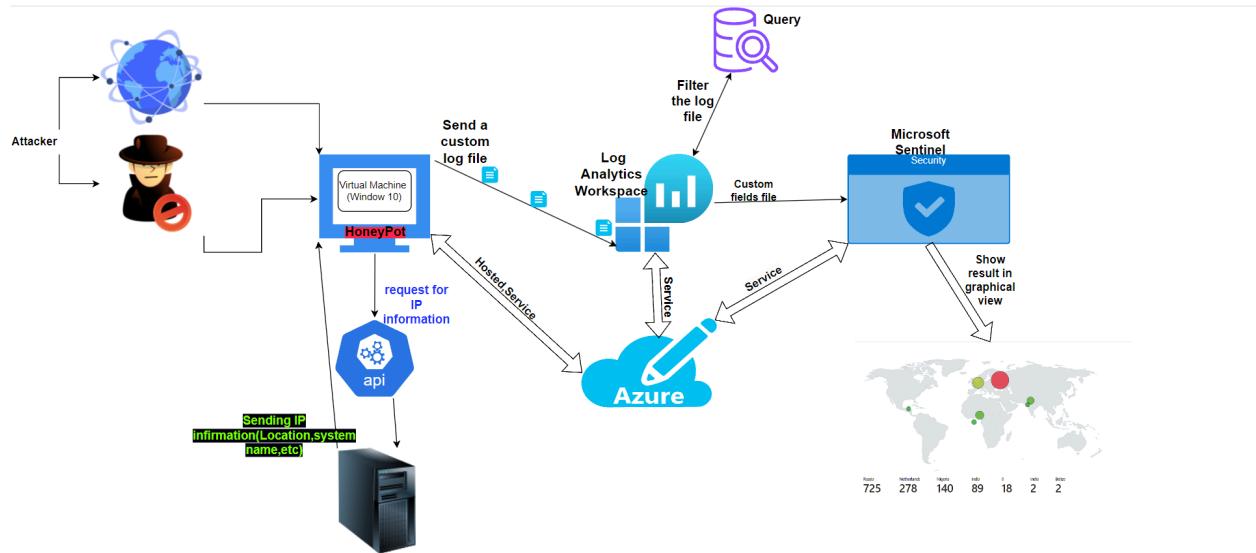


Workflow of software



Initially, we are in the process of setting up a virtual machine (VM) to serve as a honeypot in subsequent operations. This particular VM will be deployed live on the internet and deliberately exposed to external access.

The screenshot shows the Microsoft Azure portal interface for managing virtual machines. At the top, there's a navigation bar with 'Microsoft Azure', 'Upgrade' (with a gear icon), a search bar ('Search resources, services, and docs (G+)'), and user information ('mayurrathour521@gmail.com', 'DEFAULT DIRECTORY (MAYURRA...)', and a profile icon). Below the navigation bar, the main content area is titled 'Virtual machines' with a 'Create' button and a three-dot menu icon. A sub-header says 'Default Directory (mayurrathour521@gmail.onmicrosoft.com)'. The main content area displays a message: 'Showing 0 to 0 of 0 records.' followed by 'No virtual machines to display'. It includes a small icon of a computer monitor with a hexagon on it. Below the message, there's a note: 'Create a virtual machine that runs Linux or Windows. Select an image from the marketplace or use your own customized image.' There are also two blue links: 'Learn more about Windows virtual machines' and 'Learn more about Linux virtual machines'. At the bottom right, there's a 'Give feedback' button with a speech bubble icon.

Configuring our VM

Home > Create a virtual machine >

Create a virtual machine

Basics Disks Networking Management Monitoring Advanced Tags Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. [Learn more](#)

Project details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Resource group * [Create new](#)

Instance details
Virtual machine name * Region *

< Previous [Next : Disks >](#) [Review + create](#) [Give feedback](#)

Instance details
Virtual machine name * Region * Availability options Security type Image * [See all images](#) | [Configure VM generation](#)
 This image is compatible with additional security features. [Click here to swap to the Trusted launch security type.](#)

VM architecture Arm64 x64
⚠️ Arm64 is not supported with the selected image.

Run with Azure Spot discount

Now we are in networking part ,
NIC is just like a firewall for our VM

Create a virtual machine

Basics Disks **Networking** Management Monitoring Advanced Tags Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution. [Learn more](#)

Network interface
When creating a virtual machine, a network interface will be created for you.

Virtual network * [Create new](#)
Subnet *
Public IP [Create new](#)
NIC network security group None Basic Advanced
Public inbound ports * None Allow selected ports

Adding an inbound rule to our NIC firewall entails permitting desired traffic types to reach our VM. We aim to facilitate the accessibility of various traffic forms, including TCP, Ping, and Syn scans, thereby ensuring comprehensive connectivity.

The screenshot shows the Azure portal interface for creating a virtual machine. On the left, there's a navigation bar with 'Home > Create a virtual machine >' followed by 'Create network security group ...'. The main area shows a 'honeypot-nsg' network security group with sections for 'Inbound rules' (empty) and 'Outbound rules' (empty). A modal window titled 'Add inbound security rule' is open, showing the configuration for a new rule:

- Name ***: honeypot-nsg
- Inbound rules**: No results.
- + Add an inbound rule**
- Protocol**: Any (radio button selected)
- Action**: Allow (radio button selected)
- Priority ***: 100
- Name ***: AllowAnyCustomAnyInbound
- Description**: Allowing traffic from any port to any port
- OK** button (highlighted in blue)
- Add** and **Cancel** buttons
- Give feedback** link

VM all config details

Basics

Subscription : Free Trial

Resource group : (new) honeypotlab69

Virtual machine name : honeypot

Region : South Africa North

Availability options : No infrastructure redundancy required

Security type : Standard

Image : Windows 10

VM architecture : x64

Size : Standard B1s (1 vcpu, 1 GiB memory)

Enable Hibernation (preview) : No

Username : XYZ

Already have a Windows license? : Yes

License type : Windows Client

Azure Spot : No

Disks

OS disk size : Image default
OS disk type : Premium SSD LRS
Use managed disks : Yes
Delete OS disk with VM : Enabled
Ephemeral OS disk : No

Networking

Virtual network :: (new) honeypot-vnet
Subnet :: (new) default (10.0.0.0/24)
Public IP :: (new) honeypot-ip
NIC network security group :: (new) honeypot-nsg
Accelerated networking :: Off
Place this virtual machine behind an existing load balancing solution? :: No
Delete public IP and NIC when VM is deleted :: Disabled

