中企酷客 v2.0

系统安装指南

中企动力科技股份有限公司

目 录

[1 产品概述 1-1](#_Toc505847654)

[1.1 产品简介 1-1](#_Toc505847655)

[1.2 系统架构 1-1](#_Toc505847656)

[1.3 逻辑架构 1-1](#_Toc505847657)

[2 安装概述 2-2](#_Toc505847658)

[2.1 部署目标 2-2](#_Toc505847659)

[2.2 部署结构 2-2](#_Toc505847660)

[3 软硬件需求 3-1](#_Toc505847661)

[3.1 服务器需求 3-1](#_Toc505847662)

[3.2 存储需求 3-1](#_Toc505847663)

[3.3 网络需求 3-1](#_Toc505847664)

[3.3.1 NAT网关需求 3-1](#_Toc505847665)

[3.3.2 外网IP及带宽需求 3-1](#_Toc505847666)

[3.3.3 网络端口需求 3-2](#_Toc505847667)

[3.4 操作系统需求 3-2](#_Toc505847668)

[3.5 基础软件需求 3-3](#_Toc505847669)

[3.6 外部服务需求 3-3](#_Toc505847670)

[4 安装配置说明 4-1](#_Toc505847671)

[4.1 系统安装 4-1](#_Toc505847672)

[4.1.1 安装部署规划 4-1](#_Toc505847673)

[4.2 系统配置 4-1](#_Toc505847674)

[5 操作系统配置 5-1](#_Toc505847675)

[5.1 组件升级 5-1](#_Toc505847676)

[5.1.1 Ncat安装 5-1](#_Toc505847677)

[5.2 内核调整 5-1](#_Toc505847678)

[5.3 网络设置 5-1](#_Toc505847679)

[5.3.1 DNS配置 5-1](#_Toc505847680)

[5.3.2 防火墙配置 5-2](#_Toc505847681)

[5.4 时间设置 5-2](#_Toc505847682)

[5.4.1 服务器时间同步 5-2](#_Toc505847683)

[5.5 用户设置 5-2](#_Toc505847684)

[5.5.1 创建zceo用户 5-2](#_Toc505847685)

[5.5.2 创建应用安装目录 5-2](#_Toc505847686)

[5.5.3 设置本地时间 5-3](#_Toc505847687)

[6 基础组件安装 6-1](#_Toc505847688)

[6.1 安装JDK 6-1](#_Toc505847689)

[6.2 安装ZooKeeper集群 6-2](#_Toc505847690)

[6.2.1 下载安装包 6-2](#_Toc505847691)

[6.2.2 上传安装包 6-2](#_Toc505847692)

[6.2.3 安装ZooKeeper 6-2](#_Toc505847693)

[6.2.4 参数配置 6-2](#_Toc505847694)

[6.2.5 安装验证 6-4](#_Toc505847695)

[7 数据库服务安装 7-1](#_Toc505847696)

[7.1 安装概述 7-1](#_Toc505847697)

[7.1.1 简介 7-1](#_Toc505847698)

[7.1.2 部署架构 7-1](#_Toc505847699)

[7.1.3 安装流程 7-1](#_Toc505847700)

[7.2 MySQL安装配置 7-1](#_Toc505847701)

[7.3 Mycat安装配置 7-1](#_Toc505847702)

[7.3.2 上传安装包 7-1](#_Toc505847703)

[7.3.3 安装Mycat 7-2](#_Toc505847704)

[7.3.4 配置逻辑数据库 7-2](#_Toc505847705)

[7.3.5 配置ZooKeeper相关信息 7-3](#_Toc505847706)

[7.3.6 启动Mycat集群 7-3](#_Toc505847707)

[7.3.7 验证Mycat集群部署状态 7-3](#_Toc505847708)

[7.4 PostgreSQL安装配置 7-4](#_Toc505847709)

[7.5 Pgpool-II安装配置 7-4](#_Toc505847710)

[7.6 配置HAProxy集群 7-4](#_Toc505847711)

[8 分布式缓存服务安装 8-1](#_Toc505847712)

[8.1 安装概述 8-1](#_Toc505847713)

[8.1.1 8-1](#_Toc505847714)

[8.2 安装过程 8-1](#_Toc505847715)

[8.2.1 下载源代码包 8-1](#_Toc505847716)

[8.2.2 编译源码 8-1](#_Toc505847717)

[8.2.3 配置Redis Cluster 8-2](#_Toc505847718)

[8.2.4 启动Redis Cluster 8-3](#_Toc505847719)

[8.2.5 节点握手配置 8-3](#_Toc505847720)

[8.2.6 分配哈希槽 8-4](#_Toc505847721)

[8.2.7 配置Slave节点 8-4](#_Toc505847722)

[9 消息队列服务安装 9-1](#_Toc505847723)

[9.1 安装说明 9-1](#_Toc505847724)

[9.1.1 安装概述 9-1](#_Toc505847725)

[9.1.2 部署架构 9-1](#_Toc505847726)

[9.1.3 安装流程 9-1](#_Toc505847727)

[9.2 安装过程 9-1](#_Toc505847728)

[9.2.1 下载安装包 9-1](#_Toc505847729)

[9.2.2 上传安装包 9-2](#_Toc505847730)

[9.2.3 安装RocketMQ 9-2](#_Toc505847731)

