1. 环境软件要求

|  |  |  |
| --- | --- | --- |
| 服务器系统 | Linux |  |
| JDK | 1.8 | pdp-cloud/software/jdk-8u221-linux-x64.tar.gz |
| Mysql | 5.7以上 | pdp-cloud/software/mysql-5.7.25-linux-glibc2.12-x86\_64.tar.gz |
| Maven | 3.6 | pdp-cloud/software/apache-maven-3.6.2-bin.tar.gz |
| Redis | 5 | pdp-cloud/software/redis-5.0.5.tar.gz |
| RocketMQ | 4.9 | pdp-cloud/software/rocketmq-all-4.9.3-bin-release.zip  pdp-cloud/software/rocketmq-console-ng-1.0.1.jar |
| Minio | 最新 | pdp-cloud/software/minio |
| Nginx | 1.17.9 | pdp-cloud/software/nginx-1.17.9.tar.gz |

1. 中间件安装
2. JDK安装

|  |
| --- |
| 1、检查一下系统中的jdk版本  java -version  返回：  openjdk version "1.8.0\_102"  OpenJDK Runtime Environment (build 1.8.0\_102-b14)  OpenJDK 64-Bit Server VM (build 25.102-b14, mixed mode)  2、检测jdk安装包  rpm -qa | grep java  返回：  java-1.7.0-openjdk-1.7.0.111-2.6.7.8.el7.x86\_64  python-javapackages-3.4.1-11.el7.noarch  tzdata-java-2016g-2.el7.noarch  javapackages-tools-3.4.1-11.el7.noarch  java-1.8.0-openjdk-1.8.0.102-4.b14.el7.x86\_64  java-1.8.0-openjdk-headless-1.8.0.102-4.b14.el7.x86\_64  java-1.7.0-openjdk-headless-1.7.0.111-2.6.7.8.el7.x86\_64  3、卸载openjdk  rpm -e --nodeps tzdata-java-2016g-2.el7.noarch  rpm -e --nodeps java-1.7.0-openjdk-1.7.0.111-2.6.7.8.el7.x86\_64  rpm -e --nodeps java-1.7.0-openjdk-headless-1.7.0.111-2.6.7.8.el7.x86\_64  rpm -e --nodeps java-1.8.0-openjdk-1.8.0.102-4.b14.el7.x86\_64  rpm -e --nodeps java-1.8.0-openjdk-headless-1.8.0.102-4.b14.el7.x86\_64  查看卸载情况：  rpm -qa | grep java  返回：  python-javapackages-3.4.1-11.el7.noarch  javapackages-tools-3.4.1-11.el7.noarch  4、安装新的jdk  上传  jdk-8u221-linux-x64.tar.gz  创建文件夹  mkdir -p /usr/lib/jvm  解压进文件夹  tar -zxvf jdk-8u221-linux-x64.tar.gz -C /usr/lib/jvm  5、设置环境变量  vim /etc/profile  添加：  export JAVA\_HOME=/usr/lib/jvm/jdk1.8.0\_221  export JRE\_HOME=${JAVA\_HOME}/jre  export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib  export PATH=${JAVA\_HOME}/bin:$PATH  6、执行profile文件  source /etc/profile  这样可以使配置不用重启即可立即生效  7、检查新安装的jdk  java -version  返回：  java version "1.8.0\_221"  Java(TM) SE Runtime Environment (build 1.8.0\_221-b11)  Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode) |

1. Maven安装

|  |
| --- |
| 1.先上传Maven软件  解压：  tar -xzf apache-maven-3.6.2-bin.tar.gz  创建目录:  mkdir -p /opt/maven  移动文件:  mv apache-maven-3.6.2/\* /opt/maven  2.配置环境变量  vim /etc/profile  添加：  export MAVEN\_HOME=/opt/maven  export PATH=$PATH:$JAVA\_HOME/bin:$MAVEN\_HOME/bin  配置生效：  source /etc/profile  查找Maven版本：  mvn -v  3.maven配置  创建本地仓库目录  mkdir -p /opt/repo  修改Maven的settings.xmls  vim /opt/maven/conf/settings.xml  添加仓库：  <localRepository>/opt/repo</localRepository>  添加阿里云私服地址：  <mirror>  <id>alimaven</id>  <mirrorOf>central</mirrorOf>  <name>aliyun maven</name>  <url>http://maven.aliyun.com/nexus/content/groups/public/</url>  </mirror> |

