

First World Vocational College Skills Competition

“Cloud Computing” Competition Task Paper

Scenario description: A start-up Internet company planned to build an enterprise private cloud and deploy its business website to the private cloud platform. It planned to hand over the task to engineers A and B. Engineers A and B cooperated with each other to build a private cloud platform, apply for cloud hosts and install and deploy cloud services such as databases, networks and storage. Finally, they deployed and optimized enterprise applications on the private cloud platform.

As the company’s business types increased, the company planned to deploy the K8S platform, migrate business websites to the K8S platform and deploy enterprise applications in full containerization, so as to realize continuous integration and continuous release of enterprise applications.

As the company’s business volume increased, the company decided to use a hybrid container cloud architecture. After completing the cost assessment and project budget, engineers deployed K8S clusters on both local and public clouds to complete the deployment of e-commerce application micro-service clusters to achieve traffic control.

Session I: Private cloud (30 points)

The company plans to use OpenStack to build an enterprise cloud platform for resource pooling and elastic management, centralized management of enterprise applications and unified security authentication and authorization.

The system architecture is shown in Figure 1, and the IP address planning is shown in Table 1.

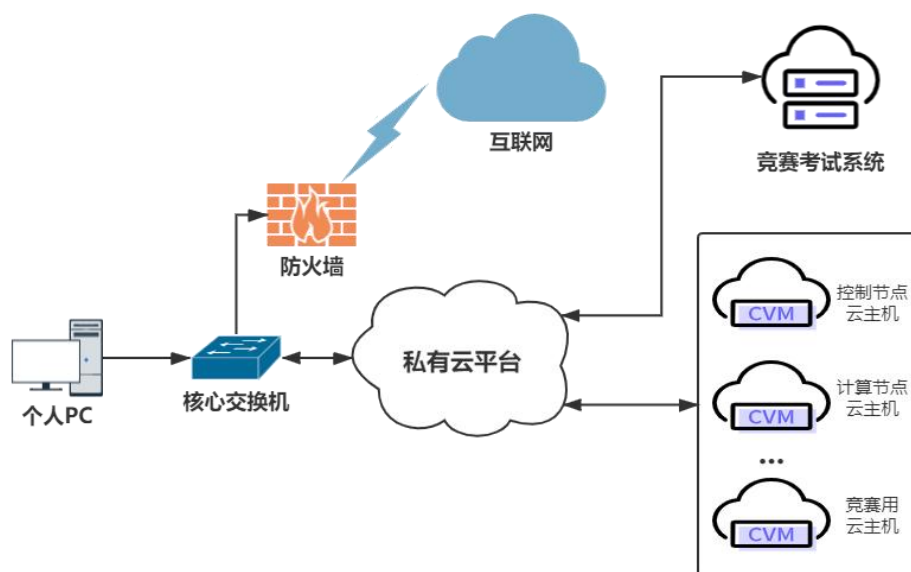


Figure 1 System Architecture Diagram

个人 PC	Personal PC
-------	-------------

核心交换机	Core switch
防火墙	Firewall
私有云平台	Private cloud platform
互联网	Internet
竞赛考试系统	Competition exam system
控制节点主机	Control node cloud host
计算机节点主机	Compute node cloud host
竞赛用云主机	Cloud host for Competition

Table 1 IP Address Planning TableEquipment name	Name of host	Interface	IP address	Notes
Cloud server 1	Controller	eth0	172.129.x.0/24	Vlan x
		eth1	Customized	Build by oneself
Cloud server 2	Computer	eth0	172.129.x.0/24	Vlan x
		eth1	Customized	Build by oneself
Cloud server 3 ... Cloud server n	Customized	eth0	172.129.x.0/24	
PC-1		Local connection	172.24.16.0/24	PC use

Notes:

1. The Competition is conducted in a cluster mode. Each team is provided with independent tenants and users during the Competition. The resource quota of each user is the same. Competitors log in to the private cloud platform for the Competition with their usernames and passwords, and create a cloud host to answer the questions;
2. The X in the table is the workstation number; competitors should create the address of the second network card by themselves according to the question when building the OpenStack;
3. According to the information given in the table, competitors should check the hardware connection and network equipment configuration to ensure that the network connection is normal;
4. Account resources, competition resource packages and accessories required for the Competition will be given in the competition workstation information form and equipment confirmation form;
5. During the Competition, in order to ensure the security of servers, competitors must modify the server password by themselves; when submitting information in the competition system, competitors must confirm their IP address, user name and password.

Task 1. Private cloud service building (10 points)

Question 1. Basic environment building (5 points)

1. Use the provided username and password, log in to the Cloud Computing platform for the Competition, and use the CentOS7.9 image to create two cloud hosts. Use the configuration of 4v_8G_100G_50G for the flavor. Use the provided network for the first network card, and create the Network by yourself for the second network card (the network segment is 10.10.X.0/24, and X is the workstation number). Ensure that the network communicates normally after the cloud host is built, and then configure the server as follows:

(1) Set the name of the control node host as controller, and set the name of the computing node host as compute;

(2) Modify the hosts file to map IP addresses to host names;

2. Set the controller node to access the compute node without a key.

3. Use the provided http service address. Under the http service, there are network yum sources of centos and iaas. Use the http sources as the network sources for installing the iaas platform. Set http.repo, the yum source files of the controller node and compute node, respectively.

4. Deploy the chrony server on the controller node, allow other nodes to synchronize time, start the server and set it to boot; specify the controller node as the upstream NTP server on the compute node, restart the server and set it to start on boot.

5. Divide a blank partition into two 20G partitions on the compute node.

Question 2. OpenStack platform building (5 points)

Complete the installation, building, operation and maintenance tasks of the OpenStack platform (The installation script framework has been provided). Table 2 provides the relevant service parameters that need to be used during the installation process.

Table 2 Password Table

Name of service	Variable	Parameter/password
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 (Extranet card name)

1. Use the provided script framework `iaas-pre-host.sh` and `openrc.sh` environment variable files to fill in the script, install the basic components of the OpenStack platform on the controller and compute nodes, respectively, and complete the relevant configuration.
2. Use the provided script framework `iaas-install-mysql.sh` to fill in the script, install mariadb, memcached, rabbitmq and other services on the controller node and complete the relevant configuration.
3. Use the provided script framework `iaas-install-keystone.sh` to fill in the script, install the keystone service on the controller node and complete the relevant configuration.
4. Use the provided script framework `iaas-install-glance.sh` to fill in the script, install the glance service on the controller node and complete the relevant configuration.
5. Use the provided script frameworks `iaas-install-nova-controller.sh` and `iaas-install-nova-compute.sh` to fill in the script, install the nova service on the controller and compute nodes and complete the configuration.
6. Use the provided script frameworks `iaas-install-neutron-controller.sh` and `iaas-install-neutron-compute.sh` to fill in the script, install the neutron service on the controller and compute nodes and complete the configuration.
7. Use the provided script framework `iaas-install-dashboard.sh` to fill in the script, install the dashboard service on the controller node and complete the relevant configuration.
8. Use the provided script frameworks `iaas-install-cinder-controller.sh` and `iaas-install-cinder-compute.sh` to fill in the script, install the cinder service on the controller and compute nodes and complete the configuration.
9. Use the provided scripting frameworks `iaas-install-swift-controller.sh` and `iaas-install-swift-compute.sh` to fill in the script, install the swift service on the controller and compute nodes and complete the configuration. .
10. Use the provided script framework `iaas-install-heat.sh` to fill in the script, install the heat choreography service and complete the relevant configuration.
11. Use the provided script framework `iaas-install-cloudkitty.sh` to fill in the script, install the cloudkitty billing service and complete the relevant configuration.
12. Use the provided script framework `iaas-install-blazar.sh` to fill in the script, install and add the Blazar-resource booking service and complete the related configuration.

