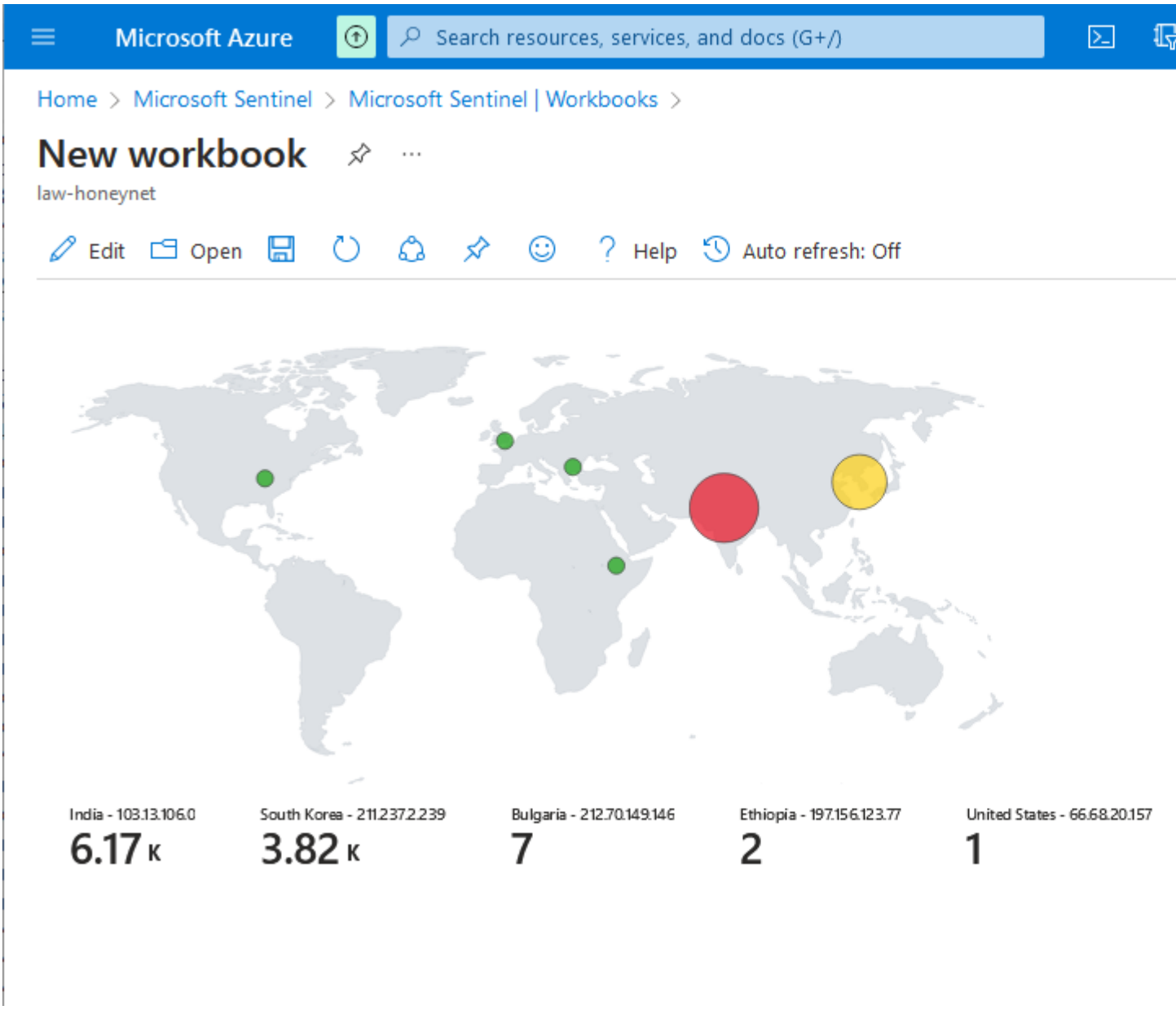
Esteban E. Valverde

Create and monitor a honeypot in Microsoft Azure

Project description

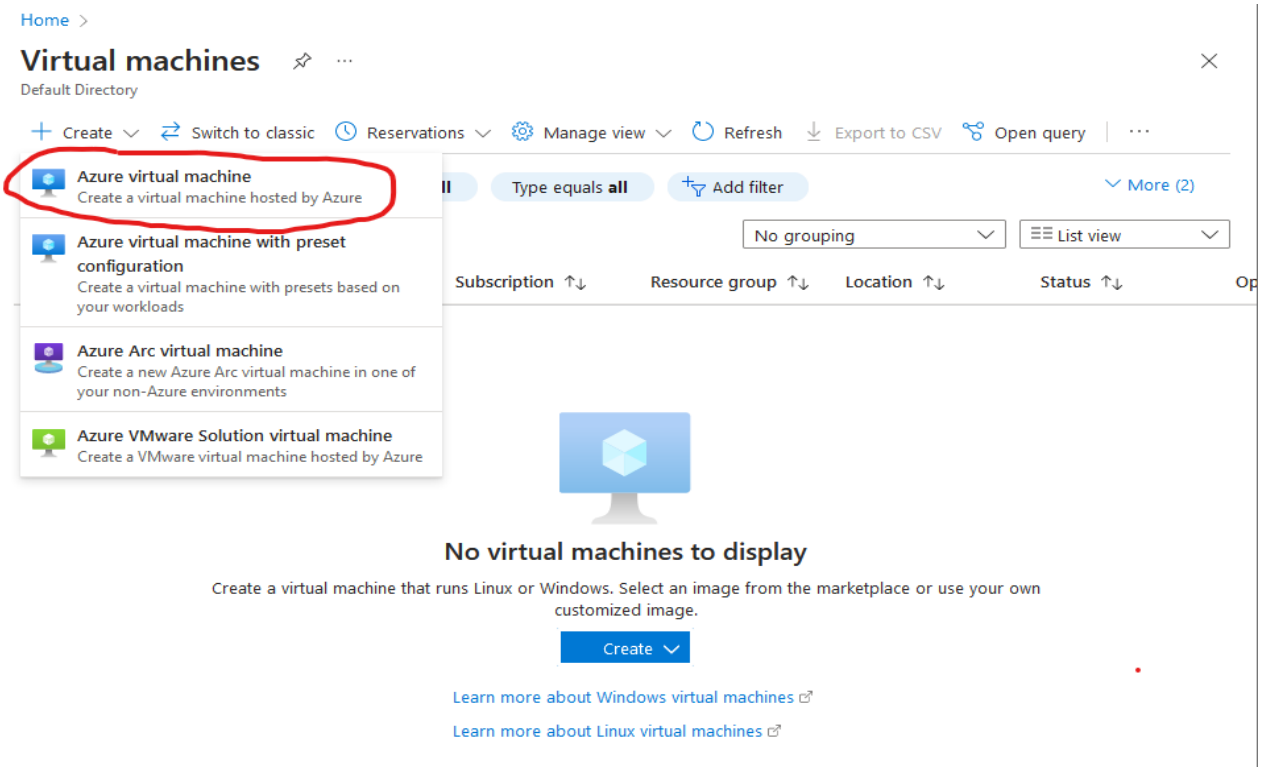
The security team at my organization decided that we need to take proactive steps to understand attack techniques and detect threats as early as possible, especially since a new internet-facing host has just been deployed into the network and the team recently reported unusual, failed RDP (Remote Desktop Protocol) login attempts. To aid in this, I plan to create a honeypot machine with similar characteristics to the host, and open port 3389 (RDP), to lure attackers in and learn more about their hacking tactics, techniques, and procedures. By the end, we can expect a Microsoft Sentinel map illustration, showcasing the geolocation of potential attackers' IP addresses and the number of unsuccessful attempts to our honeypot. It should look something like this:



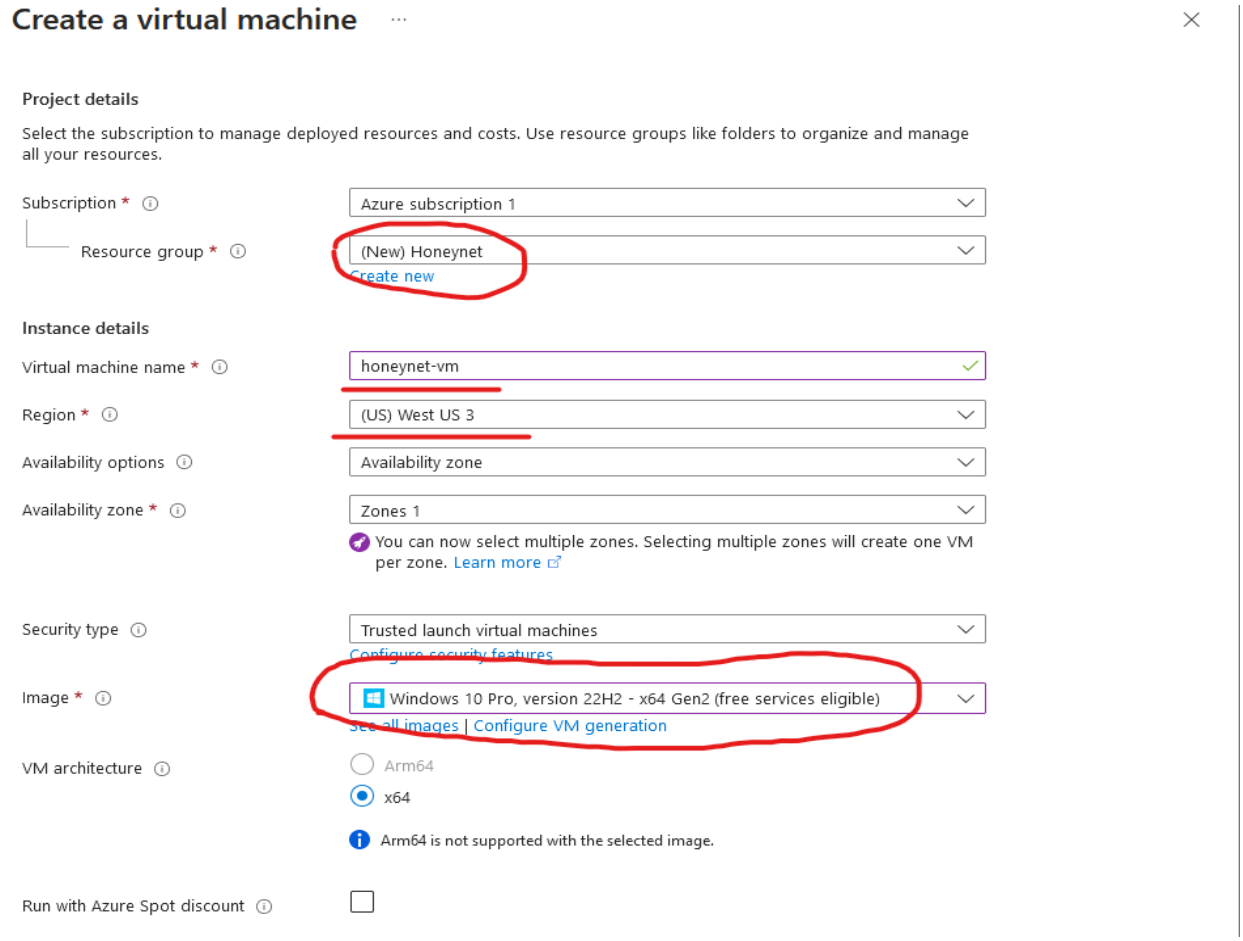
Step 1: Build Azure Virtual Machine

The first step in creating our honeypot is to navigate to portal.azure.com and register for a free account, Microsoft Azure gives us a free $200 credit to try their services so there shouldn't be any cost in case you wish to follow along.
After setting up our account, on our Azure dashboard, we will search for "**Virtual Machines**" in the search bar and click the first option. Then, we will click the "**Create**" button as shown below to start setting up our machine:

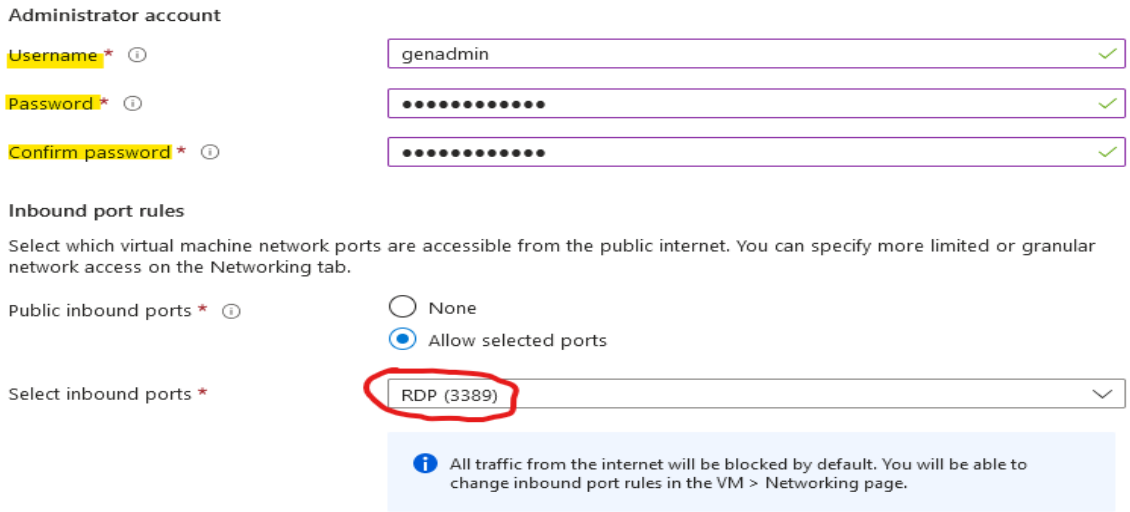
Esteban E. Valverde



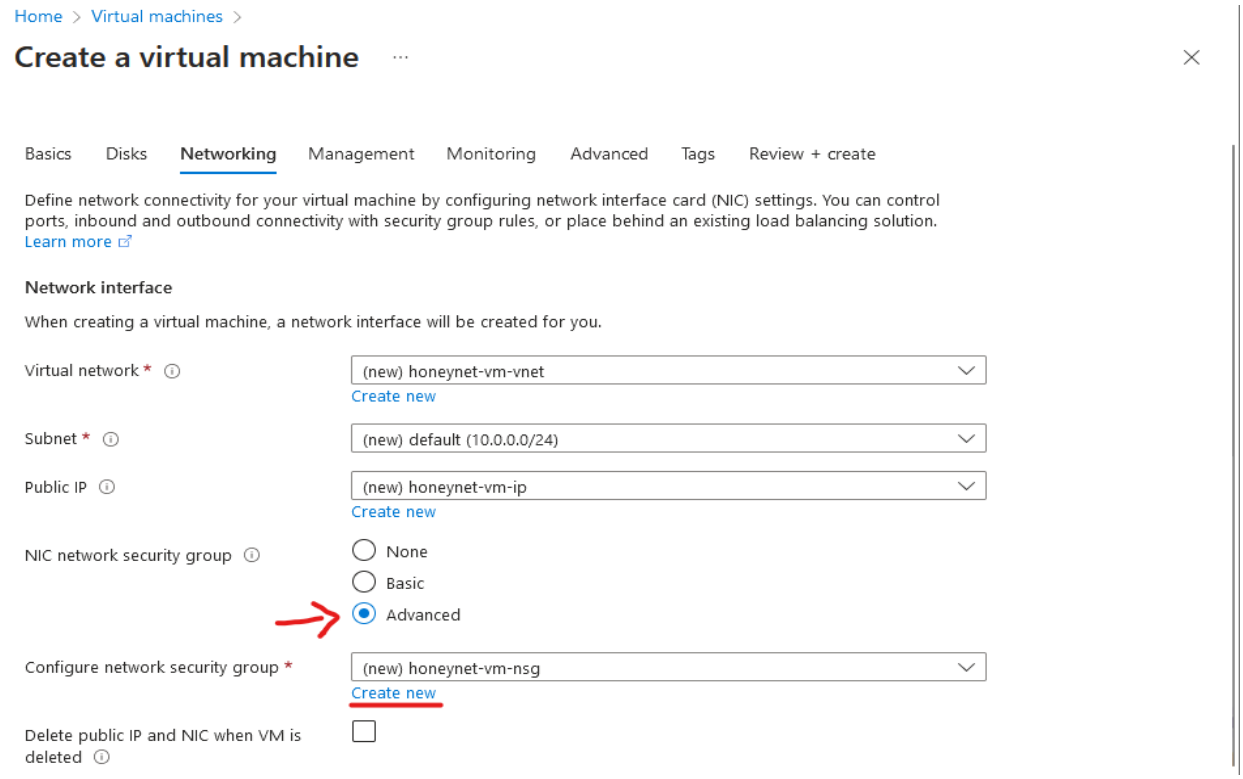
Next, we must specify the VM's basic details, such as creating a new group to assign the VM to, name the VM, choose your region, and for this project I am assigning Windows 10 Pro as the OS for the honeypot. This is how the configuration should look like:



Following , we will move to the next screen, where we will assign a username and password for the root account and assign the inbound port to "**RDP (3389)**" since we will be monitoring for failed RDP attempts.



Next, since we are opening our computer to the whole internet, we have to create a new Network Interface Controller security group. For this, we'll choose "**Advanced**" under '**NIC network security group'**, and "**Create new**":



Next, we will want to delete any default inbound/outbound rules, and then select "**Add an inbound rule**". We're going to make sure the following settings are applied:

Source: Any

Source Port Ranges: *

Destination: Any

Service: Custom

Destination Port Ranges: *

Protocol: Any

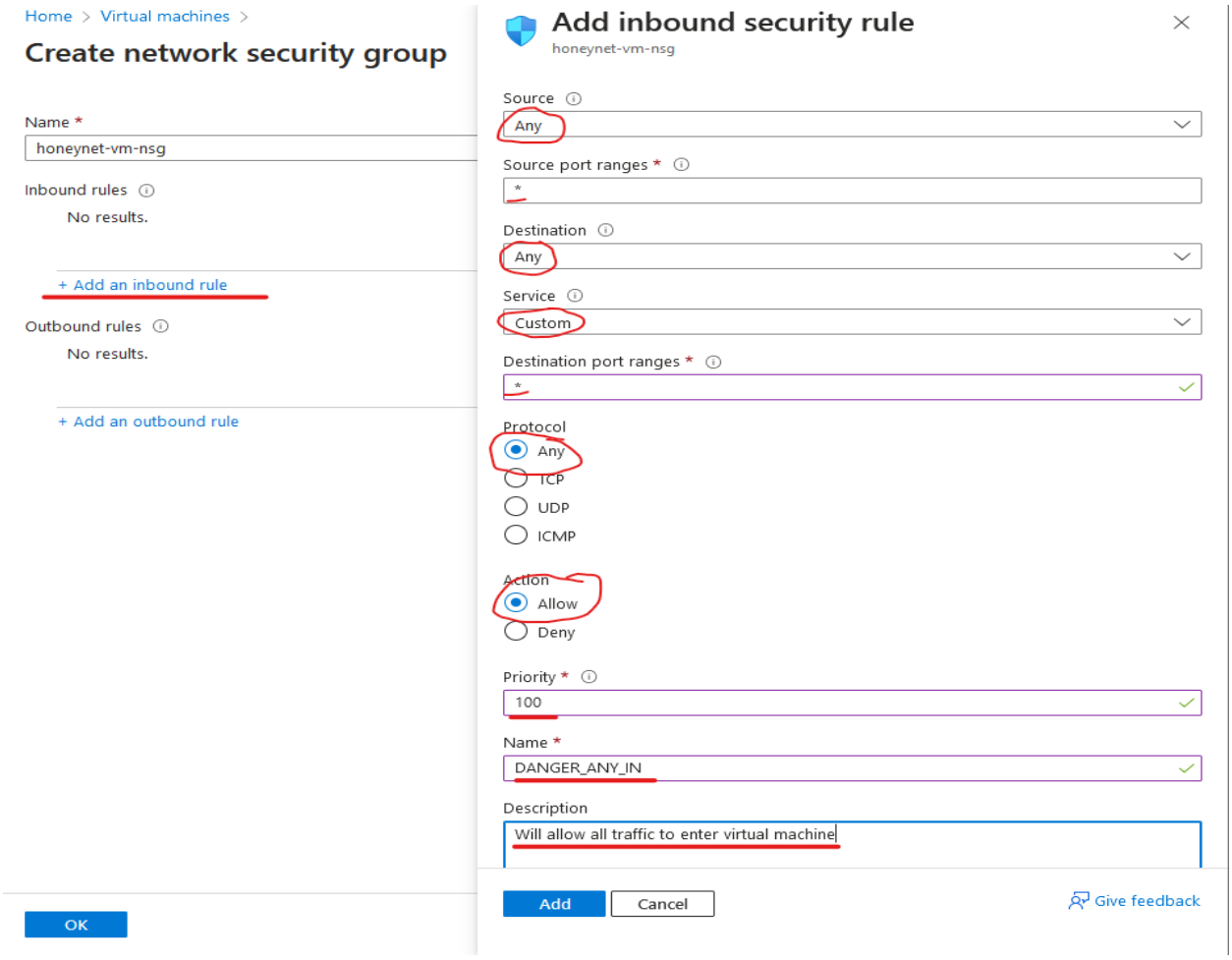
Action: Allow

Priority: 100

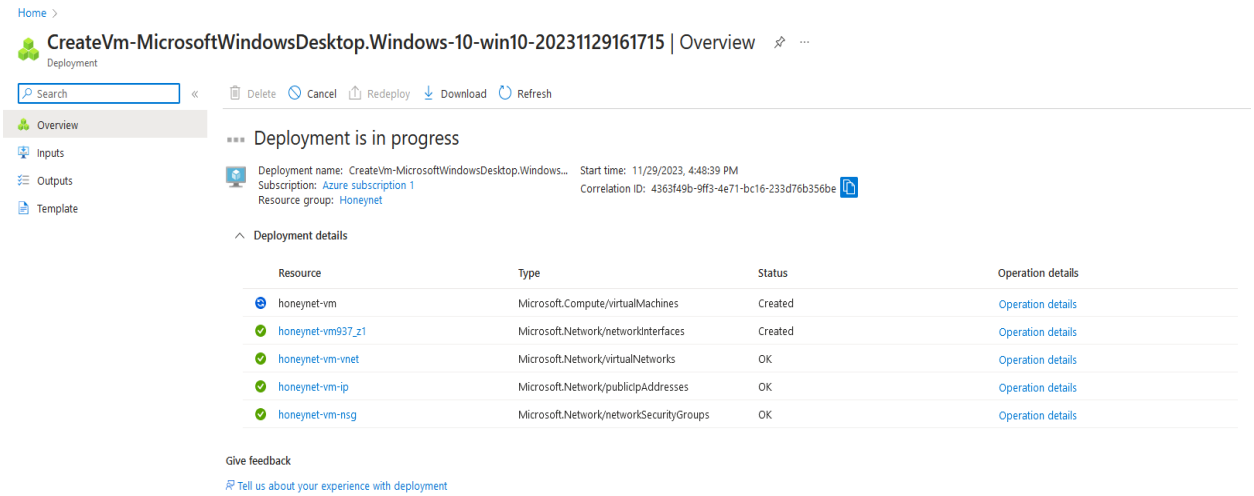
Name: (name your rule).

Description: (add for your reference)

Then, we will press '**Review'**, and '**Create'**.

