# Linux系统下部署环境配置

编辑记录

第一版 LCTR 2020年10月14日

目录

[Linux系统下部署环境配置 1](#_Toc55206908)

[*说明* 2](#_Toc55206909)

[ **使用的Linux系统为CentOS7** 2](#_Toc55206910)

[ **请使用root用户登录系统** 2](#_Toc55206911)

[ **如果系统安装在主机上的虚拟机中，还需要配置主机的防火墙，并且配置相应的端口转发规则** 2](#_Toc55206912)

[1. **Hyper-V** 2](#_Toc55206913)

[ **参考资料** 2](#_Toc55206914)

[***1.*** CentOS系统镜像下载 2](#_Toc55206915)

[*配置* 3](#_Toc55206916)

[ **常用配置** 3](#_Toc55206917)

[ **服务常用命令** 3](#_Toc55206918)

[ **中文拼音输入法** 4](#_Toc55206919)

[ **用户管理** 5](#_Toc55206920)

[ **Firewall（防火墙）** 6](#_Toc55206921)

[ **关闭SELinux** 7](#_Toc55206922)

[ **JAVA JDK（JAVA环境）** 8](#_Toc55206923)

[ **文件操作** 9](#_Toc55206924)

[ **FTP（文件传输协议）** 11](#_Toc55206925)

[ **VNC（桌面共享）** 15](#_Toc55206926)

[ **Tomcat 8（应用服务器）** 17](#_Toc55206927)

[ **Nginx（应用服务器）** 18](#_Toc55206928)

[ **ZooKeeper（分布式应用程序协调服务）** 20](#_Toc55206929)

[ **Kafka（分布式发布订阅消息系统）** 25](#_Toc55206930)

[ **Elasticsearch（搜索服务）** 28](#_Toc55206931)

[ **.Net Core控制台应用** 31](#_Toc55206932)

## ***说明***

* **使用的Linux系统为CentOS7**
* **请使用root用户登录系统**
* **如果系统安装在主机上的虚拟机中，还需要配置主机的防火墙，并且配置相应的端口转发规则**
  1. **Hyper-V**
     1. **端口转发规则配置方法（使用powershell）**
        + **查询**

1. netsh interface portproxy show v4tov4
   * + - **新增**
2. netsh interface portproxy add v4tov4 listenaddress=192.168.1.101 listenport=5901 connectaddress=192.168.1.83 connectport=5901

* 192.168.1.101是指主机IP，5901是指主机监听端口，2.168.1.83是指虚拟机IP，5901是指虚拟机监听端口；
  + - * **移除**

1. netsh interface portproxy delete v4tov4 listenaddress=192.168.1.101 listenport=5901

* 192.168.1.101是主机IP，5901是主机监听端口；
* **参考资料**
  1. [CentOS系统镜像下载](https://www.centos.org/centos-linux/)

