

****

***DON’T JUST THINK***

***MAKE IT HAPPEN!***

**BABJI NEELAM**

**FOUNDER & CEO**

**ABOUT THE PROGRAM**

**Project space** is a feature rich environment designed to encourage individuals to develop and showcase real-time problems and further find effective solutions that can help overcome them. Every individual who chooses to utilize this space pockets themselves a chance that could possibly get them one step closer to their dream job.

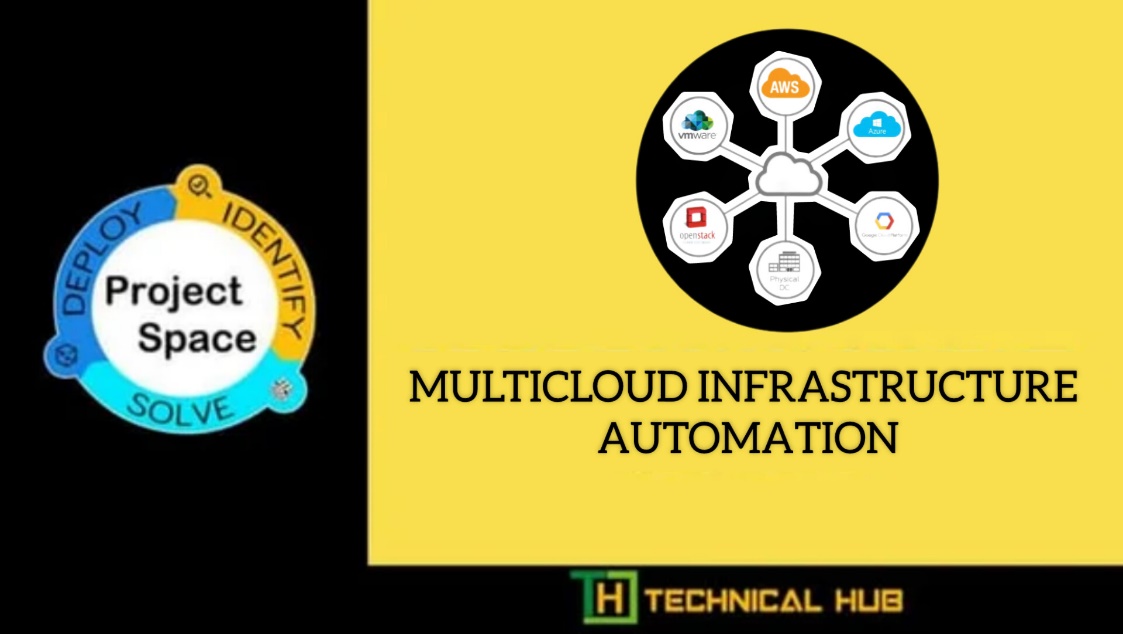


Project space is a great platform which allows you to apply the theoretical concepts you learn in your coursework to real-world situations. This helps you to better understand and internalize the material, as you can see how it is applied in practice. This practical application of knowledge helps you develop a deeper understanding of the subject matter and enhances your critical thinking and problem-solving skills.

It often involves collaboration with fellow students, faculty, or industry professionals. This allows you to build a network of contacts and develop relationships with people who may be valuable resources in your future career. Collaborating with others also helps you develop teamwork and communication skills, which are essential in many professional settings.

**PROJECT TITLE**

**MULTICLOUD INFRASTRUCTURE AUTOMATION**



Under the esteemed supervision of

Mr. VIJAY

Mr. Bobby

Mr. Surya Ashok

TEAM NAME: **STRATEGIC SQUAD**

SARITHA : 20P31A1218

TEJASWINI : 20P31A1233

SAI CHAITANYA : 20P31A1239

SUBHA SARANYA : 21P35A1201

# PROJECT OVERVIEW

# Multicloud infrastructure automation is the practice of automating the management and deployment of resources across multiple cloud environments, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

# The goal of multicloud infrastructure automation is to streamline the process of managing resources across multiple cloud providers by using automation tools and processes. This can help organizations to achieve greater agility, efficiency, and scalability, while reducing the risk of errors caused by manual processes.

# Multicloud infrastructure automation typically involves the use of Infrastructure as Code (IaC) tools, such as Terraform, CloudFormation, or Azure Resource Manager, to automate the provisioning of infrastructure resources. These tools allow organizations to define infrastructure resources in code, which can be version-controlled, tested, and deployed using automated processes.

# In addition to IaC, multicloud infrastructure automation can also involve the use of configuration management tools, such as Ansible or Chef, to automate the configuration and management of resources across multiple cloud platforms.

# Monitoring and alerting are also important aspects of multicloud infrastructure automation. Organizations can use cloud-native monitoring tools, such as AWS CloudWatch or GCP Stackdriver, or third-party monitoring tools, such as Datadog or New Relic, to monitor their multicloud infrastructure and receive alerts when issues arise.

# Overall, multicloud infrastructure automation is a key practice for organizations that want to manage resources across multiple cloud platforms efficiently and effectively. By using automation tools and processes, organizations can reduce the complexity of managing multiple cloud environments and improve their ability to meet business needs.

# 

**SCOPE OF THE PROJECT**

The scope of multicloud infrastructure automation refers to the specific boundaries and objectives of automating the management and deployment of infrastructure across multiple cloud environments. This includes public clouds, private clouds, and hybrid clouds.

The primary goal of multicloud infrastructure automation is to simplify the management and deployment of infrastructure resources across different cloud environments. This involves automating the provisioning, configuration, and scaling of infrastructure resources, such as virtual machines, storage, and networking components, to support the needs of applications and workloads.

The scope of multicloud infrastructure automation typically includes the following:

* **Cloud resource provisioning**: Automating the process of creating, configuring, and managing cloud resources across multiple clouds.
* **Configuration management**: Automating the configuration of infrastructure resources to ensure consistency and compliance with organizational policies and standards.
* **Application deployment**: Automating the deployment of applications to multiple cloud environments, while ensuring consistency and compatibility across different clouds.
* **Scaling and optimization**: Automating the scaling of infrastructure resources to support changing demands and optimizing resource usage to reduce costs and improve performance.
* **Monitoring and reporting**: Automating the monitoring of infrastructure resources and generating reports to help identify issues and optimize resource usage.

The scope of multicloud infrastructure automation can vary depending on the specific needs and objectives of the organization. However, in general, the goal is to automate as much of the infrastructure management and deployment process as possible, while maintaining control and visibility across multiple cloud environments.

**PURPOSE OF THE PROJECT**

The purpose of multicloud infrastructure automation is to simplify and streamline the management and deployment of infrastructure resources across multiple cloud environments. With multicloud infrastructure automation, organizations can more easily manage infrastructure resources in a consistent and repeatable way, regardless of the underlying cloud platform.

