kubernetes实训二：Pod配置文件

**一、实验目的**

学习了解kubernetes配置文件的创建以及使用

**二、实验描述**

1. kubernetes配置文件的创建以及使用
2. 主要步骤：
   * 创建namespace
   * 查看namespace
   * 创建Pod配置文件
   * 使用配置文件创建Pod

**三、实验环境**

* 系统版本：centos 7.6
* kubernetes版本： 1.25

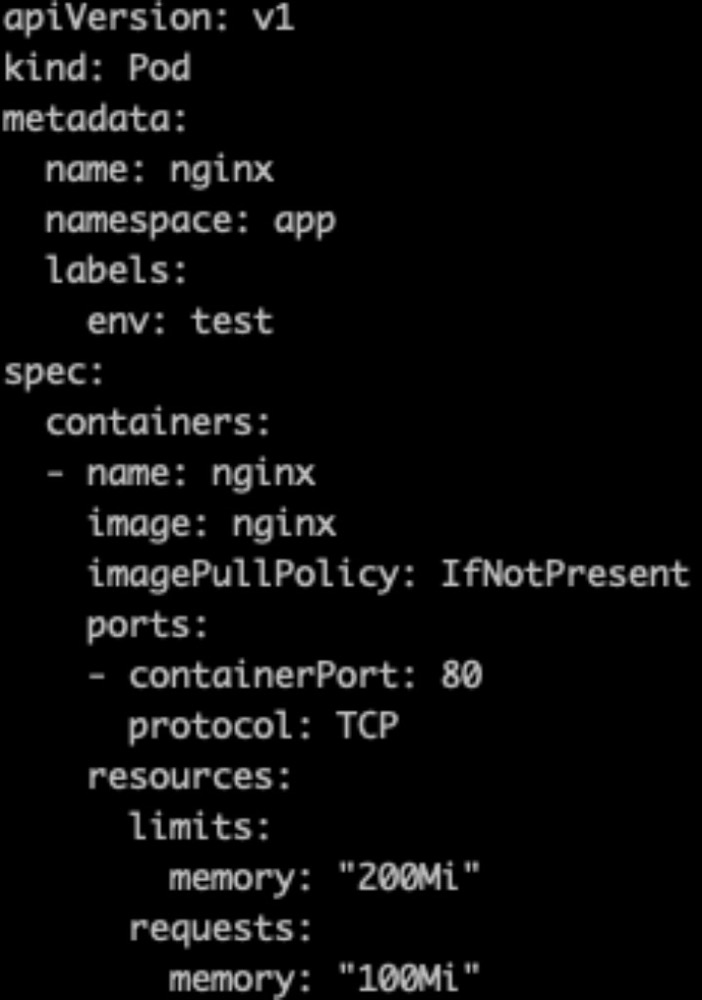
**四、实验知识点**

* linux系统基础配置
* 配置文件基础命令

**五、实验步骤**

Kubernetes 命名空间（Namespace）是一种用于隔离和组织集群资源的机制。每个命名空间都具有自己的资源范围，可以将相关资源组织在一起，以便更好地管理和隔离应用程序。以下是一个实操示例，演示如何创建一个命名空间配置文件并将其应用到 Kubernetes 集群中。

## 基本知识：配置文件详解



apiVersion: 指明使用的 api 的版本

Kind: 指明资源对象类型

metadata:

name: 指明 元数据中的名称

namespace: 所属的名称空间

labels:

env: test 给pod打一个标签

spec:

containers: # 使用一个列表，配置 容器的属性

name: 容器在k8s中显示的名字

image 在容器仓库中，真实存在的、具体的镜像的名称

imagePullPolicy: 镜像拉取策略

ports：

containerPort: 开放的容器端口

protocol: 使用的协议

resources: # 资源配置

limits: # 限制内存最多使用 200M

memory: "200Mi"

requests: # 申请100M的内存

memory: "100Mi"

## 步骤 1：创建命名空间配置文件

创建一个名为 `my-namespace.yaml` 的命名空间配置文件，该文件定义了要创建的命名空间。

```yaml

apiVersion: v1

kind: Namespace

metadata:

name: my-namespace