## ***配置***

* **常用配置**
  + **服务常用命令**
    - 设置开机自启

1. systemctl enable A.service

* A.service指服务配置文件；
  + - 启动

1. systemctl start A.service

* A.service指服务配置文件；
  + - 状态

1. systemctl status A.service

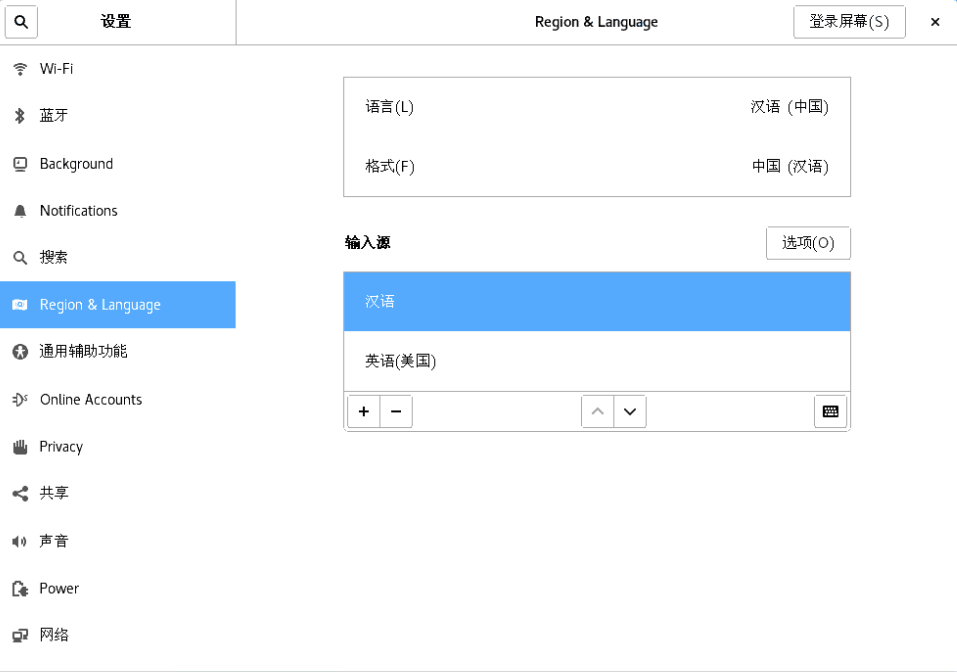
* A.service指服务配置文件；
  + - 停止

1. systemctl stop A.service

* A.service指服务配置文件；
  + - 修改服务配置文件后重新加载配置

1. systemctl daemon-reload
   * **中文拼音输入法**
     + 安装
2. yum install ibus-libpinyin

* 安装后需要重启；
* 重启后打开设置在输入源中进行添加；



* + **用户管理**
    - 切换

1. su

* 切换为root用户；

1. su user01

* 切换为user01 用户；
  + - 新增

1. adduser user01

* 新增一个名为user01用户；

1. passwd user01

* 为该用户user01置密码；
* 高级新增

1. useradd user01 -d /home/user01 -s /bin/false
2. chown -R user01:user /home/vsftpd

* 新增一个名为user01用户，指定用户目录为 /home/user01，登录终端设为不允许登录系统 /bin/false，并授权该目录；
* :user表示该用户所在的用户组，如没有则无需输入此内容；
* 新增用户组

1. groupadd group01

* 新增用户组group01；

1. useradd user01 -g group01

* 将用户user01添加至组group01；

* + **Firewall（防火墙）**
    - 开放端口

1. firewall-cmd –zone=public –add-service=http –permanent

* 注意，如果是为ftp服务开放端口，则http需改为ftp；

1. firewall-cmd –zone=public -add-port=8080/tcp –permanent
   * + 如果要移除规则，则add改为remove；
     + 8080改为想要开放的端口号，如果是多个端口，格式为8080-8090；
     + 如使用udp传输控制协议，则tcp需改为udp；
     + 重启服务
2. systemctl restart firewalld
   * + 此命令会重启防火墙，使新增的规则生效；

* + **关闭SELinux**
    - [编辑](#编辑文本文件)配置文件

1. vim /etc/selinux/config
2. # This file controls the state of SELinux on the system.
3. # SELINUX= can take one of these three values:
4. # enforcing - SELinux security policy is enforced.
5. # permissive - SELinux prints warnings instead of enforcing.
6. # disabled - No SELinux policy is loaded.
7. #SELINUX=enforcing
8. #禁用SELINUX.
9. SELINUX=disabled
10. # SELINUXTYPE= can take one of three values:
11. # targeted - Targeted processes are protected,
12. # minimum - Modification of targeted policy. Only selected processes are protected.
13. # mls - Multi Level Security protection.
14. #注释SELINUXTYPE.
15. #SELINUXTYPE=targeted

* + **JAVA JDK（JAVA环境）**
    - 检查系统是否已安装了java，如已安装，则需要执行以下命令去卸载；

1. java -version

#如出现一下信息，则表示已安装了java.

java version "15" 2020-09-15

Java(TM) SE Runtime Environment (build 15+36-1562)

Java HotSpot(TM) 64-Bit Server VM (build 15+36-1562, mixed mode, sharing)

1. rpm -qa|grep java

#进一步检测java版本，如果版本过低或者是openjdk，则需要进行卸载.

python-javapackages-3.4.1-11.el7.noarch

tzdata-java-2020a-1.el7.noarch

javapackages-tools-3.4.1-11.el7.noarch

1. yum -y remove java-\*

#卸载已安装的java.

