# **Serving部署指南**

# **1.服务器配置**

| **服务器** |  |
| --- | --- |
| 数量 | 1 or 2 |
| 配置 | 8 core /16GB memory / 500GB硬盘/10M带宽 |
| 操作系统 | CentOS linux 7.2及以上 |
| 依赖包 | yum源： gcc gcc-c++ make openssl-devel supervisor gmp-devel mpfr-devel libmpc-devel libaio numactl autoconf automake libtool libffi-devel snappy snappy-devel zlib zlib-devel bzip2 bzip2-devel lz4-devel libasan （可以使用初始化脚本env.sh安装） |
| 用户 | 用户：app，属主：apps（app用户需可以sudo su root而无需密码） |
| 文件系统 | 1. 500G硬盘挂载在/ data目录下； 2.创建/ data / projects目录，目录属主为：app:apps |

# **2.集群规划**

| **party** | **主机名** | **IP地址** | **操作系统** |
| --- | --- | --- | --- |
| PartyA | VM\_0\_1\_centos | 192.168.0.1 | CentOS 7.2 |
| PartyB | VM\_0\_2\_centos | 192.168.0.2 | CentOS 7.2 |

# **3.基础环境配置**

## **3.1 hostname配置(可选)**

****1）修改主机名****

****在192.168.0.1 root用户下执行：****

hostnamectl set-hostname VM\_0\_1\_centos

****在192.168.0.2 root用户下执行：****

hostnamectl set-hostname VM\_0\_2\_centos

****2）加入主机映射****

****在目标服务器（192.168.0.1 192.168.0.2）root用户下执行：****

vim /etc/hosts

192.168.0.1 VM\_0\_1\_centos

192.168.0.2 VM\_0\_2\_centos

## **3.2 关闭selinux(可选)**

****在目标服务器（192.168.0.1 192.168.0.2）root用户下执行：****

sed -i '/^SELINUX/s/=.\*/=disabled/' /etc/selinux/config

setenforce 0

## **3.3 修改Linux最大打开文件数**

****在目标服务器（192.168.0.1 192.168.0.2）root用户下执行：****

vim /etc/security/limits.conf

\* soft nofile 65536

\* hard nofile 65536

## **3.4 关闭防火墙(可选)**

****在目标服务器（192.168.0.1 192.168.0.2）root用户下执行****

systemctl disable firewalld.service

systemctl stop firewalld.service

systemctl status firewalld.service

## **3.5 软件环境初始化**

****1）创建用户****

****在目标服务器（192.168.0.1 192.168.0.2）root用户下执行****

groupadd -g 6000 apps

useradd -s /bin/bash -g apps -d /home/app app

passwd app

****2）配置sudo****

****在目标服务器（192.168.0.1 192.168.0.2）root用户下执行****

vim /etc/sudoers.d/app

app ALL=(ALL) ALL

app ALL=(ALL) NOPASSWD: ALL

Defaults !env\_reset

****3）配置ssh无密登录****

****a. 在目标服务器（192.168.0.1 192.168.0.2）app用户下执行****

su app

ssh-keygen -t rsa

cat ~/.ssh/id\_rsa.pub >> /home/app/.ssh/authorized\_keys

chmod 600 ~/.ssh/authorized\_keys

****b.合并id\_rsa\_pub文件****

拷贝192.168.0.1的authorized\_keys 到192.168.0.2 ~/.ssh目录下,追加到192.168.0.2的id\_rsa.pub到authorized\_keys，然后再拷贝到192.168.0.1

****在192.168.0.1 app用户下执行****

scp ~/.ssh/authorized\_keys app@192.168.0.2:/home/app/.ssh

输入密码

****在192.168.0.2 app用户下执行****

cat ~/.ssh/id\_rsa.pub >> /home/app/.ssh/authorized\_keys

scp ~/.ssh/authorized\_keys app@192.168.0.1:/home/app/.ssh

覆盖之前的文件

****c. 在目标服务器（192.168.0.1 192.168.0.2）app用户下执行ssh 测试****

ssh app@192.168.0.1

ssh [app@192.168.0.2](mailto:app@192.168.0.2)

