

Question 6 : Terraform

Table of Content

Creating the Q_6_TF project.

Fig 1.1: After creating Q_6_TF.tf file, populating the config with nano.	2
Fig 1.2: Initializing the populated TF project and validating the file.	2
Fig 1.3: The instance is created in TF from the local VM.	3
Fig 1.4: Instance i-0b7093992caaede47 is created on AWS and visible on the web portal.	3

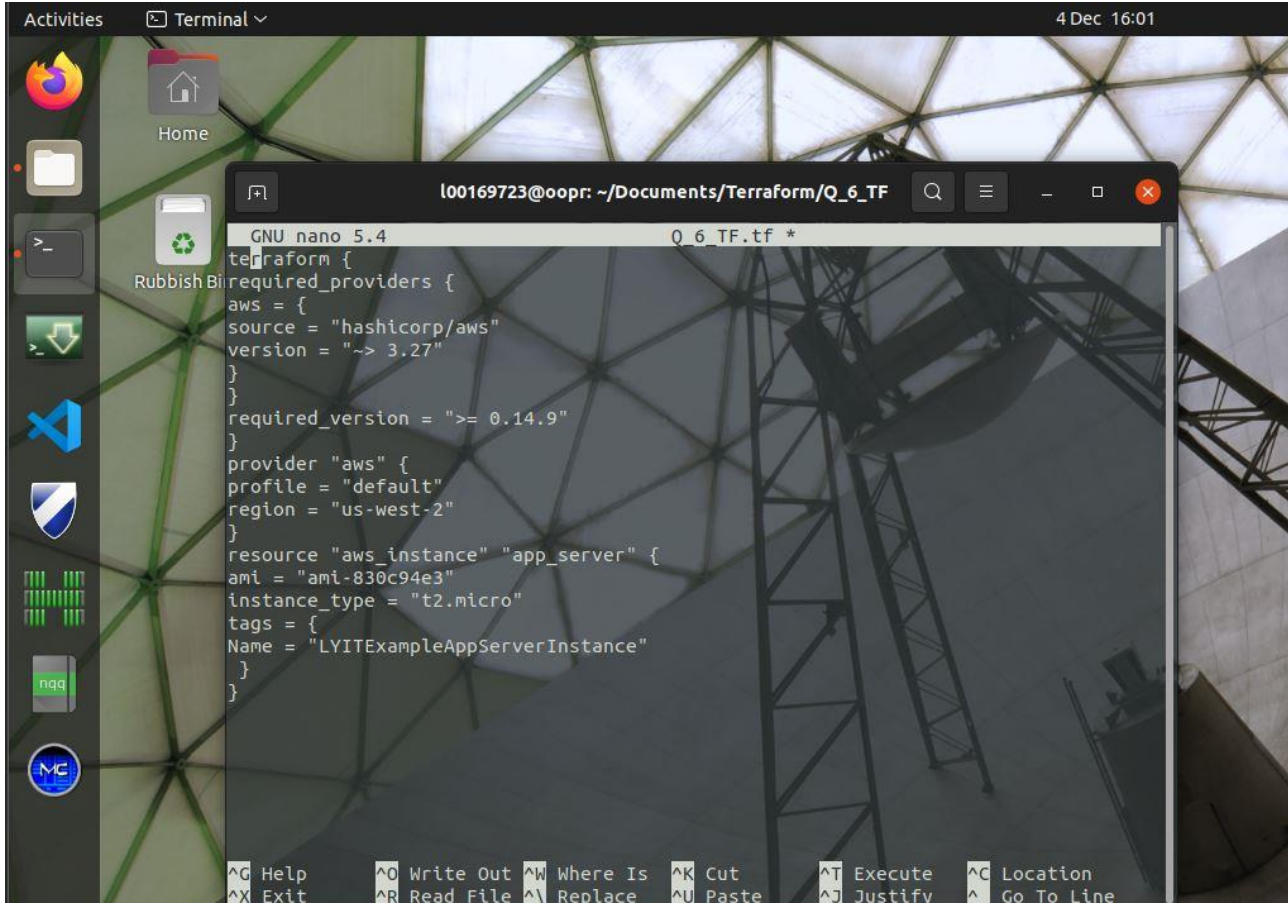
Creating variables.

Fig 2.1: Updated the AMI in the Q_6_TF.tf file. Changes were applied.	4
Fig 2.2: Modified region field to region = var.region.	4
Fig 2.3: Receiving error messages.	5
Fig 2.4: Refreshed Terraform config and the previous instance was terminated.	
Instance i-094aa05bdcb352fd2 was created. Created and modified variables.tf.	5
Fig 2.5: Changes reflected in AWS Instances.	5
Fig 2.6: Output of "terraform graph" of the latest i-094aa05bdcb352fd2 instance.	6
Fig 2.7: Updated Q_6_TF.tf file.	6
Fig 2.8: Updated variables.tf file.	7

