K8S (1) Minikube

Following the edX LFS158X, we use Minikube to set up the cluster for demoing purposes.
minikube uses kubeasm as a bootstrap.

The recommended tutorial method didn't work on my local machine ($ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-darwin-amd64) since my Mac M1 is an arm chip.
Check the release page https://storage.googleapis.com/minikube/ and download the matching version.

```
$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-darwin-
amd64
$ sudo install minikube-darwin-arm64 /usr/local/bin/minikube
```

minikube comes with kubectl bin but you have to use $ minikue kubectl every time (the kubectl is installed in minikube's virtual machine). Recommend to install kubectl directly.

```
$ brew install kubectl
$ brew install kubernetes-cli
$ kubectl version
Client Version: v1.29.0
```

If you are also using a Mac arm chip, you might have issues using VirtualBox as default driver. It is a known issue that VirtualBox doesn't support arm chip. Check the minikube support driver list and choose compatible driver. Refer here if you choose Qemu. (I turn to Qemu when trying the NodePort demo later)

kubectl needs the cluster control plane endpoint and credential to access the minikube API Server. Check the config file ~/.kube/config or use command

```
$ kubectl config view
apiVersion: v1
clusters:
- cluster:
certificate-authority: /Users/chenyang/.minikube/ca.crt
server: https://127.0.0.1:65227
```

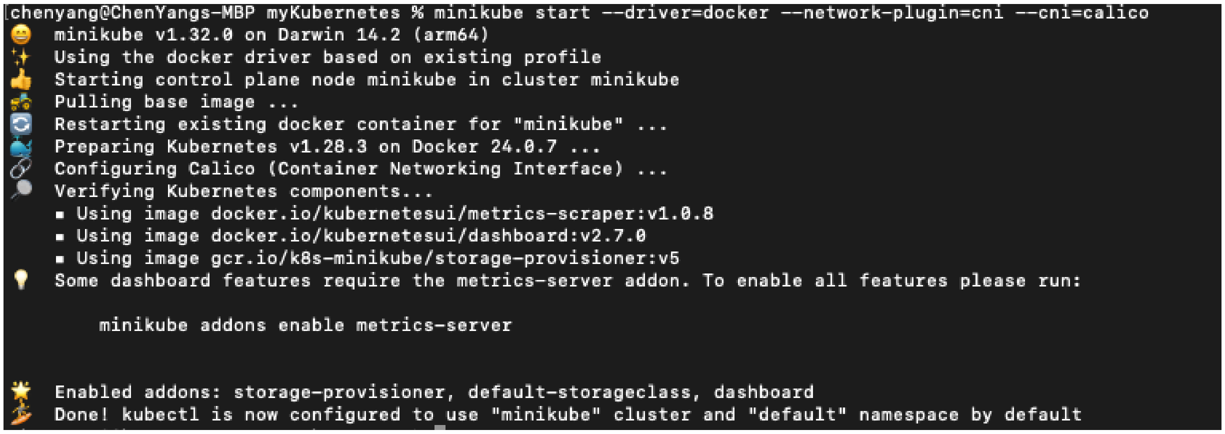
```
name: minikube
contexts:
- context:
cluster: minikube
extensions:
- extension:
last-update: Fri, 22 Dec 2023 09:31:36 PST
provider: minikube.sigs.k8s.io
version: v1.32.0
name: context_info
namespace: default
user: minikube
name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
user:
client-certificate: /Users/chenyang/.minikube/profiles/minikube/client.crt
client-key: /Users/chenyang/.minikube/profiles/minikube/client.key

$ kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:65227
CoreDNS is running at https://127.0.0.1:65227/api/v1/namespaces/kube-
system/services/kube-dns:dns/proxy
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

Notice the ~/.kube/config is for minikube only, other container orchestration may save the config in different place.
The API Server's endpoint is https://127.0.0.1:65227, matches the cluster-info. Also the included the authN key/certificate for our first user minikube.
We will discuss about the context and user when we talk about authN.

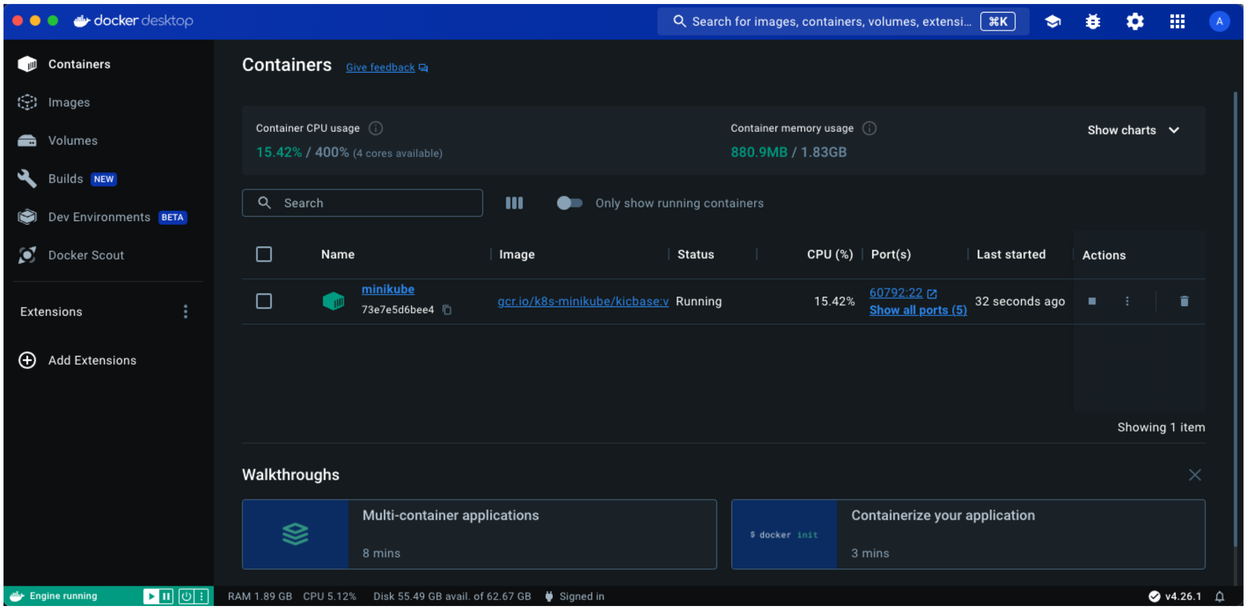
Here I use Docker. The minikube start command is

