Container Registry

User Guide

User Guide

Container Registry

Build a repository

Overview

In full compliance with the open-source container technology standards, Alibaba Cloud Container Registry provides the ability of building images based on Dockerfiles. This service eases the entire process from code repository to container application deployment, allowing you to containerize your services quickly and efficiently.

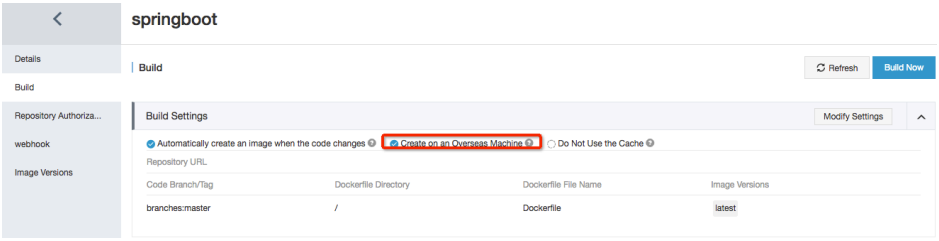
Features

Container Repository supports automatically triggering the build when the code changes.

If **Automatically create an image when the code changes** is selected in **Build Settings**, the image can be automatically built after you submit the code, without requiring you to manually trigger the build.

Container Repository supports overseas build.

You might depend on overseas sources when building your code. However, considering the network environment, Container Repository supports building an image on an overseas machine. Then, the image built on an overseas machine is pushed back to the repository in the specified region.



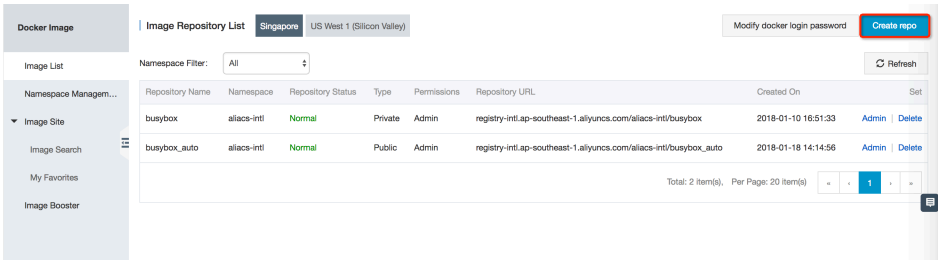
Container Repository supports multi-stage build.

Alibaba Cloud Container Repository also supports the latest multi-stage build characteristics.

