本设计并实现了一个基于 Rancher 部署的高可用及分布式学生信息管理系统。设计基于阿里云服务器，目的在于利用 Docker 容器技术搭建一个有实际应用价值的应用，本设计所部署的具体应用为高可用及分布式的学生信息管理系统。该应用部署难度中等，且很具有实际应用价值。项目是采用多台具体部署 Web 应用、数据库及负载均衡容器节点。因涉及到的主机数较多，单独使用命令进行 Docker 的工作量比较重，为方便各节点内 Docker 容器的部署，故采用了 Rancher 对每台主机进行管理，大大提高了容器部署的效率。

项目采用一台主机作为负载均衡及容器调度节点，三台主机作为 Web 应用容器节点，其中一台同时作为数据库节点。负载均衡节点可以向三台 Web 容器节点进行负载均衡访问调度。数据库节点为三台 Web 容器提供数据存储服务。

通过各种关键的部署，所有节点的功能运行正常。经测试后，负载均衡节点能在其中一个节点宕机或者负载量过大的情况下实现自动分流切换，且数据库能保持事务的完整性，达到项目预期的效果。

1.1.1 浅谈 Docker

Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的镜像中，然后发布到任何流行的 Linux 或 Windows 机器上，也可以实现虚拟化。容器是完全使用沙箱机制，相互之间不会有任何接口。

Docker 核心解决的问题是利用 LXC 来实现类似 VM 的功能，从而利用更加节省的硬件资源提供给用户更多的计算资源。同 VM 的方式不同, LXC 其并不是一套硬件虚拟化方法 - 无法归属到全虚拟化、部分虚拟化和半虚拟化中的任意一个，而是一个操作系统级虚拟化方法, 理解起来可能并不像 VM 那样直观。所以我们从虚拟化到 docker 要解决的问题出发，看看他是怎么满足用户虚拟化需求的。

1.1.2 浅谈 Rancher

Rancher 是一个开源的企业级容器管理平台。通过 Rancher，企业再也不必自己使用一系列的开源软件去从头搭建容器服务平台。Rancher 提供了在生产环境中使用的管理 Docker 和 Kubernetes 的全栈化容器部署与管理平台。

Rancher 由以下四个部分组成：

1. 基础设施编排

Rancher 可以使用任何公有云或者私有云的 Linux 主机资源。Linux 主机可以是虚拟机，也可以是物理机。Rancher 仅需要主机有 CPU，内存，本地磁盘和网络资源。从 Rancher 的角度来说，一台云厂商提供的云主机和一台自己的物理机是一样的。

Rancher 为运行容器化的应用实现了一层灵活的基础设施服务。Rancher 的基础设施服务包括网络， 存储， 负载均衡， DNS 和安全模块。Rancher 的基础设施服务也是通过容器部署的，所以同样 Rancher 的基础设施服务可以运行在任何 Linux 主机上。

2. 容器编排与调度

很多用户都会选择使用容器编排调度框架来运行容器化应用。Rancher 包含了当前全部主流的编排调度引擎，例如 Docker Swarm， Kubernetes， 和 Mesos。同一个用户可以创建 Swarm 或者 Kubernetes 集群。并且可以使用原生的 Swarm 或者 Kubernetes 工具管理应用。

除了 Swarm，Kubernetes 和 Mesos 之外，Rancher 还支持自己的 Cattle 容器编排调度引擎。Cattle 被广泛用于编排 Rancher 自己的基础设施服务以及用于 Swarm 集群，Kubernetes 集群和 Mesos 集群的配置，管理与升级。

3. 应用商店

Rancher 的用户可以在应用商店里一键部署由多个容器组成的应用。用户可以管理这个部署的应用，并且可以在这个应用有新的可用版本时进行自动化的升级。Rancher 提供了一个由 Rancher 社区维护的应用商店，其中包括了一系列的流行应用。Rancher 的用户也可以创建自己的私有应用商店。

4. 企业级权限管理

Rancher 支持灵活的插件式的用户认证。支持 Active Directory，LDAP，Github 等认证方式。Rancher 支持在环境级别的基于角色的访问控制 (RBAC)，可以通过角色来配置某个用户或者用户组对开发环境或者生产环境的访问权限。

1.1.3 浅谈分布式应用

分布式应用（distributed application）指的是应用程序分布在不同计算机上，通过网络来共同完成一项任务的工作方式。

分布式应用程序是指：应用程序分布在不同计算机上，通过网络来共同完成一项任务。通常为服务器/客户端模式。研究一下当前的应用程序开发，你会发现一个绝对的倾向：人们开始偏爱基于浏览器的客户应用程序。这当然不是因为客户能够提供更好的用户界面，而是因为它能够避免花在桌面应用程序发布上的高成本。发布桌面应用程序成本很高，一半是因为应用程序安装和配置的问题，另一半是因为客户和服务器之间通信的问题。

许多商用程序还面临另一个问题，那就是与其他程序的互操作性。如果所有的应用程序都是使用 COM 或.NET 语言写的，并且都运行在 Windows 平台上，那就天下太平了。然而，事实上大多数商业数据仍然在大型主机上以非关系文件(VSAM)的形式存放，并由 COBOL 语言编写的大型机程序访问。而且，还有很多商用程序继续在使用 C++、Java、Visual Basic 和其他各种各样的语言编写。除了最简单的程序之外，所有的应用程序都需要与运行在其他异构平台上的应用程序集成并进行数据交换。这样的任务通常都是由特殊的方法，如文件传输和分析，消息队列，还有仅适用于某些情况的的 API，如 IBM 的"高级程序到程序交流(APPC)"等来完成的。在以前，没有一个应用程序通信标准，是独立于平台、组建模型和编程语言的。只有通过 Web Service，客户端和服务器才能够自由的用 HTTP 进行通信，不论两个程序的平台和编程语言是什么。

