**Lesson 02 Demo 05**

**Launching the Kubernetes Dashboard**

**Objective:** To deploy the Kubernetes dashboard to facilitate the management and troubleshooting of cluster resources and applications

**Tools required:** kubeadm, kubectl, kubelet, and containerd

**Prerequisites:** A Kubernetes cluster (refer to Demo 01 from Lesson 01 for setting up a cluster)

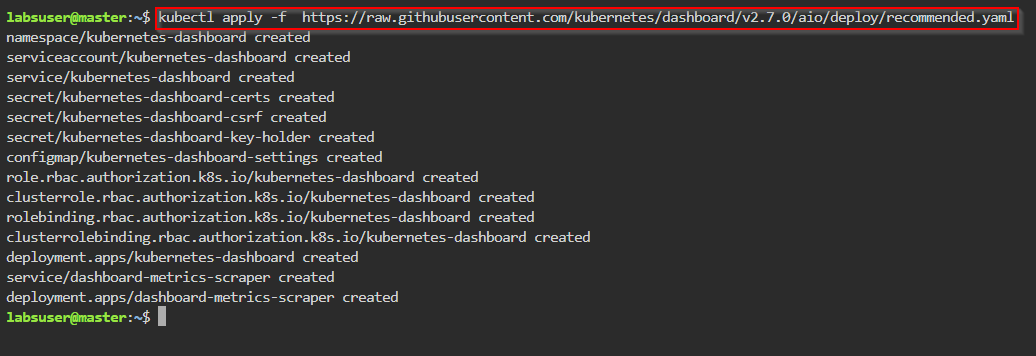
Steps to be followed:

1. Implement the dashboard deployment
2. Validate the pod, service, and deployment creation
3. Confirm the dashboard service type
4. Access the master node IP
5. Log in to the service dashboard
6. Access the Kubernetes dashboard

**Step 1:** **Implement** **the** **dashboard deployment**

* 1. Run the following command to deploy the dashboard user interface:

**kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml**



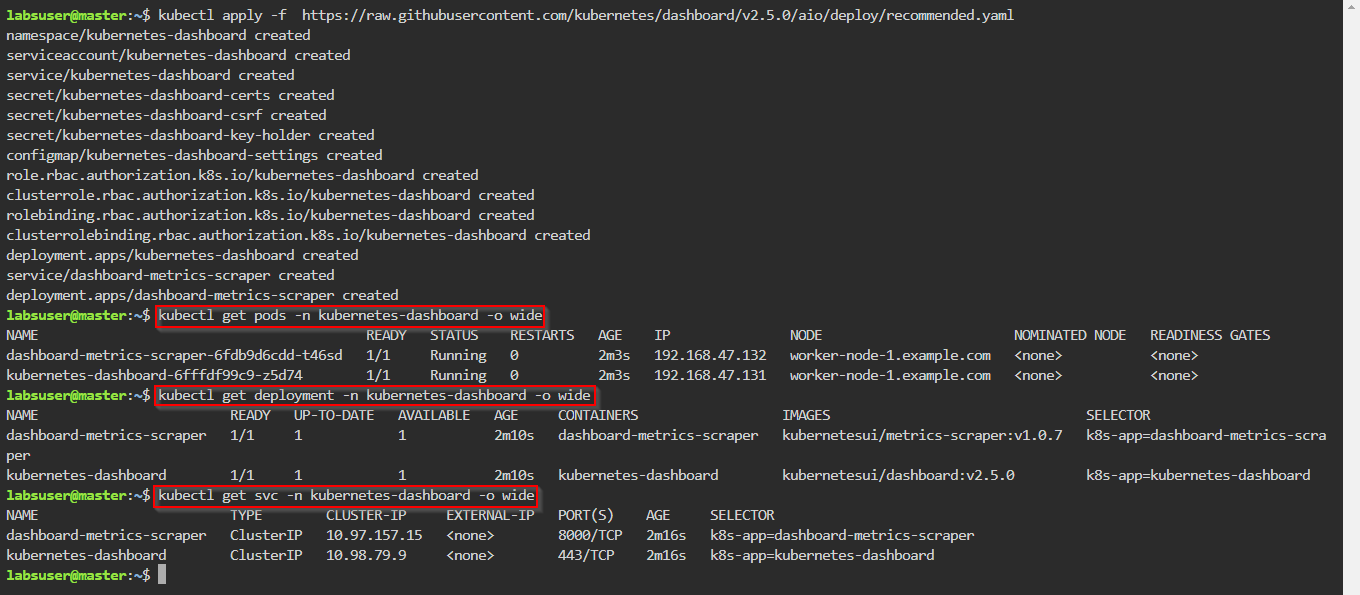
**Step 2: Validate the pod, service, and deployment creation**

