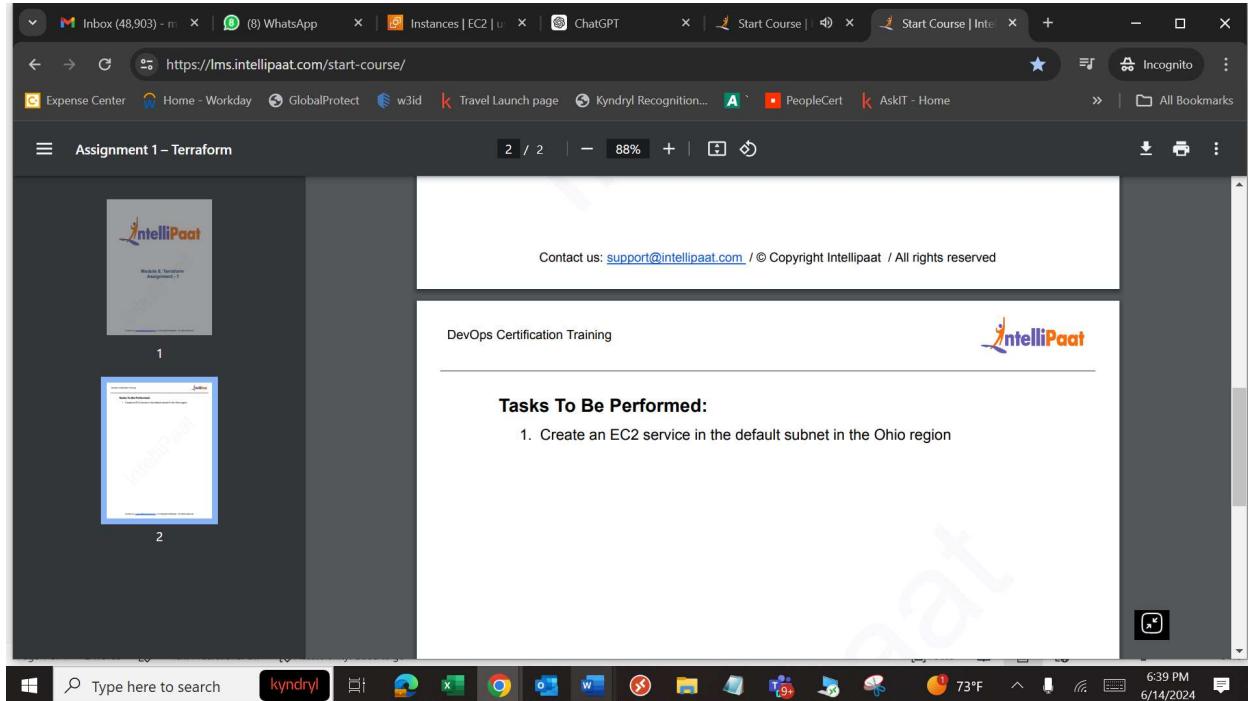


## Terraform assignments

### Assignment 1



Solution:

EC2 > Instances > Launch an instance

## Launch an instance Info

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 Info**

Name  Add additional tags

**Application and OS Images (Amazon Machine Image) Info**

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

Recents Quick Start

Amazon macOS Ubuntu Windows Red Hat SUSE Li

**Summary**

Number of instances Info

Software Image (AMI)  
Canonical, Ubuntu, 22.04 LTS, ...read more  
ami-0e001c9271cf7f3b9

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
devopssecuritygroup

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)

Cancel **Launch instance** Review commands

© 2024, Amazon Web Services,

This screenshot shows the 'Launch an instance' wizard in the AWS Management Console. It starts with a 'Name and tags' section where the name 'tf' is entered. Below it is the 'Application and OS Images (Amazon Machine Image)' section, which includes a search bar and a 'Quick Start' tab selected. A summary panel on the right lists the instance configuration: 1 instance, Canonical Ubuntu 22.04 LTS AMI, t2.micro instance type, devopssecuritygroup security group, and 1 volume (8 GiB). A tooltip for the free tier is displayed over the summary panel. At the bottom are 'Cancel', 'Launch instance' (in orange), and 'Review commands' buttons.

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

Recents    Quick Start

Amazon Linux    macOS    Ubuntu    Windows    Red Hat    SUSE Li

Browse more AMIs  
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type    Free tier eligible

ami-0e001c9271cf7f3b9 (64-bit (x86)) / ami-058b428b3b45defec (64-bit (Arm))  
Virtualization: hvm   ENA enabled: true   Root device type: ebs

Description  
Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2024-04-11

Architecture    AMI ID  
64-bit (x86)    ami-0e001c9271cf7f3b9    Verified provider

▼ Instance type [Info](#) | [Get advice](#)

Instance type

Shell    Feedback

**t2.micro**

Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.0162 USD per Hour  
On-Demand SUSE base pricing: 0.0116 USD per Hour  
On-Demand RHEL base pricing: 0.0716 USD per Hour  
On-Demand Linux base pricing: 0.0116 USD per Hour

Free tier eligible

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

▼ 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*

devops

Create new key pair

▼ Network settings [Info](#)

VPC - *required* | [Info](#)

vpc-031a983d5066ebe3b (default)  
172.31.0.0/16

Subnet | [Info](#)

No preference

Create new subnet

vpc-031a983d5066ebe3b (default) ▾

Subnet [Info](#)

No preference [Create new subnet](#)

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of [free tier allowance](#)

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group  Select existing security group

Common security groups [Info](#)

Select security groups [Compare security group rules](#)

devopssecuritygroup sg-0bb9c53a77e8cc1eb X  
VPC: vpc-031a983d5066ebe3b

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▼ Configure storage [Info](#) Advanced

1x  GiB  Root volume (Not encrypted)

[Help](#) [Feedback](#)

▼ Configure storage [Info](#)

Advanced

1x 8 GiB gp2 ▼ Root volume (Not encrypted)

i Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage X

[Add new volume](#)

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

i Click refresh to view backup information ↻  
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems [Edit](#)

► Advanced details [Info](#)

Instances (1/1) <a href="#">Info</a>										
<input type="text"/> Find Instance by attribute or tag (case-sensitive)										
<input checked="" type="checkbox"/>	Name <a href="#">▼</a>	Instance ID	Instance state <a href="#">▼</a>	Instance type <a href="#">▼</a>	Status check <a href="#">▼</a>	Alarm status	Availability Zone <a href="#">▼</a>	Public IPv4 DNS <a href="#">▼</a>	Public IP	Launch instances <a href="#">▼</a>
<input checked="" type="checkbox"/>	tf	i-0ab930563562f7cb6	<span style="color: green;">Running</span> <a href="#">▼</a> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	<span style="color: green;">Initializing</span> <a href="#">▼</a>	<a href="#">View alarms</a> <a href="#">+</a>	us-east-1a	ec2-52-91-173-3.comp...	52.91.17	<a href="#">Launch instances</a> <a href="#">▼</a>

```
Command Prompt - ssh -i devops.pem ubuntu@52.91.173.3
Microsoft Windows [Version 10.0.19045.4412]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Manohar>cd downloads

C:\Users\Manohar\Downloads>ssh -i devops.pem ubuntu@52.91.173.3
The authenticity of host '52.91.173.3 (52.91.173.3)' can't be established.
ECDSA key fingerprint is SHA256:pVrQAt3nVmAYKqoyp3jP0Wz+nUPnanHvBz5TsxDew04.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

```
ubuntu@ip-172-31-27-252: ~
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-172-31-27-252:~$
```

```
ubuntu@ip-172-31-27-252:~$ sudo hostnamectl set-hostname terraform
ubuntu@ip-172-31-27-252:~$ exit
logout
Connection to 52.91.173.3 closed.
```

```
C:\Users\Manohar\Downloads>ssh -i devops.pem ubuntu@52.91.173.3
```

### Update the instance

```
sudo apt update
```

```
ubuntu@terraform:~$ sudo apt update
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 [128 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1517 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [259 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1933 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [329 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [858 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [166 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.2 kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7588 B]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1731 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [318 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1990 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [338 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1086 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [251 kB]
```

### Install terraform

```
sudo apt-get update && sudo apt-get install -y gnupg software-properties-common
```

```
ubuntu@terraform:~$ sudo apt-get update && sudo apt-get install -y gnupg software-properties-common
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 [128 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1731 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1086 kB]
Fetched 2945 kB in 1s (2222 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.2.27-3ubuntu2.1).
software-properties-common is already the newest version (0.99.22.9).
0 upgraded, 0 newly installed, 0 to remove and 49 not upgraded.
ubuntu@terraform:~$
```

```
wget -O- https://apt.releases.hashicorp.com/gpg | \
gpg --dearmor | \
sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null
```

```
ubuntu@terraform:~$ wget -O- https://apt.releases.hashicorp.com/gpg | \
> gpg --dearmor | \
> sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null
--2024-06-14 13:58:13-- https://apt.releases.hashicorp.com/gpg
Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com)... 99.84.108.3, 99.84.108.36, 99.84.108.40, ...
Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com)|99.84.108.3|: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

2024-06-14 13:58:13 (1.67 GB/s) - written to stdout [3980/3980]

ubuntu@terraform:~$
```

```
gpg --no-default-keyring \
--keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg \
--fingerprint
```

```
ubuntu@terraform:~$ gpg --no-default-keyring \
> --keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg \
> --fingerprint
gpg: directory '/home/ubuntu/.gnupg' created
gpg: /home/ubuntu/.gnupg/trustdb.gpg: trustdb created
/usr/share/keyrings/hashicorp-archive-keyring.gpg
-----
pub    rsa4096 2023-01-10 [SC] [expires: 2028-01-09]
      798A EC65 1542 8C8E 42EE AA16 FCBC A621 E701
uid            [ unknown] HashiCorp Security (HashiCorp Package Signing) <security+packaging@hashicorp.com>
sub    rsa4096 2023-01-10 [S] [expires: 2028-01-09]

ubuntu@terraform:~$
```

```
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
```

```
ubuntu@terraform:~$ echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] \  
> https://apt.releases.hashicorp.com $(lsb_release -cs) main" | \  
> sudo tee /etc/apt/sources.list.d/hashicorp.list  
deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com jammy main  
ubuntu@terraform:~$
```

```
sudo apt update
```

```
ubuntu@terraform:~$ sudo apt update  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Get:5 https://apt.releases.hashicorp.com jammy InRelease [12.9 kB]  
Get:6 https://apt.releases.hashicorp.com jammy/main amd64 Packages [136 kB]  
Fetched 149 kB in 1s (178 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
49 packages can be upgraded. Run 'apt list --upgradable' to see them.  
ubuntu@terraform:~$
```

```
sudo apt-get install terraform
```

```
ubuntu@terraform:~$ sudo apt-get install terraform
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 49 not upgraded.
Need to get 27.7 MB of archives.
After this operation, 88.2 MB of additional disk space will be used.
Get:1 https://apt.releases.hashicorp.com jammy/main amd64 terraform amd64 1.8.5-1 [27.7 MB]
Fetched 27.7 MB in 0s (81.6 MB/s)
Selecting previously unselected package terraform.
(Reading database ... 65283 files and directories currently installed.)
Preparing to unpack .../terraform_1.8.5-1_amd64.deb ...
Unpacking terraform (1.8.5-1) ...
Setting up terraform (1.8.5-1) ...
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.

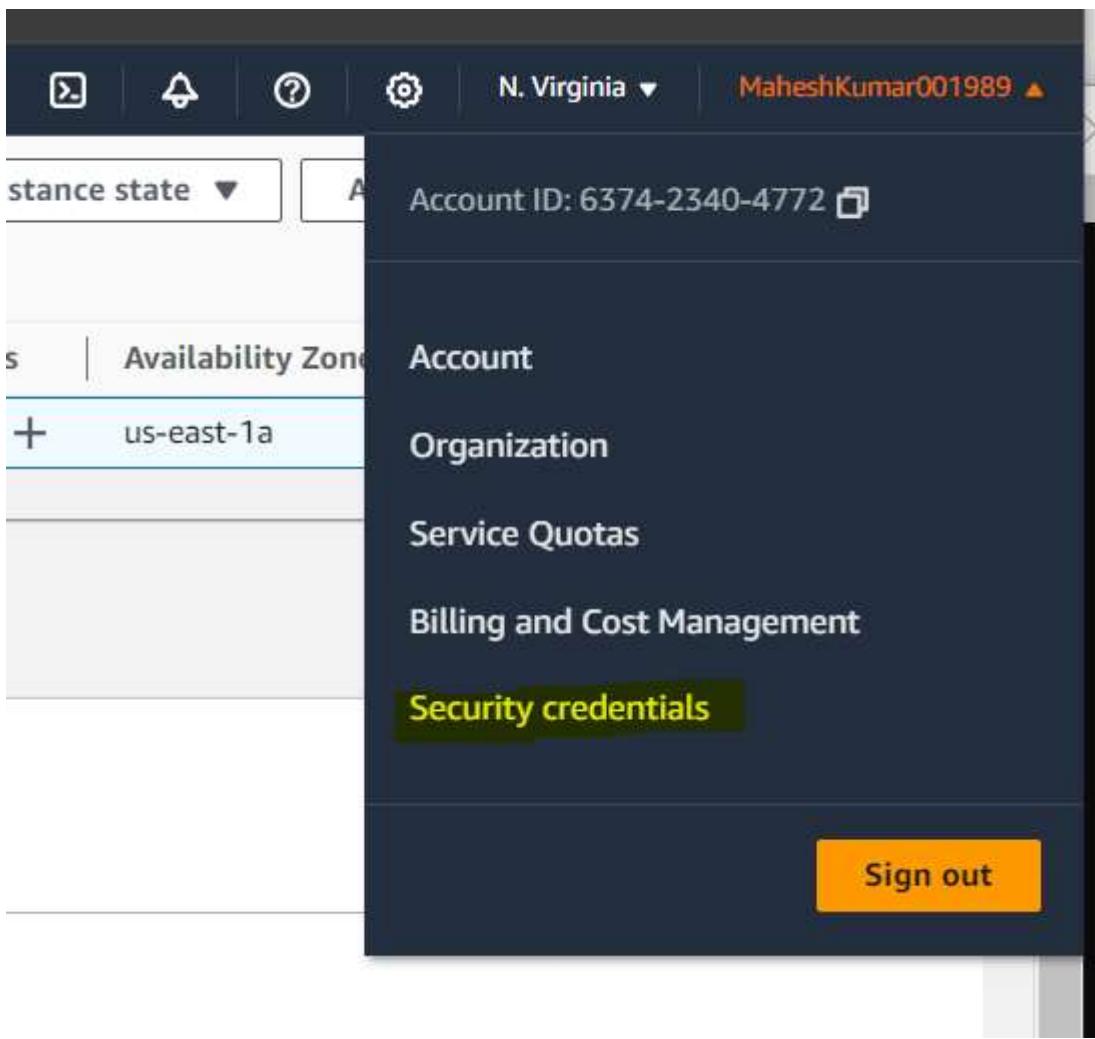
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@terraform:~$
```

terraform --version