[9.2.4 配置NameServer 9-2](#_Toc505847732)

[9.2.5 配置Broker 9-3](#_Toc505847733)

[9.3 初始化Topic 9-4](#_Toc505847734)

[10 服务治理框架安装 10-1](#_Toc505847735)

[10.1 安装概述 10-1](#_Toc505847736)

[10.1.1 10-1](#_Toc505847737)

[10.2 安装准备 10-1](#_Toc505847738)

[10.3 安装过程 10-1](#_Toc505847739)

[10.3.1 JVM设置 10-1](#_Toc505847740)

[10.3.2 Dubbo-Admin安装 10-2](#_Toc505847741)

[11 服务网关安装 11-1](#_Toc505847742)

[11.1 安装概述 11-1](#_Toc505847743)

[11.1.1 部署架构 11-1](#_Toc505847744)

[11.1.2 安装流程 11-1](#_Toc505847745)

[11.2 安装过程 11-1](#_Toc505847746)

[11.2.1 上传安装包 11-1](#_Toc505847747)

[11.2.2 安装Kong 11-1](#_Toc505847748)

[11.2.3 配置Kong 11-2](#_Toc505847749)

[11.2.4 配置环境变量 11-2](#_Toc505847750)

[11.2.5 启动服务 11-2](#_Toc505847751)

[12 反向代理安装 12-1](#_Toc505847752)

[12.1 12-1](#_Toc505847753)

[12.1.1 12-1](#_Toc505847754)

插图目录

[图1-1 系统整体架构 1-1](#_Toc503168445)

[图1-2 系统逻辑架构 1-1](#_Toc503168446)

[图2-1 系统部署结构 2-2](#_Toc503168447)

表格目录

[表3-1 虚拟机资源需求 3-1](#_Toc503168453)

[表3-2 NAT网关需求 3-2](#_Toc503168454)

[表3-3 带宽需求 3-2](#_Toc503168455)

[表3-4 基础软件列表 3-2](#_Toc503168456)

[表3-5 外部服务列表 3-3](#_Toc503168457)

[表4-1 安装部署规划 4-1](#_Toc503168458)

[表4-2 二级域名规划 4-2](#_Toc503168459)

[表4-3 端口规划 4-2](#_Toc503168460)

[表4-4 数据库规划 4-3](#_Toc503168461)

[表4-5 数据库用户规划 4-3](#_Toc503168462)

# 产品概述

## 产品简介

## 系统架构

中企酷客系统整体架构如下图所示：

系统整体架构

## 逻辑架构

系统各中间件和应用系统间的逻辑架构如下图所示：

系统逻辑架构

# 安装概述

## 部署目标

## 部署结构

本次上线版本，系统部署结构如下图所示：

中企酷客系统部署架构

# 软硬件需求

## 服务器需求

系统只需安装于虚拟机环境中，具体的虚拟机资源需求如下：

虚拟机资源需求

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 运行环境 | 服务器名称 | 数量 | 配置要求 | 备注说明 |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
|  |  | 总计： |  |  |  |

## 存储需求

## 网络需求

### NAT网关需求

系统为互联网应用，对互联网的访问要求如下：

### 外网IP及带宽需求

系统生产及测试环境对外网IP和带宽的需求

带宽需求

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 带宽 | 服务器 | 域名绑定需求 | 绑定域名 |
| 1 |  |  |  |  |
|  |  |  |  |  |
| 2 |  |  |  |  |

### 网络端口需求

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 操作系统需求

64位Linux操作系统，如CentOS 7.3及以上版本。

## 基础软件需求

系统运行需要的基础软件和中间件如下：

基础软件列表

|  |  |  |  |
| --- | --- | --- | --- |
|  | 软件类型 | 软件名称 | 版本 |
| 1 | Java虚拟机 | JDK | 1.8 |
| 2 | 反向代理 | TEngine | 2.0.3 |
| 3 | Web服务器 | Nginx | 1.12.0 |
| 4 | 应用服务器 | Tomcat | 8.5.15 |
| 5 | 分布式协调器 | ZooKeeper | 3.4.11 |
| 6 | 分布式缓存 | Redis | 3.2.11 |
| 7 | 消息队列 | RocketMQ | 4.2.0 |
| 8 | 数据库中间件 | MyCat | 1.6.5 |
| 9 |  | Pgpool-II |  |
| 10 | 数据库系统 | MySQL | 5.6.35 |
| 11 |  | PostgreSQL | 9.6 |
| 12 | API网关 | Kong |  |
| 13 | 公共服务 | 对象存储服务 |  |
| 14 |  |  |  |

## 外部服务需求

系统运行依赖于以下外部系统或服务

外部服务列表

|  |  |  |  |
| --- | --- | --- | --- |
|  | 服务类型 | 服务名称 | 备注 |
| 1 | 短信发送服务 | 亿美 |  |
| 2 | 社交媒体 | 新浪微博 |  |
| 3 | CDN | 百度云加速 |  |

# 安装配置说明

## 系统安装

### 安装部署规划

## 系统配置

# 操作系统配置

## 组件升级

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 组件名称 |  |  |
| 1 | Ncat |  |  |
| 2 |  |  |  |
| 3 |  |  |  |

