Name: Snehal Patil D15A 39

ADVANCE DEVOPS EXP: 4

Aim: To install Kubectl and execute Kubectl commands to manage the Kubernetes clusterand deploy Your First Kubernetes Application.

Theory:

What is kubectl?

kubectl is the command-line interface (CLI) used to interact with a Kubernetes cluster. It allows users to manage cluster resources, deploy applications, inspect and manage cluster components, and much more. Using kubectl, you can communicate with the Kubernetes API server to issue commands and queries.

Common kubectl commands:

- kubectl get: View information about resources.
- kubectl describe: Detailed description of resources.
- kubectl create/apply: Create or update resources.
- kubectl delete: Delete resources.

kubectl plays a crucial role in the day-to-day operation of a Kubernetes cluster.

Basic Concepts in Kubernetes

Before diving into the application deployment process, it's important to understand a few key Kubernetes objects:

1. **Pods**: The smallest deployable unit in Kubernetes. A pod encapsulates one or more containers (usually a single container) that share the same network namespace and storage.
2. **Deployments**: A Kubernetes resource that defines how to create and manage pods. It ensures the specified number of pod replicas are running at any given time and handles updates and rollbacks.
3. **Services**: An abstraction that defines how to access the pods. A service allows you to expose your pods to internal or external clients.
4. **ReplicaSets**: Ensures that a specified number of pod replicas are running at all times. It is managed by a Deployment, but can also be used independently.

1.1 Install prerequisites:

sudo apt-get update

sudo apt-get install -y apt-transport-https ca-certificates curl

```
root@ip-172-31-87-198:/home/ubuntu# sudo apt-get update -y
sudo apt-get install -y software-properties-common curl apt-transport-https ca-c
ertificates gpg

sudo curl -fsSL https://pkgs.k8s.io/addons:/cri-o:/prerelease:/main/deb/Release.
key | sudo gpg --dearmor -o /etc/apt/keyrings/cri-o-apt-keyring.gpg
echo "deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.i
o/addons:/cri-o:/prerelease:/main/deb/ /" | sudo tee /etc/apt/sources.list.d/cri
-o.list

sudo apt-get update -y
sudo apt-get install -y cri-o
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:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri
-o:/prerelease:/main/deb  InRelease
Ign:5 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Err:7 https://packages.cloud.google.com/apt kubernetes-focal Release
  404  Not Found [IP: 172.253.122.100 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not
have a Release file.
N: Updating from such a repository can't be done securely, and is therefore disa
bled by default.
N: See apt-secure(8) manpage for repository creation and user configuration deta
ils.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.48).
curl is already the newest version (8.5.0-2ubuntu10.4).
apt-transport-https is already the newest version (2.7.14build2).
ca-certificates is already the newest version (20240203).
gpg is already the newest version (2.4.4-2ubuntu17).
0 upgraded, 0 newly installed, 0 to remove and 130 not upgraded.
File '/etc/apt/keyrings/cri-o-apt-keyring.gpg' exists. Overwrite? (y/N) y
deb [signed-by=/etc/apt/keyrings/cri-o-apt-keyring.gpg] https://pkgs.k8s.io/addo
ns:/cri-o:/prerelease:/main/deb/ /
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
```

1. **Add the GPG key for Kubernetes:**

sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg

```
root@ip-172-31-87-198:/home/ubuntu# sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-ke
https://packages.cloud.google.com/apt/doc/apt-key.gpg
curl: (2) no URL specified
curl: try 'curl --help' or 'curl --manual' for more information
hash: https://packages.cloud.google.com/apt/doc/apt-key.gpg: No such file or directory
```

2. **Add the Kubernetes repository:**

echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-focal main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

```
root@ip-172-31-87-198:/home/ubuntu# echo "deb [signed-by=/usr/share/keyrings/kubernetes-archiv
https://apt.kubernetes.io/ kubernetes-focal main" | sudo tee
/etc/apt/sources.list.d/kubernetes.list
deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg]
https://apt.kubernetes.io/ kubernetes-focal main
```

