Migrate MySQL image from Docker Desktop and Deploy in Huawei Cloud Elastic Cloud Service (ECS)

This document described an approach on migrating MYSQL image from Docker Desktop which located in local computer and deploy the image in Huawei Cloud Elastic Cloud Service(ECS).

Elastic Cloud Server (ECS) provides secure, scalable, on-demand compute resources, enabling you to flexibly deploy applications and workloads. Elastic Cloud Server (ECS) provides secure, scalable, on-demand compute resources, enabling you to flexibly deploy applications and workloads.

Docker is an open platform for **developing, shipping, and running applications**. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications.

Table of Content

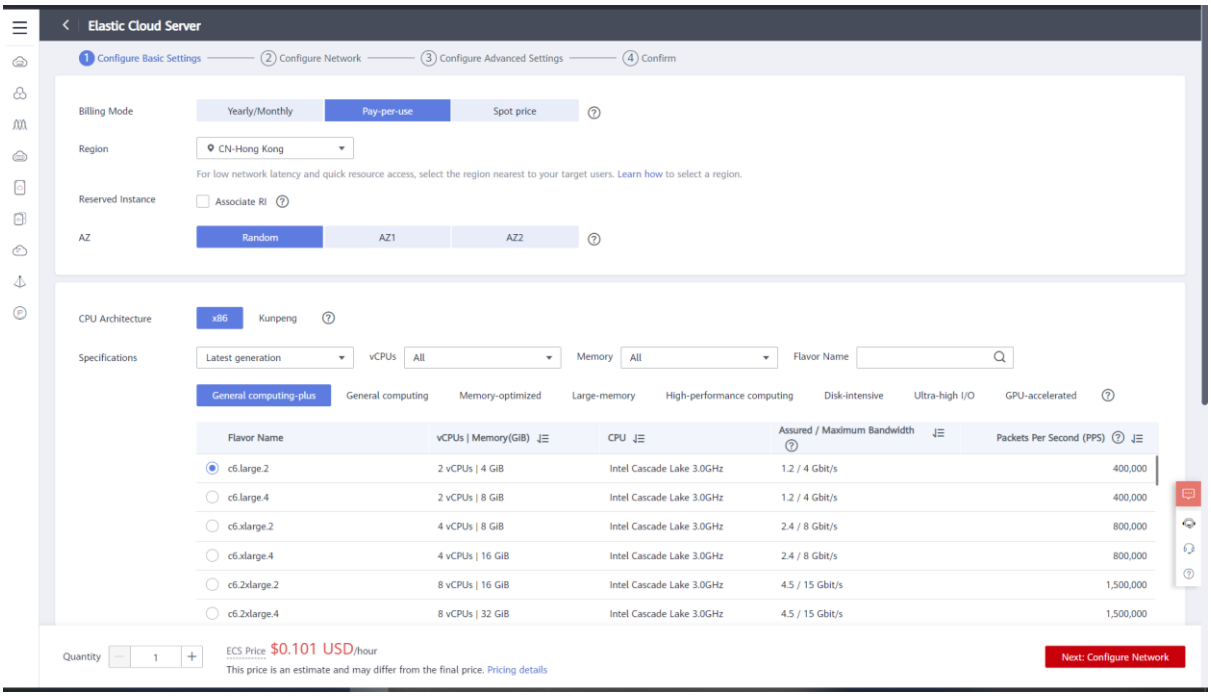
Pre-requites:

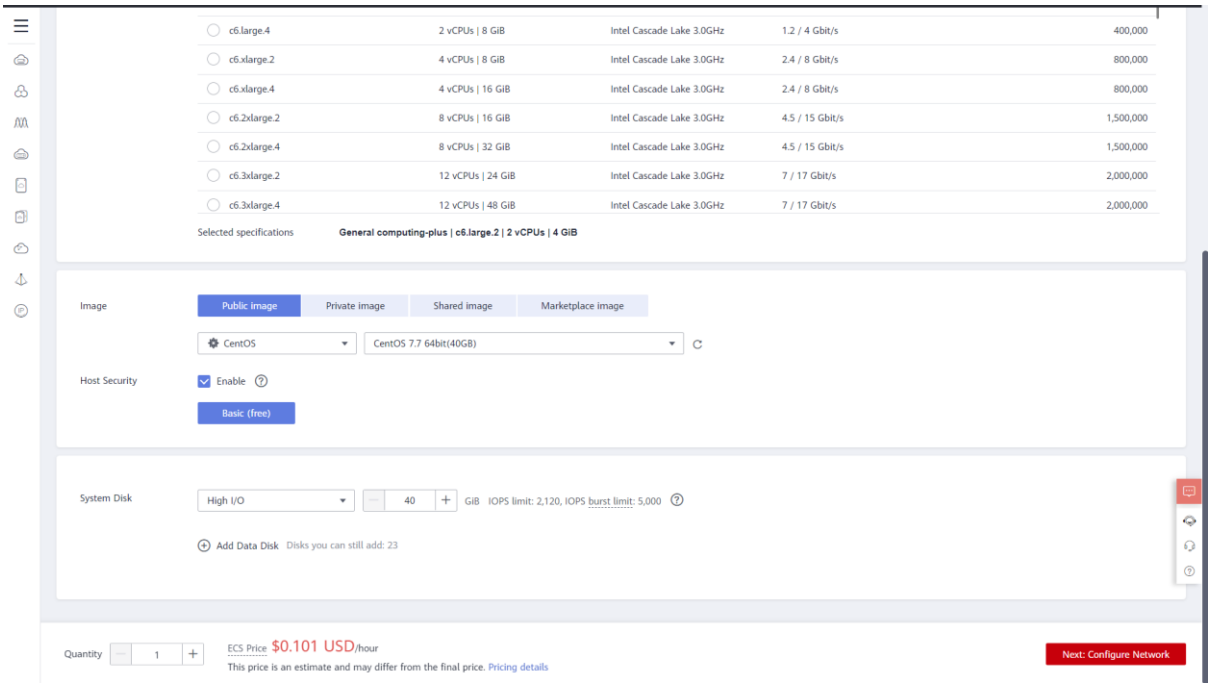
1. Downloaded and install Docker Desktop in local computer.
   Website: https://docs.docker.com/desktop/install/windows-install/

