

Alibaba Cloud Container Service

クイックスタート

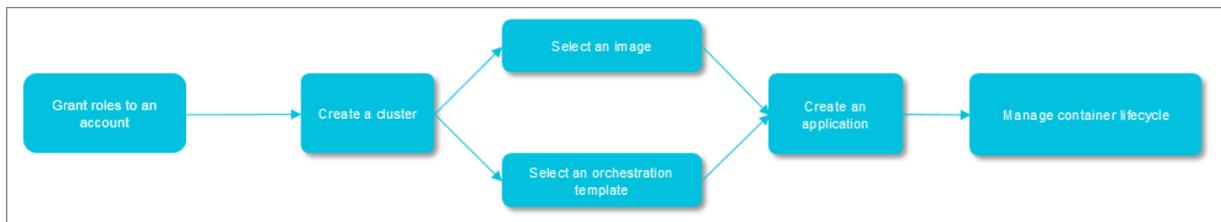
Document Version20190629

目次

1 ワークフロー.....	1
2 イメージからの Nginx Web サーバーの作成.....	2
3 オーケストレーションテンプレートによる WordPress の作成.....	6
4 Docker ツールによるクラスターへの接続.....	10

1 ワークフロー

Container Service のワークフローの全体像は以下になります。



手順 1: アカウントに対するロールの付与

詳しくは、「[ロールの権限付与](#)」をご参照ください。

手順 2: クラスターの作成

クラスターのネットワーク環境を選択、ノード数およびクラスターの構成を設定できます。

手順 3: イメージまたはオーケストレーションテンプレートによるアプリケーションの作成

既存のイメージまたはオーケストレーションテンプレートを選択するか、新しいイメージまたはオーケストレーションテンプレートを作成します。

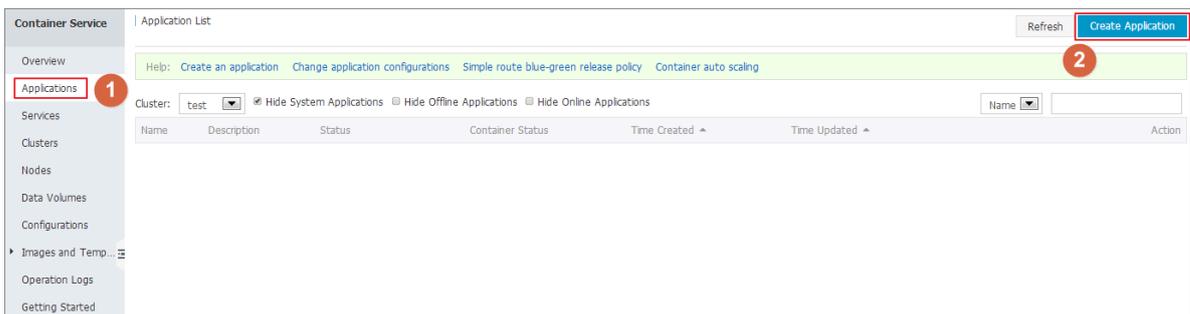
お使いのアプリケーションが複数のイメージによりサポートされるサービスで構成される場合、オーケストレーションテンプレートを使用してアプリケーションを作成します。

手順 4: デプロイ後のアプリケーションステータスおよび関連するサービスとコンテナの情報の確認

2 イメージからの Nginx Web サーバーの作成

これまでにクラスターを作成していない場合は、まずクラスターを作成します。クラスターの作成について詳しくは、「[クラスターの作成](#)」をご参照ください。

1. [Container Service コンソール](#)にログインします。
2. 左側のナビゲーションウィンドウから [アプリケーション] をクリックし、右上の [アプリケーションの作成] をクリックします。



