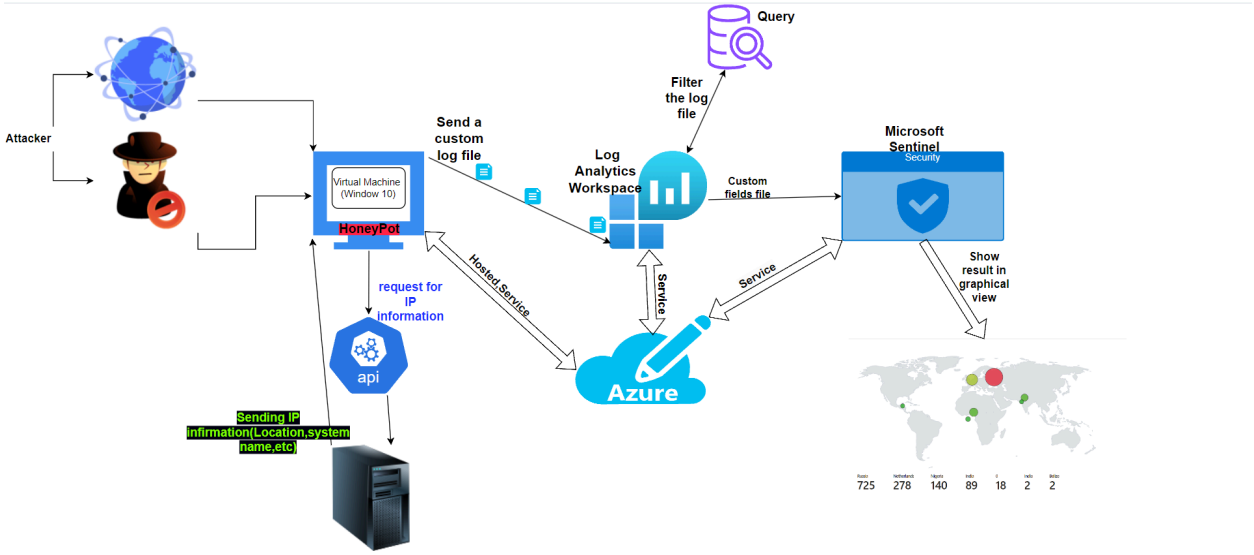
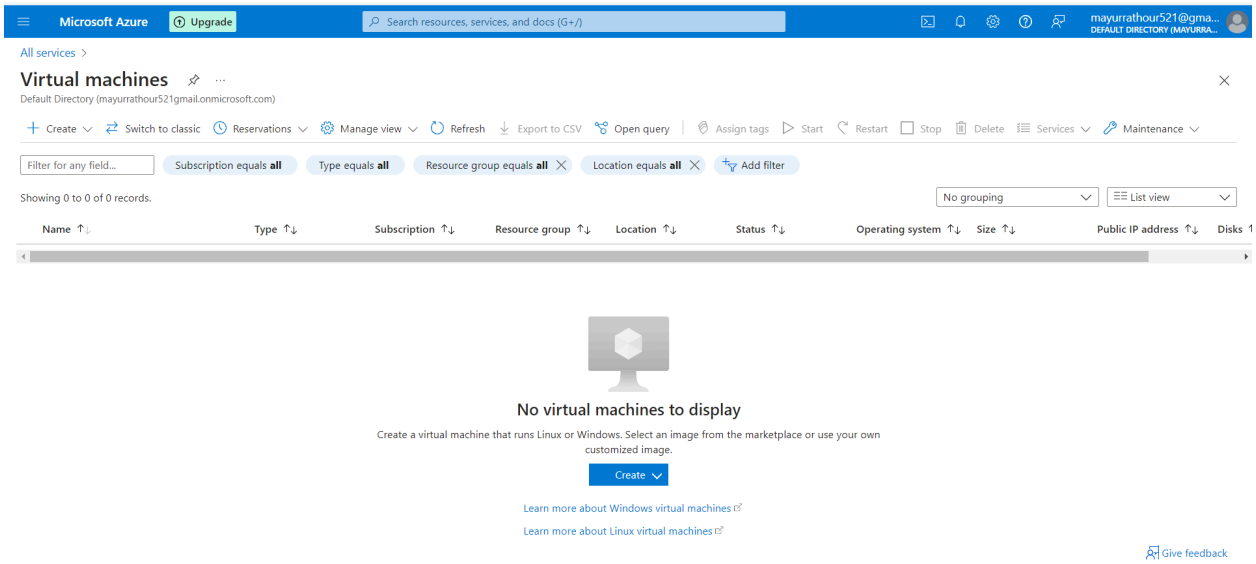
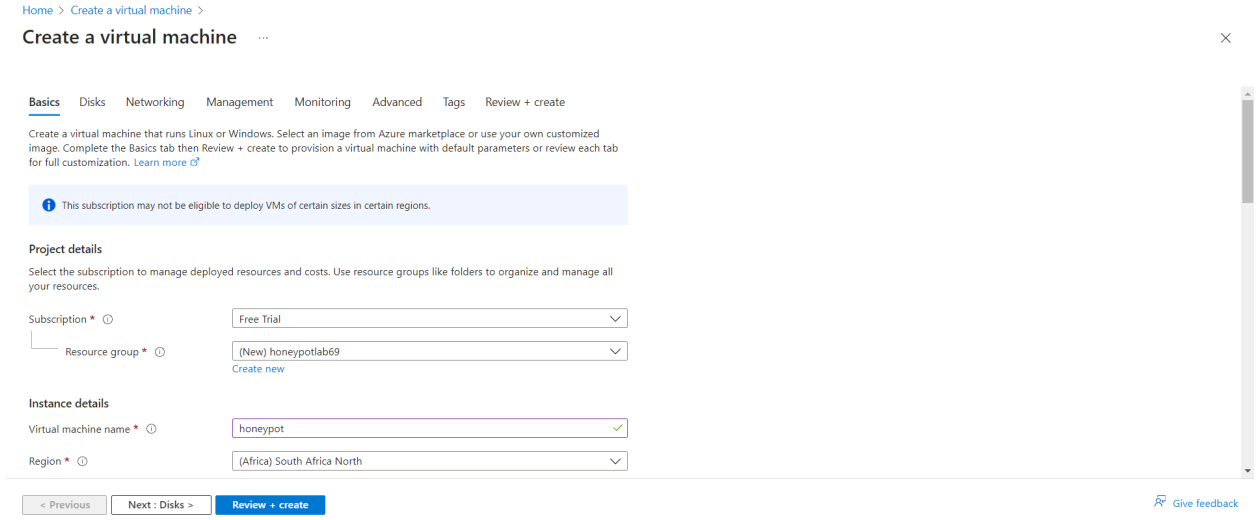
Workflow of software

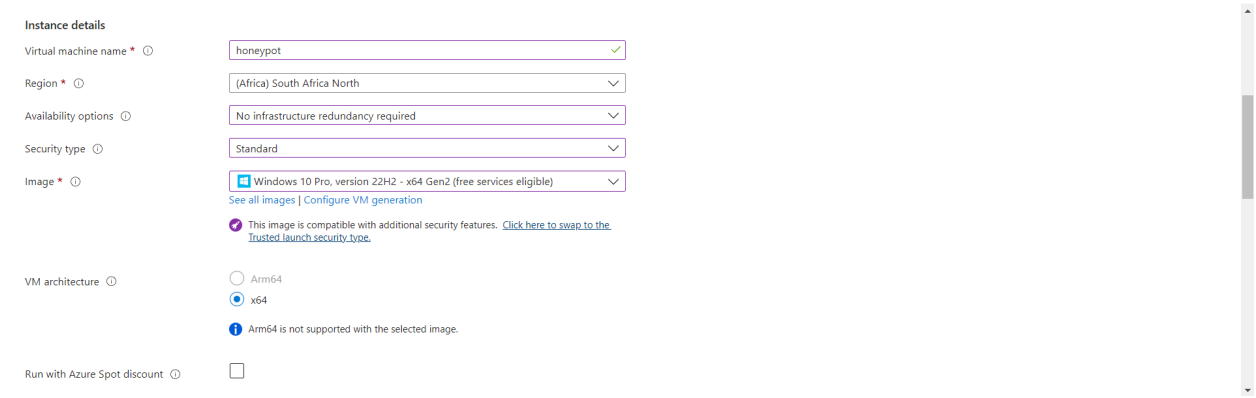


Initially, we are in the process of setting up a virtual machine (VM) to serve as a honeypot in subsequent operations. This particular VM will be deployed live on the internet and deliberately exposed to external access.

