

Task 20: Pods and Services in Kubernetes Deployment

Connect an instance install awscli

▼ 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
aws


macOS
Mac



Ubuntu
ubuntu

Windows
Microsoft

Red Hat
Red Hat

SUSE Linux
SUSE


Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Instances (1/1) Info									
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/> All states ▼									
<input type="text" value="Instance ID = i-0e4527411d99bd7dd"/> Clear filters < 1 > ⚙									
<input checked="" type="checkbox"/>	Name ↗	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP	
<input checked="" type="checkbox"/>	kubernetes	i-0e4527411d99bd7dd	Running 	t2.micro	Initializing 	View alarms +	us-east-1b	ec2-3-85	

```
ubuntu@ip-172-31-16-51:~$ aws --version
aws-cli/2.17.24 Python/3.11.9 Linux/6.8.0-1009-aws exe/x86_64.ubuntu.24
```

```
ubuntu@ip-172-31-16-51:~$ aws configure
AWS Access Key ID [None]: AKIASEMTU4DA3V62INAB
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
```

1. Install kubectl

```
ubuntu@ip-172-31-16-51:~$ curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectl
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 57.4M 100 57.4M 0 0 8852k 0 0:00:06 0:00:06 --:--:-- 12.4M
ubuntu@ip-172-31-16-51:~$ chmod +x ./kubectl
ubuntu@ip-172-31-16-51:~$ sudo mv ./kubectl /usr/local/bin
ubuntu@ip-172-31-16-51:~$ kubectl version --short -client
Client Version: v1.19.6-eks-49a6c0
```

2. Install eksctl:

```
ubuntu@ip-172-31-16-51:~$ curl --silent --location
sudo mv /tmp/eksctl /usr/local/bin
eksctl version
0.188.0
```

Create a Kubernetes cluster

```
ubuntu@ip-172-31-16-51:~$ eksctl create cluster --name eks \
  --region us-east-1 \
  --node-type t2.small
```

```
2024-08-07 11:32:42 [✓] all EKS cluster resources for "eks" have been created
2024-08-07 11:32:42 [✓] created 0 nodegroup(s) in cluster "eks"
2024-08-07 11:32:42 [i] nodegroup "ng-75555f69" has 2 node(s)
2024-08-07 11:32:42 [i] node "ip-192-168-3-43.ec2.internal" is ready
2024-08-07 11:32:42 [i] node "ip-192-168-34-6.ec2.internal" is ready
2024-08-07 11:32:42 [i] waiting for at least 2 node(s) to become ready in "ng-75555f69"
2024-08-07 11:32:42 [i] nodegroup "ng-75555f69" has 2 node(s)
2024-08-07 11:32:42 [i] node "ip-192-168-3-43.ec2.internal" is ready
2024-08-07 11:32:42 [i] node "ip-192-168-34-6.ec2.internal" is ready
2024-08-07 11:32:42 [✓] created 1 managed nodegroup(s) in cluster "eks"
2024-08-07 11:32:43 [i] kubectl command should work with "/home/ubuntu/.kube/config", try 'kubectl get nodes'
2024-08-07 11:32:43 [✓] EKS cluster "eks" in "us-east-1" region is ready
```

```
ubuntu@ip-172-31-16-51:~$ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
ip-192-168-3-43.ec2.internal	Ready	<none>	3m8s	v1.30.2-eks-1552ad0
ip-192-168-34-6.ec2.internal	Ready	<none>	2m53s	v1.30.2-eks-1552ad0

Instances (1/4) <small>Info</small>								
Find Instance by attribute or tag (case-sensitive)								
All states ▾								
<input type="checkbox"/>	Name ↗	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone ▾	
<input type="checkbox"/>	kubernetes	i-0e4527411d99bd7dd	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	
<input type="checkbox"/>	eks-ng-75555f69-Node	i-0822abc8f39d81000	Running	t2.small	Initializing	View alarms +	us-east-1a	
<input checked="" type="checkbox"/>	Jenkins	i-0c23bcc546a66e06f	Stopped	t2.micro	-	View alarms +	us-east-1a	
<input type="checkbox"/>	eks-ng-75555f69-Node	i-0bb8f919e94878d1d	Running	t2.small	Initializing	View alarms +	us-east-1d	

1. Deployment file:

vi deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: web
  template:
    metadata:
      labels:
        app: web
    spec:
      containers:
        - name: nginx-container
          image: nginx:latest
          ports:
            - containerPort: 80
```

```

ubuntu@ip-172-31-16-51:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE      VERSION
ip-192-168-3-43.ec2.internal        Ready    <none>   3m8s    v1.30.2-eks-1552ad0
ip-192-168-34-6.ec2.internal        Ready    <none>   2m53s    v1.30.2-eks-1552ad0
ubuntu@ip-172-31-16-51:~$ vi deployment.yaml
ubuntu@ip-172-31-16-51:~$ kubectl apply -f deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@ip-172-31-16-51:~$ kubectl get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/nginx-deployment-7656c686d6-8dvlz  1/1      Running    0           47s
pod/nginx-deployment-7656c686d6-rqskn  1/1      Running    0           47s
pod/nginx-deployment-7656c686d6-xjhzq  1/1      Running    0           47s

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/kubernetes                  ClusterIP      10.100.0.1     <none>         443/TCP    14m

NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/nginx-deployment    3/3      3              3            48s

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/nginx-deployment-7656c686d6  3          3          3        48s