##### [下载JDK（Linux Compressed Archive）](https://www.oracle.com/java/technologies/javase-jdk15-downloads.html)文件，[并解压](#编辑文本文件》文件解压)至自行指定的目录 /lib/jvm/java/；

* + - [删除](#编辑文本文件》删除文件)下载的文件（可选，相当于清理垃圾）
    - [编辑](#编辑文本文件)配置文件；

1. vim /etc/profile
2. #配置环境变量，在文件末尾添加以下内容，/lib/jvm/java/jdk-15指JDK文件的目录.
3. JAVA\_HOME=/lib/jvm/java/jdk-15
4. PATH=$JAVA\_HOME/bin:$PATH
5. CLASSPATH=$JAVA\_HOME/jre/lib/ext:$JAVA\_HOME/lib/tools.jar
6. export PATH JAVA\_HOME CLASSPATH

##### 使配置文件在当前终端立即生效；

1. source /etc/profile
   * **文件操作**
     + 创建目录
2. mkdir /etc/a
   * + /etc/a表示要创建的目录
     + 进入目录
3. cd /etc/a
   * + /etc/a表示要进入的目录
     + 返回上级目录
4. ../
   * + 创建&编辑文本文件
5. vim /etc/a.conf
   * + /etc/a.conf表示文件地址；

i

* + - * 切换到输入模式；

:

* + - * 切换到底线命令模式；

:set number

* + - * + 显示行号；

:wq

* + - * + 保存并退出；

:w

* + - * + 保存；

:q

* + - * + 退出；

##### 复制文件

1. cp /etc/a /etc/a.bak
   * + /etc/a表示原文件目录或地址；
     + /etc/a.bak表示新的文件目录或地址；

##### 移动文件

1. mv /etc/a /etc/b

* /etc/a表示原文件目录或地址；
* /etc/a.bak表示新的文件目录或地址；

##### 文件解压

1. tar -zxvf /etc/a.tar.gz -C /etc/a

* /etc/a.tar.gz表示要提取的文件的地址；
* /etc/a 表示提取文件的存储地址；

##### 删除文件

1. rm -rf /etc/a.tar.gz

* /etc/a.tar.gz表示文件地址；

##### 文件授权

1. chown -R 777 /etc/a

* 此语句会更改指定目录或文件的所有者；
* /home/a 表示文件地址；

1. chown -R user01 /etc/a

* 此语句递归目录下所有文件授权给指定的用户
* user01 表示被授权的用户；
* /etc/a 表示将要授权的目录或文件 ；
  + **FTP（文件传输协议）**
    - 安装

1. yum install -y vsftpd
   * + [编辑](#编辑文本文件)配置文件；
2. vim /etc/vsftpd/vsftpd.conf

#参考行号修改以下配置.

1. anonymous\_enable=NO
2. anon\_mkdir\_write\_enable=YES
3. chown\_uploads=YES
4. async\_abor\_enable=YES
5. ascii\_upload\_enable=YES
6. ascii\_download\_enable=YES
7. ftpd\_banner=Welcome to blah FTP service.
8. chroot\_local\_user=YES

#在文件末尾添加以下配置.

1. use\_localtime=YES
2. listen\_port=21

#21为ftp服务端口号,可自行更改；

1. idle\_session\_timeout=300
2. guest\_enable=YES
3. guest\_username=vsftpd
4. user\_config\_dir=/etc/vsftpd/vconf
5. data\_connection\_timeout=1
6. virtual\_use\_local\_privs=YES
7. pasv\_min\_port=40000

#40000为最小端口号,可自行更改；

1. pasv\_max\_port=40010

#40010为最大端口号,可自行更改；

1. accept\_timeout=5
2. connect\_timeout=1
3. allow\_writeable\_chroot=YES

##### [创建](#编辑文本文件》创建目录)虚拟用户文件;

1. vim /etc/vsftpd/virtusers

#输入虚拟用户用户名和密码（可自行定义）.

1. user01
2. 123456

##### 生成虚拟用户数据文件 ;

1. db\_load -T -t hash -f /etc/vsftpd/virtusers /etc/vsftpd/virtusers.db
2. chmod 600 /etc/vsftpd/virtusers.db

#设定PAM验证文件，并指定对虚拟用户数据库文件进行读取.

