

# 2021 年全国职业院校技能大赛高职组云计算赛项试卷

## 【赛程名称】云计算赛项第二场-容器云

【赛程时间】2021-06-09 09:00:00 至 2021-06-09 16:00:00

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]

【适用平台】私有云

【题目 1】安装 Docker CE 和 Docker Compose[1 分]

使用提供的 centos7.5-paas 镜像启动两台云主机 master 和 node，flavor 如上表所示。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。

- 1.查看 docker 版本是否安装正确计 0.5 分
- 2.查看 docker-compose 版本安装是否正确计 0.5 分

### 【题目 2】安装私有仓库[1 分]

在 master 节点安装私有仓库，导入/opt/images 目录下所有镜像，并推送到私有仓库。完成后提交 master 节点的用户名、密码和 IP 到答题框。

- 1.查看 Harbor 关联的容器是否正常运行计 0.5 分
- 2.查看 Harbor 是否正确安装计 0.5 分

### 【题目 3】容器编排[3 分]

在 master 节点上编写/root/docker-compose.yaml 文件（需要用到的镜像包 SkyWalking.tar 在 http 服务下），具体要求如下：

- (1) 容器 1 名称：elasticsearch；镜像：elasticsearch:7.8.0；端口映射：9200:9200；
- (2) 容器 2 名称：oap；镜像：apache/skywalking-oap-server:8.0.1-es7；端口映射：11800:11800，12800:12800；
- (3) 容器 3 名称：ui；镜像：apache/skywalking-ui:8.0.1；端口映射：8082:8080。

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

- 1.docker-compose ps 命令查看容器为 UP 计 1.4 分
- 2.查看 Skywalking 首页返回正确计 1.6 分

### 【任务 2】基于 Docker 容器的 web 应用系统部署[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。

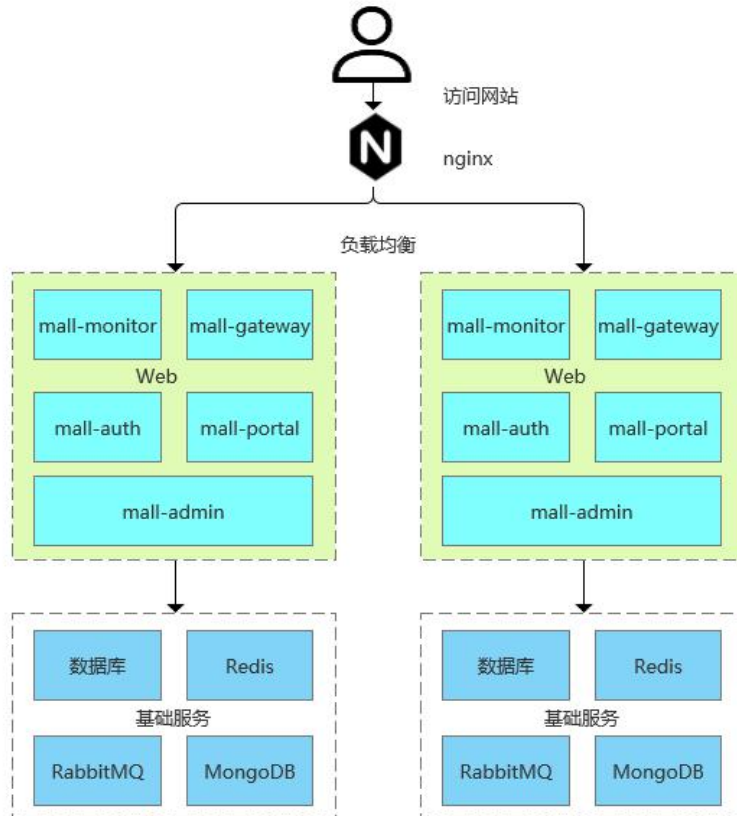


图 1 ChinaSkillsMall 电商应用系统架构图

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。（所需要的软件包 mall-swarm.tar.gz 在 http 服务中）。

【适用平台】私有云

【题目 1】容器化部署 MariaDB[2 分]

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

- (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。

- 1.docker build 容器镜像, 返回正确计 1 分
- 2.启动容器, 查看容器内部数据库字符编码, 有正确返回计 0.5 分
- 3.进入容器内部, 有正确的数据库表计 0.5 分