2 总体设计

2.1 设计总流程图

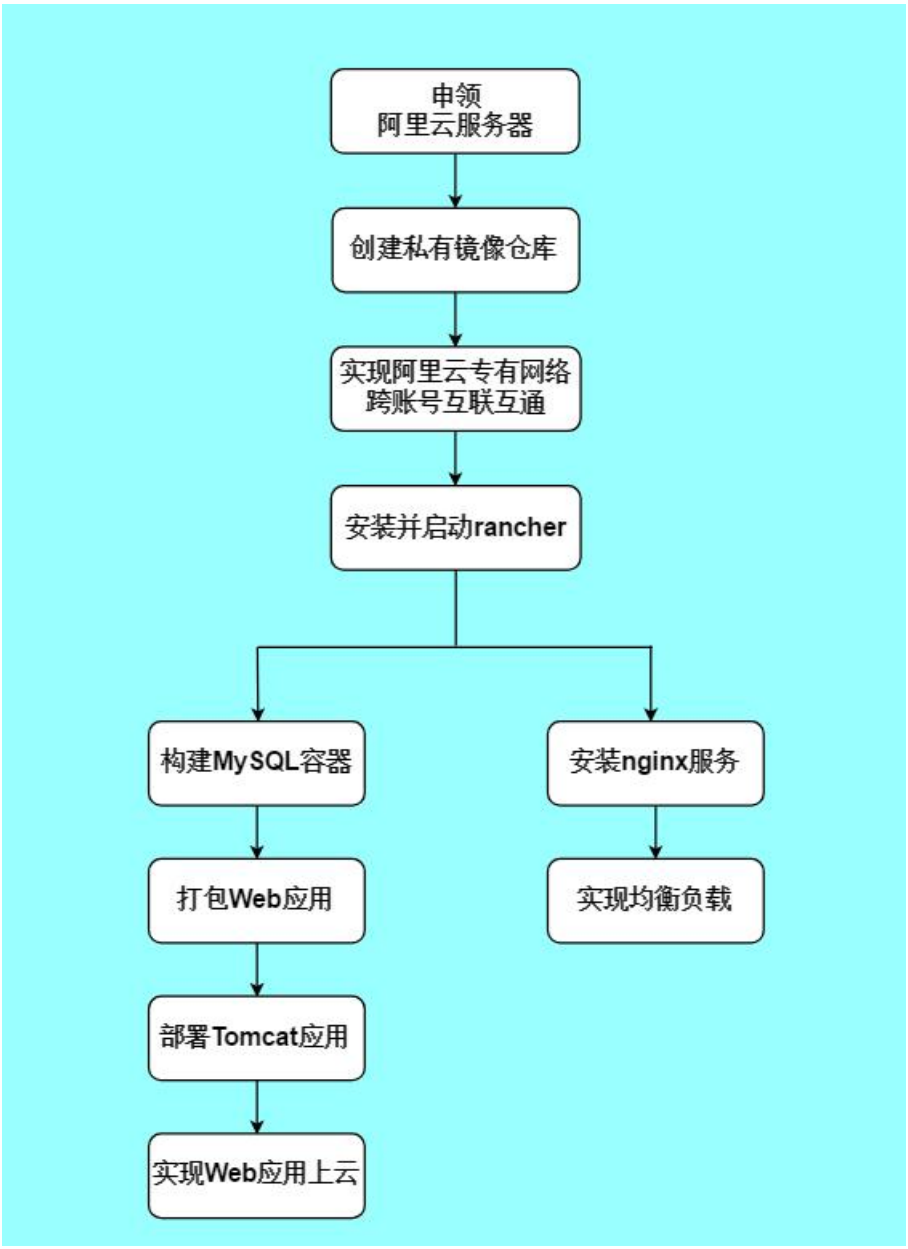

图 2.1 设计总流程图

2.1.1 创建镜像仓库，类型定为私有

图 3.1 创建镜像仓库 1

2.1.2 设为本地仓库，方便从本地 Ubuntu 中拉取镜像

图 3.2 创建镜像仓库 2

图 3.3 镜像仓库

3. 将镜像推送到Registry

```
$ sudo docker login --usern          registry.cn-shenzhen.aliyuncs.com
$ sudo docker tag [ImageId] registry.cn-shenzhen.aliyuncs.com/[redacted]:[镜像版本号]
$ sudo docker push registry.cn-shenzhen.aliyuncs.com/[redacted]:[镜像版本号]
```

图 3.4 推送镜像到 Registry

. 从Registry中拉取镜像

```
$ sudo docker pull registry.cn-shenzhen.aliyuncs.com/[redacted]:[镜像版本号]
```

