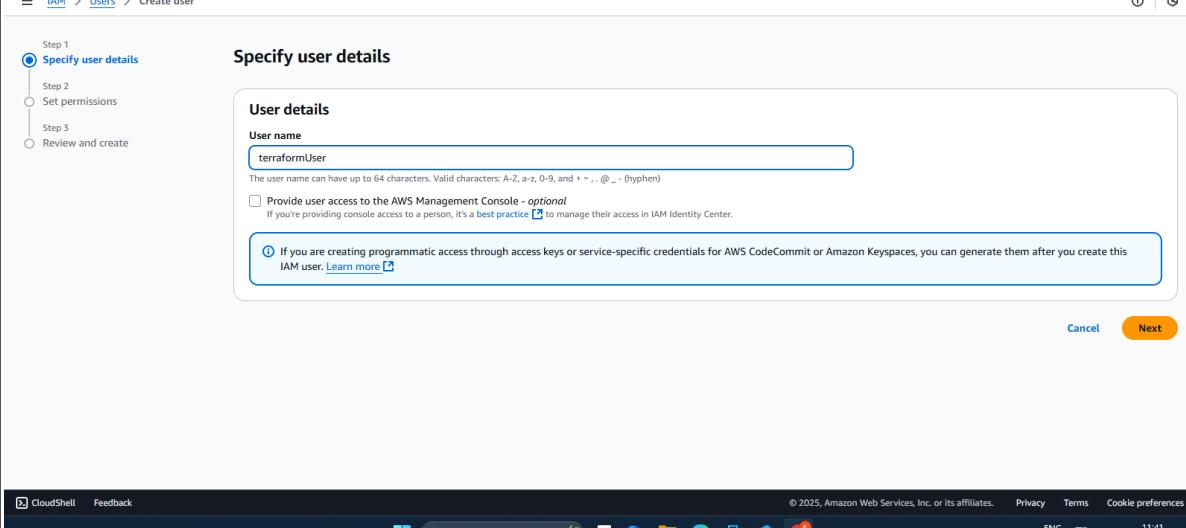


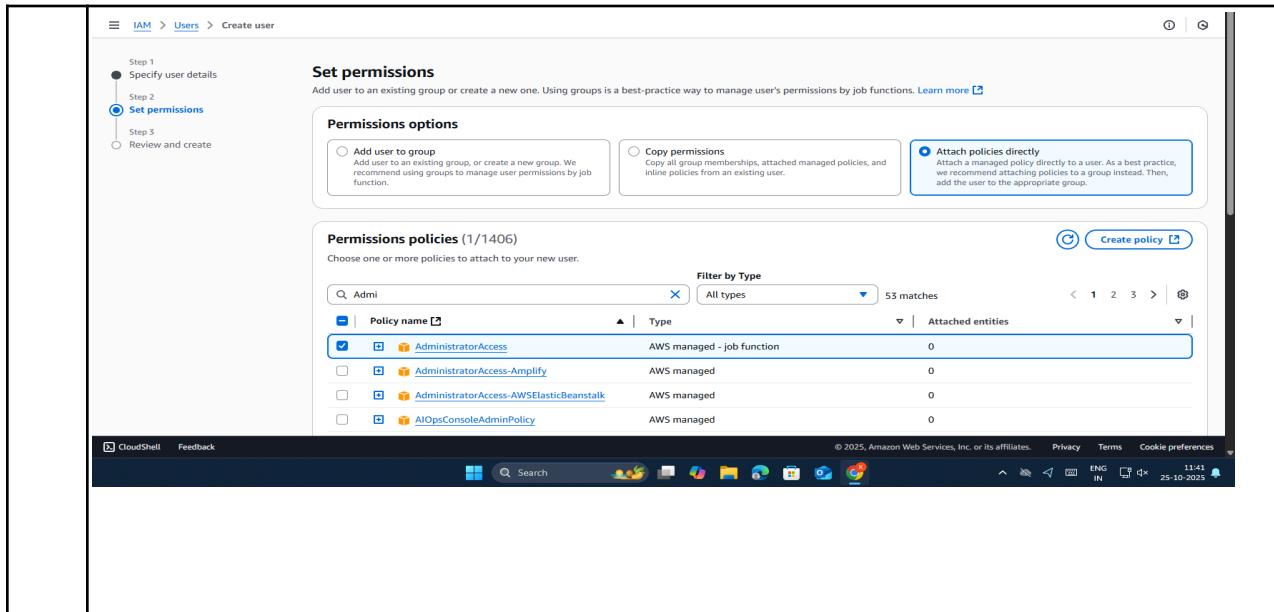
**St. Francis Institute of Technology**  
**Borivali (West), Mumbai-400103**  
**Department of Information Technology**  
**ADL Practical Exam**

Name: Khushi Senghani	Date:25/10/25
Class:TEITB3	Exam seat no:
PID: 231104	Time:11:15-12:15
Roll No:37	No. of Pages:
Duration: One Hour	

<b>Sr No</b>	<b>Problem Statement (Type here): To install terraform on a Windows/Linux machine and build,apply and destroy EC2 instance over AWS using terraform</b>
1.	<p>Step 1 :Login into aws account go to users select create user</p> <p>Screenshot 1:</p>  <p>Screenshot 2:</p> 
2.	<p>Step 2 (internal step 1):select the Attach policies directly option and select the AdministratorAccess</p> <p>Screenshot 2:</p>

