

## **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

<b>Pre-requisites:</b> .....	3
<b>Procedure:</b> .....	6
<b>1.Install Docker in ECS using remote login.....</b>	6
<b>2.Pull public MySQL image from Docker Hub marketplace and run Docker container. ....</b>	8
<b>Step 1: Pull the MySQL Docker Image from marketplace using command prompt in local computer. ....</b>	8
<b>Step 2: Rename the SQL image in Docker Desktop.....</b>	8
<b>Step 3: Run MySQL image on a container using terminal. ....</b>	9
<b>Step 4: Connect to the MySQL Docker Container in Docker Desktop locally.....</b>	9
<b>3.Push modified image to Docker Hub.....</b>	12
<b>4.Pull MySQL image from Docker Hub to Huawei Cloud ECS instance.....</b>	14
<b>References and Credits to (Website): .....</b>	15

## Pre-requisites:

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

Billing Mode: Yearly/Monthly **Pay-per-use** Spot price

Region: CN-Hong Kong

Reserved Instance: ☐ Associate RI

AZ: Random AZ1 AZ2

CPU Architecture: x86 Kunpeng

Specifications: Latest generation vCPUs All Memory All Flavor Name

**General computing-plus** General computing Memory-optimized Large-memory High-performance computing Disk-intensive Ultra-high I/O GPU-accelerated

Flavor Name	vCPUs   Memory (GiB)	CPU	Assured / Maximum Bandwidth	Packets Per Second (PPS)
<input checked="" type="radio"/> c6.large.2	2 vCPUs   4 GiB	Intel Cascade Lake 3.0GHz	1.2 / 4 Gbit/s	400,000
<input type="radio"/> c6.large.4	2 vCPUs   8 GiB	Intel Cascade Lake 3.0GHz	1.2 / 4 Gbit/s	400,000
<input type="radio"/> c6.xlarge.2	4 vCPUs   8 GiB	Intel Cascade Lake 3.0GHz	2.4 / 8 Gbit/s	800,000
<input type="radio"/> c6.xlarge.4	4 vCPUs   16 GiB	Intel Cascade Lake 3.0GHz	2.4 / 8 Gbit/s	800,000
<input type="radio"/> c6.2xlarge.2	8 vCPUs   16 GiB	Intel Cascade Lake 3.0GHz	4.5 / 15 Gbit/s	1,500,000
<input type="radio"/> c6.2xlarge.4	8 vCPUs   32 GiB	Intel Cascade Lake 3.0GHz	4.5 / 15 Gbit/s	1,500,000

Quantity: 1 ECS Price: **\$0.101 USD/hour**  
This price is an estimate and may differ from the final price. [Pricing details](#)

**Next: Configure Network**

---

**Selected specifications: General computing-plus | c6.large.2 | 2 vCPUs | 4 GiB**

Image: Public image Private image Shared image Marketplace image

CentOS CentOS 7.7 64bit(40GB)

Host Security: ☒ Enable Basic (free)

System Disk: High I/O 40 GiB IOPS limit: 2,120, IOPS burst limit: 5,000

Add Data Disk Disks you can still add: 23

Quantity: 1 ECS Price: **\$0.101 USD/hour**  
This price is an estimate and may differ from the final price. [Pricing details](#)

**Next: Configure Network**

Configure Basic Settings

Configure Network

Configure Advanced Settings

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--

C

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](#)

