Howdy!.

This blog post explains how to set up dashboard that can contrl k8s cluster or monitor resources.

Prereqs)

<https://wnapdlf.blogspot.com/> ‘s posts about kvm+vagrant+k8s... && linux command && or more.

1. Frist create dashboad and account

2. Proxy and ssh tunneling to connecto from local pc(work station).

3. Connect url , paste secret and test.

4. When things are not went smooth.

5. Create pod with dashboard.

Refered site)https://kubernetes.io/docs/tasks/access-application-cluster/web-ui-dashboard/

1. Frist create dashboad and account

[vagrant@kubemaster ~]$ wget https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0-beta1/aio/deploy/recommended.yaml

--2019-07-12 18:40:09-- https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0-beta1/aio/deploy/recommended.yaml

Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.228.133

Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.228.133|:443... connected.

HTTP request sent, awaiting response... 200 OK

Length: 6920 (6.8K) [text/plain]

Saving to: ‘recommended.yaml’

recommended.yaml 100%[==================================================>] 6.76K --.-KB/s in 0s

2019-07-12 18:40:10 (63.0 MB/s) - ‘recommended.yaml’ saved [6920/6920]

[vagrant@kubemaster ~]$ kubectl create -f recommended.yaml

namespace/kubernetes-dashboard created

serviceaccount/kubernetes-dashboard created

service/kubernetes-dashboard created

secret/kubernetes-dashboard-certs created

secret/kubernetes-dashboard-csrf created

secret/kubernetes-dashboard-key-holder created

configmap/kubernetes-dashboard-settings created

role.rbac.authorization.k8s.io/kubernetes-dashboard created

clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created

rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created

clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created

deployment.apps/kubernetes-dashboard created

service/dashboard-metrics-scraper created

deployment.apps/kubernetes-metrics-scraper created

[vagrant@kubemaster ~]$ kb get po -n kubernetes-dashboard

NAME READY STATUS RESTARTS AGE

kubernetes-dashboard-5c8f9556c4-tdgh5 1/1 Running 0 7m43s

kubernetes-metrics-scraper-86456cdd8f-q5cn8 1/1 Running 0 7m43s

**2.** **Proxy and ssh tunneling to connecto from local pc(work station).**

[vagrant@kubemaster ~]$ kubectl proxy&

[1] 3196

[vagrant@kubemaster ~]$ Starting to serve on 127.0.0.1:8001

[vagrant@kubemaster ~]$

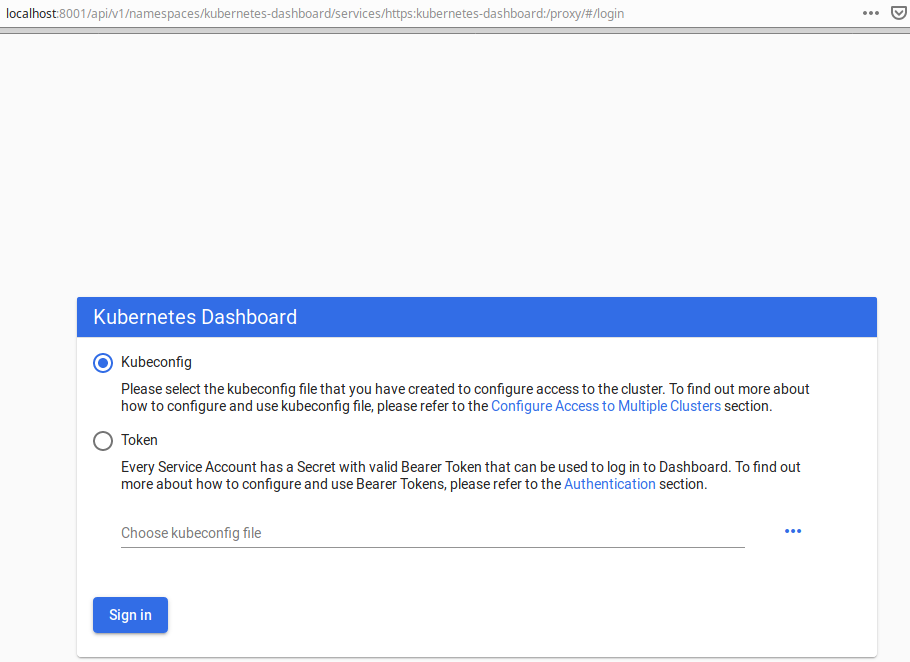
oyj@oyj-X555QG:~/INSTALL/u18kvk8s/k8s/dashboard$ ssh -i ../id\_rsa -L 8001:localhost:8001 vagrant@10.1.0.2

