

# DevOps Assignment

Q1. Use CodeDeploy in order to deploy code to EC2 and keep code in S3 bucket.

The screenshot shows the AWS EC2 Instances page. A single instance is listed:

- Name:** CodeDeploy MyServer
- Instance ID:** i-047194df3b9c9903d
- Instance state:** Running
- Instance type:** t2.micro
- Status check:** 2/2 checks passed
- Alarm status:** No alarms
- Availability Zone:** us-east-1c

```
I, [2023-07-07T13:02:29.54872 #3442] INFO -- : Downloading packages from bucket aws-codedeploy-us-east-1 and key releases/codedeploy-agent-1.6.0-49.noarch.rpm...
I, [2023-07-07T13:02:29.40805 #3442] INFO -- : Endpoint: https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/releases/codedeploy-agent-1.6.0-49.noarch.rpm...
I, [2023-07-07T13:02:29.407988 #3442] INFO -- : Executing '/usr/bin/yum -y localinstall /tmp/codedeploy-agent-1.6.0-49.noarch.rpm ...
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Examining /tmp/codedeploy-agent-1.6.0-49.noarch.rpm: codedeploy-agent-1.6.0-49.noarch
Marking /tmp/codedeploy-agent-1.6.0-49.noarch.rpm-20230707-3442-6ojctc.rpm to be installed
Resolving Dependencies
--> Processing Transaction check
--> Package codedeploy-agent.noarch 0:1.6.0-49 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
=====
Package          Arch      Version           Repository      Size
=====
Installing:
codedeploy-agent      noarch   1.6.0-49          /codedeploy-agent-1.6.0-49.noarch.rpm-20230707-3442-6ojctc      13 M
Transaction Summary
=====
Install 1 Package

Total size: 13 M
Installed size: 13 M
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction

pre hook : 1
Checking if there is already a process named codedeploy-agent running.
  Installing : codedeploy-agent-1.6.0-49.noarch

post hook : 1
Check if there is a codedeployagent config file.
  Start codedeploy-agent in post hook if this is a first install.
  Verifying  : codedeploy-agent-1.6.0-49.noarch

Installed:
  codedeploy-agent.noarch 0:1.6.0-49

Complete!
I, [2023-07-07T13:02:31.838199 #3442] INFO -- : Update check complete.
I, [2023-07-07T13:02:31.838261 #3442] INFO -- : Stopping updater.
[root@ip-172-31-87-123 ~]# sudo service codedeploy-agent status
The AWS CodeDeploy agent is running as PID 3523
[root@ip-172-31-87-123 ~]#
```

Screenshot of the AWS CodeDeploy console showing a successful deployment.

**Deployment details:**

Application	Deployment ID	Status
developmentInstances	d-64RBQ12F0	<span style="color: green;">Succeeded</span>
Deployment configuration	Deployment group	Initiated by
CodeDeployDefault.spurbs	developmentInstances	User action
Deployment description		
-		

**Revision details:**

Revision location	Revision created	Revision description
s3://mybucketapplication/myapplication/SampleApp_Linux.zip	4 minutes ago	Application revision registered by Deployment ID: d-64RBQ12F0

CloudShell   Feedback   Language   © 2023, Amazon Web Services India Private Limited or its affiliates.   Privacy   Terms   Cookie preferences

Screenshot of the AWS S3 Management Console showing a bucket named "mybucketapplication".

**Buckets:**

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- IAM Access Analyzer for S3

