**Terraform Task -1**

**Task Description:**

Launch Linux EC2 instances in two regions using a single Terraform file.

**Techstacks needs to be used:**

* AWS EC2
* Terraform
* AWS CLI

**How do I submit my work?**

* Push all your work files to GitHub (Code & O/P screenshot images must).
* Submit your URLs in the portal.

**Terms and Conditions?**

* You agree to not share this confidential document with anyone.
* You agree to open-source your code (it may even look good on your profile!). Do not mention our company’s name anywhere in the code.
* We will never use your source code under any circumstances for any commercial purposes; this is just a basic assessment task.

**NOTE:** Any violation of Terms and conditions is strictly prohibited. You are bound to adhere to it.

**Solutions:**

**1. Objective**

To launch two Linux EC2 instances in **two different AWS regions** (Mumbai and N. Virginia) using a **single Terraform configuration file** and validate successful deployment through AWS Console and CLI.

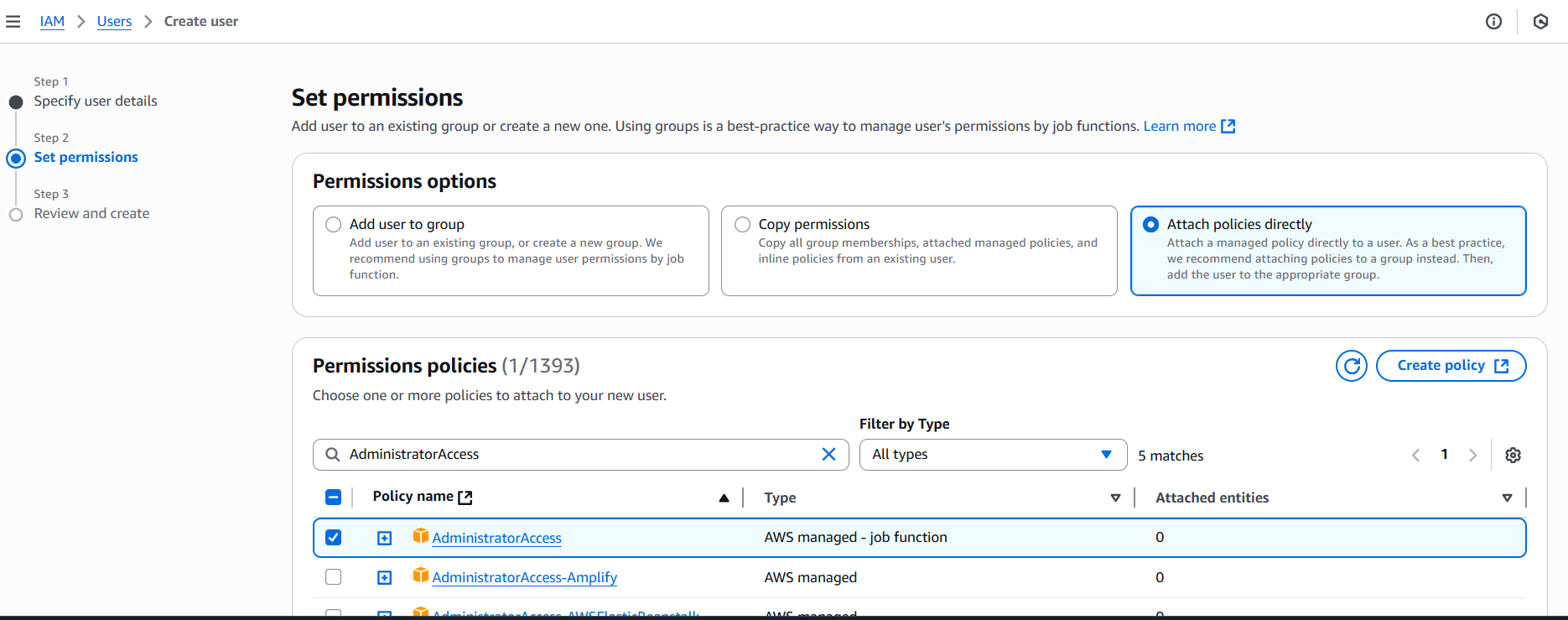
**2. Tools & Technologies Used**

* **AWS CLI** – to configure and authenticate with AWS account.
* **Terraform v1.13.3** – for Infrastructure as Code (IaC).
* **Git Bash** – to execute Terraform and Git commands on Windows.
* **Git & GitHub** – for version control and submission.
* **Amazon EC2** – to host virtual machines.

