



ASSIGNMENT_2

Name:-D. Sai Rajitha

Email-id:- sairajithavarma@gmail.com

Course:- Devops

Trainer:- Madhukar Reddy

Batch:- 120-5pm

Step1:- Create an instance

- Name tag given as “terraform”
- Select AMI “ubuntu”
- Create keypair as “terraform”
- Launch instance

Screenshot of the AWS EC2 Launch an instance page.

Name and tags

Name: terraform

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

Quick Start

Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux Enterprise Server

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ami-09a7535106fb42d5

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2

Launch instance

Screenshot of the AWS EC2 Launch an instance page, showing the selection of an AMI.

Quick Start

Amazon Linux, macOS, Ubuntu, Windows, Red Hat, SUSE Linux Enterprise Server

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
ami-09a7535106fb42d5 (64-bit (x86)) / ami-de4b1df799f55bb8b (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-09a7535106fb42d5 Verified provider

Instance type

Free tier eligible

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ami-09a7535106fb42d5

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

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

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2

Launch instance

The screenshot shows two consecutive screenshots of the AWS CloudShell interface.

Screenshot 1: Create key pair

- Key pair name:** terraform
- Key pair type:** RSA (selected)
- Private key file format:** .pem (selected)
- A warning message: "When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)"
- Create key pair** button

Screenshot 2: Instances (1/1) Info

- Instances:** terraform (i-01dceb956db871aca)
- Status:** Running
- Type:** t2.micro
- State:** Initializing
- Availability Zone:** ap-northeast-2c
- Public IP:** ec2-43-200-112-144

Instance: i-01dceb956db871aca (terraform)

- Details:** Platform: Ubuntu (Inferred), AMI ID: ami-09a7555106fb42d5, Monitoring: disabled, AMI name: ubuntu/images/hvm-ssd/ubuntu-jammy-22.04-amd64-server-20240301, Launch time: 2024-03-01T10:45:00Z, AMI location: ap-northeast-2a
- Status and alarms:** New
- Monitoring:** None
- Security:** None
- Networking:** None
- Storage:** None
- Tags:** None

Step2:- Connect to instance

- Sudo -l to change normal user to root user
- apt update -y
- Launch instance

The screenshot shows the AWS EC2 Instance Connect interface. At the top, it displays the instance ID: i-01dceb956db871aca (terraform). Below this, there are two connection options: "Connect using EC2 Instance Connect" (selected) and "Connect using EC2 Instance Connect Endpoint". The "Public IP address" is listed as 43.201.115.75. The "Username" field contains "ubuntu". 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." At the bottom right are "Cancel" and "Connect" buttons.

CloudShell Feedback

The screenshot shows the AWS CloudShell terminal. It displays system information as of Mar 26 2024, including:

- System load: 0.7509765625
- Processes: 101
- Usage of /: 20.4% of 7.57GB
- Users logged in: 0
- Memory usage: 21%
- IPv4 address for eth0: 172.31.41.245
- Swap usage: 0%

It also shows that expanded security maintenance for applications is not enabled, and 0 updates can be applied immediately. It encourages enabling ESM Apps for future updates and provides a link to https://ubuntu.com/esm. The terminal then lists available updates, noting they are more than a week old and encouraging users to run sudo apt update. It ends with a note about the programs included being free software and the lack of warranty, followed by a root prompt: "ubuntu@ip-172-31-41-245:~\$ sudo -i". The public IP is listed as 43.201.115.75 and the private IP as 172.31.41.245.

