



## ASSIGNMENT – 3

COURSE : DEVOPS

Trainer : Mr . MADHUKAR

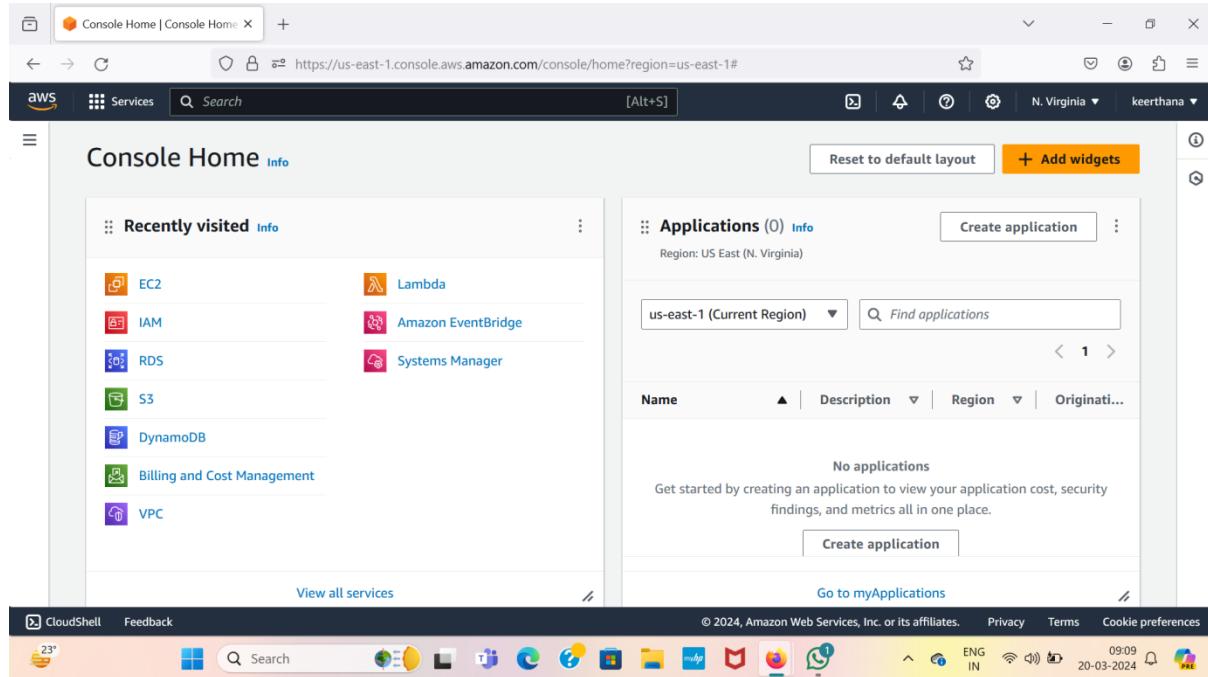
NAME:A GNANESHWAR

Mail id : gnaneshwar502@gmail.com

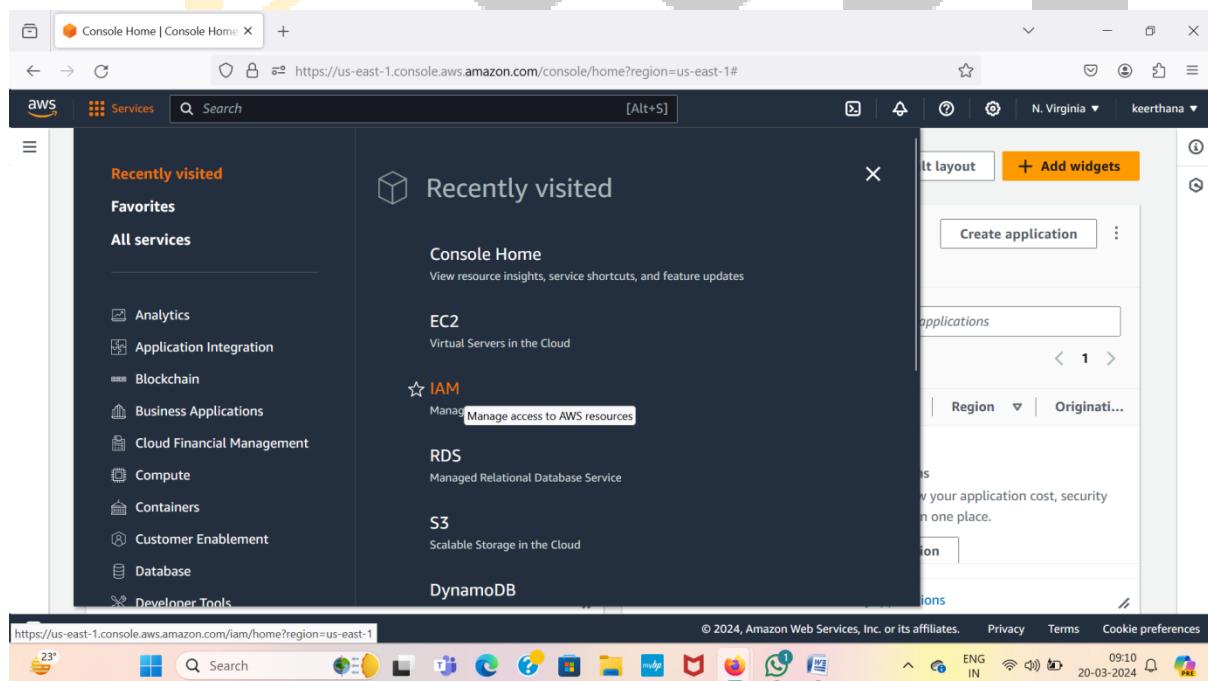


# 1 . By using Terraform create an S3 Bucket and upload the file to the S3 bucket ?

- Log into AWS Account
- After login go to console home page



- No go to IAM then create a user and add "S3" permissions to created user



- Now click on users
- Click on create user then enter user name and click on check box of provide user access
- Click on I want create user then we can choose automated password either custom password
- Uncheck the user must create a new password then click on next

The screenshot shows the AWS IAM Dashboard. A green notification bar at the top says "User "name" deleted." Below it, the main interface displays "Security recommendations" with two items: "Add MFA for root user" (with a yellow warning icon) and "Root user has no active access keys" (with a green checkmark icon). Under "IAM resources", there are tabs for User groups, Users, Roles, Policies, and Identity providers. The "Users" tab is selected, showing a table with one row: "No resources to display". On the left sidebar, the "Users" option under "Access management" is highlighted. The browser address bar shows the URL: https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/home.

This screenshot shows the "Users" page within the IAM service. At the top right, there is a prominent orange "Create user" button. The main area displays a table with columns: "User name", "Path", "Group:", "Last activity", "MFA", and "Pass". A search bar is located above the table. The browser address bar shows the URL: https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users.