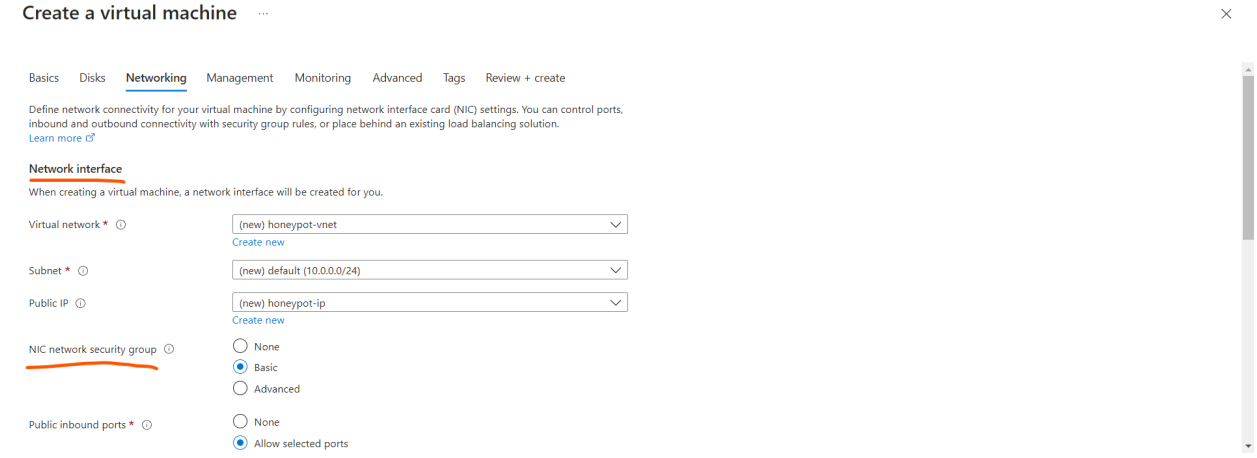


Configuring our VM

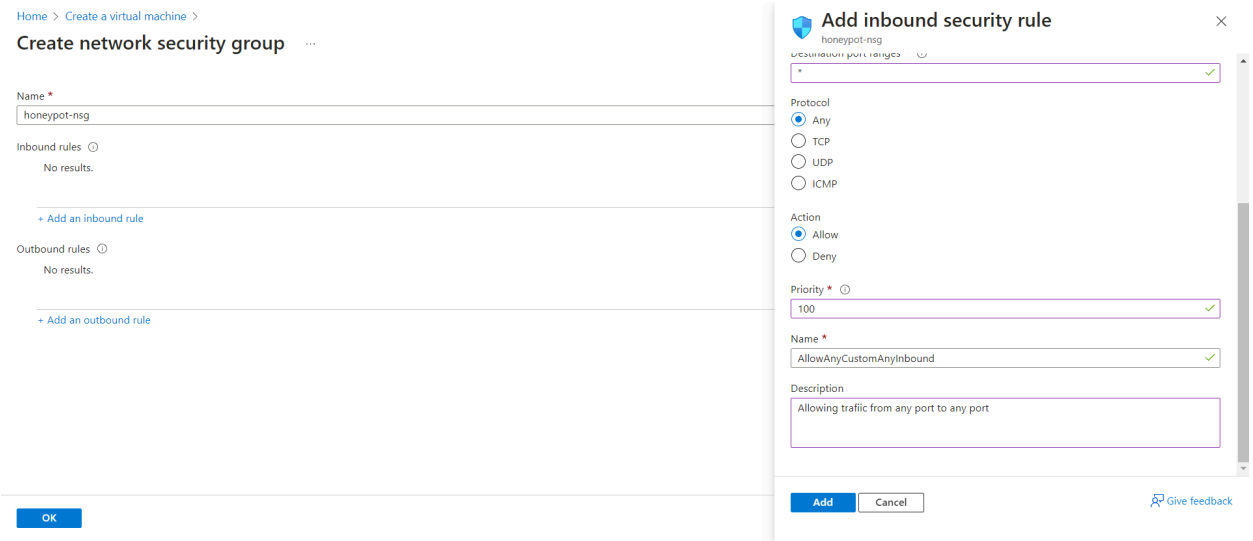




Now we are in networking part ,
NIC is just like a firewall for our VM



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.



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

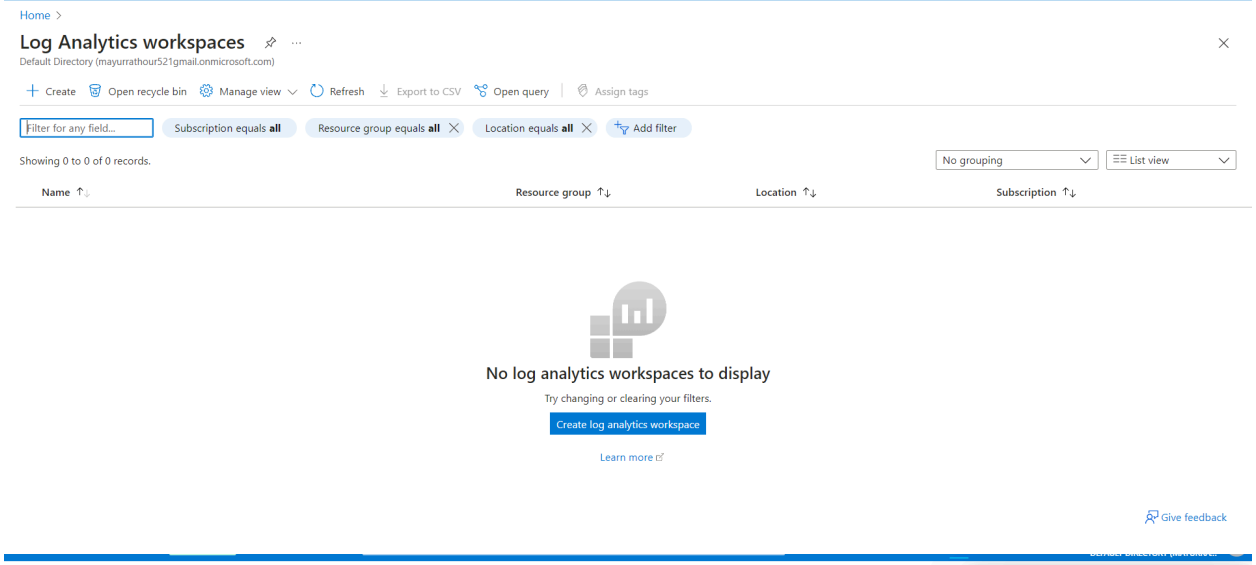
Alerts :: Off
Boot diagnostics :: On
Enable OS guest diagnostics :: Off
Enable application health monitoring :: Off

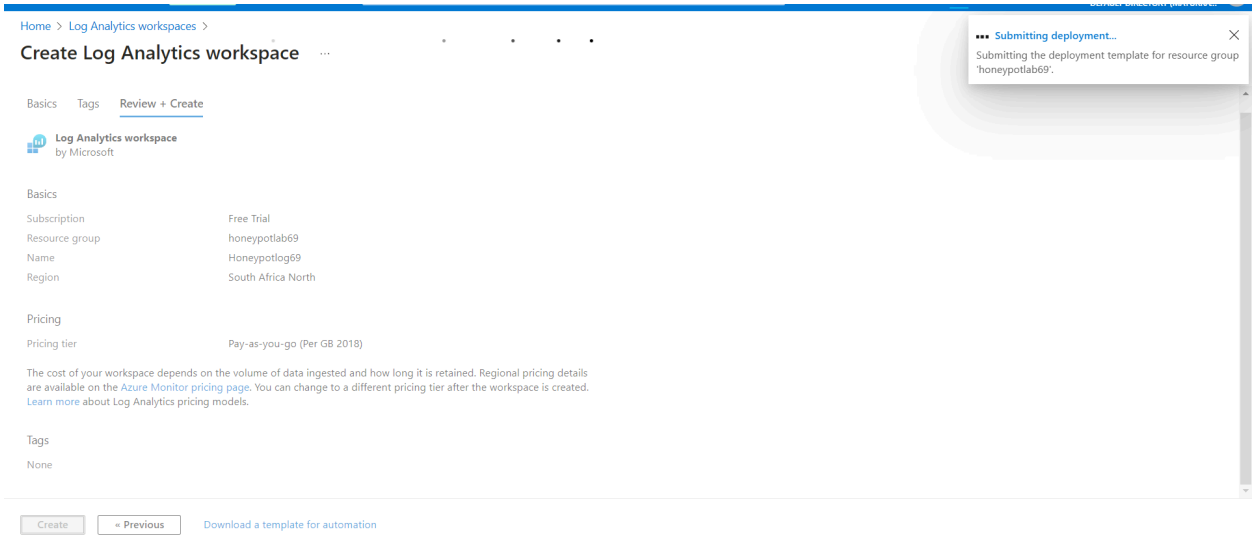
Advanced

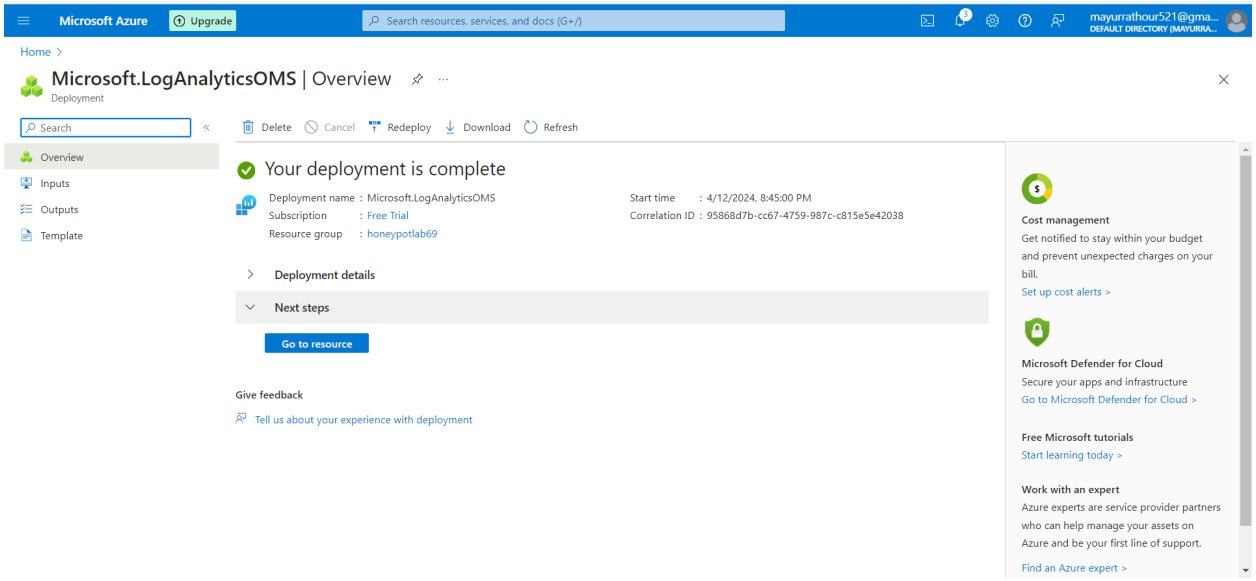
Extensions :: None
VM applications :: None
Cloud init :: No
User data :: No
Disk controller type :: SCSI
Proximity placement group :: None
Capacity reservation group :: None

