22AIE305: CLOUD COMPUTING

Minikube

Minikube is a lightweight Kubernetes cluster setup you can use to develop and test applications on Kubernetes. You can create single and multi-node clusters with Minikube.

Minikube is a cross-platform, community-driven Kubernetes distribution, which is targeted to be used primarily in local environments.

It deploys a single-node cluster, which is an excellent option for having a simple Kubernetes cluster up and running on a localhost.

Minikube is designed to be used as a virtual machine (VM), and the default VM runtime is VirtualBox. At the same time, extensibility is one of the critical benefits of Minikube, so it's possible to use it with drivers outside of VirtualBox.

Minikube Prerequisites

- 2 CPUs or more
- 2GB of free memory
- 20GB of free disk space (on the drive where installed)
- Internet connection
- Container or virtual machine manager, such as Docker, Hyperkit, HyperV, KVM, Parallels, Podman, VirtualBox, or VMWare. Ensure you install any of the tools before you start with Minikube installation.

Installing Minikube

The official Minikube website https://minikube.sigs.k8s.io/docs/start/

Minikube also available at **https://github.com/kubernetes/minikube**

You can also Clone it from

https://github.com/kubernetes/minikube.git

You'll need to download the latest release of Minikube for Windows **AMD64.exe** system, rename it to minikube.exe, and add it to your path.

https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download

Once installed, validate the installation by running the command 'minikube version'.

Starting Minikube

```
# For Oracle virtualbox users
minikube start --driver=virtualbox
# For Docker users (on Windows)
minikube start -p dev --container-runtime=docker --vm=true
# for Git users
git clone https://github.com/kubernetes/minikube.git
```

To get the node IP of minikube, execute the following command. You can use the IP to access nodePorts.

```
minikube ip
```

Starting Minikube

When you run minikube start, Minikube will create a local virtual machine and deploy all necessary Kubernetes components into it.

This VM will be configured with Docker and Kubernetes via a single binary known as the local Kube.

The minikube start command creates a new virtual machine based on the Minikube image, which includes Docker and RKP container images and a local Kube library.

Use `kubectl get pods’ or `minikube status` to verify whether it is properly setup.

Create a Namespace

Namespaces in Kubernetes serve as a mechanism for dividing cluster resources between multiple users, applications, or environments.

Creating separate namespaces for different applications or environments (e.g., development, staging, production) is a common practice. It can hold all the resources related to our application.

Minikube dashboard

Give the command “minikube dashboard” to bring up the dashboard.

Dashboard is very convenient to interact with Minikube.

Alternately, you can use command prompt (in “Run as Administrator” mode to interact with it)

```
minikube start \
--kubernetes-version stable \
--nodes 2 \
--cpus 2 \
--memory 2000 \
--cni calico
```

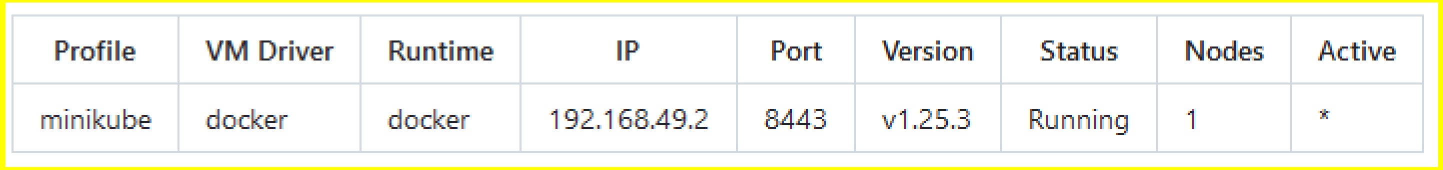
Sample commands

```
minikube start
# Start a new terminal, and leave this running.
minikube dashboard
kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-
    images/agnhost:2.39 -- /agnhost netexec --http-port=8080
kubectl get deployments
kubectl get pods
kubectl get events
```

Ip address

Use the following command to get the IP address of minikube.

minikube profile list



Profile	VM Driver	Runtime	IP	Port	Version	Status	Nodes	Active
minikube	docker	docker	192.168.49.2	8443	v1.25.3	Running	1	*

Creating a Service

By default, the Pod is only accessible by its internal IP address within the Kubernetes cluster.

To make the Container accessible from outside the Kubernetes virtual network, you have to expose the Pod as a Kubernetes Service.

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

```
kubectl expose deployment hello-node --type=LoadBalancer --port=8080
```

View the Service you created:

kubectl get services

On cloud providers that support load balancers, an external IP address would be provisioned to access the Service.

On minikube, the LoadBalancer type makes the Service accessible through the minikube service command.

minikube service hello-node

minikube addons list

```
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
```

```
minikube addons enable metrics-server
kubectl get pod,svc -n kube-system
NAME                                        READY  STATUS    RESTARTS   AGE
pod/coredns-5644d7b6d9-mh9ll                 1/1    Running   0          34m
pod/coredns-5644d7b6d9-pqd2t                 1/1    Running   0          34m
pod/metrics-server-67fb648c5                 1/1    Running   0          26s
pod/etcd-minikube                            1/1    Running   0          34m
pod/influxdb-grafana-b29w8                   2/2    Running   0          26s
pod/kube-addon-manager-minikube              1/1    Running   0          34m
pod/kube-apiserver-minikube                  1/1    Running   0          34m
pod/kube-controller-manager-minikube         1/1    Running   0          34m
pod/kube-proxy-rnlps                         1/1    Running   0          34m
pod/kube-scheduler-minikube                  1/1    Running   0          34m
pod/storage-provisioner                      1/1    Running   0          34m
```

Cleaning up Minikube

```
kubectl delete service hello-node
kubectl delete deployment hello-node
minikube stop
minikube delete
```