

Steps to Follow

1. Created a free Tier account on AWS
 - a) Created a IAM user of my name
 - b) Give Necessary permission to the users

The screenshot displays the AWS IAM console interface. On the left, a sidebar shows navigation options like 'Access', 'ent', 'ent', and 'ement New'. The main content area is divided into two sections. The top section, titled 'Users (1)', shows a table with one user, 'saurabh-shah', with a last activity of '9 minutes ago' and a password age of '5 hours'. The bottom section, titled 'Permissions policies (4)', shows a table with four policies: 'AmazonEC2FullAccess', 'AmazonS3FullAccess', 'AmazonS3ReadOnlyAccess', and 'IAMUserChangePassword', all of which are 'AWS managed' and 'Attached via Directly'.

User name	Path	Group	Last activity	MFA	Password age
saurabh-shah	/	0	9 minutes ago	-	5 hours

Policy name	Type	Attached via
AmazonEC2FullAccess	AWS managed	Directly
AmazonS3FullAccess	AWS managed	Directly
AmazonS3ReadOnlyAccess	AWS managed	Directly
IAMUserChangePassword	AWS managed	Directly

2. Install the aws cli on the local
Configure the aws account

```
saurabh@saurabh-HP-Laptop-15-da0xxx:~$ aws --version
aws-cli/1.36.9 Python/3.10.12 Linux/6.8.0-49-generic botocore/1.35.68
```

3. Install terraform On local

```
saurabh@saurabh-HP-Laptop-15-da0xxx:~$ terraform --version
Terraform v1.9.8-dev
on linux_amd64
+ provider registry.terraform.io/hashicorp/aws v5.77.0

Your version of Terraform is out of date! The latest version
is 1.9.8. You can update by downloading from https://www.terraform.io/downloads.html
saurabh@saurabh-HP-Laptop-15-da0xxx:~$
```

1. I had Created the Infra via the Terraform , so it main.tf consists of
 - a) Creation of s3 bucket with name one2n-s3
 - b) Creation of AWS instance in the region

After Creation of ec2 it will create run commands on the ec2 server

```
provider "aws" {
  region = "us-east-1" # Choose your region
}

resource "aws_s3_bucket" "bucket" {
  bucket = "one2n-s3"
  acl    = "private"
}

resource "aws_security_group" "web_sg" {
  name_prefix = "web_sg"
  egress {
    cidr_blocks = ["0.0.0.0/0"]
    from_port   = 0
    to_port     = 0
    protocol    = "tcp"
  }

  ingress {
    cidr_blocks = ["0.0.0.0/0"]
    from_port   = 80
    to_port     = 80
    protocol    = "tcp"
  }
}

resource "aws_instance" "web_instance" {
  ami           = "ami-0453ec754f44f9a4a" # Amazon Linux 2 AMI ID (change based on region)
  instance_type = "t2.micro"
  security_groups = [aws_security_group.web_sg.name]
  user_data = <<-EOF
    #!/bin/bash
    yum update -y
    yum install -y python3
    pip3 install Flask boto3
  >>>EOF
}
```

```
user_data = <<-EOF
  #!/bin/bash
  yum update -y
  yum install -y python3
  pip3 install Flask boto3
  cd /home/ec2-user
  #git clone https://github.com/yourusername/your-repository.git
  git clone https://github.com/shahi-saurabh/s3-list-service.git
  cd s3-list-service
  python3 app.py
EOF

tags = {
  Name = "FlaskAppInstance"
}

output "web_instance_ip" {
  value = aws_instance.web_instance.public_ip
}
```

537: No write since last change (add ! to override)

Main .tf

Once the EC2 and S3 is created you can look onto the aws portal.

Instances (1) Info Last updated less than a minute ago Connect Instance state Actions Launch instances

All states

Instance state = running Clear filters < 1 > Settings

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	FlaskAppInsta...	I-0011d57954ed6aeea	Running	t2.micro	2/2 checks passed	View alarms

I have configure the ec2 security group with the following ports

inBound rule

Inbound rules (3) Manage tags Edit inbound rules

< 1 > Settings

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol
<input type="checkbox"/>	-	sgr-0462ff637f3e057b4	IPv4	SSH	TCP
<input type="checkbox"/>	-	sgr-0da13a554685a2338	IPv4	Custom TCP	TCP
<input type="checkbox"/>	-	sgr-094fc4c5ae2f6c78a	IPv4	HTTP	TCP

Outbound rule

Outbound rules (2) Manage tags Edit outbound rules

< 1 > Settings

<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol
<input type="checkbox"/>	-	sgr-054bcd9b343ca541	IPv4	Custom TCP	TCP
<input type="checkbox"/>	-	sgr-0517e484954f9dd53	IPv4	All traffic	All

one2n-s3Info

Objects

Properties

Permissions

Metrics

Management

Access Points

Objects (1)Info

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

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](#)

Find objects by prefix

< 1 >

Name

Type

Last modified

Size

Storage class

dir1/

Folder

-

-

-

I had copied the file from my local to s3 by the following command

```
file1.txt
saurabh@saurabh-HP-Laptop-15-da0xxx:~$ aws s3 cp file1.txt s3://one2n-s3/dir1/file1.txt
upload: ./file1.txt to s3://one2n-s3/dir1/file1.txt
```

one2n-s3Info

Objects

Properties

Permissions

Metrics

Management

Access Points

Objects (1)Info

Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

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](#)

Find objects by prefix

< 1 >

Name

Type

Last modified

Size

Storage class

dir1/

Folder

-

-

-

OUTPUT SHOWING THE DIR OF S3

```
[ec2-user@ip-172-31-84-239 s3-list-service]$ curl http://172.31.84.239:5000/list-bucket-content
{
  "content": [
    "dir1"
  ]
}
[ec2-user@ip-172-31-84-239 s3-list-service]$
```