Azure Home Lab Project:
Building a Security Monitoring Environment

By Lucio Rodrigues

Introduction

This project demonstrates the end-to-end setup of a security monitoring environment in Microsoft Azure, designed to simulate real-world attack visibility and incident analysis.

By deploying a honeypot virtual machine, forwarding its logs to a central workspace, and integrating Azure Sentinel for threat detection and visualisation, I created a controlled environment to practice SIEM use cases and threat hunting techniques.

The following documentation provides a structured walkthrough of the setup process, including screenshots of each step. It is written to guide others through reproducing the lab while also showcasing the applied knowledge gained from the project.

Completed Project Overview

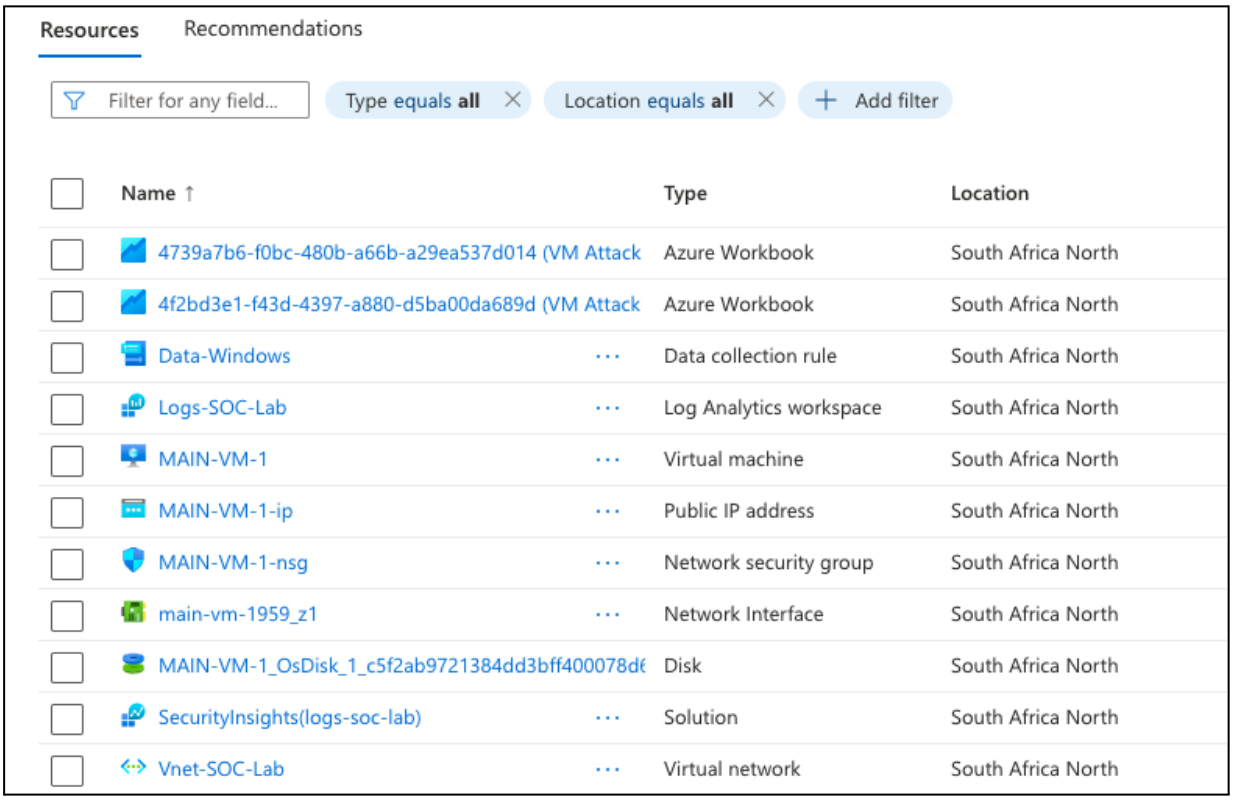
The completed lab environment provides a fully functional security monitoring pipeline within Azure. A honeypot virtual machine was deployed, exposed to the internet, and configured to forward security events into a centralised Log Analytics Workspace.

Azure Sentinel was then layered on top to enable correlation, enrichment with a custom geolocation watchlist, and visualisation of global attack patterns.

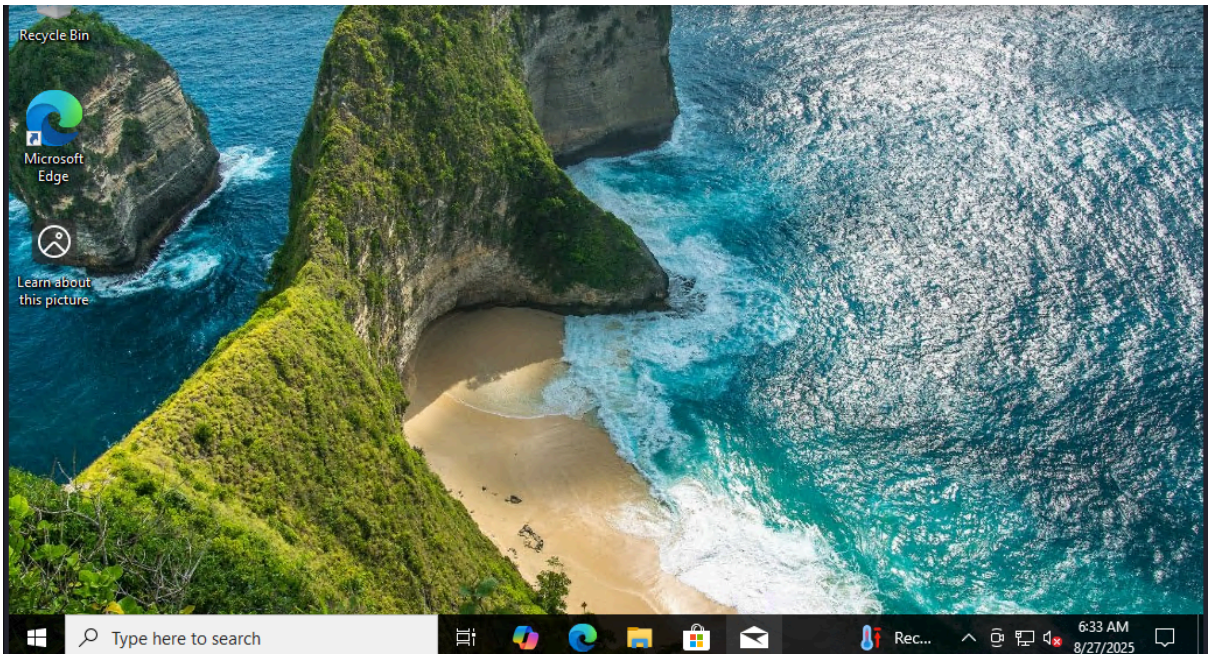
Figures below illustrate the completed setup:

- **Figure 1:** High-level Azure architecture (Resource Group, VM, NSG, Workbook, Log Analytics, Sentinel).

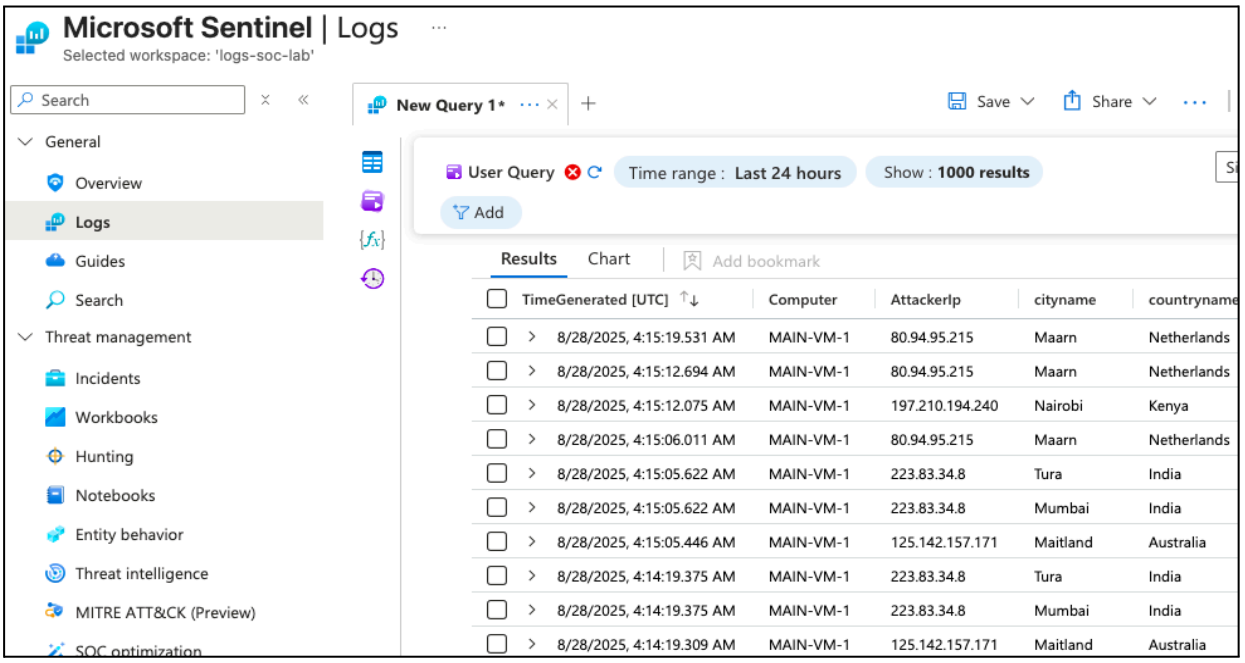
Resource Group with all required resources.



Windows 10 Honeypot VM running on Kali Linux.

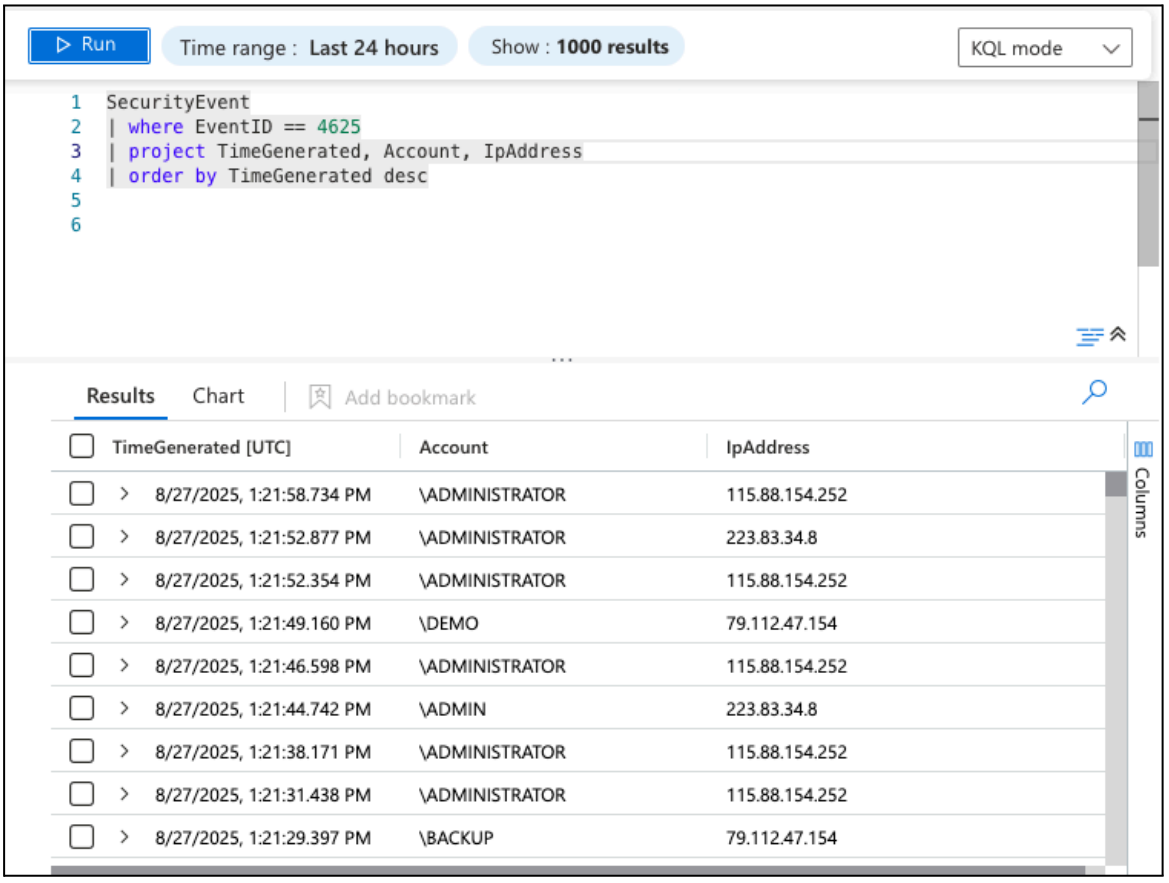


Sentinel setup with ingested logs.

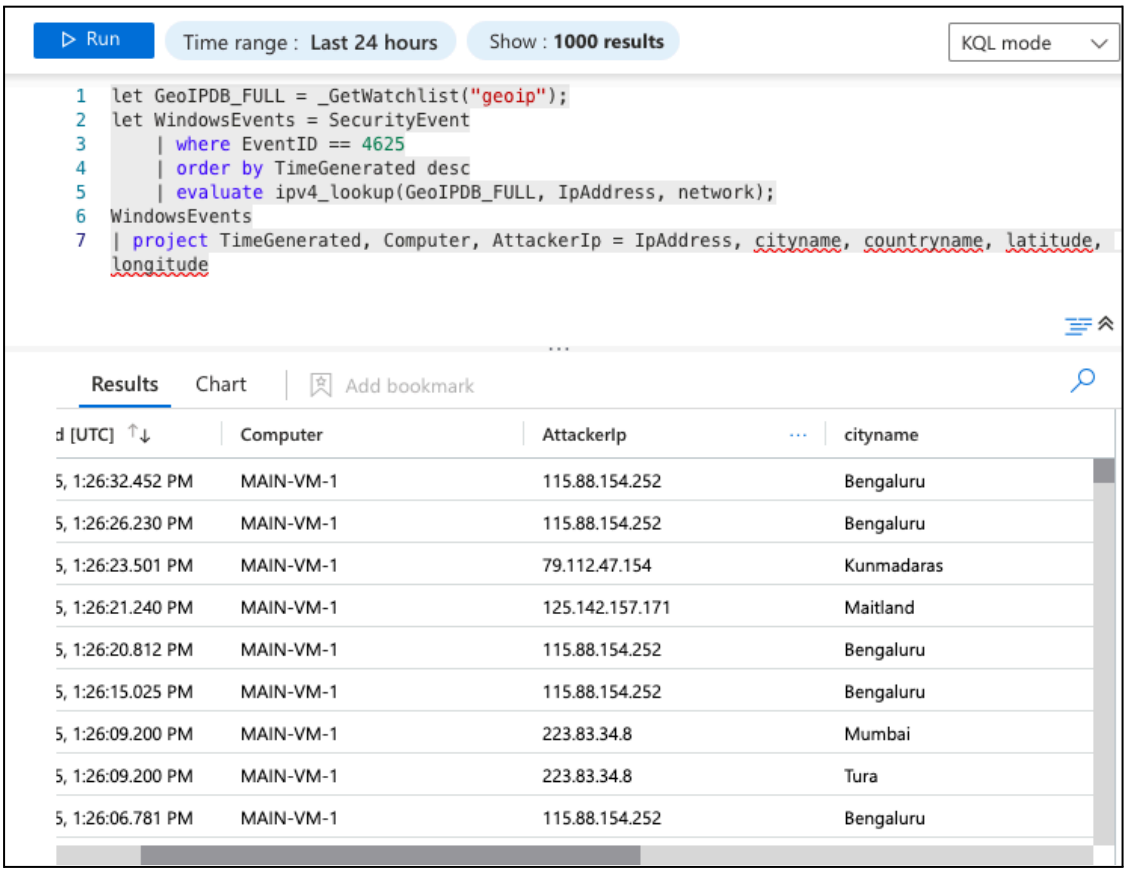


- **Figure 2:** Sample KQL query results.

Query for failed logons (Code = 4625)

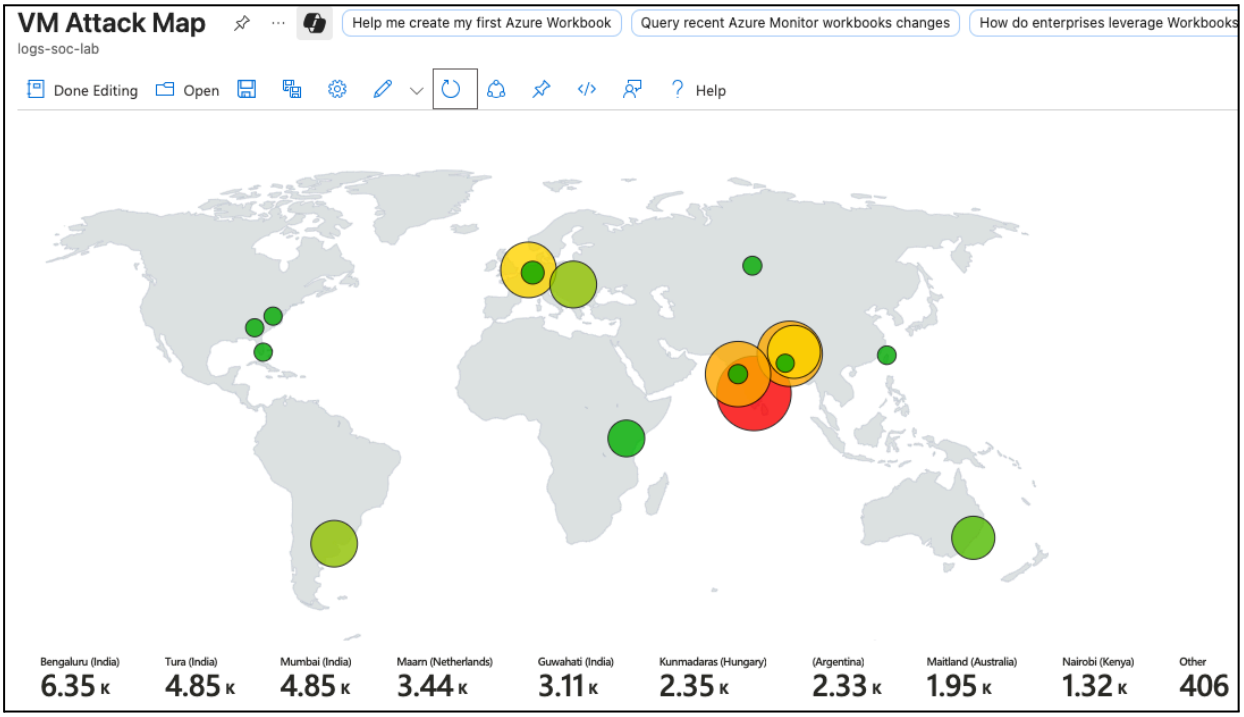


Query to enrich logs with geo-location context.



- **Figure 3:** World map visualisation of attacker IP origins.

  12 hours of attack data.



With the completed environment in mind, let's dive into the step-by-step process of building this lab. The lab shows how to configure a security monitoring environment in Azure, delivering practical experience directly applicable to real-world cybersecurity roles.

