Case Study 03

Infrastructure as Code with Terraform

Name: Harsh Vijay Jadhav
Class: D15B
Roll no.: 23

- **Concepts Used**: Terraform, AWS S3, and EC2.
- **Problem Statement**: "Use Terraform to provision an AWS EC2 instance and an S3 bucket. Deploy a sample static website on the S3 bucket using the EC2 instance as the backend server."
- **Tasks**:
    - Write a Terraform script to create an EC2 instance and an S3 bucket.
    - Deploy the static website on the S3 bucket.
    - Use the EC2 instance to interact with the S3 bucket and log the actions.

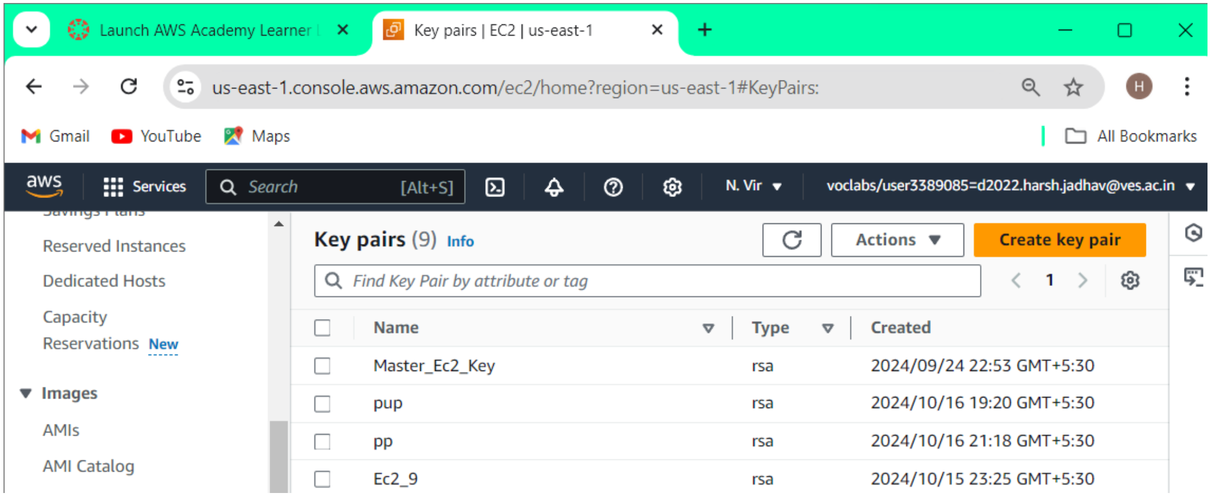
SOLUTION

Overview

This report covers the use of Terraform to provision an AWS EC2 instance and S3 bucket, deploy a static website on the S3 bucket, and use the EC2 instance to interact with the S3 bucket and log actions. The tasks were completed step by step, with screenshots provided after each step to demonstrate successful execution.

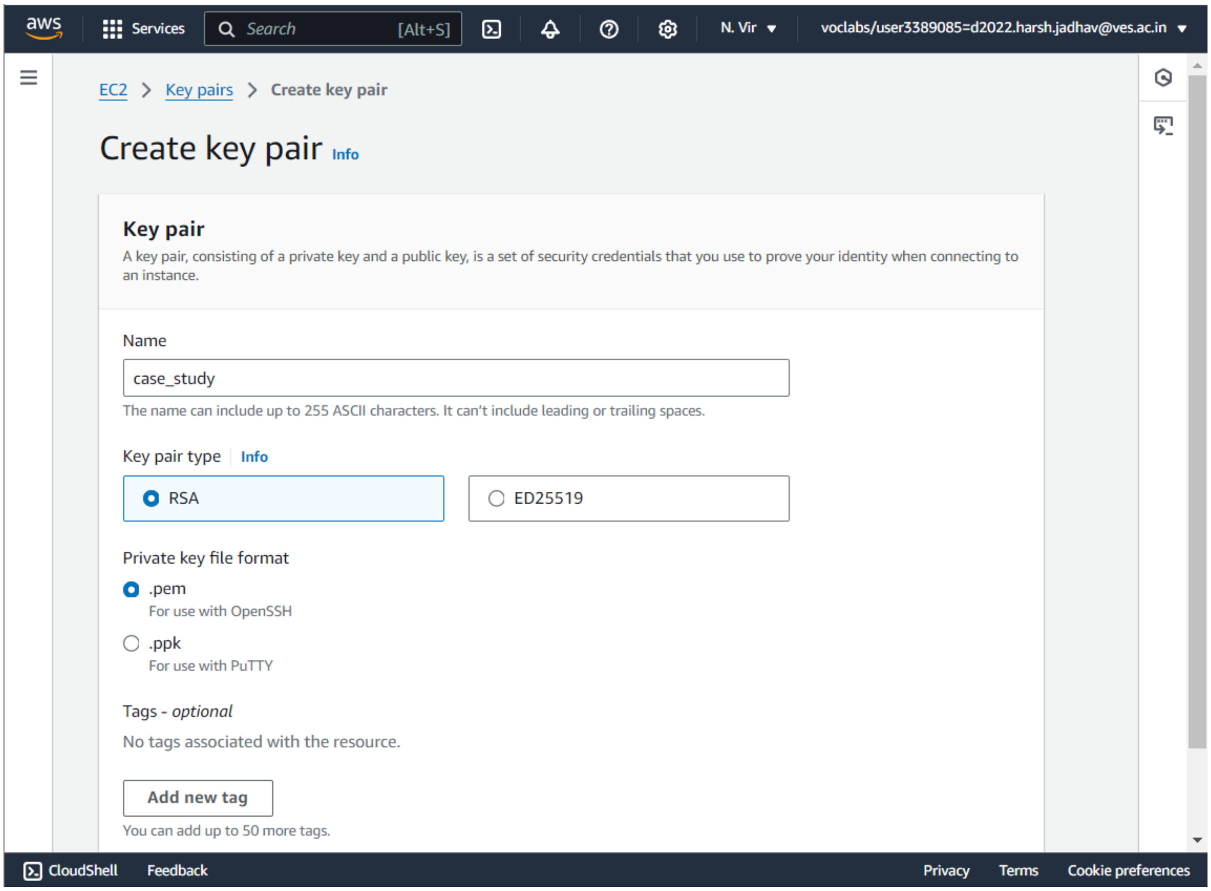
---

1. Go to the AWS Management Console and create a key pair in the EC2 section:
- Open the EC2 Dashboard.
- Click on "Key Pairs" under Network & Security.
- Click on "Create Key Pair".


Launch AWS Academy Learner
Key pairs | EC2 | us-east-1
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#KeyPairs:
Gmail
YouTube
Maps
All Bookmarks
aws
Services
Search
[Alt+S]
N. Vir
voclabs/user3389085=d2022.harsh.jadhav@ves.ac.in
Reserved Instances
Dedicated Hosts
Capacity Reservations New
Images
AMIs
AMI Catalog
Key pairs (9) Info
Actions
Create key pair
Find Key Pair by attribute or tag
1

Name	Type	Created
Master_Ec2_Key	rsa	2024/09/24 22:53 GMT+5:30
pup	rsa	2024/10/16 19:20 GMT+5:30
pp	rsa	2024/10/16 21:18 GMT+5:30
Ec2_9	rsa	2024/10/15 23:25 GMT+5:30

Name the key pair, e.g., harsh-case-study, and click Create Key Pair.



EC2 > Key pairs > Create key pair

Create key pair Info

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

case_study

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type Info

● RSA ○ ED25519

Private key file format

● .pem
For use with OpenSSH

○ .ppk
For use with PuTTY

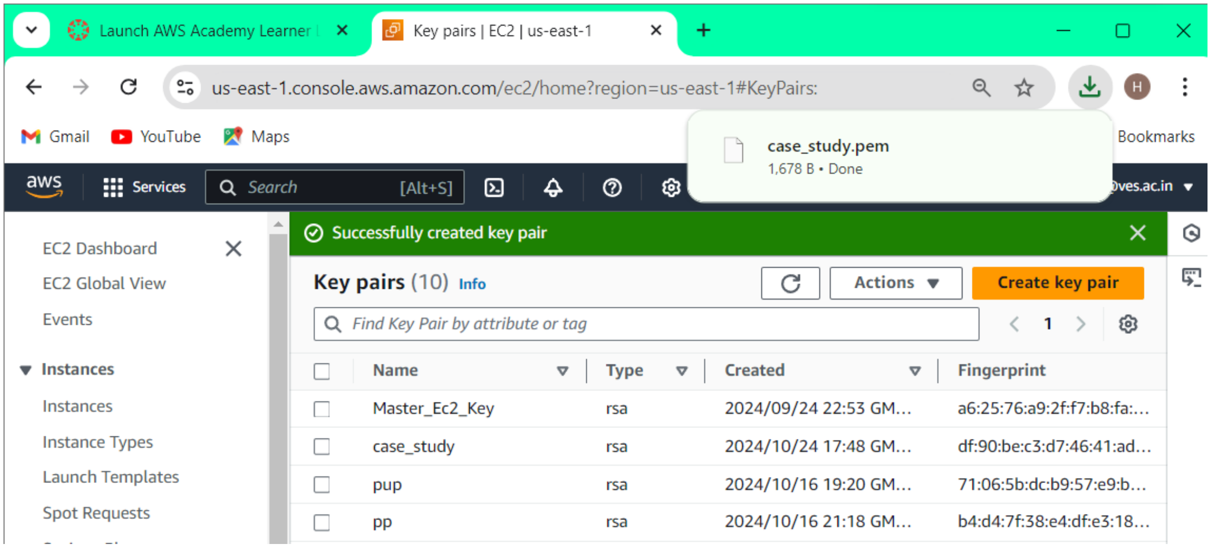
Tags - *optional*

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

The .pem file will automatically download. Ensure it's saved securely, as you'll need it later to connect to your EC2 instance.



