天翼云

容器镜像服务

用户使用指南

天翼云科技有限公司

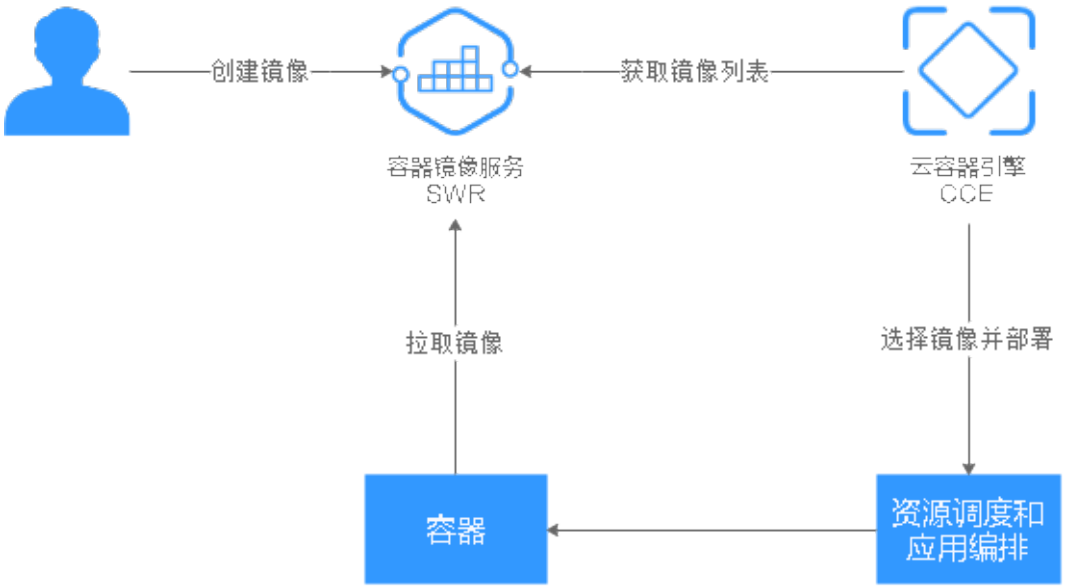
目 录

1 欢迎使用容器镜像服务

容器镜像服务（SoftWare Repository for Container，简称 SWR）是一种支持镜像全生命周期管理的服务，提供简单易用、安全可靠的镜像管理功能，包括镜像的上传、下载、删除等。

SWR 提供私有镜像库，并支持细粒度的权限管理，可以为不同用户分配相应的访问权限（读取、编辑、管理）。SWR 还支持容器镜像版本更新自动触发部署。您只需要为镜像设置一个触发器，通过触发器，可以在每次镜像版本更新时，自动更新云容器引擎（CCE）中使用该镜像部署的应用。

图1-1 SWR 使用流程



2 容器引擎基础知识

容器引擎（Docker）是一个开源的引擎，可以轻松的为任何应用创建一个轻量级、可移植的应用镜像。

安装前的准备工作

在安装容器引擎前，请了解容器引擎的基础知识，具体请参见 Docker Documentation。

选择容器引擎的版本

容器引擎几乎支持在所有操作系统上安装，用户可以根据需要选择要安装的容器引擎版本，具体请参见 https://docs.docker.com/engine/install/。

说明

- 由于 SWR 支持容器引擎 1.11.2 及以上版本上传镜像，建议下载对应版本。
- 安装容器引擎需要连接互联网，内网服务器需要绑定弹性公网 IP 后才能访问。

安装容器引擎

你可以根据自己的操作系统选择对应的安装步骤：

- **Linux 操作系统下安装**

  在 Linux 操作系统下，可以使用如下命令快速安装 Docker 的最新稳定版本。如果您想安装其他特定版本的 Docker，可参考安装 Docker。

```
curl -fsSL get.docker.com -o get-docker.sh
sh get-docker.sh
sudo systemctl daemon-reload
sudo systemctl restart docker
```

制作容器镜像

本节指导您通过 Dockerfile 定制一个简单的 Web 应用程序的容器镜像。Dockerfile 是一个文本文件，其内包含了若干指令（Instruction），每一条指令构建一层，因此每一条指令的内容，就是描述该层应当如何构建。

使用 Nginx 镜像创建容器应用，在浏览器访问时则会看到默认的 Nginx 欢迎页面，本节以 Nginx 镜像为例，修改 Nginx 镜像的欢迎页面，定制一个新的镜像，将欢迎页面改为“Hello, SWR!”。

步骤 1 以 root 用户登录容器引擎所在机器。

步骤 2 创建一个名为 Dockerfile 的文件。

mkdir mynginx

cd mynginx

touch Dockerfile

步骤 3 编辑 Dockerfile。

vim Dockerfile

增加文件内容如下：

```
FROM nginx
RUN echo '<h1>Hello, SWR!</h1>' > /usr/share/nginx/html/index.html
```

Dockerfile 指令介绍如下。

- FROM 语句：表示使用 nginx 镜像作为基础镜像，一个 Dockerfile 中 FROM 是必备的指令，并且必须是第一条指令。
- RUN 语句：格式为 RUN <命令>，表示执行 echo 命令，在显示器中显示一段“Hello, SWR!”的文字。

按“Esc”，输入**:wq**，保存并退出。

步骤 4 使用 **docker build** [*选项*] <*上下文路径*> 构建镜像。

docker build -t nginx:v1 .

- -t nginx:v1：指定镜像的名称和版本。
- **.**：指定 Dockerfile 所在目录，镜像构建命令将该路径下所有的内容打包给容器引擎帮助构建镜像。

步骤 5 执行以下命令，可查看到已成功部署的 nginx 镜像，版本为 v1。

docker images

----结束

制作镜像压缩包

本节指导您将容器镜像制作成 tar 或 tar.gz 文件压缩包。

步骤 1 以 root 用户登录容器引擎所在机器。

步骤 2 执行如下命令查看镜像。

docker images

查看需要导出的镜像及 tag。

步骤 3 执行如下命令制作镜像压缩包。

docker save [OPTIONS] IMAGE [IMAGE...]

说明

OPTIONS：--output 或-o，表示导出到文件。

压缩包格式为：.tar 或.tar.gz。

使用 docker save 制作镜像压缩包时，请用{image}:{tag}，不要用 image id，否则无法在 swr 页面上传。

示例：

```
$ docker save nginx:latest > nginx.tar
$ ls -sh nginx.tar
108M nginx.tar

$ docker save php:5-apache > php.tar.gz
$ ls -sh php.tar.gz
372M php.tar.gz

$ docker save --output nginx.tar nginx
$ ls -sh nginx.tar
108M nginx.tar

$ docker save -o nginx-all.tar nginx    # 将 nginx 所有版本打包
$ docker save -o nginx-latest.tar nginx:latest
```

----结束

导入镜像文件

本章节将指导你通过 docker load 命令将镜像压缩包导入为一个镜像。

执行方式有 2 种：

docker load < 路径/文件名.tar

docker load --input 或者-i 路径/文件名.tar

示例：

```
$ docker load --input fedora.tar
```

3 快速入门

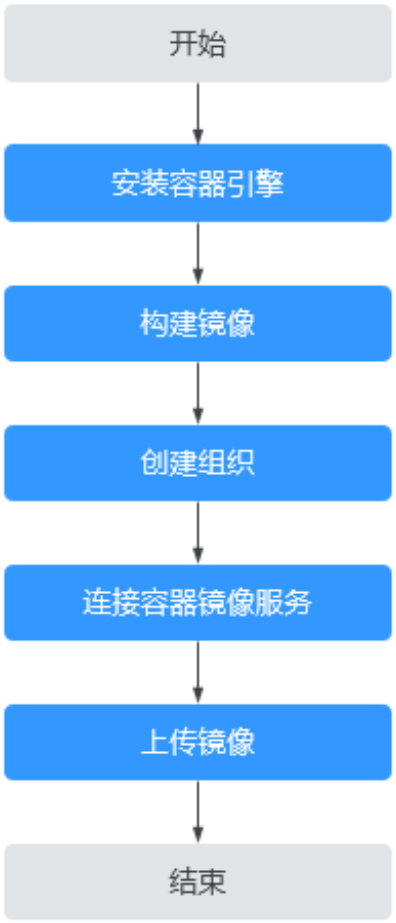
3.1 使用容器引擎客户端上传镜像

入门指引

容器镜像服务是一种支持容器镜像全生命周期管理的服务，提供简单易用、安全可靠的镜像管理功能，帮助用户快速部署容器化服务。本文档以一个 2048 应用为例，帮助您学习如何安装容器引擎并构建镜像，以及如何使用容器引擎客户端上传镜像到容器镜像仓库。

您将按以下流程学习如何使用容器镜像服务。

图3-1 入门流程



准备工作

在使用容器镜像服务前，您需要完成天翼云账号注册的准备工作。

如果您还没有天翼云帐号，请首先完成账号创建。

一、安装容器引擎

首先，您需要准备一台虚拟机并安装容器引擎，请确保容器引擎为 **1.11.2** 及以上版本。

步骤 1 创建一台带有弹性公网 IP 的 Linux 弹性云主机。

作为演示，弹性云主机和公网 IP 的规格不需要太高，例如弹性云主机的规格为“1vCPUs | 2GB”、公网 IP 带宽为“1 Mbit/s”即可，操作系统以选择“CentOS 7.6”为例。

说明

- 您也可以使用其他机器安装容器引擎，不创建弹性云主机。
- 如果您使用的是 Centos 操作系统，建议选择 CentOS7、CentOS7.2、CentOS7.3、CentOS7.4、CentOS7.5、CentOS7.6 操作系统版本，否则可能导致安装异常。

步骤 2 创建完成后返回弹性云主机列表，单击操作列的“远程登录”，以 root 用户登录弹性云主机。

步骤 3 使用如下命令快速安装容器引擎。

```
curl -fsSL get.docker.com -o get-docker.sh
sh get-docker.sh
sudo systemctl daemon-reload
sudo systemctl restart docker
```

----结束

二、构建镜像

步骤 1 在安装容器引擎的虚拟机上执行以下命令，下载 2048 应用的源码。

git clone https://gitee.com/jorgensen/2048.git

说明

若提示“git: command not found”表示未安装 Git 工具，请先安装该工具（可使用 **yum install git** 命令）。

步骤 2 下载成功后，进入“2048”目录。

cd 2048

步骤 3 修改 Dockerfile 文件。

vim Dockerfile

```
FROM nginx
COPY . /usr/share/nginx/html

EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

- FROM：指定基础镜像 nginx。
- COPY：将 2048 源码拷贝到容器内的“/usr/share/nginx/html”目录。
- EXPOSE：暴露容器的 80 端口。
- CMD：指定容器运行时的默认命令。

按“Esc”，输入**:wq**，保存并退出。

步骤 4 使用 docker build 命令构建镜像。

docker build -t 2048 .

其中，

- -t 表示给镜像加一个标签，也就是给镜像取名，这里镜像名为 2048。
- . 表示上下文路径，镜像构建命令将该路径下的所有内容打包给容器引擎帮助构建镜像。

步骤 5 执行以下命令，查看已成功构建的 2048 镜像，版本为默认的 latest。

docker images

