

## Module-3: Cloud Testing Hands-on

### Hands-on Assignments : (any 3 mandatory)

1. Create a VM and install docker and test the images inside the container with the commands
2. Install ansible and connect the nodes to master and execute the adhoc commands on ansible masternode vm and test the packages installed or not
3. Launch minikube cluster and execute the commands to test the pod, container with the kubectl commands
4. Create the poc on open shift architecture and its components and commands used on cluster

### SOLUTION:

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

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

Before you begin

This tutorial provides a container image that uses NGINX to echo back all the requests.

### Create a minikube cluster

1. Click **Launch Terminal**

**Note:** If you installed minikube locally, run `minikube start`. Before you run `minikube dashboard`, you should open a new terminal, start `minikube dashboard` there, and then switch back to the main terminal.

2. Open the Kubernetes dashboard in a browser:

minikube dashboard

Click Launch Terminal & check minikube version

```
Terminal Preview Port 30000 +
* minikube v1.18.0 on Ubuntu 18.04 (amd64)
* Using the none driver based on existing profile

X The requested memory allocation of 2200MiB does not leave room for system overhead (total system memory: 2460MiB). You may face stability issues.
* Suggestion: Start minikube with less memory allocated: 'minikube start --memory=2200mb'

* Starting control plane node minikube in cluster minikube
* Running on localhost (CPUs=2, Memory=2460MB, Disk=194868MB) ...
* OS release is Ubuntu 18.04.5 LTS
* Preparing Kubernetes v1.20.2 on Docker 19.03.13 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring local host environment ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v4
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
  - Using image k8s.gcr.io/metrics-server-amd64:v0.2.1
* The 'metrics-server' addon is enabled
  - Using image kubernetesui/dashboard:v2.1.0
  - Using image kubernetesui/metrics-scraper:v1.0.4
* Some dashboard features require the metrics-server addon. To enable all features please run:

  minikube addons enable metrics-server

* The 'dashboard' addon is enabled
Kubernetes Started
$
$ minikube version

Command 'minikube' not found, did you mean:

  command 'minitube' from deb minitube

Try: apt install <deb name>

$ minikube version
minikube version: v1.18.0
commit: ec61815d60f66a6e4f6353030a40b12362557caa-dirty
$
```

## Start a cluster

```
Terminal Preview Port 30000 +
* The 'metrics-server' addon is enabled
  - Using image kubernetesui/dashboard:v2.1.0
  - Using image kubernetesui/metrics-scraper:v1.0.4
* Some dashboard features require the metrics-server addon. To enable all features please run:

    minikube addons enable metrics-server

* The 'dashboard' addon is enabled
Kubernetes Started
$
$ minikube version

Command 'minikube' not found, did you mean:

  command 'minitube' from deb minitube

Try: apt install <deb name>

$ minikube version
minikube version: v1.18.0
commit: ec61815d60f66a6e4f6353030a40b12362557caa-dirty
$ minikube start
* minikube v1.18.0 on Ubuntu 18.04 (amd64)
* Using the none driver based on existing profile

X The requested memory allocation of 2200MiB does not leave room for system overhead (total system memory: 2460MiB). You may face stability issues.
* Suggestion: Start minikube with less memory allocated: 'minikube start --memory=2200mb'

* Starting control plane node minikube in cluster minikube
* Updating the running none "minikube" bare metal machine ...
* OS release is Ubuntu 18.04.5 LTS
* Preparing Kubernetes v1.20.2 on Docker 19.03.13 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Configuring local host environment ...
* Verifying Kubernetes components...
  - Using image kubernetesui/dashboard:v2.1.0
  - Using image kubernetesui/metrics-scraper:v1.0.4
  - Using image k8s.gcr.io/metrics-server-amd64:v0.2.1
  - Using image gcr.io/k8s-minikube/storage-provisioner:v4
* Enabled addons: storage-provisioner, metrics-server, default-storageclass, dashboard
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
$
```

