AZURE CLOUD SECURITY LAB

Enterprise Security Implementation and Documentation

Project Report v2.0 | June 2026

Author	Mohammad Ansab Siddiqui
Role	Technical Consultant -- ITSM and Solutions
Date	June 2026
Platform	Microsoft Azure -- Free Tier (200 USD Credit)
Region	Central India
Classification	Portfolio Project -- Public
Portfolio	ansabsid.github.io

2	9	4
Azure VMs	**Sentinel Connectors**	**KQL Detection Rules**
Ubuntu 22.04 LTS	Microsoft Sentinel SIEM	Custom threat detection

Table of Contents

1. Executive Summary

This document presents the complete design, implementation, and validation of an enterprise-grade cloud security laboratory built on Microsoft Azure. The lab was constructed within the Azure free tier using a 200 USD credit, demonstrating that professional-grade security infrastructure can be built and validated at near-zero cost.

The project demonstrates end-to-end capability across four core disciplines: cloud infrastructure management, network architecture and segmentation, security operations, and threat detection using a Security Information and Event Management (SIEM) platform.

Real-world attack scenarios were simulated using Kali Linux, a professional penetration testing distribution running locally on an Apple Mac via UTM virtualisation. These attacks were detected by Microsoft Sentinel using custom KQL detection rules. The lab proves a complete red team versus blue team cycle: attack, detect, investigate, and respond.

> **Key Outcome**
> A fully operational cloud security lab that successfully detected real attack traffic from Kali Linux, raised automatic incidents in Microsoft Sentinel, and demonstrated enterprise-standard network segmentation -- all within the Azure free tier.

1.1 Skills Demonstrated

Skill Domain	Capability Demonstrated	Evidence
Azure Administration	VM deployment, VNet design, NSG configuration, Bastion	Two VMs, 4 subnets, layered NSG rules
Network Security	Network segmentation, DMZ architecture, firewall rules	Deny-all default, tiered subnet access controls
SIEM Operations	Sentinel deployment, 10 data connectors, log analytics	10 connectors (9 connected), 4 custom KQL rules
Threat Detection	Custom KQL rules, automatic incident creation, alert tuning	Port scan and SSH brute force incidents confirmed
Offensive Security	Nmap, Nikto, Gobuster from Kali Linux against live Azure VM	Real attack traffic detected in Sentinel logs

Cost Management	Free tier discipline, VM deallocation after each session	Under 5 USD total spend across entire project

2. Project Objectives

The laboratory was designed to achieve the following objectives:

1. Build a multi-tier cloud network architecture in Microsoft Azure that mirrors enterprise deployment patterns.
2. Implement network security controls including Network Security Groups, subnet segmentation, and a DMZ architecture with a dedicated jump server.
3. Deploy Microsoft Sentinel as a cloud-native SIEM and connect it to multiple data sources including Azure Activity logs and VM Syslog.
4. Create custom threat detection rules using Kusto Query Language covering brute force attacks, port scanning, unauthorised logins, and infrastructure changes.
5. Simulate real-world attack scenarios from Kali Linux and validate that the SIEM detects and raises incidents correctly.
6. Document the complete implementation to demonstrate practical cloud security knowledge for professional portfolio purposes.

> **Scope**
> This is an intentional security lab. All attack simulations were performed against infrastructure owned and controlled by the author in an isolated Azure environment. No external systems were targeted.