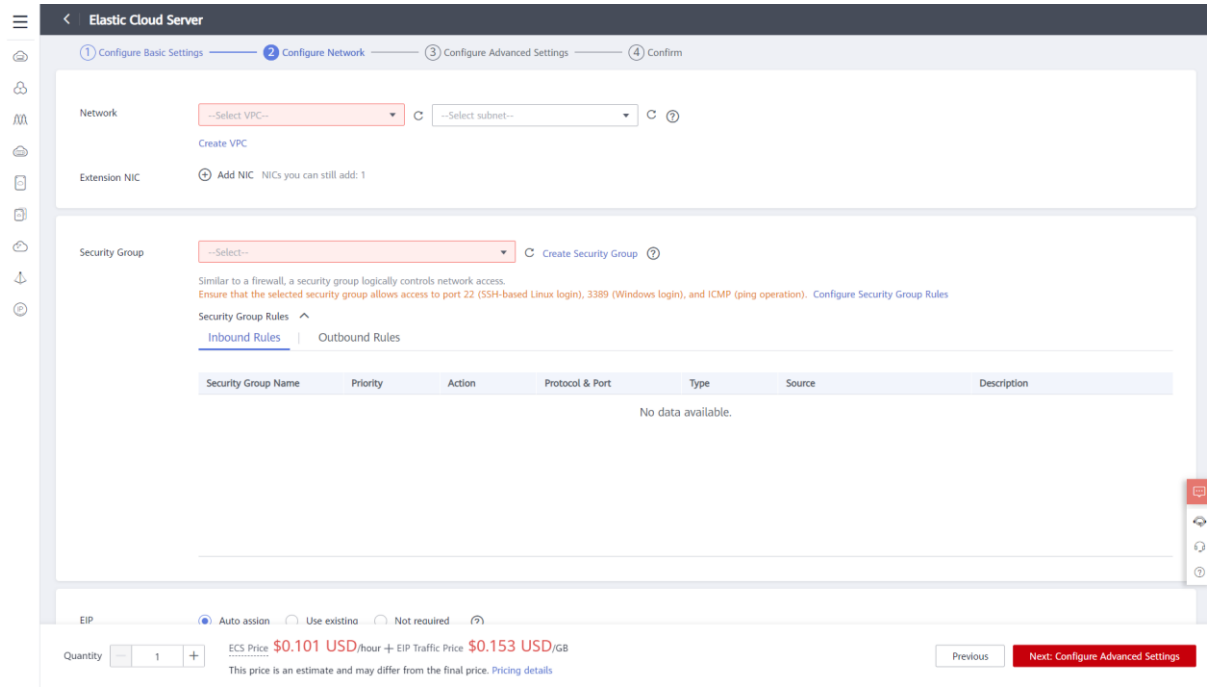
2. Create an account for Docker Hub.
   Website: https://hub.docker.com/signup

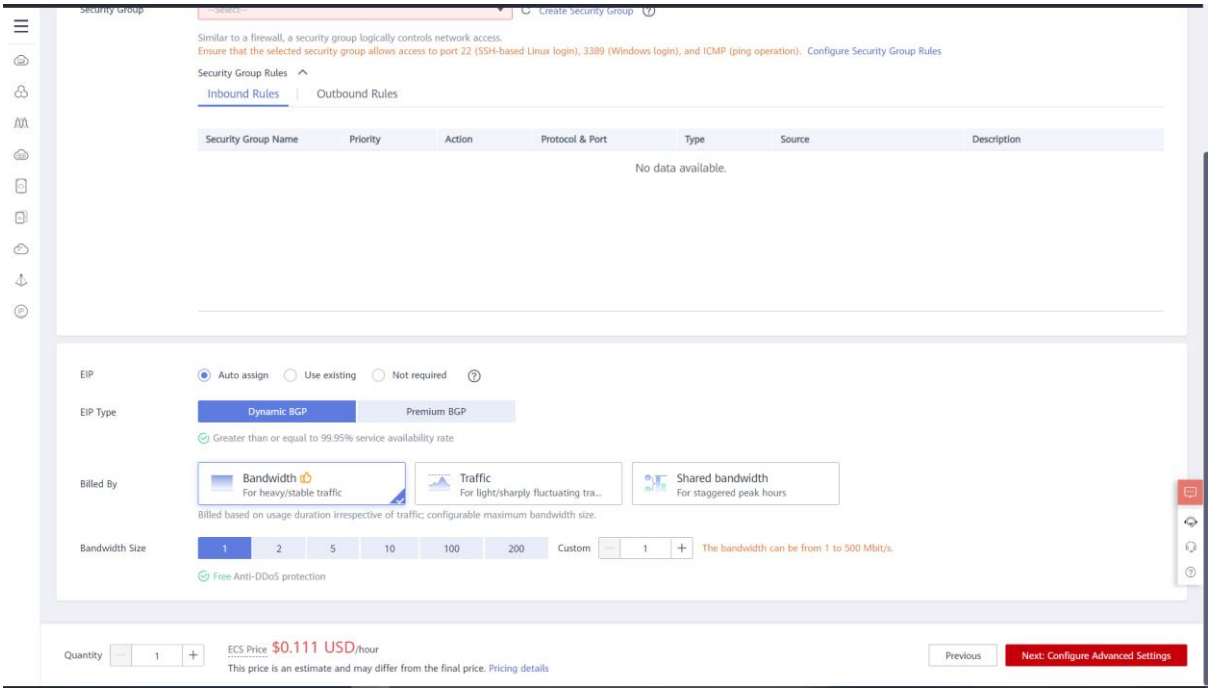
3. Create an Linux based ECS in specific region (Hong Kong) using main Huawei Cloud Account.