CloudShell Feedback

```

System load: 0.7509765625 Processes: 101
Usage of /: 20.4% of 7.57GB Users logged in: 0
Memory usage: 21% IPv4 address for eth0: 172.31.41.245
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

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-172-31-41-245:~$ sudo -i
root@ip-172-31-41-245:~# apt update -y
i-01dceb956db871aca (terraform)
PublicIPs: 43.201.115.75 PrivateIPs: 172.31.41.245

```



```

CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
Instances | E... EC2 Instance EC2 Instance Users | IAM Install | Terraform Terraform hashicorp/ai... Install | Terraform Install | Terraform Terraform T...
← → ⌂ ap-northeast-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01dceb956db871aca&osUser=ubuntu&region=ap-northeast-2&sshPort...
Seoul rajitha
aws Services Search [Alt+S]
Get:10 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1505 kB]
Get:12 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [290 kB]
Get:13 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1628 kB]
Get:14 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [273 kB]
Get:15 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1059 kB]
Get:16 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [240 kB]
Get:17 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.1 kB]
Get:18 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [42.4 kB]
Get:19 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [1.1 kB]
Get:20 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [472 B]
Get:21 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [67.1 kB]
Get:22 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.0 kB]
Get:23 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:24 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:25 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [28.4 kB]
Get:26 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.2 kB]
Get:27 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [644 B]
Get:28 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:29 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1290 kB]
Get:30 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [230 kB]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1600 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [268 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [851 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [162 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.1 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7476 B]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
93% [11 Packages store 0 B]

i-01dceb956db871aca (terraform)
PublicIPs: 43.201.115.75 PrivateIPs: 172.31.41.245

```



```

CloudShell Feedback
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
Instances | E... EC2 Instance EC2 Instance Users | IAM Install | Terraform Terraform hashicorp/ai... Install | Terraform Install | Terraform Terraform T...
Seoul rajitha
aws Services Search [Alt+S]

```

Step3:- Create an IAM user

- Go to IAM
- Click on create user
- User name here I am giving user name as “user1”
- Attach policies here I gave administrative access
- Click on create user

The screenshot shows the AWS IAM search results for the term 'iam'. The search bar at the top contains 'Search results for 'iam'' and the search term 'iam'. Below the search bar, there are two main sections: 'Services (11)' and 'Features (22)'. The 'Services' section includes links to IAM, Marketplace, AWS App Mesh, and other services like CloudWatch Metrics and Lambda. The 'Features' section includes links to Groups, Roles, and IAM features. A context menu is open over the 'See all 11 results' link in the 'Services' section, showing options such as 'Open link in new tab', 'Open link in new window', 'Open link in incognito window', 'Open link as...', 'Save link as...', 'Copy link address', and 'Inspect'.

The screenshot shows the AWS IAM Users page. The left sidebar navigation includes 'Identity and Access Management (IAM)', 'Dashboard', 'Access management' (with 'User groups', 'Users', 'Roles', 'Policies', 'Identity providers', and 'Account settings' listed), 'Access reports' (with 'Access Analyzer', 'External access', 'Unused access', 'Analyzer settings', 'Credential report', 'Organization activity', and 'Service control policies (SCPs)'), and 'CloudShell' and 'Feedback' buttons. The main content area is titled 'Users (0) Info' and displays a message: 'An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.' It includes a search bar, a table header with columns for 'User name', 'Path', 'Group', 'Last activity', 'MFA', 'Password age', and 'Console last sign-in', and a message 'No resources to display'.

Screenshot of the AWS IAM User Creation Step 2: Set permissions page.

User details:

- User name: user1
- Provide user access to the AWS Management Console - optional:

Are you providing console access to a person?

User type:

- 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 Keyspaces, 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.

Permissions policies (1/1182):

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

Filter by Type: All types

Policy name	Type	Attached entities
<input checked="" type="checkbox"/> AdministratorAccess	AWS managed - job function	0
<input type="checkbox"/> AdministratorAccess-Amplify	AWS managed	0
<input type="checkbox"/> AdministratorAccess-AWSElasticBea...	AWS managed	0
<input type="checkbox"/> AmazonAPIGatewayAdministrator	AWS managed	0
<input type="checkbox"/> AmazonNimbleStudio-StudioAdmin	AWS managed	0
<input type="checkbox"/> AmazonSageMakerAdmin-ServiceC...	AWS managed	0
<input type="checkbox"/> AmazonSecurityLakeAdministrator	AWS managed	0
<input type="checkbox"/> AmazonWorkSpacesAdmin	AWS managed	0