### Ncat安装

执行以下指令安装：

yum install nc -y

## 内核调整

## 网络设置

### DNS配置

将各服务器的DNS服务器地址配置为内部DNS服务器

### 防火墙配置

在各服务器上根据部署的服务和中间件，打开相应端口的网络访问权限，具体端口参考下表：

网络端口使用列表

## 时间设置

### 服务器时间同步

#### NTP客户端配置

## 用户设置

### 创建zceo用户

创建zceo用户及同名用户组；

执行命令：

[root@master1-40-12 /]# useradd -d /home/zceo -m -s /bin/bash -U zceo

设置zceo用户登录密码；

执行以下命令，按提示信息输入密码：

[root@master1-40-12 zceo]# passwd -f zceo

Changing password for user zceo.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

切换到zceo用户；

[root@master1-40-12 ~]# su – zceo

[zceo@master1-40-12 ~]$

### 创建应用安装目录

执行以下命令，创建“/apps”目录：

[root@master1-40-12 /]# mkdir /apps

[root@master1-40-12 /]# chown zceo:zceo /apps

### 设置本地时间

将用户的时区设置为东8区（北京时间）；

执行命令：

[zceo@master1-40-12 ~]$ echo "" >> /home/zceo/.bash\_profile

[zceo@master1-40-12 ~]$ echo "# Set Local Time Zone" >> /home/zceo/.bash\_profile

[zceo@master1-40-12 ~]$ echo "export TZ='Asia/Shanghai'" >> /home/zceo/.bash\_profile

[zceo@master1-40-12 ~]$ source ~/.bash\_profile

# 基础组件安装

## 安装JDK

下载安装包

通过Web网页，或执行以下命令下载JDK安装包到本地：

wget --no-cookies --no-check-certificate --header "Cookie: gpw\_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/8u151-b12/e758a0de34e24606bca991d704f6dcbf/jdk-8u151-linux-x64.tar.gz"

注意：

此处命令中为Java 8 Update 151版SDK，请根据具体下载版本对命令做相应修改。

上传安装包

执行以下命令，将JDK安装包上传到各服务器：

scp jdk-8u151-linux-x64.tar.gz zceo@node-40-57:~

注意：

此处命令中node-40-57为上传的目标服务器hostname，请根据具体情况修改为要上传的hostname或IP。

安装JDK

在各服务器安装包所在路径下执行以下命令解压安装包：

tar zxvf jdk-8u151-linux-x64.tar.gz -C /apps

设置环境变量

在各服务器上执行以下命令设置JDK环境变量：

echo ' ' >> /home/zceo/.bash\_profile

echo '# Set Java Environment' >> /home/zceo/.bash\_profile

echo 'JAVA\_HOME=/apps/jdk1.8.0\_151' >> /home/zceo/.bash\_profile

echo 'PATH=$JAVA\_HOME/bin:$PATH' >> /home/zceo/.bash\_profile

echo 'CLASSPATH=.:$JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar' >> /home/zceo/.bash\_profile

echo 'export JAVA\_HOME PATH CLASSPATH' >> /home/zceo/.bash\_profile

source /home/zceo/.bash\_profile

安装验证；

执行以下命令：

java -version

如果显示类似如下版本信息，则表示安装成功：

java version "1.8.0\_151"

Java(TM) SE Runtime Environment (build 1.8.0\_151-b12)

Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)

## 安装ZooKeeper集群

### 下载安装包

通过Web页面，或执行以下命令下载ZooKeeper安装包：

wget https://mirrors.tuna.tsinghua.edu.cn/apache/zookeeper/zookeeper-3.4.11/zookeeper-3.4.11.tar.gz

### 上传安装包

执行以下命令，将ZooKeeper安装包上传到各服务器：

scp zookeeper-3.4.11.tar.gz zceo@node-40-57:~

注意：

此处命令中node-40-57为上传的目标服务器hostname，请根据具体情况修改为要上传的hostname或IP。

### 安装ZooKeeper

在各服务器安装包所在路径下执行以下命令进行安装：

tar zxvf zookeeper-3.4.11.tar.gz -C /apps

### 参数配置

在ZooKeeper集群的各服务器上按以下步骤进行参数配置：

执行以下命令创建相关子目录

mkdir -p /apps/zookeeper-3.4.11/logs

mkdir -p /apps/zookeeper-3.4.11/snapshot

参考如下内容，编辑conf/zoo.cfg配置文件

tickTime=2000

initLimit=20

syncLimit=5

dataDir=/apps/zookeeper-3.4.11/snapshot

dataLogDir=/apps/zookeeper-3.4.11/data

clientPort=2181

server.1=master1-40-12:2888:3888

server.2=node-40-57:2888:3888

server.3=node-40-58:2888:3888

注意：

配置项server.1、server.2、server.3等中的数字即该服务器在ZooKeeper集群中的ID，后续配置的myid文件需与该值保持一致

设置myid

创建一个myid文件，设置内容为上一步骤zoo.cfg文件中配置的ID值。

参考如下命令：

echo ’1’ > /apps/zookeeper-3.4.11/snapshot/myid

配置JMX参数

修改“bin/zkServer.sh”脚本文件，找到以下几行代码：

# use POSTIX interface, symlink is followed automatically

ZOOBIN="${BASH\_SOURCE-$0}"

ZOOBIN="$(dirname "${ZOOBIN}")"

ZOOBINDIR="$(cd "${ZOOBIN}"; pwd)"

