



# 云计算课程报告 1

专 业 计算机科学与技术

班 级 16052317

学 号 16041321

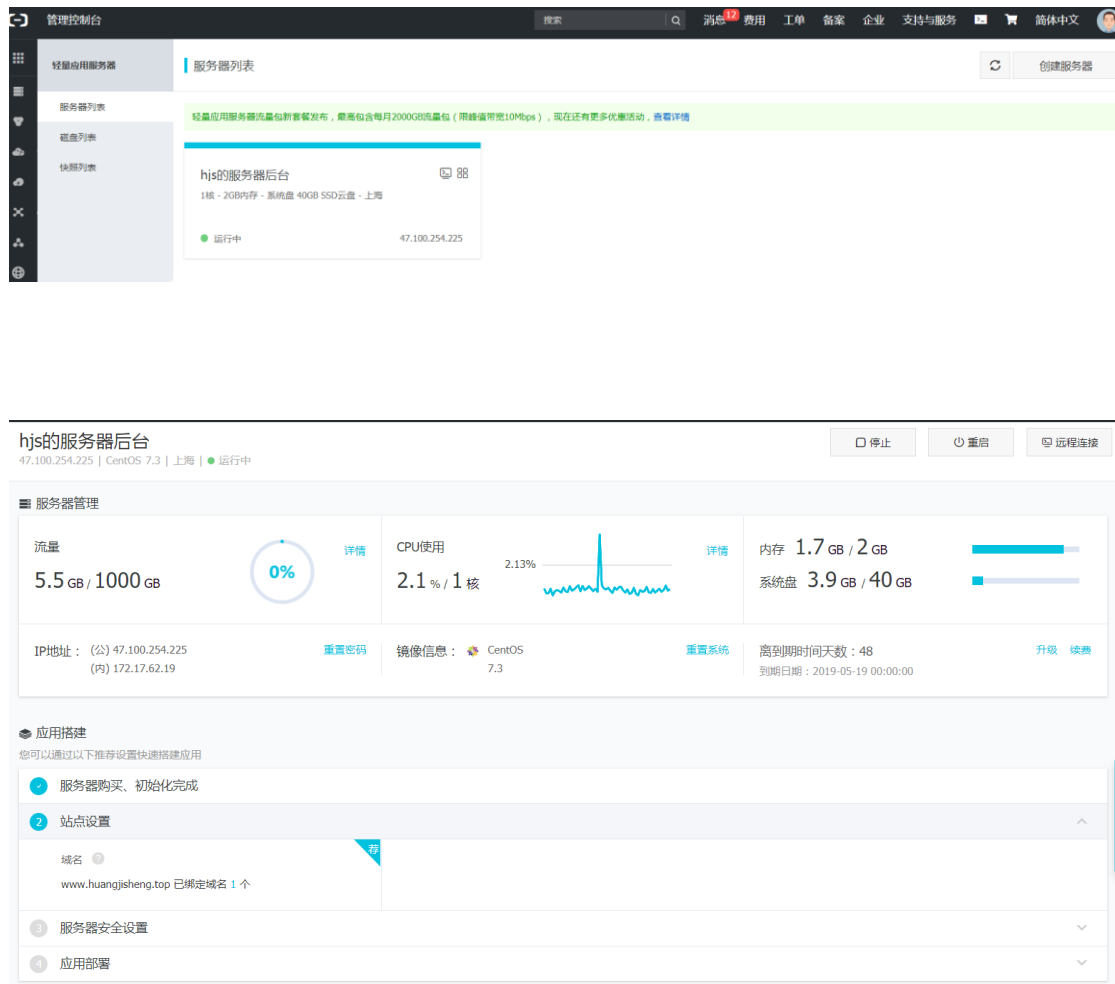
姓 名 黄继升

杭州电子科技大学计算机学院

2019 年 3 月

# 前言

之前在开发 javaweb 项目时购买过阿里云服务器和阿里云数据库，现在阿里云服务器仍在一直使用和续费中，但云数据库上个月就过期了，上星期想要续费云数据库时，发现好像价格变贵了，因此也不太好下手。所以我在这里把之前我曾经手动记录下来的关于阿里云服务器的搭建、DNS 域名的解析、SSL 证书的购买和云数据库的配置等过程加以整理，作为我这一次上机的实验报告内容。<https://huangjisheng.top> 是我的服务器地址。



## (1) 关于购买和使用阿里云轻量型应用服务器部署 java web 项目

实际上本来是想购买云服务器 ECS 的，但是由于重度手残买成了轻量型应用服务器，郁闷了很久。相比于传统 ECS，服务器的配置简化了，带宽也明显提升，当然轻量二字不是随便说的，有利也有弊。它的配置方式和 ECS 基本是一样的，但这里也把配置方式讲一下。

### 一.购买轻量型应用服务器

对于学生党们，阿里云有针对学生的云翼计划，每个月 10 块。

所以首先自己要先注册阿里云，然后经过个人验证以及学生认证，最后通过后就可以开始购买我们的学生服务器。

关于云翼计划，直接在百度上搜云翼计划进入后就可以了。

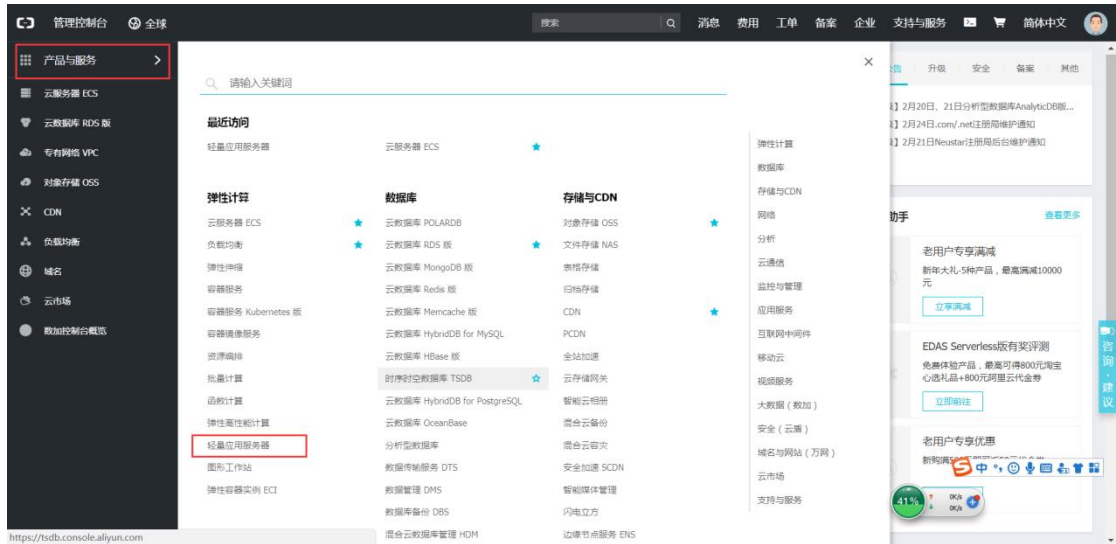
购买流程我简单讲一下，服务器的地址一般有华南区，华东区和华北区。实际上购买哪个地址的服务器都是可以的，你选择了哪个地区的服务器，该地区的用户人群访问你的服务器就会比其他地区的人快。我这里选择的是华东区的服务器。然后系统镜像我这里暂时选择的是 windows 版本的，这里后面在服务器设置是可以进行系统更改的，不用太在意。



### 二.将远程主机配置成服务器

进入管理控制台

产品与服务->轻量应用服务器

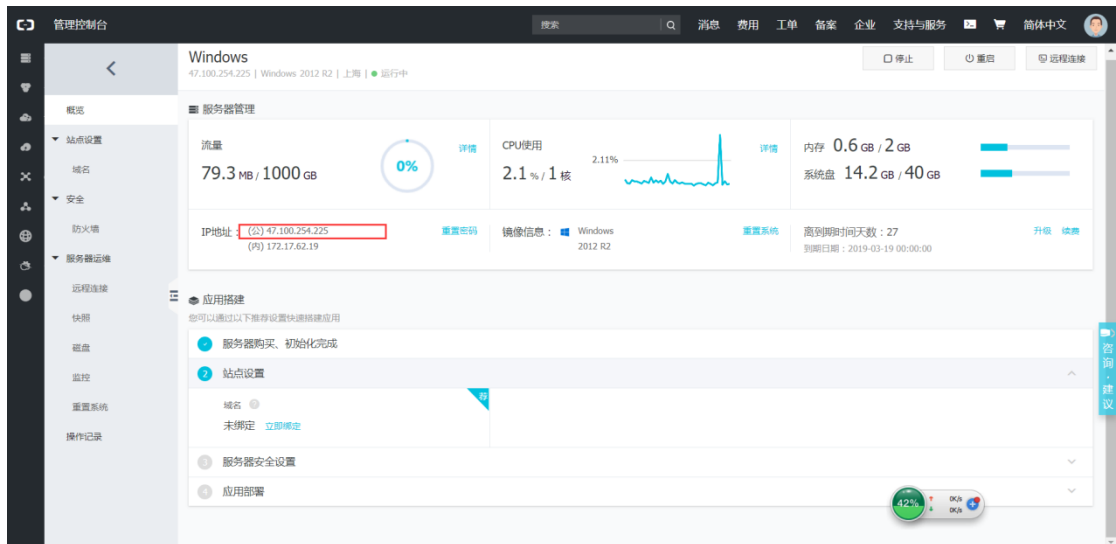


之后就可以看到我们的服务器实例，这里只有一个，是我刚刚购买的服务器

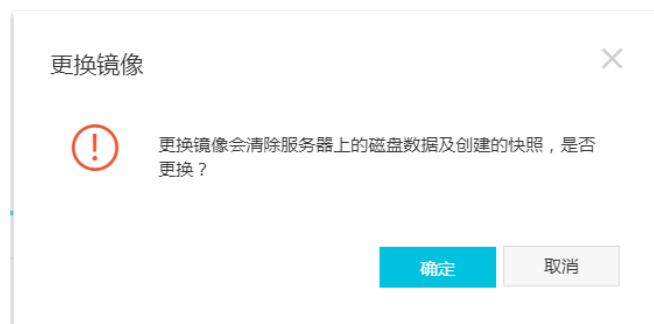
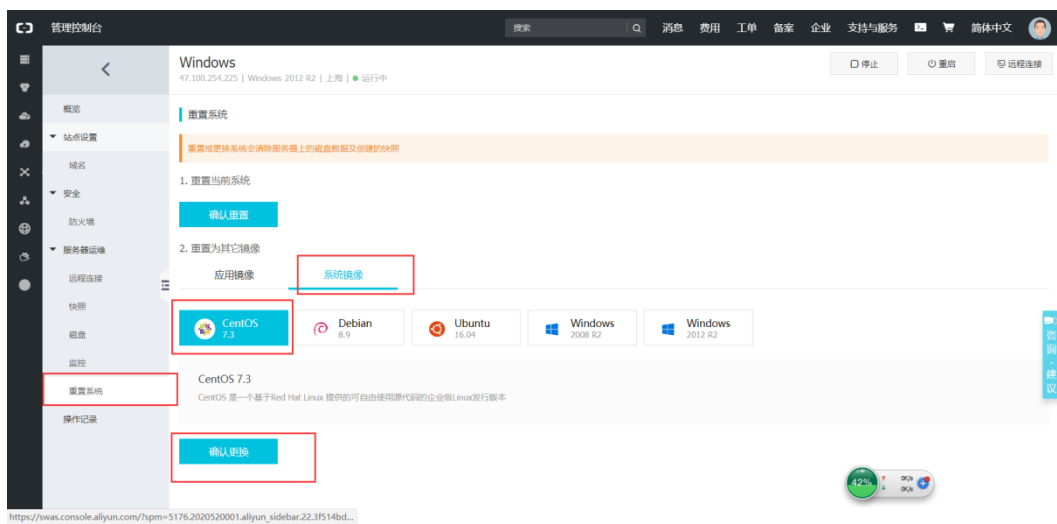


