Linux 环境 EIP7 产品多服务部署手册

广州宏天软件股份有限公司

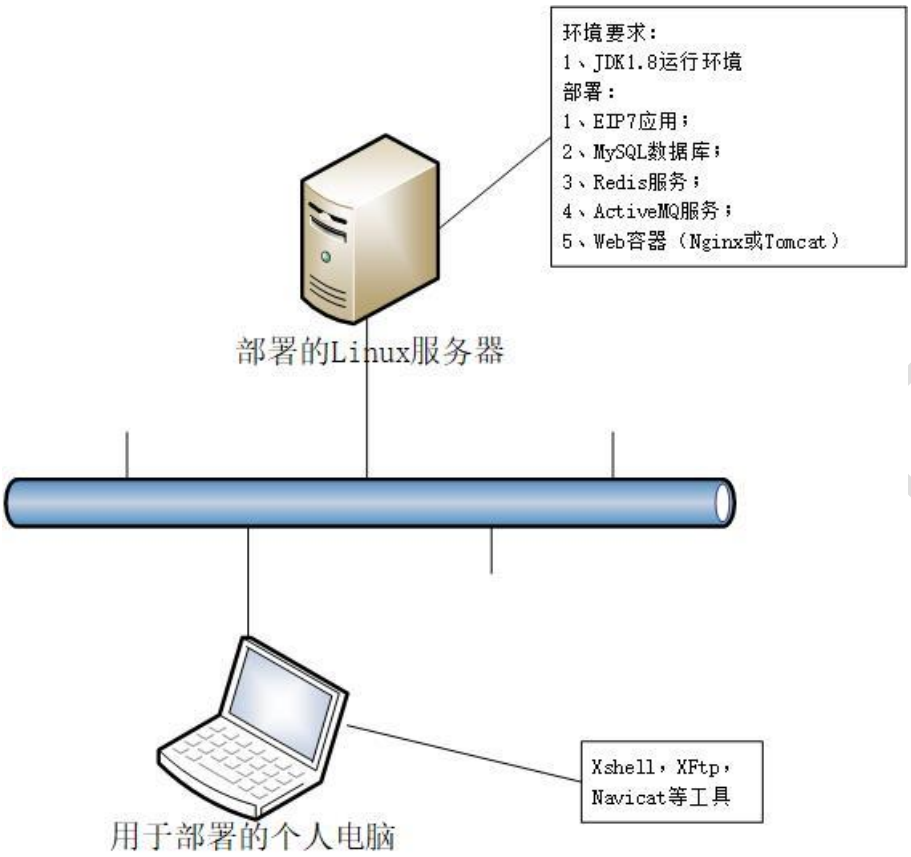
2020 年 4 月 20 日

版本管理

序号	版本	作者	日期	说明
1	V1.0	何一帆	2019 年 10 月 17 日	初始版本
2	V1.1	凌朋金	2020 年 4 月 20 日	EIP7.2 版本部署

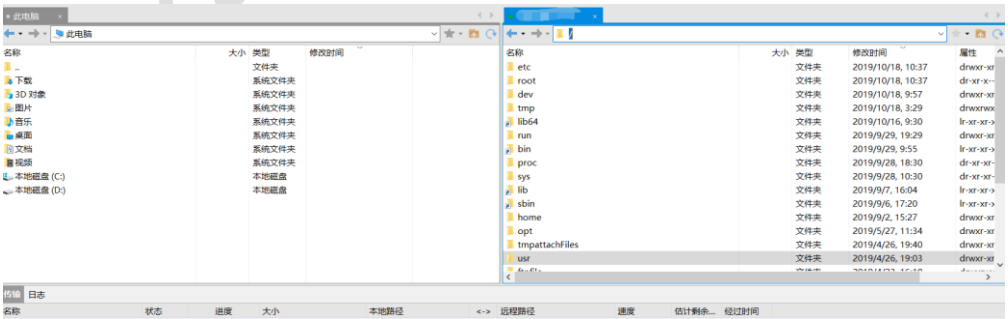
目录

1 总体部署图

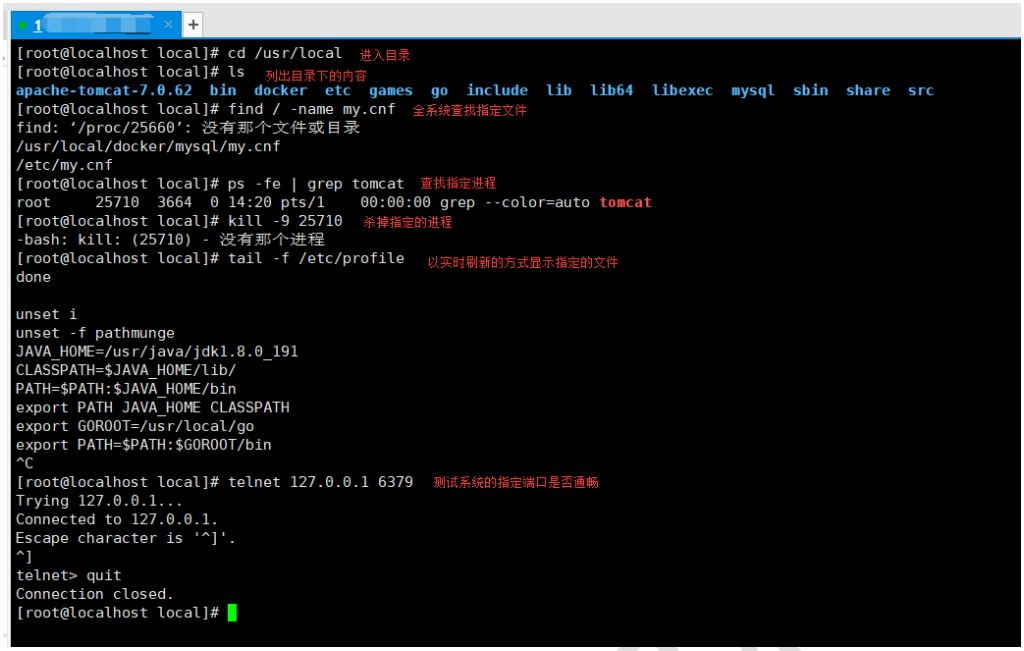


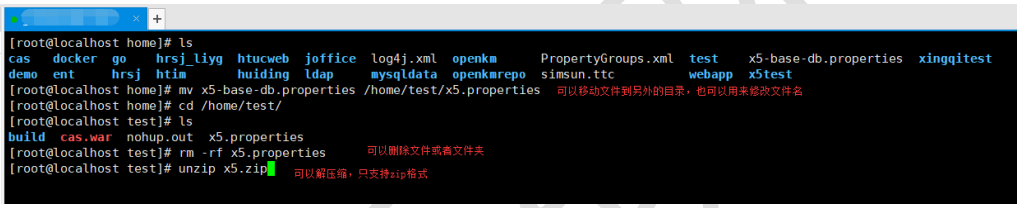
2 常用 linux 命令

1. 使用 xshell 等工具远程连接到 linux 服务器，通过 xftp 可以上传下载文件，在 linux 中没有像 windows 一样分为 C 盘、D 盘等不同的盘符，所有目录都通过/开头来访问，如下图所示