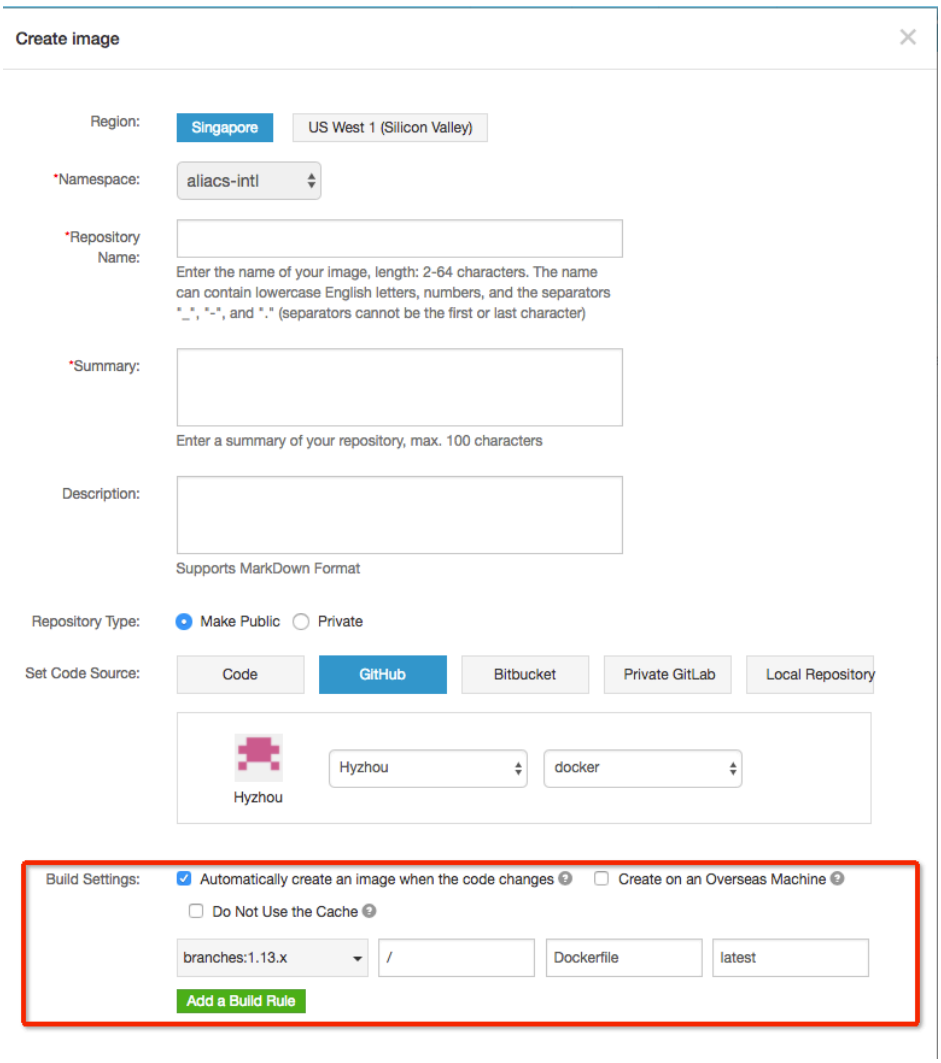
Set build rules

1. Log on to the Container Registry console.

   Click **Create repo** in the upper-right corner.



   Set the build rules.



- **Automatically create an image when the code changes:** With this check box selected, the build rule is automatically triggered when code is submitted from a branch.
- **Create on an Overseas Machine:** With this check box selected, the image is built on an overseas machine and then pushed back to the repository in the specified region.

  Note: Sometimes the network between China Mainland and abroad is not stable, the image might fail to be pushed back because of timeout.

- **Do Not Use the Cache:** With this check box selected, the dependent base image is pulled again each time an image is built, which might slow down the build time.
- **Select branch or tag:** Set the code branch for the build.
- **Dockerfile directory:** Set the directory where the Dockerfile is located.

  Note: The directory here refers to a relative directory, which uses the root directory of the code branch as the parent directory.

- **Dockerfile name:** Set a name for the Dockerfile. The default name is Dockerfile.
- **Image version:** Set an image tag, such as **latest**.

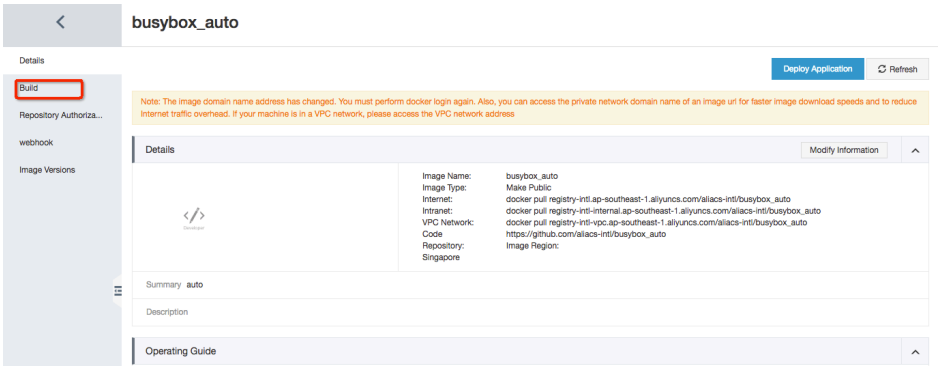
Click **Create repo** in the lower right corner to create the image repository.

