CLOUD INFRASTRUCTURE AUTOMATION USING ANSIBLE

MS PROJECT PROPOSAL

By

Vinay Gummadavelli
Swetha Bonthula

Table of Contents

Table of Figures

1. ABSTRACT

The way we think about computer infrastructure has changed dramatically because of cloud computing. Instead ofbuying and maintaining expensive hardware, businesses can now rent resources in the cloud on an as-needed basis. However, managing these resources can be daunting, especially for large-scale deployments. This project aims to simplify cloud infrastructure management by usingPython, Ansible, and automation tools to streamline the deployment and configuration of cloud networking resources on AWS, GCP, and Azure.

Tools and Technologies: - Cloud Services, AZURE, AWS, GCP, Ansible, IaaS, Python scripting, Networking.

2. ACKNOWLEDGEMENT

We are immensely grateful to our esteemed project supervisor, Professor Nader F. Mir of the Electrical Engineering Department at San Jose State University. His guidance and support have been pivotal throughout our project. His invaluable advice and constant encouragement have been a cornerstone of our research journey. Professor Mir's extensive knowledge and passionate approach have been inspirational, and we feel fortunate to have had him as our mentor and advisor.

Our sincere gratitude is also extended to Professor Birsen Sirkeci, Ph.D., MSc, BSc, Associate Professor of Electrical Engineering at San Jose State University. Our project experience has been tremendously enhanced by her astute criticism and helpful recommendations.

We are not just grateful to the university. We sincerely thank our families, roommates, friends, and mentors for their unwavering moral support and encouragement. Their presence has been a source of strength and motivation throughout our research.

A special note of appreciation goes to our fellow students. Their commitment and hard work during those countless late nights were instrumental in the timely completion of this project report. Their dedication and teamwork have been invaluable.

Vinay Gummadavelli and Swetha Bonthula

MSEE

San Jose State University

3. INTRODUCTION

3.1 Cloud Computing and Automation:

3.1.1 Importance and Literature review of cloud computing

Cloud computing is a concept that facilitates global access to information, data, and computational resources via the internet. Internet allows its consumers to communicate from anywhere in the world, using electronic devices such as Mobiles, Computers. One of its defining prime features is the "pay-as-you-go" model, which eliminates the need for significant upfront infrastructure investments and provides cost savings.

Cloud commuting has primarily two main models based on the infrastructure: public clouds and private clouds. Public clouds are hosted by third-party providers and serve multiple clients, while private clouds are dedicated to a single organization. This duality offers flexibility for organizations to choose the model that aligns with their specific requirements.

According to the International Data Corporation (IDC), vendor revenue from cloud IT infrastructure products, including servers, enterprise storage, and Ethernet switches, experienced impressive year-over-year growth in various regions. Notably, the Middle East & Africa and Western Europe saw significant growth, reflecting the global trend toward embracing cloud technologies [2].

Cloud environments are prized for their agility and scalability, making it possible to predict user behavior and innovate applications or products swiftly. The rapid provisioning of resources empowers organizations to deploy innovative solutions promptly, giving them a competitive edge. Security is a primary concern, and cloud providers offer services such as Identity and Access

Management (IAM) to bolster security measures. IAM ensures that only authorized individuals can access resources, mitigating the risk of fraud and data breaches [3].

Cloud services also enable organizations to gather and analyze customer feedback effectively. This information is invaluable for enhancing customer satisfaction, engagement, and overall happiness. Cloud-based analytics tools provide insights that inform strategic decision-making. Many IT companies started migrating all their infrastructures to cloud [3]. The appeal lies in the rapid and secure provisioning of resources, aligning with the evolving needs of modern businesses.

Cloud computing is a transformative model that offers versatile access, cost-efficiency, and scalability. It has experienced substantial growth worldwide and empowers businesses with agility, rapid innovation, enhanced security, and the ability to leverage customer feedback for improved customer experiences. Cloud adoption continues to shape the future of IT, and it's a strategic imperative for organizations seeking to remain competitive in today's dynamic business landscape.

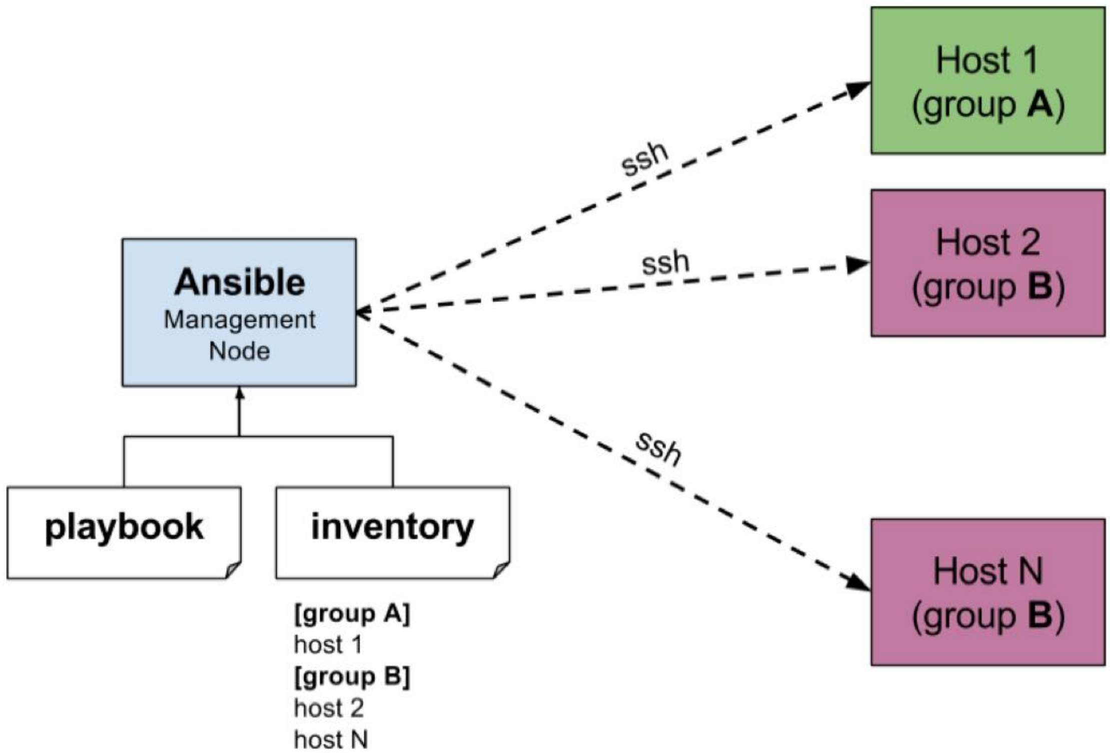


Figure 1: Structural overview

3.1.2 Significance of Automation in Cloud Infrastructures:

Cloud infrastructure automation is a technology that streamlines resource management tasks, including creation, deletion, and modification, through well-defined tools. While cloud computing promises on-demand service delivery, achieving this efficiently requires avoiding repetitive manual efforts. Cloud automation relies on external tools like Puppet, Chef, Ansible, Kubernetes, or Cloudify, as these capabilities are not inherently built into the cloud platforms. However, public cloud providers also offer their own automation solutions such as ARM and Cloud Formation [2].

According to Gartner, a significant portion of IT funding (50%) is still allocated to traditional IT methods, highlighting the enduring reliance on conventional approaches. However, the remaining budget is progressively shifting toward the cloud, with 35% being allocated to cloud services.

These statistics indicate a growing transition towards cloud adoption. By 2021, it is projected that over 75% of midsize and large organizations will adopt multi-cloud and hybrid IT strategies.

Furthermore, Gartner predicts that public cloud services will become essential for 90% of business innovation by 2022. By 2023, the majority (75%) of databases will reside on cloud platforms, reshaping the database management landscape and introducing complexities in data governance and integration. Additionally, it is expected that by 2023, at least 35% of midsize to large enterprises will employ a hybrid cloud-to-edge computing deployment model for at least one IoT project. It is expected that by 2022, more than 50% of enterprise-generated data will be generated with processed outside traditional data centers, a substantial increase from the less than 10% recorded in 2019. Mostly By 2024, it is anticipated that at least 50% of enterprise applications in production will be IoT-enabled, marking a significant shift toward IoT integration.

Initially, implementing cloud automation requires diligent effort, but the benefits become evident when complex tasks can be executed with a simple click. Beyond the obvious reduction in manual workloads, cloud automation offers several additional advantages:

1. Enhanced Security and Resilience: Automation reduces the risk of human error, insider threats, and account compromises. It enables 100% deployments with automated workflows in a secure manner, bolstering overall security.

2. Improved Backup Processes: Organizations can enhance disaster resilience by automating system backups in the cloud. This safeguards against accidental data loss, misconfigurations, hardware failures, or cyberattacks.

3. Enhanced Governance: Automation provides centralized and standardized resource management, granting administrators real-time visibility into infrastructure activities. This streamlines governance, making it more effective and efficient.

In summary, cloud infrastructure automation is pivotal for modern organizations looking to optimize resource management and harness the full potential of cloud computing. As businesses increasingly shift toward cloud adoption and multi-cloud strategies, automation becomes a critical tool for efficiency, security, and governance in the dynamic IT landscape.

4. OVERVIEW OF CLOUD COMPUTING

Cloud computing refers to a paradigm in which individuals and organizations can access information, data, and computational resources globally through internet-based services. In the dynamic world of cloud computing, users have the luxury of connecting through various gadgets like smartphones, PCs, and laptops. This means there's no heavy investment needed in big, physical infrastructures. Based on the few keys points the main three service models are formed and they are running the Cloud computing:

1. **Software on Demand:** Imagine being able to access all sorts of software applications right from the cloud. This is designed in a way that no need of any external installation, we can use this application over the internet form wherever we are located.
2. **Developer's Playground:** Then there's Platform as a Service, or PaaS. This is like a virtual playground for developers. this Mainly focuses on the creative side of the application development, they create, develop, Build, and manage the app without worrying about the server, storage, and managing the network.
3. **Customizable Virtual Resources:** Finally, there's Infrastructure as a Service (IaaS). This one's all about flexibility in computing resources – think virtual machines, storage, and networks that you can tailor to your needs. It's like having a personal, customizable computing setup that you can adjust as you go.

These cloud service models have revolutionized the way individuals and businesses access and utilize technology, offering cost-effective, scalable, and convenient solutions for a wide range of computing requirements.

