一、搭建swarm集群

|  |  |  |  |
| --- | --- | --- | --- |
| 主机 | 类型 | 主机用户名 | 主机密码 |
| 172.16.0.21 | centos | docker | O9i8u7y6 |

第一步 、初始化swarm集群:

docker swarm init --advertise-addr 172.16.0.21

第二步 、获取节点token:

docker swarm join-token manager||worker

第三步 、加入节点到swarm集群:

docker swarm join --token token\_value

注：根据第二步获取token值替换 token\_value

二、划分swarm集群网络

设置集群子网，分别为，coss、守护服务、租户、应用

|  |  |  |  |
| --- | --- | --- | --- |
| 网络名称 | 类型 | 应用范围 | 执行命令 |
| vlan\_coss | overlay | coss | docker network create --driver overlay --subnet 10.0.0.1/24 vlan\_coss |
| vlan\_daemon | overlay | 守护服务 | docker network create --driver overlay --subnet 10.0.0.2/24 vlan\_daemon |
| vlan\_tenant | overlay | 租户 | docker network create --driver overlay --subnet 10.0.0.3/24 vlan\_tenant |
| vlan\_app | overlay | 应用 | docker network create --driver overlay --subnet 10.0.0.4/24 vlan\_app |

三、设置数据卷保存数据

按使用范围设置的数据卷

|  |  |  |  |
| --- | --- | --- | --- |
| 数据卷名称 | 应用范围 | 执行命令 | 备注 |
| resource\_app | 应用包 | docker volume create --driver local --name resource\_app | jar|war|tar |
| resource\_file | 文件 | docker volume create --driver local --name resource\_file | 图片|其他 |
| resource\_registry | 私有镜像库 | docker volume create --driver local --name resource\_registry | 镜像存在 |
| resource\_swarm\_data | Swarm ui | docker volume create --driver local --name resource\_swarm\_data | swarm ui 数据 |

四、部署私有镜像库

第一步、创建镜像库：

指定数据卷、对外端口。

docker run -d --name --restart=always registry -p 50001:5000 -v resource\_registry:/var/lib/registry registry:2

第一步、设置镜像库域名：

更改/etc/docker/daemon.json 配置文件，设置授信域名让其他docker宿主机可以访问。

{

"insecure-registries":[ "registry.ib2000.net:50001"]

}

五、部署swarm集群ui

指定数据卷、对外端口。

|  |
| --- |
| docker service create \  --name swarm\_console \  --publish 9001:9000 \  --replicas 1 \  --constraint 'node.role == manager' \  --mount type=bind,src=/var/run/docker.sock,dst=/var/run/docker.sock \  --mount type=volume,src=resource\_swarm\_data,dst=/data \  registry.ib2000.net:50001/portainer \  -H unix:///var/run/docker.sock |

六、搭建nginx代理

搭建nginx代理，分外层nginx和里层nginx，外层nginx映射里层nginx，里层nginx代理swarm service服务。

**第一步、搭建外层nginx：**

1、下载对应当前系统版本的nginx包(package)