Outputs.tf and Displaying IP

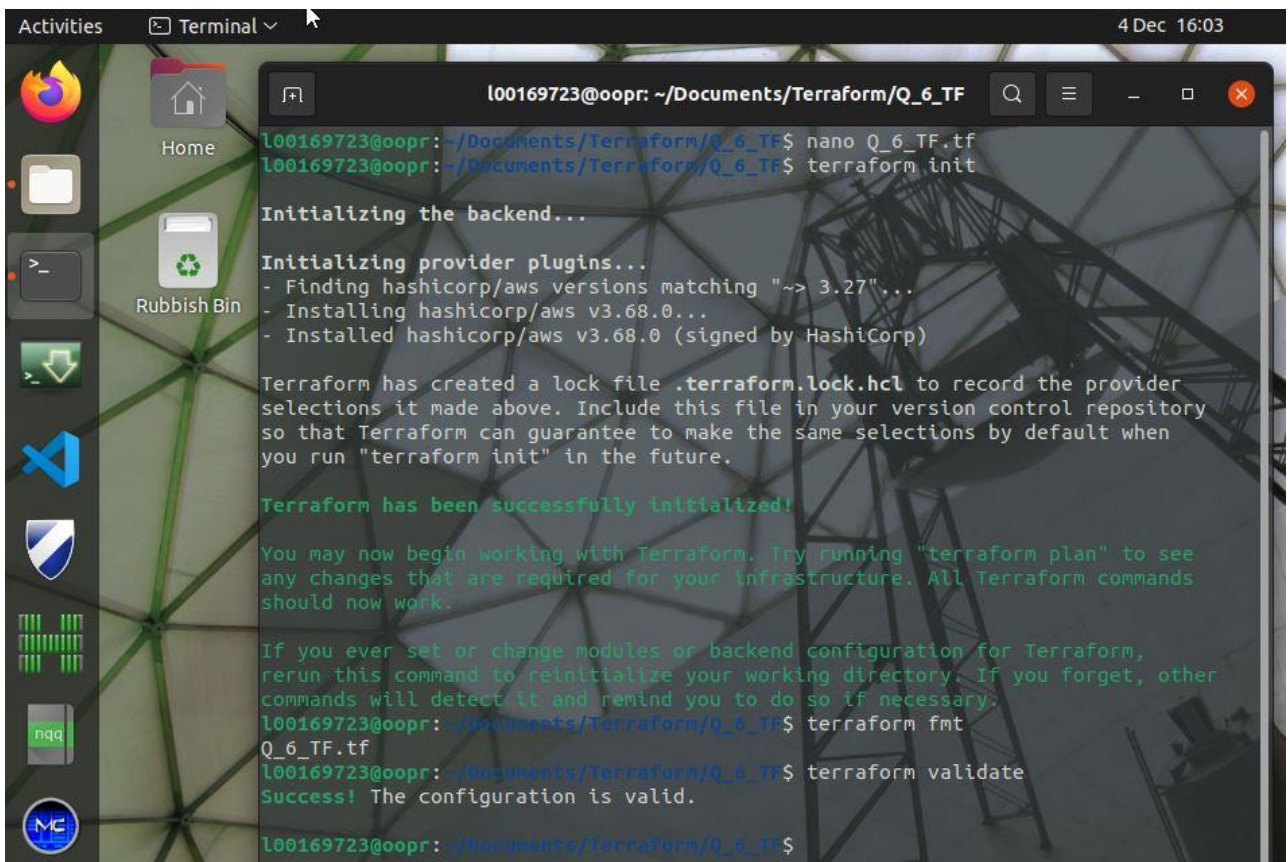
Fig 3.1: Created outs.tf file.	8
Fig 3.2: Refreshed config, output is displayed at the last step after applied the new config.	8
Fig 3.3: Running terraform output command on local VM.	9
Fig 3.4: AWS displays the latest instance with the IP address for cross-reference.	9

1.: Creating the Q_6_TF project.



```
GNU nano 5.4 Q_6_TF.tf *
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "~> 3.27"
    }
  }
  required_version = ">= 0.14.9"
}
provider "aws" {
  profile = "default"
  region = "us-west-2"
}
resource "aws_instance" "app_server" {
  ami = "ami-830c94e3"
  instance_type = "t2.micro"
  tags = {
    Name = "LYITExampleAppServerInstance"
  }
}
```

Fig 1.1: After creating Q_6_TF.tf file, populating the config with nano.



```
l00169723@oopr: ~/Documents/Terraform/Q_6_TF
l00169723@oopr:~/Documents/Terraform/Q_6_TF$ nano Q_6_TF.tf
l00169723@oopr:~/Documents/Terraform/Q_6_TF$ terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "~> 3.27"...
- Installing hashicorp/aws v3.68.0...
- Installed hashicorp/aws v3.68.0 (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.
l00169723@oopr: ~/Documents/Terraform/Q_6_TF$ terraform fmt
Q_6_TF.tf
l00169723@oopr:~/Documents/Terraform/Q_6_TF$ terraform validate
Success! The configuration is valid.

l00169723@oopr:~/Documents/Terraform/Q_6_TF$
```

Fig 1.2: Initializing the populated TF project and validating the file.

