# linux下AWS集群部署

## 一、环境准备

假设某用户生产环境全部为Linux，部署要求如下：

| **IP** | **说明** |
| --- | --- |
| 192.168.100.30 | 数据库服务 |
| 192.168.100.10 | AWS实例1，提供完整的Web和App服务 |
| 192.168.100.20 | AWS实例2，提供完整的Web和App服务 |

注：

1、在本地宿主服务器创建3个干净的Linux虚拟机

2、将标准AWS PaaS平台解压到服务器的/data/awsinst/下

3、完成192.168.100.30的MYSQL数据库安装和AWS PaaS数据库的创建及初始化

4、修改/data/awsinst/bin/conf/server.xml的jdbc连接配置

5、测试执行aws\_startup.sh和httpd\_startup.sh能够正常启动

6、完成以上任务，可验证基本环境已就绪

## **二、参数配置**

### 1、Web层参数设置，aws-server.xml

/data/awsinst/webserver/webapps/portal/WEB-INF/classes/aws-server.xml

修改cluster配置项，参考如下：

<cluster service="true" group="aws cluster group - dev" />

### 2、App层参数设置，server.xml

/data/awsinst/bin/conf/server.xml

修改cluster配置项，参考如下：

<cluster service="true" group="aws cluster group - dev" />

注：

1、生产环境可把- dev修改为- prd

2、以上操作在两台AWS服务器都要进行相同的配置

## **编写启停脚本**

### 在data目录下创建cluster

cd /data

mkdir cluster

### 创建web服务启动脚本

#### 创建脚本

cd /data/cluster/

vi httpd\_startup.sh

#### 编写脚本

#!/bin/sh

echo "HTTP Cluster Server - Startup"

cd /data/awsinst/bin

./httpd-startup.sh -Dserver.ip=192.168.100.10 -Dserver.id=web1

#非图形界面nohup启动脚本：

#nohup ./httpd-startup.sh -Dserver.ip=192.168.100.10 -Dserver.id=web1 >/dev/null 2>&1 &

#### 设置脚本可执行

chmod +x httpd\_startup.sh

### 创建web服务关闭脚本

#### 创建脚本

cd /data/cluster/

vi httpd\_shutdown.sh

#### 编写脚本

#!/bin/sh

echo "HTTP Cluster Server - Shutdown"

cd /data/awsinst/bin

./httpd-shutdown.sh

#### 设置脚本可执行

chmod +x httpd\_shutdown.sh

### 创建app服务启动脚本

#### 创建脚本

cd /data/cluster/

vi aws\_startup.sh

#### 编写脚本

#!/bin/sh

echo "AWS Cluster Server - Startup"

cd /data/awsinst/bin

./aws\_startup.sh -Daws.ip=192.168.100.10 -Daws.inst=app1

#非图形界面nohup启动脚本

#nohup ./aws\_startup.sh -Daws.ip=192.168.100.10 -Daws.inst=app1 >/dev/null 2>&1 &

#### 设置脚本可执行

chmod +x aws\_startup.sh

### 创建APP服务关闭脚本

#### 创建脚本

cd /data/cluster/

vi aws\_shutdown.sh

#### 编写脚本

#!/bin/sh

echo "AWS Cluster Server - Shutdown"

cd /data/awsinst/bin

./aws\_shutdown.sh

#### 设置脚本可执行

chmod +x aws\_shutdown.sh

注：

1. 以上操作两台AWS服务器都要进行相同的配置
2. web服务启动脚本中-Dserver.ip需要修改为对应机器的IP，-Dserver.id两台机器不能重复
3. app服务启动脚本中-Daws.ip需要修改为对应机器的IP，-Daws.inst两台机器不能重复

