

HCE 最佳实践

文档版本 01
发布日期 2024-12-06



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 部署 MySQL.....	1
2 安装 Docker.....	4
3 搭建 FTP 站点.....	7
4 搭建 SFTP.....	10
5 PostgreSQL 部署.....	12
6 Redis 一主多从部署.....	16
7 Tomcat 安装.....	18
8 安装 Kafka.....	20
9 Gnome 桌面安装.....	22
10 Apache 安装.....	23
11 部署 Django 服务.....	25
12 在线安装 lnmp.....	30
13 qperf 编译支持 IPV6.....	32

1 部署 MySQL

简介

MySQL 是一种开源关系型数据库管理系统。与其他关系型数据库一样，MySQL 将数据存储在由行和列组成的表中。用户可以使用结构化查询语言（通常称为 SQL）定义、操作、控制和查询数据。本教程介绍如何在 HCE 2.0 上安装部署 MySQL。

准备工作

- 准备一台 ECS 实例，并分配公网 IP 或者弹性公网 IP（EIP）。
- 实例安全组的入方向规则已放通 22、3306 端口。

操作步骤

步骤1 安装MySQL服务端和客户端。

1. 执行以下命令安装MySQL服务端和客户端。

```
dnf install mysql-server mysql-common mysql -y
```

2. 执行以下命令查看MySQL的版本号。

```
mysql -V
```

如果返回如下类似信息，表示安装成功。

```
mysql Ver 8.0.37 for Linux on x86_64 (Source distribution)
```

步骤2 配置MySQL。

1. 运行以下命令，启动并将MySQL服务设置为开机自启动。

```
systemctl start mysqld
```

2. 执行以下命令查看服务状态。

```
systemctl status mysqld
```

如果显示 active (running) 则表示启动成功。

说明

如果要将 mysqld 设置为开机自启动，则需要执行如下命令。

```
systemctl enable mysqld
```

3. 执行以下命令，对MySQL进行安全配置。

```
mysql_secure_installation
```

根据个人需要和提示信息，自定义选项内容。

```
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 // 是否设置密码组件

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 // 选择密码安全策略
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 into 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 // 是否删除匿名用户
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 // 是否禁止root账户远程登录
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 // 是否删除测试数据库
- 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 // 是否重新加载权限表
Success.

All done!
```

步骤3 远程连接数据库。

1. 执行以下命令在MySQL server端连接数据库。

```
mysql -uroot -p
```

然后输入上述为root用户创建的密码。

2. 输入以下语句创建测试账号和密码。
`create user 'test'@'%' identified by 'test****';`
其中**test**为账号，**test******为对应密码。
3. 输入以下语句为test账号授权数据库所有权限。
`grant all privileges on *.* to 'test'@'%';`
4. 输入以下语句刷新权限。
`flush privileges;`
5. 执行以下语句退出数据库。
`exit`
6. 在另一台服务器上安装MySQL client后输入以下命令进行远程连接。
`mysql -h <MySQL server IP> -utest -p`
或者使用Navicat、VSCode插件进行远程连接。

----结束



注意

以上关于密码、权限相关配置仅用于测试，业务环境请谨慎使用

2 安装 Docker

简介

Docker是一款开源的应用容器引擎，具有可移植性、可扩展性、高安全性和可管理性等优势。开发者可将应用程序和依赖项打包到一个可移植的容器中，快速发布到Linux机器上并实现虚拟化，实现更高效地构建、部署和管理应用程序。本教程介绍如何在HCE 2.0服务器中安装Docker。

准备工作

准备一台ECS实例。

操作步骤

步骤1 安装Docker软件

1. 执行以下命令安装Docker。
`dnf install docker`
2. 执行以下命令启动Docker服务。
`systemctl start docker`

步骤2 验证Docker是否安装成功。

1. 执行以下命令运行hello-world镜像。
`docker run hello-world`
2. 如果显示类似以下内容，表示安装成功。

```
[root@secure ~]# docker run helloworld
Unable to find image 'helloworld:latest' locally
docker: Error response from daemon: pull access denied for helloworld, repository name does not match image name. 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: sha256:88ec8acaa3ec199d3b7eaf73588f4518c25f9d34f58ce9a0df68429c5af48e8d
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/
```

