

FaceService 部署文档

架构拓扑图:

FS 集群中的 DataCenter 可以动态扩展在任何机器或者硬盘上面
DB 集群也可以分布式部署在多台机器

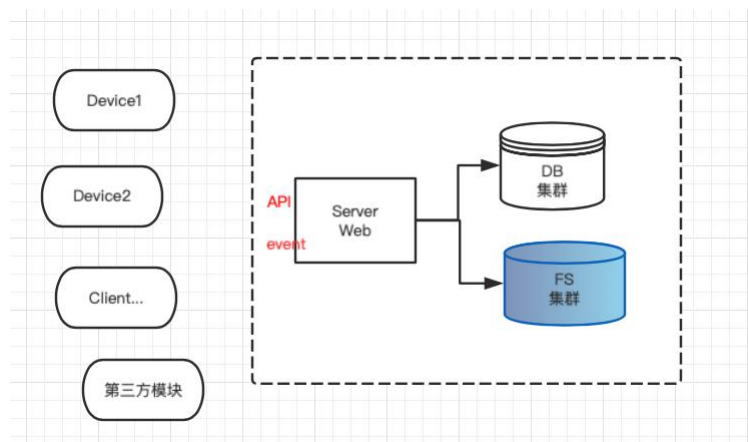


图 2 内网部署

通用部分

部署 SeaweeFS 文件服务器

安装配置 golang

下载:

wget <https://studygolang.com/dl/golang/go1.14.linux-amd64.tar.gz>

解压到 /usr/local

tar -C /usr/local -xzf go1.14.linux-amd64.tar.gz

配置环境变量:

vi /etc/profile

文件最后面加上:

```
export GOROOT=/usr/local/go
```

```
export PATH=$PATH:$GOROOT/bin
```

```
export GOPATH=/root/go
```

```
export PATH=$PATH:$GOPATH/BIN
```

配置文件生效

```
source /etc/profile
```

测试

```
go --version
```

安装 seaweedFS

下载

```
wget
```

```
https://github.com/chrislusf/seaweedfs/releases/download/1.74/linux\_amd64.tar.gz
```

```
tar -xzf linux_amd64.tar.gz
```

注：以下路径根据机器磁盘的具体情况自己定义

```
mkdir -p /home/data/fileData
```

```
# 创建 volume 目录
```

```
mkdir -p /home/data/t_v1
```

```
# 创建 Volume 及 Master 日志文件
```

```
touch /home/data/t_v1_sfs.log
```

```
touch server_sfs.log
```

```
mkdir /home/mount
```

启动 master 节点

```
nohup ./weed master -mdir=/home/data/fileData &>> server_sfs.log &
```

```
mkdir -p /home/data/fileData/t_v1
```

```
mkdir -p /home/data/fileData/logs
```

```
touch /home/data/fileData/logs/t_v1_sfs.log
```

```
nohup ./weed volume -dir=/home/data/fileData/t_v1 -mserver="localhost:9333"
```

```
-port=8081 &>>/home/data/fileData/logs/t_v1_sfs.log &
```

安装 Redis

```
wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

```
sudo yum install epel-release-latest-7.noarch.rpm
```

```
sudo yum install -y redis
```

安装 MongoDB

配置 yum 源

<http://mirrors.aliyun.com/mongodb/yum/redhat/7Server/mongodb-org> 查看版本

```
cd /etc/yum.repos.d
```

```
vim mongodb-org-4.2.repo
```

增加如下内容

```
[mongodb-org]
```

```
name=MongoDB Repository
```

```
baseurl=http://mirrors.aliyun.com/mongodb/yum/redhat/7Server/mongodb-org/4.2/x86\_64/
```

```
gpgcheck=0
```

```
enabled=1
```

运行安装命令

```
yum -y install mongodb-org
```

查看 mongo 安装位置 `whereis mongod`

查看修改配置文件：`vim /etc/mongod.conf`

启动服务

```
mongod --config /etc/mongod.conf
```

新建数据库：

```
mongo
```

```
use guarder
```

安装 Nodejs

```
curl -sL https://rpm.nodesource.com/setup_12.x | sudo bash -
```

```
sudo yum install -y nodejs
```

```
node -version
```

单点部署模式（不用 Nginx 部署前端）建议用 Jenkins

```
cd guarderServer
```

安装依赖包

```
npm -g install yarn
```

```
yarn install
```

如果前端非 Nginx 方式部署，可以用以下方式，Nginx 方式请略过

拷贝前端的编译后文件到服务端的发布目录(下面路径自己补充)

```
cp -f -r guarder_client/dist/* guarder_server/app/public
```

替换一些加载路径

```
sed -i 's/"\Vumi/"\public\Vumi/g' /guarderServer/app/public/index.html
```

修改认证密钥

```
/guarderServer/app/config/config.default.js
```

Config.jwt.sectet 修改成你要的密钥

启动服务

```
npm start
```

服务会有守护进程自动维护

停止服务

```
npm stop
```

修改服务端口

如果你不想让应用占用 8999 端口，可以修改如下文件

```
package.json > scripts.start --port 8999 改成你想要的端口
```

测试服务是否运行

```
http://ip:port/api/release
```