Now we are creating a Log Analytics Workspace

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

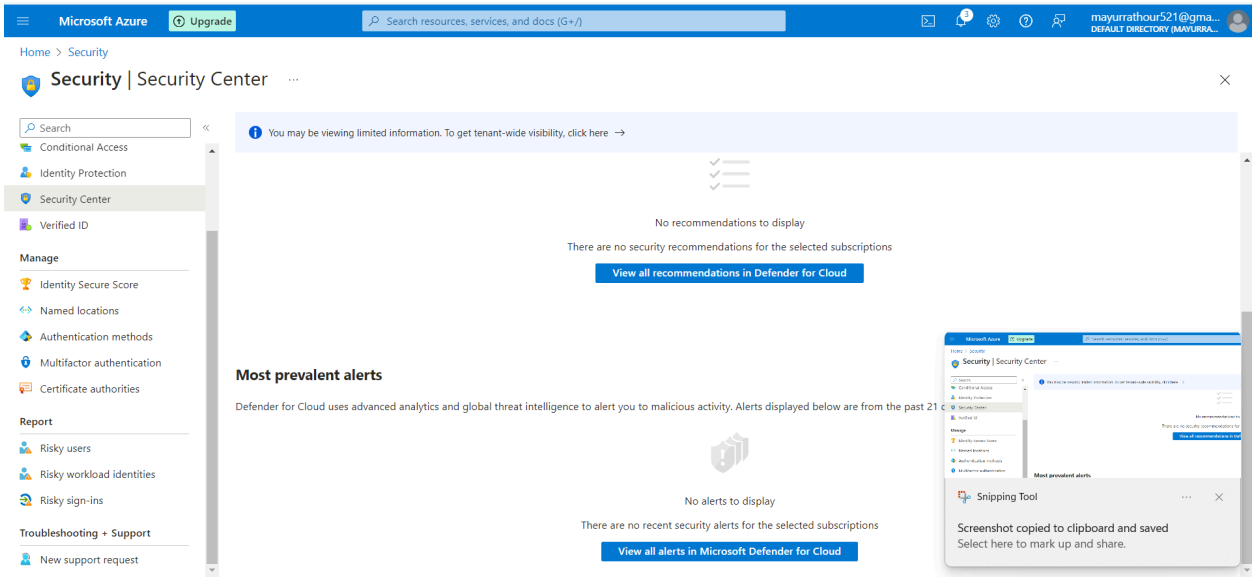


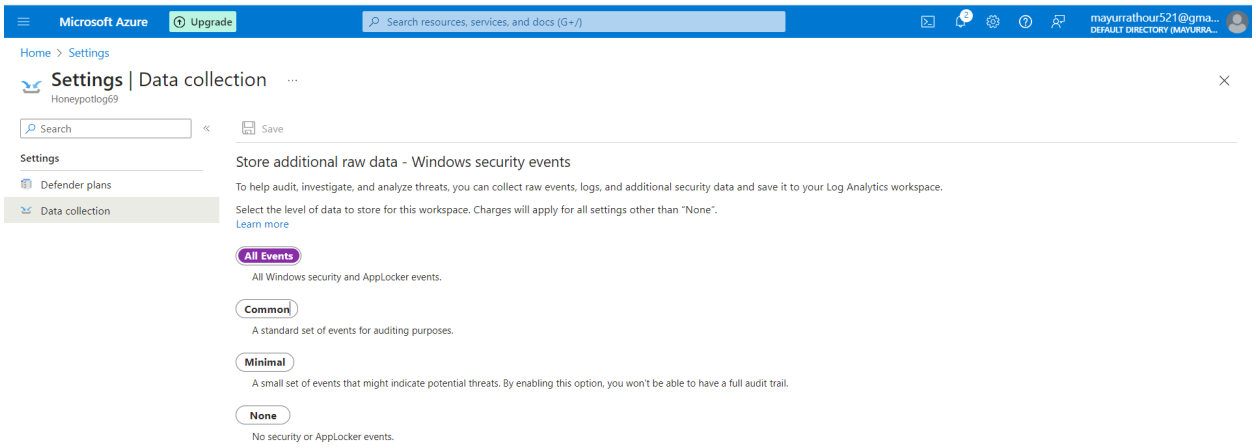



Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
mayurrathour521@gma...
DEFAULT DIRECTORY (MAYURRA...
Home
Microsoft.LogAnalyticsOMS | Overview
Deployment
Search
Overview
Inputs
Outputs
Template
Delete
Cancel
Redeploy
Download
Refresh
Your deployment is complete
Deployment name : Microsoft.LogAnalyticsOMS
Subscription : Free Trial
Resource group : honeypotlab69
Start time : 4/12/2024, 8:45:00 PM
Correlation ID : 95868d7b-cc67-4759-987c-c815e5e42038
Deployment details
Next steps
Go to resource
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 >

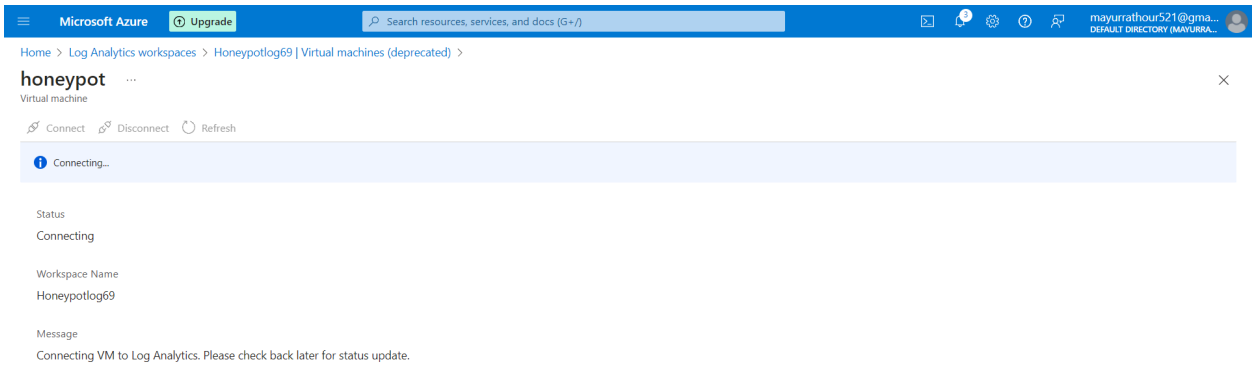
Using security center

- To gather information from our VM



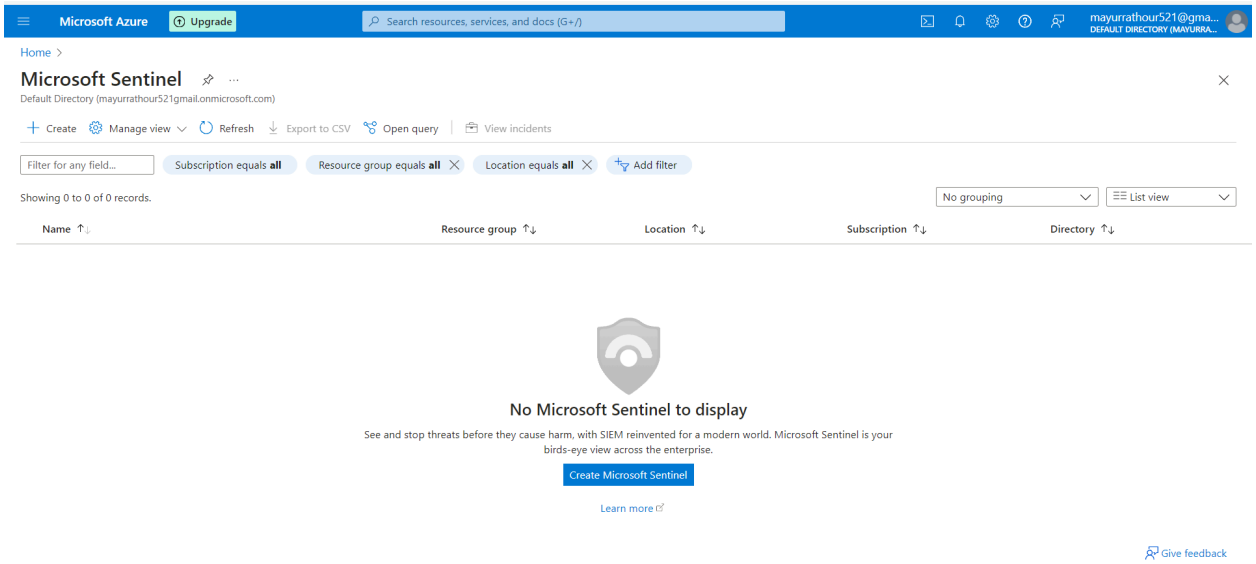


Connecting our VM to log analytical service



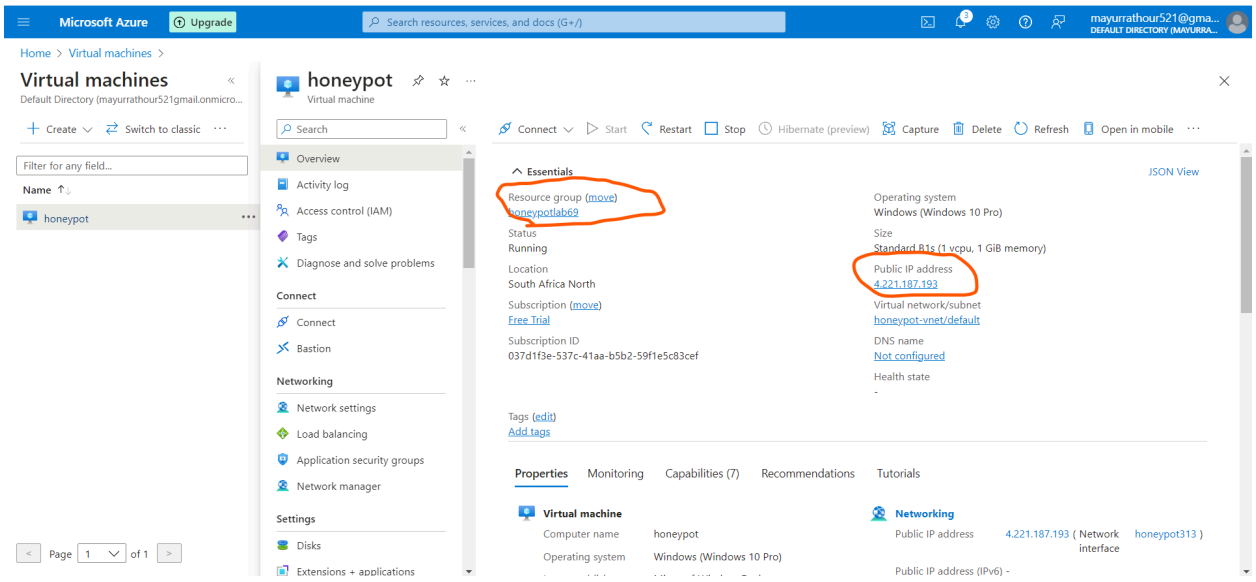
Now we are going to configure Microsoft sentinel which eventually act as our SEIM in future