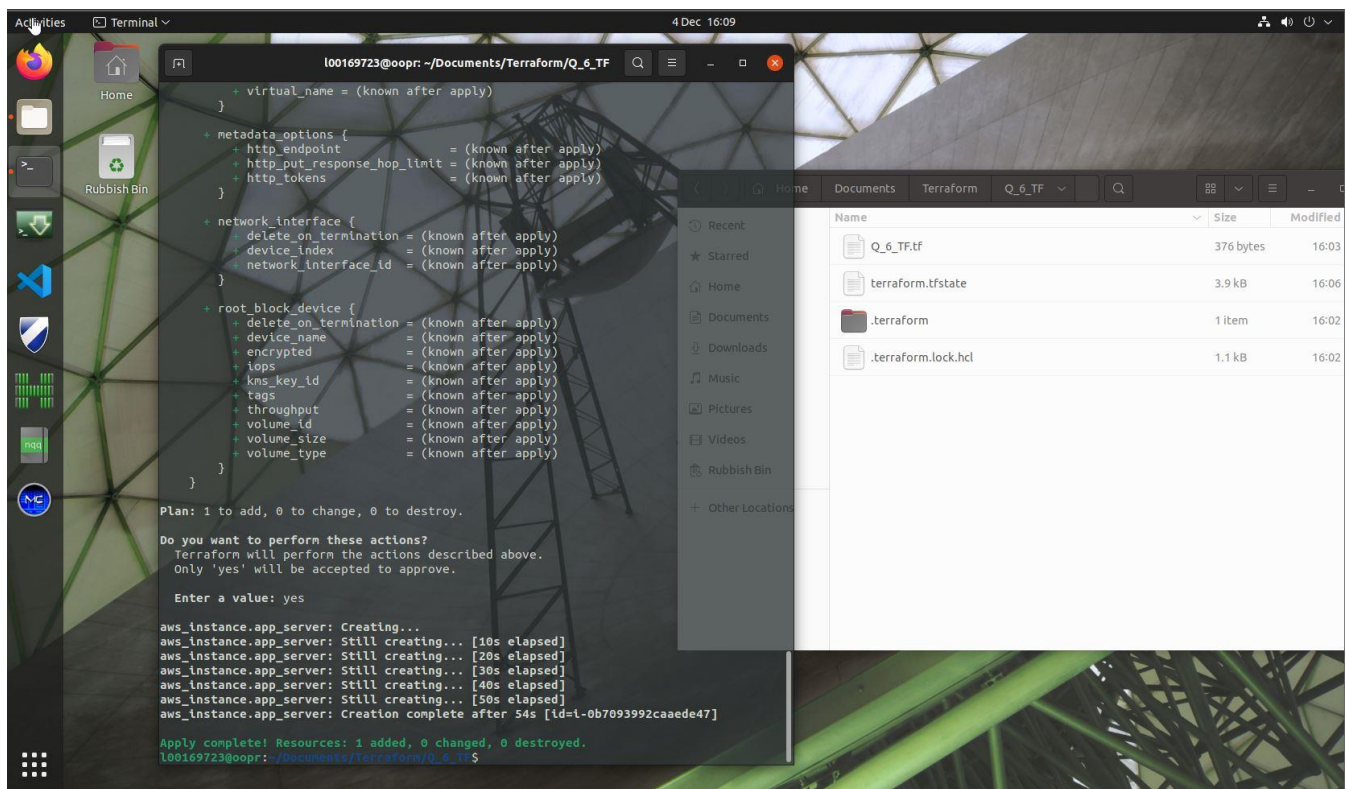


Fig 1.3: The instance is created in TF from the local VM.

Instances (2) [Info](#)

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	LYITExampleA...	i-02c0aa31a658f1c8e	Terminated	t2.micro	-	No alarms +
<input type="checkbox"/>	LYITExampleA...	i-0b7093992caaede47	Running	t2.micro	Initializing	No alarms +

Fig 1.4: Instance i-0b7093992caaede47 is created on AWS and visible on the web portal.

2.: Creating variables.



```
Q_6_TF.tf x
home > l00169723 > Documents > Terraform > Q_6_TF > Q_6_TF.tf > ...
1 terraform
2 required_providers {
3   aws = {
4     source = "hashicorp/aws"
5     version = "~> 3.27"
6   }
7 }
8 required_version = ">= 0.14.9"
9 }
10 provider "aws" {
11   profile = "default"
12   region = "us-west-2"
13 }
14 resource "aws_instance" "app_server" {
15   ami = "ami-08d70e59c07c61a3a"
16   instance_type = "t2.micro"
17   tags = {
18     Name = "LYITExampleAppServerInstance"
19   }
20 }
21 }
```

```
~ throughput          = 0 -> (known after apply)
~ volume_id           = "vol-0eb01431023a84c3d" -> (known after apply)
~ volume_size         = 8 -> (known after apply)
~ volume_type         = "standard" -> (known after apply)
}

Plan: 1 to add, 0 to change, 1 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.
l00169723@oopr:~/Documents/Terraform/Q_6_TF$ terraform apply
```

Fig 2.1: Updated the AMI in the Q_6_TF.tf file. Changes were applied.

