|  |  |  |
| --- | --- | --- |
| Automate infrastructure in AWS cloud | | |
| Team Number | 62 | |
| Team Names | Kamalatha poluparthi | Maha Swetha chirla |
| V.D.Jyothi Gumpena | Sai Lakshmi pogiri |
| Mentor Name | Shaifu/Vijay | |
| Team SPOC Number | Kamalatha poluparthi | |
| Field: AWS platform, cloud formation service, terraform(DevOps tool) | | |

**Introduction:**

WHAT IS AWS?

Aws stands for Amazon web services. It is an evolving and world’s largest cloud computing platform, provided by Amazon. It is more flexible, Scalable and cost efficient. AWS provides 4 types of cloud services namely: software as a service (SAAS), platform as a service (PAAS) , Infrastructure as a service(IAAS),Anything as a service( XAAS).Most used AWS services are like Amazon EC2 instance, RDS , VPC ,Cloud watch and so on.

WHAT IS CLOUD INFRASTRUCTURE?

Cloud Infrastructure is a collection of Hardware and software elements needed to enable cloud computing.it includes computing power networking and storage as well as an interface for users to access their initialized resources. This concept comes under IAAS. AWS provides clients pay-as-you-go access to storage, networking, servers, and other computing resources in the cloud. It is user choice to select the applications and Operating system. Hardware and power resources are maintained by AWS.

For brief understanding on infrastructure let us consider an example, Launching an instance with specific configurations like RAM, storage, network ,AMI(Amazon machine Image), security groups. These all configurations and resources combinedly forms an infrastructure.



**Scope of the project:**

Actually, Infrastructure is built by humans manually. sometimes it may take short or long period of time for building an infrastructure, that is completely depends upon the requirements of the client.

Problem

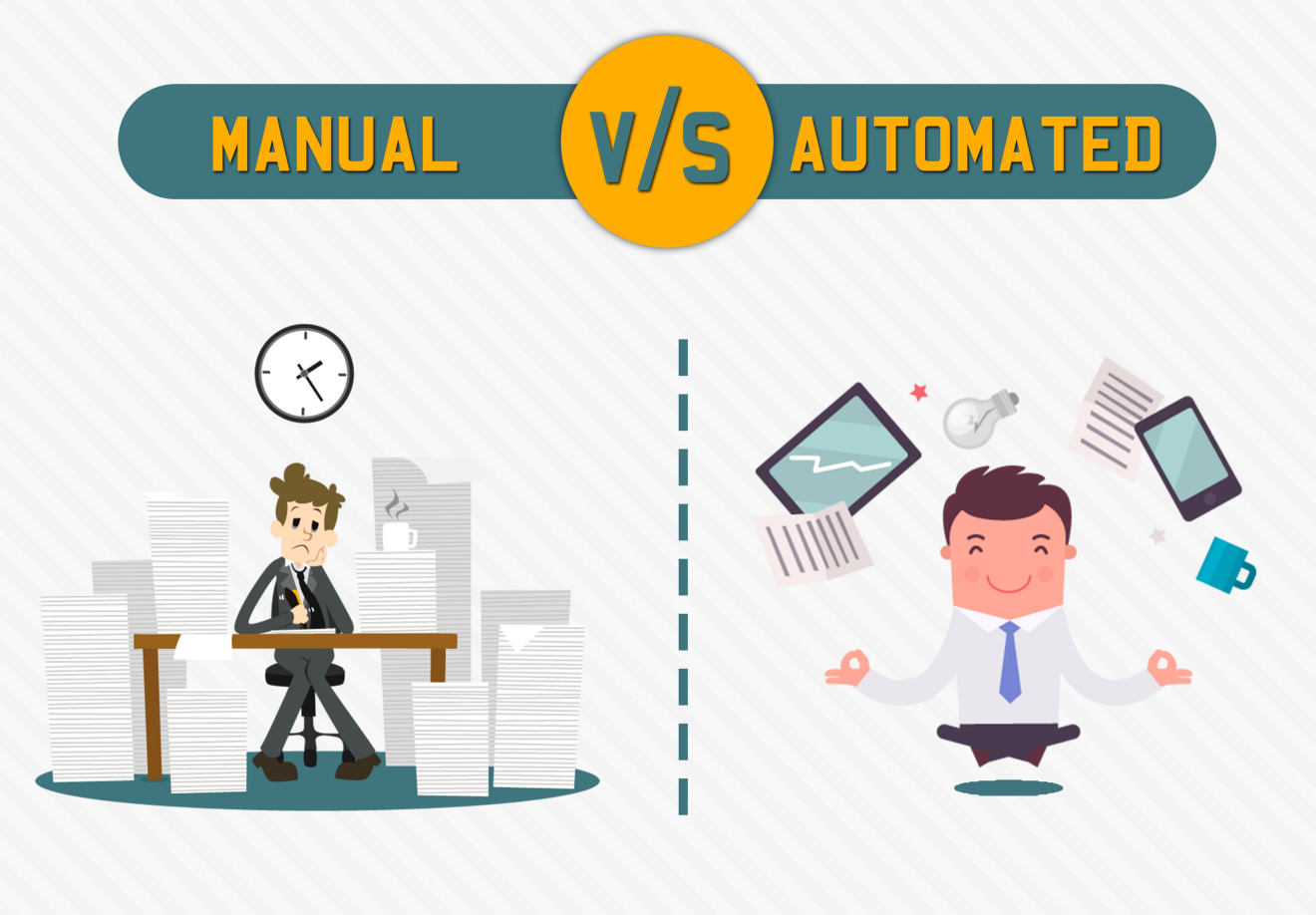
If user needs to create 1000 virtual servers. Manually it is very difficult because, it is hard to choose each and every configuration and resource.it is time consuming process, and there might be a chance of getting human errors.

Solution is this project!

The main motto of this project is to Automating the infrastructure. Through automation no Human Errors arises, needs very short period of time for launching any number of infrastructures.

Automation can be done in two ways:

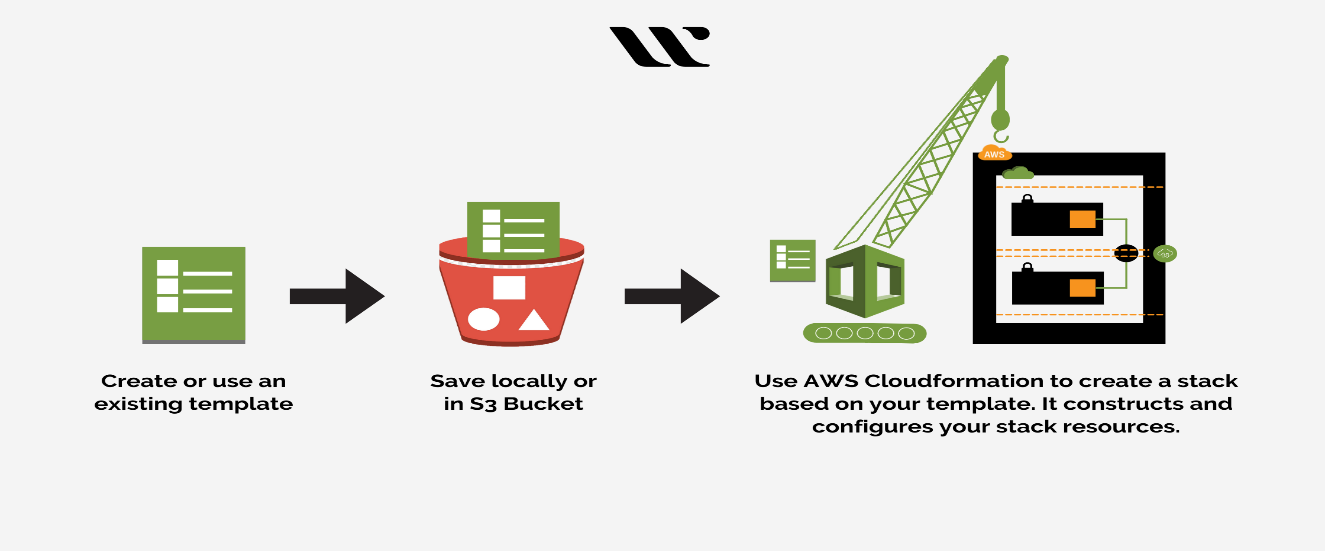
1. Cloud formation (AWS service)
2. Terraform (DevOps tool)