---

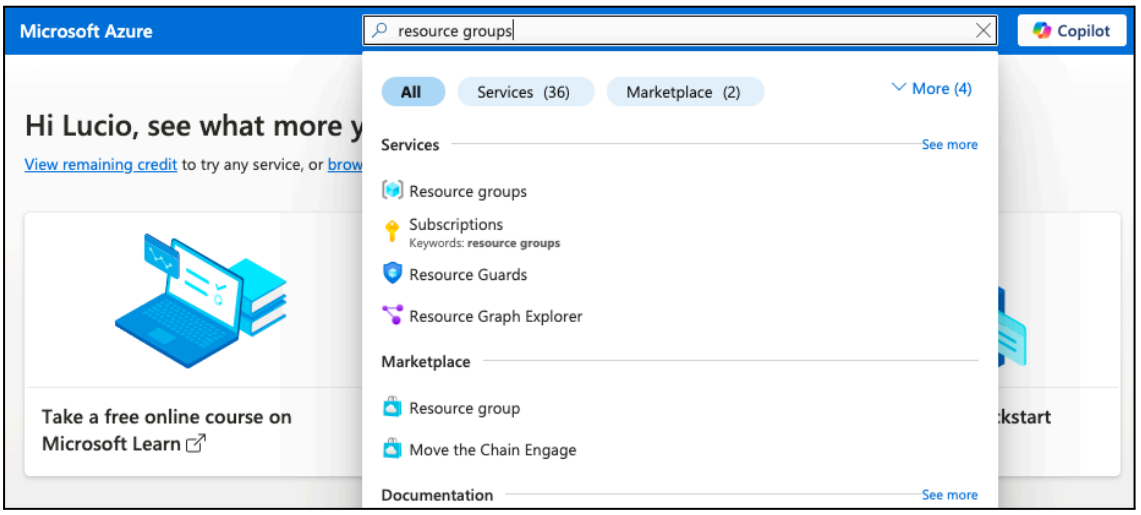
1. Azure Resource Setup

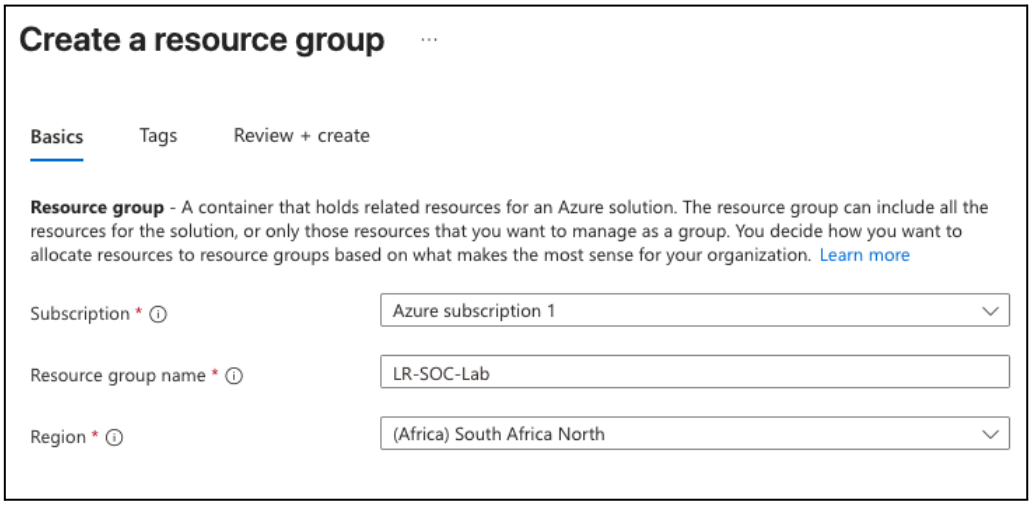
1.1 Resource Group

- Set up a dedicated resource group to logically contain project resources.
- Ensured permissions and policies were properly applied.

Figures:

Create a resource group



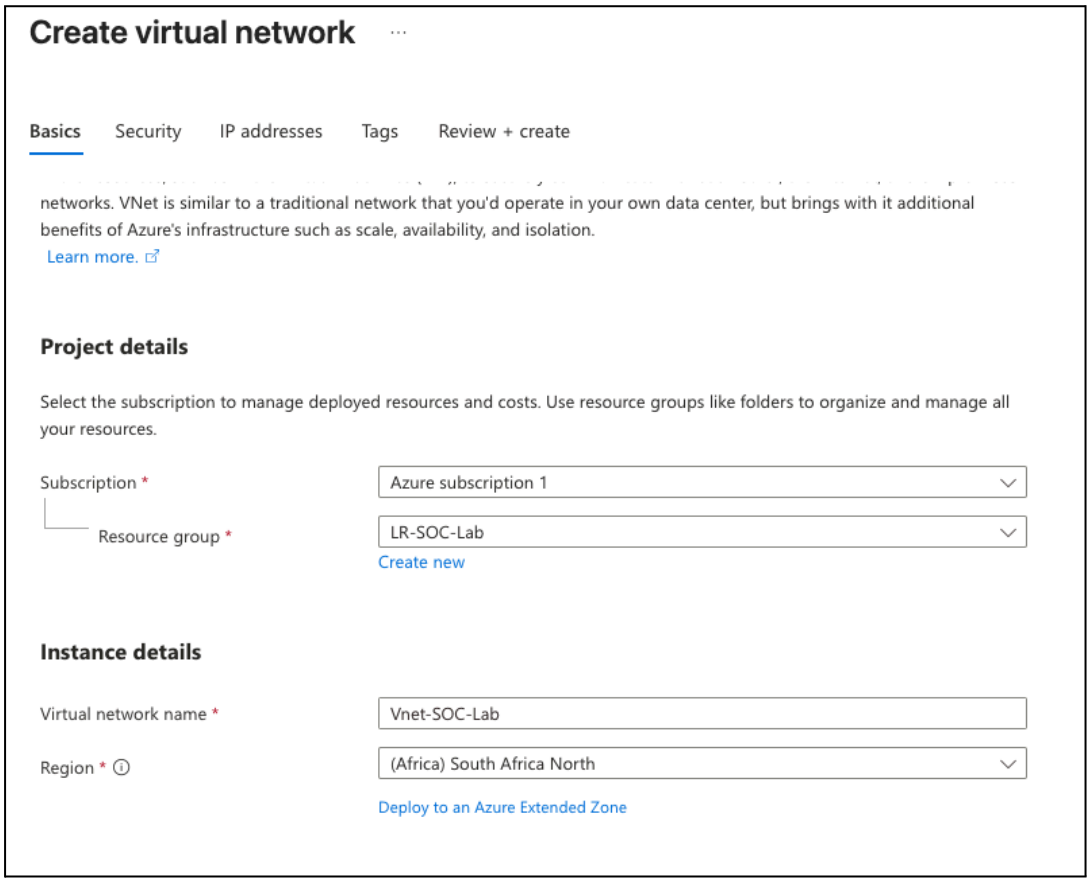


1.2 Virtual Network and Subnet

- Configured a virtual network (VNet) with an associated subnet.
- This subnet serves as the foundation for connecting the virtual machine honeypot.

Figures:

Create a Vnet.



Section Takeaway

This foundational setup ensured all project resources were logically grouped and networked, providing a controlled environment for subsequent monitoring and security configurations.

---

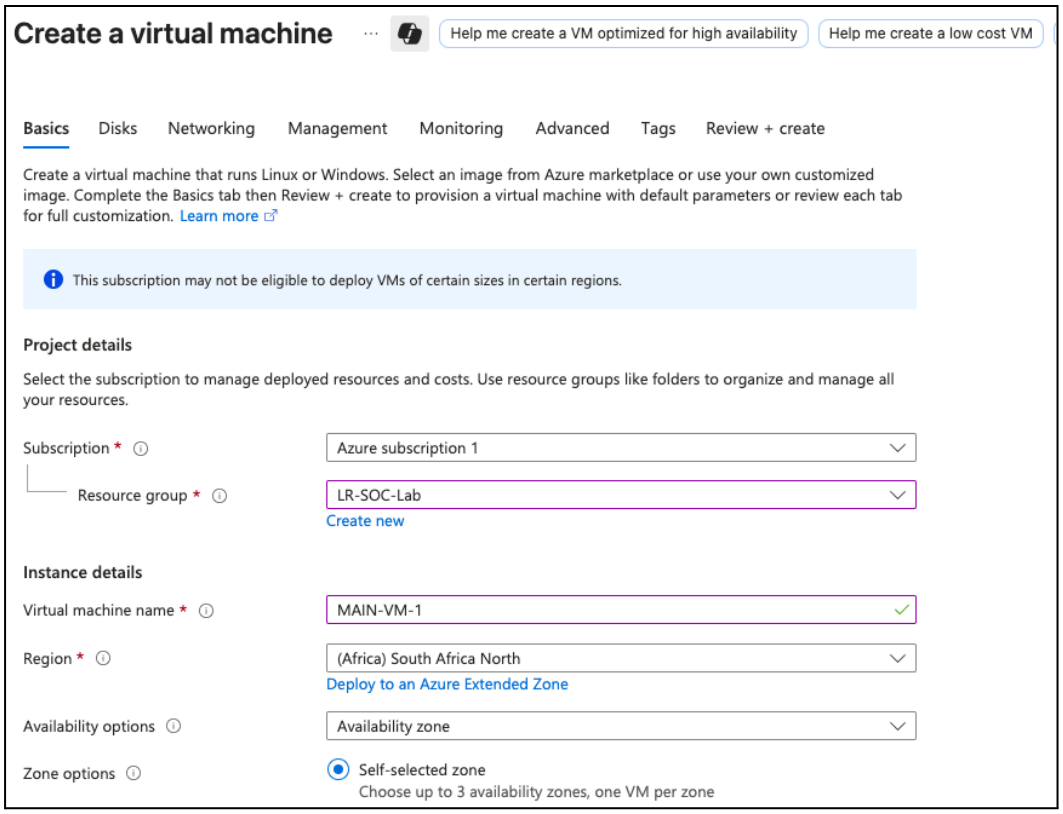
2. Virtual Machine Honeypot

2.1 Deployment

- Deployed a Windows Virtual Machine (VM) within the subnet.
- Configured basic OS settings and ensured accessibility.

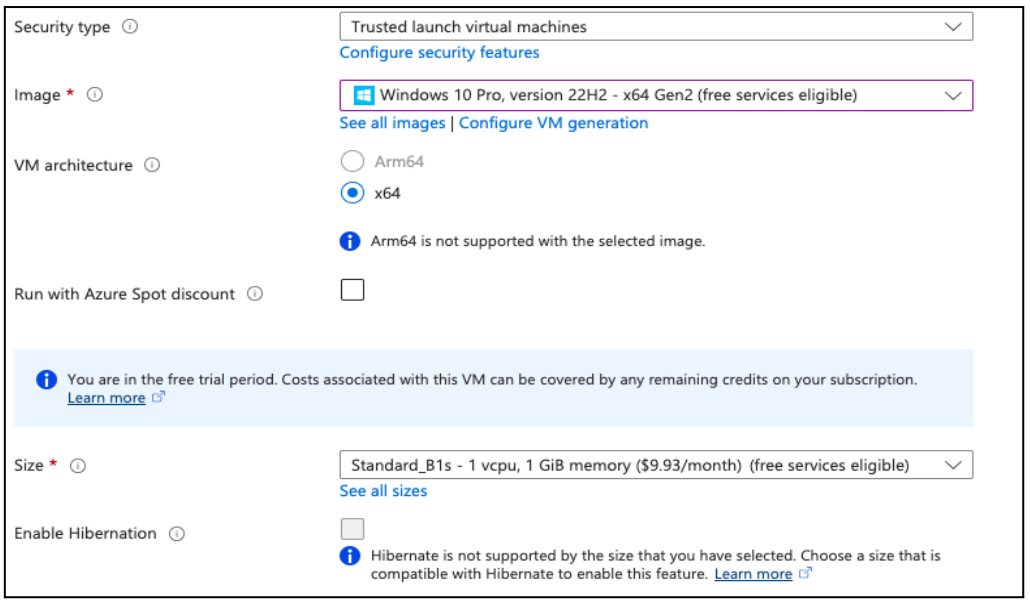
Figures

Create a VM.

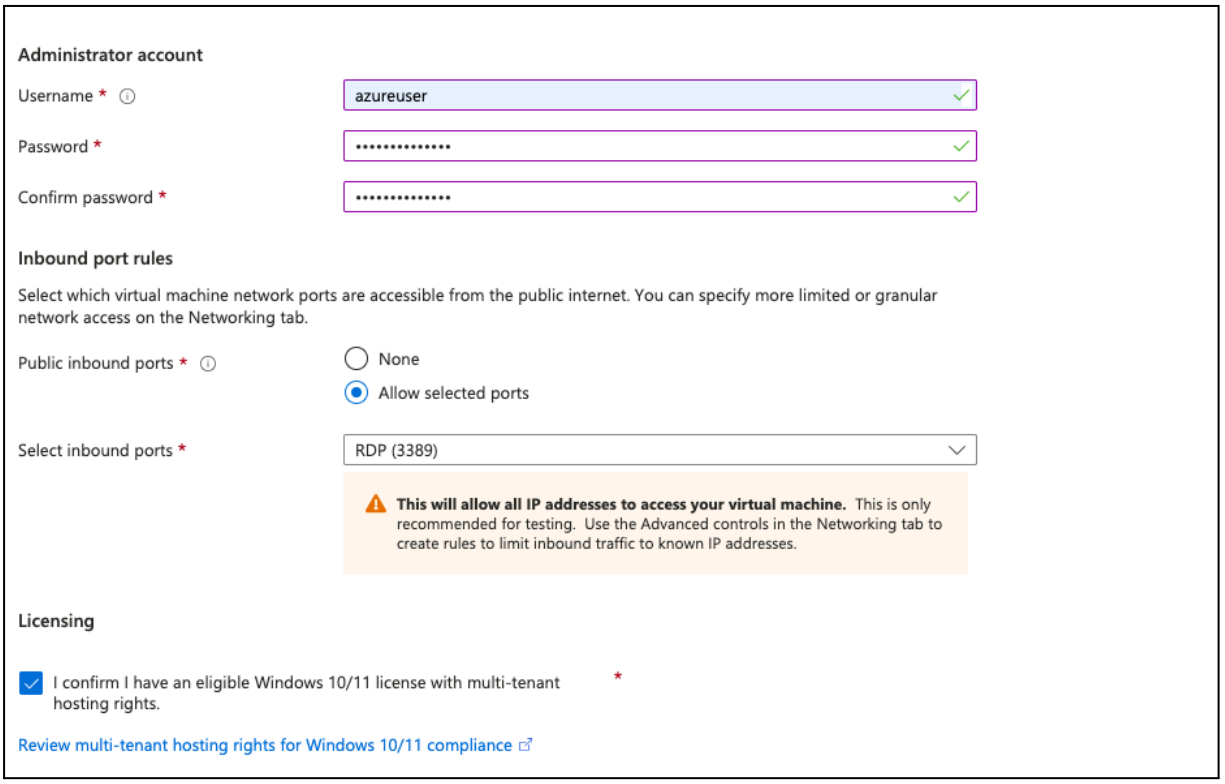


When deploying the virtual machine, Azure provides multiple options for both the operating system image and the VM size.

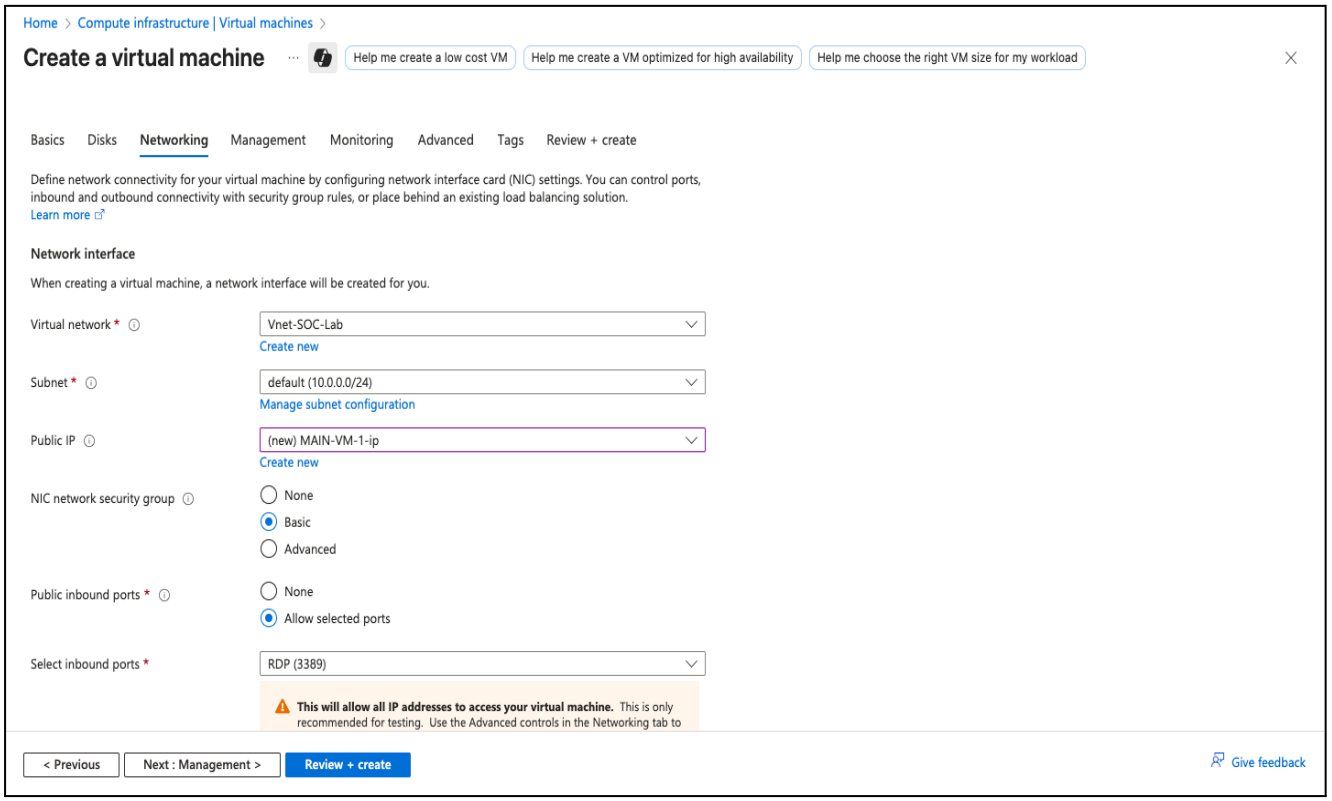
Depending on the use case, different configurations may be more suitable; however, for this lab I selected Windows 10 Pro, version 22H2 - x64 Gen2 as the image and Standard_B1s as the size to balance cost efficiency with functionality.



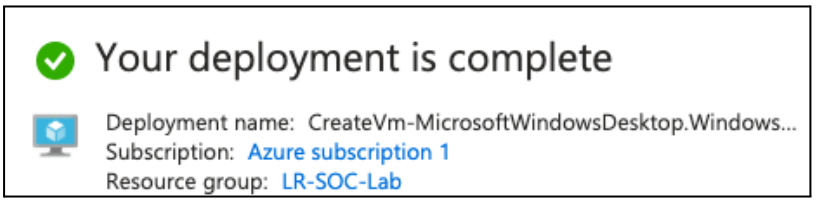
Configure the admin account with a username and password, and allow RDP (Remote Desktop Protocol).



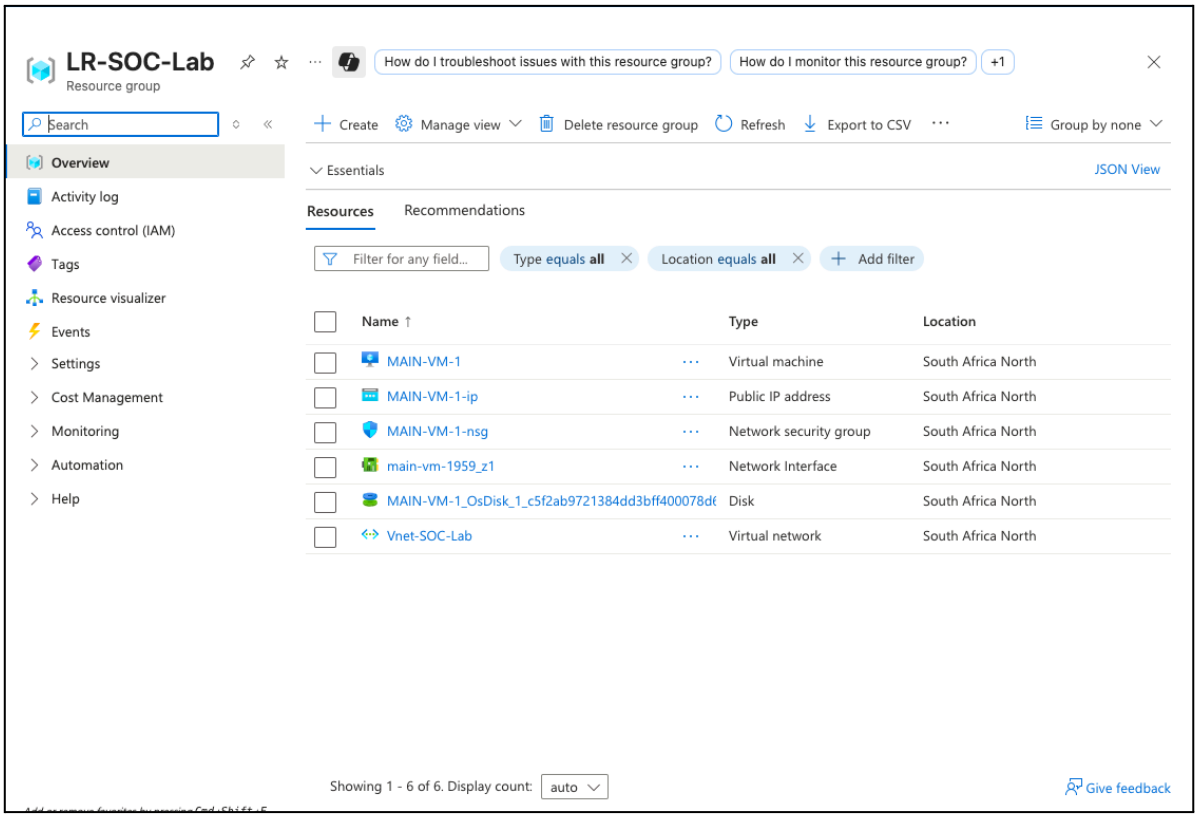
Input a Vnet name. Subnet & Public IP settings are left at default settings.



Complete.



Inside of your resource group you'll find all of your new resources.



