

Ubuntu 14下 LNMP环境搭建

2017年3月7日 18:24

参考

<http://blog.csdn.net/STFPHP/article/details/53492723>

首先用Xshell5连接上阿里云ECS

```
Welcome to aliyun Elastic Compute Service!
```

看到这句就说明连接上了

先装Nginx

直接输入apt-get install nginx, 然后发现炸了

```
Last login: Tue Mar  7 18:11:55 2017
root@iZwz9ffj2tm0lwx4vwplisZ:~# apt-get install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package nginx
```

不知道为什么, 反正用apt-get update命令更新一下资源列表就好了

```
root@iZwz9ffj2tm0lwx4vwplisZ:~# apt-get update
```

然后再apt-get install nginx

```
root@iZwz9ffj2tm0lwx4vwplisZ:~# apt-get install nginx
```

安装好之后, 使用 **dpkg -S nginx** 命令来搜索 nginx相关文件, 可以从命令显示结果看出 Nginx默认的安装位置是/etc/nginx目录, 其配置文件nginx.conf也是在该目录下, 并且在 etc/init.d 下有 nginx的启动程序, 该目录下的程序都会在系统开启时启动。

```
root@iZwz9ffj2tm0lwx4vwplisZ:~# dpkg -s nginx
Package: nginx
Status: install ok installed
Priority: optional
Section: httpd
Installed-Size: 94
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: all
Version: 1.4.6-1ubuntu3.7
Depends: nginx-core (>= 1.4.6-1ubuntu3.7) | nginx-full (>= 1.4.6-1ubuntu3.7)
ubuntu3.7.1~) | nginx-full (<= 1.4.6-1ubuntu3.7.1~) | nginx-light (<= 1.4.6-1
Description: small, powerful, scalable web/proxy server
 Nginx ("engine X") is a high-performance web and reverse proxy server
 created by Igor Sysoev. It can be used both as a standalone web server
 and as a proxy to reduce the load on back-end HTTP or mail servers.
.
 This is a dependency package to install either nginx-core (by default),
 nginx-full, nginx-light, nginx-extras, or nginx-naxsi.
Homepage: http://nginx.net
Original-Maintainer: Kartik Mistry <kartik@debian.org>
```

使用sudo service nginx start来启动Nginx

使用netstat -anp来查询端口使用情况，80端口是Nginx，为Listen就是使用中

```
root@iZwz9ffj2tm0lwx4vwplisZ:~# sudo service nginx start
root@iZwz9ffj2tm0lwx4vwplisZ:~# netstat -anp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:80             0.0.0.0:*               LISTEN      1677/nginx
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      756/sshd
tcp        0      0 172.18.172.243:22      125.71.229.7:17287      ESTABLISHED 1074/0
tcp        0      0 172.18.172.243:57591   106.11.68.13:80        ESTABLISHED 1010/AlibabaDun
tcp6       0      0 :::80                  :::*                     LISTEN      1677/nginx
udp        0      0 172.18.172.243:123     0.0.0.0:*               1004/ntpd
udp        0      0 127.0.0.1:123          0.0.0.0:*               1004/ntpd
udp        0      0 0.0.0.0:123            0.0.0.0:*               1004/ntpd
udp6       0      0 :::123                 :::*                     1004/ntpd
```

在浏览器访问根目录就能看见Nginx的欢迎页面了

120.77.37.195

ge 导入的书签 清水河畔 哔哩哔哩 斗鱼 慕课网 Google 翻译 百度 120.77.37.195 Spring4 视频教程 【2013.9.5修正版图】

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

再装PHP7

因为Ubuntu软件库里没有PHP7，因此要连另一个库

使用下列命令：

```
sudo apt-add-repository ppa:ondrej/php
sudo apt-get update
sudo apt-get install php7.1 php7.1-fpm
```

在输入第一行命令时，可能会报错

```
root@iZwz9ffj2tm0lwx4vwplisZ:~# apt-add-repository ppa:ondrej/php
The program 'apt-add-repository' is currently not installed. You can install it by typing:
apt-get install software-properties-common
```

跟着提示来，输入apt-get install software-properties-common安装一个包后，再使用上面的命令安装

安装好后输入php -v命令来看是否安装成功

```
root@iZwz9ffj2tm0lwx4vwplisZ:~# php -v
PHP 7.1.2-4+deb.sury.org-trusty+1 (cli) (built: Mar  2 2017 13:23:06) ( NTS )
Copyright (c) 1997-2017 The PHP Group
Zend Engine v3.1.0, Copyright (c) 1998-2017 Zend Technologies
    with Zend OPcache v7.1.2-4+deb.sury.org-trusty+1, Copyright (c) 1999-2017, by Zend Technologies
```

Nginx 与 PHP-FPM集成

Socket方式

在 /etc/nginx/sites-available/default 配置文件中（网站根目录也在是这里更改）
修改成如下：

```
root /usr/share/nginx/html;
index index.php index.html;

# Make site accessible from http://localhost/
server_name jinsong.online;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ /index.php;
    # Uncomment to enable naxsi on this location
    # include /etc/nginx/naxsi.rules
}

location ~ \.php$ {
    fastcgi_split_path_info ^(.+\.(php|php5))(/.+)$;
    # NOTE: You should have "cgi.fix_pathinfo = 0;" in php.ini

    # With php5-cgi alone:
    # fastcgi_pass 127.0.0.1:9000;
    # With php5-fpm:
    fastcgi_pass unix:/run/php/php7.1-fpm.sock;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param PATH_INFO $fastcgi_path_info;
    fastcgi_index index.php;
    include fastcgi_params;
}
```