点击进入，可以看到我们当前服务器的各项参数

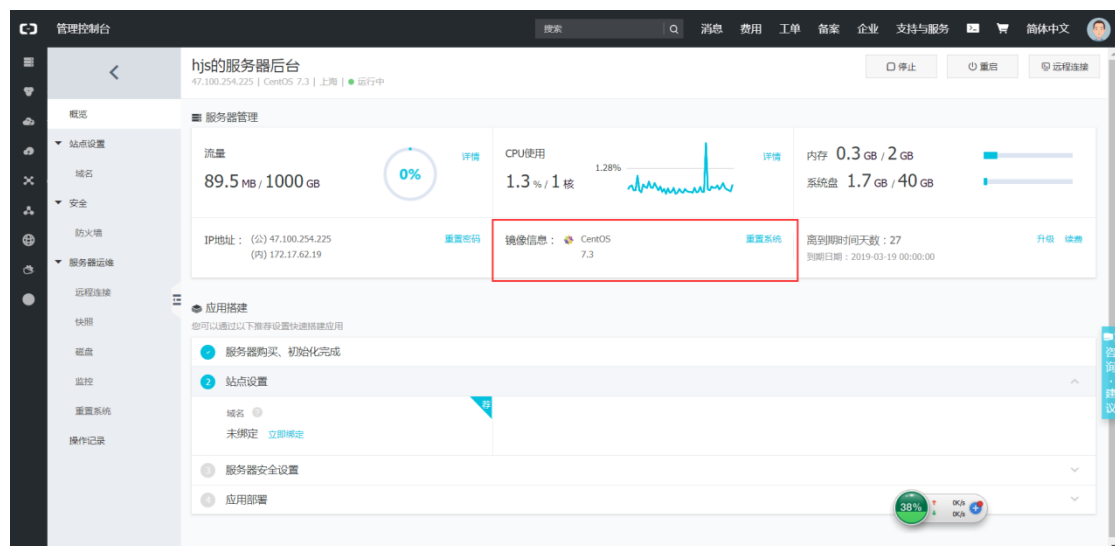
画红圈的部分是我们的服务器的公网 ip 地址，简单来说，就是别的用户要访问我的服务器，就必须通过这个 ip 地址来访问。这里记住它一下。



为了方便，我准备重置下系统镜像，换成 CentOS。但如果是想重置系统，也可以在如下的设置里进行重置，但注意如果原系统上已经有重要文件，重置系统和重置镜像，都会清空丢失！



重置后效果如下：



为什么要选择 linux 的系统镜像，而不是用 windows 的系统呢？其实普通用户来说，使用 windows 系统的话，操作和配置上会更加简单和直观，而 linux 多以命令行来执行操作，相对来说在配置和后期的操作上会显得更为麻烦一些。但是 linux 系统服务器的安全性会比 windows 更高，因为 windows 会更容易受到黑客的攻击而导致服务器崩溃。总而言之，如果想从事服务器端后台工作的开发者来说，还是选择 linux 系统更为合适。

接下来就是一系列的配置流程了：

接下来安装 Xshell 和 Xftp

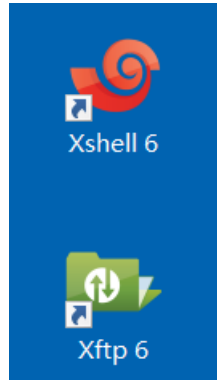
**Xshell：**控制远端云服务器主机的软件

即 Xshell 能在我们的 windows 界面下访问远端不同搭载在不同操作系统上的服务器。简单地说就是我们可以透过这款软件来控制我们的云服务器 ECS 或轻量型应用云服务器。

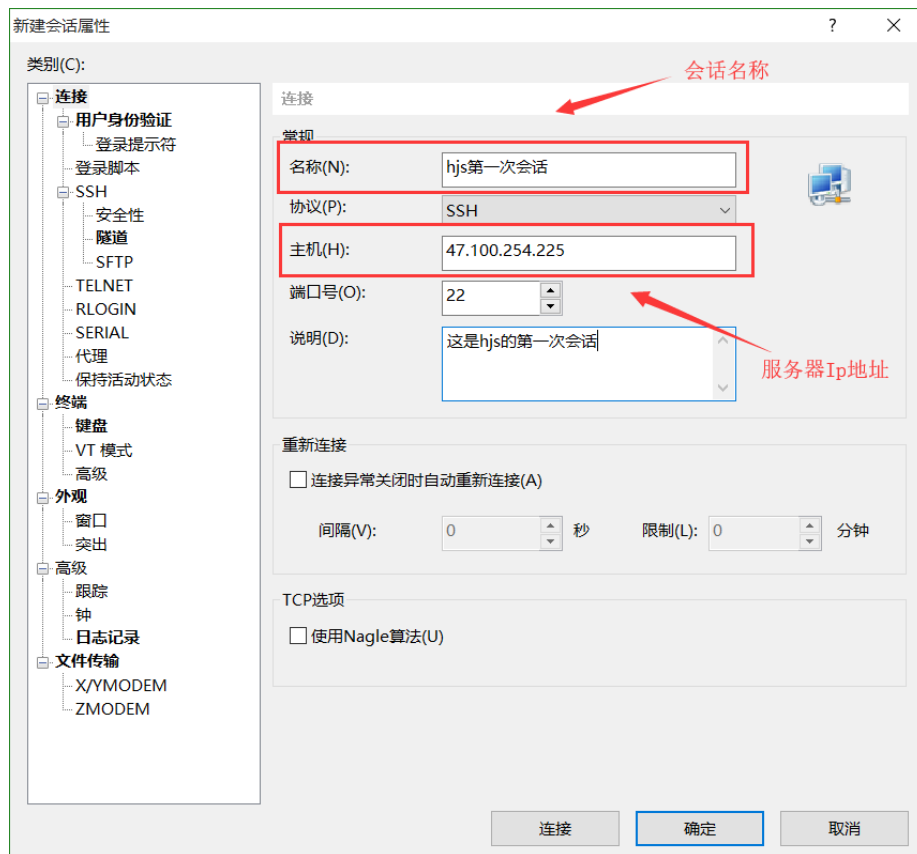
**xftp：**能在当前电脑和远端云服务器之间传输文件的软件

一款功能强大的基于 windows 平台的 SFTP、FTP 文件传输软件。简单地说通过这款软件我们能在当前主机和远端云服务器主机之间传输文件。

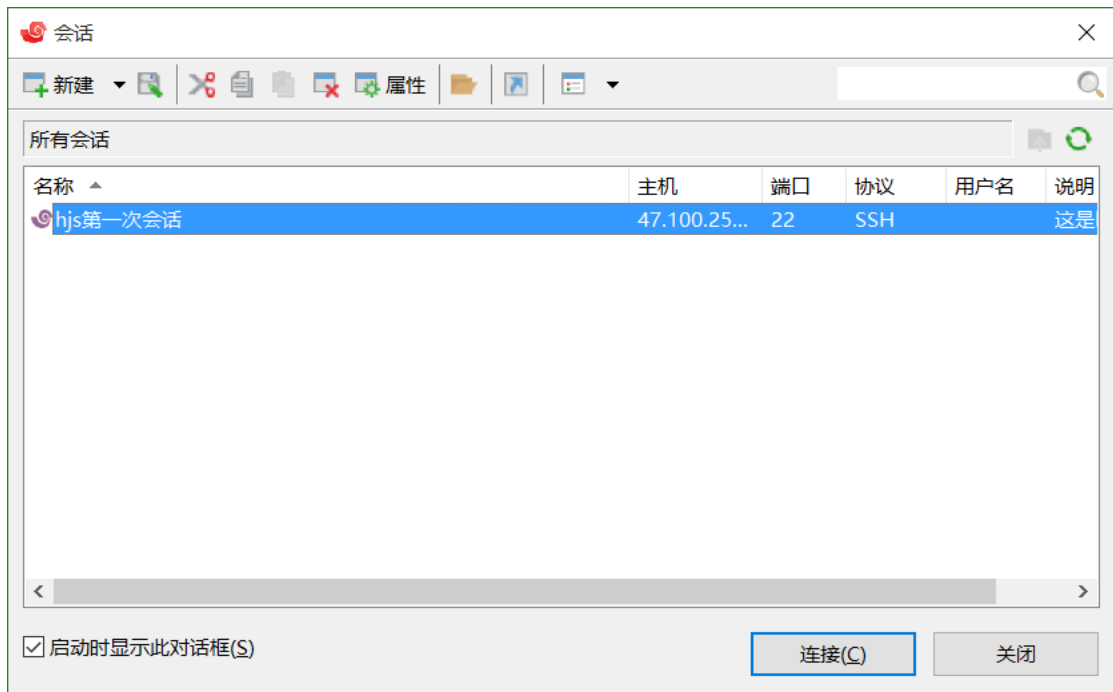
这两款软件如果用作学校或家庭用途，为免费。如果用作商业用途，就要收费。具体差别我没有仔细去研究。可以直接从官网下载，也可以从我的网盘链接里下载。



