

# 首届世界职业院校技能大赛

## “云计算”竞赛赛卷

场景描述：某初创互联网公司规划搭建企业私有云，并将业务网站部署至私有云平台，拟将该任务交给工程师 A 与 B。工程师 AB 相互配合搭建私有云平台、申请云主机、安装部署数据库、网络、存储等云服务，最终将企业应用部署在私有云平台上并优化。

随着公司业务类型增加，公司拟部署 K8S 平台，将业务网站迁移至 K8S 平台，企业应用全容器化部署，并实现企业应用的持续集成与持续发布。

随着公司业务量的增加，公司决定使用混合容器云架构，完成费用成本评估与项目预算后，工程师在本地和公有云上均部署 K8S 集群，完成电商应用微服务集群部署，实现流量控制。

### 第一场次：私有云（30 分）

公司拟使用 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。

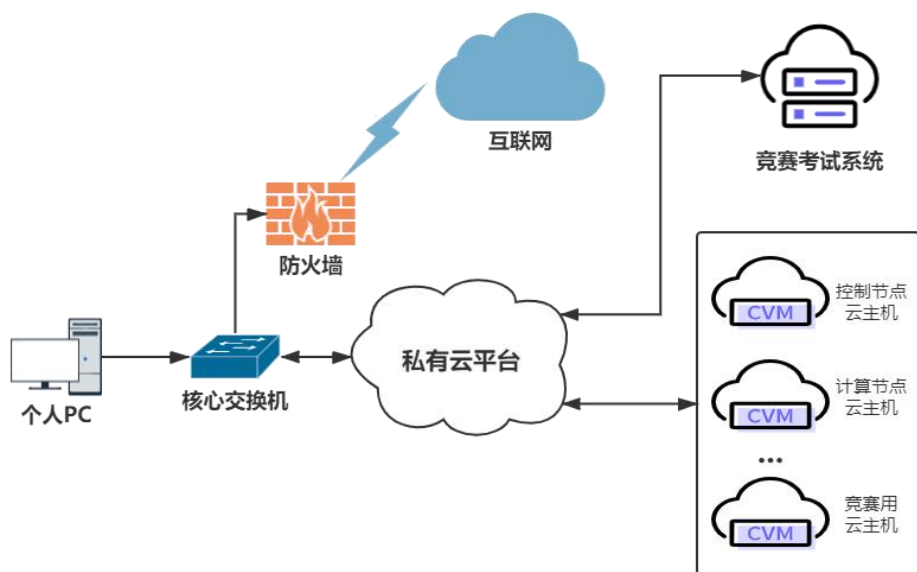


图 1 系统架构图

表 1 IP 地址规划表

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	
... 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. 竞赛所需要的账号资源、竞赛资源包与附件均会在竞赛工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，须自行修改服务器密码；在竞赛系统提交信

息时，须确认自己的 IP 地址，用户名和密码。

## 任务一、私有云服务搭建（10分）

### 题目 1. 基础环境搭建（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 CentOS7.9 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；

2.设置 `controller` 节点可以无密钥访问 `compute` 节点。

3.使用提供的 `http` 服务地址，在 `http` 服务下，存在 `centos` 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`。

4.在 `controller` 节点上部署 `chrony` 服务器，允许其他节点同步时间，启动服务并设置为开机启动；在 `compute` 节点上指定 `controller` 节点为上游 `NTP` 服务器，重启服务并设为开机启动。

5.在 `compute` 节点上利用空白分区划分 2 个 20G 分区。

### 题目 2. OpenStack 平台搭建（5分）

需要完成 `openstack` 平台的安装搭建及运维任务（已提供安装脚本框架），表 2 提供了在安装过程中需要使用的相关服务参数。

表 2 密码表

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	enp9s0 (外网卡名)

