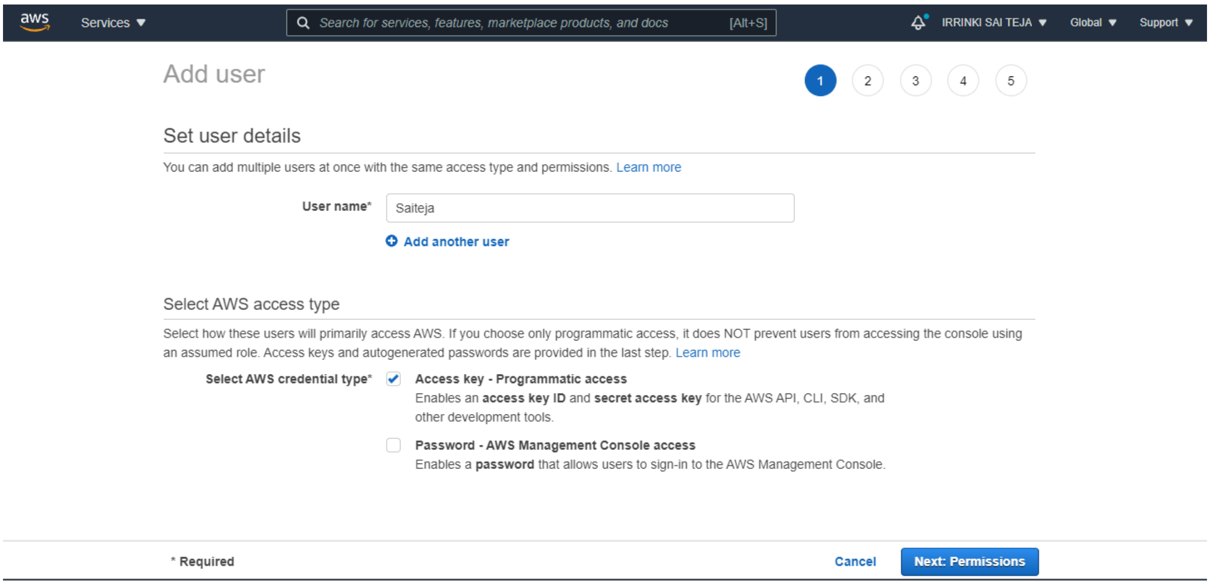
AWS CLI (Command Line Interface)

Setting up IAM user

Creating User with Access-key



Add user

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* Saiteja

Add another user

Select AWS access type

Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console using an assumed role. Access keys and autogenerated passwords are provided in the last step. Learn more

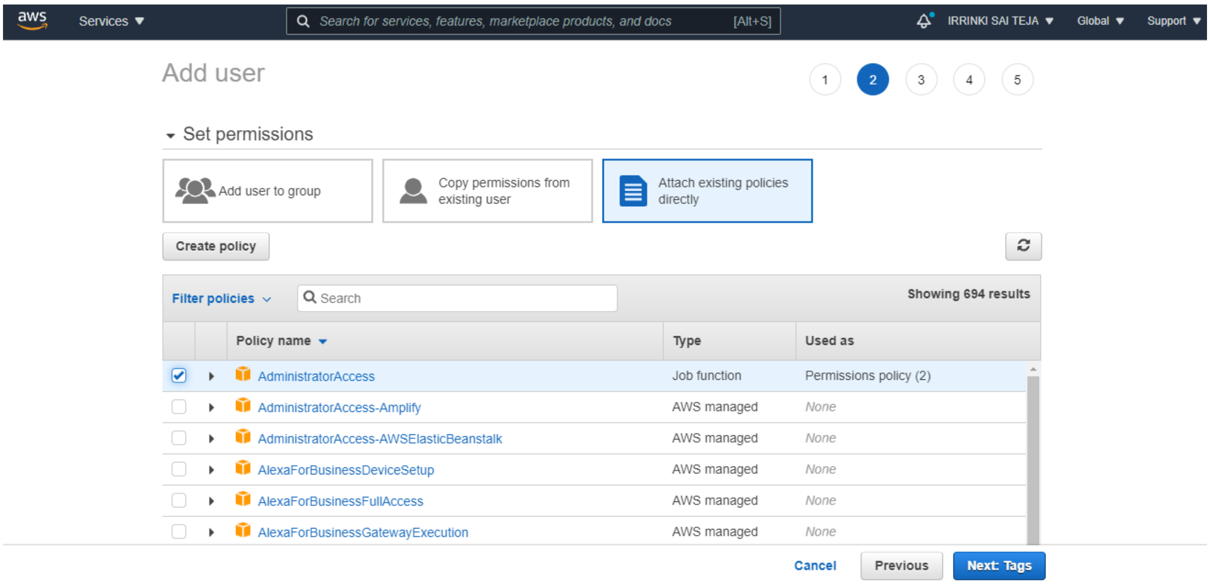
Select AWS credential type*

☑ **Access key - Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

□ **Password - AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

* Required

Set permissions & Attaching Policies

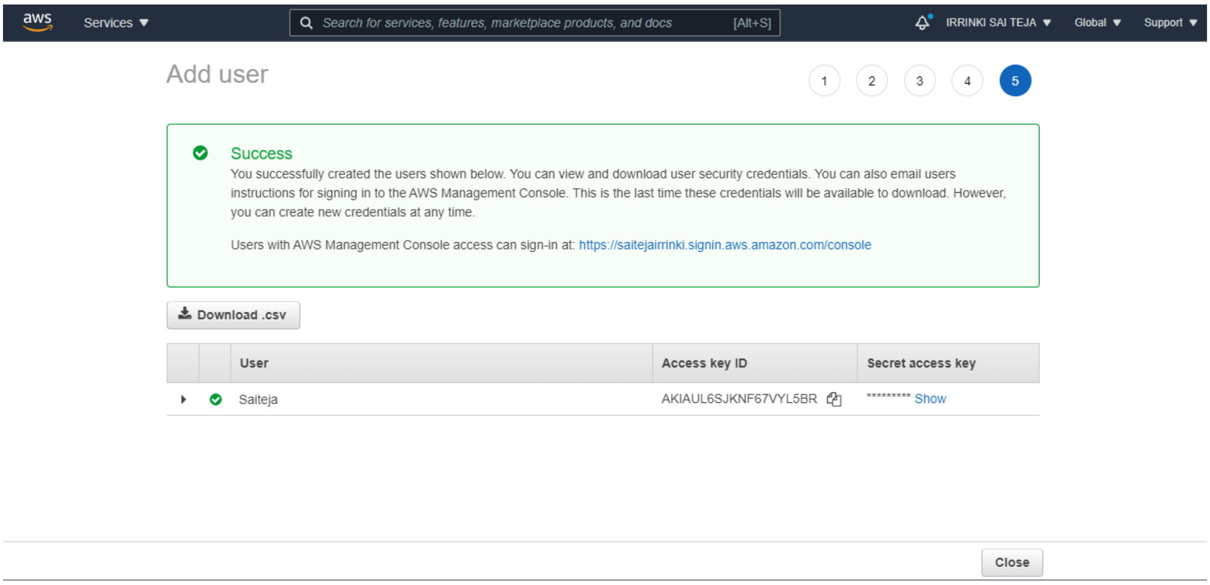

aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
IRRINKI SAI TEJA ▼
Global ▼
Support ▼
1 2 3 4 5
Add user to group
Copy permissions from existing user
Attach existing policies directly
Create policy
Filter policies
Search
Cancel
Previous
Next: Tags

Add user

Set permissions

Showing 694 results

		Policy name	Type	Used as
☑	▸	AdministratorAccess	Job function	Permissions policy (2)
□	▸	AdministratorAccess-Amplify	AWS managed	None
□	▸	AdministratorAccess-AWSElasticBeanstalk	AWS managed	None
□	▸	AlexaForBusinessDeviceSetup	AWS managed	None
□	▸	AlexaForBusinessFullAccess	AWS managed	None
□	▸	AlexaForBusinessGatewayExecution	AWS managed	None



Add user

Success

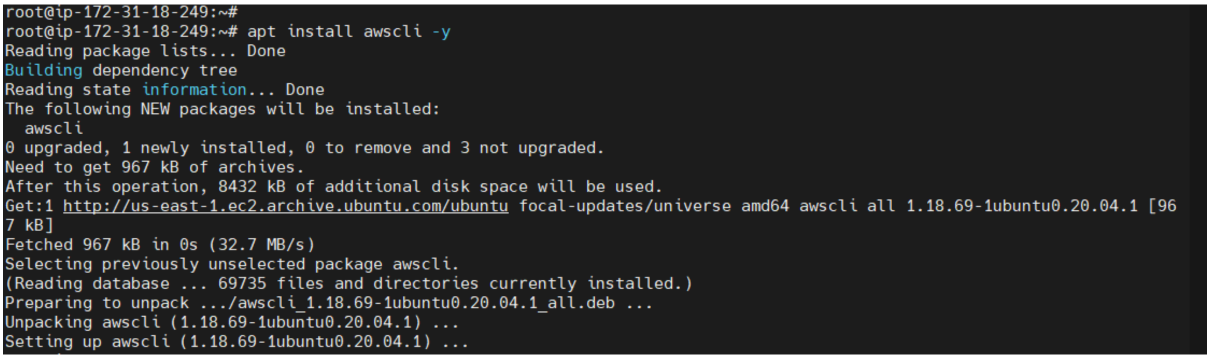
You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: https://saitejairrinki.signin.aws.amazon.com/console

		User	Access key ID	Secret access key
▸	✓	Saiteja	AKIAUL6SJKNF67VYL5BR	********* Show

Installing AWS-CLI

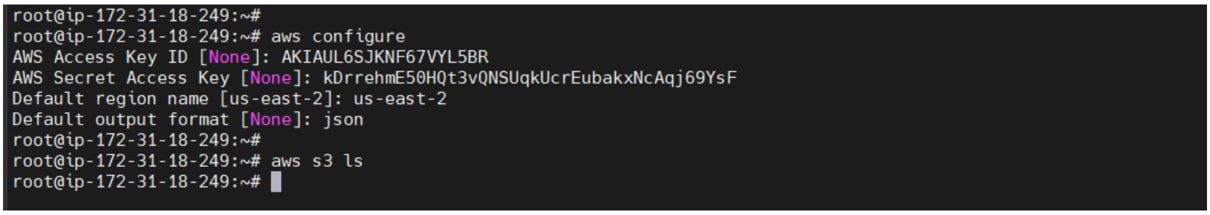
# apt update
# apt install awscli -y



```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# apt install awscli -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  awscli
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 967 kB of archives.
After this operation, 8432 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 awscli all 1.18.69-1ubuntu0.20.04.1 [96
7 kB]
Fetched 967 kB in 0s (32.7 MB/s)
Selecting previously unselected package awscli.
(Reading database ... 69735 files and directories currently installed.)
Preparing to unpack .../awscli_1.18.69-1ubuntu0.20.04.1_all.deb ...
Unpacking awscli (1.18.69-1ubuntu0.20.04.1) ...
Setting up awscli (1.18.69-1ubuntu0.20.04.1) ...
```

Configure AWS CLI with IAM user Credentials with specific Region

# aws configure



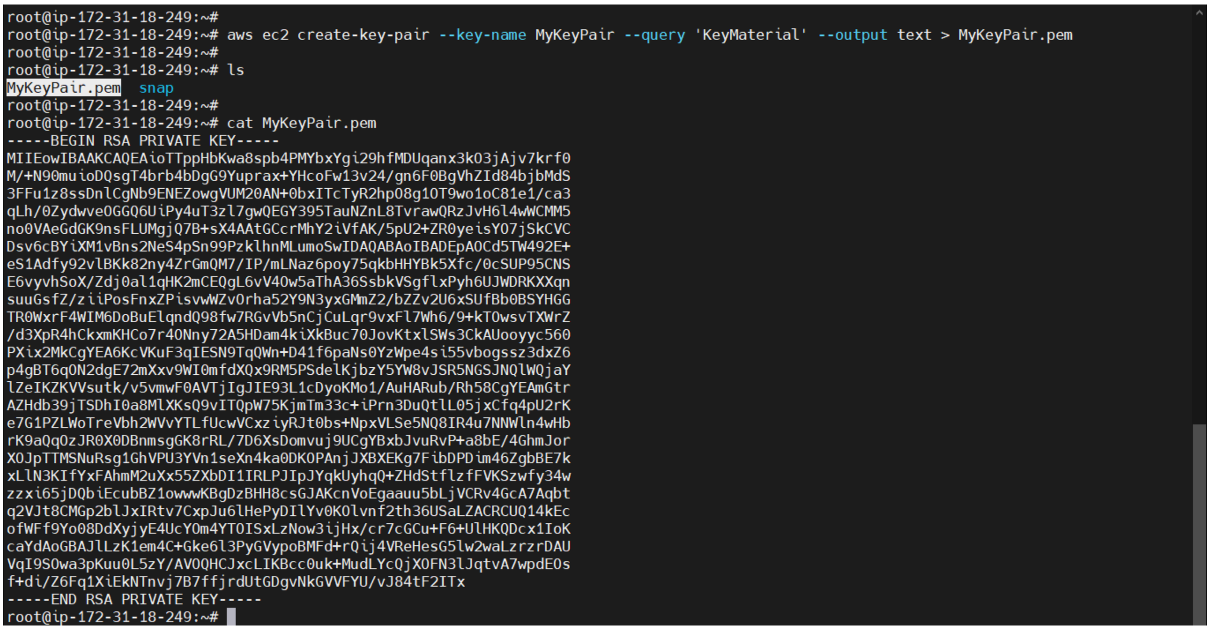
```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws configure
AWS Access Key ID [None]: AKIAUL6SJKNF67VYL5BR
AWS Secret Access Key [None]: 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
Default region name [us-east-2]: us-east-2
Default output format [None]: json
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws s3 ls
root@ip-172-31-18-249:~# 
```