```
ubuntu@terraform:~$ terraform --version
Terraform v1.8.5
on linux_amd64
ubuntu@terraform:~$
```

Generate credentials

Access key & secret access key



Access key deleted																									
<input type="checkbox"/> 637423404772	<input type="checkbox"/> e96ab4adc2fb20f1de607e778800acd52bab640e5280a1ad5754e41f837b64a7																								
<b>Multi-factor authentication (MFA) (0)</b>																									
Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. <a href="#">Learn more</a>																									
<table><thead><tr><th>Type</th><th>Identifier</th><th>Certifications</th><th>Created on</th></tr></thead><tbody><tr><td colspan="4">No MFA devices. Assign an MFA device to improve the security of your AWS environment</td></tr><tr><td colspan="4"><a href="#">Assign MFA device</a></td></tr></tbody></table>		Type	Identifier	Certifications	Created on	No MFA devices. Assign an MFA device to improve the security of your AWS environment				<a href="#">Assign MFA device</a>															
Type	Identifier	Certifications	Created on																						
No MFA devices. Assign an MFA device to improve the security of your AWS environment																									
<a href="#">Assign MFA device</a>																									
<b>Access keys (0)</b>																									
Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. <a href="#">Learn more</a>																									
<table><thead><tr><th>Access key ID</th><th>Created on</th><th>Access key last used</th><th>Region last used</th><th>Service last used</th><th>Status</th></tr></thead><tbody><tr><td colspan="6">No access keys</td></tr><tr><td colspan="6">As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. <a href="#">Learn more</a></td></tr><tr><td colspan="6"><a href="#">Create access key</a></td></tr></tbody></table>		Access key ID	Created on	Access key last used	Region last used	Service last used	Status	No access keys						As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. <a href="#">Learn more</a>						<a href="#">Create access key</a>					
Access key ID	Created on	Access key last used	Region last used	Service last used	Status																				
No access keys																									
As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. <a href="#">Learn more</a>																									
<a href="#">Create access key</a>																									

IAM > Security credentials > Create access key

Step 1  
Alternatives to root user access keys

Step 2  
Retrieve access key

## Alternatives to root user access keys Info

**⚠ Root user access keys are not recommended**

We don't recommend that you create root user access keys. Because you can't specify the root user in a permissions policy, you can't limit its permissions, which is a best practice.

Instead, use alternatives such as an IAM role or a user in IAM Identity Center, which provide temporary rather than long-term credentials. [Learn More](#)

If your use case requires an access key, create an IAM user with an access key and apply least privilege permissions for that user. [Learn More](#)

**Continue to create access key?**

I understand creating a root access key is not a best practice, but I still want to create one.

[Cancel](#) [Create access key](#)

IAM > Security credentials > Create access key

Step 1  
Alternatives to root user access keys

Step 2  
Retrieve access key

## Retrieve access key Info

**Access key**

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
<input type="text"/> AKIAZI2LF13GG2NATTZ	<input type="text"/> ***** <a href="#">Show</a>

**Access key best practices**

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the [best practices for managing AWS access keys](#).

[Download .csv file](#) [Done](#)

sudo nano main.tf

```
ubuntu@terraform: ~
GNU nano 6.2                                     main.tf
provider "aws" {
    region = "us-east-2"
    access_key = "AKIAZI2LFI3SGG2NATTZ"
    secret_key = "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"
}
resource "aws_instance" "assignment-1"{
    ami = "ami-0f30a9c3a48f3fa79"
    instance_type = "t2.micro"
    key_name = "devops"
    tags = {
        Name = "assignment-1"
    }
}
```

### Script -1

```
provider "aws" {
    region = "us-east-2"
    access_key = "AKIAZI2LFI3SGG2NATTZ"
    secret_key = "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"
}

resource "aws_instance" "assignment-1"{
    ami = "ami-0f30a9c3a48f3fa79"
    instance_type = "t2.micro"
    key_name = "devops"
    tags = {
        Name = "assignment-1"
    }
}
```

```
ubuntu@terraform:~$ terraform init
```

```
ubuntu@terraform:~$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.54.0...
- Installed hashicorp/aws v5.54.0 (signed by HashiCorp)

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

Terraform has been successfully initialized!

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

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

```
ubuntu@terraform:~$ terraform plan
```

```
c4. ubuntu@terraform: ~
A managed resource "t2" "micro" has not been declared in the root module.

ubuntu@terraform:~$ sudo nano main.tf
ubuntu@terraform:~$ terraform plan

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

Terraform will perform the following actions:

# aws_instance.assignment-1 will be created
+ resource "aws_instance" "assignment-1" {
    + ami                                = "ami-0f30a9c3a48f3fa79"
    + arn                                = (known after apply)
    + associate_public_ip_address        = (known after apply)
    + availability_zone                  = (known after apply)
    + cpu_core_count                     = (known after apply)
    + cpu_threads_per_core              = (known after apply)
    + disable_api_stop                  = (known after apply)
    + disable_api_termination           = (known after apply)
    + ebs_optimized                      = (known after apply)
    + get_password_data                 = false
    + host_id                            = (known after apply)
    + host_resource_group_arn            = (known after apply)
    + iam_instance_profile               = (known after apply)
    + id                                 = (known after apply)
    + instance_initiated_shutdown_behavior = (known after apply)
    + instance_lifecycle                = (known after apply)
    + instance_state                     = (known after apply)
```

```
ubuntu@terraform: ~
+ public_dns          = (known after apply)
+ public_ip           = (known after apply)
+ secondary_private_ips = (known after apply)
+ security_groups     = (known after apply)
+ source_dest_check   = true
+ spot_instance_request_id = (known after apply)
+ subnet_id           = (known after apply)
+ tags
  + "Name" = "assignment-1"
}
+ tags_all            = {
  + "Name" = "assignment-1"
}
+ tenancy              = (known after apply)
+ user_data            = (known after apply)
+ user_data_base64      = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}

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

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.
ubuntu@terraform:~$
```

ubuntu@terraform:~\$ terraform apply

```
ubuntu@terraform: ~
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.
ubuntu@terraform:~$ terraform apply

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

Terraform will perform the following actions:

# aws_instance.assignment-1 will be created
+ resource "aws_instance" "assignment-1" {
  + ami                         = "ami-0f30a9c3a48f3fa79"
  + arn                         = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone           = (known after apply)
  + cpu_core_count               = (known after apply)
  + cpu_threads_per_core        = (known after apply)
  + disable_api_stop             = (known after apply)
  + disable_api_termination      = (known after apply)
  + ebs_optimized                = (known after apply)
  + get_password_data            = false
  + host_id                      = (known after apply)
  + host_resource_group_arn       = (known after apply)
  + iam_instance_profile          = (known after apply)
  + id                           = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle            = (known after apply)
  + instance_state                = (known after apply)
  + instance_type                 = "t2.micro"
  + ipv6_address_count            = (known after apply)
  + ipv6_addresses                = (known after apply)
  + key_name                     = "devops"
  + monitoring                   = (known after apply)
  + outpost_arn                  = (known after apply)
  + password_data                 = (known after apply)
  + placement_group                = (known after apply)
  + placement_partition_number     = (known after apply)
  + primary_network_interface_id  = (known after apply)
  + private_dns                   = (known after apply)
```

```
ubuntu@ubuntu:~$ terraform apply
+ source_dest_check          = true
+ spot_instance_request_id   = (known after apply)
+ subnet_id                  = (known after apply)
+ tags
  + "Name" = "assignment-1"
}
+ tags_all                   = {
  + "Name" = "assignment-1"
}
+ tenancy                    = (known after apply)
+ user_data                  = (known after apply)
+ user_data_base64            = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids     = (known after apply)

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

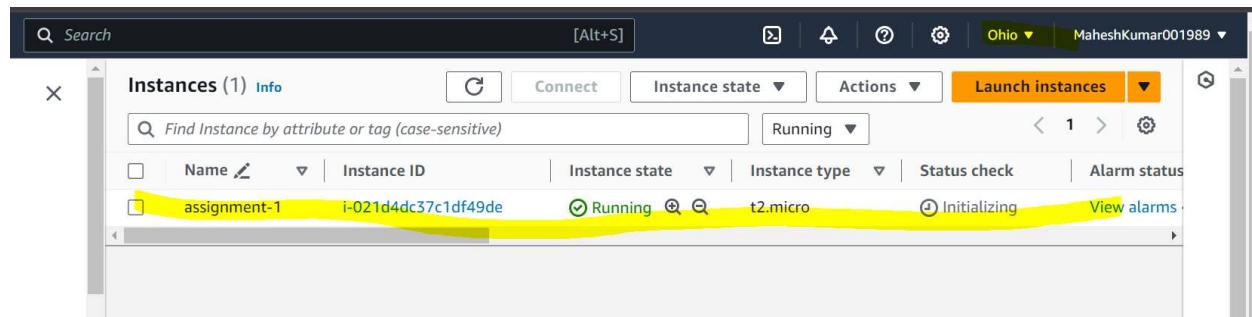
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

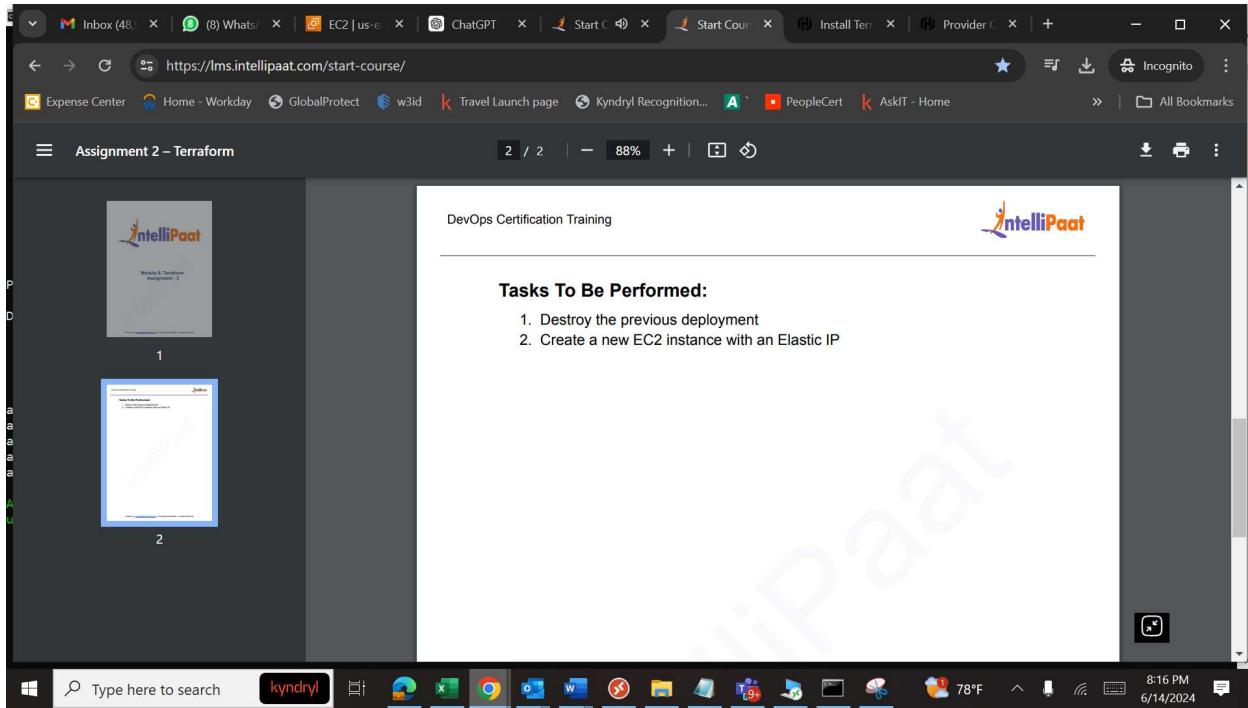
aws_instance.assignment-1: Creating...
aws_instance.assignment-1: Still creating... [10s elapsed]
aws_instance.assignment-1: Still creating... [20s elapsed]
aws_instance.assignment-1: Still creating... [30s elapsed]
aws_instance.assignment-1: Creation complete after 32s [id=i-021d4dc37c1df49de]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
ubuntu@ubuntu:~$
```

Our instance successfully created in the Ohio region.



## Assignment 2



## Solution:

```
ubuntu@terraform:~$ terraform destroy
```

```
ubuntu@terraform:~$ terraform destroy
aws_instance.assignment-1: Still creating... [30s elapsed]
aws_instance.assignment-1: Creation complete after 32s [id=i-021d4dc37c1df49de]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
ubuntu@terraform:~$ terraform destroy
aws_instance.assignment-1: Refreshing state... [id=i-021d4dc37c1df49de]

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

Terraform will perform the following actions:

# aws_instance.assignment-1 will be destroyed
- resource "aws_instance" "assignment-1" {
    - ami = "ami-0f30a9c3a48f3fa79" -> null
    - arn = "arn:aws:ec2:us-east-2:637423404772:instance/i-021d4dc37c1df49de" -> null
    - associate_public_ip_address = true -> null
    - availability_zone = "us-east-2a" -> null
    - cpu_core_count = 1 -> null
    - cpu_threads_per_core = 1 -> null
    - disable_api_stop = false -> null
    - disable_api_termination = false -> null
    - ebs_optimized = false -> null
    - get_password_data = false -> null
    - hibernation = false -> null
    - id = "i-021d4dc37c1df49de" -> null
    - instance_initiated_shutdown_behavior = "stop" -> null
    - instance_state = "running" -> null
    - instance_type = "t2.micro" -> null
    - ipv6_address_count = 0 -> null
    - ipv6_addresses = [] -> null
    - key_name = "devops" -> null
    - monitoring = false -> null
    - placement_partition_number = 0 -> null
    - primary_network_interface_id = "eni-08ade09b17f999392" -> null
    - private_dns = "ip-172-31-15-72.us-east-2.compute.internal" -> null
    - private_ip = "172.31.15.72" -> null
    - public_dns = "ec2-18-222-166-199.us-east-2.compute.amazonaws.com" -> null
    - public_ip = "18.222.166.199" -> null
    - secondary_private_ips = [] -> null
    - security_groups = [
```

```

ubuntu@terraform: ~
  root_block_device {
    - delete_on_termination = true -> null
    - device_name          = "/dev/sda1" -> null
    - encrypted            = false -> null
    - iops                 = 100 -> null
    - tags                 = {} -> null
    - tags_all             = {} -> null
    - throughput           = 0 -> null
    - volume_id            = "vol-03ae4b4f3dddef8008" -> null
    - volume_size           = 8 -> null
    - volume_type           = "gp2" -> null
  }
}

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

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_instance.assignment-1: Destroying... [id=i-021d4dc37c1df49de]
aws_instance.assignment-1: Still destroying... [id=i-021d4dc37c1df49de, 10s elapsed]
aws_instance.assignment-1: Still destroying... [id=i-021d4dc37c1df49de, 20s elapsed]
aws_instance.assignment-1: Still destroying... [id=i-021d4dc37c1df49de, 30s elapsed]
aws_instance.assignment-1: Still destroying... [id=i-021d4dc37c1df49de, 40s elapsed]
aws_instance.assignment-1: Destruction complete after 40s

Destroy complete! Resources: 1 destroyed.
ubuntu@terraform:~$ .

