K8s Cluster

The project's root folder is install_k8s. The project to create the cluster was divided into roles to facilitate the execution of roles. The structure of the roles was divided as follows:

Roles

- create-cluster
  This as the name says creates the cluster. The main file for this role is: init-cluster.yml

- get-token
  This captures the token in an auxiliary way in the playbooks (main-newwrk.yml, main-master.yml) so that they have the keys and certificates to add the new servers to the cluster. The main file for this role is: init-cluster.yml

- ha-proxy
  This configures the HA proxy. The main file for this role is: init-cluster.yml

- install-helm
  This performs the download of the binaries for the execution of the helm that is used in the creation of the monitoring.

- install-k8s
  This performs the installation of the packages: docker, kubectl, kubeadmin, kubelet.

- install-monit-tools
T his installs the prometheus operator stack.

- join-workers
  This adds workers to the cluster when deploying the cluster.

- new-master
  This adds the new masters to the existing cluster.

- new-workers
  This adds the new masters to the existing cluster.

Playbook files and hosts.

In the ***hosts f***ile we have all groups of computers used by ansible:

In the ***[kubernetes]*** group, we must place all hosts that will have the kubernetes role, for future actions that must be performed in all kubernetes.

In the ***[k8s-master]*** group there should be all servers that have the role of master.

In the ***[k8s-workers]*** group there should be all servers that have the role of worker.

In the ***[k8s-master-new]*** group there must be all servers that will become master, after the completion of the main-master.yml playbook the server must be removed from this group.

In the ***[k8s-master-init]*** group there is the server that was used for the initial creation of the cluster.

In the ***[k8s-master-new]*** group there must be all the servers that will become a worker, after the completion of the main-newwrk.yml playbook the server should be removed from this group.

In the group ***[ha-proxy]*** there is the server that is the haproxy

In the ***[k8s-workers: vars]*** and ***[k8s-master-init: vars]*** groups there are variations used in the creation of the cluster.

Resolution Name

It is very important that all machines have hosts registered in /etc/hosts.

Execution playbooks

main-base.yml: Install base programs to run ansible, it must be run on all servers.
main-createcluster.yml: This cleans up any old configuration and recreates the cluster.
main-Install-k8s.yml: Install docker, kubeadm, kubelet, kubectl.
main-master.yml: Add a new server with the master paple.
main-monititools.yml: Installs prometheus operator monitoring.
main-newwrk.yml: Add a new server with the worker paple.

main-proxy.yml: install the haproxy packages and configure it.

Structure

The structure used was the re-command by kubernetes inc, following the structure of figure 1

