

2021 年全国职业院校技能大赛（高职组）

“云计算”试题库

第一场次题目：OpenStack 平台部署与运维（1）

某企业拟使用 OpenStack 搭建一个企业云平台，以实现资源池化弹性管理、企业应用集中管理、统一安全认证和授权等管理。

系统架构如图 1 所示，IP 地址规划如表 1 所示。

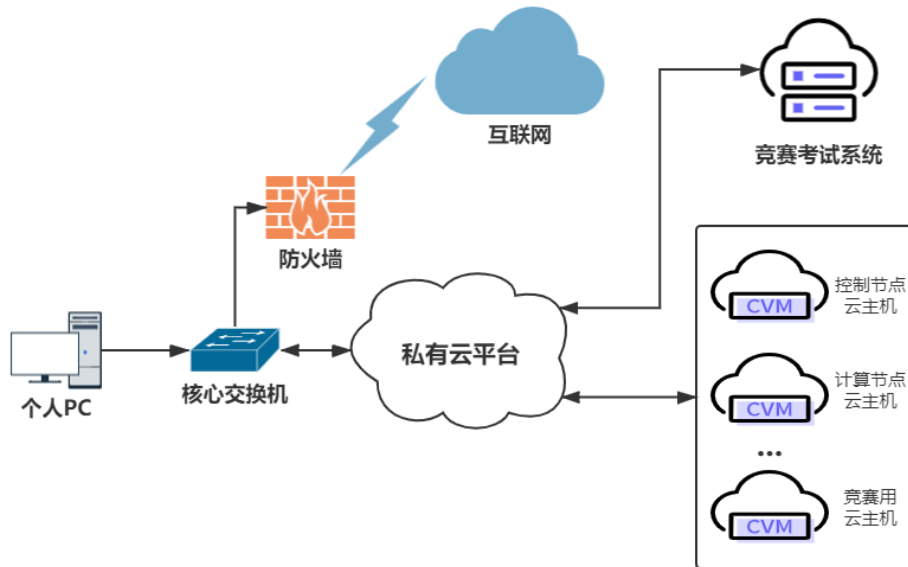


图 1 系统架构图

表 1 IP 地址规划

设备名称	主机名	接口	IP 地址	说明
云服务器1	Controller	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器2	Compute	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器3 ... 云服务器 n	自定义	eth0	172.129.x.0/24	
PC-1		本地连接	172.24.16.0/24	PC 使用

说明：

- 1.竞赛使用集群模式进行，比赛时给每个参赛队提供独立的租户与用户，各用户的资源配额相同，选手通过用户名与密码登录竞赛用私有云平台，创建云主机进行相应答题；
- 2.表中的 x 为工位号；在进行 OpenStack 搭建时的第二块网卡地址根据题意自行创建；
- 3.根据图表给出的信息，检查硬件连线及网络设备配置，确保网络连接正常；
- 4.考试所需要的账号资源、竞赛资源包与附件均会在考位信息表与设备确认单中给出；
- 5.竞赛过程中，为确保服务器的安全，请自行修改服务器密码；在考试系统提交信息时，请确认自己的 IP 地址，用户名和密码。

任务 1 基础运维任务（5 分）

1. 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；

完成后提交控制节点的用户名、密码和 IP 地址到答题框。

2. 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。完成后提交控制节点的用户名、密码和 IP 地址到答题框。

3. 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。完成后提交控制节点的用户名、密码和 IP 地址到答题框。

4. 在 compute 节点上利用空白分区划分 2 个 20G 分区。完成后提交计算节点的用户名、密码和 IP 地址到答题框。

任务 2 OpenStack 搭建与运维任务（10 分）

任务需要完成 openstack 平台的安装搭建及运维任务（无安装脚本），表 2 提供了在安装过程中需要使用相关服务参数。

表 2 密码

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	eth1 (外网卡名, 以实际为准)

1. 使用提供的脚本框架 `iaas-pre-host.sh` 和 `openrc.sh` 环境变量文件, 填充脚本, 在 `controller` 和 `compute` 节点上分别安装 `openstack` 平台的基础组件并完成相关配置。

2. 使用提供的脚本框架 `iaas-install-mysql.sh`, 填充脚本, 在 `controller` 节点上安装 `mariadb`、`mencached`、`rabbitmq` 等服务并完成相关配置。

3. 使用提供的脚本框架 `iaas-install-keystone.sh`, 填充脚本, 在 `controller` 节点上安装 `keystone` 服务并完成相关配置。

4. 使用提供的脚本框架 `iaas-install-glance.sh`, 填充脚本, 在 `controller` 节点上安装 `glance` 服务并完成相关配置。

5. 使用提供的脚本框架 `iaas-install-nova-controller.sh` 和 `iaas-install-nova-compute.sh`, 填充脚本, 在 `controller` 和 `compute` 节点上安装 `nova` 服务并完成配置。

6. 使用提供的脚本框架 `iaas-install-neutron-controller.sh` 和 `iaas-install-neutron-compute.sh`, 填充脚本, 在 `controller` 和 `compute` 节点上安装 `neutron` 服务并完成配置。

7. 使用提供的脚本框架 `iaas-install-dashboard.sh`, 填充脚本, 在 `controller` 节点上安装 `dashboard` 服务并完成相关配置。

8. 使用提供的脚本框架 `iaas-install-cinder-controller.sh` 和 `iaas-install-cinder-compute.sh`, 填充脚本, 在 `controller` 和 `compute` 节点上安装 `cinder` 服务并完成配置。

9. 使用提供的脚本框架 `iaas-install-swift-controller.sh` 和 `iaas-install-swift-compute.sh`, 填充脚本, 在 `controller` 和 `compute` 节点上安装 `swift` 服务并完成配置。。

10. 使用提供的脚本框架 `iaas-install-heat.sh`, 填充脚本, 在 `controller` 节点上安装 `heat` 服务并完成相关配置。

(注: 为了避免 OpenStack 不能成功搭建而影响后面竞赛进程, 任务 3-4 向选手提供 openstack 平台 (无 dashboard 组件), 供参赛选手进行竞赛)

任务 3 OpenStack 云平台 API 运维开发 (10 分)

1. 在 `controller` 节点的 `/root` 目录下编写 Python 程序 `create_user.py` 文件, 对接 `openstack api`, 创建一个用户, 描述为 “API create user!” (如果存在同名用户, 代码中需先进行删除操作)。

2. 在 `controller` 节点的 `/root` 目录下编写 Python 程序 `create_flavor.py` 文件, 对接 `openstack api`, 创建一个云主机类型 `pvm_flavor`: `vcpu` 为 2 个、内存为 2048M、硬盘大小 20G (如果存在同名主机类型, 代码中需先进行删除操作)。输出 `flavor` 名称、`id` 和详细信息。

3.在 controller 节点的/root 目录下按要求编写 Python 程序 create_network.py 文件，对接 openstack api，要求：（1）为平台创建内部网络 pvm_int,子网名称为 pvm_intsubnet;（2）设置云主机网络子网 IP 网段为 192.168.x.0/24（其中 x 是考位号），网关为 192.168.x.1（如果存在同名内网，代码中需先进行删除操作）。输出内网名称、id 和内网详细信息。

4.在 controller 节点的/root 目录下按要求编写 Python 程序 create_glance.py 文件，对接 openstack api，要求上传并使用镜像文件 cirros-0.3.4-x86_64-disk.img 创建 glance 镜像 pvm_image（如果存在同名镜像，代码中需先进行删除操作）。输出镜像名称、id 和详细信息。

5. 在 controller 节点的/root 目录下按要求编写 Python 程序 create_vm.py 文件，对接 openstack api，要求使用 pvm_image、pvm_flavor、pvm_intsubnet、pvm_router 组件创建 1 台主机 pvm1（如果存在同名虚拟主机，代码中需先进行删除操作）。输出虚拟主机的名称、ID 和详细信息。

任务 4 OpenStack 云平台运维开发（10 分，本任务只公布考试范围，不公布赛题）

此任务包含多个子任务，内容涉及编写 Python 脚本调用 OpenStack API 对 OpenStack 云平台进行管理和运维。使用自动化运维工具 Ansible 对云主机进行批量系统配置、批量程序部署、批量运行命令等运维操作。

第二场次题目：容器云平台部署与运维（1）

说明：完成本任务需要两台安装了 CentOS7.5 操作系统的云主机 master 和 node，镜像包中有本次容器云平台部署和运维所需的所有文件。

云梦公司技术部产品开发上线周期长，用户规模大且版本更新频繁。新版本的每次上线，产品都要承受极大的压力。引入 CICD (Continuous Integration 持续集成、Continuous Delivery 持续交付) 和灰度发布成了公司的当务之急，研发团队决定搭建基于 Kubernetes 的 CICD 环境，希望基于这个平台来实现 DevOps 的部分流程，并基于 Kubernetes 实现业务系统的灰度发布。

为了能够让云梦公司开发的 web 应用系统产品能够缩减开发周期，快速迭代版本，并实现业务从老版本到新版本的平滑过渡，避免升级过程中出现的问题对用户造成的影响。研发部决定使用微服务架构，实现基于 Kubernetes 的容器化部署、CICD 和灰度发布。

节点规划如表 1 所示。

表 1 容器云平台节点规划

节点角色	主机名	VCPUS	内存	硬盘
Master、Harbor、CICD	master	8	12G	100G
Worker Node	node	4	8G	100G

任务 1 Docker CE 及私有仓库安装任务（5 分）

1. 在 master、node 各节点中分别安装 DockerCE 和 docker-compose。完成后提交 master 节点的用户名、密码和 IP 到答题框。

2. 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 master 节点的用户名、密码和 IP 到答题框。

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

- （1）容器 1 名称：owncloud；镜像：owncloud: latest；
- （2）挂载路径：/data/db/owncloud:/var/www/html/data；
- （3）owncloud 端口映射：5679:80；
- （4）容器 2 名称 owncloud-db；镜像：mysql: 5.6；

(5) 数据库密码：123456。

完成后编排部署 ownCloud，并提交 master 节点的用户名、密码和 IP 到答题框。

任务 2 基于容器的 web 应用系统部署任务（10 分）

云梦公司开发了一套基于 SpringBoot+MyBatis 微服务架构的 ChinaSkillsMall 电商系统，并实现全容器化部署，ChinaSkillsMall 电商应用系统架构如图 1 所示，系统说明如表 2 所示。

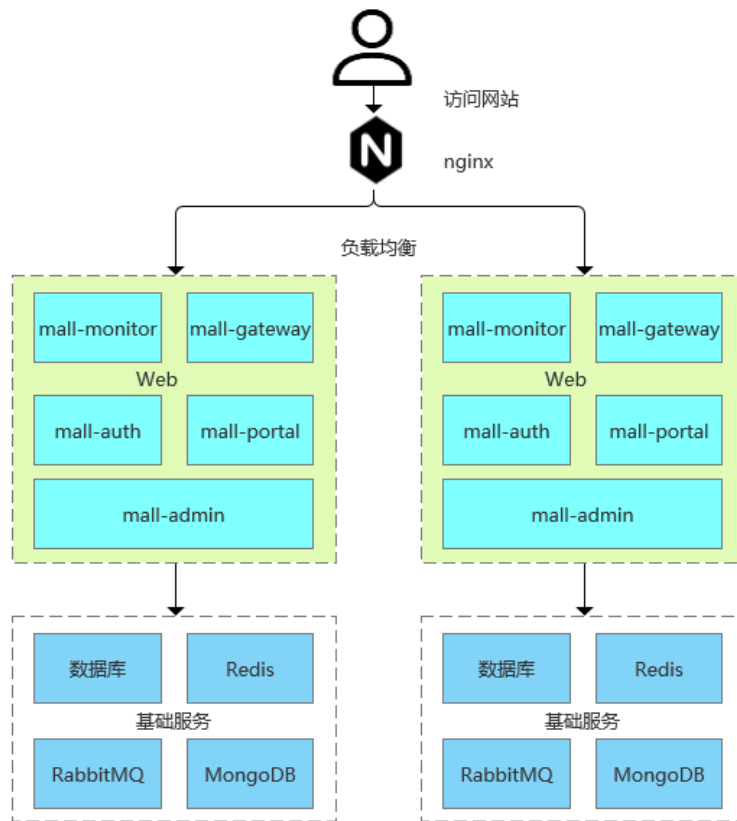


图 1 ChinaSkillsMall 电商应用系统架构图

表 2 ChinaSkillsMall 电商应用系统说明

模块	说明
mall-monitor	监控中心
mall-gateway	微服务网关
mall-auth	认证中心
mall-admin	商城后台服务

mall-portal	商城前台服务
-------------	--------

将 MariaDB 数据库组件、Redis 消息组件、RabbitMQ 消息组件、Nacos-Registry 注册中心服务组件和前端 Nginx 组件按照要求进行容器化。

1. 在 master 节点上编写 /root/mall-swarm/Dockerfile-mariadb 文件构建 chinaskillmall-mariadb:v1.1 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装并初始化 mariadb，密码：root;
- (4) 设置数据库编码为 UTF-8;
- (5) 开放端口：3306
- (6) 设置 mariadb 开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写 /root/mall-swarm/Dockerfile-redis 文件构建 chinaskillmall-redis:v1.1 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装 redis 服务;
- (4) 修改配置文件中的 bind 127.0.0.1 为 bind 0.0.0.0;
- (5) 设置 redis 免密，并关闭保护模式;
- (6) 开放端口：6379;
- (7) 设置服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

3. 在 master 节点上编写 /root/mall-swarm/Dockerfile-nacos 文件构建 chinaskillmall-nacos:v1.1 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装 jdk 工具和 nacos-server 工具;
- (4) 设置 nacos 服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

4. 在 master 节点上安装 node 工具，对 mall-admin-web 项目进行打包，然后编写 /root/mall-swarm/Dockerfile-nginx 文件构建 chinaskillmall-nginx:v1.1 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装 nginx 服务;
- (4) 将 mall-admin-web 项目打包后的文件拷贝到/usr/share/nginx/html 目录下;
- (5) 设置 nginx 服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

5. 在 master 节点上编写/root/mall-swarm/docker-compose.yaml 文件，具体要求为

- (1) 容器 1 名称：chinaskillmall-admin; 镜像：mall/mall-admin:1.0-SNAPSHOT; 端口映射：8080:8080;
- (2) 容器 2 名称： chinaskillmall-auth; 镜像： mall/mall-auth:1.0-SNAPSHOT; 端口映射：8401:8401;
- (3) 容器 3 名称： chinaskillmall-gateway; 镜像： mall/mall-gateway:1.0-SNAPSHOT; 端口映射：8201:8201;
- (4) 容器 4 名称： chinaskillmall-mysql; 镜像： chinaskillmall-mariadb:v1.1; 端口映射：3306:3306;
- (5) 容器 5 名称： chinaskillmall-nacos; 镜像： chinaskillmall-nacos:v1.1; 端口映射：8848:8848;
- (6) 容器 6 名称： chinaskillmall-nginx; 镜像： chinaskillmall-nginx:v1.1; 端口映射：8888:80;
- (7) 容器 7 名称： chinaskillmall-redis; 镜像： chinaskillmall-redis:v1.1; 端口映射：6379:6379。

完成后编排部署 ChinaSkillsMall 商城。然后提交 master 节点的用户名、密码和 IP 到答题框。

任务 3 基于容器的持续集成部署任务（10 分）

该公司决定采用 GitLab +Jenkins 来构建 CICD 环境，以缩短新功能开发上线周期，及时

满足客户的需求，实现 DevOps 的部分流程，来减轻部署运维的负担，实现可视化容器生命周期管理、应用发布和版本迭代更新，请完成 GitLab + Jenkins + Kubernetes 的 CI/CD 环境部署。CI/CD 应用系统架构如图 2 所示。

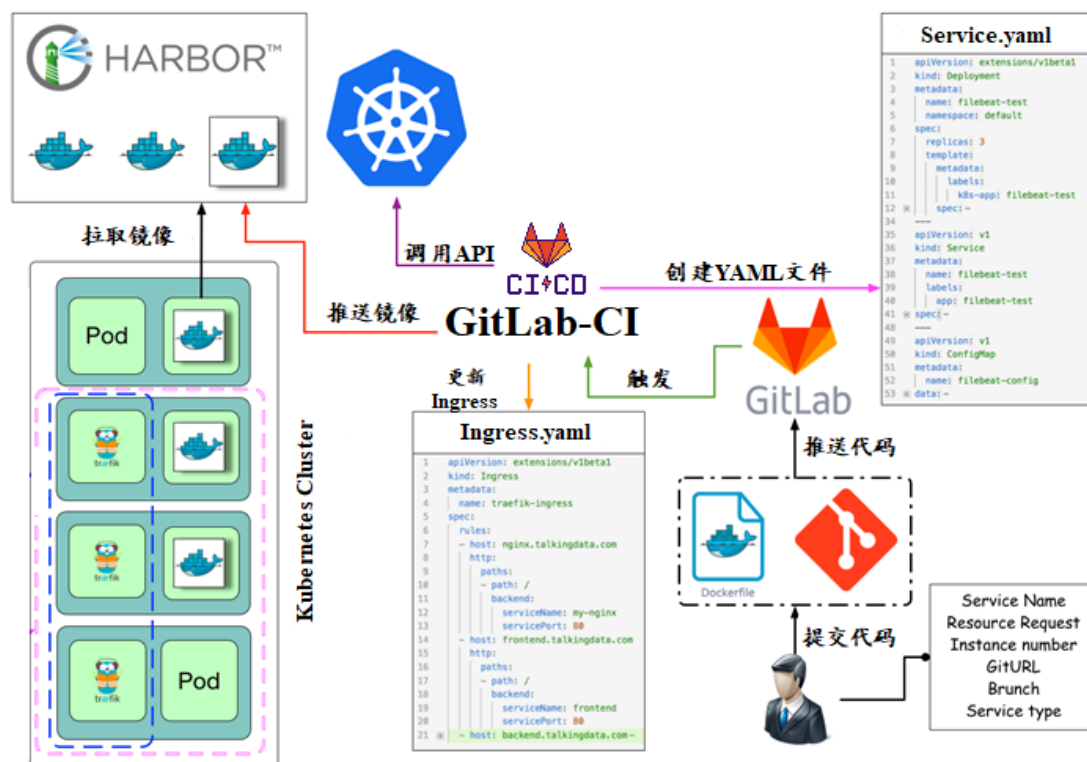


图 2 CI/CD 应用系统架构图

1. 在 master 节点上编写/root/jenkins/docker-compose.yaml 文件编排部署 Jenkins 服务，具体要求为

- (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: springcloud; 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 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写/root/gitlab/docker-compose.yaml 文件编排部署 Gitlab 服务，具体要求为

- (1) 容器名称: gitlab;
- (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 Springcloud, 将/opt/Springcloud 中的代码上传到 Springcloud 项目中。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

3. 配置 Jenkins 连接 Gitlab, 具体要求为

- (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 master 节点的用户名、密码和 IP 到答题框。

4.配置 Jenkins 连接 maven, 具体要求为

- (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 master 节点的用户名、密码和 IP 到答题框。

5.配置 CI/CD, 具体要求为

- (1) 新建一个流水线任务 Springcloud;
- (2) 编写流水线脚本, 构建 Springcloud 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 springcloud 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 springcloud。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

6. 将 Springcloud 项目中的代码上传到 Gitlab 中触发构建, 完成后提交 master 节点的用户名、密码和 IP 到答题框。

任务 4 Kubernetes 容器云平台部署与运维（10 分，本任务只公布考试范围，不公布赛题）

云梦公司为应对购物节活动，对电商网站进行活动模块开发，公司决定采用基于 Istio 的灰度发布（又名金丝雀发布）来实现业务从老版本到新版本的平滑过渡，并避免升级过程中出现的问题对用户造成的影响。

请根据要求，完成 Kubernetes 的集群运维、调度、网络、存储、安全、应用生命周期管理、日志/监控等运维任务。

第三场次题目：公有云部署与运维（1）

云梦是一家科技公司，现在公司开发全面转向云原生，因此需要对公司内部项目搭建基于云原生的 Devops 相关服务。

某公有云是一家成熟的公有云提供商，为客户提供 K8S、云网络、云硬盘、云防火墙、负载均衡等服务，可创建 Web 服务，共享文件存储服务，数据库服务，数据库集群等服务。

根据上述公有云平台的特性，完成公有云中的各项运维工作。

任务 1 基础设施构建（10 分）

1.登录华为云平台，创建两台云实例，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) CPU 架构：鲲鹏计算；
- (4) 规格：kc1.xlarge.2；
- (5) 镜像：CentOS 7.5 64 位；
- (6) 系统盘：高 IO 50G 硬盘；
- (7) 公网带宽：按带宽计费，5Mbps；
- (8) 实例名称：ChinaSkill-node-1、ChinaSkill-node-2；
- (9) 登录方式：使用密码登录，密码自定义。

创建完成后，提交当前的 AK（SecretId）、SK（SecretKey）和 ChinaSkill-node-1 的实例 ID 到答题框。

2.在公有云中完成虚拟私有云的创建，具体要求为

- (1) 在贵阳区域进行创建操作；
- (2) 创建一个名为 intnetX 的内部网络：IP 地址为 172.16.1.0/16；
- (3) 创建子网名称为 intnetX-server：IP 地址为 172.16.1.0/24；
- (4) 创建子网名称为 intnetX-mysql：IP 地址为 172.16.2.0/24；

完成创建后，提交当前的 AK（SecretId）、SK（SecretKey）和 intnetX 网络的 ID 到答题框。

3.使用创建的 ChinaSkill-node-1 和 ChinaSkill-node-2 进行切换内网操作，将这两个云主机的内网，切换到 intnetX 网络的 intnetX-server 子网。切换完成后，保证该云主机还能够正常上网。操作完成后，提交 ChinaSkill-node-1 节点的用户名、密码和公网 IP 地址到答题框。

4.创建一台云数据库，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) 数据库版本：MySQL 5.7；
- (4) 实例类型：主备；
- (5) 实例规格：鲲鹏 2 核 8GB/40GB；
- (6) 虚拟私有云：intnetX-mysql；
- (7) 字符集：UTF8；
- (8) 端口：3306；
- (9) 数据库密码：Root123456；
- (10) 数据库名称：chinaskill-mysql；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和云数据库的实例 ID 到答题框。

5.创建一个安全组，具体要求为

- (1) 名称：intnetX-security；
- (2) 允许策略：只放行源地址为 172.16.1.0/24 访问 3306 端口；
- (3) 允许策略：只放行源地址为 172.16.1.0/24 使用 ping 命令访问；
- (4) 关联实例：将 intnetX-security 安全组关联至所创建的数据库中；

6.使用提供的软件包，在 ChinaSkill-node-1 和 ChinaSkill-node-2 中部署 WordPress 的 Web 服务即 LNMP+WordPress。使用云数据库 chinaskill-mysql 为后端数据库。完成 WordPress 系统上云后，将 ChinaSkill-node-1 节点的 WordPress 访问地址提交到答题框。

7.创建一个负载均衡器 chinaskill-elb，将 ChinaSkill-node-1 和 ChinaSkill-node-2 加入负载均衡的后端。设置一个可用的公网服务 IP，服务端口为 80。配置监听器，监听 80 端口。对浮动公共 IP 进行 Web 访问测试。完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和负载均衡器的 ID 到答题框。

8.创建一个云服务器备份存储库名为 server_backup，容量为 100G。完成后，将当前的 AK (SecretId)、SK (SecretKey) 和存储库 ID 到答题框。

9.创建一个公网 NAT 网关，具体配置为

- (1) 名称为：kcloud-nat；
- (2) 虚拟私有云：intnetX；

- (3) 子网: `intnetX-server`;
- (4) 规则: 内部子网地址访问外网;
- (5) 保证使用 `intnetX-server` 云服务器可以使用 `kcloud-nat` 网关访问外网。

创建完成后提交当前的 AK (`SecretId`)、SK (`SecretKey`) 和 NAT 网关的 ID 到答题框。

任务 2 系统管理与维护 (10 分)

1.给云数据库 `chinaskill-mysql` 创建只读数据库 `chinaskill-read` 实例。使用 2vCPU、4GB 内存规格。并给 `chinaskill-mysql` 申请读写分离地址。完成后, 提交当前的 AK (`SecretId`)、SK (`SecretKey`) 和只读数据库实例 ID 到答题框。

2.登录云服务器 `ChinaSkill-node-1` 和 `ChinaSkill-node-2` 中, 修改数据库连接地址为云数据库读写地址。将 `ChinaSkill-node-1` 云服务器制作镜像文件 `chinaskill-image`。完成后, 提交当前的 AK (`SecretId`)、SK (`SecretKey`) 和镜像 ID 到答题框。

3.新建一个启动配置, 具体要求为

- (1) 启动配置名称: `template-exam`;
- (2) 计费模式: 按量计费;
- (3) 地域: 上海;
- (4) CPU 架构: 鲲鹏计算;
- (5) 规格: `kc1.xlarge.2`;
- (6) 镜像: `chinaskill-image`;
- (7) 系统盘: 高 IO 50G 硬盘;
- (8) 登录方式: 使用密码登录, 密码自定义。

创建一个伸缩组, 具体要求为

- (1) 名称: `as-exam`;
- (2) 最小伸缩数: 1; 起始实例数: 1; 最大伸缩数: 5;
- (3) 启动配置: `template-exam`;
- (4) 使用负载均衡: `chinaskill-elb`;
- (5) 移出策略: 移出最旧的实例。

为伸缩组 `as-exam` 新建告警触发策略, 具体要求为

- (1) 如果实例的内存利用率在 5 分钟内的最大值小于 40%, 且连续发生 3 次。则实例

数减少 1 台。冷却 60 秒；

(2) 如果实例的内存利用率在 5 分钟内的最大值大于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒；

(3) 如果实例的 CPU 利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例数减少 1 台。冷却 60 秒；

(4) 如果实例的 CPU 利用率在 5 分钟内的平均值大于等于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒。

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和伸缩组的实例 ID 到答题框。

4.使用公有云平台，申请对象存储服务，存储桶名字自定义。自行编写一个静态网页，网页的内容为“welcome to China”，使用申请的对象存储服务，托管该静态网页。托管成功后，提交你对象存储的访问域名到答题框。

5.创建一个鲲鹏架构集群，具体要求为

- (1) 集群名称: kcloud;
- (2) 集群版本: v1.15.11;
- (3) 地域: 贵阳;
- (4) 集群管理规模: 50 节点;
- (5) 控制节点数: 3;
- (6) 节点使用子网: intnetX-server;
- (7) Pod 实例上限: 64;
- (8) 容器网段: 10.10.0.0/16。

节点配置信息要求为

- (1) 节点名称: kcloud-server;
- (2) 节点规格: kc.xlarge.2
- (3) 节点: EulerOS 2.8

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和集群的 ID 到答题框。

6.使用插件管理在 kcloud 容器集群中安装 dashboard 可视化监控界面。完成后，提交 dashboard 访问地址到答题框。

7.在 kcloud 集群中安装 kubectl 命令，使用 kubectl 命令管理 kcloud 集群。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

8.在 kcloud 集群中定义名为 liveness-http 探针和 liveness-tcp 探针，监控 kcloud 集群。完

成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

9.使用容器镜像服务作为镜像仓库，将提供的 httpd-2.4.33 和 httpd-2.4.35 镜像上传至服务中。提交当前的 AK（SecretId）、SK（SecretKey）和组织名称到答题框。

10.在 kcloud 集群中创建一个 Deployment，使用所上传的 httpd-2.4.33 容器镜像，配置使其可以通过外部公网进行访问。完成后提交外部访问地址到答题框。

任务 3 自动化运维（10 分，本任务只公布考试范围，不公布赛题）

编写 Python 脚本调用公有云 API 完成公有云运维工作。

第一场次题目：OpenStack 平台部署与运维（2）

某企业拟使用 OpenStack 搭建一个企业云平台，以实现资源池化弹性管理、企业应用集中管理、统一安全认证和授权等管理。

系统架构如图 1 所示，IP 地址规划如表 1 所示。

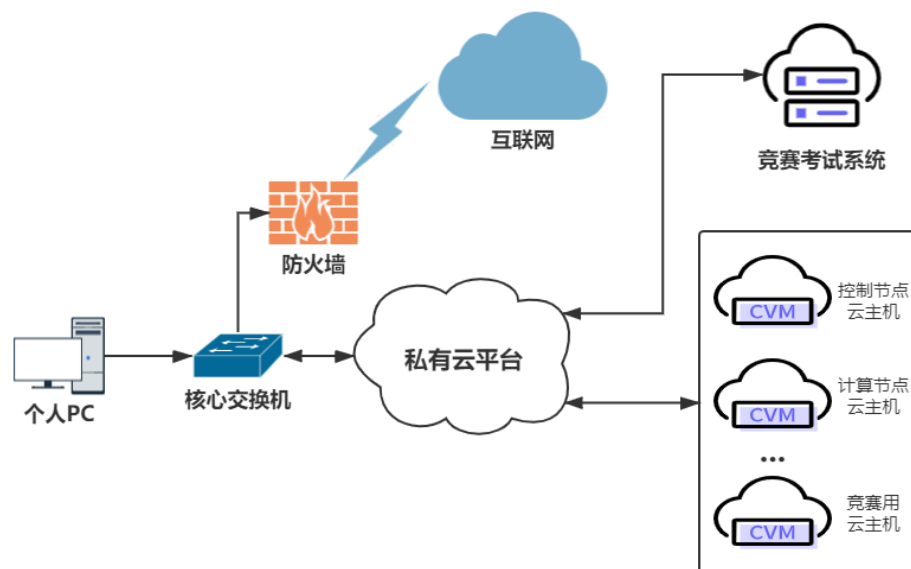


图 1 系统架构图

表 1 IP 地址规划

设备名称	主机名	接口	IP 地址	说明
云服务器1	Controller	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器2	Compute	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器3 ... 云服务器 n	自定义	eth0	172.129.x.0/24	
PC-1		本地连接	172.24.16.0/24	PC 使用

说明：

1.竞赛使用集群模式进行，比赛时给每个参赛队提供独立的租户与用户，各用户的资源配额相同，选手通过用户名与密码登录竞赛用私有云平台，创建云主机进行相应答题；

2.表中的 x 为工位号；在进行 OpenStack 搭建时的第二块网卡地址根据题意自行创建；

3.根据图表给出的信息，检查硬件连线及网络设备配置，确保网络连接正常；

4.考试所需要的账号资源、竞赛资源包与附件均会在考位信息表与设备确认单中给出；

5.竞赛过程中，为确保服务器的安全，请自行修改服务器密码；在考试系统提交信息时，请确认自己的 IP 地址，用户名和密码。

任务 1 基础运维任务（5 分）

1. 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；

完成后提交控制节点的用户名、密码和 IP 地址到答题框。

2. 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。完成后提交控制节点的用户名、密码和 IP 地址到答题框。

3. 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。完成后提交控制节点的用户名、密码和 IP 地址到答题框。

4. 在 compute 节点上利用空白分区划分 2 个 20G 分区。完成后提交计算节点的用户名、密码和 IP 地址到答题框。

任务 2 OpenStack 搭建与运维任务（10 分）

本任务需要完成 openstack 平台的安装搭建及运维任务（无安装脚本），表 2 提供了在安装过程中需要使用相关服务参数。

表 2 密码

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	eth1 (外网卡名, 以实际为准)

1. 使用提供的脚本框架 `iaas-pre-host.sh` 和 `openrc.sh` 环境变量文件, 填充脚本, 在 `controller` 和 `compute` 节点上分别安装 `openstack` 平台的基础组件并完成相关配置。

2. 使用提供的脚本框架 `iaas-install-mysql.sh`, 填充脚本, 在 `controller` 节点上安装 `mariadb`、`mencached`、`rabbitmq` 等服务并完成相关配置。

3. 使用提供的脚本框架 `iaas-install-keystone.sh`, 填充脚本, 在 `controller` 节点上安装 `keystone` 服务并完成相关配置。

4. 使用提供的脚本框架 `iaas-install-glance.sh`, 填充脚本, 在 `controller` 节点上安装 `glance` 服务并完成相关配置。

5. 使用提供的脚本框架 `iaas-install-nova-controller.sh` 和 `iaas-install-nova-compute.sh`, 填充脚本, 在 `controller` 和 `compute` 节点上安装 `nova` 服务并完成配置。

6. 使用提供的脚本框架 `iaas-install-neutron-controller.sh` 和 `iaas-install-neutron-compute.sh`, 填充脚本, 在 `controller` 和 `compute` 节点上安装 `neutron` 服务并完成配置。

7. 使用提供的脚本框架 `iaas-install-dashboard.sh`, 填充脚本, 在 `controller` 节点上安装 `dashboard` 服务并完成相关配置。

8. 使用提供的脚本框架 `iaas-install-cinder-controller.sh` 和 `iaas-install-cinder-compute.sh`, 填充脚本, 在 `controller` 和 `compute` 节点上安装 `cinder` 服务并完成配置。

9. 使用提供的脚本框架 `iaas-install-swift-controller.sh` 和 `iaas-install-swift-compute.sh`, 填充脚本, 在 `controller` 和 `compute` 节点上安装 `swift` 服务并完成配置。。

10. 使用提供的脚本框架 `iaas-install-heat.sh`, 填充脚本, 在 `controller` 节点上安装 `heat` 服务并完成相关配置。