1. Enter the following commands to verify if the pods, services, and deployments have been created:

**kubectl get pods -n kubernetes-dashboard -o wide**

**kubectl get deployment -n kubernetes-dashboard -o wide**

**kubectl get svc -n kubernetes-dashboard -o wide**

****

1. Run the following command to access the service outside the cluster and edit the service type from **ClusterIP** to **NodePort**:

**kubectl edit svc -n kubernetes-dashboard kubernetes-dashboard**

**A screen shot of a computer

Description automatically generated**

**A screen shot of a computer

Description automatically generated**

**A computer screen with a mouse pointer

Description automatically generated**

**Step 3: Confirm the dashboard service type**

1. Run the following command to confirm that the service type has been changed to **NodePort**:

**kubectl get svc -n kubernetes-dashboard -o wide**

A screen shot of a computer

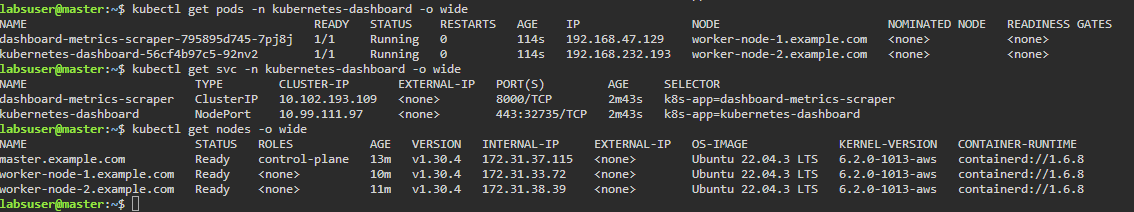
Description automatically generated

1. Run the following commands to determine the location of the pod:

**kubectl get pods -n kubernetes-dashboard -o wide**

**kubectl get svc -n kubernetes-dashboard -o wide**

**kubectl get nodes -o wide**

****

**Note:** In this case, the pod is running on **worker-node1**. Note down the **IP** and **NodePort** of node1.

1. Use the **INTERNAL-IP** as **172.31.33.72** and PORT(S) as **32735** and copy the link: **https:// 172.31.33.72:32735**

