

Kubernetes 编排工具问题解答

In-Class Assignment 2

September 30, 2025

1 问题 1：编排工具的关键作用

问题：Orchestration tools, such as Kubernetes, play a key role in the server infrastructure for the modern applications.

- (a) Explain how these tools help manage and scale application servers.
- (b) Describe how orchestration tools facilitate automated deployment, scaling, and management of application servers.

答案：

1.1 (a) 编排工具如何帮助管理和扩展应用服务器

- **资源管理：**自动分配和管理计算资源（CPU、内存、存储），确保应用获得所需资源
- **负载均衡：**在多个服务器实例间分配流量，提高性能和可用性
- **健康监控：**持续监控应用状态，自动重启失败的实例
- **水平扩展：**根据负载自动增加或减少服务器实例数量
- **服务发现：**自动管理服务间的网络通信和依赖关系

1.2 (b) 编排工具如何促进自动化部署、扩展和管理

- **声明式配置：**通过 YAML 文件定义期望状态，系统自动维护该状态
- **滚动更新：**无停机时间地更新应用版本
- **自动扩缩容：**基于 CPU 使用率、内存或自定义指标自动调整实例数量
- **故障恢复：**自动检测并替换失败的节点或容器
- **配置管理：**集中管理应用配置和密钥

2 问题 2: Pod、Deployment 和 Service 的区别

问题: Explain the difference between a Pod, Deployment, and Service.

答案:

- Pod:
 - Kubernetes 中最小的可部署单元
 - 包含一个或多个紧密耦合的容器
 - 共享网络和存储资源
 - 生命周期短暂, 可能随时被删除和重建
- Deployment:
 - 管理 Pod 副本的高级控制器
 - 确保指定数量的 Pod 副本始终运行
 - 支持滚动更新和回滚
 - 提供声明式的应用部署和管理
- Service:
 - 为 Pod 提供稳定的网络访问入口
 - 通过标签选择器将流量路由到后端 Pod
 - 提供负载均衡功能
 - 即使 Pod 重建, Service 的 IP 和 DNS 名称保持不变

3 问题 3: Kubernetes 中的 Namespace

问题: What is a Namespace in Kubernetes? Please list one example.

答案:

Namespace 是 Kubernetes 中的虚拟集群概念, 用于在同一物理集群中创建多个逻辑隔离的环境。它提供:

- 资源隔离和组织
- 访问控制和权限管理
- 资源配额限制
- 命名空间内的资源名称唯一性

示例:

```
apiVersion: v1
kind: Namespace
metadata:
  name: development
```

常见的默认 namespace 包括:

- default: 默认命名空间
- kube-system: 系统组件命名空间
- kube-public: 公共资源命名空间

4 问题 4: Kubelet 的作用和检查节点的命令

问题: Explain the role of the Kubelet. How do you check the nodes in a Kubernetes cluster? (kubectl command expected)

答案:

4.1 Kubelet 的作用

- 运行在每个节点上的主要代理程序
- 负责管理节点上的 Pod 生命周期
- 与 API Server 通信, 接收 Pod 规范
- 监控容器健康状态并报告给控制平面
- 管理容器的启动、停止和重启
- 执行健康检查和资源监控

4.2 检查集群节点的 kubectl 命令

```
# 查看所有节点
kubectl get nodes

# 查看节点详细信息
kubectl get nodes -o wide

# 查看特定节点的详细描述
kubectl describe node <node-name>

# 查看节点状态
kubectl get nodes --show-labels
```

5 问题 5: ClusterIP、NodePort 和 LoadBalancer 服务的区别

问题: What is the difference between ClusterIP, NodePort, and LoadBalancer services?