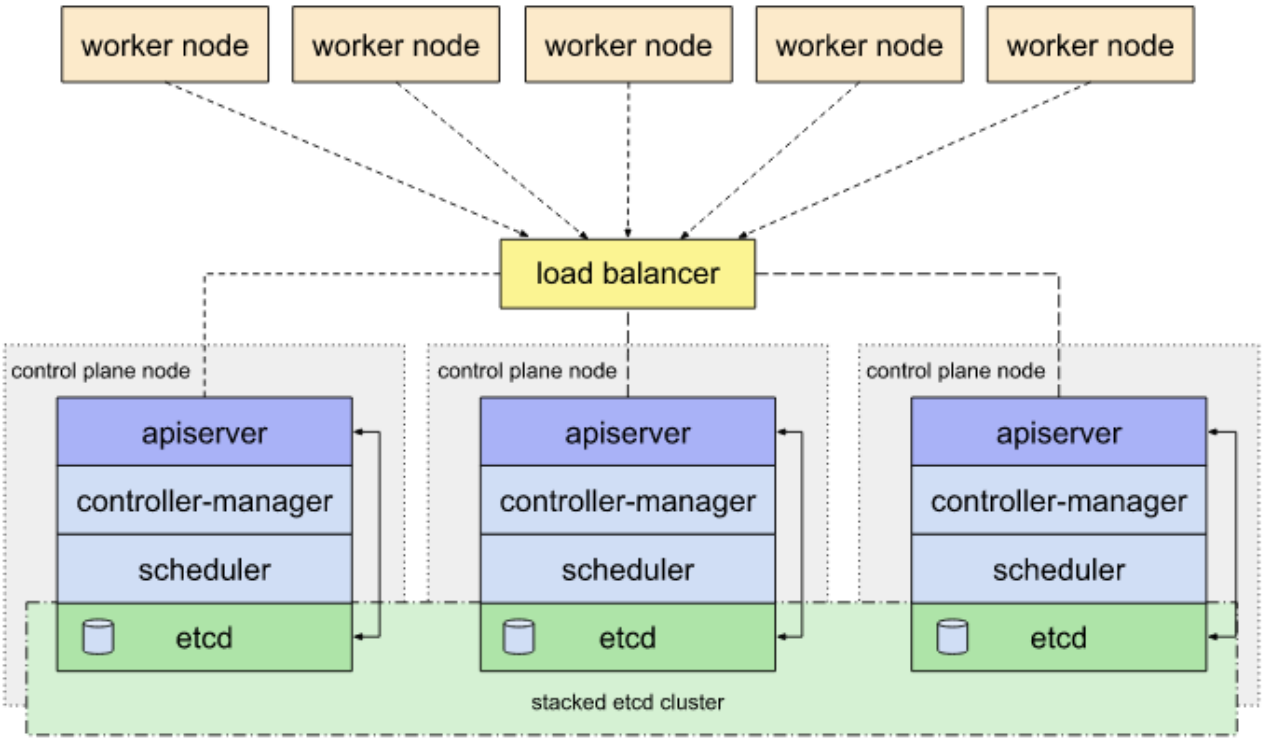


Figura 1

In this image we have the drawing of how the cluster is configured. For the environment.

