

docker和k8s的相关两门网课, docker和k8s的相关指令

github: https://github.com/microcosmx/train_ticket

云服务登陆方式:

xshell

云服务器下载上传文件:

filzezilla

基础集群部署:

k8s部署

新的master节点步骤(一般都是加新的worker节点, 不完全和master相同):

给生成的服务镜像打tag

配置私有仓库:

docker 启动失败的解决方法

验证私有仓库:

推送镜像到远程的私有仓库:

Deploy 到k8s集群里面:

新的worker节点:

配置私有仓库:

docker 启动失败的解决方法

验证私有仓库:

其他:批量删除docker镜像:

其他:无法push到本地仓库:

服务只需要在master节点上跑, 不用在所有节点上运行

其他:k8s批量删除服务:

其他:kubectl apply 和 kubectl create的区别

其他:centos的docker卸载和报错解决

资源限制

mvn clean package 编译的错误, 应该是编译的问题

WARNING: IPv4 forwarding is disabled. Networking will not work

修改maven的默认仓库

master节点部署项目顺序

报错:consider setting COMPOSE_HTTP_TIMEOUT to a higher value (current value: 60).

k8s常用命令要掌握

目前问题:

接下来的工作:

知识

任务分配

项目进度

docker和k8s的相关两门网课， docker和k8s的相关指令

github: https://github.com/microcosmx/train_ticket

云服务登陆方式:

xshell

云服务器下载上传文件:

filzezilla

基础集群部署:

k8s部署

由于节点都在国内，所以直接用腾讯云提供的服务

参考文件:

train_ticket/Document/k8s/k8s install/k8s安装和部署流程.txt

train_ticket/Document/k8s/ReadMe.md

新的master节点步骤(一般都是加新的worker节点，不完全和master相同):

- yum intsall ntp
- ntpdate ntp.sjtu.edu.cn

-----时间同步-----

- kubectl apply -f kube-flannel.yml

-----配置网络-----

- yum install git -y
- git clone https://github.com/microcosmx/train_ticket.git
- yum -y install java-1.8.0-openjdk-devel
- wget <http://apache.fayea.com/maven/maven-3/3.3.9/binaries/apache-maven-3.3.9-bin.tar.gz>
- tar -zxvf apache-maven-3.3.9-bin.tar.gz
- mv ./apache-maven-3.3.9 /usr/local/webserver/
- vi /etc/profile
- 在尾部追加行:
export MAVEN_HOME=/usr/local/webserver/apache-maven-3.3.9
export PATH=**MAVEN_HOME/bin** :{PATH}

- source /etc/profile

-----mvn 安装完成-----

- sudo yum install epel-release -y
 - sudo yum install -y python-pip -y
 - sudo pip install docker-compose
- docker-compose安装完成--

- docker-compose build

-----构建完镜像-----

- 卸载重装docker

为了保持和官方docker版本一致，需要重装docker

docker version 查看 若不为17.03需要重装

位置:

train_ticket/Document/k8s/k8s install/docker17.03/

yum install docker-ce-selinux-17.03.2.ce-1.el7.centos.noarch.rpm

yum install docker-ce-17.03.2.ce-1.el7.centos.x86_64.rpm

systemctl start docker.service

systemctl restart docker

-----若docker启动报错-----

- docker 启动失败的解决方法

https://blog.csdn.net/qq_35904833/article/details/74189383

<https://serverfault.com/questions/780844/cant-start-docker-daemon-invalid-image>

- journalctl -xe 看运行报错
-----删除两个文件，留备份-----
- mv /var/run/docker.pid /var/run/docker_old.pid
- mv /var/lib/docker/ /var/lib/docker_old/
- systemctl status docker 查看状态，active即为成功

-----docker重装完成-----

- vi /etc/hosts
所有节点都要配本地文件，否则会其他节点拉不到镜像，服务会均匀分布到其他节点上
- regserv的写过去
/etc/hosts:

```
1 172.17.0.9 regserv
2 172.17.0.3 manager-01
3 172.17.0.15 manager-02
4 172.17.0.14 manager-03
5 172.17.0.12 worker-01
6 172.17.0.13 worker-02
7 172.17.0.2 worker-03
8 172.17.0.16 worker-04
9 172.17.0.10 worker-05
10 172.17.0.6 worker-06
11 172.17.0.5 worker-07
12 172.17.0.4 worker-08
13 172.17.0.17 worker-09
14 172.17.0.7 worker-10
```