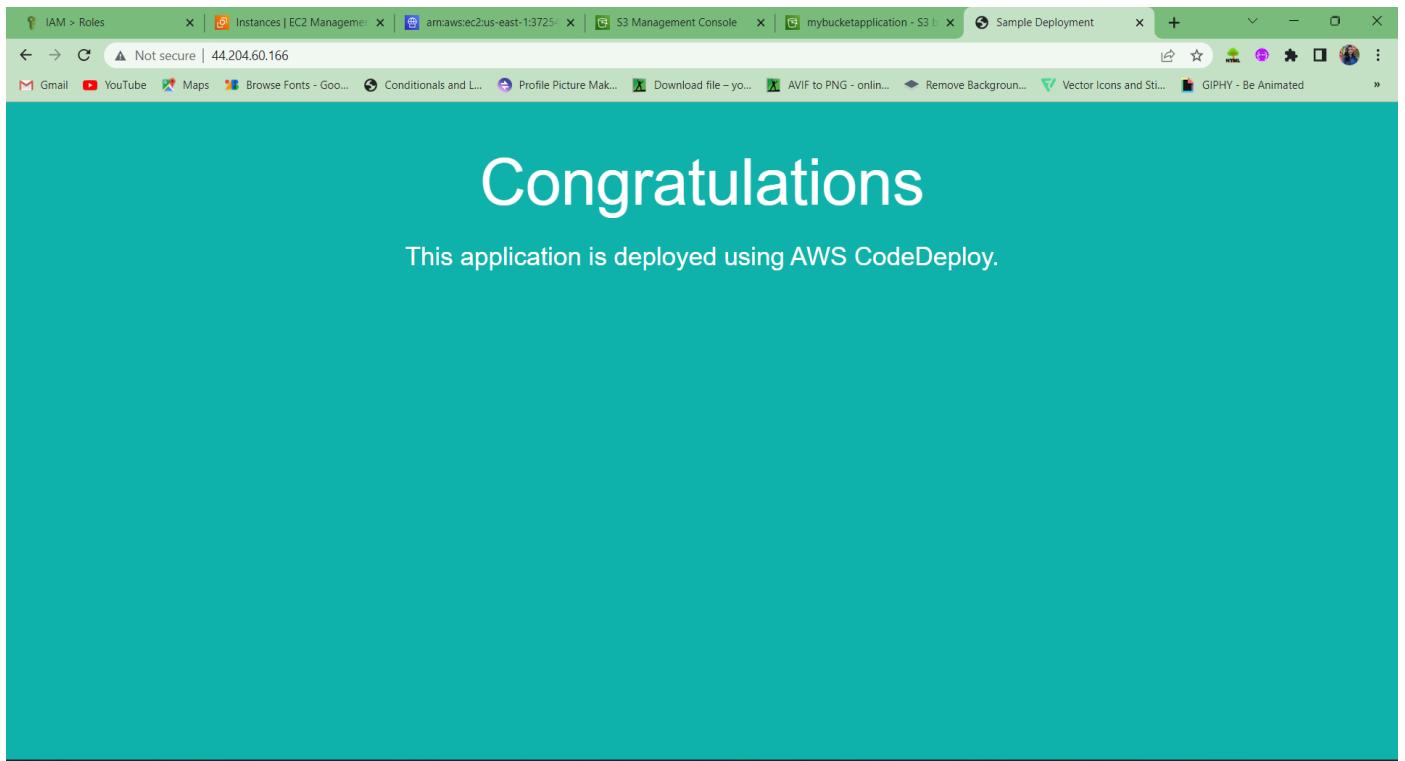
**Storage Lens:**

- Dashboards
- AWS Organizations settings

**Objects:**

Name	Type	Last modified	Size	Storage class
SampleApp_Linux.zip	zip	July 7, 2023, 19:02:19 (UTC+05:30)	5.4 KB	Standard

CloudShell   Feedback   Language   © 2023, Amazon Web Services India Private Limited or its affiliates.   Privacy   Terms   Cookie preferences



## Q2. USING TERRAFORM CREATE AN AWS EKS CLUSTER CREATE FOLLOWING COMPONENTS

- I.VPC-2 PUBLIC SHEET AND 2 PRIVATE SUBNETS.
- II.IAM ROLE – FOR MASTER AND WORKER.
- III.WORKER NODE FOR CLUSTER USING AMAZON LINUX.