```

Previous instance successfully terminated

Instances (1) <a href="#">Info</a>		<a href="#">C</a>	<a href="#">Connect</a>	<a href="#">Instance state ▾</a>	<a href="#">Actions ▾</a>	<a href="#">Launch instances</a>	<a href="#">☰</a>
<a href="#">Find Instance by attribute or tag (case-sensitive)</a>				<a href="#">Terminated ▾</a>	< 1 >		<a href="#">⚙️</a>
<input type="checkbox"/>	Name <a href="#">✍</a> ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	<a href="#">View alarms</a>
<input type="checkbox"/>	assignment-1	i-021d4dc37c1df49de	<a href="#">Terminated</a> <a href="#">🔍</a> <a href="#">🔍</a>	t2.micro	-		<a href="#">View alarms</a>

ubuntu@terraform:~\$ sudo nano main.tf

```

aws_instance.completer: ResourceSet - destroyed.
ubuntu@terraform:~$ sudo nano main.tf

```

## Script 2

```

provider "aws" {

  region = "us-east-2"

  access_key = "AKIAZI2LFI3SGG2NATTZ"

  secret_key = "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"

}

```

```
resource "aws_instance" "assignment-2"{
  ami = "ami-0f30a9c3a48f3fa79"
  instance_type = "t2.micro"
  key_name = "devops"
  tags = {
    Name = "assignment-2"
  }
}

resource "aws_eip" "eip"{
  domain = "vpc"
}

resource "aws_eip_association" "eip_assoc"{
  instance_id = aws_instance.assignment-2.id
  allocation_id = aws_eip.eip.id
}
```

```
ubuntu@terraform: ~
GNU nano 6.2
provider "aws" {
    region = "us-east-2"
    access_key = "AKIAZI2LFI3SGG2NATTZ"
    secret_key = "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"
}
resource "aws_instance" "assignment-2"{
    ami = "ami-0f30a9c3a48f3fa79"
    instance_type = "t2.micro"
    key_name = "devops"
    tags = {
        Name = "assignment-2"
    }
}
resource "aws_eip" "eip"{
    domain = "vpc"
}
resource "aws_eip_association" "eip_assoc"{
    instance_id = aws_instance.assignment-2.id
    allocation_id = aws_eip.eip.id
}
```

```
ubuntu@terraform:~$ ubuntu@terraform:~$ terraform init
```

```
ubuntu@terraform:~$ ubuntu@terraform:~$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.54.0

Terraform has been successfully initialized!
```

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

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

```
ubuntu@terraform:~$ terraform plan
```

```
ubuntu@terraform:~  
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.  
ubuntu@terraform:~$ sudo nano main.tf  
ubuntu@terraform:~$ terraform plan  
  
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:  
+ create  
  
Terraform will perform the following actions:  
  
# aws_eip.eip will be created  
+ resource "aws_eip" "eip" {  
    + allocation_id      = (known after apply)  
    + arn                = (known after apply)  
    + association_id    = (known after apply)  
    + carrier_ip         = (known after apply)  
    + customer_owned_ip = (known after apply)  
    + domain             = "vpc"  
    + id                 = (known after apply)  
    + instance            = (known after apply)  
    + network_border_group = (known after apply)  
    + network_interface   = (known after apply)  
    + private_dns         = (known after apply)  
    + private_ip          = (known after apply)  
    + ptr_record          = (known after apply)  
    + public_dns          = (known after apply)  
    + public_ip           = (known after apply)  
    + public_ipv4_pool    = (known after apply)  
    + tags_all            = (known after apply)  
    + vpc                 = (known after apply)  
}  
  
# aws_eip_association.eip_assoc will be created  
+ resource "aws_eip_association" "eip_assoc" {  
    + allocation_id      = (known after apply)  
    + id                 = (known after apply)  
    + instance_id        = (known after apply)  
    + network_interface_id = (known after apply)  
    + private_ip_address = (known after apply)  
    + public_ip          = (known after apply)  
}  
844 PM  
6/14/2024
```

```
ubuntu@terraform:~  
+ key_name              = "devops"  
+ monitoring            = (known after apply)  
+ outpost_arn           = (known after apply)  
+ password_data         = (known after apply)  
+ placement_group       = (known after apply)  
+ placement_partition_number = (known after apply)  
+ primary_network_interface_id = (known after apply)  
+ private_dns           = (known after apply)  
+ private_ip             = (known after apply)  
+ public_dns             = (known after apply)  
+ public_ip              = (known after apply)  
+ secondary_private_ips = (known after apply)  
+ security_groups        = (known after apply)  
+ source_dest_check     = true  
+ spot_instance_request_id = (known after apply)  
+ subnet_id              = (known after apply)  
+ tags  
  + "Name" = "assignment-2"  
+ tags_all              = {  
  + "Name" = "assignment-2"  
}  
+ tenancy                = (known after apply)  
+ user_data              = (known after apply)  
+ user_data_base64        = (known after apply)  
+ user_data_replace_on_change = false  
+ vpc_security_group_ids = (known after apply)  
}  
  
Plan: 3 to add, 0 to change, 0 to destroy.  
  
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.  
ubuntu@terraform:~$
```

```
ubuntu@terraform:~$ terraform apply  
844 PM  
6/14/2024
```

```
ubuntu@terraform: ~
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.
ubuntu@terraform:~$ 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_eip.eip will be created
+ resource "aws_eip" "eip" {
  + allocation_id      = (known after apply)
  + arn                = (known after apply)
  + association_id    = (known after apply)
  + carrier_ip         = (known after apply)
  + customer_owned_ip = (known after apply)
  + domain             = "vpc"
  + id                 = (known after apply)
  + instance            = (known after apply)
  + network_border_group = (known after apply)
  + network_interface   = (known after apply)
  + private_dns         = (known after apply)
  + private_ip          = (known after apply)
  + ptr_record          = (known after apply)
  + public_dns          = (known after apply)
  + public_ip           = (known after apply)
  + public_ipv4_pool    = (known after apply)
  + tags_all            = (known after apply)
  + vpc                = (known after apply)
}

# aws_eip_association.eip_assoc will be created
+ resource "aws_eip_association" "eip_assoc" {
  + allocation_id      = (known after apply)
  + id                 = (known after apply)
  + instance_id         = (known after apply)
  + network_interface_id = (known after apply)
  + private_ip_address = (known after apply)
  + public_ip           = (known after apply)
}

Windows PowerShell
Type here to search kyndryl 8:45 PM 6/14/2024
```

```
ubuntu@terraform: ~
+ ami                      = "ami-0f30a9c3a48f3fa79"
+ arn                      = (known after apply)
+ associate_public_ip_address = (known after apply)
+ availability_zone        = (known after apply)
+ cpu_core_count           = (known after apply)
+ cpu_threads_per_core     = (known after apply)
+ disable_api_stop          = (known after apply)
+ disable_api_termination   = (known after apply)
+ ebs_optimized             = (known after apply)
+ get_password_data         = false
+ host_id                  = (known after apply)
+ host_resource_group_arn   = (known after apply)
+ iam_instance_profile      = (known after apply)
+ id                       = (known after apply)
+ instance_initiated_shutdown_behavior = (known after apply)
+ instance_lifecycle        = (known after apply)
+ instance_state            = (known after apply)
+ instance_type             = "t2.micro"
+ ipv6_address_count        = (known after apply)
+ ipv6_addresses            = (known after apply)
+ key_name                 = "devops"
+ monitoring               = (known after apply)
+ outpost_arn               = (known after apply)
+ password_data             = (known after apply)
+ placement_group           = (known after apply)
+ placement_partition_number = (known after apply)
+ primary_network_interface_id = (known after apply)
+ private_dns               = (known after apply)
+ private_ip                = (known after apply)
+ public_dns                = (known after apply)
+ public_ip                 = (known after apply)
+ secondary_private_ips     = (known after apply)
+ security_groups           = (known after apply)
+ source_dest_check         = true
+ spot_instance_request_id  = (known after apply)
+ subnet_id                = (known after apply)
+ tags                      = {
  + "Name" = "assignment-2"
}
+ tags_all                 = {
  + "Name" = "assignment-2"
}

Windows PowerShell
Type here to search kyndryl 8:45 PM 6/14/2024
```

```
ubuntu@terraform: ~
+ public_ip          = (known after apply)
+ secondary_private_ips  = (known after apply)
+ security_groups      = (known after apply)
+ source_dest_check    = true
+ spot_instance_request_id = (known after apply)
+ subnet_id           = (known after apply)
+ tags                = {
    + "Name" = "assignment-2"
}
+ tags_all            = {
    + "Name" = "assignment-2"
}
+ tenancy              = (known after apply)
+ user_data            = (known after apply)
+ user_data_base64     = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}

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

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value:
```

```
ubuntu@terraform: ~
+ tenancy              = (known after apply)
+ user_data            = (known after apply)
+ user_data_base64     = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}

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

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.assignment-2: Creating...
aws_eip.eip: Creating...
aws_eip.eip: Creation complete after 1s [id=eipalloc-08f633baef5f30fb6]
aws_instance.assignment-2: Still creating... [10s elapsed]
aws_instance.assignment-2: Still creating... [20s elapsed]
aws_instance.assignment-2: Still creating... [30s elapsed]
aws_instance.assignment-2: Creation complete after 32s [id=i-09991165e172c78fe]
aws_eip_association.eip_assoc: Creating...
aws_eip_association.eip_assoc: Creation complete after 1s [id=eipassoc-09b5911f56c08627b]

apply complete! Resources: 3 added, 0 changed, 0 destroyed.
ubuntu@terraform:~$
```

Instance got created

The image shows two screenshots of the AWS Management Console. The top screenshot is the 'Instances' dashboard, showing one instance named 'assignment-2' with ID i-09991165e172c78fe, which is running on a t2.micro type. The bottom screenshot is the 'Elastic IP addresses' dashboard, showing one public IP address allocated: 3.133.41.40, with allocation ID eipalloc-08f633baef5f30f.

### Assignment 3

A screenshot of a web browser window titled 'Assignment 3 – Terraform'. The page content includes a sidebar with two images labeled 1 and 2, and the main area has the following text:

DevOps Certification Training

**Tasks To Be Performed:**

1. Destroy the previous deployment
2. Create 2 EC2 instances in Ohio and N.Virginia respectively
3. Rename Ohio's instance to 'hello-ohio' and Virginia's instance to 'hello-virginia'

```
ubuntu@terraform:~$ terraform destroy
```

```
ubuntu@terraform: ~
apply complete! Resources: 3 added, 0 changed, 0 destroyed.
ubuntu@terraform: $ terraform destroy
aws_eip.eip: Refreshing state... [id=eipalloc-08f633baef5f30fb6]
aws_instance.assignment-2: Refreshing state... [id=i-09991165e172c78fe]
aws_eip_association.eip_assoc: Refreshing state... [id=eipassoc-09b5911f56c08627b]

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

Terraform will perform the following actions:

# aws_eip.eip will be destroyed
- resource "aws_eip" "eip" {
    - allocation_id      = "eipalloc-08f633baef5f30fb6" -> null
    - arn                = "arn:aws:ec2:us-east-2:637423404772:elastic-ip/eipalloc-08f633baef5f30fb6" -> null
    - association_id    = "eipassoc-09b5911f56c08627b" -> null
    - domain             = "vpc" -> null
    - id                 = "eipalloc-08f633baef5f30fb6" -> null
    - instance            = "i-09991165e172c78fe" -> null
    - network_border_group = "us-east-2" -> null
    - network_interface   = "eni-05842bbe59a00b4d0" -> null
    - private_dns         = "ip-172-31-11-139.us-east-2.compute.internal" -> null
    - private_ip          = "172.31.11.139" -> null
    - public_dns          = "ec2-3-133-41-40.us-east-2.compute.amazonaws.com" -> null
    - public_ip            = "3.133.41.40" -> null
    - public_ipv4_pool    = "amazon" -> null
    - tags               = "{}" -> null
    - tags_all            = "{}" -> null
    - vpc                = true -> null
  } # (4 unchanged attributes hidden)

# aws_eip_association.eip_assoc will be destroyed
- resource "aws_eip_association" "eip_assoc" {
    - allocation_id      = "eipalloc-08f633baef5f30fb6" -> null
    - id                 = "eipassoc-09b5911f56c08627b" -> null
    - instance_id         = "i-09991165e172c78fe" -> null
    - network_interface_id = "eni-05842bbe59a00b4d0" -> null
    - private_ip_address  = "172.31.11.139" -> null
    - public_ip           = "3.133.41.40" -> null
}

```

```
ubuntu@terraform: ~
- tags_all            = "{}" -> null
- vpc                = true -> null
} # (4 unchanged attributes hidden)

# aws_eip_association.eip_assoc will be destroyed
- resource "aws_eip_association" "eip_assoc" {
    - allocation_id      = "eipalloc-08f633baef5f30fb6" -> null
    - id                 = "eipassoc-09b5911f56c08627b" -> null
    - instance_id         = "i-09991165e172c78fe" -> null
    - network_interface_id = "eni-05842bbe59a00b4d0" -> null
    - private_ip_address  = "172.31.11.139" -> null
    - public_ip           = "3.133.41.40" -> null
}

# aws_instance.assignment-2 will be destroyed
- resource "aws_instance" "assignment-2" {
    - ami                = "ami-0f30a9c3a48f3fa79" -> null
    - arn                = "arn:aws:ec2:us-east-2:637423404772:instance/i-09991165e172c78fe" -> null
    - associate_public_ip_address = true -> null
    - availability_zone     = "us-east-2a" -> null
    - cpu_core_count        = 1 -> null
    - cpu_threads_per_core  = 1 -> null
    - disable_api_stop       = false -> null
    - disable_api_termination = false -> null
    - ebs_optimized          = false -> null
    - get_password_data      = false -> null
    - hibernation            = false -> null
    - id                   = "i-09991165e172c78fe" -> null
    - instance_initiated_shutdown_behavior = "stop" -> null
    - instance_state          = "running" -> null
    - instance_type            = "t2.micro" -> null
    - ipv6_address_count      = 0 -> null
    - ipv6_addresses          = [] -> null
    - key_name                = "devops" -> null
    - monitoring              = false -> null
    - placement_partition_number = 0 -> null
    - primary_network_interface_id = "eni-05842bbe59a00b4d0" -> null
    - private_dns              = "ip-172-31-11-139.us-east-2.compute.internal" -> null
    - private_ip                = "172.31.11.139" -> null
    - public_dns                = "ec2-3-133-41-40.us-east-2.compute.amazonaws.com" -> null
}