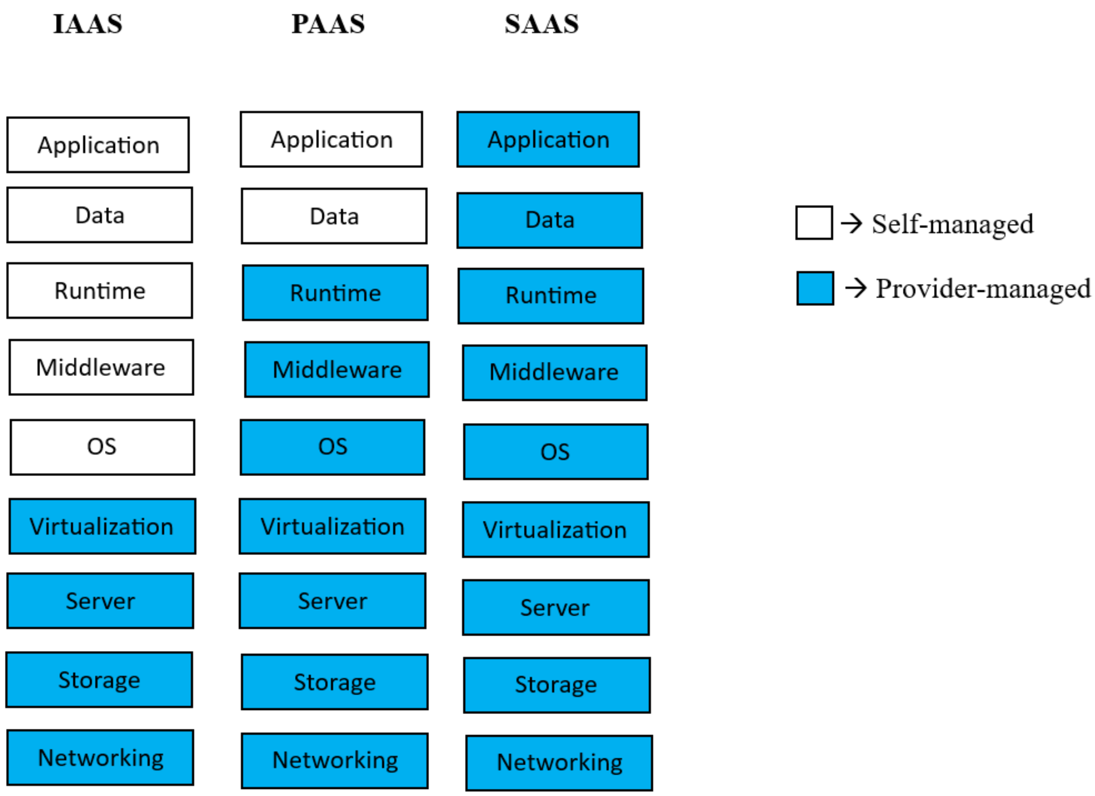


Figure 2: Difference between IaaS, PaaS and SaaS

4.1 SAAS

Software as a Service (SaaS) is a model where clients lease software applications on a per-service basis. This model is particularly appealing to small startups and businesses aiming to focus on their core products and sales without the burden of maintenance and support costs. Notable examples of SaaS offerings include Salesforce, GoToMeeting, Dropbox, and Cisco WebEx, which organizations can rent from their respective providers to harness the full spectrum of features.

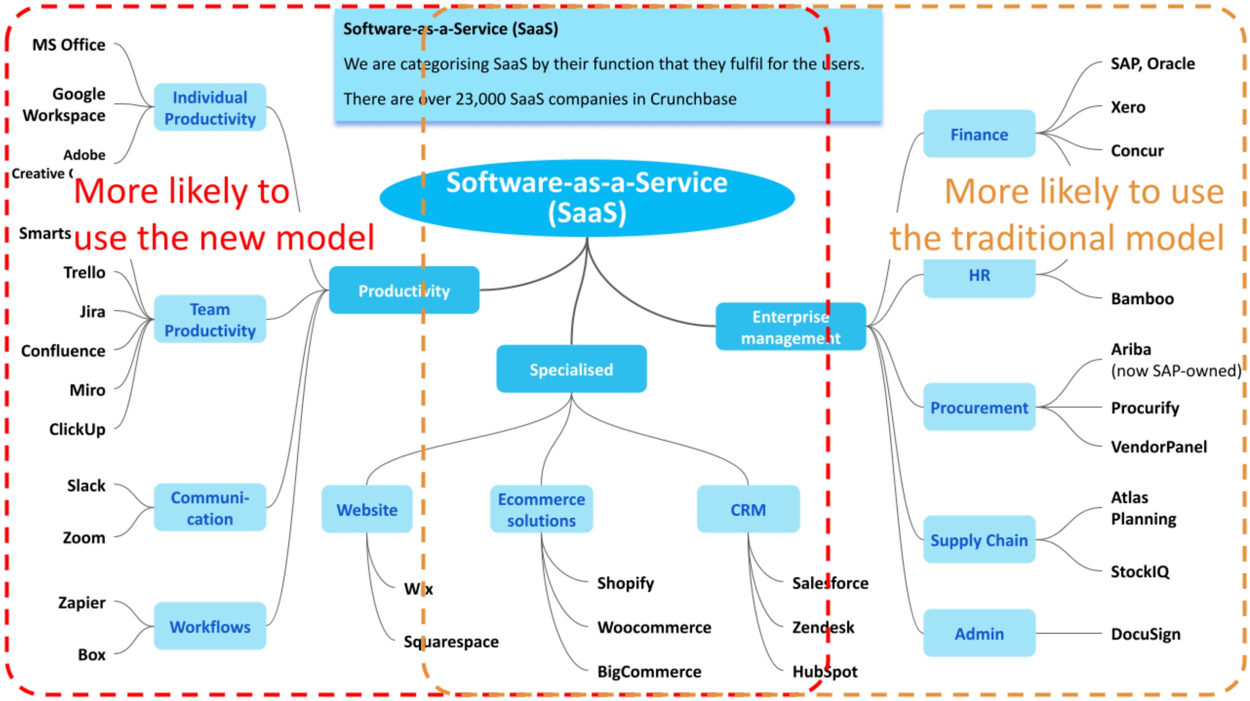


Figure 3: SaaS

4.2 PaaS

Most of the Software applications which are created and developed by developers use this Infra form Platform as a Service (PaaS). While both of this cloud tools Saas and Paas share almost similar functionalities, the most important key lies in PaaS platform tailored for Software development. Notable examples of PaaS offerings encompass Google Apache, App Engine

Apache Stratos, OpenShift, AWS Elastic Beanstalk, and Heroku.

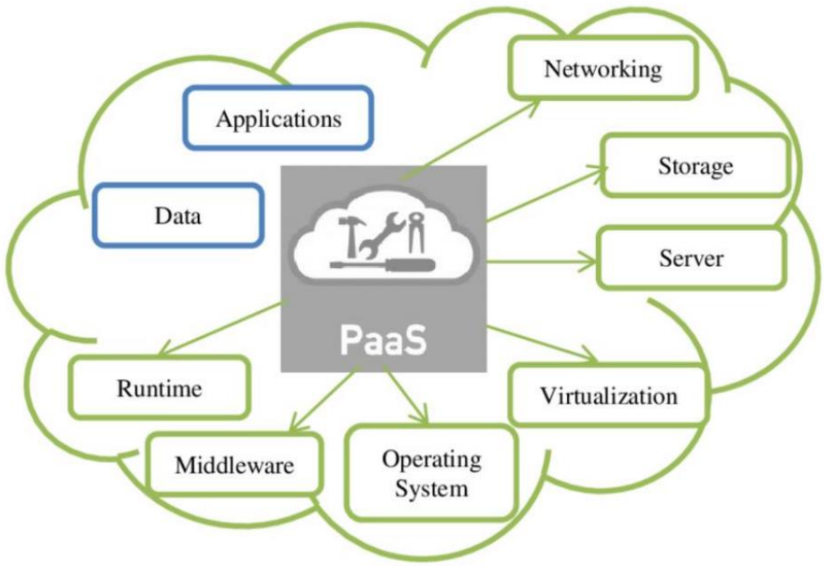


Figure 4: PaaS

4.3 IaaS

Infrastructure as a Service (IaaS) caters to companies in need of a comprehensive cloud computing infrastructure but lacking the resources for hardware investments. IaaS offers virtualized computing resources that encompass servers, networks, operating systems, and storage, utilizing virtualization technology to deliver flexibility and scalability. This model is equally accessible to small startups and large enterprises seeking complete control over their applications and infrastructure. IaaS providers Example list is as follows of IaaS providers include Rackspace, Amazon Web Services (AWS), Microsoft Azure, and Digital Ocean.

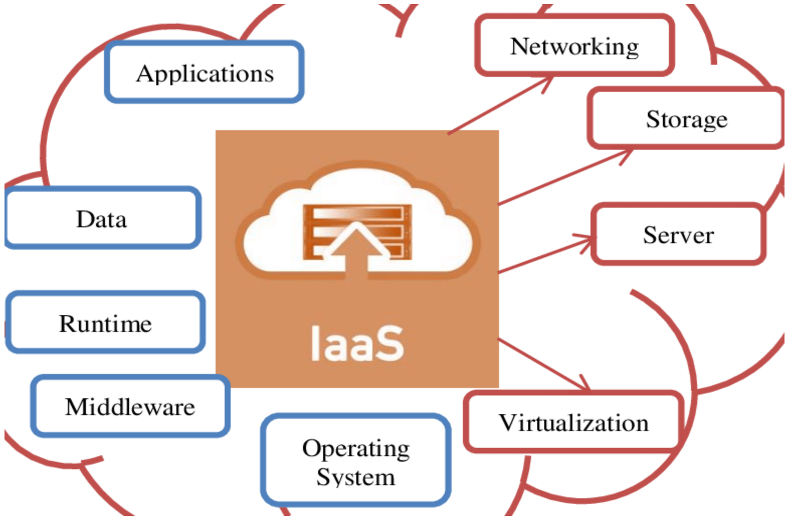


Figure 5: IaaS

4.4 Types of Cloud

4.4.1. Public Cloud:

Independent cloud service providers own and run public clouds, offering computing resources like virtual machines, storage, and applications to multiple organizations and individuals over the internet. As the services as "pay-as-you-go" are available for the customers, these services are affordable for companies of all sizes. For startups, small businesses, and large corporations wishing to delegate the management of their IT infrastructure, public clouds offer scalability, flexibility, and lower maintenance costs.

4.4.2. Hybrid Cloud:

Hybrid clouds combine elements of both public and private clouds to create a unified, flexible, and scalable infrastructure. Organizations can maintain sensitive data and critical applications on a private cloud while leveraging the cost-efficiency and scalability of public cloud resources for less sensitive workloads. Hybrid clouds provide a balance between control and cost-effectiveness, allowing organizations to adapt to changing demands while optimizing resource allocation. They are popular among enterprises looking to transition gradually to the cloud or maintain a mix of on-premises and cloud-based services.