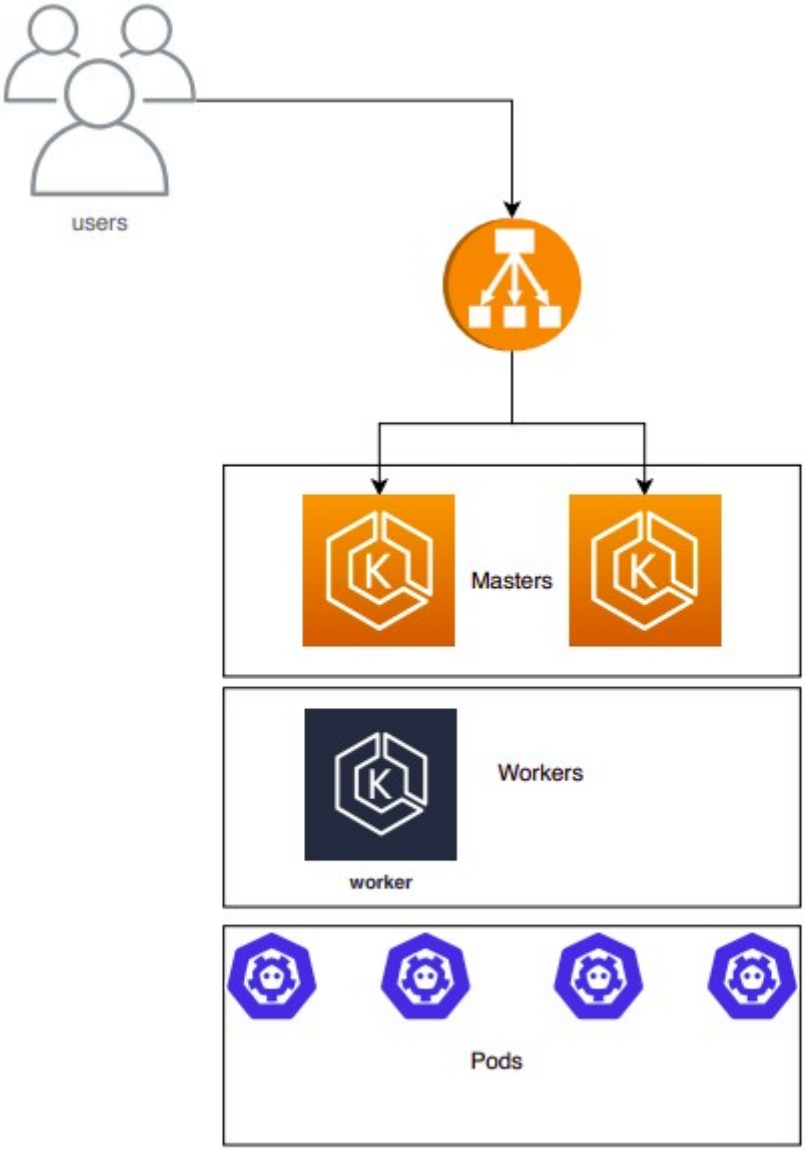
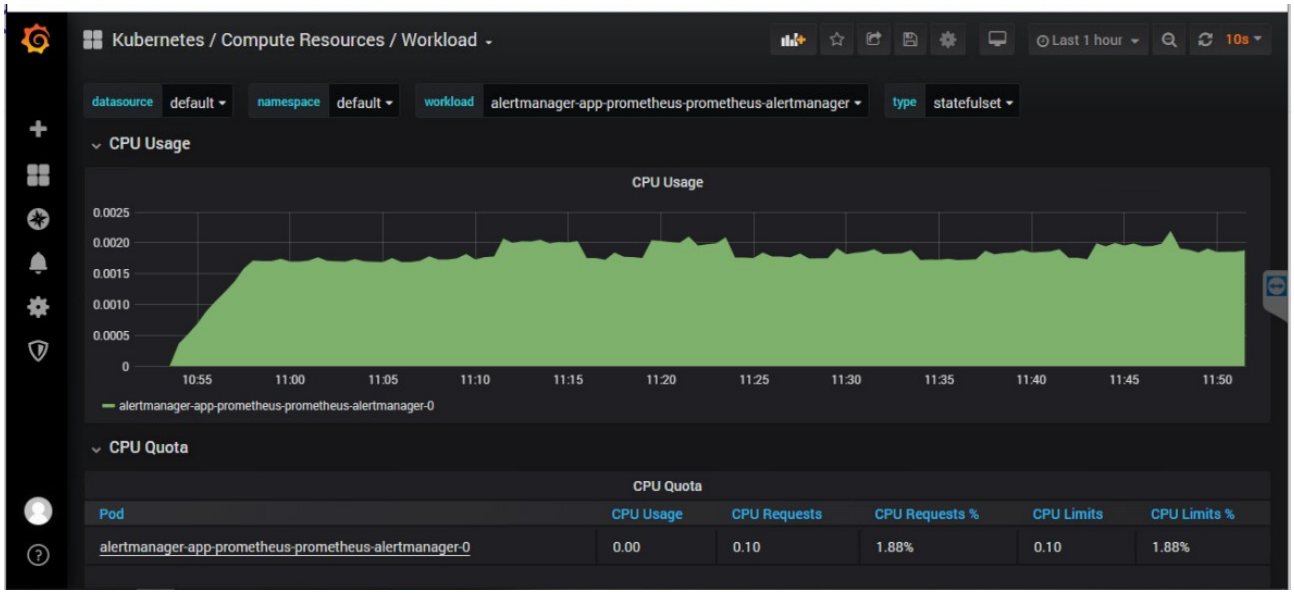