Management

Microsoft Defender for Cloud :: None
System assigned managed identity :: Off
Login with Microsoft Entra ID :: Off
Auto-shutdown :: Off
Enable hotpatch :: Off
Patch orchestration options :: OS-orchestrated patching: patches will be installed by OS

Monitoring

Alerts :: Off
Boot diagnostics :: On
Enable OS guest diagnostics :: Off
Enable application health monitoring :: Off

Advanced

Extensions :: None
VM applications :: None
Cloud init :: No
User data :: No
Disk controller type :: SCSI
Proximity placement group :: None
Capacity reservation group :: None

Now we are creating a Log Analytics Workspace

- We are going to ingest log from windows and make a custom log's

Home >

Log Analytics workspaces

Default Directory (mayurraethur521@gmail.onmicrosoft.com)

+ Create Open recycle bin Manage view Refresh Export to CSV Open query Assign tags

Filter for any field... Subscription equals all Resource group equals all Location equals all Add filter

Showing 0 to 0 of 0 records.

Name ↑ Resource group ↑↓ Location ↑↓ Subscription ↑↓

No grouping List view

 No log analytics workspaces to display

Try changing or clearing your filters.

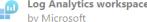
Create log analytics workspace Learn more ↗

Give feedback

Home > Log Analytics workspaces >

Create Log Analytics workspace

Basics Tags Review + Create

 Log Analytics workspace by Microsoft

Basics

Subscription	Free Trial
Resource group	honeypotlab69
Name	Honeypotlog69
Region	South Africa North

Pricing

Pricing tier	Pay-as-you-go (Per GB 2018)
--------------	-----------------------------

The cost of your workspace depends on the volume of data ingested and how long it is retained. Regional pricing details are available on the [Azure Monitor pricing page](#). You can change to a different pricing tier after the workspace is created. [Learn more](#) about Log Analytics pricing models.

Tags

None

Create < Previous Download a template for automation

Submitting deployment...
Submitting the deployment template for resource group 'honeypotlab69'.

Microsoft Azure Upgrade Search resources, services, and docs (G+) mayurrahour521@gmail.com DEFAULT DIRECTORY (MAYURA...)

Home > Microsoft.LogAnalyticsOMS | Overview Deployment

Search Delete Cancel Redeploy Download Refresh

Overview Deployment details Next steps Go to resource

Your deployment is complete

Deployment name : Microsoft.LogAnalyticsOMS
Subscription : Free Trial
Resource group : honeypotlab69

Start time : 4/12/2024, 8:45:00 PM
Correlation ID : 95868d7b-cc67-4759-987c-c815e5e42038

Give feedback Tell us about your experience with deployment

Cost management
Get notified to stay within your budget and prevent unexpected charges on your bill.
[Set up cost alerts >](#)

Microsoft Defender for Cloud
Secure your apps and infrastructure
[Go to Microsoft Defender for Cloud >](#)

Free Microsoft tutorials
[Start learning today >](#)

Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
[Find an Azure expert >](#)

Using security center

- To gather information from our VM

The screenshot shows the Microsoft Azure Security Center interface. The left sidebar includes links for Conditional Access, Identity Protection, Security Center (which is selected), and Verified ID. Under Manage, there are sections for Identity Secure Score, Named locations, Authentication methods, Multifactor authentication, and Certificate authorities. The Troubleshooting + Support section has a 'New support request' link. The main content area displays a message: 'You may be viewing limited information. To get tenant-wide visibility, click here →'. It also says 'No recommendations to display' and 'There are no security recommendations for the selected subscriptions'. A blue button 'View all recommendations in Defender for Cloud' is present. Below this, a section titled 'Most prevalent alerts' states: 'Defender for Cloud uses advanced analytics and global threat intelligence to alert you to malicious activity. Alerts displayed below are from the past 21 days.' It shows a shield icon and the message 'No alerts to display'. A blue button 'View all alerts in Microsoft Defender for Cloud' is shown. A Snipping Tool window is overlaid on the right, showing the same security center interface with the message 'Screenshot copied to clipboard and saved'.

The screenshot shows the Microsoft Azure Settings | Data collection page. The left sidebar includes links for Settings, Defender plans, and Data collection (which is selected). The main content area is titled 'Store additional raw data - Windows security events'. It explains that raw events, logs, and additional security data can be collected and saved to a Log Analytics workspace. It asks to select the level of data to store (with 'None' as the default). Below this, four event collection options are listed: 'All Events' (selected), 'Common', 'Minimal', and 'None'. Each option has a brief description. A 'Save' button is located at the top right of the main content area.

Connecting our VM to log analytical service

The screenshot shows the Microsoft Azure portal interface. At the top, there's a blue header bar with the Microsoft Azure logo, an 'Upgrade' button, a search bar containing 'Search resources, services, and docs (G+ /)', and user information ('mayurrathour521@gma...'). Below the header are several icons: a square with a dot, a gear, a question mark, a refresh symbol, and a person icon.

The main content area shows a virtual machine named 'honeypot' under the 'Log Analytics workspaces' section. It indicates the machine is a 'Virtual machine'. There are three buttons at the top of the card: 'Connect' (with a gear icon), 'Disconnect' (with a gear icon), and 'Refresh' (with a circular arrow icon). A progress bar below the buttons shows the status as 'Connecting...' with an info icon.

Under the 'Status' section, it says 'Connecting'. In the 'Message' section, it states 'Workspace Name Honeypotlog69' and 'Connecting VM to Log Analytics. Please check back later for status update.'

- Now we are going to configure Microsoft sentinel which eventually act as our SIEM in future
- Selecting our machine to be get connected to our Sentinel

No Microsoft Sentinel to display

See and stop threats before they cause harm, with SIEM reinvented for a modern world. Microsoft Sentinel is your birds-eye view across the enterprise.

Create Microsoft Sentinel

Learn more ↗

Give feedback ↗

As evident from the presence of a public IP address assigned to our VM, we can confidently affirm the successful creation of our virtual machine. Furthermore, this connectivity enables seamless integration with both log analytical and Sentinel services, thereby enhancing our operational capabilities.