教程说还要修改/etc/php/7.1/fpm/pool.d/www.conf文件中的内容变成如下

```
; Note: This value is mandatory.
listen = /run/php/php7.1-fpm.sock
```

但实际默认就是这样的啊！

配置好后重启服务

```
sudo /etc/init.d/nginx restart
sudo /etc/init.d/php7.1-fpm restart
```

最后安装MySQL

Apt-get安装MySQL5.7

<http://eyehere.net/2016/installation-of-mysql-server-5-7-on-ubuntu-14-04/>

卸载mysql

http://blog.csdn.net/qq_20545159/article/details/52026039

`sudo apt-get install mysql-server mysql-client php7.1-mysql`

使用命令 `netstat -anp`，会发现 3306端口正在被监听了

装好数据库后，要把数据库设置成utf-8支持中文编码

在/etc/mysql/my.cnf这个文件中增加如下两行代码

```
[client]
default-character-set=utf8 #增加这一行，指定字符集
port                = 3306
socket               = /var/run/mysqld/mysqld.sock

.....

[mysqld]
character_set_server=utf8 #增加这一行，指定字符集
user                = mysql
pid-file             = /var/run/mysqld/mysqld.pid
```

注意：CentOS安装的mysql配置文件在/etc/my.cnf

[client]的配置有，只用增加[mysqld]character_set_server=utf8

重启服务，字符集就成了UTF-8

```
$ sudo start mysql
$ sudo mysql -uroot -p
mysql> use root
mysql> show variables like '%char%'; #更改前存在的用户字符集不会改变，以后新建的用户字符集都变成了UTF-8
+-----+-----+
| Variable_name | Value |
+-----+-----+
| character_set_client | utf8 |
| character_set_connection | utf8 |
| character_set_database | latin1 |
| character_set_filesystem | binary |
| character_set_results | utf8 |
| character_set_server | utf8 |
| character_set_system | utf8 |
| character_sets_dir | /usr/share/mysql/charsets/ |
+-----+-----+
```

搞定！

Nginx下多CI框架部署

2017年3月9日 13:33

参考如下

来自 <<https://www.chenvudong.com/archives/codeigniter-in-nginx-and-url-rewrite.html>>

首先要修改CI的config.php

设置\$config['uri_protocol'] = "PATH_INFO";

然后修改nginx配置

我的配置如下

```
server {  
    listen 80;  
    listen [::]:80 ipv6only=on;  
    server_name localhost;  
  
    root /usr/share/nginx/html;  
    index index.php index.html index.htm;  
  
    location / {  
        try_files $uri $uri/ /index.php;  
        #try_files $uri $uri/ /index.php?$uri&$args;  
    }  
  
    location ~\.(php|\.*)$ {  
        fastcgi_split_path_info ^(.+\.php)(/.+)$;  
  
        #       # With php5-cgi alone:  
        #       fastcgi_pass 127.0.0.1:9000;  
        #       # With php7.1-fpm:  
  
        #       fastcgi_pass unix:/run/php/php7.0-fpm.sock;  
        #       fastcgi_pass 127.0.0.1:9000;  
        fastcgi_pass unix:/run/php/php7.1-fpm.sock;  
  
        fastcgi_param  PATH_INFO $fastcgi_path_info;  
        fastcgi_param  SCRIPT_FILENAME $document_root$fastcgi_script_name;  
        fastcgi_index  index.php;  
        include fastcgi_params;  
    }  
  
    # deny access to .htaccess files, if Apache's document root  
    # concurs with nginx's one  
    #
```





```
location ~ /\.ht {  
    deny all;  
}  
}
```

就可以访问多个CI框架了，但Url中无法去掉index.php

服务器疑难杂症

2017年3月14日 14:09

部署在nginx服务器下的CI框架中的控制器文件一定要大写开头命名

Windows10 (C:) > wamp64 > www > blog > application > controllers			
名称	修改日期	类型	大小
 Admin.php	2017/3/14 13:21	PHP 文件	9
 Home.php	2017/3/14 13:21	PHP 文件	11
 index.html	2017/1/9 22:32	Microsoft Edge	1
 Welcome.php	2017/1/9 22:32	PHP 文件	1

Config.php中

\$config['base_url'] = '<http://jinsong.online/blog/>'; 要设置成网站名, 而且要加http://不能是localhost,

配置nginx首页

location / {

#下面的proxy_pass 配置了访问[www.jinsong.online](http://jinsong.online)跳转的地址

proxy_pass <http://jinsong.online/blog/index.php/home/index;>

try_files \$uri \$uri/ /index.php;

#try_files \$uri \$uri/ /index.php?\$uri&\$args;

}

附上完整代码

```
server {
    listen 80;
    listen [::]:80 ipv6only=on;
    server_name jinsong.online;

    root /usr/share/nginx/html;
    index index.php index.html;

    location / {
        #下面的proxy_pass 配置了访问www.jinsong.online跳转的地址
        proxy_pass http://jinsong.online/blog/index.php/home/index;
        try_files $uri $uri/ /index.php;
        #try_files $uri $uri/ /index.php?$uri&$args;
    }

    location ~\.php($/.*)*$ {
        fastcgi_split_path_info ^(.+\.php)(/.+)$;

        fastcgi_pass unix:/run/php/php7.1-fpm.sock;

        fastcgi_param PATH_INFO $fastcgi_path_info;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_index index.php;
        include fastcgi_params;
    }

    location ~ /\.ht {
        deny all;
    }
}
```

```
,  
  
location ~ /\.ht {  
    deny all;  
}  
}
```


Ueditor 问题汇总

2017年3月20日 20:34

Ueditor上传附件大小

php（本机是wampserver）配置限制了上传文件大小。。。 [php.ini](#) 里边 post_max_size 、 upload_max_filesize 这两个参数改一下就好了

