

Linux 环境 EIP7 产品部署手册

广州宏天软件股份有限公司

2020 年 4 月 20 日

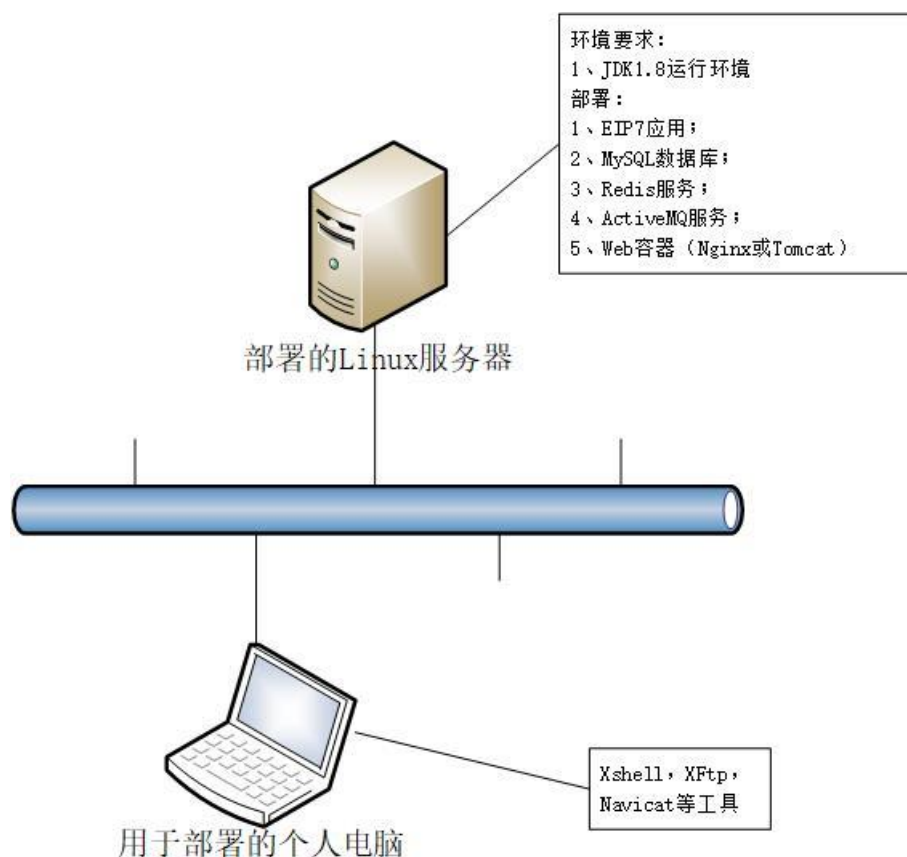
版本管理

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

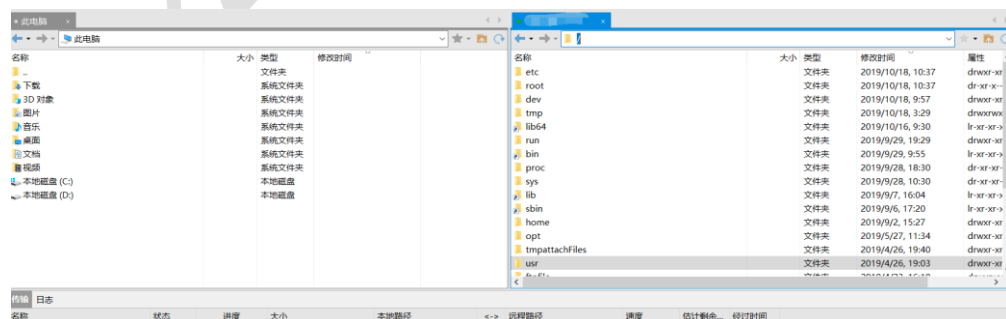
Linux 环境 EIP7 产品部署手册.....	1
1 总体部署图.....	4
2 常用 linux 命令.....	4
3 运行环境和中间件安装	6
3.1 安装 JDK1.8.....	6
3.2 安装 MySQL	8
3.3 安装 Redis.....	12
3.4 安装 ActiveMQ.....	14
4 应用部署	15
4.1 Java 应用.....	15
4.2 Web 应用	18
4.2.1 Nginx 的安装和配置.....	18
4.2.2 部署 web 端.....	20