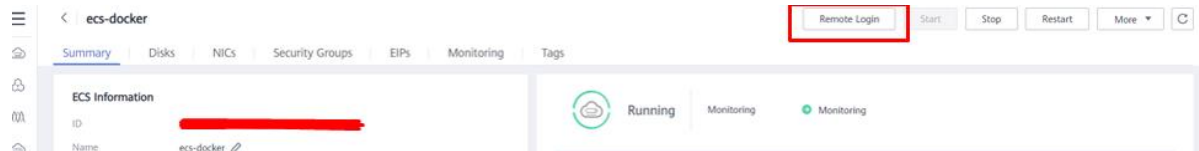
Previous

Next: Confirm

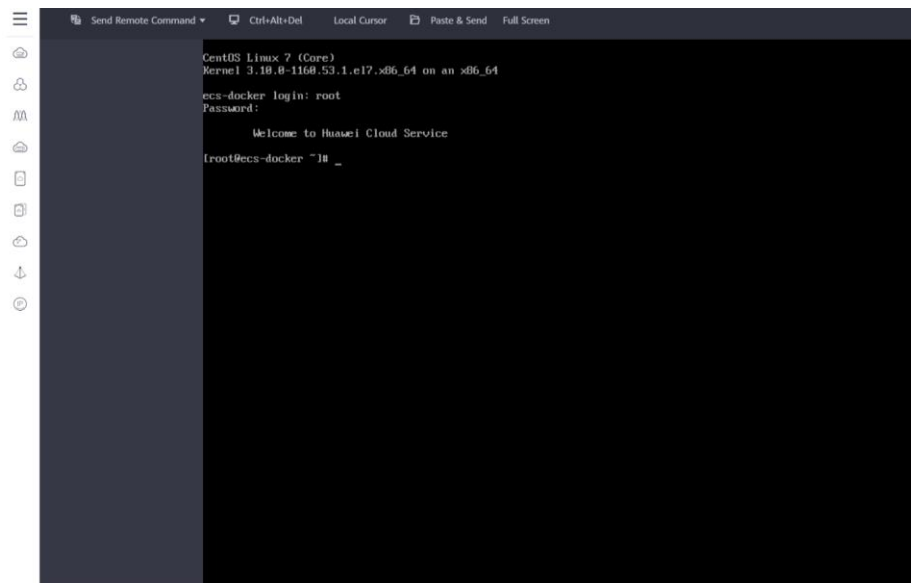
## Procedure:

### 1.Install Docker in ECS using remote login.

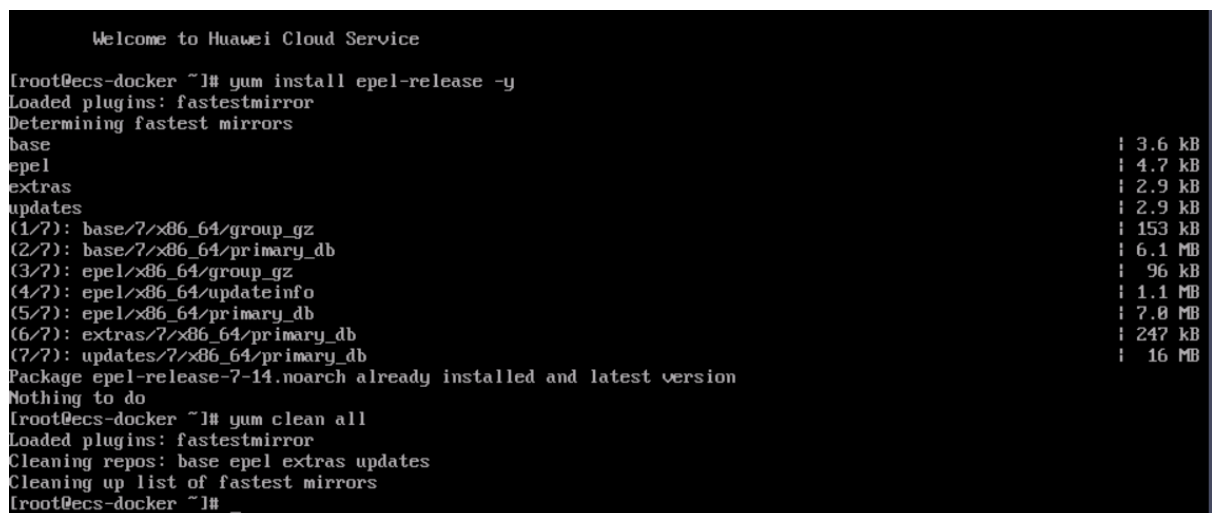
Website: [https://support.huaweicloud.com/intl/en-us/bestpractice-ecs/en-us\\_topic\\_0141067581.html](https://support.huaweicloud.com/intl/en-us/bestpractice-ecs/en-us_topic_0141067581.html)



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]:

- yum install epel-release -y
- 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.el7
  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  libcgroupp.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`

```

[root@ecs-docker ~]# docker --version
Docker version 20.10.17, build 100c701

