

Advance Devops Experiment 4

Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes cluster and deploy Your First Kubernetes

Application.

Theory: What is kubectl?

kubectl is the command-line tool for interacting with Kubernetes clusters. It allows you to manage Kubernetes resources by creating, updating, and deleting pods, deployments, services, and more.

Prerequisites

A Kubernetes cluster running either locally (e.g., with Minikube, Kind, or Docker Desktop) or remotely (cloud-based, such as Google Kubernetes Engine (GKE), Amazon Elastic Kubernetes Service (EKS), or Azure Kubernetes Service (AKS)).

kubectl installed on your local machine to interact with the cluster.

Step 1:

Go to AWS Academia in services select EC2 and create 3 instance with instance type t2.medium and names as node1, node2 and master

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name
Exp_4 [Add additional tags](#)

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Recents **Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Li [Browse more AMIs](#)

Summary

Number of instances [Info](#)
1

Software image (AMI)
Canonical, Ubuntu, 24.04, amd64...[read more](#)
ami-0e86e20dae9224db8

Virtual server type (instance type)
t2.medium

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million I/Os. [X](#)

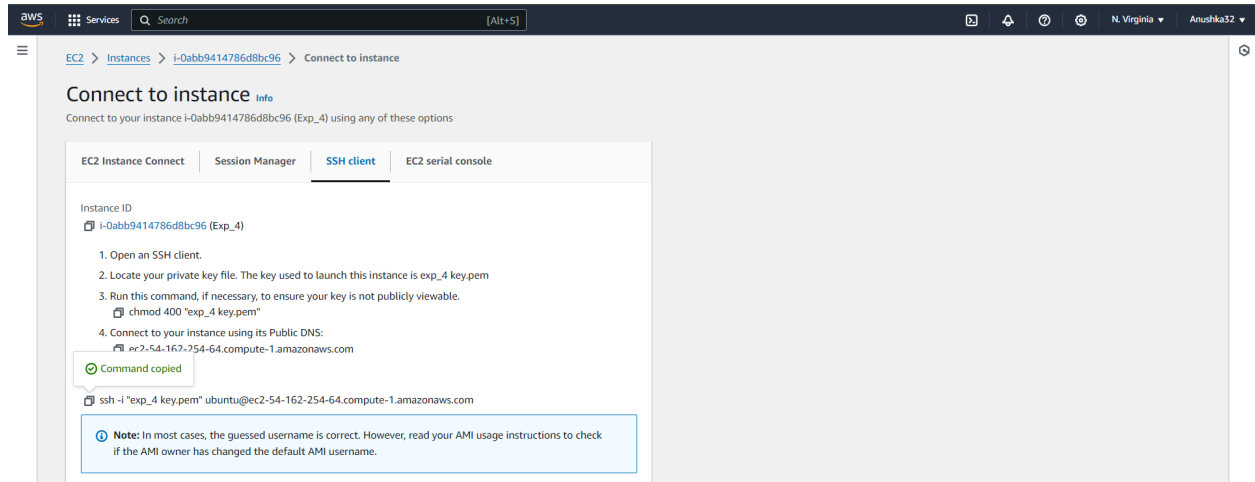
Cancel [Launch instance](#) [Review commands](#)

Success
Successfully initiated launch of Instance (i-0abb9414786d8b096)

[Launch log](#)

Step 2: Create a new key pair and name it as myKey1 and download as .pem file.

Open command prompt run the following command



Step 3: Now open the folder in the terminal where our .pem key is stored and paste the Example

command (starting with ssh -i) in the terminal.(ssh -i "Master_Ec2_Key.pem" ubuntu@ec2-54-196-129-215.compute-1.amazonaws.com)

```
System information as of Tue Sep 24 21:28:51 UTC 2024

System load:  0.08          Processes:            117
Usage of /:   22.8% of 6.71GB Users logged in:      0
Memory usage: 5%           IPv4 address for enX0: 172.31.86.116
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-86-116:~$
```

Step 4: Run the below commands to install and setup Docker.