- Selecting our machine to be get connected to our Sentinel

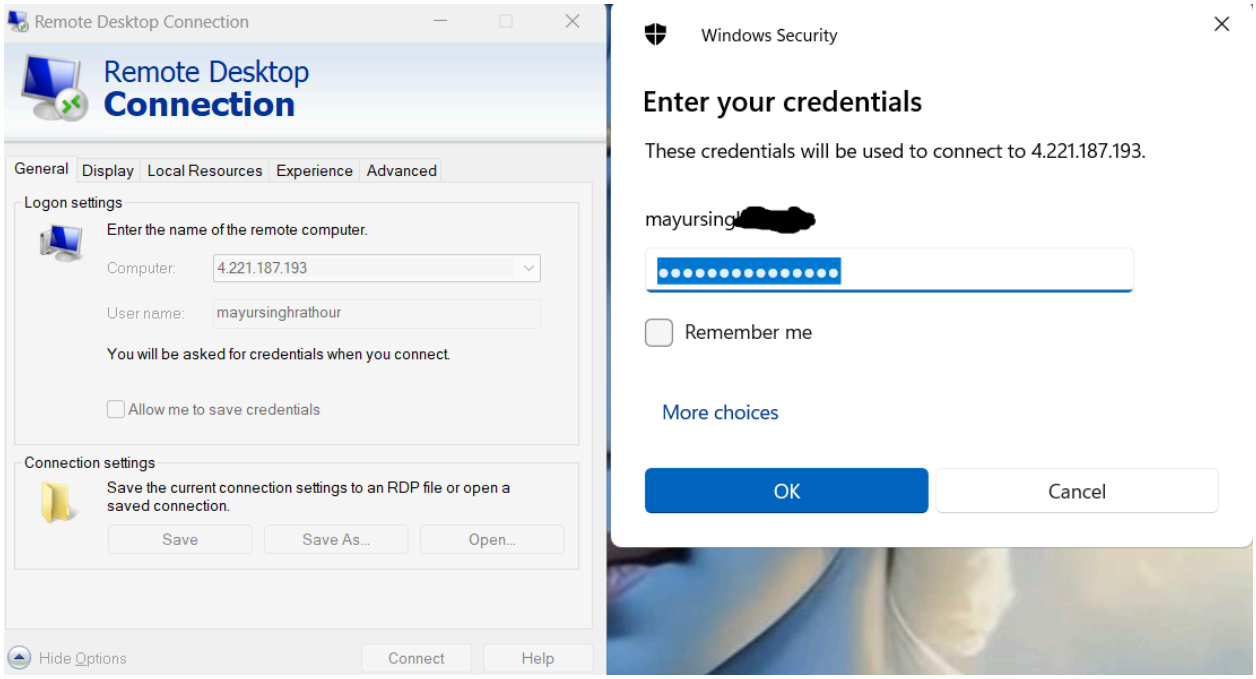


As evident from the presence of a public IP address assigned to our VM, we can confidently affirm the successful creation of our virtual machine. Furthermore, this connectivity enables seamless integration with both log analytical and Sentinel services, thereby enhancing our operational capabilities.

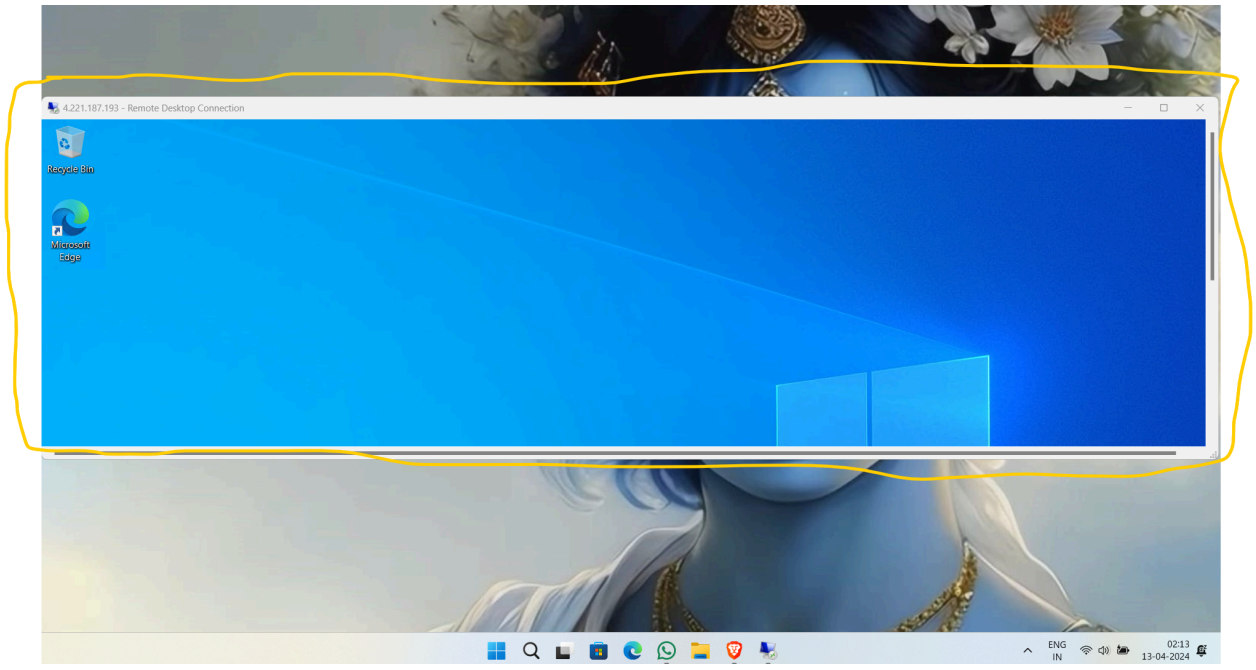
- Now get got the public ip address for our VM we can connect it remotely with our personal system



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

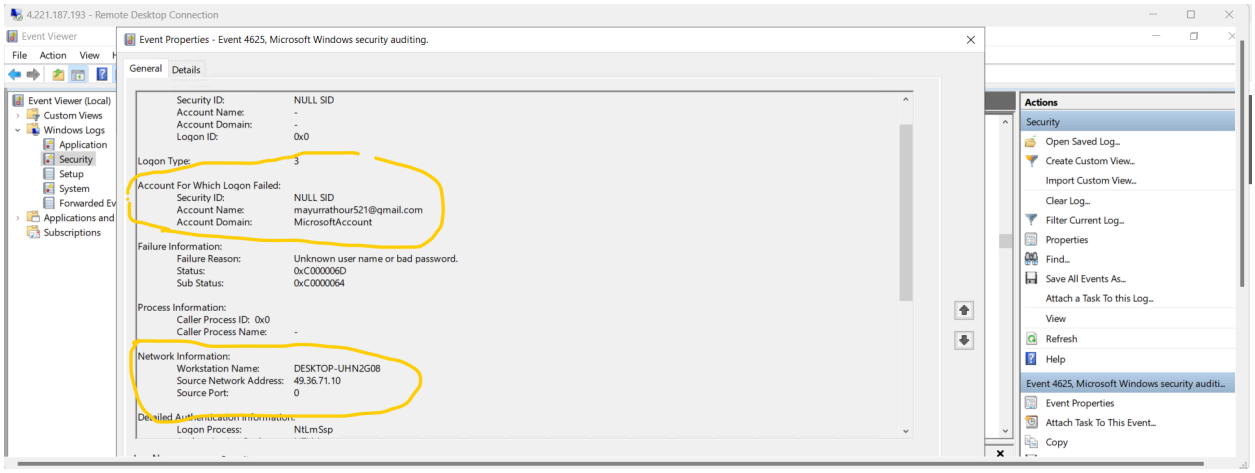


As we can see we have successfully connected windows 10 (VM) to our local system

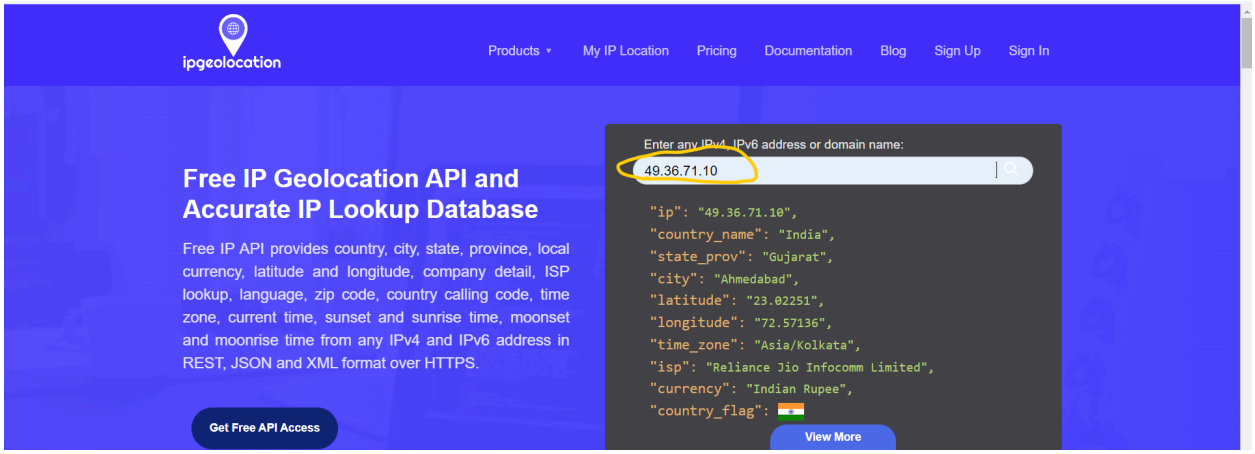


Upon inspection of the event viewer within our VM system, we have observed a failed attempt to access the VM, revealing the IP address of the originating system involved in the unauthorized access endeavor.

- 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.



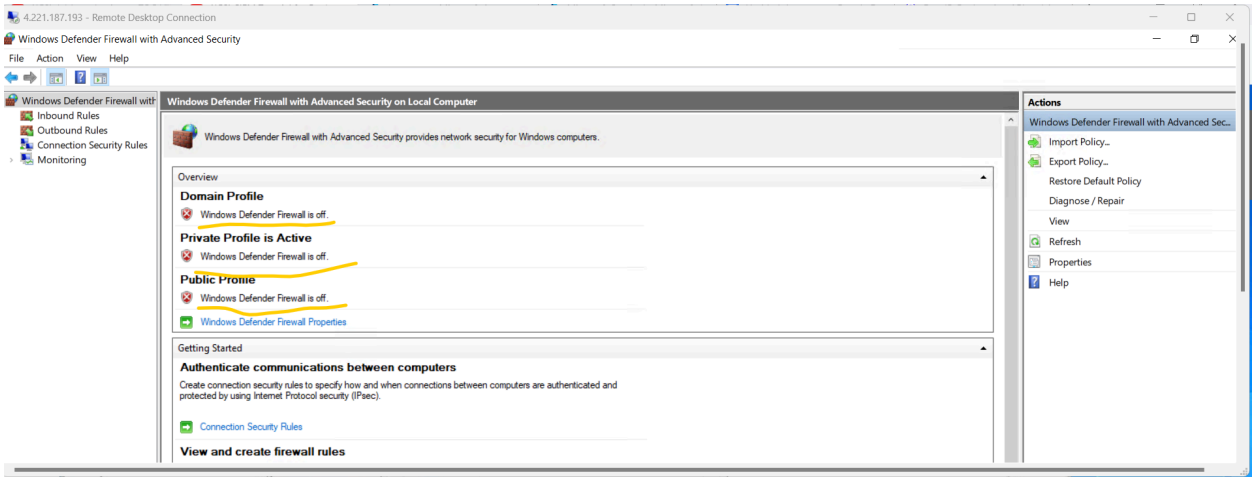
Our single attacker location (find statically)

