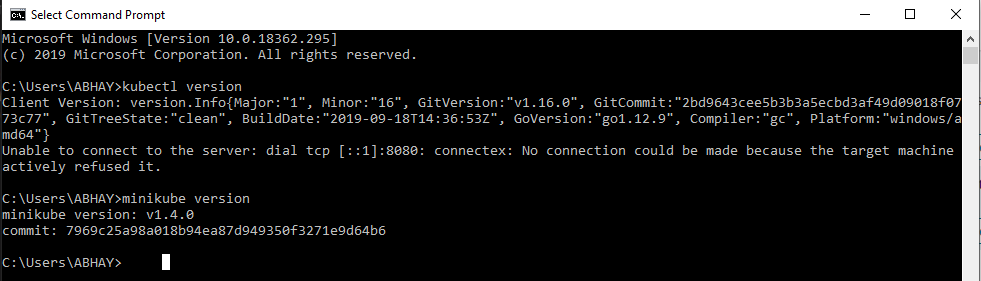
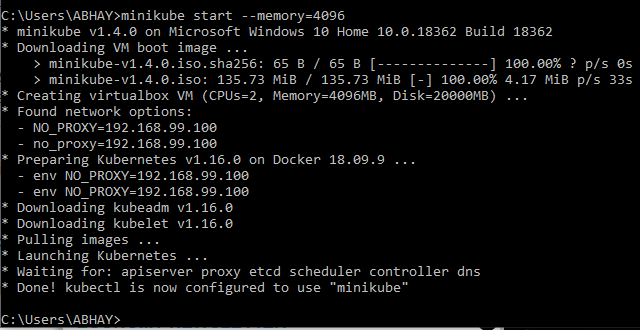
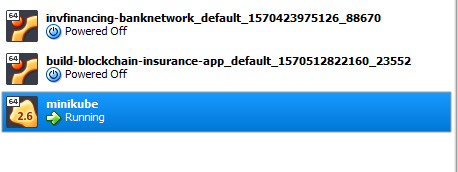
mini kube will not be installed in the project VM it'll be directly installed on your host machine.



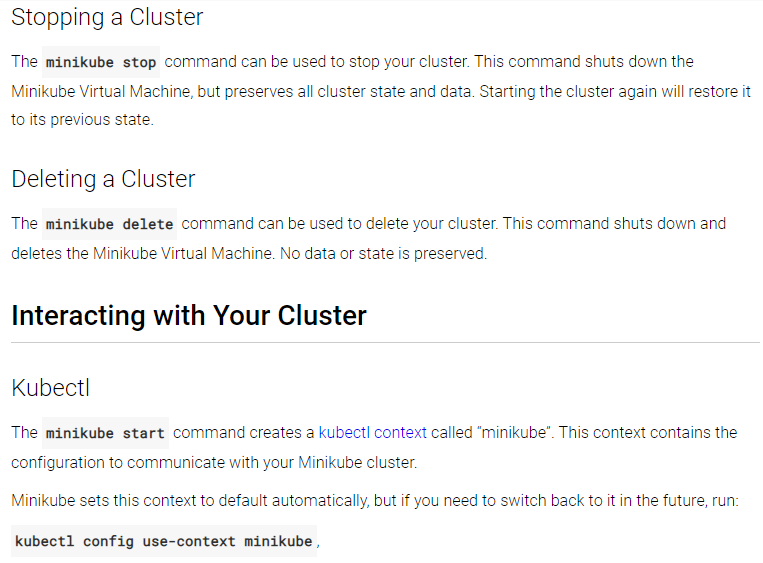


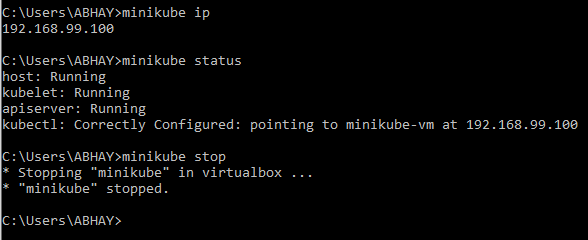




Minikube Kubernetees cluster has responded with just one service which is used internally by this Kubernetees cluster.

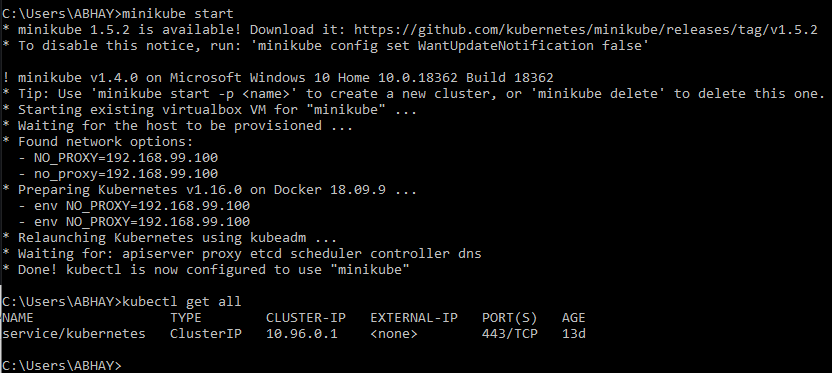
At this point you should have a mini cube and Kubectl setup on your machine.