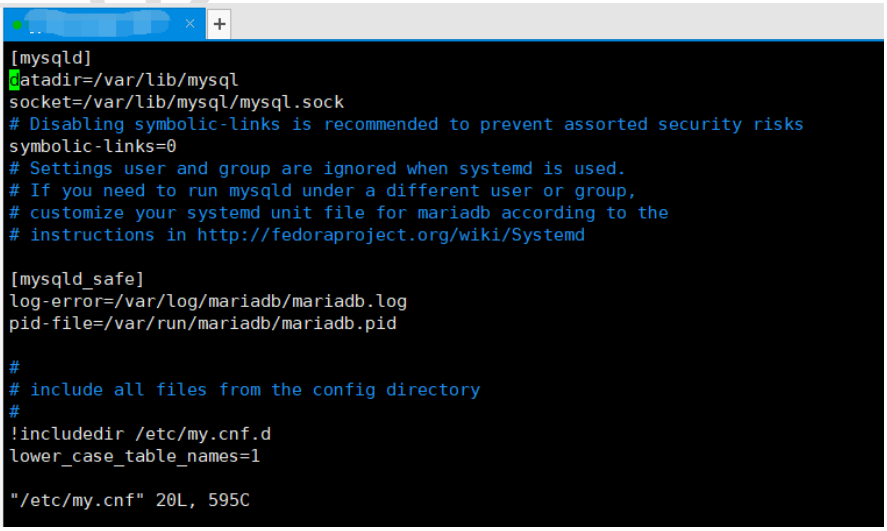
2. 常用的命令





3. yum 命令可以用来安装系统中不存在的命令，比如 vim 命令不存在时，可以通过 yum install vim 来安装。注意在安装的过程中系统可能会跳出一些询问，输入 y 回车就可以继续安装了

4. vim 命令可以对文本文件进行修改，例如 vim /etc/my.cnf 可以打开 mysql 的配置文件，刚打开文本时为浏览模式，不能添加、删除、修改任何内容。文本内容过多时，快捷键 shift+g 可以直接定位到文本的最后一行；内容过多也可以通过"/搜索内容"来快速查找，有多个匹配结果时 n 可以查看下一个。按 insert 进入输入模式，完成编辑以后按 esc 按钮退出编辑模式，输入:q!放弃保存，输入:wq 则保存修改。



```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid

#
# include all files from the config directory
#
!includedir /etc/my.cnf.d
lower_case_table_names=1

/mysql
```

5. 防火墙配置

```
[root@localhost hotent]# firewall-cmd --zone=public --list-ports   查看防火墙白名单
3306/tcp
[root@localhost hotent]# firewall-cmd --permanent --zone=public --add-port=8088/tcp
success                                                            添加8088端口到白名单
[root@localhost hotent]# firewall-cmd --zone=public --list-ports
3306/tcp
[root@localhost hotent]# firewall-cmd --reload   需要重启防火墙，新添加的端口才会生效
success
[root@localhost hotent]# firewall-cmd --zone=public --list-ports
3306/tcp 8088/tcp
[root@localhost hotent]#
```

3 运行环境和中间件安装

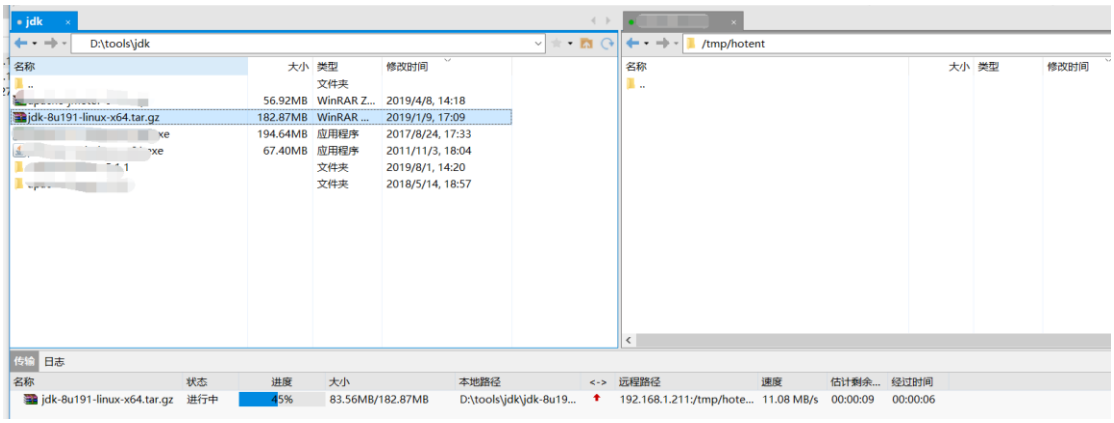
3.1 安装 JDK1.8

使用 xshell 工具连接 linux 服务器，在命令行中输入 java -version，检查 JDK 是否安装以及版本是否为 1.8

```
[root@localhost ~]# java -version
java version "1.8.0_191"
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)
[root@localhost ~]#
```

如果 JDK 未安装或者版本不正确，则重新安装 JDK。

1. 将 jdk 的安装包拷贝到 linux 服务器（通过 xftp 上传上去，在/tmp 目录下创建一个 hotent 目录，用于存放上传的文件）



2. 将该文件拷贝到/usr/local/java 目录下，并解压

```
[root@localhost local]# mkdir /usr/local/java
[root@localhost local]# cd /tmp/hotent
[root@localhost hotent]# ls
jdk-8u191-linux-x64.tar.gz
[root@localhost hotent]# mv jdk-8u191-linux-x64.tar.gz /usr/local/java/jdk1.8.tar.gz
[root@localhost hotent]# cd /usr/local/java/
[root@localhost java]# ls
jdk1.8.tar.gz
[root@localhost java]# tar -zxvf jdk1.8.tar.gz
```