```

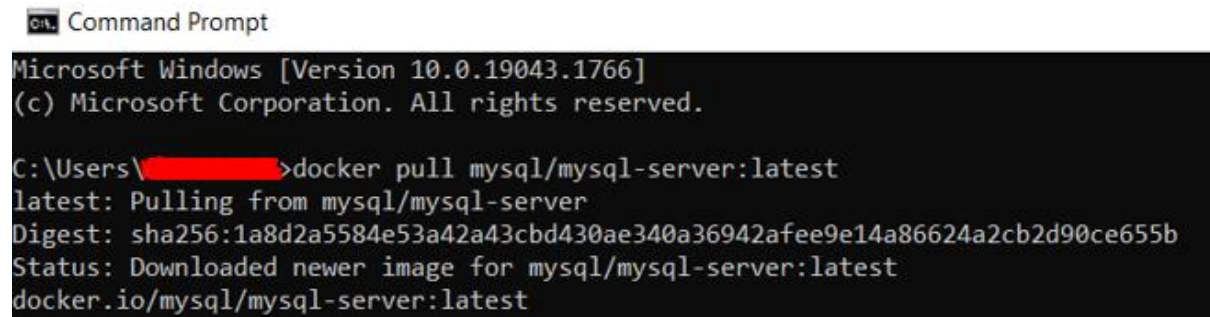
Description: Check for install docker version

[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.**



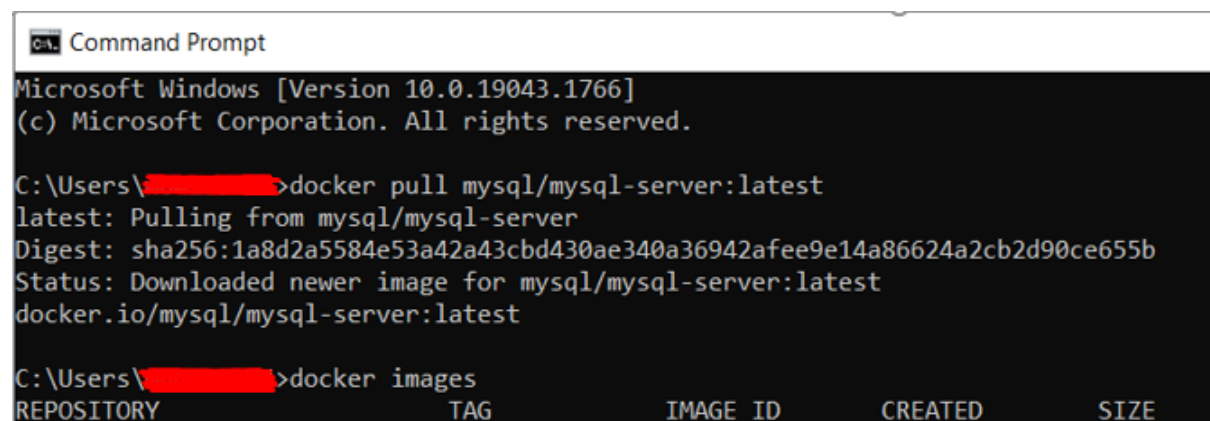
```

Command Prompt
Microsoft Windows [Version 10.0.19043.1766]
(c) Microsoft Corporation. All rights reserved.

C:\Users\>docker pull mysql/mysql-server:latest
latest: Pulling from mysql/mysql-server
Digest: sha256:1a8d2a5584e53a42a43cbd430ae340a36942afee9e14a86624a2cb2d90ce655b
Status: Downloaded newer image for mysql/mysql-server:latest
docker.io/mysql/mysql-server:latest
  
```

[Command Line]

- a. `docker pull mysql/mysql-server:latest`



```

Command Prompt
Microsoft Windows [Version 10.0.19043.1766]
(c) Microsoft Corporation. All rights reserved.

C:\Users\>docker pull mysql/mysql-server:latest
latest: Pulling from mysql/mysql-server
Digest: sha256:1a8d2a5584e53a42a43cbd430ae340a36942afee9e14a86624a2cb2d90ce655b
Status: Downloaded newer image for mysql/mysql-server:latest
docker.io/mysql/mysql-server:latest

C:\Users\>docker images
REPOSITORY              TAG               IMAGE ID          CREATED          SIZE
mysql/mysql-server      latest           5a9594052aec     2 months ago    437.97 MB
  
```

Description: Check for images downloaded locally

[Command Line]

