|  |  |
| --- | --- |
| **EXP NO: 12** | **12. DEVELOPMENT OF SIMPLE EXPERIMENTS USING MINIKUBE** |
| **DATE:** |

# AIM:

To set up and manage a Kubernetes cluster using Minikube and deploy an application.

**INTRODUCTION:**

Minikube is a lightweight Kubernetes implementation that runs a single-node cluster on a local machine. It is primarily used for development, testing, and learning Kubernetes concepts without requiring a full multi-node setup. Minikube enables developers to create, manage, and test Kubernetes deployments efficiently.

**PURPOSE:**

The objective of this experiment is to:

* Install and verify Minikube and Kubernetes versions.
* Start a Minikube cluster using Docker as the driver.
* Deploy an Nginx application within the Kubernetes cluster.
* Manage deployments and services using kubectl commands.
* Expose the Nginx deployment as a service and access it via a browser.

**WHY MINIKUBE?**

This experiment provides hands-on experience with Kubernetes cluster management and application deployment. It covers key concepts such as:

* **Minikube Cluster Initialization**: Setting up a Kubernetes environment locally.
* **Deployments and Pods**: Managing containerized applications within Kubernetes.
* **Service Exposure**: Making applications accessible externally.
* **Cluster Monitoring**: Checking cluster status, services, and deployments.
* **Real-World Application**: Learning the fundamentals of Kubernetes, which is essential for cloud-native application development.

# PROCEDURE:

**Step 1:** Start.

**Step 2:** Check the installed Kubernetes client version. Command:

# kubectl version --client

**Step 3:** Check the installed Minikube version. Command:

# minikube version

**Step 4:** Start Minikube using Docker as the driver. Command:

# minikube start --driver=docker

**Step 5:** Check the status of Minikube to ensure it is running correctly. Command:

# minikube status

**Step 6:** Display cluster information. Command:

# kubectl cluster-info

**Step 7:** Create a deployment using the Nginx image. Command:

# kubectl create deployment nginx-deployment --image=nginx

**Step 8:** List all available deployments to confirm the creation of nginx-deployment. Command:

# kubectl get deployments

**Step 9:** List all running pods in the cluster. Command:

# kubectl get pods

**Step 10:** Expose the deployment as a service, making it accessible on port 80 using a NodePort service type.

Command:

# kubectl expose deployment nginx-deployment --port=80 --type=NodePort

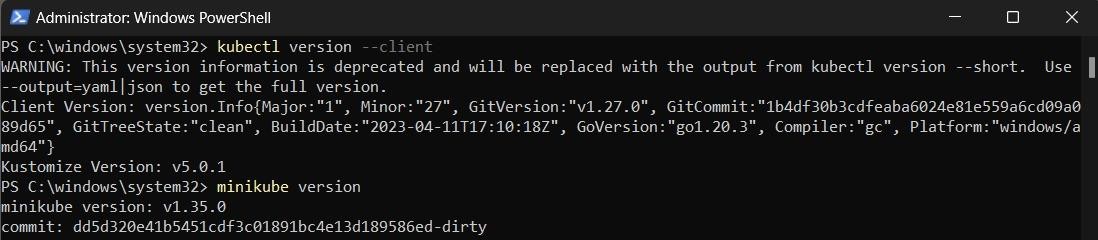
**Step 11:** List all running services to verify that nginx-deployment is exposed. Command:

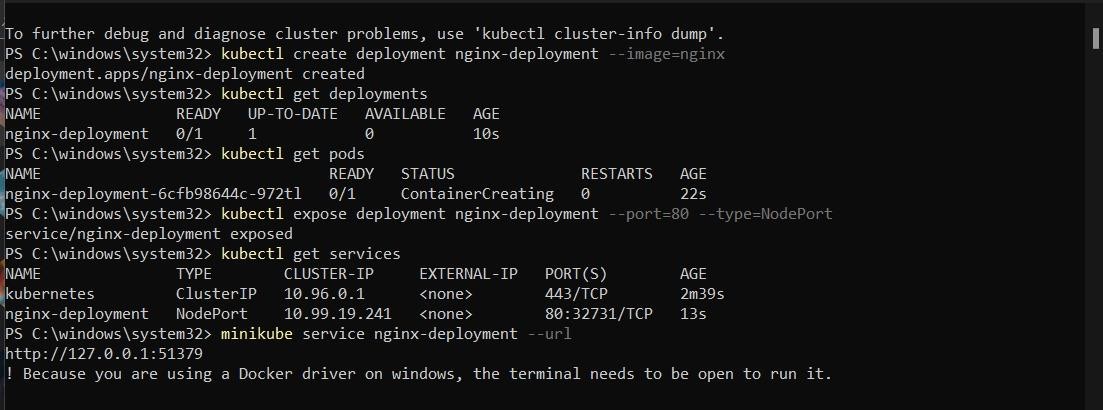
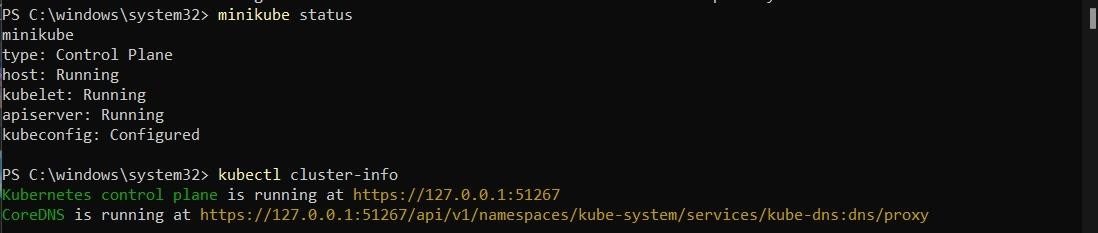
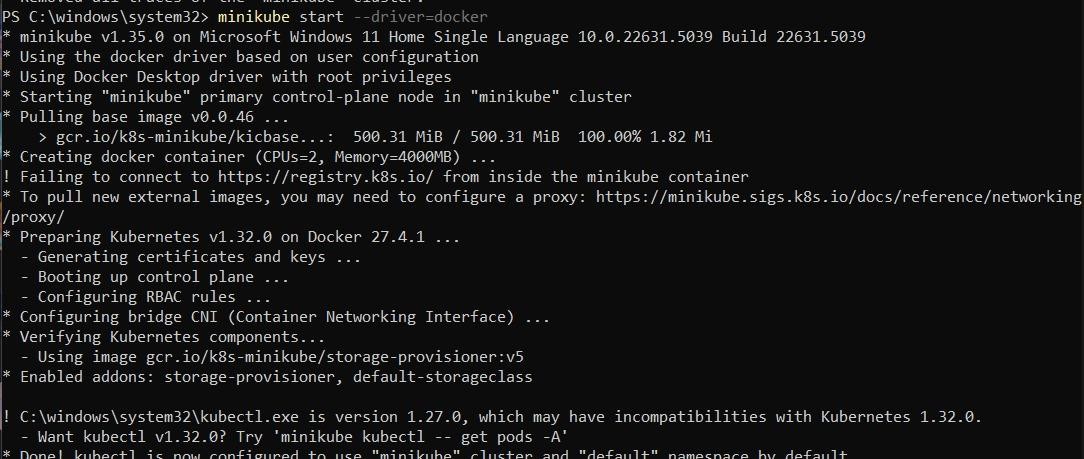
# kubectl get services