步骤3 Docker的使用。

1. 管理Docker守护进程。

- 运行守护进程

```
systemctl start docker
```

- 停止守护进程

```
systemctl stop docker
```

- 重启守护进程

```
systemctl restart docker
```

2. 管理镜像。

- 修改标签

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

- 查看已有镜像

```
docker images
```

- 强制删除镜像

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

3. 管理容器

- 进入容器

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

- 退出容器

```
exit
```

- 进入后台运行的容器

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

- 将容器做成镜像

```
docker commit <容器ID> [<仓库名>[:<标签>]]
```

比如：

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

⚠ 注意

上述方法安装的docker版本为18.09.0，如果需要安装更高版本的Docker，请参考[docker文档](#)进行安装。

----结束

3 搭建 FTP 站点

简介

vsftpd (very secure FTP daemon) 是Linux下的一款小巧轻快、安全易用的FTP服务器软件。本教程介绍如何在HCE 2.0上安装并配置vsftpd。

准备工作

- 准备一台ECS实例，并分配公网IP或者弹性公网IP（EIP）。
- 安全组的入方向规则已放行21端口。

操作步骤

步骤1 安装vsftpd软件并启动。

1. 执行如下命令安装vsftpd。
`dnf install vsftpd`
2. 执行如下命令启动vsftpd服务。
`systemctl start vsftpd`
3. 执行如下命令查看服务状态。
`systemctl status vsftpd`

如果显示**active (running)**则表示启动成功，类似效果如下图。

```
[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
```

📖 说明

如果需要将vsftpd设置为开机自启动，则需要再执行如下命令。

```
systemctl enable vsftpd
```

步骤2 为FTP服务添加用户。

1. 执行如下命令为FTP服务创建一个用户。
`adduser ftp`
2. 执行如下命令为ftp用户设置密码。

```
passwd ftp
```

步骤3 配置vsftpd服务。

1. 执行如下命令为FTP服务创建目录和文件（可根据个人需要修改目录）。

```
mkdir -p /data/ftp/  
touch /data/ftp/test.txt
```

2. 执行如下命令将上述目录的拥有者设置为新用户。

```
chown -R ftp:ftp /data/ftp/
```

3. 打开“/etc/vsftpd/vsftpd.conf”文件，找到如下参数进行配置。

```
# 监听IPv4 sockets。  
listen=YES  
# 根据需要选择是否配置监听IPv6  
listen_ipv6=YES  
  
# 在配置文件的末尾添加下列参数：  
# 设置本地用户登录后所在目录。  
local_root=/data/ftp/hce  
#全部用户被限制在主目录。  
chroot_local_user=YES  
#开启被动模式。  
pasv_enable=YES  
pasv_address=<FTP服务器公网IP地址>  
chroot_list_enable=NO # 是否允许用户访问其他目录  
# 如果chroot_list_enable配置为YES，则需要配置该选项；该配置为一个文件，包含哪些用户可以访问其他目录  
# chroot_list_file=/etc/vsftpd/chroot_list  
# 设置被动模式下，可使用的端口范围，建议把端口范围设置在一段比较高的范围内，有助于提高访问FTP服务器的安全性。  
# 可使用的端口范围的最小值  
pasv_min_port=<port number>  
# 可使用的端口范围的最大值。  
pasv_max_port=<port number>
```

📖 说明

除上述提及的参数，其他参数保持默认值即可。

⚠️ 注意

无论是否配置chroot_list_enable，都需要创建“/etc/vsftpd/chroot_list”文件。

4. 执行如下命令重启vsftpd服务。

```
systemctl restart vsftpd
```

步骤4 验证。

1. 执行如下命令查看vsftpd服务所使用的端口。

```
netstat -natp | grep vsftpd
```

默认情况下所使用的端口是21。

2. 在安全组界面，入方向放开21端口和上述配置的5000~5010端口。
3. 在windows文件管理器中输入ftp://<FTP服务器公网IP>:21访问FTP服务，输入密码后效果如下图。



⚠ 注意

如果遇到以下错误，需要在“/etc/vsftpd/vsftpd.conf”文件末尾增加 allow_writeable_chroot=YES配置，然后再重启vsftpd服务。

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

----结束

4 搭建 SFTP

简介

SFTP (Secure File Transfer Protocol) 是一种用于安全访问、传输和管理大文件和敏感数据的网络协议，它通过SSH协议运行，能够支持SSH完整的安全和身份验证功能。本教程介绍如何在HCE 2.0上部署SFTP服务。

