Amazon ECR

用户指南

API 版本 2015-09-21

Amazon ECR: 用户指南

Amazon Web Services 文档中描述的 Amazon Web Services 服务或功能可能因区域而异。要查看适用于中国区域的差异，请参阅中国的 Amazon Web Services 服务入门。

Table of Contents

什么是 Amazon Elastic Container Registry？

Amazon Elastic Container Registry (Amazon ECR) 是 Amazon 托管容器映像注册表服务，它安全、可扩展且可靠。Amazon ECR 支持私有存储库，其具有使用 Amazon IAM 的基于资源的权限。这样，指定用户或 Amazon EC2 实例可以访问您的容器存储库和映像。您可以使用首选 CLI 推送、提取和管理 Docker 映像、Open Container Itistry (OCI) 映像和 OCI 兼容构件。

> Note
>
> Amazon ECR 也支持公共容器映像存储库。有关更多信息，请参阅 Amazon ECR Public 用户指南中的什么是 Amazon ECR Public。

Amazon 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 Amazon 客户提供直接反馈。有关更多信息，请参阅 Amazon 容器路线图。

Amazon ECR 的组件

Amazon ECR 包括以下组件：

注册表

我们为每个 Amazon 账户均提供了一个 Amazon ECR 私有注册表；您可以在注册表中创建一个或以上存储库，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

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

存储库

Amazon ECR 存储库包含您的 Docker 镜像、Open Container Initiative (OCI) 镜像和 OCI 兼容构件。有关更多信息，请参阅 Amazon ECR 私有存储库 (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 私有存储库策略 (p. 22)。

映像

您可以对存储库推送和提取容器映像。这些映像可以在开发系统中本地使用，也可以在 Amazon ECS 任务定义和 Amazon EKS Pod 规范中使用。有关更多信息，请参阅 将 Amazon ECR 映像与 Amazon ECS 结合使用 (p. 71) 和 将 Amazon ECR 映像与 Amazon EKS 结合使用 (p. 72)。

Amazon ECR 的功能

Amazon ECR 提供以下功能：

- 生命周期策略有助于管理存储库中映像的生命周期。您可以定义导致清理未使用映像的规则。您可以在将规则应用到存储库之前对其进行测试。有关更多信息，请参阅 生命周期策略 (p. 48)。

- 映像扫描有助于识别容器映像中的软件漏洞。每个存储库都可以配置为在推送时扫描。这可确保扫描推送到存储库的每个新映像。然后，您可以检索映像扫描的结果。有关更多信息，请参阅 镜像扫描 (p. 60)。
- 跨区域和跨账户复制使您可以更轻松地将映像放置在需要的位置。它配置为注册表设置，并基于每个区域。有关更多信息，请参阅 私有注册表设置 (p. 14)。
- 缓存提取规则提供了在私有 Amazon ECR 注册表中缓存远程公有注册表中的存储库的方法。使用缓存提取规则时，Amazon ECR 将定期访问远程注册表，以确保 Amazon ECR 私有注册表中缓存的镜像为最新状态。有关更多信息，请参阅 使用缓存提取规则 (p. 37)。

如何开始使用 Amazon ECR

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

Amazon ECR 定价

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

对 Amazon ECR 进行设置

如果您已注册 Amazon 并已在使用 Amazon Elastic Container Service (Amazon ECS) 或 Amazon Elastic Kubernetes Service (Amazon EKS)，则距离使用 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。结合使用 Amazon CLI 与 Amazon ECR 时，我们建议您使用版本 Amazon CLI，它支持最新的 Amazon ECR 功能。如果在 Amazon CLI 中没有看到对 Amazon ECR 功能的支持，可以升级到最新版本。有关更多信息，请参阅 http://aws.amazon.com/cli/。

按照以下任务完成设置，以便首次将容器镜像推送到 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。

注册 Amazon

在注册 Amazon 时，系统将为您的 Amazon 账户自动注册所有服务，包括 Amazon ECR。您只需为使用的服务付费。

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

创建 Amazon 账户

1. 打开 https://portal.aws.amazon.com/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 Amazon 账号，因为在下一个任务中您会用到它。

创建 IAM 用户

Amazon 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 Amazon 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 Amazon 账户的凭证访问 Amazon，而建议您改用 Amazon Identity and Access Management (IAM)。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 IAM 用户的凭证来访问 Amazon。

如果您已注册 Amazon 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. 选择 Root user（根用户）并输入您的 Amazon Web Services 账户 电子邮件地址，以账户拥有者身份登录 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 Users（用户），然后选择 Add users（添加用户）。
3. 对于 User name（用户名），输入 **`Administrator`**。
4. 选中 Amazon Web Services Management Console access (Amazon Web Services Management Console 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，Amazon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 Amazon managed - job function（Amazon 托管 – 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 Amazon 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（刷新）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（创建用户）。

您可使用这一相同的流程创建更多组和用户，并允许您的用户访问 Amazon Web Services 账户 资源。要了解有关使用策略限制用户对特定 Amazon 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 Amazon 控制台注销，然后使用以下 URL，其中 your_aws_account_id 是您的 Amazon 账号,不带连字符 (例如，如果您的 Amazon 账号是 `1234-5678-9012`，则您的 Amazon 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 Amazon 账户 ID，可以创建账户别名。从 IAM 控制面板，选择自定义，然后输入账户别名，例如您的公司名称。有关更多信息，请参阅 IAM 用户指南中的您的 Amazon 账户 ID 及其别名。

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 用户登录链接下进行检查。

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

开始通过 Amazon Web Services Management Console 使用 Amazon ECR

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 对 Amazon ECR 进行设置 (p. 3) 中的步骤。

创建映像存储库

存储库是您存储 Amazon ECR 中 Docker 或 Open Container Initiative (OCI) 映像的地方。当您每次在 Amazon ECR 中推送或提取映像时，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提取映像。

1. 打开位于 https://console.aws.amazon.com/ecr/ 的 Amazon ECR 控制台。
2. 选择开始使用。
3. 对于 Visibility settings（可见性设置），请选择 Private（私密）。
4. 对于 Repository name（存储库名称），请指定存储库的名称。
5. 对于标签不变性，选择存储库的标签可变性设置。配置有不可变标签的存储库会阻止覆盖映像标签。有关更多信息，请参阅 镜像标签可变性 (p. 59)。
6. 对于推送扫描，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。

   Important

   已弃用在存储库级别配置镜像扫描，以支持在注册表级别配置镜像扫描。有关更多信息，请参阅 镜像扫描 (p. 60)。
7. 对于 KMS encryption（KMS 加密）,请选择是否使用 Amazon Key Management Service 服务中存储的 Amazon KMS 密钥进行服务器端加密。有关此功能的更多信息，请参阅静态加密 (p. 96)。
8. 选择创建存储库。

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 Amazon CLI 结合使用 (p. 7)。

1. 选择您创建的存储库，并选择查看推送命令以查看将映像推送到新存储库的步骤。
2. 运行登录命令，此命令通过在终端窗口中使用控制台的命令来针对注册表验证 Docker 客户端的身份。此命令提供一个在 12 小时内有效的授权令牌。
3. (可选) 如果您有要让镜像推送的 Dockerfile，请为新存储库构建并标记镜像。在终端窗口中使用来自控制台的 docker build 命令。确定您与您的 Dockerfile 位于同一目录中。
4. 通过将控制台中的 docker tag 命令粘贴到终端窗口中来为 Amazon ECR 注册表 URI 和新的存储库标记镜像。此控制台命令假设您的镜像是通过上一步中的 Dockerfile 构建得来的。如果您未通过 Dockerfile 构建镜像，请将 *repository*:latest 的第一个实例更换为要推送的本地镜像的镜像 ID 或镜像名称。
5. 通过在终端窗口中使用 docker push 命令来将新标记的镜像推送到存储库。

6. 选择关闭。

将 Amazon ECR 与 Amazon CLI 结合使用

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

有关可用于管理 Amazon 资源的其他工具的更多信息，包括不同的 Amazon 开发工具包、IDE 工具包和 Windows PowerShell 命令行工具，请参阅 http://aws.amazon.com/tools/。

先决条件

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

如果您尚未安装最新 Amazon 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 Amazon CLI

可以使用 Amazon 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 Amazon 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 Amazon 任务的脚本。

要在 Amazon ECR 中使用 Amazon CLI，请安装最新的 Amazon CLI 版本 (Amazon CLI 中从 1.9.15 版本开始提供 Amazon ECR 功能)。可以使用 aws --version 命令查看 Amazon CLI 版本。有关安装 Amazon CLI 或升级到最新版本的信息，请参阅Amazon Command Line Interface用户指南中的安装 Amazon Command Line Interface。

安装 Docker

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

您无需本地开发系统即可使用 Docker。如果您已在使用 Amazon EC2，则可启动 Amazon Linux 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 镜像 (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，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的启动实例。
2. 连接到您的实例。有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的连接到您的 Linux 实例。
3. 更新实例上已安装的程序包和程序包缓存。

   ```
   sudo yum update -y
   ```

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

   ```
   sudo 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 进程守护程序的权限。如果您看到以下错误，请尝试重启您的实例：

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

步骤 1：创建 Docker 镜像

在本节中，您将创建简单 Web 应用程序的 Docker 映像，并在本地系统或 Amazon EC2 实例上测试此映像，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 Amazon ECS 任务定义中使用它。

创建简单 Web 应用程序的 Docker 镜像

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

```
touch Dockerfile
```

2. 编辑您刚刚创建的 Dockerfile 并添加以下内容。

```
FROM public.ecr.aws/docker/library/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 镜像。`RUN` 指令将更新包缓存，安装一些适用于 Web 服务器的程序包，然后将“Hello World!” 内容写入到 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 镜像。

   Note

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

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

4. 运行 docker images 以验证是否已正确创建镜像。

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

   输出：

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

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

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

   Note

   来自 Apache Web 服务器的输出将显示在终端窗口中。您可以忽略“`Could not reliably determine the 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 Amazon EC2 实例，这将是服务器的 Public DNS 值，此值与您用于通过 SSH 连接到实例的地址相同。确保实例的安全组允许端口 80 上的入站流量。
   - 如果您正在本地运行 Docker，可将您的浏览器指向 http://localhost/。
   - 如果您正在 Windows 或 MacOS 计算机上使用 docker-machine，请使用 docker-machine ip 命令查找托管 Docker 的 VirtualBox VM 的 IP 地址，并将 *machine-name* 替换为您正在使用的 Docker 计算机的名称。

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

   您应看到一个显示“Hello World!”语句的 网页。

7. 通过键入 Ctrl + c 来停止 Docker 容器。

步骤 2：向您的默认注册表验证身份

安装并配置 Amazon 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 推送和提取镜像。Amazon CLI 提供 get-login-password 命令来简化身份验证过程。

`get-login-password` 是在使用 Amazon CLI 时向 Amazon ECR 私有注册表进行身份验证的首选方法。确保您已配置 Amazon CLI 与 Amazon 交互。有关更多信息，请参阅 Amazon Command Line Interface 用户指南中的 Amazon CLI 配置基础。

将 Amazon ECR 授权令牌传递给 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

如果您收到错误信息或者 get-login-password 命令不可用，请确保您使用的是最新版本的 Amazon CLI。有关安装 Amazon CLI 或升级到最新版本的信息，请参阅《Amazon Command Line Interface 用户指南》中的安装 Amazon Command Line Interface。

- get-login-password (Amazon CLI)

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

- Get-ECRLoginCommand (Amazon Tools for Windows PowerShell)

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

步骤 3：创建存储库

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

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

步骤 4：推送镜像到 Amazon ECR

现在您可以推送镜像到上一部分中创建的 Amazon ECR 存储库。您使用 docker CLI 推送镜像，但必须满足一些先决条件才能正常工作：

- 安装最低版本的 docker：1.7
- 已使用 docker login 配置 Amazon ECR 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，即可将镜像推送到您在帐户的默认注册表中新创建的存储库中。

标记镜像并推送到 Amazon ECR

1. 列出您存储在本地的镜像，以识别要标记和推送的镜像。

```
docker images
```

输出：

```
REPOSITORY          TAG                 IMAGE ID            CREATED             VIRTUAL
 SIZE
```

```
hello-world         latest              e9ffedc8c286        4 minutes ago       241MB
```

2. 标记镜像并推送到存储库。

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

3. 推送镜像。

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

输出：

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

步骤 5：从 Amazon ECR 提取镜像

在推送镜像到 Amazon ECR 存储库后，可以从其他位置提取该镜像。可使用 docker CLI 提取镜像，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 docker：1.7
- 已使用 docker login 配置 Amazon ECR 授权令牌。
- Amazon ECR 存储库存在且用户有从该存储库提取的权限。

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

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

输出：

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

步骤 6：删除镜像

如果您不再需要一个存储库中的某个镜像，则可以使用 batch-delete-image 命令将其删除。要删除镜像，您必须指定它所在的存储库，并指定镜像的 `imageTag` 或 `imageDigest` 值。以下示例删除 `hello-repository` 存储库中镜像标签为 `latest` 的镜像。

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

步骤 7：删除存储库

如果您不再需要整个存储库中的所有镜像，您可以删除存储库。默认情况下，您不能删除包含镜像的存储库；但是，`--force` 标记允许此操作。要删除包含镜像的存储库 (及其中的所有镜像)，请运行以下命令。

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

Amazon ECR 私有注册表

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

私有注册表概念

- 您的原定设置私有注册表的 URL 为 `https://aws_account_id.dkr.ecr.region.amazonaws.com`。
- 预设情况下，您的账户可以读取和写入私有注册表中的存储库。但是，IAM 用户需拥有调用 Amazon ECR API 的权限才能从私有存储库中推送或提取镜像。Amazon ECR 提供了多个托管策略来控制不同级别的用户访问。有关更多信息，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 90)。
- 您必须为 Docker 客户端授予注册表权限，以便使用 docker push 和 docker pull 命令对该私有注册表中的存储库执行推送和提取镜像操作。有关更多信息，请参阅私有注册表身份验证 (p. 13)。
- 可通过 IAM 用户访问策略及存储库策略对私有存储库加以控制。有关存储库策略的更多信息，请参阅私有存储库策略 (p. 22)。
- 通过配置私有注册表的复制，私有注册表中的存储库可以在自己的私有注册表中的区域之间进行复制，也可以跨单独的账户进行复制。有关更多信息，请参阅 私有镜像复制 (p. 43)。

私有注册表身份验证

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

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

我们提供以下各节详细介绍的注册表身份验证方法。

使用 Amazon ECR 凭证辅助程序

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

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

使用授权令牌

授权令牌的权限范围与用于检索身份验证令牌的 IAM 委托人的权限范围相匹配。身份验证令牌用于访问您的 IAM 委托人有权访问且有效期为 12 小时的任何 Amazon ECR 注册表。要获得授权令牌，您必须使用 GetAuthorizationToken API 操作来检索包含用户名 `AWS` 和编码密码的 base64 编码授权令牌。该 Amazon 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 CLI 为 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 Amazon CLI。有关更多信息，请参阅 Amazon Command Line Interface 用户指南中的安装 Amazon Command Line Interface。

- get-login-password (Amazon CLI)

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

- Get-ECRLoginCommand (Amazon Tools for Windows PowerShell)

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

使用 HTTP API 身份验证

Amazon ECR 支持 Docker 注册表 HTTP API。但是，由于 Amazon ECR 属于私有注册表，因此您必须为每个 HTTP 请求提供授权令牌。您可以通过使用 curl 的 -H 选项来添加 HTTP 授权标头，以传递由 get-authorization-token Amazon 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 Amazon 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. 要向 API 进行身份验证，可将 $TOKEN 变量传递到 curl 命令的 -H 选项。例如，以下命令会列出 Amazon ECR 存储库中的镜像标签。有关更多信息，请参阅 Docker 注册表 HTTP API 参考文档。

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

   您可以在一个 (扩展) 代码行中执行所有这些操作：

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

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

私有注册表设置

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

- 注册表权限 – 您可以使用注册表权限策略向 Amazon 主体授予复制和缓存提取功能的权限。有关更多信息，请参阅私有注册表权限 (p. 15)。
- 缓存提取规则 – 您可以创建缓存提取规则，以缓存 Amazon ECR 私有注册表中的外部公有注册表中的镜像。有关更多信息，请参阅使用缓存提取规则 (p. 37)。
- 复制—您可以为跨区域或跨账户复制配置存储库。有关更多信息，请参阅 私有镜像复制 (p. 43)。
- 扫描配置—默认情况下，为您的注册表启用基本扫描。您可以启用增强扫描功能，该功能提供一种自动持续的扫描模式，可扫描操作系统和编程语言包漏洞。有关更多信息，请参阅镜像扫描 (p. 60)。

私有注册表权限

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

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

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

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

主题

- 设置私有注册表权限声明 (p. 15)
- 删除私有注册表权限声明 (p. 17)
- 私有注册表策略示例 (p. 17)

设置私有注册表权限声明

您可以使用以下步骤添加或更新注册表的权限策略。您可以为每个注册表添加多个策略声明。有关示例策略，请参阅 私有注册表策略示例 (p. 17)。

主题

- 私有注册表复制权限 (p. 15)
- 私有注册表缓存提取权限 (p. 17)

私有注册表复制权限

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

注册表策略必须授予 `ecr:ReplicateImage` API 操作权限。此 API 是一个内部 Amazon ECR API，可在区域或账户之间复制镜像。您还可以授予 `ecr:CreateRepository` 权限，该权限允许 Amazon ECR 在您的

注册表中创建存储库 (如果存储库尚不存在)。如果未提供 `ecr:CreateRepository` 权限，则必须在注册表中手动创建与源存储库名称相同的存储库。如果两者均未完成，复制将失败。任何失败的 CreateRepository 或 ReplicateImage API 操作都会显示在 CloudTrail 中。

要配置复制的权限策略 (Amazon Web Services Management Console)

要为私有注册表配置复制权限策略 (Amazon Web Services Management Console)

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

要配置复制的权限策略 (Amazon CLI)

为私有注册表配置权限策略 (Amazon CLI)

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

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

2. 使用策略文件创建注册表策略。

```
aws ecr put-registry-policy \
      --policy-text file://registry_policy.json \
      --region us-west-2
```

3. 检索要确认的注册表策略。

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

私有注册表缓存提取权限

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

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

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

删除私有注册表权限声明

您可以使用以下步骤删除注册表的所有权限策略声明。

删除私有注册表的权限策略 (Amazon Web Services Management Console)

1. 打开位于 https://console.aws.amazon.com/ecr/ 的 Amazon ECR 控制台。
2. 从导航栏中，选择您在其中配置注册表权限策略的区域。
3. 在导航窗格中，选择注册表。
4. 在注册表页面上，选择您的私有注册表，然后选择权限。
5. 在私有注册表权限页面上，选择删除。
6. 在删除注册表策略确认屏幕上，选择删除策略。

删除私有注册表的权限策略 (Amazon CLI)

1. 删除注册表策略。

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

2. 检索要确认的注册表策略。

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

私有注册表策略示例

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

Note

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

示例：允许源账户的根用户复制所有存储库

```
{
    "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带有前缀 prod- 的所有存储库。

```
{
    "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/prod-*"
            ]
        }
    ]
}
```

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 Amazon ECR 控制台的存储库部分执行相同的操作。Amazon ECR 还与 Docker CLI 集成，因此您可以将镜像从开发环境推送到存储库，并提取。

主题

私有存储库概念

- 默认情况下，您的账户可以读取和写入默认注册表中的存储库 (*aws_account_id*.dkr.ecr.*region*.amazonaws.com)。但是，IAM 用户需拥有权限才能调用 Amazon ECR API 并从您的存储库中推送或提取镜像。Amazon ECR 提供了多个托管策略来控制不同级别的用户访问。有关更多信息，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 90)。
- 可通过 IAM 用户访问策略及个别存储库策略对存储库加以控制。有关更多信息，请参阅 私有存储库策略 (p. 22)。
- 存储库名称可支持命名空间，您可以使用命名空间分组相似的存储库。例如，如果多个团队使用相同的注册表，团队 A 可以使用 `team-a` 命名空间，团队 B 可以使用 `team-b` 命名空间。这样，每个团队都有自己的名为 `web-app` 的镜像，每个镜像都以团队命名空间开头。这种配置可在不产生干扰的情况下同时使用每个团队上的这些镜像。团队 A 的镜像是 `team-a/web-app`，团队 B 的镜像是 `team-b/web-app`。
- 您的镜像可以在自己的注册表中跨区域和跨账户复制到其他存储库。您可以通过在注册表设置中指定复制配置来执行此操作。有关更多信息，请参阅 私有注册表设置 (p. 14)。

创建私有存储库

您的容器镜像存储在 Amazon ECR 存储库中。请按照以下步骤以使用 Amazon Web Services Management Console 创建私有存储库。有关使用 Amazon CLI 创建存储库的步骤，请参阅 步骤 3：创建存储库 (p. 10)。

创建存储库 (Amazon Web Services Management Console)

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

称必须以字母开头，并且只能包含小写字母、数字、连字符、下划线和正斜杠。不支持使用双连字符、下划线或正斜杠。

7. 对于标签不变性，选择存储库的标签可变性设置。配置有不可变标签的存储库会阻止覆盖镜像标签。有关更多信息，请参阅镜像标签可变性 (p. 59)。
8. 对于 Scan on push（推送时扫描），尽管您可以在存储库级别为基础扫描指定扫描设置，但最佳实践是在私有注册表级别指定扫描配置。通过在私有注册表级别指定扫描设置，您可以启用增强扫描或基本扫描，并定义用于指定扫描哪些存储库的筛选条件。有关更多信息，请参阅镜像扫描 (p. 60)。
9. 对于 KMS 加密，选择是否使用 Amazon Key Management Service 启用存储库中的镜像加密。预设情况下，启用 KMS 加密后，Amazon ECR 会使用带有别名 `aws/ecr` 的 Amazon 托管式密钥 (KMS 密钥)。当您首次创建启用 KMS 加密的存储库时，将在您的账户中创建此密钥。有关更多信息，请参阅 静态加密 (p. 96)。
10. 当启用 KMS 加密时，选择客户加密设置 (高级) 以选择自己的 KMS 密钥。该 KMS 密钥必须与集群同在一个区域中。选择创建 Amazon KMS 密钥导航到 Amazon KMS 控制台，从中创建自己的密钥。
11. 选择创建存储库。
12. (可选) 选择创建的存储库，并选择查看推送命令以查看将镜像推送到新存储库的步骤。要详细了解如何将镜像推送到存储库，请参阅推送镜像 (p. 31)。

查看私有存储库详细信息

创建存储库后，您可以在 Amazon Web Services Management Console 中查看与存储库相关的详细信息：

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

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

查看存储库信息 (Amazon Web Services Management Console)

1. 从 https://console.aws.amazon.com/ecr/repositories 打开 Amazon ECR 控制台。
2. 从导航栏中，选择包含要查看的存储库的区域。
3. 在导航窗格中，选择存储库。
4. 在 Repositories（存储库）页面上，选择 Private（私有）选项卡，然后选择要查看的存储库。
5. 在存储库详细信息页面上，控制台默认为 Images（镜像）视图。使用导航菜单查看其他存储库相关信息。
   - 选择 Summary（摘要）查看存储库详细信息并拉取该存储库的计数数据。
   - 选择 Images（镜像）以查看与存储库中的镜像相关的信息。要查看有关镜像的更多信息，请选择镜像标签。有关更多信息，请参阅查看镜像详细信息 (p. 36)。

     如果您要删除的镜像没有标签，您可以选择要删除的存储库左侧的框，然后选择删除。有关更多信息，请参阅 删除镜像 (p. 41)。
   - 选择权限以查看适用于存储库的存储库策略。有关更多信息，请参阅 私有存储库策略 (p. 22)。
   - 选择生命周期策略以查看适用于存储库的生命周期策略规则。此处还可查看生命周期事件历史记录。有关更多信息，请参阅 生命周期策略 (p. 48)。

- 选择标签以查看适用于存储库的元数据标签。

编辑私有存储库

可以编辑现有存储库以更改其镜像标签可变性和镜像扫描设置。

编辑存储库（Amazon Web Services Management Console）

1. 从 https://console.aws.amazon.com/ecr/repositories 打开 Amazon ECR 控制台。
2. 从导航栏中，选择包含要编辑的存储库的区域。
3. 在导航窗格中，选择存储库。
4. 在 Repositories（存储库）页面上，选择 Private（私有）选项卡，然后选择要编辑的存储库，并选择 Edit（编辑）。
5. 对于标签不变性，选择存储库的标签可变性设置。配置有不可变标签的存储库会阻止覆盖镜像标签。有关更多信息，请参阅镜像标签可变性 (p. 59)。
6. 对于 Image scan settings（镜像扫描设置），尽管您可以在存储库级别为基础扫描指定扫描设置，但最佳实践是在私有注册表级别指定扫描配置。通过在私有注册表级别指定扫描设置，您可以启用增强扫描或基本扫描，并定义用于指定扫描哪些存储库的筛选条件。有关更多信息，请参阅镜像扫描 (p. 60)。
7. 对于 Encryption settings（加密设置），此字段仅供查看，因为存储库的加密设置在存储库创建完成之后无法更改。
8. 选择保存以更新存储库设置。

删除私有存储库

如果存储库已使用完毕，您可以删除它。当您在 Amazon Web Services Management Console 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（Amazon Web Services Management Console）

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

   Important

   选定存储库中的所有镜像也会被删除。

私有存储库策略

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，仅创建存储库的 Amazon 账户有权访问存储库。您可以应用策略文档来允许针对您的存储库的其他权限。

主题

- 存储库策略与 IAM 策略 (p. 23)

存储库策略与 IAM 策略

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

在确定某个特定 IAM 用户或角色可对存储库执行的操作时，将同时使用 Amazon ECR 存储库策略和 IAM 策略。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 要求用户有权通过 IAM 策略调用 `ecr:GetAuthorizationToken` API，然后才能对注册表进行身份验证并对任意 Amazon ECR 存储库推送或提取任意镜像。Amazon ECR 提供多个托管 IAM 策略，用于控制不同级别下的用户访问；有关更多信息，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 90)。

您可以使用这两个策略类型之一来控制对您的存储库的访问，如以下示例中所示。

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

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

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 Resource Name (ARN) 格式的更多信息，请参阅 资源 (p. 80)。