图 3.5 从 Registry 中拉取镜像

2.1.3 镜像上传

首先就是要把镜像打包好，然后拉到 Ubuntu 上，再从 Ubuntu 中 copy 到 Tomcat 容器中，最后在通过 pull 拉到仓库

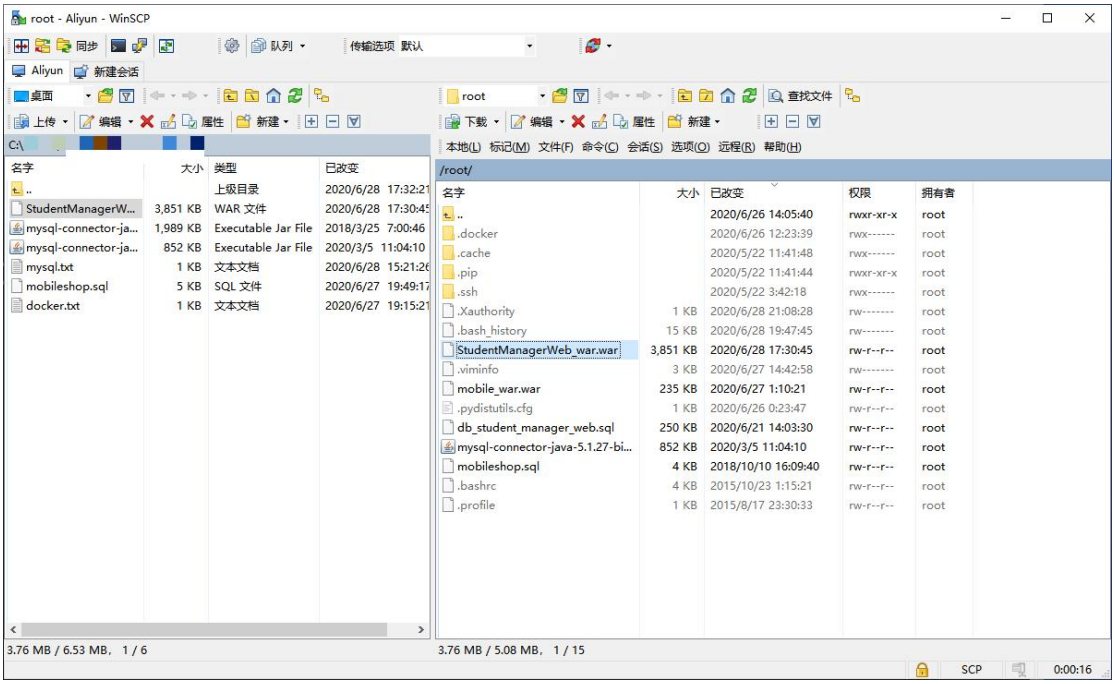

图 3.6 利用 WinSCP 与服务器进行文件传输 1

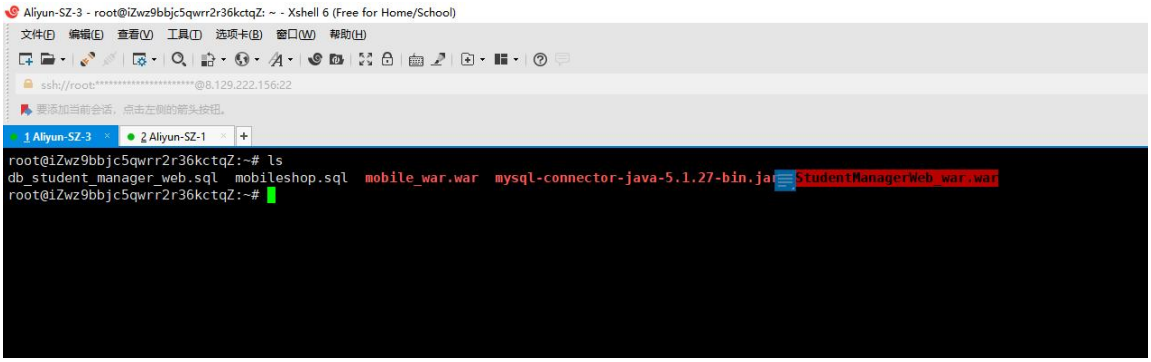

图 3.7 利用 WinSCP 与服务器进行文件传输 2

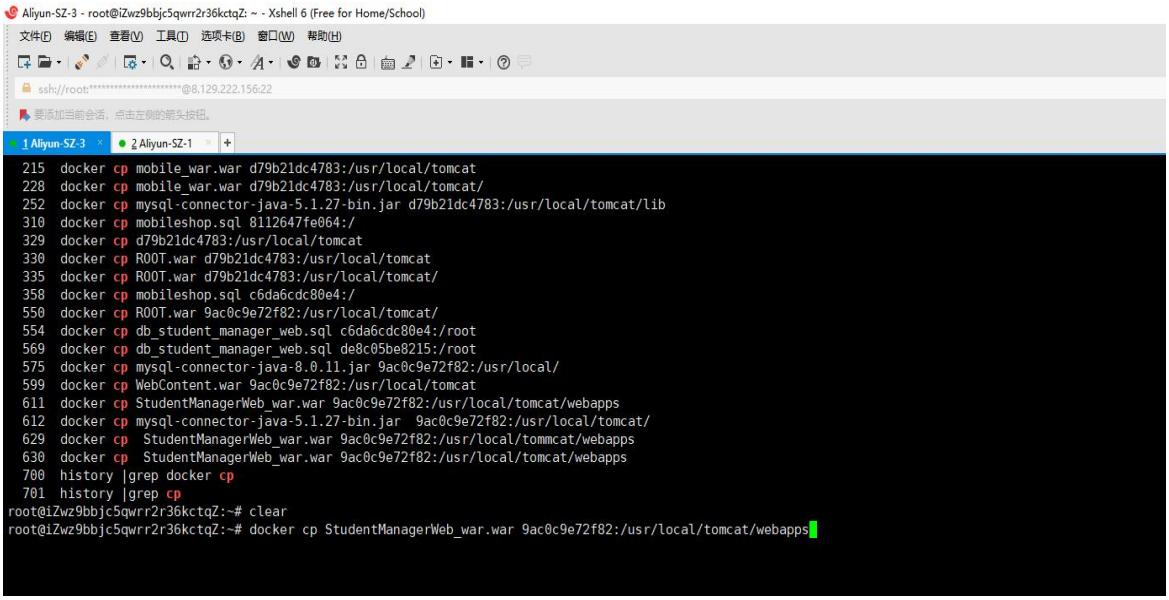

图 3.8 利用 WinSCP 与服务器进行文件传输 3

2.2 打包上传 JSP 项目

2.2.1 修改 JSP 项目

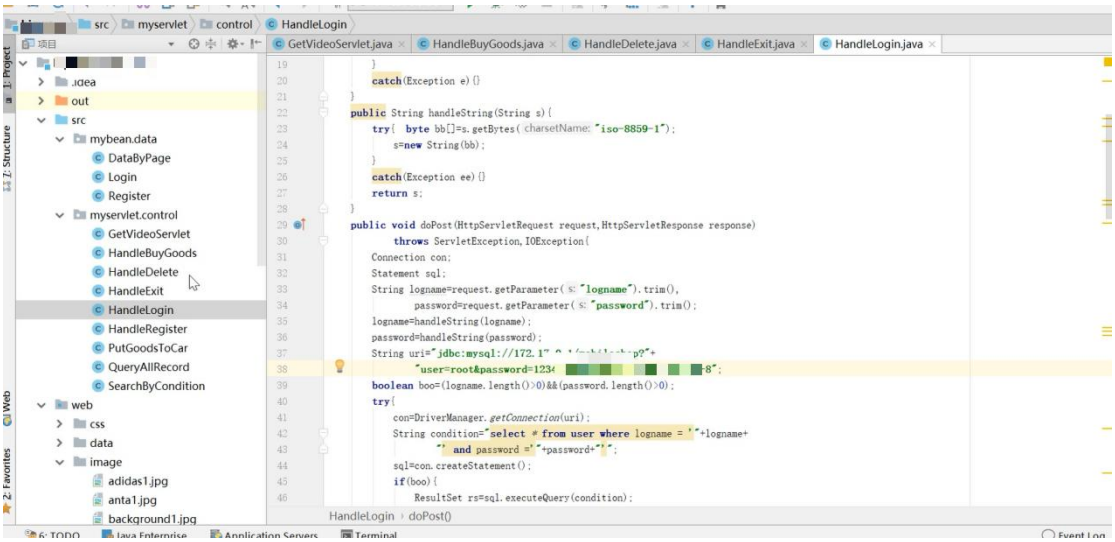

图 3.9 更改数据库配置

2.2.2 打包工程

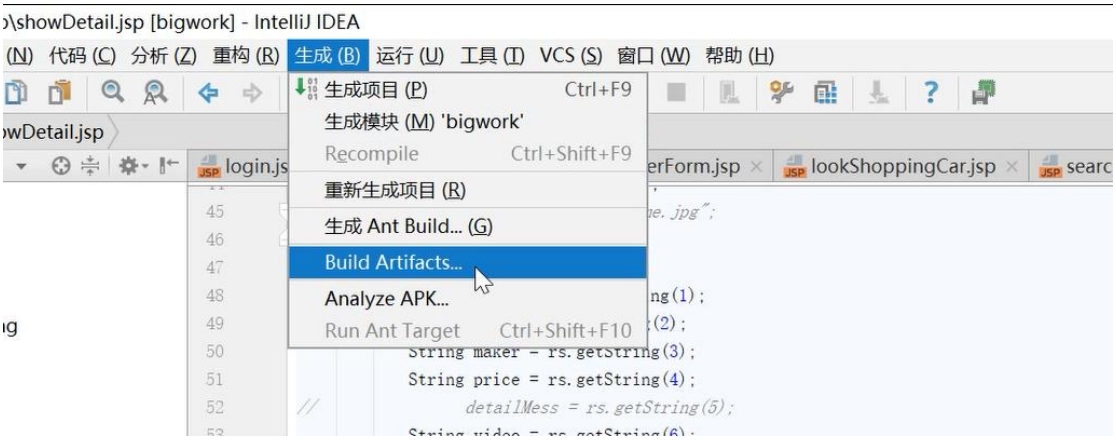

图 3.11 打包工程 1

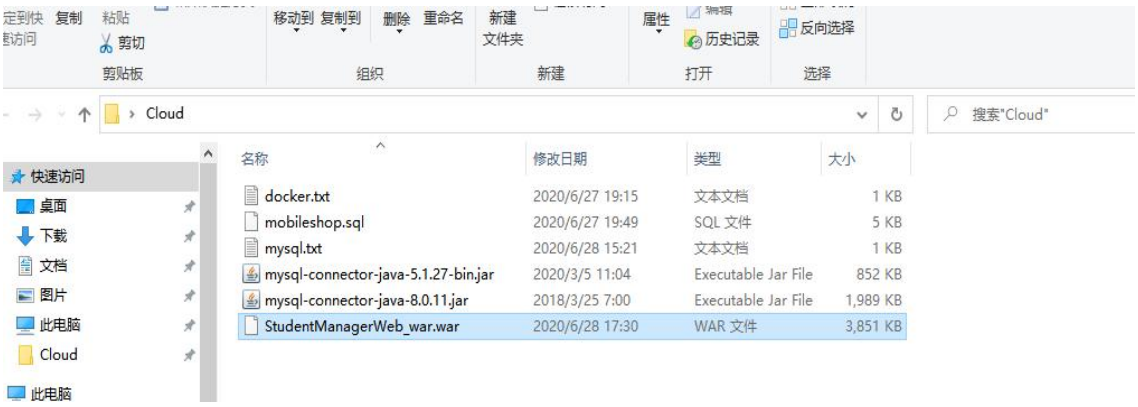

图 3.12 打包工程 2

2.2.3 转储 sql 文件

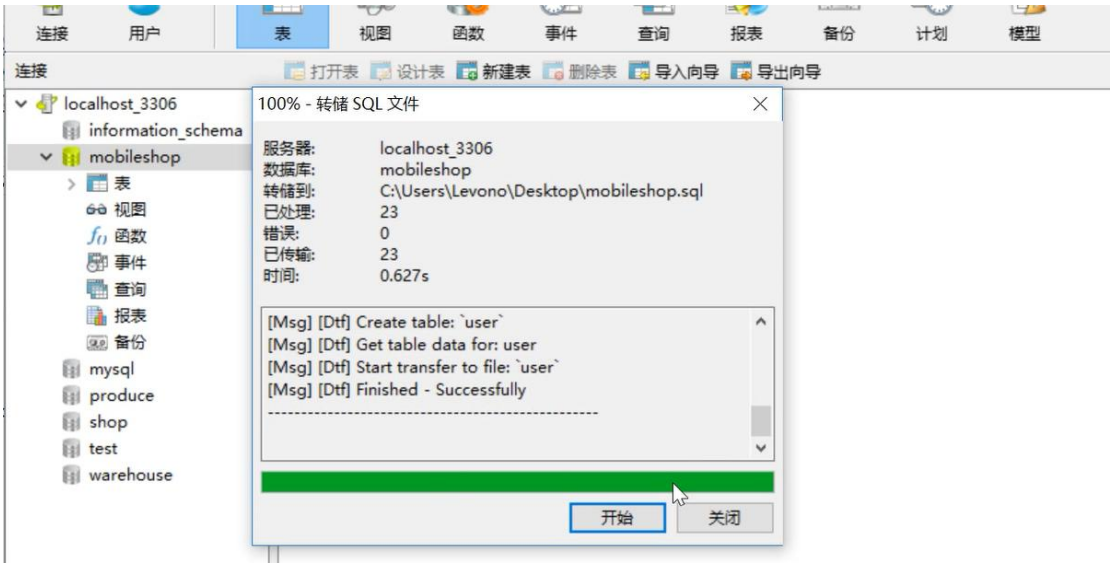

图 3.13 转储 SQL 文件

2.2.4 上传工程

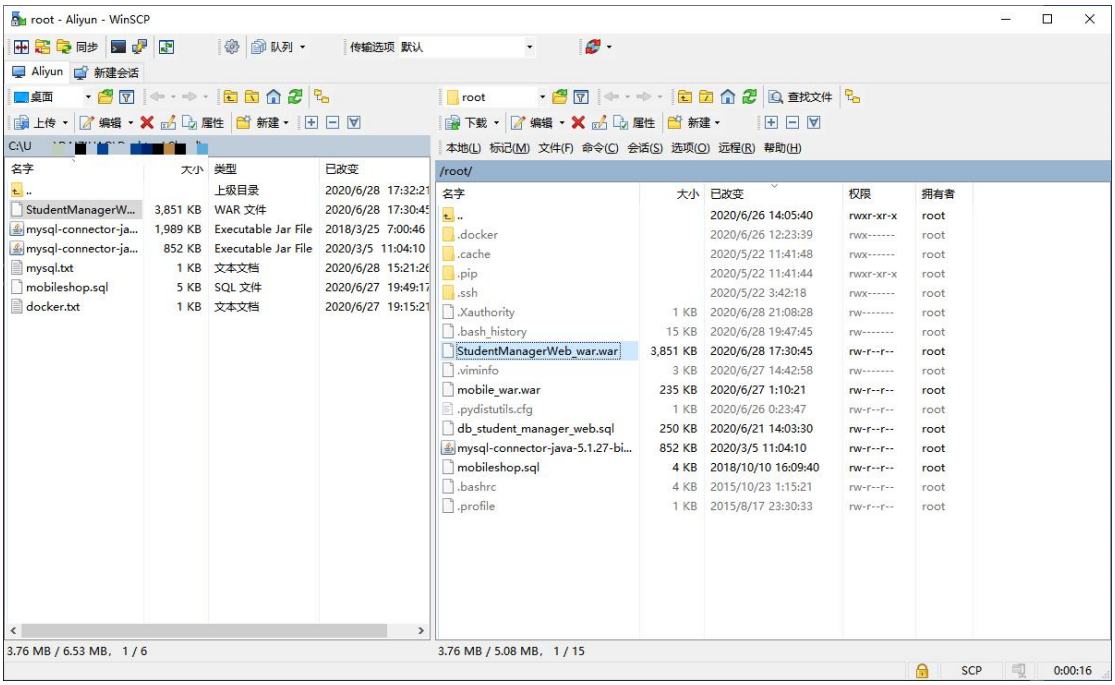

图 3.14 利用 WinSCP 发送文件

2.3 添加 Tomcat 服务

2.3.1 设计详情

1. 查询 docker 镜像库中的 tomcat

```
Welcome to Alibaba Cloud Elastic Compute Service !

Last login: Sun Jun 28 21:19:53 2020 from 112.96.194.116
root@iZff6nq8h7e9rkZ:~# docker search tomcat
NAME                              DESCRIPTION                                     STARS               OFFICIAL
tomcat                            Apache Tomcat is an open source implementati…   2765                [OK]
tomee                             Apache TomEE is an all-Apache Java EE certif…   79                  [OK]
dordoka/tomcat                    Ubuntu 14.04, Oracle JDK 8 and Tomcat 8 base…   54
bitnami/tomcat                    Bitnami Tomcat Docker Image                     35
kubeguide/tomcat-app              Tomcat image for Chapter 1                      28
consol/tomcat-7.0                 Tomcat 7.0.57, 8080, "admin/admin"              17
cloudesire/tomcat                 Tomcat server, 6/7/8                            15
aallam/tomcat-mysql               Debian, Oracle JDK, Tomcat & MySQL              13
arm32v7/tomcat                    Apache Tomcat is an open source implementati…   10
rightctrl/tomcat                  CentOS , Oracle Java, tomcat application ssl…   6
```

图 3.15 查询 tomcat 镜像

2. docker 拉取 tomcat 镜像，并用 docker images 命令查看是否拉取成功

```
root@iZff6nq8h7e9rkZ:~# docker pull tomcat
Using default tag: latest
latest: Pulling from library/tomcat
e9afc4f90ab0: Already exists
989e6b19a265: Already exists
af14b6c2f878: Already exists
