**Installation of nodes in Kubernetes cluster**

Link:- <https://phoenixnap.com/kb/how-to-install-kubernetes-on-centos>

Prerequisite:-

1. Docker for containerization

2. Disable swap memory

*Sudo swapoff -a*

Comment swap memory in file

*Vi /etc/fstab*

OR

*sudo swapoff -a*

*sudo sed -i '/ swap / s/^/#/' /etc/fstab*

Step 1:- Change host name for identification of Master and Worker node.

Link:- https://www.cyberciti.biz/faq/ubuntu-change-hostname-command/

*sudo vi /etc/hostname*

*sudo vi /etc/hosts*

*Sudo shutdown now -r*

Step 2:- Install kubeadm Master as well as Worker nodes

Link:- https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/

* Update the apt package index and install packages needed to use the Kubernetes apt repository:

*sudo apt-get update sudo apt-get install -y apt-transport-https ca-certificates curl*

* Download the Google Cloud public signing key:

*sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg*

* Add the Kubernetes apt repository:

*echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main"| sudo tee /etc/apt/sources.list.d/kubernetes.list*

* Update apt package index, install kubelet, kubeadm and kubectl, and pin their version:

*sudo apt-get update*

*sudo apt-get install -y kubelet kubeadm kubectl*

*sudo apt-mark hold kubelet kubeadm kubectl*

**On Master node:-**

* Docker change cgroup driver to systemd:-

Link:-https://stackoverflow.com/questions/43794169/docker-change-cgroup-driver-to-systemd

A solution that does not involve editing systemd units or drop-ins would be to create (or edit) the *sudo vi /etc/docker/daemon.json* configuration file and to include the following:

*{*

*"exec-opts": ["native.cgroupdriver=systemd"]*

*}*

After saving it, restart your docker service.

*sudo systemctl restart docker*

Installing a Pod network add-on

You must deploy a [Container Network Interface](https://v1-17.docs.kubernetes.io/docs/concepts/extend-kubernetes/compute-storage-net/network-plugins/#cni) (CNI) based Pod network add-on so that your Pods can communicate with each other.  
**We use Flannel here.**

* To initialize Kubernetes Master below command run on Master node:-

Link:- https://v1-17.docs.kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/

*sudo kubeadm init --pod-network-cidr=10.244.0.0/16*

To start using your cluster, you need to run the following as a regular user:

*mkdir -p $HOME/.kube*

*sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config*

*sudo chown $(id -u):$(id -g) $HOME/.kube/config*

Alternatively, if you are the root user, you can run:

*export KUBECONFIG=/etc/kubernetes/admin.conf*

* Use below command to check available nodes,pods:-

*kubectl get nodes*

*kubectl get pods*

*kubectl get pods -A*

*You should now deploy a pod network to the cluster.*

*Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:*

*https://kubernetes.io/docs/concepts/cluster-administration/addons/*

***kubectl apply -f <https://raw.githubusercontent.com/coreos/flannel/2140ac876ef134e0ed5af15c65e414cf26827915/Documentation/kube-flannel.yml>***

Then you can join any number of worker nodes by running the following on each as root:

*sudo kubeadm join 192.168.11.188:6443 --token 3g3k2q.g2jjxc45u1qaak13 \*

*--discovery-token-ca-cert-hash sha256:4f7ba14a7e52b72dd17569b36039c211740d7982e73acbf43f6451f5a56aeb63*

**Troubleshooting kubeadm**

*https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/troubleshooting-kubeadm/*

**On Woker-Node:-**

* Docker change cgroup driver to systemd:-

Link:-https://stackoverflow.com/questions/43794169/docker-change-cgroup-driver-to-systemd

A solution that does not involve editing systemd units or drop-ins would be to create (or edit) the *sudo vi /etc/docker/daemon.json* configuration file and to include the following:

*{*

*"exec-opts": ["native.cgroupdriver=systemd"]*

*}*

After saving it, restart your docker service.