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
```

```
ubuntu@ip-172-31-86-116:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo tee
/etc/apt/trusted.gpg.d/docker.gpg > /dev/null
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFit2ioBEADhWpZ8/wvZ6hUTiXOWQHXMALaFHCpH9hAtr4F1y2+0YdbtMuth
lqqwp028AqyY+PRfVMtSYMBjuQuu5byyKR01BbqYhuS3jtgQmLjZ/bJvXqnmVXh
38UuLaiz077PxyxQhu5BbqntTPQMfiyqEiU+BKbq2WmANUKQf+1AmZY/Iru0Xbnq
L4C1+gJ8vfmXQe99npCaxEjaNRVYfOS8QcixNzHUYnb6emjLANyEVLZzeqo7XKl7
UrwV5inawTSzWwvtjEjj4nJL8NsLwscLPQUhTQ+7BbQXAwAmeHCUTQIvvWXqw0N
cmhh4HgeQscQHYgOJjjDVfoY5MucvgLbIgCqfzAHW9jxmRL4qbMZj+b1XoePEtht
ku4bIQN1X5P07fNWzLgaRL5Z4POXDDZTLIQ/EL58j9kp4bnWRCJW0lya+f8ocodo
vZZ+Doi+fy4D5ZGrL4XEcIQP/Lv5uFyf+kQtL/94VFYVJ0leAv8W92KdgDkhTcTD
G7c0tIkVEKNUq48b3aQ64N0ZQW7fVjfoKwEZdOqPE72Pa45jrZzvUFxSpdiNk2tZ
XYukHjLxxEgBdC/J3cMMNRE1F4NCA3ApfV1Y7/hTe0nmDuDYwr9/obA8t016Yljj
q5rdkymPf4JF8mXUW5eCN1vAFHxeg9ZWmhBtQmGxXnw9M+z6hWwc6ahmwARAQAB
tCtEb2NrZXIuUmVsZWZfZSAoQ0UgZGVikaSA8ZG9ja2VyQGRRvY2tLci5jb20+iQI3
BBMRCQAhBQJYRefAAhsvBQsJCAcDBRUKCQoLBRyCAwEAAh4BAheAAAoJFET2BqDwO
```

sudo apt-get update

sudo apt-get install -y docker-ce

```
/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
ubuntu@ip-172-31-86-116:~$ sudo apt-get update
sudo apt-get install -y docker-ce
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://download.docker.com/linux/ubuntu noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt
/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0
  pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7
  libslirp0 pigz slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 139 not upgraded.
Need to get 123 MB of archives.
After this operation, 442 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 libltdl7 amd64 2.4.7-7build1 [40.3 kB]
```

```
sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
```

```
no in guests are running validated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-86-116:~$ sudo mkdir -p /etc/docker
cat <<EOF | sudo tee /etc/docker/daemon.json
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
EOF
{
"exec-opts": ["native.cgroupdriver=systemd"]
}
ubuntu@ip-172-31-86-116:~$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
ubuntu@ip-172-31-86-116:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o
/etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
gpg: missing argument for option "-o"
-bash: /etc/apt/keyrings/kubernetes-apt-keyring.gpg: No such file or directory
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
ubuntu@ip-172-31-86-116:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
```

```
sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
```

```
ubuntu@ip-172-31-86-116:~$ sudo systemctl enable docker
sudo systemctl daemon-reload
sudo systemctl restart docker
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
ubuntu@ip-172-31-86-116:~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o
/etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
gpg: missing argument for option "-o"
-bash: /etc/apt/keyrings/kubernetes-apt-keyring.gpg: No such file or directory
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /
ubuntu@ip-172-31-86-116:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
E: Malformed entry 1 in list file /etc/apt/sources.list.d/kubernetes.list (URI)
E: The list of sources could not be read.
ubuntu@ip-172-31-86-116:~$
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o
/etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
```

Step 5: Run the below command to install Kubernets.

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor
-o
/etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' | sudo tee
/etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
```

```
ubuntu@ip-172-31-86-116:~$ sudo mkdir -p /etc/apt/keyrings
ubuntu@ip-172-31-86-116:~$ sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://download.docker.com/linux/ubuntu noble InRelease
Get:6 https://prod-cdn.packages.k8s.io/repositories/isc/kubernetes:/core:/stable:/v1.31/deb InRelease [1186 B]
Get:7 https://prod-cdn.packages.k8s.io/repositories/isc/kubernetes:/core:/stable:/v1.31/deb Packages [4865 B]
Fetched 6051 B in 1s (11.3 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt
/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  conntrack cri-tools kubernetes-cni
The following NEW packages will be installed:
  conntrack cri-tools kubeadm kubectl kubelet kubernetes-cni
0 upgraded, 6 newly installed, 0 to remove and 139 not upgraded.
Need to get 87.4 MB of archives.
After this operation, 314 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 conntrack amd64 1:1.4.8-1ubuntu1 [37.9 kB]
Get:2 https://prod-cdn.packages.k8s.io/repositories/isc/kubernetes:/core:/stable:/v1.31/deb cri-tools 1.31.1-1.1 [15.7
MB]
Get:3 https://prod-cdn.packages.k8s.io/repositories/isc/kubernetes:/core:/stable:/v1.31/deb kubeadm 1.31.1-1.1 [11.4 M
```

```
ubuntu@ip-172-31-86-116:~$ kubectl describe pod nginx-deployment-d556bf558-dmdsr
Name:          nginx-deployment-d556bf558-dmdsr
Namespace:     default
Priority:       0
Service Account: default
Node:          <none>
Labels:        app=nginx
               pod-template-hash=d556bf558
Annotations:   <none>
Status:        Pending
IP:            <none>
IPs:           <none>
Controlled By: ReplicaSet/nginx-deployment-d556bf558
```

```

ubuntu@ip-172-31-86-116:~$ kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
error: flag needs an argument: 'f' in -f
See 'kubectl apply --help' for usage.
-bash: https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml: No such file or directory
ubuntu@ip-172-31-86-116:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@ip-172-31-86-116:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-86-116:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-dmdsr    0/1     Pending   0           10s
nginx-deployment-d556bf558-kf9l4    0/1     Pending   0           10s
ubuntu@ip-172-31-86-116:~$
POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
kubectl port-forward $POD_NAME 8081:80
error: unable to forward port because pod is not running. Current status=Pending
ubuntu@ip-172-31-86-116:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untaint
ed
error: at least one taint update is required
ubuntu@ip-172-31-86-116:~$ kubectl get nodes
NAME                                STATUS   ROLES    AGE     VERSION
ip-172-31-86-116                    Ready   control-plane  4m15s   v1.31.1
ubuntu@ip-172-31-86-116:~$