3. 查看解压出来的目录名

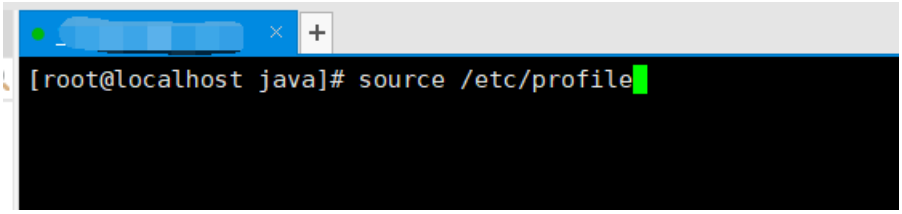
```
[root@localhost java]# ls
jdk1.8.0_191  jdk1.8.tar.gz
[root@localhost java]#
```

4. 配置 jdk 到环境变量，通过 vim 命令编辑/etc/profile，进入编辑模式后，shift+g 定位到文件的最后一行，点击 insert 命令进入编辑模式，编辑完以后按 ecs 按钮，输入:wq+回车则保存文件，输入:q!则放弃保存。

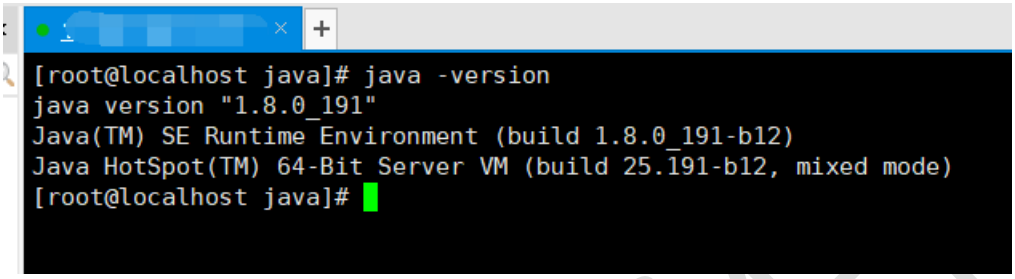
JACA_HOME 为 jdk 路径

```
for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
export JAVA_HOME=/usr/local/java/jdk1.8.0_191
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

5. 完成配置后输入 source /etc/profile 刷新环境变量

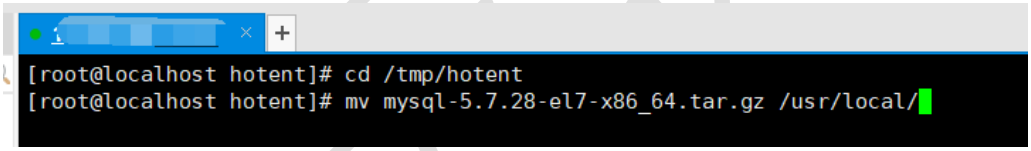


6. 现在输入 java -version 确认 jdk 环境变量是否配置正确，版本是否一致

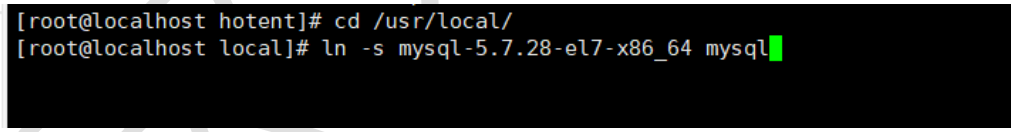


3.2 安装 MySQL

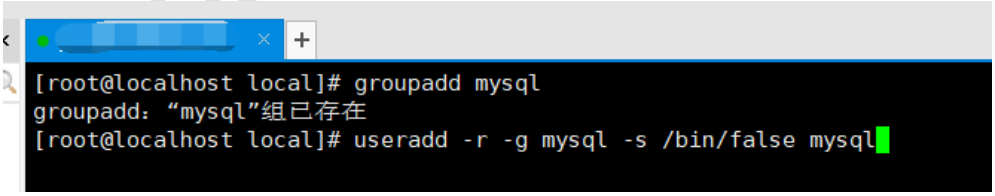
1. 将 mysql 的安装包上传到/tmp/hotent 目录下，然后解压至/usr/local/目录，通过命令 tar -zxvf mysql-5.7.28-el7-x86_64.tar.gz 可以完成解压操作



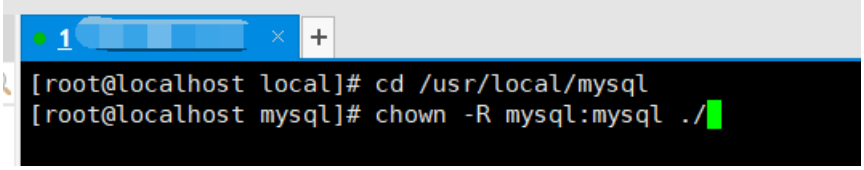
2. 进入/usr/local 目录，为 mysql 安装目录创建软链接



3. 为 linux 系统添加 mysql 用户组和 mysql 用户(-s /bin/false 参数指定 mysql 用户仅拥有所有权，而没有登录权限)



4. 进入 mysql 安装目录，并设置目录的拥有者为新建的 mysql 用户

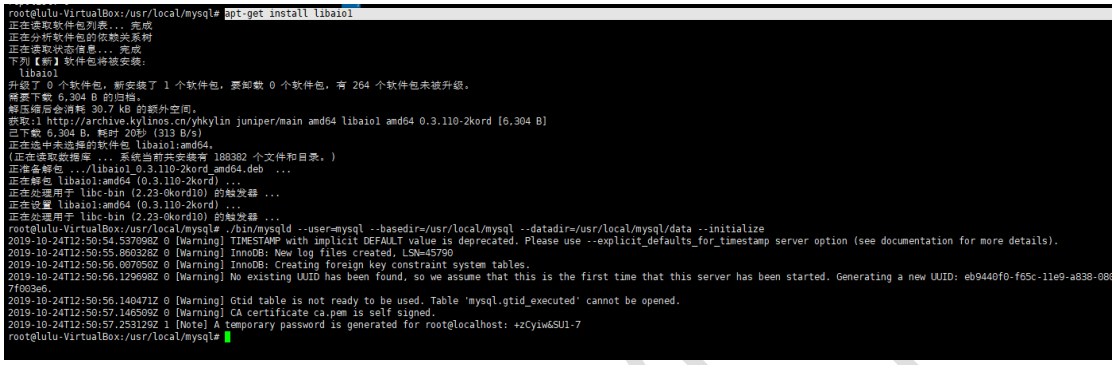


5. 安装 mysql

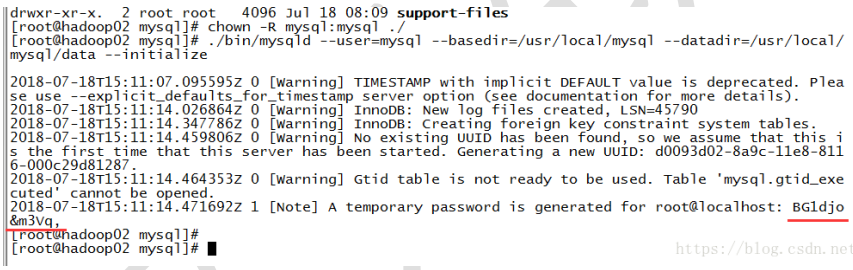
```
./bin/mysqld --user=root --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data --initialize
```

如果执行这一步时提示找不到路径，则表示少了一个包，可以使用以下命令：

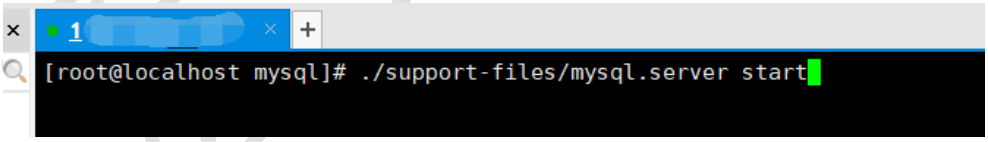
apt-get install libaio1



6. 注意观察控制台的输入内容，如果看到类似如下内容，标识安装成功，将随机生成的初始密码保存起来



7. 启动 mysql 服务



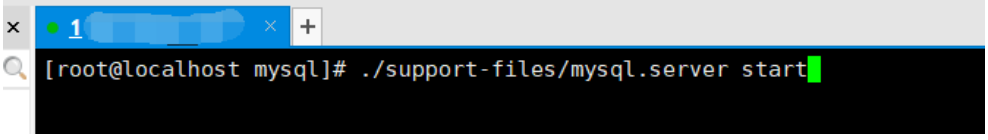
8. 如果出现如下错误，则说明 mysql 配置文件/etc/my.cnf 中的路径不对，修改内容如下，datadir 和 socket 都修改成 mysql 的安装目录下，增加[client]板块，用于命令行连接 mysql 数据库

