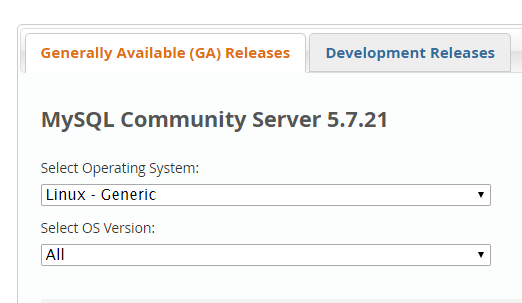
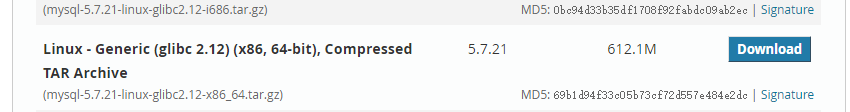
## 下载压塑包

[**Linux CentOS6.8下解压安装mysql-5.7.14完整介绍**](http://www.cnblogs.com/bangaj/p/6014397.html)

[**centos6.8下使用压缩包安装mysql**](http://blog.csdn.net/yangchuan_csdn91/article/details/54632598) <https://dev.mysql.com/downloads/mysql/>





<https://dev.mysql.com/downloads/file/?id=474755>



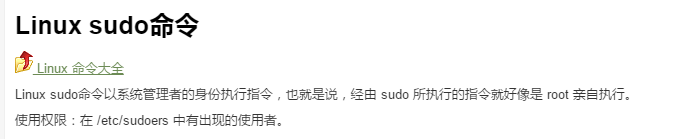
下载

wget http://dev.mysql.com/get/Downloads/MySQL-5.7/mysql-5.7.11-linux-glibc2.5-x86\_64.tar.gz

安装文章

1.下载和解压mysql数据库

|  |
| --- |
| 1.下载和解压mysql数据库 http://blog.csdn.net/lppklm/article/details/50977775 |



1. 暴力删除mysql用户

删除语法: userdel -r -f mysql(如果没有，会提示你mysql不存在，不用在意)

2. 添加分组

添加语法: groupadd mysql

3. 添加mysql用户

添加语法: useradd -g mysql mysql

4. 给mysql用户设置密码

passwd mysql

sudo tar -zxvf mysql-5.7.21-linux-glibc2.12-x86\_64.tar.gz  
[root@localhost local]# cp -r mysql-5.7.21-linux-glibc2.12-x86\_64 /usr/local/mysql  
sudo mv mysql-5.7.11-linux-glibc2.5-x86\_64/ mysql-5.7.11



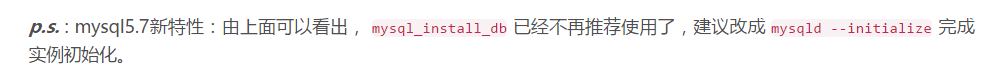


./mysql\_install\_db --user=mysql --basedir=/usr/local/data/mysql/ --datadir=/usr/local/soft/mysql/data/

改为：

./ mysqld --user=mysql --basedir=/usr/local/soft/mysql/ --datadir=/usr/local/soft/mysql/data/ --initialize

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



**p.s.** : mysql5.7新特性：由上面可以看出， mysql\_install\_db 已经不再推荐使用了，建议改成 mysqld --initialize 完成实例初始化。

<http://blog.csdn.net/cryhelyxx/article/details/49757217>

# MYSQL 安装

## Linux(Centos7)yum安装最新mysql

<http://blog.csdn.net/gebitan505/article/details/54613549>

## denied for user 'root'@'localhost'

[**重置密码解决MySQL for Linux错误 ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)**](http://www.cnblogs.com/gumuzi/p/5711495.html)

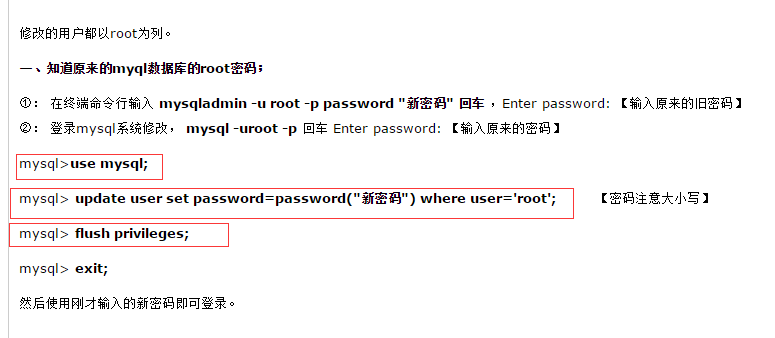
<http://www.cnblogs.com/gumuzi/p/5711495.html>

|  |
| --- |
| systemctl stop mysqld |
| #vim /etc/my.cnf(注：windows下修改的是my.ini)  skip-grant-tables |
|  |
| systemctl start mysqld |
| grant all privileges on \*.\* to 'root'@'localhost' identified by 'root' with grant option; |
| mysqladmin -u root password 123456 |
| http://blog.51cto.com/53cto/1841404  update user set authentication\_string = password('123456'),password\_expired = 'N', password\_last\_changed = now() where user = 'root'   flush privileges |
|  |
| https://www.cnblogs.com/ivictor/p/5142809.html  设置 远程连接  SHOW VARIABLE  SHOW VARIABLES LIKE 'validate\_password%';  set global validate\_password\_policy=**0**; |

<http://blog.csdn.net/guoguoshizhuo/article/details/47168305>

<https://www.cnblogs.com/kyosusan/p/5198934.html>

update user set password=password('123456') where user='root'





<http://blog.csdn.net/Lh19931122/article/details/77996213>





# Repeat：Centos 7 and mysql 5.21 安装

## 配置YUM源

在MySQL官网中下载YUM源rpm安装包：http://dev.mysql.com/downloads/repo/yum/



### #下载mysql源安装包

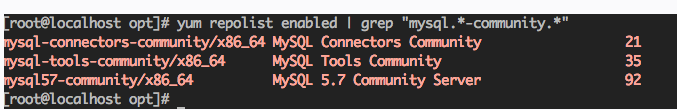
shell> wget http://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm

### #安装mysql源

shell> yum localinstall mysql57-community-release-el7-8.noarch.rpm

### 检查mysql源是否安装成功

shell> yum repolist enabled | grep "mysql.\*-community.\*"



#### 安装版本选择

|  |
| --- |
| 看到上图所示表示安装成功。  可以修改vim /etc/yum.repos.d/mysql-community.repo源，改变默认安装的mysql版本。比如要安装5.6版本，将5.7源的enabled=1改成enabled=0。然后再将5.6源的enabled=0改成enabled=1即可。改完之后的效果如下所示：  这里写图片描述 |