User name  
s3bucketuser

Provide user access to the AWS Management Console - optional  
If you're providing console access to a person, it's a [best practice](#) to manage their access in IAM Identity Center.

**Are you providing console access to a person?**

Specify a user in Identity Center - Recommended  
We recommend that you use Identity Center to provide console access to a person. With Identity Center, you can centrally manage user access to their AWS accounts and cloud applications.

I want to create an IAM user  
We recommend that you create IAM users only if you need to enable programmatic access through access keys, service-specific credentials for AWS CodeCommit or Amazon Keypairs, or a backup credential for emergency account access.

Console password

Autogenerated password  
You can view the password after you create the user.

Custom password  
Enter a custom password for the user.

Show password

Users must create a new password at next sign-in - Recommended  
Users automatically get the [IAMUserChangePassword](#) policy to allow them to change their own password.

**If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keypairs, you can generate them after you create this IAM user. [Learn more](#)**

Cancel **Next**

- Click on Attach policy then select s3 full access permission policy then click on next
- Click on Create a user.

Screenshot of the AWS IAM 'Create user' wizard, Step 2: Set permissions.

The 'Permissions options' section shows three choices:

- Add user to group: Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- Copy permissions: Copy all group memberships, attached managed policies, and inline policies from an existing user.
- Attach policies directly: Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

The 'Permissions policies' section lists 1181 policies, with a search bar and a 'Create policy' button.

Filter by Type: s3

Policy name	Type	Attached entities
AmazonDMSRedshiftS...	AWS managed	0
AmazonS3FullAccess	AWS managed	0
AmazonS3ObjectLamb...	AWS managed	0
AmazonS3OutpostsFul...	AWS managed	0

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

Create user | IAM | Global

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/create

aws Services Search [Alt+S]

keerthana

<input type="checkbox"/>	<a href="#">AmazonS3OutpostsFullAccess</a>	AWS managed	0
<input type="checkbox"/>	<a href="#">AmazonS3OutpostsReadOnlyAccess</a>	AWS managed	0
<input type="checkbox"/>	<a href="#">AmazonS3ReadOnlyAccess</a>	AWS managed	0
<input type="checkbox"/>	<a href="#">AWSBackupServiceRole</a>	AWS managed	0
<input type="checkbox"/>	<a href="#">AWSBackupServiceRoleForAWSBackup</a>	AWS managed	0
<input type="checkbox"/>	<a href="#">AWSS3OnOutpostsServerlessExecutionRole</a>	AWS managed	0
<input type="checkbox"/>	<a href="#">LambdaRecordToS3</a>	AWS managed	0
<input type="checkbox"/>	<a href="#">QuickSightAccessForS3</a>	AWS managed	0
<input type="checkbox"/>	<a href="#">S3StorageLensServiceRole</a>	AWS managed	0

▶ Set permissions boundary - optional

Cancel Previous Next

Create user | IAM | Global

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/create

aws Services Search [Alt+S]

keerthana

Step 4  
Retrieve password

Permissions summary

Name	Type	Used as
<a href="#">AmazonS3FullAccess</a>	AWS managed	Permissions policy

Tags - optional

Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Previous Create user



The screenshot shows the AWS IAM 'Create user' process at Step 4: Retrieve password. A success message 'User created successfully' is displayed. The 'Console sign-in details' section shows the sign-in URL (<https://339712815762.signin.aws.amazon.com/console>), user name ('s3bucketuser'), and console password ('\*\*\*\*\*'). Buttons for 'Cancel', 'Download .csv file', and 'Return to users list' are visible.

- Now click on created user
- Create a access key for user
- Click on create user and select CLI (Command Line Interface) then click on next enter description then create a access key

The screenshot shows the AWS IAM 'Users' page. The left sidebar shows 'Identity and Access Management (IAM)' with 'Access management' expanded, showing 'Users'. The main area displays a table of users with one entry: 's3bucketuser'. The table includes columns for User name, Path, Group, Last activity, MFA, and Pass. Buttons for 'Search', 'Delete', and 'Create user' are present.

s3bucketuser | IAM | Global

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/details/s3bucketuser?section=open

aws Services Search [Alt+S] Global keerthana

**Identity and Access Management (IAM)**

Search IAM

Dashboard

Access management

User groups

**Users**

Roles

Policies

Identity providers

Account settings

Access reports

Access Analyzer

CloudShell Feedback

ARN: arn:aws:iam::339712815762:user/s3bucketuser

Console access: Enabled without MFA

Access key 1: Create access key

Created: March 20, 2024, 09:12 (UTC+05:30)

Last console sign-in: Never

Permissions Groups Tags Security credentials Access Advisor

Permissions policies (1): Add permissions

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type

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

24° 09:12 20-03-2024

Create access key | IAM | Global

https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/users/details/s3bucketuser/create-access-key

aws Services Search [Alt+S] Global keerthana

Step 1: Access key best practices & alternatives

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Step 2 - optional: Set description tag

Step 3: Retrieve access keys

Use case

Command Line Interface (CLI)  
You plan to use this access key to enable the AWS CLI to access your AWS account.

Local code  
You plan to use this access key to enable application code in a local development environment to access your AWS account.