Strategy to accomplish the project:

1.Automate infrastructure using cloud formation

* AWS Cloud Formation is a service that helps you model and set up your AWS resources so that you can spend less time managing those resources that run in Aws.
* You create a template that describes all the AWS resources and configurations that you want (like Amazon EC2 instances and its configuration).
* Cloud Formation takes care of provisioning and configuring those resources for you. You don't need to individually create and configure AWS resources.
* Templates must be in the form of YAML or JSON formatted file. Resources are managed in a single unit called stack. we can create, update, delete stacks.
* You can work with Stack by attaching template .so that we can automate the infrastructure by the specified specifications in template.



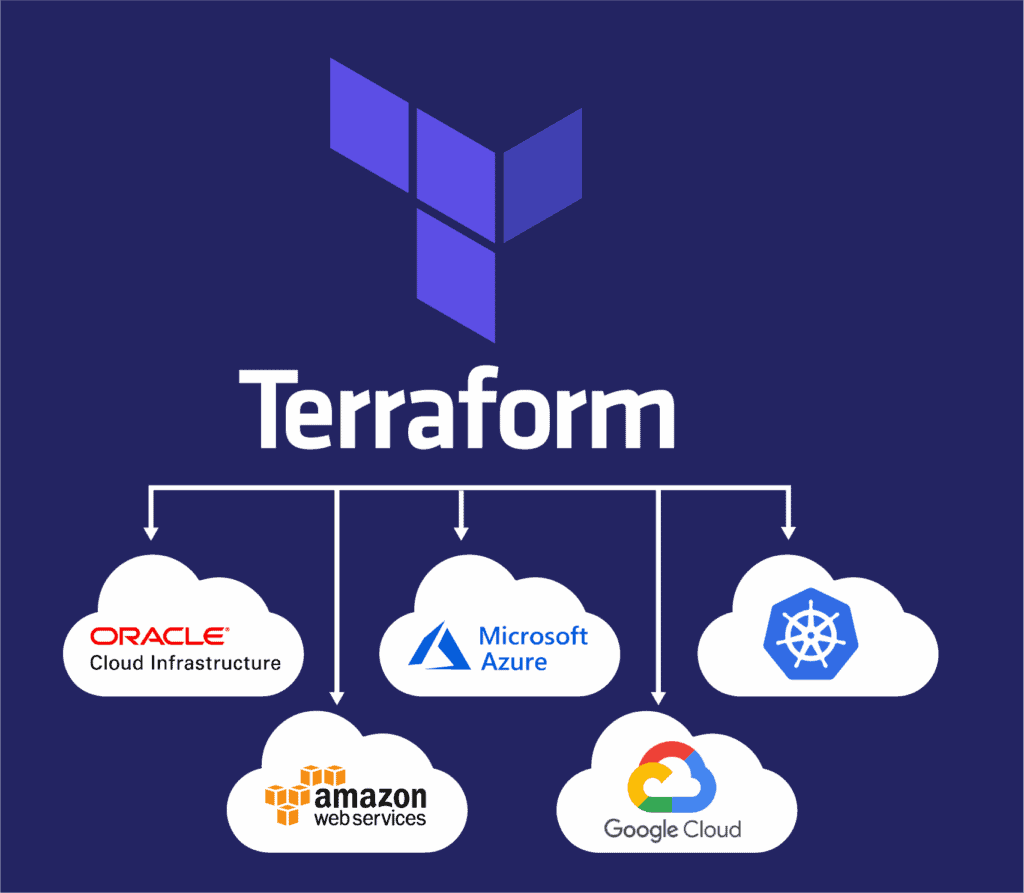
DRAWBACKS

* As it is confined to AWS platform.so other cloud service platforms like azure, Kubernetes etc. Because it is internal service of AWS.
* It is not possible to 100% check a template locally, it is only possible to be certain of syntactic correctness without first executing it.
* At least, not always; in fact, there are many configurations and/or resources that cannot be specified in an AWS CloudFormation template.
* Need permissions to create instances.

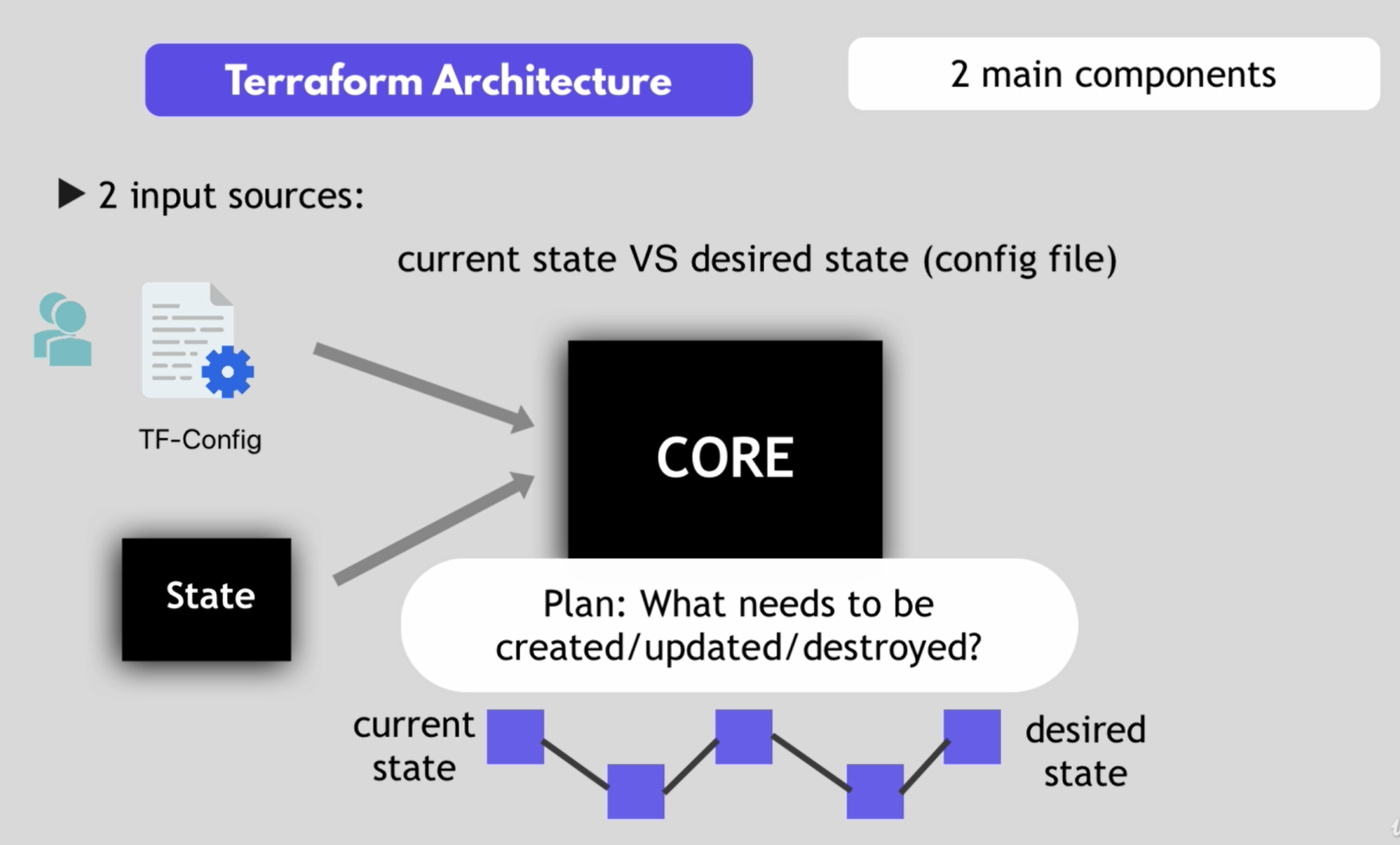
**To overcome above drawbacks, terraform, a DevOps tool, comes into the picture!**

