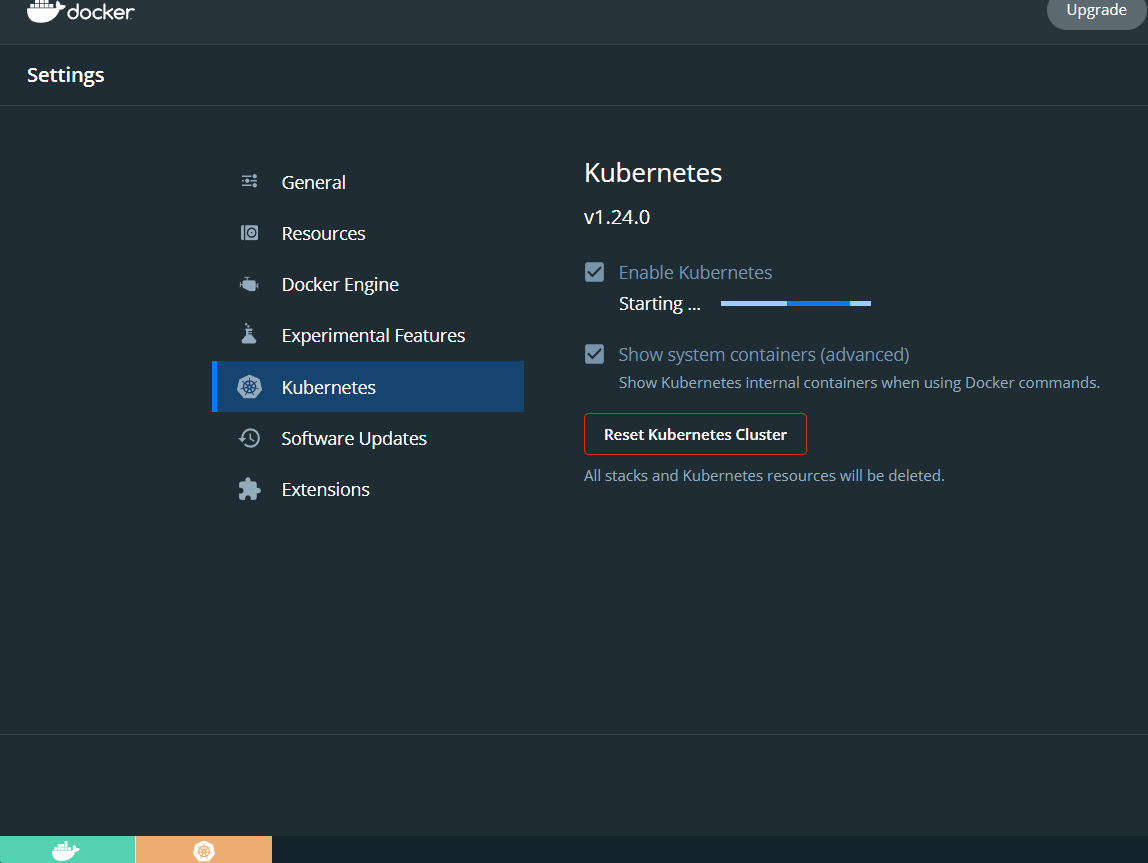
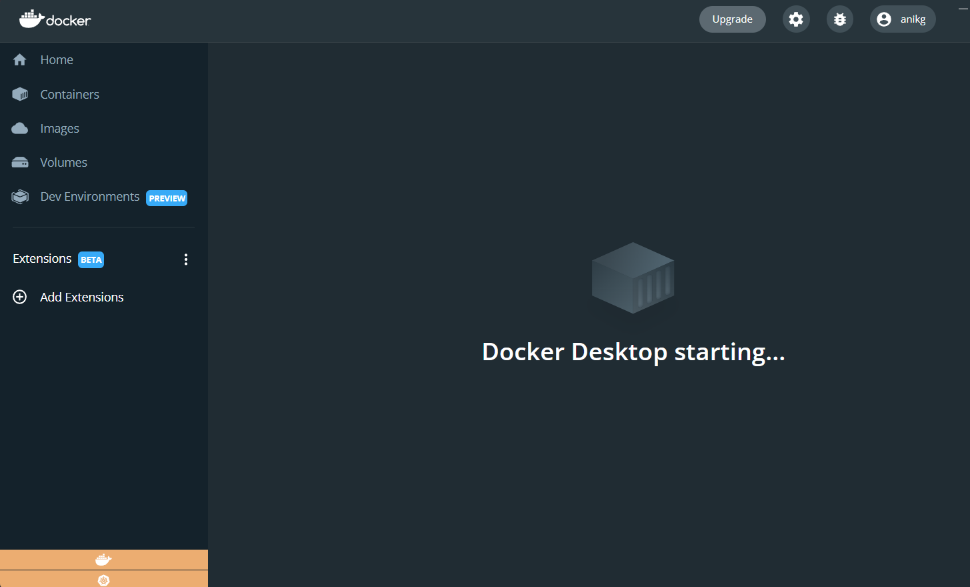