## 2、安装MySQL

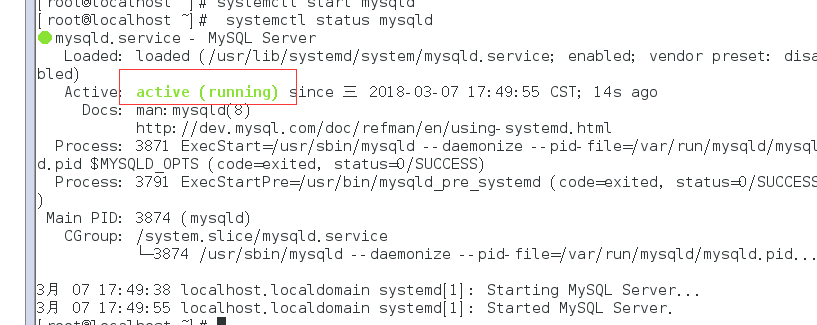
shell> yum install mysql-community-server

## 3、启动MySQL服务

shell> systemctl start mysqld

### 查看MySQL的启动状态

systemctl status mysqld



## 4、开机启动

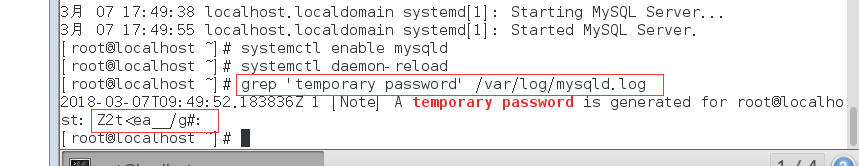
shell> systemctl enable mysqld

shell> systemctl daemon-reload

## 5、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：

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

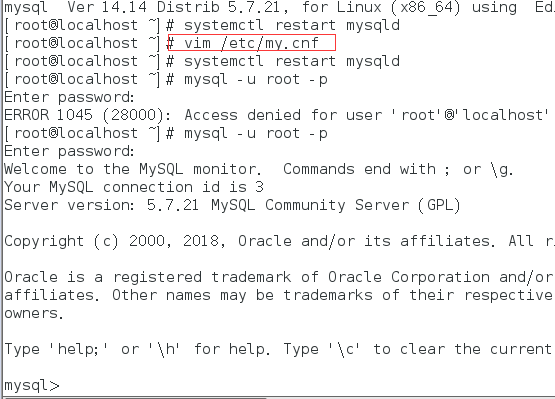


ALTER USER 'root'@'localhost' IDENTIFIED BY '12345678'

|  |
| --- |
| vim /etc/my.cnf  skip-grant-tables=1 |

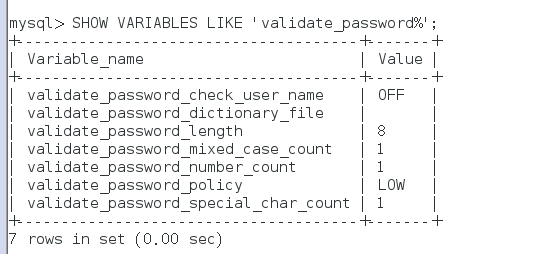
systemctl restart mysqld

|  |
| --- |
| use mysql  update user set authentication\_string = password('12345678'),password\_expired = 'N', password\_last\_changed = now() where user = 'root';  flush privileges; |



## 6.设置远程访问连接

SHOW VARIABLES LIKE 'validate\_password%';



set global validate\_password\_policy=0;

### 设置访问账户权限：

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY '12345678' WITH GRANT OPTION;

flush privileges;

### 开启防火墙3306端口

#firewall-cmd --zone=public --add-port=3306/tcp --permanent //permanent永久生效，没有此参数重启后失效

#### 重启防火墙

firewall-cmd --reload

# Nginx 安装

https://www.linuxidc.com/Linux/2016-09/134907.htm

## 1.nginx安装环境

### Gcc

如果没有gcc环境，需要安装gcc：yum install gcc-c++

|  |  |
| --- | --- |
| gcc版本 | **gcc -v** |
| 查看Gcc  rpm -qa|grep gcc |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### PCRE

|  |
| --- |
| PCRE(Perl Compatible Regular Expressions)是一个Perl库，包括 perl 兼容的正则表达式库。nginx的http模块使用pcre来解析正则表达式，所以需要在linux上安装pcre库  **yum install -y pcre pcre-devel**    http://blog.csdn.net/qianlong4526888/article/details/9497165  yum install -y pcre pcre-devel --setopt=protected\_multilib=false  注：pcre-devel是使用pcre开发的一个二次开发库。nginx也需要此库。 |
| perl -v |
| rpm -qa|grep perl |
|  |
|  |
|  |

### zlib

zlib库提供了很多种压缩和解压缩的方式，nginx使用zlib对http包的内容进行gzip，所以需要在linux上安装zlib库。

**yum install -y zlib zlib-devel**

**yum install -y zlib zlib-devel** --setopt=protected\_multilib=false

### openssl

OpenSSL 是一个强大的安全套接字层密码库，囊括主要的密码算法、常用的密钥和证书封装管理功能及SSL协议，并提供丰富的应用程序供测试或其它目的使用。

nginx不仅支持http协议，还支持https（即在ssl协议上传输http），所以需要在linux安装openssl库。

**yum install -y openssl openssl-devel**

## 下载对应当前系统版本的nginx包(package)

# wget  <http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.ngx.noarch.rpm>

## 建立nginx的yum仓库

# rpm -ivh nginx-release-centos-7-0.el7.ngx.noarch.rpm

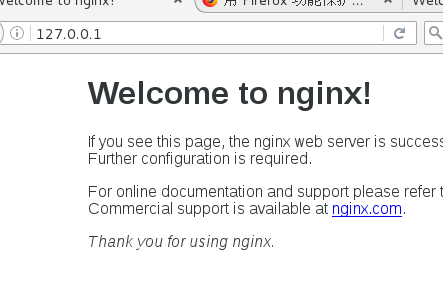
## 下载并安装nginx

# yum install nginx

## 启动nginx服务

systemctl start nginx

## 访问

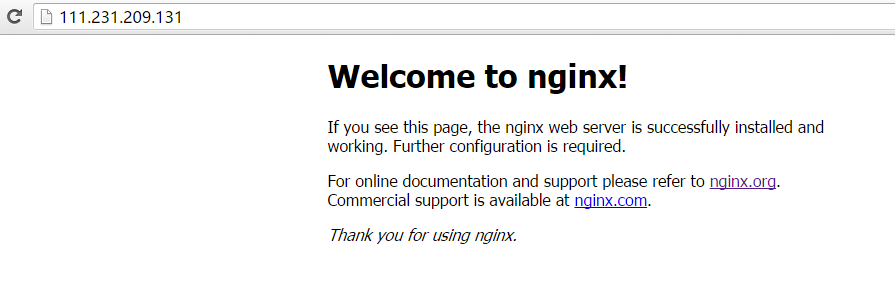


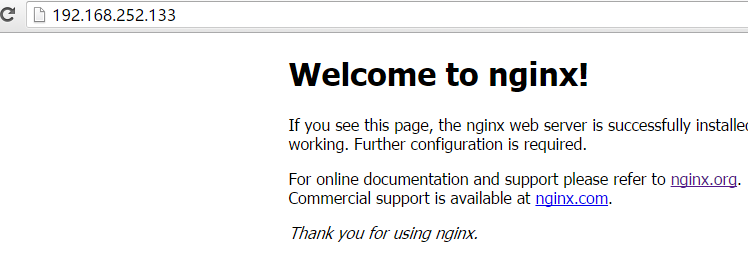
### 开启防火墙80端口

#firewall-cmd --zone=public --add-port=80/tcp --permanent //permanent永久生效，没有此参数重启后失效