## Check kubectl version

```
Terminal Preview Port 30000 +
minikube addons enable metrics-server

* The 'dashboard' addon is enabled
Kubernetes Started
$
$ minikube version

Command 'minikube' not found, did you mean:

  command 'minitube' from deb minitube

Try: apt install <deb name>

$ minikube version
minikube version: v1.18.0
commit: ec61815d60f66a6e4f6353030a40b12362557caa-dirty
$ minikube start
* minikube v1.18.0 on Ubuntu 18.04 (amd64)
* Using the none driver based on existing profile

X The requested memory allocation of 2200MiB does not leave room for system overhead (total system memory: 2460MiB). You may face stability issues.
* Suggestion: Start minikube with less memory allocated: 'minikube start --memory=2200mb'

* Starting control plane node minikube in cluster minikube
* Updating the running none "minikube" bare metal machine ...
* OS release is Ubuntu 18.04.5 LTS
* Preparing Kubernetes v1.20.2 on Docker 19.03.13 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Configuring local host environment ...
* Verifying Kubernetes components...
  - Using image kubernetesui/dashboard:v2.1.0
  - Using image kubernetesui/metrics-scraper:v1.0.4
  - Using image k8s.gcr.io/metrics-server-amd64:v0.2.1
  - Using image gcr.io/k8s-minikube/storage-provisioner:v4
* Enabled addons: storage-provisioner, metrics-server, default-storageclass, dashboard
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
$ kubectl version
Client Version: version.Info{Major:"1", Minor:"20", GitVersion:"v1.20.4", GitCommit:"c87da0bd6e03ec3fea7933c4b5263d15laafd07c", GitTreeState:"clean", BuildDate:"2021-02-18T16:12:00Z", GoVersion:"go1.15.8", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"20", GitVersion:"v1.20.2", GitCommit:"faecb196815e248d3ecfb03c680a4507229c2a56", GitTreeState:"clean", BuildDate:"2021-01-13T13:20:00Z", GoVersion:"go1.15.5", Compiler:"gc", Platform:"linux/amd64"}
$
```

## Cluster info

```
Terminal Preview Port 30000 +
Try: apt install <deb name>

$ minikube version
minikube version: v1.18.0
commit: ec61815d60f66a6e4f6353030a40b12362557caa-dirty
$ minikube start
* minikube v1.18.0 on Ubuntu 18.04 (amd64)
* Using the none driver based on existing profile

X The requested memory allocation of 2200MiB does not leave room for system overhead (total system memory: 2460MiB). You may face stability issues.
* Suggestion: Start minikube with less memory allocated: 'minikube start --memory=2200mb'

* Starting control plane node minikube in cluster minikube
* Updating the running none "minikube" bare metal machine ...
* OS release is Ubuntu 18.04.5 LTS
* Preparing Kubernetes v1.20.2 on Docker 19.03.13 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Configuring local host environment ...
* Verifying Kubernetes components...
  - Using image kubernetesui/dashboard:v2.1.0
  - Using image kubernetesui/metrics-scrapers:v1.0.4
  - Using image k8s.gcr.io/metrics-server-amd64:v0.2.1
  - Using image gcr.io/k8s-minikube/storage-provisioner:v4
* Enabled addons: storage-provisioner, metrics-server, default-storageclass, dashboard
* Done! kubect1 is now configured to use "minikube" cluster and "default" namespace by default
$ kubect1 version
Client Version: version.Info{Major:"1", Minor:"20", GitVersion:"v1.20.4", GitCommit:"e87da0bd6e03fea7933c4b5263d151aafd07c", GitTreeState:"clean", BuildDate:"2021-02-18T16:12:00Z", GoVersion:"go1.15.8", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"20", GitVersion:"v1.20.2", GitCommit:"faecb196815e248d3ecfb03c680a4507229c2a56", GitTreeState:"clean", BuildDate:"2021-01-13T13:20:00Z", GoVersion:"go1.15.5", Compiler:"gc", Platform:"linux/amd64"}
$ kubect1 cluster info
Error: unknown command "cluster" for "kubect1"

Did you mean this?
    cluster-info

Run 'kubect1 --help' for usage.
$ kubect1 cluster-info
Kubernetes control plane is running at https://10.0.0.9:8443
KubeDNS is running at https://10.0.0.9:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubect1 cluster-info dump'.
$
```