```
{
  "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"
```

```
    ]
    }]
}
```

设置私有存储库策略声明

您可以通过以下步骤在 Amazon Web Services Management Console 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 私有存储库策略示例 (p. 25)。

> **Important**
>
> Amazon ECR 要求用户有权通过 IAM 策略调用 `ecr:GetAuthorizationToken` API，然后才能对注册表进行身份验证并从任意 Amazon ECR 存储库推送或提取任意镜像。Amazon ECR 提供多个托管 IAM 策略，用于控制不同级别下的用户访问；有关更多信息，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 90)。

设置存储库策略声明

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

   > **Note**
   >
   > 如果您未看到权限选项，请确保您处于存储库镜像列表视图中。
6. 在编辑权限页面上，选择添加声明。
7. 对于声明名称，输入声明的名称。
8. 对于效果，选择策略语句产生的结果是允许还是明确拒绝。
9. 对于委托人，选择策略声明应用到的范围。有关更多信息，请参阅 IAM 用户指南中的 Amazon JSON 策略元素：委托人。

   - 您可以将该声明应用于所有经过身份验证的 Amazon 用户，方法是选择每个人 (*) 复选框。
   - 对于服务委托人，指定服务委托人名称 (例如 `ecs.amazonaws.com`) 以将声明应用到特定的服务。
   - 对于 Amazon 账户 ID，指定一个 Amazon 账号 (例如，`111122223333`) 以将声明应用到特定 Amazon 账户。可以使用逗号分隔的列表指定多个账户。

     > **Important**
     >
     > 您授予权限的账户必须启用所创建存储库策略的区域，否则将发生错误。
   - 对于 IAM 实体，选择 Amazon 账户下将应用声明的角色或用户。

     > **Note**
     >
     > 对于 Amazon Web Services Management Console 中当前不支持的较复杂的存储库策略，您可以使用 set-repository-policy Amazon CLI 命令应用此策略。
10. 对于操作，从各个 API 操作的列表中选择应当应用策略声明的 Amazon ECR API 操作的范围。
11. 完成后，选择保存设置策略。
12. 对要添加的每个存储库策略重复以上步骤。

删除私有存储库策略声明

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. 从 https://console.aws.amazon.com/ecr/repositories 打开 Amazon ECR 控制台。
2. 从导航栏中，选择包含要从其中删除策略声明的存储库的区域。
3. 在导航窗格中，选择存储库。
4. 在存储库页面上，选择要从其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. 在编辑权限页面上，选择删除。

私有存储库策略示例

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 要求用户有权通过 IAM 策略调用 `ecr:GetAuthorizationToken` API，然后才能对注册表进行身份验证并从任意 Amazon ECR 存储库推送或提取任意镜像。Amazon ECR 提供多个托管 IAM 策略，用于控制不同级别下的用户访问；有关更多信息，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 90)。

示例：允许一个或多个 IAM 用户

以下存储库策略允许一个或多个 IAM 用户向存储库推送和提取镜像。

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

示例：允许其他账户

以下存储库策略允许特定账户推送镜像。

> **Important**
>
> 您授予权限的账户必须启用所创建存储库策略的区域，否则将发生错误。

```
{
```

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

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

Note

对于 Amazon Web Services Management Console 中当前不支持的较复杂的存储库策略，您可以使用 set-repository-policy Amazon CLI 命令应用此策略。

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

示例：拒绝所有

以下存储库策略拒绝所有账户中的所有用户提取镜像。

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "DenyPull",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ]
        }
    ]
}
```

示例：限制对特定 IP 地址的访问权限

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

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

`Condition` 块使用 `NotIpAddress` 条件和 `aws:SourceIp` 条件键（这是 Amazon 范围的条件键）。有关这些条件键的更多信息，请参阅 Amazon 全局条件上下文键。`aws:sourceIp` IPv4 值使用标准 CIDR 表示法。有关更多信息，请参阅 IAM 用户指南中的 IP 地址条件运算符。

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

示例：允许 Amazon 服务

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

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

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

标记私有存储库

为了帮助您管理您的 Amazon ECR 存储库，您可以选择通过 Amazon 资源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 Amazon 资源标签并演示如何创建标签。

有关标签的基本知识

标签是为Amazon资源分配的标记。每个标签都由键 和值组成，这两个参数都由您定义。

标签可让您按各种标准 (例如用途、所有者或环境) 对 Amazon 资源进行分类。这在您拥有许多同类型资源时很有用 - 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，应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 Amazon Web Services Management Console、Amazon CLI 和 Amazon ECR API 处理标签。

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

给您的 资源加标签

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标签选项对现有资源应用标签。

如果您使用的是 Amazon ECR API、Amazon CLI 或 Amazon 开发工具包，则可以使用 `CreateRepository` API 操作上的 `tags` 参数对新存储库应用标签，或使用 `TagResource` API 操作对现有资源应用标签。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，请记得其它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 Amazon 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 用量报告 (p. 109)。

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 Amazon Billing 用户指南中的月度成本分配报告。

> Note
>
> 如果您已启用报告，则可以在 24 小时后查看当月的数据。

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标签来查看标签。

要将标签条件到存储库 (Amazon Web Services Management Console)

1. 打开位于 https://console.aws.amazon.com/ecr/ 的 Amazon ECR 控制台。
2. 从导航栏中，选择要使用的区域。
3. 在导航窗格中，选择存储库。
4. 在存储库页面上，选择要查看的存储库。
5. 在存储库：***repository_name*** 页面上，从导航窗格中选择标签。
6. 在标签页面上，选择添加标签、编辑标签。
7. 在编辑标签页面上，为每个标签指定键和值，然后选择保存。

要从单个资源中删除标签 (Amazon Web Services Management Console)

1. 打开位于 https://console.aws.amazon.com/ecr/ 的 Amazon ECR 控制台。
2. 从导航栏中，选择要使用的区域。
3. 在存储库页面上，选择要查看的存储库。
4. 在存储库：***repository_name*** 页面上，从导航窗格中选择标签。
5. 在标签页面上，选择编辑。
6. 在编辑标签页面上，选择要删除的每个标签对应的删除，然后选择保存。

通过 Amazon 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	Amazon 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 Amazon 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 2：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 3：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 4：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
```

私有镜像

Amazon Elastic Container Registry (Amazon ECR) 在私有存储库中存储 Docker 镜像、Open Container Initiative (OCI) 镜像和 OCI 兼容构件。您可以使用 Docker CLI 或首选客户端从存储库推送和提取镜像。

主题

推送镜像

您可以将 Docker 镜像、清单列表和 Open Container Initiative (OCI) 镜像以及兼容的构件推送到您的私有存储库。以下各页对这些内容进行了详述。

Amazon ECR 还提供了一种方法，通过在您的私有注册表设置中指定复制配置，将镜像复制到其他存储库、您自己注册表中的各个区域和不同账户。有关更多信息，请参阅 私有注册表设置 (p. 14)。

主题

推送镜像所需的 IAM 权限

Amazon ECR 要求用户具有以下权限才能推送镜像。按照授予最小权限的最佳实践，您可以将这些权限范围缩小到特定存储库，也可以授予所有存储库的权限。用户必须通过请求授权令牌，向其推送镜像的每个 Amazon ECR 注册表进行身份验证。Amazon ECR 提供多个托管 IAM policy，在不同级别控制用户访问；有关更多信息，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 90)。

以下 IAM policy 授予推送镜像所需的权限，而不将范围划定到特定存储库。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:CompleteLayerUpload",
                "ecr:GetAuthorizationToken",
                "ecr:UploadLayerPart",
                "ecr:InitiateLayerUpload",
                "ecr:BatchCheckLayerAvailability",
                "ecr:PutImage"
            ],
            "Resource": "*"
        }
    ]
}
```

以下 IAM policy 授予将镜像和作用域推送到特定存储库所需的权限。必须将存储库指定为完整的 Amazon Resource Name (ARN)。

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

推送 Docker 镜像

您可以使用 docker push 命令将容器镜像推送到 Amazon ECR 存储库。Amazon ECR 还支持创建和推送用于多架构镜像的 Docker 清单列表。清单列表中引用的每个镜像都必须已经被推送到您的存储库。有关更多信息，请参阅 推送多架构镜像 (p. 33)。

推送 Docker 镜像到 Amazon ECR 存储库

在推送镜像之前，Amazon ECR 存储库必须存在。有关更多信息，请参阅 the section called “创建存储库” (p. 20)。

1. 向要向其推送镜像的 Amazon ECR 注册表验证 Docker 客户端的身份。必须针对每个注册表获得授权令牌，令牌有效期为 12 小时。有关更多信息，请参阅 私有注册表身份验证 (p. 13)。

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

Important

如果收到错误，请安装或更新到最新版本的 Amazon CLI。有关更多信息，请参阅 Amazon Command Line Interface 用户指南中的安装 Amazon Command Line Interface。

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

2. 如果在要推送的注册表中还没有您的镜像存储库，请创建它。有关更多信息，请参阅 创建私有存储库 (p. 20)。
3. 识别要推送的本地镜像。运行 docker images 命令列出系统中的容器镜像。

```
docker images
```

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

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

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

```
docker tag e9ae3c220b23 aws_account_id.dkr.ecr.region.amazonaws.com/my-repository:tag
```

5. 使用 docker push 命令推送镜像：

```
docker push aws_account_id.dkr.ecr.region.amazonaws.com/my-repository:tag
```

6. (可选) 通过重复 Step 4 (p. 33) 和 Step 5 (p. 33)，向镜像应用任何其他标签并将这些标签推送到 Amazon ECR。

推送多架构镜像

Amazon ECR 支持创建和推送用于多架构镜像的 Docker 清单列表。清单列表是通过指定一个或多个镜像名称创建的镜像列表。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镜像创建的。清单列表不是必需项。有关更多信息，请参阅 Docker 清单。

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 ECR 镜像一样在 Amazon ECS 任务定义或 Amazon EKS Pod 规范中提取或引用。

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。您必须已将镜像推送到您的存储库，才能在 Docker 清单中引用。有关如何推送镜像的信息，请参阅 推送 Docker 镜像 (p. 32)。

将多架构 Docker 镜像推送到 Amazon ECR 存储库

在推送镜像之前，Amazon ECR 存储库必须存在。有关更多信息，请参阅 the section called “创建存储库” (p. 20)。

1. 向要向其推送镜像的 Amazon ECR 注册表验证 Docker 客户端的身份。必须针对每个注册表获得授权令牌，令牌有效期为 12 小时。有关更多信息，请参阅 私有注册表身份验证 (p. 13)。

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

Important

如果收到错误，请安装或更新到最新版本的 Amazon CLI。有关更多信息，请参阅 Amazon Command Line Interface 用户指南中的安装 Amazon Command Line Interface。

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

2. 列出存储库中的镜像，确认镜像标签。

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

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

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

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

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

5. 将 Docker 清单列表推送到您的 Amazon ECR 存储库。

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

推送 Helm Chart

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 Chart 推送到 Amazon ECR。

有关在 Amazon EKS 中使用 Amazon ECR 托管的 Helm Chart 的更多信息，请参阅 使用 Amazon EKS 安装托管在 Amazon ECR 上的 Helm Chart (p. 72)。

将 Helm Chart 推送到 Amazon ECR 存储库

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

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

   b. 在 templates 文件夹中创建 ConfigMap。

```
cd helm-test-chart/templates
cat <<EOF > configmap.yaml
apiVersion: v1
```

```
kind: ConfigMap
metadata:
  name: helm-test-chart-configmap
data:
  myvalue: "Hello World"
EOF
```

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

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

输出

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

4. 创建存储库以存储 Helm Chart。存储库的名称应与步骤 2 中创建 Helm Chart 时使用的名称匹配。有关更多信息，请参阅 创建私有存储库 (p. 20)。

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

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

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

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

7. 描述您的 Helm Chart。

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

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

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

```
}
```

8. （可选）有关其他步骤，请安装 Helm configmap 并开始使用 Amazon EKS。有关更多信息，请参阅 使用 Amazon EKS 安装托管在 Amazon ECR 上的 Helm Chart (p. 72)。

查看镜像详细信息

将镜像推送到存储库后，可以在 Amazon Web Services Management Console 中查看其信息。所包括的详细信息如下：

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

查看镜像详细信息 (Amazon Web Services Management Console)

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

提取镜像

如果希望运行 Amazon ECR 中可用的 Docker 镜像，可以使用 docker pull 命令将其提取到本地环境。可以从原定设置注册表或与其他 Amazon 账户关联的注册表执行此操作。要在 Amazon ECS 任务定义中使用 Amazon ECR 镜像，请参阅 。将 Amazon ECR 映像与 Amazon ECS 结合使用 (p. 71)

> **Important**
>
> Amazon ECR 要求用户有权通过 IAM policy 调用 `ecr:GetAuthorizationToken` API，然后才能对注册表进行身份验证并从任意 Amazon ECR 存储库推送或提取任意镜像。Amazon ECR 提供多个托管 IAM policy，在不同级别控制用户访问；有关更多信息，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 90)。

从 Amazon ECR 存储库提取 Docker 镜像

1. 将您的 Docker 客户端验证到要从中提取镜像的 Amazon ECR 注册表。必须针对每个注册表获得授权令牌，令牌有效期为 12 小时。有关更多信息，请参阅 私有注册表身份验证 (p. 13)。
2. (可选) 识别要提取的镜像。
   - 可以使用 aws ecr describe-repositories 命令列出注册表中的存储库：

     ```
     aws ecr describe-repositories
     ```

     上述示例注册表包含一个名为 `amazonlinux` 的存储库。
   - 可以使用 aws ecr describe-images 命令描述存储库中的镜像：

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

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

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

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

Important

如果您收到 *`repository-url`* `not found: does not exist or no pull access` 错误，您可能需要向 Amazon ECR 验证您的 Docker 客户端。有关更多信息，请参阅 私有注册表身份验证 (p. 13)。

使用缓存提取规则

Amazon ECR 支持私有 Amazon ECR 注册表中的远程公有注册表中的缓存存储库。Amazon ECR 目前支持为 Amazon ECR Public 和 Quay 创建缓存提取规则。为外部公有注册表创建了缓存提取后，只需使用 Amazon ECR 私有注册表 URI 从该外部公有注册表中提取镜像，然后 Amazon ECR 将会创建存储库并缓存该镜像。当使用 Amazon ECR 私有注册表 URI 提取缓存镜像时，Amazon ECR 会检查远程注册表以查看是否有新版本的镜像，并且会最多每 24 小时更新一次您的私有注册表。

使用缓存提取的注意事项

使用 Amazon ECR 缓存提取时，应考虑以下内容。

- 以下区域不支持创建缓存提取规则：
  - 中国（北京）(`cn-north-1`)
  - 中国（宁夏）(`cn-northwest-1`)
  - AmazonGovCloud (US-East)`us-gov-east-1`
  - AmazonGovCloud (US-West)`us-gov-west-1`
- 如果使用提取缓存提取镜像，首次提取镜像时不支持 Amazon ECR FIPS 服务终端节点。但是，使用 Amazon ECR FIPS 服务终端节点可以处理后续提取。
- 您最多可以为私有注册表创建 10 个缓存提取规则。
- 当缓存的镜像通过 Amazon ECR 私有注册表 URI 提取时，镜像提取将由 Amazon IP 地址启动。这可以确保镜像提取不被计入公有注册表所具有的任何提取率配额。
- 当缓存的镜像通过 Amazon ECR 私有注册表 URI 提取时，Amazon ECR 最多每 24 小时检查一次远程存储库，以验证缓存的镜像是否为最新版本。此计时器基于缓存镜像的最后一次提取。
- 当使用缓存提取规则提取多架构镜像时，清单列表和清单列表中引用的每个镜像都会被提取到 Amazon ECR 存储库中。如果您只想提取特定架构，则可以使用与架构关联的镜像摘要或标签，而不是与清单列表关联的标签来提取镜像。
- Amazon ECR 使用服务相关的 IAM 角色，该角色为 Amazon ECR 提供了为您创建存储库和推送缓存镜像所需的权限。服务相关 IAM 角色在创建缓存提取规则时自动创建。有关更多信息，请参阅 用于缓存提取的 Amazon ECR 服务相关角色 (p. 88)。
- 默认情况下，提取缓存镜像的 IAM 用户、组或角色具有通过其 IAM policy 授予他们的权限。您可以使用 Amazon ECR 私有注册表权限策略进一步限定 IAM 实体的权限范围。有关更多信息，请参阅 使用注册表权限 (p. 38)。
- 使用缓存提取工作流创建的 Amazon ECR 存储库与任何其他 Amazon ECR 存储库受到同等对待。支持所有存储库功能，例如复制和镜像扫描。

- 当使用缓存提取规则创建新存储库时，标签不变性默认被禁用。如果您手动启用存储库标签不变性，Amazon ECR 可能无法更新缓存的镜像。
- 当使用缓存提取规则创建新存储库时，Amazon KMS 加密默认被禁用。如果您想使用 Amazon KMS 加密，您可以在首次提取镜像之前手动创建存储库。
- 首次使用缓存提取规则提取镜像时，如果您已将 Amazon ECR 配置为通过 Amazon PrivateLink 使用接口 VPC 终端节点，您需要在同一个 VPC 中创建一个带有 NAT 网关的公有子网，然后将从私有子网路由到互联网的所有出站流量引至 NAT 网关，才能使提取操作正常工作。后续的镜像提取不需要此操作。有关更多信息，请参阅 Amazon Virtual Private Cloud 用户指南中的场景：从私有子网访问互联网。

所需的 IAM 权限

除了对私有注册表进行身份验证和推送及拉取镜像所需的 Amazon ECR API 权限之外，还需要以下其他权限才能使用拉取缓存规则。

- `ecr:CreatePullThroughCacheRule` – 授予创建拉取缓存规则的权限。此权限必须通过基于身份的 IAM policy 授予。
- `ecr:BatchImportUpstreamImage` – 授权检索外部镜像并将其导入到您的私有注册表。可以通过使用私有注册表权限策略、基于身份的 IAM policy 或通过使用基于资源的存储库权限策略授予此权限。有关使用存储库权限的更多信息，请参阅 私有存储库策略 (p. 22)。
- `ecr:CreateRepository` – 授予在私有注册表中创建存储库的权限。如果存储缓存图像的存储库不存在，则需要此权限。可以通过基于身份的 IAM policy 或私有注册表权限策略授予此权限。

使用注册表权限

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

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

1. 打开位于 https://console.aws.amazon.com/ecr/ 的 Amazon ECR 控制台。
2. 从导航栏中，选择您在其中配置私有注册表权限语句的区域。
3. 在导航窗格中，选择 Private registry（私有注册表）、Registry permissions（注册表权限）。
4. 在 Registry permissions（注册表权限）页面上，选择 Generate statement（生成语句）。
5. 对于要创建的每个缓存提取权限策略语句，请执行以下操作。

    a. 对于 Policy type（策略类型），请选择 Pull through cache policy（推送缓存策略）。
    b. 对于 Statement id（语句 ID），为推送缓存语句策略提供名称。
    c. 对于 IAM entities（IAM 实体），指定要包含在策略中的 IAM 用户、组或角色。
    d. 对于 Repository namespace（存储库命名空间），选择要与策略关联的推送缓存规则。
    e. 对于 Repository names（存储库名称），指定要应用规则的存储库基本名称。例如，如果您想在 Amazon ECR Public 上指定 Amazon Linux 存储库，存储库名称将为 `amazonlinux`。

要创建私有注册表的权限策略 (Amazon CLI)

使用以下命令 Amazon CLI 命令来通过 Amazon CLI 指定私有注册表权限。

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

```
{
```

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

Important

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

2. 使用 put-registry-policy 命令设置注册表策略。

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

创建缓存提取规则

您可以为包含您要在 Amazon ECR 私有注册表中缓存的镜像的每个外部公有注册表创建一个缓存提取规则。

要创建缓存提取规则 (Amazon Web Services Management Console)

要创建缓存提取规则 (Amazon Web Services Management Console)

1. 打开位于 https://console.aws.amazon.com/ecr/ 的 Amazon ECR 控制台。
2. 从导航栏中，选择要配置私有注册表设置的区域。
3. 在导航窗格中，选择 Private registry（私有注册表）、Pull through cache（缓存提取）。
4. 在 Pull through cache configuration（缓存提取配置）页面上，选择 Add rule（添加规则）。
5. 在 Create pull through cache rule（创建缓存提取规则）页面上，执行以下操作。
   a. 对于 Public registry（公有注册表），选择其中一个预配置的公有注册表。
   b. 对于 Amazon ECR repository namespace（Amazon ECR 存储库命名空间），指定缓存从源公有注册表中提取镜像时要使用的存储库命名空间。默认情况下，已填充命名空间，但也可以指定自定义命名空间。
   c. 选择 Save（保存）以将缓存提取规则保存到注册表设置中。
6. 对要创建的每个缓存提取重复上一步骤。单独为每个区域创建了缓存提取规则。

要创建缓存提取规则 (Amazon CLI)

使用以下 Amazon CLI 命令通过 Amazon CLI 为私有注册表创建缓存提取规则。

- create-pull-through-cache-rule (Amazon CLI)

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

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

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

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

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

使用缓存提取镜像

为外部公有注册表创建缓存提取规则后，只需使用 Amazon ECR 存储库 URI 提取远程镜像，然后镜像将在本地缓存。下面是支持的公有注册表的格式。如果您在使用拉取缓存规则拉取上游镜像时收到错误，请参阅使用拉取缓存规则进行拉取时出错 (p. 126)，以查看最常见的错误以及如何解决这些错误。