(注: 为了避免 OpenStack 不能成功搭建而影响后面竞赛进程, 任务 3-4 向选手提供 openstack 平台 (无 dashboard 组件), 供参赛选手进行竞赛)

任务 3 OpenStack 云平台 API 运维开发 (10 分)

1. 在 `controller` 节点的 `/root` 目录下按要求编写 Python 程序 `create_sec.py` 文件, 对接 `openstack api`, 要求在云平台上创建一个安全组 `pvm_sec`, 开放 20、21、22、80、3306 端口 (如存在同名安全组, 代码中需先进行删除操作)。输出安全组名称、`id` 和详细信息。

2. 在 `controller` 节点的 `/root` 目录下按要求编写 Python 程序 `create_vm.py` 文件, 对接 `openstack api`, 要求使用 `pvm_image`、`pvm_flavor`、`pvm_intsubnet`、`pvm_router`、和 `pvm_sec` 创建 1 台主机 `pvm1` (如果存在同名虚拟机, 代码中需先进行删除操作)。输出虚拟机的名称、`ID` 和详细信息。

3. 在 controller 节点的/root 目录下按要求编写 Python 程序 create_ip.py 文件，对接 openstack api，要求为云主机 pvm1 绑定浮动 IP。输出云主机名称、id、浮动 IP 和详细信息。

4. 在 controller 节点的/root 目录下按要求编写 Python 程序 create_cinder.py 文件，对接 openstack api，要求调用 cinder 相关接口创建 cinder 卷，卷名称为 pvm_cdisk、大小为 10G、类型为 LVM（如果存在同名卷，代码中需先进行删除操作）。输出 cinder 卷名称、id 和详细信息。

5. 在 controller 节点的/root 目录下按要求编写 Python 程序 attach_lvm.py 文件，对接 openstack api，要求将 pvm_cdisk 卷附加到云主机 pvm1。输出云主机名称、ID 和详细信息。

任务 4 OpenStack 云平台运维开发（10 分，本任务只公布考试范围，不公布赛题）

此任务包含多个子任务，内容涉及编写 Python 脚本调用 OpenStack API 对 OpenStack 云平台进行管理和运维。使用自动化运维工具 Ansible 对云主机进行批量系统配置、批量程序部署、批量运行命令等运维操作。

第二场次题目：容器云平台部署与运维（2）

说明：完成本任务需要两台安装了 CentOS7.5 操作系统的云主机 master 和 node，镜像包中有本次容器云平台部署和运维所需的所有文件。

云梦公司技术部产品开发上线周期长，用户规模大且版本更新频繁。新版本的每次上线，产品都要承受极大的压力。引入 CICD (Continuous Integration 持续集成、Continuous Delivery 持续交付) 和灰度发布成了公司的当务之急，研发团队决定搭建基于 Kubernetes 的 CICD 环境，希望基于这个平台来实现 DevOps 的部分流程，并基于 Kubernetes 实现业务系统的灰度发布。

为了能够让云梦公司开发的 web 应用系统产品能够缩减开发周期，快速迭代版本，并实现业务从老版本到新版本的平滑过渡，避免升级过程中出现的问题对用户造成的影响。研发部决定使用微服务架构，实现基于 Kubernetes 的容器化部署、CICD 和灰度发布。

节点规划如表 1 所示。

表 1 容器云平台节点规划

节点角色	主机名	VCPUS	内存	硬盘
Master、Harbor、CICD	master	8	12G	100G
Worker Node	node	4	8G	100G

任务 1 Docker CE 及私有仓库安装任务（5 分）

1. 在 master、node 各节点中分别安装 DockerCE 和 docker-compose。完成后提交 master 节点的用户名、密码和 IP 到答题框。

2. 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 master 节点的用户名、密码和 IP 到答题框。

3. 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 master 节点的用户名、密码和 IP 到答题框。

任务 2 基于容器的 web 应用系统部署任务（10 分）

云梦公司开发了一套基于 SpringBoot+MyBatis 微服务架构的 ChinaSkillsMall 电商系统，并实现全容器化部署，ChinaSkillsMall 电商应用系统架构如图 1 所示，系统说明如表 2 所示。

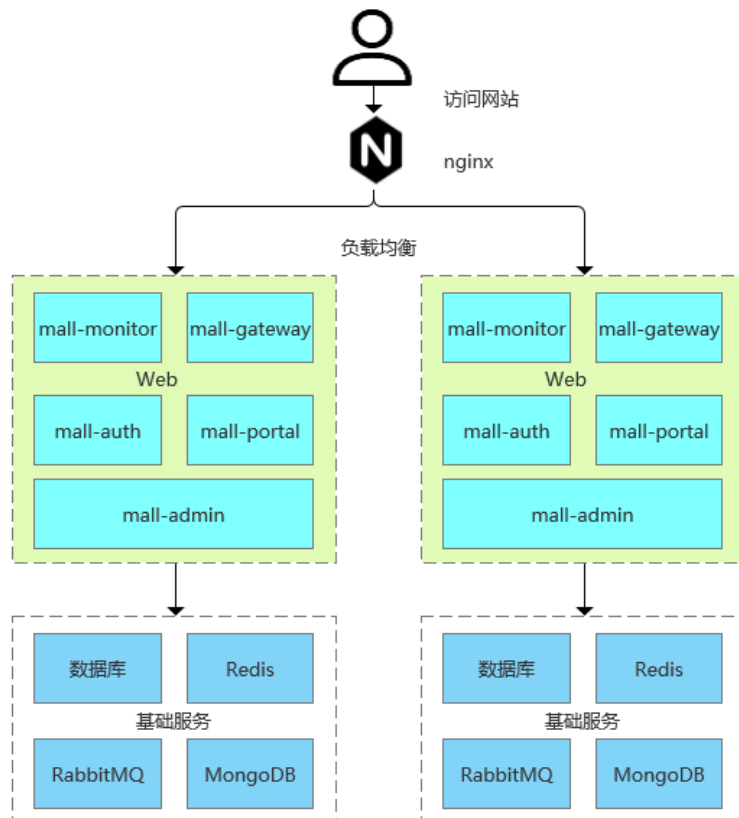


图 1 ChinaSkillsMall 电商应用系统架构图

表 2 ChinaSkillsMall 电商应用系统说明

模块	说明
mall-monitor	监控中心
mall-gateway	微服务网关
mall-auth	认证中心
mall-admin	商城后台服务
mall-portal	商城前台服务

将 MariaDB 数据库组件、Redis 消息组件、RabbitMQ 消息组件、Nacos-Registry 注册中心服务组件和前端 Nginx 组件按照要求进行容器化。

1. 在 master 节点上编写 /root/mall-swarm/Dockerfile-mariadb 文件构建 chinaskillmall-mariadb:v1.1 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装并初始化 mariadb，密码：root;
- (4) 设置数据库编码为 UTF-8;
- (5) 开放端口：3306
- (6) 设置 mariadb 开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写 /root/mall-swarm/Dockerfile-redis 文件构建 chinaskillmall-redis:v1.1 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装 redis 服务;
- (4) 修改配置文件中的 bind 127.0.0.1 为 bind 0.0.0.0;
- (5) 设置 redis 免密，并关闭保护模式;
- (6) 开放端口：6379;
- (7) 设置服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

3. 在 master 节点上编写 /root/mall-swarm/Dockerfile-nacos 文件构建 chinaskillmall-nacos:v1.1 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装 jdk 工具和 nacos-server 工具;
- (4) 设置 nacos 服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

4. 在 master 节点上安装 node 工具，对 mall-admin-web 项目进行打包，然后编写 /root/mall-swarm/Dockerfile-nginx 文件构建 chinaskillmall-nginx:v1.1 镜像，具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 nginx 服务;
- (4) 将 mall-admin-web 项目打包后的文件拷贝到/usr/share/nginx/html 目录下;
- (5) 设置 nginx 服务开机自启。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

5. 在 master 节点上编写/root/mall-swarm/docker-compose.yaml 文件, 具体要求为

- (1) 容器 1 名称: chinaskillmall-admin; 镜像: mall/mall-admin:1.0-SNAPSHOT; 端口映射: 8080:8080;
- (2) 容器 2 名称: chinaskillmall-auth; 镜像: mall/mall-auth:1.0-SNAPSHOT; 端口映射: 8401:8401;
- (3) 容器 3 名称: chinaskillmall-gateway; 镜像: mall/mall-gateway:1.0-SNAPSHOT; 端口映射: 8201:8201;
- (4) 容器 4 名称: chinaskillmall-mysql; 镜像: chinaskillmall-mariadb:v1.1; 端口映射: 3306:3306;
- (5) 容器 5 名称: chinaskillmall-nacos; 镜像: chinaskillmall-nacos:v1.1; 端口映射: 8848:8848;
- (6) 容器 6 名称: chinaskillmall-nginx; 镜像: chinaskillmall-nginx:v1.1; 端口映射: 8888:80;
- (7) 容器 7 名称: chinaskillmall-redis; 镜像: chinaskillmall-redis:v1.1; 端口映射: 6379:6379。

完成后编排部署 ChinaSkillsMall 商城。然后提交 master 节点的用户名、密码和 IP 到答题框。

任务 3 基于容器的持续集成部署任务 (10 分)

该公司决定采用 GitLab + Jenkins 来构建 CICD 环境, 以缩短新功能开发上线周期, 及时满足客户的需求, 实现 DevOps 的部分流程, 来减轻部署运维的负担, 实现可视化容器生命周期管理、应用发布和版本迭代更新, 请完成 GitLab + Jenkins + Kubernetes 的 CICD 环境部署。CICD 应用系统架构如图 2 所示。

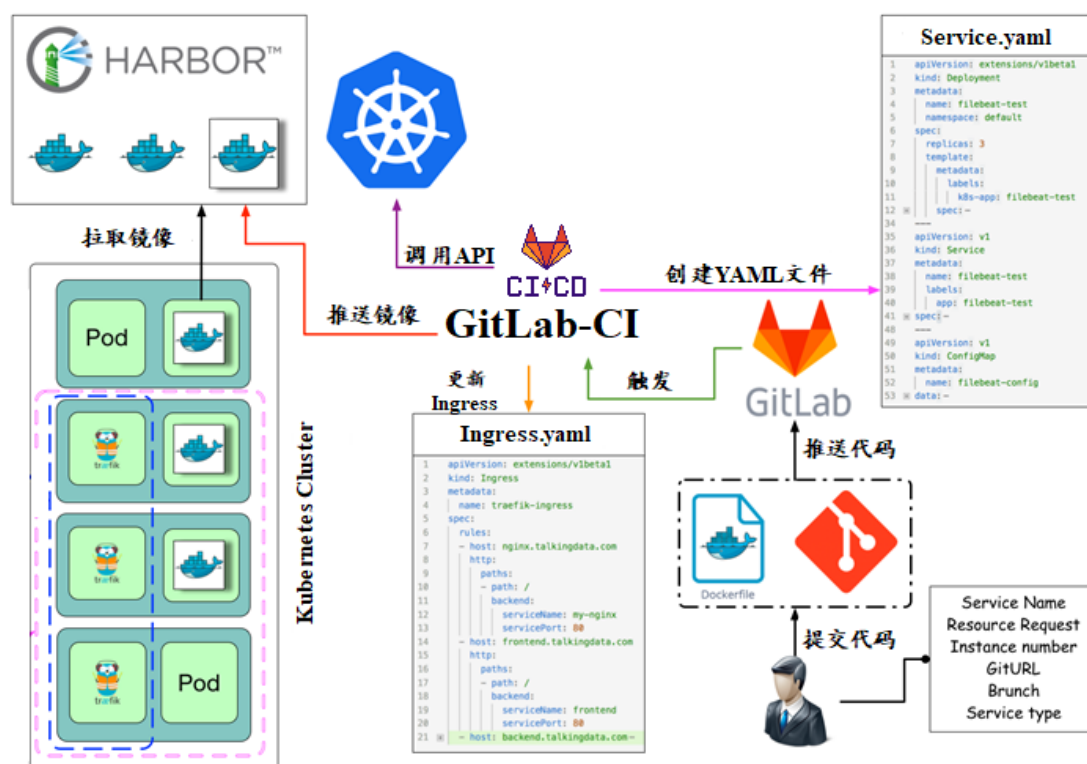


图 2 CI/CD 应用系统架构图

1. 在 master 节点上编写/root/jenkins/docker-compose.yaml 文件编排部署 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：springcloud；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 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写/root/gitlab/docker-compose.yaml 文件编排部署 Gitlab 服务，具体要求如下：

- (1) 容器名称：gitlab；
- (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 Springcloud, 将/opt/Springcloud 中的代码上传到 Springcloud 项目中。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

4. 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 master 节点的用户名、密码和 IP 到答题框。

4.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 master 节点的用户名、密码和 IP 到答题框。

5.配置 CI/CD, 具体要求如下:

- (1) 新建一个流水线任务 Springcloud;
- (2) 编写流水线脚本, 构建 Springcloud 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 springcloud 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 springcloud。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

7. 将 Springcloud 项目中的代码上传到 Gitlab 中触发构建, 完成后提交 master 节点的用户名、密码和 IP 到答题框。

任务 4 Kubernetes 容器云平台部署与运维 (10 分, 本任务只公布考试范围, 不公布赛题)

当应用上线以后, 运维面临的一大挑战是如何能够在不影响已上线业务的情况下进行升级。在无法百分百避免版本升级故障的情况下, 需要通过一种方式进行可控的版本发布,

把故障影响控制在可以接受的范围内，并可以快速回退。公司决定采用基于 Istio 的灰度发布（又名金丝雀发布）来实现业务从老版本到新版本的平滑过渡，并避免升级过程中出现的问题对用户造成的影响。

请根据要求，完成 Kubernetes 的集群运维、调度、网络、存储、安全、应用生命周期管理、日志/监控等运维任务。

第三场次题目：公有云部署与运维（2）

云梦是一家科技公司，现在公司开发全面转向云原生，因此需要对公司内部项目搭建基于云原生的 Devops 相关服务。

某公有云是一家成熟的公有云提供商，为客户提供 K8S、云网络、云硬盘、云防火墙、负载均衡等服务，可创建 Web 服务，共享文件存储服务，数据库服务，数据库集群等服务。

根据上述公有云平台的特性，完成公有云中的各项运维工作。

任务 1 基础设施构建（10 分）

1.在公有云中完成虚拟私有云的创建，具体要求为

- (1) 在贵阳区域进行创建操作；
- (2) 创建一个名为 intnetX 的虚拟私有云：IP 地址为 172.16.1.0/16；
- (3) 创建子网名称为 intnetX-server：IP 地址为 172.16.1.0/24；
- (4) 创建一个名为 net-data 的虚拟私有云：网段为 192.168.1.0/16
- (5) 创建子网名称为 net-mysql：IP 地址为 192.168.1.0/24；

完成创建后，提交当前的 AK（SecretId）、SK（SecretKey）和 intnetX 网络的 ID 到答题框。

2.创建对等连接 intnetX-net，配置 intnetX 虚拟私有云和 net-data 虚拟机私有云连通。操作完成后，提交当前的 AK（SecretId）、SK（SecretKey）和对等连接的 ID 到答题框。

3.登录华为云平台，创建两台云实例，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) CPU 架构：鲲鹏计算；
- (4) 规格：kc1.xlarge.2；
- (5) 镜像：CentOS 7.5 64 位；
- (6) 系统盘：高 IO 50G 硬盘；
- (7) 虚拟网络：intnetX-server；
- (8) 实例名称：ChinaSkill-node-1、ChinaSkill-node-2；
- (9) 登录方式：使用密码登录，密码自定义。

创建完成后，提交当前的 AK（SecretId）、SK（SecretKey）和 ChinaSkill-node-1 的实例 ID 到答题框。

4.购买云硬盘实例，使用超高 IO 磁盘类型，申请购买 100G 磁盘大小。创建完成后，提

交当前的 AK (SecretId)、SK (SecretKey) 和云硬盘 ID 到答题框。

5.创建一台云数据库，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) 数据库版本：MySQL 5.7；
- (4) 实例类型：主备；
- (5) 实例规格：鲲鹏 2 核 8GB/40GB；
- (6) 虚拟私有云：net-mysql；
- (7) 字符集：UTF8；
- (8) 端口：3306；
- (9) 数据库密码：Root123456；
- (10) 数据库名称：chinaskill-mysql；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和云数据库的实例 ID 到答题框。

6.给云数据库 chinaskill-mysql 创建只读数据库 chinaskill-read 实例。使用 2vCPU、4GB 内存规格。并给 chinaskill-mysql 申请读写分离地址。完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和只读数据库实例 ID 到答题框。

7.创建一个网络 ACL，具体要求为

- (1) 名称：intnetX-acl；
- (2) 允许策略：只放行源地址为 172.16.1.0/24 访问 3306 端口；
- (3) 允许策略：只放行源地址为 172.16.1.0/24 使用 ping 命令访问；
- (4) 关联子网：关联网络 ACL 至 net-mysql 子网中；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和网络 ACL ID 到答题框。

8.使用公有云平台，申请对象存储服务，存储桶名字自定义。自行编写一个静态网页，网页的内容为“welcome to China”，使用申请的对象存储服务，托管该静态网页。托管成功后，提交你对象存储的访问域名到答题框。

9.使用提供的软件包，在 ChinaSkill-node-1 和 ChinaSkill-node-2 中部署 WordPress 的 Web 服务即 LNMP+WordPress。使用云数据库 chinaskill-mysql 读写分离地址。完成 WordPress 系统上云后，将 ChinaSkill-node-1 节点的 WordPress 访问地址提交到答题框。

任务 2 系统管理与维护（10 分）

1. 创建一个负载均衡器 chinaskill-elb，将 ChinaSkill-node-1 和 ChinaSkill-node-2 加入负载均衡的后端。设置一个可用的公网服务 IP，服务端口为 80。配置监听器，监听 80 端口。对浮动公共 IP 进行 Web 访问测试。完成后，提交当前的 AK（SecretId）、SK（SecretKey）和负载均衡器的 ID 到答题框。

2. 创建一个云服务器备份存储库名为 server_backup，容量为 100G。将 ChinaSkill-node-1 云服务器制作镜像文件 chinaskill-image。完成后，将当前的 AK（SecretId）、SK（SecretKey）和存储库 ID 到答题框。

3. 新建一个启动配置，具体要求为

- （1）启动配置名称：template-exam；
- （2）计费模式：按量计费；
- （3）地域：贵阳；
- （4）CPU 架构：鲲鹏计算；
- （5）规格：kc1.xlarge.2；
- （6）镜像：chinaskill-image；
- （7）系统盘：高 IO 50G 硬盘；
- （8）登录方式：使用密码登录，密码自定义。

创建一个伸缩组，具体要求为

- （1）名称：as-exam；
- （2）最小伸缩数：1；起始实例数：1；最大伸缩数：5；
- （3）启动配置：template-exam；
- （4）使用负载均衡：chinaskill-elb；
- （5）移出策略：移出最旧的实例。

为伸缩组 as-exam 新建告警触发策略，具体要求为

（1）如果实例的内存利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例数减少 1 台。冷却 60 秒；

（2）如果实例的内存利用率在 5 分钟内的最大值大于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒；

（3）如果实例的 CPU 利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例

数减少 1 台。冷却 60 秒；

(4) 如果实例的 CPU 利用率在 5 分钟内的平均值大于等于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒。

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和伸缩组的实例 ID 到答题框。

4. 创建一个鲲鹏架构集群，具体要求为

- (1) 集群名称: kcloud;
- (2) 集群版本: v1.15.11;
- (3) 地域: 贵阳;
- (4) 集群管理规模: 50 节点;
- (5) 控制节点数: 3;
- (6) 节点使用子网: intnetX-server;
- (7) Pod 实例上限: 64;
- (8) 容器网段: 10.10.0.0/16。

节点配置信息要求为

- (1) 节点名称: kcloud-server;
- (2) 节点规格: kc.xlarge.2
- (3) 节点: EulerOS 2.8

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和集群的 ID 到答题框。

5. 使用插件管理在 kcloud 容器集群中安装 dashboard 可视化监控界面。完成后，提交 dashboard 访问地址到答题框。

6. 购买云硬盘存储卷，具体配置为

- (1) 名称为: ckcloud-disk;
- (2) 集群: kcloud;
- (3) 容量: 30G;
- (4) 类型: 高 I/O;

完成后，创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和云硬盘的 ID 到答题框。

7. 在 kcloud 集群中安装 kubectl 命令，使用 kubectl 命令管理 kcloud 集群。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

8.在 kcloud 集群中定义名为 liveness-http 探针和 liveness-tcp 探针，监控 kcloud 集群。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

9.使用容器镜像服务作为镜像仓库，将提供的 httpd-2.4.33 和 httpd-2.4.35 镜像上传至服务中。提交当前的 AK（SecretId）、SK（SecretKey）和组织名称到答题框。

10.在 kcloud 集群中创建一个 Deployment，使用开源镜像中心的 nginx:latest，配置使其可以通过外部公网进行访问。完成后提交外部访问地址到答题框。

任务 3 自动化运维（10 分，本任务只公布考试范围，不公布赛题）

编写 Python 脚本调用公有云 API 完成公有云运维工作。

第一场次题目：OpenStack 平台部署与运维（3）

某企业拟使用 OpenStack 搭建一个企业云平台，以实现资源池化弹性管理、企业应用集中管理、统一安全认证和授权等管理。

系统架构如图 1 所示，IP 地址规划如表 1 所示。

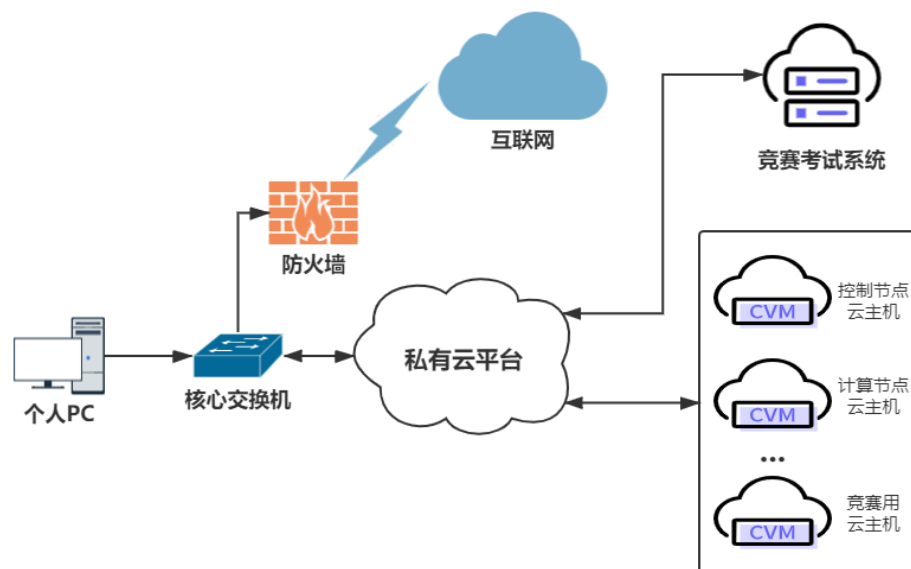


图 1 系统架构图

表 1 IP 地址规划

设备名称	主机名	接口	IP 地址	说明
云服务器1	Controller	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器2	Compute	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器3 ... 云服务器 n	自定义	eth0	172.129.x.0/24	用于实操题
PC-1		本地连接	172.24.16.0/24	PC 使用

说明：

1.竞赛使用集群模式进行，比赛时给每个参赛队提供独立的租户与用户，各用户的资源配额相同，选手通过用户名与密码登录竞赛用私有云平台，创建云主机进行相应答题；

2.表中的 x 为工位号；在进行 OpenStack 搭建时的第二块网卡地址根据题意自行创建；

3.根据图表给出的信息，检查硬件连线及网络设备配置，确保网络连接正常；

4.考试所需要的账号资源、竞赛资源包与附件均会在考位信息表与设备确认单中给出；

5.竞赛过程中，为确保服务器的安全，请自行修改服务器密码；在考试系统提交信息时，请确认自己的 IP 地址，用户名和密码。

任务 1 基础运维任务（5 分）

1. 使用提供的用户名密码，登录竞赛用的云计算平台，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网络，第二张网卡使用的网络需自行创建（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；

完成后提交控制节点的用户名、密码和 IP 地址到答题框。

2. 使用提供的 http 服务地址，在 http 服务下，存放有 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。完成后提交控制节点的用户名、密码和 IP 地址到答题框。

3. 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。完成后提交控制节点的用户名、密码和 IP 地址到答题框。

4. 在 compute 节点上利用空白分区划分 2 个 20G 分区。完成后提交计算节点的用户名、密码和 IP 地址到答题框。

任务 2 OpenStack 搭建与运维任务（10 分）

任务需要完成 openstack 平台的安装搭建及运维任务（无安装脚本），表 2 提供了在安装过程中需要使用相关服务参数。

表 2 密码

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	eth1（外网卡名，以实际为准）

1. 使用提供的脚本框架 `iaas-pre-host.sh` 和 `openrc.sh` 环境变量文件，填充脚本，在 `controller` 和 `compute` 节点上分别安装 `openstack` 平台的基础组件并完成相关配置。

2. 使用提供的脚本框架 `iaas-install-mysql.sh`，填充脚本，在 `controller` 节点上安装 `mariadb`、`mencached`、`rabbitmq` 等服务并完成相关配置。

3. 使用提供的脚本框架 `iaas-install-keystone.sh`，填充脚本，在 `controller` 节点上安装 `keystone` 服务并完成相关配置。

4. 使用提供的脚本框架 `iaas-install-glance.sh`，填充脚本，在 `controller` 节点上安装 `glance` 服务并完成相关配置。

5. 使用提供的脚本框架 `iaas-install-nova-controller.sh` 和 `iaas-install-nova-compute.sh`，填充脚本，在 `controller` 和 `compute` 节点上安装 `nova` 服务并完成配置。

6. 使用提供的脚本框架 `iaas-install-neutron-controller.sh` 和 `iaas-install-neutron-compute.sh`，填充脚本，在 `controller` 和 `compute` 节点上安装 `neutron` 服务并完成配置。

7. 使用提供的脚本框架 `iaas-install-dashboard.sh`，填充脚本，在 `controller` 节点上安装 `dashboard` 服务并完成相关配置。

8. 使用提供的脚本框架 `iaas-install-cinder-controller.sh` 和 `iaas-install-cinder-compute.sh`，填充脚本，在 `controller` 和 `compute` 节点上安装 `cinder` 服务并完成配置。

9. 使用提供的脚本框架 `iaas-install-swift-controller.sh` 和 `iaas-install-swift-compute.sh`，填充脚本，在 `controller` 和 `compute` 节点上安装 `swift` 服务并完成配置。

10. 使用提供的脚本框架 `iaas-install-heat.sh`，填充脚本，在 `controller` 节点上安装 `heat` 服务并完成相关配置。

（注：为了避免 OpenStack 不能成功搭建而影响后面竞赛进程，任务 3-4 向选手提供 openstack 平台（无 dashboard 组件），供参赛选手进行竞赛）

任务 3 OpenStack 云平台 API 运维开发（10 分）

1. 在 `controller` 节点的 `/root` 目录下编写 Python 程序 `create_user.py` 文件，对接 `openstack api`，创建一个用户，描述为“API create user!”（如果存在同名用户，代码中需先进行删除操作）。

2. 在 `controller` 节点的 `/root` 目录下按要求编写 Python 程序 `create_flavor.py` 文件，对接

openstack api, 要求创建云主机类型 pvm_heat: 内存 1024M、硬盘 20G、vcpu 1 个 (如果存在同名主机类型, 代码中需先进行删除操作)。输出云主机类型名称、id 和详细信息。

3. 在 controller 节点的/root 目录下按要求编写 Python 程序 create_network.py 文件, 对接 openstack api, 要求: (1) 为平台创建内部网络 pvm_int, 子网名称为 pvm_intsubnet; (2) 设置云主机网络子网 IP 网段为 192.168.x.0/24 (其中 x 是考位号), 网关为 192.168.x.1 (如果存在同名内网, 代码中需先进行删除操作)。输出内网名称、id 和详细信息。

4. 在 controller 节点的/root 目录下按要求编写 Python 程序 create_vm.py 文件, 对接 openstack api, 要求使用 pvm_image、pvm_flavor、pvm_intsubnet、pvm_router 组件创建 1 台主机 pvm1 (如果存在同名虚拟机, 代码中需先进行删除操作)。输出虚拟机名称、ID 和详细信息。

5. 在 controller 节点的/root 目录下按要求编写 Python 程序 create_glance.py 文件, 对接 openstack api, 要求上传并使用镜像文件 cirros-0.3.4-x86_64-disk.img 创建 glance 镜像 pvm_image (如果存在同名镜像, 代码中需先进行删除操作)。输出镜像名称、id 和详细信息。

任务 4 OpenStack 云平台运维与开发 (10 分, 任务只公布考试范围, 不公布赛题)

任务包含多个子任务, 内容包含编写 Python 脚本调用 OpenStack API 对 OpenStack 云平台进行管理和运维; 使用自动化运维工具 Ansible 对云主机进行批量系统配置、批量程序部署、批量运行命令等运维操作。

第二场次题目：容器云平台部署与运维（3）

说明：完成本任务需要两台安装了 CentOS7.5 操作系统的云主机 master 和 node，镜像包中有本次容器云平台部署和运维所需的所有文件。

云梦公司技术部产品开发上线周期长，用户规模大且版本更新频繁。新版本的每次上线，产品都要承受极大的压力。引入 CICD (Continuous Integration 持续集成、Continuous Delivery 持续交付) 和灰度发布成了公司的当务之急，研发团队决定搭建基于 Kubernetes 的 CICD 环境，希望基于这个平台来实现 DevOps 的部分流程，并基于 Kubernetes 实现业务系统的灰度发布。

为了能够让云梦公司开发的 web 应用系统产品能够缩减开发周期，快速迭代版本，并实现业务从老版本到新版本的平滑过渡，避免升级过程中出现的问题对用户造成的影响。研发部决定使用微服务架构，实现基于 Kubernetes 的容器化部署、CICD 和灰度发布。

节点规划如表 1 所示。

表 1 容器云平台节点规划

节点角色	主机名	VCPUS	内存	硬盘
Master、Harbor、CICD	master	8	12G	100G
Worker Node	node	4	8G	100G

任务 1 Docker CE 及私有仓库安装任务（5 分）

1. 在 master、node 各节点中分别安装 DockerCE 和 docker-compose。完成后提交 master 节点的用户名、密码和 IP 到答题框。

2. 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 master 节点的用户名、密码和 IP 到答题框。

3. 在 master 节点上编写/root/rabbitmq/docker-compose.yaml 文件，具体要求为

- （1）容器名称：rabbitmq；镜像：rabbitmq:3.8.3-management；
- （2）设置 rabbitmq 默认用户及密码均为 root；
- （3）设置容器重启策略为 always；
- （4）启用 rabbitmq 管理插件。

完成后编排部署 rabbitmq，并提交 master 节点的用户名、密码和 IP 到答题框。

任务 2 基于容器的 web 应用系统部署任务（10 分）

将云梦公司开发的基于微服务架构的 ChinaSkillsSA 智能文本的情绪分析（Sentiment Analysis）的系统实现全容器化部署，ChinaSkillsSA 智能分析系统应用系统架构如图 1 所示，ChinaSkillsSA 智能分析系统说明如表 2 所示。

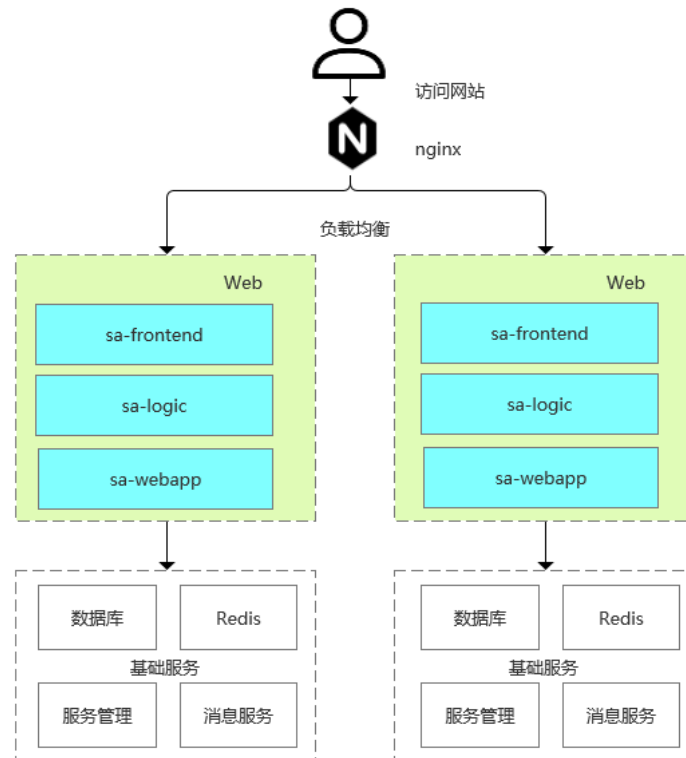


图 1 ChinaSkillsSA 智能分析系统架构图

表 2 ChinaSkillsSA 智能分析系统说明

模块	使用技术	说明
sa-frontend	Node.js	前端项目，使用 Vue.js 开发。
sa-logic	Python Flask	输入文本句子的情绪分析微服务，使用 Python 开发。
sa-webapp	Spring Cloud	Web 服务，处理 sa-frontend 请求，调用 sa-logic 微服务，使用 Java 语言开发。