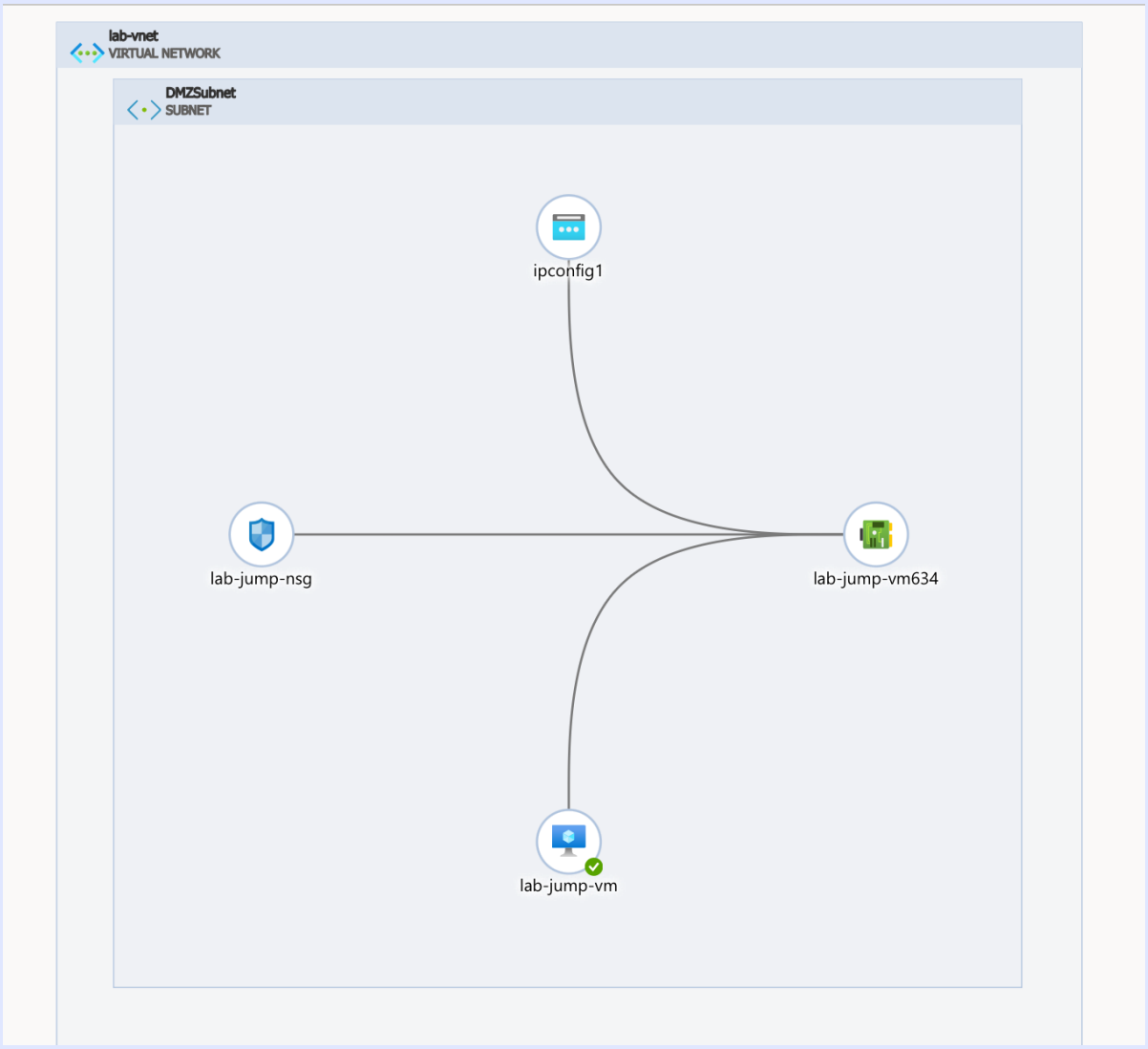
3. Architecture Design

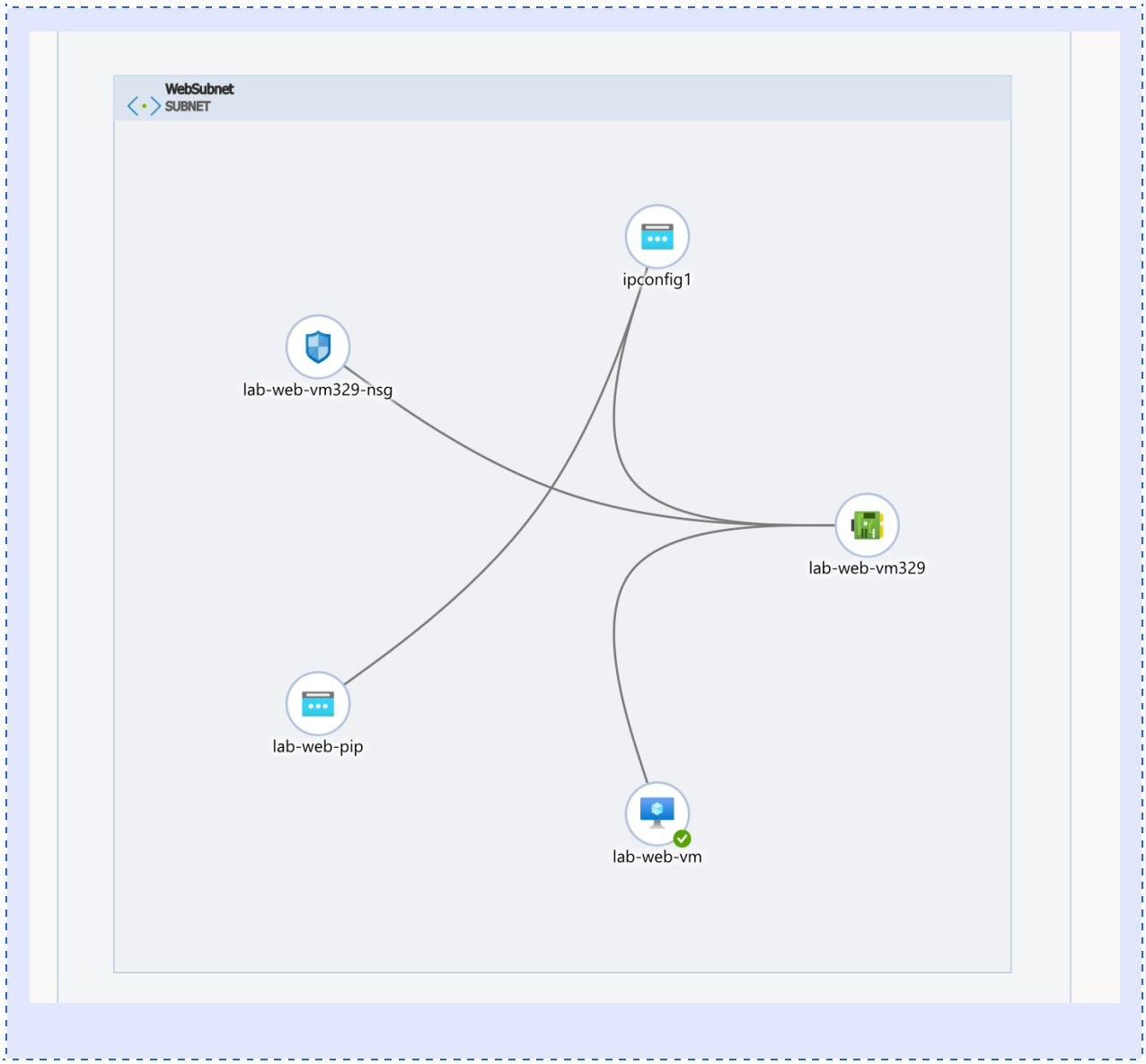
3.1 Architecture Overview

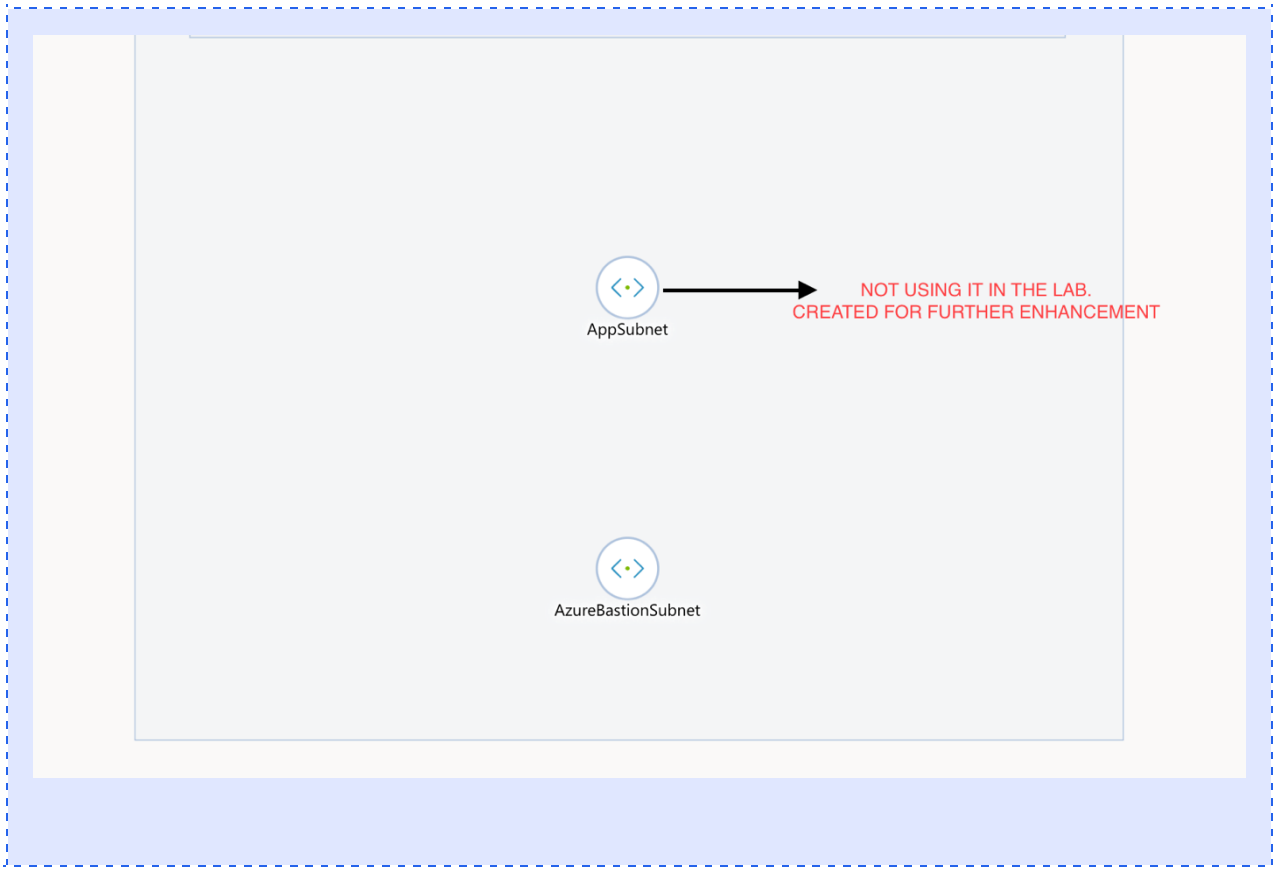
The lab implements a two-VM network architecture with a simulated attacker operating from outside the network boundary. The design follows the principle of defence in depth -- using multiple layers of security controls rather than relying on any single mechanism.

[SCREENSHOT SS-01]

Azure Network Watcher Topology View showing VNet, subnets, VMs, and NSG associations




WebSubnet
SUBNET
ipconfig1
lab-web-vm329-nsg
lab-web-vm329
lab-web-pip
lab-web-vm



3.2 Network Architecture

All resources are deployed in a single Virtual Network with four dedicated subnets, each serving a distinct security purpose:

Subnet	CIDR Range	Purpose	Security Controls
DMZSubnet	10.0.1.0/24	Jump Server -- controlled admin entry point	NSG: Allow SSH from admin IP only, Deny-all default
WebSubnet	10.0.2.0/24	Web Server -- internet-facing attack target	NSG: Allow HTTP from internet, Allow SSH from Kali IP only
AppSubnet	10.0.3.0/24	Application tier -- reserved for future expansion	NSG: Deny-all default (no active resources deployed)
AzureBastionSubnet	10.0.5.0/26	Azure Bastion managed service	/26 minimum required by Microsoft for Bastion deployment

Note on WebSubnet Access
The web VM has a static public IP assigned for attack simulation. Kali Linux connects directly over the internet. The NSG allows SSH only from Kali IP and HTTP from any source. All other inbound traffic is denied by the default deny-all rule.

3.3 Virtual Machine Inventory

VM Name	OS	Size	Subnet	Public IP	Role
lab-jump-vm	Ubuntu 22.04 LTS	B2ats_v2 (2 vCPU, 4GB)	DMZSubnet	None (Bastion only)	Jump Server / Admin Entry Point
lab-web-vm	Ubuntu 22.04 LTS	B2ats_v2 (2 vCPU, 4GB)	WebSubnet	Static: 20.219.23.147	Web Server / Attack Target

VM Sizing Note
B1ls (1 vCPU) was the target free tier size but was unavailable across all regions at the time of deployment. B2ats_v2 (2 vCPU, 4GB) was the smallest available size in Central India. Both VMs use this size. Total vCPU usage of 4 matched the regional quota exactly -- no quota increase was required.

3.4 Security Architecture Principles

- Zero Public Exposure by Default: No VM has a public IP by default. A public IP is assigned to lab-web-vm only for attack simulation and dissociated after each session.
- Bastion-Only Management Access: All administrative connections go through Azure Bastion, eliminating exposed SSH ports on the internet for management purposes.
- Jump Server Architecture: Administrative access to internal VMs flows through the jump server in the DMZ, creating a single audited entry point.
- Deny-All Default Rules: Every NSG enforces a deny-all inbound rule -- traffic not explicitly permitted is blocked.
- Network Segmentation: Each subnet has its own NSG with rules specific to its function and trust level.
- SIEM Monitoring: VM system logs are forwarded to Microsoft Sentinel for continuous monitoring and threat detection.

4. Implementation

4.1 Azure Environment Setup

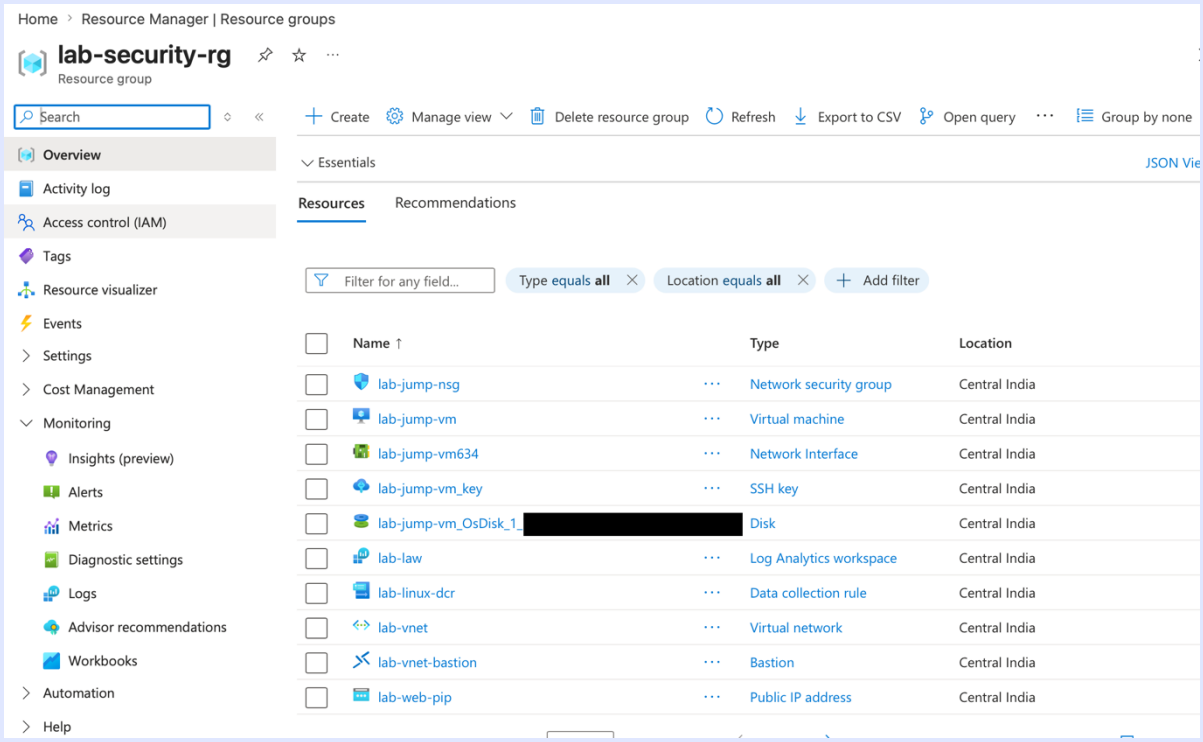
4.1.1 Resource Group

A dedicated Resource Group was created to contain all lab resources, enabling clean cost tracking and single-command deletion of the entire environment when needed.

Resource Group Name: lab-security-rg
Subscription: Azure Free Trial
Region: Central India
Purpose: Container for all lab resources -- VMs, VNet, NSGs, Sentinel workspace

[SCREENSHOT SS-02]

Azure Portal -- Resource Group overview showing all deployed resources in lab-security-rg



4.1.2 Virtual Network

A /16 Virtual Network provides 65,534 usable host addresses divided into four subnets. The AzureBastionSubnet requires a minimum of /26 as mandated by Microsoft for Bastion deployments.

VNet Name: lab-vnet
Address Space: 10.0.0.0/16
Region: Central India
Subnets: DMZSubnet /24, WebSubnet /24, AppSubnet /24, AzureBastionSubnet /26
DNS: Azure-provided (default)

4.2 Virtual Machine Deployment

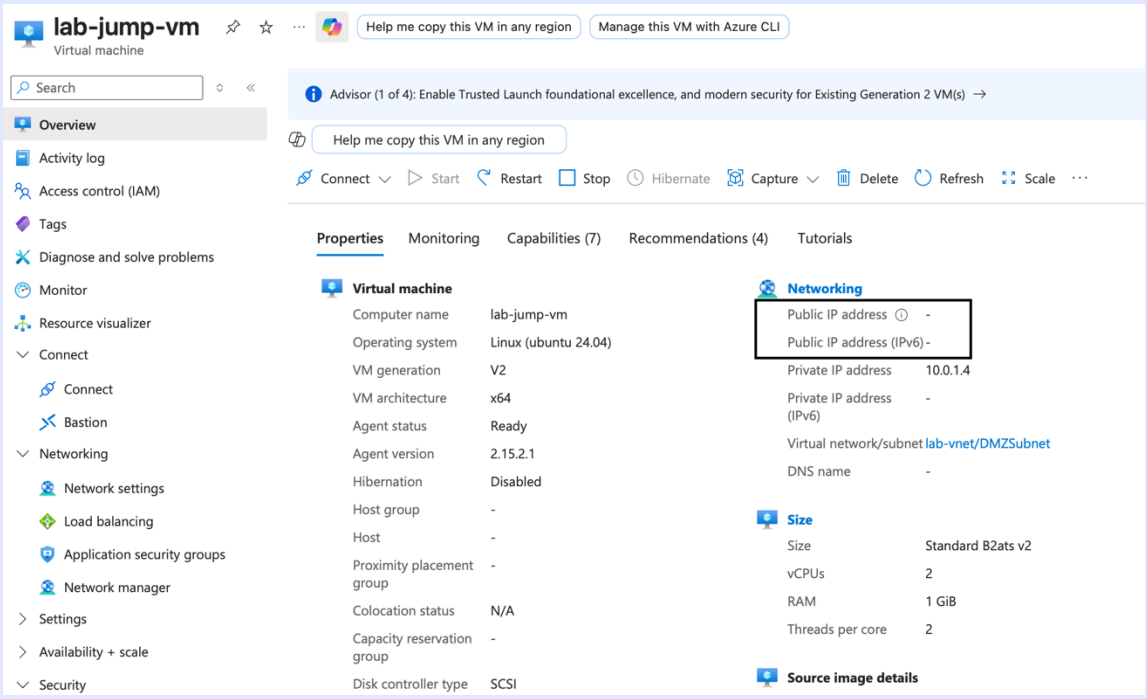
4.2.1 Jump Server (lab-jump-vm)

The jump server is deployed in the DMZ subnet with no public IP. Administrators connect via Azure Bastion and then SSH to internal VMs using private IP addresses.

> **Why a Jump Server?**
> Without a jump server every internal VM would need a public IP or an exposed management port. The jump server concentrates all administrative access through a single audited entry point, reducing the attack surface and creating a complete record of all admin sessions.