Task 2. Private cloud service operation and maintenance (10 points)

Question 3. OpenStack cloud platform operation and maintenance (4 points)

1. Use the provided cloud security framework components to optimize the security policy of the self-built OpenStack cloud platform from http to https.
2. On the self-built OpenStack platform, tune the OpenStack platform by modifying relevant parameters. The corresponding tuning operations are as follows:

- (1) Set the memory overselling ratio to 1.5 times;
- (2) Set the heartbeat check time of the nova service to 120 seconds.
3. On the self-built OpenStack platform, use the Swift object storage service to modify the corresponding configuration file, and use the object storage Swift as the back-end storage of the glance image service.
4. On the self-built OpenStack platform, write the heat template createvm.yml file. The function of the template is to create a cloud host as required.
5. On the self-built OpenStack platform, expand the cinder storage space to 10G.
6. On the OpenStack private cloud platform, create a cloud host, use the provided software package to write a one-click deployment script, which allows one-click deployment of the gpmall mall application system.
7. Use the manila shared file system service to enable manila to provide a unified management service for shared file systems in a multi-tenant cloud environment.
8. Use the Blazar service to enable administrators to provide resource reservations in OpenStack for different resource types, both virtual (such as instances and volumes) and physical (such as hosts and storage).
9. Use the cloudkitty billing service to process data from different monitoring indicator backends and create billing rules to achieve the purpose of cost accounting.
10. Use the provided iaas-error1 image to create a cloud host. There is an incorrect OpenStack platform in the cloud host after creation. The error is that the cinder service cannot be used normally. Please troubleshoot the error according to the error information so that the cinder service can be used normally.

Question 4. Application deployment (3 points)

1. Use a private cloud platform to apply for a cloud host, install and deploy database services, and set up automatic startup.
2. Use a private cloud platform to apply for a cloud host, install and deploy the Redis service, and set up automatic startup.
3. Use a private cloud platform to apply for a cloud host, and install and deploy Nginx services.
4. Use a private cloud platform to apply for cloud hosts, install and deploy zookeeper and Kafka services.
5. Use a private cloud platform to apply for a cloud host, and deploy enterprise applications with the above installed services.

Question 5. Application architecture upgrade (3 points)

1. Upgrade the database of the enterprise application to a database high-availability cluster, namely, a mariadb_galera_cluster cluster.
2. Add a load balancing HA service to the database cluster. The weights for accessing the three database nodes should be 3, 3, and 4.
3. Upgrade the Redis service of enterprise applications to the Redis master and slave

services.

4. Upgrade the deployment method to Ansible automated deployment and write Ansible playbooks, so when the Ansible playbooks are implemented, the enterprise cluster applications can be deployed with one click.

Task 3. Private cloud operation, maintenance and development (10 points)

Question 6. OpenStack automated operation and maintenance

1. Use Python language, encapsulate the Python class of flavor_manager based on OpenStack APIs to realize the addition, deletion, inspection and modification of OpenStack cloud host type management, return the operation result in json format, and submit the realized code file after completion.

2. Use Python language, encapsulate the Python class of server_manager based on OpenStack APIs to realize the addition, deletion, inspection, and modification of OpenStack host management, return the operation results in json format, and submit the realized code file after completion.

3. Use Python language, write an image management program based on OpenStack APIs to realize all image queries and modifications, and submit the realized code file after completion.

4. Use Python language, connect with the programs of cloud host type management, cloud host management, and image management based on the Python click framework to realize customized command line management tools, keep the commands consistent with the OpenStack commands, and complete the unit test of python code writing, and submit the realized code file after completion.