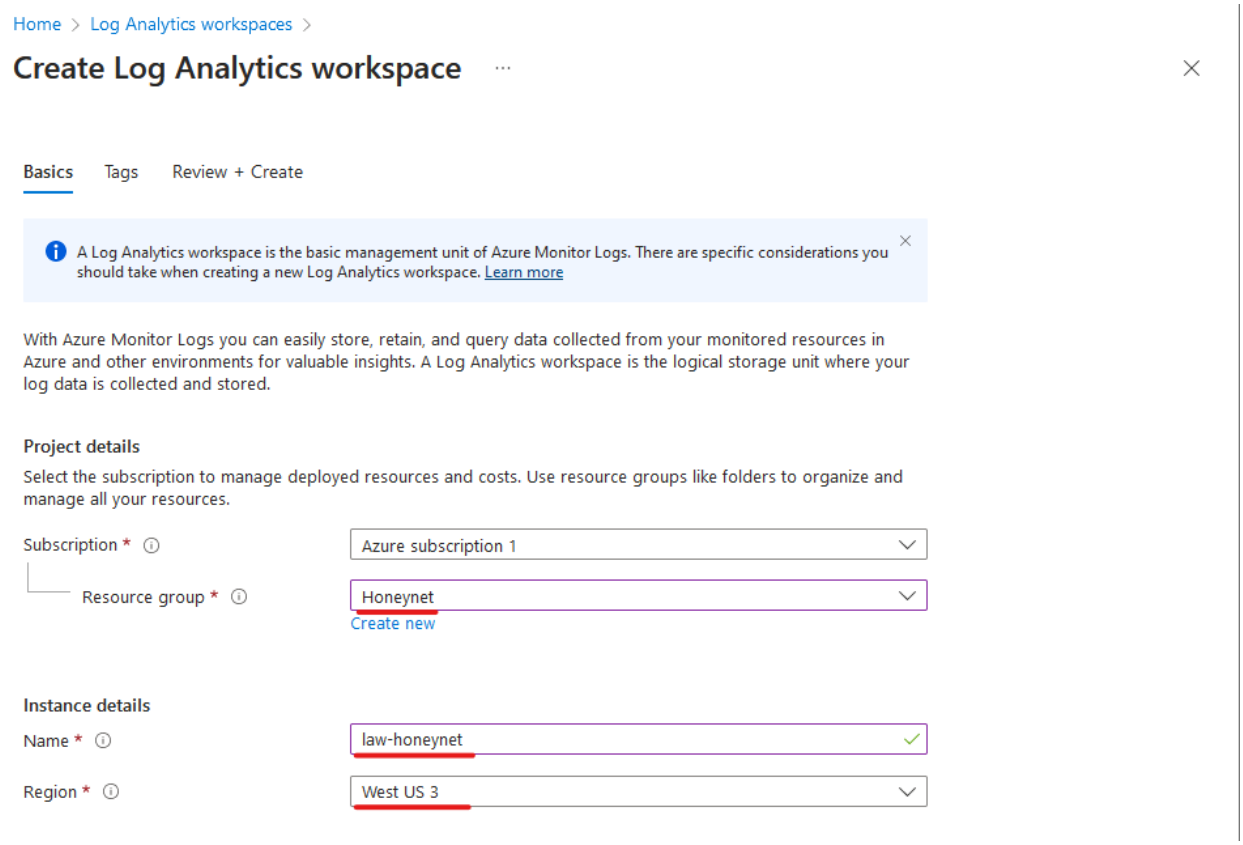


Our virtual machine is now being deployed, this may take a couple of minutes so we will move on to creating our Log Analytics Workspace (LAW) in the meantime.



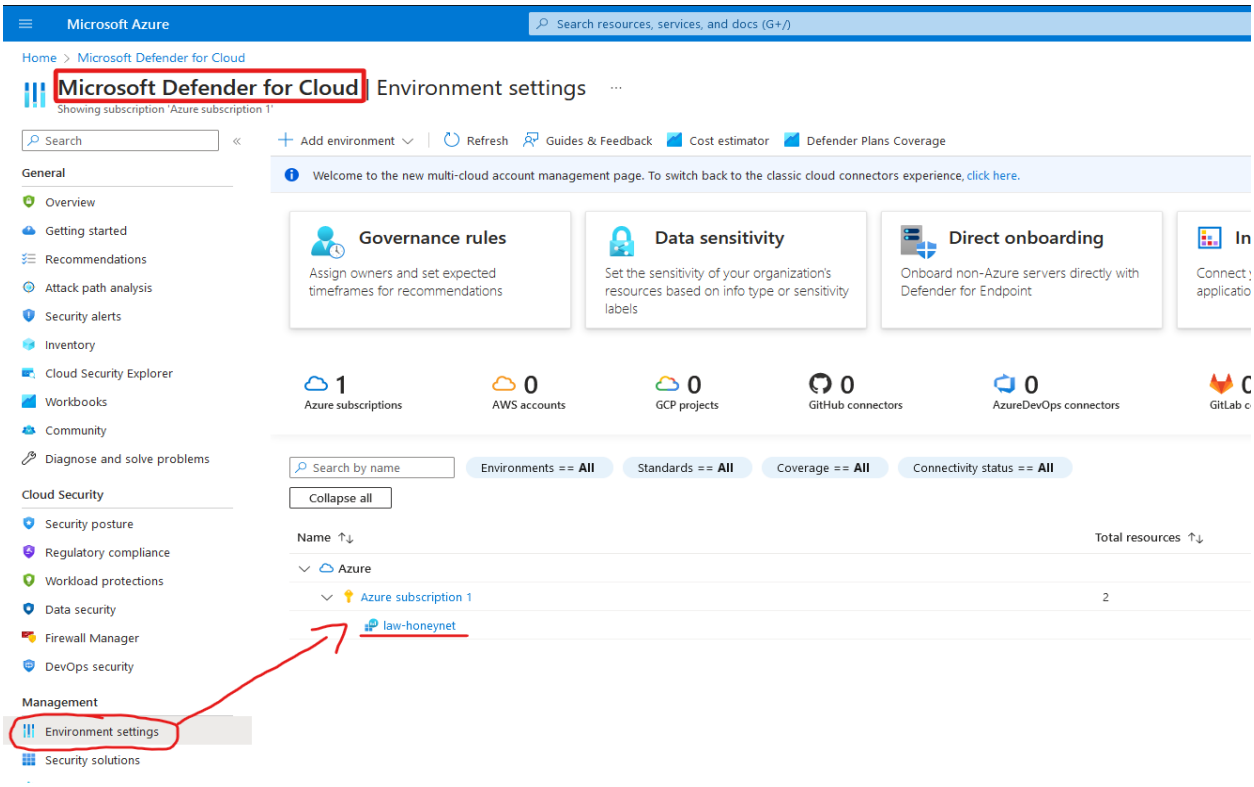
Step 2: Create Log Analytics Workspace

The next step is to create a log analytics workspace so we can ingest and assign the logs to and analyze later with Azure Sentinel. We will search for “**LAW**” on the search bar and click the first result, we will create a new LAW, assign it to the same region and resource group we previously created, and name it. Then we will click ‘**Review+Create**’ and create once it passes validation.