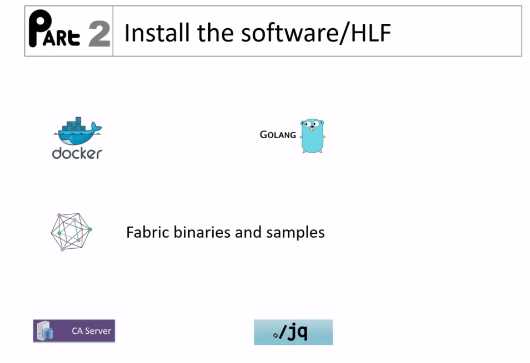


Minikube documentation -:

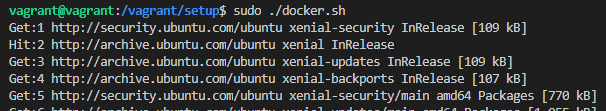
<https://kubernetes.io/docs/tutorials/hello-minikube/>



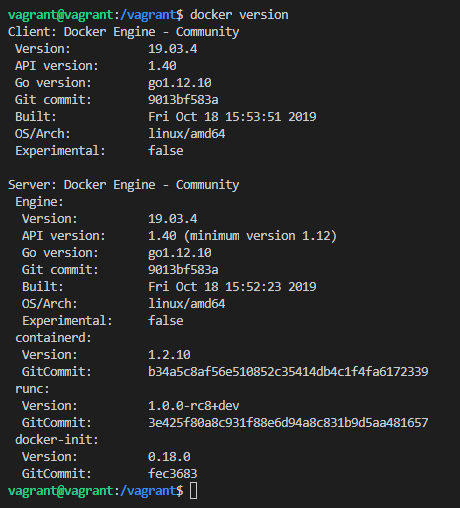
11 HLF2 – Course Project Setup

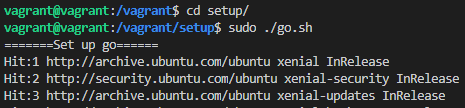


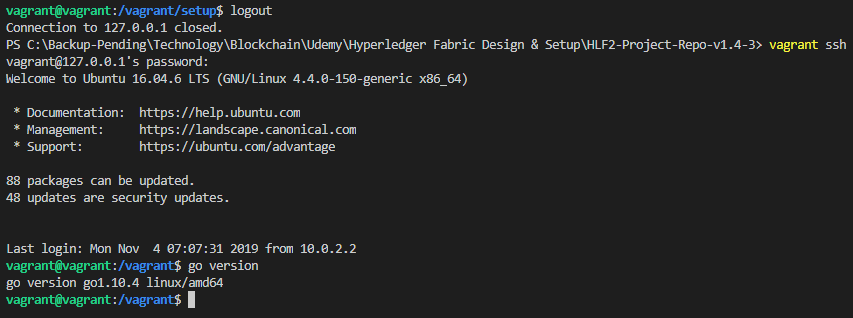




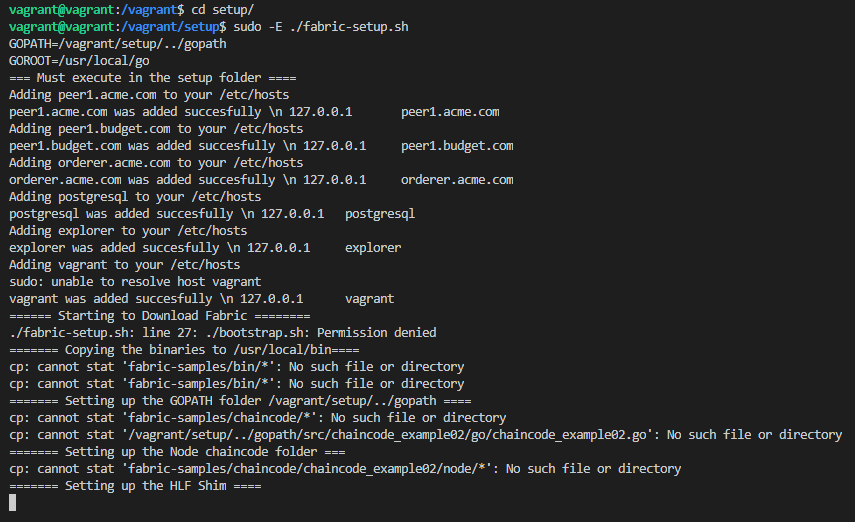


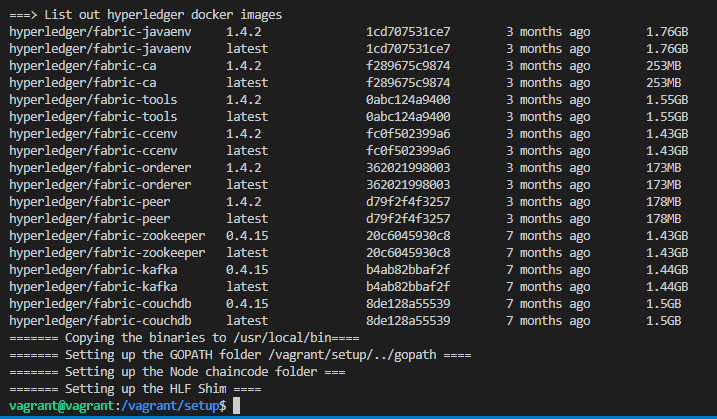


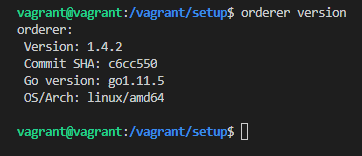




Do thorough analysis of below snapshot

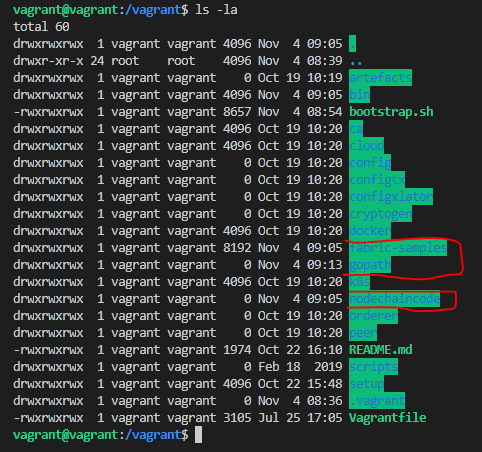


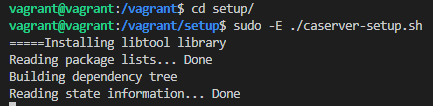


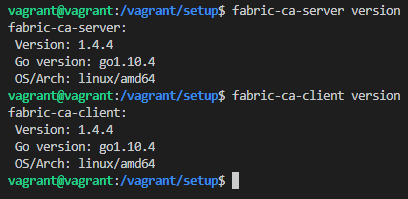


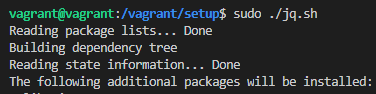
Under project root folder additional folders are created.

