# Project:
# Securing the Perimeter

**JUBRIL EDUN:**

*25th March, 2024*

# Project Overview:

## This project focuses on four key concepts:

1. Designing a secure network architecture
2. Building a secure network architecture in azure
3. Monitoring with a SIEM
4. Zero Trust Model

Section 1

# Designing a Secure Network Architecture

# Section 1: Designing the Network

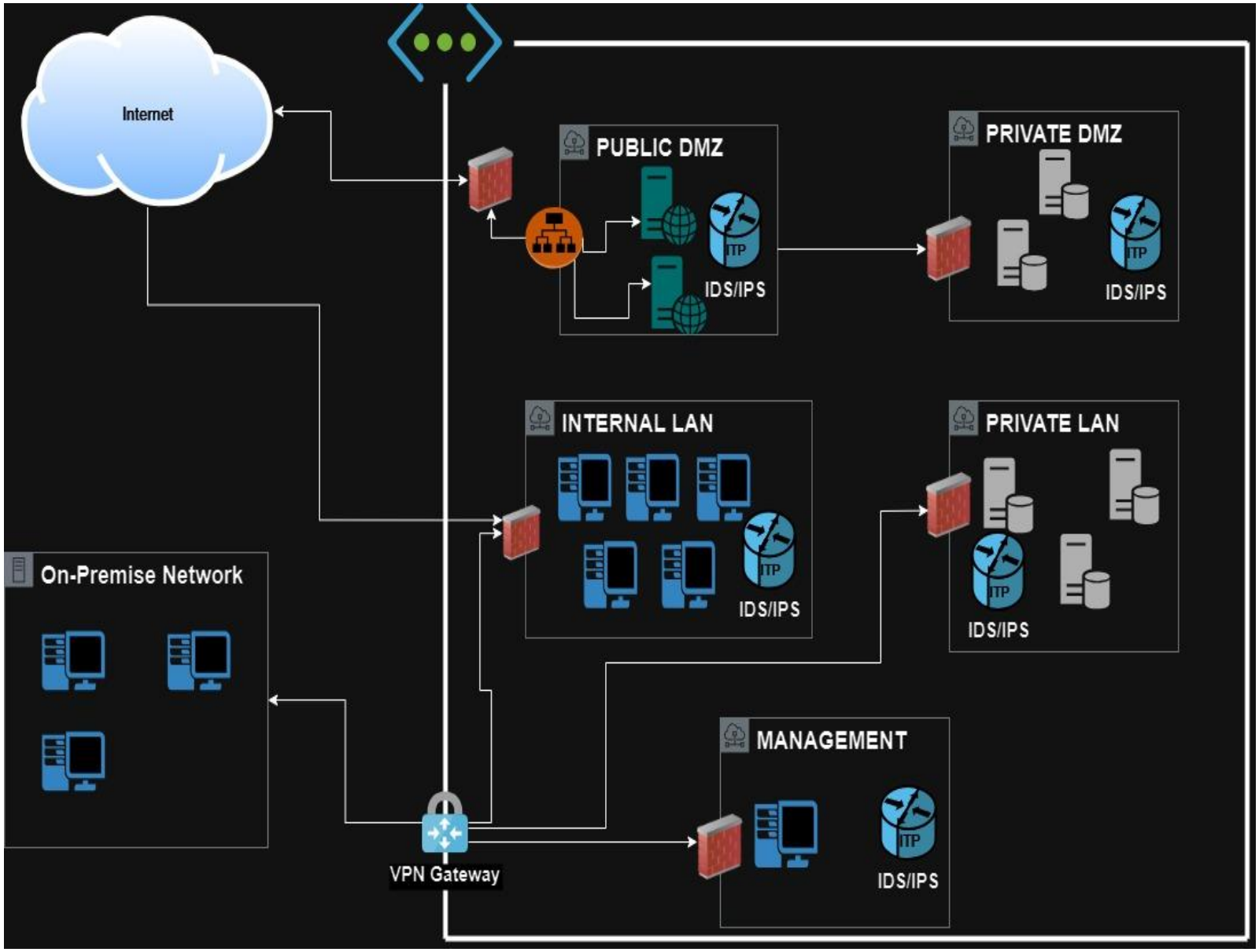
**XYZ's secure network design requirements:**

1) An on-premise network that has 3 workstations in it.

2) A Virtual Network with the following segments:

- Public DMZ with two web servers and a load balancer in it.
- Private DMZ with two database servers.
- Management LAN with one management server in it.
- Internal LAN with 5 workstations in it.
- Private Secure LAN with 3 database servers.
- A VPN gateway connecting the on-premise network to your Virtual Network.
- Placement of security devices in the architecture, including load balancer(s), firewall(s), IDS/IPS device(s).

# 1.1 Network Design



Section 2

# Building a Secure Network Architecture in Azure

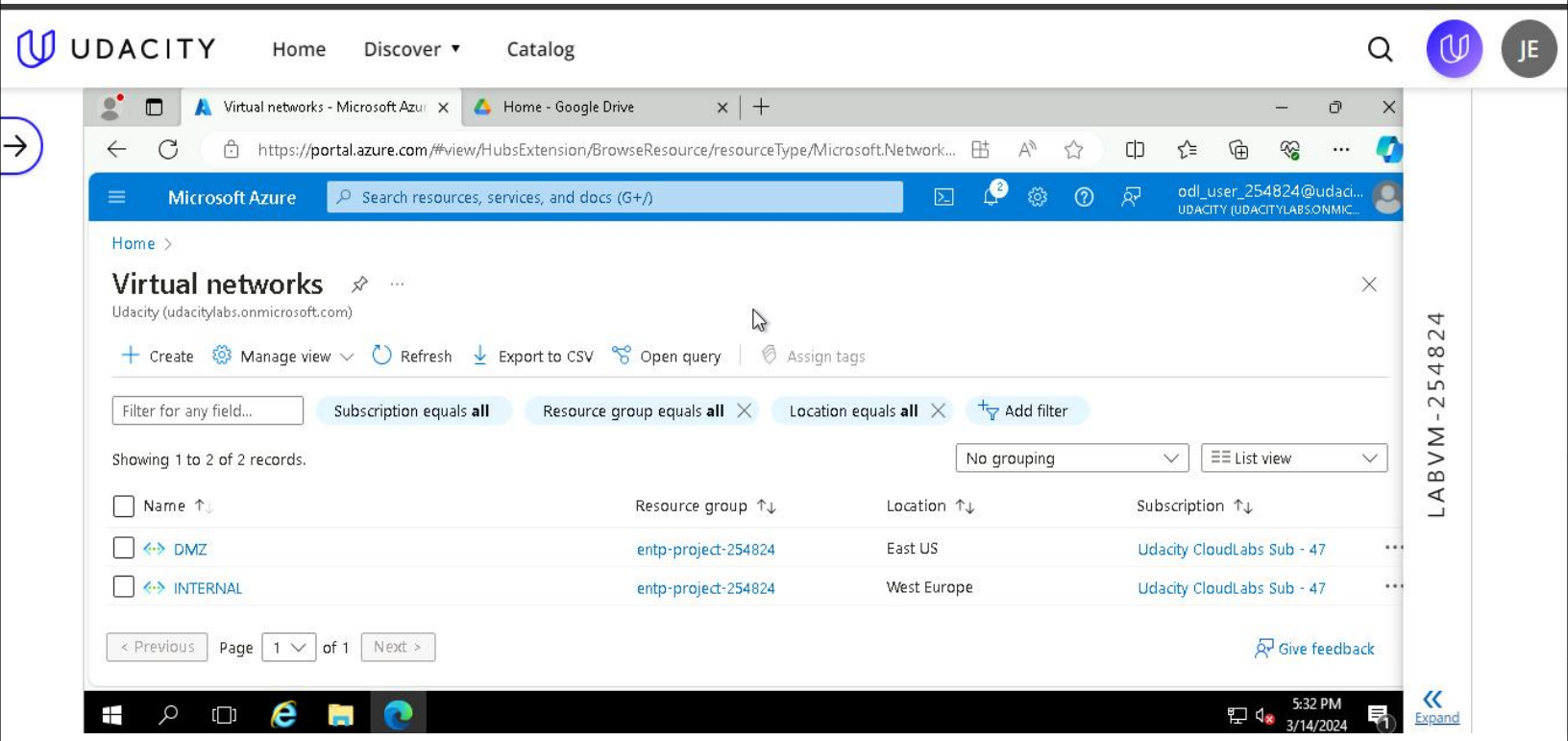
# Section 2: Building the Network

**After designing the network architecture, the design was presented to XYZ's stakeholders. They're all on board with the design, and have given the green light to start building the architecture out in Azure.**