准备工作

- 准备一台ECS实例，并分配公网IP或者弹性公网IP（EIP）。
- 安全组的入方向规则已放行22端口。

操作步骤

步骤1 配置SFTP服务。

1. 执行如下命令创建sftp组。
`groupadd sftp`
2. 执行如下命令，创建一个用户，用户登录SFTP服务。
`useradd -g sftp -s /bin/false usftp`
3. 执行如下命令为上述新用户设置密码。
`passwd usftp`
4. 执行如下命令为sftp用户组创建home目录。
`mkdir -p /data/sftp/usftp`
5. 执行如下命令修改usftp用户登入时的目录。
`usermod -d /data/sftp/usftp usftp`
6. 打开/etc/ssh/sshd_config文件，找到以下参数进行配置。
Subsystem sftp internal-sftp
在文件末尾添加如下配置：
Match user usftp # 匹配usftp用户
AllowTcpForwarding no # 不允许转发TCP协议
X11Forwarding no # 不允许进行 X11 转发
使用chroot将用户根目录指定到/data/sftp/%u，%u代表用户名
ChrootDirectory /data/sftp/
ForceCommand internal-sftp # 强制执行内部sftp
7. 执行如下命令，创建测试文件。
`touch /data/sftp/usftp/test.txt`
8. 依次执行如下命令设置目录权限。
`chown -R usftp:sftp /data/sftp/usftp`
`chmod 755 /data/sftp/usftp`

步骤2 验证。

1. 在另外一台机器上，输入以下命令连接上述SFTP服务。
sftp usftp@<SFTP服务公网IP>
2. 输入密码后执行ls命令查看测试文件，效果如下图。
sftp> ls
test.txt

----结束

5 PostgreSQL 部署

简介

PostgreSQL是一个开源的、高度稳定的数据库系统，为SQL的不同功能提供支持，如外键、子查询、触发器和不同的用户定义的类型和功能。它进一步增强了SQL语言，提供了一些细致地扩展数据工作负载的功能。它主要用于移动、网络、地理空间和分析应用程序领域，被业界誉为“最先进的开源数据库”。本教程介绍如何在HCE 2.0上安装部署PostgreSQL。

准备工作

- 准备两台ECS实例，并分配公网IP或者弹性公网IP（EIP）。
- 安全组入方向都已放开5432端口。

操作步骤

步骤1 配置PostgreSQL主节点。

1. 执行如下命令安装服务端、客户端和相关组件：

```
dnf install postgresql postgresql-contrib postgresql-server
```
2. 执行如下命令初始化数据库：

```
postgresql-setup --initdb --unit postgresql
```
3. 依次执行如下命令启动服务并查看服务状态：

```
systemctl start postgresql  
systemctl status postgresql
```

如果显示**active (running)**则表示服务启动成功。

📖 说明

如果需要将PostgreSQL服务设置为开机自启动，则需要执行以下命令。

- ```
systemctl enable postgresql
systemctl daemon-reload
```
4. 执行以下命令登录postgres账号。

```
su - postgres
```
  5. 执行以下语句进入psql终端。

```
psql
```
  6. 输入以下语句创建账号，并配置密码和权限。

```
CREATE ROLE replica login replication ENCRYPTED PASSWORD 'replicationxxx!';
```

其中账号为**replica**，权限(**login**和**replication**)为登录和备份权限，密码为**replicationxxx!**。

7. 输入以下语句查看创建的账号。

```
SELECT username FROM pg_user;
```

返回结果类似下图则表示创建成功。

```
postgres=# SELECT username FROM pg_user;
username

postgres
replica
(2 rows)
```

8. 输入以下语句查看创建的权限。

```
SELECT rolname FROM pg_roles;
```

返回结果类似如下图则表示创建成功。

```
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. 执行以下语句退出psql终端。

```
\q
```

10. 执行以下语句退出postgres用户。

```
exit
```

11. 编辑“/var/lib/pgsql/data/pg\_hba.conf”文件配置replica用户白名单。

找到IPv4 local connections部分，添加以下信息：