[SCREENSHOT SS-03]

Azure Portal -- lab-jump-vm overview showing running status, private IP, no public IP



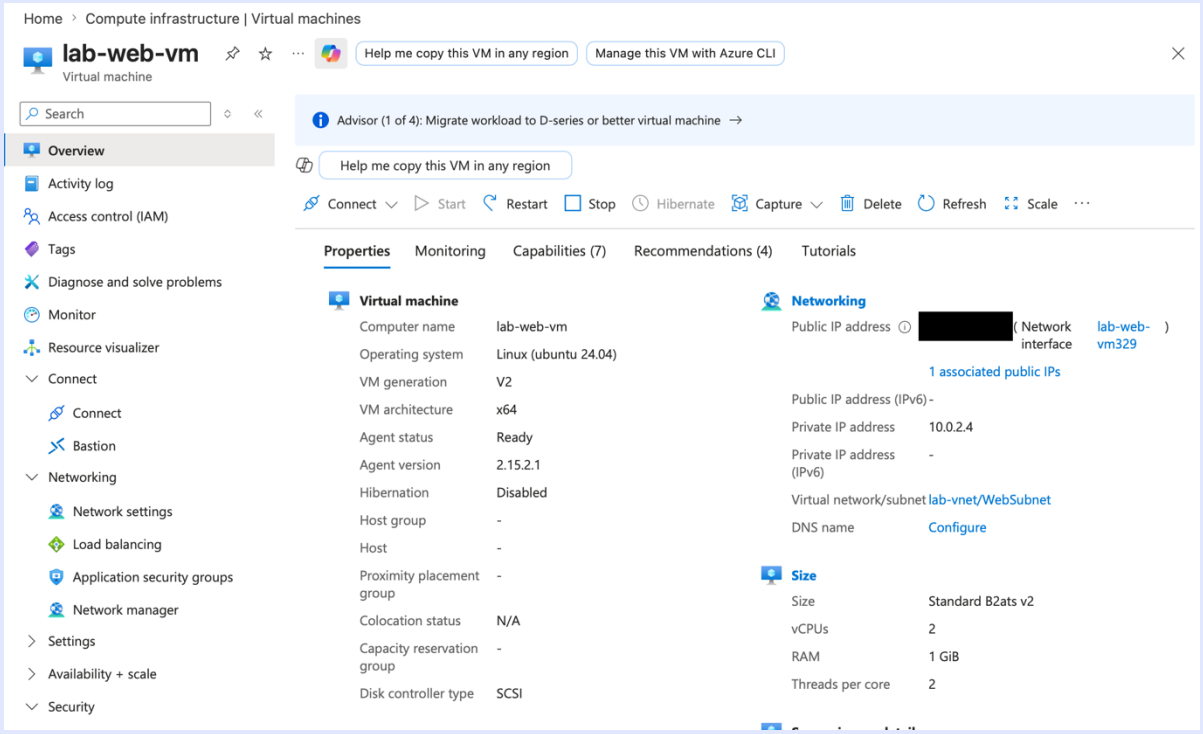
4.2.2 Web Server (lab-web-vm)

The web server runs Ubuntu 22.04 LTS with Nginx installed, representing an internet-facing service and the primary attack target for the security validation exercises.

```
sudo apt update && sudo apt install nginx -y
sudo systemctl enable nginx && sudo systemctl start nginx
```

[SCREENSHOT SS-04]

Azure Portal -- lab-web-vm overview showing running status, public IP 20.219.23.147, WebSubnet



4.3 Network Security Configuration

4.3.1 NSG Rules -- lab-web-vm329-nsg

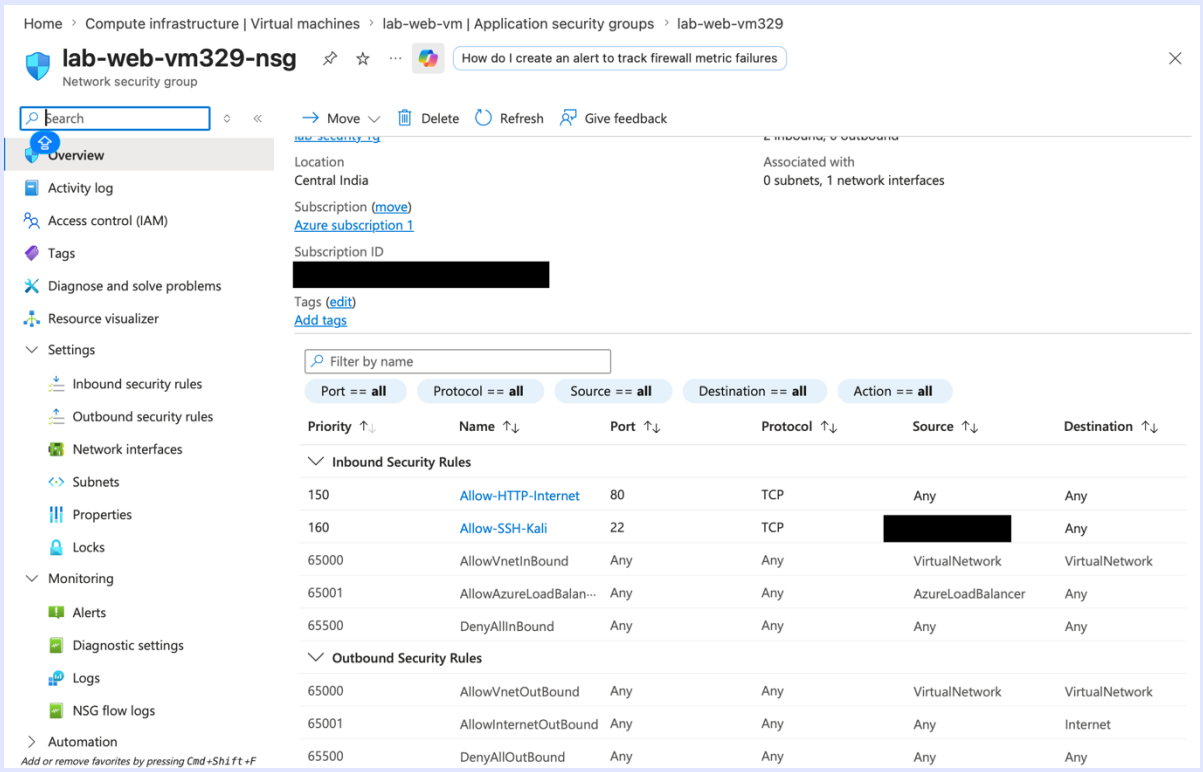
The web server NSG is associated with the network interface of lab-web-vm. Rules are evaluated from lowest to highest priority number and the first matching rule wins.

Priority	Rule Name	Port	Protocol	Source	Action	Purpose
150	Allow-HTTP-Internet	80	TCP	Any	Allow	Web traffic for Nikto and Gobuster scans
160	Allow-SSH-Kali	22	TCP	Hidden IP	Allow	SSH only from the Kali Linux attacker IP
65000	AllowVnetInBound	Any	Any	VirtualNetwork	Allow	Azure default -- internal VNet traffic

65001	AllowAzureLoadBalancer	Any	Any	AzureLoadBalancer	Allow	Azure default -- infrastructure health probes
65500	DenyAllInBound	Any	Any	Any	Deny	Azure default -- deny all unmatched traffic

[SCREENSHOT SS-05]

Azure Portal -- lab-web-vm329-nsg inbound security rules



4.3.2 NSG Rules -- lab-jump-nsg

The jump server NSG was manually created and attached to the NIC of lab-jump-vm, as Azure did not auto-create one when the VM was deployed without an NSG selected.

Priority	Rule Name	Port	Protocol	Source	Action	Purpose
100	Allow-SSH-Kali	22	TCP	Hidden IP	Allow	SSH management from authorised Kali IP only

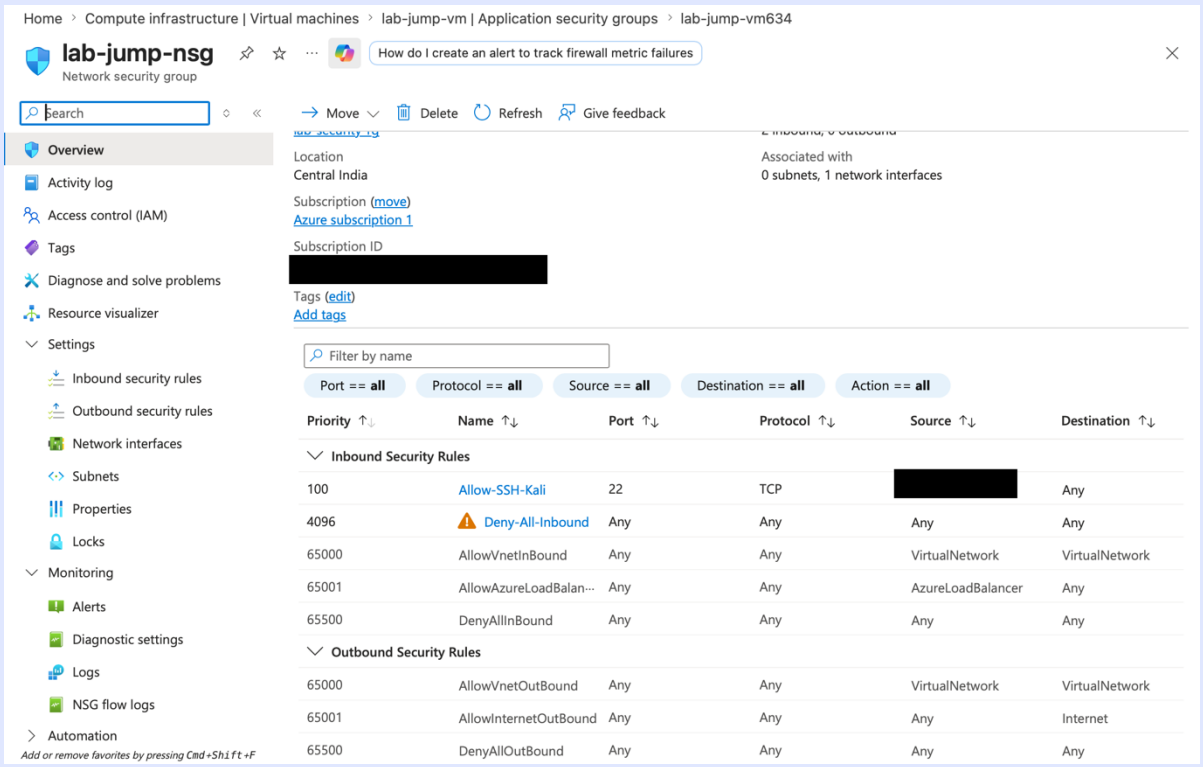
4096	Deny-All-Inbound	Any	Any	Any	Deny	Custom deny-all -- explicit block before Azure defaults
65000	AllowVnetInBound	Any	Any	VirtualNetwork	Allow	Azure default -- VNet internal traffic
65001	AllowAzureLoadBalancer	Any	Any	AzureLoadBalancer	Allow	Azure default -- health probes
65500	DenyAllInBound	Any	Any	Any	Deny	Azure default deny-all fallback

Note on Custom Deny-All

The jump NSG has a custom Deny-All-Inbound at priority 4096 in addition to Azure's built-in deny at 65500. The custom rule fires first and provides an explicitly documented control layer separate from the Azure default.

[SCREENSHOT SS-06]

Azure Portal -- lab-jump-nsg inbound security rules



4.3.3 NSG Flow Logs

Created Via: Azure Network Watcher (avoids template validation errors in the NSG blade)
Storage Account: labflowlogs1998
Storage Redundancy: LRS (Locally Redundant -- lowest cost tier)
Retention Period: 7 days
Flow Log Version: Version 2
Traffic Analytics: Disabled (avoids additional per-GB charge)
NSGs Covered: lab-web-vm329-nsg and lab-jump-nsg

4.4 Azure Bastion Deployment

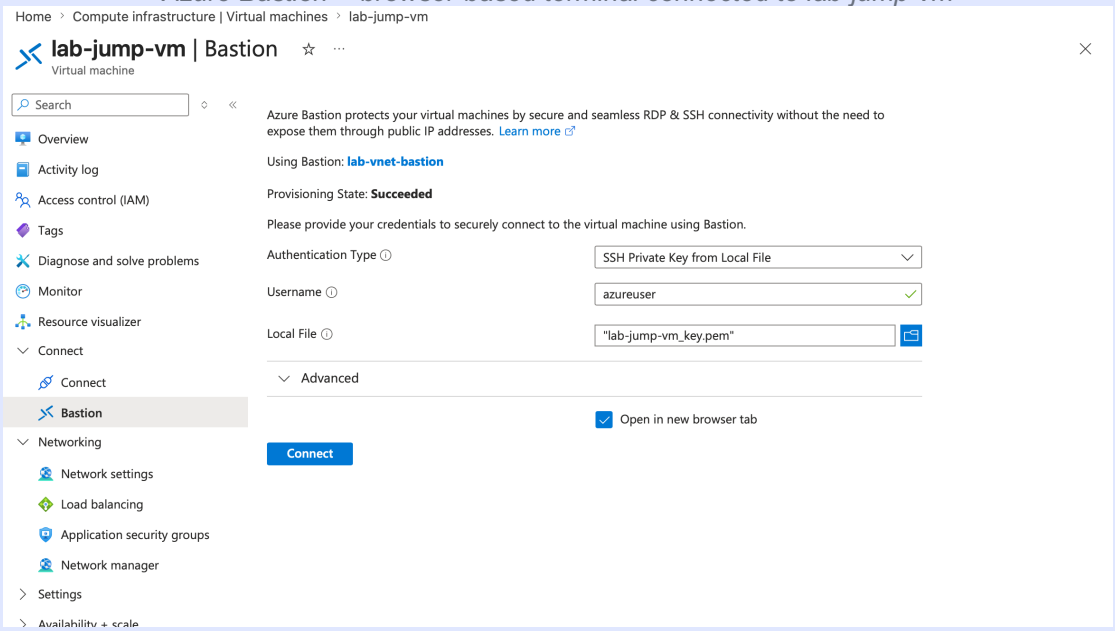
Azure Bastion provides browser-based SSH connectivity to VMs without public IPs or exposed management ports. It requires a dedicated subnet of minimum /26 named AzureBastionSubnet.