(1) 运行 Xshell，新建会话



点击确定建立会话，选中建立的会话，点击连接

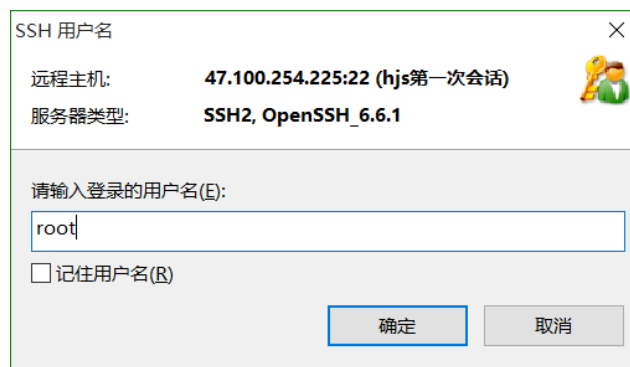


弹出如下对话框，点击接受并保存：

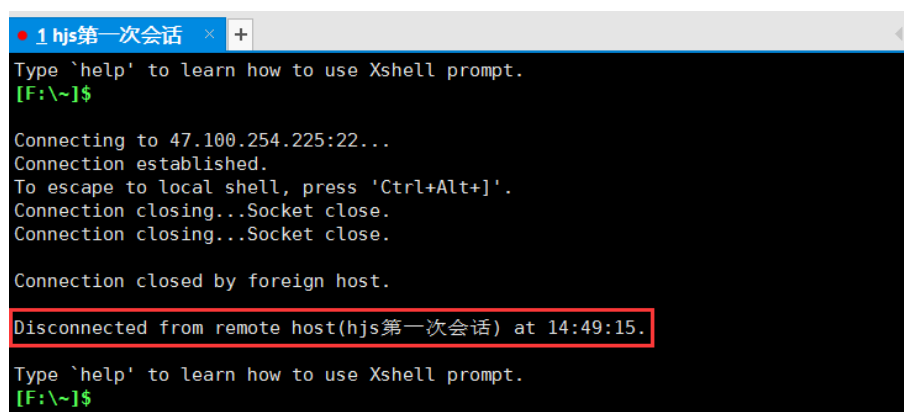


弹出如下对话框，输入登录用户名 **root**，记住用户名，再点击确定

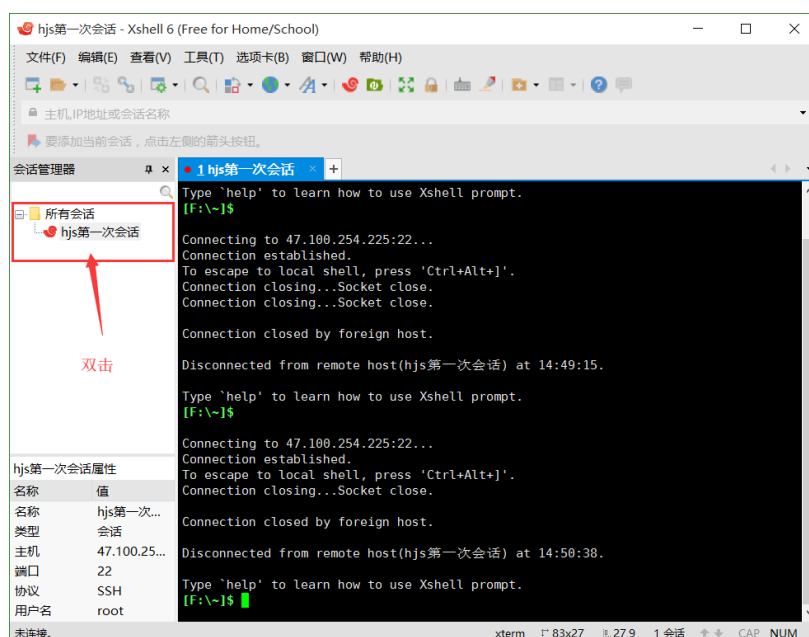




此时命令框显示连接断开??



这是因为我们还没有输入密码。。。



输入密码，再点击确定

SSH用户身份验证

远程主机: 47.100.254.225:22 (hjs第一次会话)

登录名: root

服务器类型: SSH2, OpenSSH\_6.6.1

请在下面选择恰当的身份验证方法并提供登录所需的信息。

☒ Password(P)

密码(W): .....

☐ Public Key(U)

用户密钥(K): ..... 浏览(B)...

密码(H): .....

☐ Keyboard Interactive(I)

使用键盘输入用户身份验证。

☒ 记住密码(R)

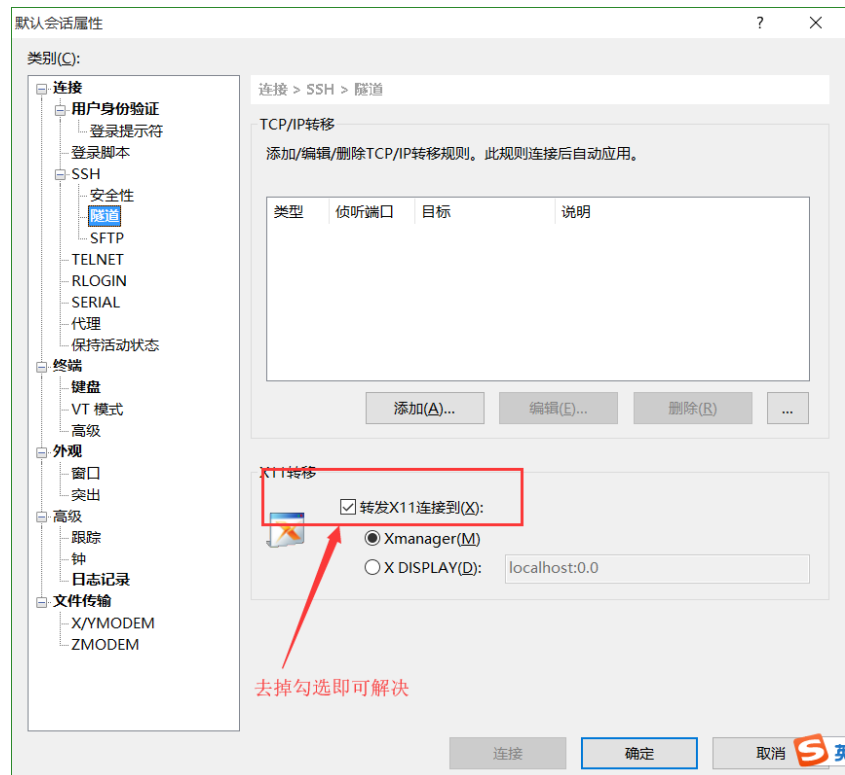
确定 取消

此时对话框弹出连接建立成功信息。此时我们当前主机与远端的阿里云服务器建立成功连接。但是也弹出了警告：WARNING! The remote SSH server rejected X11 forwarding request.

```
hjs第一次会话 - root@iz01762evc6i3az:~ - Xshell 6 (Free for Home/School)
文件(F) 编辑(E) 查看(V) 工具(T) 选项卡(B) 窗口(W) 帮助(H)
ssh://root@47.100.254.225:22
要添加当前会话，点击左侧的箭头按钮。
会话管理器
  x 1 hjs第一次会话 x +
  所有会话
    hjs第一次会话
hjs第一次会话属性
名称 值
名称 hjs第一次...
类型 会话
主机 47.100.25...
端口 22
协议 SSH
用户名 root
Disconnected from remote host(hjs第一次会话) at 14:49:15.
Type 'help' to learn how to use Xshell prompt.
[F:\~]$
Connecting to 47.100.254.225:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+J'.
Connection closing...Socket close.
Connection closed by foreign host.
Disconnected from remote host(hjs第一次会话) at 14:50:38.
Type 'help' to learn how to use Xshell prompt.
[F:\~]$
Connecting to 47.100.254.225:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+J'.
WARNING! The remote SSH server rejected X11 forwarding request.
Last login: Tue Feb 19 13:41:39 2019
Welcome to Alibaba Cloud Elastic Compute Service !
[root@iz01762evc6i3az ~]#
```

解决方法:

左上角: 文件-属性-隧道



再双击会话重新建立连接, 解决。

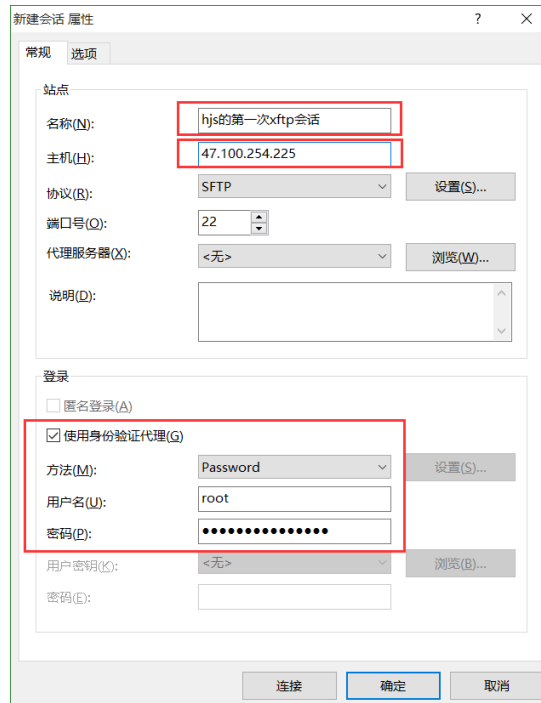
```
Connecting to 47.100.254.225:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+J'.

Last login: Tue Feb 19 14:56:57 2019 from 120.229.255.50

Welcome to Alibaba Cloud Elastic Compute Service !

[root@iz01762evc6i3az ~]#
```