```

```
ubuntu@terraform: ~
- ipv6_addresses          = [] -> null
- key_name                = "devops" -> null
- monitoring              = false -> null
- placement_partition_number = 0 -> null
- primary_network_interface_id = "eni-05842bbe59a00b4d0" -> null
- private_dns              = "ip-172-31-11-139.us-east-2.compute.internal" -> null
- private_ip               = "172.31.11.139" -> null
- public_dns               = "ec2-3-133-41-40.us-east-2.compute.amazonaws.com" -> null
- public_ip                = "3.133.41.40" -> null
- secondary_private_ips    = [] -> null
- security_groups          = [
    - "default",
]
] -> null
- source_dest_check        = true -> null
- subnet_id                = "subnet-0c6d7066576526423" -> null
- tags                      = {
    - "Name" = "assignment-2"
}
} -> null
- tags_all                 = {
    - "Name" = "assignment-2"
}
} -> null
- tenancy                  = "default" -> null
- user_data_replace_on_change = false -> null
- vpc_security_group_ids   = [
    - "sg-08d26f5903649e3e5",
]
] -> null
# (7 unchanged attributes hidden)

- capacity_reservation_specification {
    capacity_reservation_preference = "open" -> null
}

- cpu_options {
    - core_count      = 1 -> null
    - threads_per_core = 1 -> null
}
# (1 unchanged attribute hidden)
}

- credit_specification {
    - cpu_credits = "standard" -> null
}

```

```
ubuntu@terraform: ~
- maintenance_options {
    - auto_recovery = "default" -> null
}

- metadata_options {
    - http_endpoint           = "enabled" -> null
    - http_protocol_ipv6     = "disabled" -> null
    - http_put_response_hop_limit = 1 -> null
    - http_tokens             = "optional" -> null
    - instance_metadata_tags = "disabled" -> null
}

- private_dns_name_options {
    - enable_resource_name_dns_a_record = false -> null
    - enable_resource_name_dns_aaaa_record = false -> null
    - hostname_type                 = "ip-name" -> null
}

- root_block_device {
    - delete_on_termination = true -> null
    - device_name          = "dev/sda1" -> null
    - encrypted            = false -> null
    - iops                 = 100 -> null
    - tags                 = {} -> null
    - tags_all              = {} -> null
    - throughput            = 0 -> null
    - volume_id             = "vol-0a8c4989ca4549e0a" -> null
    - volume_size            = 8 -> null
    - volume_type            = "gp2" -> null
}
# (1 unchanged attribute hidden)
}

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

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: _
```

```

ubuntu@terraform: ~
  + tags_all           = {} -> null
  - tags_all           = {} -> null
  - throughput         = 0 -> null
  - volume_id          = "vol-0a8c4989ca4549e0a" -> null
  - volume_size         = 8 -> null
  - volume_type         = "gp2" -> null
  # (1 unchanged attribute hidden)
}

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

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_eip_association.eip_assoc: Destroying... [id=eipassoc-09b5911f56c08627b]
aws_eip_association.eip_assoc: Destruction complete after 1s
aws_eip.eip: Destroying... [id=eipalloc-08f633baef5f30fb6]
aws_instance.assignment-2: Destroying... [id=i-09991165e172c78fe]
aws_eip.eip: Destruction complete after 2s
aws_instance.assignment-2: Still destroying... [id=i-09991165e172c78fe, 10s elapsed]
aws_instance.assignment-2: Still destroying... [id=i-09991165e172c78fe, 20s elapsed]
aws_instance.assignment-2: Still destroying... [id=i-09991165e172c78fe, 30s elapsed]
aws_instance.assignment-2: Destruction complete after 40s

Destroy complete! Resources: 3 destroyed.
ubuntu@terraform:~$ ■

```

ubuntu@terraform:~\$ sudo nano main.tf

### script 3

```

provider "aws" {
    alias = "Ohio"
    region = "us-east-2"
    access_key = "AKIAZI2LFI3SGG2NATTZ"
    secret_key = "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"
}

provider "aws" {
    alias = "N_Virginia"
    region = "us-east-1"
    access_key = "AKIAZI2LFI3SGG2NATTZ"
    secret_key = "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"
}

resource "aws_instance" "hello-Ohio"{

```

```
provider = aws.Ohio

ami = "ami-0f30a9c3a48f3fa79"

instance_type = "t2.micro"

key_name = "devops"

tags = {

    Name = "hello-Ohio"
}

}

resource "aws_instance" "hello-virginia"{

    provider = aws.N_Virginia

    ami = "ami-0e001c9271cf7f3b9"

    instance_type = "t2.micro"

    key_name = "devops"

    tags = {

        Name = "hello-virginia"
    }

}
```

```
ubuntu@terraform: ~
GNU nano 6.2
provider "aws" {
    alias = "Ohio"
    region = "us-east-2"
    access_key = "AKIAZI2LFI3SGG2NATTZ"
    secret_key = "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"
}
provider "aws" {
    alias = "N_Virginia"
    region = "us-east-1"
    access_key = "AKIAZI2LFI3SGG2NATTZ"
    secret_key = "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"
}
resource "aws_instance" "hello-Ohio"{
    provider = aws.Ohio
    ami = "ami-0f30a9c3a48f3fa79"
    instance_type = "t2.micro"
    key_name = "devops"
    tags = {
        Name = "hello-Ohio"
    }
}
resource "aws_instance" "hello-virginia"{
    provider = aws.N_Virginia
    ami = "ami-0e001c9271cf7f3b9"
    instance_type = "t2.micro"
    key_name = "devops"
    tags = {
        Name = "hello-virginia"
    }
}
```

ubuntu@terraform:~\$ terraform init

```
ubuntu@terraform:~$ sudo nano main.tf
ubuntu@terraform:~$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.54.0

Terraform has been successfully initialized!

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

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

ubuntu@terraform:~\$ terraform plan

```
ubuntu@terraform: ~
commands will detect it and remind you to do so if necessary.
ubuntu@terraform:~$ terraform plan

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

Terraform will perform the following actions:

# aws_instance.hello-Ohio will be created
+ resource "aws_instance" "hello-Ohio" {
    ami                      = "ami-0f30a9c3a48f3fa79"
    ami                        = (known after apply)
    arn                      = (known after apply)
    associate_public_ip_address = (known after apply)
    availability_zone          = (known after apply)
    cpu_core_count              = (known after apply)
    cpu_threads_per_core        = (known after apply)
    disable_api_stop            = (known after apply)
    disable_api_termination      = (known after apply)
    ebs_optimized                = (known after apply)
    get_password_data           = false
    host_id                    = (known after apply)
    host_resource_group_arn      = (known after apply)
    iam_instance_profile         = (known after apply)
    id                          = (known after apply)
    instance_initiated_shutdown_behavior = (known after apply)
    instance_lifecycle           = (known after apply)
    instance_state                = (known after apply)
    instance_type                  = "t2.micro"
    ipv6_address_count           = (known after apply)
    ipv6_addresses                 = (known after apply)
    key_name                     = "devops"
    monitoring                   = (known after apply)
    outpost_arn                  = (known after apply)
    password_data                 = (known after apply)
    placement_group                = (known after apply)
    placement_partition_number      = (known after apply)
    primary_network_interface_id = (known after apply)
    private_dns                   = (known after apply)
    private_ip                     = (known after apply)
    public_dns                     = (known after apply)
    public_ip                      = (known after apply)

}

ubuntu@kyndryl:~
```

```
ubuntu@terraform: ~
+ public_dns          = (known after apply)
+ public_ip           = (known after apply)
+ secondary_private_ips = (known after apply)
+ security_groups     = (known after apply)
+ source_dest_check   = true
+ spot_instance_request_id = (known after apply)
+ subnet_id           = (known after apply)
+ tags                = {
    + "Name" = "hello-Ohio"
  }
+ tags_all            = {
    + "Name" = "hello-Ohio"
  }
+ tenancy              = (known after apply)
+ user_data            = (known after apply)
+ user_data_base64     = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}

# aws_instance.hello-virginia will be created
resource "aws_instance" "hello-virginia" {
  + ami                  = "ami-0e001c9271cf7f3b9"
  + arn                  = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone     = (known after apply)
  + cpu_core_count        = (known after apply)
  + cpu_threads_per_core  = (known after apply)
  + disable_api_stop      = (known after apply)
  + disable_api_termination = (known after apply)
  + ebs_optimized         = (known after apply)
  + get_password_data     = false
  + host_id               = (known after apply)
  + host_resource_group_arn = (known after apply)
  + iam_instance_profile   = (known after apply)
  + id                    = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle     = (known after apply)
  + instance_state         = (known after apply)
  + instance_type          = "t2.micro"
  + ipv6_address_count     = (known after apply)
}

Windows PowerShell
Type here to search kyndryl 9:16 PM 6/14/2024
```

```
ubuntu@terraform: ~
+ primary_network_interface_id = (known after apply)
+ private_dns                 = (known after apply)
+ private_ip                   = (known after apply)
+ public_dns                   = (known after apply)
+ public_ip                     = (known after apply)
+ secondary_private_ips        = (known after apply)
+ security_groups               = (known after apply)
+ source_dest_check             = true
+ spot_instance_request_id     = (known after apply)
+ subnet_id                     = (known after apply)
+ tags                = {
    + "Name" = "hello-virginia"
  }
+ tags_all            = {
    + "Name" = "hello-virginia"
  }
+ tenancy              = (known after apply)
+ user_data            = (known after apply)
+ user_data_base64     = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}

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

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.
ubuntu@terraform:~$
```

```
ubuntu@terraform:~$ terraform apply
```

```
ubuntu@terraform: ~
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.
ubuntu@terraform:~$ terraform apply

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

Terraform will perform the following actions:

# aws_instance.hello-Ohio will be created
+ resource "aws_instance" "hello-Ohio" {
    + ami                                = "ami-0f30a9c3a48f3fa79"
    + arn                                = (known after apply)
    + associate_public_ip_address        = (known after apply)
    + availability_zone                  = (known after apply)
    + cpu_core_count                     = (known after apply)
    + cpu_threads_per_core              = (known after apply)
    + disable_api_stop                  = (known after apply)
    + disable_api_termination           = (known after apply)
    + ebs_optimized                      = (known after apply)
    + get_password_data                 = false
    + host_id                            = (known after apply)
    + host_resource_group_arn           = (known after apply)
    + iam_instance_profile              = (known after apply)
    + id                                 = (known after apply)
    + instance_initiated_shutdown_behavior = (known after apply)
    + instance_lifecycle                = (known after apply)
    + instance_state                    = (known after apply)
    + instance_type                     = "t2.micro"
    + ipv6_address_count                = (known after apply)
    + ipv6_addresses                    = (known after apply)
    + key_name                           = "devops"
    + monitoring                         = (known after apply)
    + outpost_arn                       = (known after apply)
    + password_data                     = (known after apply)
    + placement_group                   = (known after apply)
    + placement_partition_number        = (known after apply)
    + primary_network_interface_id     = (known after apply)
    + private_dns                        = (known after apply)
    + private_ip                         = (known after apply)
}

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

Do you want to perform these actions? [y/N]: y
ubuntu@terraform:~
```

```
ubuntu@terraform: ~
+ source_dest_check          = true
+ spot_instance_request_id   = (known after apply)
+ subnet_id                  = (known after apply)
+ tags                       = {
    + "Name" = "hello-Ohio"
}
+ tags_all                   = {
    + "Name" = "hello-Ohio"
}
+ tenancy                     = (known after apply)
+ user_data                  = (known after apply)
+ user_data_base64            = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids     = (known after apply)
}

# aws_instance.hello-virginia will be created
+ resource "aws_instance" "hello-virginia" {
    + ami                                = "ami-0e001c9271cf7fb9"
    + arn                                = (known after apply)
    + associate_public_ip_address        = (known after apply)
    + availability_zone                  = (known after apply)
    + cpu_core_count                     = (known after apply)
    + cpu_threads_per_core              = (known after apply)
    + disable_api_stop                  = (known after apply)
    + disable_api_termination           = (known after apply)
    + ebs_optimized                      = (known after apply)
    + get_password_data                 = false
    + host_id                            = (known after apply)
    + host_resource_group_arn           = (known after apply)
    + iam_instance_profile              = (known after apply)
    + id                                 = (known after apply)
    + instance_initiated_shutdown_behavior = (known after apply)
    + instance_lifecycle                = (known after apply)
    + instance_state                    = (known after apply)
    + instance_type                     = "t2.micro"
    + ipv6_address_count                = (known after apply)
    + ipv6_addresses                    = (known after apply)
    + key_name                           = "devops"
    + monitoring                         = (known after apply)
    + outpost_arn                       = (known after apply)
}

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

Do you want to perform these actions? [y/N]: y
ubuntu@terraform:~
```

```
ubuntu@terraform: ~
+ public_ip          = (known after apply)
+ secondary_private_ips = (known after apply)
+ security_groups    = (known after apply)
+ source_dest_check = true
+ spot_instance_request_id = (known after apply)
+ subnet_id          = (known after apply)
+ tags               = {
    + "Name" = "hello-virginia"
}
+ tags_all           = {
    + "Name" = "hello-virginia"
}
+ tenancy             = (known after apply)
+ user_data           = (known after apply)
+ user_data_base64    = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}

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

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value:
```

```
ubuntu@terraform: ~
+ tenancy             = (known after apply)
+ user_data           = (known after apply)
+ user_data_base64    = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}