在这几行代码之前加入以下几行环境变量设置：

JMXPORT=21812

JMXAUTH=false

JMXSSL=false

JMXLOG4J=true

配置日志文件

编辑ZooKeeper目录下“conf/log4j.properties”文件，将以下几个配置项修改为本文档指定的配置：

zookeeper.root.logger=INFO, ROLLINGFILE

zookeeper.log.dir=/apps/zookeeper-3.4.11/logs/

zookeeper.tracelog.dir=/apps/zookeeper-3.4.11/logs/

log4j.appender.ROLLINGFILE=org.apache.log4j.DailyRollingFileAppender

注释掉以下内容：

#log4j.appender.ROLLINGFILE.MaxFileSize=10MB

编辑ZooKeeper目录下“bin/zkEnv.sh”文件，找到以下几行代码：

if [ "x${ZOO\_LOG\_DIR}" = "x" ]

then

ZOO\_LOG\_DIR="."

fi

if [ "x${ZOO\_LOG4J\_PROP}" = "x" ]

then

ZOO\_LOG4J\_PROP="INFO,CONSOLE"

fi

修改为：

if [ "x${ZOO\_LOG\_DIR}" = "x" ]

then

ZOO\_LOG\_DIR="$ZOOBINDIR/../logs/"

fi

if [ "x${ZOO\_LOG4J\_PROP}" = "x" ]

then

ZOO\_LOG4J\_PROP="INFO,ROLLINGFILE"

fi

启动ZooKeeper集群

在各节点执行“zkServer.sh”脚本，启动ZooKeeper：

[zceo@master1-40-12 zookeeper-3.4.11]$ /apps/zookeeper-3.4.11/bin/zkServer.sh start

ZooKeeper JMX enabled by default

ZooKeeper remote JMX Port set to 21812

ZooKeeper remote JMX authenticate set to false

ZooKeeper remote JMX ssl set to false

ZooKeeper remote JMX log4j set to true

Using config: /apps/zookeeper-3.4.11/bin/../conf/zoo.cfg

Starting zookeeper ... STARTED

### 安装验证

在各节点执行以下命令检查ZooKeeper状态

[zceo@master1-40 zookeeper-3.4.11]$ /apps/zookeeper-3.4.11/bin/zkServer.sh status

ZooKeeper JMX enabled by default

ZooKeeper remote JMX Port set to 21812

ZooKeeper remote JMX authenticate set to false

ZooKeeper remote JMX ssl set to false

ZooKeeper remote JMX log4j set to true

Using config: /apps/zookeeper-3.4.11/bin/../conf/zoo.cfg

Mode: follower

如无输出错误信息，并且最后一行显示为“Mode: follower”或“Mode: leader”则表示Zookeeper运行状态正常。

# 数据库服务安装

## 安装概述

### 简介

关于数据库服务的设计原则、整体架构、实现细节等，请参考《SCRM架构设计文档》。

### 部署架构

### 安装流程

## MySQL安装配置

MySQL的安装、配置及异步复制等的配置步骤，请根据实际软硬件环境，参考MySQL官网相关文档进行操作优化，此处不做详细描述。

## Mycat安装配置

#### 下载安装包

通过Web页面，或执行以下命令下载Mycat安装包：

wget http://dl.mycat.io/1.6.5/Mycat-server-1.6.5-release-20180122220033-linux.tar.gz

### 上传安装包

执行以下命令，将Mycat安装包上传到各服务器：

scp Mycat-server-1.6.5-release-20180122220033-linux.tar.gz zceo@node-40-57:~

注意：

此处命令中node-40-57为上传的目标服务器hostname，请根据具体情况修改为要上传的hostname或IP。

### 安装Mycat

在各服务器安装包所在路径下执行以下命令进行安装：

tar zxvf Mycat-server-1.6.5-release-20180122220033-linux.tar.gz -C /apps

Mycat将被安装到/apps/mycat目录下。

### 配置逻辑数据库

根据实际业务情况，编辑Mycat的配置文件server.xml、schema.xml、rule.xml、sequence\_time\_conf.properties、tenant\_db\_sharding\_mapper.txt）；

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 文件名称 | 说明 | 备注 |
| 1 | server.xml | Mycat全局配置 |  |
| 2 | schema.xml | 逻辑库配置 |  |
| 3 | rule.xml | 分片规则配置 |  |
| 4 | sequence\_time\_conf.properties | 全局ID生成算法配置 |  |
| 5 | tenant\_db\_sharding\_mapper.txt | 分片规则映射文件 |  |

注意：

各配置文件的配置方式，请参考架构设计文档，数据库服务架构部分，及相关技术文档。

上传配置文件

将上一步编辑好的配置文件上传到第一个Mycat服务器的/apps/mycat/conf/zkconf目录下；

scp server.xml zceo@node-40-57:/apps/mycat/conf/zkconf/

scp schema.xml zceo@node-40-57:/apps/mycat/conf/zkconf/

scp rule.xml zceo@node-40-57:/apps/mycat/conf/zkconf/

scp sequence\_time\_conf.properties zceo@node-40-57:/apps/mycat/conf/zkconf/

scp tenant\_db\_sharding\_mapper.txt zceo@node-40-57:/apps/mycat/conf/zkconf/

注意：

此处命令中node-40-57为上传的目标服务器hostname，请根据具体情况修改为要上传的hostname或IP。

### 配置ZooKeeper相关信息

初始化各Mycat节点的ZooKeeper配置：

参考以下内容，编辑各Mycat节点的“/apps/mycat/conf/myid.properties“文件：

loadZk=true

zkURL=10.12.40.12:2181,10.12.40.57:2181,10.12.40.58:2181

clusterId=SCRM\_Mycat-Cluster

myid=mycat\_node\_01

clusterSize=2

clusterNodes=mycat\_node\_01,mycat\_node\_02

#server booster ; booster install on db same server,will reset all minCon to 2

type=server

boosterDataHosts=dataHost1

注意：

* zkURL：修改为实际ZooKeeper服务器地址
* myid：该服务器节点的别名
* clusterSize：集群中Mycat服务器数量
* clusterNodes：各节点的myid