- Now get got the public ip address for our VM we can connect it remotely with our personal system

Virtual machines >

honeypot Virtual machine

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Connect

Connect

Bastion

Networking

Network settings

Load balancing

Application security groups

Network manager

Settings

Disk

Extensions + applications

Resource group (move)
honeypotlab69

Status
Running

Location
South Africa North

Subscription (move)
Free Trial

Subscription ID
037d1f3e-537c-41aa-b5b2-59f1e5c83cef

Operating system
Windows (Windows 10 Pro)

Size
Standard B1s (1 vcpu, 1 GiB memory)

Public IP address
4.221.187.193

Virtual network/subnet
[honeypot-vnet/default](#)

DNS name
[Not configured](#)

Health state

Tags (edit)
Add tags

Properties Monitoring Capabilities (7) Recommendations Tutorials

Virtual machine

Computer name
honeypot

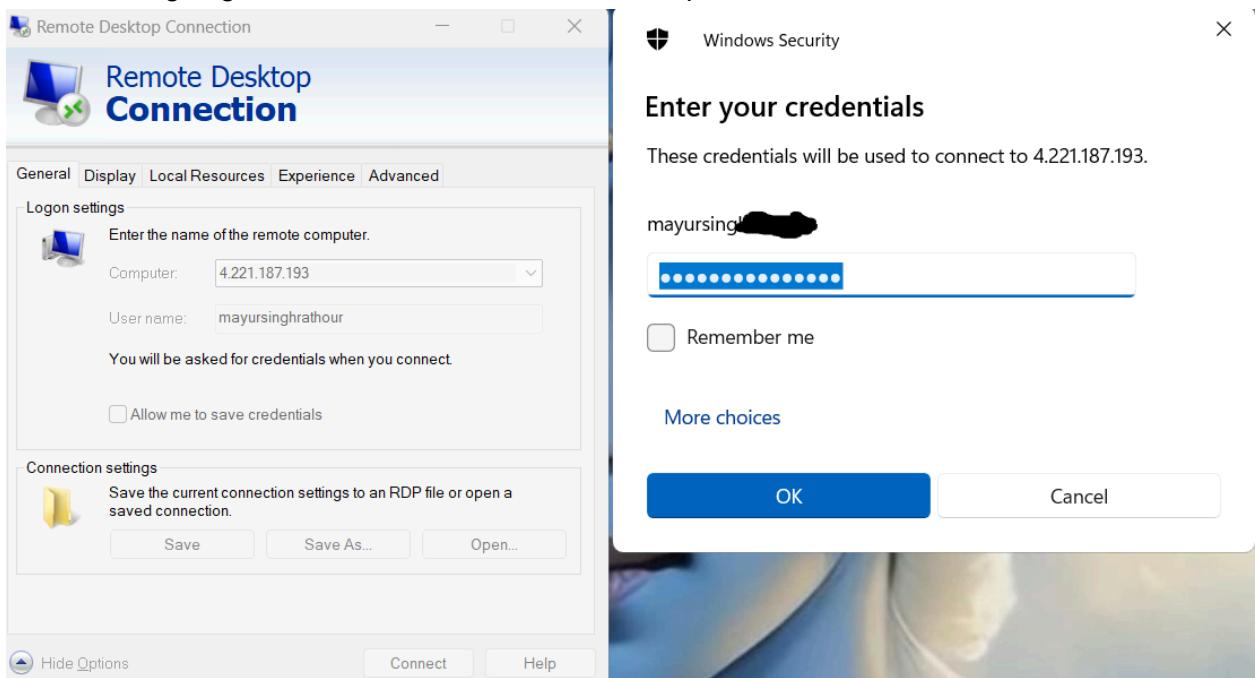
Operating system
Windows (Windows 10 Pro)

Networking

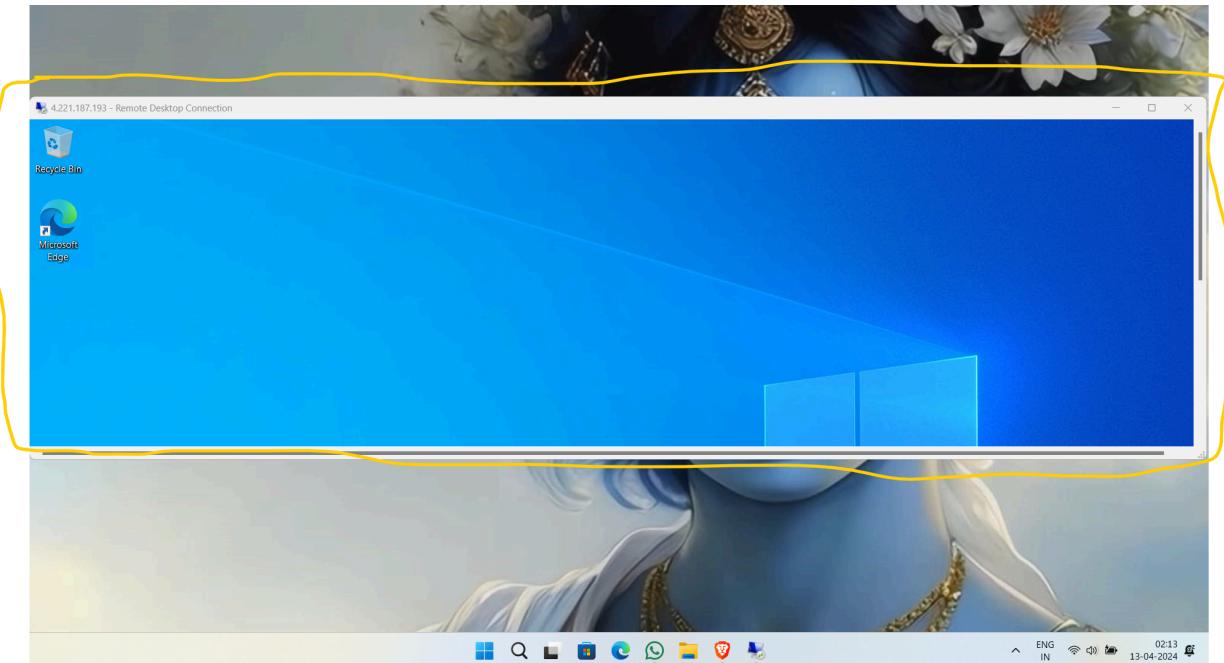
Public IP address
4.221.187.193 (Network interface [honeypot313](#))