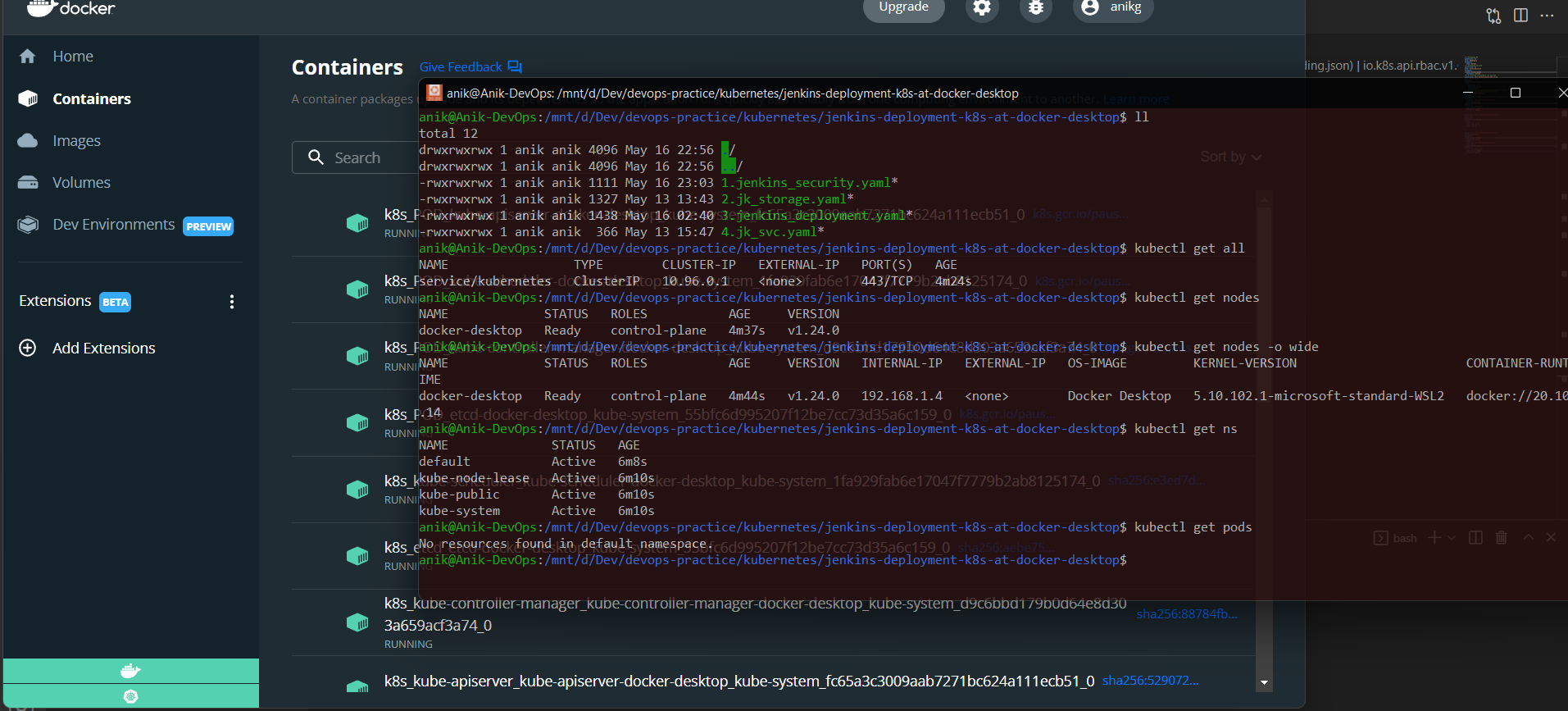
Helm Project

