VPC 终端节点策略Amazon ECR

1. 打开 Amazon VPC 控制台 https://console.amazonaws.cn/vpc/。
2. 在导航窗格中，选择终端节点.
3. 如果您没有为 Amazon ECR 创建 VPC 终端节点，请参阅为 创建 VPC 终端节点Amazon ECR (p. 84).
4. 选择要向其中添加策略的 Amazon ECR VPC 终端节点，然后在屏幕下半部分中选择 Policy 选项卡。
5. 选择 Edit Policy（编辑策略），然后对策略进行更改。
6. 选择 Save（保存）以保存策略。

Amazon ECR 监控

您可以使用 Amazon ECR 监控 Amazon CloudWatch API 使用情况，此工具可从 Amazon ECR 收集原始数据，并将数据处理为便于读取的近乎实时的指标。这些统计数据会保存两周，从而使您能够访问历史信息并了解 API 使用情况。Amazon ECR 指标数据以一分钟为间隔自动发送到 CloudWatch。有关 CloudWatch 的更多信息，请参阅 Amazon CloudWatch 用户指南。

Amazon ECR 根据您的 API 使用情况提供了授权、映像推送和映像拉取操作的指标。

监控是保持 Amazon ECR 和您的 AWS 解决方案的可靠性、可用性和性能的重要方面。我们建议您从组成 AWS 解决方案的资源收集监控数据，以便更轻松地调试出现的多点故障。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：

- 您的监控目标是什么？
- 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？

下一步是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，以便将其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

可视化服务配额并设置警报

您可以使用 CloudWatch 控制台可视化服务配额，并查看当前用量与服务配额的比较情况。您还可以设置警报，以便在接近配额时收到通知。

可视化服务配额并选择性地设置警报

1. 通过以下网址打开 CloudWatch 控制台：https://console.amazonaws.cn/cloudwatch/。
2. 在导航窗格中，选择 Metrics.
3. 在 All metrics (所有指标) 选项卡上，选择 Usage (用量)，然后选择 By AWS Resource (按 AWS 资源).

   这将显示服务配额使用指标的列表。
4. 选中其中一个指标旁边的复选框。

   该图表显示您该 AWS 资源的当前使用情况。

5. 要将服务配额添加到图表，请执行以下操作：

   a. 选择 Graphed metrics 选项卡。

   b. 选择 Math expression (数学表达式)、Start with an empty expression (从空表达式开始). 然后在新行中，在 Details (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 Actions （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 Conditions (条件) 下，确保 Threshold type (阈值类型) 为 Static (静态)，并将 Whenever Expression1 is (当 Expression1 为) 设置为 Greater (大于). 在 than (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 Next (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 Next.

   e. 在下一页上，输入警报的名称和描述，然后选择 Next (下一步).

   f. 选择 Create alarm (创建警报).

Amazon ECR 使用情况指标

您可以使用 CloudWatch 用量指标来提供账户资源使用情况的可见性。使用这些指标在 CloudWatch 图表和控制面板上可视化当前服务用量。

Amazon ECR 用量指标与 AWS 服务配额对应。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 服务配额的更多信息，请参阅Amazon ECR 服务配额 (p. 100)。

Amazon ECR 在 `AWS/Usage` 命名空间中发布以下指标。

指标	描述
`CallCount`	来自您的账户的 API 操作调用的数量。资源由与指标关联的维度定义。 此指标最有用的统计数据是 `SUM`，它表示定义期间来自所有贡献者的值的总和。

以下维度用于优化由 发布的用量指标。Amazon ECR.

维度	描述
`Service`	包含该资源的 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`.

维度	描述
Resource	正在运行的资源的类型。目前，Amazon ECR 返回有关以下 API 操作的 API 使用情况的信息。 • GetAuthorization 令牌 • BatchCheckLayerAvailability • InitiateLayerUpload • UploadLayerPart • CompleteLayerUpload • PutImage • BatchGetImage • GetDownloadUrlForLayer
Class	要跟踪资源的级别。目前，Amazon ECR 不使用类维度。

Amazon ECR 使用率报告

AWS 提供了称为 Cost Explorer 的免费报告工具，该工具可让您分析 Amazon ECR 资源的成本和使用率。

使用 Cost Explorer 查看使用率和成本的图表。您可以查看前 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。

成本和使用率报告中的计量数据显示跨所有 Amazon ECR 存储库的使用率。有关更多信息，请参阅 标记资源以便于计费 (p. 29).