Note

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

Amazon ECR Public

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

Quay

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

删除缓存提取规则

您可以删除缓存提取规则以停止缓存行为。删除缓存提取规则不会对缓存的存储库或镜像产生任何影响，它只会阻止将来的缓存行为。

要删除缓存提取规则 (Amazon Web Services Management Console)

要删除缓存提取规则 (Amazon Web Services Management Console)

1. 打开位于 https://console.aws.amazon.com/ecr/ 的 Amazon ECR 控制台。
2. 从导航栏中，选择要配置私有注册表设置的区域。
3. 在导航窗格中，选择 Private registry（私有注册表）、Pull through cache（缓存提取）。
4. 在 Pull through cache configuration（缓存提取配置）页面上，选择要删除的缓存提取规则，然后选择 Delete rule（删除规则）。
5. 在导航窗格中，选择 Private registry（私有注册表）、Registry permissions（注册表权限）。
6. （可选）在 Registry permissions（注册表权限）页面中，查看现有的注册表权限策略语句。您可以删除任何与已删除的缓存提取规则的存储库命名空间关联的注册表权限策略语句。

要删除缓存提取规则 (Amazon CLI)

使用以下 Amazon CLI 命令通过 Amazon CLI 删除缓存提取规则。

- delete-pull-through-cache-rule (Amazon CLI)

  以下示例删除了使用 ecr-public 存储库前缀的缓存提取规则。

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

删除镜像

如果您结束使用镜像，可以从存储库中删除它。如果您结束使用存储库，可以删除整个存储库以及其中的所有镜像。有关更多信息，请参阅 删除私有存储库 (p. 22)。

作为手动删除镜像的替代方法，您可以创建存储库生命周期策略，以便更好地控制存储库中镜像的生命周期管理。生命周期策略自动执行此过程。有关更多信息，请参阅 生命周期策略 (p. 48)。

删除镜像 (Amazon Web Services Management Console)

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

删除镜像 (Amazon CLI)

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

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

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

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

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

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

   要删除多个镜像，您可以在请求中指定多个镜像标签或镜像摘要。

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

重新为镜像添加标签

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

重新为镜像添加标签 (Amazon CLI)

使用 Amazon CLI 重新为镜像添加标签

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

```
MANIFEST=$(aws ecr batch-get-image --repository-name amazonlinux --
image-ids imageTag=latest --output json | jq --raw-output --join-output
 '.images[0].imageManifest')
```

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

   Note

   如果 `--image-tag` 选项在您的 Amazon CLI 版本中不可用，请升级到最新版本。有关更多信息，请参阅 Amazon Command Line Interface 用户指南中的安装 Amazon Command Line Interface。

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

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

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

   您可以在一个 (扩展) 代码行中执行所有这些操作：

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

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

使用 Amazon Tools for Windows PowerShell 重新为镜像添加标签

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

   Note

   如果您的系统上没有可用的 Get-ECRImageBatch cmdlet，请参阅 Amazon Tools for Windows PowerShell 用户指南中的设置 Amazon Tools for Windows PowerShell。

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

2. 将该镜像的清单写入到 *$Manifest* 环境变量。

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

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

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

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

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

   您可以在一个 (扩展) 代码行中执行所有这些操作：

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

私有镜像复制

您可以配置 Amazon ECR 私有注册表以支持存储库的复制。Amazon ECR 同时适用于跨区域和跨账户复制。要进行跨账户复制，目标账户必须配置注册表权限策略，以允许从源注册表进行复制。有关更多信息，请参阅 私有注册表权限 (p. 15)。

主题

私有镜像复制的注意事项

使用私有镜像复制时应注意以下事项。

- 只有在配置复制之后，推送到存储库的内容才会被复制。存储库中任何先前存在的内容都不会复制。为存储库配置复制后，Amazon ECR 将保持目标和源同步。

- 首次配置私有注册表以进行复制时，Amazon ECR 会代表您创建服务相关 IAM 角色。服务相关 IAM 角色授予 Amazon ECR 复制服务在注册表中创建存储库和复制镜像所需的权限。有关更多信息，请参阅 对 Amazon ECR 使用服务相关角色 (p. 86)。
- 要进行跨账户复制，私有注册表目标必须授予允许源注册表复制其镜像的权限。通过设置私有注册表权限策略来完成此授权。有关更多信息，请参阅 私有注册表权限 (p. 15)。
- 如果更改私有注册表的权限策略以删除权限，则以前授予权限的任何进行中复制都可能完成。
- 必须先为某个账户启用区域，然后才能在该区域内执行任何复制操作。有关更多信息，请参阅 Amazon Web Services 一般参考中的管理 Amazon 区域。
- 不支持在 Amazon 分区之间进行跨区域复制。例如，`us-west-2` 中的存储库无法复制到 `cn-north-1`。有关 Amazon 分区的更多信息，请参阅 Amazon 一般参考中的 ARN 格式。
- 私有注册表的复制配置最多可以包含 25 个跨所有规则的唯一目标，最多共有 10 个规则。每个规则最多可包含 100 个筛选条件。这允许为包含用于生产和测试的镜像的存储库指定单独的规则。
- 复制配置支持通过指定存储库前缀来筛选私有注册表中复制的存储库。有关示例，请参阅 示例：使用存储库筛选条件配置跨区域复制 (p. 46)。
- 每次镜像推送只会执行一次复制操作。例如，如果您配置了从 `us-west-2` 到 `us-east-1` 以及从 `us-east-1` 到 `us-east-2` 的跨区域复制，则推送到 `us-west-2` 的镜像仅复制到 `us-east-1`，它不会再复制到 `us-east-2`。此行为同时适用于跨区域和跨账户复制。
- 大多数映像会在不到 30 分钟的时间内复制，但在极少数情况下，复制可能需要更长的时间。
- 注册表复制不执行任何删除操作。复制镜像和存储库不再使用时，可以手动删除它们。
- 存储库策略 (包括 IAM policy) 和生命周期策略不会被复制，而且除了对为其定义的存储库之外，不会产生任何影响。
- 不会复制存储库设置。预设情况下，在由于复制操作而创建的所有存储库上，标签不变性、镜像扫描和 KMS 加密设置都处于禁用状态。可以在创建存储库后更改标签不变性和镜像扫描设置。但是，该设置仅适用于设置更改后推送的镜像。
- 如果在存储库上启用了标签不变性，并且复制了与现有镜像使用相同标签的镜像，则该镜像将被复制，但不包含重复的标签。这可能会形成未标记的镜像。

配置私有镜像复制

分别对每个区域配置复制设置。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 (Amazon Web Services Management Console)

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

   Important

   要进行跨账户复制，目标账户必须配置注册表权限策略，以允许执行复制。有关更多信息，请参阅 私有注册表权限 (p. 15)。
9. (可选) 在添加筛选条件页面上，为复制规则指定一个或多个筛选条件，然后选择添加。对要与复制操作相关联的每个筛选条件重复此步骤。筛选条件指定为存储库名称前缀。如果未指定筛选条件，则会复制所有镜像。添加所有筛选条件后，选择下一步。

10. 在存储库的审核和提交页面上，查看复制规则配置，然后选择提交规则。

配置注册表复制设置 (Amazon CLI)

1. 创建包含要为注册表定义的复制规则的 JSON 文件。复制配置最多可以包含 10 个规则，所有规则最多包含 25 个唯一目标，每个规则最多包含 100 个筛选条件。要在自己的账户中配置跨区域复制，请指定自己的账户 ID。有关更多示例，请参阅 私有镜像复制示例 (p. 46)：

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

2. 创建注册表的复制配置。

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

3. 确认您的注册表设置。

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

查看复制状态

单个容器镜像的复制状态可以通过使用镜像标签或镜像摘要进行查询来查看。

检查复制状态 (Amazon Web Services Management Console)

1. 从 https://console.aws.amazon.com/ecr/repositories 打开 Amazon ECR 控制台。
2. 从导航栏中，选择作为复制注册表来源的区域。
3. 在导航窗格中，选择存储库。
4. 在存储库页面上，选择要检查其复制状态的存储库。
5. 在存储库详细信息页面上，选择镜像标签以检查复制状态。
6. 对于镜像复制状态，验证复制状态。您可以根据镜像标签或镜像摘要查看复制状态。

检查复制状态 (Amazon CLI)

- 可以使用以下命令根据镜像标签查看存储库内容的复制状态。

```
aws ecr describe-image-replication-status \
     --repository-name repository_name \
     --image-id imageTag=image_tag \
     --region us-west-2
```

- 可以使用以下命令根据镜像摘要查看存储库内容的复制状态。

```
aws ecr describe-image-replication-status \
     --repository-name repository_name \
     --image-id imageDigest=image_digest \
     --region us-west-2
```

私有镜像复制示例

以下示例说明如何使用私有镜像复制。

示例：配置跨区域复制到单个目标区域

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

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

示例：使用存储库筛选条件配置跨区域复制

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

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

示例：配置跨区域复制到多个目标区域

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

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

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

示例：配置跨账户复制

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

> **Important**
>
> 要进行跨账户复制，目标账户必须配置注册表权限策略，以允许进行复制。有关更多信息，请参阅 私有注册表权限 (p. 15)。

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

示例：在配置中指定多个规则

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

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

```
    "filterType": "PREFIX_MATCH"
   }]
  }
 ]
}
```

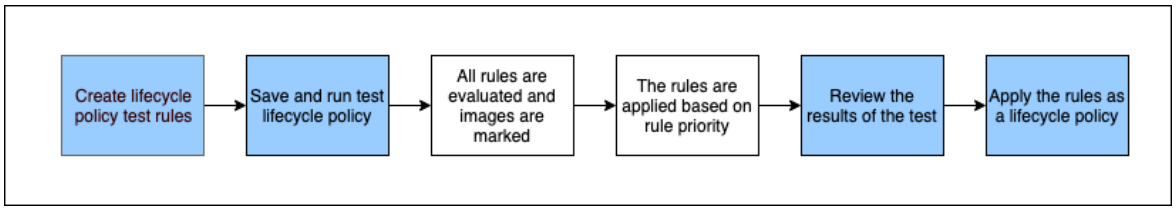
生命周期策略

Amazon ECR 生命周期策略提供了对私有存储库中镜像的生命周期管理的更多控制。生命周期策略是一组规则，其中的每个规则为 Amazon ECR 定义一个操作。这提供了一种自动清理容器镜像的方法，例如根据使用期限或计数过期的镜像。创建生命周期策略后，受影响的镜像会在 24 小时内过期。当 Amazon ECR 基于生命周期策略执行操作时，这将在 Amazon CloudTrail 中记录为一个事件。有关更多信息，请参阅使用 Amazon CloudTrail 记录 Amazon ECR 操作 (p. 113)。

生命周期策略工作原理

生命周期策略由一条或多条规则组成，这些规则确定存储库中的哪些镜像应过期。在考虑使用生命周期策略时，务必使用生命周期策略预览来确认生命周期策略视为过期的镜像，然后再将其应用到存储库。将生命周期策略应用到存储库后，您应发现受影响的镜像将在 24 小时内过期。当 Amazon ECR 基于生命周期策略执行操作时，这将在 Amazon CloudTrail 中记录为一个事件。有关更多信息，请参阅使用 Amazon CloudTrail 记录 Amazon ECR 操作 (p. 113)。

下图显示了生命周期策略工作流程。



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

生命周期策略评估规则

生命周期策略评估器负责解析生命周期策略的明文 JSON，评估所有规则，然后根据规则优先级将这些规则应用于存储库中的镜像。下文更详细地解释了生命周期策略评估器采用的逻辑。有关示例，请参阅生命周期策略的示例 (p. 53)。

- 无论规则优先级如何，评估器都会同时评估所有规则。评估所有规则后，将根据规则优先级应用这些规则。
- 镜像由一条或零条规则设为过期。
- 与规则的标记要求匹配的镜像不能被优先级较低的规则设为过期。
- 规则永远不能标记已由较高优先级规则标记的镜像，但仍然可以将其识别为未过期。

- 规则集必须包含一组唯一的标签前缀。
- 只允许一个规则选择未标记的镜像。
- 如果清单列表引用了映像，则在未先删除清单列表的情况下，该映像无法过期。
- 过期始终按 pushed_at_time 排序，并且始终是较早的镜像在较新的镜像之前过期。
- 使用 tagPrefixList 时，如果 tagPrefixList 值中的全部标签均与任何镜像的标签相符，则该镜像成功匹配。
- 使用 countType = imageCountMoreThan，镜像基于 pushed_at_time 从最新到早排序，然后所有大于指定计数的镜像都将过期。
- 使用 countType = sinceImagePushed，其 pushed_at_time 早于指定天数 (基于 countNumber) 的所有镜像均会过期。

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 生命周期策略的示例 (p. 53)。

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

Note

仅当 tagStatus 为 tagged 时，才能使用 tagPrefixList 参数。仅当 countType 为 sinceImagePushed 时，才能使用 countUnit 参数。

生命周期策略参数

生命周期策略分为以下几个部分：

主题

- 操作 (p. 51)

规则优先级

`rulePriority`

类型：整数

必需：是

设置应用规则的顺序，从低到高。优先级为 1 的生命周期策略规则将首先应用，优先级为 2 的规则将下一个应用，依此类推。当您向某个生命周期策略添加规则时，必须为每个规则赋予一个唯一的 `rulePriority` 值。但是，在策略中的各规则之间，值不需要顺序。具有 `tagStatus` 值 `any` 的规则必须具有最大的 `rulePriority` 值并且最后被评估。

描述

`description`

类型：字符串

必需：否

(可选) 描述生命周期策略中规则的用途。

标签状态

`tagStatus`

类型：字符串

必需：是

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

标签前缀列表

`tagPrefixList`

类型：list[string]

必需：是，仅当 `tagStatus` 设置为标记

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

计数类型

`countType`

类型：字符串

必需：是

指定要应用于镜像的计数类型。

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

计数单位

countUnit

类型：字符串

必需：是，仅当 countType 设置为 sinceImagePushed 时

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

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

计数

countNumber

类型：整数

必需：是

指定计数数量。可接受的值为正整数 (0 不是可接受的值)。

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

操作

type

类型：字符串

必需：是

指定操作类型。支持的值为 expire。

创建生命周期策略预览

生命周期策略预览提供了在应用生命周期策略之前查看其对镜像存储库影响的方法。在将生命周期策略应用到存储库之前进行预览被认为是一种最佳实践。下面的过程演示如何创建生命周期策略预览。

创建生命周期策略预览 (Amazon Web Services Management Console)

1. 从 https://console.aws.amazon.com/ecr/repositories 打开 Amazon ECR 控制台。
2. 从导航栏中，选择包含要对其执行生命周期策略预览的存储库的区域。

3. 在导航窗格中，选择存储库。
4. 在 Repositories（存储库）页面的 Private（私有）选项卡中，选择一个存储库以查看存储库镜像列表。
5. 在存储库镜像列表视图的左侧导航窗格中选择 Lifecycle Policy（生命周期策略）。

   Note

   如果您未看到生命周期策略选项，请确保您处于存储库镜像列表视图中。
6. 在存储库生命周期策略页面上，选择 Edit test rules（编辑测试规则）、Create rule（创建规则）。
7. 输入每个测试生命周期策略规则的以下详细信息。
   a. 对于规则优先级，输入该规则优先级的编号。
   b. 对于规则描述，输入该生命周期策略规则的说明。
   c. 对于镜像状态，选择已标记、未标记或者任何。
   d. 如果您为镜像状态指定 `Tagged`，然后设置标签前缀，则可以有选择地指定根据其执行生命周期策略操作的镜像标签列表。如果指定 `Untagged`，此字段必须为空。
   e. 对于匹配条件，选择自推送镜像以来或者镜像计数超过 (如果适用) 的值。
   f. 选择保存。
8. 重复第 5-7 步以创建其他测试生命周期策略规则。
9. 要运行生命周期策略预览，请选择保存并运行测试。
10. 在测试生命周期规则的镜像匹配下方，查看生命周期策略预览的影响。
11. 如果对预览结果满意，请选择应用为生命周期策略以创建具有指定规则的生命周期策略。应用生命周期策略后，受影响的镜像会在 24 小时内过期。
12. 如果您对预览结果不满意，可以删除一个或多个测试生命周期规则，创建一个或多个规则来替换它们，然后再重复测试。如果不将测试生命周期规则作为生命周期策略应用，则测试规则将保留在控制台中。

创建生命周期策略

生命周期策略允许您创建一组规则，这些规则会让未使用的存储库镜像过期。以下程序显示如何创建生命周期策略。创建生命周期策略后，受影响的镜像会在 24 小时内过期。

Important

最佳实践是创建生命周期策略预览，以确保受生命周期策略规则影响的镜像符合预期。有关更多信息，请参阅创建生命周期策略预览 (p. 51)。

创建生命周期策略 (Amazon Web Services Management Console)

使用控制台创建生命周期策略

1. 从 https://console.aws.amazon.com/ecr/repositories 打开 Amazon ECR 控制台。
2. 从导航栏中，选择包含要对其创建生命周期策略的存储库的区域。
3. 在导航窗格中，选择存储库。
4. 在 Repositories（存储库）页面的 Private（私有）选项卡中，选择一个存储库以查看存储库镜像列表。
5. 在存储库镜像列表视图的左侧导航窗格中选择 Lifecycle Policy（生命周期策略）。

   Note

   如果您未看到生命周期策略选项，请确保您处于存储库镜像列表视图中。
6. 在存储库生命周期策略页面上，选择 Create rule（创建规则）。
7. 输入生命周期策略规则的以下详细信息。
   a. 对于规则优先级，输入该规则优先级的编号。

b. 对于规则描述，输入该生命周期策略规则的说明。
c. 对于镜像状态，选择已标记、未标记或者任何。
d. 如果您为镜像状态指定 `Tagged`，然后设置标签前缀，则可以有选择地指定根据其执行生命周期策略操作的镜像标签列表。如果指定 `Untagged`，此字段必须为空。
e. 对于匹配条件，选择自推送镜像以来或者镜像计数超过 (如果适用) 的值。
f. 选择保存。

8. 重复步骤 5-7 以创建其他生命周期策略规则。

创建生命周期策略 (Amazon CLI)

使用 Amazon CLI 创建生命周期策略

1. 获取要为其创建生命周期策略的存储库的名称。

```
aws ecr describe-repositories
```

2. 创建名为 `policy.json` 的本地文件，其中包含生命周期策略的内容。有关生命周期策略示例，请参阅生命周期策略的示例 (p. 53)。
3. 通过指定存储库名称并引用创建的生命周期策略 JSON 文件来创建生命周期策略。

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

生命周期策略的示例

以下是示例生命周期策略，其中显示了语法。

主题

根据镜像使用期限筛选

以下示例显示了策略的生命周期策略语法，该策略使超过 `14` 天的未标记镜像过期：

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

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

根据镜像计数筛选

以下示例显示了仅保留一个未标记镜像并使所有其他镜像过期的策略的生命周期策略语法：

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

根据多个规则筛选

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

示例 A

存储库内容：

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

生命周期策略文本：

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

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

示例 B

这与前面的示例相同，但规则优先级顺序会更改以展示结果。

存储库内容：

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

生命周期策略文本：

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

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

在单个规则中筛选多个标签

以下示例为单个规则中的多个标签前缀指定生命周期策略语法。此处提供了一个示例存储库和生命周期策略以及结果的说明。

示例 A

在单个规则上指定多个标签前缀时，镜像必须与所有列出的标签前缀匹配。

存储库内容：

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

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

此生命周期策略的逻辑是：

- 规则 1 标识了标记为 `alpha` 和 `beta` 的镜像 它可以看到镜像 C 和 F。它应该标记超过五天的镜像，镜像 C 符合此条件。

- 结果：镜像 C 已过期。

示例 B

以下示例说明标签不是独占的。

存储库内容：

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

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

此生命周期策略的逻辑是：

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

筛选所有镜像

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

示例 A

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

存储库内容：

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

```
{
```

```
    "rules": [
        {
            "rulePriority": 1,
            "description": "Rule 1",
            "selection": {
                "tagStatus": "any",
                "countType": "imageCountMoreThan",
                "countNumber": 1
            },
            "action": {
                "type": "expire"
            }
        }
    ]
}
```

此生命周期策略的逻辑是：

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

示例 B

以下示例说明了将所有规则类型组合在一个策略中的生命周期策略。

存储库内容：

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

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

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

镜像标签可变性

您可以将存储库配置为启用标签可变性，以防止覆盖镜像标签。为存储库配置了不可变标签后，如果您尝试推送某个镜像但其标签在存储库中已存在，将返回 `ImageTagAlreadyExistsException` 错误。当存储库启用标签不可变性时，这会影响所有标签，您不能将某些标签配置为不可变，而将其他标签配置为可变。

使用 Amazon Web Services Management Console 和 Amazon 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镜像标签的可变性。对于控制台步骤，请参阅 创建私有存储库 (p. 20) 和 编辑私有存储库 (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镜像存储库。

- create-repository (Amazon CLI)

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

- New-ECRRepository (Amazon Tools for Windows PowerShell)

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

更新现有存储库的镜像标签可变性设置

使用以下命令之一更新现有存储库的镜像标签可变性设置。

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

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

- Write-ECRImageTagMutability (Amazon Tools for Windows PowerShell)

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

镜像扫描

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

- 增强扫描—Amazon ECR 与 Amazon Inspector 集成以提供对您的存储库的自动连续扫描。扫描容器镜像是否存在操作系统和编程语言包漏洞。随着新漏洞的出现，扫描结果将更新，Amazon Inspector 会向 EventBridge 发送一个事件以通知您。
- 基本扫描—Amazon ECR 使用开源 Clair 项目中的常见漏洞和披露 (CVE) 数据库。通过基本扫描，您可以将存储库配置为在推送时扫描，也可以执行手动扫描，而 Amazon ECR 会提供扫描结果列表。

使用筛选条件

为私有注册表配置镜像扫描后，您可以指定扫描所有存储库，也可以指定筛选条件来限定存储库扫描范围。

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

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

指定筛选条件后，没有通配符的筛选条件将匹配包含该筛选条件的所有存储库名称。带通配符 (*) 的筛选条件匹配任何存储库名称，通配符会在其中替换存储库名称中的零个或多个字符。下表提供了示例，其中存储库名称在水平轴上表示，示例筛选条件在垂直轴上指定。

	prod	repo-prod	prod-repo	repo-prod-repo	prodrepo
prod	✓是	✓是	✓是	✓是	✓是
*prod	✓是	✓是	✗否	✗否	✗否
prod*	✓是	✗否	✓是	✗否	✓是
prod	✓是	✓是	✓是	✓是	✓是
prod*repo	✗否	✗否	✓是	✗否	✓是

主题

- 增强扫描 (p. 61)

- 基本扫描 (p. 67)

增强扫描

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

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

增强扫描的注意事项

启用 Amazon ECR 增强扫描时应考虑以下因素。

- 以下区域不支持增强扫描：
  - 亚太地区（大阪）(`ap-northeast-3`)
  - 亚太地区（雅加达）（`ap-southeast-3`）
  - 非洲（开普敦）(`af-south-1`)
- Amazon Inspector 支持扫描特定操作系统。获取完整列表，请参阅 Amazon Inspector 用户指南中的支持的操作系统 - Amazon ECR 扫描。
- Amazon Inspector 使用服务相关 IAM 角色，该角色提供了为存储库提供增强扫描所需的权限。为私有注册表启用增强扫描后，Amazon Inspector 会自动创建服务相关 IAM 角色。有关更多信息，请参阅 Amazon Inspector 用户指南中的将服务相关角色用于 Amazon Inspector。
- 如果您的私有注册表启用了增强扫描功能，则只能使用增强扫描来扫描与扫描筛选条件匹配的所有存储库。不符合任何筛选条件的任何存储库将使用 `Off` 扫描频率，并且不会被扫描。不支持使用增强扫描进行手动扫描。有关更多信息，请参阅 使用筛选条件 (p. 60)。
- 如果您为推送扫描和连续扫描分别指定了筛选条件，并且同一存储库符合多个筛选条件，则 Amazon ECR 会强制执行该存储库的连续扫描筛选条件，而不是推送扫描筛选条件。
- 对存储库启用连续扫描后，如果过去 30 天内没有根据镜像推送时间戳更新镜像，则会暂停对该镜像的连续扫描。暂停扫描的镜像将显示扫描状态 `SCAN_ELIGIBILITY_EXPIRED`。
- 启用增强扫描后，当存储库的扫描频率发生变化时，Amazon ECR 会向 EventBridge 发送一个事件。当初始扫描完成且创建、更新或关闭了镜像扫描结果时，Amazon Inspector 会向 EventBridge 发送事件。