将 sa-frontend 组件、sa-logic 组件和 sa-webapp 组件按照要求进行容器化。

1. 在 master 节点上编写 /root/sentiment-analyzer/Dockerfile-logic 文件构建

chinaskillsa-logic:v1.0 镜像，具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 Python3;
- (4) 使用 pip 安装提供的离线模块包。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写 /root/sentiment-analyzer/Dockerfile-webapp 文件构建 chinaskillsa-webapp:v1.1 镜像，具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 go 工具;
- (4) 使用 go 构建 webapp 项目;
- (5) 开放 8080 端口。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

3. 在 master 节点上编写 /root/sentiment-analyzer/Dockerfile-frontend 文件构建 chinaskillsa-frontend:v1.1 镜像，具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 nodejs 工具及 nginx;
- (4) 开放端口: 80;
- (5) 设置 nginx 开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

4. 在 master 节点上编写 /root/sentiment-analyzer/docker-compose.yaml 文件，具体要求为

- (1) 容器 1 名称: sa-logic; 镜像: chinaskillsa-logic:v1.1; 端口映射: 9001:5000;
- (2) 容器 2 名称: sa-webapp; 镜像: chinaskillsa-webapp:v1.1; 端口映射: 9002:8080;
- (3) 容器 3 名称: sa-frontend; 镜像: chinaskill-frontend:v1.1; 端口映射: 9003:80。

完成后编排部署情感分析系统，并能对输入的语句进行情感分析。然后提交 master 节

点的用户名、密码和 IP 到答题框。

任务 3 基于容器的持续集成部署任务（10 分）

公司决定采用 GitLab + Jenkins 来构建 CI/CD 环境，以缩短新功能开发上线周期，及时满足客户的需求，实现 DevOps 的部分流程，来减轻部署运维的负担，实现可视化容器生命周期管理、应用发布和版本迭代更新，完成 GitLab + Jenkins + Kubernetes 的 CI/CD 环境部署。CI/CD 应用系统架构如图 2 所示。

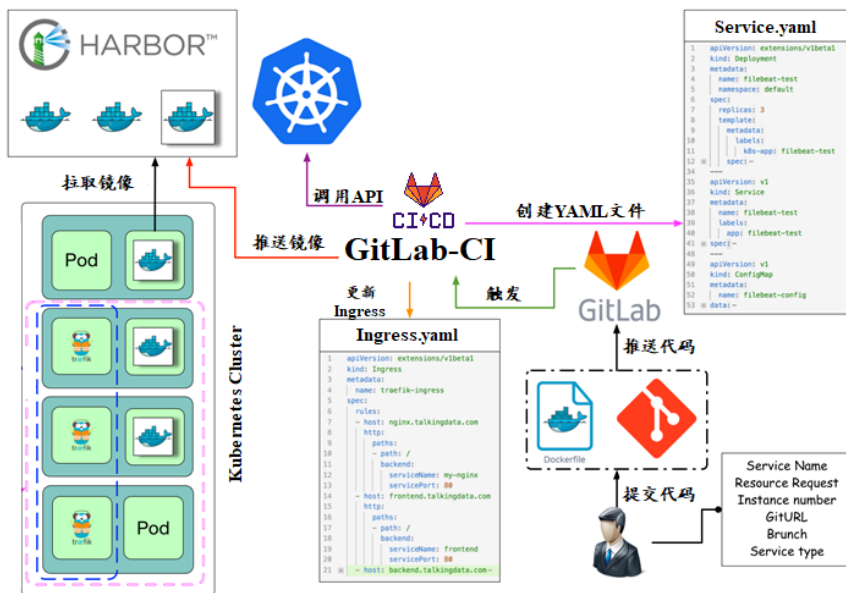


图 2 CI/CD 应用系统架构图

1. 在 master 节点上编写/root/jenkins/docker-compose.yaml 文件编排部署 Jenkins 服务，具体要求为

- (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: springcloud; 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 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写/root/gitlab/docker-compose.yaml 文件编排部署 Gitlab 服务，具体要求为

- (1) 容器名称: gitlab;
- (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 Springcloud, 将/opt/Springcloud 中的代码上传到 Springcloud 项目中。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

5. 配置 Jenkins 连接 Gitlab, 具体要求为

- (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 master 节点的用户名、密码和 IP 到答题框。

4.配置 Jenkins 连接 maven, 具体要求为

- (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 master 节点的用户名、密码和 IP 到答题框。

5.配置 CI/CD, 具体要求为

- (1) 新建一个流水线任务 Springcloud;
- (2) 编写流水线脚本, 构建 Springcloud 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 springcloud 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 springcloud。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

8. 将 Springcloud 项目中的代码上传到 Gitlab 中触发构建, 完成后提交 master 节点的用

用户名、密码和 IP 到答题框。

任务 4 Kubernetes 容器云平台部署与运维（10 分，本任务只公布考试范围，不公布赛题）

云梦公司为实现互联网智能服务的升级，确保原有微服务正常运行支撑业务需求，新升级的微服务能够进行测试和替代，公司决定采用基于 Istio 的灰度发布（又名金丝雀发布）来实现业务从老版本到新版本的平滑过渡，并避免升级过程中出现的问题对用户造成的影响。

请根据要求，完成 Kubernetes 的集群运维、调度、网络、存储、安全、应用生命周期管理、日志/监控等运维任务。

第三场次题目：公有云部署与运维（3）

云梦是一家科技公司，现在公司开发全面转向云原生，因此需要对公司内部项目搭建基于云原生的 Devops 相关服务。

某公有云是一家成熟的公有云提供商，为客户提供 K8S、云网络、云硬盘、云防火墙、负载均衡等服务，可创建 Web 服务，共享文件存储服务，数据库服务，数据库集群等服务。

根据上述公有云平台的特性，完成公有云中的各项运维工作。

任务 1 基础设施构建（10 分）

1.在公有云中完成虚拟私有云的创建，具体要求为

- (1) 在贵阳区域进行创建操作；
- (2) 创建一个名为 intnetX 的虚拟私有云：IP 地址为 172.16.1.0/16；
- (3) 创建子网名称为 intnetX-server：IP 地址为 172.16.1.0/24；
- (4) 创建一个名为 net-data 的虚拟私有云：网段为 192.168.1.0/16
- (5) 创建子网名称为 net-mysql：IP 地址为 192.168.1.0/24；

完成创建后，提交当前的 AK（SecretId）、SK（SecretKey）和 intnetX 网络的 ID 到答题框。

2.创建对等连接 intnetX-net，配置 intnetX 虚拟私有云和 net-data 虚拟机私有云连通。操作完成后，提交当前的 AK（SecretId）、SK（SecretKey）和对等连接的 ID 到答题框。

3.登录华为云平台，创建两台云实例，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) CPU 架构：鲲鹏计算；
- (4) 规格：kc1.xlarge.2；
- (5) 镜像：CentOS 7.5 64 位；
- (6) 系统盘：高 IO 50G 硬盘；
- (7) 虚拟网络：intnetX-server；
- (8) 实例名称：ChinaSkill-node-1、ChinaSkill-node-2；
- (9) 登录方式：使用密码登录，密码自定义。

创建完成后，提交当前的 AK（SecretId）、SK（SecretKey）和 ChinaSkill-node-1 的实例 ID 到答题框。

4.创建一台云数据库，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) 数据库版本：MySQL 5.7；
- (4) 实例类型：主备；
- (5) 实例规格：鲲鹏 2 核 8GB/40GB；
- (6) 虚拟私有云：net-mysql；
- (7) 字符集：UTF8；
- (8) 端口：3306；
- (9) 数据库密码：Root123456；
- (10) 数据库名称：chinaskill-mysql；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和云数据库的实例 ID 到答题框。

5.给云数据库 chinaskill-mysql 创建只读数据库 chinaskill-read 实例。使用 2vCPU、4GB 内存规格。并给 chinaskill-mysql 申请读写分离地址。完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和只读数据库实例 ID 到答题框。

6.创建一个 Redis 缓存实例，具体要求为

- (1) 版本号：5.0；
- (2) 实例类型：主备；
- (3) 副本数：2；
- (4) 实例规格：4G；
- (5) 子网：net-mysql；
- (6) 密码：Root123456；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和实例 ID 到答题框。

7.在 chinaskill-node-2 云服务器上，使用提供的软件包安装 kafka 和 zookeeper 服务。安装完成后运行服务。完成后提交连接 chinaskill-node-2 节点的用户名、密码和公网 IP 地址到答题框。

8.在 chinaskill-node-1 云服务上部署 gpmall，mysql 使用云数据库分写分离地址，配置公有云 Redis 访问路径，使用提供的公有云软件包 gpmall-package 完成商城的上云操作。并设置 gpmall 所需 jar 软件包开启自动启动。完成后提交 chinaskill-node-1 的用户名、密码和公网 IP 到答题框。

9.创建一个云服务器备份存储库名为 `server_backup`，容量为 100G。将 `ChinaSkill-node-1` 云服务器制作镜像文件 `chinaskill-image`。完成后，将当前的 AK（SecretId）、SK（SecretKey）和存储库 ID 到答题框。

任务 2 系统管理与维护（10 分）

1.使用所创建的私有镜像 `chinaskill-image` 购买云服务器 `ChinaSkill-node-3`，配置与 `ChinaSkill-node-1` 云服务器相同。完成后提交 `chinaskill-node-3` 的用户名、密码和公网 IP 到答题框。

2.创建一个负载均衡器 `chinaskill-elb`，将 `ChinaSkill-node-1` 和 `ChinaSkill-node-3` 加入负载均衡的后端。设置一个可用的公网服务 IP，服务端口为 80。配置监听器，监听 80 端口。对浮动公共 IP 进行 Web 访问测试。完成后，提交当前的 AK（SecretId）、SK（SecretKey）和负载均衡器的 ID 到答题框。

3.新建一个启动配置，具体要求为

- (1) 启动配置名称：`template-exam`；
- (2) 计费模式：按量计费；
- (3) 地域：贵阳；
- (4) CPU 架构：鲲鹏计算；
- (5) 规格：`kc1.xlarge.2`；
- (6) 镜像：`chinaskill-image`；
- (7) 系统盘：高 IO 50G 硬盘；
- (8) 登录方式：使用密码登录，密码自定义。

创建一个伸缩组，具体要求为

- (1) 名称：`as-exam`；
- (2) 最小伸缩数：1；起始实例数：1；最大伸缩数：5；
- (3) 启动配置：`template-exam`；
- (4) 使用负载均衡：`chinaskill-elb`；
- (5) 移出策略：移出最旧的实例。

为伸缩组 `as-exam` 新建告警触发策略，具体要求为

(1) 如果实例的内存利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例数减少 1 台。冷却 60 秒；

(2) 如果实例的内存利用率在 5 分钟内的最大值大于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒；

(3) 如果实例的 CPU 利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例数减少 1 台。冷却 60 秒；

(4) 如果实例的 CPU 利用率在 5 分钟内的平均值大于等于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒。

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和伸缩组的实例 ID 到答题框。

4. 创建一个鲲鹏架构集群，具体要求为

- (1) 集群名称: kcloud;
- (2) 集群版本: v1.15.11;
- (3) 地域: 贵阳;
- (4) 集群管理规模: 50 节点;
- (5) 控制节点数: 3;
- (6) 节点使用子网: intnetX-server;
- (7) Pod 实例上限: 64;
- (8) 容器网段: 10.10.0.0/16。

节点配置信息要求为

- (1) 节点名称: kcloud-server;
- (2) 节点规格: kc.xlarge.2
- (3) 节点: EulerOS 2.8

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和集群的 ID 到答题框。

5. 使用插件管理在 kcloud 容器集群中安装 dashboard 可视化监控界面。完成后，提交 dashboard 访问地址到答题框。

6. 购买云硬盘存储卷，具体配置为

- (1) 名称为: ccloud-disk;
- (2) 集群: kcloud;
- (3) 容量: 30G;
- (4) 类型: 高 I/O;

完成后，创建完成后提交当前的 AK（SecretId）、SK（SecretKey）和云硬盘的 ID 到答题框。

7.在 kcloud 集群中安装 kubectl 命令，使用 kubectl 命令管理 kcloud 集群。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

8.使用提供的 helm 软件包，在 kcloud 集群中安装 helm 服务。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

9.使用容器镜像服务作为镜像仓库，将提供的 httpd-2.4.33 和 httpd-2.4.35 镜像上传至服务中。提交当前的 AK（SecretId）、SK（SecretKey）和组织名称到答题框。

10.在 kcloud 集群中创建一个 Deployment，使用所上传的 httpd-2.4.33 容器镜像，配置使其可以通过外部公网进行访问。完成后提交外部访问地址到答题框。

任务 3 自动化运维（10 分，本任务只公布考试范围，不公布赛题）

编写 Python 脚本调用公有云 API 完成公有云运维工作。

第一场次题目：OpenStack 平台部署与运维（4）

某企业拟使用 OpenStack 搭建一个企业云平台，以实现资源池化弹性管理、企业应用集中管理、统一安全认证和授权等管理。

系统架构如图 1 所示，IP 地址规划如表 1 所示。

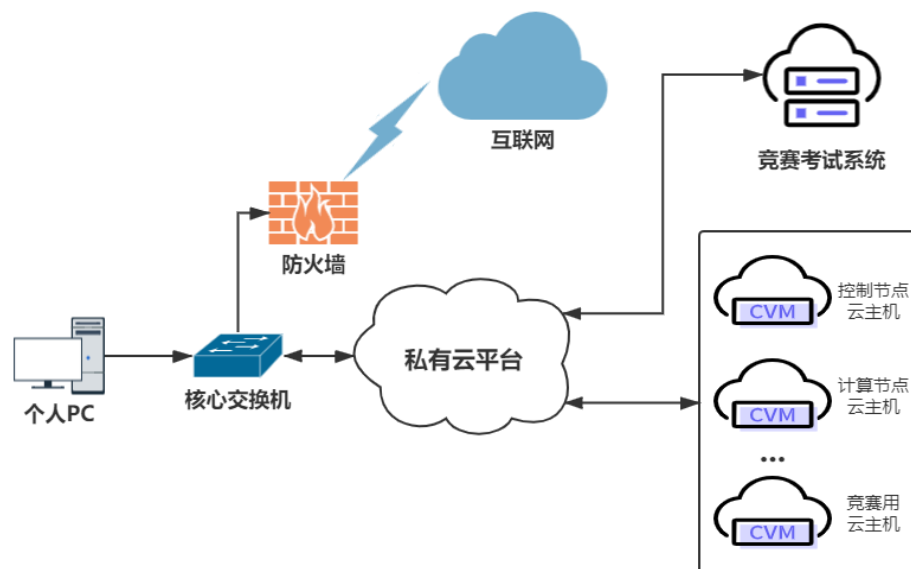


图 1 系统架构图

表 1 IP 地址规划

设备名称	主机名	接口	IP 地址	说明
云服务器1	Controller	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器2	Compute	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器3	自定义	eth0	172.129.x.0/24	用于实操题
... 云服务器 n				
PC-1		本地连接	172.24.16.0/24	PC 使用

说明：

- 1.竞赛使用集群模式进行，比赛时给每个参赛队提供独立的租户与用户，各用户的资源配额相同，选手通过用户名与密码登录竞赛用私有云平台，创建云主机进行相应答题；
- 2.表中的 x 为工位号；在进行 OpenStack 搭建时的第二块网卡地址根据题意自行创建；
- 3.根据图表给出的信息，检查硬件连线及网络设备配置，确保网络连接正常；
- 4.考试所需要的账号资源、竞赛资源包与附件均会在考位信息表与设备确认单中给出；
- 5.竞赛过程中，为确保服务器的安全，请自行修改服务器密码；在考试系统提交信息时，请确认自己的 IP 地址，用户名和密码。

任务 1 基础运维任务（5 分）

1. 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；

完成后提交控制节点的用户名、密码和 IP 地址到答题框。

2. 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。完成后提交控制节点的用户名、密码和 IP 地址到答题框。

3. 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。完成后提交控制节点的用户名、密码和 IP 地址到答题框。

4. 在 compute 节点上利用空白分区划分 2 个 20G 分区。完成后提交计算节点的用户名、密码和 IP 地址到答题框。

任务 2 OpenStack 搭建与运维任务（10 分）

本任务需要完成 openstack 平台的安装搭建及运维任务（无安装脚本），表 2 提供了在安装过程中需要使用相关服务参数。

表 2 密码

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	eth1（外网卡名，以实际为准）

1. 使用提供的脚本框架 `iaas-pre-host.sh` 和 `openrc.sh` 环境变量文件，填充脚本，在 `controller` 和 `compute` 节点上分别安装 `openstack` 平台的基础组件并完成相关配置。
2. 使用提供的脚本框架 `iaas-install-mysql.sh`，填充脚本，在 `controller` 节点上安装 `mariadb`、`mencached`、`rabbitmq` 等服务并完成相关配置。
3. 使用提供的脚本框架 `iaas-install-keystone.sh`，填充脚本，在 `controller` 节点上安装 `keystone` 服务并完成相关配置。
4. 使用提供的脚本框架 `iaas-install-glance.sh`，填充脚本，在 `controller` 节点上安装 `glance` 服务并完成相关配置。
5. 使用提供的脚本框架 `iaas-install-nova-controller.sh` 和 `iaas-install-nova-compute.sh`，填充脚本，在 `controller` 和 `compute` 节点上安装 `nova` 服务并完成配置。
6. 使用提供的脚本框架 `iaas-install-neutron-controller.sh` 和 `iaas-install-neutron-compute.sh`，填充脚本，在 `controller` 和 `compute` 节点上安装 `neutron` 服务并完成配置。
7. 使用提供的脚本框架 `iaas-install-dashboard.sh`，填充脚本，在 `controller` 节点上安装 `dashboard` 服务并完成相关配置。
8. 使用提供的脚本框架 `iaas-install-cinder-controller.sh` 和 `iaas-install-cinder-compute.sh`，填充脚本，在 `controller` 和 `compute` 节点上安装 `cinder` 服务并完成配置。
9. 使用提供的脚本框架 `iaas-install-swift-controller.sh` 和 `iaas-install-swift-compute.sh`，填充脚本，在 `controller` 和 `compute` 节点上安装 `swift` 服务并完成配置。。
10. 使用提供的脚本框架 `iaas-install-heat.sh`，填充脚本，在 `controller` 节点上安装 `heat` 服务并完成相关配置。

（注：为了避免 OpenStack 不能成功搭建而影响后面竞赛进程，任务 3-4 向选手提供 openstack 平台（无 dashboard 组件），供参赛选手进行竞赛）

任务 3 OpenStack 云平台 API 运维开发（10 分）

1. 在 `controller` 节点的 `/root` 目录下按要求编写 Python 程序 `create_glance.py` 文件，对接 `openstack api`，要求上传并使用镜像文件 `pylinux.qcow2` 创建 `glance` 镜像 `pvm_image`（如果存在同名镜像，代码中需先进行删除操作）。输出镜像名称、`id` 和详细信息。
2. 在 `controller` 节点的 `/root` 目录下按要求编写 Python 程序 `create_route.py` 文件，对接 `openstack api`，要求获取 `openstack` 平台外网 ID，创建路由 `pvm_router`，添加内网子网接口到路由，实现内外网网络互通（如果存在同名路由，代码中需先进行删除操作）。输出路由名称、`id` 和详细信息。
3. 使用云平台提供的 `manila` 环境，在 `controller` 节点的 `/root` 目录下按要求编写 Python 程序 `create_ml.py` 文件，对接 `openstack api`，要求添加 `nfs` 扩展卷 `pvm_nfs`，大小 20G、添加 `share-network` 为 `demo-share-network1`（如果存在同名 `nfs` 扩展卷，代码中需先进行删除

操作)。输出 nfs 扩展卷名称、id 和详细信息。

4. 1. 在 controller 节点的/root 目录下编写 Python 程序 create_user.py 文件, 对接 openstack api, 创建一个用户, 描述为 “API create user!” (如果存在同名用户, 代码中需先进行删除操作)。

5. 使用云平台搭建 heat 环境, 在 controller 节点的/root 目录下按要求编写 Python 程序 create_heat02.py 文件, 对接 openstack api, 要求调用 server.yaml 脚本创建名为 pvm_stack 的堆栈(在/root 目录下自行创建 server.yaml 文件, 实现创建主机类型 m1.flavor: ID 为 123456、内存 1024M、硬盘 20G、vcpu 2 个) (如果存在同名堆栈, 代码中需先进行删除操作)。输出堆栈名、id 和详细信息。

任务 4 OpenStack 云平台运维开发 (10 分, 本任务只公布考试范围, 不公布赛题)

此任务包含多个子任务, 内容涉及编写 Python 脚本调用 OpenStack API 对 OpenStack 云平台进行管理和运维。使用自动化运维工具 Ansible 对云主机进行批量系统配置、批量程序部署、批量运行命令等运维操作。

第二场次题目：容器云平台部署与运维（4）

说明：完成本任务需要两台安装了 CentOS7.5 操作系统的云主机 master 和 node，镜像包中有本次容器云平台部署和运维所需的所有文件。

云梦公司技术部产品开发上线周期长，用户规模大且版本更新频繁。新版本的每次上线，产品都要承受极大的压力。引入 CICD (Continuous Integration 持续集成、Continuous Delivery 持续交付) 和灰度发布成了公司的当务之急，研发团队决定搭建基于 Kubernetes 的 CICD 环境，希望基于这个平台来实现 DevOps 的部分流程，并基于 Kubernetes 实现业务系统的灰度发布。

为了能够让云梦公司开发的 web 应用系统产品能够缩减开发周期，快速迭代版本，并实现业务从老版本到新版本的平滑过渡，避免升级过程中出现的问题对用户造成的影响。研发部决定使用微服务架构，实现基于 Kubernetes 的容器化部署、CICD 和灰度发布。

节点规划如表 1 所示。

表 1 容器云平台节点规划

节点角色	主机名	VCPUS	内存	硬盘
Master、Harbor、CICD	master	8	12G	100G
Worker Node	node	4	8G	100G

任务 1 Docker CE 及私有仓库安装任务（5 分）

1. 在 master、node 各节点中分别安装 DockerCE 和 docker-compose。完成后提交 master 节点的用户名、密码和 IP 到答题框。

2. 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 master 节点的用户名、密码和 IP 到答题框。

3. 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 master 节点的用户名、密码和 IP 到答题框。

任务 2 基于容器的 web 应用系统部署任务（10 分）

将云梦公司开发的基于微服务架构的 web 应用系统 ChinaSkillsMall 重新实现全容器化部署，原微服务架构的 Web 应用系统架构如图 1 所示，模块功能介绍图表 2 所示。

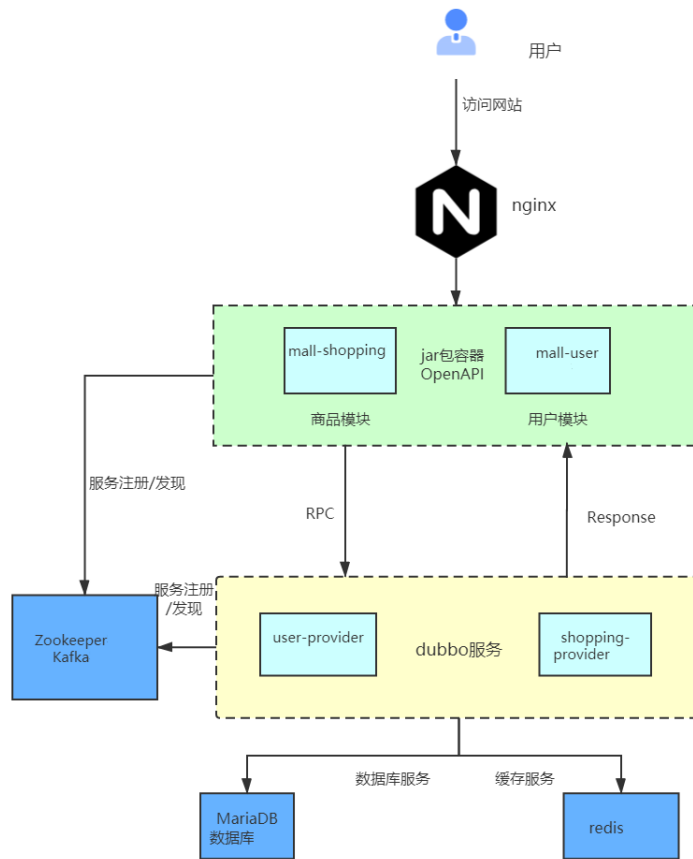


图 1 ChinaSkillsMall 系统架构图

表 2 ChinaSkillsMall 模块功能介绍

模块	使用技术	备注
mall.sql	MySQL	网站的数据库
dist 目录	Nginx	网站的前端项目
mall-shopping	web 项目	8081 端口，商品/购物车/首页渲染等交互

mall-user		8082 端口，提供用户相关的交互，如登录、注册、个人中心等
user-provider	后端服务	提供用户相关服务
shopping-provider		提供购物车、推荐商品、商品等服务

将 MySQL 数据库组件、Redis 组件、Kafka 组件、Zookeeper 组件按照要求进行容器化。

1. 在 master 节点上编写 /root/mall-swarm/Dockerfile-redis 文件构建 chinaskillmall-redis:v1.1 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装 redis 服务;
- (4) 修改配置文件中的 bind 127.0.0.1 为 bind 0.0.0.0;
- (5) 设置 redis 免密，并关闭保护模式;
- (6) 开放端口：6379;
- (7) 设置服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写 /root/mall-swarm/Dockerfile-mariadb 文件构建 chinaskillmall-mariadb:v1.0 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装并初始化 mariadb，密码：123456;
- (4) 设置数据库默认字符编码为 UTF-8;
- (5) 创建数据库 gpmall 并导入 gpmall.sql;
- (6) 设置 mariadb 开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

3. 在 master 节点上编写 /root/gpmall/Dockerfile-zookeeper 文件构建 chinaskillmall-zookeeper:v1.1 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;

- (3) 安装 zookeeper 服务;
- (4) 开放端口 2181;
- (5) 设置 zookeeper 服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

4. 在 master 节点上编写/root/gpmall/Dockerfile-kafka 文件构建 chinaskillmall-kafka:v1.1 镜像，具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 kafka 服务;
- (4) 开放端口 9092;
- (5) 设置 zookeeper 服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

5. 在 master 节点上编写/root/gpmall/Dockerfile-nginx 文件构建 chinaskillmall-nginx:v1.1 镜像，具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 nginx 服务;
- (4) 将提供的软件包 dist.tar 和 jar 包拷贝进镜像;
- (5) 设置 nginx 服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

6. 在 master 节点上编写/root/gpmall/gpmall.yaml 文件，具体要求为

- (1) 在 default 命名空间下部署该服务;
- (2) service 名称: chinaskillmall;
- (3) 以 NodePort 方式对外暴露端口，端口映射: 30080:80;

完成后编排部署 ChinaSkillMall 商城。然后提交 master 节点的用户名、密码和 IP 到答题框。

任务3 基于容器的持续集成部署任务（10分）

该公司决定采用 GitLab + Jenkins 来构建 CICD 环境，以缩短新功能开发上线周期，及时满足客户的需求，实现 DevOps 的部分流程，来减轻部署运维的负担，实现可视化容器生命周期管理、应用发布和版本迭代更新，请完成 GitLab + Jenkins + Kubernetes 的 CICD 环境部署。CICD 应用系统架构如图 2 所示。

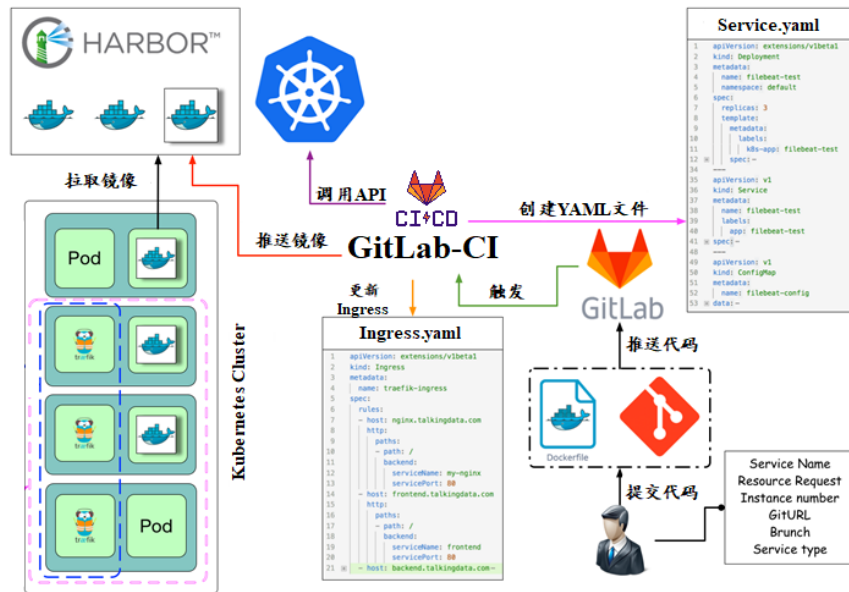


图 2 CICD 应用系统架构图

1. 在 master 节点上编写/root/jenkins/docker-compose.yaml 文件编排部署 Jenkins 服务，具体要求为

- (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: springcloud; 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 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写/root/gitlab/docker-compose.yaml 文件编排部署 Gitlab 服务，具体要求为

- (1) 容器名称: gitlab;
- (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 Springcloud, 将/opt/Springcloud 中的代码上传到 Springcloud 项目中。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

6. 配置 Jenkins 连接 Gitlab, 具体要求为

- (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 master 节点的用户名、密码和 IP 到答题框。

4.配置 Jenkins 连接 maven, 具体要求为

- (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 master 节点的用户名、密码和 IP 到答题框。

5.配置 CI/CD, 具体要求为

- (1) 新建一个流水线任务 Springcloud;
- (2) 编写流水线脚本, 构建 Springcloud 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 springcloud 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 springcloud。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

9. 将 Springcloud 项目中的代码上传到 Gitlab 中触发构建, 完成后提交 master 节点的用户名、密码和 IP 到答题框。

任务 4 Kubernetes 容器云平台部署与运维（10 分，本任务只公布考试范围，不公布赛题）

云梦公司为实现企业业务系统容器化，保证旧系统的服务正常运营，新开发的微服务逐步替换，公司决定采用基于 Istio 的灰度发布（又名金丝雀发布）来实现业务从老版本到新版本的平滑过渡，并避免升级过程中出现的问题对用户造成的影响。

请根据要求，完成 Kubernetes 的集群运维、调度、网络、存储、安全、应用生命周期管理、日志/监控等运维任务。

第三场次题目：公有云部署与运维（4）

云梦是一家科技公司，现在公司开发全面转向云原生，因此需要对公司内部项目搭建基于云原生的 Devops 相关服务。

某公有云是一家成熟的公有云提供商，为客户提供 K8S、云网络、云硬盘、云防火墙、负载均衡等服务，可创建 Web 服务，共享文件存储服务，数据库服务，数据库集群等服务。

根据上述公有云平台的特性，完成公有云中的各项运维工作。

任务 1 基础设施构建（10 分）

1.在公有云中完成虚拟私有云的创建，具体要求为

- （1）在贵阳区域进行创建操作；
- （2）创建一个名为 intnetX 的虚拟私有云：IP 地址为 172.16.1.0/16；
- （3）创建子网名称为 intnetX-server：IP 地址为 172.16.1.0/24；
- （4）创建一个名为 net-data 的虚拟私有云：网段为 192.168.1.0/16
- （5）创建子网名称为 net-mysql：IP 地址为 192.168.1.0/24；

完成创建后，提交当前的 AK（SecretId）、SK（SecretKey）和 intnetX 网络的 ID 到答题框。

2.创建对等连接 intnetX-net，配置 intnetX 虚拟私有云和 net-data 虚拟机私有云连通。操作完成后，提交当前的 AK（SecretId）、SK（SecretKey）和对等连接的 ID 到答题框。

3.登录华为云平台，创建两台云实例，具体要求为

- （1）计费模式：按需计费；
- （2）地域：贵阳；
- （3）CPU 架构：鲲鹏计算；
- （4）规格：kc1.xlarge.2；
- （5）镜像：CentOS 7.5 64 位；
- （6）系统盘：高 IO 50G 硬盘；
- （7）虚拟网络：intnetX-server；
- （8）实例名称：ChinaSkill-node-1、ChinaSkill-node-2；
- （9）登录方式：使用密码登录，密码自定义。

创建完成后，提交当前的 AK（SecretId）、SK（SecretKey）和 ChinaSkill-node-1 的实例 ID 到答题框。

4.创建一台云数据库，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) 数据库版本：MySQL 5.7；
- (4) 实例类型：主备；
- (5) 实例规格：鲲鹏 2 核 8GB/40GB；
- (6) 虚拟私有云：net-mysql；
- (7) 字符集：UTF8；
- (8) 端口：3306；
- (9) 数据库密码：Root123456；
- (10) 数据库名称：chinaskill-mysql；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和云数据库的实例 ID 到答题框。