A black background with white text

Description automatically generated

**Note:** In your case, the IP and NodePort will be different. Change the IP and NodePort accordingly:

https:// <<your worker-node-1>>:<<NodePort>>

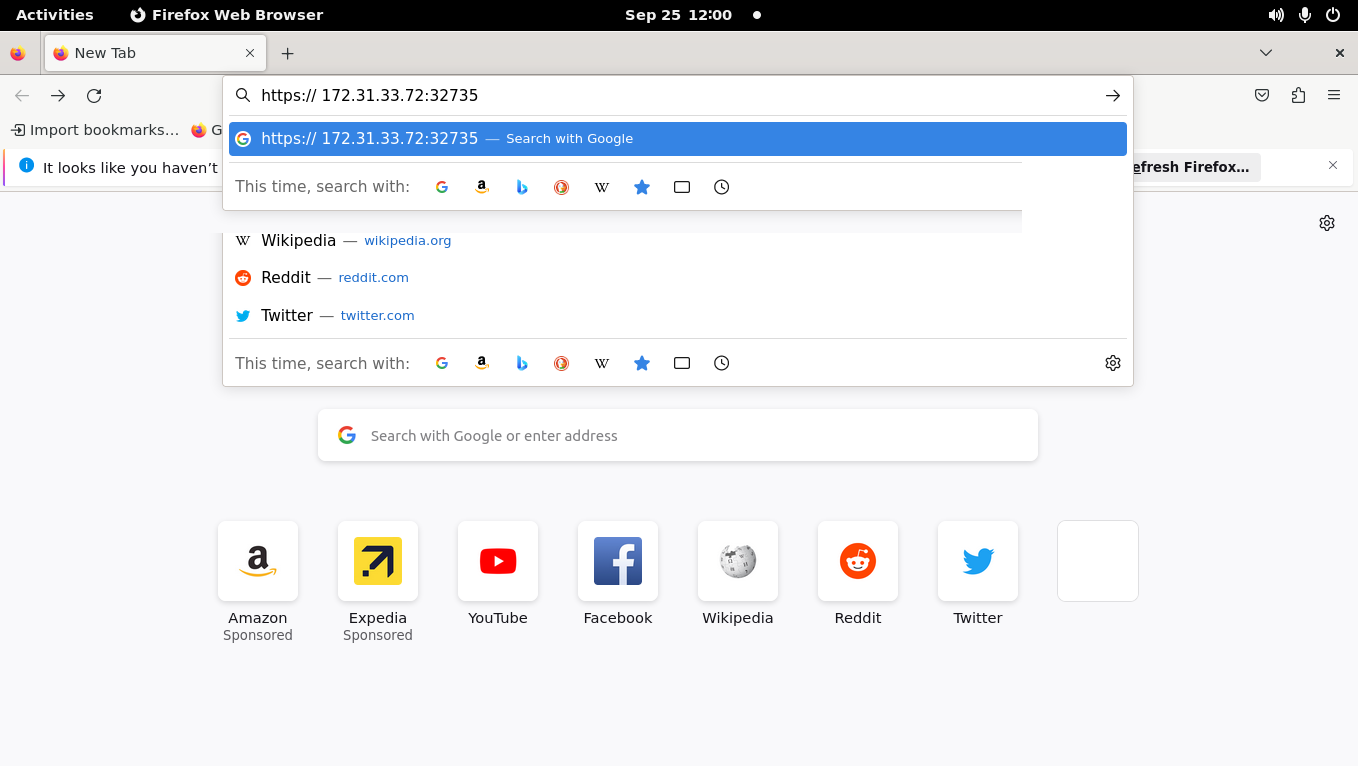
**Step 4: Access the master node IP**

1. Navigate to the LMS dashboard, click on **master**, and then click on **desktop**

**A black screen with green text

Description automatically generated**

1. Open Firefox, paste the copied link from step 3.3 in to the search bar, and press Enter

****

1. Click on the **Advanced** button

**A screenshot of a computer

Description automatically generated**

1. Click on **Accept the Risk and Continue**

**A screenshot of a computer error message

Description automatically generated**

**Step 5: Log in to the service dashboard**

1. Create a service account by running the following command and then input the code in the master node:

**vi ServiceAccount.yaml**

**A screen shot of a computer

Description automatically generated**

**apiVersion: v1**

**kind: ServiceAccount**

**metadata:**

**name: admin-user**

**namespace: kubernetes-dashboard**

A screenshot of a computer

Description automatically generated

1. Apply the YAML file with the command:

**kubectl apply -f ServiceAccount.yaml**

A computer screen with white text

Description automatically generated

1. Create a **yaml** file for cluster role binding using the below command and code:

**vi ClusterRoleBinding.yaml**

A screenshot of a computer

Description automatically generated

**apiVersion: rbac.authorization.k8s.io/v1**

**kind: ClusterRoleBinding**

**metadata:**

**name: admin-user**

**roleRef:**

**apiGroup: rbac.authorization.k8s.io**

**kind: ClusterRole**

**name: cluster-admin**

**subjects:**

**- kind: ServiceAccount**

**name: admin-user**

**namespace: kubernetes-dashboard**

A screenshot of a computer

Description automatically generated

1. Run the following command to create cluster role binding:

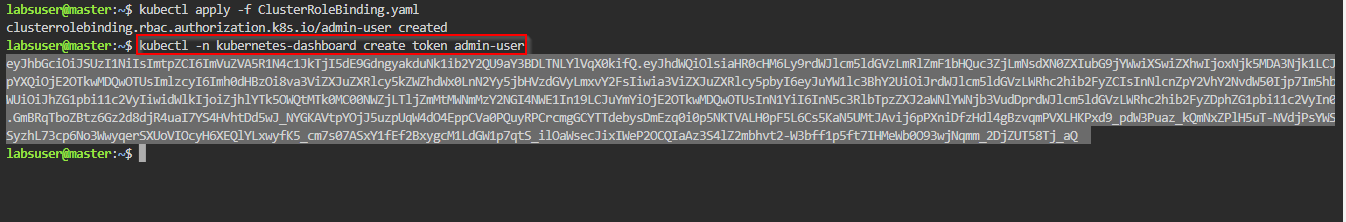
**kubectl apply -f ClusterRoleBinding.yaml**

