Step 1: Check Docker is installed and running

Sudo docker --version

Sudo docker run hello-world

```
sudipta@SAMBIT:~$ sudo docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
sudipta@SAMBIT:~$
sudipta@SAMBIT:~$ sudo docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

sudipta@SAMBIT:~$
```

Step 2: Install kind (Kubernetes in Docker) and verify

curl -Lo ./kind https://kind.sigs.k8s.io/dl/v0.22.0/kind-linux-amd64

chmod +x ./kind

sudo mv ./kind /usr/local/bin/kind

kind version

```
sudipta@SAMBIT:~$ curl -Lo ./kind https://kind.sigs.k8s.io/dl/v0.22.0/kind-linux-amd64
chmod +x ./kind
sudo mv ./kind /usr/local/bin/kind
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    97  100    97    0     0    147      0 --:--:-- --:--:-- --:--:--   147
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 6245k  100 6245k    0     0  2022k      0  0:00:03  0:00:03 --:--:-- 5953k
sudipta@SAMBIT:~$ kind version
kind v0.22.0 go1.20.13 linux/amd64
sudipta@SAMBIT:~$
```

Create a Directory (optional but recommended for better structure)

mkdir -p ~/k8s-lab/kind && cd ~/k8s-lab/kind

```
sudipta@SAMBIT:~$ mkdir -p ~/k8s-lab/kind && cd ~/k8s-lab/kind
sudipta@SAMBIT:~/k8s-lab/kind$
```

Step 3: Create kind-cluster.yaml file under ~/k8s-lab/kind and paste the configuration

sudo nano kind-cluster.yaml

Configuration:

```
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
nodes:
- role: control-plane
  kubeadmConfigPatches:
    - |
      kind: InitConfiguration
      nodeRegistration:
        kubeletExtraArgs:
          system-reserved: "memory=700Mi"
          kube-reserved: "memory=800Mi"
- role: worker
  kubeadmConfigPatches:
    - |
      kind: JoinConfiguration
      nodeRegistration:
        kubeletExtraArgs:
          system-reserved: "memory=300Mi"
          kube-reserved: "memory=300Mi"
- role: worker
  kubeadmConfigPatches:
    - |
      kind: JoinConfiguration
      nodeRegistration:
        kubeletExtraArgs:
          system-reserved: "memory=300Mi"
          kube-reserved: "memory=300Mi"
- role: worker
```

```
  kubeadmConfigPatches:
    - |
      kind: JoinConfiguration
      nodeRegistration:
        kubeletExtraArgs:
          system-reserved: "memory=300Mi"
          kube-reserved: "memory=300Mi"
- role: worker
  kubeadmConfigPatches:
    - |
      kind: JoinConfiguration
      nodeRegistration:
        kubeletExtraArgs:
          system-reserved: "memory=300Mi"
          kube-reserved: "memory=300Mi"
```

```
sudipta@SAMBIT:~/k8s-lab/kind$ sudo nano kind-cluster.yaml
sudipta@SAMBIT:~/k8s-lab/kind$ cat kind-cluster.yaml
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
nodes:
- role: control-plane
  kubeadmConfigPatches:
    - |
      kind: InitConfiguration
      nodeRegistration:
        kubeletExtraArgs:
          system-reserved: "memory=700Mi"
          kube-reserved: "memory=800Mi"
- role: worker
  kubeadmConfigPatches:
    - |
      kind: JoinConfiguration
      nodeRegistration:
        kubeletExtraArgs:
          system-reserved: "memory=300Mi"
          kube-reserved: "memory=300Mi"
- role: worker
  kubeadmConfigPatches:
    - |
      kind: JoinConfiguration
      nodeRegistration:
        kubeletExtraArgs:
          system-reserved: "memory=300Mi"
          kube-reserved: "memory=300Mi"
```

```
- role: worker
  kubeadmConfigPatches:
    - |
      kind: JoinConfiguration
      nodeRegistration:
        kubeletExtraArgs:
          system-reserved: "memory=300Mi"
          kube-reserved: "memory=300Mi"
- role: worker
  kubeadmConfigPatches:
    - |
      kind: JoinConfiguration
      nodeRegistration:
        kubeletExtraArgs:
          system-reserved: "memory=300Mi"
          kube-reserved: "memory=300Mi"
sudipta@SAMBIT:~/k8s-lab/kind$
```

Step 4. Create the Cluster Using the YAML File

sudo kind create cluster --config kind-cluster.yaml --name my-k8s-lab

```
sudipta@SAMBIT:~/k8s-lab/kind$ sudo kind create cluster --config kind-cluster.yaml --name my-k8s-lab
Creating cluster "my-k8s-lab" ...
 ✓ Ensuring node image (kindest/node:v1.29.2) 🖼
 ✓ Preparing nodes 📦 📦 📦 📦 📦
 ✓ Writing configuration 📜
 ✓ Starting control-plane 🕹️
 ✓ Installing CNI 🔌
 ✓ Installing StorageClass 💾
 ✓ Joining worker nodes 🚜
Set kubectl context to "kind-my-k8s-lab"
You can now use your cluster with:

kubectl cluster-info --context kind-my-k8s-lab

Have a question, bug, or feature request? Let us know! https://kind.sigs.k8s.io/#community 🙂
sudipta@SAMBIT:~/k8s-lab/kind$
```