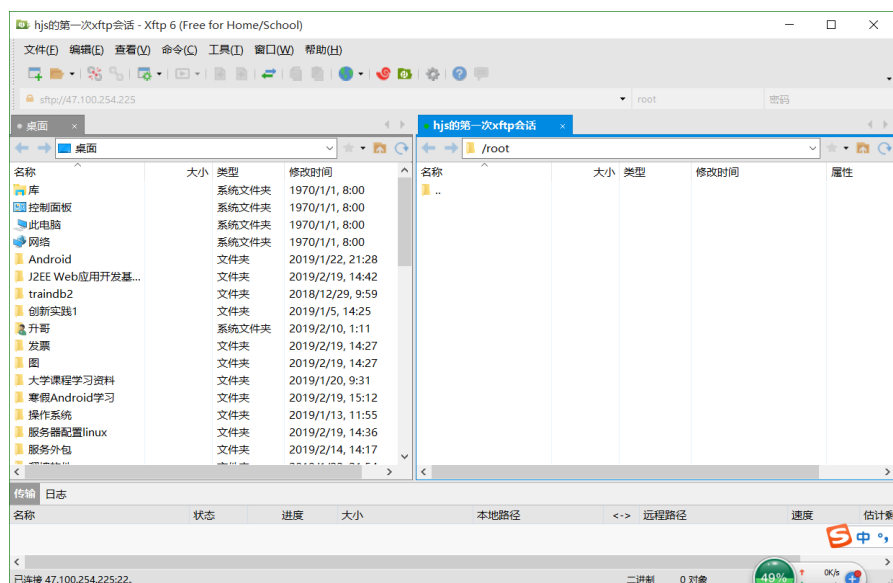
(2) 运行 Xftp, 按照类似的方式建立会话。



点击确定，Xftp 运行界面如下：

左边栏是我们当前主机的文件系统，这个显而易见

右边栏是我们远程服务器主机的 linux 文件系统，在右边点击返回键，你可以看到该系统下多个文件目录，我们后面会这里创建文件夹，来存放我们传输过去的 jdk 等安装文件。我们可以在这里手动之间创键，也可以在阿里云的服务器后台里通过建立连接后，以 linux 命令方式来创建。后面只介绍 linux 命令的创建的创建、安装和配置方式



还有一种方法建立 Xftp 会话：

直接在 Xshell 左上栏：窗口->传输新建文件 来免密打开并建立 Xftp 会话。

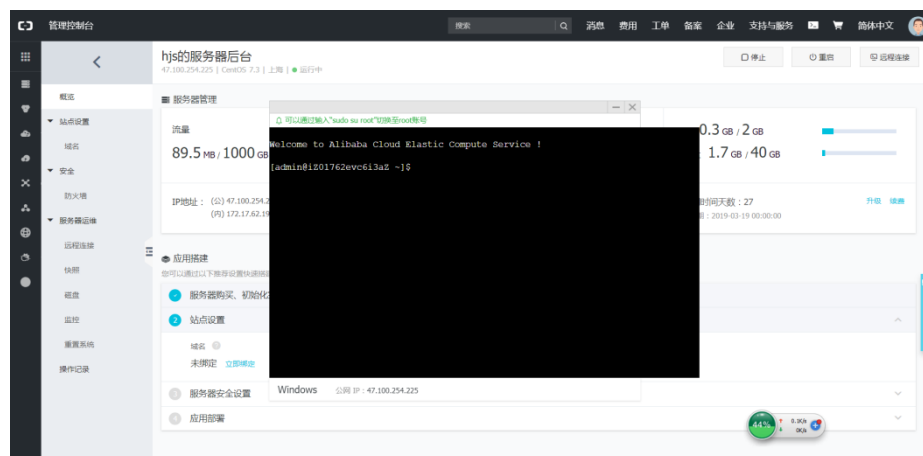
接下来就是部署我们 java 项目的准备工作了

在这里说一下，在云服务器上部署 Java 项目，一共有三种方式，分别为 Java 镜像部署、一键安装包部署，以及手动部署。前两种方式的特点是非常方便和快捷，但是要需要花比较高的价钱来购买服务。手动部署就是我们自己来进行人工操作了，就是很费事。

接下来还是介绍下手动部署：

### （1）主机下载 linux 版本的 jdk 和 tomcat

接下来点击远程连接，进入 linux 命令行界面，开始我们的配置工作：



首先，用 `sudo su` 命令获得管理员权限，因为接下来会涉及到系统环境配置等管理员权限的操作。

我们先在 linux 系统下的 `usr` 中新建文件夹 `java` 来存放我们的 `jdk` 和 `tomcat`

```
可以通过输入“sudo su root”切换至root账号
Last login: Tue Feb 19 16:11:24 2019 from 47.102.58.80

Welcome to Alibaba Cloud Elastic Compute Service !

[admin@iz01762evc6i3az ~]$ sudo su
[root@iz01762evc6i3az admin]# cd /usr
[root@iz01762evc6i3az usr]# mkdir java
[root@iz01762evc6i3az usr]# ls
bin  games  java  lib64  local  share  tmp
etc  include lib  libexec sbin  src
[root@iz01762evc6i3az usr]# tar -
```

创建成功。

在 `xftp` 中将下载好的 `jdk` 传到该目录中

hjs的第一次xftp会话				
/usr/java				
名称	大小	类型	修改时间	属性
..				
jdk-8u144-linux-x64...	176.92MB	WinRAR...	2019/2/19, 16:21	-rw-r--r--

回到阿里云后台命令行界面，检查到传输成功了。

```
[root@iz01762evc6i3az usr]# cd /usr/java
[root@iz01762evc6i3az java]# ls
jdk-8u144-linux-x64.tar.gz
[root@iz01762evc6i3az java]#
```

同理把 apache 压缩包也传输进这个文件夹里。

```
[root@iz01762evc6i3az java]# ls
apache-tomcat-8.5.38.tar.gz  jdk-8u144-linux-x64.tar.gz
```

解压：

```
[root@iz01762evc6i3az java]# tar xzvf jdk-8u144-linux-x64.tar.gz
[root@iz01762evc6i3az java]# tar -xzvf apache-tomcat-8.5.38.tar.gz
```

```
[root@iz01762evc6i3az java]# ls
apache-tomcat-8.5.38      jdk1.8.0_144
apache-tomcat-8.5.38.tar.gz  jdk-8u144-linux-x64.tar.gz
```

我们在这里修改一下文件名，方便后面配置时可以少写点命令，毕竟 linux 就是傻瓜式的用命令行来操作的。最后显示下当前文件项目，可看到更改后的显示如下所示。

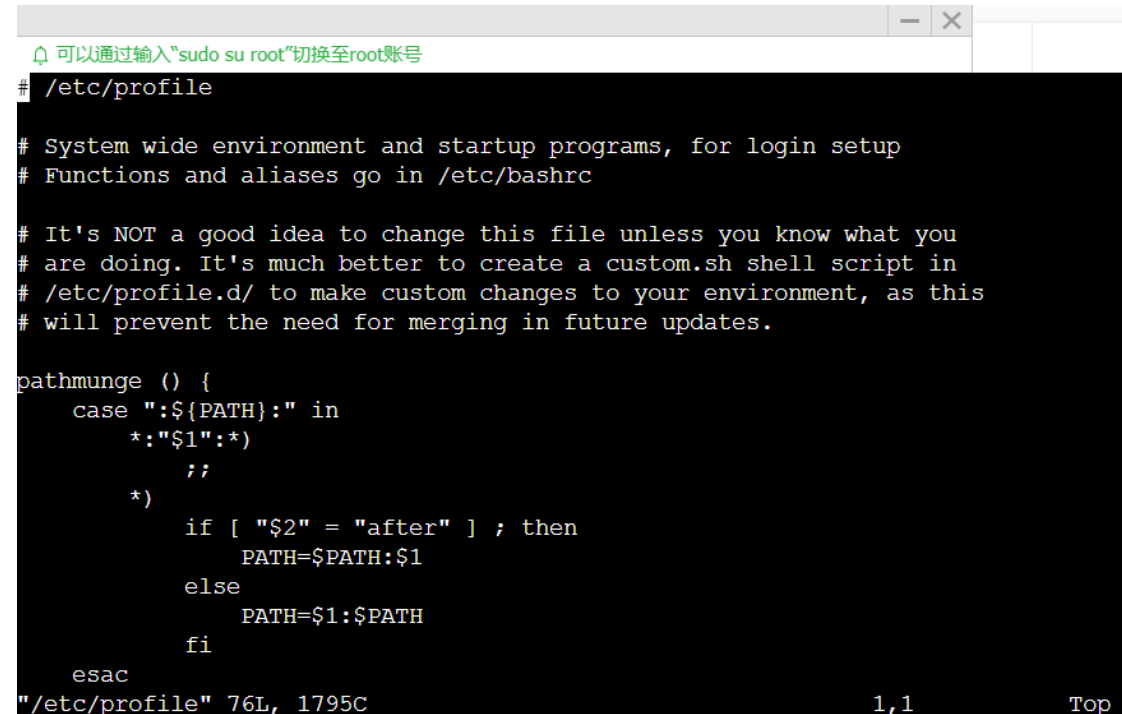
```
[root@iz01762evc6i3az java]# mv jdk1.8.0_144 jdk8
[root@iz01762evc6i3az java]# mv apache-tomcat-8.5.38 tomcat8.5
[root@iz01762evc6i3az java]# ls
apache-tomcat-8.5.38.tar.gz  jdk8  jdk-8u144-linux-x64.tar.gz  tomcat8.5
```

接下来配置环境变量

输入:

```
[root@iz01762evc6i3az java]# vim etc/profile
```

显示效果如下:



```
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case ":${PATH}:" in
        *: "$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
                PATH=$PATH:$1
            else
                PATH=$1:$PATH
            fi
    esac
}

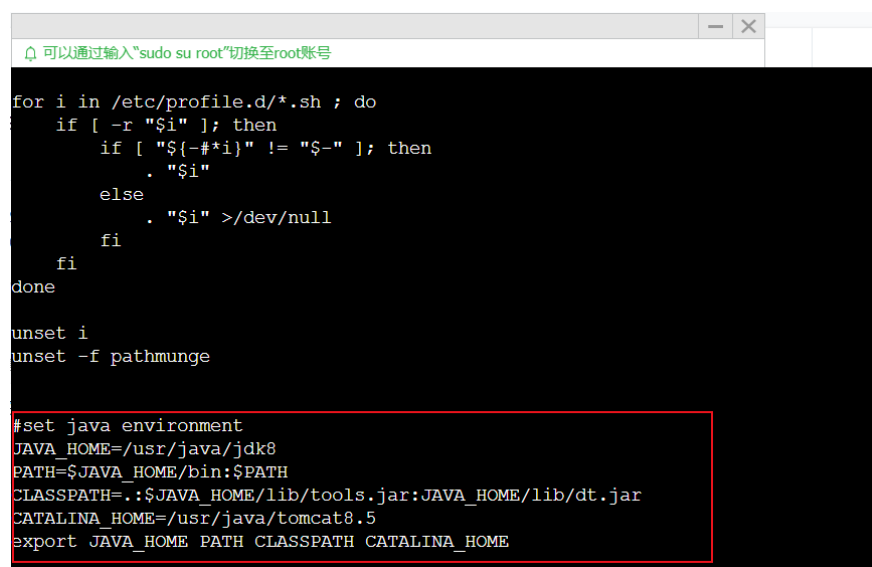
"/etc/profile" 76L, 1795C 1,1 Top
```