1. ****需要的软件版本****

Git 1.8+

Maven 3.5+

Redis 4.0+

Jdk 1.8+

Zookeeper 3.5.5+

# **4.项目部署**

注：此指导安装目录默认为/data/projects/，执行用户为app，安装时根据具体实际情况修改。

## **4.1 代码获取和打包**

****在目标服务器（192.168.0.1 具备外网环境）app用户下执行****:

****注意：服务器需已安装好git和maven 3.5+****

进入执行节点的/data/projects/目录，执行：

cd /data/projects/

git clone https://github.com/FederatedAI/FATE-Serving.git

## **4.2 配置文件修改和示例**

****在目标服务器（192.168.0.1）app用户下执行****

进入到FATE-Serving目录下的FATE-Serving/cluster-deploy/scripts目录下，修改配置文件allinone\_cluster\_configurations.sh.

配置文件allinone\_cluster\_configurations.sh说明：

| **配置项** | **配置项意义** | **配置项值** | **说明** |
| --- | --- | --- | --- |
| user | 操作用户 | 默认为app | 使用默认值 |
| host\_guest | Host和guest的服务器ip | (192.168.0.1 192.168.0.2) | 部署host，只填写一个ip，  部署host和guest，填写两个ip。 |
| roll\_hostAndguest | host和guest调用roll的ip | (192.168.0.1 192.168.0.2) | 部署host，只填写一个ip，  部署host和guest对应roll的ip |
| deploy\_dir | Serving安装路径 | 默认为 /data/projects | 使用默认值 |
| apply\_zk | 是否使用zk | 默认true | 使用默认值 |
| host\_redis\_ip | Host连接的redis的ip | 127.0.0.1 |  |
| host\_redis\_port | Host连接的redis的端口号 | 6379 |  |
| host\_redis\_password | Host连接的redis的密码 | fate\_dev |  |
| guest\_redis\_ip | Guest连接的redis的ip | 127.0.0.1 |  |
| guest\_redis\_port | Guest连接的redis的端口号 | 6379 |  |
| guest\_redis\_password | Guest连接的redis的密码 | fate\_dev |  |
| host\_zk\_url | Host连接的zk地址 | zookeeper://localhost:2181 |  |
| guest\_zk\_url | Guest连接的zk地址 | zookeeper://localhost:2181 |  |
| workMode | Fate\_flow工作模式 1:集群 0:单机 | 1 |  |

