



ASSIGNMENT – 3

COURSE : DEVOPS

Trainer : Mr . MADHUKAR

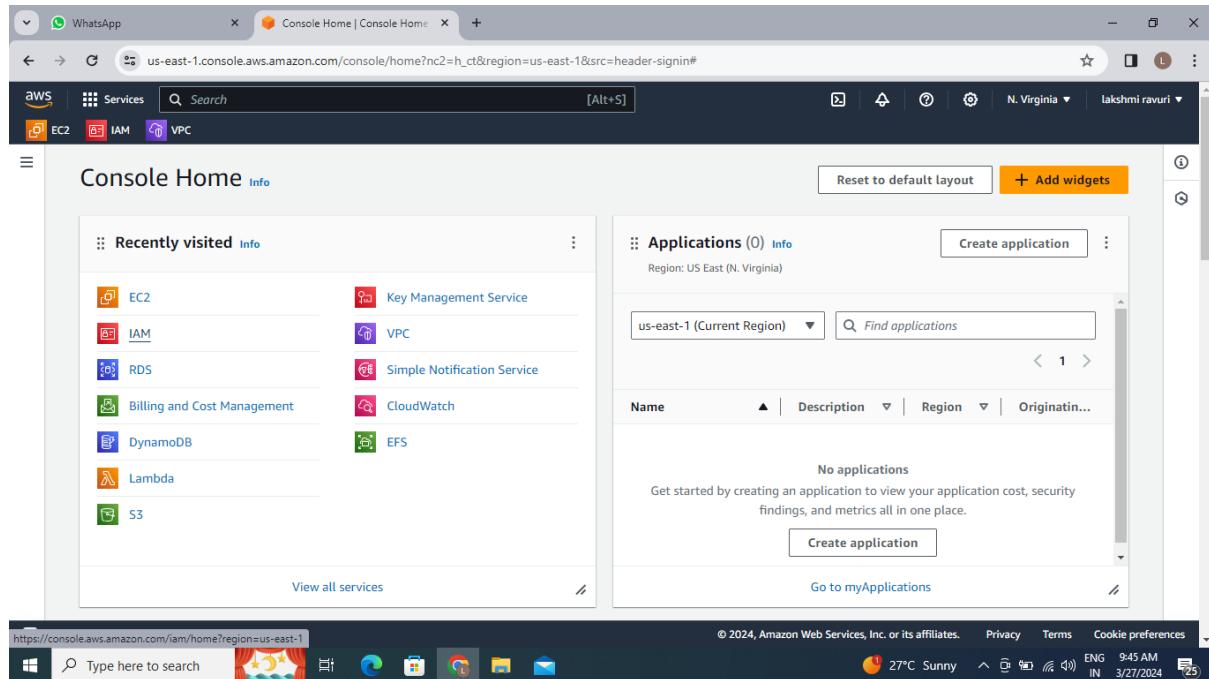
NAME:R.Lakshmi

Mail id :ravurilakshmi2315@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

VCUBE
Software Solutions Pvt. Ltd.

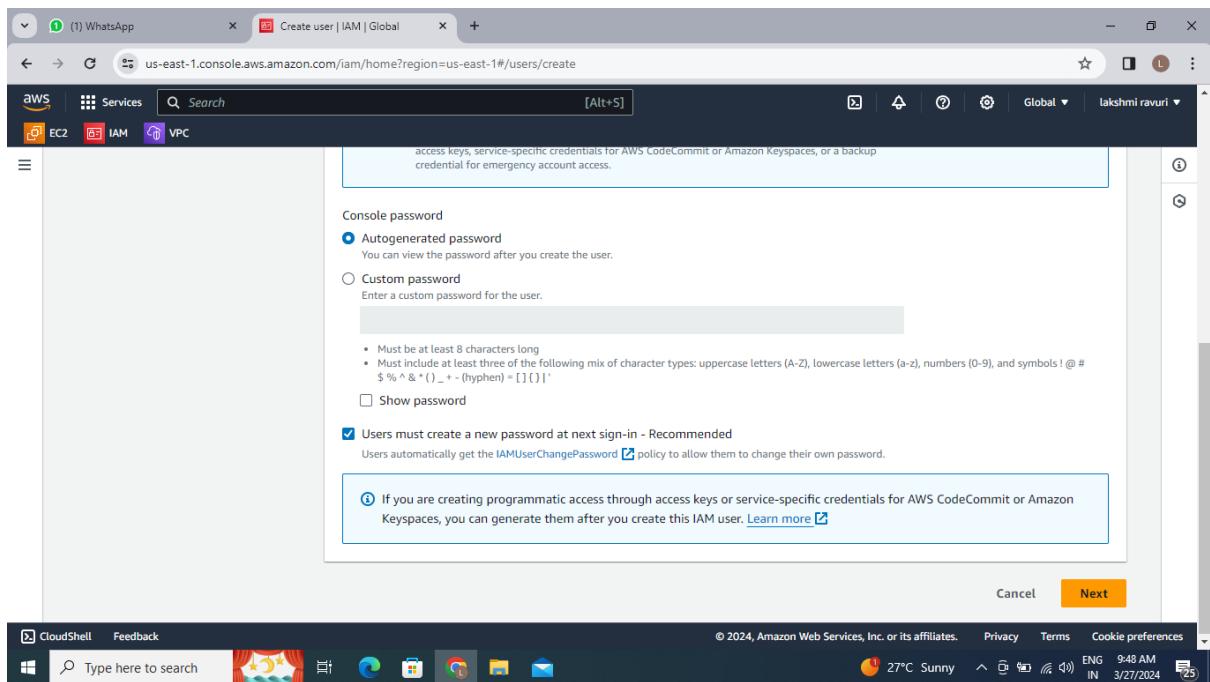
The screenshot shows the AWS CloudWatch Metrics console. A search bar at the top contains the text 'ia'. Below the search bar, there are two main sections: 'Services' and 'Features'. The 'Services' section lists various AWS services like IAM, EC2, RDS, and S3. The 'Features' section lists features like IAM Identity Center, MediaStore, and MediaLive. On the right side of the screen, there is a sidebar with options like 'Add widgets', 'Create application', and 'Originating...'. The bottom of the screen shows the Windows taskbar with icons for CloudShell, Feedback, and several other applications.

- 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

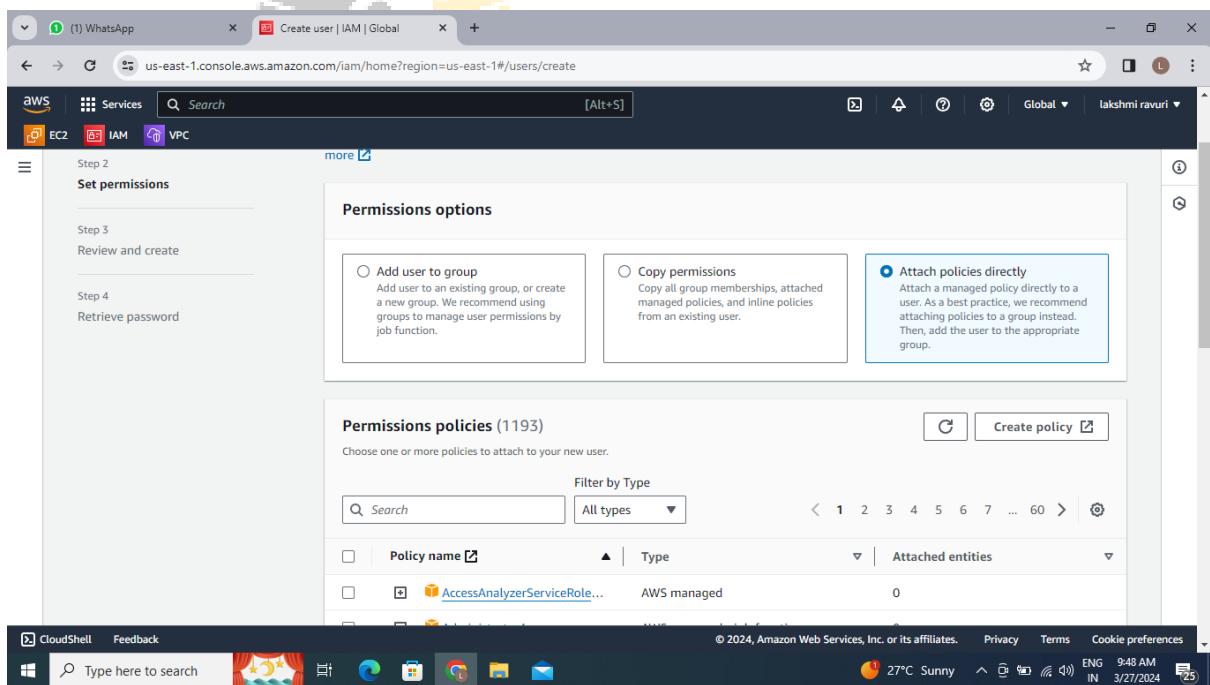
The screenshot shows the AWS IAM Dashboard. The left sidebar has a navigation menu with 'Identity and Access Management (IAM)' selected. Under 'Access management', there are links for User groups, Users, Roles, Policies, Identity providers, and Account settings. Under 'Access reports', there are links for Access Analyzer and External access. The main content area is titled 'IAM Dashboard' and includes sections for 'Security recommendations' (with a red notification dot) and 'AWS Account'. The 'Security recommendations' section lists 'Add MFA for root user' (with a warning icon) and 'Root user has no active access keys' (with a green checkmark). The 'AWS Account' section shows the Account ID (891377137975), Account Alias (Create), and a Sign-in URL (https://891377137975.signin.aws.amazon.com/console). There is also a 'Quick Links' section with a link to 'My security credentials'. The bottom of the screen shows the Windows taskbar with icons for CloudShell, Feedback, and several other applications.