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

2、建立nginx的yum仓库

rpm -ivh nginx-release-centos-7-0.el7.ngx.noarch.rpm

3、下载并安装nginx

yum install nginx

4、启动nginx服务

systemctl start nginx

5、添加证书

复制证书到/etc/nginx/crt/ 目录

/etc/nginx/crt/www.ib2000.net\_bundle.crt

/etc/nginx/crt/www.ib2000.net.key

6、修改配置文件

|  |  |
| --- | --- |
| /etc/nginx/nginx.conf | /etc/nginx/conf.d/www.ib2000.conf |
| user docker;  worker\_processes 1;  error\_log /var/log/nginx/error.log warn;  pid /var/run/nginx.pid;  events {  worker\_connections 1024;  }  http {  include /etc/nginx/mime.types;  default\_type application/octet-stream;  log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '  '$status $body\_bytes\_sent "$http\_referer" '  '"$http\_user\_agent" "$http\_x\_forwarded\_for"';  access\_log /var/log/nginx/access.log main;  sendfile on;  #tcp\_nopush on;  keepalive\_timeout 65;  #gzip on;  include /etc/nginx/conf.d/\*.conf;  fastcgi\_connect\_timeout 300;  fastcgi\_send\_timeout 300;  fastcgi\_read\_timeout 300;  fastcgi\_buffer\_size 256k;  fastcgi\_buffers 4 128k;  fastcgi\_busy\_buffers\_size 256k;  fastcgi\_temp\_file\_write\_size 256k;  } | upstream tomcat\_portal {  ip\_hash;  server www.ib2000.net:82 weight=5;  }  server {  listen 80;  server\_name www.ib2000.net;  location / {  root html;  index index.html index.htm index.jsp;  proxy\_pass http://tomcat\_portal$request\_uri;  proxy\_redirect off;  proxy\_set\_header Cookie $http\_cookie;  proxy\_set\_header Host $host:$server\_port;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  client\_max\_body\_size 10m;  client\_body\_buffer\_size 128k;  proxy\_connect\_timeout 90;  proxy\_send\_timeout 90;  proxy\_read\_timeout 90;  proxy\_buffer\_size 4k;  proxy\_buffers 4 32k;  proxy\_busy\_buffers\_size 64k;  proxy\_temp\_file\_write\_size 64k;  }  }  server {  listen 443;  server\_name www.ib2000.net;  ssl on;  ssl\_certificate crt/www.ib2000.net\_bundle.crt; #sslkey/www.ib2000.net\_bundle.crt;  ssl\_certificate\_key crt/www.ib2000.net.key; #sslkey/www.ib2000.net.key;  ssl\_session\_timeout 5m;  ssl\_protocols TLSv1 TLSv1.1 TLSv1.2;  ssl\_ciphers AESGCM:ALL:!DH:!EXPORT:!RC4:+HIGH:!MEDIUM:!LOW:!aNULL:!eNULL;  ssl\_prefer\_server\_ciphers on;  location / {  root html;  index index.html index.htm index.jsp;  proxy\_pass http://tomcat\_portal;  proxy\_redirect off;  proxy\_set\_header Cookie $http\_cookie;  proxy\_set\_header Host $host;  proxy\_set\_header X-Real-IP $remote\_addr;  proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;  client\_max\_body\_size 10m;  client\_body\_buffer\_size 128k;  proxy\_connect\_timeout 90;  proxy\_send\_timeout 90;  proxy\_read\_timeout 90;  proxy\_buffer\_size 4k;  proxy\_buffers 4 32k;  proxy\_busy\_buffers\_size 64k;  proxy\_temp\_file\_write\_size 64k;  }  } |

7、重新加载配置文件

/usr/sbin/nginx -s reload

**第二步、搭建里层nginx：**

1、部署nginx到swarm集群中

|  |
| --- |
| docker service create --name service\_nginx \  --constraint "node.role == manager" \  --mount type=bind,src=/home/docker/config/nginx/service/nginx.conf,dst=/etc/nginx/nginx.conf \  --publish 82:80 \  --network vlan\_daemon \  --host "www.ib2000.net:172.16.0.21" \  nginx:latest |