按 i 键进入编辑状态



```
-- INSERT -- 1,1 Top
```

注意现在开始不要出现误删或误加的操作, 这里是系统环境变量的配置, 会影响系统其它位置的运行。一直跳到文本的最末尾, 配置 JDK 和 tomcat 环境如下:



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

unset i
unset -f pathmunge

#set java environment
JAVA_HOME=/usr/java/jdk8
PATH=$JAVA_HOME/bin:$PATH
CLASSPATH=.:$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/dt.jar
CATALINA_HOME=/usr/java/tomcat8.5
export JAVA_HOME PATH CLASSPATH CATALINA_HOME
```

按 `esc` 键退出编辑状态, `:wq` 保存并退出  
显示效果如下:

```
unset i
unset -f pathmunge

#set java environment
JAVA_HOME=/usr/java/jdk8
PATH=$JAVA_HOME/bin:$PATH
CLASSPATH=.:$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/dt.jar

CATALINA_HOME=/usr/java/tomcat8.5
export JAVA_HOME PATH CLASSPATH CATALINA_HOME
"/etc/profile" 85L, 2009C written
[root@iz01762evc6i3az java]#
```

启用配置:

```
[root@iz01762evc6i3az java]# source /etc/profile
```

输入 `javac` 检验:

```
ers
-d <directory>          Specify where to place generated class files
-s <directory>          Specify where to place generated source files
-h <directory>          Specify where to place generated native header files
es
-implicit:{none,class}  Specify whether or not to generate class files for
implicitly referenced files
-encoding <encoding>    Specify character encoding used by source files
-source <release>       Provide source compatibility with specified release
e
-target <release>       Generate class files for specific VM version
-profile <profile>       Check that API used is available in the specified
profile
-version                Version information
-help                   Print a synopsis of standard options
-Akey[=value]           Options to pass to annotation processors
-X                      Print a synopsis of nonstandard options
-J<flag>                Pass <flag> directly to the runtime system
-Werror                 Terminate compilation if warnings occur
@<filename>            Read options and filenames from file
[root@iz01762evc6i3az java]#
```

进入 `tomcat` 下的 `bin` 目录执行启动脚本, 启动 `tomcat`

```
[root@iz01762evc6i3az java]# cd /usr/java/tomcat8.5/bin
[root@iz01762evc6i3az bin]# ./startup.sh
```

注: `./shutdown.sh` 为关闭脚本命令。可关闭 `tomcat`  
效果如下:



```
[root@iz01762evc6i3az java]# cd /usr/java/tomcat8.5/bin
[root@iz01762evc6i3az bin]# ./startup.sh
Using CATALINA_BASE:   /usr/java/tomcat8.5
Using CATALINA_HOME:   /usr/java/tomcat8.5
Using CATALINA_TMPDIR: /usr/java/tomcat8.5/temp
Using JRE_HOME:        /usr/java/jdk8
Using CLASSPATH:        /usr/java/tomcat8.5/bin/bootstrap.jar:/usr/java/tomcat8.5/bin/tomcat-juli.jar
Tomcat started.
[root@iz01762evc6i3az bin]#
```

本来以为大功告成，访问 <http://47.100.254.225:8080>，发现还是访问不了...  
研究了好一会，发现还需要阿里云添加安全组规则

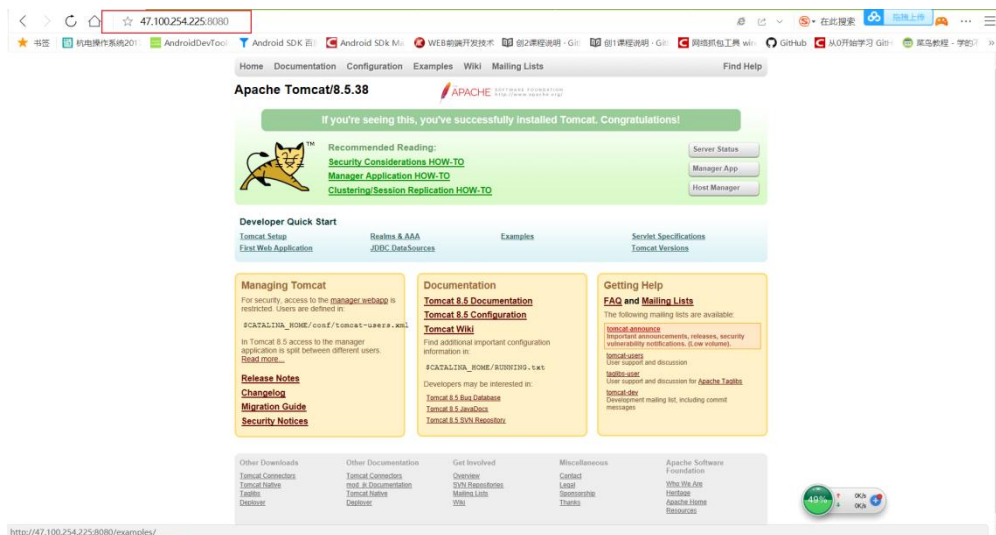
1. 进入控制台
2. 选择 轻量应用服务器
3. 在服务器列表中选择需要配置的服务器
4. 在左侧 安全 中选择 防火墙
5. 在右侧选择 添加规则
6. 在 应用类型、协议、端口范围 三项中，分别填 自定义、TCP、8080

添加防火墙规则

应用类型	协议	端口范围	操作
自定义	TCP	8080	
+添加规则			

确定

现在可以访问了。



接下来是对数据库的下载和配置

在 centos7 以上的版本系统中，默认搭载的数据库从 MySQL 改变为 Mariadb。这个数据库没有用过，也不知道怎么样，这里我把它删了，换成我熟悉的 MySQL。

卸载 Mariadb

```
[root@iz01762evc6i3az admin]# rpm -qa|grep mariadb
mariadb-libs-5.5.52-1.el7.x86_64
[root@iz01762evc6i3az admin]# rpm -e --nodeps mariadb-libs-5.5.52-1.el7.x86_64
```

在安装 MySQL 之前我们最好再排除一下系统有没有可能已经存在了 MySQL，如果存在了，是肯定安装不成功的，如果安装了，用 `# rpm -e mysql` 命令删除

```
[root@iz01762evc6i3az admin]# # rpm -qa | grep mysql
[root@iz01762evc6i3az admin]#
```

如果直接安装配置自己传过去的下载好的 MySQL tar 包，可能会出现比较多的问题，需要另外自己下载依赖以及修改配置等。这里为了方便直接采用 yum 安装

```
[root@iz01762evc6i3az admin]# rpm -Uvh http://dev.mysql.com/get/mysql-community-release-el7-5.noarch.rpm
```

```
[root@iz01762evc6i3az admin]# yum -y install mysql-community-server
```

开机自动启动：

```
[root@iz01762evc6i3az admin]# systemctl enable mysqld
```

启动 MySQL 服务进程

```
[root@iz01762evc6i3az admin]# systemctl start mysqld
```

配置 root 用户密码

```
[root@iz01762evc6i3az admin]# mysql_secure_installation
```

针对以下情况连续三次回车，出现 ERROR 提醒不要管，因为我们是第一次安装，没有设置过密码，因此默认为 blank。

```
Enter current password for root (enter for none):
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
```

第三次回车后，出现以下信息：

```
Enter current password for root (enter for none):  
OK, successfully used password, moving on...  
  
Setting the root password ensures that nobody can log into the MySQL  
root user without the proper authorisation.  
  
Set root password? [Y/n] Y
```

选 Y，表示我们给 root 用户设置密码

要注意，设置密码时它是设置隐藏没让你看到的...不要因为键盘失灵或卡死了。直接输入就是了。

```
New password:  
Re-enter new password:  
Password updated successfully!  
Reloading privilege tables..  
... Success!  
  
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? [Y/n]
```

接下来选择是否删除临时用户，选择是。  
然后选择是否不允许远程登录，选择否

```
Remove anonymous users? [Y/n] Y  
... 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? [Y/n] n
```

最后它问你是否要删除本来存在在 MySQL 的一个 test 数据库，该数据库是用于测试的，删  
不删取决于你。

```
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? [Y/n] n  
... skipping.  
  
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? [Y/n] n
```

最后同意刷新权限表，就此，数据库和配置结束了。

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

Reload privilege tables now? [Y/n] Y
... Success!

All done! If you've completed all of the above steps, your MySQL
installation should now be secure.

Thanks for using MySQL!

Cleaning up...
[root@iz01762evc6i3az admin]#
```

我们想要进入数据库执行 SQL 语句，可以输入一下命令，开启一个新的 shell 窗口，链接数据库。然后就可以进行 CURD 的命令了。

```
[root@iz01762evc6i3az admin]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.6.43 MySQL Community Server (GPL)

Copyright (c) 2000, 2019, 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>
```