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

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.hello-Ohio: Creating...
aws_instance.hello-virginia: Creating...
aws_instance.hello-Ohio: Still creating... [10s elapsed]
aws_instance.hello-virginia: Still creating... [10s elapsed]
aws_instance.hello-Ohio: Still creating... [20s elapsed]
aws_instance.hello-virginia: Still creating... [20s elapsed]
aws_instance.hello-Ohio: Still creating... [30s elapsed]
aws_instance.hello-virginia: Still creating... [30s elapsed]
aws_instance.hello-Ohio: Creation complete after 32s [id=i-0f1126ac582fb30a0]
aws_instance.hello-virginia: Creation complete after 32s [id=i-0f8b93b423a69fd10]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
Ubuntu@terraform:~$
```

Instances (2) <a href="#">Info</a>										
<a href="#">Find Instance by attribute or tag (case-sensitive)</a>										
<a href="#">Running</a>										
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public		
tf	i-0ab930563562f7cb6	<span>Running</span>	Q Q	t2.micro	<span>2/2 checks passed</span>	<a href="#">View alarms</a> +	us-east-1a	ec2-52-91-173-3.comp...	52.91.1	
hello-virginia	i-0f8b93b423a69fd10	<span>Running</span>	Q Q	t2.micro	<span>Initializing</span>	<a href="#">View alarms</a> +	us-east-1d	ec2-34-204-12-229.co...	34.204.	

Instances (1) <a href="#">Info</a>		<a href="#">C</a>	<a href="#">Connect</a>	Instance state ▾	Actions ▾	<a href="#">Launch instances</a> ▾	<a href="#">G</a>
<input type="text"/> Find Instance by attribute or tag (case-sensitive)				Running ▾		< 1 >	<a href="#">G</a>
Name	Instance ID	Instance state	Instance type	Status check	Alarm status		
<input type="checkbox"/> hello-Ohio	i-0f1126ac582fb30a0	<a href="#">Running</a> <a href="#">Q</a> <a href="#">C</a>	t2.micro	<a href="#">Initializing</a>	<a href="#">View alarms +</a>		

## Assignment 4

Assignment 4 – Terraform

DevOps Certification Training

**Tasks To Be Performed:**

1. Destroy the previous deployments
2. Create a VPC with the required components using Terraform
3. Deploy an EC2 instance inside the VPC

Solution:

```
ubuntu@terraform:~$ terraform destroy
```

```
ubuntu@terraform: ~
aws_instance.hello-virginia: Still creating... [30s elapsed]
aws_instance.hello-Ohio: Creation complete after 32s [id=i-0f1126ac582fb30a0]
aws_instance.hello-virginia: Creation complete after 32s [id=i-0f8b93b423a69fd10]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
ubuntu@terraform:~$ terraform destroy
aws_instance.hello-Ohio: Refreshing state... [id=i-0f1126ac582fb30a0]
aws_instance.hello-virginia: Refreshing state... [id=i-0f8b93b423a69fd10]

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

Terraform will perform the following actions:

# aws_instance.hello-Ohio will be destroyed
resource "aws_instance" "hello-Ohio" {
    - ami                                = "ami-0f30a9c3a48f3fa79" -> null
    - arn                                = "arn:aws:ec2:us-east-2:637423404772:instance/i-0f1126ac582fb30a0" -> null
    - associate_public_ip_address          = true -> null
    - availability_zone                   = "us-east-2a" -> null
    - cpu_core_count                      = 1 -> null
    - cpu_threads_per_core                = 1 -> null
    - disable_api_stop                   = false -> null
    - disable_api_termination             = false -> null
    - ebs_optimized                       = false -> null
    - get_password_data                  = false -> null
    - hibernation                          = false -> null
    - id                                  = "i-0f1126ac582fb30a0" -> null
    - instance_initiated_shutdown_behavior = "stop" -> null
    - instance_state                      = "running" -> null
    - instance_type                        = "t2.micro" -> null
    - ipv6_address_count                 = 0 -> null
    - ipv6_addresses                      = [] -> null
    - key_name                            = "devops" -> null
    - monitoring                           = false -> null
    - placement_partition_number          = 0 -> null
    - primary_network_interface_id       = "eni-0eb1586823d1c5990" -> null
    - private_dns                          = "ip-172-31-3-35.us-east-2.compute.internal" -> null
    - private_ip                           = "172.31.3.35" -> null
    - public_dns                           = "ec2-3-14-246-27.us-east-2.compute.amazonaws.com" -> null
    - public_ip                            = "3.14.246.27" -> null
}
```

```
ubuntu@terraform: ~
- ebs_optimized                         = false -> null
- get_password_data                     = false -> null
- hibernation                           = false -> null
- id                                    = "i-0f1126ac582fb30a0" -> null
- instance_initiated_shutdown_behavior = "stop" -> null
- instance_state                         = "running" -> null
- instance_type                          = "t2.micro" -> null
- ipv6_address_count                    = 0 -> null
- ipv6_addresses                         = [] -> null
- key_name                             = "devops" -> null
- monitoring                            = false -> null
- placement_partition_number           = 0 -> null
- primary_network_interface_id         = "eni-0eb1586823d1c5990" -> null
- private_dns                           = "ip-172-31-3-35.us-east-2.compute.internal" -> null
- private_ip                            = "172.31.3.35" -> null
- public_dns                            = "ec2-3-14-246-27.us-east-2.compute.amazonaws.com" -> null
- public_ip                             = "3.14.246.27" -> null
- secondary_private_ips                = [] -> null
- security_groups                      = [
    - "default",
] -> null
- source_dest_check                    = true -> null
- subnet_id                            = "subnet-0c6d7066576526423" -> null
- tags                                  = {
    - "Name" = "hello-Ohio"
} -> null
- tags_all                             = {
    - "Name" = "hello-Ohio"
} -> null
- tenancy                               = "default" -> null
- user_data_replace_on_change          = false -> null
- vpc_security_group_ids               = [
    - "sg-08d26f5903649e3e5",
] -> null
# (7 unchanged attributes hidden)

- capacity_reservation_specification {
    - capacity_reservation_preference = "open" -> null
}

- cpu_options {
```

```
ubuntu@terraform: ~
  capacity_reservation_specification {
    capacity_reservation_preference = "open" -> null
  }

  - cpu_options {
    - core_count      = 1 -> null
    - threads_per_core = 1 -> null
    # (1 unchanged attribute hidden)
  }

  - credit_specification {
    - cpu_credits = "standard" -> null
  }

  - enclave_options {
    - enabled = false -> null
  }

  - maintenance_options {
    - auto_recovery = "default" -> null
  }

  - metadata_options {
    - http_endpoint      = "enabled" -> null
    - http_protocol_ipv6 = "disabled" -> null
    - http_put_response_hop_limit = 1 -> null
    - http_tokens        = "optional" -> null
    - instance_metadata_tags = "disabled" -> null
  }

  - private_dns_name_options {
    - enable_resource_name_dns_a_record   = false -> null
    - enable_resource_name_dns_aaaa_record = false -> null
    - hostname_type                      = "ip-name" -> null
  }

  - root_block_device {
    - delete_on_termination = true -> null
    - device_name          = "/dev/sda1" -> null
    - encrypted            = false -> null
    - iops                 = 100 -> null
  }

  - volume {
    - delete_on_termination = true -> null
    - device_name          = "/dev/sdb" -> null
    - encrypted            = false -> null
    - iops                 = 100 -> null
  }
}

ubuntu@terraform: ~
```

```
# aws_instance.hello-virginia will be destroyed
resource "aws_instance" "hello-virginia" {
  ami                  = "ami-0e001c9271cf7f3b9" -> null
  arn                 = "arn:aws:ec2:us-east-1:637423404772:instance/i-0f8b93b423a69fd10" -> null
  associate_public_ip_address = true -> null
  availability_zone   = "us-east-1d" -> null
  cpu_core_count      = 1 -> null
  cpu_threads_per_core = 1 -> null
  disable_api_stop    = false -> null
  disable_api_termination = false -> null
  ebs_optimized       = false -> null
  get_password_data   = false -> null
  hibernation         = false -> null
  id                  = "i-0f8b93b423a69fd10" -> null
  instance_initiated_shutdown_behavior = "stop" -> null
  instance_state       = "running" -> null
  instance_type        = "t2.micro" -> null
  ipv6_address_count  = 0 -> null
  ipv6_addresses       = [] -> null
  key_name             = "devops" -> null
  monitoring           = false -> null
  placement_partition_number = 0 -> null
  primary_network_interface_id = "eni-030147d4390fdaadb" -> null
  private_dns          = "ip-172-31-92-186.ec2.internal" -> null
  private_ip           = "172.31.92.186" -> null
  public_dns           = "ec2-34-204-12-229.compute-1.amazonaws.com" -> null
  public_ip             = "34.204.12.229" -> null
  secondary_private_ips = [] -> null
  security_groups {
    - "default",
    ] -> null
  }
  source_dest_check    = true -> null
  subnet_id            = "subnet-0fa31e321fa3d99f5" -> null
  tags {
    - "Name" = "hello-virginia"
  } -> null
  tags_all {
    - "Name" = "hello-virginia"
  } -> null
  tenancy              = "default" -> null
  user_data_replace_on_change = false -> null
}

ubuntu@terraform: ~
```

```
ubuntu@terraform: ~
+ secondary_private_ips          = [] -> null
- security_groups                = [
-   - "default",
] -> null
- source_dest_check              = true -> null
- subnet_id                      = "subnet-0fa31e321fa3d99f5" -> null
- tags                           = {
-   - "Name" = "hello-virginia"
} -> null
- tags_all                       = {
-   - "Name" = "hello-virginia"
} -> null
- tenancy                         = "default" -> null
- user_data_replace_on_change     = false -> null
- vpc_security_group_ids          = [
-   - "sg-0827d52200bc10ba2",
] -> null
# (7 unchanged attributes hidden)

- capacity_reservation_specification {
-   - capacity_reservation_preference = "open" -> null
}

- cpu_options {
-   - core_count        = 1 -> null
-   - threads_per_core = 1 -> null
} # (1 unchanged attribute hidden)

- credit_specification {
-   - cpu_credits = "standard" -> null
}

- enclave_options {
-   - enabled = false -> null
}

- maintenance_options {
-   - auto_recovery = "default" -> null
}

Windows PowerShell
Type here to search kyndryl
File Explorer
This PC
Downloads
OneDrive
Recycle Bin
Task View
Run
File
Edit
View
Insert
Format
Help
9:31 PM
6/14/2024
```

```
ubuntu@terraform: ~
- http_tokens           = "optional" -> null
- instance_metadata_tags = "disabled" -> null
}

- private_dns_name_options {
-   - enable_resource_name_dns_a_record = false -> null
-   - enable_resource_name_dns_aaaa_record = false -> null
-   - hostname_type                   = "ip-name" -> null
}

- root_block_device {
-   - delete_on_termination = true -> null
-   - device_name          = "/dev/sda1" -> null
-   - encrypted             = false -> null
-   - iops                  = 100 -> null
-   - tags                 = {} -> null
-   - tags_all              = {} -> null
-   - throughput            = 0 -> null
-   - volume_id             = "vol-0526cff8b2e50f34" -> null
-   - volume_size            = 8 -> null
-   - volume_type            = "gp2" -> null
} # (1 unchanged attribute hidden)
}

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

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value:
```

```

ubuntu@terraform:~#
      # {1 unchanged attribute hidden}
    }
}

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

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_instance.hello-Ohio: Destroying... [id=i-0f1126ac582fb30a0]
aws_instance.hello-virginia: Destroying... [id=i-0f8b93b423a69fd10]
aws_instance.hello-Ohio: Still destroying... [id=i-0f1126ac582fb30a0, 10s elapsed]
aws_instance.hello-virginia: Still destroying... [id=i-0f8b93b423a69fd10, 10s elapsed]
aws_instance.hello-Ohio: Still destroying... [id=i-0f1126ac582fb30a0, 20s elapsed]
aws_instance.hello-virginia: Still destroying... [id=i-0f8b93b423a69fd10, 20s elapsed]
aws_instance.hello-Ohio: Still destroying... [id=i-0f1126ac582fb30a0, 30s elapsed]
aws_instance.hello-virginia: Still destroying... [id=i-0f8b93b423a69fd10, 30s elapsed]
aws_instance.hello-Ohio: Still destroying... [id=i-0f1126ac582fb30a0, 40s elapsed]
aws_instance.hello-virginia: Still destroying... [id=i-0f8b93b423a69fd10, 40s elapsed]
aws_instance.hello-Ohio: Destruction complete after 40s
aws_instance.hello-virginia: Destruction complete after 40s

Destroy complete! Resources: 2 destroyed.
ubuntu@terraform:~$
```

The screenshot shows the AWS Lambda console interface. At the top, there's a search bar and a navigation bar with icons for Home, Lambda, CloudWatch, and others. The main area is titled "Instances (2) Info". It has a search bar labeled "Find Instance by attribute or tag (case-sensitive)" and a dropdown set to "Terminated". Below this is a table with columns: Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. Two rows are listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
assignment-2	i-09991165e172c78fe	Terminated	t2.micro	-	<a href="#">View alarms</a>
hello-Ohio	i-0f1126ac582fb30a0	Terminated	t2.micro	-	<a href="#">View alarms</a>

This screenshot shows the AWS Lambda console with a single instance listed. The interface is identical to the previous one, with a search bar, a "Terminated" filter, and a table. The table has an additional column for Availability Zone and Public IPv4 DNS. One row is shown:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
hello-virginia	i-0f8b93b423a69fd10	Terminated	t2.micro	-	<a href="#">View alarms</a>	us-east-1d	-

```
ubuntu@terraform:~$ sudo nano main.tf
```

```
ubuntu@terraform:~$ sudo nano main.tf
ubuntu@terraform:~$ sudo nano main.tf
```

```
provider "aws" {
```

```
  region = "us-east-2"
```

```
access_key = "AKIAZI2LFI3SGG2NATTZ"
secret_key = "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"
}

resource "aws_instance" "assignment-4"{
    ami = "ami-0f30a9c3a48f3fa79"
    instance_type = "t2.micro"
    subnet_id = aws_subnet.assignment-4-subnet.id
    associate_public_ip_address = true
    key_name = "devops"
    tags = {
        Name = "assignment-4"
    }
}

resource "aws_vpc" "assignment-4-vpc"{
    cidr_block = "10.10.0.0/16"
    tags = {
        Name = "assignment-4-vpc"
    }
}

resource "aws_internet_gateway" "gw" {
    vpc_id = aws_vpc.assignment-4-vpc.id
    tags = {
        Name = "main-gw"
    }
}

resource "aws_subnet" "assignment-4-subnet"{
    vpc_id = aws_vpc.assignment-4-vpc.id
    cidr_block = "10.10.0.0/18"
    map_public_ip_on_launch = true
}
```

```
availability_zone = "us-east-2a"
tags = {
  Name = "assignment-4-subnet"
}
resource "aws_route_table" "public" {
  vpc_id = aws_vpc.assignment-4-vpc.id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gw.id
  }
  tags = {
    Name = "public-rt"
  }
}
resource "aws_route_table_association" "public" {
  subnet_id   = aws_subnet.assignment-4-subnet.id
  route_table_id = aws_route_table.public.id
}
```