**Screenshots shown in next slides:**

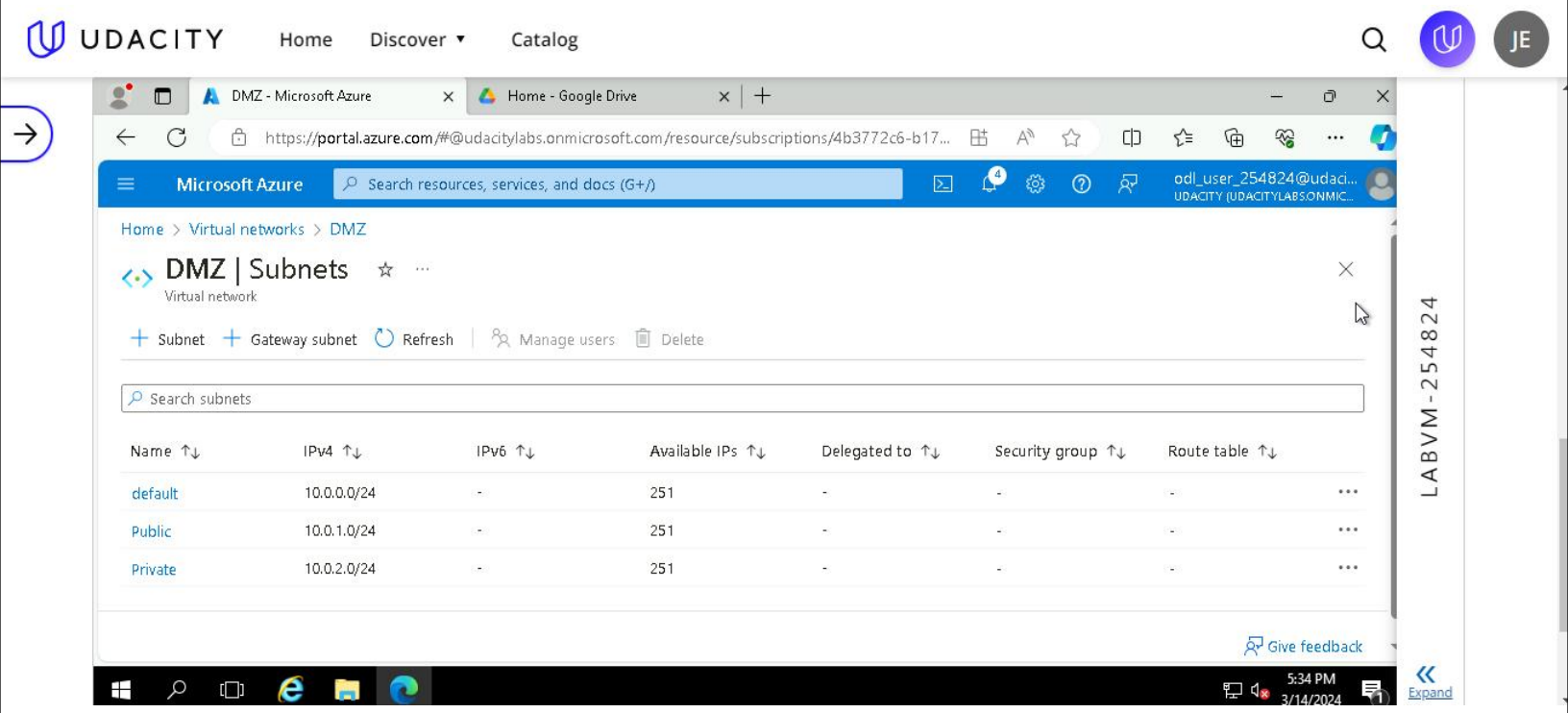
# 2.1.1 VNETs screenshot

**Two Azure Virtual Networks (DMZ and Internal network) were created in the resource group ‘entp-project’.**



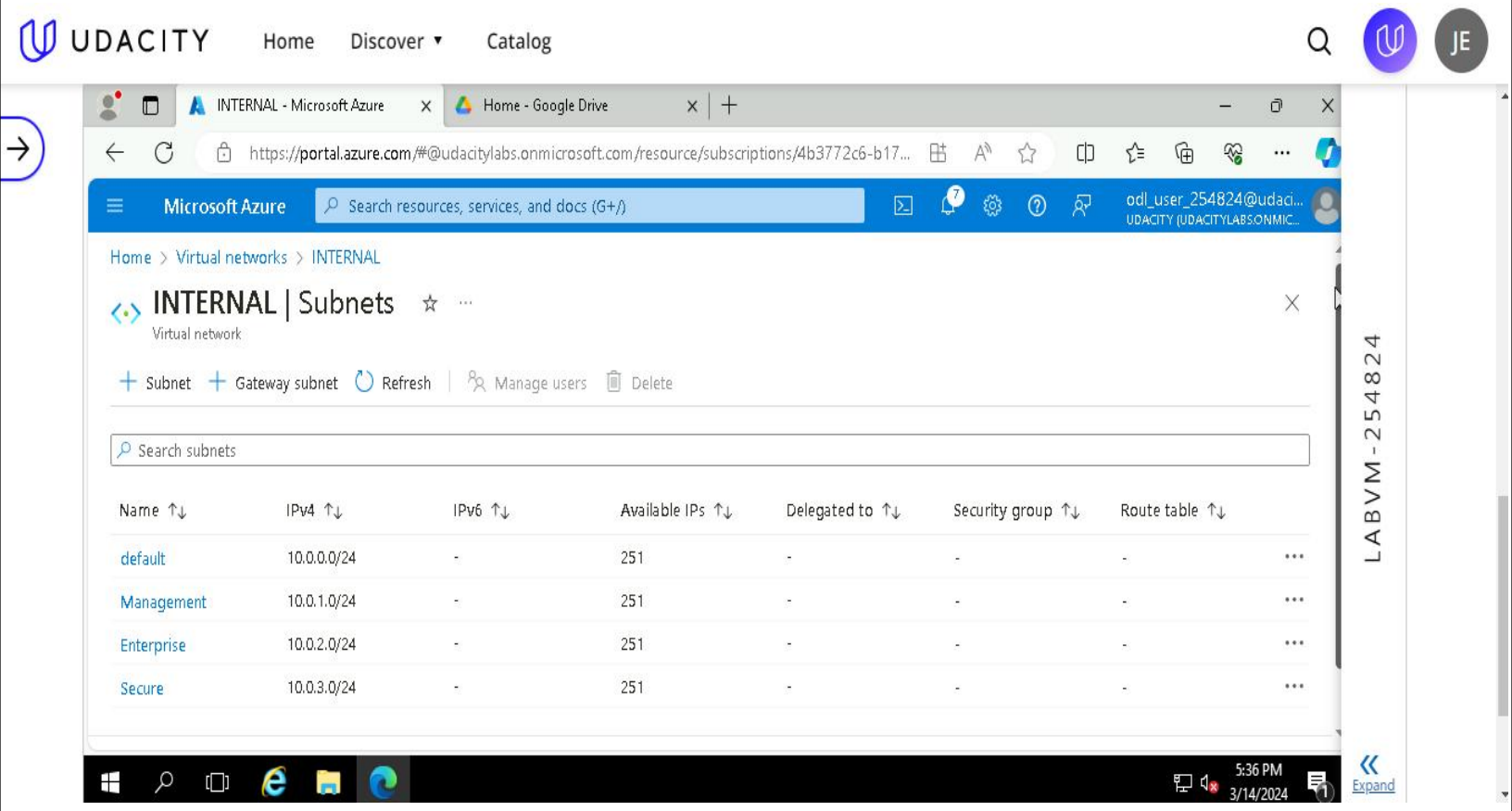
# 2.1.2 DMZ subnets screenshot

**Two subnets (Public and Private) within the DMZ VNet were created.**



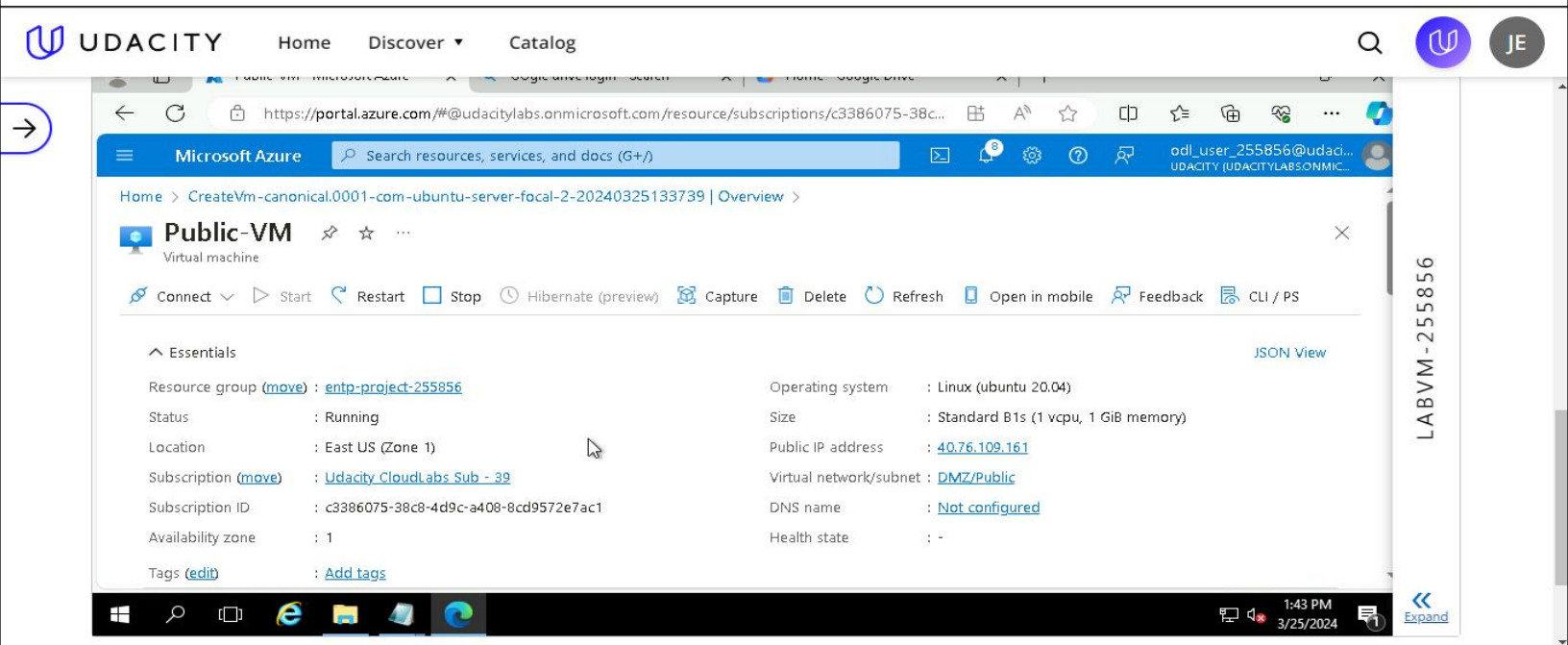
# 2.1.3 Internal subnets screenshot

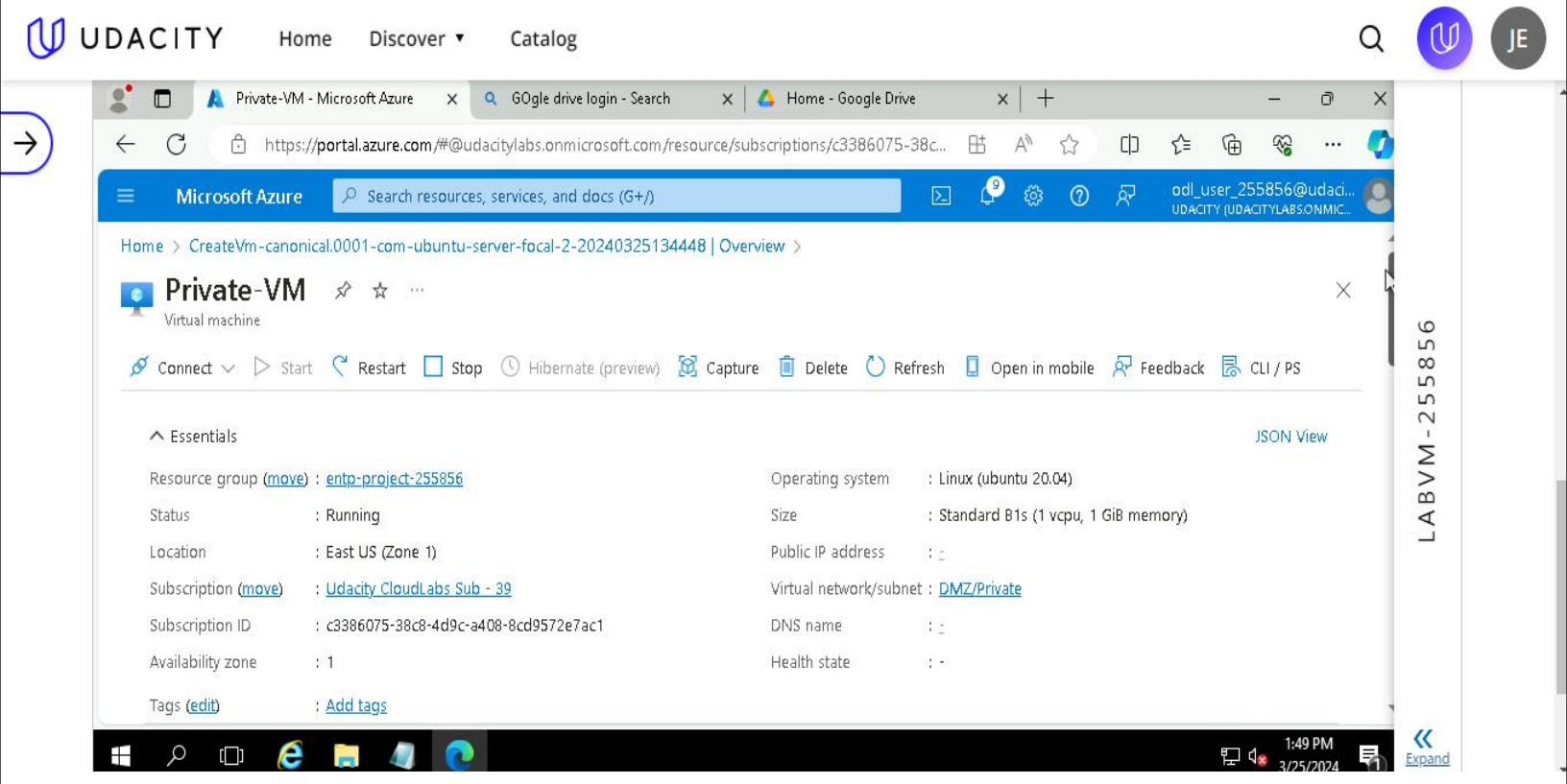
**Three subnets in the internal network (Management, Secure, and Enterprise) were created.**



# 2.2.1 VMs Screenshot

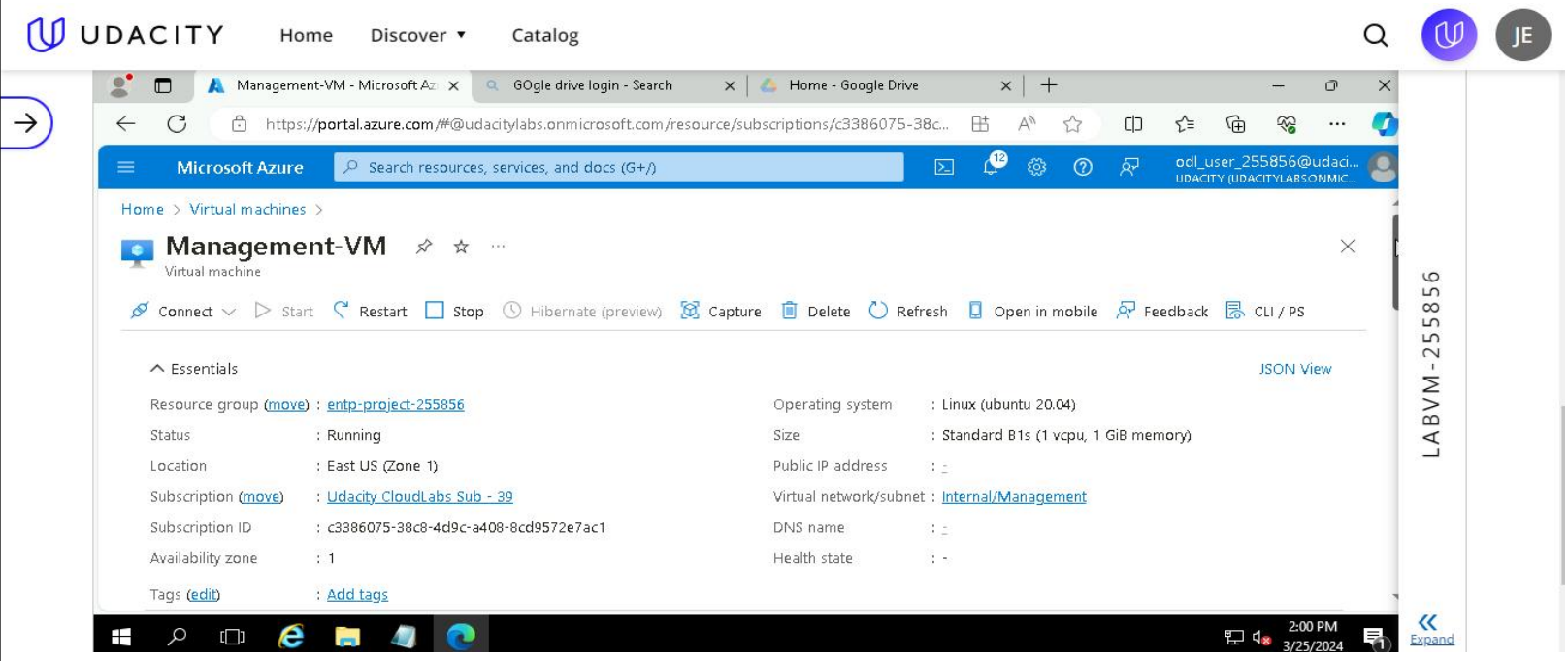
**One VM was created in each of the public and private DMZ subnets.**