Public IP address (IPv6) -

Now we are going to connect VM with remote desktop connection software

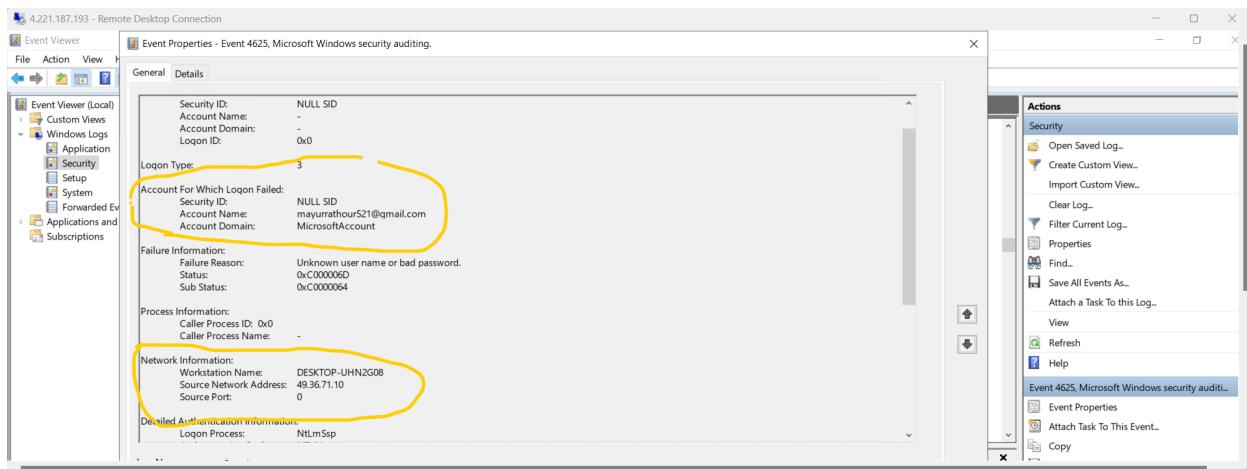


As we can see we have successfully connected windows 10 (VM) to our local system



Upon inspection of the event viewer within our VM system, we have observed a failed attempt to access the VM, revealing the IP address of the originating system involved in the unauthorized access endeavor.

- Why is it important to have the IP address of an attacker? Because with it, we can locate the attacker's location using a third-party service and create a log file with this information.



Our single attacker location (find statically)

The screenshot shows the ipgeolocation.com website. An IP address (49.36.71.10) is entered into the search bar. The resulting JSON data is displayed below:

```

{
  "ip": "49.36.71.10",
  "country_name": "India",
  "state_prov": "Gujarat",
  "city": "Ahmedabad",
  "latitude": "23.02251",
  "longitude": "72.57136",
  "time_zone": "Asia/Kolkata",
  "isp": "Reliance Jio Infocomm Limited",
  "currency": "Indian Rupee",
  "country_flag": "INDIA"
}

```

Now we are going to turn off our VM firewall so anyone can send request to it

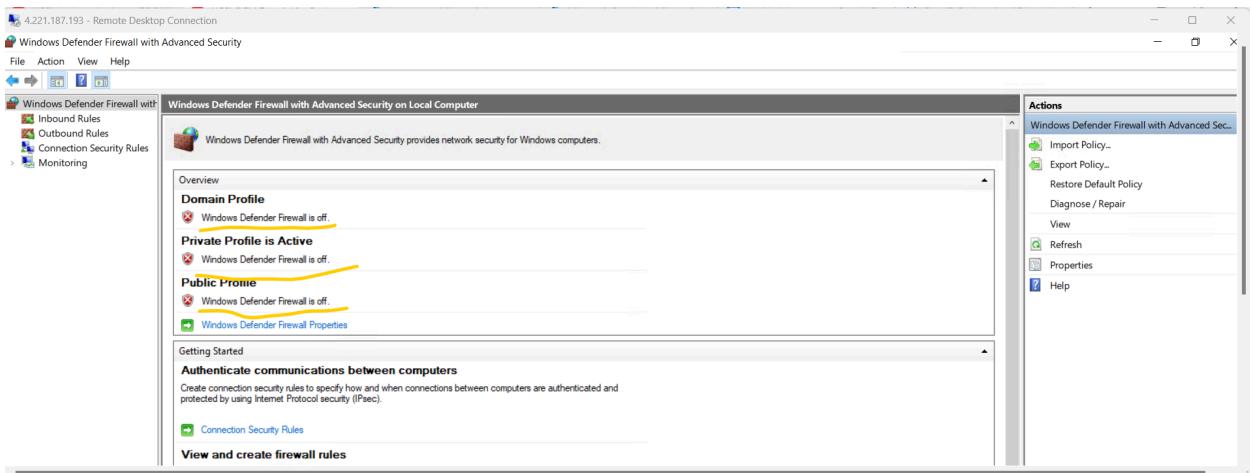
- As we can see firewall is on our VM system

```
Command Prompt - ping 4.221.187.193 X + v

C:\Users\mayur>ping 4.221.187.193 -t

Pinging 4.221.187.193 with 32 bytes of data:
Request timed out.
```

- Now our firewall is off and can accept any kind of traffic from remote location



```
C:\Users\mayur>ping 4.221.187.193 -t

Pinging 4.221.187.193 with 32 bytes of data:
Reply from 4.221.187.193: bytes=32 time=293ms TTL=98
Reply from 4.221.187.193: bytes=32 time=296ms TTL=98
Reply from 4.221.187.193: bytes=32 time=301ms TTL=98
Reply from 4.221.187.193: bytes=32 time=294ms TTL=98
Reply from 4.221.187.193: bytes=32 time=296ms TTL=98
Reply from 4.221.187.193: bytes=32 time=295ms TTL=98
Reply from 4.221.187.193: bytes=32 time=294ms TTL=98
Reply from 4.221.187.193: bytes=32 time=294ms TTL=98
Reply from 4.221.187.193: bytes=32 time=295ms TTL=98
Reply from 4.221.187.193: bytes=32 time=294ms TTL=98
Reply from 4.221.187.193: bytes=32 time=293ms TTL=98
Reply from 4.221.187.193: bytes=32 time=295ms TTL=98
Reply from 4.221.187.193: bytes=32 time=308ms TTL=98
Reply from 4.221.187.193: bytes=32 time=295ms TTL=98
Reply from 4.221.187.193: bytes=32 time=295ms TTL=98
Reply from 4.221.187.193: bytes=32 time=296ms TTL=98
Reply from 4.221.187.193: bytes=32 time=295ms TTL=98
Reply from 4.221.187.193: bytes=32 time=293ms TTL=98
Reply from 4.221.187.193: bytes=32 time=295ms TTL=98
Reply from 4.221.187.193: bytes=32 time=295ms TTL=98
Reply from 4.221.187.193: bytes=32 time=32ms TTL=98
Reply from 4.221.187.193: bytes=32 time=356ms TTL=98
Reply from 4.221.187.193: bytes=32 time=295ms TTL=98
Reply from 4.221.187.193: bytes=32 time=294ms TTL=98
Reply from 4.221.187.193: bytes=32 time=294ms TTL=98
Reply from 4.221.187.193: bytes=32 time=293ms TTL=98

Reply from 4.221.187.193: bytes=32 time=296ms TTL=98
Reply from 4.221.187.193: bytes=32 time=294ms TTL=98

Ping statistics for 4.221.187.193:
    Packets: Sent = 56, Received = 56, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 293ms, Maximum = 356ms, Average = 298ms
```