Once it is done try some aws cli commands like *# aws s3 ls*
If u have any buckets in your s3 it will list

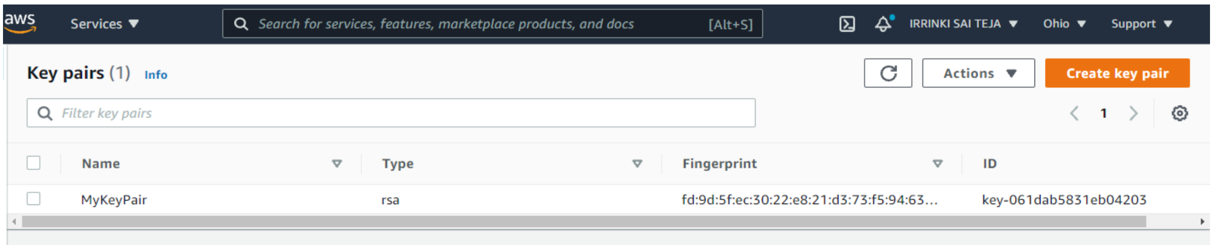
EC2 – Elastic Compute Cloud

Create a key pair

# aws ec2 create-key-pair --key-name <keypair-Name> --query 'KeyMaterial' --output text > <keypair-Name.pem>



```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text > MyKeyPair.pem
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# ls
MyKeyPair.pem  snap
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# cat MyKeyPair.pem
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAioTTppHbKwa8spb4PMYbxYgi29hfMDUqanx3k03jAjv7krf0
M/+N90muioDQsgT4brb4bDgG9Yuprax+YHcoFw13v24/gn6F0BgVhZId84bjbMdS
3FFu1z8ssDnlCgNb9ENEZowgVUM20AN+0bxITcTyR2hp08g10T9wo1oC81e1/ca3
qLh/0ZydwveOGGQ6UiPy4uT3zl7gwQEGY395TauNZnL8TvrawQRzJvH6l4wWCMM5
no0VAeGdGK9nsFLUMgjQ7B+sX4AAtGCcrMhY2iVfAK/5pU2+ZR0yeisYO7jSkCVC
Dsv6cBYiXM1vBns2NeS4pSn99PzklhnMLumoSwIDAQABAoIBADEpAOCd5TW492E+
eS1Adfy92vlBKk82ny4ZrGmQM7/IP/mLNaz6poy75qkbHHYBk5Xfc/0cSUP95CNS
E6vyvhSoX/Zdj0al1qHK2mCEQgL6vV40w5aThA36SsbkVSgflxPyh6UJWDRKXXqn
suuGsfZ/ziiPosFnxZPisvwWZv0rha52Y9N3yxGMmZ2/bZZv2U6xSUfBb0BSYHGG
TR0WxrF4WIM6DoBuElqndQ98fw7RGvVb5nCjCuLqr9vxFl7Wh6/9+kTOwsvTXWrZ
/d3XpR4hCkxmKHCo7r40Nny72A5HDam4kiXkBuc70JovKtxlSWs3CkAUooyyc560
PXix2MkCgYEA6KcVKuF3qIESN9TqQWn+D41f6paNs0YzWpe4si55vbogssz3dxZ6
p4gBT6qON2dgE72mXxv9WI0mfdXQx9RM5PSdelKjbzY5YW8vJSR5NGSJNQlWQjaY
lZeIKZKVVsutk/v5vmwF0AVTjIgJIE93L1cDyoKMo1/AuHARub/Rh58CgYEAmGtr
AZHdb39jTSDhI0a8MlXKsQ9vITQpW75KjmTm33c+iPrn3DuQtlL05jxCfq4pU2rK
e7G1PZLWoTreVbh2WVvYTLfUcwVCxziyRJt0bs+NpxVLSe5NQ8IR4u7NNWln4wHb
rK9aQqOzJR0X0DBnmsgGK8rRL/7D6XsDomvuj9UCgYBxbJvuRvP+a8bE/4GhmJor
XOJpTTMSNuRsg1GhVPU3YVn1seXn4ka0DKOPAnjJXBXEKg7FibDPDim46ZgbBE7k
xLlN3KIfYxFAhmM2uXx55ZXbDI1IRLPJIpJYqkUyhqQ+ZHdStflzfFVKSzwfy34w
zzxi65jDQbiEcubBZ1owwwKBgDzBHH8csGJAKcnVoEgaauu5bLjVCRv4GcA7Aqbt
q2VJt8CMGp2blJxIRtv7CxpJu6lHePyDIlYv0KOlvnf2th36USaLZACRCUQ14kEc
ofWFf9Yo08DdXyjyE4UcYOm4YTOISxLzNow3ijHx/cr7cGCu+F6+UlHKQDcx1IoK
caYdAoGBAJlLzK1em4C+Gke6l3PyGVypoBMFd+rQij4VReHesG5lw2waLzrzrDAU
VqI9SOwa3pKuu0L5zY/AVOQHCJxcLIKBcc0uk+MudLYcQjXOFN3lJqtvA7wpdEOs
f+di/Z6Fq1XiEkNTnvj7B7ffjrdUtGDgvNkGVVFYU/vJ84tF2ITx
-----END RSA PRIVATE KEY-----
root@ip-172-31-18-249:~#
```



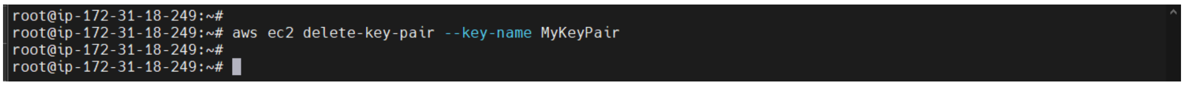
Key pairs (1) Info

□	Name	Type	Fingerprint	ID
□	MyKeyPair	rsa	fd:9d:5f:ec:30:22:e8:21:d3:73:f5:94:63...	key-061dab5831eb04203

Delete a key pair

To delete a key pair, run the aws ec2 delete-key-pair command, substituting MyKeyPair with the name of the pair to delete.

# aws ec2 delete-key-pair --key-name <keypair-Name>



```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws ec2 delete-key-pair --key-name MyKeyPair
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~#
```

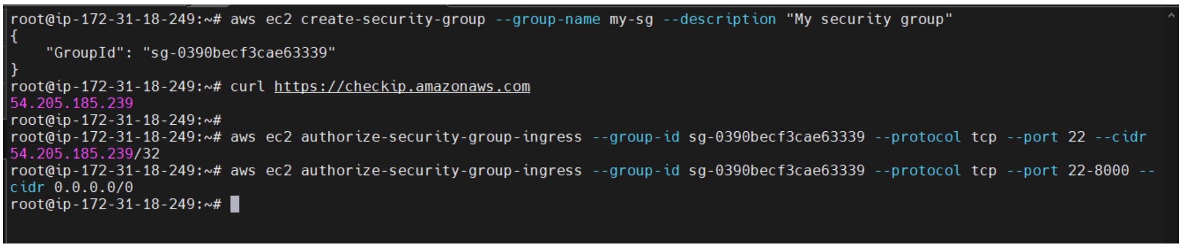
Create a Security Group & Adding Inbound rules

# aws ec2 create-security-group --group-name <security grp Name> --description "<Description>"

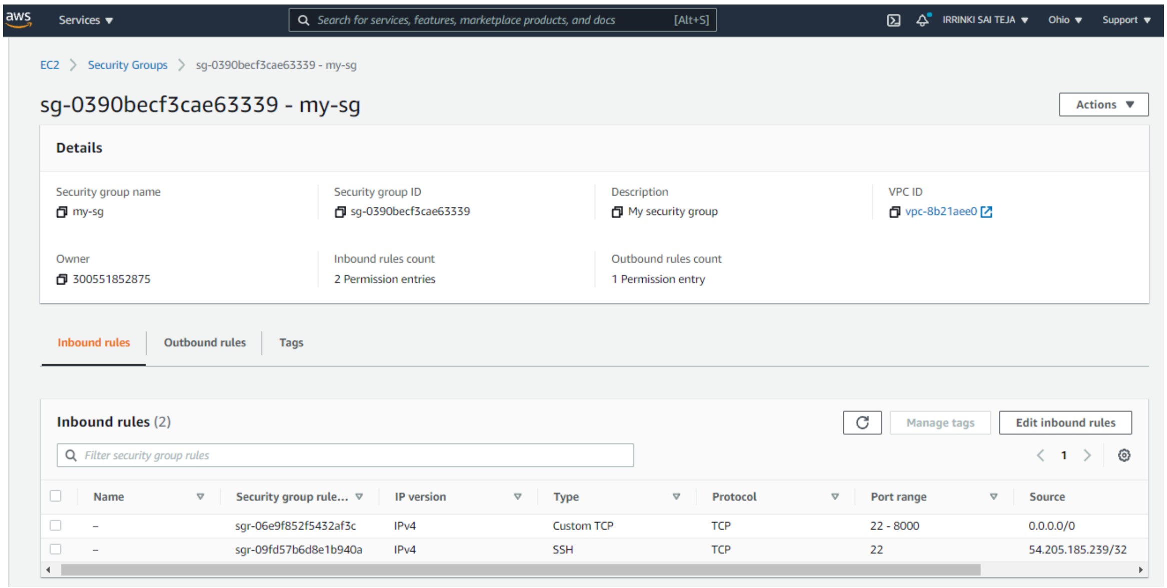
# curl https://checkip.amazonaws.com

# aws ec2 authorize-security-group-ingress --group-id <security group Id> --protocol tcp --port <port Number> --cidr <ip address>

# aws ec2 authorize-security-group-ingress --group-id <security grp Id>--protocol tcp --port 22-8000 --cidr 0.0.0.0/0



```
root@ip-172-31-18-249:~# aws ec2 create-security-group --group-name my-sg --description "My security group"
{
    "GroupId": "sg-0390becf3cae63339"
}
root@ip-172-31-18-249:~# curl https://checkip.amazonaws.com
54.205.185.239
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws ec2 authorize-security-group-ingress --group-id sg-0390becf3cae63339 --protocol tcp --port 22 --cidr
54.205.185.239/32
root@ip-172-31-18-249:~# aws ec2 authorize-security-group-ingress --group-id sg-0390becf3cae63339 --protocol tcp --port 22-8000 --
cidr 0.0.0.0/0
root@ip-172-31-18-249:~# 
```



aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
IRRINKI SAI TEJA ▼
Ohio ▼
Support ▼

EC2 > Security Groups > sg-0390becf3cae63339 - my-sg

sg-0390becf3cae63339 - my-sg

Details

Security group name: my-sg

Security group ID: sg-0390becf3cae63339

Description: My security group

VPC ID: vpc-8b21aee0

Owner: 300551852875

Inbound rules count: 2 Permission entries

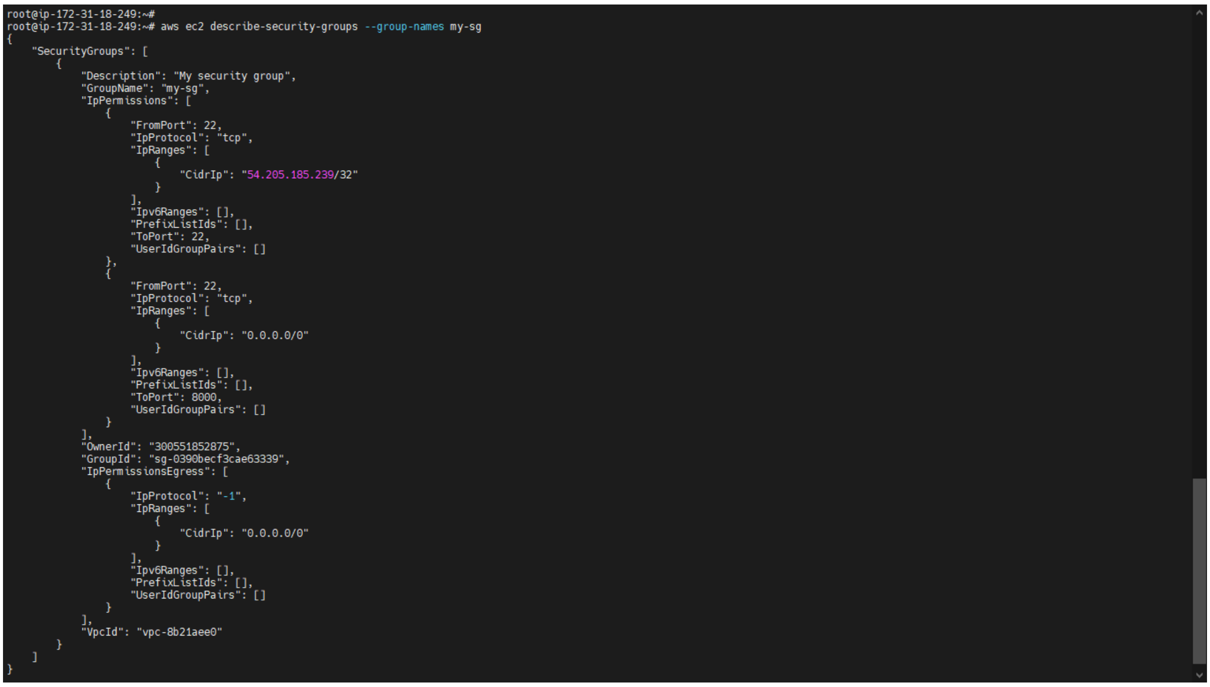
Outbound rules count: 1 Permission entry

Inbound rules (2)

□	Name	Security group rule...	IP version	Type	Protocol	Port range	Source
□	–	sgr-06e9f852f5432af3c	IPv4	Custom TCP	TCP	22 - 8000	0.0.0.0/0
□	–	sgr-09fd57b6d8e1b940a	IPv4	SSH	TCP	22	54.205.185.239/32

To view the initial information for my-sg, run the aws ec2 describe-security-groups command. For an EC2-Classic security group, you can reference it by its name.