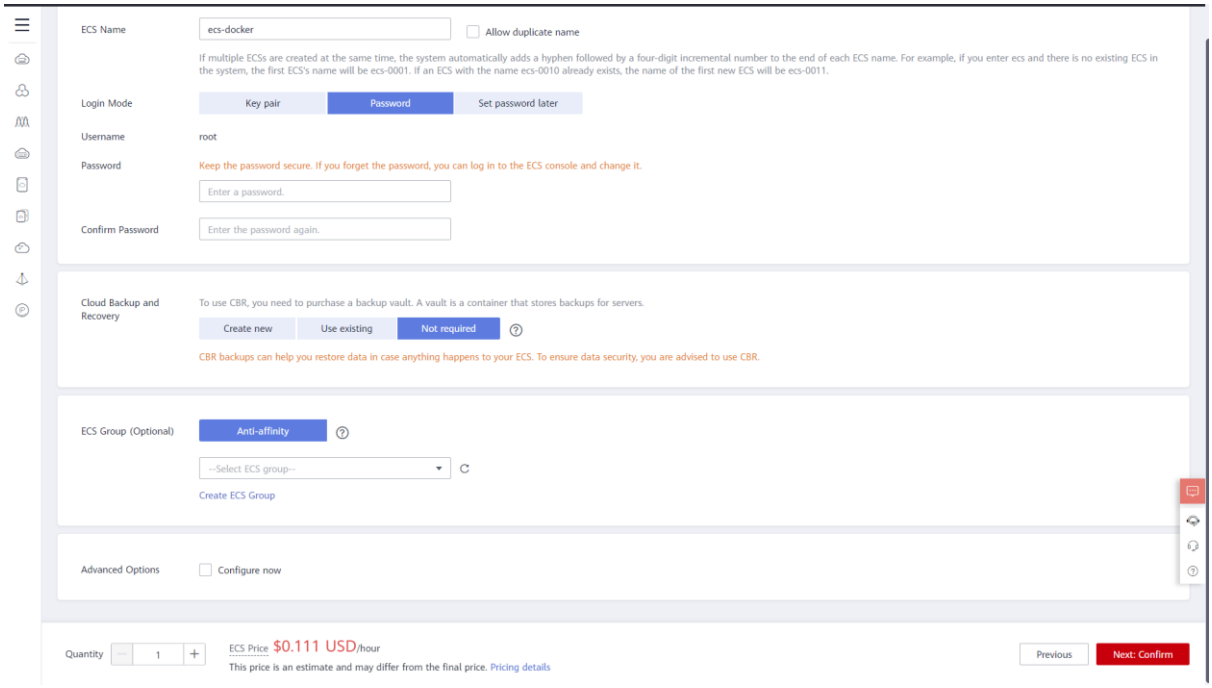
   Website: https://support.huaweicloud.com/intl/en-us/ecs/index.html






Elastic Cloud Server
1 Configure Basic Settings
2 Configure Network
3 Configure Advanced Settings
4 Confirm
Network
--Select VPC--
--Select subnet--
Create VPC
Extension NIC
Add NIC NICs you can still add: 1
Security Group
--Select--
Create Security Group
Similar to a firewall, a security group logically controls network access.
Ensure that the selected security group allows access to port 22 (SSH-based Linux login), 3389 (Windows login), and ICMP (ping operation). Configure Security Group Rules
Security Group Rules
Inbound Rules
Outbound Rules
Security Group Name
Priority
Action
Protocol & Port
Type
Source
Description
No data available.
EIP
Auto assign
Use existing
Not required
Quantity
1
ECS Price $0.101 USD/hour + EIP Traffic Price $0.153 USD/GB
This price is an estimate and may differ from the final price. Pricing details
Previous
Next: Configure Advanced Settings

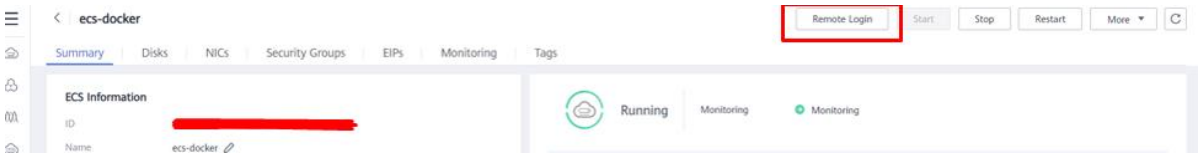

Security Group
Create Security Group
Similar to a firewall, a security group logically controls network access.
Ensure that the selected security group allows access to port 22 (SSH-based Linux login), 3389 (Windows login), and ICMP (ping operation). Configure Security Group Rules
Security Group Rules
Inbound Rules
Outbound Rules
Security Group Name
Priority
Action
Protocol & Port
Type
Source
Description
No data available.
EIP
Auto assign
Use existing
Not required
EIP Type
Dynamic BGP
Premium BGP
Greater than or equal to 99.95% service availability rate
Billed By
Bandwidth
For heavy/stable traffic
Traffic
For light/sharply fluctuating tra...
Shared bandwidth
For staggered peak hours
Billed based on usage duration irrespective of traffic; configurable maximum bandwidth size.
Bandwidth Size
1
2
5
10
100
200
Custom
1
The bandwidth can be from 1 to 500 Mbit/s.
Free Anti-DDoS protection
Quantity
1
ECS Price $0.111 USD/hour
This price is an estimate and may differ from the final price. Pricing details
Previous
Next: Configure Advanced Settings


ECS Name
ecs-docker
Allow duplicate name
If multiple ECSs are created at the same time, the system automatically adds a hyphen followed by a four-digit incremental number to the end of each ECS name. For example, if you enter ecs and there is no existing ECS in the system, the first ECS's name will be ecs-0001. If an ECS with the name ecs-0010 already exists, the name of the first new ECS will be ecs-0011.
Login Mode
Key pair
Password
Set password later
Username
root
Password
Keep the password secure. If you forget the password, you can log in to the ECS console and change it.
Enter a password.
Confirm Password
Enter the password again.
Cloud Backup and Recovery
To use CBR, you need to purchase a backup vault. A vault is a container that stores backups for servers.
Create new
Use existing
Not required
CBR backups can help you restore data in case anything happens to your ECS. To ensure data security, you are advised to use CBR.
ECS Group (Optional)
Anti-affinity
--Select ECS group--
Create ECS Group
Advanced Options
Configure now
Quantity
1
ECS Price $0.111 USD/hour
This price is an estimate and may differ from the final price. Pricing details
Previous
Next: Confirm