The screenshot shows the AWS IAM service interface. The left sidebar is titled "Identity and Access Management (IAM)" and includes sections for Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings), and Access reports (Access Analyzer, External access). The main content area is titled "Users (0) Info" and contains a message stating, "An IAM user is an identity with long-term credentials that is used to interact with AWS in an account." A search bar and a table header with columns for User name, Path, Group, Last activity, MFA, and Password age are visible. The table body displays the message "No resources to display". At the top right, there are "Create user" and "Delete" buttons.

The screenshot shows the "Create user" wizard at Step 1: "Specify user details". The left sidebar lists steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main content area is titled "Specify user details" and contains a "User details" section. It shows a "User name" input field with the value "s3bucketuser". Below it is a note: "The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)". A checked checkbox option "Provide user access to the AWS Management Console - optional" is present, with a note: "If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center." A callout box asks, "Are you providing console access to a person?", with two options: "Specify a user in Identity Center - Recommended" (radio button unselected) and "I want to create an IAM user" (radio button selected). The note for the second option states: "We recommend that you create IAM users only if you need to enable programmatic access through the AWS CLI or AWS SDKs, or if you need to use AWS Lambda, Amazon Kinesis, or Amazon CloudWatch Metrics triggers." The bottom of the screen shows the Windows taskbar and system status.



- 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 User Creation wizard, Step 2: Set permissions policies. The search bar shows "s3".

Permissions policies (1/1193)

Choose one or more policies to attach to your new user.

Filter by Type: All types (22 matches)

Policy name	Type	Attached entities
AmazonDMSRedshiftS3Role	AWS managed	0
AmazonS3FullAccess	AWS managed	1
AmazonS3ObjectLambdaE...	AWS managed	0
AmazonS3OutpostsFullAcc...	AWS managed	0
AmazonS3OutpostsReadO...	AWS managed	0
AmazonS3ReadOnlyAccess	AWS managed	0
AWSBackupServiceRolePol...	AWS managed	0
AWSBackupServiceRolePol...	AWS managed	0
AWSS3OnOutpostsService...	AWS managed	0

Screenshot of the AWS IAM User Creation wizard, Step 2: Set permissions policies. The search bar shows "s3".

Permissions policies (1/1193)

Choose one or more policies to attach to your new user.

Filter by Type: All types (22 matches)

Policy name	Type	Attached entities
QuickSightAccessForS3Sto...	AWS managed	0
s3crr_for_assignment-120bat...	Customer managed	1
s3crr_for_laks-source-bucket_...	Customer managed	1
s3crr_for_siva-vcube-120_1d...	Customer managed	1
s3crr_for_siva-vcube-120_b9f...	Customer managed	1
s3replicate_for_assignment-1...	Customer managed	1
s3replicate_for_laks-source-b...	Customer managed	1

▶ Set permissions boundary - optional

Screenshot of the AWS IAM User Creation wizard, Step 2: Set permissions policies. The search bar shows "s3".

Cancel Previous Next

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

CloudShell Feedback 27°C Sunny ENG 9:49 AM IN 3/27/2024

The image consists of three vertically stacked screenshots of the AWS IAM 'Create user' wizard, specifically Step 4: Retrieve password.

Screenshot 1: Shows the 'Permissions summary' section. It lists two managed policies: 'AmazonS3FullAccess' and 'IAMUserChangePassword'. Both are categorized as 'Permissions policy' and are listed under the 'Used as' column.

Name	Type	Used as
AmazonS3FullAccess	AWS managed	Permissions policy
IAMUserChangePassword	AWS managed	Permissions policy

Screenshot 2: Shows the 'Tags - optional' section. It indicates 'No tags associated with the resource.' and provides a button to 'Add new tag'. A note states: 'You can add up to 50 more tags.'

Screenshot 3: Shows a green success message: 'User created successfully'. It says: 'You can view and download the user's password and email instructions for signing in to the AWS Management Console.' Below this, it shows the 'Retrieve password' step details, including the 'Console sign-in details' section which lists the sign-in URL (<https://891377137975.sigin.aws.amazon.com/console>), user name ('s3bucketuser'), and console password ('*****'). Buttons for 'Download .csv file' and 'Return to users list' are at the bottom.

- 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

Two screenshots of the AWS IAM console are shown side-by-side, illustrating the creation and configuration of a new user.

Screenshot 1: User Creation

The top screenshot shows the "Users" page after creating a new user named "s3bucketuser". A green success message at the top states: "User created successfully. You can view and download the user's password and email instructions for signing in to the AWS Management Console." The "Users (1) Info" table lists the new user:

User name	Path	Group	Last activity	MFA	Password age
s3bucketuser	/	0	-	-	-

Screenshot 2: User Configuration

The bottom screenshot shows the "s3bucketuser" user details page. The "Summary" section provides basic information:

ARN	Console access	Access key
arn:aws:iam::891377137975:user/s3bucketuser	Enabled without MFA	Access key 1 Create access key

The "Permissions" tab is selected, showing two attached policies:

Permissions policies (2)	Add permissions
Permissions are defined by policies attached to the user directly or through groups.	C Remove Add permissions ▾

Both screenshots include a Windows taskbar at the bottom with icons for CloudShell, Feedback, Start, Task View, Edge, Google Chrome, File Explorer, and Mail.

The screenshot shows the AWS IAM 'Create access key' wizard. The current step is 'Access key best practices & alternatives'. A note at the top says: 'Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.' Below this, there are three sections: 'Step 1 Access key best practices & alternatives', 'Step 2 - optional Set description tag', and 'Step 3 Retrieve access keys'. The 'Step 1' section is expanded, showing a 'Use case' dropdown with the following options:

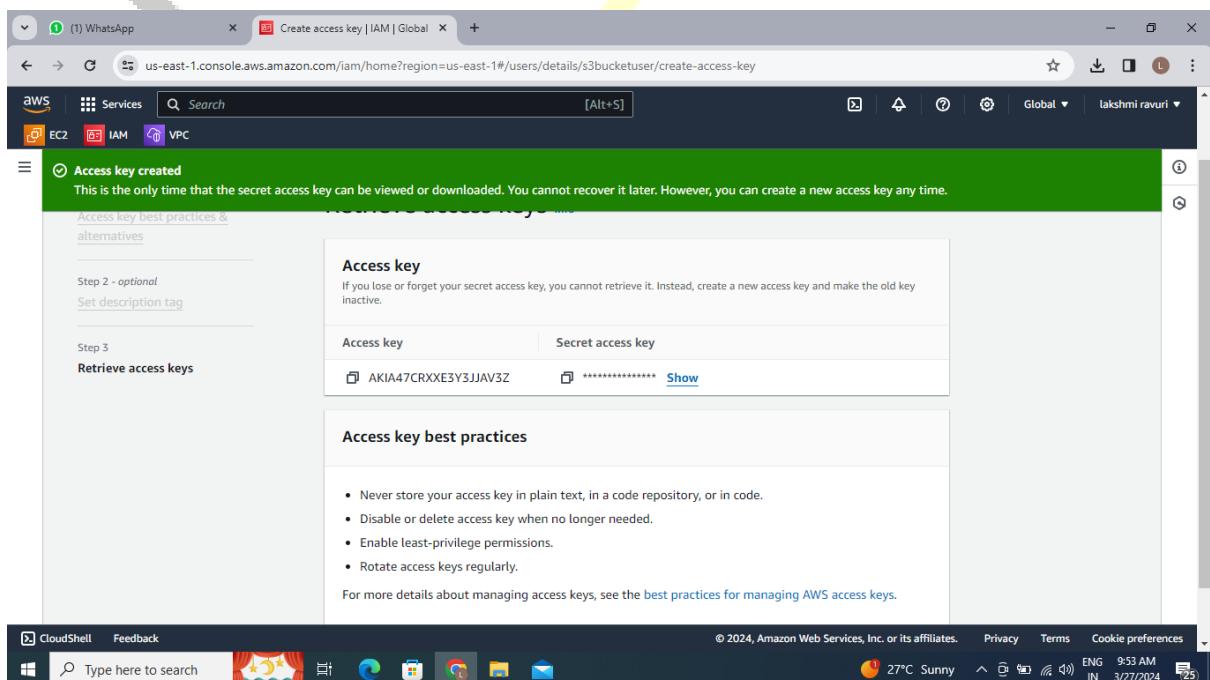
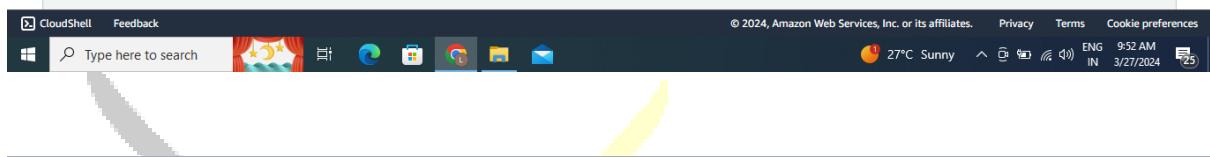
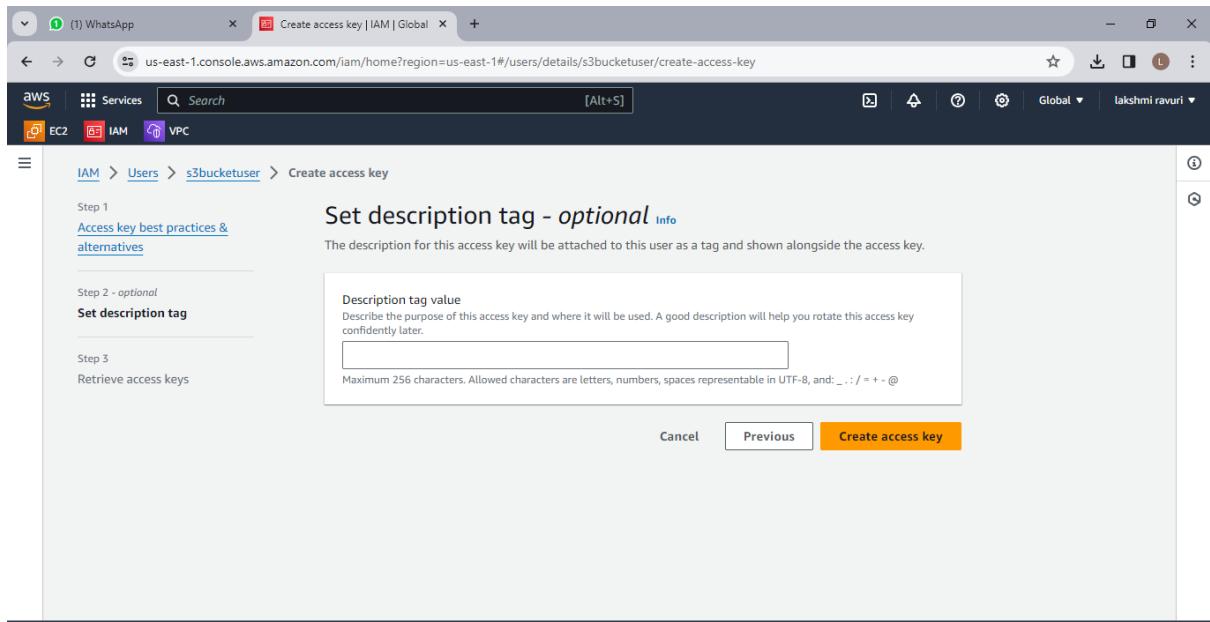
- Command Line Interface (CLI)**
You plan to use this access key to enable the AWS CLI to access your AWS account.
- Local code**
You plan to use this access key to enable application code in a local development environment to access your AWS account.
- Application running on an AWS compute service**
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.

Below the use case dropdown, there is a yellow box titled 'Alternatives recommended' containing the following text:

A Alternatives recommended

- Use [AWS CloudShell](#), a browser-based CLI, to run commands. [Learn more](#)
- Use the [AWS CLI V2](#) and enable authentication through a user in IAM Identity Center. [Learn more](#)

At the bottom of the page, there is a 'Confirmation' section with a checkbox labeled 'I understand the above recommendation and want to proceed to create an access key.' and two buttons: 'Cancel' and 'Next'.



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

The screenshot shows the AWS IAM console with a search bar at the top containing 'ec2'. On the left, a sidebar for 'Access key creation' is open, showing steps 1 through 3. Step 1 is titled 'This is the only way to access your EC2 instances' and includes links for 'IAM > User' and 'Access key best alternatives'. Step 2 is 'Step 2 - optional' with 'Set descriptive names' and Step 3 is 'Step 3' with 'Retrieve access keys'. The main content area shows 'Search results for "ec2"' under 'Services (13)'. It lists 'EC2' (Virtual Servers in the Cloud), 'EC2 Image Builder' (A managed service to automate build, customize and deploy OS images), 'Recycle Bin' (Protect resources from accidental deletion), and 'Amazon Inspector' (Continual vulnerability management at scale). A note at the bottom says '• Enable least-privilege permissions.'

The screenshot shows the AWS EC2 Instances page. The URL is 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances: v=3:\$case=true%5C;client:false;\$regex=tags:false%5C;client:false'. The sidebar on the left has sections for 'Instances' (selected), 'Images', and 'AMIs'. The main content area is titled 'Instances Info' and shows a search bar with '[Alt+S]'. It displays a message 'No instances' and 'You do not have any instances in this region'. A 'Launch instances' button is visible. The bottom of the screen shows a taskbar with various icons and system status.

Screenshot of the AWS CloudShell interface showing the EC2 Instances launch wizard.

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: ec2-s3

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

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

Launch instance

CloudShell Feedback Type here to search 27°C Sunny ENG IN 9:55 AM 3/27/2024 (25)

Screenshot of the AWS CloudShell interface showing the EC2 Instances launch wizard.

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
ami-080e1f13689e07408 (64-bit (x86)) / ami-0a55ba1c20b74fc30 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2024-03-01

Architecture: 64-bit (x86) **AMI ID**: ami-080e1f13689e07408 **Verified provider**