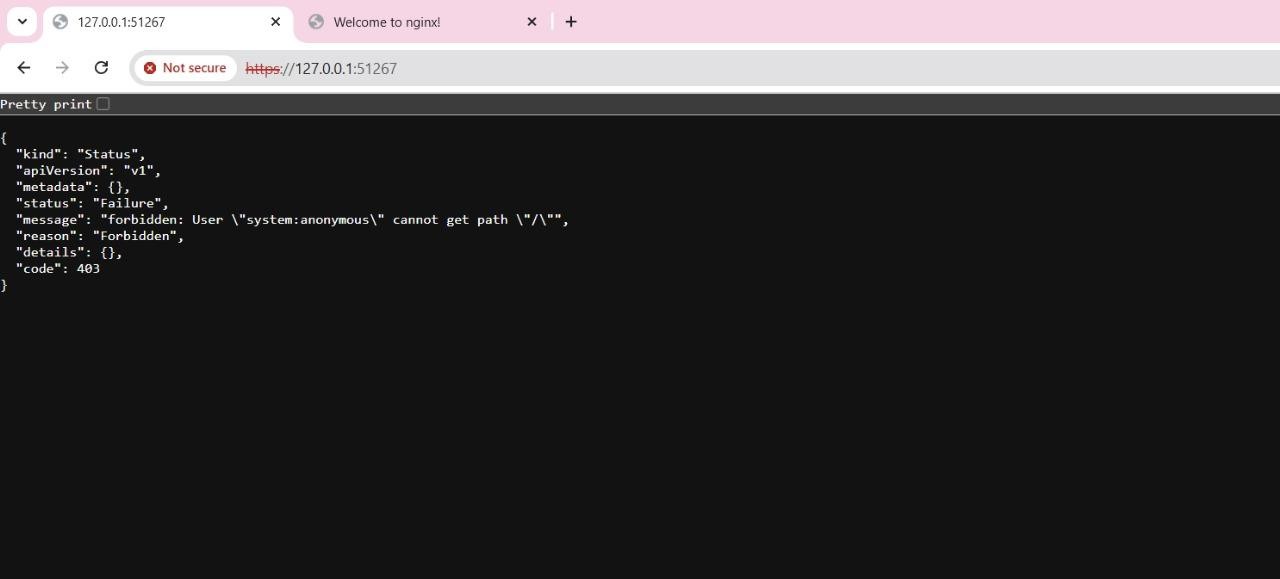
**Step 12:** Retrieve the external URL of the service to access it in a browser. Command:

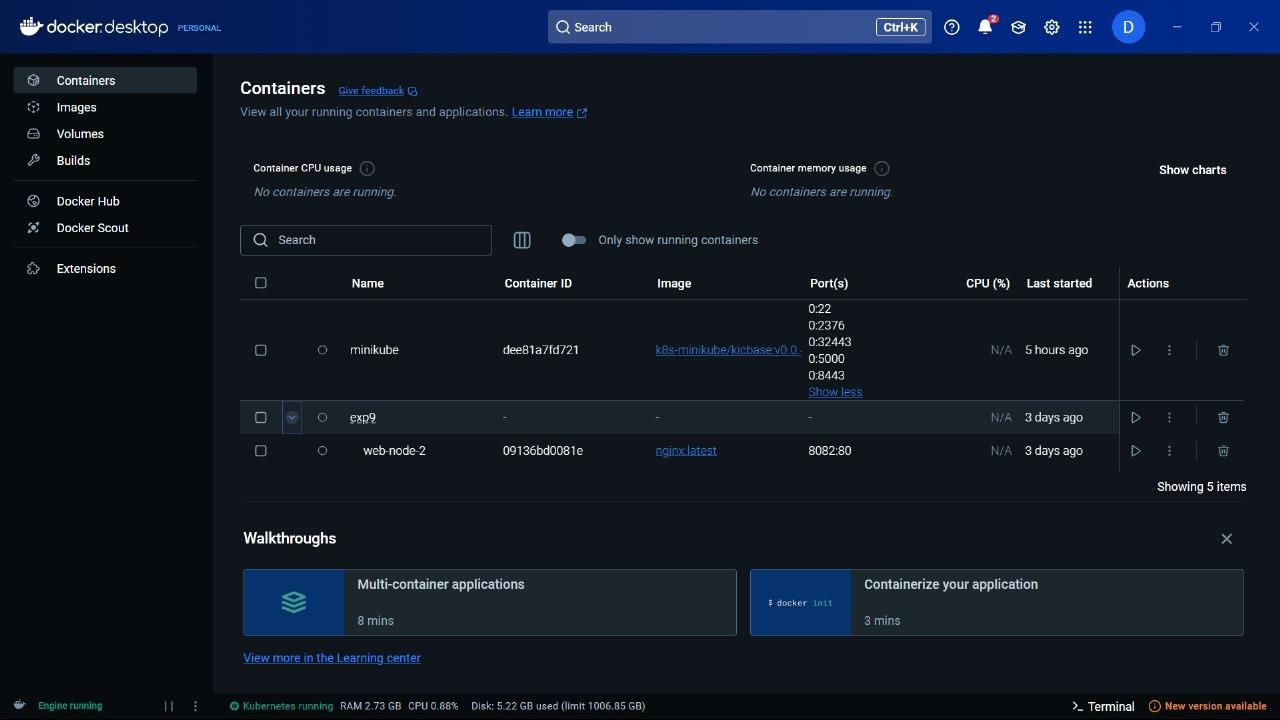
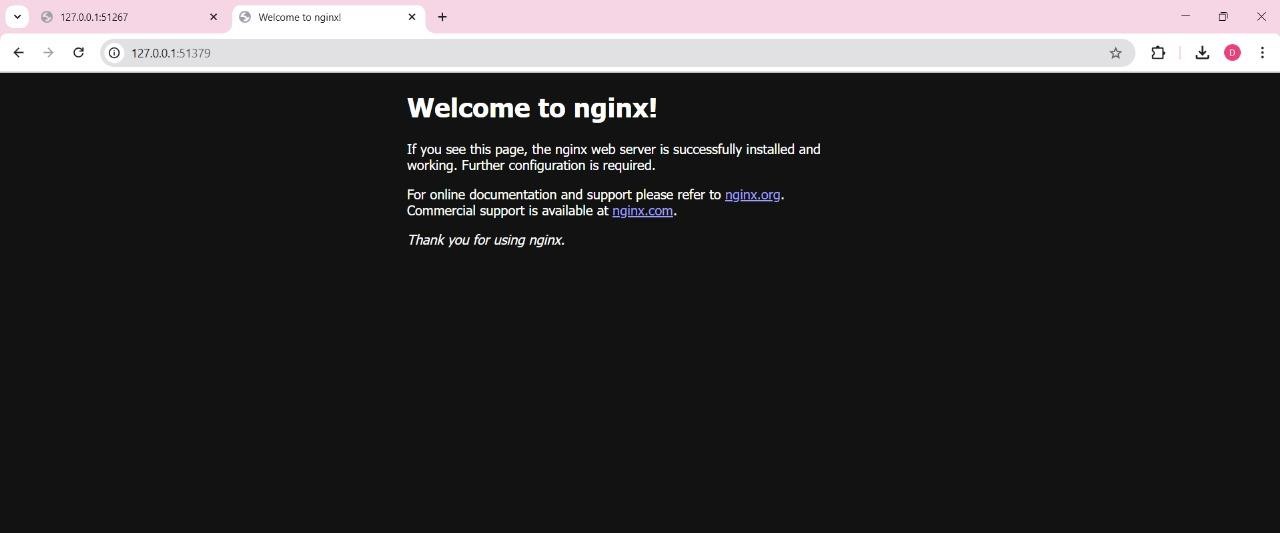
# minikube service nginx-deployment --url

# OUTPUT:



****



****

**RESULT:**

Thus, a Kubernetes cluster was successfully set up using Minikube, and the Nginx deployment was created and exposed as a service.