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

```
[root@localhost local]# cd /usr/local 进入目录
[root@localhost local]# ls 列出目录下的内容
apache-tomcat-7.0.62 bin docker etc games go include lib lib64 libexec mysql sbin share src
[root@localhost local]# find / -name my.cnf 全系统查找指定文件
find: '/proc/25660': 没有那个文件或目录
/usr/local/docker/mysql/my.cnf
/etc/my.cnf
[root@localhost local]# ps -fe | grep tomcat 查找指定进程
root      25710  3664  0 14:20 pts/1    00:00:00 grep  --color=auto tomcat
[root@localhost local]# kill -9 25710 杀掉指定的进程
-bash: kill: (25710) - 没有那个进程
[root@localhost local]# tail -f /etc/profile 以实时刷新的方式显示指定的文件
done

unset i
unset -f pathmunge
JAVA_HOME=/usr/java/jdk1.8.0_191
CLASSPATH=$JAVA_HOME/lib/
PATH=$PATH:$JAVA_HOME/bin
export PATH JAVA_HOME CLASSPATH
export GOROOT=/usr/local/go
export PATH=$PATH:$GOROOT/bin
^C
[root@localhost local]# telnet 127.0.0.1 6379 测试系统的指定端口是否通畅
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
^]
telnet> quit
Connection closed.
[root@localhost local]#
```

```
[root@localhost home]# ls
cas docker go hrsj liyg htucweb joffice log4j.xml openkm PropertyGroups.xml test x5-base-db.properties xingqitest
demo ent hrsj htiia huiding ldap mysqldata openkmrepo simsun.ttc webapp x5test
[root@localhost home]# mv x5-base-db.properties /home/test/x5.properties 可以移动文件到另外的目录，也可以用来修改文件名
[root@localhost home]# cd /home/test/
[root@localhost test]# ls
build cas.war nohup.out x5.properties
[root@localhost test]# rm -rf x5.properties 可以删除文件或文件夹
[root@localhost test]# unzip x5.zip 可以解压缩，只支持zip格式
```