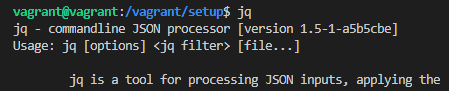
These three folders were created as part of the execution of the fabrics setup script



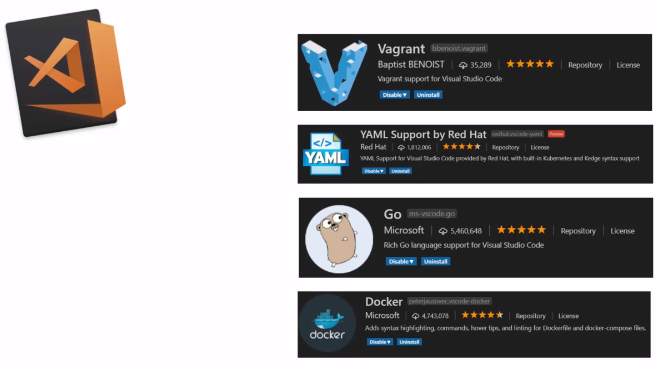




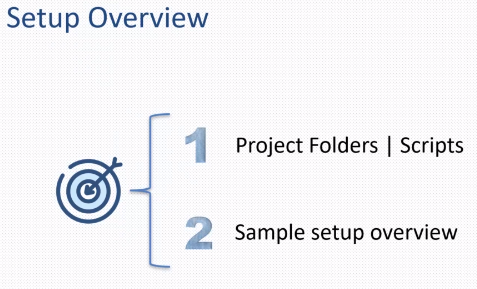




Visual studio code extensions

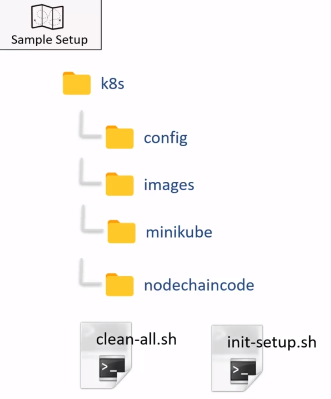


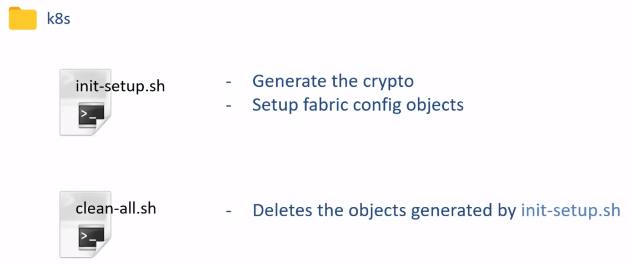
Fabric Setup on K8s Requirements



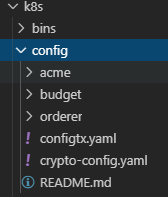
Kubernetes was introduced in version 1.4.3

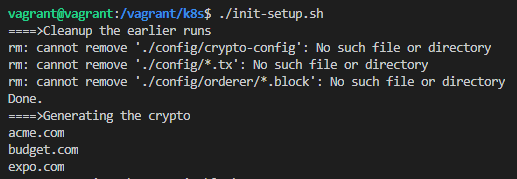
K8s is Kubernetes project root folder.