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#KeyPairs:

case_study.pem
1,678 B • Done

Successfully created key pair

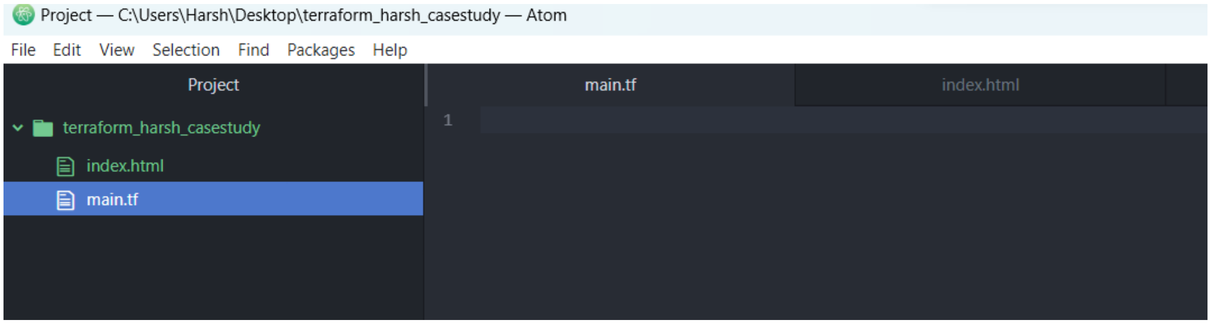
Key pairs (10) Info

Find Key Pair by attribute or tag

Name	Type	Created	Fingerprint
Master_Ec2_Key	rsa	2024/09/24 22:53 GM...	a6:25:76:a9:2f:f7:b8:fa:...
case_study	rsa	2024/10/24 17:48 GM...	df:90:be:c3:d7:46:41:ad...
pup	rsa	2024/10/16 19:20 GM...	71:06:5b:dc:b9:57:e9:b...
pp	rsa	2024/10/16 21:18 GM...	b4:d4:7f:38:e4:df:e3:18...

2. Open VS Code and generate two files (main.tf and index.html):

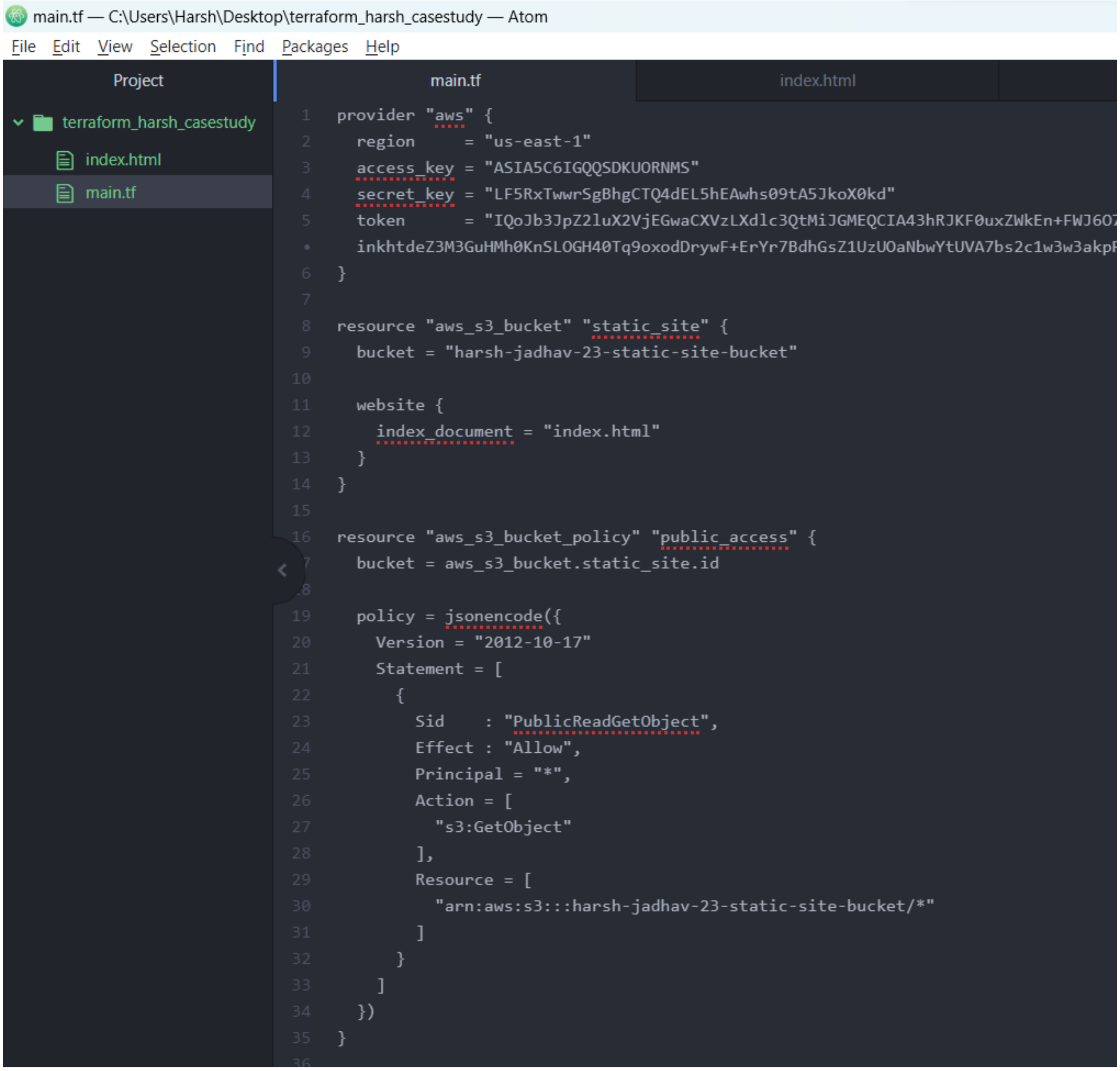
- Create a new folder in your local workspace.
- Inside that folder, create two files: main.tf (Terraform configuration) and index.html (the static website content).


Project — C:\Users\Harsh\Desktop\terraform_harsh_casestudy — Atom
File
Edit
View
Selection
Find
Packages
Help
Project
terraform_harsh_casestudy
index.html
main.tf
main.tf
index.html
1

- Paste the following content into main.tf:

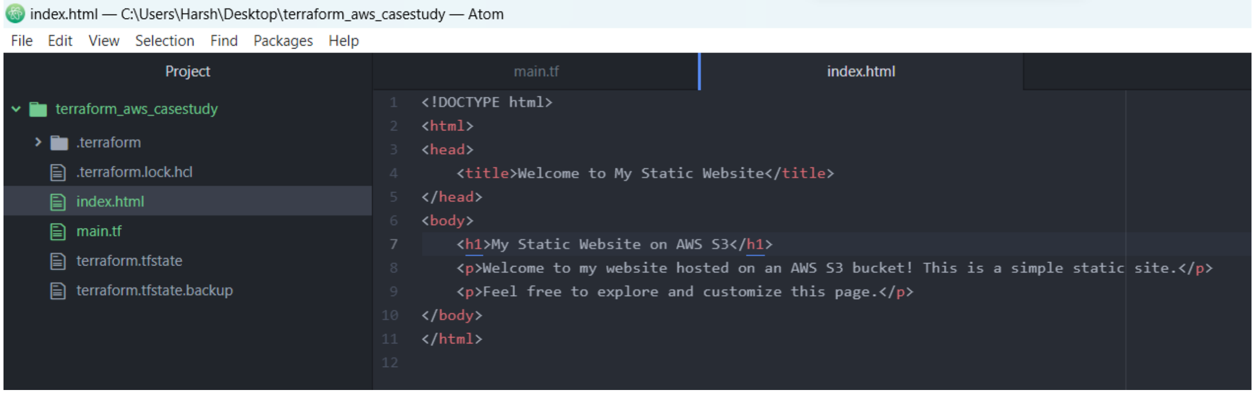
```
provider "aws" {
  region     = "us-east-1"
  access_key = "ASIA5C6IGQQSDKUORNMS"
  secret_key = "LF5RxTwwrSgBhgCTQ4dEL5hEAwhs09tA5JkoX0kd"
  token                                                                      =
"IQoJb3JpZ2luX2VjEGwaCXVzLXdlc3QtMiJGMEQCIA43hRJKF0uxZWkEn+FWJ6O7LV+4LtVVRq
y9s9XfP5SIAiARO/HE6VTNhqGcaPgTc8rZAYacdKPyQ06mbOeShR/ssyq/AgjV//////////8BEAEaDDg5
OTY3ODYzNTA0NCIMXM1C8vJBnKETXQm4KpMCQNPAY4PE7wCylwW+Ek3z1i1AfPvteWfdH1
G5gcVkhLc0LInFTu2JZkcaJNAJnzALfCxtectHyzSbvg+SK96QLfIP68zjVQ7oXuHCl+xIYlJSY4eWVj9
dsU+1TPkXcpotAYoZ8IBXtBs8/ty/ZgX3FPD8vIGNBm2id0DnWKPRzazV5hXYovM231Y47KyPzlsP
dX3dt880jmq5omA0zByn6WoF6F1t/inkhtdeZ3M3GuHMh0KnSLOGH40Tq9oxodDrywF+ErYr7BdhGs
Z1UzUOaNbwYtUVA7bs2c1w3w3akpPpLrC5Se89SQgPrxgIHWEDUGrlfjBjNyrIDTxq+N7LpiGFbdU
67ipFjkx0cPY14UCeQcEwyPHouAY6ngE9EW1nYp1zKMlpZ62HBxh2DE7OTH4uGonSxRizaV2dKW
+Vk5Xcd3wdaPSu8ifxMCo2uhDPizTv8lqH/WVsnZaQ1YixnELzxbiEZqolyZdPjLt/H/0sExXLKDmXh/
PZW210negbaW/U8WZYcrUrG/EaI4J3Yck9gZjIqQe/JVi/buJQVA8RemhbUkU4ie82baPh735QuIXAX
OpuhSIFrw=="
}

resource "aws_s3_bucket" "static_site" {
  bucket = "harsh-jadhav-23"

  website {
    index_document = "index.html"
  }
}

resource "aws_s3_bucket_policy" "public_access" {
  bucket = aws_s3_bucket.static_site.id

  policy = jsonencode({
    Version = "2012-10-17"
    Statement = [
      {
        Sid     : "PublicReadGetObject",
        Effect : "Allow",
        Principal = "*",
        Action = [
          "s3:GetObject"
        ],
        Resource = [
          "arn:aws:s3:::harsh-jadhav-23/*"
        ]
      }
    ]
  })
}

resource "aws_s3_bucket_object" "index" {
  bucket       = aws_s3_bucket.static_site.bucket
  key          = "index.html"
  source       = "C:/Users/Harsh/Desktop/terraform_aws_casestudy/index.html"
  content_type = "text/html"
}

resource "aws_s3_bucket_public_access_block" "example" {
  bucket                  = aws_s3_bucket.static_site.id

  block_public_acls       = false
  ignore_public_acls      = false
  block_public_policy     = false
  restrict_public_buckets = false
}

resource "aws_instance" "web_server" {
  ami                         = "ami-06b21ccaeff8cd686"
  instance_type               = "t2.micro"
  associate_public_ip_address = true
  key_name                    = "harsh_case_study_aws"
  vpc_security_group_ids      = [aws_security_group.allow_ssh.id]

  tags = {
    Name = "Harsh-Jadhav-23-Ec2"
  }

  user_data = <<-EOF
  #!/bin/bash
  echo "Starting setup..." > /var/log/s3_access.log

  yum install -y aws-cli >> /var/log/s3_access.log 2>&1
  aws configure set aws_access_key_id "ASIA5C6IGQQSDKUORNMS" >> /var/log/s3_access.log 2>&1
  aws configure set aws_secret_access_key "LF5RxTwwrSgBhgCTQ4dEL5hEAwhs09tA5JkoX0kd" >>
/var/log/s3_access.log 2>&1
  aws configure set aws_session_token
"IQoJb3JpZ2luX2VjEGwaCXVzLXdlc3QtMiJGMEQCIA43hRJKF0uxZWkEn+FWJ6O7LV+4LtVVRq
y9s9XfP5SIAiARO/HE6VTNhqGcaPgTc8rZAYacdKPyQ06mbOeShR/ssyq/AgjV//////////8BEAEaDDg5
OTY3ODYzNTA0NCIMXM1C8vJBnKETXQm4KpMCQNPAY4PE7wCylwW+Ek3z1i1AfPvteWfdH1
G5gcVkhLc0LInFTu2JZkcaJNAJnzALfCxtectHyzSbvg+SK96QLfIP68zjVQ7oXuHCl+xIYlJSY4eWVj9
dsU+1TPkXcpotAYoZ8IBXtBs8/ty/ZgX3FPD8vIGNBm2id0DnWKPRzazV5hXYovM231Y47KyPzlsP
```

```
dX3dt880jmq5omA0zByn6WoF6F1t/inkhtdeZ3M3GuHMh0KnSLOGH40Tq9oxodDrywF+ErYr7BdhGs
Z1UzUOaNbwYtUVA7bs2c1w3w3akpPpLrC5Se89SQgPrxgIHWEDUGrlfjBjNyrIDTxq+N7LpiGFbdU
67ipFjkx0cPY14UCeQcEwyPHouAY6ngE9EW1nYp1zKMlpZ62HBxh2DE7OTH4uGonSxRizaV2dKW
+Vk5Xcd3wdaPSu8ifxMCo2uhDPizTv8lqH/WVsnZaQ1YixnELzxbiEZqolyZdPjLt/H/0sExXLKDmXh/
PZW210negbaW/U8WZYcrUrG/EaI4J3Yck9gZjIqQe/JVi/buJQVA8RemhbUkU4ie82baPh735QuIXAX
OpuhSIFrw==" >> /var/log/s3_access.log 2>&1
  aws configure set default.region "us-east-1" >> /var/log/s3_access.log 2>&1

  aws s3 cp s3://harsh-jadhav-23/index.html /home/ec2-user/index.html >> /var/log/s3_access.log 2>&1
  EOF
}

resource "aws_security_group" "allow_ssh" {
  name        = "allow_ssh"
  description = "Allow SSH inbound traffic"

  ingress {
    from_port   = 22
    to_port     = 22
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

output "instance_ip" {
  value = aws_instance.web_server.public_ip
}

output "s3_bucket_url" {
  value = aws_s3_bucket.static_site.website_endpoint
}
```


main.tf — C:\Users\Harsh\Desktop\terraform_harsh_casestudy — Atom
File Edit View Selection Find Packages Help
Project
terraform_harsh_casestudy
index.html
main.tf
main.tf
index.html

```
provider "aws" {
  region     = "us-east-1"
  access_key = "ASIA5C6IGQQSDKUORNMS"
  secret_key = "LF5RxTwwrSgBhgCTQ4dEL5hEAwhs09tA5JkoX0kd"
  token      = "IQoJb3JpZ2luX2VjEGwaCXVzLXdlc3QtMiJGMEQCIA43hRJKF0uxZWkEn+FWJ6O
  inkhtdeZ3M3GuHMh0KnSLOGH40Tq9oxodDrywF+ErYr7BdhGsZ1UzUOaNbwYtUVA7bs2c1w3w3akp
}

resource "aws_s3_bucket" "static_site" {
  bucket = "harsh-jadhav-23-static-site-bucket"

  website {
    index_document = "index.html"
  }
}

resource "aws_s3_bucket_policy" "public_access" {
  bucket = aws_s3_bucket.static_site.id

  policy = jsonencode({
    Version = "2012-10-17"
    Statement = [
      {
        Sid    : "PublicReadGetObject",
        Effect : "Allow",
        Principal = "*",
        Action = [
          "s3:GetObject"
        ],
        Resource = [
          "arn:aws:s3:::harsh-jadhav-23-static-site-bucket/*"
        ]
      }
    ]
  })
}
```

Create index.html with content:

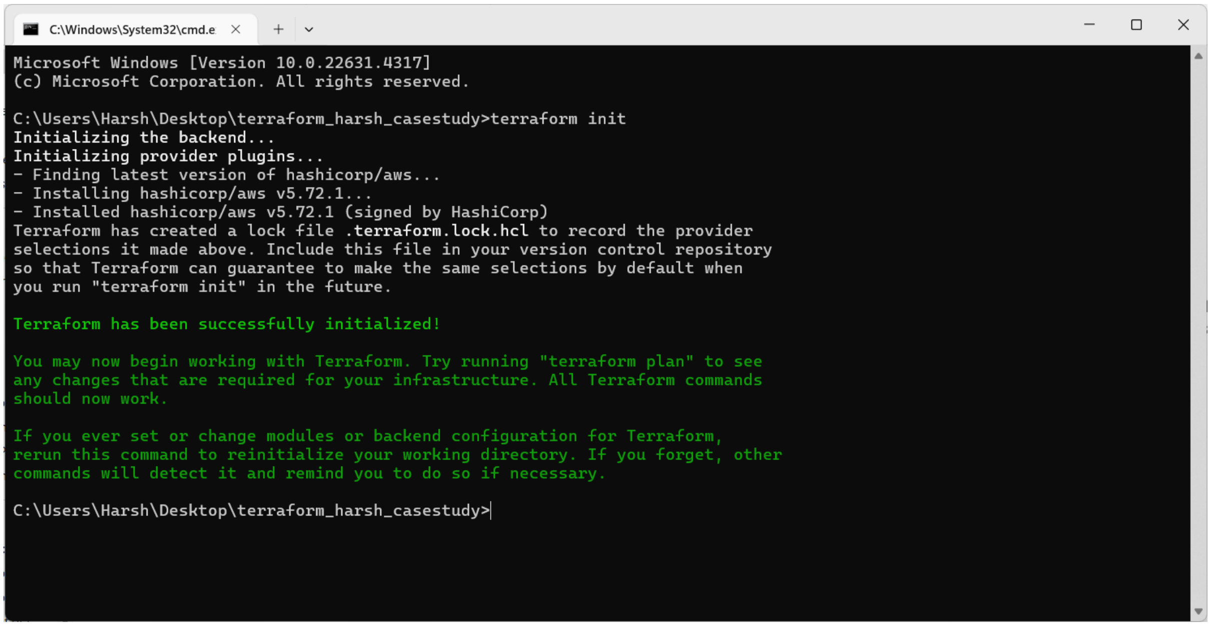


```
<!DOCTYPE html>
<html>
<head>
    <title>Welcome to My Static Website</title>
</head>
<body>
    <h1>My Static Website on AWS S3</h1>
    <p>Welcome to my website hosted on an AWS S3 bucket! This is a simple static site.</p>
    <p>Feel free to explore and customize this page.</p>
</body>
</html>
```