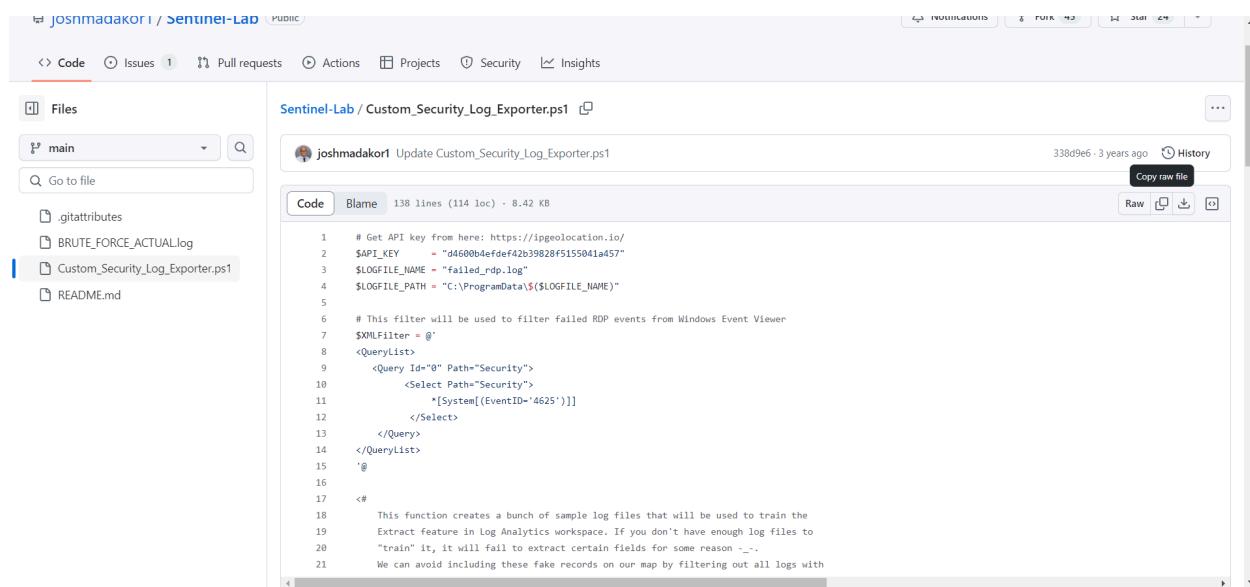
Now we are going to take help of a github repo, which will automatically take the failed audit event from our **Event viewer** we will send those event to to our 3rd party website and that website is going to send us information about that ip address for example Location,ISP,Logituted and etc

Github repo link -

https://github.com/joshmadakor1/Sentinel-Lab/blob/main/Custom_Security_Log_Exporter.ps1

IP geolocation finder link -

<https://ipgeolocation.io>



The screenshot shows a GitHub repository page for 'joshmadakor1 / Sentinel-Lab'. The repository is public. The main page displays a list of files, with 'Custom_Security_Log_Exporter.ps1' being the selected file. The code editor shows the PowerShell script content:

```
1 # Get API key from here: https://ipgeolocation.io/
2 $API_KEY      = "d4d00b4efdef42b39828f5155041a457"
3 $LOGFILE_NAME = "failed_rdp.log"
4 $LOGFILE_PATH = "C:\ProgramData\($LOGFILE_NAME)"
5
6 # This filter will be used to filter failed RDP events from Windows Event Viewer
7 $XMLFilter = @'
8 <QueryList>
9   <Query Id="0" Path="Security">
10     <Select Path="Security">
11       *[System[({EventID='4625'})]]
12     </Select>
13   </Query>
14 </QueryList>
15 '@
16
17 <#
18   This function creates a bunch of sample log files that will be used to train the
19   Extract feature in Log Analytics workspace. If you don't have enough log files to
20   "train" it, it will fail to extract certain fields for some reason _.
21   We can avoid including these fake records on our map by filtering out all logs with
```

We will be needed and API key of this geolocation website to make our github code work properly

- We can find API from here

The screenshot shows the homepage of ipgeolocation.io. At the top, there's a navigation bar with links for Products, My IP Location, Pricing, Documentation, Blog, Sign Up, and Sign In. Below the navigation is a search bar with the placeholder "Enter any IPv4, IPv6 address or domain name:" containing the IP address "49.36.71.10". To the right of the search bar is a magnifying glass icon. On the left side of the main content area, there's a section titled "Free IP Geolocation API and Accurate IP Lookup Database" with a brief description of the service. In the center, there's a large button labeled "Get Free API Access" which is circled in yellow. To the right of the button, the results for the IP address are displayed in a JSON-like format:

```
"ip": "49.36.71.10",
"country_name": "India",
"state_prov": "Gujarat",
"city": "Ahmedabad",
"latitude": "23.02251",
"longitude": "72.57136",
"time_zone": "Asia/Kolkata",
"isp": "Reliance Jio Infocomm Limited",
"currency": "Indian Rupee",
"country_flag": "INDIA FLAG IMAGE"
```

At the bottom right of the results area is a blue "View More" button.