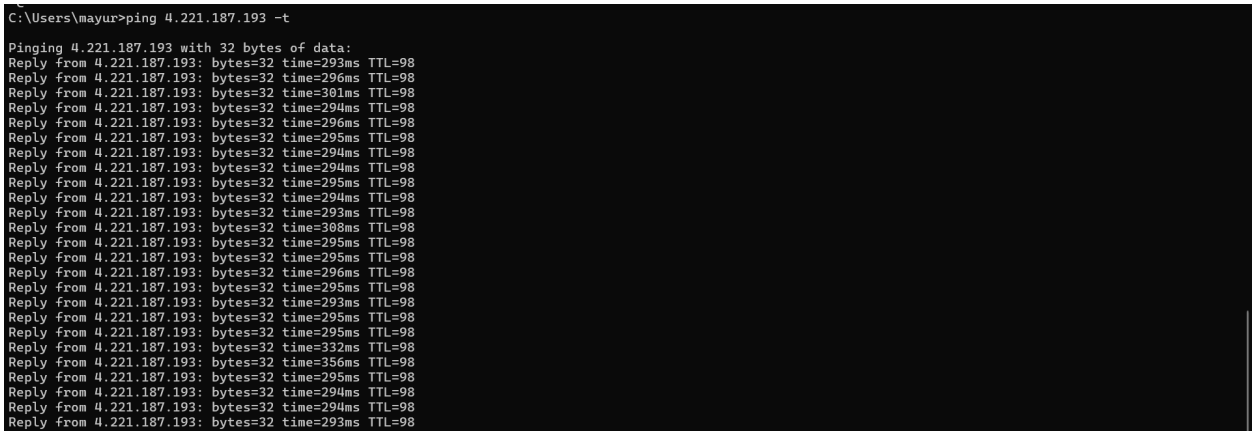


Now we are going to turn off our VM firewall so anyone can send request to it

- As we can see firewall is on our VM system

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

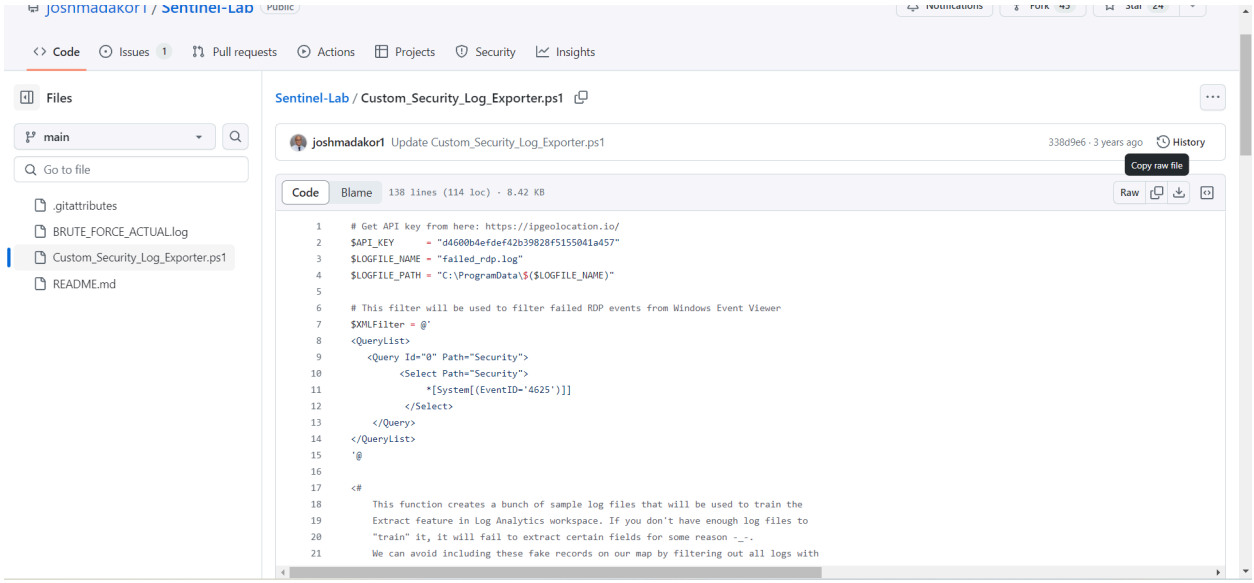




Now we are going to take help of a github repo, which will automatically take the failed audit event from our **Event viewer** we will send those event to to our 3rd party website and that website is going to send us information about that ip address for example Location,ISP,Logituted and etc

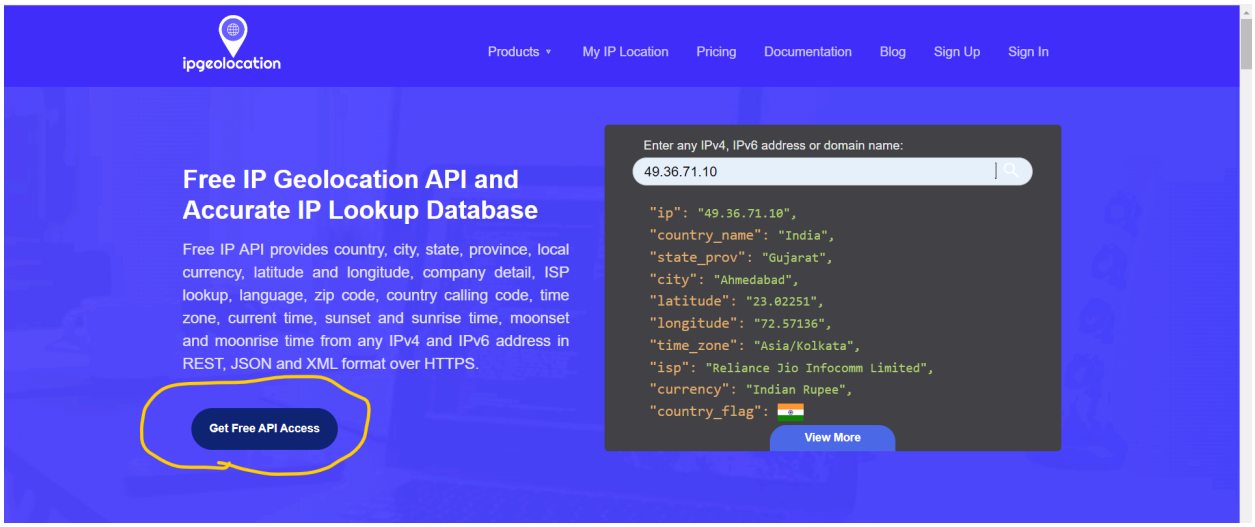
Github repo link -
https://github.com/joshmadakor1/Sentinel-Lab/blob/main/Custom_Security_Log_Exporter.ps1

IP geolocation finder link -
https://ipgeolocation.io



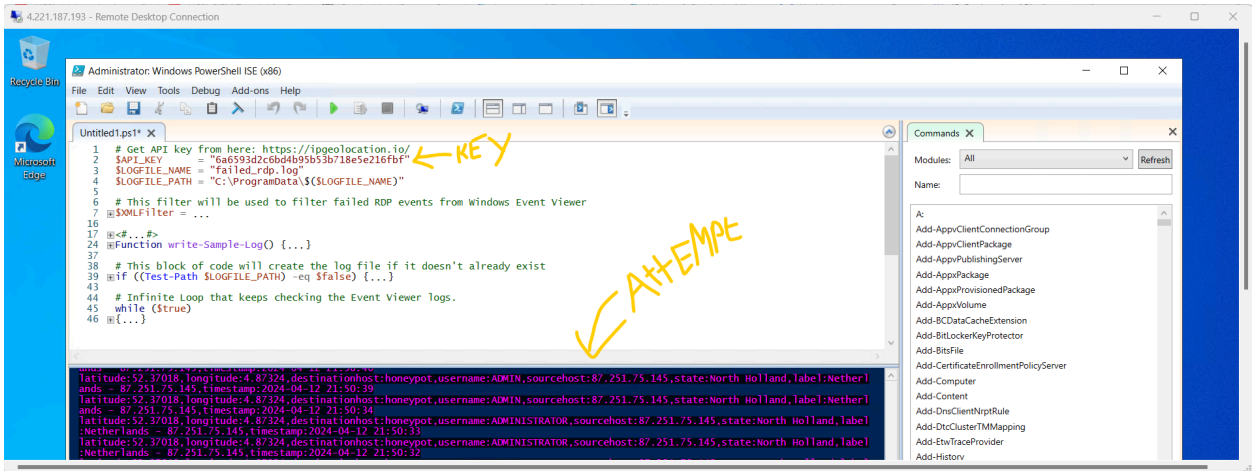
We will be needed and API key of this geolocation website to make our github code work properly

- We can find API from here



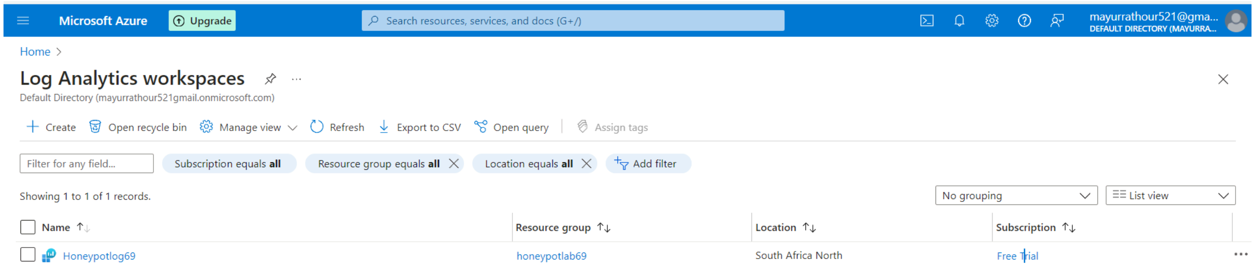
We have to run our git code in VM powershell ISE