#### 重启防火墙

firewall-cmd –reload





## **Nginx配置信息**

网站文件存放默认目录

/usr/share/nginx/[html](https://edu.aliyun.com/jiaocheng/291)

自定义Nginx站点配置文件存放目录

/etc/nginx/conf.d/

Nginx全局配置

/etc/nginx/nginx.conf

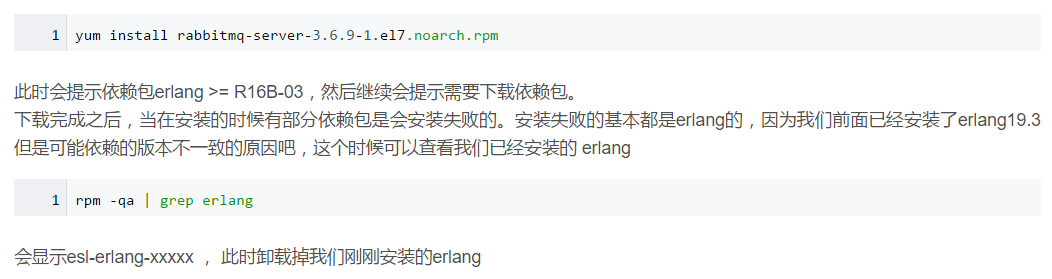


# RabbitMQ 安装

## 安装Rabbitmq

### 下载

wget <http://www.rabbitmq.com/releases/rabbitmq-server/v3.6.9/rabbitmq-server-3.6.9-1.el7.noarch.rpm>



rmp -e esl-erlang

### 安装

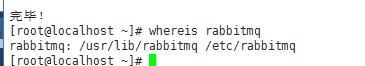
yum install rabbitmq-server-3.6.9-1.el7.noarch.rpm

### 获取安装路径

whereis rabbitmq

whereis erlang

rabbitmq: /usr/lib/rabbitmq /etc/rabbitmq



### 新增管理插件

#### 安装插件

rabbitmq-plugins enable rabbitmq\_management

#### 浏览管理页面

<http://127.0.0.1:15672/#/>

#### 登录前 新增用户

rabbitmqctl  add\_user  admin  123456

rabbitmqctl set\_user\_tags admin administrator

rabbitmqctl set\_permissions -p "/" admin ".\*" ".\*" ".\*" //（可以有）

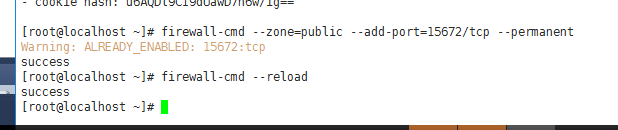
service rabbitmq-server restart

### 开启防火墙15672端口

#firewall-cmd --zone=public --add-port=15672/tcp --permanent //permanent永久生效，没有此参数重启后失效

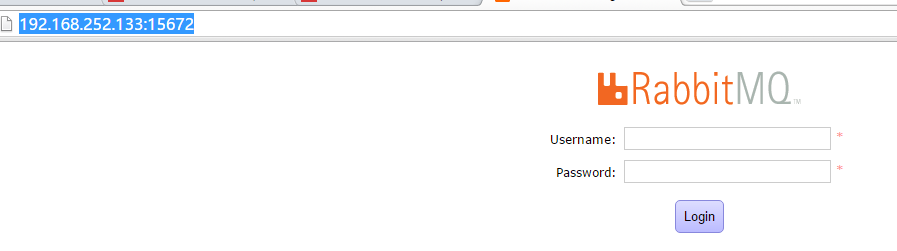
#### 重启防火墙

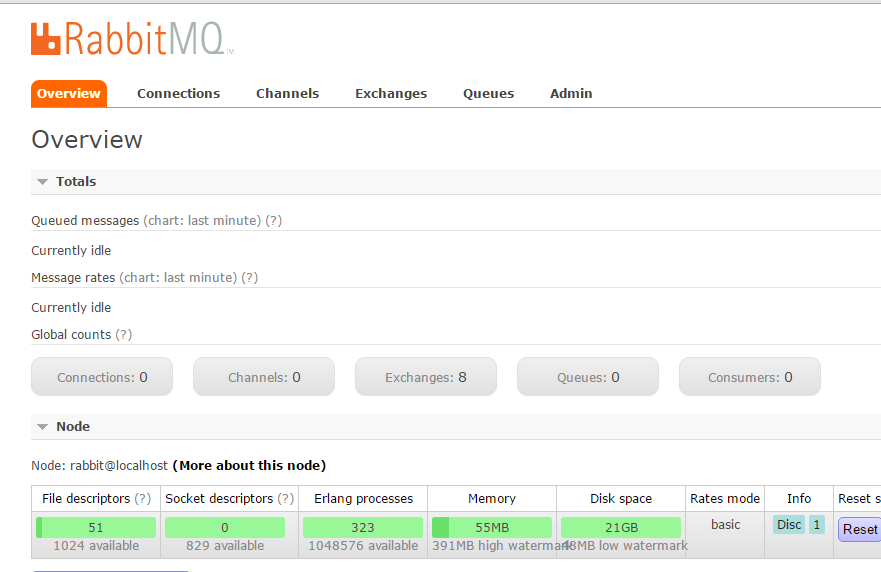
firewall-cmd --reload



### 远程访问网页

<http://192.168.252.133:15672/>





# centos7使用yum安装Redis

## 1.centos7使用yum安装Redis时

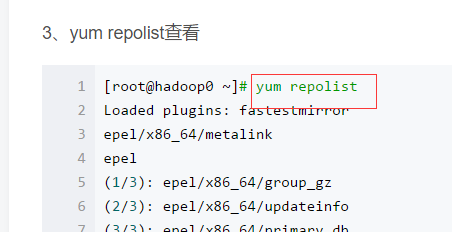
，可能会有安装源的问题出现。安装epel源，CentOS默认的安装源在官方的centos.org上，而redis在第三方的yum源里，因此无法安装。这就是我们常常在yum源里找不到各种软件的原因，还需要自己去wget，然后configure,make,make install，这个过程太痛苦了，并且卸载软件的时候还容易出错。