```
host all all <从节点的IPv4网段> md5
host replication replica <从节点的IPv4网段> md5
```

将自己部署的从节点IPv4网段信息填入上述对应位置。

12. 编辑“/var/lib/pgsql/data/postgresql.conf”文件，找到如下参数，进行配置：

```
listen_addresses = '*' #监听的内网 IP 地址
max_connections = 100 #最大连接数，从库的 max_connections 必须要大于主库的
wal_level = hot_standby #启用热备模式
synchronous_commit = on #开启同步复制
max_wal_senders = 32 #同步最大的进程数量
wal_sender_timeout = 60s #流复制主机发送数据的超时时间
```

13. 最后执行以下命令重启服务。

```
systemctl restart postgresql
```

**步骤2** 配置PostgreSQL从节点。

1. 执行以下命令在从节点安装服务端、客户端和相关组件。

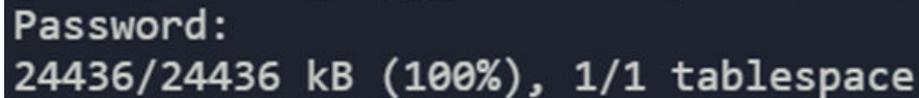
```
dnf install postgresql postgresql-contrib postgresql-server
```

2. 执行以下命令在从节点创建备份目录：

```
pg_basebackup -D /var/lib/pgsql/data -h <主库IP> -p 5432 -U replica -X stream -P -R
```

其中-X表示以stream方式包含所需的WAL文件，-P表示显示进度，-h表示主库的IP信息，-p表示主库端口，-U表示主库的账户，-R表示自动写replication的配置信息；

如果显示如下图类似的信息，则表示备份成功。



```
Password:
24436/24436 kB (100%), 1/1 tablespace
```

此时，会在“/var/lib/pgsql/data”目录生成standby.signal文件，并在postgresql.auto.conf文件自动写入主库连接信息。

3. 执行以下命令修改目录所属的用户组。

```
chown -R postgres.postgres /var/lib/pgsql/data
```

4. 最后执行以下命令重启服务。

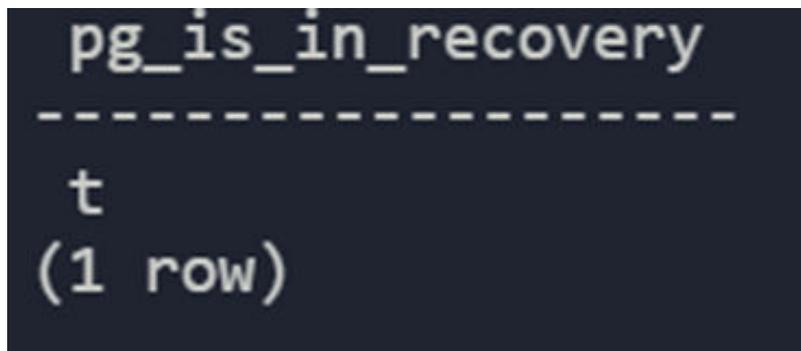
```
systemctl start postgresql
```

**步骤3** 验证配置是否成功。

1. 在从节点执行以下命令验证是否已变成从节点。

```
sudo -u postgres psql -c "SELECT pg_is_in_recovery()"
```

如果显示得到t，则表示已变为从节点，类似下图。



```
pg_is_in_recovery

t
(1 row)
```

2. 在主节点执行如下命令获取从节点信息。

```
sudo -u postgres psql -x -c "SELECT * FROM pg_stat_replication" -d postgres
```

如果得到类似如下图的信息，则表示配置成功。



# 6 Redis 一主多从部署

## 简介

Redis ( Remote Dictionary Server ) 是一个使用C语言编写的开源、基于内存、分布式、可选持久性的键值对存储数据库。Redis 是一个功能丰富的存储系统，适用于多种场景，包括缓存、会话存储、排行榜、实时分析等。它有广泛的应用，并且拥有活跃的社区支持。本教程介绍如何在HCE 2.0上安装部署Redis。

## 准备工作

- 准备两台ECS，并分配公网IP或者弹性公网IP ( EIP )；其中一台作为Redis 主节点，一台作为从节点。
- 安全组入方向都已放开6379端口。

## 操作步骤

### 步骤1 安装Redis软件及配置。

1. 在两台ECS上分别执行如下命令安装Redis。  