```

sudo systemctl enable --now kubelet

sudo kubeadm init --pod-network-cidr=10.244.0.0/16

```

ubuntu@ip-172-31-86-116:~$ sudo apt-get install -y socat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
  slirp4netns
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  socat
0 upgraded, 1 newly installed, 0 to remove and 139 not upgraded.
Need to get 374 kB of archives.
After this operation, 1649 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 socat amd64 1.8.0.0-4build3 [374 kB]
Fetched 374 kB in 0s (14.0 MB/s)
Selecting previously unselected package socat.
(Reading database ... 68108 files and directories currently installed.)
Preparing to unpack .../socat_1.8.0.0-4build3_amd64.deb ...
Unpacking socat (1.8.0.0-4build3) ...
Setting up socat (1.8.0.0-4build3) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

```

sudo mkdir -p /etc/containerd

sudo containerd config default | sudo tee /etc/containerd/config.toml

sudo systemctl restart containerd

sudo systemctl enable containerd

sudo systemctl status containerd

```

ubuntu@ip-172-31-86-116:~$ sudo systemctl restart containerd
sudo systemctl enable containerd
sudo systemctl status containerd
● containerd.service - containerd container runtime
   Loaded: loaded (/usr/lib/systemd/system/containerd.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-09-24 21:37:08 UTC; 220ms ago
     Docs: https://containerd.io
   Main PID: 4764 (containerd)
    Tasks: 7
   Memory: 13.4M (peak: 14.0M)
      CPU: 55ms
   CGroup: /system.slice/containerd.service
           └─4764 /usr/bin/containerd

Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.940185812Z" level=info msg="Start subscrib>
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.940228819Z" level=info msg="Start recoveri>
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.940274910Z" level=info msg="Start event mo>
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.940284757Z" level=info msg="Start snapshot>
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.940292060Z" level=info msg="Start cni netw>
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.940298061Z" level=info msg="Start streamin>
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.940208336Z" level=info msg="serving... addr>
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.941160760Z" level=info msg="serving... addr>
Sep 24 21:37:08 ip-172-31-86-116 systemd[1]: Started containerd.service - containerd container runtime.
Sep 24 21:37:08 ip-172-31-86-116 containerd[4764]: time="2024-09-24T21:37:08.941884864Z" level=info msg="containerd suc>

```

Step 6: Initialize the Kubecluster

`sudo kubeadm init --pod-network-cidr=10.244.0.0/16`

```

ubuntu@ip-172-31-86-116:~$ sudo kubeadm init --pod-network-cidr=10.244.0.0/16
[init] Using Kubernetes version: v1.31.0
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action beforehand using 'kubeadm config images pull'
W0924 21:37:32.938984 4937 checks.go:846] detected that the sandbox image "registry.k8s.io/pause:3.8" of the containe
r runtime is inconsistent with that used by kubeadm.It is recommended to use "registry.k8s.io/pause:3.10" as the CRI san
dbox image.
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [ip-172-31-86-116 kubernet.es default kubernet.es.default
t.svc kubernet.es.default.svc.cluster.local] and IPs [10.96.0.1 172.31.86.116]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
[certs] Generating "etcd/server" certificate and key
[certs] etcd/server serving cert is signed for DNS names [ip-172-31-86-116 localhost] and IPs [172.31.86.116 127.0.0.1 :
:1]
[certs] Generating "etcd/peer" certificate and key

```

Copy the mkdir and chown commands from the top and execute them.