来自 <<http://www.bubuko.com/infodetail-1659715.html>>

Ueditor图片无法显示

脑 > 本地磁盘 (D:) > wamp64 > www > GuanGongWei > ueditor > php

名称	修改日期	类型
 action_crawler.php	2017/8/22 14:00	PHP 文件
 action_list.php	2017/8/22 14:00	PHP 文件
 action_upload.php	2017/8/22 14:00	PHP 文件
 config.json	2017/9/6 15:44	JSON File

```
config.json
1  /* 前后端通信相关的配置,注释只允许使用多行方式 */
2  {
3      /* 上传图片配置项 */
4      "imageActionName": "uploadimage", /* 执行上传图片的action名称 */
5      "imageFieldName": "upfile", /* 提交的图片表单名称 */
6      "imageMaxSize": 2048000, /* 上传大小限制,单位B */
7      "imageAllowFiles": [".png", ".jpg", ".jpeg", ".gif", ".bmp"], /*
8      "imageCompressEnable": true, /* 是否压缩图片,默认是true */
9      "imageCompressBorder": 1600, /* 图片压缩最长边限制 */
10     "imageInsertAlign": "none", /* 插入的图片浮动方式 */
11     "imageUrlPrefix": "http://localhost", /* 图片访问路径前缀 */
```

这个地方要修改!!!

首先发现问题

找到了编辑器的上传处理类 Uploader.class.php, 大约110行的位置找到了上传失败的提示位置,

将

```
$this->stateInfo = $this->getStateInfo("ERROR_CREATE_DIR");
```

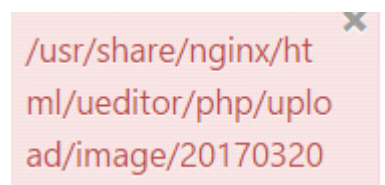
注释掉,

然后加了一句

```
$this->stateInfo = $dirname;
```

保存。

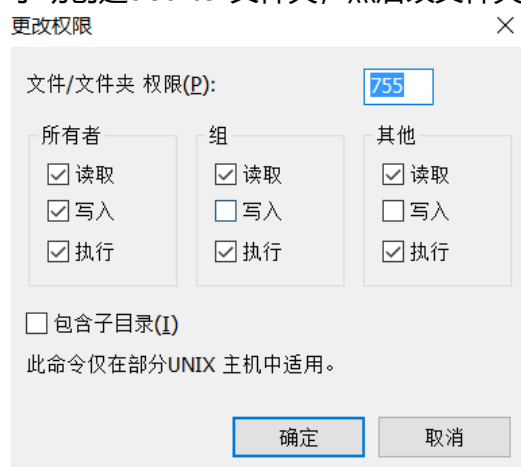
然后使用ueditor上传，错误提示的上传路径是：



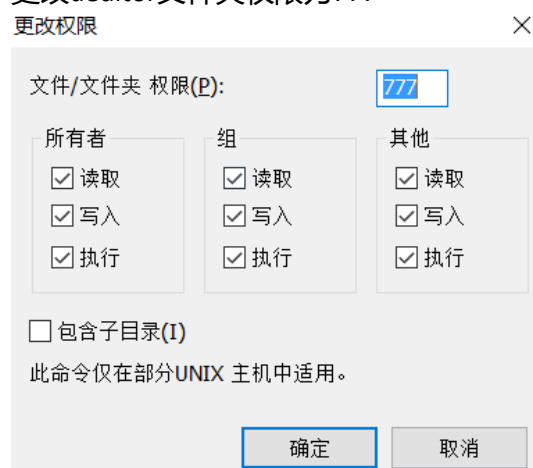
然后去找这个路径，发现在/usr/share/nginx/html/下根本没有创建ueditor文件夹

然后解决问题

手动创建ueditor文件夹，然后改文件夹的权限是755



更改ueditor文件夹权限为777



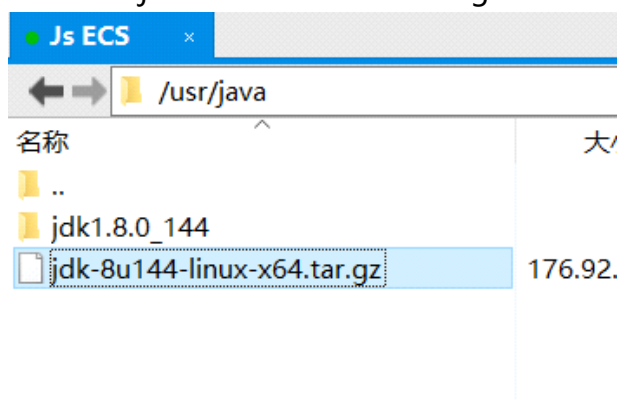
上传图片成功！

Ubuntu下安装Java

2017年8月6日 11:43

- 1、去官网上下载Linux版本的Java
- 2、在服务器上新建目录/usr/java/
- 3、解压tar.gz包

```
tar -zxvf jdk-8u144-linux-x64.tar.gz
```



- 4、vim /etc/profile

在末尾添加以下

方案一：

```
export JAVA_HOME=/usr/java/jdk1.8.0_144
export ZOOKEEPER_HOME=/usr/zookeeper/zookeeper-3.4.11
export PATH=$JAVA_HOME/bin:$ZOOKEEPER_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
```

方案二：

```
export JAVA_HOME=/usr/mysoftware/java/jdk1.8.0_161
export PATH=$PATH:$JAVA_HOME/bin
```

- 5、source /etc/profile

- 6、测试 java -version

```
root@iZwz9ffj2tm0lwx4vwplisZ:/usr/java# java -version
java version "1.8.0_144"
Java(TM) SE Runtime Environment (build 1.8.0_144-b01)
Java HotSpot(TM) 64-Bit Server VM (build 25.144-b01, mixed mode)
```

OK!!!!