- a. `docker images`

mysql/mysql-server	latest	5a9594052aec	2 months ago	437.97 MB
--------------------	--------	--------------	--------------	-----------

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.**

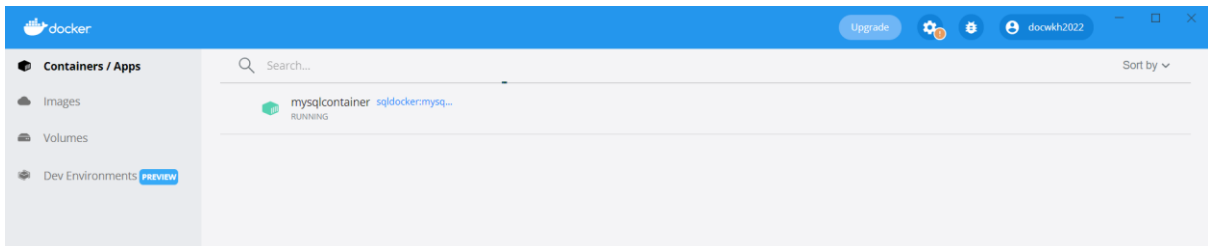
```
C:\Users\>docker run --name=mysqlcontainer -d sqldocker:mysqlimage
C:\Users\>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
62a2a2a2a2a2	sqldocker:mysqlimage	"/entrypoint.sh mysql_"	2 minutes ago	Up 53 seconds (healthy)	3306/tcp, 33060-33061/tcp	mysqlcontainer

Description: Deploying new container

[Command Line]

- docker run --name=[container\_name] -d [image\_tag\_name]
  - 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/>.*

```

C:\Users\>docker logs mysqlcontainer
[Entrypoint] MySQL Docker Image 8.0.29-1.2.8-server
[Entrypoint] No password option specified for new database.
[Entrypoint] A random onetime password will be generated.
[Entrypoint] Initializing database
2022-07-09T05:31:18.234775Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.29) initializing of server in progress as process 18
2022-07-09T05:31:18.245986Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2022-07-09T05:31:18.669548Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2022-07-09T05:31:20.165229Z 0 [Warning] [MY-010453] [Server] root@localhost is created with an empty password ! Please consider switching off the --initialize-insecure option.
[Entrypoint] Database initialized
2022-07-09T05:31:23.803314Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.29) starting as process 65
2022-07-09T05:31:23.816150Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2022-07-09T05:31:23.919810Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2022-07-09T05:31:24.171603Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2022-07-09T05:31:24.171656Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.
2022-07-09T05:31:24.188462Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Socket: /var/run/mysqld/mysqld.sock
2022-07-09T05:31:24.188502Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.29' socket: '/var/lib/mysql/mysql.sock' port: 0 MySQL Community Server - GPL.
Warning: Unable to load '/usr/share/zoneinfo/iso3166.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/leapseconds.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/tzdata.zi' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/zone.tab' as time zone. Skipping it.
Warning: Unable to load '/usr/share/zoneinfo/zone1970.tab' as time zone. Skipping it.
[Entrypoint] GENERATED ROOT PASSWORD:
[Entrypoint] ignoring /docker-entrypoint-initdb.d/*

2022-07-09T05:31:25.584900Z 11 [System] [MY-013172] [Server] Received SHUTDOWN from user root. Shutting down mysqld (Version: 8.0.29).
2022-07-09T05:31:27.047499Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown complete (mysqld 8.0.29) MySQL Community Server - GPL.
[Entrypoint] Server shut down
[Entrypoint] Setting root user as expired. Password will need to be changed before database can be used.

[Entrypoint] MySQL init process done. Ready for start up.

[Entrypoint] Starting MySQL 8.0.29-1.2.8-server
2022-07-09T05:31:27.773680Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.29) starting as process 1
2022-07-09T05:31:27.780275Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2022-07-09T05:31:27.865411Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2022-07-09T05:31:28.033055Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2022-07-09T05:31:28.033098Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this channel.
2022-07-09T05:31:28.055464Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqld.sock
2022-07-09T05:31:28.055559Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.29' socket: '/var/lib/mysql/mysql.sock' port: 3306 MySQL Community Server - GPL.