## 2、非官方的yum推荐用fedora的epel仓库。

yum添加epel源的命令为：yum install epel-release然后回车。

[root@hadoop0 ~]# yum install epel-release

## 3.yum repolist 查看



## 4、执行安装redis命令：yum install redis即可

[root@hadoop0 ~]# yum install redis

## 5、查看Redis安装了哪些文件

find / -name "redis\*"



## 6、启动Redis服务：使用service redis start命令启动redis服务端。

[root@hadoop0 ~]# service redis start

## 7、打开Redis客户端，使用命令：redis-cli 即可打开客户端

[root@hadoop0 ~]# redis-cli

127.0.0.1:6379>

## 8、远程访问，

### 解决办法：

### 1、conf/redis.conf  里面的

bind 127.0.0.1这一行注释掉，或者bind 0.0.0.0

这里的bind指的是只有指定的网段才能远程访问这个redis。  注释掉后，就没有这个限制了。

或者bind 自己所在的网段

### 2、在etc/redis.conf里面

protected-mode 要设置成no      （默认是设置成yes的， 防止了远程访问，在redis3.2.3版本后）

vi /etc/redis.conf

### 3、我的服务器用的是firewall-cmd

firewall-cmd --zone=public --add-port=6379/tcp --permanent

firewall-cmd --reload#重启firewall

### 4.重启 Redis

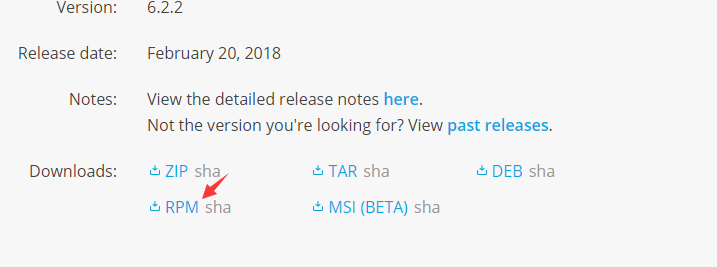
service redis restart

# ES 安装

<https://www.cnblogs.com/zhenyuyaodidiao/p/4948000.html> Es 安装

<https://www.elastic.co/cn/products/elasticsearch>





## 下载

wget <https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.2.2.rpm>

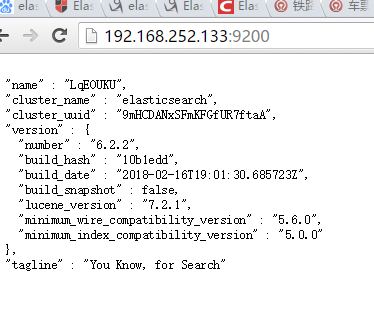
### 安装

yum install elasticsearch-6.2.2.rpm

### 修改配置==

vim /etc/elasticsearch/elasticsearch.yml

|  |
| --- |
| 取消network.host前面的注释#，并修改ip为0.0.0.0，或者本机的外网ip192.168.1.108都可以，但是0.0.0.0适用范围广。  修改配置以后要重启es,但是es的重启还有点烦，要先杀死es进程关闭，然后再次启动  <https://www.cnblogs.com/hongdada/p/7887455.html> |

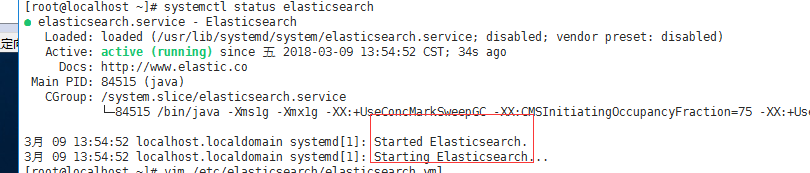


### 启动 Es

systemctl start elasticsearch

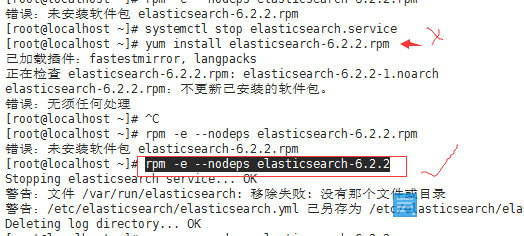
### 状态查询

systemctl status elasticsearch



### 卸载

rpm -e --nodeps elasticsearch-6.2.2

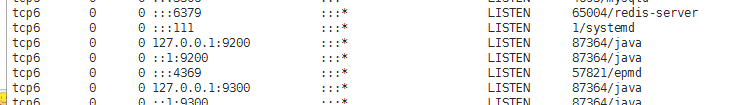


### 查询



端口监控

netstat -nltp



### 防火墙设置

|  |
| --- |
| firewall-cmd --permanent --add-port={9200/tcp,9300/tcp}  firewall-cmd --zone=public --add-port=9300/tcp –permanent  firewall-cmd --zone=public --add-port=9200/tcp --permanent |
| firewall-cmd --reload |
| firewall-cmd --list-all **（查看防火墙开发端口）** |

# Kibana 安装

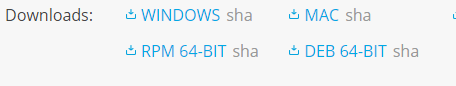
<https://www.elastic.co/cn/products>

## 下载

Wget https://artifacts.elastic.co/downloads/kibana/kibana-6.2.2-x86\_64.rpm



<https://www.elastic.co/cn/downloads/kibana>



## 安装

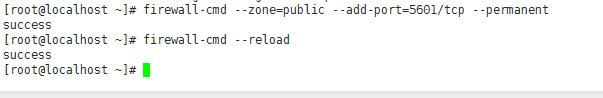
yum install kibana-6.2.2-x86\_64.rpm

## 配置

|  |
| --- |
| vim /etc/kibana/kibana.yml |
| server.port: 5601  server.host: "0.0.0.0"  elasticsearch.url: "http://192.168.252.133:9200"  kibana.index: ".kibana" |
| 重启  [root@localhost ~]# systemctl restart kibana.service  [root@localhost ~]# systemctl status kibana.service |
|  |

## 防火墙设置

|  |
| --- |
| firewall-cmd --zone=public --add-port=5601/tcp --permanent |
| firewall-cmd --reload |



## 浏览

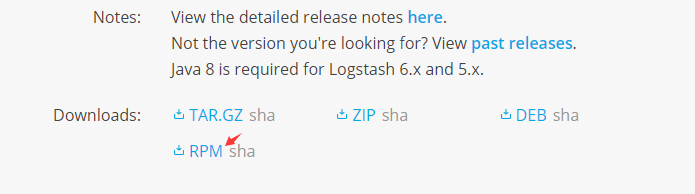
<http://192.168.252.133:5601/>

# logstash 安装

<https://www.elastic.co/cn/downloads/logstash>

## 下载

wget <https://artifacts.elastic.co/downloads/logstash/logstash-6.2.2.rpm>



# JDK 安装

## 查看yum库中的Java安装包

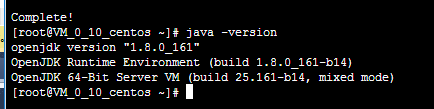
yum -y list java\*

## 使用yum安装Java环境

yum -y install java-1.8.0-openjdk\*

## 查看刚安装的Java版本信息。

◆输入：java -version 可查看Java版本；



◆输入：javac 可查看Java的编译器命令用法（可略）。

# ZooKeeper

http://blog.csdn.net/jiaodacailei/article/details/79303032

## 2.下载安装

下载到本地

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. cd /usr/local/src
2. wget http://mirrors.hust.edu.cn/apache/zookeeper/zookeeper-3.4.11/zookeeper-3.4.11.tar.gz

