# 练习题

## 1、基础运维任务（5分）

###### 1、基础环境配置（2分）

使用提供的用户名密码，登录竞赛用的云计算平台，自行使用CentOS7.5镜像创建两台云主机，flavor使用4v\_8G\_100G\_50G的配置，第一张网卡使用提供的net网络，第二张网卡使用的网络自行创建（网段为10.10.X.0/24，X为工位号）。创建完云主机后确保网络正常通信，然后按以下要求配置服务器：

（1）设置控制节点主机名为controller，设置计算节点主机名为compute；

（2）修改hosts文件将IP地址映射为主机名；

完成后提交提交过程中所用到的命令以及查询结果。

###### 2、Yum源配置（1分）

使用提供的http服务地址，在http服务下，存在centos7.5和iaas的网络yum源，使用该http源作为安装iaas平台的网络源。分别设置controller节点和compute节点的yum源文件http.repo。完成后提交提交过程中所用到的命令以及查询结果。

###### 3、时间同步配置（1分）

在controller节点上部署chrony服务器，允许其他节点同步时间，启动服务并设置为开机启动；在compute节点上指定controller节点为上游NTP服务器，重启服务并设为开机启动。完成后提交提交过程中所用到的命令以及查询结果。

###### 4、计算节点分区（1分）

在compute节点上利用空白分区划分2个20G分区。完成后提交用到的命令和查询的结果

## 2、OpenStack搭建任务（15分）

###### 1、基础安装（1分）

在控制节点和计算节点上分别安装iaas-xiandian软件包，根据表2配置两个节点脚本文件中的基本变量（配置脚本文件为/etc/xiandian/openrc.sh）。

表2 云平台配置信息

|  |  |  |
| --- | --- | --- |
| 服务名称 | 变量 | 参数/密码 |
| Mysql | root | 000000 |
| Keystone | 000000 |
| Glance | 000000 |
| Nova | 000000 |
| Neutron | 000000 |
| Heat | 000000 |
| Zun | 000000 |
| Keystone | DOMAIN\_NAME | demo |
| Admin | 000000 |
| Rabbit | 000000 |
| Glance | 000000 |
| Nova | 000000 |
| Neutron | 000000 |
| Heat | 000000 |
| Zun | 000000 |
| Neutron | Metadata | 000000 |
| External Network | ensXXX/enpXXX（第二块网卡名） |

完成后提交查询结果。

###### 2、数据库安装（2分）[mysql.md](file:///C:\Users\34934\Desktop\新建文件夹\mysql.md)

在controller节点上使用iaas-install-mysql.sh 脚本安装Mariadb、Memcached、RabbitMQ等服务。安装服务完毕后，完成下列题目。

（1）登录数据库服务，创建库test，并在库test中创建表company（表结构如(id int not null primary key,name varchar(50),addr varchar(255))所示），在表company中插入一条数据(1,"alibaba","china")。

（2）.使用RabbitMQ服务的相关命令创建用户chinaskill，密码为chinapd，并赋予该用户administrator权限。

完成后提交查询数据库数据和查询rabbitmq用户的结果截图或文本。

{

}

###### 3、Keystone服务安装（2分）

在controller节点上使用iaas-install-keystone.sh 脚本安装Keystone服务。完成后提交提交过程中所用到的命令以及查询结果。

###### 4、Glance安装（1分）

在controller节点上使用iaas-install-glance.sh脚本安装glance 服务。完成后提交提交过程中所用到的命令以及查询结果。、

###### 5、Nova安装（2分）

在controller节点和compute节点上分别使用iaas-install-nova -controller.sh脚本、iaas-install-nova-compute.sh脚本安装Nova服务。完成后提交提交过程中所用到的命令以及查询结果。

###### 6、Neutron安装（1分）

在controller节点和compute节点上分别修改iaas-install-neutron-controller.sh脚本、iaas-install-neutron-compute.sh脚本分别安装 Neutron 服务，执行完脚本后，网络默认是vlan模式。完成后提交提交过程中所用到的命令以及查询结果。