```
$ minikube start --driver=docker --network-plugin=cni --cni=calico
```

The usage of the CNI will be explain when we talk about Service.



```
chenyang@ChenYangs-MBP myKubernetes % minikube start --driver=docker --network-plugin=cni --cni=calico
😄  minikube v1.32.0 on Darwin 14.2 (arm64)
✨  Using the docker driver based on existing profile
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
🔄  Restarting existing docker container for "minikube" ...
🐳  Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
🔗  Configuring Calico (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
    ▪ Using image docker.io/kubernetesui/dashboard:v2.7.0
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
💡  Some dashboard features require the metrics-server addon. To enable all features please run:

        minikube addons enable metrics-server


🌟  Enabled addons: storage-provisioner, default-storageclass, dashboard
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```



Containers

Container CPU usage 15.42% / 400% (4 cores available)

Container memory usage 880.9MB / 1.83GB

□	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
□	minikube 73e7e5d6bee4	gcr.io/k8s-minikube/kicbase:v	Running	15.42%	60792:22 Show all ports (5)	32 seconds ago	

Showing 1 item

Walkthroughs

Multi-container applications 8 mins

$ docker init — Containerize your application 3 mins

RAM 1.89 GB CPU 5.12% Disk 55.49 GB avail. of 62.67 GB Signed in v4.26.1

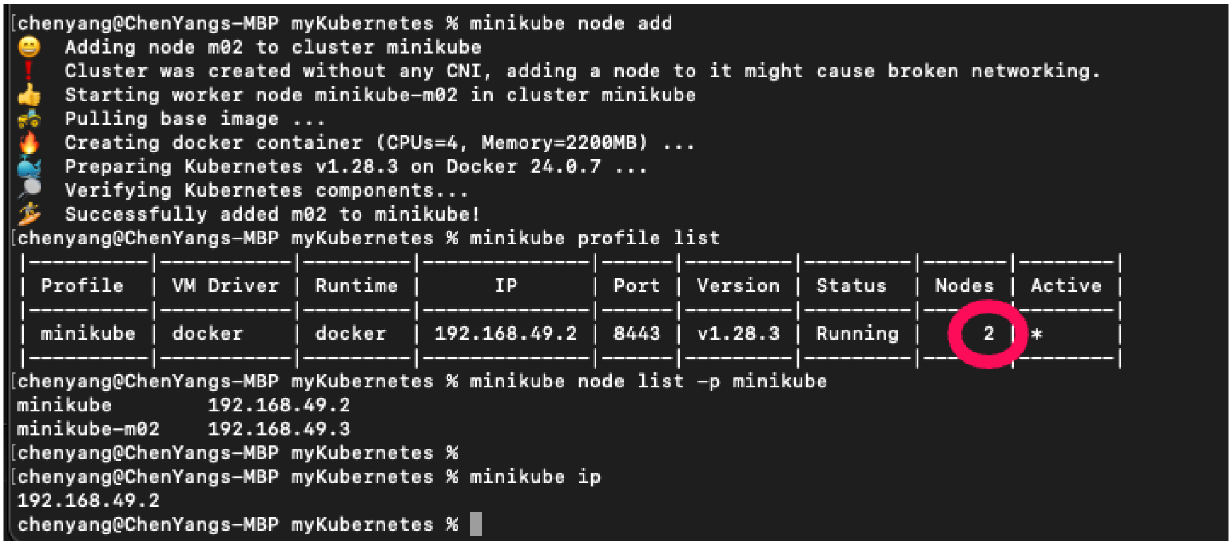
minikube common command:

```
$ minikube start --driver=virtualbox --nodes=3 --disk-size=10g --cpus=2 --memory=4g
--kubernetes-version=v1.25.1 --cni=calico --container-runtime=cri-o -p multivbox
$ minikube start -p minikube
$ minikube stop -p minikube
$ minikube delete -p minibox
$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
$ minikube profile list
// Every profile maps to one Cluster. If you didn't assign name, the first one will
be minikube
```

