PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)

Semester:V **NameofStudent:Tanmay Padule**
Academic Year: 2025-26 **StudentID:23104156**
Class / Branch: TE IT B
Subject:AdvancedDevops Lab(ADL)
NameofInstructor:Prof.ManjushaK.

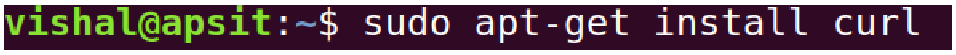
EXPERIMENTNO.06

Aim:ToBuild,change,anddestroyAWSinfrastructureUsingTerraform.

Pre-requistes:

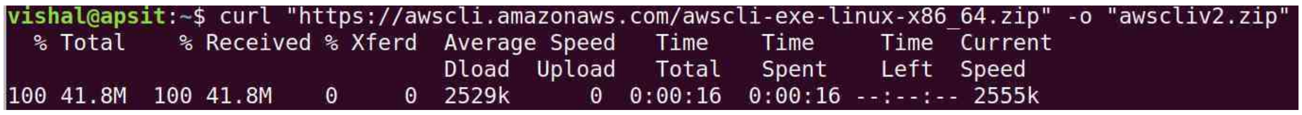
1. InstalltheAWSCLIversion2onLinux

FollowthesestepsfromthecommandlinetoinstalltheAWSCLIonLinux.
Installcurlonlinux



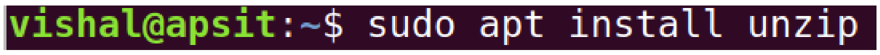
```
vishal@apsit:~$ sudo apt-get install curl
```

vishal@apsit:~$curl"https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip"-o"awscliv2.zip"



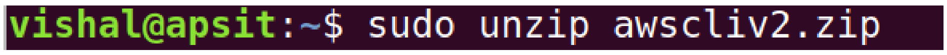
```
vishal@apsit:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 41.8M  100 41.8M    0     0  2529k      0  0:00:16  0:00:16 --:--:-- 2555k
```

vishal@apsit:~$sudoaptinstallunzip



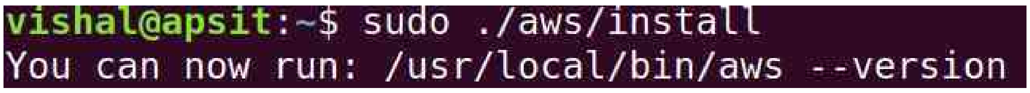
```
vishal@apsit:~$ sudo apt install unzip
```

vishal@apsit:~$sudounzipawscliv2.zip



```
vishal@apsit:~$ sudo unzip awscliv2.zip
```

vishal@apsit:~$sudo./aws/install

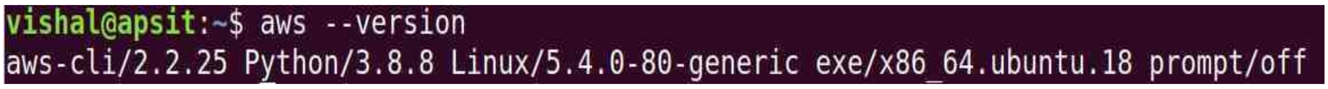


```
vishal@apsit:~$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
```

vishal@apsit:~$aws--version

itshoulddisplaythebelowoutout.

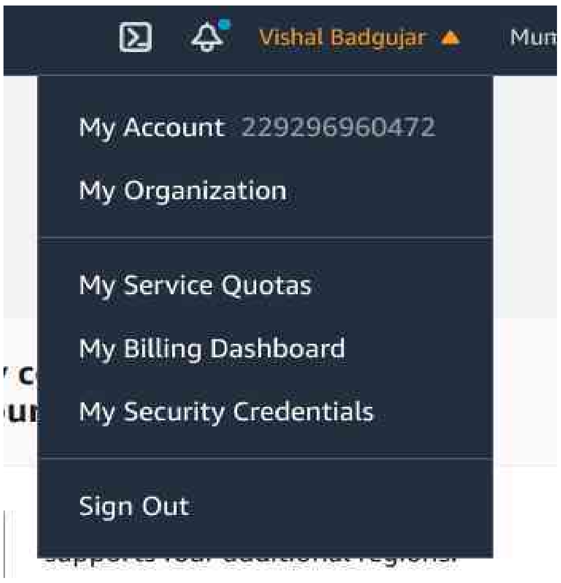
aws-cli/2.1.29Python/3.8.8Linux/5.4.0-1038-awsexe/x86_64.ubuntu.18prompt/off

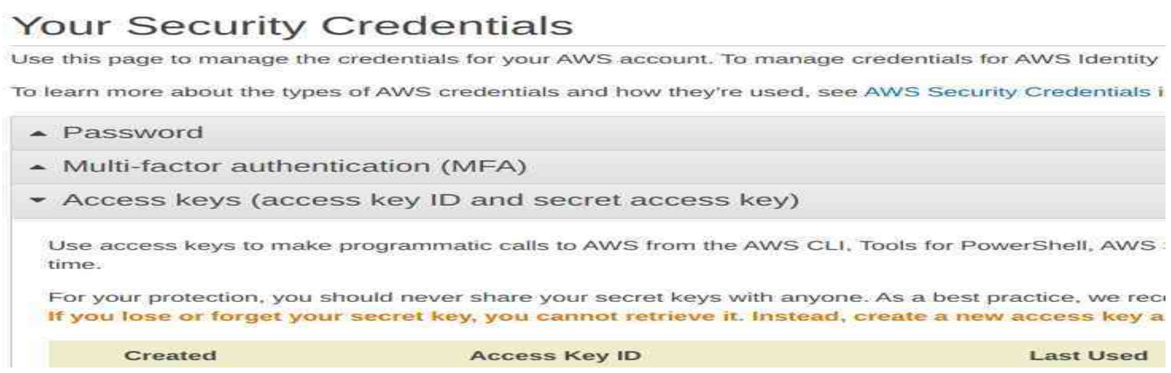


```
vishal@apsit:~$ aws --version
aws-cli/2.2.25 Python/3.8.8 Linux/5.4.0-80-generic exe/x86_64.ubuntu.18 prompt/off
```

2. Createanewaccesskeyifyoudon'thaveone.Makesureyoudownloadthekeysinyour local machine.

LogintoAWSconsole,clickonusernameandgotoMysecuritycredentials.




Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity

To learn more about the types of AWS credentials and how they're used, see AWS Security Credentials i

- Password
- Multi-factor authentication (MFA)
- Access keys (access key ID and secret access key)

Use access keys to make programmatic calls to AWS from the AWS CLI, Tools for PowerShell, AWS time.

For your protection, you should never share your secret keys with anyone. As a best practice, we rec
If you lose or forget your secret key, you cannot retrieve it. Instead, create a new access key a

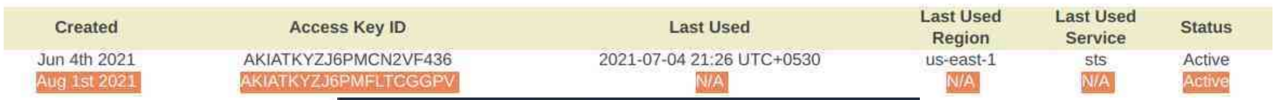
Created	Access Key ID	Last Used

Continueonsecuritycredentials,clickonaccesskeys

Performbelowcommandsin Linuxwhereyouhaveinstalled Terraform

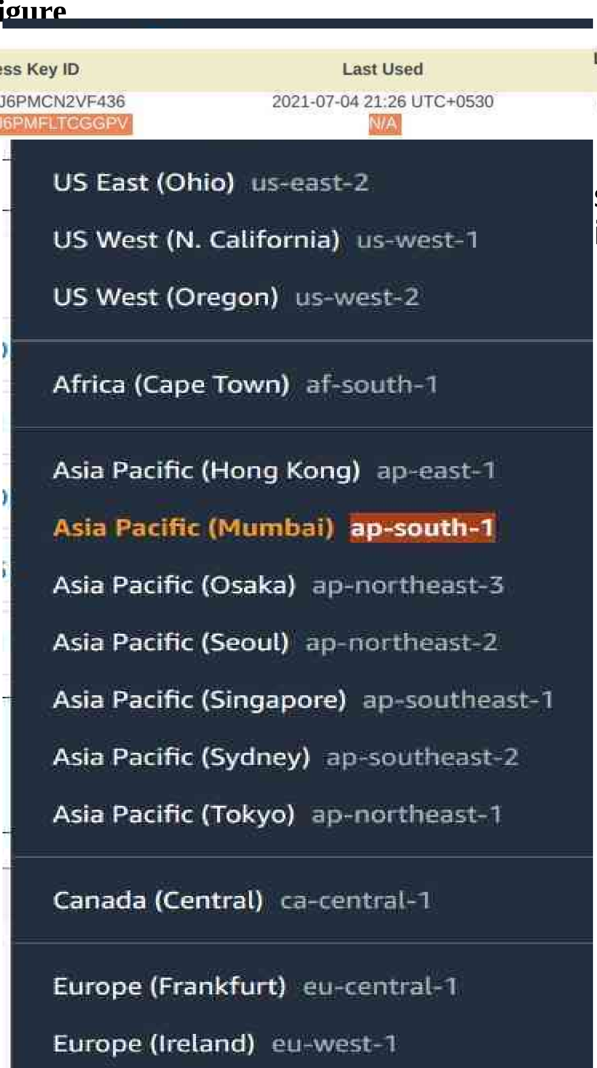
Firstsetupyouraccesskeys,secretkeysandregioncodelocally.

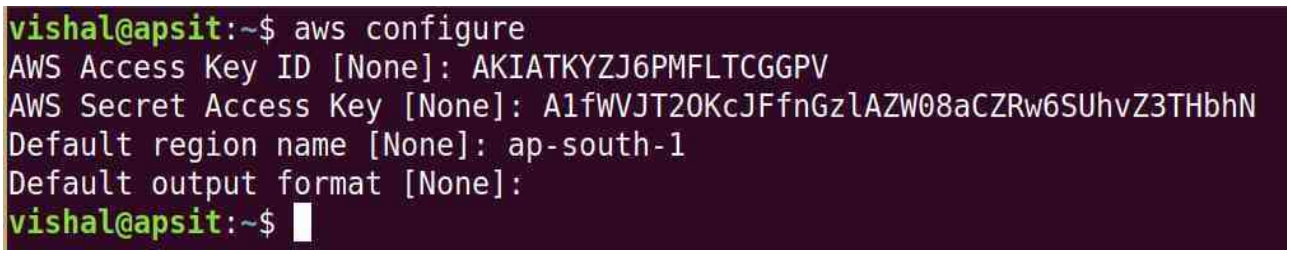
vishal@apsit:~$awsconfigure



Created	Access Key ID	Last Used	Last Used Region	Last Used Service	Status
Jun 4th 2021	AKIATKYZJ6PMCN2VF436	2021-07-04 21:26 UTC+0530	us-east-1	sts	Active
Aug 1st 2021	AKIATKYZJ6PMFLTCGGPV	N/A	N/A	N/A	Active

You can check region as shown in below image :