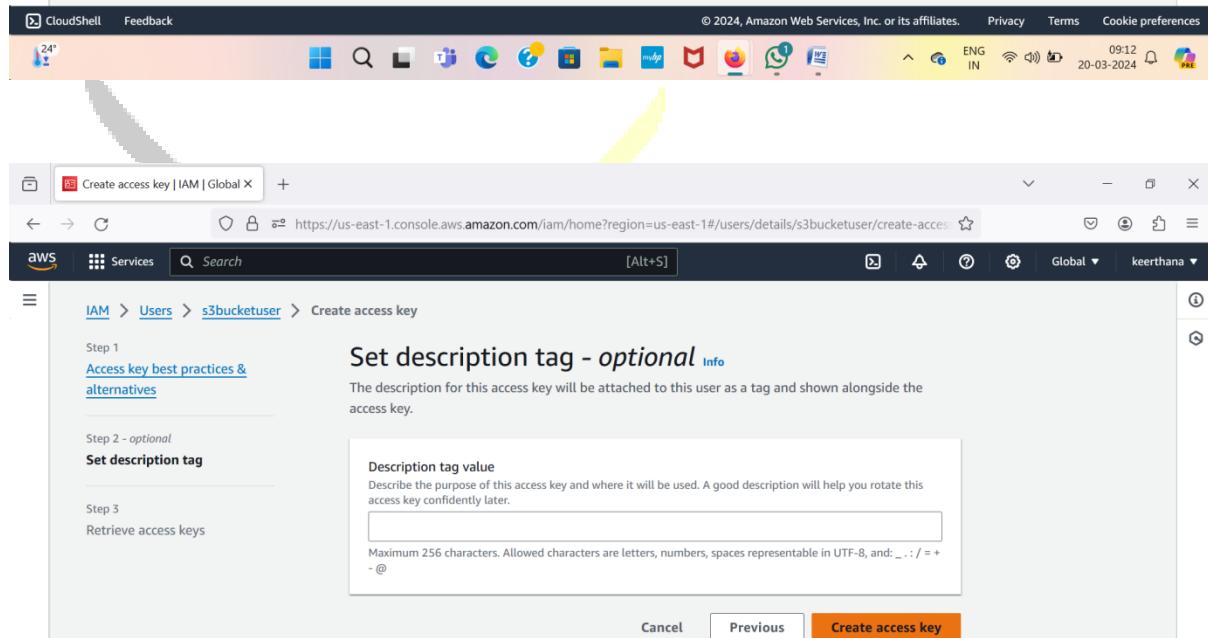
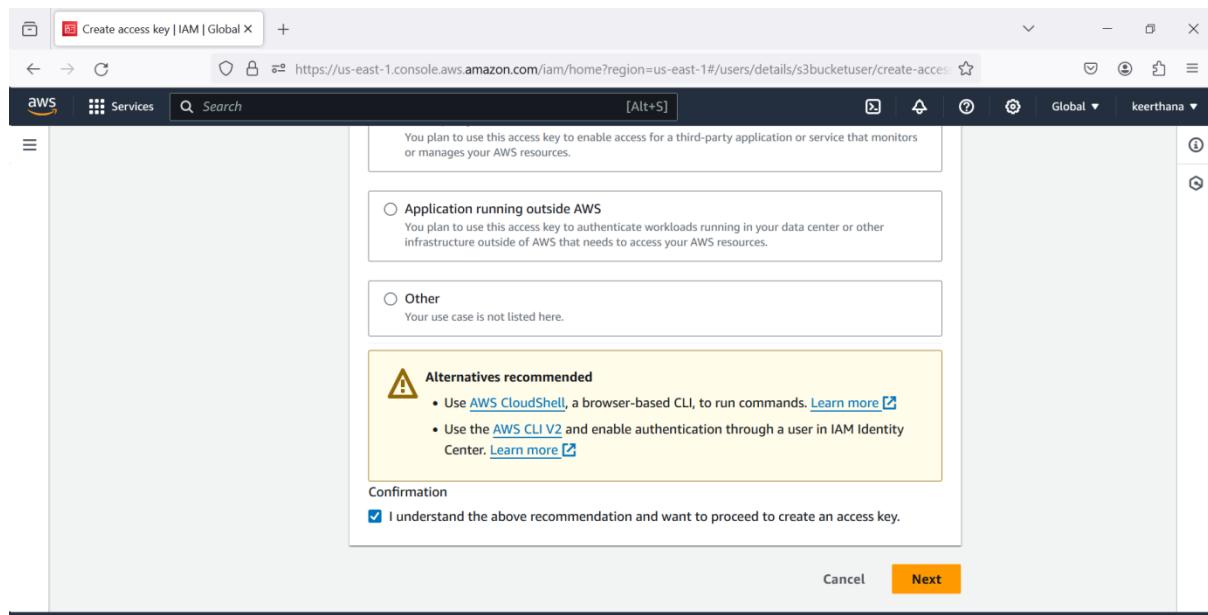
Application running on an AWS compute service  
You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.

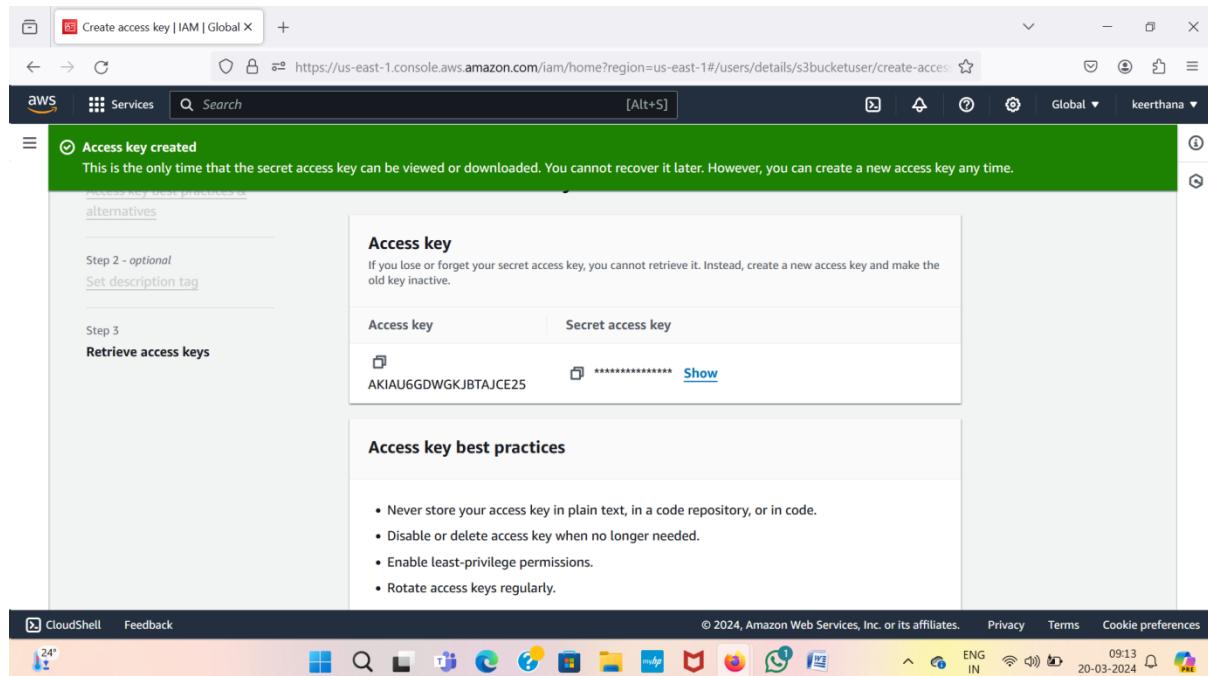
Third-party service  
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