```
dnf install redis -y
```
2. 在两台ECS上分别执行如下命令启动服务。  

```
systemctl start redis
```

#### 说明

如果需要将Redis服务设置为开机自启动，则需要执行如下命令。

```
systemctl enable redis
```

3. 执行如下命令查看服务状态。  

```
systemctl status redis
```

如果显示**active (running)**，则表示服务启动成功。
4. 在主节点，编辑“/etc/redis.conf”文件，找到如下属性进行配置。  

```
bind 0.0.0.0 # 根据需要改成对应IP，这里改成任意IP
requirepass ***** # 配置密码
```
5. 在从节点，编辑“/etc/redis.conf”文件，找到如下属性进行配置。  

```
bind 0.0.0.0 # 根据需要改成对应IP，这里改成任意IP
requirepass ***** # 配置密码
slaveof <主节点IP> <主节点port>
masterauth <主节点密码>
```
6. 在两台ECS上分别执行如下命令重启服务。  

```
systemctl restart redis
```



# 7 Tomcat 安装

## 简介

Tomcat是一个免费的，开放源代码的Web应用服务器，是Apache软件基金会项目中的一个核心项目，是一款比较流行的web应用服务器。本教程介绍如何在HCE 2.0上安装部署Tomcat。

## 准备工作

- 准备一台ECS，并分配公网IP或者弹性公网IP（EIP）。
- 安全组入方向已放开8080端口。

## 操作步骤

### 步骤1 软件安装。

1. 执行以下命令安装Java。  

```
dnf install java-1.8.0-openjdk
```
2. 执行如下命令验证是否安装成功。  

```
java -version
```
3. 执行如下命令安装Tomcat。  

```
dnf install tomcat
```

Tomcat会被安装到“/usr/share/tomcat”目录。

### 步骤2 配置软件。

1. 编辑“/etc/profile”配置环境变量，新增如下内容。  

```
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. 执行以下命令激活上述环境变量。  

```
source /etc/profile
```
3. 清空“/usr/share/tomcat/conf/server.xml”文件，粘贴以下内容。  

```
<?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>
```

保存并退出，可根据需要，自定义上述配置。

4. 执行以下命令创建上述配置中appbase和docbase填入的目录。  
mkdir -p /data/wwwroot/default
5. 然后执行以下命令将上述目录所属用户设置为tomcat。  
chown -R tomcat.tomcat /data/wwwroot/
6. 新建“/usr/share/tomcat/bin/setenv.sh”文件，输入以下内容配置JVM内存参数。  
JAVA\_OPTS='-Djava.security.egd=file:/dev/./urandom -server -Xms256m -Xmx496m -Dfile.encoding=UTF-8'
7. 执行以下命令启动Tomcat服务。  
systemctl start tomcat
8. 执行如下命令查看Tomcat服务状态。  
systemctl status tomcat

如果显示**active (running)**则表示服务启动成功。

### 步骤3 验证Tomcat安装是否成功。

1. 执行以下命令，新增测试页面。  
echo Tomcat test > /data/wwwroot/default/index.jsp
2. 然后在浏览器输入http://<Tomcat服务公网IP>:8080进行访问，如果显示如下图则表示安装成功。



Tomcat test

----结束



以上相关配置仅用于测试，业务环境请谨慎使用。

# 8 安装 Kafka

## 简介

Kafka是一个拥有高吞吐、可持久化、可水平扩展，支持流式数据处理等多种特性的分布式消息流处理中间件，采用分布式消息发布与订阅机制，在日志收集、流式数据传输、在线/离线系统分析、实时监控等领域有广泛的应用。

本教程介绍如何在HCE 2.0上安装部署Kafka。

## 准备工作

- 准备一台ECS，并分配公网IP或者弹性公网IP（EIP）。
- 安全组入方向已放开9092端口

## 操作步骤

### 步骤1 安装kafka软件。

执行以下命令安装软件。

```
dnf install kafka
```

执行结束后，Kafka会被安装到/opt/kafka目录。

### 步骤2 软件配置。

1. 编辑“/opt/kafka/config/server.properties”文件，找到如下属性进行修改。

```
listeners=PLAINTEXT://<内网IP>:9092
advertised.listeners=PLAINTEXT://<公网IP>:9092
```
2. 新建“/lib/systemd/system/zookeeper.service”文件，并输入以下内容。

```
[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
```
- 新建 “/lib/systemd/system/kafka.service” 文件，并输入以下内容。

```
[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
```
 - 依次输入以下命令，启动Kafka和Zookeeper服务。