在Mycat配置文件所在节点，执行ZooKeeper数据初始化脚本。

[zceo@master1-40-12 ~]$ cd /apps/mycat/bin

[zceo@master1-40-12 bin]$ ./init\_zk\_data.sh

o2018-02-05 14:45:31 INFO JAVA\_CMD=java

o2018-02-05 14:45:31 INFO Start to initialize /mycat of ZooKeeper

o2018-02-05 14:45:32 INFO Done

### 启动Mycat集群

启动各Mycat节点：

[zceo@master1-40-12 bin]$ /apps/mycat/bin/mycat start

Starting Mycat-server...

### 验证Mycat集群部署状态

验证Mycat运行状态

执行以下命令，如果输出日志“Mycat-server is running”，则表明节点运行正常。

[zceo@master1-40-12 mycat]$ ./bin/mycat status

Mycat-server is running (121094).

验证ZooKeeper信息是否正常

zkCli.sh连接到ZooKeeper集群中任意一个节点：

[zceo@master1-40-12 zookeeper-3.4.11]$ ./bin/zkCli.sh -server 10.12.40.57:2181

/apps/zookeeper-3.4.11/bin

Connecting to 10.12.40.57:2181

Welcome to ZooKeeper!

JLine support is enabled

WATCHER::

WatchedEvent state:SyncConnected type:None path:null

查看mycat集群配置信息是否已同步到ZooKeeper中：

[zk: 10.12.40.57:2181(CONNECTED) 1] ls /mycat/SCRM\_Mycat\_Cluster

[schema, server, cache, line, lock, rules, sequences, bindata, ruledata, command, migrate]

如已同步，则表明ZooKeeper信息正常。

## PostgreSQL安装配置

PostgreSQL的安装、配置及异步复制的配置步骤，请DBA及运维人员根据实际软硬件环境和运维SOP，参考PostgreSQL官网相关文档进行操作优化，此处不做详细描述。

## Pgpool-II安装配置

Pgpool-II的安装、配置步骤，请DBA及运维人员根据实际软硬件环境和运维SOP，参考Pgpool-II官方相关文档进行操作优化，此处不做详细描述。

## 配置HAProxy集群

HAProxy的安装步骤，及Keepalived高可用配置，请运维人员根据实际软硬件环境和运维SOP，参考HAProxy、Keepalived的官方相关文档进行操作，此处不做详细描述。

# 分布式缓存服务安装

## 安装概述

### 

## 安装过程

### 下载源代码包

下载安装包

通过Web页面，或执行以下命令下载Redis安装包：

wget http://download.redis.io/releases/redis-3.2.11.tar.gz

上传安装包

执行以下命令，将Redis安装包上传到各服务器：

scp redis-3.2.11.tar.gz zceo@node-40-57:~

注意：

此处命令中node-40-57为上传的目标服务器hostname，请根据具体情况修改为要上传的hostname或IP。

### 编译源码

各服务器节点上执行以下编译操作

解压安装包

tar zxvf redis-3.2.11.tar.gz -C /apps/

编译Redis

首先确保已安装了GCC，然后执行以下命令，进行编译：

cd /apps/redis-3.2.11

make MALLOC=libc

### 配置Redis Cluster

在各Redis Cluster节点上配置集群Master节点配置文件

使用默认配置作为模板，创建Master节点配置文件

cd /apps/redis-3.2.11

cp redis.conf redis-master.conf

配置Master节点

编辑“redis-master.conf”文件，将以下配置项修改为对应值：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 配置项 | 设置值 | 备注 |
| 1 | bind | 0.0.0.0 |  |
| 2 | port | 6379 |  |
| 3 | daemonize | yes |  |
| 4 | pidfile | /apps/redis-3.2.11/pid/redis\_master.pid |  |
| 5 | requirepass | Cecook | Redis验证密码，根据自己需要进行设置 |
| 6 | cluster-enabled | yes |  |
| 7 | cluster-config-file | nodes-master.conf |  |
| 8 | cluster-node-timeout | 15000 |  |

配置Slave节点

编辑“redis-slave.conf”文件，将以下配置项修改为对应值：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 配置项 | 设置值 | 备注 |
| 1 | bind | 0.0.0.0 |  |
| 2 | port | 6380 | 服务端口 |
| 3 | daemonize | yes |  |
| 4 | pidfile | /apps/redis-3.2.11/pid/redis\_slave.pid |  |
| 5 | requirepass | Cecook | Redis验证密码，根据自己需要进行设置 |
| 6 | cluster-enabled | Yes |  |
| 7 | cluster-config-file | nodes-slave.conf |  |
| 8 | cluster-node-timeout | 15000 |  |