Modify build rules

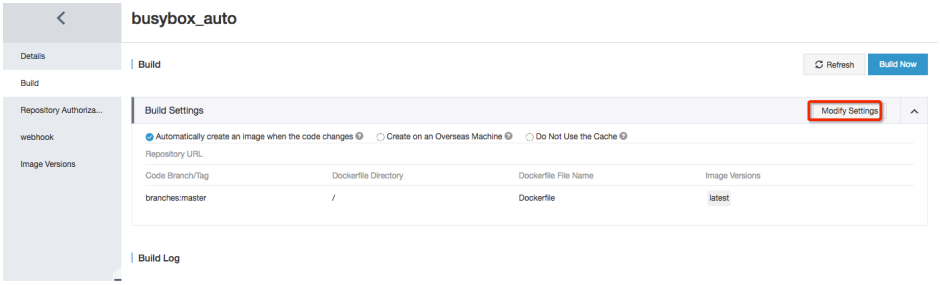
Log on to the Container Registry console.

Click **Manage** at the right of an image repository to enter the repository details page.

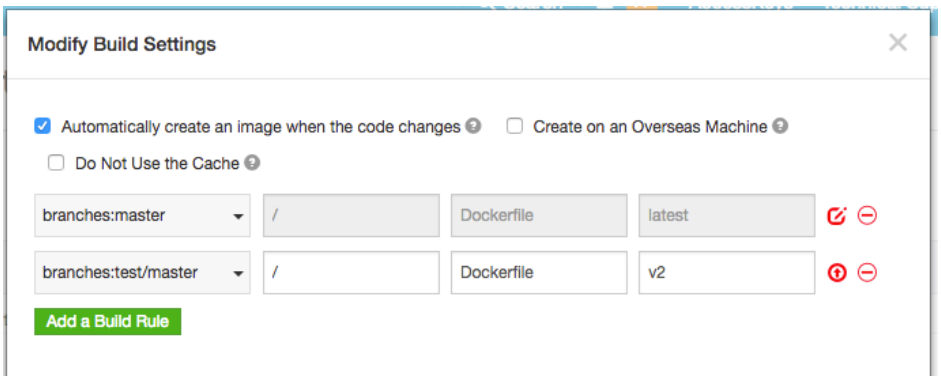
Click **Build** in the left-side navigation pane to enter the image build details page.



Click **Modify Settings** on the right to modify the build rules.



Modify and set the build rules.



- **Automatically create an image when the code changes:** With this check box selected, the build rule is automatically triggered when code is submitted from a branch.
- **Create on an Overseas Machine:** With this check box selected, the image is built on an overseas machine and then pushed back to the repository in the specified region.

  > **Note:** Sometimes the network between China Mainland and abroad is not stable, the image might fail to be pushed back because of timeout.

- **Do Not Use the Cache:** With this check box selected, the dependent base image is pulled again each time an image is built, which might slow down the build time.
- **Add a Build Rule:** Create a build rule.
- : Delete this build rule.

Click **OK** to finish the modification.

Build an image

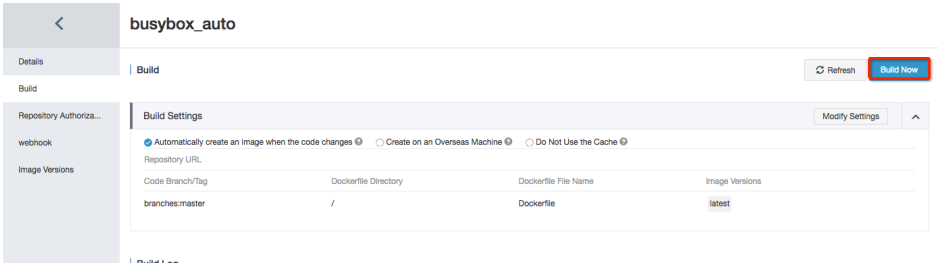
Log on to the Container Registry console.

Select the region.