The screenshot shows the AWS EC2 Instances page with the search bar set to 'vpc'. The main table displays three instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Put
eks-worker	i-00cf171d8271716f	Running	t2.medium	2/2 checks passed	No alarms	us-east-1b	-
eks-worker	i-04ee40a79c489ebdf	Running	t2.medium	2/2 checks passed	No alarms	us-east-1a	-
<b>proj2</b>	<b>i-0a38917f53589f500</b>	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2

The instance details for 'proj2' are shown in the expanded view:

**Instance: i-0a38917f53589f500 (proj2)**

**Details** | Security | Networking | Storage | Status checks | Monitoring | Tags

**Instance summary**

Instance ID: i-0a38917f53589f500 (proj2)	Public IPv4 address: 54.152.232.200   open address	Private IPv4 addresses: 172.31.88.203
IPv6 address: -	Instance state: Running	Public IPv4 DNS: ec2-54-152-232-200.compute-1.amazonaws.com   open address
Hostname type: IP name: ip-172-31-88-203.ec2.internal	Private IP DNS name (IPv4 only): ip-172-31-88-203.ec2.internal	Elastic IP addresses: -
Answer private resource DNS name: -	Instance type: t2.micro	-

The screenshot shows the AWS EC2 Instances page with the search bar set to 'vpc'. The main table displays three instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Put
eks-worker	i-00cf171d8271716f	Running	t2.medium	2/2 checks passed	No alarms	us-east-1b	-
<b>eks-worker</b>	<b>i-04ee40a79c489ebdf</b>	Running	t2.medium	2/2 checks passed	No alarms	us-east-1a	-
proj2	i-0a38917f53589f500	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2

The instance details for 'eks-worker' are shown in the expanded view:

**Instance: i-04ee40a79c489ebdf (eks-worker)**

**Details** | Security | Networking | Storage | Status checks | Monitoring | Tags

**Instance summary**

Instance ID: i-04ee40a79c489ebdf (eks-worker)	Public IPv4 address: -	Private IPv4 addresses: 10.0.2.129
IPv6 address: -	Instance state: Running	Public IPv4 DNS: -
Hostname type: IP name: ip-10-0-2-129.ec2.internal	Private IP DNS name (IPv4 only): ip-10-0-2-129.ec2.internal	Elastic IP addresses: -
Answer private resource DNS name: -	Instance type: t2.medium	-

Screenshot of the AWS EC2 Instances page showing three running instances: eks-worker (t2.medium), eks-worker (t2.medium), and proj2 (t2.micro). The proj2 instance is selected.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
eks-worker	i-00cf1712d8271716f	Running	t2.medium	2/2 checks passed	No alarms	us-east-1b
eks-worker	i-04ee40a79c489ebdf	Running	t2.medium	2/2 checks passed	No alarms	us-east-1a
proj2	i-0a38917f53589f500	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c

Details tab for the selected instance (i-00cf1712d8271716f):

Instance ID: i-00cf1712d8271716f (eks-worker)	Public IPv4 address: -	Private IPv4 addresses: 10.0.3.151
IPv6 address: -	Instance state: Running	Public IPv4 DNS: -
Hostname type: IP name: ip-10-0-3-151.ec2.internal	Private IP DNS name (IPv4 only): ip-10-0-3-151.ec2.internal	Elastic IP addresses: -
Answer private resource DNS name: -	Instance type: t2.medium	-

Screenshot of the AWS IAM User details page for user proj\_2.

**Summary:**

ARN: arn:aws:iam::372541270057:user/proj_2	Console access: Disabled	Access key 1: AKIAVNPJQVAU73CQLNOL - Active, Used today, Created today
Created: July 08, 2023, 17:01 (UTC+05:30)	Last console sign-in: -	Access key 2: Not enabled

**Security credentials:**

Console sign-in link: https://372541270057.sigin.aws.amazon.com/console	Console password: Not enabled
---	-------------------------------

(2) WhatsApp

IAM > Users > proj\_2 > Create

Install Terraform | Terraform | Help

us-east-1.console.aws.amazon.com/iamv2/home?region=us-east-1#/users/details/proj\_2/create-access-key

Gmail YouTube Maps Browse Fonts - Go... Conditional and L... Profile Picture Mak... Download file - yo... AVIF to PNG - onlin... Remove Background... Vector Icons and Sti... GIPHY - Be Animated

AWS Services Search [Alt+S]

Global Sonali Rath

**Access key created**

This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1  
Access key best practices & alternatives

Step 2 - optional  
Set description tag

Step 3  
Retrieve access keys

**Retrieve access keys** Info

**Access key**

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
AKIAVNPJQVAU73CQLNOL	***** Show

**Access key best practices**

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [Best practices for managing AWS access keys](#).