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。

```
[mysql]
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

"/etc/my.cnf" 20L, 595C
```

```
[mysql.d]
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 mysql 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 添加8088端口到白名单
success
[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

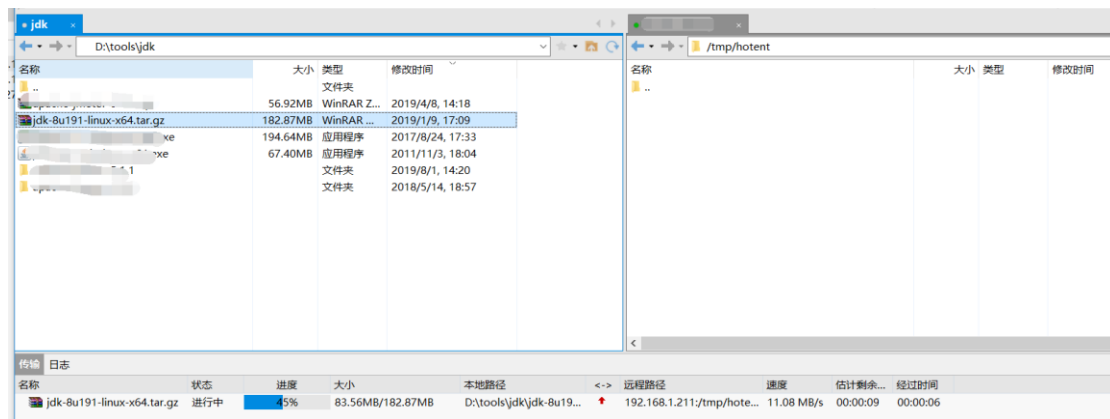
3.1 安装 JDK1.8

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

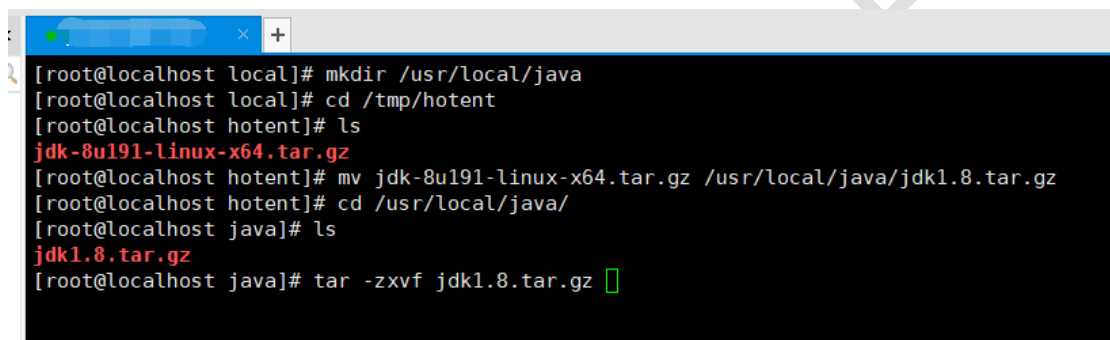
```
[root@localhost ~]# java -version
java version "1.8.0_191"
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)
[root@localhost ~]#
```

如果 JDK 未安装或者版本不正确，则重新安装 JDK。

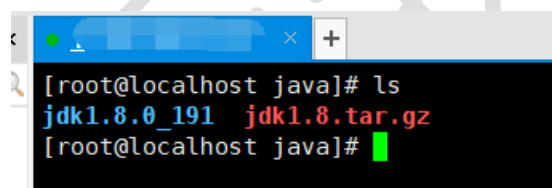
1. 将 jdk 的安装包拷贝到 linux 服务器（通过 xftp 上传上去，在 /tmp 目录下创建一个 hotent 目录，用于存放上传的文件）



2. 将该文件拷贝到/usr/local/java 目录下，并解压



3. 查看解压出来的目录名

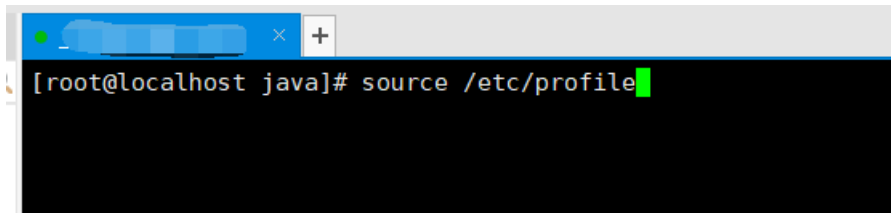


4. 配置 jdk 到环境变量，通过 vim 命令编辑/etc/profile，进入编辑模式后，shift+g 定位到文件的最后一行，点击 insert 命令进入编辑模式，编辑完以后按 esc 按钮，输入:wq+回车则保存文件，输入:q!则放弃保存。

JACA_HOME 为 jdk 路径

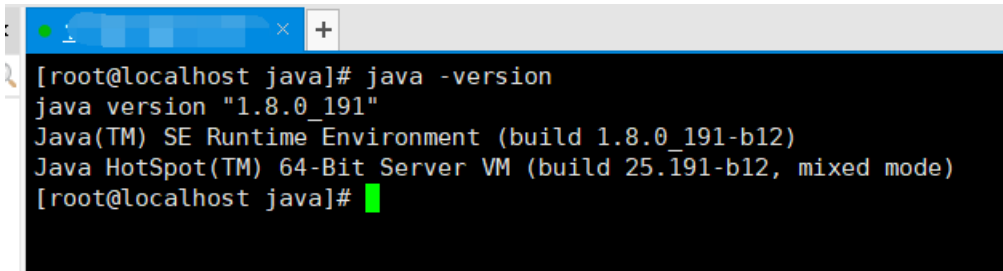


- 完成配置后输入 `source /etc/profile` 刷新环境变量



```
[root@localhost java]# source /etc/profile
```

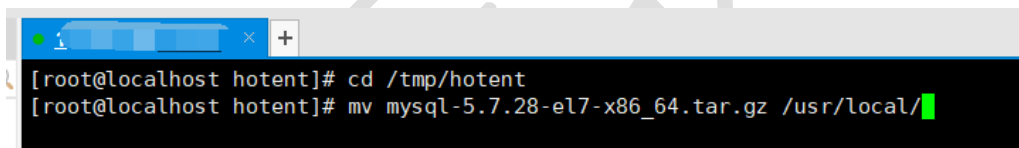
- 现在输入 `java -version` 确认 jdk 环境变量是否配置正确，版本是否一致



```
[root@localhost java]# 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 java]#
```

3.2 安装 MySQL

- 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

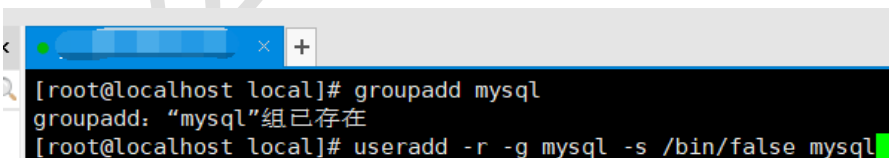