有关创建 AWS 成本和使用率报告的更多信息，请参阅 中的 AWS 成本和使用率报告。AWS Billing and Cost Management 用户指南

Amazon ECR 事件和 EventBridge

利用 Amazon EventBridge，您可以自动执行您的 AWS 服务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。AWS 服务中的事件将近乎实时传输到 EventBridge 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操作。可自动触发的操作包括：

- 将事件添加到 中的日志组CloudWatch Logs
- 调用 AWS Lambda 函数
- 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 AWS SMS 队列

有关更多信息，请参阅Amazon EventBridge 中的 Amazon EventBridge 用户指南 入门.

来自 Amazon ECR 的示例事件

以下是来自 的示例事件。Amazon ECR. 尽最大努力发出事件。

已完成映像推送的事件

每个映像推送完成后，将发送以下事件。有关更多信息，请参阅 推送 Docker 映像 (p. 32).

```
{
    "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-repo",
        "image-digest":
 "sha256:7f5b2640fe6fb4f46592dfd3410c4a79dac4f89e4782432e0378abcd1234",
        "action-type": "PUSH",
        "image-tag": "latest"
    }
}
```

已完成映像扫描的事件

每个映像扫描完成后，将发送以下事件。`finding-severity-counts` 参数仅返回严重性级别的值（如果存在）。例如，如果图像不包含 `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 映像扫描 (p. 53).

```
{
    "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-repo"
    ],
    "detail": {
        "scan-status": "COMPLETE",
        "repository-name": "my-repo",
        "finding-severity-counts": {
        "CRITICAL": 10,
        "MEDIUM": 9
      },
        "image-digest":
 "sha256:7f5b2640fe6fb4f46592dfd3410c4a79dac4f89e4782432e0378abcd1234",
        "image-tags": []
    }
}
```

映像删除的事件

删除映像时将发送以下事件。有关更多信息，请参阅 删除映像 (p. 37).

```
{
    "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-repo",
        "image-digest":
 "sha256:7f5b2640fe6fb4f46592dfd3410c4a79dac4f89e4782432e0378abcd1234",
        "action-type": "DELETE",
        "image-tag": "latest"
    }
}
```

使用 Amazon ECR 记录 AWS CloudTrail 操作

Amazon ECR 与 AWS CloudTrail 集成，后者是一项服务，该服务提供由用户、角色或 AWS 中的 Amazon ECR 服务执行的操作的记录。CloudTrail 将以下 Amazon ECR 操作作为事件捕获：

- 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执行的所有操作
- 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

在创建跟踪时，可以使 CloudTrail 事件持续传送到 Amazon S3 存储桶（包括 Amazon ECR. 的事件）。如果您不配置跟踪，则仍可在 CloudTrail 控制台的 Event history (事件历史记录). 中查看最新事件。通过使用此信息，可以确定向 Amazon ECR 发出的请求、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 User Guide。

Amazon ECR 中的 信息CloudTrail

在您创建 CloudTrail 账户时，即针对该账户启用了 AWS Amazon ECR 中发生活动时，该活动将记录在 CloudTrail 事件中，并与其他 AWS 服务事件一同保存在 Event history (事件历史记录). 中。您可以在 AWS 账户中查看、搜索和下载最新事件。有关更多信息，请参阅使用 CloudTrail 事件历史记录查看事件.