Instance type

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ami-080e1f13689e07408

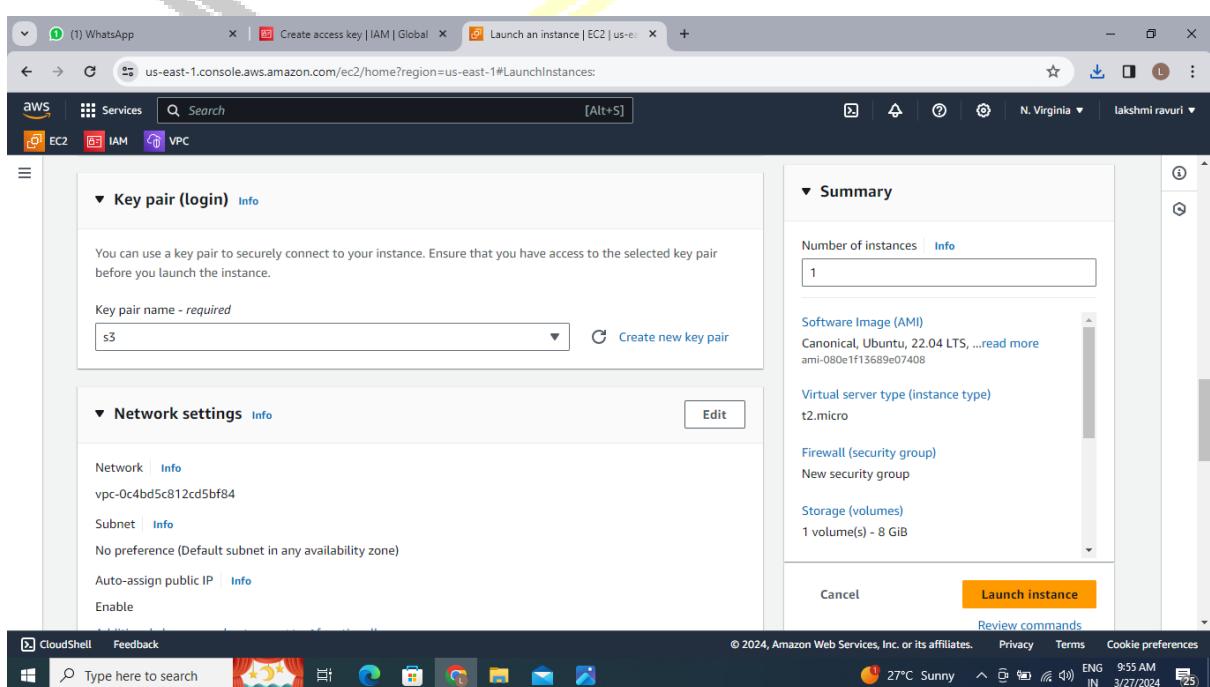
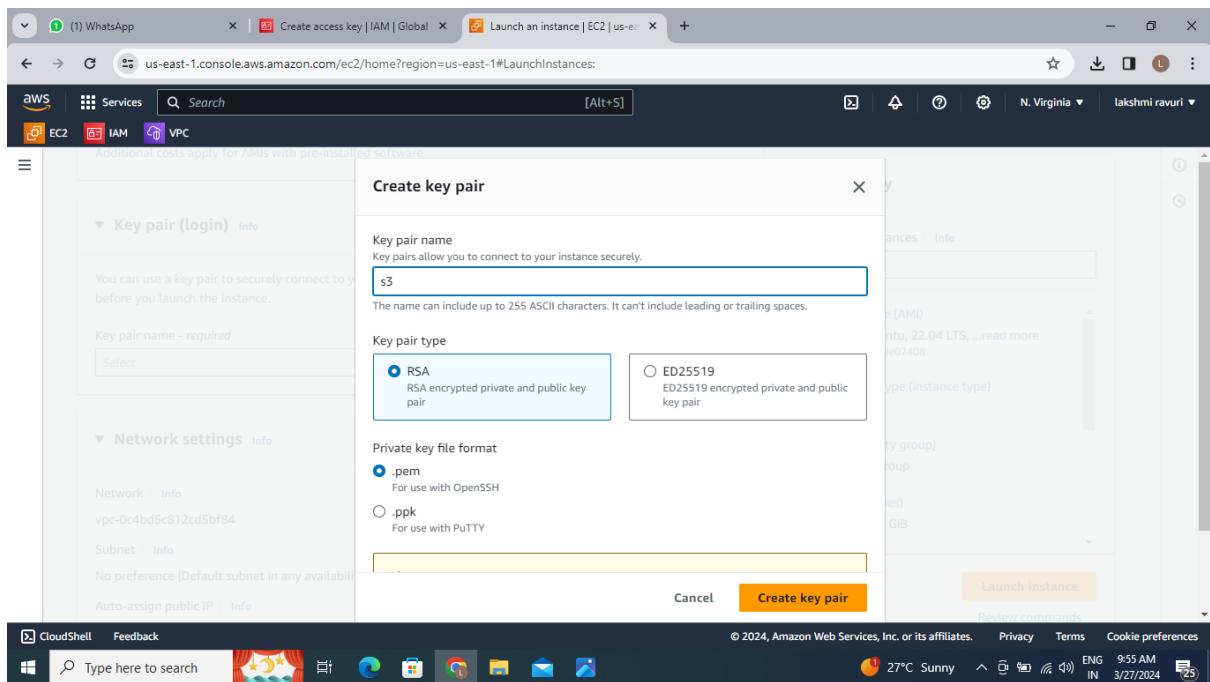
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

Launch instance

CloudShell Feedback Type here to search 27°C Sunny ENG IN 9:55 AM 3/27/2024 (25)



Screenshot of the AWS CloudShell interface. The terminal window shows the output of a Lambda function execution:

```
Execution ID: 1234567890abcdef1234567890abcdef
Function Name: HelloWorld
Region: us-east-1
Runtime: Node.js 18
Memory: 128 MB
Duration: 123 ms
Logs:
```

The logs output is:

```
hello world
```

Screenshot of the AWS CloudShell interface. The terminal window shows the output of a Lambda function execution:

```
Execution ID: 1234567890abcdef1234567890abcdef
Function Name: HelloWorld
Region: us-east-1
Runtime: Node.js 18
Memory: 128 MB
Duration: 123 ms
Logs:
```

The logs output is:

```
hello world
```

Screenshot of the AWS Cloud Console showing the EC2 Instances details page for an instance named i-06ccc44b4333a12c2 (ec2-s3). The instance is running and has a public IPv4 address of 3.91.255.206.

Instance summary for i-06ccc44b4333a12c2 (ec2-s3)

Instance ID	Public IPv4 address	Private IPv4 addresses
i-06ccc44b4333a12c2 (ec2-s3)	3.91.255.206 [open address]	172.31.205
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-3-91-255-206.compute-1.amazonaws.com [open address]
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-31-31-205.ec2.internal	ip-172-31-31-205.ec2.internal	-
Answer private resource DNS name	Instance type	AWS Compute Optimizer finding
IPv4 (A)	t2.micro	[Opt-in to AWS Compute Optimizer for recommendations.]
Auto-assigned IP address	VPC ID	Learn more
3.91.255.206 [Public IP]	vpc-0c4bd5c812cd5bf84	-
IAM Role	Subnet ID	Auto Scaling Group name
-	subnet-03b71ac03ef0b660d	-

CloudShell **Feedback** © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 27°C Sunny ENG IN 9:57 AM 3/27/2024 25

Screenshot of the AWS Cloud Console showing the 'Connect to instance' dialog for the same instance. The 'Connect using EC2 Instance Connect' option is selected, and the public IP address is listed as 3.91.255.206. A note at the bottom states: 'Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.'

Instance ID: i-06ccc44b4333a12c2 (ec2-s3)

Connection Type:

- Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
- Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address: 3.91.255.206

Username: ubuntu

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

Cancel **Connect**

CloudShell **Feedback** © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 27°C Sunny ENG IN 9:57 AM 3/27/2024 25

(1) WhatsApp Create access key | IAM | Global Instance details | EC2 | us-east-1 EC2 Instance Connect | us-east-1

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-06ccc44b4333a12c2&osUser=ubuntu®ion=us-east-1&sshPort=22

AWS Services Search [Alt+S] N. Virginia lakshmi ravuri

EC2 IAM VPC

* Support: <https://ubuntu.com/pro>

System information as of Wed Mar 27 04:27:33 UTC 2024

System load: 0.61669921875 Processes: 102
Usage of /: 20.4% of 7.57GB Users logged in: 0
Memory usage: 20% IPv4 address for eth0: 172.31.31.205
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

Last login: Wed Mar 27 04:27:35 2024 from 18.206.107.27
ubuntu@ip-172-31-31-205:~\$ sudo -i
root@ip-172-31-31-205:~# clear

i-06ccc44b4333a12c2 (ec2-s3)

PublicIPs: 3.91.255.206 PrivateIPs: 172.31.31.205

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

Type here to search

27°C Sunny ENG IN 9:58 AM 3/27/2024 25



- Now install awscli in ec2 instance commands is
Apt update -y
Apt install awscli -y



```

root@ip-172-31-31-205:~# apt update -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 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 [1505 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1290 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [290 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1628 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1059 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [240 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]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [472 B]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [67.1 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.0 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.0 kB]

i-06ccc44b4333a12c2 (ec2-s3)

PublicIPs: 3.91.255.206 PrivateIPs: 172.31.31.205

```



```

aws CloudShell Feedback
Type here to search 27°C Sunny ENG IN 9:59 AM 3/27/2024 (25)

```



```

root@ip-172-31-31-205:~# apt list --upgradable
19 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ip-172-31-31-205:~# 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 fonts-droid-fallback fonts-noto-mono fonts-urw-base35 ghostscript groff gsffonts
  hicolor-icon-theme imagemagick imagemagick-6-common imagemagick-6.q16 libacm3 libavahi-client3 libavahi-common-data libavahi-common3
  libcairo2 libcurl3 libdatrie1 libdav1d5 libde265-0 libdeflate0 libdjvu-libre-text libdjvu-libre21 libfftw3-double3 libfontconfig1 libgomp1
  libgraphite2-3 libgs9 libgs9-common libharfbuzz0 libheif1 libice6 libidn12 libjbig0.35 liblibase25 libimagequant0 libjbig0 libjbig2dec0
  libjpeg-turbo8 libjpeg8 libjxr-tools libjxr0 liblcms2-2 liblqr-1-0 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 libtiff5 libwebrtc7 libwebpdmux2 libwebpdmux3 libwmflite-0.2-7 libx265-199 libxaw7
  libxcb-render0 libxcb-shm0 libxmu6 libxpm4 libxrender1 libxt6 mailcap mime-support netpbm poppler-data putils python3-botcore
  python3-dateutil python3-docutils python3-jmespath python3-olefile python3-pygments python3-roman python3-rsa python3-s3transfer
  sgmil-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 fmpkg gimp gnuplot grads graphviz hp2xx html2ps libwmf-bin mplayer povray radiance sane-utils texlive-base-bin transfig ufraw-batch

i-06ccc44b4333a12c2 (ec2-s3)

PublicIPs: 3.91.255.206 PrivateIPs: 172.31.31.205

```



```

CloudShell Feedback
Type here to search 27°C Sunny ENG IN 10:01 AM 3/27/2024 (25)

```

- After Installing awscli then configure the user to awscli

Aws configure

Access key = ""

Secret access key = ""