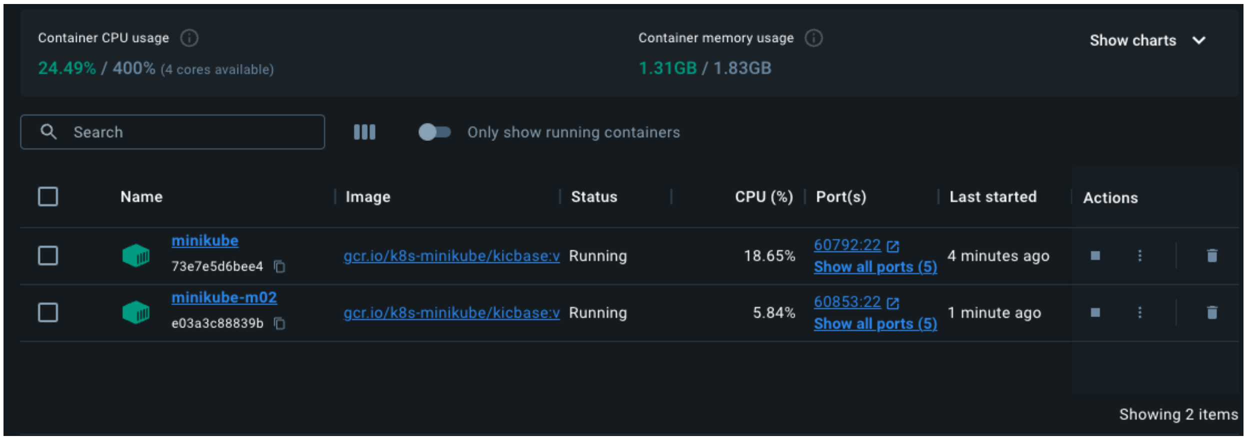
```
| Profile | VM Driver | Runtime | IP | Port | Version | Status | Nodes | Active |
|----------|-----------|---------|--------------|------|---------|---------|-------
|--------|
| minikube | docker | docker | 192.168.49.2 | 8443 | v1.28.3 | Stopped | 1 | * |
|----------|-----------|---------|--------------|------|---------|---------|-------
|--------|
$ minikube profile minikube
// Set minikube as your default profile
$ minikube node list -p minibox
minikube 192.168.49.2
$ minikube ip
```

After you created cluster, you can still scale up/down the number of node by using command

```
$ minikube node add
$ minikube node delete minikube-m02
```



```
[chenyang@ChenYangs-MBP myKubernetes % minikube node add
😄  Adding node m02 to cluster minikube
❗  Cluster was created without any CNI, adding a node to it might cause broken networking.
👍  Starting worker node minikube-m02 in cluster minikube
🚜  Pulling base image ...
🔥  Creating docker container (CPUs=4, Memory=2200MB) ...
🐳  Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
🔎  Verifying Kubernetes components...
🏄  Successfully added m02 to minikube!
[chenyang@ChenYangs-MBP myKubernetes % minikube profile list
|----------|-----------|---------|--------------|------|---------|---------|-------|--------|
| Profile  | VM Driver | Runtime |      IP      | Port | Version | Status  | Nodes | Active |
|----------|-----------|---------|--------------|------|---------|---------|-------|--------|
| minikube | docker    | docker  | 192.168.49.2 | 8443 | v1.28.3 | Running |     2 | *      |
|----------|-----------|---------|--------------|------|---------|---------|-------|--------|
[chenyang@ChenYangs-MBP myKubernetes % minikube node list -p minikube
minikube        192.168.49.2
minikube-m02    192.168.49.3
[chenyang@ChenYangs-MBP myKubernetes %
[chenyang@ChenYangs-MBP myKubernetes % minikube ip
192.168.49.2
chenyang@ChenYangs-MBP myKubernetes %
```



Container CPU usage
24.49% / 400% (4 cores available)

Container memory usage
1.31GB / 1.83GB

□	Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
□	minikube 73e7e5d6bee4	gcr.io/k8s-minikube/kicbase:v	Running	18.65%	60792:22 Show all ports (5)	4 minutes ago	
□	minikube-m02 e03a3c88839b	gcr.io/k8s-minikube/kicbase:v	Running	5.84%	60853:22 Show all ports (5)	1 minute ago	

Showing 2 items

If you meet "not enough resource"

```
$ minikube config set cpu 3
$ minikube config set memory 4092
```