1.比赛时间安排：本项目比赛总时间为240分钟，模块A私有云平台的搭建与运维占比30%，模块B容器的编排与运维占比40%，模块C企业级应用的自动化部署和运维占比30%

|  |
| --- |
| A模块：OpenStack平台部署与运维（样题） **业务场景：**  某企业拟使用OpenStack搭建一个企业云平台，用于部署各类企业应用对外对内服务。云平台可实现IT资源池化，弹性分配，集中管理，性能优化以及统一安全认证等。系统结构如下图：  企业云平台的搭建使用竞赛平台提供的两台云服务计算节点和控制节点控制器，配置如下表：    说明:  ①选手自行检查工位pc机硬件及网络是否正常;1.选手自行检查工位PC机硬件及网络是否正常；  ②竞赛使用集群模式进行，给每个参赛队提供账号和密码及考试系统的账号和密码。选手通过用户名与密码分别登录云和考试系统;  ③考试用到的软件包都在云主机/opt下。  ④表1中的公网IP和私网IP以自己云主机显示为准，每个人的公网IP和私网IP不同。使用第三方软件远程连接云主机，使用公网IP连接。 任务1私有云平台环境初始化 ①根据表1中的IP地址规划，设置各服务器节点的IP地址，确保网络正常通信，设置云服务器1主机名为Controller，云服务器2主机名为Compute，并修改hosts文件将IP地址映射为主机名，关闭防火墙并设置为开机不启动，设置SELinux为Permissive 模式。  ②将提供的CentOS-7-x86\_64-DVD-1804.iso和qdkills\_cloud\_iaas.iso光盘镜像上传到Compute节点的/root目录下，然后在/opt目录下分别创建centos目录和openstack目录，并将镜像文件CentOS-7-x86\_64-DVD-1804.iso挂载到centos目录下，将镜像文件qdkills\_cloud\_iaas.iso挂载到openstack目录下。  ③在Compute节点上利用centos目录中的软件包安装vsftpd服务器并设置开机自启动，提供yum仓库服务，并分别设置controller节点和compute节点的yum源文件ftp.repo，其中节点的地址使用IP形式。  ④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；并在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。  ⑤在compute节点上查看分区情况，并利用空白分区划分2个20G分区。 任务2 OpenStack平台搭建 ①在 controller 节点和 compute 节点分别安装 iaas-xiandian 软件包，修改脚本文件基本变量（脚本文件为/etc/xiandian/openrc.sh），修改完成后使用命令生效该脚本文件。  ②在 compute 节点配置/etc/xiandian/openrc.sh 文件，根据环境情况修改参数，块存储服务的后端使用第二块硬盘的第一个分区，生效该参数文件。  ③分别在 controller 节点和 compute 节点执行 iaas-pre-host.sh 文件(不需要重启云主机)。  ④在 controller 节点执行 iaas-install-mysql.sh 脚本，会自行安装 mariadb、memcached、rabbitmq 等服务和完成相关配置。执行完成后修改配置文件将缓存 CACHESIZE 修改为 128,并重启相应服务。  ⑤在 controller 节点执行 iaas-install-keystone.sh 脚本，会自行安装 keystone 服务和完成相关配置。使用 openstack 命令，创建一个名为 tom 的账户，密码为 tompassword123,邮箱为[tom@example.com](mailto:tom@example.com)  ⑥在 controller 节点执行 iaas-install-glance.sh 脚本，会自行安装 glance 服务和完成相关配 置 。 完 成 后 使 用 openstack 命 令 , 创 建 一 个 名 为 cirros 的 镜 像 ， 镜 像 文 件 使 用cirros-0.3.4-x86\_64-disk.img。  ⑦在 controller 节点执行 iaas-install-nova-controller.sh，compute 节点执行iaas-install-nova-compute.sh，会自行安装 nova 服务和完成相关配置。使用 nova 命令创建一个名为 t,ID 为 5，内存为 2048MB,磁盘容量为 10GB,vCPU 数量为 2 的云主机类型。  ⑧在 controller 节点执行 iaas-install-neutron-controller.sh,compute 节点执行iaas-install-neutron-compute.sh，会自行安装 neutron 服务并完成配置。创建云主机外部网络 ext-net，子网为 ext-subnet，云主机浮动 IP 可用网段为192.168.10.100 ~ 192. 168.10.200，网关为 192.168.100.1。  ⑨在 controller 节点执行 iaas-install-dashboard.sh 脚本，会自行安装 dashboard 服务并完成配置。请修改 nova 配置文件，使之能通过公网 IP 访问 dashboard 首页。 