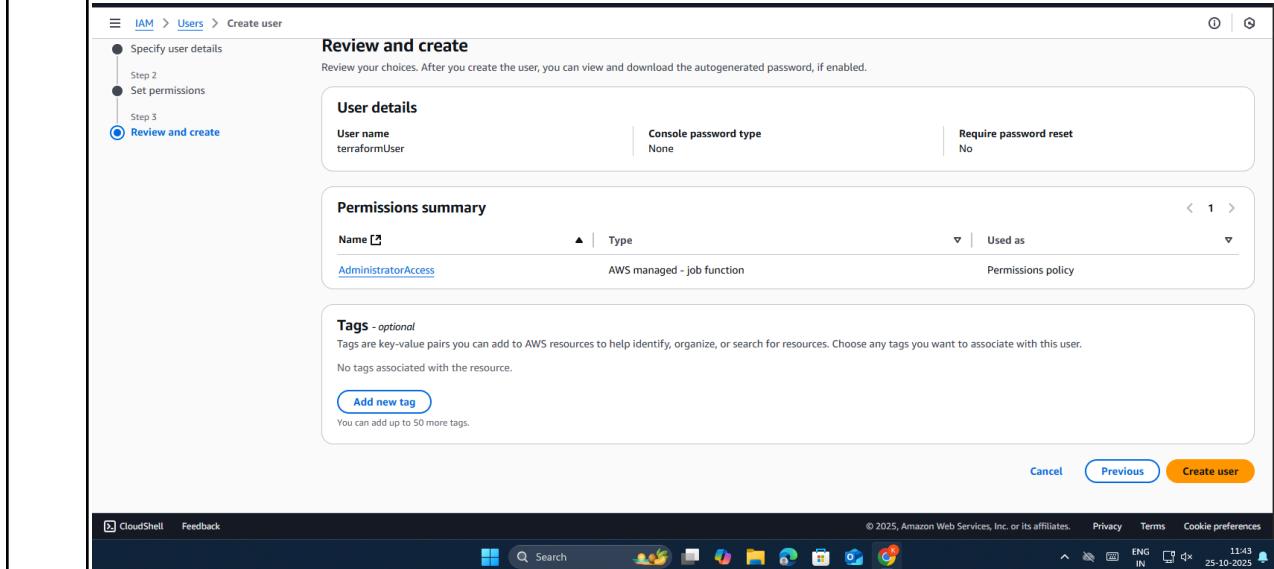
**St. Francis Institute of Technology**  
**Borivali (West), Mumbai-400103**  
**Department of Information Technology**  
**ADL Practical Exam**

Name: Khushi Senghani	Date:25/10/25
Class:TEITB3	Exam seat no:
PID: 231104	Time:11:15-12:15
Roll No:37	No. of Pages:
Duration: One Hour	



The screenshot shows the 'Set permissions' step of creating a user in AWS IAM. The 'Attach policies directly' option is selected. A list of policies is shown, with 'AdministratorAccess' being selected.

3. Step 3 (internal step 2): Click on create user and create the user successfully  
Screenshot 3:

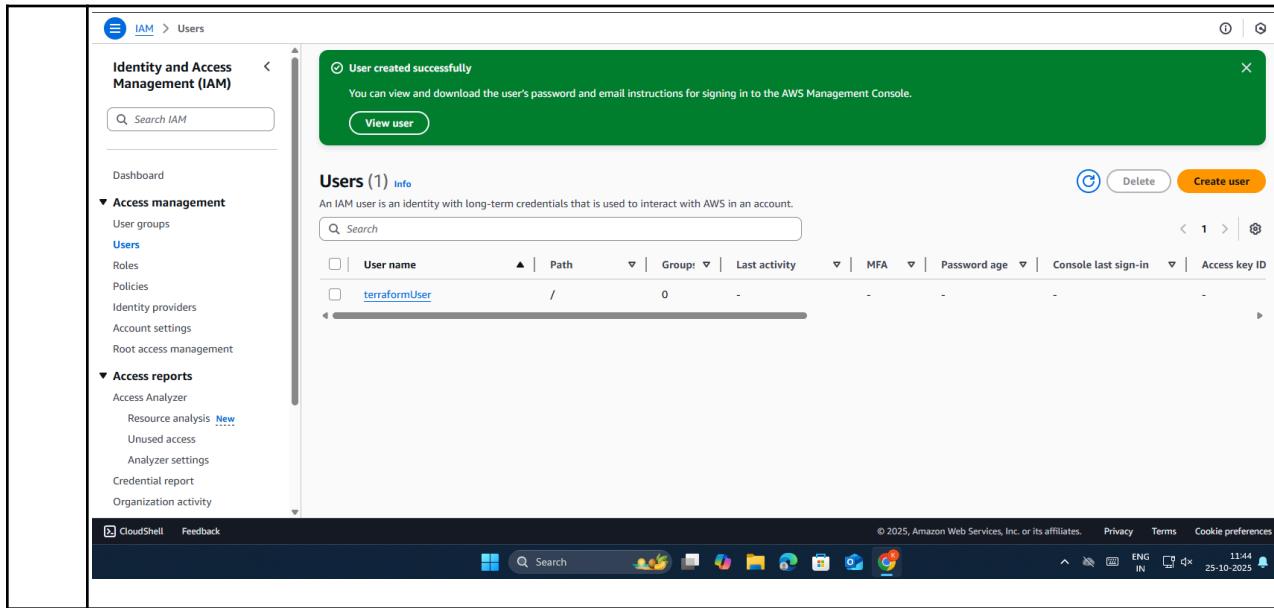


The screenshot shows the 'Review and create' step of creating a user in AWS IAM. It displays user details, a permissions summary (showing 'AdministratorAccess' selected), and a tags section. The 'Create user' button is at the bottom right.