## **4.3 部署**

****1）打包****

按照上述配置含义修改allinone\_cluster\_configurations.sh文件对应的配置项后，然后在

FATE-Serving/serving-server/cluster-deploy/scripts目录下执行部署脚本 : sh package.sh

1. ****启动项目****

cd /data/projects/fate-serving

sh services.sh all start

****如果是两台进各自的分别执行一个****

****4)查看所有状态****

sh services.sh all status

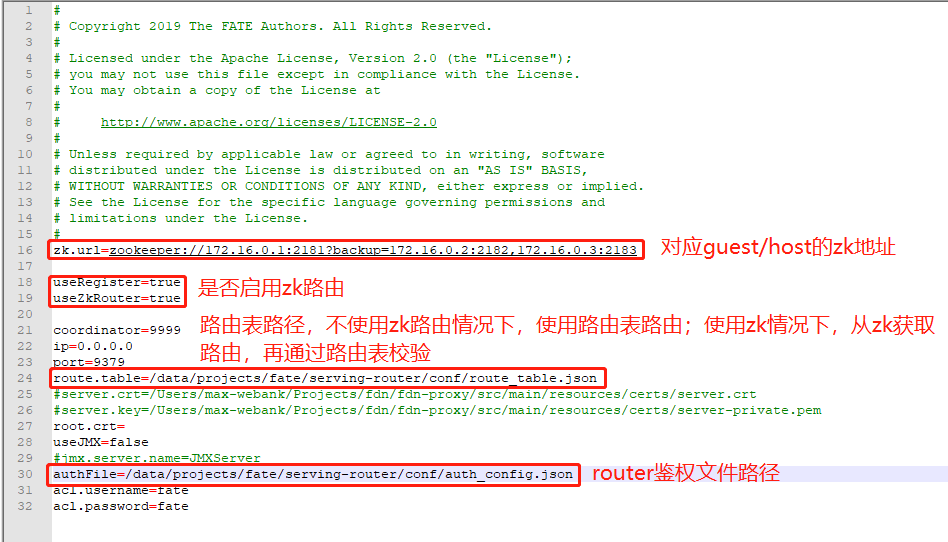
****5)关闭所有****

sh services.sh all stop

# 配置文件详解

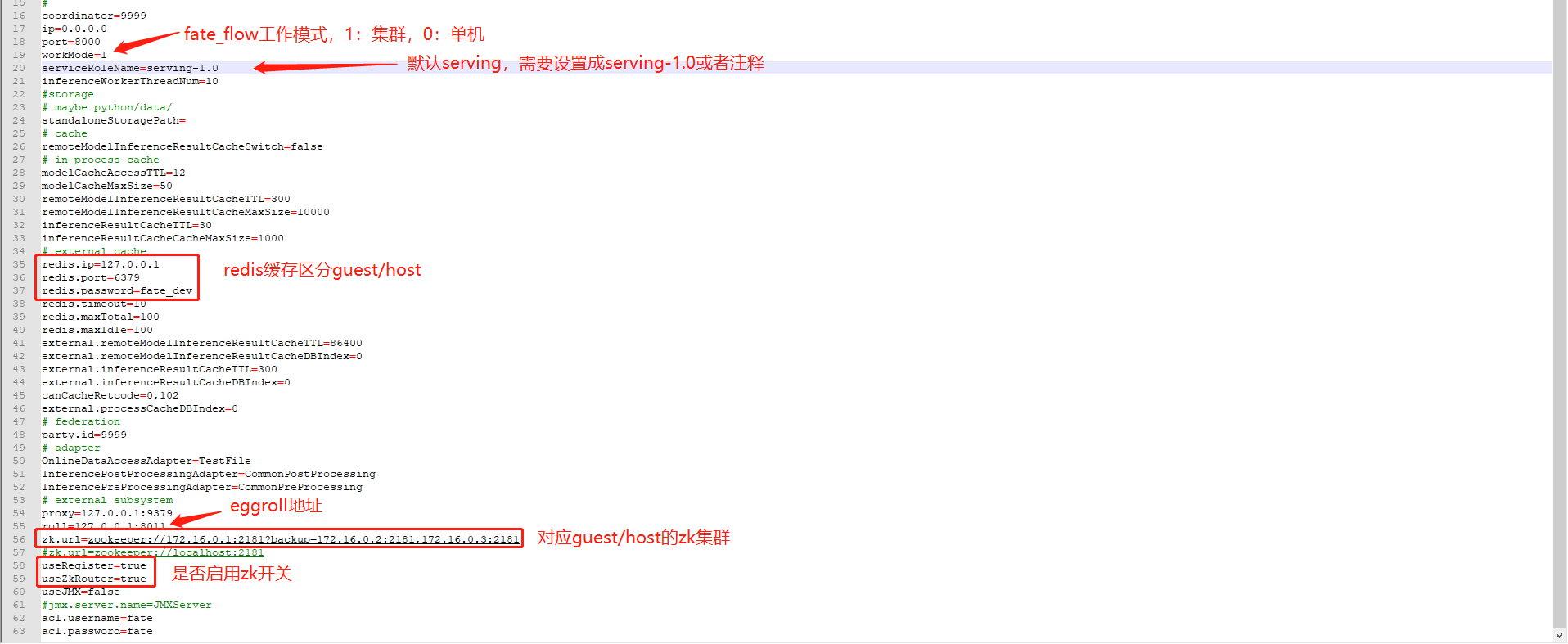
router配置文件详解

## router/conf/proxy.properties



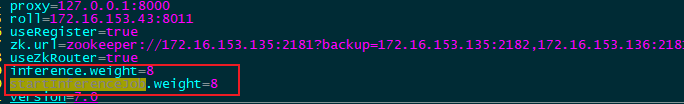
Serving配置文件详解

/serving-server/conf/serving-server.properties



其中：  
  
userRegister：服务注册的治理开关  
  
useZkRouter：服务订阅的治理开关

如果我们有做serving集群，那么我们可以配置该serving服务内提供的接口的被访问权重



命名规则：  
“.weight”前面是需要配置的接口名，例如上图配置的inference接口和startInferenceJob接口  
权重计算方式：  
权重的计算方式为，集群内所有该接口的权重值相加等于分母，每个服务该接口的权重值等于分子

详情查看:

<https://github.com/FederatedAI/FATE-Serving/blob/master/README.md>