##### [备份](#编辑文本文件》复制文件)文件之后再[编辑](#编辑文本文件)配置文件；

1. vim /etc/pam.d/vsftpd

#注释原有auth和account注释，输入新的配置，如果为32位系统，则lib64改为lib.

1. #%PAM-1.0
2. session optional pam\_keyinit.so force revoke
3. #auth required pam\_listfile.so item=user sense=deny file=/etc/vsftpd/ftpusers
4. auth sufficient /lib64/security/pam\_userdb.so db=/etc/vsftpd/virtusers
5. onerr=succeed
6. auth required pam\_shells.so
7. auth include password-auth
8. #account include password-auth
9. account sufficient /lib64/security/pam\_userdb.so db=/etc/vsftpd/virtusers
10. session required pam\_loginuid.so
11. session include password-auth

##### [新增](#用户管理》高级新增)系统用户，用户名vsftpd，用户目录 /home/vsftpd，登录终端设为不允许登录系统 /bin/false，并授权用户目录，用户组为vsftpd（如果不存在则需要[新增](#用户管理》新增用户组)该用户组）；

##### 建立虚拟用户个人配置文件

##### [创建目录](#编辑文本文件》创建目录) /etc/vsftpd/vconf并[进入](#编辑文本文件》进入目录)；

1. touch user01

* 建立虚拟用户user01的配置文件，用户名可自行指定；

##### [编辑](#编辑文本文件)虚拟用户配置文件 user01；

1. vim user01

#输入一下内容.

1. local\_root=/home/vsftpd/user01/

#/home/vsftpd是系统用户目录.

#user01是虚拟用户用户名.

1. write\_enable=YES
2. anon\_world\_readable\_only=NO
3. anon\_upload\_enable=YES
4. anon\_mkdir)write\_enable=YES
5. anon\_other\_write\_enable=YES

##### [创建](#编辑文本文件》创建目录)虚拟用户user01的根目录，并将该目录给该虚拟用户;

1. mkdir -p /home/vsftpd/user01

#/home/vsftpd是系统用户目录.

#user01是虚拟用户用户名.

1. chown -R user01 /home/vsftpd/user01/

#/home/vsftpd是系统用户目录.

#user01是虚拟用户用户名.

##### [开放防火墙](#开放防火墙)的21，40000-40010端口，注意-server参数为ftp；

##### [关闭SELINUX；](#关闭SELinux)

##### 管理ftp服务的基本操作命令；

###### 设置开机启动；

1. systemctl enable vsftpd.service

###### 启动服务；

1. systemctl start vsftpd.service

###### 停止服务；

1. systemctl stop vsftpd.service

###### 查看状态；

1. systemctl status vsftpd.service
   * **VNC（桌面共享）**
     + 安装VNC；
2. yum install -y tigervnc-server
3. rpm -qa|grep tigervnc-server

#验证安装是否成功.

* + - 创建桌面；

1. cp /lib/systemd/system/vncserver@.service /etc/systemd/system/vncserver@:1.service

#复制服务配置，:1为桌面号，要创建多个桌面则累加即可.

#桌面号同时也决定了端口号，桌面对应的端口号为5900+N，例如:1桌面的端口为5901.

##### [编辑](#编辑文本文件)配置文件；

1. vim /etc/systemd/system/vncserver@:1.service

#修改<USER>为root（也可以是其他的系统用户）.

1. # Clean any existing files in /tmp/.X11-unix environment
2. ExecStartPre=/bin/sh -c '/usr/bin/vncserver -kill %i > /dev/null 2>&1 || :'
3. ExecStart=/usr/bin/vncserver\_wrapper root %i
4. ExecStop=/bin/sh -c '/usr/bin/vncserver -kill %i > /dev/null 2>&1 || :'
5. [Install]
6. WantedBy=multi-user.target
   * + 设置VNC密码；
7. vncpasswd
   * + 启动服务；
8. systemctl start vncserver@:1.service
   * + 重新加载服务配置文件；
9. systemctl daemon-reload
   * + 设置服务开机自动启动；
10. systemctl enable vncserver@:1.service
    * + 查看端口是否在监听；
11. netstat -lunpt|grep Xvnc