一切准备就绪，最后就差把我的 javaweb 项目部署到服务器上了。

首先，在 IntelliJ Idea 中把我们的项目打包成 war 项目：

File—>

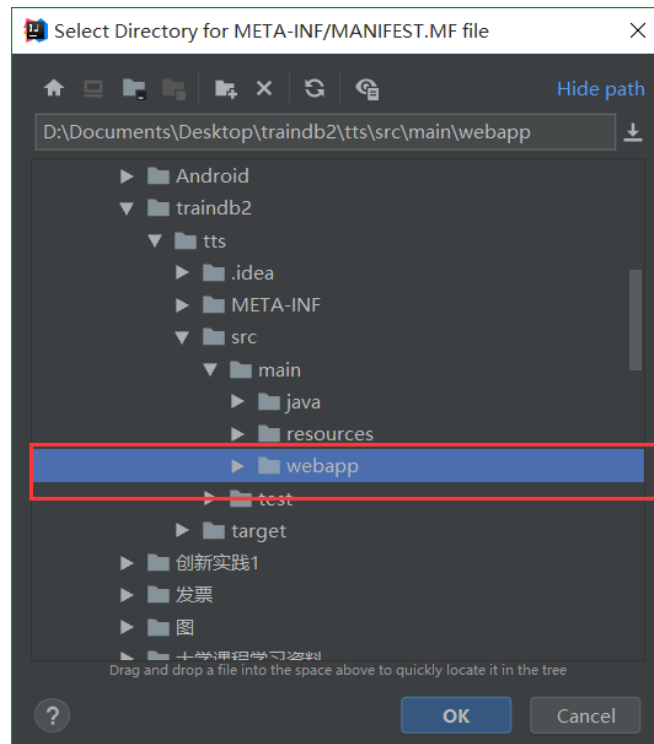
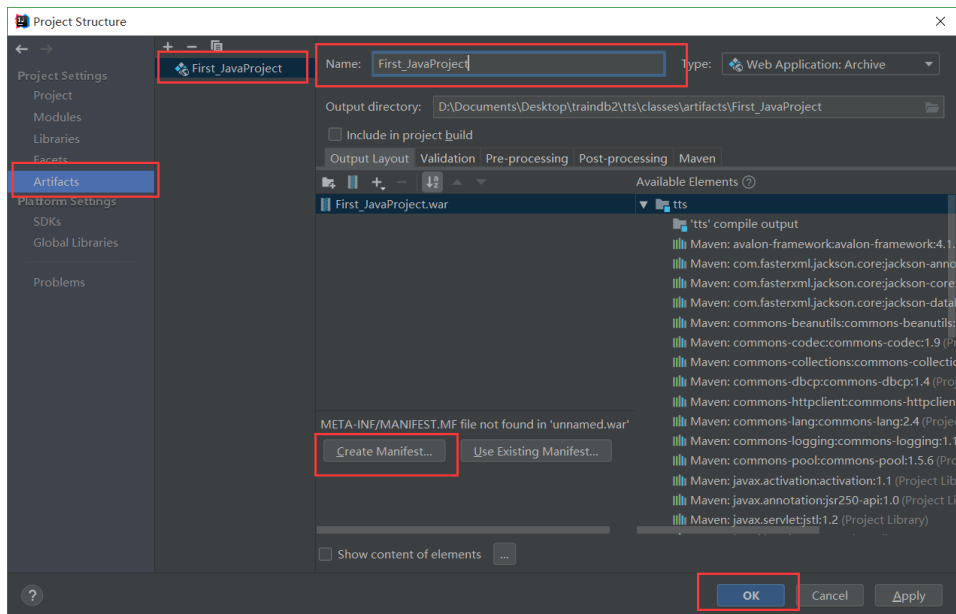
Project Structure—>

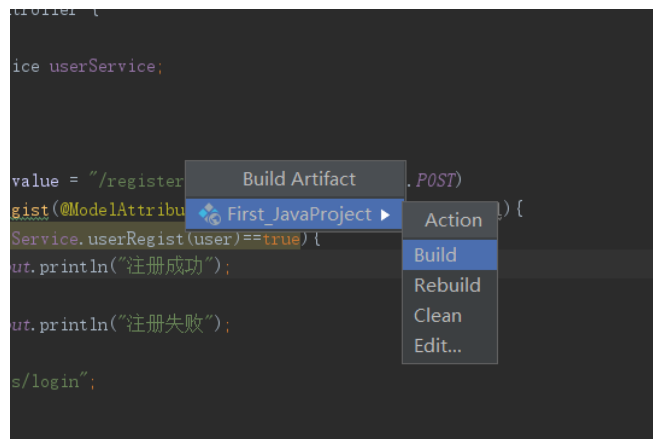
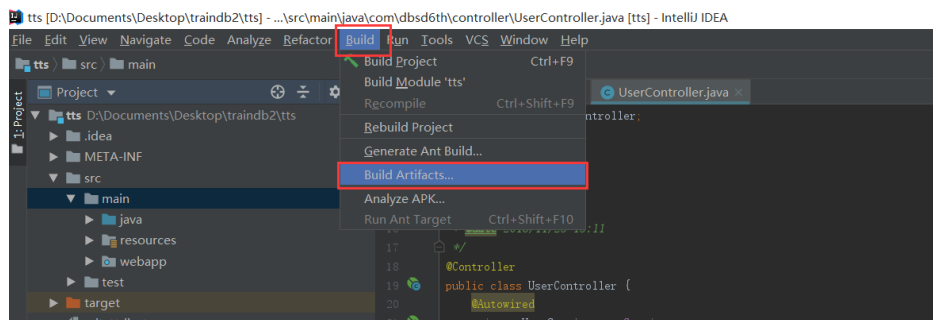
Artifacts—>

点击+号—>

Web Application:Archive

划红线部分为必须填写的部分





traindb2 > tts > classes > artifacts > First_JavaProject				搜索"First_JavaPr
名称	修改日期	类型	大小	
First_JavaProject.war	2019/2/20 0:46	WAR 文件	1 KB	

将生成 war 文件发送服务器主机的 Tomcat 目录下的 webapps 文件夹下。  
 重启 Tomcat，就可以访问我们的项目。当然还有我们项目对应的 sql 文件也要导入到服务器端的 mysql 来执行。这里就先略了。

(2) 进一步利用阿里云进行个人学习博客的开发环境搭建

### 一、安装 node.js

#新建目录 www

(1)mkdir /www

(2)cd /www

(3) wget <https://npm.taobao.org/mirrors/node/v8.2.1/node-v8.2.1-linux-x64.tar.xz>

(4) tar -xvf node-v8.2.1-linux-x64.tar.xz

#进入解压目录下的 bin 目录，执行 ls 命令

(4)cd node-v8.2.1-linux-x64/bin && ls

```
[root@iz01762evc6i3az www]# cd node-v8.2.1-linux-x64/bin
[root@iz01762evc6i3az bin]# ls
node  npm  npx
```

#测试是否安装成功:

(5) ./node -v

```
[root@iz01762evc6i3az bin]# ./node -v
v8.2.1
```

(6) ./npm -v

```
[root@iz01762evc6i3az bin]# ./npm -v
5.3.0
```

#现在 node 和 npm 还不能全局使用，做个全局软链接

(7)ln -s /www/node-v8.2.1-linux-x64/bin/node /usr/local/bin/node

(8)ln -s /www/node-v8.2.1-linux-x64/bin/npm /usr/local/bin/npm

### 二、安装 nginx

(1)yum -y install nginx

#测试:

(2)nginx -v

```
Complete!
[root@iz01762evc6i3az bin]# nginx -v
nginx version: nginx/1.12.2
```

(3) vi /etc/yum.repos.d/nginx.repo

[nginx]

name=nginx repo

baseurl=http://nginx.org/packages/centos/\$releasever/\$basearch/

gpgcheck=0

enabled=1

(4) systemctl start nginx # 启动 Nginx

chkconfig nginx on # 设置开机启动 Nginx

(5) vi /etc/nginx/conf.d/ghost.conf

server {

listen 80;

server\_name 47.100.254.225; #这里改为你的域名或 ip

```

        location / {
            proxy_set_header    X-Real-IP $remote_addr;
            proxy_set_header    Host      $http_host;
            proxy_pass            http://127.0.0.1:2368;
        }
    }
}

```

保存退出

(6)systemctl start nginx#重启 nginx

最后进行域名访问:



### 三、安装 MySQL

#下载 mysql 源安装包

(1) wget <http://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm>

#安装 mysql 源

(2) yum localinstall mysql57-community-release-el7-8.noarch.rpm

#检查 mysql 源是否安装成功

(3) yum repolist enabled | grep "mysql.\*-community.\*"

#安装 MySQL

(3) yum install mysql-community-server

#启动 MySQL 服务

(4)systemctl start mysqld

#查看 MySQL 的启动状态

(5)systemctl status mysqld

#开机启动

(6) systemctl enable mysqld

(7) 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

(8) grep 'temporary password' /var/log/mysqld.log

#登陆并修改默认密码

(10) mysql -u root -p

(11) mysql>ALTER USER 'root'@'localhost' IDENTIFIED BY '新密码!';

我的密码是:557523

#一些常规设置

chkconfig mysqld on # 设置开机启动 Mysql

anonymous users? [Y/n] # 删除匿名用户

Disallow root login remotely? [Y/n] # 禁止 root 用户远程登录

Remove test database and access to it? [Y/n] #删除默认的 test 数据库

Reload privilege tables now? [Y/n] # 刷新授权表使修改生效

#为避免数据库存放的中文数据变成乱码，可以进行 MySQL 编码设置

#退出 mysql

(12)mysql>exit

(13) vi /etc/my.cnf

(14)按 i 进入编辑模式，移动上下左右键

[client]

default-character-set=utf8

[mysql]

default-character-set=utf8

[mysqld]

character-set-server=utf8

collation-server=utf8\_general\_ci

(15)esc 键退出编辑模式

(16):wq 保存数据

(17)

mysql -u root -p

mysql>create database ghost;

mysql>grant all privileges on ghost.\* to 'hjs'@'%' identified by '557523';

#重新读取权限表中的数据到内存，不用重启 mysql 就可以让权限生效

mysql>flush privileges

grant all privileges on 库名.表名 to '用户名'@'IP 地址' identified by '密码' with grant option;flush privileges;

库名:要远程访问的[数据库](#)名称,所有的数据库使用 “\*”

表名:要远程访问的数据库下的表的名称，所有的表使用 “\*”

用户名:要赋给远程访问权限的用户名称

IP 地址:可以远程访问的电脑的 IP 地址，所有的地址使用 “%”

密码:要赋给远程访问权限的用户对应使用的密码

```
mysql> grant all privileges on ghost.* to 'hjs'@'%' identified by '557523';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

#### 四、安装 Ghost

(1)#mkdir -p /var/www/ghost

(2)#cd /var/www/ghost

(2)#curl -L <https://ghost.org/zip/ghost-latest.zip> -o ghost.zip

(3)#yum install unzip -y

(4)#unzip ghost.zip

(5)#npm install -production

该命令可能会出现权限问题，这里出错了很久，不知道为什么一直没有访问权限来读写一些文件。

改为: #npm install -production --unsafe -perm 即可解决

(6)#npm start

出现红色框内容表示 ghost 安装成功

```
[2019-03-25 08:38:10] INFO Ghost is running in development...
[2019-03-25 08:38:10] INFO Listening on: 127.0.0.1:2368
[2019-03-25 08:38:10] INFO Url configured as: http://localhost:2368/
[2019-03-25 08:38:10] INFO Ctrl+C to shut down
[2019-03-25 08:38:10] INFO Ghost boot 6.282s
```

(7)许多教程都说接下来在 ghost 工程目录下寻找 config.example.js 模板来配置自己的 config.js 文件。然而在现在许多版本在 ghost 目录下根本找不到这个配置文件。研究了很久，才知道应该进行如下操作：

(8)#vim /var/www/ghost/core/server/config/env/config.development.json

```
{
  "url": "http://47.100.254.225:2368",
  "database": {
    "client": "mysql",
    "connection": {
      "host": "127.0.0.1",
      "user": "hjs",
      "password": "557523",
      "database": "ghost",
      "charset": "utf-8"
    },
    "debug": false
  },
  "paths": {
    "contentPath": "content/"
  },
  "privacy": {
    "useRpcPing": false,
```