3. Initialize Terraform:

- Open the terminal in VS Code, navigate to the folder containing main.tf, and run:

terraform init



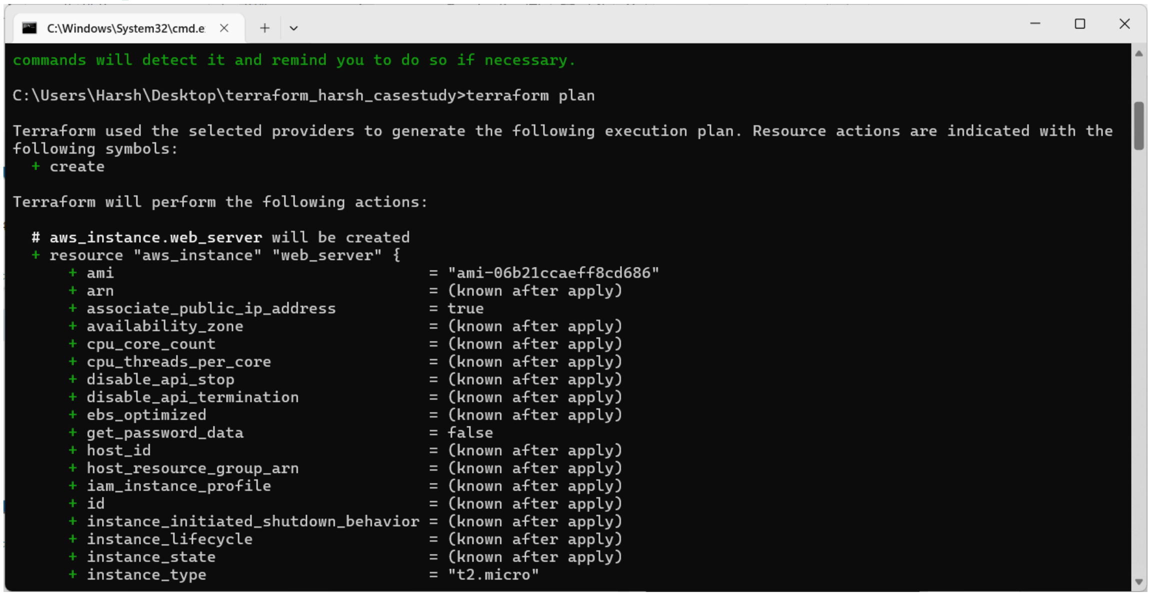
```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Harsh\Desktop\terraform_harsh_casestudy>terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.72.1...
- Installed hashicorp/aws v5.72.1 (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,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

C:\Users\Harsh\Desktop\terraform_harsh_casestudy>
```

- This will initialize Terraform and download necessary provider plugins.

4. Plan and apply Terraform configuration:

- Run the following commands to plan and deploy resources:

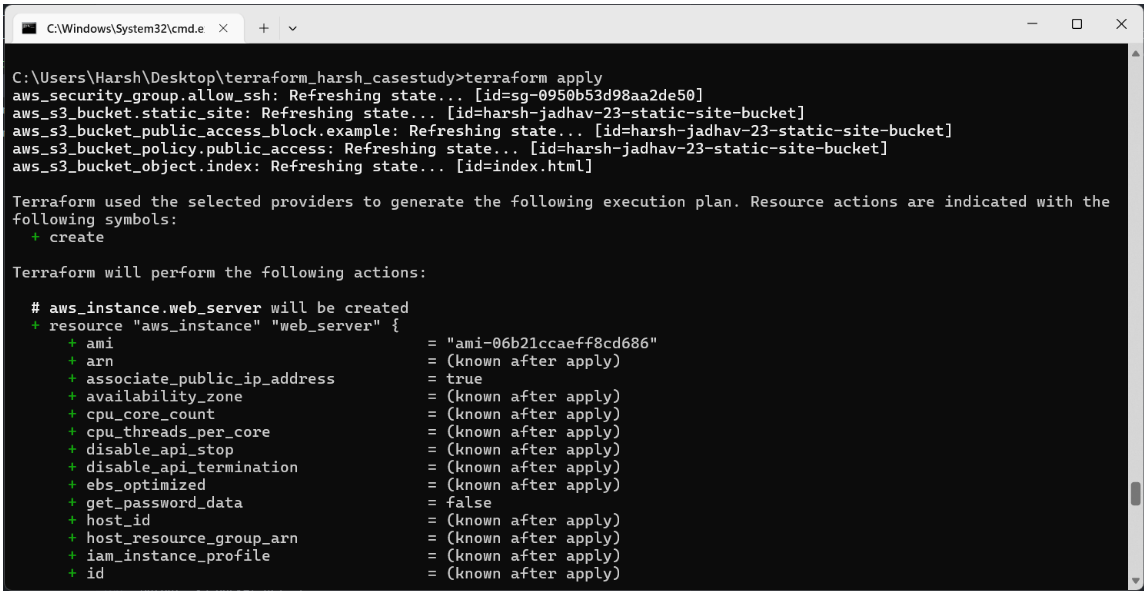
terraform plan



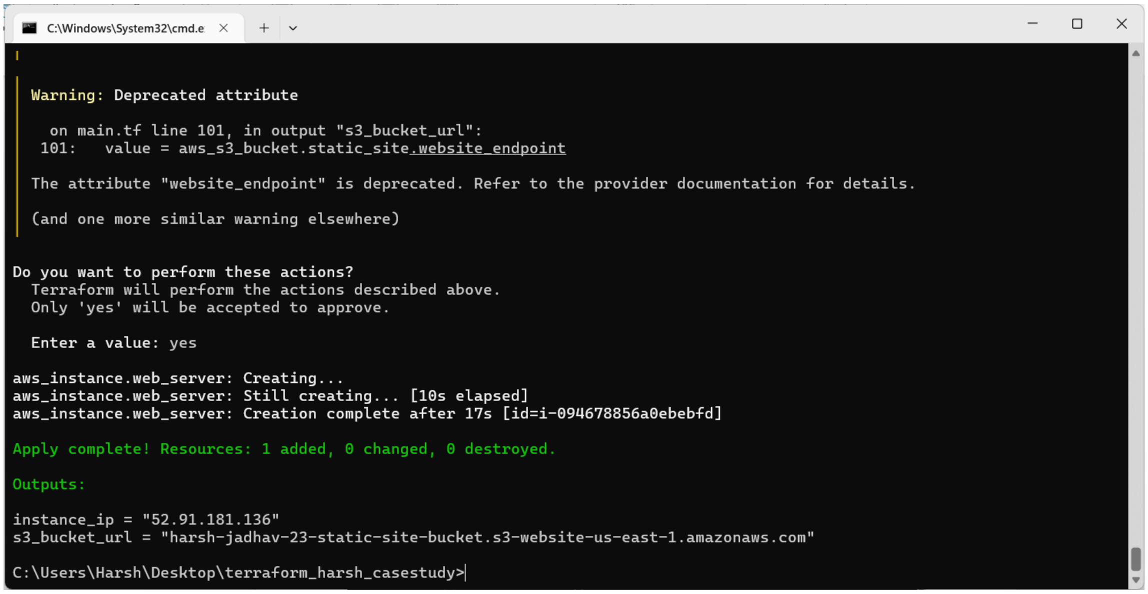
```
commands will detect it and remind you to do so if necessary.

C:\Users\Harsh\Desktop\terraform_harsh_casestudy>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_instance.web_server will be created
  + resource "aws_instance" "web_server" {
      + ami                                  = "ami-06b21ccaeff8cd686"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
```

terraform apply

- Type yes when prompted to proceed.