Ubuntu 安装mysql

2018年4月23日 10:25

MySQL安装

<https://www.linuxidc.com/Linux/2017-05/143864.htm>

指定中文编码

[中文编码](#)

Ubuntu部署Maven项目

2017年8月6日 14:52

方法一

详看牛客网视频

<https://www.nowcoder.com/study/live/10/14/1>

- 1、服务器上安装好Java和Tomcat，用到的数据库也先放到服务器上
- 2、本地Maven项目打包（SpringBoot不影响任何操作）

(1)入口文件要继承SpringBootServletInitializer

```
@SpringBootApplication
public class MySecondKillApplication extends SpringBootServletInitializer {

    public static void main(String[] args) {
        SpringApplication.run(MySecondKillApplication.class, args);
    }
}
```

(2)除了入口文件的main函数，其他所有的main函数都要注释掉

(3)所有有关css\JS的引用和JS文件里的引用路径都要使用../而不是/

```
login.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta name="viewport" content="width=device-width, initial-scal
5 <meta charset="utf-8"/>
6 <title>Login Page | 金洵售后系统</title>
7
8
9 <link rel="stylesheet" href="../../assets/css/amazeui.min.css"/>
```

```
<!-- 本地js文件，前端的各种操作都在这里 -->
<!-- 部署到服务器上时，这里的src路径要写成../scripts/seckill.js而不是/scripts/seckill.js -->
<script src="../../scripts/seckill.js" type="text/javascript"></script>
```

```
//部署到服务器上时，这里的路径要写../time/now，不能写成/time/now
//否则会找不到路径，貌似js里的路径和有关js的路径都要这样写！！！！
now: function () {
    return '../time/now';
},
```

(4)本地数据库账号密码设置与服务器上要一致，localhost不用改

```
application.properties
1 #数据库配置
2 spring.datasource.url=jdbc:mysql://localhost:3306/mysecondkill?usel
3 spring.datasource.username=root
4 spring.datasource.password=83862973
5
```

(5)pom.xml中打包类型改为war，version标签是打包好的文件名后半部分

以上修改都不影响本地运行

```
<groupId>com.jinsong</groupId>
<artifactId>mySecondKill</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>war</packaging> <!-- 打包部署到服务器时，这里要把jar改成war -->
```

(5.1)添加一个依赖

```
<!-- 打成war包放入Tomcat中运行时需要添加的依赖 -->
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-tomcat</artifactId>
<scope>provided</scope>
</dependency>
```

(6)cmd进入到项目根目录，采用mvn clean package -Dmaven.test.skip=true命令打包项目

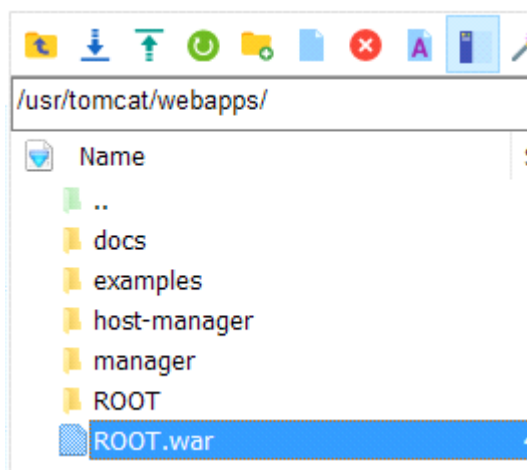
```
C:\Users\18894\Desktop\Design\JAVA\MoemilAfterSales>mvn clean package -Dmaven.test.skip=true
```

(7)这是项目根目录target/下打包好的

电脑 > 本地磁盘 (D:) > GitLocalRepository > mySecondKill > mySecondKill > target				
名称	修改日期	类型	大	
classes	2017/8/6 15:12	文件夹		
generated-sources	2017/8/6 14:40	文件夹		
m2e-wtp	2017/8/6 14:36	文件夹		
maven-archiver	2017/8/6 14:40	文件夹		
maven-status	2017/8/6 14:40	文件夹		
mySecondKill-0.0.1-SNAPSHOT	2017/8/6 15:08	文件夹		
test-classes	2017/8/6 14:35	文件夹		
mySecondKill-0.0.1-SNAPSHOT.war	2017/8/6 15:08	WAR 文件	2	
mySecondKill-0.0.1-SNAPSHOT.war.original	2017/8/6 15:08	ORIGINAL 文件	2	

(8)可重命名这个war包，比如为ROOT.war，然后放到服务器上去

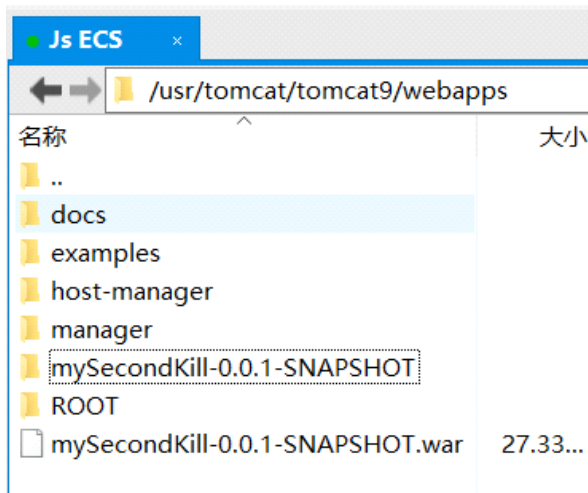
(9)为了避免SpringSecurity等框架的路径错误，最好把项目作为Tomcat的根目录访问，最简单的方法是删除原Tomcat/webapps/ROOT文件夹，然后把项目重命名为ROOT.war，然后放到webapps文件夹内。如下：ROOT内是我自己的项目