The screenshot shows the 'Review and create' step of the AWS IAM user creation wizard. On the left, a sidebar lists steps: Step 1 (Specify user details), Step 2 (Set permissions), Step 3 (Review and create), and Step 4 (Retrieve password). The main area displays 'User details' for a user named 'user1'. It shows the console password type as 'Autogenerated' and the 'Require password reset' option set to 'No'. Below this is the 'Permissions summary' table, which contains one item: 'AdministratorAccess' (AWS managed - job function). The table has columns for Name, Type, and Used as. The 'Tags - optional' section indicates 'No tags associated with the resource' and provides an 'Add new tag' button. At the bottom right are 'Cancel', 'Previous', and 'Create user' buttons.

The screenshot shows the 'user1' user details page in the AWS IAM console. The left sidebar shows the IAM navigation path: IAM > Users > user1. The main area is titled 'user1 info' and includes a 'Summary' section with ARN (arn:aws:iam::058264148783:user/user1), Console access status (Enabled without MFA), and Access key count (1). Below this is a 'Permissions' tab showing 'Permissions policies (1)'. A table lists the attached policy: 'AdministratorAccess' (AWS managed - job function, Attached via Directly). The table has columns for Policy name, Type, and Attached via.

Step4:- To create access and secret access key

- Click on user
- Click on create accesskey
- Click on command line interface
- Description tag “123456” this can be anything
- Aws configure
- Copy the access and secret access key from user
- Region “ap-northeast-2”

- Default output should be table

The screenshot shows the AWS IAM 'Create access key' wizard. The current step is 'Access key best practices & alternatives'. The page title is 'Access key best practices & alternatives' with an 'Info' link. A note below says 'Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.' On the left, there's a sidebar with 'Step 1: Access key best practices & alternatives', 'Step 2 - optional: Set description tag', and 'Step 3: Retrieve access keys'. The main content area is titled 'Use case' and contains several 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.
- Application running outside AWS
You plan to use this access key to authenticate workloads running in your data center or other infrastructure outside of AWS that needs to access your AWS resources.
- Other
Your use case is not listed here.

At the bottom, there's a 'Alternatives recommended' section with a warning icon:

- 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](#)

A confirmation box at the bottom asks if the user understands the recommendation and wants to proceed to create an access key. There are 'Cancel' and 'Next Step' buttons.

The screenshot shows the AWS IAM 'Create access key' wizard. The current step is 'Step 2 - optional: Set description tag'. A text input field contains the value '123456'. Below the input field is a note: 'Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . / = + ~ @'. At the bottom right are 'Cancel', 'Previous', and 'Create access key' buttons.

Set description tag - optional

The description for this access key will be attached to this user as a tag and shown alongside the access key.

Description tag value

Describe the purpose of this access key and where it will be used. A good description will help you rotate this access key confidently later.

123456

Maximum 256 characters. Allowed characters are letters, numbers, spaces representable in UTF-8, and: _ . / = + ~ @