答案:

- ClusterIP:
 - 默认服务类型

- 只能在集群内部访问
- 分配一个集群内部 IP 地址
- 适用于内部服务通信
- NodePort:
 - 在每个节点上开放一个特定端口 (30000-32767)
 - 可以通过任何节点的 IP:NodePort 访问服务
 - 自动创建 ClusterIP
 - 适用于开发和测试环境
- LoadBalancer:
 - 创建外部负载均衡器 (需要云提供商支持)
 - 自动创建 NodePort 和 ClusterIP
 - 提供外部可访问的 IP 地址
 - 适用于生产环境的外部访问

6 问题 6: 使用 kubectl 扩展 Deployment 到 5 个副本

问题: How do you scale a Deployment to 5 replicas using kubectl?

答案:

```
# 方法1: 使用scale命令
kubectl scale deployment <deployment-name> --replicas=5

# 方法2: 使用patch命令
kubectl patch deployment <deployment-name> -p '{"spec":{"replicas":5}}'

# 验证扩展结果
kubectl get deployment <deployment-name>
```

7 问题 7: 无停机更新 Deployment 镜像

问题: How would you update the image of a Deployment without downtime?

答案:

```
# 方法1: 使用set image命令
kubectl set image deployment/<deployment-name> <container-name>=<new-image>

# 方法2: 使用patch命令
kubectl patch deployment <deployment-name> -p '{"spec":{"template":{"spec":{"containers":[{"name":"<container-name>","image":"<new-image>"}]}}}}'
```

```
# 查看滚动更新状态
kubectl rollout status deployment/<deployment-name>

# 如需回滚
kubectl rollout undo deployment/<deployment-name>
```

8 问题 8：将 Deployment 暴露给外部流量

问题： How do you expose a Deployment to external traffic?

答案：

```
# 方法1：创建NodePort服务
kubectl expose deployment <deployment-name> --type=NodePort --port=80

# 方法2：创建LoadBalancer服务
kubectl expose deployment <deployment-name> --type=LoadBalancer --port=80

# 方法3：使用kubectl create service
kubectl create service nodeport <service-name> --tcp=80:8080

# 方法4：通过Ingress（需要先创建Service）
kubectl expose deployment <deployment-name> --port=80
# 然后创建Ingress资源
```

9 问题 9：Kubernetes 调度决策机制

问题： How does Kubernetes scheduling decide which node a Pod runs on?

答案：

Kubernetes 调度器通过以下步骤决定 Pod 运行的节点：

9.1 调度过程

1. **过滤阶段：** 排除不符合条件的节点
 - 资源需求（CPU、内存）
 - 节点选择器和亲和性规则
 - 污点和容忍度
 - 端口冲突检查
2. **评分阶段：** 对符合条件的节点打分
 - 资源利用率均衡
 - 亲和性偏好
 - 镜像本地性
 - 节点负载均衡
3. **选择阶段：** 选择得分最高的节点

9.2 影响因素

- 节点资源可用性
- Pod 的资源请求和限制
- 节点选择器 (nodeSelector)
- 亲和性和反亲和性规则
- 污点 (Taints) 和容忍度 (Tolerations)

10 问题 10: Ingress 的作用及其与 Service 的区别

问题: What is the role of Ingress and how does it differ from a Service?

答案:

10.1 Ingress 的作用

- 管理集群外部对服务的 HTTP/HTTPS 访问
- 提供负载均衡、SSL 终止和基于名称的虚拟主机
- 支持路径路由和域名路由
- 集中管理外部访问规则

10.2 Ingress 与 Service 的区别

特性	Service	Ingress
层级	L4 (传输层)	L7 (应用层)
协议	TCP/UDP	HTTP/HTTPS
路由	基于端口	基于路径/域名
SSL	不支持	支持 SSL 终止
负载均衡	简单轮询	高级负载均衡
成本	每个服务需要 LoadBalancer	单个入口点

10.3 示例配置

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example-ingress
spec:
  rules:
  - host: example.com
    http:
      paths:
      - path: /app1
```

```
pathType: Prefix
backend:
  service:
    name: app1-service
    port:
      number: 80
```