jenkins-deployment-k8s-at-docker-desktop

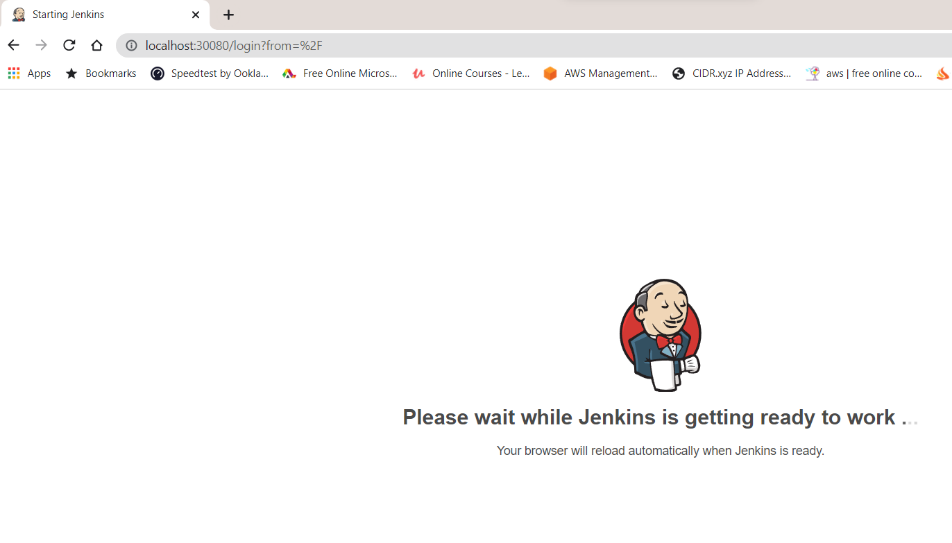
1. Starting docker desktop:
2. Make sure Kubernetes feature is enabled and stared at docker desktop & Verify



3. here deploying Jenkins as an application. Deployment must be Production standard – high available.

>>>

1. Deploying Jenkins controller at isolated **namespace**. **2**. Jenkins CICD DevOps tool: As that may need access to cluster & resources for usage, **Service account** & **RBAC** created with Clusterrole & role, role bind to Service account for Jenkins. **3**. Jenkins configuration and Jenkins data must need a storage solution for remain available outside of the pod lifecycle & backup purpose. **4**. As we are using local cluster for So here **Storage class** used local. **5**. **PersistentVolume** & **claim** maunted with Jenkins deployment volume mount. **6**. To set high availability of Jenkins application here kind used as **Deployment**. with 1 replicaset for controller node. **7**. Taking Jenkins latest **docker Image** form docker hub “ jenkins/jenkins:lts”. **8**. **Readiness** & **livenessProbe** enabled to control the health of an application. Failing liveness probe will restart the container, whereas failing readiness probe will stop our application from serving traffic. **9.** Jenkins containers listen ports 8080 & 50000. To access the 8080 port here used Kind: **Service** **NodePort** with Custom $(nodeport) from browser.

Total k8s services used:

1. Namespace 2. ServiceAccount

3. ClusterRole 4. ClusterRoleBinding

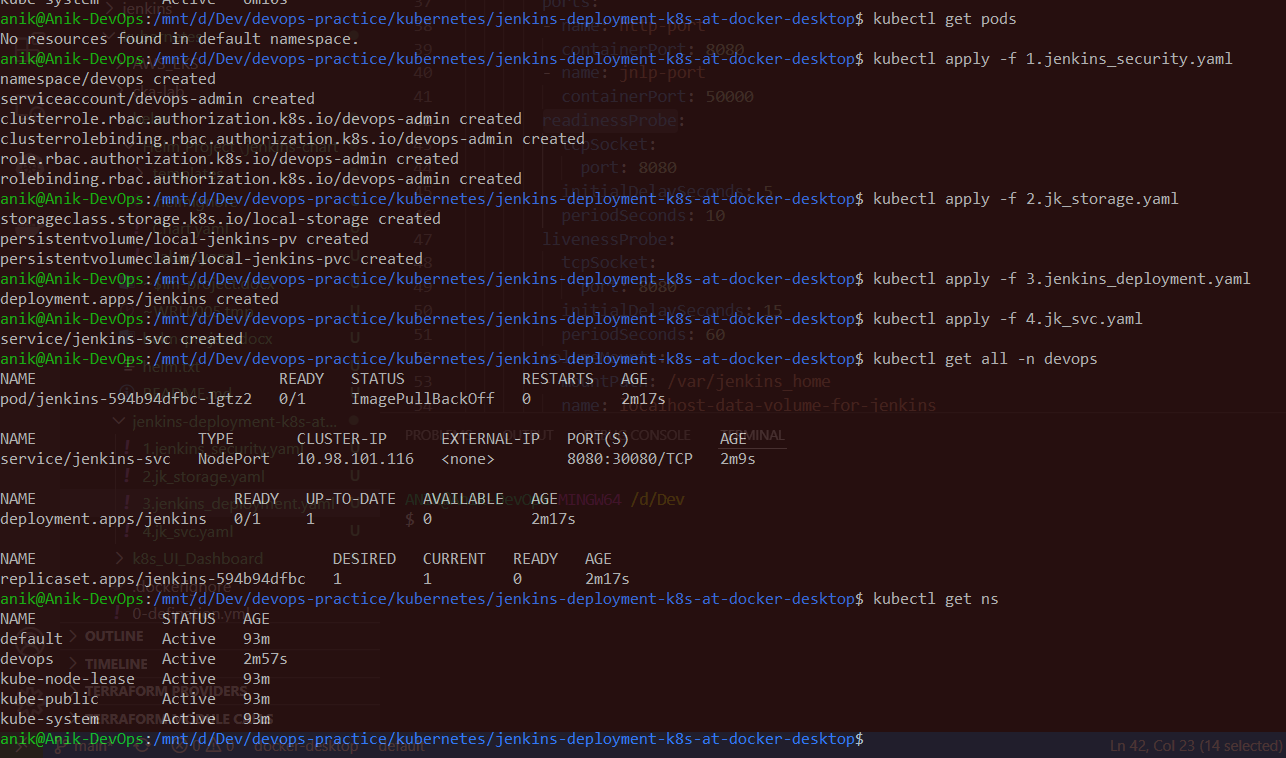
5. Role 6. RoleBinding

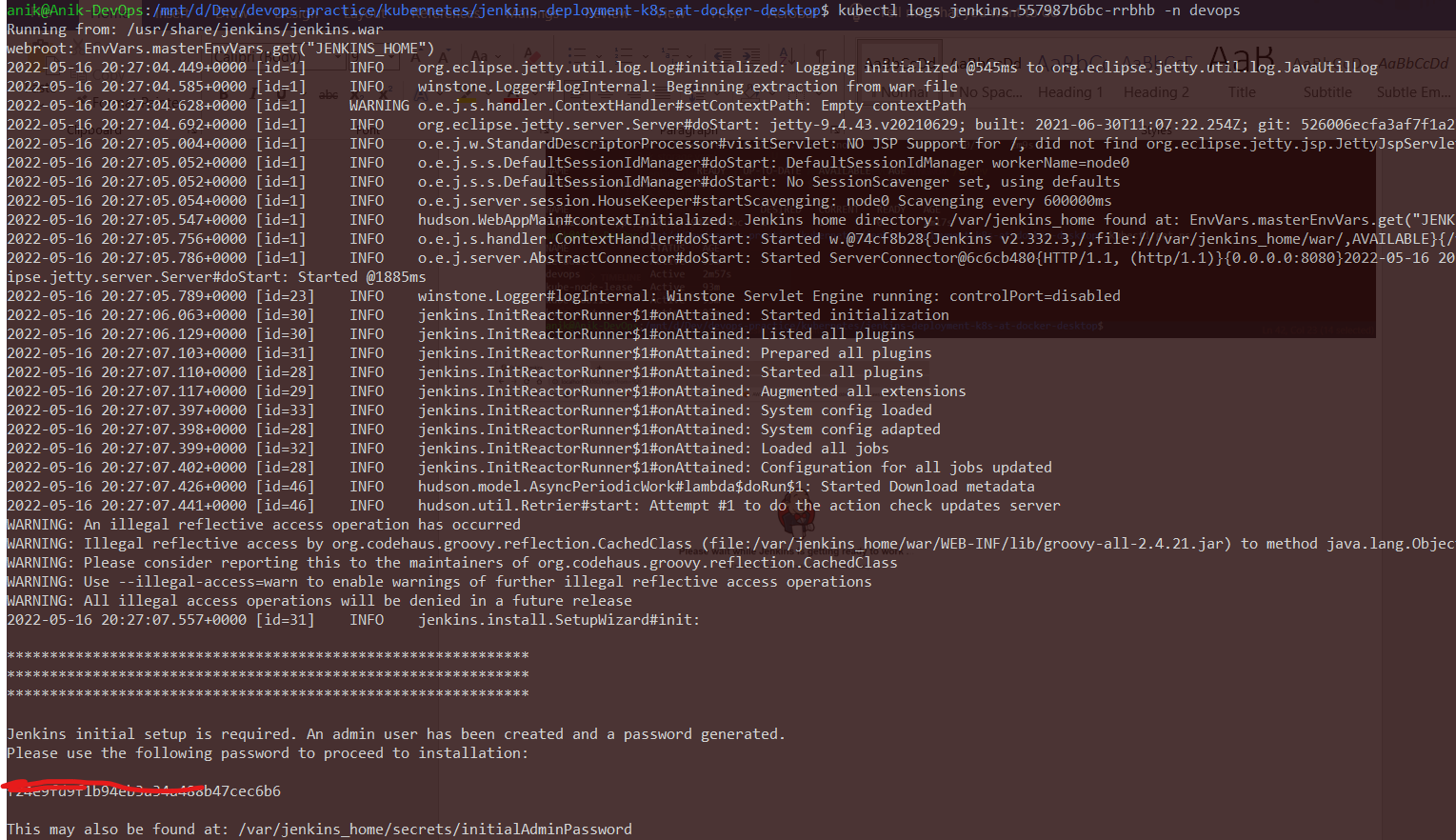