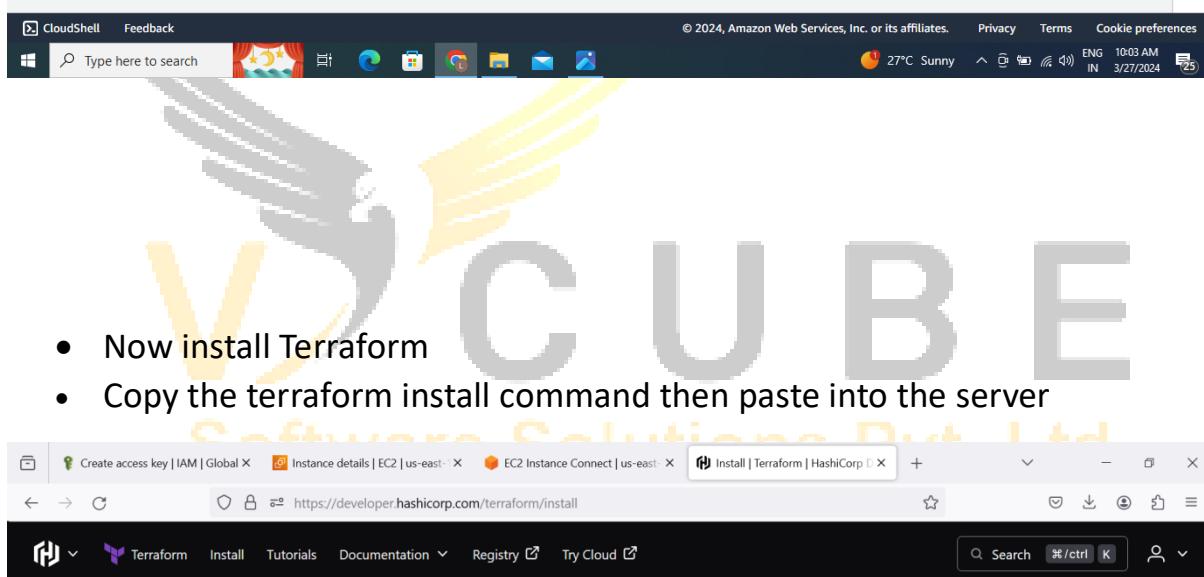
Default region = ""

Output format = ""

```
root@ip-172-31-205:~# aws configure
AWS Access Key ID [None]: AKIA47CRXXE3Y3JJAV3Z
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1d
Default output format [None]: table
root@ip-172-31-205:~#
```

i-06ccc44b4333a12c2 (ec2-s3)

Public IPs: 3.91.255.206 Private IPs: 172.31.31.205



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

The screenshot shows the HashiCorp Terraform website's 'Install Terraform' page. The main header has links for Create access key, Instance details, EC2 Instance Connect, and Install Terraform. Below the header, there's a sidebar with Operating Systems: macOS, Windows, Linux (selected), FreeBSD, OpenBSD, and Solaris. The main content area has sections for Linux (Package manager and Binary download), About Terraform (with a definition and links to docs), and Featured docs (with links to Introduction to Terraform, Configuration Language, Terraform CLI, Terraform Cloud, and Provider Use). At the bottom, there's a cookie consent banner and a system tray with various icons.

```

Default output format [None]: table
root@ip-172-31-31-205:~# 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
--2024-03-27 04:34:19-- https://apt.releases.hashicorp.com/
Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com)... 99.84.108.36, 99.84.108.40, 99.84.108.74, ...
Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com)|99.84.108.36|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3980 (3.9K) [binary/octet-stream]
Saving to: 'STDOUT'

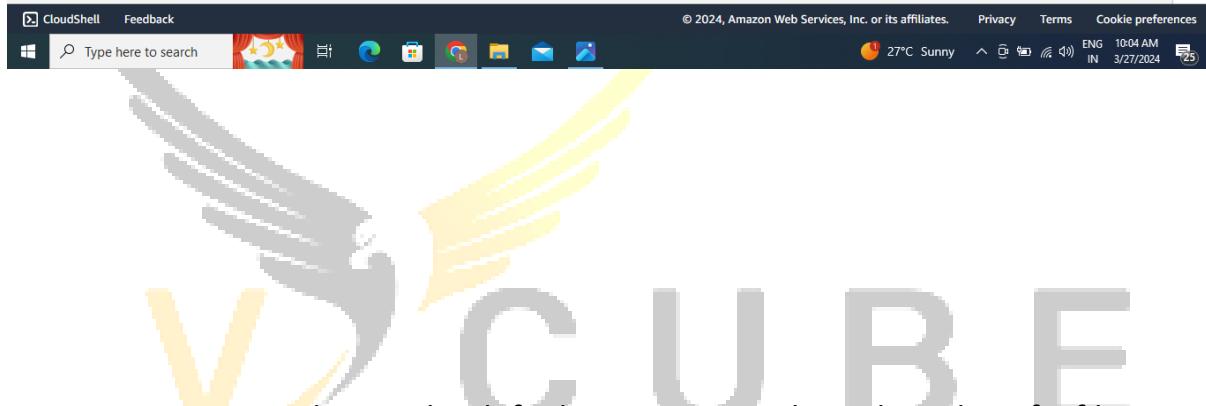
100%[=====] 3.89K --.-KB/s   in 0s

2024-03-27 04:34:19 (136 MB/s) - written to stdout [3980/3980]

deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com jammy main
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease

i-06ccc44b4333a12c2 (ec2-s3)
PublicIPs: 3.91.255.206 PrivateIPs: 172.31.31.205

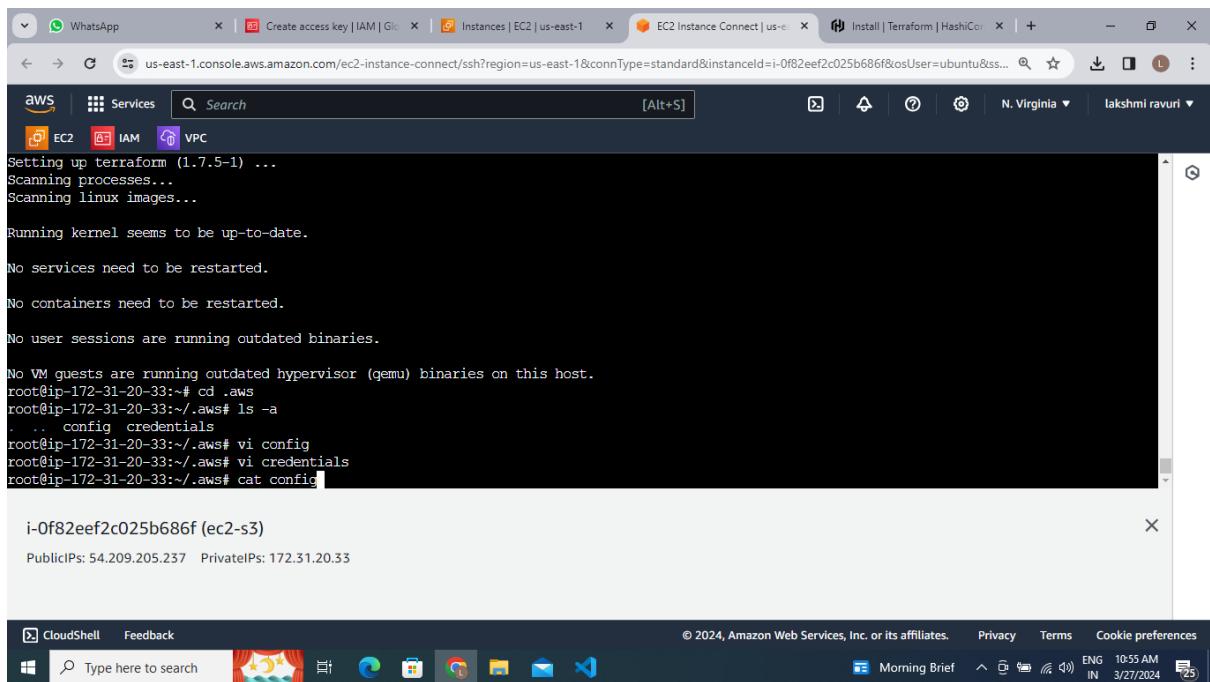
```



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

Software Solutions Pvt. Ltd.