### 启动Redis Cluster

在各Redis Cluster节点上，分别启动Master和Slave节点。

启动Master节点

cd /apps/redis-3.2.11

./src/redis-server ./redis-master.conf

启动Slave节点

cd /apps/redis-3.2.11

./src/redis-server ./redis-slave.conf

### 节点握手配置

使用Redis自带客户端，登录到任意一个集群节点

cd /apps/redis-3.2.11

./src/redis-cli -h 10.12.40.57 -p 6379 -a cecook

参数说明

* -h：连接的服务节点IP
* -p：连接的服务节点端口
* -a：登录密码

节点间握手设置

执行“cluster meet”命令将各节点加入集群：

10.12.40.57:6379> cluster meet 10.12.40.57 6379

OK

10.12.40.57:6379> cluster meet 10.12.40.57 6380

OK

10.12.40.57:6379> cluster meet 10.12.40.58 6379

OK

10.12.40.57:6379> cluster meet 10.12.40.58 6380

OK

10.12.40.57:6379> cluster meet 10.12.40.69 6379

OK

10.12.40.57:6379> cluster meet 10.12.40.69 6380

OK

状态验证

通过“cluster nodes”命令查看集群各节点情况：

10.12.40.69:6379> cluster nodes

de37f68f12160c90959d635d5c8b1cbb0bfd9795 10.12.40.57:6380 master - 0 1517825586413 2 connected

dc12f7e2e7a851ffd96ee0ce46d3ef548205c945 10.12.40.57:6379 master - 0 1517825589418 1 connected

02cd9e85215ea6474666d9b63f41ceb26fb9d31d 10.12.40.69:6380 master - 0 1517825587415 5 connected

df3e7c5dca66ed7dae2ba15c79e1f4628f0ac6f3 10.12.40.58:6379 master - 0 1517825588417 0 connected

0acc07d5217b943203ae3a1357c7a24daf945a16 10.12.40.58:6380 master - 0 1517825585912 3 connected

8fb566dce5dfe0823f231c7e706d26ca94b7edb9 10.12.40.69:6379 myself,master - 0 0 4 connected

如以上信息所示，如果所有节点都加入集群，并为connected状态，则表示配置成功。

### 分配哈希槽

将集群哈希槽，平均分配到3个Master节点，第一个节点分配0~5461，第二个节点分配5462~10922，第三个节点分配10923~16383。

分配哈希槽

在命令行下执行以下命令：

cd /apps/redis-3.2.11

./src/redis-cli -h 10.12.40.57 -p 6379 -a cecook2017 cluster addslots {0..5461}

./src/redis-cli -h 10.12.40.58 -p 6379 -a cecook2017 cluster addslots {5462..10922}

./src/redis-cli -h 10.12.40.69 -p 6379 -a cecook2017 cluster addslots {10923..16383}

操作验证

登录到任意一个集群节点，执行“cluster nodes”命令，查看结群节点状态。如操作成功，各节点分配的哈希槽，将显示于节点信息末尾。

10.12.40.69:6379> cluster nodes

de37f68f12160c90959d635d5c8b1cbb0bfd9795 10.12.40.57:6380 master - 0 1517825586413 2 connected

dc12f7e2e7a851ffd96ee0ce46d3ef548205c945 10.12.40.57:6379 master - 0 1517825589418 1 connected 0-5461

02cd9e85215ea6474666d9b63f41ceb26fb9d31d 10.12.40.69:6380 master - 0 1517825587415 5 connected

df3e7c5dca66ed7dae2ba15c79e1f4628f0ac6f3 10.12.40.58:6379 master - 0 1517825588417 0 connected 5462-10922

0acc07d5217b943203ae3a1357c7a24daf945a16 10.12.40.58:6380 master - 0 1517825585912 3 connected

8fb566dce5dfe0823f231c7e706d26ca94b7edb9 10.12.40.69:6379 myself,master - 0 0 4 connected 10923-16383

### 配置Slave节点

将剩余的3个节点，分别配置为各Master节点的从节点

在命令行通过redis-cli执行“cluster replicate”命令：

cd /apps/redis-3.2.11

./src/redis-cli -h 10.12.40.58 -p 6380 -a cecook2017 cluster replicate dc12f7e2e7a851ffd96ee0ce46d3ef548205c945

./src/redis-cli -h 10.12.40.69 -p 6380 -a cecook2017 cluster replicate df3e7c5dca66ed7dae2ba15c79e1f4628f0ac6f3

./src/redis-cli -h 10.12.40.57 -p 6380 -a cecook2017 cluster replicate 8fb566dce5dfe0823f231c7e706d26ca94b7edb9

说明：

类似8fb566dce5dfe0823f231c7e706d26ca94b7edb9这样的字符串，为对应Master的标识，可通过“cluster nodes”命令获得。

配置验证

执行“cluster nodes”命令，查看各节点状态，如上一步配置的各节点，已变为“slave”状态，则表明配置成功。

10.12.40.57:6379> cluster nodes

8fb566dce5dfe0823f231c7e706d26ca94b7edb9 10.12.40.69:6379 master - 0 1517826255709 4 connected 10923-16383

de37f68f12160c90959d635d5c8b1cbb0bfd9795 10.12.40.57:6380 slave 8fb566dce5dfe0823f231c7e706d26ca94b7edb9 0 1517826254707 4 connected

df3e7c5dca66ed7dae2ba15c79e1f4628f0ac6f3 10.12.40.58:6379 master - 0 1517826252705 0 connected 5462-10922