```
[root@hadoop02 mysql]#
[root@hadoop02 mysql]# ./support-files/mysql.server start
Starting MySQL.2018-07-18T15:15:20.305540Z mysqld_safe Directory '/var/lib/mysql' for UNIX socket file don't exists.
 ERROR! The server quit without updating PID file (/var/lib/mysql/hadoop02.pid).
[root@hadoop02 mysql]#
```



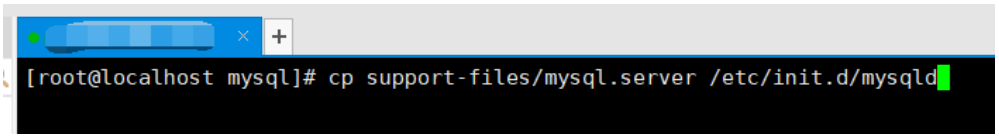
```
[mysqld]
port=3306
datadir=/usr/local/mysql/data
socket=/usr/local/mysql/mysql.sock
user=mysql
max_connections=151
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0

# 设置忽略大小写
lower_case_table_names = 1

# 指定编码
character-set-server=utf8

collation-server=utf8_general_ci

# 设置 sql 模式，解决日期默认值等问题
sql_mode = "STRICT_TRANS_TABLES,NO_ENGINE_SUBSTITUTION,NO_AUTO_CREATE_USER"

# 开启 ip 绑定
bind-address = 0.0.0.0

[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

#指定客户端连接 mysql 时的 socket 通信文件路径
[client]
socket=/usr/local/mysql/mysql.sock

default-character-set=utf8
```

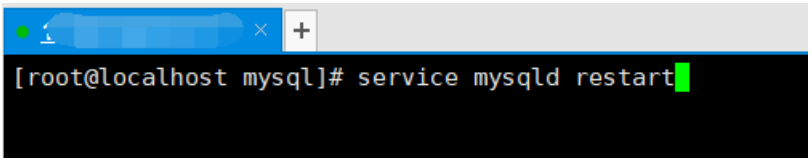
9. 重新开启 mysql 服务，如下图所示则开启成功




```
~
"/etc/my.cnf" 29L, 590C written
[root@hadoop02 mysql]# ./support-files/mysql.server start
Starting MySQL. SUCCESS!
[root@hadoop02 mysql]#
```

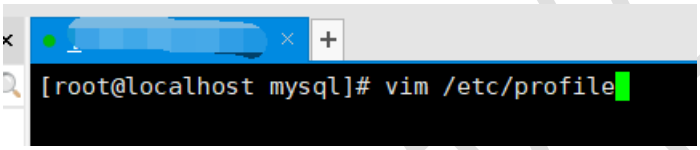

10. 将 mysql 进程放入系统进程中，并重新启动 mysql 服务

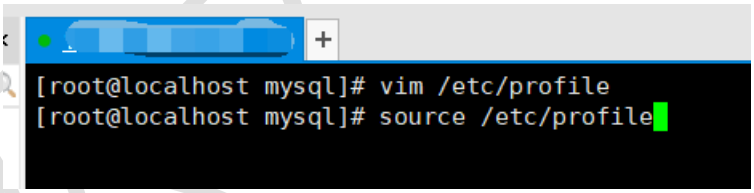




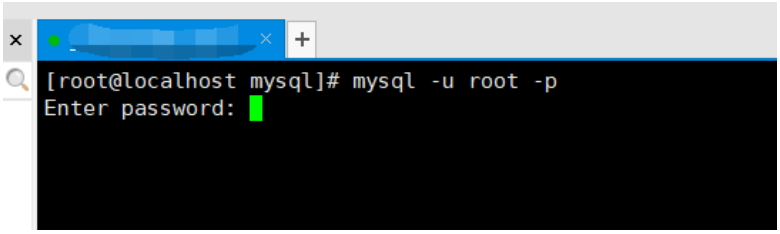