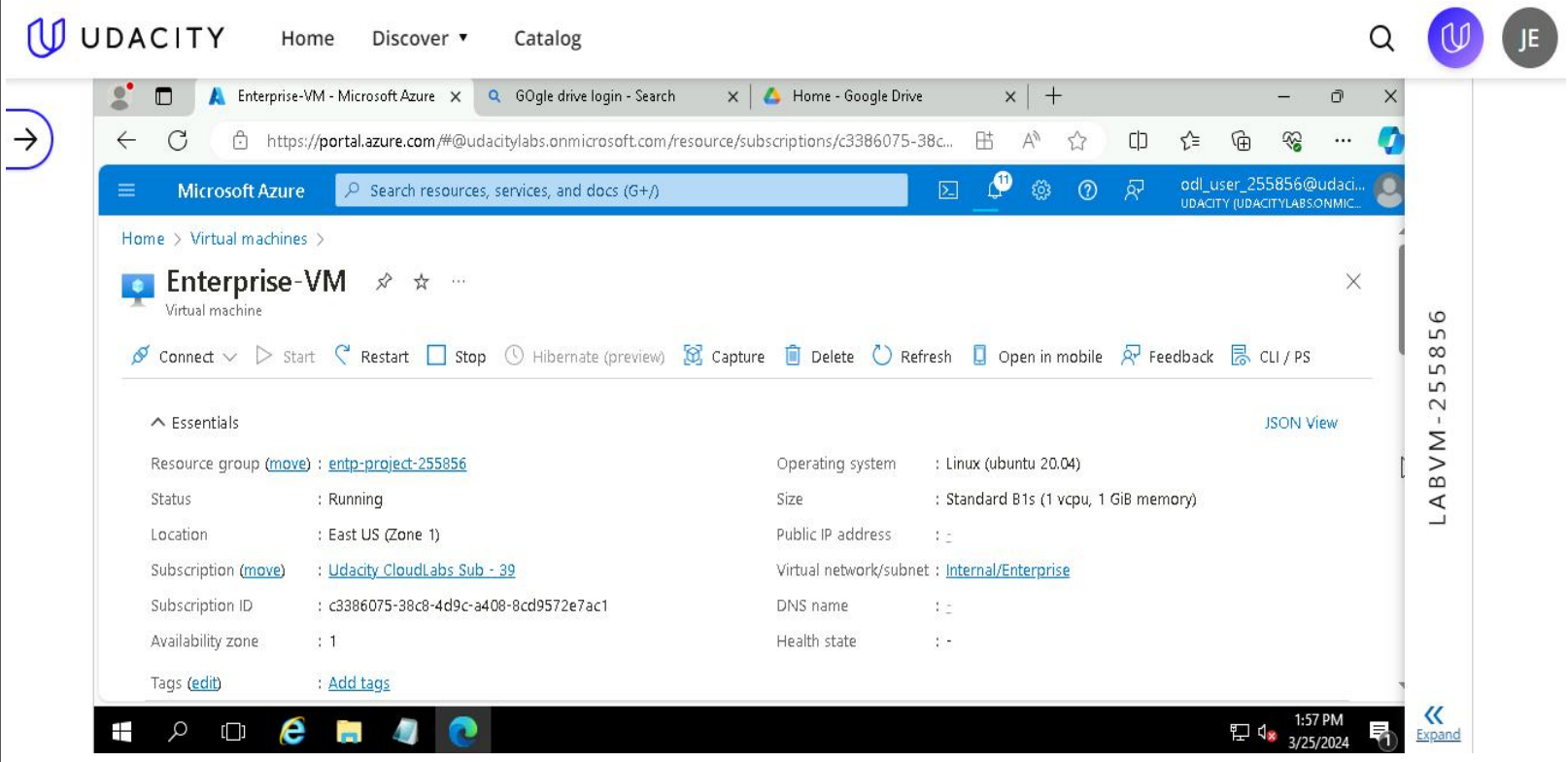


# 2.2.2 VMs Screenshot

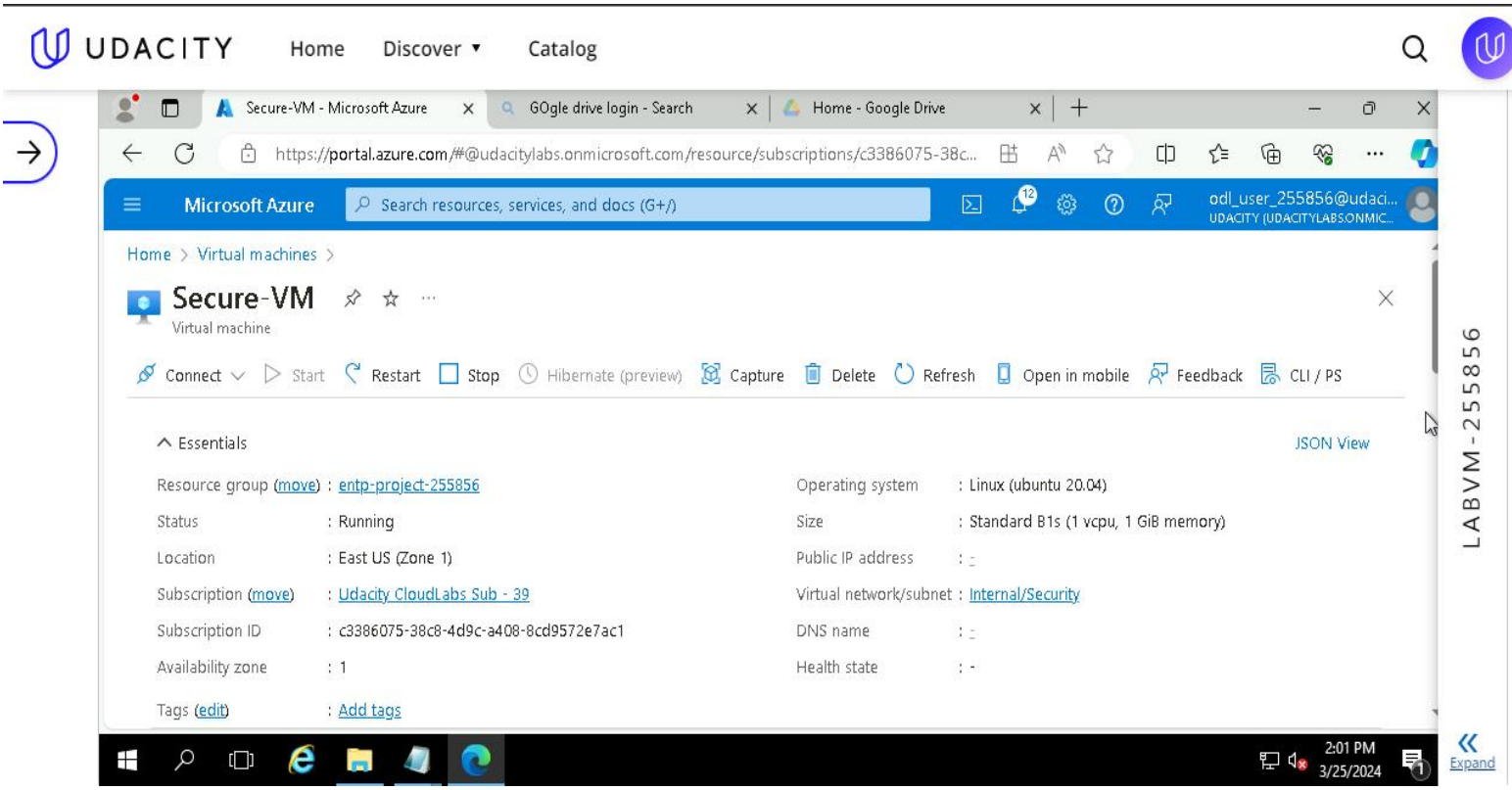
**One VM was created in each of the Management, Secure, and Enterprise internal subnets.**



# 2.2.2 VMs Screenshot

**One VM was created in each of the Management, Secure, and Enterprise internal subnets.**



# 2.3 Secure Routing

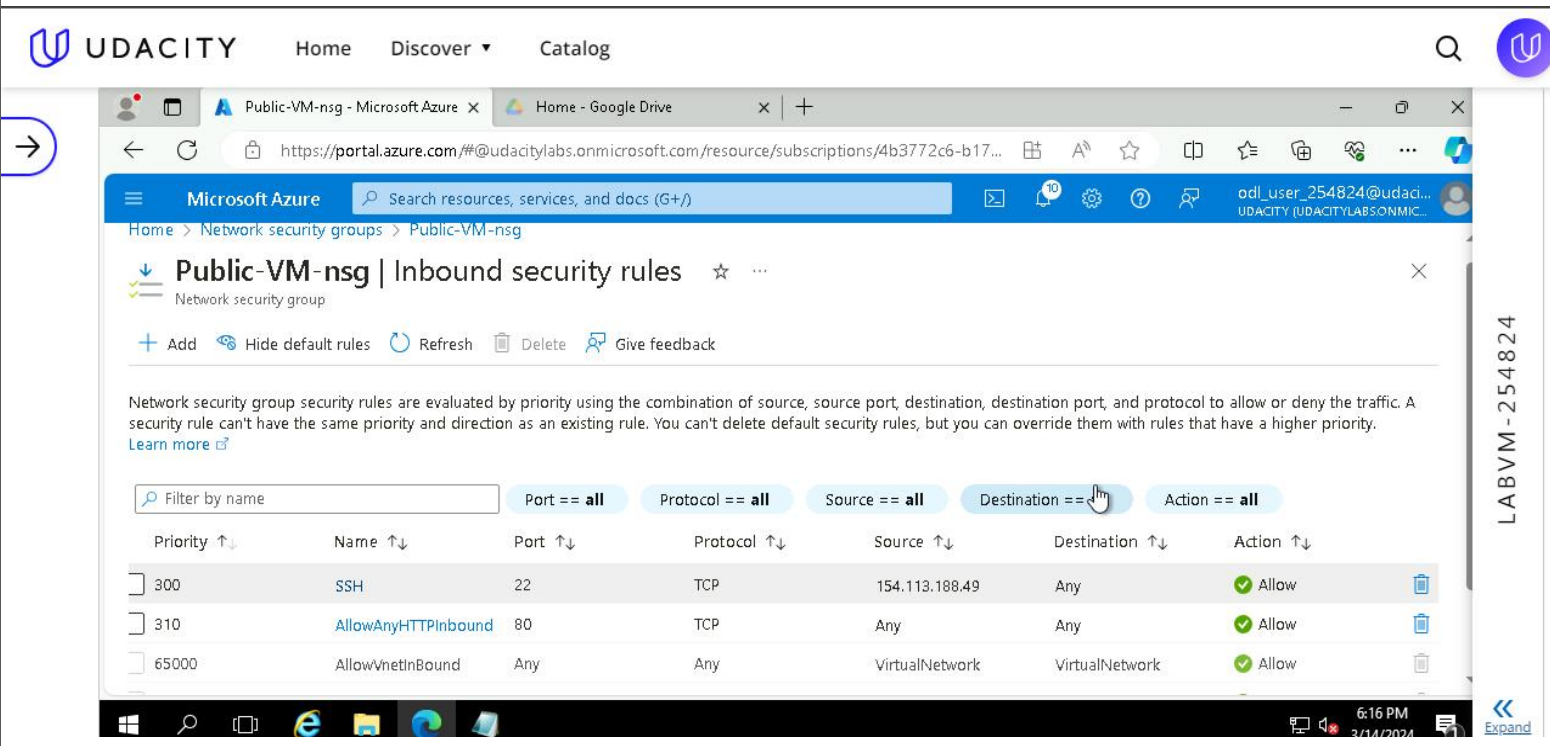
**Secure routing was configured, by creating network traffic rules, within the Virtual Network and Subnets following secure best practices.**