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



```
Setting up terraform (1.7.5-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

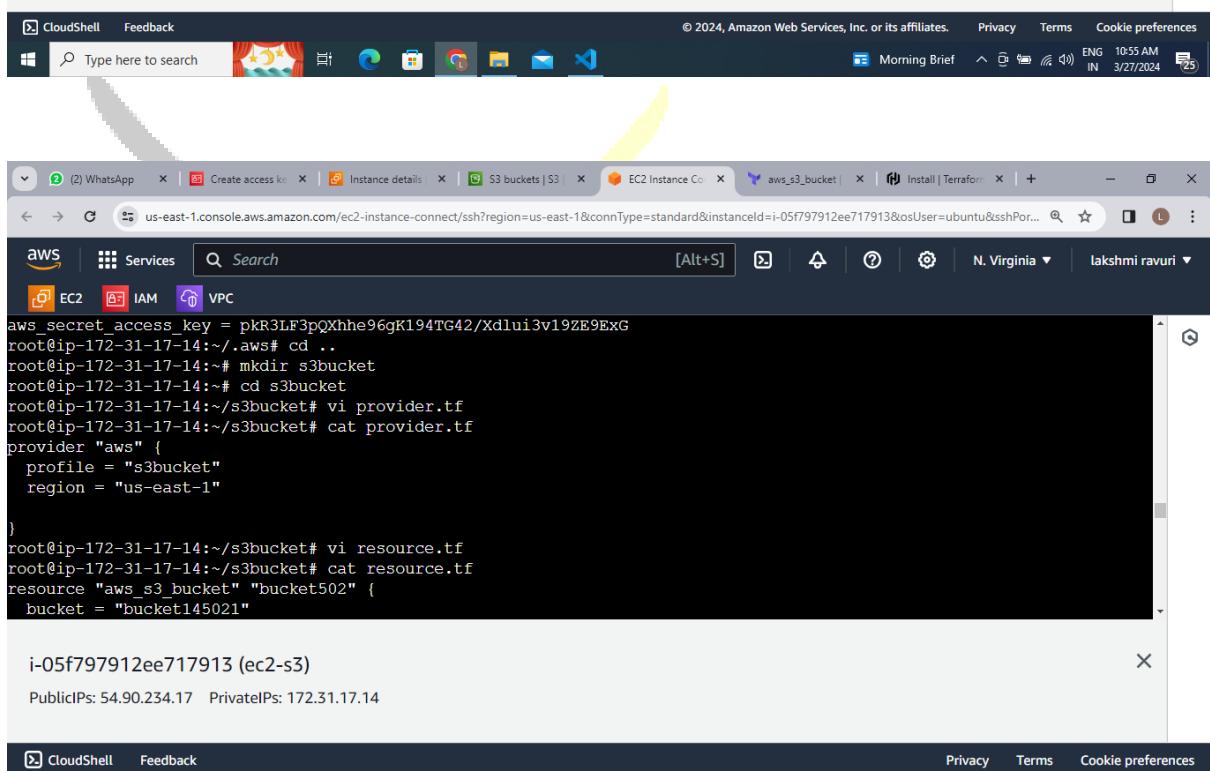
No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

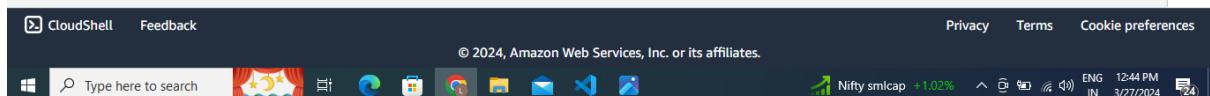
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-20-33:~# cd .aws
root@ip-172-31-20-33:~/.aws# ls -a
.  .. config credentials
root@ip-172-31-20-33:~/.aws# vi config
root@ip-172-31-20-33:~/.aws# vi credentials
root@ip-172-31-20-33:~/.aws# cat config
```

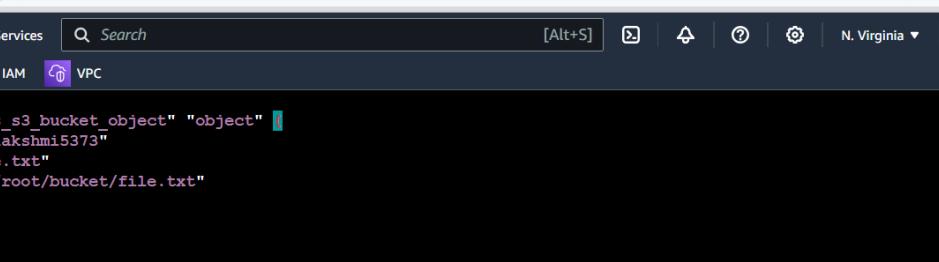
i-0f82eef2c025b686f (ec2-s3)
Public IPs: 54.209.205.237 Private IPs: 172.31.20.33



```
aws_secret_access_key = pkR3LF3pQXhhe96gK194TG42/Xdlui3v19ZE9ExG
root@ip-172-31-17-14:~/~.aws# cd ..
root@ip-172-31-17-14:~# mkdir s3bucket
root@ip-172-31-17-14:~# cd s3bucket
root@ip-172-31-17-14:~/s3bucket# vi provider.tf
root@ip-172-31-17-14:~/s3bucket# cat provider.tf
provider "aws" {
  profile = "s3bucket"
  region = "us-east-1"
}
root@ip-172-31-17-14:~/s3bucket# vi resource.tf
root@ip-172-31-17-14:~/s3bucket# cat resource.tf
resource "aws_s3_bucket" "bucket502" {
  bucket = "bucket145021"
```

i-05f797912ee717913 (ec2-s3)
Public IPs: 54.90.234.17 Private IPs: 172.31.17.14





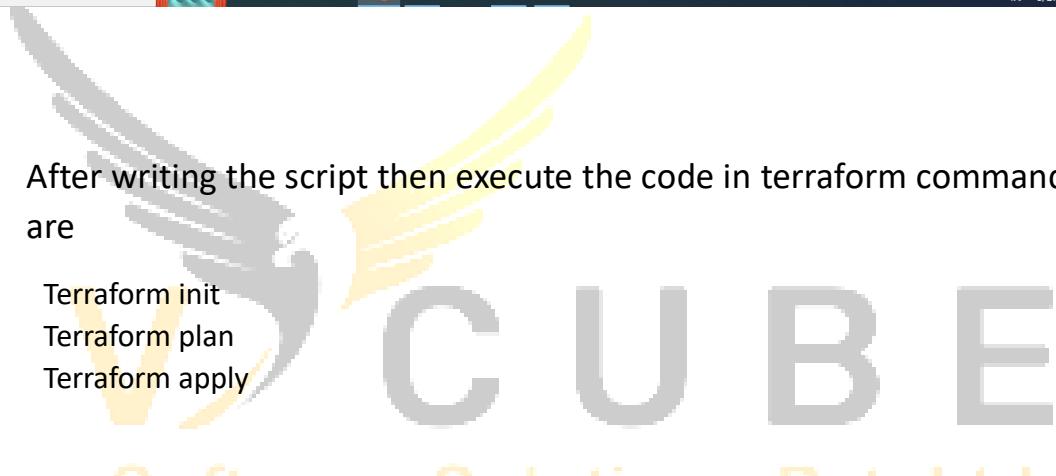
The screenshot shows a browser window with multiple tabs open, including 'Create access key', 'Instance details', 'S3 buckets | S3', 'EC2 Instance Co...', 'aws_s3_bucket', and 'Install | Terraform'. The main content area is a CloudShell terminal window titled 'aws' with the following Terraform code:

```
resource "aws_s3_bucket_object" "object" {
  bucket = "lakshmi5373"
  key    = "file.txt"
  source = "/root/bucket/file.txt"
```

Below the code, there is a scrollable terminal history. At the bottom of the terminal window, there is a message box containing the EC2 instance ID and its public and private IP addresses.

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

Terraform init
Terraform plan
Terraform apply



Create access key | IAM | Global | Lakshmi5373 - S3 bucket | S3 | EC2 Instance Connect | us-east-1 | Install | Terraform | HashiCorp | +

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?region=us-east-1&connType=standard&instanceId=i-068a89cf0f22216a&osUser=ubuntu&sshPort... | Search | [Alt+S] | N. Virginia | lakshmi ravuri

Services | Search | EC2 | IAM | VPC

```
root@ip-172-31-91-136:~/s3bucket# terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.42.0...
- Installed hashicorp/aws v5.42.0 (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!
```

i-068a89cf0f22216a (ec2-s3)

Public IPs: 54.174.14.20 Private IPs: 172.31.91.136

CloudShell | Feedback | Privacy | Terms | Cookie preferences

Type here to search | 31°C Partly cloudy | ENG IN 9:29 PM 3/27/2024 | 25

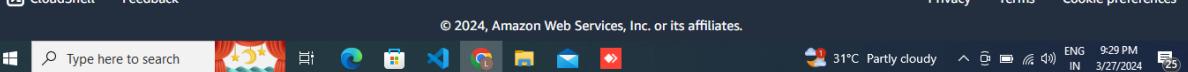
```
root@ip-172-31-91-136:~/s3bucket# terraform plan

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.bucket will be created
+ resource "aws_s3_bucket" "bucket" {
  + acceleration_status      = (known after apply)
  + acl                      = "private"
  + arn                      = (known after apply)
  + bucket                   = "lakshmi5373"
  + bucket_domain_name       = (known after apply)

i-068a89cf0f22216a (ec2-s3)
```



