

PRACTICAL NO: - 4**❏ Install Kubectl on Ubuntu****Installing Required Packages for HTTPS and Certificate Transport on Ubuntu**

```

ubuntu@ip-172-31-29-63: ~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1806 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2023 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [295 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [13.3 kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [2377 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [352 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [17.8 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2437 kB]
Get:19 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [409 kB]
Get:20 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [584 B]
Get:21 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [902 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [419 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [616 B]

```

```

ubuntu@ip-172-31-29-63: ~$ sudo apt-get install -y apt-transport-https ca-certificates curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
ca-certificates set to manually installed.
The following NEW packages will be installed:
  apt-transport-https
The following packages will be upgraded:
  curl libcurl4
2 upgraded, 1 newly installed, 0 to remove and 67 not upgraded.
Need to get 485 kB of archives.
After this operation, 170 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 apt-transport-https all 2.4.13 [1510 B]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 curl amd64 7.81.0-1ubuntu1.17 [194 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcurl4 amd64 7.81.0-1ubuntu1.17 [290 kB]
Fetched 485 kB in 0s (17.3 MB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 65320 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.4.13_all.deb ...
Unpacking apt-transport-https (2.4.13) ...
Preparing to unpack .../curl_7.81.0-1ubuntu1.17_amd64.deb ...
Unpacking curl (7.81.0-1ubuntu1.17) over (7.81.0-1ubuntu1.16) ...

```

Add the GPG key & repository for Kubernetes

```

ubuntu@ip-172-31-29-63: ~$ curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.29/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:/v1.29/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.29/deb/ /

```

Configure sysctl settings for Kubernetes networking

```

ubuntu@ip-172-31-29-63: ~$ sudo sysctl --system
* Applying /etc/sysctl.d/10-console-messages.conf ...
kernel.printk = 4 4 1 7
* Applying /etc/sysctl.d/10-ipv6-privacy.conf ...
net.ipv6.conf.all.use_tempaddr = 2
net.ipv6.conf.default.use_tempaddr = 2
* Applying /etc/sysctl.d/10-kernel-hardening.conf ...
kernel.kptr_restrict = 1
* Applying /etc/sysctl.d/10-magic-sysrq.conf ...
kernel.sysrq = 176
* Applying /etc/sysctl.d/10-network-security.conf ...
net.ipv4.conf.default.rp_filter = 2
net.ipv4.conf.all.rp_filter = 2
* Applying /etc/sysctl.d/10-ptrace.conf ...
kernel.yama.ptrace_scope = 1
* Applying /etc/sysctl.d/10-zero-page.conf ...
vm.mmap_min_addr = 65536
* Applying /etc/sysctl.d/50-cloudimg-settings.conf ...
net.ipv4.neigh.default.gc_thresh2 = 15360
net.ipv4.neigh.default.gc_thresh3 = 16384
net.netfilter.nf_conntrack_max = 1048576
* Applying /usr/lib/sysctl.d/50-default.conf ...
kernel.core_uses_pid = 1
net.ipv4.conf.default.rp_filter = 2

```

Install Kubernetes tools on all nodes.

```

ubuntu@ip-172-31-29-63:~$ sudo apt-get update -y
sudo apt-get install -y kubelet="1.29.0-*" kubectl="1.29.0-*" kubeadm="1.29.0-*"
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:5 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.29/deb InRelease [1189 B]
Get:6 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.29/deb Packages [14.0 kB]
Fetched 15.1 kB in 0s (31.9 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Selected version '1.29.0-1.1' (isv:kubernetes:core:stable:v1.29:pkgs.k8s.io [amd64]) for 'kubelet'
Selected version '1.29.0-1.1' (isv:kubernetes:core:stable:v1.29:pkgs.k8s.io [amd64]) for 'kubectl'
Selected version '1.29.0-1.1' (isv:kubernetes:core:stable:v1.29:pkgs.k8s.io [amd64]) for 'kubeadm'
The following additional packages will be installed:
  conntrack cri-tools ebtables kubernetes-cni socat
The following NEW packages will be installed:

```

□ Set up Kubernetes Cluster

If you haven't already set up a Kubernetes cluster (e.g., with kubeadm), use minikube or any managed Kubernetes service (like EKS, GKE, etc.) to get a cluster running.