The primary benefits of multicloud infrastructure automation include:

* **Increased agility**: With automation, infrastructure resources can be provisioned and scaled more quickly and easily, allowing organizations to respond more rapidly to changing business requirements.
* **Improved consistency and reliability**: Automation helps to ensure that infrastructure resources are configured and deployed consistently, reducing the risk of errors and improving overall system reliability.
* **Reduced operational overhead**: By automating infrastructure management tasks, organizations can reduce the amount of manual effort required to manage their cloud environments, freeing up IT resources for other tasks.
* **Better resource utilization**: Automation can help organizations optimize resource usage and reduce costs by automatically scaling 11infrastructure resources up or down based on demand.
* **Improved security and compliance**: Automation can help ensure that infrastructure resources are configured and managed in compliance with organizational policies and security standards.

Multicloud infrastructure automation can encompass a wide range of tasks and workflows, from resource provisioning and configuration management to application deployment and scaling. By automating these tasks, organizations can improve efficiency, reduce costs, and improve the overall performance and reliability of their cloud environments.

**TOOLS USED**

**Terraform:**

Terraform is an open-source infrastructure as code (IAC) tool that allows users to define, provision, and manage infrastructure resources across multiple cloud providers and on-premises data centers. It was created by HashiCorp, the same company behind other popular DevOps tools such as Vagrant, Consul, and Vault.

With Terraform, users can write declarative configuration files in a human-readable syntax called HashiCorp Configuration Language (HCL) or JSON, that describe the desired state of their infrastructure resources. Terraform then applies these configuration files to create, modify, or delete resources in a consistent and automated manner.

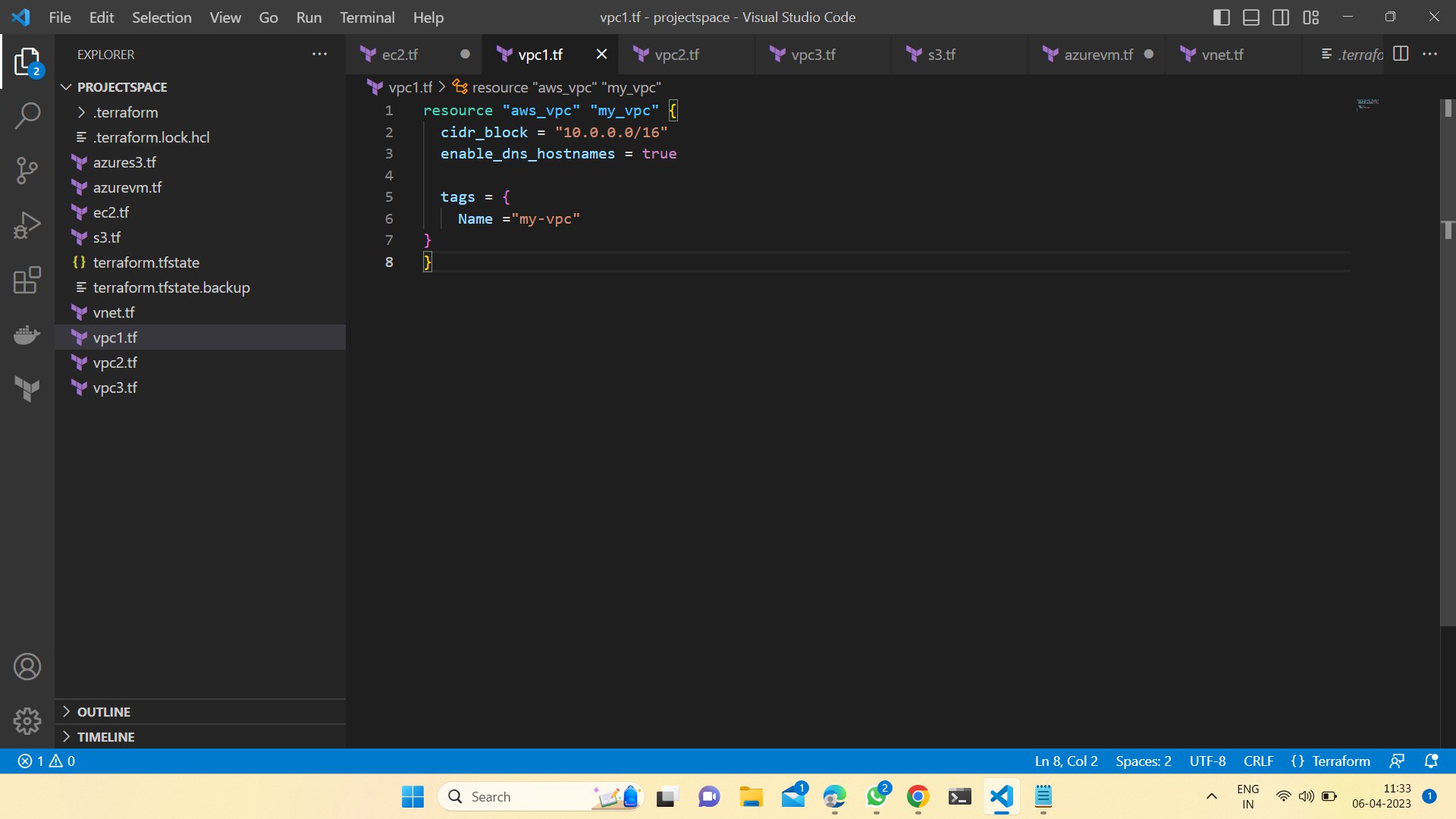
Terraform supports a wide range of infrastructure resources, including virtual machines, storage volumes, network interfaces, load balancers, and more, and integrates with major cloud providers such as AWS, Azure, Google Cloud Platform, and Alibaba Cloud.

One of the key benefits of Terraform is its ability to manage infrastructure resources across multiple cloud providers and data centers, enabling users to adopt a multi-cloud strategy that avoids vendor lock-in and provides greater flexibility and resilience.