```
Starting MySQL. SUCCESS!
[root@hadoop02 mysql]# ps -ef|grep mysql
root      2875     1  0 08:20 pts/0    00:00:00 /bin/sh /usr/local/mysql/bin/mysqld_safe --datadir=/usr/local/mysql/data --pid-fi
mysql     3114  2875  0 08:20 pts/0    00:00:00 /usr/local/mysql/bin/mysqld --basedir=/usr/local/mysql --datadir=/usr/local/mysql
l --log-error=/var/log/mysqld.log --pid-file=/usr/local/mysql/data/hadoop02.pid --socket=/usr/local/mysql/mysql.sock --port=3306
root      3186  2481  0 08:22 pts/0    00:00:00 grep mysql
[root@hadoop02 mysql]# cp support-files/mysql.server /etc/init.d/mysqld
[root@hadoop02 mysql]# service mysqld restart
Shutting down MySQL.. SUCCESS!
Starting MySQL. SUCCESS!
[root@hadoop02 mysql]#
```


11. 配置 mysql 的环境变量，并重新编译环境变量





12. 使用 mysql 命令，账号 root，密码是刚才生成的随机密码来登录 mysql（在 Enter password 后面输入内容时，不会显示输入的内容、而且光标不会移动，不用担心，密码其实已经输入了，只是为了保密不做显示而已，输完回车就可以了）



13. 成功登录后显示界面如下，初次登录会要求你修改 root 的密码，使用如下命令来修改密码

```
[root@hadoop02 mysql]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.22

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

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>
```



```
alter user 'root'@'localhost' identified by 'root';
```

14. 设置允许远程连接数据库

```
use mysql

update user set user.Host='%' where user.User='root';

flush privileges;
```

15. 输入 quit;退出 mysql 命令行模式，通过 telnet 命令测试 mysql 端口是否开通

```
[root@localhost mysql]# telnet 127.0.0.1 3306
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
R
5.5.64-MariaDB1!+:~2x,k _OO~MCH"n&lqmysql_native_password
```

16. 在用于部署的个人电脑上通过 Navicat 连接这个数据库，并完成数据库的初始化操作

3.3 安装 Redis

17. Redis 的安装需要在 linux 上编译，所以要先检测是否有 gcc 编译环境，如果没有编译环境，协调客户方安装 gcc 编译环境
18. Gcc 安装说明：https://www.cnblogs.com/dail1990/p/10142728.html

```
[root@localhost test]# gcc --version
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36)
Copyright © 2015 Free Software Foundation, Inc.
本程序是自由软件；请参看源代码的版权声明。本软件没有任何担保；
包括没有适销性和某一专用目的下的适用性担保。
[root@localhost test]#
```

19. 上传 redis 安装包到/tmp/hotent 目录，并解压到/usr/local/目录

```
[root@localhost hotent]# ls
redis-5.0.4.tar.gz
[root@localhost hotent]# tar -zxvf redis-5.0.4.tar.gz -C /usr/local
redis-5.0.4/
redis-5.0.4/.gitignore
redis-5.0.4/00-RELEASENOTES
redis-5.0.4/BUGS
redis-5.0.4/CONTRIBUTING
redis-5.0.4/COPYING
redis-5.0.4/INSTALL
redis-5.0.4/MANIFESTO
redis-5.0.4/Makefile
redis-5.0.4/README.md
redis-5.0.4/deps/
redis-5.0.4/deps/Makefile
redis-5.0.4/deps/README.md
redis-5.0.4/deps/hiredis/
redis-5.0.4/deps/hiredis/.gitignore
redis-5.0.4/deps/hiredis/.travis.yml
redis-5.0.4/deps/hiredis/CHANGELOG.md
redis-5.0.4/deps/hiredis/COPYING
redis-5.0.4/deps/hiredis/Makefile
```

20. 进入/usr/local/redis-5.0.4 目录并执行 make 命令

```
[root@localhost redis-5.0.4]# make MALLOC=libc
cd src && make all
make[1]: Entering directory `/usr/local/redis-5.0.4/src'
    CC Makefile.dep
make[1]: Leaving directory `/usr/local/redis-5.0.4/src'
make[1]: Entering directory `/usr/local/redis-5.0.4/src'
rm -rf redis-server redis-sentinel redis-cli redis-benchmark redis-check-rdb redis-check-aof *.o *.gcda *.gcno *.gcov redis.info
ep dict-benchmark
(cd ../deps && make distclean)
```

```
[root@localhost redis-5.0.4]# cd src && make install
    CC Makefile.dep

Hint: It's a good idea to run 'make test' ;)

    INSTALL install
    INSTALL install
    INSTALL install
    INSTALL install
    INSTALL install
[root@localhost src]#
```

21. 编译安装成功后，在 src 目录下执行./redis-server 启动 redis 服务

```
[root@CDH-143 src]# ./redis-server
24496:C 11 Mar 13:49:56.298 # oooOoooOoooOo Redis is starting oooOoooOoooOo
24496:C 11 Mar 13:49:56.298 # Redis version=4.0.11, bits=64, commit=00000000, modified=0, pid=24496, just started
24496:C 11 Mar 13:49:56.298 # Warning: no config file specified, using the default config. In order to specify a config file use ./redis-server
/path/to/redis.conf
                _._
           _.-``__ ''-._
      _.-``    `.  `_.  ''-._           Redis 4.0.11 (00000000/0) 64 bit
  .-`` .-```.  ```\/    _.,_ ''-._
 (    '      ,       .-`  | `,    )     Running in standalone mode
 |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
 |    `-._   `._    /     _.-'    |     PID: 18685
  `-._    `-._  `-./  _.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |           http://redis.io
  `-._    `-._`-.__.-'_.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |
  `-._    `-._`-.__.-'_.-'    _.-'
      `-._    `-.__.-'    _.-'
          `-._        _.-'
              `-.__.-'

24496:M 11 Mar 13:49:56.302 # Server initialized 24496:M 11 Mar 13:49:56.303 # WARNING overcommit_memory is set to 0! Background save may fail under low
memory condition. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for
this to take effect. 24496:M 11 Mar 13:49:56.303 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency and
memory usage issues with Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your
/etc/rc.local in order to retain the setting after a reboot. Redis must be restarted after THP is disabled. 24496:M 11 Mar 13:49:56.303 * Ready to accept
connections
```

22. 复制 redis 的配置文件

```
[root@CDH-143 src]# mkdir /etc/redis
[root@CDH-143 src]# cd ..
[root@CDH-143 redis-4.0.11]# ls
00-RELEASENOTES CONTRIBUTING deps Makefile README.md runtest runtest-sentinel src utils
BUGS COPYING INSTALL MANIFESTO redis.conf runtest-cluster sentinel.conf tests
[root@CDH-143 redis-4.0.11]# cp redis.conf /etc/redis/6379.conf
```

23. 修改 redis 为后台启动

```
[root@CDH-143 redis-4.0.11]# vi /etc/redis/6379.conf

# A reasonable value for this option is 300 seconds, which is the new
# Redis default starting with Redis 3.2.1.
tcp-keepalive 300