任务 3 OpenStack 运维任务 ①使用命令创建名称为 group\_web 的安全组该安全组的描述为” Custom security group”，用 openstack 命令为安全组添加 icmp 规则和 ssh 规则允许任意 ip 地址访问 web,完成后查看该安全组的详细信息.  ②在 keystone 中创建 shop 项目添加描述为”Hello shop”，完成后使用 openstack 命令禁用该项目，然后使用 openstack 命令查看该项目的详细信息。  ③使用 nova 命令查看 admin 租户的当前配额值，将 admin 租户的实例配额提升到 13。登 录 controller 节 点 ，使用 glance 相 关 命 令 上 传 镜 像 ， 源 使 用CentOS\_7.5\_x86\_64\_XD.qcow2，名字为 centos7.5，修改这个镜像为共享状态，并设置最小磁盘为 5G。  ④请修改 glance 后端配置文件，将项目的映像存储限制为 10GB,完成后重启 glance 服务。  ⑤在 controller 节点执行 iaas-install-cinder-controller.sh, compute 节点执行iaas-install-cinder-compute.sh，在 controller 和 compute 节点上会自行安装 cinder 服务并完成配置。创建一个名为 lvm 的卷类型，创建该类型规格键值对，要求 lvm 卷类型对应 cinder后端驱动 lvm 所管理的存储资源,名字 lvm\_test，大小 1G 的云硬盘并查询该云硬盘的详细信息。  ⑥请使用数据库命令将所有数据库进行备份,备份文件名为 openstack.sql，完成后使用命令查看文件属性其中文件大小以 mb 显示。  ⑦进入数据库，创建本地用户 examuser，密码为 000000，然后查询 mysql 数据库中的user 表的 user,host,password 字段。然后赋予这个用户所有数据库的“查询”“删除”“更新”“创建”的权限。  ⑧请使用 openstack 命令创建一个名为 test 的 cinder 卷，卷大小为 5G。完成后使用 cinder命令列出卷列表并查看 test 卷的详细信息。  ⑨为了减缓来自实例的数据访问速度的变慢，OpenStack Block Storage 支持对卷数据复制带宽的速率限制。请修改 cinder 后端配置文件将卷复制带宽限制为最高 100 MiB/s。  ⑩在controller节点执行 iaas-install-swift-controller.sh, compute节点执行iaas-install-swift-compute.sh,在controller和compute节点上会自行安装 swift 服务并完成配置。创建一个名为 file 的容器。  修改openrc.sh中的内容，然后在controller节点执行iaas-install-nova-compute.sh，把controller节点的资源添加到云平台。 任务四 OpenStack 架构任务（3 分） 公司内部拥有一套私有云系统，为了调试该私有云，需要编写一些测试用脚本进行功能性测试，作为公司私有云维护人员请你完成以下工作。  1.请使用 openstack 命令创建一个浮动 IP 地址，完成后使用 openstack 命令查看该浮动IP 的 id，请编写一个名为 floating\_show.sh 的脚本，该脚本$1 变量为浮动 ip 的 id，对接 neutron服务端点获取该浮动 IP 的详细信息。脚本使用 curl 向 api 端点传递参数，为了兼容性考虑不得出现 openstack 命令。  2.请编写脚本 floating\_delete.sh，完成浮动 IP 的删除。设置一个$1 变量，当用户向$1 传递一个浮动 IP 的 id，即可完成该浮动 IP 的删除。脚本使用 curl 向 api 端点传递参数，为了兼容性考虑不得出现 openstack 命令。 |
| B模块：容器的编排与运维（样题） 说明：本任务提供有4台服务器master、node1、node2和harbor，都安装了centos7.5操作系统，在/opt/centos目录下有CentOS-7-x86\_64-DVD-1804系统光盘文件所有文件，在/opt/containerk8s目录下有本次容器云运维所需的所有文件。   任务 1 容器云平台环境初始化 ①master 节点主机名设置为 master、node1 节点主机名设置为 node1、node2 节点主机名设置为 node2、harbor 节点主机名设置为 harbor,所有节点关闭 swap，并配置 hosts 映射。  ②将提供的 CentOS-7-x86\_64-DVD-1804.iso 和 qdkills\_cloud\_paas.iso 光盘镜像文件移动到 master 节点 /root 目录下，然后在 /opt 目录下使用命令创建 centos 目录和 paas 目录，并将镜像文件 CentOS-7-x86\_64-DVD-1804.iso 永久挂载到 centos 目录下，将镜像文件qdskills\_cloud\_paas.iso 永久挂载到 /opt/paas 目录下。  ③在 master 节点首先将系统自带的 yum 源移动到/home 目录，然后为 master 节点配置本地 yum 源，yum 源文件名为 local.repo。  ④在 master 节点安装 ftp 服务，将 ftp 共享目录设置为 /opt/。  ⑤为 node1 节点和 node2 节点分别配置 ftp 源，yum 源文件名称为 ftp.repo，其中 ftp 服务器地址为 master 节点,配置 ftp 源时不要写 IP 地址，配置之后，两台机器都安装 kubectl 包作为安装测试。  ⑥在 master 节点上部署 chrony 服务器，允许其它节点同步时间，启动服务并设置为开机自启动；在其他节点上指定 master 节点为上游 NTP 服务器，重启服务并设为开机自启动。  ⑦为四台服务器设置免密登录，保证服务器之间能够互相免密登录。 任务2 k8s 搭建任务 ①在所有节点上安装dokcer-ce,并设置为开机自启动。  ②所有节点配置阿里云镜像加速地址(<https://7hw6x2is.mirror.aliyuncs.com>)并把启动引擎设置为systemd，配置成功后加载配置文件并重启docker服务。  ③在master节点/opt/images目录下使用tar归档文件载入镜像。  ④在master节点使用/opt/docker-compose/v2.10.2-docker-compose-linux-x86\_64文件安装docker-compose。安装完成后执行docker-compose version命令。  ⑤在master节点解压/opt/harbor/ harbor-offline-installer-v2.5.3.tgz离线安装包，然后安装harbor仓库，并修改相应的yml文件，使各节点默认docker仓库为harbor仓库地址。  ⑥在master节点执行/opt/k8s\_image\_push.sh将所有镜像上传至docker仓库。  ⑦执行/opt/k8s\_con\_ner\_bui\_install.sh部署Kubeadm、containerd、nerdctl和buildkit。  ⑧使用 kubeadm 命令生成 yaml 文件，并修改 yaml 文件，设置 kubernetes 虚拟内部网段地址为 10.244.0.0/16，通过该 yaml 文件初始化 master 节点，然后使用 kube-flannel.yaml 完成控制节点初始化设置。  ⑨使用 kube-flannel.yaml 安装 kubernetes 网络插件，安装完成后使用命令查看节点状态。  ⑩给kubernetes创建证书,命名空间为kubernetes-dashboard,涉及到的所有文件命名为dashboard例如dashboard.crt。  修改/opt/yaml/dashboard/recommended.yaml的镜像来源为本地Harbor仓库,然后使用/opt/yaml/dashboard/recommended.yaml和/opt/yaml/dashboard/dashboard-adminuser.yaml安装kubernetes dashboard界面，完成后查看首页。  为了能使pod调度到master节点,用命令删除污点。在浏览器访问dashboard（<https://IP:30001>）  在node节点执行k8s\_node\_install.sh，将该节点加入kubernetes集群。完成后在master节点上查看所有节点状态。  **任务 3 EFK 日志平台构建**  ①导入镜像  将提供的 efk-img.tar.gz 压缩包中的镜像导入到 master 节点，并使用命令将镜像上传至 haboor 镜像仓库中。  ②NFS 配置  在 master 节点、node1 节点、node2 节点分别安装 nfs 服务， master 节点作为服务端， 把/data/volume1 目录作为共享目录，只允许 192.168.10 网段访问。    ③RBAC 配置  在 master 节点，编写 sa.yaml，创建名称为 nfs-provisioner 的 SA 账号。  ④RBAC 配置  编写 rbac.yaml ，对创建的 sa 账号进行 RBAC 授权，基于 yaml 文件创建完成后使用命令分别查看 sa 账号和 rbac 授权信息。  ⑤StorageClass 动态绑定  编写 nfs-deploy.yaml 文件，基于 nfs-client-provisioner 镜像创建 nfs-provisioner 的  deployment 对象，绑定 nfs 服务端的共享目录。  ⑥StorageClass 动态绑定  编写 storageclass.yaml 文件，创建 storageclass 动态绑定 nfs-provisioner，完成后查看 nfs-provisioner 的 pod 及 storageclasses 对象。  ⑦ 通过 statefulset 创建 elasticsearch 集群  编写 es-statefulset.yaml，通过 yaml 文件构建 elasticsearch 的 statefulset 集群，集群中有3 个副本名字分别为 es-cluster-0、es-cluster-1、es-cluster-2，并且使用上述 storageclass 提供 的存储，使用 elasticsearch:7.2.0 镜像，并且声明 9200 端口为 api 端口，9300 端口为内部访 问 端 口 ， 并 且 添 加 busybox 的 初 始 化 容 器 对 elasticsearch的数据目录 /usr/share/elasticsearch/data 进行授权操作。    ⑧. 创建 headless service  编写 es-svc.yaml 文件，为 elasticsearch 的 pod 创建一个 headless service，并在 service 中声明 9200 和 9300 端口。  ⑨. Kibana 可视化 UI 界面部署  编写 kibana.yaml，通过该文件创建 deployment 和 service，其中 deployment 基于 kibana:7.2.0 镜像创建并通过环境变量 ELASTICSEARCH\_URL 指定 elasticsearch 服务地址；  service 代理 kibana 的 pod 服务，并且使用 NodePort 类型。创建成功后在浏览器访问 Kibana 的 UI 界面。  ⑩. Fluentd 组件部署  编写 fluentd.yaml，通过 yaml 文件创建 DaemonSet 控制器部署 fluentd 服务，并在该文件中同时编写相关的 sa 账号和 rbac 内容，创建成功后保证可以正确采集容器内的日志。 |