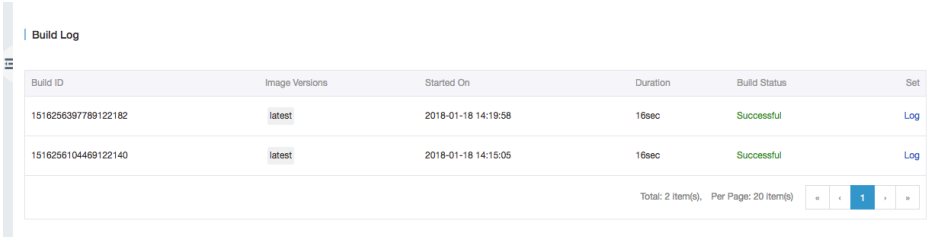
Click **Manage** at the right of an image repository to enter the repository details page.

Click **Build** in the left-side navigation pane.

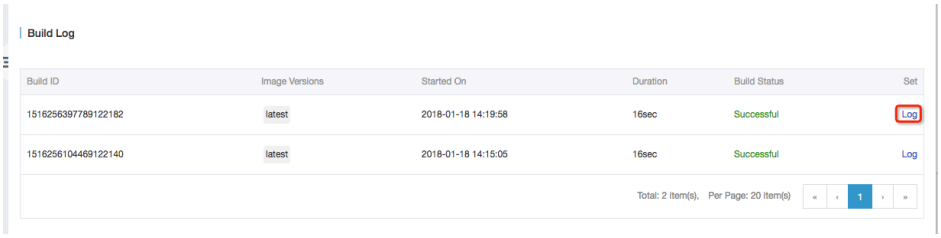
Click **Build Now** in the upper right corner to build an image based on the build rules you set.



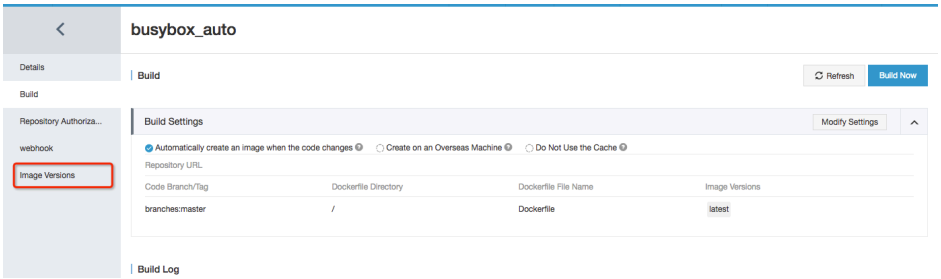
A new build record is generated after you click **Build Now**.



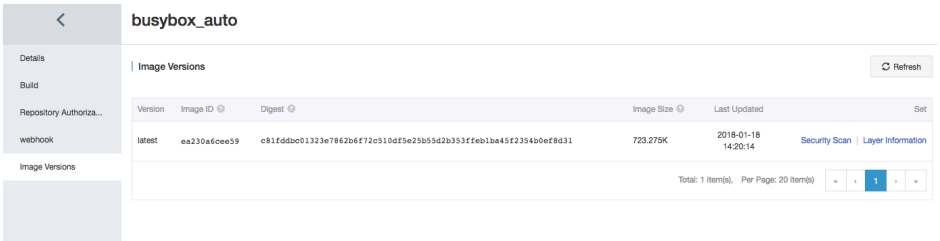
Click **Log** at the right of the build record to view the build logs.



After the image is built, click **Image Versions** in the left-side navigation pane to view the list of built images.



View the list of all the image versions.



Basic use of a namespace

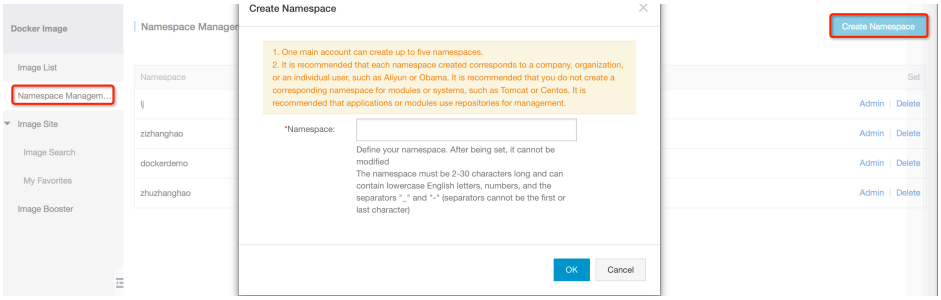
Best practices for a namespace

- A namespace is a collection of repositories. We recommend that you group the repositories of a company or organization in one namespace.
    - Use company name as the namespace: aliyun, alibaba
    - Use team or organization as the namespace: misaka-team

Main features of a namespace

Create a namespace.