4. Step 4 (internal step 3): The user is created successfully click on the user  
Screenshot 4:

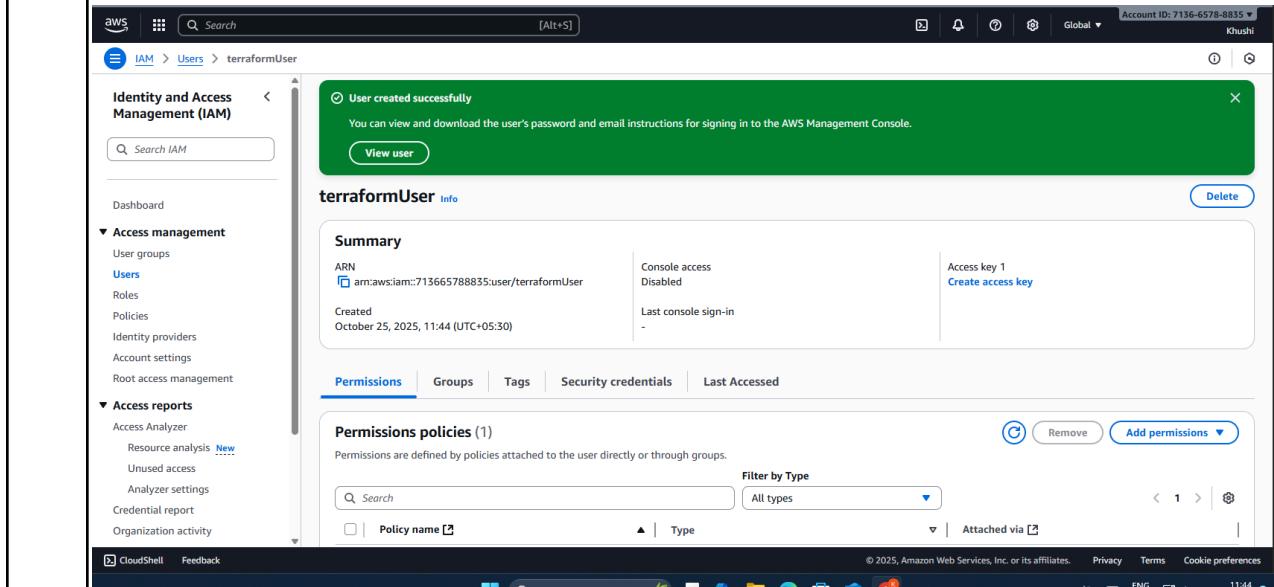
**St. Francis Institute of Technology**  
**Borivali (West), Mumbai-400103**  
**Department of Information Technology**  
**ADL Practical Exam**

Name: Khushi Senghani	Date:25/10/25
Class:TEITB3	Exam seat no:
PID: 231104	Time:11:15-12:15
Roll No:37	No. of Pages:
Duration: One Hour	



The screenshot shows the AWS Identity and Access Management (IAM) service. In the left sidebar, under 'Access management', 'Users' is selected. The main area displays a table titled 'Users (1) Info'. A green success message at the top states 'User created successfully' with a link to 'View user'. The table lists one user: 'terraformUser' with a path of '/' and an activity count of '0'. The bottom of the screen shows the standard AWS navigation bar.

5. Step 5 (stop/delete step) :click on create access key  
Screenshot 5:

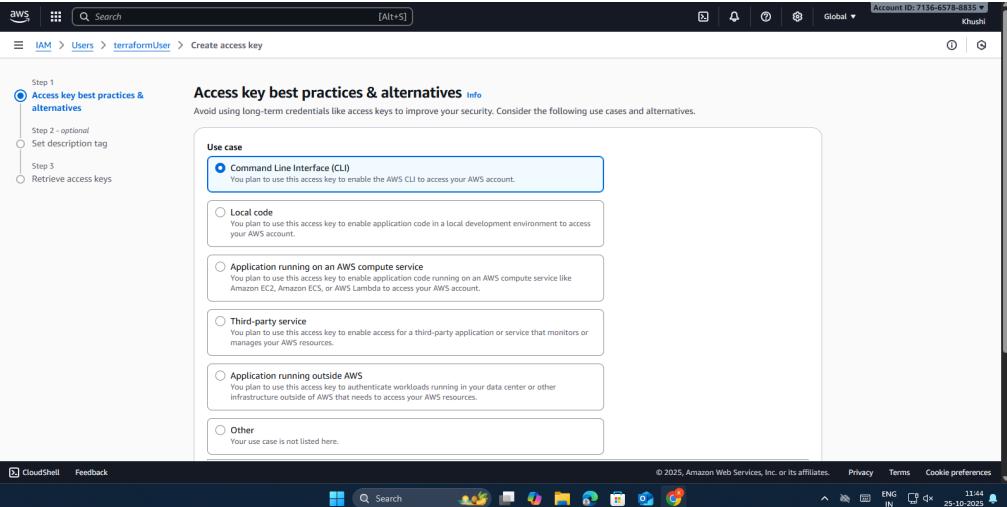
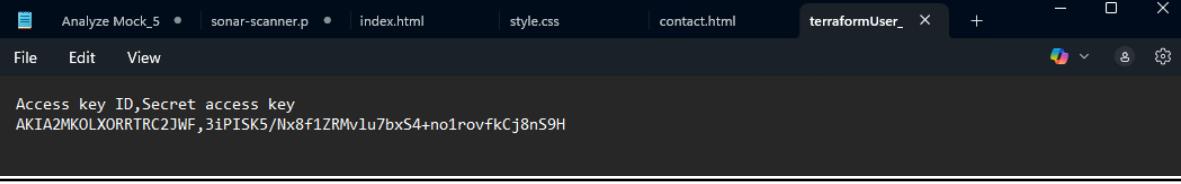
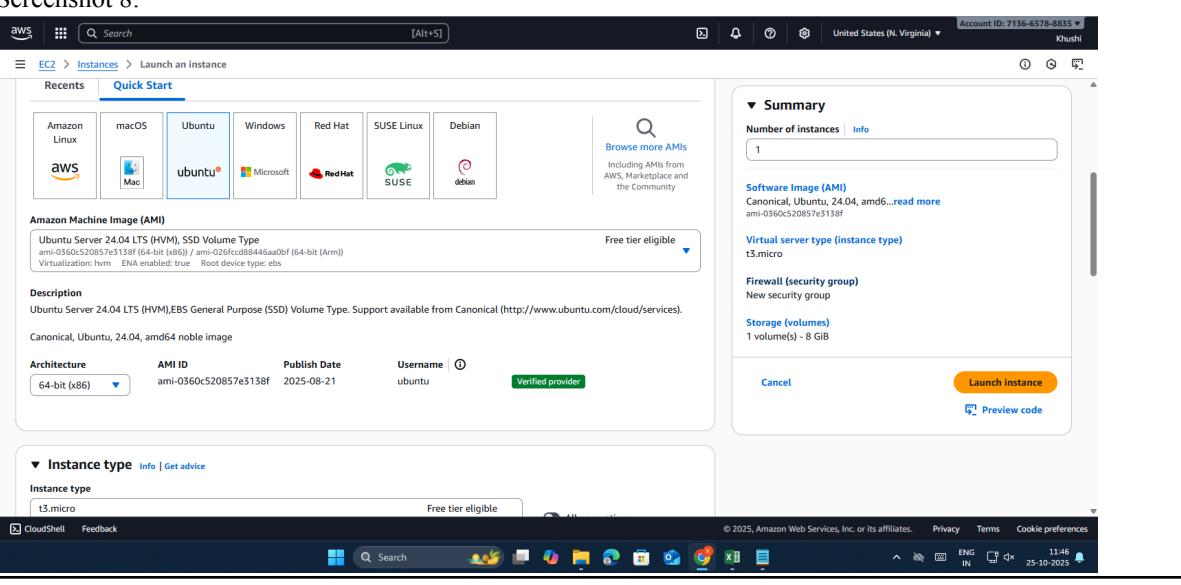