5573c4b30949: Already exists
fb1a405f128d: Already exists
612a9f566fdc: Pull complete
cf63ebed1142: Downloading [========>                                          ]  31.78MB/196.2MB
fbb20561cd50: Download complete
e99c920870d7: Download complete
b7f793f2be47: Download complete
```

图 3.16 拉取 tomcat 镜像

```
root@iZff6nq8h7e9rkZ:~# docker images
REPOSITORY                    TAG                 IMAGE ID            CREATED             SIZE
tomcat                        8.5                 e010d327a904        2 weeks ago         529MB
mysql                         5.7                 9cfcce23593a        2 weeks ago         448MB
busybox                       latest              1c35c4412082        3 weeks ago         1.22MB
rancher/server                stable              98d8bb571885        7 weeks ago         1.08GB
rawmind/alpine-traefik        1.7.18-0            58b9635f96a3        9 months ago        185MB
rancher/scheduler             v0.8.6              fbedeaddc3e9        17 months ago       248MB
rawmind/alpine-traefik        1.7.4-0             4a7e515f5e05        20 months ago       154MB
rancher/agent                 v1.2.11             1cc7591af4f5        23 months ago       243MB
rancher/net                   v0.13.17            f170c38e3763        23 months ago       311MB
rancher/dns                   v0.17.4             678bde0de4d2        23 months ago       249MB
rancher/healthcheck           v0.3.8              ce78cf69cc0b        24 months ago       391MB
rancher/metadata              v0.10.4             02104eb6e270        24 months ago       251MB
rancher/network-manager       v0.7.22             13381626c510        24 months ago       256MB
rancher/hello-world           latest              cab3bc026f39        24 months ago       18.2MB
nginx                         latest              958a7ae9e569        3 years ago         109MB
rancher/net                   holder              665d9f6e8cc1        3 years ago         267MB
rancher/lb-service-haproxy    v0.6.4              9b8c61dc1db6        3 years ago         335MB
tutum/hello-world             latest              31e17b0746e4        4 years ago         17.8MB
root@iZff6nq8h7e9rkZ:~#
```