## Get nodes

```
Terminal Preview Port 30000 +
minikube version: v1.18.0
commit: ec61815d60f66a6e4f6353030a40b12362557caa-dirty
$ minikube start
* minikube v1.18.0 on Ubuntu 18.04 (amd64)
* Using the none driver based on existing profile

X The requested memory allocation of 2200MiB does not leave room for system overhead (total system memory: 2460MiB). You may face stability issues.
* Suggestion: Start minikube with less memory allocated: 'minikube start --memory=2200mb'

* Starting control plane node minikube in cluster minikube
* Updating the running none "minikube" bare metal machine ...
* OS release is Ubuntu 18.04.5 LTS
* Preparing Kubernetes v1.20.2 on Docker 19.03.13 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Configuring local host environment ...
* Verifying Kubernetes components...
  - Using image kubernetesui/dashboard:v2.1.0
  - Using image kubernetesui/metrics-scrapers:v1.0.4
  - Using image k8s.gcr.io/metrics-server-amd64:v0.2.1
  - Using image gcr.io/k8s-minikube/storage-provisioner:v4
* Enabled addons: storage-provisioner, metrics-server, default-storageclass, dashboard
* Done! kubect1 is now configured to use "minikube" cluster and "default" namespace by default
$ kubect1 version
Client Version: version.Info{Major:"1", Minor:"20", GitVersion:"v1.20.4", GitCommit:"e87da0bd6e03fea7933c4b5263d151aafd07c", GitTreeState:"clean", BuildDate:"2021-02-18T16:12:00Z", GoVersion:"go1.15.8", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"20", GitVersion:"v1.20.2", GitCommit:"faecb196815e248d3ecfb03c680a4507229c2a56", GitTreeState:"clean", BuildDate:"2021-01-13T13:20:00Z", GoVersion:"go1.15.5", Compiler:"gc", Platform:"linux/amd64"}
$ kubect1 cluster info
Error: unknown command "cluster" for "kubect1"

Did you mean this?
    cluster-info

Run 'kubect1 --help' for usage.
$ kubect1 cluster-info
Kubernetes control plane is running at https://10.0.0.9:8443
KubeDNS is running at https://10.0.0.9:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubect1 cluster-info dump'.
$ kubect1 get nodes
NAME          STATUS    ROLES          AGE      VERSION
minikube      Ready     control-plane,master  27m      v1.20.2
$
```

## Create deployment

```
Terminal Preview Port 30000 +
* minikube v1.18.0 on Ubuntu 18.04 (amd64)
* Using the none driver based on existing profile

X The requested memory allocation of 2200MiB does not leave room for system overhead (total system memory: 2460MiB). You may face stability issues.
* Suggestion: Start minikube with less memory allocated: 'minikube start --memory=2200mb'

* Starting control plane node minikube in cluster minikube
* Updating the running none "minikube" bare metal machine ...
* OS release is Ubuntu 18.04.5 LTS
* Preparing Kubernetes v1.20.2 on Docker 19.03.13 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Configuring local host environment ...
* Verifying Kubernetes components...
  - Using image kubernetesui/dashboard:v2.1.0
  - Using image kubernetesui/metrics-scraper:v1.0.4
  - Using image k8s.gcr.io/metrics-server-amd64:v0.2.1
  - Using image gcr.io/k8s-minikube/storage-provisioner:v4
* Enabled addons: storage-provisioner, metrics-server, default-storageclass, dashboard
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
$ kubectl version
Client Version: version.Info{Major:"1", Minor:"20", GitVersion:"v1.20.4", GitCommit:"e87da0bd6e03c3fea7933c4b5263d151aafd07c", GitTreeState:"clean", BuildDate:"2021-02-18T16:12:00Z", GoVersion:"go1.15.8", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"20", GitVersion:"v1.20.2", GitCommit:"faecb196815e248d3ecfb03c680a4507229c2a56", GitTreeState:"clean", BuildDate:"2021-01-13T13:20:00Z", GoVersion:"go1.15.5", Compiler:"gc", Platform:"linux/amd64"}
$ kubectl cluster info
Error: unknown command "cluster" for "kubectl"

Did you mean this?
    cluster-info

Run 'kubectl --help' for usage.
$ kubectl cluster-info
Kubernetes control plane is running at https://10.0.0.9:8443
KubeDNS is running at https://10.0.0.9:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
$ kubectl get nodes
NAME      STATUS    ROLES    AGE     VERSION
minikube  Ready     control-plane,master  27m     v1.20.2
$
$ kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.39 -- /agnhost netexec --http-port=8080
deployment.apps/hello-node created
$
```