Cancel Previous Create access key

The screenshot shows the AWS CloudShell terminal window. The output of the 'aws configure' command is displayed, including the configuration for the standard instance connect connection. The terminal shows the host IP (43.201.115.75) and private IP (172.31.41.245).

```
update-alternatives: using /usr/bin/stream-im6.q16 to provide /usr/bin/stream-im6 (stream-im6) in auto mode
update-alternatives: using /usr/bin/display-im6.q16 to provide /usr/bin/display (display) in auto mode
update-alternatives: using /usr/bin/display-im6.q16 to provide /usr/bin/display-im6 (display-im6) in auto mode
update-alternatives: using /usr/bin/montage-im6.q16 to provide /usr/bin/montage (montage) in auto mode
update-alternatives: using /usr/bin/montage-im6.q16 to provide /usr/bin/montage-im6 (montage-im6) in auto mode
update-alternatives: using /usr/bin/mogrify-im6.q16 to provide /usr/bin/mogrify (mogrify) in auto mode
update-alternatives: using /usr/bin/mogrify-im6.q16 to provide /usr/bin/mogrify-im6 (mogrify-im6) in auto mode
Setting up imagemagick (8:6.9.11.60+dfsg-1.3ubuntu0.22.04.3) ...
Processing triggers for install-info (6.8-4build1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for shared-mime-info (2.1-2) ...
Processing triggers for sgml-base (1.30) ...
Setting up docutils-common (0.17.1+dfsg-2) ...
Processing triggers for sgml-base (1.30) ...
Setting up python3-docutils (0.17.1+dfsg-2) ...
Setting up awscli (1.22.34-1) ...
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-172-31-41-245:~# aws configure
```

i-01dceb956db871aca (terraform)

PublicIPs: 43.201.115.75 PrivateIPs: 172.31.41.245

The screenshot shows the AWS CloudShell terminal window. It displays the host information, including the IP address (43.201.115.75), private IP (172.31.41.245), and the fact that the host is running the AWS Lambda environment.

```
CloudShell Feedback
```

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

i-01dceb956db871aca (lambda)

PublicIPs: 43.201.115.75 PrivateIPs: 172.31.41.245

The screenshot shows the AWS IAM 'Access key created' page. It displays the copied access key ID (AKIAQ3EGQLMXRHZUGAQ) and its corresponding secret access key (represented by a series of asterisks). A green banner at the top states: 'Access key created. This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.' Below this, there's a section titled 'Access key best practices' with several recommendations.

```

update-alternatives: using /usr/bin/mogrify-im6.q16 to provide /usr/bin/mogrify-im6 (mogrify-im6) in auto mode
Setting up imagemagick (8:6.5.11.60+dfsg-1.3ubuntu0.22.04.3) ...
Processing triggers for install-info (6.8-4build1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for shared-mime-info (2.1-2) ...
Processing triggers for sgml-base (1.30) ...
Setting up docutils-common (0.17.1+dfsg-2) ...
Processing triggers for sgml-base (1.30) ...
Setting up python3-docutils (0.17.1+dfsg-2) ...
Setting up awscli (1.22.34-1) ...
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-172-31-41-245:~# aws configure
AWS Access Key ID [None]: AKIAQ3EGQLMXRHZUGAQ
AWS Secret Access Key [None]: LzvMj7DYYXhEP+8hmtonEO5b/wRY0K4jT/43gc9
Default region name [None]: ap-northeast-2
Default output format [None]: table

```

i-01dceb956db871aca (terraform)
PublicIPs: 43.201.115.75 PrivateIPs: 172.31.41.245

Step5:- install terraform to install terraform

- Go to the browser search terraform install in ubuntu
- Copy and paste the command
- Terraform will be installed
- Create a directory mkdir “terraform”
- We have to create the terraform block where we can see the info about version etc “vi terraformblock.tf” tf is the extension of terraform file
- We get code from registry for terraformblock file

```

terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.44.0"
    }
  }
}