3. これから作成するアプリケーションに関する以下の設定を完了させ、[イメージによる作成] をクリックしします。

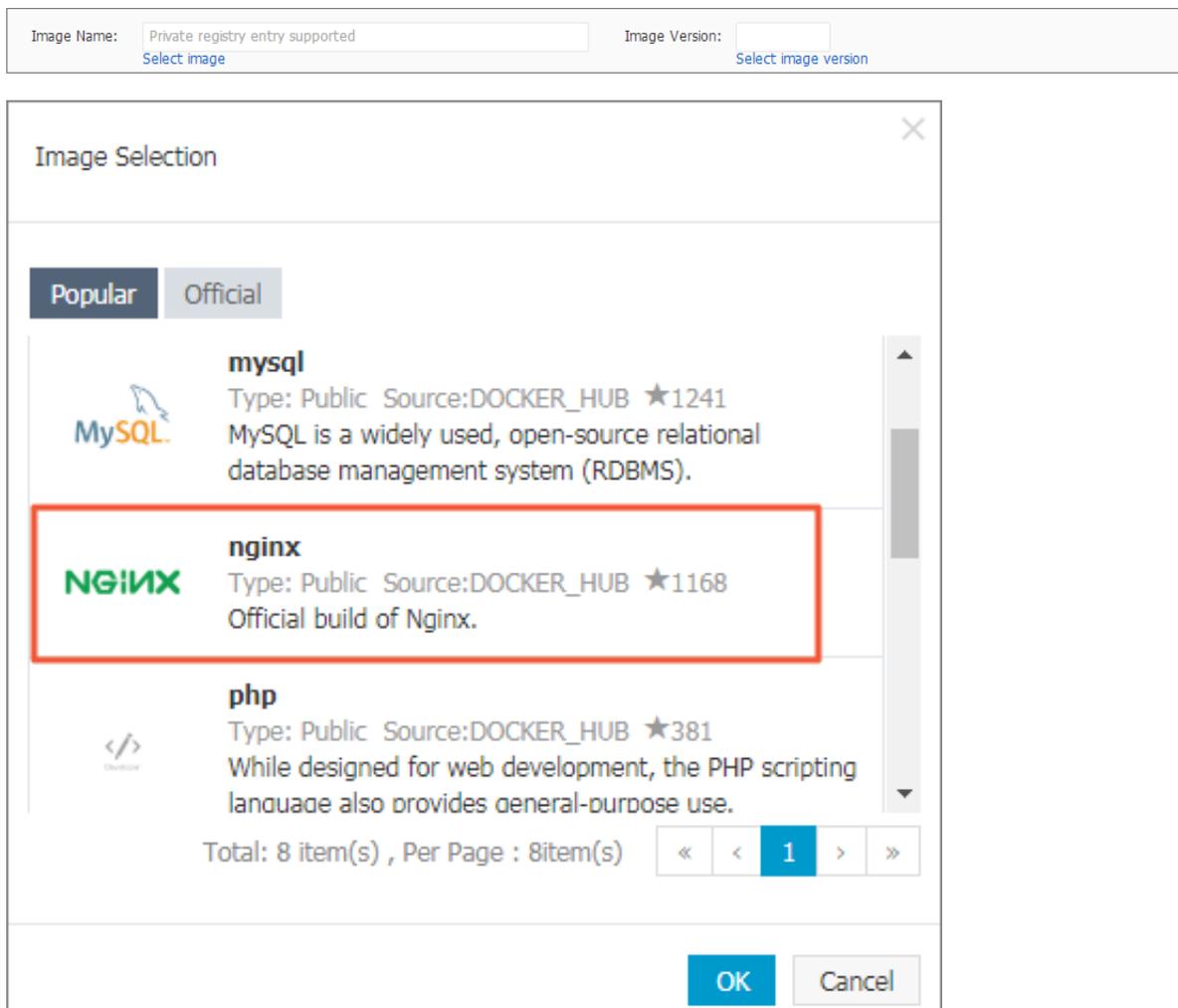
- ・ **名前:** アプリケーションの名前を入力します。このページの例では、「nginx」と入力します。
- ・ **バージョン:** このアプリケーションのバージョンを入力します。デフォルトでは、1.0 が入力されます。
- ・ **クラスター:** アプリケーションをデプロイするクラスターを選択します。
- ・ **更新:** アプリケーションの更新方法です。標準リリースまたは、Blue - Green リリースを選択します。詳しくは、[リリース方法の紹介](#)をご参照ください。
- ・ **説明:** アプリケーションの情報を入力します。入力した説明はアプリケーションリストのページに表示されます。
- ・ **Docker イメージの取り込み:** このチェックボックスをオンすると、イメージのタグが変更されていなくても、Container Service により最新の Docker イメージがリポジトリから取り込まれ、アプリケーションがデプロイされます。