## Get deployments

```
Terminal Preview Port 30000 +
X The requested memory allocation of 2200MiB does not leave room for system overhead (total system memory: 2460MiB). You may face stability issues.
* Suggestion: Start minikube with less memory allocated: 'minikube start --memory=2200mb'

* Starting control plane node minikube in cluster minikube
* Updating the running none "minikube" bare metal machine ...
* OS release is Ubuntu 18.04.5 LTS
* Preparing Kubernetes v1.20.2 on Docker 19.03.13 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Configuring local host environment ...
* Verifying Kubernetes components...
  - Using image kubernetesui/dashboard:v2.1.0
  - Using image kubernetesui/metrics-scraper:v1.0.4
  - Using image k8s.gcr.io/metrics-server-amd64:v0.2.1
  - Using image gcr.io/k8s-minikube/storage-provisioner:v4
* Enabled addons: storage-provisioner, metrics-server, default-storageclass, dashboard
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
$ kubectl version
Client Version: version.Info{Major:"1", Minor:"20", GitVersion:"v1.20.4", GitCommit:"e87da0bd6e03c3fea7933c4b5263d151aafd07c", GitTreeState:"clean", BuildDate:"2021-02-18T16:12:00Z", GoVersion:"go1.15.8", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"20", GitVersion:"v1.20.2", GitCommit:"faecb196815e248d3ecfb03c680a4507229c2a56", GitTreeState:"clean", BuildDate:"2021-01-13T13:20:00Z", GoVersion:"go1.15.5", Compiler:"gc", Platform:"linux/amd64"}
$ kubectl cluster info
Error: unknown command "cluster" for "kubectl"

Did you mean this?
    cluster-info

Run 'kubectl --help' for usage.
$ kubectl cluster-info
Kubernetes control plane is running at https://10.0.0.9:8443
KubeDNS is running at https://10.0.0.9:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
$ kubectl get nodes
NAME      STATUS    ROLES    AGE     VERSION
minikube  Ready     control-plane,master  27m     v1.20.2
$
$ kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.39 -- /agnhost netexec --http-port=8080
deployment.apps/hello-node created
$ kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
hello-node    1/1     1            1           31s
$
```