| C模块：企业级应用的自动化部署和运维 **任务1企业级应用的自动化部署**  **1.ansible自动化运维工具的安装**  请使用提供的软件包在monitor节点安装ansible，安装完成后使用ansible --version命令验证是否安装成功。为所有节点添加test用户，设置用户密码为000000，为test用户设置免密sudo配置ssh免密登 录，使monitor节点能够免密登录所有节点的test用户。  **2.ansible 自动化运维工具的初始化**  创建 /root/ansible 目录作为工作目录，在该目录内创建 ansible.cfg 文件并完成以下配置，清单文件位  置为 /root/ansible/inventory，登录用户为 test，登录时不需要输入密码。设置并行主机数量为 2，允 许 test 用户免密提权到 root.将 ansible.cfg 文件内容粘贴到答题框。   1. 主机清单的编写。   编写主机清单文件，创建monitor用户组,monitor用户组内添加monitor主机，创建slave用户组, slave 组内添加slave1和slave2主机，主机名不得使用IP地址。   1. **使用自动化工具对 master 节点进行初始化**   请编写 prometheus.yml 控制 monitor 主机组，使用对应模块将 SELinux 临时状态和开机启动状态也 设置为 disabled。请使用 ansible 对应模块安装时间同步服务，使用文本编辑模块将该务的作用域设 置为 0.0.0.0/0，并设置状态为启动和开机自动启动。首先将提供的prometheus-2.37.0.linuxamd64.tar.gz 使用文件拷贝模块将该压缩包拷贝到目标主机/usr/local/ 下，使用 shell 模块解压该压缩包。  5.使用自动化运维工具完成企业级应用的部署。  编写prometheus.yml.j2模板文件，将所有slave节点信息添加到该文件中，但是被管节点的主机名信息 必须使用变量IP地址可以手动输入。完成后请创建node\_exporter.yml文件，编写第一play，将该play 命名为slave，该play控制的主机组为slave，使用ansible模块node\_exporter-1.3.1.linuxamd64.tar.gz发送到slave主机组的/usr/local/下，使用一个shel模块解压该压缩包，并启动该服务。随后编写第二个play，将第二个play命名为monitor，第二个play控制monitor节点，首先使用ansible模块将prometheus.yml.j2文件传输到monitor节点，然后使用script模块将prometheus启动。使用对应模块将grafana-8.1.2-1.x86\_64.rpm包发送到被控节点的/mnt/目录下，然后使用对应模块将该软件包安装，安装完成后设置grafana服务启动并设置开机自动启动。使用浏览器登录prometheus查看prometheus是否成功监控所有slave节点。  **任务 2 企业级应用的运维**  **1.使用 prometheus 监控 mysqld 服务**  将提供的mysqld\_exporter-0.14.0.linux-amd64.tar.gz 发送到agent虚拟机/usr/local/目录下解压并安装 mariadb服务。进入mariadb数据库中创建mysqld\_monitor用户并授权，然后创建mariadb配置文件，内容为数据库用户名密码。启动mysqld\_exporter组件确保9104端口启动。回到prometheus节点修改prometheus.yml文件并添加mysql被监控信息。重启prometheus，随后web界面刷新并查看mysqld被监控信息。  **⒉.安装alertmanager报警组件**  将提供的alertmanager-0.21.0.linux-amd64.tar.gz上传到prometheus节点/usr/local/目录下并解压，创建软连接alertmanager-0.23.0.linux-amd64/alertmanager。创建service启动文件名为alertmanager.service，然后启动alertmanager查看9093端口。在prometheus.yml配置文件中添加alertmanager信息并重新启动prometheus服务，在agent上停止node\_exporter服务。到web界面中查看警报管理器状态是否正常和agent状态是否异常。  **3.alertmanager告警邮件文件编写**  Prometheus虚拟机/usr/local/akertmanager/中存在着一个alertmanager.yml文件，请根据提供的地址和模板编写告警所发送到的email邮箱地址信息。 |