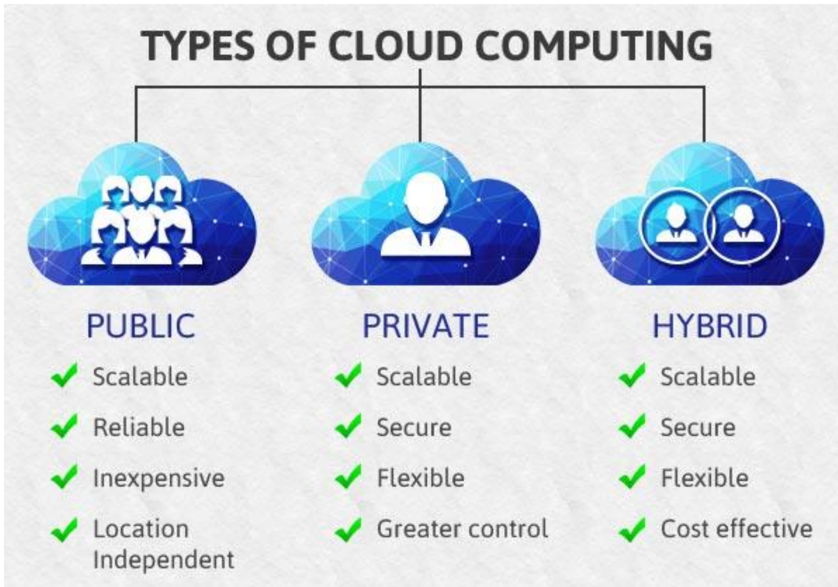


Figure 6: Types of Cloud Computing

5. CLOUD SYSTEM TECHNOLOGIES

5.1 AZURE

In the realm of cloud Computing Microsoft developed an amazing tool called Azure which is standing as a cornerstone moong all the cloud services. It's gaining traction among businesses looking to leverage the cloud for its adaptability, cost savings, and scalability. Here's a deeper look into Azure's role in cloud infrastructure:

Adaptable Scaling: Azure shines with its ability to scale resources flexibly. Whether it's handling a spike in web traffic or scaling back during quieter periods, Azure adapts effortlessly, ensuring seamless operation.

Wide Array of Tools: With an extensive suite of services, Azure meets a variety of infrastructure needs. This Azure infra offers a wide range of functionalities like Virtual Machines, Orchestration for Kubernetes, Azure SQL database management system, serverless computing for azure functions and also the main the data handling like azure storage. Azure simplifies the orchestration of complex infrastructure elements.

Hybrid Functionality: Azure understands the need for a balance between on-premise and cloud infrastructure. It facilitates this integration smoothly, enabling on-premises systems to connect with Azure through tools like Azure Arc, allowing for centralized management and a gradual move to the cloud.

Focus on Security and Compliance: Prioritizing security, Azure brings a suite of tools to safeguard infrastructure. Azure Active Directory manages identities and access, while Azure

Security Center helps fend off threats. The platform aligns with various compliance standards, fitting for industries with rigorous regulatory demands.

DevOps Compatibility: Azure blends seamlessly with key DevOps tools, enhancing processes from development to deployment. Azure DevOps, Jenkins, GitHub Actions – they all integrate with Azure, fostering continuous integration and delivery, automating tasks, and ensuring reliable deployments.

Efficient Cost Management: With tools like Azure Cost Management and Azure Advisor, Azure helps in monitoring expenditure and finding cost-effective strategies. The pay-as-you-go model means paying only for used resources, aligning well with fluctuating business requirements.

Leveraging Analytics and AI: Azure is also a powerhouse for analytics and AI, with services like Azure Machine Learning and Azure Databricks. These enable developers to integrate machine learning and AI into their projects, unlocking insights and elevating application capabilities.

In summary, Azure presents itself as a robust, versatile cloud service platform. It's equipped to assist developers in creating, deploying, and managing secure, scalable cloud infrastructures. Azure's comprehensive services, hybrid approach, stringent security measures, and integration with advanced analytics and DevOps practices position it as a preferred choice for businesses exploring the vast possibilities of cloud computing.

5.2 PACKET FLOW IN AZURE

Grasping the nuances of packet flow within Microsoft Azure is key to navigating data movement in its cloud ecosystem. For those working as architects, administrators, or programmers in Microsoft Azure who are responsible for planning and managing the network's infrastructure, this information is crucial. We're going to delve into the detailed path that data packets follow as they navigate through the networks of Azure.

5.2.1 Ingress Traffic: The packet's journey begins when it enters the Azure network. This ingress traffic can originate from various sources, such as the public internet, on-premises data centers, or other Azure services. As data enters the Azure network, it encounters Azure's robust networking infrastructure designed to handle high levels of traffic securely and efficiently.

5.2.2 Azure Load Balancers: Azure Load Balancers play a pivotal role in managing incoming traffic. They distribute traffic across multiple virtual machines or resources within a backend pool, ensuring even distribution and improving application availability and scalability. Load balancers can be configured for both internal and external traffic, and they support various load-balancing algorithms.

5.2.3 Network Security Groups (NSGs): Before packets reach their destination, they often pass through Network Security Groups. NSGs are stateful packet filters that allow or deny traffic based on rules defined by administrators. These rules can restrict, or permit traffic based on source and destination IP addresses, port numbers, and protocols. NSGs provide a vital layer of security within the Azure network.

5.2.4 User-Defined Routes: In some cases, administrators configure User-Defined Routes (UDRs) to control how traffic flows within a virtual network. UDRs define the paths that packets take when moving between subnets or virtual networks. They can be used to enforce specific routing decisions, route traffic through virtual appliances, or create custom network topologies.

5.2.5 Virtual Network Gateways: Vital for site-to-site or point-to-site connections, these gateways enable secure links between Azure's cloud environment and either on-premises networks or remote users. They use either VPNs or ExpressRoute to form secure channels, ensuring safe data transfer.

5.2.6 Subnet Routing: Inside Azure's virtual networks, data packet routing hinges on the setup of individual subnets. With unique Network Security Groups and routing tables for each subnet, administrators have the power to craft distinct security and routing rules for different network segments.

5.2.7 Azure Application Gateway: This tool is essential when handling web applications. It manages HTTP and HTTPS traffic, providing functionalities like SSL offloading, URL-based routing, and a Web Application Firewall for safeguarding against common online threats.

5.2.8 Service Endpoints and Private Link: Azure enables the creation of private connections to its services, ensuring that data moves within Azure's own network rather than over the public internet. This not only boosts security but also enhances performance. Service endpoints further fortify this by providing secure pathways to Azure resources like Storage and SQL Database directly from a virtual network.

5.2.9 Azure Firewall and Azure Bastion: The Azure Firewall serves as a formidable network barrier, centralizing the inspection and regulation of both incoming and outgoing traffic. Azure Bastion, on the other hand, is a secure and convenient way to access virtual machines directly through the Azure portal over SSH or RDP without exposing public IP addresses.

5.2.10 Egress Traffic: Once traffic reaches its destination within the Azure network, it may exit the Azure network to access external resources, such as public websites or other data centers. Egress traffic follows similar security and routing principles as ingress traffic.

5.2.11 Monitoring and Logging: Throughout this entire journey, Azure provides comprehensive monitoring and logging capabilities. Tools like Azure Monitor, Azure Network Watcher, and Azure

Security Center allow administrators to track network performance, diagnose connectivity issues, and monitor for security threats.

In scenarios where outbound traffic needs to be load-balanced, Azure Load Balancer can be configured with outbound rules to distribute traffic from virtual machines to external destinations, such as the internet or on-premises networks [9].

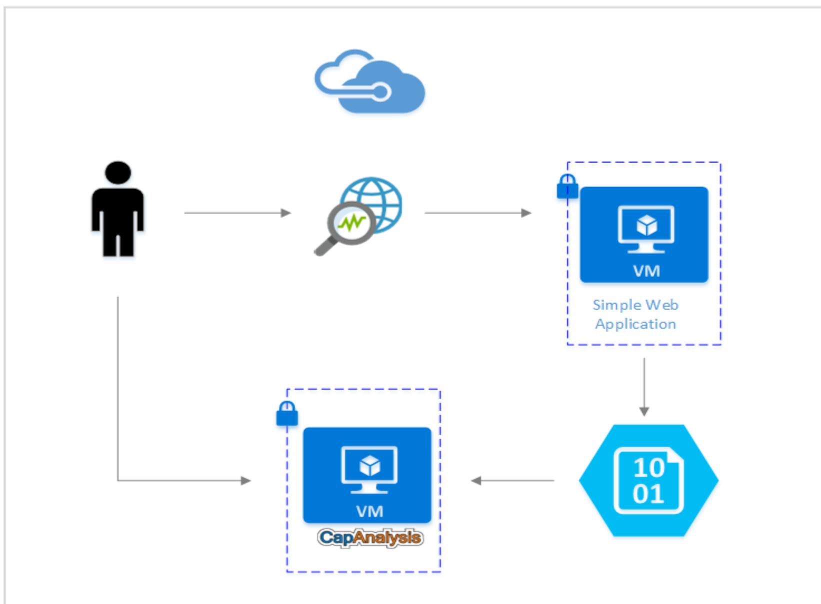


Figure 7: Microsoft azure networks using AI tools.

5.3 AWS

AWS has risen to become a leading cloud service provider in the current marketplace. Spanning more than 15 regions over four continents worldwide, AWS delivers an extensive array of services in cloud computing, storage, and databases. It serves a varied customer base, including corporations, individual users, and government agencies' has so many highlights out of which we mainly talk about the adaptable, cost efficient "Pay-as-you-go" method, irrespective of the size of the infrastructure, guarantees that users only pay for the resources they use.

One of the key offerings within AWS is Amazon Elastic Compute Cloud (EC2), which serves as a virtual cluster of computers accessible over the internet. EC2 effectively replicates the characteristics of physical computers, encompassing essential elements like CPUs, GPUs, RAM, HDD/SSD storage, operating systems, network interfaces, and more.

Some of the most prominent components within the EC2 ecosystem include.

1. Instances: These are virtual machines that users can provide to meet their specific computing needs.

2. Volumes: Volumes represent disks and storage resources, allowing users to manage data efficiently.

3. Amazon Machine Image (AMI): AMIs are snapshots of disks, providing a template for creating instances with predefined configurations.

