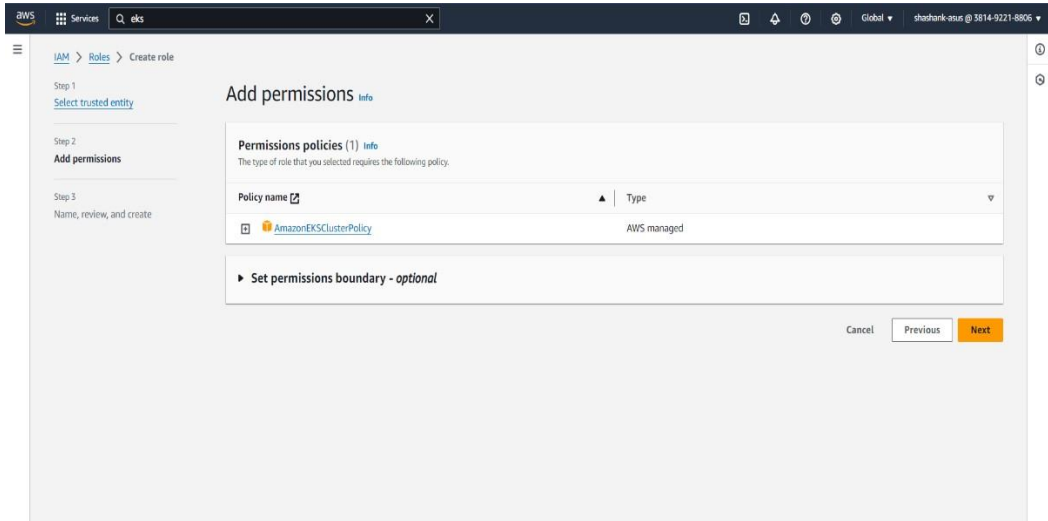


CDEC B24

Name: Kunal vijay Chavhan

Task: Hosting of nginx and tomcat using manifest file

1. Create IAM role for EKS and give EKS permission



2. Create IAM role for EC2 and give permission as below.

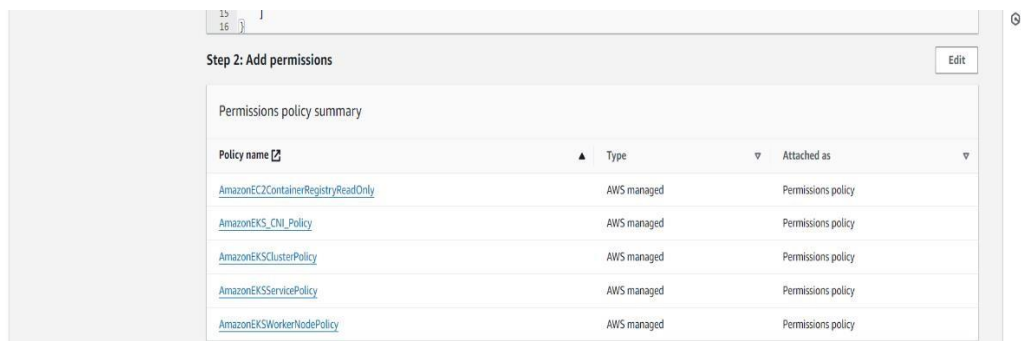
AmazonEC2ContainerRegistryReadOnly

AmazonEKS_CNI_Policy

AmazonEKSClusterPolicy

AmazonEKSServicePolicy

AmazonEKSWorkerNodePolicy



3. Create a cluster.

Extended support for Kubernetes versions pricing

New prices for extended support will start in the April billing cycle. For more information, see the [blog post](#).

EKS > Clusters > Create EKS cluster

Step 1

Configure cluster

Step 2

Specify networking

Step 3

Configure observability

Step 4

Select add-ons

Step 5

Configure selected add-ons settings

Step 6

Review and create

Review and create

Step 1: Cluster

Edit

Cluster configuration

Name	shashank-eks-cluster	Kubernetes version	1.29
Cluster service role	arn:aws:iam::381492218806:role/shashank-eks-role	Kubernetes cluster administrator access	Allow cluster administrator access
Authentication mode	EKS API and ConfigMap		

Tags (0)

Tags that you've added. Each tag consists of a key and an optional value.