2.2 Security Posture (Intentionally Vulnerable)

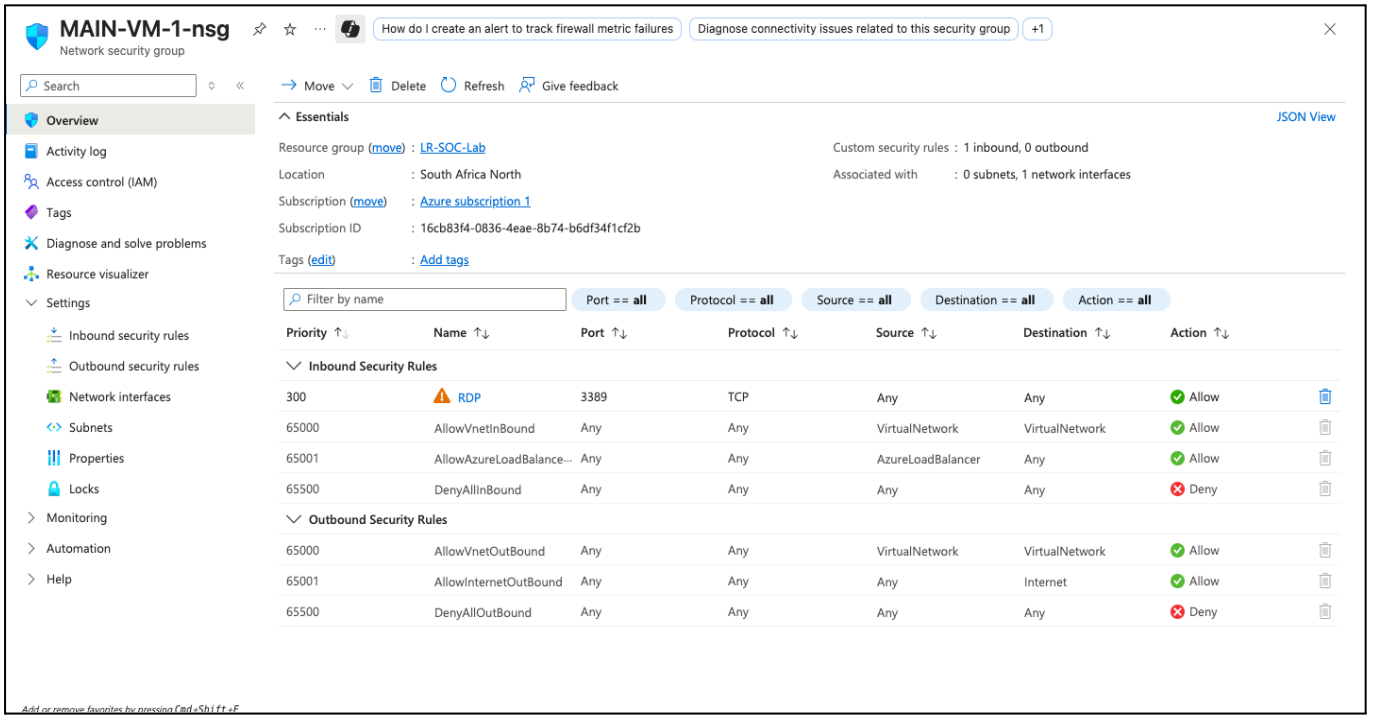
- Opened the Network Security Group (NSG) and cloud firewall completely to the public internet.
- Disabled internal firewalls inside the VM, ensuring maximum exposure.
- This configuration allowed the VM to attract real-world brute-force and probing attempts.

To intentionally expose the virtual machine as a honeypot, I configured the Network Security Group (NSG) with an inbound rule that allowed unrestricted access from the public internet.

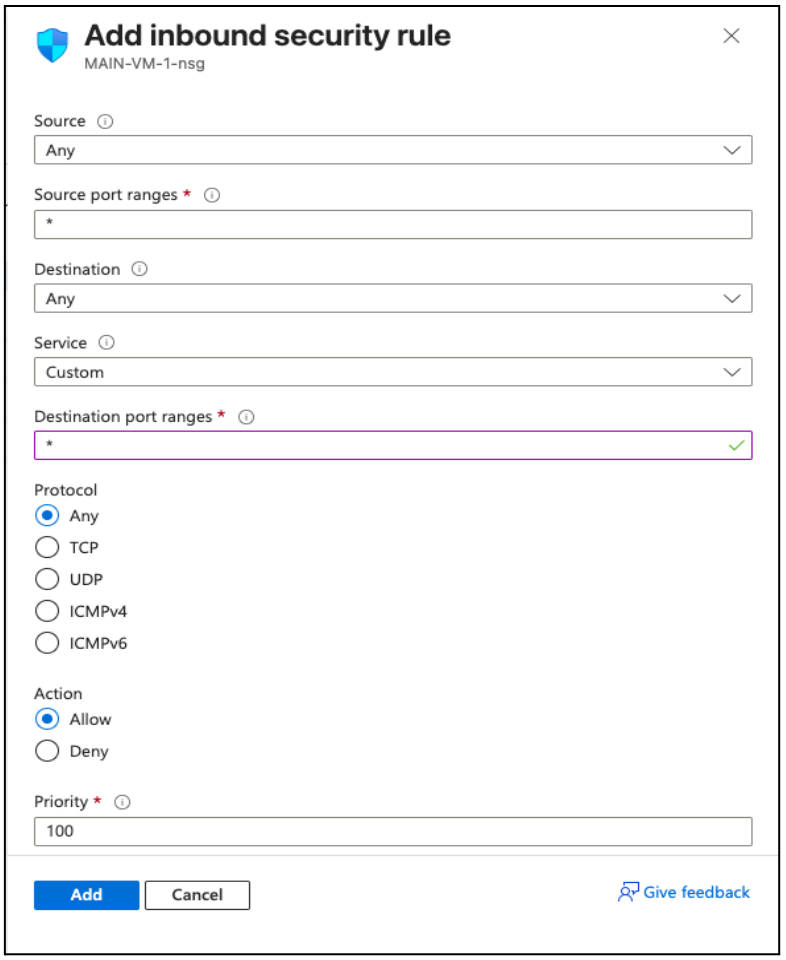
This rule was given the highest priority to ensure it would override any existing restrictions. Instead of leaving only RDP open, I removed the default RDP rule and created a new inbound security rule under the *Settings → Inbound security rules* section, effectively making the VM fully accessible for external attack traffic.

Figures:

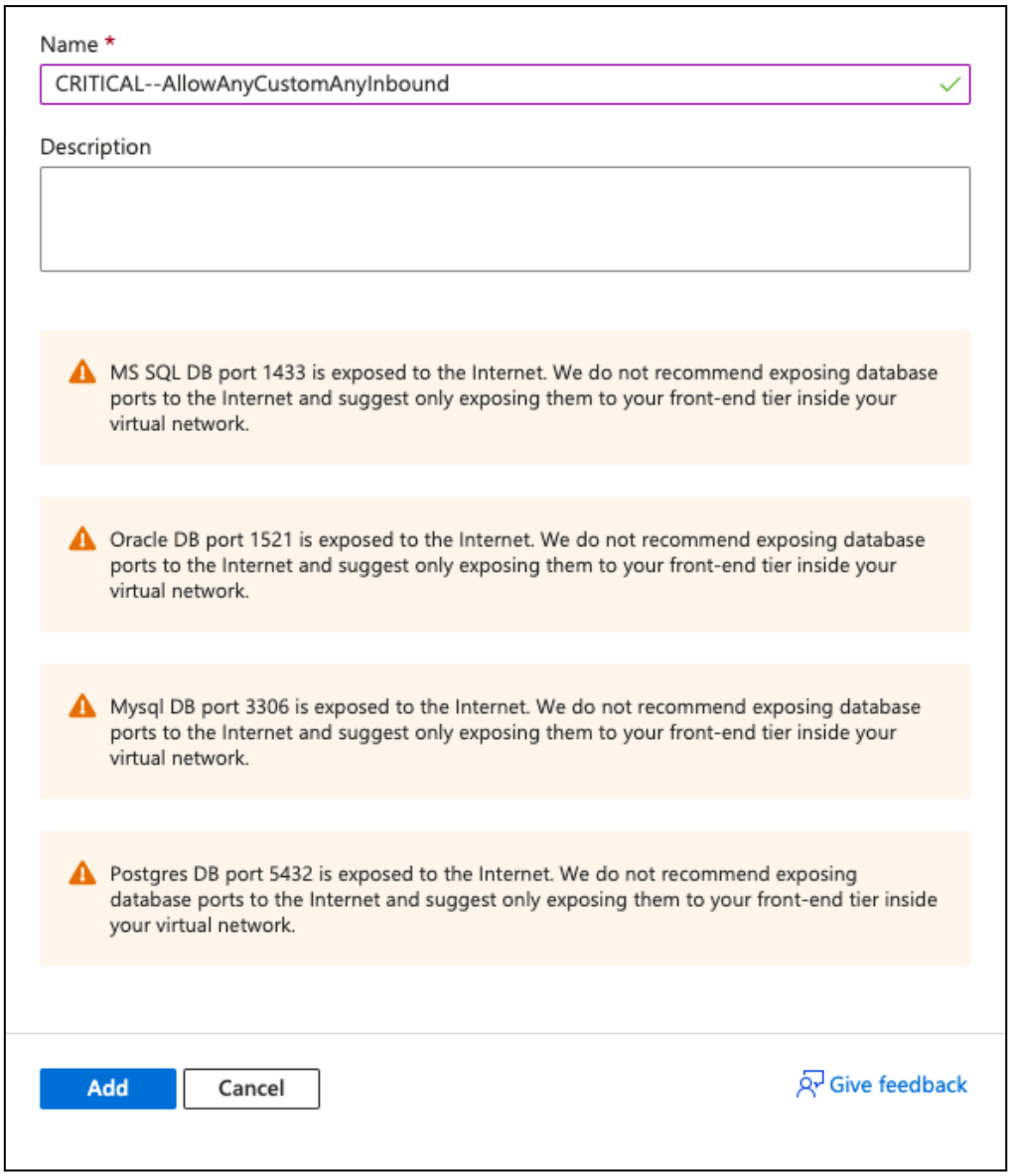
NSG overview.



Inbound security rule settings.

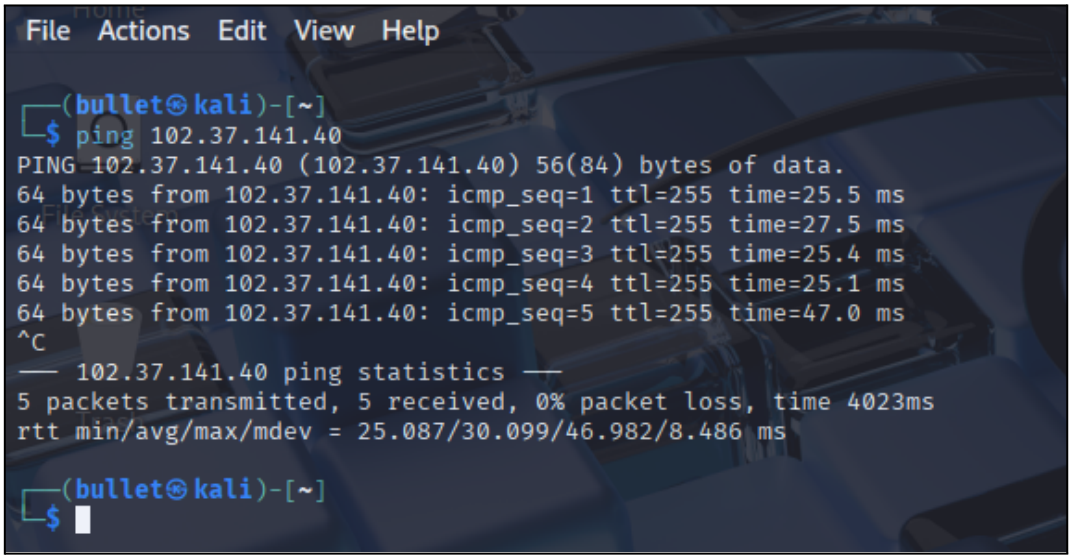


The inbound security rule was named *CRITICAL* to clearly distinguish it as the intentionally permissive rule. As indicated by the warnings, this configuration is highly insecure and should only be implemented in a controlled lab environment for educational purposes.

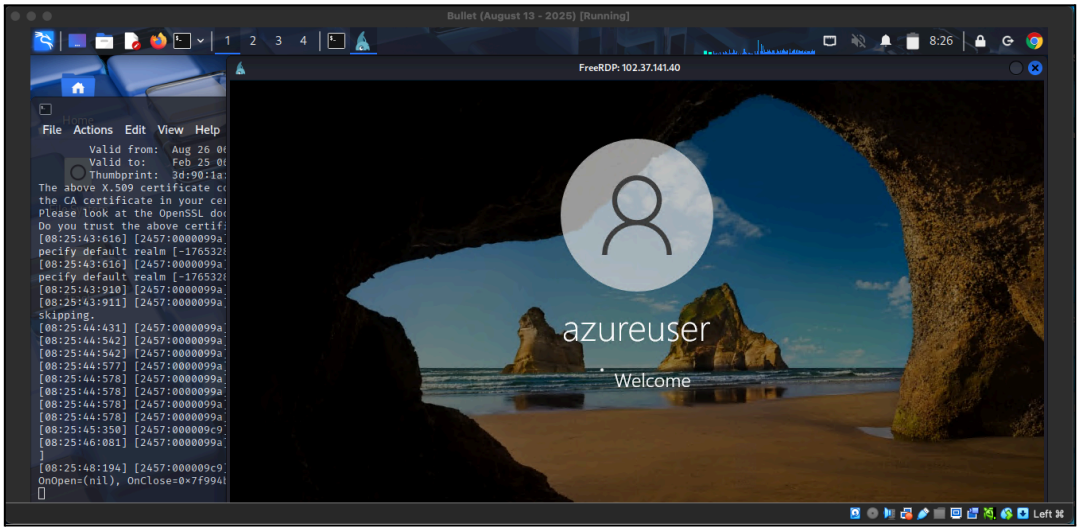


- *Next is to establish a connection to the machine.* -

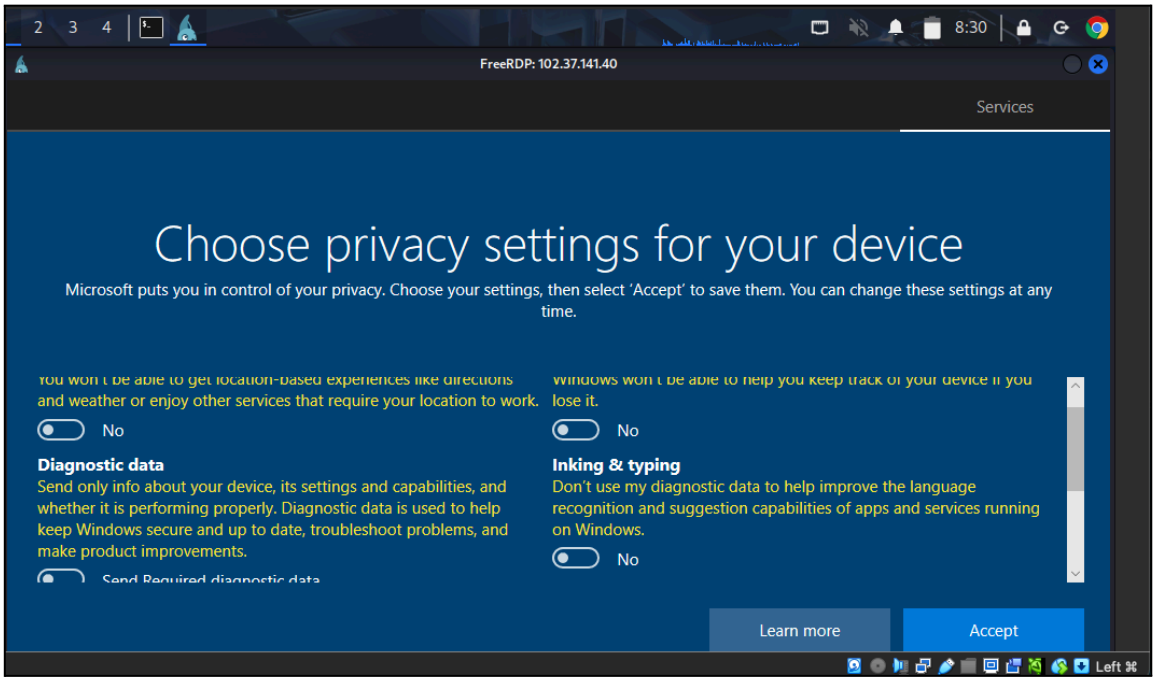
Verify connectivity by pinging the VM using its public IP address provided by Azure. This ensures the machine is reachable before proceeding with further configuration.



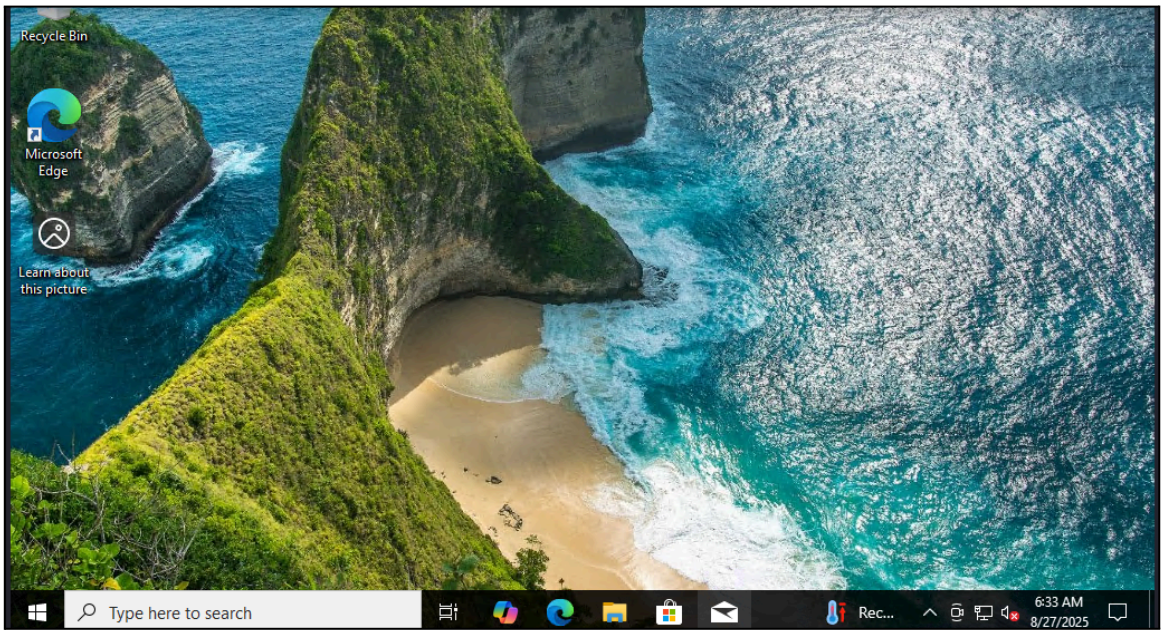
From my Linux virtual machine, I accessed the newly deployed Azure VM using the public IP address and credentials provided by Azure. The connection was established via RDP using the *xfreerdp* client, demonstrating secure remote access to the instance for monitoring and configuration purposes.



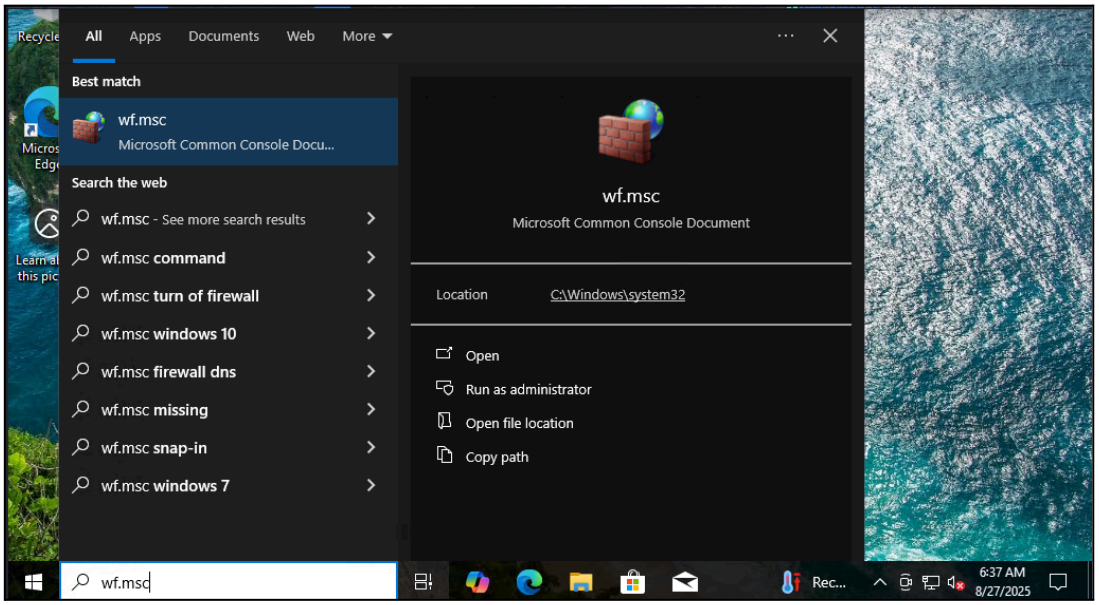
Disable all privacy settings.