This screenshot shows the detailed view for the 'terraformUser' created in the previous step. The top banner again says 'User created successfully'. Below it, the 'terrafromUser' info card shows basic details like ARN, creation date, and console access status. On the right, there's a section for 'Access key 1' with a 'Create access key' button. At the bottom, the 'Permissions' tab is active, showing a table for 'Permissions policies (1)' with a single row for 'Policy name'. The bottom of the screen shows the AWS navigation bar.

- Step 6:Select command line interface  
Screenshot 6:

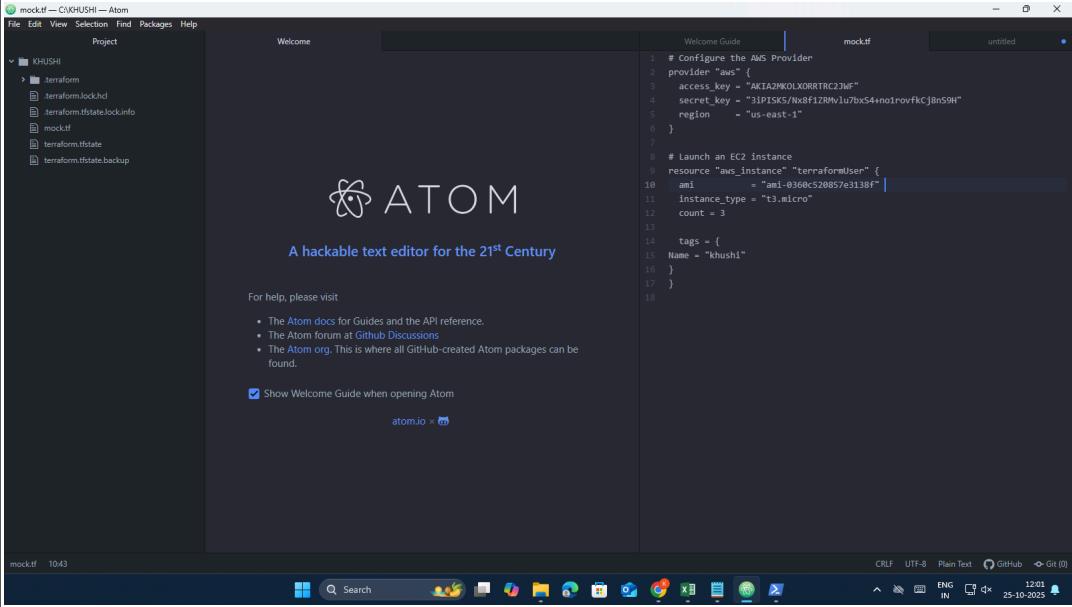
**St. Francis Institute of Technology**  
**Borivali (West), Mumbai-400103**  
**Department of Information Technology**  
**ADL Practical Exam**

Name: Khushi Senghani	Date:25/10/25
Class:TEITB3	Exam seat no:
PID: 231104	Time:11:15-12:15
Roll No:37	No. of Pages:
Duration: One Hour	

	 <p>The screenshot shows the 'Access key best practices &amp; alternatives' step in the AWS IAM 'Create access key' wizard. It lists several use cases for the access key:</p> <ul style="list-style-type: none"> <li><b>Command Line Interface (CLI)</b>: You plan to use this access key to enable the AWS CLI to access your AWS account.</li> <li><b>Local code</b>: You plan to use this access key to enable application code in a local development environment to access your AWS account.</li> <li><b>Application running on an AWS compute service</b>: You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.</li> <li><b>Third-party service</b>: You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.</li> <li><b>Application running outside AWS</b>: You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.</li> <li><b>Other</b>: Your use case is not listed here.</li> </ul>
	<p>Step 7: Download the csv file and copy the access key and secret key  Screenshot 7:</p>  <p>The screenshot shows a browser window with multiple tabs open. The active tab is titled 'terraformUser_'. It contains the text 'Access key ID, Secret access key' followed by the copied values: AKIA2MKOLXORRTRC2JWF, 3iPISK5/Nx8f1ZRMvlu7bxS4+no1rovfkCj8nS9H.</p>
	<p>Step 8: From launch instance copy the ami id  Screenshot 8:</p>  <p>The screenshot shows the 'Launch an instance' step in the AWS EC2 'Launch instances' wizard. It displays the 'Quick Start' section with various AMI options. The selected AMI is 'Ubuntu Server 24.04 LTS (HVM), SSD Volume Type' (ami-0360c520857e3138f). The summary panel on the right shows the selected AMI as Canonical, Ubuntu, 24.04, amd64... and the instance type as t3.micro.</p>
	<p>Step 9: Create an atom file and write the code.  Screenshot 9:</p>

**St. Francis Institute of Technology**  
**Borivali (West), Mumbai-400103**  
**Department of Information Technology**  
**ADL Practical Exam**