```
# docker images
REPOSITORY   TAG      IMAGE ID      CREATED            SIZE
2048         latest   8d421c503ed0   About a minute ago   134MB
nginx        latest   dd34e67e3371   6 days ago          133MB
```

您还可以看到一个 nginx 镜像，这个镜像是从镜像仓库下载下来，作为 2048 镜像的基础镜像使用的。

步骤 6 （可选）运行容器镜像。

镜像构建成功后，您可以执行 docker run 命令运行容器镜像。

docker run -p 8080:80 2048

docker run 命令会启动一个容器，命令中-p 是将虚拟机的 8080 端口映射到容器的 80 端口，即虚拟机的 8080 端口的流量会映射到容器的 80 端口，当您在本地机器的浏览器访问“https://ECS 的弹性公网 IP:8080”时，就会访问到容器中，此时浏览器返回的内容就是 2048 应用页面。

----结束

三、创建组织

组织用于隔离镜像，并为帐号下的 IAM 用户指定不同的权限（读取、编辑、管理）。

步骤 1 登录 SWR 管理控制台。

步骤 2 选择左侧导航栏的“组织管理”，单击页面右上角的“创建组织”。

步骤 3 填写组织名称，单击“确定”。

----结束

四、连接容器镜像服务

步骤 1 登录 SWR 管理控制台。

步骤 2 选择左侧导航栏的“总览”，单击页面右上角的“登录指令”，在弹出的页面中单击 复制登录指令。

图3-2 登录指令

说明

此处生成的登录指令有效期为 24 小时，若需要长期有效的登录指令，请参考获取长期有效登录指令。

步骤 3 在安装容器引擎的虚拟机中执行上一步复制的登录指令。

登录成功会显示“Login Succeeded”。

----结束

步骤 5：上传镜像

步骤 1 在安装容器引擎的虚拟机上执行以下命令，为 2048 镜像打标签。

docker tag [镜像名称 1:版本名称 1] [镜像仓库地址]/[组织名称]/[镜像名称 2:版本名称 2]

其中，

- **[镜像名称 1:版本名称 1]**：请替换为您所要上传的实际镜像的名称和版本名称。
- **[镜像仓库地址]**：可在 SWR 控制台上查询，或在“步骤 4：连接容器镜像服务”中登录指令末尾的域名即为镜像仓库地址。
- **[组织名称]**：请替换为“步骤 3：创建组织”中创建的组织。
- **[镜像名称 2:版本名称 2]**：请替换为您期待的镜像名称和镜像版本。

示例：

docker tag 2048:latest registry.cn-jssz1.ctyun.cn /cloud-develop/2048:v1

步骤 2 上传镜像至镜像仓库。

docker push [镜像仓库地址]/[组织名称]/[镜像名称 2:版本名称 2]

示例：

docker push registry.cn-jssz1.ctyun.cn/cloud-develop/2048:v1

终端显示如下信息，表明上传镜像成功。

```
The push refers to repository [registry.cn-jssz1.ctyun.cn/cloud-develop/2048]
fbce26647e70: Pushed
fb04ab8effa8: Pushed
8f736d52032f: Pushed
009f1d338b57: Pushed
678bbd796838: Pushed
d1279c519351: Pushed
f68ef921efae: Pushed
v1: digest: sha256:0cdfc7910db531bfa7726de4c19ec556bc9190aad9bd3de93787e8bce3385f8d
size: 1780
```

返回 SWR 管理控制台，在“我的镜像”页面，执行刷新操作后可查看到对应的镜像信息。

步骤 3 镜像上传成功后，你可以使用已上传的镜像在云容器引擎中部署工作负载。

----结束

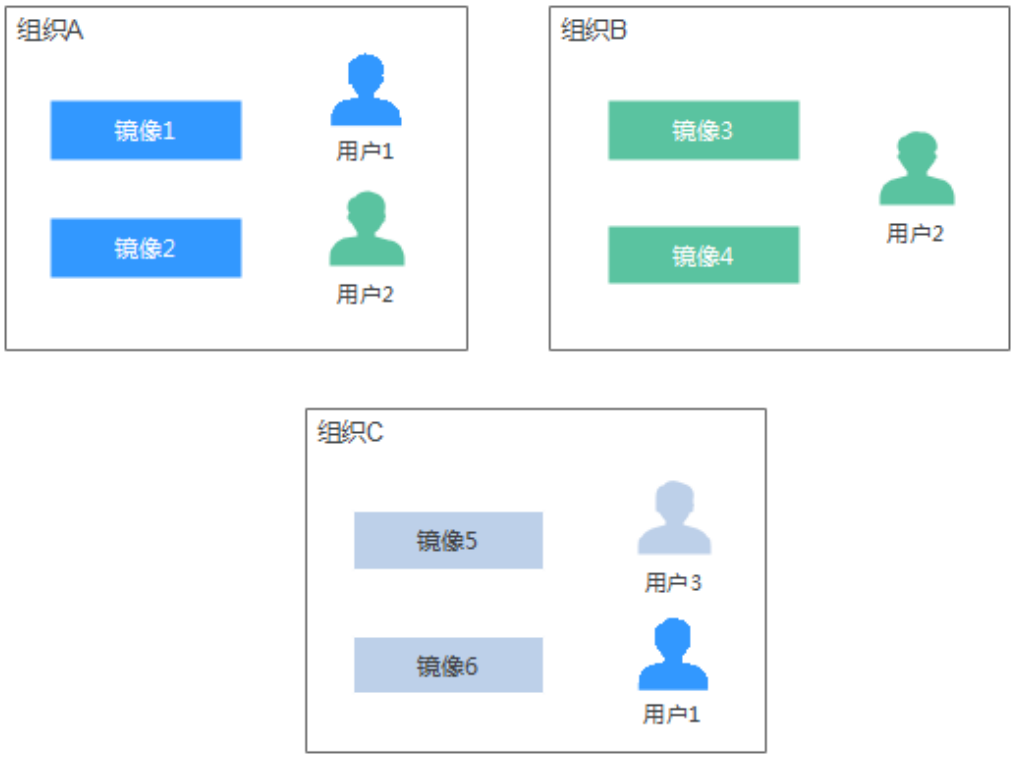
4 组织管理

操作场景

组织用于隔离镜像仓库，每个组织可对应一个公司或部门，将其拥有的镜像集中在该组织下。在不同的组织下，可以有同名的镜像。同一 IAM 用户可属于不同的组织，如下图所示。

SWR 支持为帐户下 IAM 用户分配相应的访问权限（读取、编辑、管理），具体请参见“授权管理”。

图4-1 组织



创建组织

容器镜像服务为您提供组织管理功能，方便您根据自身组织架构来构建镜像的资源管理。上传镜像前，请先创建组织。

步骤 1 登录容器镜像服务控制台。

步骤 2 单击控制台左上角的，选择区域和项目。

步骤 3 在左侧导航栏单击“组织管理”，进入组织管理页面。

步骤 4 单击页面右上角的“创建组织”按钮，在弹框中填写“组织名称”，然后单击“确定”。

图4-2 创建组织

说明

- 组织名称全局唯一，即当前区域下，组织名称唯一。创建组织时如果提示组织已存在，可能该组织名称已被其他用户使用，请重新设置一个组织名称。
- 用户在 IAM 中被授予 SWR Admin 或 Tenant Administrator 策略才有创建组织的权限。

----结束

查看组织中的镜像

创建组织后，您可以查看当前组织中的镜像。

步骤 1 登录容器镜像服务控制台。

步骤 2 单击控制台左上角的，选择区域和项目。

步骤 3 在左侧导航栏选择“组织管理”，单击右侧组织名称。

步骤 4 单击“镜像”页签，查看当前组织中的镜像。

图4-3 查看组织中的镜像

----结束

删除组织

删除组织前，请先删除组织下的所有镜像。

步骤 1 登录容器镜像服务控制台。

步骤 2 单击控制台左上角的，选择区域和项目。

步骤 3 在左侧导航栏选择“组织管理”，单击右侧组织名称。

步骤 4 单击右上角“删除”按钮，在弹出的对话框中根据提示输入“DELETE”，然后单击“确定”。

----结束

5 镜像管理

5.1 客户端上传镜像（推荐）

操作场景

客户端上传镜像，是指在安装了容器引擎客户端的机器上使用 docker 命令将镜像上传到容器镜像服务的镜像仓库。

如果容器引擎客户端机器为云上的 ECS 或 CCE 节点，根据机器所在区域有两种网络链路可以选择：

- 若机器与容器镜像仓库在同一区域，则上传镜像走内网链路。
- 若机器与容器镜像仓库不在同一区域，则上传镜像走公网链路，机器需要绑定弹性公网 IP。

约束与限制

- 使用客户端上传镜像，镜像的每个 layer 大小不能超过 10G。
- 上传镜像的容器引擎客户端版本必须为 1.11.2 及以上。

前提条件

已创建组织，请参见创建组织。

操作步骤

步骤 1 制作容器镜像或导入镜像文件。

步骤 2 连接容器镜像服务。

- 登录容器镜像服务控制台。
- 选择左侧导航栏的“总览”，单击页面右上角的“登录指令”，在弹出的页面中单击 复制登录指令。

图5-1 登录指令

说明

- 此处生成的登录指令有效期为 24 小时，若需要长期有效的登录指令，请参见获取长期有效登录指令。获取了长期有效的登录指令后，在有效期内的临时登录指令仍然可以使用。
- 登录指令末尾的域名为镜像仓库地址，请记录该地址，后面会使用到。

- 在安装容器引擎的机器中执行上一步复制的登录指令。
  登录成功会显示“Login Succeeded”。

步骤 3 在安装容器引擎的机器上执行如下命令，为 **nginx** 镜像打标签。

docker tag [镜像名称 1:版本名称 1] [镜像仓库地址]/[组织名称]/[镜像名称 2:版本名称 2]

其中，

- [镜像名称 1:版本名称 1]：请替换为您所要上传的实际镜像的名称和版本名称。
- [镜像仓库地址]：可在 SWR 控制台上查询，即步骤 2 中登录指令末尾的域名。
- [组织名称]：请替换为您创建的组织。
- [镜像名称 2:版本名称 2]：请替换为您期待的镜像名称和镜像版本。

示例：

docker tag nginx:v1 registry.cn-jssz1.ctyun.cn/cloud-develop/nginx:v1

步骤 4 上传镜像至镜像仓库。

docker push [镜像仓库地址]/[组织名称]/[镜像名称 2:版本名称 2]

示例：

docker push registry.cn-jssz1.ctyun.cn/cloud-develop/nginx:v1

终端显示如下信息，表明上传镜像成功。

```
The push refers to repository [registry.cn-jssz1.ctyun.cn/cloud-develop/nginx:v1]
fbce26647e70: Pushed
```