**Screenshots shown in next slides**

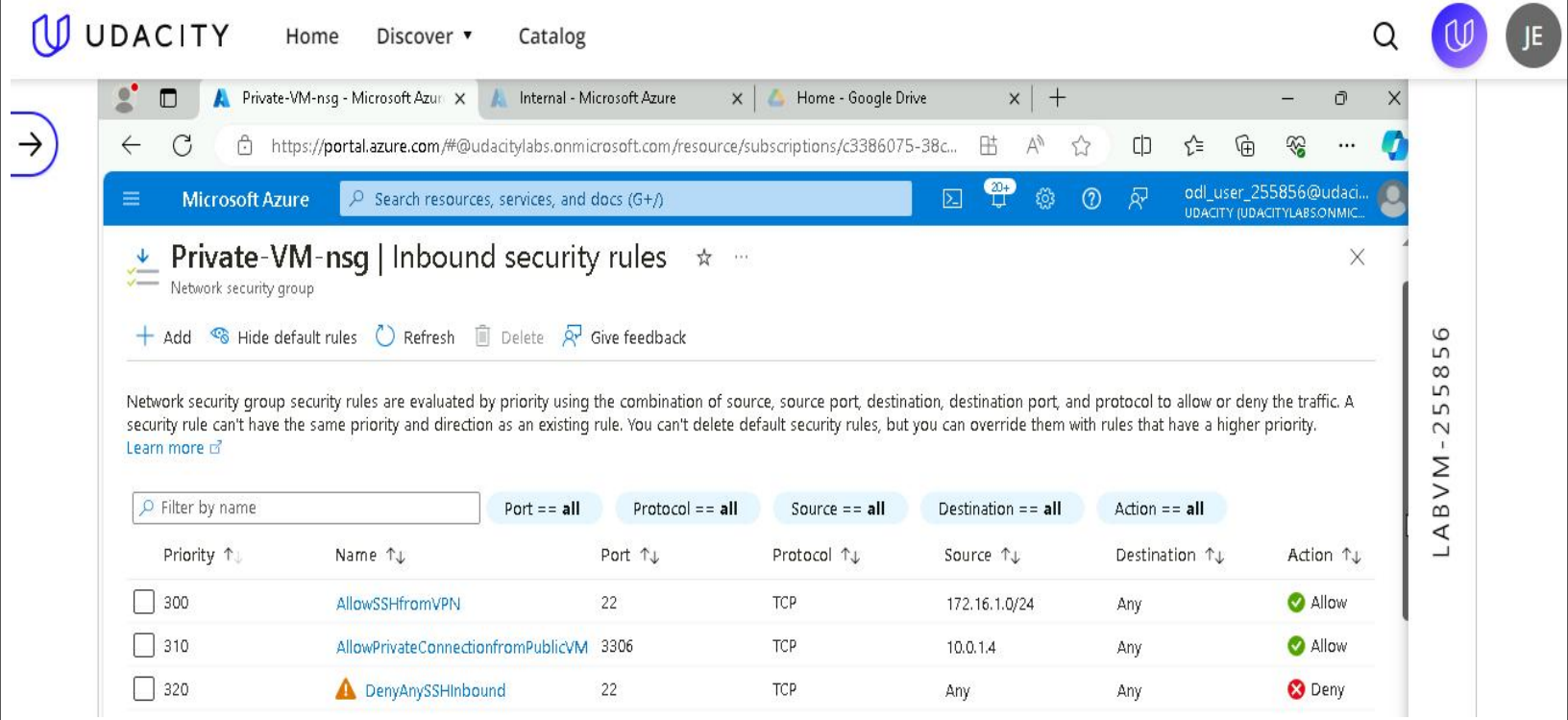
# 2.3.1 Screenshot

## Traffic rules in DMZ.

### Traffic rule for the Public Web Server

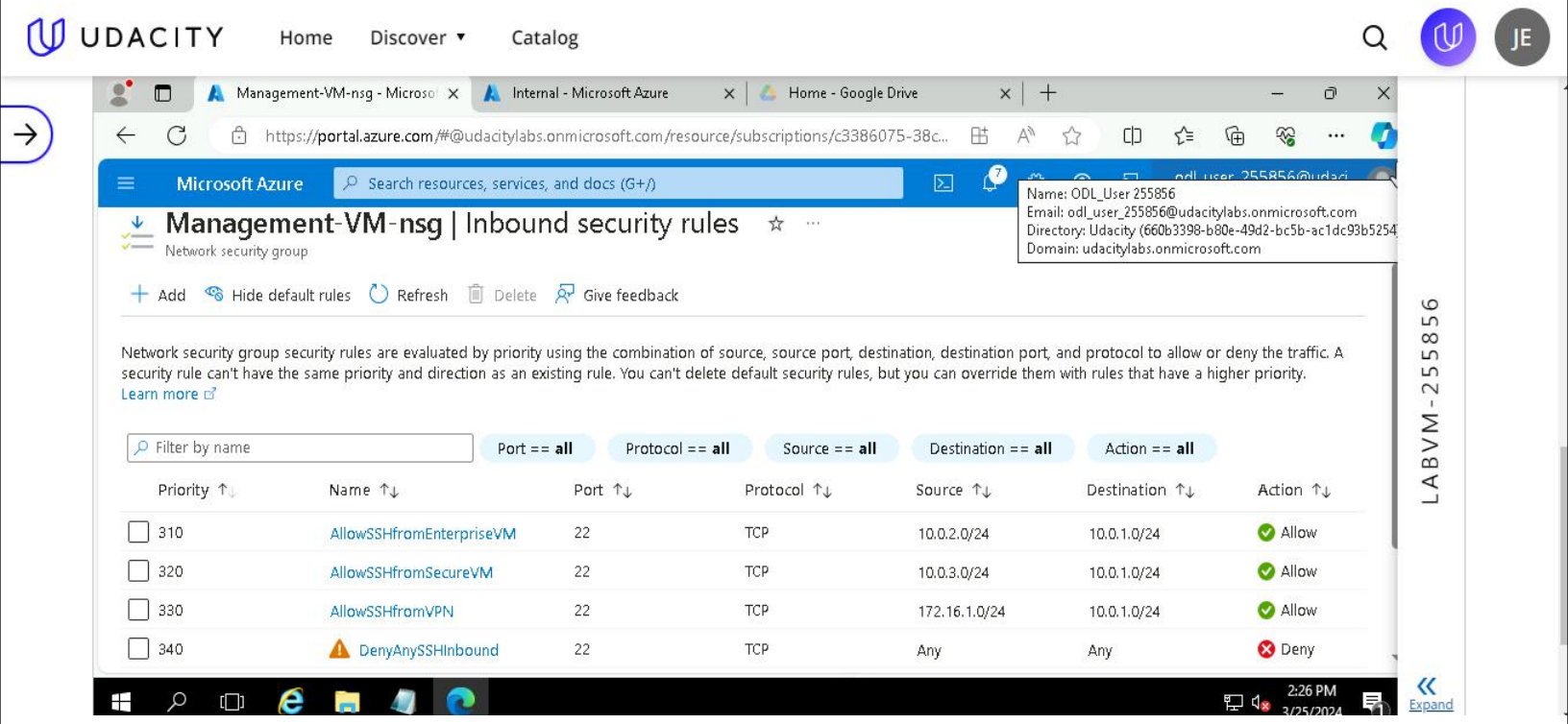


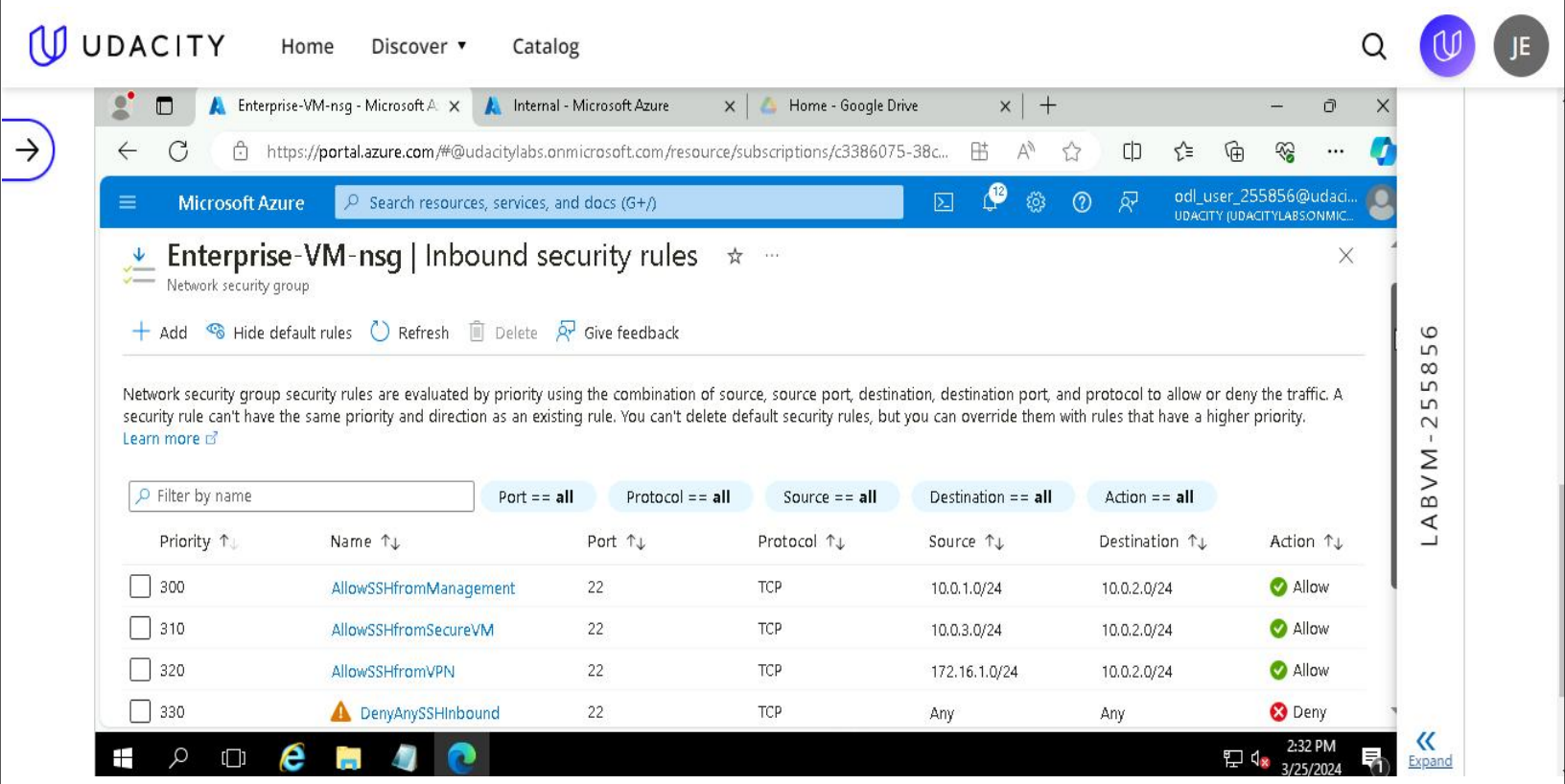

### Traffic rule for the Private Database Server



# 2.3.2 Screenshot

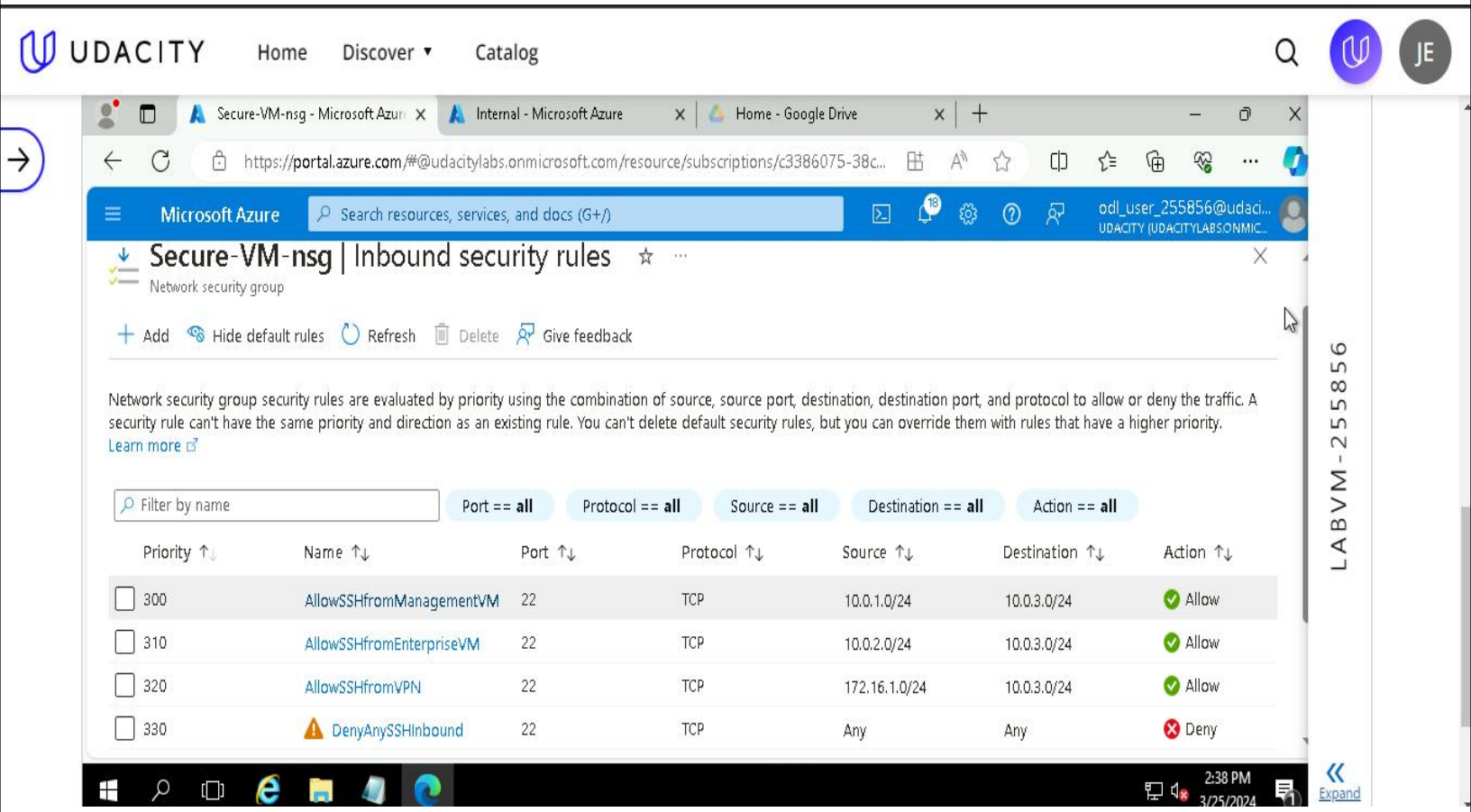
## Traffic rules in Internal network.



# 2.3.2 Screenshot

## Traffic rules in Internal network.



# 2.4 VPN Access

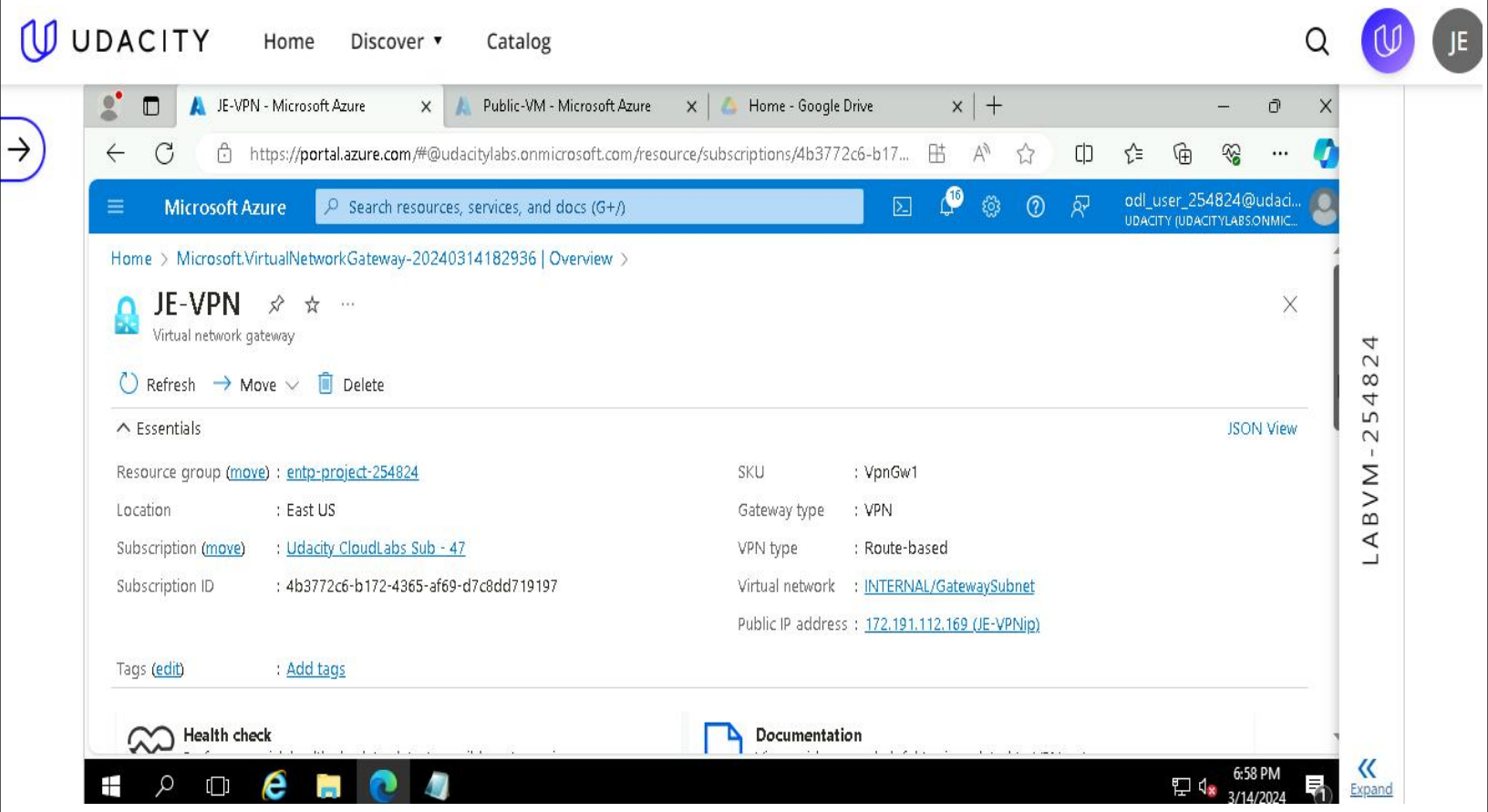
**A VPN to secure access to the internal network was created.**

**After creating the VPN, the VPN connection was tested by attempting to connect to one of the VMs in the internal network.**