解压

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. tar -xzvf zookeeper-3.4.11.tar.gz

创建的安装目录，并移动解压后的目录到其下

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. mkdir -p ../zookeeper
2. mv zookeeper-3.4.11/ ../zookeeper

创建数据目录，用于存放数据

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. mkdir -p /var/lib/zookeeper

创建配置

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. cd ../zookeeper/zookeeper-3.4.11/conf
2. cp zoo\_sample.cfg zoo.cfg

修改其中配置dataDir：

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. vi zoo.cfg

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. dataDir=/var/lib/zookeeper

3.启动服务

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. cd ..
2. ./bin/zkServer.sh start

输出如下信息，表示启动成功

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. Connecting to 127.0.0.1:2181
2. 2018-02-09 14:56:10,914 [myid:] - INFO  [main:Environment@100] - Client environment:zookeeper.version=3.4.11-37e277162d567b55a07d1755f0b31c32e93c01a0, built on 11/01/2017 18:06 GMT
3. 2018-02-09 14:56:10,922 [myid:] - INFO  [main:Environment@100] - Client environment:host.name=jiaodacailei-thinkpad-centos7
4. 2018-02-09 14:56:10,922 [myid:] - INFO  [main:Environment@100] - Client environment:java.version=1.8.0\_161
5. 2018-02-09 14:56:10,927 [myid:] - INFO  [main:Environment@100] - Client environment:java.vendor=Oracle Corporation
6. 2018-02-09 14:56:10,927 [myid:] - INFO  [main:Environment@100] - Client environment:java.home=/usr/local/jdk/jdk1.8.0\_161/jre
7. 2018-02-09 14:56:10,927 [myid:] - INFO  [main:Environment@100] - Client environment:java.class.path=/usr/local/zookeeper/zookeeper-3.4.11/bin/../build/classes:/usr/local/zookeeper/zookeeper-3.4.11/bin/../build/lib/\*.jar:/usr/local/zookeeper/zookeeper-3.4.11/bin/../lib/slf4j-log4j12-1.6.1.jar:/usr/local/zookeeper/zookeeper-3.4.11/bin/../lib/slf4j-api-1.6.1.jar:/usr/local/zookeeper/zookeeper-3.4.11/bin/../lib/netty-3.10.5.Final.jar:/usr/local/zookeeper/zookeeper-3.4.11/bin/../lib/log4j-1.2.16.jar:/usr/local/zookeeper/zookeeper-3.4.11/bin/../lib/jline-0.9.94.jar:/usr/local/zookeeper/zookeeper-3.4.11/bin/../lib/audience-annotations-0.5.0.jar:/usr/local/zookeeper/zookeeper-3.4.11/bin/../zookeeper-3.4.11.jar:/usr/local/zookeeper/zookeeper-3.4.11/bin/../src/java/lib/\*.jar:/usr/local/zookeeper/zookeeper-3.4.11/bin/../conf::/usr/local/jdk/jdk1.8.0\_161/lib/dt.jar:/usr/local/jdk/jdk1.8.0\_161/lib/tools.jar:/usr/local/jdk/jdk1.8.0\_161/jre/lib/dt.jar
8. 2018-02-09 14:56:10,927 [myid:] - INFO  [main:Environment@100] - Client environment:java.library.path=/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/usr/lib
9. 2018-02-09 14:56:10,928 [myid:] - INFO  [main:Environment@100] - Client environment:java.io.tmpdir=/tmp
10. 2018-02-09 14:56:10,928 [myid:] - INFO  [main:Environment@100] - Client environment:java.compiler=<NA>
11. 2018-02-09 14:56:10,928 [myid:] - INFO  [main:Environment@100] - Client environment:os.name=Linux
12. 2018-02-09 14:56:10,928 [myid:] - INFO  [main:Environment@100] - Client environment:os.arch=amd64
13. 2018-02-09 14:56:10,928 [myid:] - INFO  [main:Environment@100] - Client environment:os.version=3.10.0-693.el7.x86\_64
14. 2018-02-09 14:56:10,929 [myid:] - INFO  [main:Environment@100] - Client environment:user.name=root
15. 2018-02-09 14:56:10,929 [myid:] - INFO  [main:Environment@100] - Client environment:user.home=/root
16. 2018-02-09 14:56:10,929 [myid:] - INFO  [main:Environment@100] - Client environment:user.dir=/usr/local/zookeeper/zookeeper-3.4.11
17. 2018-02-09 14:56:10,932 [myid:] - INFO  [main:ZooKeeper@441] - Initiating client connection, connectString=127.0.0.1:2181 sessionTimeout=30000 watcher=org.apache.zookeeper.ZooKeeperMain$MyWatcher@579bb367
18. Welcome to ZooKeeper!
19. 2018-02-09 14:56:10,983 [myid:] - INFO  [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@1035] - Opening socket connection to server 127.0.0.1/127.0.0.1:2181. Will not attempt to authenticate using SASL (unknown error)
20. JLine support is enabled
21. 2018-02-09 14:56:11,141 [myid:] - INFO  [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@877] - Socket connection established to 127.0.0.1/127.0.0.1:2181, initiating session
22. [zk: 127.0.0.1:2181(CONNECTING) 0] 2018-02-09 14:56:11,242 [myid:] - INFO  [main-SendThread(127.0.0.1:2181):ClientCnxn$SendThread@1302] - Session establishment complete on server 127.0.0.1/127.0.0.1:2181, sessionid = 0x1000150f9ff0000, negotiated timeout = 30000
24. WATCHER::
26. WatchedEvent state:SyncConnected type:None path:null

4.启动客户端

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. ./bin/zkCli.sh -server 127.0.0.1:2181

输入如下，等待用户输入命令：

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. [zk: 127.0.0.1:2181(CONNECTED) 0]

5.获取帮助

输入如下命令，可以获取帮助信息：

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. help

输出信息如下：

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. ZooKeeper -server host:port cmd args
2. stat path [watch]
3. set path data [version]
4. ls path [watch]
5. delquota [-n|-b] path
6. ls2 path [watch]
7. setAcl path acl
8. setquota -n|-b val path
9. history
10. redo cmdno
11. printwatches on|off
12. delete path [version]
13. sync path
14. listquota path
15. rmr path
16. get path [watch]
17. create [-s] [-e] path data acl
18. addauth scheme auth
19. quit
20. getAcl path
21. close
22. connect host:port