### 【题目 2】容器化部署 Redis[1.5 分]

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

- (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。

- 1.docker build 构建镜像, 有正确的返回计 0.4 分
- 2.启动容器, 进入容器查看 redis 的 info 信息, 有 connected\_clients:1 计 0.5 分
- 3.进入容器查看配置文件, 取消保护模式计 0.6 分

### 【题目 3】容器化部署 Nacos-Registry[2 分]

在 master 节点上编写 /root/mall-swarm/Dockerfile-nacos 文件构建

chinaskillmall-nacos:v1.1 镜像，具体要求如下：

- (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。

- 1.docker build 构建镜像，有正确的返回计 0.6 分
- 2.启动容器，查看 nacos 首页有三方 css 开始，第三方 js 结束计 1.4 分

#### 【题目 4】容器化部署前端服务[2 分]

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，具体要求如下：

- (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。

- 1.docker build 构建镜像，有正确的返回计 1 分
- 2.启动容器并进入容器，查看/usr/share/nginx/html/目录下是否存在 index.html，计 0.5 分
- 3.查看容器内 npm 的版本号是否正确计 0.5 分

#### 【题目 5】编排部署 ChinaSkillsMall 商城[2.5 分]

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

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

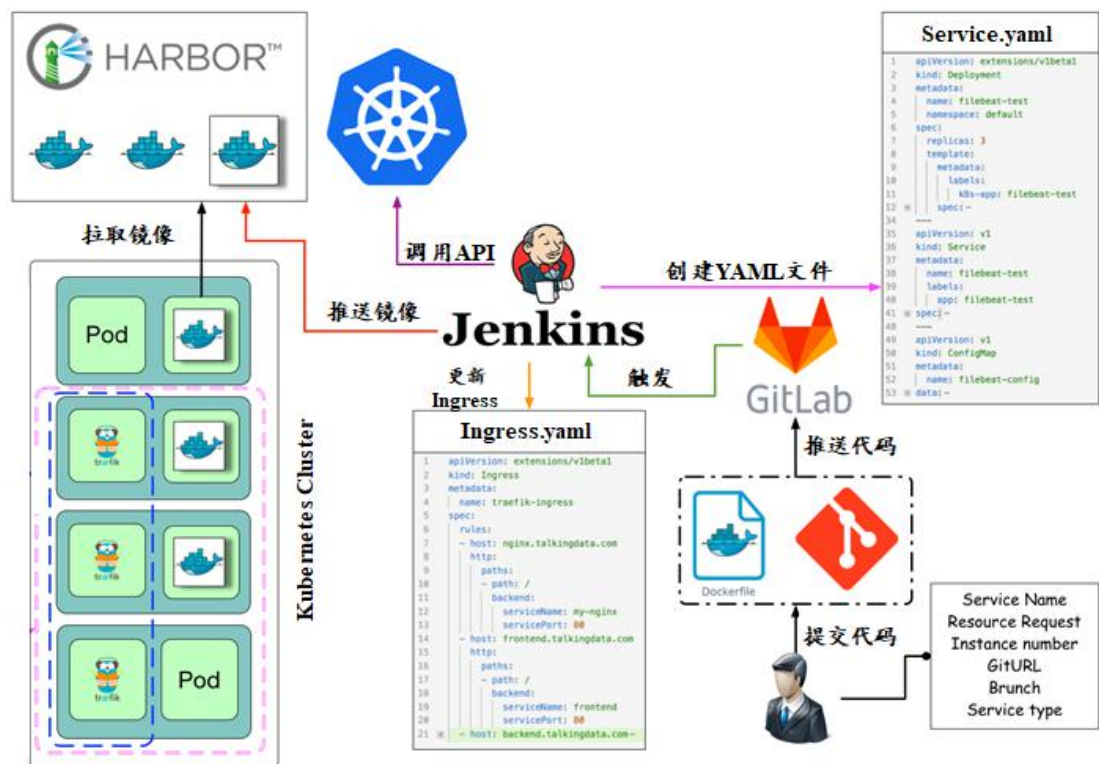
完成后编排部署 ChinaSkillsMall 商城。然后提交 master 节点的用户名、密码和 IP 到答  
题框。

1.使用 docker-compose ps 命令查看所有容器是否正常启动, 计 1 分

2.查看商城页面, 是否有正确的返回计 1.5 分

### 【任务 3】基于 Kubernetes 构建持续集成[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 (所需要的软件包 CICD\_Offline.tar 在 http 服务中)。CICD 应用系统架构如下:



【适用平台】私有云

### 【题目 1】安装 Jenkins 环境[2 分]

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。

- 1.docker-compose ps 查看 Jenkins 容器是否正常启动计 0.4 分
- 2.查看 http://localhost:8080/pluginManager/installed 是否有正确返回计 1.2 分
- 3.进入 Jenkins 容器查看容器版本是否正确计 0.4 分

**【题目 2】安装 Gitlab 环境[1 分]**

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。

- 1.使用 docker-compose ps 命令查看 gitlab 是否正常启动计 0.4 分
- 2.使用 git clone http://master:81/root/springcloud.git /data/springcloud && ls /data/springcloud 命令, 查看是否有正确的返回计 0.6 分

**【题目 3】配置 Jenkins 连接 Gitlab[1 分]**

配置 Jenkins 连接 Gitlab, 具体要求如下:

- (1) 设置 Outbound requests;
- (2) 生成 “Access Tokens” 并命名为 jenkins;
- (3) 设置 Jenkins 取消对'/project' end-point 进行身份验证;
- (4) 测试 Jenkins 与 Gitlab 的连通性。

完成后提交 master 节点的用户名、密码和 IP 到答题框。

- 1.查看 http://localhost:8080/credentials/界面是否有 GitLab API token 返回计 1 分

**【题目 4】配置 Jenkins 连接 maven[1 分]**

配置 Jenkins 连接 maven, 具体要求如下:

- (1) 采用 docker in docker 的方式在 Jenkins 内安装 maven;
- (2) 在 Jenkins 中配置 maven 信息。

完成后提交 master 节点的用户名、密码和 IP 到答题框。



- 1.进入 Jenkins 容器查看 mvn 的版本信息是否正确计 0.6 分
- 2.查看 `curl -L http://localhost:8080/configureTools/ |grep local` 是否有 `value="/usr/local/maven"` 返回计 0.4 分

**【题目 5】配置 CI/CD[1 分]**

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。

1. 编写流水线脚本，使用命令查看 `curl -u springcloud:000000 http://master:8080/job/springcloud/configure`，有正确的返回计 1 分

**【题目 6】触发 CI/CD[4 分]**

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。

- 1.使用命令 `kubectl -n springcloud get service` 查看 service 有正确的 config 和 gateway 计 0.4 分
- 2.使用命令 `kubectl -n springcloud get pods` 查看 pods，有正确的 config 和 gateway 计 0.4 分
3. 查看 `curl -L http://localhost:8080/job/Springcloud/wfapi/runs?fullStages=true|grep -o "status":"SUCCESS"`，返回"status":"SUCCESS"计 0.8 分
- 4.`curl http://master:30010/`返回 Accumulation account 计 1 分
- 5.能正确 pull config 和 gateway 镜像计 1 分
- 6.进入 Jenkins 容器查看/root/.m2/repository 内容有 aopalliance 和 commons-beanutils 计 0.4 分

## 【任务 4】Kubernetes 容器云平台部署与运维[10 分]

【适用平台】私有云

### 【题目 1】健康检查--ExecAction[0.5 分]

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

(1) Pod 名称：liveness-exec；

(2) 命名空间：default；

(3) 镜像：busybox；容器名称：liveness；

(4) 容器启动时运行命令 “touch /tmp/healthy; sleep 60; rm -rf /tmp/healthy; sleep 600”，此命令在容器启动时创建/tmp/healthy 文件，并于 60 秒之后将其删除；

(5) 存活性探针运行 “test -e /tmp/healthy” 命令检查文件的存在性，若文件存在则返回状态码为 0，表示成功通过测试；

(6) 启动后延时 5 秒开始运行检测；

(7) 每隔 5 秒执行一次 liveness probe。

完成后使用该 yaml 文件创建 Pod，并提交 master 节点的用户名、密码和 IP 到答题框。

1.使用 `kubectly apply -f liveness_exec.yaml && kubectly get pods` 命令查看，能查看到 liveness-exec 这个 pod 并且正常运行计 0.1 分