```



此配置文件定义了一个名为 "my-namespace" 的命名空间。

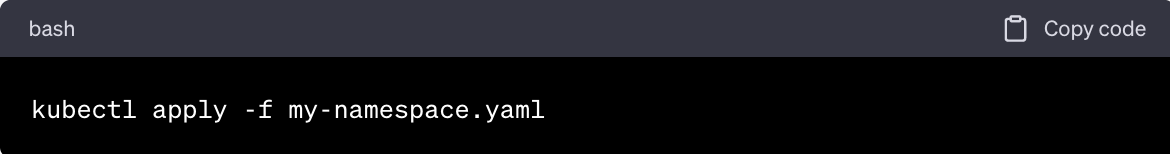
## 步骤 2：应用命名空间配置文件

使用以下命令将命名空间配置文件应用到 Kubernetes 集群中：

```bash

kubectl apply -f my-namespace.yaml

```



这会将配置文件中定义的命名空间创建并部署到集群中。如果成功，将收到一个表示命名空间已创建的消息。

## 步骤 3：验证命名空间

要验证命名空间是否已成功创建，使用以下命令：

```bash

kubectl get namespaces

```



将看到一个输出，其中包含了集群中所有命名空间的列表，包括刚刚创建的 "my-namespace"。

## 步骤 4：创建 Pod 配置文件

创建一个名为 `my-pod.yaml` 的 Pod 配置文件，该文件将定义要创建的 Pod。

```yaml

apiVersion: v1

kind: Pod

metadata:

name: my-pod

namespace: my-namespace # 指定命名空间

spec:

containers:

- name: nginx-container

image: nginx:latest

ports:

- containerPort: 80 # 容器内部监听的端口

```



此配置文件定义了一个名为 "my-pod" 的 Pod，其中包含一个名为 "nginx-container" 的容器，该容器使用了最新版本的 Nginx 镜像。我们在 `metadata` 部分明确指定了命名空间为 `my-namespace`。这将确保创建的 Pod 属于 `my-namespace` 命名空间。

## 步骤 5：应用 Pod 配置文件

使用以下命令将 Pod 配置文件应用到 Kubernetes 集群中：

```bash

kubectl apply -f my-pod.yaml

```



这会将配置文件中定义的 Pod 创建并部署到集群中。如果成功，将收到一个表示 Pod 已创建的消息。

## 步骤 6：检查 Pod 状态

可以使用以下命令来检查 Pod 的状态：

```bash

kubectl get pods

```



看到一个输出，其中包含了集群中所有 Pod 的列表，包括刚刚创建的"my-pod"

## 步骤 7：访问 Pod

要访问 Pod，可以使用 `kubectl port-forward` 命令将本地端口与 Pod 端口绑定，例如：

```bash

kubectl port-forward my-pod 8080:80

```



这将把 Pod 的端口 80 映射到本地端口 8080。现在，可以在浏览器中访问 `http://服务器ip:8080` 来查看 Nginx 默认欢迎页面。（红色：需要保证每个学生使用不同端口）

## 步骤 8：清理资源

要删除 Pod，可以使用以下命令：

```bash

kubectl delete pod my-pod

```



这会删除刚刚创建的 Pod。

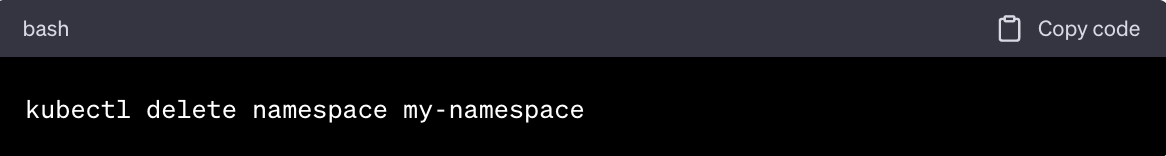
## 步骤 9：清理资源

要删除命名空间及其内的资源，可以使用以下命令：

```bash

kubectl delete namespace my-namespace

```



这将删除刚刚创建的命名空间以及其中的所有资源（如 Pod、Service 等）。

这是一个基本的 Kubernetes Pod 配置文件的实际操作示例。通过创建、部署和管理 Pod，可以开始构建更复杂的应用程序并学习如何使用 Kubernetes 管理容器化应用程序。根据需求，后续配置文件中可以包括更多的选项，以定义容器的资源限制、环境变量、卷挂载等。

**六、总结**

本章介绍了配置文件的一些常见操作。