上面是所有可以使用的命令

6.ls命令

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. ls /

默认根节点下有一个zookeeper节点

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. [zookeeper]

7.创建节点

创建节点采用create命令，创建之后查看根路径，发现多了一个节点；

通过get命令，可以查询节点路径对应的数据

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. [zk: 127.0.0.1:2181(CONNECTED) 4] create /zk\_test test\_data
2. Created /zk\_test

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. [zk: 127.0.0.1:2181(CONNECTED) 6] ls /
2. [zookeeper, zk\_test]

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. [zk: 127.0.0.1:2181(CONNECTED) 7] get /zk\_test
2. test\_data
3. cZxid = 0x2
4. ctime = Fri Feb 09 15:19:11 CST 2018
5. mZxid = 0x2
6. mtime = Fri Feb 09 15:19:11 CST 2018
7. pZxid = 0x2
8. cversion = 0
9. dataVersion = 0
10. aclVersion = 0
11. ephemeralOwner = 0x0
12. dataLength = 9
13. numChildren = 0

8.修改节点数据

set命令可以修改路径对应的节点数据，修改后再通过get命令查看结果

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. [zk: 127.0.0.1:2181(CONNECTED) 8] set /zk\_test test\_data\_change

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. cZxid = 0x2
2. ctime = Fri Feb 09 15:19:11 CST 2018
3. mZxid = 0x3
4. mtime = Fri Feb 09 15:21:31 CST 2018
5. pZxid = 0x2
6. cversion = 0
7. dataVersion = 1
8. aclVersion = 0
9. ephemeralOwner = 0x0
10. dataLength = 16
11. numChildren = 0

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. [zk: 127.0.0.1:2181(CONNECTED) 10] get /zk\_test

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. test\_data\_change
2. cZxid = 0x2
3. ctime = Fri Feb 09 15:19:11 CST 2018
4. mZxid = 0x3
5. mtime = Fri Feb 09 15:21:31 CST 2018
6. pZxid = 0x2
7. cversion = 0
8. dataVersion = 1
9. aclVersion = 0
10. ephemeralOwner = 0x0
11. dataLength = 16
12. numChildren = 0

9.删除节点

delete命令可以删除路径对应的节点，删除之后，通过ls命令查看其父路径，可以看到节点已经减少一个

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. [zk: 127.0.0.1:2181(CONNECTED) 11] delete /zk\_test
2. [zk: 127.0.0.1:2181(CONNECTED) 12] ls /
3. [zookeeper]

10.退出客户端

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. quit

11.停止服务

**[plain]** [view plain](http://blog.csdn.net/jiaodacailei/article/details/79303032) [copy](http://blog.csdn.net/jiaodacailei/article/details/79303032)

1. ./bin/zkServer.sh stop

# CentOs7使用yum源安装jdk和tomcat

## jdk安装

查询yum源支持的jdk的rpm包

yum list | grep jdk

* 1

安装jdk-1.8.0版本

yum -y install java-1.8.0-openjdk\*

* 1

安装后，执行java -version

[root@localhost ~]# java -version

openjdk version "1.8.0\_144"

OpenJDK Runtime Environment (build 1.8.0\_144-b01)

OpenJDK 64-Bit Server VM (build 25.144-b01, mixed mode)

* 1
* 2
* 3
* 4

安装成功。

## tomcat安装

执行安装命令

[root@localhost ~]# yum -y install tomcat

* 1

查询tomcat是否安装成功

[root@localhost ~]# rpm -q tomcat

tomcat-7.0.76-2.el7.noarch

* 1
* 2

安装成功。

[root@localhost ~]# systemctl start tomcat.service

## 安装Tomcat

现在您已准备好安装Tomcat 7.运行以下命令安装Tomcat软件包：

sudo yum install tomcat

答y在确认提示安装Tomcat。 这将安装Tomcat 7及其相关项，比如Java，它也将创建tomcat用户。

最重要的Tomcat的文件将位于/usr/share/tomcat 。 如果你已经有了，你想运行一个Tomcat应用程序，你可以将它放在/usr/share/tomcat/webapps的目录，配置Tomcat，并重新启动Tomcat服务。但在本教程中，我们将安装一些其他软件包，帮助您管理Tomcat应用程序和虚拟主机。

让我们快速更改Tomcat在启动时使用的Java选项。打开Tomcat配置文件：

sudo vi /usr/share/tomcat/conf/tomcat.conf

下面添加JAVA\_OPTS行添加到文件。 随意改变Xmx和MaxPermSize值，这些设置会影响Tomcat会使用多少内存：

/ etc / default / tomcat7 – JAVA\_OPTS

JAVA\_OPTS="-Djava.security.egd=file:/dev/./urandom -Djava.awt.headless=true -Xmx512m -XX:MaxPermSize=256m -XX:+UseConcMarkSweepGC"

保存并退出。

请注意，Tomcat服务将不会运行。

## 安装管理包

如果您刚开始使用Apache Tomcat，您很可能想要安装一些管理工具，这些工具将帮助您部署Java应用程序和管理虚拟主机。幸运的是，有一些软件包将这些工具作为Web应用程序。

要安装默认Tomcat根页面（tomcat-webapps）和Tomcat Web应用程序管理器和Virtual Host Manager（tomcat-admin-webapps），请运行以下命令：

sudo yum install tomcat-webapps tomcat-admin-webapps

答y在确认提示。

这增加了ROOT ， examples ， sample ， manager和host-manager Web应用到tomcat/webapps的目录。

### 安装在线文档（可选）

如果要安装Tomcat文档，以便默认Tomcat页面上的所有链接都可以运行，请运行以下命令：

sudo yum install tomcat-docs-webapp tomcat-javadoc

答y在提示安装文档软件包。

## 配置Tomcat Web管理界面

为了使用在上一步安装的manager webapp，我们必须添加一个登录到我们的Tomcat服务器。我们将通过编辑这样做tomcat-users.xml的文件：

sudo vi /usr/share/tomcat/conf/tomcat-users.xml

此文件填充有描述如何配置文件的注释。您可能需要以下字里行间删除所有的评论，或者你可能会离开他们，如果你要引用的例子：

tomcat-users.xml摘录

<tomcat-users>

...

