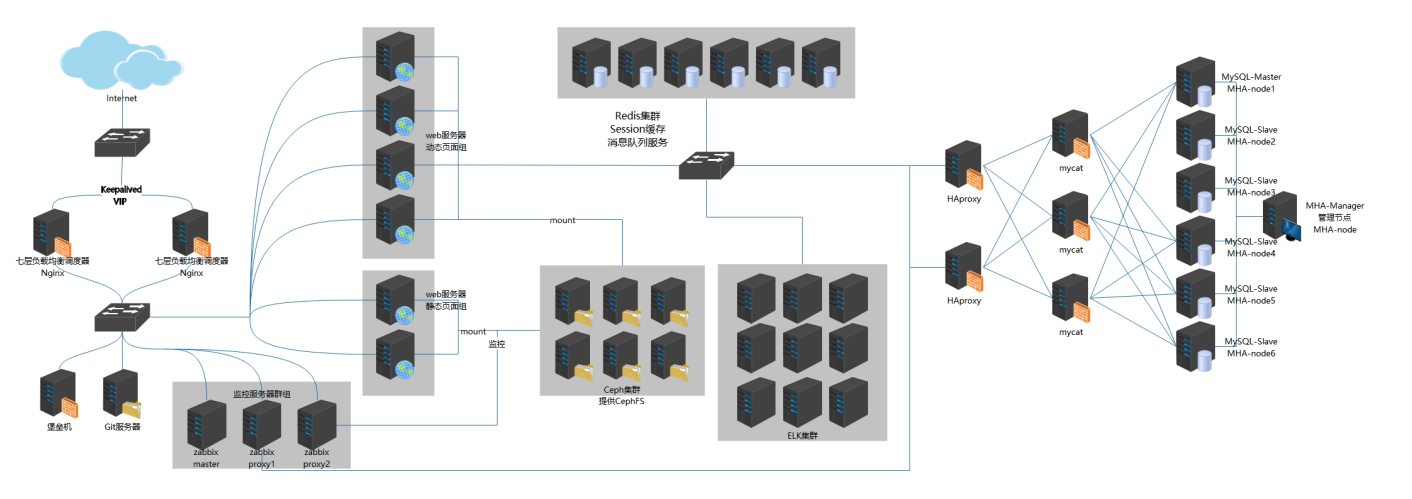
# 企业WEB服务器架构部署

1. TOPO&IP规划
2. TOPO



1. 服务器基本说明

所有主机系统均为CentOS7.5

堡垒机可用密码登录，其他所有内网机仅能使用密钥登录

堡垒机绑定弹性公网IP用于登录控制所有主机

模板机是所有内网主机的镜像模板

1. IP规划：

堡垒机：

jump-server；192.168.0.199

Git服务器：

git-server：192.168.0.198

七层负载均衡：

VIP：192.168.0.200 # VIP绑定一个公网IP：139.9.61.120

Nproxy0[1:2]：192.168.0.20[1:2]

WEB集群：

web0[1:4]；192.168.0.10[1:4] # 动态页面组

web0[5:6]；192.168.0.10[5:6] # web05,06为静态页面组

Redis集群：

Rnode0[1:6]；192.168.0.11[1:6]

Ceph集群：

Cnode0[1:6]；192.168.0.12[1:6]

ELK集群：

ES0[1:5]；192.168.0.13[1:5]

136-137 # 暂未定

Kibana01；192.168.0.39

Logstash01；192.168.0.38

数据库集群：

HAproxy0[1:2]；192.168.0.14[1:2]

Mycat0[1:3]；192.168.0.15[1:3]

Dnode0[1:6]；192.168.0.16[1:6]

MHA-Manager01；192.168.0.170

备：无

1. 具体部署实施
2. 堡垒机

命令：

mkdir -p /etc/yum.repos.d/repo\_bak/

mv /etc/yum.repos.d/\*.repo /etc/yum.repos.d/repo\_bak/

curl -o /etc/yum.repos.d/CentOS-Base.repo <http://mirrors.myhuaweicloud.com/repo/CentOS-Base-7.repo>

yum clean all

yum repolist

yum install lrzsz

yum install ansible

systemctl stop ntpd

systemctl stop postfix

yum remove postfix ntp

yum -y install chrony

vim /etc/chrony.conf

server ntp.myhuaweicloud.com iburst # 配置为华为云的时间服务器

systemctl restart chronyd

chronyc sources -v

备：

安装ftp

配置自定义yum源仓库，供内网主机使用

1. 模板机

命令：

mkdir -p /etc/yum.repos.d/repo\_bak/

mv /etc/yum.repos.d/\*.repo /etc/yum.repos.d/repo\_bak/

curl -o /etc/yum.repos.d/CentOS-Base.repo <http://mirrors.myhuaweicloud.com/repo/CentOS-Base-7.repo>

vim /etc/yum.repos.d/local.repo # 配置指定自定义yum源地址

[local]

name=localrepo

baseurl=ftp://192.168.0.199/myyum/

enabled=1

gpgcheck=0

yum clean all

yum repolist

yum install lrzsz

systemctl stop ntpd

systemctl stop postfix

yum remove postfix ntp

yum -y install chrony

vim /etc/chrony.conf

server ntp.myhuaweicloud.com iburst # 配置为华为云的时间服务器

systemctl restart chronyd

chronyc sources -v

备：无

1. 七层负载均衡调度器

命令：

yum -y install keepalived

vim /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id nproxy01 #nproxy02########