5.使用公有云平台，申请对象存储服务，存储桶名字自定义。自行编写一个静态网页，网页的内容为“welcome to China”，使用申请的对象存储服务，托管该静态网页。托管成功后，提交你对象存储的访问域名到答题框。

6.创建一个网络 ACL，具体要求为

- (1) 名称：intnetX-acl；
- (2) 允许策略：只放行源地址为 172.16.1.0/24 访问 3306 端口；
- (3) 允许策略：只放行源地址为 172.16.1.0/24 访问 2181 端口；
- (4) 允许策略：只放行源地址为 172.16.1.0/24 访问 9092 端口；
- (5) 关联子网：关联网络 ACL 至 net-mysql 子网中；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和网络 ACL ID 到答题框。

7.创建一个 Redis 缓存实例，具体要求为

- (1) 版本号：5.0；
- (2) 实例类型：主备；
- (3) 副本数：2；
- (4) 实例规格：4G；
- (5) 子网：net-mysql；
- (6) 密码：Root123456；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和实例 ID 到答题框。

8.根据 ChinaSkill-node-1 云服务器配置购买 chinaskill-node-3 云服务器上,此服务器使用 net-mysql 私有网络。使用提供的软件包安装 kafka 和 zookeeper 服务。安装完成后运行服务。完成后提交连接 chinaskill-node-3 节点的用户名、密码和公网 IP 地址到答题框。

9.在 ChinaSkill-node-1 和 ChinaSkill-node-2 云服务上部署 gpmall, mysql 使用云数据库地址,配置公有云 Redis 访问路径,使用提供的公有云软件包 gpmall-package 完成商城的上云操作。并设置 gpmall 所需 jar 软件包开启自动启动。完成后提交 ChinaSkill-node-1 的用户名、密码和公网 IP 到答题框。

任务 2 系统管理与维护 (10 分)

1.创建一个云服务器备份存储库名为 server_backup,容量为 100G。将 ChinaSkill-node-1 云服务器制作镜像文件 chinaskill-image。完成后,将当前的 AK (SecretId)、SK (SecretKey) 和存储库 ID 到答题框。

2.创建一个负载均衡器 chinaskill-elb,将 ChinaSkill-node-1 和 ChinaSkill-node-2 加入负载均衡的后端。设置一个可用的公网服务 IP,服务端口为 80。配置监听器,监听 80 端口。对浮动公共 IP 进行 Web 访问测试。完成后,提交当前的 AK (SecretId)、SK (SecretKey) 和负载均衡器的 ID 到答题框。

3.新建一个启动配置,具体要求为

- (1) 启动配置名称: template-exam;
- (2) 计费模式: 按量计费;
- (3) 地域: 贵阳;
- (4) CPU 架构: 鲲鹏计算;
- (5) 规格: kc1.xlarge.2;
- (6) 镜像: chinaskill-image;
- (7) 系统盘: 高 IO 50G 硬盘;
- (8) 登录方式: 使用密码登录,密码自定义。

创建一个伸缩组,具体要求为

- (1) 名称: as-exam;
- (2) 最小伸缩数: 1; 起始实例数: 1; 最大伸缩数: 5;
- (3) 启动配置: template-exam;

(4) 使用负载均衡: chinaskill-elb;

(5) 移出策略: 移出最旧的实例。

为伸缩组 as-exam 新建告警触发策略, 具体要求为

(1) 如果实例的内存利用率在 5 分钟内的最大值小于 40%, 且连续发生 3 次。则实例数减少 1 台。冷却 60 秒;

(2) 如果实例的内存利用率在 5 分钟内的最大值大于 80%, 且连续发生 3 次。则实例数增加 1 台。冷却 60 秒;

(3) 如果实例的 CPU 利用率在 5 分钟内的最大值小于 40%, 且连续发生 3 次。则实例数减少 1 台。冷却 60 秒;

(4) 如果实例的 CPU 利用率在 5 分钟内的平均值大于等于 80%, 且连续发生 3 次。则实例数增加 1 台。冷却 60 秒。

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和伸缩组的实例 ID 到答题框。

4. 创建一个鲲鹏架构集群, 具体要求为

(1) 集群名称: kcloud;

(2) 集群版本: v1.15.11;

(3) 地域: 贵阳;

(4) 集群管理规模: 50 节点;

(5) 控制节点数: 3;

(6) 节点使用子网: intnetX-server;

(7) Pod 实例上限: 64;

(8) 容器网段: 10.10.0.0/16。

节点配置信息要求为

(1) 节点名称: kcloud-server;

(2) 节点规格: kc.xlarge.2

(3) 节点: EulerOS 2.8

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和集群的 ID 到答题框。

5. 使用插件管理在 kcloud 容器集群中安装 dashboard 可视化监控界面。完成后, 提交 dashboard 访问地址到答题框。

6. 购买云硬盘存储卷, 具体配置为

(1) 名称为: ckcloud-disk;

(2) 集群: kcloud;

(3) 容量: 30G;

(4) 类型: 高 I/O;

完成后, 创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和云硬盘的 ID 到答题框。

7.在 kcloud 集群中安装 kubectl 命令, 使用 kubectl 命令管理 kcloud 集群。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

8.使用提供的 helm 软件包, 在 kcloud 集群中安装 helm 服务。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

9.使用容器镜像服务作为镜像仓库, 将提供的 httpd-2.4.33 和 httpd-2.4.35 镜像上传至服务中。提交当前的 AK (SecretId)、SK (SecretKey) 和组织名称到答题框。

10.在 kcloud 集群中创建一个 Deployment, 使用开源镜像中心的 nginx:latest, 配置使其可以通过外部公网进行访问。完成后提交外部访问地址到答题框。

任务3 自动化运维 (10分, 本任务只公布考试范围, 不公布赛题)

编写 Python 脚本调用公有云 API 完成公有云运维工作。

第一场次题目：OpenStack 平台部署与运维（5）

某企业拟使用 OpenStack 搭建一个企业云平台，以实现资源池化弹性管理、企业应用集中管理、统一安全认证和授权等管理。

系统架构如图 1 所示，IP 地址规划如表 1 所示。

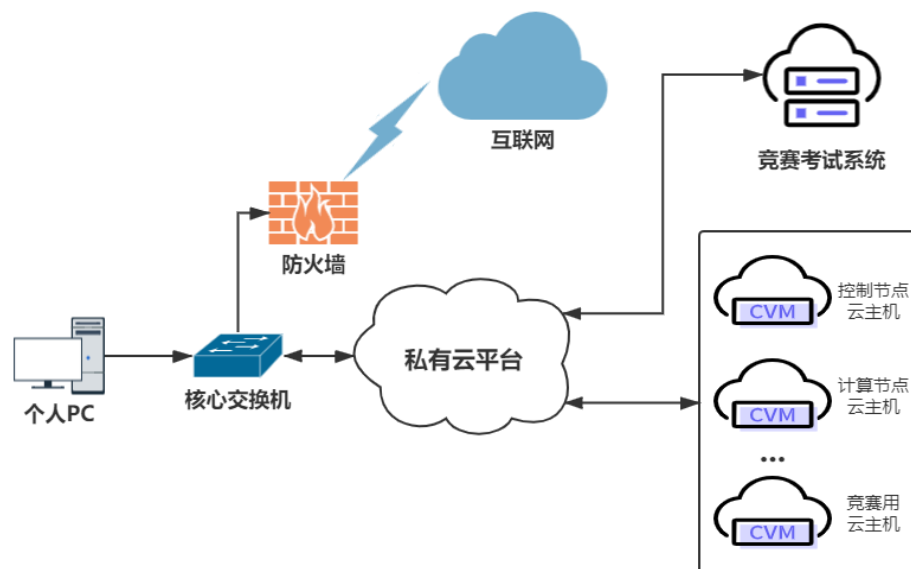


图 1 系统架构图

表 1 IP 地址规划

设备名称	主机名	接口	IP 地址	说明
云服务器1	Controller	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器2	Compute	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器3	自定义	eth0	172.129.x.0/24	用于实操题
云服务器 n				
PC-1		本地连接	172.24.16.0/24	PC 使用

说明：

- 1.竞赛使用集群模式进行，比赛时给每个参赛队提供独立的租户与用户，各用户的资源配额相同，选手通过用户名与密码登录竞赛用私有云平台，创建云主机进行相应答题；
- 2.表中的 x 为工位号；在进行 OpenStack 搭建时的第二块网卡地址根据题意自行创建；
- 3.根据图表给出的信息，检查硬件连线及网络设备配置，确保网络连接正常；
- 4.考试所需要的账号资源、竞赛资源包与附件均会在考位信息表与设备确认单中给出；
- 5.竞赛过程中，为确保服务器的安全，请自行修改服务器密码；在考试系统提交信息时，请确认自己的 IP 地址，用户名和密码。

任务 1 基础运维任务（5 分）

1. 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；

完成后提交控制节点的用户名、密码和 IP 地址到答题框。

2. 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。完成后提交控制节点的用户名、密码和 IP 地址到答题框。

3. 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。完成后提交控制节点的用户名、密码和 IP 地址到答题框。

4. 在 compute 节点上利用空白分区划分 2 个 20G 分区。完成后提交计算节点的用户名、密码和 IP 地址到答题框。

任务 2 OpenStack 搭建与运维任务（10 分）

本任务需要完成 openstack 平台的安装搭建及运维任务（无安装脚本），表 2 提供了在安装过程中需要使用相关服务参数。

表 2 密码

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	eth1（外网卡名，以实际为准）

1. 使用提供的脚本框架 `iaas-pre-host.sh` 和 `openrc.sh` 环境变量文件，填充脚本，在 `controller` 和 `compute` 节点上分别安装 `openstack` 平台的基础组件并完成相关配置。
2. 使用提供的脚本框架 `iaas-install-mysql.sh`，填充脚本，在 `controller` 节点上安装 `mariadb`、`mencached`、`rabbitmq` 等服务并完成相关配置。
3. 使用提供的脚本框架 `iaas-install-keystone.sh`，填充脚本，在 `controller` 节点上安装 `keystone` 服务并完成相关配置。
4. 使用提供的脚本框架 `iaas-install-glance.sh`，填充脚本，在 `controller` 节点上安装 `glance` 服务并完成相关配置。
5. 使用提供的脚本框架 `iaas-install-nova-controller.sh` 和 `iaas-install-nova-compute.sh`，填充脚本，在 `controller` 和 `compute` 节点上安装 `nova` 服务并完成配置。
6. 使用提供的脚本框架 `iaas-install-neutron-controller.sh` 和 `iaas-install-neutron-compute.sh`，填充脚本，在 `controller` 和 `compute` 节点上安装 `neutron` 服务并完成配置。
7. 使用提供的脚本框架 `iaas-install-dashboard.sh`，填充脚本，在 `controller` 节点上安装 `dashboard` 服务并完成相关配置。
8. 使用提供的脚本框架 `iaas-install-cinder-controller.sh` 和 `iaas-install-cinder-compute.sh`，填充脚本，在 `controller` 和 `compute` 节点上安装 `cinder` 服务并完成配置。
9. 使用提供的脚本框架 `iaas-install-swift-controller.sh` 和 `iaas-install-swift-compute.sh`，填充脚本，在 `controller` 和 `compute` 节点上安装 `swift` 服务并完成配置。。
10. 使用提供的脚本框架 `iaas-install-heat.sh`，填充脚本，在 `controller` 节点上安装 `heat` 服务并完成相关配置。

（注：为了避免 OpenStack 不能成功搭建而影响后面竞赛进程，任务 3-4 向选手提供 openstack 平台（无 dashboard 组件），供参赛选手进行竞赛）

任务 3 OpenStack 云平台 API 运维开发（10 分）

1. 在 `controller` 节点的 `/root` 目录下按要求编写 Python 程序 `create_sec.py` 文件，对接 `openstack api`，要求在云平台上创建一个新的安全组 `pvm_sec`，开放 20、21、22、80、3306 端口。输出安全组名称、id 和详细信息。
2. 在 `controller` 节点的 `/root` 目录下按要求编写 Python 程序 `create_vm.py` 文件，对接 `openstack api`，要求上传并使用镜像文件 `pylinux.qcow2` 创建 `glance` 镜像 `pvm_image`。输出该镜像的镜像名称、id 和详细信息。
3. 在 `controller` 节点的 `/root` 目录下按要求编写 Python 程序 `create_vm.py` 文件，对接 `openstack api`，要求使用 `pvm_image`、`pvm_flavor`、`pvm_intsubnet`、`pvm_router`、和 `pvm_sec` 创建 1 台主机 `pvm1`。输出虚拟机的名称、ID 和详细信息。
4. 在 `controller` 节点的 `/root` 目录下按要求编写 Python 程序 `attach_lvm.py` 文件，对接

openstack api, 要求将 pvm_cdisk 卷附加到云主机 pvm1。输出对应云主机名称、ID 和详细信息。

5. 在 controller 节点的 /root 目录下编写 Python 程序 create_user.py 文件, 对接 openstack api, 创建一个用户, 描述为 “API create user!” (如果存在同名用户, 代码中需先进行删除操作)。

任务 4 OpenStack 云平台运维开发 (10 分, 本任务只公布考试范围, 不公布赛题)

此任务包含多个子任务, 内容涉及编写 Python 脚本调用 OpenStack API 对 OpenStack 云平台进行管理和运维。使用自动化运维工具 Ansible 对云主机进行批量系统配置、批量程序部署、批量运行命令等运维操作。

第二场次题目：容器云平台部署与运维（5）

说明：完成本任务需要两台安装了 CentOS7.5 操作系统的云主机 master 和 node，镜像包中有本次容器云平台部署和运维所需的所有文件。

云梦公司技术部产品开发上线周期长，用户规模大且版本更新频繁。新版本的每次上线，产品都要承受极大的压力。引入 CICD (Continuous Integration 持续集成、Continuous Delivery 持续交付) 和灰度发布成了公司的当务之急，研发团队决定搭建基于 Kubernetes 的 CICD 环境，希望基于这个平台来实现 DevOps 的部分流程，并基于 Kubernetes 实现业务系统的灰度发布。

为了能够让云梦公司开发的 web 应用系统产品能够缩减开发周期，快速迭代版本，并实现业务从老版本到新版本的平滑过渡，避免升级过程中出现的问题对用户造成的影响。研发部决定使用微服务架构，实现基于 Kubernetes 的容器化部署、CICD 和灰度发布。

节点规划如表 1 所示。

表 1 容器云平台节点规划

节点角色	主机名	VCPUS	内存	硬盘
Master、Harbor、CICD	master	8	12G	100G
Worker Node	node	4	8G	100G

任务 1 Docker CE 及私有仓库安装任务（5 分）

1. 在 master、node 各节点中分别安装 DockerCE 和 docker-compose。完成后提交 master 节点的用户名、密码和 IP 到答题框。

2. 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 master 节点的用户名、密码和 IP 到答题框。

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

- （1）容器名称：rabbitmq；镜像：rabbitmq:3.8.3-management；
- （2）设置 rabbitmq 默认用户及密码均为 root；
- （3）设置容器重启策略为 always；
- （4）启用 rabbitmq 管理插件。

完成后编排部署 rabbitmq，并提交 master 节点的用户名、密码和 IP 到答题框。

任务 2 基于容器的 web 应用系统部署任务（10 分）

将云梦公司开发的基于微服务架构的 ChinaSkillsSA 智能文本的情绪分析（Sentiment Analysis）的系统实现全容器化部署，ChinaSkillsSA 智能分析系统应用系统架构如图 1 所示，ChinaSkillsSA 智能分析系统说明如表 2 所示。

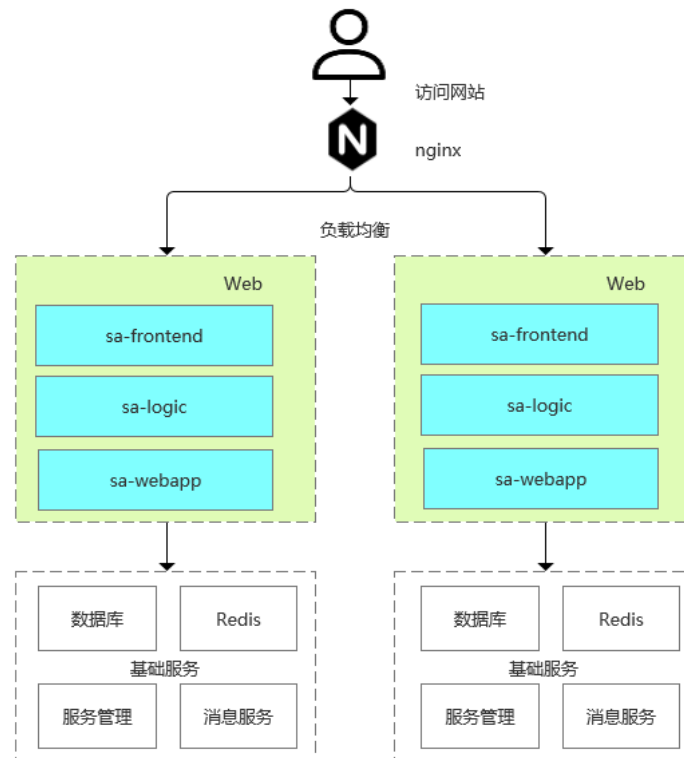


图 1 ChinaSkillsSA 智能分析系统架构图

表 2 ChinaSkillsSA 智能分析系统说明

模块	使用技术	说明
sa-frontend	Node.js	前端项目，使用 Vue.js 开发。
sa-logic	Python Flask	输入文本句子的情绪分析微服务，使用 Python 开发。
sa-webapp	Spring Cloud	Web 服务，处理 sa-frontend 请求，调用 sa-logic 微服务，使用 Java 语言开发。

将 MySQL 数据库组件、Redis 组件、Kafka 组件、Zookeeper 组件按照要求进行容器化。

1. 在 master 节点上编写 /root/mall-swarm/Dockerfile-redis 文件构建

chinaskillmall-redis:v1.1 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装 redis 服务;
- (4) 修改配置文件中的 bind 127.0.0.1 为 bind 0.0.0.0;
- (5) 设置 redis 免密，并关闭保护模式;
- (6) 开放端口：6379;
- (7) 设置服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写 /root/mall-swarm/Dockerfile-mariadb 文件构建 chinaskillmall-mariadb:v1.0 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装并初始化 mariadb，密码：123456;
- (4) 设置数据库默认字符编码为 UTF-8;
- (5) 创建数据库 gpmall 并导入 gpmall.sql;
- (6) 设置 mariadb 开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

3. 在 master 节点上编写 /root/gpmall/Dockerfile-zookeeper 文件构建 chinaskillmall-zookeeper:v1.1 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装 zookeeper 服务;
- (4) 开放端口 2181;
- (5) 设置 zookeeper 服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

4. 在 master 节点上编写 /root/gpmall/Dockerfile-kafka 文件构建 chinaskillmall-kafka:v1.1

镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装 kafka 服务;
- (4) 开放端口 9092;
- (5) 设置 zookeeper 服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

5. 在 master 节点上编写/root/gpmall/Dockerfile-nginx 文件构建 chinaskillmall-nginx:v1.1

镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装 nginx 服务;
- (4) 将提供的软件包 dist.tar 和 jar 包拷贝进镜像;
- (5) 设置 nginx 服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

7. 在 master 节点上编写/root/gpmall/gpmall.yaml 文件，具体要求为

- (1) 在 default 命名空间下部署该服务;
- (2) service 名称：chinaskillmall;
- (3) 以 NodePort 方式对外暴露端口，端口映射：30080:80;

完成后编排部署 ChinaSkillMall 商城。然后提交 master 节点的用户名、密码和 IP 到答题框。

任务 3 基于容器的持续集成部署任务（10 分）

该公司决定采用 GitLab + Jenkins 来构建 CICD 环境，以缩短新功能开发上线周期，及时满足客户的需求，实现 DevOps 的部分流程，来减轻部署运维的负担，实现可视化容器生命周期管理、应用发布和版本迭代更新，请完成 GitLab + Jenkins + Kubernetes 的 CICD 环境部署。CICD 应用系统架构如图 2 所示。

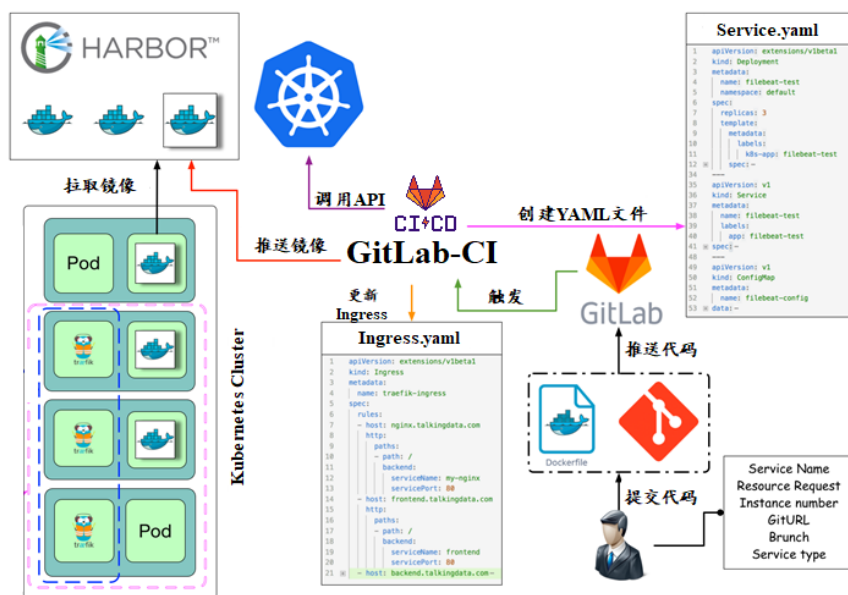


图 2 CI/CD 应用系统架构图

1. 在 master 节点上编写/root/jenkins/docker-compose.yaml 文件编排部署 Jenkins 服务，具体要求为

- (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: springcloud; 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 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写/root/gitlab/docker-compose.yaml 文件编排部署 Gitlab 服务，具体要求为

- (1) 容器名称: gitlab;
- (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 Springcloud, 将/opt/Springcloud 中的代码上传到 Springcloud 项目中。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

7. 配置 Jenkins 连接 Gitlab, 具体要求为

- (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 master 节点的用户名、密码和 IP 到答题框。

4.配置 Jenkins 连接 maven, 具体要求为

- (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 master 节点的用户名、密码和 IP 到答题框。

5.配置 CI/CD, 具体要求为

- (1) 新建一个流水线任务 Springcloud;
- (2) 编写流水线脚本, 构建 Springcloud 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 springcloud 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 springcloud。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

10. 将 Springcloud 项目中的代码上传到 Gitlab 中触发构建, 完成后提交 master 节点的用户名、密码和 IP 到答题框。

任务 4 Kubernetes 容器云平台部署与运维 (10 分, 本任务只公布考试范围, 不公布赛题)

当应用上线以后, 运维面临的一大挑战是如何能够在不影响已上线业务的情况下进行升级, 通过逐渐调整用户量, 进行新服务的测试和升级, 对服务异常能够快速回滚。公司决定采用基于 Istio 的灰度发布来实现业务从老版本到新版本的平滑过渡, 并避免升级过程中出现的问题对用户造成的影响。

请根据要求，完成 Kubernetes 的集群运维、调度、网络、存储、安全、应用生命周期管理、日志/监控等运维任务。

第三场次题目：公有云部署与运维（5）

云梦是一家科技公司，现在公司开发全面转向云原生，因此需要对公司内部项目搭建基于云原生的 Devops 相关服务。

某公有云是一家成熟的公有云提供商，为客户提供 K8S、云网络、云硬盘、云防火墙、负载均衡等服务，可创建 Web 服务，共享文件存储服务，数据库服务，数据库集群等服务。

根据上述公有云平台的特性，完成公有云中的各项运维工作。

任务 1 基础设施构建（10 分）

1.在公有云中完成虚拟私有云的创建，具体要求为

- （1）在贵阳区域进行创建操作；
- （2）创建一个名为 intnetX 的虚拟私有云：IP 地址为 172.16.1.0/16；
- （3）创建子网名称为 intnetX-server：IP 地址为 172.16.1.0/24；
- （4）创建一个名为 net-data 的虚拟私有云：网段为 192.168.1.0/16
- （5）创建子网名称为 net-mysql：IP 地址为 192.168.1.0/24；

完成创建后，提交当前的 AK（SecretId）、SK（SecretKey）和 intnetX 网络的 ID 到答题框。

2.创建对等连接 intnetX-net，配置 intnetX 虚拟私有云和 net-data 虚拟机私有云连通。操作完成后，提交当前的 AK（SecretId）、SK（SecretKey）和对等连接的 ID 到答题框。

3.登录华为云平台，创建两台云实例，具体要求为

- （1）计费模式：按需计费；
- （2）地域：贵阳；
- （3）CPU 架构：鲲鹏计算；
- （4）规格：kc1.xlarge.2；
- （5）镜像：CentOS 7.5 64 位；
- （6）系统盘：高 IO 50G 硬盘；
- （7）虚拟网络：intnetX-server；
- （8）实例名称：ChinaSkill-node-1、ChinaSkill-node-2；
- （9）登录方式：使用密码登录，密码自定义。

创建完成后，提交当前的 AK（SecretId）、SK（SecretKey）和 ChinaSkill-node-1 的实例 ID 到答题框。

4.创建一台云数据库，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) 数据库版本：MySQL 5.7；
- (4) 实例类型：主备；
- (5) 实例规格：鲲鹏 2 核 8GB/40GB；
- (6) 虚拟私有云：net-mysql；
- (7) 字符集：UTF8；
- (8) 端口：3306；
- (9) 数据库密码：Root123456；
- (10) 数据库名称：chinaskill-mysql；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和云数据库的实例 ID 到答题框。

5.使用公有云平台，申请对象存储服务，存储桶名字自定义。自行编写一个静态网页，网页的内容为“welcome to China”，使用申请的对象存储服务，托管该静态网页。托管成功后，提交你对象存储的访问域名到答题框。

6.创建一个网络 ACL，具体要求为

- (1) 名称：intnetX-acl；
- (2) 允许策略：只放行源地址为 172.16.1.0/24 访问 3306 端口；
- (3) 允许策略：只放行源地址为 172.16.1.0/24 访问 2181 端口；
- (4) 允许策略：只放行源地址为 172.16.1.0/24 访问 9092 端口；
- (5) 关联子网：关联网络 ACL 至 net-mysql 子网中；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和网络 ACL ID 到答题框。

7.创建一个 Redis 缓存实例，具体要求为

- (1) 版本号：5.0；
- (2) 实例类型：主备；
- (3) 副本数：2；
- (4) 实例规格：4G；
- (5) 子网：net-mysql；
- (6) 密码：Root123456；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和实例 ID 到答题框。

8.根据 ChinaSkill-node-1 云服务器配置购买 chinaskill-node-3 云服务器上，此服务器使用 net-mysql 私有网络。使用提供的软件包安装 kafka 和 zookeeper 服务。安装完成后运行服务。完成后提交连接 chinaskill-node-3 节点的用户名、密码和公网 IP 地址到答题框。

9.在 ChinaSkill-node-1 和 ChinaSkill-node-2 云服务上部署 gpmall，mysql 使用云数据库地址，配置公有云 Redis 访问路径，使用提供的公有云软件包 gpmall-package 完成商城的上云操作。并设置 gpmall 所需 jar 软件包开启自动启动。完成后提交 ChinaSkill-node-1 的用户名、密码和公网 IP 到答题框。

任务 2 系统管理与维护（10 分）

1.使用所创建的私有镜像 chinaskill-image 购买云服务器 ChinaSkill-node-3，配置与 ChinaSkill-node-1 云服务器相同。完成后提交 chinaskill-node-3 的用户名、密码和公网 IP 到答题框。

2.创建一个负载均衡器 chinaskill-elb，将 ChinaSkill-node-1 和 ChinaSkill-node-3 加入负载均衡的后端。设置一个可用的公网服务 IP，服务端口为 80。配置监听器，监听 80 端口。对浮动公共 IP 进行 Web 访问测试。完成后，提交当前的 AK（SecretId）、SK（SecretKey）和负载均衡器的 ID 到答题框。

3.新建一个启动配置，具体要求为

- (1) 启动配置名称：template-exam；
- (2) 计费模式：按量计费；
- (3) 地域：贵阳；
- (4) CPU 架构：鲲鹏计算；
- (5) 规格：kc1.xlarge.2；
- (6) 镜像：chinaskill-image；
- (7) 系统盘：高 IO 50G 硬盘；
- (8) 登录方式：使用密码登录，密码自定义。

创建一个伸缩组，具体要求为

- (1) 名称：as-exam；
- (2) 最小伸缩数：1；起始实例数：1；最大伸缩数：5；
- (3) 启动配置：template-exam；
- (4) 使用负载均衡：chinaskill-elb；

(5) 移出策略：移出最旧的实例。

为伸缩组 as-exam 新建告警触发策略，具体要求为

(1) 如果实例的内存利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例数减少 1 台。冷却 60 秒；

(2) 如果实例的内存利用率在 5 分钟内的最大值大于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒；

(3) 如果实例的 CPU 利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例数减少 1 台。冷却 60 秒；

(4) 如果实例的 CPU 利用率在 5 分钟内的平均值大于等于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒。

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和伸缩组的实例 ID 到答题框。

4. 创建一个鲲鹏架构集群，具体要求为

(1) 集群名称：kcloud；

(2) 集群版本：v1.15.11；

(3) 地域：贵阳；

(4) 集群管理规模：50 节点；

(5) 控制节点数：3；

(6) 节点使用子网：intnetX-server；

(7) Pod 实例上限：64；

(8) 容器网段：10.10.0.0/16。

节点配置信息要求为

(1) 节点名称：kcloud-server；

(2) 节点规格：kc.xlarge.2

(3) 节点：EulerOS 2.8

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和集群的 ID 到答题框。

5. 使用插件管理在 kcloud 容器集群中安装 dashboard 可视化监控界面。完成后，提交 dashboard 访问地址到答题框。

6. 购买云硬盘存储卷，具体配置为

(1) 名称为：ckloud-disk；

(2) 集群：kcloud；

(3) 容量: 30G;

(4) 类型: 高 I/O;

完成后, 创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和云硬盘的 ID 到答题框。

7.在 kcloud 集群中安装 kubectl 命令, 使用 kubectl 命令管理 kcloud 集群。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

8.使用提供的 helm 软件包, 在 kcloud 集群中安装 helm 服务。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

9.使用容器镜像服务作为镜像仓库, 将提供的 httpd-2.4.33 和 httpd-2.4.35 镜像上传至服务中。提交当前的 AK (SecretId)、SK (SecretKey) 和组织名称到答题框。

10.在 kcloud 集群中创建一个 Deployment, 使用所上传的 httpd-2.4.33 容器镜像, 配置使其可以通过外部公网进行访问。完成后提交外部访问地址到答题框。

任务 3 自动化运维 (10 分, 本任务只公布考试范围, 不公布赛题)

编写 Python 脚本调用公有云 API 完成公有云运维工作。

第一场次题目：OpenStack 平台部署与运维（6）

某企业拟使用 OpenStack 搭建一个企业云平台，以实现资源池化弹性管理、企业应用集中管理、统一安全认证和授权等管理。

系统架构如图 1 所示，IP 地址规划如表 1 所示。

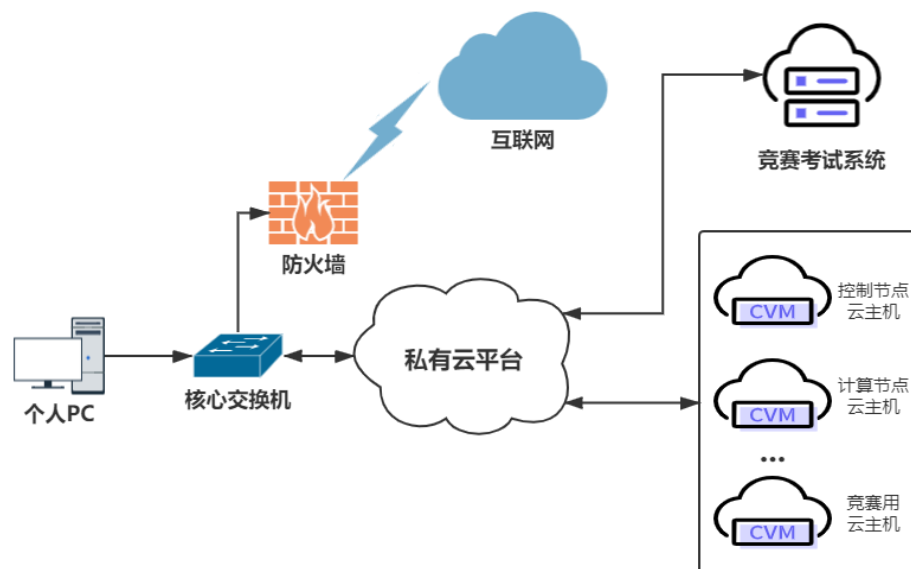


图 1 系统架构图

表 1 IP 地址规划

设备名称	主机名	接口	IP 地址	说明
云服务器1	Controller	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器2	Compute	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器3	自定义	eth0	172.129.x.0/24	用于实操题
云服务器 n				
PC-1		本地连接	172.24.16.0/24	PC 使用