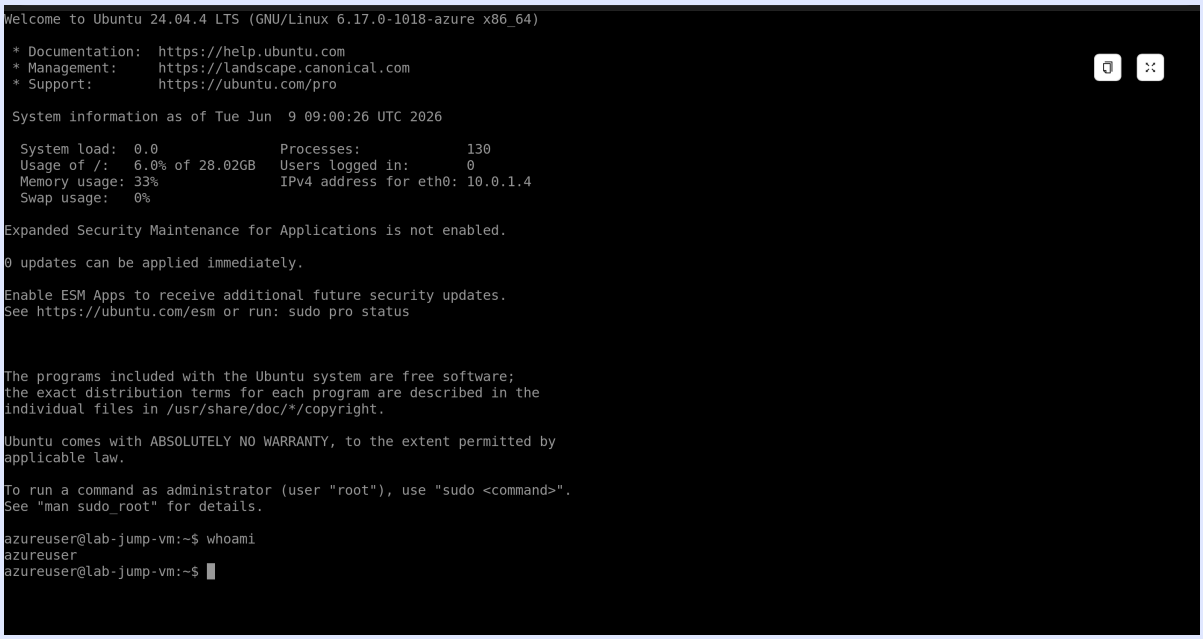
Bastion Name: lab-bastion
Subnet: AzureBastionSubnet (10.0.5.0/26)
SKU: Basic
Public IP: lab-bastion-pip (static)

- Eliminates public IPs on all management VMs
- All sessions TLS-encrypted through the Azure portal -- no direct internet exposure
- Supports SSH key authentication for Linux VMs

[SCREENSHOT SS-07]

Azure Bastion -- browser-based terminal connected to lab-jump-vm




Welcome to Ubuntu 24.04.4 LTS (GNU/Linux 6.17.0-1018-azure x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
System information as of Tue Jun 9 09:00:26 UTC 2026
System load: 0.0
Usage of /: 6.0% of 28.02GB
Memory usage: 33%
Swap usage: 0%
Processes: 130
Users logged in: 0
IPv4 address for eth0: 10.0.1.4
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
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>".
See "man sudo_root" for details.
azureuser@lab-jump-vm:~$ whoami
azureuser
azureuser@lab-jump-vm:~$

5. SIEM Implementation -- Microsoft Sentinel

5.1 Architecture

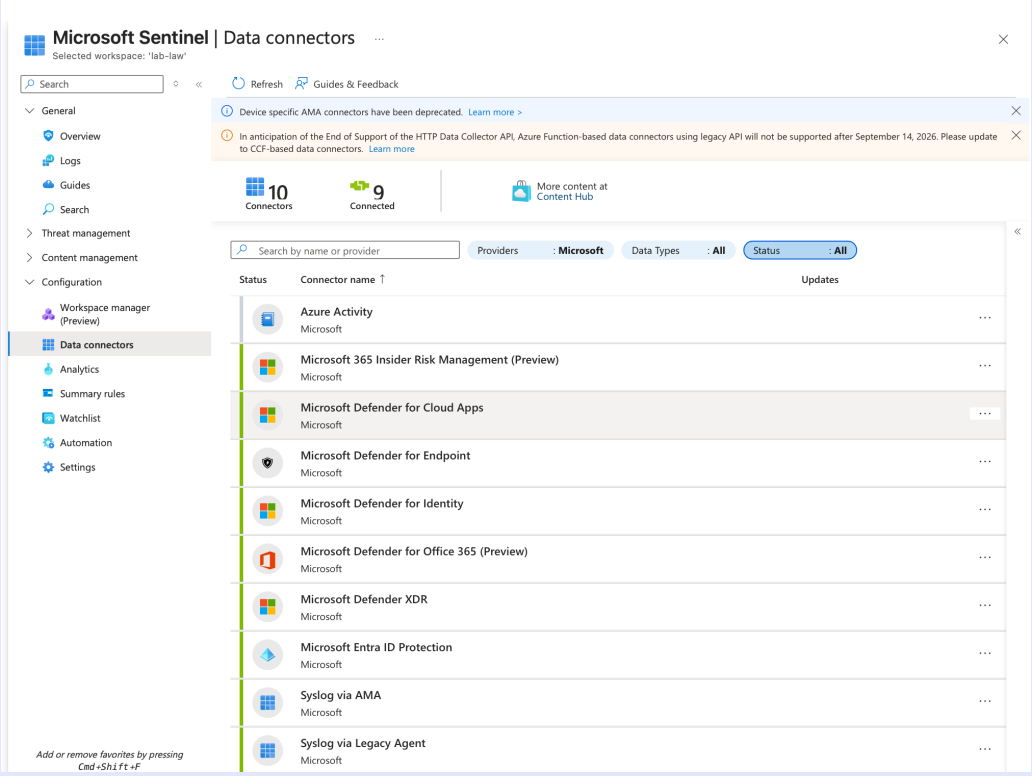
Component	Name	Purpose
Log Analytics Workspace	lab-law	Log storage, indexing, and KQL query engine
Microsoft Sentinel	Enabled on lab-law	SIEM -- analytics rules, incidents, investigation
Data Collection Rule	lab-linux-dcr	Defines logs to collect from Linux VMs via AMA
Azure Monitor Agent	Auto-installed via DCR	Forwards VM syslog data to Log Analytics

5.2 Data Connectors

Ten data connectors are configured in the Sentinel workspace. Nine are successfully connected. Azure Activity is pending completion of a policy remediation task.

[SCREENSHOT SS-08]

Microsoft Sentinel -- Data Connectors page showing all 10 connectors and their status



Connector Name	Data Type	Status
Microsoft Defender XDR	Security alerts and incidents	Connected
Microsoft Defender for Cloud Apps	Cloud app activity	Connected
Microsoft Defender for Endpoint	Endpoint alerts and telemetry	Connected
Microsoft Defender for Identity	Identity-based threats	Connected
Microsoft Defender for Office 365 (Preview)	Email and collaboration threats	Connected
Microsoft Entra ID Protection	Identity risk events	Connected
Microsoft 365 Insider Risk Management (Preview)	Insider risk signals	Connected
Syslog via AMA	Linux VM auth and system logs	Connected
Syslog via Legacy Agent	Legacy syslog collection	Connected
Azure Activity	Azure control plane operations	Not Connected -- remediation task pending

> **Azure Activity Connector**
> The Azure Activity connector uses an Azure Policy assignment. After creating the policy via the Launch Wizard, a remediation task must be run manually. Go to Azure Policy > Assignments > find the Sentinel assignment > Remediation > Create remediation task. The connector should show Connected within 30 minutes.

5.3 Log Verification

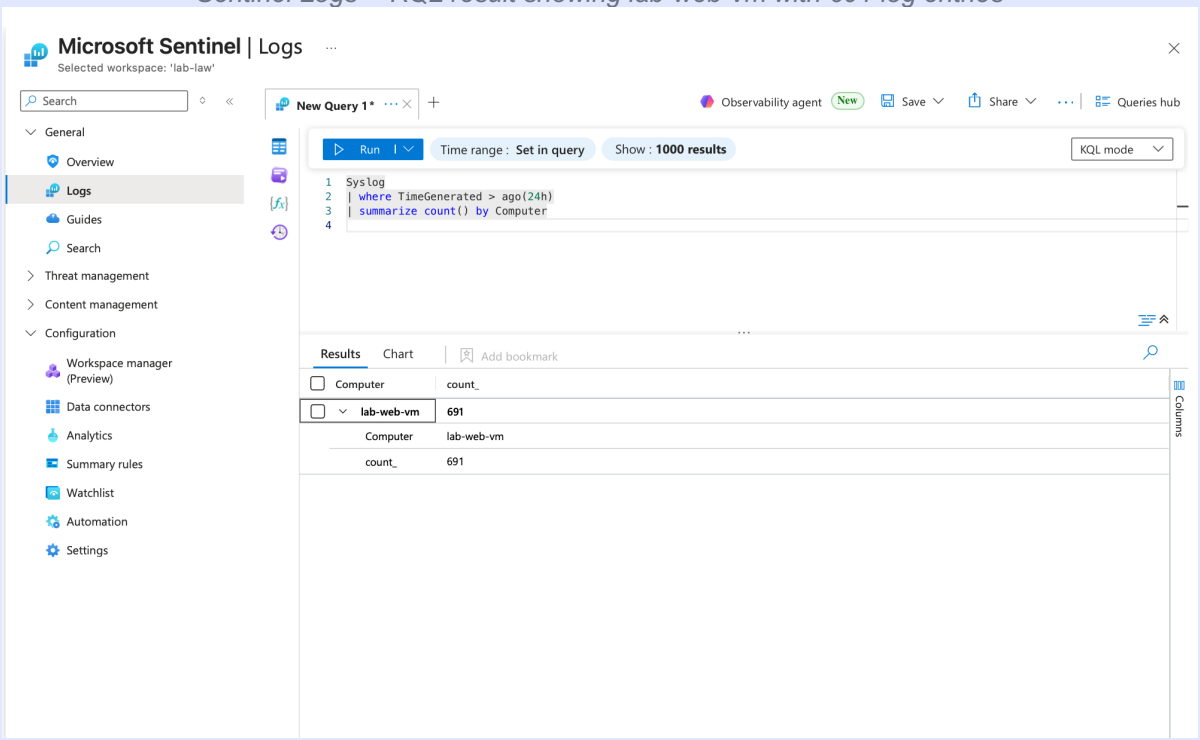
Before creating analytics rules, log ingestion was verified using KQL:

```
Syslog
| where TimeGenerated > ago(24h)
| summarize count() by Computer
```

Result: lab-web-vm returned 691 log entries, confirming syslog data was flowing correctly to Sentinel.

[SCREENSHOT SS-09]

Sentinel Logs -- KQL result showing lab-web-vm with 691 log entries



6. Threat Detection Rules

Four custom analytics rules were created in Microsoft Sentinel via the Microsoft Defender portal (security.microsoft.com). Each rule runs on a schedule and creates an incident automatically when the detection threshold is exceeded.