Download .csv file Done

```
ec2-54-227-136-116.compute-1.amazonaws.com:22 - root@ip-172-31-84-144:~ - Xshell 7 (Free for Home/School)
File Edit View Tools Tab Window Help
Session Manager  x  1 ec2-54-227-136-116.com... + 
ssh://ec2-user@ec2-54-227-136-116.compute-1.amazonaws.com:22
To add the current session, click on the left arrow button.

All Sessions

The following dependency selections recorded in the lock file are inconsistent with the current configuration:
- provider registry.terraform.io/hashicorp/aws: required by this configuration but no version is selected

To make the initial dependency selections that will initialize the dependency lock file, run:
terraform init

[root@ip-172-31-84-144 ~]# terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.7.0...
- Installed hashicorp/aws v5.7.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
run this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[root@ip-172-31-84-144 ~]# terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_autoscaling_group.worker_asg_ass2 will be created
+ resource "aws_autoscaling_group" "worker_asg_ass2" {
    + arn = (known after apply)
    + availability_zones = (known after apply)
```

ec2-54-152-232-200.compute-1.amazonaws.com:22 - root@ip-172-31-88-203:~ - Xshell 7 (Free for Home/School)

File Edit View Tools Tab Window Help

ssh://ec2-user@ec2-54-152-232-200.compute-1.amazonaws.com:22

To add the current session, click on the left arrow button.

Session Manager

All Sessions

```
aws eks cluster eks_cluster_sonali: Still creating... [3m30s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [3m40s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [3m50s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [4m10s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [4m20s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [4m30s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [4m40s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [4m50s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [5m0s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [5m10s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [5m20s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [5m30s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [5m40s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [5m50s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [6m0s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [6m10s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [6m20s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [6m30s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [6m40s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [6m50s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [7m0s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [7m10s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [7m20s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [7m30s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [7m40s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [7m50s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [8m0s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [8m10s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [8m20s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [8m30s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [8m40s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [8m50s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [9m0s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [9m10s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [9m20s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [9m30s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [9m40s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [9m50s elapsed]
aws eks cluster eks_cluster_sonali: Still creating... [10m0s elapsed]
aws eks cluster eks_cluster_sonali: Creation complete after 10m59s [id:my-eks-cluster_sonali:rath]
```

Name	All Sess...
Type	Folder
Sub Items	6
Host	
Port	22
Protocol	SSH
User Name	
Description	

Apply complete! Resources: 10 added, 0 changed, 0 destroyed.

[root@ip-172-31-88-203 ~]#

ssh://ec2-user@ec2-54-152-232-200.compute-1.amazonaws.com:22

SSH2 xterm 1 210x49 8.49.28 1 session ENG IN 17:30 08-07-2023 10 CAP NUM

IAM > Users > Create ... Auto Scaling groups | EC2 Manager Internet gateways | VPC Manager Install Terraform | Terraform | Home ... +

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroups:id=eks-worker-asg\_sonali;view=details

Gmail YouTube Maps Browse Fonts - Google Conditional and L... Profile Picture Mak... Download file - yo... AVIF to PNG - onlin... Remove Background... Vector Icons and St... GIPHY - Be Animated

AWS Services Q vpc

New EC2 Experience Learn more

EC2 > Auto Scaling groups

Auto Scaling groups (1/1) Info Launch configurations Launch templates Actions Create Auto Scaling group

Search your Auto Scaling groups

Name	Launch template/configuration	Instances	Status	Desired capacity	Min
eks-worker-asg_sonali	eks-worker-It_sonali[rath202307081141]	2	-	2	2

Auto Scaling group: eks-worker-asg\_sonali

Details Activity Automatic scaling Instance management Monitoring Instance refresh

Group details

Auto Scaling group name	Desired capacity	Status	Amazon Resource Name (ARN)
eks-worker-asg_sonali	2	-	arn:aws:autoscaling:us-east-1:372541270057:autoScalingGroup:bb3532d7-1de5-4619-ad46-a630dc318401:autoScalingGroupName/eks-worker-asg_sonali
Date created	Minimum capacity		
Sat Jul 08 2023 17:11:44 GMT+0530 (India Standard Time)	2		
	Maximum capacity		

CloudShell Feedback Language

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS VPC Management Console showing the list of VPCs. The 'Your VPCs' section displays two VPCs: 'vpc-0eb3271437a850b3e' and 'eks-vpc\_sonali'. The 'eks-vpc\_sonali' VPC is selected.