1. Mysql安装

|  |
| --- |
| 1.安装  cd /opt  上传安装包  解压  tar -xzf mysql-5.7.25-linux-glibc2.12-x86\_64.tar.gz  重命名  mv mysql-5.7.25-linux-glibc2.12-x86\_64 mysql  创建data目录  cd /opt/mysql  mkdir data  创建用户组以及用户和密码  groupadd mysql  useradd -g mysql mysql  授权用户  chown -R mysql.mysql /opt/mysql  切换到bin目录下  cd bin  初始化数据库  ./mysqld --user=mysql --basedir=/opt/mysql --datadir=/opt/mysql/data/ --initialize --lower-case-table-names=1  初始化命令执行后，在最后会显示初始密码，记得一定要先保存下来，第一次登录需要使用。  初始化参数说明：  【--user=mysql】：安装mysql,用户身份是mysql用户组。  【--basedir=/usr/local/mysql】：指定了安装 MySQL 的安装路径。  【--datadir=/usr/local/mysql/data/】：指定了 MySQL 的数据库文件放在什么路径下。  【--initialize】：初始化。  【--lower-case-table-names=1】：是否数据目录所在的文件系统对文件名的大小写敏感。0-大小写敏感；1-大小写不敏感；2-大小写不敏感  2.编辑my.cnf文件  vim /etc/my.cnf  修改：全部清空  [mysql]  #字符集  default-character-set = utf8mb4  [mysqld]  #端口  port = 3306  basedir=/opt/mysql  datadir=/opt/mysql/data  socket=/opt/mysql/mysql.sock  character-set-server=UTF8MB4  #默认引擎  default-storage-engine=INNODB  lower\_case\_table\_names = 1  #最大连接数  max\_connections=2000  # Disabling symbolic-links is recommended to prevent assorted security risks  symbolic-links=0  # Settings user and group are ignored when systemd is used.  # If you need to run mysqld under a different user or group,  # customize your systemd unit file for mariadb according to the  # instructions in http://fedoraproject.org/wiki/Systemd    # [mysqld\_safe]  log-error=/opt/mysql/data/mysqld-error.log  pid-file=/opt/mysql/data/mysqld.pid  3.添加mysql服务到系统服务中  cd /opt/mysql  cp -a ./support-files/mysql.server /etc/init.d/mysql  授权以及添加服务  chmod +x /etc/init.d/mysql  chkconfig --add mysql  4.启动mysql  service mysql start  停止  service mysql stop  查看启动状态  service mysql status  将mysql命令添加到服务  ln -s /opt/mysql/bin/mysql /usr/bin  5.设置密码  登录，mysql密码，使用之前初始密码  mysql -uroot -p  回车 输入初始密码  [报错: Cannot connect to local MYSQL server through socket '/tmp/mysql.sock'(2)]  [解决:设置软连接: ln -s /opt/mysql/mysql.sock /tmp/mysql.sock]  重新登陆连接  修改【root】用户的密码  ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql\_native\_password BY '!QAZ1qaz';  执行【flush privileges;】使密码立即生效  flush privileges;  6.设置远程访问  选择mysql数据库  use mysql;  报错:  Reading table information for completion of table and column names  You can turn off this feature to get a quicker startup with -A  修改远程连接并立即生效  update user set host='%' where user='root';  flush privileges;  退出连接  exit |

1. Redis安装

|  |
| --- |
| 1.安装  cd /opt  解压  tar -zvxf redis-5.0.5.tar.gz  更改文件夹名称为redis  mv redis-5.0.5 redis  编译  cd /opt/redis  make  安装  make install PREFIX=/opt/redis  prefix这个关键字的作用是编译的时候用于指定程序存放的路径。  2.启动redis  此时未修改redis任何配置文件，默认连接redis密码为空。  采取后台进程方式启动  cd /opt/redis  ./bin/redis-server& ./redis.conf  查看Redis是否正在运行  (1)采取查看进程方式  ps -aux|grep redis  (2)采取端口监听查看方式  netstat -lanp|grep 6379  3.redis配置  修改配置文件  cd /opt/redis  vim redis.conf  操作：  1).注释掉bind 127.0.0.1  #如果放开bind 127.0.0.1即只有本机可以访问，如果注释掉，所有环境都可以访问。生产等比较重要的环境还是建议绑定访问的ip  2).保护模式修改：protected-mode的值默认为yes，将其修改为no  #该模式控制外部网是否可以连接redis服务，默认是yes,所以默认我们外网是无法访问的，如需外网连接rendis服务则需要将此属性改为no  3).守护进程修改：daemonize的值默认为no，将其修改为yes  #yes表示启用守护进程，默认是no即不以守护进程方式运行  4).redis密码修改:requirepass  #设置 redis 连接密码:witroot  5).修改访问端口:port=19003  4.重启  ps -ef|grep redis  kill -9 进程号  cd /opt/redis  ./bin/redis-server ./redis.conf  因设置了守护进程，此操作同采取后台进程方式  5.验证  cd /opt/redis/bin  # ip 端口 密码  ./redis-cli -h 192.168.196.xx -p 19003 -a witroot  [redis默认16个库,单节点需要指定数据库，数据库之间隔离不共享，切换：select 数字]  停止  ps aux | grep redis | grep -v grep | awk '{print $2}' | xargs kill -9 |