图 3.17 显示 tomcat 镜像

3. 对主机添加标签，以便后面进行调度使用

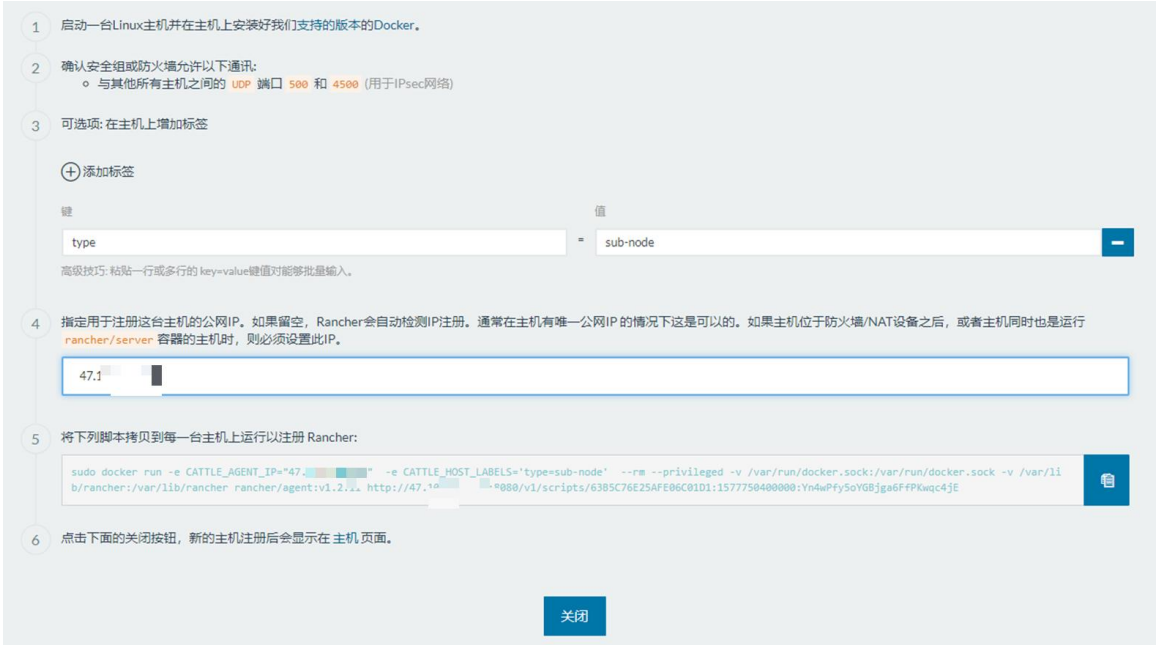

图 3.18 添加主机标签

4. 添加阿里云私有镜像库

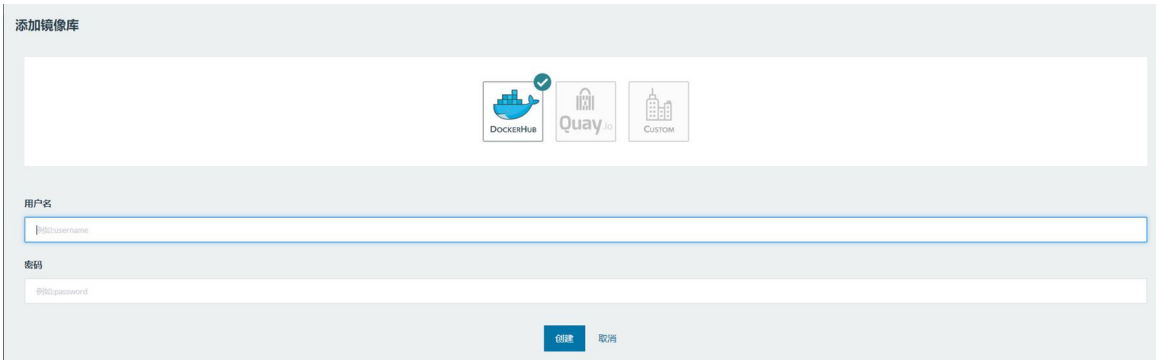

图 3.19 添加私有镜像库

5. 将需要部署的项目封装到 tomcat 容器中，再将其打包成镜像 push 到私有库当中

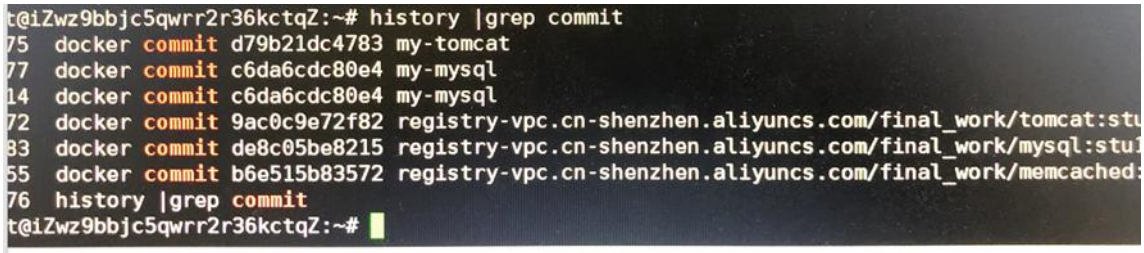

图 3.20 commit 封装镜像库

图 3.21 push tomcat 镜像

5. 对所有需要部署应用的主机添加特定标签，在 Rancher 对带有对应标签的主机进行调度后进行一键部署。

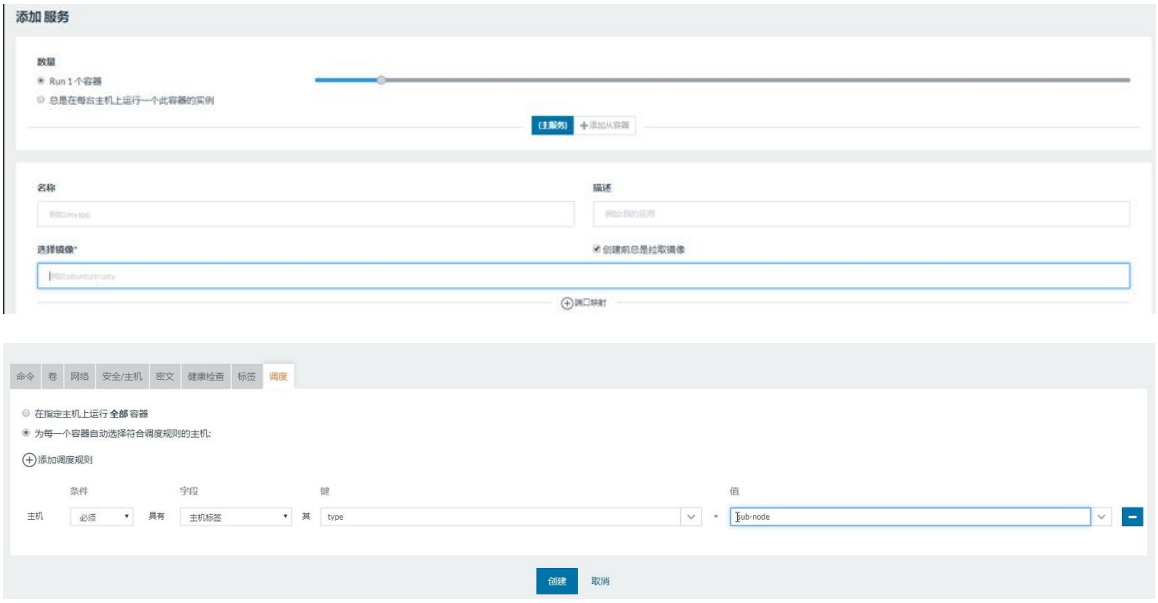

图 3.22 调度主机

图 3.23 部署 tomcat

2.3.2 部署结果

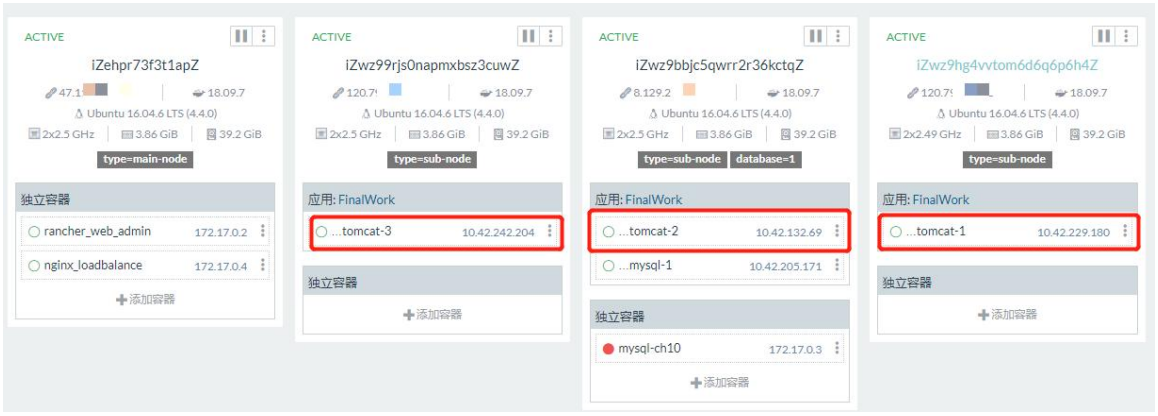

图 3.24 tomcat 部署结果

2.4 添加 MySQL 服务

2.4.1 设计详情

Rancher 应用商店库中提供有 MySQL，可以直接查看进行配置安装。本次项目选择 MySQL5.7 版本进行部署。

1. 查询 Docker 中的 MySQL。

```
Welcome to Alibaba Cloud Elastic Compute Service !

Last login: Sun Jun 28 21:55:32 2020 from 112.96.194.116
root@iZff6nq8h7e9rkZ:~# docker search mysql
NAME                              DESCRIPTION                                     STARS               OFFICIAL            AUTOMATED
mysql                             MySQL is a widely used, open-source relation…   9687                [OK]
mariadb                           MariaDB is a community-developed fork of MyS…   3524                [OK]
mysql/mysql-server                Optimized MySQL Server Docker images. Create…   706                                     [OK]
centos/mysql-57-centos7           MySQL 5.7 SQL database server                   77
mysql/mysql-cluster               Experimental MySQL Cluster Docker images. Cr…   70
centurylink/mysql                 Image containing mysql. Optimized to be link…   61                                      [OK]
bitnami/mysql                     Bitnami MySQL Docker Image                      43                                      [OK]
deitch/mysql-backup               REPLACED! Please use http://hub.docker.com/r…   41                                      [OK]
tutum/mysql                       Base docker image to run a MySQL database se…   35
schickling/mysql-backup-s3        Backup MySQL to S3 (supports periodic backup…   30                                      [OK]
prom/mysqld-exporter                                                              28                                      [OK]
```

图 3.25 搜索 MySQL

2. Docker 拉取 MySQL 镜像，并用 Docker images 命令查看是否拉取成功。

```
Run a command in a new container