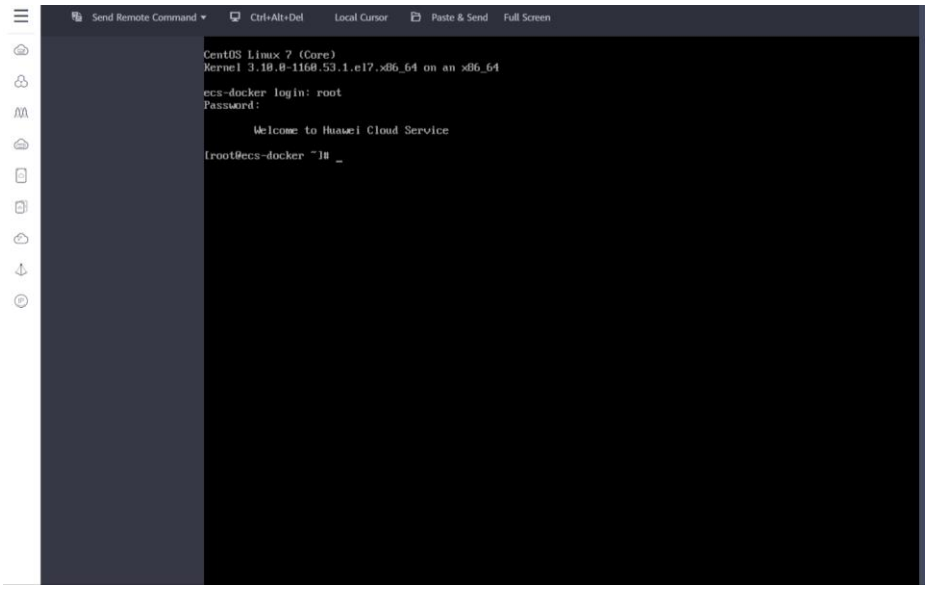
Procedure:

1.Install Docker in ECS using remote login.

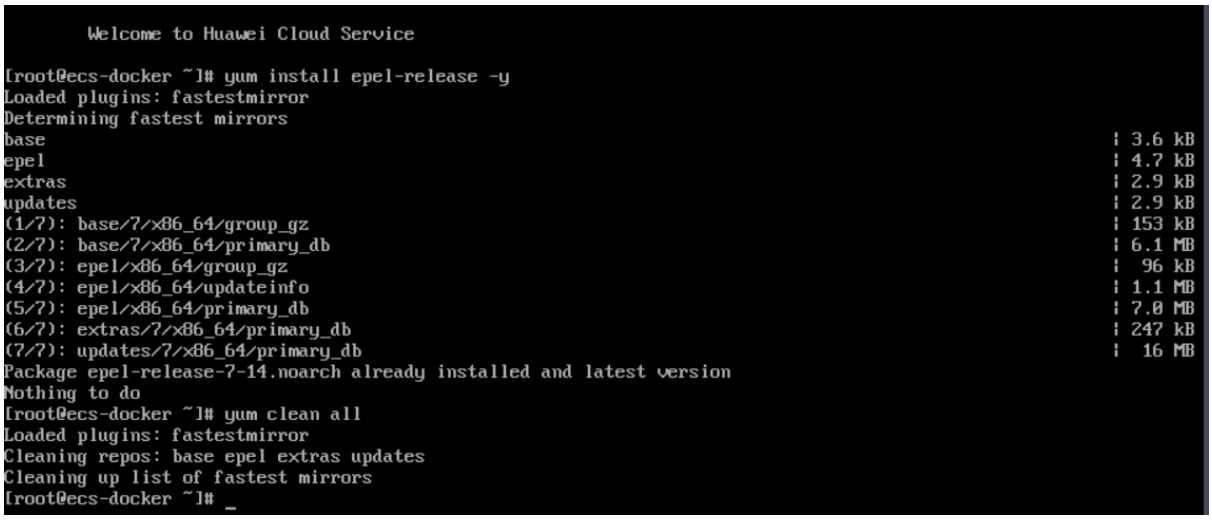
Website: https://support.huaweicloud.com/intl/en-us/bestpractice-ecs/en-us_topic_0141067581.html



Description: Presses on 'Remote Login' button for specific ECS.



Description: Access to ECS console after login with username and password.



Description: Add a YUM repository.

[Command Line]:

a. yum install epel-release -y

b. yum clean all

```
Installed:
  device-mapper-persistent-data.x86_64 0:0.8.5-3.el7_9.2  lvm2.x86_64 7:2.02.187-6.el7_9.5  yum-utils.noarch 0:1.1

Dependency Installed:
  device-mapper-event.x86_64 7:1.02.170-6.el7_9.5            device-mapper-event-libs.x86_64 7:1.02.170-6.el7_9.
  libaio.x86_64 0:0.3.109-13.el7                             lvm2-libs.x86_64 7:2.02.187-6.el7_9.5
  python-chardet.noarch 0:2.2.1-3.el7                        python-kitchen.noarch 0:1.1.1-5.el7

Complete!
```

Description: Install yum-utils.

[Command Line]

a. sudo yum install -y yum-utils device-mapper-persistent-data lvm2

```
[root@ecs-docker ~]# sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
Loaded plugins: fastestmirror
adding repo from: https://download.docker.com/linux/centos/docker-ce.repo
grabbing file https://download.docker.com/linux/centos/docker-ce.repo to /etc/yum.repos.d/docker-ce.repo
repo saved to /etc/yum.repos.d/docker-ce.repo
```

Description: Configure the YUM repository for Docker.

