# Kubernetes Dashboard Installation and Views

Last updated on May 22,2019 *29.6K Views*

Samarpit

Kubernetes Dashboard is a general purpose, web-based UI for Kubernetes clusters. It allows users to manage applications running in the cluster and troubleshoot them, as well as manage the cluster itself. So, if you get yourself ***certified in Kubernetes***, and master this tool, you can uplift your career as a DevOps Engineer.

So before moving on let us see what are the topics, we will be covering in this blog:



## What is Kubernetes Dashboard?

A Kubernetes dashboard is a web-based Kubernetes user interface which is used to deploy containerized applications to a Kubernetes cluster, troubleshoot the applications, and manage the cluster itself along with its attendant resources.

### Uses of Kubernetes Dashboard

- To get an overview of applications running on your cluster.
- To create or modify the individual Kubernetes resources for example Deployments, Jobs, etc.
- It provides the information on the state of Kubernetes resources in your cluster, and on any errors that may have occurred.

## Installing the Kubernetes Dashboard

### How to Deploy Kubernetes Dashboard?

Run the following command to deploy the dashboard:

```
kubectl create -f
https://raw.githubusercontent.com/kubernetes/dashboard/master/src/deploy/recommended/kubernetes-
dashboard.yaml
```

## Accessing Dashboard using the kubectl

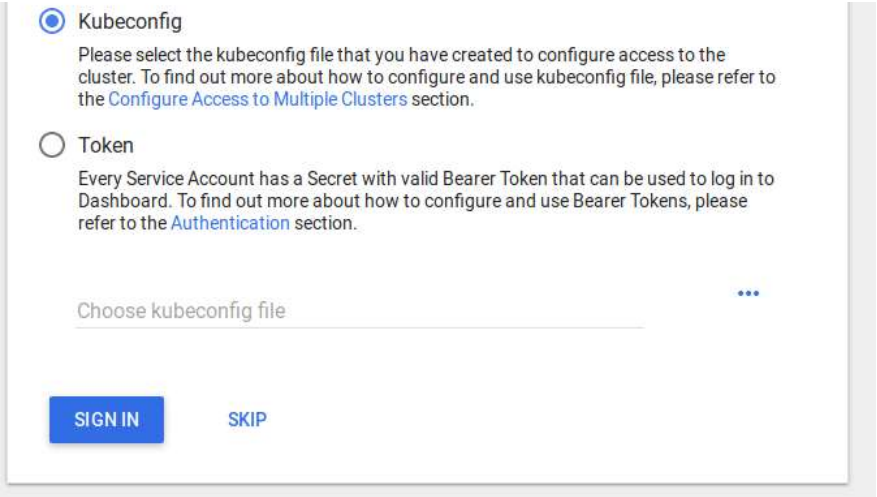
```
kubectl proxy
```

It will proxy server between your machine and Kubernetes API server.

Now, to view the dashboard in the browser, navigate to the following address in the browser of your Master VM:

```
http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/
```

You will then be prompted with this page, to enter the credentials:




### Create a Cluster Admin service account

In this step, we will create the service account for the dashboard and get its credentials.

***Note**: Run all these commands in a **new terminal**, otherwise your kubectl proxy command will stop.*



Run the following commands:

This command will create a service account for a dashboard in the default namespace

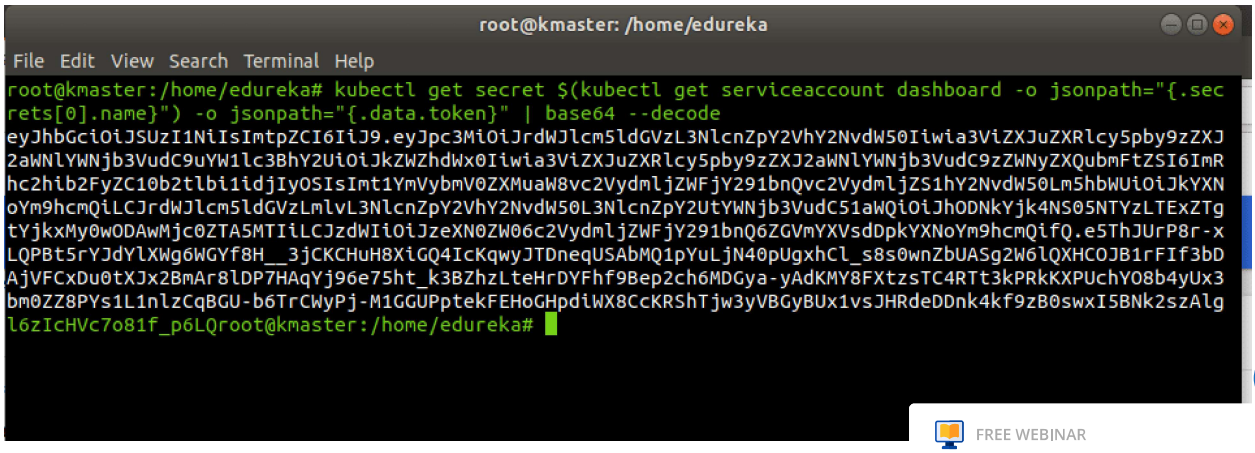
```
kubectl create serviceaccount dashboard -n default
```