**Screenshots shown in next slides:**

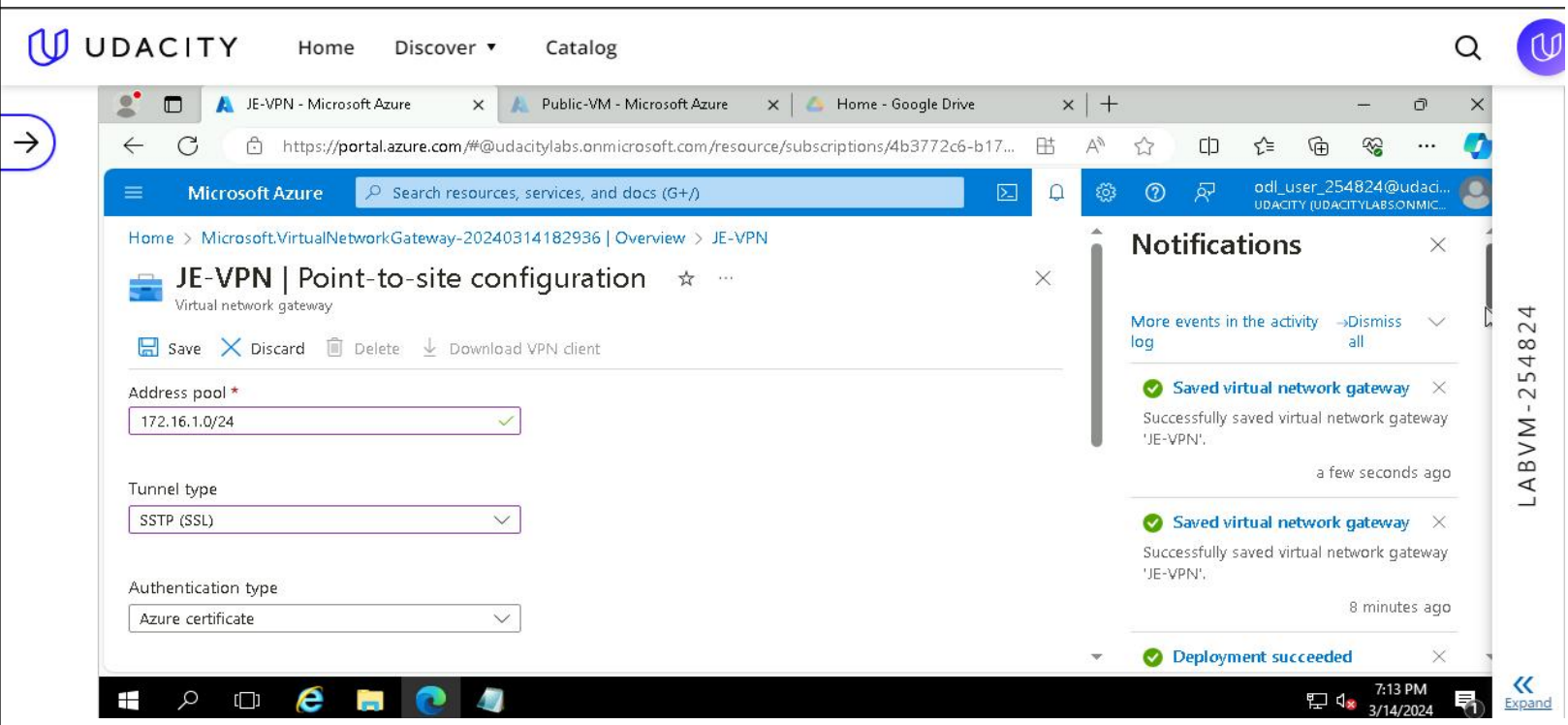
# 2.4.1 VPN screenshot

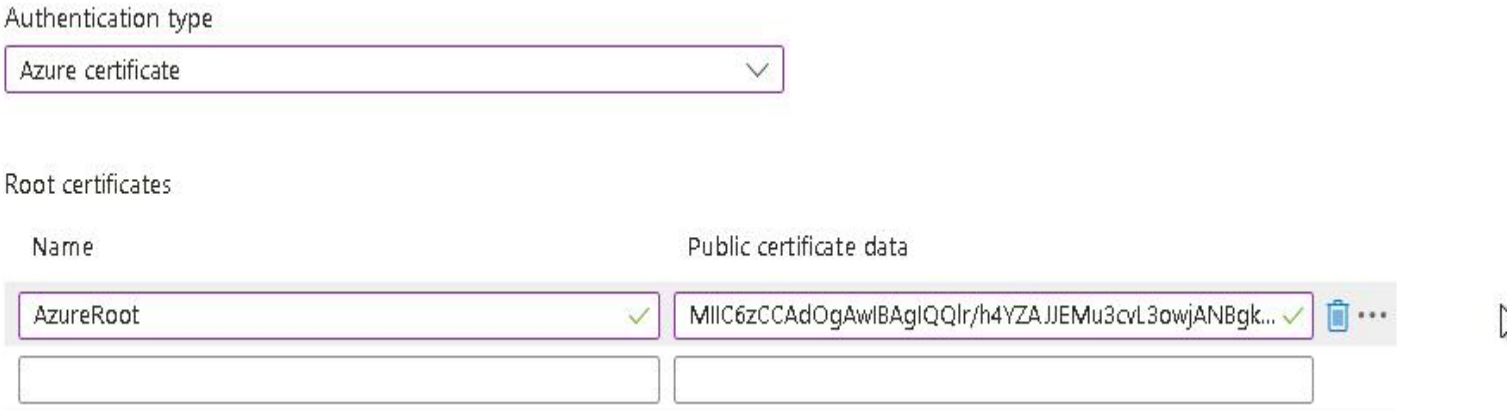
**A Virtual Network Gateway was created for VPN connection to the internal network.**



# 2.4.1 Point-to-Site screenshot

## A Point-to-Site connection was configured on the created Virtual Network Gateway

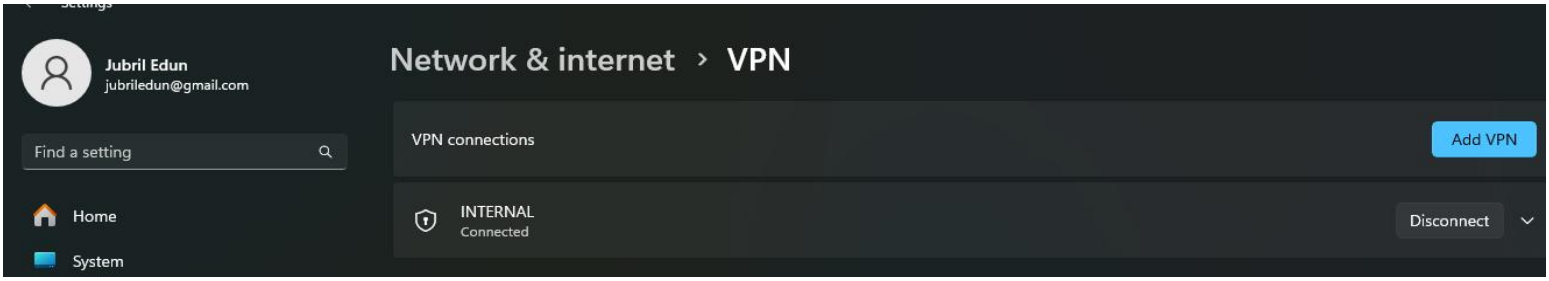


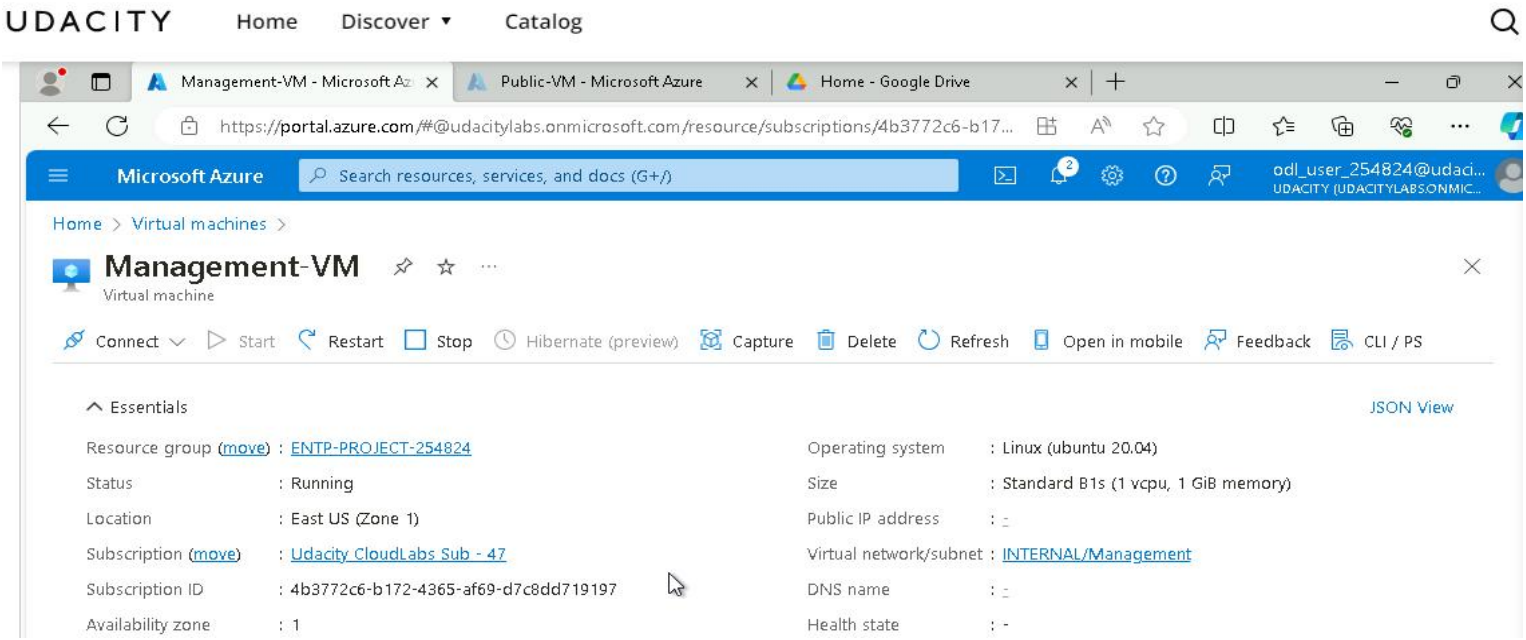
# 2.4.2 VPN test screenshot

## Testing VPN connection

### Successful connection to the VPN gateway from PC




### Management VM with no Public IP



# 2.4.2 VPN test screenshot

## Testing VPN connection

### Routing rule allowing connection from the VPN Gateway Subnet

Filter by name | Port == all | Protocol == all | Source == all | Destination == all | Action == all

| Priority | Name | Port | Protocol | Source | Destination | Action |
|---|---|---|---|---|---|---|
| Inbound Security Rules | | | | | | |
| 210 | AllowInboundtrafficov... | 22 | TCP | 172.16.1.0/24 | Any | Allow |
| 65000 | AllowVnetInBound | Any | Any | VirtualNetwork | VirtualNetwork | Allow |
| 65001 | AllowAzureLoadBalanc... | Any | Any | AzureLoadBalancer | Any | Allow |
| 65500 | DenyAllInBound | Any | Any | Any | Any | Deny |
| Outbound Security Rules | | | | | | |

### Successful SSH login to the ManagementVM using the Private IP

```
azureuser@Management-VM: ~
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>ssh azureuser@10.0.1.4
The authenticity of host '10.0.1.4 (10.0.1.4)' can't be established.
ED25519 key fingerprint is SHA256:gNeYp/9BYTAWmTPwveqmX4tr3k0WEOjdy2M84/uQRPQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.1.4' (ED25519) to the list of known hosts.
Enter passphrase for key 'C:\Users\hp/.ssh/id_rsa':
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1057-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of Thu Mar 14 19:33:31 UTC 2024

  System load:  0.0               Processes:             101
  Usage of /:   5.2% of 28.89GB   Users logged in:       0
  Memory usage: 31%               IPv4 address for eth0: 10.0.1.4
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

22 updates can be applied immediately.
18 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
azureuser@Management-VM:~$ echo "I was able to successfully a VPN conection to the Management VM in a private subnet"
I was able to successfully a VPN conection to the Management VM in a private subnet
```

## Section 3

# Continuous Monitoring with a SIEM

# Section 3: Build the SIEM

**After buiding the secure network architecture, a SIEM solution (ELK) to monitor the enterprise network and alert about potential attacks was set up.**

**Screenshots shown in next slides:**

# 3.1.1 Screenshot

## A VM was created in the private DMZ and the ELK Server was configured on it.

### Configuring the ELK Server

```
azureuser@Private-VM:~$ sudo apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
```

```
Reading state information... Done
13 packages can be upgraded. Run 'apt list --upgradable' to see them.
azureuser@Private-VM:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc ubuntu-fan
Suggested packages:
```

```
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.3) ...
Processing triggers for libc-bin (2.31-0ubuntu9.14) ...
azureuser@Private-VM:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-9 dpkg-dev fakeroot g++ g++-9 gcc gcc-9 gcc-9-base libalgorithm-diff-perl
```