## 安装nginx

### 安装 gcc

先输入命令 gcc -v 如果打印出版本号就不需要安装，可跳过这一步

yum install -y gcc automake autoconf libtool make

### 安装gcc-c++

yum install -y gcc-c++

### 安装pcre

yum install -y pcre pcre-devel

### 安装zlib

yum install -y zlib zlib-devel

### 安装openssl

yum install -y openssl openssl-devel

### 安装nginx

#### 添加nginx的源

#### Centos6配置nginx源

rpm -ivh http://nginx.org/packages/centos/6/noarch/RPMS/nginx-release-centos-6-0.el6.ngx.noarch.rpm

#### Centos7配置nginx源

rpm -ivh http://nginx.org/packages/centos/7/noarch/RPMS/nginx-release-centos-7-0.el7.ngx.noarch.rpm

#### 安装nginx

yum install -y nginx

### 安装之后，可以查看nginx的默认安装目录

whereis nginx

**以下是Nginx的默认路径：**

(1) Nginx全局配置：/etc/nginx/nginx.conf

(2) PID目录：/var/run/nginx.pid

(3) 错误日志：/var/log/nginx/error.log

(4) 访问日志：/var/log/nginx/access.log

(5) 默认站点目录：/usr/share/nginx/html

(6) 默认站点配置：/etc/nginx/conf.d/default.conf

### 启动nginx

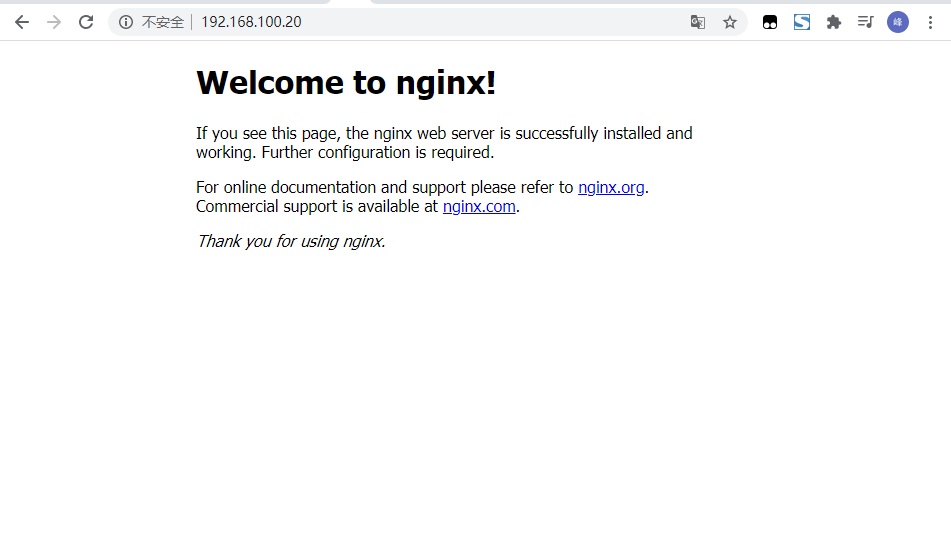