```
fb04ab8effa8: Pushed
8f736d52032f: Pushed
009f1d338b57: Pushed
678bbd796838: Pushed
d1279c519351: Pushed
f68ef921efae: Pushed
v1: digest: sha256:0cdfc7910db531bfa7726de4c19ec556bc9190aad9bd3de93787e8bce3385f8d
size: 1780
```

返回容器镜像服务控制台，在“我的镜像”页面，执行刷新操作后可查看到对应的镜像信息。

----结束

5.2 获取长期有效登录指令

操作场景

本章节介绍如何获取长期有效的登录指令，长期有效登录指令的有效期为永久。

说明

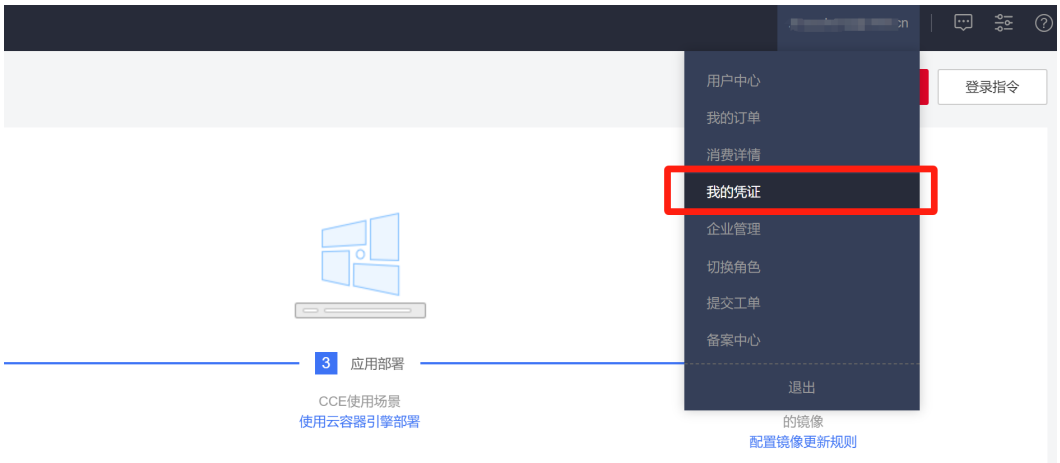
- 为保证安全，获取登录指令过程建议在开发环境执行。

操作步骤

步骤 1 获取区域项目名称、镜像仓库地址。

- 登录 SWR 管理控制台。
- 将鼠标移至页面右上角用户名称上，在下拉菜单中，单击“我的凭证”。

图5-2 我的凭证



- 在项目列表中找到您的虚拟机的所属区域及项目：

图5-3 区域与项目

- 您可以按照获取到的项目信息拼接镜像仓库地址，拼接方式为 ：registry.区域项目名称.ctyun.cn

  如用户虚拟机所在区域为苏州，那么对应的镜像仓库地址为：registry.cn-jssz1.ctyun.cn。

步骤 2 获取 AK/SK 访问密钥。

说明

访问密钥即 AK/SK（Access Key ID/Secret Access Key），表示一组密钥对，用于验证调用 API 发起请求的访问者身份，与密码的功能相似。如果您已有 AK/SK，可以直接使用，无需再次获取。

- 登录 IAM 管理控制台，将鼠标移到用户名处，单击“我的凭证”。
- 在左侧导航栏中选择“访问密钥”，单击“新增访问密钥”。
- 输入描述信息，单击“确定”。
- 在弹出的提示页面单击“立即下载”。

  下载成功后，在“credentials”文件中即可获取 AK 和 SK 信息。

表5-1 *credentials 文件示例*

User Name	*Access Key Id*	*Secret Access Key*
*a*****	*RVHVMX******	*H3nPwzgZ*******

说明

为防止访问密钥泄露，建议您将其保存到安全的位置。

步骤 3 登录一台 Linux 系统的计算机，执行如下命令获取登录密钥。

printf "AK" | openssl dgst -binary -sha256 -hmac "SK" | od -An -vtx1 | sed 's/[\n]//g' | sed 'N;s/\n//'

请将 **AK** 替换为步骤 2 *credentials 文件*的 *Access Key Id*，**SK** 替换为步骤 2 *credentials 文件*的 *Secret Access Key*。

示例：

```
printf "RVHVMX******" | openssl dgst -binary -sha256 -hmac "H3nPwzgZ******" | od -An -vtx1 | sed 's/[ \n]//g' | sed 'N;s/\n//'
```

执行上面的命令后，我们得到的登录密钥如下：

```
cab4ceab4a1545***************
```

说明

以上密钥仅为示例，请以实际获得的密钥为准。

步骤 4 **使用如下的格式拼接登录指令。**

docker login -u [区域项目名称]**@**[AK] **-p** [登录密钥] [镜像仓库地址]

其中，区域项目名称和镜像仓库地址在步骤 1 中获取，AK 在步骤 2 中获取，登录密钥为步骤 3 的执行结果。

示例：

```
docker login -u cn-jssz1@RVHVMX****** -p cab4ceab4a1545*************** registry.cn-jssz1.ctyun.cn
```

当显示“Login Succeeded”，即为登录成功。

说明

- 登录密钥字符串是经过加密的，无法逆向解密，从-p 无法获取到 SK。
- 获取的登录指令可在其他机器上使用并登录。

步骤 5 **（可选）当您退出仓库时，请使用以下命令删除您的认证信息。**

```
cd /root/.docker/
rm -f config.json
```

步骤 6 **（可选）使用 history -c 命令清理相关使用痕迹，避免隐私信息泄露。**

----结束

5.3 页面上传镜像

操作场景

本章节介绍如何通过页面上传镜像。从页面上传镜像，是指直接通过控制台页面将镜像上传到容器镜像服务的镜像仓库。

约束与限制

- 每次最多上传 10 个文件，单个文件大小（含解压后）不得超过 2G。
- 仅支持上传 1.11.2 及以上容器引擎客户端版本制作的镜像压缩包。

前提条件

- 已创建组织，请参见创建组织。
- 镜像已保存为 tar 或 tar.gz 文件，具体请参见制作镜像压缩包。

操作步骤

步骤 1 登录容器镜像服务控制台。

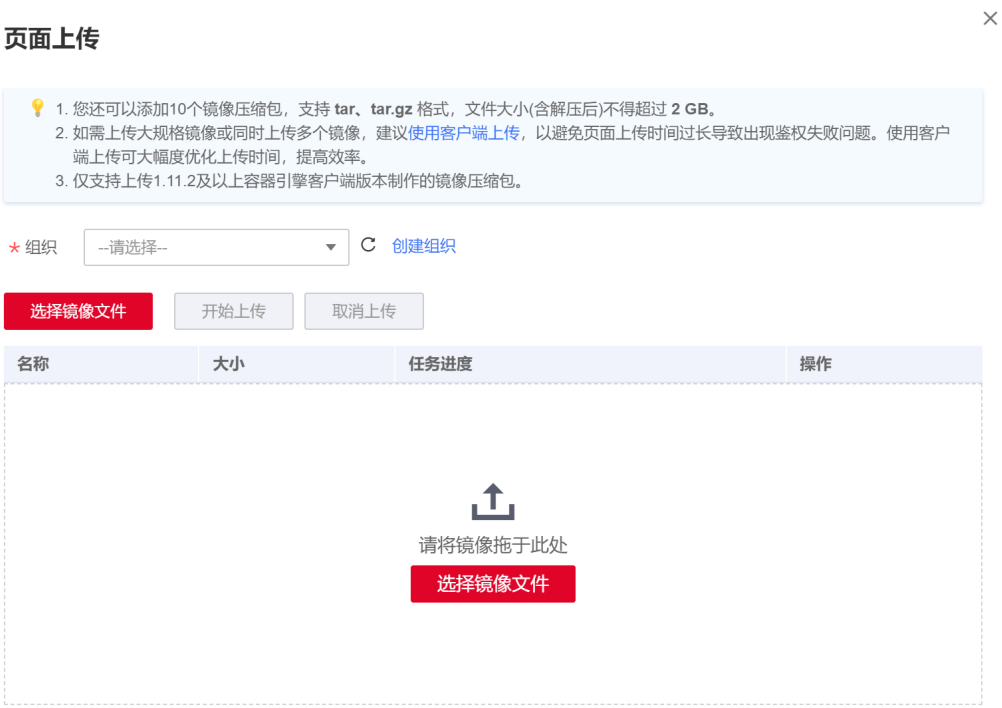
步骤 2 在左侧导航栏选择“我的镜像”，单击右上角“页面上传”。

步骤 3 在弹出的窗口中选择组织，单击“选择镜像文件”，选择要上传的镜像文件。

说明

多个镜像同时上传时，镜像文件会按照顺序逐个上传，不支持并发上传。

图5-4 上传镜像



步骤 4 单击“开始上传”。

待任务进度显示“上传完成”，表示镜像上传成功。

----结束

5.4 下载镜像

操作场景

您可以使用 docker pull 命令下载容器镜像服务中的镜像。

前提条件

- 在下载镜像前，请确保您的下载虚拟机网络畅通。
- 在下载镜像前，请联系管理员在控制台授权容器镜像服务下载权限。
- “我的镜像”展示当前用户所有的自有镜像（该用户所在组织所拥有的镜像）和共享镜像（该组织下其他用户共享的私有镜像）。

- IAM 用户创建后，需要管理员在组织中为您添加授权，您才具有该组织内镜像的读取、编辑等权限。详情请参考授权管理。

下载“我的镜像”

步骤 1 以 root 用户登录容器引擎所在的虚拟机。

步骤 2 参考“客户端上传镜像”中的方法，获取登录访问权限，连接容器镜像服务。

步骤 3 登录容器镜像服务控制台。

步骤 4 在左侧导航栏选择“我的镜像”，单击右侧镜像名称。

步骤 5 在镜像详情页面中，单击对应镜像版本“下载指令”列的复制图标，复制镜像下载指令。

图5-5 获取镜像下载指令

步骤 6 在虚拟机中执行步骤 5 复制的镜像下载指令。

示例：**docker pull registry.cn-jssz1.ctyun.cn/group/nginx:v2.0.0**

使用 **docker images** 命令查看是否下载成功。