###### 7、Doshboard安装（1分）

在controller节点上使用iaas-install-dashboad.sh脚本安装dashboad服务。完成后提交提交过程中所用到的命令以及查询结果。

###### 8、Swift安装（2分）[swift.md](file:///C:\Users\34934\AppData\Roaming\Microsoft\Word\swift.md)

在控制节点和计算节点上分别使用iaas-install-swift-controller.sh和iaas-install-swift-compute.sh脚本安装Swift服务并创建test容器。完成后提交提交过程中所用到的命令以及查询结果。

###### 9、Cinder 安装（2分）

在控制节点和计算节点分别使用iaas-install-cinder-controller.sh、iaas-install-cinder-compute.sh脚本安装Cinder服务，使用cinder命令创建一个名字叫blockvolume，大小为2G的云硬盘。完成后提交提交过程中所用到的命令以及查询结果。

###### 10、Heat安装（1分）

在控制节点上使用iaas-install-heat.sh脚本安装Heat服务。完成后提交提交过程中所用到的命令以及查询结果。该题为云平台搭建最后一题，做完该题后请不要删除自行搭建的云平台，在后面的运维题目中会使用到该平台。

## 3、OpenStack运维任务（15分）

###### 1、glance 镜像管理（1分）[glance.md](file:///C:\Users\34934\AppData\Roaming\Microsoft\Word\glance.md)

使用自己搭建的openstack云平台，使用http:///cirros-0.3.4-x86\_64-disk.img镜像，使用命令创建一个名为cirros的镜像。完成后提交提交过程中所用到的命令以及查询结果。

###### 2、heat管理（3分）[heat.md](file:///C:\Users\34934\AppData\Roaming\Microsoft\Word\heat.md)

使用自己搭建的openstack云平台，在/root目录下编写模板server.yaml，创建名为“m1.flavor”、 ID 为 1234、内存为1024MB、硬盘为20GB、vcpu数量为 1的云主机类型。完成后提交提交过程中所用到的命令以及查询结果。（在提交信息前请准备好yaml模板执行的环境）

###### 3、网络管理（1分）

使用自己搭建的openstack云平台，创建云主机网络extnet，子网extsubnet，虚拟

网段为192.168.100.0/24， 网关为192.168.100.1，段ID默认写100。完成后提交提交过程中所用到的命令以及查询结果。

###### 4、云主机管理（1分）

使用自己搭建的openstack云平台，基于“cirros”镜像、flavor使用“m1.flavor”、extnet的网络，创建一台虚拟机VM1，并启动VM1。提交提交过程中所用到的命令以及查询结果。

###### 5、快照管理（1分）[openstack\快照管理.md](file:///C:\Users\34934\AppData\Roaming\Microsoft\Word\openstack\快照管理.md)

使用自己搭建的openstack云平台，将云主机VM1保存为qcow2格式的快照并保存到controller节点/root/cloudsave目录下，保存名字为csccvm.qcow2。完成后提交云主机的用户名、密码和IP地址到答题框。

###### 6、虚拟机调整flavor（2分）

使用自己搭建的openstack云平台，使用cirros镜像，flavor使用1vcpu/512M内存/10G硬盘，创建云主机cscc\_vm，假设在使用过程中，发现该云主机配置太低，需要调整，请修改相应配置，将dashboard界面上的云主机调整实例大小可以使用，将该云主机实例大小调整为1vcpu/1G内存/20G硬盘。完成后提交所修改配置文件节点的IP地址、用户名和密码到答题框。

###### 7、swift分片存储

使用自己搭建的openstack云平台，自行安装Swift服务，新建名为chinaskill的容器，将cirros-0.3.4-x86\_64-disk.img镜像上传到chinaskill容器中，并设置分段存放，每一段大小为10M。完成后提交提交过程中所用到的命令以及查询结果。

###### 8、云平台安全策略提升

使用OpenStack私有云平台，使用http:///https.tar.gz中的软件包，安装必要组件，将私有云平台的访问策略从http提升至https。完成后提交提交过程中所用到的命令以及查询结果。

###### 9、RabbitMQ集群

使用提供的OpenStack私有云平台，创建三个centos7.5系统的云主机，使用http:///RabbitMQ.tar.gz中的软件包安装RabbitMQ服务，安装完毕后，搭建RabbitMQ集群，并打开RabbitMQ服务的图形化监控页面插件。集群使用普通集群模式，其中第一台做磁盘节点，另外两台做内存节点。完成后提交磁盘节点的用户名、密码和IP地址到答题框