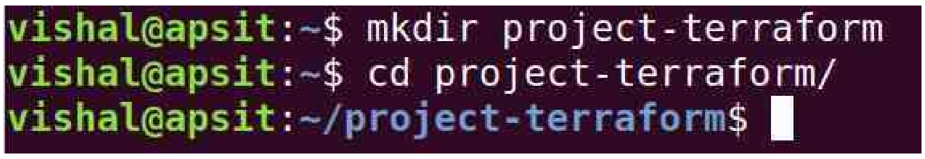

US East (Ohio) us-east-2
US West (N. California) us-west-1
US West (Oregon) us-west-2
Africa (Cape Town) af-south-1
Asia Pacific (Hong Kong) ap-east-1
Asia Pacific (Mumbai) ap-south-1
Asia Pacific (Osaka) ap-northeast-3
Asia Pacific (Seoul) ap-northeast-2
Asia Pacific (Singapore) ap-southeast-1
Asia Pacific (Sydney) ap-southeast-2
Asia Pacific (Tokyo) ap-northeast-1
Canada (Central) ca-central-1
Europe (Frankfurt) eu-central-1
Europe (Ireland) eu-west-1



```
vishal@apsit:~$ aws configure
AWS Access Key ID [None]: AKIATKYZJ6PMFLTCGGPV
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]:
vishal@apsit:~$
```

CreateoneDirectoryforTerraform projectinwhichallfilesofterraformwecansave

vishal@apsit:~$cd~
vishal@apsit:~$mkdirproject-terraform
vishal@apsit:~$ cd project-terraform

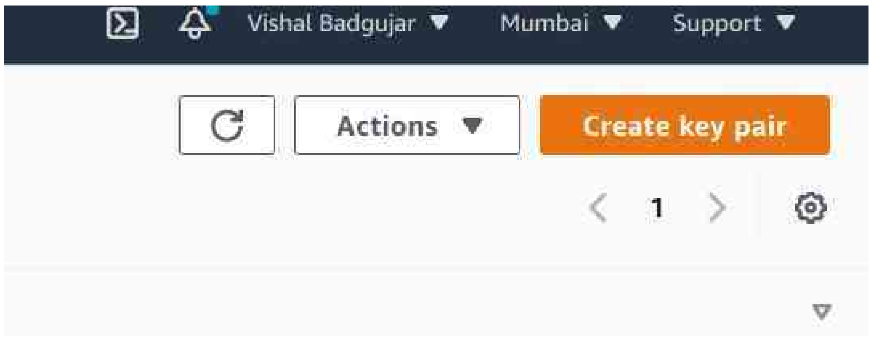


```
vishal@apsit:~$ mkdir project-terraform
vishal@apsit:~$ cd project-terraform/
vishal@apsit:~/project-terraform$
```

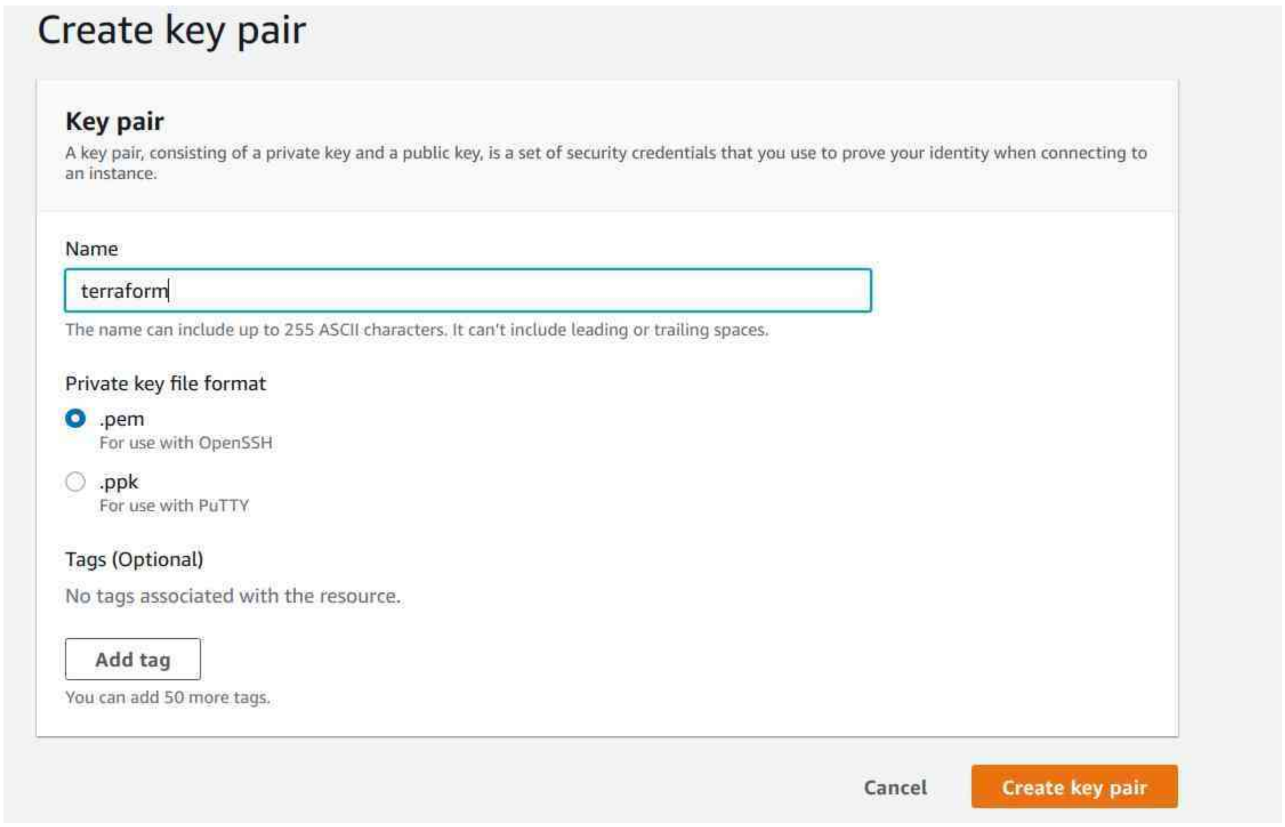
<u>CreateTerraformFiles</u>

<u>vishal@apsit</u>:~$sudonanovariables.tf

Inordertoprovidekeynameinvariablesfirstcreatekeypairasshown:



Givenametokeypairfileas**terraform**



Create key pair

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

terraform

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

Private key file format

[✓ .pem]
For use with OpenSSH

.ppk
For use with PuTTY

Tags (Optional)

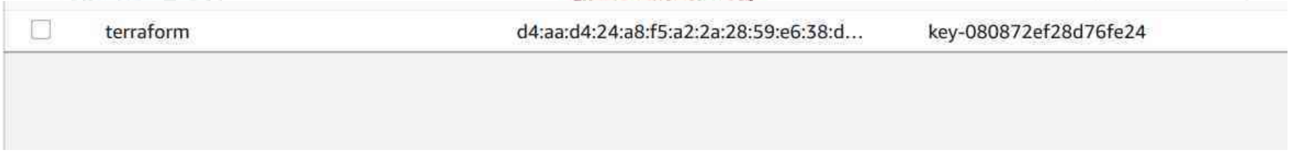
No tags associated with the resource.