说明：

- 1.竞赛使用集群模式进行，比赛时给每个参赛队提供独立的租户与用户，各用户的资源配额相同，选手通过用户名与密码登录竞赛用私有云平台，创建云主机进行相应答题；
- 2.表中的 x 为工位号；在进行 OpenStack 搭建时的第二块网卡地址根据题意自行创建；
- 3.根据图表给出的信息，检查硬件连线及网络设备配置，确保网络连接正常；
- 4.考试所需要的账号资源、竞赛资源包与附件均会在考位信息表与设备确认单中给出；
- 5.竞赛过程中，为确保服务器的安全，请自行修改服务器密码；在考试系统提交信息时，请确认自己的 IP 地址，用户名和密码。

任务 1 基础运维任务（5 分）

1.使用提供的用户名密码，登录提供的 OpenStack 私有云平台，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；

完成后提交控制节点的用户名、密码和 IP 地址到答题框。

2.使用提供的用户名密码，登录提供的 OpenStack 私有云平台，自行使用 CentOS7.5 镜像创建一台云主机，flavor 使用 2v_4G_50G 的配置，使用单网卡。启动后使用提供的 ansible.tar.gz 软件包在这个节点上安装 ansible 服务并配置 ansible 节点与 controller、compute 节点的 hosts 主机名映射。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

3.配置 ansible 节点无秘钥连接 controller 节点和 compute 节点，配置完成后并完成 ssh 连接两个节点的 hostname 进行测试。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

4.在 ansible 节点配置 ansible 的 hosts 文件，要求创建两个组分别为 controller 和 compute，controller 组下主机节点为 controller 节点；compute 组下主机节点为 compute。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

5.在 compute 节点上利用空白分区划分 2 个 20G 分区。完成后提交计算节点的用户名、密码和 IP 地址到答题框。

任务 2 OpenStack 搭建任务（10 分）

1.使用提供的 openstack_ansible.tar.gz 项目包解压至 ansible 节点的/opt 目录下，然后编辑 roles 目录下 init/tasks 中的 main.yaml；编辑 group_vars 目录下的 all 文件（openstack 中的密码都设置为 000000）；编辑 install_openstack.yaml 文件，要求执行 install_openstack.yaml 文件可以在 controller 节点和 compute 节点执行 init 这个 role 来安装 iaas-pre-host。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。（考试系统会进入你的 ansible 节点来执行 install_openstack.yaml，请确保你的环境处于正确的可执行状态）

2.在 ansible 节点，编辑 roles/mariadb/tasks 下的 main.yaml 文件和 install_openstack.yaml，在 controller 节点上安装数据库服务。自行执行 install_openstack.yaml 文件安装数据库服务。

3.在 ansible 节点，编辑 roles/keystone/tasks 下的 main.yaml 文件和 install_openstack.yaml，

在 controller 节点上安装 keystone 服务。自行执行 install_openstack.yaml 文件安装 keystone 服务。

4.在 ansible 节点, 编辑 roles/glance/tasks 下的 main.yaml 文件和 install_openstack.yaml, 在 controller 节点上安装 glance 服务。自行执行 install_openstack.yaml 文件安装 glance 服务。

5.在 ansible 节点, 编辑 roles/nova-controller/tasks 下的 main.yaml 文件、roles/nova-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 nova 服务。

6.在 ansible 节点, 编辑 roles/neutron-controller/tasks 下的 main.yaml 文件、roles/neutron-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 neutron 服务。

7.在 ansible 节点, 编辑 roles/dashboard/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点上安装 dashboard 服务。

8.在 ansible 节点, 编辑 roles/swift-controller/tasks 下的 main.yaml 文件、roles/swift-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 swift 服务。

9.在 ansible 节点, 编辑 roles/cinder-controller/tasks 下的 main.yaml 文件、roles/cinder-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 cinder 服务。

10.在 ansible 节点, 编辑 roles/heat/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点上安装 heat 服务。

任务 3 OpenStack 云平台运维 (10 分)

1.在自行搭建的 OpenStack 平台上,使用命令创建一个名为 Fmin, ID 为 1,内存为 1024MB, 磁盘为 10GB, vcpu 数量为 1 的云主机类型。

2.在自行搭建的 OpenStack 平台上, 创建云主机网络 extnet, 子网 extsubnet, 虚拟机网段为 192.168.100.0/24, 网关为 192.168.100.1, 段 ID 默认写 100, 网络使用 vlan 模式。

3.在自行搭建的 OpenStack 平台上,基于“cirros”镜像、1vCPU/1G /10G 的 flavor、extsubnet 的网络, 创建一台虚拟机 VM1, 启动 VM1。

4.在自己搭建的 OpenStack 平台上, 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 cscvm.qcow2。

5.在自己搭建的 OpenStack 平台上, 使用 cinder 服务, 创建一个名为“lvm”的卷类型, 创建 1 块卷类型为 lvm 的 1G 云盘, 并附加到虚拟机 VM1 上。

6.在自己搭建的 OpenStack 平台, 通过修改相关参数对 openstack 平台进行调优操作, 相应的调优操作有:

- (1)预留前 2 个物理 CPU,把后面的所有 CPU 分配给虚拟机使用(假设 vcpu 为 16 个);
- (2) 设置 cpu 超售比例为 4 倍;

7.登录提供的私有云平台，创建一台 centos7.5 的云主机，flavor 使用带有附加硬盘的类型。连接到该云主机，使用附加的硬盘，划分 4 个 10G 的分区，使用这 4 个分区创建一个 raid5 级别的磁盘阵列，其中 1 个分区作为热备盘。

8.登录提供的私有云平台，创建两台 centos7.5 的云主机，使用提供的软件包，将这两台云主机上安装 MariaDB 数据库服务，并配置为主从数据库。

9.登录提供的私有云平台，创建一台 centos7.5 的云主机，使用提供的软件包，在这台云主机上安装 LNMP 环境，并将提供的 WordPress 案例包部署上去。

10.登录提供的私有云平台，使用镜像 OpenStack-error-1 创建云主机“all-in-one”(账号：root;密码：000000)。创建后的云主机内有错误的 OpenStack 平台，其错误为无法使用 glance 镜像服务，请排查错误，使 glance 服务变的可用。

任务 4 OpenStack 云平台运维开发（10 分，本任务只公布考试范围，不公布赛题）

此任务包含 2-4 个子任务，内容涉及编写 Python 脚本调用 OpenStack API 对 OpenStack 云平台进行管理和运维。使用自动化运维工具 Ansible 对云主机进行批量系统配置、批量程序部署、批量运行命令等运维操作。

第二场次题目：容器云平台部署与运维（6）

说明：完成本任务需要两台安装了 CentOS7.5 操作系统的云主机 master 和 node，镜像包中有本次容器云平台部署和运维所需的所有文件。

云梦公司技术部产品开发上线周期长，用户规模大且版本更新频繁。新版本的每次上线，产品都要承受极大的压力。引入 CICD (Continuous Integration 持续集成、Continuous Delivery 持续交付) 和灰度发布成了公司的当务之急，研发团队决定搭建基于 Kubernetes 的 CICD 环境，希望基于这个平台来实现 DevOps 的部分流程，并基于 Kubernetes 实现业务系统的灰度发布。

为了能够让云梦公司开发的 web 应用系统产品能够缩减开发周期，快速迭代版本，并实现业务从老版本到新版本的平滑过渡，避免升级过程中出现的问题对用户造成的影响。研发部决定使用微服务架构，实现基于 Kubernetes 的容器化部署、CICD 和灰度发布。

节点规划如表 1 所示。

表 1 容器云平台节点规划

节点角色	主机名	VCPUS	内存	硬盘
Master、Harbor、CICD	master	8	12G	100G
Worker Node	node	4	8G	100G

任务 1 Docker CE 及私有仓库安装任务（5 分）

1. 在 master 节点中使用提供的脚本完成 Docker CE、docker-compose 以及 Harbor 仓库的安装，导入/opt/images 目录下的所有镜像，并推送到私有仓库。
2. 在 master、node 节点完成 Kubernetes 集群的安装。
3. 在 node 节点上从仓库中拉取 redis:latest 镜像，创建 docker-compse.yaml 文件，编排部署 Redis 服务，并设置 restart 策略。

任务 2 基于容器的 web 应用系统部署任务（10 分）

将云梦公司开发的基于微服务架构的 web 应用系统 ChinaSkillsMall 重新实现全容器化部署，原微服务架构的 Web 应用系统架构如图 1 所示，模块功能介绍图表 2 所示。

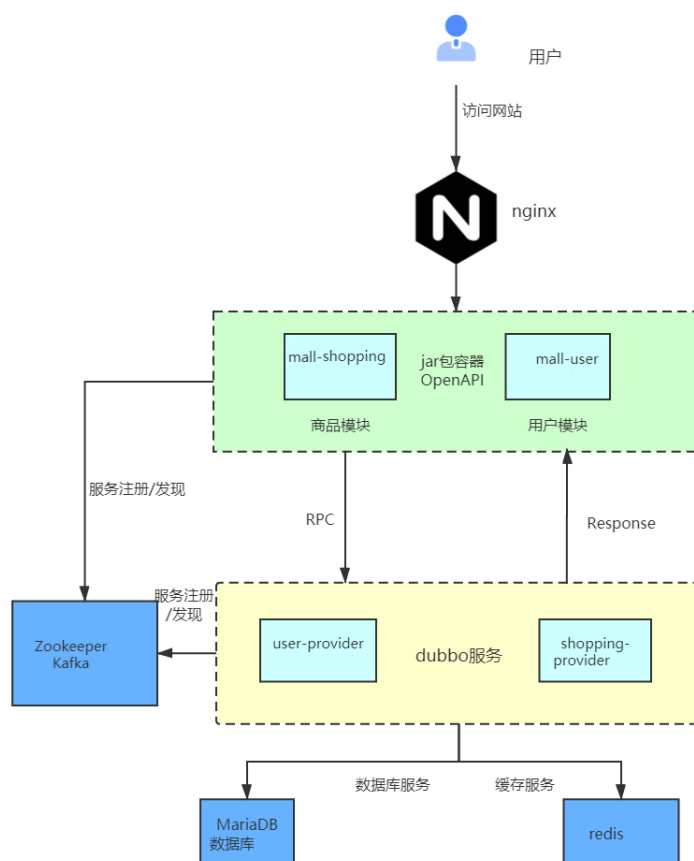


图 1 ChinaSkillsMall 系统架构图

表 2 ChinaSkillsMall 模块功能介绍

模块	使用技术	备注
mall.sql	MySQL	网站的数据库
dist 目录	Nginx	网站的前端项目
mall-shopping	web 项目	8081 端口，商品/购物车/首页渲染等交互
mall-user		8082 端口，提供用户相关的交互，如登录、注册、个人中心等
user-provider	后端服务	提供用户相关服务
shopping-provider		提供购物车、推荐商品、商品等服务

将 MySQL 数据库组件、Redis 组件、Kafka 组件、Zookeeper 组件按照要求进行容器化。

1. 在 master 节点上编写 /root/mall-swarm/Dockerfile-redis 文件构建 chinaskillmall-redis:v1.1 镜像，具体要求为

(1) 基础镜像: centos:centos7.5.1804;

- (2) 作者: Chinaskill;
- (3) 安装 redis 服务;
- (4) 修改配置文件中的 bind 127.0.0.1 为 bind 0.0.0.0;
- (5) 设置 redis 免密, 并关闭保护模式;
- (6) 开放端口: 6379;
- (7) 设置服务开机自启。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写 /root/mall-swarm/Dockerfile-mariadb 文件构建 chinaskillmall-mariadb:v1.0 镜像, 具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装并初始化 mariadb, 密码: 123456;
- (4) 设置数据库默认字符编码为 UTF-8;
- (5) 创建数据库 gpmall 并导入 gpmall.sql;
- (6) 设置 mariadb 开机自启。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

3. 在 master 节点上编写 /root/gpmall/Dockerfile-zookeeper 文件构建 chinaskillmall-zookeeper:v1.1 镜像, 具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 zookeeper 服务;
- (4) 开放端口 2181;
- (5) 设置 zookeeper 服务开机自启。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

4. 在 master 节点上编写 /root/gpmall/Dockerfile-kafka 文件构建 chinaskillmall-kafka:v1.1 镜像, 具体要求为

- (1) 基础镜像: centos:centos7.5.1804;

- (2) 作者: Chinaskill;
- (3) 安装 kafka 服务;
- (4) 开放端口 9092;
- (5) 设置 zookeeper 服务开机自启。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

5. 在 master 节点上编写/root/gpmall/Dockerfile-nginx 文件构建 chinaskillmall-nginx:v1.1 镜像, 具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 nginx 服务;
- (4) 将提供的软件包 dist.tar 和 jar 包拷贝进镜像;
- (5) 设置 nginx 服务开机自启。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

8. 在 master 节点上编写/root/gpmall/gpmall.yaml 文件, 具体要求为

- (1) 在 default 命名空间下部署该服务;
- (2) service 名称: chinaskillmall;
- (3) 以 NodePort 方式对外暴露端口, 端口映射: 30080:80;

完成后编排部署 ChinaSkillMall 商城。然后提交 master 节点的用户名、密码和 IP 到答题框。

任务 3 基于容器的持续集成部署任务 (10 分)

该公司决定采用 GitLab + Jenkins 来构建 CICD 环境, 以缩短新功能开发上线周期, 及时满足客户的需求, 实现 DevOps 的部分流程, 来减轻部署运维的负担, 实现可视化容器生命周期管理、应用发布和版本迭代更新, 请完成 GitLab + Jenkins + Kubernetes 的 CICD 环境部署。CICD 应用系统架构如图 2 所示。

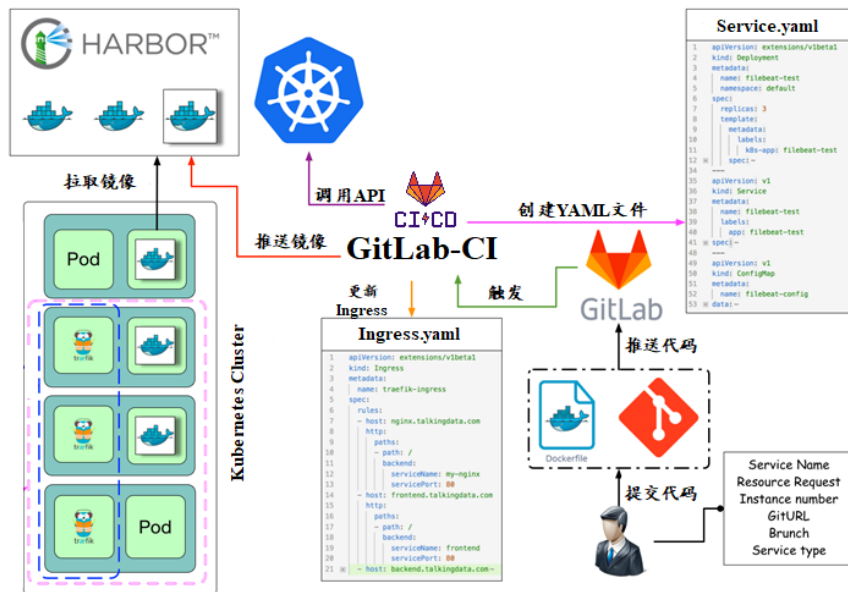


图 2 CI/CD 应用系统架构图

1. 在 master 节点上编写/root/jenkins/docker-compose.yaml 文件编排部署 Jenkins 服务，具体要求为

- (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: gpmall; 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 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写/root/gitlab/docker-compose.yaml 文件编排部署 Gitlab 服务，具体要求为

- (1) 容器名称: gitlab;
- (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 gpmall, 将 gpmall 中的代码上传到 gpmall 项目中;

(7) 实现通过 SSH 链接克隆项目。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

3. 配置 Jenkins 连接 Gitlab, 具体要求为

(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 master 节点的用户名、密码和 IP 到答题框。

4. 配置 CI/CD, 具体要求为

(1) 新建一个流水线任务 gpmall;

(2) 编写流水线脚本, 构建 gpmall 项目中的商城服务, 将构建后的镜像自动上传到 Harbor 仓库的 gpmall 项目, 并自动发布服务到 Kubernetes 集群的 default 命名空间下;

(3) 配置 Webhook;

(4) 在 Harbor 中新建公开项目 gpmall。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

5. 将 gpmall 项目中的 gpmall.yaml 文件补全, 然后将 gpmall 项目代码重新上传到 Gitlab 中触发构建, 完成后提交 master 节点的用户名、密码和 IP 到答题框。

任务 4 Kubernetes 容器云平台部署与运维 (10 分, 本任务只公布考试范围, 不公布赛题)

当应用上线以后, 运维面临的一大挑战是如何能够在不影响已上线业务的情况下进行升级。在无法百分百避免版本升级故障的情况下, 需要通过一种方式进行可控的版本发布, 把故障影响控制在可以接受的范围内, 并可以快速回退。公司决定采用基于 Istio 的灰度发布 (又名金丝雀发布) 来实现业务从老版本到新版本的平滑过渡, 并避免升级过程中出现的问题对用户造成的影响。

请根据要求, 完成 Kubernetes 的集群运维、调度、网络、存储、安全、应用生命周期

管理、日志/监控等运维任务。

第三场次题目：公有云部署与运维（6）

云梦是一家科技公司，现在公司开发全面转向云原生，因此需要对公司内部项目搭建基于云原生的 Devops 相关服务。

某公有云是一家成熟的公有云提供商，为客户提供 K8S、云网络、云硬盘、云防火墙、负载均衡等服务，可创建 Web 服务，共享文件存储服务，数据库服务，数据库集群等服务。

根据上述公有云平台的特性，完成公有云中的各项运维工作。

任务 1 基础设施构建（10 分）

1.登录华为云平台，创建两台云实例，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) CPU 架构：鲲鹏计算；
- (4) 规格：kc1.xlarge.2；
- (5) 镜像：CentOS 7.5 64 位；
- (6) 系统盘：高 IO 50G 硬盘；
- (7) 公网带宽：按带宽计费，5Mbps；
- (8) 实例名称：ChinaSkill-node-1、ChinaSkill-node-2；
- (9) 登录方式：使用密码登录，密码自定义。

创建完成后，提交当前的 AK（SecretId）、SK（SecretKey）和 ChinaSkill-node-1 的实例 ID 到答题框。

2.在公有云中完成虚拟私有云的创建，具体要求为

- (1) 在贵阳区域进行创建操作；
- (2) 创建一个名为 intnetX 的内部网络：IP 地址为 172.16.1.0/16；
- (3) 创建子网名称为 intnetX-server：IP 地址为 172.16.1.0/24；
- (4) 创建子网名称为 intnetX-mysql：IP 地址为 172.16.2.0/24；

完成创建后，提交当前的 AK（SecretId）、SK（SecretKey）和 intnetX 网络的 ID 到答题框。

3.使用创建的 ChinaSkill-node-1 和 ChinaSkill-node-2 进行切换内网操作，将这两个云主机的内网，切换到 intnetX 网络的 intnetX-server 子网。切换完成后，保证该云主机还能够正常上网。操作完成后，提交 ChinaSkill-node-1 节点的用户名、密码和公网 IP 地址到答题框。

4.创建一台云数据库，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) 数据库版本：MySQL 5.7；
- (4) 实例类型：主备；
- (5) 实例规格：鲲鹏 2 核 8GB/40GB；
- (6) 虚拟私有云：intnetX-mysql；
- (7) 字符集：UTF8；
- (8) 端口：3306；
- (9) 数据库密码：Root123456；
- (10) 数据库名称：chinaskill-mysql；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和云数据库的实例 ID 到答题框。

5.创建一个安全组，具体要求为

- (1) 名称：intnetX-security；
- (2) 允许策略：只放行源地址为 172.16.1.0/24 访问 3306 端口；
- (3) 允许策略：只放行源地址为 172.16.1.0/24 使用 ping 命令访问；
- (4) 关联实例：将 intnetX-security 安全组关联至所创建的数据库中；

6.使用提供的软件包，在 ChinaSkill-node-1 和 ChinaSkill-node-2 中部署 WordPress 的 Web 服务即 LNMP+WordPress。使用云数据库 chinaskill-mysql 为后端数据库。完成 WordPress 系统上云后，将 ChinaSkill-node-1 节点的 WordPress 访问地址提交到答题框。

7.创建一个负载均衡器 chinaskill-elb，将 ChinaSkill-node-1 和 ChinaSkill-node-2 加入负载均衡的后端。设置一个可用的公网服务 IP，服务端口为 80。配置监听器，监听 80 端口。对浮动公共 IP 进行 Web 访问测试。完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和负载均衡器的 ID 到答题框。

8.创建一个云服务器备份存储库名为 server_backup，容量为 100G。完成后，将当前的 AK (SecretId)、SK (SecretKey) 和存储库 ID 到答题框。

9.创建一个公网 NAT 网关，具体配置为

- (1) 名称为：kcloud-nat；
- (2) 虚拟私有云：intnetX；

- (3) 子网: `intnetX-server`;
- (4) 规则: 内部子网地址访问外网;
- (5) 保证使用 `intnetX-server` 云服务器可以使用 `kcloud-nat` 网关访问外网。

创建完成后提交当前的 AK (`SecretId`)、SK (`SecretKey`) 和 NAT 网关的 ID 到答题框。

任务 2 系统管理与维护 (10 分)

1. 创建一个负载均衡器 `chinaskill-elb`, 将 `ChinaSkill-node-1` 和 `ChinaSkill-node-2` 加入负载均衡的后端。设置一个可用的公网服务 IP, 服务端口为 80。配置监听器, 监听 80 端口。对浮动公共 IP 进行 Web 访问测试。完成后, 提交当前的 AK (`SecretId`)、SK (`SecretKey`) 和负载均衡器的 ID 到答题框。

2. 创建一个云服务器备份存储库名为 `server_backup`, 容量为 100G。将 `ChinaSkill-node-1` 云服务器制作镜像文件 `chinaskill-image`。完成后, 将当前的 AK (`SecretId`)、SK (`SecretKey`) 和存储库 ID 到答题框。

3. 新建一个启动配置, 具体要求为

- (1) 启动配置名称: `template-exam`;
- (2) 计费模式: 按量计费;
- (3) 地域: 贵阳;
- (4) CPU 架构: 鲲鹏计算;
- (5) 规格: `kc1.xlarge.2`;
- (6) 镜像: `chinaskill-image`;
- (7) 系统盘: 高 IO 50G 硬盘;
- (8) 登录方式: 使用密码登录, 密码自定义。

创建一个伸缩组, 具体要求为

- (1) 名称: `as-exam`;
- (2) 最小伸缩数: 1; 起始实例数: 1; 最大伸缩数: 5;
- (3) 启动配置: `template-exam`;
- (4) 使用负载均衡: `chinaskill-elb`;
- (5) 移出策略: 移出最旧的实例。

为伸缩组 `as-exam` 新建告警触发策略, 具体要求为

- (1) 如果实例的内存利用率在 5 分钟内的最大值小于 40%, 且连续发生 3 次。则实例

数减少 1 台。冷却 60 秒；

(2) 如果实例的内存利用率在 5 分钟内的最大值大于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒；

(3) 如果实例的 CPU 利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例数减少 1 台。冷却 60 秒；

(4) 如果实例的 CPU 利用率在 5 分钟内的平均值大于等于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒。

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和伸缩组的实例 ID 到答题框。

4. 创建一个鲲鹏架构集群，具体要求为

- (1) 集群名称: kcloud;
- (2) 集群版本: v1.15.11;
- (3) 地域: 贵阳;
- (4) 集群管理规模: 50 节点;
- (5) 控制节点数: 3;
- (6) 节点使用子网: intnetX-server;
- (7) Pod 实例上限: 64;
- (8) 容器网段: 10.10.0.0/16。

节点配置信息要求为

- (1) 节点名称: kcloud-server;
- (2) 节点规格: kc.xlarge.2
- (3) 节点: EulerOS 2.8

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和集群的 ID 到答题框。

5. 使用插件管理在 kcloud 容器集群中安装 dashboard 可视化监控界面。完成后，提交 dashboard 访问地址到答题框。

6. 购买云硬盘存储卷，具体配置为

- (1) 名称为: ckcloud-disk;
- (2) 集群: kcloud;
- (3) 容量: 30G;
- (4) 类型: 高 I/O;

完成后，创建完成后提交当前的 AK（SecretId）、SK（SecretKey）和云硬盘的 ID 到答题框。

7.在 kcloud 集群中安装 kubectl 命令，使用 kubectl 命令管理 kcloud 集群。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

8.在 kcloud 集群中定义名为 liveness-http 探针和 liveness-tcp 探针，监控 kcloud 集群。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

9.使用容器镜像服务作为镜像仓库，将提供的 httpd-2.4.33 和 httpd-2.4.35 镜像上传至服务中。提交当前的 AK（SecretId）、SK（SecretKey）和组织名称到答题框。

10.在 kcloud 集群中创建一个 Deployment，使用开源镜像中心的 nginx:latest，配置使其可以通过外部公网进行访问。完成后提交外部访问地址到答题框。

任务 3 自动化运维（10 分，本任务只公布考试范围，不公布赛题）

编写 Python 脚本调用公有云 API 完成公有云运维工作。

第一场次题目：OpenStack 平台部署与运维（7）

某企业拟使用 OpenStack 搭建一个企业云平台，以实现资源池化弹性管理、企业应用集中管理、统一安全认证和授权等管理。

系统架构如图 1 所示，IP 地址规划如表 1 所示。

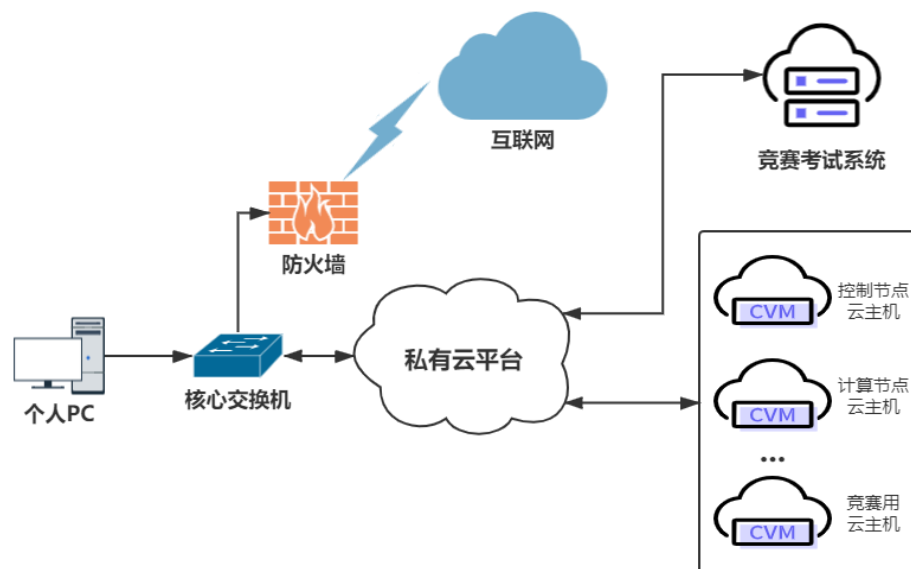


图 1 系统架构图

表 1 IP 地址规划

设备名称	主机名	接口	IP 地址	说明
云服务器 1	Controller	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器 2	Compute	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器 3	自定义	eth0	172.129.x.0/24	用于实操题
云服务器 n				
PC-1		本地连接	172.24.16.0/24	PC 使用

说明：

- 1.竞赛使用集群模式进行，比赛时给每个参赛队提供独立的租户与用户，各用户的资源配额相同，选手通过用户名与密码登录竞赛用私有云平台，创建云主机进行相应答题；
- 2.表中的 x 为工位号；在进行 OpenStack 搭建时的第二块网卡地址根据题意自行创建；
- 3.根据图表给出的信息，检查硬件连线及网络设备配置，确保网络连接正常；
- 4.考试所需要的账号资源、竞赛资源包与附件均会在考位信息表与设备确认单中给出；
- 5.竞赛过程中，为确保服务器的安全，请自行修改服务器密码；在考试系统提交信息时，请确认自己的 IP 地址，用户名和密码。

任务 1 基础运维任务（5 分）

- 1.使用提供的用户名密码，登录提供的 OpenStack 私有云平台，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；

完成后提交控制节点的用户名、密码和 IP 地址到答题框。

2.使用提供的用户名密码，登录提供的 OpenStack 私有云平台，自行使用 CentOS7.5 镜像创建一台云主机，flavor 使用 2v_4G_50G 的配置，使用单网卡。启动后使用提供的 ansible.tar.gz 软件包在这个节点上安装 ansible 服务并配置 ansible 节点与 controller、compute 节点的 hosts 主机名映射。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

3.配置 ansible 节点无秘钥连接 controller 节点和 compute 节点，配置完成后并完成 ssh 连接两个节点的 hostname 进行测试。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

4.在 ansible 节点配置 ansible 的 hosts 文件，要求创建两个组分别为 controller 和 compute，controller 组下主机节点为 controller 节点；compute 组下主机节点为 compute。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

5.在 compute 节点上利用空白分区划分 2 个 20G 分区。完成后提交计算节点的用户名、密码和 IP 地址到答题框。

任务 2 OpenStack 搭建任务（10 分）

1.使用提供的 openstack_ansible.tar.gz 项目包解压至 ansible 节点的/opt 目录下，然后编辑 roles 目录下 init/tasks 中的 main.yaml；编辑 group_vars 目录下的 all 文件（openstack 中的密码都设置为 000000）；编辑 install_openstack.yaml 文件，要求执行 install_openstack.yaml 文件可以在 controller 节点和 compute 节点执行 init 这个 role 来安装 iaas-pre-host。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。（考试系统会进入你的 ansible 节点来执行 install_openstack.yaml，请确保你的环境处于正确的可执行状态）

2.在 ansible 节点，编辑 roles/mariadb/tasks 下的 main.yaml 文件和 install_openstack.yaml，在 controller 节点上安装数据库服务。自行执行 install_openstack.yaml 文件安装数据库服务。

3.在 ansible 节点，编辑 roles/keystone/tasks 下的 main.yaml 文件和 install_openstack.yaml，在 controller 节点上安装 keystone 服务。自行执行 install_openstack.yaml 文件安装 keystone 服务。

4.在 ansible 节点，编辑 roles/glance/tasks 下的 main.yaml 文件和 install_openstack.yaml，

在 controller 节点上安装 glance 服务。自行执行 install_openstack.yaml 文件安装 glance 服务。

5.在 ansible 节点, 编辑 roles/nova-controller/tasks 下的 main.yaml 文件、roles/nova-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 nova 服务。

6.在 ansible 节点, 编辑 roles/neutron-controller/tasks 下的 main.yaml 文件、roles/neutron-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 neutron 服务。

7.在 ansible 节点, 编辑 roles/dashboard/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点上安装 dashboard 服务。

8.在 ansible 节点, 编辑 roles/swift-controller/tasks 下的 main.yaml 文件、roles/swift-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 swift 服务。

9.在 ansible 节点, 编辑 roles/cinder-controller/tasks 下的 main.yaml 文件、roles/cinder-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 cinder 服务。

10.在 ansible 节点, 编辑 roles/heat/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点上安装 heat 服务。

任务 3 OpenStack 云平台运维 (10 分)

1.在自行搭建的 OpenStack 平台上, 基于 cirros-0.3.4-x86_64-disk.img 镜像创建一个名为 cirros 的镜像, 并打标签为 small。

2.在自行搭建的 OpenStack 平台上, 编写 heat 模板 server.yml, 模板作用为创建名为 “m1.flavor”、ID 为 1234、内存为 1024MB、硬盘为 10GB、vcpu 数量为 2 的云主机类型。

3.在自行搭建的 OpenStack 平台上, 创建云主机网络 extnet, 子网 extsubnet, 虚拟机网段为 192.168.100.0/24, 网关为 192.168.100.1, 段 ID 默认写 100, 网络使用 vlan 模式。

4.在 controller 控制节点上, 安装 libguestfs-tools 工具的时候, 会发生依赖包的冲突, 请解决依赖关系的报错, 完成 libguestfs-tools 工具的安装。

5.在自行搭建的 OpenStack 平台上, 使用 cirros 镜像创建云主机, flavor 使用 1vcpu/512M 内存/1G 硬盘, 创建云主机 cscv_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/2G 硬盘。

6.在 controller 节点上创建名为 chinaskill 的容器, 并获取该容器的存放路径; 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。

7.登录提供的私有云平台, 使用 centos7.5 镜像创建三台云主机来搭建 rabbitmq 集群。使用普通集群模式, 其中一台做磁盘节点, 另外两台做内存节点, 配置完毕后启动 rabbitmq 服务。

8.登录提供的私有云平台，使用 centos7.5 镜像创建两台云主机，使用提供的软件包。在这两台云主机上安装 Redis 服务，并配置成 Redis 主从架构。

9.登录提供的私有云平台，创建一台 centos7.5 的云主机，flavor 使用带有附加硬盘的类型。连接到该云主机，使用附加的硬盘，要求分出两个大小为 5G 的分区。使用两个分区，创建名为 chinaskill-vg 的卷组。

10.登录提供的私有云平台，使用镜像 OpenStack-error-2 创建云主机“all-in-one”(账号：root；密码：000000)。创建后的云主机内有错误的 OpenStack 平台，错误现象为创建云主机报错，请排除错误，使云平台可以正常使用。

