Huawei Cloud EulerOS Tutorials

Issue 01

Date 2024-12-06

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Deploying MySQL

Introduction

MySQL is an open-source relational database management system. Like other relational databases, MySQL stores data in tables consisting of rows and columns. Users can define, manipulate, control, and query data using a structured query language (usually called SQL). This tutorial describes how you can deploy MySQL in Huawei Cloud EulerOS (HCE) 2.0.

Preparations

- Prepare an ECS and assign a public IP address or elastic IP address (EIP) to the ECS.
- Ensure that inbound security group rules allow traffic to flow to the ECS over ports 22 and 3306.

Procedure

Step 1 Install the MySQL server and client.

1. Run the following command to install the MySQL server and client:
   ```
   dnf install mysql-server mysql-common mysql -y
   ```
2. Run the following command to check the MySQL version:
   ```
   mysql -V
   ```
   If information similar to the following is displayed, the MySQL server and client are successfully installed:
   ```
   mysql  Ver 8.0.37 for Linux on x86_64 (Source distribution)
   ```

Step 2 Configure MySQL.

1. Run the following command to start MySQL:
   ```
   systemctl start mysqld
   ```
2. Run the following command to check the MySQL status:
   ```
   systemctl status mysqld
   ```
   If **active (running)** is displayed, MySQL is started.

   NOTE

   To set mysqld to automatically enable upon system boot, run the following command:
   systemctl enable mysqld

3. Run the following command to perform security configuration for MySQL:

```
mysql_secure_installation
```

Customize options based on personal requirements and prompt information.

```
Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: Y // Choose whether to set the password component.

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary                  file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 2 // Select the password security policy.
Please set the password for root here.

New password:

Re-enter new password:

Estimated strength of the password: 100
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y
By default, a MySQL installation has an anonymous user,
allowing anyone to log in to MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y // Choose whether to remove
anonymous users.
Success.


Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y // Choose whether to
disallow remote login of the root user.
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.


Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y // Choose whether
to remove the test database.
 - Dropping test database...
Success.

 - Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y // Choose whether to reload
privilege tables.
```



```
Success.

All done!
```

Step 3 Remote connect to the database.

1. Run the following command on the MySQL server to connect to the database:
   ```
   mysql -uroot -p
   ```
   Enter the password of the **root** user.
2. Enter the following statement to create a test account and password:
   ```
   create user 'test'@'%' identified by 'test****';
   ```
   test indicates the account name, and **test****** indicates the password.
3. Execute the following statement to grant all database permissions to the test account:
   ```
   grant all privileges on *.* to 'test'@'%';
   ```
4. Execute the following statement to update the permissions:
   ```
   flush privileges;
   ```
5. Execute the following command to exit the database:
   ```
   exit
   ```
6. Run the following command on the MySQL client to remotely connect to the database:
   ```
   mysql -h <IP address of the MySQL client> -utest -p
   ```
   Alternatively, use the Navicat or Visual Studio Code plug-in for remote connections.

----End

CAUTION

The password and permission configurations are used only for tests. Exercise caution when using them in the service environment.

2 Installing Docker

Introduction

Docker is an open-source application container engine that features portability, scalability, high security, and manageability. Developers can package applications and dependencies into portable containers, quickly release them to Linux servers, and use the OS-level virtualization to build, deploy, and manage applications more efficiently. This tutorial describes how you install Docker in HCE 2.0.

Preparations

Prepare an ECS.

Procedure

Step 1 Install Docker.

1. Run the following command to install Docker:
   ```
   dnf install docker
   ```
2. Run the following command to start Docker:
   ```
   systemctl start docker
   ```

Step 2 Verify that Docker is installed.

1. Run the following command to run the hello-world image:
   ```
   docker run hello-world
   ```
2. If information similar to the following is displayed, Docker is installed:


```
[root@secure ~]# docker run helloworld
Unable to find image 'helloworld:latest' locally
docker: Error response from daemon: pull access denied for helloworld, repositor
See 'docker run --help'.
[root@secure ~]# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:88ec0acaa3ec199d3b7eaf73588f4518c25f9d34f58ce9a0df68429c5af48e8d
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Step 3 Use Docker.

1. Manage the Docker daemon.
   - Run the daemon.

     ```
     systemctl start docker
     ```

   - Stop the daemon.

     ```
     systemctl stop docker
     ```

   - Restart the daemon.

     ```
     systemctl restart docker
     ```

2. Manage the images.
   - Edit the label.

     ```
     docker tag hello-world:latest hello-world:v1
     ```

   - View existing images.

     ```
     docker images
     ```

   - Forcibly delete an image.

     ```
     docker rmi -f hello-world:v1
     ```

3. Manage containers.
   - Enter a container.

     ```
     docker run -it ImageId /bin/bash
     ```

   - Exit a container.

     ```
     exit
     ```

   - Access a container running in the backend.

     ```
     docker exec -it <Container ID> /bin/bash
     ```

   - Create an image from a container.

     ```
     docker commit <Container ID> [<Repository name>[:<Label>]]
     ```

     Example command:


```
docker commit 135****9f757 hello-world:v1
```

CAUTION

Docker 18.09.0 is installed. To install a later version, see **Docker Documentation**.

----End

3 Setting Up an FTP Site

Introduction

vsftpd (very secure FTP daemon), is an FTP server for Unix-like systems, including Linux. This tutorial describes how you can deploy vsftpd in HCE 2.0.

Preparations

- Prepare an ECS and assign a public IP address or EIP to the ECS.
- Ensure that inbound security group rules allow traffic to flow to the ECS over port 21.

Procedure

Step 1 Install vsftpd and start it.

1. Run the following command to install vsftpd:
   ```
   dnf install vsftpd
   ```
2. Run the following command to start vsftpd:
   ```
   systemctl start vsftpd
   ```
3. Run the following command to check the vsftpd status:
   ```
   systemctl status vsftpd
   ```
   If **active (running)** is displayed, vsftpd is started.
   ```
   [root@hce2 ~]# systemctl status vsftpd
   ● vsftpd.service - Vsftpd ftp daemon
        Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled; vendor preset: disabled)
        Active: active (running) since Tue 2023-10-10 15:55:11 CST; 1s ago
       Process: 21033 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited, status=0/SUCCESS)
      Main PID: 21034 (vsftpd)
         Tasks: 1 (limit: 198730)
        Memory: 408.0K
        CGroup: /system.slice/vsftpd.service
                └─ 21034 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
   ```

NOTE

To set vsftpd to automatically enable upon system boot, run the following command:

```
systemctl enable vsftpd
```

Step 2 Add a user for the FTP service.

1. Run the following command to create a user for the FTP service:
   ```
   adduser ftp
   ```


2. Run the following command to set a password of the created user:
```
passwd ftp
```

Step 3 Configure vsftpd.

1. Run the following commands to create directories and files for the FTP service (you can change the directories as required):
```
mkdir -p /data/ftp/
touch /data/ftp/test.txt
```
2. Run the following command to set the created user as the owner of the directories:
```
chown -R ftp:ftp /data/ftp/
```
3. Open the **/etc/vsftpd/vsftpd.conf** file and set the following parameters:
```
# Listen to IPv4 sockets.
listen=YES
# Determine whether to configure IPv6 listening.
listen_ipv6=YES

# Add the following parameters to the end of the configuration file:
# Set the directory where the local user resides after login.
local_root=/data/ftp/hce
# Restrict all users to the home directory.
chroot_local_user=YES
# Enable the passive mode.
pasv_enable=YES
pasv_address=<Public IP address of the FTP server>
chroot_list_enable=NO # Determine whether to allow users to access other directories.
# If chroot_list_enable is set to YES, you need to set chroot_list_file to a file that contains the users who can access other directories.
# chroot_list_file=/etc/vsftpd/chroot_list
# Set the port range that can be used in passive mode. Set a large port range to improve the security of accessing the FTP server.
# Minimum port in the available port range
pasv_min_port=<port number>
# Maximum port in the available port range
pasv_max_port=<port number>
```

NOTE

Retain the default values for other parameters.

CAUTION

The **/etc/vsftpd/chroot_list** file must be created regardless of whether **chroot_list_enable** is configured.

4. Run the following command to restart vsftpd:
```
systemctl restart vsftpd
```

Step 4 Verify vsftpd.

1. Run the following command to check the port used by vsftpd:
```
netstat -natp | grep vsftpd
```
By default, port 21 is used.
2. Configure inbound security group rules to enable port 21 and ports 5000 to 5010.
3. In the Windows file manager, enter **ftp://***<Public IP address of the FTP server>***:21** to access the FTP service. Enter the password, as shown in the following figure.

CAUTION

If the following error occurs, add **allow_writeable_chroot=YES** to the end of the **/etc/vsftpd/vsftpd.conf** file and restart vsftpd:

```
500 OOPS: vsftpd: refusing to run with writable root inside chroot()
```

----End

4 Deploying the SFTP Service

Introduction

Secure File Transfer Protocol (SFTP) is a file transfer protocol that leverages a set of utilities that provide secure access to a remote computer to deliver secure communications. It is considered by many to be the optimal method for secure file transfer. It leverages SSH and provides security and identity authentication functions of SSH. This tutorial describes how you can deploy the SFTP service in HCE 2.0.

Preparations

- Prepare an ECS and assign a public IP address or EIP to the ECS.
- Ensure that inbound security group rules allow traffic to flow to the ECS over port 22.

Procedure

Step 1 Configure the SFTP service.

1. Run the following command to create a user group named **sftp**:
```
groupadd sftp
```
2. Run the following command to create a user for accessing the SFTP service:
```
useradd -g sftp -s /bin/false usftp
```
3. Run the following command to set a password of the created user:
```
passwd usftp
```
4. Run the following command to create a home directory for the **sftp** user group:
```
mkdir -p /data/sftp/usftp
```
5. Run the following command to change the login directory of the **usftp** user:
```
usermod -d /data/sftp/usftp usftp
```
6. Open the **/etc/ssh/sshd_config** file and set the following parameters:
```
Subsystem sftp internal-sftp
# Append the following content to the end of the file:
Match user usftp # Match the usftp user.
AllowTcpForwarding no # TCP forwarding is not allowed.
X11Forwarding no # X11 forwarding is not allowed.
# Use chroot to specify /data/sftp/%u as the root directory of the user. %u indicates the user name.
ChrootDirectory /data/sftp/
ForceCommand internal-sftp # Forcibly execute internal-sftp.
```


7. Run the following command to create a test file:
   ```
   touch /data/sftp/usftp/test.txt
   ```
8. Run the following commands to set the directory permission:
   ```
   chown -R usftp:sftp /data/sftp/usftp
   chmod 755 /data/sftp/usftp
   ```

Step 2 Verify the SFTP service.

1. On the other host, run the following command to connect to the SFTP service:
   ```
   sftp usftp@<Public IP address of the SFTP service>
   ```
2. Enter the password and run the **ls** command to view the test file.
   ```
   sftp> ls
   test.txt
   ```

----End

5 Deploying PostgreSQL

Introduction

PostgreSQL is an open-source, highly-stable database system that provides SQL features, such as foreign keys, subqueries, triggers, and different user-defined data types and their functions. It further enhances the SQL language and provides some fine-grained capabilities to scale data workloads. It is mainly used in mobile, network, geospatial, and analysis applications and is known as the most advanced open source database in the industry. This tutorial describes how you can deploy PostgreSQL in HCE 2.0.

Preparations

- Prepare two ECSs and assign a public IP address or EIP to each ECS.
- Ensure that inbound security group rules allow traffic to flow to the ECSs over port 5432.

Procedure

Step 1 Configure the master PostgreSQL node.

1. Run the following command to install the server, client, and related components:
```
dnf install postgresql postgresql-contrib postgresql-server
```
2. Run the following command to initialize the database:
```
postgresql-setup --initdb --unit postgresql
```
3. Run the following commands in sequence to start PostgreSQL and check the PostgreSQL status:
```
systemctl start postgresql
systemctl status postgresql
```
If **active (running)** is displayed, PostgreSQL is started.

NOTE

To set PostgreSQL to automatically enable upon system boot, run the following commands:
```
systemctl enable postgresql
systemctl daemon-reload
```
4. Run the following command to log in to the database as user **postgres**:
```
su - postgres
```


5. Execute the following statement to access the psql terminal:
   ```
   psql
   ```
6. Execute the following statement to create an account and set the password and permission:
   ```
   CREATE ROLE replica login replication ENCRYPTED PASSWORD 'replicationxxx!';
   ```
   replica is the account name, and **replicationxxx!** is the password. **replica** has the login and replication permissions.
7. Execute the following statement to view the created account:
   ```
   SELECT usename FROM pg_user;
   ```
   Information similar to the following is displayed.
   ```
   postgres=# SELECT usename FROM pg_user;
    usename
   ----------
    postgres
    replica
   (2 rows)
   ```
8. Execute the following statement to view the permissions:
   ```
   SELECT rolname FROM pg_roles;
   ```
   Information similar to the following is displayed.
   ```
   postgres=# SELECT rolname FROM pg_roles;
              rolname
   ---------------------------
    pg_monitor
    pg_read_all_settings
    pg_read_all_stats
    pg_stat_scan_tables
    pg_read_server_files
    pg_write_server_files
    pg_execute_server_program
    pg_signal_backend
    postgres
    replica
   (10 rows)
   ```
9. Execute the following statement to exit the psql terminal:
   ```
   \q
   ```
10. Execute the following statement to exit user **postgres**:
    ```
    exit
    ```
11. Edit the **/var/lib/pgsql/data/pg_hba.conf** file to configure the **replica** user whitelist.

    Find **IPv4 local connections** and add the following information:
    ```
    host    all             all             <IPv4 address range of the slave mode>          md5
    host    replication     replica         <IPv4 address range of the slave mode>          md5
    ```
    Enter the IPv4 address range of the slave node.
12. Open the **/var/lib/pgsql/data/postgresql.conf** file and configure the following parameters:

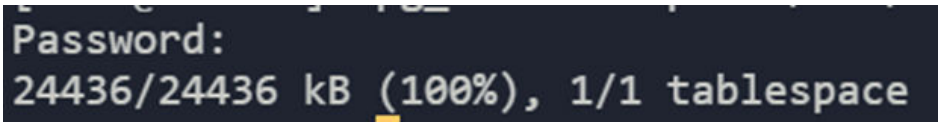