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

步骤3 验证。

- 使用以下命令创建Topic。

```
/opt/kafka/bin/kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test
```
- 使用以下命令查看刚才创建的Topic。

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

如果部署正常，则会输出**test**。

----结束



以上相关配置仅用于测试，业务环境请谨慎使用。

9 Gnome 桌面安装

简介

Gnome是一种桌面环境，它的目标系统是Linux，但是大部分的BSD系统也支持Gnome。本教程介绍如何在HCE 2.0上安装Gnome图形化界面。

准备工作

准备一台ECS主机。

操作步骤

步骤1 安装Gnome桌面

执行如下命令安装Gnome桌面。

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

步骤2 设置为默认图形化桌面启动。

1. 执行以下命令将图形化界面设置为默认启动。

```
systemctl set-default graphical.target
```

2. 之后执行如下命令重启。

```
reboot
```

----结束

10 Apache 安装

简介

Apache HTTP Server（简称Apache）是Apache软件基金会的一个开源网页服务器，可以在大多数操作系统中运行，由于其能跨平台、安全性高而被广泛使用，是最流行的Web服务器软件之一，其拥有的特性包括支持FastCGI、支持SSL、集成Perl处理模块等。本教程介绍如何在HCE 2.0中部署Apache。

准备工作

- 准备一台ECS，并分配公网IP或者弹性公网IP（EIP）。
- 安全组入方向已放开80端口。

操作步骤

步骤1 安装Apache软件。

1. 执行如下命令安装Apache服务。

```
dnf install httpd httpd-devel
```

执行完后，Apache服务配置文件路径为：“/etc/httpd/conf/httpd.conf”

2. 执行以下命令查看Apache版本号。

```
httpd -v
```

输出内容类似如下。

```
[root@localhost system]# httpd -v
Server version: Apache/2.4.51 (Unix)
Server built: Feb 9 2022 09:00:41
```

3. 执行如下命令启动Apache服务。

```
systemctl start httpd
```

如果输出显示**active (running)**，则表示启动成功。

说明

如果要Apache服务设置为开机自启动，则需要依次执行以下命令。

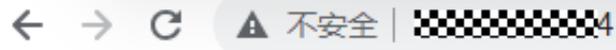
```
systemctl enable httpd
systemctl daemon-reload
```

步骤2 验证。

1. 执行以下命令创建测试页面。

```
echo test > /var/www/html/index.html
```

2. 在浏览器中输入http://<公网IP>, 如果显示如下图, 则表示部署成功。



test

----结束

11 部署 Django 服务

简介

Django是一个开源的Web应用框架，使用Python语言编写，主要用于搭建Web项目。本教程介绍如何在HCE 2.0中使用Nginx+uWSGI部署Django项目。

准备工作

- 准备一台ECS，并分配公网IP或者弹性公网IP（EIP）。
- 安全组入方向已放开80、8001、8002端口。

操作步骤

步骤1 安装Nginx服务。

1. 执行以下命令安装Nginx。

```
dnf install nginx
```
2. 执行以下命令启动Nginx服务。

```
systemctl start nginx
```
3. 执行以下命令查看Nginx服务状态。

```
systemctl status nginx
```

如果显示**active (running)**，则表示启动成功。

步骤2 安装uWSGI服务。

1. 执行以下命令安装相关依赖。

```
dnf install python3-devel gcc
```
2. 执行以下命令安装uWSGI。

```
pip install uwsgi
```
3. 执行以下命令查看uWSGI版本号。

```
uwsgi --version
```
4. 编辑hello.py文件，并输入以下内容。

```
def application(env, reply):  
    reply('200 ok', [('Content-Type', 'text/html')])  
    return [b"Hello!"]
```
5. 执行以下命令启动uWSGI服务。

```
uwsgi --http :8001 --wsgi-file hello.py
```
6. 在浏览器中输入网址：`http://<公网IP>:8001`访问上述服务，效果如下图。



Hello!