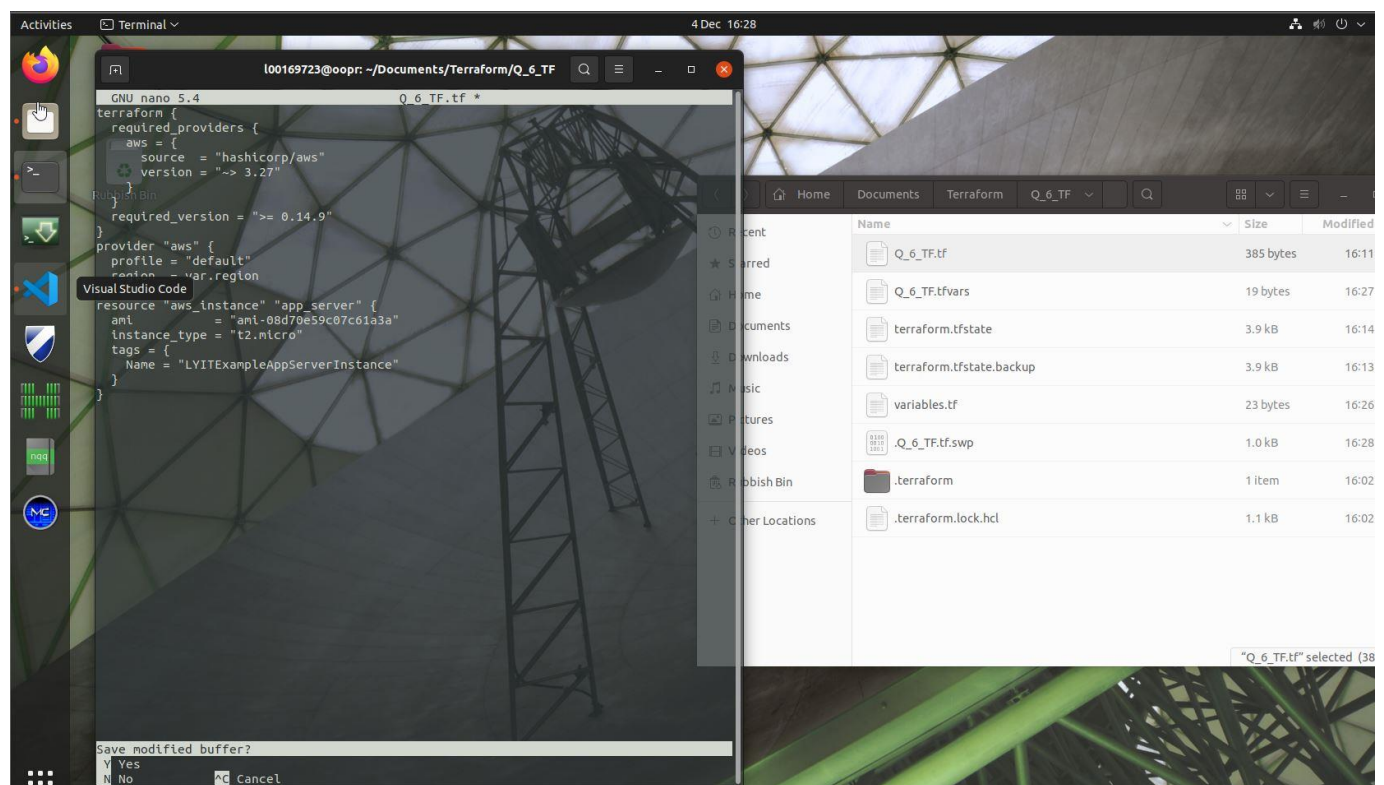


Fig 2.2: Modified region field to region = var.region.

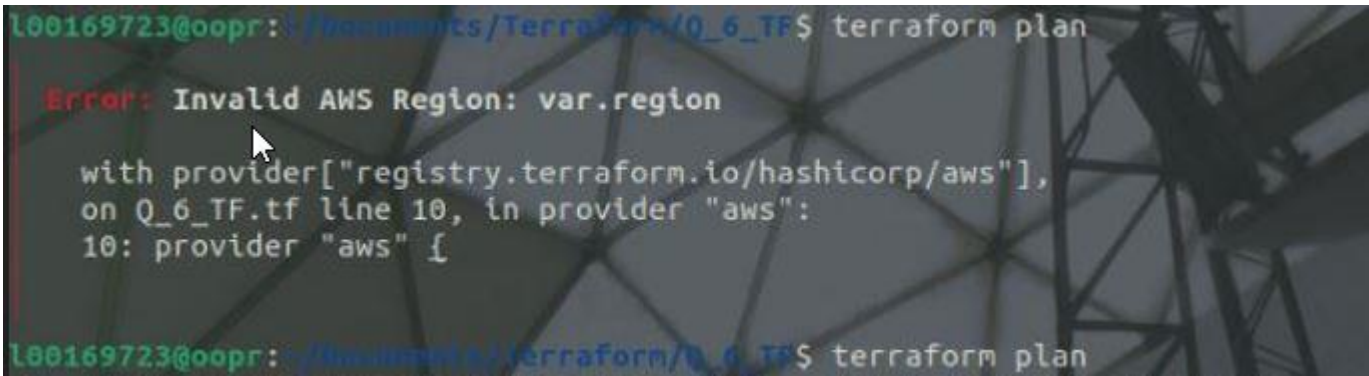


Fig 2.3: Receiving error messages.