root@iZff6nq8h7e9rkZ:~# docker run -d -p 3306:3306 mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
8559a31e96f4: Already exists
d51ce1c2e575: Already exists
c2344adc4858: Already exists
fcf3ceff18fc: Already exists
16da0c38dc5b: Already exists
b905d1797e97: Already exists
4b50d1c6b05c: Already exists
c75914a65ca2: Pull complete
1ae8042bdd09: Pulling fs layer
453ac13c00a3: Pulling fs layer
9e680cd72f08: Waiting
a6b5dc864b6c: Waiting
```

图 3.26 安装并运行 MySQL

```
root@iZff6nq8h7e9rkZ:~# docker images
REPOSITORY                  TAG                 IMAGE ID            CREATED             SIZE
tomcat                      8.5                 e010d327a904        2 weeks ago         529MB
mysql                       5.7                 9cfcce23593a        2 weeks ago         448MB
busybox                     latest              1c35c4412082        3 weeks ago         1.22MB
rancher/server              stable              98d8bb571885        7 weeks ago         1.08GB
rawmind/alpine-traefik      1.7.18-0            58b9635f96a3        9 months ago        185MB
rancher/scheduler           v0.8.6              fbedeaddc3e9        17 months ago       248MB
rawmind/alpine-traefik      1.7.4-0             4a7e515f5e05        20 months ago       154MB
rancher/agent               v1.2.11             1cc7591af4f5        23 months ago       243MB
rancher/net                 v0.13.17            f170c38e3763        23 months ago       311MB
rancher/dns                 v0.17.4             678bde0de4d2        24 months ago       249MB
rancher/healthcheck         v0.3.8              ce78cf69cc0b        24 months ago       391MB
rancher/metadata            v0.10.4             02104eb6e270        24 months ago       251MB
rancher/network-manager     v0.7.22             13381626c510        24 months ago       256MB
rancher/hello-world         latest              cab3bc026f39        2 years ago         18.2MB
nginx                       latest              958a7ae9e569        3 years ago         109MB
rancher/net                 holder              665d9f6e8cc1        3 years ago         267MB
rancher/lb-service-haproxy  v0.6.4              9b8c61dc1db6        3 years ago         335MB
tutum/hello-world           latest              31e17b0746e4        4 years ago         17.8MB
root@iZff6nq8h7e9rkZ:~#
```