We have to run our git code in VM powershell ISE

- As we can see we have successfully run the code and got desired output
- In the following img we can see someone from netherlands is trying to get access of our system by bruteforce
- With free API key we can only have 1000 req per day

The screenshot shows a Windows PowerShell ISE window titled "Untitled1.ps1*". The code in the editor is a PowerShell script that attempts to log into a honeypot. It includes an API key, logs to a file, and uses an infinite loop to check event viewer logs. A yellow arrow points to the API key line with the text "KEY". A yellow checkmark with the text "ATTEMPT" points to the log entries in the output pane. The log entries show multiple failed login attempts from an IP address in North Holland, Netherlands, with timestamps and source host details.

```
# Get API key from here: https://ipgeolocation.io/
$API_KEY = "6a6593d2c6bd4b95b53b718e5e216fb"
$logFile_Name = "Failed_rdp.log"
$logFile_Path = "C:\ProgramData\$($LogFile_Name)"

# This filter will be used to filter failed RDP events from Windows Event Viewer
$XMLFilter = ...
$Function write-Sample-Log() {}

# This block of code will create the log file if it doesn't already exist
if ((Test-Path $LogFile_Path) -eq $false) {}

# Infinite Loop that keeps checking the Event Viewer logs.
while ($true)
{...}

Latitude:52.37018,longitude:4.87324,destinationhost:honeypot,username:ADMIN,sourcehost:87.251.75.145,state:North Holland,label:Netherlands - 87.251.75.145,timestamp:2024-04-12 21:50:39
Latitude:52.37018,longitude:4.87324,destinationhost:honeypot,username:ADMIN,sourcehost:87.251.75.145,state:North Holland,label:Netherlands - 87.251.75.145,timestamp:2024-04-12 21:50:34
Latitude:52.37018,longitude:4.87324,destinationhost:honeypot,username:ADMIN,sourcehost:87.251.75.145,state:North Holland,label:Netherlands - 87.251.75.145,timestamp:2024-04-12 21:50:33
Latitude:52.37018,longitude:4.87324,destinationhost:honeypot,username:ADMINISTRATOR,sourcehost:87.251.75.145,state:North Holland,label:Netherlands - 87.251.75.145,timestamp:2024-04-12 21:50:32
Latitude:52.37018,longitude:4.87324,destinationhost:honeypot,username:ADMINISTRATOR,sourcehost:87.251.75.145,state:North Holland,label:Netherlands - 87.251.75.145,timestamp:2024-04-12 21:50:32
```

Now we are done at generating log file in our VM so now we are going to connect that log file with our microsoft log analytics workspace

- As we can see our honeypot group inside log analytics which we created in starting

The screenshot shows the Microsoft Azure Log Analytics workspaces page. At the top, there's a search bar with placeholder text 'Search resources, services, and docs (G+/-)'. The main title is 'Log Analytics workspaces' with a '...' button. Below it, a sub-header says 'Default Directory (mayurathour521@gmail.com.microsoft.com)'. A toolbar includes buttons for '+ Create', 'Open recycle bin', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', and 'Assign tags'. There are also filter buttons for 'Subscription equals all', 'Resource group equals all', and 'Location equals all', along with an 'Add filter' button. The results section shows 'Showing 1 to 1 of 1 records.' with one item listed: 'Name' (HoneyPotLog69), 'Resource group' (honeypotlab69), 'Location' (South Africa North), and 'Subscription' (Free Trial). The bottom right corner has grouping and list view options.

Name	Resource group	Location	Subscription
HoneyPotLog69	honeypotlab69	South Africa North	Free Trial

Now we are going to create our own custom log

- But as we can see that it is as for now we are unable to connect our VM log file to workspace so we are going to create duplicate log file in our own system

We have successfully loaded our log file to our cloud workspace

The screenshot shows the 'Create a custom log' wizard in the Microsoft Azure portal. The current step is 'Record delimiter'. The 'Record delimiter' section is selected, with the 'New line' option chosen. A preview window shows several log entries separated by new lines. At the bottom, there are 'Previous' and 'Next' buttons.

Now we have to connect our VM log file to our analytical workspace

- We have to put exact location of our log file which we had created in our VM(windows)

The screenshot shows the 'Create a custom log' wizard in the Microsoft Azure portal. The current step is 'Collection paths'. The 'Collection paths' section is selected. A table shows a single entry with 'Type' set to 'Windows' and 'Path' set to 'C:\ProgramData\failed_rdp.log'. At the bottom, there are 'Previous' and 'Next' buttons.

Finally creating our custom log file

Sample log name: failed_rdp.log

Record delimiter: New line

Collection paths: Windows folder C:\ProgramData\failed_rdp.log

Custom log name: Failed_rdp_CL

Description: Failed login attempt log file

« Previous Create

- Created successfully

Succeeded
Custom log creation succeeded

Now we have to load the data from failed_rdp.log file to our query section

- It will take 30 min to load for the first time

TimeGenerated [UTC]	Computer	RawData	Type	_ResourceId
4/14/2024, 6:15:42.519 AM	honeypot	latitude:47.91542,longitude:-120.60306,destinationhostsample...	FAILED_RDP_WITH_GEO_C1	/subscriptions/037d1f3e-537c-41aa-b5b2-59fe5c83cef...
4/14/2024, 6:15:42.519 AM	honeypot	latitude:-22.90906,longitude:-47.06455,destinationhostsam...	FAILED_RDP_WITH_GEO_C1	/subscriptions/037d1f3e-537c-41aa-b5b2-59fe5c83cef...
4/14/2024, 6:15:42.519 AM	honeypot	latitude:52.37022,longitude:4.89517,destinationhostsample...	FAILED_RDP_WITH_GEO_C1	/subscriptions/037d1f3e-537c-41aa-b5b2-59fe5c83cef...
4/14/2024, 6:15:42.519 AM	honeypot	latitude:40.71455,longitude:-74.00714,destinationhostsample...	FAILED_RDP_WITH_GEO_C1	/subscriptions/037d1f3e-537c-41aa-b5b2-59fe5c83cef...
4/14/2024, 6:15:42.519 AM	honeypot	latitude:33.99762,longitude:-6.84737,destinationhostsample...	FAILED_RDP_WITH_GEO_C1	/subscriptions/037d1f3e-537c-41aa-b5b2-59fe5c83cef...
4/14/2024, 6:15:42.519 AM	honeypot	latitude:-5.32558,longitude:100.28595,destinationhostsample...	FAILED_RDP_WITH_GEO_C1	/subscriptions/037d1f3e-537c-41aa-b5b2-59fe5c83cef...
4/14/2024, 6:15:42.519 AM	honeypot	latitude:41.05722,longitude:28.84926,destinationhostsample...	FAILED_RDP_WITH_GEO_C1	/subscriptions/037d1f3e-537c-41aa-b5b2-59fe5c83cef...