7. StorageClass 8. PersistentVolume

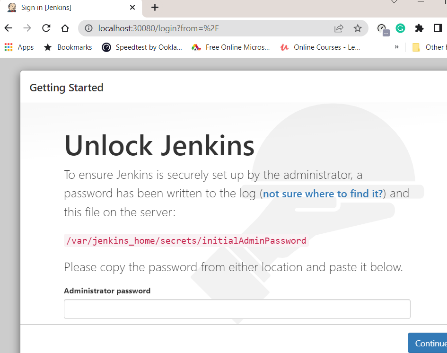
9. PersistentVolumeClaim 10. Deployment

11. Service

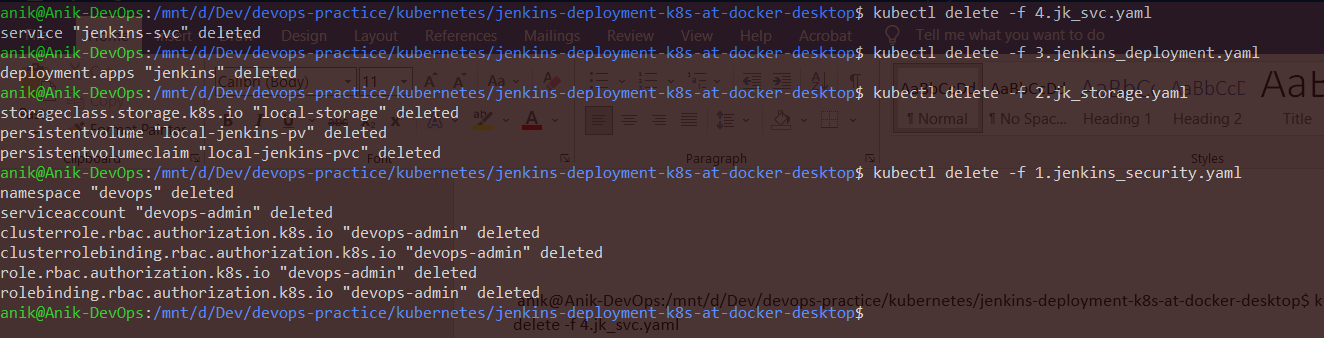
Here Code Link:

Steps:





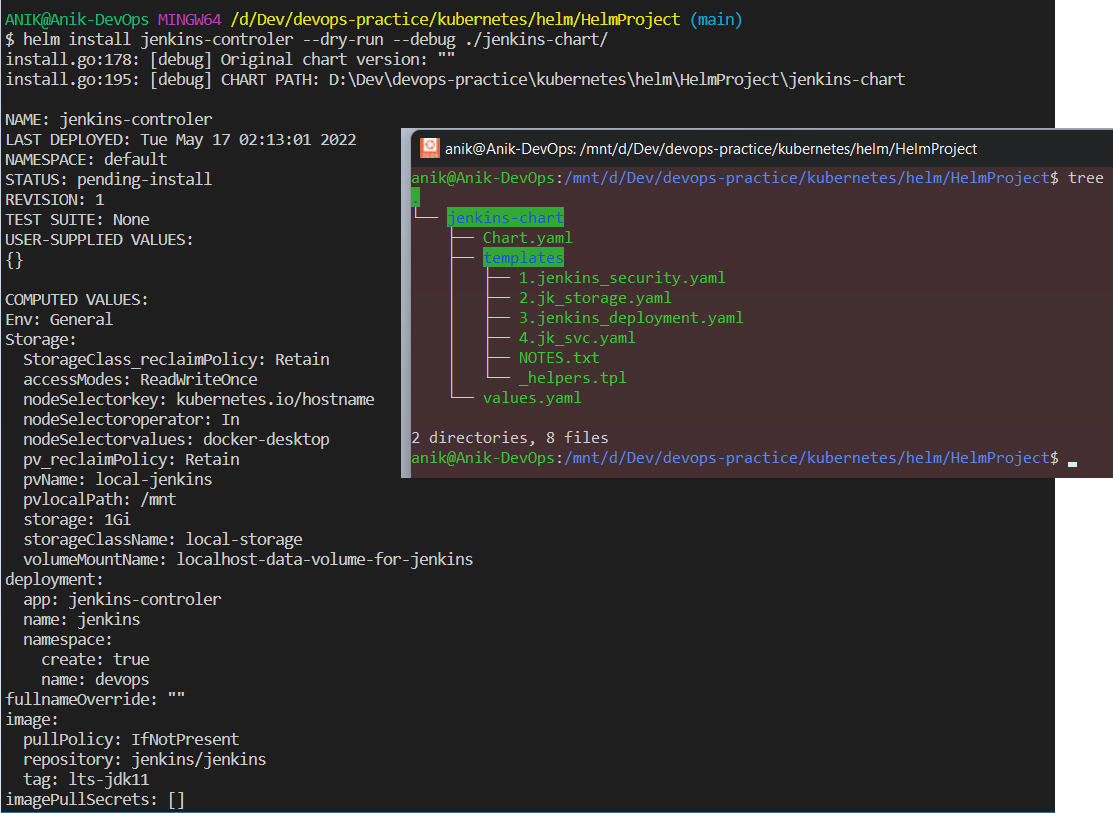
So now deployed each manifest file at a time and now destroying 1 by 1.



That’s really difficult for more complicated and huge application. So here Helm chart is the ultimate solution. You can verbalize, you can use functions, conditions, trim you repetitive codes etc.

**Helm** uses a packaging format called charts. A chart is a collection of files that describe a related set of Kubernetes resources. A single chart might be used to deploy something simple or something complex.

Code Link:

Steps: Try dry run your code to see the deployent expected outputs, : helm install <name >--dry-run --debug ./path

Then just a command **helm install jenkins-controler ./jenkins-chart** to deploy the entire application. &c helm uninstall jenkins-controler to destroy entire application.