2.查看 liveness-exec 的详细信息，有正确的返回计 0.4 分

### 【题目 2】更新证书[1.5 分]

Kubernetes 默认的证书有效期只有一年时间，对于某些场景下一个足够长的证书有效期是非常有必要的。请将 Kubernetes 集群证书的有效期延长至 10 年。完成后提交 master 节点的用户名、密码及 IP 到答题框。（需要用到的软件包 `kubernetes_v1.18.1.tar.gz` 在 http 服务下）

1.使用命令 `kubeadm alpha certs check-expiration|grep 9y|awk '{print$1" "$7}'` 查看证书时间，有正确的返回 `admin.conf 9y、apiserver 9y、apiserver-etc-client 9y` 计 1.5 分，每个计 0.5 分

### 【题目 3】自定义资源管理[1 分]

在 Kubernetes 中一切都可视作资源，通过自定义资源我们可以向 Kubernetes API 中增

加新资源类型。在 master 节点/root 目录下编写 yaml 文件 resource.yaml，具体要求如下：

- (1) 自定义资源名称： crontabs.stable.example.com；
- (2) 指定自定义资源作用范围为命名空间；
- (3) 指定每个版本都可以通过 served 标志来独立启用或禁止；
- (4) 指定其中有且只有一个版本必需被标记为存储版本 v1。

完成后使用该 yaml 文件创建自定义资源，并提交 master 节点的用户名、密码和 IP 到答题框。

- 1.使用命令 kubectl get crd 查看，返回 crontabs.stable.example.com 计 0.1 分
2. 使用命令 kubectl describe crd crontabs.stable.example.com 查看，有正确的返回 Group: stable.example.com、Scope: Namespaced、v1 共计 0.9 分

#### 【题目 4】HPA 管理[0.5 分]

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

- (1) HPA 名称： deployment-hpa；
- (2) 命名空间： default；
- (3) 基于 deployment 进行伸缩，副本数伸缩范围： 1--10；
- (4) 期望每个 Pod 的 CPU 和内存使用率为 50%。

完成后使用该 yaml 文件创建 HPA，并提交 master 节点的用户名、密码和 IP 到答题框。

1. 运行 /root/deployment-hpa.yaml 并 kubectl get HorizontalPodAutoscaler，返回 deployment-hpa Deployment 计 0.2 分
- 2.使用 kubectl describe horizontalpodautoscaler.autoscaling/deployment-hpa 命令查看，有正确返回计 0.3 分

#### 【题目 5】NetworkPolicy 管理 [0.5 分]

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

- (1) NetworkPolicy 名称： default-deny；
- (2) 命名空间： default；
- (3) 默认禁止所有入 Pod 流量。

完成后使用该 yaml 文件创建 NetworkPolicy，并提交 master 节点的用户名、密码和 IP

到答题框。

- 1.使用命令 `kubectl apply -f network-policy-deny.yaml && kubectl get networkpolicy` 查看，有 `default-deny` 返回计 0.2 分
- 2.使用命令 `kubectl describe networkpolicy default-deny` 查看，有 `Selected pods are isolated for ingress connectivity` 返回计 0.3 分

**【题目 6】修改 Pod 数量限制[0.5 分]**

Kubernetes 默认每个节点只能启动 110 个 Pod，由于业务需要，将每个节点默认限制的 Pod 数量改为 200。完成后提交 master 节点的用户名、密码和 IP 到答题框。

- 1.使用命令 `kubectl describe node master node|grep -w pods|grep 200|wc -l` 查看，返回 4 计 0.5 分

**【题目 7】修改 NodePort 端口范围[0.5 分]**

Kubernetes 以 NodePort 方式暴露服务，默认的端口范围为 30000-32767，将 NodePort 的端口范围修改为 20000-65535。完成后提交 master 节点的用户名、密码和 IP 到答题框。

1. 使用 `kubectl describe pod $apiserver_pods -n kube-system` 查看，返回 `--service-node-port-range=20000-65535` 计 0.5 分

**【题目 8】金丝雀发布-安装 Istio[1 分]**

由于 Kubernetes 使用的 IPVS 模块需要系统内核版本支持，试使用提供的软件包（`Canary_v1.0.tar.gz` 在 http 服务下）将系统内核进行升级，在 Kubernetes 集群上完成 Istio 的安装，并将 default Namespace 设置自动注入。完成后提交 master 节点的用户名、密码及 IP 到答题框。