给生成的服务镜像打tag

- 把Docker Tag&Upload.md中的地址换成我们要的

```
1 %s/10.141.211.160:5000/regserv:5000/g
```

- 以下为已经改写好的

```
1  docker tag ts/ts-admin-basic-info-service regserv:5000/master/cluster-  
   ts-admin-basic-info-service  
2  docker tag ts/ts-admin-order-service regserv:5000/master/cluster-ts-  
   admin-order-service  
3  docker tag ts/ts-admin-route-service regserv:5000/master/cluster-ts-  
   admin-route-service  
4  docker tag ts/ts-admin-travel-service regserv:5000/master/cluster-ts-  
   admin-travel-service  
5  docker tag ts/ts-admin-user-service regserv:5000/master/cluster-ts-  
   admin-user-service  
6  docker tag ts/ts-assurance-service regserv:5000/master/cluster-ts-  
   assurance-service  
7  docker tag ts/ts-basic-service regserv:5000/master/cluster-ts-basic-  
   service  
8  docker tag ts/ts-cancel-service regserv:5000/master/cluster-ts-cancel-  
   service  
9  docker tag ts/ts-config-service regserv:5000/master/cluster-ts-config-  
   service  
10 docker tag ts/ts-consign-price-service regserv:5000/master/cluster-ts-  
   consign-price-service  
11 docker tag ts/ts-consign-service regserv:5000/master/cluster-ts-consign-  
   service  
12 docker tag ts/ts-contacts-service regserv:5000/master/cluster-ts-  
   contacts-service  
13 docker tag ts/ts-execute-service regserv:5000/master/cluster-ts-execute-  
   service  
14 docker tag ts/ts-food-map-service regserv:5000/master/cluster-ts-food-  
   map-service  
15 docker tag ts/ts-food-service regserv:5000/master/cluster-ts-food-  
   service  
16 docker tag ts/ts-inside-payment-service regserv:5000/master/cluster-ts-  
   inside-payment-service  
17 docker tag ts/ts-login-service regserv:5000/master/cluster-ts-login-  
   service  
18 docker tag ts/ts-news-service regserv:5000/master/cluster-ts-news-  
   service  
19 docker tag ts/ts-notification-service regserv:5000/master/cluster-ts-  
   notification-service  
20 docker tag ts/ts-order-other-service regserv:5000/master/cluster-ts-  
   order-other-service  
21 docker tag ts/ts-order-service regserv:5000/master/cluster-ts-order-  
   service  
22 docker tag ts/ts-payment-service regserv:5000/master/cluster-ts-payment-  
   service
```

```
23 docker tag ts/ts-preserve-other-service regserv:5000/master/cluster-ts-
    preserve-other-service
24 docker tag ts/ts-preserve-service regserv:5000/master/cluster-ts-
    preserve-service
25 docker tag ts/ts-price-service regserv:5000/master/cluster-ts-price-
    service
26 docker tag ts/ts-rebook-service regserv:5000/master/cluster-ts-rebook-
    service
27 docker tag ts/ts-register-service regserv:5000/master/cluster-ts-
    register-service
28 docker tag ts/ts-route-plan-service regserv:5000/master/cluster-ts-
    route-plan-service
29 docker tag ts/ts-route-service regserv:5000/master/cluster-ts-route-
    service
30 docker tag ts/ts-seat-service regserv:5000/master/cluster-ts-seat-
    service
31 docker tag ts/ts-security-service regserv:5000/master/cluster-ts-
    security-service
32 docker tag ts/ts-sso-service regserv:5000/master/cluster-ts-sso-service
33 docker tag ts/ts-station-service regserv:5000/master/cluster-ts-station-
    service
34 docker tag ts/ts-ticket-office-service regserv:5000/master/cluster-ts-
    ticket-office-service
35 docker tag ts/ts-ticketinfo-service regserv:5000/master/cluster-ts-
    ticketinfo-service
36 docker tag ts/ts-train-service regserv:5000/master/cluster-ts-train-
    service
37 docker tag ts/ts-travel2-service regserv:5000/master/cluster-ts-travel2-
    service
38 docker tag ts/ts-travel-service regserv:5000/master/cluster-ts-travel-
    service
39 docker tag ts/ts-travel-plan-service regserv:5000/master/cluster-ts-
    travel-plan-service
40 docker tag ts/ts-ui-dashboard regserv:5000/master/cluster-ts-ui-
    dashboard
41 docker tag ts/ts-verification-code-service regserv:5000/master/cluster-
    ts-verification-code-service
42 docker tag ts/ts-voucher-service regserv:5000/master/cluster-ts-voucher-
    service
43 docker tag mongo regserv:5000/master/cluster-ts-mongo
44 docker tag mysql regserv:5000/master/cluster-ts-mysql
45 docker tag rabbitmq:management regserv:5000/master/cluster-ts-rabbitmq-
    management
46 docker tag redis regserv:5000/master/cluster-ts-redis
47 docker tag openzipkin/zipkin regserv:5000/master/cluster-ts-openzipkin-
    zipkin
```

