Question : Launch minikube cluster and execute the commands to test the pod, container with the kubectl commands.

First step we need to open URL on browser

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

After opening above link:

1. Click on Open terminal
2. Open the Kubernetes dashboard in a browser:

Need to run command >> minikube dashboard

1. Katacoda environment only: At the top of the terminal pane, click the plus sign, and then click **Select port to view on Host 1**.
2. Katacoda environment only: Type 30000, and then click Display Port.

## Open Dashboard with URL

If you don't want to open a web browser, run the dashboard command with the --url flag to emit a URL:

minikube dashboard –url

## Create a Deployment

A Kubernetes [Pod](https://kubernetes.io/docs/concepts/workloads/pods/) is a group of one or more Containers, tied together for the purposes of administration and networking. The Pod in this tutorial has only one Container. A Kubernetes [Deployment](https://kubernetes.io/docs/concepts/workloads/controllers/deployment/) checks on the health of your Pod and restarts the Pod's Container if it terminates. Deployments are the recommended way to manage the creation and scaling of Pods.

Katacoda environment only: At the top of the terminal pane, click the plus sign, and then click Open New Terminal.

Use the **kubectl create** command to create a Deployment that manages a Pod. The Pod runs a Container based on the provided Docker image.

Need to run command >> kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.39 -- /agnhost netexec --http-port=8080

View the Deployment: Need to run command >> kubectl get deployments

View the Pod: Need to run command >> kubectl get pods

View cluster events: Need to run command >> kubectl get events

View the kubectl configuration: Need to run command >> kubectl config view

**Note:** For more information about kubectl commands, see the [kubectl overview](https://kubernetes.io/docs/reference/kubectl/).

**Create a Service**

By default, the Pod is only accessible by its internal IP address within the Kubernetes cluster. To make the hello-node Container accessible from outside the Kubernetes virtual network, you have to expose the Pod as a Kubernetes [*Service*](https://kubernetes.io/docs/concepts/services-networking/service/).

1. Expose the Pod to the public internet using the kubectl expose command:

Need to run command >> kubectl expose deployment hello-node --type=LoadBalancer --port=8080

1. View the Service you created: Need to run command >> kubectl get services
2. Run the following command: Need to run command >> minikube service hello-node

Enable addons

The minikube tool includes a set of built-in [addons](https://kubernetes.io/docs/concepts/cluster-administration/addons/) that can be enabled, disabled and opened in the local Kubernetes environment.

1. List the currently supported addons: Need to run command >> minikube addons list

Output display >>The output is similar to:

addon-manager: enabled

dashboard: enabled

default-storageclass: enabled

efk: disabled

freshpod: disabled

gvisor: disabled

helm-tiller: disabled

ingress: disabled

ingress-dns: disabled

logviewer: disabled

metrics-server: disabled

nvidia-driver-installer: disabled

nvidia-gpu-device-plugin: disabled

registry: disabled

registry-creds: disabled

storage-provisioner: enabled

storage-provisioner-gluster: disabled

1. Enable an addon, for example, metrics-server: Need to run command >> minikube addons enable metrics-server

OUTPUT >> The output is similar to:The 'metrics-server' addon is enabled

1. View the Pod and Service you created: Need to run command >> kubectl get pod,svc -n kube-system
2. Disable metrics-server: Need to run command >> minikube addons disable metrics-server

## Clean up

Now you can clean up the resources you created in your cluster:

kubectl delete service hello-node

kubectl delete deployment hello-node

Optionally, stop the Minikube virtual

minikube stop

Optionally, delete the Minikube VM:

minikube delete