</tomcat-users>

您将要新增谁可以访问用户manager-gui和admin-gui （前面我们安装了管理接口）。您可以通过定义类似于以下示例的用户来执行此操作。请务必将用户名和密码更改为安全：

tomcat-users.xml – 管理用户

<tomcat-users>

<user username="admin" password="password" roles="manager-gui,admin-gui"/>

</tomcat-users>

保存并退出tomcat-users.xml的文件。

现在我们准备好启动Tomcat服务。

## 启动Tomcat

要使更改生效，请重新启动Tomcat服务：

sudo systemctl start tomcat

如果由于某种原因而提前启动服务，请改用以下命令运行restart命令：

sudo systemctl restart tomcat

### 启用Tomcat服务

如果您希望Tomcat在每次启动服务器时运行，则需要启用服务：

sudo systemctl enable tomcat

现在我们准备好访问Web界面。

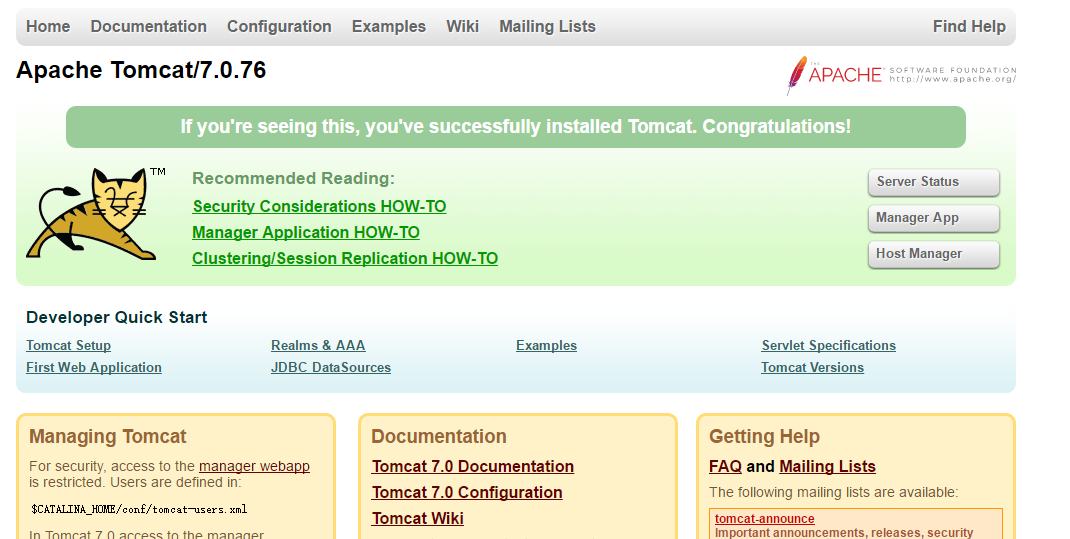
## 访问Web界面

现在Tomcat已经启动并运行，让我们在Web浏览器中访问Web管理界面。您可以通过访问服务器的公共IP地址，在端口8080上：

Open in web browser:http://server\_IP\_address:8080

http://111.231.209.131:8080/

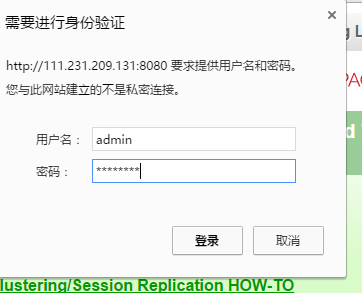
你会看到类似下面的图像：

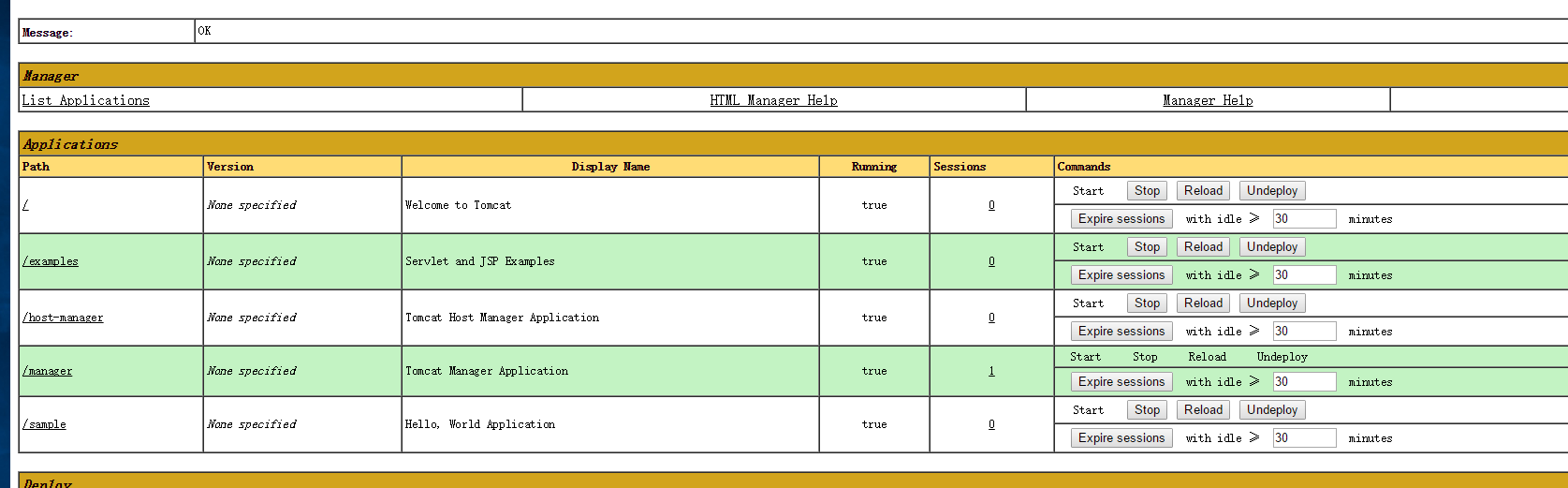


正如你可以看到，有一些链接到您之前安装的管理webapps。

让我们来看看管理器应用程序，通过该链接或访问http:// server\_IP\_address :8080/manager/html ：

http://111.231.209.131:8080/manager/html



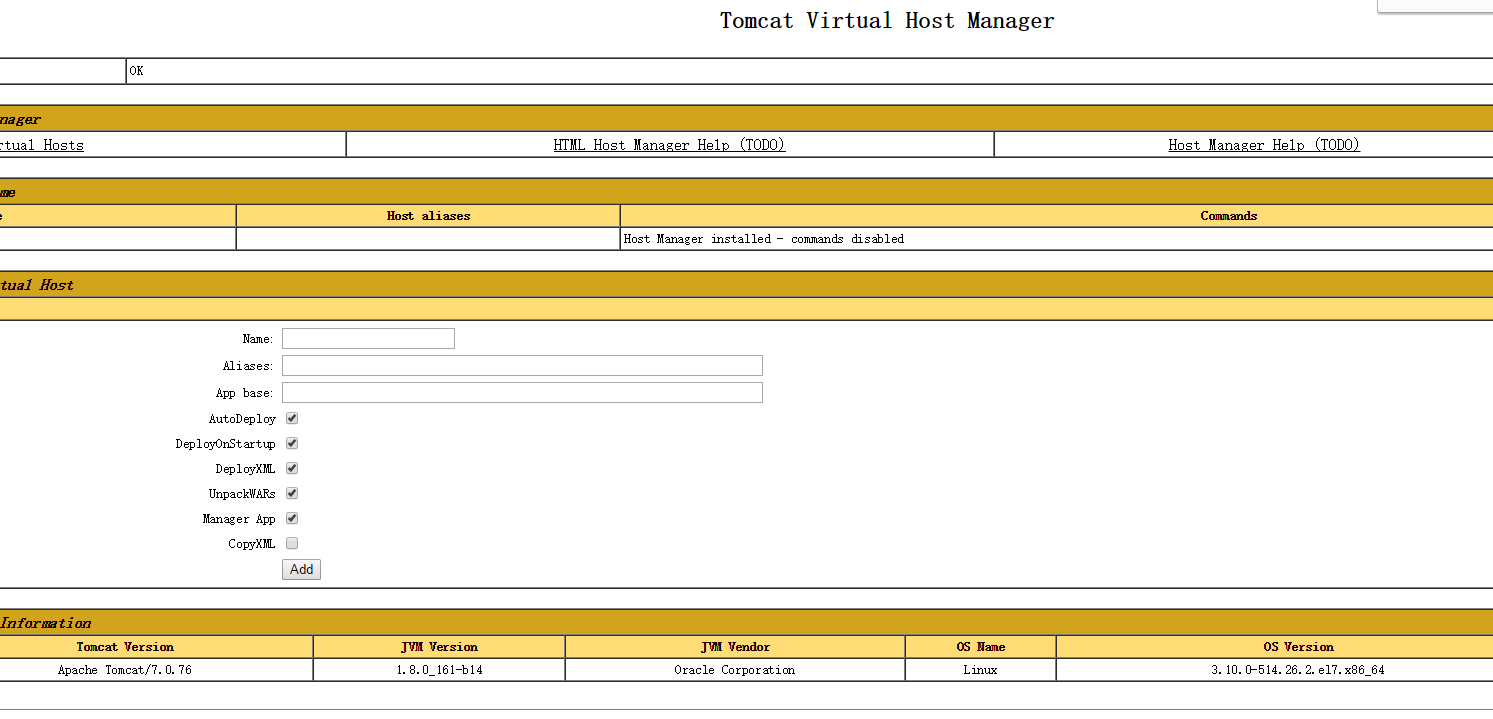


Web应用程序管理器用于管理Java应用程序。您可以在此处开始，停止，重新加载，部署和取消部署。您还可以对应用程序运行一些诊断（即查找内存泄漏）。最后，有关您的服务器的信息位于此页面的最下方。