所需的 IAM 权限

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

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

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "inspector2:Enable",
                "inspector2:Disable",
                "inspector2:ListFindings",
```

```
                "inspector2:ListAccountPermissions",
                "inspector2:ListCoverage"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "inspector2.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

启用增强扫描

要启用增强扫描 (Amazon Web Services Management Console)

为私有注册表 (Amazon Web Services Management Console) 启用增强扫描

扫描配置在每个区域的私有注册表级别定义。

1. 从 https://console.aws.amazon.com/ecr/repositories 打开 Amazon ECR 控制台。
2. 从导航栏中，选择要为其设置扫描配置的区域。
3. 在导航窗格中，选择 Private registry（私有注册表）、Scanning（扫描）。
4. 在 Scanning configuration（扫描配置）页面中，为 Scan type（扫描类型）选择 Enhanced scanning（增强扫描）。
5. （可选）默认情况下，Enhanced scanning（增强扫描）处于选中状态时，所有存储库都设置为连续扫描。您可以通过取消选中 Continuously scan all repositories（连续扫描所有存储库）复选框来更改默认扫描配置。然后，您可以将所有存储库配置为在推送时扫描，也可以为连续扫描和推送时扫描指定单独的扫描筛选条件。设置扫描筛选条件后，您可以选择 Preview repository matches（预览存储库匹配）以验证注册表中的哪些存储库与定义的筛选条件匹配。

   Important

   没有通配符的筛选条件将匹配包含该筛选条件的所有存储库名称。带通配符 (*) 的筛选条件匹配存储库名称，通配符会替换存储库名称中的零个或多个字符。
6. 选择保存。
7. 在您要在其中启用增强扫描的每个区域中重复这些步骤。

要启用增强扫描 (Amazon CLI)

使用以下 Amazon CLI 命令通过 Amazon CLI 为您的私有注册表启用增强扫描。您可以使用 `rules` 对象指定扫描筛选条件。

- put-registry-scanning-configuration (Amazon CLI)

  以下示例为您的私有注册表启用增强扫描。默认情况下，如果没有指定 `rules`，Amazon ECR 会将扫描配置设置为对所有存储库进行持续扫描。

```
aws ecr put-registry-scanning-configuration \
     --scan-type ENHANCED \
```

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

以下示例为您的私有注册表启用增强扫描并指定扫描筛选条件。示例中的扫描筛选条件允许连续扫描名称中带有 prod 的所有存储库。

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

以下示例为您的私有注册表启用增强扫描并指定多个扫描筛选条件。示例中的扫描筛选条件允许连续扫描名称中带有 prod 的所有存储库并且仅为所有其他存储库支持推送时扫描。

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

更改增强扫描持续时间

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

更改增强扫描持续时间设置

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

EventBridge 事件

启用增强扫描后，当存储库的扫描频率发生变化时，Amazon ECR 会向 EventBridge 发送一个事件。当初始扫描完成且创建、更新或关闭了镜像扫描结果时，Amazon Inspector 会向 EventBridge 发送事件。

存储库扫描频率更改事件

为注册表启用增强扫描后，当启用了增强扫描的资源发生更改时，Amazon ECR 将发送以下事件。这包括正在创建的新存储库、正在更改的存储库的扫描频率，或者在启用了增强扫描功能的存储库中创建或删除镜像的时间。有关更多信息，请参阅 镜像扫描 (p. 60)。

```
{
 "version": "0",
 "id": "0c18352a-a4d4-6853-ef53-0abEXAMPLE",
 "detail-type": "ECR Scan Resource Change",
 "source": "aws.ecr",
 "account": "123456789012",
 "time": "2021-10-14T20:53:46Z",
```

```
 "region": "us-east-1",
 "resources": [],
 "detail": {
  "action-type": "SCAN_FREQUENCY_CHANGE",
  "repositories": [{
    "repository-name": "repository-1",
    "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-1",
    "scan-frequency": "SCAN_ON_PUSH",
    "previous-scan-frequency": "MANUAL"
   },
   {
    "repository-name": "repository-2",
    "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-2",
    "scan-frequency": "CONTINUOUS_SCAN",
    "previous-scan-frequency": "SCAN_ON_PUSH"
   },
   {
    "repository-name": "repository-3",
    "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-3",
    "scan-frequency": "CONTINUOUS_SCAN",
    "previous-scan-frequency": "SCAN_ON_PUSH"
   }
  ],
  "resource-type": "REPOSITORY",
  "scan-type": "ENHANCED"
 }
}
```

初始镜像扫描的事件（增强扫描）

为注册表启用增强扫描后，当初始镜像扫描完成时，Amazon Inspector 会发送以下事件。`finding-severity-counts` 参数仅返回严重性级别的值 (如果存在)。例如，如果镜像不包含任何 `CRITICAL` 级别的结果，则不会返回任何关键计数。有关更多信息，请参阅 增强扫描 (p. 61)。

事件模式：

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

输出示例：

```
{
    "version": "0",
    "id": "739c0d3c-4f02-85c7-5a88-94a9EXAMPLE",
    "detail-type": "Inspector2 Scan",
    "source": "aws.inspector2",
    "account": "123456789012",
    "time": "2021-12-03T18:03:16Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:ecr:us-east-2:123456789012:repository/amazon/amazon-ecs-sample"
    ],
    "detail": {
        "scan-status": "INITIAL_SCAN_COMPLETE",
        "repository-name": "arn:aws:ecr:us-east-2:123456789012:repository/amazon/amazon-
ecs-sample",
        "finding-severity-counts": {
            "CRITICAL": 7,
            "HIGH": 61,
            "MEDIUM": 62,
            "TOTAL": 158
```

```
        },
        "image-digest":
 "sha256:36c7b282abd0186e01419f2e58743e1bf635808231049bbc9d77e5EXAMPLE",
        "image-tags": [
            "latest"
        ]
    }
}
```

镜像扫描结果更新的事件（增强扫描）

为注册表启用增强扫描后，当镜像扫描结果被创建、更新或关闭时，Amazon Inspector 会发送以下事件。有关更多信息，请参阅 增强扫描 (p. 61)。

事件模式：

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

输出示例：

```
{
    "version": "0",
    "id": "42dbea55-45ad-b2b4-87a8-afaEXAMPLE",
    "detail-type": "Inspector2 Finding",
    "source": "aws.inspector2",
    "account": "123456789012",
    "time": "2021-12-03T18:02:30Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:ecr:us-east-2:123456789012:repository/amazon/amazon-ecs-sample/
sha256:36c7b282abd0186e01419f2e58743e1bf635808231049bbc9d77eEXAMPLE"
    ],
    "detail": {
        "awsAccountId": "123456789012",
        "description": "In libssh2 v1.9.0 and earlier versions, the SSH_MSG_DISCONNECT
 logic in packet.c has an integer overflow in a bounds check, enabling an attacker to
 specify an arbitrary (out-of-bounds) offset for a subsequent memory read. A crafted
 SSH server may be able to disclose sensitive information or cause a denial of service
 condition on the client system when a user connects to the server.",
        "findingArn": "arn:aws:inspector2:us-east-2:123456789012:finding/
be674aaddd0f75ac632055EXAMPLE",
        "firstObservedAt": "Dec 3, 2021, 6:02:30 PM",
        "inspectorScore": 6.5,
        "inspectorScoreDetails": {
            "adjustedCvss": {
                "adjustments": [],
                "cvssSource": "REDHAT_CVE",
                "score": 6.5,
                "scoreSource": "REDHAT_CVE",
                "scoringVector": "CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N",
                "version": "3.0"
            }
        },
        "lastObservedAt": "Dec 3, 2021, 6:02:30 PM",
        "packageVulnerabilityDetails": {
            "cvss": [
                {
                    "baseScore": 6.5,
                    "scoringVector": "CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N",
                    "source": "REDHAT_CVE",
```

```
                    "version": "3.0"
                },
                {
                    "baseScore": 5.8,
                    "scoringVector": "AV:N/AC:M/Au:N/C:P/I:N/A:P",
                    "source": "NVD",
                    "version": "2.0"
                },
                {
                    "baseScore": 8.1,
                    "scoringVector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:H",
                    "source": "NVD",
                    "version": "3.1"
                }
            ],
            "referenceUrls": [
                "https://access.redhat.com/errata/RHSA-2020:3915"
            ],
            "source": "REDHAT_CVE",
            "sourceUrl": "https://access.redhat.com/security/cve/CVE-2019-17498",
            "vendorCreatedAt": "Oct 16, 2019, 12:00:00 AM",
            "vendorSeverity": "Moderate",
            "vulnerabilityId": "CVE-2019-17498",
            "vulnerablePackages": [
                {
                    "arch": "X86_64",
                    "epoch": 0,
                    "name": "libssh2",
                    "packageManager": "OS",
                    "release": "12.amzn2.2",
                    "sourceLayerHash":
 "sha256:72d97abdfae3b3c933ff41e39779cc72853d7bd9dc1e4800c5294dEXAMPLE",
                    "version": "1.4.3"
                }
            ]
        },
        "remediation": {
            "recommendation": {
                "text": "Update all packages in the vulnerable packages section to their
 latest versions."
            }
        },
        "resources": [
            {
                "details": {
                    "awsEcrContainerImage": {
                        "architecture": "amd64",
                        "imageHash":
 "sha256:36c7b282abd0186e01419f2e58743e1bf635808231049bbc9d77e5EXAMPLE",
                        "imageTags": [
                            "latest"
                        ],
                        "platform": "AMAZON_LINUX_2",
                        "pushedAt": "Dec 3, 2021, 6:02:13 PM",
                        "registry": "123456789012",
                        "repositoryName": "amazon/amazon-ecs-sample"
                    }
                },
                "id": "arn:aws:ecr:us-east-2:123456789012:repository/amazon/amazon-ecs-
sample/sha256:36c7b282abd0186e01419f2e58743e1bf635808231049bbc9d77EXAMPLE",
                "partition": "N/A",
                "region": "N/A",
                "type": "AWS_ECR_CONTAINER_IMAGE"
            }
        ],
        "severity": "MEDIUM",
```

```
        "status": "ACTIVE",
        "title": "CVE-2019-17498 - libssh2",
        "type": "PACKAGE_VULNERABILITY",
        "updatedAt": "Dec 3, 2021, 6:02:30 PM"
    }
}
```

检索镜像扫描查找结果

您可以检索上次完成的镜像扫描的扫描结果。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。

有关扫描镜像时的常见问题的排查详细信息，请参阅 排查镜像扫描问题 (p. 128)。

检索镜像扫描结果 (Amazon Web Services Management Console)

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

1. 从 https://console.aws.amazon.com/ecr/repositories 打开 Amazon ECR 控制台。
2. 从导航栏中，选择您的存储库所在的区域。
3. 在导航窗格中，选择存储库。
4. 在存储库页面上，选择包含要扫描检索结果的镜像的存储库。
5. 在 Images（镜像）页面的 Vulnerabilities（漏洞）下，为要检索其扫描结果的镜像选择 See findings（查看详细信息）。
6. 查看 Findings（结果）时，Name（名称）列中的漏洞名称是指向 Amazon Inspector 控制台的链接，在该控制台中，您可以查看更多详细信息。

检索镜像扫描结果 (Amazon CLI)

通过 Amazon CLI 使用以下 Amazon CLI 命令检索镜像扫描结果。您可以使用 `imageTag` 或 `imageDigest` 指定镜像，这两者都可以使用 list-images CLI 命令获取。

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

  以下示例使用镜像标签。

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

  以下示例使用镜像摘要。

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

基本扫描

Amazon ECR 提供基本扫描类型，该类型使用开源 Clair 项目中的常见漏洞和披露 (CVE) 数据库。在私有注册表上启用基本扫描后，您可以配置存储库筛选条件以指定将哪些存储库设置为推送时扫描，或者也可以执行手动扫描。Amazon ECR 提供了扫描结果的列表。对于每个容器镜像，可以每 24 小时扫描一次。Amazon ECR 使用开源 Clair 项目中的常见漏洞和披露 (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镜像的安全性的信息。有关 Clair 的更多信息，请参见 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。

使用基本扫描时，您可以指定按推送筛选条件进行扫描，以指定将哪些存储库设置为在推送新镜像时进行镜像扫描。对于任何不符合推送筛选条件的存储库，都将设置为 manual（手动）扫描频率，这意味着必须手动触发扫描才能执行扫描。可以为每个镜像检索上次完成的镜像扫描结果。Amazon ECR 在镜像扫描完成后向 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. 110)。

有关扫描镜像时的常见问题的排查详细信息，请参阅 排查镜像扫描问题 (p. 128)。

启用基本扫描

默认情况下，Amazon ECR 在所有私有注册表上启用基本扫描。因此，除非您更改了私有注册表上的扫描设置，否则不必启用基本扫描。您可以使用以下步骤验证是否启用了基本扫描并在推送筛选条件上定义一个或多个扫描。

要为私有注册表 (Amazon Web Services Management Console) 启用基本扫描

扫描配置在每个区域的私有注册表级别定义。

1. 从 https://console.aws.amazon.com/ecr/repositories 打开 Amazon ECR 控制台。
2. 从导航栏中，选择要为其设置扫描配置的区域。
3. 在导航窗格中，选择 Private registry（私有注册表）、Scanning（扫描）。
4. 在 Scanning configuration（扫描配置）页面中，为 Scan type（扫描类型）选择 Basic scanning（基本扫描）。
5. 默认情况下，您的所有存储库都设置为手动扫描。您可以选择通过指定 Scan on push filters（推送时扫描筛选条件）来启用推送时扫描。您可以为所有存储库或单个存储库设置推送时扫描。有关更多信息，请参阅 使用筛选条件 (p. 60)。

手动扫描镜像

当您要扫描存储库中未配置为推送时扫描的镜像时，可以手动开始镜像扫描。每天只能扫描一次镜像。此限制包括初始推送时扫描 (如果已启用) 以及任何手动扫描。

有关扫描镜像时的常见问题的排查详细信息，请参阅 排查镜像扫描问题 (p. 128)。

开始手动扫描镜像 (控制台)

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

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

开始手动扫描镜像 (Amazon CLI)

使用以下 Amazon CLI 命令启动镜像的手动扫描。您可以使用 `imageTag` 或 `imageDigest` 指定镜像，这两者都可以使用 list-images CLI 命令获取。

- start-image-scan (Amazon CLI)

以下示例使用镜像标签。

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

以下示例使用镜像摘要。

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

开始手动扫描镜像 (Amazon Tools for Windows PowerShell)

使用以下 Amazon Tools for Windows PowerShell 命令启动镜像的手动扫描。您可以使用 `ImageId_ImageTag` 或 `ImageId_ImageDigest` 指定镜像，这两者都可以使用 Get-ECRImage CLI 命令获取。

- Get-ECRImageScanFinding (Amazon Tools for Windows PowerShell)

  以下示例使用镜像标签。

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

  以下示例使用镜像摘要。

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

检索镜像扫描查找结果

您可以检索上次完成的镜像扫描的扫描结果。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。

有关扫描镜像时的常见问题的排查详细信息，请参阅 排查镜像扫描问题 (p. 128)。

检索镜像扫描结果 (控制台)

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

1. 从 https://console.aws.amazon.com/ecr/repositories 打开 Amazon ECR 控制台。
2. 从导航栏中，选择您创建存储库的区域。
3. 在导航窗格中，选择存储库。
4. 在存储库页面上，选择包含要扫描检索结果的镜像的存储库。
5. 在镜像页面的漏洞列下，选择要检索其扫描结果的镜像的详细信息。

检索镜像扫描结果 (Amazon CLI)

通过 Amazon CLI 使用以下 Amazon CLI 命令检索镜像扫描结果。您可以使用 `imageTag` 或 `imageDigest` 指定镜像，这两者都可以使用 list-images CLI 命令获取。

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

  以下示例使用镜像标签。

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

以下示例使用镜像摘要。

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

检索镜像扫描结果 (Amazon Tools for Windows PowerShell)

使用以下 Amazon Tools for Windows PowerShell 命令检索镜像扫描结果。您可以使用 `ImageId_ImageTag` 或 `ImageId_ImageDigest` 指定镜像，这两者都可以使用 Get-ECRImage CLI 命令获取。

- Get-ECRImageScanFinding (Amazon Tools for Windows PowerShell)

  以下示例使用镜像标签。

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

以下示例使用镜像摘要。

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

容器镜像清单格式

Amazon ECR 支持以下容器镜像清单格式：

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

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

- 能够为单个镜像使用多个标签。
- 支持存储 Windows 容器镜像。有关更多信息，请参阅 Amazon Elastic Container Service 开发人员指南中的将 Windows镜像推送到 Amazon ECR。

Amazon ECR 镜像清单转换

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

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

在从 Amazon ECR 按标签提取镜像时，Amazon ECR 将返回存储在存储库中的镜像清单格式。仅当客户端理解该格式时，才会将其返回。如果客户端不理解所存储的镜像清单格式，则 Amazon ECR 会将镜像清

单转换为客户端能够理解的格式。例如，如果 Docker 1.9 客户端请求的镜像清单存储格式为 Docker Image Manifest V2 Schema 2，那么 Amazon ECR 将以 Docker Image Manifest V2 Schema 1 格式返回该清单。下表介绍了按标签提取镜像时 Amazon ECR 支持的可用转换：

客户端请求的架构	作为 V2 Schema 1 推送到 ECR	作为 V2 Schema 2 推送到 ECR	作为 OCI 推送到 ECR
V2 Schema 1	无需转换	已转换为 V2 Schema 1	已转换为 V2 Schema 1
V2 Schema 2	无可用转换，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

如果按摘要提取镜像，则没有可用的转换。您的客户端必须了解存储在 Amazon ECR 中的镜像清单格式。如果您在 Docker 1.9 或更旧版本的客户端上按摘要请求 Docker Image Manifest V2 Schema 2 镜像，则无法提取镜像。有关更多信息，请参阅 Docker 文档中的注册表兼容性。
在此示例中，如果按标签请求同一镜像，Amazon ECR 会将镜像清单转换为客户端能够理解的格式。镜像提取成功。

将 Amazon ECR 映像与 Amazon ECS 结合使用

您可以在 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 开发人员指南中的经 Amazon ECS 优化的 AMI 版本。
- 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。要检查您的角色是否支持 Amazon ECR，请参阅 Amazon Elastic Container Service 开发人员指南中的 Amazon ECS 容器实例 IAM 角色。

- 在 Amazon ECS 任务定义中，确保对 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-repository`*`:`*`latest`*。

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

```
{
    "family": "task-definition-name",
    ...
    "containerDefinitions": [
        {
            "name": "container-name",
            "image": "aws_account_id.dkr.ecr.region.amazonaws.com/my-repository:latest",
            ...
        }
    ],
    ...
}
```

将 Amazon ECR 映像与 Amazon EKS 结合使用

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

- 对于在托管式节点或自行管理的节点上托管的 Amazon EKS 工作负载，必须提供 Amazon EKS Worker 节点 IAM 角色（`NodeInstanceRole`）。Amazon EKS Worker 节点 IAM 角色必须包含以下适用于 Amazon ECR 的 IAM 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 Amazon EKS 入门中的 `eksctl` 或 Amazon CloudFormation 模板创建集群和工作节点组，则预设情况下，这些 IAM 权限将应用于您的工作节点 IAM 角色。
- 对于在 Amazon Fargate 上托管的 Amazon EKS 工作负载，您必须使用 Fargate 容器组（Pod）执行角色，该角色为您的 Pod 提供了从私有 Amazon ECR 存储库中提取镜像的权限。有关更多信息，请参阅创建 Fargate Pod 执行角色。
- 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-repository*`:`*latest*。

使用 Amazon EKS 安装托管在 Amazon ECR 上的 Helm Chart

您在 Amazon ECR 中托管的 Helm Chart 可以安装在您的 Amazon EKS 集群上。以下步骤演示了这一点。

先决条件

开始使用之前，请确保您已完成以下步骤。

- 安装最新版本的 Helm 客户端。这些步骤是使用 Helm 版本 `3.9.0` 编写的。有关更多信息，请参阅安装 Helm。
- 您至少已在计算机上安装了 Amazon CLI 的版本 `1.23.9` 或 `2.6.3`。有关更多信息，请参阅安装或更新 Amazon CLI 的最新版本。
- 您已将 Helm Chart 推送到您的 Amazon ECR 存储库。有关更多信息，请参阅推送 Helm Chart (p. 34)。
- 您已配置 `kubectl` 以使用 Amazon EKS。有关更多信息，请参阅 Amazon EKS 用户指南中的为 Amazon EKS 创建 `kubeconfig`。如果集群的以下命令成功，说明您已正确配置。

```
kubectl get svc
```

将 Amazon ECR 托管的 Helm Chart 安装到 Amazon EKS 集群

1. 向托管 Helm Chart 的 Amazon ECR 注册表验证您的 Helm 客户端。必须针对每个注册表获得授权令牌，令牌有效期为 12 小时。有关更多信息，请参阅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
   ```

2. 安装图表。将 *helm-test-chart* 替换为您的存储库，并将 *0.1.0* 替换为 Helm 图表的标签。

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

   输出应如下所示：

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

3. 验证图表安装。

   ```
   helm list -n default
   ```

   输出示例：

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

4. （可选）查看已安装的 Helm 图表 `ConfigMap`。

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

5. 完成后，您可以从集群中删除图表版本。

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

Amazon Linux 容器镜像

构建 Amazon Linux 容器镜像的软件组件与 Amazon Linux AMI 中包含的软件组件相同。作为 Docker 工作负载的基本镜像，它可用在任何环境中。如果您在 Amazon EC2 中针对应用程序使用 Amazon Linux AMI，就可以使用 Amazon Linux 容器镜像将您的应用程序容器化。

可以在本地开发环境中使用 Amazon Linux 容器镜像，然后使用 Amazon ECS 将应用程序推送到 Amazon。有关更多信息，请参阅 将 Amazon ECR 映像与 Amazon ECS 结合使用 (p. 71)。

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

从 Docker Hub 提取 Amazon Linux 容器镜像

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