```
ubuntu@terraform: ~
GNU nano 6.2
provider "aws"{
  region = "us-east-2"
  access_key = "AKIAZI2LFI3SGG2NATTZ"
  secret_key = "e7DzAGTs004JX99gSjs+tN+1AM+qwmllDaBeYlg"
}
resource "aws_instance" "assignment-4"{
  ami = "ami-0f30a9c3a48f3fa79"
  instance_type = "t2.micro"
  subnet_id = aws_subnet.assignment-4-subnet.id
  associate_public_ip_address = true
  key_name = "devops"
  tags = {
    Name = "assignment-4"
  }
}
resource "aws_vpc" "assignment-4-vpc"{
  cidr_block = "10.10.0.0/16"
  tags = {
    Name = "assignment-4-vpc"
  }
}
resource "aws_internet_gateway" "gw" {
  vpc_id = aws_vpc.assignment-4-vpc.id
  tags = {
    Name = "main-gw"
  }
}
resource "aws_subnet" "assignment-4-subnet"{
  vpc_id = aws_vpc.assignment-4-vpc.id
  cidr_block = "10.10.0.0/18"
  map_public_ip_on_launch = true
  availability_zone = "us-east-2a"
  tags = {
    Name = "assignment-4-subnet"
  }
}
resource "aws_route_table" "public" {
  vpc_id = aws_vpc.assignment-4-vpc.id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gw.id
  }
  tags = {
    Name = "public-rt"
  }
}
resource "aws_route_table_association" "public" {
  subnet_id = aws_subnet.assignment-4-subnet.id
  route_table_id = aws_route_table.public.id
}

main.tf
```

```
ubuntu@terraform: ~
GNU nano 6.2
cidr_block = "10.10.0.0/16"
tags = {
  Name = "assignment-4-vpc"
}
resource "aws_internet_gateway" "gw" {
  vpc_id = aws_vpc.assignment-4-vpc.id
  tags = {
    Name = "main-gw"
  }
}
resource "aws_subnet" "assignment-4-subnet"{
  vpc_id = aws_vpc.assignment-4-vpc.id
  cidr_block = "10.10.0.0/18"
  map_public_ip_on_launch = true
  availability_zone = "us-east-2a"
  tags = {
    Name = "assignment-4-subnet"
  }
}
resource "aws_route_table" "public" {
  vpc_id = aws_vpc.assignment-4-vpc.id
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = aws_internet_gateway.gw.id
  }
  tags = {
    Name = "public-rt"
  }
}
resource "aws_route_table_association" "public" {
  subnet_id = aws_subnet.assignment-4-subnet.id
  route_table_id = aws_route_table.public.id
}

main.tf
```

ubuntu@terraform:~\$ terraform init

```
ubuntu@terraform:~$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.54.0

Terraform has been successfully initialized!

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

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

ubuntu@terraform:~\$ terraform plan

```
A managed resource "aws_assignment-4-subnet" "id" has not been declared in the root module.

ubuntu@terraform:~$ sudo nano main.tf
ubuntu@terraform:~$ terraform plan

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

Terraform will perform the following actions:

# aws_instance.assignment-4 will be created
+ resource "aws_instance" "assignment-4" {
    ami                               = "ami-0f30a9c3a48f3fa79"
    arn                             = (known after apply)
    associate_public_ip_address      = true
    availability_zone                = (known after apply)
    cpu_core_count                   = (known after apply)
    cpu_threads_per_core            = (known after apply)
    disable_api_stop                 = (known after apply)
    disable_api_termination          = (known after apply)
    ebs_optimized                    = (known after apply)
    get_password_data                = false
    host_id                          = (known after apply)
    host_resource_group_arn          = (known after apply)
    iam_instance_profile              = (known after apply)
    id                                = (known after apply)
    instance_initiated_shutdown_behavior = (known after apply)
    instance_lifecycle               = (known after apply)
    instance_state                   = (known after apply)
    instance_type                     = "t2.micro"
    ipv6_address_count               = (known after apply)
    ipv6_addresses                   = (known after apply)
    key_name                         = "devops"
    monitoring                       = (known after apply)
    outpost_arn                      = (known after apply)
    password_data                    = (known after apply)
    placement_group                  = (known after apply)

}
```

```
ubuntu@terraform: ~
+ ipv6_address_count      = (known after apply)
+ ipv6_addresses          = (known after apply)
+ key_name                 = "devops"
+ monitoring               = (known after apply)
+ outpost_arn              = (known after apply)
+ password_data            = (known after apply)
+ placement_group          = (known after apply)
+ placement_partition_number = (known after apply)
+ primary_network_interface_id = (known after apply)
+ private_dns               = (known after apply)
+ private_ip                = (known after apply)
+ public_dns                = (known after apply)
+ public_ip                 = (known after apply)
+ secondary_private_ips    = (known after apply)
+ security_groups           = (known after apply)
+ source_dest_check         = true
+ spot_instance_request_id = (known after apply)
+ subnet_id                 = (known after apply)
+ tags                      = {
    + "Name" = "assignment-4"
}
+ tags_all                 = {
    + "Name" = "assignment-4"
}
+ tenancy                  = (known after apply)
+ user_data                = (known after apply)
+ user_data_base64          = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids    = (known after apply)
}

# aws_internet_gateway.gw will be created
+ resource "aws_internet_gateway" "gw" {
    + arn      = (known after apply)
    + id       = (known after apply)
    + owner_id = (known after apply)
    + tags     = {
}

```

```
ubuntu@terraform: ~
+ tags_all = {
    + "Name" = "main-gw"
}
+ vpc_id   = (known after apply)
}

# aws_route_table.public will be created
+ resource "aws_route_table" "public" {
    + arn      = (known after apply)
    + id       = (known after apply)
    + owner_id = (known after apply)
    + propagating_vgws = (known after apply)
    + route    = [
        +
        + cidr_block          = "0.0.0.0/0"
        + gateway_id          = (known after apply)
        # (11 unchanged attributes hidden)
    ],
    + tags     = {
        + "Name" = "public-rt"
    }
    + tags_all   = {
        + "Name" = "public-rt"
    }
    + vpc_id     = (known after apply)
}

# aws_route_table_association.public will be created
+ resource "aws_route_table_association" "public" {
    + id      = (known after apply)
    + route_table_id = (known after apply)
    + subnet_id     = (known after apply)
}

# aws_subnet.assignment-4-subnet will be created
+ resource "aws_subnet" "assignment-4-subnet" {
```

```
ubuntu@terraform:~$ terraform plan
+ cidr_block          = "10.10.0.0/16"
+ default_network_acl_id = (known after apply)
+ default_route_table_id = (known after apply)
+ default_security_group_id = (known after apply)
+ dhcp_options_id      = (known after apply)
+ enable_dns_hostnames = (known after apply)
+ enable_dns_support   = true
+ enable_network_address_usage_metrics = (known after apply)
+ id                  = (known after apply)
+ instance_tenancy     = "default"
+ ipv6_association_id  = (known after apply)
+ ipv6_cidr_block      = (known after apply)
+ ipv6_cidr_block_network_border_group = (known after apply)
+ main_route_table_id  = (known after apply)
+ owner_id             = (known after apply)
+ tags                = [
  + "Name" = "assignment-4-vpc"
]
+ tags_all             = {
  + "Name" = "assignment-4-vpc"
}
}

Plan: 6 to add, 0 to change, 0 to destroy.

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.
ubuntu@terraform:~$
```

ubuntu@terraform:~\$ terraform apply

```
ubuntu@terraform:~$ terraform apply
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.
ubuntu@terraform:~$ terraform apply

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

Terraform will perform the following actions:

# aws_instance.assignment-4 will be created
+ resource "aws_instance" "assignment-4" {
  + ami                      = "ami-0f30a9c3a48f3fa79"
  + arn                      = (known after apply)
  + associate_public_ip_address = true
  + availability_zone         = (known after apply)
  + cpu_core_count            = (known after apply)
  + cpu_threads_per_core      = (known after apply)
  + disable_api_stop           = (known after apply)
  + disable_api_termination    = (known after apply)
  + ebs_optimized              = (known after apply)
  + get_password_data          = false
  + host_id                   = (known after apply)
  + host_resource_group_arn    = (known after apply)
  + iam_instance_profile       = (known after apply)
  + id                         = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle          = (known after apply)
  + instance_state              = (known after apply)
  + instance_type               = "t2.micro"
  + ipv6_address_count          = (known after apply)
  + ipv6_addresses              = (known after apply)
  + key_name                   = "devops"
  + monitoring                 = (known after apply)
  + outpost_arn                 = (known after apply)
  + password_data               = (known after apply)
  + placement_group             = (known after apply)
}
```

```
ubuntu@terraform: ~
+ monitoring          = (known after apply)
+ outpost_arn         = (known after apply)
+ password_data       = (known after apply)
+ placement_group     = (known after apply)
+ placement_partition_number = (known after apply)
+ primary_network_interface_id = (known after apply)
+ private_dns         = (known after apply)
+ private_ip          = (known after apply)
+ public_dns          = (known after apply)
+ public_ip           = (known after apply)
+ secondary_private_ips = (known after apply)
+ security_groups     = (known after apply)
+ source_dest_check   = true
+ spot_instance_request_id = (known after apply)
+ subnet_id           = (known after apply)
+ tags               = [
+   + "Name" = "assignment-4"
+ ]
+ tags_all            = {
+   + "Name" = "assignment-4"
+ }
+ tenancy              = (known after apply)
+ user_data            = (known after apply)
+ user_data_base64     = (known after apply)
+ user_data_replace_on_change = false
+ vpc_security_group_ids = (known after apply)
}

# aws_internet_gateway.gw will be created
resource "aws_internet_gateway" "gw" {
+ arn      = (known after apply)
+ id       = (known after apply)
+ owner_id = (known after apply)
+ tags     = {
+   + "Name" = "main-gw"
+ }
+ tags_all = {
+   + "Name" = "main-gw"
+ }
+ vpc_id   = (known after apply)
}

ubuntu@terraform: ~
```

```
+ tags      = {
+   + "Name" = "main-gw"
+ }
+ tags_all = {
+   + "Name" = "main-gw"
+ }
+ vpc_id    = (known after apply)
}

# aws_route_table.public will be created
resource "aws_route_table" "public" {
+ arn      = (known after apply)
+ id       = (known after apply)
+ owner_id = (known after apply)
+ propagating_vgws = (known after apply)
+ route    = [
+   + {
+     + cidr_block          = "0.0.0.0/0"
+     + gateway_id          = (known after apply)
+     # (11 unchanged attributes hidden)
+   },
+ ],
+ tags      = {
+   + "Name" = "public-rt"
+ }
+ tags_all = {
+   + "Name" = "public-rt"
+ }
+ vpc_id    = (known after apply)
}

# aws_route_table_association.public will be created
resource "aws_route_table_association" "public" {
+ id      = (known after apply)
+ route_table_id = (known after apply)
+ subnet_id     = (known after apply)
}
```

```
ubuntu@terraform: ~
# aws_route_table_association.public will be created
+ resource "aws_route_table_association" "public" {
  + id          = (known after apply)
  + route_table_id = (known after apply)
  + subnet_id    = (known after apply)
}

# aws_subnet.assignment-4-subnet will be created
+ resource "aws_subnet" "assignment-4-subnet" {
  + arn           = (known after apply)
  + assign_ipv6_address_on_creation = false
  + availability_zone      = "us-east-2a"
  + availability_zone_id   = (known after apply)
  + cidr_block            = "10.10.0.0/18"
  + enable_dns64          = false
  + enable_resource_name_dns_a_record_on_launch = false
  + enable_resource_name_dns_aaaa_record_on_launch = false
  + id                  = (known after apply)
  + ipv6_cidr_block_association_id = (known after apply)
  + ipv6_native          = false
  + map_public_ip_on_launch = true
  + owner_id             = (known after apply)
  + private_dns_hostname_type_on_launch = (known after apply)
  + tags                = {
    + "Name" = "assignment-4-subnet"
  }
  + tags_all            = {
    + "Name" = "assignment-4-subnet"
  }
  + vpc_id              = (known after apply)
}

# aws_vpc.assignment-4-vpc will be created
+ resource "aws_vpc" "assignment-4-vpc" {
  + arn           = (known after apply)
  + cidr_block    = "10.10.0.0/16"
}

Windows PowerShell
Type here to search kyndryl 11:33 PM
6/14/2024
```

```
ubuntu@terraform: ~
  + "Name" = "assignment-4-vpc"
}
}

Plan: 6 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_vpc.assignment-4-vpc: Creating...
aws_vpc.assignment-4-vpc: Creation complete after 1s [id=vpc-0edd1d81eee18f354]
aws_internet_gateway.gw: Creating...
aws_subnet.assignment-4-subnet: Creating...
aws_internet_gateway.gw: Creation complete after 0s [id=igw-0cfbf239836aac467]
aws_route_table.public: Creating...
aws_route_table.public: Creation complete after 1s [id=rtb-07814ed7efc469e8d]
aws_subnet.assignment-4-subnet: Still creating... [10s elapsed]
aws_subnet.assignment-4-subnet: Creation complete after 11s [id=subnet-00b7177dd8b1364dc]
aws_instance.assignment-4: Creating...
aws_route_table_association.public: Creating...
aws_route_table_association.public: Creation complete after 0s [id=rtbassoc-06e250ced79c75896]
aws_instance.assignment-4: Still creating... [10s elapsed]
aws_instance.assignment-4: Still creating... [20s elapsed]
aws_instance.assignment-4: Still creating... [30s elapsed]
aws_instance.assignment-4: Creation complete after 31s [id=i-042e5090b2f412bef]

Apply complete! Resources: 6 added, 0 changed, 0 destroyed.
ubuntu@terraform: $
```

Terraform will perform the actions described above.

```
Windows PowerShell
Type here to search kyndryl 11:34 PM
6/14/2024
```

Instance successfully created

Instances (1) <a href="#">Info</a>		<a href="#">C</a>	<a href="#">Connect</a>	Instance state ▾	Actions ▾	<a href="#">Launch instances</a> ▾	<a href="#">G</a>
		<a href="#">Find Instance by attribute or tag (case-sensitive)</a>		Running ▾	< 1 >	<a href="#">View alarms</a> +	<a href="#">G</a>
<input type="checkbox"/>	Name <a href="#">✎</a>	Instance ID	Instance state	Instance type	Status check	Alarm status	<a href="#">G</a>
<input type="checkbox"/>	assignment-4	i-042e5090b2f412bef	<span>Running</span> <a href="#">Q</a> <a href="#">Q</a>	t2.micro	<span>2/2 checks passed</span>	<a href="#">View alarms</a> +	<a href="#">G</a>

Public ip is assigned to the instance

i-042e5090b2f412bef (assignment-4)			<a href="#">G</a>
Instance ID	Public IPv4 address	Private IPv4 addresses	
<a href="#">i-042e5090b2f412bef (assignment-4)</a>	<a href="#">3.147.193.65   open address</a>	<a href="#">10.10.7.205</a>	
IPv6 address	Instance state	Public IPv4 DNS	
-	<span>Running</span>	-	
Hostname type	Private IP DNS name (IPv4 only)		
IP name: ip-10-10-7-205.us-east-2.compute.internal	<a href="#">ip-10-10-7-205.us-east-2.compute.internal</a>		

VPC created

Your VPCs (1/2) <a href="#">Info</a>						<a href="#">C</a>	Actions ▾	<a href="#">Create VPC</a>	<a href="#">G</a>
						<a href="#">C</a>	< 1 >	<a href="#">G</a>	<a href="#">G</a>
<input type="checkbox"/>	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR				
<input checked="" type="checkbox"/>	assignment-4-vpc	vpc-0edd1d81eee18f354	<span>Available</span>	10.10.0.0/16	-				
<input type="checkbox"/>	-	vpc-08f937497665e0196	<span>Available</span>	172.31.0.0/16	-				

Subnet created

<input checked="" type="checkbox"/>	assignment-4-subnet	subnet-00b7177dd8b1364dc	<span>Available</span>	subnet-00b7177dd8b136...   assig...