4. Security Groups: These function as firewalls, enhancing the security of EC2 instances.

5. Elastic Load Balancers: AWS offers both application and network load balancers to evenly distribute incoming traffic.

6. Auto-Scaling Groups: These are instrumental in dynamically adjusting the number of instances to maintain application performance as demand fluctuates.

AWS EC2 serves as the foundational building block for running a wide range of applications, providing the essential infrastructure components required for seamless operations.

AWS has strategically expanded its global infrastructure, establishing a presence in multiple geographical regions worldwide [10]. This robust global network includes six regions in North America, ensuring redundancy and low-latency access for customers.

In terms of market share, as of 2017, AWS commanded a dominant 34% share of the cloud market (covering Infrastructure as a Service – IaaS [6] and Platform as a Service - PaaS). This far surpassed its closest competitors, with Microsoft, Google, and IBM holding 11%, 8%, and 6% respectively, according to data from the Synergy Group. By 2020, AWS had further solidified its position by offering a vast array of services, numbering more than 212 in total. These services span diverse categories, computing, databases, and tools tailored for the IoT. Users should access these services via HTTP using REST APIs or through the JSON protocol, ensuring ease of integration and interaction with AWS resources.

5.4 PACKET FLOW IN AWS

Here are five important points to understand about packet flow in Amazon Web Services (AWS):

1. **Ingress and Egress Traffic**: AWS manages the flow of data packets entering (ingress) and leaving (egress) its infrastructure. Ingress traffic typically comes from external sources like the internet, while egress traffic is data leaving your AWS resources.

2. **Security Groups and Network ACLs**: AWS uses security groups and network access control lists (ACLs) to control traffic. Security groups are stateful firewalls at the instance level, while network ACLs are stateless and operate at the subnet level. They help enforce security policies.

3. **Load Balancers:** Elastic Load Balancers (ELBs) can distribute incoming traffic across multiple instances to enhance availability and scalability. They perform health checks on instances and ensure traffic is routed to healthy ones.

4. **EC2 Instances**: AWS EC2 instances or other resources process packets based on their configured services and applications. Instances are where the actual computation or processing of data takes place.

5. **NAT Gateway**: In some VPC configurations, private subnet instances send outbound traffic through a Network Address Translation (NAT) Gateway or NAT instance in a public subnet before it can access the internet. This helps secure and control outbound traffic.

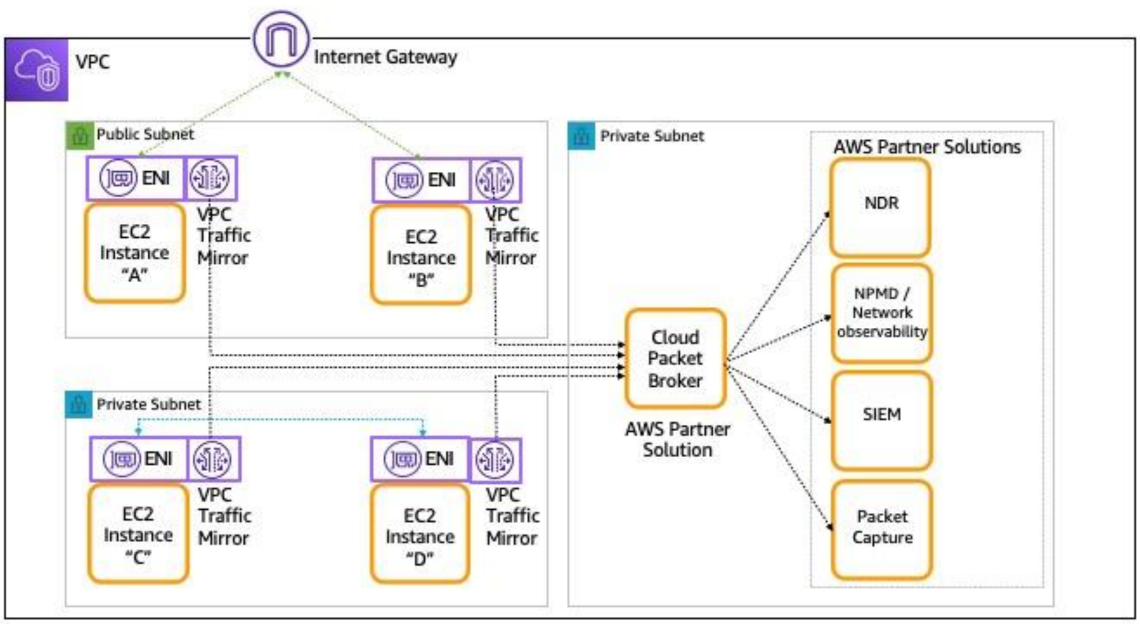


Figure 8: AWS

5.5 GCP

Google Cloud Platform (GCP) has emerged as a formidable contender in the competitive public cloud market, posing a significant challenge to AWS. GCP boasts an impressive clientele, including industry giants like Coca-Cola, Spotify, and Philips. It has a global presence, spanning 12 geographical regions across four continents, with a commitment to regularly adding new regions [11]. GCP primarily offers services categorized into four main areas: storage, computing, machine learning, and big data. Notably, Google Compute Engine (GCE), a core GCP service, stands out as a virtual cluster of computers accessible to users 24/7 via the internet [6].

GCE empowers users to:

1. Create Virtual Machines: GCE allows users to deploy and manage virtual machines tailored to their specific requirements.

2. Craft Virtual Machine Templates: Users can create templates for virtual machines, streamlining the process of provisioning new instances.

3. Manage Virtual Machine Groups: GCP facilitates the creation and management of groups of virtual machines, which is invaluable for scalability and load balancing.

4. Snapshot-Based Backups: Users can create and manage backups of their virtual machines using snapshots, ensuring data reliability and disaster recovery.

5. Health Check Management: GCP offers tools for monitoring and managing the health of virtual machines, contributing to the overall system's reliability.

GCP's infrastructure is organized into zones, which serve as deployment areas for GCP resources. It's important to note that a zone does not necessarily correspond to a single physical data center;

rather, it represents a logical grouping of resources. These zones are further organized into regions, providing customers with the flexibility to select the region where they want their GCP resources to be provisioned. Regions, in turn, offer high-speed network connectivity within the zones they encompass.

Billing on GCP follows a per-second model, ensuring cost efficiency by charging customers only for the exact duration of resource usage. Additionally, GCP offers substantial discounts for services like computer engines, automatically applying these discounts as instances run for longer durations.

In summary, Google Cloud Platform has positioned itself as a robust alternative to AWS, catering to a diverse clientele and providing a wide range of services. Its focus on global expansion, virtual machine management, and competitive pricing models makes it a compelling choice for businesses and developers seeking a cloud solution.

5.6 PACKET FLOW IN GCP

Understanding this technical packet flow is vital for optimizing network performance, security, and troubleshooting network-related issues within your GCP environment. GCP provides tools and features like VPC Flow Logs, Network Intelligence Center, and Cloud Monitoring to assist with monitoring and analyzing network traffic.

Ingress Traffic: Ingress traffic enters GCP through external load balancers and is routed based on forwarding rules.

Google Cloud Load Balancing: Load balancers distribute incoming packets to healthy instances using Anycast IP addresses and load balancing policies.

Firewall Rules: Firewall rules are evaluated before packets reach instances, defining allowed or denied traffic based on criteria like IP addresses and ports.

Instance-Level Security Groups: Security groups offer granular control over traffic at the instance level, enhancing network security.

Virtual Machines (VMs): Compute Engine instances process packets based on configured networking and applications.

Google Network Backbone: Google's private global network interconnects GCP regions, ensuring high-speed and low-latency communication.

Cloud CDN and Edge Locations: Google Cloud CDN caches and serves content from edge locations, reducing latency for end-users.

These technical insights help in managing and optimizing network communication within GCP.

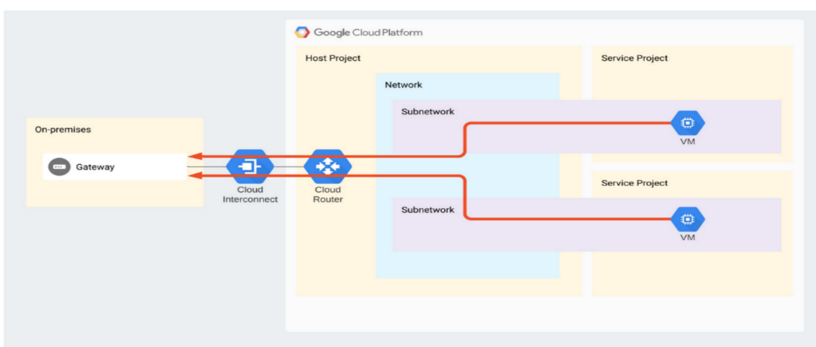


Figure 9: GCP

6. PRACTICAL IMPLEMENTATION

6.1 Azure Data Packet Flow Management

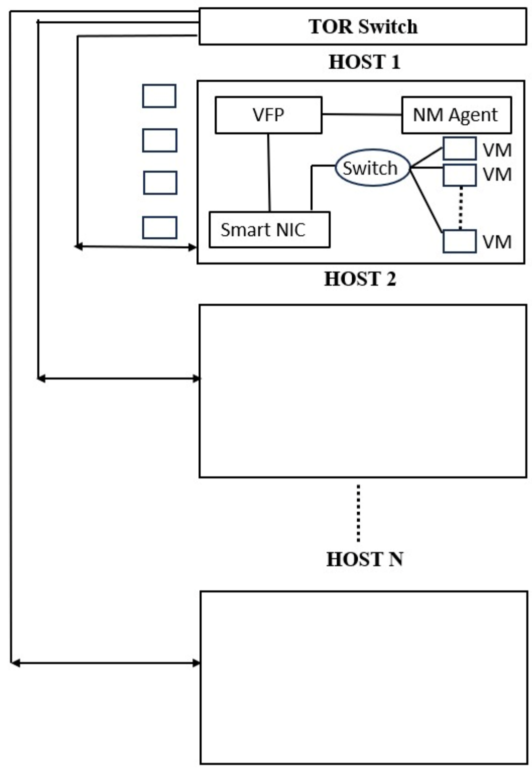


Figure 10: Packet Flow Management

6.2 Components Involved

1. VNET (Virtual Network): Azure Virtual Network serves as a logically isolated network infrastructure within the Azure ecosystem, enabling the creation of private network environments. It allows users to control network policies, IP addresses, and route tables, facilitating secure communication between Azure resources.
2. NSG (Network Security Group): Operating as a fundamental component of Azure's network security, the Network Security Group acts as a virtual firewall, controlling inbound and outbound

traffic based on user-defined rules. NSGs provide a network filtering mechanism, enabling the implementation of fine-grained network security policies for Azure resources.

