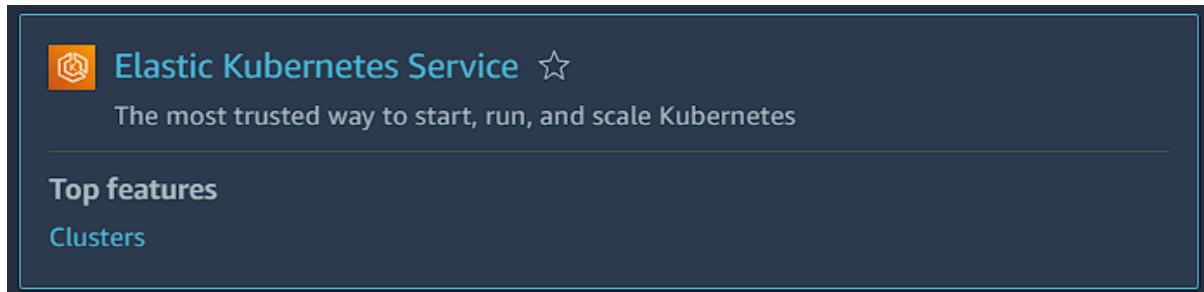
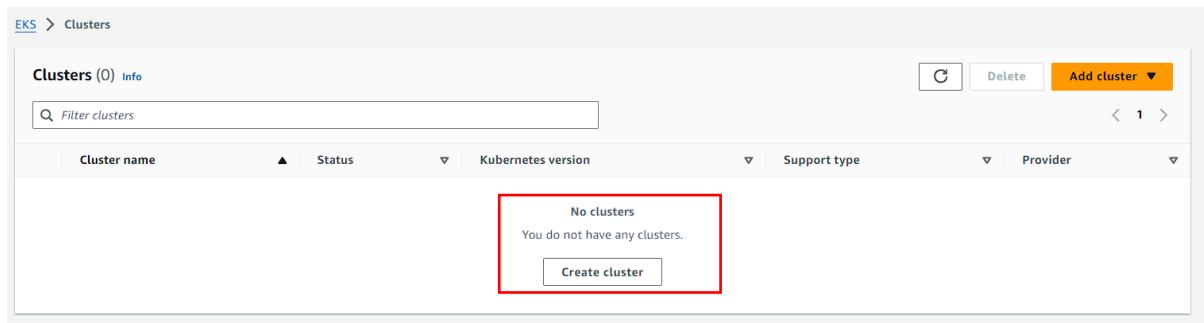


Elastic Kubernetes Service

1. In this lab you are going to learn about EKS hands on.
2. First search EKS on AWS Console and navigate to it. Choose this service accordingly.



3. Here you need to create a cluster. So, click on create cluster.



4. Now give it a name then choose the version. Now we need to select an IAM role that will allow the control plane of Kubernetes to manage the AWS resources on your behalf.

Cluster configuration Info

Name

Enter a unique name for this cluster. This property cannot be changed after the cluster is created.

The cluster name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 100.

Kubernetes version Info

Select Kubernetes version for this cluster.



i Kubernetes version 1.27 reaches the end of standard support on July 24, 2024. If you don't update your cluster to a later version before that date, it will automatically enter extended support. After the extended support preview ends, clusters on versions in extended support will be subject to additional fees. [Learn more](#).

Cluster service role Info

Select the IAM role to allow the Kubernetes control plane to manage AWS resources on your behalf. This property cannot be changed after the cluster is created. To create a new role, follow the instructions in the [Amazon EKS User Guide](#).



5. Now navigate to IAM in another tab and there go to roles and click on create roles.
6. Here you need to select service as EKS and use case as EKS Cluster.

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case



Choose a use case for the specified service.

Use case

 EKS

Allows EKS to manage clusters on your behalf.

 EKS - Cluster

Allows access to other AWS service resources that are required to operate clusters managed by EKS.

7. Now the permission will be automatically attached to it. Then just create your role.

Add permissions Info

Permissions policies (1) Info

The type of role that you selected requires the following policy.

Policy name

▲ | Type

 AmazonEKSClusterPolicy

AWS managed

► Set permissions boundary - *optional*

[Cancel](#)

[Previous](#)

[Next](#)

8. After that come back to EKS and select your role.

Cluster service role Info

Select the IAM role to allow the Kubernetes control plane to manage AWS resources on your behalf. This property cannot be changed after the cluster is created. To create a new role, follow the instructions in the [Amazon EKS User Guide](#).