##### [开放防火墙](#开放防火墙)的5901端口（桌面对应的端口号）；

##### [关闭SELINUX；](#关闭SELinux)

##### 客户端（Windows）使用说明；

* + - 下载[VNC Viewer](https://www.realvnc.com/en/connect/download/viewer/);
    - 新建连接，VNC Server填写Ip+端口号，例如192.168.1.2:5901;
  + **Tomcat 8（应用服务器）**

##### 需要[JAVA环境](#JAVA环境)；

##### [下载](https://tomcat.apache.org/download-80.cgi)文件，并[解压](#编辑文本文件》文件解压)到想要放置该服务器的目录（如果要做集群，则复制到多个目录中）；

##### [进入文件根目录](#编辑文本文件》进入目录)中，[编辑](#编辑文本文件)配置文件；

1. vim /conf/server.xml

#修改默认的8080端口号.

1. <Connector port="8080" protocol="HTTP/1.1"
2. connectionTimeout="20000"
3. redirectPort="8443" />
4. <!-- A "Connector" using the shared thread pool-->

##### [返回文件根目录](#编辑文本文件》返回上级目录)，启动tomcat；

1. ./bin/startup.sh

##### [开放防火墙](#开放防火墙)的指定的端口；

* + **Nginx（应用服务器）**

##### [创建](#编辑文本文件)yum存储库文件；

1. vim /etc/yum.repos.d/nginx.repo

#在文件中输入以下内容.

1. [nginx-stable]
2. name=nginx stable repo
3. baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
4. gpgcheck=1
5. enabled=1
6. gpgkey=https://nginx.org/keys/nginx\_signing.key
7. module\_hotfixes=true
8. [nginx-mainline]
9. name=nginx mainline repo
10. baseurl=http://nginx.org/packages/mainline/centos/$releasever/$basearch/
11. gpgcheck=1
12. enabled=0
13. gpgkey=https://nginx.org/keys/nginx\_signing.key
14. module\_hotfixes=true

* 默认会使用稳定【stable】的存储库，如要切换至mainline的，执行以下命令；

1. sudo yum-config-manager –enable nginx-mainline

##### 安装；

1. sudo yum install nginx

* 如果提示要确认GPG Key，请验证值为【573B FD6B 3D8F BC64 1079 A6AB ABF5 BD82 7BD9 BF62】时再确认；

##### 测试配置是否正常；

1. sudo nginx -t

##### 启动；

1. systemctl start nginx

* 启动后通过访问http://localhost:80端口查看是否正常；

##### 开机启动；

1. systemctl enable nginx

##### 重启；

1. nginx -s reload

* + **ZooKeeper（分布式应用程序协调服务）**

##### 需要[JAVA环境](#JAVA环境)；

##### [下载](http://mirror.bit.edu.cn/apache/zookeeper/stable/apache-zookeeper-3.5.8-bin.tar.gz)文件，并[解压](#编辑文本文件》文件解压)到想要放置该服务的目录中；

##### [进入文件根目录](#编辑文本文件》进入目录)中，使用样例配置文件复制一个新的配置文件，之后可[编辑](#编辑文本文件)该配置文件更改默认配置；

1. cp /conf/zoo\_sample.cfg /conf/zoo.cfg
2. vim /conf/zoo.cfg
3. # The number of milliseconds of each tick
4. tickTime=2000

#设置initLimit和syncLimit配置的基础单位，2000表示2000ms.

1. # The number of ticks that the initial
2. # synchronization phase can take
3. initLimit=10

#ZooKeeper集群模式下包含多个zk进程，其中一个进程为leader，余下的进程为follower。当follower最初与leader建立连接时，它们之间会传输相当多的数据，尤其是follower的数据落后leader很多。initLimit配置follower与leader之间建立连接后进行同步的最长时间.

1. # The number of ticks that can pass between
2. # sending a request and getting an acknowledgement
3. syncLimit=5

#配置follower和leader之间发送消息，请求和应答的最大时间长度.

1. # the directory where the snapshot is stored.
2. # do not use /tmp for storage, /tmp here is just
3. # example sakes.
4. dataDir=/tmp/zookeeper

#设置数据文件存放目录,集群模式下还需要创建一个myid文件，内容只有一行，为1-255之间的一个数字，表示zk进程的id.

1. # the port at which the clients will connect
2. clientPort=2181

#设置客户端端口号.