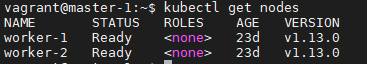
*sudo systemctl restart docker*

* We can join any number of worker nodes by running the following on each as root:

sudo kubeadm join 192.168.11.188:6443 --token 3g3k2q.g2jjxc45u1qaak13 \

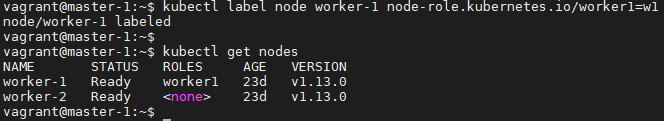
--discovery-token-ca-cert-hash sha256:4f7ba14a7e52b72dd17569b36039c211740d7982e73acbf43f6451f5a56aeb63

**How to add roles to nodes in Kubernetes?**

Default  
[[](https://i.stack.imgur.com/JpPXw.png)](https://i.stack.imgur.com/JpPXw.png)

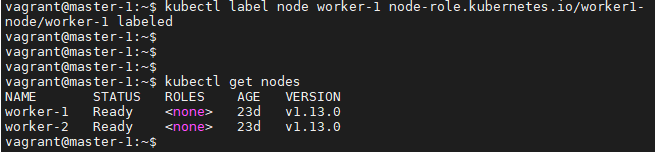
Add Role

kubectl label node <node name> node-role.kubernetes.io/<role name>=<key - (any name)>

[[](https://i.stack.imgur.com/m6Cyc.png)](https://i.stack.imgur.com/m6Cyc.png)

Remove Role

kubectl label node <node name> node-role.kubernetes.io/<role name>-

[[](https://i.stack.imgur.com/mQebY.png)](https://i.stack.imgur.com/mQebY.png)

**Deploying the Dashboard UI**

<https://www.replex.io/blog/how-to-install-access-and-add-heapster-metrics-to-the-kubernetes-dashboard>

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

<https://stackoverflow.com/questions/54081636/access-kubernetes-web-ui-from-remote-client>

Step 1:- The Dashboard UI is not deployed by default. To deploy it, run the following command:

*kubectl apply -f [https://raw.githubusercontent.com/kubernetes/dashboard/v2.2.0/aio/deploy/recommended.yaml](https://raw.githubusercontent.com/kubernetes/dashboard/v2.2.0/aio/deploy/recommended.yaml )*

Step 2:- Command line proxy

You can access Dashboard using the kubectl command-line tool by running the following command:

*kubectl proxy*

Step 3 :- Open new tab

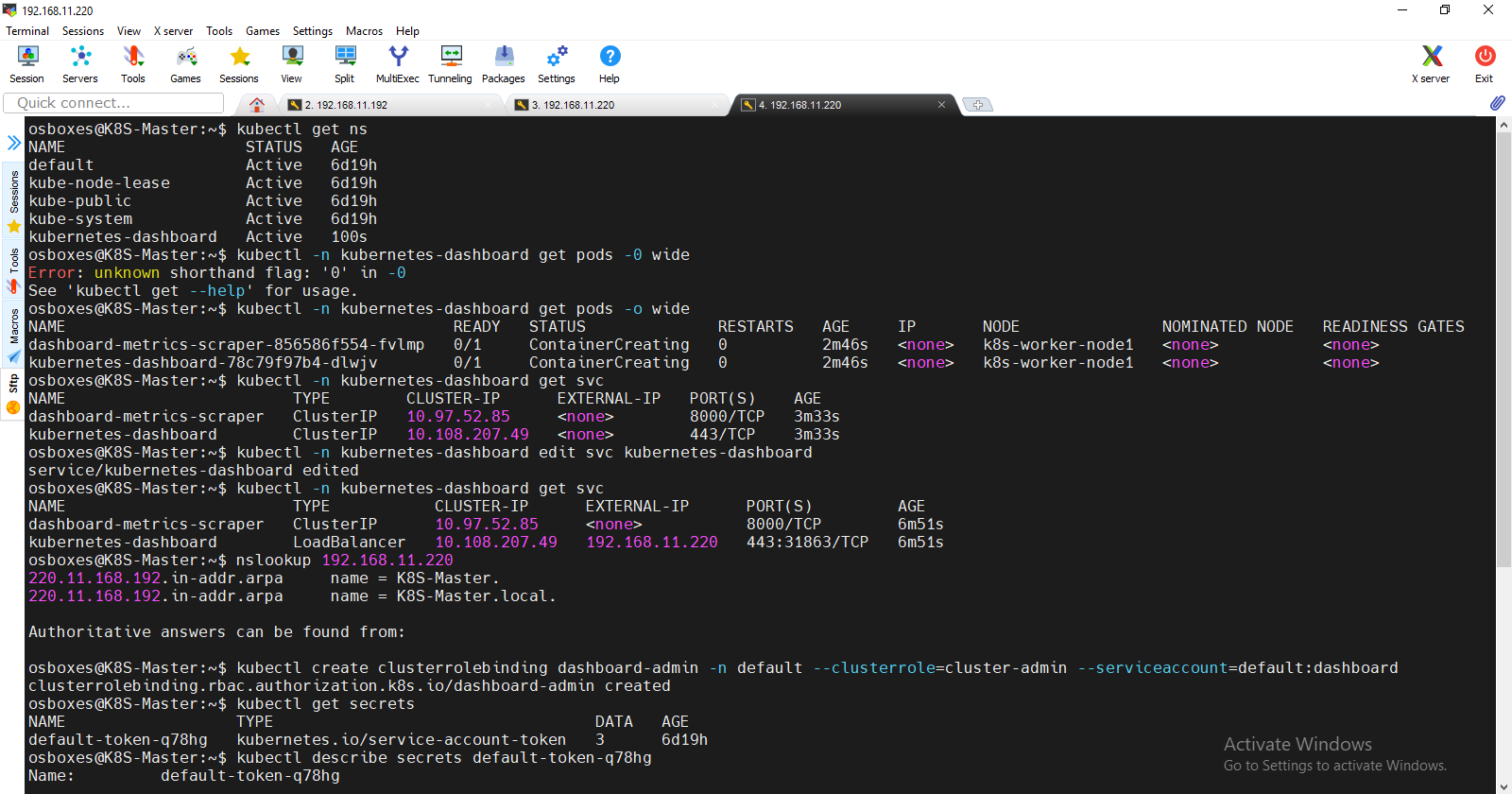
Check kubernetes Namespaces-

*kubectl get ns*

*kubectl -n kubernetes-dashboard get pods -o wide*

Step 4:- Check Kubernetes services-

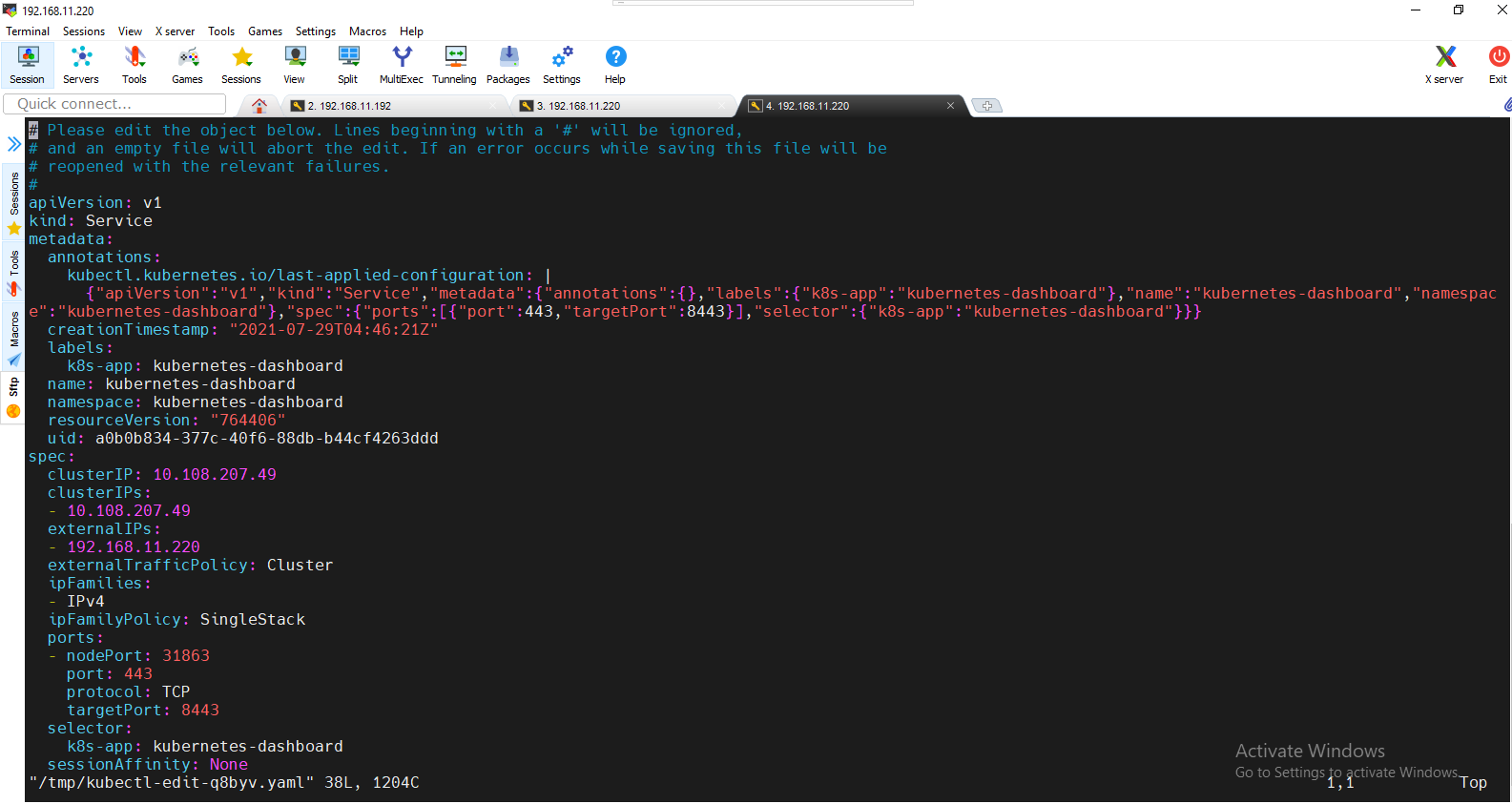
*kubectl -n kubernetes-dashboard get svc*



Step 5 :- To edit

*kubectl -n kubernetes-dashboard edit svc kubernetes-dashboard*

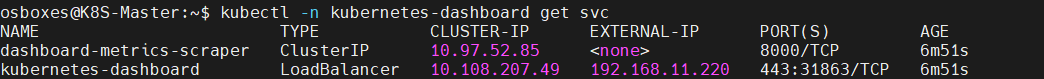
The above kubernetes-dashboard-service will work, by going to https://192.168.11.220:31863 , where 192.168.11.220 is the IP address of your Kubernetes Controller.

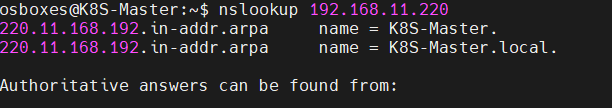


Save above configuration usin :wq!