効率を向上されるため、Container Service によりイメージはキャッシュされます。アプリケーションのデプロイ時、イメージのタグがローカルキャッシュのイメージのタグと同一の場合、Container Service により、リポジトリからイメージを取り込む代わりに、キャッシュされたイメージが使用されます。そのため、コードおよびイメージを変更した際に、業務の都合上イメージのタグを変更しなかった場合、Container Service はローカ

ルにキャッシュされた古いイメージを使用してアプリケーションをデプロイします。このチェックボックスをオンにすると、Container Service はキャッシュされたイメージを無視します。また、最新のイメージとコードが常に使用されるように、アプリケーションのデプロイ時に Container Service はリポジトリから再びイメージを取り込みます。

4. [イメージの選択] をクリックします。[基本] タブの "Nginx" を選択し、[OK] をクリックします。

デフォルトでは、Container Service により最新バージョンのイメージが使用されます。他のバージョンを使用する場合、[イメージバージョンの選択] をクリックします。[イメージバージョン] ダイアログボックスが表示されます。バージョンを選択し、[OK] をクリックします。



5. [ポートマッピング] で、ホストおよびコンテナのポートマッピングを設定します。インターネットによるコンテナ上の Nginx サーバーへのアクセスを可能にするために、[Web ルート] を設定します。

a) [ポートマッピング] の隣にある "+" アイコンをクリックし、コンテナポートフィールドに、それぞれ「80」および「443」を入力し、2つのポートマッピングを追加します。このページの例では、ホストポートは指定されません。

b) Web ルートを設定します。

- ・ [Web ルート] の隣にある "+" アイコンをクリックします。
- ・ [コンテナポート] フィールドに「80」と入力します。これは、Nginx コンテナの 80 番ポートを示します。
- ・ [ドメイン] フィールドに「nginx」と入力します。ドメイン名のプリフィックスである、`nginx` のみ入力されます。ドメイン名のプリフィックスが、`XXX` の場合、以下のようなドメイン名が取得できています。

`XXX.$cluster_id.$region_id.alicontainer.com` これはテスト用です。

このページの例では、以下のようなテスト用のドメイン名を取得します。 `nginx`

`.c9b424ed59 1eb4892a2d 18dd264a6f dfb . cn - hangzhou .`

`alicontainer.com`

Port Mapping: [Add domain names to services exposed to the public network](#)

Host Port	Container Port	Protocol
e.g. 8080	80	TCP
e.g. 8080	443	TCP

The host port cannot be set to 9080,2376,3376

Web Routing: [Expose HTTP services through acsrouting](#)

Container Port	Domain
80	nginx

Note: All domain names for a port must be entered in one entry.