**3. Prerequisites**

1. AWS account with IAM user having AdministratorAccess.
2. AWS CLI configured with Access Key and Secret Key.
3. Terraform installed and verified using terraform -version.
4. Git installed and working.

**Create IAM USER with access key and secret key**



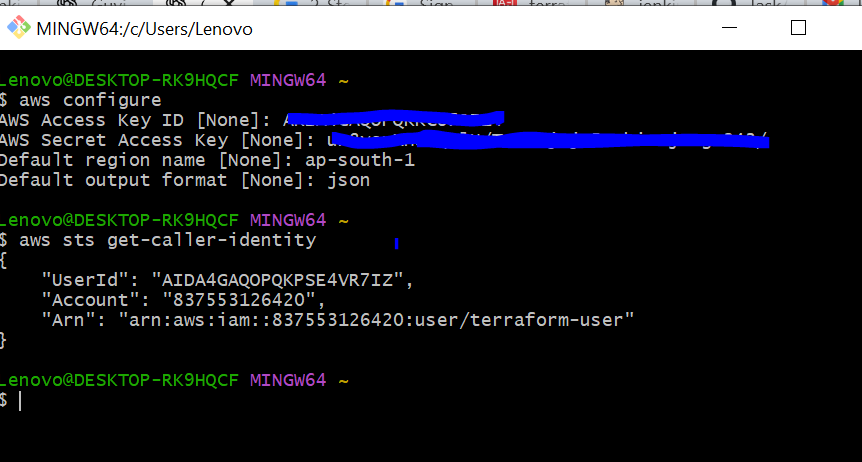
Step 1 — Configure your AWS CLI

Connection successful

Terraform setup

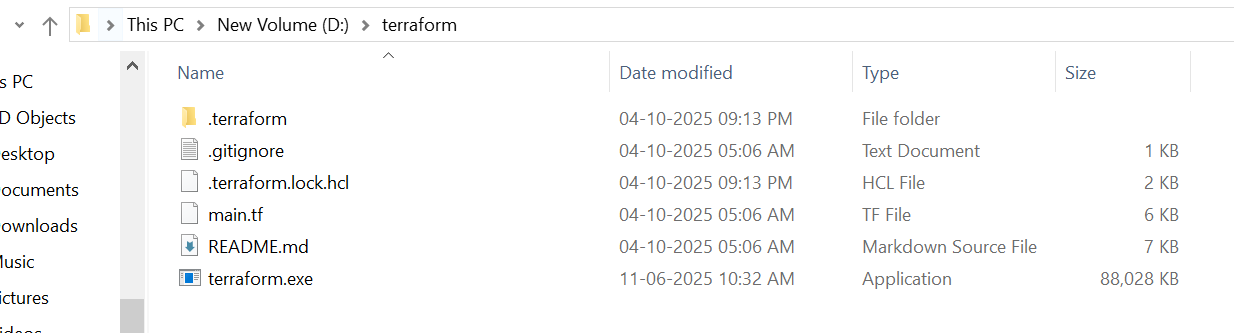
**4. Steps Followed**

1. Step 1: Create IAM User and Configure AWS CLI
2. Created a new IAM user named terraform-user in AWS IAM.
3. Enabled programmatic access (CLI).
4. Assigned **AdministratorAccess** policy.
5. Created an Access Key and Secret Key.
6. Configured AWS CLI using:



**Step 2: Install and Verify Terraform**

1. Downloaded Terraform for Windows from the HashiCorp website.
2. Extracted and placed terraform.exe in C:\Windows\System32.
3. Verified installation using:



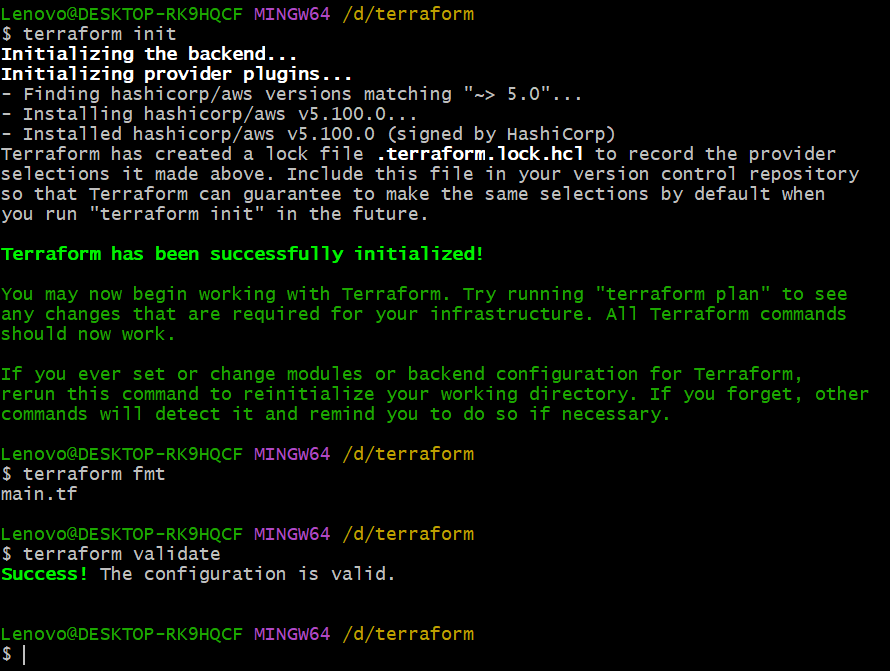
### ****Step 3: Downloaded and Opened Terraform Project****

* Downloaded the provided **terraform-multi-region-ec2.zip** file.
* Extracted the folder and opened **Git Bash** inside it.
* Initialized Terraform:

terraform init

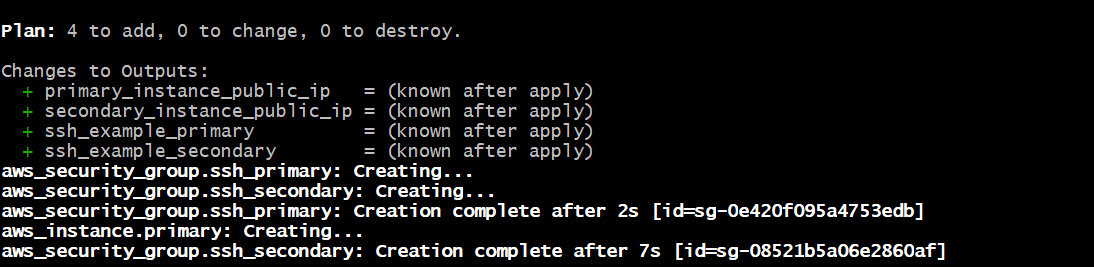
terraform fmt

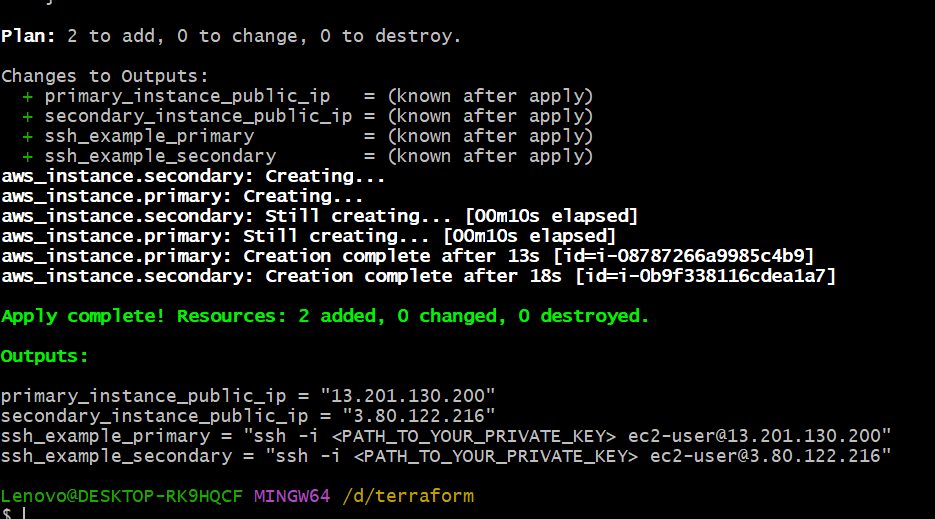
terraform validate



* All commands executed successfully.