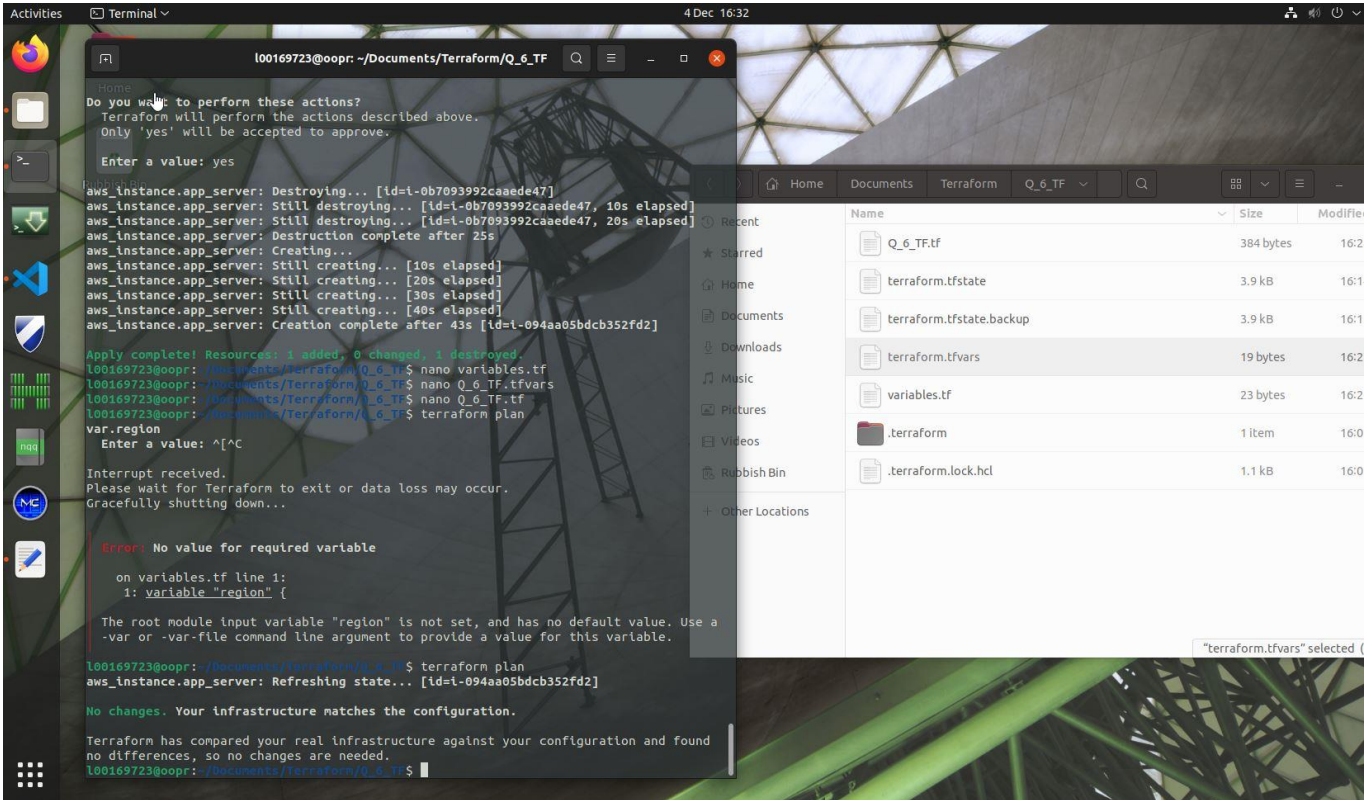


Fig 2.4: Refreshed Terraform config and the previous instance was terminated. Instance i-094aa05bdc352fd2 was created. Created and modified variables.tf.

Instances (3) Info								Connect
Filter instances								
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	
<input type="checkbox"/>	LYITExampleA...	i-02c0aa31a658f1c8e	Terminated	t2.micro	—	No alarms	us-west-2c	
<input type="checkbox"/>	LYITExampleA...	i-0b7093992caaede47	Terminated	t2.micro	—	No alarms	us-west-2b	
<input type="checkbox"/>	LYITExampleA...	i-094aa05bdc352fd2	Running	t2.micro	—	No alarms	us-west-2b	

Fig 2.5: Changes reflected in AWS Instances.

```
100169723@oopr:~/Documents/Terraform/Q_6_TF$ terraform plan
aws_instance.app_server: Refreshing state... [id=i-094aa05bdc352fd2]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found
no differences, so no changes are needed.
100169723@oopr:~/Documents/Terraform/Q_6_TF$ terraform graph
digraph
{
  compound = "true"
  newrank = "true"
  subgraph "root" {
    "[root] aws_instance.app_server (expand)" [label = "aws_instance.app_server", shape = "box"]
    "[root] provider[\"registry.terraform.io/hashicorp/aws\"]" [label = "provider[\"registry.terraform.io/hashicorp/aws\"]", shape = "diamond"]
    "[root] var.region" [label = "var.region", shape = "note"]
    "[root] aws_instance.app_server (expand)" -> "[root] provider[\"registry.terraform.io/hashicorp/aws\"]"
    "[root] meta.count-boundary (EachMode fixup)" -> "[root] aws_instance.app_server (expand)"
    "[root] provider[\"registry.terraform.io/hashicorp/aws\"] (close)" -> "[root] aws_instance.app_server (expand)"
    "[root] provider[\"registry.terraform.io/hashicorp/aws\"]" -> "[root] var.region"
    "[root] root" -> "[root] meta.count-boundary (EachMode fixup)"
    "[root] root" -> "[root] provider[\"registry.terraform.io/hashicorp/aws\"] (close)"
  }
}
```

Fig 2.6: Output of "terraform graph" of the latest i-094aa05bdc352fd2 instance.