################################ GENERAL #####################################

# By default Redis does not run as a daemon. Use 'yes' if you need it.
# Note that Redis will write a pid file in /var/run/redis.pid when daemonized.
daemonize yes

# If you run Redis from upstart or systemd, Redis can interact with your
# supervision tree. Options:
#   supervised no      - no supervision interaction
#   supervised upstart - signal upstart by putting Redis into SIGSTOP mode
"/etc/redis/6379.conf" 1317L, 58767C written
[root@CDH-143 redis-4.0.11]#
```

24. 将启动文件复制到 init.d 中，并通过 vim /etc/init.d/redisd 命令修改启动文件的内容，修改完:wq 保存

```
[root@CDH-143 redis-4.0.11]# cp utils/redis_init_script /etc/init.d/redisd
```

```
1 192.168.1.211
#!/bin/sh
#
# chkconfig: 2345 10 90
# description: Start and Stop redisd
# Simple Redis init.d script conceived to work on Linux systems
# as it does use of the /proc filesystem.

### BEGIN INIT INFO
# Provides:     redis_6379
# Default-Start:        2 3 4 5
# Default-Stop:         0 1 6
# Short-Description:    Redis data structure server
# Description:          Redis data structure server. See https://redis.io
### END INIT INFO

REDISPORT=6379
EXEC=/usr/local/redis-5.0.4/src/redis-server
CLIEXEC=/usr/local/redis-5.0.4/src/redis-cli
PIDFILE=/usr/local/redis-5.0.4/redis_${REDISPORT}.pid
CONF="/etc/redis/${REDISPORT}.conf"
```

25. 添加脚本的执行权限，并设置为系统服务

```
[root@CDH-143 redis-4.0.11]# chmod +x /etc/init.d/redisd
```

```
[root@CDH-143 redis-4.0.11]# chkconfig --add redisd
[root@CDH-143 redis-4.0.11]# chkconfig --list redisd
```

26. 通过服务命令测试 redis 的启动、停止、查看状态

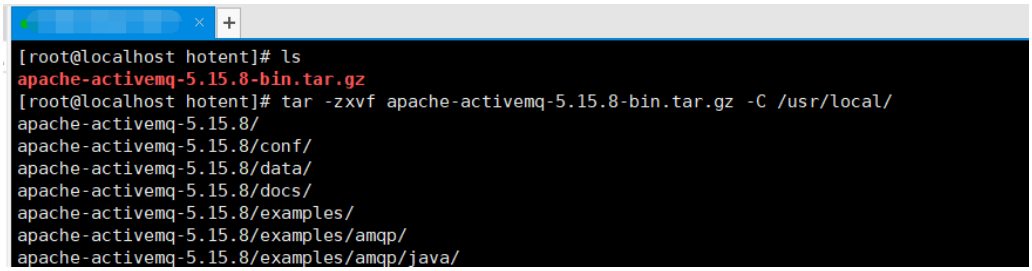
```
[root@CDH-143 redis-4.0.11]# service redisd start
Starting Redis server...
9084:C 11 Mar 15:23:36.469 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
9084:C 11 Mar 15:23:36.469 # Redis version=4.0.11, bits=64, commit=00000000, modified=0, pid=9084, just started
9084:C 11 Mar 15:23:36.469 # Configuration loaded
[root@CDH-143 redis-4.0.11]#
[root@CDH-143 redis-4.0.11]# ps -aux | grep redis
root 3575 0.0 0.0 38260 2140 ? Ssl 15:09 0:00 /opt/redis/redis-4.0.11/src/redis-server 127.0.0.1:6379
root 8870 0.0 0.0 112660 956 pts/2 S+ 15:22 0:00 grep --color=auto redis
```

27. 通过 telnet 命令查看 6379 端口上的 redis 服务是否正常

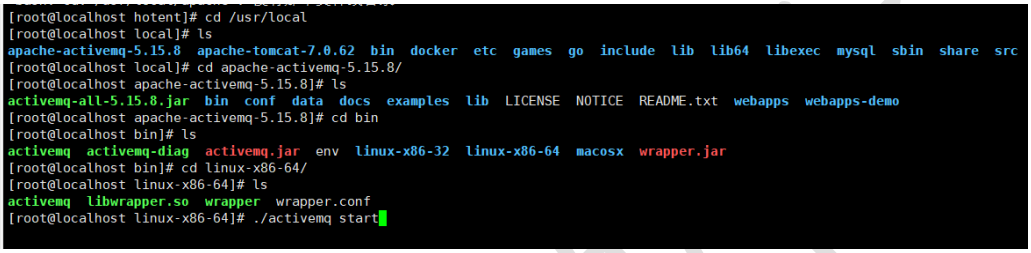
3.4 安装 ActiveMQ

1. activemq 的安装包上传到/tmp/hotent 目录，并解压缩到/usr/local/activemq

目录



2. 进入解压后的目录/usr/local/apache-activemq-5.15.8/bin/linux-x86-64，执行启动服务的命令

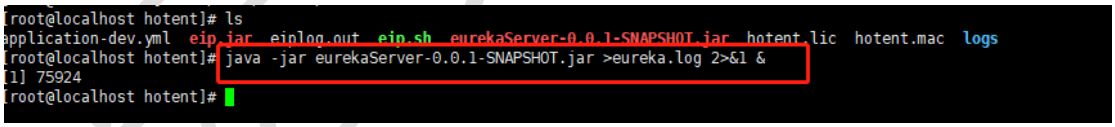


3. 通过 telnet 命令测试 61616 和 61614 端口是否正常

3.5 部署 EurekaServer

1. 上传 eurekaServer 包至目录

2. 使用命令启动 eurekaServer

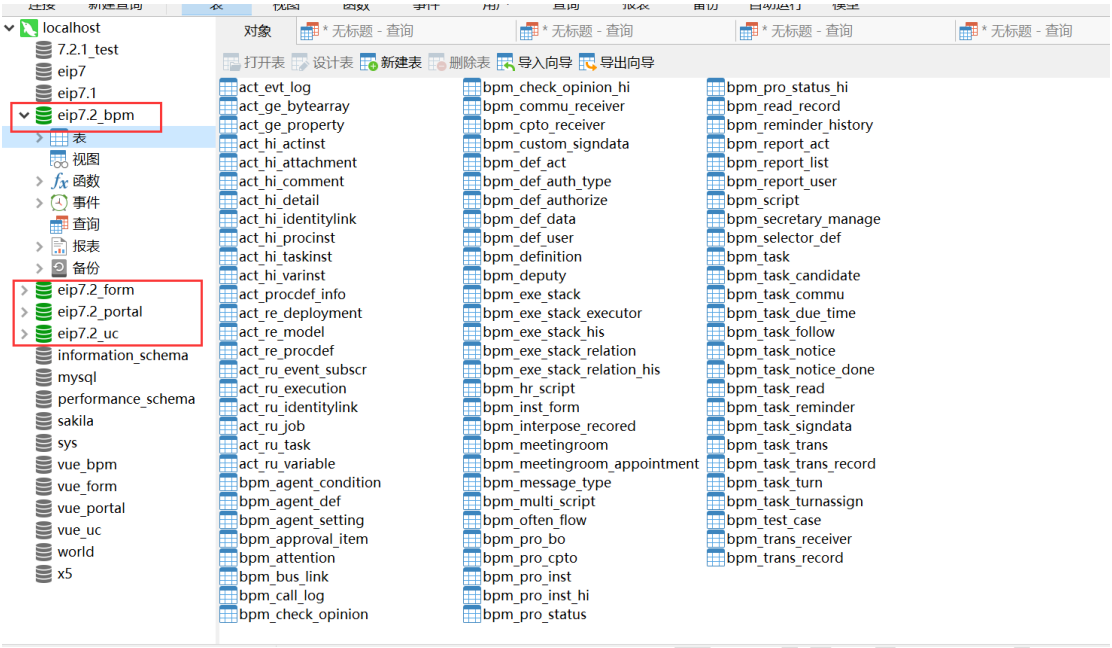


3. 通过 telnet 命令查看 8761 的 eureakServer 服务是否正常

4 应用部署

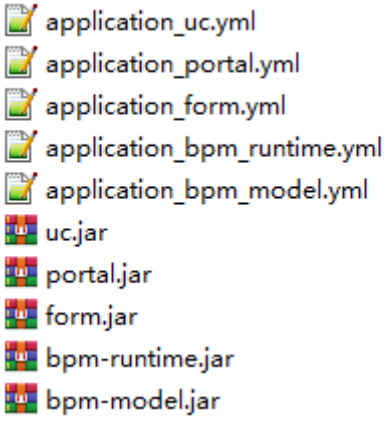
4.1 Java 应用

1. 使用数据库可视化工具（如 navicat）新建数据库，选择字符集(utf-8)和排序规则,数据库名称自定义，，为更好辨认对应微服务的数据库可创建如下图所示：



2. 执行数据库脚本，找到我们提供的 sql 文件，对应 bpm/form/portal/uc 数据库分别执行，执行步骤请参考《初始化 sql 执行说明》

3. 将要部署的 5 个服务包及对应的配置文件和启动脚本上传至服务器。修改配置文件中数据库参数（连接各自对应的数据库，bpm-runtime 及 bpm-model 服务连接同一个 bpm 数据库）、eureka 参数、redis 参数、jms 参数、mqtt 参数以及 base.domain 地址（指向部署服务器的 ip）.