Step 5: Install Kubectl

sudo mkdir -p /etc/apt/keyrings

**echo "deb [signed-by=/etc/apt/keyrings/kubernetes-archive-keyring.gpg] **

**https://apt.kubernetes.io/ kubernetes-xenial main" | **

sudo tee /etc/apt/sources.list.d/kubernetes.list

deb [signed-by=/etc/apt/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main

KUBECTL_VERSION=v1.29.3

echo $KUBECTL_VERSION

curl -LO https://dl.k8s.io/release/${KUBECTL_VERSION}/bin/linux/amd64/kubectl

chmod +x kubectl

sudo mv kubectl /usr/local/bin/

kubectl version –client

```
sudipta@SAMBIT:~/k8s-lab/kind$ KUBECTL_VERSION=v1.29.3
sudipta@SAMBIT:~/k8s-lab/kind$ echo $KUBECTL_VERSION
v1.29.3
sudipta@SAMBIT:~/k8s-lab/kind$ curl -LO "https://dl.k8s.io/release/${KUBECTL_VERSION}/bin/linux/amd64/kubectl"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0    422      0 --:--:-- --:--:-- --:--:--   423
100 47.4M  100 47.4M    0     0  6949k      0  0:00:06  0:00:06 --:--:-- 7699k
sudipta@SAMBIT:~/k8s-lab/kind$ chmod +x kubectl
sudipta@SAMBIT:~/k8s-lab/kind$ sudo mv kubectl /usr/local/bin/
sudipta@SAMBIT:~/k8s-lab/kind$ kubectl version --client
Client Version: v1.29.3
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
sudipta@SAMBIT:~/k8s-lab/kind$ sudo kubectl get nodes
NAME                       STATUS   ROLES           AGE   VERSION
my-k8s-lab-control-plane   Ready    control-plane   10m   v1.29.2
my-k8s-lab-worker          Ready    <none>          10m   v1.29.2
my-k8s-lab-worker2         Ready    <none>          10m   v1.29.2
my-k8s-lab-worker3         Ready    <none>          10m   v1.29.2
my-k8s-lab-worker4         Ready    <none>          10m   v1.29.2
sudipta@SAMBIT:~/k8s-lab/kind$
```

Step 6: Verify Cluster Info

```
sudipta@SAMBIT:~/k8s-lab/kind$ sudo kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:46615
CoreDNS is running at https://127.0.0.1:46615/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
sudipta@SAMBIT:~/k8s-lab/kind$ sudo kubectl get nodes
NAME                       STATUS   ROLES           AGE   VERSION
my-k8s-lab-control-plane   Ready    control-plane   30m   v1.29.2
my-k8s-lab-worker          Ready    <none>          29m   v1.29.2
my-k8s-lab-worker2         Ready    <none>          29m   v1.29.2
my-k8s-lab-worker3         Ready    <none>          29m   v1.29.2
my-k8s-lab-worker4         Ready    <none>          29m   v1.29.2
sudipta@SAMBIT:~/k8s-lab/kind$ sudo kubectl get pods -A
NAMESPACE            NAME                                                READY   STATUS    RESTARTS   AGE
kube-system          coredns-76f75df574-86dgx                            1/1     Running   0          29m
kube-system          coredns-76f75df574-ttw4q                            1/1     Running   0          29m
kube-system          etcd-my-k8s-lab-control-plane                       1/1     Running   0          30m
kube-system          kindnet-6xz5v                                       1/1     Running   0          29m
kube-system          kindnet-kv6br                                       1/1     Running   0          29m
kube-system          kindnet-m2xtf                                       1/1     Running   0          29m
kube-system          kindnet-pmhs2                                       1/1     Running   0          29m
kube-system          kindnet-xn7hm                                       1/1     Running   0          29m
kube-system          kube-apiserver-my-k8s-lab-control-plane             1/1     Running   0          30m
kube-system          kube-controller-manager-my-k8s-lab-control-plane    1/1     Running   0          30m
kube-system          kube-proxy-982xc                                    1/1     Running   0          29m
kube-system          kube-proxy-fsmcn                                    1/1     Running   0          29m
kube-system          kube-proxy-fv4bg                                    1/1     Running   0          29m
kube-system          kube-proxy-fww6n                                    1/1     Running   0          29m
kube-system          kube-proxy-p46j4                                    1/1     Running   0          29m
kube-system          kube-scheduler-my-k8s-lab-control-plane             1/1     Running   0          30m
local-path-storage   local-path-provisioner-7577fdbbfb-5lm2b             1/1     Running   0          29m
sudipta@SAMBIT:~/k8s-lab/kind$
```