Add tag

You can add 50 more tags.

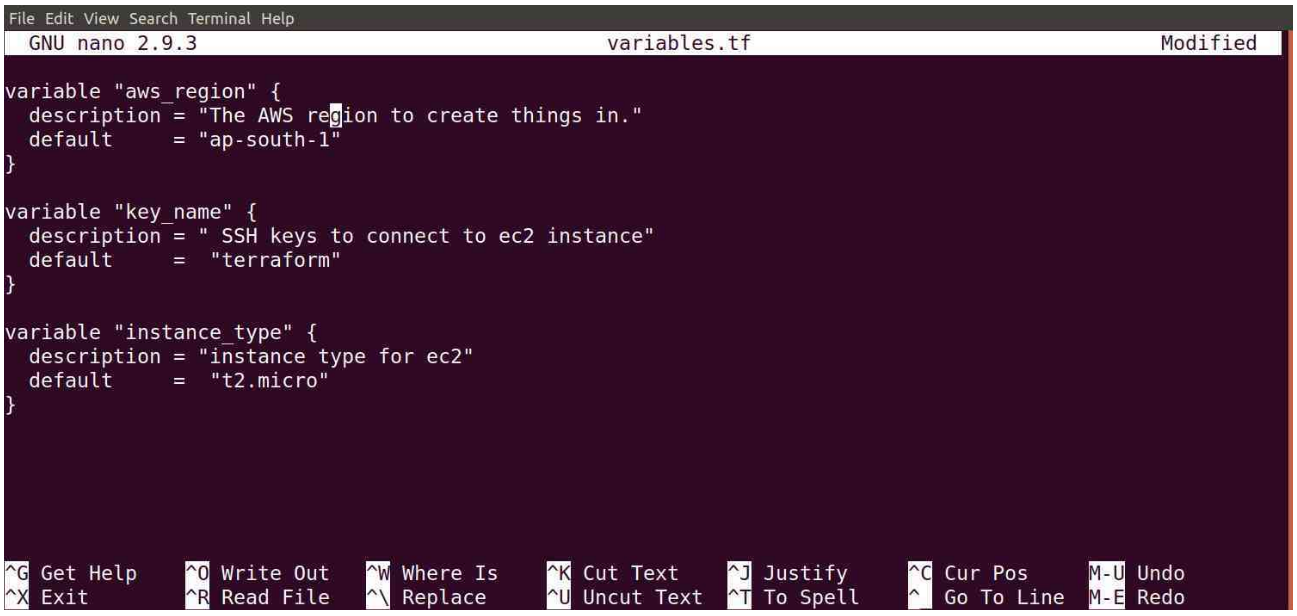
Cancel   Create key pair

Keypairisgenerated



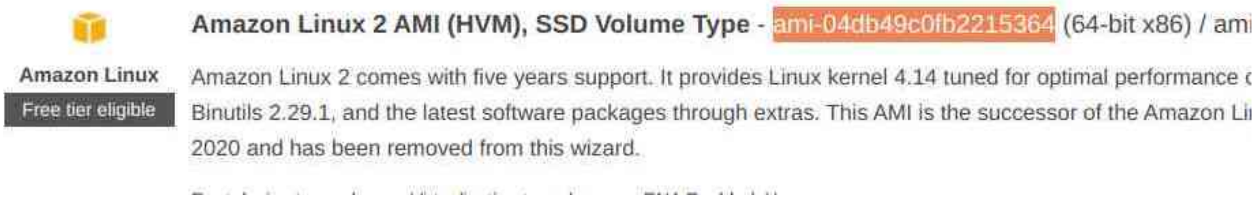
terraform d4:aa:d4:24:a8:f5:a2:2a:28:59:e6:38:d... key-080872ef28d76fe24

UseyourRegionandKeynameinvariable.tf asshownandprovideinstancetypewhichyouwant to create.



```
GNU nano 2.9.3                          variables.tf                           Modified

variable "aws_region" {
  description = "The AWS region to create things in."
  default     = "ap-south-1"
}

variable "key_name" {
  description = " SSH keys to connect to ec2 instance"
  default     =  "terraform"
}

variable "instance_type" {
  description = "instance type for ec2"
  default     =  "t2.micro"
}
```

AftercreatingvariableterraformfilenotedowntheAMIIDofinstancewhichuwanttocreate which we will use to configure our instance in main.tf file.

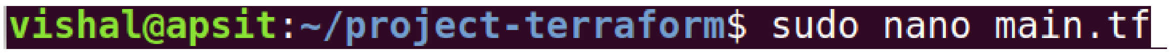


Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-04db49c0fb2215364 (64-bit x86) / ami

Amazon Linux
Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance
Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Li
2020 and has been removed from this wizard.