任务 4 OpenStack 云平台运维开发（10 分，本任务只公布考试范围，不公布赛题）

此任务包含 2-4 个子任务，内容涉及编写 Python 脚本调用 OpenStack API 对 OpenStack 云平台进行管理和运维。使用自动化运维工具 Ansible 对云主机进行批量系统配置、批量程序部署、批量运行命令等运维操作。

第二场次题目：容器云平台部署与运维（7）

说明：完成本任务需要两台安装了 CentOS7.5 操作系统的云主机 master 和 node，镜像包中有本次容器云平台部署和运维所需的所有文件。

云梦公司技术部产品开发上线周期长，用户规模大且版本更新频繁。新版本的每次上线，产品都要承受极大的压力。引入 CICD (Continuous Integration 持续集成、Continuous Delivery 持续交付) 和灰度发布成了公司的当务之急，研发团队决定搭建基于 Kubernetes 的 CICD 环境，希望基于这个平台来实现 DevOps 的部分流程，并基于 Kubernetes 实现业务系统的灰度发布。

为了能够让云梦公司开发的 web 应用系统产品能够缩减开发周期，快速迭代版本，并实现业务从老版本到新版本的平滑过渡，避免升级过程中出现的问题对用户造成的影响。研发部决定使用微服务架构，实现基于 Kubernetes 的容器化部署、CICD 和灰度发布。

节点规划如表 1 所示。

表 1 容器云平台节点规划

节点角色	主机名	VCPUS	内存	硬盘
Master、Harbor、CICD	master	8	12G	100G
Worker Node	node	4	8G	100G

任务 1 Docker CE 及私有仓库安装任务（5 分）

1. 在 master 节点中使用提供的脚本完成 Docker CE、docker-compose 以及 Harbor 仓库的安装，导入/opt/images 目录下的所有镜像，并推送到私有仓库。
2. 在 master、node 节点完成 Kubernetes 集群的安装。
3. 在 node 节点上从仓库中拉取 rabbitmq:latest 镜像，创建 docker-compse.yaml 文件，编排部署 RabbitMQ 服务，并设置 restart 策略。

任务 2 基于容器的 web 应用系统部署任务（10 分）

将云梦公司开发的基于微服务架构的 web 应用系统 ChinaSkillsMall 重新实现全容器化部署，原微服务架构的 Web 应用系统架构如图 1 所示，模块功能介绍图表 2 所示。

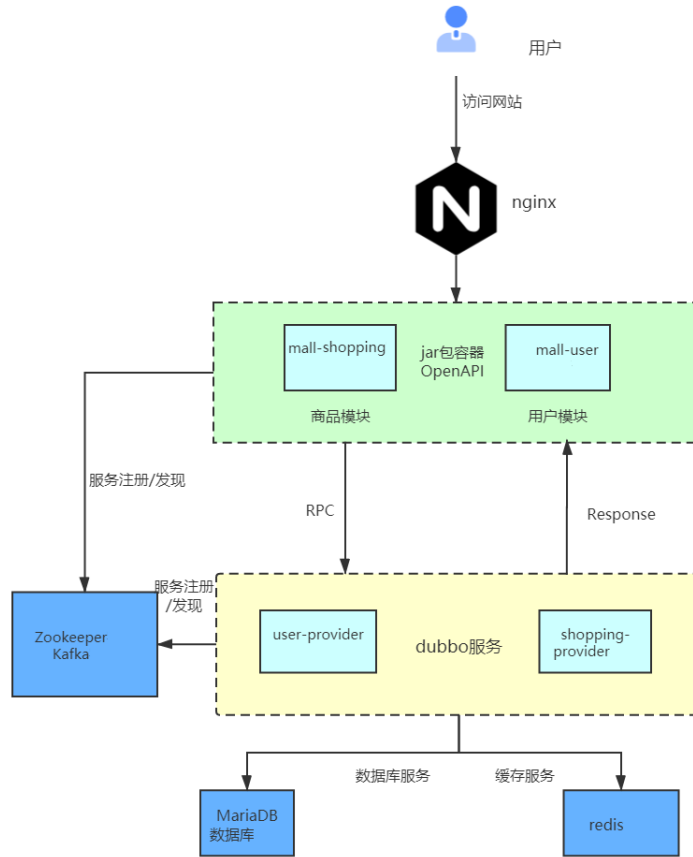


图 1 ChinaSkillsMall 系统架构图

表 2 ChinaSkillsMall 模块功能介绍

模块	使用技术	备注
mall.sql	MySQL	网站的数据库
dist 目录	Nginx	网站的前端项目
mall-shopping	web 项目	8081 端口，商品/购物车/首页渲染等交互
mall-user		8082 端口，提供用户相关的交互，如登录、注册、个人中心等
user-provider	后端服务	提供用户相关服务
shopping-provider		提供购物车、推荐商品、商品等服务

将 MySQL 数据库组件、Redis 组件、Kafka 组件、Zookeeper 组件按照要求进行容器化。

1. 在 master 节点上编写 /root/mall-swarm/Dockerfile-redis 文件构建 chinaskillmall-redis:v1.1 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装 redis 服务;
- (4) 修改配置文件中的 bind 127.0.0.1 为 bind 0.0.0.0;
- (5) 设置 redis 免密，并关闭保护模式;
- (6) 开放端口：6379;
- (7) 设置服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写 /root/mall-swarm/Dockerfile-mariadb 文件构建 chinaskillmall-mariadb:v1.0 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装并初始化 mariadb，密码：123456;
- (4) 设置数据库默认字符编码为 UTF-8;
- (5) 创建数据库 gpmall 并导入 gpmall.sql;
- (6) 设置 mariadb 开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

3. 在 master 节点上编写 /root/gpmall/Dockerfile-zookeeper 文件构建 chinaskillmall-zookeeper:v1.1 镜像，具体要求为

- (1) 基础镜像：centos:centos7.5.1804;
- (2) 作者：Chinaskill;
- (3) 安装 zookeeper 服务;
- (4) 开放端口 2181;
- (5) 设置 zookeeper 服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

4. 在 master 节点上编写/root/gpmall/Dockerfile-kafka 文件构建 chinaskillmall-kafka:v1.1 镜像，具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 kafka 服务;
- (4) 开放端口 9092;
- (5) 设置 zookeeper 服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

5. 在 master 节点上编写/root/gpmall/Dockerfile-nginx 文件构建 chinaskillmall-nginx:v1.1 镜像，具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 nginx 服务;
- (4) 将提供的软件包 dist.tar 和 jar 包拷贝进镜像;
- (5) 设置 nginx 服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

9. 在 master 节点上编写/root/gpmall/gpmall.yaml 文件，具体要求为

- (1) 在 default 命名空间下部署该服务;
- (2) service 名称: chinaskillmall;
- (3) 以 NodePort 方式对外暴露端口，端口映射: 30080:80;

完成后编排部署 ChinaSkillMall 商城。然后提交 master 节点的用户名、密码和 IP 到答题框。

任务 3 基于容器的持续集成部署任务（10 分）

该公司决定采用 GitLab +Jenkins 来构建 CICD 环境，以缩短新功能开发上线周期，及时

满足客户的需求，实现 DevOps 的部分流程，来减轻部署运维的负担，实现可视化容器生命周期管理、应用发布和版本迭代更新，请完成 GitLab + Jenkins + Kubernetes 的 CICD 环境部署。CICD 应用系统架构如图 2 所示。

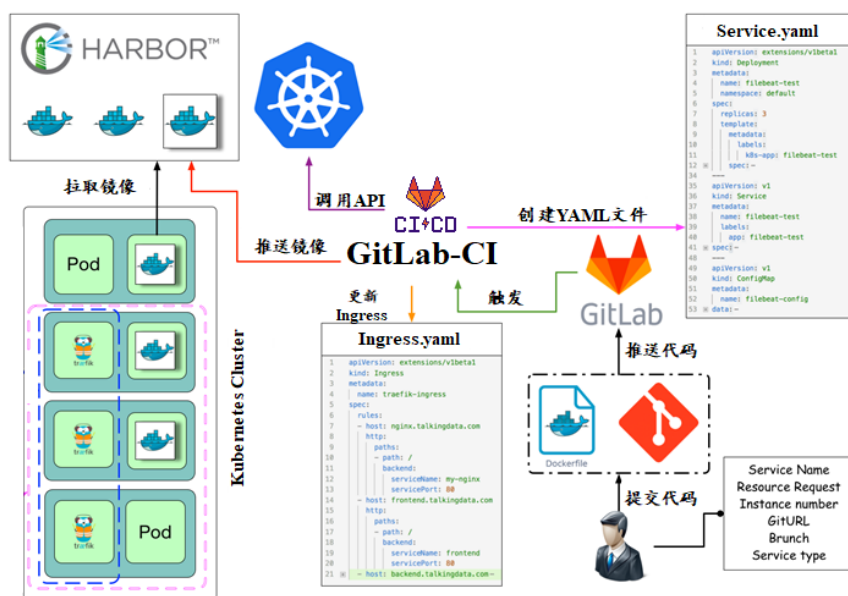


图 2 CICD 应用系统架构图

1. 在 master 节点上编写/root/jenkins/docker-compose.yaml 文件编排部署 Jenkins 服务，具体要求为

- (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: gpmall; 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 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写/root/gitlab/docker-compose.yaml 文件编排部署 Gitlab 服务，具体要求为

- (1) 容器名称: gitlab;

- (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 gpmall，将 gpmall 中的代码上传到 gpmall 项目中；
- (7) 实现通过 SSH 链接克隆项目。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

4. 配置 Jenkins 连接 Gitlab，具体要求为

- (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 master 节点的用户名、密码和 IP 到答题框。

4.配置 CI/CD，具体要求为

- (1) 新建一个流水线任务 gpmall；
- (2) 编写流水线脚本，构建 gpmall 项目中的商城服务，将构建后的镜像自动上传到 Harbor 仓库的 gpmall 项目，并自动发布服务到 Kubernetes 集群的 default 命名空间下；
- (3) 配置 Webhook；
- (4) 在 Harbor 中新建公开项目 gpmall。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

5.将 gpmall 项目中的 gpmall.yaml 文件补全，然后将 gpmall 项目代码重新上传到 Gitlab 中触发构建，完成后提交 master 节点的用户名、密码和 IP 到答题框。

任务 4 Kubernetes 容器云平台部署与运维（10 分，本任务只公布考试范围，不公布赛题）

当应用上线以后，运维面临的一大挑战是如何能够在不影响已上线业务的情况下进行

升级。在无法百分百避免版本升级故障的情况下，需要通过一种方式进行可控的版本发布，把故障影响控制在可以接受的范围内，并可以快速回退。公司决定采用基于 Istio 的灰度发布（又名金丝雀发布）来实现业务从老版本到新版本的平滑过渡，并避免升级过程中出现的问题对用户造成的影响。

请根据要求，完成 Kubernetes 的集群运维、调度、网络、存储、安全、应用生命周期管理、日志/监控等运维任务。

第三场次题目：公有云部署与运维（7）

云梦是一家科技公司，现在公司开发全面转向云原生，因此需要对公司内部项目搭建基于云原生的 Devops 相关服务。

某公有云是一家成熟的公有云提供商，为客户提供 K8S、云网络、云硬盘、云防火墙、负载均衡等服务，可创建 Web 服务，共享文件存储服务，数据库服务，数据库集群等服务。

根据上述公有云平台的特性，完成公有云中的各项运维工作。

任务 1 基础设施构建（10 分）

1.在公有云中完成虚拟私有云的创建，具体要求为

- （1）在贵阳区域进行创建操作；
- （2）创建一个名为 intnetX 的虚拟私有云：IP 地址为 172.16.1.0/16；
- （3）创建子网名称为 intnetX-server：IP 地址为 172.16.1.0/24；
- （4）创建一个名为 net-data 的虚拟私有云：网段为 192.168.1.0/16
- （5）创建子网名称为 net-mysql：IP 地址为 192.168.1.0/24；

完成创建后，提交当前的 AK（SecretId）、SK（SecretKey）和 intnetX 网络的 ID 到答题框。

2.创建对等连接 intnetX-net，配置 intnetX 虚拟私有云和 net-data 虚拟机私有云连通。操作完成后，提交当前的 AK（SecretId）、SK（SecretKey）和对等连接的 ID 到答题框。

3.登录华为云平台，创建两台云实例，具体要求为

- （1）计费模式：按需计费；
- （2）地域：贵阳；
- （3）CPU 架构：鲲鹏计算；
- （4）规格：kc1.xlarge.2；
- （5）镜像：CentOS 7.5 64 位；
- （6）系统盘：高 IO 50G 硬盘；
- （7）虚拟网络：intnetX-server；
- （8）实例名称：ChinaSkill-node-1、ChinaSkill-node-2；
- （9）登录方式：使用密码登录，密码自定义。

创建完成后，提交当前的 AK（SecretId）、SK（SecretKey）和 ChinaSkill-node-1 的实例 ID 到答题框。

4.创建一台云数据库，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) 数据库版本：MySQL 5.7；
- (4) 实例类型：主备；
- (5) 实例规格：鲲鹏 2 核 8GB/40GB；
- (6) 虚拟私有云：net-mysql；
- (7) 字符集：UTF8；
- (8) 端口：3306；
- (9) 数据库密码：Root123456；
- (10) 数据库名称：chinaskill-mysql；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和云数据库的实例 ID 到答题框。

5.给云数据库 chinaskill-mysql 创建只读数据库 chinaskill-read 实例。使用 2vCPU、4GB 内存规格。并给 chinaskill-mysql 申请读写分离地址。完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和只读数据库实例 ID 到答题框。

6.创建一个 Redis 缓存实例，具体要求为

- (1) 版本号：5.0；
- (2) 实例类型：主备；
- (3) 副本数：2；
- (4) 实例规格：4G；
- (5) 子网：net-mysql；
- (6) 密码：Root123456；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和实例 ID 到答题框。

7.在 chinaskill-node-2 云服务器上，使用提供的软件包安装 kafka 和 zookeeper 服务。安装完成后运行服务。完成后提交连接 chinaskill-node-2 节点的用户名、密码和公网 IP 地址到答题框。

8.在 chinaskill-node-1 云服务上部署 gpmall，mysql 使用云数据库分写分离地址，配置公有云 Redis 访问路径，使用提供的公有云软件包 gpmall-package 完成商城的上云操作。并设置 gpmall 所需 jar 软件包开启自动启动。完成后提交 chinaskill-node-1 的用户名、密码和公网 IP 到答题框。

9.创建一个云服务器备份存储库名为 `server_backup`，容量为 100G。将 `ChinaSkill-node-1` 云服务器制作镜像文件 `chinaskill-image`。完成后，将当前的 AK（SecretId）、SK（SecretKey）和存储库 ID 到答题框。

任务 2 系统管理与维护（10 分）

1.创建一个云服务器备份存储库名为 `server_backup`，容量为 100G。将 `ChinaSkill-node-1` 云服务器制作镜像文件 `chinaskill-image`。完成后，将当前的 AK（SecretId）、SK（SecretKey）和存储库 ID 到答题框。

2.创建一个负载均衡器 `chinaskill-elb`，将 `ChinaSkill-node-1` 和 `ChinaSkill-node-2` 加入负载均衡的后端。设置一个可用的公网服务 IP，服务端口为 80。配置监听器，监听 80 端口。对浮动公共 IP 进行 Web 访问测试。完成后，提交当前的 AK（SecretId）、SK（SecretKey）和负载均衡器的 ID 到答题框。

3.新建一个启动配置，具体要求为

- (1) 启动配置名称：`template-exam`；
- (2) 计费模式：按量计费；
- (3) 地域：贵阳；
- (4) CPU 架构：鲲鹏计算；
- (5) 规格：`kc1.xlarge.2`；
- (6) 镜像：`chinaskill-image`；
- (7) 系统盘：高 IO 50G 硬盘；
- (8) 登录方式：使用密码登录，密码自定义。

创建一个伸缩组，具体要求为

- (1) 名称：`as-exam`；
- (2) 最小伸缩数：1；起始实例数：1；最大伸缩数：5；
- (3) 启动配置：`template-exam`；
- (4) 使用负载均衡：`chinaskill-elb`；
- (5) 移出策略：移出最旧的实例。

为伸缩组 `as-exam` 新建告警触发策略，具体要求为

(1) 如果实例的内存利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例数减少 1 台。冷却 60 秒；

(2) 如果实例的内存利用率在 5 分钟内的最大值大于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒；

(3) 如果实例的 CPU 利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例数减少 1 台。冷却 60 秒；

(4) 如果实例的 CPU 利用率在 5 分钟内的平均值大于等于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒。

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和伸缩组的实例 ID 到答题框。

4. 创建一个鲲鹏架构集群，具体要求为

- (1) 集群名称: kcloud;
- (2) 集群版本: v1.15.11;
- (3) 地域: 贵阳;
- (4) 集群管理规模: 50 节点;
- (5) 控制节点数: 3;
- (6) 节点使用子网: intnetX-server;
- (7) Pod 实例上限: 64;
- (8) 容器网段: 10.10.0.0/16。

节点配置信息要求为

- (1) 节点名称: kcloud-server;
- (2) 节点规格: kc.xlarge.2
- (3) 节点: EulerOS 2.8

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和集群的 ID 到答题框。

5. 使用插件管理在 kcloud 容器集群中安装 dashboard 可视化监控界面。完成后，提交 dashboard 访问地址到答题框。

6. 购买云硬盘存储卷，具体配置为

- (1) 名称为: ckcloud-disk;
- (2) 集群: kcloud;
- (3) 容量: 30G;
- (4) 类型: 高 I/O;

完成后，创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和云硬盘的 ID 到答题框。

7.在 kcloud 集群中安装 kubectl 命令，使用 kubectl 命令管理 kcloud 集群。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

8.使用提供的 helm 软件包，在 kcloud 集群中安装 helm 服务。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

9.使用容器镜像服务作为镜像仓库，将提供的 httpd-2.4.33 和 httpd-2.4.35 镜像上传至服务中。提交当前的 AK（SecretId）、SK（SecretKey）和组织名称到答题框。

10.在 kcloud 集群中创建一个 Deployment，使用开源镜像中心的 nginx:latest，配置使其可以通过外部公网进行访问。完成后提交外部访问地址到答题框。

任务 3 自动化运维（10 分，本任务只公布考试范围，不公布赛题）

编写 Python 脚本调用公有云 API 完成公有云运维工作。

第一场次题目：OpenStack 平台部署与运维（8）

某企业拟使用 OpenStack 搭建一个企业云平台，以实现资源池化弹性管理、企业应用集中管理、统一安全认证和授权等管理。

系统架构如图 1 所示，IP 地址规划如表 1 所示。

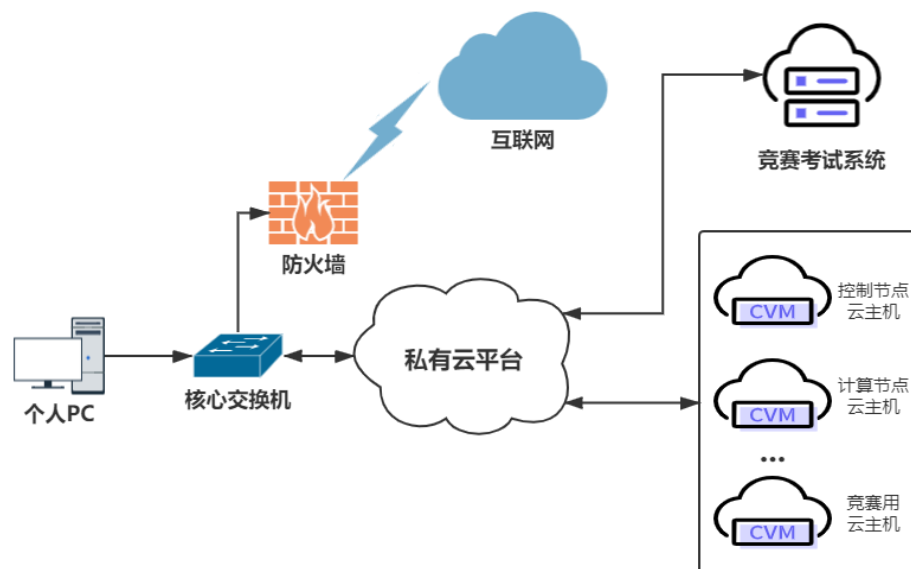


图 1 系统架构图

表 1 IP 地址规划

设备名称	主机名	接口	IP 地址	说明
云服务器1	Controller	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器2	Compute	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器3	自定义	eth0	172.129.x.0/24	用于实操题
云服务器 n				
PC-1		本地连接	172.24.16.0/24	PC 使用

说明：

- 1.竞赛使用集群模式进行，比赛时给每个参赛队提供独立的租户与用户，各用户的资源配额相同，选手通过用户名与密码登录竞赛用私有云平台，创建云主机进行相应答题；
- 2.表中的 x 为工位号；在进行 OpenStack 搭建时的第二块网卡地址根据题意自行创建；
- 3.根据图表给出的信息，检查硬件连线及网络设备配置，确保网络连接正常；
- 4.考试所需要的账号资源、竞赛资源包与附件均会在考位信息表与设备确认单中给出；
- 5.竞赛过程中，为确保服务器的安全，请自行修改服务器密码；在考试系统提交信息时，请确认自己的 IP 地址，用户名和密码。

任务 1 基础运维任务（5 分）

- 1.使用提供的用户名密码，登录提供的 OpenStack 私有云平台，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；

完成后提交控制节点的用户名、密码和 IP 地址到答题框。

2.使用提供的用户名密码，登录提供的 OpenStack 私有云平台，自行使用 CentOS7.5 镜像创建一台云主机，flavor 使用 2v_4G_50G 的配置，使用单网卡。启动后使用提供的 ansible.tar.gz 软件包在这个节点上安装 ansible 服务并配置 ansible 节点与 controller、compute 节点的 hosts 主机名映射。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

3.配置 ansible 节点无秘钥连接 controller 节点和 compute 节点，配置完成后并完成 ssh 连接两个节点的 hostname 进行测试。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

4.在 ansible 节点配置 ansible 的 hosts 文件，要求创建两个组分别为 controller 和 compute，controller 组下主机节点为 controller 节点；compute 组下主机节点为 compute。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

5.在 compute 节点上利用空白分区划分 2 个 20G 分区。完成后提交计算节点的用户名、密码和 IP 地址到答题框。

任务 2 OpenStack 搭建任务（10 分）

1.使用提供的 openstack_ansible.tar.gz 项目包解压至 ansible 节点的/opt 目录下，然后编辑 roles 目录下 init/tasks 中的 main.yaml；编辑 group_vars 目录下的 all 文件（openstack 中的密码都设置为 000000）；编辑 install_openstack.yaml 文件，要求执行 install_openstack.yaml 文件可以在 controller 节点和 compute 节点执行 init 这个 role 来安装 iaas-pre-host。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。（考试系统会进入你的 ansible 节点来执行 install_openstack.yaml，请确保你的环境处于正确的可执行状态）

2.在 ansible 节点，编辑 roles/mariadb/tasks 下的 main.yaml 文件和 install_openstack.yaml，在 controller 节点上安装数据库服务。自行执行 install_openstack.yaml 文件安装数据库服务。

3.在 ansible 节点，编辑 roles/keystone/tasks 下的 main.yaml 文件和 install_openstack.yaml，在 controller 节点上安装 keystone 服务。自行执行 install_openstack.yaml 文件安装 keystone 服务。

4.在 ansible 节点，编辑 roles/glance/tasks 下的 main.yaml 文件和 install_openstack.yaml，

在 controller 节点上安装 glance 服务。自行执行 install_openstack.yaml 文件安装 glance 服务。

5.在 ansible 节点, 编辑 roles/nova-controller/tasks 下的 main.yaml 文件、roles/nova-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 nova 服务。

6.在 ansible 节点, 编辑 roles/neutron-controller/tasks 下的 main.yaml 文件、roles/neutron-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 neutron 服务。

7.在 ansible 节点, 编辑 roles/dashboard/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点上安装 dashboard 服务。

8.在 ansible 节点, 编辑 roles/swift-controller/tasks 下的 main.yaml 文件、roles/swift-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 swift 服务。

9.在 ansible 节点, 编辑 roles/cinder-controller/tasks 下的 main.yaml 文件、roles/cinder-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 cinder 服务。

10.在 ansible 节点, 编辑 roles/heat/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点上安装 heat 服务。

任务 3 OpenStack 云平台运维 (10 分)

1.使用提供的云安全框架组件, 将自行搭建的 OpenStack 云平台的安全策略从 http 优化至 https。

2.在自行搭建的 OpenStack 平台上, 使用 glance 相关命令上传镜像, 镜像源为 CentOS_7.5_x86_64.qcow2, 名为 Gmirror1, min_ram 为 2048M, min_disk 为 20G。

3.使用 qemu-img 相关命令, 查询 Gmirror1 镜像的 compat 版本, 然后将 Gmirror1 镜像的 compat 版本修改为 0.10 (该操作是为了适配某些低版本的云平台)。

4.在自行搭建的 OpenStack 平台上, 通过修改相关参数对 openstack 平台进行调优操作, 相应的调优操作有:

- (1) 设置内存超售比例为 1.5 倍;
- (2) 设置 nova 服务心跳检查时间为 120 秒。

5.在自行搭建的 OpenStack 平台上, 使用 Swift 对象存储服务, 修改相应的配置文件, 使对象存储 Swift 作为 glance 镜像服务的后端存储。

6.在自行搭建的 OpenStack 平台上, 编写 heat 模板 createvm.yml 文件, 模板作用为按照要求创建一个云主机。

7.在自行搭建的 OpenStack 平台上, 对 cinder 存储空间进行扩容操作, 要求将 cinder 存储空间扩容 10G。

8.登录提供的私有云平台, 创建一台 centos7.5 的云主机, 使用提供的软件包, 在这台

云主机上安装数据库、redis、zookeeper 和 kafka 等服务，然后将商城应用部署在该云主机上，实现网站的访问。

9.登录提供的私有云平台，创建三台 centos7.5 的云主机，使用提供的软件包，在这三台云主机安装上安装 Redis 服务，并配置成 Redis 哨兵模式。

10.登录提供的私有云平台，使用镜像 OpenStack-error-3 创建云主机“all-in-one”(账号：root；密码：000000)。创建后的云主机内有错误的 OpenStack 平台，错误现象为 cinder 服务无法正常使用，请结合报错信息排查错误，使 cinder 服务可以正常使用。

任务 4 OpenStack 云平台运维开发（10 分，本任务只公布考试范围，不公布赛题）

此任务包含 2-4 个子任务，内容涉及编写 Python 脚本调用 OpenStack API 对 OpenStack 云平台进行管理和运维。使用自动化运维工具 Ansible 对云主机进行批量系统配置、批量程序部署、批量运行命令等运维操作。

第二场次题目：容器云平台部署与运维（8）

说明：完成本任务需要两台安装了 CentOS7.5 操作系统的云主机 master 和 node，镜像包中有本次容器云平台部署和运维所需的所有文件。

云梦公司技术部产品开发上线周期长，用户规模大且版本更新频繁。新版本的每次上线，产品都要承受极大的压力。引入 CICD (Continuous Integration 持续集成、Continuous Delivery 持续交付) 和灰度发布成了公司的当务之急，研发团队决定搭建基于 Kubernetes 的 CICD 环境，希望基于这个平台来实现 DevOps 的部分流程，并基于 Kubernetes 实现业务系统的灰度发布。

为了能够让云梦公司开发的 web 应用系统产品能够缩减开发周期，快速迭代版本，并实现业务从老版本到新版本的平滑过渡，避免升级过程中出现的问题对用户造成的影响。研发部决定使用微服务架构，实现基于 Kubernetes 的容器化部署、CICD 和灰度发布。

节点规划如表 1 所示。

表 1 容器云平台节点规划

节点角色	主机名	VCPUS	内存	硬盘
Master、Harbor、CICD	master	8	12G	100G
Worker Node	node	4	8G	100G

任务 1 Docker CE 及私有仓库安装任务（5 分）

1. 在 master 节点中使用提供的脚本完成 Docker CE、docker-compose 以及 Harbor 仓库的安装，导入/opt/images 目录下的所有镜像，并推送到私有仓库。
2. 在 master、node 节点完成 Kubernetes 集群的安装。
3. 在 node 节点上从仓库中拉取 skywalking:latest 和 kibana:latest 镜像，创建 docker-compse.yaml 文件，编排部署 Skywalking 服务，并设置 restart 策略。

任务 2 基于容器的 web 应用系统部署任务（10 分）

云梦公司开发了一套基于 SpringBoot+MyBatis 微服务架构的 ChinaSkillsMall 电商系统，并实现全容器化部署，ChinaSkillsMall 电商应用系统架构如图 1 所示，系统说明如表 2 所示。

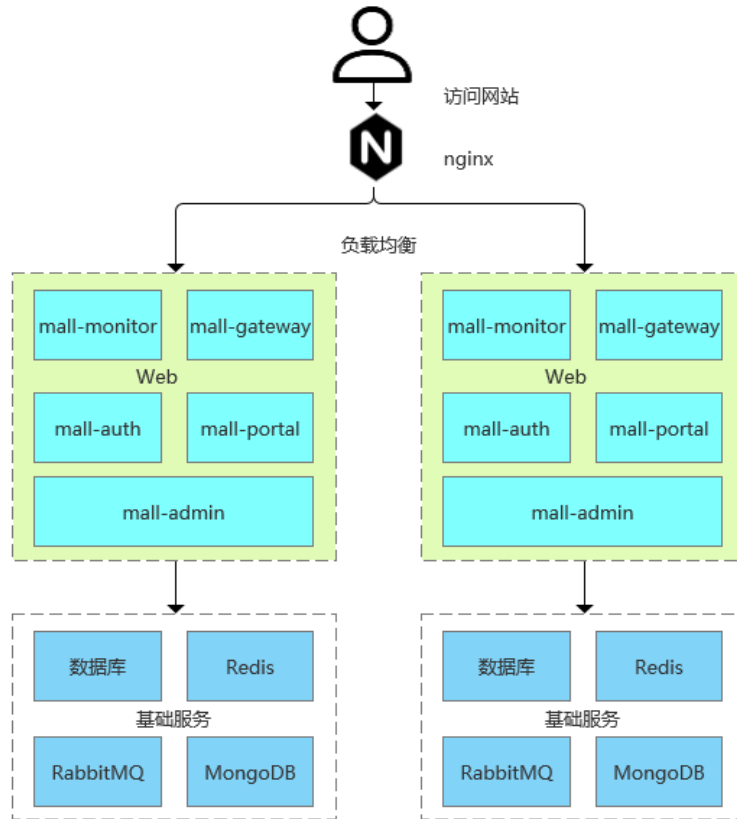


图 1 ChinaSkillsMall 电商应用系统架构图

表 2 ChinaSkillsMall 电商应用系统说明

模块	说明
mall-monitor	监控中心
mall-gateway	微服务网关
mall-auth	认证中心
mall-admin	商城后台服务
mall-portal	商城前台服务

将 MariaDB 数据库组件、Redis 消息组件、RabbitMQ 消息组件、Nacos-Registry 注册中心服务组件和前端 Nginx 组件按照要求进行容器化。

1. 在 master 节点上编写 /root/mall-swarm/Dockerfile-mariadb 文件构建 chinaskillmall-mariadb:v1.1 镜像，具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装并初始化 mariadb, 密码: root;
- (4) 设置数据库编码为 UTF-8;
- (5) 开放端口: 3306
- (6) 设置 mariadb 开机自启。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写 /root/mall-swarm/Dockerfile-redis 文件构建 chinaskillmall-redis:v1.1 镜像, 具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 redis 服务;
- (4) 修改配置文件中的 bind 127.0.0.1 为 bind 0.0.0.0;
- (5) 设置 redis 免密, 并关闭保护模式;
- (6) 开放端口: 6379;
- (7) 设置服务开机自启。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

3. 在 master 节点上编写 /root/mall-swarm/Dockerfile-nacos 文件构建 chinaskillmall-nacos:v1.1 镜像, 具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 jdk 工具和 nacos-server 工具;
- (4) 设置 nacos 服务开机自启。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

4. 在 master 节点上安装 node 工具, 对 mall-admin-web 项目进行打包, 然后编写 /root/mall-swarm/Dockerfile-nginx 文件构建 chinaskillmall-nginx:v1.1 镜像, 具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 nginx 服务;
- (4) 将 mall-admin-web 项目打包后的文件拷贝到/usr/share/nginx/html 目录下;

(5) 设置 nginx 服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

5. 在 master 节点上编写/root/mall-swarm/docker-compose.yaml 文件，具体要求为

(1) 容器 1 名称: chinaskillmall-admin; 镜像: mall/mall-admin:1.0-SNAPSHOT; 端口映射: 8080:8080;

(2) 容器 2 名称: chinaskillmall-auth; 镜像: mall/mall-auth:1.0-SNAPSHOT; 端口映射: 8401:8401;

(3) 容器 3 名称: chinaskillmall-gateway; 镜像: mall/mall-gateway:1.0-SNAPSHOT; 端口映射: 8201:8201;

(4) 容器 4 名称: chinaskillmall-mysql; 镜像: chinaskillmall-mariadb:v1.1; 端口映射: 3306:3306;

(5) 容器 5 名称: chinaskillmall-nacos; 镜像: chinaskillmall-nacos:v1.1; 端口映射: 8848:8848;