```
listen_addresses = '*' # Private IP address listened on
max_connections = 100 # Maximum number of connections. The value of max_connections of the slave node must be greater than that of the master node.
wal_level = hot_standby # Enable the hot standby mode.
synchronous_commit = on # Enable synchronous replication.
max_wal_senders = 32 # Maximum number of synchronization processes
wal_sender_timeout = 60s # Timeout for the streaming replication host to send data
```

13. Run the following command to restart PostgreSQL:
```
systemctl restart postgresql
```

Step 2 Configure the slave PostgreSQL node.

1. Run the following command to install the server, client, and related components on the slave node:
```
dnf install postgresql postgresql-contrib postgresql-server
```
2. Run the following command to create a backup directory on the slave node:
```
pg_basebackup -D /var/lib/pgsql/data -h <IP address of the master node> -p 5432 -U replica -X stream -P -R
```
-X indicates the required WAL stream. **-P** indicates the progress. **-h** indicates the IP address of the master node. **-p** indicates the port number of the master node. **-U** indicates the account for logging in to the master node. **-R** indicates that the replication configuration is automatically written.

If information similar to the following is displayed, the backup is successful.

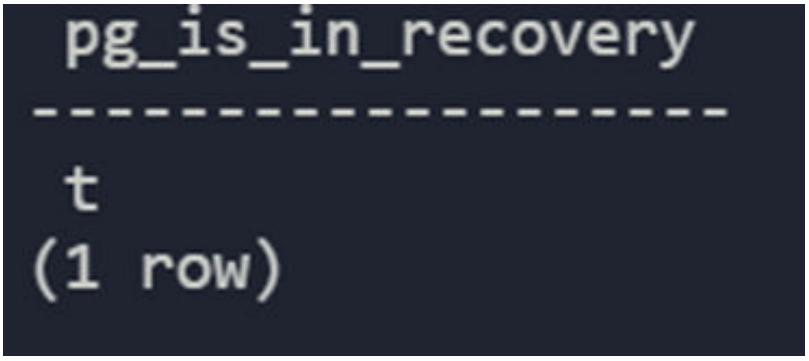


The **standby.signal** file is generated in the **/var/lib/pgsql/data** directory, and information about the master database connections is automatically written into the **postgresql.auto.conf** file.

3. Run the following command to modify the user group that the directory belongs to:
```
chown -R postgres.postgres /var/lib/pgsql/data
```
4. Run the following command to restart PostgreSQL:
```
systemctl start postgresql
```

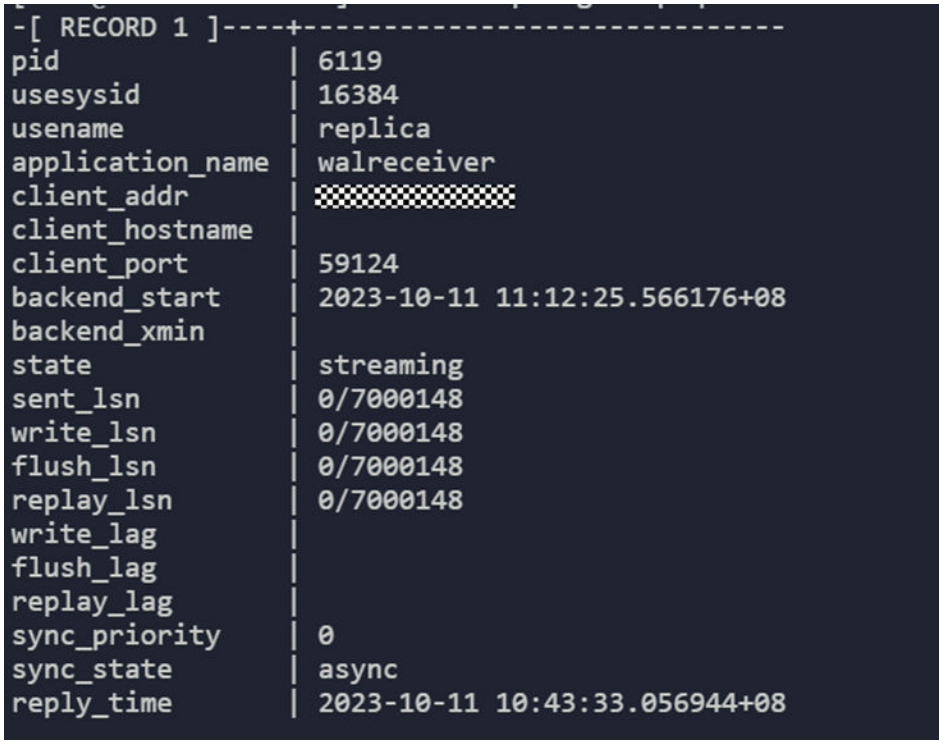
Step 3 Check whether the configuration is successful.

1. Run the following command on the slave node to check whether the node has become the slave node:
```
sudo -u postgres psql -c "SELECT pg_is_in_recovery()"
```
If **t** is displayed, the node has become the slave node.



2. Run the following command on the master node to obtain the slave node information:
```
sudo -u postgres psql -x -c "SELECT * FROM pg_stat_replication" -d postgres
```

If information similar to the following is displayed, the configuration is successful.



----End

CAUTION

The password and permission configurations are used only for tests. Exercise caution when using them in the service environment.

6 Deploying Redis

Introduction

Remote Dictionary Server (Redis) is an open-source, in-memory, distributed, and durable key-value pair storage database written in C. Redis is a storage system with rich functions and applies to various scenarios, including caching, session storage, rankings, and real-time analysis. It is widely used and has active community support. This tutorial describes how you can deploy Redis in HCE 2.0.

Preparations

- Prepare two ECSs and assign a public IP address or EIP to each ECS. One ECS functions as the master Redis node, and the other serves as the slave Redis node.
- Ensure that inbound security group rules allow traffic to flow to the ECSs over port 6379.

Procedure

Step 1 Install and configure Redis.

1. Run the following command on the two ECSs to install Redis:
   ```
   dnf install redis -y
   ```
2. Run the following command on the two ECSs to start Redis:
   ```
   systemctl start redis
   ```

   NOTE

   To set Redis to automatically enable upon system boot, run the following command:
   ```
   systemctl enable redis
   ```
3. Run the following command to check the Redis status:
   ```
   systemctl status redis
   ```
   If **active (running)** is displayed, Redis is started.
4. On the master node, edit the **/etc/redis.conf** file and configure the following attributes:
   ```
   bind 0.0.0.0      # Replace the value with the IP address of the master node. In this example, enter any IP address.
   requirepass ******* # Set a password.
   ```
5. On the slave node, edit the **/etc/redis.conf** file and configure the following attributes:


```
bind 0.0.0.0                    # Replace the value with the IP address of the slave node. In this example, enter any IP address.
requirepass *******             # Set a password.
slaveof <IP address of the master node> <Port of the master node>
masterauth <Password of the master node>
```

6. Run the following command on the two ECSs to restart Redis:

```
systemctl restart redis
```

Step 2 Verify Redis.

1. Run the following command on the master node to connect to Redis:

```
redis-cli
auth <Password>
```

2. Run the following command to view the node information:

```
127.0.0.1:6379> info replication
# Replication
role:master
connected_slaves:1
slave0:ip=x.x.x.x,port=6379,state=online,offset=4382,lag=0
master_replid:5d68ccf7722f461cc5f004c7e96fd7c506990508
master_replid2:0000000000000000000000000000000000000000
master_repl_offset:4382
second_repl_offset:-1
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:1
repl_backlog_histlen:4382
127.0.0.1:6379>
```

----End

CAUTION

The preceding configuration and Redis deployment architecture are used only for tests. Exercise caution when using them in the service environment.

7 Installing Tomcat

Introduction

Apache Tomcat (or simply Tomcat) is an open source web server and servlet container developed by the Apache Software Foundation. This tutorial describes how you can deploy Tomcat in HCE 2.0.

Preparations

- Prepare an ECS and assign a public IP address or EIP to the ECS.
- Ensure that inbound security group rules allow traffic to flow to the ECS over port 8080.

Procedure

Step 1 Install Tomcat.

1. Run the following command to install Java:
   ```
   dnf install java-1.8.0-openjdk
   ```
2. Run the following command to check whether the installation is successful:
   ```
   java -version
   ```
3. Run the following command to install Tomcat:
   ```
   dnf install tomcat
   ```
   The Tomcat is installed in the **/usr/share/tomcat** directory.

Step 2 Configure Tomcat.

1. Add the following content to the **/etc/profile** file:
   ```
   JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.342.b07-0.hce2.x86_64/jre
   PATH=$PATH:$JAVA_HOME/bin
   CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
   export JAVA_HOME CLASSPATH PATH
   ```
2. Run the following command to activate the preceding environment variables:
   ```
   source /etc/profile
   ```
3. Clear the **/usr/share/tomcat/conf/server.xml** file and paste the following content:
   ```
   <?xml version="1.0" encoding="UTF-8"?> <Server port="8006" shutdown="SHUTDOWN"> <Listener
   className="org.apache.catalina.core.JreMemoryLeakPreventionListener"/> <Listener
   className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener"/> <Listener
   className="org.apache.catalina.core.ThreadLocalLeakPreventionListener"/> <Listener
   className="org.apache.catalina.core.AprLifecycleListener"/> <GlobalNamingResources> <Resource
   ```