```
# docker images
REPOSITORY                                    TAG      IMAGE ID       CREATED       SIZE
registry.cn-jssz1.ctyun.cn/group/nginx   v2.0.0   22f2bf2e2b4f   5 hours ago   22.8MB
```

步骤 7 （可选）执行如下命令将镜像保存为归档文件。

docker save [镜像名称:版本名称] > [归档文件名称]

示例：**docker save registry.cn-jssz1.ctyun.cn/group/nginx:v2.0.0 > nginx.tar**

----结束

5.5 编辑镜像属性

操作场景

镜像上传后默认为私有镜像，您可以设置镜像的属性，包括镜像的类型（“公开”或“私有”）、分类及描述。

公开镜像所有用户都能下载，私有镜像则受具体权限管理控制。您可以为用户添加授权，授权完成后，用户享有读取、编辑或管理私有镜像的权限，具体请参见在镜像详情中添加授权。

操作步骤

步骤 1 登录容器镜像服务控制台。

步骤 2 在左侧菜单栏选择“我的镜像”，单击右侧要编辑镜像的名称。

步骤 3 在镜像详情页面，单击右上角“编辑”，在弹出的窗口中根据需要编辑类型（“公开”或“私有”）、分类及描述，然后单击“确定”。

图5-6 编辑镜像属性

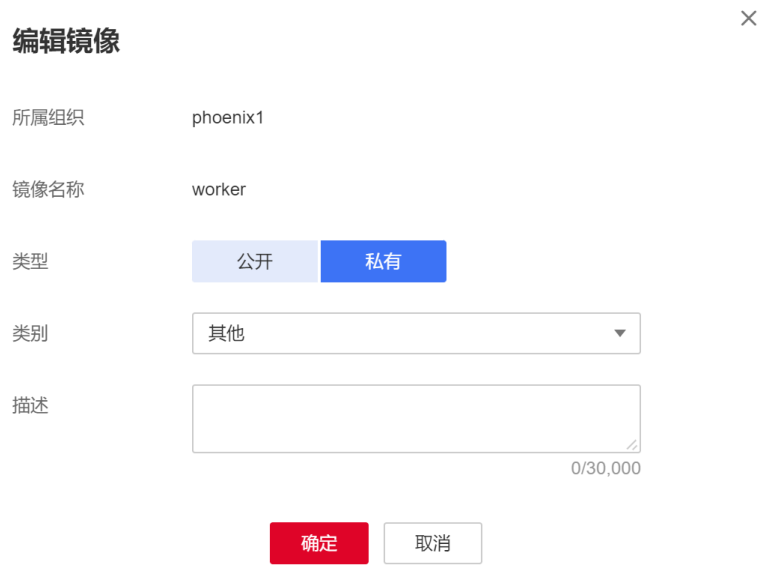


表5-2 编辑镜像

参数	说明
组织	镜像所属组织。
名称	镜像名称。
类型	镜像类型，可选择： • 公开 • 私有 说明 公开镜像所有用户都可以下载使用。 • 如果您的机器与镜像仓库在同一区域，访问仓库是通过内网访问。 • 如果您的机器与镜像仓库在不同区域，通过公网才能访问仓库，下载跨区域仓库的镜像需要机器可以访问公网。

参数	说明
分类	镜像分类，可选择： • 应用服务器 • Linux • Windows • Arm • 框架与应用 • 数据库 • 语言 • 其他
描述	输入镜像仓库描述，0-30000 个字符。

----结束

5.6 共享私有镜像

操作场景

镜像上传后，您可以共享**私有镜像**给其他帐号，并授予下载该镜像的权限。

被共享的用户需要登录容器镜像服务控制台，在“我的镜像 > 他人共享”页面查看共享的镜像。被共享的用户单击镜像名称，可进入镜像详情页面查看镜像版本、下载指令等。

约束与限制

- 仅具备该私有镜像管理权限的 IAM 用户才能共享镜像，被共享者只有只读权限，只能下载镜像。
- 镜像共享功能只能在同一区域内使用，不支持在不同区域间镜像共享。
- 一个私有镜像最多可以共享给 500 个租户。

操作步骤

步骤 1 登录容器镜像服务控制台。

步骤 2 在左侧导航栏选择“我的镜像”，单击右侧镜像的名称。

步骤 3 在镜像详情页面选择“共享”页签。

步骤 4 单击“共享镜像”，根据下表填写相关参数，然后单击“确定”。

图5-7 共享镜像

表5-3 共享镜像

参数	说明
共享给	输入帐号名称。
截止日期	选择共享截止日期。如勾选“永久有效”，则共享永久有效。
描述	输入描述，0-1000 个字符。
权限	当前仅支持“下载”权限。

步骤 5 共享完成后，您可以在“我的镜像 > 自有镜像”中，勾选“我共享的镜像”，查看所有共享的镜像。

----结束

5.7 添加触发器

操作场景

容器镜像服务可搭配云容器引擎 CCE 一起使用，实现镜像版本更新时自动更新使用该镜像的应用。您只需要为镜像添加一个触发器，通过触发器，可以在每次生成新的镜像版本时，自动执行更新动作，如：自动更新使用该镜像的应用。

前提条件

更新应用镜像版本之前，请确保已创建容器应用，将镜像部署到云容器引擎 CCE 或云容器实例 CCI。

如未创建，请登录云容器引擎工作负载页面进行创建，具体创建方法请参见云容器引擎用户指南。

操作步骤

步骤 1 登录容器镜像服务控制台。

步骤 2 在左侧导航栏选择“我的镜像”，单击右侧镜像名称，进入镜像详情页。

步骤 3 选择“触发器”页签，单击“添加触发器”，填写相关参数，然后单击“确定”。

图5-8 添加触发器

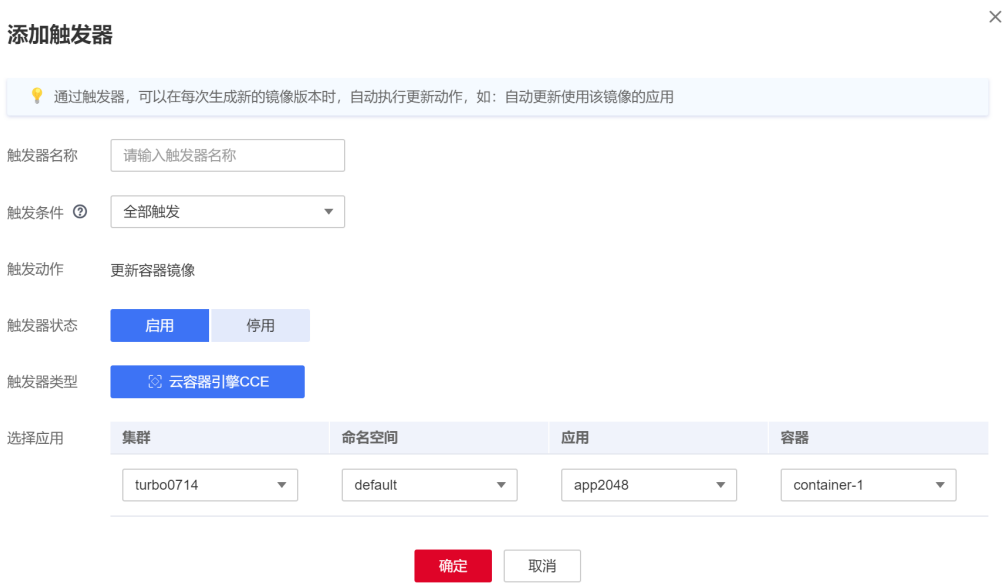


表5-4 触发器

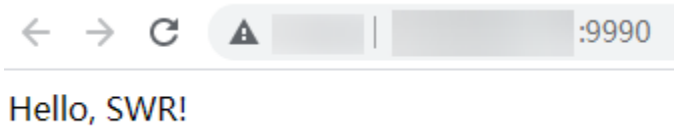
参数	说明
触发器名称	自定义触发器的名称。 字母开头，由字母、数字、下划线_、中划线-组成，下划线、中划线不能连续且不能作为结尾，1-64 个字符。
触发条件	支持如下三种触发条件，当镜像有新版本时，触发部署应用。 • 全部触发：有新的镜像版本生成或镜像版本发生更新时，触发部署。 • 指定版本号触发：有指定镜像版本生成或更新时，触发部署。 • 正则触发：有符合正则表达式的镜像版本生成或更新时，触发部

参数	说明
	署。正则表达式规则如下： – *：匹配不包含路径分隔符“/”的任何字段。 – **：匹配包含路径分隔符“/”的任何字段。 – ?：匹配任何单个非“/”的字符。 – {选项 1，选项 2, ...}：同时匹配多个选项。
触发动作	当前仅支持更新容器的镜像，需指定更新的应用，以及该应用下的容器。
触发器状态	选择“启用”。
触发器类型	选择“云容器引擎 CCE”。
选择应用	选择要更新镜像的容器。

----结束

示例 1：触发条件为“全部触发”

假设有一个欢迎页面为“Hello, SWR!”的 Nginx 镜像（版本号为 v1），使用该镜像创建了名称为“nginx”的无状态负载，该负载提供对外访问。



Hello, SWR!

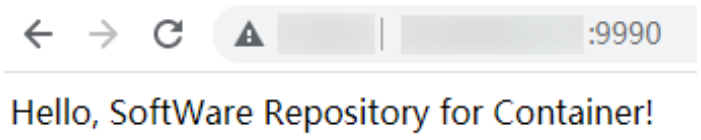
- 为 Nginx 镜像添加触发器。

  触发器名称填写“All_tags”，触发条件选择“全部触发”，选择使用了 Nginx 镜像的无状态负载及容器。
- 为 Nginx 镜像新增一个 v2 版本，该版本的欢迎页面为“Hello, SoftWare Repository for Container!”。
- 确认是否触发成功。

  在“触发器”页签，单击 图标，查看触发结果为“成功”。

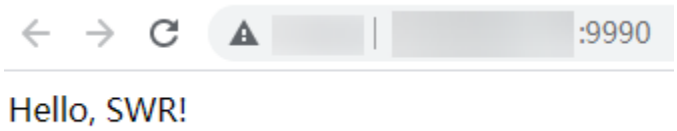
  图5-9 触发结果

工作负载的访问页面已变更为“Hello, SoftWare Repository for Container!”。



示例 2：触发条件为“正则触发”

假设有一个欢迎页面为“Hello, SWR!”的 Nginx 镜像（版本号为 v1），使用该镜像创建了名称为“nginx”的无状态负载，该负载提供对外访问。



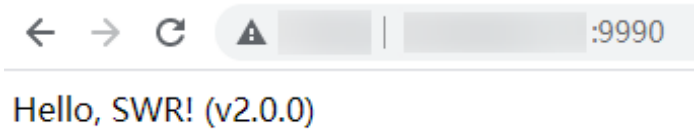
- 为 Nginx 镜像添加触发器。

  触发器名称填写“Tags_regular_expression”，触发条件选择“正则触发”，输入正则表达式：^v2.*（匹配以 v2 开头的版本号），选择使用了 Nginx 镜像的无状态负载及容器。
- 为 Nginx 镜像新增一个 v1.0.0 版本，该版本的欢迎页面为“Hello, SWR! (v1.0.0)”。
- 为 Nginx 镜像再新增一个 v2.0.0 版本，该版本的欢迎页面为“Hello, SWR! (v2.0.0)”。
- 确认是否触发成功。

  在“触发器”页签，单击 图标，查看触发结果。可以看出，只有 v2.0.0 版本被触发了应用的自动更新，符合设置的正则表达式规则。

  图5-10 触发结果示例

工作负载的访问页面已变更为“Hello, SWR! (v2.0.0)”。



5.8 添加镜像老化规则

操作场景

镜像上传后，您可以添加镜像老化规则。容器镜像服务提供了如下两种类型的镜像老化处理规则，规则设置完成后，系统会根据已定义的规则自动执行镜像老化操作。

- 存活时间：设置该类型的老化规则后，留存时间超过指定时间的老旧镜像将被删除。
- 版本数目：设置该类型的老化规则后，留存镜像超过指定值时，老旧镜像将被删除。