**Nginx启停命令：**

systemctl start nginx.service

systemctl stop nginx.service

systemctl restart nginx.service

systemctl status nginx.service



当你看到这个页面即表示nginx安装成功。

**其他常用的nginx命令**

nginx -s reload ：修改配置后重新加载生效

nginx -s reopen ：重新打开日志文件

nginx -t -c /path/to/nginx.conf 测试nginx配置文件是否正确

nginx -s stop ：快速停止nginx

quit ：完整有序的停止nginx

其他的停止nginx 方式：

ps -ef | grep nginx

kill -QUIT 主进程号 ：从容停止Nginx

kill -TERM 主进程号 ：快速停止Nginx

pkill -9 nginx ：强制停止Nginx

### 配置nginx.conf文件



Nginx 配置详解

<https://www.runoob.com/w3cnote/nginx-setup-intro.html>

<https://www.cnblogs.com/1214804270hacker/p/9325150.html>

<http://www.ha97.com/5194.html>

## 配置NFS共享

### 服务端配置

#### 1）关闭系统防火墙

查看防火墙状态：systemctl status firewalld.service

临时关闭防火墙：systemctl stop firewalld.service

临时打开防火墙：systemctl start firewalld.service

永久关闭防火墙：systemctl disable firewalld.service

#### 关闭selinux

临时关闭SELinux：setenforce 0