Nowcreatemain.tffile:



```
vishal@apsit:~/project-terraform$ sudo nano main.tf
```

```
provider"aws"{
  region=var.aws_region
}


#Createsecuritygroupwithfirewallrules
resource"aws_security_group""security_jenkins_port"{name
                   = "security_jenkins_port"
  description="securitygroupforjenkins"


  ingress{
    from_port=8080
    to_port    =
    8080protocol
                   ="tcp"
    cidr_blocks=["0.0.0.0/0"]
  }
```

```
ingress
  { from_port= 22
```

```
    to_port     =
    22protocol
                    ="tcp"
    cidr_blocks=["0.0.0.0/0"]
  }
#outboundfromjenkisserver egress {
    from_port=0
    to_port     = 65535protocol
                     = "tcp"
    cidr_blocks=["0.0.0.0/0"]
  }

  tags={
    Name="security_jenkins_port"
  }
}

resource"aws_instance""myFirstInstance"{ami
                  ="ami-
  0b9064170e32bde34"key_name =
  var.key_name
  instance_type = var.instance_type
  security_groups=["security_jenkins_port"] tags= {
    Name="jenkins_instance"
  }
}
```

#CreateElastic IPaddress

resource"aws_eip""myFirstInstance"{vpc

= true

instance=aws_instance.myFirstInstance.id tags=

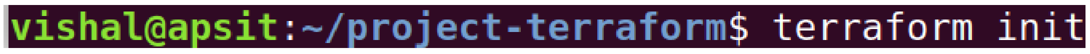
{

Name="jenkins_elstic_ip"

}

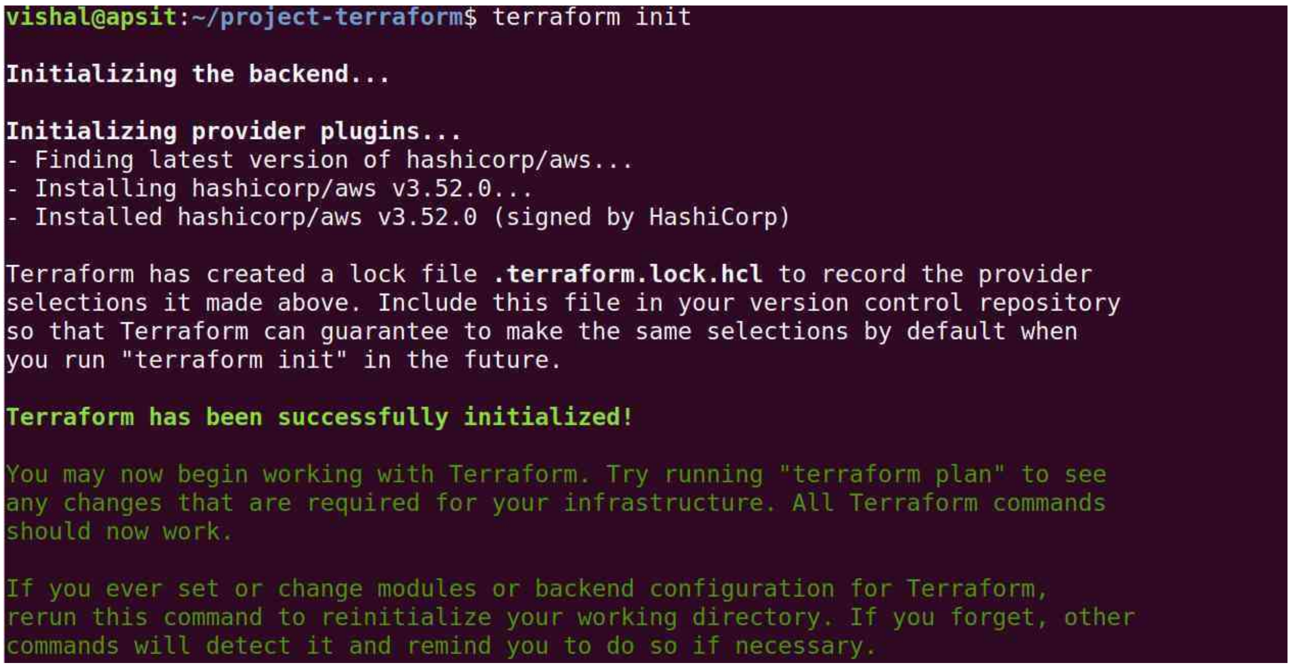
}

PutAMI-IDinabovehighlighted spaceandNowexecutethebelowcommand:



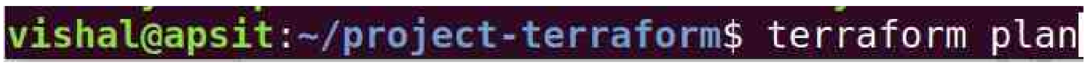
```
vishal@apsit:~/project-terraform$ terraform init
```

youshouldseelikebelowscreenshot.