Click **Create Namespace** and enter the namespace name.



Currently, a primary account can create five namespaces.

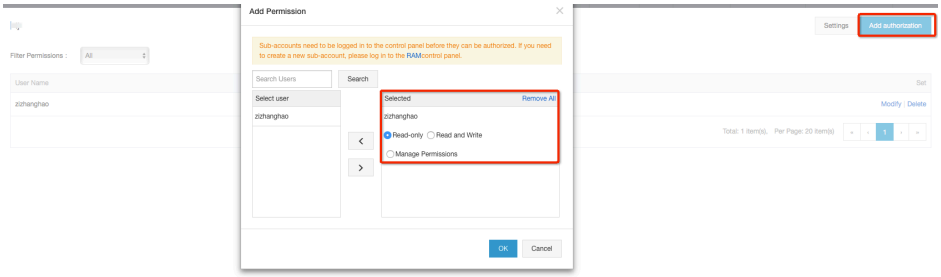
Authorize a namespace.

- Grant the permissions for a namespace to sub-accounts. The authorization applies to all the repositories in the namespace.

  The sub-accounts must first log on to the console to set the Container Registry logon password.

    - READ: The read-only permission, with which you can only pull the images of the repositories in the namespace.
    - WRITE: The write permission, with which you can push the images of the repositories in the namespace.
    - ADMIN: The administrator permission, with which you can manage a

namespace.



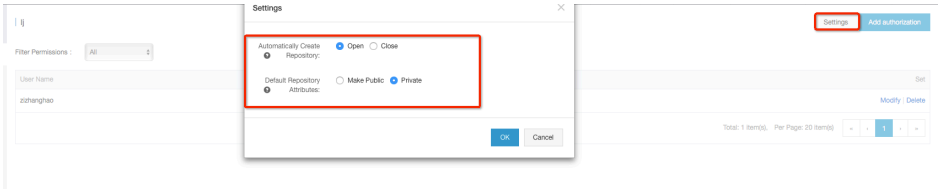
Configure a namespace.

By default, the service allows you to directly push images, and the system automatically creates a repository according to the repository name.

You can disable the automatic creation by selecting **Close** for **Automatically Create Repository**.

Currently, the repositories automatically created by the service for pushing images are private by default.

To make the automatically created repositories public by default, you can select **Make Public** for **Default Repository Attributes**.



Basic use of a repository

Best practices for a repository

- A repository is a collection of images. We recommend that you group all versions of images for an application or feature in one repository.
    - Use software package as the repository name: centos, jetty
    - Use application name as the repository name: console-web, console-service

Main features of a repository

- Repository visibility settings
    - Set the repository as a public one. Then, the repository is open and allows all users to download anonymously.
    - Set the repository as a private one. Then, only the accounts with permissions can log on to view and download from the repository.
- Image deployment
    - Click **Deploy** in the repository to go to Container Service for deployment.
- Repository image query
    - List the images in a repository and obtain the Digest and ImageId of images.
    - Check the layer information of the image, including the size and metadata for each layer of the image.
    - Perform image security scan to identify potential vulnerabilities in the image and provide solutions for some of them.
- Webhook
    - Enable the images in a repository to trigger notifications. After an image is uploaded, the access address set by you is automatically triggered.
    - Sequentially connect the downstream processes of an image service.
- Repository authorization
    - Grant the permissions for a repository to sub-accounts. The sub-accounts must first log on to the console to set passwords.
        - READ: The read-only permission, with which you can only pull the images.
        - WRITE: The write permission, with which you can push the images.
        - ADMIN: The administrator permission, with which you can manage a repository.
- Image build service
    - Manage your source code repositories. After the code is submitted, an image is built according to the build rules you set and then pushed to your repository.
    - Sequentially connect the upstream processes of an image service.