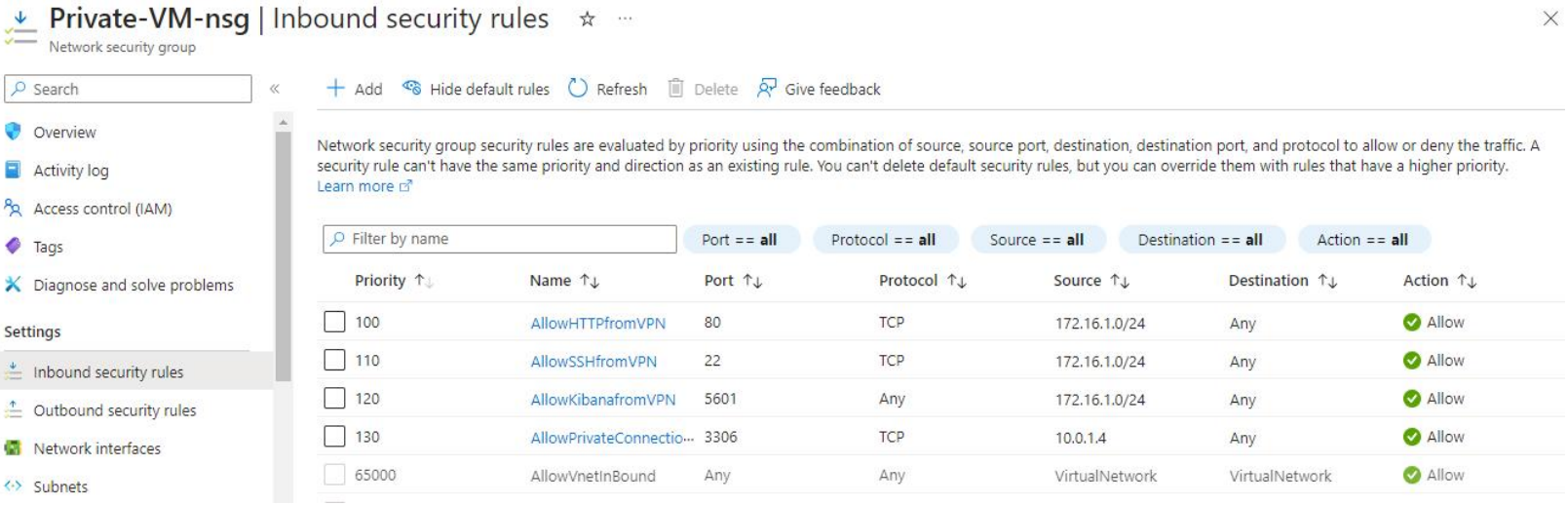
```
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.3) ...
Processing triggers for libc-bin (2.31-0ubuntu9.14) ...
azureuser@Private-VM:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp cpp-9 dpkg-dev fakeroot g++ g++-9 gcc gcc-9 gcc-9-base libalgorithm-diff-perl
```

```
    Not uninstalling requests at /usr/lib/python3/dist-packages, outside environment /usr
    Can't uninstall 'requests'. No files were found to uninstall.
Successfully installed charset-normalizer-3.3.2 docker-7.0.0 packaging-24.0 requests-2.31.0 urllib3-2.2.1
azureuser@Private-VM:~$ sudo sysctl -w vm.max_map_count=262144
vm.max_map_count = 262144
azureuser@Private-VM:~$ sudo docker pull sebp/elk:761
761: Pulling from sebp/elk
c64513b74145: Downloading [=====================>                             ]  15.02MB/31.66MB
```

```
Status: Downloaded newer image for sebp/elk:761
docker.io/sebp/elk:761
azureuser@Private-VM:~$ sudo docker run -p 5601:5601 -p 9200:9200 -p 5044:5044 -it --name elk sebp/elk:761
 * Starting periodic command scheduler cron                                                                    [ OK ]
 * Starting Elasticsearch Server
future versions of Elasticsearch will require Java 11; your Java version from [/usr/lib/jvm/java-8-openjdk-amd64/jre] does not meet this requirement
Exception in thread "main" java.lang.RuntimeException: starting java failed with [1]
output:
```

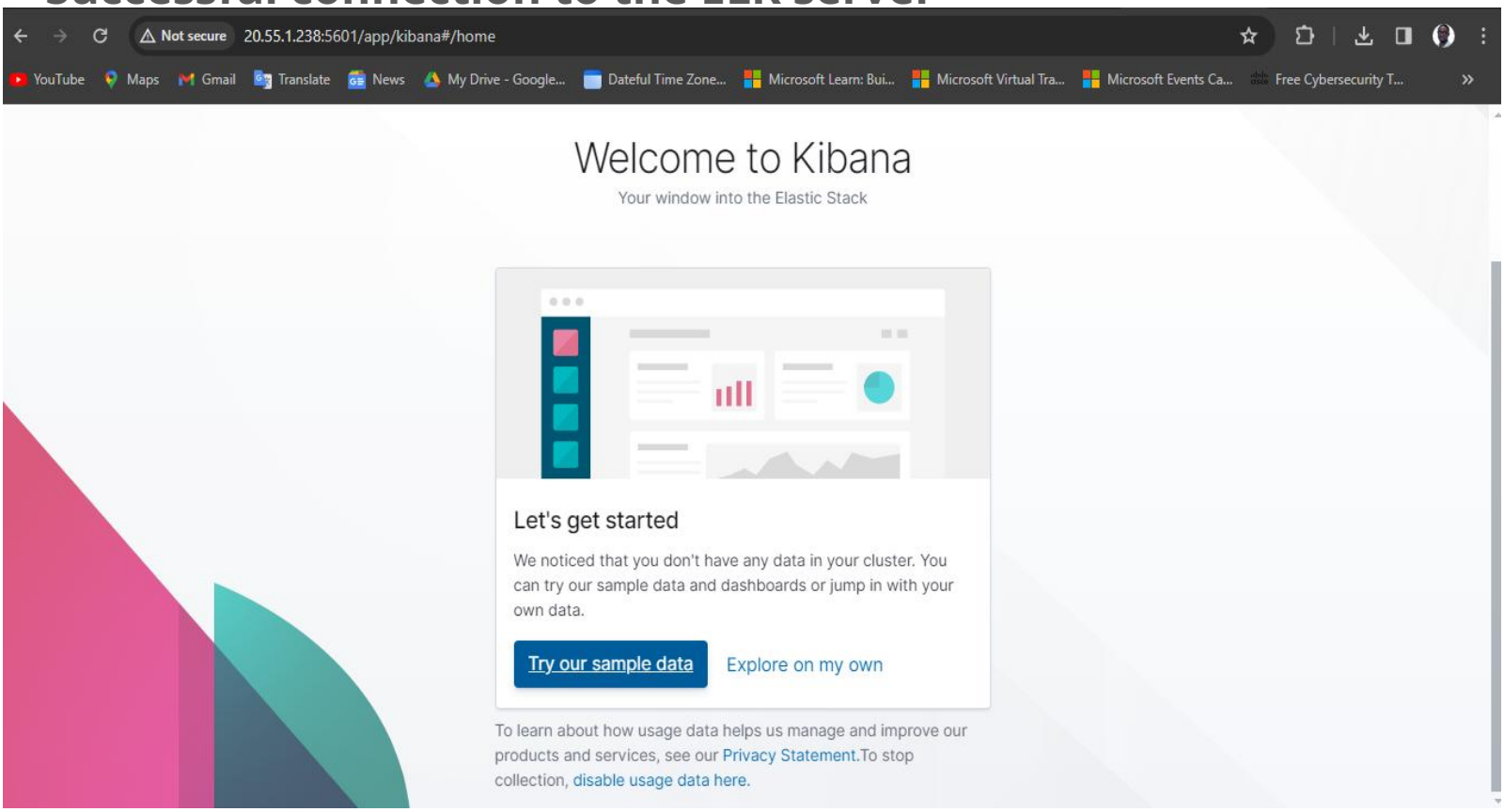
# 3.1.2 Screenshot

## Routing was setup to allow only traffic inbound to the server



# 3.1.2 ELK Server Screenshot,

## Successful connection to the ELK server



# 3.2 Ingest Logs

**In this next section, Filebeat was installed on the Web server to serve as ingest source for the ELK server. Filebeat will send logs from the Web server to the ELK server.**

**Screenshots shown in next slides:**

# 3.2.1 Filebeat Screenshot

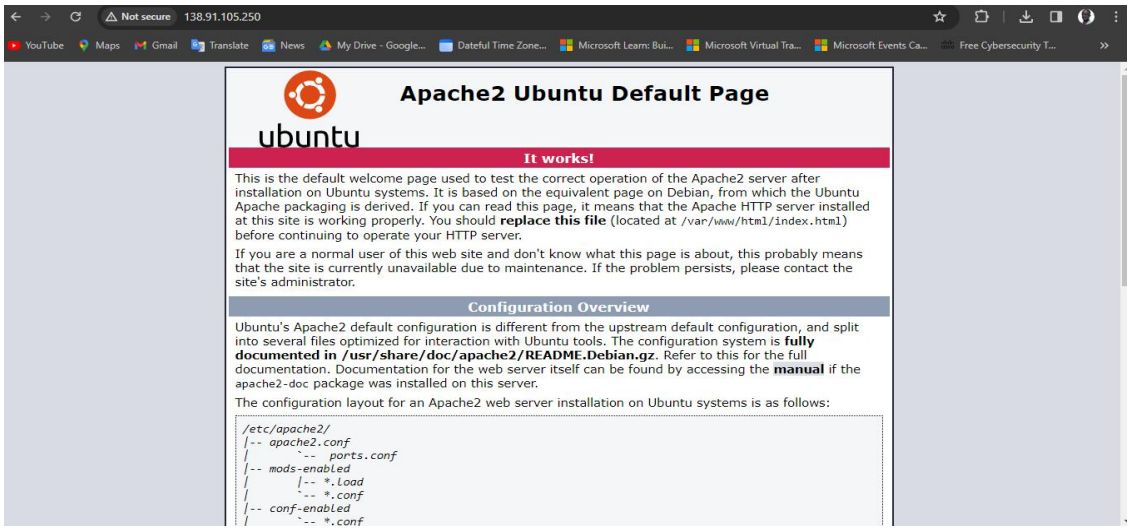
**Filebeat was installed on the web server and the Filebeat service was made active to forward log data from the web server to the ELK server.**

## Configuring the Web Server

```
Building dependency tree
Reading state information... Done
13 packages can be upgraded. Run 'apt list --upgradable' to see them.
azureuser@Public-VM:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
```

```
Processing triggers for ufw (0.36-6ubuntu1.1) ...
Processing triggers for systemd (245.4-4ubuntu3.23) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.14) ...
azureuser@Public-VM:~$ sudo service apache2 start
azureuser@Public-VM:~$
```

## Successful connection to the Web server




## Installing Filebeat on the Web server and editing the filebeat.yml file

```
azureuser@Public-VM:~$ curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-7.4.0-amd64.deb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 23.1M  100 23.1M    0     0  36.3M      0 --:--:-- --:--:-- --:--:-- 36.3M
azureuser@Public-VM:~$ sudo dpkg -i filebeat-7.4.0-amd64.deb
Selecting previously unselected package filebeat.
(Reading database ... 59706 files and directories currently installed.)
Preparing to unpack filebeat-7.4.0-amd64.deb ...
Unpacking filebeat (7.4.0) ...
Setting up filebeat (7.4.0) ...
Processing triggers for systemd (245.4-4ubuntu3.23) ...
azureuser@Public-VM:~$ cd /etc/filebeat
azureuser@Public-VM:/etc/filebeat$ ls
fields.yml  filebeat.reference.yml  filebeat.yml  modules.d
azureuser@Public-VM:/etc/filebeat$ sudo nano filebeat.yml
azureuser@Public-VM:/etc/filebeat$
azureuser@Public-VM:/etc/filebeat$
```

# 3.2.2 Filebeat Screenshot

## Filebeat was configured to route web server logs to Elasticsearch

### Editing the filebeat.yml file

```
# Configure what output to use when sending the data collected by the beat.

#---------------------------- Elasticsearch output ------------------------------
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["10.0.1.4:9200"]

  # Optional protocol and basic auth credentials.
  #protocol: "https"
  #username: "elastic"
  #password: "changeme"
```

```
# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:

  # Kibana Host
  # Scheme and port can be left out and will be set to the default (http and 5601)
  # In case you specify and additional path, the scheme is required: http://localhost:5601/path
```

### Enabling the system and apache modules to send logs to the ELK server

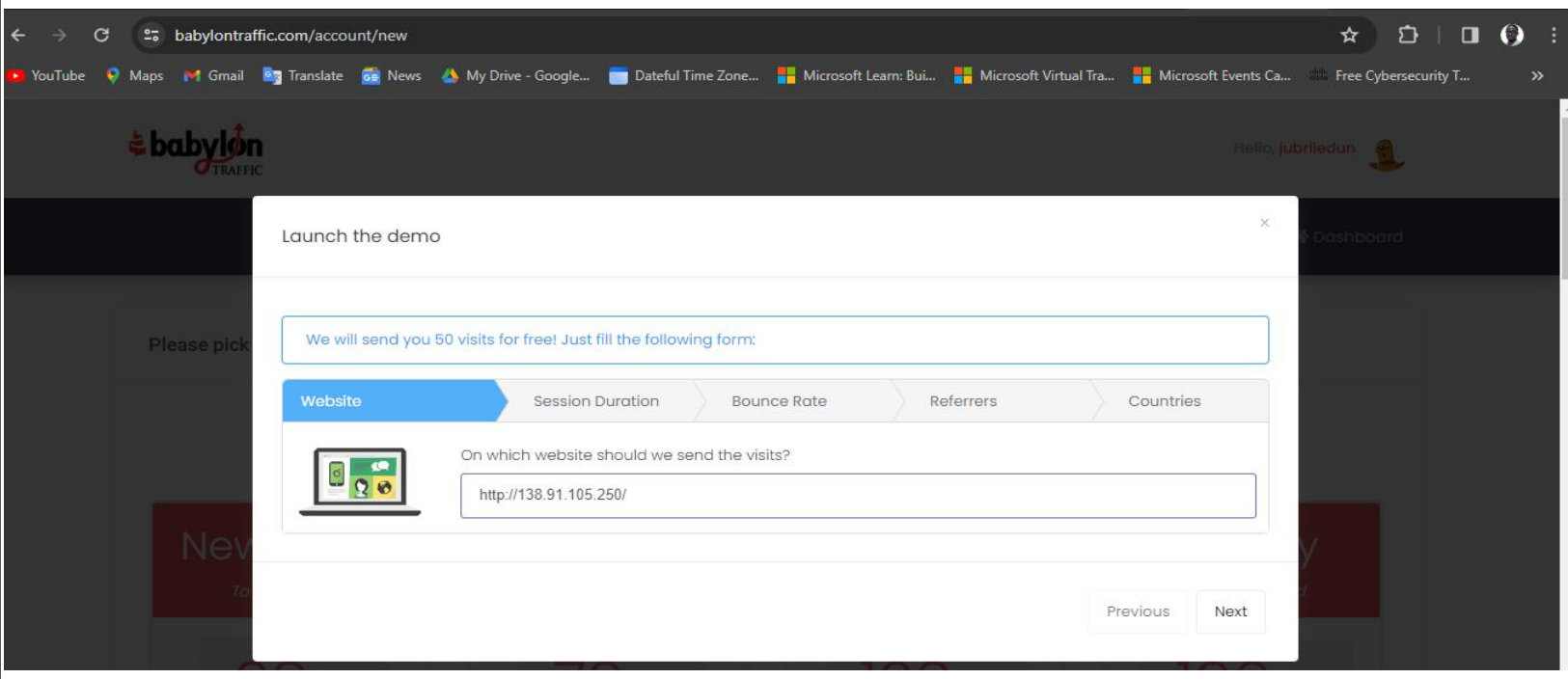
```
--- google.com ping statistics ---
587 packets transmitted, 587 received, 0% packet loss, time 586982ms
rtt min/avg/max/mdev = 2.415/3.005/12.292/0.651 ms
azureuser@Public-VM:/etc/filebeat$ sudo nano filebeat.yml
azureuser@Public-VM:/etc/filebeat$ sudo filebeat modules enable system
Enabled system
azureuser@Public-VM:/etc/filebeat$ sudo filebeat modules enable apache
Enabled apache
azureuser@Public-VM:/etc/filebeat$
```