Last login: Fri Jul 12 10:02:03 2019 from 10.1.0.1

[vagrant@kubemaster ~]$

#From localhost with browser

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

[

**3. 3. Connect url , paste secret and test.**

**1) create sample account user.**

Referred site) https://github.com/kubernetes/dashboard/wiki/Creating-sample-user

oyj@oyj-X555QG:~/INSTALL/u18kvk8s/k8s/dashboard$ cat adm-user.yaml

apiVersion: v1

kind: ServiceAccount

metadata:

name: admin-user

namespace: kube-system

oyj@oyj-X555QG:~/INSTALL/u18kvk8s/k8s/dashboard$ kb create -f adm-user.yaml

serviceaccount/admin-user created

oyj@oyj-X555QG:~/INSTALL/u18kvk8s/k8s/dashboard$ cat cluster-rold-bind.yaml

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: admin-user

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole

name: cluster-admin

subjects:

- kind: ServiceAccount

name: admin-user

namespace: kube-system

oyj@oyj-X555QG:~/INSTALL/u18kvk8s/k8s/dashboard$ kb create -f cluster-rold-bind.yaml

clusterrolebinding.rbac.authorization.k8s.io/admin-user created

#Bearer Token

oyj@oyj-X555QG:~/INSTALL/u18kvk8s/k8s/dashboard$ kubectl -n kube-system describe secret $(kubectl -n kube-system get secret | grep admin-user | awk '{print $1}')