Once your cluster is ready, verify the nodes: `kubectl get nodes`

```

ubuntu@ip-172-31-29-63:~$ kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
ip-172-31-20-115    Ready    <none>    27m   v1.29.0
ip-172-31-20-200    Ready    <none>    24m   v1.29.0
ip-172-31-29-63     Ready    control-plane 55m   v1.29.0
ubuntu@ip-172-31-29-63:~$

```

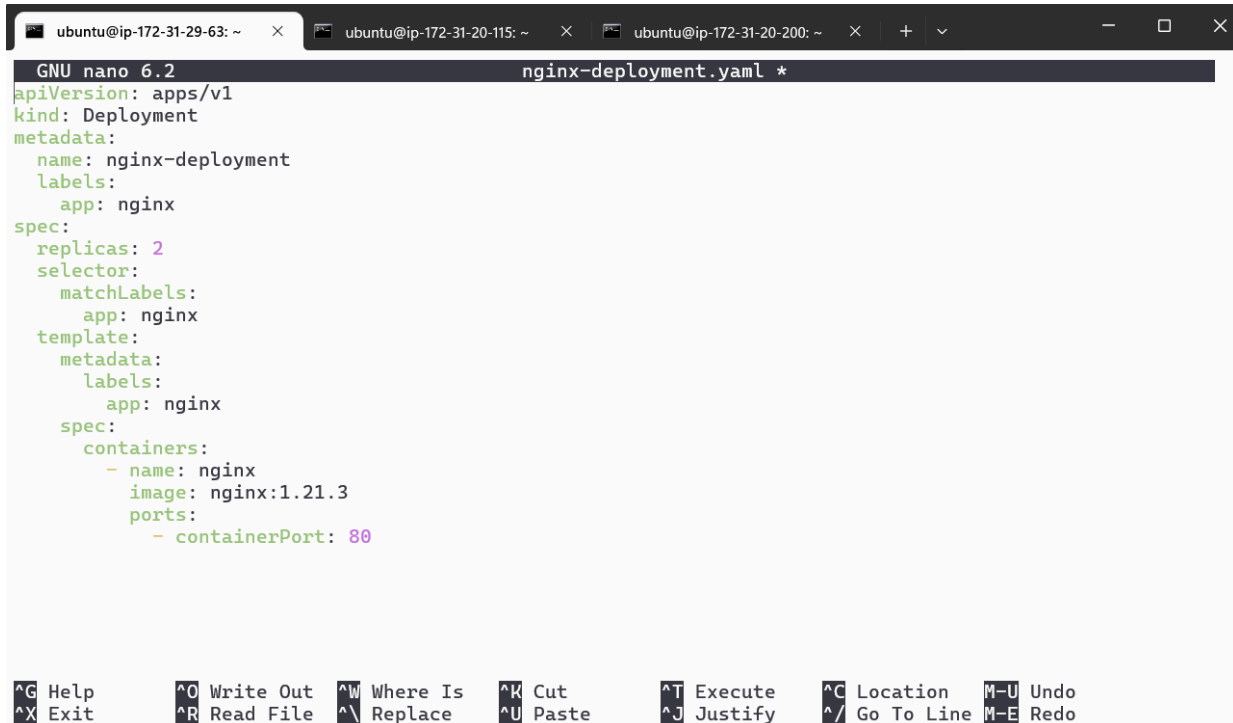
□ Deploying Your Application on Kubernetes

Create the Deployment YAML file

Create the YAML file: Use a text editor to create a file named `nginx-deployment.yaml`

```
ubuntu@ip-172-31-29-63:~$ nano nginx-deployment.yaml
ubuntu@ip-172-31-29-63:~$ |
```

Add the Deployment Configuration: Copy and paste the following YAML content into the file. Save and exit the editor (Press Ctrl+X, then Y, and Enter).



```
GNU nano 6.2 nginx-deployment.yaml *
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.21.3
          ports:
            - containerPort: 80

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line M-E Redo
```

Create the Service YAML File: Create another file named nginx-service.yaml

```
ubuntu@ip-172-31-29-63:~$ nano nginx-service.yaml
ubuntu@ip-172-31-29-63:~$ |
```

Add the Service Configuration: Copy and paste the following YAML content into the file given below.

```

ubuntu@ip-172-31-29-63: ~
ubuntu@ip-172-31-20-115: ~
ubuntu@ip-172-31-20-200: ~
GNU nano 6.2 nginx-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer

```

Deploy the Application: Use kubectl to create the Deployment and Service from the YAML files.

```

ubuntu@ip-172-31-29-63:~$ kubectl apply -f nginx-deployment.yaml --validate=false
deployment.apps/nginx-deployment created

```

```

ubuntu@ip-172-31-29-63:~$ kubectl apply -f nginx-service.yaml --validate=false
service/nginx-service created
ubuntu@ip-172-31-29-63:~$ |

```

Verify the Deployment: Check the status of your Deployment, Pods and Services

```

ubuntu@ip-172-31-29-63:~$ kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment    1/2     2            1           9m25s
ubuntu@ip-172-31-29-63:~$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6b4d6fdbf-6w4bm    1/1     Running   4 (98s ago) 9m18s
nginx-deployment-6b4d6fdbf-bhcwm    1/1     Running   4 (70s ago) 9m18s
ubuntu@ip-172-31-29-63:~$ kubectl get services
NAME            TYPE          CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes     ClusterIP     10.96.0.1       <none>        443/TCP          111m
nginx-service   LoadBalancer 10.110.88.111   <pending>     80:30132/TCP     110s
ubuntu@ip-172-31-29-63:~$ |

```

Describe the deployment

```

ubuntu@ip-172-31-29-63:~$ kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment 2/2     2            2           11m
ubuntu@ip-172-31-29-63:~$ kubectl describe deployment
Name:          nginx-deployment
Namespace:     default
CreationTimestamp: Sun, 15 Sep 2024 19:39:41 +0000
Labels:        app=nginx
Annotations:   deployment.kubernetes.io/revision: 1
Selector:      app=nginx
Replicas:      2 desired | 2 updated | 2 total | 1 available | 1 unavailable
StrategyType:  RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
    nginx:
      Image:      nginx:1.21.3
      Port:       80/TCP
      Host Port:  80/TCP
      Environment: <none>
      Mounts:      <none>
      Volumes:     <none>
Conditions:
  Type           Status  Reason
  ----           -
  Progressing    True    NewReplicaSetAvailable
  Available      False   MinimumReplicasUnavailable
OldReplicaSets: <none>
NewReplicaSet:  nginx-deployment-6b4d6fdbf (2/2 replicas created)
Events:
  Type      Reason              Age   From               Message
  ----      -
  Normal    ScalingReplicaSet   11m   deployment-controller  Scaled up replica set nginx-deployment-6b4d6fdbf to 2
ubuntu@ip-172-31-29-63:~$

```

Verify Service: Run the following command to check the services running in your cluster:

```

ubuntu@ip-172-31-29-63:~$ kubectl get service
NAME          TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes    ClusterIP     10.96.0.1       <none>           443/TCP          114m
nginx-service  LoadBalancer 10.110.88.111   <pending>        80:30132/TCP     4m59s
ubuntu@ip-172-31-29-63:~$

```

Forward the Service Port to Your Local Machine

```

ubuntu@ip-172-31-45-227:~$ kubectl port-forward service/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80

```

```

ubuntu@ip-172-31-45-227:~$ kubectl port-forward service/nginx-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
^Cubuntu@ip-172-31-45-227:~$ kubectl port-forward service/nginx-service 8081:8080
Forwarding from 127.0.0.1:8081 -> 80
Forwarding from [::1]:8081 -> 80
^Cubuntu@ip-172-31-45-227:~$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-deployment-776b8fd845-k9cx4 1/1     Running   0           113m
ubuntu@ip-172-31-45-227:~$ kubectl logs nginx-deployment-776b8fd845-k9cx4
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/09/12 06:35:51 [notice] 1#1: using the "epoll" event method
2024/09/12 06:35:51 [notice] 1#1: nginx/1.27.1
2024/09/12 06:35:51 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/09/12 06:35:51 [notice] 1#1: OS: Linux 6.5.0-1022-aws
2024/09/12 06:35:51 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/09/12 06:35:51 [notice] 1#1: start worker processes
2024/09/12 06:35:51 [notice] 1#1: start worker process 24
2024/09/12 06:35:51 [notice] 1#1: start worker process 25

```

Access the Application Locally

Open a Web Browser: Now open your web browser and go to the following URL: <http://localhost:8080>

You should see the application (in this case, Nginx) that you have deployed running in the Kubernetes cluster, served locally via port 8080.