```

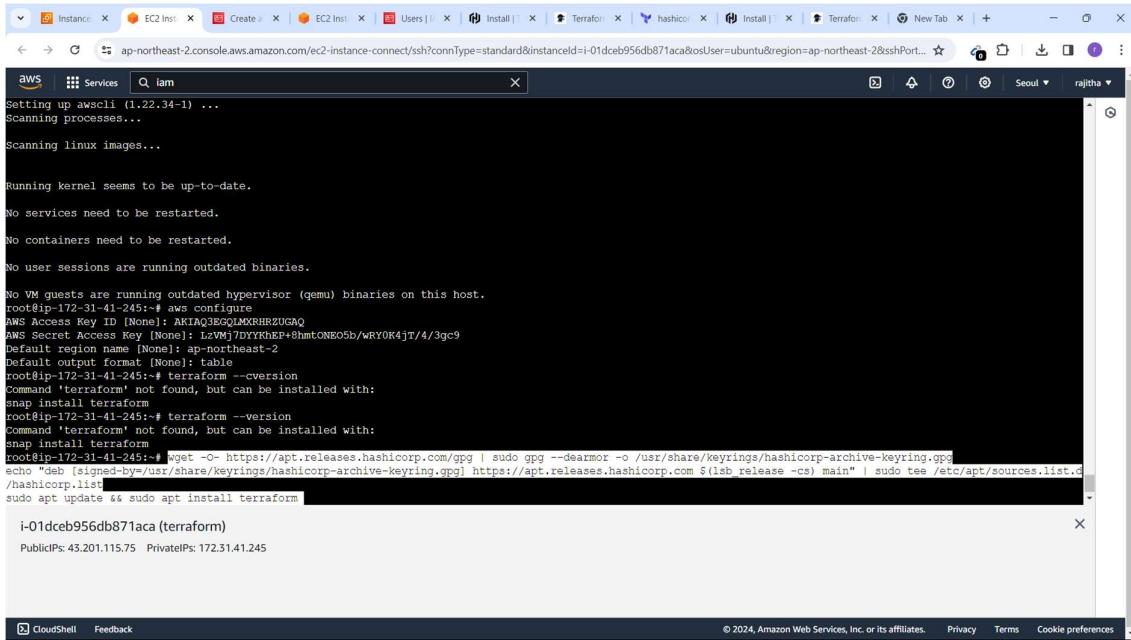
- Now we have to create provider file where we get all info who and what is providing and also in which region “vi provider.tf”

```

provider "aws" {
  profile = "default"
  region = "ap-northeast-1"
}

```

- After that the main resource block in this we have all info about what resource we are creating “vi resource.tf”



The screenshot shows a CloudShell terminal window with multiple tabs at the top. The active tab displays a terminal session on an EC2 instance. The session starts with AWS CLI initialization and then proceeds to install Terraform via apt-get. The output shows the download of the GPG key and the addition of the HashiCorp repository to /etc/apt/sources.list.d/.

```

Setting up awsccli (1.22.34-1) ...
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-172-31-41-245:~# aws configure
AWS Access Key ID [None]: AKIAQ3E3GQIMXKRH2UGAQ
AWS Secret Access Key [None]: LzWmj7DYYRhP+8mtCNEObwRYOK4jT/4/3gc9
Default region name [None]: ap-northeast-2
Default output format [None]: table
root@ip-172-31-41-245:~# terraform --version
Command 'terraform' not found, but can be installed with:
snap install terraform
root@ip-172-31-41-245:~# terraform --version
Command 'terraform' not found, but can be installed with:
snap install terraform
root@ip-172-31-41-245:~# 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
i-01dceb956db871aca (terraform)
PublicIPs: 43.201.115.75 PrivateIPs: 172.31.41.245

```

aws Services Search: iam

```

Length: 3980 (3.9K) [binary/octet-stream]
Saving to: 'STDOUT'

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

2024-03-26 13:29:05 (1.34 GB/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://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-northeast-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://apt.releases.hashicorp.com jammy InRelease [12.9 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 https://apt.releases.hashicorp.com jammy/main amd64 Packages [123 kB]
Fetched 246 kB in 1s (205 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
19 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  terraform
0 upgraded, 1 newly installed, 0 to remove and 19 not upgraded.
Need to get 26.9 MB of archives.
After this operation, 94.5 MB of additional disk space will be used.
[Waiting for headers]
```

i-01dceb956db871aca (terraform)

PublicIPs: 43.201.115.75 PrivateIPs: 172.31.41.245

CloudShell Feedback

aws Services Search: iam

```

required_providers {
    aws = {
        source = "hashicorp/aws"
        version = "5.42.0"
    }
}

-- INSERT --
11
```

i-01dceb956db871aca (terraform)