Name: Khushi Senghani	Date:25/10/25
Class:TEITB3	Exam seat no:
PID: 231104	Time:11:15-12:15
Roll No:37	No. of Pages:
Duration: One Hour	



```

# Configure the AWS Provider
provider "aws" {
  access_key = "AKIA2HKOLXORRTRC2JME"
  secret_key = "3IPISK5/1nx8f1Z8Wvlu7bx54+no1rovfkCjBn59H"
  region     = "us-east-1"
}

# Launch an EC2 instance
resource "aws_instance" "terraformUser" {
  ami           = "ami-036b528857e313bf"
  instance_type = "t3.micro"
  count         = 3

  tags = {
    Name = "khushi"
  }
}

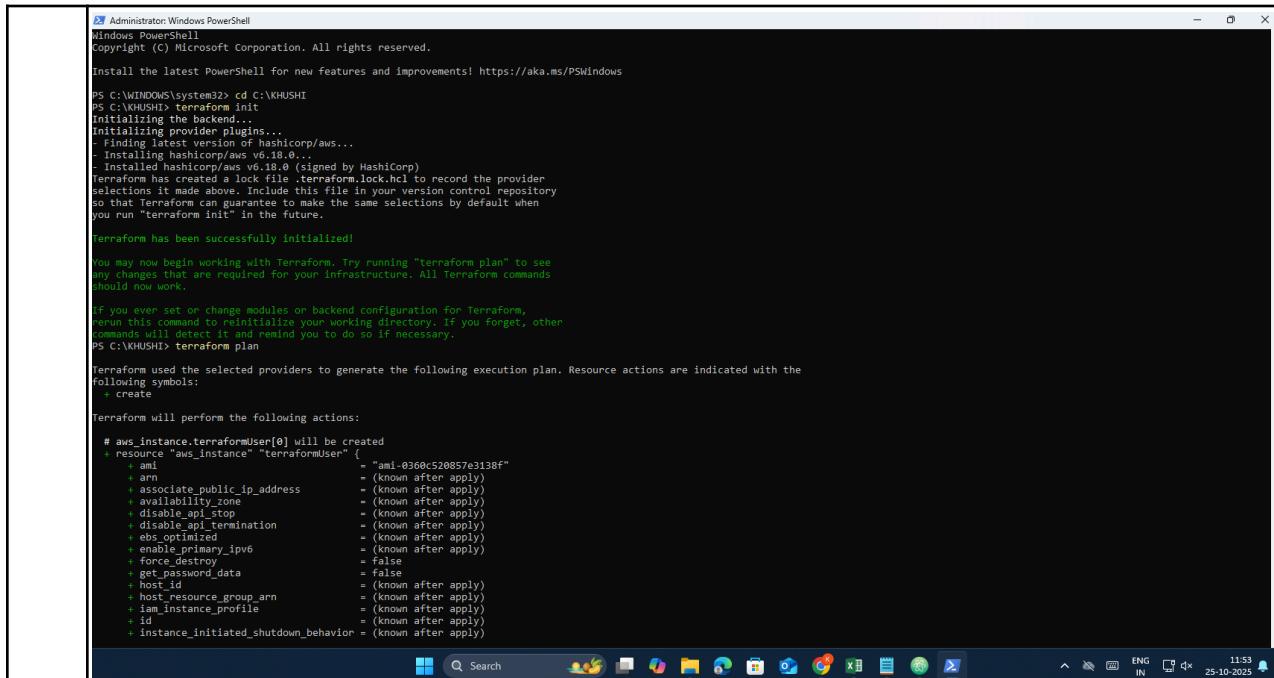
```

Screenshot of File Explorer showing the directory structure: KHUSHI/.terraform/.terraform.lock.hcl, mock, terraform.tfstate, terraform.tfstate.backup.

| Step 10:open powershell as administrator and give the path to your file and perform terraform init command Screenshot 10: |

**St. Francis Institute of Technology**  
**Borivali (West), Mumbai-400103**  
**Department of Information Technology**  
**ADL Practical Exam**

Name: Khushi Senghani	Date:25/10/25
Class:TEITB3	Exam seat no:
PID: 231104	Time:11:15-12:15
Roll No:37	No. of Pages:
Duration: One Hour	



```

Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> cd C:\KHUSHI
PS C:\KHUSHI> terraform init
Initializing the backend...
Initializing provider plugins...
[Truncated]
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 on 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.
PS C:\KHUSHI> terraform plan

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

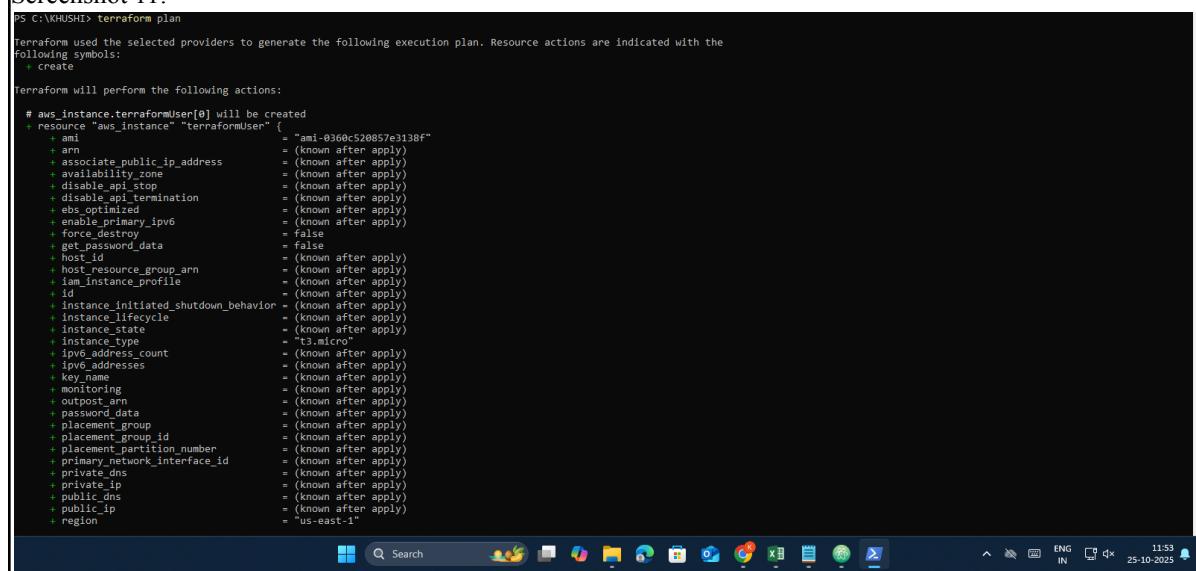
Terraform will perform the following actions:

# aws_instance.terraformUser[0] will be created
+ resource "aws_instance" "terraformUser" {
    + ami                               = "ami-0360c520857e3138f"
    + ami_id                           = "(known after apply)"
    + arn                             = "(known after apply)"
    + associate_public_ip_address      = "(known after apply)"
    + availability_zone                = "(known after apply)"
    + disable_api_stop                 = "(known after apply)"
    + disable_api_termination          = "(known after apply)"
    + ebs_optimized                   = "(known after apply)"
    + enable_primary_ipv6              = "(known after apply)"
    + force_destroy                    = false
    + get_password_data               = "(known after apply)"
    + 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)"
}