Session II: Container cloud (40 points)

Task 1. Container cloud service building

Question 1. Container cloud platform building (5 points)

1. Complete the installation of the Kubernetes cluster on the master and node nodes.
2. Complete the installation of the KubeVirt environment in the Kubernetes cluster.
3. Complete the installation of the service grid Istio environment in the Kubernetes cluster.
4. After the cluster is deployed, use kubectl's own sorting function to list all the Pods in the cluster and sort them by the name field.

Task 2. Container cloud application deployment (15 points)

Question 2. Container-based web application system deployment task (7 points)

The company has developed a set of distributed system kits based on Spring Cloud and implemented full containerized deployment. The main module of the system is shown in Table 3:

Table 3 System Module Table

Module	Notes
--------	-------

Config	Configuration center
Gateway	Microservice gateway
Eureka	Registration center
Spring Boot Admin	Source software for monitoring and management
Zipkin	Distributed real-time data tracking system
Hello Service	Service registration

Containerize the above components as required.

1. Write a Dockerfile and create a Config image based on the centos:7.9.2009 base image, generate an image named chinaskill-config:v1.0, and push it to the private storehouse.
2. Write a Dockerfile, and create a Gateway image based on the centos:7.9.2009 base image, generate an image named chinaskill-gateway:v1.0, and push it to the private storehouse.
3. Write a Dockerfile, create an Eureka image based on the centos:7.9.2009 base image, generate an image named chinaskill-eureka:v1.0, and push it to the private storehouse.
4. Write a Dockerfile, create a Spring Boot Admin image based on the centos:7.9.2009 base image, generate an image named chinaskill-admin:v1.0, and push it to the private storehouse.
5. Write a Dockerfile to create a Zipkin image based on the centos:7.9.2009 base image, generate an image named chinaskill-zipkin:v1.0, and push it to the private storehouse.
6. Write a Dockerfile, create a Hello Service image based on the centos:7.9.2009 base image, generate an image named chinaskill-hello:v1.0, and push it to the private storehouse.
7. Write the docker-compose.yaml file and use the above image choreography to deploy the Spring Cloud distributed system.

Question 3. Kubernetes-based building of CICD (8 points)

The company decided to use the architecture of GitLab + Jenkins + Harbor + Kubernetes to build a CICD environment, so as to shorten the development and launch cycle of new functions, meet customer needs in a timely manner, and realize part of the DevOps process to reduce the burden of deployment, operation and maintenance, and realize visual container life cycle management, application release and version iterative update, and complete CICD environment deployment. The CICD application system architecture is shown in Figure 2:

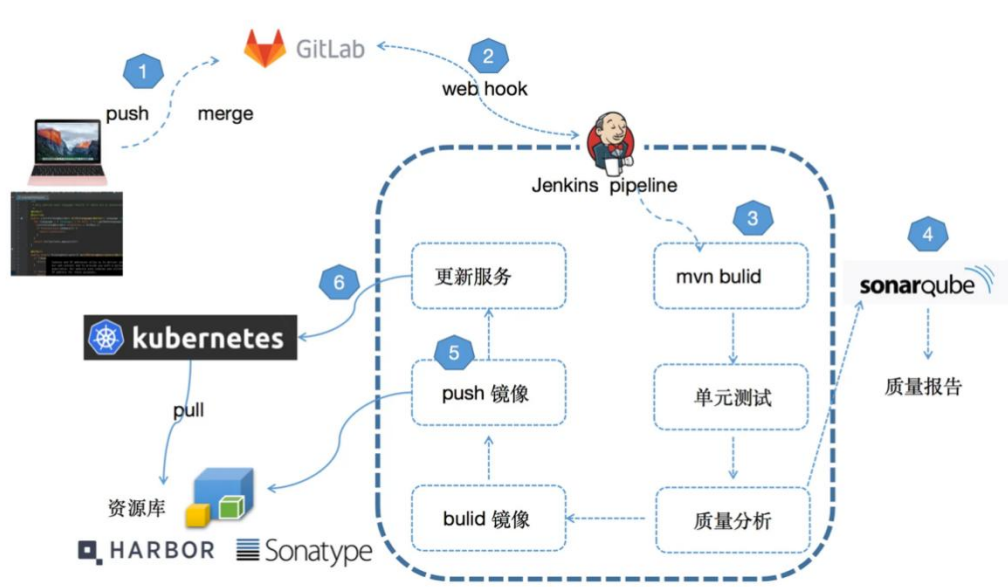


Figure 2 CICD Application System Architecture Diagram

资源库	Resource library
更新服务	Update service
Push 镜像	Push image
Build 镜像	Build image
单元测试	Unit testing
质量分析	Quality analysis
质量报告	Quality report

1. Complete the coding of the Piggy Metrics project.
2. Complete the deployment of the GitLab program in the devops namespace of Kubernetes cluster, and use the NodePort method to expose port 80 of the GitLab service as 30081.
3. Complete the deployment of the Jenkins service in the devops namespace of Kubernetes cluster, and use the NodePort method to expose the port 80 of the Jenkins service as 30880.
4. Complete the configuration of GitLab tool and Maven tool in Jenkins.
5. Complete the configuration of webhooks in GitLab.
6. Create a new pipeline task springcloud in Jenkins, and write a Jenkinsfile. The stages of the entire build process should include test, build, release, review, and deploy.
7. Push the compiled Piggy Metrics project code to the GitLab storehouse and trigger automatic builds, so as to complete the automatic implementation of code compilation, image building, image push and service release.

Task 3. Container cloud service operation and maintenance (10 points)

Question 4. Container-based operation and maintenance of Kubernetes (5 points)

1. Set the master node as a taint and reschedule all allowed Pods on that node.
2. Create a Pod under the default namespace of the Kubernetes cluster. The specific requirements are as follows:

- (1) Pod name: nginx;
- (2) Specify the image nginx: 1.17;
- (3) Container restart strategy: always;

Complete the building of the Pod.

3. Label the master nodes with “disktype=ssd” and “exam=chinaskill”, and then create a Deployment in the default namespace of the Kubernetes cluster. The specific requirements are as follows:

- (1) Deployment name: nginx-deployment;
- (2) Specify image nginx: latest;
- (3) Pods can only be scheduled on nodes with the label “disktype=ssd”;
- (4) Nodes with the label “exam=chinaskill” should be scheduled first.

Complete the building of the Deployment.

4. Create a cluster role in the Kubernetes cluster. The specific requirements are as follows:

- (1) Cluster role name: deployment-clusterrole;
- (2) This role has the right to build Deployment, Daemonset, and StatefulSet.

Complete the creation of the cluster role.

5. Create a network policy in the Kubernetes cluster, and the specific requirements are as follows:

- (1) Network policy name: exam-network;
- (2) For Pods under the namespace test, only allow the access to the Pods under the same namespace , and the port 9000 of the Pods.

Complete the creation of the network policy.

6. The default validity period of the Kubernetes certificate is only one year. Please extend the validity period of the Kubernetes certificate. The specific requirements are as follows:

- (1) Extend the validity period of the Kubernetes cluster certificate to 100 years;
- (2) Recompile the Kubernetes source code;
- (3) Complete the change of kubeadm.

Complete the change of the Kubernetes certificate.

7. Complete the microservice-istio microservice application deployment in the

Kubernetes cluster and istio grid service. The specific requirements are as follows:

- (1) Compile the microservice-istio application code, improve the docker-build.sh script, build the microservice image and upload it.
- (2) Write the microservice-istio deployment yaml to complete the microservice-istio service grid deployment.

Please complete the creation of the request route.

8. Create a service circuit breaker policy for the microservice-istio application. The specific requirements are as follows:

- (1) Write circuit-breaker.yaml rules for the DestinationRule fuse, specify maxConnections as 1 and http1MaxPendingRequests as 1 in the settings
- (2) Initiate a network request through fortio, request 3 concurrent connections (-c 3) and 30 requests (-n 30), and output the fuse test information.