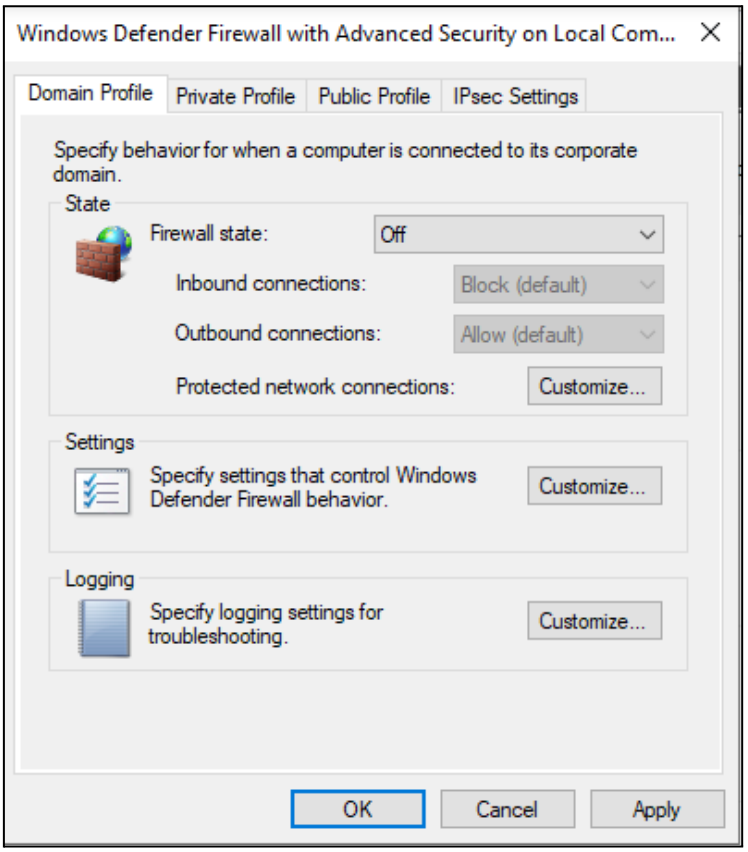
Connected.



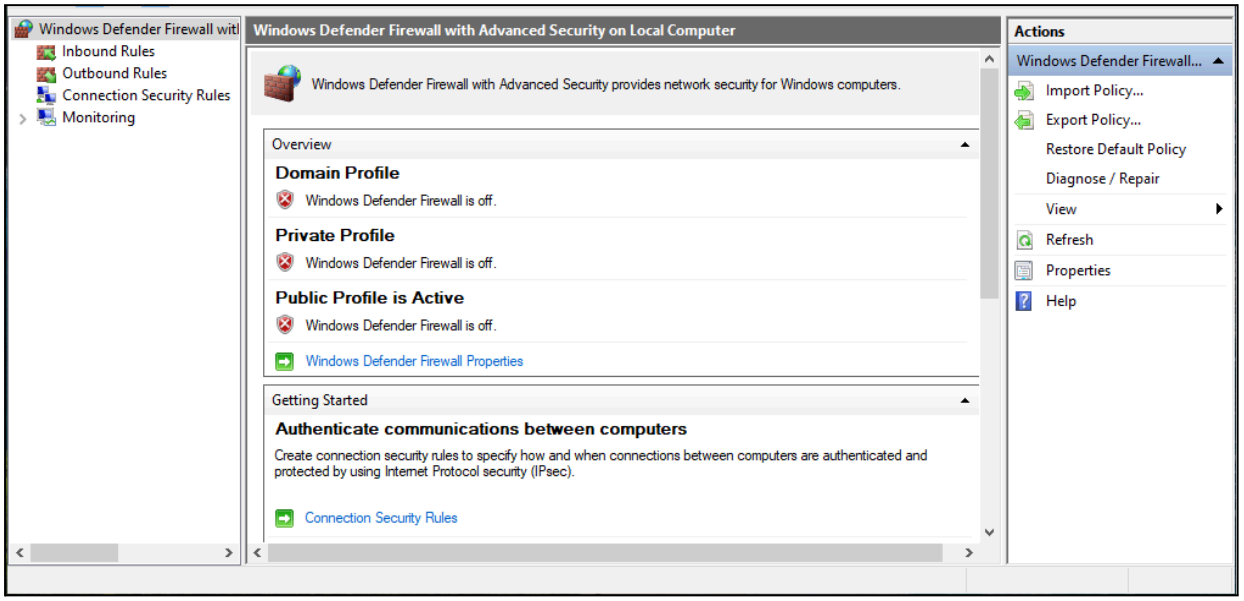
Go to wf.msc to access security settings.



Turn Firewall off on all profiles.



Results.



Section Takeaway

By deliberately exposing the VM to the public internet, it serves as an effective honeypot to attract and log real-world attack traffic for analysis.

---

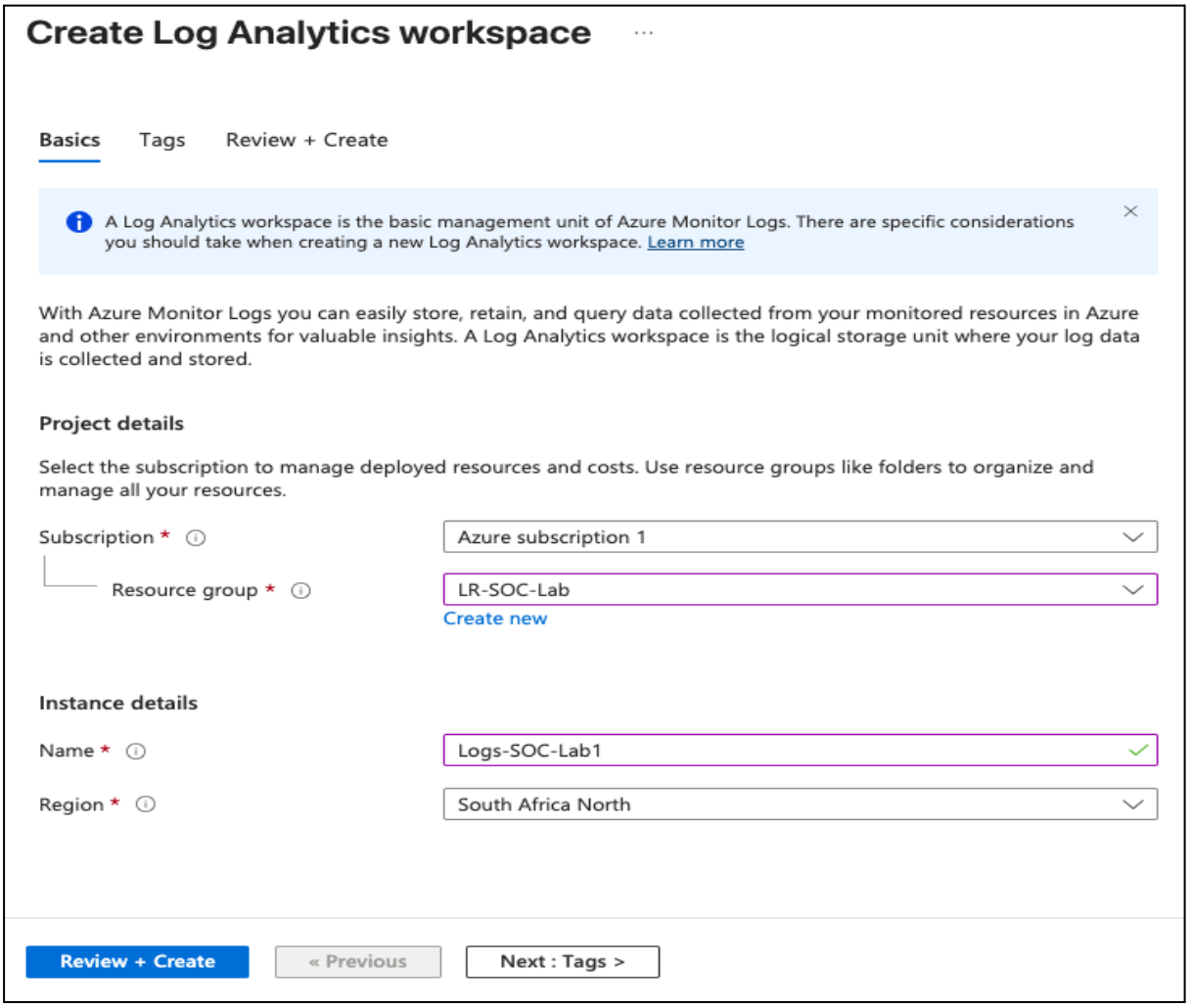
3. Log Collection and Centralisation

3.1 Log Analytics Workspace

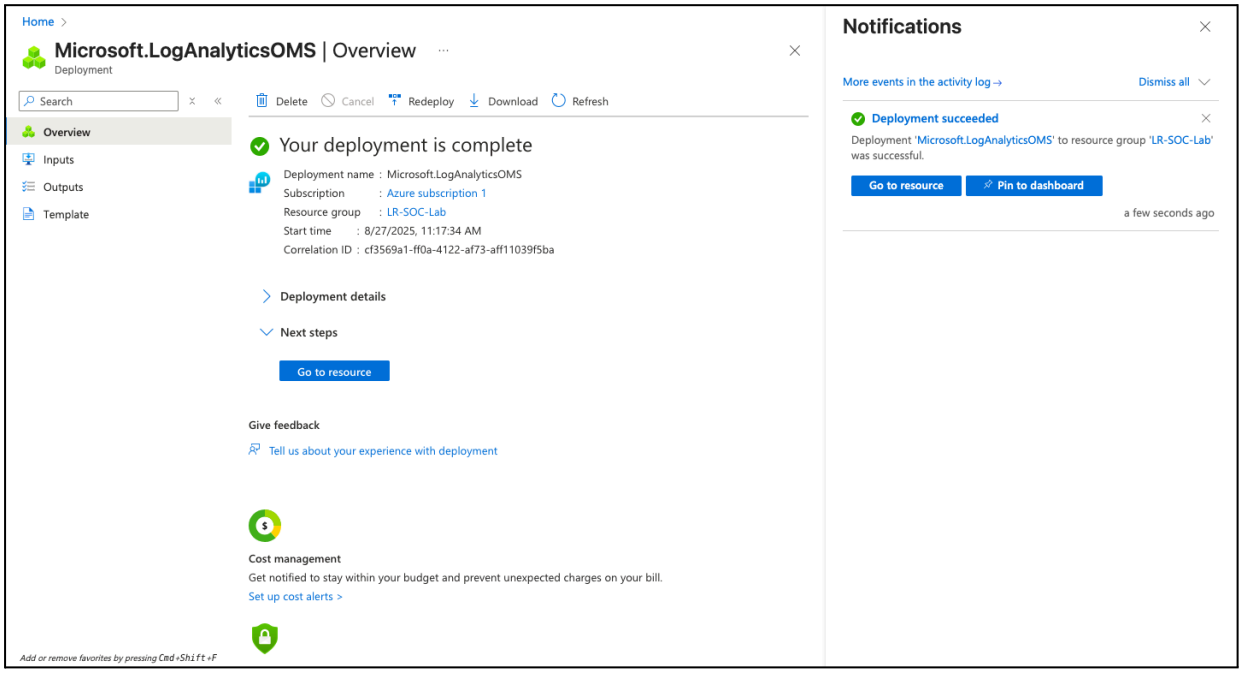
- Deployed a Log Analytics Workspace to serve as the central log repository.
- Configured retention policies for security-focused log storage.

Figures:

Create Log Analytics workspace.



Results.

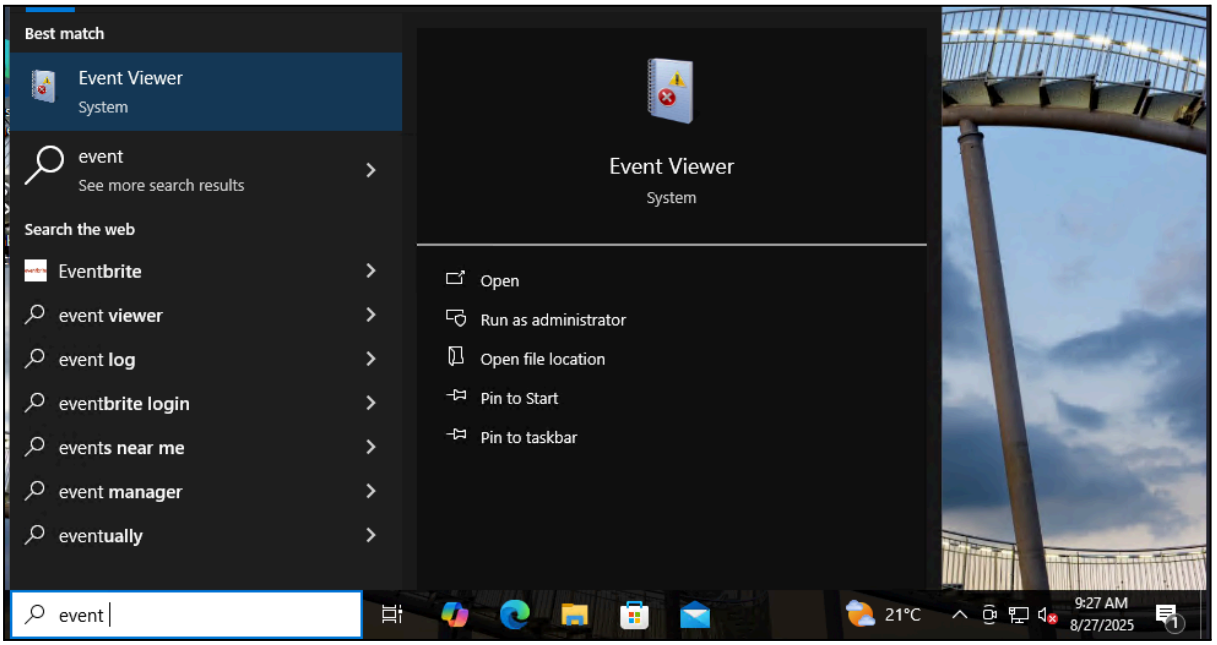


3.2 Azure Monitoring Agent

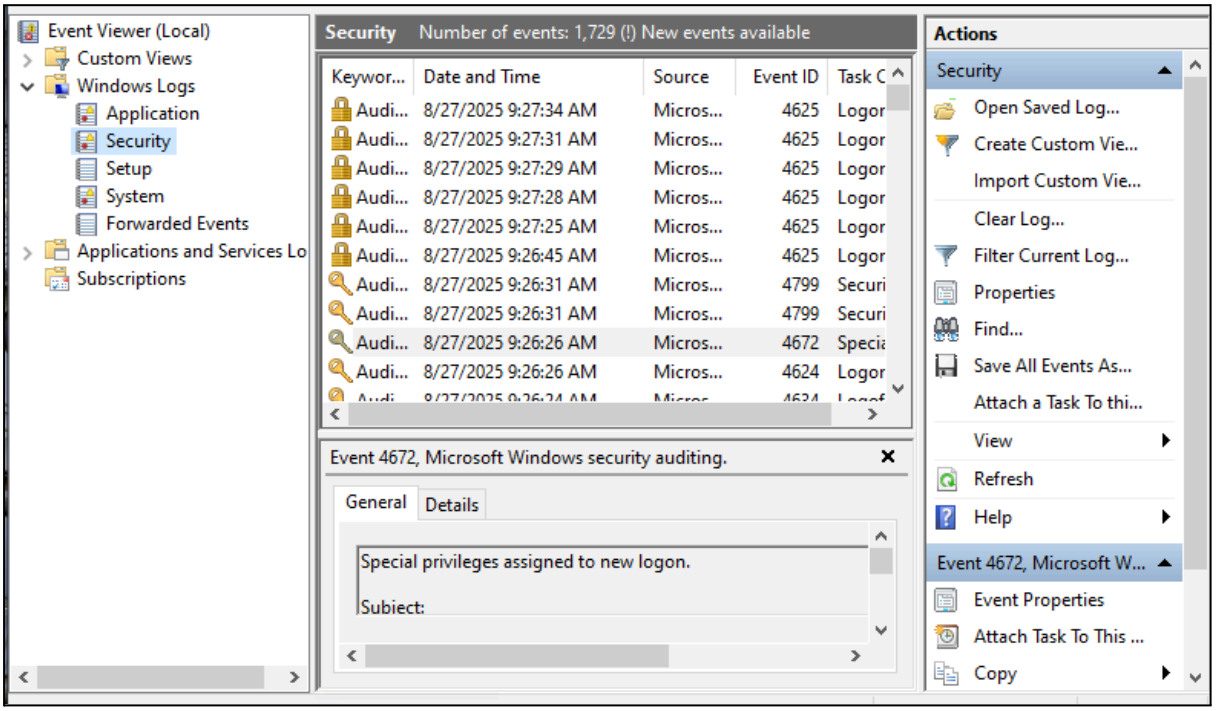
- Installed and configured the Azure Monitoring Agent on the VM.
- Forwarded all Windows Security Event Logs to the Log Analytics Workspace.

Figures:

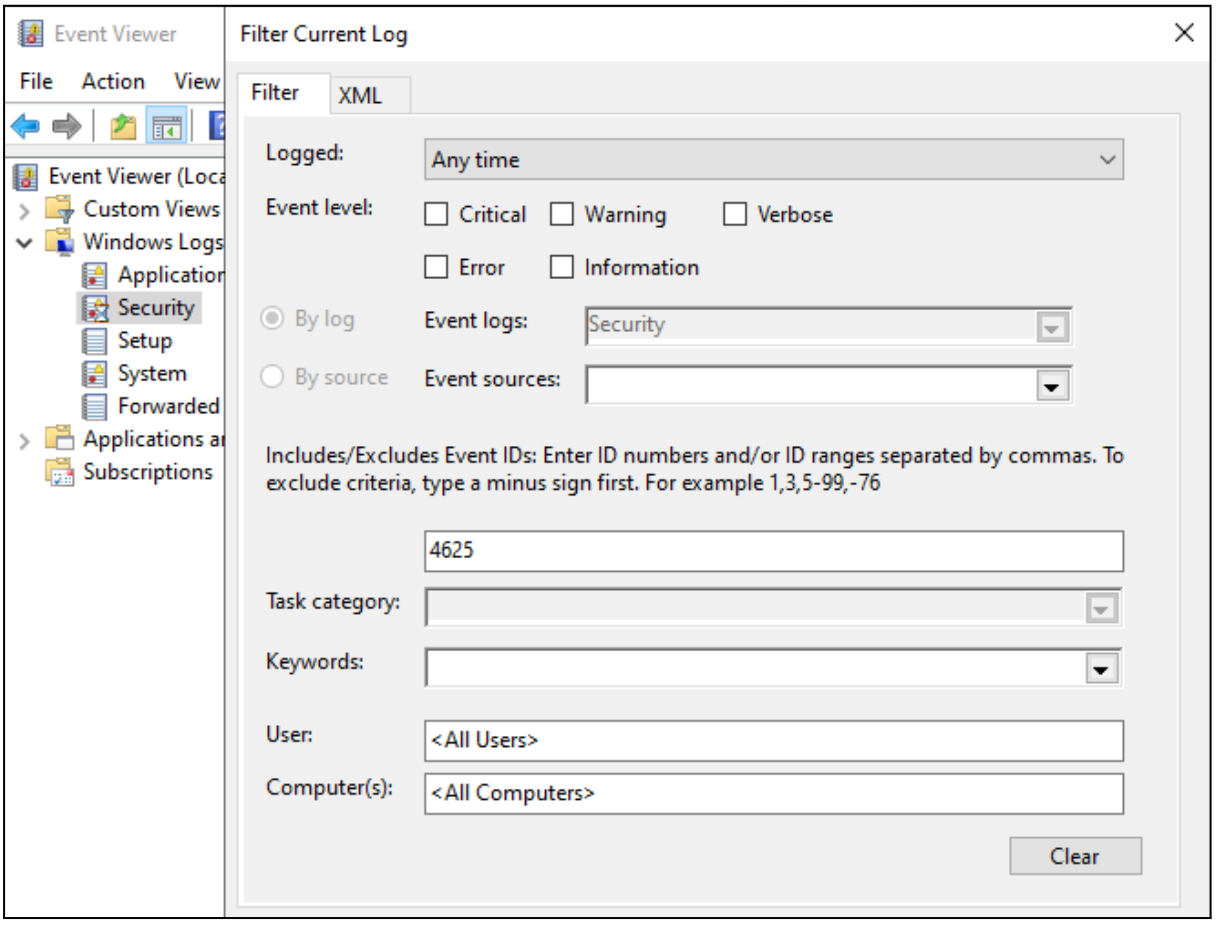
Open Event Viewer on your honeypot VM.