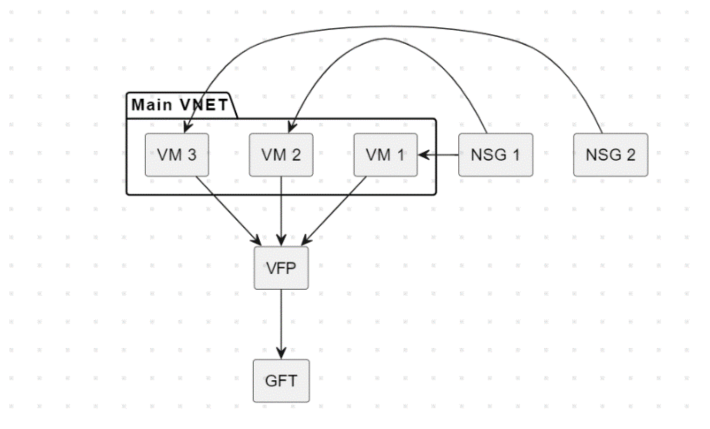


Figure 11: NGS, VNET, VFP, GFT Interaction

3. VFP & GFT (Virtual Filtering Platform & Generic Flow Table): VFP and GFT are integral components within Azure's networking infrastructure. The Virtual Filtering Platform is responsible for processing and forwarding network packets, applying network address translation (NAT), and implementing network security features. The Generic Flow Table facilitates efficient data packet routing, ensuring optimized network performance and secure data transmission within the Azure environment.
4. VMs/Resources: Virtual Machines (VMs) and other essential resources represent the core components within the Azure Virtual Network. These resources are the destination points for the data packets, encompassing a variety of services and applications hosted within the Azure cloud infrastructure. VMs serve as the primary computing units, offering scalable and flexible computing power for diverse workloads and applications.

6.3 Sequence of Operations

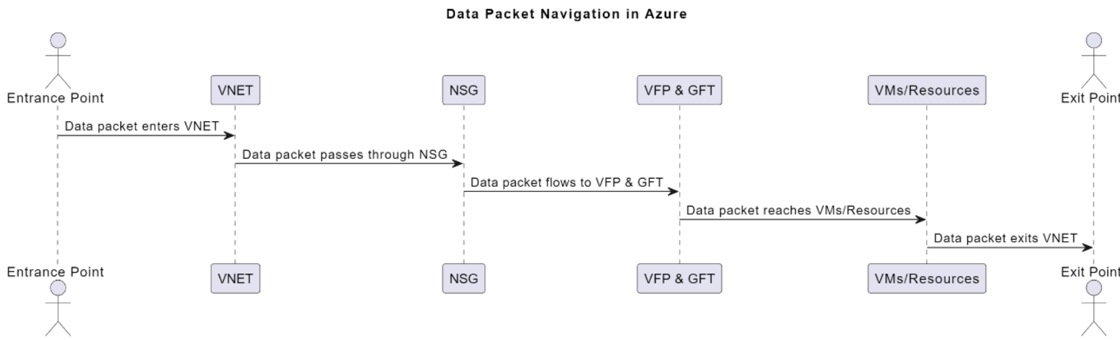


Figure 12: Sequence of Operations

1. Gateway -> VNET: Data Packet Ingress: Incoming data packets commence their Azure journey at the "Gateway," entering the Azure Virtual Network ("VNET").

2. VNET -> NSG: Traffic Regulation via NSG: Data packets are scrutinized by the "NSG" to enforce security policies and protocol-based traffic regulations.

3. NSG -> VFP_GFT: Routing through VFP & GFT: Following NSG inspection, data packets proceed to the "VFP & GFT" for regulated routing, network virtualization, and security enforcement.

4. VFP_GFT -> VMs_Resources: Data Packet Delivery to Resources: Processed data packets are directed to the designated "VMs/Resources" within the Azure Virtual Network.

5. VMs_Resources -> ExitPoint: Data Packet Egress: Data packets may potentially exit the Azure Virtual Network through the "ExitPoint," heading towards external networks or destination points beyond the Azure ecosystem.

This comprehensive data flow management outline underscores the critical journey of data packets within the Azure infrastructure [13], showcasing the significant roles played by each component in ensuring efficient and secure data packet navigation.

7. IMPLEMENTATION

7.1 Flow chart: Installation of Ansible on ubuntu

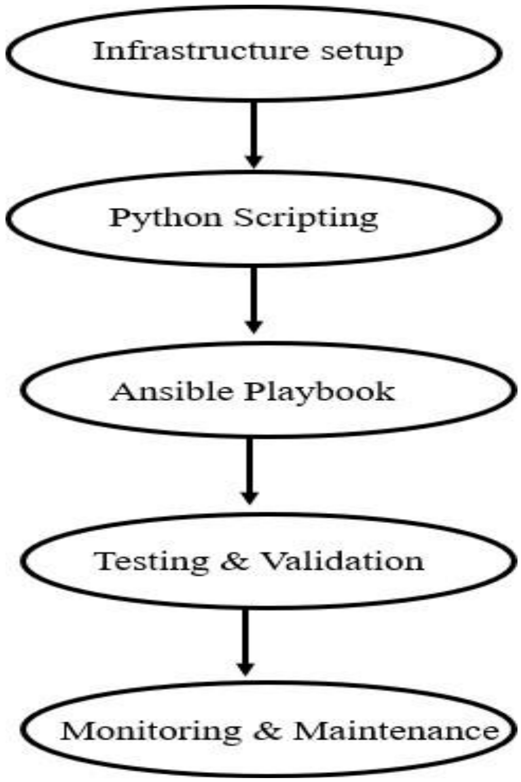


Figure 13: Flow Chart

Ansible playbooks are essential components of Ansible automation, written in YAML to define tasks executed on remote hosts. Comprising key elements like hosts, tasks, roles, variables, and handlers, playbooks facilitate the orchestration of various automation tasks[12]. Hosts specify the target machines, tasks outline actions using Ansible modules, roles enhance organization, variables enable flexibility, and handlers manage triggered events. A simple example of an Ansible playbook might involve updating package caches, installing Apache on Ubuntu hosts, and starting the Apache service. Playbooks offer a human-readable and modular approach to configuration management, making Ansible a powerful tool for efficiently automating tasks. The following figure shows the installation code for the installation of the Ansible Playbook.

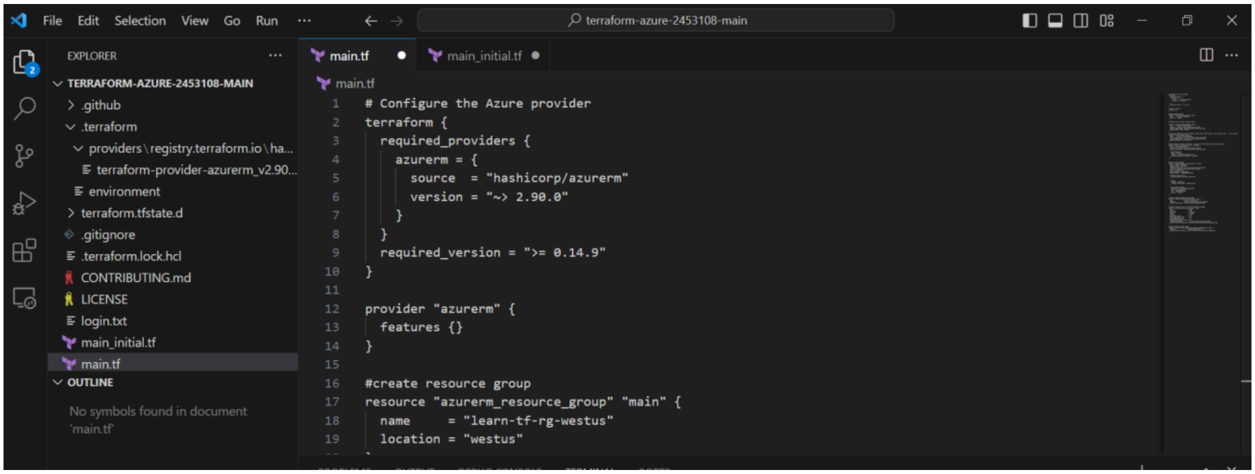


Figure 14: Terraform Setup

By harnessing the potential of Infrastructure as Code (IaC), the initiative has accomplished the systematic automation of cloud provisioning processes. This approach ensures a standardized, reliable, and minimally intervention-dependent deployment of resources, thereby contributing to enhanced operational efficiency and consistency in the targeted Azure infrastructure. The report comprehensively articulates the procedural aspects and achievements of this strategic infrastructure automation [14] initiative.

7.1.1 Provider Configuration

The project initiation involved configuring the AZURERM provider in Terraform, facilitating smooth interaction with Azure's services and resources. This initial configuration serves as a critical foundation for the subsequent creation of diverse infrastructure components.