< 1 >

Key	Value
No tags	
This cluster does not have any tags.	

Step 2: Networking

Edit

Networking

These properties cannot be changed after the cluster is created.

VPC	Subnets	Security groups
vpc-0f692367e7315726d	subnet-0b975bf6e85fbfeac subnet-04a74a5846e4c229c	sg-00347fdf46666f2e2
Cluster IP address family	IPv4	

Cluster endpoint access

API server endpoint access	Public access source allowlist
Public	0.0.0.0/0

Step 3: Observability

Edit

Control plane logging

API server	Audit	Authenticator
off	off	off
Controller manager	Scheduler	
off	off	

Step 4: Add-ons

Edit

Selected add-ons

Find add-on

< 1 >

Add-on name	Type	Status
coredns	networking	Installed by default
eks-pod-identity-agent	security	Ready to install
kube-proxy	networking	Installed by default
vpc-cni	networking	Installed by default

Step 5: Versions

Edit

Selected add-ons version

Add-on name	Version
coredns	v1.11.1-eksbuild.4
Add-on name	Version
kube-proxy	v1.29.0-eksbuild.1
Add-on name	Version
vpc-cni	v1.16.0-eksbuild.1
Add-on name	Version
eks-pod-identity-agent	v1.2.0-eksbuild.1

Cancel

Previous

Create

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

4. After creation of cluster add node group to it.

The screenshot displays the AWS Management Console interface for configuring a new node group. The breadcrumb navigation at the top shows the path: **EKS** > **Clusters** > **shashank-eks-cluster** > **Node groups** > **Add node group**. The page is divided into four steps: Step 1 (Configure node group), Step 2 (Set compute and scaling configuration), Step 3 (Specify networking), and Step 4 (Review and create). Step 2 is the active step.

Set compute and scaling configuration

Node group compute configuration
These properties cannot be changed after the node group is created.

AMI type [info](#)
Select the EKS-optimized Amazon Machine Image for nodes.
Amazon Linux 2 (AL2_x86_64)

Capacity type
Select the capacity purchase option for this node group.
On-Demand

Instance types [info](#)
Select instance types you prefer for this node group.
Enter an instance type:
t3.medium
vCPU: 2 vCPUs Memory: 4 GiB Network: Up to 5 Gbps Max ENI: 3 Max IP: 18

Disk size
Select the size of the attached EBS volume for each node.
20 GiB

Node group scaling configuration

Desired size
Set the desired number of nodes that the group should launch with initially.
1 nodes
Desired node size must be greater than or equal to 0

Minimum size
Set the minimum number of nodes that the group can scale in to.
1 nodes
Minimum node size must be greater than or equal to 0

Maximum size
Set the maximum number of nodes that the group can scale out to.
2 nodes
Maximum node size must be greater than or equal to 1 and cannot be lower than the minimum size

Node group update configuration [info](#)

Maximum unavailable
Set the maximum number or percentage of unavailable nodes to be tolerated during the node group version update.

☒ **Number**
Enter a number

☐ **Percentage**
Specify a percentage

Value
1 node
Node count must be greater than 0.

Cancel Previous **Next**

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

5. After adding node group to the cluster open cloud shell and configure it using command.

aws configure

(add your access key, secret access key, region)

```
[cloudshell-user@ip-10-140-121-42 ~]$ aws configure
AWS Access Key ID [None]: AKIA3MEFHRQULL6Z
AWS Secret Access Key [None]: C1Qb7MTX0W9d6
Default region name [None]:
Default output format [None]:
[cloudshell-user@ip-10-140-121-42 ~]$ kubectl cluster-info
-bash: kubectl: command not found
[cloudshell-user@ip-10-140-121-42 ~]$ kubectl cluster-info
-bash: kubectl: command not found
[cloudshell-user@ip-10-140-121-42 ~]$ kubectl cluster-info
E0329 05:53:19.209542    234 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080: connect: connection refused
E0329 05:53:19.211173    234 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080: connect: connection refused
E0329 05:53:19.211858    234 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080: connect: connection refused
E0329 05:53:19.216686    234 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080: connect: connection refused
E0329 05:53:19.219099    234 memcache.go:265] couldn't get current server API group list: Get "http://localhost:8080/api?timeout=32s": dial tcp 127.0.0.1:8080: connect: connection refused

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
The connection to the server localhost:8080 was refused - did you specify the right host or port?
[cloudshell-user@ip-10-140-121-42 ~]$ aws eks --region us-east-1 update-kubeconfig --name EKS-cluster
Added new context arn:aws:eks:us-east-1:471112957025:cluster/EKS-cluster to /home/cloudshell-user/.kube/config
[cloudshell-user@ip-10-140-121-42 ~]$ kubectl cluster-info
Kubernetes control plane is running at https://9EBF926C1E88FCCC100B88848E29A25C.gr7.us-east-1.eks.amazonaws.com
CoreDNS is running at https://9EBF926C1E88FCCC100B88848E29A25C.gr7.us-east-1.eks.amazonaws.com/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

6. Create pod.yml & service.yml file in your VS code and upload files on your git repo.

7. Create pod file for nginx and tomcat with extension pod.yml.