注:

独自のドメイン名を入力することもできます。独自ドメイン名の追加方法は、「[シンプルルーティング-ドメイン名の設定](#)」をご参照ください。ルーティングに使用するコンテナポートおよび HTTP サービスのドメイン名の設定方法は、「[ラベルの説明](#)」の「ルーティング」をご参照ください。ルーティングサービスがどのようにリクエストをコンテナに転送するかについては、「[シンプルルーティング-HTTP および HTTPS のサポート](#)」をご参照ください。

- [作成] をクリックします。Container Service により、上記の設定に応じてアプリケーション "nginx" が作成されます。
- アプリケーションの作成が成功したことを示すページが表示されます。[アプリケーションリストの表示] をクリックするか、ページ上の [アプリケーションリストに戻る] をクリックするか、左側のナビゲーションウィンドウから [アプリケーション] をクリックします。アプリケーション名 [nginx] をクリックし、アプリケーションの詳細を表示します。

Name	Description	Status	Container Status	Time Created	Time Updated	Action
nginx		● Ready	Ready:1 Stop:0	2017-03-31 14:13:11	2017-03-31 14:13:11	Stop Update Delete Redeploy Events
wordpress		● Ready	Ready:4 Stop:0	2017-03-31 13:48:31	2017-03-31 13:49:04	Stop Update Delete Redeploy Events

- サービスにあるサービス名 [nginx] をクリックし、サービスの詳細を表示します。

Name	Application	Status	Container Status	Image	Action
nginx	nginx	● Running	Running:1 Stop:0	nginx:latest	Stop Restart Reschedule Update Delete Events

- サービス "nginx" のアクセスエンドポイントをクリックします。Nginx サーバーのデフォルトのウェルカムページが表示されます。

Overview							
Service Name: nginx	Application: nginx	Image: nginx:latest	Number: 1	● Ready			
Access Endpoint: http://nginx.							
Containers							
Name/ID	Status	Health Check	Image	Port	Container IP	Node IP	Action
nginx/nginx_1 71e7acc772a685b...	running	Normal	nginx:latest sha256:5e69fe4b3...	443/tcp 80/tcp	172. ...	172. ...	Delete Stop Monitor Logs Web Terminal



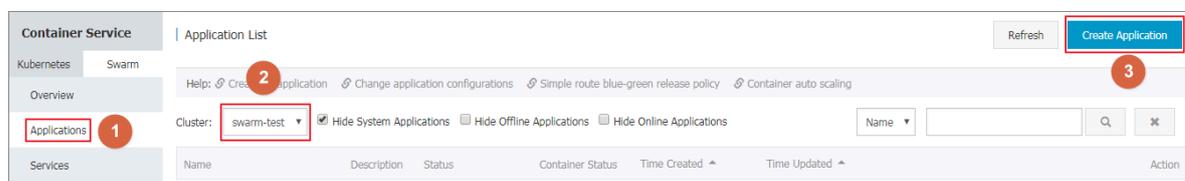
注:

ウェルカムページにアクセスできない場合は、「[どのようにリンク接続問題のトラブルシューティングを行いますか。](#)」をご参照ください。

3 オーケストレーションテンプレートによる WordPress の作成

これまでにクラスターを作成していない場合は、まずクラスターを作成します。クラスターの作成について詳しくは、「[クラスターの作成](#)」をご参照ください。

1. [Container Service コンソール](#)にログインします。
2. 左側のナビゲーションウィンドウから [アプリケーション] をクリックし、右上の [アプリケーションの作成] をクリックします。