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，安装 `heat` 编排服务并完成相关配置。

11.使用提供的脚本框架 `iaas-install-cloudkitty.sh`，填充脚本，安装 `cloudkitty` 计费服务并完成相关配置。

12.使用提供的脚本框架 `iaas-install-blazar.sh`，填充脚本，安装增加 `Blazar`-资源预订服务并完成相关配置。

## 任务二、私有云服务运维（10分）

### 题目 3. OpenStack 云平台运维（4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,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。

3.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。

4.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。

5.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。

6.在 OpenStack 私有云平台，创建一台云主机，使用提供的软件包，编写一键部署脚本，要求可以一键部署 gpmall 商城应用系统。

7.使用 manila 共享文件系统服务，使 manila 为多租户云环境中的共享文件系统提供统一的管理服务。

8.使用 Blazar 服务,使得管理员可以在 OpenStack 中为虚拟(实例,卷等)和物理(主机,存储等)的不同资源类型提供资源预留的能力。

9.使用 cloudkitty 计费服务，处理来自不同监控指标后端的数据并

进行计费规则创建。以达到费用核算目的。

10.使用提供的 `iaas-error1` 镜像创建云主机，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。

#### 题目 4. 应用部署（3分）

1.使用私有云平台，申请云主机，安装部署数据库服务并设置开机自启。

2.使用私有云平台，申请云主机，安装部署 `Redis` 服务并设置开机自启。

3.使用私有云平台，申请云主机，安装部署 `Nginx` 服务。

4.使用私有云平台，申请云主机，安装部署 `zookeeper` 和 `Kafka` 服务。

5.使用私有云平台，申请云主机，配合上述已安装的服务，部署企业应用。

#### 题目 5. 应用架构升级（3分）

1.将企业应用的数据库升级为数据库高可用集群即 `mariadb_galera_cluster` 集群。

2.为数据库集群增加一个负载均衡 `HA` 服务，要求访问三个数据库节点的权重为 3, 3, 4。

3.将企业应用的 `Redis` 服务升级为主从 `Redis` 服务。

4.升级部署方式为 `Ansible` 自动化部署，编写 `Ansible` 剧本，要求执行 `Ansible` 剧本可以一键部署企业集群应用。

## 任务三、私有云运维开发（10分）

### 题目 6. OpenStack 自动化运维

1. 使用 Python 语言，基于 OpenStack APIs，封装 flavor\_manager 的 Python 类，实现 OpenStack 云主机类型管理的增、删、查、改，以 json 格式返回操作结果，完成后提交实现代码文件。

2. 使用 Python 语言，基于 OpenStack APIs，封装 server\_manager 的 Python 类，实现 OpenStack 主机管理的增、删、查、改，以 json 格式返回操作结果，完成后提交实现代码文件。

3. 使用 Python 语言，基于 OpenStack APIs，编写镜像管理程序，实现所有镜像查询与修改，完成后提交实现代码文件。

4. 使用 Python 语言，基于 Python click 框架，对接云主机类型管理、云主机管理、镜像管理的程序，实现自定义的命令行管理工具，命令同 openstack 命令保持一致，并完成单元测试 python 代码编写，完成后提交实现代码文件。

## 第二场次：容器云（40分）

### 任务一、容器云服务搭建

#### 题目 1.容器云平台搭建（5分）

- 1.在 master 和 node 节点完成 Kubernetes 集群的安装。
- 2.在 Kubernetes 集群中完成 KubeVirt 环境的安装。
- 3.在 Kubernetes 集群中完成服务网格 Istio 环境的安装。
- 4.集群部署完成后使用 kubectl 自带排序功能列出集群内所有的



Pod 并以 name 字段排序。

## 任务二、容器云应用部署（15分）

题目 2.基于容器的 web 应用系统部署任务（7分）

公司开发了一套基于 Spring Cloud 的分布式系统套件，并实现全容器化部署，系统主要模块如表 3 所示：

表 3 系统模块表

模块	说明
Config	配置中心
Gateway	微服务网关
Eureka	注册中心
Spring Boot Admin	监控和管理的源软件
Zipkin	分布式实时数据追踪系统
Hello Service	服务注册

