

Assignment 2:

Working with EC2 instances

Create a EC2 instance

Connect to the instance

01main.tf x

C: > terraform > 01main.tf

```
1  provider "aws" {
2      region      = "us-east-2"
3      access_key  = "AKIA6INCOCVQAXSMPY55"
4      secret_key  = "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"
5  }
6
7  resource "aws_instance" "cloudknowledgeindia" {
8      ami          = "ami-00dfe2c7ce89a450b"
9      instance_type = "t2.micro"
10
11     tags = {
12         name = "assignment2"
13     }
14 }
```


C:\Windows\System32\cmd.exe

Microsoft Windows [Version 10.0.18362.295]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\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 v3.58.0

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.

C:\terraform>



Type here to search




```
C:\Windows\System32\cmd.exe
- http_endpoint
- http_put_response_hop_limit = (known after apply)
- http_tokens = (known after apply)
}

- network_interface {
- delete_on_termination = (known after apply)
- device_index = (known after apply)
- network_interface_id = (known after apply)
}

+ root_block_device {
+ delete_on_termination = (known after apply)
+ device_name = (known after apply)
+ encrypted = (known after apply)
+ iops = (known after apply)
+ kms_key_id = (known after apply)
+ tags = (known after apply)
+ throughput = (known after apply)
+ volume_id = (known after apply)
+ volume_size = (known after apply)
+ volume_type = (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.cloudknowledgeindia: Creating...
aws_instance.cloudknowledgeindia: Still creating... [10s elapsed]
aws_instance.cloudknowledgeindia: Still creating... [20s elapsed]
aws_instance.cloudknowledgeindia: Still creating... [30s elapsed]
aws_instance.cloudknowledgeindia: Still creating... [40s elapsed]
aws_instance.cloudknowledgeindia: Still creating... [50s elapsed]
aws_instance.cloudknowledgeindia: Creation complete after 50s [id=i-0ff8518009c802ea2]
```