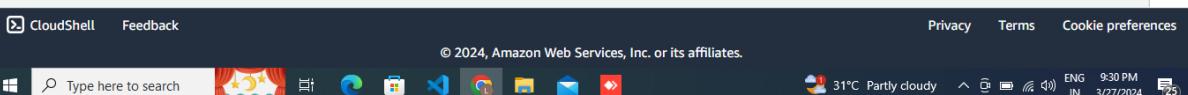
```
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
root@ip-172-31-91-136:~/s3bucket# terraform apply

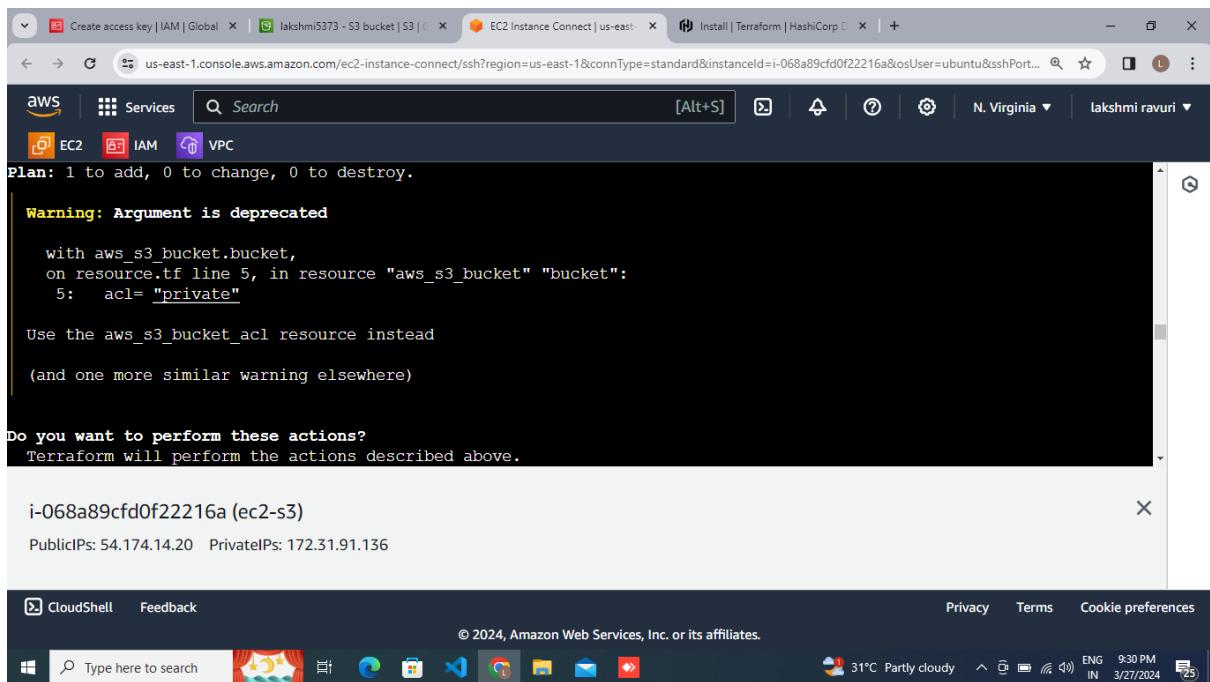
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_s3_bucket.bucket will be created
+ resource "aws_s3_bucket" "bucket" {
  + acceleration_status      = (known after apply)

i-068a89cf0f22216a (ec2-s3)
```





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

Warning: Argument is deprecated

```
with aws_s3_bucket.bucket,
on resource.tf line 5, in resource "aws_s3_bucket" "bucket":
  5:   acl= "private"
```

Use the aws_s3_bucket_acl resource instead

(and one more similar warning elsewhere)

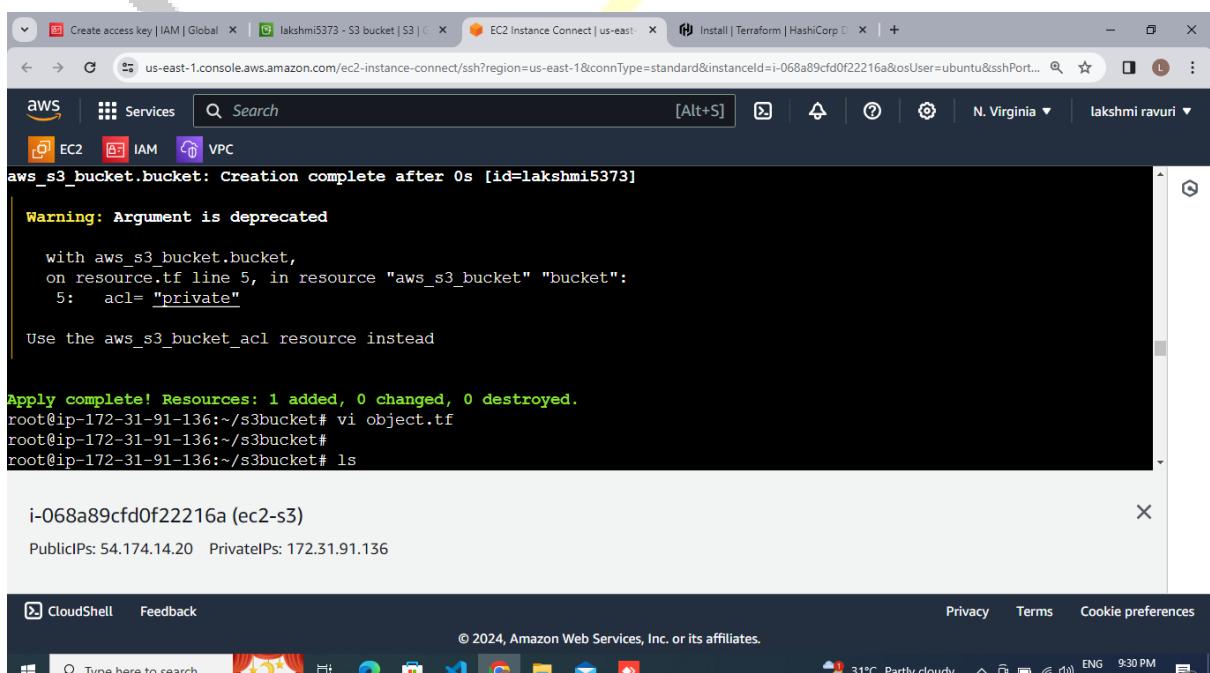
Do you want to perform these actions?
Terraform will perform the actions described above.

i-068a89cf0f22216a (ec2-s3)
PublicIPs: 54.174.14.20 PrivateIPs: 172.31.91.136

CloudShell Feedback Privacy Terms Cookie preferences

© 2024, Amazon Web Services, Inc. or its affiliates.

Windows Start button Type here to search Taskbar icons Weather 31°C Partly cloudy ENG IN 9:30 PM 3/27/2024



aws_s3_bucket.bucket: Creation complete after 0s [id=lakshmi5373]

Warning: Argument is deprecated

```
with aws_s3_bucket.bucket,
on resource.tf line 5, in resource "aws_s3_bucket" "bucket":
  5:   acl= "private"
```

Use the aws_s3_bucket_acl resource instead

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
root@ip-172-31-91-136:~/s3bucket# vi object.tf
root@ip-172-31-91-136:~/s3bucket#
root@ip-172-31-91-136:~/s3bucket# ls

i-068a89cf0f22216a (ec2-s3)
PublicIPs: 54.174.14.20 PrivateIPs: 172.31.91.136

CloudShell Feedback Privacy Terms Cookie preferences

© 2024, Amazon Web Services, Inc. or its affiliates.