```

Step 11:perform terraform plan command.

Screenshot 11:



```

PS C:\KHUSHI> 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.terraformUser[0] will be created
+ resource "aws_instance" "terraformUser" {
    + ami                               = "ami-0360c520857e3138f"
    + ami_id                           = "(known after apply)"
    + arn                             = "(known after apply)"
    + associate_public_ip_address      = "(known after apply)"
    + availability_zone                = "(known after apply)"
    + disable_api_stop                 = "(known after apply)"
    + disable_api_termination          = "(known after apply)"
    + ebs_optimized                   = "(known after apply)"
    + enable_primary_ipv6              = "(known after apply)"
    + force_destroy                    = false
    + get_password_data               = "(known after apply)"
    + 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
    + instance_type
    + ipv6_address_count
    + ipv6_addresses
    + key_name
    + monitoring
    + outpost_arn
    + private_ip
    + public_ip
    + region
}
```

Step 12:Perform terraform apply command

Screenshot 12:

**St. Francis Institute of Technology  
Borivali (West), Mumbai-400103  
Department of Information Technology  
ADL Practical Exam**

Name: Khushi Senghani	Date:25/10/25
Class:TEITB3	Exam seat no:
PID: 231104	Time:11:15-12:15
Roll No:37	No. of Pages:
Duration: One Hour	

```
s C:\KHUSHI> terraform apply

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.terraformUser[0] will be created
+ resource "aws_instance" "terraformUser" {
    ami           = "ami-03360c20857e3138f"
    amiOwnerId    = "(known after apply)"
    associate_public_ip_address = "(known after apply)"
    availability_zone      = "(known after apply)"
    disable_api_stop       = "(known after apply)"
    disable_ami_termination = "(known after apply)"
    ebs_optimized          = "(known after apply)"
    enable_primary_ipv6    = "(known after apply)"
    ephemeral_storage      = true
    get_password_data     = false
    host_id             = "(known after apply)"
    host_resource_group_arn = "(known after apply)"
    iam_instance_profile = "(known after apply)"
    id                 = "i-05f0e022c1b3cb418e"
    instance_initiated_shutdown_behavior = "(known after apply)"
    instance.lifecycle   = "(known after apply)"
    instance.state      = "(known after apply)"
    instance_type       = "t3.micro"
    ipv6.address_count  = "(known after apply)"
    ipv6.addresses      = "(known after apply)"
    key_name            = "(known after apply)"
    monitoring          = "(known after apply)"
    outpost_arn         = "(known after apply)"
    password_data       = "(known after apply)"
    placement_group     = "(known after apply)"
    placement_group_id = "(known after apply)"
    placement_partition_number = "(known after apply)"
    placement_interface_id = "(known after apply)"
    private_dns         = "(known after apply)"
    private_ip          = "(known after apply)"
    public_dns          = "(known after apply)"
    public_ip           = "(known after apply)"
    region              = "us-east-1"
    security_group_ids = "(known after apply)"
    security_groups     = "(known after apply)"
    source_dest_check   = true
    spot_instance_request_id = "(known after apply)"

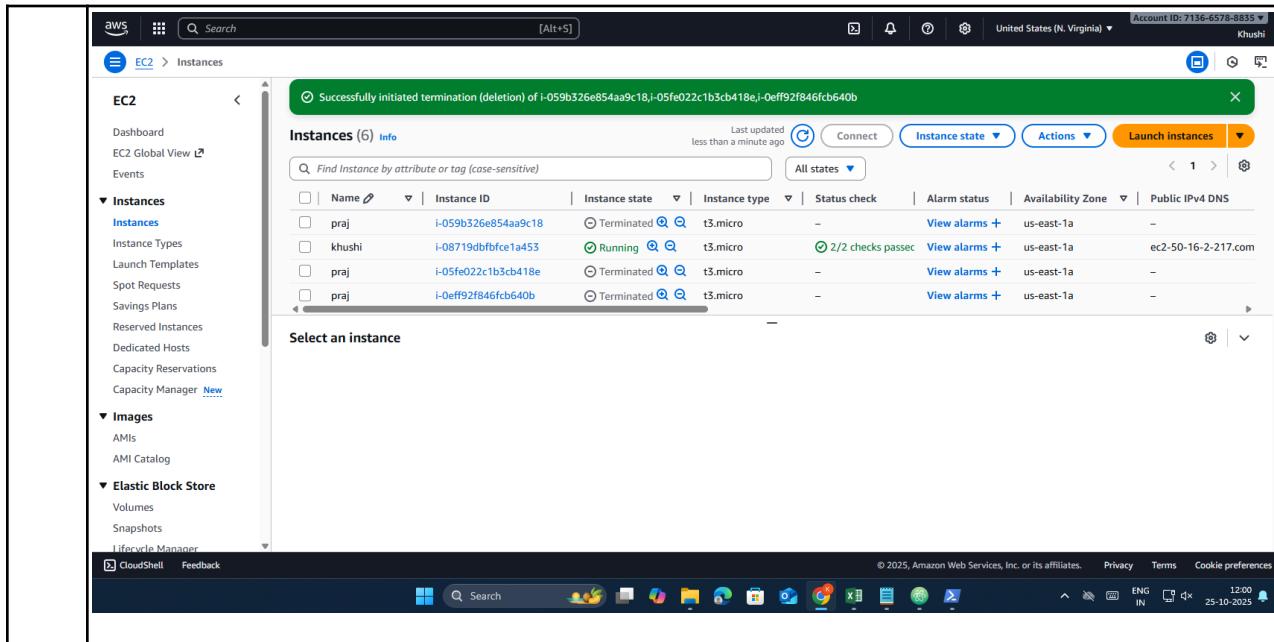
aws_instance.terraformUser[2]: Creating...
aws_instance.terraformUser[0]: Creating...
aws_instance.terraformUser[1]: Creating...
aws_instance.terraformUser[2]: Still creating... [00m10s elapsed]
aws_instance.terraformUser[0]: Still creating... [00m10s elapsed]
aws_instance.terraformUser[1]: Still creating... [00m10s elapsed]
aws_instance.terraformUser[2]: Creation complete after 17s [id=i-0eff92f846fc640b]
aws_instance.terraformUser[1]: Creation complete after 19s [id=i-05f0e022c1b3cb418e]
aws_instance.terraformUser[0]: Creation complete after 20s [id=i-059b326e854aa9c18]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
```