[Command Line]

a. sudo yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo

```
Dependency Installed:
  audit-libs-python.x86_64 0:2.8.5-4.el7                     checkpolicy.x86_64 0:2.5-8.el7
  container-selinux.noarch 2:2.119.2-1.911c772.el7_8         containerd.io.x86_64 0:1.6.6-3.1.el7
  docker-ce-cli.x86_64 1:20.10.17-3.el7                      docker-ce-rootless-extras.x86_64 0:20.10.17-3.el
  docker-scan-plugin.x86_64 0:0.17.0-3.el7                   fuse-overlayfs.x86_64 0:0.7.2-6.el7_8
  fuse3-libs.x86_64 0:3.6.1-4.el7                            libcgroup.x86_64 0:0.41-21.el7
  libsemanage-python.x86_64 0:2.5-14.el7                     policycoreutils-python.x86_64 0:2.5-34.el7
  python-IPy.noarch 0:0.75-6.el7                             setools-libs.x86_64 0:3.3.8-4.el7
  slirp4netns.x86_64 0:0.4.3-4.el7_8

Complete!
[root@ecs-docker ~]#
```

```
[root@ecs-docker ~]# systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker
[root@ecs-docker ~]#
```

Description: Install and run Docker.

[Command Line]

a. sudo yum install docker-ce

b. systemctl enable docker

c. systemctl start docker

```
Croot@ecs-docker ~]# docker --version                        t     ker.service to /usr/lib/systemd/system/docker.
Docker version 20.10.17, build 100c701   e
```

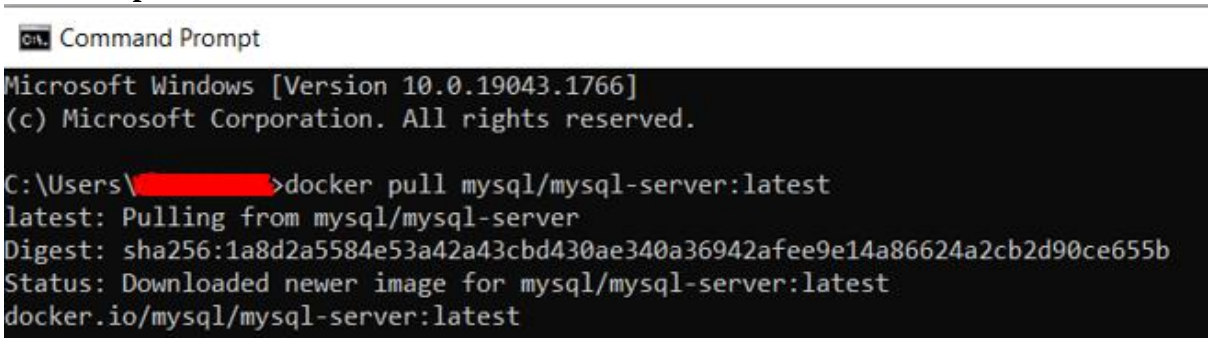
Description: Check for install docker version

[Command Line]

a. docker –version

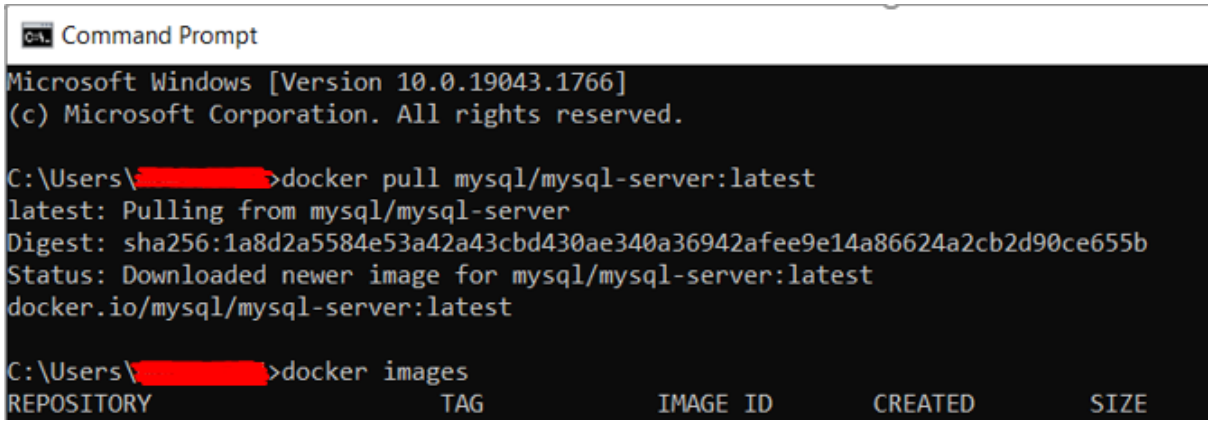
2.Pull public MySQL image from Docker Hub marketplace and run Docker container.

Step 1: Pull the MySQL Docker Image from marketplace using command prompt in local computer.



[Command Line]

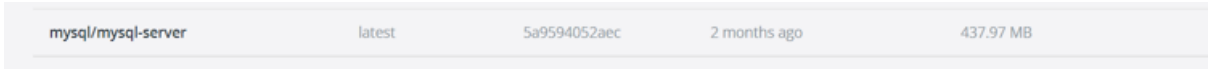
a. docker pull mysql/mysql-server:latest



Description: Check for images downloaded locally

[Command Line]

a. docker images

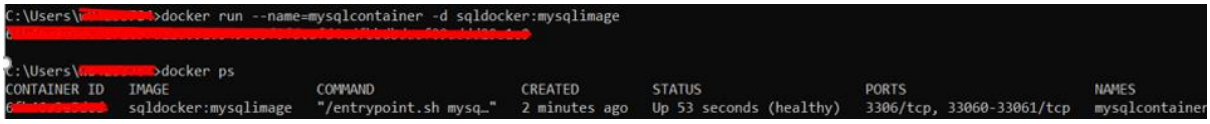


Description: Installed image can be found in docker desktop.

Step 2: Rename the SQL image in Docker Desktop.