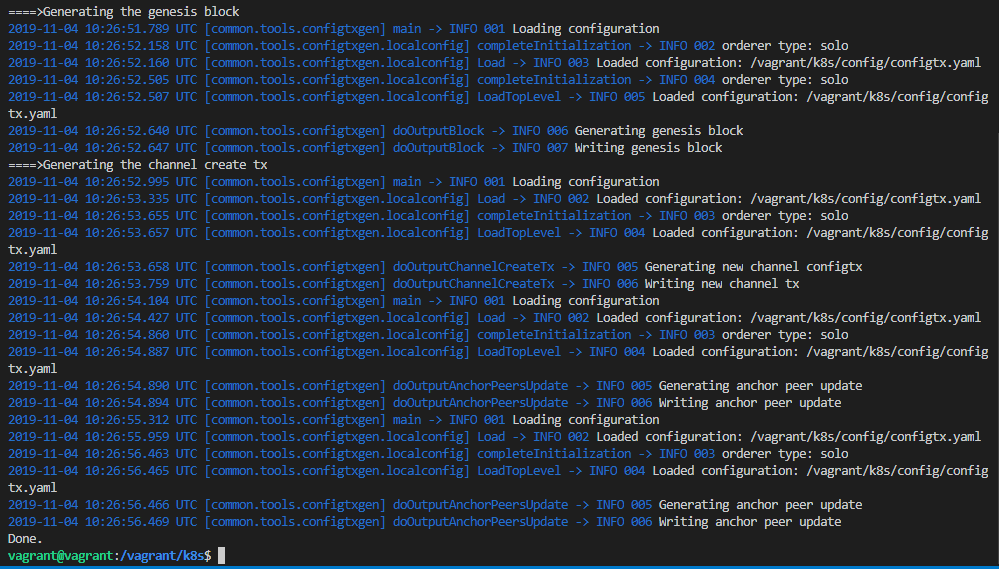




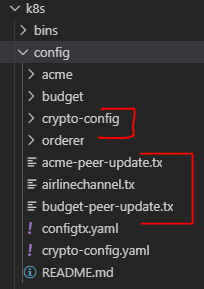
Before execution K8 subfolder



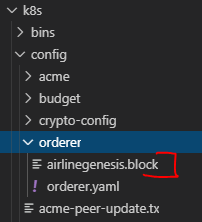


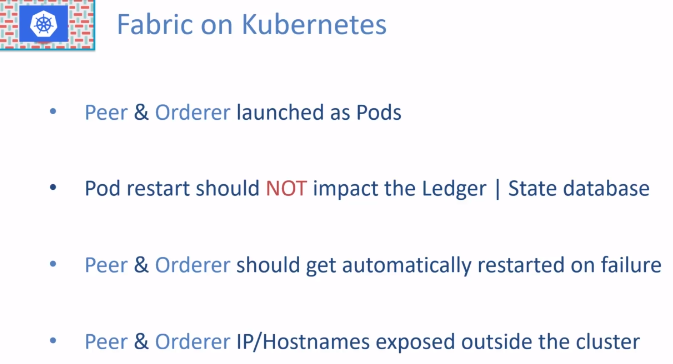


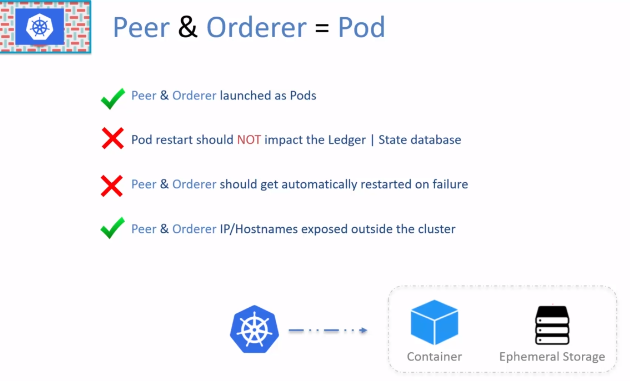
And this is going to generate the crypto config and it will also generate the various configuration objects.

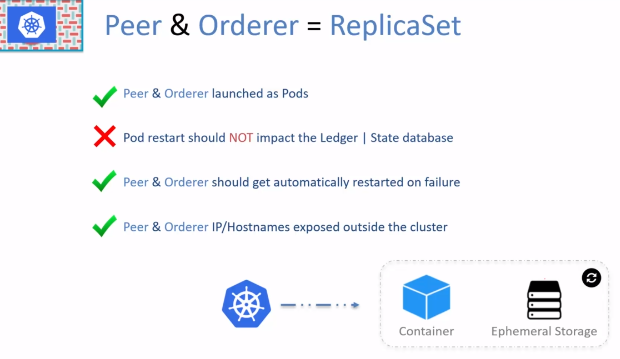


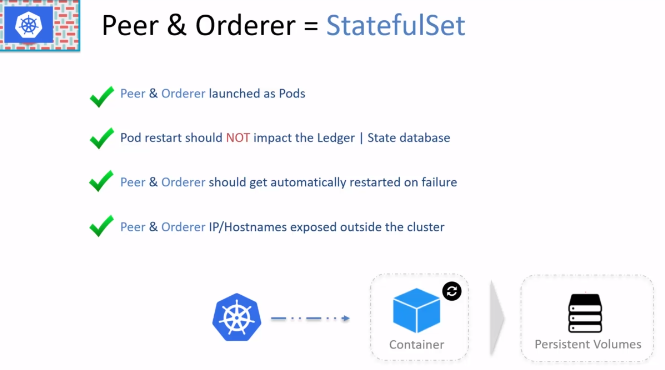
Under the orderer you will see the airline genesis block.

