Tutorial: Setting Up Kubernetes Dashboard on an Existing Cluster

Introduction:

In this tutorial, you'll learn how to set up the Kubernetes Dashboard, a web-based user interface for managing Kubernetes clusters. Kubernetes Dashboard provides an intuitive way to visualize and interact with your cluster's resources, making it easier to monitor and manage your applications.

What is the Kubernetes Dashboard?

Kubernetes Dashboard is a general-purpose, web-based UI for Kubernetes clusters. It allows users to:

- View and manage applications deployed on the cluster.
- Inspect and troubleshoot cluster resources.
- Monitor resource usage and health status.
- Create and manage Kubernetes objects like Pods, Deployments, Services, and more.
- Access cluster metrics and logs.

How Kubernetes Dashboard Helps:

1. Visibility: Kubernetes Dashboard provides a graphical representation of your cluster's resources, allowing you to easily visualize the current state of your applications and infrastructure.
2. Simplicity: With an intuitive user interface, Kubernetes Dashboard simplifies complex Kubernetes operations, making it accessible to developers, operators, and administrators alike.
3. Efficiency: Perform common tasks such as deploying applications, scaling resources, and troubleshooting issues directly from the dashboard, reducing the need for manual CLI interactions.
4. Monitoring: Monitor resource usage, health status, and application performance metrics, enabling proactive management and optimization of your cluster.

5. Centralized Management: Manage multiple Kubernetes clusters from a single dashboard, streamlining administrative tasks and improving overall cluster management.

Now, let's proceed with the step-by-step setup of Kubernetes Dashboard on your existing Kubernetes cluster.

Step 1: Install Helm for Ubuntu

```
curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | sudo tee
/usr/share/keyrings/helm.gpg > /dev/null sudo apt-get install apt-transport-https
--yes echo "deb [arch=$(dpkg --print-architecture)
signed-by=/usr/share/keyrings/helm.gpg] https://baltocdn.com/helm/stable/debian/
all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list
```

```
sudo apt-get update sudo apt-get install helm
```

Step 2: Deploy Kubernetes Dashboard

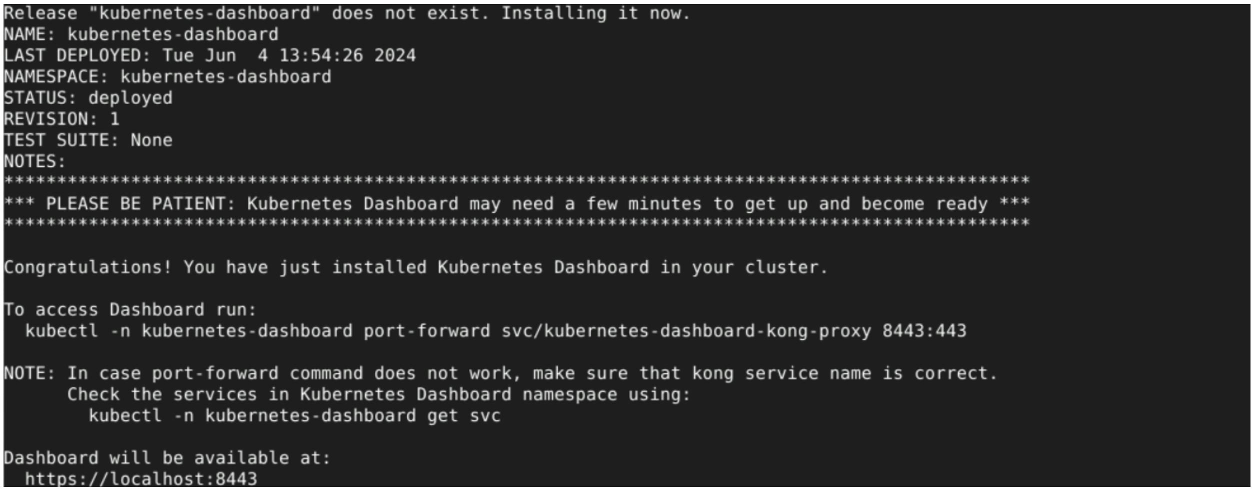
2.1 Add Kubernetes Dashboard repository

```
helm repo add kubernetes-dashboard https://kubernetes.github.io/dashboard/
```