将以上组件按照要求进行容器化。

1. 编写 Dockerfile 基于 centos:7.9.2009 基础镜像制作 Config 镜像，生成镜像名为 chinaskill-config:v1.0，并推送其到私有仓库。
2. 编写 Dockerfile 基于 centos:7.9.2009 基础镜像制作 Gateway 镜像，生成镜像名为 chinaskill-gateway:v1.0，并推送其到私有仓库。
3. 编写 Dockerfile 基于 centos:7.9.2009 基础镜像制作 Eureka 镜像，生成镜像名为 chinaskill-eureka:v1.0，并推送其到私有仓库。
4. 编写 Dockerfile 基于 centos:7.9.2009 基础镜像制作 Spring Boot Admin 镜像，生成镜像名为 chinaskill-admin:v1.0，并推送其到私有仓库。
5. 编写 Dockerfile 基于 centos:7.9.2009 基础镜像制作 Zipkin 镜像，生成镜像名为 chinaskill-zipkin:v1.0，并推送其到私有仓库。

6. 编写 Dockerfile 基于 centos:7.9.2009 基础镜像制作 Hello Service 镜像，生成镜像名为 chinaskii-hello:v1.0，并推送其到私有仓库。

7. 编写 docker-compose.yaml 文件，使用上述镜像编排部署 Spring Cloud 分布式系统。

### 题目 3. 基于 Kubernetes 构建 CICD (8 分)

公司决定采用 GitLab + Jenkins + Harbor + Kubernetes 架构来构建 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，完成 CICD 环境部署。CICD 应用系统架构如图 2 所示：

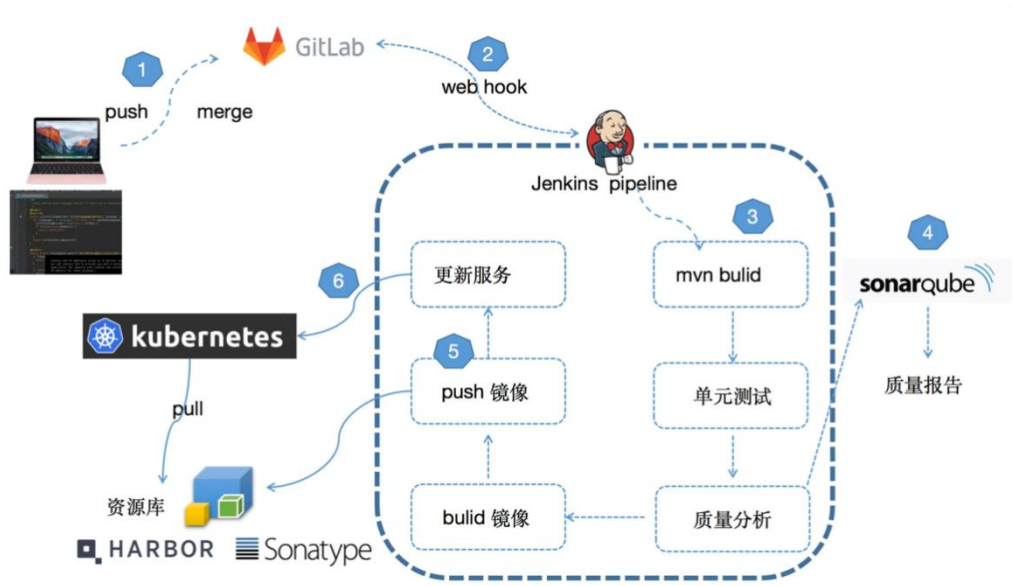


图 2 CICD 应用系统架构图

1. 完成 Piggy Metrics 项目代码的编写。
2. 在 Kubernetes 集群 devops 命名空间下完成 GitLab 程序的部署，使用 NodePort 方式将 GitLab 服务的 80 端口对外暴露为 30081。