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



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

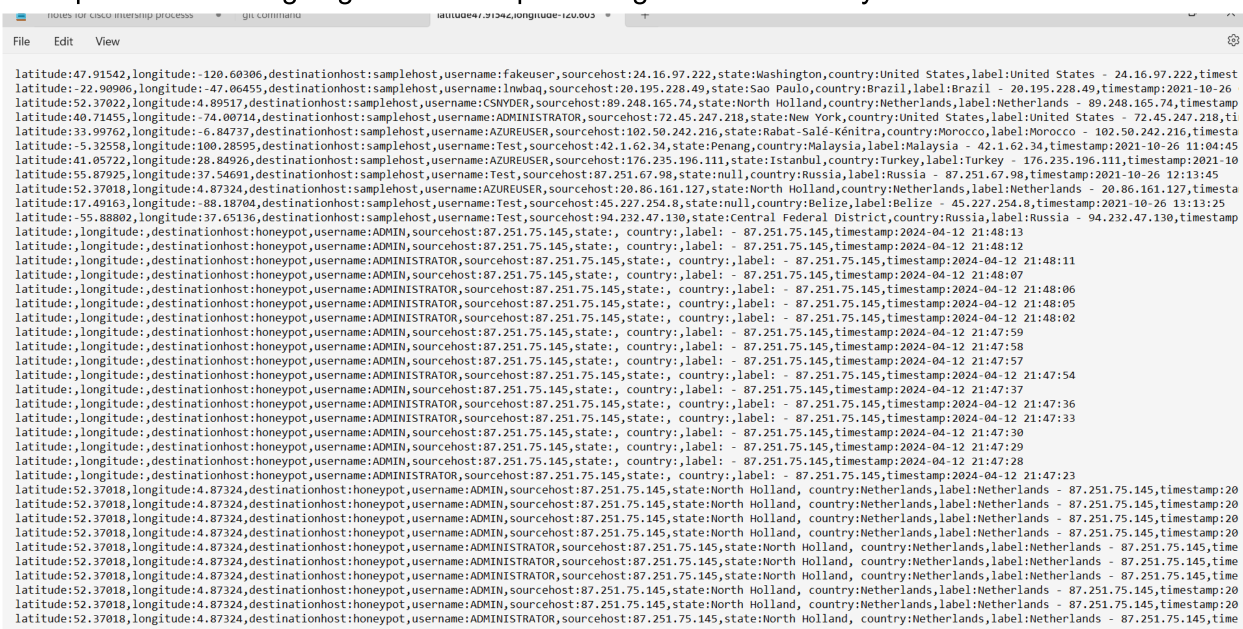
- As we can see our honeypot group inside log analytics which we created in starting



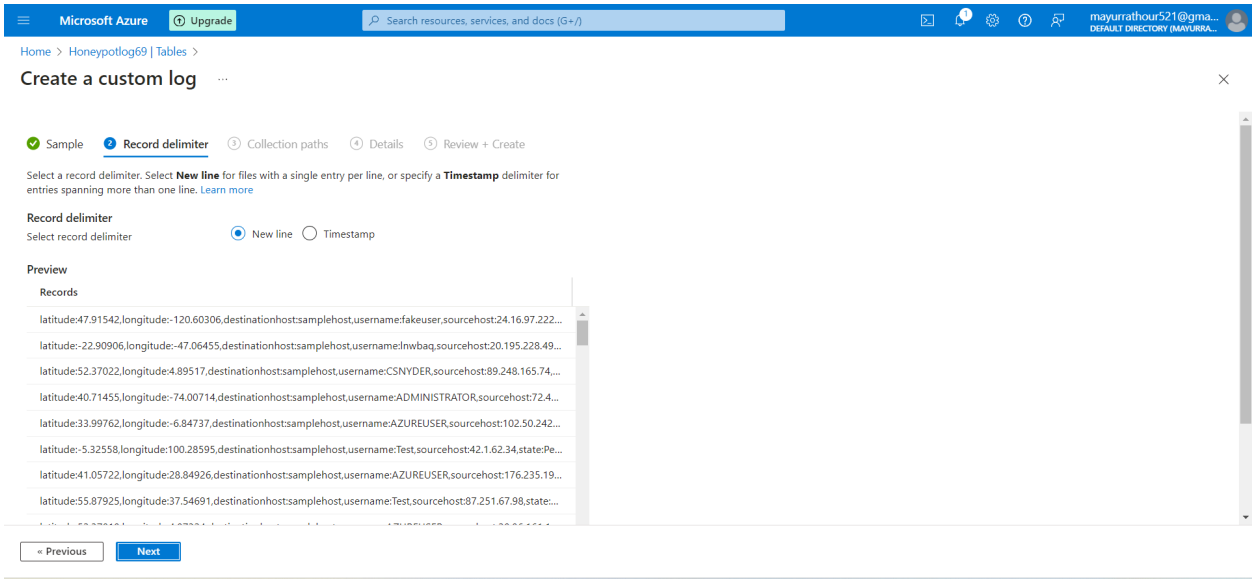
-

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

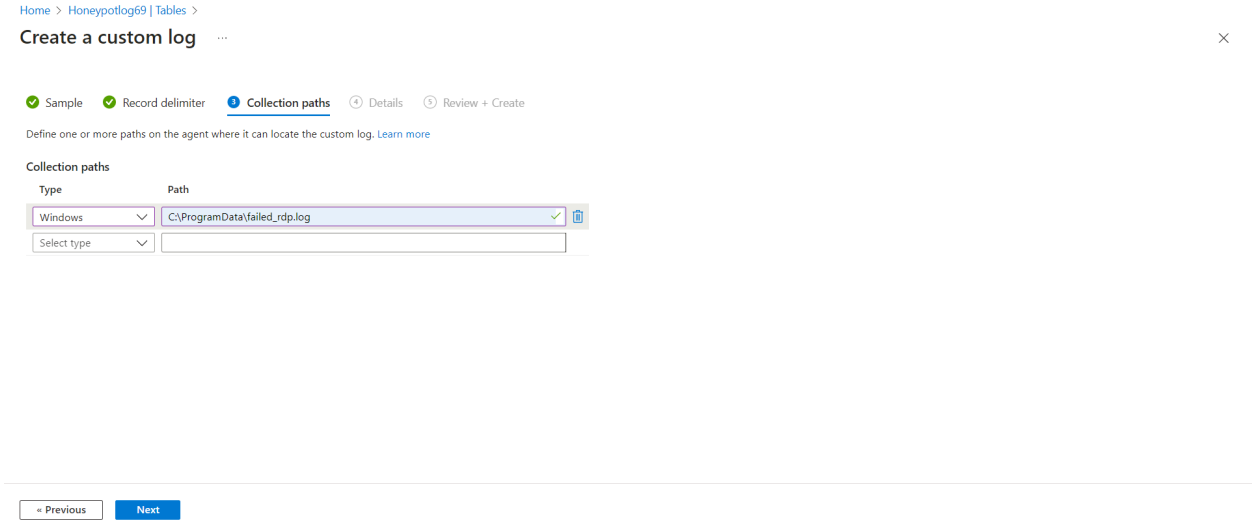


We have successfully loaded our log file to our cloud workspace

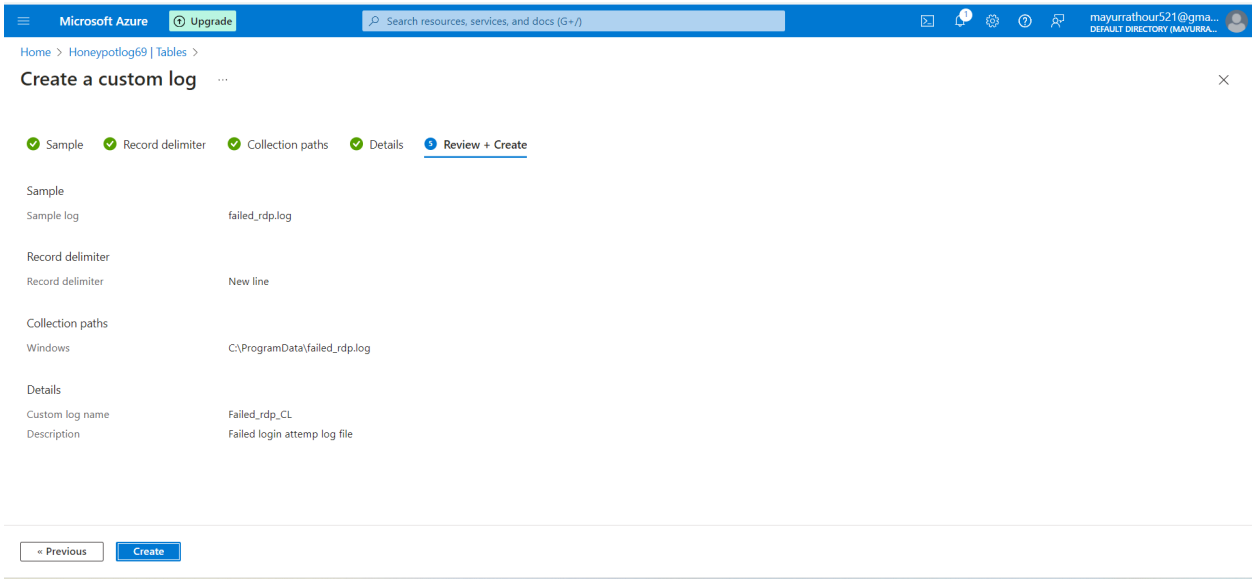


Now we have to connect our VM log file to our analytical workspace

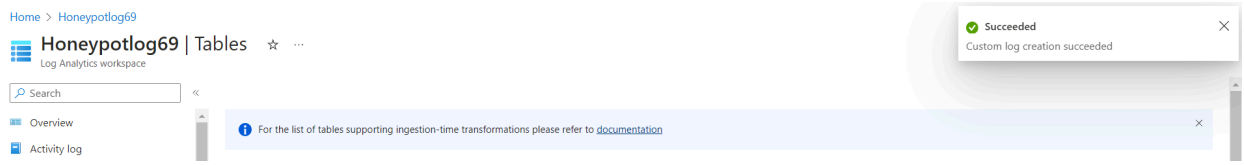
- We have to put exact location of our log file which we had created in our VM(windows)