有五个基础镜像

```
1 mongo:latest
2 mysql:latest
3 rabbitmq:management
4 openzipkin/zipkin:latest
5 redis:latest
```

这五个基础镜像需要自己拉取

```
1 docker pull mongo:latest
2 docker pull mysql:latest
3 docker pull rabbitmq:management
4 docker pull openzipkin/zipkin:latest
5 docker pull redis:latest
```

配置私有仓库:

若未配置私有仓库，则需配置私有仓库，若已经配置完成，则跳过私有仓库的设置:

<https://blog.csdn.net/u010397369/article/details/42422243>

vi /etc/docker/daemon.json

```
{
  "insecure-registries": ["regserv:5000"]
}
```

若要加速，为下列:

```
{
  "registry-mirrors": [
    "https://registry.docker-cn.com"
  ],
  "insecure-registries": ["regserv:5000"]
}
```

保存退出，再次重启docker

systemctl restart docker

否则会有:

上传镜像报错：server gave HTTP response to HTTPS client

docker 启动失败的解决方法

https://blog.csdn.net/qq_35904833/article/details/74189383

<https://serverfault.com/questions/780844/cant-start-docker-daemon-invalid-image>

- yum install docker
- journalctl -xe 看运行报错
-----删除两个文件，留备份-----
- mv /var/run/docker.pid /var/run/docker_old.pid
- mv /var/lib/docker/ /var/lib/docker_old/
删除了/var/lib/docker的话相当于所有镜像都删除了
- systemctl status docker 查看状态，active即为成功

验证私有仓库：

每个节点上都要配置好私有仓库：

测试：

```
1 docker pull regserv:5000/master/cluster-ts-travel2-service
```

推送镜像到远程的私有仓库：

将打好tag的镜像推到私有仓库：(本地仓库拉取速度远快于远程仓库)