图 3.27 查看 MySQL 镜像

3. 将 MySQL 镜像进行修改、封装，push 到私有镜像库中。

图 3.28 commit 封装镜像

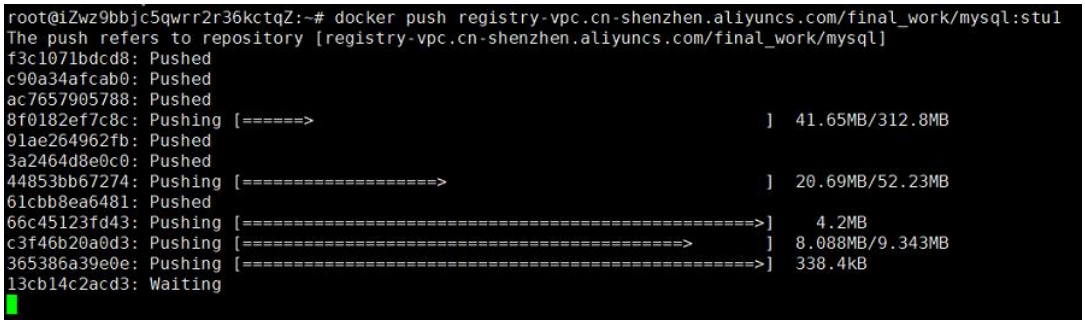

图 3.29 push MySQL 镜像到私有镜像库中

4. 从 Rancher 中对指定的一台主机添加 MySQL 服务。

图 3.30 添加 MySQL 服务

2.4.2 添加结果

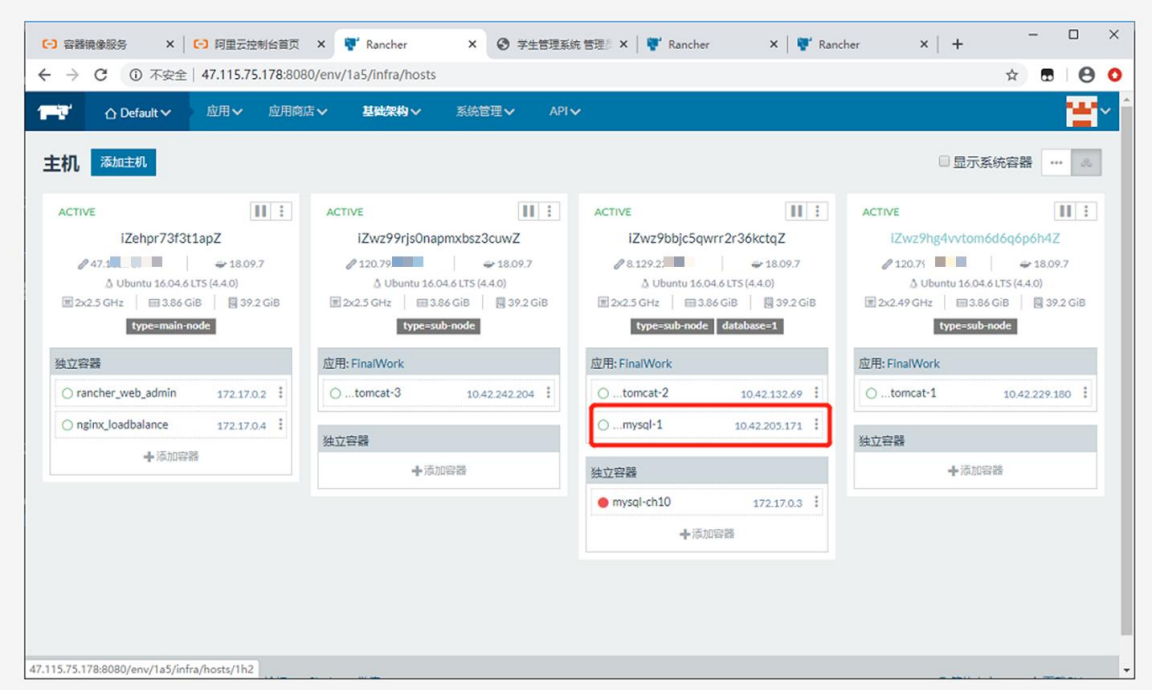

图 3.31 构建 MySQL 结果

2.5 阿里云专有网络（VPC）跨账号互联互通

2.5.1 实现详情

1. 建立云企业网

图 3.32 建立云企业网

2. 在其他主机上授权

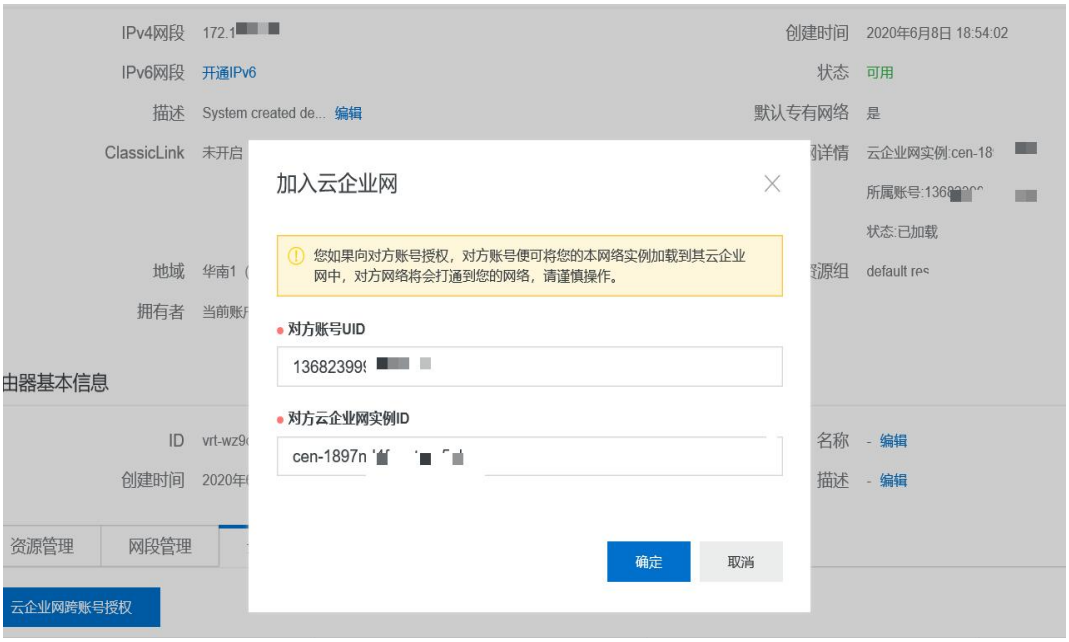

图 3.33 授权

3. 在主账号的云企业网中加载其他账号的专有网络实例

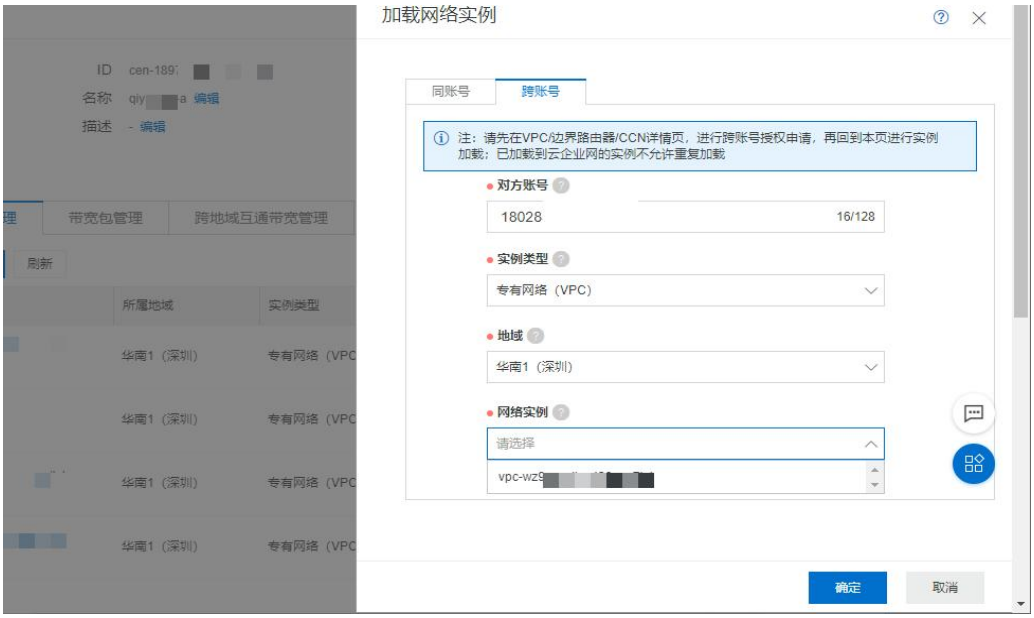

图 3.34 加载网络示例

4. 查看路由表是否构建成功


网络实例管理
带宽包管理
跨地域互通带宽管理
路由信息
云服务
Private Zone
路由策略
地域
华南1 (深圳)
刷新
目标网段
路由类型
匹配策略
路由属性
下一跳
地域 华南1 (深圳)
类型 VPC
实例ID vpc-wz982
去其他地域策略
100.6
系统
查看详情
172.
云企业网
查看详情
172.
云企业网
查看详情
172.
云企业网
查看详情
可用
华南1 (深圳)
192.
云企业网
查看详情
可用
华南1 (深圳)