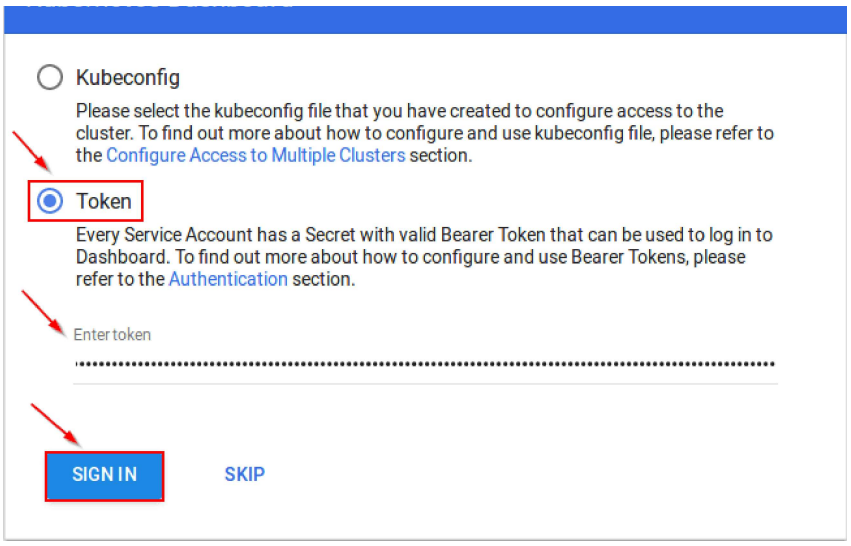
Add the cluster binding rules to your dashboard account

```
kubectl create clusterrolebinding dashboard-admin -n default  --clusterrole=cluster-admin  --
serviceaccount=default:dashboard
```

Copy the secret token required for your dashboard login using the below command:

```
kubectl get secret $(kubectl get serviceaccount dashboard -o jsonpath="{.secrets[0].name}") -o jsonpath="
{.data.token}" | base64 --decode
```



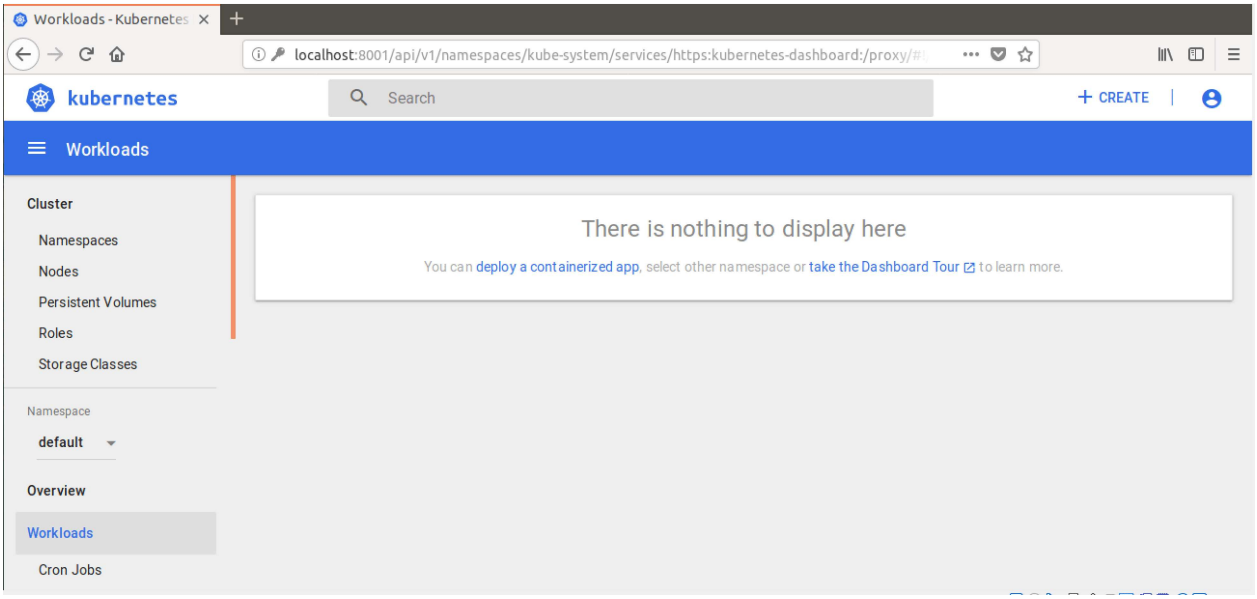



After Sign In you will land to Kubernetes Homepage.