PublicIPs: 43.201.115.75 PrivateIPs: 172.31.41.245

CloudShell Feedback

```
aws Services Q iam
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-172-31-41-245:~# mkdir terraform
root@ip-172-31-41-245:~# cd terraform
root@ip-172-31-41-245:~/terraform# vi terraformblock.tf
root@ip-172-31-41-245:~/terraform# vi provider.tf

i-01dceb956db871aca (terraform)
PublicIPs: 43.201.115.75 PrivateIPs: 172.31.41.245

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

aws Services Q iam
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-172-31-41-245:~# mkdir terraform
root@ip-172-31-41-245:~# cd terraform
root@ip-172-31-41-245:~/terraform# vi terraformblock.tf
root@ip-172-31-41-245:~/terraform# vi provider.tf
root@ip-172-31-41-245:~/terraform# vi resource.tf

i-01dceb956db871aca (terraform)
PublicIPs: 43.201.115.75 PrivateIPs: 172.31.41.245

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

```

aws Services Search [Alt+S] rajitha
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro

System information as of Tue Mar 26 13:22:52 UTC 2024
System load: 0.7509765625 Processes: 101
Usage of /: 20.4% of 7.57GB Users logged in: 0
Memory usage: 21% IPv4 address for eth0: 172.31.41.245
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.
24 updates can be applied immediately.
15 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
7 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Tue Mar 26 13:22:53 2024 from 13.209.1.60
ubuntu@ip-172-31-41-245:~$ sudo -i
root@ip-172-31-41-245:~# ls
snap  terraform
root@ip-172-31-41-245:~# cd terraform
root@ip-172-31-41-245:~/terraform# ls
provider.tf  terraformlock.tf
root@ip-172-31-41-245:~/terraform# vi hello.txt
root@ip-172-31-41-245:~/terraform# i-01dceb956db871aca (terraform)
PublicIPs: 3.35.37.91 PrivateIPs: 172.31.41.245

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

```

Step6:- here we are creating s3 bucket and in that bucket we are also uploading the file the resource we will be in the resource block. So to create the file we should have a file

- Vi hello.txt
- Enter the data here I am giving “hi all! Good evening”
- File got created
- The path will be root/terraform/hello.txt
- The code for resource and file uploading I have given in the resource block

```

resource "aws_s3_bucket" "bucket" {
  bucket = "rajitha123456"

  tags = {
    Name = "Rajitha"
  }
}

resource "aws_s3_bucket_object" "file" {
  bucket = aws_s3_bucket.bucket.id

```

```

key  = "hello.txt"

source = "/root/terraform/hello.txt"

}

```

```

hi all! Good evening

-- INSERT --
i-01dceb956db871aca (terraform)
Public IPs: 3.35.37.91 Private IPs: 172.31.41.245

Management: https://landscape.canonical.com
Support: https://ubuntu.com/pro

System information as of Tue Mar 26 13:22:52 UTC 2024

System load: 0.7509765625 Processes: 101
Usage of /: 20.4% of 7.57GB Users logged in: 0
Memory usage: 21% IPV4 address for eth0: 172.31.41.245
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

24 updates can be applied immediately.
15 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

7 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Tue Mar 26 13:22:53 2024 from 13.209.1.60
ubuntu@ip-172-31-41-245:~$ sudo -i
root@ip-172-31-41-245:~# ls
snap_ terraform
root@ip-172-31-41-245:~# cd terraform
root@ip-172-31-41-245:~/terraform# ls
provider.tf terraformblock.tf
root@ip-172-31-41-245:~/terraform# vi hello.txt
root@ip-172-31-41-245:~/terraform# vi resource.tf
i-01dceb956db871aca (terraform)
Public IPs: 3.35.37.91 Private IPs: 172.31.41.245

```

Step7:- for intilization we use

- **terraform init**

```

aws | Services | Search | [Alt+S]
24 updates can be applied immediately.
15 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