A screen shot of a computer

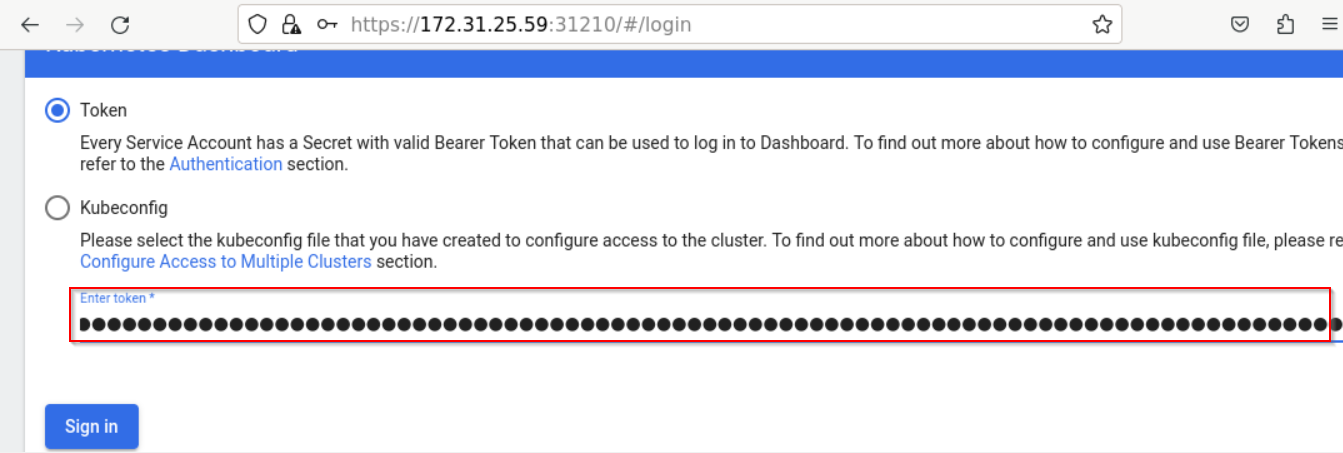
Description automatically generated

1. Retrieve the token to log in by running the following command:

**kubectl -n kubernetes-dashboard create token admin-user**

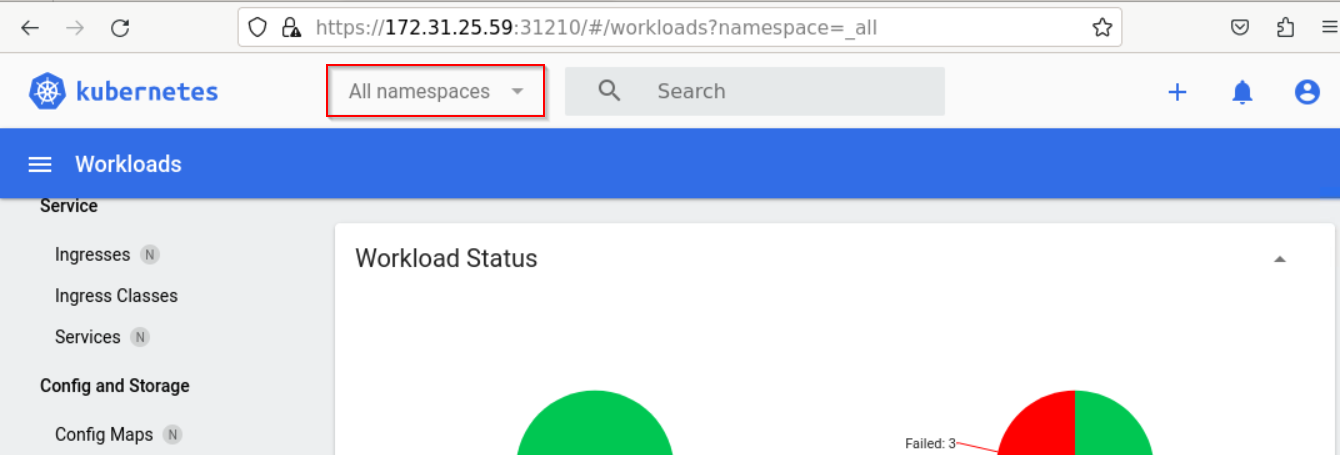
****

1. Copy the token and paste it in to the Kubernetes dashboard log in screen and then click on **Sign in**