0acc07d5217b943203ae3a1357c7a24daf945a16 10.12.40.58:6380 slave dc12f7e2e7a851ffd96ee0ce46d3ef548205c945 0 1517826251703 3 connected

dc12f7e2e7a851ffd96ee0ce46d3ef548205c945 10.12.40.57:6379 myself,master - 0 0 1 connected 0-5461

02cd9e85215ea6474666d9b63f41ceb26fb9d31d 10.12.40.69:6380 slave df3e7c5dca66ed7dae2ba15c79e1f4628f0ac6f3 0 1517826256709 5 connected

# 消息队列服务安装

## 安装说明

### 安装概述

### 部署架构

### 安装流程

消息队列服务的安装流程，分为以下几个步骤：

上传安装包

解压安装包

配置并启动NameServer集群

配置并启动Broker集群

初始化Topic

## 安装过程

### 下载安装包

通过Web页面，或执行以下命令下载RocketMQ安装包：

wget http://mirrors.hust.edu.cn/apache/rocketmq/4.2.0/rocketmq-all-4.2.0-bin-release.zip

### 上传安装包

执行以下命令，将RocketMQ安装包上传到各服务器：

scp rocketmq-all-4.2.0-bin-release.zip zceo@node-40-57:~

注意：

此处命令中node-40-57为上传的目标服务器hostname，请根据具体情况修改为要上传的hostname或IP。

### 安装RocketMQ

在各服务器安装包所在路径下执行以下命令进行安装：

unzip rocketmq-all-4.2.0-bin-release.zip -d /apps/rocketmq-4.2.0

RocketMQ将被安装到/apps/rocketmq-4.2.0目录下。

### 配置NameServer

配置日志输出

编辑RocketMQ安装目录下“conf/logback\_namesrv.xml”文件。

配置内存等JVM参数

编辑RocketMQ安装目录下“bin/runserver.sh”文件，根据实际业务及软硬件环境修改调整相应配置

启动NameServer

nohup /apps/rocketmq-4.2.0/bin/mqnamesrv &

检查NameServer工作状态

任意RocketMQ节点上执行以下命令，检查各NameServer状态：

/apps/rocketmq-4.2.0/bin/mqadmin getNamesrvConfig -n 10.12.40.12:9876

说明

-n 参数后指定的IP及端口为目标NameServer服务器的IP和端口号。

返回类似以下配置信息则表明该NameServer节点工作正常：

============10.12.40.12:9876============

serverChannelMaxIdleTimeSeconds = 120

listenPort = 9876

serverCallbackExecutorThreads = 0

serverAsyncSemaphoreValue = 64

serverSocketSndBufSize = 65535

rocketmqHome = /apps/rocketmq-4.2.0

clusterTest = false

serverSelectorThreads = 3

useEpollNativeSelector = false

orderMessageEnable = false

serverPooledByteBufAllocatorEnable = true

kvConfigPath = /home/zceo/namesrv/kvConfig.json

serverWorkerThreads = 8

serverSocketRcvBufSize = 65535

productEnvName = center

serverOnewaySemaphoreValue = 256

configStorePath = /home/zceo/namesrv/namesrv.properties

### 配置Broker

配置日志输出

编辑RocketMQ安装目录下“conf/logback\_broker.xml”文件。

配置Broker基本参数

编辑RocketMQ安装目录下“conf/2m-2s-async”目录下各个配置文件，对以下配置项进行相应修改：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 配置项 | 参数值 | 备注 |
| 1 | brokerClusterName | SCRM\_MQ\_Cluster | Broker集群名称 |
| 2 | fileReserveTime | 120 | 文件保留时长（单位：小时） |
| 3 | listenPort | 10911或20911 | Master节点配置为10911；  Slave节点配置为20911 |
| 4 | namesrvAddr | NameServer地址列表 | 采用“IP1:Port1;IP2:Port2的形式 |
| 5 | brokerIP1 | 客户端IP | 指定连接到NameServer使用的IP |
| 6 | storePathRootDir | /apps/rocketmq-4.2.0/store/Broker名称 | 同一服务器上的Broker存储目录不要冲突 |
| 7 | storePathCommitLog | apps/rocketmq-4.2.0/store/Broker名称/commitLog | 同一服务器上的Broker存储目录不要冲突 |

配置内存等JVM参数

编辑RocketMQ安装目录下“bin/runbroker.sh”文件，根据实际业务及软硬件环境修改调整相应配置

启动Broker的Master节点

Broker-a节点上执行以下命令，启动Broker-a

cd /apps/rocketmq-4.2.0

nohup ./bin/mqbroker -c ./conf/2m-2s-async/broker-a.properties &

Broker-b节点上执行以下命令，启动Broker-b

cd /apps/rocketmq-4.2.0

nohup ./bin/mqbroker -c ./conf/2m-2s-async/broker-b.properties &

启动Broker的Slave节点

Broker-a节点上执行以下命令，启动Broker-b的Slave

cd /apps/rocketmq-4.2.0

nohup ./bin/mqbroker -c ./conf/2m-2s-async/broker-b-s.properties &

Broker-b节点上执行以下命令，启动Broker-a的Slave

cd /apps/rocketmq-4.2.0

nohup ./bin/mqbroker -c ./conf/2m-2s-async/broker-a-s.properties &

检查Broker工作状态

任意RocketMQ节点上执行以下命令，查看各Broker状态：

[zceo@node-40-58 rocketmq-4.2.0]$ ./bin/mqadmin clusterList -n "10.12.40.12:9876;10.12.40.57:9876"