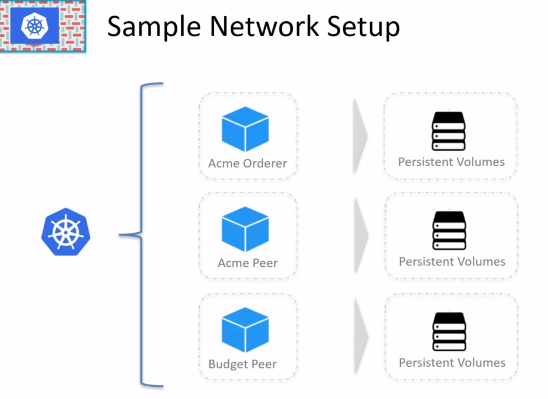


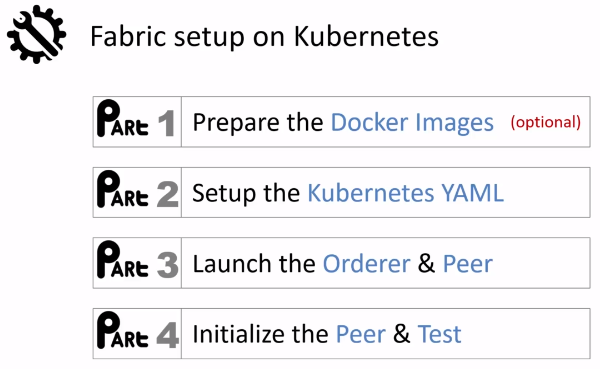


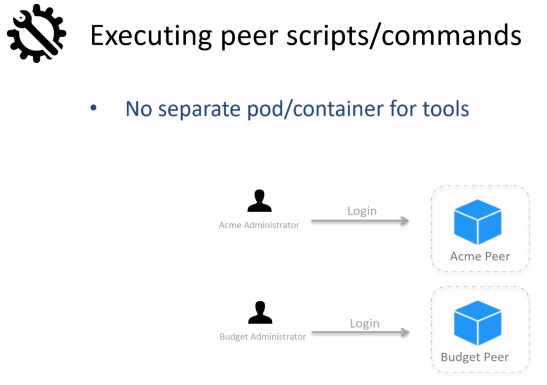








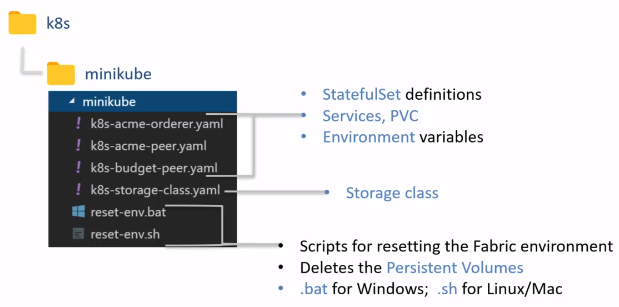


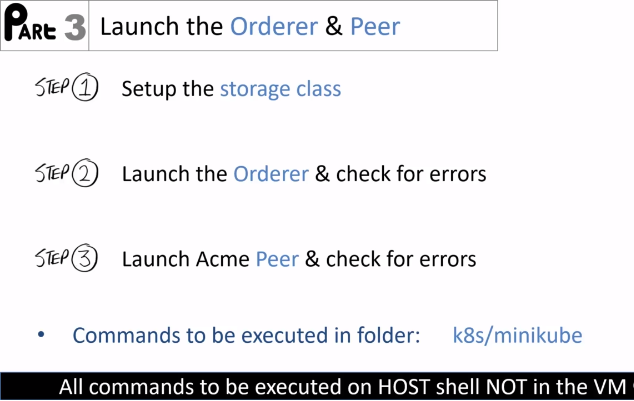




The Kubernetes YAML files are available under the subfolder miniKube.

Storage class YAML file has the storage class definition and then these 3 YAML file that has the definition for the state full set for orderer and the two peers.







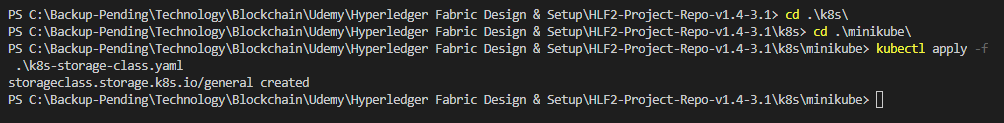
Also let's ensure that there are no pods running in the minikube environment.



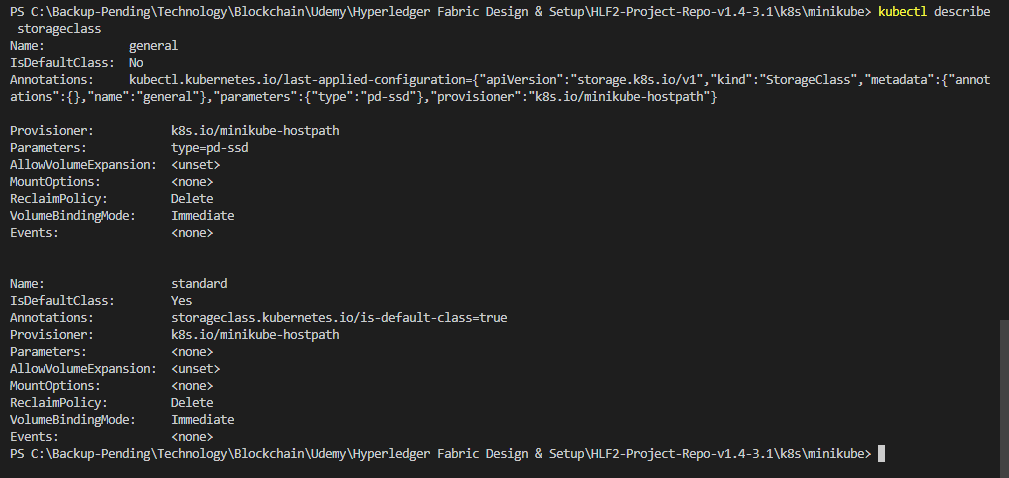
At this time please note that all of these commands are getting executed on the host machine shell. In my case windows and not on the VM

Now we will set up the storage class by executing

kubectl apply -f .\k8s-storage-class.yaml

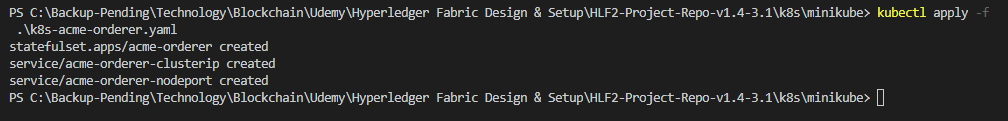


Let’s validate it

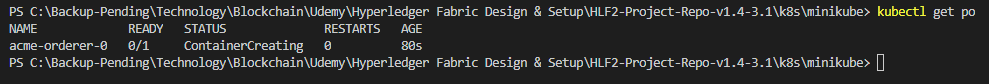


Now create the orderer pod by executing

kubectl apply -f .\k8s-acme-orderer.yaml



Check the state of the pod

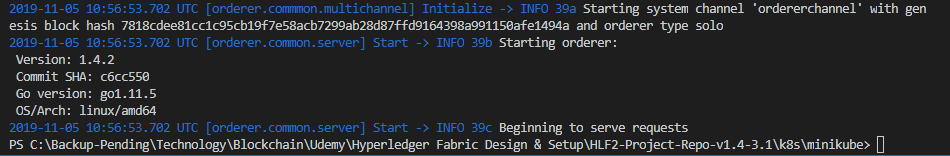


We can see that the orderer is running.