Finally creating our custom log file

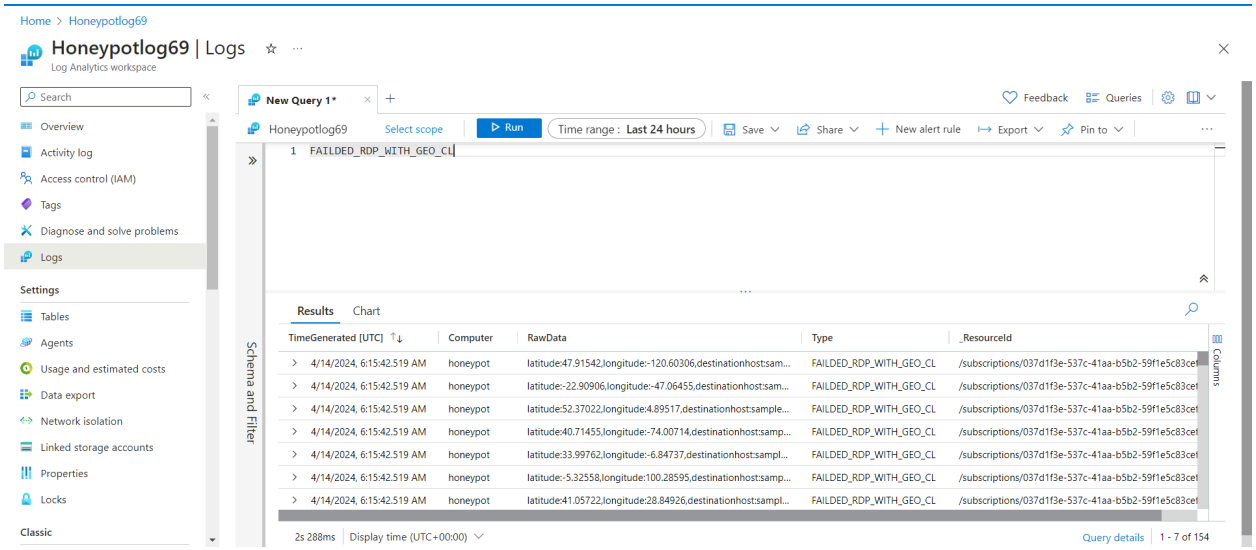


- Created successfully



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

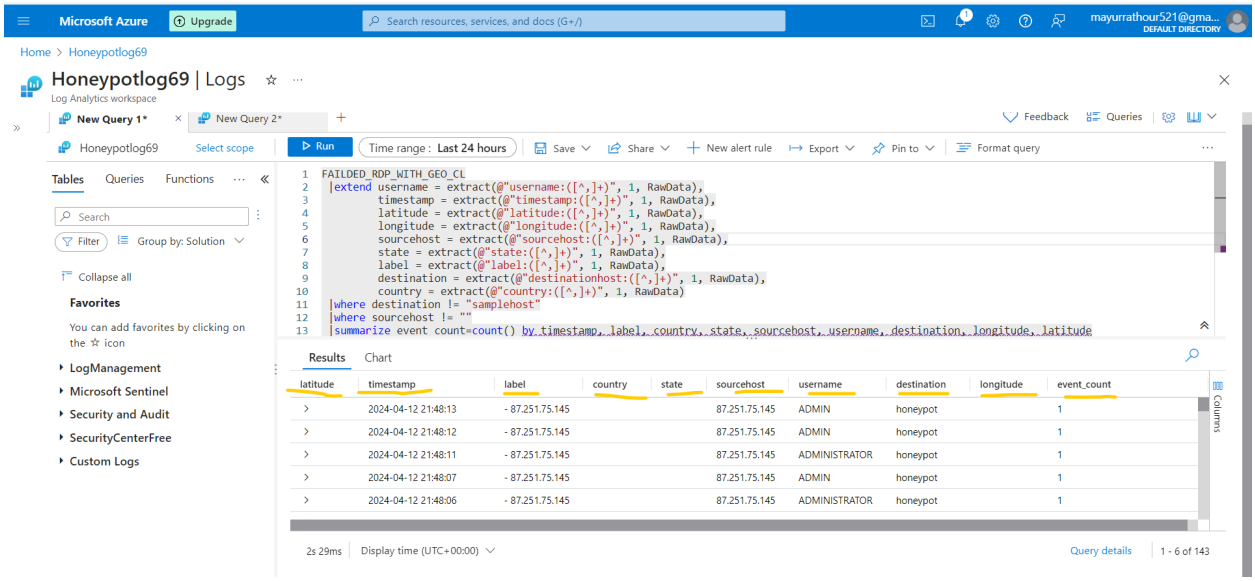
- It will take 30 min to load for the first time



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

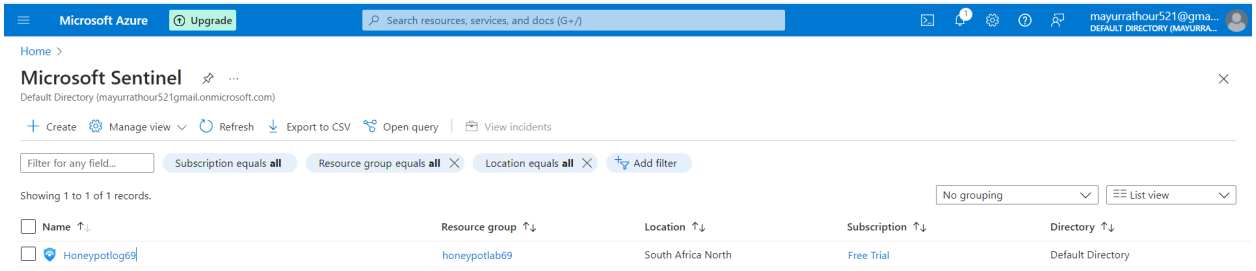
- With help of this query we are going to separate the data from one column to multiple column
- **To do so we are going to type KQL logic for custom fields**

```
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
```

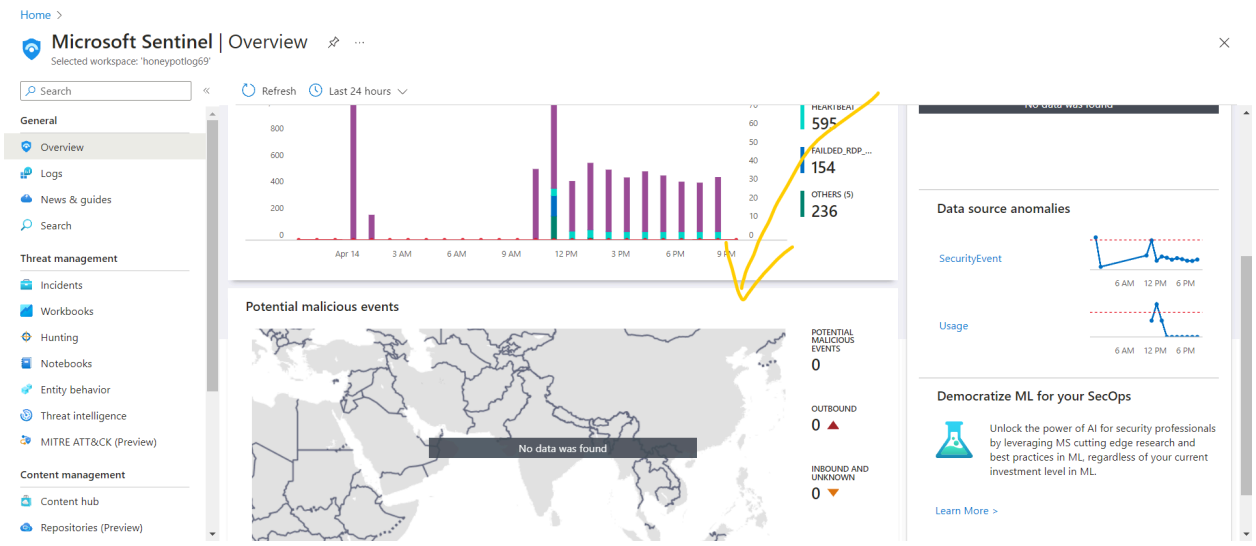


-

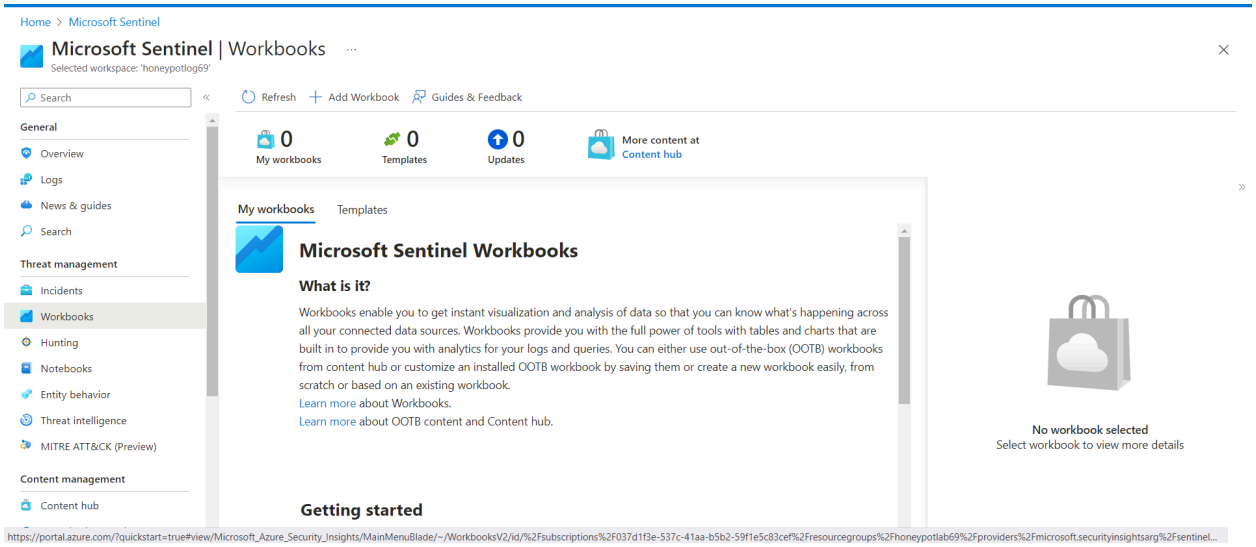
Now we going to Microsoft Sentinel for our final steps



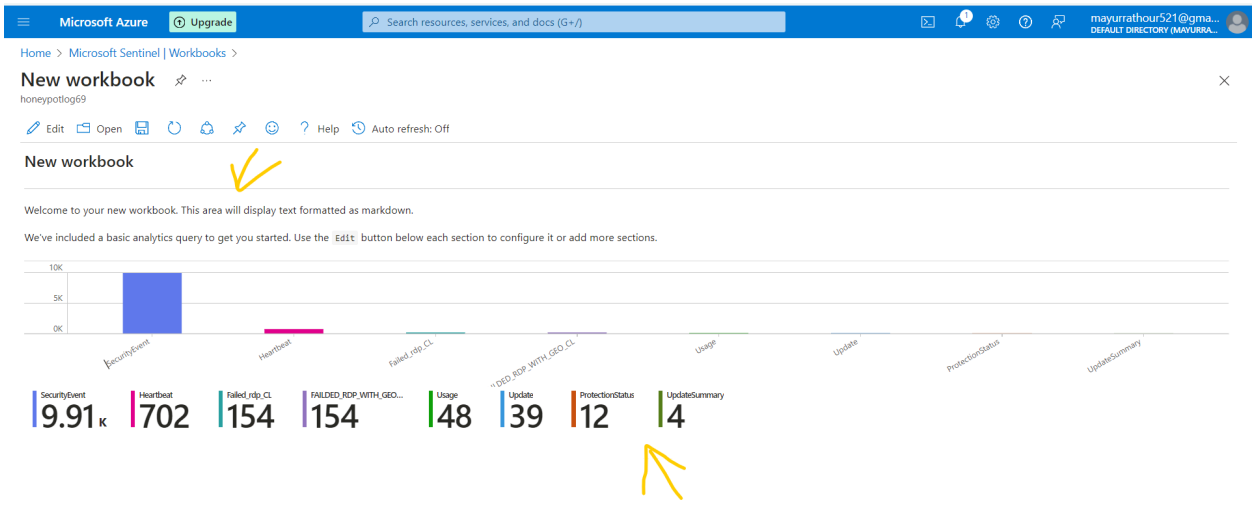
As we can see as for now we have no data found for any kind of attack