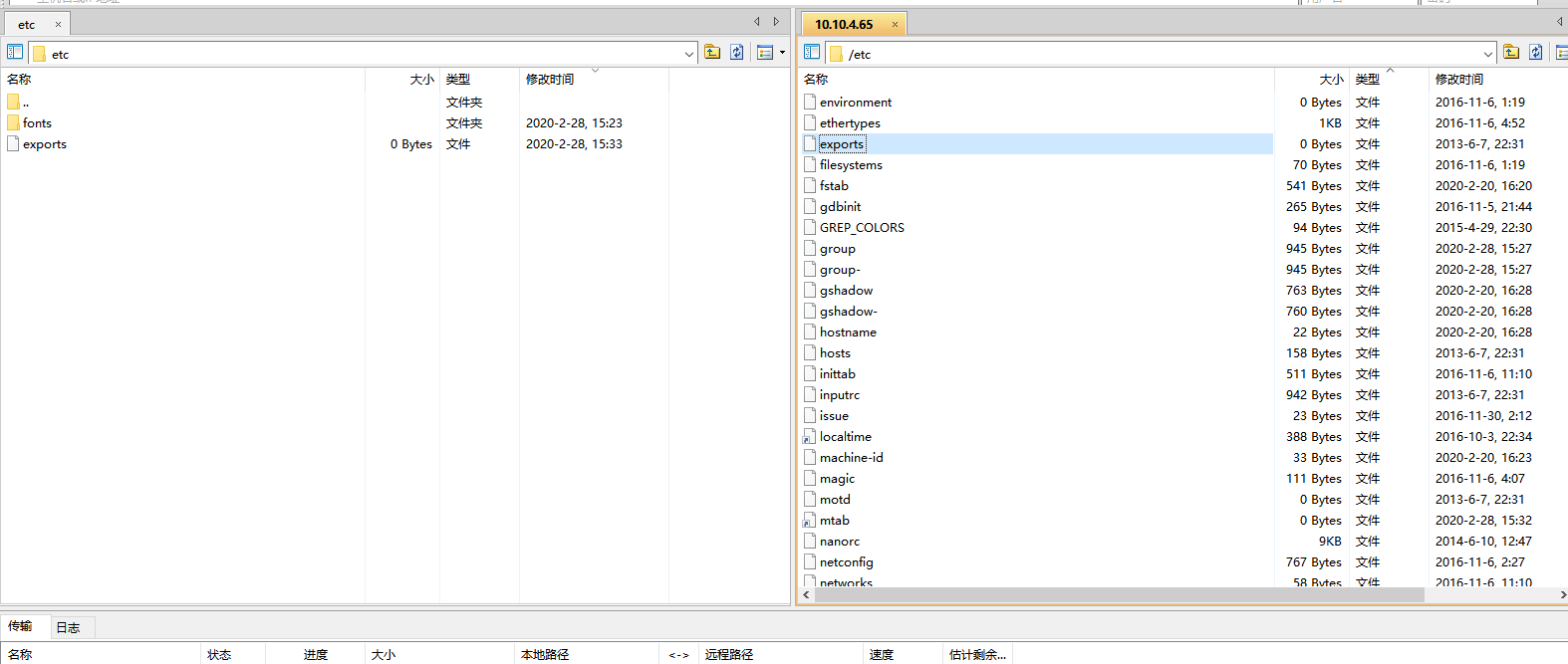
临时打开SELinux：setenforce 1

永久关闭SELinux：编辑/etc/sysconfig/selinux配置文件,把SELINUX设置成disabled,重启生效

#### 安装所需的软件包

执行命令：yum install -y rpc-bind nfs-utils

#### 配置共享文件目录，编辑配置文件





#### 启动服务和设置开启启动

启动nfs服务：systemctl start nfs

设置nfs服务开机启动：systemctl enable nfs

查看nfs服务的状态：systemctl status nfs

重启nfs服务：systemctl restart nfs

启动rpc服务：systemctl start rpcbind

设置rpc服务开机启动：systemctl enable rpcbind

查看rpc服务的状态：systemctl status rpcbind

重启nfs服务：systemctl restart rpcbind

查看本机共享的文件系统：exportfs

### 客户端配置

#### 1）客户端关闭防火墙和selinux,方法同上.

#### 2）关闭firewall

systemctl stop firewalld.service #停止firewall

systemctl disable firewalld.service #禁止firewall开机启动

firewall-cmd --state #查看默认防火墙状态（关闭后显示notrunning，开启后显示running）

#### 3）客户端安装NFS软件包,并把NFS服务设为开机自启动,方法同上.

#### 4）挂载共享的NFS文件系统

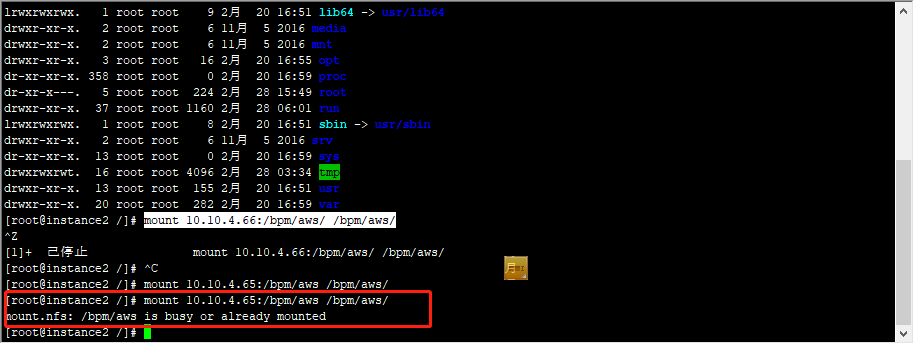
挂载的时候报错：mount.nfs: access denied by server while mounting