访问不需要再加项目名称！！！！



- 3、把这个war文件放到服务器Tomcat/webapps/下
那个对应的mySecondKill文件夹是放入war文件后自动生成的



- 4、服务器访问测试

注意url

同样的文件，服务器上URL是http://120.77.37.195:8080/mySecondKill-0.0.1-SNAPSHOT/list

本地是http://127.0.0.1:8080/list

多了war包的名字





都是OK的！！！！

4.很致命的错误：tomcat的webapps文件夹下有多个项目的war包

解决办法：此时需要我们去tomcat中的conf文件夹下，打开server.xml文件，然后在Host中修改，添加如下：

```
1 //docBase中的值是我们放到webapps文件夹下的war包的名
2 <Context path="/" docBase="map-1.0-SNAPSHOT" debug="0" privileged="true" reloadable="true" />
```

*这样就能防止webapps文件夹下有多个war的时候服务器找不到文件而报404的错误了！

方法二

pom中修改为war，然后添加下面的plugin避免编译报错

```
<!-- 这个plugin使得测试出错不影响项目的编译。可以避免Run as Maven ins
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <configuration>
    <testFailureIgnore>true</testFailureIgnore>
  </configuration>
</plugin>
```

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <configuration>
    <testFailureIgnore>true</testFailureIgnore>
  </configuration>
</plugin>
```

最后右键pom.xml Run As Maven Install

Ubuntu 安装Tomcat (单个、多个)

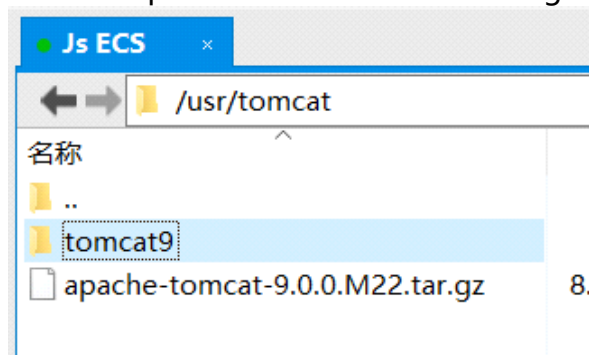
2017年8月6日 12:31

Ubuntu下安装Tomcat9

1、去官网下载Tomcat9 tar.gz

2、复制到服务器目录下解压

`tar -zxvf apache-tomcat-9.0.0.M22.tar.gz`



3、将解压后的文件夹重命名为tomcat9

`mv apache-tomcat-9.0.0.M22 tomcat9`

4、进入tomcat9/bin/目录下直接启动tomcat9

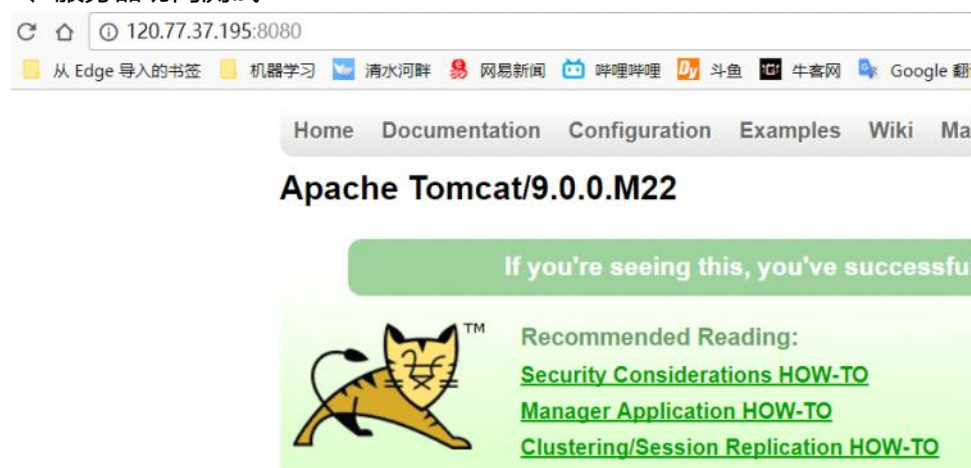
```
root@iZwz9ffj2tm01wx4vwplisZ:/usr/tomcat/tomcat9/bin# ./startup.sh
Using CATALINA_BASE:   /usr/tomcat/tomcat9
Using CATALINA_HOME:   /usr/tomcat/tomcat9
Using CATALINA_TMPDIR: /usr/tomcat/tomcat9/temp
Using JRE_HOME:        /usr/java/jdk1.8.0_144/jre
Using CLASSPATH:       /usr/tomcat/tomcat9/bin/bootstrap.jar:/usr/t
Tomcat started.
```

显示Tomcat started, 启动成功

下面是进入了tomcat9/bin目录下

```
[root@localhost bin]# ./startup.sh
Using CATALINA_BASE:   /usr/mysoftware/tomcat/tomcat9
Using CATALINA_HOME:   /usr/mysoftware/tomcat/tomcat9
Using CATALINA_TMPDIR: /usr/mysoftware/tomcat/tomcat9/temp
Using JRE_HOME:        /usr/mysoftware/java/jdk1.8.0_151/jre
Using CLASSPATH:       /usr/mysoftware/tomcat/tomcat9/bin/boot
t9/bin/tomcat-juli.jar
Tomcat started.
```

5、服务器访问测试



启动Tomcat命令

进入tomcat/bin目录后 ./startup.sh