Question 5. Virtual machine-based operation and maintenance of Kubernetes (5 points)

According to Gartner's latest prediction, by 2022, 75% of production applications will be running in container environments. For technical reasons, most of a company's architecture still runs on virtual machines. After technical research, KubeVirt supports complete virtual machine management, and can allow the perfect migration of some applications that cannot be migrated to containers due to outdated kernel versions or language problems. Therefore, the company decided to migrate all business to Kubernetes clusters.

1. Enable live migration in KubeVirt to migrate a running virtual machine instance from one node to another, while the workload continues to run and maintain continuous access.
2. Use the provided OpenStack qcow2 image to create a VMI under the default namespace of the Kubernetes cluster. The specific requirements are as follows:

- (1) VMI name: chinaskill-vmi;
- (2) Operation strategy: Manual;
- (3) Disk drive: virtio.

Complete the creation of the VMI.

3. Write the YAML file vmi-sshkey.yaml on the master node to perform dynamic key injection for the virtual instance chinaskill-vmi. The specific requirements are as follows:

- (1) Use qemuGuestAgent to attach the access credential api to chinaskill-vmi.

Use SSH to verify VMI connectivity after running the YAML file.

Task 4. Container cloud operation, maintenance development (10 points)

Question 6. Kubernetes operation, maintenance and development (10 points)

1. Apply for a cloud host, and complete the installation of the Python 3.6.8 environment and Kubernetes library on the cloud host.

2. Write Python code to list all Pods, Deployments and Services in the kube-system namespace of the Kubernetes cluster.
3. Write Python code and use the nginx:1.17 image to create a Deployment named test in the default namespace of the Kubernetes cluster, and the number of replicas is 3.
4. Write Python code, modify the image file in the Deployment in the previous question to nginx:1.18, and update the Deployment in a rolling manner.
5. Use Python to write a Kubernetes scheduler, create a client that API calls, monitor Pod changes through stream, and when a Pod with Pending state is detected, schedule it through the written scheduler. The scheduler method is to randomly select an available one node, use the API and bind the Pod and the host through the create_namespaced_binding method.

Session III: Hybrid cloud (30 points)

Task 1. Public cloud cluster building (15 points)

Question 1. Public cloud cost budget (5 points)

According to the requirements for application system planning, use cost estimation tools for public cloud to complete public cloud resource selection, cost statistics and cost estimation, and calculate the best cloud efficiency.

Question 2. Public cloud network building (5 points)

Complete the building of the virtual private cloud in the public cloud, and create an internal network named intnetX: the IP address is 172.16.0.0/16; the subnet name is intnetX-sql: the IP address is 172.16.1.0/24;

Purchase elastic load balancing elb and apply for public IP for access.

Question 3. Public cloud container cluster building (5 points)

Manage the public cloud container engine cluster, and complete the following operations of building, operation and maintenance.

1. Use the intnetX internal network to create a container cluster kcloud in the public cloud.
2. Install the kubectl command in the kcloud cluster and use the kubectl command to manage the kcloud cluster.
3. Deploy the helm tool in the kcloud cluster, use the provided wordpress charts package, and write a yaml file to create the PV volume required by wordpress. Deploy the wordpress service through the helm tool.
4. Deploy the application istio service grid in the kcloud cluster, use the provided V1 and V2 container images to configure grayscale release, and set the V1 version access traffic to 40% and V2 version access traffic to 60%.

Task 2. Hybrid cloud cluster configuration (5 points)

Question 4. Interfacing between local container cloud and public cloud

1. Configure the Kubernetes cluster in the OpenStack platform, use kubekey to install Kubesphere and other related components, so as to realize the connection between the

local Kubernetes cluster and the public cloud cluster..