Step 6:- Now check External IP showing in services-

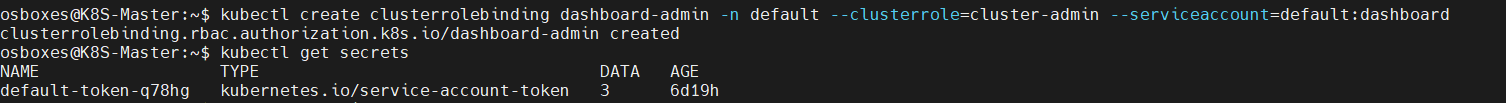
kubectl -n kubernetes-dashboard get svc





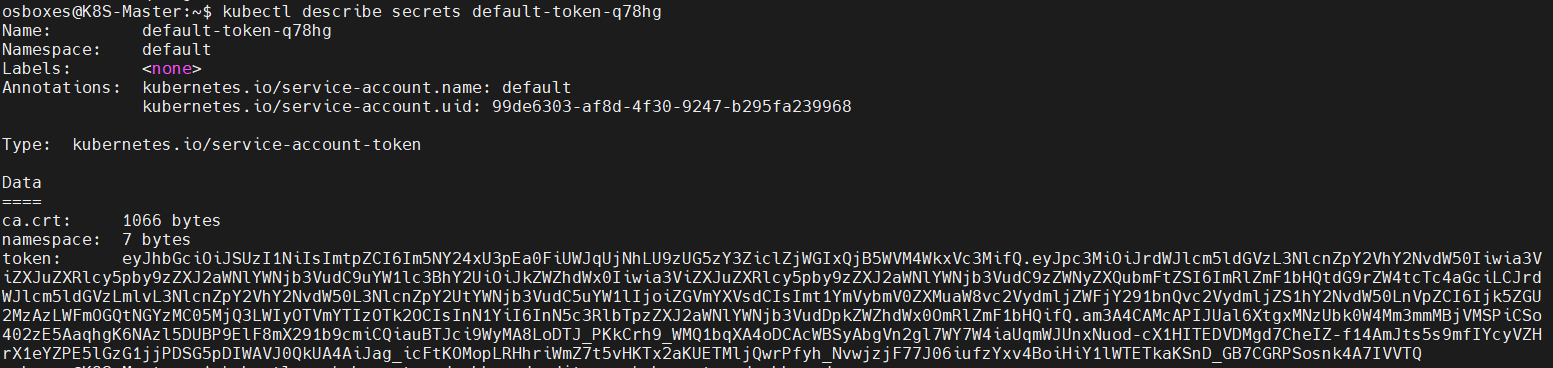
Step7:-

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



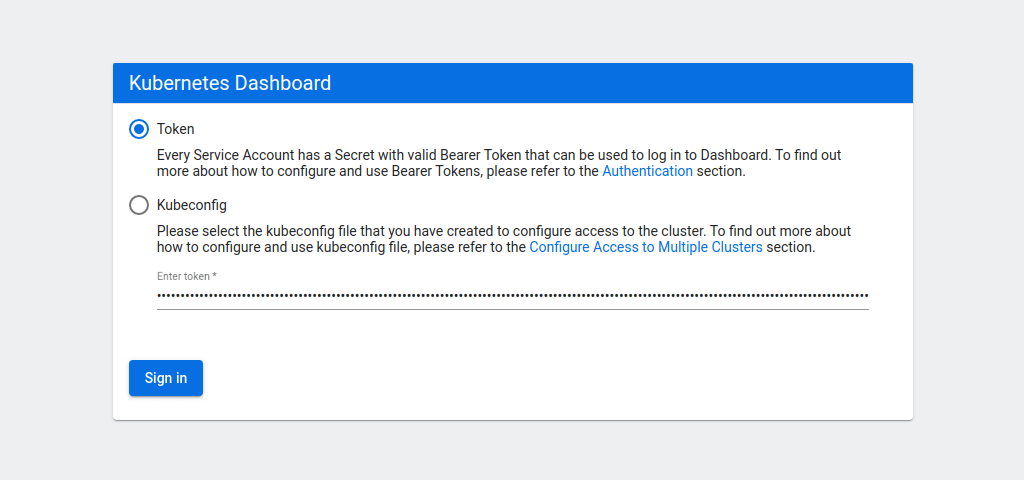
Step8:-

kubectl describe secrets default-token-q78hg



Step 9:-

Open browser and enter URL :- https://192.168.11.220



Step 10:- Enter above generated token (from step 8) and sign In