# aws ec2 describe-security-groups --group-names <security grp Name>

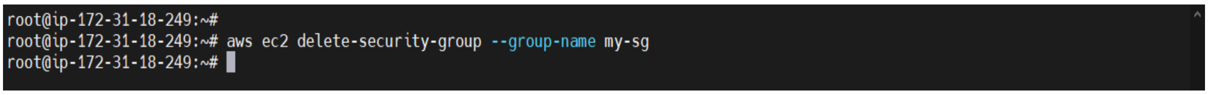


```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws ec2 describe-security-groups --group-names my-sg
{
    "SecurityGroups": [
        {
            "Description": "My security group",
            "GroupName": "my-sg",
            "IpPermissions": [
                {
                    "FromPort": 22,
                    "IpProtocol": "tcp",
                    "IpRanges": [
                        {
                            "CidrIp": "54.205.185.239/32"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": [],
                    "ToPort": 22,
                    "UserIdGroupPairs": []
                },
                {
                    "FromPort": 22,
                    "IpProtocol": "tcp",
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": [],
                    "ToPort": 8000,
                    "UserIdGroupPairs": []
                }
            ],
            "OwnerId": "300551852875",
            "GroupId": "sg-0390becf3cae63339",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": [],
                    "UserIdGroupPairs": []
                }
            ],
            "VpcId": "vpc-8b21aee0"
        }
    ]
}
```

Delete your security group

The following command example deletes the EC2-Classic security group named.

# aws ec2 delete-security-group --group-name <security grp Name>



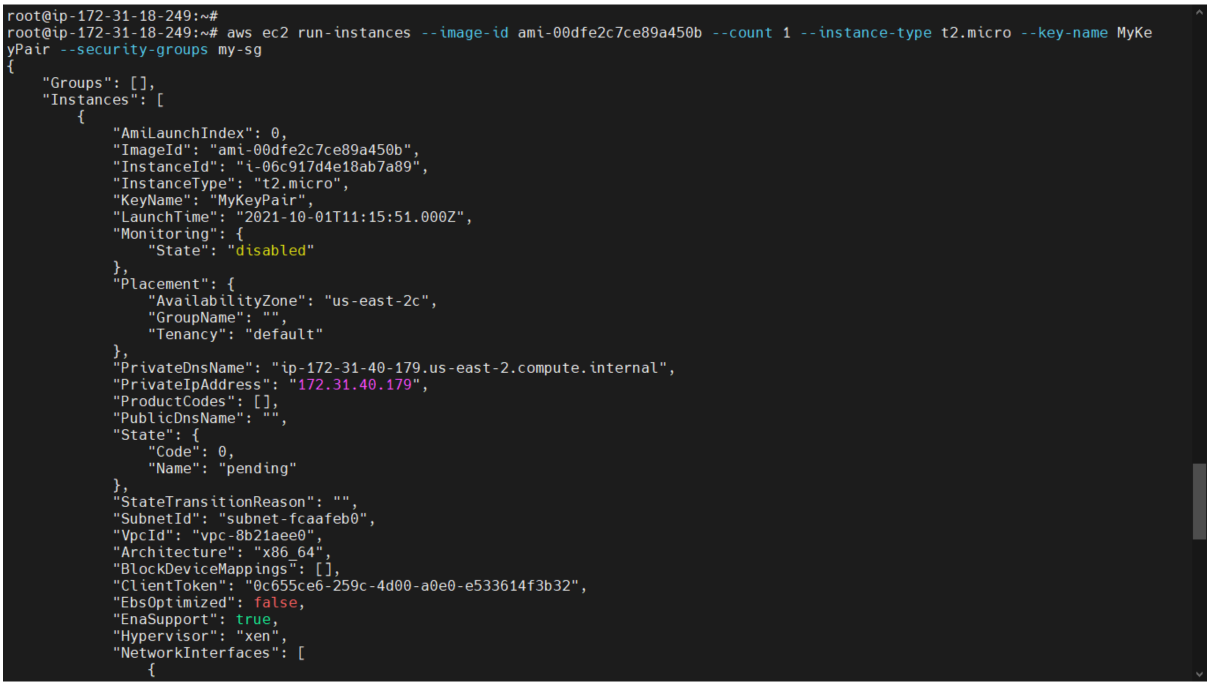
```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws ec2 delete-security-group --group-name my-sg
root@ip-172-31-18-249:~#
```

Launch Instance

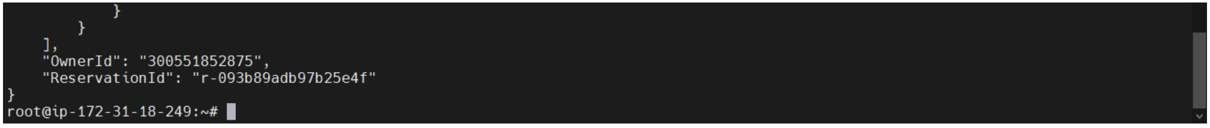
You can use the following command to launch a t2.micro instance in EC2-Classic. Replace the italicized parameter values with your own.

You can get the ami Id’s from documentation or console for your required Instance.

# aws ec2 run-instances --image-id <ami-Id> --count 1 --instance-type <type> --key-name <keypair-Name> --security-groups <security grp Name>



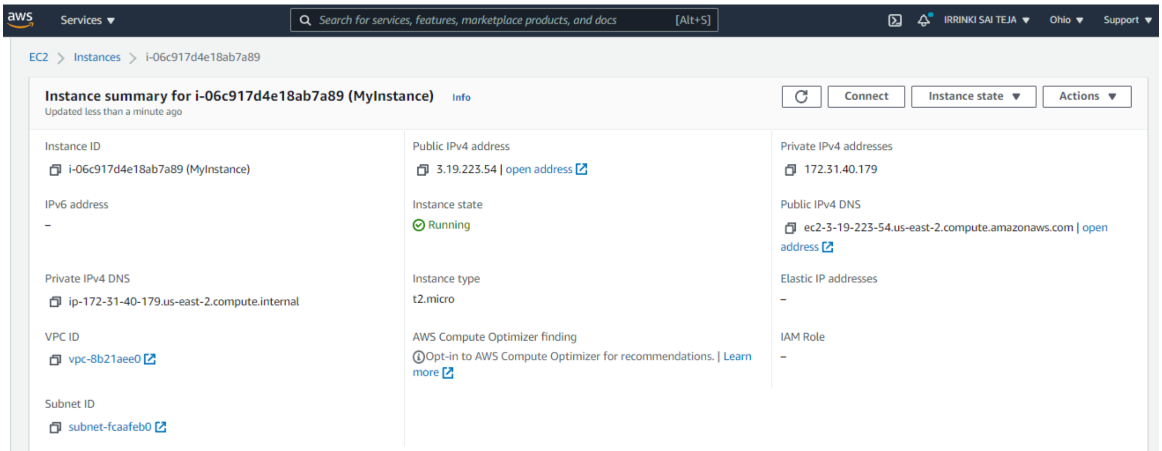
```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws ec2 run-instances --image-id ami-00dfe2c7ce89a450b --count 1 --instance-type t2.micro --key-name MyKe
yPair --security-groups my-sg
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-00dfe2c7ce89a450b",
            "InstanceId": "i-06c917d4e18ab7a89",
            "InstanceType": "t2.micro",
            "KeyName": "MyKeyPair",
            "LaunchTime": "2021-10-01T11:15:51.000Z",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-2c",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-172-31-40-179.us-east-2.compute.internal",
            "PrivateIpAddress": "172.31.40.179",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "StateTransitionReason": "",
            "SubnetId": "subnet-fcaafeb0",
            "VpcId": "vpc-8b21aee0",
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "0c655ce6-259c-4d00-a0e0-e533614f3b32",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
```



```
            }
        }
    ],
    "OwnerId": "300551852875",
    "ReservationId": "r-093b89adb97b25e4f"
}
root@ip-172-31-18-249:~#
```

Add a tag to your Instance

# aws ec2 create-tags --resources <Instance-Id>--tags Key=Name,Value=<value>


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
IRRINKI SAI TEJA ▼
Ohio ▼
Support ▼
EC2 > Instances > i-06c917d4e18ab7a89
Info
Connect
Instance state ▼
Actions ▼

Instance summary for i-06c917d4e18ab7a89 (MyInstance)

Updated less than a minute ago

Instance ID: i-06c917d4e18ab7a89 (MyInstance)

Public IPv4 address: 3.19.223.54 | open address

Private IPv4 addresses: 172.31.40.179

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-3-19-223-54.us-east-2.compute.amazonaws.com | open address

Private IPv4 DNS: ip-172-31-40-179.us-east-2.compute.internal

Instance type: t2.micro

Elastic IP addresses: –

VPC ID: vpc-8b21aee0

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

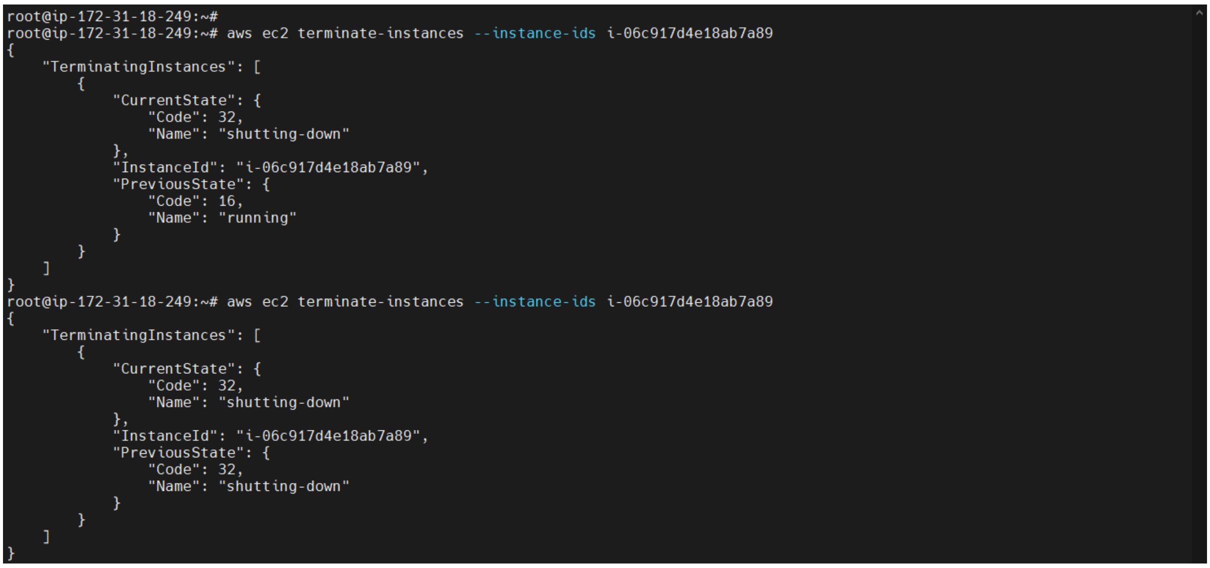
IAM Role: –

Subnet ID: subnet-fcaafeb0

Terminate your Instance

To delete an instance, you use the command aws ec2 terminate-instances to delete it.