Name: admin-user-token-d9wbc

Namespace: kube-system

Labels: <none>

Annotations: kubernetes.io/service-account.name: admin-user

kubernetes.io/service-account.uid: 2193e725-5690-437a-a354-1086f2edd98a

Type: kubernetes.io/service-account-token

Data

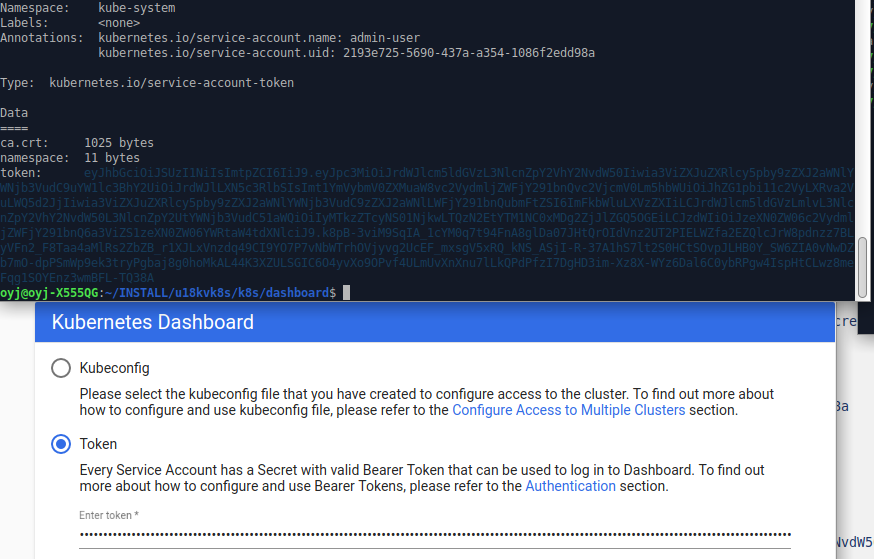
====

ca.crt: 1025 bytes

namespace: 11 bytes

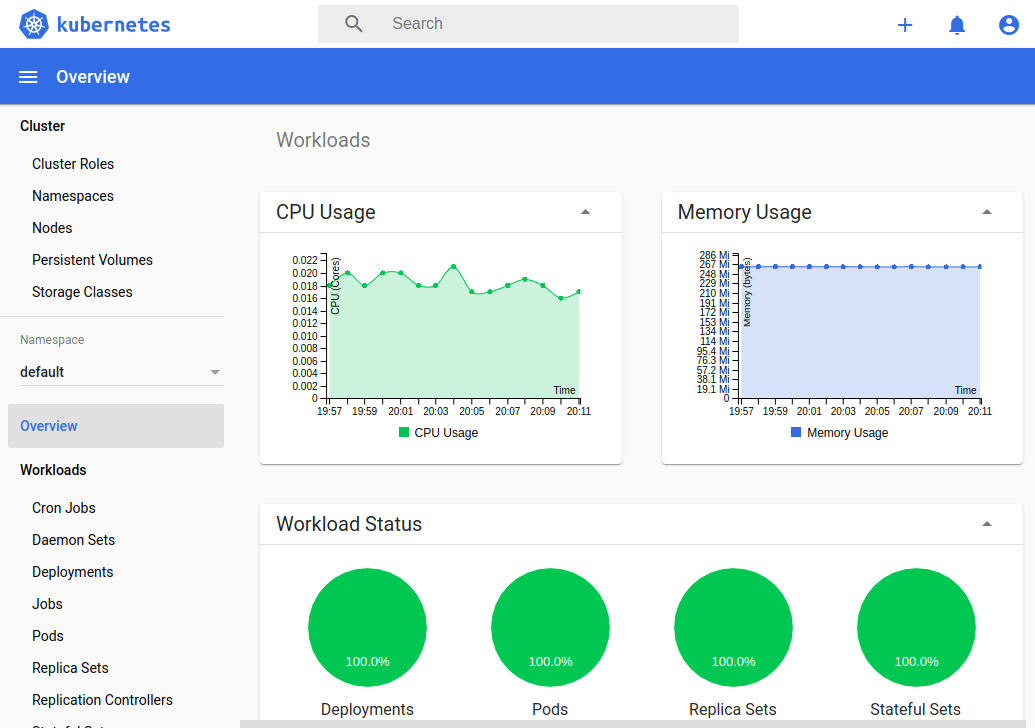
token: eyJhbGciOiJSUzI1NiIsImtpZCI6IiJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlLXN5c3RlbSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJhZG1pbi11c2VyLXRva2VuLWQ5d2JjIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6ImFkbWluLXVzZXIiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC51aWQiOiIyMTkzZTcyNS01NjkwLTQzN2EtYTM1NC0xMDg2ZjJlZGQ5OGEiLCJzdWIiOiJzeXN0ZW06c2VydmljZWFjY291bnQ6a3ViZS1zeXN0ZW06YWRtaW4tdXNlciJ9.k8pB-3viM9SqIA\_1cYM0q7t94FnA8glDa07JHtQrOIdVnz2UT2PIELWZfa2EZQlcJrW8pdnzz7BLyVFn2\_F8Taa4aMlRs2ZbZB\_r1XJLxVnzdq49CI9YO7P7vNbWTrhOVjyvg2UcEF\_mxsgV5xRQ\_kNS\_ASjI-R-37A1hS7lt2S0HCtSOvpJLHB0Y\_SW6ZIA0vNwDZb7mO-dpPSmWp9ek3tryPgbaj8g0hoMkAL44K3XZULSGIC6O4yvXo9OPvf4ULmUvXnXnu7lLkQPdPfzI7DgHD3im-Xz8X-WYz6Dal6C0ybRPgw4IspHtCLwz8meFqg1SOYEnz3wmBFL-TQ38A

#Copy and paste above token like below pic.



**2) Connect and monitor.**

#After login we should see like below.



**4. When things are not went smooth.**

#We can debug with like below.

[vagrant@kubemaster ~]$ kb get pod -n kubernetes-dashboard

NAME READY STATUS RESTARTS AGE

kubernetes-dashboard-5c8f9556c4-tdgh5 1/1 Running 0 101m

kubernetes-metrics-scraper-86456cdd8f-q5cn8 1/1 Running 0 101m

[vagrant@kubemaster ~]$ kb logs -f kubernetes-dashboard-5c8f9556c4-tdgh5 -n kubernetes-dashboard

**5. Create pod with dashboard.**

Conclusion)

With dashboard setup, basically we can do anything that we can do in console command.

It is wonderful tool that is being provided by community and google.com.

It is great for basic monigoring on k8s cluster. Sometimes we can deploy k8s resources(pods,services,sts and so on) with this dashboard when we cannot use console and so on.

Thanks for reading!.