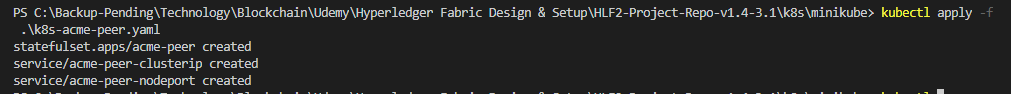
So next we are gonna check the logs in the order container.



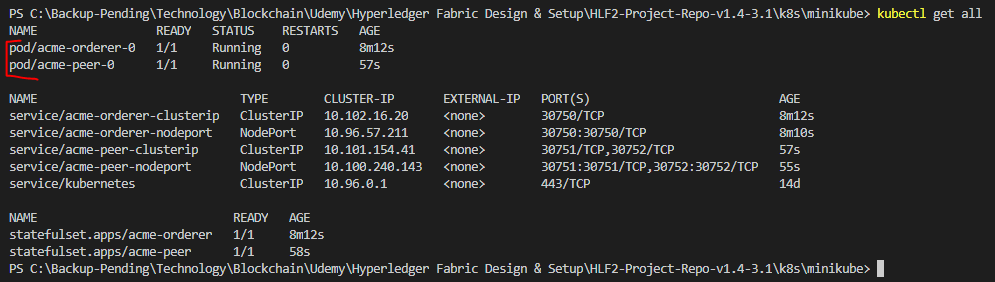
Here are the logs from the orderer



And it looks like the order is up.



Check the status, now this should show 2 pods

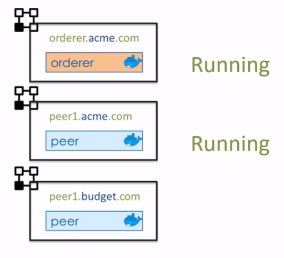


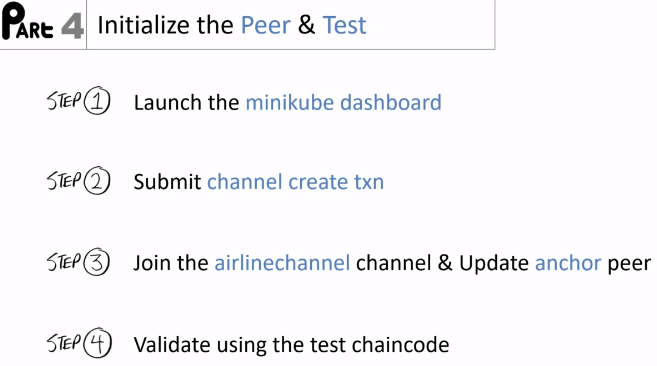
Check the logs and it doesn’t give any error



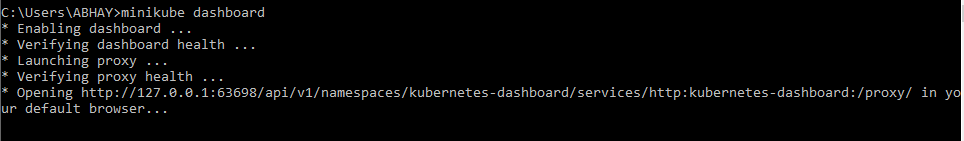
The current state of environment

The current state of our environment is that we have the Acme orderer and the Acme Peer running as pods in the minikube environment.

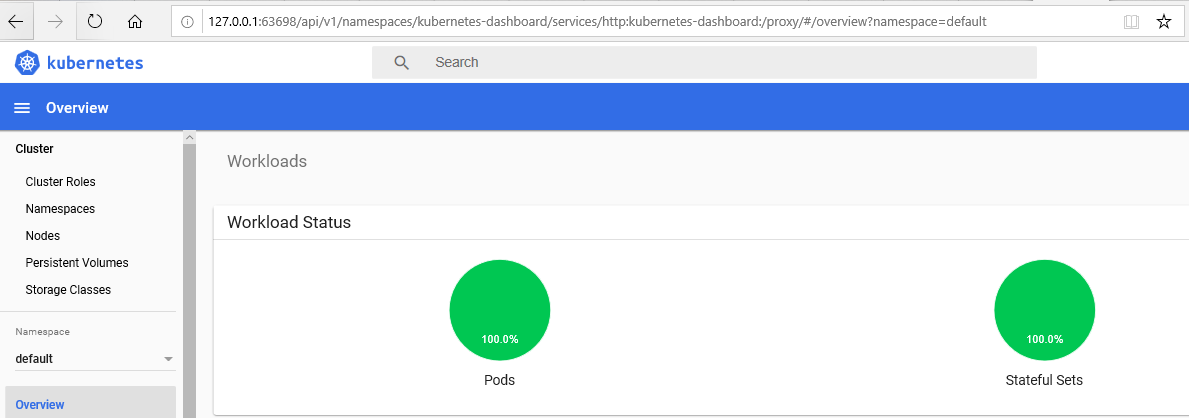




Launch a minikube dashboard by executing

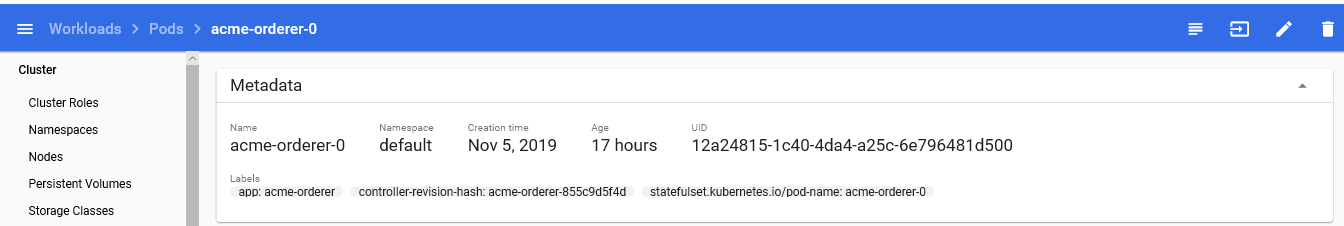


This will open up the dashboard in a browser window

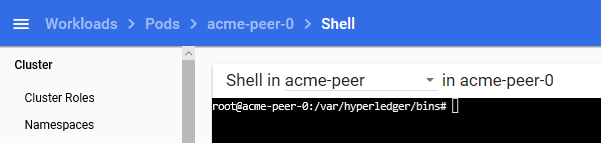


You can actually browse through the dashboard to get information about the Kubernetes cluster running.