2.2 Deploy the Dashboard

```
helm upgrade --install kubernetes-dashboard
kubernetes-dashboard/kubernetes-dashboard --create-namespace --namespace
```

```
kubernetes-dashboard
```



```
Release "kubernetes-dashboard" does not exist. Installing it now.
NAME: kubernetes-dashboard
LAST DEPLOYED: Tue Jun  4 13:54:26 2024
NAMESPACE: kubernetes-dashboard
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
*************************************************************************************************
*** PLEASE BE PATIENT: Kubernetes Dashboard may need a few minutes to get up and become ready ***
*************************************************************************************************

Congratulations! You have just installed Kubernetes Dashboard in your cluster.

To access Dashboard run:
  kubectl -n kubernetes-dashboard port-forward svc/kubernetes-dashboard-kong-proxy 8443:443

NOTE: In case port-forward command does not work, make sure that kong service name is correct.
      Check the services in Kubernetes Dashboard namespace using:
        kubectl -n kubernetes-dashboard get svc

Dashboard will be available at:
  https://localhost:8443
```

2.3 Check Deployment Status

```
helm status kubernetes-dashboard --namespace kubernetes-dashboard
```

Step 3: Access the Dashboard

3.1 Port Forward to Access the Dashboard

```
kubectl -n kubernetes-dashboard port-forward svc/kubernetes-dashboard-kong-proxy
8443:443
```

3.2 Obtain Access Token

Run the yaml file for creating service account and secret. This step is needed to create an admin-user and setup the credentials that have privileges to access the dashboard.

```
kubectl apply -f k8s-dashboard.yaml
```

Above yaml file is available here

https://github.com/devopsmay24batch/mydevopsrepo/blob/main/monitoring/k8s-dashboard.yaml

3.3 Get the token

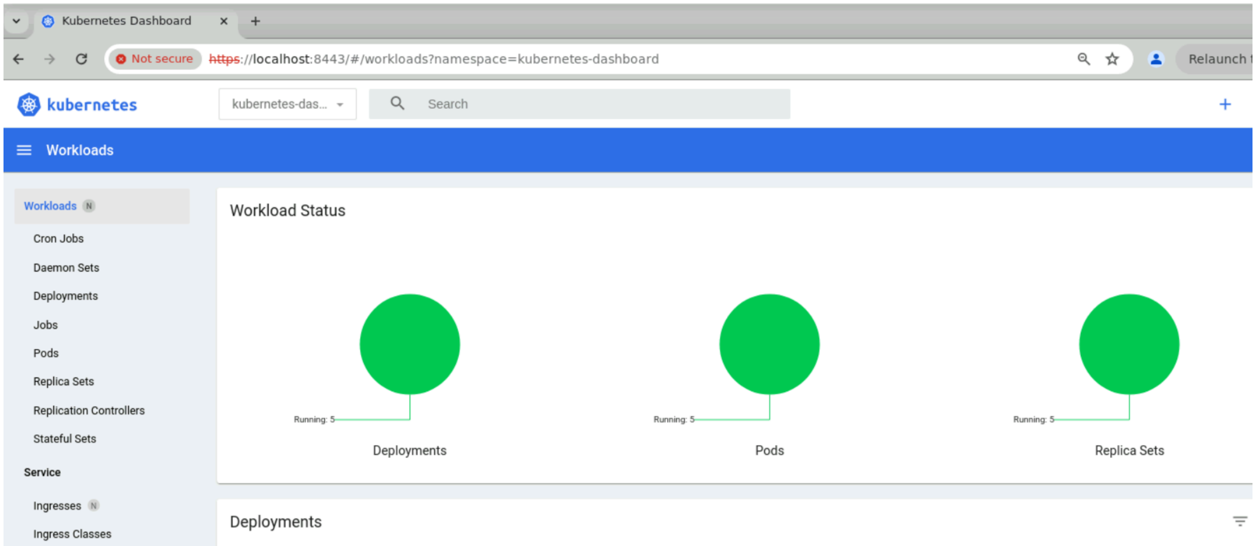
Get the token by running the below command. Copy the token from the output and use it on the dashboard page that prompts for Bearer Token:

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

3.4 Access the dashboard

Access the dashboard using https://localhost:8443 and accept the SSL certificate warning.

Explore various Workloads like PODS, Deployments, Services. This will show how kubectl shows them from a command line but all in the graphical user interface all in one dashboard.



Try the scale option to increase/decrease the Desired Replicas for the Deployment.

Conclusion:

Congratulations! You've successfully set up the Kubernetes Dashboard on your existing Kubernetes cluster. With the Dashboard installed, you now have a powerful tool at your disposal for monitoring, managing, and troubleshooting your Kubernetes workloads. Explore the features of the Dashboard to streamline your Kubernetes operations and enhance your cluster management experience.

Happy dashboarding!