```
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: ompod
5    labels:|
6    app: new-app
7  spec:
8    containers:
9    - name: nginx
10      image: nginx:latest
11      ports:
12      - containerPort: 80
13        protocol: TCP
14    - name: tomcat
15      image: tomcat:latest
16      ports:
17      - containerPort: 8080
18        protocol: TCP
```

8. Create service file for nginx and tomcat with extension service.yml.

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: nodesvc
5  spec:
6    selector:
7      app: new-app
8    type: NodePort
9    ports:
10     - protocol: TCP
11       port: 80
12       targetPort: 80
13       name: nginx
14
15     - protocol: TCP
16       port: 8080
17       targetPort: 8080
18       name: tomcat
```

8. After completing the script create pod using command.

```
#git clone <your-git-repo-link>
```

```
#git clone https://github.com/Nirmalomkar/Kubernetes.git
```

```
(in my case my file present in Kubernetes repo.)
```

```
#ls
```

```
 #(goes upto your pod.yml file for creation of node)
```

```
#kubectl apply -f pods.yml
```

```
#kubectl get pods
```

```
#kubectl get -o wide
```

```
pods
```

```
#kubectl describe pods
```

```
#kubectl apply -f service.yml
```

```
#kubectl get srv (services)
```

```
[cloudshell-user@ip-10-136-34-135 ~]$ git clone https://github.com/chavhankunal/Kubernetes.git
Cloning into 'Kubernetes'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), done.
[cloudshell-user@ip-10-136-34-135 ~]$ ls
Kubernetes
[cloudshell-user@ip-10-136-34-135 ~]$ cd Kubernetes/
[cloudshell-user@ip-10-136-34-135 Kubernetes]$ ls
pod.yml  README.md  service.yml
[cloudshell-user@ip-10-136-34-135 Kubernetes]$ kubectl apply -f pod.yml
pod/ompod created
[cloudshell-user@ip-10-136-34-135 Kubernetes]$ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
nginx     1/1     Running   0           68m
ompod     2/2     Running   0           24s
tomcat    1/1     Running   0           57m
[cloudshell-user@ip-10-136-34-135 Kubernetes]$ kubectl apply -f service.yml
service/nodesvc created
[cloudshell-user@ip-10-136-34-135 Kubernetes]$ kubectl get svc
NAME      TYPE        CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
Kubernetes ClusterIP  10.100.0.1       <none>            443/TCP          105m
nginx     NodePort    10.100.71.182    <none>            80:32660/TCP     67m
nodesvc   NodePort    10.100.288.242   <none>            80:32359/TCP,8080:32021/TCP 8s
tomcat    NodePort    10.100.229.66    <none>            8080:30610/TCP   57m
[cloudshell-user@ip-10-136-34-135 Kubernetes]$ kubectl get -o wide node
NAME      STATUS    ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE      KERNEL-VERSION   CONTAINER-RUNTIME
ip-172-31-85-108.ec2.internal Ready      <none>    99m   v1.29.0-eks-5e0fdde  172.31.85.108  3.95.221.247  Amazon Linux 2  5.10.210-201.852.amzn2.x86_64  containerd://1.7.11
```

9. After creation of service hit the IP of your instance which is created while creation of node group.

For nginx <instance-IP>:<port-IP>

In my case; <http://3.95.221.247:32359/>



10. For tomcat;
<http://3.95.221.247:32021/>