```
[root@localhost hotent]# cd /tmp/hotent
[root@localhost hotent]# mv mysql-5.7.28-el7-x86_64.tar.gz /usr/local/
```

- 进入 `/usr/local` 目录，为 mysql 安装目录创建软链接

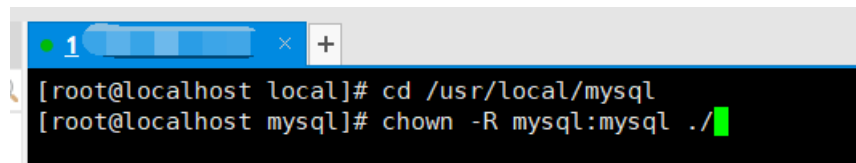
```
[root@localhost hotent]# cd /usr/local/
[root@localhost local]# ln -s mysql-5.7.28-el7-x86_64 mysql
```

- 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)



```
[root@localhost local]# groupadd mysql
groupadd: "mysql"组已存在
[root@localhost local]# useradd -r -g mysql -s /bin/false mysql
```

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



```
[root@localhost local]# cd /usr/local/mysql
[root@localhost mysql]# chown -R mysql:mysql ./
```

- 安装 mysql

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

```
[root@localhost local]# cd /usr/local/mysql
[root@localhost mysql]# ./bin/mysqld --user=mysql --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
```

```
root@lulu-VirtualBox:/usr/local/mysql# apt-get install libaio1
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列【新】软件包将被安装：
libaio1
升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 264 个软件包未被升级。
需要下载 6,304 B 的归档。
解压后需要 30.7 kB 的额外空间。
获取 1 http://archive.kylinos.cn/yhkylin juniper/main amd64 libaio1 amd64 0.3.110-2kord [6,304 B]
已下载 6,304 B，耗时 20秒 (213 B/s)
正在选中要选择的软件包 libaio1:amd64。
(正在读取数据库 ... 系统当前共安装有 188382 个文件和目录。)
准备复制 .../libaio1_0.3.110-2kord_amd64.deb ...
正在解包 libaio1:amd64 (0.3.110-2kord) ...
正在处理用于 libc-bin (2.23-0kord10) 的触发器 ...
正在设置 libaio1:amd64 (0.3.110-2kord) ...
正在处理用于 libc-bin (2.23-0kord10) 的触发器 ...
root@lulu-VirtualBox:/usr/local/mysql# ./bin/mysqld --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data --initialize
2018-07-18T15:11:07.095595Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
2018-07-18T15:11:14.026864Z 0 [Warning] InnoDB: New log files created, LSN=45790
2018-07-18T15:11:14.347786Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
2018-07-18T15:11:14.459806Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: d0093d02-8a9c-11e8-8116-000c29d81287.
2018-07-18T15:11:14.464353Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
2018-07-18T15:11:14.471692Z 1 [Note] A temporary password is generated for root@localhost: BGldj0&m3Vq.
root@lulu-VirtualBox:/usr/local/mysql#
```

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

```
drwxr-xr-x. 2 root root 4096 Jul 18 08:09 support-files
[root@hadoop02 mysql]# chown -R mysql:mysql ./
[root@hadoop02 mysql]# ./bin/mysqld --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data --initialize
2018-07-18T15:11:07.095595Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
2018-07-18T15:11:14.026864Z 0 [Warning] InnoDB: New log files created, LSN=45790
2018-07-18T15:11:14.347786Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
2018-07-18T15:11:14.459806Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: d0093d02-8a9c-11e8-8116-000c29d81287.
2018-07-18T15:11:14.464353Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
2018-07-18T15:11:14.471692Z 1 [Note] A temporary password is generated for root@localhost: BGldj0&m3Vq.
[root@hadoop02 mysql]#
[root@hadoop02 mysql]#
```

<https://blog.csdn.net/dc282614966>

7. 启动 mysql 服务

```
[root@localhost mysql]# ./support-files/mysql.server start
```

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]#
```

<https://blog.csdn.net/dc282614966>

```
[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"

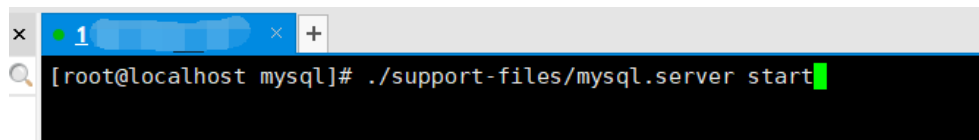
# 开启 ip 绑定
bind-address = 0.0.0.0

[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