2、修改配置文件

|  |
| --- |
| /home/docker/config/nginx/service/nginx.conf |
| user nginx;  worker\_processes 8;    error\_log /var/log/nginx/error.log warn;  pid /var/run/nginx.pid;  events {  worker\_connections 1024;  }  http {  include /etc/nginx/mime.types;  default\_type application/octet-stream;  log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '  '$status $body\_bytes\_sent "$http\_referer" '  '"$http\_user\_agent" "$http\_x\_forwarded\_for"';  access\_log /var/log/nginx/access.log main;  resolver 127.0.0.11;  sendfile on;  #tcp\_nopush on;  keepalive\_timeout 65;  #gzip on;  # include /etc/nginx/conf.d/\*.conf;  server {  listen 80;  server\_name www.ib2000.net;  #charset koi8-r;  #access\_log /var/log/nginx/host.access.log main;  #location / {  # root /usr/share/nginx/html;  # index index.html index.htm;  #}  #非托管，追加文件.test  #include /etc/nginx/conf.d/\*.test;  #托管  #location ~ ^/(css|img|js|tpl)/ {  # proxy\_pass http://192.168.100.88/saas\_web\_admin$request\_uri;  #}  #rewrite ^/(.\*) http://www.ib2000.net/saas\_web\_cas/login permanent;  location / {  #proxy\_set\_header Host www.ib2000.net;  proxy\_set\_header Host $host:$server\_port;  rewrite ^/ http://www.ib2000.net/saas\_web\_cas/login redirect;  }  location ~ /(\w+)/ {  proxy\_set\_header Host www.ib2000.net;  proxy\_pass http://$1$request\_uri;  }  }  fastcgi\_connect\_timeout 300;  fastcgi\_send\_timeout 300;  fastcgi\_read\_timeout 300;  fastcgi\_buffer\_size 256k;  fastcgi\_buffers 4 128k;  fastcgi\_busy\_buffers\_size 256k;  fastcgi\_temp\_file\_write\_size 256k;  } |

3、重新加载nginx

docker service scale service\_nginx=0

docker service scale service\_nginx=1

七、部署服务

部署coss|paas|saas服务

|  |  |
| --- | --- |
| 服务名称 | 执行脚本 |
| coss | /home/docker/docker\_coss.sh |
| paas | /home/docker/docker\_paas.sh |
| saas | /home/docker/docker\_saas.sh |

|  |  |  |  |
| --- | --- | --- | --- |
| 服务脚本 | 示例 | | 备注 |
| /home/docker/swarm\_service.sh | 启动服务 | sh swarm\_service.sh start -w coss1 | 去掉-w参数时为模糊匹配 |
| 停止服务 | sh swarm\_service.sh stop -w coss1 |
| 删除服务 | sh swarm\_service.sh rm -w coss1 |

八、注意事项

1、docker默认存储驱动

"storage-driver":"overlay",

2、 修改docker默认存储目录

"data-root":"/home/docker/docker\_data",

3、添加docker 镜像下载加速

"registry-mirrors": ["[https://registry.docker-cn.com](https://registry.docker-cn.com/)"],

4、添加授信镜像库

"insecure-registries":[ "registry.ib2000.net:50001"]

5、开放2375端口，以便调用docker api

"hosts": ["tcp://0.0.0.0:2375","unix:///var/run/docker.sock"],

6、centos 系统 docker /etc/docker/daemon.json 配置文件修改如下

|  |
| --- |
| {  "debug":true,  "registry-mirrors": ["https://registry.docker-cn.com"],  "storage-driver":"overlay",  "data-root":"/home/docker/docker\_data",  "hosts": ["tcp://0.0.0.0:2375","unix:///var/run/docker.sock"],  "insecure-registries":[ "registry.ib2000.net:50001"]  } |

7、centos 系统 下载并配置nginx时，修改了nginx.conf配置文件，重启nginx服务，重启nginx时，/var/run/nginx.pid会出现权限不足问题。

解决方案为：/usr/sbin/nginx -s reload 重新加载配置文件

8、centos 系统防火墙问题，即部署docker swarm 服务时，服务访问外部服务出现host unreachable 。

解决方案为：替换centos 防火墙未iptables，并检查/etc/sysconfig/iptables是否有无映射规则，如无则添加映射规则，然后重启防火墙service iptables restart