```
root@iZwz9ffj2tm01wx4vwplisZ:/usr/tomcat/tomcat9/bin# ./startup.sh
```

停止Tomcat命令

进入tomcat/bin目录后 ./shutdown.sh

Linux下Tomcat端口号以及进程相关命令

来自 <<http://blog.csdn.net/theonegis/article/details/49121497>>

外部访问虚拟机的Tomcat

CentOS自带防火墙，要关掉，否则无法访问

每次重启CentOS都要关闭一次！！！！

来自 <http://blog.csdn.net/qq_29352959/article/details/53816652>

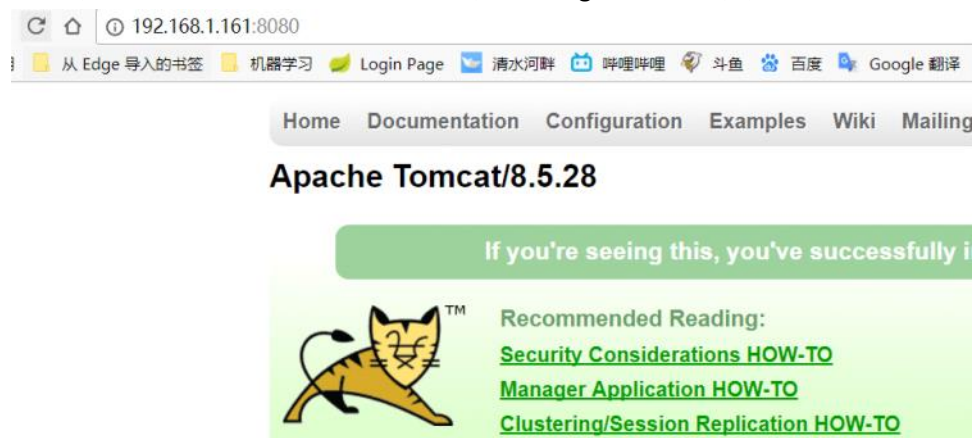
最上面的Complete! 是yum install iptables成功

之后就是下面的操作

service iptables stop firewallld.service&& sudo systemctl disable firewallld.service

```
Complete!
[root@localhost bin]# service iptables stop
Redirecting to /bin/systemctl stop iptables.service
Failed to stop iptables.service: Unit iptables.service not loaded.
[root@localhost bin]# stop firewallld.service&& systemctl disable firewallld.service
bash: stop: command not found...
Similar command is: 'top'
[root@localhost bin]# stop firewallld.service
bash: stop: command not found...
Similar command is: 'top'
[root@localhost bin]# service iptables stop firewallld.service&& sudo systemctl disable firewallld.service
Redirecting to /bin/systemctl stop firewallld.service iptables.service
Failed to stop iptables.service: Unit iptables.service not loaded.
[root@localhost bin]#
```

然后就能访问了！！ 192.168.1.161是ifconfig查到的虚拟机IP地址



单机部署多个Tomcat

参考下面的文章，一步一步照着做

来自 <<https://www.cnblogs.com/shihaiming/p/5896283.html>>

Ubuntu Nginx安装及配置

2018年4月26日 16:54

参考

<https://www.cnblogs.com/zhouxinfei/p/7862285.html>

Nginx安装

Ubuntu下直接apt-get install nginx

```
root@iZwz9f9222teogvo391tctZ:~# apt-get install nginx
```

安装完成即可，在/usr/sbin/目录下是nginx命令所在目录，在/etc/nginx/目录下是nginx所有的配置文件，用于配置nginx服务器以及负载均衡等信息

查看nginx进程是否启动

ps -ef|grep nginx

```
root@iZwz9f9222teogvo391tctZ:~# ps -ef|grep nginx
root      844      1  0 16:54 ?        00:00:00 nginx: master process /usr/sbin
nginx -g daemon on; master_process on;
www-data  845    844  0 16:54 ?        00:00:00 nginx: worker process
root      904    449  0 16:56 pts/0    00:00:00 grep --color=auto nginx
```

启动nginx服务器命令

service nginx start

安装之后Nginx会自动启动的

停止nginx服务命令

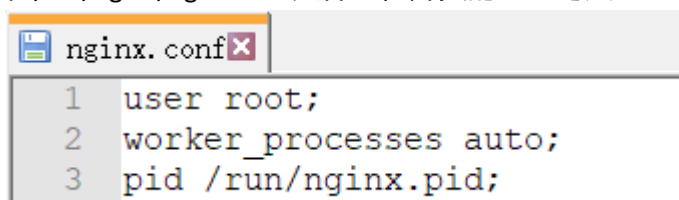
service nginx stop

重新启动加载

nginx -s reload

Nginx+Tomcat（请求转发到8080端口）

1. 在/etc/nginx/nginx.conf文件里，开头的user 写成user root



```
1 user root;
2 worker_processes auto;
3 pid /run/nginx.pid;
```

2. 其他默认配置都不用改，在http{末端添加server{}}

下面这个server是写在http里面的

```
http {
```

```
#我的配置
#upstream my_tomcat_server{
#    server 112.74.51.120:8080;
#}
```

```

#我的配置
#upstream my_tomcat_server{
#    server 112.74.51.120:8080;
#}

#我的配置
server{
    listen 80;
    server_name sh.jinhuan123.com;
    charset utf-8;

    #动静分离
    location ~ .*\. (jpg|png|js|css)$
    {
        #root是静态资源存放的位置
        root    /usr/share/nginx/html/resources/static;
        if (-f $request_filename) {
            expires 1d;
            break;
        }
    }

