**Azure-Firewall-Lab-Exp**

1. **Summary of Azure Infrastructure**
2. **Diagram**
3. **Azure Infrastructure with Firewall and Bastion Host Using Terraform**
4. **Check log flow**
5. **Summary of Azure Infrastructure**

**This Terraform-based solution provisions a secure and access-controlled Azure environment consisting of:**

**1. Core Networking**

* Virtual Network (VNet): A /16 address space (10.0.0.0/16) split into three subnets:
  + subnet-demo – for hosting the virtual machine
  + AzureFirewallSubnet – for the Azure Firewall (mandatory naming)
  + AzureBastionSubnet – for Bastion Host (mandatory naming)

**2. Compute**

* Linux Virtual Machine: Ubuntu 22.04 LTS VM with public IP and SSH access enabled via NSG

**3. Security**

* Network Security Group (NSG): Allows only port 22 (SSH) inbound for the VM
* Azure Firewall:
  + Deployed in its own subnet
  + Linked to a Firewall Policy with defined application and network rules
  + Route table redirects all outbound traffic from the VM through the firewall

**4. Access Control**

* Azure Bastion Host:
  + Enables secure, browser-based RDP/SSH access to the VM
  + Reduces the need for public IPs (although this deployment still includes one for demo/testing)

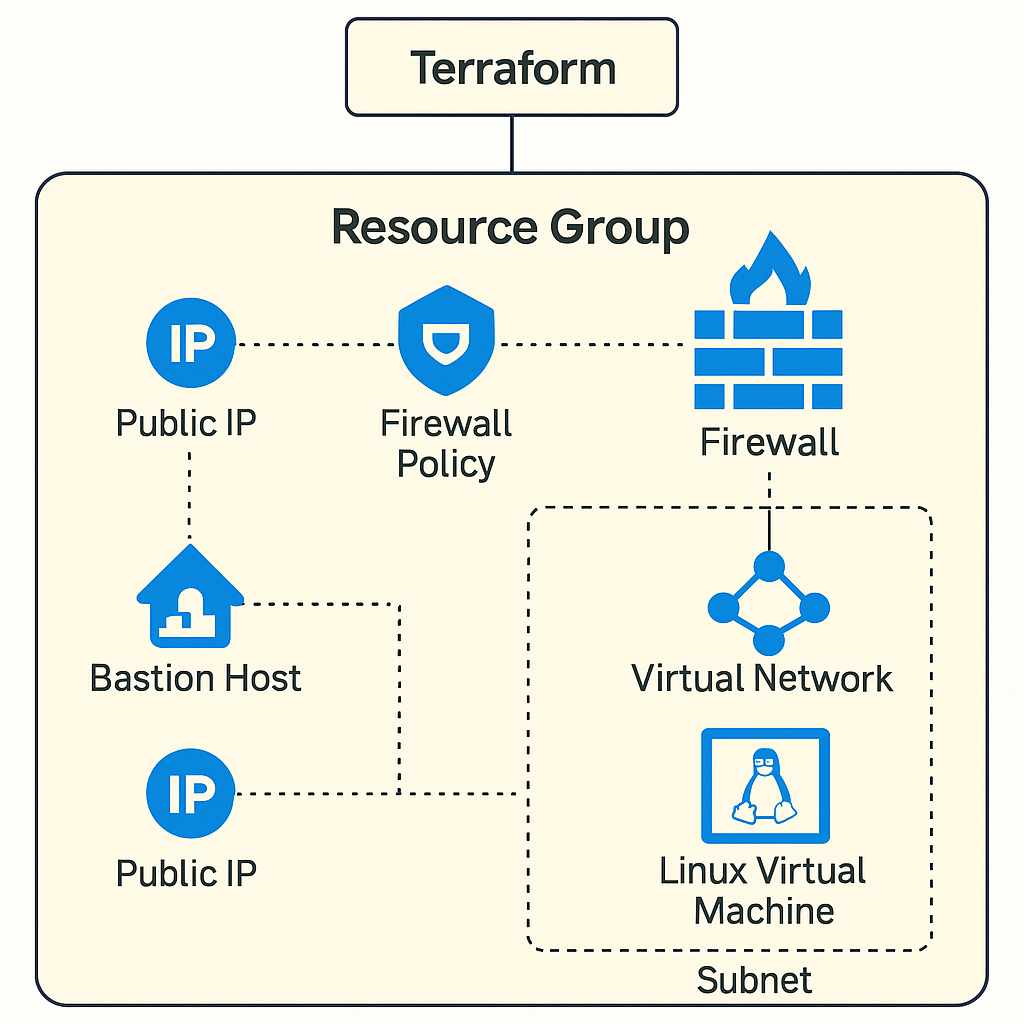
**5. Routing**

* Custom Route Table:
  + Associated with the VM subnet
  + Forces all outbound internet traffic (0.0.0.0/0) through the Azure Firewall using VirtualAppliance

**🛡️ Firewall Policy Highlights**

* Deny Rule: Blocks access to yahoo.com
* Allow Rule: Permits all other web traffic (HTTP/HTTPS)
* Network Rules: Allow DNS (TCP/UDP 53) and SSH (TCP 22)

1. **Diagram**



1. **Azure Infrastructure with Firewall and Bastion Host Using Terraform**

This document provides a comprehensive explanation of deploying a secure Azure infrastructure using Terraform. It includes a Linux virtual machine, Azure Firewall with a defined policy, and secure access through Azure Bastion Host. Each section is broken down with explanations and associated Terraform code.

**1. Provider Configuration**

provider "azurerm" {

features {}

subscription\_id = "<your-subscription-id>"

}

**Explanation**: This sets up the Azure provider. The features {} block is required but can be empty. The subscription\_id specifies which Azure subscription to deploy resources into.