# aws ec2 terminate-instances --instance-ids <Instance-Id>

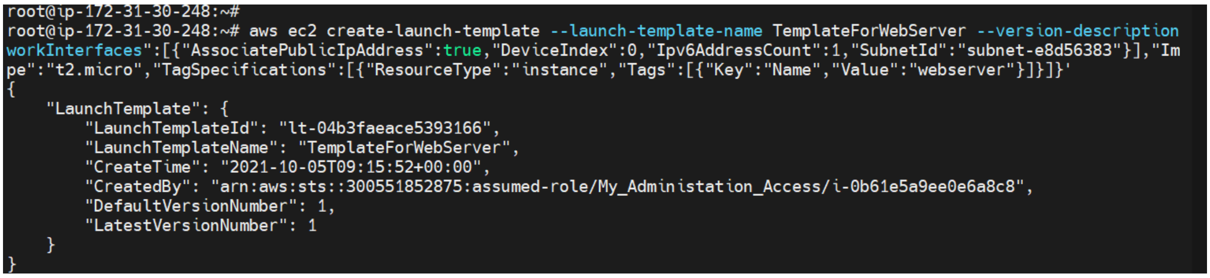


```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws ec2 terminate-instances --instance-ids i-06c917d4e18ab7a89
{
    "TerminatingInstances": [
        {
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "InstanceId": "i-06c917d4e18ab7a89",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
root@ip-172-31-18-249:~# aws ec2 terminate-instances --instance-ids i-06c917d4e18ab7a89
{
    "TerminatingInstances": [
        {
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "InstanceId": "i-06c917d4e18ab7a89",
            "PreviousState": {
                "Code": 32,
                "Name": "shutting-down"
            }
        }
    ]
}
```

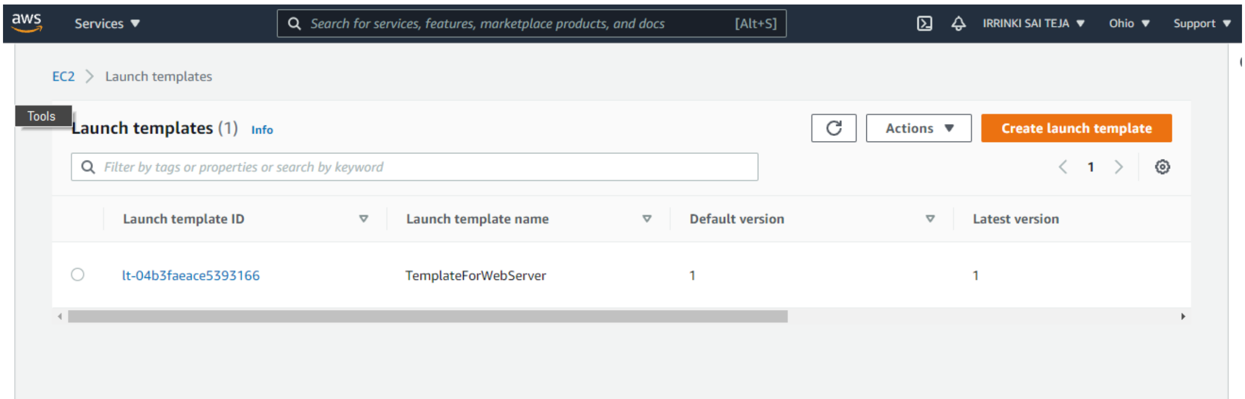
Create Launch Template

# aws ec2 create-launch-template --launch-template-name <Name>":[{"AssociatePublicIpAddress":true,"DeviceIndex":0,"Ipv6AddressCount":1," SubnetId":"pe":"<Instance type","TagSpecifications":[{"ResourceType":"instance"," Tags":[{"Key":"Name","Value":"<value>"}]}]}

{



```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws ec2 create-launch-template --launch-template-name TemplateForWebServer --version-description
workInterfaces":[{"AssociatePublicIpAddress":true,"DeviceIndex":0,"Ipv6AddressCount":1,"SubnetId":"subnet-e8d56383"}],"Im
pe":"t2.micro","TagSpecifications":[{"ResourceType":"instance","Tags":[{"Key":"Name","Value":"webserver"}]}]}'
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-04b3faeace5393166",
        "LaunchTemplateName": "TemplateForWebServer",
        "CreateTime": "2021-10-05T09:15:52+00:00",
        "CreatedBy": "arn:aws:sts::300551852875:assumed-role/My_Administation_Access/i-0b61e5a9ee0e6a8c8",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
    }
}
```



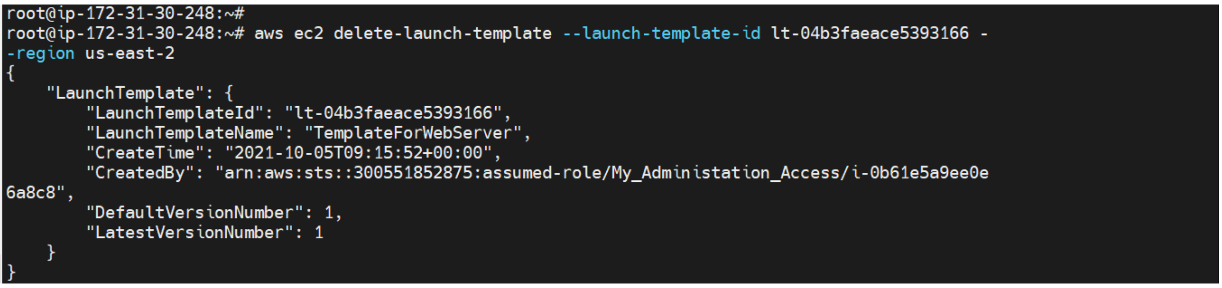
aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
IRRINKI SAI TEJA ▼
Ohio ▼
Support ▼
EC2 > Launch templates
Tools
Info
Actions ▼
Create launch template
Filter by tags or properties or search by keyword
1

Launch templates (1)

	Launch template ID	Launch template name	Default version	Latest version
○	lt-04b3faeace5393166	TemplateForWebServer	1	1

Delete Launch Template

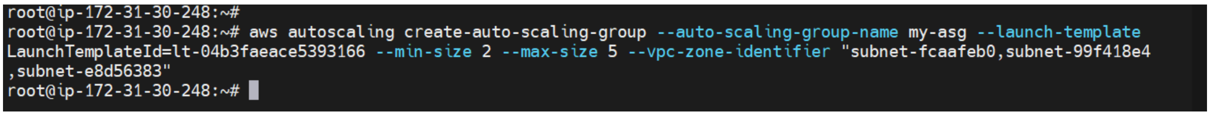
# aws ec2 delete-launch-template --launch-template-id < template id> --region <region>



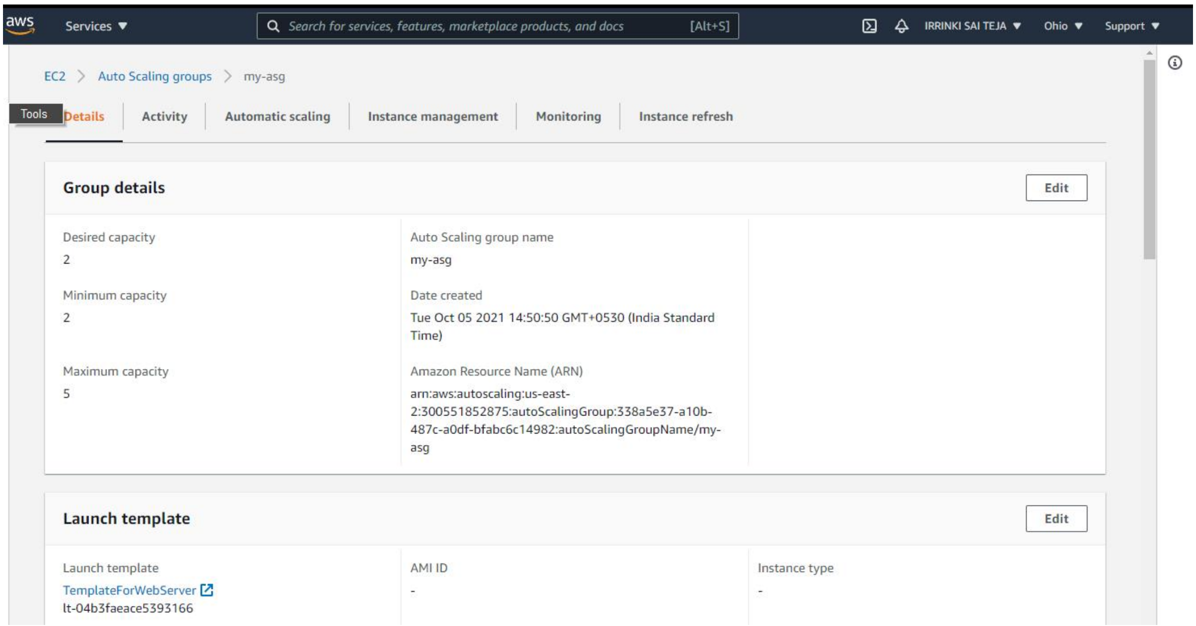
```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws ec2 delete-launch-template --launch-template-id lt-04b3faeace5393166 -
-region us-east-2
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-04b3faeace5393166",
        "LaunchTemplateName": "TemplateForWebServer",
        "CreateTime": "2021-10-05T09:15:52+00:00",
        "CreatedBy": "arn:aws:sts::300551852875:assumed-role/My_Administation_Access/i-0b61e5a9ee0e
6a8c8",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
    }
}
```

Creating Auto-Scaling group

# aws autoscaling create-auto-scaling-group --auto-scaling-group-name <Name> --launch-LaunchTemplateId=<template – id > --min-size 2 --max-size 5 --vpc-zone-identifier "subnet1-id,subnet2-id,subnet3-id"



```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws autoscaling create-auto-scaling-group --auto-scaling-group-name my-asg --launch-template
LaunchTemplateId=lt-04b3faeace5393166 --min-size 2 --max-size 5 --vpc-zone-identifier "subnet-fcaafeb0,subnet-99f418e4
,subnet-e8d56383"
root@ip-172-31-30-248:~#
```



Group details

Desired capacity
2

Minimum capacity
2

Maximum capacity
5

Auto Scaling group name
my-asg

Date created
Tue Oct 05 2021 14:50:50 GMT+0530 (India Standard Time)

Amazon Resource Name (ARN)
arn:aws:autoscaling:us-east-2:300551852875:autoScalingGroup:338a5e37-a10b-487c-a0df-bfabc6c14982:autoScalingGroupName/my-asg

Launch template

Launch template
TemplateForWebServer
lt-04b3faeace5393166

AMI ID
-

Instance type
-

Delete your Auto-Scaling Group

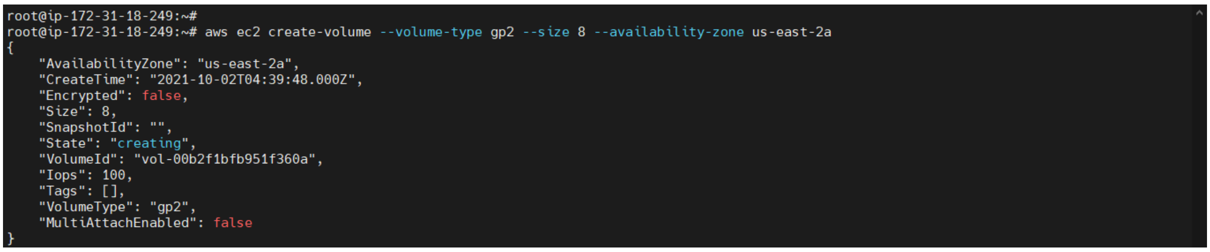
# aws autoscaling delete-auto-scaling-group --auto-scaling-group-name < Auto -Scaling group Name >

EBS – Elastic Block Storage

Create EBS Volume

To create an empty General Purpose SSD (gp2) volume

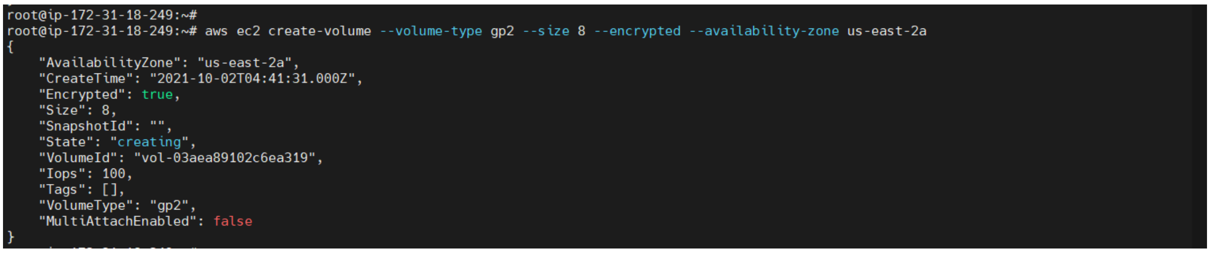
# aws ec2 create-volume --volume-type <volume type> --size <size in number> --availability-zone <zone>



```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws ec2 create-volume --volume-type gp2 --size 8 --availability-zone us-east-2a
{
    "AvailabilityZone": "us-east-2a",
    "CreateTime": "2021-10-02T04:39:48.000Z",
    "Encrypted": false,
    "Size": 8,
    "SnapshotId": "",
    "State": "creating",
    "VolumeId": "vol-00b2f1bfb951f360a",
    "Iops": 100,
    "Tags": [],
    "VolumeType": "gp2",
    "MultiAttachEnabled": false
}
```

To create an encrypted volume

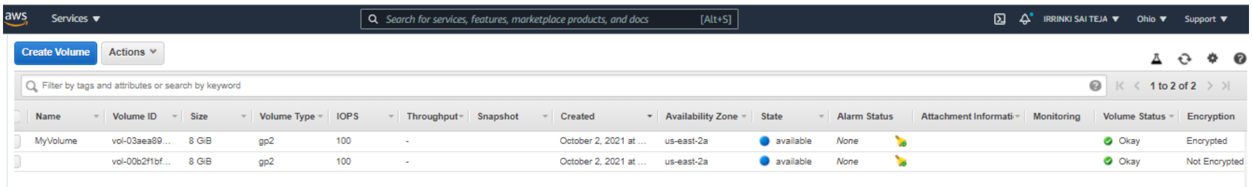
# aws ec2 create-volume --volume-type <volume type> --size <size in number> --encrypted --availability-zone <zone>



```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws ec2 create-volume --volume-type gp2 --size 8 --encrypted --availability-zone us-east-2a
{
    "AvailabilityZone": "us-east-2a",
    "CreateTime": "2021-10-02T04:41:31.000Z",
    "Encrypted": true,
    "Size": 8,
    "SnapshotId": "",
    "State": "creating",
    "VolumeId": "vol-03aea89102c6ea319",
    "Iops": 100,
    "Tags": [],
    "VolumeType": "gp2",
    "MultiAttachEnabled": false
}
```

To create a volume with tags

# aws ec2 create-tags --resources <volume-id> --tags Key=Name,Value=<value>



aws
Services
Search for services, features, marketplace products, and docs
[Alt+S]
IRRINKI SAI TEJA
Ohio
Support
Create Volume
Actions
Filter by tags and attributes or search by keyword
1 to 2 of 2

Name	Volume ID	Size	Volume Type	IOPS	Throughput	Snapshot	Created	Availability Zone	State	Alarm Status	Attachment Informati	Monitoring	Volume Status	Encryption
MyVolume	vol-03aea89...	8 GiB	gp2	100	-		October 2, 2021 at ...	us-east-2a	● available	None			✓ Okay	Encrypted
	vol-00b2f1bf...	8 GiB	gp2	100	-		October 2, 2021 at ...	us-east-2a	● available	None			✓ Okay	Not Encrypted

To Delete a Volume

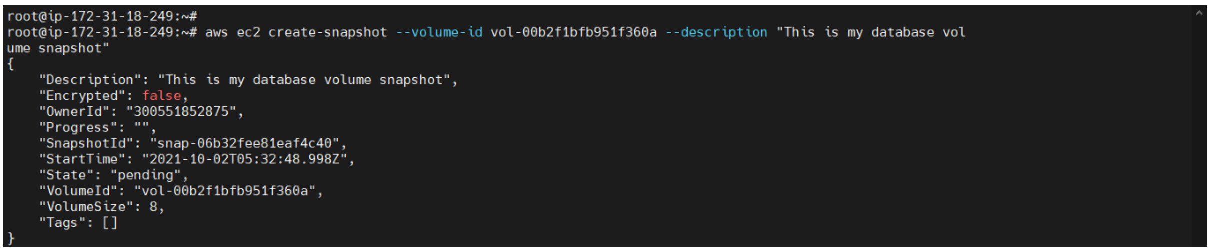
# aws ec2 delete-volume --volume-id <volume Id>

Output: None

To create a snapshot

This example command creates a snapshot of the volume with a volume ID of <volume Id> and a short description to identify the snapshot.