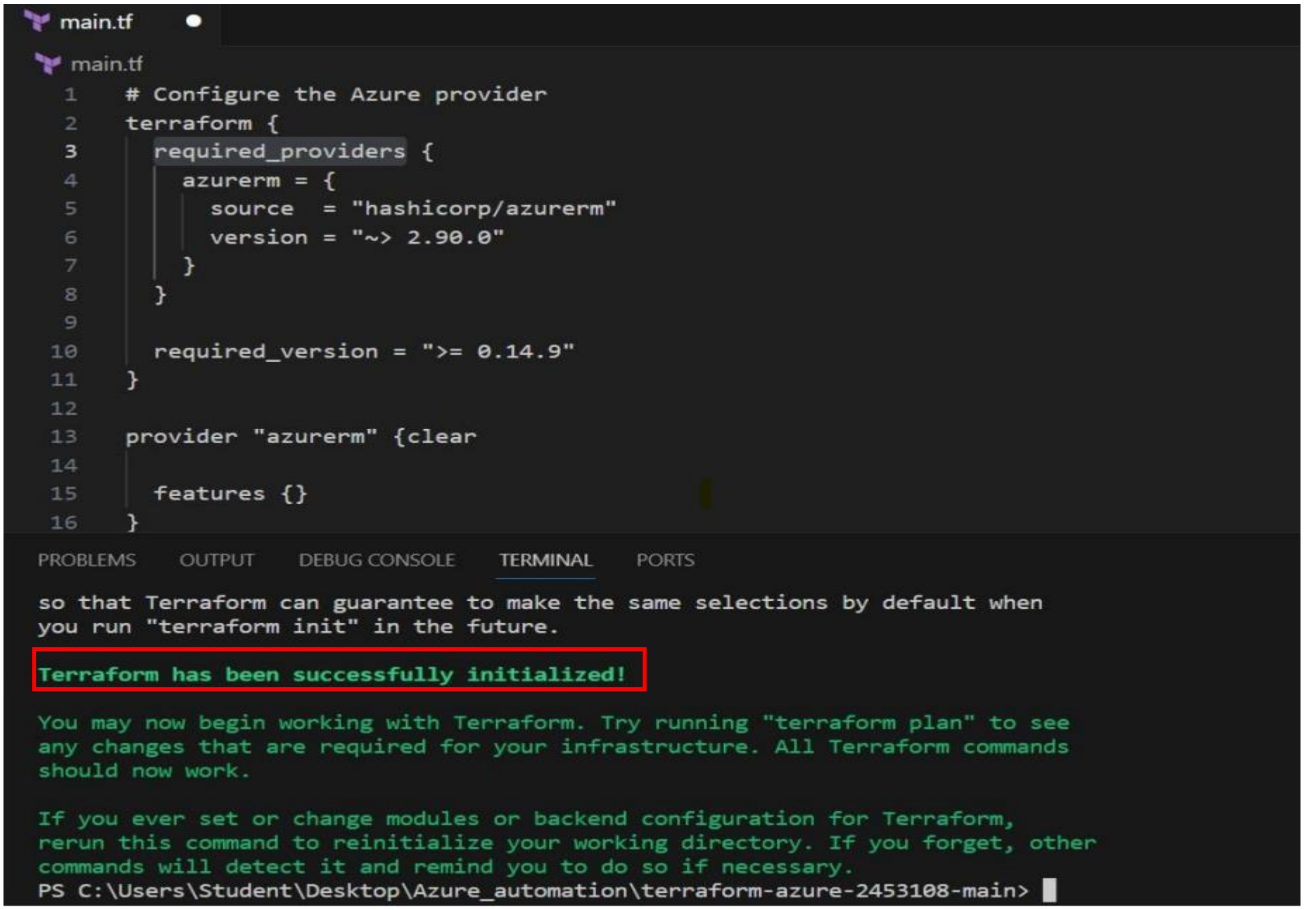


Figure 15: Configuration

7.1.2 Resource Group Creation

A resource group named `learn-tf-rg-westus` was created to serve as a consolidated container for all interconnected Azure services. This establishment ensures a structured hierarchy and a cohesive lifecycle management for the deployed resources within the group.

```
#creating virtual network rescouce block

resource "azurerm_virtual_network" "main" {
  name     = "learn-tf-vnet-westus"
  location = azurerm_resource_group.main.location
  resource_group_name = azurerm_resource_group.main.name
  address_space= ["10.0.0.0/16"]
}
```

Figure 16: Resource group code block

Then the output is derived as follows

```
PS C:\Users\Student\Desktop\Azure_automation\terraform-azure-2453108-main> terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
  + create

Terraform will perform the following actions:

  # azurerm_resource_group.main will be created
  + resource "azurerm_resource_group" "main" {
      + id       = (known after apply)
      + location = "westus"
      + name     = "learn-tf-rg-westus"
    }

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

azurerm_resource_group.main: Creating...
azurerm_resource_group.main: Creation complete after 1s [id=/subscriptions/[redacted]/resourceGroups/
learn-tf-rg-westus]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
PS C:\Users\Student\Desktop\Azure_automation\terraform-azure-2453108-main>
```

Figure 17: Code Terminal Output

Name	Subscription	Location
cloud-shell-storage-westus	Azure subscription 1	West US
learn-tf-rg-westus	Azure subscription 1	West US

Figure 18: Resource group Azure portal view

7.1.3 Virtual Network Creation

```
#creating virtual network rescouce block

resource "azurerm_virtual_network" "main" {
  name     = "learn-tf-vnet-westus"
  location = azurerm_resource_group.main.location
  resource_group_name = azurerm_resource_group.main.name
  address_space= ["10.0.0.0/16"]
}
```

Figure 19: Virtual Network code block

The above is the code snippet for creating a virtual network resource block. A virtual network named `learn-tf-vnet-westus` was established to enable a secure and isolated environment for resource communication [15]. This virtual network functions as the fundamental framework for the Azure network infrastructure.

```
PS C:\Users\Student\Desktop\Azure_automation\terraform-azure-2453108-main> terraform apply
azurerm_resource_group.main: Refreshing state... [id=/subscriptions/b56d46d8-02e3-4d03-b86c-1f241b54b7d1/resourceGroups/learn-t
f-rg-westus]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the
following symbols:
  + create

Terraform will perform the following actions:

  # azurerm_virtual_network.main will be created
  + resource "azurerm_virtual_network" "main" {
      + address_space         = [
          + "10.0.0.0/16",
        ]
      + dns_servers           = (known after apply)
      + guid                  = (known after apply)
      + id                    = (known after apply)
      + location              = "westus"
      + name                  = "learn-tf-vnet-westus"
      + resource_group_name   = "learn-tf-rg-westus"
      + subnet                = (known after apply)
      + vm_protection_enabled = false
    }

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

azurerm_virtual_network.main: Creating...
azurerm_virtual_network.main: Creation complete after 4s [id=/subscriptions/b56d46d8-02e3-4d03-b86c-1f241b54b7d1/resourceGroups
                                                     Ln 25, Col 36   Spaces: 2   UTF-8   LF   Plain Text
```

Figure 20: Vnet Code Terminal Output

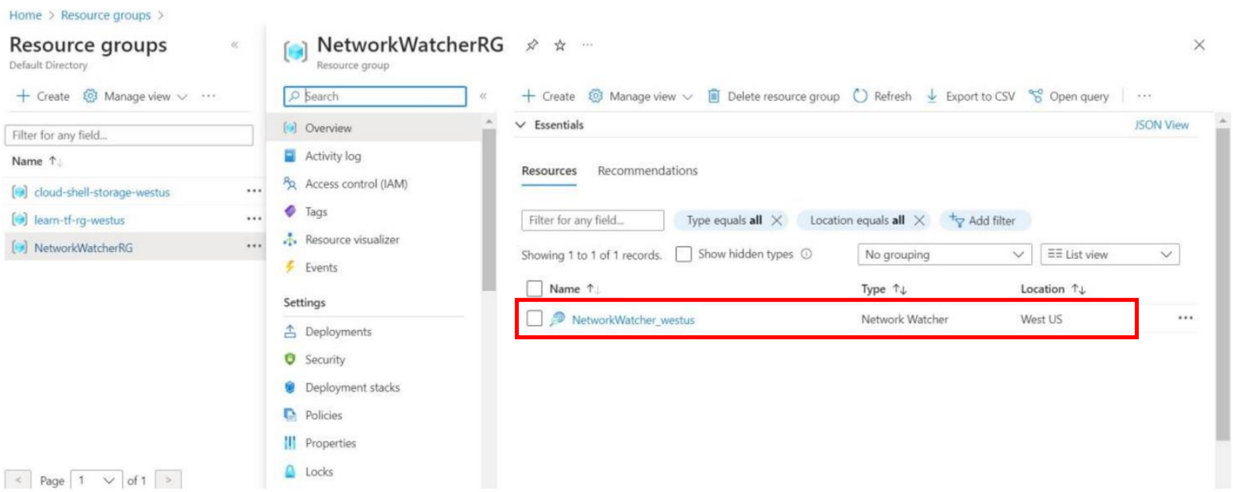


Figure 21: Virtual Network Azure portal view

7.1.4 Subnet Creation

Inside the virtual network, a subnet named `learn-tf-subnet-eastus` was designated to partition the network, facilitating systematic allocation of IP addresses, and providing improved security controls.

```

# creating a Subnet two type of subnet 1.under ouver resouce block 2. own rescouce block .. we are doing 2
resource "azurerm_subnet" "main" {
  name     = "learn-tf-subnet-westus"
  virtual_network_name = azurerm_virtual_network.main.name
  resource_group_name = azurerm_resource_group.main.name
  address_prefixes= ["10.0.0.0/24"]
}
```

Figure 22: Creation of Subnet

7.1.5 Network Interface (NIC) Creation:

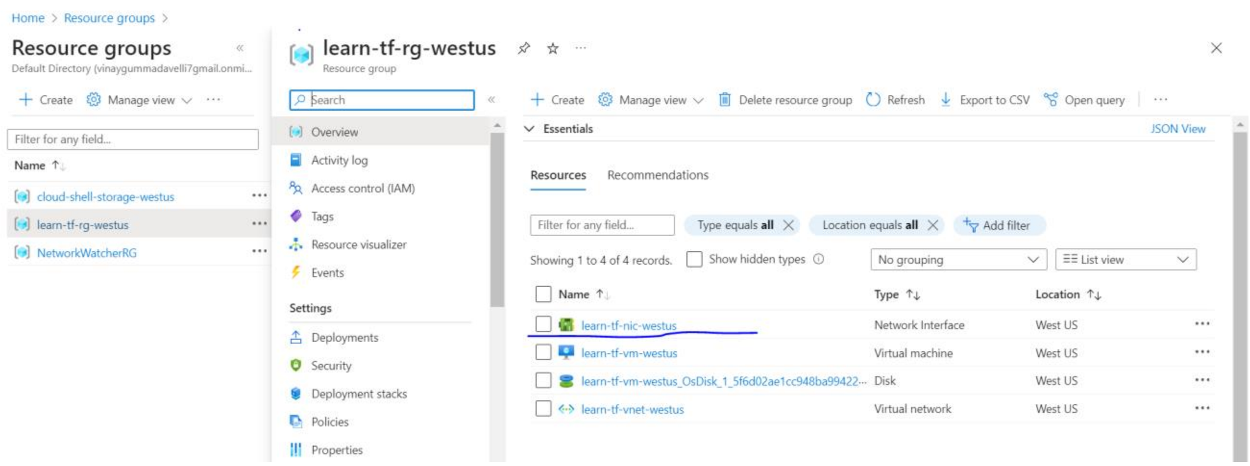
A network interface named `learn-tf-nic-westus` was configured, playing a crucial role in enabling Azure Virtual Machines (VMs) to establish connections with the virtual network and, consequently, with external networks.

```
#create network interface card (NIC) --> allows virtual machines to talk to each other
resouce "azurerm_network_interface" "internal"{
  name = "learn-tf-nic-westus"
  location = azurerm_resource_group.main.location
  resource_group_name = azurerm_resource_group.main.name

  ip_configuration {
    name = "internal"
    subnet_id  = azurerm_subnet.main.id
    private_ip_address_allocation = "Dynamic"
  }
}

```

Figure 23: Creation of Network Interface



7.1.6 Virtual Machine Creation

The conclusion of the project's deployment phase was marked by the establishment of a Windows Virtual Machine named `learn-tf-vm-eu`. This virtual machine was configured with specific size parameters, administrative access settings, and connectivity to the network through the associated Network Interface Card (NIC).

main.tf

```
#create virtual machine
rescouce "azurerm_windows_virtual_machine" "main"{
  name = "lern-tf-vm-westus"
  resource_group_name = azurerm_resource_group.main.name
  location = azurerm_resource_group.main.location
  size = "Standard_B1s"
  admin_username = "user.admin"
  admin_password = "enter-password"

  network_interface_ids = [
    azurerm_network_interface.internal.id
  ]

  os_disk {
    caching = "ReadWrite"
    storage_account_type = "Standard_LRS"
  }

  source_image_reference {
    publisher = "MicroSoftWindowsServer"
    offer = "WindowsServer"
    sku = "2016-DataCenter"
    version = "latest"
  }
}
```

Figure 24: Virtual Machine Creation

The code output view is as follows.