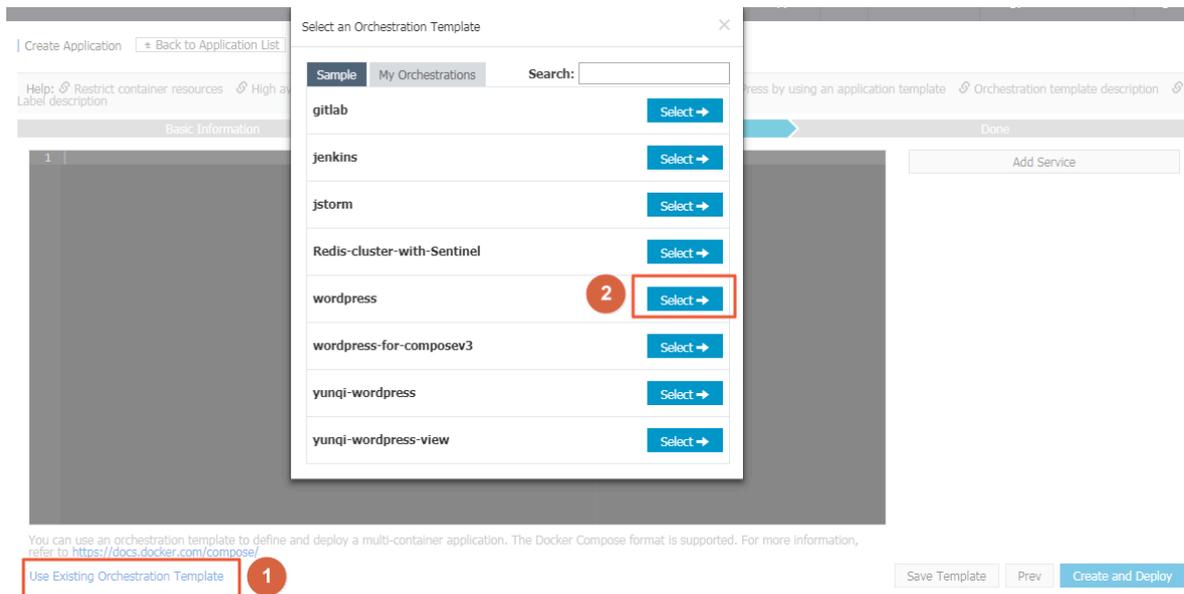
3. これから作成するアプリケーションに関する以下の設定を完了させ、[オーケストレーションテンプレートによる作成] をクリックします。

- ・ **名前:** アプリケーションの名前を入力します。このページの例では、「wordpress-test」と入力します。
- ・ **バージョン:** このアプリケーションのバージョンを入力します。デフォルトでは、1.0 が入力されます。
- ・ **クラスター:** アプリケーションをデプロイするクラスターを選択します。
- ・ **更新:** アプリケーションの更新方法です。[標準リリース]または、[Blue-Green リリース]を選択します。詳しくは、「[リリースポリシーの照会](#)」をご参照ください。
- ・ **説明:** アプリケーションの情報を入力します。入力した説明はアプリケーションリストのページに表示されます。
- ・ **Docker イメージの取り込み:** このチェックボックスをオンすると、イメージのタグが変更されていなくても、Container Service により最新の Docker イメージがリポジトリから取り込まれ、アプリケーションがデプロイされます。

The screenshot shows the 'Basic Information' tab of the Container Service console. The form contains the following fields and options:

- Name:** A text input field containing 'wordpress-test'. Below it, a note states: 'The name should be 1-64 characters long, and can contain numbers, English letters and hyphens, but cannot start with a hyphen.'
- Version:** A text input field containing '1.0'.
- Cluster:** A dropdown menu with 'test' selected.
- Update:** A dropdown menu with 'Standard Release' selected.
- Description:** A large text area for entering application details.
- Pull Docker Image:** A checked checkbox.
- Buttons:** Two buttons at the bottom right: 'Create with Image' and 'Create with Orchestration Template'. The 'Create with Orchestration Template' button is highlighted with a red border.

4. [既存のオーケストレーションテンプレートの使用] をクリックし、WordPress テンプレートの隣の [選択] をクリックします。



5. 対応する設定を変更します。

テンプレートを直接変更するか、[編集] をクリックしたサービスを編集します。

`aliyun . routing . port_80 : http :// wordpress は、 http :// wordpress .$ testDomain` からのリクエストを示し、コンテナが正常に実行されるとコンテナのポート 80 に転送されます。

6. [作成とデプロイ] をクリックします。
7. アプリケーションの作成が成功したことを示すページが表示されます。[アプリケーションリストの表示] をクリックするか、ページ上の [アプリケーションリストに戻る] をクリックするか、左側のナビゲーションウィンドウから [アプリケーション] をクリックします。アプリケーション名 [wordpress-test] をクリックし、アプリケーションの詳細を表示します。