- 1.使用命令 `kubectl -n istio-system get all` 查看，有 `service/istio-egressgateway ClusterIP |, deployment.apps/istio-egressgateway 1/1, replicaset.apps/istiod` 计 0.3 分
- 2.使用命令 `kubectl get ns --show-labels=true|grep default` 查看，返回 `istio-injection=enabled` 计 0.3 分
- 3.使用命令 `uname -r` 查看系统内核版本 `5.4.119-1.el7.elrepo.x86_64` 计 0.4 分

**【题目 9】** 金丝雀发布-流量控制[0.5 分]

使用赛项提供的文件(istio-1.9.5/samples/helloworld/helloworld.yaml)在 default 命名空间下完成 helloworld 服务的部署，然后设置路由规则来控制流量分配，创建一个虚拟服务 helloworld；再创建一个目标规则 helloworld，将 10% 的流量发送到金丝雀版本(v2)。完成后提交 master 节点的用户名、密码及 IP 到答题框。

1. 使用命令 `kubectl get VirtualService, DestinationRule -n virtualservice.networking.istio.io/helloworld -o yaml`，返回 `VirtualService: helloworld`，`DestinationRule: helloworld` 计 0.2 分
2. 使用命令 `kubectl describe VirtualService helloworld | grep Route -A 10 | xargs`，返回 `Subset: v1 Weight: 90, Subset: v2 Weight: 10` 计 0.3 分

**【题目 10】** 金丝雀发布-熔断[0.5 分]

使用赛项提供的文件(istio-1.9.5/samples/httpbin/httpbin-fortio.yaml)在 default 命名空间下完成 httpbin 服务的部署，创建一个目标规则 httpbin，在调用 httpbin 服务时应用熔断设置，具体要求为

- (1) 定义到目标主机的 HTTP1/TCP 最大连接数为 1；
- (2) 定义针对一个目标的 HTTP 请求的最大排队数量为 1；
- (3) 定义对某一后端的请求中，一个连接内能够发出的最大请求数量为 1。

完成后提交 master 节点的用户名、密码及 IP 到答题框。

1. 使用命令 `kubectl describe DestinationRule httpbin` 查看，返回 `maxConnections: 1, http1MaxPendingRequests: 1, maxRequestsPerConnection: 1` 计 0.5 分

**【题目 11】** 金丝雀发布-流量镜像[1 分]

使用赛项提供的文件(istio-1.9.5/samples/httpbin/httpbin-sleep.yaml)在 default 命名空间下完成 httpbin-v1、httpbin-v2 以及 sleep 服务的部署，创建一个虚拟服务 httpbin 和一个目标规则 httpbin，将所有流量路由到 httpbin-v1 服务，然后将 100% 的相同流量镜像（即发送）到 httpbin-v2 服务。完成后提交 master 节点的用户名、密码及 IP 到答题框。

1. 使用命令 `kubectl describe virtualservice httpbin` 查看，返回有 `Mirror Percent: 100, Weight: 100` 计 1 分

**【题目 12】** 金丝雀发布-Ingress Gateway [1.5 分]

使用赛项提供的文件(istio-1.9.5/samples/httpbin/httpbin.yaml)在 default 命名空间下完成 httpbin 服务的部署，在 80 端口为 HTTP 流量配置一个网关 httpbin-gateway，并为 httpbin 服务创建了虚拟服务配置 httpbin，包含 1 个路由规则，允许流量流向路径/headers，并允许通过浏览器访问服务。完成后提交 master 节点的用户名、密码及 IP 到答题框。

- 1.使用命令 `kubectl get virtualservice` 查看，返回有 httpbin [httpbin-gateway] [\*]计 0.5 分
- 2.使用命令 `kubectl describe virtualservice httpbin` 查看，返回有 Prefix: /headers 计 0.5 分
- 3.使用命令 `curl -s -I -HHost:httpbin.example.com "http://node:$(kubectl -n istio-system get service istio-ingressgateway -o jsonpath='{.spec.ports[?(@.name=="http2")].nodePort}"/headers"` 查看。返回有 HTTP/1.1 200 OK 计 0.5 分

**【题目 13】** 污点与容忍[0.5 分]

请将 master 节点设置为污点，策略设置为资源尽量不调度到污点节点中去。完成后提交 master 节点的用户名、密码和 IP 到答题框。

- 1.使用命令 `kubectl describe node master` 查看，返回有 PreferNoSchedule 计 0.5 分