此外，对于特定版本的镜像可通过添加过滤策略来保留，免受老化规则的影响。

约束与限制

一个镜像仅支持添加一个老化规则。如需添加新的老化规则，需要删除已有老化规则。

操作步骤

步骤 1 登录容器镜像服务控制台。

步骤 2 在左侧导航栏选择“我的镜像”，单击右侧镜像名称，进入镜像详情页。

步骤 3 选择“镜像老化”页签，单击“添加规则”，填写相关参数，然后单击“确定”。

图5-11 创建老化规则

表5-5 添加镜像老化规则

参数	说明

参数	说明
规则类型	分为存活时间和版本数目。 • 存活时间：设置该类型的老化规则后，留存时间超过指定时间的老旧镜像将被删除。 • 版本数目：设置该类型的老化规则后，留存镜像超过指定值时，老旧镜像将被删除。
保留天数	镜像留存的最大天数，可设置为 1~365 的整数。规则类型设置为“存活时间”时，需要配置此参数。
保留数目	镜像留存的最大数目，可设置为 1~1000 的整数。规则类型设置为“版本数目”时，需要配置此参数。
过滤标签	输入将被过滤的镜像版本，在应用老化规则前指定版本的镜像将被过滤掉。
过滤正则	输入将被过滤的版本正则式，在应用老化规则前所有版本号满足正则表达式的镜像将被过滤掉。

镜像老化规则添加成功后，系统会立即进行一次查询，清理掉符合老化规则的镜像，且在“老化日志”中显示清理结果。

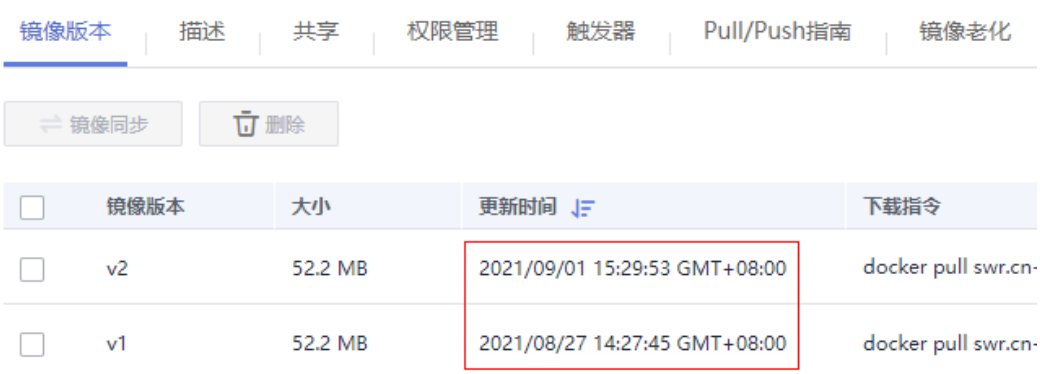
图5-12 查看规则列表和老化日志

----结束

示例 1：规则类型为“存活时间”

假设“nginx”镜像包含两个版本：v1 和 v2，更新时间如下图：

图5-13 镜像版本



- 添加老化规则。

  规则类型为“存活时间”，保留天数为“3”。
- 确认规则是否生效。

  查看“老化日志”，v1 版本的镜像留存时间超过 3 天（假设当前时间为 2021/09/01 16:00:00），因此被自动清除。

图5-14 老化日志

查看“镜像版本”，v1 版本已被清除，只剩 v2 版本。

图5-15 镜像版本 V2

以上说明老化规则已生效。

示例 2：规则类型为“版本数目”，且设置“过滤正则”

假设“nginx”镜像包含四个版本：v1、v2、v1.0.0、v2.0.0，如下图：

图5-16 nginx 镜像版本

- 添加老化规则。

  规则类型为“版本数目”，保留数目为“1”，过滤正则为：^v2.*（匹配以 v2 开头的版本号）。
- 确认规则是否生效。

  因为 v2 和 v2.0.0 版本匹配设置的正则表达式，在应用老化规则前会被过滤掉，v1 和 v1.0.0 版本只会保留一个，v1 更老旧，因此会被清除掉。

  查看“老化日志”和“镜像版本”，v1 版本被清除，说明老化规则已生效。

图5-17 老化日志示例

图5-18 镜像版本示例

6 授权管理

操作场景

如果您需要对容器镜像服务进行权限管理，您需要使用统一身份认证服务 IAM 对操作用户配置权限，当您具有 SWR Admin 或者 Tenant Administrator 系统权限时，您就拥有了 SWR 的管理员权限，可以在 SWR 中为其他 IAM 用户进行授权。

说明

拥有 SWR 管理员权限的用户，默认拥有所有组织下的镜像管理权限，即使该用户不在组织的授权用户列表中。

如果您没有 SWR 的管理员权限，就需要已拥有 SWR 管理员权限的用户在 SWR 中进行授权管理，为您添加对某个镜像的权限或对某个组织中所有镜像的权限。

场景示例：

- 示例一：我是拥有 ServiceStage Developer 权限（SWR 只读权限）的 IAM 用户，想要下载 SWR 管理员所创建的“group”组织下的“nginx”镜像。

  策略：SWR 管理员在“nginx”镜像详情中为您授予“读取”权限，授权完成后，您将享有下载该镜像的权限。
- 示例二：我是 SWR 管理员，需要给公司内部员工授权一个组织的镜像上传权限。

  策略：您在组织详情“用户”页签下为该员工授予“编辑”权限。

授权方法

容器镜像服务中给 IAM 用户添加权限有如下两种方法：

- **在镜像详情中添加授权：**授权完成后，IAM 用户享有读取/编辑/管理该镜像的权限。
- **在组织中添加授权：**使 IAM 用户对组织内所有镜像享有读取/编辑/管理的权限。

图6-1 用户权限

容器镜像服务中为用户添加的权限有如下三种类型：

- 读取：只能下载镜像，不能上传。
- 编辑：下载镜像、上传镜像、编辑镜像属性以及添加触发器。
- 管理：下载镜像、上传镜像、删除镜像或版本、编辑镜像属性、添加授权、添加触发器以及共享镜像。

说明

页面上传镜像功能要求具备组织的编辑或管理权限，在镜像详情中添加的编辑或管理权限不支持页面上传镜像。

在镜像详情中添加授权

在镜像详情中为 IAM 用户添加授权，授权完成后，该帐号下 IAM 用户享有读取/编辑/管理该镜像的权限。

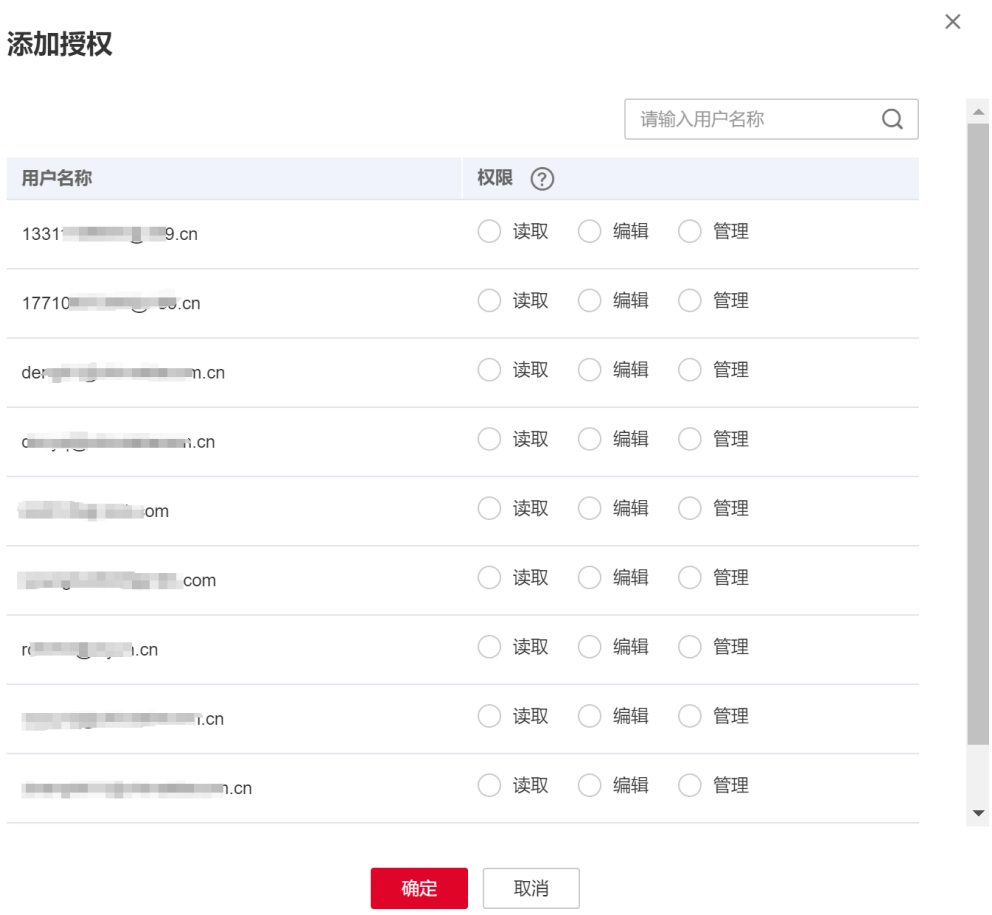
步骤 1 登录容器镜像服务控制台。

步骤 2 在左侧导航栏选择“我的镜像”，单击右侧待编辑镜像的名称。

步骤 3 在镜像详情页面选择“权限管理”页签。

步骤 4 单击“添加授权”，选择 IAM 用户名称，添加“读取/编辑/管理”的权限，添加后，该 IAM 用户享有对应权限。

图6-2 在镜像详情中添加授权



----结束

在镜像详情中修改/删除授权

您还可以在镜像详情中修改用户权限及删除用户权限。

- 修改授权：在“权限管理”页签下用户所在行单击“修改”，在“权限”所在列选择新的权限，然后单击“保存”。

  图6-3 在镜像详情中修改授权

- 删除授权：在“权限管理”页签下用户所在行单击“删除”。在弹出的对话框中根据提示输入“DELETE”，然后单击“确定”。

图6-4 在镜像详情中删除授权

在组织中添加授权

IAM 用户创建后，需要管理员在组织中为用户添加授权，使 IAM 用户对组织内所有镜像享有读取/编辑/管理的权限。

只有具备“管理”权限的帐号和 IAM 用户才能添加授权。

步骤 1 登录容器镜像服务控制台。

步骤 2 在左侧菜单栏选择“组织管理”，单击右侧组织名称后的“详情”。

步骤 3 在“用户”页签下单击“添加授权”，在弹出的窗口中为 IAM 用户选择权限，然后单击“确定”。

图6-5 在组织中添加授权

----结束

在组织中修改/删除授权

您还可以在组织中修改用户权限及删除用户权限。

- 修改授权：在“用户”页签下用户所在行单击“修改”，在“权限”所在列选择新的权限，然后单击“保存”。

- 在“用户”页签下用户所在行单击“删除”。在弹出的对话框中根据提示输入“DELETE”，然后单击“确定”。

----结束

7 审计

7.1 支持云审计的关键操作

操作场景

云审计服务（Cloud Trace Service，CTS），提供对各种云资源操作记录的收集、存储和查询功能，可用于支撑安全分析、合规审计、资源跟踪和问题定位等常见应用场景。

通过云审计服务，您可以记录与 SWR 相关的操作事件，便于日后的查询、审计和回溯。

支持审计的关键操作列表

表7-1 云审计服务支持的 SWR 操作列表

操作名称	资源类型	事件名称
创建组织权限	usernamespaceauth	createUserNamespaceAuth
修改组织权限	usernamespaceauth	updateUserNamespaceAuth
删除组织权限	usernamespaceauth	deleteUserNamespaceAuth
创建版本	version	createVersion
修改版本	version	updateVersion
删除版本	version	deleteVersion
上传镜像包	image	uploadImagePackage
创建组织	usernamespace	createUserNamespace
删除组织	usernamespace	deleteUserNamesapce
创建触发器	trigger	createTrigger
