

1 Kubernetes

This document provides some instructions to work with Kubernetes cluster.

1.1 K8s Dashboard

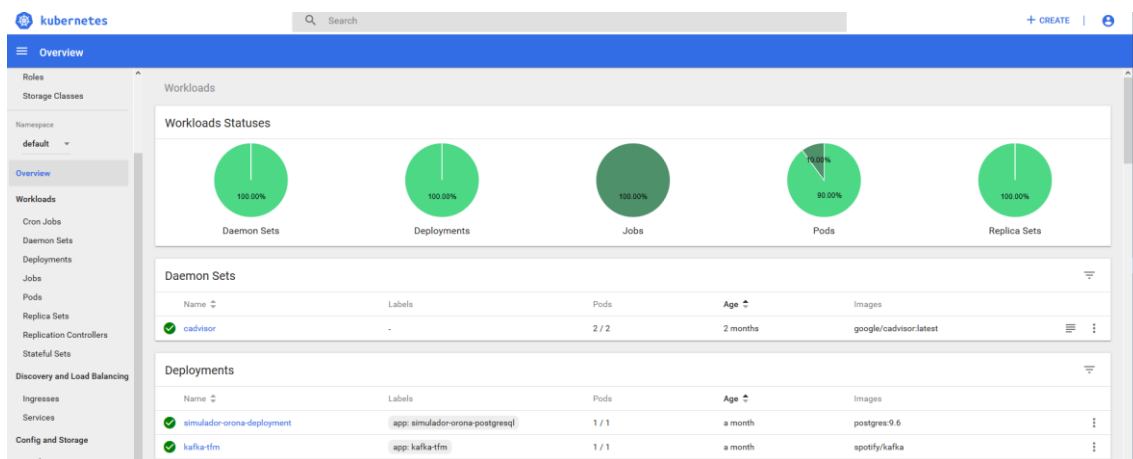
By default, the Kubernetes Dashboard is not accesible externally, so we have to proxy it using the next command on one of the masters (make sure to replace the MASTER-IP with your real IP):

```
$ kubectl proxy --address <<MASTER-IP>> --accept-hosts='^*$'
```

Once it started, we can access the Kubernetes cluster by the next link (make sure to replace the MASTER-IP with your real IP):

```
http://<<MASTER-IP>>:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/
```

So we can see the next UI:



1.2 Scale K8s

This section explained how to add new agents to the Kubernetes cluster that is actually deployed and running, without having any downtime in local servers and how to scale in the AWS cloud.

1.2.1 Scale K8s locally

1. The cluster nodes and the new nodes we are going to add, they have to know each other by hostname (in case it is actually configured, go to the step 2).
 - Deploy **hostnames** playbook.

```
$ make hostnames
```

2. Besides knowing each other, they also have to have SSH Access without password each other (in case it is actually configured, go to the step 3).

- Deploy **ssh-keys** playbook

```
$ make ssh-keys
```

3. Scale Kubernetes cluster.

- Add the new IPs to the “inventory” of the **kubernetes-ha** playbook (keeping the actual cluster nodes IPs).
- Deploy **kubernetes-ha** playbook.

```
$ make kubernetes-ha
```

1.2.2 Scale K8s on AWS

1. The first thing we have to do, is edit file “variables.tf”:
 - Private_slaves_count: (Number of private slaves you want to have, if they are 2, and you want 3 more, you have to put 5).
2. After updating the number of slaves we want to add to the cluster, just run again the “setup-cluster.sh” script and it will automatically configure and add the new agents to the Kubernetes cluster. (“Check the script and comment all the playbooks you don’t want to scale, by default it scale all the platform”).

1.3 Remove K8s node

Using the following command, we can remove completely a node from the kubernetes cluster:

```
$ sudo kubeadm reset
```