}

vrrp\_script chk\_nginx {

script "/etc/keepalived/nginx\_check.sh"

interval 2

weight -5

fall 2

rise 1

}

vrrp\_instance VI\_1 {

state MASTER #BACKUP##########

interface eth0

virtual\_router\_id 200

priority 100 #99##########

advert\_int 1

authentication {

auth\_type PASS

auth\_pass 6666

}

track\_script {

chk\_nginx

}

virtual\_ipaddress {

192.168.0.200

}

}

vim /etc/keepalived/nginx\_check.sh

#!/bin/bash

A=`ps -C nginx --no-header |wc -l`

if [ $A -eq 0 ];then

/usr/local/nginx/sbin/nginx

sleep 2

if [ `ps -C nginx --no-header |wc -l` -eq 0 ];then

killall keepalived

fi

fi

chmod +x /etc/keepalived/nginx\_check.sh

重启keal的playbook: # 由于经常要重启keepalived服务，所以我们在堡垒机上创建一个playbook来操作。

Ansible配置可参考(四)WEB集群中的ansible配置

vim reload-keal.yml

---

- hosts: nweb

remote\_user: root

tasks:

- name: reload keepalived

service:

state: restarted

name: keepalived

- shell: iptables -F

Nginx实现动静分离，这里不讲解如何安装nginx

vim /usr/local/nginx/conf/nginx.conf

http {

upstream dynamic\_websrv {

server 192.168.0.101 weight=2 max\_fails=1 fail\_timeout=20;

server 192.168.0.102 weight=2 max\_fails=1 fail\_timeout=20;

server 192.168.0.103 weight=2 max\_fails=1 fail\_timeout=20;

server 192.168.0.104 weight=2 max\_fails=1 fail\_timeout=20;

}

upstream static\_websrv {

server 192.168.0.105 weight=2 max\_fails=1 fail\_timeout=20;

server 192.168.0.106 weight=2 max\_fails=1 fail\_timeout=20;

}

server {

location / {

proxy\_pass http://dynamic\_websrv;

root html;

index index.html index.htm;

}

location ~ .\*.(gif|jpg|jpeg|png|bmp|swf|css|js)$ {

proxy\_pass http://static\_websrv;

}

}

}

备：

配置文件中#xxxx###############表示：当前配置为主负载均衡服务器的配置，而#中的xxxx为备负载均衡服务器所需要配置的配置项。

1. WEB集群

Ansible说明：

由于6台服务器web配置都一致，所以直接用ansible操作

以下命令无特殊标示时，默认在堡垒机上执行。

命令：

mkdir pl-web/

cd pl-web/

vim ansible.cfg

[defaults]

inventory = myhosts

host\_key\_checking = False

vim myhosts

[web:children]

dweb

sweb

nweb

[dweb]

web0[1:4]

[sweb]

web0[5:6]

[nweb]

Nproxy0[1:2]

vim i-web.yml

---

- hosts: web

remote\_user: root

tasks:

- name: install nginx,php,php-fpm,php-mysql

yum:

name: nginx,php,php-fpm,php-mysql

state: latest

- name: start service

service:

state: started

name: php-fpm

enabled: yes

ansible-playbook i-web.yml

vim e-web.yml

---

- hosts: web

remote\_user: root

tasks:

- name: edit nginx.conf

copy:

src: ./nginx.conf

dest: /usr/local/nginx/conf/nginx.conf

owner: root

group: root

mode: 0644

tags: copy

notify: reloadnginx

- name: copy test-page

copy:

src: ./test.php

dest: /usr/local/nginx/html/test.php

owner: nginx

group: nginx

mode: 0644

tags: send-test-page

notify: reloadphp

handlers:

- name: reloadnginx

shell: /usr/local/nginx/sbin/nginx || /usr/local/nginx/sbin/nginx -s reload

- name: reloadphp

service:

name: php-fpm

state: restarted

备：无

1. Ceph集群

在堡垒机上操作：

mkdir -p /etc/yum.repos.d/repo\_bak/

mv /etc/yum.repos.d/\*.repo /etc/yum.repos.d/repo\_bak/

curl -o /etc/yum.repos.d/CentOS-Base.repo http://mirrors.myhuaweicloud.com/repo/CentOS-Base-7.repo

wget -O /etc/yum.repos.d/epel.repo <http://mirrors.huaweicloud.com/repo/epel-7.repo>

vim /etc/yum.repos.d/ceph.repo

[ceph]

name=ceph

baseurl=http://mirrors.huaweicloud.com/ceph/rpm-jewel/el7/x86\_64/

gpgcheck=0

priority =1

[ceph-noarch]