```yaml
server:
  port: 8088
  address: 0.0.0.0
  sessionTimeout: 30
  contextPath: /
  undertow:
    io-threads: 2
    worker-threads: 30
  session:
    timeout: 30
  compression:
    enabled: true
    mime-types: 'text/html,text/xml,text/plain,text/css,text/javascript,application/javascript,application/json'
    min-response-size: 1024
```

1 服务端口配置

```yaml
# imServer setting
mqtt:
  host: 127.0.0.1
  port: 61614

# ueditor setting  端口五合一打包时记得修改为对应的五合一端口
base.domain: http://127.0.0.1:8084
```

```yaml
    datacenterId: 1

eureka:
  client:
    healthcheck:
      enabled:  true
    service-url:
      defaultZone: http://127.0.0.1:8761/eureka/
  instance:
    lease-expiration-duration-in-seconds: 30
    lease-renewal-interval-in-seconds: 10
    # 注册到eureka时使用ip地址
    prefer-ip-address: true
    instance-id: ${spring.cloud.client.ip-address}:${server.port}

  # feign配置
feign:
  hystrix:
    enabled: true
  httpclient:
    enabled: true
  client:
    config:
      default:
        connectTimeout: 50000
        readTimeout: 50000
        loggerLevel: full

  # 断路器配置
hystrix:
  metrics:
    enabled: true
  command:
    default:
      execution:
        isolation:
```

1 微服务启动需使用eureka，修改eureka信息

如果需要对数据库密码加密，可以取消对 public-key 的注释，并在宏天公开平台生成加密后的密钥和 public-key

密码和公钥获取方式：进入宏天公开开放的平台，进入以下菜单

设计中心/元件管理/数据源/添加

4. 配置执行权限给 5 个启动脚本，并通过命令启动应用服务，应用的日志输出到各自的日志文件，可以通过 tail -f 命令来查看

（以 uc 服务为例）

```
[root@localhost hotent]# chmod +x uc.sh
[root@localhost hotent]# ./uc.sh start
uc.jar start success
```

出现授权错误（见下图）请将目录下的 hotent.mac 文件发给宏天人员用于生成授权文件

```
[root@localhost hotent]# tail -f eiplog.out

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.0.1.RELEASE)

[license]: Machine verifying...
open hotent.lic: no such file or directory
[license]: Write successfully: hotent.mac, the path was: /home/hotent
```

将授权文件 hotent.lic 上传至 jar 包目录下，重新启动。

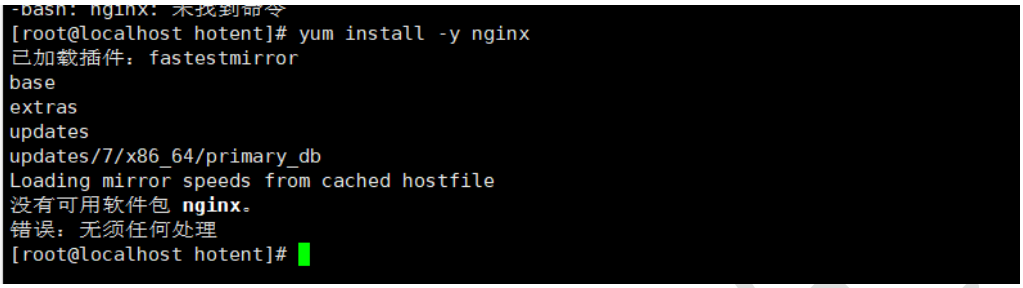
```
[root@localhost hotent]# tail -f eiplog.out

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::        (v2.0.1.RELEASE)