EKS-cluster-role



9. On the next page for networking keep everything to default choose your default VPC and security group.

10. In cluster endpoint access change it to Public.

Cluster endpoint access Info

Configure access to the Kubernetes API server endpoint.

Public

The cluster endpoint is accessible from outside of your VPC. Worker node traffic will leave your VPC to connect to the endpoint.

Public and private

The cluster endpoint is accessible from outside of your VPC. Worker node traffic to the endpoint will stay within your VPC.

Private

The cluster endpoint is only accessible through your VPC. Worker node traffic to the endpoint will stay within your VPC.

► Advanced settings

11. After that just keep everything to default and create you EKS cluster.

12. Once you have your cluster in place now the next step is to navigate to compute option.

Cluster info [Info](#)

Status Active	Kubernetes version Info 1.27	Support type Standard support until July 24, 2024	Provider EKS
----------------------------------	---	--	-----------------

[Overview](#) | [Resources](#) | **Compute** | [Networking](#) | [Add-ons](#) | [Access](#) | [Observability](#) | [Upgrade insights](#) | [Update history](#) | [Tags](#)

Details

API server endpoint https://AB2EDC1FCA76B53651938CAE1900A384.gr7.ap-south-1.eks.amazonaws.com	OpenID Connect provider URL https://oidc.eks.ap-south-1.amazonaws.com/id/AB2EDC1FCA76B53651938CAE1900A384	Created 24 minutes ago
Certificate authority LS0L51CRUDJTBDRVJUSUZJQ0FURS0tLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lJY2E0bE9neGdWSFF3RFFZSktvWklodmNOQVFTEJRQxdGVEVUTUJFROExVuUK	Cluster IAM role ARN arn:aws:iam::878893308172:role/EKS-cluster-role	Cluster ARN arn:aws:eks:ap-south-1:878893308172:cluster/democluster
		Platform version Info eks.11

13. In compute option click on add node group.

Node groups (0) [Info](#)

Group name	▲ Desired size	AMI release version	▼ Launch template	▼ Status
No node groups This cluster does not have any node groups. Nodes that are not part of an Amazon EKS managed node group are not shown in the AWS console.				
Add node group				

14. In node group first you need to give it a name, then you need to choose an IAM role.
15. After that you need to define an IAM role for it. For this role you can create a new role but instead I have used my previous role which I used for EC2 instance.
16. I can make use of the ec2 SQS role that I used earlier on. See, in the end you are going to have EC2 instances that are going to behave as your nodes. They will have Kubernetes running on them and they'll go ahead and run your containers. Now in terms of the permissions, I anyway had the Amazon SQS full access and also the Amazon EC2 container registry power user.
17. In your role add the below permissions and then add that role in your node group.

Permissions policies (5) [Info](#)

You can attach up to 10 managed policies.

Filter by Type			
<input type="checkbox"/> Policy name	Type	Attached entities	
<input type="checkbox"/> AmazonEC2ContainerRegistryPowerUser	AWS managed	1	
<input type="checkbox"/> AmazonEC2ContainerRegistryReadOnly	AWS managed	1	
<input type="checkbox"/> AmazonEKS_CNI_Policy	AWS managed	1	
<input type="checkbox"/> AmazonEKSWorkerNodePolicy	AWS managed	1	
<input type="checkbox"/> AmazonSQSFullAccess	AWS managed	3	

Node group configuration

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

Name
Assign a unique name for this node group.

The node group name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 63.

Node IAM role [Info](#)
Select the IAM role that will be used by the nodes. To create a new role, go to the [IAM console](#).
 [Edit](#) [Delete](#)

Info The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.
[Learn more](#)

18. Now on the next page keep everything to default and scroll down.
19. You can change the instance type to t2.micro if you want to reduce the cost of billing.

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.

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

Instance types [Info](#)
Select instance types you prefer for this node group.
 [Edit](#) [Delete](#)

t3.medium [X](#)
vCPU: 2 vCPUs Memory: 4 GiB Network: Up to 5 Gigabit Max ENI: 3 Max IPs: 18

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

20. Here in the node group scaling just change the desired size to 1 and minimum size to 1.
21. After that just create your node group.