2.Automate infrastructure using Terraform

* Terraform is a open-source, flexible and it allows us to automate, manage the infrastructure and also services that run on the platform.
* Terraform supports all cloud vendors like AWS, azure, Kubernetes and many other.
* Terraform is a declarative nature. It directly defines end state of the system.
* Terraform is a tool for infrastructure provisioning which prepares the server for application deployment.



Terraform Architecture

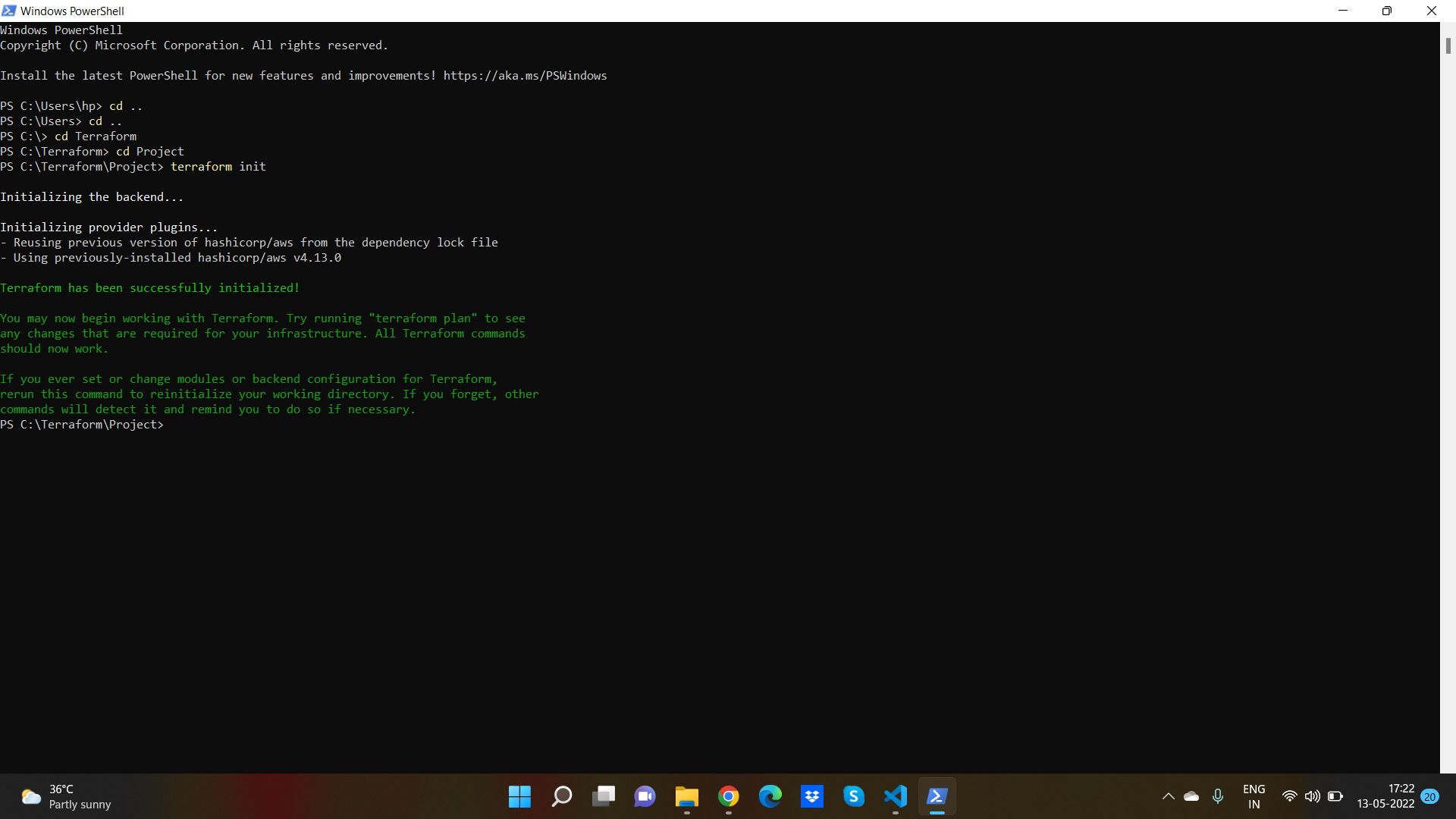


* Terraform architecture is having 2 main components
* CORE-it is having 2 input sources
* State And TF configuration file
* User need to define what needs to be created, update, destroy.
* PROVIDERS:
* Provider is just code that knows how to communicate to specific technology or platforms.
* Some of the platforms are like AWS, AZURE, KUBERNETES.