```
docker pull amazonlinux
```

2. (可选) 在本地运行容器。

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

Amazon Elastic Container Registry 中的安全性

Amazon 十分重视云安全性。作为 Amazon 客户，您将从专为满足大多数安全敏感型企业的要求而打造的数据中心和网络架构中受益。

安全性是 Amazon 和您的共同责任。责任共担模式将其描述为云的安全性和云中的安全性：

- 云的安全性 – Amazon负责保护在Amazon云中运行Amazon服务的基础设施。Amazon还向您提供可安全使用的服务。作为 Amazon 合规性计划的一部分，第三方审核人员将定期测试和验证安全性的有效性。要了解适用于 Amazon ECR 的合规性计划，请参阅合规性计划范围内的 Amazon 服务。
- 云中的安全性 - 您的责任由您使用的 Amazon 服务决定。您还需要对其它因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。

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

主题

适用于 Amazon Elastic Container Registry 的 Identity and Access Management

Amazon Identity and Access Management (IAM) 是一项 Amazon Web Service，可以帮助管理员安全地控制对 Amazon 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）使用 Amazon ECR 资源。IAM 是一项无需额外费用即可使用的 Amazon Web Service。

主题

- 排查 Amazon Elastic Container Registry 的身份和访问权限问题 (p. 93)

Audience

Amazon Identity and Access Management (IAM) 的使用方式因您可以在 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，您的管理员会为您提供所需的凭证和权限。随着您使用更多 Amazon ECR 功能来完成工作，您可能需要额外权限。了解如何管理访问权限可帮助您向管理员请求适合的权限。如果您无法访问 Amazon ECR 中的功能，请参阅排查 Amazon Elastic Container Registry 的身份和访问权限问题 (p. 93)。

服务管理员 – 如果您在公司负责管理 Amazon ECR 资源，您可能对 Amazon ECR 具有完全访问权限。您有责任确定您的服务用户应访问哪些 Amazon ECR 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要了解有关您的公司如何将 IAM 与 Amazon ECR 搭配使用的更多信息，请参阅Amazon Elastic Container Registry 如何与 IAM 结合使用 (p. 79)。

IAM 管理员 – 如果您是 IAM 管理员，您可能希望了解如何编写策略以管理对 Amazon ECR 的访问的详细信息。要查看您可在 IAM 中使用的 Amazon ECR 基于身份的策略示例，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 90)。

使用身份进行身份验证

身份验证是您使用身份凭证登录 Amazon 的方法。有关使用Amazon Web Services Management Console 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Amazon Web Services Management Console。

您必须作为Amazon Web Services 账户根用户、IAM 用户或代入 IAM 角色以进行身份验证（登录到 Amazon）。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。在这些情况下，您的管理员以前使用 IAM 角色设置了联合身份验证。在您使用来自其它公司的凭证访问 Amazon 时，您间接地代入了角色。

要直接登录到Amazon Web Services Management Console，请将密码与根用户电子邮件地址或 IAM 用户名一起使用。您可以使用根用户或 IAM 用户访问密钥以编程方式访问 Amazon。Amazon 提供了 SDK 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 Amazon 工具，则必须自行对请求签名。使用 Signature Version 4（用于对入站 API 请求进行验证的协议）完成此操作。有关验证请求的更多信息，请参阅《Amazon 一般参考》中的 Signature Version 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它安全信息。例如，Amazon 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 Amazon 中使用多重身份验证 (MFA)。

Amazon Web Services 账户根用户

当您创建 Amazon Web Services 账户时，最初使用的是一个对账户中所有 Amazon Web Services和资源拥有完全访问权限的登录身份。此身份称为 Amazon Web Services 账户根用户，使用您创建账户时所用的电子邮件地址和密码登录，即可获得该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关要求您以根用户身份登录的任务的完整列表，请参阅《Amazon 一般参考》中的需要根用户凭证的任务。

IAM 用户和组

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

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

IAM 角色

IAM 角色是Amazon Web Services 账户中具有特定权限的身份。它类似于 IAM 用户，但与特定人员不关联。您可以通过切换角色，在 Amazon Web Services Management Console 中暂时代入 IAM 角色。您可以调用 Amazon CLI 或 Amazon API 操作或使用自定义 URL 以代入角色。有关使用角色的方法的更多信息，请参阅 IAM 用户指南 中的使用 IAM 角色。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 临时 IAM 用户权限 – IAM 用户可以代入 IAM 角色，以暂时获得不同的权限以执行特定的任务。
- 联合用户访问 – 您可以不创建 IAM 用户，而是使用来自 Amazon Directory Service、您的企业用户目录、Web 身份提供商或 IAM Identity Center 身份存储的现有身份。这些身份称为联合身份。要向联合身份分配权限，您可以创建角色并为角色定义权限。当外部身份进行身份验证时，该身份将与角色相关联并被授予其定义的权限。如果您使用 IAM Identity Center，则需要配置权限集。IAM Identity Center 将权限集与 IAM 中的角色相关联，以控制您的身份在进行身份验证后可以访问的内容。有关身份联合验证的更多信息，请参阅《IAM 用户指南》中的为第三方身份提供商创建角色。有关 IAM Identity Center 的更多信息，请参阅《Amazon IAM Identity Center (successor to Amazon Single Sign-On) 用户指南》中的什么是 IAM Identity Center？。
- 跨账户访问 – 您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些 Amazon Web Services，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 – 某些 Amazon Web Services使用其它 Amazon Web Services中的功能。例如，当您在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Simple Storage Service（Amazon S3）中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
  - 主体权限 – 当您使用 IAM 用户或角色在 Amazon 中执行操作时，您将被视为主体。策略向主体授予权限。使用某些服务时，您可能会执行一个操作，此操作然后在不同服务中触发另一个操作。在这种情况下，您必须具有执行这两个操作的权限。要查看某个操作是否需要策略中的其他相关操作，请参阅服务授权参考 中的 Amazon Elastic Compute Cloud 的操作、资源和条件键。
  - 服务角色 – 服务角色是服务代表您在您的账户中执行操作而担任的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅 IAM 用户指南中的创建向 Amazon Web Service 委派权限的角色。
  - 服务相关角色 – 服务相关角色是与 Amazon Web Service关联的一种服务角色。服务可以代入代表您执行操作的角色。服务相关角色显示在您的 IAM 账户中，并归该服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 Amazon EC2 上运行的应用程序 – 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 Amazon CLI 或 Amazon API 请求的应用程序的临时凭证。这优先于在 EC2 实例中存储访问密钥。要将 Amazon 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 角色为 Amazon EC2 实例上运行的应用程序授予权限。

要了解是使用 IAM 角色还是 IAM 用户，请参阅 IAM 用户指南中的何时创建 IAM 角色（而不是用户）。

使用策略管理访问

您将创建策略并将其附加到 Amazon 身份或资源，以控制 Amazon 中的访问。策略是 Amazon 中的对象；在与身份或资源相关联时，策略定义它们的权限。在主体（用户、根用户或角色会话）发出请求时，Amazon 将评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略在 Amazon 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 策略概览。

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

每个 IAM 实体（用户或角色）最初没有任何权限。原定设置情况下，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 policy 定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。具有该策略的用户可以从 Amazon Web Services Management Console、Amazon CLI 或 Amazon API 获取角色信息。

基于身份的策略

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

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

基于资源的策略

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

其他策略类型

Amazon 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 – 权限边界是一个高级功能，用于设置基于身份的策略可以为 IAM 实体（IAM 用户或角色）授予的最大权限。您可为实体设置权限边界。这些结果权限是实体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 Amazon Organizations 中的最大权限。Amazon Organizations 服务可以分组和集中管理您的企业拥有的多个 Amazon Web Services 账户 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的权限，包括每个 Amazon Web Services 账户 根用户。有关 Organizations 和 SCP 的更多信息，请参阅 Amazon Organizations 用户指南中的 SCP 的工作原理。

- 会话策略 – 会话策略是当您以编程方式为角色或联合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 Amazon 如何确定在涉及多种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 结合使用

在使用 IAM 管理对 Amazon ECR 的访问权限之前，您应该了解哪些 IAM 功能可用于 Amazon ECR。要大致了解 Amazon ECR 和其他 Amazon 服务如何与 IAM 一起使用，请参阅《IAM 用户指南》中的与 IAM 一起使用的 Amazon 服务。

主题

Amazon ECR 基于身份的策略

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

操作

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

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 Amazon 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。

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

要在单个语句中指定多项操作，请使用逗号将它们隔开，如下所示：

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

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

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

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

资源

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

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

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

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

Amazon ECR 存储库资源具有以下 ARN：

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

有关 ARN 格式的更多信息，请参阅 Amazon Resource Name（ARN）和 Amazon 服务命名空间。

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

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

要指定属于特定账户的所有存储库，请使用通配符 (*)：

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

要在单个语句中指定多个资源，请使用逗号分隔 ARN。

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

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

条件键

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

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

如果您在一个语句中指定多个 `Condition` 元素，或在单个 `Condition` 元素中指定多个键，则 Amazon 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 Amazon 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，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅 IAM 用户指南 中的 IAM policy 元素：变量和标签。

Amazon支持全局条件键和特定于服务的条件键。要查看所有 Amazon 全局条件键，请参阅《IAM 用户指南》中的 Amazon 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 Amazon 全局条件键，请参阅《IAM 用户指南》中的 Amazon 全局条件上下文键。

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. 92)。

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

示例

要查看 Amazon ECR 基于身份的策略的示例，请参阅 Amazon Elastic Container Registry 基于身份的策略示例 (p. 90)。

Amazon ECR 基于资源的策略

基于资源的策略是 JSON 策略文档，它们指定了指定委托人可在 Amazon ECR 资源上执行的操作以及在什么条件下可执行。Amazon ECR 支持针对 Amazon ECR 存储库的基于资源的权限策略。基于资源的策略允许您基于资源向其他 账户授予使用权限。您也可以使用基于资源的策略以允许 Amazon 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Amazon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 IAM 用户指南 中的 IAM 角色与基于资源的策略有何不同。

Amazon ECR 服务仅支持一种类型的基于资源的策略（称为容器策略），这种策略附加到存储库。这个策略定义哪些委托人实体（账户、用户、角色和联合身份用户）可以在存储库上执行操作。

要了解如何将基于资源的策略附加到存储库，请参阅私有存储库策略 (p. 22)。

示例

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

基于 Amazon ECR 标签的授权

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

要查看基于身份的策略（用于根据资源上的标签来限制对该资源的访问）的示例，请参阅使用基于标签的访问控制 (p. 92)。

Amazon ECR IAM 角色

IAM 角色是 Amazon 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，担任 IAM 角色或担任跨账户角色。您可以通过调用 Amazon STS API 操作（如 AssumeRole 或 GetFederationToken）获得临时安全凭证。

Amazon ECR 支持使用临时凭证。

服务相关角色

服务相关角色允许 Amazon 服务访问其它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，并归该服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。

Amazon ECR 支持服务相关角色。有关更多信息，请参阅 对 Amazon ECR 使用服务相关角色 (p. 86)。

适用于 Amazon Elastic Container Registry 的 Amazon 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 Amazon EC2 实例。借助这些策略，您可对 Amazon ECR 资源和 API 操作的访问权限进行不同级别的控制。您可以直接应用这些策略，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 参考中的操作。

主题

AmazonEC2ContainerRegistryFullAccess

您可以将 `AmazonEC2ContainerRegistryFullAccess` 策略附加得到 IAM 身份。

您可以使用此托管策略作为起点，根据您的具体要求创建自己的 IAM policy。例如，您可以创建一个策略，专门为 IAM 用户或角色提供完全管理员访问权限，以管理 Amazon ECR 的使用。借助 Amazon ECR 生命周期策略功能，您可以指定存储库中镜像的生命周期管理。生命周期策略事件以 CloudTrail 事件的形式报告。Amazon ECR 与 Amazon CloudTrail 集成，因此它可以直接在 Amazon ECR 控制台中显示您的生命周期策略事件。`AmazonEC2ContainerRegistryFullAccess` 托管 IAM policy包含促进此行为的 `cloudtrail:LookupEvents` 权限。

权限详细信息

此策略包含以下权限：

- `ecr` - 允许委托人完全访问所有 Amazon ECR API。
- `cloudtrail` - 允许委托人查找管理事件或 CloudTrail 捕获的 Amazon CloudTrail 洞察事件。

`AmazonEC2ContainerRegistryFullAccess` 策略如下所示。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:*",
                "cloudtrail:LookupEvents"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "replication.ecr.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

AmazonEC2ContainerRegistryPowerUser

您可以将 AmazonEC2ContainerRegistryPowerUser 策略附加得到 IAM 身份。

此策略授予管理权限，以允许 IAM 用户读写存储库，但不允许他们删除存储库或更改应用于存储库的策略文档。

权限详细信息

此策略包含以下权限：

- ecr - 允许委托人读取和写入存储库，以及读取生命周期策略。委托人不会被授予删除存储库或更改应用于它们的生命周期策略的权限。

AmazonEC2ContainerRegistryPowerUser 策略如下所示。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:GetRepositoryPolicy",
                "ecr:DescribeRepositories",
                "ecr:ListImages",
                "ecr:DescribeImages",
                "ecr:BatchGetImage",
                "ecr:GetLifecyclePolicy",
                "ecr:GetLifecyclePolicyPreview",
                "ecr:ListTagsForResource",
                "ecr:DescribeImageScanFindings",
                "ecr:InitiateLayerUpload",
                "ecr:UploadLayerPart",
                "ecr:CompleteLayerUpload",
                "ecr:PutImage"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

AmazonEC2ContainerRegistryReadOnly

您可以将 AmazonEC2ContainerRegistryReadOnly 策略附加得到 IAM 身份。

此策略授予 Amazon ECR 的只读权限。这包括列出存储库及其中镜像的功能。它还包括使用 Docker CLI 从 Amazon ECR 提取镜像的功能。

权限详细信息

此策略包含以下权限：

- ecr - 允许委托人读取存储库及其各自的生命周期策略。

AmazonEC2ContainerRegistryReadOnly 策略如下所示。

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

AWSECRPullThroughCache_ServiceRolePolicy

您不能将 AWSECRPullThroughCache_ServiceRolePolicy 托管的 IAM policy 附加到您的 IAM 实体。此策略附加到服务相关角色，该角色允许 Amazon ECR 通过缓存提取工作流将镜像推送到存储库。有关更多信息，请参阅对 Amazon ECR 使用服务相关角色 (p. 86)。

ECRReplicationServiceRolePolicy

您不能将 ECRReplicationServiceRolePolicy 托管的 IAM policy 附加到您的 IAM 实体。此附加到服务相关角色的策略允许 Amazon ECR 代表您执行操作。有关更多信息，请参阅 对 Amazon ECR 使用服务相关角色 (p. 86)。

Amazon 托管策略的 Amazon ECR 更新

查看有关 Amazon ECR（自从其开始跟踪更新更改以来）的 Amazon 托管策略更新的详细信息。有关此页面更改的自动提示，请订阅 Amazon ECR 文档历史记录页面上的 RSS 源。

更改	说明	日期
AWSECRPullThroughCache_ServiceRolePolicy (p. 84) – 新策略	Amazon ECR 添加了新策略。此策略与用于缓存提取功能的 `AWSServiceRoleForECRPullThroughCache` 服务相关角色相关。	2021 年 11 月 29 日
ECRReplicationServiceRolePolicy (p. 84) – 新策略	Amazon ECR 添加了新策略。此策略与用于复制功能的 `AWSServiceRoleForECRReplication` 服务相关角色相关。	2020 年 12 月 4 日
AmazonEC2ContainerRegistryFullAccess (p. 80) - 对现有策略的更新	Amazon ECR 将新权限添加到 `AmazonEC2ContainerRegistryFullAccess` 策略。这些权限允许委托人创建 Amazon ECR 服务相关角色。	2020 年 12 月 4 日
AmazonEC2ContainerRegistryReadOnly (p. 81) - 对现有策略的更新	Amazon ECR 将新权限添加到 `AmazonEC2ContainerRegistryReadOnly` 策略，该策略允许委托人读取生命周期策略、列出标签以及描述镜像的扫描结果。	2019 年 12 月 10 日
AmazonEC2ContainerRegistryPowerUser (p. 80) - 对现有策略的更新	Amazon ECR 将新权限添加到 `AmazonEC2ContainerRegistryPowerUser` 策略。这些权限允许委托人读取生命周期策略、列出标签以及描述镜像的扫描结果。	2019 年 12 月 10 日
AmazonEC2ContainerRegistryFullAccess (p. 80) - 对现有策略的更新	Amazon ECR 将新权限添加到 `AmazonEC2ContainerRegistryFullAccess` 策略。这些权限允许委托人查找管理事件或 CloudTrail 捕获的 Amazon CloudTrail 洞察事件。	2017 年 11 月 10 日
AmazonEC2ContainerRegistryReadOnly (p. 81) - 对现有策略的更新	Amazon ECR 将新权限添加到 `AmazonEC2ContainerRegistryReadOnly` 策略。这些权限允许委托人描述 Amazon ECR 镜像。	2016 年 10 月 11 日
AmazonEC2ContainerRegistryPowerUser (p. 80) - 对现有策略的更新	Amazon ECR 将新权限添加到 `AmazonEC2ContainerRegistryPowerUser` 策略。这些权限允许委托人描述 Amazon ECR 镜像。	2016 年 10 月 11 日
AmazonEC2ContainerRegistryReadOnly (p. 81) - 新策略	Amazon ECR 添加了一个新策略，用于向 Amazon ECR 授予只读权限。这些权限包括列出存储库及其中镜像的功能。它们还包括使用 Docker CLI 从 Amazon ECR 提取镜像的功能。	2015 年 12 月 21 日
AmazonEC2ContainerRegistryPowerUser (p. 80) - 新策略	Amazon ECR 添加了一项新策略，该策略授予管理权限，从而允许 IAM 用户读写存储库，但不允许他们删除存储库或更改应用于存储库的策略文档。	2015 年 12 月 21 日

更改	说明	日期
AmazonEC2ContainerRegistryFullAccess (p. 107) - 新策略	Amazon ECR 添加了新策略。此策略授予 Amazon ECR 的完全访问权限。	2015 年 12 月 21 日
Amazon ECR 开始跟踪更改	Amazon ECR 开始跟踪其 Amazon 托管策略的更改。	2021 年 6 月 24 日

对 Amazon ECR 使用服务相关角色

Amazon Elastic Container Registry (Amazon ECR) 使用 Amazon Identity and Access Management (IAM) 服务相关角色以提供使用复制和缓存提取功能所需的权限。服务相关角色是一种独特类型的 IAM 角色，它与 Amazon ECR 直接相关。服务相关角色由 Amazon ECR 预定义。它包含该服务支持私有注册表的复制和缓存提取功能所需的所有权限。为注册表配置复制或缓存提取之后，系统会自动创建服务相关角色。有关更多信息，请参阅私有注册表设置 (p. 14)。

服务相关角色可让您更轻松地通过 Amazon ECR 设置复制和缓存提取。这是因为，使用该角色，您就不必手动添加所有必要的权限。Amazon ECR 定义其服务相关角色的权限，除非另外定义，否则只有 Amazon ECR 可以代入该角色。定义的权限包括信任策略和权限策略。不能将该权限策略附加到任何其他 IAM 实体。

只有先禁用注册表上的复制或缓存提取之后，才能删除相应的服务相关角色。这可以确保您不会无意中删除 Amazon ECR 对这些功能所需的权限。

有关支持服务相关角色的其他服务的信息，请参阅与 IAM 配合使用的 Amazon 服务。在此链接页面上，查找在服务相关角色列中具有是值的服务。选择是与查看该服务的服务相关角色文档的链接。

主题

- Amazon ECR 服务相关角色支持的区域 (p. 86)
- 用于复制的 Amazon ECR 服务相关角色 (p. 86)
- 用于缓存提取的 Amazon ECR 服务相关角色 (p. 88)

Amazon ECR 服务相关角色支持的区域

Amazon ECR 在提供 Amazon ECR 服务的所有区域支持使用服务相关角色。有关 Amazon ECR 区域可用性的更多信息，请参阅 Amazon 区域和端点。

用于复制的 Amazon ECR 服务相关角色

Amazon ECR 的服务相关角色权限

Amazon ECR 使用名为 AWSServiceRoleForECRReplication 的服务相关角色 - 允许 Amazon ECR 跨多个账户复制镜像。

AWSServiceRoleForECRReplication 服务相关角色信任以下服务以担任该角色：

- `replication.ecr.amazonaws.com`

以下 `ECRReplicationServiceRolePolicy` 角色权限策略允许 Amazon ECR 对资源使用以下操作：

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:CreateRepository",
                "ecr:ReplicateImage"
            ],
            "Resource": "*"
        }
    ]
}
```

Note

`ReplicateImage` 是 Amazon ECR 用于复制的内部 API，不能直接调用。

必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅 IAM 用户指南中的服务相关角色权限。

为 Amazon ECR 创建服务相关角色

无需手动创建 Amazon ECR 服务相关角色。当您在 Amazon Web Services Management Console、Amazon CLI、或Amazon API 中为注册表配置复制设置时，Amazon ECR 将为您创建服务相关角色。

如果您删除了此服务相关角色然后需要再次创建它，则可以使用相同的流程在您的账户中重新创建此角色。当您为注册表配置复制设置时，Amazon ECR 将再次为您创建服务相关角色。

为 Amazon ECR 编辑服务相关角色

Amazon ECR 不允许您手动编辑 AWSServiceRoleForECRReplication 服务相关角色。在创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。不过，您可以使用 IAM 编辑角色的说明。有关更多信息，请参阅 IAM 用户指南中的编辑服务相关角色。

为 Amazon ECR 删除服务相关角色

如果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的注册表复制配置，才能手动删除服务相关角色。

Note

如果您尝试删除资源，而 Amazon ECR 服务仍在使用角色，则删除操作可能会失败。如果发生这种情况，请等待几分钟，然后重试。

要删除 AWSServiceRoleForECRReplication 服务相关角色所使用的 Amazon ECR 资源

1. 打开位于 https://console.aws.amazon.com/ecr/ 的 Amazon ECR 控制台。
2. 从导航栏中，选择为其设置复制配置的区域。
3. 在导航窗格中，选择注册表设置。
4. 同时选择跨区域复制和跨账户复制设置。
5. 选择保存。

使用 IAM 手动删除服务相关角色

使用 IAM 控制台、Amazon CLI 或 Amazon API 删除 AWSServiceRoleForECRReplication 服务相关角色。有关更多信息，请参见 IAM 用户指南中的删除服务相关角色。

用于缓存提取的 Amazon ECR 服务相关角色

Amazon ECR 使用名为 AWSServiceRoleForECRPullThroughCache 的服务相关角色，该角色允许 Amazon ECR 通过缓存提取工作流将镜像推送到您的存储库。

Amazon ECR 的服务相关角色权限

AWSServiceRoleForECRPullThroughCache 服务相关角色信任以下服务来代入该角色。

- `pullthroughcache.ecr.amazonaws.com`

以下 `AWSECRPullThroughCache_ServiceRolePolicy` 权限策略将附加到服务相关角色，并允许 Amazon ECR 使用以下操作。

```
{
 "Version": "2012-10-17",
 "Statement": [{
  "Effect": "Allow",
  "Action": [
   "ecr:GetAuthorizationToken",
   "ecr:BatchCheckLayerAvailability",
   "ecr:InitiateLayerUpload",
   "ecr:UploadLayerPart",
   "ecr:CompleteLayerUpload",
   "ecr:PutImage"
  ],
  "Resource": "*"
 }]
}
```

必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅 IAM 用户指南中的服务相关角色权限。

为 Amazon ECR 创建服务相关角色

无需手动为缓存提取创建 Amazon ECR 服务相关角色。当您在 Amazon Web Services Management Console、Amazon CLI 或 Amazon API 中为私有注册表创建缓存提取规则时，Amazon ECR 将为您创建服务相关角色。

如果您删除了此服务相关角色然后需要再次创建它，则可以使用相同的流程在您的账户中重新创建此角色。当您为私有注册表创建缓存提取规则时，如果服务相关角色尚不存在，则 Amazon ECR 将为您再次创建该角色。

为 Amazon ECR 编辑服务相关角色

Amazon ECR 不允许您手动编辑 AWSServiceRoleForECRPullThroughCache 服务相关角色。在创建服务相关角色后，您将无法更改角色的名称，因为可能有多种实体引用该角色。但是可以使用 IAM 编辑角色说明。有关更多信息，请参阅 IAM 用户指南中的编辑服务相关角色。