Now we are going to create custom fields and extract raw data from custom log

- With help of this query we are going to separate the data from one column to multiple column
- **To do so we are going to type KQL logic for custom fields**

FAILED_LOG_GEO_LC_CL

```
|extend username = extract(@"username:([^,]+)", 1, RawData),
    timestamp = extract(@"timestamp:([^,]+)", 1, RawData),
    latitude = extract(@"latitude:([^,]+)", 1, RawData),
    longitude = extract(@"longitude:([^,]+)", 1, RawData),
    sourcehost = extract(@"sourcehost:([^,]+)", 1, RawData),
    state = extract(@"state:([^,]+)", 1, RawData),
    label = extract(@"label:([^,]+)", 1, RawData),
    destination = extract(@"destinationhost:([^,]+)", 1, RawData),
    country = extract(@"country:([^,]+)", 1, RawData)
|where destination != "samplehost"
|where sourcehost != ""
|summarize event_count=count() by timestamp, label, country, state, sourcehost,
    username, destination, longitude, latitude
```

The screenshot shows the Microsoft Azure Log Analytics workspace interface. The top navigation bar includes 'Microsoft Azure', 'Upgrade', a search bar, and user information. Below the header, the URL 'Honeypotlog69' is visible. The main area displays a query titled 'Honeypotlog69 | Logs'. The query itself is:

```
1 FAILED_RDP_WITH_GEO_CL
2 |extend username = extract(@"username:([^,]+)", 1, RawData),
3     timestamp = extract(@"timestamp:([^,]+)", 1, RawData),
4     latitude = extract(@"latitude:([^,]+)", 1, RawData),
5     longitude = extract(@"longitude:([^,]+)", 1, RawData),
6     sourcehost = extract(@"sourcehost:([^,]+)", 1, RawData),
7     state = extract(@"state:([^,]+)", 1, RawData),
8     label = extract(@"label:([^,]+)", 1, RawData),
9     destination = extract(@"destinationhost:([^,]+)", 1, RawData),
10    country = extract(@"country:([^,]+)", 1, RawData)
11 |where destination != "samplehost"
12 |where sourcehost != ""
13 |summarize event_count=count() by timestamp, label, country, state, sourcehost, username, destination, longitude, latitude
```

The results table shows the following data:

latitude	timestamp	label	country	state	sourcehost	username	destination	longitude	event_count
>	2024-04-12 21:48:13	-87.251.75.145			87.251.75.145	ADMIN	honeypot	1	
>	2024-04-12 21:48:12	-87.251.75.145			87.251.75.145	ADMIN	honeypot	1	
>	2024-04-12 21:48:11	-87.251.75.145			87.251.75.145	ADMINISTRATOR	honeypot	1	
>	2024-04-12 21:48:07	-87.251.75.145			87.251.75.145	ADMIN	honeypot	1	
>	2024-04-12 21:48:06	-87.251.75.145			87.251.75.145	ADMINISTRATOR	honeypot	1	

At the bottom of the results table, it says '2s 29ms | Display time (UTC+00:00) | Query details | 1 - 6 of 143'.

Now we going to Microsoft Sentinel for our final steps

The screenshot shows the Microsoft Sentinel search interface. At the top, there's a header bar with the Microsoft Azure logo, an 'Upgrade' button, a search bar ('Search resources, services, and docs (G+)'), and a user profile ('mayurrao@outlook.com'). Below the header is a toolbar with buttons for 'Create', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', and 'View incidents'. A filter bar at the top allows filtering by 'Subscription equals all', 'Resource group equals all', 'Location equals all', and 'Add filter'. The main area displays a single record: 'Honeypotlog69' from the 'honeypotlab69' resource group in 'South Africa North' location, under 'Free Trial' subscription and 'Default Directory'. The record is listed in 'List view'.

As we can see as for now we have no data found for any kind of attack

The screenshot shows the Microsoft Sentinel Overview page. On the left, a navigation sidebar includes sections for General (Overview, Logs, News & guides, Search), Threat management (Incidents, Workbooks, Hunting, Notebooks, Entity behavior, Threat intelligence, MITRE ATT&CK (Preview)), and Content management (Content hub, Repositories (Preview)). The main area features a chart titled 'Last 24 hours' showing event counts over time, with a callout highlighting 'PEAK HOUR' (595 events) and other categories like 'FAILED_RDP...' (154) and 'OTHERS (5)' (236). Below the chart is a map titled 'Potential malicious events' with a message 'No data was found'. To the right, there are three sections: 'Data source anomalies' (SecurityEvent and Usage charts), and 'Democratize ML for your SecOps' (with a callout to 'Unlock the power of AI for security professionals'). A 'Learn More >' link is also present.

Now we will go to workshop

Home > Microsoft Sentinel

Microsoft Sentinel | Workbooks

Selected workspace: 'honeypotlog69'

Search Refresh Add Workbook Guides & Feedback

My workbooks Templates Updates More content at Content hub

General Overview Logs News & guides Search Threat management Incidents Workbooks Hunting Notebooks Entity behavior Threat intelligence MITRE ATT&CK (Preview) Content management Content hub

My workbooks Templates

Microsoft Sentinel Workbooks

What is it?

Workbooks enable you to get instant visualization and analysis of data so that you can know what's happening across all your connected data sources. Workbooks provide you with the full power of tools with tables and charts that are built in to provide you with analytics for your logs and queries. You can either use out-of-the-box (OOTB) workbooks from content hub or customize an installed OOTB workbook by saving them or create a new workbook easily, from scratch or based on an existing workbook.

[Learn more about Workbooks.](#) [Learn more about OOTB content and Content hub.](#)

No workbook selected Select workbook to view more details

https://portal.azure.com/?quickstart=true#view/Microsoft_Azure_Security_Insights/MainMenuBlade~/~/WorkbooksV2/id/%2Fsubscriptions%2F037d1f3e-537c-41aa-b5b2-59f1e5c83cef%2Fresourcegroups%2Fhoneypotlab69%2Fproviders%2Fmicrosoft.securityinsightsarg%2Fsentinel...

We are not going to need default GUI, so we are going to remove it from our workshop

Microsoft Azure Upgrade Search resources, services, and docs (G+)

Home > Microsoft Sentinel | Workbooks

New workbook honeypotlog69

Edit Open Refresh Help Auto refresh: Off

New workbook

Welcome to your new workbook. This area will display text formatted as markdown.

We've included a basic analytics query to get you started. Use the Edit button below each section to configure it or add more sections.