**Impact and outcome**

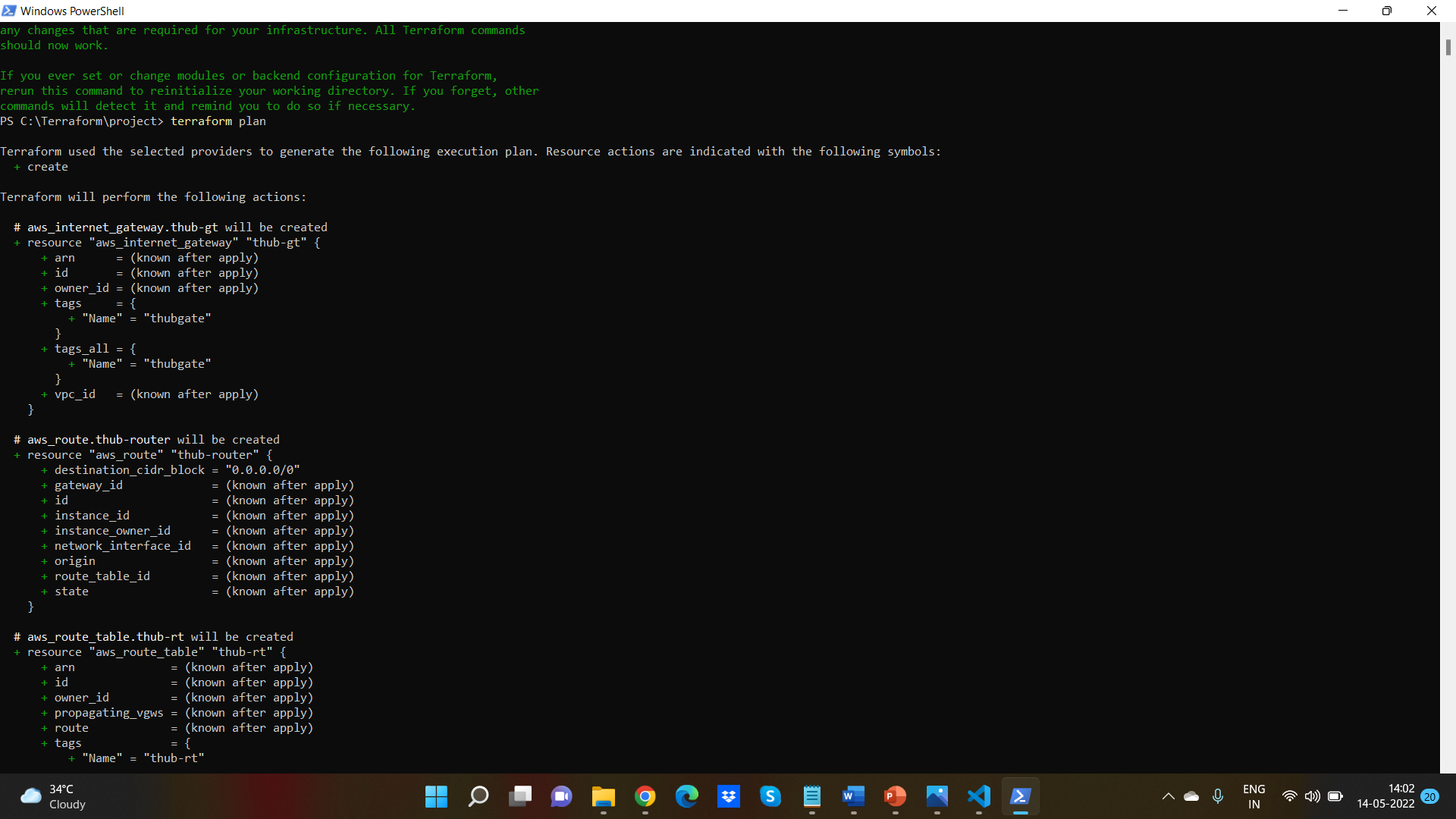
* init

The terraform init command is used to initialize a working directory containing Terraform configuration ++++++files.



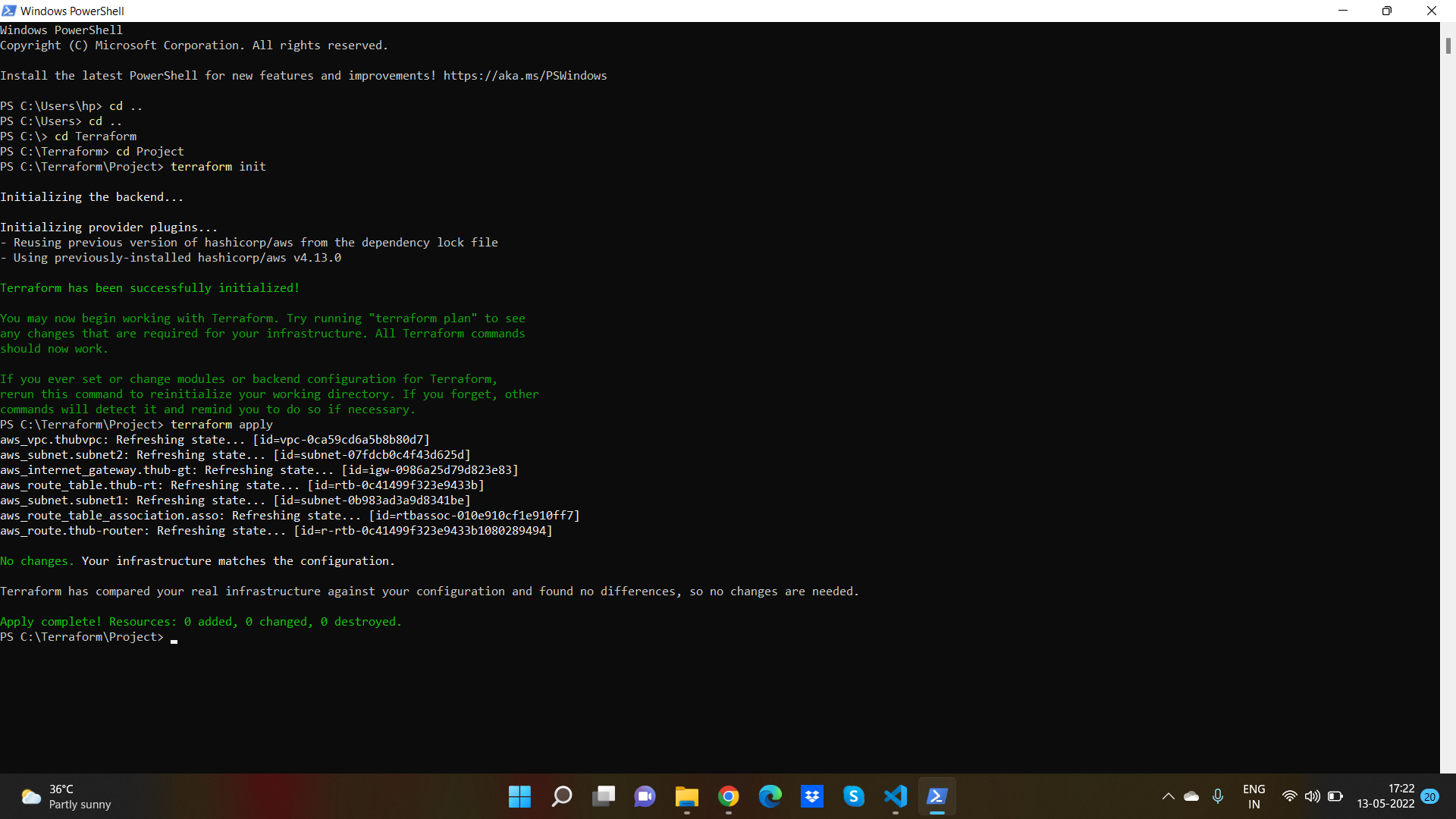
* plan

creates an execution plan, which lets you preview the changes that Terraform plans to make to your infrastructure.