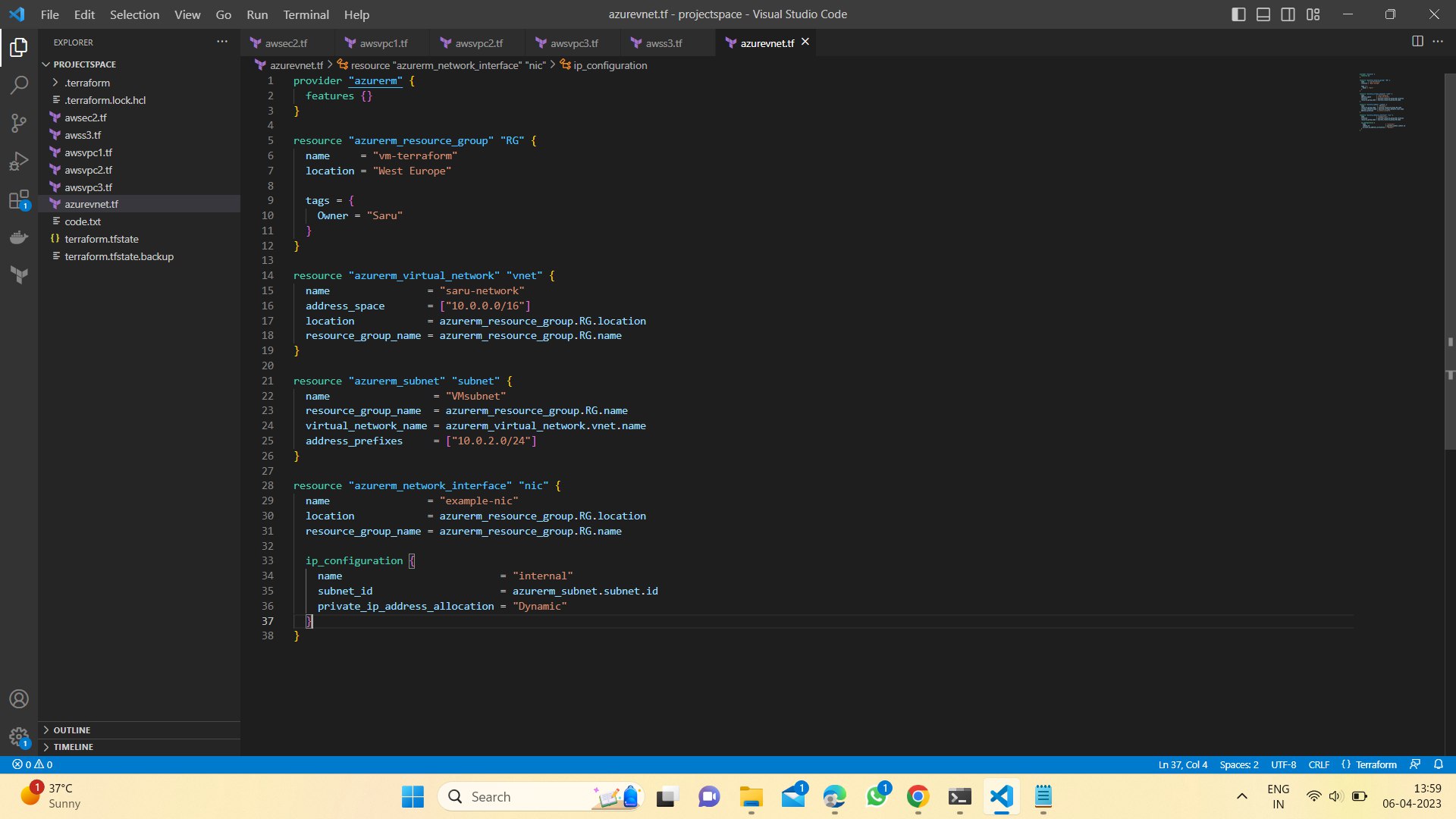
Terraform also includes features such as state management, plan preview, and version control integration, which help users manage their infrastructure resources more effectively and minimize the risk of configuration errors.

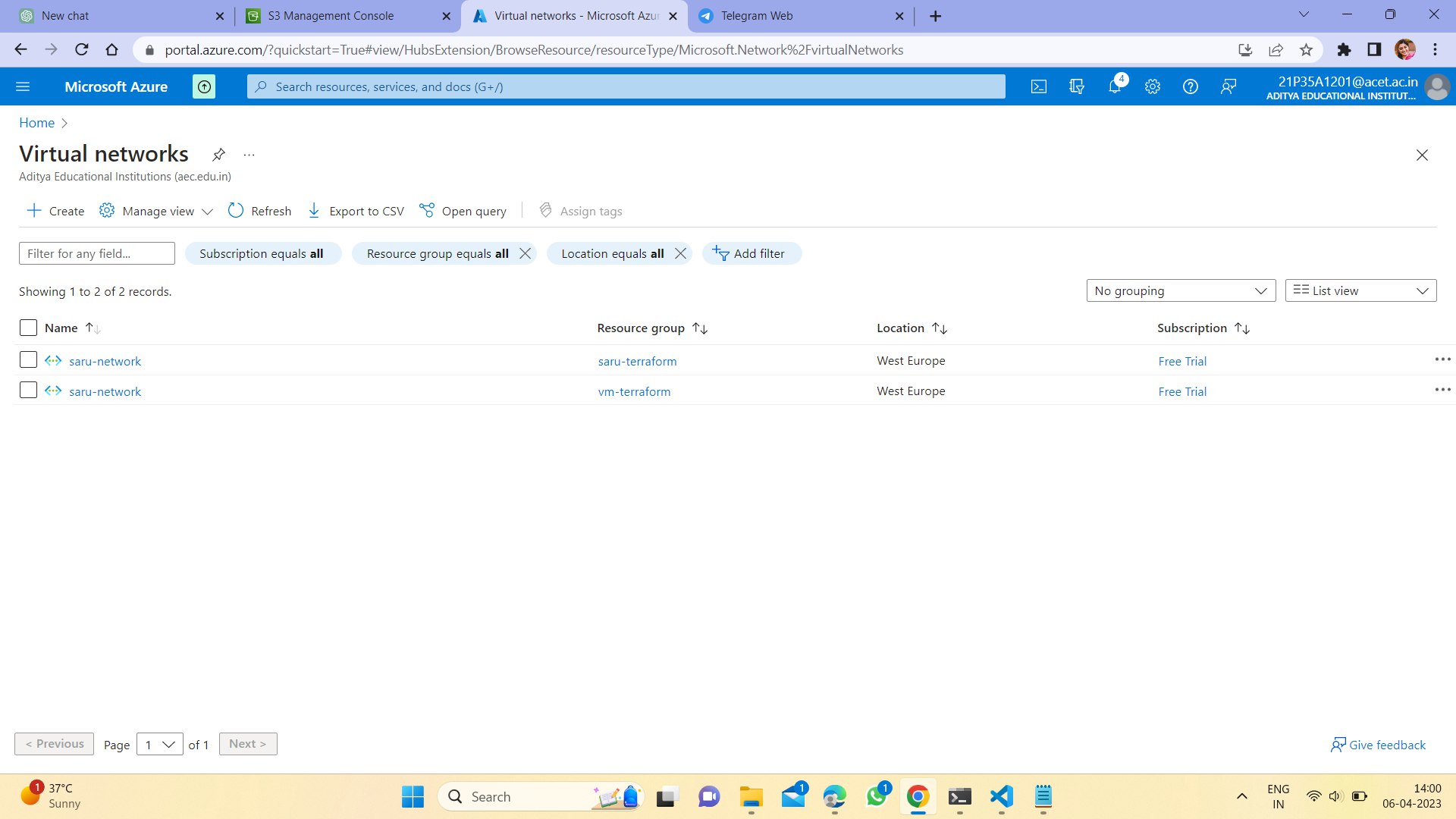
Azure Resource Manager: Azure Resource Manager is a Microsoft Azure tool that can be used to provision and manage resources across multiple Azure subscriptions and regions.

**AWS VPC using Terraform:**



**AZURE VNET using Terraform:**

****

****

**TECHNOLOGIES USED**

**Amazon Web Services:**

Amazon Web Services (AWS) is a cloud computing platform offered by Amazon, providing a range of infrastructure and services for building and deploying applications and services in the cloud. AWS offers a comprehensive set of tools and services for businesses and individuals to build, scale, and manage their IT infrastructure, including:

Compute: AWS provides computing resources, such as EC2 instances, containers, and serverless computing options, including AWS Lambda.

Storage: AWS offers various storage options, including object storage with Amazon S3, block storage with Amazon EBS, and file storage with Amazon EFS.

Database: AWS provides various database options, including relational databases such as Amazon RDS, NoSQL databases such as Amazon DynamoDB, and in-memory databases such as Amazon ElastiCache.