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

C:\terraform>



Type here to search



ENG
IN


```

- ephemeral_block_device {
  - device_name = (known after apply)
  - no_device   = (known after apply)
  - virtual_name = (known after apply)
}

+ metadata_options {
  + http_endpoint
  + http_put_response_hop_limit = (known after apply)
  + http_tokens                 = (known after apply)
}

+ network_interface {
  + delete_on_termination = (known after apply)
  + device_index          = (known after apply)
  + network_interface_id  = (known after apply)
}

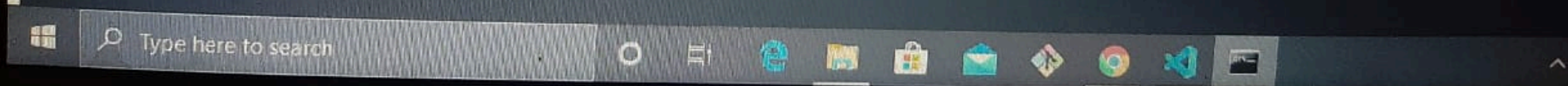
+ root_block_device {
  + delete_on_termination = (known after apply)
  + device_name           = (known after apply)
  + encrypted             = (known after apply)
  + iops                  = (known after apply)
  + kms_key_id            = (known after apply)
  + tags                  = (known after apply)
  + throughput            = (known after apply)
  + volume_id             = (known after apply)
  + volume_size           = (known after apply)
  + volume_type           = (known after apply)
}
}

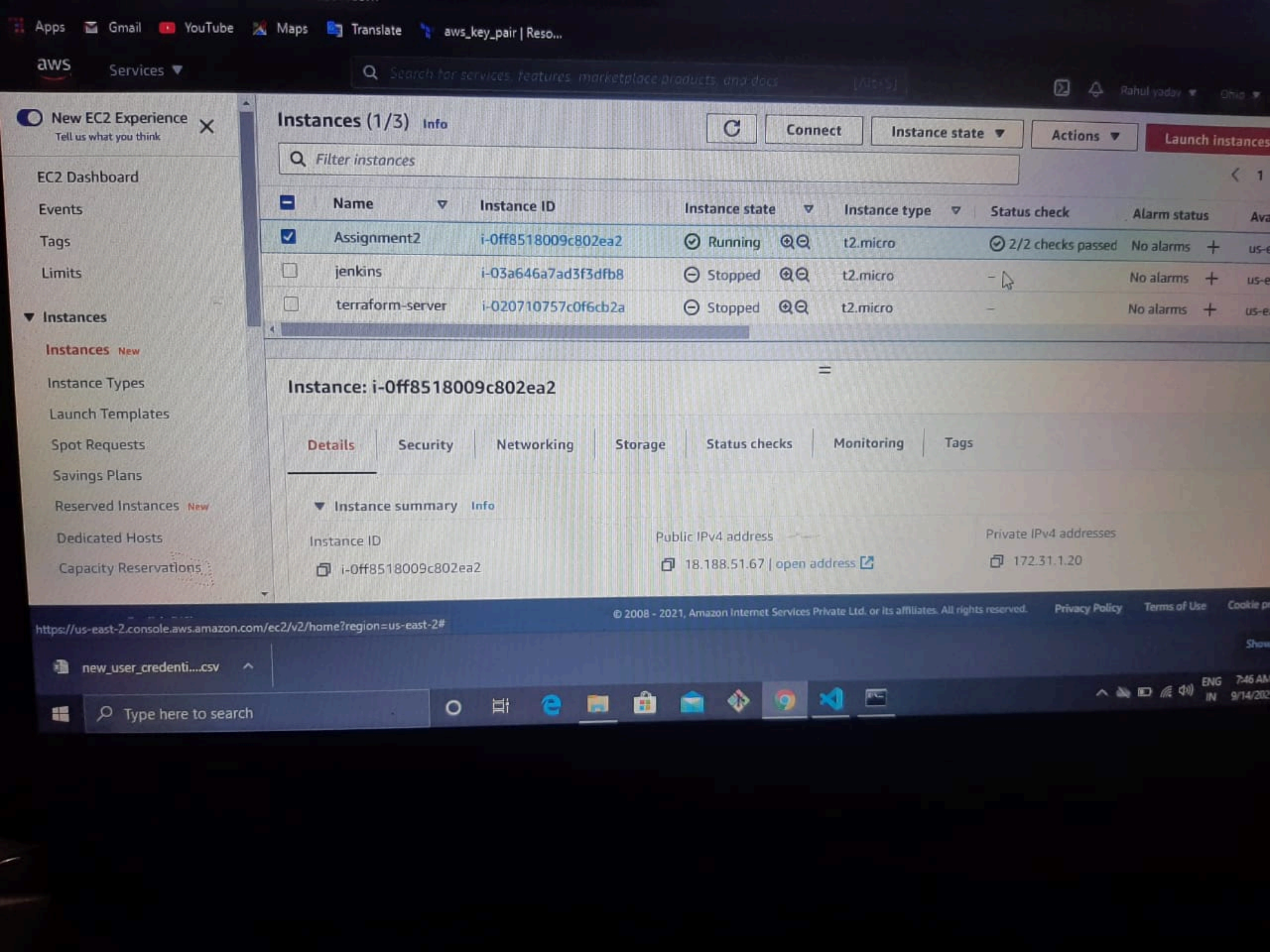
```

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 ap

C:\terraform>terraform apply





ec2-user@ip-172-31-44-114:~

rahu1@DESKTOP-PORGFUH MINGW64 ~/Downloads
\$

rahu1@DESKTOP-PORGFUH MINGW64 ~/Downloads
\$ ssh -i terraform-key.pem ec2-user@18.225.35.238

The authenticity of host '18.225.35.238 (18.225.35.238)' can't be established.
ED25519 key fingerprint is SHA256:W/kQ1y4Ih2KBG1YdzgstLAEhIRLKNZKjnG6k0TG1190.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:1: 13.58.53.10
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '18.225.35.238' (ED25519) to the list of known hosts.
Last login: Mon Sep 6 07:48:23 2021 from 112.79.121.31

```
  _ | _ | _ )  
  _ | ( /  
  _ | \ | _ |
```

Amazon Linux 2 AMI

<https://aws.amazon.com/amazon-linux-2/>
[ec2-user@ip-172-31-44-114 ~]\$