3. 在 Kubernetes 集群 devops 命名空间下完成 Jenkins 服务的部署，使用 NodePort 方式将 Jenkins 服务的 8080 端口对外暴露为 30880。
4. 在 Jenkins 中完成 GitLab 工具和 Maven 工具的配置。
5. 在 GitLab 中完成 webhooks 的配置。
6. 在 Jenkins 中新建流水线任务 springcloud，并编写 Jenkinsfile，要求整个构建流程的阶段包含 test、build、release、review 和 deploy。
7. 将编写的 Piggy Metrics 项目代码推送到 GitLab 仓库并触发自动构建，完成代码编译、镜像构建、镜像推送和服务发布的自动化实现。

### 任务三、容器云服务运维（10分）

#### 题目 4.Kubernetes 基于容器的运维（5分）

1.将 master 节点设置为污点，并且重新调度该节点上所有允许的 Pods。

2.在 Kubernetes 集群 default 命名空间下创建 Pod，具体要求如下：

- （1）Pod 名称：nginx；
- （2）指定镜像 nginx:1.17；
- （3）容器重启策略：always；

完成该 Pod 的创建。

3.为 master 节点打上标签 “disktype=ssd” 和 “exam=chinaskill”，然后在 Kubernetes 集群 default 命名空间下创建 Deployment，具体要求如下：

- （1）Deployment 名称：nginx-deployment；
- （2）指定镜像 nginx:latest；

(3) 要求 Pod 只能调度到具有标签 “disktype=ssd” 的节点上;

(4) 具有标签 “exam=chinaskill” 的节点优先被调度。

完成该 Deployment 的创建。

4.在 Kubernetes 集群中创建一个集群角色，具体要求如下:

(1) 集群角色名称: deployment-clusterrole;

(2) 该角色拥有对 Deployment、Daemonset、StatefulSet 的创建权限。

完成该集群角色的创建。

5.在 Kubernetes 集群中创建一个网络策略，具体要求如下:

(1) 网络策略名称: exam-network;

(2) 针对 namespace test 下的 Pod，只允许相同 namespace 下的 Pod 访问，并且可访问 Pod 的 9000 端口。

完成该网络策略的创建。

6.Kubernetes 默认的证书有效期只有一年时间，请延长 Kubernetes 证书的延有效期，具体要求如下:

(1) 将 Kubernetes 集群证书的有效期延长至 100 年;

(2) 重编译 Kubernetes 源码;

(3) 完成 kubeadm 的变更。

完成 Kubernetes 证书的变更。

7.在 Kubernetes 集群、istio 网格服务中完成 microservice-istio 微服务应用部署，具体要求如下:

(1) 编译 microservice-istio 应用代码，完善 docker-build.sh 脚本，

构建微服务镜像并上传。

(2) 编写 `microservice-istio` 部署 `yaml`，完成 `microservice-istio` 服务网格部署。

请完成请求路由的创建。

8. 为 `microservice-istio` 应用创建服务熔断策略，具体要求如下：

(1) 编写 `DestinationRule` 熔断器 `circuit-breaker.yaml` 规则，设置中指定 `maxConnections` 为 1、`http1MaxPendingRequests` 为 1

(2) 通过 `fortio` 发起网络请求，请求以 3 个并发连接 (`-c 3`) 和 30 次请求 (`-n 30`)，输出熔断测试信息。

题目 5. `Kubernetes` 基于虚拟机的运维 (5 分)

根据 `Garnter` 的最新预测，到 2022 年将会有 75% 的生产应用全部跑在容器环境之上。由于技术原因，某公司的多数架构仍运行在虚拟机上。经过技术调研，`KubeVirt` 支持完全的虚拟机纳管，可以完美迁移因为内核版本过于陈旧或者及语言问题而无法迁移到容器的部分应用，所以公司决定将业务全部迁移到 `Kubernetes` 集群中。