修改触发器	trigger	updateTrigger
删除触发器	trigger	deleteTrigger

操作名称	资源类型	事件名称
创建仓库权限	userrepositoryauth	createUserRepositoryAuth
修改仓库权限	userrepositoryauth	updateUserRepositoryAuth
删除仓库权限	userrepositoryauth	deleteUserRepositoryAuth
创建镜像仓库	imagerepository	createImageRepository
修改镜像仓库	imagerepository	updateImageRepository
删除镜像仓库	imagerepository	deleteImageRepository
删除镜像版本	imagetag	deleteImageTag
生成登录指令	dockerlogincmd	createDockerConfig
创建共享镜像	imagerepositoryaccessdomain	createImageRepositoryAccessDomain
修改共享镜像	imagerepositoryaccessdomain	updateImageRepositoryAccessDomain
删除共享镜像	imagerepositoryaccessdomain	deleteImageRepositoryAccessDomain

7.2 查看云审计日志

操作场景

开启了云审计服务（CTS）后，系统开始记录 SWR 相关的操作。CTS 会保存最近 1 周的操作记录。

本小节介绍如何在 CTS 管理控制台查看最近 1 周的操作记录。

操作步骤

步骤 1 登录 CTS 管理控制台。

步骤 2 选择左侧导航栏的“事件列表”，进入事件列表页面。

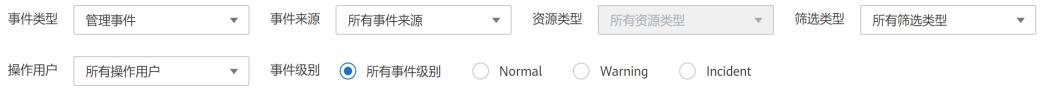
步骤 3 事件记录了云资源的操作详情，设置筛选条件，单击“查询”。

当前事件列表支持四个维度的组合查询，详细信息如下：

- 事件类型、事件来源、资源类型和筛选类型。

  在下拉框中选择查询条件。其中，“事件类型”选择“管理事件”，“事件来源”选择“SWR”。

图7-1 设置筛选条件



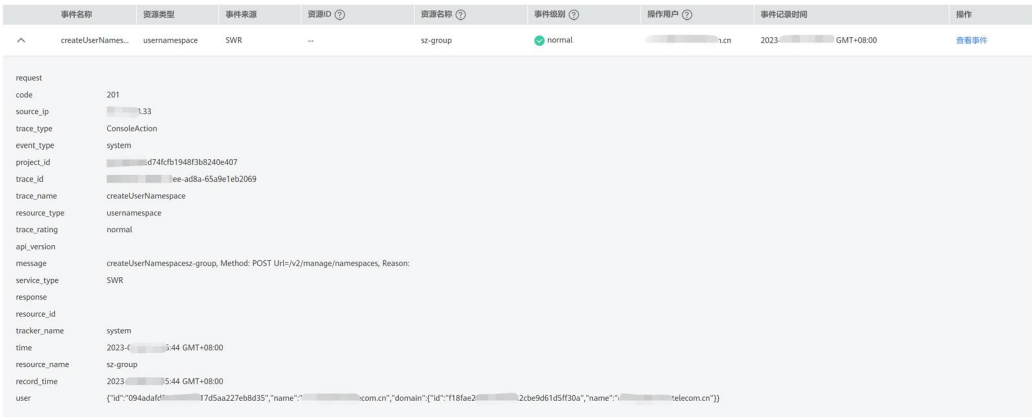
其中，筛选类型选择“按资源 ID”时，还需手动输入某个具体的资源 ID，目前仅支持全字匹配模式的查询。

筛选类型选择“按资源名称”时，选框下拉列表会自动显示符合筛选条件的资源名称。

- 操作用户：在下拉框中选择某一具体的操作用户。
- 事件级别：可选项为“所有事件级别”、“Normal”、“Warning”、“Incident”，只可选择其中一项。
- 时间范围：可选项为“最近 1 小时”、“最近 1 天”、“最近 1 周”和“自定义时间段”，本示例选择“最近 1 周”。

步骤 4 在需要查看的事件左侧，单击 图标展开该事件的详细信息。

图7-2 展开事件



步骤 5 在需要查看的事件右侧，单击“查看事件”，弹出一个窗口，显示了该操作事件结构的详细信息。

----结束

8 常见问题

8.1 通用类

8.1.1 什么是容器镜像服务

容器镜像服务（SoftWare Repository for Container，简称 SWR）是一种支持容器镜像全生命周期管理的服务，提供简单易用、安全可靠的镜像管理功能，帮助用户快速部署容器化服务。

8.1.2 什么是组织？

组织用于隔离镜像仓库，每个组织可对应一个公司或部门，将其拥有的镜像集中在该组织下。

同一用户可属于不同的组织。容器镜像服务支持为组织中不同用户分配相应的访问权限（读取、编辑、管理）。

8.1.3 容器镜像服务是否收费？

容器镜像服务暂不收费，目前可免费使用。

8.1.4 SWR 最多能存储多少个镜像？

SWR 目前对上传的镜像数量没有限制，您可以根据需要上传镜像。

8.1.5 容器镜像服务的带宽多大？

容器镜像服务的带宽会根据用户使用情况动态变化。

8.1.6 容器镜像服务的配额是多少？

容器镜像服务对镜像数量没有配额限制，您可以根据需要上传镜像。

容器镜像服务对单个用户的组织数量限定了配额。如您需要添加更多的组织，请提交工单申请。

表8-1 容器镜像服务配额

资源类型	配额
组织	5

8.1.7 长期有效的登录指令与临时登录指令的区别是什么？

- 临时的登录指令代指 6 个小时后会过期失效，不能再被使用的登录指令。可应用在临时使用，对外单次授权等场景中，对安全性要求较高的生产集群也可通过定时刷新的方式进行使用。
- 长期有效的登录指令有效期为永久。可应用在前期测试、CICD 流水线及容器集群拉取镜像等场景中。

获取了长期有效的登录指令后，在有效期内的临时登录指令仍然可以使用。

长期有效的登录指令与临时登录指令均不受限制，可以多台机器多人同时登录。

8.1.8 SWR 私有镜像最多可以共享给多少个租户？

500 个。

8.2 镜像类

8.2.1 如何安装容器引擎？

容器引擎几乎支持在所有操作系统上安装，用户可以根据需要选择要安装的容器引擎版本。

说明

- 容器镜像服务支持使用容器引擎 1.11.2 及以上版本上传镜像。
- 安装容器引擎需要连接互联网，内网服务器需要绑定弹性 IP 后才能访问。

另外，在 Linux 操作系统下，可以使用如下命令快速安装容器引擎。

curl -fsSL get.docker.com -o get-docker.sh

sh get-docker.sh

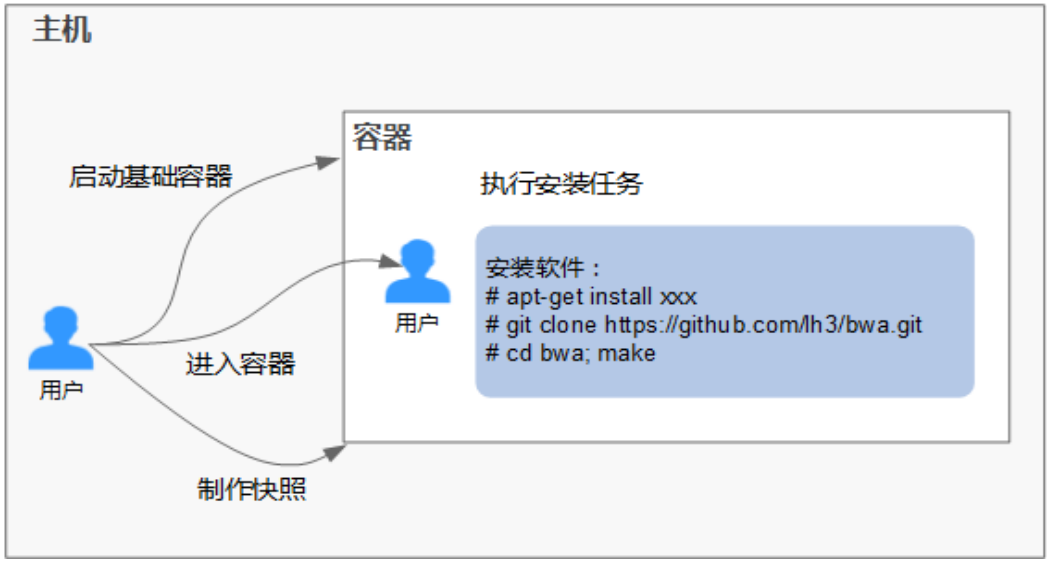
8.2.2 如何制作容器镜像？

自行制作容器镜像，主要有两种方法：

- 制作快照方式获得镜像（偶尔制作的镜像）：在基础镜像上（比如 Ubuntu），先登录镜像系统并安装容器引擎软件，然后整体制作快照。
- Dockerfile 方式构建镜像（经常更新的镜像）：将软件安装的流程写成 Dockerfile，使用 docker build 构建成容器镜像。

方法一：制作快照方式获得镜像

如果后续镜像没有变化，可采用方法一制作镜像。



具体操作如下：

- 找一台主机，安装容器引擎软件。
- 启动一个空白的基础容器，并进入容器。

  例如：启动一个 CentOS 的容器。

  docker run -it centos
- 执行安装任务。

  yum install XXX

  git clone https://github.com/lh3/bwa.git

  cd bwa;make

  说明

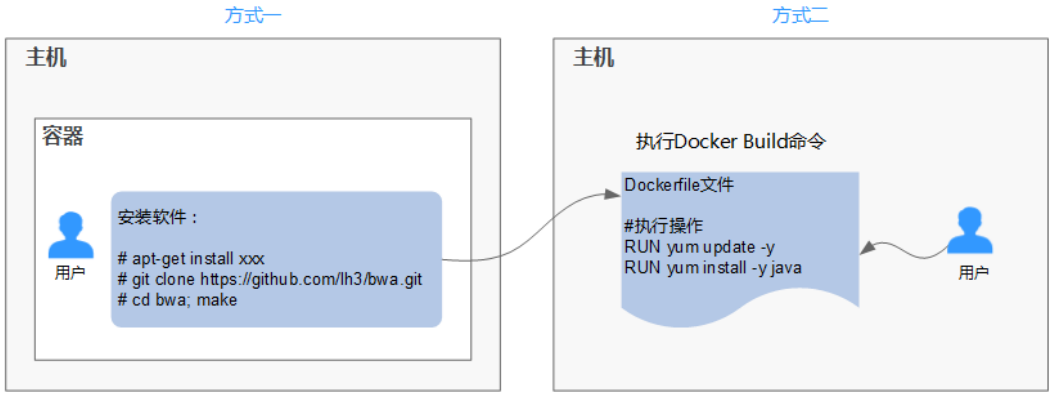
  请预先安装好 Git，并检查本机是否有 ssh key 设置。
- 输入 **exit** 退出容器。
- 制作快照。

  docker commit -m "xx" -a "test" container-id test/image:tag
  - -a：提交的镜像作者。
  - container-id：步骤 2 中的容器 id。可以使用 **docker ps -a** 查询得到容器 id。
  - -m：提交时的说明文字。
  - test/image:tag：仓库名/镜像名:TAG 名。
- 执行 **docker images** 可以查看到制作完成的容器镜像。

方法二：使用 Dockerfile 方式构建

如果后续镜像经常变更（例如某个软件更新版本），则需要采用方法二制作镜像。若仍采用方法一制作镜像，则每次变更都需要重新执行方法一的命令，操作过程比较繁琐，所以建议使用自动化制作镜像的方法。

其实就是将方法一制作镜像的方法，用文件方式写出来（文件名为 Dockerfile）。然后执行：**docker build -t test/image:tag.**命令（命令中“.”表示 Dockerfile 文件的路径），自动完成镜像制作。



简单的 Dockerfile 示例：

说明

如果依赖外部网络，请搭建网络环境，并保证网络可用。