现在，让我们来看看主机管理，通过链接或访问http:// server\_IP\_address :8080/host-manager/html/ ：

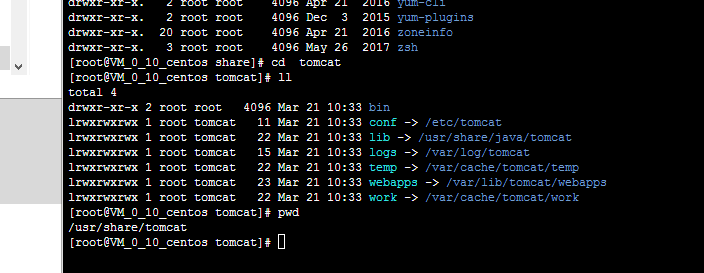
http://111.231.209.131:8080/host-manager/html/

从Virtual Host Manager页面，您可以添加要从中提供应用程序的虚拟主机。



|  |
| --- |
|  |

<http://blog.csdn.net/qq_27739989/article/details/78047106>



5，测试服务

浏览器中输入 : http://ip:8080/

检查服务器能正确响应。

或者终端命令行输入: wget htp://localhost:8080/

http:// 111.231.209.131:8080

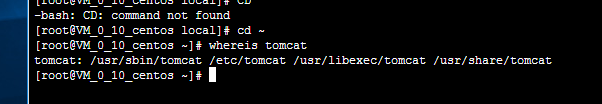
**最后需要在安全组入口方向添加所使用的端口**

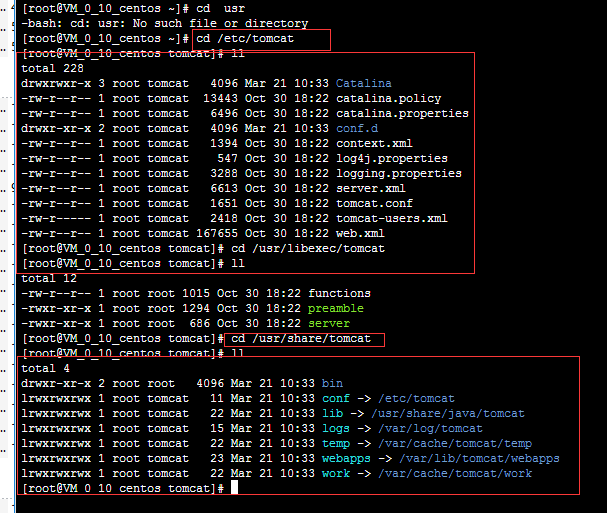
1.发现安装好tomcat后，发现无法访问tomcat首页，后来发现防火墙没有开放8080端口。

需注意的是：CentOS 7防火墙换用Friewalld了，所以要用以下命令将端口号加进防火墙：

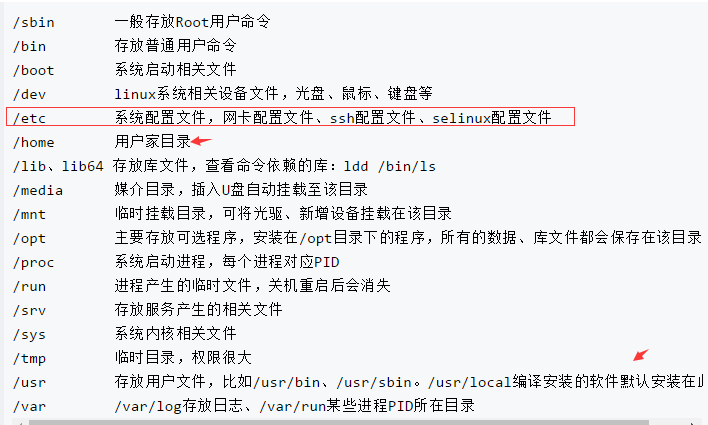
*firewall-cmd --zone=public --add-port=8080/tcp --permanent*

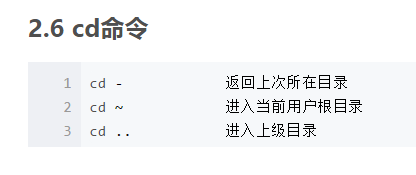
*firewall-cmd --reload*





<http://blog.csdn.net/github_39069288/article/details/72886947>





# CentOS7搭建FastDFS V5.11分布式文件系统及Java整合详细过程(一)

https://blog.csdn.net/m0\_37797991/article/details/73381648

# Vsftp

<https://www.linuxidc.com/Linux/2016-12/138639.htm>