```

Execute deployment.yaml file:

kubectl apply -f deployment.yaml

2. Service.yaml file:

vi service.yaml

```

apiVersion: v1
kind: Service
metadata:
  name: nginx-svc
spec:
  selector:
    app: web
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
  type: LoadBalancer

```

```

ubuntu@ip-172-31-16-51:~$ vi service.yaml
ubuntu@ip-172-31-16-51:~$ kubectl apply -f service.yaml
service/nginx-svc created
ubuntu@ip-172-31-16-51:~$ kubectl get services
NAME      TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
kubernetes ClusterIP      10.100.0.1     <none>         443/TCP    17m
nginx-svc LoadBalancer  10.100.70.29   all120f34822884b4f9e0bf5f0edd003-718690748.us-east-1.elb.amazonaws.com 80:31963/TCP 22s
ubuntu@ip-172-31-16-51:~$ kubectl get all
NAME                                READY    STATUS    RESTARTS   AGE
pod/nginx-deployment-7656c686d6-8dvlz  1/1      Running    0           3m25s
pod/nginx-deployment-7656c686d6-rqskn  1/1      Running    0           3m25s
pod/nginx-deployment-7656c686d6-xjhzq  1/1      Running    0           3m25s

NAME                                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
service/kubernetes                  ClusterIP      10.100.0.1     <none>         443/TCP    17m
service/nginx-svc                  LoadBalancer  10.100.70.29   all120f34822884b4f9e0bf5f0edd003-718690748.us-east-1.elb.amazonaws.com 80:31963/TCP 40s


NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
deployment.apps/nginx-deployment    3/3      3              3            3m25s

NAME                                DESIRED    CURRENT    READY    AGE
replicaset.apps/nginx-deployment-7656c686d6  3          3          3        3m25s

```

Execute service.yaml file: kubectl apply -f service.yaml

⚠ Not secure a1120f34822884b4f9e0bfb5f0edd003-718690748.us-east-1.elb.amazonaws.com

cul...  minikube start | min...

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

3. Pod.yaml file:

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
    tier: dev
spec:
  containers:
  - name: nginx-container
    image: nginx
```

```
ubuntu@ip-172-31-16-51:~$ vi pod.yaml
ubuntu@ip-172-31-16-51:~$ kubectl apply -f pod.yaml
pod/nginx-pod created
```

```
ubuntu@ip-172-31-16-51:~$ kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-7656c686d6-8dvlz  1/1     Running   0           6m57s
pod/nginx-deployment-7656c686d6-rqskn  1/1     Running   0           6m57s
pod/nginx-deployment-7656c686d6-xjhzq  1/1     Running   0           6m57s
pod/nginx-pod                          1/1     Running   0           25s

NAME                                TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
service/kubernetes                  ClusterIP     10.100.0.1    <none>         443/TCP          20m
service/nginx-svc                   LoadBalancer 10.100.70.29 a1120f34822884b4f9e0bfb5f0edd003-718690748.us-east-1.elb.amazonaws.com 80:31963/TCP    4m12s

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment    3/3     3             3           6m57s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deployment-7656c686d6  3         3         3       6m57s
ubuntu@ip-172-31-16-51:~$
```

Execute pod.yaml file:

kubectl apply -f pod.yaml