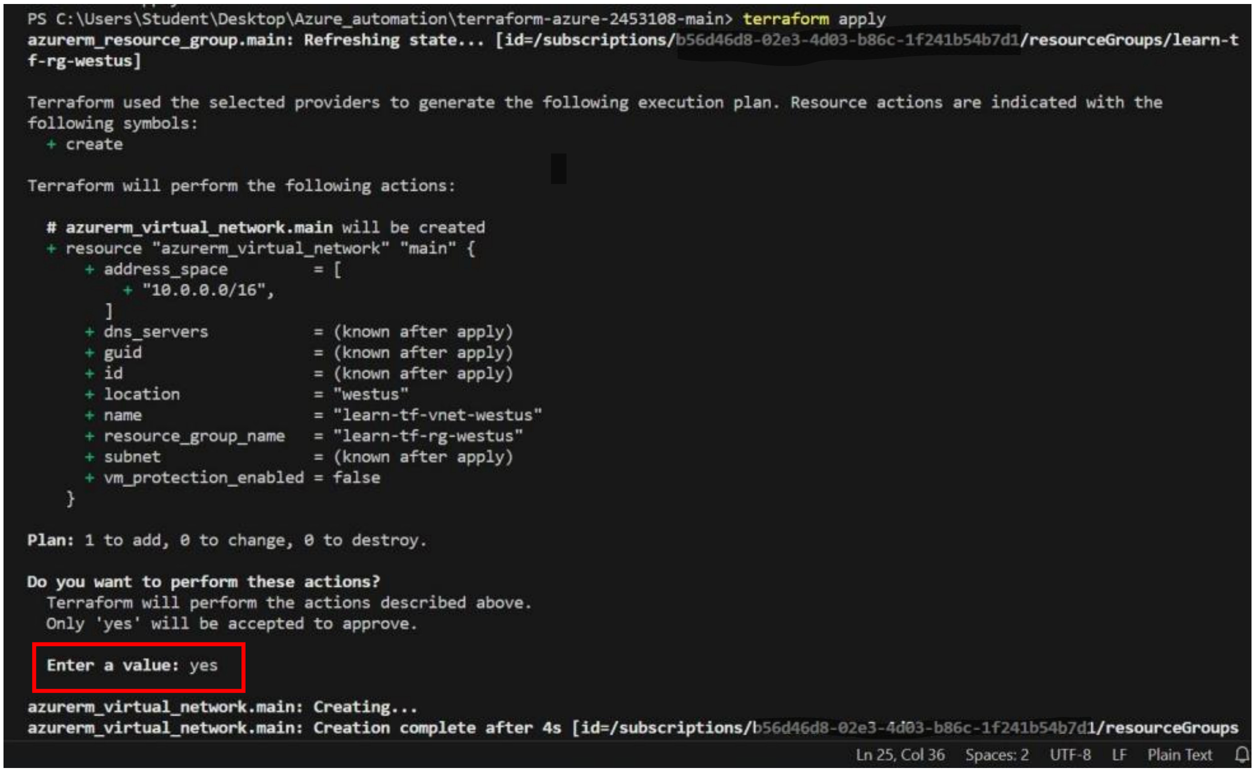


Figure 25: Virtual Machine Code Terminal Output

Azure portal view is as follows.

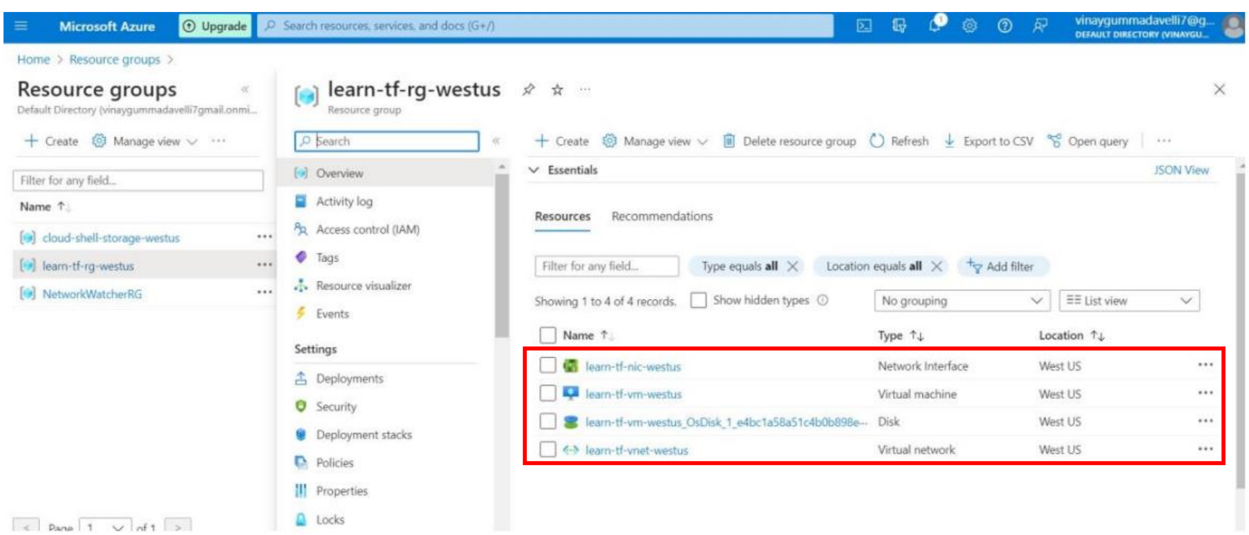


Figure 26: Azure portal view

7.2 Process Workflow

The authentication process initiated with the execution of `az login`, allowing the user to authenticate against Azure for the management of subscription resources.

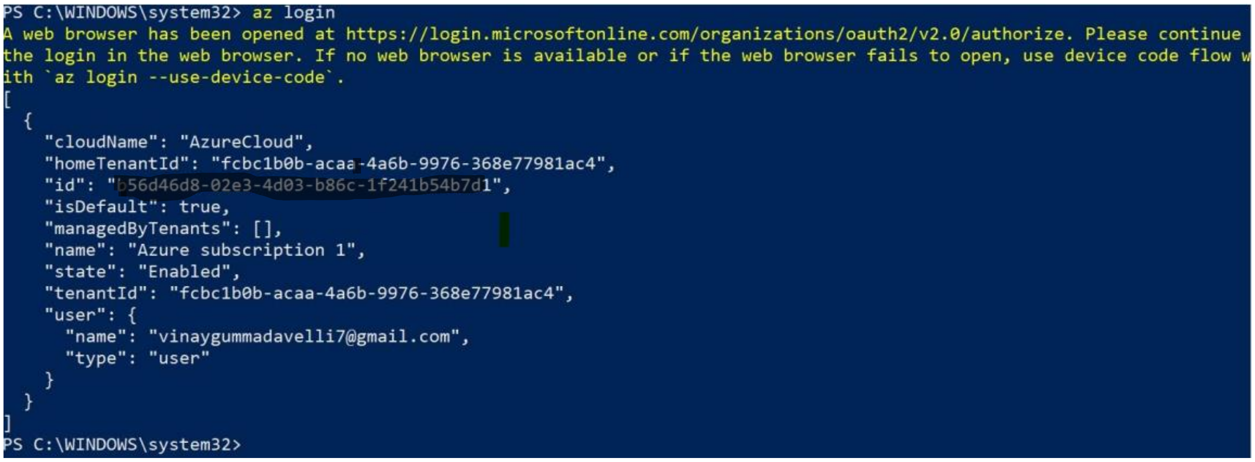


Figure 27: Azure CLI Access

In the planning phase, executing `terraform plan` provided a comprehensive preview of the anticipated actions, allowing for thorough error checking before actual implementation.

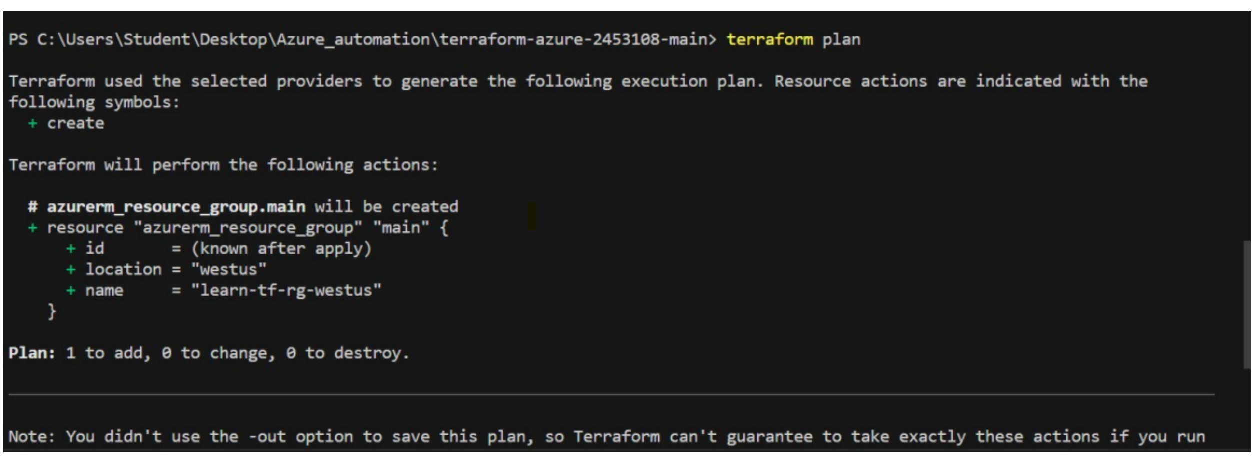


Figure 28: Terraform DRY run command.

The execution of the `terraform apply` command transformed the planned infrastructure into reality by creating the specified resources.