1.2 Install kubectl:

Now install kubectl

Sudo apt-get update

Sudo apt-get install –y kubectl

```
root@ip-172-31-87-198:/home/ubuntu# sudo apt-get update
sudo apt-get install -y 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://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/addons:/cri-o:/prerelease
Ign:6 https://packages.cloud.google.com/apt kubernetes-focal InRelease
Err:7 https://packages.cloud.google.com/apt kubernetes-focal Release
  404  Not Found [IP: 172.253.122.102 443]
Reading package lists... Done
E: The repository 'https://apt.kubernetes.io kubernetes-focal Release' does not have a Release
N: Updating from such a repository can't be done securely, and is therefore disabled by defaul
N: See apt-secure(8) manpage for repository creation and user configuration details.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
kubectl is already the newest version (1.29.0-1.1).
0 upgraded, 0 newly installed, 0 to remove and 130 not upgraded.
```

```
root@ip-172-31-87-198:/home/ubuntu# nano nginx-deployment.yaml
root@ip-172-31-87-198:/home/ubuntu# nano nginx-service.yaml
```

Verifying the installation:

Kubectl version --client

```
root@ip-172-31-87-198:/home/ubuntu# kubectl version --client
Client Version: v1.29.0
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
```

Step 2: Deploying the Application on Kubernetes

2.1 Setting up Kubernetes Cluster

1. 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 acluster running.
2. Once your cluster is ready,confirm that all the nodes

   are successfully connected and operational.

   Command: kubectl get nodes

```
root@ip-172-31-87-198:/home/ubuntu# kubectl get nodes
NAME               STATUS   ROLES           AGE   VERSION
ip-172-31-80-64    Ready    <none>          18s   v1.29.0
ip-172-31-81-208   Ready    <none>          35s   v1.29.0
ip-172-31-87-198   Ready    control-plane   43m   v1.29.0
```

Step 3:**Create the Deployment YAML file**

a)Creating the YAML file: Use a text editor to create a file named nginx-deployment.yaml And nginx-service.yaml

```
root@ip-172-31-87-198:/home/ubuntu# nano nginx-deployment.yaml
root@ip-172-31-87-198:/home/ubuntu# nano nginx-service.yaml
root@ip-172-31-87-198:/home/ubuntu# kubectl apply of nginx-deployment.yaml
error: Unexpected args: [of nginx-deployment.yaml]
See 'kubectl apply -h' for help and examples
root@ip-172-31-87-198:/home/ubuntu# kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
root@ip-172-31-87-198:/home/ubuntu# kubectl apply -f nginx-service.yaml
service/nginx-service created
```

b)Adding the Deployment Configuration to nginx-deployment.yaml and nginx-service.yaml

root@ip-172-31-87-198: /home/ubuntu

```
  GNU nano 7.2                    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
                          [ Read 22 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

root@ip-172-31-87-198: /home/ubuntu

```
  GNU nano 7.2                    nginx-service.yaml
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer
```

Step 4:Applying the YAML Files

a) Deploying the Application: Use kubectl to create the Deployment and Service from the YAML files.

```
root@ip-172-31-87-198:/home/ubuntu# kubectl apply -f nginx-deployment.yaml
deployment.apps/nginx-deployment created
root@ip-172-31-87-198:/home/ubuntu# kubectl apply -f nginx-service.yaml
service/nginx-service created
```

Verifying the Deployment and also describing the deployment:

Check the status of your Deployment,Pods and Services.