Networking: AWS offers a range of networking services, including virtual private clouds (VPCs), load balancing, and content delivery with Amazon CloudFront.

Security: AWS provides a wide range of security services, including identity and access management (IAM), encryption, and threat detection.

Analytics: AWS offers a range of analytics and machine learning services, including Amazon SageMaker, Amazon Redshift, and Amazon Athena.

Management: AWS provides a range of management and monitoring tools, including AWS CloudFormation, AWS Config, and AWS CloudWatch.

Overall, AWS provides a flexible and scalable platform for businesses and individuals to build and manage their IT infrastructure, with a range of tools and services that can be customized to meet specific needs and requirements

**SERVICES USED:**

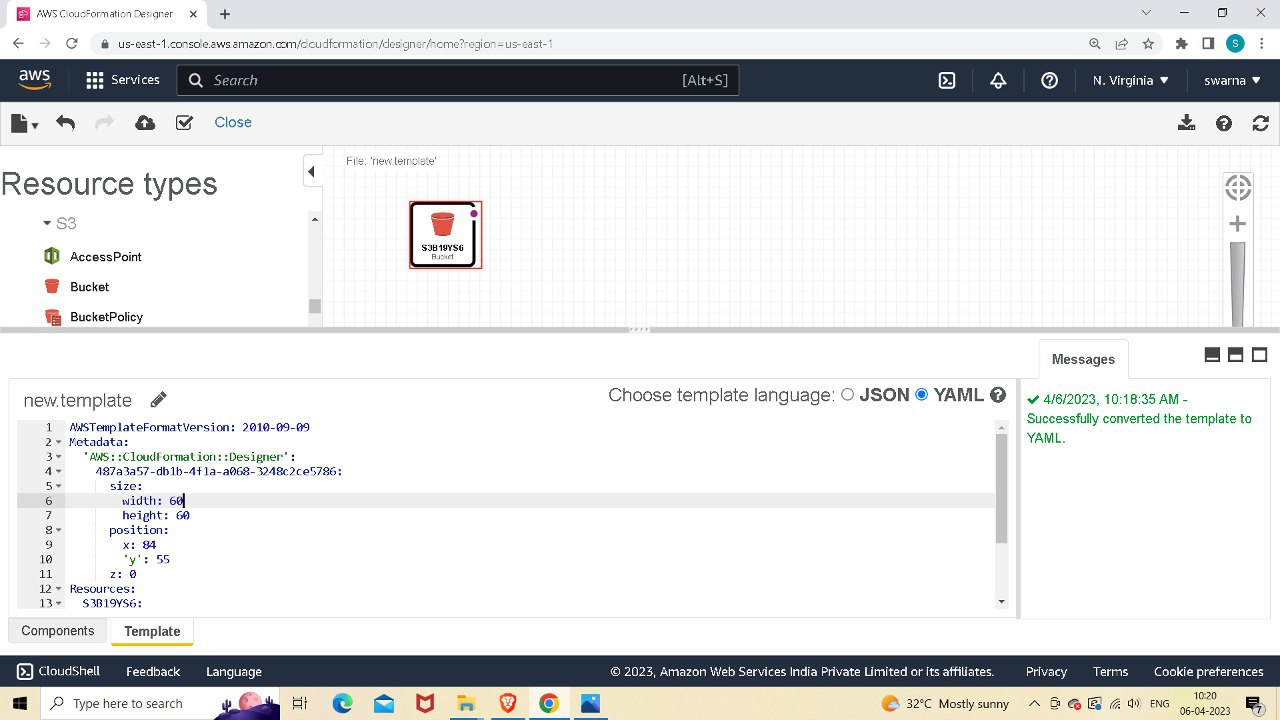
**CloudFormation:**

AWS CloudFormation is a service that allows users to define and provision their AWS infrastructure and resources in a declarative way, using templates. It enables users to create a collection of AWS resources and manage them as a single unit, known as a stack.

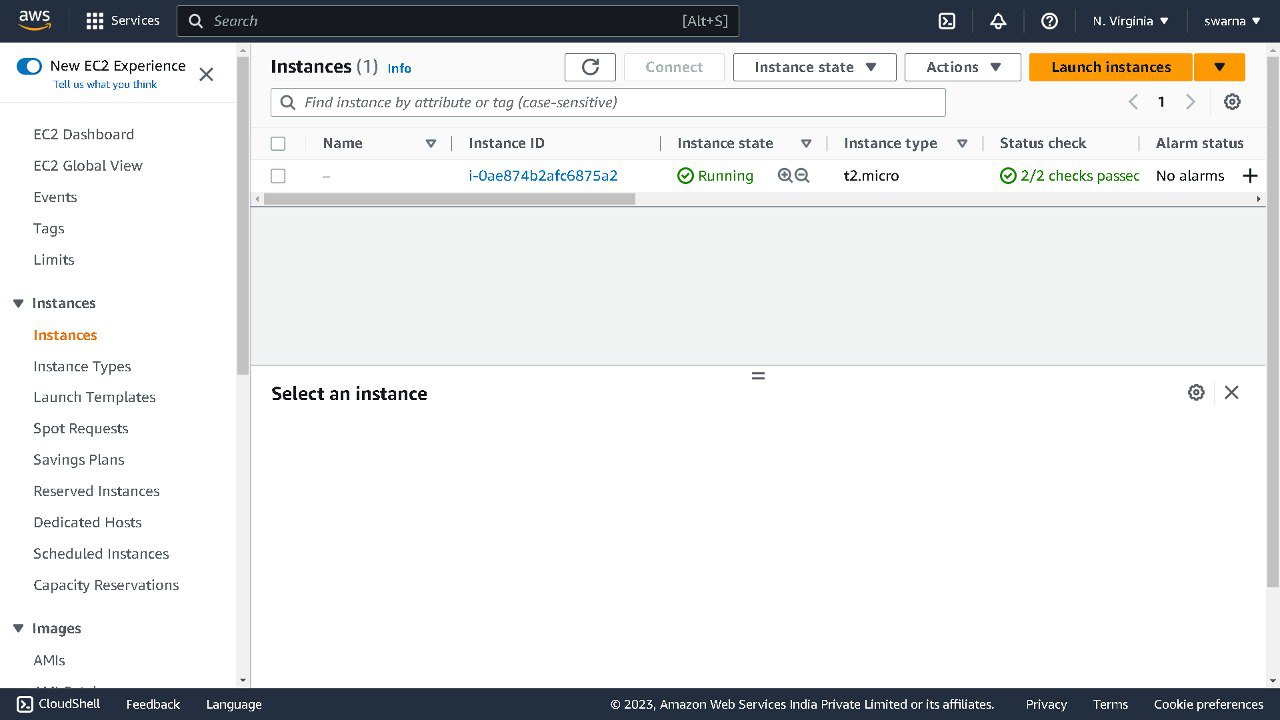
CloudFormation templates are written in JSON or YAML format and can include various AWS resources, such as Amazon EC2 instances, Amazon S3 buckets, Amazon RDS databases, and more. Users can define the properties and configurations of these resources in the template, as well as their dependencies and relationships with other resources.