```
[root@localhost mysql]# ./support-files/mysql.server start
```

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

<https://blog.csdn.net/dc282614966>

10. 将 mysql 进程放入系统进程中，并重新启动 mysql 服务

```
[root@localhost mysql]# cp support-files/mysql.server /etc/init.d/mysql
```

```
[root@localhost mysql]# service mysqld restart
```

```
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
1 --log-error=/var/log/mysql.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/mysql
[root@hadoop02 mysql]# service mysqld restart
Shutting down MySQL. SUCCESS!
Starting MySQL. SUCCESS!
[root@hadoop02 mysql]#
```

<https://blog.csdn.net/dc282614966>

11. 配置 mysql 的环境变量，并重新编译环境变量

```
[root@localhost mysql]# vim /etc/profile
```

```
unset i
unse
C
PAT
ex
O
export PATH
export PATH=$PATH:/usr/local/mysql/bi
```

```
[root@localhost mysql]# vim /etc/profile
[root@localhost mysql]# source /etc/profile
```

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）

```
[root@localhost mysql]# mysql -u root -p
Enter password:
```

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]# source /etc/profile
[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> 
```

<https://blog.csdn.net/dc282614966>

```
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-MariaDB!+:~2x,k _00~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

1. 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
2. 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]# 
```

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

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

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

6. 复制 redis 的配置文件

7. 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
# supervised systemd - signal systemd by writing pid= to the 'pidfile'
# supervised unix - signal systemd by writing pid= to the 'pidfile'
[root@CDH-143 redis-4.0.11]#
```

8. 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
```

```
#!/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
```

9. 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
```

10. 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 # 0000000000000000 Redis is starting 0000000000000000
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
```

11. 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

```
[root@localhost hotent]# ls
apache-activemq-5.15.8-bin.tar.gz
[root@localhost hotent]# tar -zxvf apache-activemq-5.15.8-bin.tar.gz -C /usr/local/
apache-activemq-5.15.8/
apache-activemq-5.15.8/conf/
apache-activemq-5.15.8/data/
apache-activemq-5.15.8/docs/
apache-activemq-5.15.8/examples/
apache-activemq-5.15.8/examples/amqp/
apache-activemq-5.15.8/examples/amqp/java/
```

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

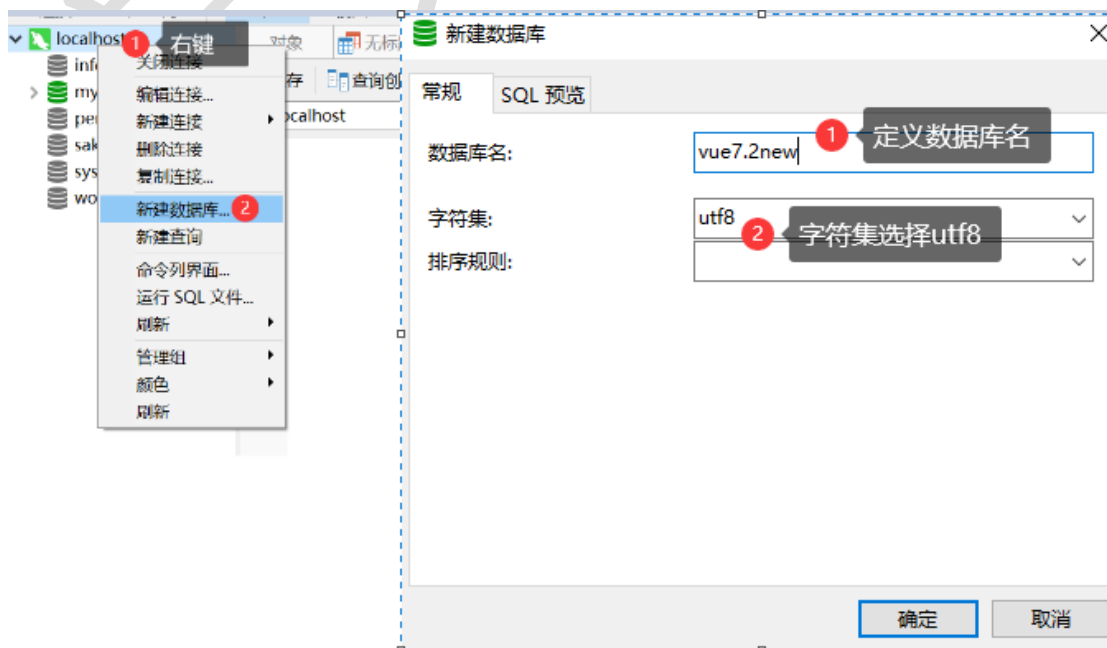
```
[root@localhost hotent]# cd /usr/local
[root@localhost local]# ls
apache-activemq-5.15.8 apache-tomcat-7.0.62 bin docker etc games go include lib lib64 libexec mysql sbin share src
[root@localhost local]# cd apache-activemq-5.15.8/
[root@localhost apache-activemq-5.15.8]# ls
activemq-all-5.15.8.jar bin conf data docs examples lib LICENSE NOTICE README.txt webapps webapps-demo
[root@localhost apache-activemq-5.15.8]# cd bin
[root@localhost bin]# ls
activemq activemq-diag activemq.jar env linux-x86-32 linux-x86-64 macosx wrapper.jar
[root@localhost bin]# cd linux-x86-64/
[root@localhost linux-x86-64]# ls
activemq libwrapper.so wrapper wrapper.conf
[root@localhost linux-x86-64]# ./activemq start
```