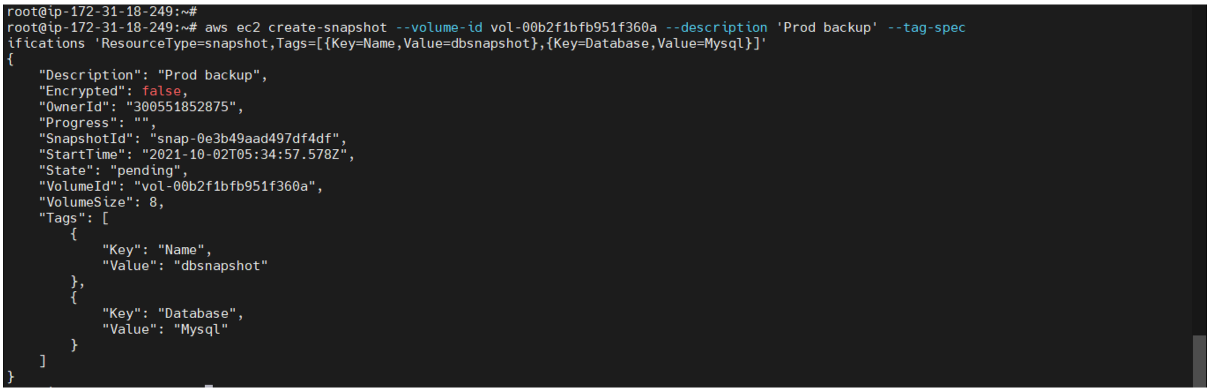
# aws ec2 create-snapshot --volume-id <volume Id> --description "<Description>"



```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws ec2 create-snapshot --volume-id vol-00b2f1bfb951f360a --description "This is my database vol
ume snapshot"
{
    "Description": "This is my database volume snapshot",
    "Encrypted": false,
    "OwnerId": "300551852875",
    "Progress": "",
    "SnapshotId": "snap-06b32fee81eaf4c40",
    "StartTime": "2021-10-02T05:32:48.998Z",
    "State": "pending",
    "VolumeId": "vol-00b2f1bfb951f360a",
    "VolumeSize": 8,
    "Tags": []
}
```

To create a snapshot with tags

# aws ec2 create-snapshot --volume-id <volume Id> --description 'Prod backup' --tag-specifications 'ResourceType=snapshot,Tags=[{Key=Name,Value=<value>},{Key=Database,Value =Mysql}]'

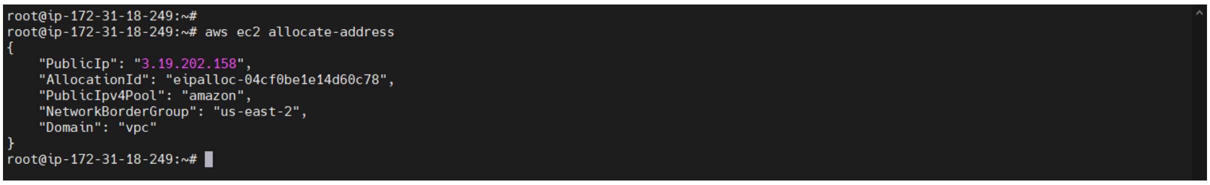


```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws ec2 create-snapshot --volume-id vol-00b2f1bfb951f360a --description 'Prod backup' --tag-spec
ifications 'ResourceType=snapshot,Tags=[{Key=Name,Value=dbsnapshot},{Key=Database,Value=Mysql}]'
{
    "Description": "Prod backup",
    "Encrypted": false,
    "OwnerId": "300551852875",
    "Progress": "",
    "SnapshotId": "snap-0e3b49aad497df4df",
    "StartTime": "2021-10-02T05:34:57.578Z",
    "State": "pending",
    "VolumeId": "vol-00b2f1bfb951f360a",
    "VolumeSize": 8,
    "Tags": [
        {
            "Key": "Name",
            "Value": "dbsnapshot"
        },
        {
            "Key": "Database",
            "Value": "Mysql"
        }
    ]
}
```

To allocate an Elastic IP address for EC2-Classic

The following allocate-address example allocates an Elastic IP address to use with an instance in EC2-Classic.

# aws ec2 allocate-address



```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws ec2 allocate-address
{
    "PublicIp": "3.19.202.158",
    "AllocationId": "eipalloc-04cf0be1e14d60c78",
    "PublicIpv4Pool": "amazon",
    "NetworkBorderGroup": "us-east-2",
    "Domain": "vpc"
}
root@ip-172-31-18-249:~#
```

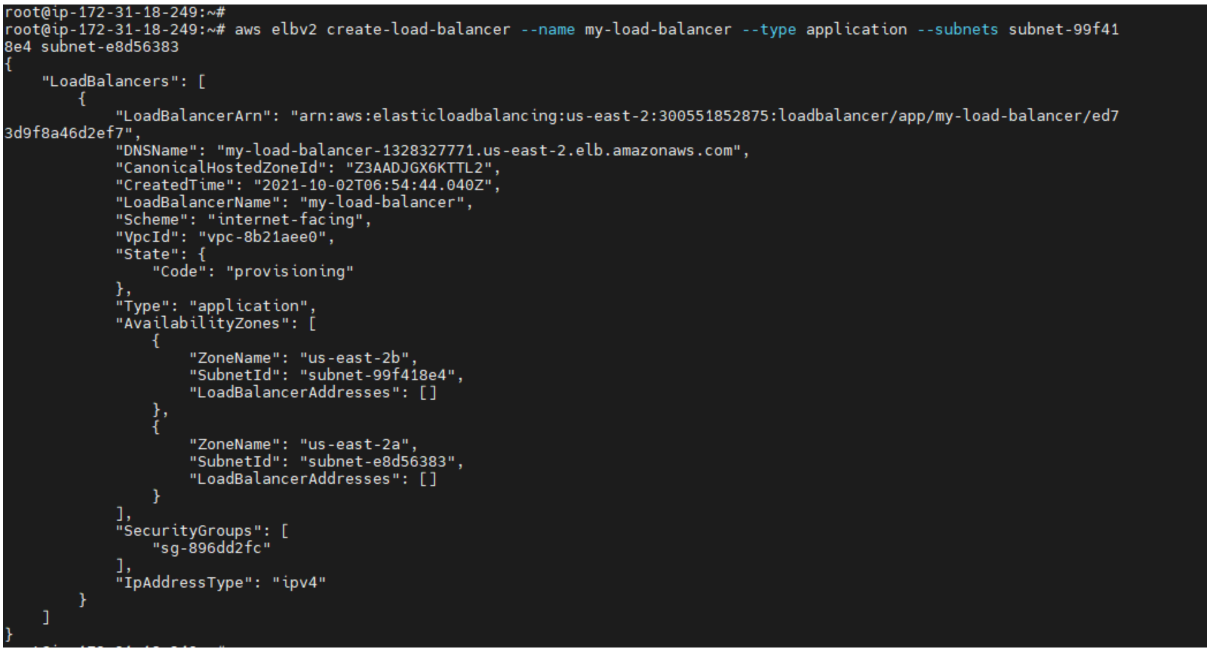
ELB – Elastic Load Balancer

Create-load-balancer

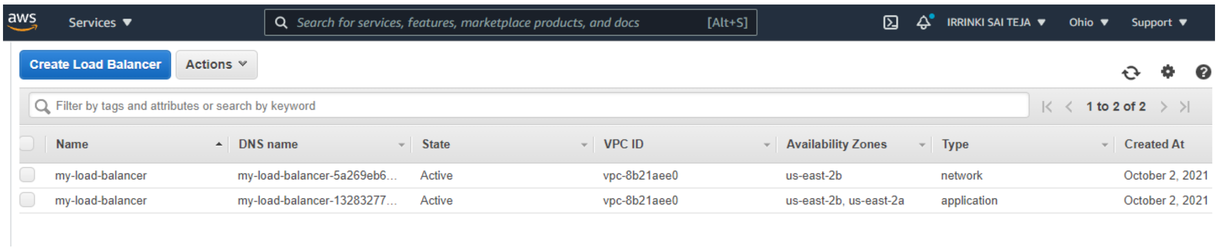
To create an Application load balancer

The below commands to find subnet id & Instance Id
# aws ec2 describe-subnets
# aws ec2 describe-instances

# aws elbv2 create-load-balancer --name <Load balancer Name>--type <type> --subnets <subnet-Id> <subnet-Id>



```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws elbv2 create-load-balancer --name my-load-balancer --type application --subnets subnet-99f41
8e4 subnet-e8d56383
{
    "LoadBalancers": [
        {
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-east-2:300551852875:loadbalancer/app/my-load-balancer/ed7
3d9f8a46d2ef7",
            "DNSName": "my-load-balancer-1328327771.us-east-2.elb.amazonaws.com",
            "CanonicalHostedZoneId": "Z3AADJGX6KTTL2",
            "CreatedTime": "2021-10-02T06:54:44.040Z",
            "LoadBalancerName": "my-load-balancer",
            "Scheme": "internet-facing",
            "VpcId": "vpc-8b21aee0",
            "State": {
                "Code": "provisioning"
            },
            "Type": "application",
            "AvailabilityZones": [
                {
                    "ZoneName": "us-east-2b",
                    "SubnetId": "subnet-99f418e4",
                    "LoadBalancerAddresses": []
                },
                {
                    "ZoneName": "us-east-2a",
                    "SubnetId": "subnet-e8d56383",
                    "LoadBalancerAddresses": []
                }
            ],
            "SecurityGroups": [
                "sg-896dd2fc"
            ],
            "IpAddressType": "ipv4"
        }
    ]
}
```

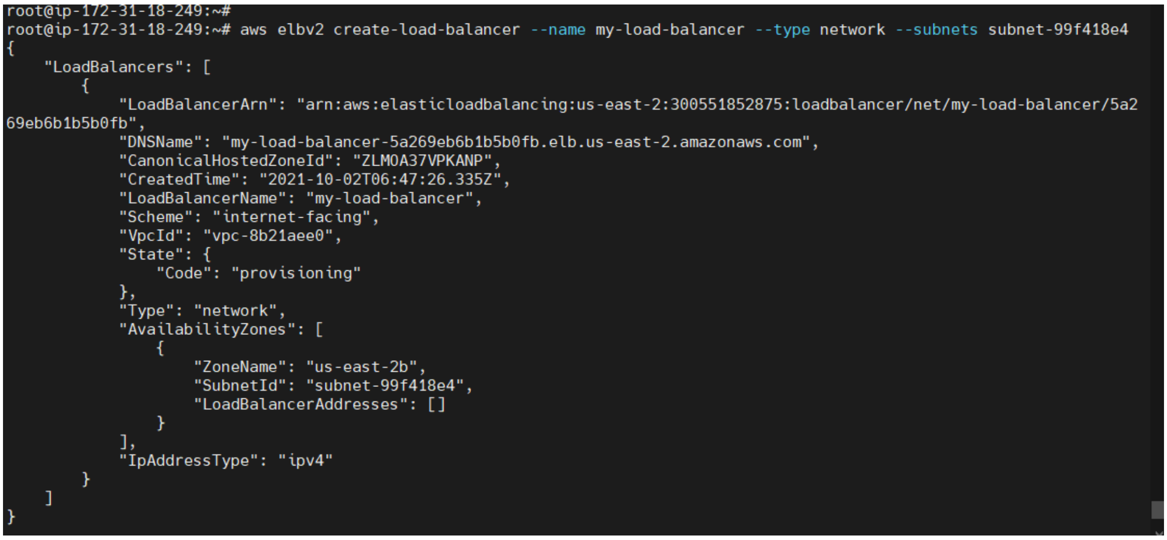


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
IRRINKI SAI TEJA ▼
Ohio ▼
Support ▼
Create Load Balancer
Actions ▼
Filter by tags and attributes or search by keyword
1 to 2 of 2