name=cephnoarch

baseurl=http://mirrors.huaweicloud.com/ceph/rpm-jewel/el7/noarch/

gpgcheck=0

priority =1

[ceph-source]

name=Ceph source packages

baseurl=http://mirrors.huaweicloud.com/ceph/rpm-jewel/el7/SRPMS/

gpgcheck=0

priority=1

在安装之前先把node主机做基本配置

Cnode0[1:6]都配置IP+hostname的hosts，相互能免密ssh，时间同步

还需要配置内网node主机上网： # 这里我们通过堡垒机做代理上网

在堡垒机网卡选项里源/目的检查关闭，在堡垒机内开启路由转发，这里介绍都是临时配置，关机后就失效：

echo "1" > /proc/sys/net/ipv4/ip\_forward

配置地址转换：

iptables -t nat -A POSTROUTING -o eth0 -s 192.168.0.0/24 -j SNAT --to 192.168.0.199 # 199是堡垒机的内网IP

在控制台-网络控制台-找到对应的内网-添加一条路由-目的地址0.0.0.0/0，下一跳就是堡垒机IP，测试：ping baidu.com

还在堡垒机上执行： # 下面是安装

mkdir /ceph-cluster

cd /ceph-cluster

ceph-deploy new cnode01 cnode02 cnode03

ceph-deploy install cnode01 cnode02 cnode03 cnode04 cnode05 cnode06

# 安装过程如果报python-urllib3模块装不上，可能是因为本身就有安装或者是pip版本过底，只需要在执行ceph-deploy install报错后再把各个node主机上的python-urllib3模块和pip用pip install --upgrade XXX升级即可

ceph-deploy mon create-initial

for i in cnode0{1..3}; do ssh $i "parted /dev/vdb mklabel gpt;parted /dev/vdb mkpart primary 1M 50%;sleep 1;parted /dev/vdb mkpart primary 50% 100%" & done

for i in cnode0{1..3}; do ssh $i "chown ceph.ceph /dev/vdb1;chown ceph.ceph /dev/vdb2" & done

vim 70-vdb.rules

ENV{DEVNAME}=="/dev/vdb1",OWNER="ceph",GROUP="ceph"

ENV{DEVNAME}=="/dev/vdb2",OWNER="ceph",GROUP="ceph"

for i in cnode0{1..3}; do scp 70-vdb.rules $i:/etc/udev/rules.d/; done

cd /ceph-cluster/

ceph-deploy disk zap cnode01:vdc cnode01:vdd

ceph-deploy disk zap cnode02:vdc cnode02:vdd

ceph-deploy disk zap cnode03:vdc cnode03:vdd

ceph-deploy osd create cnode01:vdc:/dev/vdb1 cnode01:vdd:/dev/vdb2

ceph-deploy osd create cnode02:vdc:/dev/vdb1 cnode02:vdd:/dev/vdb2

ceph-deploy osd create cnode03:vdc:/dev/vdb1 cnode03:vdd:/dev/vdb2

ceph -s # 显示too few PGs per OSD (16 < min 30)

解决：

ceph osd pool set rbd pg\_num 128

ceph osd pool set rbd pgp\_num 128

将配置文件和密码共享给mds和client：

ceph-deploy admin [clientIP|clientHostname]

配置mds文件系统：

ceph-deploy mds create cnode04 cnode05 cnode06

ceph osd pool create cephfs\_data 128 # 可以再cnode4-6的任意一台执行

ceph osd pool create cephfs\_metadata 128 # 可以再cnode4-6的任意一台执行

ceph fs new myfs1 cephfs\_metadata cephfs\_data

ceph fs ls # 查看文件系统信息

ceph mds stat # 查看文件系统状态

客户端挂载：

yum -y install ceph-common

vim /etc/fstab

192.168.0.121:6789:/ /usr/local/nginx/html/ ceph name=admin,secretfile=/etc/ceph/secret.key,noatime 0 2

# 这里把cephfs挂载到每个web服务器的网页根目录

vim /etc/ceph/secret.key

AQDI1c9c9zyyLxAAS75B4rh3hJjQXgKXPE0qYA==

# 这串密码可以用cat /etc/ceph/ceph.client.admin.keyring 查看

1. 数据库集群

说明:在部署数据库集群Dnode0[1:6]时,使用ansible批量部署.

命令：

在堡垒机上:

mkdir work

cd work

vim ansible.cfg

[defaults]

inventory = myhosts

host\_key\_checking = False

vim myhosts

[node]

192.168.1.16[1:6]

[mag]

192.168.1.170

ansible all -m authorized\_key -a "user=root exclusive=true manage\_dir=true key='$(< /root/.ssh/id\_rsa.pub)'" -k # 需要免密才需要敲这一条

# 需要MySQL的软件包

备：无

1. ELK集群

命令：

备：无

1. Redis集群-消息队列集群

命令：

备：无

1. Zabbix监控服务器

命令：

备：无

1. Git服务器

命令：

备：无

备：无

1. 总结