要持续记录 AWS 账户中的事件（包括 Amazon ECR 的事件），请创建跟踪。通过跟踪，CloudTrail 可将日志文件传送至 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。此跟踪在 AWS 分区中记录事件，并将日志文件传送至您指定的 Amazon S3 存储桶。此外，您还可以配置其他 AWS 服务，以分析在 CloudTrail 日志中收集的事件数据并采取操作。有关更多信息，请参阅：

- 为您的 AWS 账户创建跟踪
- AWS 服务与 CloudTrail 日志的集成
- 为 配置 Amazon SNS 通知 CloudTrail
- 接收来自多个区域的 CloudTrail 日志文件和从多个账户接收 CloudTrail 日志文件

Amazon ECR 记录所有 CloudTrail API 操作，Amazon Elastic Container Registry API Reference. 中介绍了这些操作。在执行常见任务时，将在 CloudTrail 日志文件中为该任务包含的每个 API 操作生成节。例如，在创建存储库时，将在 `GetAuthorizationToken` 日志文件中生成 `CreateRepository`、`SetRepositoryPolicy` 和 CloudTrail 部分。当您将映像推送到存储库时，将生成 `InitiateLayerUpload`、`UploadLayerPart`、`CompleteLayerUpload` 和 `PutImage` 部分。在拉取映像时，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 (p. 93).

每个事件或日志条目都包含有关生成请求的人员的信息。身份信息帮助您确定以下内容：

- 请求是使用根用户凭证还是 IAM 用户凭证发出的
- 请求是使用角色还是联合身份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 CloudTrail `userIdentity` 元素。

了解 Amazon ECR 日志文件条目

跟踪是一种配置，可用于将事件作为日志文件传送到您指定的 Amazon S3 存储桶。CloudTrail 日志文件包含一个或多个日志条目。一个事件表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数和其他信息的信息。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会以任何特定顺序显示。

CloudTrail 日志条目示例

以下是针对一些常见 CloudTrail 任务的 Amazon ECR 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，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。在实际的 CloudTrail 日志文件中，有来自多个 AWS 服务的条目和事件。