## 4、OpenStack运维开发任务(15分)

###### 1、使用python调用api实现创建flavor(7分)

使用自己搭建的openstack云平台，在controller节点上，根据http:///Python-api.tar.gz中的软件，完成python3.6软件和依赖库的安装。在controller节点的/root目录下创建create\_flavor.py文件，在该文件中编写python代码对接openstack api，要求在openstack私有云平台上创建一个云主机类型，名字为test、vcpu为1个、内存为1024m、硬盘为20G、ID为199999。执行完代码要求输出“云主机类型创建成功”。根据上述要求编写python代码，完成后，提交安装过程中用到的命令以及最终的代码。

###### 2、Ansible脚本开发(8分)

使用OpenStack私有云平台，创建4台系统为centos7.5的云主机，其中一台作为Ansible的母机并命名为ansible，另外三台云主机命名为node1、node2、node3，通过http:///ansible.tar.gz软件包在ansible节点安装Ansible服务；使用这一台母机，编写Ansible脚本（在/root目录下创建example目录作为Ansible工作目录，部署的入口文件命名为cscc\_install.yaml），对其他三台云主机进行安装高可用数据库集群（MariaDB\_Galera\_cluster，数据库密码设置为123456）的操作（可以使用gpmall.gz中的gpmall-repo目录作为yum源安装数据库服务）。完成后提交Ansible的hosts文件以及最终的脚本代码。

## 5、Docker CE及私有仓库安装任务（5分）

###### 1、安装Docker CE和Docker Compose（1分）

登录提供的私有云平台，使用centos-paas镜像创建master和node节点。启动后在master和node节点中分别安装并启动DockerCE和docker-compose服务。Chinaskills \_cloud\_paas.iso在云主机的root目录下。完成后提交过程中所用到的命令以及查询结果。

###### 2、安装私有仓库（1分）

在master节点安装私有仓库，导入/opt/images目录下所有镜像，并推送到私有仓库。完成后提交过程中所用到的命令以及查询结果。

###### 3、容器编排（3分）

在master节点上编写/root/redis/docker-compose.yaml文件编排部署Redis集群，具体要求如下：

（1）实现Redis集群一主二从，主节点容器名称：redis-master，从节点容器名称：redis-slave-1、redis-slave-2；

（2）所有容器重启策略：always

（3）Redis密码：123456；

（4）主节点端口映射：6370：

完成后编排部署Redis集群，并提交过程中所用到的命令以及查询结果。

## 6、基于Docker容器的web应用系统部署(15分)

###### 1、容器化部署Redis(3分)

在master节点上编写/root/redis/Dockerfile文件构建chinaskill-redis:v1.1镜像，具体要求如下：（需要用到的软件包路径http:///gpmall-single.tar.gz）

（1）基础镜像：centos:centos7.5.1804；

（2）作者：Chinaskill；

（3）修改配置文件中的bind 127.0.0.1为bind 0.0.0.0；

（4）设置Redis免密，并关闭保护模式；

（5）开放端口：6379；

（6）设置服务开机自启。

完成后构建镜像，并提交过程中所用到的命令以及查询结果。

###### 2、容器化部署MariaDB(3分)

在master节点上编写/root/mariadb/Dockerfile文件构建chinaskill-mariadb:v1.1镜像，具体要求如下：（需要用到的软件包路径http:///gpmall-single.tar.gz）

（1）基础镜像：centos:centos7.5.1804；

（2）作者：Chinaskill；

（3）设置数据库密码：123456；

（4）创建数据库gpmall并导入数据库文件gpmall.sql；

（5）设置字符编码：UTF-8；

（6）开放端口：3306；

（7）设置服务开机自启。

完成后构建镜像，并提交过程中所用到的命令以及查询结果。

###### 3、容器化部署Zookeeper(2分)

在master节点上编写/root/zookeeper/Dockerfile文件构建chinaskill-zookeeper:v1.1镜像，具体要求如下：（需要用到的软件包路径http:///gpmall-single.tar.gz）