1. Minio安装

|  |
| --- |
| 1.安装  cd /opt  mkdir minio  cd minio  将minio文件上传进文件夹  #创建配置文件夹  mkdir config  # 创建data文件夹  mkdir data  #赋予文件执行权限  chmod +x minio  2.脚本设置账户密码  #脚本  #创建脚本文件  vim /opt/minio/minio.sh  #添加：  #!/bin/bash  export MINIO\_ROOT\_USER=root  export MINIO\_ROOT\_PASSWORD=witrootroot  nohup ./minio server --address ":9000" --console-address ":19002" --config-dir ./config ./data > ./minio.log 2>&1 &  [MINIO\_ROOT\_USER:账号]  [MINIO\_ROOT\_PASSWORD:密码]  #给这个文件授予权限  chmod 777 minio.sh  #启动  sh minio.sh  #查看  ps -ef | grep minio  #关闭进程可以使用：  使用kill -9 进程id号，即可关闭进程  #测试访问：  http://ip:19002   1. 使用浏览器访问并创建桶:   Create bucket -> pdp  设置隐私：Manage->Summary->Access Policy->public      4.前端直接根据fileKey预览图片  前提：将桶的权限[Summary -> Access Policy]设置成：public  使用：公网IP+内网端口[9000]/fileKey |

1. RocketMQ安装

|  |
| --- |
| 1.配置环境  推荐64位操作系统，Linux/Unix/Mac；  64位JDK 1.8+；  Maven 3.2.x；  2.安装  #解压  unzip rocketmq-all-4.9.3-bin-release.zip  #移动  mv rocketmq-4.9.3 rocketmq  目录结构：  benchmark：这里面就是一些可以直接运行，进行测试的 sh 文件  bean：该目录是比较常用的，里面是一些可执行文件  conf：这里面就是一些配置文件  lib：就是 mq 所依赖的第三方 jar 包  3.配置JVM参数  RocketMQ Broker 默认配置的 jvm 虚拟机内存比较大，运行时报错误内存不足，可以修改 bin 目录下的 runserver.sh和runbroker.sh 文件，将分配内存改小  vim bin/runbroker.sh    vim bin/runserver.sh    #都改成  -server -Xms256m -Xmx256m -Xmn256m  4.设置环境变量  vim /etc/profile  #添加  export NAMESRV\_ADDR=服务器IP:9876  #重新加载环境变量  source /etc/profile  5.启动  cd /opt/rocketmq  mkdir log  nohup bin/mqnamesrv > log/mqname.log 2>&1 &  #开启自动创建topic  nohup bin/mqbroker -n 服务器地址:9876 -c conf/broker.conf autoCreateTopicEnable=true > log/borker.log 2>&1 &  #若启动失败，日志中出现：  #nohup: failed to run command ‘bin/mqbroker’: Permission denied  #则对当前总目录进行授权：  sudo chmod -R xxx #某一目录  6.查看进程  jps  #出现两个  17239 BrokerStartup  17150 NameservStartup  #关闭 Broker  sh bin/mqshutdown broker  #关闭 NameServer  sh bin/mqshutdown namesrv  7.模拟发送消息 和 消费消息  目前 我们 NameServer 和 Broker 都已经启动成功，就可以进行发送和消费消息，  RocketMQ 安装包默认提供了一个模拟发送消息和消费消息的测试类，我们来验证一下：  #创建生产者发送消息  sh bin/tools.sh org.apache.rocketmq.example.quickstart.Producer  #消费消息  sh bin/tools.sh org.apache.rocketmq.example.quickstart.Consumer   1. 安装可视化界面   mkdir -p /opt/rocketmq-console  mkdir -p /opt/rocketmq-console/data  mkdir -p /opt/rocketmq-console/log  #上传并运行jar包  cd /opt/rocketmq-console  #添加执行权限  chmod +777 rocketmq-console-ng-1.0.1.jar  #启动并设置端口  nohup java -jar rocketmq-console-ng-1.0.1.jar --server.port=19004 >> /opt/rocketmq-console/log/out.log 2>&1 &  运行成功在浏览器输入 http://ip:19004 即可看到运行界面，账号密码: admin/admin |

1. Linux开机启动

|  |
| --- |
| 1.编写sh文件  mkdir -p /opt/start  cd /opt/start  vim start.sh  编写：[必须以：#!/bin/bash 开头]    2.对sh文件授权  chmod +x /opt/start/start.sh  或  chmod +777 /opt/start/start.sh  3.修改rc.local  vim /etc/rc.d/rc.local  添加脚本：  source /opt/startFile/start.sh  4.对rc.local授权  chmod +777 rc.local |

1. 配置防火墙

查看：

systemctl status firewalld

关闭：

systemctl stop firewalld

systemctl disable firewalld

开放端口: 8088

firewall-cmd --zone=public --add-port=8088/tcp --permanent

关闭端口

firewall-cmd --zone=public --remove-port=8088/tcp --permanent

开放或关闭端口后，需要刷新

firewall-cmd --reload