6.1 Rule 1 -- SSH Brute Force Attack

Rule Name: SSH Brute Force Attack
Severity: High
MITRE ATT&CK Tactic: Credential Access (TA0006)
Run Frequency: Every 5 minutes
Lookback Window: 5 minutes
Alert Threshold: Greater than 0 results
Detection Logic: 5 or more failed SSH login attempts from the same IP within 5 minutes

KQL Query:

```
Syslog
| where ProcessName == "sshd"
| where SyslogMessage contains "Failed password"
    or SyslogMessage contains "Invalid user"
    or SyslogMessage contains "Connection closed by invalid user"
| extend SourceIP = extract(@"from (\S+)", 1, SyslogMessage)
| where isnotempty(SourceIP)
| summarize FailureCount = count() by SourceIP, Computer, bin(TimeGenerated, 5m)
| where FailureCount >= 5
```

6.2 Rule 2 -- Port Scan Detection

Rule Name: Port Scan Detected
Severity: Medium
MITRE ATT&CK Tactic: Discovery (TA0007)
Run Frequency: Every 5 minutes
Lookback Window: 5 minutes
Alert Threshold: Greater than 0 results
Detection Logic: Single IP connecting to more than 10 different ports within 5 minutes

6.3 Rule 3 -- SSH Login from Unknown IP

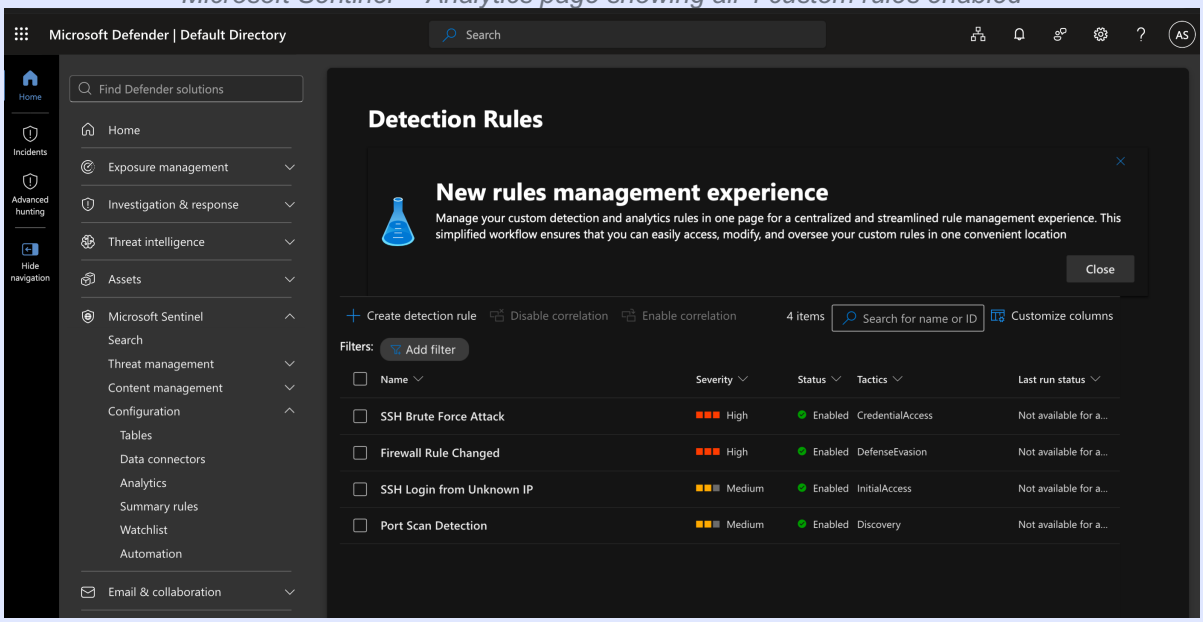
Rule Name: SSH Login from Unknown IP
Severity: Medium
MITRE ATT&CK Tactic: Initial Access (TA0001)
Run Frequency: Every 10 minutes
Lookback Window: 10 minutes
Alert Threshold: Greater than 0 results
Detection Logic: Any successful SSH authentication -- all logins trigger investigation

6.4 Rule 4 -- Firewall Rule Changed

Rule Name: Firewall Rule Changed
Severity: High
MITRE ATT&CK Tactic: Defense Evasion (TA0005)
Run Frequency: Every 15 minutes
Lookback Window: 15 minutes
Alert Threshold: Greater than 0 results
Detection Logic: Any write operation against NSG security rules in the subscription

[SCREENSHOT SS-10]

Microsoft Sentinel -- Analytics page showing all 4 custom rules enabled



7. Attack Simulation -- Red Team Exercise

7.1 Attacker Setup

Attacker Platform: Kali Linux (UTM virtualisation on Apple Mac)
Source IP: Kali Linux VM Public IP
Target IP:(lab-web-vm public IP)
Target OS: Ubuntu 22.04 LTS
Target Services: SSH (port 22), HTTP (port 80) via Nginx
Attack Tools Used: Nmap, Nikto, Gobuster, SSH invalid user bash loop

7.2 Attack 1 -- Network Reconnaissance (Nmap)

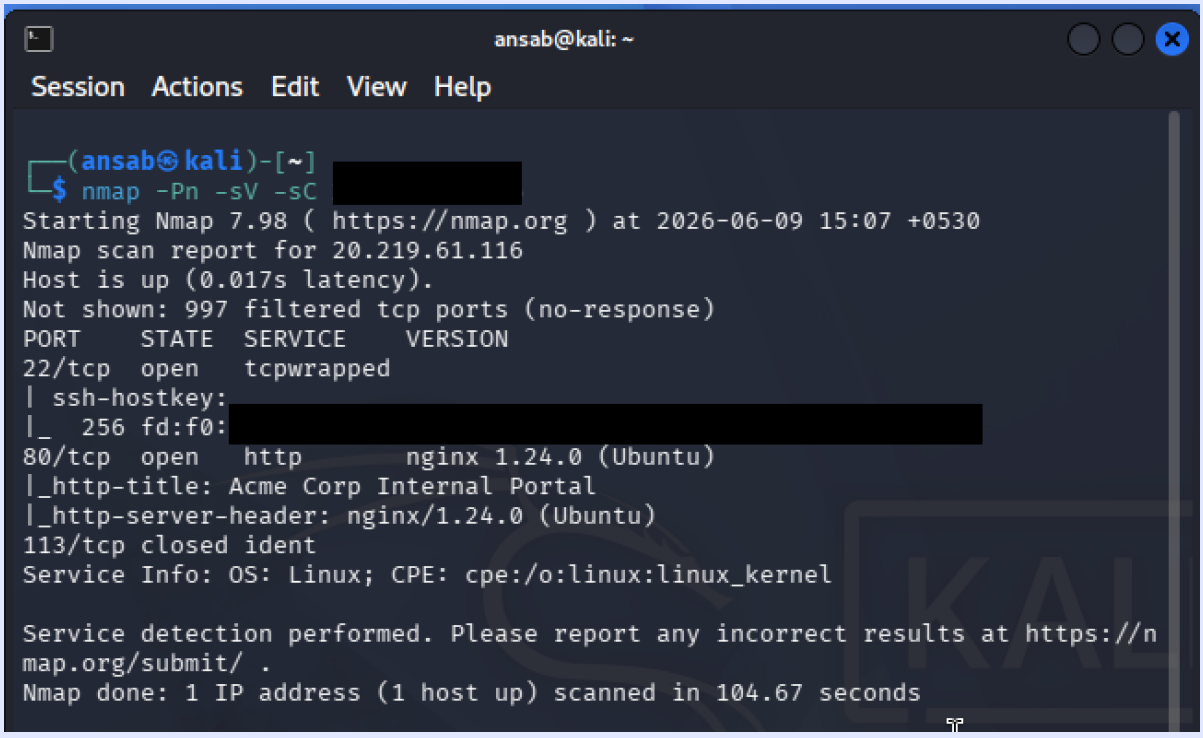
Nmap port and service scan representing the reconnaissance phase -- the attacker maps exposed services before planning further action.

```
nmap -Pn -sV -sC TARGET_PUBLIC_IP
```

The -Pn flag skips ICMP ping since the VM blocks ICMP by default. Scan identified open ports 22 (SSH) and 80 (HTTP Nginx).

[SCREENSHOT SS-11]

Kali terminal -- Nmap results showing open ports 22 and 80 with service versions



7.3 Attack 2 -- Web Vulnerability Scan (Nikto)

Nikto sends hundreds of HTTP probes to identify misconfigurations, exposed files, and known vulnerabilities. Every probe generates an Nginx access log entry.

```
nikto -h http://TARGET_PUBLIC_IP
```

[SCREENSHOT SS-12]

Kali terminal -- Nikto scan running against lab-web-vm

```
┌──(ansab㉿kali)-[~]
└─$ nikto -h http://[redacted]
- Nikto v2.6.0
---------------------------------------------------------------------------
+ Target IP:          [redacted]
+ Target Hostname:
+ Target Port:        80
+ Platform:           Unknown
+ Start Time:         2026-06-09 15:14:27 (GMT5.5)
---------------------------------------------------------------------------
+ Server: nginx/1.24.0 (Ubuntu)
+ ERROR: Failed to check for updates: 403
+ No CGI Directories found (use '-C all' to force check all possible dirs). C
GI tests skipped.
+ [013587] /: Suggested security header missing: strict-transport-security. S
ee: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transpor
t-Security
+ [013587] /: Suggested security header missing: referrer-policy. See: https:
//developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referrer-Policy
+ [013587] /: Suggested security header missing: content-security-policy. See
: https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP
+ [013587] /: Suggested security header missing: permissions-policy. See: htt
ps://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Permissions-Policy
+ [013587] /: Suggested security header missing: x-content-type-options. See:
 https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Opt
ions
^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^[[A^
[[A^[[A


+ [007342] /: X-Frame-Options header is deprecated and was replaced with the
Content-Security-Policy HTTP header with the frame-ancestors directive. See:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/X-Frame-O
ptions
+ [007352] /: The X-Content-Type-Options header is not set. This could allow
the user agent to render the content of the site in a different fashion to th
e MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulner
abilities/missing-content-type-header/
+ 8218 requests: 0 errors and 7 items reported on the remote host
+ End Time:           2026-06-09 15:21:04 (GMT5.5) (397 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

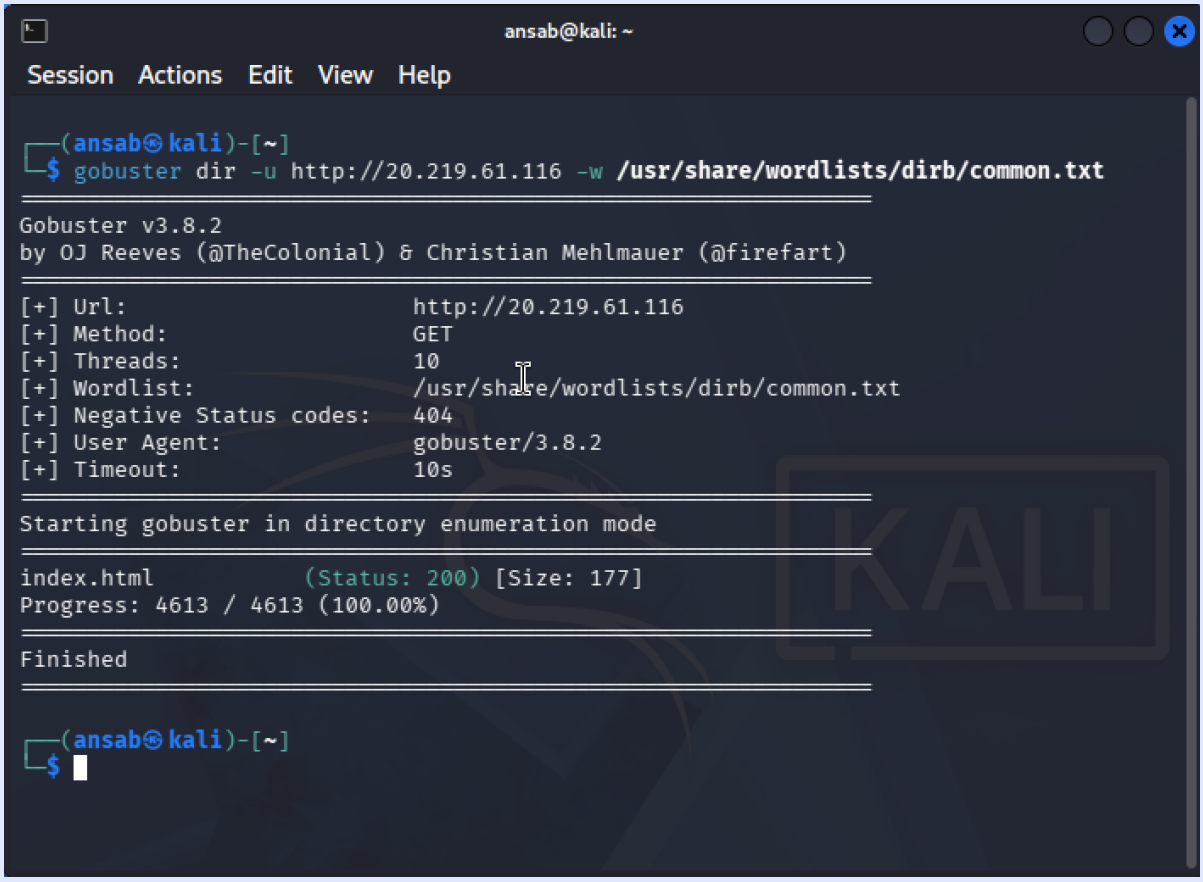
7.4 Attack 3 -- Directory Enumeration (Gobuster)

Gobuster brute-forces common web paths to discover hidden directories, admin panels, and configuration files.