（1）基础镜像：centos:centos7.5.1804；

（2）作者：Chinaskill；

（3）开放端口：2181；

（4）设置服务开机自启。

完成后构建镜像，使用构建的镜像运行容器myzookeeper，并提交过程中所用到的命令以及查询结果。

###### 4、容器化部署Kafka(2分)

在master节点上编写/root/kafka/Dockerfile文件构建chinaskill-kafka:v1.1镜像，具体要求如下：（需要用到的软件包路径http:///gpmall-single.tar.gz）

（1）基础镜像：centos:centos7.5.1804；

（2）作者：Chinaskill；

（3）开放端口：9092；

（4）设置服务开机自启。

完成后构建镜像，并提交过程中所用到的命令以及查询结果。

###### 5、容器化部署Nginx(3分)

在master节点上编写/root/nginx/Dockerfile文件构建chinaskill-nginx:v1.1镜像，具体要求如下：（需要用到的软件包路径http:///gpmall-single.tar.gz）

（1）基础镜像：centos:centos7.5.1804；

（2）作者：Chinaskill；

（3）编写/etc/nginx/conf.d/default.conf文件，配置反向代理，将80端口请求转发到8081、8082和8083；

（4）将dist中的文件复制到/usr/share/nginx/html/目录下；

（5）复制所有的jar包到镜像中；

（6）开放端口：80、443、8081、8082、8083；

（7）设置服务开机自启。

完成后构建镜像，并提交过程中所用到的命令以及查询结果。

###### 6、编排部署商城(2分)

在master节点上编写/root/chinaskillmall/docker-compose.yaml文件部署，具体要求如下：

（1）容器1名称：mall-mysql；镜像：chinaskill-mariadb:v1.1；端口映射：13306:3306；

（2）容器2名称：mall-redis；镜像：chinaskill-redis:v1.1；端口映射：16379:6379；

（3）容器3名称：mall-kafka；镜像：chinaskill-kafka:v1.1；端口映射：19092:9092；

（4）容器4名称：mall-zookeeper；镜像：chinaskill-zookeeper:v1.1；端口映射：12181:2181；

（5）容器5名称：mall-nginx；镜像：chinaskill-nginx:v1.1；端口映射：83:80，1443:443；自启动所有的jar包程序。

完成后编排部署商城，并能成功访问商城首页。然后提交过程中所用到的命令以及查询结果。

## 7、基于Kubernetes构建持续集成(15分)

###### 1、安装Jenkins环境(3分)

在master节点上使用镜像jenkins/jenkins:2.262-centos部署Jenkins服务，具体要求如下：

（1）容器名称：jenkins；

（2）端口映射：8080:8080；

（3）使用root身份生成容器；

（4）离线安装Jenkins插件；

（5）设置Jenkins用户：chinaskill；密码：000000；

（6）在授权策略中配置“任何用户可以做任何事(没有任何限制)”。

使用chinaskill用户登录Jenkins，完成后提交过程中所用到的命令以及查询结果。（需要用到的软件包路径yun2016$\Jenkins\_offline.tar）

###### 2、安装Gitlab环境(2分)

在master节点上使用镜像gitlab/gitlab-ce:12.9.2-ce.0部署Gitlab服务，具体要求如下：

（1）容器名称：mygitlab；

（2）端口映射：1022:22、81:80、443:443；

（3）容器重启策略：always；

（4）设置root用户及密码；

（5）使用root用户登录Gitlab，密码：00000000；

（6）新建项目ChinaskillProject，将/opt/ChinaskillProject中的代码上传到ChinaskillProject项目中。

完成后提交过程中所用到的命令以及查询结果。（需要用到的软件包路径<http:///Jenkins_offline.tar>）

###### 3、配置Jenkins连接Gitlab(2分)

配置Jenkins连接Gitlab，具体要求如下：

（1）设置Outbound requests；

（2）生成“Access Tokens”并命名为jenkins；

（3）设置Jenkins取消对'/project' end-point进行身份验证；

（4）测试Jenkins与Gitlab的连通性。

完成后提交过程中所用到的命令以及查询结果。（需要用到的软件包路径yun2016$\[Jenkins\_offli ne.tar](http:///Jenkins_offline.tar)）

###### 4、配置Jenkins连接maven(2分)

配置Jenkins连接maven，具体要求如下：

（1）采用docker in docker的方式在Jenkins内安装maven；

（2）在Jenkins中配置maven信息。

完成后提交过程中所用到的命令以及查询结果。（需要用到的软件包路径yun2016$\[Jenkins\_offline.tar](http:///Jenkins_offline.tar)）