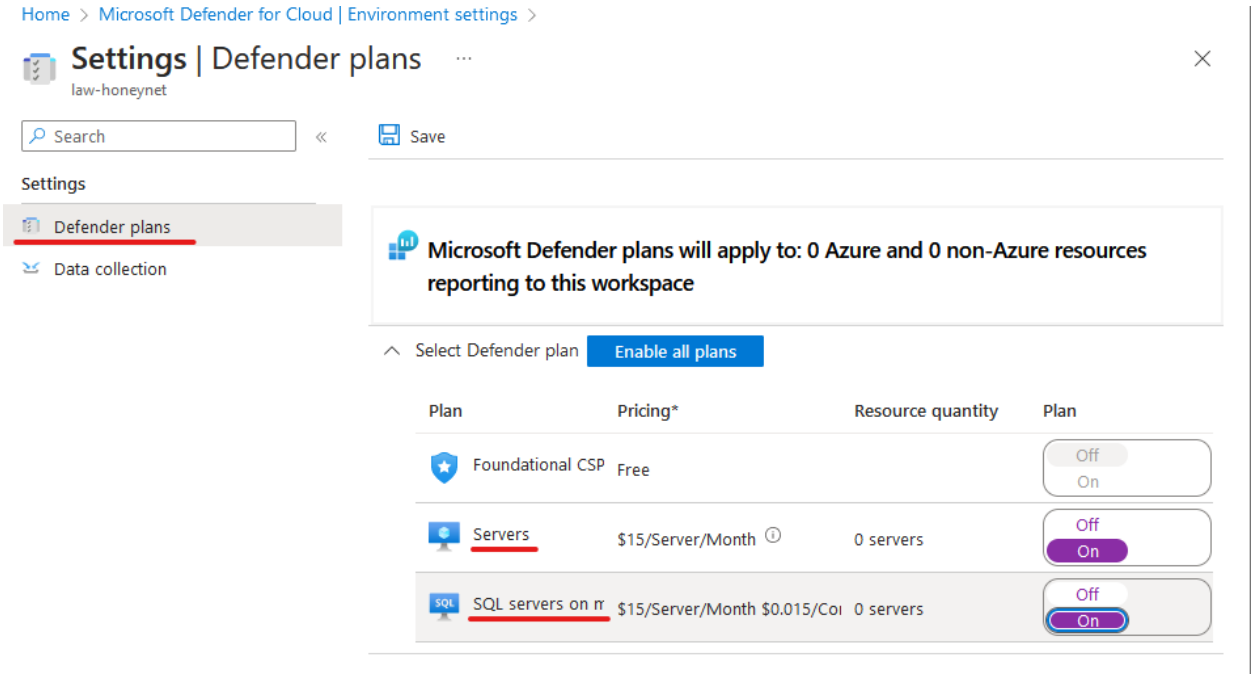


Next, we are going to search for “**Microsoft Defender For Cloud”** on the search bar and click the first result, once there, we will click on "**Environment Settings"** under ‘**Management**’, click the dropdown and select the LAW we previously created:

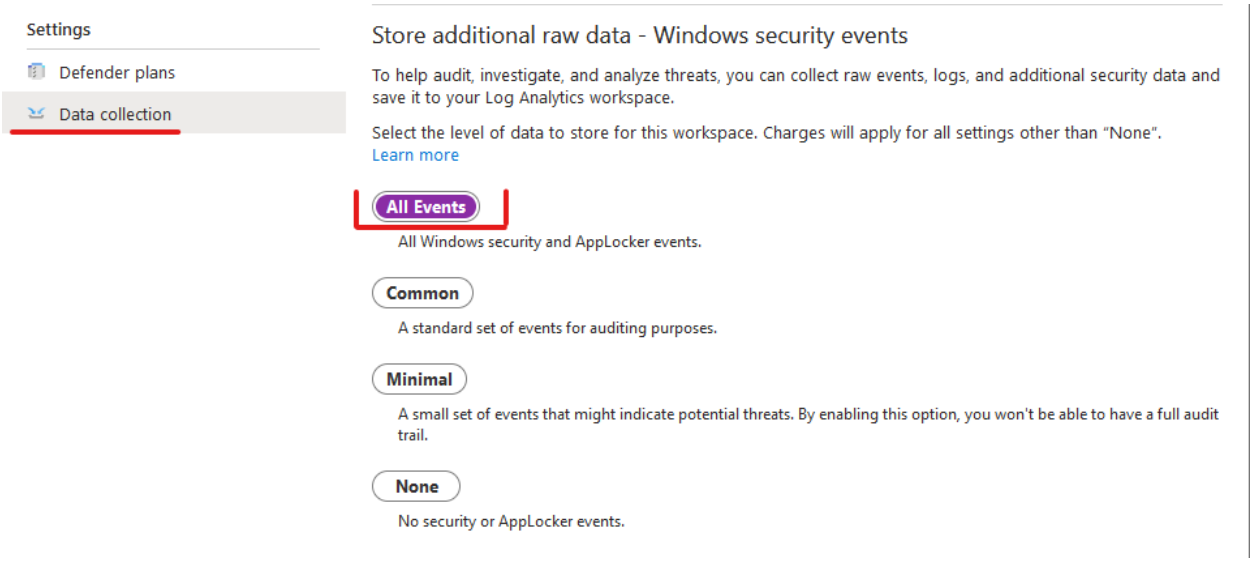
Esteban E. Valverde



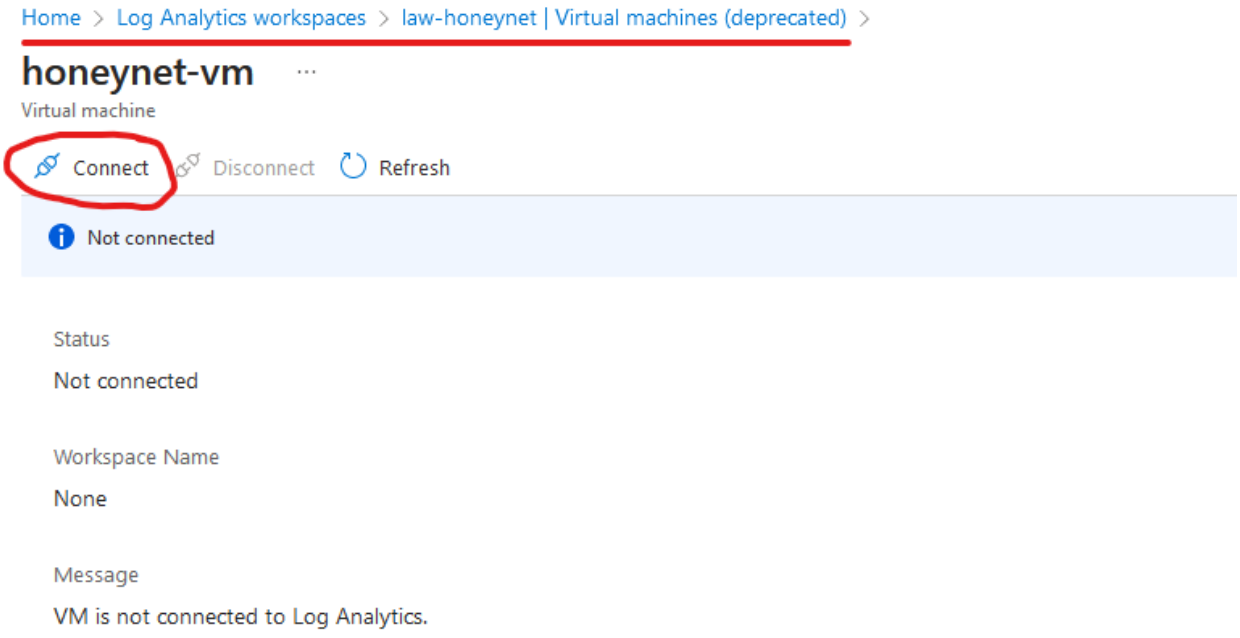
We want to make sure that our Servers and SQL Servers are both on, then we will click "**Data Collection**" on the left menu