CloudShell Feedback

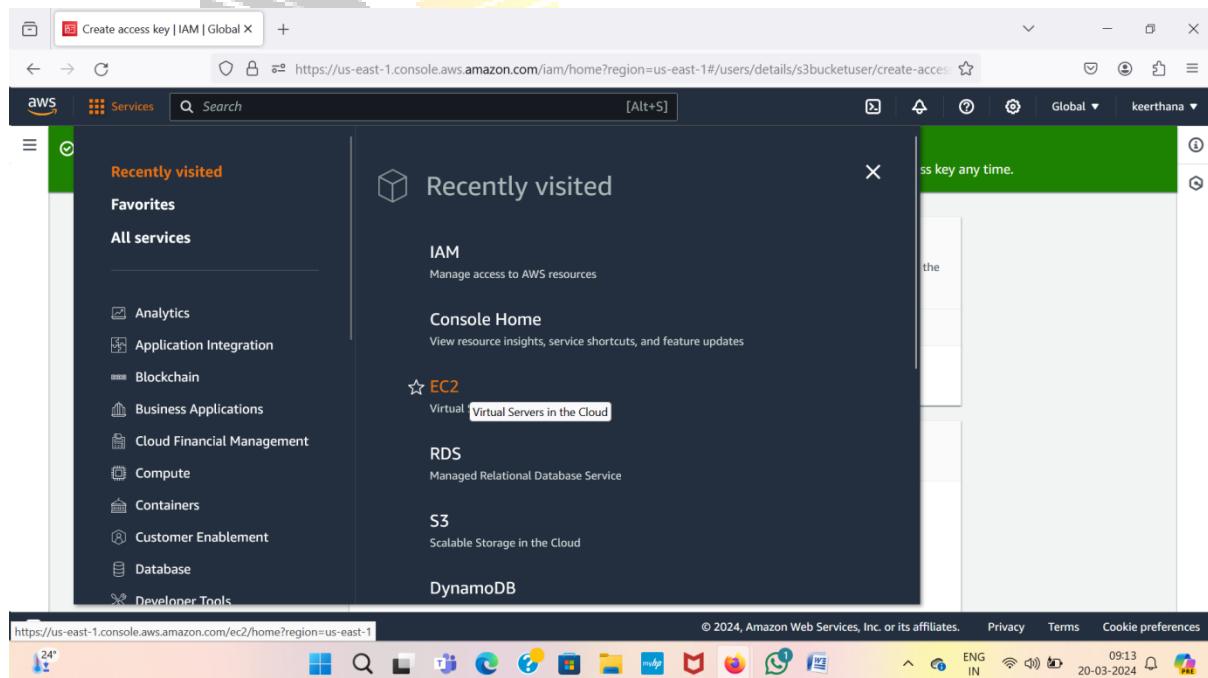
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

24° 09:12 20-03-2024





- Now go to EC2
  - Create a ec2 instance in ubuntu then connect the ec2 instance



Screenshot of the AWS EC2 Instances page and the Launch Instances wizard.

**EC2 Instances Page:**

- Region: us-east-1
- Instances: No instances
- Actions: Launch instances

**Launch Instances Wizard - Step 1: Name and tags:**

- Name: ec2-S3
- Number of instances: 1
- Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ami-080e1f13689e07408
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

**Launch Instances Wizard - Step 2: Quick Start:**

- Quick Start options: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE, Browse more AMIs
- Buttons: Cancel, Launch instance, Review commands

Screenshot of the AWS EC2 Launch Instances wizard - Step 1: Set instance details.

**Instance type:** t2.micro (Free tier eligible)

**Key pair (login):** Select (Create new key pair)

**Network settings:** Edit

**Summary:**

- Number of instances: 1
- Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ami-080e1f13689e07408
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

**Buttons:** Cancel, Launch instance, Review commands

Screenshot of the AWS EC2 Launch Instances wizard - Step 2: Create key pair.

**Create key pair:**

**Key pair name:** s3

**Key pair type:**  
 RSA (RSA encrypted private and public key pair)  
 ED25519 (ED25519 encrypted private and public key pair)

**Private key file format:**  
 .pem (For use with OpenSSH)  
 .ppk (For use with PuTTY)

**Buttons:** Cancel, Create key pair, Launch Instance, Review commands

**Key pair (login)** Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

s3

[Create new key pair](#)

**Network settings** [Info](#)

Network [Info](#)  
vpc-01656471fb3fca66f

Subnet [Info](#)  
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

**Summary**

Number of instances [Info](#)  
1

Software Image (AMI)  
Canonical, Ubuntu, 22.04 LTS, ...[read more](#)  
ami-080e1f13689e07408

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GiB

[Cancel](#) [Launch instance](#) [Review commands](#)

**Success**  
Successfully initiated launch of instance (i-048ad59bb29cdbd76)

[Launch log](#)

**Next Steps**

What would you like to do next with this instance, for example "create alarm" or "create backup"

Create billing and free tier usage alerts  
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage

Connect to your instance  
Once your instance is running, log into it from your local computer.  
[Connect to instance](#)

Connect an RDS database  
Configure the connection between an EC2 instance and a database to allow traffic flow between them.  
[Connect to RDS database](#)

Create EBS snapshot policy  
Create a policy that automates the creation, retention, and deletion of EBS snapshots  
[Create EBS snapshot policy](#)

[CloudShell](#) [Feedback](#) © 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#) 09:16 20-03-2024

Screenshot of the AWS EC2 Instances page showing a single running instance named "ec2-S3".

The instance details are as follows:

Attribute	Value
Instance ID	i-048ad59bb29cdbd76
Instance state	Running
Instance type	t2.micro
Status check	Initializing

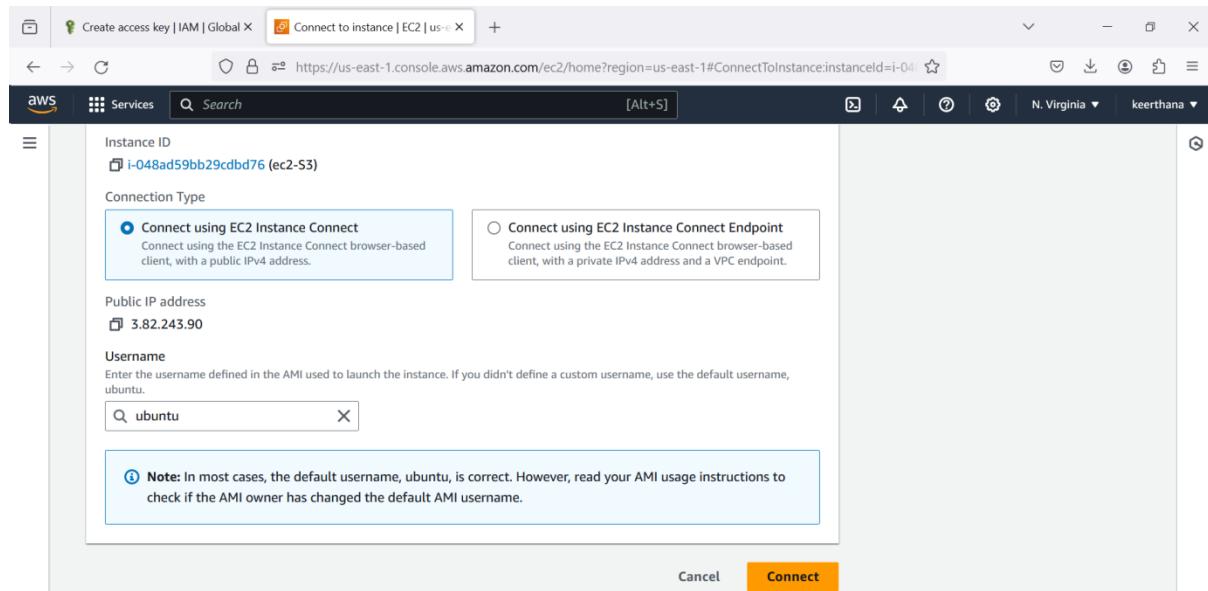
The screenshot also shows the Windows taskbar at the bottom.