(6) 容器 6 名称: chinaskillmall-nginx; 镜像: chinaskillmall-nginx:v1.1; 端口映射: 8888:80;

(7) 容器 7 名称: chinaskillmall-redis; 镜像: chinaskillmall-redis:v1.1; 端口映射: 6379:6379。

完成后编排部署 ChinaSkillsMall 商城。然后提交 master 节点的用户名、密码和 IP 到答题框。

任务 3 基于容器的持续集成部署任务 (10 分)

该公司决定采用 GitLab + Jenkins 来构建 CICD 环境，以缩短新功能开发上线周期，及时满足客户的需求，实现 DevOps 的部分流程，来减轻部署运维的负担，实现可视化容器生命周期管理、应用发布和版本迭代更新，请完成 GitLab + Jenkins + Kubernetes 的 CICD 环境部署。CICD 应用系统架构如图 2 所示。

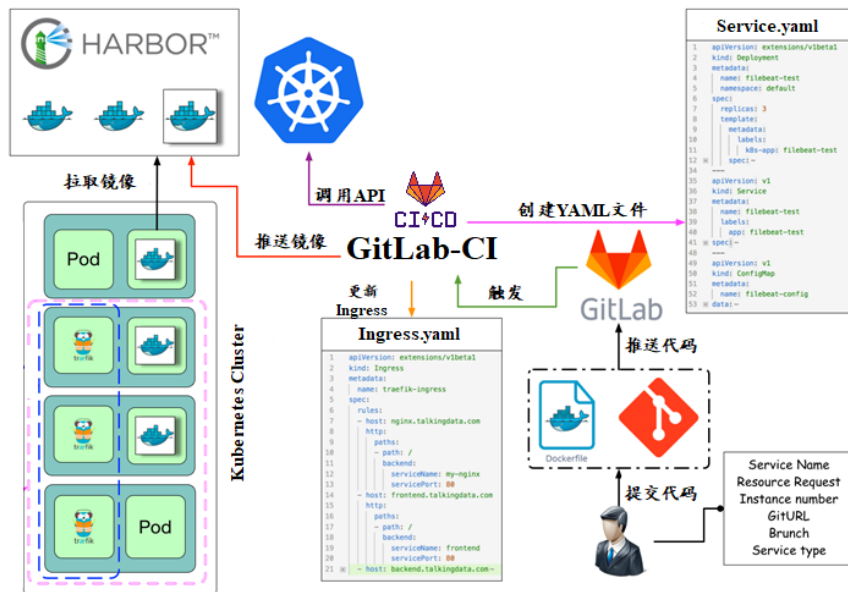


图 2 CI/CD 应用系统架构图

1. 在 master 节点上编写/root/jenkins/docker-compose.yaml 文件编排部署 Jenkins 服务，具体要求为

- (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: gpmall; 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 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写/root/gitlab/docker-compose.yaml 文件编排部署 Gitlab 服务，具体要求为

- (1) 容器名称: gitlab;
- (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 gpmall, 将 gpmall 中的代码上传到 gpmall 项目中;

(7) 实现通过 SSH 链接克隆项目。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

5. 配置 Jenkins 连接 Gitlab, 具体要求为

(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 master 节点的用户名、密码和 IP 到答题框。

4.配置 CI/CD, 具体要求为

(1) 新建一个流水线任务 gpmall;

(2) 编写流水线脚本, 构建 gpmall 项目中的商城服务, 将构建后的镜像自动上传到 Harbor 仓库的 gpmall 项目, 并自动发布服务到 Kubernetes 集群的 default 命名空间下;

(3) 配置 Webhook;

(4) 在 Harbor 中新建公开项目 gpmall。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

5.将 gpmall 项目中的 gpmall.yaml 文件补全, 然后将 gpmall 项目代码重新上传到 Gitlab 中触发构建, 完成后提交 master 节点的用户名、密码和 IP 到答题框。

任务 4 Kubernetes 容器云平台部署与运维 (10 分, 本任务只公布考试范围, 不公布赛题)

当应用上线以后, 运维面临的一大挑战是如何能够在不影响已上线业务的情况下进行升级。在无法百分百避免版本升级故障的情况下, 需要通过一种方式进行可控的版本发布, 把故障影响控制在可以接受的范围内, 并可以快速回退。公司决定采用基于 Istio 的灰度发布 (又名金丝雀发布) 来实现业务从老版本到新版本的平滑过渡, 并避免升级过程中出现的问题对用户造成的影响。

请根据要求, 完成 Kubernetes 的集群运维、调度、网络、存储、安全、应用生命周期

管理、日志/监控等运维任务。

第三场次题目：公有云部署与运维（8）

云梦是一家科技公司，现在公司开发全面转向云原生，因此需要对公司内部项目搭建基于云原生的 Devops 相关服务。

某公有云是一家成熟的公有云提供商，为客户提供 K8S、云网络、云硬盘、云防火墙、负载均衡等服务，可创建 Web 服务，共享文件存储服务，数据库服务，数据库集群等服务。

根据上述公有云平台的特性，完成公有云中的各项运维工作。

任务 1 基础设施构建（10 分）

1.在公有云中完成虚拟私有云的创建，具体要求为

- (1) 在贵阳区域进行创建操作；
- (2) 创建一个名为 intnetX 的虚拟私有云：IP 地址为 172.16.1.0/16；
- (3) 创建子网名称为 intnetX-server：IP 地址为 172.16.1.0/24；
- (4) 创建一个名为 net-data 的虚拟私有云：网段为 192.168.1.0/16
- (5) 创建子网名称为 net-mysql：IP 地址为 192.168.1.0/24；

完成创建后，提交当前的 AK（SecretId）、SK（SecretKey）和 intnetX 网络的 ID 到答题框。

2.创建对等连接 intnetX-net，配置 intnetX 虚拟私有云和 net-data 虚拟机私有云连通。操作完成后，提交当前的 AK（SecretId）、SK（SecretKey）和对等连接的 ID 到答题框。

3.登录华为云平台，创建两台云实例，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) CPU 架构：鲲鹏计算；
- (4) 规格：kc1.xlarge.2；
- (5) 镜像：CentOS 7.5 64 位；
- (6) 系统盘：高 IO 50G 硬盘；
- (7) 虚拟网络：intnetX-server；
- (8) 实例名称：ChinaSkill-node-1、ChinaSkill-node-2；
- (9) 登录方式：使用密码登录，密码自定义。

创建完成后，提交当前的 AK（SecretId）、SK（SecretKey）和 ChinaSkill-node-1 的实例 ID 到答题框。

4.购买云硬盘实例，使用超高 IO 磁盘类型，申请购买 100G 磁盘大小。创建完成后，提

交当前的 AK (SecretId)、SK (SecretKey) 和云硬盘 ID 到答题框。

5.创建一台云数据库，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) 数据库版本：MySQL 5.7；
- (4) 实例类型：主备；
- (5) 实例规格：鲲鹏 2 核 8GB/40GB；
- (6) 虚拟私有云：net-mysql；
- (7) 字符集：UTF8；
- (8) 端口：3306；
- (9) 数据库密码：Root123456；
- (10) 数据库名称：chinaskill-mysql；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和云数据库的实例 ID 到答题框。

6.给云数据库 chinaskill-mysql 创建只读数据库 chinaskill-read 实例。使用 2vCPU、4GB 内存规格。并给 chinaskill-mysql 申请读写分离地址。完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和只读数据库实例 ID 到答题框。

7.创建一个网络 ACL，具体要求为

- (1) 名称：intnetX-acl；
- (2) 允许策略：只放行源地址为 172.16.1.0/24 访问 3306 端口；
- (3) 允许策略：只放行源地址为 172.16.1.0/24 使用 ping 命令访问；
- (4) 关联子网：关联网络 ACL 至 net-mysql 子网中；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和网络 ACL ID 到答题框。

8.使用公有云平台，申请对象存储服务，存储桶名字自定义。自行编写一个静态网页，网页的内容为“welcome to China”，使用申请的对象存储服务，托管该静态网页。托管成功后，提交你对象存储的访问域名到答题框。

9.使用提供的软件包，在 ChinaSkill-node-1 和 ChinaSkill-node-2 中部署 WordPress 的 Web 服务即 LNMP+WordPress。使用云数据库 chinaskill-mysql 读写分离地址。完成 WordPress 系统上云后，将 ChinaSkill-node-1 节点的 WordPress 访问地址提交到答题框。

任务 2 系统管理与维护（10 分）

1.使用所创建的私有镜像 chinaskill-image 购买云服务器 ChinaSkill-node-3，配置与 ChinaSkill-node-1 云服务器相同。完成后提交 chinaskill-node-3 的用户名、密码和公网 IP 到答题框。

2.创建一个负载均衡器 chinaskill-elb，将 ChinaSkill-node-1 和 ChinaSkill-node-3 加入负载均衡的后端。设置一个可用的公网服务 IP，服务端口为 80。配置监听器，监听 80 端口。对浮动公共 IP 进行 Web 访问测试。完成后，提交当前的 AK（SecretId）、SK（SecretKey）和负载均衡器的 ID 到答题框。

3.新建一个启动配置，具体要求为

- (1) 启动配置名称：template-exam；
- (2) 计费模式：按量计费；
- (3) 地域：贵阳；
- (4) CPU 架构：鲲鹏计算；
- (5) 规格：kc1.xlarge.2；
- (6) 镜像：chinaskill-image；
- (7) 系统盘：高 IO 50G 硬盘；
- (8) 登录方式：使用密码登录，密码自定义。

创建一个伸缩组，具体要求为

- (1) 名称：as-exam；
- (2) 最小伸缩数：1；起始实例数：1；最大伸缩数：5；
- (3) 启动配置：template-exam；
- (4) 使用负载均衡：chinaskill-elb；
- (5) 移出策略：移出最旧的实例。

为伸缩组 as-exam 新建告警触发策略，具体要求为

(1) 如果实例的内存利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例数减少 1 台。冷却 60 秒；

(2) 如果实例的内存利用率在 5 分钟内的最大值大于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒；

(3) 如果实例的 CPU 利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例

数减少 1 台。冷却 60 秒；

(4) 如果实例的 CPU 利用率在 5 分钟内的平均值大于等于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒。

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和伸缩组的实例 ID 到答题框。

4. 创建一个鲲鹏架构集群，具体要求为

- (1) 集群名称: kcloud;
- (2) 集群版本: v1.15.11;
- (3) 地域: 贵阳;
- (4) 集群管理规模: 50 节点;
- (5) 控制节点数: 3;
- (6) 节点使用子网: intnetX-server;
- (7) Pod 实例上限: 64;
- (8) 容器网段: 10.10.0.0/16。

节点配置信息要求为

- (1) 节点名称: kcloud-server;
- (2) 节点规格: kc.xlarge.2
- (3) 节点: EulerOS 2.8

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和集群的 ID 到答题框。

5. 使用插件管理在 kcloud 容器集群中安装 dashboard 可视化监控界面。完成后，提交 dashboard 访问地址到答题框。

6. 购买云硬盘存储卷，具体配置为

- (1) 名称为: ckcloud-disk;
- (2) 集群: kcloud;
- (3) 容量: 30G;
- (4) 类型: 高 I/O;

完成后，创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和云硬盘的 ID 到答题框。

7. 在 kcloud 集群中安装 kubectl 命令，使用 kubectl 命令管理 kcloud 集群。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

8. 使用提供的 helm 软件包，在 kcloud 集群中安装 helm 服务。完成后提交连接 kcloud

集群节点的用户名、密码和公网 IP 地址到答题框。

9.使用容器镜像服务作为镜像仓库，将提供的 `httpd-2.4.33` 和 `httpd-2.4.35` 镜像上传至服务中。提交当前的 AK（SecretId）、SK（SecretKey）和组织名称到答题框。

10.在 `kcloud` 集群中创建一个 `Deployment`，使用所上传的 `httpd-2.4.33` 容器镜像，配置使其可以通过外部公网进行访问。完成后提交外部访问地址到答题框。

任务 3 自动化运维（10 分，本任务只公布考试范围，不公布赛题）

编写 Python 脚本调用公有云 API 完成公有云运维工作。

第一场次题目：OpenStack 平台部署与运维（9）

某企业拟使用 OpenStack 搭建一个企业云平台，以实现资源池化弹性管理、企业应用集中管理、统一安全认证和授权等管理。

系统架构如图 1 所示，IP 地址规划如表 1 所示。

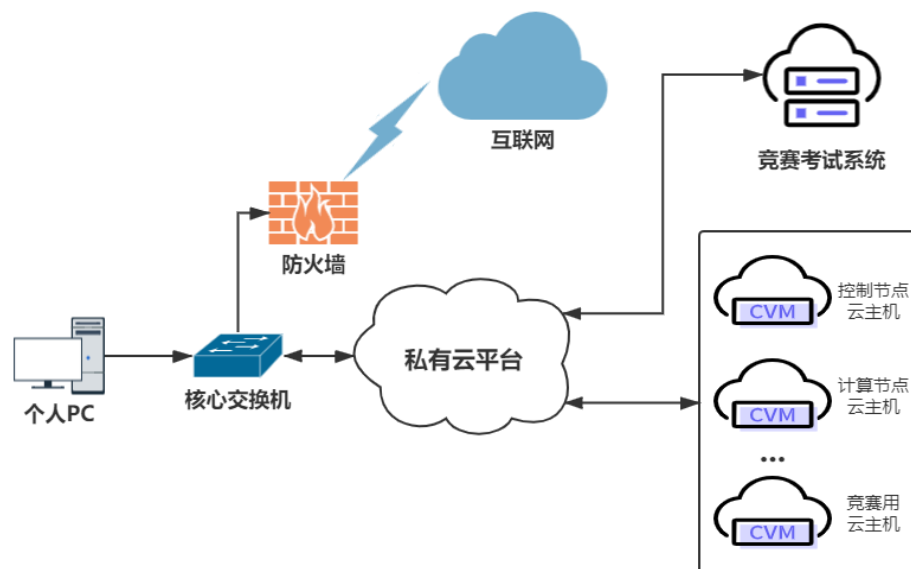


图 1 系统架构图

表 1 IP 地址规划

设备名称	主机名	接口	IP 地址	说明
云服务器1	Controller	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器2	Compute	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器3	自定义	eth0	172.129.x.0/24	用于实操题
云服务器 n				
PC-1		本地连接	172.24.16.0/24	PC 使用

说明：

- 1.竞赛使用集群模式进行，比赛时给每个参赛队提供独立的租户与用户，各用户的资源配额相同，选手通过用户名与密码登录竞赛用私有云平台，创建云主机进行相应答题；
- 2.表中的 x 为工位号；在进行 OpenStack 搭建时的第二块网卡地址根据题意自行创建；
- 3.根据图表给出的信息，检查硬件连线及网络设备配置，确保网络连接正常；
- 4.考试所需要的账号资源、竞赛资源包与附件均会在考位信息表与设备确认单中给出；
- 5.竞赛过程中，为确保服务器的安全，请自行修改服务器密码；在考试系统提交信息时，请确认自己的 IP 地址，用户名和密码。

任务 1 基础运维任务（5 分）

- 1.使用提供的用户名密码，登录提供的 OpenStack 私有云平台，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；

完成后提交控制节点的用户名、密码和 IP 地址到答题框。

2.使用提供的用户名密码，登录提供的 OpenStack 私有云平台，自行使用 CentOS7.5 镜像创建一台云主机，flavor 使用 2v_4G_50G 的配置，使用单网卡。启动后使用提供的 ansible.tar.gz 软件包在这个节点上安装 ansible 服务并配置 ansible 节点与 controller、compute 节点的 hosts 主机名映射。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

3.配置 ansible 节点无秘钥连接 controller 节点和 compute 节点，配置完成后并完成 ssh 连接两个节点的 hostname 进行测试。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

4.在 ansible 节点配置 ansible 的 hosts 文件，要求创建两个组分别为 controller 和 compute，controller 组下主机节点为 controller 节点；compute 组下主机节点为 compute。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

5.在 compute 节点上利用空白分区划分 2 个 20G 分区。完成后提交计算节点的用户名、密码和 IP 地址到答题框。

任务 2 OpenStack 搭建任务（10 分）

1.使用提供的 openstack_ansible.tar.gz 项目包解压至 ansible 节点的/opt 目录下，然后编辑 roles 目录下 init/tasks 中的 main.yaml；编辑 group_vars 目录下的 all 文件（openstack 中的密码都设置为 000000）；编辑 install_openstack.yaml 文件，要求执行 install_openstack.yaml 文件可以在 controller 节点和 compute 节点执行 init 这个 role 来安装 iaas-pre-host。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。（考试系统会进入你的 ansible 节点来执行 install_openstack.yaml，请确保你的环境处于正确的可执行状态）

2.在 ansible 节点，编辑 roles/mariadb/tasks 下的 main.yaml 文件和 install_openstack.yaml，在 controller 节点上安装数据库服务。自行执行 install_openstack.yaml 文件安装数据库服务。

3.在 ansible 节点，编辑 roles/keystone/tasks 下的 main.yaml 文件和 install_openstack.yaml，在 controller 节点上安装 keystone 服务。自行执行 install_openstack.yaml 文件安装 keystone 服务。

4.在 ansible 节点，编辑 roles/glance/tasks 下的 main.yaml 文件和 install_openstack.yaml，

在 controller 节点上安装 glance 服务。自行执行 install_openstack.yaml 文件安装 glance 服务。

5.在 ansible 节点, 编辑 roles/nova-controller/tasks 下的 main.yaml 文件、roles/nova-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 nova 服务。

6.在 ansible 节点, 编辑 roles/neutron-controller/tasks 下的 main.yaml 文件、roles/neutron-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 neutron 服务。

7.在 ansible 节点, 编辑 roles/dashboard/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点上安装 dashboard 服务。

8.在 ansible 节点, 编辑 roles/swift-controller/tasks 下的 main.yaml 文件、roles/swift-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 swift 服务。

9.在 ansible 节点, 编辑 roles/cinder-controller/tasks 下的 main.yaml 文件、roles/cinder-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 cinder 服务。

10.在 ansible 节点, 编辑 roles/heat/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点上安装 heat 服务。

任务 3 OpenStack 云平台运维 (10 分)

1.使用提供的云安全框架组件, 将自行搭建的 OpenStack 云平台的安全策略从 http 优化至 https。

2.在自行搭建的 OpenStack 平台上, 修改 glance 镜像存放的后端存储。

3.在 controller 节点上, 编写脚本 mysqlbak.sh, 要求执行该脚本可以备份数据库, 并存放在/opt/mysqlbak 目录下。

4.在 controller 节点上, 创建名为 test 的容器, 并上传 skill.txt 文件 (文件可自行创建) 至 test 容器中。

5.在自行搭建的 OpenStack 平台上, 使用 cirros 镜像创建云主机 vm1, 并使用 cinder 服务在这个云主机上挂载一个大小为 2G 的硬盘。

6.在自行搭建的 OpenStack 平台上, 修改相关参数对 openstack 平台进行调优操作, 相应的调优操作有:

- (1) 设置 cpu 超售比例为 4 倍;
- (2) 设置内存超售比例为 1.5 倍;
- (3) 预留 2048mb 内存, 这部分内存不能被虚拟机使用;
- (4) 预留 10240mb 磁盘, 这部分磁盘不能被虚拟机使用;

7.登录提供的私有云平台, 创建一台 centos7.5 的云主机, 使用提供的软件包, 在这台

云主机上安装 zabbix 监控服务，然后配置该服务监控 controller 节点。

8.登录提供的私有云平台，创建两台 centos7.5 的云主机，使用提供的软件包，将这两台云主机上安装 MariaDB 数据库服务，并配置为主从数据库。

9.登录提供的私有云平台，再创建一台 centos7.5 的云主机，使用提供的软件包与上一题配置完成的主从数据库，将这三台云主机配置为数据库读写分离架构。

10.登录提供的私有云平台，使用镜像 OpenStack-error-4 创建云主机“all-in-one”(账号：root；密码：000000)。创建后的云主机内有错误的 OpenStack 平台，错误现象为 swift 服务无法正常使用，请结合报错信息排查错误，使 swift 服务可以正常使用。

任务 4 OpenStack 云平台运维开发（10 分，本任务只公布考试范围，不公布赛题）

此任务包含 2-4 个子任务，内容涉及编写 Python 脚本调用 OpenStack API 对 OpenStack 云平台进行管理和运维。使用自动化运维工具 Ansible 对云主机进行批量系统配置、批量程序部署、批量运行命令等运维操作。

第二场次题目：容器云平台部署与运维（9）

说明：完成本任务需要两台安装了 CentOS7.5 操作系统的云主机 master 和 node，镜像包中有本次容器云平台部署和运维所需的所有文件。

云梦公司技术部产品开发上线周期长，用户规模大且版本更新频繁。新版本的每次上线，产品都要承受极大的压力。引入 CICD (Continuous Integration 持续集成、Continuous Delivery 持续交付) 和灰度发布成了公司的当务之急，研发团队决定搭建基于 Kubernetes 的 CICD 环境，希望基于这个平台来实现 DevOps 的部分流程，并基于 Kubernetes 实现业务系统的灰度发布。

为了能够让云梦公司开发的 web 应用系统产品能够缩减开发周期，快速迭代版本，并实现业务从老版本到新版本的平滑过渡，避免升级过程中出现的问题对用户造成的影响。研发部决定使用微服务架构，实现基于 Kubernetes 的容器化部署、CICD 和灰度发布。

节点规划如表 1 所示。

表 1 容器云平台节点规划

节点角色	主机名	VCPUS	内存	硬盘
Master、Harbor、CICD	master	8	12G	100G
Worker Node	node	4	8G	100G

任务 1 Docker CE 及私有仓库安装任务（5 分）

1. 在 master 节点中使用提供的脚本完成 Docker CE、docker-compose 以及 Harbor 仓库的安装，导入/opt/images 目录下的所有镜像，并推送到私有仓库。
2. 在 master、node 节点完成 Kubernetes 集群的安装。
3. 在 node 节点上从仓库中拉取 elasticsearch:latest 和 kibana:latest 镜像，创建 docker-compse.yaml 文件，编排部署 ES 集群，并设置 restart 策略。

任务2 基于容器的 web 应用系统部署任务（10 分）

将云梦公司开发的基于微服务架构的 ChinaSkillsSA 智能文本的情绪分析（Sentiment Analysis）的系统实现全容器化部署，ChinaSkillsSA 智能分析系统应用系统架构如图 1 所示，ChinaSkillsSA 智能分析系统说明如表 2 所示。

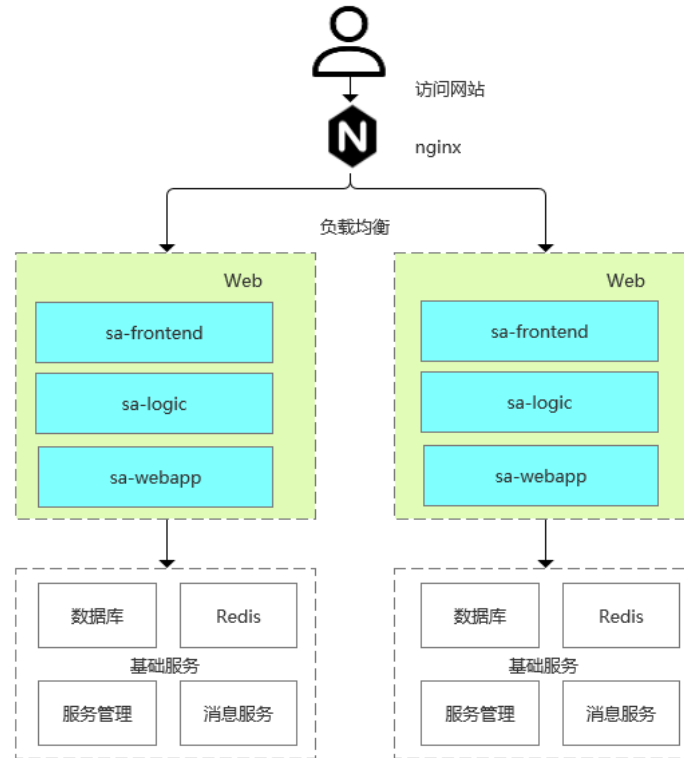


图 1 ChinaSkillsSA 智能分析系统架构图

表 2 ChinaSkillsSA 智能分析系统说明

模块	使用技术	说明
sa-frontend	Node.js	前端项目，使用 Vue.js 开发。
sa-logic	Python Flask	输入文本句子的情绪分析微服务，使用 Python 开发。
sa-webapp	Spring Cloud	Web 服务，处理 sa-frontend 请求，调用 sa-logic 微服务，使用 Java 语言开发。

将 sa-frontend 组件、sa-logic 组件和 sa-webapp 组件按照要求进行容器化。

1. 在 master 节点上编写 /root/sentiment-analyzer/Dockerfile-logic 文件构建 chinaskillsa-logic:v1.0 镜像，具体要求为

(1) 基础镜像：centos:centos7.5.1804;

- (2) 作者: Chinaskill;
- (3) 安装 Python3;
- (4) 使用 pip 安装提供的离线模块包。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写 /root/sentiment-analyzer/Dockerfile-webapp 文件构建 chinaskillsa-webapp:v1.1 镜像, 具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 go 工具;
- (4) 使用 go 构建 webapp 项目;
- (5) 开放 8080 端口。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

3. 在 master 节点上编写 /root/sentiment-analyzer/Dockerfile-frontend 文件构建 chinaskillsa-frontend:v1.1 镜像, 具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 nodejs 工具及 nginx;
- (4) 开放端口: 80;
- (5) 设置 nginx 开机自启。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

4. 在 master 节点上编写 /root/sentiment-analyzer/docker-compose.yaml 文件, 具体要求为

- (1) 容器 1 名称: sa-logic; 镜像: chinaskillsa-logic:v1.1; 端口映射: 9001:5000;
- (2) 容器 2 名称: sa-webapp; 镜像: chinaskillsa-webapp:v1.1; 端口映射: 9002:8080;
- (3) 容器 3 名称: sa-frontend; 镜像: chinaskill-frontend:v1.1; 端口映射: 9003:80。

完成后编排部署情感分析系统, 并能对输入的语句进行情感分析。然后提交 master 节点的用户名、密码和 IP 到答题框。

任务 3 基于容器的持续集成部署任务（10 分）

该公司决定采用 GitLab + Jenkins 来构建 CI/CD 环境，以缩短新功能开发上线周期，及时满足客户的需求，实现 DevOps 的部分流程，来减轻部署运维的负担，实现可视化容器生命周期管理、应用发布和版本迭代更新，请完成 GitLab + Jenkins + Kubernetes 的 CI/CD 环境部署。CI/CD 应用系统架构如图 2 所示。

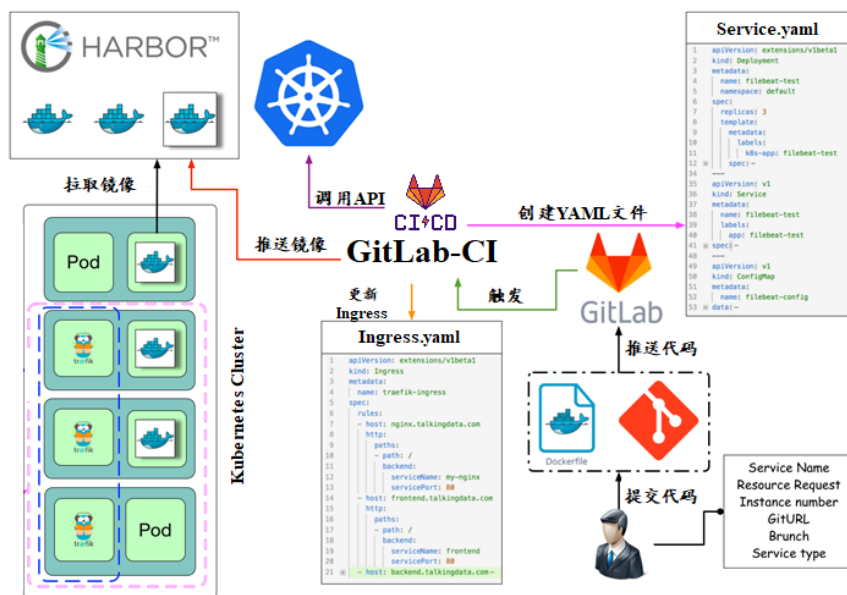


图 2 CI/CD 应用系统架构图

1. 在 master 节点上编写/root/jenkins/docker-compose.yaml 文件编排部署 Jenkins 服务，具体要求为

- (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：gpmall；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 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写/root/gitlab/docker-compose.yaml 文件编排部署 Gitlab 服务，具体要求为

- (1) 容器名称: gitlab;
- (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 gpmall, 将 gpmall 中的代码上传到 gpmall 项目中;
- (7) 实现通过 SSH 链接克隆项目。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

6. 配置 Jenkins 连接 Gitlab, 具体要求为

- (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 master 节点的用户名、密码和 IP 到答题框。

4.配置 CI/CD, 具体要求为

- (1) 新建一个流水线任务 gpmall;
- (2) 编写流水线脚本, 构建 gpmall 项目中的商城服务, 将构建后的镜像自动上传到 Harbor 仓库的 gpmall 项目, 并自动发布服务到 Kubernetes 集群的 default 命名空间下;
- (3) 配置 Webhook;
- (4) 在 Harbor 中新建公开项目 gpmall。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

5.将 gpmall 项目中的 gpmall.yaml 文件补全, 然后将 gpmall 项目代码重新上传到 Gitlab 中触发构建, 完成后提交 master 节点的用户名、密码和 IP 到答题框。

任务 4 Kubernetes 容器云平台部署与运维（10 分，本任务只公布考试范围，不公布赛题）

当应用上线以后，运维面临的一大挑战是如何能够在不影响已上线业务的情况下进行升级。在无法百分百避免版本升级故障的情况下，需要通过一种方式进行可控的版本发布，把故障影响控制在可以接受的范围内，并可以快速回退。公司决定采用基于 Istio 的灰度发布（又名金丝雀发布）来实现业务从老版本到新版本的平滑过渡，并避免升级过程中出现的问题对用户造成的影响。

请根据要求，完成 Kubernetes 的集群运维、调度、网络、存储、安全、应用生命周期管理、日志/监控等运维任务。

第三场次题目：公有云部署与运维（9）

云梦是一家科技公司，现在公司开发全面转向云原生，因此需要对公司内部项目搭建基于云原生的 Devops 相关服务。

某公有云是一家成熟的公有云提供商，为客户提供 K8S、云网络、云硬盘、云防火墙、负载均衡等服务，可创建 Web 服务，共享文件存储服务，数据库服务，数据库集群等服务。

根据上述公有云平台的特性，完成公有云中的各项运维工作。

任务 1 基础设施构建（10 分）

1.登录华为云平台，创建两台云实例，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) CPU 架构：鲲鹏计算；
- (4) 规格：kc1.xlarge.2；
- (5) 镜像：CentOS 7.5 64 位；
- (6) 系统盘：高 IO 50G 硬盘；
- (7) 公网带宽：按带宽计费，5Mbps；
- (8) 实例名称：ChinaSkill-node-1、ChinaSkill-node-2；
- (9) 登录方式：使用密码登录，密码自定义。

创建完成后，提交当前的 AK（SecretId）、SK（SecretKey）和 ChinaSkill-node-1 的实例 ID 到答题框。

2.在公有云中完成虚拟私有云的创建，具体要求为

- (1) 在贵阳区域进行创建操作；
- (2) 创建一个名为 intnetX 的内部网络：IP 地址为 172.16.1.0/16；
- (3) 创建子网名称为 intnetX-server：IP 地址为 172.16.1.0/24；
- (4) 创建子网名称为 intnetX-mysql：IP 地址为 172.16.2.0/24；

完成创建后，提交当前的 AK（SecretId）、SK（SecretKey）和 intnetX 网络的 ID 到答题框。

3.使用创建的 ChinaSkill-node-1 和 ChinaSkill-node-2 进行切换内网操作，将这两个云主机的内网，切换到 intnetX 网络的 intnetX-server 子网。切换完成后，保证该云主机还能够正常上网。操作完成后，提交 ChinaSkill-node-1 节点的用户名、密码和公网 IP 地址到答题框。

4.创建一台云数据库，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) 数据库版本：MySQL 5.7；
- (4) 实例类型：主备；
- (5) 实例规格：鲲鹏 2 核 8GB/40GB；
- (6) 虚拟私有云：intnetX-mysql；
- (7) 字符集：UTF8；
- (8) 端口：3306；
- (9) 数据库密码：Root123456；
- (10) 数据库名称：chinaskill-mysql；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和云数据库的实例 ID 到答题框。

5.创建一个安全组，具体要求为

- (1) 名称：intnetX-security；
- (2) 允许策略：只放行源地址为 172.16.1.0/24 访问 3306 端口；
- (3) 允许策略：只放行源地址为 172.16.1.0/24 使用 ping 命令访问；
- (4) 关联实例：将 intnetX-security 安全组关联至所创建的数据库中；

6.使用提供的软件包，在 ChinaSkill-node-1 和 ChinaSkill-node-2 中部署 WordPress 的 Web 服务即 LNMP+WordPress。使用云数据库 chinaskill-mysql 为后端数据库。完成 WordPress 系统上云后，将 ChinaSkill-node-1 节点的 WordPress 访问地址提交到答题框。