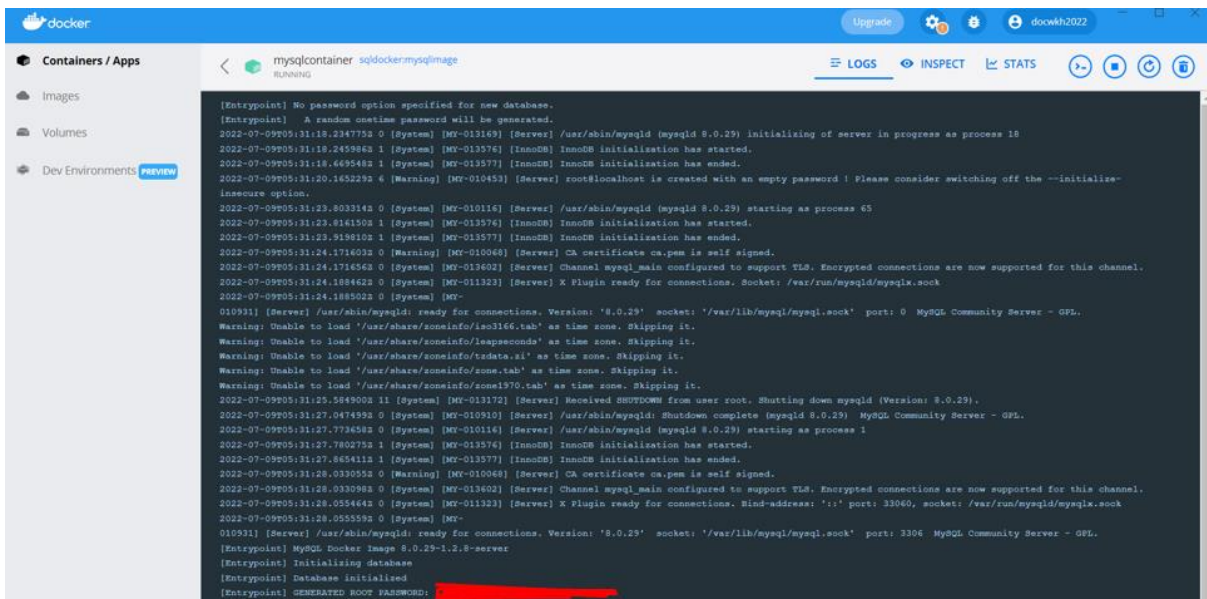
```

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

```

C:\Users\>docker exec -it mysqlcontainer bash
bash-4.4#

```

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.

The screenshot shows the Docker Hub 'Create Repository' interface. The repository name is 'docwkh2022/sql-docker-demo'. The visibility is set to 'Public' (appears in Docker Hub search results). A 'Pro tip' box on the right provides CLI commands: `docker tag local-image:tagname new-repo:tagname` and `docker push new-repo:tagname`. The page also indicates 'Using 0 of 1 private repositories'.

Description: Create a respiratory in Docker Hub Account


**docwkh2022 / sql-docker-demo**

**Description**

*This repository does not have a description*

⌚ Last pushed: 14 minutes ago

**Docker commands**

To push a new tag to this repository,

```
docker push docwkh2022/sql-docker-demo:tagname
```


[Public View](#)

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

Description: Docker tagging

```
C:\Users\>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


**docwkh2022 / sql-docker-demo**

**Description**



*This repository does not have a description*

⌚ Last pushed: in a few seconds

**Tags and Scans**

⚠️ VULNERABILITY SCANNING - DISABLED [Enable](#)

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
 <b>sqlimage</b>		---	in a few seconds

[See all](#)

☐ Sort by **Newest**

☐

**TAG**

[sqlimage](#)

Last pushed 6 minutes ago by [docwkh2022](#)

**DIGEST**

[e6bcad4ed6d9](#)

**OS/ARCH**

linux/amd64

**LAST PULL**

---

**COMPRESSED SIZE**

123.57 MB

**Pull command copied**

```
docker pull docwkh2022/sql-docker-...
```

[Delete](#)

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
954bdf52d78f: 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
9807447b87493bf645971d3500cab3bd696d14d1505e98cea5c6d7c7ec1e18e5
[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`

```
[root@ecs-docker ~]# docker exec -it 9807447b8749 bash
bash-4.4# mysql -uroot -p
Enter password:
```

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/](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>