	Name	DNS name	State	VPC ID	Availability Zones	Type	Created At
□	my-load-balancer	my-load-balancer-5a269eb6...	Active	vpc-8b21aee0	us-east-2b	network	October 2, 2021
□	my-load-balancer	my-load-balancer-13283277...	Active	vpc-8b21aee0	us-east-2b, us-east-2a	application	October 2, 2021

To create an Network load balancer

```
# aws elbv2 create-load-balancer --name <Load balancer Name>--type type --subnets <subnet-Id>
```



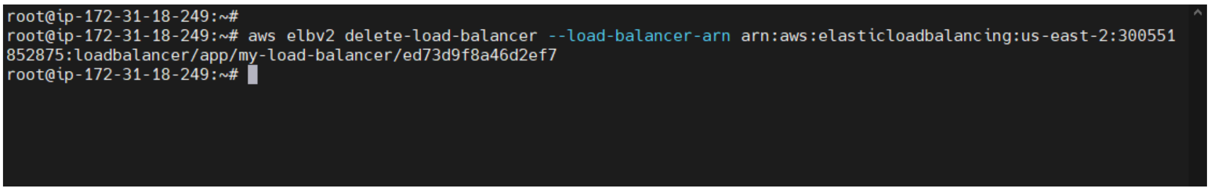
```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws elbv2 create-load-balancer --name my-load-balancer --type network --subnets subnet-99f418e4
{
    "LoadBalancers": [
        {
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-east-2:300551852875:loadbalancer/net/my-load-balancer/5a2
69eb6b1b5b0fb",
            "DNSName": "my-load-balancer-5a269eb6b1b5b0fb.elb.us-east-2.amazonaws.com",
            "CanonicalHostedZoneId": "ZLMOA37VPKANP",
            "CreatedTime": "2021-10-02T06:47:26.335Z",
            "LoadBalancerName": "my-load-balancer",
            "Scheme": "internet-facing",
            "VpcId": "vpc-8b21aee0",
            "State": {
                "Code": "provisioning"
            },
            "Type": "network",
            "AvailabilityZones": [
                {
                    "ZoneName": "us-east-2b",
                    "SubnetId": "subnet-99f418e4",
                    "LoadBalancerAddresses": []
                }
            ],
            "IpAddressType": "ipv4"
        }
    ]
}
```

To register instances with a load balancer

```
# aws elb register-instances-with-load-balancer --load-balancer-name <Load balancer Name> --instances <Instance-Id>
```

To Delete a Specific Load balancer

```
# aws elbv2 delete-load-balancer --load-balancer-arn <arn end point>
```

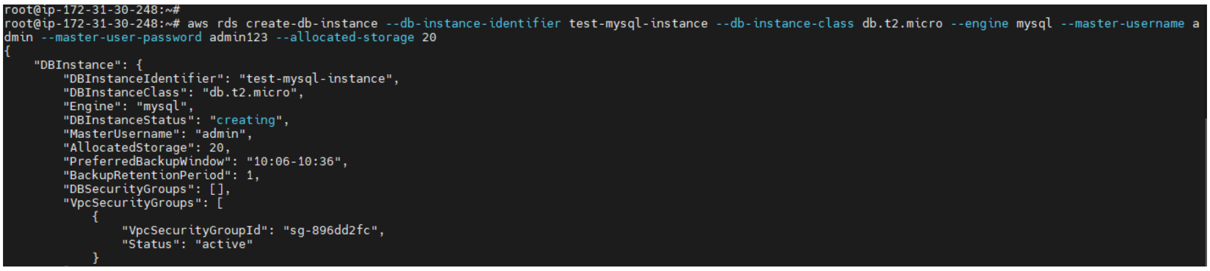


```
root@ip-172-31-18-249:~#
root@ip-172-31-18-249:~# aws elbv2 delete-load-balancer --load-balancer-arn arn:aws:elasticloadbalancing:us-east-2:300551
852875:loadbalancer/app/my-load-balancer/ed73d9f8a46d2ef7
root@ip-172-31-18-249:~#
```

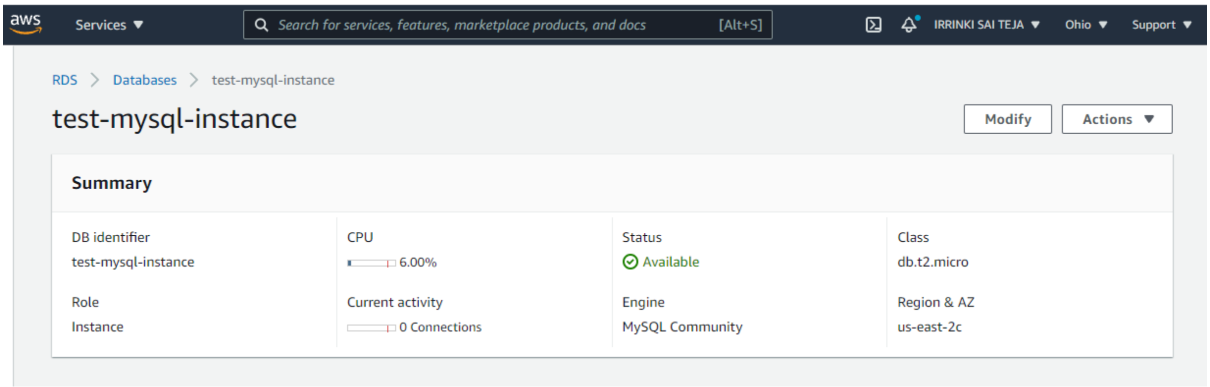
RDS

Create-db-Instance

# aws rds create-db-instance --db-instance-identifier <db - Name> --db-instance-class <db.type> --engine <Database Engine> --master-username <username> --master-user-password <password> --allocated-storage <storage in numbers>



```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws rds create-db-instance --db-instance-identifier test-mysql-instance --db-instance-class db.t2.micro --engine mysql --master-username a
dmin --master-user-password admin123 --allocated-storage 20
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t2.micro",
        "Engine": "mysql",
        "DBInstanceStatus": "creating",
        "MasterUsername": "admin",
        "AllocatedStorage": 20,
        "PreferredBackupWindow": "10:06-10:36",
        "BackupRetentionPeriod": 1,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-896dd2fc",
                "Status": "active"
            }
```


aws
Services ▼
Search for services, features, marketplace products, and docs [Alt+S]
IRRINKI SAI TEJA ▼
Ohio ▼
Support ▼
RDS > Databases > test-mysql-instance
Modify
Actions ▼

test-mysql-instance

Summary

DB identifier
test-mysql-instance

CPU
6.00%

Status
Available

Class
db.t2.micro

Role
Instance

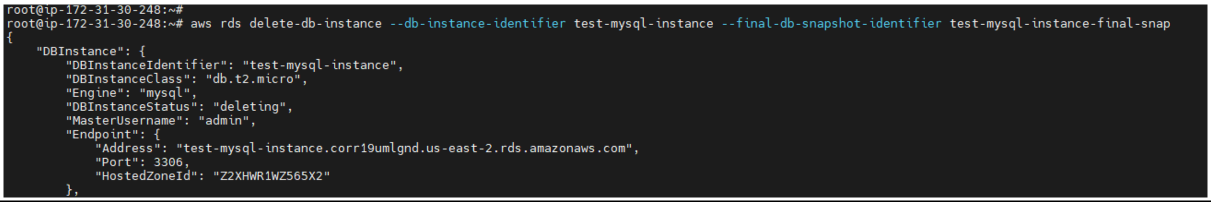
Current activity
0 Connections

Engine
MySQL Community

Region & AZ
us-east-2c

To delete your db-Instance

# aws rds delete-db-instance --db-instance-identifier <db - Name> --final-db-snapshot-identifier <db - Name>-final-snap



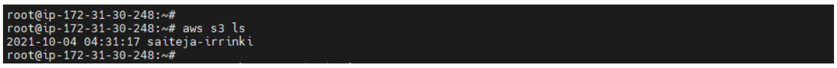
```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws rds delete-db-instance --db-instance-identifier test-mysql-instance --final-db-snapshot-identifier test-mysql-instance-final-snap
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t2.micro",
        "Engine": "mysql",
        "DBInstanceStatus": "deleting",
        "MasterUsername": "admin",
        "Endpoint": {
            "Address": "test-mysql-instance.corr19umlgnd.us-east-2.rds.amazonaws.com",
            "Port": 3306,
            "HostedZoneId": "Z2XHWR1WZ565X2"
        },
```

S3 – Simple Storage Service

List Buckets & Objects

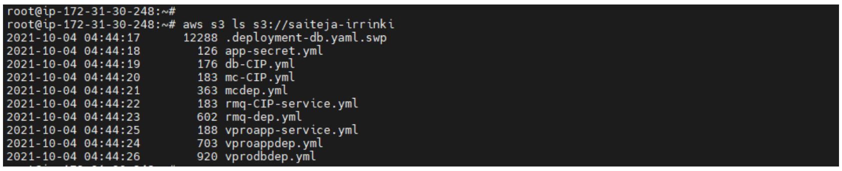
To list your buckets, folders, or objects, use the s3 ls command. Using the command without a target or options lists all buckets.

# aws s3 ls



```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws s3 ls
2021-10-04 04:31:17 saiteja-irrinki
root@ip-172-31-30-248:~#
```

# aws s3 ls s3://<bucket name>



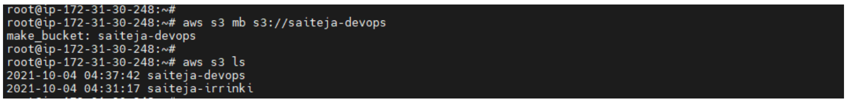
```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws s3 ls s3://saiteja-irrinki
2021-10-04 04:44:17      12288 .deployment-db.yaml.swp
2021-10-04 04:44:18        126 app-secret.yml
2021-10-04 04:44:19        176 db-CIP.yml
2021-10-04 04:44:20        183 mc-CIP.yml
2021-10-04 04:44:21        363 mcdep.yml
2021-10-04 04:44:22        183 rmq-CIP-service.yml
2021-10-04 04:44:23        602 rmq-dep.yml
2021-10-04 04:44:25        188 vproapp-service.yml
2021-10-04 04:44:24        703 vproappdep.yml
2021-10-04 04:44:26        920 vprodbdep.yml
```

Create a bucket

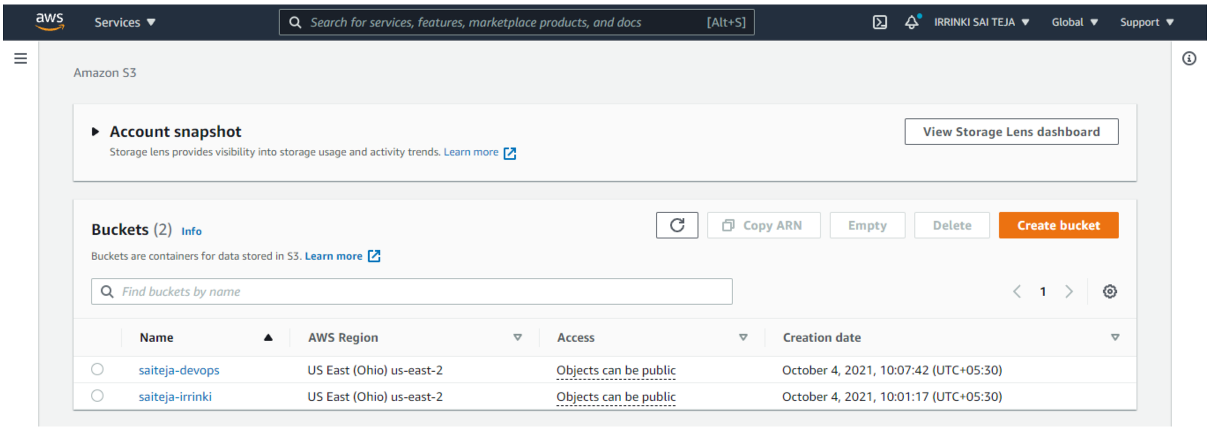
Use the s3 mb command to make a bucket. Bucket names must be ***globally*** unique (unique across all of Amazon S3) and should be DNS compliant.

Bucket names can contain lowercase letters, numbers, hyphens, and periods. Bucket names can start and end only with a letter or number, and cannot contain a period next to a hyphen or another period.