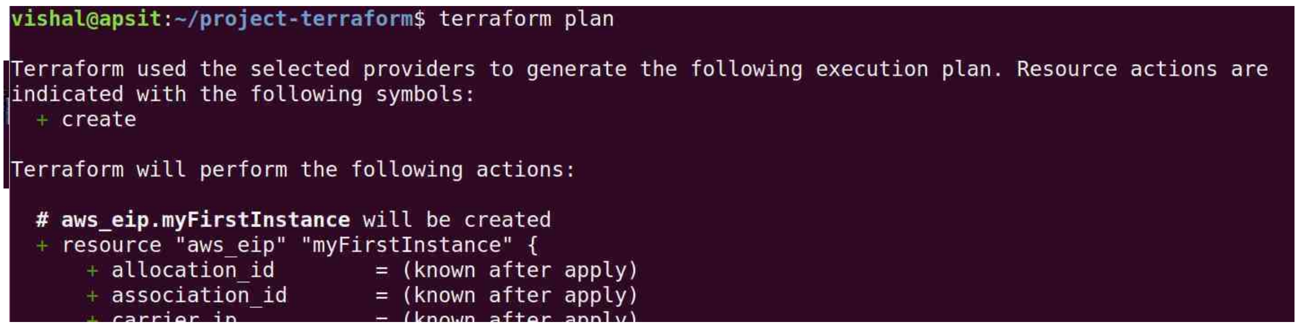
```
vishal@apsit:~/project-terraform$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v3.52.0...
- Installed hashicorp/aws v3.52.0 (signed by HashiCorp)

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

Terraform has been successfully initialized!

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

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Executethebelowcommand



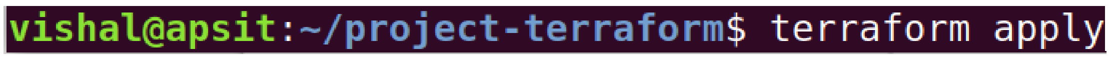
```
vishal@apsit:~/project-terraform$ terraform plan
```

theabovecommandwillshowhowmanyresourceswillbeadded. Plan: 3 to
add, 0 to change, 0 to destroy.



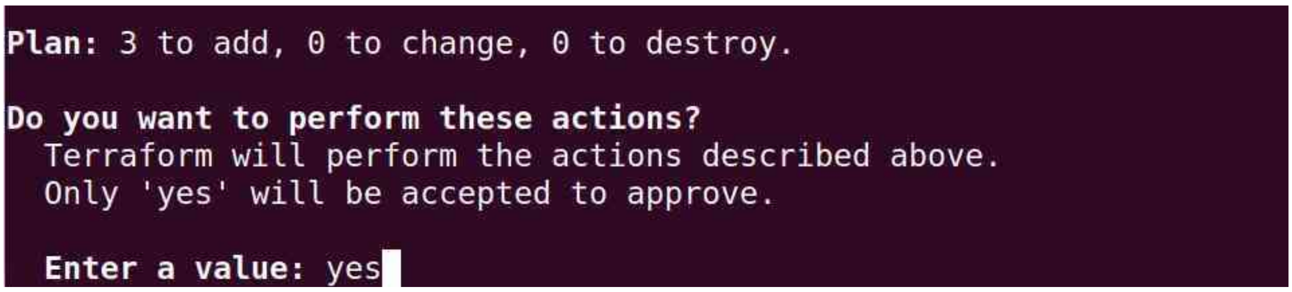
```
vishal@apsit:~/project-terraform$ terraform plan

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_eip.myFirstInstance will be created
  + resource "aws_eip" "myFirstInstance" {
      + allocation_id        = (known after apply)
      + association_id       = (known after apply)
      + carrier_ip           = (known after apply)
```

Executethebelowcommand



```
vishal@apsit:~/project-terraform$ terraform apply
```

ProvidethevalueasYesforapplyingterraform



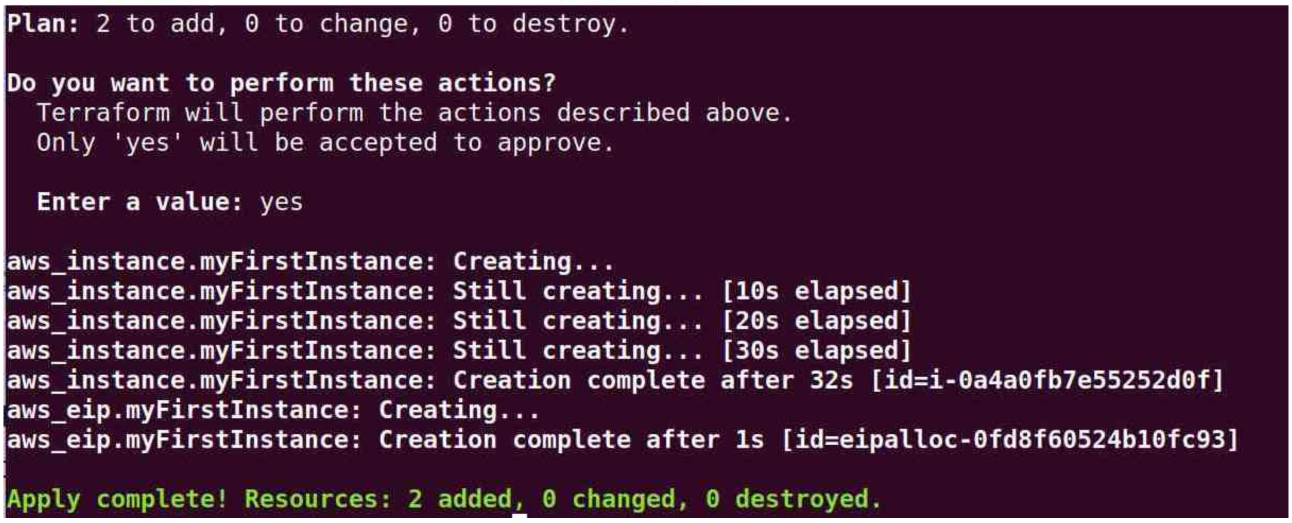
```
Plan: 3 to add, 0 to change, 0 to destroy.