说明你没有在服务器端：修改/etc/exports文件,定义NFS共享

#把10.10.4.65这个主机上的共享目录挂载到两个实例服务器的/bpm/aws上

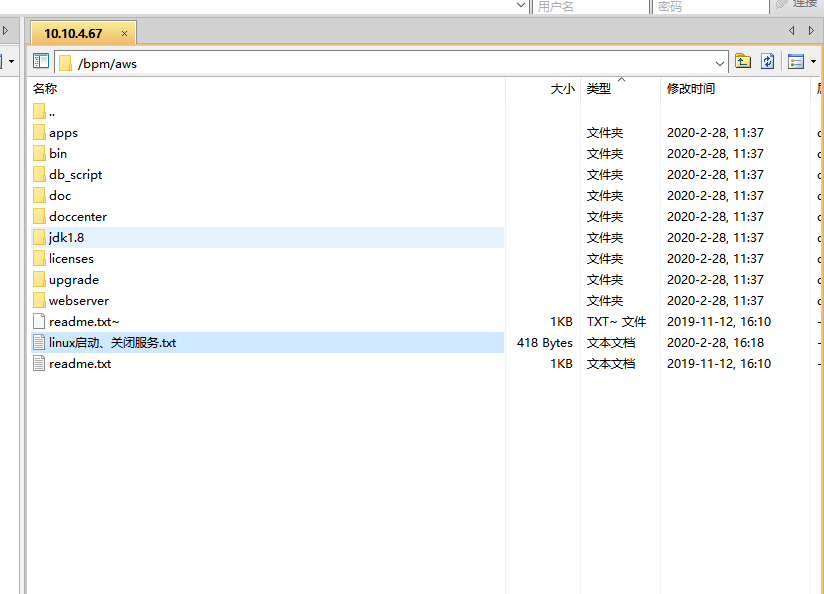
命令：mount 10.10.4.65:/bpm/aws /bpm/aws

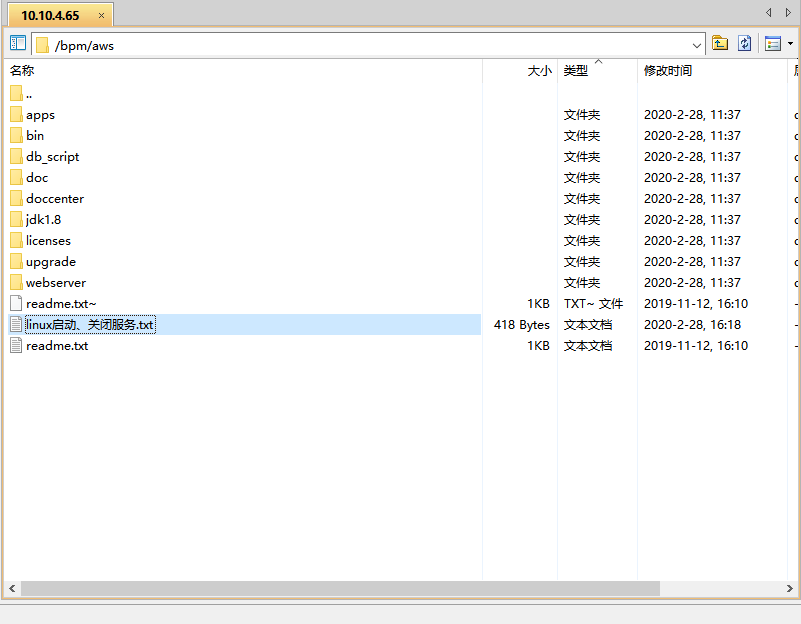
或者 mount -t nfs 10.10.4.65:/bpm/aws /bpm/aws

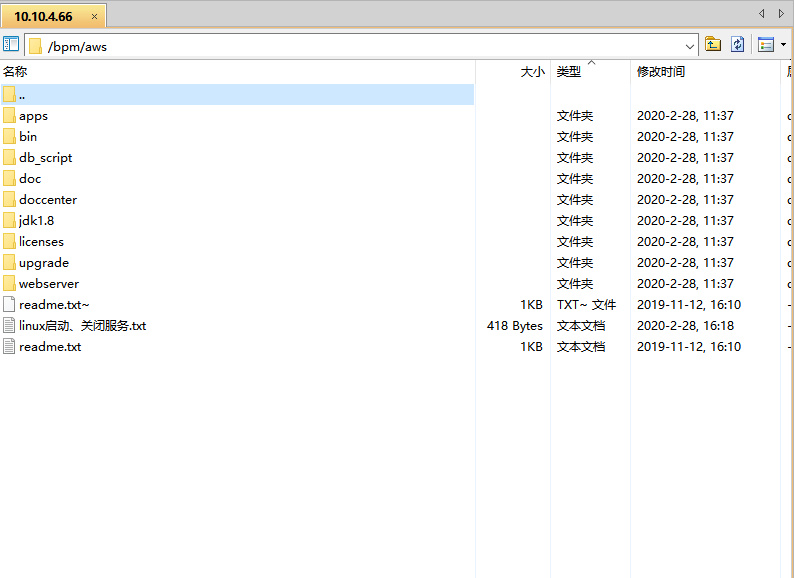
挂载成功

#### **4）把共享目录写入系统挂载文件系统**

### 测试共享是否成功







## 启动服务

进入192.168.100.10，新开两个shell窗口，分别执行如下操作，观察启动的控制台提示：

cd /data/cluster/

./httpd\_startup.sh

cd /data/cluster/

./aws\_startup.sh

进入192.168.100.20，新开两个shell窗口，执行上述同样的脚本。

## 五、验证服务

服务启动就绪后，在宿主机或其他用户主机打开浏览器，如下地址应可以正常访问：

http://192.168.100.10:8088/portal/console/

http://192.168.100.20:8088/portal/console/