Route table created

<input checked="" type="checkbox"/>	public-rt	rtb-07814ed7efc469e8d	subnet-00b7177dd8b136...	-	No

## routes

Routes (2)		Edit routes	
Destination	Target	Status	Propagated
0.0.0.0/0	<a href="#">igw-0cfbf239836aac467</a>	Active	No
10.10.0.0/16	local	Active	No

## subnet association

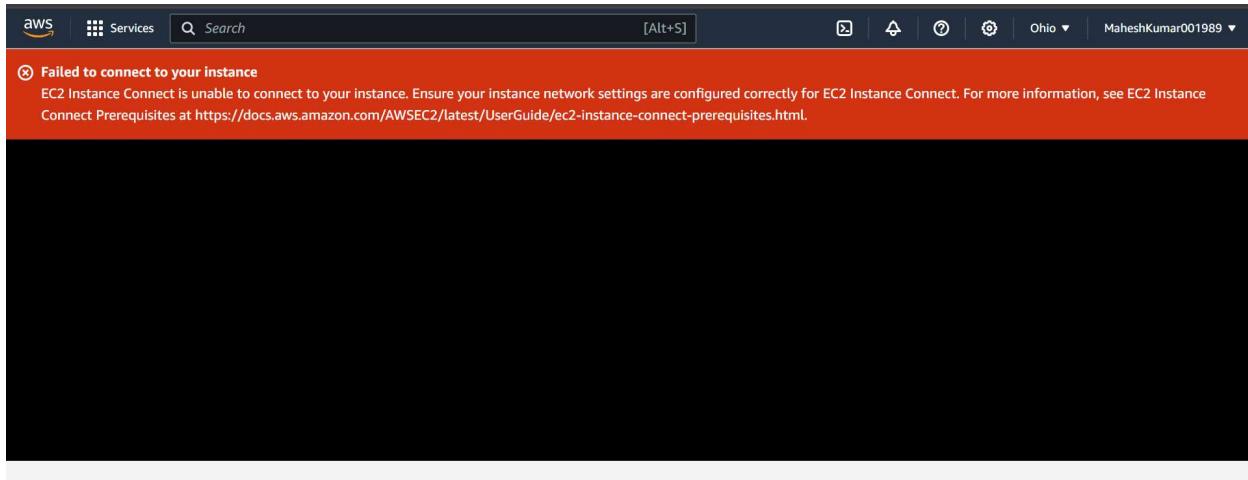
Explicit subnet associations (1)		Edit subnet associations	
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
assignment-4-subnet	<a href="#">subnet-00b7177dd8b1364dc</a>	10.10.0.0/18	-

Internet gateway created & attached to our VPC

Internet gateways (1/2) <a href="#">Info</a>			
<a href="#">Actions</a> <a href="#">Create internet gateway</a>			
Name	Internet gateway ID	State	VPC ID
-	<a href="#">igw-093ccae459951294b</a>	Attached	<a href="#">vpc-08f937497665e0196</a>
<input checked="" type="checkbox"/> main-gw	<a href="#">igw-0cfbf239836aac467</a>	Attached	<a href="#">vpc-0edd1d81eee18f354 a</a>

Details			
Internet gateway ID <a href="#">igw-0cfbf239836aac467</a>	State <span style="color: green;">Attached</span>	VPC ID <a href="#">vpc-0edd1d81eee18f354  assignment-4-vpc</a>	Owner <a href="#">637423404772</a>

Tried connecting to instance but getting this error



Security group is bounded

The screenshot shows the 'Edit inbound rules' page for a security group. The URL is EC2 > Security Groups > sg-014cd4e66699868fb - default > Edit inbound rules. The page title is 'Edit inbound rules' with an 'Info' link. A note below says 'Inbound rules control the incoming traffic that's allowed to reach the instance.' The main table has columns: Security group rule ID, Type, Protocol, Port range, Source, and Description - optional. One row is visible: sgr-0a746a2f81ea1952e, All traffic, All, All, sg-014cd4e66699868fb, and an empty description field. A tooltip for the source shows the full ID: sg-014cd4e66699868fb. A 'Delete' button is also present. At the bottom, there's an 'Add rule' button and a note about allowing all IP addresses.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0a746a2f81ea1952e	All traffic	All	All	sg-014cd4e66699868fb	

**Inbound rules** [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
-	All traffic	All	All	An...	0.0.0.0/0

**Inbound rules** [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
-	All traffic	All	All	An...	0.0.0.0/0

**⚠️** Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. [X](#)

[Cancel](#) [Preview changes](#) [Save rules](#)

successfully connected to the ec2 instance

```
aws | Services | Search [Alt+S] | □ | 🔍 | ⓘ | ⚙ | Ohio | MaheshKumar001989

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-10-10-7-205:~$
```

## Assignment 5

DevOps Certification Training

### Tasks To Be Performed:

1. Destroy the previous deployments
2. Create a script to install Apache2
3. Run this script on a newly created EC2 instance
4. Print the IP address of the instance in a file on the local once deployed

1

2

11:46 PM  
6/14/2024

### Solution:

```
ubuntu@terraform:~$ terraform destroy
```

```
ubuntu@terraform:~  
ubuntu@terraform:~$ terraform destroy  
aws_vpc.assignment-4-vpc: Refreshing state... [id=vpc-0edd1d81eee18f354]  
aws_subnet.assignment-4-subnet: Refreshing state... [id=subnet-00b7177d8b1364dc]  
aws_internet_gateway.gw: Refreshing state... [id=igw-0cfbf239836aacd467]  
aws_route_table.public: Refreshing state... [id=rtb-07814ed7efc469e8d]  
aws_instance.assignment-4: Refreshing state... [id=i-042e5090b2f412bef]  
aws_route_table_association.public: Refreshing state... [id=rtbassoc-06e250ced79c75896]  
  
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:  
- destroy  
  
Terraform will perform the following actions:  
  
# aws_instance.assignment-4 will be destroyed  
- resource "aws_instance" "assignment-4" {  
    - ami = "ami-0f30a9c3a48f3fa79" -> null  
    - arn = "arn:aws:ec2:us-east-2:637423404772:instance/i-042e5090b2f412bef" -> null  
    - associate_public_ip_address = true -> null  
    - availability_zone = "us-east-2a" -> null  
    - cpu_core_count = 1 -> null  
    - cpu_threads_per_core = 1 -> null  
    - disable_api_stop = false -> null  
    - disable_api_termination = false -> null  
    - ebs_optimized = false -> null  
    - get_password_data = false -> null  
    - hibernation = false -> null  
    - id = "i-042e5090b2f412bef" -> null  
    - instance_initiated_shutdown_behavior = "stop" -> null  
    - instance_state = "running" -> null  
    - instance_type = "t2.micro" -> null  
    - ipv6_address_count = 0 -> null  
    - ipv6_addresses = [] -> null  
    - key_name = "devops" -> null  
    - monitoring = false -> null  
    - placement_partition_number = 0 -> null  
    - primary_network_interface_id = "eni-0ffcc6bfd10337fde" -> null  
}  
Terraform will destroy all your managed infrastructure, as shown above.  
Windows PowerShell 6.1.17760.1 11:49 PM 6/14/2024
```

```
ubuntu@terraform:~  
- "Name" = "assignment-4"  
} -> null  
- tags_all = {  
    - "Name" = "assignment-4"  
} -> null  
- tenancy = "default" -> null  
- user_data_replace_on_change = false -> null  
- vpc_security_group_ids = [  
    - "sg-014cd4e66699868fb",  
] -> null  
# (8 unchanged attributes hidden)  
  
- capacity_reservation_specification {  
    - capacity_reservation_preference = "open" -> null  
}  
  
- cpu_options {  
    - core_count = 1 -> null  
    - threads_per_core = 1 -> null  
    # (1 unchanged attribute hidden)  
}  
  
- credit_specification {  
    - cpu_credits = "standard" -> null  
}  
  
- enclave_options {  
    - enabled = false -> null  
}  
  
- maintenance_options {  
    - auto_recovery = "default" -> null  
}  
  
- metadata_options {  
    - http_endpoint = "enabled" -> null  
    - http_protocol_ipv6 = "disabled" -> null  
}  
Terraform will destroy all your managed infrastructure, as shown above.  
Windows PowerShell 6.1.17760.1 11:49 PM 6/14/2024
```

```
ubuntu@terraform: ~
  - instance_metadata_tags      = "disabled" -> null
  }

  - private_dns_name_options {
    - enable_resource_name_dns_a_record   = false -> null
    - enable_resource_name_dns_aaaa_record = false -> null
    - hostname_type                      = "ip-name" -> null
  }

  - root_block_device {
    - delete_on_termination = true -> null
    - device_name          = "/dev/sda1" -> null
    - encrypted            = false -> null
    - iops                 = 100 -> null
    - tags                 = {} -> null
    - tags_all              = {} -> null
    - throughput            = 0 -> null
    - volume_id             = "vol-01eb86279e525d40a" -> null
    - volume_size            = 8 -> null
    - volume_type            = "gp2" -> null
    # (1 unchanged attribute hidden)
  }
}

# aws_internet_gateway.gw will be destroyed
resource "aws_internet_gateway" "gw" {
  - arn      = "arn:aws:ec2:us-east-2:637423404772:internet-gateway/igw-0cfbf239836aac467" -> null
  - id       = "igw-0cfbf239836aac467" -> null
  - owner_id = "637423404772" -> null
  - tags     = {
    - "Name" = "main-gw"
  } -> null
  - tags_all = {
    - "Name" = "main-gw"
  } -> null
  - vpc_id   = "vpc-0edd1d81eee18f354" -> null
}
Terraform will destroy all your managed infrastructure, as shown above.
```

Windows Taskbar: Type here to search [kyndryl] 11:49 PM 6/14/2024

```
ubuntu@terraform: ~
# aws_internet_gateway.gw will be destroyed
resource "aws_internet_gateway" "gw" {
  - arn      = "arn:aws:ec2:us-east-2:637423404772:internet-gateway/igw-0cfbf239836aac467" -> null
  - id       = "igw-0cfbf239836aac467" -> null
  - owner_id = "637423404772" -> null
  - tags     = {
    - "Name" = "main-gw"
  } -> null
  - tags_all = {
    - "Name" = "main-gw"
  } -> null
  - vpc_id   = "vpc-0edd1d81eee18f354" -> null
}

# aws_route_table.public will be destroyed
resource "aws_route_table" "public" {
  - arn      = "arn:aws:ec2:us-east-2:637423404772:route-table/rtb-07814ed7efc469e8d" -> null
  - id       = "rtb-07814ed7efc469e8d" -> null
  - owner_id = "637423404772" -> null
  - propagating_vgws = [] -> null
  - route {
    - {
      - cidr_block      = "0.0.0.0/0"
      - gateway_id      = "igw-0cfbf239836aac467"
      # (11 unchanged attributes hidden)
    },
  ] -> null
  - tags     = {
    - "Name" = "public-rt"
  } -> null
  - tags_all = {
    - "Name" = "public-rt"
  } -> null
  - vpc_id   = "vpc-0edd1d81eee18f354" -> null
}

# aws_route_table_association.public will be destroyed
Terraform will destroy all your managed infrastructure, as shown above.
```

Windows Taskbar: Type here to search [kyndryl] 11:49 PM 6/14/2024

```
ubuntu@terraform: ~
+ {
+   - cidr_block          = "0.0.0.0/0"
+   - gateway_id          = "igw-0cfbf239836aac467"
+   # (11 unchanged attributes hidden)
+ },
] -> null
- tags           = {
-   - "Name" = "public-rt"
} -> null
- tags_all       = {
-   - "Name" = "public-rt"
} -> null
- vpc_id         = "vpc-0edd1d81eee18f354" -> null
}

# aws_route_table_association.public will be destroyed
- resource "aws_route_table_association" "public" {
-   id              = "rtbassoc-06e250ced79c75896" -> null
-   route_table_id = "rtb-07814ed7efc469e8d" -> null
-   subnet_id      = "subnet-00b7177dd8b1364dc" -> null
# (1 unchanged attribute hidden)
}

# aws_subnet.assignment-4-subnet will be destroyed
- resource "aws_subnet" "assignment-4-subnet" {
-   arn             = "arn:aws:ec2:us-east-2:637423404772:subnet/subnet-00b7177dd8b1364dc" -> null
-   assign_ip6_address_on_creation = false -> null
-   availability_zone        = "us-east-2a" -> null
-   availability_zone_id     = "use2-az1" -> null
-   cidr_block               = "10.10.0.0/18" -> null
-   enable_dns64              = false -> null
-   enable_lni_at_device_index = 0 -> null
-   enable_resource_name_dns_a_record_on_launch = false -> null
-   enable_resource_name_dns_aaaa_record_on_launch = false -> null
-   id                      = "subnet-00b7177dd8b1364dc" -> null
-   ipv6_native              = false -> null
-   map_customer_owned_ip_on_launch = false -> null
}
Terraform will destroy all your managed infrastructure, as shown above.

Windows Taskbar: Type here to search [kyndryl] 11:49 PM 6/14/2024
```

```
ubuntu@terraform: ~
- private_dns_hostname_type_on_launch      = "ip-name" -> null
- tags           = {
-   - "Name" = "assignment-4-subnet"
} -> null
- tags_all       = {
-   - "Name" = "assignment-4-subnet"
} -> null
- vpc_id         = "vpc-0edd1d81eee18f354" -> null
}

# aws_vpc.assignment-4-vpc will be destroyed
- resource "aws_vpc" "assignment-4-vpc" {
-   arn             = "arn:aws:ec2:us-east-2:637423404772:vpc/vpc-0edd1d81eee18f354" -> null
-   assign_generated_ipv6_cidr_block        = false -> null
-   cidr_block                 = "10.10.0.0/16" -> null
-   default_network_acl_id            = "acl-0f0a841fe49fb6cc6" -> null
-   default_route_table_id           = "rtb-07bd80dfc54562105" -> null
-   default_security_group_id        = "sg-014cd4e66699868fb" -> null
-   dhcp_options_id                 = "dopt-0d13ef9aa778c8eaa" -> null
-   enable_dns_hostnames            = false -> null
-   enable_dns_support              = true -> null
-   enable_network_address_usage_metrics = false -> null
-   id                          = "vpc-0edd1d81eee18f354" -> null
-   instance_tenancy                = "default" -> null
-   ipv6_netmask_length             = 0 -> null
-   main_route_table_id             = "rtb-07bd80dfc54562105" -> null
-   owner_id                     = "637423404772" -> null
-   tags           = {
-   - "Name" = "assignment-4-vpc"
} -> null
- tags_all       = {
-   - "Name" = "assignment-4-vpc"
} -> null
# (4 unchanged attributes hidden)
}

Terraform will destroy all your managed infrastructure, as shown above.

Windows Taskbar: Type here to search [kyndryl] 11:49 PM 6/14/2024
```

```
ubuntu@terraform:~
```

```
    } -> null
  - tags_all          = {
    - "Name" = "assignment-4-vpc"
  } -> null
  # (4 unchanged attributes hidden)
}

Plan: 0 to add, 0 to change, 6 to destroy.

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_instance.assignment-4: Destroying... [id=i-042e5090b2f412bef]
aws_route_table_association.public: Destroying... [id=rtbassoc-06e250ced79c75896]
aws_route_table_association.public: Destruction complete after 1s
aws_route_table.public: Destroying... [id=rtb-07814ed7efc469e8d]
aws_route_table.public: Destruction complete after 0s
aws_internet_gateway.gw: Destroying... [id=igw-0cfbf239836aac467]
```