```
#Version 1.0.1
FROM centos:latest

#设置 root 用户为后续命令的执行者
USER root

#执行操作
RUN yum update -y
RUN yum install -y java

#使用&&拼接命令
RUN touch test.txt && echo "abc" >>abc.txt

#对外暴露端口
EXPOSE 80 8080 1038

#添加网络文件
ADD https://www.baidu.com/img/bd_logo1.png /opt/

#设置环境变量
ENV WEBAPP_PORT=9090

#设置工作目录
WORKDIR /opt/

#设置启动命令
ENTRYPOINT ["ls"]
```

```
#设置启动参数
CMD ["-a", "-l"]

#设置卷
VOLUME ["/data", "/var/www"]

#设置子镜像的触发操作
ONBUILD ADD . /app/src
ONBUILD RUN echo "on build excuted" >> onbuild.txt
```

Dockerfile 基本语法

- FROM：

  指定待扩展的父级镜像（基础镜像）。除注释之外，文件开头必须是一个 FROM 指令，后面的指令便在这个父级镜像的环境中运行，直到遇到下一个 FROM 指令。通过添加多个 FROM 命令，可以在同一个 Dockerfile 文件中创建多个镜像。

- MAINTAINER：

  声明创建镜像的作者信息：用户名、邮箱，非必须参数。

- RUN：

  修改镜像的命令，常用来安装库、安装程序以及配置程序。一条 RUN 指令执行完毕后，会在当前镜像上创建一个新的镜像层，接下来的指令会在新的镜像上继续执行。RUN 语句有两种形式：

  - **RUN yum update**：在**/bin/sh** 路径中执行的指令。
  - **RUN ["yum", "update"]**：直接使用系统调用 **exec** 来执行。
  - **RUN yum update && yum install nginx**：使用&&符号将多条命令连接在同一条 RUN 语句中。

- EXPOSE：

  指明容器内进程对外开放的端口，多个端口之间使用空格隔开。

  运行容器时，通过设置参数**-P**（大写）即可将 EXPOSE 里所指定的端口映射到主机上其他的随机端口，其他容器或主机可以通过映射后的端口与此容器通信。

  您也可以通过设置参数**-p**（小写）将 Dockerfile 中 EXPOSE 中没有列出的端口设置成公开。

- ADD：

  向新镜像中添加文件，这个文件可以是一个主机文件，也可以是一个网络文件，也可以是一个文件夹。

  - 第一个参数：源文件（夹）。
    - 如果是相对路径，必须是相对于 Dockerfile 所在目录的相对路径。
    - 如果是 URL，会将文件先下载下来，然后再添加到镜像里。
  - 第二个参数：目标路径。
    - 如果源文件是主机上的 zip 或者 tar 形式的压缩文件，容器引擎会先解压缩，然后将文件添加到镜像的指定位置。
    - 如果源文件是一个通过 URL 指定的网络压缩文件，则不会解压。

- VOLUME：

在镜像里创建一个指定路径（文件或文件夹）的挂载点，这个容器可以来自主机或者其它容器。多个容器可以通过同一个挂载点共享数据，即便其中一个容器已经停止，挂载点也仍可以访问。

- WORKDIR：

  为接下来执行的指令指定一个新的工作目录，这个目录可以是绝对目录，也可以是相对目录。根据需要，WORKDIR 可以被多次指定。当启动一个容器时，最后一条 WORKDIR 指令所指的目录将作为容器运行的当前工作目录。

- ENV：

  设置容器运行的环境变量。在运行容器的时候，通过设置**-e** 参数可以修改这个环境变量值，也可以添加新的环境变量。

  例如：

  docker run -e WEBAPP_PORT=8000 -e WEBAPP_HOST=www.example.com ...

- CMD：

  用来设置启动容器时默认运行的命令。

- ENTRYPOINT：

  用来指定容器启动时的默认运行的命令。区别在于：运行容器时添加在镜像之后的参数，对 ENTRYPOINT 是拼接，CMD 是覆盖。

  - 若在 Dockerfile 中指定了容器启动时的默认运行命令为 **ls -l**，则运行容器时默认启动命令为 **ls -l**，例如：
    - **ENTRYPOINT ["ls", "-l"]**：指定容器启动时的程序及参数为 **ls -l**。
    - **docker run centos**：当运行 centos 容器时，默认执行的命令是 **docker run centos ls -l**
    - **docker run centos -a**：当运行 centos 容器时拼接了-a 参数，则默认运行的命令是 **docker run centos ls -l -a**
  - 若在 Dockerfile 中指定了容器启动时的默认运行命令为**--entrypoint**，则在运行容器时如果需要替换默认运行命令，可以通过添加**--entrypoint** 参数来替换 Dockerfile 中的指定。例如：

    docker run gutianlangyu/test --entrypoint echo "hello world"

- USER：

  为容器的运行及 RUN、CMD、ENTRYPOINT 等指令的运行指定用户或 UID。

- ONBUILD：

  触发器指令。构建镜像时，容器引擎的镜像构建器会将所有的 ONBUILD 指令指定的命令保存到镜像的元数据中，这些命令在当前镜像的构建过程中并不会执行。只有新的镜像使用 FROM 指令指定父镜像为当前镜像时，才会触发执行。

  使用 FROM 以这个 Dockerfile 构建出的镜像为父镜像，构建子镜像时：

ONBUILD ADD . /app/src：自动执行 **ADD . /app/src**

8.2.3 如何制作镜像压缩包？

使用 docker save 命令可将容器镜像制作成 tar 或 tar.gz 文件压缩包，具体命令格式如下：

docker save [OPTIONS] IMAGE [IMAGE...]

OPTIONS 说明：--output、-o，表示导出到文件。

示例：

```
$ docker save nginx:latest > nginx.tar
$ ls -sh nginx.tar
108M nginx.tar

$ docker save php:5-apache > php.tar.gz
$ ls -sh php.tar.gz
372M php.tar.gz

$ docker save --output nginx.tar nginx
$ ls -sh nginx.tar
108M nginx.tar

$ docker save -o nginx-all.tar nginx
$ docker save -o nginx-latest.tar nginx:latest
```

8.2.4 docker push 使用什么协议将镜像推动到 SWR？

HTTPS 协议。

8.2.5 同名称同 tag 的镜像上传后会覆盖之前的镜像吗？

会覆盖，保留最新上传的镜像。

8.2.6 SWR 单个 layer 的最大限制是多少？

使用客户端向 SWR 上传镜像，镜像的每个 layer 大小不能超过 10G。

8.2.7 为什么通过客户端上传和页面上传的镜像大小不一样？

问题描述

假设在本地 Docker 客户端上制作了一个 nginx 镜像，版本号为 v2.0.0，使用 **docker images** 命令查询 SIZE 为 **22.8MB**：

```
$ docker images
REPOSITORY                    TAG           IMAGE ID       CREATED         SIZE
nginx                         v2.0.0        22f2bf2e2b4f   9 days ago      22.8MB
```

- 通过客户端（即执行 docker push 命令）上传该镜像至 SWR 镜像仓库，查询大小为 **9.5MB**。

镜像版本	大小	更新时间
v2.0.0	9.5 MB	2021/09/01 19:46:12 GMT+08:00

- 在本地 Docker 客户端将镜像打包为 tar 格式，将 nginx.tar 下载至本地后，通过页面方式上传至 SWR 镜像仓库，查询大小为 **23.2MB**。

镜像版本	大小	更新时间
v2.0.0	23.2 MB	2021/09/10 09:59:28 GMT+08:00

可以发现，通过客户端和页面上传的镜像大小不一样。

原因分析

使用客户端上传的镜像每一层 layer 都进行了 tgz 压缩，而页面上传的镜像包是经过 docker 打包的，每一层 layer 只进行了打包，没有压缩。所以两种方式上传的镜像大小显示会不一致。

8.2.8 是否支持跨区域同步镜像？

目前不支持跨区域同步镜像。

如果您想把最新推送的镜像同步到其他区域镜像仓库内，您可以先将镜像下载到本地，然后上传到需要同步区域的镜像仓库中。

8.2.9 SWR 支持 ARM 镜像上传吗？

SWR 镜像仓库对镜像的内核架构是没有任何限制的，您上传 ARM 架构的镜像和上传 x86 的镜像是没有区别的，直接上传即可。

8.2.10 如何通过 API 上传镜像到 SWR？

SWR 暂时没有开放镜像上传的 API，您可以使用 docker push 或页面方式上传镜像。

8.2.11 如何通过 API 下载镜像？

SWR 暂不支持通过 API 下载镜像。如果要下载镜像，请通过 docker pull 命令来进行下载操作。

8.2.12 docker pull 下载的镜像存放在什么地方？

docker pull 将镜像下载到本地节点上，您可以通过 docker save 命令将镜像保存成 tar 归档文件。

8.2.13 是否支持跨区域下载镜像？

支持。

SWR 当前支持跨区域公网下载，请确保下载机可访问公网且获取正确的仓库登录指令。例如：您想下载“广州 4”区域的镜像，则需要获取“广州 4”区域的镜像仓库登录指令。

8.2.14 控制台页面的镜像可以下载到本地吗？

控制台页面的镜像不能直接下载至本地，您可以参考以下方法操作：

- 在镜像详情页获取镜像的下载指令。

- 在安装了 Docker 客户端的机器上执行上一步的指令下载镜像。
  示例：
  docker pull registry.cn-jssz1.ctyun.cn /group/nginx:v1
- 将镜像保存为.tar 或.tar.gz 格式的文件。
  示例：
  docker save nginx:v1 > nginx.tar

下载文件至本地。

8.3 故障类

8.3.1 为什么创建组织失败？

问题现象：创建组织失败，页面提示该组织已经存在，但在“组织管理”页面没有查询到该组织。

解决办法：组织名称全局唯一，即当前区域下，组织名称唯一。

创建组织时如果提示组织已存在，可能该组织名称已被其他用户使用，请重新设置一个组织名称。

8.3.2 CCE 工作负载拉取 SWR 镜像异常，提示为“未登录”

当 CCE 工作负载无法正常拉取 SWR 的镜像，且提示“未登陆”时，请排查该工作负载的 yaml 文件中是否存在“imagePullSecrets”字段，且 name 参数值需固定为 default-secret。

示例：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - image: nginx
        imagePullPolicy: Always
        name: nginx
      imagePullSecrets:
      - name: default-secret
```

8.3.3 为什么登录指令执行失败？

登录指令执行失败有以下几种情况：

1. 容器引擎未安装正确，报如下所示错误：

   “docker: command not found”

   解决方法：重新安装 docker 容器引擎。

   - 由于容器镜像服务支持容器引擎 1.11.2 及以上版本上传镜像，建议下载对应版本。
   - 安装容器引擎需要连接互联网，内网服务器需要绑定弹性公网 IP 后才能访问。

2. 临时登录指令已过期或登录指令中区域项目名称、AK、登录密钥错误，报如下所示错误：

   “unauthorized: authentication required”

   解决方法：登录容器镜像服务控制台，在左侧菜单栏选择“我的镜像”，单击右侧“客户端上传”获取登录指令。

   a. 获取临时的登录指令：单击“生成临时登录指令”，在弹出的页面中单击 复制登录指令。

   b. 获取长期有效的登录指令：单击“如何获取长期有效登录指令”，具体方法请参见用户指南 > 镜像管理 > 获取长期有效登录指令。

3. 登录指令中镜像仓库地址错误，报如下所示错误：

   “Error llgging in to v2 endpoint, trying next endpoint: Get https://{{endpoint}}/v2/: dial tcp: lookup {{endpoint}} on xxx.xxx.xxx.xxx:53 : no such host”