When a CloudFormation stack is created, updated, or deleted, CloudFormation manages the entire lifecycle of the stack and its resources, handling dependencies, rollback, and error handling automatically. This makes it easy to create and manage complex infrastructure and applications in AWS, while maintaining consistency and reproducibility across different environments.

CloudFormation is often used in conjunction with other AWS services, such as AWS CodePipeline and AWS CodeBuild, to enable continuous integration and delivery (CI/CD) workflows for infrastructure and application deployments.



**Output:**

****

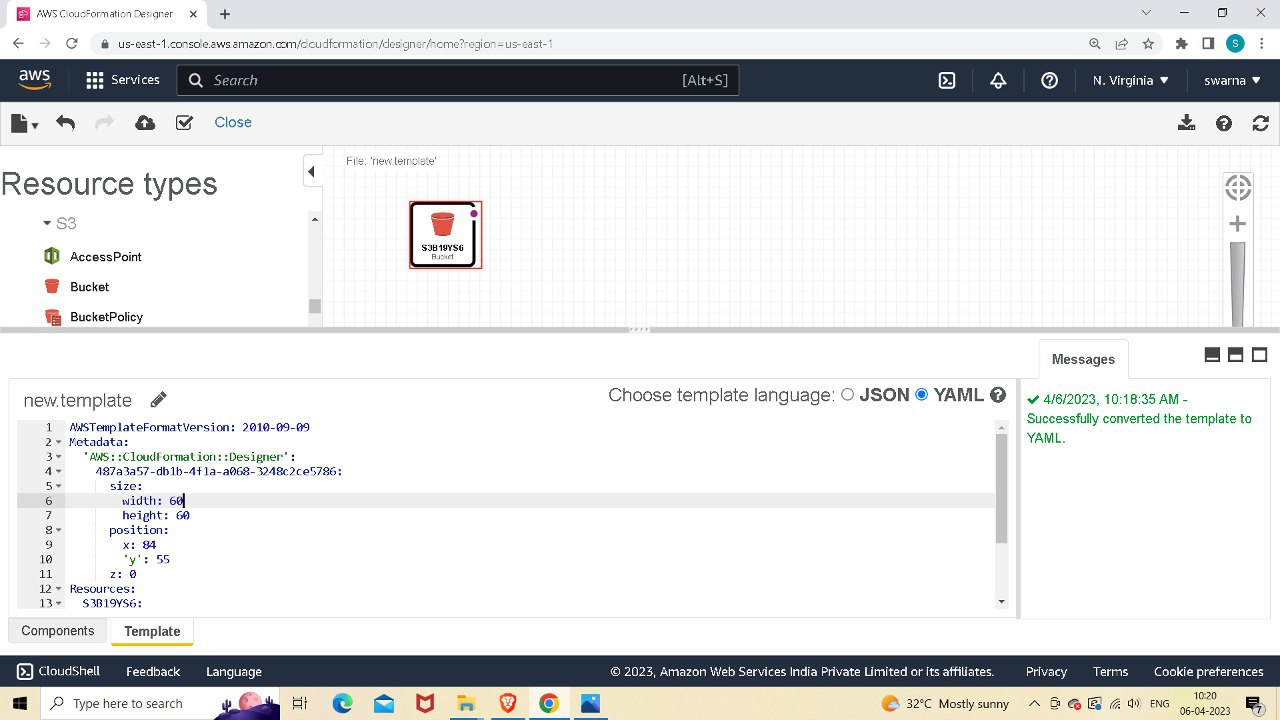
**AZURE:**

Microsoft Azure, always referred to as Azure. is a cloud computing platform operated by Microsoft that provides access, management, and development of applications and services via globally-distributed data centers. Microsoft Azure has multiple capabilities such as software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS) and supports many different programming languages, tools, and frameworks, including both Microsoft-specific and third-party software and systems.

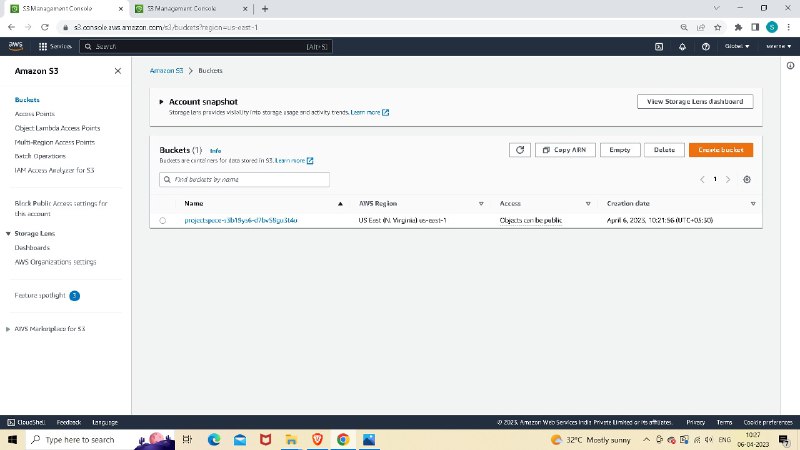
Azure, announced at Microsoft's Professional Developers Conference (PDC) in October 2008, went by the internal project codename "Project Red Dog", and was formally released in February 2010 as Windows Azure, before being renamed Microsoft Azure on March 25, 2014.

**PROOF OF CONCEPT**

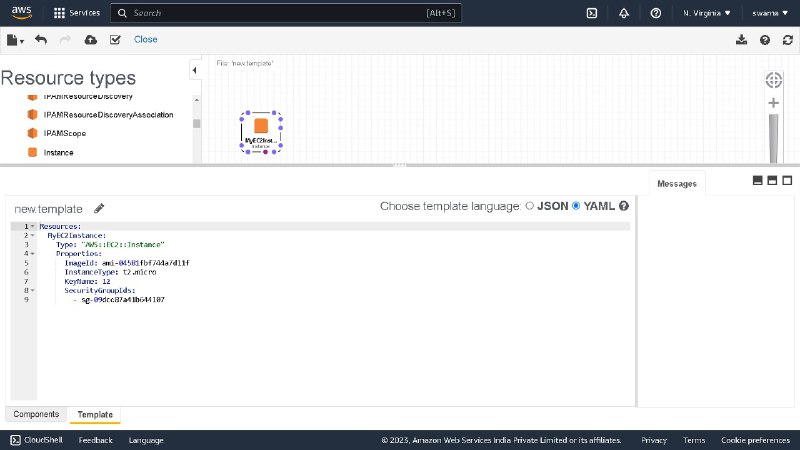
**Creating S3 bucket using CloudFormation:**