###### 5、配置CI/CD(2分)

配置CI/CD，具体要求如下：

（1）新建一个流水线任务ChinaskillProject；

（2）编写流水线脚本，构建ChinaskillProject项目中的gateway和config服务，将构建后的镜像自动上传到Harbor仓库的chinaskillproject项目，并自动发布gateway和config服务到Kubernetes集群的springcloud命名空间下；

（3）配置Webhook；

（4）在Harbor中新建公开项目chinaskillproject。

完成后提交过程中所用到的命令以及查询结果。（需要用到的软件包路径yun2016$\[Jenkins\_offline.tar](http:///Jenkins_offline.tar)）

###### ，6、触发CI/CD(2分)

将ChinaskillProject项目中的代码上传到Gitlab中触发构建，完成后提交过程中所用到的命令以及查询结果。（需要用到的软件包路径：yun2016$\[Jenkins\_offli ne.tar](http:///Jenkins_offline.tar) ）

## 8、Kubernetes容器云平台部署与运维(15分)[..\容器云\Kubernetes容器云平台部署与运维.md](file:///C:\Users\34934\AppData\Roaming\Microsoft\容器云\Kubernetes容器云平台部署与运维.md)

###### 1、修改Pod数量限制(1分)

Kubernetes 默认每个节点只能启动110个Pod，由于业务需要，需要在所有节点上运行大量资源消耗非常小的Pod，请将每个节点默认限制的Pod数量改为200。完成后提交过程中所用到的命令以及查询结果。

###### 2、修改NodePort端口范围(1分)

Kubernetes以NodePort方式暴露服务默认的端口范围为30000-32767，请将NodePort的端口范围修改为20000-65535。完成后提交过程中所用到的命令以及查询结果。

###### 3、 Pod管理(2分)

在master节点/root目录下编写yaml文件nginx.yaml，具体要求如下：

（1）Pod名称：nginx-pod；

（2）命名空间：default；

（3）容器名称：mynginx；

（4）镜像：nginx；拉取策略：IfNotPresent；

（5）容器端口：80。

完成后使用该yaml文件创建Pod，并提交过程中所用到的命令以及查询结果。

###### 4、Pod安全策略(2分)

在master节点/root目录下编写yaml文件policy.yaml，具体要求如下：

（1）安全策略名称：pod-policy；

（2）仅禁止创建特权模式的Pod；

（3）其它所有字段都被允许。

完成后使用该yaml文件创建安全策略，并提交过程中所用到的命令以及查询结果。

###### 5、Deployment管理(2分)

在master节点/root目录下编写yaml文件nginx-deployment.yaml，具体要求如下：

（1）Deployment名称：nginx-deployment；

（2）命名空间：default；

（3）Pod名称：nginx-deployment，副本数：2；

（4）网络：hostNetwork；

（5）镜像：nginx；

（6）容器端口：80

完成后使用该yaml文件创建Deployment，并提交过程中所用到的命令以及查询结果。

###### 6、RBAC管理(2分)

在master节点/root目录下编写yaml文件role.yaml，具体要求如下：

（1）Role名称：pod-reader；

（2）命名空间：default；

（3）对default命名空间内的Pod拥有get、watch、list的权限。

完成后使用该yaml文件创建Role，并提交过程中所用到的命令以及查询结果。

###### 7、HPA管理(2分)

在master节点/root目录下编写yaml文件deployment-hpa.yaml，具体要求如下：

（1）HPA名称：deployment-hpa；

（2）命名空间：default；

（3）基于deployment进行伸缩，副本数伸缩范围：1--10；

（4）期望每个Pod的CPU和内存使用率为50%。

完成后使用该yaml文件创建HPA，并提交过程中所用到的命令以及查询结果。

###### 8、更新证书(3分)

Kubernetes默认的证书有效期只有一年时间，对于某些场景下一个足够长的证书有效期是非常有必要的。请将Kubernetes集群证书的有效期延长至10年。完成后提交过程中所用到的命令以及查询结果（需要用到的软件包在/桌面/附件/容器云附件/kubernetes\_v1.18.1.tar.gz）