主题

示例：创建存储库操作

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 `CreateRepository` 操作。


```
{
    "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 AWS KMS 存储库时的 Amazon ECR CreateGrant API 操作

以下示例显示了一个 CloudTrail 日志条目，该条目演示在创建启用了 KMS 加密的 AWS KMS 存储库时的 Amazon ECR CreateGrant 操作。对于启用 KMS 加密创建的每个存储库，您应该会在 CloudTrail 中看到两个 CreateGrant 日志条目。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAIEP6W46J43IG7LXAQ",
        "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：映像推送操作

下面的示例显示了一个 CloudTrail 日志条目，该条目说明的是使用 `PutImage` 操作推送映像。

> **Note**
>
> 当推送映像时，您在 `InitiateLayerUpload` 日志中还可以看到 `UploadLayerPart`、`CompleteLayerUpload` 和 CloudTrail 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/
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
\": \"sha256:7ab043301a6187ea3293d80b30ba06c7bf1a0c3cd4c43d10353b31bc0cecfe7d
\"\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}"
 },
 "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
\": \"sha256:7ab043301a6187ea3293d80b30ba06c7bf1a0c3cd4c43d10353b31bc0cecfe7d
\"\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：映像拉取操作

下面的示例显示了一个 CloudTrail 日志条目，该条目说明的是使用 `BatchGetImage` 操作拉取映像。

> **Note**
>
> 当拉取映像时，如果本地尚没有映像，您在 `GetDownloadUrlForLayer` 日志中还将看到 CloudTrail 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：映像生命周期策略操作

以下示例显示了一个 CloudTrail 日志条目，该条目演示映像何时由于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。


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

Amazon ECR 服务配额

下表提供了 的默认服务配额。Amazon Elastic Container Registry (Amazon ECR).

服务配额	描述	默认配额值	可调整
已注册的存储库	可在每个区域中创建的存储库的最大数量。	10000	是
每个存储库的映像数	每个存储库的最大映像数。	10000	是

下表提供映像推送和映像拉取操作所涉及的每个 Amazon ECR API 操作的默认速率配额。

Amazon ECR 操作	API 操作	描述	默认配额值	可调整
身份验证	GetAuthorizationToken 请求的速率	您在每个区域中每秒可以发出的 GetAuthorizationToken API 请求数。	200	是
映像推送	BatchCheckLayerAvailability 请求的速率	您在每个区域中每秒可以发出的 BatchCheckLayerAvailability API 请求数。 将映像推送到存储库时，会检查每个映像层以验证之前是否已上传它。如果已上传，则会跳过映像层。	200	是
	InitiateLayerUpload 请求的速率	您在每个区域中每秒可以发出的 InitiateLayerUpload API 请求数。 推送映像时，对于每个尚未上传的映像层，会调用一次 InitiateLayerUpload API。是否已上传映像层由 BatchCheckLayerAvailability API 操作确定。	10	是
	CompleteLayerUpload 请求的速率	您在每个区域中每秒可以发出的 CompleteLayerUpload API 请求数。 推送映像时，对于每个新的映像	10	是

Amazon ECR 操作	API 操作	描述	默认配额值	可调整
		层，都会调用一次 CompleteLayerUpload API，以验证上传是否已完成。		
	UploadLayerPart 请求的速率	您在每个区域中每秒可以发出的 UploadLayerPart API 请求数。 推送映像时，每个新映像层都会分段上传。每个映像层部分的最大大小可以为 20,971,520 字节（或约 20MB）。对于每个新的映像层部分，都会调用一次 UploadLayerPart API。	260	是
	PutImage 请求的速率	您在每个区域中每秒可以发出的 PutImage API 请求数。 推送映像并上传所有新的映像层后，将调用一次 PutImage API，以创建或更新映像清单以及与该映像关联的标签。	10	是
映像拉取	BatchGetImage 请求的速率	您在每个区域中每秒可以发出的 BatchGetImage API 请求数。 拉取映像时，会调用 BatchGetImage API 一次以检索映像清单。	1000	
	GetDownloadUrlForLayer 请求的速率	您在每个区域中每秒可以发出的 GetDownloadUrlForLayer API 请求数。 拉取映像时，对于每个尚未缓存的映像层调用一次 GetDownloadUrlForLayer API。	1500	是

下表提供了 Amazon ECR 和 Docker 映像的其他配额，这些配额无法更改。

Note

下表中提到的层分段信息仅当您直接调用 Amazon ECR API 操作以便为映像推送操作启动分段上传时才适用。极少执行此动作。我们建议您使用 Docker CLI 来拉取、标记和推送映像。

服务配额	描述	配额值	可调整
层分段	层分段数上限。此配额仅在您使用 Amazon ECR API 操作直接启动映像推送操作的分段上传时才适用。	1000	否