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
```

Plan:3toadd,0tochange,0todestroy. Do you

want to perform these actions?
Terraformwillperformtheactionsdescribedabove. Only 'yes'
will be accepted to approve.

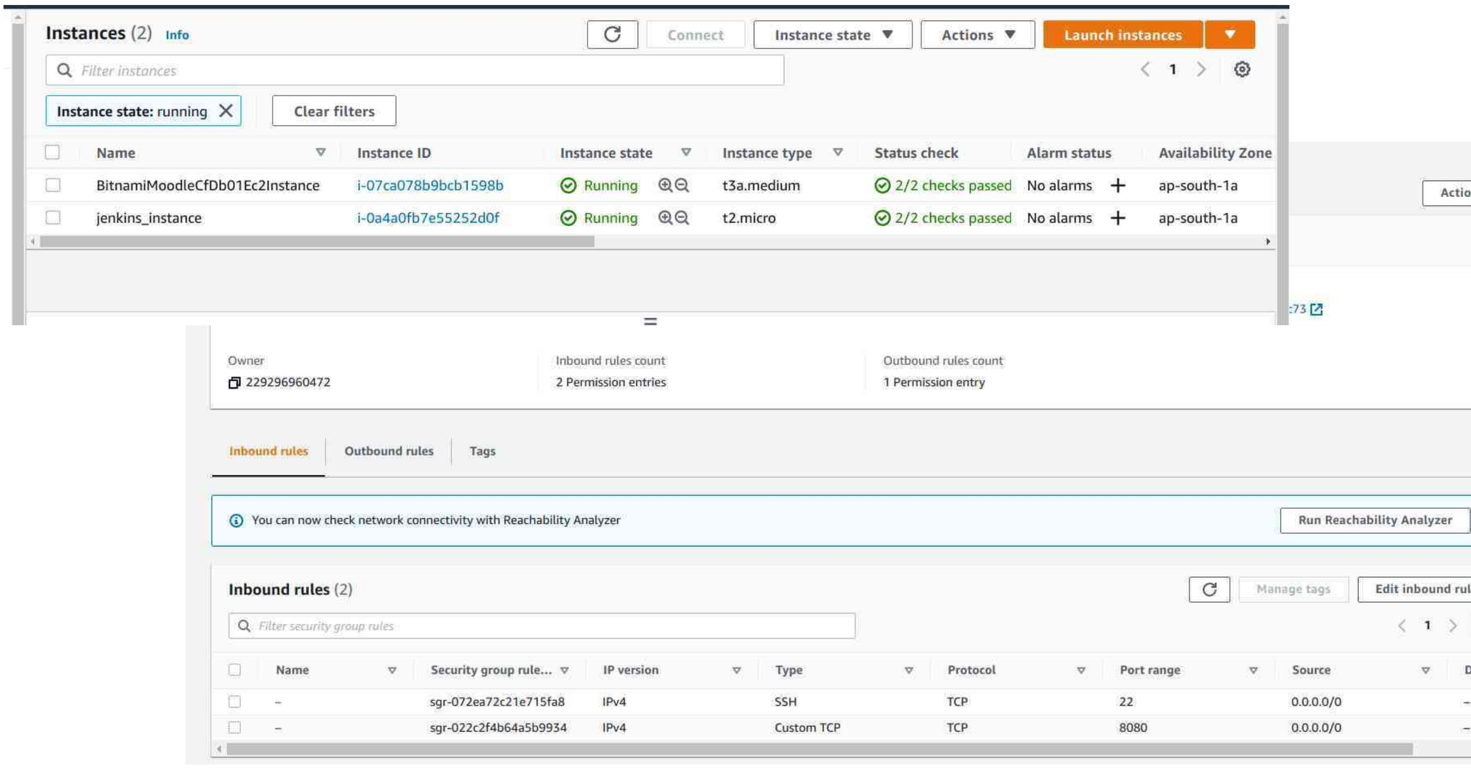
Enteravalue:yes

Applycomplete!Resources:3added,0changed,0destroyed.



```
Plan: 2 to add, 0 to change, 0 to destroy.

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.myFirstInstance: Creating...
aws_instance.myFirstInstance: Still creating... [10s elapsed]
aws_instance.myFirstInstance: Still creating... [20s elapsed]
aws_instance.myFirstInstance: Still creating... [30s elapsed]
aws_instance.myFirstInstance: Creation complete after 32s [id=i-0a4a0fb7e55252d0f]
aws_eip.myFirstInstance: Creating...
aws_eip.myFirstInstance: Creation complete after 1s [id=eipalloc-0fd8f60524b10fc93]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

NowlogintoEC2console,toseethenewinstancesupandrunning,youcanseeJenkins_instanceis up and running which we deploy from terraform.



Instances (2) Info

Instance state: running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☐	BitnamiMoodleCfDb01Ec2Instance	i-07ca078b9bcb1598b	Running	t3a.medium	2/2 checks passed	No alarms	ap-south-1a
☐	jenkins_instance	i-0a4a0fb7e55252d0f	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a

Owner: 229296960472
Inbound rules count: 2 Permission entries
Outbound rules count: 1 Permission entry

You can now check network connectivity with Reachability Analyzer

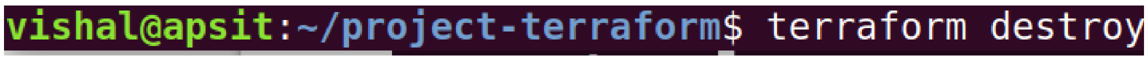
Inbound rules (2)

	Name	Security group rule...	IP version	Type	Protocol	Port range	Source
☐	–	sgr-072ea72c21e715fa8	IPv4	SSH	TCP	22	0.0.0.0/0
☐	–	sgr-022c2f4b64a5b9934	IPv4	Custom TCP	TCP	8080	0.0.0.0/0

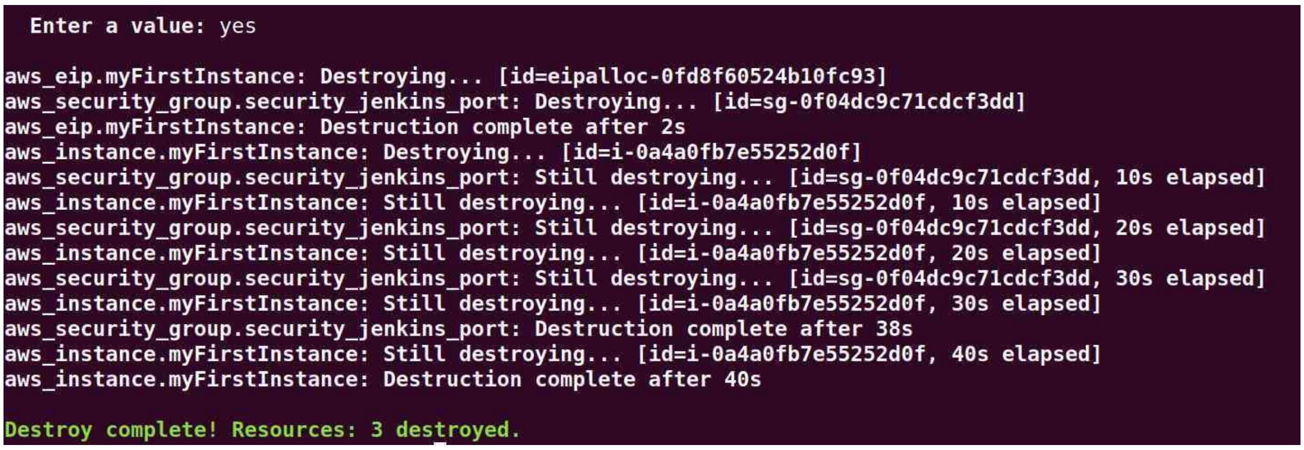
Youcanalsocheckthesecuritygroupr e s o u r c e detailswhichyoucreatedfromterraform:

Terraformdestroy

youcanalsodestroyordelete yourinstancebyusingterraformdestroycommand:

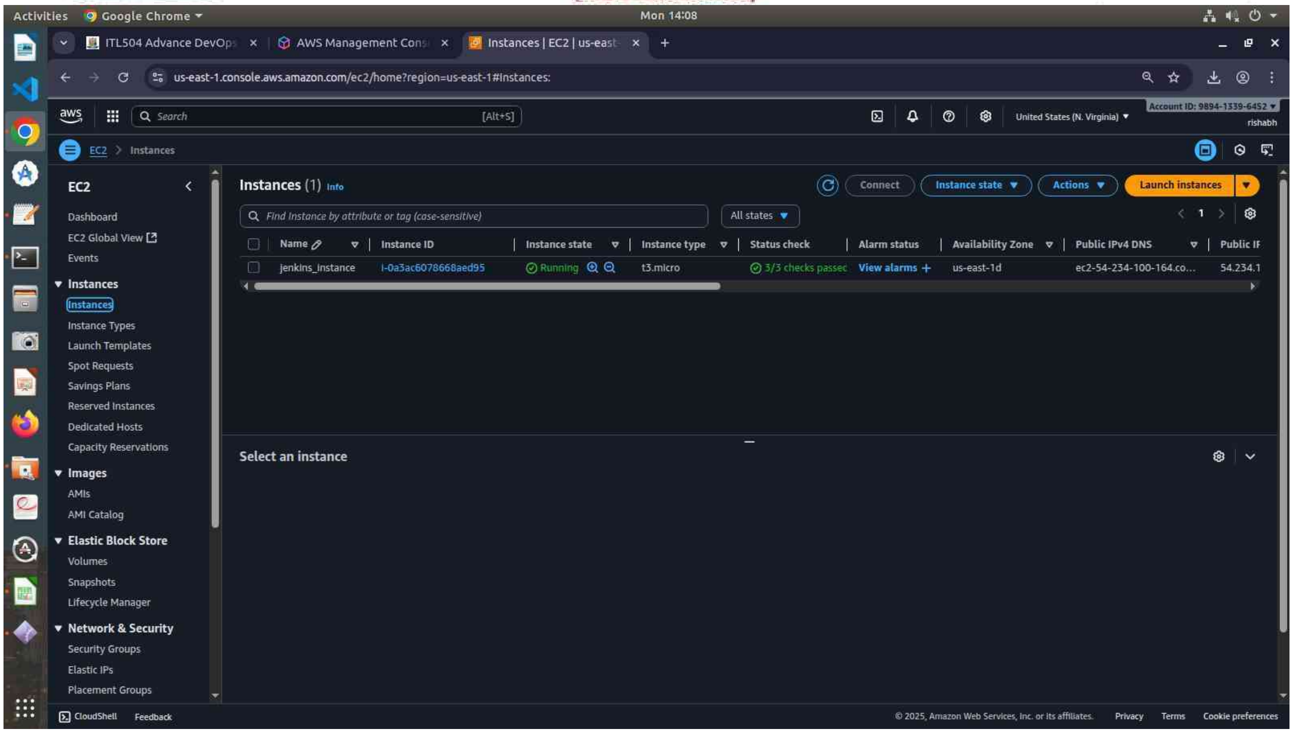


```
vishal@apsit:~/project-terraform$ terraform destroy
```



```
  Enter a value: yes

aws_eip.myFirstInstance: Destroying... [id=eipalloc-0fd8f60524b10fc93]
aws_security_group.security_jenkins_port: Destroying... [id=sg-0f04dc9c71cdcf3dd]
aws_eip.myFirstInstance: Destruction complete after 2s
aws_instance.myFirstInstance: Destroying... [id=i-0a4a0fb7e55252d0f]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-0f04dc9c71cdcf3dd, 10s elapsed]
aws_instance.myFirstInstance: Still destroying... [id=i-0a4a0fb7e55252d0f, 10s elapsed]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-0f04dc9c71cdcf3dd, 20s elapsed]
aws_instance.myFirstInstance: Still destroying... [id=i-0a4a0fb7e55252d0f, 20s elapsed]
aws_security_group.security_jenkins_port: Still destroying... [id=sg-0f04dc9c71cdcf3dd, 30s elapsed]
aws_instance.myFirstInstance: Still destroying... [id=i-0a4a0fb7e55252d0f, 30s elapsed]
aws_security_group.security_jenkins_port: Destruction complete after 38s
aws_instance.myFirstInstance: Still destroying... [id=i-0a4a0fb7e55252d0f, 40s elapsed]
aws_instance.myFirstInstance: Destruction complete after 40s

Destroy complete! Resources: 3 destroyed.
```

Nowyoucanseeinstancewhichyoucreatedbyusingterraformisdeletedsuccessfullyfromaws console also you can check it will removed successfully:



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IF
jenkins_instance	i-0a3ac6078668aed95	Running	t3.micro	3/3 checks passec	View alarms +	us-east-1d	ec2-54-234-100-164.co...	54.234.1

AlltheResourcesincludingSecuritygroups,EC2instancesusingterraformwillbedeleted.Inthis way we can automate infrastructure set up using terrform in aws cloud.

Conclusion:herewelearnedtocreateaterrforminstance