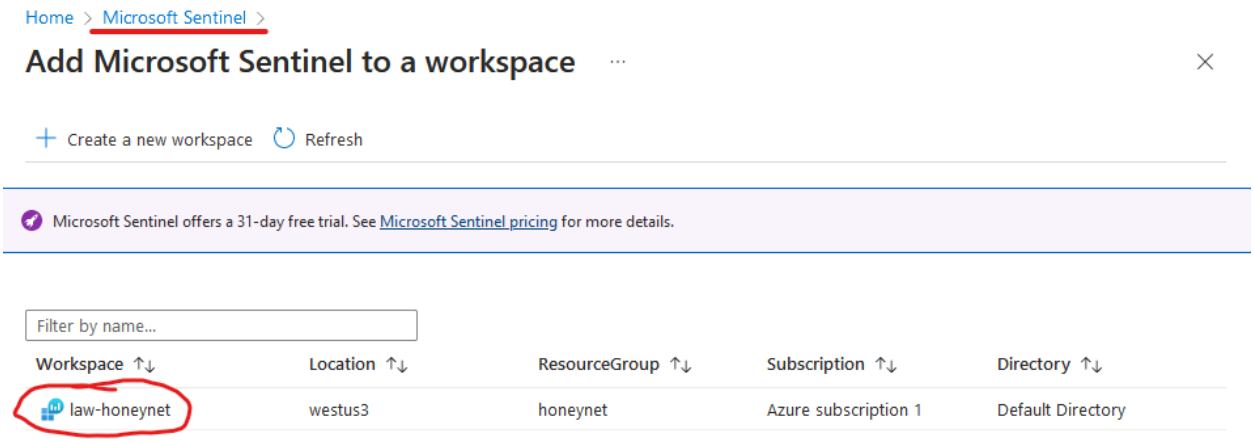
Here, we'll want to click "**All Events**" and hit **SAVE**.



Next, we will navigate back to Log Analytics Workspaces and choose the LAW we created earlier.
We are going to connect it to our virtual machine. This should take a few minutes.

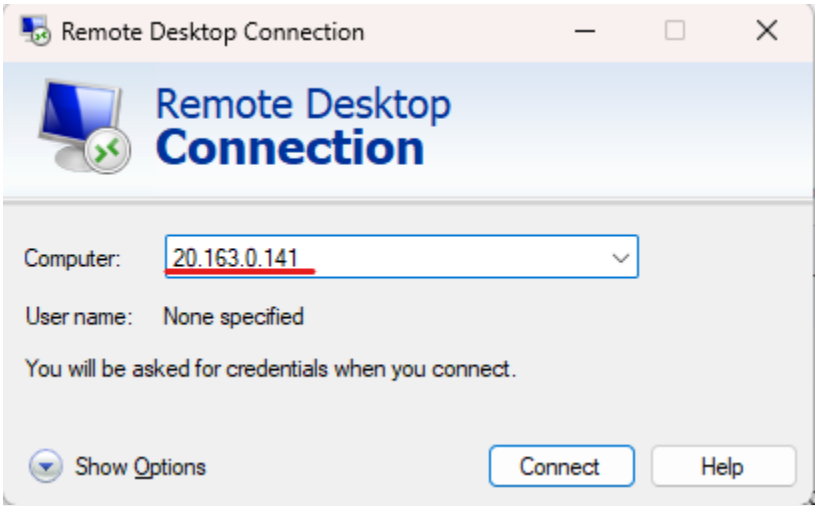


Next, we will search for '**Microsoft Sentinel**' on the search bar, and our newly created LAW should be there, we'll click it, and connect our workspace to Microsoft Sentinel:



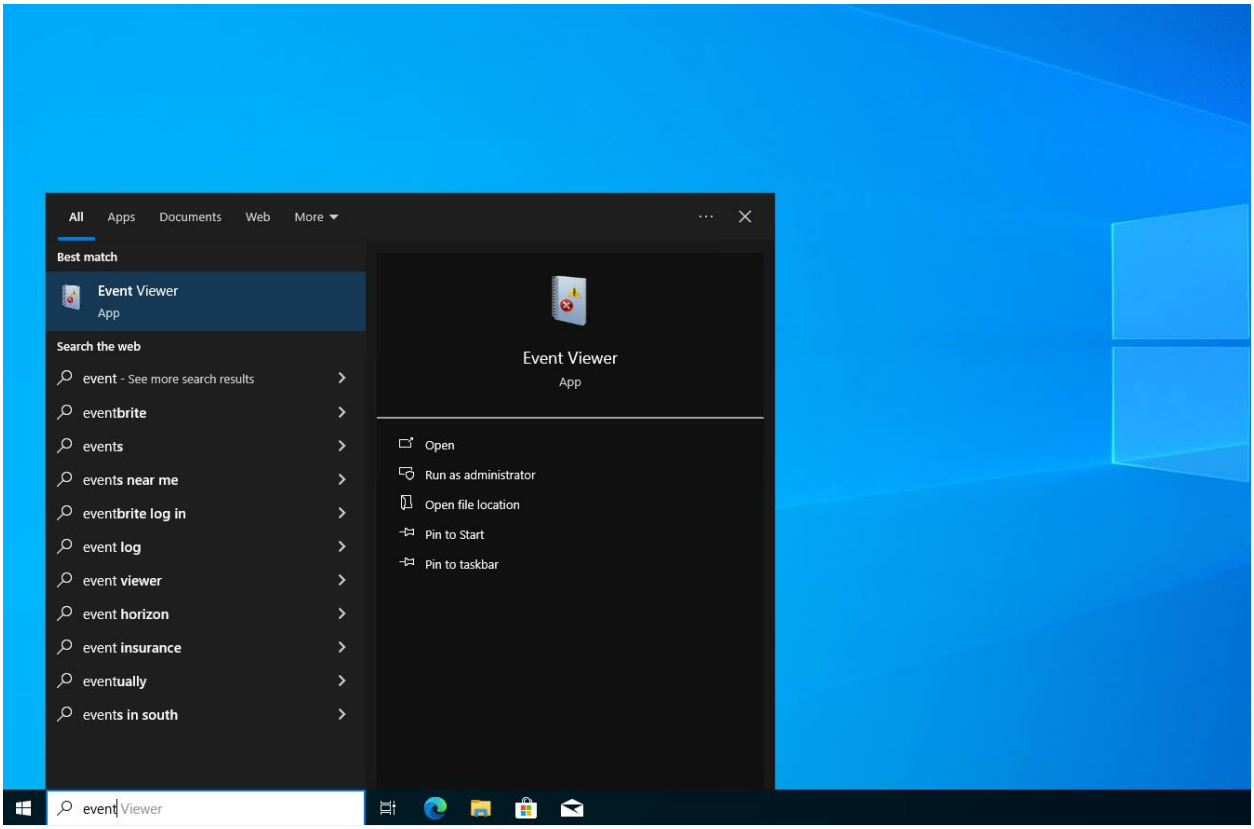
Step 3: Connect and Configure Virtual Machine.