为 Amazon ECR 删除服务相关角色

如果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的注册表缓存提取规则，才能手动删除服务相关角色。

> **Note**
>
> 如果您尝试删除资源，而 Amazon ECR 服务仍在使用角色，则删除操作可能会失败。如果发生这种情况，请等待几分钟，然后重试。

要删除 AWSServiceRoleForECRPullThroughCache 服务相关角色所使用的 Amazon ECR 资源

1. 打开位于 https://console.aws.amazon.com/ecr/ 的 Amazon ECR 控制台。
2. 从导航栏中，选择创建缓存提取规则所在的区域。
3. 在导航窗格中，选择 Private registry（私有注册表）、Pull through cache（缓存提取）。
4. 选择您的缓存提取规则，然后选择 Delete rule（删除规则）。

使用 IAM 手动删除服务相关角色

使用 IAM 控制台、Amazon CLI 或 Amazon API 删除 AWSServiceRoleForECRPullThroughCache 服务相关角色。有关更多信息，请参见 IAM 用户指南中的删除服务相关角色。

跨服务混淆代理问题防范

混淆代理问题是一个安全性问题，即不具有操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在 Amazon 中，跨服务模拟可能会导致混淆代理问题。一个服务（呼叫服务）调用另一项服务（所谓的服务)时，可能会发生跨服务模拟。可以操纵调用服务，使用其权限以在其他情况下该服务不应有访问权限的方式对另一个客户的资源进行操作。为了防止这种情况，Amazon 提供可帮助您保护所有服务的服务委托人数据的工具，这些服务委托人有权限访问账户中的资源。

我们建议在资源策略中使用 `aws:SourceArn` 或 `aws:SourceAccount` 全局条件上下文键，以限制 Amazon ECR 为其他服务提供的资源访问权限。如果您只希望将一个资源与跨服务访问相关联，请使用 `aws:SourceArn`。如果您想允许该账户中的任何资源与跨服务使用操作相关联，请使用 `aws:SourceAccount`。

防范混淆代理问题最有效的方法是使用 `aws:SourceArn` 全局条件上下文键和资源的完整 ARN。如果不知道资源的完整 ARN，或者正在指定多个资源，请针对 ARN 未知部分使用带有通配符字符 (*) 的 `aws:SourceArn` 全局上下文条件键。例如：`arn:aws:`*`servicename`*`:`*`region`*`:`*`123456789012`*`:*`。

如果 `aws:SourceArn` 值不包含账户 ID，例如 Amazon S3 存储桶 ARN，您必须使用两个全局条件上下文密钥来限制权限。

`aws:SourceArn` 的值必须为 ResourceDescription。

以下示例显示如何在 Amazon ECR 存储库中使用 `aws:SourceArn` 和 `aws:SourceAccount` 全局条件上下文键来允许 Amazon CodeBuild 访问与该服务集成所需的 Amazon ECR API操作，同时防止混淆代理问题。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"CodeBuildAccess",
         "Effect":"Allow",
         "Principal":{
            "Service":"codebuild.amazonaws.com"
         },
         "Action":[
            "ecr:BatchGetImage",
            "ecr:GetDownloadUrlForLayer"
         ],
         "Condition":{
            "ArnLike":{
               "aws:SourceArn":"arn:aws:codebuild:region:123456789012:project/project-name"
            },
            "StringEquals":{
               "aws:SourceAccount":"123456789012"
            }
         }
```

```
        }
    ]
}
```

Amazon 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 Amazon Web Services Management Console、Amazon CLI 或 Amazon API 执行任务。IAM 管理员必须创建 IAM policy，以便为用户和角色授予权限以对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 IAM 用户指南 中的在 JSON 选项卡上创建策略。

主题

策略最佳实践

基于身份的策略确定某个人是否可以创建、访问或删除您账户中的 Amazon ECR 资源。这些操作可能会使 Amazon Web Services 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：

- Amazon 托管策略及转向最低权限许可入门 - 要开始向用户和工作负载授予权限，请使用 Amazon 托管策略来为许多常见使用场景授予权限。您可以在 Amazon Web Services 账户 中找到这些策略。我们建议通过定义特定于您的使用场景的 Amazon 客户管理型策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 Amazon 托管策略或工作职能的 Amazon 托管策略。
- 应用最低权限 – 在使用 IAM policy 设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM policy 中的条件进一步限制访问权限 – 您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果通过特定 Amazon Web Service（例如 Amazon CloudFormation）使用服务操作，您还可以使用条件来授予对服务操作的访问权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM policy，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言 (JSON) 和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的 IAM Acess Analyzer 策略验证。
- 需要多重身份验证 (MFA) – 如果您的账户需要 IAM 用户或根用户，请启用 MFA 来提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的配置受 MFA 保护的 API 访问。

有关 IAM 中的最佳实践的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实践。

使用 Amazon ECR 控制台

要访问 Amazon Elastic Container Registry 控制台，您必须具有一组最低的权限。这些权限必须允许您列出和查看有关您 Amazon 账户中的 Amazon ECR 资源的详细信息。如果您创建的基于身份的策略比所需的最低权限更严格，则无法为具有该策略的实体（IAM 用户或角色）正常运行控制台。

要确保这些实体仍然可以使用 Amazon ECR 控制台，请将 `AmazonEC2ContainerRegistryReadOnly` Amazon 托管策略附加到这些实体。有关更多信息，请参阅 IAM 用户指南中的为用户添加权限：

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

对于只需要调用 Amazon CLI 或 Amazon API 的用户，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

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上完成此操作或者以编程方式使用 Amazon CLI 或 Amazon 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: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

在此示例中，您想要为您的 Amazon 账户中的 IAM 用户授予访问其中一个 Amazon ECR 存储库 `my-repo` 的权限。您还希望允许用户推送、提取和列出镜像。

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"ListImagesInRepository",
         "Effect":"Allow",
         "Action":[
            "ecr:ListImages"
         ],
         "Resource":"arn:aws:ecr:us-east-1:123456789012:repository/my-repo"
      },
      {
         "Sid":"GetAuthorizationToken",
         "Effect":"Allow",
         "Action":[
            "ecr:GetAuthorizationToken"
         ],
         "Resource":"*"
      },
      {
         "Sid":"ManageRepositoryContents",
         "Effect":"Allow",
         "Action":[
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:GetRepositoryPolicy",
                "ecr:DescribeRepositories",
                "ecr:ListImages",
                "ecr:DescribeImages",
                "ecr:BatchGetImage",
                "ecr:InitiateLayerUpload",
                "ecr:UploadLayerPart",
                "ecr:CompleteLayerUpload",
                "ecr:PutImage"
         ],
         "Resource":"arn:aws:ecr:us-east-1:123456789012:repository/my-repo"
      }
   ]
}
```

使用基于标签的访问控制

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，请参阅标记私有存储库 (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 policy 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 的身份和访问权限问题

可以使用以下信息，以帮助您诊断和修复在使用 Amazon ECR 和 IAM 时可能遇到的常见问题。

主题

我无权在 Amazon ECR 中执行操作

如果 Amazon Web Services Management Console 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:iam::123456789012:user/mateojackson is not authorized to perform:
 ecr:DescribeRepositories on resource: my-repo
```

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

我无权执行 iam:PassRole

如果您收到一个错误，表明您无权执行 `iam:PassRole` 操作，则必须更新策略以允许您将角色传递给 Amazon ECR。

有些 Amazon Web Services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此操作。Mary 不具有将角色传递到服务的权限。

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

在这种情况下，必须更新 Mary 的策略以允许她执行 `iam:PassRole` 操作。

如果您需要帮助，请联系您的 Amazon 管理员。管理员是向您提供登录凭证的人。

我想要查看我的访问密钥

在创建 IAM 用户访问密钥后，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，即便是为了帮助找到您的规范用户 ID 也不行。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。如果丢失了您的秘密访问密钥，您必须为 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一个 IAM 实体（用户或角色）。它们将使用该实体的凭证访问 Amazon。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 委派用户和组。

我希望允许我的 Amazon 账户以外的人访问我的 Amazon ECR 资源

您可以创建一个角色，以便其它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 结合使用 (p. 79)。
- 要了解如何为您拥有的 Amazon Web Services 账户 中的资源提供访问权限，请参阅 IAM 用户指南中的为您拥有的另一个 Amazon Web Services 账户 中的 IAM 用户提供访问权限。
- 要了解如何为第三方Amazon Web Services 账户提供您的资源的访问权限，请参阅 IAM 用户指南中的为第三方拥有的Amazon Web Services 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 用户指南中的为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 角色与基于资源的策略有何不同。

Amazon ECR 中的数据保护

Amazon 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，Amazon 负责保护运行所有 Amazon Web Services 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 Amazon Web Services 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 Amazon Web Services 账户 凭证并使用 Amazon 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 Amazon 资源进行通信。建议使用 TLS 1.2 或更高版本。
- 使用 Amazon CloudTrail 设置 API 和用户活动日志记录。
- 使用 Amazon 加密解决方案以及 Amazon 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 Simple Storage Service（Amazon S3）中的个人数据。
- 如果在通过命令行界面或 API 访问 Amazon 时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准 (FIPS) 第 140-2 版》。

我们强烈建议您切勿将机密信息或敏感信息（例如您客户的电子邮件地址）放入标签或自由格式字段（例如名称字段）。这包括当您通过控制台、API、Amazon CLI 或 Amazon SDK 使用 Amazon ECS 或其他 Amazon 服务时。您在用于名称的标签或自由格式字段中输入的任何数据都可能会用于计费或诊断日志。当您向外部服务器提供 URL 时，强烈建议您不要在 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 将镜像存储在 Amazon ECR 管理的 Amazon S3 存储桶中。默认情况下，Amazon ECR 使用具有 Amazon S3 托管加密密钥的服务器端加密，从而使用 AES-256 加密算法对静态数据进行加密。这不需要您采取任何行动，且不会另外收取费用。有关更多信息，请参阅 Amazon Simple Storage Service 用户指南中的借助使用 Amazon S3 托管式加密密钥的服务器端加密 (SSE-S3) 保护数据。

要更完善地控制 Amazon ECR 存储库的加密，您可以将服务器端加密与存储在 Amazon Key Management Service (Amazon KMS) 中的 KMS 密钥结合使用。使用 Amazon KMS 加密数据时，您可以使用原定设置的 Amazon 托管式密钥（由 Amazon ECR 管理），或者指定自己的 KMS 密钥（称为客户管理的密钥）。有关更多信息，请参阅 Amazon Simple Storage Service 用户指南中的借助使用在 Amazon KMS (SSE-KMS) 中存储 KMS 密钥的服务器端加密保护数据。

每个 Amazon ECR 存储库都有一个加密配置，该配置在创建存储库时进行设置。您可以在每个存储库上使用不同的加密配置。有关更多信息，请参阅 创建私有存储库 (p. 20)。

创建启用 Amazon KMS 加密的存储库时，KMS 密钥用于加密存储库的内容。此外，Amazon ECR 添加 KMS 密钥的 Amazon KMS 授权，其中 Amazon ECR 存储库作为被授权委托人。

以下内容提供了对 Amazon ECR 如何与 Amazon KMS 集成以加密和解密存储库的高层级理解：

1. 创建存储库时，Amazon ECR 会发送 DescribeKey 调用给 Amazon KMS，以验证和检索加密配置中指定的 KMS 密钥的 Amazon Resource Name (ARN)。
2. Amazon ECR 发送两个 CreateGrant 请求给 Amazon KMS，以在 KMS 密钥上创建授权，从而允许 Amazon ECR 使用数据密钥加密和解密数据。
3. 推送镜像时，GenerateDataKey 请求会发送到 Amazon KMS，其指定用于加密镜像层和清单的 KMS 密钥。
4. Amazon KMS 生成一个新的数据密钥，使用指定的 KMS 密钥对其进行加密，并发送加密的数据密钥，以便与镜像层元数据和镜像清单一起存储。
5. 推送镜像时，Decrypt 请求会发送到 Amazon KMS，同时指定加密的数据密钥。
6. Amazon KMS 解密加密的数据密钥，然后将解密的数据密钥发送到 Amazon S3。
7. 数据密钥用于在提取镜像层之前对其进行解密。
8. 删除存储库时，Amazon ECR 会发送两个 RetireGrant 请求给 Amazon KMS，以停用为存储库创建的授权。

注意事项

将 Amazon KMS 加密与 Amazon ECR 结合使用时，应考虑以下要点。

- 如果您使用 KMS 加密创建 Amazon ECR 存储库，并且未指定 KMS 密钥，则 Amazon ECR 预设情况下将使用带有别名 `aws/ecr` 的 Amazon 托管式密钥。首次创建启用 KMS 加密的存储库时，在您的账户中创建此 KMS 密钥。
- 当您将 KMS 加密与自己的 KMS 密钥结合使用时，密钥必须与您的存储库位于同一个区域中。
- Amazon ECR 代表您创建的授权不应被撤销。如果您撤销授予 Amazon ECR 使用账户中 Amazon KMS 密钥的授权，则 Amazon ECR 无法访问此数据、加密推送到存储库的新镜像，也无法在提取这些镜像时对其进行解密。当您撤销 Amazon ECR 授权时，更改将立即生效。要撤销访问权限，则应删除存储库，而不是撤销该授权。删除存储库后，Amazon ECR 会代表您停用授权。
- 使用 Amazon KMS 密钥会产生相应费用。有关更多信息，请参阅 Amazon Key Management Service 定价。

所需的 IAM 权限

创建或删除使用 Amazon KMS 进行服务器端加密的 Amazon ECR 存储库时，所需的权限取决于您正在使用的特定 KMS 密钥。

使用 Amazon ECR 的 Amazon 托管式密钥 时所需的 IAM 权限

预设情况下，已为 Amazon ECR 存储库启用 Amazon KMS 加密，但未指定 KMS 密钥，而是使用 Amazon ECR 的 Amazon 托管式密钥。当使用 Amazon ECR 的 Amazon 托管 KMS 密钥加密存储库时，任何有权创建存储库的委托人也可以启用 Amazon KMS 加密。但是，删除存储库的 IAM 委托人必须具有 `kms:RetireGrant` 权限。这可在停用在创建存储库中已添加到 Amazon KMS 密钥的授权。

以下示例 IAM policy 可作为内联策略添加到用户，以确保用户具有删除启用加密的存储库所需的最低权限。可以使用资源参数指定用于加密存储库的 KMS 密钥。

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

使用客户托管密钥时所需的 IAM 权限

在创建使用客户托管密钥启用 Amazon KMS 加密的存储库时，对于创建存储库的用户或角色，必须具有 KMS 密钥策略和 IAM policy 的权限。

创建自己的 KMS 密钥时，您可以使用原定设置密钥策略 Amazon KMS 创建，也可以指定自己的密钥策略。为确保客户托管密钥仍然可由账户拥有者管理，KMS 密钥的密钥策略应允许账户根用户的所有 Amazon KMS 操作。可以向密钥策略添加额外的作用域权限，但至少应向根用户授予管理 KMS 密钥的权限。要仅允许将 KMS 密钥用于源自 Amazon ECR 的请求，可将 kms:ViaService 条件密钥与 `ecr.<region>.amazonaws.com` 值结合使用。

以下示例键策略给出拥有 KMS 密钥完全访问权限的 Amazon 账户（根用户）。有关此示例密钥策略更多信息，请参阅Amazon Key Management Service 开发人员指南中的允许访问 Amazon 账户并启用 IAM policy。

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

```
}
```

创建存储库的 IAM 用户、IAM 角色或 Amazon 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 policy 中。`kms:CreateGrant` 和 `kms:DescribeKey` 权限可以添加到 KMS 密钥的密钥策略或创建存储库的用户或角色的 IAM policy 中。有关 Amazon KMS 权限如何工作的更多信息，请参阅Amazon Key Management Service 开发人员指南中的 Amazon KMS API 权限：操作和资源参考。

以下示例 IAM policy 可作为内联策略添加到用户，以确保用户拥有创建启用加密的存储库所需的最低权限，并在完成存储库时删除存储库。可以使用资源参数指定用于加密存储库的 Amazon KMS key。

```
{
    "Version": "2012-10-17",
    "Id": "ecr-kms-permissions",
    "Statement": [
        {
            "Sid":
 "AllowAccessToCreateAndRetireTheGrantsAssociatedWithTheKeyAsWellAsDescribeTheKey",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:RetireGrant",
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:us-
west-2:111122223333:key/b8d9ae76-080c-4043-92EXAMPLE"
        }
    ]
}
```

创建存储库时，允许用户在控制台中列出 KMS 密钥

使用 Amazon ECR 控制台创建存储库时，您可以授予权限，允许用户在启用存储库加密时于区域中列出客户托管的 KMS 密钥。以下 IAM policy 示例显示了使用控制台时列出 KMS 密钥和别名所需的权限。

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:ListKeys",
      "kms:ListAliases",
      "kms:DescribeKey"
    ],
    "Resource": "*"
  }
}
```

监控 Amazon ECR 与 Amazon KMS 的集成

您可以使用 Amazon CloudTrail 跟踪 Amazon ECR 代表您向 Amazon KMS 发送的请求。CloudTrail 日志中的日志条目包含加密上下文密钥，以便更容易识别它们。

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，Amazon 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。

在其发送给 Amazon KMS 的 GenerateDataKey 和 Decrypt 请求中，Amazon ECR 将使用具有两个名称–值对的加密上下文，这些名称–值对用于标识正在使用的存储库和 Amazon S3 存储桶。如下例所示。名称不会变化，但与其组合的加密上下文会因每个值而异。

```
"encryptionContext": {
    "aws:s3:arn": "arn:aws:s3:::us-west-2-starport-manifest-bucket/EXAMPLE1-90ab-cdef-fedc-ba987BUCKET1/sha256:a7766145a775d39e53a713c75b6fd6d318740e70327aaa3ed5d09e0ef33fc3df",
    "aws:ecr:arn": "arn:aws:ecr:us-west-2:111122223333:repository/repository-name"
}
```

您可以使用加密上下文在审计记录和日志中标识这些加密操作（例如 Amazon 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。

- aws:s3:arn – 第一个名称 - 值对标识存储桶。密钥是 `aws:s3:arn`。值是 Amazon S3 存储桶的 Amazon Resource Name (ARN)。

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

  例如，如果存储桶的 ARN 是 `arn:aws:s3:::us-west-2-starport-manifest-bucket/EXAMPLE1-90ab-cdef-fedc-ba987BUCKET1/sha256:a7766145a775d39e53a713c75b6fd6d318740e70327aaa3ed5d09e0ef33fc3df`，加密上下文将包括以下对。

  ```
  "arn:aws:s3:::us-west-2-starport-manifest-bucket/EXAMPLE1-90ab-cdef-fedc-ba987BUCKET1/sha256:a7766145a775d39e53a713c75b6fd6d318740e70327aaa3ed5d09e0ef33fc3df"
  ```

- aws:ecr:arn – 第二个名称 - 值对标识存储库的 Amazon Resource Name (ARN)。密钥是 `aws:ecr:arn`。值是存储库的 ARN。

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

  例如，如果储存库的 ARN 是 `arn:aws:ecr:us-west-2:111122223333:repository/repository-name`，加密上下文将包括以下对。

  ```
  "aws:ecr:arn": "arn:aws:ecr:us-west-2:111122223333:repository/repository-name"
  ```

问题排查

使用控制台删除 Amazon ECR 存储库时，如果存储库已成功删除，但 Amazon ECR 无法停用添加到存储库 KMS 密钥的授权，您将收到以下错误消息。

```
The repository [{repository-name}] has been deleted successfully but the grants created by
 the kmsKey [{kms_key}] failed to be retired
```

出现此问题时，您可以亲自停用存储库的 Amazon KMS 授权。

要手动停用存储库的 Amazon KMS 授权

1. 列出用于存储库的 Amazon KMS 密钥的授权。`key-id` 值包含在您从控制台收到的错误中。您也可以使用 `list-keys` 命令以列出 Amazon 托管式密钥 和账户中特定区域内的客户托管 KMS 密钥。

   ```
   aws kms list-grants \
   ```

```
    --key-id b8d9ae76-080c-4043-9237-c815bfc21dfc
    --region us-west-2
```

输出包括 `EncryptionContextSubset` 以及存储库的 Amazon Resource Name (ARN)。这可用于确定添加到密钥中的哪个授权是您想要停用的授权。`GrantId` 值将在下一步中停用授权时使用。

2. 停用添加到存储库中的 Amazon KMS 密钥的每个授权。将 *GrantId* 的值替换为上一步的输出中的授权 ID。

```
aws kms retire-grant \
    --key-id b8d9ae76-080c-4043-9237-c815bfc21dfc \
    --grant-id GrantId \
    --region us-west-2
```

Amazon Elastic Container Registry 的合规性验证

作为多个 Amazon 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 Amazon 服务列表，请参阅合规性计划范围内的 Amazon 服务。有关常规信息，请参阅Amazon合规性计划。

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

您在使用 Amazon ECR 时的合规性责任取决于您数据的敏感度、贵公司的合规性目标以及适用的法律法规。Amazon 提供以下资源来帮助您满足合规性：

- 安全性与合规性 Quick Start 指南安全性与合规性 Quick Start 指南 - 这些部署指南讨论了架构注意事项，并提供了在 Amazon 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Amazon创建符合 HIPAA 标准的应用程序。
- Amazon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。
- Amazon Config 开发人员指南中的使用规则评估资源 – 此 Amazon 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。
- Amazon Security Hub – 此 Amazon 服务提供了 Amazon 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 Registry 中的基础设施安全性

作为一项托管式服务，Amazon Elastic Container Registry 由 Amazon Web Services：安全流程概览白皮书中所述的 Amazon 全球网络安全流程提供保护。

您可以使用 Amazon 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 Amazon Security Token Service (Amazon STS) 生成临时安全凭证来对请求进行签名。

您可以从任何网络位置调用这些 API 操作，但 Amazon ECR 不支持基于资源的访问策略，其中可以包含基于源 IP 地址的限制。您还可以使用 Amazon ECR 策略来控制来自特定 Amazon Virtual Private Cloud (Amazon VPC) 终端节点或特定 VPC 的访问。事实上，这隔离了在 Amazon 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 (Amazon PrivateLink) (p. 101)。

Amazon ECR 接口 VPC 终端节点 (Amazon 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 Amazon PrivateLink 提供支持，您可以使用该技术通过私有 IP 地址私下访问 Amazon ECR API。AmazonPrivateLink 将 VPC 和 Amazon ECR 之间的所有网络流量限制在 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 Amazon PrivateLink 和 VPC 终端节点的更多信息，请参阅 Amazon VPC 用户指南中的接口 VPC 终端节点。

Amazon ECR VPC 终端节点的注意事项

在为 Amazon ECR 配置 VPC 终端节点之前，请注意以下事项：

- 要允许 Amazon EC2 实例上托管的 Amazon ECR 任务从 Amazon ECR 中提取私有镜像，请确保您也为 Amazon ECS 创建接口 VPC 终端节点。有关更多信息，请参阅 Amazon Elastic Container Service 开发人员指南中的接口 VPC 终端节点 (AmazonPrivateLink)。