```
gobuster dir -u http://TARGET_PUBLIC_IP -w /usr/share/wordlists/dirb/common.txt
```

[SCREENSHOT SS-13]

Kali terminal -- Gobuster directory enumeration against the web server



7.5 Attack 4 -- SSH Authentication Failure Simulation

Since the VM uses SSH key authentication with password auth disabled, Hydra cannot be used directly. A structured bash loop of SSH connection attempts with invalid usernames was used instead. Each attempt generates an 'Invalid user' entry in /var/log/auth.log which Sentinel ingests and the brute force rule detects.

```
for i in {1..20}; do
  ssh -o StrictHostKeyChecking=no -o ConnectTimeout=5 wronguser$i@TARGET_PUBLIC_IP
```

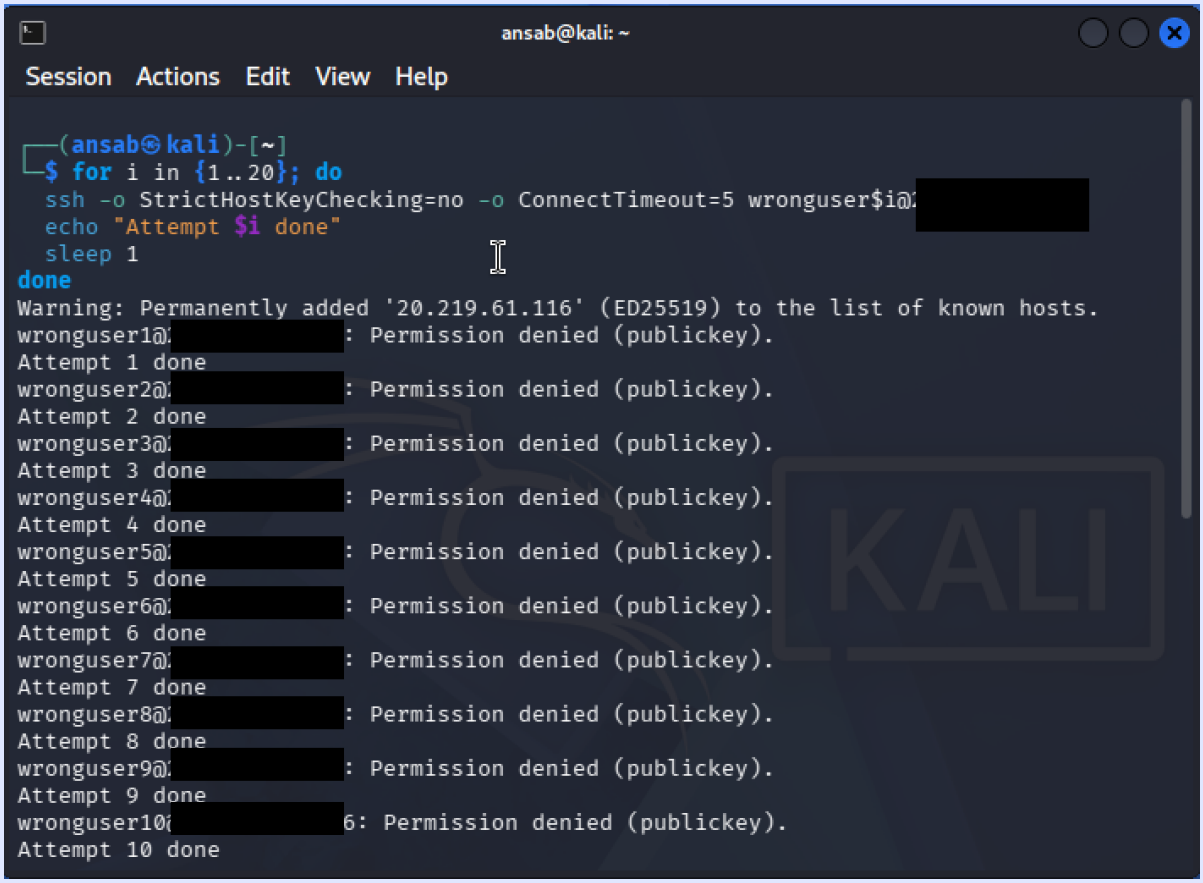
```
  echo "Attempt $i done"
  sleep 1
done
```

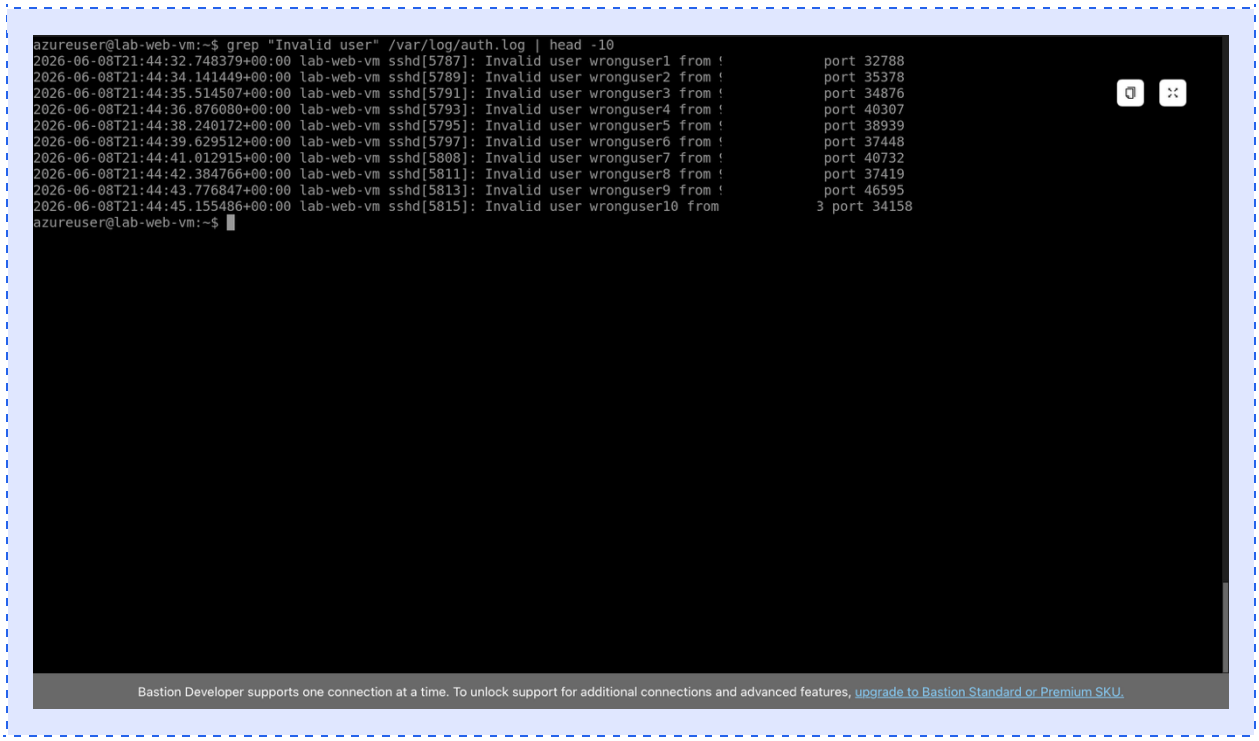
Verified directly on the server:

```
sudo tail -20 /var/log/auth.log
```

[SCREENSHOT SS-14]

lab-web-vm Bastion terminal -- auth.log showing Invalid user wronguser1-20 from Kali IP




azureuser@lab-web-vm:~$ grep "Invalid user" /var/log/auth.log | head -10
2026-06-08T21:44:32.748379+00:00 lab-web-vm sshd[5787]: Invalid user wronguser1 from 9 port 32788
2026-06-08T21:44:34.141449+00:00 lab-web-vm sshd[5789]: Invalid user wronguser2 from 9 port 35378
2026-06-08T21:44:35.514507+00:00 lab-web-vm sshd[5791]: Invalid user wronguser3 from 9 port 34876
2026-06-08T21:44:36.876080+00:00 lab-web-vm sshd[5793]: Invalid user wronguser4 from 9 port 40307
2026-06-08T21:44:38.240172+00:00 lab-web-vm sshd[5795]: Invalid user wronguser5 from 9 port 38939
2026-06-08T21:44:39.629512+00:00 lab-web-vm sshd[5797]: Invalid user wronguser6 from 9 port 37448
2026-06-08T21:44:41.012915+00:00 lab-web-vm sshd[5808]: Invalid user wronguser7 from 9 port 40732
2026-06-08T21:44:42.384766+00:00 lab-web-vm sshd[5811]: Invalid user wronguser8 from 9 port 37419
2026-06-08T21:44:43.776847+00:00 lab-web-vm sshd[5813]: Invalid user wronguser9 from 9 port 46595
2026-06-08T21:44:45.155486+00:00 lab-web-vm sshd[5815]: Invalid user wronguser10 from 3 port 34158
azureuser@lab-web-vm:~$
Bastion Developer supports one connection at a time. To unlock support for additional connections and advanced features, upgrade to Bastion Standard or Premium SKU.

8. Detection and Incident Response

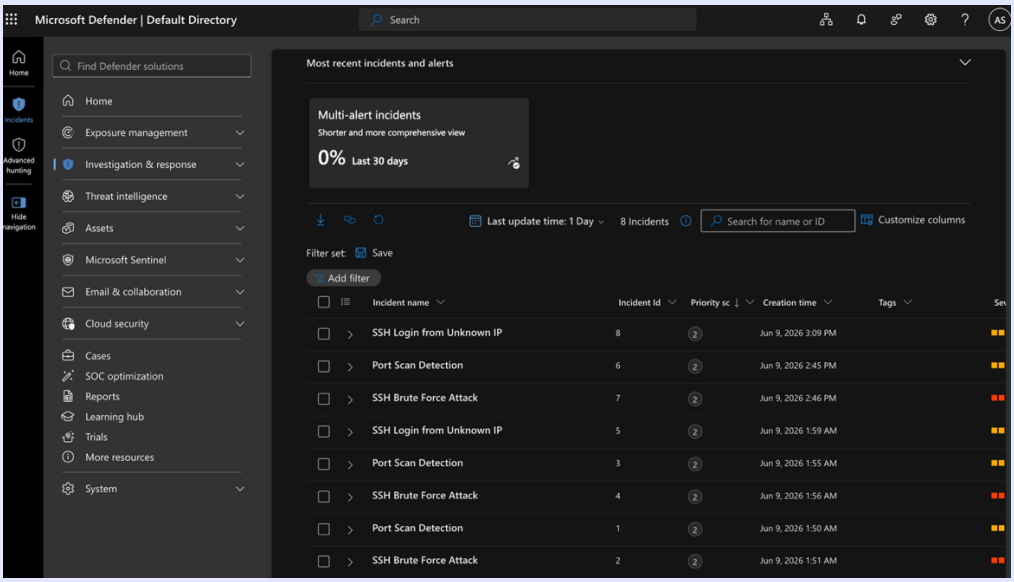
8.1 Confirmed Detections

Incident	Rule Triggered	Severity	Source	Status
Port Scan Detected	Port Scan Detection	Medium	Kali (92.97.201.83)	Confirmed -- incident raised automatically
SSH Brute Force Attack	SSH Brute Force Attack	High	Kali (92.97.201.83)	Confirmed -- incident raised automatically
Firewall Rule Changed	Firewall Rule Changed	High	Admin portal action	Confirmed -- triggered by test NSG rule change

Note on Incident Entities
When investigating incidents in Sentinel, the Entities panel shows the affected host (lab-web-vm). The attacker source IP is visible in the raw evidence tab and KQL query results -- not in the destination IP field of entities, which reflects internal VM addressing.

[SCREENSHOT SS-15]

Microsoft Sentinel -- Incidents page showing all triggered incidents with severity and timestamps

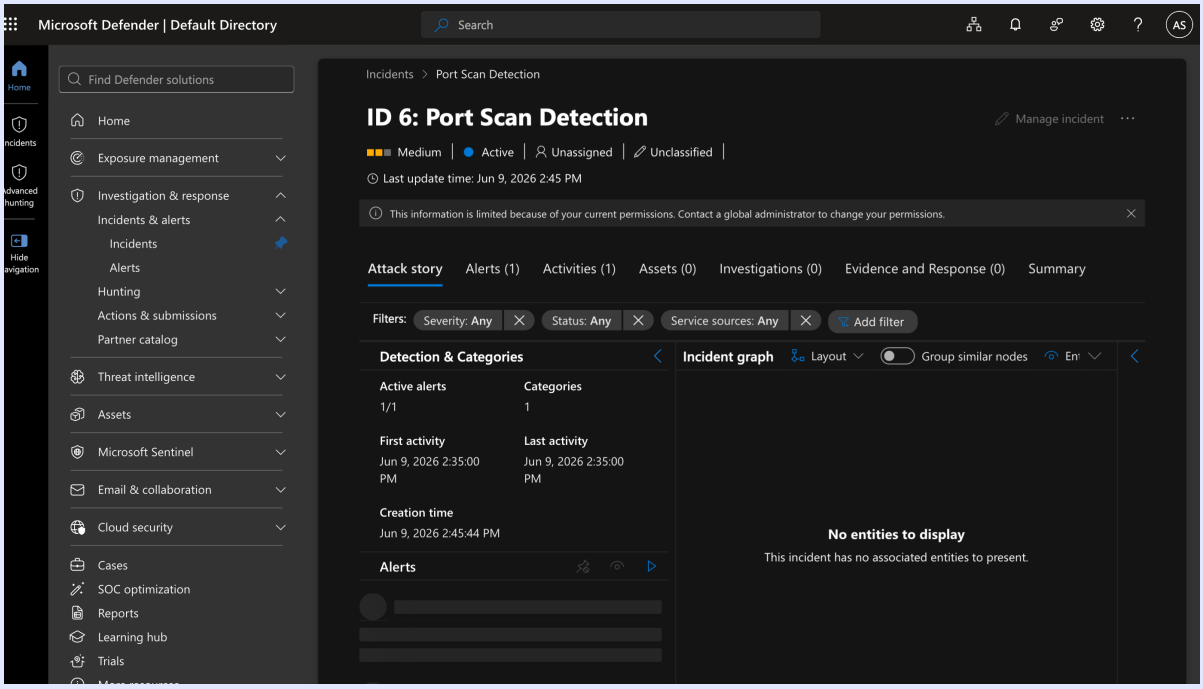


8.2 Port Scan Incident

Sentinel correlated multiple denied connection attempts from Kali IP 92.97.201.83 across different ports and raised a Medium severity incident automatically.

[SCREENSHOT SS-16]

Microsoft Sentinel -- Port Scan incident detail showing entity, timeline, and raw evidence

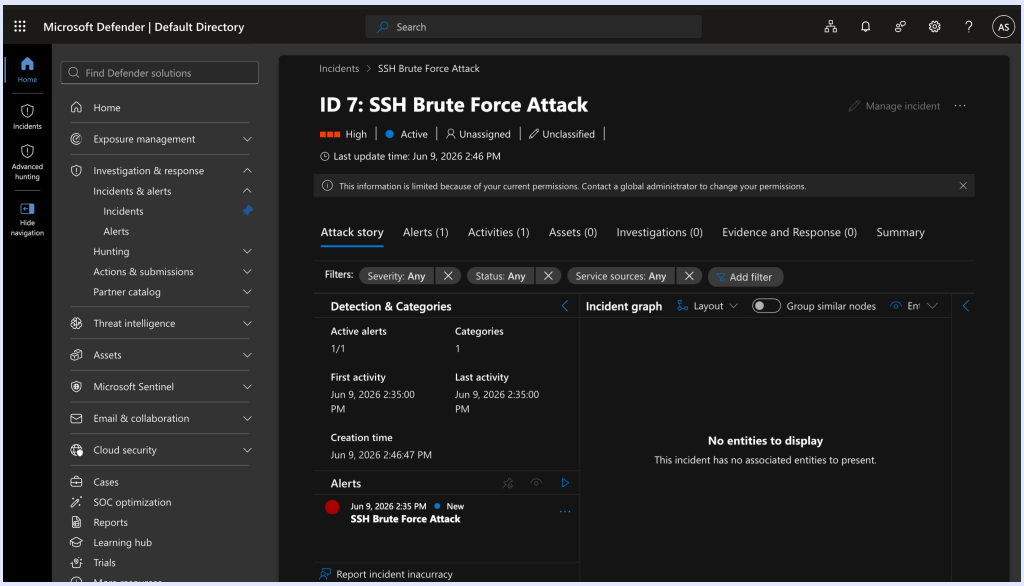


8.3 SSH Brute Force Incident

The brute force incident was raised after the invalid user loop generated more than 5 failed authentication events within a 5-minute window.

[SCREENSHOT SS-17]

Microsoft Sentinel -- SSH Brute Force incident showing FailureCount, source IP, and affected computer

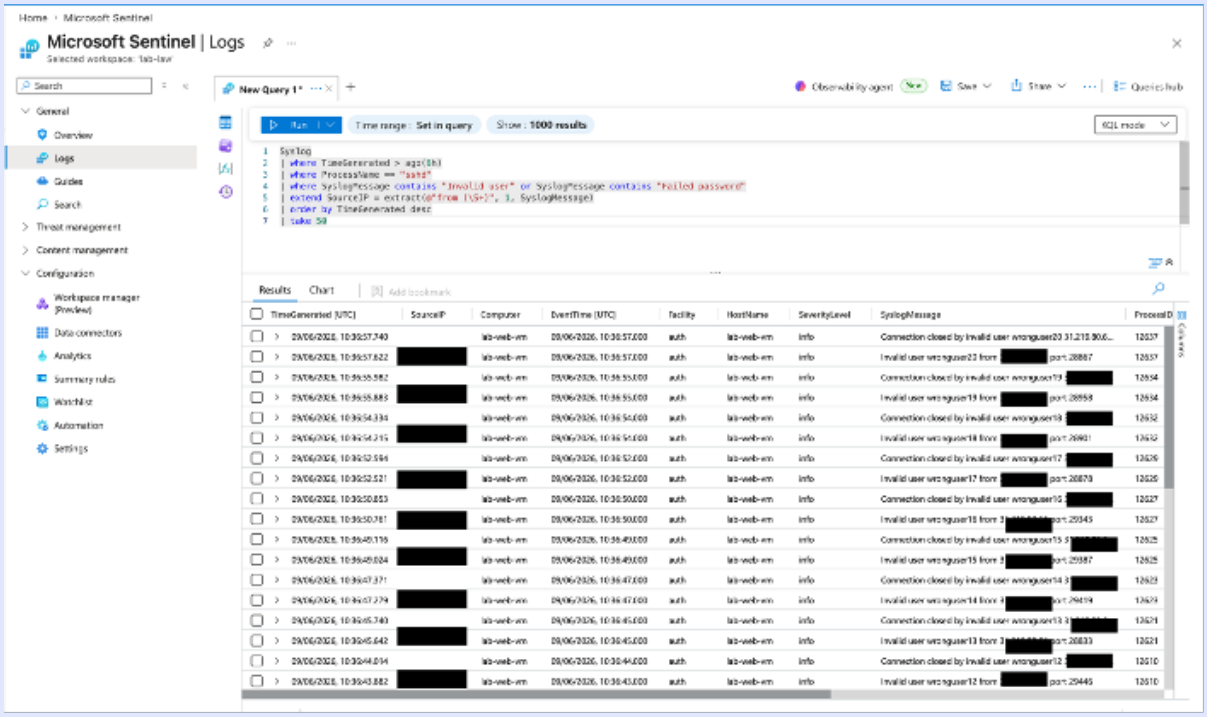


8.4 Log Evidence