Screenshot of the AWS EC2 Instance Details page for the instance i-048ad59bb29cdbd76 (ec2-S3).

The instance summary table contains the following information:

Attribute	Value
Instance ID	i-048ad59bb29cdbd76 (ec2-S3)
IPV6 address	-
Hostname type	IP name: ip-172-31-84-72.ec2.internal
Answer private resource DNS name IPv4 (A)	ip-172-31-84-72.ec2.internal
Auto-assigned IP address	3.82.243.90 [Public IP]
Public IPv4 address	3.82.243.90 [open address]
Private IP4 addresses	172.31.84.72
Public IPv4 DNS	ec2-3-82-243-90.compute-1.amazonaws.com [open address]
Instance type	t2.micro
VPC ID	vpc-01656471fb3fca66f
Elastic IP addresses	-
AWS Compute Optimizer finding	Opt-in to AWS Compute Optimizer for recommendations.

The screenshot also shows the Windows taskbar at the bottom.



```
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
24°
Create access key | IAM | Global X Instance details | EC2 | us-east-1 X EC2 Instance Connect | us-east-1 X + 09:16 20-03-2024 ENG IN

aws Services Search [Alt+S] N. Virginia keerthana

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

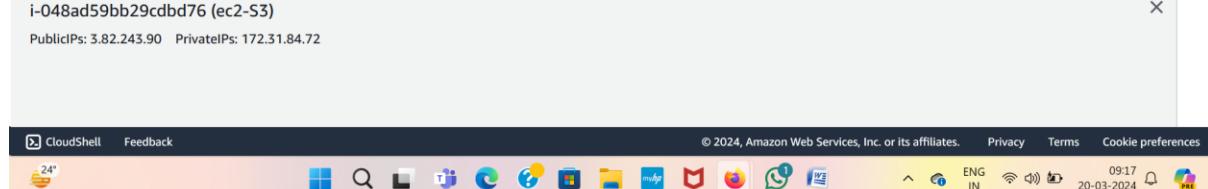
The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-84-72:~$ sudo -i
root@ip-172-31-84-72:~#
```



- Now install awscli in ec2 instance commands is

Apt update -y

Apt install awscli -y

```
root@ip-172-31-84-72:~# apt update -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease [119 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [109 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1502 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [289 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1619 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [271 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1058 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [239 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.1 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [42.1 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [10.1 kB]

i-048ad59bb29cdbc76 (ec2-S3)
Public IPs: 3.82.243.90 Private IPs: 172.31.84.72
```



```
root@ip-172-31-84-72:~# apt install awscli -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
bzip2 docutils-common fontconfig fontconfig-config fonts-droid-fallback fonts-noto-monospace fonts-urw-base35 ghostscript groff gsfonts
hicolor-icon-theme imagemagick-6-common imagemagick-6.q16 libbam3 libavahi-client3 libavahi-common-data libavahi-common3
libcairo2 libcurl5 libdatriel libdav1d5 libde265-0 libdeflate0 libdjvuibre-text libdjvuibre21 libfftw3-double3 libfontconfig1 libgomp1
libgraphite2-3 libgs9 libgs9-common libharfbuzz0b libheif1 libice6 libidn12 libijs-0.35 libimbase25 libimagequant0 libjbig0 libjbig2dec0
libjpeg-turbo8 libjpeg8 libjxr-tools libjxr0 liblcms2-2 liblqr-1 libltdl7 libmagickcore-6.q16-6 libmagickcore-6.q16-6-extra
libmagickwand-6.q16-6 libnetpbm10 libopenexr25 libopenjp2-7 libpango-1.0-0 libpangocairo-1.0-0 libpangoft2-1.0-0 libpaper-utils libpaper1
libpixman-1-0 libraqm0 libsm6 libthai-data libthai0 libtiff5 libwebp7 libwebpdemux2 libwebrtcvpx3 libwmflite-0.2-7 libx265-199 libxaw7
libxcb-render0 libxcb-shm0 libxmu6 libxpm4 libxrender1 libxt6 mailcap mime-support netpbm poppler-data psutils python3-botocore
python3-dateutil python3-docutils python3-jmespath python3-olefile python3-pil python3-pygments python3-roman python3-rsa
python3-s3transfer sgml-base x11-common xml-core
Suggested packages:
bzip2-doc fonts-noto fonts-freefont-otf | fonts-freefont-ttf fonts-texgyre ghostscript-x imagemagick-doc autotrace cups-bsd | lpr | lprng
enscript ffmpeg gimp gnuplot grads graphviz hp2xx html2ps libwmf-bin mplayer povray radiance sane-utils texlive-base-bin transfig
ufraw-batch xdg-utils cups-common libfftw3-bin libfftw3-dev liblcms2-utils inkscape poppler-utils fonts-japanese-mincho
| fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic fonts-aphic-ukai fonts-aphic-uming fonts-nanum docutils-doc
```



- After Installing awscli then configure the user to awscli

Aws configure

Access key = ""

Secret access key = ""

Default region = ""

Output format = ""

```
root@ip-172-31-84-72:~# aws configure
AWS Access Key ID [None]: 
AWS Secret Access Key [None]: 
Default region name [None]: us-east-2
Default output format [None]: table
root@ip-172-31-84-72:~# 
```

i-048ad59bb29cdbd76 (ec2-S3)  
PublicIPs: 3.82.243.90 PrivateIPs: 172.31.84.72

- Now install Terraform
- Copy the terraform install command then paste into the server

The screenshot shows the Terraform installation page on the HashiCorp developer site. The left sidebar lists operating systems: macOS, Windows, Linux (selected), FreeBSD, OpenBSD, and Solaris. The main content area is titled "Linux" and contains sections for "Package manager" and "Binary download". The "Package manager" section shows a terminal command:

```
$ wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
$ echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
sudo apt update && sudo apt install terraform
```

The "Binary download" section provides links for "386" (Version: 1.7.5) and "AMD64" (Version: 1.7.5). To the right, there's an "About Terraform" summary and a "Featured docs" sidebar with links to Introduction to Terraform, Configuration Language, Terraform CLI, Terraform Cloud, and Provider Use.