Task 3. Hybrid cloud cluster application deployment (10 points)

Question 5. Hybrid cloud service cases (10 points)

Bookinfo is an online bookstore microservice application. The Bookinfo application is divided into four separate microservices: (1) Productpage. This microservice will call two microservices of details and reviews to generate pages. (2) Details. This microservice contains book information. (3) Reviews. This microservice contains book-related reviews. It also calls the ratings microservice. (4) Ratings. This microservice contains rating information consisting of book reviews.

There are three versions of the reviews microservice:

The Version 1 does not call the ratings service.

The Version 2 calls the ratings service and show rating informations with 1 to 5 black star icons.

The Version 3 calls the ratings service and shows rating information with 1 to 5 red star icons.

Implement the deployment of the application service grid, and use the Istio control plane to provide the application with functions such as service routing, monitoring data collection, and policy enforcement. The deployment architecture is shown in Figure 3:

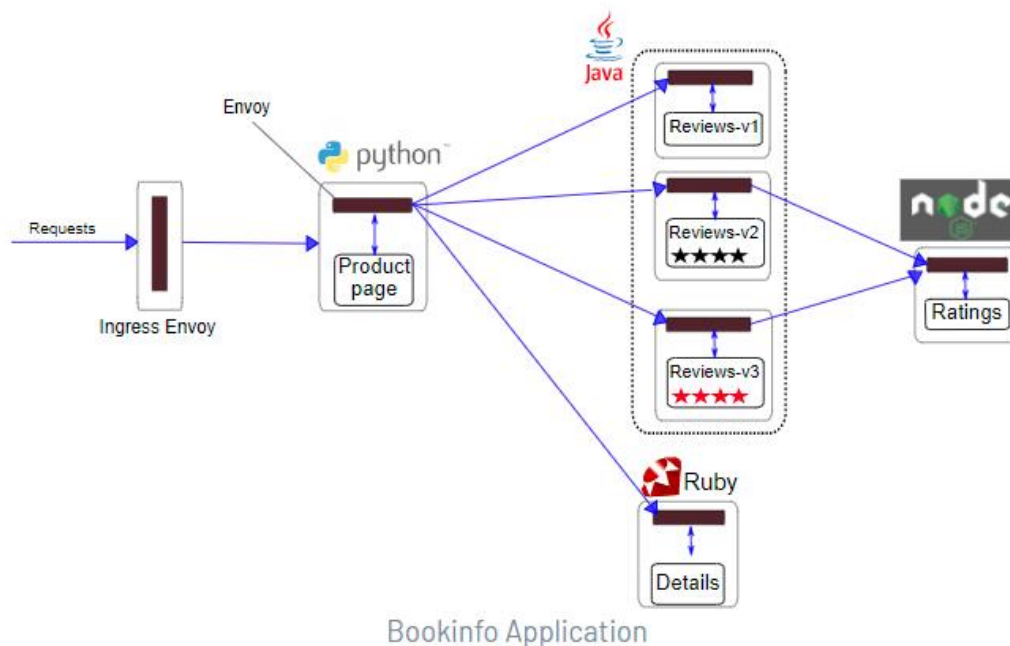


Figure 3 Deployment Architecture Diagram

1. Write a Dockerfile to build the productpage:v1.0 image. The specific requirements are as follows:

- (1) Base image: centos:7.9.2009
- (2) Install the Python3.6.8 tool;

- (3) Install the productpage service and set up automatic startup.
2. Design the application microservice deployment architecture, and complete the deployment of the bookinfo microservice application by modifying the yaml file of the microservice application in the hybrid cloud container cluster.
3. Configure the traffic control release policy in the hybrid cloud container cluster, and set 60% of the traffic to access the local container cluster and 40% of the traffic to access the public cloud container cluster. Implement the grayscale release of bookinfo.
4. Set circuit breaker rules in the hybrid cloud container cluster, set circuit breaker rules for the productpage microservice in Bookinfo, and verify that the circuit breaker mechanism is triggered by the load fortio test client.