图 3.35 查看路由表

2.6 Nginx 负载均衡服务搭建

负载均衡建立在现有网络结构之上，它提供了一种廉价有效透明的方法扩展网络设备和服务器的带宽、增加吞吐量、加强网络数据处理能力、提高网络的灵活性和可用性。

负载均衡（Load Balance）其意思就是分摊到多个操作单元上进行执行，例如 Web 服务器、FTP 服务器、企业关键应用服务器和其它关键任务服务器等，从而共同完成工作任务。

搭建过程：

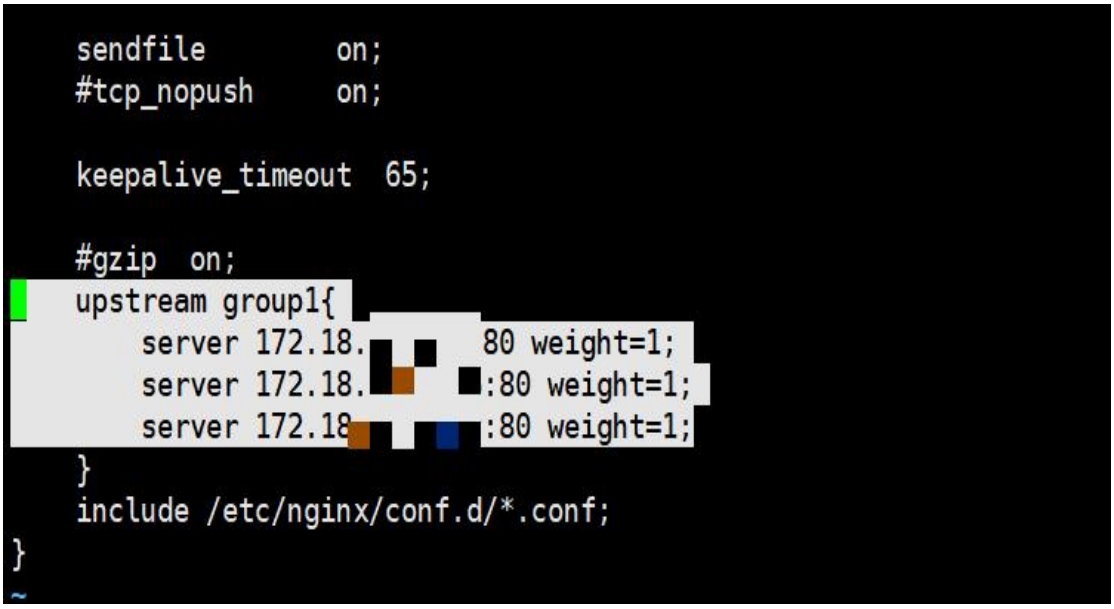

图 3.37 添加负载均衡组

2.6.1 编辑/etc/nginx/nginx.conf 配置文件

加入如下配置信息

```
upstream group1{
    server 172.18.***.*1:80 weight=1;
    server 172.18.***.*2:80 weight=1;
    server 172.18.***.*3:80 weight=1;
    }
```

说明：建立一个负载均衡组，组名为 group1，内有 3 个节点，分别为 172.18.***.*1:80；172.18.***.*2:80；172.18.***.*3:80，为了方便演示，三个节点的负载权重均为 1，在实际部署中，可以各节点的实际业务负载灵活地设置出合适地权重值

```
#charset koi8-r;
#access_log  /var/log/nginx/host.access.log  main;

location / {
   # root   /usr/share/nginx/html;
   # index  index.html index.htm;
    proxy_pass http://group1;
}

#error_page  404              /404.html;

# redirect server error pages to the static page /50x.html
#
error_page   500 502 503 504  /50x.html;
```

图 3.38 对负载均衡组进行声明

2.6.2 编辑/etc/conf.d/default.conf 配置文件

加入 proxy_pass http://group1 一句，表示配置匹配 Web 服务根目录的所有请求到负载均衡组 group1。

2.6.3 搭建结果

使用公网 IP 为 47.115.**.***的主机作为对外提供 Web 服务的反向代理，通过 nginx 进行反向代理，访问 3 个内网 Tomcat 容器主机，load.html 写入的是每台主机的内网 IP 地址，通过不断刷新页面，发现界面显示内容会不断变化，证明负载均衡服务部署成功。

3 系统实现

3.1 Web 应用效果

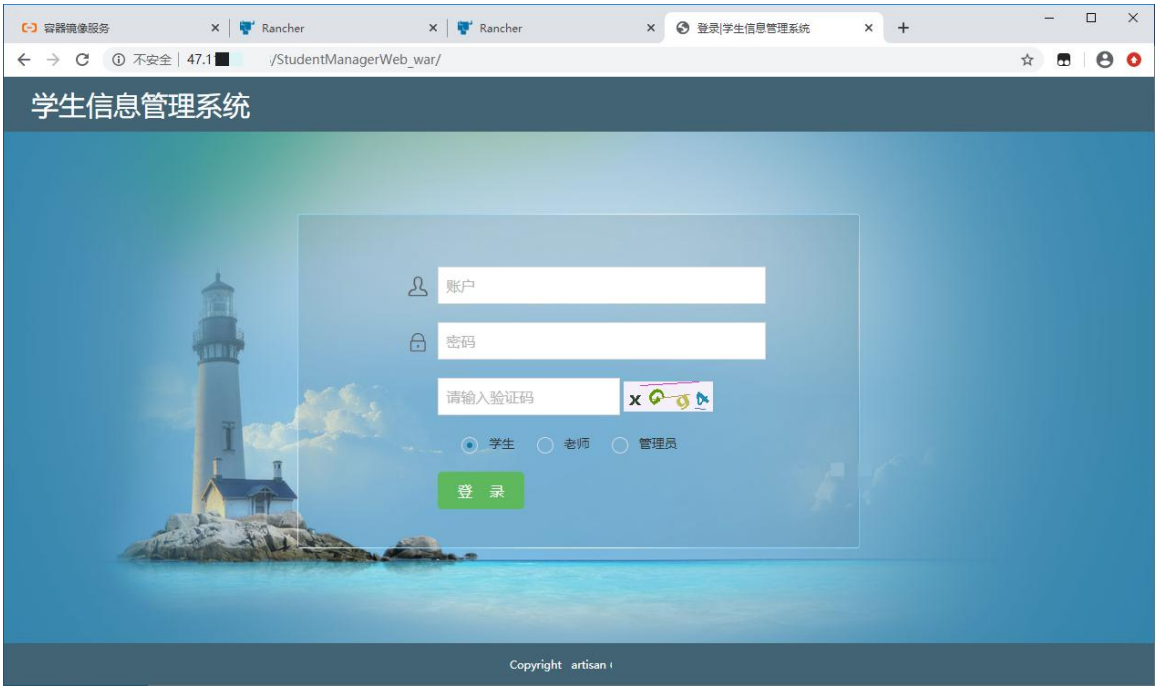

图 4.4 Web 应用效果 1