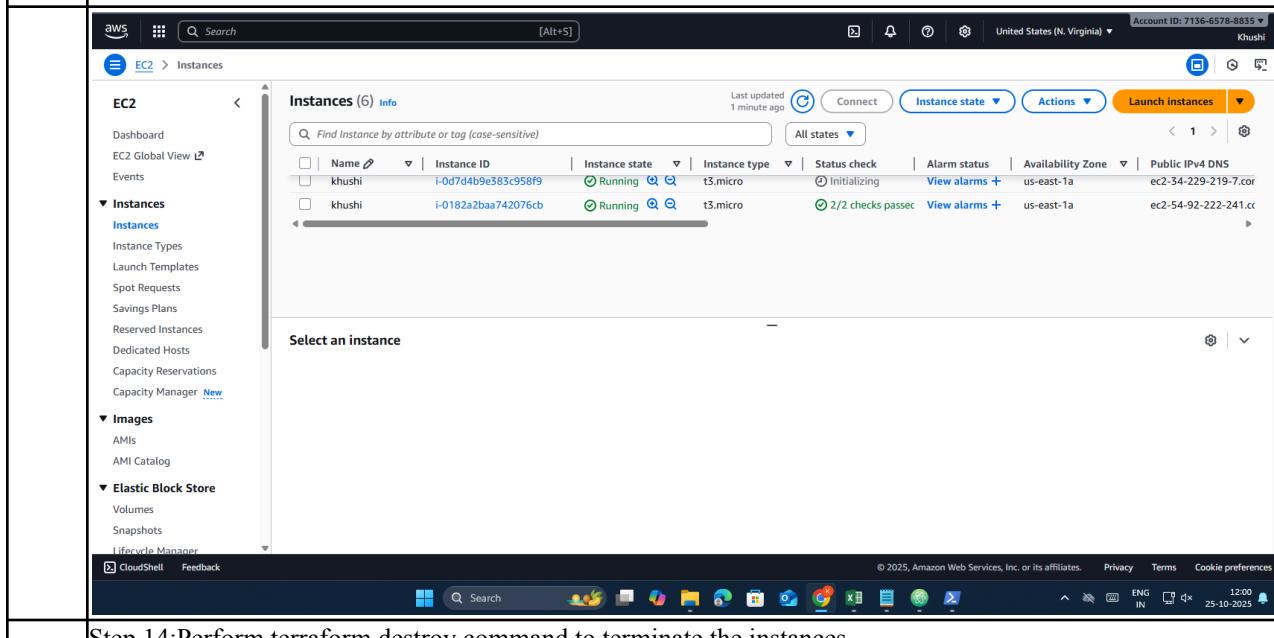
Step 13: The instances with tag khushi is shown running.  
Screenshot 13:

**St. Francis Institute of Technology**  
**Borivali (West), Mumbai-400103**  
**Department of Information Technology**  
**ADL Practical Exam**

Name: Khushi Senghani	Date:25/10/25
Class:TEITB3	Exam seat no:
PID: 231104	Time:11:15-12:15
Roll No:37	No. of Pages:
Duration: One Hour	



The screenshot shows the AWS EC2 Instances page. A green success message at the top states: "Successfully initiated termination (deletion) of i-059b326e854aa9c18, i-05fe022c1b3cb418e, i-0eff92f846fc640b". The main table lists six instances, all of which have been terminated. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. The instances listed are: praj (terminated), khushi (running), praj (terminated), praj (terminated), and two more praj entries (both terminated).

