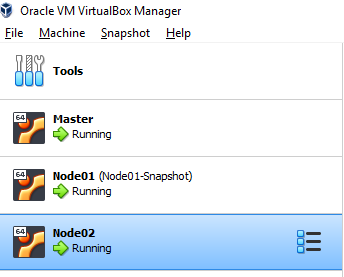
## START K8S CLUSTER

1. Start the master and worked nodes in the VM manager



1. SSH into **each** VM and execute the following command (as admin)

sudo swapoff –a

Credentials

Master: master/root

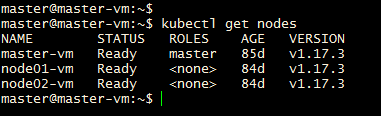
Node01: node01/root

Node02: node02/root

1. Go to master ssh session and use this command to check that the cluster is up:

kubectl get nodes

The result should “Ready” status for master node and two worker nodes.



## CONFIGURE KUBECTL ON WINDOWS VM (Developer Environment)

1. On the master node get the token of the cluster admin using the commands below. Save the token.

sudo kubectl create serviceaccount k8sadmin -n kube-system

sudo kubectl create clusterrolebinding k8sadmin --clusterrole=cluster-admin --serviceaccount=kube-system:k8sadmin

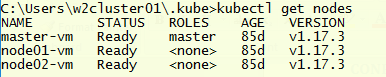
sudo kubectl -n kube-system describe secret $(sudo kubectl -n kube-system get secret | (grep k8sadmin || echo "$\_") | awk '{print $1}') | grep token: | awk '{print $2}'

1. Save the attached config file in USER\_HOME/.kube (if .kube is not created, use the window commands prompt to create one using mkdir .kube)



1. Open the config file and
   1. Replace the placeholder in clusters[0].cluster.server to IP address of master node.
   2. Replace placeholder in Users[0].user.token with the token saved in step 1.
2. Test the kubectl on windows VM now.

kubectl get nodes



## RabbitMQ

1. Clone files from https://github.com/newtonmn/kubefile.git
2. Run the rabbit MQ deployment

kubectl apply -f rabbitmq-dep-k8s.yaml

kubectl apply -f rabbitmq-dep-w-k8s.yaml

1. Run RabbitMQ k8s service

kubectl apply -f rabbitmq-svc-k8s.yaml

1. Check that RabbitMQ pods are running

kubectl get pods



1. Check that RabbitMQ service is running

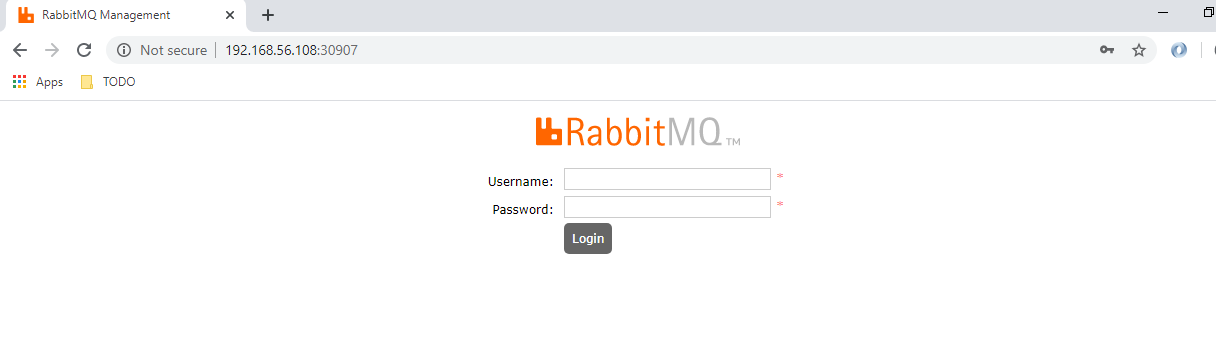
kubectl get svc



1. Since RabbitMQ service runs as NodePort service, it can be accessed externally via <node-ip:NodePort**>. Internally it can be accessed via 5672 with service name as rabbitmq-master, so rabbitmq-master:5672.**

Take node of the nodeport mapped to 15672. In this case it is 30907. So RabbitMQ console can be accessed via Node01’s IP: 30907or Node02’sIP: 30907

Accessing over Node01’s IP- <http://192.168.56.108:30907/> (Use the node ip from your VM)

s

Access using guest/guest as credentials.

## ZIPKIN

1. Clone files from https://github.com/newtonmn/kubefile.git
2. Run Zipkin Deployment

kubectl apply -f zipkin-dep-k8s.yaml

1. Run Zipkin Service

kubectl apply -f zipkin-svc-k8s.yaml

1. Check that Zipkin pods are running



1. Check that Zipkin service is running



1. Zipkin service is NodePort service. So it can be accessed externally to the NodePort mapped to 9411. Which in this case is 32619. **Internally it can accessed via service name i.e Zipkin. So internally the Zipkin URL would be Zipkin:9411.**
2. Accessing using node01’s IP(Use the node ip from your VM)

