Vpclub 项目运维

# 1.服务器OS环境:

Ubuntu 14.04.4 LTS (GNU/Linux 3.19.0-59-generic x86\_64)

＃查看系统版本号

lsb\_release -a

安装git

sudo apt-get install git

使用git 命令确认服务器是否安装全内核，确保能安装docker

修改dns文件，确保能访问ubuntu 网站

sudo vim /etc/resolv.conf

##add line

# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)

# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN

nameserver 223.5.5.5

nameserver 8.8.4.4

nameserver 8.8.8.8

修改dns文件，确保能访问ubuntu 网站（永久）

sudo vim /etc/network/interfaces

auto eth0

iface eth0 inet static

address 172.16.\*.\* (此ip为具体机器ｉｐ，每台不一样)

netmask 255.255.0.0

gateway 172.16.0.1

dns-nameservers 172.16.0.1 223.5.5.5 8.8.4.4 8.8.8.8

broadcast 255.255.255.255

network 172.16.0.0

# nameserver 8.8.4.4

# nameserver 8.8.8.8

# 2.部署工具:

docker ,docker-compose,allmake

安装 及使用 参考：

[https://docs.docker.com/engine/installation/linux/ubuntulinux/](https://docs.docker.com/)

## 2.1. docker 安装（只介绍ubuntu OS 其它请参照官方文档）

2.2. docker-compose 安装（只介绍ubuntu OS 其它请参照官方文档）

安装 及使用 参考：<https://docs.docker.com/compose/install/>

①网络安装方式  
编写一键安装脚本install-docker.sh

|  |
| --- |
| vim install-docker.sh |
| #!/bin/bash  #Run as root user  echo " Checking user..."  sleep 1  if [ $UID != 0 ];then  echo "please run as root"  exit  fi  sleep 1  export APT\_INSTALL='sudo apt-get --assume-yes install'  export APT\_UPDATE='sudo apt-get --assume-yes update'  sudo chown $USER:$USER /etc/resolv.conf  echo "nameserver 223.5.5.5" >> /etc/resolv.conf  sudo chown root:root /etc/resolv.conf  $APT\_INSTALL git  #install bridge-utils $APT\_INSTALL bridge-utils $APT\_INSTALL apt-transport-https ca-certificates  apt-key adv --keyserver hkp://p80.pool.sks-keyservers.net:80 --recv-keys 58118E89F3A912897C070ADBF76221572C52609D  echo "deb https://apt.dockerproject.org/repo ubuntu-trusty main" > /etc/apt/sources.list.d/docker.list  $APT\_UPDATE  apt-get purge lxc-docker  apt-cache policy docker-engine  $APT\_UPDATE  $APT\_INSTALL linux-image-extra-$(uname -r) linux-image-extra-virtual  $APT\_UPDATE  $APT\_INSTALL docker-engine  service docker start  docker -v  groupadd docker  usermod -aG docker $USER  $APT\_INSTALL python-pip  sudo pip install docker-compose  sudo docker-compose -v |

#授权

sudo chmod 755 install-docker.sh

#执行安装即可

sudo ./install-docker.sh

②源码安装方式

mkdir install-docker

cd install-docker

|  |
| --- |
| vim install-docker-src.sh |
| #!/bin/bash  if [ $($UID) != 0 ];then  echo "please run as root"  exit  fi  os=$(echo $(lsb\_release -i))  if [ "$os" == "Distributor ID: CentOS" ]; then  　echo "current os is $os"  export APT\_INSTALL='sudo yum --assume-yes install'  export APT\_UPDATE='sudo yum --assume-yes update'  export APT\_PURGE='sudo yum --assume-yes purge'  fi  if [ "$os" == "Distributor ID: Ubuntu" ]; then  　echo "current os is $os"  export APT\_INSTALL='sudo apt-get --assume-yes install'  export APT\_UPDATE='sudo apt-get --assume-yes update'  export APT\_PURGE='sudo apt-get --assume-yes purge'  fi  APT\_INSTALL git  GIT\_SRC=$PWD/src/git.tar.gz  DOCKER\_SRC=$PWD/src/docker-latest.tgz  function git-install(){  if [ ! -f "$GIT\_SRC" ]; then  echo "no src $GIT\_SRC"  wget https://github.com/git/git/archive/v2.1.2.tar.gz -O git.tar.gz  else  echo "has src"  cp src/git.tar.gz .  fi  tar -zxf git.tar.gz  cd git-\*  make configure  ./configure --prefix=/usr/local  sudo make install  git --version  }  function docker-install(){  if [ ! -f "$DOCKER\_SRC" ]; then  echo "no src $DOCKER\_SRC"  wget https://get.docker.com/builds/Linux/x86\_64/docker-latest.tgz  else  echo "has src"  cp src/docker-latest.tgz .  fi  tar -xvzf docker-latest.tgz  sudo mv docker/\* /usr/bin/  sudo dockerd &  } |