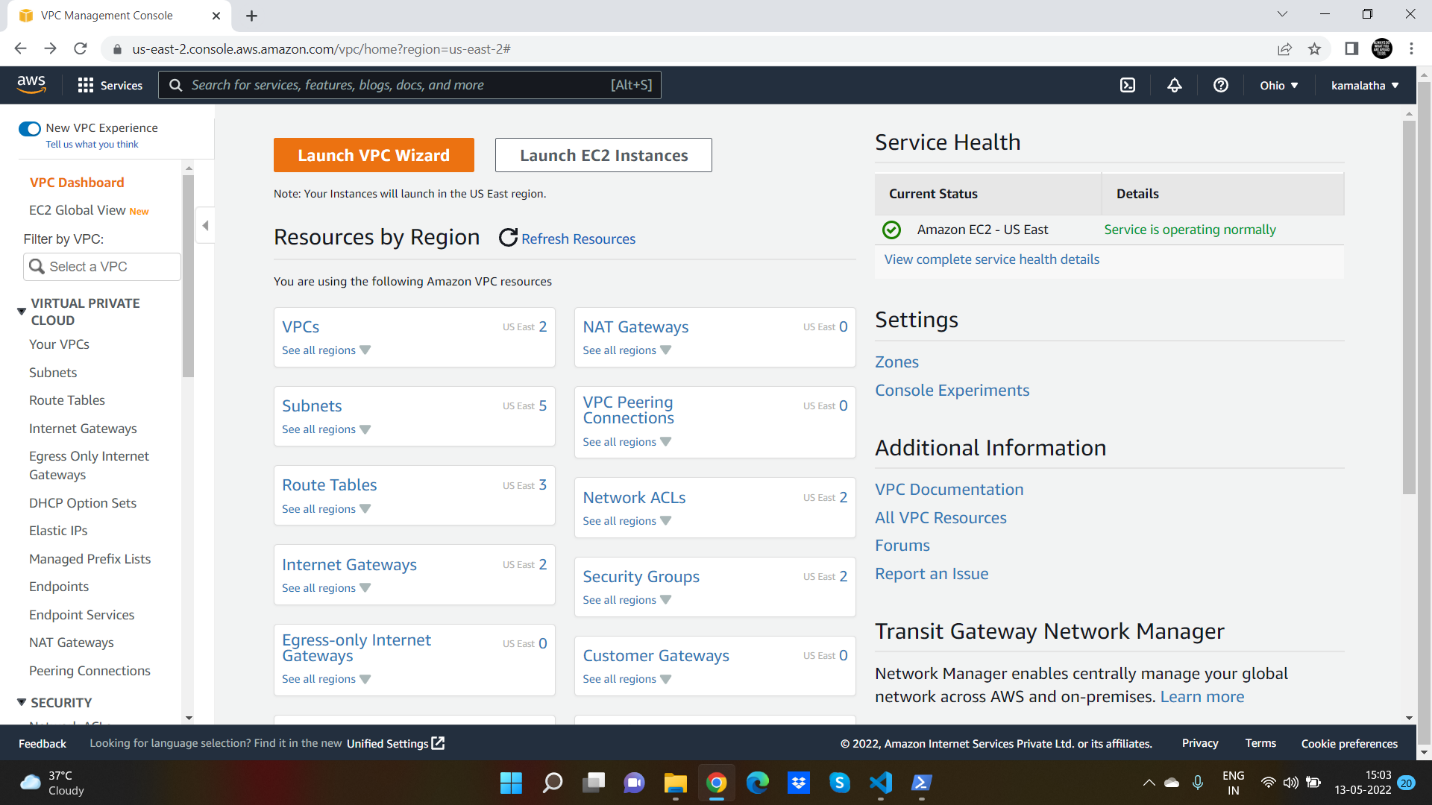
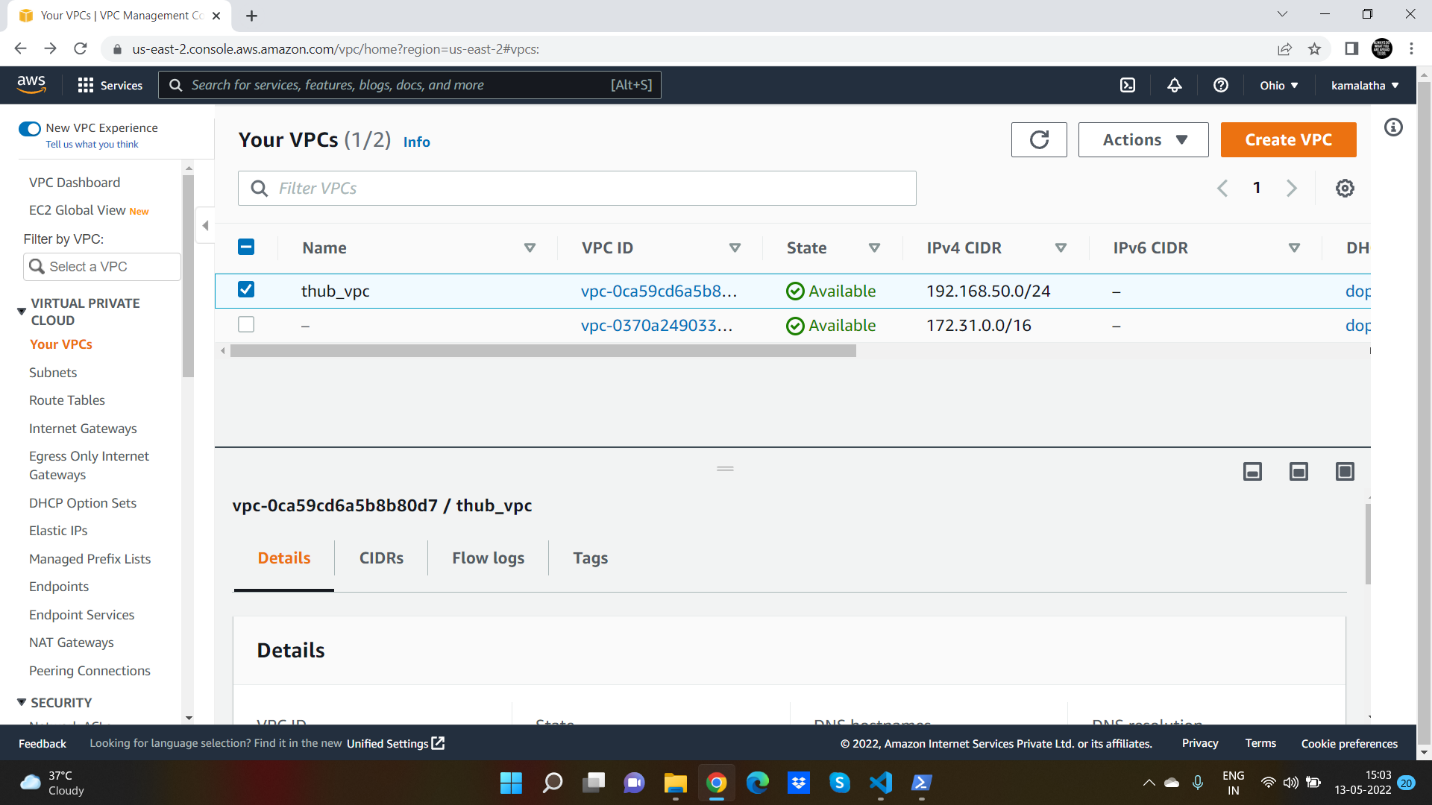