Website:https://www.janbasktraining.com/community/devops/how-to-rename-docker-images-without-rebuilding-it

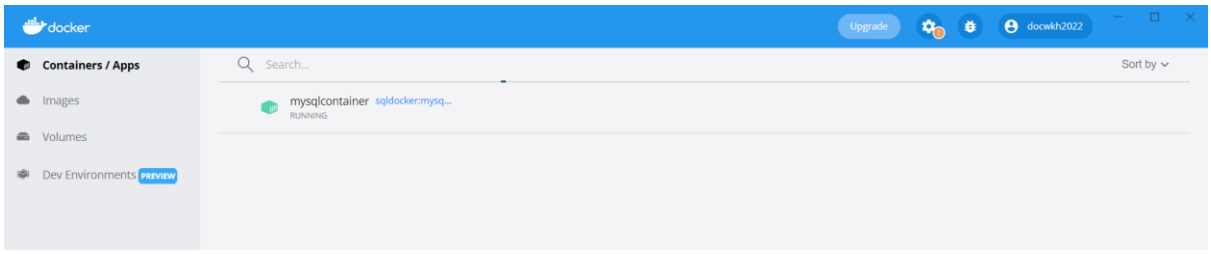
Step 3: Run MySQL image on a container using terminal.



Description: Deploying new container

[Command Line]

a. docker run --name=[container_name] -d [image_tag_name]
b. docker ps
   <!-- check to see if the MySQL container is running --!>



Step 4:Connect to the MySQL Docker Container in Docker Desktop locally.

Note:

i. For Linux

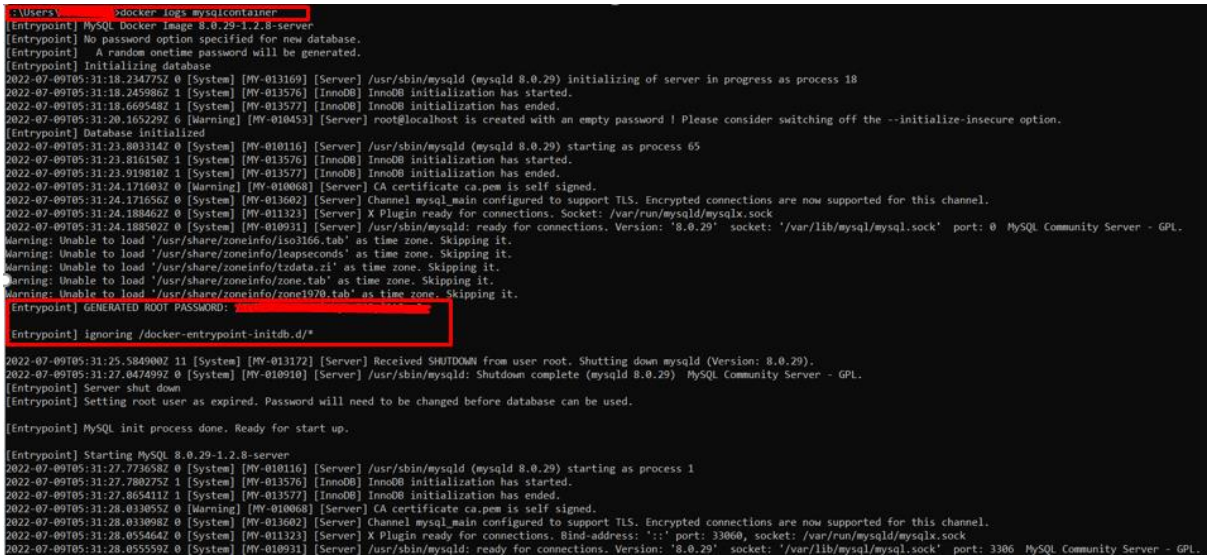
connect the MySQL server container with the host, you need to make sure the MySQL client package is installed using command prompt

[Command Line]

```
apt-get install mysql-client
```

ii. For Windows

connect the MySQL server container with the host, you need to make sure the MySQL client package is installed from http://dev.mysql.com/downloads/shell/.

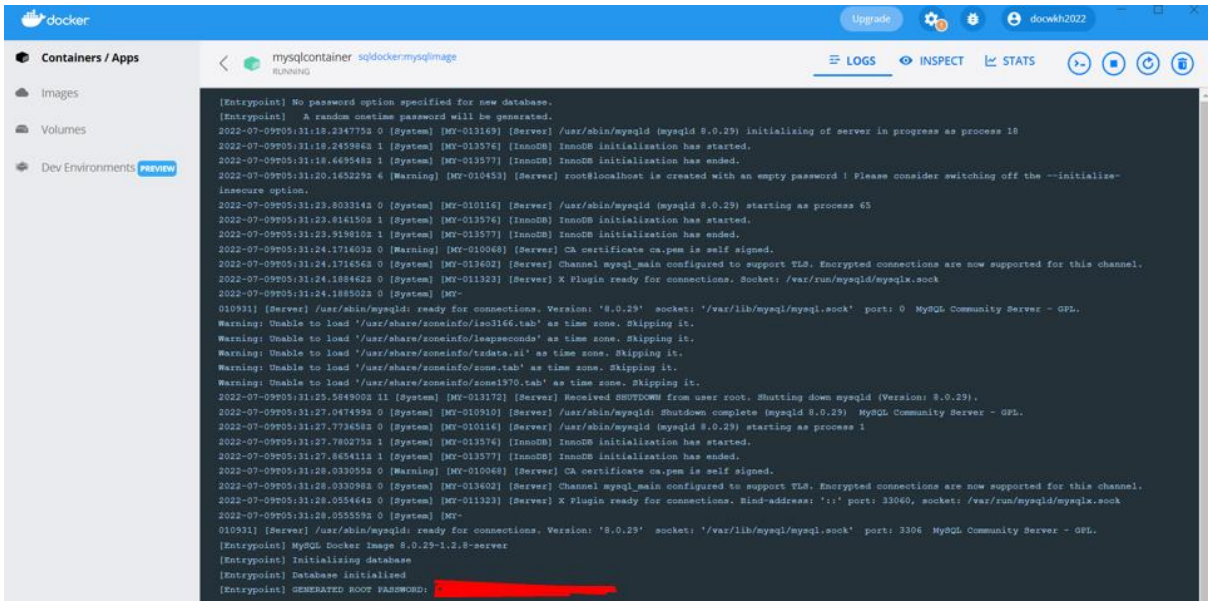


Description: Open the logs file for the MySQL container to find the generated root password.