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options
vpc-0eb3271437a850b3e	vpc-0eb3271437a850b3e	Available	172.31.0.0/16	-	dept-01
<b>eks-vpc_sonali</b>	<b>vpc-0c6a2b73f55331322</b>	<b>Available</b>	<b>10.0.0.0/16</b>	-	<b>dept-01</b>

The details for the selected VPC ('eks-vpc\_sonali') are shown below:

Details	Value	Details	Value
VPC ID	vpc-0c6a2b73f55331322	State	Available
Tenancy	Default	DHCP option set	dept-01c112fa2da07c203
Default VPC	No	IPv4 CIDR	10.0.0.0/16
Network Address Usage metrics	Disabled	Route 53 Resolver DNS Firewall rule groups	-
		Owner ID	372541270057

Screenshot of the AWS Subnets Management Console showing the list of subnets. The 'Subnets' section displays four subnets under the VPC 'eks-vpc\_sonali'.

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
-	subnet-0889d79275ee1dedb	Available	vpc-0eb3271437a850b3e	172.31.64.0/20	-
<b>public-subnet-1_sonali</b>	<b>subnet-04588fb1a813a0339</b>	<b>Available</b>	<b>vpc-0c6a2b73f55331322   eks-vpc_sonali</b>	<b>10.0.0.0/24</b>	-
-	subnet-0e0af0fff9cd7e8b	Available	vpc-0eb3271437a850b3e	172.31.0.0/20	-
-	subnet-0e543b378a9e7cd8b	Available	vpc-0eb3271437a850b3e	172.31.48.0/20	-

The details for the selected subnet ('public-subnet-1\_sonali') are shown below:

Details	Value	Details	Value
Subnet ID	subnet-04588fb1a813a0339	Subnet ARN	arn:aws:ec2:us-east-1:372541270057:subnet/subnet-04588fb1a813a0339
Available IPv4 addresses	251	State	Available
Network border group	us-east-1	IPv6 CIDR	-
Default subnet	No	VPC	vpc-0c6a2b73f55331322   eks-vpc_sonali
Customer-owned IPv4 pool	-	Auto-assign IPv6 address	No
		Auto-assign public IPv4 address	No
		IPv4 CIDR reservations	-
		Outpost ID	-
		IPv6 CIDR reservations	-

Screenshot of the AWS Subnets page showing a list of subnets and their details.

**Subnets (1/10) Info**

Subnet	Subnet ID	Status	VPC	CIDR Range	AZ
private-subnet-2_s...	subnet-053ebc15c275bc4db	Available	vpc-0c6a2b73f55331322   eks-vpc_sonali	10.0.3.0/24	-

**Details for private-subnet-2\_s...:**

IPv6 CIDR	Route table	Network ACL
-	rtb-0e0342ee77392e9d	acl-0f5dc51aeab553658
VPC	Auto-assign IPv6 address	Auto-assign customer-owned IPv4 address
vpc-0c6a2b73f55331322   eks-vpc_sonali	No	No
Default subnet	IPv4 CIDR reservations	IPv6 CIDR reservations
No	-	-
Customer-owned IPv4 pool	Outpost ID	Resource name DNS A record
-	-	Disabled
IPv6-only	Hostname type	Resource name DNS AAAA record
No	IP name	Disabled
DNS64		

Screenshot of the AWS Subnets page showing a list of subnets and their details.