#授权

sudo chmod 755 install-docker-src.sh

#执行安装即可

sudo ./install-docker-src.sh

2.3. allmake 安装

cd /opt

#download

git clone <https://github.com/allmake/allmake.git>

#install

cd allmake

sudo ./bin/allmake -i

## 2.4 配置本地镜像库

Client Operation

**firstly:**

(如果是内网自己配的域名则需要配置)

add host config

window path [c:/window/system32/drivers/etc/hosts](file:///c:/window/system32/drivers/etc/hosts)

Linux path /etc/hosts

120.76.43.11 hub.docker.vpclub.cn

1:create dir

sudo mkdir /usr/local/share/ca-certificates/docker-dev-cert

2:create file and paste the certificate

sudo rm /usr/local/share/ca-certificates/docker-dev-cert/devdockerCA.crt

sudo nano /usr/local/share/ca-certificates/docker-dev-cert/devdockerCA.crt

#add cert key

-----BEGIN CERTIFICATE-----

MIID6zCCAtOgAwIBAgIJAKAbyZuhUQJiMA0GCSqGSIb3DQEBCwUAMIGLMQswCQYD

VQQGEwJaSDELMAkGA1UECAwCR0QxCzAJBgNVBAcMAlNaMQ8wDQYDVQQKDAZWUENM

VUIxDzANBgNVBAsMBlZQQ0xVQjEdMBsGA1UEAwwUaHViLmRvY2tlci52cGNsdWIu

Y24xITAfBgkqhkiG9w0BCQEWEmNoZW4ud2VpQHZwY2x1Yi5jbjAeFw0xNjA3MTYw

ODI3NTRaFw00MzEyMDIwODI3NTRaMIGLMQswCQYDVQQGEwJaSDELMAkGA1UECAwC

R0QxCzAJBgNVBAcMAlNaMQ8wDQYDVQQKDAZWUENMVUIxDzANBgNVBAsMBlZQQ0xV

QjEdMBsGA1UEAwwUaHViLmRvY2tlci52cGNsdWIuY24xITAfBgkqhkiG9w0BCQEW

EmNoZW4ud2VpQHZwY2x1Yi5jbjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC

ggEBAMPZRrO/Fc7PrY94uiiSl4dA4r5ko0ca5j9IbW/8wmlEwLWtMXd/+ERtsDr2

mMIkSuWxY6T5MJft2Fg4iogiaRZuS/6Vqub1jqd61A9LToCfYr2Z3jrQaC8otWI6

uN83MWcmJTgbjqONO2xwVcuWfncXIClvhWWSIYBO60kFaq6RZDFdSS1Qz6W2m+mB

U6c0B8ueUz3KE8CLGsnFVT6GXq82qsQSWM5FXeAjaJT1LlPXMk8JoQ0+JyGJYV+P

1ZcDPztD/afo9hvdJIgjN1gfdE+eEJrpnD+G3EL8sbiNKa4NuOyzEp2EnOvT30B5

1wW3VaXh2UMdvgYUUubBz3fVrTsCAwEAAaNQME4wHQYDVR0OBBYEFC88ycrUhiom

u425i3EWN+sw2HNJMB8GA1UdIwQYMBaAFC88ycrUhiomu425i3EWN+sw2HNJMAwG

A1UdEwQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEBALV9NZ9nDdDpV3pcuRXCSzw9

39NJ7r1Merso3u91USOMokySU6nUsuRZ/A8tNg2s7HlRMMRvTqynDh0asT3rcK/C

m7d6Em+Benil6Esyb3D2zJFBAM1dsbwFAwpkbuxnQK2INJReAaOv6ijurkatz5ic

U7XDJxHgmI9QzBQRzKzPalzCFFC8nfwUHlZ8jluwRb77ctxxC3t+RXJatOmmCQ5q

KXPAQjwZMSgqN62hUbMhR98HQy/s/eo1DSiCQ6hRM/Cvx+aAqmvf/ujbd/BjbZ/c

X6Ep6ON2tOhY1YeEqLGy7qmCvRdz08Rq4PYWmlUYiXddEiA7BSi/8uZx1bQLO/k=

-----END CERTIFICATE-----

3. update the certificates

sudo update-ca-certificates

4.restart docker

sudo service docker restart

5.login

sudo docker login [https://hub.docker.vpclub.cn](https://hub.docker.vpclub.cn/)

(账户密码需要找陈伟13337998835)