Name	Description	Status	Container Status	Time Created	Time Updated	Action
nginx		Running	Running:1 Stop:0	2017-11-27 10:12:17	2017-11-27 10:12:17	Stop Update Delete Redeploy Events
wordpress-test		Running	Running:4 Stop:0	2017-11-27 11:16:46	2017-11-27 11:16:50	Stop Update Delete Redeploy Events

8. [サービス] にあるサービス名 [web] をクリックし、サービスの詳細を表示します。

Name	Application	Status	Container Status	Image	Action
db	wordpress-test	Running	Running:1 Stop:0	registry.aliyuncs.com/acs-sample/mysql:5.7	Stop Restart Reschedule Update Delete Events
web	wordpress-test	Running	Running:3 Stop:0	registry.aliyuncs.com/acs-sample/wordpress:4.5	Stop Restart Reschedule Update Delete Events

9. サービス "web" のアクセスエンドポイントをクリックします。これはアクセスドメイン名です。

Service: wordpress-test_web Refresh Scale

Overview

Service Name: web Application: wordpress-test Image: registry.aliyuncs.com/acs-sample/wordpress:4.5 Number: 3 ● Running

Access Endpoint: <http://wordpress-9c272e203eb59d44...cn-hangzhou.alicontainer.com>

Containers Logs Configurations Events

Name/ID	Status	Health Check	Image	Port	Container IP	Node IP	Action
wordpress-test_w... 9c272e203eb59d44...	running	Normal	registry.aliyunc... sha256:592af506c...	172.16.33.48:32772->80/tcp	172.16.33.48	172.16.33.48	Delete Stop Monitor Logs Web Terminal
wordpress-test_w... 3e09154752f26a72...	running	Normal	registry.aliyunc... sha256:592af506c...	172.16.33.47:32769->80/tcp	172.16.33.47	172.16.33.47	Delete Stop Monitor Logs Web Terminal
wordpress-test_w... 305e0485f09d7151...	running	Normal	registry.aliyunc... sha256:592af506c...	172.16.33.48:32773->80/tcp	172.16.33.48	172.16.33.48	Delete Stop Monitor Logs Web Terminal



注:

- ・このページのドメイン名はテストでのみ使用します。お使いのドメイン名をバインドします。
- ・エンドポイントにアクセスできない場合は、[どのようにリンク接続問題のトラブルシューティングを行いますか。](#) この問題のトラブルシューティングをご参照ください。

4 Docker ツールによるクラスターへの接続

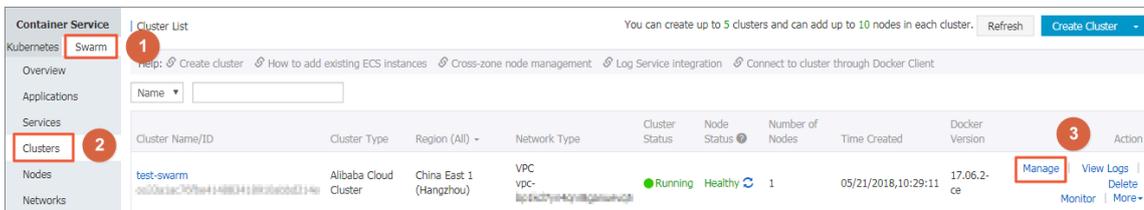
Container Service は [Docker Swarm API](#) と完全な互換性があります。Docker クライアントおよび Docker Compose などの、Docker ツールのコマンドを使用して、Docker クラスターへのアクセスおよび管理が行えます。

証明書のインストール

1. アクセスアドレスを取得します。

a. [Container Service コンソール](#)にログインします。

b. Swarm で、左側のナビゲーションウィンドウから [クラスター] をクリックします。クラスターリストで、クラスターの右側の [管理] をクリックします。



表示されるクラスターの接続情報は、以下の図のようになります。

Connection Information

To access and manage clusters, certificates granted by Alibaba Cloud are required. Each