# aws s3 mb s3:// <bucket name>



```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws s3 mb s3://saiteja-devops
make_bucket: saiteja-devops
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws s3 ls
2021-10-04 04:37:42 saiteja-devops
2021-10-04 04:31:17 saiteja-irrinki
```



Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets (2) Info

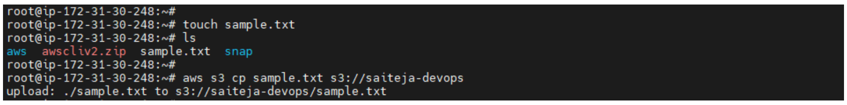
Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	saiteja-devops	US East (Ohio) us-east-2	Objects can be public	October 4, 2021, 10:07:42 (UTC+05:30)
○	saiteja-irrinki	US East (Ohio) us-east-2	Objects can be public	October 4, 2021, 10:01:17 (UTC+05:30)

Copy objects

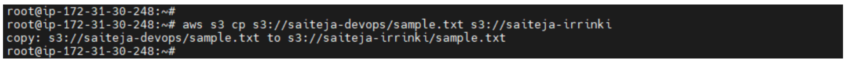
Use the s3 cp command to copy objects from a bucket or a local directory

# aws s3 cp sample.txt s3:// <bucket name>

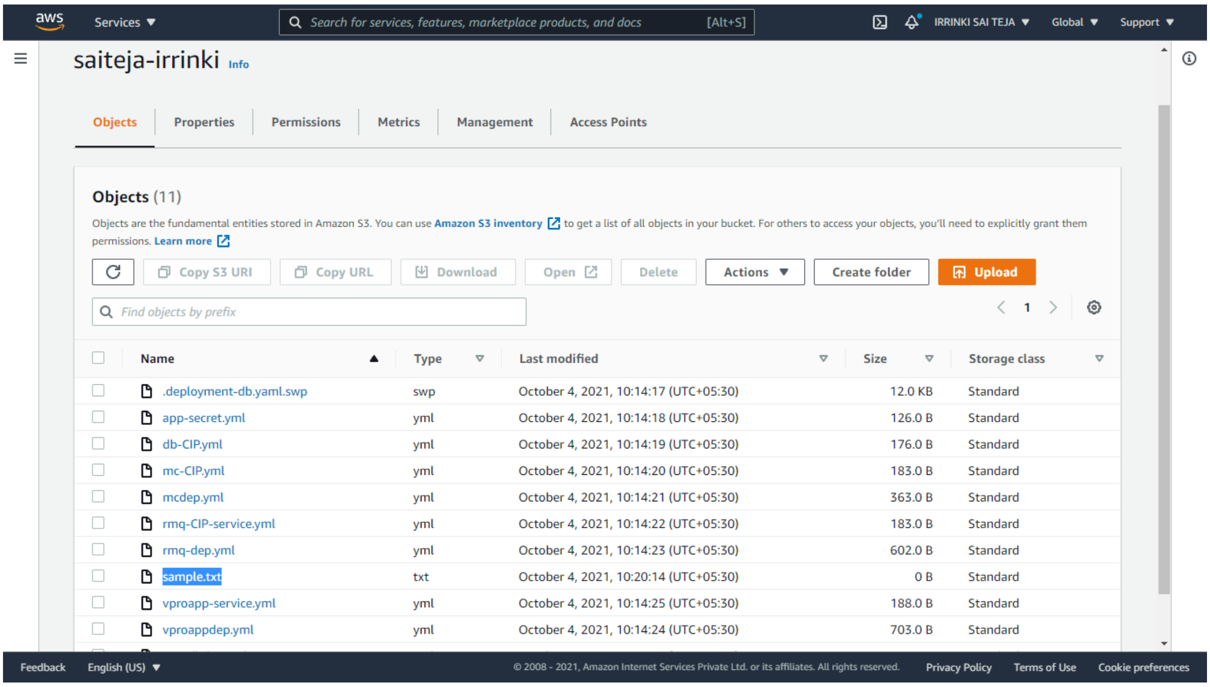


```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# touch sample.txt
root@ip-172-31-30-248:~# ls
aws  awscliv2.zip  sample.txt  snap
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws s3 cp sample.txt s3://saiteja-devops
upload: ./sample.txt to s3://saiteja-devops/sample.txt
```

# aws s3 cp s3://<source bucket/file> s3://<destination-bucket>



```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws s3 cp s3://saiteja-devops/sample.txt s3://saiteja-irrinki
copy: s3://saiteja-devops/sample.txt to s3://saiteja-irrinki/sample.txt
root@ip-172-31-30-248:~#
```



Objects (11)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

	Name	Type	Last modified	Size	Storage class
□	.deployment-db.yaml.swp	swp	October 4, 2021, 10:14:17 (UTC+05:30)	12.0 KB	Standard
□	app-secret.yml	yml	October 4, 2021, 10:14:18 (UTC+05:30)	126.0 B	Standard
□	db-CIP.yml	yml	October 4, 2021, 10:14:19 (UTC+05:30)	176.0 B	Standard
□	mc-CIP.yml	yml	October 4, 2021, 10:14:20 (UTC+05:30)	183.0 B	Standard
□	mcdep.yml	yml	October 4, 2021, 10:14:21 (UTC+05:30)	363.0 B	Standard
□	rmq-CIP-service.yml	yml	October 4, 2021, 10:14:22 (UTC+05:30)	183.0 B	Standard
□	rmq-dep.yml	yml	October 4, 2021, 10:14:23 (UTC+05:30)	602.0 B	Standard
□	sample.txt	txt	October 4, 2021, 10:20:14 (UTC+05:30)	0 B	Standard
□	vproapp-service.yml	yml	October 4, 2021, 10:14:25 (UTC+05:30)	188.0 B	Standard
□	vproappdep.yml	yml	October 4, 2021, 10:14:24 (UTC+05:30)	703.0 B	Standard

© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

Move objects

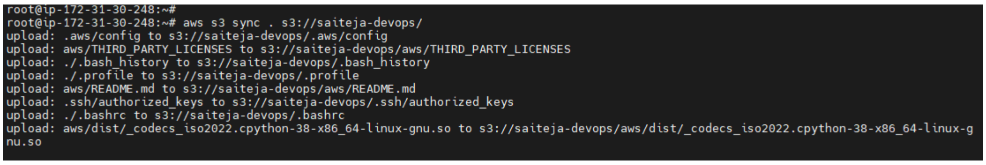
Use the s3 mv command to move objects from a bucket or a local directory.

# aws s3 mv <local file> s3:// <bucket name>

# aws s3 mv s3:// <source bucket/file> s3://<destination-bucket>

Sync Objects

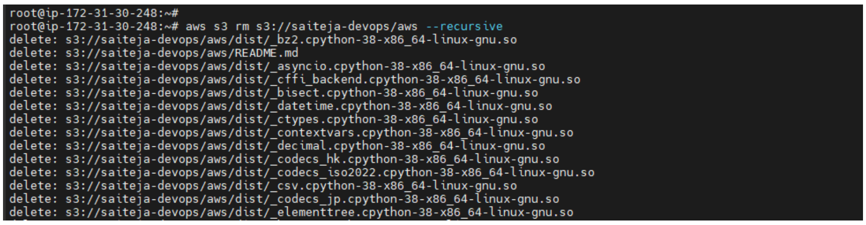
# aws s3 sync . s3://<bucket name>



```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws s3 sync . s3://saiteja-devops/
upload: .aws/config to s3://saiteja-devops/.aws/config
upload: aws/THIRD_PARTY_LICENSES to s3://saiteja-devops/aws/THIRD_PARTY_LICENSES
upload: ./.bash_history to s3://saiteja-devops/.bash_history
upload: ./.profile to s3://saiteja-devops/.profile
upload: aws/README.md to s3://saiteja-devops/aws/README.md
upload: .ssh/authorized_keys to s3://saiteja-devops/.ssh/authorized_keys
upload: ./.bashrc to s3://saiteja-devops/.bashrc
upload: aws/dist/_codecs_iso2022.cpython-38-x86_64-linux-gnu.so to s3://saiteja-devops/aws/dist/_codecs_iso2022.cpython-38-x86_64-linux-g
nu.so
```

Delete Objects

# aws s3 rm s3://<bucket name/file> --recursive



```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws s3 rm s3://saiteja-devops/aws --recursive
delete: s3://saiteja-devops/aws/dist/_bz2.cpython-38-x86_64-linux-gnu.so
delete: s3://saiteja-devops/aws/README.md
delete: s3://saiteja-devops/aws/dist/_asyncio.cpython-38-x86_64-linux-gnu.so
delete: s3://saiteja-devops/aws/dist/_cffi_backend.cpython-38-x86_64-linux-gnu.so
delete: s3://saiteja-devops/aws/dist/_bisect.cpython-38-x86_64-linux-gnu.so
delete: s3://saiteja-devops/aws/dist/_datetime.cpython-38-x86_64-linux-gnu.so
delete: s3://saiteja-devops/aws/dist/_ctypes.cpython-38-x86_64-linux-gnu.so
delete: s3://saiteja-devops/aws/dist/_contextvars.cpython-38-x86_64-linux-gnu.so
delete: s3://saiteja-devops/aws/dist/_decimal.cpython-38-x86_64-linux-gnu.so
delete: s3://saiteja-devops/aws/dist/_codecs_hk.cpython-38-x86_64-linux-gnu.so
delete: s3://saiteja-devops/aws/dist/_codecs_iso2022.cpython-38-x86_64-linux-gnu.so
delete: s3://saiteja-devops/aws/dist/_csv.cpython-38-x86_64-linux-gnu.so
delete: s3://saiteja-devops/aws/dist/_codecs_jp.cpython-38-x86_64-linux-gnu.so
delete: s3://saiteja-devops/aws/dist/_elementtree.cpython-38-x86_64-linux-gnu.so
```

Empty Bucket

# aws s3 rm s3://<bucket name> --recursive

Delete Bucket

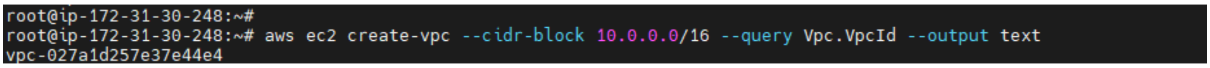
# aws s3 rb s3://<bucket name>

VPC – Virtual Private Cloud

To create a VPC and subnets using the AWS CLI

- Create a VPC with a 10.0.0.0/16 CIDR block using the following create-vpc command.

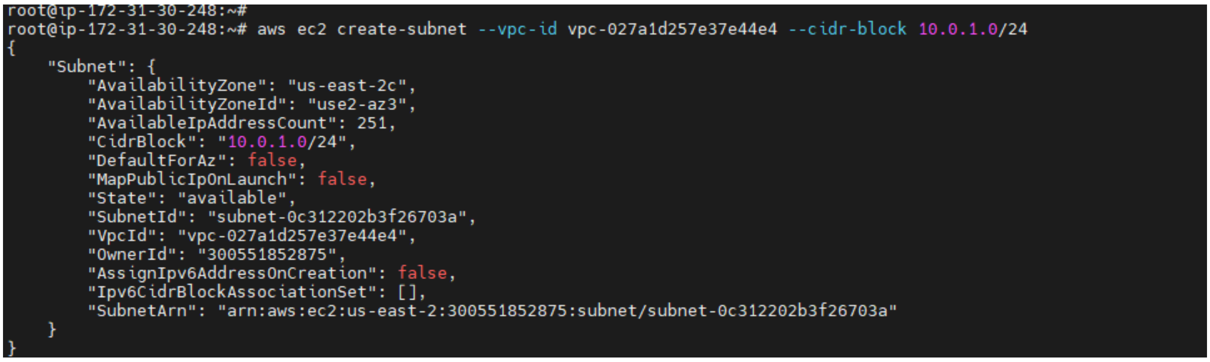
# aws ec2 create-vpc --cidr-block <Ip address> --query Vpc.VpcId --output text



```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws ec2 create-vpc --cidr-block 10.0.0.0/16 --query Vpc.VpcId --output text
vpc-027a1d257e37e44e4
```

- Using the VPC ID from the previous step, create a subnet with a 10.0.1.0/24 CIDR block using the following create-subnet command.

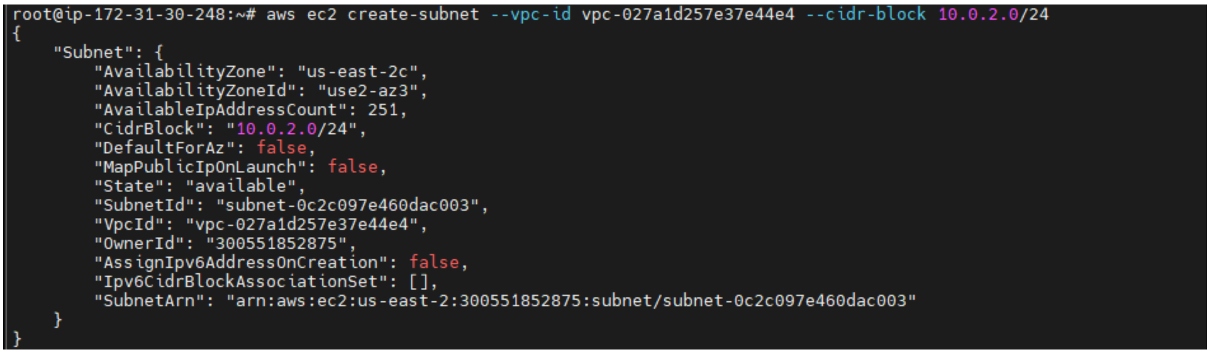
# aws ec2 create-subnet --vpc-id <vpc - Id>--cidr-block <Ip address>



```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws ec2 create-subnet --vpc-id vpc-027a1d257e37e44e4 --cidr-block 10.0.1.0/24
{
    "Subnet": {
        "AvailabilityZone": "us-east-2c",
        "AvailabilityZoneId": "use2-az3",
        "AvailableIpAddressCount": 251,
        "CidrBlock": "10.0.1.0/24",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false,
        "State": "available",
        "SubnetId": "subnet-0c312202b3f26703a",
        "VpcId": "vpc-027a1d257e37e44e4",
        "OwnerId": "300551852875",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [],
        "SubnetArn": "arn:aws:ec2:us-east-2:300551852875:subnet/subnet-0c312202b3f26703a"
    }
}
```

- Create a second subnet in your VPC with a 10.0.2.0/24 CIDR block.