The screenshot shows a terminal session on an Amazon EC2 instance (root@ip-172-31-84-72:~#). The user runs the following commands to install Terraform:

```
root@ip-172-31-84-72:~# wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
root@ip-172-31-84-72:~# echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
root@ip-172-31-84-72:~# sudo apt update && sudo apt install terraform
```

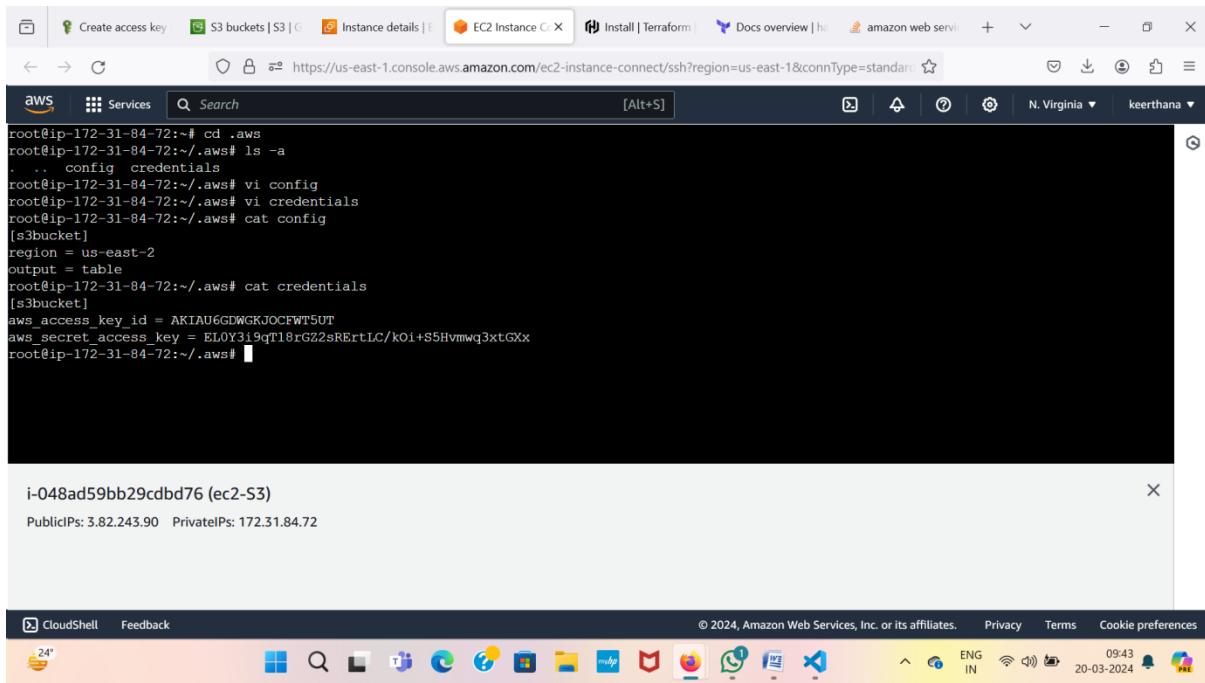
The terminal shows the progress of the wget command and the output of the apt update and apt install commands. The user also checks the installed version of Terraform:

```
root@ip-172-31-84-72:~# terraform -version
Terraform v1.7.5
```

At the bottom, it shows the CloudShell interface with "Public IPs: 3.82.243.90" and "Private IPs: 172.31.84.72".



- Now we can change the default names in credentials and config files



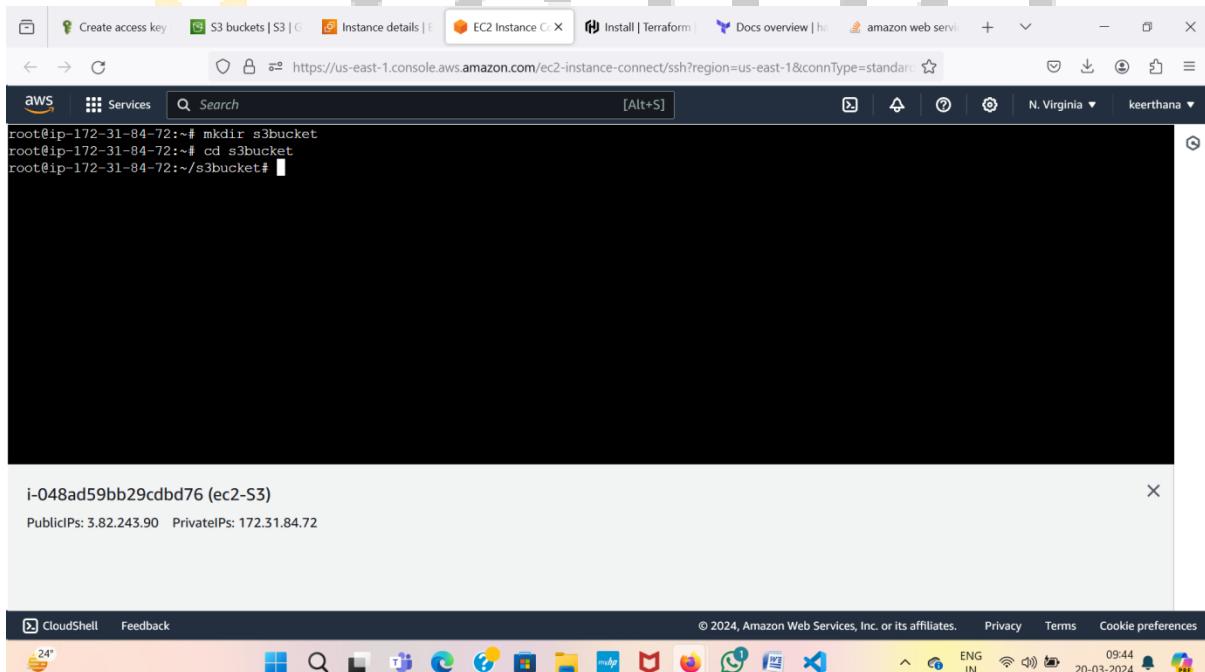
The screenshot shows a CloudShell session with the following terminal output:

```
root@ip-172-31-84-72:~# cd .aws
root@ip-172-31-84-72:~/ .aws# ls -a
.  .. config credentials
root@ip-172-31-84-72:~/ .aws# vi config
root@ip-172-31-84-72:~/ .aws# vi credentials
root@ip-172-31-84-72:~/ .aws# cat config
[s3bucket]
region = us-east-2
output = table
root@ip-172-31-84-72:~/ .aws# cat credentials
[s3bucket]
aws_access_key_id = AKIAU6GDWKGJOCFWT5UT
aws_secret_access_key = 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
root@ip-172-31-84-72:~/ .aws#
```

Below the terminal, the instance details are shown:

i-048ad59bb29cdbd76 (ec2-S3)  
Public IPs: 3.82.243.90 Private IPs: 172.31.84.72

- Now create one directory in that directory write the script of create a s3 bucket
- Create provider block and resource block



The screenshot shows a CloudShell session with the following terminal output:

```
root@ip-172-31-84-72:~# mkdir s3bucket
root@ip-172-31-84-72:~# cd s3bucket
root@ip-172-31-84-72:~/s3bucket#
```