**2. Resource Group**

resource "azurerm\_resource\_group" "main" {

name = "rg-firewall-demo"

location = "East US"

}

**Explanation**: Defines a logical container to hold all related Azure resources.

**3. Virtual Network and Subnets**

resource "azurerm\_virtual\_network" "main" {

name = "vnet-demo"

address\_space = ["10.0.0.0/16"]

location = azurerm\_resource\_group.main.location

resource\_group\_name = azurerm\_resource\_group.main.name

}

**Explanation**: Creates a virtual network (vnet-demo) with an address space of 10.0.0.0/16. This network will host all subnets.

**Subnets:**

resource "azurerm\_subnet" "main" {...} # For VM

resource "azurerm\_subnet" "firewall\_subnet" {...} # Required name: AzureFirewallSubnet

resource "azurerm\_subnet" "bastion\_subnet" {...} # Required name: AzureBastionSubnet

**Explanation**:

* main: Hosts the VM
* firewall\_subnet: Must be named AzureFirewallSubnet
* bastion\_subnet: Must be named AzureBastionSubnet

**4. Public IPs**

Each resource (Firewall, VM, Bastion) requires its own public IP.

resource "azurerm\_public\_ip" "firewall\_pip" {...}

resource "azurerm\_public\_ip" "vm\_public\_ip" {...}

resource "azurerm\_public\_ip" "bastion\_pip" {...}

**Explanation**: Static IPs are used to ensure consistency, and Bastion/Firewall require the Standard SKU.

**5. Network Interface and NSG**

resource "azurerm\_network\_interface" "vm\_nic" {...}

resource "azurerm\_network\_security\_group" "vm\_nsg" {...}

resource "azurerm\_network\_interface\_security\_group\_association" "nic\_nsg\_assoc" {...}

**Explanation**:

* vm\_nic: Attaches the VM to the subnet and assigns a public IP.
* vm\_nsg: Defines inbound rules (SSH allowed).
* nic\_nsg\_assoc: Binds the NSG to the NIC.

**6. Linux Virtual Machine**

resource "azurerm\_linux\_virtual\_machine" "ubuntu\_vm" {...}

**Explanation**:

* Provisions a basic Ubuntu VM with password-based authentication.
* Uses Standard\_B1s for cost-efficiency.
* Uses the Canonical image family (22.04 LTS).

**7. Firewall and Policy**

**Firewall Policy:**

resource "azurerm\_firewall\_policy" "main" {...}

**Explanation**: Creates a reusable policy for the Azure Firewall.

**Rule Collection Group:**

resource "azurerm\_firewall\_policy\_rule\_collection\_group" "example" {...}

**Explanation**:

* Denies access to yahoo.com.
* Allows access to all other websites.
* Permits DNS and SSH traffic.

**Azure Firewall:**

resource "azurerm\_firewall" "main" {...}

**Explanation**:

* Deploys the Azure Firewall into the required subnet.
* Applies the policy using firewall\_policy\_id.

**8. Route Table**

resource "azurerm\_route\_table" "main" {...}

resource "azurerm\_subnet\_route\_table\_association" "main" {...}

**Explanation**:

* Directs all internet-bound traffic (0.0.0.0/0) from the VM subnet to go through the Azure Firewall using VirtualAppliance as the next hop type.

**9. Azure Bastion Host**

resource "azurerm\_bastion\_host" "main" {...}

**Explanation**:

* Provides secure RDP/SSH access to VMs over the Azure Portal.
* Eliminates need for a public IP on the VM (though one is included here for demo).

**Diagram**

1. **Check Traffic Flow**

**✅ 1. Azure Firewall Flow Logs**

**🔍 What You Can Monitor**

* **Application rule hits** (like Deny yahoo.com)
* **Network rule hits** (e.g., DNS, SSH)
* **Threat intelligence logs**
* **SNAT traffic logs**

**🧰 How to Enable Firewall Logs**

1. Create a **Log Analytics Workspace**
2. Enable **Diagnostic Settings** on the Azure Firewall:
   * Categories:
     + AzureFirewallApplicationRule
     + AzureFirewallNetworkRule
     + AzureFirewallDnsProxy
     + AzureFirewallThreatIntel
   * Send to Log Analytics or Storage Account

**🚦 Traffic Generation Required?**

**Yes** – logs are only generated if traffic **hits the firewall**. You can:

* SSH into the VM (generates allow-ssh)
* Use curl to http://yahoo.com to trigger the **deny** rule
* Use dig or nslookup for DNS logs:

bash

CopyEdit

dig google.com

curl http://example.com

**✅ 2. Bastion Flow Logs**

**🔍 What You Can Monitor**

* Connection logs (who connected, to which VM, and when)
* Audit events in **Azure Activity Log**

**🧰 How to Enable Bastion Logs**

* Bastion supports **diagnostic settings**, including:
  + BastionAuditLogs
* Enable via Portal or Terraform, send to Log Analytics

**🚦 Traffic Generation Required?**

**Yes** – open a **Bastion session** to the VM to see activity.

**✅ 3. Network Flow Logs (NSG Flow Logs)**

**🔍 What You Can Monitor**

* Inbound/outbound flows for NICs/subnets with NSGs
* Allowed vs denied traffic
* Source/destination IPs, ports, protocol, action

**🧰 How to Enable NSG Flow Logs**

1. Go to the NSG (e.g., vm\_nsg)
2. Enable **Flow Logs** (requires a **Storage Account** and optionally Log Analytics)
3. Choose **Version 2** for readable output

**🚦 Traffic Generation Required?**

**Yes** – generate SSH or ICMP traffic to/from VM

**📊 Optional: Use Traffic Analytics**

You can plug all this into **Traffic Analytics** in Azure Monitor for visual flow graphs and dashboards.