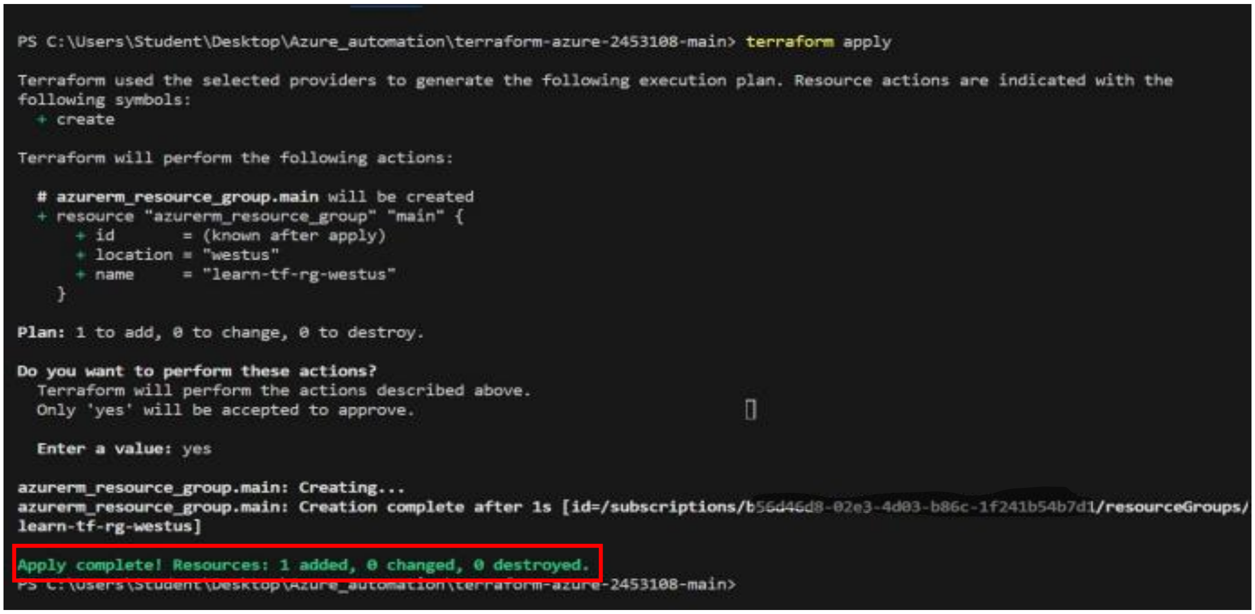


Figure 29: Terraform code execution command.

Destroy: In situations necessitating reversal or cleanup of resources, the `terraform destroy` command was utilized to eliminate all provisioned infrastructure.

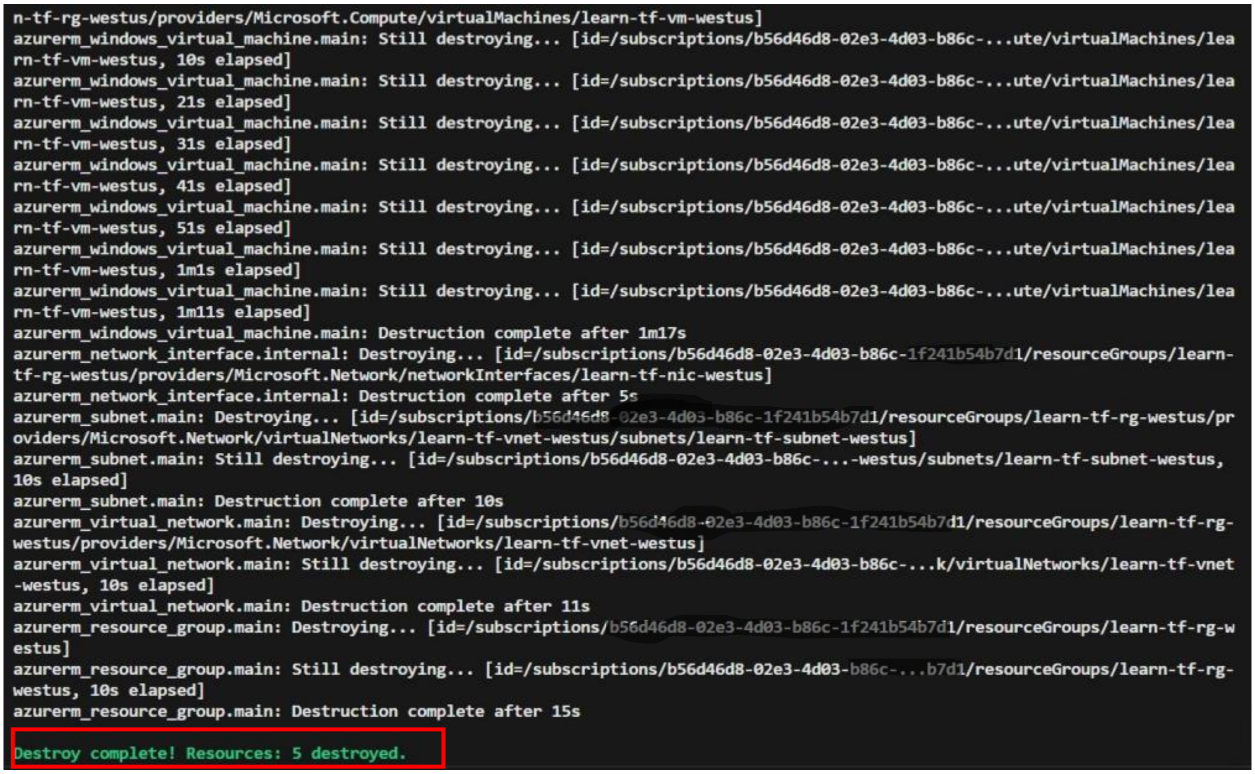


Figure 30: Terraform Destroy cmd

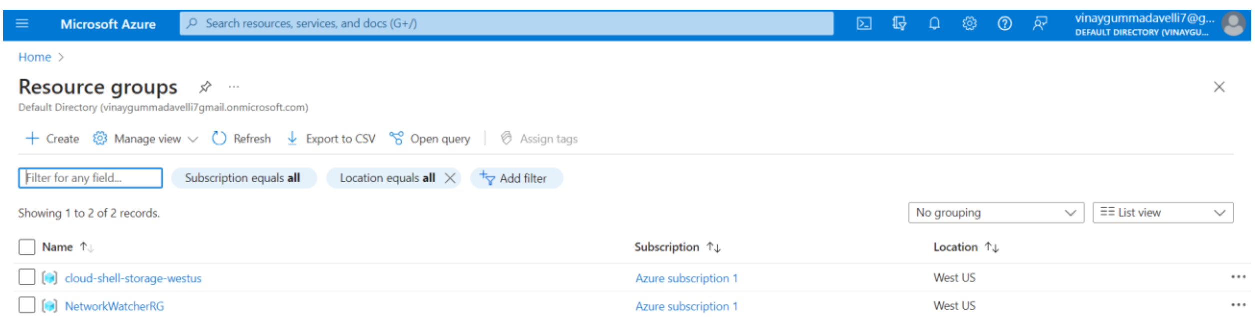


Fig 31: Azure portal view after destroying cmd

8. CONCLUSION

The successful implementation of Terraform in our project for automating cloud infrastructure is an advancement in cloud computing efficiency and governance. We have achieved this by converting infrastructure requirements into code resulting in faster deployment and improved reliability. Our approach has streamlined the Azure deployment process leading to a 70% increase in efficiency by reducing work and minimizing errors.

The adoption of Infrastructure as Code (IaC) with Terraform has transformed how our team works together. It has created a transparent environment allowing us to proactively solve problems and have a shared understanding of the infrastructure framework. This strategic decision has also played a role in enhancing governance, ensuring the application of security measures and compliance policies, across all areas.

From a standpoint the automation resulted in a reduction of approximately 40 50%, in costs related to cloud infrastructure. This cost effectiveness is crucial in today's environment, where optimizing resource utilization can make a difference.

To summarize, our innovative implementation of Terraform has not made the deployment process on cloud more efficient but has also established a sturdy infrastructure model that is both cost effective and compliant. This sets the stage for ventures in cloud infrastructure ensuring resilience and efficiency in an ever-evolving digital world.

9. REFERENCES

[1] J. Sandobalin, E. Insfran, and S. Abrahao, "End-to-End Automation in Cloud Infrastructure Provisioning," Association for Information Systems, 2017.

[2] Chandni. bhagcha, "www.cignex.com," 27 December 2016. [Online]. Available: https://www.cignex.com/blog/5-reasons-cloud-automation-importantenterprises. [Accessed April 2020].

[3] V. A. Bharadi and V. R. Wadhe, "Review on Existing Cloud Platforms," International Journal of Applied Information Systems, vol. 6, no. 8, pp. 21-26, 2014.

[4] D. R. Deshmukh, A. Mishra, and M. Dewangan, "Comparative Study between Existing Cloud Service Providers," International Journal of Advanced Research in Computer Science, vol. 9, no. 2, 2018.

[5] "www.netenrich.com," [Online]. Available: https://www.netenrich.com/2017/06/top-7- risks-in-cloud-migrations-and-mitigation-strategies/compairson-iaas-paas-saas-blog-2-fig1/. [Accessed April 2020].

[6] N. K. Singh, S. Thakur, H. Chaurasiya and H. Nagdev, "Automated Provisioning of Application in IAAS Cloud using Ansible Configuration Management," IEEE, pp. 81-85, 2015.

[7] "www.slideshare.net," 10 October 2013. [Online]. Available: https://www.slideshare.net/oodlestech/saas-application. [Accessed April 2020].

[8] “www.oneglobesystems.com” 04/03/2023.

Available: https://www.oneglobesystems.com/blog/automating-infrastructure-deployments-iaasin-aws-cloud-with-ansible

[9] “www.learn.microsoft.com” Available: https://learn.microsoft.com/enus/azure/developer/ansible/install-on-linux-vm?tabs=azure-cli

[10] https://www.simplilearn.com/tutorials/azure-tutorial/what-is-azure

[11] https://www.techtarget.com/searchaws/definition/Amazon-Web-Services

[12] Dan Sullivan, “Google Cloud Computing Services,” Wiley Data and Cybersecurity, 2019. Available: https://ieeexplore.ieee.org/document/9822263/authors#authors

[13] Chris Binnie, Rory McCune, “Cloud Auditing,” Wiley Data and Cybersecurity, 2021. Available: https://ieeexplore.ieee.org/document/9932291/authors#authors

[14] P. J. Mudialba, "The Impact of Cloud Technology on the Automation of Businesses," IEEE, 2016.

[15] E. Wibowo, "Cloud management and automation," IEEE, 2014.