6. create image and push(上一步loginSucceed则无需执行此步)

docker tag hello-world hub.docker.vpclub.cn/test-image

sudo docker push hub.docker.vpclub.cn/test-image

＃ｒｅｇｉｓｔｒｙ日志查看

sudo tail -f /var/log/upstart/docker-registry.log

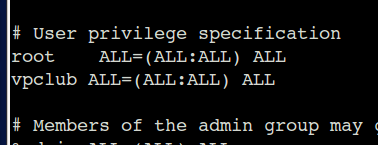
# 3.创建用户

adduser vpclub

visudo

#添加sudo 权限在root行下面添加（如下图）

vpclub ALL=(ALL:ALL) ALL



# 4.部署常用命令

1.首先创建部署根目录

mkdir -p deployment/backservices/

cd deployment/backservices/

①创建（更新）部署目录 执行 deploy.sh脚本(vim deploy.sh 并授执行权限sudo chmod 755 deploy.sh)　(６个参数)

#!/bin/bash

inet=$1

if [ "$inet" == "" ]; then

inet=eth0

fi

profile=$2

if [ "$profile" == "" ]; then

profile=test

fi

tmp=$3

if [ "$tmp" == "" ]; then

tmp=java

fi

jmx=$4

if [ "$jmx" == "" ]; then

jmx=128m

fi

ver=$5

if [ "$ver" == "" ]; then

ver=1.0.0

fi

logip=$6

if [ "$logip" == "" ]; then

logip=172.16.45.3:5000

fi

function addNewApp()

{

allmake add app=$1 ver=$ver profile=$profile ports="$2" template=$tmp inet=$inet jmx=$4 logIpAddress=$logip

}

mkdir -p 01-provider 02-consumer

pushd 01-provider

addNewApp vp-traffic-monetisation-provider "10010 23010 33010" "" 128m

addNewApp vp-traffic-monetisation-adapter-provider "10030 23030 33030" "" 128m

popd

pushd 02-consumer

addNewApp vp-traffic-monetisation-consumer "10020 23020 33020" "" 128m

addNewApp vp-traffic-monetisation-adapter-consumer "10040 23040 33040" "" 128m

popd

#执行脚本创建服务器目录

./deploy.sh eth0 dev "" "" "" 172.16.45.3:5000

（9190为tomcat端口号,21130为dubbo端口号,31130为spring-boot ssh 端口号）

②本地项目打包

allmake mvn release outdir=$HOME/deployment/outdir

③整体文件夹远程复制 scp -r deployment [vpclub@xxx](mailto:vpclub@xxx):

④进入对应服务器

ssh [vpclub@xxx](mailto:vpclub@xxx)

⑤执行启动操作（若是已启动则执行重启操作）

cd deployment

allmake docker restart or allmake docker up daemon

单独项目使用（进入指定项目目录下执行）sudo docker-compose up -d 即可

# ４.mysql,redis,mongodb,zookeeper 设置开机启动:

　#创建或者移动 数据存储路径docker-storage 到/目录

cd /docker-storage

#创建存储data目录

sudo ./init-dataDir.sh

#使用上面docker-storage文件夹（需要gitlab权限找冰寒）

**###Starting Docker Storage as a Service**

sudo chown -R root: /docker-storage

sudo nano /etc/init/docker-storage.conf

**###copy words below into /etc/init/docker-storage.conf**

description "Docker init"

start on runlevel [2345]

stop on runlevel [016]

respawn

respawn limit 10 5

chdir /docker-storage

exec /usr/local/bin/docker-compose up

**#**

# docker-storage service start

sudo service docker-storage start

#look at the current docker process

sudo docker ps

# 5.配置mysql备份

#mysql 自动备份

sudo mkdir -p /data/mysql

#timer shell -定时脚本

sudo vim /usr/sbin/backmysql

#add command添加如下命令

mysqldump -h"$(ifconfig eth0 | grep "inet addr" | cut -d ':' -f 2 | cut -d ' ' -f 1)" -uroot -p"@vpclubdev" --all-databases > /data/mysql/vpclub-"$(ifconfig eth0 | grep "inet addr" | cut -d ':' -f 2 | cut -d ' ' -f 1)"-"$(date +%Y-%m-%d\_%H%M%S)".sql

#grant privileges-授权

chmod +x /usr/sbin/backmysql

#系统自带定时任务列表添加自定义任务

sudo vim /etc/crontab

#add line

40 1 \* \* \* root sudo /usr/sbin/backmysql

＃恢复sql

＃备份并恢复sql

(导出)mysqldump -h172.16.45.3 -uroot -p -P3301 --all-databases > vpclub-back.sql

(导入)mysqldump -h120.76.240.113 -uroot -P3301 -p --all-databases < vpclub-back.sql

# 6.启动项目

进入相关项目执行sudo docker-compose up -d 即可