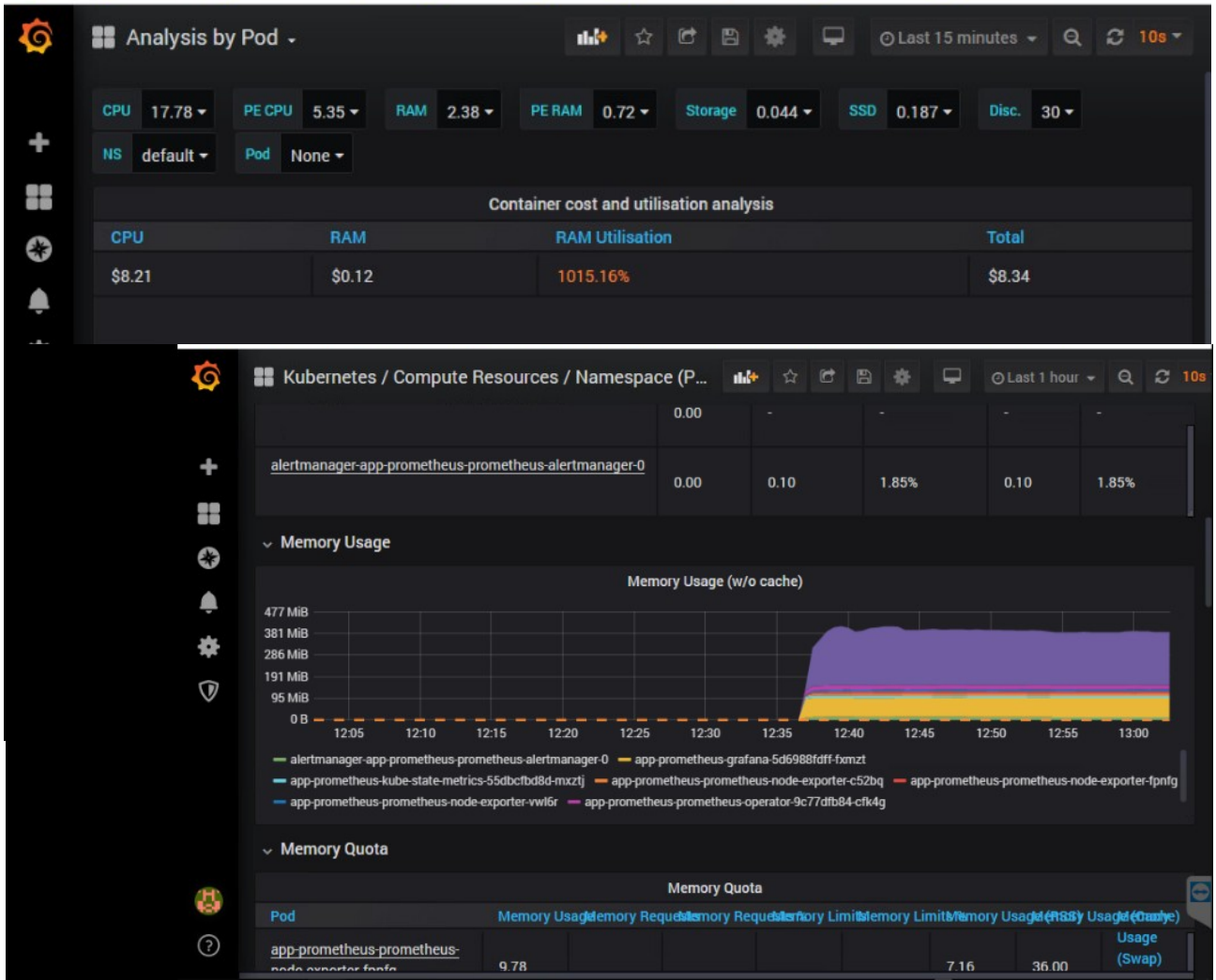
Figura 2:

```
root@haproxy:~# logout
fibo@haproxy:~$ logout
Connection to haproxy closed.
fibo@node1:~$ logout
root@node1:~# kubectl get nodes
NAME    STATUS   ROLES    AGE   VERSION
node1   Ready    master   57m   v1.18.2
node2   Ready    master   45m   v1.18.2
node3   Ready    <none>   47m   v1.18.2
```

Monitoring

For the monitoring we used the helm with the prometheus-monitor stack. These are some of the grafana dashboards:






Kubernetes / Compute Resources / Pod
Last 1 hour
10s
CPU Usage
0.15
0.10
0.05
0
10:55
11:00
11:05
11:10
11:15
11:20
11:25
11:30
11:35
11:40
11:45
11:50
alertmanager
config-reloader
requests
limits
CPU Throttling
CPU Throttling
100.00%
75.00%
50.00%
25.00%
0%


Kubernetes / Compute Resources / Cluster
Last 1 hour
10s
datasource
default
Headlines
CPU Utilisation
N/A
CPU Requests Commi...
9.250%
CPU Limits Commitm...
2.500%
Memory Utilisation
11.77%
Memory Requests Co...
1.761%
Memory Limits Comm...
1.761%
CPU
CPU Usage
0.20
0.15
0.10
0.05
0
10:55
11:00
11:05
11:10
11:15
11:20
11:25
11:30
11:35
11:40
11:45
11:50
default
kube-system