10K 5K 0K

SecurityEvent Heartbeat Failed_rdp_CL Failed_RDP_WITH_GEO_CL Usage Update ProtectionStatus UpdateSummary

9.91 k 702 154 154 48 39 12 4

Now we are going to add our query which we written above to extract common filed

This Azure Sentinel Report has no content.

Use the add button below to add items.

+ Add ▾

- Add text
- Add parameters
- Add links/tabs
- Add query (highlighted)
- Add me
- Add query
- Add group

As we can see we are getting same output for our query in our workbook

Editing query item: query - 0

Log Analytics workspace Logs Query

```
CommonLog = EXTRACTLOG("CommonLog", 2, RawData);
latitude = extract(@"latitude:[^,]+", 1, RawData),
longitude = extract(@"longitude:[^,]+", 1, RawData),
sourcehost = extract(@"sourcehost:[^,]+", 1, RawData),
state = extract(@"state:[^,]+", 1, RawData),
label = extract(@"label:[^,]+", 1, RawData),
destination = extract(@"destinationhost:[^,]+", 1, RawData),
country = extract(@"country:[^,]+", 1, RawData)
|where destination != "samplehost"
|where sourcehost != ""
|summarize event_count=count() by timestamp, label, country, state, sourcehost, username, destination, longitude, latitude
```

timestamp	↑↓ label	↑↓ country	↑↓ state	↑↓ sourcehost	↑↓ username	↑↓ destination	↑↓ longitude	↑↓ latitude	↑↓ event_count↑↓
2024-04-12 21:48:13	-87.251.75.145			87.251.75.145	ADMIN	honeypot			1
2024-04-12 21:48:12	-87.251.75.145			87.251.75.145	ADMIN	honeypot			1
2024-04-12 21:48:11	-87.251.75.145			87.251.75.145	ADMINISTRATOR	honeypot			1

Now we select map to visualization

Editing query item: query - 0

Log Analytics workspace Logs Query

```
CommonLog = EXTRACTLOG("CommonLog", 2, RawData);
latitude = extract(@"latitude:[^,]+", 1, RawData),
longitude = extract(@"longitude:[^,]+", 1, RawData),
sourcehost = extract(@"sourcehost:[^,]+", 1, RawData),
state = extract(@"state:[^,]+", 1, RawData),
label = extract(@"label:[^,]+", 1, RawData),
destination = extract(@"destinationhost:[^,]+", 1, RawData),
country = extract(@"country:[^,]+", 1, RawData)
|where destination != "samplehost"
|where sourcehost != ""
|summarize event_count=count() by timestamp, label, country, state, sourcehost, username, destination, longitude, latitude
```

timestamp	↑↓ label	↑↓ country	↑↓ state	↑↓ sourcehost	↑↓ username	↑↓ destination	↑↓ longitude	↑↓ latitude	↑↓ event_count↑↓
2024-04-14 16:33:32	Russia - 152.89.198.238	Russia	Central Federal District	LAPTOP	honeypot		37.61502	55.75696	
2024-04-14 16:33:29	Russia - 152.89.198.238	Russia	Central Federal District	LAUREN	honeypot		37.61502	55.75696	
2024-04-14 16:33:25	Russia - 152.89.198.238	Russia	Central Federal District	LIBRARY	honeypot		37.61502	55.75696	

Visualization: Map

We set label as country to read data properly

Home > Microsoft Sentinel > Microsoft Sentinel | Workbooks >

New workbook

honeypotlog69

Done Editing Open File Save Print Copy Share Help

```
label = extract(@"label:([^,]+)", 1, RawData),
destination = extract(@"destination:host:([^,]+)", 1, RawData),
country = extract(@"country:([^,]+)", 1, RawData)
|where destination != "samplehost"
|where sourcehost != ""
|summarize event_count=count() by timestamp, label, country, state, sourcehost, username, destination, longitude, latitude
```



0 725 123 97 0 18 2 2

Map Settings

Coloring type

None Thresholds Heatmap

Color by

event_count

Aggregation for color

Sum of values

Color palette

Green to Red

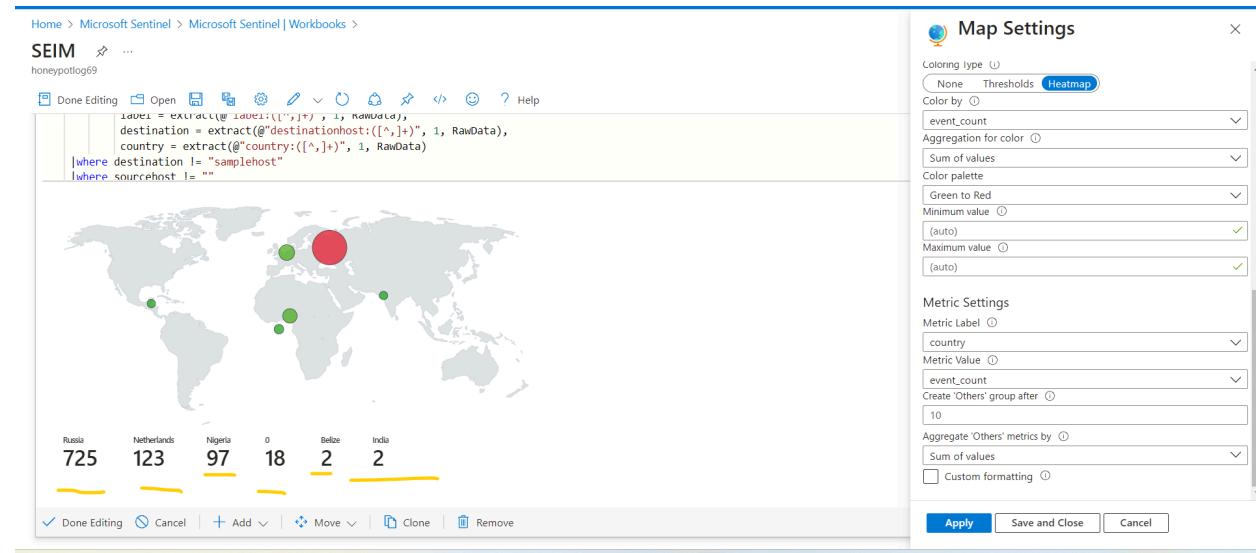
Minimum value

(auto)

Maximum value

(auto)

As we can see the attack from many country has already started



PowerShell should be on if we want to continuous monitor our system