```
root@ip-172-31-87-198:/home/ubuntu# kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           5m57s
root@ip-172-31-87-198:/home/ubuntu# kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           6m39s
```

```
root@ip-172-31-87-198:/home/ubuntu# kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   2/2     2            2           28m
root@ip-172-31-87-198:/home/ubuntu# kubectl describe deployment
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Wed, 18 Sep 2024 12:14:59 +0000
Labels:                 app=nginx
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx
Replicas:               2 desired | 2 updated | 2 total | 2 available | 0 unavai
lable
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:    0/TCP
    Environment:  <none>
```

```
root@ip-172-31-87-198:/home/ubuntu# kubectl get service
NAME            TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP      10.96.0.1        <none>        443/TCP        55m
nginx-service   LoadBalancer   10.109.148.186   <pending>     80:30162/TCP   31m
root@ip-172-31-87-198:/home/ubuntu#
```

Step 6: Ensure Service is Running

6.1 **Verify Service**: Running the following commands to check the services running in our cluster:

Command: kubectl get service

```
root@ip-172-31-87-198:/home/ubuntu# kubectl get service
NAME            TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE
kubernetes      ClusterIP      10.96.0.1      <none>        443/TCP        101m
nginx-service   LoadBalancer   10.106.17.37   <pending>     80:31687/TCP   105s
root@ip-172-31-87-198:/home/ubuntu# kubectl port-forward service/nginx-service 8
080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
^Croot@ip-172-31-87-198:/home/ubuntu# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6b4d6fdbf-n52mq    1/1     Running   0          9m20s
nginx-deployment-6b4d6fdbf-w9qjv    1/1     Running   0          9m20s
root@ip-172-31-87-198:/home/ubuntu# kubectl logs nginx-deployment-6b4d6fdbf-n52m
q
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perfo
rm configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-defau
lt.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: Launching /docker-entrypoint.d/20-envsubst-on-templates.s
h
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.s
h
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/09/18 11:34:24 [notice] 1#1: using the "epoll" event method
2024/09/18 11:34:24 [notice] 1#1: nginx/1.21.3
2024/09/18 11:34:24 [notice] 1#1: built by gcc 8.3.0 (Debian 8.3.0-6)
2024/09/18 11:34:24 [notice] 1#1: OS: Linux 6.8.0-1012-aws
2024/09/18 11:34:24 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/09/18 11:34:24 [notice] 1#1: start worker processes
2024/09/18 11:34:24 [notice] 1#1: start worker process 26
2024/09/18 11:34:24 [notice] 1#1: start worker process 27
```

```
root@ip-172-31-87-198:/home/ubuntu# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6b4d6fdbf-bqqg2    1/1     Running   0          35m
nginx-deployment-6b4d6fdbf-ptgmg    1/1     Running   0          35m
```

Step 7: Forward the Service Port to Your Local Machine

kubectl port-forward allows you to forward a port from your local machine to a port on a service running in the Kubernetes cluster.

Command:

kubectl port-forward service/<service-name> <local-port>:<service-port>

```
root@ip-172-31-87-198:/home/ubuntu# kubectl port-forward service/nginx-service 8
080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

```
root@ip-172-31-87-198:/home/ubuntu# kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6b4d6fdbf-bqqg2    1/1     Running   0          35m
nginx-deployment-6b4d6fdbf-ptgmg    1/1     Running   0          35m
root@ip-172-31-87-198:/home/ubuntu# kubectl logs nginx-deployment-6b4d6fdbf-bqqg
2
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perfo
rm configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-defau
lt.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: Launching /docker-entrypoint.d/20-envsubst-on-templates.s
h
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.s
h
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/09/18 12:34:16 [notice] 1#1: using the "epoll" event method
2024/09/18 12:34:16 [notice] 1#1: nginx/1.21.3
2024/09/18 12:34:16 [notice] 1#1: built by gcc 8.3.0 (Debian 8.3.0-6)
2024/09/18 12:34:16 [notice] 1#1: OS: Linux 6.8.0-1012-aws
2024/09/18 12:34:16 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/09/18 12:34:16 [notice] 1#1: start worker processes
2024/09/18 12:34:16 [notice] 1#1: start worker process 26
2024/09/18 12:34:16 [notice] 1#1: start worker process 27
```

Step 8: Access the Application

- Open a web browser and navigate to `http://<Node-IP>:<Port>`. You should see the NGINX application running in the Kubernetes cluster.