This is the main page for the acme peer pod

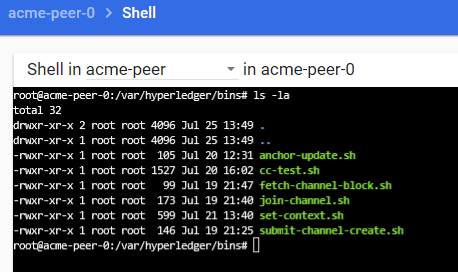


To log into the container, click on the shell icon on top right corner

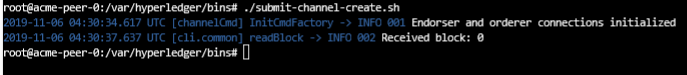


Here working directory is set to /hyperledger/bins

That has the utility shell script.

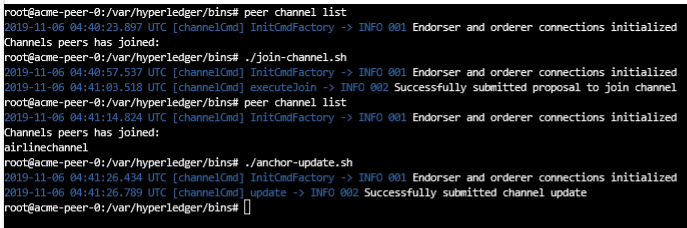


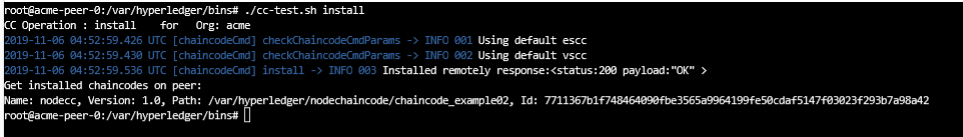
 And as the next step we have to submit the channel create transaction.

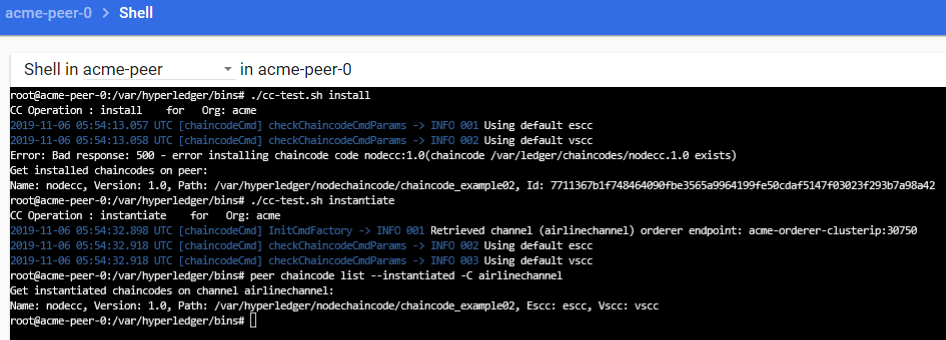


Receive block 0 message means that the channel create transaction was successfully submitted.

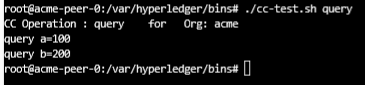
Before executing the join script. Let's just check if the peer has joined any channel.

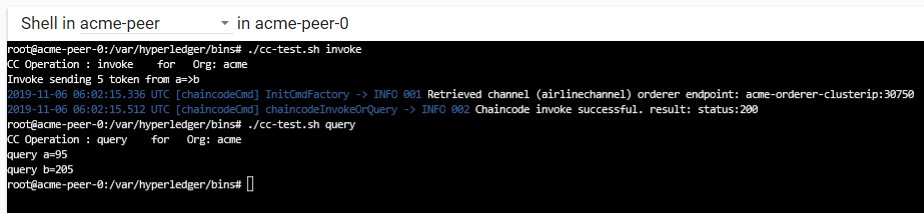


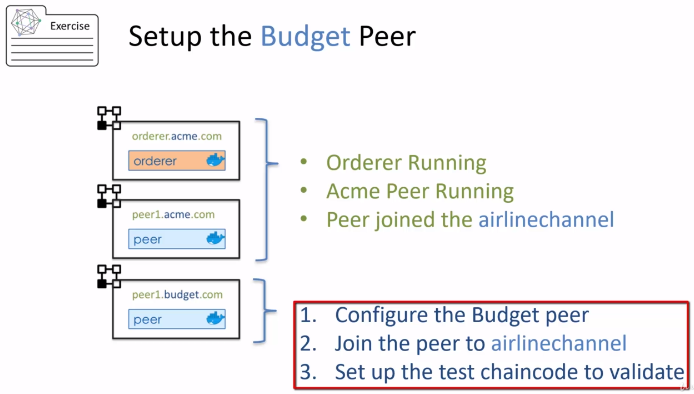


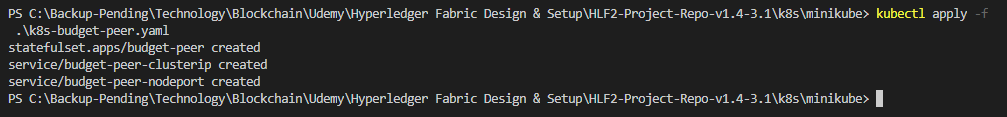


Chaincode was installed and instantiated successfully



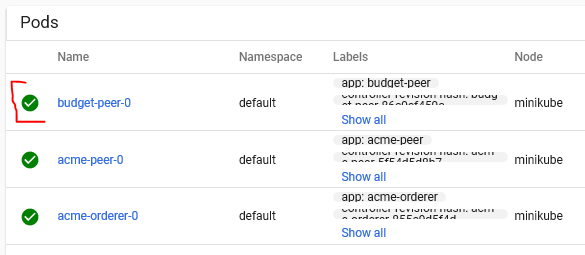






The budget peer is configured.