   解决方法：

   c. 修改登录指令中的镜像仓库地址。

   镜像仓库地址格式 : registry.区域项目名称.ctyun.cn，如“苏州”对应的镜像仓库地址为 registry.cn-jssz1.ctyun.cn。

   d. 获取临时登录指令。

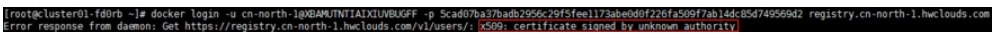
4. **x509: certificate has expired or is not yet valid**

   长期有效登录指令中 AK/SK 被删除导致，请使用有效的 AK/SK 生成登录指令。

5. **x509: certificate signed by unknown authority**

   问题原因：

   容器引擎客户端和 SWR 之间使用 HTTPS 的方式进行通信，客户端会对服务端的证书进行校验。如果服务端证书不是权威机构颁发的，则会报如下错误：x509: certificate signed by unknown authority

   解决方法：

   如果用户信赖服务端，跳过证书认证，那么可以手动配置 Docker 的启动参数，配置方法如下：

   - CentOS：

修改“/etc/docker/daemon.json”文件（如果没有，可以手动创建），在该文件内添加如下内容：

```
{
  "insecure-registries": ["{镜像仓库地址}"]
}
```

- Ubuntu：

  修改“/etc/default/docker”文件，在 DOCKER_OPTS 配置项中增加如下内容：

  ```
  DOCKER_OPTS="--insecure-registry {镜像仓库地址}"
  ```

- EulerOS：

  修改“/etc/sysconfig/docker”文件，在 INSECURE_REGISTRY 配置项中增加如下内容：

  ```
  INSECURE_REGISTRY='--insecure-registry {镜像仓库地址}'
  ```

说明

镜像仓库地址支持域名和 IP 形式。

- 域名：registry.区域项目名称.ctyun.cn。例如“苏州”的镜像仓库地址为：registry.cn-jssz1.ctyun.cn。
- IP：可通过 ping 镜像仓库地址（域名形式）获取。

配置完成后，执行 **systemctl restart docker** 重启容器引擎。

8.3.4 为什么使用客户端上传镜像失败？

denied: you do not have the permission

问题现象：使用客户端上传镜像，报如下所示错误：

“denied: you do not have the permission”

问题原因：

- 该组织名已被其他用户注册或当前 SWR 组织数量已超过配额。
- docker login 命令使用 IAM 用户的 AK、SK 生成，没有对应组织的权限。

解决方法：

- 该组织名已被其他用户注册时：建议您先创建组织然后再上传镜像。
- SWR 组织数量超过配额时：单个用户的组织数量限制为 5 个，您可以将镜像上传到已存在的组织下，也可以提交工单申请增加配额。
- 没有对应组织的权限：使用帐号授权后，可以正常推送。

denied: Image organization does not exist, you should create it first

问题现象：使用客户端上传镜像，报如下所示错误：

“denied: Image organization does not exist, you should create it first”

问题原因：docker push 命令中的组织名称不存在。

解决方法：请先创建组织，再重新上传。

tag does not exist: xxxxxx 或 An image does not exist locally with the tag: xxxxxx

问题现象：使用客户端上传镜像，报如下所示错误：

“tag does not exist: xxxxxx”

或

“An image does not exist locally with the tag: xxxxxx”

问题原因：上传的镜像或镜像版本不存在。

解决方法：通过 docker images 查看本地镜像，确认要上传的镜像名称及版本后，重新上传镜像。

name invalid: 'repository' is invalid

问题现象：使用客户端上传镜像，报如下所示错误：

“name invalid: 'repository' is invalid”

问题原因：组织命名或镜像命名不规范。

解决方法：以下分别是组织名（namespace）和仓库名（repository）的命名正则表达式：

namespace：^([a-z]+(?:(?:(?:_|__|[-]*)[a-z0-9]+)+)?)$，长度范围为：1-64；

repository：^([a-z0-9]+(?:(?:(?:_|__|[-]*)[a-z0-9]+)+)?)$，长度范围为：1-128。

您可以按照上述命名规范，重新指定上传的组织和镜像名称。

errorDetail":{"message":"error parsing HTTP 404 response body: no error details found in HTTP response body

问题现象：使用客户端上传镜像，报如下所示错误：

{"errorDetail":{"message":"error parsing HTTP 404 response body: no error details found in HTTP response body: "{\"code\":404,\"message\":\"Not Found\"}""},"error":"error parsing HTTP 404 response body: no error detail s found in HTTP response body: "

问题原因：连接不上镜像仓库，网络不通。

解决办法：请检查网络后重试。

- 内网访问

  当您使用的安装容器引擎的客户端为云上的 ECS 或 CCE 节点，且机器与容器镜像仓库在同一区域时，上传下载镜像走内网链路。您无需进行任何访问配置，直接访问 SWR 即可。

- 公网访问

该场景下安装容器引擎的机器为云上的 ECS 或 CCE 节点，机器与容器镜像仓库不在同一区域，上传下载镜像走公网链路，机器需要绑定弹性公网 IP。

8.3.5 为什么通过页面上传镜像失败？

SWR 对镜像的命名和地址有严格的规范。如果镜像的命名不规范或镜像地址不规范都会导致镜像上传失败。

镜像格式不合法或鉴权失败

问题现象：通过页面上传镜像，出现“镜像格式不合法”或“鉴权失败”的报错。

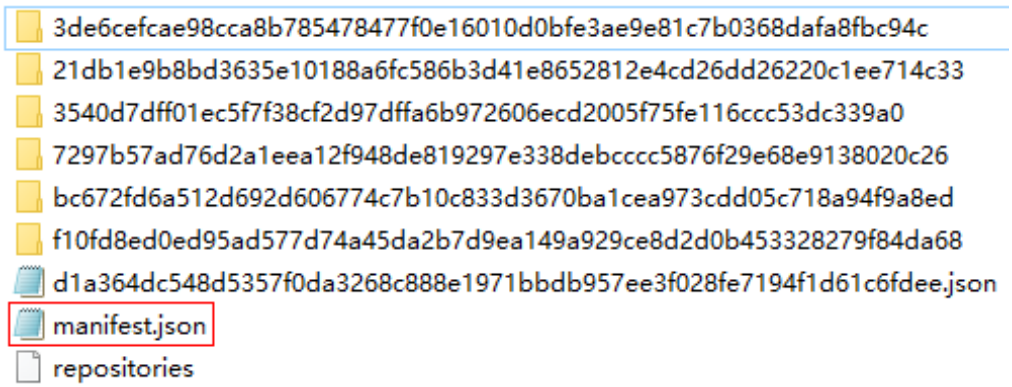
问题原因：镜像地址不规范，导致上传失败。

镜像地址各个部分的含义如下，最后的 tag（版本号）可省略，如果省略则表示 latest 版本，其余部分均不可省略，且不可多余。

样例：registry.cn-jssz1.ctyun.cn/repo_namespace/repo_name:tag

- registry.cn-jssz1.ctyun.cn 为容器镜像服务的镜像仓库地址。
- repo_namespace 为组织名称，命名正则表达式为^([a-z]+(?:(?:(?:_|__|[-]*)[a-z0-9]+)+)?)$，长度范围为：1-64。
- repo_name:tag 为镜像名称和版本号，镜像命名正则表达式为^([a-z0-9]+(?:(?:(?:(?:_|__|[-]*)[a-z0-9]+)+)?)$，长度范围为：1-128。

您可以将镜像解压，打开文件 manifest.json 文件查看 RepoTags 字段的值是否符合上述规范。



解决方法：按照命名规范，重新给镜像打 tag，然后使用 docker save 命令保存镜像，然后再使用页面上传。

须知

SWR 判定镜像名是否合法不是以用户在界面上传镜像时的文件名为依据，而是依据镜像包中的 repositories 和 manifest.json 文件。

一直卡在上传界面直到超时

问题现象：通过页面上传镜像，一直卡在上传界面直到超时。

问题原因：

- 镜像命名不规范，导致上传失败。
- 页面上传走公网链路，可能会因为网络不稳定导致传输异常。

解决方法：

- 您可以按照镜像命名规范修改镜像名称后，重新上传镜像。
- 建议更换网络环境，或者通过容器引擎客户端上传。

8.3.6 为什么 docker pull 指令执行失败？

x509: certificate sigined by unknown authority

问题现象：使用 docker pull 拉取镜像，报错“x509: certificate sigined by unknown authority”。

问题原因：

- 容器引擎客户端和 SWR 之间使用 HTTPS 的方式进行通信，客户端会对服务端的证书进行校验。如果客户端安装的根证书不完整，会报如下错误：“x509: certificate sigined by unknown authority”。
- 容器引擎客户端配置了 Proxy 导致。

解决方法：

- 如果您信赖服务端，跳过证书认证，那么可以手动配置容器引擎的启动参数，配置如下（其中地址配置成需要的即可，选择一个配置即可）：
  - /etc/docker/daemon.json（如果没有可以手动创建），在该文件内添加如下配置（注意缩进，2 个空格）：

    ```
    {
      "insecure-registries":["镜像仓库地址"]
    }
    ```

  - /etc/sysconfig/docker：

    ```
    INSECURE_REGISTRY='--insecure-registry=镜像仓库地址'
    ```

  添加配置后执行如下命令重启：**systemctl restart docker** 或 **service restart docker**。
- 执行 **docker info** 命令，检查 Proxy 配置是否正确，修改为正确的 Proxy 配置。

Error response from daemon

问题现象：使用 docker pull 拉取官方镜像，如 **docker pull elasticsearch:5** 时，报错“Error response from daemon: Get https://registry-1.docker.io/v2/: net/http: request canceled while waiting for connection (Client.Timeout exceeded while awaiting headers)”。

问题原因：Docker 默认的源为国外官方源，镜像拉取速度较慢，可能因超时而拉取失败。

解决方法：将 Docker 镜像源修改为容器镜像服务或国内镜像源地址。

Error: remote trust data does not exist

问题现象：使用 docker pull 拉取镜像，报错“Error: remote trust data does not exist”。

问题原因：客户端开启镜像签名验证，而镜像没有镜像签名层。

解决方法：查看“/etc/profile”文件中的环境变量是否设置了 **DOCKER_CONTENT_TRUST=1**，如果设置了，请将其改为 **DOCKER_CONTENT_TRUST=0**，然后执行 **source /etc/profile** 生效。

8.3.7 如何解决内网下载镜像失败？

内网下载镜像失败基本上都是由于 DNS 配置问题导致的，可以采用以下两种方法进行修改。

- 方法一：

  编辑“/etc/resolv.conf”文件，在/etc/resolv.conf 中新增一个内网 DNS 服务器地址，具体的内网 DNS 地址请参考云主机所在 VPC 子网的 DNS 设置。

  说明

  新增的 DNS 服务器地址必须位于所有原有的 DNS 服务器地址之前。

  DNS 配置操作在保存“/etc/resolv.conf”文件的修改操作后立即生效。

- 方法二：

  “/etc/resolv.conf”文件的修改操作在弹性云主机重启后会失效，需要重新进行配置。如果用户不希望每次重启弹性云主机后都重新配置 DNS，可以按如下步骤修改虚拟私有云的子网信息，将 DNS 服务器地址添加到弹性云主机对应的子网中。

须知

对虚拟私有云的子网信息的修改会影响所有使用该子网创建的弹性云主机。

e. 登录管理控制台，选择“网络 > 虚拟私有云”。

f. 修改 VPC 子网的 DNS 服务器地址。

g. 重启弹性云主机，查看“/etc/resolv.conf”文件的内容，确认其中包含待配置的 DNS 服务器地址，并且新增的 DNS 服务器地址位于其他 DNS 服务器地址之前。