**Home Page**

You'll see the home/welcome page in which you can view which system applications are running by default in the `kube-system` namespace of your cluster, for example, the Dashboard itself.






## Views of the Kubernetes Dashboard UI

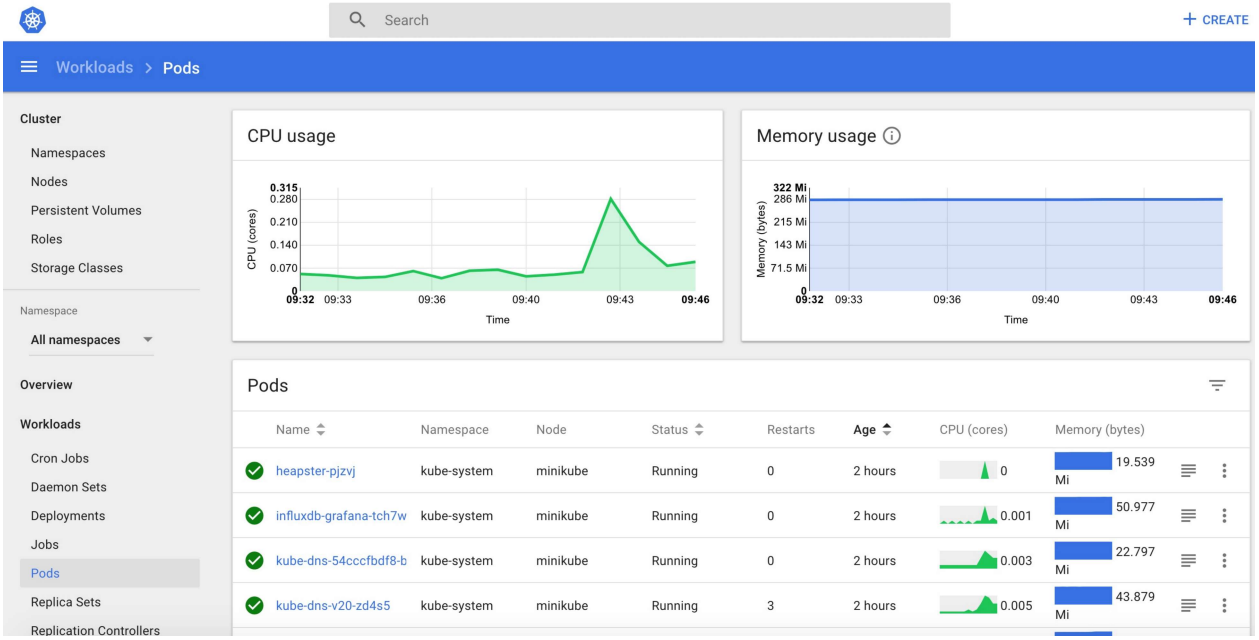
Kubernetes Dashboard consists of following dashboard views:

- Admin View

Let's start with the admin view.