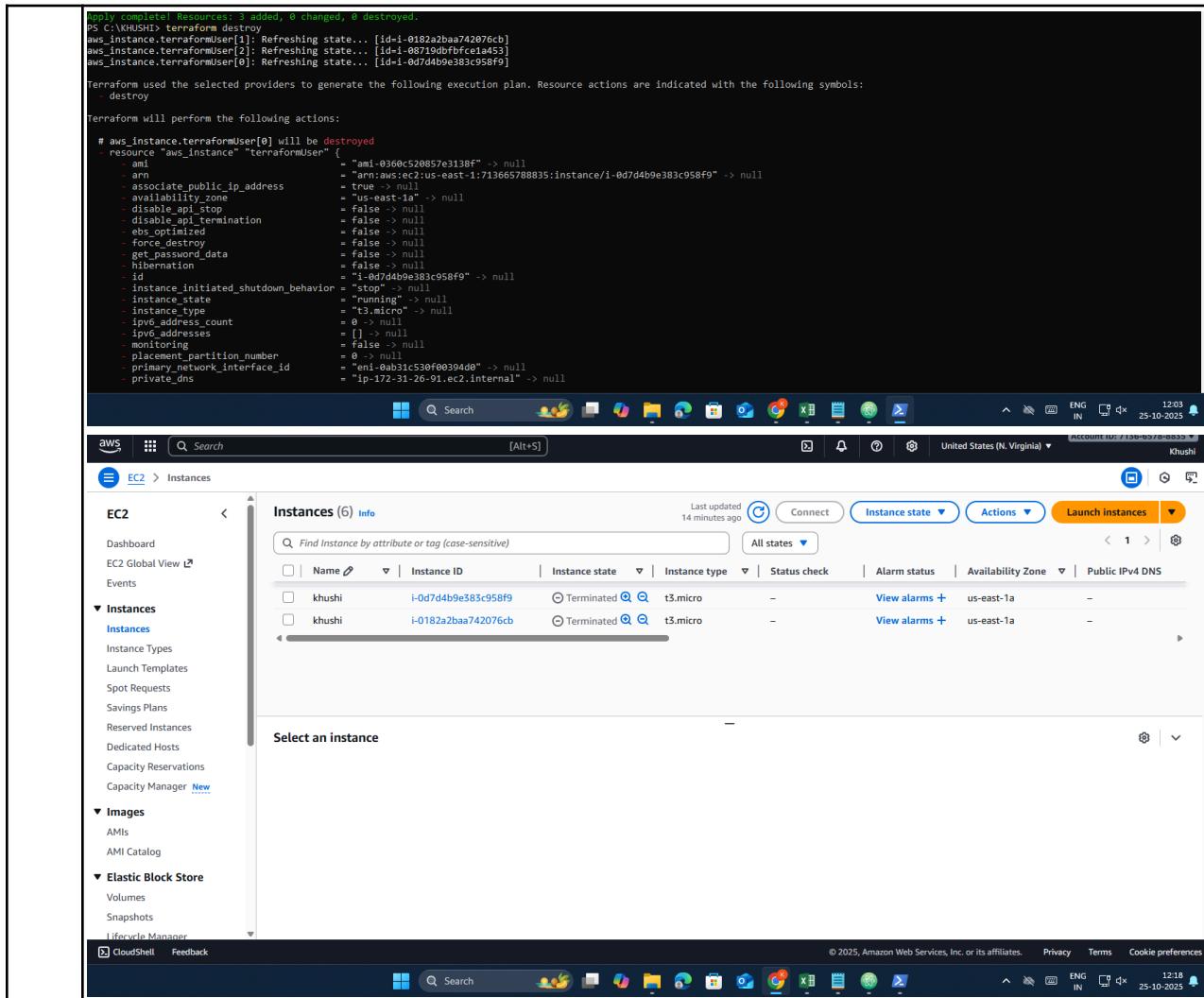


The screenshot shows the AWS EC2 Instances page. A green success message at the top states: "Successfully initiated termination (deletion) of i-059b326e854aa9c18, i-05fe022c1b3cb418e, i-0eff92f846fc640b". The main table lists six instances, all of which are now running. The table includes columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. The instances listed are: khushi (running), khushi (running), and two more khushi entries (both running).

Step 14:Perform terraform destroy command to terminate the instances.  
Screenshot 14:

**St. Francis Institute of Technology**  
**Borivali (West), Mumbai-400103**  
**Department of Information Technology**  
**ADL Practical Exam**

Name: Khushi Senghani	Date:25/10/25
Class:TEITB3	Exam seat no:
PID: 231104	Time:11:15-12:15
Roll No:37	No. of Pages:
Duration: One Hour	



The screenshot shows two windows side-by-side. The left window is a terminal window titled 'CloudShell' with the command 'terraform apply' running. The output shows the creation of three AWS instances (t3.micro) in the us-east-1a availability zone. The right window is the AWS Management Console's EC2 Instances page, showing the same three terminated instances listed under the 'Instances' section.

```

apply complete! Resources: 3 added, 0 changed, 0 destroyed.
[+] CREATING instance for destroy...
aws_instance.terraformUser[1]: Refreshing state... [id=i-0182a2baa742076cb]
aws_instance.terraformUser[2]: Refreshing state... [id=i-08710dbfbfe1a453]
aws_instance.terraformUser[0]: Refreshing state... [id=i-0d7d4b9e383c958f9]

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 "terraformUser[0]" will be destroyed
resource "aws_instance" "terraformUser" {
  ami           = "ami-038cc520857e3138f" -> null
  ami_hidden   = "ami-038cc520857e3138f" -> null
  arn          = "arn:aws:ec2:us-east-1:713665708835:instance/i-0d7d4b9e383c958f9" -> null
  associate_public_ip_address = true -> null
  availability_zone      = "us-east-1a" -> null
  disable_api_stop        = false -> null
  disable_api_termination = false -> null
  ebs_optimized           = false -> null
  force_destroy           = false -> null
  get_password_data       = false -> null
  hibernation             = false -> null
  id                      = "i-0d7d4b9e383c958f9" -> null
  instance_initiated_shutdown_behavior = "stop" -> null
  instance_state          = "running" -> null
  instance_type            = "t3.micro" -> null
  ip_address_count         = 0 -> null
  ipv6_addresses           = [] -> null
  monitoring              = false -> null
  placement_partition_number = 0 -> null
  primary_network_interface_id = "eni-0ab31c520f00394d0" -> null
  private_dns              = "ip-172-31-26-91.ec2.internal" -> null
}

```

**EC2 > Instances**

**Instances (6) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
khushi	i-0d7d4b9e383c958f9	Terminated	t3.micro	-	<a href="#">View alarms +</a>	us-east-1a	-
khushi	i-0182a2baa742076cb	Terminated	t3.micro	-	<a href="#">View alarms +</a>	us-east-1a	-

Select an instance