[license]: Machine verifying...
[license]: Machine validate success
**************************
*********应用已启动*********
**************************
```

4.2 Web 应用

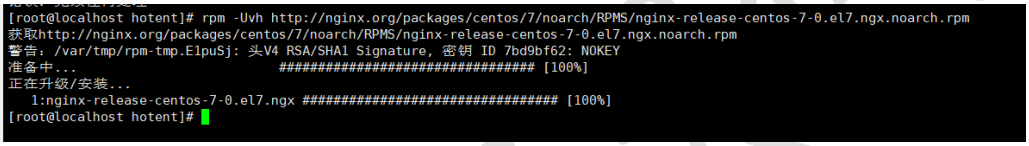
安装 Nginx，通过 Nginx 部署 web 应用。

4.2.1 Nginx 的安装和配置

1. 通过 yum install -y nginx 安装，如果没有可用的 nginx 包，如下图所示，可以通过添加源以后，再执行安装命令。

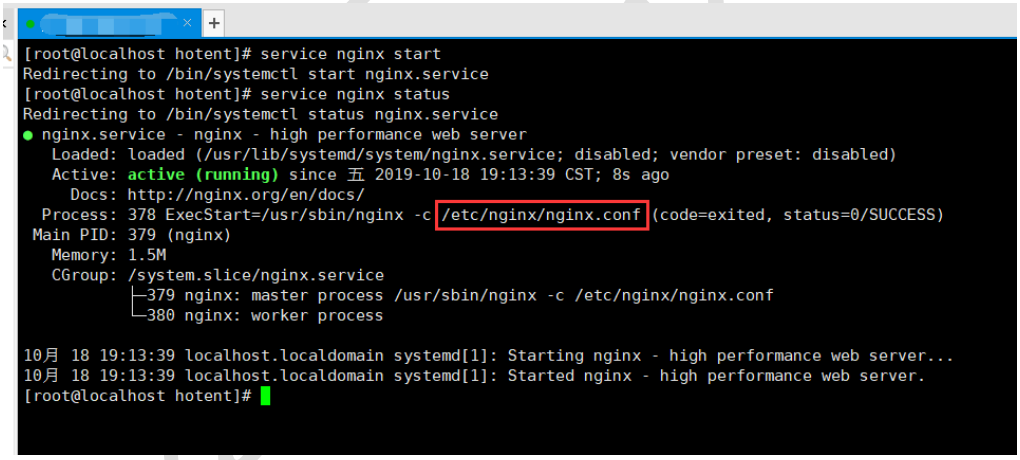



```
rpm -Uvh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.ngx.noarch.rpm
```



然后重新执行 yum install -y nginx

2. 通过命令启动 nginx 服务，并且查看服务的状态找到服务对应的配置文件



3. 如上图所示，配置文件在/etc/nginx/nginx.conf，通过 vim 命令修改这个文件，按照如下内容修改，修改完以后，:wq 保存，在/etc/nginx/目录下放入 html 文件夹，文件夹中放入前端编译后的文件，执行 nginx -s reload 重新载入配置文件

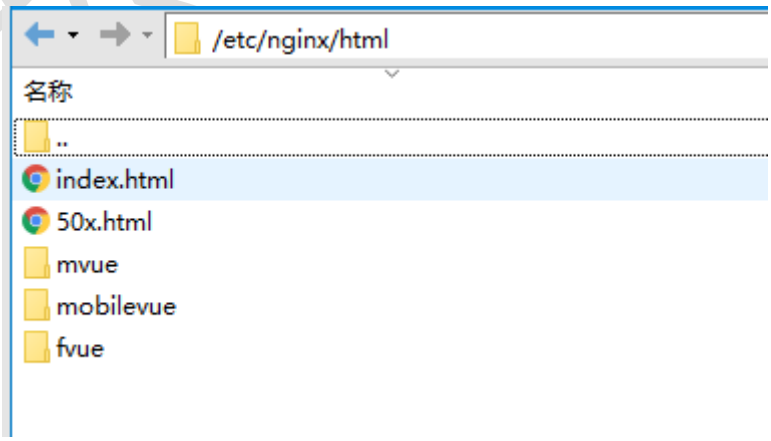
```
http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    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;

    keepalive_timeout  65;

    #gzip  on;

    include /etc/nginx/conf.d/*.conf;

    server {
        listen       8080;
        server_name  192.168.111.133;

        location / {
            root   html;
            index  index.html index.htm;
        }

        location /mvue {
            root   html;
            index  index.html index.htm;
            error_page 404 /mvue/index.html;
        }
        location /fvue {
            root   html;
            index  index.html index.htm;
            error_page 404 /fvue/index.html;
        }

        location /mobilevue {
            root   html;
            index  index.html index.htm;
            error_page 404 /mobilevue/index.html;
        }
    }
```

配置 error_page 解决访问报 404 错误

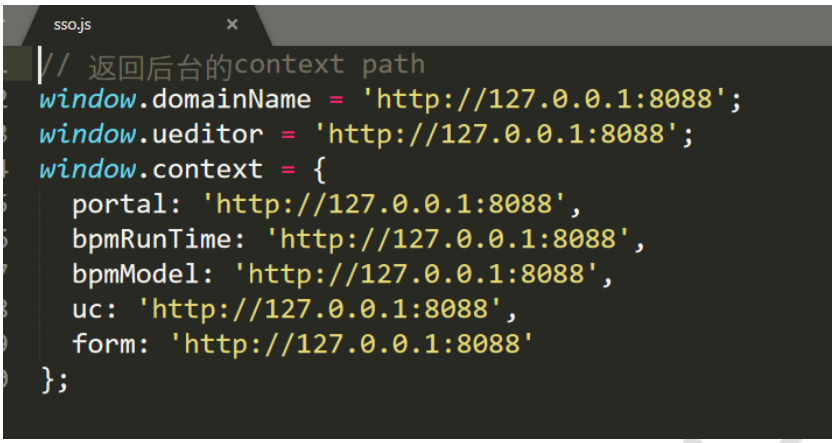
4. 可以通过 systemctl enable nginx.service 将服务设置为开机自动运行。

4.2.2 部署 web 端

对于 nginx 部署的方式，我们将 web 资源部署到 nginx 配置的 html 目录



修改 mvue/sso.js 和 mobilevue/sso.js 以及 fvue/sso.js 这三个 js 文件，将这些地址修改为 eip.jar 服务所发布的 IP 和端口。



另外，对于 mvue 目录下的 sso.js 文件，需要注意：front,mobile,manage 分别为 fvue,mobilevue,mvue 部署的地址。

以上，完成了整个 EIP7 在 linux 环境下的部署，现在通过浏览器访问 http://ip:port/mvue 来访问系统吧。