Below the terminal, the instance details are shown:

i-048ad59bb29cdbd76 (ec2-S3)  
Public IPs: 3.82.243.90 Private IPs: 172.31.84.72

```
root@ip-172-31-84-72:~/s3bucket# vi provider.tf
root@ip-172-31-84-72:~/s3bucket# cat provider.tf
provider "aws" {
  profile = "s3bucket"
  region = "ap-south-1"
}
root@ip-172-31-84-72:~/s3bucket#
```

i-048ad59bb29cdbd76 (ec2-S3)

PublicIPs: 3.82.243.90 PrivateIPs: 172.31.84.72

```
root@ip-172-31-84-72:~/s3bucket# vi provider.tf
root@ip-172-31-84-72:~/s3bucket# cat provider.tf
provider "aws" {
  profile = "s3bucket"
  region = "ap-south-1"
}
root@ip-172-31-84-72:~/s3bucket# vi resource.tf
root@ip-172-31-84-72:~/s3bucket# cat resource.tf
resource "aws_s3_bucket" "bucket502" {
  bucket = "bucket145021"
  acl   = "private"
  tags = {
    name = "bucket502"
    environment = "dev"
  }
}
root@ip-172-31-84-72:~/s3bucket#
```

i-048ad59bb29cdbd76 (ec2-S3)

PublicIPs: 3.82.243.90 PrivateIPs: 172.31.84.72

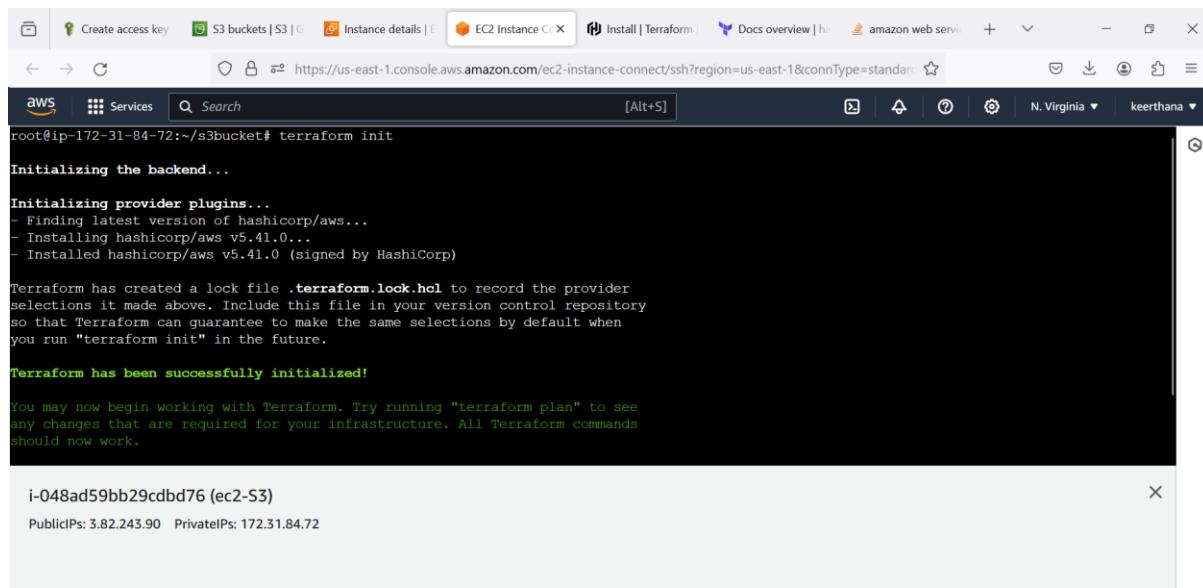


- After writing the script then execute the code in terraform commands are

Terraform init

Terraform plan

Terraform apply



```
root@ip-172-31-84-72:~/s3bucket# terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.41.0...
- Installed hashicorp/aws v5.41.0 (signed by HashiCorp)

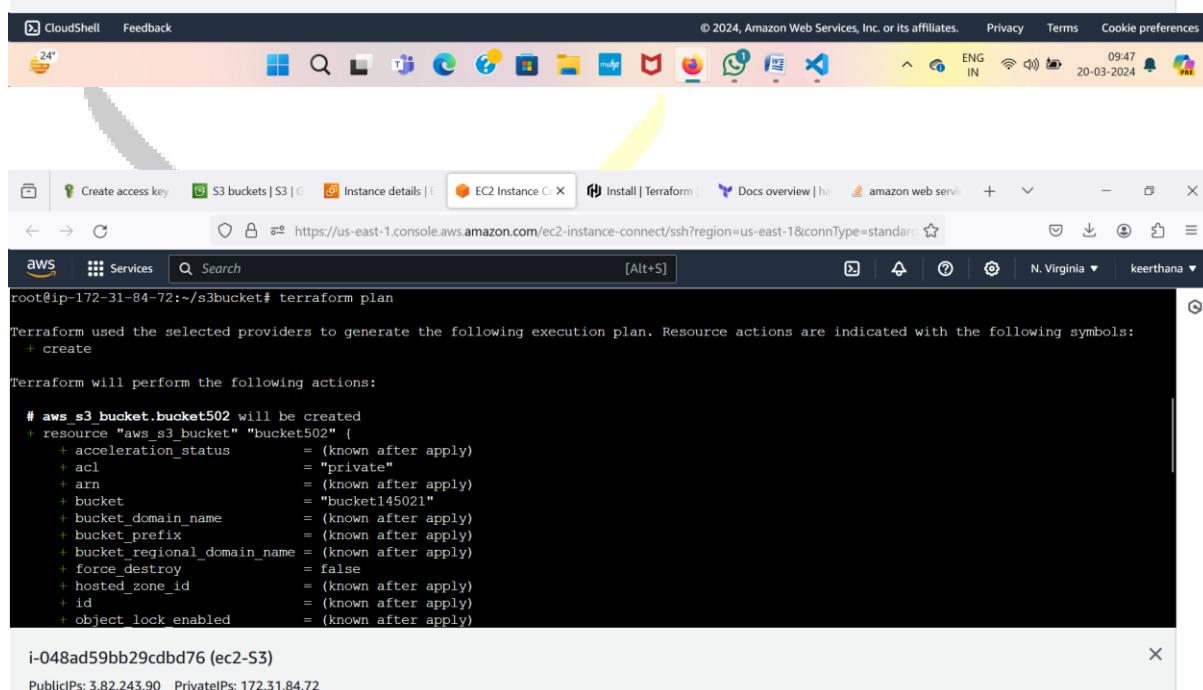
Terraform has created a lock file .terraform.lock.hcl to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future.

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.
```

i-048ad59bb29cdbc76 (ec2-S3)

Public IPs: 3.82.243.90 Private IPs: 172.31.84.72



```
root@ip-172-31-84-72:~/s3bucket# 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_s3_bucket.bucket502 will be created
+ resource "aws_s3_bucket" "bucket502" {
  + acceleration_status      = (known after apply)
  + acl                      = "private"
  + arn                      = (known after apply)
  + bucket                   = "bucket145021"
  + bucket_domain_name       = (known after apply)
  + bucket_prefix             = (known after apply)
  + bucketRegionalDomainName = (known after apply)
  + force_destroy             = false
  + hostedZoneId              = (known after apply)
  + id                        = (known after apply)
  + objectLockEnabled         = (known after apply)
}
```

i-048ad59bb29cdbc76 (ec2-S3)

Public IPs: 3.82.243.90 Private IPs: 172.31.84.72



Plan: 1 to add, 0 to change, 0 to destroy.