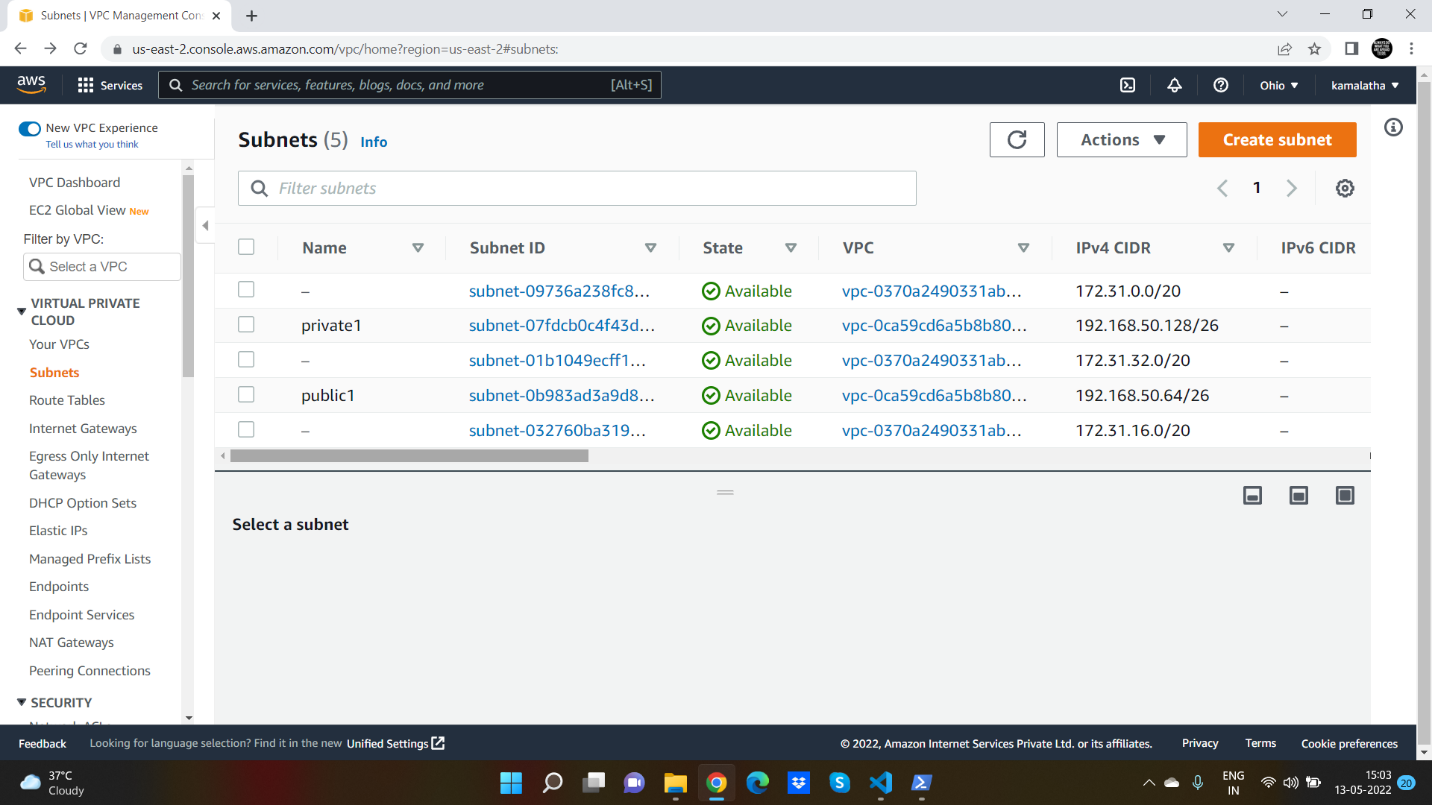
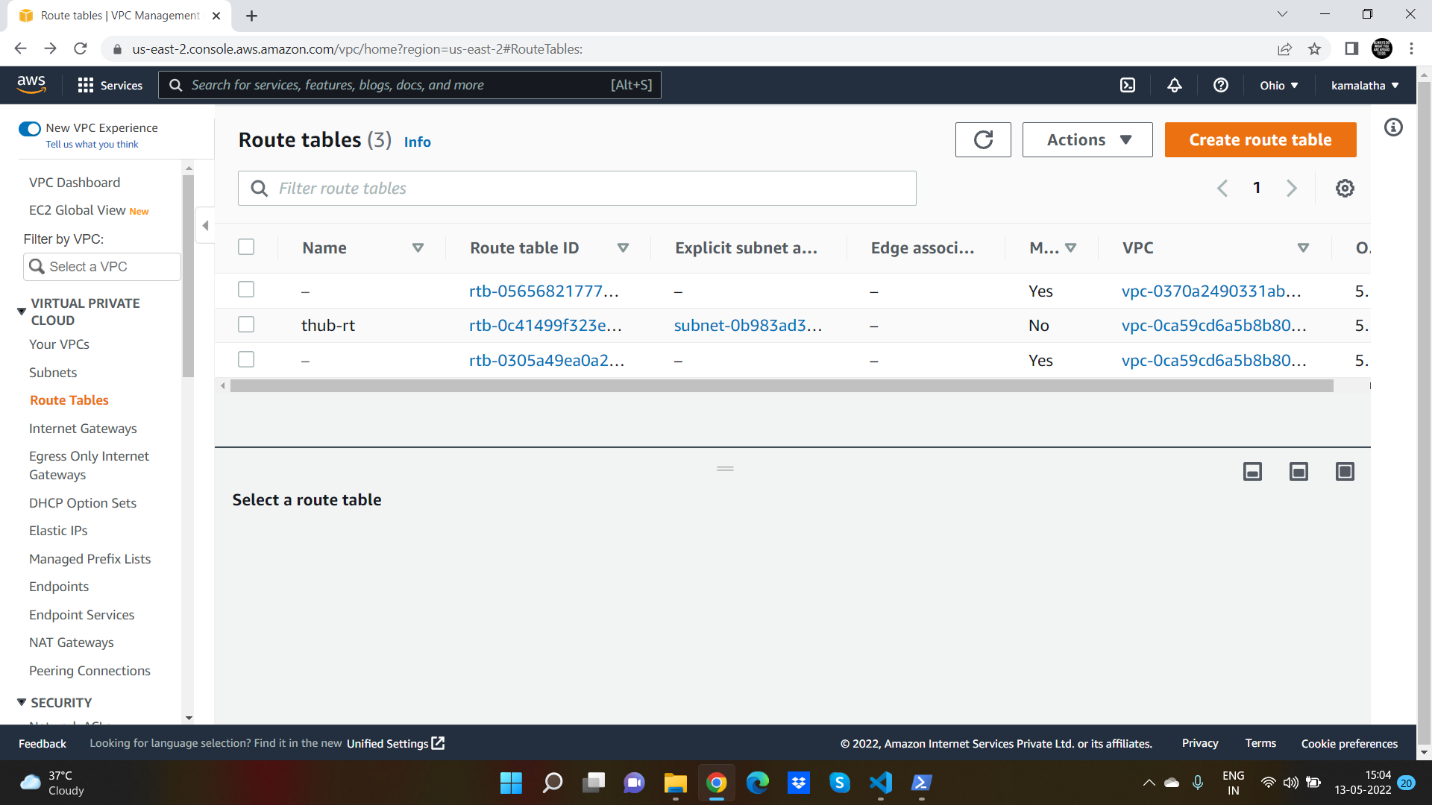
* Apply