Terraform will destroy all your managed infrastructure, as shown above.



```
ubuntu@terraform:~
```

```
    } -> null
  # (4 unchanged attributes hidden)
}

Plan: 0 to add, 0 to change, 6 to destroy.

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_instance.assignment-4: Destroying... [id=i-042e5090b2f412bef]
aws_route_table_association.public: Destroying... [id=rtbassoc-06e250ced79c75896]
aws_route_table_association.public: Destruction complete after 1s
aws_route_table.public: Destroying... [id=rtb-07814ed7efc469e8d]
aws_route_table.public: Destruction complete after 0s
aws_internet_gateway.gw: Destroying... [id=igw-0cfbf239836aac467]
aws_instance.assignment-4: Still destroying... [id=i-042e5090b2f412bef, 10s elapsed]
aws_internet_gateway.gw: Still destroying... [id=igw-0cfbf239836aac467, 10s elapsed]
aws_instance.assignment-4: Still destroying... [id=i-042e5090b2f412bef, 20s elapsed]
aws_internet_gateway.gw: Still destroying... [id=igw-0cfbf239836aac467, 20s elapsed]
aws_instance.assignment-4: Still destroying... [id=i-042e5090b2f412bef, 30s elapsed]
aws_internet_gateway.gw: Still destroying... [id=igw-0cfbf239836aac467, 30s elapsed]
aws_internet_gateway.gw: Destruction complete after 37s
aws_instance.assignment-4: Still destroying... [id=i-042e5090b2f412bef, 40s elapsed]
aws_instance.assignment-4: Destruction complete after 40s
aws_subnet.assignment-4-subnet: Destroying... [id=subnet-00b7177dd8b1364dc]
aws_subnet.assignment-4-subnet: Destruction complete after 1s
aws_vpc.assignment-4-vpc: Destroying... [id=vpc-0edd1d81eee18f354]
aws_vpc.assignment-4-vpc: Destruction complete after 0s

Destroy complete! Resources: 6 destroyed.
ubuntu@terraform:~
```

Terraform will destroy all your managed infrastructure, as shown above.



```
ubuntu@terraform:~$ sudo nano main.tf
```

```
ubuntu@terraform:~$ sudo nano main.tf
```

```
GNU nano 6.2
provider "aws"{
    region = "us-east-2"
    access_key = "AKIAZI2LFI3SGG2NATTZ"
    secret_key = "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"
}
resource "aws_instance" "assignment-5"{
    ami = "ami-0f30a9c3a48f3fa79"
    instance_type = "t2.micro"
    key_name = "devops"
    user_data = <<-EOF
        #!/bin/bash
        apt-get update -y
        apt-get install apache2 -y
        EOF
    tags = {
        Name = "assignment-5"
    }
}
output "instance_ip"{
    value = aws_instance.assignment-5.public_ip
}
resource "local_file" "instance_ip_file"{
    content = aws_instance.assignment-5.public_ip
    filename = "${path.module}/instance_ip.txt"
}
```

### Script 5

```
provider "aws"{
    region = "us-east-2"
    access_key = "AKIAZI2LFI3SGG2NATTZ"
    secret_key = "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"
}
resource "aws_instance" "assignment-5"
```

```
ami = "ami-0f30a9c3a48f3fa79"
instance_type = "t2.micro"
key_name = "devops"
user_data = <<-EOF
#!/bin/bash
apt-get update -y
apt-get install apache2 -y
EOF
tags = {
  Name = "assignment-5"
}
}
output "instance_ip"{
  value = aws_instance.assignment-5.public_ip
}
resource "local_file" "instance_ip_file"{
  content = aws_instance.assignment-5.public_ip
  filename = "${path.module}/instance_ip.txt"
}
```

ubuntu@terraform:~\$ terraform init

```
ubuntu@terraform:~$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Finding latest version of hashicorp/local...
- Using previously-installed hashicorp/aws v5.54.0
- Installing hashicorp/local v2.5.1...
- Installed hashicorp/local v2.5.1 (signed by HashiCorp)

Terraform has made some changes to the provider dependency selections recorded
in the .terraform.lock.hcl file. Review those changes and commit them to your
version control system if they represent changes you intended to make.

Terraform has been successfully initialized!

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

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

```
ubuntu@terraform:~$ terraform plan

Error: Unsupported argument

  on main.tf line 23, in resource "local_file" "instance_ip_file":
23:   conent = aws_instance.assignment-5.public_ip

  An argument named "conent" is not expected here. Did you mean "content"?

ubuntu@terraform:~$
```

```
ubuntu@terraform:~$ sudo nano main.tf

    value = aws_instance.assignment-5.public_ip
}
resource "local_file" "instance_ip_file"{
  content = aws_instance.assignment-5.public_ip
  filename = "${path.module}/instance_ip.txt"
}
```

Save and exit

```
ubuntu@terraform:~$ terraform plan
```



```
ubuntu@terraform:~$ terraform plan

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

Terraform will perform the following actions:

# aws_instance.assignment-5 will be created
+ resource "aws_instance" "assignment-5" {
    + ami                                = "ami-0f30a9c3a48f3fa79"
    + arn                                = (known after apply)
    + associate_public_ip_address        = (known after apply)
    + availability_zone                  = (known after apply)
    + cpu_core_count                     = (known after apply)
    + cpu_threads_per_core              = (known after apply)
    + disable_api_stop                  = (known after apply)
    + disable_api_termination           = (known after apply)
    + ebs_optimized                     = (known after apply)
    + get_password_data                 = false
    + host_id                            = (known after apply)
    + host_resource_group_arn           = (known after apply)
    + iam_instance_profile              = (known after apply)
    + id                                 = (known after apply)
    + instance_initiated_shutdown_behavior = (known after apply)
    + instance_lifecycle                = (known after apply)
    + instance_state                    = (known after apply)
    + instance_type                     = "t2.micro"
    + ipv6_address_count                = (known after apply)
    + ipv6_addresses                    = (known after apply)
    + key_name                           = "devops"
    + monitoring                         = (known after apply)
    + outpost_arn                        = (known after apply)
    + password_data                     = (known after apply)
    + placement_group                   = (known after apply)
    + placement_partition_number        = (known after apply)
    + primary_network_interface_id     = (known after apply)
    + private_dns                        = (known after apply)
}
```



```
ubuntu@terraform:~$ terraform plan

+ tags                                = {
    + "Name" = "assignment-5"
}
+ tags_all                             = {
    + "Name" = "assignment-5"
}
+ tenancy                               = (known after apply)
+ user_data                            = "b13702b5d2f6f88ab23e8831001719500675a1e6"
+ user_data_base64                      = (known after apply)
+ user_data_replace_on_change          = false
+ vpc_security_group_ids               = (known after apply)
}

# local_file.instance_ip_file will be created
+ resource "local_file" "instance_ip_file" {
    + content                            = (known after apply)
    + content_base64sha256                = (known after apply)
    + content_base64sha512                = (known after apply)
    + content_md5                         = (known after apply)
    + content_sha1                        = (known after apply)
    + content_sha256                      = (known after apply)
    + content_sha512                      = (known after apply)
    + directory_permission                = "0777"
    + file_permission                     = "0777"
    + filename                            = "./instance_ip.txt"
    + id                                  = (known after apply)
}

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

Changes to Outputs:
+ instance_ip = (known after apply)

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.
ubuntu@terraform:~$
```

```
ubuntu@terraform:~$ terraform apply
```

```
ubuntu@terraform: ~
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.
ubuntu@terraform:~$ terraform apply

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

Terraform will perform the following actions:

# aws_instance.assignment-5 will be created
+ resource "aws_instance" "assignment-5" {
    + ami                                = "ami-0f30a9c3a48f3fa79"
    + arn                                = "(known after apply)"
    + associate_public_ip_address        = "(known after apply)"
    + availability_zone                  = "(known after apply)"
    + cpu_core_count                     = "(known after apply)"
    + cpu_threads_per_core              = "(known after apply)"
    + disable_api_stop                  = "(known after apply)"
    + disable_api_termination           = "(known after apply)"
    + ebs_optimized                      = "(known after apply)"
    + get_password_data                 = "false"
    + host_id                            = "(known after apply)"
    + host_resource_group_arn           = "(known after apply)"
    + iam_instance_profile              = "(known after apply)"
    + id                                 = "(known after apply)"
    + instance_initiated_shutdown_behavior = "(known after apply)"
    + instance_lifecycle                = "(known after apply)"
    + instance_state                    = "(known after apply)"
    + instance_type                     = "t2.micro"
    + ipv6_address_count                = "(known after apply)"
    + ipv6_addresses                     = "(known after apply)"
    + key_name                           = "devops"
    + monitoring                         = "(known after apply)"
    + outpost_arn                        = "(known after apply)"
    + password_data                      = "(known after apply)"

  }

  # local_file.instance_ip_file will be created
  + resource "local_file" "instance_ip_file" {
    + content                             = "(known after apply)"
    + content_base64sha256                = "(known after apply)"
    + content_base64sha512                = "(known after apply)"

  }

Terraform will perform 2 actions to update your infrastructure.

There are no changes to local files.

Do you want to perform these actions?
y/n [y]:
```

```
ubuntu@terraform: ~
+ instance_type                      = "t2.micro"
+ ipv6_address_count                 = "(known after apply)"
+ ipv6_addresses                      = "(known after apply)"
+ key_name                           = "devops"
+ monitoring                         = "(known after apply)"
+ outpost_arn                        = "(known after apply)"
+ password_data                      = "(known after apply)"
+ placement_group                    = "(known after apply)"
+ placement_partition_number         = "(known after apply)"
+ primary_network_interface_id       = "(known after apply)"
+ private_dns                         = "(known after apply)"
+ private_ip                          = "(known after apply)"
+ public_dns                          = "(known after apply)"
+ public_ip                           = "(known after apply)"
+ secondary_private_ips              = "(known after apply)"
+ security_groups                    = "(known after apply)"
+ source_dest_check                  = "true"
+ spot_instance_request_id           = "(known after apply)"
+ subnet_id                           = "(known after apply)"
+ tags                               = {
    + "Name" = "assignment-5"
  }
+ tags_all                           = {
    + "Name" = "assignment-5"
  }
+ tenancy                            = "(known after apply)"
+ user_data                           = "b13702b5d2f6f88ab23e8831001719500675a1e6"
+ user_data_base64                   = "(known after apply)"
+ user_data_replace_on_change        = "false"
+ vpc_security_group_ids             = "(known after apply"
}

# local_file.instance_ip_file will be created
+ resource "local_file" "instance_ip_file" {
    + content                             = "(known after apply)"
    + content_base64sha256                = "(known after apply)"
    + content_base64sha512                = "(known after apply"

Terraform will perform 2 actions to update your infrastructure.

There are no changes to local files.

Do you want to perform these actions?
y/n [y]:
```

```
ubuntu@terraform: ~
Plan: 2 to add, 0 to change, 0 to destroy.

Changes to Outputs:
+ instance_ip = (known after apply)

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

Enter a value: yes

aws_instance.assignment-5: Creating...
aws_instance.assignment-5: Still creating... [10s elapsed]
aws_instance.assignment-5: Still creating... [20s elapsed]
aws_instance.assignment-5: Still creating... [30s elapsed]
aws_instance.assignment-5: Creation complete after 32s [id=i-01d0644514d2d462b]
local_file.instance_ip_file: Creating...
local_file.instance_ip_file: Creation complete after 0s [id=1d89cd30c210c3ba31a2ff48757ea0a5822a781a]

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

Outputs:

instance_ip = "18.116.60.143"
ubuntu@terraform:~$
```

The screenshot shows the AWS CloudWatch Instances console. At the top, there's a header with tabs for 'Instances (1)' and 'Info'. Below the header, there's a search bar and a filter dropdown set to 'Running'. A table lists one instance: 'assignment-5' with Instance ID 'i-01d0644514d2d462b', status 'Running', type 't2.micro', and a status check of 'Initializing'. There are buttons for 'Launch instances' and a gear icon.

## Outputs:

```
instance_ip = "18.116.60.143"
ubuntu@terraform:~$
```

i-01d0644514d2d462b (assignment-5)

The screenshot shows the AWS Lambda function configuration page for the instance. At the top, there are tabs for 'Details', 'Status and alarms', 'Monitoring', 'Security', 'Networking', 'Storage', and 'Tags'. The 'Details' tab is selected. Under the 'Instance summary' section, there are fields for 'Instance ID' (i-01d0644514d2d462b (assignment-5)), 'Public IPv4 address' (18.116.60.143), 'Private IPv4 addresses' (172.31.8.240), 'IPv6 address' (not shown), 'Instance state' (Running), and 'Public IPv4 DNS' (not shown).