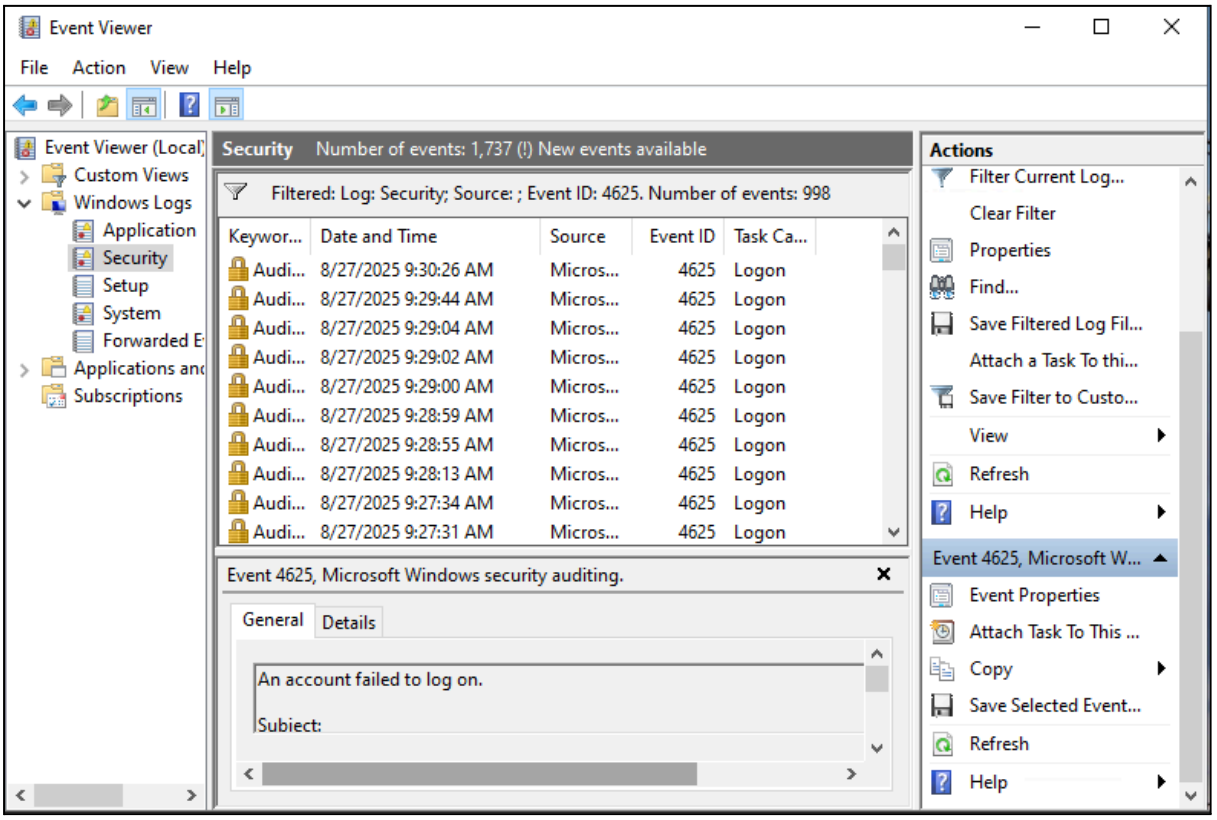
Event Viewer displaying security logs that capture various attacker activities.



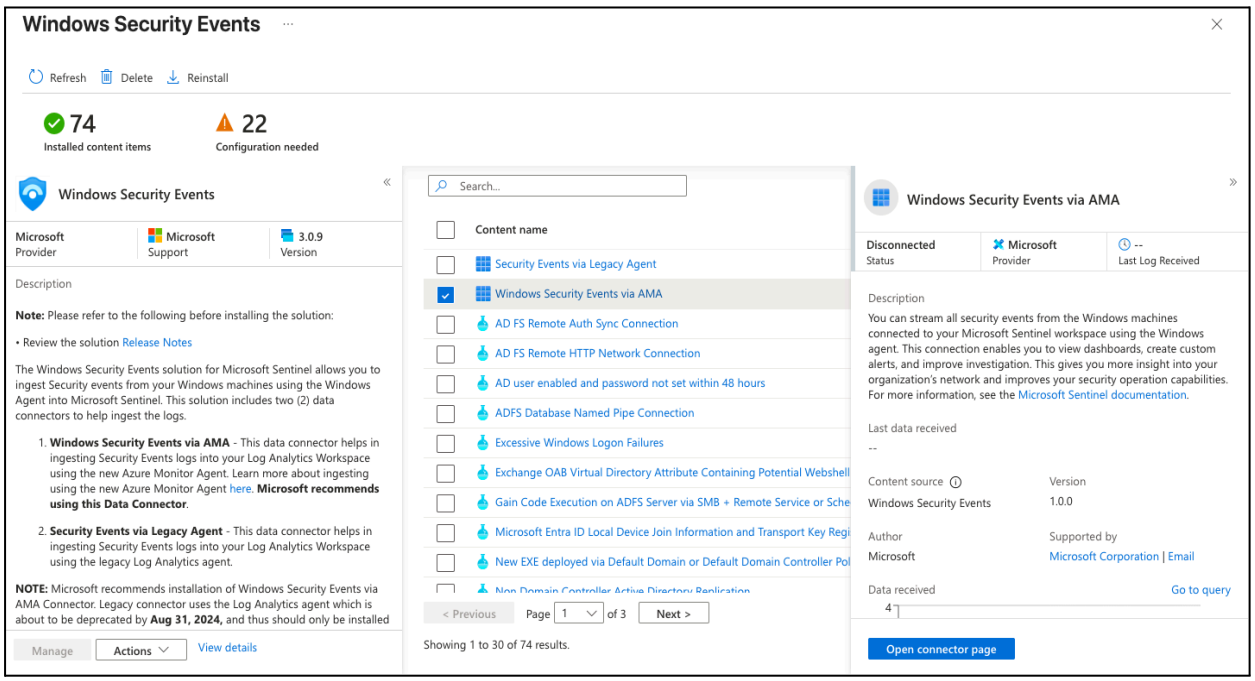
By selecting "Filter Current Log…" under the Actions pane, logs can be refined based on specific parameters for easier analysis.



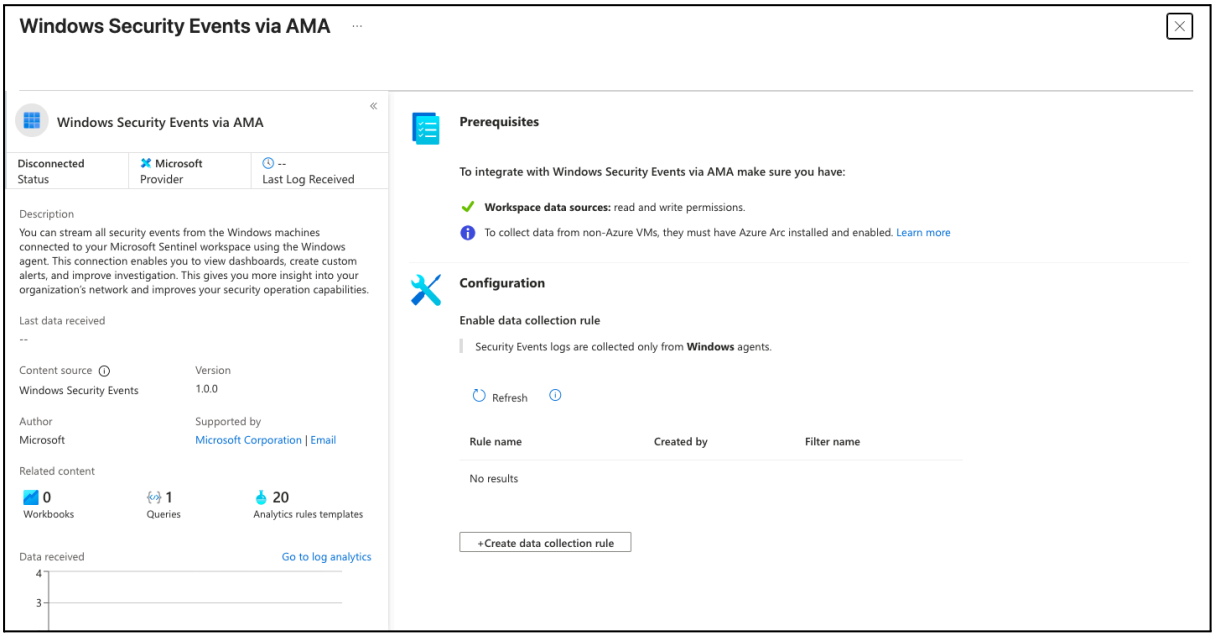
Results after filter is applied.



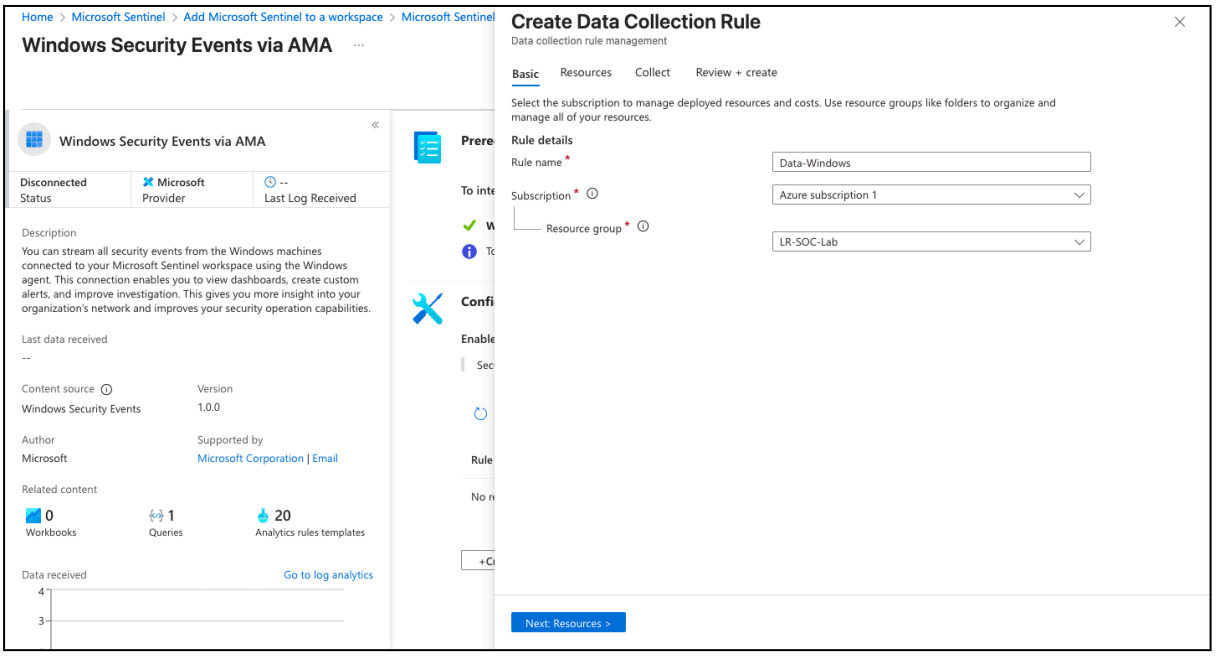
Windows Security Events ingested into Microsoft Sentinel via the Azure Monitor Agent (Content Hub configuration).



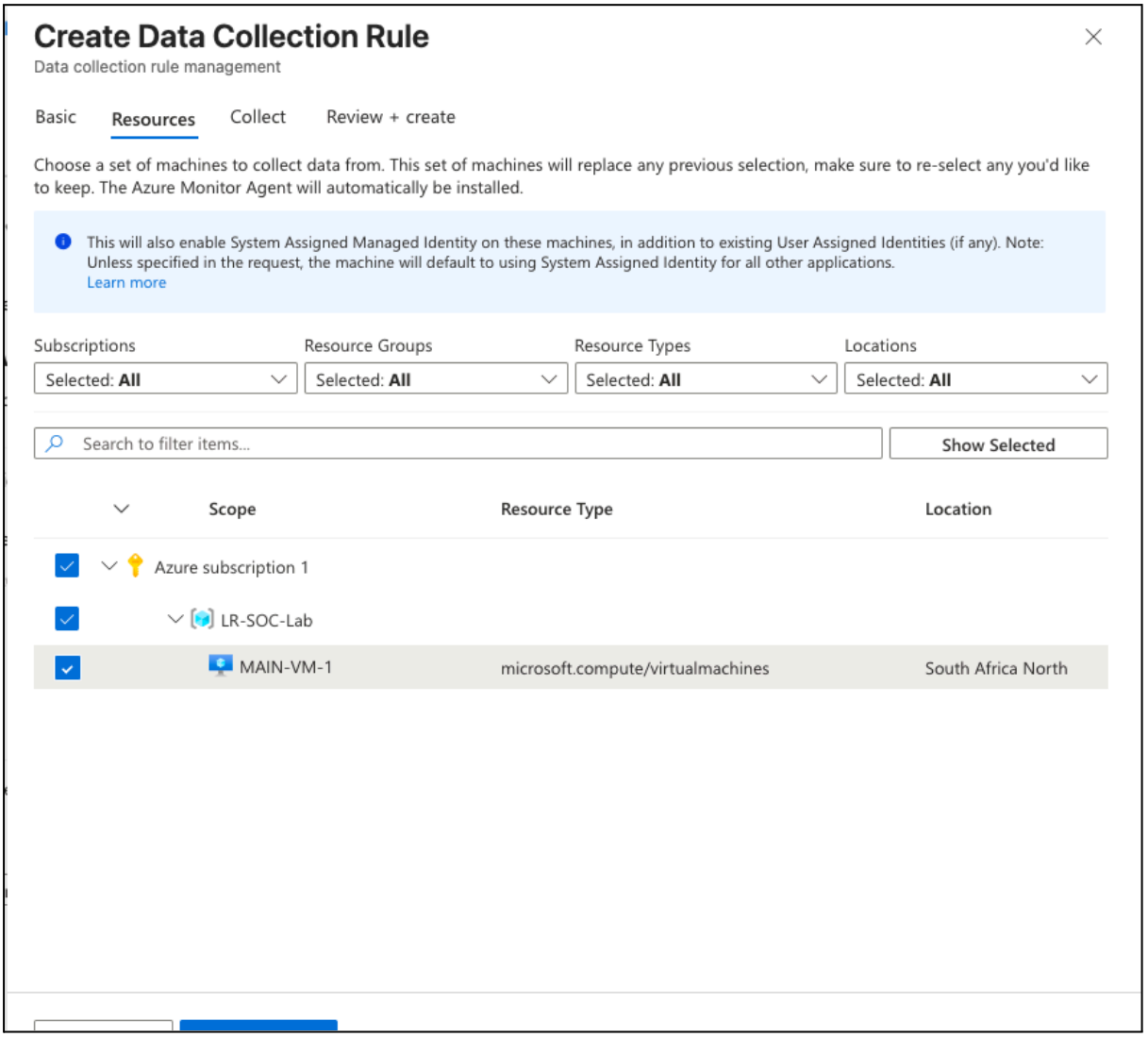
Create Data Collection Rule.



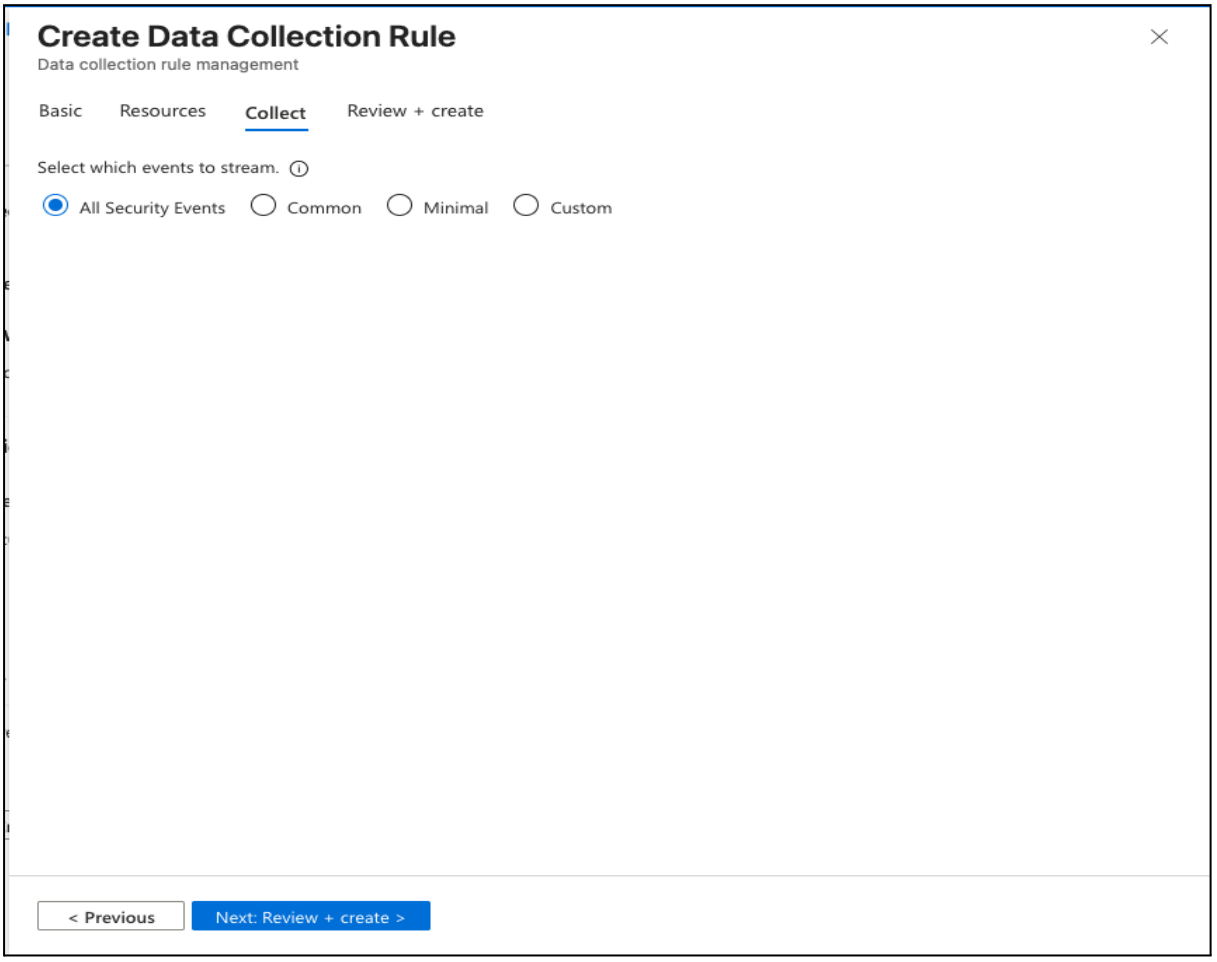
Cont.



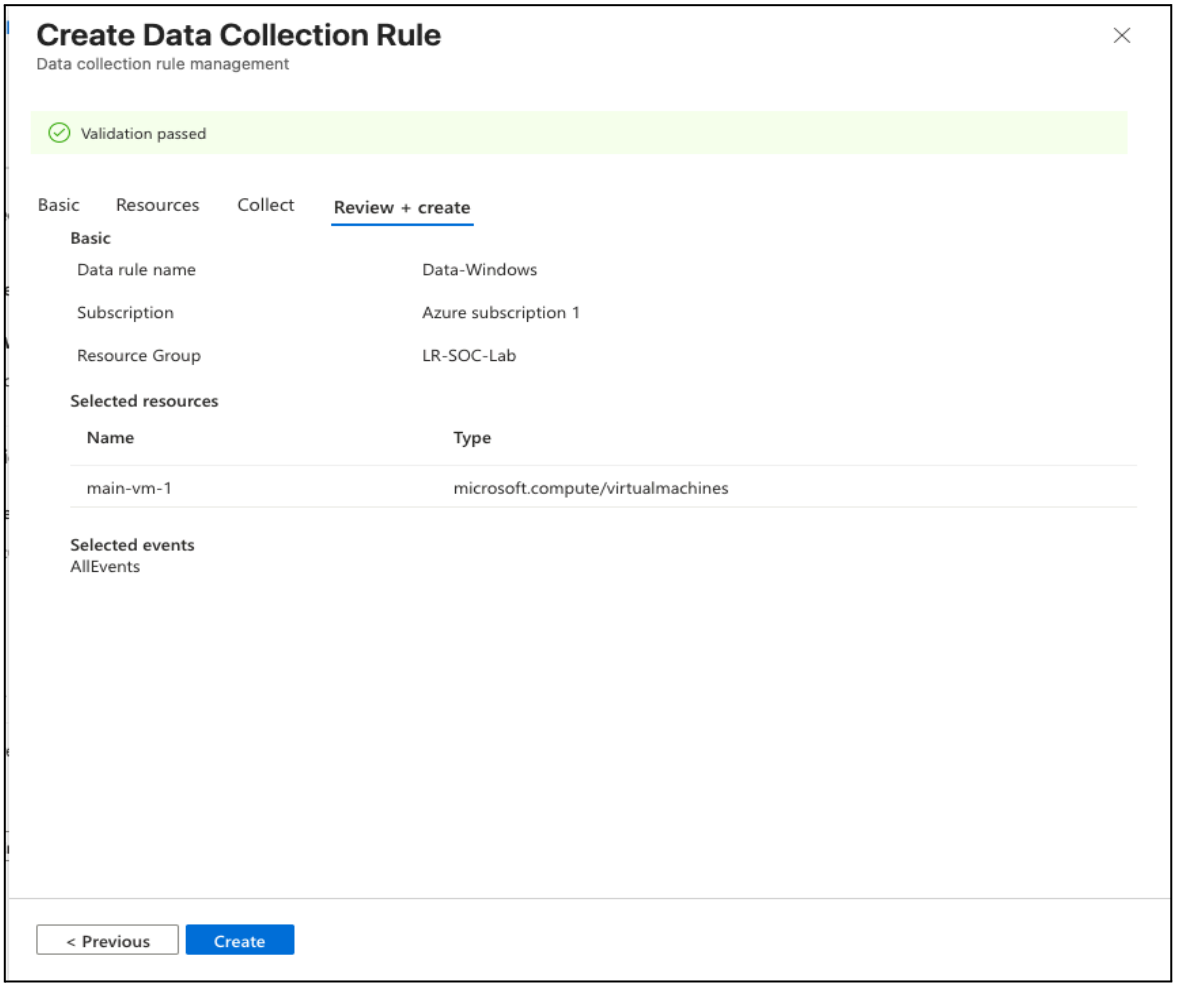
Select required resource group & honeypot VM.