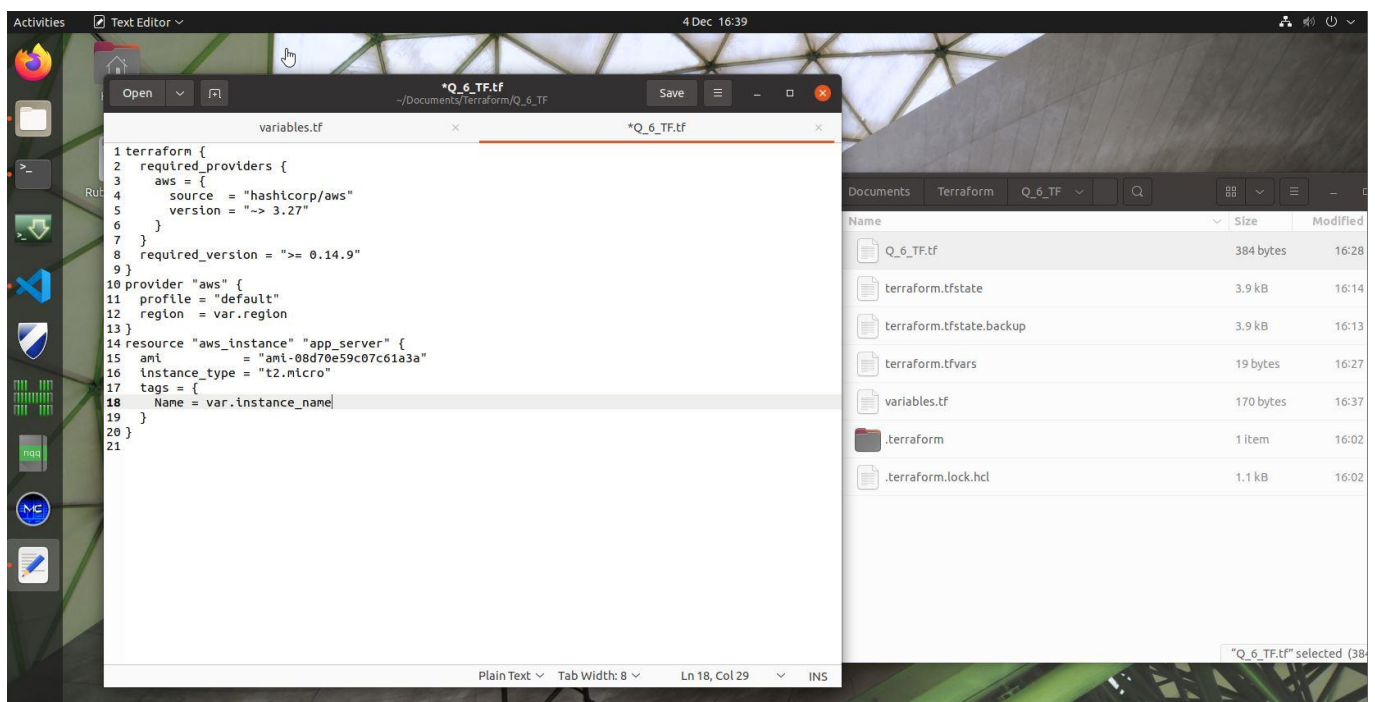


Fig 2.7: Updated Q_6_TF.tf file.