层大小上限	层的最大大小 (MiB)。**	10000	否
层分段大小下限	层分段的最小大小 (MiB)。此配额仅在您使用 Amazon ECR API 操作直接启动映像推送操作的分段上传时才适用。	5	否
层分段大小上限	层分段的最大大小 (MiB)。此配额仅在您使用 Amazon ECR API 操作直接启动映像推送操作的分段上传时才适用。	10	否
每个映像的标签数	每个映像的最大标签数。	1000	否
生命周期策略长度	生命周期策略中的最大字符数。	30,720	否
每个生命周期策略的规则数	生命周期策略中的最大规则数量。	50	否
映像扫描速率	每天每个映像的最大映像扫描次数。	1	否

** 此处所列的层大小上限由层分段大小上限 (10 MiB) 乘以层分段数上限 (1000) 计算得出。

在 Amazon ECR中管理您的 AWS 管理控制台 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 https://docs.amazonaws.cn/servicequotas/latest/userguide/intro.html 中的什么是服务配额?Service Quotas 用户指南。

可使用 Service Quotas 轻松查找所有 Amazon ECR 服务配额的值。

查看 Amazon ECR 服务配额（AWS 管理控制台）

1. 在 https://console.amazonaws.cn/servicequotas/ 上打开 Service Quotas 控制台。
2. 在导航窗格中，选择 AWS services (AWS 服务).
3. 从 AWS services (AWS 服务) 列表中，搜索并选择 Amazon Elastic Container Registry (Amazon ECR).

   在 Service quotas （服务配额） 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 默认配额以及配额值是否可调整。
4. 要查看有关服务配额的其他信息（如描述），请选择配额名称。

要请求增加配额，请参阅 https://docs.amazonaws.cn/servicequotas/latest/userguide/request-increase.html 中的请求增加配额Service Quotas 用户指南。

创建 CloudWatch 警报以监控 API 使用情况指标

Amazon ECR 提供了 CloudWatch 使用情况指标，这些指标与注册表身份验证、映像推送和映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使用情况，并配置警报以便在您的使用情况接近服务配额时提醒您。有关更多信息，请参阅 Amazon ECR 使用情况指标 (p. 89).

使用以下步骤根据其中一个 CloudWatch API 使用情况指标创建 Amazon ECR 警报。

根据您的 Amazon ECR 使用情况配额创建警报 (AWS 管理控制台)

1. 在 https://console.amazonaws.cn/servicequotas/ 上打开 Service Quotas 控制台。
2. 在导航窗格中，选择 AWS services (AWS 服务).
3. 从 AWS services (AWS 服务) 列表中，搜索并选择 Amazon Elastic Container Registry (Amazon ECR).
4. 在 Service quotas （服务配额） 列表中，选择要为其创建警报的 Amazon ECR 使用配额。
5. 在 Amazon CloudWatch Events 警报部分中，选择 Create (创建).
6. 对于 Alarm threshold （警报阈值），选择要设置为警报值的已应用配额值的百分比。
7. 对于 Alarm name (警报名称)，输入警报名称，然后选择 Create (创建).

Amazon ECR 故障排除

本章帮助您查找 Amazon Elastic Container Registry (Amazon ECR) 的诊断信息，并为常见问题和错误消息提供故障排除步骤。

主题

启用 Docker 调试输出

要开始调试任何 Docker 相关问题，都需要首先在您的主机实例上运行的 Docker 守护程序中启用 Docker 调试输出。有关在Amazon ECR容器实例Amazon ECS上使用从 拉取的映像时启用 Docker 调试的更多信息，请参阅 中的启用 Docker Amazon Elastic Container Service Developer Guide调试输出。

启用 AWS CloudTrail

有关 Amazon ECR 所返回错误的其他信息，可以通过启用 AWS CloudTrail 进行查找，它是为 AWS 账户记录 AWS 调用的一项服务。CloudTrail 可将日志文件传输给 Amazon S3 存储桶。通过使用 CloudTrail 收集的信息，您可以确定向 AWS 服务成功发出了哪些请求、发出请求的用户、请求时间等等。要了解有关 CloudTrail 的更多信息（包括如何启用该服务及如何查找日志文件），请参阅 AWS CloudTrail User Guide. 有关将 CloudTrail 与 Amazon ECR 配合使用的更多信息，请参阅使用 Amazon ECR 记录 AWS CloudTrail 操作 (p. 92).

为 优化性能Amazon ECR

以下部分提供了在使用 时可用于优化性能的设置和策略建议。Amazon ECR.

使用 Docker 1.10 及以上版本可利用同时层上传

Docker 映像由层组成，是映像的中间构建阶段。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。

使用较小基本映像

通过 Docker Hub 提供的默认映像，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映像，或使用 Docker 最小映像构建您自己的基本映像。有关更多信息，请参阅 Docker 文档中的创建基本映像。

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 映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 &&\
    rm 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 拉取镜像的延迟。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 |\
     sed -n 's/\(\d*\)[a-zA-Z]*$/\1/p')
```