```
Syslog
| where TimeGenerated > ago(1h)
| where ProcessName == "sshd"
| where SyslogMessage contains "Invalid user" or SyslogMessage contains "Failed password"
| extend SourceIP = extract(@"from (\S+)", 1, SyslogMessage)
| order by TimeGenerated desc | take 50
```

[SCREENSHOT SS-18]

Sentinel Logs -- KQL results showing failed SSH attempts from Kali IP



9. Jump Server -- Secure Administrative Access

9.1 Overview

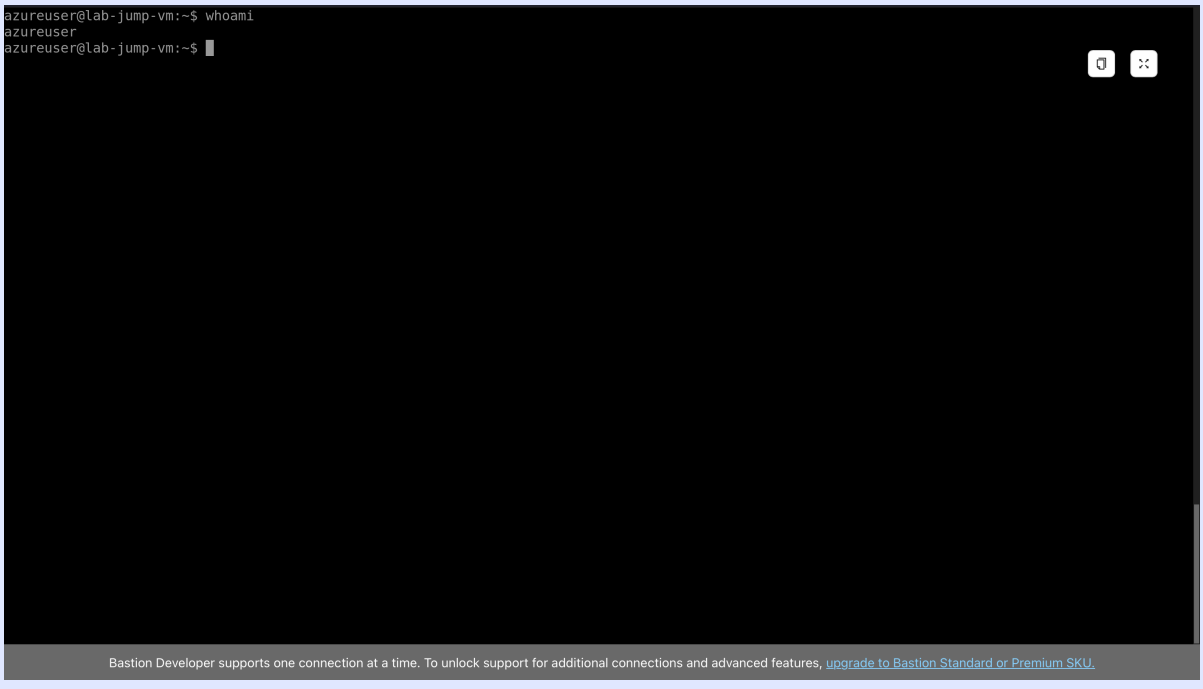
> **Security Principle**
> The jump server pattern reduces the attack surface to a single hardened entry point and creates a centralised audit trail. It is standard practice in enterprise environments and aligns with zero-trust access principles.

9.2 Administrative Access Flow

1. Open Azure Portal and navigate to lab-jump-vm.
2. Click Connect then Bastion -- a browser-based terminal opens
3. Authenticate as azureuser using SSH private key (lab-jump-key.pem)
4. The Bastion session lands on lab-jump-vm in the DMZ subnet
5. From the jump server terminal, administrative access to internal VMs is conducted over private IP addressing. The jump server acts as the single audited entry point into the internal network, with all sessions authenticated through Azure Bastion.
6. All administrative work on lab-web-vm is conducted through this audited session

[SCREENSHOT SS-19]

Azure Bastion terminal -- browser-based SSH session connected to lab-jump-vm via Azure Bastion, showing secure administrative access without public IP exposure



10. Cost Management

10.1 Resource Cost Overview

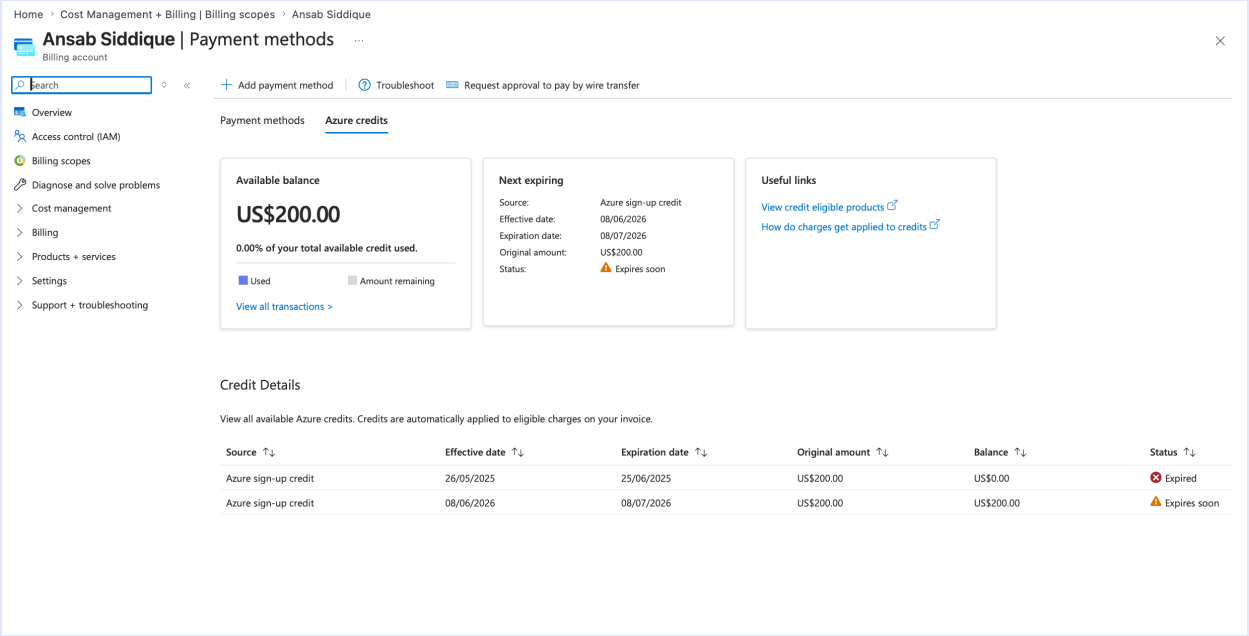
Resource	Size / SKU	Running Cost	Deallocated Cost
lab-jump-vm	B2ats_v2 (2 vCPU, 4GB)	~$0.04/hour	$0.00
lab-web-vm	B2ats_v2 (2 vCPU, 4GB)	~$0.04/hour	$0.00
Azure Bastion	Basic SKU	~$0.19/hour	N/A
Log Analytics Workspace	Pay-per-GB	Free up to 5GB/month	N/A
Microsoft Sentinel	Pay-per-GB	Free for first 31 days	N/A
Storage Account (labflowlogs1998)	LRS Standard	~$0.02/month	N/A

10.2 Cost Control Measures

- VMs deallocated after every session -- zero compute cost when not in use
- B2ats_v2 was the smallest available VM size in Central India at deployment time
- Standard LRS storage for NSG flow logs -- cheapest redundancy tier
- Traffic Analytics disabled -- avoids additional per-GB processing charge
- Public IP dissociated from lab-web-vm after each attack simulation session
- Log Analytics ingestion monitored to stay under the 5GB free monthly threshold

[SCREENSHOT SS-20]

Azure Cost Management -- total spend and remaining free credit balance



11. Challenges and Solutions

Challenge	Root Cause	Solution
B-series VMs (B1ls, B1s, B1ms) unavailable across all regions	High demand on free tier quotas and regional capacity constraints at deployment time	Used B2ats_v2 (2 vCPU, 4GB) which was available in Central India. Total vCPU usage of 4 matched the regional quota exactly with no increase needed.
NSG for jump server missing	Azure did not auto-create an NSG when the VM was deployed with None selected for NSG	Manually created lab-jump-nsg and attached it to the VM network interface via NIC settings
Flow log creation failing with template validation error	ARM template required TrafficAnalyticsEnabled parameter even when Traffic Analytics set to off	Used Azure Network Watcher to create flow logs -- the Network Watcher UI handles the parameter correctly
SSH brute force tool cannot target the VM	VM uses SSH key authentication -- password authentication is disabled by default on Azure Linux VMs	Used a bash loop of SSH connection attempts with invalid usernames, generating sshd Invalid user events that trigger the Sentinel brute force rule
Sentinel Analytics redirected to Defender portal	Microsoft migrated Sentinel Analytics to the unified Microsoft Defender portal	Used security.microsoft.com with the correct tenant ID parameter appended to the URL
Defender portal authentication failed	Personal Microsoft account not recognised in the Microsoft Services tenant	Accessed directly via URL with correct tenant ID: security.microsoft.com/sentinel/analytics?tid=3aee1a93-5a92-4ce0-8a2e-860e4c1113dd
AzureNetworkAnalytics_CL table missing	Traffic Analytics disabled so the table was not populated	Rewrote port scan detection rule to use Syslog table connection count patterns instead
Azure Activity connector showing not connected	Policy-based connectors require a manual remediation task after policy assignment	Created remediation task via Azure Policy > Assignments. Connector expected to connect within 30 minutes of task completion.

12. Skills Demonstrated

12.1 Microsoft Azure

- Created and managed Resource Groups, Virtual Networks, and Subnets
- Deployed and configured Linux VMs with custom sizing and subnet placement
- Implemented Azure Bastion for secure browser-based VM access
- Configured Network Security Groups with layered inbound and outbound rules
- Enabled NSG Flow Logs via Azure Network Watcher
- Managed VM lifecycle: start, stop, deallocate, and resize
- Configured static public IPs and dynamically associated and dissociated them

12.2 Network Security

- Designed a two-VM network with DMZ and web tier separation
- Implemented deny-all default NSG rules with explicit permit rules in priority order
- Applied principle of least privilege to all network access controls
- Implemented jump server pattern for secure administrative access
- Documented rationale behind each NSG rule for audit purposes

12.3 SIEM -- Microsoft Sentinel

- Deployed Microsoft Sentinel on a Log Analytics Workspace
- Connected 9 of 10 data sources using built-in connectors
- Created Data Collection Rules for Linux VM log forwarding via Azure Monitor Agent
- Wrote 4 custom threat detection rules in KQL
- Verified log ingestion using KQL queries before creating detection rules
- Investigated automatically created incidents with full evidence review

12.4 Offensive Security (Red Team)

- Operated Kali Linux as an external attacker simulation platform via UTM
- Conducted network reconnaissance using Nmap with service and version detection
- Performed web vulnerability scanning using Nikto
- Executed directory enumeration using Gobuster
- Simulated SSH authentication failures using a structured bash loop
- Verified attack traffic directly in VM system logs before Sentinel verification

13. Future Enhancements

Enhancement	Description	Certification Alignment
Azure Activity Connector	Complete remediation task to connect Azure Activity and enable Rule 4 with live data	AZ-500, SC-200
Windows Server VM	Add a third VM running Windows Server 2022 for Windows event log analysis	SC-200, AZ-500
Azure Defender for Servers	Enable Defender for Servers for file integrity monitoring and vulnerability assessment	SC-200
Sentinel Playbooks (SOAR)	Logic App playbooks for automated incident response -- auto-block IP on brute force detection	SC-200
Sentinel Workbook Dashboard	Custom workbook showing attack trends, top attacker IPs, and incident statistics	SC-200
CompTIA Security+ Alignment	Map all lab activities to Security+ domain objectives for study validation	Security+

14. Conclusion

This project successfully demonstrates the design, implementation, and validation of an enterprise-grade cloud security lab using Microsoft Azure. All primary objectives were achieved:

- A two-VM network architecture built with DMZ segmentation and deny-all default security rules
- Microsoft Sentinel deployed and connected to 9 of 10 data sources with 4 custom KQL detection rules
- Real attack simulations from Kali Linux triggered automatic incidents in Sentinel
- The complete attack, detect, and investigate cycle demonstrated end to end
- All technical challenges identified, documented, and resolved
- Entire lab built within the Azure free tier with minimal spend

The lab provides verifiable evidence of hands-on capability across Azure administration, network security, SIEM operations, and offensive security tooling. It complements formal certifications (CCNA, ISC2 CC, AWS CCP) with practical demonstration.

> **Portfolio**
> This project is documented at ansabsid.github.io alongside The Kill Chain (MITRE ATT&CK guide) and How a Machine Learns to Think (AI fundamentals platform).

Appendix A -- Screenshot Reference

ID	Description	Where to Take It
SS-01	Network topology	Azure Portal > Network Watcher > Topology > lab-security-rg
SS-02	Resource Group overview	Azure Portal > Resource Groups > lab-security-rg > Overview
SS-03	lab-jump-vm overview	Azure Portal > Virtual Machines > lab-jump-vm > Overview
SS-04	lab-web-vm overview	Azure Portal > Virtual Machines > lab-web-vm > Overview
SS-05	Web VM NSG rules	NSGs > lab-web-vm329-nsg > Inbound security rules
SS-06	Jump VM NSG rules	NSGs > lab-jump-nsg > Inbound security rules
SS-07	Bastion terminal	lab-jump-vm > Connect > Bastion > Connect
SS-08	Sentinel data connectors	security.microsoft.com > Sentinel > Data Connectors
SS-09	KQL log verification	Sentinel > Logs > run Syslog summarize query
SS-10	Sentinel analytics rules	security.microsoft.com > Sentinel > Analytics
SS-11	Nmap scan output	Kali > nmap -Pn -sV -sC TARGET_PUBLIC_IP
SS-12	Nikto scan running	Kali > nikto -h http://TARGET_PUBLIC_IP
SS-13	Gobuster running	Kali > gobuster dir command
SS-14	auth.log showing attacks	lab-web-vm Bastion > sudo tail -50 /var/log/auth.log
SS-15	Sentinel incidents page	security.microsoft.com > Sentinel > Incidents
SS-16	Port scan incident	Click Port Scan incident > entities and timeline
SS-17	SSH brute force incident	Click SSH Brute Force incident > details

SS-18	KQL evidence query	Sentinel > Logs > sshd Invalid user query
SS-19	Jump server SSH	Bastion > Jump Server
SS-20	Cost Management	Azure Portal > Cost Management > Credits

Appendix B -- Tools and Technologies

Tool / Service	Category	Version / SKU	Purpose
Microsoft Azure	Cloud Platform	Free Trial	Infrastructure hosting
Microsoft Sentinel	SIEM	Unified SecOps Portal	Threat detection and incident management
Log Analytics Workspace	Log Management	lab-law	Log storage and KQL query engine
Azure Bastion	Secure Access	Basic SKU	Browser-based VM access without public IPs
Ubuntu Server	Operating System	22.04 LTS	VM OS for both lab VMs
Nginx	Web Server	Latest (apt)	HTTP service on lab-web-vm
Kali Linux	Penetration Testing OS	Latest	Attack simulation platform (UTM on Mac)
Nmap	Network Scanner	v7.94	Port scanning and service detection
Nikto	Web Scanner	v2.1.6	Web server vulnerability scanning
Gobuster	Directory Brute Force	v3.6	Web directory enumeration
Bash	Scripting	Built-in Kali	SSH invalid user loop for brute force simulation
KQL	Query Language	Azure Monitor	SIEM rule authoring and log analysis
UTM	Virtualisation	macOS	Running Kali Linux on Apple Mac

Appendix C -- Author Profile

Mohammad Ansab Siddiqui

Technical Consultant -- ITSM and Solutions | Vistar, Dubai UAE

ansab.sid123@gmail.com | +971 552 552 476 | ansabsid.github.io

Certification	Issuer	Status
Cisco CCNA 200-301	Cisco	Active
AWS Certified Cloud Practitioner	Amazon Web Services	Active
ISC2 Certified in Cybersecurity (CC)	ISC2	Active
Freshservice Certified Product Expert	Freshworks	Active
Critical Infrastructure Protection (ICIP)	OPSWAT	Active
Introduction to Cybersecurity	Cisco NetAcad	Active