**— Launch Two EC2 Instances**

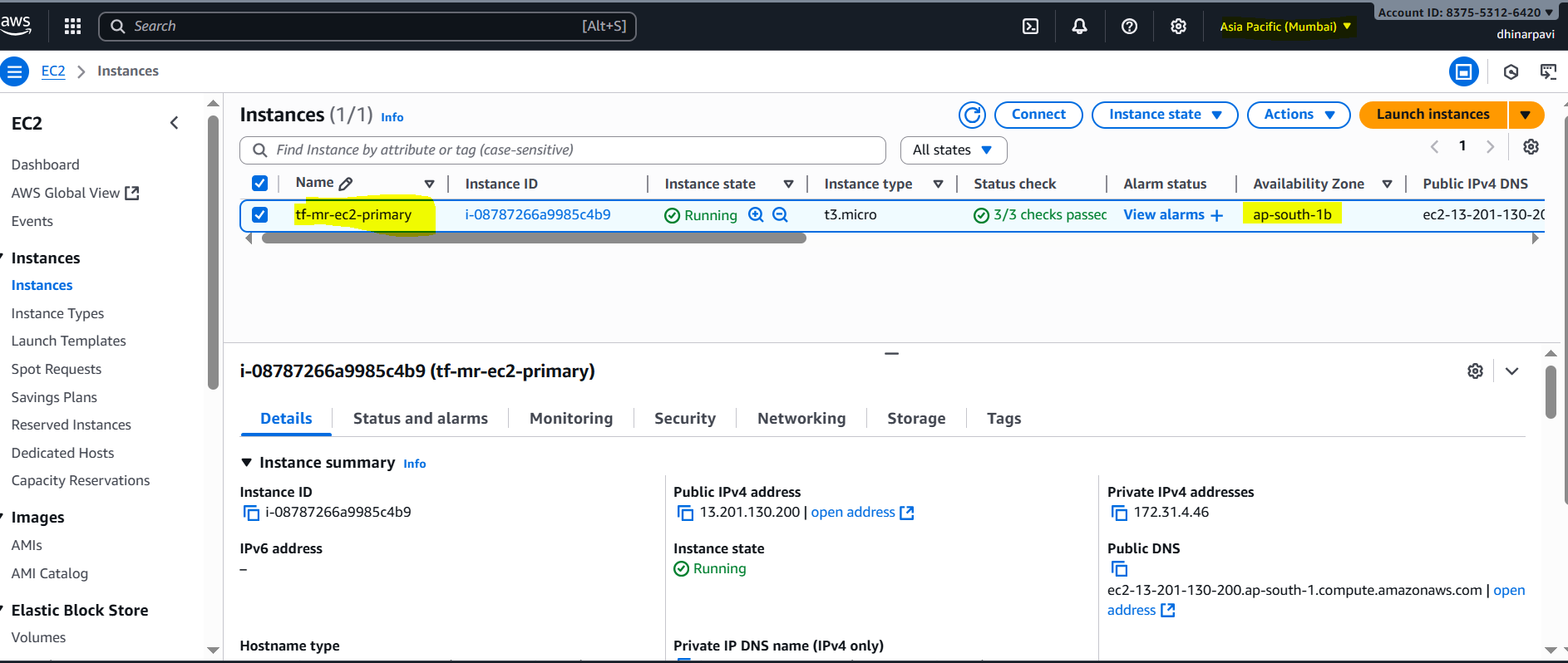




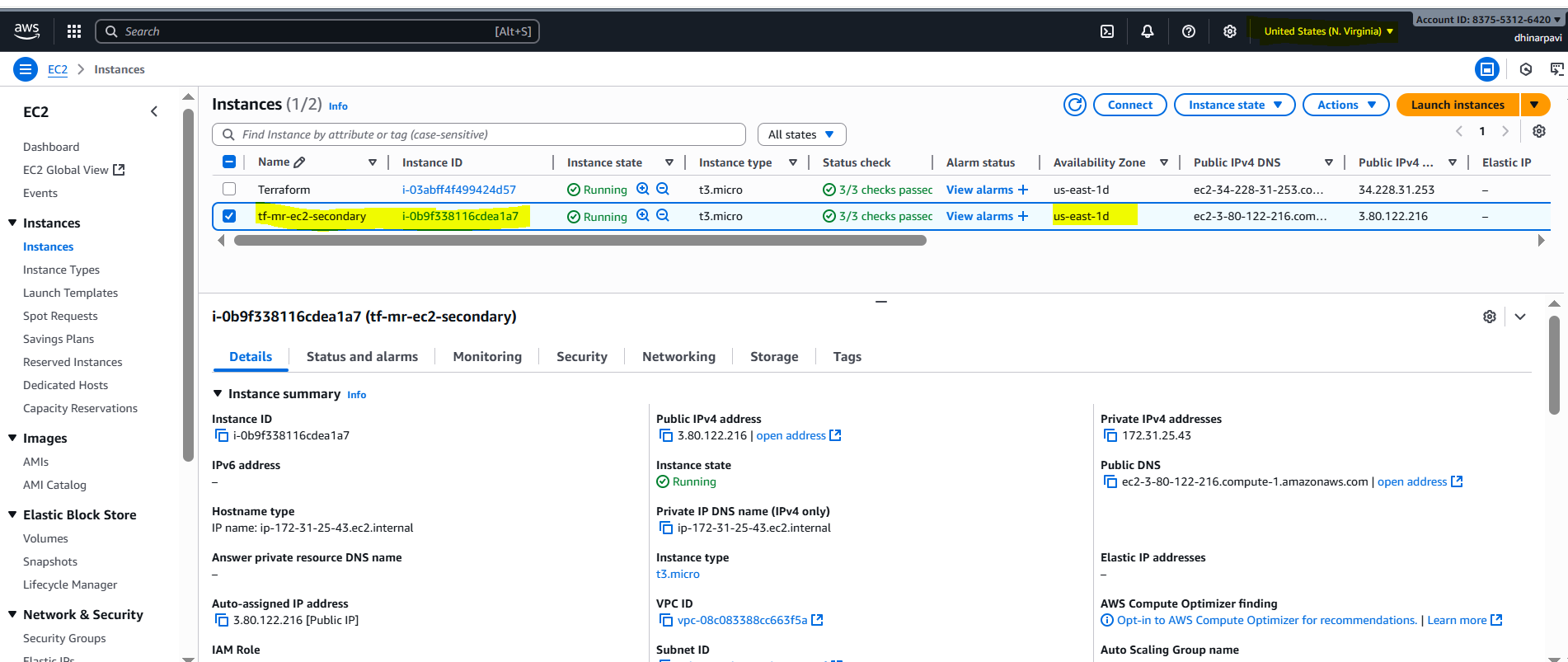
### ****Step 5: Verification in AWS Console****

* Logged into AWS Console → EC2 → Instances.
* Switched regions to verify both instances:
* **ap-south-1 (Mumbai)** → Instance: *tf-mr-ec2-primary*
* **us-east-1 (N. Virginia)** → Instance: *tf-mr-ec2-secondary*
* Both instances were in “running” state with assigned public IPs.

**ap-south-1 (Mumbai)** → Instance: *tf-mr-ec2-primary*



**us-east-1 (N. Virginia)** → Instance: *tf-mr-ec2-secondary*



Task completed successfully.  
Two Linux EC2 instances were created in different AWS regions and verified in the AWS console.