Revoke Downloaded Certificate

Cluster Access Point:

```
tcp://master4g5.cs-cn-hangzhou.aliyun.com:21003
```

User Guide:

Configure Environment Variable (Linux or Mac):

```
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://master4g5.cs-cn-hangzhou.aliyun.com:21003"
#Set the current path as the storage path for the cluster certificate
export DOCKER_CERT_PATH="$PWD"
```

Notice:

1. The certificate allows secure access to the container cluster. Please keep it secure to access the cluster.
2. If your downloaded certificate is accidentally leaked, you can revoke it and download a new one.

2. TLS 証明書のダウンロードし、保存します。

Docker クラスターへのアクセスに上記のアクセスアドレスを使用する前に、TLS 証明書を設定します。

クラスター詳細ページの [証明書のダウンロード] をクリックし TLS 証明書をダウンロードします。 `certFiles.zip` ファイルがダウンロードされます。以下の例では、ダウンロードした証明書を、`~/.acs/certs/ClusterName/` ディレクトリに保存します。

`ClusterName` はお使いのクラスター名を示します。異なるディレクトリに証明書を保存することができますが、管理を簡単にするため `~/.acs/certs/ClusterName/` ディレクトリを使用する推奨します。

```
mkdir ~/.acs/certs/ClusterName/ # Replace ClusterName
with your cluster name
cd ~/.acs/certs/ClusterName/
cp /path/to/certFiles.zip .
unzip certFiles.zip
```

`certFiles.zip` ファイルには、`ca.pem` ファイル、`cert.pem` ファイルおよび `key.pem` ファイルが含まれます。

クラスターの管理

Docker クライアントを使用したクラスターの管理

Docker クライアントを使用し、Container Service のコンテナクラスターにアクセスできます。これを行うため、以下のように証明書およびアクセスアドレスの設定をする必要があります。

- ・ コマンドラインパラメーター

```
docker --tlsverify --tlscacert=~/.acs/certs/ClusterName/
ca.pem --tlscert=~/.acs/certs/ClusterName/cert.
pem --tlskey=~/.acs/certs/ClusterName/key.pem \
-H=tcp://master2g1.cs-cn-Qingdao.aliyun.com:11599
ps # Replace ClusterName and tcp://master2g1.cs-cn-
Qingdao.aliyun.com:11599 with the actual path and
access address .
```

- ・ 環境変数

```
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://master2g1.cs-cn-Qingdao.
aliyun.com:11599" # Replace tcp://master2g1.cs-cn-
-Qingdao.aliyun.com:11599 with the actual access
address
export DOCKER_CERT_PATH=~/.acs/certs/ClusterName/ #
Replace ClusterName with the actual path
```

```
docker ps
```

上記の2つの例は、クラスターでの `docker ps` コマンドの実行方法を示しています。

`ps` を他の Docker コマンドに置き換えることができます。たとえば、`docker run` コマンドを実行して、新しいコンテナを開始できます。

Docker Compose を使用したクラスター管理

Docker Compose により、環境変数を利用したアクセスアドレスおよび証明書の宣言がサポートされます。

```
export DOCKER_TLS_VERIFY = " 1 "
export DOCKER_HOST = " tcp :// master2g1 . cs - cn - Qingdao .
aliyun . com : 11599 "
export DOCKER_CERT_PATH = ~/. acs / certs / ClusterName
docker - compose up
```

証明書の取り消し

ダウンロードした証明書が、使用中に誤って流出した場合、できるだけ速やかに証明書を取り消します。クラスター詳細ページで [ダウンロードした証明書の取り消し] をクリックし、ダウンロードした証明書を取り消します。その後、新しい証明書をダウンロードします。



注：

[ダウンロードした証明書の取り消し] のクリックにより、ダウンロードした証明書が利用できなくなります。