****

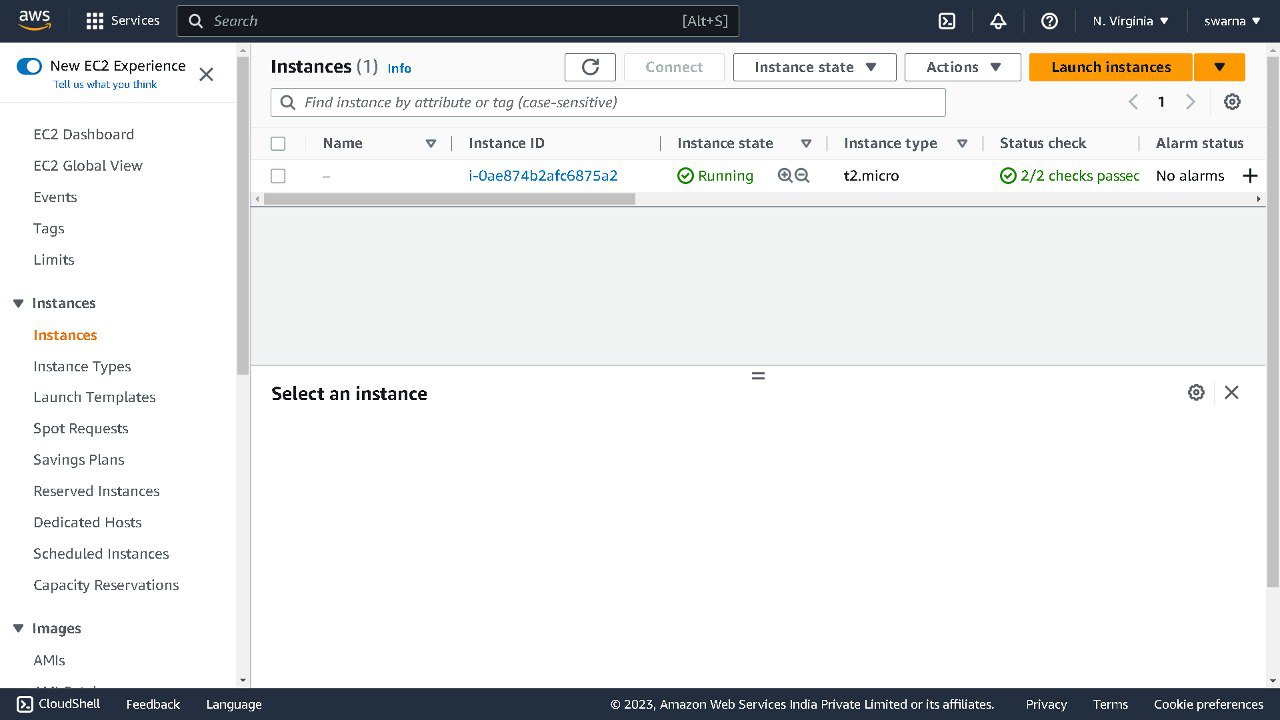
**S3 Bucket created using CloudFormation:**

****

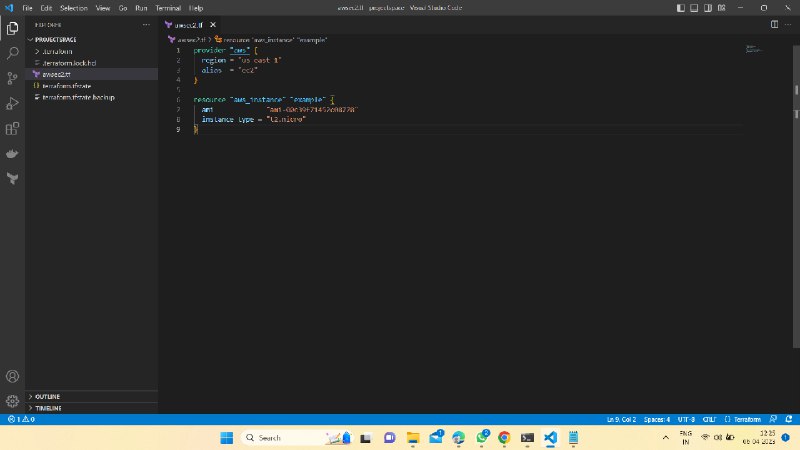
**Creating Instance using CloudFormation:**

****

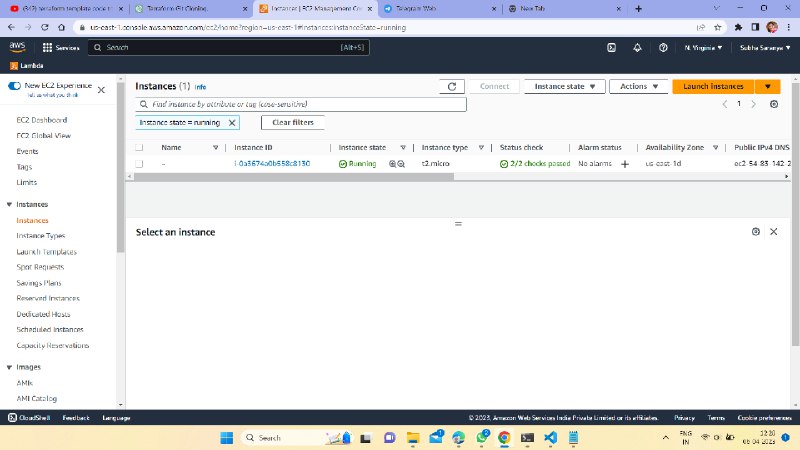
**Instance created using CloudFormation:**

