

How to Install Kubernetes Dashboard

By Mohamed Ettayeb - January 25, 2019

As a part of the [Kubernetes tutorials for beginners](#) series, my goal is to offer a full and solid list of articles that goes through the very basics definitions, the history and the need to use Kubernetes and containers until i reach the deep parts, so regardless of your technical background, i will offer you everything you need here to master Kubernetes step by step.

After installing and deploying Kubernetes for the first time, i am sure some of you wondering if there is a GUI solution to manage the Kubernetes cluster and the answer is: Yes! sure there is a full Kubernetes dashboard and it's the topic of this article! so let's take a look on how to install and use it.

What is the Kubernetes Dashboard

The Kubernetes dashboard is a friendly web user interface used to administrate and deploy containerized applications to a Kubernetes cluster, troubleshoot those applications, and manage the cluster itself along with its attendant resources.

This is the Github official repository of the Kubernetes Dashboard:

<https://github.com/kubernetes/dashboard>

You can take a look on it and maybe visualize the code there.

Uses of Kubernetes Dashboard

The Kubernetes Dashboard facilitate a lot of tasks for you like:

- Getting an overview of applications running on your cluster.
- Creating or updating the individual Kubernetes resources, e.g. deployments, jobs, etc.
- Providing the information on the state of Kubernetes resources in your cluster, and on any errors that may have occurred.

How to install the Kubernetes Dashboard

Initial Setup

You need to have a running Kubernetes cluster, if you don't, you can setup one by following one of our last articles:

[How to Install and Deploy Kubernetes on Ubuntu 16.04](#)

[How to Install and Deploy Kubernetes on CentOS 7](#)

Install the Kubernetes Dashboard

Run the following command on the master machine to deploy the dashboard:

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v1.10.1/src/deploy/recommend
```

Access to the Dashboard using the kubectl

To get the access to the Dashboard you must create a secure channel to your Kubernetes cluster by running the following command:

```
kubectl proxy
```

When the proxy server is running, you can explore the API using `curl`, `wget`, or a browser.

Next, to access to the dashboard in the browser, navigate to the following address in the browser of your Master machine:

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

Then you will get the following modal, to enter the credentials:

Our next step, will be creating the service account for the dashboard and get its credentials.

Note: You should run all these commands in a **new terminal**, otherwise your `kubectl proxy` command will stop.

The following command will create a service account for a dashboard in the default namespace:

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

Next, just add the cluster binding rules to your dashboard account with this command:

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

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 jsonp
```

How to Install Kubernetes Dashboard

Finally, just copy the secret token and paste it in Dashboard Login interface, by selecting the **token** option (the second radiobox). After Sign In you will be redirected to the Kubernetes Dashboard Homepage.

You'll see the homepage in which you can view the system applications that are running by default.

Kubernetes Dashboard UI Views

The Kubernetes Dashboard contains the following dashboard views:

The first one is the admin view.

Admin View

On it you can find the Nodes, the Namespaces, the Persistent Volumes and 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.

Admin View

Workloads View

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

Services View

Here you can find the Kubernetes resources that allow for exposing services to the external world and discovering them within a cluster.

Services View

Storage and Config View

The Storage view is where you can find the Persistent Volume Claim resources which are used by applications for storing data.

Finally, the config view which is used to shows all the Kubernetes resources that are used for live configuration of applications running in clusters.

Storage and Config View

This is the end of this article so i hope that you understanding everything until this step, if you facing any type of problems do not hesitate to comment it here and get some help.

Mohamed Ettayeb

I m a Tunisian geek who start digging on technology information since 8 years ,i am a fullstack javascript web developer and a pentester. There is nothing that i cannot learn.