可以使用 AWS CLI 参数将该区域传递给 --region 命令，或使用 aws configure 命令将该区域设为某个配置文件的默认区域。您还可以在使用 AWS 软件开发工具包进行调用时设置区域。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。

使用 时通过 Docker 命令纠正错误Amazon ECR

主题

有时，针对 Amazon ECR 运行 Docker 命令可能导致错误消息。一些常见错误消息和可能的解决办法解释如下。

从 Amazon ECR 存储库拉取镜像时，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 `Filesystem verification failed` 命令从 docker pull 存储库拉取镜像时，可能会收到错误 Amazon ECR 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拉取的 Docker 映像。为腾出空间存储新映像，可以删除旧映像。使用 docker images 命令可查看所有已下载到本地的 Docker 映像的列表及这些映像的大小。

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 docker pull 实例运行 Amazon EC2，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，调用 Amazon ECR 存储库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 拉取镜像时，出现错误：“Filesystem Layer Verification Failed”(文件系统分层验证失败)Amazon ECR

您可能在使用 `image image-name not found` 命令拉取映像时收到错误 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...
```

此错误表示映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满，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 `HTTP 403 (Forbidden)` 命令成功通过 Docker 身份验证，也可能会从 `no basic auth credentials` 或 docker push 命令收到 docker pull 错误或者错误消息 aws ecr get-login-password 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 美国东部（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。

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，请参阅存储库策略 (p. 23).

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 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，但如果尝试推送或拉取镜像，这些命令会失败。您可以从从 输出的 Docker 登录命令中的 registry 参数中删除 `https://` 方案，以解决此错误aws ecr get-login。 下面显示了不带 HTTPS 方案的示例 Docker 登录命令。

```
docker login -u AWS -p <password> <aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

Amazon ECR 错误消息问题排查

有时，通过 Amazon ECS 控制台或 AWS CLI 或触发的 API 调用会存在错误消息。一些常见错误消息和可能的解决办法解释如下。

运行 aws ecr get-login 时出现错误：“Error Response from Daemon: Invalid Registry Endpoint”(守护程序响应出错：注册表终端节点无效)

运行 aws ecr get-login 命令获取 Amazon ECR 存储库的登录凭证时，您可能看到以下错误：

```
Error response from daemon: invalid registry endpoint
    https://xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/v0/: unable to ping registry
 endpoint
    https://xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/v0/
v2 ping attempt failed with error: Get https://xxxxxxxxxxxx.dkr.ecr.us-
east-1.amazonaws.com/v2/:
    dial tcp: lookup xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com on 172.20.10.1:53:
    read udp 172.20.10.1:53: i/o timeout
```

运行 Docker Toolbox、Docker for Windows 或 Docker for Mac 的 macOS 和 Windows 系统可能会发生此错误。导致此问题的原因通常是：其他应用程序在通过本地网关更改路由器 (192.168.0.1) 时，虚拟机必须通过调用本地网关才能访问 Amazon ECR 服务。如果使用 Docker 工具箱时出现此错误，通常可以通过重启 Docker 系统环境，或重新启动本地客户端的操作系统来解决。如果该方法未能解决问题，可使用 docker-machine ssh 命令登录容器实例。可在外部主机上执行 DNS 查找，以验证其结果是否与本地主机上的结果相同。如果结果不同，请参考 Docker 工具箱的文档，确保 Docker 系统环境已正确配置。

HTTP 429：请求过多或 ThrottleException

您可能会从一个或多个 `429: Too Many Requests` 命令或 API 调用收到 `ThrottleException` 错误或 Amazon ECR 错误。如果您将 Docker 工具与 结合使用Amazon ECR，那么对于 Docker 版本 1.12.0 及更高版本，您可能会看到错误消息 `TOOMANYREQUESTS: Rate exceeded`。 对于 1.12.0 以下的 Docker 版本，您可能会看到错误 `Unknown: Rate exceeded`。

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 中不同的 API 操作有不同的限制。

例如， `GetAuthorizationToken` 操作的限制为每秒 20 个事务 （TPS），最多允许 200 TPS 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。

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 Amazon Web Services 一般参考 中的 AWS 中的错误重试和指数回退。