Repository access control

Overview

Alibaba Cloud’ s permission management includes Resource Access Management (RAM) and Security Token Service (STS). By default, the primary account has full operational authority over its own resources. RAM and STS enable users to access image resources using different subaccounts

with different permissions and also grant users temporary access authorization. Using RAM and STS can improve management flexibility and security.

For more information about how to configure authorization policies, see RAM documentation.

System configuration policies

AliyunContainerRegistryFullAccess

Grant subaccount full access permissions, the subaccount then functions the same as the primary account and can perform any operations.

```
{
"Statement": [
{
"Action": "cr:*",
"Effect": "Allow",
"Resource": "*"
}
],
"Version": "1"
}
```

AliyunContainerRegistryReadOnlyAccess

Grant a subaccount the read-only permission, the subaccount then is allowed to perform read-only operations. For example, view repository list, pull images, and so on.

```
{
"Statement": [
{
"Action": [
"cr:Get*",
"cr:List*",
"cr:PullRepository"
],
"Effect": "Allow",
"Resource": "*"
}
],
"Version": "1"
}
```

Policy configuration scenarios

Scenario 1

Scenario description: Authorize a subaccount with read-only permission of namespace (for example, juzhong). Subaccount can view the namespace information, and all relative information about the image repository. Logon to the Container Registry also allows subaccount to pull the images.

```
{
"Statement": [
{
"Action": [
"cr:Get*",
"cr:List*",
"cr:PullRepository"
],
"Effect": "Allow",
"Resource": [
"acs:cr:*:*:repository/juzhong/*",
"acs:cr:*:*:repository/juzhong"
]
}
],
"Version": "1"
}
```

Scenario 2

Scenario description: Authorize a subaccount for a certain image repository (for example, the image repository name is nginx which belongs to a namespace juzhong, and the affiliated region is China East 1).

```
{
"Statement": [
{
"Action": [
"cr:*"
],
"Effect": "Allow",
"Resource": [
"acs:cr:cn-hangzhou:*:repository/juzhong/nginx"
]
},
{
"Action": [
"cr:Get*",
"cr:List*"
],
"Effect": "Allow",
"Resource": [
"acs:cr:*:*:repository/juzhong"
]
}
```

```
],
"Version": "1"
}
```

Note:

While using RAM subaccount, pay special attention to the following instructions to avoid granting excessive permissions to subaccounts.
If you grant an administrative authority for all Alibaba Cloud resources (that is, AdministratorAccess) to a subaccount using RAM, regardless of whether you previously granted the subaccount Container Registry permissions, in this situation the subaccount has full permissions on the Container Registry.

Registry authentication rules

Resource description

When you grant a suaccount authority through RAM, the resource is described as follows:

Resource type	Resource description of authentication policies
repository	acs:cr:$regionid:$accountid:repository/$namespacename/$repositoryname acs:cr:$regionid:$accountid:repository/$namespacename/* acs:cr:$regionid:$accountid:repository/$namespacename

Parameter descripiton:

Name	Descripiton
$regionid	The region ID. You can replace it with asterisks (*).
$accountid	The cloud account ID. You can replace it with asterisks (*).
$namespacename	The name of the namespace.
$repositoryname	The name of the repository.

Authentication rules

When you access Container Registry APIs through a subaccount or STS, Container Registry checks the caller’ s permissions on RAM to make sure that the caller has the corresponding permissions. Each API determines the resources for permission check according to the involved resources and the semantics of the API. The following table lists the authentication rules for each API:

API	Authentication action	Authentication resource
Create a namespace	No authenticaion is required.	No authenticaion is required.
Delete a namespace	cr:DeleteNamespace	acs:cr:$regionid:$accountid:repository/$namespacename
Update the namespace information	cr:UpdateNamespace	acs:cr:$regionid:$accountid:repository/$namespacename
Get the specified namespace information	cr:GetNamespace	acs:cr:$regionid:$accountid:repository/$namespacename
Get a namespace list	cr:ListNamespace	*
Create a repository	cr:CreateRepository	acs:cr:$regionid:$accountid:repository/$namespacename/$repositoryname
Delete a repository	cr:DeleteRepository	acs:cr:$regionid:$accountid:repository/$namespacename/$repositoryname
Update the repository information	cr:UpdateRepository	acs:cr:$regionid:$accountid:repository/$namespacename/$repositoryname
Get the repository information	cr:GetRepository	acs:cr:$regionid:$accountid:repository/$namespacename/$repositoryname
Get a repository list	cr:ListRepository	*
Get a repository list by namespace	cr:ListRepository	*
Get the repository tags information	cr:ListRepositoryTag	acs:cr:$regionid:$accountid:repository/$namespacename/$repositoryname
Delete an image version	cr:DeleteRepositoryTag	acs:cr:$regionid:$accountid:repository/$namespacename/$repositoryname
Get the image manifest information	cr:GetRepositoryManifest	acs:cr:$regionid:$accountid:repository/$namespacename/$repositoryname
Get the image layer information	cr:GetRepositoryLayers	acs:cr:$regionid:$accountid:repository/$namespacename/$repositoryname
Get a temporary token	cr:GetAuthorizationToken	*
Pull image	cr:PullRepository	acs:cr:$regionid:$accountid:repository/$namespacename/$repositoryname
Push image	cr:PushRepository	acs:cr:$regionid:$accountid:repository/$namespacename/$repositoryname

Image security scan

Container Registry allows you to perform security scan on some fundamental Linux-based images. You can scan your images with only one click. The scan time varies depending on the image size. Generally, it takes less than three minutes to scan one image.

Currently, only some Linux-based images are supported, and only images built based on the following base images can obtain the scan results.

- **Ubuntu: 12.04 or later**
- **RedHat: 5, 6, and 7**
- **CentOS: 5, 6, and 7**
- **Oracle: 5, 6, and 7**
- **Debian: 7, 8, 9, and 10**
- **Alpine: 3.3, 3.4, 3.5, and 3.6**

Procedure

Log on to the **Container Registry console**.

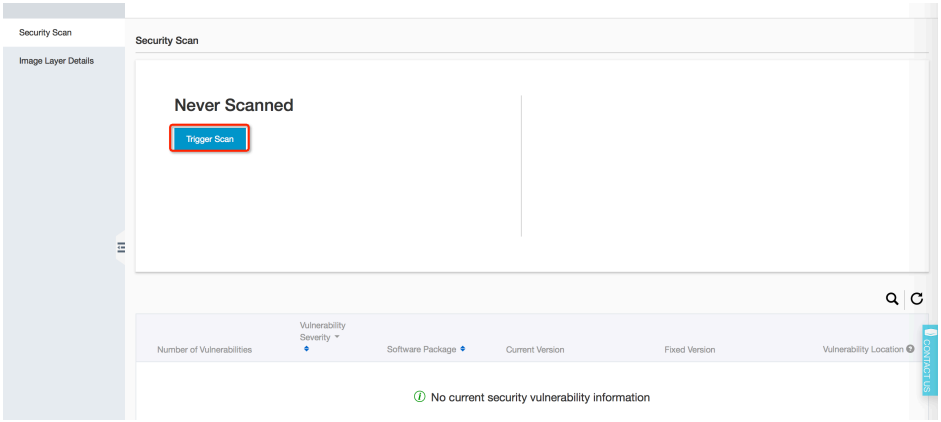
Select the region.

Click **Manage** at the right of an image repository to enter the repository details page.

Click **Image Versions** in the left-side navigation pane.