    #将请求转发给nginx
    location / {
        proxy_set_header HOST $host:$server_port;
        proxy_set_header X-Forwarded-Server $host; #隐藏端口
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_pass http://sh.jinhuan123.com:8080/; #从这里可以看到，把进来的80端口请求转发到了8080端口
    }
}

```

文字如下:

#我的配置

```

server{
    listen 80;
    server_name sh.jinhuan123.com; #域名
    charset utf-8;

    #动静分离
    location ~ .*\. (jpg|png|js|css)$
    {
        #root是静态资源存放的位置
        root    /usr/share/nginx/html/resources/static;
        if (-f $request_filename) {
            expires 1d;
            break;
        }
    }

    #将请求转发给nginx
    location / {
        proxy_set_header HOST $host:$server_port;
        proxy_set_header X-Forwarded-Server $host; #隐藏端口
        proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        proxy_pass http://sh.jinhuan123.com:8080/; #从这里可以看
到，把进来的80端口请求转发到了8080端口
    }
}

```

Nginx动静分离

就是配置location 将jpg|png|js|css这些资源请求转发到root字段配置的目录

```
#我的配置
server{
    listen 80;
    server_name sh.jinhuan123.com;
    charset utf-8;

    #动静分离
    location ~ .*\. (jpg|png|js|css)$
    {
        #root是静态资源存放的位置
        root /usr/share/nginx/html/resources/static;
        if (-f $request_filename) {
            expires 1d;
            break;
        }
    }
}
```

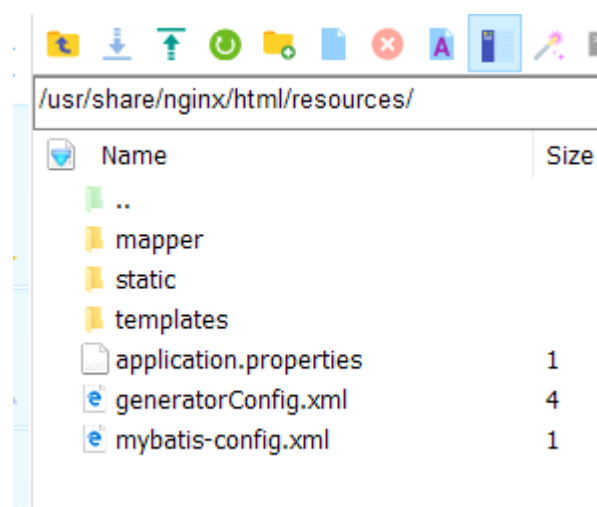
代码里css/js的路径都是

```
<!--新 Bootstrap 核心 CSS 文件 本地版-->
<link href="/bootstrap-3.3.7-dist/css/bootstrap.min.css"
```

组合起来就是/usr/share/nginx/html/resources/static/bootstrap-3.3.7.....

```
<link rel="stylesheet" href="..|/assets/css/amazeui.min.css"/>
```

为什么bootstrap不用加两个点，我也不知道呀。。。



因此，我们要把静态资源单独抽出来放到我们配置的这个目录，为什么不配置成Tomcat里的项目路径下的目录呢？貌似如果配置成Tomcat里的，那么请求就会让Tomcat处理一遍，就失去了动静分离，缓存静态资源的意义了。

CentOS配置

2018年3月2日 14:04

CentOS配置

设备	摘要
内存	2 GB
处理器	2
硬盘(SCSI)	40 GB
CD/DVD (IDE)	自动检测
网络适配器	桥接模式(自动)
USB 控制器	存在
声卡	自动检测
打印机	存在
显示器	自动检测

处理器

处理器数量(P): 1

每个处理器的核心数量(C): 2

处理器核心总数: 2

虚拟化引擎

首选模式(M): 自动

☐ 禁用二进制转换加速(D)

☐ 虚拟化 Intel VT-x/EPT 或 AMD-V/RVI(V)

☐ 虚拟化 CPU 性能计数器(U)

设备	摘要
内存	2 GB
处理器	2
硬盘(SCSI)	40 GB
CD/DVD (IDE)	自动检测
网络适配器	NAT
USB 控制器	存在
声卡	自动检测
打印机	存在
显示器	自动检测

设备状态

☐ 已连接(C)

☒ 启动时连接(Q)

网络连接

☒ 桥接模式(B): 直接连接物理网络

☒ 复制物理网络连接状态(P)

☐ NAT 模式(N): 用于共享主机的 IP 地址

☐ 仅主机模式(H): 与主机共享的专用网络

☐ 自定义(U): 特定虚拟网络

桥接模式更安全

CentOS安装Mysql

2018年3月2日 14:25

CentOS下安装mysql

来自 <<https://www.cnblogs.com/huangxinyuan650/p/6360464.html>>

这篇文章的下面这一步可以省略

然后我们来测试是否安装成功，这时如果直接像windows下一样输入mysql -uroot -p可能会出现一些错误
(ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/lib/mysql/mysql.sock' (2))
原因是mysql这个文件的权限问题，需要把这个文件的权限改为当前用户

```
sudo chown -R openscanner:openscanner /var/lib/mysql
```

其他都按照这篇文章来

本地Navicat for MySQL连接虚拟机中的MySQL

2018年3月2日 14:37

<https://coding.imooc.com/lesson/186.html#mid=11472>

直接连接会报错如下



解决方案:

1、修改mysql.user里的host字段 -> %

```
mysql> select host,user from mysql.user;
+-----+-----+
| host      | user |
+-----+-----+
| 127.0.0.1 | root |
| ::1       | root |
| localhost |      |
| localhost | root |
| localhost.localdomain |      |
| localhost.localdomain | root |
+-----+-----+
6 rows in set (0.00 sec)
```

把127.0.0.1修改为%就能所有机器都访问了