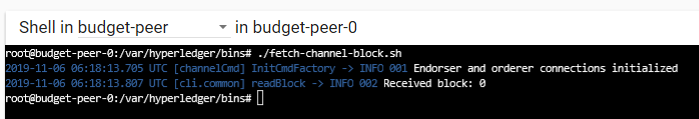
Let's go to the dashboard and refresh it.



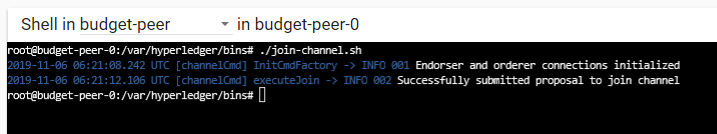
And as you can see now we have 3 pods that are active.

Log into the budget peer container, click on exec

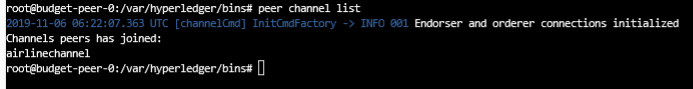
 To join the channel, first we need to fetch the genesis for the airline Channel



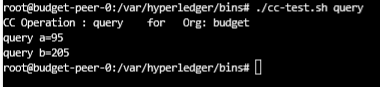
We received the block 0, and so now you can join the Channel

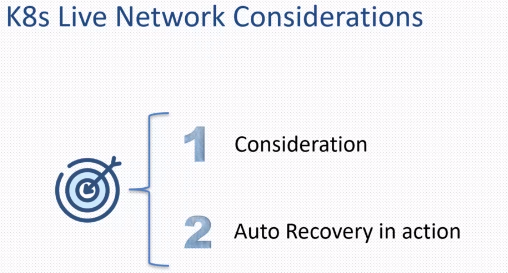


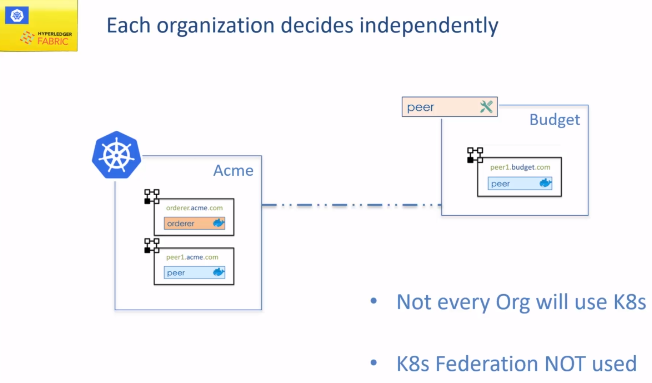
Lets make sure peer has joined the channel



At this point, we are ready to test the budget setup with the test Chaincode.







In our normal setup we created cluster IP addresses so that the peers and orders within the cluster can talk to each other.

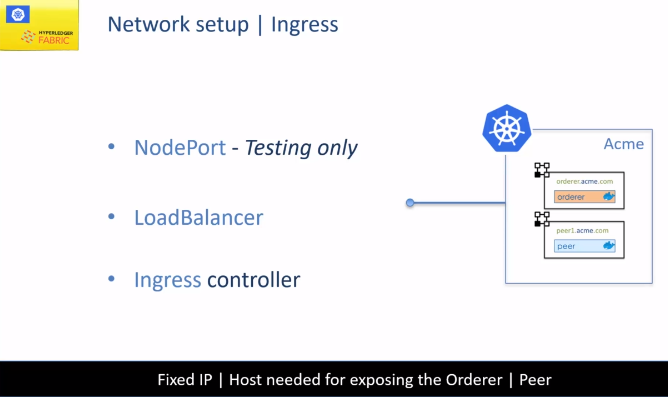
But with the cluster IP the peers and orderer pods are not exposed to the outside world.

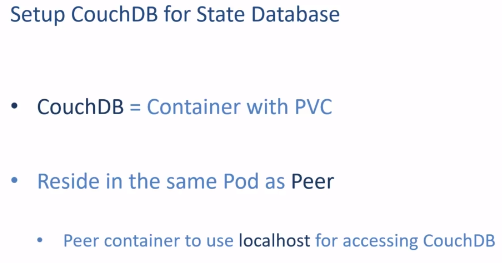
In other words a fabric network cannot be created just with cluster IP.

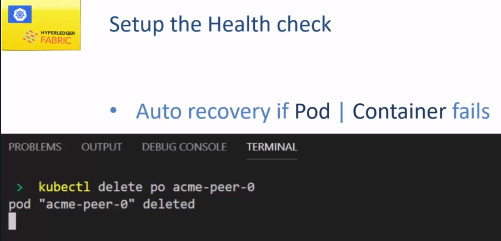
You have to expose the IP addresses for the orderer and the peer for setting of the fabric network for exposing the anchor peer and the orderer.

You can do it in three ways

1. By using NodePort service – Testing only
2. Loadbalancer – AWS, GC
3. Ingress Controller







Refresh the dashboard and you will see that the acme peer was created