**Step 6: Access the Kubernetes dashboard**

1. Click on the **All namespaces** drop-down menu

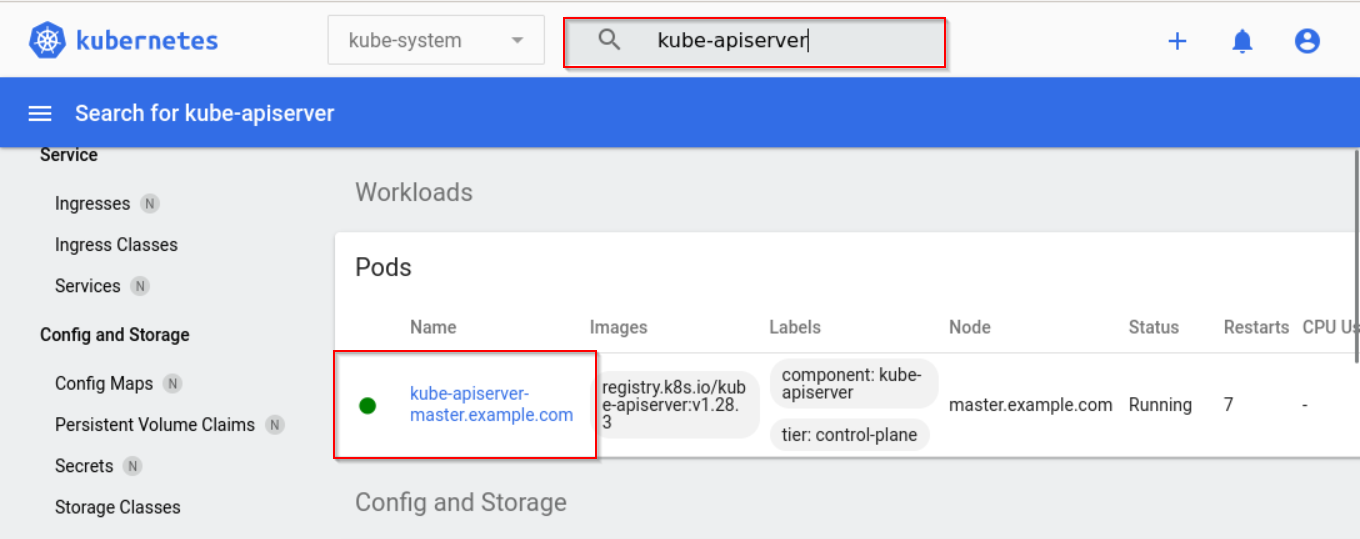
****

1. Select **kube-system**

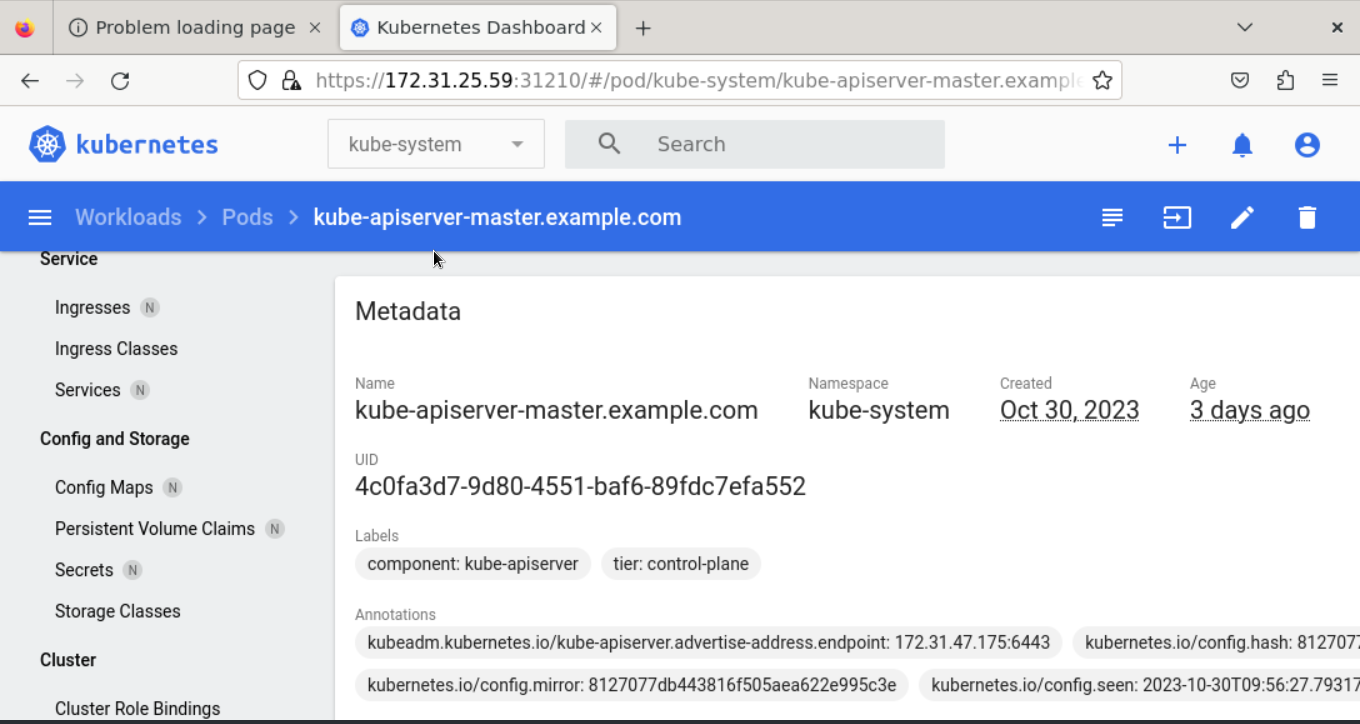
**A screenshot of a computer

Description automatically generated**

1. Use the search bar to find and select **kube-apiserver**

****

View the logs of the kube-apiserver

****

1. [OPTIONAL] Cleanup: To delete the Kubernetes dashboard version 2.5, use the following command in the master node:

**kubectl delete -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.5.0/aio/deploy/recommended.yaml**

**A screenshot of a computer program

Description automatically generated**

By following these steps, you will be able to deploy the Kubernetes Dashboard, establish secure access, and navigate the interface to monitor and manage your Kubernetes cluster.