#如果是集群模式，则配置其他服务的地址

#server.id=host :port1 :port2,id指zk进程的id，host指ip地址，port1是follower和leader交换消息所使用的端口，port2是选举leader所使用的端口.

1. server.1=192.168.1.85:2888:3888
2. server.2=192.168.1.85:2888:3888

##### 启动服务；

1. ./bin/zkServer.sh start

##### 停止服务；

1. ./bin/zkServer.sh stop

##### 查看服务状态；

1. ./bin/zkServer.sh status
2. service zookeeper status

##### 集群模式下链接某个集群，依次指定每个zk的ip地址和端口，使用逗号分隔；

1. ./bin/zkCli.sh -server 192.168.1.85:2180,192.168.1.86:2180,192.168.1.87:2180

##### [开放防火墙](#开放防火墙)的指定的端口；

##### 让其能[开机自启](#服务常用命令)（可选）；

##### 方式一：配置为服务；

1. sudo nano /etc/systemd/system/zookeeper.service
   * 命名规则可自行定义；

#在服务配置文件中写入以下内容.

1. [Unit]
2. Description=ZooKeeper 服务
3. After=syslog.target network.target
4. [Service]
5. Type=forking
6. Environment="JAVA\_HOME=/usr/lib/jvm/java/jdk-15"

#设置java jdk目录.

1. Environment="ZOO\_LOG\_DIR=/usr/local/apache/zookeeper-3.5.8-bin/logs"

#设置日志目录.

1. User=root

#使用root用户启动.

1. Group=root
2. ExecStart=/usr/local/apache/zookeeper-3.5.8-bin/bin/zkServer.sh start

#设置启动命令.

1. ExecStop=/usr/local/apache/zookeeper-3.5.8-bin/bin/zkServer.sh stop

#设置停止命令.

1. ExecReload=/usr/local/apache/zookeeper-3.5.8-bin/bin/zkServer.sh restart

#设置重启命令.

1. Restart=on-failure

#设置在失败时重启.

1. RestartSec=10

#设置重启等待时间.

1. TimeoutSec=360

#设置超时时间.

1. SyslogIdentifier=zookeeper

#设置系统日志标识.

1. [Install]
2. WantedBy=multi-user.target

##### 方式二：添加开机脚本；

1. touch /etc/rc.d/init.d/zookeeper
   * 命名规则可自行定义；

#在文件中写入以下内容.

1. #!/bin/bash
2. #chkconfig: 2345 10 90
3. #description: ZooKeeper \u670d\u52a1
4. export JAVA\_HOME=/usr/lib/jvm/java/jdk-15

#设置java jdk目录.

1. export ZOO\_LOG\_DIR=/usr/local/apache/zookeeper-3.5.8-bin/logs

#设置日志目录.

1. ZOOKEEPER\_HOME=/usr/local/apache/zookeeper-3.5.8-bin

#设置zookeeper目录.

1. su root ${ZOOKEEPER\_HOME}/bin/zkServer.sh "$1"
2. chmod +x /etc/rc.d/init.d/zookeeper

#为新建的文件添加可执行权限.

1. chkconfig --add zookeeper

#将脚本添加到开机启动项里.

1. chkconfig --list

#检查是否已添加成功.

* + 重启机器检查zookeeper的状态；
  + **Kafka（分布式发布订阅消息系统）**

##### 虽然文件中自带ZooKeeper，但还是建议自行安装[ZooKeeper](#ZooKeeper)；

##### [下载](https://downloads.apache.org/kafka/2.6.0/kafka_2.12-2.6.0.tgz)文件，并[解压](#编辑文本文件》文件解压)到想要放置该服务的目录中；

##### [进入文件根目录](#编辑文本文件》进入目录)中，[编辑](#编辑文本文件)配置文件；

1. vim /config/server.properties

#更改或添加以下配置.

1. broker.id=0

#集群模式下设置id，单机模式下直接注释即可.

1. listeners=PLAINTEXT://192.168.1.84:2161

#设置监听IP和端口号.

1. log.dirs=/tmp/kafka-logs

#设置日志文件存放目录.

1. log.retention.hours=168

#设置消息的最大持久化时间.

1. message.max.byte=5242880

#设置消息保存的最大值.

1. default.replication.factor=2

#设置保存消息的副本数，如果一个副本失效了，另一个还可以继续提供服务.

