

TERRAFORM ASSIGNMENT-1

Name: Vikram

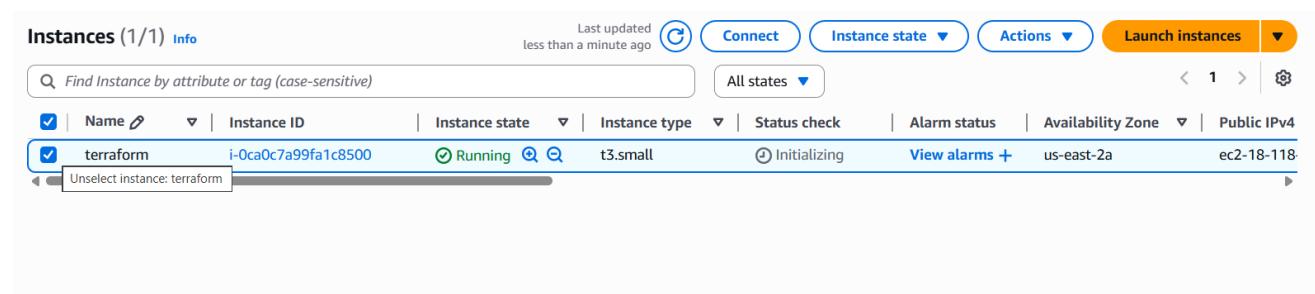
Assignment: Create EC2 Instance in Default Subnet using Terraform (Ohio Region)

Problem Statement

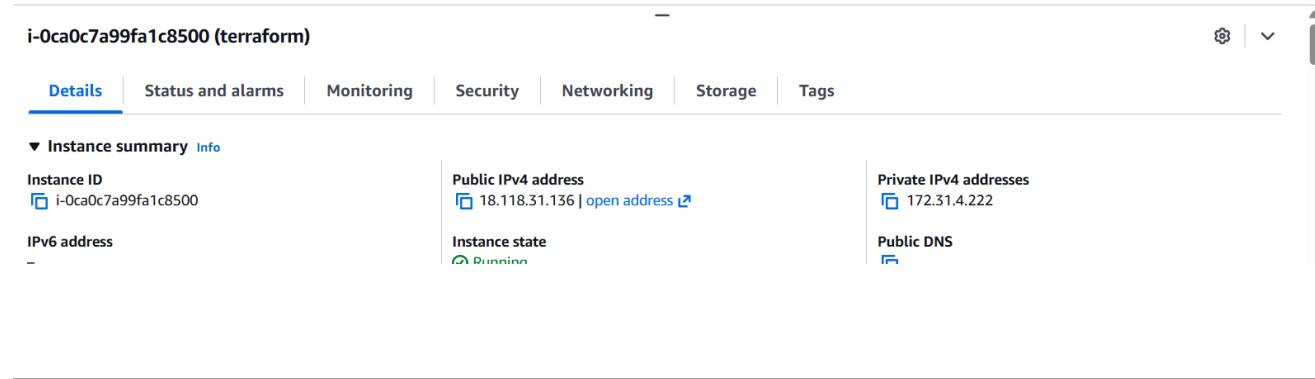
Create an EC2 instance in the default subnet of the Ohio region (us-east-2) using Terraform.

TASK 1: Launch Ubuntu EC2 Instance & Connect

Launch one Ubuntu EC2 instance in ohio and connect using EC2 Instance Connect.



The screenshot shows the AWS CloudWatch Metrics console. A single metric named "Terraform" is displayed with a value of 1. The metric has a timestamp of "Last updated less than a minute ago". Below the metrics, there is a table titled "Metrics" with columns for Metric Name, Value, Unit, and Last Value. The table shows the "Terraform" metric with a value of 1, unit of "Count", and last value of 1.



The screenshot shows the AWS EC2 Instances page. It displays a single instance named "terraform" with the following details: Instance ID: i-0ca0c7a99fa1c8500, Instance state: Running, Instance type: t3.small, Status check: Initializing, Alarm status: View alarms +, Availability Zone: us-east-2a, and Public IPv4: ec2-18-118. The instance was last updated less than a minute ago. There is a "Connect" button and a "Launch instances" button at the top right. Below the table, there is a detailed view for the instance "i-0ca0c7a99fa1c8500 (terraform)". The "Details" tab is selected, showing the instance summary. The instance ID is i-0ca0c7a99fa1c8500, the public IPv4 address is 18.118.31.136, and the private IPv4 address is 172.31.4.222. The instance state is Running, and the public DNS is listed as well.