****

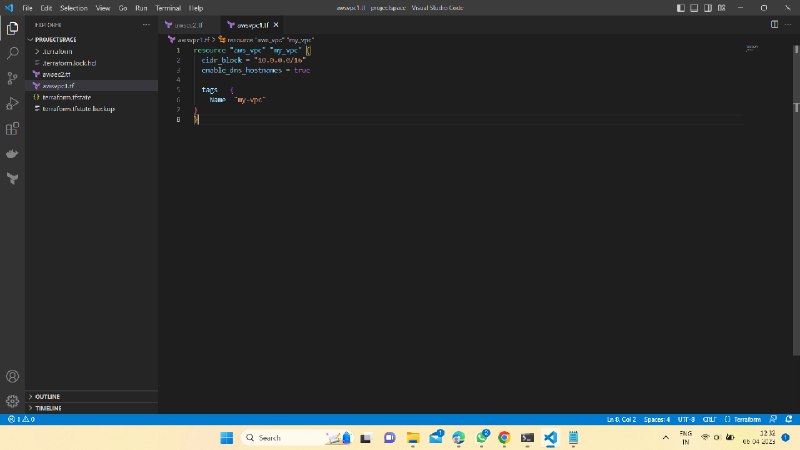
**Creating EC2 using Terraform:**

****

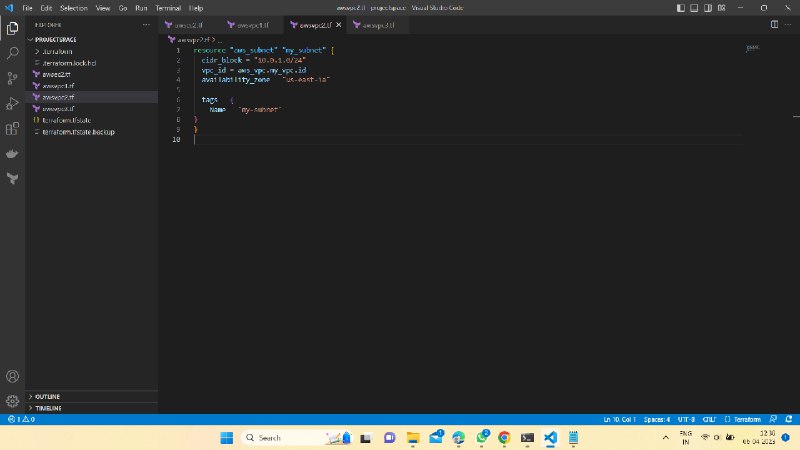
**EC2 using Terraform:**

****

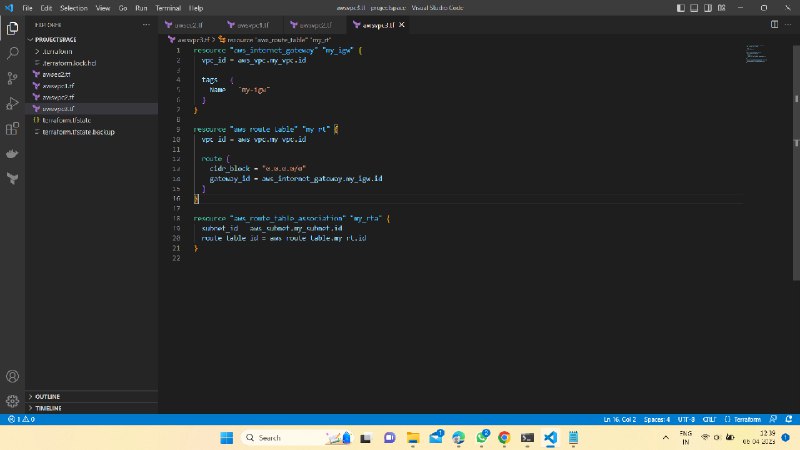
**Creating first VPC using Terraform:**

****

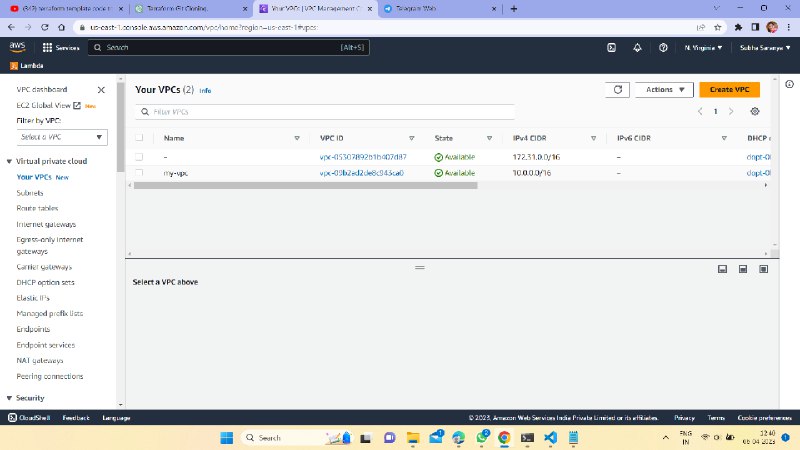
**Creating second VPC using Terraform:**