**Warning: Argument is deprecated**

```
with aws_s3_bucket.bucket502,
on resource.tf line 3, in resource "aws_s3_bucket" "bucket502":
  3:   acl = "private"
```

Use the aws\_s3\_bucket\_acl resource instead  
(and one more similar warning elsewhere)

---

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

```
root@ip-172-31-84-72:~/s3bucket#
```

i-048ad59bb29cdbc76 (ec2-S3)  
Public IPs: 3.82.243.90 Private IPs: 172.31.84.72

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

24°

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

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

```
root@ip-172-31-84-72:~/s3bucket# 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_s3_bucket.bucket502 will be created
+ resource "aws_s3_bucket" "bucket502" {
  + acceleration_status      = (known after apply)
  + acl                      = "private"
  + arn                      = (known after apply)
  + bucket                   = "bucket145021"
  + bucket_domain_name       = (known after apply)
  + bucket_prefix             = (known after apply)
  + bucketRegionalDomainName = (known after apply)
  + force_destroy             = false
  + hosted_zone_id           = (known after apply)
  + id                       = (known after apply)
  + object_lock_enabled       = (known after apply)
}
```

i-048ad59bb29cdbc76 (ec2-S3)  
Public IPs: 3.82.243.90 Private IPs: 172.31.84.72



```

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_s3_bucket.bucket502: Creating...
aws_s3_bucket.bucket502: Creation complete after 6s [id=bucket145021]

Warning: Argument is deprecated
  with aws_s3_bucket.bucket502,
  on resource.tf line 3, in resource "aws_s3_bucket" "bucket502":
  3:   acl = "private"

Use the aws_s3_bucket_acl resource instead

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
root@ip-172-31-84-72:~/s3bucket# 

```

i-048ad59bb29cdbd76 (ec2-S3)  
Public IPs: 3.82.243.90 Private IPs: 172.31.84.72

- Now go to S3 then see the bucket created or not

Name	AWS Region	Access	Creation date
bucket145021	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	March 20, 2024, 09:49:48 (UTC+05:30)

- Now upload files into created bucket
- Create create file and write something in that file and copy the file path and paste into the source in object.tf
- create one object.tf file write script for objects upload then “**terraform apply –auto-approve**” the command.

Instance details | EC2 | us- EC2 Instance Connect | us Create access key | IAM | X Install | Terraform | HashiCorp aws\_s3\_bucket\_object | Re X + - ×

https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-02a8a4b5 ☆

AWS Services Search [Alt+S] N. Virginia keerthana

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_s3 bucket.bucket: Creating...
aws_s3 bucket.bucket: Creation complete after 0s [id=bucket145021]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
root@ip-172-31-26-240:~/bucket# vi object.tf
root@ip-172-31-26-240:~/bucket# ls
object.tf provider.tf resource.tf terraform.tfstate
root@ip-172-31-26-240:~/bucket# touch file.txt
root@ip-172-31-26-240:~/bucket# ls
file.txt object.tf provider.tf resource.tf terraform.tfstate
root@ip-172-31-26-240:~/bucket# pwd
/root/bucket
root@ip-172-31-26-240:~/bucket# ^C
root@ip-172-31-26-240:~/bucket#
```

i-02a8a4b53c4a0363f (ec2-s3)

Public IPs: 54.242.240.245 Private IPs: 172.31.26.240

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

36°C Mostly sunny

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

Instance details | EC2 | us- EC2 Instance Connect | us Create access key | IAM | X Install | Terraform | HashiCorp aws\_s3\_bucket\_object | Re X + - ×

https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-02a8a4b5 ☆

AWS Services Search [Alt+S] N. Virginia keerthana

```
root@ip-172-31-26-240:~/bucket# cat object.tf
resource "aws_s3_bucket_object" "object" {
  bucket = "bucket145021"
  key    = "file.txt"
  source = "/root/bucket/file.txt"
}
root@ip-172-31-26-240:~/bucket#
```

i-02a8a4b53c4a0363f (ec2-s3)

Public IPs: 54.242.240.245 Private IPs: 172.31.26.240



```
root@ip-172-31-84-72:~/s3bucket# cat reso.tf
resource "aws_s3_object" "object" {
  bucket = "bucket145021"
  key    = "taskec2.docx"
  source = "C:/Users/mshar/OneDrive/Desktop/taskec2.docx"
}
root@ip-172-31-84-72:~/s3bucket# terraform plan
aws_s3_bucket.bucket: Refreshing state... [id=bucket145021]
```

i-048ad59bb29cdbd76 (ec2-S3)

PublicIPs: 3.82.243.90 PrivateIPs: 172.31.84.72

```
root@ip-172-31-26-240:~/bucket# cat object.tf
resource "aws_s3_bucket_object" "object" {
  bucket = "bucket145021"
  key    = "file.txt"
  source = "/root/bucket/file.txt"
}
root@ip-172-31-26-240:~/bucket# terraform apply --auto-approve
aws_s3_bucket.bucket: Refreshing state... [id=bucket145021]
```

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\_s3\_bucket\_object.object will be created

i-02a8a4b53c4a0363f (ec2-s3)

PublicIPs: 54.242.240.245 PrivateIPs: 172.31.26.240



```
1: resource "aws_s3_bucket_object" "object" {
use the aws_s3_object resource instead
(and 2 more similar warnings elsewhere)

Warning: Argument is deprecated

with aws_s3_bucket_object.object,
on object.tf line 2, in resource "aws_s3_bucket_object" "object":
2:   bucket = "bucket145021"

Use the aws_s3_object resource instead
(and 5 more similar warnings elsewhere)

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
root@ip-172-31-26-240:~/bucket#
```

i-02a8a4b53c4a0363f (ec2-s3)  
Public IPs: 54.242.240.245 Private IPs: 172.31.26.240

- Now we can see the file in the bucket.

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

bucket145021

Objects (1)

Name	Type	Last modified	Size	Storage class
file.txt	txt	March 26, 2024, 18:08:49 (UTC+05:30)	0 B	Standard

\*\*\*\*\* END \*\*\*\*\*