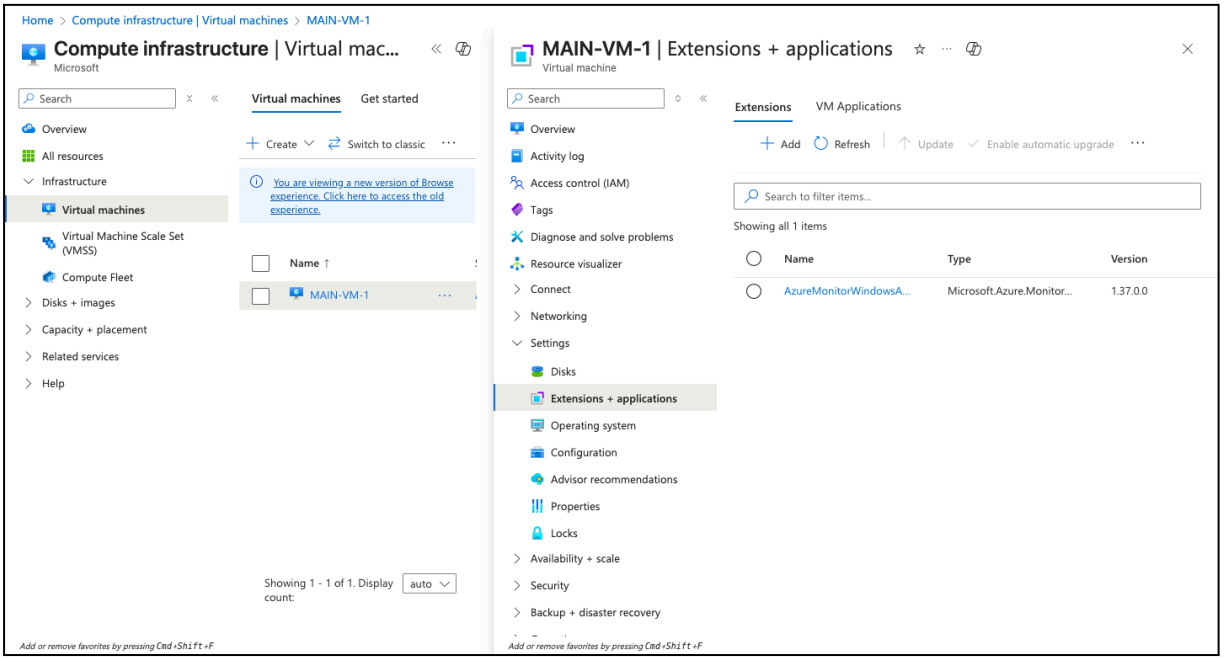
Choose the appropriate data collection type.



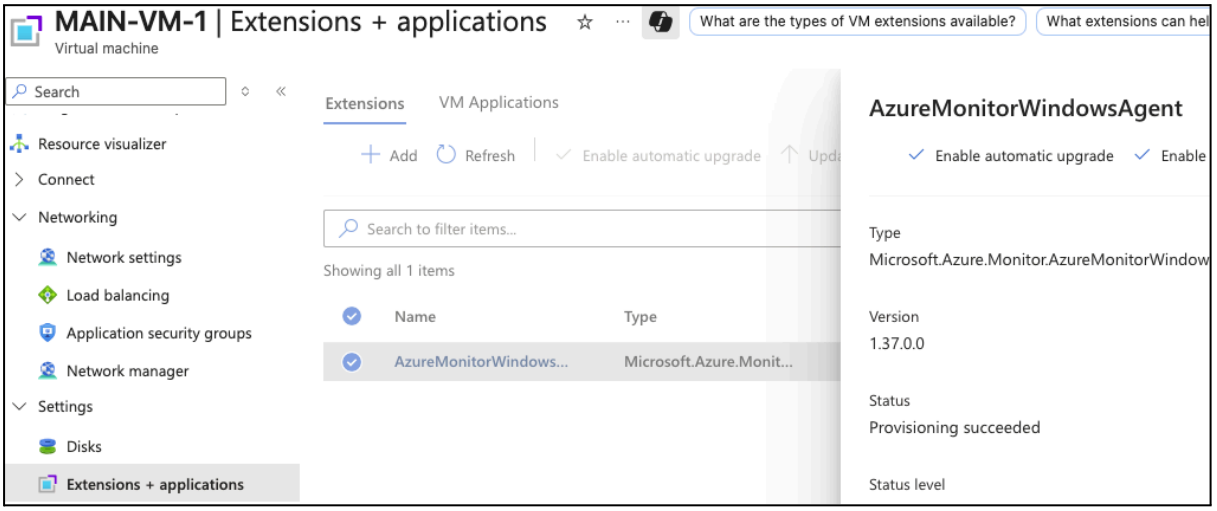
Review & Create.



Within the VM dashboard, navigate to Settings → Extensions + Applications to verify that the Azure Monitor Windows Agent has been successfully installed.



***Azure Monitor Agent** installed.*



Section Takeaway

Centralising logs in the workspace provided a single source of truth, simplifying monitoring and enabling more effective detection and analysis.

---

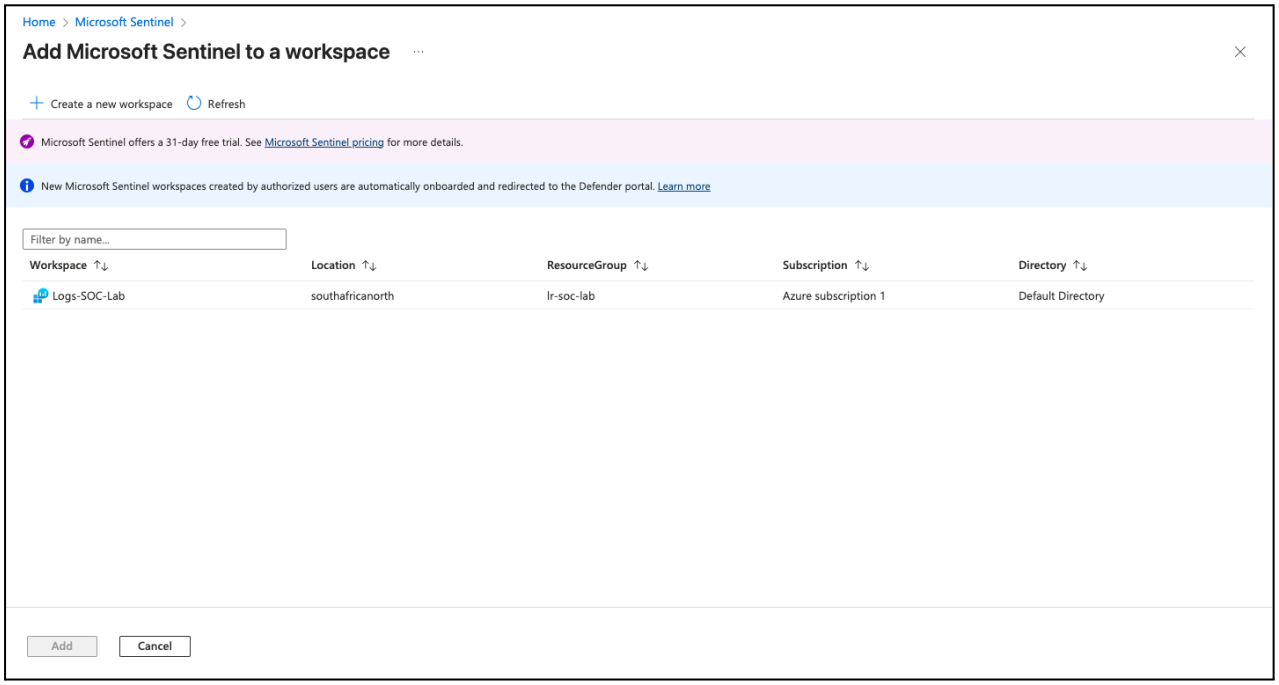
4. Sentinel SIEM Configuration

4.1 Sentinel Deployment

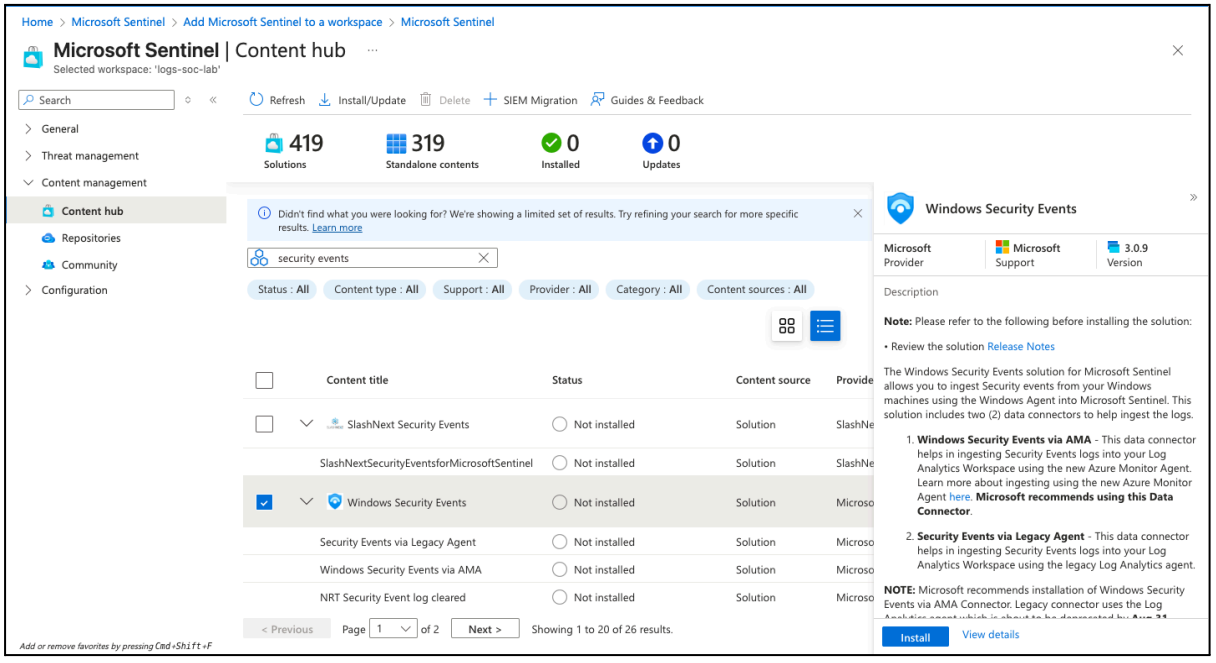
- Created an Azure Sentinel instance connected to the Log Analytics Workspace.
- Verified log ingestion from the honeypot VM.

Figures:

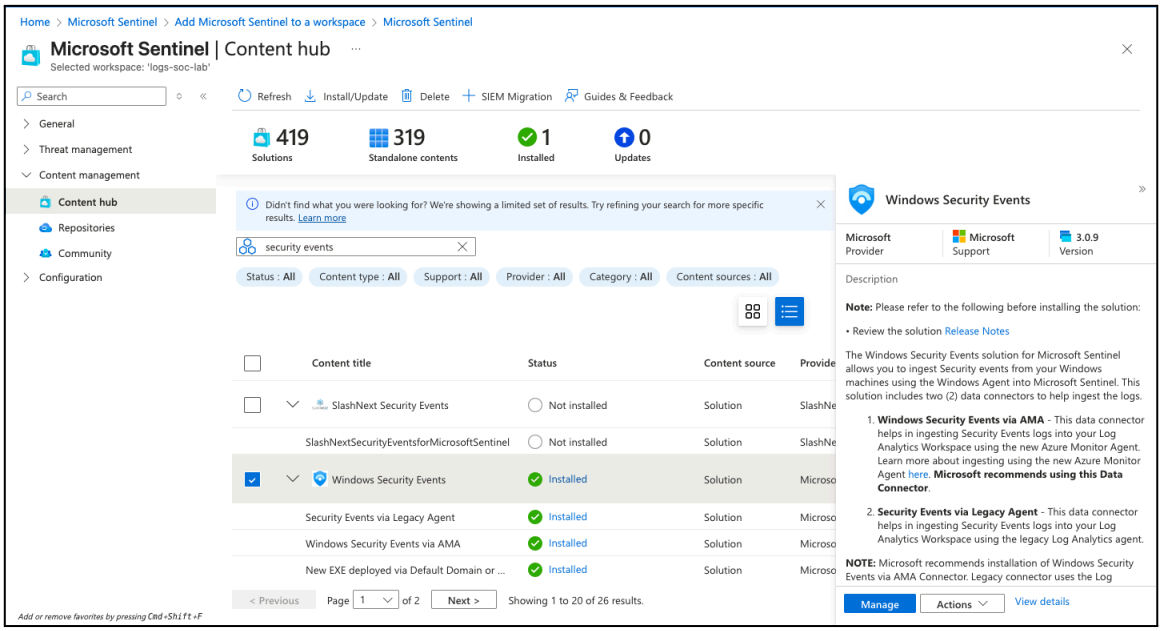
Deploy an Azure Sentinel instance and connect it to the existing Log Analytics workspace to enable centralised security monitoring and threat detection.



In the Content Hub, filter the search by 'security events' and install the Windows Security Events solution to enable ingestion of relevant security logs.



Installation complete.

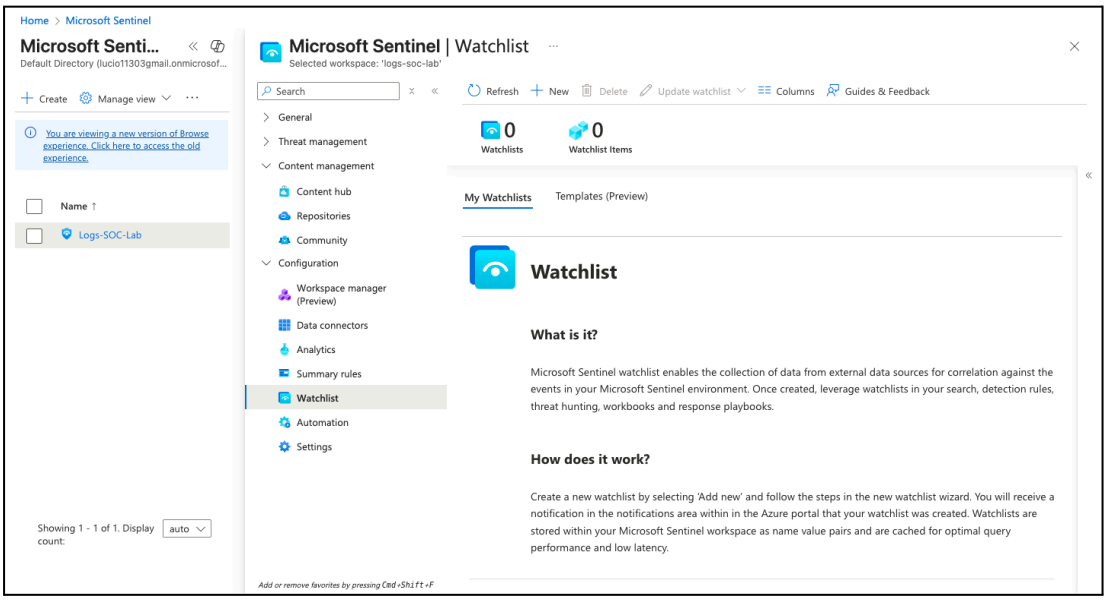


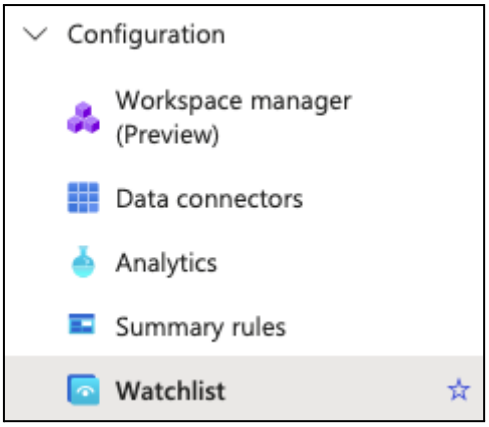
4.2 Threat Detection with Watchlists

- Uploaded a custom watchlist (spreadsheet containing IP ranges and geographic info).
- Correlated attacker IP addresses from failed logon events (4625) with the watchlist data.
- Enriched logs with geo-location context.

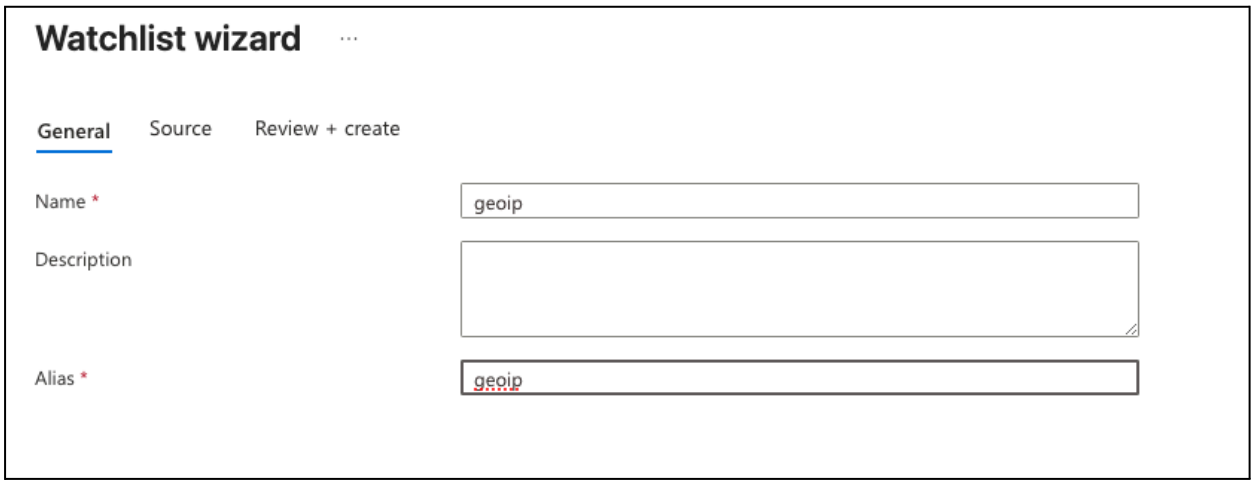
Figures:

Within Sentinel, navigate to Configuration → Watchlist and create a new watchlist instance to enrich log data with external context.

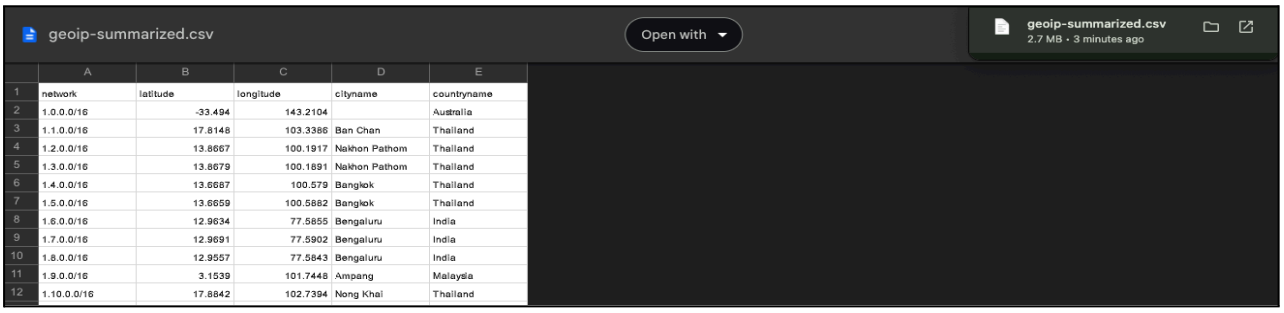




Provide a suitable name & alias.