7.创建一个负载均衡器 chinaskill-elb，将 ChinaSkill-node-1 和 ChinaSkill-node-2 加入负载均衡的后端。设置一个可用的公网服务 IP，服务端口为 80。配置监听器，监听 80 端口。对浮动公共 IP 进行 Web 访问测试。完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和负载均衡器的 ID 到答题框。

8.创建一个云服务器备份存储库名为 server_backup，容量为 100G。完成后，将当前的 AK (SecretId)、SK (SecretKey) 和存储库 ID 到答题框。

9.创建一个公网 NAT 网关，具体配置为

- (1) 名称为：kcloud-nat；
- (2) 虚拟私有云：intnetX；

- (3) 子网: `intnetX-server`;
- (4) 规则: 内部子网地址访问外网;
- (5) 保证使用 `intnetX-server` 云服务器可以使用 `kcloud-nat` 网关访问外网。

创建完成后提交当前的 AK (`SecretId`)、SK (`SecretKey`) 和 NAT 网关的 ID 到答题框。

任务 2 系统管理与维护 (10 分)

1. 创建一个云服务器备份存储库名为 `server_backup`, 容量为 100G。将 `ChinaSkill-node-1` 云服务器制作镜像文件 `chinaskill-image`。完成后, 将当前的 AK (`SecretId`)、SK (`SecretKey`) 和存储库 ID 到答题框。

2. 创建一个负载均衡器 `chinaskill-elb`, 将 `ChinaSkill-node-1` 和 `ChinaSkill-node-2` 加入负载均衡的后端。设置一个可用的公网服务 IP, 服务端口为 80。配置监听器, 监听 80 端口。对浮动公共 IP 进行 Web 访问测试。完成后, 提交当前的 AK (`SecretId`)、SK (`SecretKey`) 和负载均衡器的 ID 到答题框。

3. 新建一个启动配置, 具体要求为

- (1) 启动配置名称: `template-exam`;
- (2) 计费模式: 按量计费;
- (3) 地域: 贵阳;
- (4) CPU 架构: 鲲鹏计算;
- (5) 规格: `kc1.xlarge.2`;
- (6) 镜像: `chinaskill-image`;
- (7) 系统盘: 高 IO 50G 硬盘;
- (8) 登录方式: 使用密码登录, 密码自定义。

创建一个伸缩组, 具体要求为

- (1) 名称: `as-exam`;
- (2) 最小伸缩数: 1; 起始实例数: 1; 最大伸缩数: 5;
- (3) 启动配置: `template-exam`;
- (4) 使用负载均衡: `chinaskill-elb`;
- (5) 移出策略: 移出最旧的实例。

为伸缩组 `as-exam` 新建告警触发策略, 具体要求为

- (1) 如果实例的内存利用率在 5 分钟内的最大值小于 40%, 且连续发生 3 次。则实例

数减少 1 台。冷却 60 秒；

(2) 如果实例的内存利用率在 5 分钟内的最大值大于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒；

(3) 如果实例的 CPU 利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例数减少 1 台。冷却 60 秒；

(4) 如果实例的 CPU 利用率在 5 分钟内的平均值大于等于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒。

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和伸缩组的实例 ID 到答题框。

4. 创建一个鲲鹏架构集群，具体要求为

- (1) 集群名称: kcloud;
- (2) 集群版本: v1.15.11;
- (3) 地域: 贵阳;
- (4) 集群管理规模: 50 节点;
- (5) 控制节点数: 3;
- (6) 节点使用子网: intnetX-server;
- (7) Pod 实例上限: 64;
- (8) 容器网段: 10.10.0.0/16。

节点配置信息要求为

- (1) 节点名称: kcloud-server;
- (2) 节点规格: kc.xlarge.2
- (3) 节点: EulerOS 2.8

创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和集群的 ID 到答题框。

5. 使用插件管理在 kcloud 容器集群中安装 dashboard 可视化监控界面。完成后，提交 dashboard 访问地址到答题框。

6. 购买云硬盘存储卷，具体配置为

- (1) 名称为: ckcloud-disk;
- (2) 集群: kcloud;
- (3) 容量: 30G;
- (4) 类型: 高 I/O;

完成后，创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和云硬盘的 ID 到答题

框。

7.在 kcloud 集群中安装 kubectl 命令，使用 kubectl 命令管理 kcloud 集群。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

8.使用提供的 helm 软件包，在 kcloud 集群中安装 helm 服务。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

9.使用容器镜像服务作为镜像仓库，将提供的 httpd-2.4.33 和 httpd-2.4.35 镜像上传至服务中。提交当前的 AK（SecretId）、SK（SecretKey）和组织名称到答题框。

10.在 kcloud 集群中创建一个 Deployment，使用开源镜像中心的 nginx:latest，配置使其可以通过外部公网进行访问。完成后提交外部访问地址到答题框。

任务 3 自动化运维（10 分，本任务只公布考试范围，不公布赛题）

编写 Python 脚本调用公有云 API 完成公有云运维工作。

第一场次题目：OpenStack 平台部署与运维（10）

某企业拟使用 OpenStack 搭建一个企业云平台，以实现资源池化弹性管理、企业应用集中管理、统一安全认证和授权等管理。

系统架构如图 1 所示，IP 地址规划如表 1 所示。

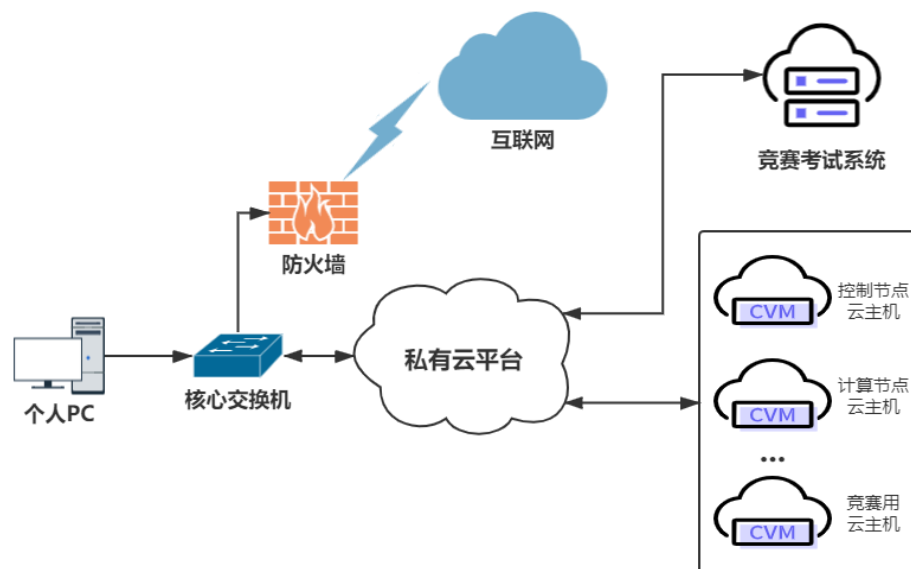


图 1 系统架构图

表 1 IP 地址规划

设备名称	主机名	接口	IP 地址	说明
云服务器1	Controller	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器2	Compute	eth0	172.129.x.0/24	Vlan x
		eth1	自定义	自行创建
云服务器3	自定义	eth0	172.129.x.0/24	用于实操题
云服务器 n				
PC-1		本地连接	172.24.16.0/24	PC 使用

说明：

- 1.竞赛使用集群模式进行，比赛时给每个参赛队提供独立的租户与用户，各用户的资源配额相同，选手通过用户名与密码登录竞赛用私有云平台，创建云主机进行相应答题；
- 2.表中的 x 为工位号；在进行 OpenStack 搭建时的第二块网卡地址根据题意自行创建；
- 3.根据图表给出的信息，检查硬件连线及网络设备配置，确保网络连接正常；
- 4.考试所需要的账号资源、竞赛资源包与附件均会在考位信息表与设备确认单中给出；
- 5.竞赛过程中，为确保服务器的安全，请自行修改服务器密码；在考试系统提交信息时，请确认自己的 IP 地址，用户名和密码。

任务 1 基础运维任务（5 分）

- 1.使用提供的用户名密码，登录提供的 OpenStack 私有云平台，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；

完成后提交控制节点的用户名、密码和 IP 地址到答题框。

2.使用提供的用户名密码，登录提供的 OpenStack 私有云平台，自行使用 CentOS7.5 镜像创建一台云主机，flavor 使用 2v_4G_50G 的配置，使用单网卡。启动后使用提供的 ansible.tar.gz 软件包在这个节点上安装 ansible 服务并配置 ansible 节点与 controller、compute 节点的 hosts 主机名映射。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

3.配置 ansible 节点无秘钥连接 controller 节点和 compute 节点，配置完成后并完成 ssh 连接两个节点的 hostname 进行测试。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

4.在 ansible 节点配置 ansible 的 hosts 文件，要求创建两个组分别为 controller 和 compute，controller 组下主机节点为 controller 节点；compute 组下主机节点为 compute。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。

5.在 compute 节点上利用空白分区划分 2 个 20G 分区。完成后提交计算节点的用户名、密码和 IP 地址到答题框。

任务 2 OpenStack 搭建任务（10 分）

1.使用提供的 openstack_ansible.tar.gz 项目包解压至 ansible 节点的/opt 目录下，然后编辑 roles 目录下 init/tasks 中的 main.yaml；编辑 group_vars 目录下的 all 文件（openstack 中的密码都设置为 000000）；编辑 install_openstack.yaml 文件，要求执行 install_openstack.yaml 文件可以在 controller 节点和 compute 节点执行 init 这个 role 来安装 iaas-pre-host。完成后提交 ansible 节点的用户名、密码和 IP 地址到答题框。（考试系统会进入你的 ansible 节点来执行 install_openstack.yaml，请确保你的环境处于正确的可执行状态）

2.在 ansible 节点，编辑 roles/mariadb/tasks 下的 main.yaml 文件和 install_openstack.yaml，在 controller 节点上安装数据库服务。自行执行 install_openstack.yaml 文件安装数据库服务。

3.在 ansible 节点，编辑 roles/keystone/tasks 下的 main.yaml 文件和 install_openstack.yaml，在 controller 节点上安装 keystone 服务。自行执行 install_openstack.yaml 文件安装 keystone 服务。

4.在 ansible 节点，编辑 roles/glance/tasks 下的 main.yaml 文件和 install_openstack.yaml，

在 controller 节点上安装 glance 服务。自行执行 install_openstack.yaml 文件安装 glance 服务。

5.在 ansible 节点, 编辑 roles/nova-controller/tasks 下的 main.yaml 文件、roles/nova-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 nova 服务。

6.在 ansible 节点, 编辑 roles/neutron-controller/tasks 下的 main.yaml 文件、roles/neutron-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 neutron 服务。

7.在 ansible 节点, 编辑 roles/dashboard/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点上安装 dashboard 服务。

8.在 ansible 节点, 编辑 roles/swift-controller/tasks 下的 main.yaml 文件、roles/swift-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 swift 服务。

9.在 ansible 节点, 编辑 roles/cinder-controller/tasks 下的 main.yaml 文件、roles/cinder-compute/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点和 compute 节点上安装 cinder 服务。

10.在 ansible 节点, 编辑 roles/heat/tasks 下的 main.yaml 文件、install_openstack.yaml 文件, 自行执行 install_openstack.yaml 文件在 controller 节点上安装 heat 服务。

任务 3 OpenStack 云平台运维 (10 分)

1.在自行搭建的 OpenStack 平台上,通过修改相关参数对 openstack 平台进行调优操作,相应的调优操作有:

(1) 预留 2048mb 内存, 这部分内存不能被虚拟机使用;

(2) 预留 10240mb 磁盘, 这部分磁盘不能被虚拟机使用。

2.在自行搭建的 OpenStack 平台上, 执行相关命令, 将 controller 节点的资源加入到 OpenStack 平台中, 即 controller 节点既是控制节点也是计算节点。

3.在自行搭建的 OpenStack 平台上, 使用 cirros 镜像创建云主机 vm1, 然后将该云主机进行手动迁移, 若原来创建在 compute 节点上的, 则迁移至 controller 节点上; 若原来创建在 controller 节点上的, 则迁移至 compute 节点上。

4.在自行搭建的 OpenStack 平台上, 使用 cinder 服务创建卷 block1, 大小为 1G, 并挂载到云主机 vm1 上使用。

5.在自行搭建的 OpenStack 平台上, 修改 nova 实例的后端存储。

6.在自行搭建的 OpenStack 平台上, 编写 heat 模板 createnet.yml, 模板作用为创建网络。

7.登录提供的私有云平台, 创建三台 centos7.5 的云主机, 使用提供的软件包, 将这三台云主机构建为 kafka 集群。

8.登录提供的私有云平台, 创建三台 centos7.5 的云主机, 使用提供的软件包, 将这三

台云主机都安装上 MariaDB 数据库服务，并配置为数据库集群，即 MariaDB_galera_cluster 数据库集群。

9.登录提供的私有云平台，再创建一台 centos7.5 的云主机，使用提供的软件包，安装 HAproxy 负载均衡服务，与上一题搭建完成的高可用数据库进行关联，完成数据库集群+负载均衡的架构。

10.登录提供的私有云平台，使用镜像 OpenStack-error-5 创建云主机“all-in-one”(账号：root；密码：000000)。创建后的云主机内有错误的 OpenStack 平台，错误现象为创建云主机时网络服务报错，请结合报错信息排查错误，使云平台可以正常使用。

任务 4 OpenStack 云平台运维开发（10 分，本任务只公布考试范围，不公布赛题）

此任务包含 2-4 个子任务，内容涉及编写 Python 脚本调用 OpenStack API 对 OpenStack 云平台进行管理和运维。使用自动化运维工具 Ansible 对云主机进行批量系统配置、批量程序部署、批量运行命令等运维操作。

第二场次题目：容器云平台部署与运维（10）

说明：完成本任务需要两台安装了 CentOS7.5 操作系统的云主机 master 和 node，镜像包中有本次容器云平台部署和运维所需的所有文件。

云梦公司技术部产品开发上线周期长，用户规模大且版本更新频繁。新版本的每次上线，产品都要承受极大的压力。引入 CICD (Continuous Integration 持续集成、Continuous Delivery 持续交付) 和灰度发布成了公司的当务之急，研发团队决定搭建基于 Kubernetes 的 CICD 环境，希望基于这个平台来实现 DevOps 的部分流程，并基于 Kubernetes 实现业务系统的灰度发布。

为了能够让云梦公司开发的 web 应用系统产品能够缩减开发周期，快速迭代版本，并实现业务从老版本到新版本的平滑过渡，避免升级过程中出现的问题对用户造成的影响。研发部决定使用微服务架构，实现基于 Kubernetes 的容器化部署、CICD 和灰度发布。

节点规划如表 1 所示。

表 1 容器云平台节点规划

节点角色	主机名	VCPUS	内存	硬盘
Master、Harbor、CICD	master	8	12G	100G
Worker Node	node	4	8G	100G

任务 1 Docker CE 及私有仓库安装任务（5 分）

1. 在 master 节点中使用提供的脚本完成 Docker CE、docker-compose 以及 Harbor 仓库的安装，导入/opt/images 目录下的所有镜像，并推送到私有仓库。
2. 在 master、node 节点完成 Kubernetes 集群的安装。
3. 在 node 节点上从仓库中拉取 skywalking:latest 和 kibana:latest 镜像，创建 docker-compse.yaml 文件，编排部署 Skywalking 服务，并设置 restart 策略。

任务 2 基于容器的 web 应用系统部署任务（10 分）

云梦公司开发了一套基于 SpringBoot+MyBatis 微服务架构的 ChinaSkillsMall 电商系统，并实现全容器化部署，ChinaSkillsMall 电商应用系统架构如图 1 所示，系统说明如表 2 所示。

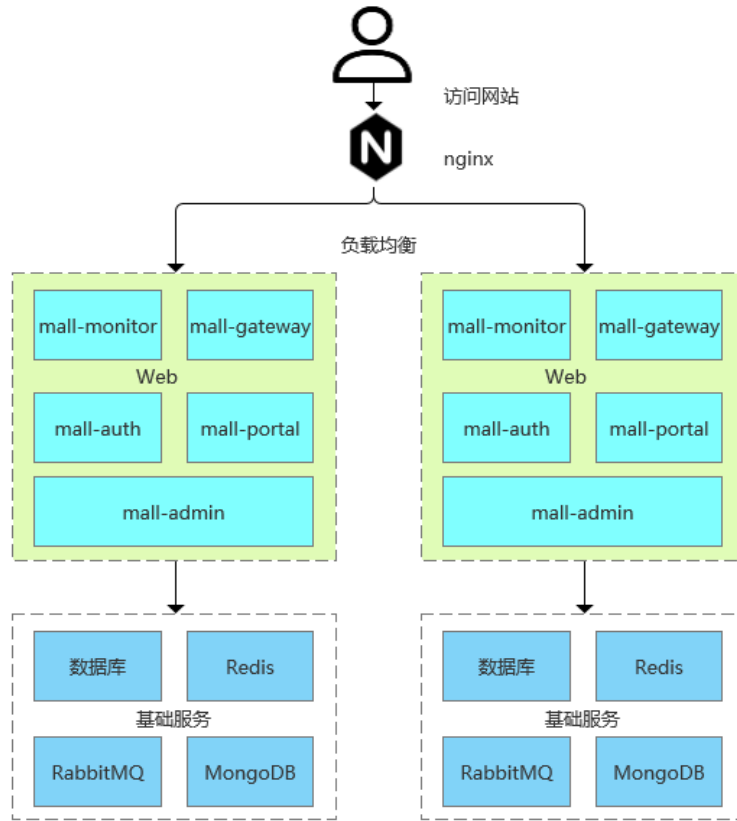


图 1 ChinaSkillsMall 电商应用系统架构图

表 2 ChinaSkillsMall 电商应用系统说明

模块	说明
mall-monitor	监控中心
mall-gateway	微服务网关
mall-auth	认证中心
mall-admin	商城后台服务
mall-portal	商城前台服务

将 MariaDB 数据库组件、Redis 消息组件、RabbitMQ 消息组件、Nacos-Registry 注册中心服务组件和前端 Nginx 组件按照要求进行容器化。

1. 在 master 节点上编写 /root/mall-swarm/Dockerfile-mariadb 文件构建 chinaskillmall-mariadb:v1.1 镜像，具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装并初始化 mariadb, 密码: root;
- (4) 设置数据库编码为 UTF-8;
- (5) 开放端口: 3306
- (6) 设置 mariadb 开机自启。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写 /root/mall-swarm/Dockerfile-redis 文件构建 chinaskillmall-redis:v1.1 镜像, 具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 redis 服务;
- (4) 修改配置文件中的 bind 127.0.0.1 为 bind 0.0.0.0;
- (5) 设置 redis 免密, 并关闭保护模式;
- (6) 开放端口: 6379;
- (7) 设置服务开机自启。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

3. 在 master 节点上编写 /root/mall-swarm/Dockerfile-nacos 文件构建 chinaskillmall-nacos:v1.1 镜像, 具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 jdk 工具和 nacos-server 工具;
- (4) 设置 nacos 服务开机自启。

完成后构建镜像, 并提交 master 节点的用户名、密码和 IP 到答题框。

4. 在 master 节点上安装 node 工具, 对 mall-admin-web 项目进行打包, 然后编写 /root/mall-swarm/Dockerfile-nginx 文件构建 chinaskillmall-nginx:v1.1 镜像, 具体要求为

- (1) 基础镜像: centos:centos7.5.1804;
- (2) 作者: Chinaskill;
- (3) 安装 nginx 服务;
- (4) 将 mall-admin-web 项目打包后的文件拷贝到/usr/share/nginx/html 目录下;

(5) 设置 nginx 服务开机自启。

完成后构建镜像，并提交 master 节点的用户名、密码和 IP 到答题框。

5. 在 master 节点上编写/root/mall-swarm/docker-compose.yaml 文件，具体要求为

(1) 容器 1 名称: chinaskillmall-admin; 镜像: mall/mall-admin:1.0-SNAPSHOT; 端口映射: 8080:8080;

(2) 容器 2 名称: chinaskillmall-auth; 镜像: mall/mall-auth:1.0-SNAPSHOT; 端口映射: 8401:8401;

(3) 容器 3 名称: chinaskillmall-gateway; 镜像: mall/mall-gateway:1.0-SNAPSHOT; 端口映射: 8201:8201;

(4) 容器 4 名称: chinaskillmall-mysql; 镜像: chinaskillmall-mariadb:v1.1; 端口映射: 3306:3306;

(5) 容器 5 名称: chinaskillmall-nacos; 镜像: chinaskillmall-nacos:v1.1; 端口映射: 8848:8848;

(6) 容器 6 名称: chinaskillmall-nginx; 镜像: chinaskillmall-nginx:v1.1; 端口映射: 8888:80;

(7) 容器 7 名称: chinaskillmall-redis; 镜像: chinaskillmall-redis:v1.1; 端口映射: 6379:6379。

完成后编排部署 ChinaSkillsMall 商城。然后提交 master 节点的用户名、密码和 IP 到答题框。

任务 3 基于容器的持续集成部署任务 (10 分)

该公司决定采用 GitLab + Jenkins 来构建 CICD 环境，以缩短新功能开发上线周期，及时满足客户的需求，实现 DevOps 的部分流程，来减轻部署运维的负担，实现可视化容器生命周期管理、应用发布和版本迭代更新，请完成 GitLab + Jenkins + Kubernetes 的 CICD 环境部署。CICD 应用系统架构如图 2 所示。

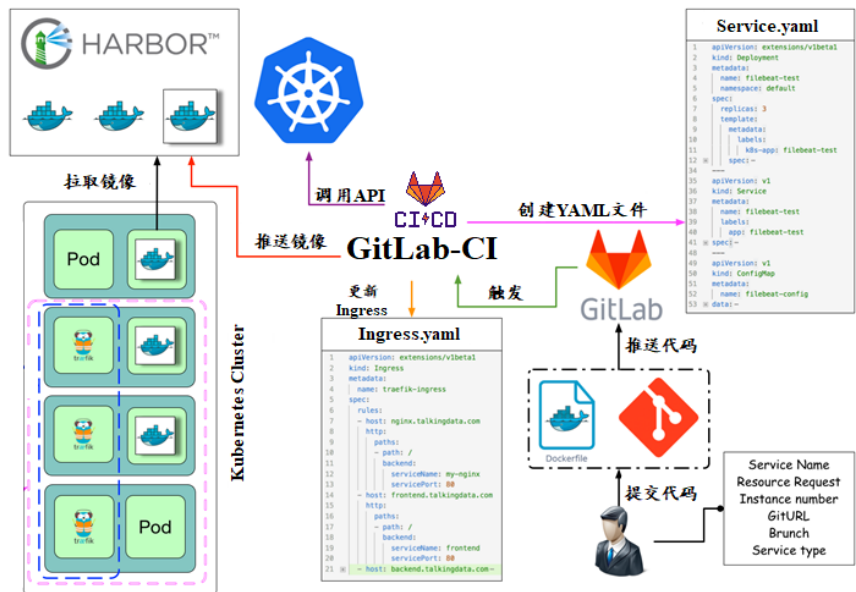


图 2 CI/CD 应用系统架构图

1. 在 master 节点上编写/root/jenkins/docker-compose.yaml 文件编排部署 Jenkins 服务，具体要求为

- (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：gpmall；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 master 节点的用户名、密码和 IP 到答题框。

2. 在 master 节点上编写/root/gitlab/docker-compose.yaml 文件编排部署 Gitlab 服务，具体要求为

- (1) 容器名称：gitlab；
- (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 gpmall, 将 gpmall 中的代码上传到 gpmall 项目中;

(7) 实现通过 SSH 链接克隆项目。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

7. 配置 Jenkins 连接 Gitlab, 具体要求为

(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 master 节点的用户名、密码和 IP 到答题框。

4.配置 CI/CD, 具体要求为

(1) 新建一个流水线任务 gpmall;

(2) 编写流水线脚本, 构建 gpmall 项目中的商城服务, 将构建后的镜像自动上传到 Harbor 仓库的 gpmall 项目, 并自动发布服务到 Kubernetes 集群的 default 命名空间下;

(3) 配置 Webhook;

(4) 在 Harbor 中新建公开项目 gpmall。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

5.将 gpmall 项目中的 gpmall.yaml 文件补全, 然后将 gpmall 项目代码重新上传到 Gitlab 中触发构建, 完成后提交 master 节点的用户名、密码和 IP 到答题框。

任务 4 Kubernetes 容器云平台部署与运维 (10 分, 本任务只公布考试范围, 不公布赛题)

当应用上线以后, 运维面临的一大挑战是如何能够在不影响已上线业务的情况下进行升级。在无法百分百避免版本升级故障的情况下, 需要通过一种方式进行可控的版本发布, 把故障影响控制在可以接受的范围内, 并可以快速回退。公司决定采用基于 Istio 的灰度发布 (又名金丝雀发布) 来实现业务从老版本到新版本的平滑过渡, 并避免升级过程中出现的问题对用户造成的影响。

请根据要求, 完成 Kubernetes 的集群运维、调度、网络、存储、安全、应用生命周期管理、日志/监控等运维任务。

第三场次题目：公有云部署与运维（10）

云梦是一家科技公司，现在公司开发全面转向云原生，因此需要对公司内部项目搭建基于云原生的 Devops 相关服务。

某公有云是一家成熟的公有云提供商，为客户提供 K8S、云网络、云硬盘、云防火墙、负载均衡等服务，可创建 Web 服务，共享文件存储服务，数据库服务，数据库集群等服务。

根据上述公有云平台的特性，完成公有云中的各项运维工作。

任务 1 基础设施构建（10 分）

1.在公有云中完成虚拟私有云的创建，具体要求为

- (1) 在贵阳区域进行创建操作；
- (2) 创建一个名为 intnetX 的虚拟私有云：IP 地址为 172.16.1.0/16；
- (3) 创建子网名称为 intnetX-server：IP 地址为 172.16.1.0/24；
- (4) 创建一个名为 net-data 的虚拟私有云：网段为 192.168.1.0/16
- (5) 创建子网名称为 net-mysql：IP 地址为 192.168.1.0/24；

完成创建后，提交当前的 AK (SecretId)、SK (SecretKey) 和 intnetX 网络的 ID 到答题框。

2.创建对等连接 intnetX-net，配置 intnetX 虚拟私有云和 net-data 虚拟机私有云连通。操作完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和对等连接的 ID 到答题框。

3.登录华为云平台，创建两台云实例，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) CPU 架构：鲲鹏计算；
- (4) 规格：kc1.xlarge.2；
- (5) 镜像：CentOS 7.5 64 位；
- (6) 系统盘：高 IO 50G 硬盘；
- (7) 虚拟网络：intnetX-server；
- (8) 实例名称：ChinaSkill-node-1、ChinaSkill-node-2；
- (9) 登录方式：使用密码登录，密码自定义。

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和 ChinaSkill-node-1 的实例 ID 到答题框。

4.创建一台云数据库，具体要求为

- (1) 计费模式：按需计费；
- (2) 地域：贵阳；
- (3) 数据库版本：MySQL 5.7；
- (4) 实例类型：主备；
- (5) 实例规格：鲲鹏 2 核 8GB/40GB；
- (6) 虚拟私有云：net-mysql；
- (7) 字符集：UTF8；
- (8) 端口：3306；
- (9) 数据库密码：Root123456；
- (10) 数据库名称：chinaskill-mysql；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和云数据库的实例 ID 到答题框。

5.使用公有云平台，申请对象存储服务，存储桶名字自定义。自行编写一个静态网页，网页的内容为“welcome to China”，使用申请的对象存储服务，托管该静态网页。托管成功后，提交你对象存储的访问域名到答题框。

6.创建一个网络 ACL，具体要求为

- (1) 名称：intnetX-acl；
- (2) 允许策略：只放行源地址为 172.16.1.0/24 访问 3306 端口；
- (3) 允许策略：只放行源地址为 172.16.1.0/24 访问 2181 端口；
- (4) 允许策略：只放行源地址为 172.16.1.0/24 访问 9092 端口；
- (5) 关联子网：关联网络 ACL 至 net-mysql 子网中；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和网络 ACL ID 到答题框。

7.创建一个 Redis 缓存实例，具体要求为

- (1) 版本号：5.0；
- (2) 实例类型：主备；
- (3) 副本数：2；
- (4) 实例规格：4G；
- (5) 子网：net-mysql；
- (6) 密码：Root123456；

创建完成后，提交当前的 AK (SecretId)、SK (SecretKey) 和实例 ID 到答题框。

8.根据 ChinaSkill-node-1 云服务器配置购买 chinaskill-node-3 云服务器上,此服务器使用 net-mysql 私有网络。使用提供的软件包安装 kafka 和 zookeeper 服务。安装完成后运行服务。完成后提交连接 chinaskill-node-3 节点的用户名、密码和公网 IP 地址到答题框。

9.在 ChinaSkill-node-1 和 ChinaSkill-node-2 云服务上部署 gpmall,mysql 使用云数据库地址,配置公有云 Redis 访问路径,使用提供的公有云软件包 gpmall-package 完成商城的上云操作。并设置 gpmall 所需 jar 软件包开启自动启动。完成后提交 ChinaSkill-node-1 的用户名、密码和公网 IP 到答题框。

任务 2 系统管理与维护 (10 分)

1.创建一个负载均衡器 chinaskill-elb,将 ChinaSkill-node-1 和 ChinaSkill-node-2 加入负载均衡的后端。设置一个可用的公网服务 IP,服务端口为 80。配置监听器,监听 80 端口。对浮动公共 IP 进行 Web 访问测试。完成后,提交当前的 AK (SecretId)、SK (SecretKey)和负载均衡器的 ID 到答题框。

2.创建一个云服务器备份存储库名为 server_backup,容量为 100G。将 ChinaSkill-node-1 云服务器制作镜像文件 chinaskill-image。完成后,将当前的 AK (SecretId)、SK (SecretKey)和存储库 ID 到答题框。

3.新建一个启动配置,具体要求为

- (1) 启动配置名称: template-exam;
- (2) 计费模式: 按量计费;
- (3) 地域: 贵阳;
- (4) CPU 架构: 鲲鹏计算;
- (5) 规格: kc1.xlarge.2;
- (6) 镜像: chinaskill-image;
- (7) 系统盘: 高 IO 50G 硬盘;
- (8) 登录方式: 使用密码登录,密码自定义。

创建一个伸缩组,具体要求为

- (1) 名称: as-exam;
- (2) 最小伸缩数: 1; 起始实例数: 1; 最大伸缩数: 5;
- (3) 启动配置: template-exam;
- (4) 使用负载均衡: chinaskill-elb;

(5) 移出策略：移出最旧的实例。

为伸缩组 `as-exam` 新建告警触发策略，具体要求为

(1) 如果实例的内存利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例数减少 1 台。冷却 60 秒；

(2) 如果实例的内存利用率在 5 分钟内的最大值大于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒；

(3) 如果实例的 CPU 利用率在 5 分钟内的最大值小于 40%，且连续发生 3 次。则实例数减少 1 台。冷却 60 秒；

(4) 如果实例的 CPU 利用率在 5 分钟内的平均值大于等于 80%，且连续发生 3 次。则实例数增加 1 台。冷却 60 秒。

创建完成后提交当前的 AK (`SecretId`)、SK (`SecretKey`) 和伸缩组的实例 ID 到答题框。

4. 创建一个鲲鹏架构集群，具体要求为

(1) 集群名称：`kcloud`；

(2) 集群版本：`v1.15.11`；

(3) 地域：贵阳；

(4) 集群管理规模：50 节点；

(5) 控制节点数：3；

(6) 节点使用子网：`intnetX-server`；

(7) Pod 实例上限：64；

(8) 容器网段：`10.10.0.0/16`。

节点配置信息要求为

(1) 节点名称：`kcloud-server`；

(2) 节点规格：`kc.xlarge.2`

(3) 节点：`EulerOS 2.8`

创建完成后提交当前的 AK (`SecretId`)、SK (`SecretKey`) 和集群的 ID 到答题框。

5. 使用插件管理在 `kcloud` 容器集群中安装 `dashboard` 可视化监控界面。完成后，提交 `dashboard` 访问地址到答题框。

6. 购买云硬盘存储卷，具体配置为

(1) 名称为：`ckloud-disk`；

(2) 集群: kcloud;

(3) 容量: 30G;

(4) 类型: 高 I/O;

完成后, 创建完成后提交当前的 AK (SecretId)、SK (SecretKey) 和云硬盘的 ID 到答题框。

7. 在 kcloud 集群中安装 kubectl 命令, 使用 kubectl 命令管理 kcloud 集群。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

8. 在 kcloud 集群中定义名为 liveness-http 探针和 liveness-tcp 探针, 监控 kcloud 集群。完成后提交连接 kcloud 集群节点的用户名、密码和公网 IP 地址到答题框。

9. 使用容器镜像服务作为镜像仓库, 将提供的 httpd-2.4.33 和 httpd-2.4.35 镜像上传至服务中。提交当前的 AK (SecretId)、SK (SecretKey) 和组织名称到答题框。

10. 在 kcloud 集群中创建一个 Deployment, 使用开源镜像中心的 nginx:latest, 配置使其可以通过外部公网进行访问。完成后提交外部访问地址到答题框。

任务 3 自动化运维 (10 分, 本任务只公布考试范围, 不公布赛题)

编写 Python 脚本调用公有云 API 完成公有云运维工作。