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

4 应用部署

4.1 Java 应用

1. 使用数据库可视化工具（如 navicat）连接服务器数据库新建数据库，选择字符集(utf-8)和排序规则,数据库名称自定义，如下图所示：



执行数据库脚本，选择对应数据库的初始化 sql 执行，然后执行索引初始化 sql

以及初始化数据 sql

名称	修改日期
change	2020/4/13 11:13
mysql 1	2020/4/13 19:31
oracle	2020/4/13 11:13
pdm	2020/4/13 11:13
sqlserver2008	2020/4/13 11:13
sqlserver2012	2020/4/13 11:13
初始化数据 3	2020/4/14 17:25
索引index 2	2020/4/13 11:15
初始化sql执行说明.docx	2020/4/7 11:54

将要部署的 eip.jar, eip.sh 和 application-uat.yml 上传到 /home/hotent 目录，修改 application-uat.yml 文件中数据库参数、redis 参数、mq 参数以及 server 端口地址。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

文件名	大小	类型
application-uat.yml	630 B	
eip.jar	115.9 MB	JAR
eip.sh	1.2 KB	

```

1 spring:
2   profiles:
3     active: dev
4   datasource:
5     dynamic:
6       datasource:
7         master:
8           username: root
9           password: N7V2jrxE9sxE3pNH56jgJ+Du3WbX14gwu9vVKPr/L8H9p627IfQvEhMErSVRyl
10          driver-class-name: com.mysql.cj.jdbc.Driver
11          url: jdbc:mysql://192.168.1.21:3306/7.2.2_dev?useUnicode=true&characterEncoding=utf8
12          druid:
13            public-key: MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBABJApc+rYF1XQW4v037oqYlkWhl
14      redis:
15        database: 6
16        host: 10.10.10.59
17        port: 6379
18        password:
19      activemq:
20        broker-url: tcp://10.10.10.59:61616
21        user:
22        password:
23

```

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

* password: (java.lang.String) 必填 加密

* url: (java.lang.String)

* initialSize: (int)

* 公钥: (java.lang.String)

测试连接 保存 取消

2. 配置执行权限给 eip.sh，并通过命令启动应用服务，应用的日志输出到 eiplog.out 文件，可以通过 tail -f eiplog.out 命令来查看

```
[root@localhost hotent]# chmod +x eip.sh
[root@localhost hotent]# ./eip.sh start
eip.jar start success
[root@localhost hotent]# tail -f eiplog.out

:: Spring Boot :: (v2.0.1.RELEASE)
```

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。

```
-bash: nginx: 未找到命令
[root@localhost hotent]# yum install -y nginx
已加载插件: fastestmirror
base
extras
updates
updates/7/x86_64/primary_db
Loading mirror speeds from cached hostfile
没有可用软件包 nginx。
错误: 无须任何处理
[root@localhost hotent]#
```

```
rpm -Uvh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7ngx.noarch.rpm
```

```
[root@localhost hotent]# rpm -Uvh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7ngx.noarch.rpm
获取http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7ngx.noarch.rpm
警告: /var/tmp/rpm-tmp.ElpuSj: 头V4 RSA/SHA1 Signature, 密钥 ID 7bd9bf62: NOKEY
准备中... ##### [100%]
正在升级/安装...
  1:nginx-release-centos-7-0.el7ngx ##### [100%]
[root@localhost hotent]#
```