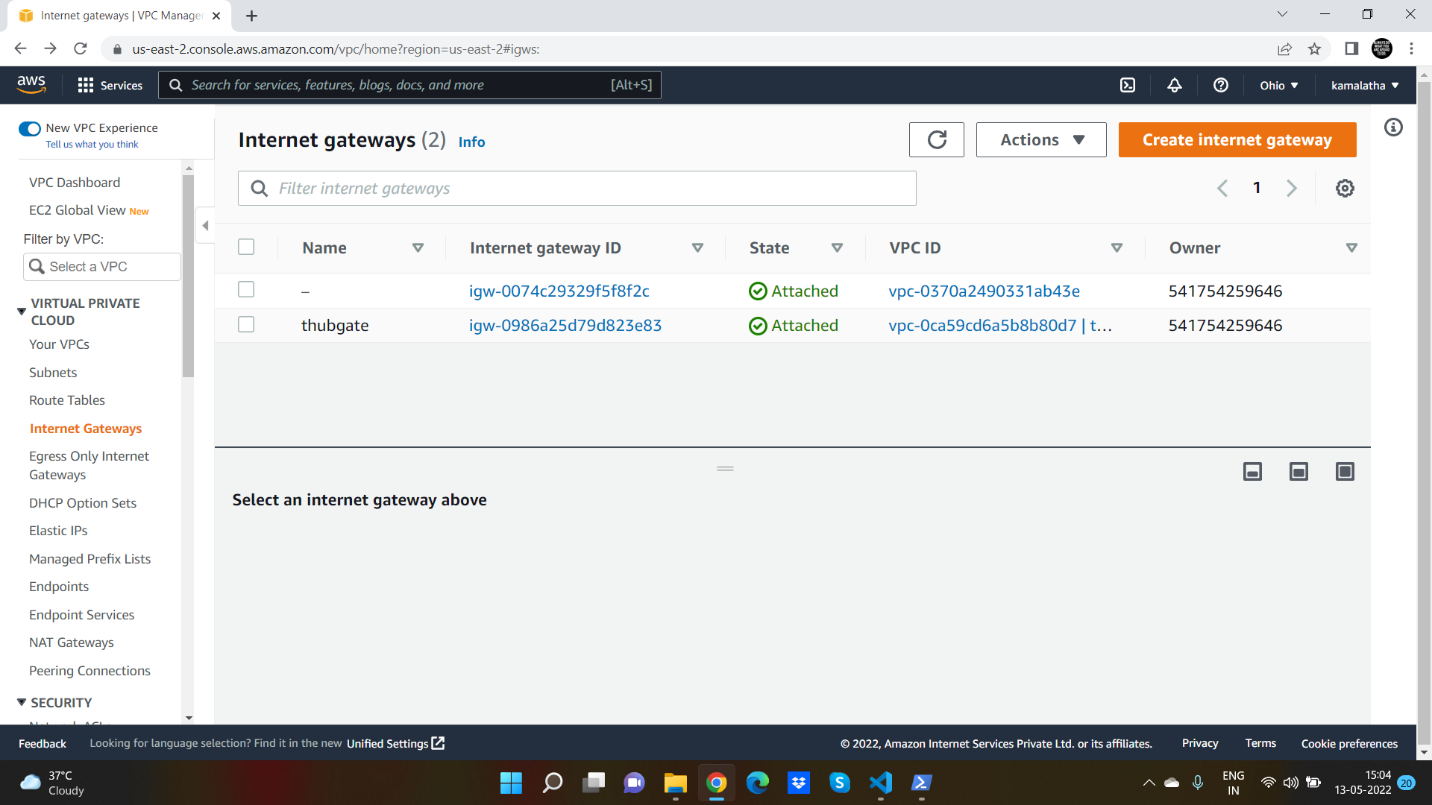
It executes the actions proposed in a terraform plan.