`mkdir -p $HOME/.kube`

`sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config`

`sudo chown $(id -u):$(id -g) $HOME/.kube/config`


```

ubuntu@ip-172-31-86-116:~$ mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config
ubuntu@ip-172-31-86-116:~$ kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
error: flag needs an argument: 'f' in -f
See 'kubectl apply --help' for usage.
-bash: https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml: No such file or directory
ubuntu@ip-172-31-86-116:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@ip-172-31-86-116:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-86-116:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-dmdsr    0/1     Pending   0           10s
nginx-deployment-d556bf558-kf9l4    0/1     Pending   0           10s
ubuntu@ip-172-31-86-116:~$
POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")

```

Add a common networking plugin called flannel as mentioned in the code.

`kubectl apply -f`

`https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml`

```

ubuntu@ip-172-31-86-116:~$ kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
error: flag needs an argument: 'f' in -f
See 'kubectl apply --help' for usage.
-bash: https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml: No such file or directory
ubuntu@ip-172-31-86-116:~$ kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
namespace/kube-flannel created
clusterrole.rbac.authorization.k8s.io/flannel created
clusterrolebinding.rbac.authorization.k8s.io/flannel created
serviceaccount/flannel created
configmap/kube-flannel-cfg created
daemonset.apps/kube-flannel-ds created
ubuntu@ip-172-31-86-116:~$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-86-116:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-d556bf558-dmdsr    0/1     Pending   0           10s
nginx-deployment-d556bf558-kf9l4    0/1     Pending   0           10s
ubuntu@ip-172-31-86-116:~$
POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")
kubectl port-forward $POD_NAME 8081:80
error: unable to forward port because pod is not running. Current status=Pending
ubuntu@ip-172-31-86-116:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untainted
error: at least one taint update is required
ubuntu@ip-172-31-86-116:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE     VERSION
ip-172-31-86-116                    Ready     control-plane  4m15s  v1.31.1
ubuntu@ip-172-31-86-116:~$

```

Step 7: Now that the cluster is up and running, we can deploy our nginx server on this cluster. Apply this deployment file using this command to create a deployment

`kubectl apply -f https://k8s.io/examples/application/deployment.yaml`


```
ubuntu@ip-172-31-86-116:~$  
POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")  
kubectl port-forward $POD_NAME 8081:80  
error: unable to forward port because pod is not running. Current status=Pending  
ubuntu@ip-172-31-86-116:~$ kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171 untainted  
error: at least one taint update is required  
ubuntu@ip-172-31-86-116:~$ kubectl get nodes  
NAME                STATUS    ROLES    AGE     VERSION  
ip-172-31-86-116    Ready     control-plane 4m15s   v1.31.1  
ubuntu@ip-172-31-86-116:~$ kubectl get pods  
NAME                READY    STATUS    RESTARTS   AGE  
nginx-deployment-d556bf558-dmdsr  0/1      Pending   0           2m52s  
nginx-deployment-d556bf558-kf9l4  0/1      Pending   0           2m52s  
ubuntu@ip-172-31-86-116:~$ kubectl get pods  
NAME                READY    STATUS    RESTARTS   AGE  
nginx-deployment-d556bf558-kf9l4  1/1      Running   0           6m29s  
nginx-deployment-d556bf558-dmdsr  1/1      Running   0           6m29s  
ubuntu@ip-172-31-86-116:~$ |
```

```
POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")  
kubectl port-forward $POD_NAME 8080:80
```

```
nginx-deployment-d556bf558-kf9l4 1/1 Running 0 6m29s  
ubuntu@ip-172-31-86-116:~$ POD_NAME=$(kubectl get pods -l app=nginx -o jsonpath="{.items[0].metadata.name}")  
kubectl port-forward $POD_NAME 8080:80  
Forwarding from 127.0.0.1:8080 -> 80  
Forwarding from [::]:8080 -> 80
```

```
kubectl taint nodes --all node-role.kubernetes.io/control-plane-node/ip-172-31-20-171  
untainted  
kubectl get nodes
```

```
Last login: Tue Sep 24 21:28:52 2024 from 103.88.83.126  
ubuntu@ip-172-31-86-116:~$ curl --head http://127.0.0.1:8080  
HTTP/1.1 200 OK  
Server: nginx/1.14.2  
Date: Tue, 24 Sep 2024 21:48:41 GMT  
Content-Type: text/html  
Content-Length: 612  
Last-Modified: Tue, 04 Dec 2018 14:44:49 GMT  
Connection: keep-alive  
ETag: "5c0692e1-264"  
Accept-Ranges: bytes  
ubuntu@ip-172-31-86-116:~$ |
```