  Important

  在 Fargate 上托管的 Amazon ECS 任务不需要 Amazon ECS 接口 VPC 终端节点。
- 使用 Linux 平台版本 `1.3.0` 或更早版本在 Fargate 上托管的 Amazon ECS 任务只需要 com.amazonaws.***region***.ecr.dkr Amazon ECR VPC 终端节点和 Amazon S3 网关终端节点，即可利用此功能。
- 使用 Linux 平台版本 `1.4.0` 或更早版本在 Fargate 上托管的 Amazon ECS 任务需要 com.amazonaws.***region***.ecr.dkr 和 com.amazonaws.***region***.ecr.api Amazon ECR VPC 端点以及 Amazon S3 网关端点，才能利用此功能。
- 使用 Windows 平台版本 `1.0.0` 或更早版本在 Fargate 上托管的 Amazon ECS 任务需要 com.amazonaws.***region***.ecr.dkr 和 com.amazonaws.***region***.ecr.api Amazon ECR VPC 端点以及 Amazon S3 网关端点，才能利用此功能。
- 从 Amazon ECR 提取容器镜像的 Amazon ECS 任务（在 Fargate 上托管）可以通过向其任务的任务执行 IAM 角色添加条件键，限制对其任务使用的特定 VPC 和服务使用的 VPC 终端节点的访问。有关更多信息，请参阅 Amazon Elastic Container Service 开发人员指南中的用于通过接口终端节点提取 Amazon ECR 镜像的 Fargate 任务的可选 IAM 权限。
- 如果从 Amazon ECR 提取容器镜像的 Amazon ECS 任务（在 Fargate 上托管）也使用 `awslogs` 日志驱动程序将日志信息发送到 CloudWatch Logs，则需要 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. 104)。
- 附加到 VPC 端点的安全组必须允许端口 443 上来自 VPC 的私有子网的传入连接。
- VPC 终端节点当前不支持跨区域请求。确保在计划向 Amazon ECR 发出 API 调用的同一区域中创建 VPC 终端节点。
- VPC 端点目前不支持 Amazon ECR 公有存储库。考虑使用缓存提取规则将公有映像托管在与 VPC 端点位于同一区域的私有存储库中。有关更多信息，请参阅使用缓存提取规则 (p. 37)。
- VPC 终端节点仅支持 Amazon 通过 Amazon Route 53 提供的 DNS。如果您希望使用自己的 DNS，可以使用条件 DNS 转发。有关更多信息，请参阅 Amazon VPC 用户指南中的 DHCP 选项集。
- 如果您的容器具有与 Amazon S3 的现有连接，则在添加 Amazon S3 网关终端节点时，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。
- 首次使用缓存提取规则提取镜像时，如果您已将 Amazon ECR 配置为通过 Amazon PrivateLink 使用接口 VPC 终端节点，您需要在同一个 VPC 中创建一个带有 NAT 网关的公有子网，然后将至互联网的所有出站

流量从私有子网路由到 NAT 网关，才能使提取操作正常工作。后续的镜像提取不需要此操作。有关更多信息，请参阅 Amazon Virtual Private Cloud 用户指南中的场景：从私有子网访问互联网。

Windows 镜像注意事项

基于 Windows 操作系统的镜像包括受许可证限制而无法分发的构件。预设情况下，当您将 Windows 镜像推送到 Amazon ECR 存储库时，包含这些构件的层不会被推送，因为它们被视为外部层. 当构件由 Microsoft 提供时，将从 Microsoft Azure 基础设施中检索外部层。因此，要使容器能够从 Azure 中提取这些外部层，除了创建 VPC 终端节点之外，还需要执行其他步骤。

当您将 Windows 镜像推送到 Amazon ECR 时，可以通过使用 Docker 守护进程中的 `--allow-nondistributable-artifacts` 标记覆盖此行为。启用后，此标记会将许可层推送到 Amazon ECR，从而使这些镜像可以通过 VPC 终端节点从 Amazon ECR 提取，而无需额外访问 Azure。

> **Important**
>
> 使用 `--allow-nondistributable-artifacts` 标记不会排除您遵守 Windows 容器基础镜像许可证条款的义务；您不能发布 Windows 内容以供公共或第三方重新分发。允许在您自己的环境中使用。

要在 Docker 安装中启用此标记，您必须修改 Docker 守护进程配置文件，根据您的 Docker 安装，该配置文件通常可以在 Docker Engine 部分下的设置或首选项菜单中配置，也可以通过直接编辑 `C:\ProgramData\docker\config\daemon.json` 文件配置。

以下是所需配置的示例。将值替换为要将镜像推送到其中的存储库 URI。

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

修改 Docker 守护进程配置文件后，您必须在尝试推送镜像之前重新启动 Docker 守护进程。通过验证基础层是否推送到您的存储库，确认推送工作运转正常。

> **Note**
>
> Windows 镜像的基础层很庞大。层太大将导致延长推送时间，并在 Amazon ECR 中增加这些镜像的存储成本。出于这些原因，我们建议仅在严格要求此选项以减少构建时间和持续存储成本时使用此选项。例如，在 Amazon ECR 中压缩之后，`mcr.microsoft.com/windows/servercore` 镜像的大小约为 1.7 GiB。

为 Amazon ECR 创建 VPC 终端节点

要为 Amazon ECR 服务创建 VPC 终端节点，请使用 Amazon VPC 用户指南中的创建接口终端节点过程。

在 Amazon EC2 实例上托管的 Amazon ECS 任务需要 Amazon ECR 终端节点和 Amazon S3 网关终端节点。

使用平台版本 `1.4.0` 或更高版本在 Fargate 上托管的 Amazon ECS 任务需要 Amazon ECR VPC 终端节点和 Amazon S3 网关终端节点。

使用平台版本 `1.3.0` 或更早版本在 Fargate 上托管的 Amazon ECS 任务只需要 cn.com.amazonaws.cn-northwest-1.ecr.dkr Amazon ECR VPC 终端节点和 Amazon S3 网关终端节点。

> **Note**
>
> 创建终端节点的顺序无关紧要。

cn.com.amazonaws.cn-northwest-1.ecr.dkr

此终端节点用于 Docker Registry API。诸如 `push` 和 `pull` 这样的 Docker 客户端命令使用此终端节点。

在创建此终端节点时，您必须启用私有 DNS 主机名。为此，请确保在创建 VPC 终端节点时，在 Amazon VPC 控制台中选择启用私有 DNS 名称选项。

cn.com.amazonaws.cn-northwest-1.ecr.api

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

在创建此终端节点时，您可以选择启用私有 DNS 主机名。在创建 VPC 终端节点时，通过在 VPC 控制台中选择启用私有 DNS 名称，可启用此设置。如果您为 VPC 终端节点启用私有 DNS 主机名，请将开发工具包或 Amazon CLI 更新到最新版本，以便在使用开发工具包或 Amazon 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 Amazon 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 ECR 提取私有镜像的 Amazon ECS 任务，您必须为 Amazon S3 创建网关终端节点。必须创建网关终端节点，因为 Amazon ECR 使用 Amazon S3 来存储您的镜像层。当容器从 Amazon ECR 下载镜像时，它们必须访问 Amazon ECR 才能获取镜像清单，然后 Amazon S3 才能下载实际镜像层。以下是包含每个 Docker 镜像层的 Amazon S3 存储桶的 Amazon Resource Name (ARN)。

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

使用 Amazon VPC 用户指南中的创建网关终端节点过程为 Amazon ECR 创建以下 Amazon S3 网关终端节点。在创建终端节点时，请务必为您的 VPC 选择路由表。

com.amazonaws.*region*.s3

Amazon S3 网关终端节点使用 IAM policy 文档来限制对服务的访问。可以使用完全访问权限策略，因为您在任务 IAM 角色或其他 IAM 用户策略中设置的任何限制仍然适用于此策略。如果要将 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 ECR 的 Amazon S3 存储桶最低权限 (p. 103)。

Amazon ECR 的 Amazon S3 存储桶最低权限

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

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

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

示例

以下示例说明如何提供对 Amazon ECR 操作所需的 Amazon S3 存储桶的访问权限。

```
{
  "Statement": [
    {
      "Sid": "Access-to-specific-bucket-only",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::prod-region-starport-layer-bucket/*"]
    }
  ]
}
```

创建 CloudWatch Logs 终端节点

使用 Fargate 启动类型的 Amazon ECS 任务，这些任务使用没有互联网网关的 VPC，同时使用 awslogs 日志驱动程序将日志信息发送到 CloudWatch Logs，需要您为 CloudWatch Logs 创建 com.amazonaws.***region***.logs 接口 VPC 终端节点。有关更多信息，请参阅 Amazon CloudWatch Logs 用户指南中的将 CloudWatch Logs 与接口 VPC 终端节点结合使用。

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，您在创建或修改端点时可将它附加到端点。如果您在创建终端节点时未附加策略，Amazon 会为您附加一个默认策略，该策略允许对服务的完全访问。端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 Amazon ECR 的终端节点策略示例。此策略允许特定 IAM 角色从 Amazon ECR 中提取镜像。

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

```
}
```

下面的终端节点策略示例阻止删除指定的存储库。

```
{
 "Statement": [{
   "Sid": "AllowAll",
   "Principal": "*",
   "Action": "*",
   "Effect": "Allow",
   "Resource": "*"
  },
  {
   "Sid": "PreventDelete",
   "Principal": "*",
   "Action": "ecr:DeleteRepository",
   "Effect": "Deny",
   "Resource": "arn:aws:ecr:region:1234567890:repository/repository_name"
  }
 ]
}
```

下面的终端节点策略示例将前面的两个示例组合到一个策略中。

```
{
 "Statement": [{
   "Sid": "AllowAll",
   "Effect": "Allow",
   "Principal": "*",
   "Action": "*",
   "Resource": "*"
  },
  {
   "Sid": "PreventDelete",
   "Effect": "Deny",
   "Principal": "*",
   "Action": "ecr:DeleteRepository",
   "Resource": "arn:aws:ecr:region:1234567890:repository/repository_name"
  },
  {
   "Sid": "AllowPull",
   "Effect": "Allow",
   "Principal": {
    "AWS": "arn:aws:iam::1234567890:role/role_name"
   },
   "Action": [
    "ecr:BatchGetImage",
    "ecr:GetDownloadUrlForLayer",
                          "ecr:GetAuthorizationToken"
   ],
   "Resource": "*"
  }
 ]
}
```

修改 Amazon ECR 的 VPC 终端节点策略

1. 通过以下网址打开 Amazon VPC 控制台：https://console.aws.amazon.com/vpc/。
2. 在导航窗格中，选择 Endpoints（端点）。
3. 如果您没有为 Amazon ECR 创建 VPC 终端节点，请参阅 为 Amazon ECR 创建 VPC 终端节点 (p. 102)。

4. 选择要将策略添加到的 Amazon ECR VPC 终端节点，然后在屏幕的下半部分中选择策略选项卡。
5. 选择编辑策略并对策略进行更改。
6. 选择保存以保存策略。

Amazon ECR 监控

您可以使用 Amazon CloudWatch 监控您的 Amazon ECR API，此工具可从 Amazon ECR 收集原始数据，并将数据处理为易读的近乎实时的指标。这些统计数据会保存两周，以便您访问历史信息并更好地了解 API 用量。Amazon ECR 指标数据会在一分钟时段内自动发送到 CloudWatch。有关 CloudWatch 的更多信息，请参阅 Amazon CloudWatch 用户指南。

Amazon ECR 基于您的 API 用量提供指标，这些指标适用于授权、镜像推送和镜像提取操作。

监控是保持 Amazon ECR 和您的 Amazon 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 Amazon 解决方案的资源中收集监控数据，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：

- 监控目的是什么？
- 您将监控哪些资源？
- 监控这些资源的频率如何？
- 您将使用哪些监控工具？
- 谁负责执行监控任务？
- 出现错误时应通知谁？

下一步，通过在不同时间和不同负载条件下测量性能，在您的环境中建立正常 Amazon ECR 性能的基准。在监控 Amazon ECR 时，存储历史监控数据，以便将此数据与新的性能数据进行比较，确定正常性能模式和性能异常，并设计解决问题的方法。

主题

可视化 Service Quotas 并设置警报

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。还可以设置警报，从而在接近配额时向您发送通知。

可视化服务配额并选择性地设置警报

1. 访问 https://console.aws.amazon.com/cloudwatch/ 打开 CloudWatch 控制台。
2. 在导航窗格中，选择指标。
3. 在所有指标选项卡上，选择用量，然后选择按 Amazon 资源。

   这将显示服务配额用量指标的列表。

4. 选中其中一个指标旁边的复选框。

   该图表显示您该 Amazon 资源的当前用量。
5. 要将服务配额添加到图表，请执行以下操作：

   a. 选择 Graphed metrics（绘制的指标）选项卡。
   b. 选择数学表达式、从空表达式开始。然后在新行中，在详细信息下，输入 **`SERVICE_QUOTA(m1)`**。

      这将向图表中添加一个新行，并显示指标中表示的资源的服务配额。
6. 要以配额百分比的形式查看您的当前用量，请添加新的表达式或更改当前 SERVICE_QUOTA 表达式。对于新的表达式，请使用 **`m1/60/SERVICE_QUOTA(m1)*100`**。
7. (可选) 要设置一个警报，以便在接近服务配额时向您发送通知，请执行以下操作：

   a. 在 **`m1/60/SERVICE_QUOTA(m1)*100`** 行上的操作下，选择警报图标。该图标看起来像一个铃铛。

      这将显示警报创建页面。
   b. 在条件下，确保阈值类型为静态，并将当 Expression1 为设置为大于。在多于下，输入 **`80`**。这将创建一个警报，当用量超过配额的 80% 时，该警报将进入 ALARM 状态。
   c. 选择下一步。
   d. 在下一页上，选择一个 Amazon SNS 主题或创建一个新主题。当警报进入 ALARM 状态时，会向此主题发送通知。然后选择下一步。
   e. 在下一页上，输入警报的名称和描述，然后选择下一步。
   f. 选择创建警报。

Amazon ECR 用量指标

您可以使用 CloudWatch 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 Amazon 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 默认服务配额的更多信息，请参阅 Amazon ECR 服务配额 (p. 121)。

Amazon ECR 在 `Amazon/Usage` 命名空间中发布以下指标。

指标	描述
`CallCount`	从您的账户调用 API 操作的次数。资源由与指标关联的维度定义。 此指标最有用的统计数据是 `SUM`，表示定义时段内来自所有贡献者的值的总和。

以下维度用于优化由 Amazon ECR 发布的用量指标。

维度	描述
`Service`	包含该资源的 Amazon 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 会返回有关以下 API 操作的 API 用量的信息。 • GetAuthorizationToken • BatchCheckLayerAvailability • InitiateLayerUpload • UploadLayerPart • CompleteLayerUpload • PutImage • BatchGetImage • GetDownloadUrlForLayer
`Class`	所跟踪的资源的类。Amazon ECR 目前不使用类维度。

Amazon ECR 用量报告

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

使用 Cost Explorer 查看用量和成本的图表。您可以查看之前 13 个月的数据，并预测您在接下来三个月内可能产生的费用。您可以使用 Cost Explorer 查看有关您一段时间内在 Amazon 资源方面的费用的模式、确定需要进一步查询的方面以及查看可用于了解您的成本的趋势。您还可以指定数据的时间范围，并按天或按月查看时间数据。

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

有关创建 Amazon 成本和用量报告的更多信息，请参阅 Amazon Billing 用户指南中的Amazon成本和用量报告。

Amazon ECR 存储库指标

Amazon ECR 将存储库拉取计数指标发送到 Amazon CloudWatch。Amazon ECR 指标数据会以 1 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。

主题

启用 CloudWatch 指标

Amazon ECR 自动发送所有存储库的存储库指标。无需执行任何手动步骤。

可用指标和维度

以下部分列出了 Amazon ECR 发送到 Amazon CloudWatch 的指标和维度。

Amazon ECR 指标

Amazon ECR 提供了可用于监控存储库的指标。您可以测量拉取计数。

`AWS/ECR` 命名空间包括以下指标。

`RepositoryPullCount`

对存储库中映像的总拉取次数。

有效维度：`RepositoryName`。

有效统计数据：平均值、最小值、最大值、总计和样本数。最有用的统计指标是和。

单位：整数。

Amazon ECR 指标的维度

Amazon ECR 指标使用 `AWS/ECR` 命名空间并提供以下维度的指标。

`RepositoryName`

此维度将筛选您为指定存储库中所有容器映像请求的数据。

查看 Amazon ECR 指标

您可以通过 CloudWatch 控制台查看 Amazon ECR 存储库指标。CloudWatch 控制台提供了精细且可自定义的资源显示。

使用 CloudWatch 控制台查看 Amazon ECR 指标

Amazon ECR 存储库指标可通过 CloudWatch 控制台查看。该控制台提供了有关 Amazon ECR 指标的详细视图，您可以根据自己的需求定制视图。有关更多信息，请参阅 Amazon CloudWatch 用户指南。

在 CloudWatch 控制台中查看指标

1. 通过以下网址打开 CloudWatch 控制台：https://console.aws.amazon.com/cloudwatch/。
2. 在左侧导航中，依次选择 Metrics（指标）、All metrics（所有指标）。
3. 在 Browse（浏览）选项卡的 Amazon Namespaces（Amazon 命名空间）下，选择 ECR。
4. 选择要查看的指标。存储库指标的范围为 ECR > Repository Metrics（ECR > 存储库指标）。

Amazon ECR 事件和 EventBridge

您可以使用 Amazon EventBridge 自动执行您的 Amazon 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。Amazon 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化操作。可自动触发的操作包括：

- 将事件添加到 CloudWatch Logs 中的日志组
- 调用 Amazon Lambda 函数
- 调用 Amazon EC2 Run Command
- 将事件中继到 Amazon Kinesis Data Streams

- 激活 Amazon Step Functions 状态机
- 通知 Amazon SNS 主题或 Amazon SQS 队列

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

来自 Amazon ECR 的示例事件

以下是来自 Amazon ECR 的示例事件。尽最大努力发出事件。

已完成镜像推送的事件

每个镜像推送完成后，将发送以下事件。有关更多信息，请参阅 推送 Docker 镜像 (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-repository-name",
        "image-digest":
 "sha256:7f5b2640fe6fb4f46592dfd3410c4a79dac4f89e4782432e0378abcd1234",
        "action-type": "PUSH",
        "image-tag": "latest"
    }
}
```

已完成镜像扫描的事件（基本扫描）

为注册表启用基本扫描后，当每个镜像扫描完成时，会发送以下事件。`finding-severity-counts` 参数仅返回严重性级别的值 (如果存在)。例如，如果镜像不包含任何 `CRITICAL` 级别的结果，则不会返回任何关键计数。有关更多信息，请参阅基本扫描 (p. 67)。

> **Note**
>
> 有关启用增强扫描后 Amazon Inspector 发出的事件的详细信息，请参阅 EventBridge 事件 (p. 63)。

```
{
    "version": "0",
    "id": "85fc3613-e913-7fc4-a80c-a3753e4aa9ae",
    "detail-type": "ECR Image Scan",
    "source": "aws.ecr",
    "account": "123456789012",
    "time": "2019-10-29T02:36:48Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ecr:us-east-1:123456789012:repository/my-repository-name"
    ],
    "detail": {
        "scan-status": "COMPLETE",
        "repository-name": "my-repository-name",
        "finding-severity-counts": {
        "CRITICAL": 10,
        "MEDIUM": 9
```

```
      },
        "image-digest":
 "sha256:7f5b2640fe6fb4f46592dfd3410c4a79dac4f89e4782432e0378abcd1234",
        "image-tags": []
    }
}
```

启用了增强扫描的资源的变更通知事件（增强扫描）

为注册表启用增强扫描后，当启用了增强扫描的资源发生更改时，Amazon ECR 将发送以下事件。这包括正在创建的新存储库、正在更改的存储库的扫描频率，或者在启用了增强扫描功能的存储库中创建或删除镜像的时间。有关更多信息，请参阅镜像扫描 (p. 60)。

```
{
 "version": "0",
 "id": "0c18352a-a4d4-6853-ef53-0ab8638973bf",
 "detail-type": "ECR Scan Resource Change",
 "source": "aws.ecr",
 "account": "123456789012",
 "time": "2021-10-14T20:53:46Z",
 "region": "us-east-1",
 "resources": [],
 "detail": {
  "action-type": "SCAN_FREQUENCY_CHANGE",
  "repositories": [{
    "repository-name": "repository-1",
    "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-1",
    "scan-frequency": "SCAN_ON_PUSH",
    "previous-scan-frequency": "MANUAL"
   },
   {
    "repository-name": "repository-2",
    "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-2",
    "scan-frequency": "CONTINUOUS_SCAN",
    "previous-scan-frequency": "SCAN_ON_PUSH"
   },
   {
    "repository-name": "repository-3",
    "repository-arn": "arn:aws:ecr:us-east-1:123456789012:repository/repository-3",
    "scan-frequency": "CONTINUOUS_SCAN",
    "previous-scan-frequency": "SCAN_ON_PUSH"
   }
  ],
  "resource-type": "REPOSITORY",
  "scan-type": "ENHANCED"
 }
}
```

镜像删除的事件

删除镜像时将发送以下事件。有关更多信息，请参阅 删除镜像 (p. 41)。

```
{
    "version": "0",
    "id": "dd3b46cb-2c74-f49e-393b-28286b67279d",
    "detail-type": "ECR Image Action",
    "source": "aws.ecr",
    "account": "123456789012",
    "time": "2019-11-16T02:01:05Z",
    "region": "us-west-2",
    "resources": [],
    "detail": {
        "result": "SUCCESS",
```

```
        "repository-name": "my-repository-name",
        "image-digest":
 "sha256:7f5b2640fe6fb4f46592dfd3410c4a79dac4f89e4782432e0378abcd1234",
        "action-type": "DELETE",
        "image-tag": "latest"
    }
}
```

使用 Amazon CloudTrail 记录 Amazon ECR 操作

Amazon ECR 与 Amazon CloudTrail 集成，后者是一项服务，提供 Amazon ECR 中由用户、角色或 Amazon 服务所采取操作的记录。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采取的所有操作
- 由于生命周期策略规则而采取的所有操作，包括成功和不成功的操作

> **Important**
>
> 由于单个 CloudTrail 事件的大小限制原因，对于有 10 个或以上镜像过期的生命周期策略操作，Amazon ECR 会向 CloudTrail 发送多个事件。此外，Amazon ECR 中每个镜像最多可包含 100 个标签。

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 控制台的事件历史记录中查看最新事件。使用此信息，您可以确定向 Amazon ECR 发出了什么请求、发出请求的 IP 地址、何人发出的请求、请求的发出时间以及其他详细信息。

有关更多信息，请参阅 Amazon CloudTrail 用户指南。

CloudTrail 中的 Amazon ECR 信息

在您创建 Amazon 账户时，将在该账户上启用 CloudTrail。当 Amazon ECR 中发生活动时，该活动将记录在 CloudTrail 事件中，并与其他 Amazon 服务事件一同保存在事件历史记录中。您可以在 Amazon 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 Amazon 账户中的事件 (包括 Amazon ECR 的事件)，请创建跟踪记录。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 Amazon 分区中记录所有区域中的事件，并将日志文件传送至您指定的 Amazon S3 存储桶。此外，您可以配置其他 Amazon 服务，分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅：

- 为您的 Amazon 账户创建跟踪
- Amazon 服务与 CloudTrail 日志集成
- 为 CloudTrail 配置 Amazon SNS 通知
- 从多个区域接收 CloudTrail 日志文件和从多个账户中接收 CloudTrail 日志文件

所有 Amazon ECR API 操作都由 CloudTrail 记录，并记录在 Amazon Elastic Container Registry API 参考中。当执行常见任务时，CloudTrail 日志文件会针对该任务中的每个 API 操作生成相应的部分。例如，当创建存储库时，CloudTrail 日志文件中将生成 `GetAuthorizationToken`、`CreateRepository` 和 `SetRepositoryPolicy` 部分。当您将某个镜像推送到存储库中时，则将生成 `InitiateLayerUpload`、`UploadLayerPart`、`CompleteLayerUpload` 和 `PutImage` 部分。当您提取镜像时，则将生成 `GetDownloadUrlForLayer` 和 `BatchGetImage` 部分。有关这些常见任务的示例，请参阅 CloudTrail 日志条目示例 (p. 114)。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息可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
- 请求是否由其它 Amazon 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。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

CloudTrail 日志条目示例

以下是针对一些常见 Amazon ECR 任务的 CloudTrail 日志条目示例

> **Note**
>
> 为提高可读性，这些示例已进行格式化处理。在 CloudTrail 日志文件，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 Amazon 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 Amazon ECR 存储库时的 Amazon KMS CreateGrant API 操作

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了创建启用 KMS 加密的 Amazon ECR 存储库时的 Amazon KMS CreateGrant 操作。对于已启用 KMS 加密创建的每个存储库，您应该在 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：镜像推送操作

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` 操作推送镜像。

> **Note**
>
> 当推送镜像时，您在 CloudTrail 日志中还可以看到 `InitiateLayerUpload`、`UploadLayerPart` 和 `CompleteLayerUpload` 引用。