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

```
[root@localhost hotent]# service nginx start
Redirecting to /bin/systemctl start nginx.service
[root@localhost hotent]# service nginx status
Redirecting to /bin/systemctl status nginx.service
● nginx.service - nginx - high performance web server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
   Active: active (running) since 五 2019-10-18 19:13:39 CST; 8s ago
     Docs: http://nginx.org/en/docs/
  Process: 378 ExecStart=/usr/sbin/nginx -c /etc/nginx/nginx.conf (code=exited, status=0/SUCCESS)
 Main PID: 379 (nginx)
    Memory: 1.5M
    CGroup: /system.slice/nginx.service
            └─379 nginx: master process /usr/sbin/nginx -c /etc/nginx/nginx.conf
              └─380 nginx: worker process

10月 18 19:13:39 localhost.localdomain systemd[1]: Starting nginx - high performance web server...
10月 18 19:13:39 localhost.localdomain systemd[1]: Started nginx - high performance web server.
[root@localhost hotent]#
```

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 Tomcat 的部署和配置

1. 将 tomcat 的 zip 包上传到/tmp/hotent 目录，并解压至/usr/local 目录

```
[root@localhost webapps]# cd /tmp/hotent
[root@localhost hotent]# ls
apache-tomcat-7.0.62.zip
[root@localhost hotent]# mv apache-tomcat-7.0.62.zip /usr/local
[root@localhost hotent]# cd /usr/local/
[root@localhost local]# ls
apache-tomcat-7.0.62  apache-tomcat-7.0.62.zip  bin  docker  etc  games  go  include  lib  lib64  libexec  mysql  sbin  share  src
[root@localhost local]# unzip apache-tomcat-7.0.62.zip
```

2. 进入 tomcat 的 bin 目录，给 startup.sh 和 catalina.sh 文件授予执行权限

```
[root@localhost local]# cd apache-tomcat-7.0.62
[root@localhost apache-tomcat-7.0.62]# ls
bin  conf  lib  LICENSE  logs  NOTICE  RELEASE-NOTES  RUNNING.txt  temp  webapps  work
[root@localhost apache-tomcat-7.0.62]# cd bin/
[root@localhost bin]# ls
bootstrp.jar  catalina.bat  catalina-tasks.xml  commons-daemon-native.tar.gz  configtest.sh  digest.bat  perfStats.log  setclasspath.bat
cas.log       catalina.sh    commons-daemon.jar  configtest.bat                daemon.sh      digest.sh    service.bat    setclasspath.sh
[root@localhost bin]# chmod +x startup.sh
[root@localhost bin]# chmod +x catalina.sh
[root@localhost bin]#
```

3. 进入 tomcat 的 conf 目录，通过 vim 命令修改 web.xml 文件，该文件内容较多，用快捷键 `shift+g` 定位到文件最后，添加如下图所示的内容

```

1 192.168.1.211 x +
[root@localhost conf]# cd ../conf/
[root@localhost conf]# ls
Catalina catalina.policy catalina.properties context.xml logging.properties server.xml tomcat-users.xml web.xml
[root@localhost conf]# vim web.xml

<!-- ===== Default Welcome File List ===== -->
<!-- When a request URI refers to a directory, the default servlet looks -->
<!-- for a "welcome file" within that directory and, if present, to the -->
<!-- corresponding resource URI for display. -->
<!-- If no welcome files are present, the default servlet either serves a -->
<!-- directory listing (see default servlet configuration on how to -->
<!-- customize) or returns a 404 status, depending on the value of the -->
<!-- listings setting. -->
<!-- -->
<!-- If you define welcome files in your own application's web.xml -->
<!-- deployment descriptor, that list *replaces* the list configured -->
<!-- here, so be sure to include any of the default values that you wish -->
<!-- to use within your application. -->

<welcome-file-list>
  <welcome-file>index.html</welcome-file>
  <welcome-file>index.htm</welcome-file>
  <welcome-file>index.jsp</welcome-file>
</welcome-file-list>

<error-page>
  <error-code>404</error-code>
  <location>/index.html</location>
</error-page>

/web-app>