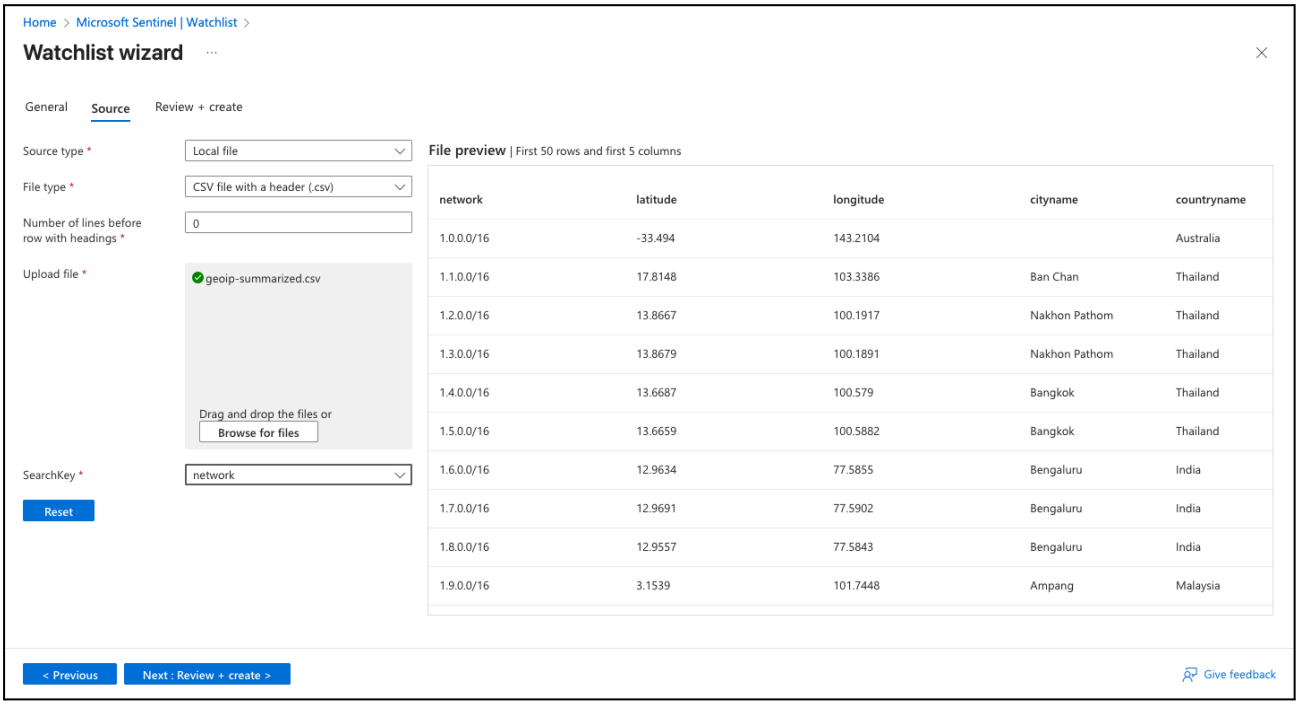


Prepare the watchlist file for upload; in this example, a pre-configured file provided by Josh Madakor is used.

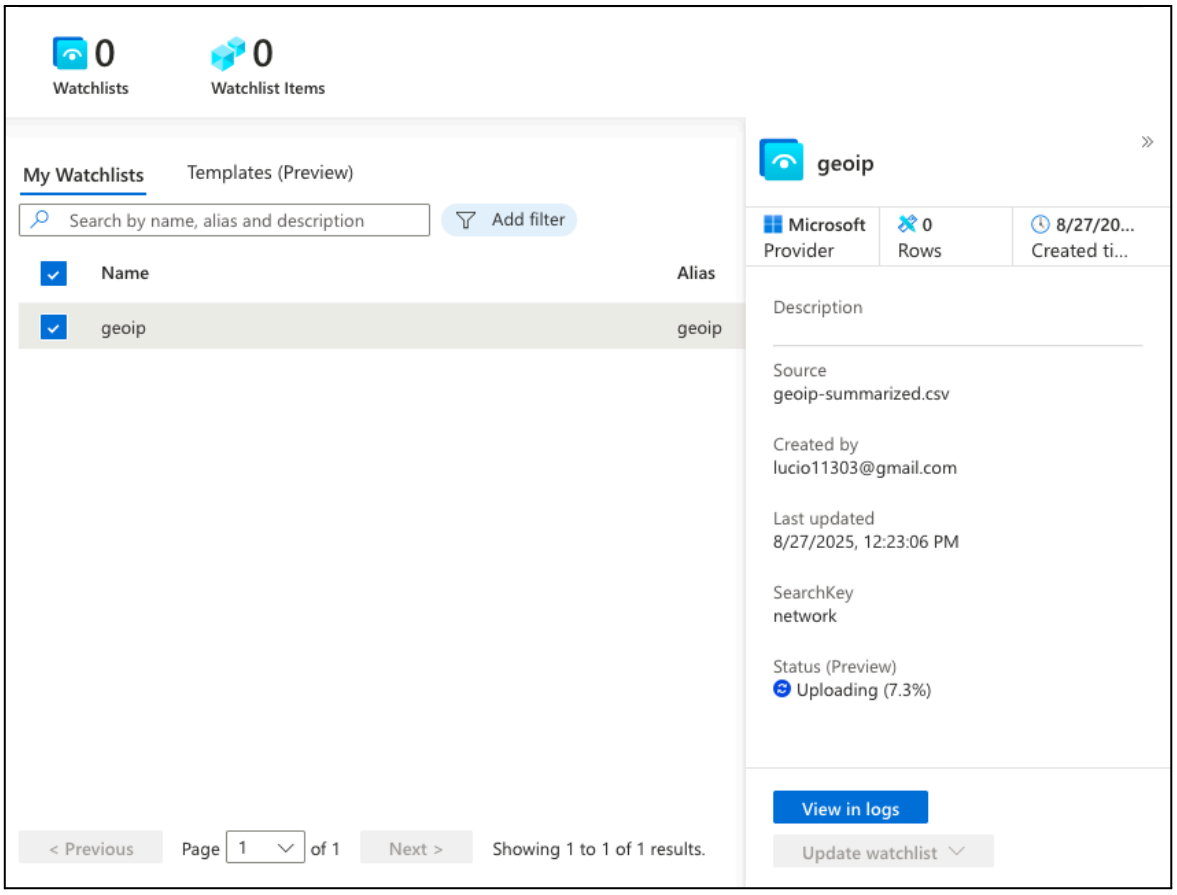


	A	B	C	D	E
1	network	latitude	longitude	cityname	countryname
2	1.0.0.0/16	-33.494	143.2104		Australia
3	1.1.0.0/16	17.8148	103.3386	Ban Chan	Thailand
4	1.2.0.0/16	13.8667	100.1917	Nakhon Pathom	Thailand
5	1.3.0.0/16	13.8679	100.1891	Nakhon Pathom	Thailand
6	1.4.0.0/16	13.6687	100.579	Bangkok	Thailand
7	1.5.0.0/16	13.6659	100.5882	Bangkok	Thailand
8	1.6.0.0/16	12.9634	77.5855	Bengaluru	India
9	1.7.0.0/16	12.9691	77.5902	Bengaluru	India
10	1.8.0.0/16	12.9557	77.5843	Bengaluru	India
11	1.9.0.0/16	3.1539	101.7448	Ampang	Malaysia
12	1.10.0.0/16	17.8842	102.7394	Nong Khai	Thailand

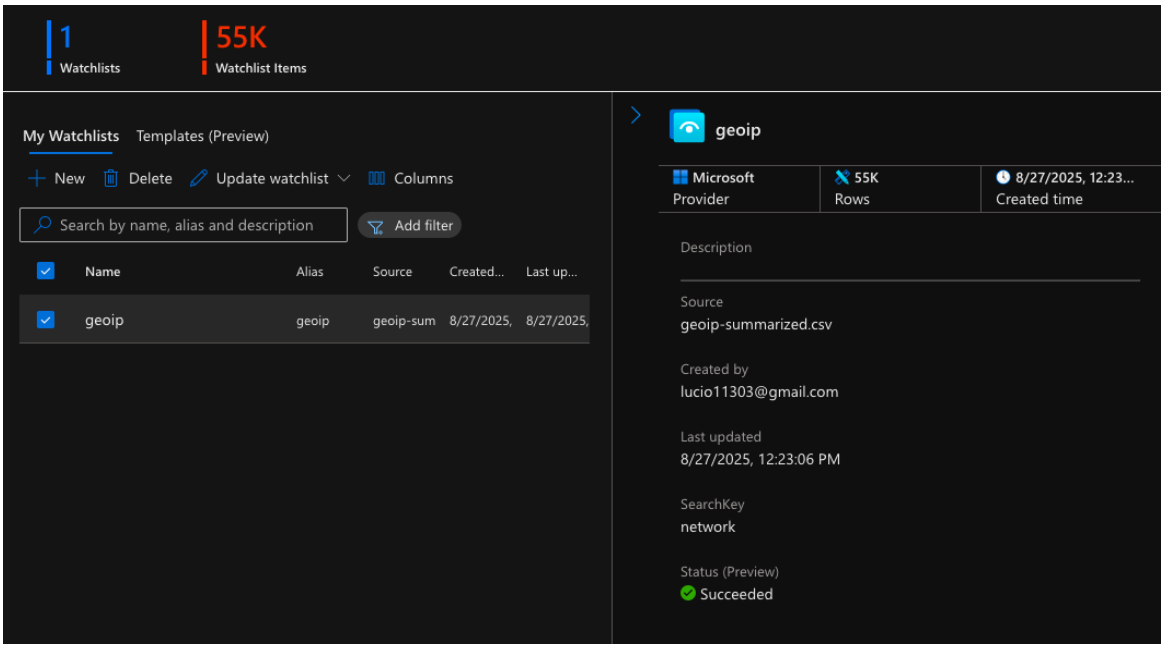
Select "network" as the SearchKey.



Wait for the selected file to upload fully.



Upload complete.



Section Takeaway

Integrating Sentinel with the workspace transformed raw event data into actionable intelligence, laying the foundation for meaningful threat hunting.

---

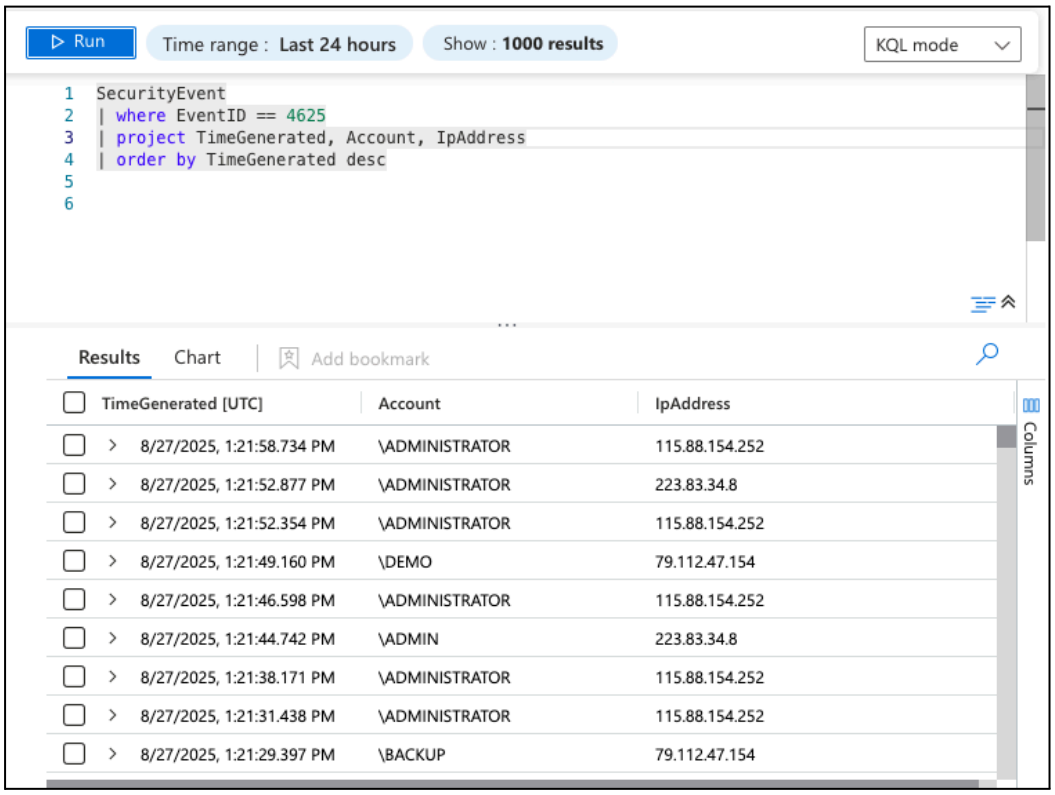
5. Threat Hunting and Visualization

5.1 KQL Queries

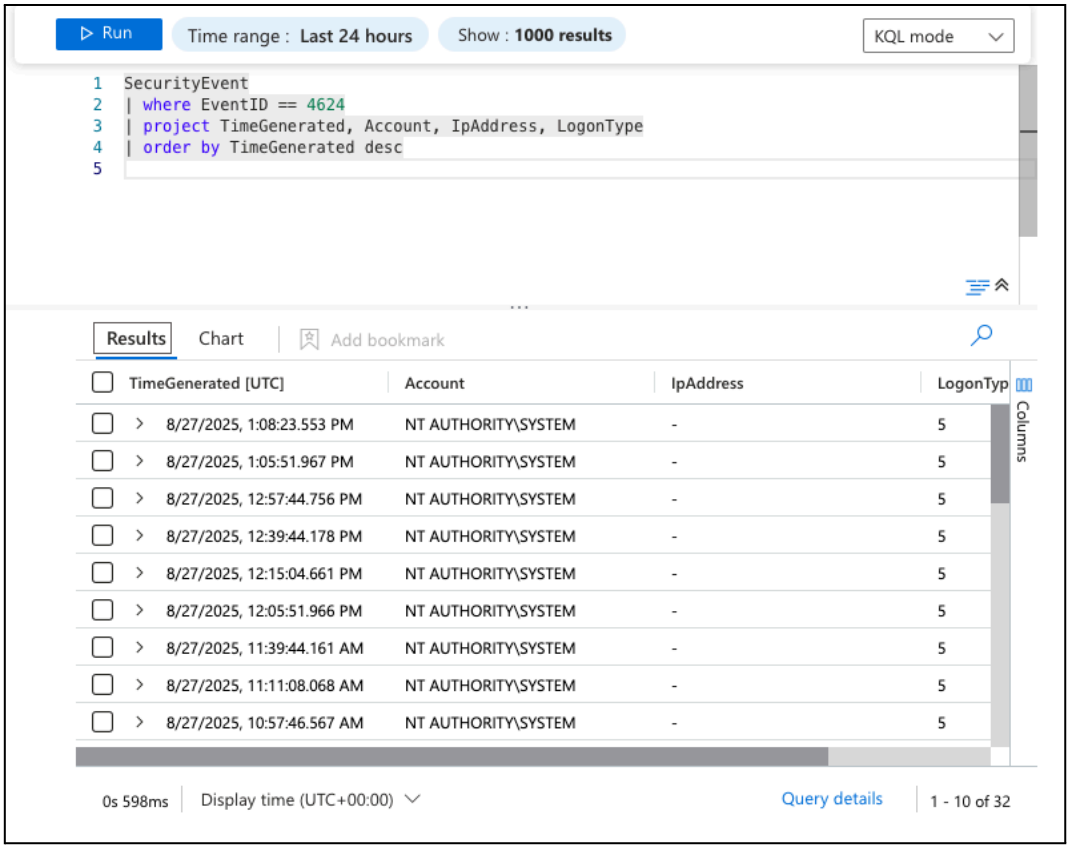
- Executed KQL queries to identify failed login attempts against the VM.
- Mapped source IP addresses to geographic locations using the watchlist.

Figures:

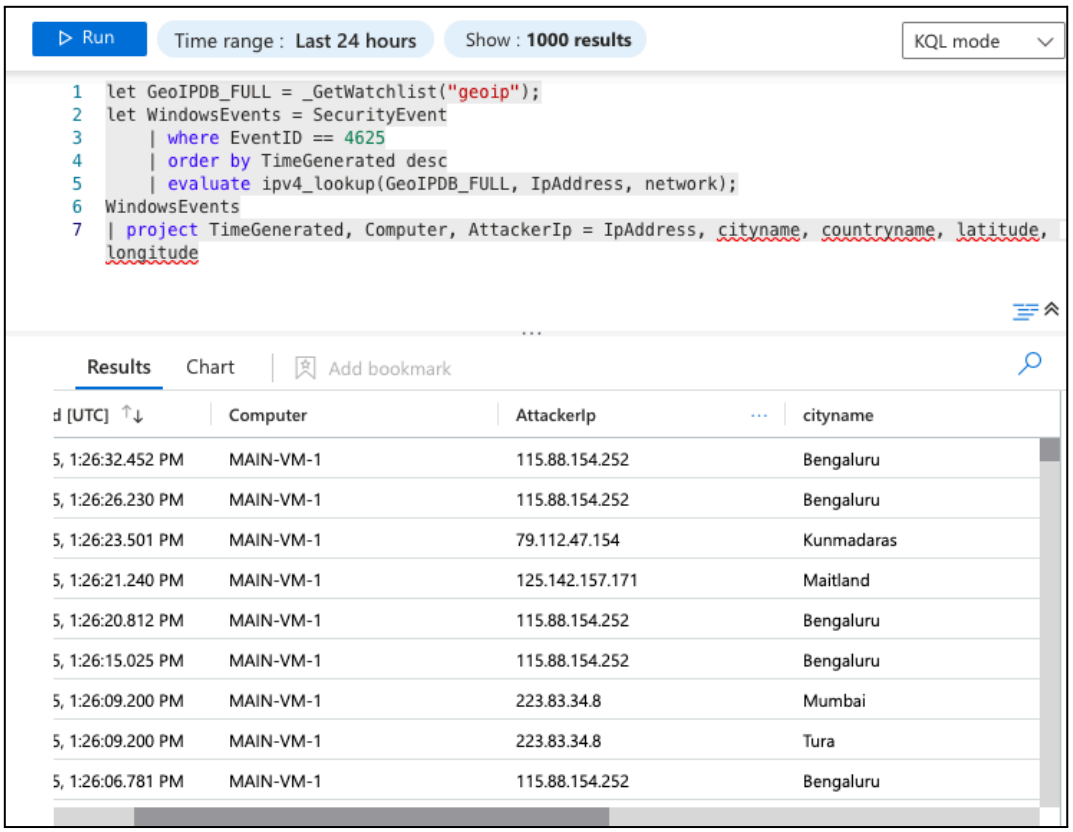
This query identifies failed Remote Desktop logon attempts (Event ID 4625) on the honeypot VM. Monitoring these attempts is important because attackers often brute-force credentials, and repeated failures from the same IP can indicate malicious behavior.

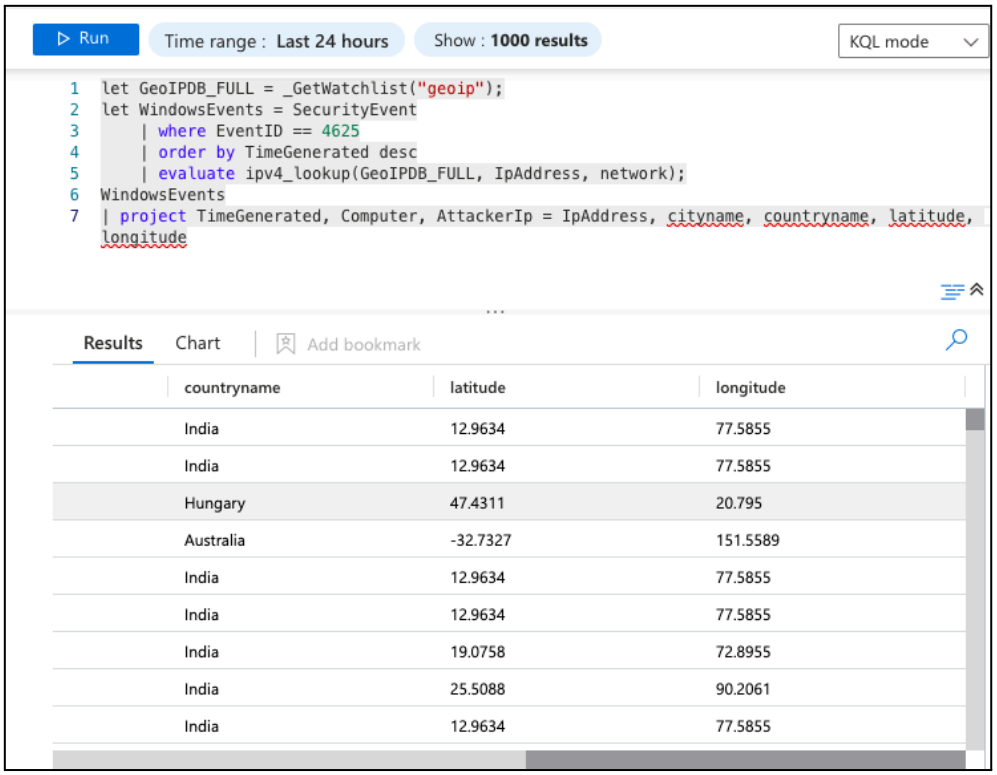


This query tracks successful logons (Event ID 4624). By comparing these with failed logons, we can determine if brute-force attempts eventually succeeded, which would indicate a compromised system.