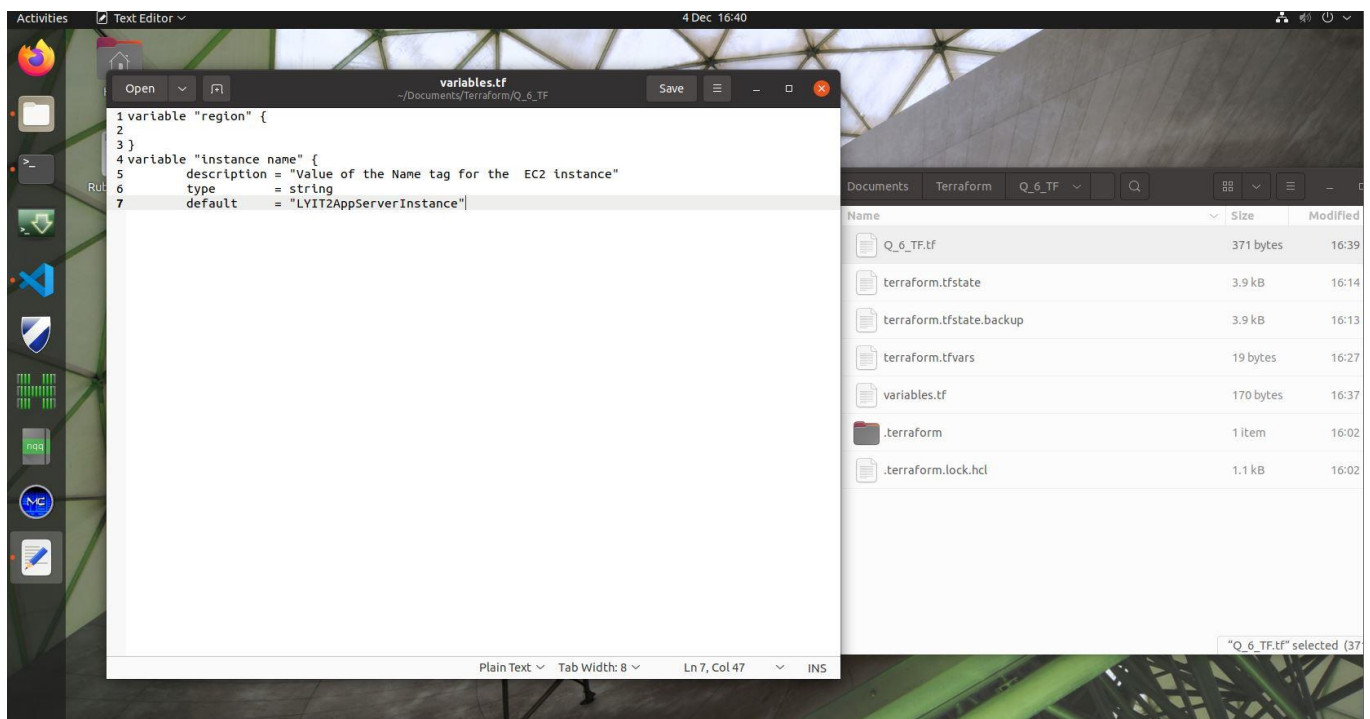


Fig 2.8: Updated variables.tf file.

3.: Outputs.tf and Displaying IP



Fig 3.1: Created outs.tf file.

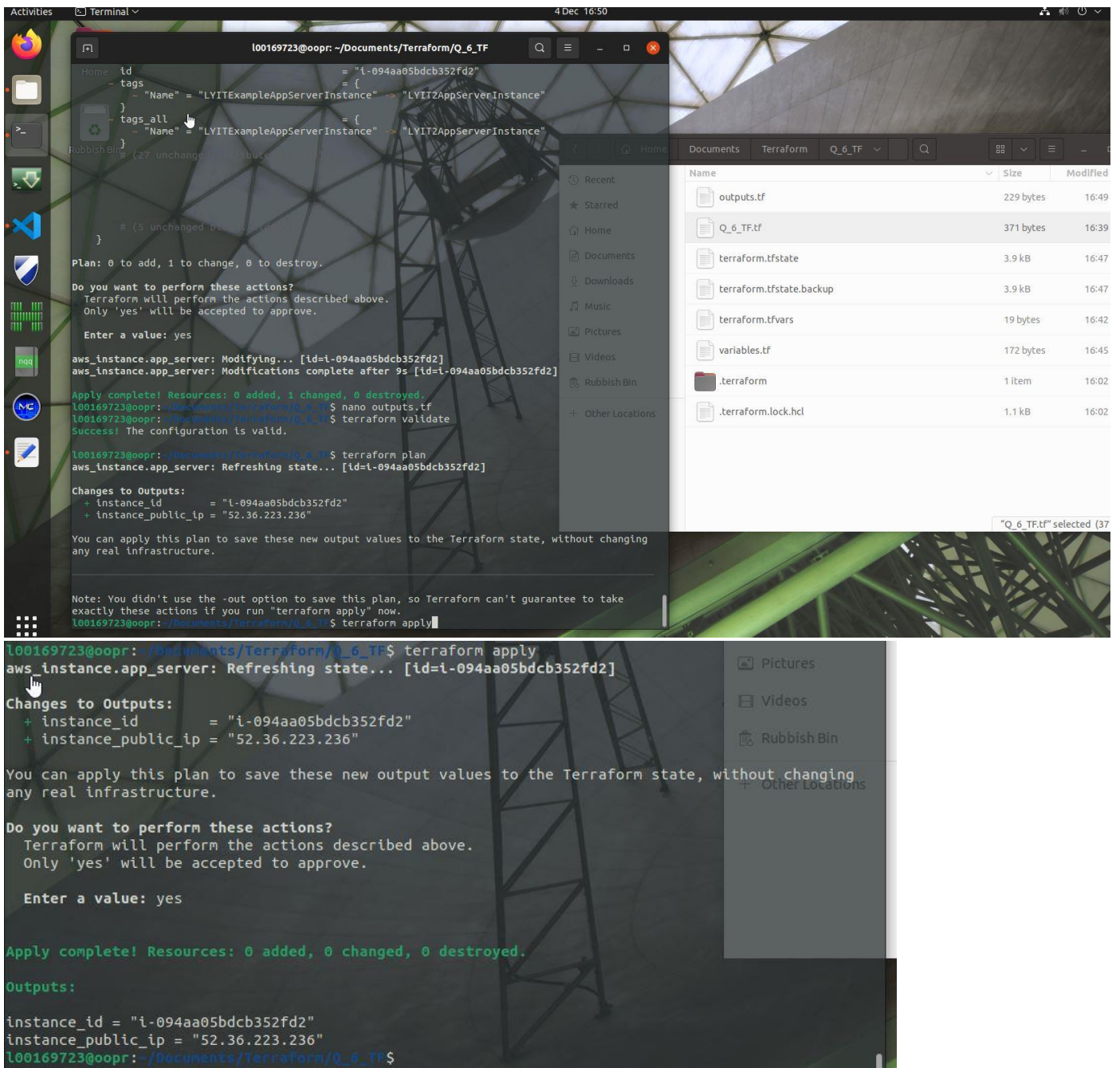


Fig 3.2: Refreshed config, output is displayed at the last step after applied the new config.