```

解决vue页面在刷新时报404错误的问题

4.进入 tomcat 的 bin 目录，执行./startup.sh 来启动 tomcat，可以通过 tail 命令查看启动日志，如下图所示，tomcat 发布的端口为 8080

```

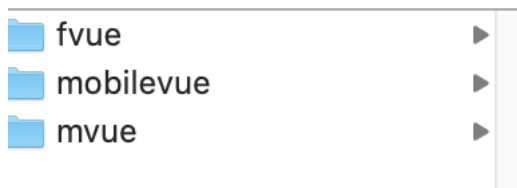
[root@localhost bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/apache-tomcat-7.0.62
Using CATALINA_HOME:   /usr/local/apache-tomcat-7.0.62
Using CATALINA_TMPDIR: /usr/local/apache-tomcat-7.0.62/temp
Using JRE_HOME:         /usr/java/jdk1.8.0_191
Using CLASSPATH:        /usr/local/apache-tomcat-7.0.62/bin/bootstrap.jar:/usr/local/apache-tomcat-7.0.62/bin/tomcat-juli.jar
Tomcat started.
[root@localhost bin]# tail -f ../logs/catalina.out
十月 18, 2019 8:04:46 下午 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory /usr/local/apache-tomcat-7.0.62/webapps/ROOT
十月 18, 2019 8:04:46 下午 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory /usr/local/apache-tomcat-7.0.62/webapps/ROOT has finished in 22 ms
十月 18, 2019 8:04:46 下午 org.apache.coyote.AbstractProtocol start
信息: Starting ProtocolHandler ["http-bio-8080"]
十月 18, 2019 8:04:46 下午 org.apache.coyote.AbstractProtocol start
信息: Starting ProtocolHandler ["ajp-bio-8009"]
十月 18, 2019 8:04:46 下午 org.apache.catalina.startup.Catalina start
信息: Server startup in 701 ms

```

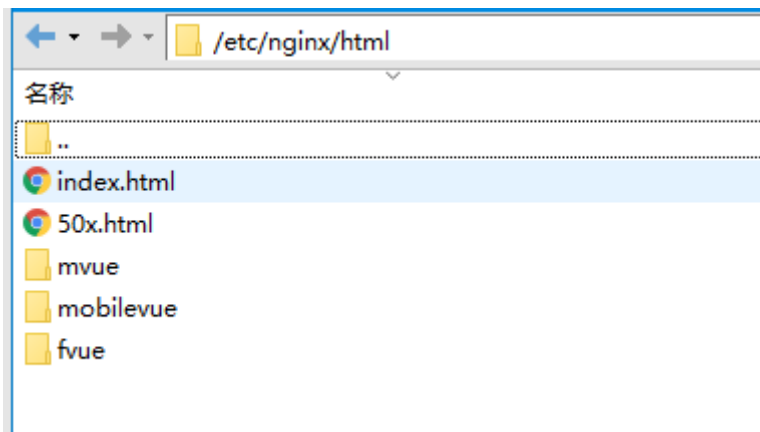
注：Nginx 和 Tomcat 分别为两个相同的 web 端部署服务，用户可根据需求使用其中一种即可。

4.2.3 部署 web 端

对于 Tomcat 部署的方式，我们将 web 资源部署到 Tomcat 配置的 webapps 目录里。Web 里文件名称需更改为对应的 mvue，fvue，mobile，如名字已经对应，则无需更改。



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。

```
sso.js
// 返回后台的context path
window.domainName = 'http://127.0.0.1:8088';
window.ueditor = 'http://127.0.0.1:8088';
window.context = {
  portal: 'http://127.0.0.1:8088',
  bpmRunTime: 'http://127.0.0.1:8088',
  bpmModel: 'http://127.0.0.1:8088',
  uc: 'http://127.0.0.1:8088',
  form: 'http://127.0.0.1:8088'
};
```

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。

```
// 返回后台的context path localhost:8088
window.domainName = 'http://192.168.111.133:8080/mvue';
window.ueditor = "/mvue"
window.context = {
  front: 'http://192.168.111.133:8080/fvue', //前端页面
  mobile: 'http://192.168.111.133:8080/mobilevue', //手机端页面
  manage: '/mvue', //管理端页面
  portal: 'http://192.168.111.133:8088',
  bpmRunTime: 'http://192.168.111.133:8088',
  bpmModel: 'http://192.168.111.133:8088',
  uc: 'http://192.168.111.133:8088',
  form: 'http://192.168.111.133:8088'
};
```

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。

访问系统第一步：进入设计中心-模板管理 点击初始化模板！

宏天软件