如：

|  |
| --- |
| # Generated by iptables-save v1.4.21 on Thu Sep 21 20:09:13 2017  \*mangle  :PREROUTING ACCEPT [557:42443]  :INPUT ACCEPT [557:42443]  :FORWARD ACCEPT [0:0]  :OUTPUT ACCEPT [344:49373]  :POSTROUTING ACCEPT [344:49373]  COMMIT  # Completed on Thu Sep 21 20:09:13 2017  # Generated by iptables-save v1.4.21 on Thu Sep 21 20:09:13 2017  \*nat  :PREROUTING ACCEPT [2:128]  :INPUT ACCEPT [2:128]  :OUTPUT ACCEPT [3:228]  :POSTROUTING ACCEPT [3:228]  :DOCKER - [0:0]  :DOCKER-INGRESS - [0:0]  -A PREROUTING -m addrtype --dst-type LOCAL -j DOCKER-INGRESS  -A PREROUTING -m addrtype --dst-type LOCAL -j DOCKER  -A OUTPUT -m addrtype --dst-type LOCAL -j DOCKER-INGRESS  -A OUTPUT ! -d 127.0.0.0/8 -m addrtype --dst-type LOCAL -j DOCKER  -A POSTROUTING -o docker\_gwbridge -m addrtype --src-type LOCAL -j MASQUERADE  -A POSTROUTING -s 172.17.0.0/16 ! -o docker0 -j MASQUERADE  -A POSTROUTING -s 172.19.0.0/16 ! -o docker\_gwbridge -j MASQUERADE  -A POSTROUTING -s 172.17.0.2/32 -d 172.17.0.2/32 -p tcp -m tcp --dport 5000 -j MASQUERADE  -A DOCKER -i docker0 -j RETURN  -A DOCKER -i docker\_gwbridge -j RETURN  -A DOCKER ! -i docker0 -p tcp -m tcp --dport 50001 -j DNAT --to-destination 172.17.0.2:5000  -A DOCKER-INGRESS -p tcp -m tcp --dport 18086 -j DNAT --to-destination 172.19.0.2:18086  -A DOCKER-INGRESS -p tcp -m tcp --dport 82 -j DNAT --to-destination 172.19.0.2:82  -A DOCKER-INGRESS -p tcp -m tcp --dport 18084 -j DNAT --to-destination 172.19.0.2:18084  -A DOCKER-INGRESS -p tcp -m tcp --dport 18085 -j DNAT --to-destination 172.19.0.2:18085  -A DOCKER-INGRESS -p tcp -m tcp --dport 18083 -j DNAT --to-destination 172.19.0.2:18083  -A DOCKER-INGRESS -p tcp -m tcp --dport 9001 -j DNAT --to-destination 172.19.0.2:9001  -A DOCKER-INGRESS -p tcp -m tcp --dport 18082 -j DNAT --to-destination 172.19.0.2:18082  -A DOCKER-INGRESS -p tcp -m tcp --dport 18089 -j DNAT --to-destination 172.19.0.2:18089  -A DOCKER-INGRESS -j RETURN  COMMIT  # Completed on Thu Sep 21 20:09:13 2017  # Generated by iptables-save v1.4.21 on Thu Sep 21 20:09:13 2017  \*filter  :INPUT ACCEPT [0:0]  :FORWARD DROP [0:0]  :OUTPUT ACCEPT [31:3011]  :DOCKER - [0:0]  :DOCKER-INGRESS - [0:0]  :DOCKER-ISOLATION - [0:0]  :DOCKER-USER - [0:0]  -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT  -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT  -A INPUT -p tcp -m tcp --dport 50001 -j ACCEPT  -A INPUT -p tcp -m tcp --dport 2375 -j ACCEPT  -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT  -A INPUT -p icmp -j ACCEPT  -A INPUT -i lo -j ACCEPT  -A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT  -A INPUT -j REJECT --reject-with icmp-host-prohibited  -A FORWARD -j DOCKER-USER  -A FORWARD -j DOCKER-INGRESS  -A FORWARD -j DOCKER-ISOLATION  -A FORWARD -o docker0 -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT  -A FORWARD -o docker0 -j DOCKER  -A FORWARD -i docker0 ! -o docker0 -j ACCEPT  -A FORWARD -i docker0 -o docker0 -j ACCEPT  -A FORWARD -o docker\_gwbridge -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT  -A FORWARD -o docker\_gwbridge -j DOCKER  -A FORWARD -i docker\_gwbridge ! -o docker\_gwbridge -j ACCEPT  -A FORWARD -j REJECT --reject-with icmp-host-prohibited  -A FORWARD -i docker\_gwbridge -o docker\_gwbridge -j DROP  -A DOCKER -d 172.17.0.2/32 ! -i docker0 -o docker0 -p tcp -m tcp --dport 5000 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m tcp --dport 18086 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m state --state RELATED,ESTABLISHED -m tcp --sport 18086 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m tcp --dport 82 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m state --state RELATED,ESTABLISHED -m tcp --sport 82 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m tcp --dport 18084 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m state --state RELATED,ESTABLISHED -m tcp --sport 18084 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m tcp --dport 18085 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m state --state RELATED,ESTABLISHED -m tcp --sport 18085 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m tcp --dport 18083 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m state --state RELATED,ESTABLISHED -m tcp --sport 18083 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m tcp --dport 9001 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m state --state RELATED,ESTABLISHED -m tcp --sport 9001 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m tcp --dport 18082 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m state --state RELATED,ESTABLISHED -m tcp --sport 18082 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m tcp --dport 18089 -j ACCEPT  -A DOCKER-INGRESS -p tcp -m state --state RELATED,ESTABLISHED -m tcp --sport 18089 -j ACCEPT  -A DOCKER-INGRESS -j RETURN  -A DOCKER-ISOLATION -i docker\_gwbridge -o docker0 -j DROP  -A DOCKER-ISOLATION -i docker0 -o docker\_gwbridge -j DROP  -A DOCKER-ISOLATION -j RETURN  -A DOCKER-USER -j RETURN  COMMIT  # Completed on Thu Sep 21 20:09:13 2017 |