```
{
    "eventVersion": "1.04",
    "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:account_name",
    "arn": "arn:aws:sts::123456789012:user/Mary_Major",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "userName": "Mary_Major",
  "sessionContext": {
   "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2019-04-15T16:42:14Z"
   }
  }
 },
 "eventTime": "2019-04-15T16:45:00Z",
 "eventSource": "ecr.amazonaws.com",
 "eventName": "PutImage",
 "awsRegion": "us-east-2",
```

```
 "sourceIPAddress": "203.0.113.12",
 "userAgent": "console.amazonaws.com",
 "requestParameters": {
  "repositoryName": "testrepo",
  "imageTag": "latest",
  "registryId": "123456789012",
  "imageManifest": "{\n   \"schemaVersion\": 2,\n   \"mediaType\": \"application/
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：镜像提取操作

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` 操作提取镜像。

> **Note**
>
> 当提取镜像时，如果本地尚没有镜像，您在 CloudTrail 日志中还将看到 `GetDownloadUrlForLayer` 引用。

```
{
    "eventVersion": "1.04",
    "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:account_name",
```

```
    "arn": "arn:aws:sts::123456789012:user/Mary_Major",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "userName": "Mary_Major",
  "sessionContext": {
   "attributes": {
    "mfaAuthenticated": "false",
    "creationDate": "2019-04-15T16:42:14Z"
   }
  }
 },
 "eventTime": "2019-04-15T17:23:20Z",
 "eventSource": "ecr.amazonaws.com",
 "eventName": "BatchGetImage",
 "awsRegion": "us-east-2",
 "sourceIPAddress": "203.0.113.12",
 "userAgent": "console.amazonaws.com",
 "requestParameters": {
  "imageIds": [{
   "imageTag": "latest"
  }],
  "acceptedMediaTypes": [
   "application/json",
   "application/vnd.oci.image.manifest.v1+json",
   "application/vnd.oci.image.index.v1+json",
   "application/vnd.docker.distribution.manifest.v2+json",
   "application/vnd.docker.distribution.manifest.list.v2+json",
   "application/vnd.docker.distribution.manifest.v1+prettyjws"
  ],
  "repositoryName": "testrepo",
  "registryId": "123456789012"
 },
 "responseElements": null,
 "requestID": "2a1b97ee-5fa3-11e9-a8cd-cd2391aeda93",
 "eventID": "c84f5880-c2f9-4585-9757-28fa5c1065df",
 "resources": [{
  "ARN": "arn:aws:ecr:us-east-2:123456789012:repository/testrepo",
  "accountId": "123456789012"
 }],
 "eventType": "AwsApiCall",
 "recipientAccountId": "123456789012"
}
```


示例：镜像生命周期策略操作

以下示例显示了一个 CloudTrail 日志条目，该条目演示镜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。

> **Important**
>
> 由于单个 CloudTrail 事件的大小限制原因，对于有 10 个或以上镜像过期的生命周期策略操作，Amazon ECR 会向 CloudTrail 发送多个事件。此外，Amazon ECR 中每个镜像最多可包含 100 个标签。


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "accountId": "123456789012",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2020-03-12T20:22:12Z",
    "eventSource": "ecr.amazonaws.com",
    "eventName": "PolicyExecutionEvent",
    "awsRegion": "us-west-2",
```

```
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "9354dd7f-9aac-4e9d-956d-12561a4923aa",
    "readOnly": true,
    "resources": [
        {
            "ARN": "arn:aws:ecr:us-west-2:123456789012:repository/testrepo",
            "accountId": "123456789012",
            "type": "AWS::ECR::Repository"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "123456789012",
    "serviceEventDetails": {
        "repositoryName": "testrepo",
        "lifecycleEventPolicy": {
            "lifecycleEventRules": [
                {
                    "rulePriority": 1,
                    "description": "remove all images > 2",
                    "lifecycleEventSelection": {
                        "tagStatus": "Any",
                        "tagPrefixList": [],
                        "countType": "Image count more than",
                        "countNumber": 2
                    },
                    "action": "expire"
                }
            ],
            "lastEvaluatedAt": 0,
            "policyVersion": 1,
            "policyId": "ceb86829-58e7-9498-920c-aa042e33037b"
        },
        "lifecycleEventImageActions": [
            {
                "lifecycleEventImage": {
                    "digest":
 "sha256:ddba4d27a7ffc3f86dd6c2f92041af252a1f23a8e742c90e6e1297bfa1bc0c45",
                    "tagStatus": "Tagged",
                    "tagList": [
                        "alpine"
                    ],
                    "pushedAt": 1584042813000
                },
                "rulePriority": 1
            },
            {
                "lifecycleEventImage": {
                    "digest":
 "sha256:6ab380c5a5acf71c1b6660d645d2cd79cc8ce91b38e0352cbf9561e050427baf",
                    "tagStatus": "Tagged",
                    "tagList": [
                        "centos"
                    ],
                    "pushedAt": 1584042842000
                },
                "rulePriority": 1
            }
        ]
    }
}
```

Amazon ECR 服务配额

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

名称	默认值	可调整	描述
复制配置中每条规则的筛选器	每个支持的区域：100 个	否	复制配置中每条规则的筛选器的最大数量。
每个存储库的镜像数	每个支持的区域：10,000 个	是	每个存储库的最大镜像数。
层分段	每个支持的区域：4,200 个	否	层分段数上限。仅当您使用 Amazon ECR API 操作直接启动镜像推送操作的分段上传时才适用。
生命周期策略长度	每个支持的区域：30,720 个	否	生命周期策略中的最大字符数。
层分段大小上限	每个支持的区域：10 个	否	层分段的最大大小 (MiB)。仅当您使用 Amazon ECR API 操作直接启动镜像推送操作的分段上传时才适用。
层大小上限	每个支持的区域：42,000 个	否	层的最大大小 (MiB)。
层分段大小下限	每个支持的区域：5 个	否	层分段的最小大小 (MiB)。仅当您使用 Amazon ECR API 操作直接启动镜像推送操作的分段上传时才适用。
BatchCheckLayerAvailability 请求的速率	每个支持的区域：每秒 1000 个	是	您在当前区域中每秒可以发出的 BatchCheckLayerAvailability 请求的最大数量。将镜像推送到存储库时，会检查每个镜像层以验证之前是否已上传它。如果已上传，则会跳过镜像层。
BatchGetImage 请求的速率	每个支持的区域：每秒 2,000 个	是	您在当前区域中每秒可以发出的 BatchGetImage 请求的最大数量。提取镜像时，会调用 BatchGetImage API 一次以检索镜像清单。如果您请求提高此 API 的配额，则还请查看 GetDownloadUrlForLayer 的使用情况。
CompleteLayerUpload 请求的速率	每个支持的区域：每秒 100 个	是	您在当前区域中每秒可以发出的 CompleteLayerUpload 请求的最大数量。推送

名称	默认值	可调整	描述
			镜像时，对于每个新的镜像层，都会调用一次 CompleteLayerUpload API，以验证上传是否已完成。
GetAuthorizationToken 请求的速率	每个支持的区域：每秒 500 个	是	您在当前区域中每秒可以发出的 GetAuthorizationToken 请求的最大数量。
GetDownloadUrlForLayer 请求的速率	每个支持的区域：每秒 3,000 个	是	您在当前区域中每秒可以发出的 GetDownloadUrlForLayer 请求的最大数量。提取镜像时，对于每个尚未缓存的镜像层调用一次 GetDownloadUrlForLayer API。如果您请求提高此 API 的配额，则还请查看 BatchGetImage 的使用情况。
InitiateLayerUpload 请求的速率	每个支持的区域：每秒 100 个	是	您在当前区域中每秒可以发出的 InitiateLayerUpload 请求的最大数量。推送镜像时，对于每个尚未上传的镜像层，会调用一次 InitiateLayerUpload API。是否已上传镜像层由 BatchCheckLayerAvailability API 操作确定。
PutImage 请求的速率	每个支持的区域：每秒 10 个	是	您在当前区域中每秒可以发出的 PutImage 请求的最大数量。推送镜像并上传所有新的镜像层后，将调用一次 PutImage API，以创建或更新镜像清单以及与该镜像关联的标签。
UploadLayerPart 请求的速率	每个支持的区域：每秒 500 个	是	您在当前区域中每秒可以发出的 UploadLayerPart 请求的最大数量。推送映像时，每个新映像层均会分段上传，每个新映像层部分都会调用一次 UploadLayerPart API。
镜像扫描速率	每个支持的区域：1 个	否	每 24 小时每个镜像的最大镜像扫描次数。
已注册的存储库	每个支持的区域：10,000 个	是	您可以在当前区域中的此账户中创建的存储库的最大数量。

名称	默认值	可调整	描述
每个生命周期策略的规则数	每个支持的区域：50 个	否	生命周期策略中的最大规则数量
每个复制配置的规则数	每个支持的区域：10 个	否	复制配置中的最大规则数。
每个镜像的标签数	每个支持的区域：1,000 个	否	每个镜像的最大标签数。
复制配置中所有规则的唯一目标	每个支持的区域：25 个	否	复制配置中所有规则的最大唯一目标数

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。有关启用 Docker 调试的更多信息，如果您使用 Amazon ECR 在 Amazon ECS 容器实例上提取的镜像，请参阅 Amazon Elastic Container Service 开发人员指南中的启用 Docker 调试输出。

启用 Amazon CloudTrail

有关 Amazon ECR 所返回错误的其他信息，可以通过启用 Amazon CloudTrail 进行查找，它是为 Amazon 账户记录 Amazon 调用的一项服务。CloudTrail 可将日志文件传送至 Amazon S3 存储桶。通过使用 CloudTrail 收集的信息，您可以确定成功向 Amazon 服务提出的请求、谁提出请求以及何时提出等信息。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 Amazon CloudTrail 用户指南。有关将 CloudTrail 与 Amazon ECR 结合使用的更多信息，请参阅 使用 Amazon CloudTrail 记录 Amazon ECR 操作 (p. 113)。

优化 Amazon ECR 的性能

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

使用 Docker 1.10 及以上版本可利用同时层上传

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

使用较小基本镜像

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

更早将更改最少的依赖性放入您的 Dockerfile

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

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

链接命令以避免不必要文件的存储

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

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

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

```
WORKDIR /tmp
RUN wget http://example.com/software.tar.gz &&\
    wget tar -xvf software.tar.gz &&\
    mv software/binary /opt/bin/myapp &&\
    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 提取镜像的延迟。如果应用程序在 Amazon EC2 实例上运行，可以使用以下 shell 代码从实例的可用区获取区域：

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

可以使用 --region 参数将该区域传递给 Amazon CLI 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 Amazon 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 Amazon ECR 时通过 Docker 命令纠正错误

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。

主题

从 Amazon ECR 存储库提取镜像时，出现错误：“Filesystem Verification Failed”(文件系统验证失败)或“404: Image Not Found”(404：找不到镜像)

在 Docker 1.9 或更高版本中使用 docker pull 命令从 Amazon ECR 存储库提取镜像时，可能会收到错误 `Filesystem verification failed`。如果使用的是 1.9 之前的 Docker 版本，则可能收到错误 `404: Image not found`。

以下为一些可能的原因及它们的解释。

本地磁盘已满

如果运行 docker pull 命令的本地磁盘已满，那么对本地文件计算的 SHA-1 哈希值可能与 Amazon ECR 计算的 SHA-1 哈希值不同。确保本地磁盘有足够的剩余空间可存储所提取的 Docker 镜像。为腾出空间存储新镜像，可以删除旧镜像。使用 docker images 命令可查看所有已下载到本地的 Docker 镜像的列表及这些镜像的大小。

由于网络错误，客户端无法连接到远程存储库

调用 Amazon ECR 存储库需要 Internet 连接正常。验证网络设置，然后验证其他工具和应用程序是否可以访问 Internet 上的资源。如果在私有子网中对 Amazon EC2 实例运行 docker pull，请验证该子网是否具有连接至 Internet 的路由。可使用网络地址转换 (NAT) 服务器或托管的 NAT 网关。

目前，对 Amazon ECR 存储库的调用还要求通过您的公司防火墙访问 Amazon Simple Storage Service (Amazon S3)。如果贵企业或组织使用的是允许服务终端节点的防火墙软件或 NAT 设备，请确保当前区域的 Amazon S3 服务终端节点在允许范围内。

如果您通过 HTTP 代理使用 Docker，可以对 Docker 进行相应的代理设置。有关更多信息，请参阅 Docker 文档中的 HTTP 代理。

从 Amazon ECR 提取镜像时，出现错误：“Filesystem Layer Verification Failed”(文件系统分层验证失败)

您可能在使用 `image image-name not found` 命令提取镜像时收到错误 docker pull。如果检查 Docker 日志，可能会看到与下面类似的错误：

```
filesystem layer verification failed for digest sha256:2b96f...
```

此错误表示镜像的一个或多个层下载失败。以下为一些可能的原因及它们的解释。

您正在使用旧版本的 Docker

在使用低于 1.10 的 Docker 版本时，有少数情况会出现此错误。请将您的 Docker 客户端升级至 1.10 或更高版本。

您的客户端遇到网络错误或磁盘错误

如前文对 `Filesystem verification failed` 消息的讨论中所述，磁盘已满或网络问题可能会导致一个或多个层无法下载。请遵循上述建议确保您的文件系统未满，并且您在网络中有对 Amazon S3 的访问权限。

使用拉取缓存规则进行拉取时出错

使用拉取缓存规则拉取上游镜像时，您可能会收到以下常见错误。

存储库不存在

指示存储库不存在的错误通常是由该存储库不存在于 Amazon ECR 私有注册表中或者未向拉取上游镜像的 IAM 主体授予 `ecr:CreateRepository` 权限所致。要解决此错误，您应该验证拉取命令中的存储库 URI 正确，已向拉取上游镜像的 IAM 主体授予所需的 IAM 权限，或者在执行上游镜像拉取之前，已在您的 Amazon ECR 私有注册表中创建要向其推送上游镜像的存储库。有关所需 IAM 权限的更多信息，请参阅 所需的 IAM 权限 (p. 38)。

以下是此错误的示例。

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

找不到请求的镜像

指示找不到镜像的错误通常是由该镜像不存在于上游注册表中或者未向拉取上游镜像的 IAM 主体授予 `ecr:BatchImportUpstreamImage` 权限但已在 Amazon ECR 私有注册表中创建存储库所致。要解决此错误，您应验证上游镜像和镜像标签名称正确且存在，并且已向拉取上游镜像的 IAM 主体授予所需的 IAM 权限。有关所需 IAM 权限的更多信息，请参阅 所需的 IAM 权限 (p. 38)。

以下是此错误的示例。

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

推送到存储库时出现 HTTP 403 错误或“no basic auth credentials”(没有基础级验证凭证) 错误

有时，即使您已使用 aws ecr get-login-password 命令成功通过 Docker 身份验证，也可能会从 docker push 或 docker pull 命令收到 `HTTP 403 (Forbidden)` 错误或者错误消息 `no basic auth credentials`。以下是此问题的一些已知的原因：

您已验证到其他区域

身份验证请求与特定的区域相关联，不能跨区域使用。例如，如果您从美国西部 (俄勒冈) 获得授权令牌，不能使用它对您在美国东部 (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。有关更多信息，请参阅 the section called “注册表身份验证” (p. 13)。

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 私有存储库策略 (p. 22)。

您的令牌已过期。

对于使用 `GetAuthorizationToken` 操作获取的令牌，默认授权令牌有效期为 12 小时。

`wincred` 凭证管理器中的错误

某些版本的适用于 Windows 的 Docker 使用名为 `wincred` 的凭证管理器，但它无法正确处理由 aws ecr get-login-password 生成的 Docker 登录命令 (有关更多信息，请参阅 https://github.com/docker/docker/issues/22910)。可以运行作为输出的 Docker 登录命令，但如果尝试推送或提取镜像，这些命令会失败。修复这个错误的方法是，对于从 aws ecr get-login-password 输出的 Docker 登录命令，删除其注册表参数中的 `https://` 方案。如下所示为不带 HTTPS 方案的 Docker 登录命令示例。

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

排查 Amazon ECR 错误消息问题

有时，通过 Amazon ECS 控制台或 Amazon CLI 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。

HTTP 429：请求过多或 ThrottleException

您可能会从一个或多个 Amazon ECR 命令或 API 调用收到 `429: Too Many Requests` 错误或 `ThrottleException` 错误。如果将 Docker 工具用于 Amazon ECR，那么对于 Docker 1.12.0 版及更高版本，可能会显示错误消息 `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)，允许高达 200 TPS 的突增。在每个区域，每个账户会收到一个可存储多达 200 点 `GetAuthorizationToken` 积分的存储桶。这些积分以每秒 20 点的速度补充。如果您的存储桶有 200 点积分，则可实现每秒 200 个 `GetAuthorizationToken` API 事务 (持续一秒)，然后无限期地维持每秒 20 个事务。

要处理限制错误，请在代码中实施增量退避重试函数。有关退避技术的信息，请参阅 Amazon Web Services 一般参考中的 Amazon 中的错误重试和指数退避。

HTTP 403：“User [arn] is not authorized to perform [operation]”(用户 [arn] 没有执行 [operation] 的权限)

尝试通过 Amazon ECR 执行操作时，您可能会收到以下错误：

```
$ aws ecr get-login-password
A client error (AccessDeniedException) occurred when calling the GetAuthorizationToken
 operation:
    User: arn:aws:iam::account-number:user/username is not authorized to perform:
    ecr:GetAuthorizationToken on resource: *
```

这表示您的用户没有获得使用 Amazon ECR 的权限，或者这些权限设置不正确。尤其是在对 Amazon ECR 执行操作时，请验证是否已授予用户访问该存储库的权限。有关创建和验证 Amazon ECR 权限的更多信息，请参阅 适用于 Amazon Elastic Container Registry 的 Identity and Access Management (p. 75)。

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)。

排查镜像扫描问题

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

UnsupportedImageError

尝试对使用 Amazon ECR 不支持其基础镜像扫描的操作系统构建的镜像进行基础扫描时，您可能会得到 `UnsupportedImageError` 错误。Amazon ECR 支持 Amazon Elastic Container Storage Storage

Simple Storage、Amazon Linux 2、Debian、Ubuntu、CentOS、Oracle Linux、Alpine 和 RHEL Linux 发行版的主要版本。一旦分发失去其供应商的支持，Amazon ECR 可能不再支持扫描它是否存在漏洞。Amazon ECR 不支持扫描从 Docker scratch 镜像构建的镜像。

Important

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

返回 UNDEFINED 严重性级别

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

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

要确定漏洞的严重性和描述，您可以直接从源查看 CVE。

了解扫描状态 SCAN_ELIGIBILITY_EXPIRED

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

- 如果过去 30 天内没有根据镜像推送时间戳更新镜像，则会暂停对该镜像的连续扫描。暂停扫描的镜像将显示扫描状态 SCAN_ELIGIBILITY_EXPIRED。
- 如果在 Amazon Inspector 控制台中更改了 ECR 重新扫描持续时间并且该时间已过，则镜像的扫描状态将会更改为 inactive 并显示原因代码 expired，并且将会计划关闭该镜像的所有关联调查结果。这会导致 Amazon ECR 控制台将扫描状态列为 SCAN_ELIGIBILITY_EXPIRED。

文档历史记录

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

更改	说明	日期
Amazon ECR 增强扫描持续时间支持	Amazon Inspector 增加了对启用增强扫描时监控存储库的持续时间设置支持。有关更多信息，请参阅更改增强扫描持续时间 (p. 63)。	2022 年 6 月 28 日
Amazon ECR 将存储库拉取计数指标发送到 Amazon CloudWatch	Amazon ECR 将存储库拉取计数指标发送到 Amazon CloudWatch。有关更多信息，请参阅Amazon ECR 存储库指标 (p. 109)。	2022 年 1 月 6 日
扩展了复制支持	Amazon ECR 添加了要复制的存储库进行筛选的支持。有关更多信息，请参阅 私有镜像复制 (p. 43)。	2021 年 9 月 21 日
适用于 Amazon ECR 的 Amazon 托管策略	Amazon ECR 添加了 Amazon 托管策略文档。有关更多信息，请参阅 适用于 Amazon Elastic Container Registry 的 Amazon 托管策略 (p. 82)。	2021 年 6 月 24 日
跨区域和跨账户复制	Amazon ECR 添加了对配置私有注册表复制设置的支持。有关更多信息，请参阅 私有注册表设置 (p. 14)。	2020 年 12 月 8 日
OCI 构件支持	Amazon ECR 添加了对推送和提取 Open Container Itistry (OCI) 构件的支持。新的参数 `artifactMediaType` 添加到 `DescribeImages` API 响应中以指示工件类型。 有关更多信息，请参阅推送 Helm Chart (p. 34)。	2020 年 8 月 24 日
静态加密	Amazon ECR 增加了对结合使用服务器端加密和 Amazon Key Management Service(Amazon KMS) 中所存储客户托管密钥配置加密的支持。 有关更多信息，请参阅静态加密 (p. 96)。	2020 年 7 月 29 日
多架构镜像	Amazon ECR 添加了对创建和推送用于多架构镜像的 Docker 清单列表的支持。 有关更多信息，请参阅 推送多架构镜像 (p. 33)。	2020 年 4 月 28 日
Amazon ECR 使用情况指标	Amazon ECR 添加了 CloudWatch 使用情况指标，该指标可让您了解账户的资源使用情况。您还可以从 CloudWatch 和 Service Quotas 控制台创建 CloudWatch 警报，以便在您的使用情况接近应用的服务配额时收到警报。 有关更多信息，请参阅 Amazon ECR 用量指标 (p. 108)。	2020 年 2 月 28 日
更新了 Amazon ECR 服务配额	更新了 Amazon ECR 服务配额，以包含每个 API 的配额。 有关更多信息，请参阅 Amazon ECR 服务配额 (p. 121)。	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

更改	说明	日期
镜像扫描	增加了对镜像扫描的支持，这有助于识别容器镜像中的软件漏洞。Amazon ECR 使用开源 CoreOS Clair 项目中的常见漏洞和披露 (CVE) 数据库，并为您提供扫描发现结果的列表。 有关更多信息，请参阅镜像扫描 (p. 60)。	2019 年 10 月 24 日
VPC 终端节点策略	增加了对在 Amazon ECR 接口 VPC 终端节点上设置 IAM 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. 104)。	2019 年 9 月 26 日
镜像标签可变性	增加了对将存储库配置为不可变的支持，以防止覆盖镜像标签。 有关更多信息，请参阅镜像标签可变性 (p. 59)。	2019 年 7 月 25 日
接口 VPC 终端节点 (Amazon PrivateLink)	添加了对配置由Amazon PrivateLink 提供支持的接口 VPC 终端节点的支持。这能让您在您的 VPC 和 Amazon ECR 之间创建私有连接，而无需通过 Internet、NAT 实例、VPN 连接或 Amazon Direct Connect 进行访问。 有关更多信息，请参阅Amazon ECR 接口 VPC 终端节点 (Amazon PrivateLink) (p. 101)。	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，请参阅标记私有存储库 (p. 28)。	2018 年 12 月 18 日
Amazon ECR 名称更改	重命名 Amazon Elastic Container Registry (以前名为 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镜像的生命周期管理。 有关更多信息，请参阅 生命周期策略 (p. 48)。	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，请参阅容器镜像清单格式 (p. 70)。	2017 年 1 月 27 日
Amazon ECR 正式发布	Amazon Elastic Container Registry (Amazon ECR) 是托管 Amazon Docker 注册表服务，其安全、可扩展且可靠。	2015 年 12 月 21 日

Amazon术语表

最新的Amazon术语，请参阅Amazon术语表中的Amazon一般参考。