****

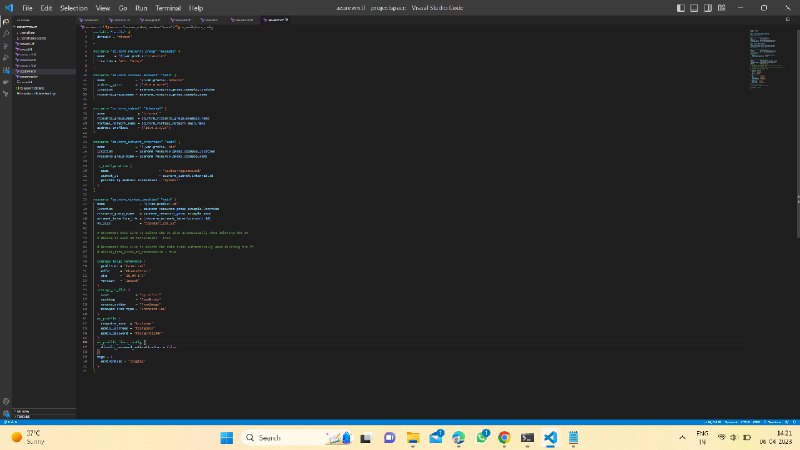
**Creating third VPC using Terraform:**

****

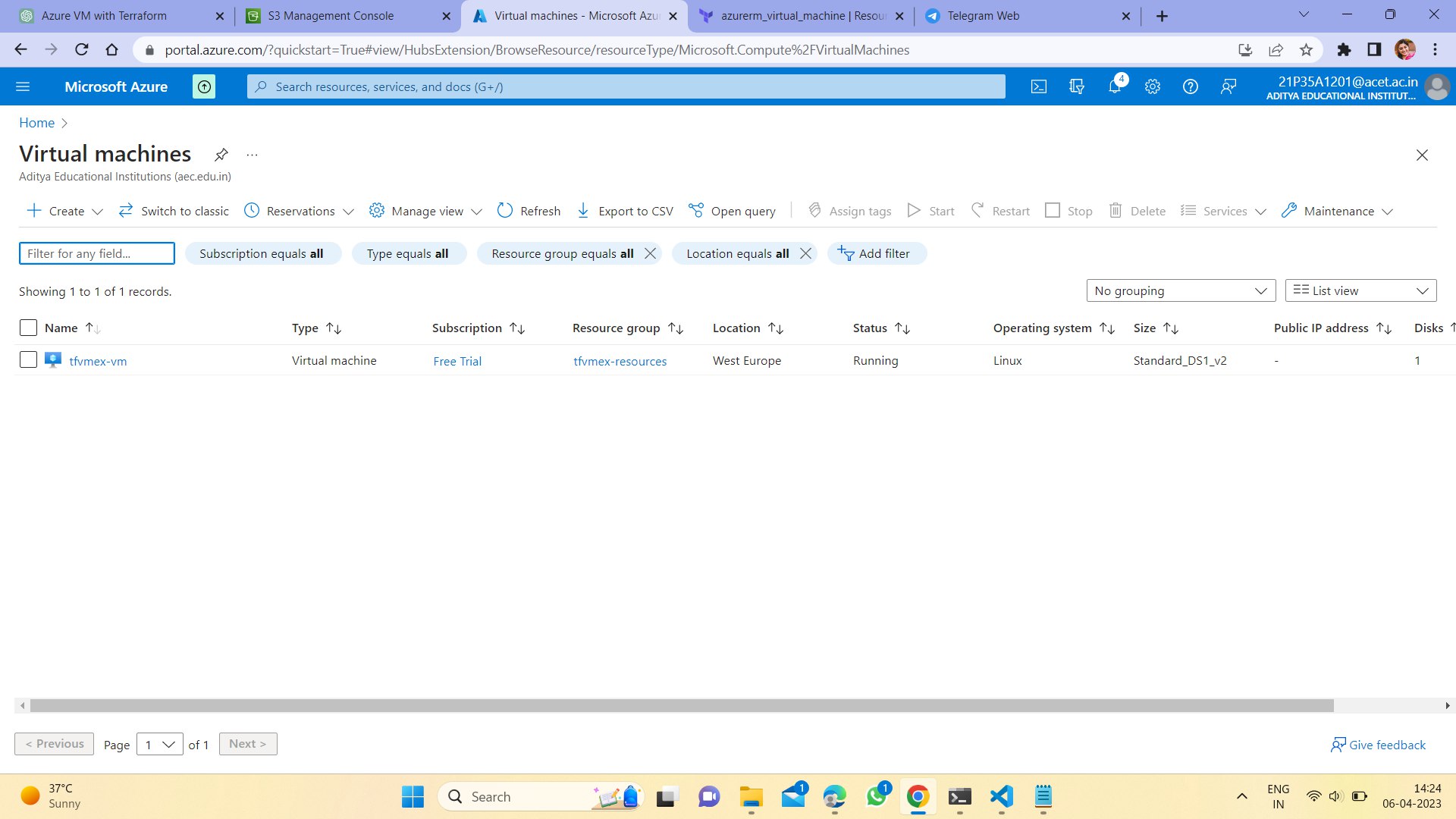
**AWS VPC using Terraform:**

****

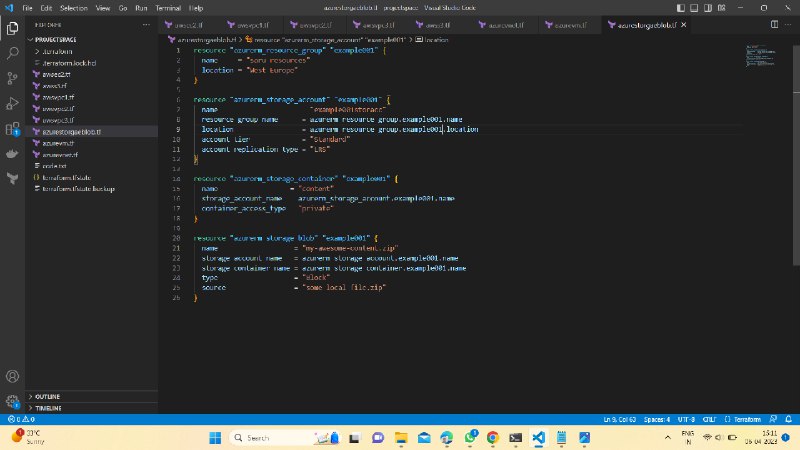
**Creating Azure Virtual Machine using Terraform:**

****

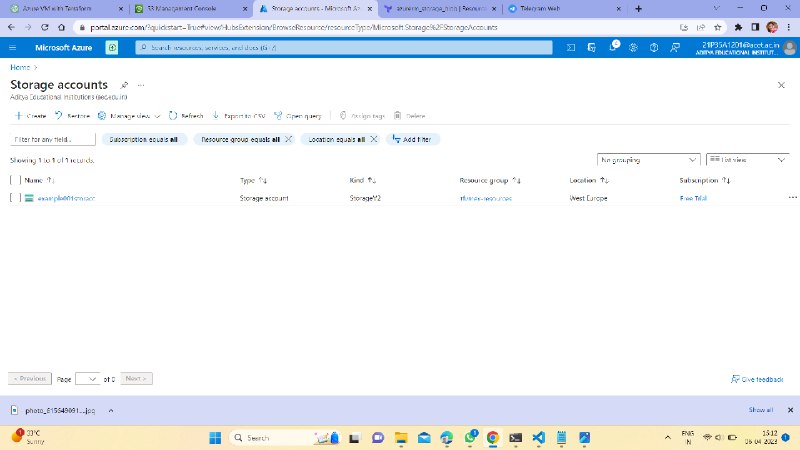
**Azure Virtual Machine Output:**

****

**Azure Storage Blob:**

****

**Azure Storage blob created using Terraform:**

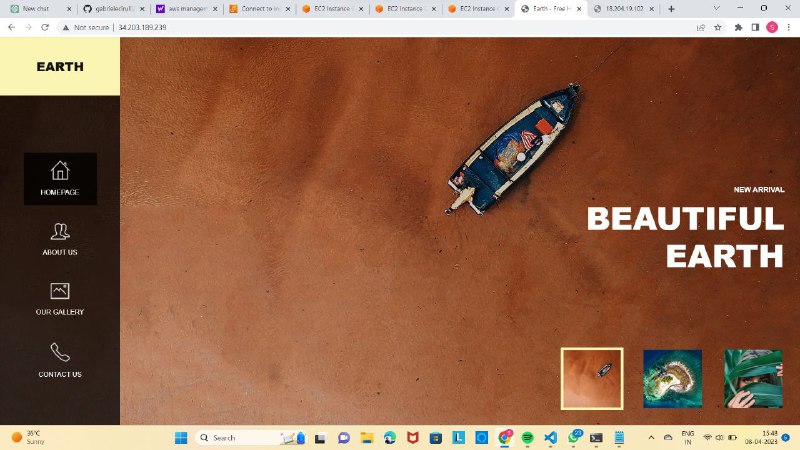
****

**REPOSITORY DETAILS:**

* [AWS\_EC2](https://github.com/TejaswiniSruthi/Project_Space_4.0/blob/main/AWS_EC2.tf)
* [AWS\_S3](https://github.com/TejaswiniSruthi/Project_Space_4.0/blob/main/AWS_S3.tf)
* [AWS\_VPC\_1](https://github.com/TejaswiniSruthi/Project_Space_4.0/blob/main/AWS_VPC.tf)
* [AWS\_VPC\_2](https://github.com/TejaswiniSruthi/Project_Space_4.0/blob/main/AWS_VPC2.tf)
* [AWS\_VPC\_3](https://github.com/TejaswiniSruthi/Project_Space_4.0/blob/main/AWS_VPC3.tf)
* [Azure\_Virtual\_Machine](https://github.com/TejaswiniSruthi/Project_Space_4.0/blob/main/Azure_vm.tf)
* [Azure\_vnet](https://github.com/TejaswiniSruthi/Project_Space_4.0/blob/main/Azure_vnet.tf)

**OUTCOME**

This is a website which is deployed automatically without using any cloud platform. This website is created using the tool Terraform.



**CONCLUSION**

Multi-Cloud Infrastructure utilizes multiple public cloud Infrastructure as a Service (IaaS) platforms. In this multi-cloud model, the differences between cloud providers are abstracted from the applications and operational procedures that work consistently across all environments.

This enables organizations to gain the benefits of each cloud provider’s strengths while mitigating the risks normally associated with relying on a single cloud provider. As organizations increasingly modernize applications and seamlessly manage and control deployment and scale across environments, multi-cloud management of underlying infrastructure becomes more important.

**HOW IT HELPS THE IT SECTOR?**

As organizations transform their application stacks by building new software or modernizing existing applications, different applications may be optimized with cloud services that are unique to each provider (Artificial Intelligence best fit on Google Cloud). There may be enterprise applications that are optimized for a specific cloud (Oracle Cloud).

There may be predominance of applications on a specific Operating System (OS) such as Microsoft Windows, that are a best fit for a particular cloud (Microsoft Azure) And there may be business reasons, such as data gravity or management tools available in limited languages, or regional team preference – that make one cloud infrastructure platform a better fit than another.

**THANK YOU**