```

        "useUpdateCheck": true
    },
    "useMinFiles": false,
    "caching": {
        "theme": {
            "maxAge": 0
        },
        "admin": {
            "maxAge": 0
        }
    }
}

```

(9)# vim /var/www/ghost/core/server/config/env/config.production.json

```

{
    "database": {
        "client": "mysql",
        "connection": {
            "host"      : "127.0.0.1",
            "user"      : "hjs",
            "password" : "557523",
            "database" : "ghost",
            "charset"  : "utf-8"
        }
    },
    "paths": {
        "contentPath": "content/"
    },
    "logging": {
        "level": "info",
        "rotation": {
            "enabled": true
        },
        "transports": ["file", "stdout"]
    }
}

```

(10)#npm start --production

此时 ghost 后台网站成功启动挂起，但是此时 ghost 占据着我们 linux 的命令行前台界面，导致我们无法进行其他命令操作，所以我们要使用进程管理器来配置 Ghost 永远运行。

#npm install pm2 -g

```
[root@iz01762evc6i3az qhost]# npm install pm2 -g
/wwww/node-v8.2.1-linux-x64/bin/pnpm2 -> /www/node-v8.2.1-linux-x64/lib/node_modules/pnpm2/bin/pnm2
/wwww/node-v8.2.1-linux-x64/bin/pnm2-dev -> /www/node-v8.2.1-linux-x64/lib/node_modules/pnm2/bin/pnm2-dev
/wwww/node-v8.2.1-linux-x64/bin/pnm2-docker -> /www/node-v8.2.1-linux-x64/lib/node_modules/pnm2/bin/pnm2-docker
/wwww/node-v8.2.1-linux-x64/bin/pnm2-runtime -> /www/node-v8.2.1-linux-x64/lib/node_modules/pnm2/bin/pnm2-runtime
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.7 (node_modules/pnm2/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.7: wanted {"os": "darwin", "arch": "any"} (cu
rrent: {"os": "linux", "arch": "x64"})

+ pm2@3.4.0
added 378 packages in 33.501s
```

```
# NODE_ENV=production pm2 start index.js --name "ghost"
```

问题: bash: pm2:command not found

说明没有配置到全局，因为 linux 不像 windows 一样能够自动创建一个环境变量去实现全局使用。

因此我们要创建一个 linux 下的软连接

```
# echo $PATH
```

```
[root@iz01762evc6i3az ghost]# echo $PATH
/usr/java/jdk8/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
```

我们可以选择任何一个以:隔开的路径作为系统环境路径

```
# ln -s /www/node-v8.2.1-linux-x64/bin/pm2 /usr/local/bin/
```

此时,若出现:

```
[root@iz01762evc6i3az ghost]# ln -s /www/node-v8.2.1-linux-x64/bin/pm2 /usr/local/bin/
ln: failed to create symbolic link '/usr/local/bin/pm2': File exists
```

```
# mv /usr/local/bin/pm2 /tmp/
```

再执行:

```
# ln -s /www/node-v8.2.1-linux-x64/bin/pm2 /usr/local/bin/
```

验证:

```
# pm2 list
```

```

      _____
     /          \
    /             \
   /               \
  /                 \
 /                   \
/                     \
\                     /
 \                   /
  \                 /
   \               /
    \             /
     \           /
      \         /
       \       /
        \     /
         \___/

Runtime Edition

PM2 is a Production Process Manager for Node.js applications
with a built-in Load Balancer.


Start and Daemonize any application:
$ pm2 start app.js


Load Balance 4 instances of api.js:
$ pm2 start api.js -i 4


Monitor in production:
$ pm2 monitor


Make pm2 auto-boot at server restart:
$ pm2 startup


To go further checkout:
http://pm2.io/


-----

[PM2] Spawning PM2 daemon with pm2_home=/root/.pm2
[PM2] PM2 Successfully daemonized
```

App name	id	version	mode	pid	status	restart	uptime	cpu	mem	user	watching
----------	----	---------	------	-----	--------	---------	--------	-----	-----	------	----------

```
Use "pm2 show <id|names>" to get more details about an app
```

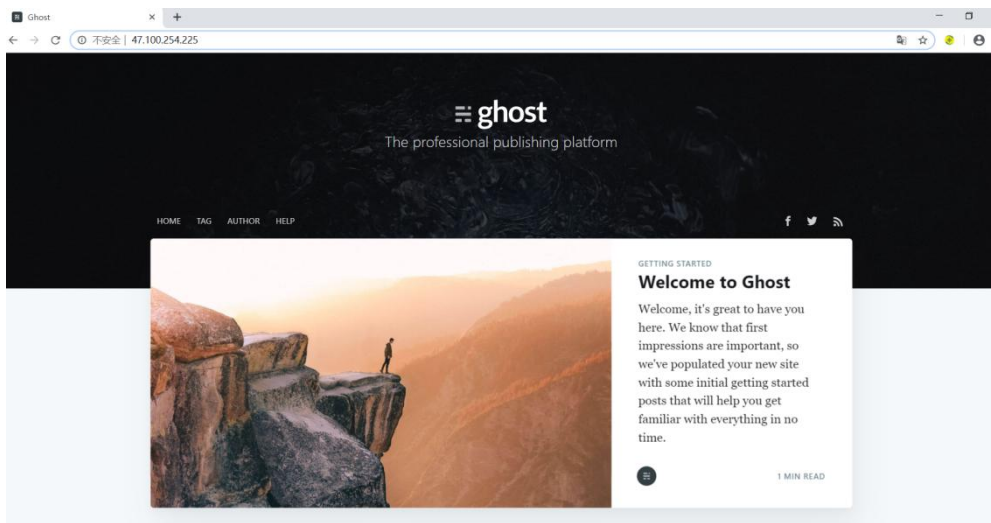
```
# NODE ENV=production pm2 start index.js --name "ghost"
```