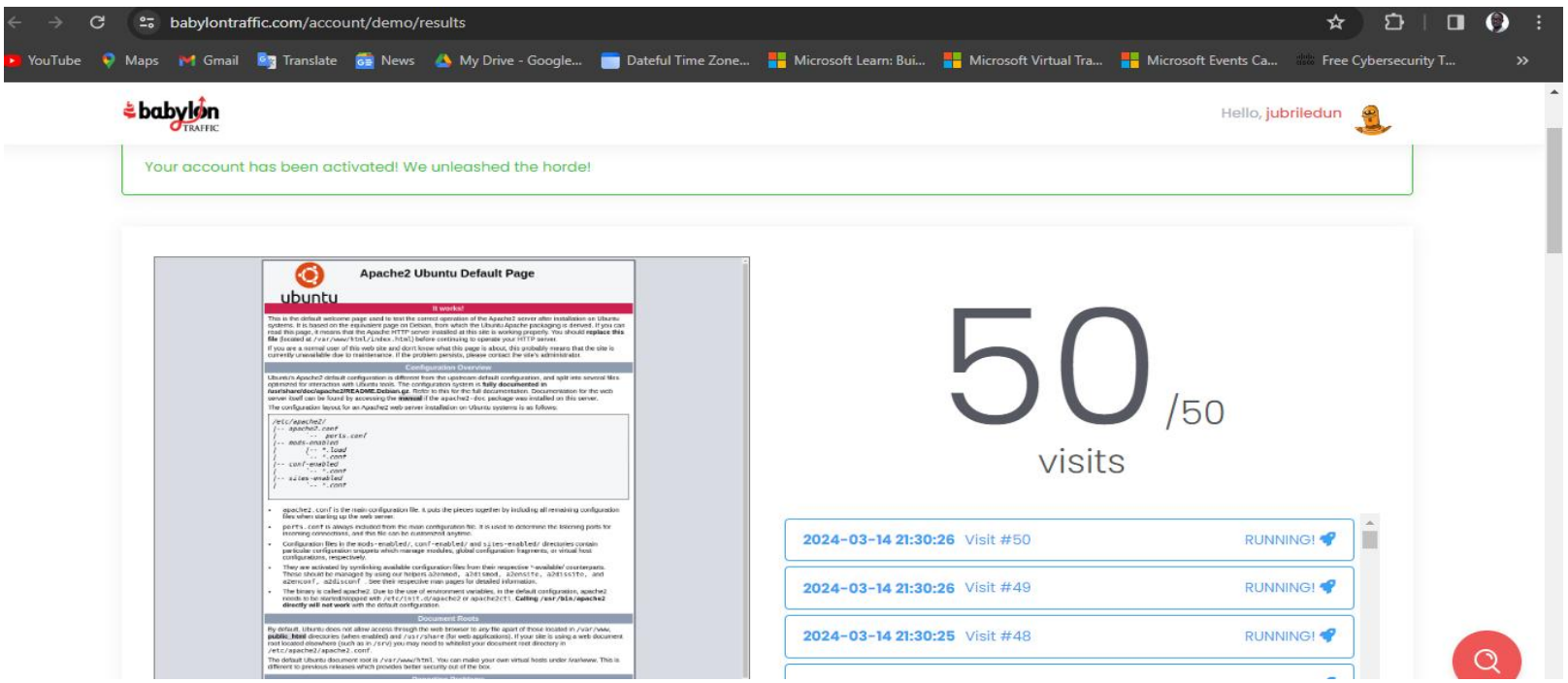
### Starting the filebeat service

```
azureuser@Public-VM:/etc/filebeat$ sudo nano filebeat.yml
azureuser@Public-VM:/etc/filebeat$ sudo filebeat setup
Index setup finished.
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
Loaded machine learning job configurations
Loaded Ingest pipelines
azureuser@Public-VM:/etc/filebeat$ sudo service filebeat start
azureuser@Public-VM:/etc/filebeat$
```

# 3.2.3 Web Traffic Screenshot

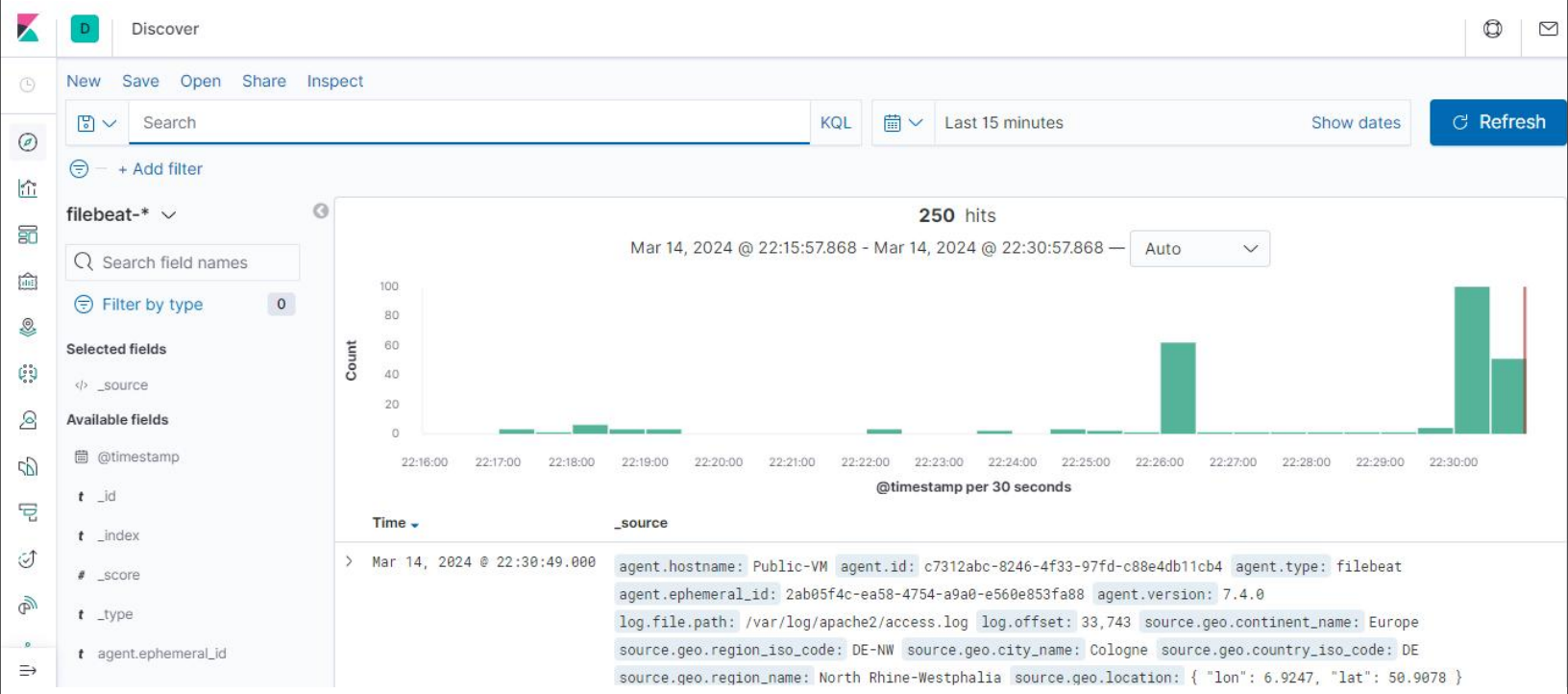
**Web traffic to the web server was simulated using https://www.babylontraffic.com.**



# 3.2.4 Logs Screenshot

## Web server logs appear in Kibana.



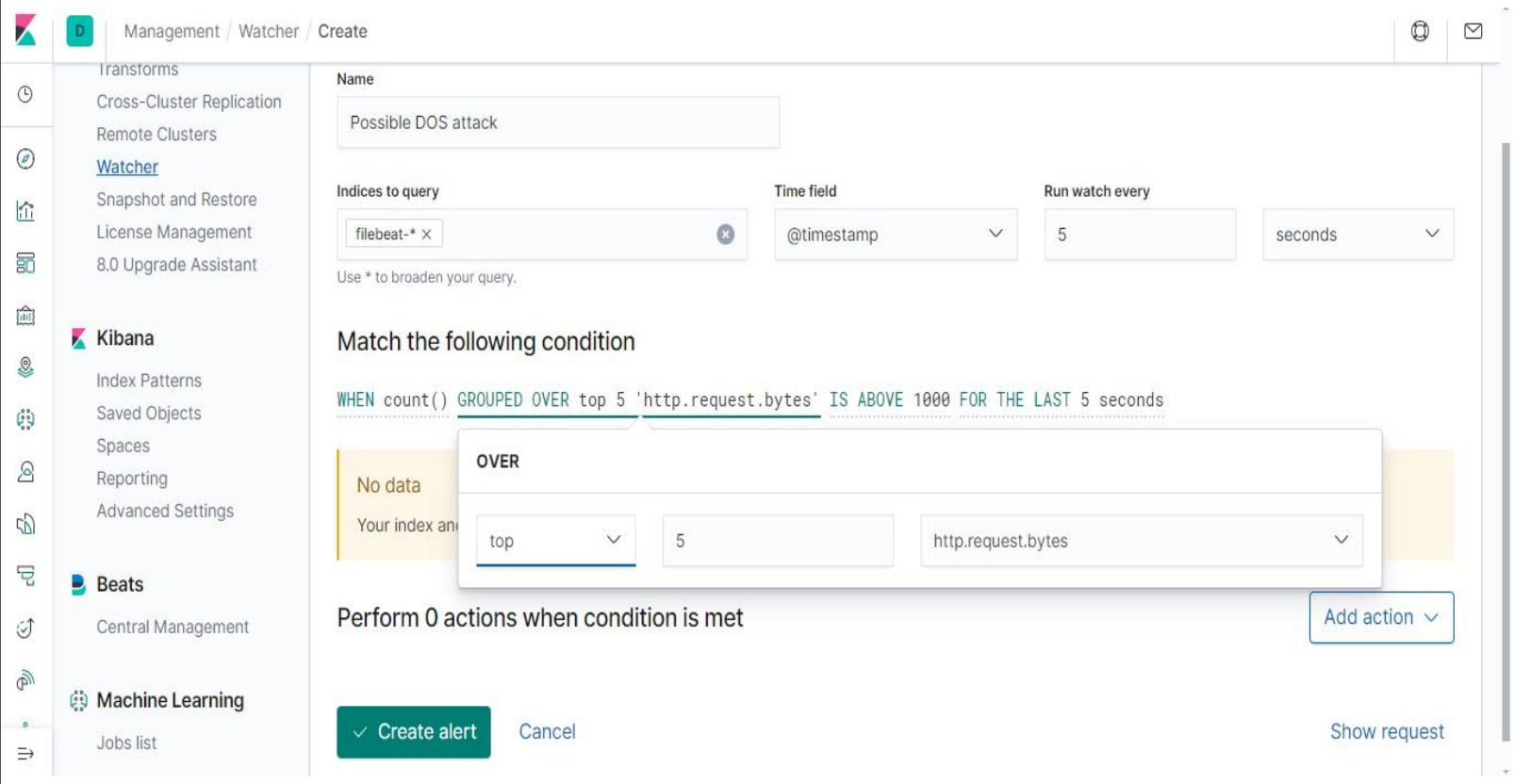

# 3.3 Build Alerts

**In this next section, alerts were created on the simulated web traffic. Alerts were built to alert of possible DoS, brute force, and probing attacks.**