1. replica.fetch.max.bytes=5242880

#设置获取消息的最大直接数.

1. zookeeper.connect=localhost:2181

#设置zookeeper链接地址.

1. vim /config/consumer.properties

#更改或添加以下配置.

1. # consumer group id
2. group.id=test-consumer-group

#设置消费者分组标识.

1. bootstrap.servers=192.168.1.84:2161

#设置kafka的IP和端口号.

##### 启动服务；

1. ./bin/kafka-server-start.sh config/server.properties

##### 停止服务；

1. ./bin/ kafka-server-stop.sh

##### 查看服务进程；

1. jps

6313 QuorumPeerMain

#zookeeper的守护进程

10169 Kafka

#kafka的守护进程

11147 Jps

##### 创建话题（Topic）；

1. ./bin/kafka-topics.sh --create --zookeeper 192.168.1.84:2160 --replication-factor 1 --partitions 1 --topic test
   * --zookeeper 192.168.1.84:2160表示zookeeper地址，也可以使用kafka的地址--bootstrap-server 192.168.1.84:2161；
   * --replication-factor 1表示副本数量，单机指定为1即可；
   * test表示要创建的话题名称；

##### 查看所有话题（Topic）；

1. ./bin/kafka-topics.sh --list --zookeeper 192.168.1.84:2160
   * --zookeeper 192.168.1.84:2160表示zookeeper地址；

##### 查看指定话题（Topic）；

1. ./bin/kafka-topics.sh –describe –topic test --bootstrap 192.168.1.84:2161
   * --bootstrap 192.168.1.84:2161表示kafka地址；

##### 删除话题（Topic）；

1. ./bin/kafka-topics.sh --delete --zookeeper 192.168.1.84:2160 --topic test
   * --zookeeper 192.168.1.84:2160表示zookeeper地址；
   * test表示要创建的话题名称；

##### 创建生产者（Producer）；

1. ./bin/kafka-console-producer.sh --broker-list 192.168.1.84:2161 --topic test
   * 192.168.1.84:2161指kafka地址；
   * test指话题名称；

##### 创建消费者（Consumer）；

1. ./bin/kafka-console-consumer.sh --bootstrap-server 192.168.1.84:2161 -topic test --from-beginning
   * 192.168.1.84:2161指表示kafka地址；
   * test指话题名称；

##### [开放防火墙](#开放防火墙)的指定的端口；

##### 配置为服务，让其能[开机自启](#服务常用命令)（可选）；

1. sudo nano /etc/systemd/system/kafka.service
   * 命名规则可自行定义；

#在服务配置文件中写入以下内容.

1. [Unit]
2. Description=Kafka 服务
3. After=zookeeper.service

#设置在zookeeper服务启动后再启动.

1. [Service]
2. Type=simple
3. Environment="JAVA\_HOME=/usr/lib/jvm/java/jdk-15"

#设置java jdk目录.

1. User=root

#使用root用户启动.

1. Group=root
2. ExecStart=/usr/local/apache/kafka\_2.12-2.6.0/bin/kafka-server-start.sh /usr/local/apache/kafka\_2.12-2.6.0/config/server.properties

#设置启动命令.

1. ExecStop=/usr/local/apache/kafka\_2.12-2.6.0/bin/kafka-server-stop.sh

#设置停止命令.

1. Restart=on-failure

#设置在失败时重启.

1. RestartSec=10

#设置重启等待时间.

1. TimeoutSec=360

#设置超时时间.

1. SyslogIdentifier=kafka

#设置系统日志标识.

1. [Install]
2. WantedBy=multi-user.target
   * **Elasticsearch（搜索服务）**

##### 需要[JAVA环境](#JAVA环境)；

##### [下载](https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.9.3-linux-x86_64.tar.gz)文件，并[解压](#编辑文本文件》文件解压)到想要放置该服务的目录中；

##### [进入文件根目录](#编辑文本文件》进入目录)中，[编辑](#编辑文本文件)配置文件；

1. vim /config/jvm.options

#修改默认的内存值，根据主机配置设置.

1. # Xms represents the initial size of total heap space
2. # Xmx represents the maximum size of total heap space
3. -Xms512m
4. -Xmx512m

##### [编辑](#编辑文本文件)配置文件；

1. vim /etc/sysctl.conf

#修改默认的线程数.