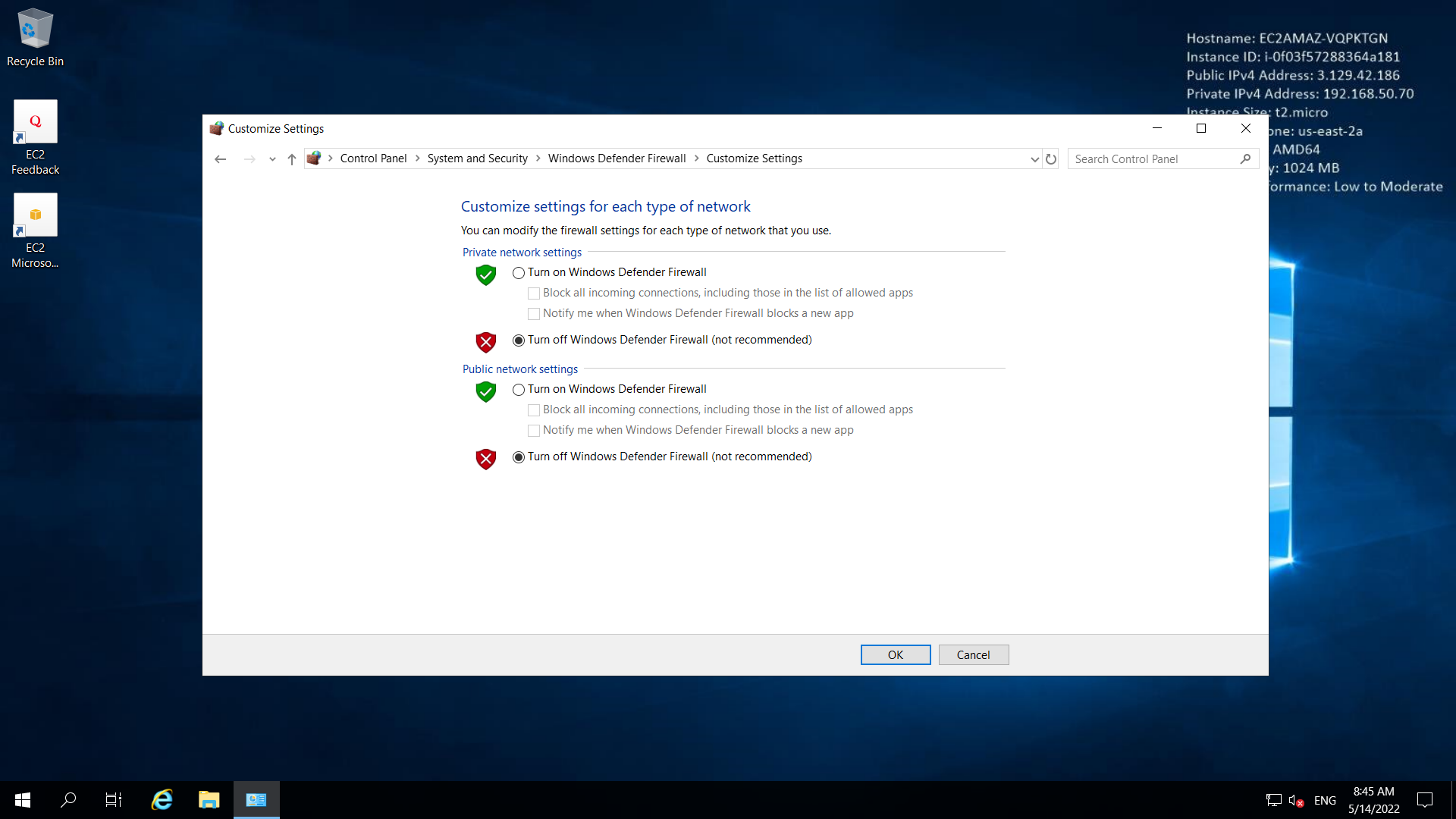


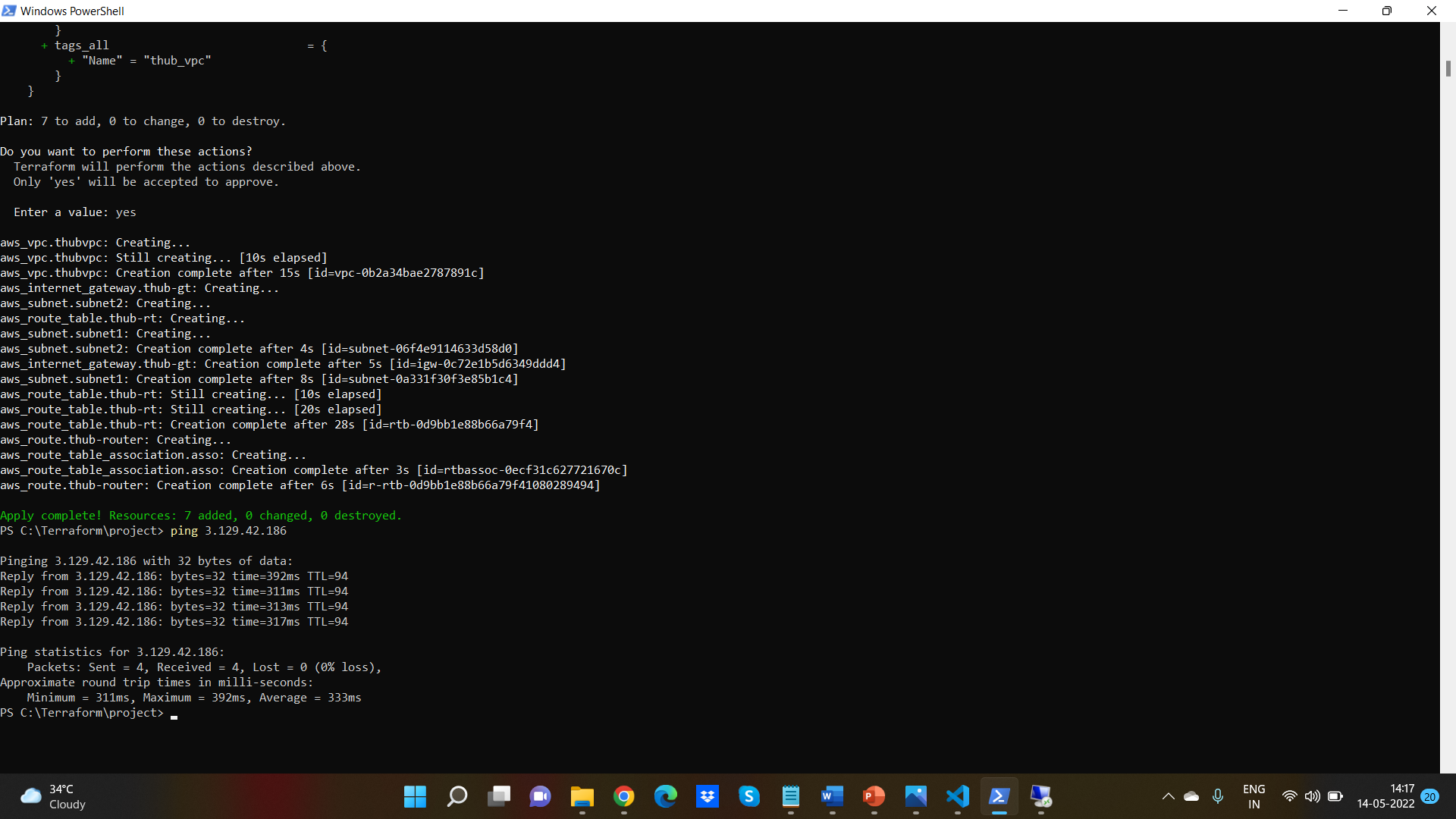
Automation output:

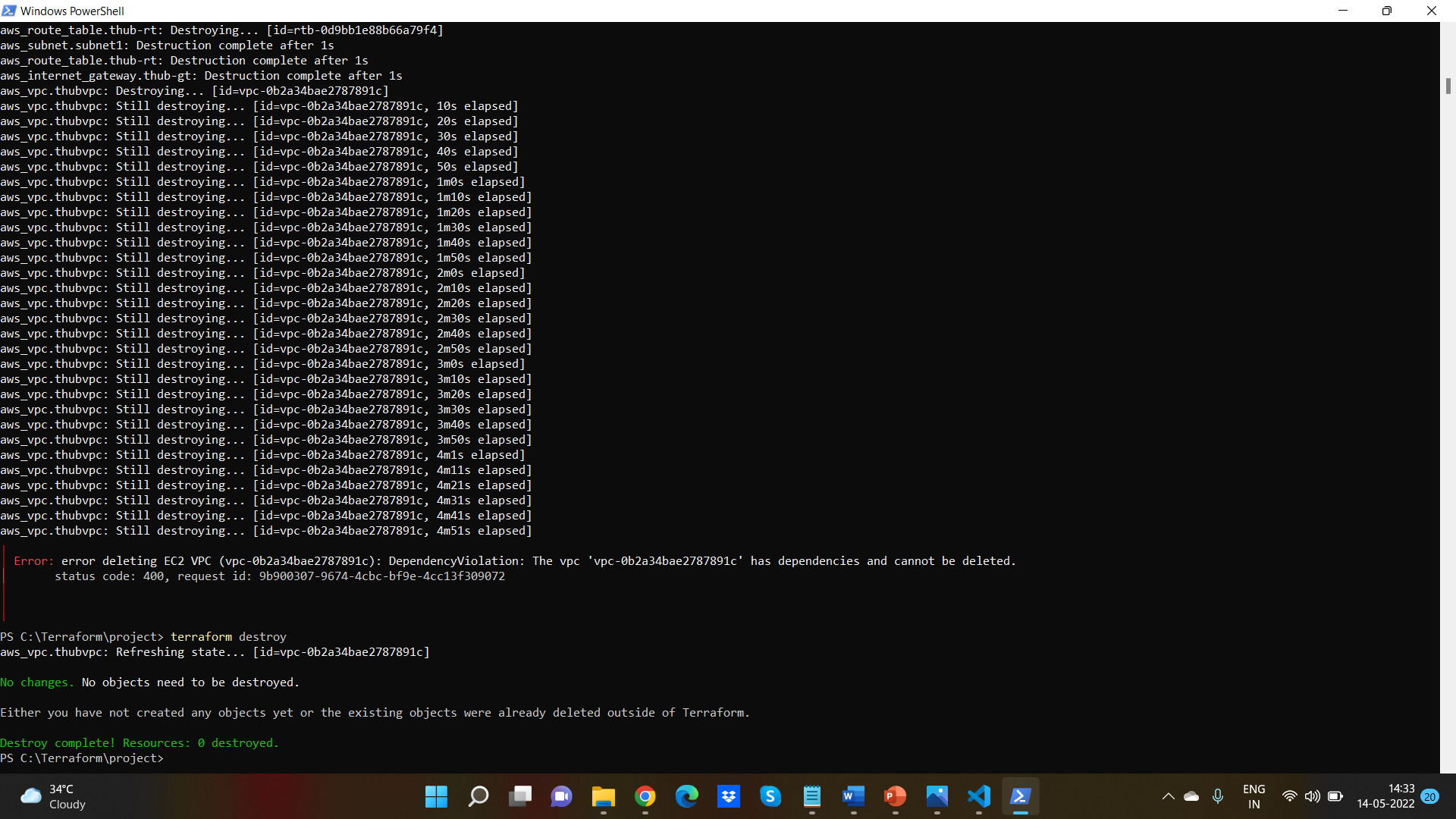






* Destroy

terminates resources defined in your Terraform configuration. This command is the reverse of terraform apply in that it terminates all the resources specified by the configuration. It does not destroy resources running elsewhere that are not described in the current configuration.



ADVANTAGES:

1. Terraform can manage infrastructure on multiple cloud platforms.  
2. The human-readable configuration language helps you write infrastructure code quickly.  
3. Terraforms state allows you to track resource changes throughout your deployments.  
4. No need to remember the current state.  
5. Terraform is an idempotent.  
6. provider translates config file to something  
 that AWS API understands.

Conclusion

We successfully automated infrastructure by using two ways defined above i.e., cloud formation and terraform .we created VPC, two subnets one is private and other one is public. Created internet gateway and route tables to connect to the internet. And finally launched an ec2- instance and got connected to internet through ping command.