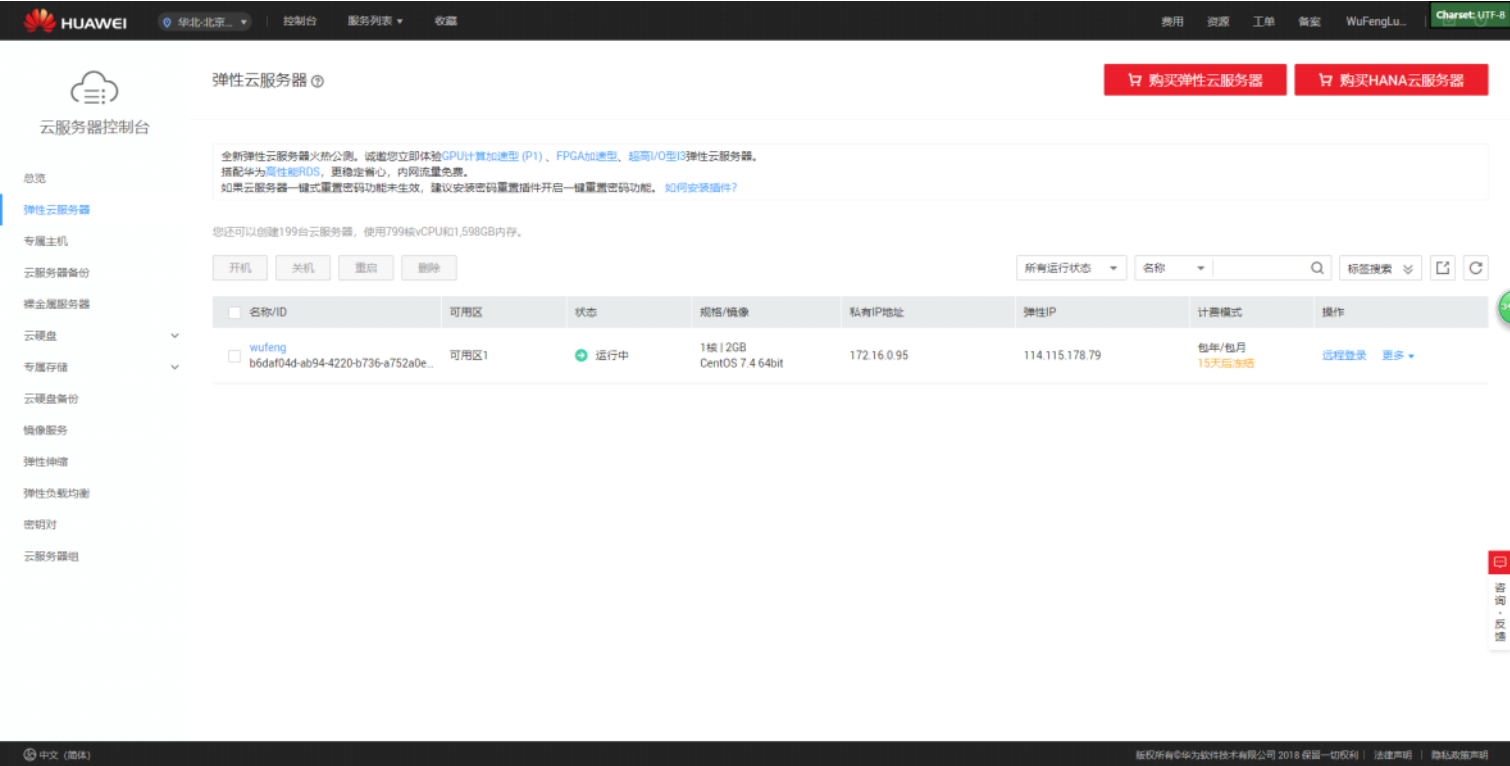


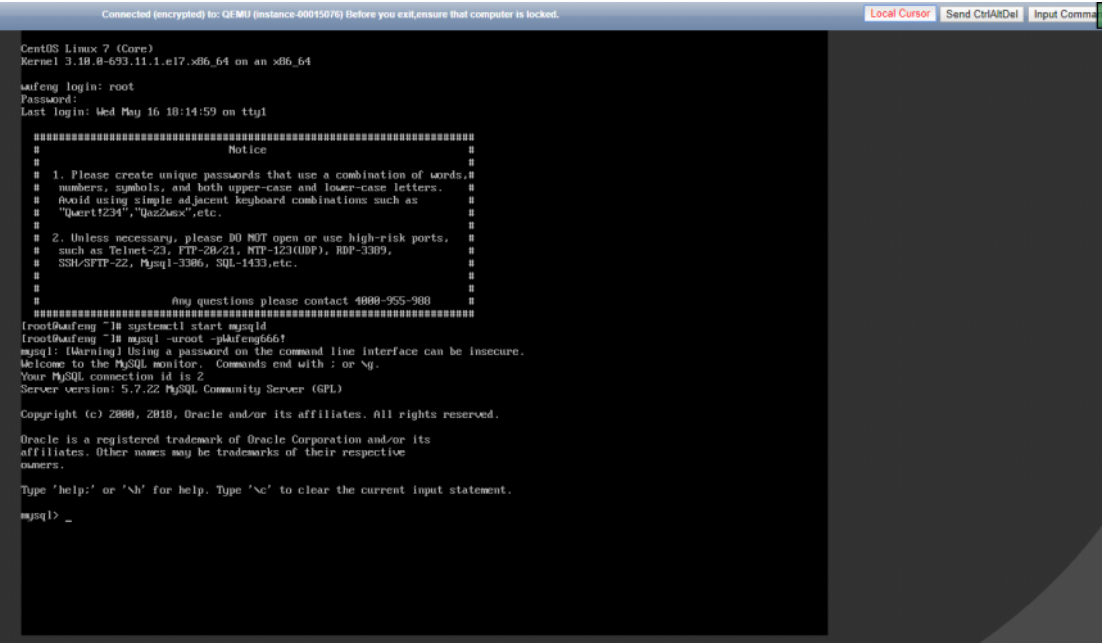
一、购买云服务器，并且远程登录云服务器

1、登入华为云官网申请免费的弹性云服务器(免费15天)。设置账户和密码，进入控制台后，选择弹性云服务器。(申请的是centos linux 7.4系统)



里面还有一个弹性IP地址(就是所谓的公网IP地址)，其他客户端可以通过该IP地址访问该云服务器。

2、点击远程登录，login: root Password: 你设置的密码



这样就登陆了云服务器了。然后就是开始搭建环境。

二、搭建环境(包括: jdk-8u172-linux-x64、apache-tomcat-8.0.52.tar、mysql57-community-release-el7-1.noarch.repo、redis-4.0.9.tar)

- 1、在官网下载相应的安装包本地，mysql可以先不用下载。
 - 2、下载filezilla软件，并在本地安装。并利用filezilla将本地的安装包上传到云服务器，具体步骤参考如下网址https://blog.csdn.net/oaim_zh/article/details/78293701
 - 3、安装jdk。
- 先在云服务器上利用cd + jdk包的目录，然后调用如下linux命令

rpm -ivh jdk-8u172-linux-x64.rpm
完成安装后，利用 java -version查看jdk的版本信息，同时可以检查是否安装成功

4、安装mysql。可以参考一下网址:
<https://blog.csdn.net/aiyongvyy/article/details/79125894>

```
5.
安装MySQL

下载mysql源安装包
wget http://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm

安装mysql源
yum localinstall mysql57-community-release-el7-8.noarch.rpm

检查mysql源是否安装成功
yum repolist enabled | grep "mysql.*-community.*"

安装MySQL
yum install mysql-community-server

启动MySQL服务
systemctl start mysqld

查看MySQL的启动状态
systemctl status mysqld

开机启动
systemctl enable mysqld

systemctl
daemon-reload

修改root本地登录密码
mysql安装完成之后，在/var/log/mysqld.log文件中给root生成了一个默认密码。通过下面的方式找到root默认密码，然后登录mysql进行修改：

grep 'temporary password' /var/log/mysqld.log

mysql -uroot -p 刚刚得到的临时密码

ALTER USER 'root'@'localhost' IDENTIFIED BY 'MyNewPass4!';

注意新密码需要满足mysql的默认密码规则（大写，小写，数字，特殊字符都得有），也可以修改mysql的密码规则

修改mysql的密码规则： set global validate_password_policy=0;
```