## Get & describe pods

```
Terminal Preview Port 30000 +
Did you mean this?
cluster-info

Run 'kubectl --help' for usage.
$ kubectl cluster-info
Kubernetes control plane is running at https://10.0.0.9:8443
KubeDNS is running at https://10.0.0.9:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
$ kubectl get nodes
NAME        STATUS    ROLES    AGE   VERSION
minikube    Ready     control-plane,master  27m   v1.20.2
$
$ kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.39 -- /agnhost netexec --http-port=8080
deployment.apps/hello-node created
$ kubectl get deployments
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
hello-node  1/1     1             1           31s
$ kubectl get nodes
NAME        STATUS    ROLES    AGE   VERSION
minikube    Ready     control-plane,master  32m   v1.20.2
$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
hello-node-87cd7d8f5-6fw85         1/1     Running   0           63s
$ kubectl describe pods
Name:                hello-node-87cd7d8f5-6fw85
Namespace:           default
Priority:             0
Node:                minikube/10.0.0.9
Start Time:          Tue, 17 Jan 2023 19:09:49 +0000
Labels:              app=hello-node
                    pod-template-hash=87cd7d8f5
Annotations:         <none>
Status:              Running
IP:                  172.18.0.6
IPs:                 172.18.0.6
Controlled By:       ReplicaSet/hello-node-87cd7d8f5
Containers:
  agnhost:
    Container ID:  docker://083305e7732c58ac561200a21f3d15e5c7980e833a1ff6033f5d8ff6be4c36f4
    Image:         registry.k8s.io/e2e-test-images/agnhost:2.39
```

## Get services

```
Terminal Preview Port 30000 +
Initialized      True
Ready           True
ContainersReady True
PodScheduled    True
Volumes:
  default-token-lwd9q:
    Type:          Secret (a volume populated by a Secret)
    SecretName:    default-token-lwd9q
    Optional:      false
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                 node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason      Age   From      Message
  ----     -
  Normal   Scheduled   82s   default-scheduler   Successfully assigned default/hello-node-87cd7d8f5-6fw85 to minikube
  Normal   Pulling     81s   kubelet        Pulling image "registry.k8s.io/e2e-test-images/agnhost:2.39"
  Normal   Pulled      78s   kubelet        Successfully pulled image "registry.k8s.io/e2e-test-images/agnhost:2.39" in 3.366534374s
  Normal   Created     78s   kubelet        Created container agnhost
  Normal   Started     77s   kubelet        Started container agnhost
$ kubectl get events
LAST SEEN   TYPE      REASON      OBJECT                                MESSAGE
2m41s      Normal    Scheduled   pod/hello-node-87cd7d8f5-6fw85       Successfully assigned default/hello-node-87cd7d8f5-6fw85 to minikube
2m40s      Normal    Pulling     pod/hello-node-87cd7d8f5-6fw85       Pulling image "registry.k8s.io/e2e-test-images/agnhost:2.39"
2m37s      Normal    Pulled      pod/hello-node-87cd7d8f5-6fw85       Successfully pulled image "registry.k8s.io/e2e-test-images/agnhost:2.39" in
3.366534374s
2m37s      Normal    Created     pod/hello-node-87cd7d8f5-6fw85       Created container agnhost
2m36s      Normal    Started     pod/hello-node-87cd7d8f5-6fw85       Started container agnhost
2m41s      Normal    SuccessfulCreate   replicaset/hello-node-87cd7d8f5       Created pod: hello-node-87cd7d8f5-6fw85
2m41s      Normal    ScalingReplicaSet   deployment/hello-node                 Scaled up replica set hello-node-87cd7d8f5 to 1
33m        Normal    Starting     node/minikube                         Starting kubelet.
33m        Normal    NodeHasSufficientMemory   node/minikube                         Node minikube status is now: NodeHasSufficientMemory
33m        Normal    NodeHasNoDiskPressure     node/minikube                         Node minikube status is now: NodeHasNoDiskPressure
33m        Normal    NodeHasSufficientPID       node/minikube                         Node minikube status is now: NodeHasSufficientPID
33m        Normal    NodeAllocatableEnforced    node/minikube                         Updated Node Allocatable limit across pods
33m        Normal    RegisteredNode            node/minikube                         Node minikube event: Registered Node minikube in Controller
33m        Normal    NodeReady                 node/minikube                         Node minikube status is now: NodeReady
33m        Normal    Starting                 node/minikube                         Starting kube-proxy.
$ kubectl get services
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
kubernetes   ClusterIP   10.96.0.1    <none>         443/TCP    33m
```