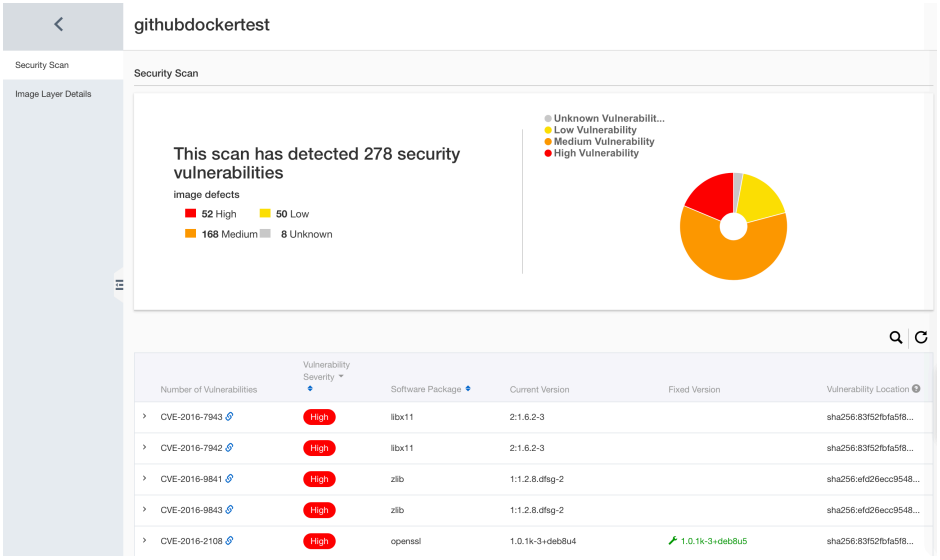
Click **Security Scan** at the right of the image version.

Trigger the security scan of the security image. After a moment, an image vulnerability report is displayed.



Vulnerability reports for image security scan

After an image security scan is completed, a vulnerability report is generated as follows.



Vulnerability information is categorized into four levels: **High**, **Medium**, **Low**, and **Unknown**. Vulnerability details and the corresponding bug fix versions are displayed.

Webhook

Overview

Alibaba Cloud Container Registry provides a Webhook for each repository, allowing you to push a notification when an image is built and therefore set up a continuous integration pipeline. Suppose

you have set a Container Service trigger for Webhook. When an image is built, the applications in Container Service are automatically triggered to pull the latest image and re-deployed.

Currently, Alibaba Cloud Container Registry provides two types of Webhook triggers: expression-based and tag-based. Expression-based trigger filters tags based on a regular expression, and only the tags matching the regular expression can trigger the Webhook. Tag-based trigger is based on a user-filtered tag list. If no triggers are set for a Webhook, the trigger is both expression-based and tag-based by default.

Example

1. Expression-based trigger

Enter a simple regular expression such as release-v.*. Then, only when an image whose version tag begins with release-v is built can the subsequent continuously integrated process be triggered. Otherwise, no trigger occurs, and the access history shows **Not Triggered** in the **Access Status Code** column.

Click **Access History** to view the access history of a Webhook.

2. Tag-based trigger

You can select a maximum of ten tags to be triggered from the list. Only tags in the list can trigger a Webhook when an image is built. Otherwise, no trigger occurs, and the access history shows **Not Triggered** in the **Access Status Code** column.

Click **Access History** to view the access history of a Webhook.

Notification contents

A Webhook notification contains the image repository information and image version information as follows. Image repository information includes the namespace, name, and region of the repository.

```
POST /payload HTTP/1.1

Content-Type: application/json
Request URL:
https://cs.console.aliyun.com/hook/trigger?triggerUrl=YzRmMWE5YzM2ZjMzYzQ0NmFiMGYzNWJlMmM2MjM2NzIyfGV4cHJlc3N8cmVkZXBsb3l8MThlMmllY2drdXYyZXw=&secret=365a4a664b45615438716a487a75695a7ac48329224b35b073c2197374e7d62a
Request method: POST

{
"push_data": {
"digest": "sha256:457f4aa83fc9a6663ab9d1b0a6e2dce25a12a943ed5bf2c1747c58d48bbb4917",
"pushed_at": "2016-11-29 12:25:46",
```

```
"tag": "latest"
},
"repository": {
"date_created": "2016-10-28 21:31:42",
"name": "repoTest",
"namespace": "namespace",
"region": "cn-hangzhou",
"repo_authentication_type": "NO_CERTIFIED",
"repo_full_name": "namespace/repoTest",
"repo_origin_type": "NO_CERTIFIED",
"repo_type": "PUBLIC"
}
}
```


Container Registry Enterprise Edition