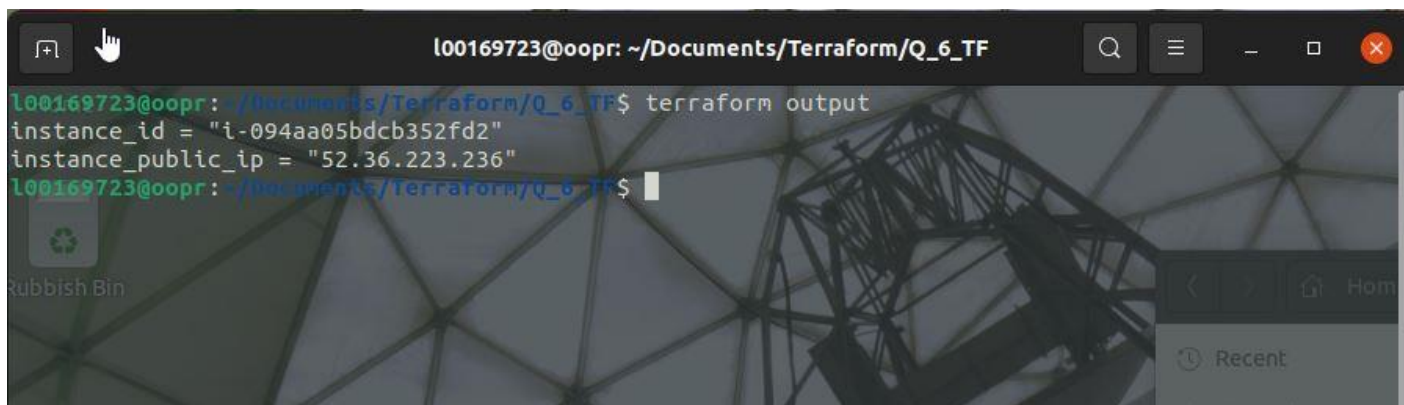
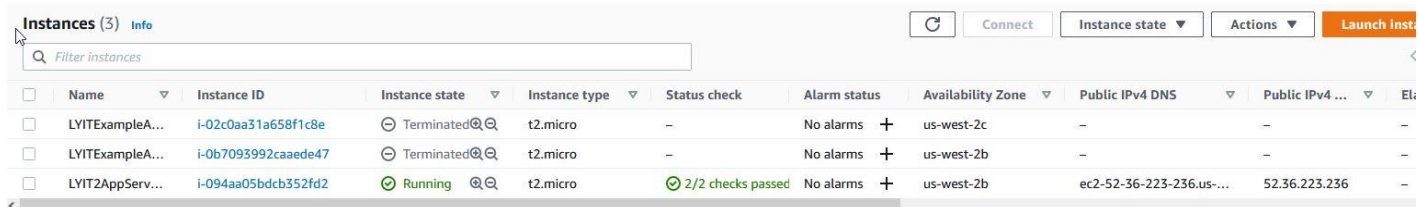
A terminal window titled 'l00169723@oopr: ~/Documents/Terraform/Q_6_TF'. The prompt is 'l00169723@oopr: ~/Documents/Terraform/Q_6_TF\$'. The command 'terraform output' has been executed, resulting in two lines of output: 'instance_id = "i-094aa05bdc352fd2"' and 'instance_public_ip = "52.36.223.236"'. The prompt is now 'l00169723@oopr: ~/Documents/Terraform/Q_6_TF\$' with a cursor. In the background, a desktop environment is visible with a 'Rubbish Bin' icon and a 'Recent' list.

Fig 3.3: Running terraform output command on local VM.

The AWS Management Console 'Instances' page. It shows a table of EC2 instances. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, and Public IPv4 address. Three instances are listed. The first two are 'Terminated' and the third is 'Running'. The 'Running' instance is highlighted in green. The 'Public IPv4 address' column shows the IP address '52.36.223.236' for the running instance.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
LYITExampleA...	i-02c0aa31a658f1c8e	Terminated	t2.micro	—	No alarms	us-west-2c	—	—
LYITExampleA...	i-0b7093992caaede47	Terminated	t2.micro	—	No alarms	us-west-2b	—	—
LYIT2AppServ...	i-094aa05bdc352fd2	Running	t2.micro	2/2 checks passed	No alarms	us-west-2b	ec2-52-36-223-236.us-...	52.36.223.236

Fig 3.4: AWS displays the latest instance with the IP address for cross-reference.