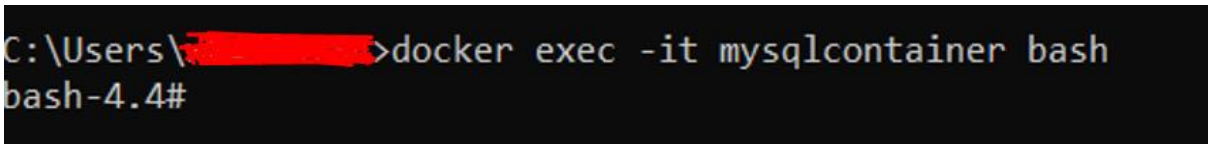
[Command Line]

a. docker logs [container_name]

Or



Description: Check it on Docker Desktop



Description: Go to the bash shell of the MySQL container

[Command Line]

a. docker exec -it [container name] bash

```
bash-4.4# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 194
Server version: 8.0.29

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Description: Login to MYSQL

[Command Line]

a. mysql -uroot -p

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'mysqlthebest';
Query OK, 0 rows affected (0.02 sec)

mysql>
```

Description: Change to new password

[Command Line]

a. ALTER USER 'root'@'localhost' IDENTIFIED BY '[newpassword]';

```
mysql> CREATE DATABASE testDB;
Query OK, 1 row affected (0.01 sec)
```

Description: Create a database with name ‘testDB’.

[Command Line]

a. CREATE DATABASE testDB;

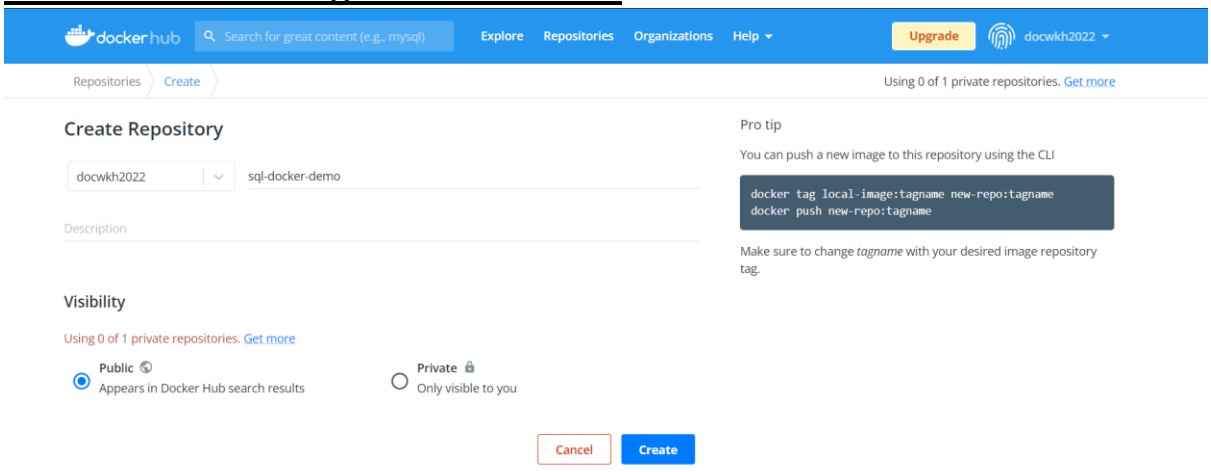
```
mysql> use testDB
Database changed
mysql> CREATE TABLE Employees(
    -> EmpID int,
    -> EmpFirstName varchar(255),
    -> EmpLastName varchar(255),
    -> TelNo varchar(255)
    -> );
Query OK, 0 rows affected (0.04 sec)
```

Description: Create table ‘Employees’ in database.

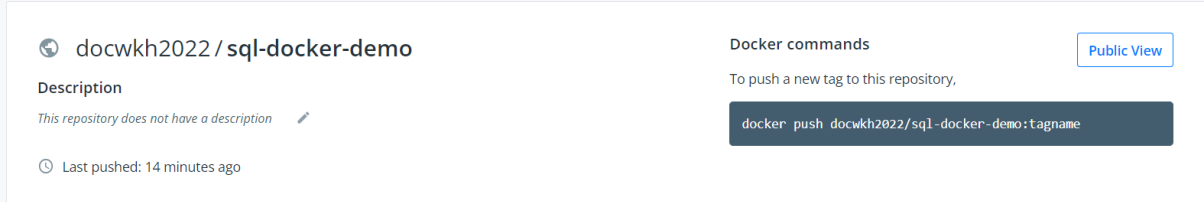
```
mysql> use testDB
Database changed
mysql> show tables;
+------------------+
| Tables_in_testDB |
+------------------+
| Employees        |
+------------------+
1 row in set (0.00 sec)
```

Description: Verify and show tables in database

3.Push modified image to Docker Hub.