Windows Start button Type here to search Taskbar icons Weather 31°C Partly cloudy ENG IN 9:30 PM 3/27/2024

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

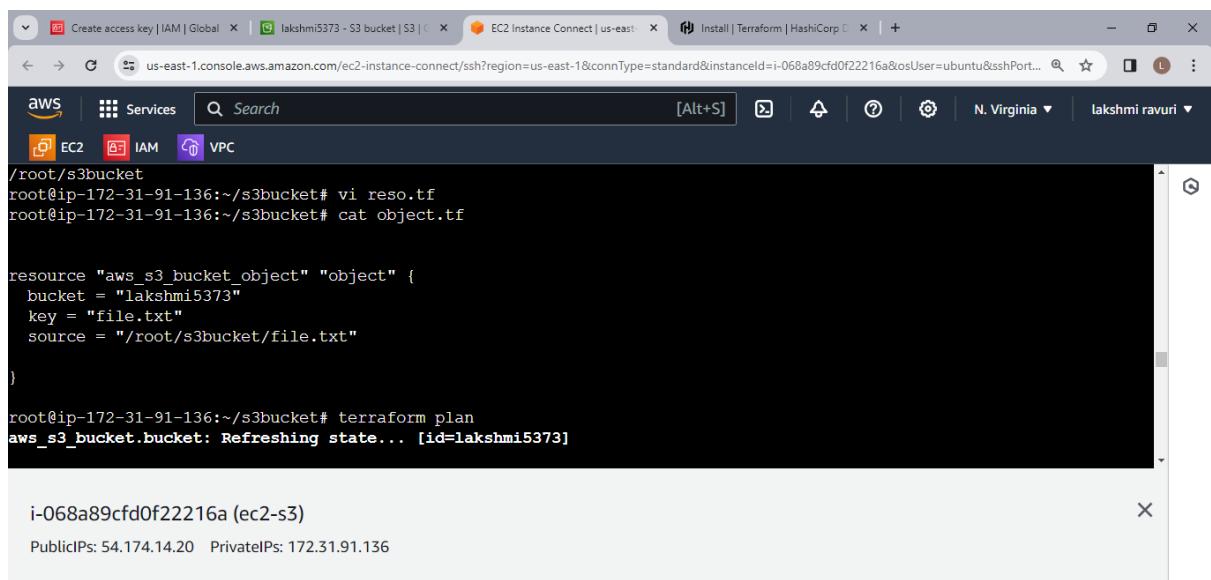
The screenshot shows the AWS S3 buckets page. On the left, there's a sidebar with various links like 'Batch Operations', 'Storage Lens', and 'Feature spotlight'. The main area displays a table for 'General purpose buckets (1)'. The table has columns for Name, AWS Region, Access, and Creation date. One row is shown for the bucket 'lakshmi5373' located in 'US East (N. Virginia) us-east-1' with 'Bucket and objects not public' access and created on 'March 27, 2024, 12:41:34 (UTC+05:30)'. At the top right of the table, there are buttons for 'Create bucket' (highlighted in yellow), 'Empty', 'Delete', and 'Copy ARN'. Below the table is a search bar with the placeholder 'Find buckets by name'.

- 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.

The screenshot shows a CloudShell session. The terminal window displays the output of a Terraform apply command. It shows the resources added and the commands run on the EC2 instance to create an S3 bucket and upload files. The session also shows the IP address and port information for the EC2 instance.

```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
root@ip-172-31-91-136:~/s3bucket# vi object.tf
root@ip-172-31-91-136:~/s3bucket#
root@ip-172-31-91-136:~/s3bucket# ls
object.tf provider.tf resource.tf terraform.tfstate
root@ip-172-31-91-136:~/s3bucket# touch file.txt
root@ip-172-31-91-136:~/s3bucket# ls
file.txt object.tf provider.tf resource.tf terraform.tfstate
root@ip-172-31-91-136:~/s3bucket# pwd
/root/s3bucket
root@ip-172-31-91-136:~/s3bucket# vi reso.tf
root@ip-172-31-91-136:~/s3bucket# cat object.tf

i-068a89cf0f22216a (ec2-s3)
PublicIPs: 54.174.14.20 PrivateIPs: 172.31.91.136
```



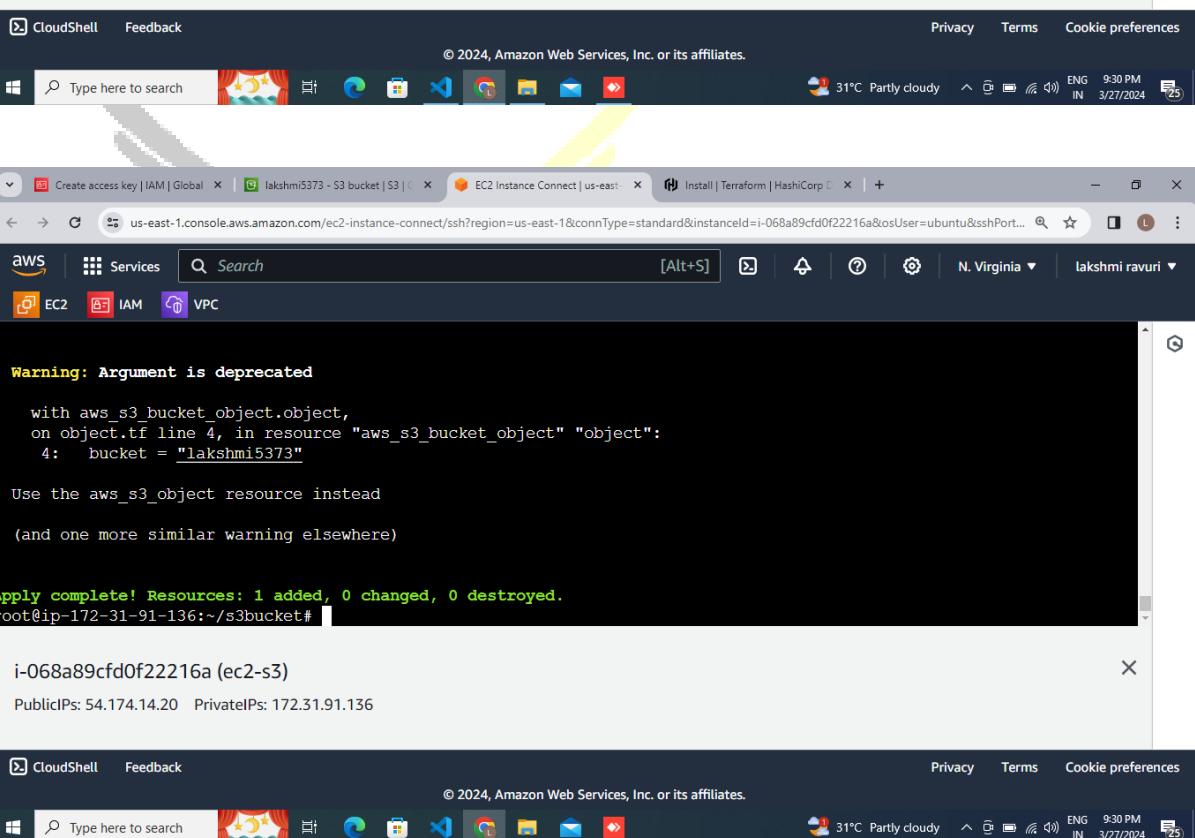
```
/root/s3bucket
root@ip-172-31-91-136:~/s3bucket# vi reso.tf
root@ip-172-31-91-136:~/s3bucket# cat object.tf

resource "aws_s3_bucket_object" "object" {
  bucket = "lakshmi5373"
  key = "file.txt"
  source = "/root/s3bucket/file.txt"
}

root@ip-172-31-91-136:~/s3bucket# terraform plan
aws_s3_bucket.bucket: Refreshing state... [id=lakshmi5373]
```

i-068a89cf0f22216a (ec2-s3)

PublicIPs: 54.174.14.20 PrivateIPs: 172.31.91.136



```
warning: Argument is deprecated
with aws_s3_bucket_object.object,
on object.tf line 4, in resource "aws_s3_bucket_object" "object":
  4:   bucket = "lakshmi5373"

Use the aws_s3_object resource instead
(and one more similar warning elsewhere)

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
root@ip-172-31-91-136:~/s3bucket#
```

i-068a89cf0f22216a (ec2-s3)

PublicIPs: 54.174.14.20 PrivateIPs: 172.31.91.136

The screenshot shows a browser window with the AWS Lambda console tab open. The terminal output displays the execution of a Terraform script to create an S3 object. The command run is:

```
root@ip-172-31-26-240:~/bucket# terraform apply --auto-approve
```

The output shows the creation of an S3 bucket object:

```
aws_s3_bucket.bucket: Refreshing state... [id=bucket145021]
```

Execution plan summary:

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

Action details:

```
Terraform will perform the following actions:

# aws_s3_bucket.object will be created
```

```
CloudShell Feedback
36°C Hot weather Search
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
ENG IN 18:08 26-03-2024

Instance details | EC2 | us EC2 Instance Connect | us Create access key | IAM | Install | Terraform | HashiC aws_s3_bucket_object | Re +
[Alt+S] https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-02a8a4b5 N. Virginia keerthana
Services Search
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-02a8a4b5c4a0363f (ec2-s3)
PublicIPs: 54.242.240.245 PrivateIPs: 172.31.26.240
```

- Now we can see the file in the bucket.

The screenshot shows the AWS S3 console interface. On the left, a sidebar titled 'Amazon S3' contains links for Buckets, Storage Lens, and other AWS services. The main area is titled 'Objects (1) Info' and displays a single object named 'file.txt'. The object details are as follows:

Name	Type	Last modified	Size	Storage class
file.txt	txt	March 27, 2024, 21:26:28 (UTC+05:30)	0 B	Standard

At the bottom of the browser window, the Windows taskbar is visible, showing icons for CloudShell, Feedback, Start button, Task View, File Explorer, Edge, File Explorer, Mail, and Task Manager.