步骤3 安装Django环境

1. 执行以下命令安装Django。

```
pip install Django
```
2. 执行以下命令初始化项目。

```
python -m django startproject django_project
```
3. 进入项目目录，编辑配置文件django_project/settings.py，找到ALLOWED_HOSTS属性修改成如下值。

```
ALLOWED_HOSTS = ['*']
```
4. 执行如下命令启动django服务。

```
python manage.py runserver 0.0.0.0:8002
```
5. 浏览器中输入网址http://<公网IP>:8002访问上述服务，效果如下。

django

[View release notes](#) for Django 4.2



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

步骤4 配置环境。

1. 编辑django_project/settings.py文件。
 - a. 在文件开头输入以下语句引入os库。

```
import os
```
 - b. 然后在文件末尾添加以下参数。

```
STATIC_ROOT = os.path.join(BASE_DIR, "static/")
```
 - c. 再执行以下命令收集所有静态文件。

```
sudo python manage.py collectstatic
```

执行以后输出类似如下。

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

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

此时项目目录就会多一个static文件夹，项目结构如下。

```
[root@hce2 django_project]# ls
```

```
db.sqlite3  django_project  manage.py  static
```

2. 编辑/etc/nginx/nginx.conf文件，配置nginx。

a. 找到http属性，新增以下内容。

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

b. 找到http内的server属性，将其修改成以下属性。

```
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;  
    }  
}
```

最终效果如下。

```
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/nginx_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. 在项目目录新建uwsgi_config.ini文件，并输入以下内容。

```
[uwsgi]
socket = 127.0.0.1:8001 #此处的8001端口需要和nginx配置文件中定义的uwsgi_pass端口一致
chdir = /root/django_project/ #指定项目目录，本示例中为/root/django_project/，根据项目修改
wsgi-file = django_project/wsgi.py #指定Django's wsgi file文件，根据项目修改
processes = 4 #最大工作进程
threads = 2 #每个工作进程processes启动后开启的线程个数
vacuum = true #环境退出时自动清理
buffer-size = 65536 #设置用于uwsgi包解析的内部缓存区大小为64k，默认是4k
```

**注意**

实际环境中，请删掉配置文件中的日志。

步骤5 验证。

1. 执行以下命令重启Nginx。
systemctl restart nginx
2. 在项目目录执行以下命令，启动uWSGI。
uwsgi --ini uwsgi_config.ini
3. 在浏览器的地址栏输入http://<公网IP>，访问相关Django页面，效果如下。

django

View [release notes](#) for Django 4.2



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

----结束



以上相关配置仅用于测试演示，业务环境请谨慎使用。

12 在线安装 lnmp

简介

lnmp软件是一个用Linux Shell编写的可以为各类linux服务器或独立主机安装LNMP(Nginx/MySQL/PHP)、LNMPA(Nginx/MySQL/PHP/Apache)、LAMP(Apache/MySQL/PHP)生产环境的Shell程序。

本教程介绍如何在HCE 2.0上安装lnmp软件。

准备工作

准备一台ECS实例，并分配公网IP或者弹性公网IP（EIP）。

操作步骤

步骤1 执行以下命令，下载lnmp2.0，并解压进入目录。

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

步骤2 执行以下命令修改main.sh文件。

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

在Get_Dist_Name函数中添加一个elif判断。

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

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

步骤3 执行以下命令修改init.sh文件。

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

在Install_Freetype函数的if判断中添加如下代码。

```
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
```

步骤4 执行以下命令开始安装。

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

当出现如下回显，则安装成功。

```
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.
```

---结束

13 qperf 编译支持 IPV6

简介

目前HCE 2.0中的qperf版本为0.4.9，不支持IPV6，如果要使用IPV6，需要从社区获取0.4.11及以上版本进行升级。

本教程介绍如何在HCE 2.0上下载qperf 0.4.11源码并编译。

准备工作

准备一台ECS实例，并分配公网IP或者弹性公网IP（EIP）。

操作步骤

步骤1 执行以下命令下载qperf 0.4.11源码文件。

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

步骤2 执行以下命令解压下载文件。

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

解压后得到目录**qperf-0.4.11**。

步骤3 执行以下命令安装依赖。

```
dnf install gcc make automake
```

步骤4 进入qperf-0.4.11目录，依次执行以下命令进行编译。

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

步骤5 验证编译结果。

编译得到的二进制在**qperf-0.4.11/src**目录下，进入该目录后执行**./qperf -version**，得到类似如下回显，表示编译成功：

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

---结束