```
mysql> update mysql.user set host='%' where user='root' and host='127.0.0.1';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select host,user from mysql.user;
+-----+-----+
| host      | user |
+-----+-----+
| %         | root |
| ::1       | root |
| localhost |      |
| localhost | root |
| localhost.localdomain |      |
| localhost.localdomain | root |
+-----+-----+
```

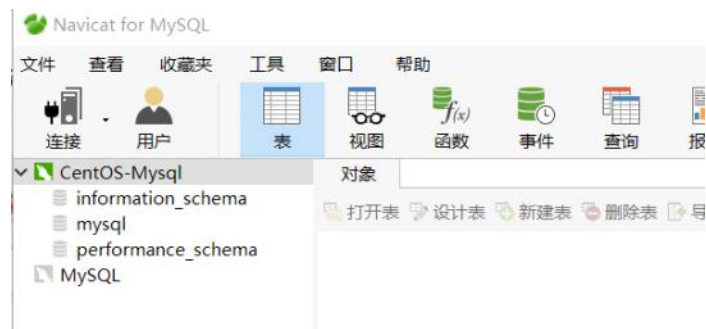
因为权限变化了，所以要刷新权限

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

Navicate要设置SSH



连接成功!!!!



Linux命令

2017年9月5日 9:44

ls -l 等同于 ll ls -d	显示当前目录下文件的属性 仅列出目录
ls -al	显示当前目录下所有文件详细信息，包括隐藏文件
cd [~] [-]	切换目录 [~目前用户身份的主目录] [-前一个工作目录]
chgrp 用户组 文件名	修改文件的用户组
chown 用户名 文件名	修改文件的所有者
chmod 770 文件名	修改文件的权限
cp A B	复制文件A为B
rm A rm -r A	删除A 删除非空目录或文件A
su root	切换为root用户
mkdir/rmdir mkdir -p test1/test2 mkdir -m 777 test2	新建目录、删除一个空目录 创建多层空目录 创建目录时设定权限
touch	新建空的文件
man 命令	查询命令详细解释
bzip2	压缩文件
bunzip2	解压文件
nano	文本编辑器
pwd	显示当前目录
echo	打印
mv 文件名 目标位置 mv A B	移动文件 重命名A为B（目录或文件都可以）
PATH="\$PATH":/root	添加/root到环境变量中
vim	进入后按i可以输入 退出按ESC然后:wq 不保存退出 按ESC 然后:q! 一般模式中： /word 向下寻找word字符串

	?word 向上寻找word字符串 n 重复前一个查找 N 与 n反向重复查找
umask -S	查看目前用户在新建文件或目录时候的默认权限
find	搜索文件，很复杂，很多参数
tar -jcv -f 名字.tar.bz2 A	压缩 “A” 变成 “名字.tar.bz2”
tar -jtv -f 名字.tar.bz2	查询
tar -jxv -f 名字.tar.bz2 [-C 指定目录]	在当前目录解压 【-C 在指定目录解压】
groupadd A groupdel A	新建用户组A 删除用户组A
groups [A]	查看自己所在的用户组，【查看A所在的用户组】
usermod usermod -G XXX B	该命令有很多参数，可以修改账号各个属性 新建xxx用户组为B的支持用户组，B原来的用户组仍有 (支持用户组不是当前用户组，有效用户组才是当前用户组)
newgrp B	切换当前用户的有效用户组为B
useradd A useradd -u 666 -g B -c "XXX" A	新增用户A (必须要设置密码才能用) 新增用户A，用户组为B，UID为666，账号全名是XXX
passwd A echo "XXX" passwd -- stdin A	给用户A设置密码，若没有A，则是给自己设置密码， 密码需要超过8个字符 设置用户A的密码为XXX
passwd -l A passwd -u A passwd -S A	使账号A密码失效 (让其无法登陆) 使账号A密码恢复 查询账号A密码状态
userdel -r A	删除用户A，连同用户主文件夹一起删除 (慎用)
setfacl -m u:A:rwx B setfacl -m g:A:rx B setfacl -b A	设置账户A针对文件B的权限为rwx (针对单独用户设置 权限) 设置用户组A针对文件B的权限为rx 消除文件A的ACL权限
getfacl B	查询文件B的权限详情
ctrl+c	终止当前程序运行
ctrl+alt+F1 ctrl+alt+F2-F7	切回图形界面 切回命令行界面
yum install XXX	CentOS的apt-get install XXX
which XXX	检测某个xxx应用是否安装

reboot	重启服务器
ifconfig	查看Linux（包括本地虚拟机的Linux）的IP地址
shift+PgUp\PgDn	命令行界面上下滚动
启动命令 &	在后台启动，不占用命令窗口，比如启动Redis的时候 ./redis-server &
kill -9 PID	关闭服务，比如某程序PID=6817 kill -9 6817 就关闭了这个服务
top	查看当前系统负载情况，如果是单核CPU 那么load average低于1说明没有线程等待
netstat -nap grep 5672	查看端口号5672是否被监听
free -m	查看内存使用情况
ps -A	显示所有运行中的进程
scp jinsong@IP地址:/远程目录/文件名 ./	拷贝远程电脑的文件到本地
df -h	显示磁盘上可用空间
du -h grep "G"	对文件和目录磁盘使用的空间查看，过滤只显示带有G的数据
ps 参数	显示进程状态
ps aux grep admin	显示admin项目是由哪个java版本启动的
god status	显示当前机器上部署的服务

Linux上运行SQL

2018年4月18日 18:45

参考下面文章

来自 <<https://blog.csdn.net/askycat/article/details/70991389>>