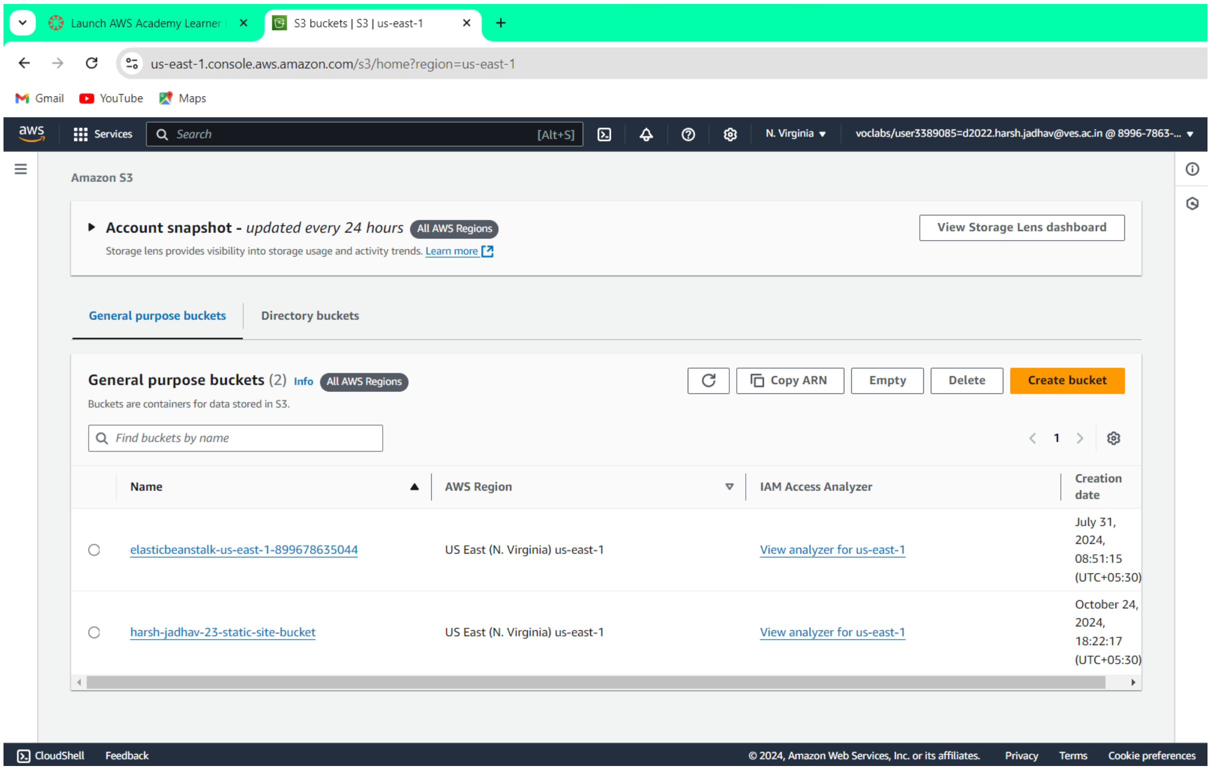
```
C:\Users\Harsh\Desktop\terraform_harsh_casestudy>terraform apply
aws_security_group.allow_ssh: Refreshing state... [id=sg-0950b53d98aa2de50]
aws_s3_bucket.static_site: Refreshing state... [id=harsh-jadhav-23-static-site-bucket]
aws_s3_bucket_public_access_block.example: Refreshing state... [id=harsh-jadhav-23-static-site-bucket]
aws_s3_bucket_policy.public_access: Refreshing state... [id=harsh-jadhav-23-static-site-bucket]
aws_s3_bucket_object.index: Refreshing state... [id=index.html]

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_instance.web_server will be created
  + resource "aws_instance" "web_server" {
      + ami                                  = "ami-06b21ccaeff8cd686"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
```



```
│
│ Warning: Deprecated attribute
│
│   on main.tf line 101, in output "s3_bucket_url":
│  101:   value = aws_s3_bucket.static_site.website_endpoint
│
│ The attribute "website_endpoint" is deprecated. Refer to the provider documentation for details.
│
│ (and one more similar warning elsewhere)
│

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.web_server: Creating...
aws_instance.web_server: Still creating... [10s elapsed]
aws_instance.web_server: Creation complete after 17s [id=i-094678856a0ebebfd]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

instance_ip = "52.91.181.136"
s3_bucket_url = "harsh-jadhav-23-static-site-bucket.s3-website-us-east-1.amazonaws.com"

C:\Users\Harsh\Desktop\terraform_harsh_casestudy>
```

5. Verify S3 bucket creation:

- Once the resources are created, go to the S3 console and verify that the bucket harsh-padyal-43-static-site-bucket was created.



Launch AWS Academy Learner | S3 buckets | S3 | us-east-1

us-east-1.console.aws.amazon.com/s3/home?region=us-east-1

Gmail YouTube Maps

aws Services Search [Alt+S] N. Virginia ▾ voclabs/user3389085=d2022.harsh.jadhav@ves.ac.in @ 8996-7863-... ▾

Amazon S3

▸ **Account snapshot -** *updated every 24 hours* All AWS Regions

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

View Storage Lens dashboard

General purpose buckets | Directory buckets

General purpose buckets (2) Info All AWS Regions

Copy ARN | Empty | Delete | Create bucket

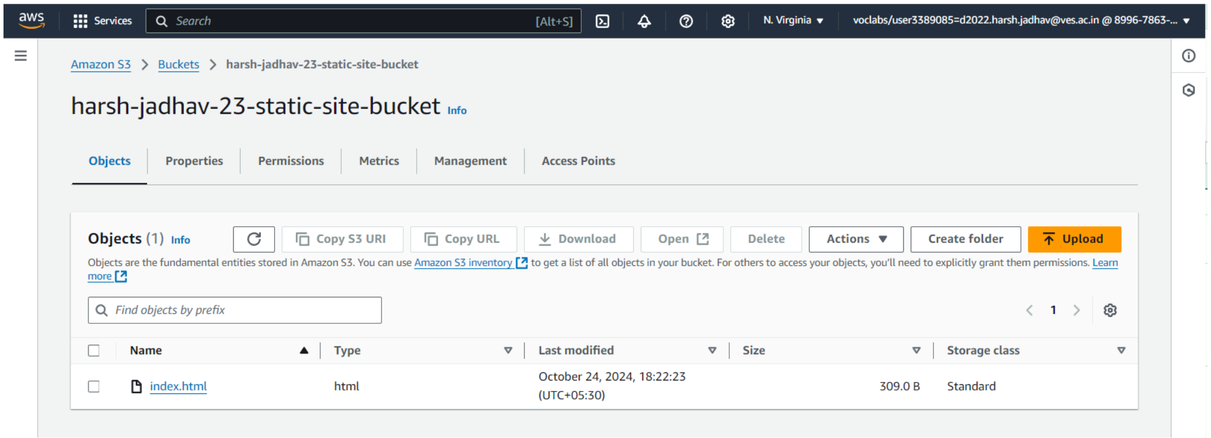
Buckets are containers for data stored in S3.

Find buckets by name

	Name ▲	AWS Region ▽	IAM Access Analyzer	Creation date
○	elasticbeanstalk-us-east-1-899678635044	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 31, 2024, 08:51:15 (UTC+05:30)
○	harsh-jadhav-23-static-site-bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	October 24, 2024, 18:22:17 (UTC+05:30)

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

- Click on the bucket, then on index.html.



aws Services Search [Alt+S] N. Virginia ▾ voclabs/user3389085=d2022.harsh.jadhav@ves.ac.in @ 8996-7863-... ▾

Amazon S3 > Buckets > harsh-jadhav-23-static-site-bucket

harsh-jadhav-23-static-site-bucket Info

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (1) Info

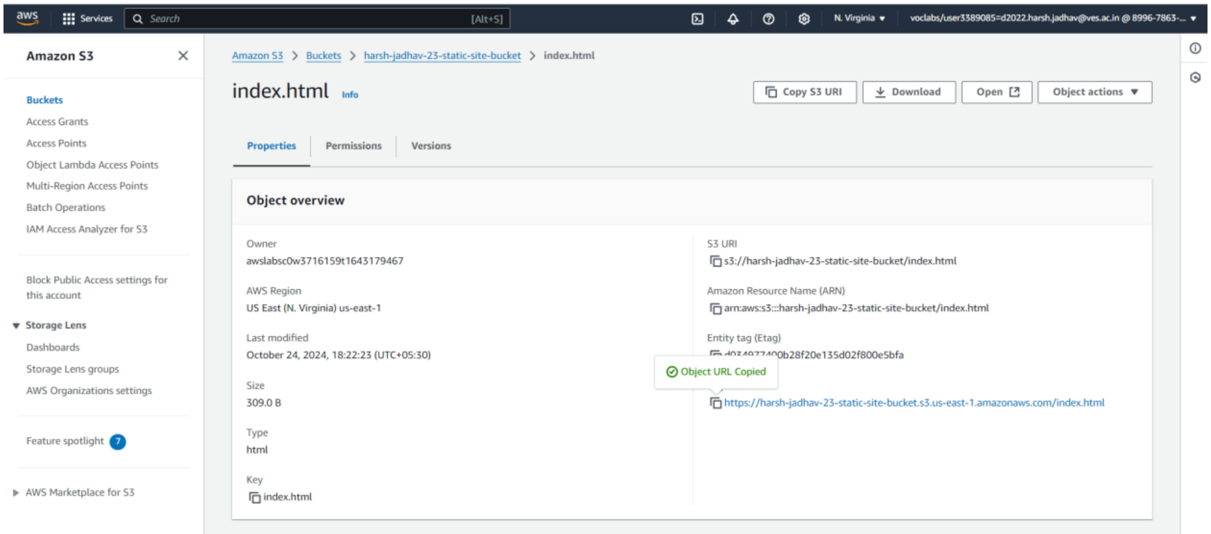
Copy S3 URI | Copy URL | Download | Open | Delete | Actions ▾ | Create folder | Upload

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

□	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class ▽
□	index.html	html	October 24, 2024, 18:22:23 (UTC+05:30)	309.0 B	Standard

- Click Open to view the content of the index.html file hosted on the S3 website.



aws
Services
Search [Alt+S]
N. Virginia
voclabs/user3389085=d2022.harsh.jadhav@ves.ac.in @ 8996-7863-...
Amazon S3
Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 7
AWS Marketplace for S3
Amazon S3 > Buckets > harsh-jadhav-23-static-site-bucket > index.html
Info
Copy S3 URI
Download
Open
Object actions
Properties
Permissions
Versions

index.html

Object overview

Owner
awslabsc0w3716159t1643179467

AWS Region
US East (N. Virginia) us-east-1

Last modified
October 24, 2024, 18:22:23 (UTC+05:30)

Size
309.0 B

Type
html

Key
index.html

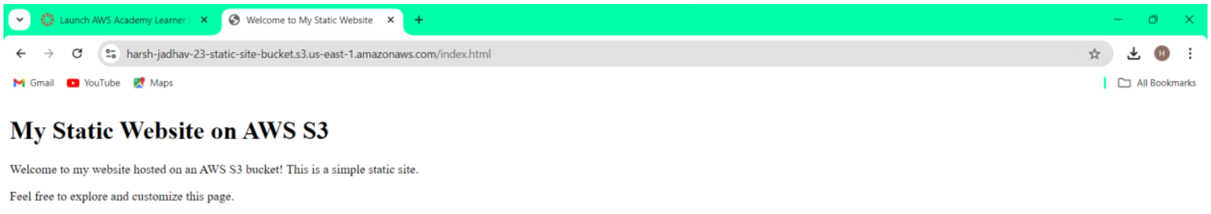
S3 URI
s3://harsh-jadhav-23-static-site-bucket/index.html