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 执行操作时，您可能会收到以下错误：Amazon ECR:

```
$ aws ecr get-login
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 的权限的更多信息，请参阅 适用于 Amazon Elastic Container Registry 的 Identity and Access Management (p. 62).

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 (p. 20).

排查映像扫描问题

以下是常见的映像扫描失败。您可以在 Amazon ECR 控制台中通过显示映像详细信息或通过 API 或 AWS CLI 使用 `DescribeImageScanFindings` API 来查看此类错误。

UnsupportedImageError

在尝试扫描使用`UnsupportedImageError`不支持映像扫描的操作系统生成的映像时，您可能会收到 Amazon ECR 错误。 Amazon ECR 支持对主要版本的 Amazon Linux、、、Amazon Linux 2Debian、Ubuntu、CentOS、Oracle Linux、Alpine 和 RHEL Linux 发行版进行包漏洞扫描。一旦分配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 暂存映像生成的映像。

返回 `UNDEFINED` 严重性级别

您可能会收到严重性等级为 的扫描结果`UNDEFINED`。 以下是造成这种情况的常见原因：

- 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。

文档历史记录

下表列出了自 上一次发布以来对文档所做的重要更改。Amazon ECR. 我们还经常更新文档来处理您发送给我们的反馈意见。

更改	描述	日期
跨区域和跨账户复制	Amazon ECR 增加了对为私有注册表配置复制设置的支持。有关更多信息，请参阅私有注册表设置 (p. 15).	2020 年 12 月 8 日
OCI 构件支持	Amazon ECR 增加了对推送和拉取开放容器计划（OCI）构件的支持。向 `artifactMediaType` API 响应`DescribeImages`添加了一个新参数，以指示构件的类型。 有关更多信息，请参阅推送 Helm 图表 (p. 34).	2020 年 8 月 24 日
静态加密	Amazon ECR 添加了对使用具有 AWS Key Management Service（AWS KMS）中存储的客户主密钥（CMKs）的服务器端加密为存储库配置加密的支持。 有关更多信息，请参阅静态加密 (p. 77).	2020 年 7 月 29 日
多架构映像	Amazon ECR 添加了对创建和推送用于多架构映像的 Docker 清单列表的支持。 有关更多信息，请参阅推送多架构映像 (p. 33).	2020 年 4 月 28 日
Amazon ECR 使用情况指标	Amazon ECR 增加了CloudWatch用量指标，这些指标为您的账户的资源用量提供了可见性。您还可以从 CloudWatch 和 CloudWatch 控制台创建Service Quotas警报，以便在您的使用量接近应用的服务配额时获得警报。 有关更多信息，请参阅Amazon ECR 使用情况指标 (p. 89).	2020 年 2 月 28 日
更新了Amazon ECR服务配额	更新了Amazon ECR服务配额以包含每 API 配额。 有关更多信息，请参阅Amazon ECR 服务配额 (p. 100).	2020 年 2 月 19 日
已添加 `get-login-password` 命令	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。 有关更多信息，请参阅使用授权令牌 (p. 13).	2020 年 2 月 4 日
映像扫描	添加了对映像扫描的支持，这有助于识别容器映像中的软件漏洞。 Amazon ECR 使用开源 CoreOS Clair 项目中的常见漏洞和披露（CVEs）数据库，并为您提供扫描结果的列表。 有关更多信息，请参阅映像扫描 (p. 53).	2019 年 10 月 24 日
VPC 终端节点策略	增加了对在 IAM 接口 VPC 终端节点上设置 Amazon ECR 策略的支持。 有关更多信息，请参阅为 Amazon ECR VPC 终端节点创建终端节点策略 (p. 86).	2019 年 9 月 26 日

更改	描述	日期
映像标签可变性	增加了对将存储库配置为不可变的支持，以防止覆盖映像标签。 有关更多信息，请参阅映像标签可变性 (p. 53).	2019 年 7 月 25 日
接口 VPC 终端节点（AWS PrivateLink）	添加了对配置由 AWS 提供支持的接口 VPC 终端节点的支持PrivateLink。 这允许您在 VPC 和 之间创建私有连接 Amazon ECR，而无需通过 Internet、NAT 实例、VPN 连接或 进行访问AWS Direct Connect。 有关更多信息，请参阅Amazon ECR 接口 VPC 终端节点（AWS PrivateLink） (p. 82).	2019 年 1 月 25 日
为资源添加标签	Amazon ECR 增加了对为存储库添加元数据标签的支持。 有关更多信息，请参阅标记 Amazon ECR 存储库 (p. 28).	2018 年 12 月 18 日
Amazon ECR 名称变更	Amazon Elastic Container Registry 已重命名（原来称为 Amazon EC2 Container Registry）。	2017 年 11 月 21 日
生命周期策略	Amazon ECR 生命周期策略使您能够指定存储库中映像的生命周期管理。 有关更多信息，请参阅生命周期策略 (p. 42).	2017 年 10 月 11 日
Amazon ECR 支持 Docker Image Manifest 2、Schema 2	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) 有关更多信息，请参阅容器映像清单格式 (p. 57).	2017 年 1 月 27 日
Amazon ECR 正式发布	Amazon Elastic Container Registry (Amazon ECR) 是一项托管 AWS Docker 注册表服务，它安全、可扩展且可靠。	2015 年 12 月 21 日

AWS 词汇表

有关最新 AWS 术语，请参阅 AWS General Reference 中的 AWS 词汇表。

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。