1. vm.max\_map\_count=655360

##### 使配置文件立即生效；

1. sysctl -p

##### [编辑](#编辑文本文件)配置文件；

1. vim /etc/security/limits.conf

#修改默认的资源参数.

1. # End of file
2. \* soft nofile 65536
3. \* hard nofile 131072
4. \* soft nproc 65536
5. \* hard nproc 131072

##### 由于es不能使用root用户启动，所以需要指定其他系统用户，建议[新增](#用户管理》高级新增)一个系统用户es001，并将其[加入用户组](#用户管理》新增用户组)esuser，并将es所在目录[授权](#编辑文本文件》文件授权)给该系统用户；

##### [编辑](#编辑文本文件)配置文件；

1. vim /etc/security/limits.d/20-nproc.conf

#设置系统用户资源参数.

1. # Default limit for number of user's processes to prevent
2. # accidental fork bombs.
3. # See rhbz #432903 for reasoning.
4. \* soft nproc 4096
5. root soft nproc unlimited
6. es001 soft nproc 65536

#es001是系统用户用户名.

1. vim /config/elasticsearch.yml

#修改以下配置.

1. node.name: node-1
2. path.data: /es /data

#修改默认数据存储地址.（可自行指定）

1. path.logs: /es/logs

#修改默认日志存储地址.（可自行指定）

1. network.host: 0.0.0.0

#允许外网访问

1. http.port: 9170

#修改默认端口号.（可自行指定）

1. cluster.initial\_master\_nodes: ["node-1"]

##### [进入文件根目录](#编辑文本文件》进入目录)中，[切换系统用户](#用户管理》切换)es001并启动服务；

1. ./bin/elasticsearch

* 如启动失败并提示max file descriptors [4096]，则为系统用户的资源参数未正确配置或未生效，请按上述的步骤重新设置；
* 查看当前用户资源参数；

1. ulimit -Sn

#输出65536则表示设置成功.

1. ulimit -Hu

#输出131072则表示设置成功.

##### [开放防火墙](#开放防火墙)的指定的端口；

* + **.Net Core控制台应用**

##### 发布应用时的配置，选择独立部署模式，可以不用单独安装.NetCore；

###### 

##### 进入目录，启动程序；

1. ./Integrate\_Api

* Integrate\_Api指应用名
* 如提示权限不够，则[授权](#编辑文本文件》文件授权)该目录；

##### [开放防火墙](#开放防火墙)的指定的端口；

##### 配置成服务并[开机自启](#服务常用命令)；

1. sudo nano /etc/systemd/system/dotNetCoreApp@:1.service

* 命名规则可自行定义；

#在服务配置文件中写入以下内容.

1. [Unit]
2. Description=【这里填写简单描述信息】
3. [Service]
4. WorkingDirectory=【这里填写应用程序根目录】
5. ExecStart=【这里填写项目启动文件路径】

#如使用dotnet启动则填写/use/bin/dotnet 【应用程序根目录】【项目.dll】.

1. Restart=always

#出错后总是重启.

1. RestartSec=10

#重启等待时间.

1. SyslogIdentifier= Test
2. User=root

#系统用户.

1. Environment=ASPNETCORE\_ENVIRONMENT=Production
2. Environment=DOTNET\_PRINT\_TELEMETRY\_MESSAGE=false
3. [Install]
4. WantedBy=multi-user.target

#服务所在的连接地址.

##### 如使用Nginx反向代理应用，则[编辑](#编辑文本文件)配置文件；

1. vim /etc/nginx/conf.d/default.conf

#在配置文件中添加新的服务器配置.

1. server {
2. listen 8005;

#8005表示监听的端口.

1. location / {
2. proxy\_pass <http://localhost:5001>;

#指定要代理的地址.

1. proxy\_http\_version 1.1;
2. proxy\_set\_header Upgrade $http\_upgrade;
3. proxy\_set\_header Connection keep-alive;
4. proxy\_set\_header Host $http\_host;
5. proxy\_cache\_bypass $http\_upgrade;
6. }
7. }

* [测试配置](#nginx》测试配置)正常无误后重启nginx；
* [开放防火墙](#开放防火墙)指定的端口；