Amazon Resource Name (ARN)
arn:aws:s3:::harsh-jadhav-23-static-site-bucket/index.html

Entity tag (Etag)
d034977400b28f20e135d02f800e5bfa

Object URL Copied

https://harsh-jadhav-23-static-site-bucket.s3.us-east-1.amazonaws.com/index.html



Launch AWS Academy Learner
Welcome to My Static Website
harsh-jadhav-23-static-site-bucket.s3.us-east-1.amazonaws.com/index.html
Gmail
YouTube
Maps
All Bookmarks

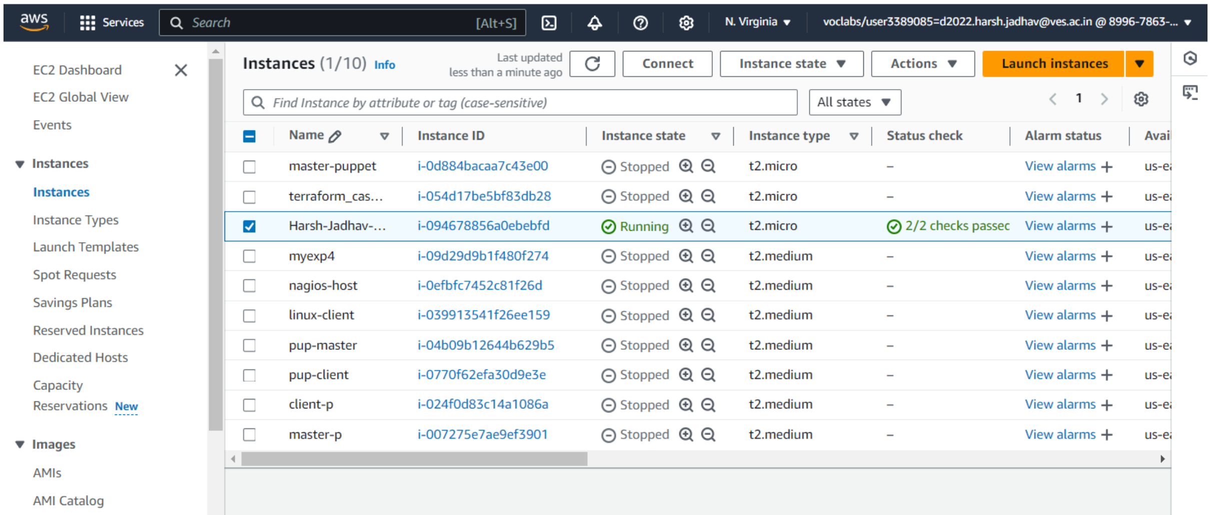
My Static Website on AWS S3

Welcome to my website hosted on an AWS S3 bucket! This is a simple static site.

Feel free to explore and customize this page.

6. Connect to the EC2 instance:

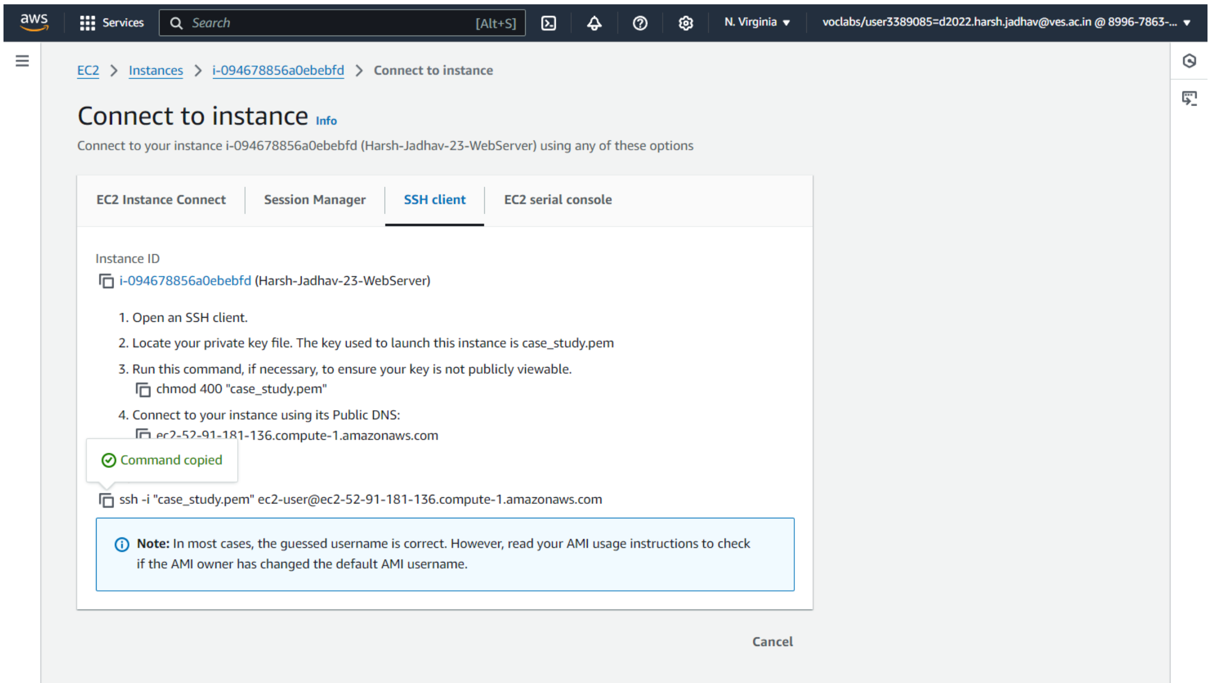
- In the EC2 console, locate your instance and click Connect.



Instances (1/10)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Avai
□	master-puppet	i-0d884bacaa7c43e00	Stopped	t2.micro	–	View alarms +	us-e
□	terraform_cas...	i-054d17be5bf83db28	Stopped	t2.micro	–	View alarms +	us-e
☑	Harsh-Jadhav-...	i-094678856a0ebebfd	Running	t2.micro	2/2 checks passec	View alarms +	us-e
□	myexp4	i-09d29d9b1f480f274	Stopped	t2.medium	–	View alarms +	us-e
□	nagios-host	i-0efbfc7452c81f26d	Stopped	t2.medium	–	View alarms +	us-e
□	linux-client	i-039913541f26ee159	Stopped	t2.medium	–	View alarms +	us-e
□	pup-master	i-04b09b12644b629b5	Stopped	t2.medium	–	View alarms +	us-e
□	pup-client	i-0770f62efa30d9e3e	Stopped	t2.medium	–	View alarms +	us-e
□	client-p	i-024f0d83c14a1086a	Stopped	t2.medium	–	View alarms +	us-e
□	master-p	i-007275e7ae9ef3901	Stopped	t2.medium	–	View alarms +	us-e

- Copy the provided SSH command to connect to the instance.


aws
Services
Search
[Alt+S]
N. Virginia ▾
voclabs/user3389085=d2022.harsh.jadhav@ves.ac.in @ 8996-7863-... ▾
EC2 > Instances > i-094678856a0ebebfd > Connect to instance
Info
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Command copied
Cancel

Connect to instance

Connect to your instance i-094678856a0ebebfd (Harsh-Jadhav-23-WebServer) using any of these options

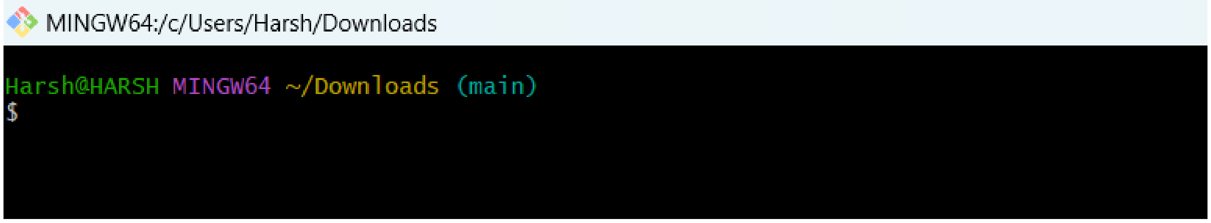
Instance ID
i-094678856a0ebebfd (Harsh-Jadhav-23-WebServer)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is case_study.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "case_study.pem"
4. Connect to your instance using its Public DNS:
   ec2-52-91-181-136.compute-1.amazonaws.com

```
ssh -i "case_study.pem" ec2-user@ec2-52-91-181-136.compute-1.amazonaws.com
```

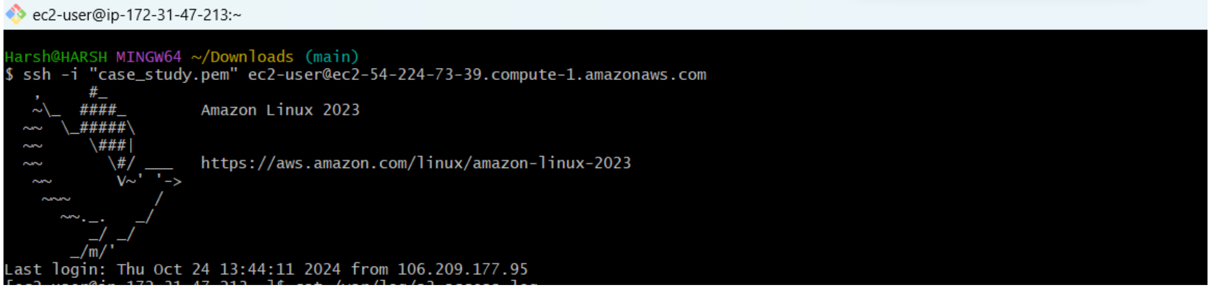
Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

- In your terminal, navigate to the directory where the .pem file was stored.



```
Harsh@HARSH MINGW64 ~/Downloads (main)
$
```

- Run the copied SSH command and type yes to confirm the connection.