```
[root@iz01762evc6i3az ghost]# NODE_ENV=production pm2 start index.js --name "ghost"
[PM2] Starting /var/www/ghost/index.js in fork_mode (1 instance)
[PM2] Done.
```

App name	id	version	mode	pid	status	restart	uptime	cpu	mem	user	watching
ghost	0	2.18.3	fork	29253	online	0	0s	0%	8.2 MB	root	disabled

Use `pm2 show <id|name>` to get more details about an app

最后使用域名登录:



以后开启/停止/重启 ghost

# pm2 start ghost

# pm2 stop ghost

# pm2 restart ghost

阿里云 nginx 配置 SSL 实现 https 访问

1.域名注册:

(1)在阿里云控制台→域名与网站（万网）→域名→信息模板  
注册并成功实名认证一个你个人的信息模板

信息模板管理							创建新信息模板
提示：域名信息模板可用于域名注册、域名持有者变更等，请填写真实、准确、完整的域名持有者信息！ 根据ICANN政策要求，域名注册人Email必须真实准确，为了方便日后管理使用，强烈建议您及时对信息模板中的Email邮箱进行验证。 <a href="#">了解更多</a>							
域名持有者类型：全部	域名审核状态：全部	持有者：	搜索				
持有者名称（中文）	注册人邮箱	邮箱验证	持有者名称（英文）	持有者类型	实名认证状态①	操作	
黄继升	326464623@qq.com	验证成功	huang ji sheng	个人	实名认证成功	<a href="#">默认模板</a>   <a href="#">删除</a>   <a href="#">查看</a>	

## (2) 阿里云查询可用域名

注册域名请您先查询

huangjisheng

.top ▼

查域名

☒ huangjisheng.top 可注册， 1元/首年

☐ huangjisheng.work 可注册， 1元/首年

☐ huangjisheng.store 可注册， 1元/首年

去结算

## (2) 结算

确认订单

确认订单

支付

支付成功

产品名称	产品内容	产品年限	优惠	价格	操作
.top 域名	huangjisheng.top .top域名首购特价1元，还送250G邮箱 (限普通用户注册)	1年	省0元	¥ 1.00	

您的域名持有者为：☒ 个人 ☐ 企业

如何选择？

以下为您曾使用过的域名信息，您可以直接选择使用：

创建新的信息模板

1、按照ICANN政策要求，域名持有人Email邮箱必须真实准确，必须完成邮箱真实性验证的模板才能正常使用。[了解更多](#)

☒ 黄继升

huang ji sheng

326464623@qq.com (已验证)

已实名认证 [详情](#)

收起

企业邮箱免费版：( 50账号，5G/账号 )

☒ 立即开通，同意 [《企业邮箱免费版服务条款》](#)

选择绑定邮箱的域名：

huangjisheng.top ▼

绑定邮箱的域名仅支持英文域名：  
.com|.net|.xin|.xyz|.top|.cn|.site|.club|.ltd|.group|.shop|.wang

满足日常需求，初创公司首选！域名注册成功后，自动开通企业邮箱并添加邮箱解析；每位会员只能开通一个企业邮箱免费版。[了解更多](#)

重要提醒：每个帐号只能设置一个有效的发票抬头进行使用。发票抬头在 [用户中心-发票管理](#) 中设置

订单金额：¥ 1.00

立即购买

☐ 我已阅读，理解并接受 [《域名服务条款》](#)

## (3) 支付成功

支付

确认订单

支付

支付成功

✓ 恭喜，支付成功！

您可以[点击此处](#)索取发票或进入[域名控制台](#)进行产品管理

您还需要：[注册商标](#)保护您的品牌，[早注册早保护](#)

用户指南：[如何将域名指向网站](#)

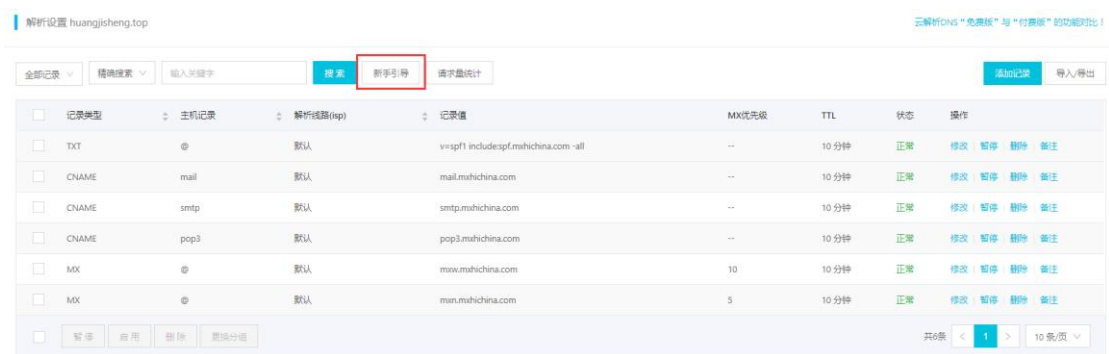
提示：[.com|.net|.cn|.xin|.top|.xyz|.vip|.club|.shop|.wang|.ren](#)等域名注册成功后必须进行域名实名认证，否则域名会处于Serverhold状态，无法正常使用。[域名实名认证问题FAQ](#)

## 2.添加域名解析

### (1)在阿里云控制台→域名与网站（万网）→云解析 DNS



### (2)点击进入该域名实例，并点击右上方新手引导



### (3)网站解析→输入公网 Ip



(3)快速设置解析完成

快速设置解析完成

✓ 解析记录设置完成！

成功添加2条记录：

记录类型	主机记录	解析线路	记录值	MX优先级	TTL
A	@	默认	47.100.254.225	--	10 分钟
A	www	默认	47.100.254.225	--	10 分钟

取消

确定

(4)网站备案

本以为这样就可以访问自己的网站了，谁知道弹出这么个提示：

☆ huangjisheng.top

85%

在此搜索

WEB 前端开发技术

包2课程说明 · Git

网络抓包工具 wireshark

GitHub

从0开始学习 Git

菜鸟教程 - 学的手到擒来

上网认证登录

学习CSS布局

Eclipse Color Theme

浙

温馨提示

English

该网站暂时无法访问

尊敬的用户，您好！

很抱歉，该网站暂时无法访问，可能由以下原因导致：

原因一：未备案或未接入；根据《非经营性互联网信息服务备案管理办法》，网站需要完成备案或接入。

原因二：网站内容与备案信息不符或备案信息不准确；根据《非经营性互联网信息服务备案管理办法》，网站内容需与备案信息一致，且备案信息需真实有效。建议网站管理员尽快修改网站信息。

[点此进入备案管理平台](#)

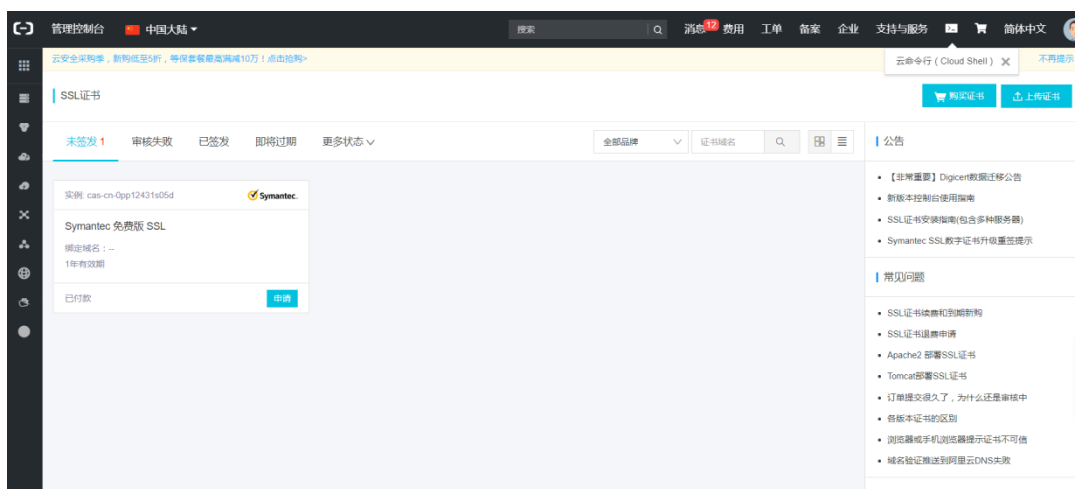
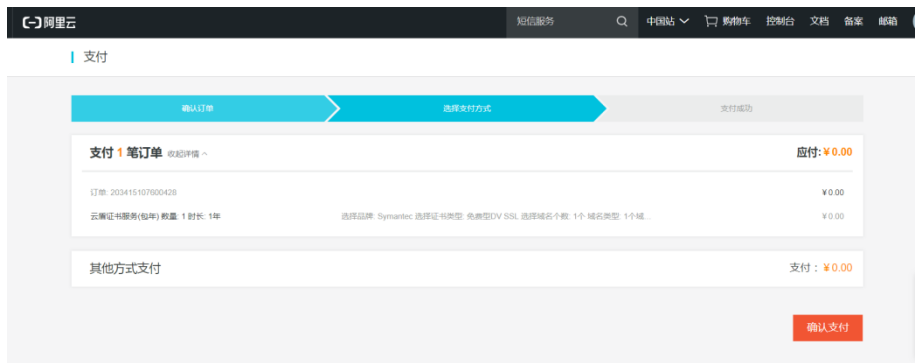
(2)采用阿里推荐的方法进行备案服务号申请

结果发现：

阿里云服务器申请备案服务号需购买时长在 3 个月及以上（含续费）  
所以大家自己考虑是否愿意购买三个月以上的服务器，再进行域名备案服务号申请。这里操作略。







点击申请

## 证书申请



填写申请

验证信息

- \* 证书绑定域名:  ?
- \* 所在地:  v
- \* 域名验证方式: ☒ 自动DNS验证 ? ☐ 手工DNS验证 ? ☐ 文件验证 ?
- \* 申请人姓名:
- \* 申请人手机号:  ?
- \* 申请人邮箱:  ?
- \* CSR生成方式: ☒ 系统生成 ☐ 手动填写

点击验证

## 证书申请



填写申请

验证信息

DNS验证方式一般需要由您的域名DNS管理人员进行操作。按照以下提示信息添加DNS解析记录，该验证信息在证书签发后可删除。

配置项目	配置项值
域名授权验证类型	DNS
记录类型	TXT ?
主机记录	_dnsauth ?
记录值	20190325000005euef5y67cqvvfe1at8hzab7lo5mowbacvmd0qqacagf4jab80 ?

验证

验证成功

最后提交审核  
状态变为已签发:

## SSL证书

未签发 审核失败 已签发 1 即将过期 更多状态 v

实例 cas-cn-0pp12431s05d

cert-1553563834475  
绑定域名: huangjisheng.top, www.huangjisheng.top  
到期时间: 2020年3月25日

已签发 [部署到云产品](#) v | [下载](#) | [吊销](#)

点击下载，选择 nginx 版本进行下载



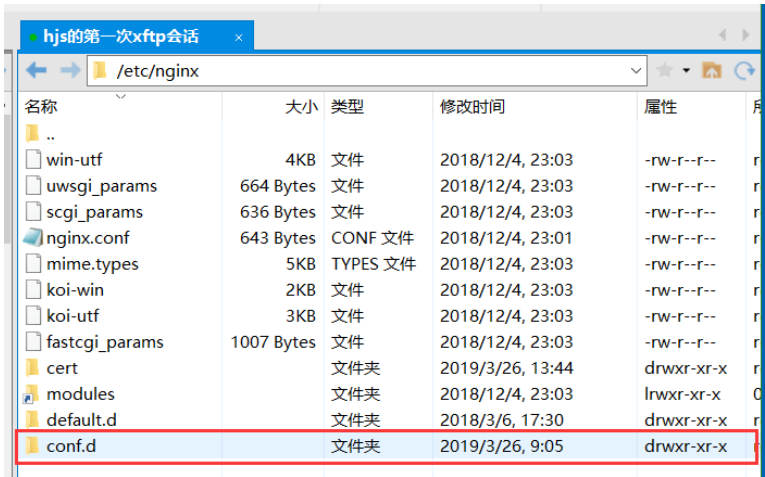
解压得到两个文件：

1977761\_huangjisheng.top.pem

1977761\_huangjisheng.top.key

为了方便直接手动进行创建工作

在/var/etc/nginx/conf 下创建文件夹 cert



将 SSL 的两个证书文件放入 cert 文件夹



修改 nginx.conf 配置文件，映射到服务器环境中

#vi /etc/nginx/conf.d/ghost.conf

```
#HTTPS server
upstream ghost{
    server 127.0.0.1:2368;
}
server {
    listen 443 ssl;
    server_name huangjisheng.top;
    ssl on;
    ssl_certificate cert/1977761_huangjisheng.top.pem; #pem证书文件名
    ssl_certificate_key cert/1977761_huangjisheng.top.key; #key证书文件名
    ssl_session_timeout 5m;
    ssl_ciphers ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_prefer_server_ciphers on;
    location / {
        proxy_set_header    X-Real-IP $remote_addr;
        proxy_set_header    Host      $http_host;
        proxy_pass            http://127.0.0.1:2368;
    }
}

#http自动跳转https,实现重定向
server {
    listen 80;
    server_name huangjisheng.top;
    rewrite ^(.*)$ https://$host$1 permanent;# 把http的域名请求转成https
}
```

#HTTPS server

server {

listen 443 ssl;

server\_name www.huangjisheng.top;

ssl on;

ssl\_certificate cert/1977761\_huangjisheng.top.pem; #pem 证书文件名

ssl\_certificate\_key cert/1977761\_huangjisheng.top.key; #key 证书文件名

ssl\_session\_timeout 5m;

ssl\_ciphers

ECDHE-RSA-AES128-GCM-SHA256:ECDHE:ECDH:AES:HIGH:!NULL:!aNULL:!MD5:!ADH:!RC4;

ssl\_protocols TLSv1 TLSv1.1 TLSv1.2;

ssl\_prefer\_server\_ciphers on;

location / {

proxy\_set\_header X-Real-IP \$remote\_addr;

proxy\_set\_header Host \$http\_host;

proxy\_pass http://127.0.0.1:2368;

}

}

#http 自动跳转 https,实现重定向

server {

listen 80;

server\_name huangjisheng.top;

rewrite ^(.\*)\$ https://\$host\$1 permanent;# 把 http 的域名请求转成 https

}

#检查 nginx 配置文件是否可用，successful 表示可用

# nginx -t

```
[root@iz01762evc6i3az ~]# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

#重新加载配置文件使配置生效

# nginx -s reload

#重启 nginx

# systemctl restart nginx.service

最后要注意：在阿里云控制台保持 HTTPS 443 端口开放

hjs的服务器后台  
47.106.294.225 | CentOS 7.3 | 上海 | 运行中

防火墙

应用类型	协议	端口范围	操作
HTTP	TCP	80	修改   删除
HTTPS	TCP	443	修改   删除
RDP	TCP	3389	修改   删除
SSH	TCP	22	修改   删除
自定义	TCP	8080	修改   删除
自定义	TCP	8079	修改   删除
自定义	TCP	2368	修改   删除

共有8条数据，每页显示条数：10

访问 <https://huangjisheng.top/>