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

22. Here you will see the status is saying it is in creating state. Wait for some time.

EKS > Clusters > [democluster](#) > Node group: EKSnodes

EKSnodes

[Edit](#) [Delete](#)

Node group configuration Info		
Kubernetes version 1.27	AMI type Info AL2_x86_64	Status Creating
AMI release version Info 1.27.9-20240213	Instance types t3.medium	Disk size 20 GiB

23. After sometime you will see that it is in the active state. Also you have a node in place.

Node group configuration Info		
Kubernetes version 1.27	AMI type Info AL2_x86_64	Status Active
AMI release version Info 1.27.9-20240213	Instance types t3.medium	Disk size 20 GiB

[Details](#) [Nodes](#) [Health issues 0](#) [Kubernetes labels](#) [Update config](#) [Kubernetes taints](#) [Update history](#) [Tags](#)

Details			
Node group ARN arn:aws:eks:ap-south-1:878893308172:nodemanager/democluster/EKSnodes/d2c6d62f-5136-e1a3-1f9a-240e6102874c	Autoscaling group name eks-EKSnodes-d2c6d62f-5136-e1a3-1f9a-240e6102874c	Capacity type On-Demand	Subnets subnet-090908b5b996470fc subnet-01c162e279b989d09 subnet-05b1afa053579e078
Created 2 minutes ago	Node IAM role ARN arn:aws:iam::878893308172:role/ec2-demo-sqs	Desired size 1 node	Configure remote access to nodes off
		Minimum size 1 node	
		Maximum size 2 nodes	

24. I can now deploy a container on this node as part of my cluster based on the image that I have in the Elastic Container registry service.
25. We need to make use of the kube Ctl tool. The kube Ctl tool allows you to interact with your Kubernetes cluster.
26. Now in another tab open AWS Cloud shell and there you need to install kubectl.
27. You can use the AWS documentation to install it.

<https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html>

28. You can also use these commands.

```
curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.9/2024-01-04/bin/linux/amd64/kubectl.sha256
```

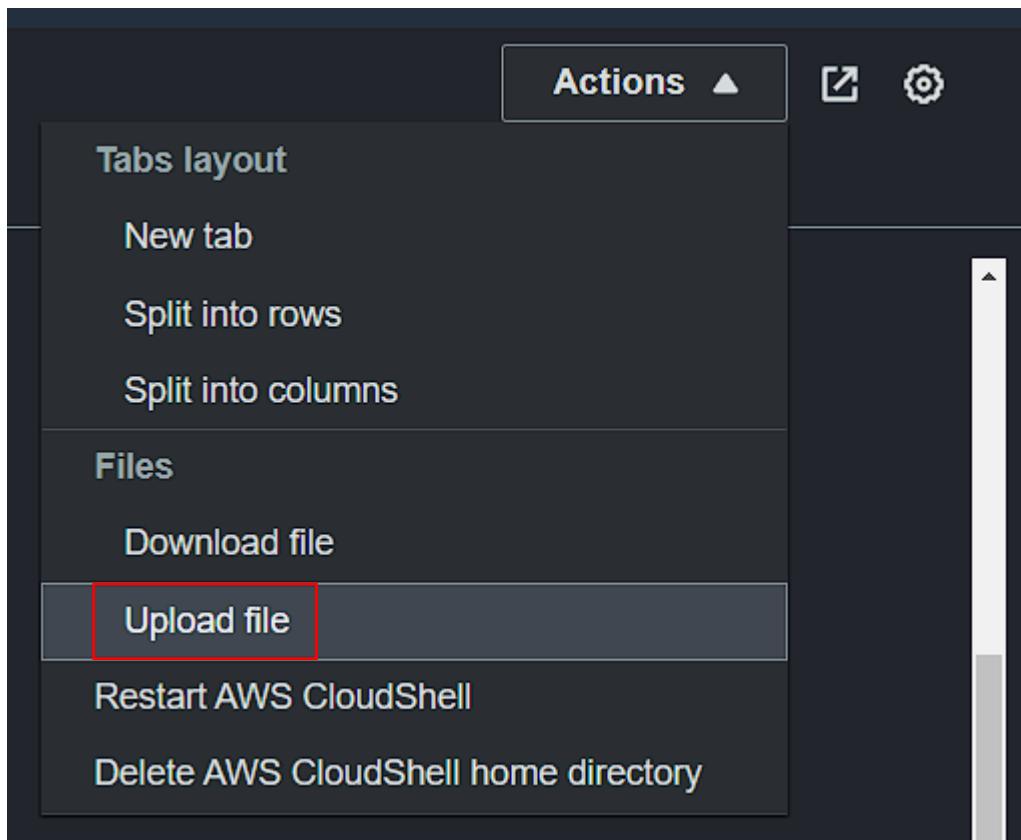
```
mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH
```

```
aws eks update-kubeconfig --region ap-south-1 --name democluster
```

```
kubectl get nodes
```

```
[cloudshell-user@ip-10-130-18-217 ~]$ curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.9/2024-01-04/bin/darwin/amd64/kubectl.sha256
[cldshll-user@ip-10-130-18-217 ~]$ curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.9/2024-01-04/bin/linux/amd64/kubectl
[cldshll-user@ip-10-130-18-217 ~]$ chmod +x ./kubectl
[cldshll-user@ip-10-130-18-217 ~]$ mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$HOME/bin:$PATH
[cldshll-user@ip-10-130-18-217 ~]$ aws eks update-kubeconfig --region ap-south-1 --name democluster
Updated context arn:aws:eks:ap-south-1:878893308172:cluster/democluster in /home/cloudshell-user/.kube/config
[cldshll-user@ip-10-130-18-217 ~]$ kubectl get nodes
NAME           STATUS  ROLES   AGE   VERSION
ip-172-31-6-215.ap-south-1.compute.internal  Ready   <none>  17m   v1.27.9-eks-5e0fdde
[cldshll-user@ip-10-130-18-217 ~]$
```