同样的，需要修改

以下是修改完的版本：

```
1 docker push regserv:5000/master/cluster-ts-admin-basic-info-service
2 docker push regserv:5000/master/cluster-ts-admin-order-service
3 docker push regserv:5000/master/cluster-ts-admin-route-service
4 docker push regserv:5000/master/cluster-ts-admin-travel-service
5 docker push regserv:5000/master/cluster-ts-admin-user-service
6 docker push regserv:5000/master/cluster-ts-assurance-service
7 docker push regserv:5000/master/cluster-ts-basic-service
8 docker push regserv:5000/master/cluster-ts-cancel-service
9 docker push regserv:5000/master/cluster-ts-config-service
10 docker push regserv:5000/master/cluster-ts-consign-price-service
11 docker push regserv:5000/master/cluster-ts-consign-service
12 docker push regserv:5000/master/cluster-ts-contacts-service
```



```
13 docker push regserv:5000/master/cluster-ts-execute-service
14 docker push regserv:5000/master/cluster-ts-food-map-service
15 docker push regserv:5000/master/cluster-ts-food-service
16 docker push regserv:5000/master/cluster-ts-inside-payment-service
17 docker push regserv:5000/master/cluster-ts-login-service
18 docker push regserv:5000/master/cluster-ts-news-service
19 docker push regserv:5000/master/cluster-ts-notification-service
20 docker push regserv:5000/master/cluster-ts-order-other-service
21 docker push regserv:5000/master/cluster-ts-order-service
22 docker push regserv:5000/master/cluster-ts-payment-service
23 docker push regserv:5000/master/cluster-ts-preserve-other-service
24 docker push regserv:5000/master/cluster-ts-preserve-service
25 docker push regserv:5000/master/cluster-ts-price-service
26 docker push regserv:5000/master/cluster-ts-rebook-service
27 docker push regserv:5000/master/cluster-ts-register-service
28 docker push regserv:5000/master/cluster-ts-route-plan-service
29 docker push regserv:5000/master/cluster-ts-route-service
30 docker push regserv:5000/master/cluster-ts-seat-service
31 docker push regserv:5000/master/cluster-ts-security-service
32 docker push regserv:5000/master/cluster-ts-sso-service
33 docker push regserv:5000/master/cluster-ts-station-service
34 docker push regserv:5000/master/cluster-ts-ticket-office-service
35 docker push regserv:5000/master/cluster-ts-ticketinfo-service
36 docker push regserv:5000/master/cluster-ts-train-service
37 docker push regserv:5000/master/cluster-ts-travel2-service
38 docker push regserv:5000/master/cluster-ts-travel-service
39 docker push regserv:5000/master/cluster-ts-travel-plan-service
40 docker push regserv:5000/master/cluster-ts-ui-dashboard
41 docker push regserv:5000/master/cluster-ts-verification-code-service
42 docker push regserv:5000/master/cluster-ts-voucher-service
43 docker push regserv:5000/master/cluster-ts-mongo
44 docker push regserv:5000/master/cluster-ts-mysql
45 docker push regserv:5000/master/cluster-ts-rabbitmq-management
46 docker push regserv:5000/master/cluster-ts-redis
47 docker push regserv:5000/master/cluster-ts-openziplin-zipkin
```

Deploy 到k8s集群里面:

先修改deployment文件:

把原ip:5000都修改为你的regserv:5000

然后要跑和文档同目录的不能跑在上级目录deploy文件混淆

train_ticket/Document/k8s/

```
kubectl apply -f ts-deployment-part1.yml
```

```
kubectl apply -f ts-deployment-part2.yml
```

```
kubectl apply -f ts-deployment-part3.yml
```

新的worker节点:

- yum intsall ntp
- ntpdate ntp.sjtu.edu.cn
-----时间同步-----
- kubectl apply -f kube-flannel.yml
-----配置网络-----
- yum install git -y
- git clone https://github.com/microcosmx/train_ticket.git

配置私有仓库:

若未配置私有仓库，则需配置私有仓库，若已经配置完成，则跳过私有仓库的设置:

<https://blog.csdn.net/u010397369/article/details/42422243>

```
vi /etc/docker/daemon.json
```

```
{  
  "insecure-registries": ["regserv:5000"]  
}
```

若要加速，为下列:

```
{  
  "registry-mirrors": [  
    "https://registry.docker-cn.com"  
  ],  
  "insecure-registries": ["regserv:5000"]  
}
```

```
}
```

保存退出，再次重启docker

systemctl restart docker

否则会有：

上传镜像报错：server gave HTTP response to HTTPS client

docker 启动失败的解决方法

https://blog.csdn.net/qq_35904833/article/details/74189383

<https://serverfault.com/questions/780844/cant-start-docker-daemon-invalid-image>

- yum install docker
- journalctl -xe 看运行报错
-----删除两个文件，留备份-----
- mv /var/run/docker.pid /var/run/docker_old.pid
- mv /var/lib/docker/ /var/lib/docker_old/
删除了/var/lib/docker的话相当于所有镜像都删除了
- systemctl status docker 查看状态，active即为成功

验证私有仓库：

每个节点上都要配置好私有仓库：

测试：

```
1 docker pull regserv:5000/master/cluster-ts-travel2-service
```

• 其他:批量删除docker镜像：

```
docker rmi $(docker images | grep "ts-" | awk '{print $3}')
```

```
docker rm $(docker ps -a | grep Exited | awk '{print $1}') //删除异常  
停止的docker容器
```