Description: Create a respiratory in Docker Hub Account

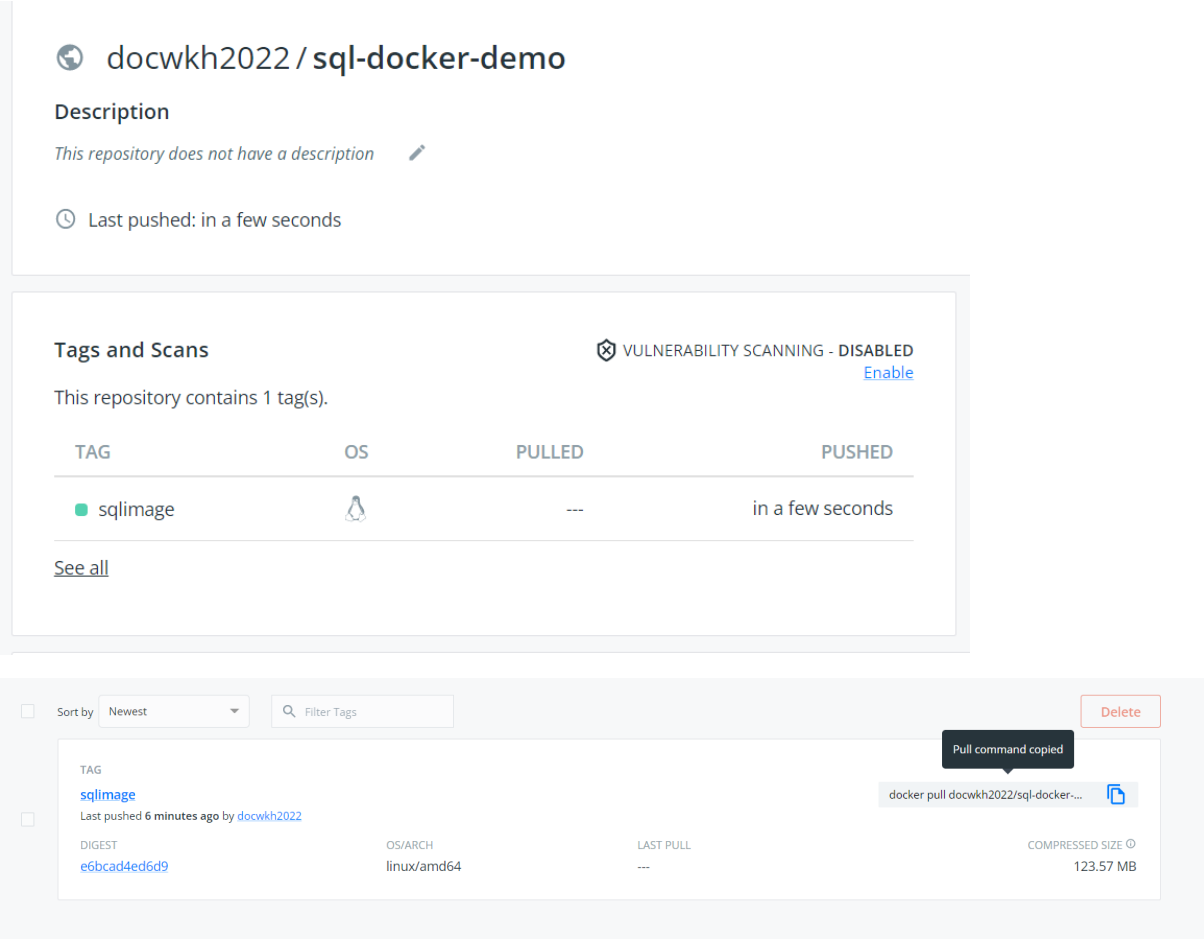



```
C:\Users\[redacted]>docker tag sqldocker:mysqlimage docwkh2022/sql-docker-demo:sqlimage
```

Description: Docker tagging

```
C:\Users\[redacted]>docker push docwkh2022/sql-docker-demo:sqlimage
The push refers to repository [docker.io/docwkh2022/sql-docker-demo]
a72e061dc097: Pushed
62f291a6eca0: Pushed
ce81e7bd2ccb: Pushed
945553f84c3c: Pushed
09ab6c566ea4: Pushed
23af469955bf: Pushed
d2db75568ee6: Pushed
sqlimage: digest: sha256:e6bcad4ed6d971791e639512b740fa2a4c0298a62fa5b52abe4e83436266de9e size: 1781
```

Description: Docker Push to Docker hub



Description: Image launched in Docker Hub account.

4.Pull MySQL image from Docker Hub to Huawei Cloud ECS instance.

```
[root@ecs-docker ~]# docker pull docwkh2022/sql-docker-demo:sqlimage
```

```
[root@ecs-docker ~]# docker pull docwkh2022/sql-docker-demo:sqlimage
sqlimage: Pulling from docwkh2022/sql-docker-demo
e4430e06691f: Pull complete
354bdf52d78f: Pull complete
4ff7ce558aad: Pull complete
513ac8e07801: Pull complete
49e9433767ee: Pull complete
71938d9f5ee9: Pull complete
1150d7e8dd59: Pull complete
Digest: sha256:e6bcad4ed6d971791e639512b740fa2a4c0298a62fa5b52abe4e83436266de9e
Status: Downloaded newer image for docwkh2022/sql-docker-demo:sqlimage
docker.io/docwkh2022/sql-docker-demo:sqlimage
```

```
[root@ecs-docker ~]# docker images
REPOSITORY                  TAG         IMAGE ID       CREATED         SIZE
docwkh2022/sql-docker-demo  sqlimage    5a9594052aec   2 months ago    438MB
```

Description: Pull MySQL image from Docker Hub to Huawei Cloud ECS instance using ECS instance remote login console.

[Command Line]

a. docker pull docwkh2022/sql-docker-demo:sqlimage

   <!—Following the pull command in the respiratory of Docker Hub -!>

```
[root@ecs-docker ~]# docker run -it -d -p 80:80 --name sqlcontainer -v /data/:/var/www/httpd/ 5a9594052aec
9807447b87493bfb45971d3500cab3bd696d14d1505e98cea5c6d7c7ec1e18e5
[root@ecs-docker ~]#
```

Description: Run the MySQL image in Container with Huawei Cloud ECS

Description: Check for generated temporary mysql password

[Command Line]

a. docker logs [Container ID]

```
[root@ecs-docker ~]# docker exec -it 9807447b8749 bash
bash-4.4#
```

Description: Go to the bash shell of the MySQL container

[Command Line]

a. docker exec -it [Container ID] bash

Description: Login to mysql.

[Command Line]

a. mysql -uroot –p

References and Credits to (Website):

1. https://github.com/jakewright/tutorials/tree/master/docker/02-docker-compose
2. https://docs.docker.com/get-started/02_our_app/
3. https://docs.docker.com/desktop/windows/
4. https://www.janbasktraining.com/community/devops/how-to-rename-docker-images-without-rebuilding-it
5. https://phoenixnap.com/kb/mysql-docker-container