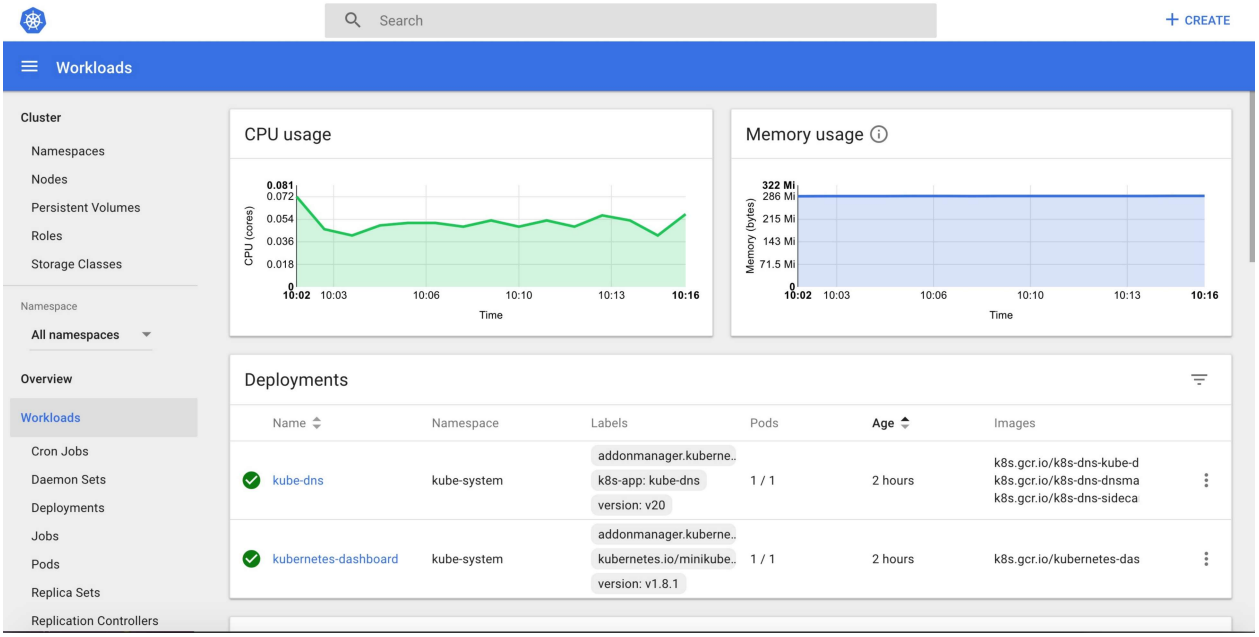
### Admin View

It lists Nodes, Namespaces, and Persistent Volumes which has a detailed view of them, where node list view contains CPU and memory usage metrics aggregated across all Nodes and the details view shows the metrics for a Node, its specification, status, allocated resources, events, and pods running on the node.



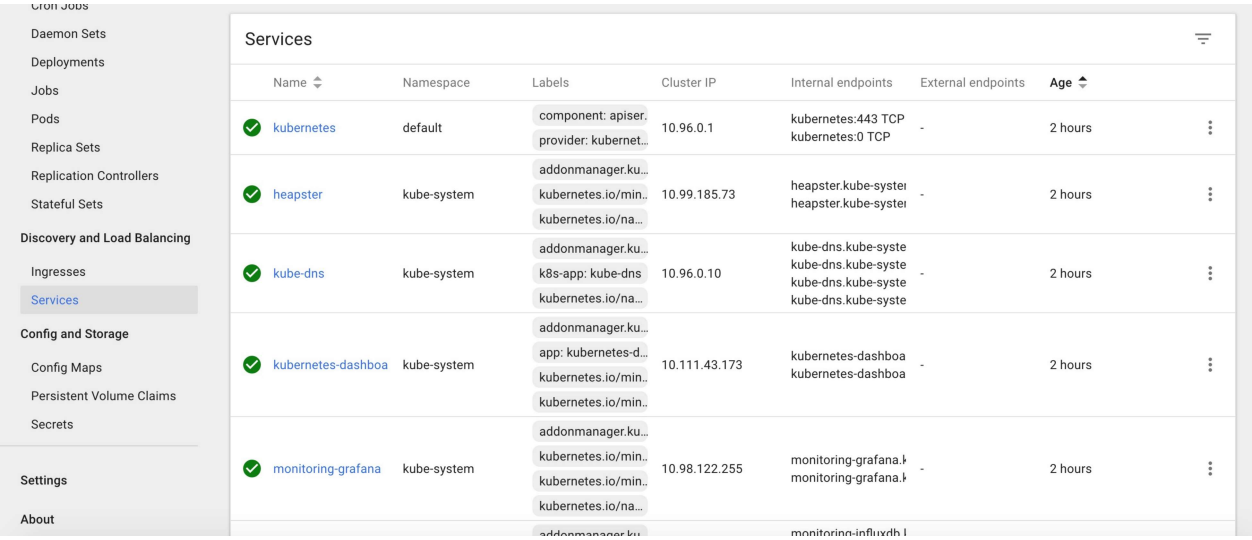

### Workloads View

It is the entry point view that shows all applications running in the selected namespace. It summarizes the actionable information about the workloads, for example, the number of ready pods for a Replica Set or current memory usage for a Pod.




### Services View

It shows the shows Kubernetes resources that allow for exposing services to the external world and discovering them within a cluster.




**Storage and Config View**

The Storage view shows Persistent Volume Claim resources which are used by applications for storing data whereas config view is used to shows all the Kubernetes resources that are used for live configuration of applications running in clusters.



*Got a question for us? Please mention it in the Continuous Integration Tools comments section and we will get back to you.*