**Screenshots shown in next slides:**

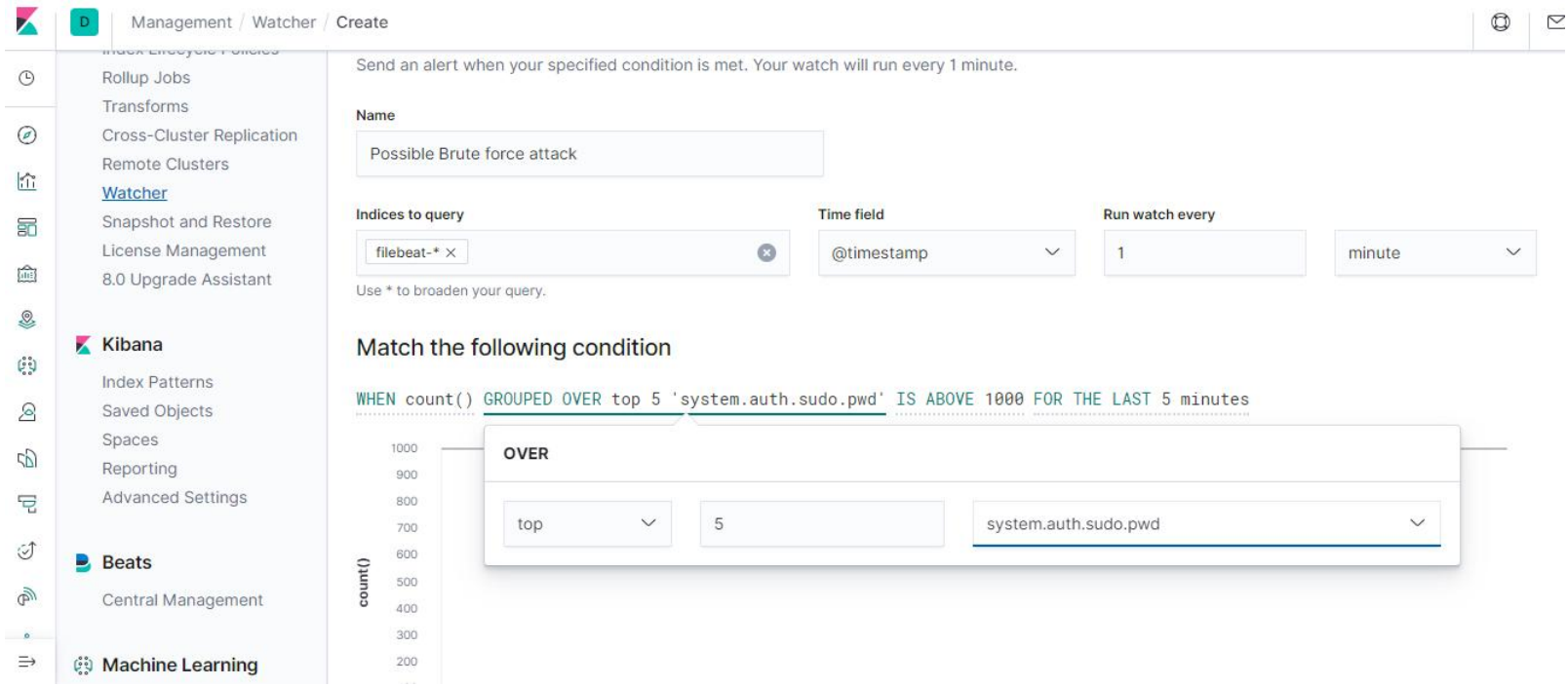
# 3.3.1 Alert Screenshot

## Alert for possible DoS attack.



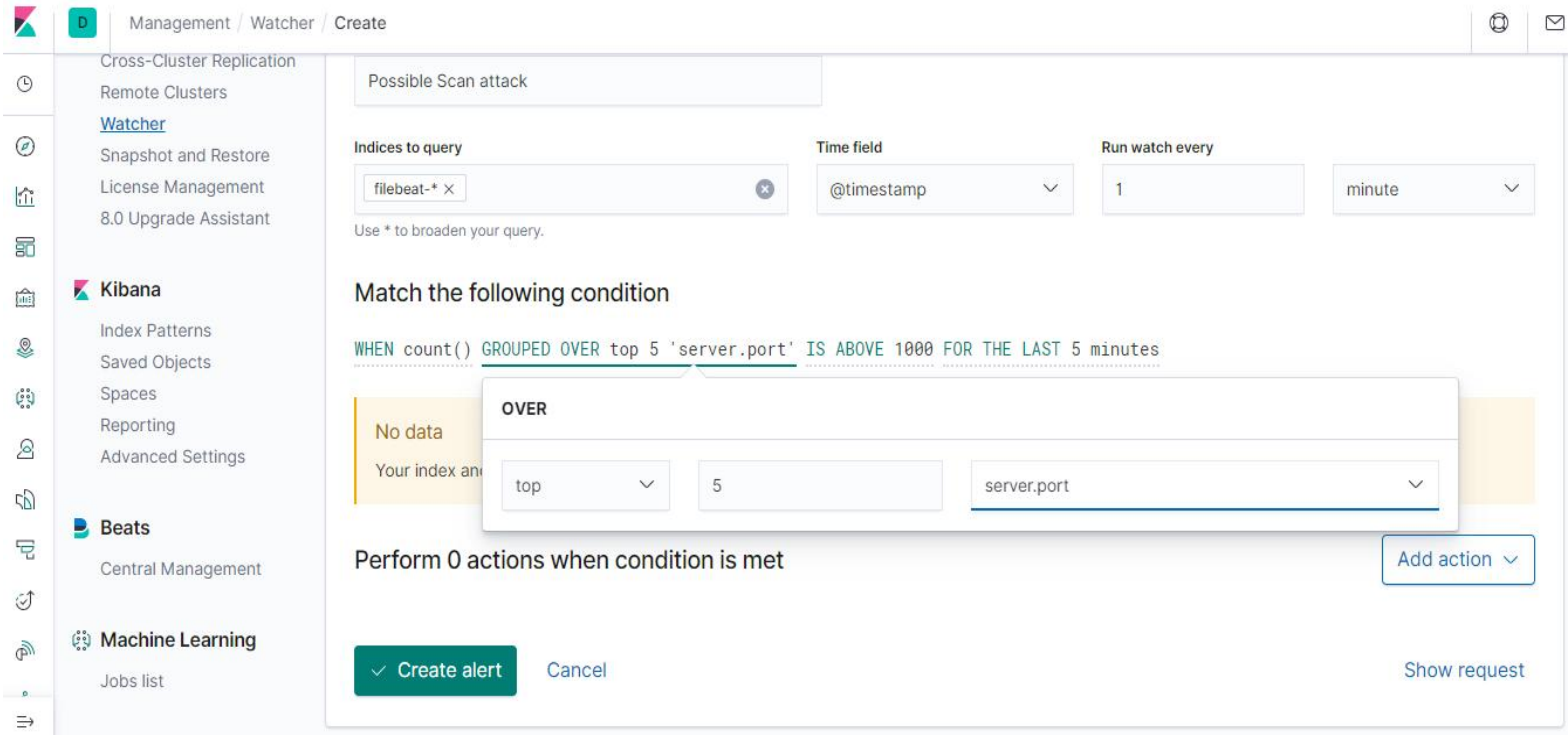
# 3.3.2 Alert Screenshot

## Alert for possible Brute Force attack.



# 3.3.3 Alert Screenshot

**Alert for possible scanning attack. During the scan, an attacker is looking to identify what ports are open.**



# 3.4 Incident Response Playbook

**In this next section, incident response playbooks were created, detailing steps to be taken in response to each of the alerts created in the last section.**

**The Incident Response Playbook was created using the National Institute for Standards and Technology (NIST) incident response plan which states four phases:**

- **Preparation**
- **Detection & Analysis**
- **Containment, Eradication and Recovery**
- **Post-Incident Activity**

# 3.4.1 Incident Response Playbook (DoS)

## Incident Response Playbook for DoS attack

- **Preparation**

**In preparing for a DoS attack, roles and responsiblities should be clearly defined (e.g. SOC analysts, CIRT etc.), communication channels shoudl be in place, including real-time collaboration tools and alternative communication channels, implement monitoring tools (e.g. SIEM), implement DOS protection tools (e.g. WAF) etc.**

- **Detection & Analysis**

**In this phase, the implemented monitoring tools will be used to monitor for high traffic volumes, unusual patterns etc and alert the necessary stakeholders if an attack occurs. Logs from the monitoring tools will be reveiwed and analyzed for possible DOS attacks or false positives.**

- **Containment, Eradication and Recovery**

**In this phase, the DOS attack will be contained by isolating the affected systems from the network, redirecting the traffic to a sandbox environment. Eradication can be done by identifying the source of the DOS attacke and blocking traffice from that source and blacklisting the IP address. After containing and eradicating the attack, we can slowly recover the affected systmes by ensureing proper system hardening, secure configuration, patching etc. of the network and system before bringing them back online.**

- **Post-Incident Activity**

**In this phase, we document lessons learned from the incident and how well we were able to handle the incident and where area of improvement is needed. Communication to the necessary stakeholders will be done in the phase also.**

# 3.4.2 Incident Response Playbook (Brute Force)

## Incident Response Playbook for Brute Force attack

- **Preparation**

**In preparing for a Brute Force attack, roles and responsiblities should be clearly defined (e.g. SOC analysts, CIRT etc.), communication channels should be in place, including real-time collaboration tools and alternative communication channels, implement monitoring tools (e.g. SIEM, IDS), implement MFA, limit login attempts, account lockout policy etc.**

- **Detection & Analysis**

**In this phase, the implemented monitoring tools will be used to monitor for unusual login patterns, multiple failed login attempts etc and alert the necessary stakeholders. Logs from the monitoring tools will be reveiwed and analyzed for possible Brute Force attacks or false positives.**

- **Containment, Eradication and Recovery**

**In this phase, the Brute Force attack will be contained by disabling compromised accounts to prevent lateral movement. Eradication can be done by identifying the source of the attacks and blocking traffic from that source and blacklisting the IP address, investigating the affected system to confirm how much damage has been casued by the attack. We can slowly recover the affected systems by ensuring password complexities, secure configuration etc. of the system before enabling the affected systems..**

- **Post-Incident Activity**

**In this phase, we document lessons learned from the incident and how well we were able to handle the incident and where area of improvement is needed. Communication to the necessary stakeholders will be done in this phase also.**

# 3.4.3 Incident Response Playbook (Scan)

## Incident Response Playbook for Probing (Scan) attack

- **Preparation**

**In preparing for a DoS attack, roles and responsiblities should be clearly defined (e.g. SOC analysts, CIRT etc.), communication channels shoudl be in place, including real-time collaboration tools and alternative communication channels, implement network segmentation, disable unsecure ports and unneccessary open ports, implement IDS etc.**

- **Detection & Analysis**

**In this phase, the IDS will be used to alert the necessary stakeholders when probing attack  is identified. Logs from the IDS will be reveiwed and analyzed to know what systems the attacker is probing and discover their intentions.**

- **Containment, Eradication and Recovery**

**In this phase, the attakcers IP address will be blocked, any unnecessary ports will be closed to prevent futue probing attacks and ensure system hardening.**

- **Post-Incident Activity**

**In this phase, we document lessons learned from the incident and how well we were able to handle the incident and where area of improvement is needed. Communication to the necessary stakeholders will be done in the phase also.**

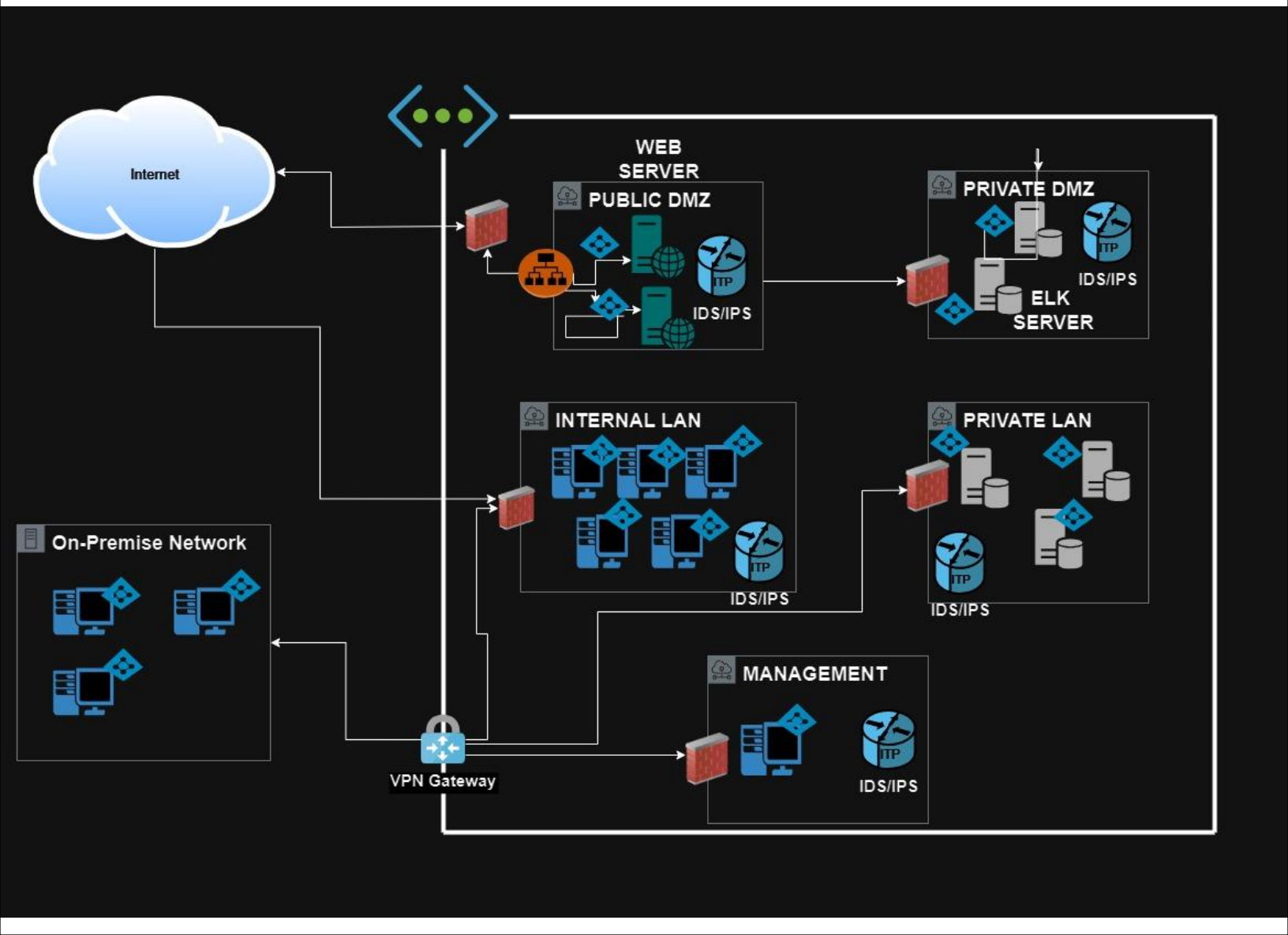
Section 4

# Designing a Zero Trust Model

# Section 4: Zero Trust Model

**In this section a Zero Trust model was designed explaining the differences between the network architecture desinged in the previous section and the Zero Trust Model architecture**

## Zero Trust Model Diagram:



# 4.1 Secure Architecture vs. Zero Trust

**Difference between the Zero Trust model architecture and the secure network architecture design based on the following topics:**

**-Segmentation:** In Zero trust, network segments are smaller and more granular where each device is treated as a separate entity, hence each device is authenticated and authorized individually.
In Secure architecture, segmentation involves dividing the network into zones. This is less granular compared to Zero Trust.

**-User Identification and Access:** In Zero trust, authentication and authorization depends on Identity and Access management solutions, which require users to authenticate themselves before accessing any resources.
In Secure architecture, users are also authenticated based on Identity, but access is more broad compared to Zero trust.

**-Concept of Trust:** In Zero Trust model, the network is always assumed to be compromised, hence zero trust verifies every request before granting access to resources.
In Secure architecture, trust is assumed in the trusted zone, hence, the need for firewalls, IDS/IPS etc. installed at the perimter of each subnetwork.

# 4.1 Secure Architecture vs. Zero Trust

**Difference between the Zero Trust model architecture and the secure network architecture design based on the following topics:**

**-Data Security:** In Zero trust, security is more data-centric. Data access is strictiy enforced based on user identites, sensitivity of data being accessed, lease privilege etc.
In Secure architecture, data security is not as granular as Zero trust. The focuse is more on perimeter defense and network-level security.

# Thank You