• 其他:无法push到本地仓库：

解决之后呢，往regserv推，可能会遇到，regserv已经满了的情况，那就删空间

• 服务只需要在master节点上跑，不用在所有节点上运行

• 其他:k8s批量删除服务:

```
1 kubectl get pods | grep Evicted | awk '{print $1}' | xargs kubectl
  delete pod
2 第一个是获取pods信息,第二个是过滤与Evicted有关的,第三个是第一列,第四个是作为
  kubectl参数执行
3 所以:
4 kubectl get pods| awk '{print $1}'| xargs kubectl delete pods
5 删除所有pods,无视状态
6 kubectl get deploy| awk '{print $1}'| xargs kubectl delete deploy
7 删除所有deploy,无视状态
8 kubectl get svc| awk '{print $1}'| xargs kubectl delete svc
9 删除所有service,无视状态
```

- 卸载重装docker

为了保持和官方docker版本一致,需要重装docker

docker version 查看 若不为17.03需要重装

位置:

train_ticket/Document/k8s/k8s install/docker17.03/

yum install docker-ce-selinux-17.03.2.ce-1.el7.centos.noarch.rpm

yum install docker-ce-17.03.2.ce-1.el7.centos.x86_64.rpm

systemctl start docker.service

systemctl restart docker

-----若docker启动报错-----

- docker 启动失败的解决方法

https://blog.csdn.net/qq_35904833/article/details/74189383

<https://serverfault.com/questions/780844/cant-start-docker-daemon-invalid-image>

- journalctl -xe 看运行报错

-----删除两个文件,留备份-----

- mv /var/run/docker.pid /var/run/docker_old.pid

- mv /var/lib/docker/ /var/lib/docker_old/

- systemctl status docker 查看状态, active即为成功

-----docker重装完成-----

- vi /etc/hosts

所有节点都要配本地文件,否则会其他节点拉不到镜像,服务会均匀分布到其他

节点上

- regserv的写过去
/etc/hosts:

```
1 172.17.0.9 regserv
2 172.17.0.3 manager-01
3 172.17.0.15 manager-02
4 172.17.0.14 manager-03
5 172.17.0.12 worker-01
6 172.17.0.13 worker-02
7 172.17.0.2 worker-03
8 172.17.0.16 worker-04
9 172.17.0.10 worker-05
10 172.17.0.6 worker-06
11 172.17.0.5 worker-07
12 172.17.0.4 worker-08
13 172.17.0.17 worker-09
14 172.17.0.7 worker-10
```

• 其他:kubectl apply 和 kubectl create的区别

不需要删除微服务之后再重新部署，修改文件后，重新输入kubectl apply 部署文件名.yml就可以更新

kubctl create需要删除后再重新create

• 其他:centos的docker卸载和报错解决

centos卸载docker: <https://www.jianshu.com/p/438f5fdc696b>

centos安装文档里的docker报错:

Error: docker-ce-selinux conflicts with 2:container-selinux-2.12-2.gite7096ce.el7.noarch

<https://github.com/docker/for-linux/issues/20>

sudo yum erase docker-engine-selinux

• 资源限制

可能有资源限制，worker-01一跑就卡，这个需要测试

- **mvn clean package 编译的错误，应该是编译的问题**

1.0那个报错，到那个目录去改名字，改成jar包就可以了

- **WARNING: IPv4 forwarding is disabled. Networking will not work**

`docker-compose -f docker-compose.yml build`

这一步遇到报错：

WARNING: IPv4 forwarding is disabled. Networking will not work

解决方案：

<https://blog.csdn.net/weiguang1017/article/details/76212203>

- **修改maven的默认仓库**

<https://blog.csdn.net/timo1160139211/article/details/54289938>

`mvn -n` 可以看到 `mvn config` 文件夹的位置

- **master节点部署项目顺序**

先mvn编译出jar包，然后再微服务化

mvn报错的时候仔细看报错信息，对应目录下应该有相应的jar包

`mvn clean -e -X`：Debug模式

- **报错:consider setting COMPOSE_HTTP_TIMEOUT to a higher value (current value: 60).**

<https://stackoverflow.com/questions/36488209/how-to-override-the-default-value-of-compose-http-timeout-with-docker-compose-co>

k8s常用命令要掌握

目前问题:

三个节点上，大部分服务已启动，仍有一些未启动，需要解决
查看

kubectl get pod

kubectl describe pod pod.name

解决相应的问题

接下来的工作:

服务监控，获取相应的指标

找指标对服务重新划分

重构服务

知识

- docker
- kubectl
- springboot
- springcloud

任务分配

项目进度