9、其他docker宿主机，访问私有镜像库时不能访问问题。

解决方案为：检查/etc/docker/daemon.json配置文件是否存在如下配置

"insecure-registries":[ "registry.ib2000.net:50001"]

10、ubuntu 16.04 系统不能加载/etc/docker/daemon.json配置文件或 DOCKER\_OPTS不起作用。

解决方案:

|  |
| --- |
| ##ubuntu16.04 docker 启动错误 systemctl status docker.service  ##DOCKER\_OPTS 不起作用  #修改： sudo vim /lib/systemd/system/docker.service    EnvironmentFile=/etc/default/docker  ExecStart=/usr/bin/docker daemon -H fd:// $DOCKER\_OPTS  or  EnvironmentFile=/etc/docker/daemon.json  ExecStart=/usr/bin/dockerd    #重启  systemctl daemon-reload  sudo service docker restart |

11、nfs 共享目录，文件不能共享问题

解决方案：检查nfs客户端是安装

whereis nfs-common

sudo apt-get install nfs-common

12、图片服务器nginx映射，图片找不到问题

解决方案：使用nginx 别名

|  |
| --- |
| server {  listen 80;  server\_name images.ib2000.net;  location /image {  alias /home/docker/docker\_data/volumes/resource\_file/\_data;  }  } |

13、z\_stack 搭建swawrm 集群时出现

"agent: session failed" error="rpc error: code = 10 desc = dispatcher is stopped"

error "memberlist ping 172.16.0.51:12456:7946 ...."

解决方案：检查网络资源是否充足，docker swarm 底层 rpc ping 频繁

14、docker 日志文件

|  |  |
| --- | --- |
| 系统类型 | 文件目录 |
| ubuntu14.04 | /var/log/upstart/docker.log |
| ubuntu16.04 | /var/log/sysconfig.log |
| Coentos 7 | /var/log/messages |