返回类似以下配置信息则表明各Broker节点工作正常：

#Cluster Name #Broker Name #BID #Addr #Version #InTPS(LOAD) #OutTPS(LOAD) #PCWait(ms) #Hour #SPACE

SCRM\_MQ\_Cluster broker-a 0 10.12.40.58:10911 V4\_2\_0\_SNAPSHOT 0.00(0,0ms) 0.00(0,0ms) 0 421641.10 -1.0000

SCRM\_MQ\_Cluster broker-a 1 10.12.40.69:20911 V4\_2\_0\_SNAPSHOT 0.00(0,0ms) 0.00(0,0ms) 0 421641.10 0.1229

SCRM\_MQ\_Cluster broker-b 0 10.12.40.69:10911 V4\_2\_0\_SNAPSHOT 0.00(0,0ms) 0.00(0,0ms) 0 421641.10 -1.0000

SCRM\_MQ\_Cluster broker-b 1 10.12.40.58:20911 V4\_2\_0\_SNAPSHOT 0.00(0,0ms) 0.00(0,0ms) 0 421641.10 -1.0000

## 初始化Topic

根据业务需要，和实际软硬件环境，创建Topic。

# 服务治理框架安装

## 安装概述

### 

## 安装准备

## 安装过程

### JVM设置

执行以下命令，进行JVM相关参数设置：

echo ' ' >> /home/cecook/.bashrc

echo '# Set Dubbo Memory OPTS' >> /home/cecook/.bashrc

echo ‘export DUBBO\_MEM\_OPTS=“ -server -Xmx5g -Xms5g -Xmn512m -XX:PermSize=128m -Xss256k -XX:+DisableExplicitGC -XX:+UseConcMarkSweepGC -XX:+CMSParallelRemarkEnabled -XX:+UseCMSCompactAtFullCollection -XX:LargePageSizeInBytes=128m -XX:+UseFastAccessorMethods -XX:+UseCMSInitiatingOccupancyOnly -XX:CMSInitiatingOccupancyFraction=70”’ >> /home/cecook/.bashrc

source /home/cecook/.bashrc

### Dubbo-Admin安装

Dubbo-

# 服务网关安装

## 安装概述

### 部署架构

### 安装流程

服务网关Kong的安装流程，分为以下几个步骤：

上传Kong安装包

解压Kong安装包

配置Kong配置文件

配置环境变量

启动Kong服务

## 安装过程

### 上传安装包

执行以下命令，将Kong安装包上传到各服务器：

scp kong.tar.gz zceo@node-40-57:~

注意：

此处命令中node-40-57为上传的目标服务器hostname，请根据具体情况修改为要上传的hostname或IP。

### 安装Kong

在各服务器安装包所在路径下执行以下命令进行安装包解压：

tar zxvf kong.tar.gz -C /apps

Kong将被安装到/apps/kong和/apps/openresty这两个目录下。

### 配置Kong

编辑Kong安装目录“/apps/kong/kong”下的配置文件“conf/kong.conf”，根据实际软硬件环境修改以下几项配置：

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 配置项 | 值 | 备注 |
| 1 | proxy\_listen | 服务绑定IP：端口 | 服务端口，默认18000 |
| 2 | proxy\_listen\_ssl | 服务绑定IP：端口 | 服务端口（SSL），默认18443 |
| 3 | admin\_listen | 服务绑定IP：端口 | 管理端口，默认18001 |
| 4 | admin\_listen\_ssl | 服务绑定IP：端口 | 管理端口（SSL），默认18444 |
| 5 | nginx\_user | zceo zceo | 运行Kong服务的用户组和用户名 |
| 6 | pg\_host |  | PostgreSQL服务器IP |
| 7 | pg\_port | 5432 | PostgreSQL服务器端口 |
| 8 | pg\_user | scrmuser | PostgreSQL服务器用户名 |
| 9 | pg\_password | 密码 | PostgreSQL服务器密码 |
| 10 | pg\_database | api\_gateway\_kong | PostgreSQL数据库名 |

### 配置环境变量

各Kong节点上执行以下命令，配置环境变量：

echo ‘export OPENRESTY\_HOME=/apps/openresty’ >> ~/.bashrc

echo ‘export KONG\_HOME=/apps/kong’ >> ~/.bashrc

echo ‘export LUA\_PATH="/apps/kong/lua/5.1/?.lua;/apps/kong/lua/5.1/?/init.lua"’ >> ~/.bashrc

echo ‘export LUA\_CPATH="/apps/kong/lib/lua/5.1/?.so"’ >> ~/.bashrc

echo ‘export PATH=$PATH:$KONG\_HOME/bin:$OPENRESTY\_HOME/bin’ >> ~/.bashrc

source ~/.bashrc

### 启动服务

各Kong节点上，在命令行执行以下命令，启动服务：

kong start

如服务启动成功，将返回提示：

Kong Started.

# 反向代理安装

## 

### 