**Subnets (1/10) Info**

Subnet	Subnet ID	Status	VPC	CIDR Range	AZ
private-subnet-2_s...	subnet-053ebc15c275bc4db	Available	vpc-0c6a2b73f55331322   eks-vpc_sonali	10.0.3.0/24	-
public-subnet-2_s...	subnet-0d4fdc1be12af1764	Available	vpc-0c6a2b73f55331322   eks-vpc_sonali	10.0.1.0/24	-
-	subnet-078721afc6af6b599	Available	vpc-0eb3271437a850b3e	172.31.16.0/20	-
private-subnet-1_s...	subnet-02bca4106ebe91eff	Available	vpc-0c6a2b73f55331322   eks-vpc_sonali	10.0.2.0/24	-
-	subnet-0eb321065ec3d2fe1e	Available	vpc-0eb3271437a850b3e	172.31.80.0/20	-

**Details for public-subnet-2\_s...:**

IPv6 CIDR	Route table	Network ACL
-	rtb-0e0342ee77392e9d	acl-0f5dc51aeab553658
VPC	Auto-assign IPv6 address	Auto-assign customer-owned IPv4 address
vpc-0c6a2b73f55331322   eks-vpc_sonali	No	No
Default subnet	IPv4 CIDR reservations	IPv6 CIDR reservations
No	-	-
Customer-owned IPv4 pool	Outpost ID	Resource name DNS A record
-	-	Disabled
IPv6-only	Hostname type	Resource name DNS AAAA record
No	IP name	Disabled
DNS64		

**Subnets (1/10) Info**

Name	ID	Status	VPC	CIDR
private-subnet-2_s...	subnet-053ebc15c275bc4db	Available	vpc-0c6a2b73f55331322   eks...	10.0.3.0/24
public-subnet-2_s...	subnet-0d4fdc1be12af1764	Available	vpc-0c6a2b73f55331322   eks...	10.0.1.0/24
-	subnet-078721afc6fb599	Available	vpc-0eb3271437a850b3e	172.31.16.0/20
private-subnet-1_s...	subnet-02bac4106ebe91eff	Available	vpc-0c6a2b73f55331322   eks...	10.0.2.0/24
-	subnet-0eb31065ec3d2fe1e	Available	vpc-0eb3271437a850b3e	172.31.80.0/20

**249**

Network border group	IPv6 CIDR	us-east-1a	use1-az6
us-east-1	-	Route table	Network ACL
Default subnet	VPC	rtb-0e00342ee77392e9d	acl-0f5dc51aeab553658
No	vpc-0c6a2b73f55331322   eks-vpc_sonali	Auto-assign IPv6 address	Auto-assign customer-owned IPv4 address
Customer-owned IPv4 pool	No	No	No
-	Auto-assign public IPv4 address	IPv4 CIDR reservations	IPv6 CIDR reservations
IPv6-only	No	-	-
No	Outpost ID	Resource name DNS A record	Resource name DNS AAAA record
DNS64	IP name	Disabled	Disabled