29. After you have executed the above commands successfully now you need to download deployment.yaml file from GitHub.
30. In CloudShell if you will click on actions, you can see an option for upload file. Use this option to upload that deployment.yaml file.



31. One more thing before you upload this file. You need to change the URI of this file with your URI of ECR.

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: app-deployment
5    namespace: default
6    labels:
7      app: app-deployment
8  spec:
9    replicas: 1
10   selector:
11     matchLabels:
12       app: app-deployment
13   template:
14     metadata:
15       labels:
16         app: app-deployment
17     spec:
18       containers:
19         - name: send-messages
20           image: 878893308172.dkr.ecr.ap-south-1.amazonaws.com/sendmessages:latest
```

32. Once everything is done and you have uploaded that YAML file. Then you need to run this command.

```
kubectl apply -f Deployment.yml
```

```
[cloudshell-user@ip-10-130-18-217 ~]$ kubectl apply -f Deployment.yml
deployment.apps/app-deployment created
[cloudshell-user@ip-10-130-18-217 ~]$
```

33. Afterwards if you will go to your nodes on EKS you will see all the nodes that are working as pods including you deployment file.

Pods (6) Info			
Name	Status	Created	IP
app-deployment-6f987dcc4c-j5bj5	Running	a minute ago	172.31.15.220
aws-node-szrkd	Running	26 minutes ago	172.31.6.215
coredns-669bc8b447-c8djn	Running	an hour ago	172.31.2.175
coredns-669bc8b447-cwv5n	Running	an hour ago	172.31.7.110
eks-pod-identity-agent-xtj84	Running	26 minutes ago	172.31.6.215
kube-proxy-mn4kc	Running	26 minutes ago	172.31.6.215

34. Then if you will navigate to SQS you can see your messages in place.

Messages (10)					
	ID	Sent	Size	Receive count	
<input type="checkbox"/>	02d565d2-427e-4a9d-a9a0-aa1d81da300e	2024-02-15T16:40+05:30	51 bytes	1	View details Delete
<input type="checkbox"/>	3df5babf-07f1-4cb7-8f2d-d976a57228d6	2024-02-15T16:40+05:30	52 bytes	1	View details Delete
<input type="checkbox"/>	bde1fb7d-4668-4d4f-89d2-ac7282b54aad	2024-02-15T16:40+05:30	52 bytes	1	View details Delete
<input type="checkbox"/>	34a78e6b-6521-41a0-baa1-2bc13526ffd5	2024-02-15T16:40+05:30	51 bytes	1	View details Delete
<input type="checkbox"/>	c4acb89a-0510-47f7-bc06-7629ac5fca51	2024-02-15T16:40+05:30	52 bytes	1	View details Delete
<input type="checkbox"/>	3c2f114c-a47b-4ce8-9c18-4b3313ab3c51	2024-02-15T16:41+05:30	52 bytes	1	View details Delete
<input type="checkbox"/>	6f0da301-cec2-49a5-9b84-66cb81fedc8b	2024-02-15T16:41+05:30	52 bytes	1	View details Delete
<input type="checkbox"/>	95a6bf0c-0281-43c8-b7d1-644e0defcbe2	2024-02-15T16:41+05:30	52 bytes	1	View details Delete
<input type="checkbox"/>	2d623fad-fb03-4d3f-9f7e-c7afaabebe85	2024-02-15T16:41+05:30	52 bytes	1	View details Delete
<input type="checkbox"/>	4334891c-250c-45d1-b1a2-198bdc668bdf	2024-02-15T16:42+05:30	52 bytes	1	View details Delete

35. In order to perform cleanup first go to clusters then open it and navigate to Compute sections. From there delete your node. Then delete your cluster.