TASK 2: Install Terraform

```
sudo apt update
```

```
sudo apt install -y gnupg software-properties-common curl
```

```
curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -
```

```
sudo apt-add-repository "deb https://apt.releases.hashicorp.com $(lsb_release -cs) main"
```

```
sudo apt update
```

```
sudo apt install -y terraform
```

```
terraform -version
```

```
ubuntu@ip-10-0-3-201:~$ sudo apt update
sudo apt install -y gnupg software-properties-common curl
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1684 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [311 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [175 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 c-n-f Metadata [15.8 kB]
```

```
ubuntu@ip-10-0-3-201:~$ curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -
sudo apt-add-repository "deb https://apt.releases.hashicorp.com $(lsb_release -cs) main"
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
Repository: 'deb https://apt.releases.hashicorp.com noble main'
Description:
Archive for codename: noble components: main
More info: https://apt.releases.hashicorp.com
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_apt_releases_hashicorp_com-noble.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_apt_releases_hashicorp_com-noble.list
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 https://apt.releases.hashicorp.com noble InRelease [12.9 kB]
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://apt.releases.hashicorp.com/noble/main amd64 Packages [215 kB]
```

```
ubuntu@ip-10-0-3-201:~$ curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -
sudo apt-add-repository "deb https://apt.releases.hashicorp.com $(lsb_release -cs) main"
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
Repository: 'deb https://apt.releases.hashicorp.com noble main'
Description:
Archive for codename: noble components: main
More info: https://apt.releases.hashicorp.com
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_apt_releases_hashicorp_com-noble.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_apt_releases_hashicorp_com-noble.list
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 https://apt.releases.hashicorp.com noble InRelease [12.9 kB]
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 https://apt.releases.hashicorp.com/noble/main amd64 Packages [215 kB]
Fetched 228 kB in 3 (375 kB/s)
Reading package lists... Done
W: https://apt.releases.hashicorp.com/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
ubuntu@ip-10-0-3-201:~$ █
```

```
ubuntu@ip-10-0-3-201:~$ sudo apt update
sudo apt install -y terraform
terraform -version
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://apt.releases.hashicorp.com noble InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
68 packages can be upgraded. Run 'apt list --upgradable' to see them.
W: https://apt.releases.hashicorp.com/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
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 68 not upgraded.
Need to get 30.6 MB of archives.
After this operation, 101 MB of additional disk space will be used.
```

```
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.  
Terraform v1.14.3  
on linux_amd64  
ubuntu@ip-10-0-3-201:~$
```

```
i-0109add2c4a1b622a (terraform)
```

TASK 3: Configure AWS Credentials

```
aws configure
```

Enter:

- AWS Access Key
- AWS Secret Key
- Region: us-east-2
- Output: json

```
ubuntu@ip-172-31-4-222:~$ aws configure  
AWS Access Key ID [None]: AKIAQ47TESP3CDYCXWWX  
AWS Secret Access Key [None]: 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  
Default region name [None]: us-east-2  
Default output format [None]: json  
ubuntu@ip-172-31-4-222:~$
```

TASK 4: Create Terraform Project

```
mkdir terraform-ec2
```

```
cd terraform-ec2
```

```
nano main.tf
```

main.tf

```
provider "aws" {  
  region = "us-east-2"  
}  
  
resource "aws_instance" "my_ec2" {
```

```
ami      = "ami-0f5fcdfbd140e4ab7"
instance_type = "t3.micro"

tags = {
  Name = "Terraform-EC2"
}

}
```

Save and exit

```
ubuntu@ip-172-31-4-222:~$ mkdir terraform-ec2
cd terraform-ec2
ubuntu@ip-172-31-4-222:~/terraform-ec2$ nano main.tf
```

```
GNU nano 7.2
provider "aws" {
  region = "us-east-2"
}

resource "aws_instance" "my_ec2" {
  ami           = "ami-0f5fcdfbd140e4ab7"
  instance_type = "t3.micro"

  tags = {
    Name = "Terraform-EC2"
  }
}
```

TASK 5: Initialize Terraform

terraform init

```
ubuntu@ip-172-31-4-222:~/terraform-ec2$ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.27.0...
- Installed hashicorp/aws v6.27.0 (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@ip-172-31-4-222:~/terraform-ec2$
```

TASK 6: Validate Configuration

terraform validate

```
ubuntu@ip-172-31-4-222:~/terraform-ec2$ terraform validate
Success! The configuration is valid.
```

```
ubuntu@ip-172-31-4-222:~/terraform-ec2$
```

TASK 7: Create EC2 Instance

terraform plan

terraform apply

Type yes when asked.

```
ubuntu@ip-172-31-4-222:~/terraform-ec2$ 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.my_ec2 will be created
+ resource "aws_instance" "my_ec2" {
    + ami = "ami-0f5fcdfbd140e4ab7"
    + arn = "(known after apply)"
    + associate_public_ip_address = "(known after apply)"
    + availability_zone = "(known after apply)"
    + disable_api_stop = "(known after apply)"
    + disable_api_termination = "(known after apply)"
    + ebs_optimized = "(known after apply)"
    + enable_primary_ipv6 = "(known after apply)"
    + force_destroy = false
    + 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)"}
```

i-0ca0c7a99fa1c8500 (terraform)

Public IPs: 18.118.31.136 Private IPs: 172.31.4.222

```

ubuntu@ip-172-31-4-222:~/terraform-ec2$ 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.my_ec2 will be created
+ resource "aws_instance" "my_ec2" {
    + ami                               = "ami-0f5fcdfbd140e4ab7"
    + arn                               = (known after apply)
    + associate_public_ip_address      = (known after apply)
    + availability_zone                = (known after apply)
    + disable_api_stop                 = (known after apply)
    + disable_api_termination          = (known after apply)
    + ebs_optimized                    = (known after apply)
    + enable_primary_ipv6              = (known after apply)
    + force_destroy                    = false
    + 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)
}

+ private_dns_name_options (known after apply)

+ root_block_device (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.my_ec2: Creating...
aws_instance.my_ec2: Still creating... [00m10s elapsed]
aws_instance.my_ec2: Creation complete after 12s [id=i-01b993b2abbc6632c]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-4-222:~/terraform-ec2$ 
```

Verification

Check in AWS Console → EC2 → Instances

Instance named Terraform-EC2 is created in Ohio (us-east-2) and placed in default subnet.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, Global View, Instances, Launch Templates, Requests, Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager, Pages, MIs, and Catalog. The main area has a title 'Instances (1/2) Info'. It displays a table with one row for the instance 'Terraform-EC2'. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. The instance details panel below shows the instance ID (i-01b993b2abbc6632c), its public IP (52.14.15.165), IAM Role (None), IMDSv2 (Required), and various ARNs and IDs related to the instance's configuration.

Conclusion

Successfully created EC2 instance in the default subnet of Ohio region using Terraform.