```
name="UserDatabase" auth="Container"  type="org.apache.catalina.UserDatabase"
description="User database that can be updated and saved"
factory="org.apache.catalina.users.MemoryUserDatabaseFactory"  pathname="conf/tomcat-
users.xml"/> </GlobalNamingResources> <Service name="Catalina"> <Connector port="8080"
protocol="HTTP/1.1"  connectionTimeout="20000"  redirectPort="8443"  maxThreads="1000"
minSpareThreads="20"  acceptCount="1000"  maxHttpHeaderSize="65536"  debug="0"
disableUploadTimeout="true"  useBodyEncodingForURI="true"  enableLookups="false"
URIEncoding="UTF-8"/> <Engine name="Catalina" defaultHost="localhost"> <Realm
className="org.apache.catalina.realm.LockOutRealm"> <Realm
className="org.apache.catalina.realm.UserDatabaseRealm"   resourceName="UserDatabase"/> </
Realm> <Host name="localhost" appBase="/data/wwwroot/default" unpackWARs="true"
autoDeploy="true"> <Context path="" docBase="/data/wwwroot/default" debug="0"
reloadable="false" crossContext="true"/> <Valve
className="org.apache.catalina.valves.AccessLogValve" directory="logs"
prefix="localhost_access_log." suffix=".txt" pattern="%h %l %u %t %s %b" /> </Host> </Engine> </
Service> </Server>
```

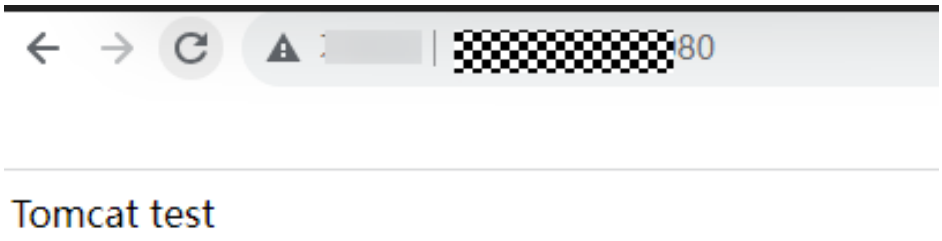
Save the settings and exit. You can customize the preceding configuration as required.

4. Run the following command to create the directories specified by **appbase** and **docbase** in the preceding configuration:
   ```
   mkdir -p /data/wwwroot/default
   ```
5. Run the following command to set **tomcat** as the owner of the preceding directory:
   ```
   chown -R tomcat.tomcat /data/wwwroot/
   ```
6. Create the **/usr/share/tomcat/bin/setenv.sh** file and enter the following information to set JVM memory parameters:
   ```
   JAVA_OPTS='-Djava.security.egd=file:/dev/./urandom -server -Xms256m -Xmx496m -
   Dfile.encoding=UTF-8'
   ```
7. Run the following command to start Tomcat:
   ```
   systemctl start tomcat
   ```
8. Run the following command to check the Tomcat status:
   ```
   systemctl status tomcat
   ```

   If **active (running)** is displayed, Tomcat is started.

Step 3 Check whether Tomcat is installed.

1. Run the following command to create a test page:
   ```
   echo Tomcat test > /data/wwwroot/default/index.jsp
   ```
2. Enter **http://***<Public IP address of the Tomcat service>***:8080** in the address box of the browser. If the following page is displayed, the installation is successful.



----End

CAUTION

The preceding configuration is used only for tests. Exercise caution when using the configuration in the service environment.

8 Installing Kafka

Introduction

Kafka is distributed pub/sub messaging middleware with high throughput, data persistence, horizontal scalability, and stream data processing. Common applications include log collection, data streaming, online/offline system analytics, and real-time monitoring.

This tutorial describes how you can deploy Kafka in HCE 2.0.

Preparations

- Prepare an ECS and assign a public IP address or EIP to the ECS.
- Ensure that inbound security group rules allow traffic to flow to the ECS over port 9092.

Procedure

Step 1 Install Kafka.

Run the following command to install Kafka:

```
dnf install kafka
```

After the command is executed, Kafka is installed in the **/opt/kafka** directory.

Step 2 Configure Kafka.

1. Open the **/opt/kafka/config/server.properties** file and modify the following attributes:
   ```
   listeners=PLAINTEXT://<Private IP address>:9092
   advertised.listeners=PLAINTEXT://<Public IP address>:9092
   ```
2. Create the **/lib/systemd/system/zookeeper.service** file and enter the following information:
   ```
   [Unit]
   Description=Zookeeper service
   After=network.target

   [Service]
   Type=simple
   Environment="PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
   User=root
   Group=root
   ```



```
ExecStart=/opt/kafka/bin/zookeeper-server-start.sh /opt/kafka/config/zookeeper.properties
ExecStop=/opt/kafka/bin/zookeeper-server-stop.sh
Restart=on-failure
SuccessExitStatus=143

[Install]
WantedBy=multi-user.target
```

3. Create the **/lib/systemd/system/kafka.service** file and enter the following information:

```
[Unit]
Description=Apache Kafka server (broker)
After=network.target  zookeeper.service

[Service]
Type=simple
Environment="PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
User=root
Group=root
ExecStart=/opt/kafka/bin/kafka-server-start.sh /opt/kafka/config/server.properties
ExecStop=/opt/kafka/bin/kafka-server-stop.sh
Restart=on-failure
SuccessExitStatus=143

[Install]
WantedBy=multi-user.target
```

4. Run the following commands in sequence to start Kafka and ZooKeeper:

```
systemctl daemon-reload
systemctl start zookeeper
systemctl start kafka
```

Step 3 Verify Kafka.

1. Run the following command to create a topic:

```
 /opt/kafka/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions
1 --topic test
```

2. Run the following command to view the created topic:

```
/opt/kafka/bin/kafka-topics.sh --list --zookeeper localhost:2181
```

   If **test** is displayed, Kafka is deployed.

----End

CAUTION

The preceding configuration is used only for tests. Exercise caution when using the configuration in the service environment.

9 Installing GNOME

Introduction

GNOME (GNU Network Object Model Environment) is a graphical user interface (GUI) and set of computer desktop applications for users of the Linux operating system. This tutorial describes how you can install GNOME in HCE 2.0.

Preparations

Prepare an ECS.

Procedure

Step 1 Install GNOME.

Run the following command to install GNOME:

```
dnf install gnome-initial-setup gnome-terminal
```

Step 2 Set GNOME to start upon system boot.

1. Run the following command to set GNOME to start upon system boot:
   ```
   systemctl set-default graphical.target
   ```
2. Run the following command to restart GNOME:
   ```
   reboot
   ```

----End

10 Installing Apache

Introduction

Apache HTTP Server (Apache for short) is a free open-source cross-platform web server software developed by the Apache Software Foundation. It can run in most operating systems and is widely used because of its high security. As one of the most popular web server software, Apache offers features such as support for FastCGI and SSL, and integration with Perl. This tutorial describes how you can deploy Apache in HCE 2.0.

Preparations

- Prepare an ECS and assign a public IP address or EIP to the ECS.
- Ensure that inbound security group rules allow traffic to flow to the ECS over port 80.

Procedure

Step 1 Install Apache.

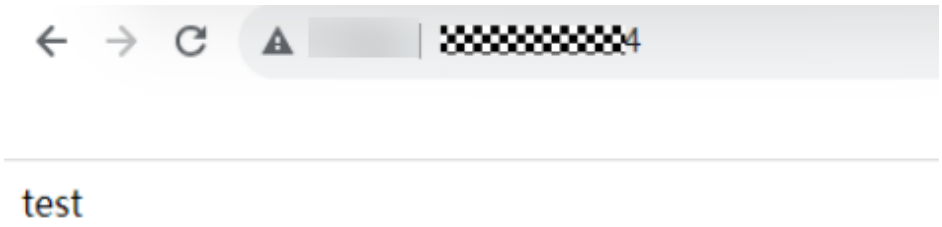
1. Run the following command to install Apache:
   ```
   dnf install httpd httpd-devel
   ```
   After the command is executed, the Apache configuration file is **/etc/httpd/conf/httpd.conf**.
2. Run the following command to check the Apache version:
   ```
   httpd -v
   ```
   Information similar to the following is displayed:
   ```
   [root@localhost system]# httpd -v
   Server version: Apache/2.4.51 (Unix)
   Server built:   Feb  9 2022 09:00:41
   ```
3. Run the following command to start Apache:
   ```
   systemctl start httpd
   ```
   If **active (running)** is displayed, Apache is started.

   NOTE

   To set Apache to automatically enable upon system boot, run the following commands in sequence:
   ```
   systemctl enable httpd
   systemctl daemon-reload
   ```


Step 2 Verify Apache.

1. Run the following command to create a test page:
   ```
   echo test > /var/www/html/index.html
   ```
2. Enter **http://***<Public IP address>* in the address box of the browser. If the following figure is displayed, the deployment is successful.



test

----End

11 Deploying Django

Introduction

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. This tutorial describes how you can use Nginx and uWSGI to deploy the Django project in HCE 2.0.

Preparations

- Prepare an ECS and assign a public IP address or EIP to the ECS.
- Ensure that inbound security group rules allow traffic to flow to the ECS over ports 80, 8001, and 8002.

Procedure

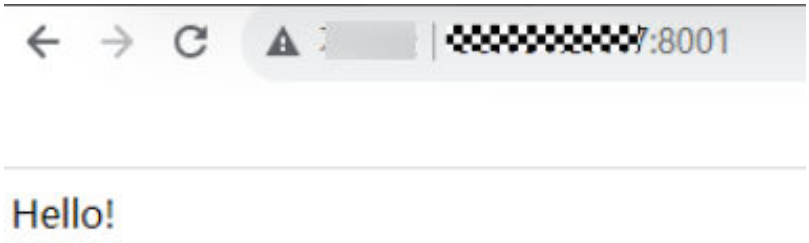
Step 1 Install Nginx.

1. Run the following command to install Nginx:
   ```
   dnf install nginx
   ```
2. Run the following command to start Nginx:
   ```
   systemctl start nginx
   ```
3. Run the following command to check the Nginx status:
   ```
   systemctl status nginx
   ```
   If **active (running)** is displayed, Nginx is started.

Step 2 Install uWSGI.

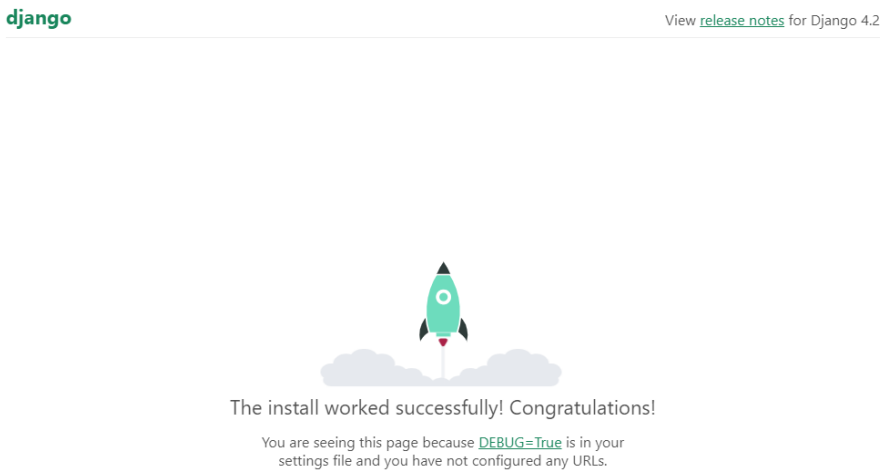
1. Run the following command to install the uWSGI dependency:
   ```
   dnf install python3-devel gcc
   ```
2. Run the following command to install uWSGI:
   ```
   pip install uwsgi
   ```
3. Run the following command to check the uWSGI version:
   ```
   uwsgi --version
   ```
4. Edit the **hello.py** file and enter the following information:
   ```
   def application(env, reply):
       reply('200 ok',[('Content-Type','text/html')])
       return [b"Hello!"]
   ```
5. Run the following command to start uWSGI:
   ```
   uwsgi --http :8001 --wsgi-file hello.py
   ```


6. Enter **http://***<Public IP address>***:8001** in the address box of the browser to access uWSGI.



Step 3 Install the Django environment.

1. Run the following command to install Django:
```
pip install Django
```
2. Run the following command to initialize the project:
```
python -m django startproject django_project
```
3. Go to the project directory, edit the **django_project/settings.py** configuration file, and change the value of **ALLOWED_HOSTS** to the following:
```
ALLOWED_HOSTS = ["*"]
```
4. Run the following command to start Django:
```
python manage.py runserver 0.0.0.0:8002
```
5. Enter **http://***<Public IP address>***:8002** in the address box of the browser to access Django.



Step 4 Configure the environment.

1. Edit the **django_project/settings.py** file.
   a. Enter the following statement at the beginning of the file to import the OS library:
   ```
   import os
   ```
   b. Add the following parameter to the end of the file:
   ```
   STATIC_ROOT = os.path.join(BASE_DIR, "static/")
   ```
   c. Run the following command to collect all static files:
   ```
   sudo python manage.py collectstatic
   ```
   Information similar to the following is displayed:

```
[root@hce2 django_project]# sudo python manage.py collectstatic

125 static files copied to '/root/django_project/static'.
```

A **static** directory is added to the project directory. The project structure is as follows:

```
[root@hce2 django_project]# ls
db.sqlite3  django_project  manage.py  static
```

2. Edit the **/etc/nginx/nginx.conf** file to configure Nginx.
   a. Find the **http** attribute and add the following content:

```
upstream django {
    server 127.0.0.1:8001;
}
```

   b. Find the **server** attribute under **http** and change it to the following:

```
server {
    listen       80;
    server_name  django_project;

    charset      utf-8;
    location  /static {
        autoindex on;
        alias /root/django_project/static;
    }
    location / {
        uwsgi_pass 127.0.0.1:8001;
        include uwsgi_params;
        include /etc/nginx/uwsgi_params;
        uwsgi_param UWSGI_SCRIPT iCourse.wsgi;
        uwsgi_param UWSGI_CHDIR /iCourse;
        index index.html index.htm;
        client_max_body_size 35m;
        index index.html index.htm;
}
```

   The following is displayed.


```
http {
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 4096;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
    # for more information.
    include /etc/nginx/conf.d/*.conf;

    upstream django {
        server 127.0.0.1:8001;
    }

    server {
        listen       80;
        server_name  django_project;

        charset      utf-8;
        location  /static {
                autoindex on;
                alias /root/django_project/static;
        }
        location / {
                uwsgi_pass 127.0.0.1:8001;
                include uwsgi_params;
                include /etc/nginx/uwsgi_params;
                uwsgi_param UWSGI_SCRIPT iCourse.wsgi;
                uwsgi_param UWSGI_CHDIR /iCourse;
                index index.html index.htm;
                client_max_body_size 35m;
                index index.html index.htm;
        }
    }
```

3. Create the **uwsgi_config.ini** file in the project directory and enter the following:

```
[uwsgi]
socket = 127.0.0.1:8001 # Port 8001 must be the same as the uwsgi_pass port defined in the Nginx configuration file.
chdir = /root/django_project/ # Specify the project directory. In this example, the directory is /root/django_project/. Modify it based on the project.
wsgi-file = django_project/wsgi.py # Specify the Django's wsgi file and modify it based on the project.
processes = 4 # Maximum number of working processes
threads = 2 # Number of threads started after each working process is started
vacuum = true # Automatic cleanup upon environment exit
buffer-size = 65536 # Set the size of the internal buffer for parsing uwsgi packets to 64 KB. The default value is 4 KB.
```

CAUTION

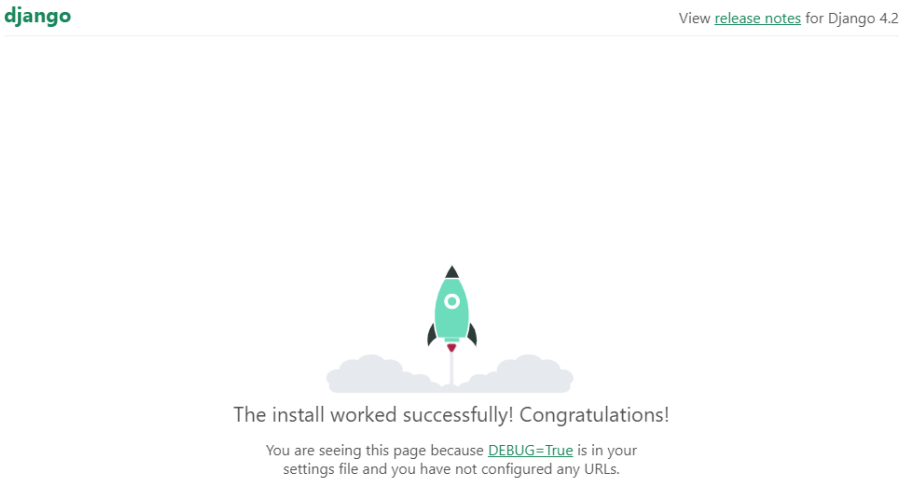
In the actual environment, delete the logs in the configuration file.

Step 5 Verify Django.

1. Run the following command to restart Nginx:

```
systemctl restart nginx
```


2. Run the following command in the project directory to start uWSGI:
```
uwsgi --ini uwsgi_config.ini
```
3. Enter **http://***<Public IP address>* in the address box of the browser to access the Django page.



----End

CAUTION

The preceding configuration is used only for tests. Exercise caution when using the configuration in the service environment.

12 Installing lnmp Online

Introduction

lnmp is compiled using Linux Shell and can be used to install LNMP (Nginx, MySQL, and PHP), LNMPA (Nginx, MySQL, PHP, and Apache), or LAMP (Apache, MySQL, and PHP) for a variety of Linux servers or independent servers.

This tutorial describes how you can install lnmp in HCE 2.0.

Preparations

Prepare an ECS and assign a public IP address or EIP to the ECS.

Procedure

Step 1 Run the following command to download lnmp2.0 and decompress it to the directory:

```
wget http://soft.vpser.net/lnmp/lnmp2.0.tar.gz -O lnmp2.0.tar.gz && tar zxf lnmp2.0.tar.gz && cd lnmp2.0
```

Step 2 Run the following command to modify the **main.sh** file:

```
vim include/main.sh
```

Add an **elif** judgment to the **Get_Dist_Name** function.

```
elif grep -Eqi "Huawei Cloud EulerOS" /etc/issue || grep -Eq "Huawei Cloud EulerOS" /etc/*-release; then
    DISTRO='HCE'
    PM='yum'
```

Step 3 Run the following command to modify the **init.sh** file:

```
vim include/init.sh
```

Add the following to the **if** judgment of the **Install_Freetype** function:

```
echo "${HCE_Version}" | grep -Eqi "^2\.[0-9]" ||
```



```
Install_Freetype()
{
    if echo "${HCE_Version}" | grep -Eqi "^2\.[0-9]" ||echo "${Ubuntu_Version}" | grep -Eqi "^1[89]\.|2[0-9]\." ||
        Download_Files ${Download_Mirror}/lib/freetype/${Freetype_New_Ver}.tar.xz ${Freetype_New_Ver}.tar.xz
        Echo_Blue "[+] Installing ${Freetype_New_Ver}"
        Tar_Cd ${Freetype_New_Ver}.tar.xz ${Freetype_New_Ver}
        ./configure --prefix=/usr/local/freetype --enable-freetype-config
```

Step 4 Run the following command to start the installation:

```
./install.sh lnmp
```

If the following information is displayed, the installation is successful.

```
Checking ...
Nginx: OK
MySQL: OK
PHP: OK
PHP-FPM: OK
Clean Web Server src directory...
+------------------------------------------------------------------------+
|          LNMP V2.0 for HCE Linux Server, Written by Licess             |
+------------------------------------------------------------------------+
|           For more information please visit https://lnmp.org           |
+------------------------------------------------------------------------+
|    lnmp status manage: lnmp {start|stop|reload|restart|kill|status}    |
+------------------------------------------------------------------------+
|  phpMyAdmin: http://IP/phpmyadmin/                                     |
|  phpinfo: http://IP/phpinfo.php                                        |
|  Prober:  http://IP/p.php                                              |
+------------------------------------------------------------------------+
|  Add VirtualHost: lnmp vhost add                                       |
+------------------------------------------------------------------------+
|  Default directory: /home/wwwroot/default                              |
+------------------------------------------------------------------------+
|  MySQL/MariaDB root password: 123456                                   |
+------------------------------------------------------------------------+
+-----------------------------------------+
|    Manager for LNMP, Written by Licess  |
+-----------------------------------------+
|              https://lnmp.org           |
+-----------------------------------------+
nginx (pid 244532) is running...
php-fpm is runing!
 SUCCESS! MySQL running (245151)
State  Recv-Q Send-Q Local Address:Port Peer Address:PortProcess
LISTEN 0      511          0.0.0.0:80        0.0.0.0:*
LISTEN 0      511          0.0.0.0:80        0.0.0.0:*
LISTEN 0      128          0.0.0.0:22        0.0.0.0:*
LISTEN 0      150                *:3306            *:*
LISTEN 0      128             [::]:22           [::]:*
Install lnmp takes 25 minutes.
Install lnmp V2.0 completed! enjoy it.
```

----End

13 Compiling qperf to Support IPv6

Introduction

qperf 0.4.9 in HCE 2.0 does not support IPv6. To use IPv6, you need to obtain qperf 0.4.11 or later from the community and then upgrade qperf 0.4.9 to qperf 0.4.11 or later.

This tutorial describes how you can download and compile the qperf 0.4.11 source code in HCE 2.0.

Preparations

Prepare an ECS and assign a public IP address or EIP to the ECS.

Procedure

Step 1 Run the following command to download the qperf 0.4.11 source code package:

```
wget https://github.com/linux-rdma/qperf/archive/refs/tags/v0.4.11.tar.gz
```

Step 2 Run the following command to decompress the package:

```
tar -xf v0.4.11.tar.gz
```

After the decompression, the **qperf-0.4.11** directory is generated.

Step 3 Run the following command to install the dependency:

```
dnf install gcc make automake
```

Step 4 Go to the **qperf-0.4.11** directory and run the following commands in sequence for compilation:

```
./cleanup
./autogen.sh
./configure
make
```

Step 5 Verify the compilation result.

The binary files obtained after compilation are stored in the **qperf-0.4.11/src** directory. Go to the directory and run the **./qperf –version** command. If information similar to the following is displayed, the compilation is successful and qperf is upgraded from 0.4.9 to 0.4.11.


```
[root@localhost src]# ./qperf --version
qperf 0.4.11
[root@localhost src]#
```

----End