1. 在 `KubeVirt` 中启用实时迁移功能，以实现将一台正在运行的虚拟机实例从一个节点迁移到另一个节点，期间工作负载继续运行且能保持持续访问。

2. 使用提供的 `OpenStack qcow2` 镜像在 `Kubernetes` 集群 `default` 命名空间下创建 `VMI`，具体要求如下：

(1) `VMI` 名称: `chinaskill-vmi`;

(2) 运行策略: `Manual`;

(3) 磁盘驱动: virtio。

完成该 VMI 的创建。

3. 在 master 节点上编写 YAML 文件 vmi-sshkey.yaml 为虚拟实例 chinaskill-vmi 进行动态密钥注入，具体要求如下：

(1) 使用 qemuGuestAgent 将访问凭证 api 附加到 chinaskill-vmi。

完成后运行该 YAML 文件并使用 SSH 验证 VMI 的连通性。

## 任务四、容器云运维开发（10分）

题目 6. Kubernetes 运维开发（10分）

1. 申请一台云主机，在云主机上完成 Python3.6.8 环境和 Kubernetes 库的安装。

2. 编写 Python 代码，列出 Kubernetes 集群 kube-system 命名空间下所有的 Pod、Deployment 和 Service。

3. 编写 Python 代码，在 Kubernetes 集群 default 命名空间下使用 nginx:1.17 镜像创建一个名为 test 的 Deployment，副本数为 3。

4. 编写 Python 代码，修改上一题 Deployment 中的镜像文件为 nginx:1.18，并对该 Deployment 进行滚动更新。

5. 使用 Python 编写一个 Kubernetes 调度器，创建一个 API 调用的 client，通过 stream 监听 Pod 的变化，当检测到有 Pending 状态的 Pod 时，通过编写的调度器进行调度，scheduler 方法就是随机选择一个可用的节点，通过 create\_namespaced\_binding 方法调用 API 并绑定 Pod 和主机。

## 第三场次：混合云（30分）

### 任务一、公有云集群搭建（15分）

#### 题目 1.公有云成本预算（5分）

1.根据应用系统规划需求，使用公有云的费用预估工具，完成公有云资源选择、费用统计与成本预估，计算最佳云效率。

#### 题目 2.公有云网络创建（5分）

1.在公有云中完成虚拟私有云的创建，创建一个名为 `intnetX` 的内部网络：IP 地址为 `172.16.0.0/16`；创建子网名称为 `intnetX-sql`：IP 地址为 `172.16.1.0/24`；

2.购买弹性负载均衡 `elb`，并申请公网 IP 进行访问。

#### 题目 3.公有云容器集群搭建（5分）

管理公有云云容器引擎集群，完成以下搭建与运维操作。

1.使用 `intnetX` 内部网络在公有云中创建一个容器集群 `kcloud`。

2.在 `kcloud` 集群中安装 `kubectl` 命令，使用 `kubectl` 命令管理 `kcloud` 集群。

3.在 `kcloud` 集群中部署 `helm` 工具，使用提供的 `wordpress charts` 包，编写 `yaml` 文件创建 `wordpress` 所需 `pv` 卷。通过 `helm` 工具部署 `wordpress` 服务。

4.在 `kcloud` 集群中部署应用 `istio` 服务网格，使用提供的 `V1` 和 `V2` 两个容器镜像配置灰度发布，并设置 `V1` 版本访问流量为 `40%`，`V2` 版本访问流量为 `60%`。

## 任务二、混合云集群配置（5分）

题目 4.本地容器云与公有云对接

1. 配置 openstack 平台中的 kubernetes 集群，使用 kubekey 安装 kubedns 等相关组件，实现本地 kubernetes 集群与公有云集群对接。