## Main.tf(terraform file)

```
# Define the provider
```

```
provider "aws" {
```

```
    region = "us-east-1" # Replace with your desired
    region
```

```
}
```

```
# Create VPC for the EKS cluster
```

```
resource "aws_vpc" "rath_eks_vpc_sonali" {
```

```
    cidr_block = "10.0.0.0/16" # Replace with your  
desired CIDR block
```

```
tags = {  
    Name = "eks-vpc_sonali"  
}  
}
```

```
# Create public subnets for the EKS cluster  
  
resource "aws_subnet" "rath_public_subnet_1_sonali"  
{  
    vpc_id          = aws_vpc.rath_eks_vpc_sonali.id  
    cidr_block      = "10.0.0.0/24" # Replace with your  
desired CIDR block  
    availability_zone = "us-east-1a" # Replace with  
your desired availability zone
```

```
tags = {  
    Name = "public-subnet-1_sonalirath"  
}  
}
```

```
resource "aws_subnet" "rath_public_subnet_2_sonali" {
    vpc_id          = aws_vpc.rath_eks_vpc_sonali.id
    cidr_block      = "10.0.1.0/24" # Replace with your
    desired CIDR block
    availability_zone = "us-east-1b" # Replace with
    your desired availability zone

    tags = {
        Name = "public-subnet-2_sonalirath"
    }
}

# Create private subnets for the EKS cluster

resource "aws_subnet"
"rath_private_subnet_1_sonali" {
    vpc_id          = aws_vpc.rath_eks_vpc_sonali.id
    cidr_block      = "10.0.2.0/24" # Replace with your
    desired CIDR block
    availability_zone = "us-east-1a" # Replace with
    your desired availability zone
```

```
tags = {  
    Name = "private-subnet-1_sonalirath"  
}  
}  
  
}
```

```
resource "aws_subnet"  
"rath_private_subnet_2_sonali" {  
    vpc_id          = aws_vpc.rath_eks_vpc_sonali.id  
    cidr_block      = "10.0.3.0/24" # Replace with your  
desired CIDR block  
    availability_zone = "us-east-1b" # Replace with  
your desired availability zone
```

```
tags = {  
    Name = "private-subnet-2_sonalirath"  
}  
}  
  
}
```

```
# Create IAM roles for the EKS cluster  
resource "aws_iam_role" "rathsonali_eks_worker_role"  
{
```

```
name = "eks-worker-role_sonalirath"
```

```
assume_role_policy = jsonencode({
```

```
    Version = "2012-10-17"
```

```
    Statement = [
```

```
        {
```

```
            Action = "sts:AssumeRole"
```

```
            Effect = "Allow"
```

```
            Principal = {
```

```
                Service = "ec2.amazonaws.com"
```

```
            }
```

```
        }
```

```
    ]
```

```
})
```

```
managed_policy_arns = [
```

```
    "arn:aws:iam::aws:policy/AmazonEKSClusterPolicy"
```

```
]
```

```
}
```

```
resource "aws_iam_role" "rathsonali_eks_master_role"
{
  name = "eks-master-role_sonalirath"

  assume_role_policy = jsonencode({
    Version  = "2012-10-17"
    Statement = [
      {
        Action  = "sts:AssumeRole"
        Effect  = "Allow"
        Principal = {
          Service = "eks.amazonaws.com"
        }
      }
    ]
  })
}
```

```
managed_policy_arns = [
  "arn:aws:iam::aws:policy/AmazonEKSClusterPolicy"
]
```

```
}

# Create the EKS cluster

resource "aws_eks_cluster" "eks_cluster_sonali" {

    name      = "my-eks-cluster_sonalirath"
    role_arn =
    aws_iam_role.rathsonali_eks_master_role.arn
    version   = "1.27" # Replace with a supported
    Kubernetes version

    vpc_config {

        subnet_ids = [
            aws_subnet.rath_public_subnet_1_sonali.id,
            aws_subnet.rath_public_subnet_2_sonali.id,
            aws_subnet.rath_private_subnet_1_sonali.id,
            aws_subnet.rath_private_subnet_2_sonali.id,
        ]
    }
}

# Create the launch template for worker nodes

resource "aws_launch_template" "worker_lt_sonali" {

    name_prefix  = "eks-worker-lt_sonalirath"
```

```
image_id      = "ami-04823729c75214919" # Replace  
with the desired Amazon Linux AMI ID
```

```
instance_type = "t2.medium"    # Replace with your  
desired instance type
```

```
block_device_mappings {  
  device_name = "/dev/xvda"  
  ebs {  
    volume_size = 8 # Replace with your desired root  
volume size in GB
```

```
  }  
}  
}
```

```
# Create the autoscaling group for worker nodes
```

```
resource "aws_autoscaling_group"  
"worker_asg_sonali" {  
  name          = "eks-worker-asg_sonali"  
  launch_template {  
    id      = aws_launch_template.worker_lt_sonali.id  
    version = "$Latest"  
  }
```

```
min_size      = 2 # Replace with your desired  
minimum number of worker nodes
```

```
max_size      = 5 # Replace with your desired  
maximum number of worker nodes
```

```
desired_capacity = 2 # Replace with your desired  
initial number of worker nodes
```

```
vpc_zone_identifier = [  
    aws_subnet.rath_private_subnet_1_sonali.id,  
    aws_subnet.rath_private_subnet_2_sonali.id,  
]
```

```
tag {  
    key      = "Name"  
    value     = "eks-worker"  
    propagate_at_launch = true  
}
```

**NAME: SONALI RATH**

**SIC NUMBER : 20BCEE70**