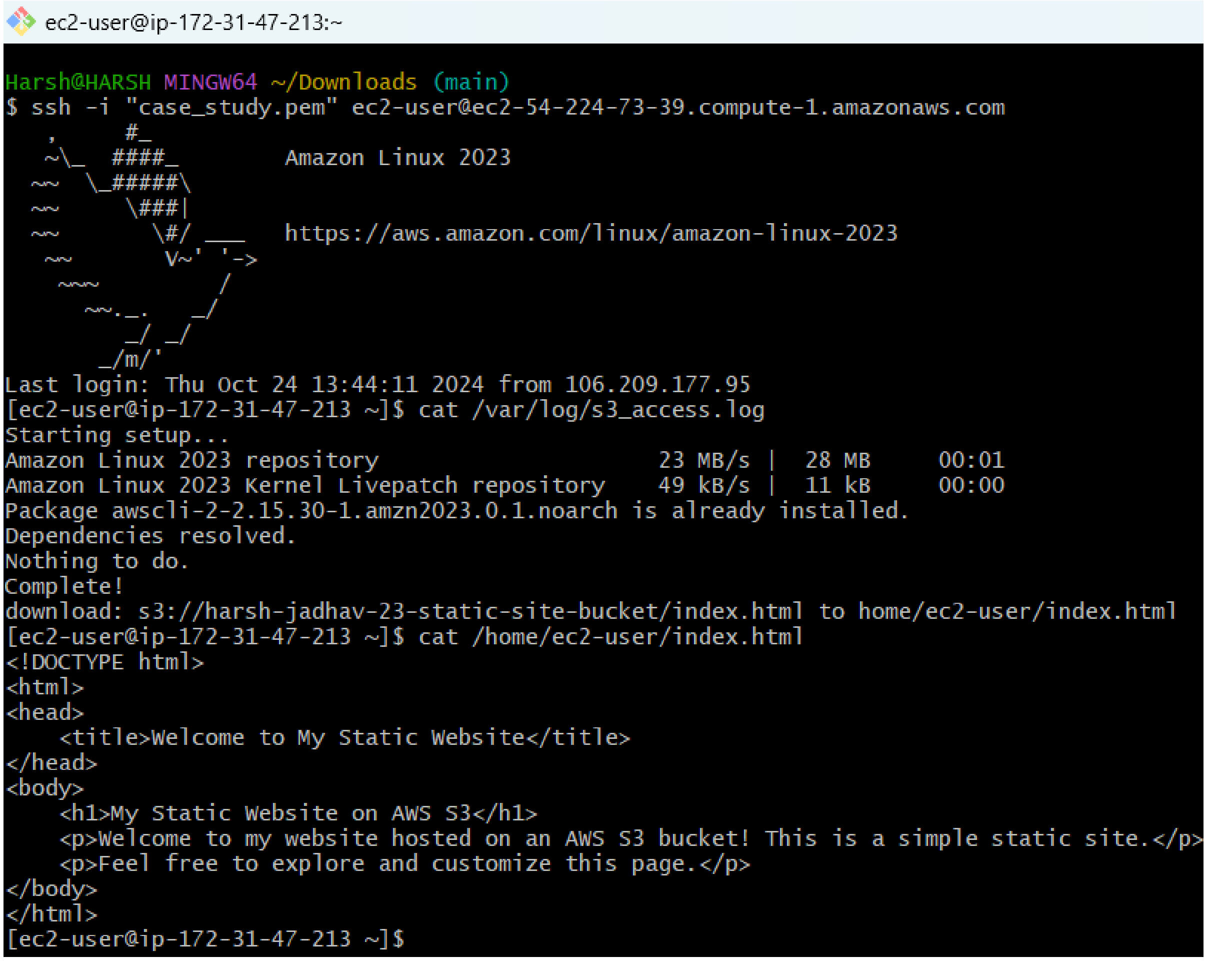


```
Harsh@HARSH MINGW64 ~/Downloads (main)
$ ssh -i "case_study.pem" ec2-user@ec2-54-224-73-39.compute-1.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Thu Oct 24 13:44:11 2024 from 106.209.177.95
```

7. View logs on EC2 instance:

- After connecting to the instance, run the following command to view the interaction logs:

cat /var/log/s3_access.log



ec2-user@ip-172-31-47-213:~

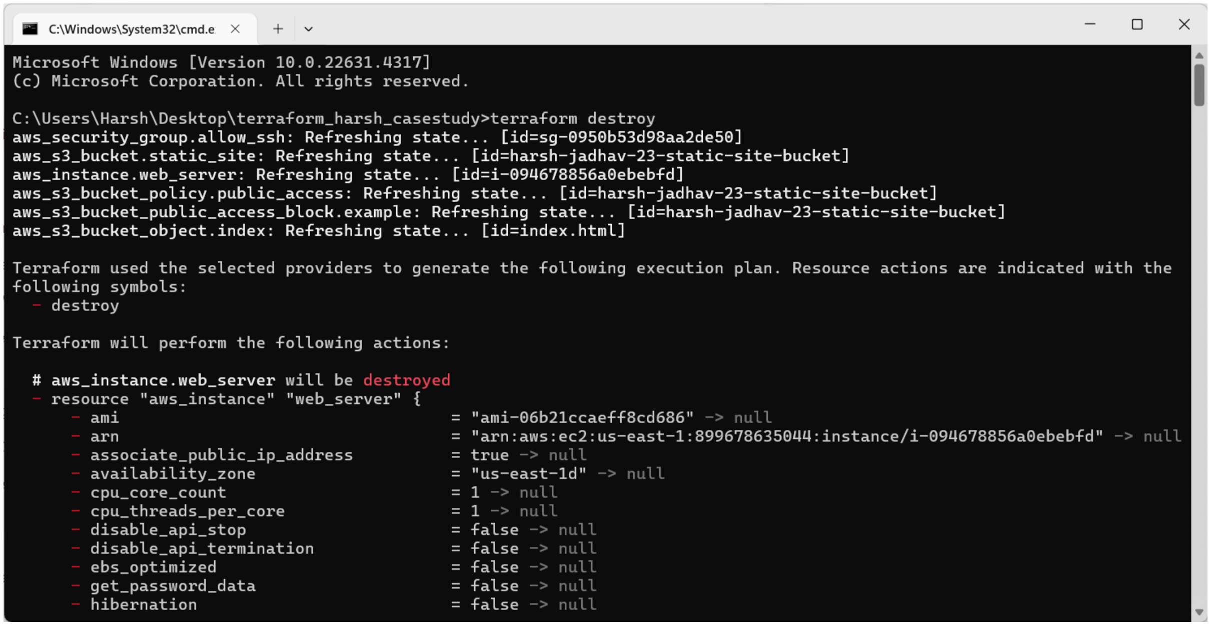
```
Harsh@HARSH MINGW64 ~/Downloads (main)
$ ssh -i "case_study.pem" ec2-user@ec2-54-224-73-39.compute-1.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Thu Oct 24 13:44:11 2024 from 106.209.177.95
[ec2-user@ip-172-31-47-213 ~]$ cat /var/log/s3_access.log
Starting setup...
Amazon Linux 2023 repository                     23 MB/s |  28 MB     00:01
Amazon Linux 2023 Kernel Livepatch repository    49 kB/s |  11 kB     00:00
Package awscli-2-2.15.30-1.amzn2023.0.1.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
download: s3://harsh-jadhav-23-static-site-bucket/index.html to home/ec2-user/index.html
[ec2-user@ip-172-31-47-213 ~]$ cat /home/ec2-user/index.html
<!DOCTYPE html>
<html>
<head>
    <title>Welcome to My Static Website</title>
</head>
<body>
    <h1>My Static Website on AWS S3</h1>
    <p>Welcome to my website hosted on an AWS S3 bucket! This is a simple static site.</p>
    <p>Feel free to explore and customize this page.</p>
</body>
</html>
[ec2-user@ip-172-31-47-213 ~]$
```

- The log will show the status of the AWS CLI installation, configuration, and S3 interaction.