## 任务三、混合云集群应用部署（10分）

题目 5.混合云服务案例（10分）

Bookinfo 是一个在线书店微服务应用，Bookinfo 应用分为四个单独的微服务：（1）productpage. 这个微服务会调用 details 和 reviews 两个微服务，用来生成页面。（2）details. 这个微服务中包含了书籍的信息。（3）reviews. 这个微服务中包含了书籍相关的评论。它还会调用 ratings 微服务。（4）ratings. 这个微服务中包含了由书籍评价组成的评级信息。

reviews 微服务有 3 个版本：

v1 版本不会调用 ratings 服务。

v2 版本会调用 ratings 服务，并使用 1 到 5 个黑色星形图标来显示评分信息。

v3 版本会调用 ratings 服务，并使用 1 到 5 个红色星形图标来显示评分信息。

实现该应用服务网格的部署，利用 Istio 控制平面为应用提供服务路由、监控数据收集以及策略实施等功能。部署架构如图 3 所示：



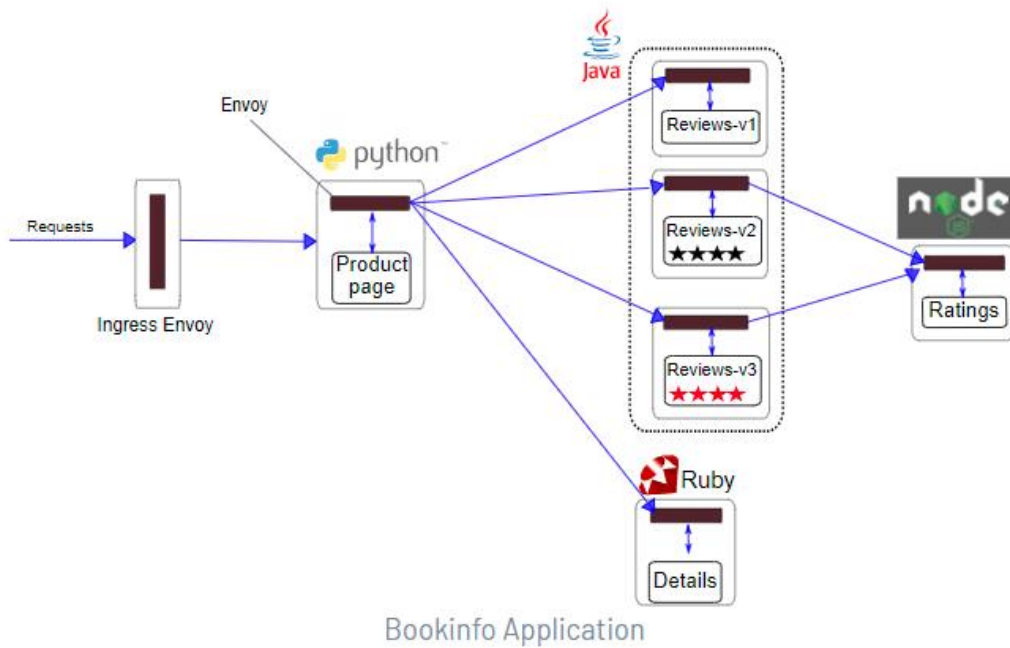


图 3 部署架构图

1.编写 Dockerfile 构建 productpage:v1.0 镜像，具体要求如下：

- (1) 基础镜像：centos:7.9.2009
- (2) 安装 Python3.6.8 工具；
- (3) 安装 productpage 服务并设置开机自启。

2.规划应用微服务部署架构，在混合云容器集群中通过修改微服务应用的 yaml 文件来完成 bookinfo 微服务应用的部署。

3.在混合云容器集群中配置流量控制发布策略，设置 60%流量访问本地容器集群，40%流量访问公有云容器集群。实现 bookinfo 的灰度发布。

4.在混合云容器集群中设置熔断规则，对 Bookinfo 其中的 productpage 微服务设置熔断规则，并通过负载 fortio 测试客户端触发熔断机制进行验证。