7 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Tue Mar 26 13:22:53 2024 from 13.209.1.60
ubuntu@ip-172-31-41-245:~$ sudo -i
root@ip-172-31-41-245:~# ls
snap_terraform
root@ip-172-31-41-245:~# cd terraform
root@ip-172-31-41-245:~/terraform# ls
provider.tf  terraformlock.tf
root@ip-172-31-41-245:~/terraform# vi hello.txt
root@ip-172-31-41-245:~/terraform# vi resource.tf
root@ip-172-31-41-245:~/terraform# rm resource.tf
root@ip-172-31-41-245:~/terraform# vi resource.tf
root@ip-172-31-41-245:~/terraform# ls
hello.txt  provider.tf  resource.tf  terraformblock.tf
root@ip-172-31-41-245:~/terraform# vi resource.tf
root@ip-172-31-41-245:~/terraform# terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.42.0"...
- Installing hashicorp/aws v5.42.0...

i-01dceb956db871aca (terraform)
PublicIPs: 3.35.37.91 PrivateIPs: 172.31.41.245

```

Step8:- for apply we use

- **terraform apply --auto-approve**

```

aws | Services | Search | [Alt+S]
root@ip-172-31-41-245:~/terraform# vi resource.tf
root@ip-172-31-41-245:~/terraform# rm resource.tf
root@ip-172-31-41-245:~/terraform# vi resource.tf
root@ip-172-31-41-245:~/terraform# ls
hello.txt  provider.tf  resource.tf  terraformblock.tf
root@ip-172-31-41-245:~/terraform# vi resource.tf
root@ip-172-31-41-245:~/terraform# terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.42.0"...
- 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!

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,
run this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
root@ip-172-31-41-245:~/terraform# terraform apply --auto-approve

i-01dceb956db871aca (terraform)
PublicIPs: 3.35.37.91 PrivateIPs: 172.31.41.245

```

```

aws s3 bucket bucket: Creating...
aws s3 bucket bucket: Creation complete after 1s [id=rajitha123456]
aws s3 bucket_object file: Creating...
aws s3 bucket_object file: Creation complete after 0s [id=hello.txt]

Warning: Deprecated Resource
with aws_s3_bucket_object.file,
on resource.tf line 8, in resource "aws_s3_bucket_object" "file":
  8: resource "aws_s3_bucket_object" "file" {

use the aws_s3_object resource instead
(and 2 more similar warnings elsewhere)

Warning: Argument is deprecated
with aws_s3_bucket_object.file,
on resource.tf line 9, in resource "aws_s3_bucket_object" "file":
  9:   bucket = aws_s3_bucket.bucket.id

Use the aws_s3_object resource instead
(and 3 more similar warnings elsewhere)

apply complete! Resources: 2 added, 0 changed, 0 destroyed.
root@ip-172-31-41-245:~/terraform#

```

i-01dceb956db871aca (terraform)
PublicIPs: 3.35.37.91 PrivateIPs: 172.31.41.245

Step9:- As you can see bucket created with the name “rajitha123456” and file got uploaded in that bucket “hello.txt”

Name	AWS Region	Access	Creation date
rajitha123456	Asia Pacific (Seoul) ap-northeast-2	Bucket and objects not public	March 26, 2024, 20:03:57 (UTC+05:30)

The screenshot shows the AWS S3 console interface. On the left, a sidebar navigation bar includes links for 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 (expanded), Dashboards, Storage Lens groups, AWS Organizations settings, Feature spotlight, and AWS Marketplace for S3. The main content area displays the 'Objects (1) Info' section for the bucket 'rajitha123456'. It shows a single object named 'hello.txt' with a type of 'txt'. The object was last modified on March 26, 2024, at 23:37:56 (UTC+05:30) and has a size of 21.0 B, categorized under Standard storage class. Below the object list is a search bar labeled 'Find objects by prefix' and a toolbar with actions like Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload.

By this we can say with the use of terraform we can create resources in the form of code