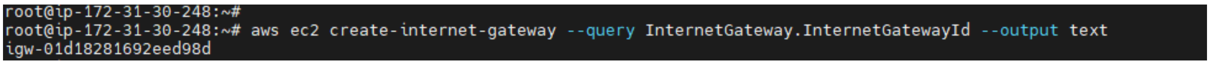
# aws ec2 create-subnet --vpc-id <vpc - Id>--cidr-block <Ip address>



```
root@ip-172-31-30-248:~# aws ec2 create-subnet --vpc-id vpc-027a1d257e37e44e4 --cidr-block 10.0.2.0/24
{
    "Subnet": {
        "AvailabilityZone": "us-east-2c",
        "AvailabilityZoneId": "use2-az3",
        "AvailableIpAddressCount": 251,
        "CidrBlock": "10.0.2.0/24",
        "DefaultForAz": false,
        "MapPublicIpOnLaunch": false,
        "State": "available",
        "SubnetId": "subnet-0c2c097e460dac003",
        "VpcId": "vpc-027a1d257e37e44e4",
        "OwnerId": "300551852875",
        "AssignIpv6AddressOnCreation": false,
        "Ipv6CidrBlockAssociationSet": [],
        "SubnetArn": "arn:aws:ec2:us-east-2:300551852875:subnet/subnet-0c2c097e460dac003"
    }
}
```

- Create an internet gateway using the following create-internet-gateway command.

# aws ec2 create-internet-gateway --query InternetGateway.InternetGatewayId --output text



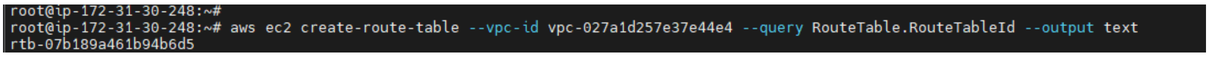
```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws ec2 create-internet-gateway --query InternetGateway.InternetGatewayId --output text
igw-01d18281692eed98d
```

- Using the ID from the previous step, attach the internet gateway to your VPC using the following attach-internet-gateway command.

# aws ec2 attach-internet-gateway --vpc-id <vpc - Id>--internet-gateway-id <Igw - Id>

- Create a custom route table for your VPC using the following create-route-table command.

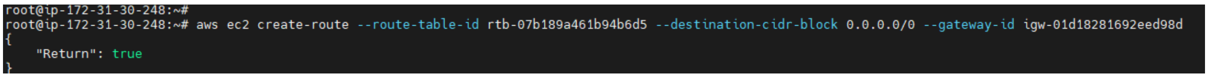
# aws ec2 create-route-table --vpc-id <vpc - Id>--query RouteTable.RouteTableId --output text



```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws ec2 create-route-table --vpc-id vpc-027a1d257e37e44e4 --query RouteTable.RouteTableId --output text
rtb-07b189a461b94b6d5
```

- Create a route in the route table that points all traffic (0.0.0.0/0) to the internet gateway using the following create-route command.

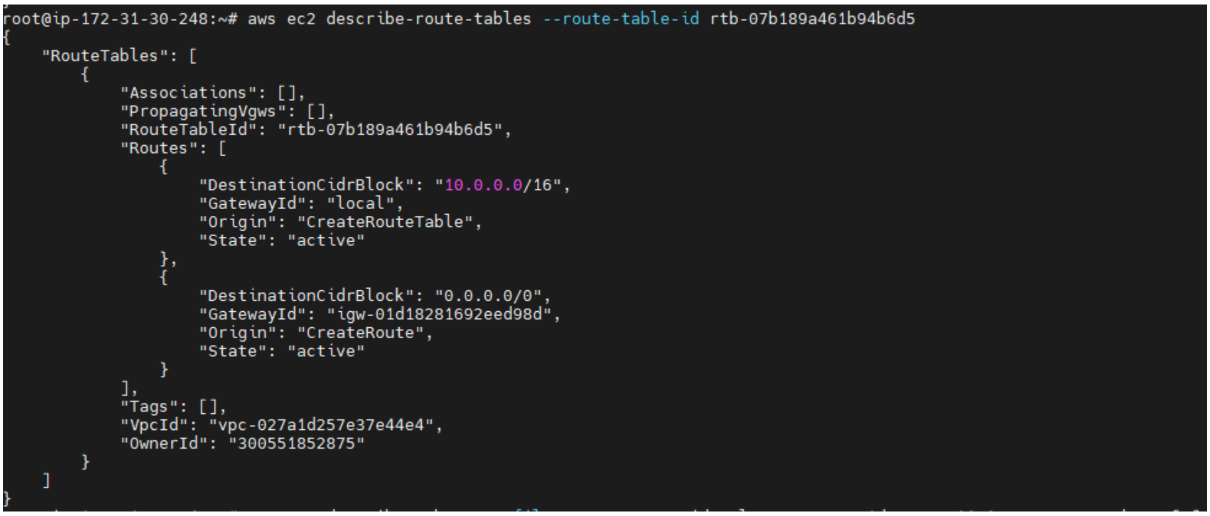
# aws ec2 create-route --route-table-id <route table - Id>--destination-cidr-block 0.0.0.0/0 --gateway-id <Igw - Id>



```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws ec2 create-route --route-table-id rtb-07b189a461b94b6d5 --destination-cidr-block 0.0.0.0/0 --gateway-id igw-01d18281692eed98d
{
    "Return": true
}
```

- You can describe the route table using the following describe-route-tables command.

# aws ec2 describe-route-tables --route-table-id <route table - Id>

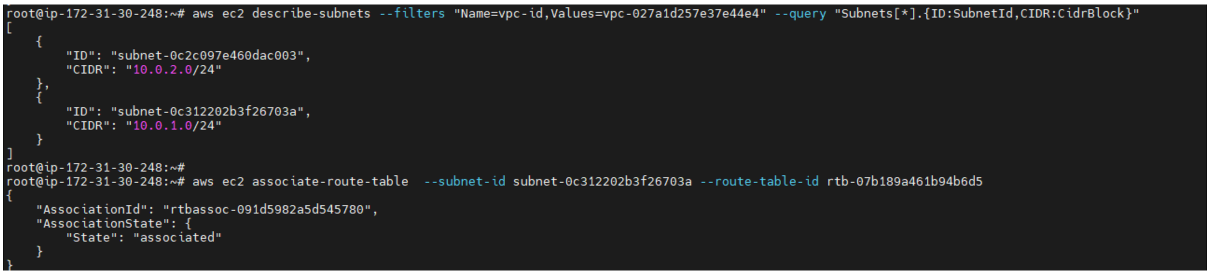


```
root@ip-172-31-30-248:~# aws ec2 describe-route-tables --route-table-id rtb-07b189a461b94b6d5
{
    "RouteTables": [
        {
            "Associations": [],
            "PropagatingVgws": [],
            "RouteTableId": "rtb-07b189a461b94b6d5",
            "Routes": [
                {
                    "DestinationCidrBlock": "10.0.0.0/16",
                    "GatewayId": "local",
                    "Origin": "CreateRouteTable",
                    "State": "active"
                },
                {
                    "DestinationCidrBlock": "0.0.0.0/0",
                    "GatewayId": "igw-01d18281692eed98d",
                    "Origin": "CreateRoute",
                    "State": "active"
                }
            ],
            "Tags": [],
            "VpcId": "vpc-027a1d257e37e44e4",
            "OwnerId": "300551852875"
        }
    ]
}
```

- The route table is currently not associated with any subnet. You need to associate it with a subnet in your VPC so that traffic from that subnet is routed to the internet gateway.

# aws ec2 describe-subnets --filters "Name=vpc-id,Values=<vpc –Id>

" --query "Subnets[*].{ID:SubnetId,CIDR:CidrBlock}"



```
root@ip-172-31-30-248:~# aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-027a1d257e37e44e4" --query "Subnets[*].{ID:SubnetId,CIDR:CidrBlock}"
[
    {
        "ID": "subnet-0c2c097e460dac003",
        "CIDR": "10.0.2.0/24"
    },
    {
        "ID": "subnet-0c312202b3f26703a",
        "CIDR": "10.0.1.0/24"
    }
]
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws ec2 associate-route-table  --subnet-id subnet-0c312202b3f26703a --route-table-id rtb-07b189a461b94b6d5
{
    "AssociationId": "rtbassoc-091d5982a5d545780",
    "AssociationState": {
        "State": "associated"
    }
}
```

- You can choose which subnet to associate with the custom route table, for example, *subnet-0c312202b3f26703a*, and associate it using the associate-route-table command. This subnet is your public subnet.

# aws ec2 associate-route-table --subnet-id <subnet-Id> --route-table-id <route table - Id>

CLEAN UP

Delete your subnets:

```
# aws ec2 delete-subnet --subnet-id <subnet-Id>
```

Delete your custom route table:

```
# aws ec2 delete-route-table --route-table-id <route table - Id>
```

Detach your internet gateway from your VPC:

```
# aws ec2 detach-internet-gateway --internet-gateway-id <Igw -Id> --vpc-id <vpc- Id>
```

Delete your internet gateway:

```
# aws ec2 delete-internet-gateway --internet-gateway-id <Igw - Id>
```

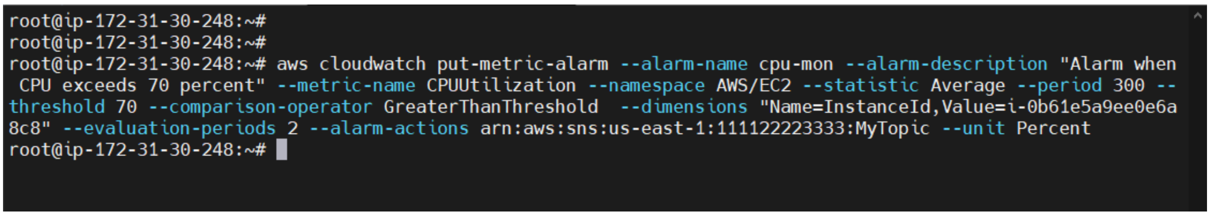
Delete your VPC:

```
# aws ec2 delete-vpc --vpc-id <vpc- Id>
```

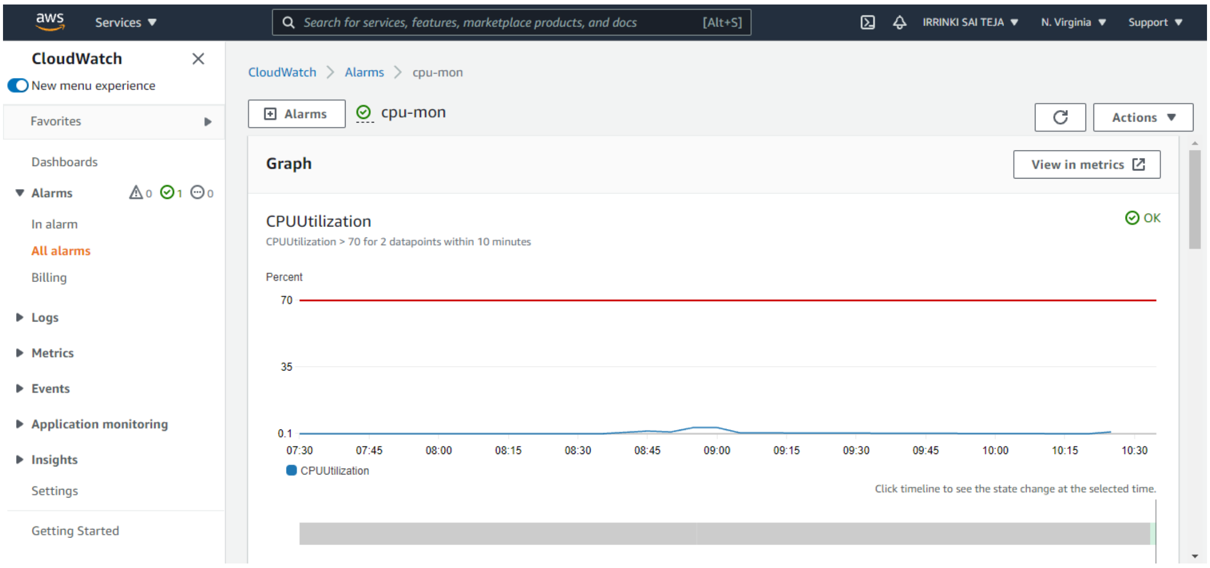
Cloud Watch

Creating Alarm

# aws cloudwatch put-metric-alarm --alarm-name <Alarm name> --alarm-description "<Description>" --metric-name <Metric> --namespace AWS/EC2 --statistic Average --period 300 --threshold <70> --comparison-operator <GreaterThanThreshold> --dimensions "Name=InstanceId,Value=<Id>" --evaluation-periods 2 --alarm-actions <SNS – arn > --unit Percent



```
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~#
root@ip-172-31-30-248:~# aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm when
 CPU exceeds 70 percent" --metric-name CPUUtilization --namespace AWS/EC2 --statistic Average --period 300 --
threshold 70 --comparison-operator GreaterThanThreshold  --dimensions "Name=InstanceId,Value=i-0b61e5a9ee0e6a
8c8" --evaluation-periods 2 --alarm-actions arn:aws:sns:us-east-1:111122223333:MyTopic --unit Percent
root@ip-172-31-30-248:~#
```



CPUUtilization > 70 for 2 datapoints within 10 minutes

Click timeline to see the state change at the selected time.

Delete Your Alarm

# aws cloudwatch delete-alarms --alarm-names <Alarm name>

Disable your Alarm

# aws cloudwatch disable-alarm-actions --alarm-names <Alarm name>

Enable your Alarm

# aws cloudwatch enable-alarm-actions --alarm-names <Alarm name