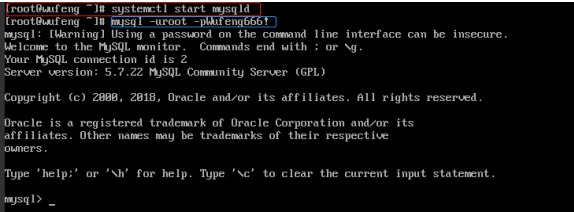
修改root本地登录密码

mysql安装完成之后，在/var/log/mysqld.log文件中给root生成了一个默认密码。通过下面的方式找到root默认密码，然后登录mysql进行修改：

grep 'temporary password' /var/log/mysqld.log

获取到mysql初始的密码后，通过mysql -uroot -p密码 登录mysql。通过下面的语句来更改密码

ALTER USER 'root'@'localhost' IDENTIFIED BY '新的密码';
新的密码的命名规则：大小写字母，数字，特殊符号。例如：
Wufeng666!



红色的方框表示启动mysql，
蓝色方框表示登录mysql。
接下来需要开启远程连接，通常云服务器不会开放mysql的3306端口，因此我们需要自己来开放这个端口，使得本地的客户端可以连接到该mysql服务器。

名称/ID	可用区	状态	规格/镜像	私有IP地址
<div><div></div><div>wufeng b6daf04d-ab94-4220-b736-a752a0e...</div></div>	wufeng b6daf04d-ab94-4220-b736-a752a0e679b8		1核 2GB CentOS 7.4 64bit	172.16.0.95

点击这个名称，然后点击安全组-->更改安全组规则-->快速添加规则。选择相应的端口打钩，然后确定。

云硬盘 网卡 安全组 弹性IP 监控 标签

更改安全组

sg16366ae3f6c44061

Nic1: 172.16.0.95

出方向规则 1

入方向规则 6

ID: 848333e5-d5a4-4054-8ba6-8ddc392f4ac7

更改安全组规则

方向	类型	协议	端口范围/ICMP类型	远端
出方向	IPv4	Any	Any	Any
入方向	IPv4	Any	Any	sg16366ae3f6c44061 (848333e5-d5a4-40...
入方向	IPv4	TCP	22	0.0.0.0/0
入方向	IPv4	TCP	3389	0.0.0.0/0
入方向	IPv4	TCP	80	0.0.0.0/0
入方向	IPv4	TCP	3306	0.0.0.0/0
入方向	IPv4	TCP	6379	0.0.0.0/0

规则 服务器

添加规则 快速添加规则 出方向规则: 1 入方向规则: 6

方向	类型	协议	端口范围/ICMP类型	远端	操作
出方向	IPv4	Any	Any	Any	删除
入方向	IPv4	Any	Any	sg16366ae3f6c44061 (848333e5-d5a4-4054-8ba6-8ddc392f4ac7)	删除
入方向	IPv4	TCP	22	0.0.0.0/0	删除

快速添加规则

* 方向

入方向 出方向

常用端口

☐ SSH(22)

☐ telnet(23)

☒ HTTP(80)

☐ HTTPS(443)

☐ MS SQL(1433)

☐ Oracle(1521)

☒ My SQL(3306)

☐ RDP(3389)

☐ PostgreSQL(5432)

☒ Redis(6379)

自定义端口

TCP 示例: 22或22-30

* 源地址

IP地址 安全组

0.0.0.0/0 0

确定 取消

开放完端口之后，登录到云服务器的mysql中， use mysql --> grant select,update,insert,delete on *.* to 'work'@'%' identified by '密码'; ----> flush privileges; 使配置生效
work 代表user的名字， %为host的名字， 表示所有的外部客户端都可以连接。用于本地Navicat连接云服务器上的这个mysql的用户。

新建连接

常规 高级 SSL SSH HTTP

连接名:

主机名称或 IP 地址: localhost

端口: 3306

用户名: root

密码:

☒ 保存密码

连接测试 确定 取消

在Navicat上，新建连接-->在黄色区域随便填写（一般写云的名字），红色部分填写弹性IP地址，蓝色部分填写上面部分创建的mysql的用户： work。 ---->连接测试

5、安装redis。

(1) 解压redis安装包

tar -zxvf redis-4.0.9.tar.gz

由于首次安装，所以还需要安装gcc-c++

yum install gcc

安装完成后，进入到redis-4.0.9

cd redis-4.0.9

在该目录下，通过ls 可以看到该目录下的所有文件，通过

vi redis.conf 编辑修改两处，

vi进入后，通过输入/bind查找到 bind=127.0.0.1。并将其用#注释(蓝色的地方是首先查找的地方，按键盘的向下按钮，找到红色标记处，按按钮可进行编辑，用#注释)

```
##### LOADMODULES #####
# Load modules at startup. If the server is not able to load modules
# it will abort. It is possible to use multiple loadmodule directives.
#
# loadmodule /path/to/my_module.so
# loadmodule /path/to/other_module.so
##### NETWORK #####
# By default, if no "bind" configuration directive is specified, Redis listens
# for connections from all the network interfaces available on the server.
# It is possible to listen to just one or multiple selected interfaces using
# the "bind" configuration directive, followed by one or more IP addresses.
#
# Examples:
#
# bind 192.168.1.100 10.0.0.1
# bind 127.0.0.1 ::1
#
# --- WARNING --- If the computer running Redis is directly exposed to the
# internet, binding to all the interfaces is dangerous and will expose the
# instance to everybody on the internet. So by default we uncomment the
# following bind directive, that will force Redis to listen only into
# the IPv4 loopback interface address (this means Redis will be able to
# accept connections only from clients running into the same computer it
# is running).
#
# IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES
# JUST COMMENT THE FOLLOWING LINE.
#
# bind 127.0.0.1
```

第一处修改后，按Esc按钮，再输入/daemonize，找到后，按按钮进行编辑，将daemonize no修改为daemonize yes. 按Esc按钮，然后输入:wq，就可以保存并退出。

```
##### 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 READY=1 to $NOTIFY_SOCKET
# supervised auto    - detect upstart or systemd method based on
#                      UPSTART_JOB or NOTIFY_SOCKET environment variables
# Note: these supervision methods only signal "process is ready."
#       They do not enable continuous liveness pings back to your supervisor.
# supervised no
#
# If a pid file is specified, Redis writes it where specified at startup
# and removes it at exit.
#
# When the server runs non daemonized, no pid file is created if none is
# specified in the configuration. When the server is daemonized, the pid file
# is used even if not specified, defaulting to "/var/run/redis.pid".
#
# Creating a pid file is best effort: if Redis is not able to create it
# nothing bad happens, the server will start and run normally.
# daemonize
```

修改完成后，redis.conf配置就可以支持redis守护进程，一直在后台运行，并且可以远程连接。在redis-4.0.9目录下，直接输入make 进行安装。

安装完成后，通过输入

src/redis-server redis.conf （该语句的意思是以redis.conf的配置运行redis）

启动完成后，通过

ps -ef | grep redis 查看redis进程。如下所示，则表示redis一直处于运行中，

```
[root@anfeng redis-4.0.9]# ps -ef | grep redis
root      14632      1  0 14:22  ?        00:00:01 src/redis-server *:6379
root      15420  2253  0 14:32  tty1    00:00:00 grep --color=auto redis
```

接下来，测试redis。输入 src/redis-cli，回车后，输入ping，会自动回复pong。

```
[root@anfeng redis-4.0.9]# src/redis-cli
127.0.0.1:6379> ping
PONG
127.0.0.1:6379> _
```

输入exit退出。进行Tomcat的安装。

6、安装Tomcat。cd到Tomcat的安装包目录中，运行

tar -zxvf apache-tomcat-8.0.52.tar.gz

解压安装包。然后运行

./apache-tomcat-8.0.52/bin/startup.sh

启动Tomcat。

环境搭建完成。