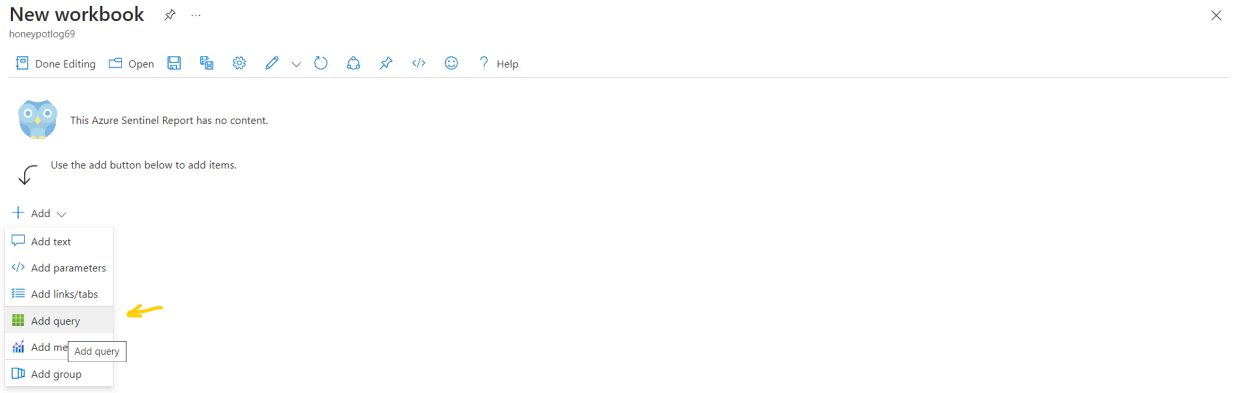
Now we will go to workshop



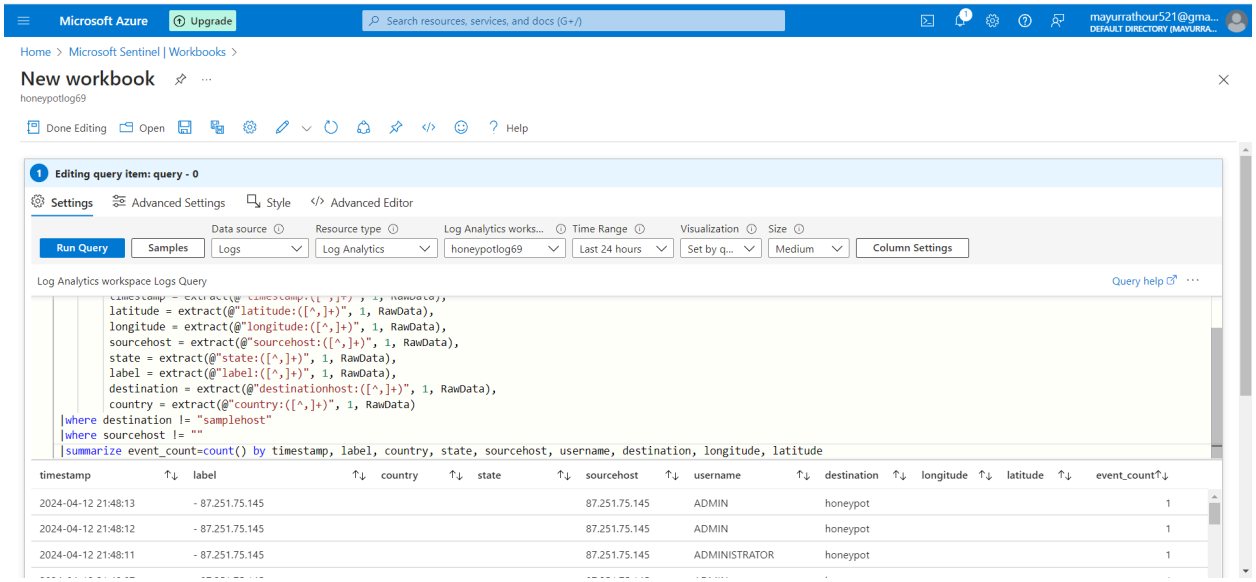
We are not going to need default GUI, so we are going to remove it from our workshop



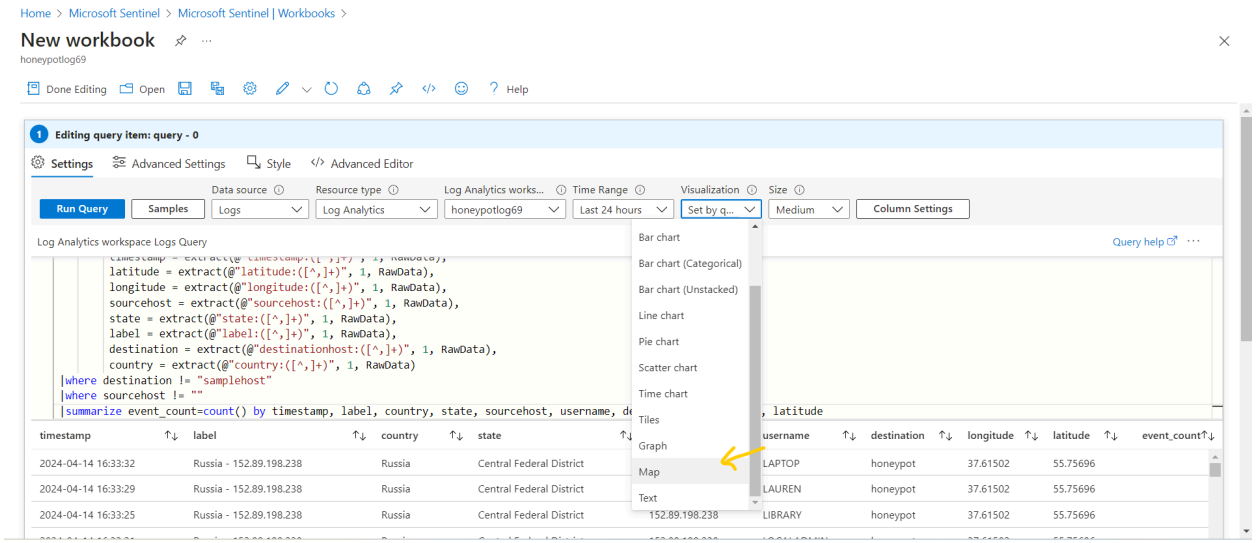
Now we are going to add our query which we written above to extract common filed



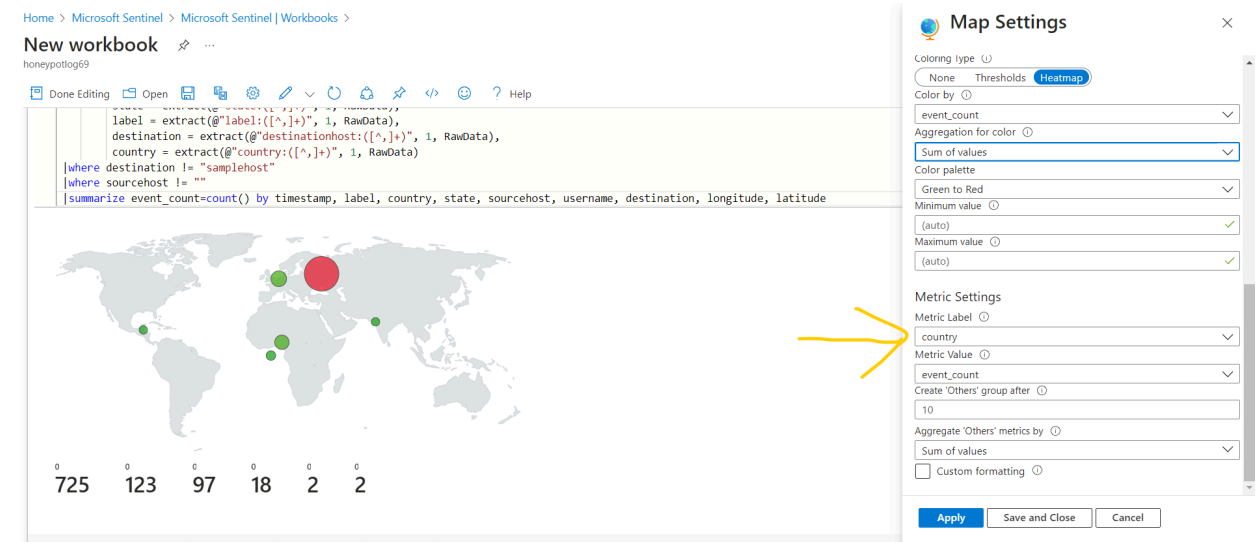
As we can see we are getting same output for our query in our workbook



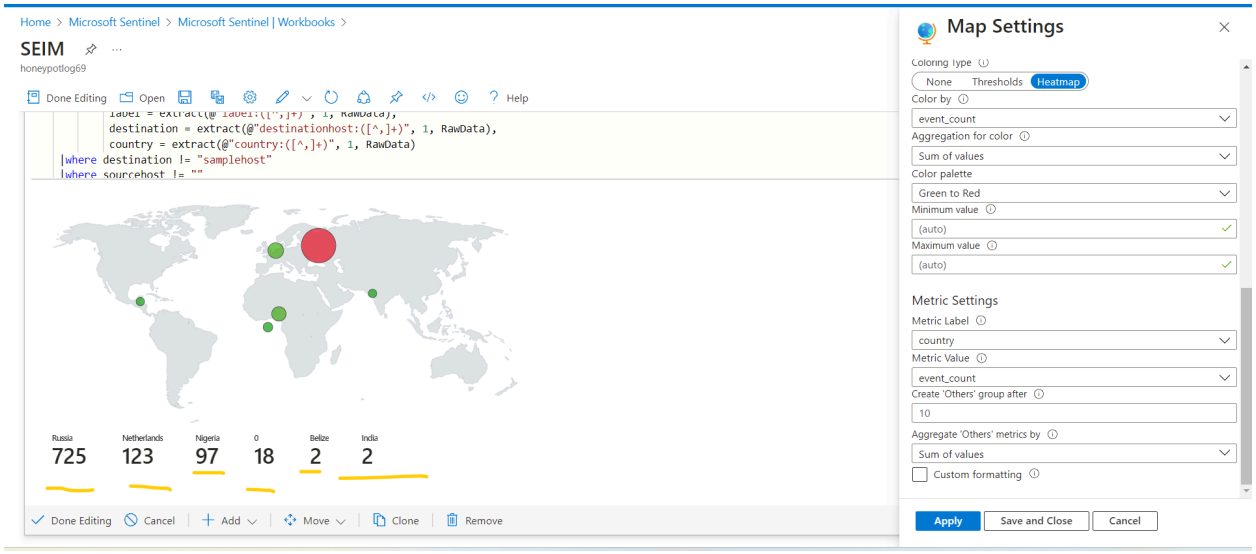
Now we select map to visualization



We set label as country to read data properly



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



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