This query correlates attacker IP addresses from failed logon events with the custom watchlist containing geolocation data. This enrichment allows us to visualise where attacks are originating globally, providing greater situational awareness of threat patterns.



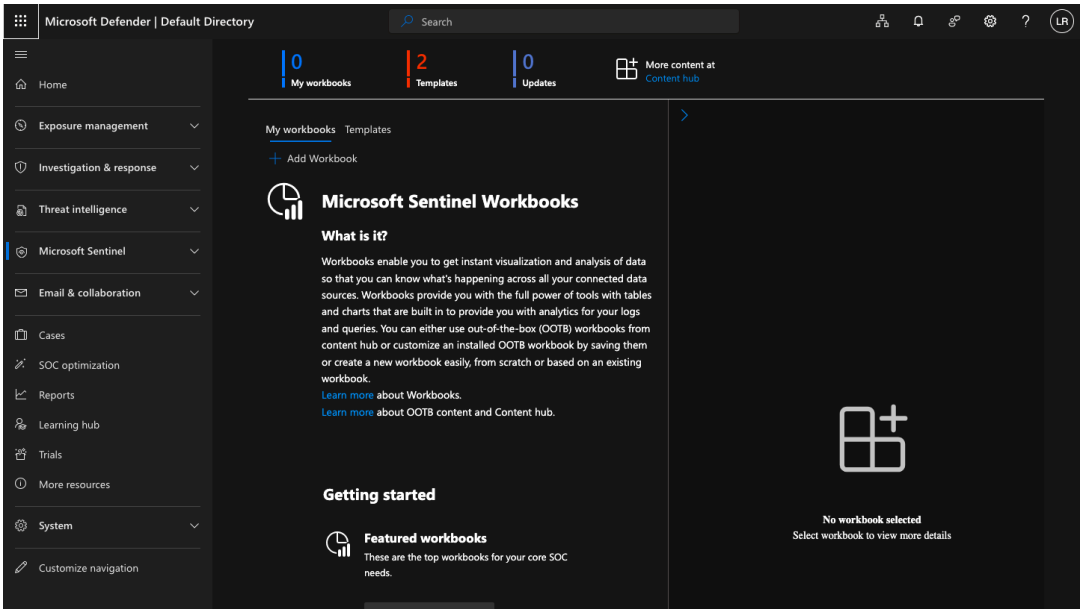


5.2 Attack Mapping

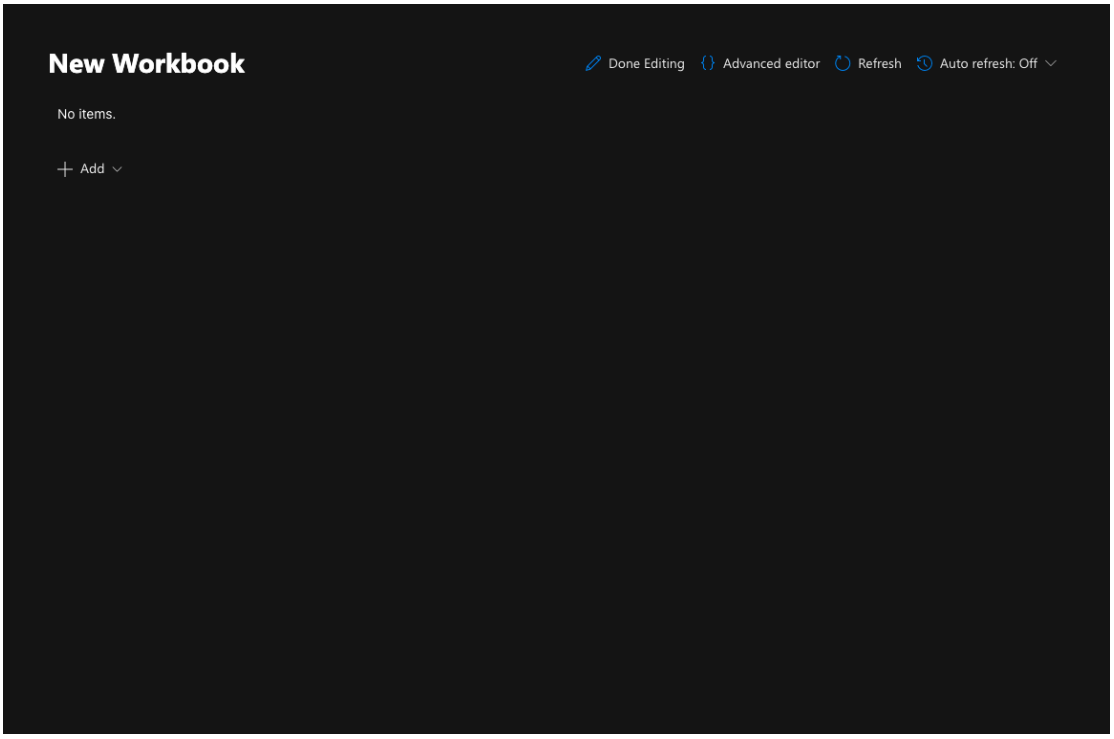
- Created a geographical visualisation within Sentinel showing attacker origins.
- This provided clear insight into global attack distribution.

Screenshot Example:

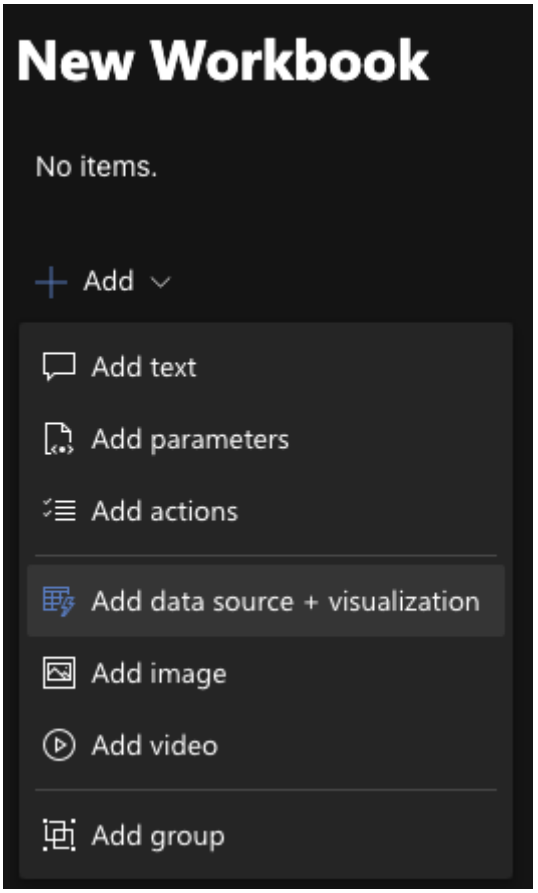
Create a sentinel workbook.



Clear pre-existing rules.



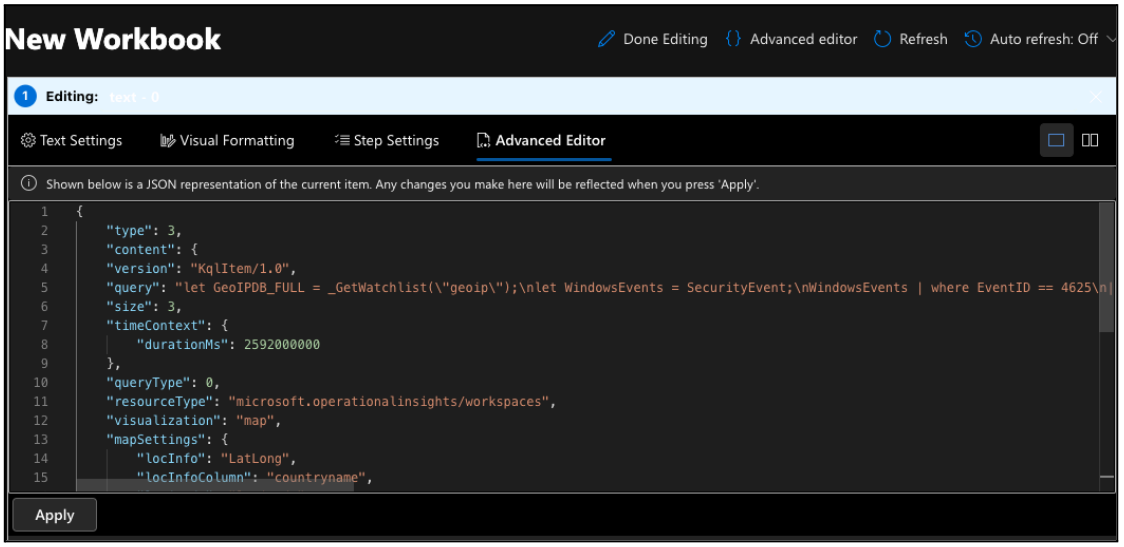
Create a new "Add data source + visualization" workbook.



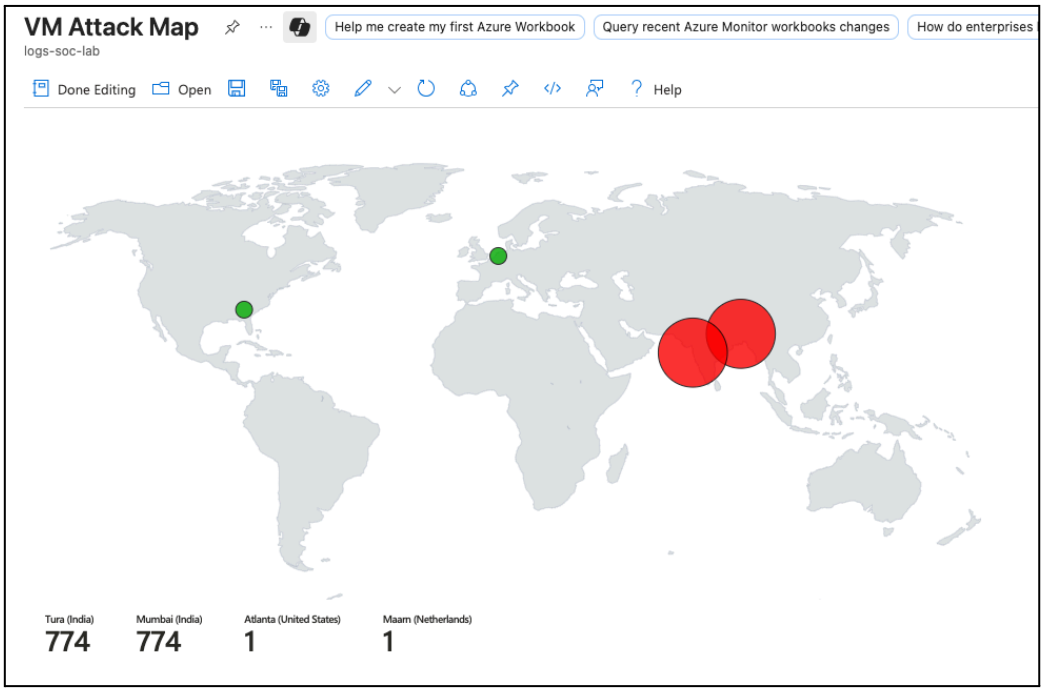
JSON code for the attack map, sourced from Josh Madakor, used to visualise attacker locations.

```
{
        "type": 3,
        "content": {
        "version": "KqlItem/1.0",
        "query": "let GeoIPDB_FULL = _GetWatchlist(\"geoip\");\nlet WindowsEvents = SecurityEvent;\nWindowsEvents | where EventID
== 4625\n| order by TimeGenerated desc\n| evaluate ipv4_lookup(GeoIPDB_FULL, IpAddress, network)\n| summarize FailureCount =
count() by IpAddress, latitude, longitude, cityname, countryname\n| project FailureCount, AttackerIp = IpAddress, latitude,
longitude, city = cityname, country = countryname,\nfriendly_location = strcat(cityname, \" (\", countryname, \")\");",
        "size": 3,
        "timeContext": {
                "durationMs": 2592000000
        },
        "queryType": 0,
        "resourceType": "microsoft.operationalinsights/workspaces",
        "visualization": "map",
        "mapSettings": {
                "locInfo": "LatLong",
                "locInfoColumn": "countryname",
                "latitude": "latitude",
                "longitude": "longitude",
                "sizeSettings": "FailureCount",
                "sizeAggregation": "Sum",
                "opacity": 0.8,
                "labelSettings": "friendly_location",
                "legendMetric": "FailureCount",
                "legendAggregation": "Sum",
                "itemColorSettings": {
                "nodeColorField": "FailureCount",
                "colorAggregation": "Sum",
                "type": "heatmap",
                "heatmapPalette": "greenRed"
                }
        }
        },
        "name": "query - 0"
}
```

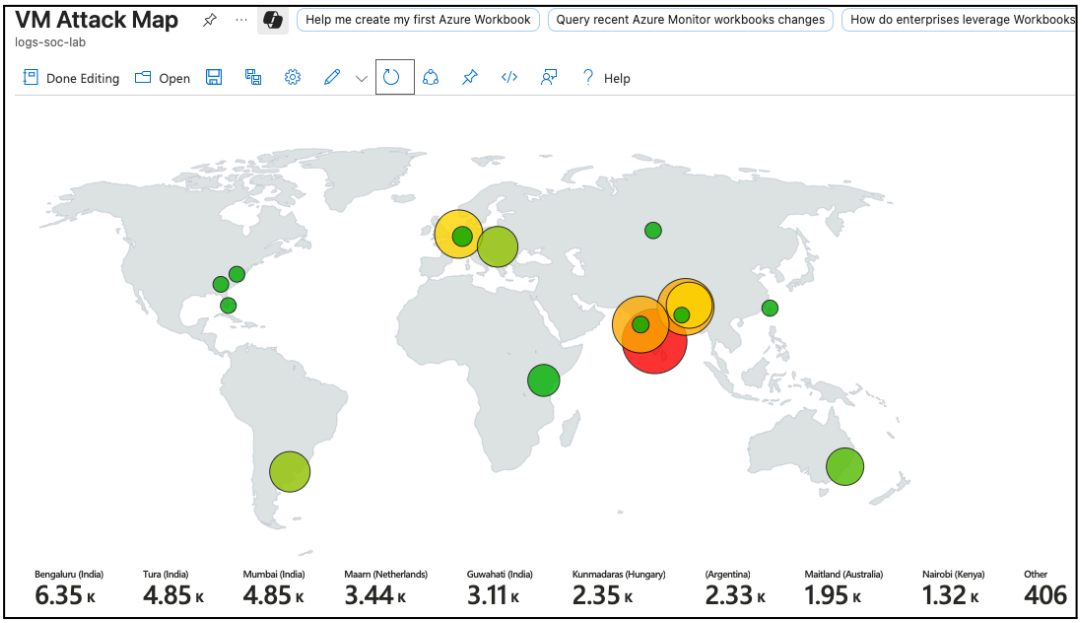
Navigate to Advanced Editor, clear pre-existing code and insert the map code.



3 hours of attack data.



12 hours of attack data.



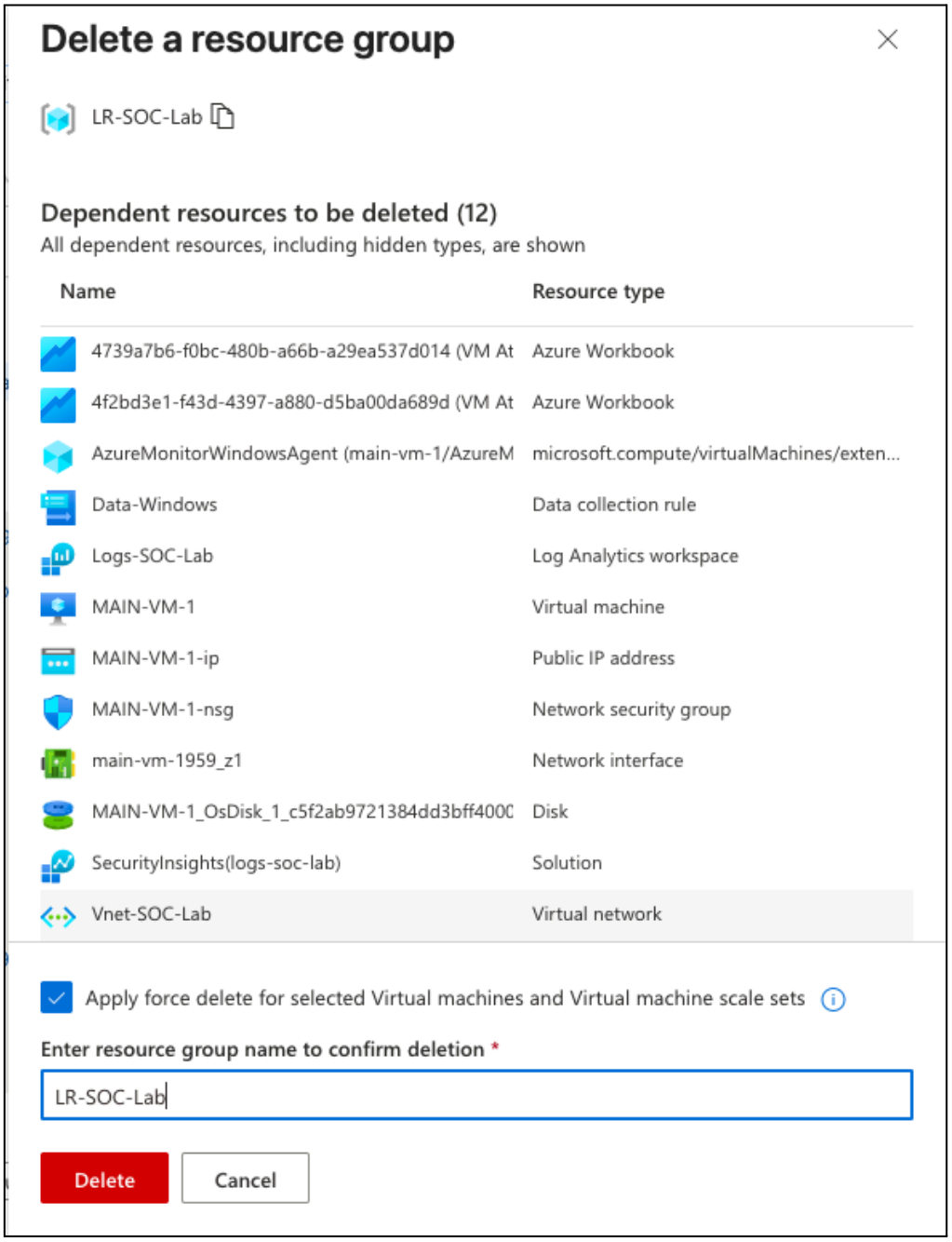
Section Takeaway

Through KQL queries and geographic mapping, raw attack logs were enriched with context, making patterns of malicious activity clear and actionable.

Cleanup and Teardown

Once the lab is complete, it is best practice to remove all deployed resources to reduce unnecessary costs and eliminate exposure of public-facing services. In Azure, this process is straightforward:

- **Delete the Virtual Machine (VM):** Remove the VM and any associated disks.
- **Remove the Network Security Group (NSG):** Delete custom inbound rules to prevent open attack surfaces.
- **Delete the Log Analytics Workspace:** If no longer needed, removing the workspace ensures logs and data ingestion stop, preventing additional charges.
- **Delete the Microsoft Sentinel Instance:** If it was only used for this lab, removing the instance avoids ongoing costs.
- **Clean Up Resource Groups:** Deleting the entire resource group will automatically remove all associated resources in one step.


Delete a resource group
LR-SOC-Lab
Dependent resources to be deleted (12)
All dependent resources, including hidden types, are shown
Name
Resource type
4739a7b6-f0bc-480b-a66b-a29ea537d014 (VM At
Azure Workbook
4f2bd3e1-f43d-4397-a880-d5ba00da689d (VM At
Azure Workbook
AzureMonitorWindowsAgent (main-vm-1/AzureM
microsoft.compute/virtualMachines/exten...
Data-Windows
Data collection rule
Logs-SOC-Lab
Log Analytics workspace
MAIN-VM-1
Virtual machine
MAIN-VM-1-ip
Public IP address
MAIN-VM-1-nsg
Network security group
main-vm-1959_z1
Network interface
MAIN-VM-1_OsDisk_1_c5f2ab9721384dd3bff400(
Disk
SecurityInsights(logs-soc-lab)
Solution
Vnet-SOC-Lab
Virtual network
Apply force delete for selected Virtual machines and Virtual machine scale sets
Enter resource group name to confirm deletion *
LR-SOC-Lab
Delete
Cancel

Final Things Learned

Completing this project reinforced my understanding of cloud-based security monitoring and SIEM operations. I gained practical insights into:

- Building and securing Azure environments while deliberately exposing a honeypot for research.
- Using Sentinel and KQL to detect, enrich, and visualise attack data.
- How real-world brute-force and probing activity appears in logs.
- The importance of contextual enrichment (watchlists, geolocation) for actionable threat analysis.
- The end-to-end process of turning raw logs into meaningful security intelligence.

This project not only strengthened my Azure and Sentinel skills but also mirrored workflows commonly used in professional security operations, providing a strong foundation for practical threat hunting and incident response.