8. Destroy resources:

- Once your testing is complete, destroy the resources to avoid incurring costs by running:

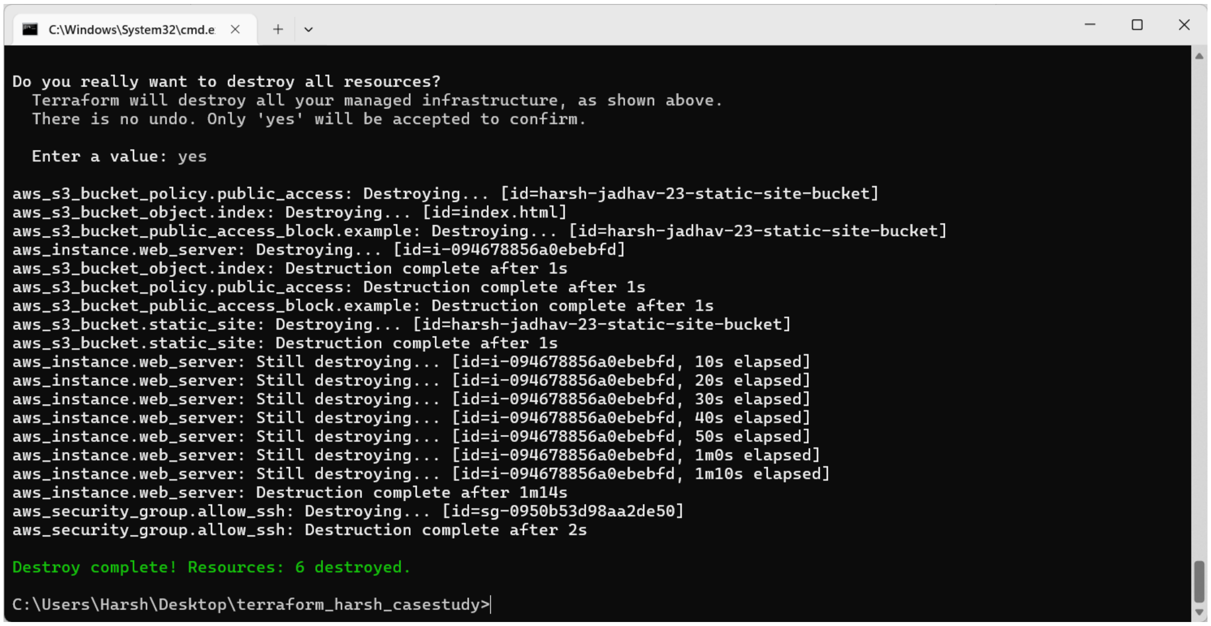
terraform destroy



C:\Windows\System32\cmd.e

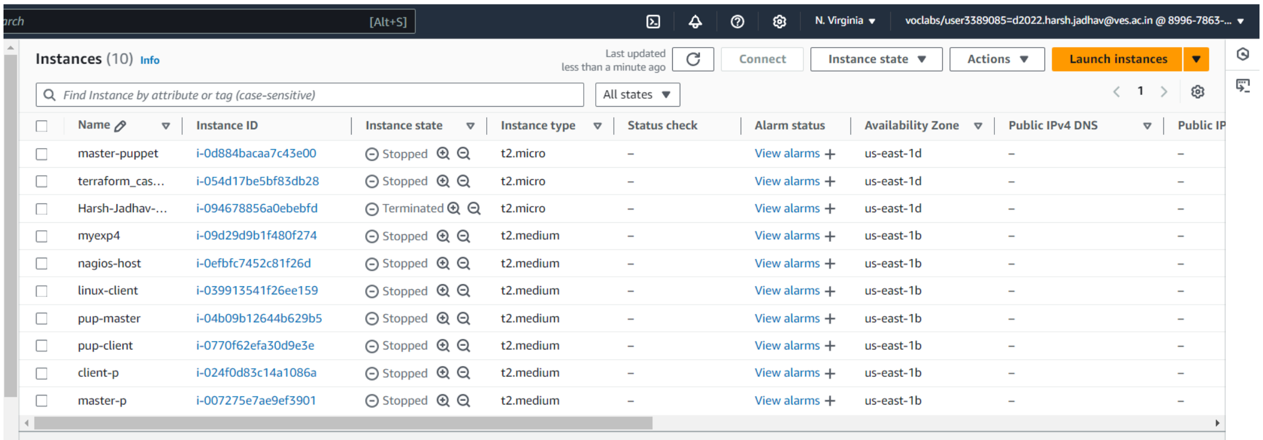
```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Harsh\Desktop\terraform_harsh_casestudy>terraform destroy
aws_security_group.allow_ssh: Refreshing state... [id=sg-0950b53d98aa2de50]
aws_s3_bucket.static_site: Refreshing state... [id=harsh-jadhav-23-static-site-bucket]
aws_instance.web_server: Refreshing state... [id=i-094678856a0ebebfd]
aws_s3_bucket_policy.public_access: Refreshing state... [id=harsh-jadhav-23-static-site-bucket]
aws_s3_bucket_public_access_block.example: Refreshing state... [id=harsh-jadhav-23-static-site-bucket]
aws_s3_bucket_object.index: Refreshing state... [id=index.html]

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

Terraform will perform the following actions:

  # aws_instance.web_server will be destroyed
  - resource "aws_instance" "web_server" {
      - ami                                  = "ami-06b21ccaeff8cd686" -> null
      - arn                                  = "arn:aws:ec2:us-east-1:899678635044:instance/i-094678856a0ebebfd" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "us-east-1d" -> null
      - cpu_core_count                       = 1 -> null
      - cpu_threads_per_core                 = 1 -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
```

- Confirm the destruction by typing yes



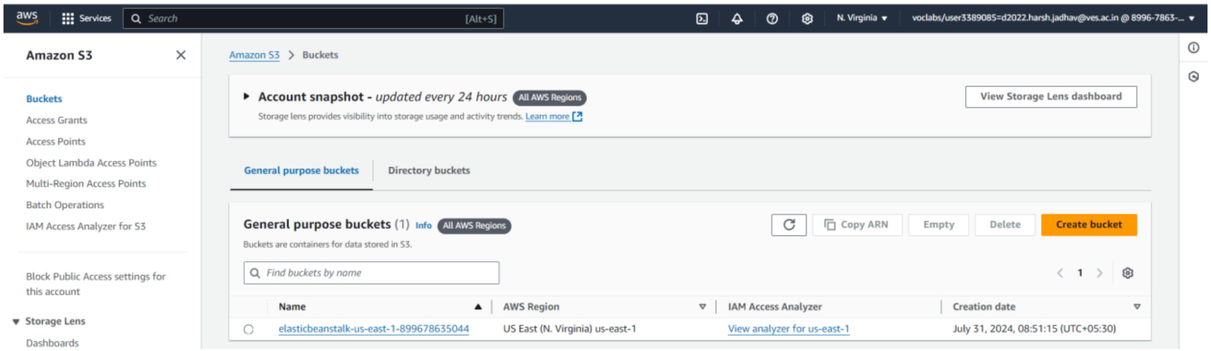
```
Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_s3_bucket_policy.public_access: Destroying... [id=harsh-jadhav-23-static-site-bucket]
aws_s3_bucket_object.index: Destroying... [id=index.html]
aws_s3_bucket_public_access_block.example: Destroying... [id=harsh-jadhav-23-static-site-bucket]
aws_instance.web_server: Destroying... [id=i-094678856a0ebebfd]
aws_s3_bucket_object.index: Destruction complete after 1s
aws_s3_bucket_policy.public_access: Destruction complete after 1s
aws_s3_bucket_public_access_block.example: Destruction complete after 1s
aws_s3_bucket.static_site: Destroying... [id=harsh-jadhav-23-static-site-bucket]
aws_s3_bucket.static_site: Destruction complete after 1s
aws_instance.web_server: Still destroying... [id=i-094678856a0ebebfd, 10s elapsed]
aws_instance.web_server: Still destroying... [id=i-094678856a0ebebfd, 20s elapsed]
aws_instance.web_server: Still destroying... [id=i-094678856a0ebebfd, 30s elapsed]
aws_instance.web_server: Still destroying... [id=i-094678856a0ebebfd, 40s elapsed]
aws_instance.web_server: Still destroying... [id=i-094678856a0ebebfd, 50s elapsed]
aws_instance.web_server: Still destroying... [id=i-094678856a0ebebfd, 1m0s elapsed]
aws_instance.web_server: Still destroying... [id=i-094678856a0ebebfd, 1m10s elapsed]
aws_instance.web_server: Destruction complete after 1m14s
aws_security_group.allow_ssh: Destroying... [id=sg-0950b53d98aa2de50]
aws_security_group.allow_ssh: Destruction complete after 2s

Destroy complete! Resources: 6 destroyed.

C:\Users\Harsh\Desktop\terraform_harsh_casestudy>
```

Ec2 instance terminated



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
□	master-puppet	i-0d884bacaa7c43e00	Stopped	t2.micro	–	View alarms +	us-east-1d	–	–
□	terraform_cas...	i-054d17be5bf83db28	Stopped	t2.micro	–	View alarms +	us-east-1d	–	–
□	Harsh-Jadhav-...	i-094678856a0ebebfd	Terminated	t2.micro	–	View alarms +	us-east-1d	–	–
□	myexp4	i-09d29d9b1f480f274	Stopped	t2.medium	–	View alarms +	us-east-1b	–	–
□	nagios-host	i-0efbfc7452c81f26d	Stopped	t2.medium	–	View alarms +	us-east-1b	–	–
□	linux-client	i-039913541f26ee159	Stopped	t2.medium	–	View alarms +	us-east-1b	–	–
□	pup-master	i-04b09b12644b629b5	Stopped	t2.medium	–	View alarms +	us-east-1b	–	–
□	pup-client	i-0770f62efa30d9e3e	Stopped	t2.medium	–	View alarms +	us-east-1b	–	–
□	client-p	i-024f0d83c14a1086a	Stopped	t2.medium	–	View alarms +	us-east-1b	–	–
□	master-p	i-007275e7ae9ef3901	Stopped	t2.medium	–	View alarms +	us-east-1b	–	–

S3 bucket deleted



aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3389085=d2022.harsh.jadhav@ves.ac.in @ 8996-7863-...
Amazon S3
Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Amazon S3 > Buckets
Account snapshot - updated every 24 hours
All AWS Regions
View Storage Lens dashboard
General purpose buckets
Directory buckets
General purpose buckets (1) Info
All AWS Regions
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
1

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

Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
○	elasticbeanstalk-us-east-1-899678635044	US East (N. Virginia) us-east-1	View analyzer for us-east-1	July 31, 2024, 08:51:15 (UTC+05:30)

Conclusion

In this exercise, Terraform was successfully used to provision an AWS EC2 instance and an S3 bucket, deploy a static website, and log interactions between the EC2 instance and the S3 bucket. The setup demonstrates the power of Infrastructure as Code to automate AWS provisioning and configuration.