Our next step towards building our honeypot involves connecting it into the actual VM we created to properly configure the endpoint. To do this, first we will find and copy the machine's public IP address in our Azure's '**Virtual Machine**' menu. Then, we will use this address to connect to the VM through RDP, using the admin credentials we previously created to gain access.

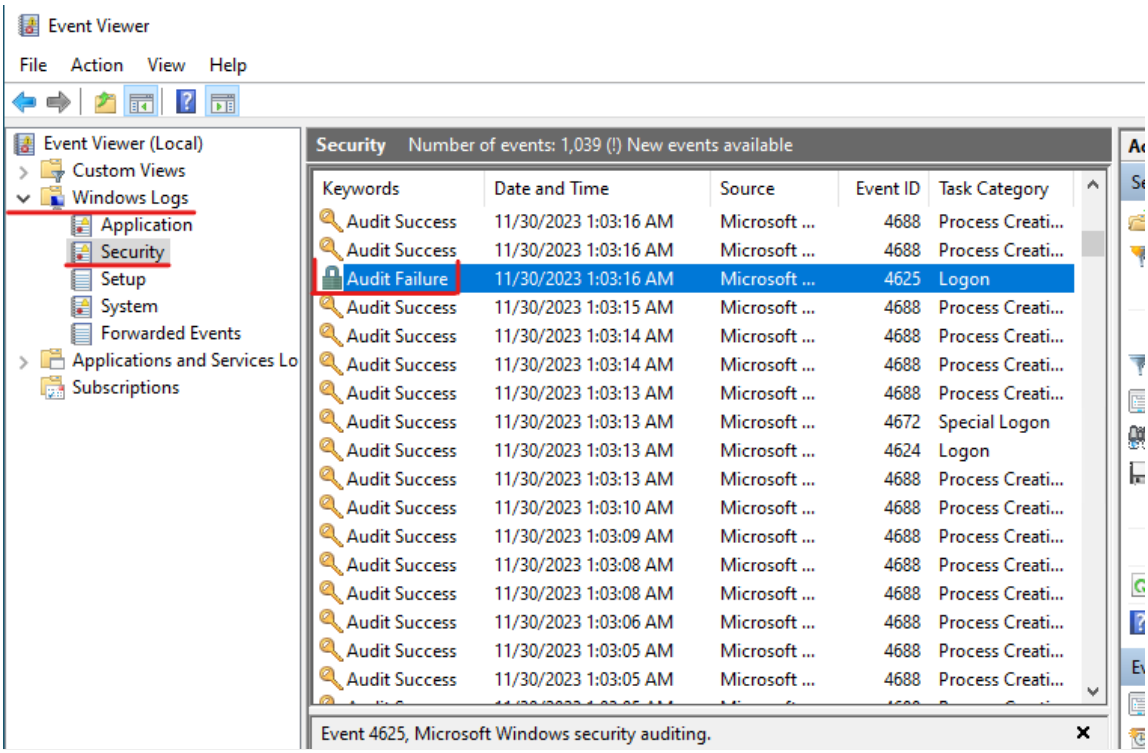


Once inside of our VM, we will search for the Event Viewer in Windows search bar, this is the storage for the logs of the metrics that we want to track (failed RDP attempts).

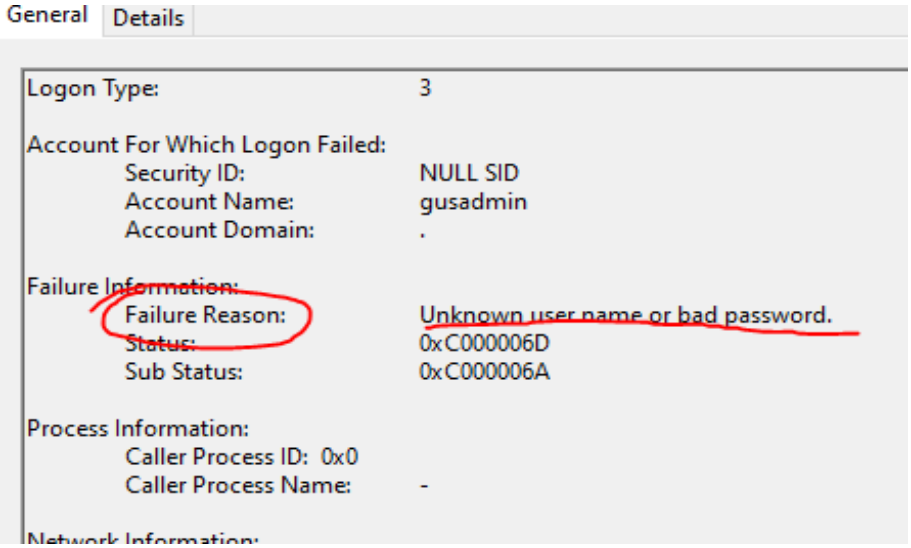
Esteban E. Valverde



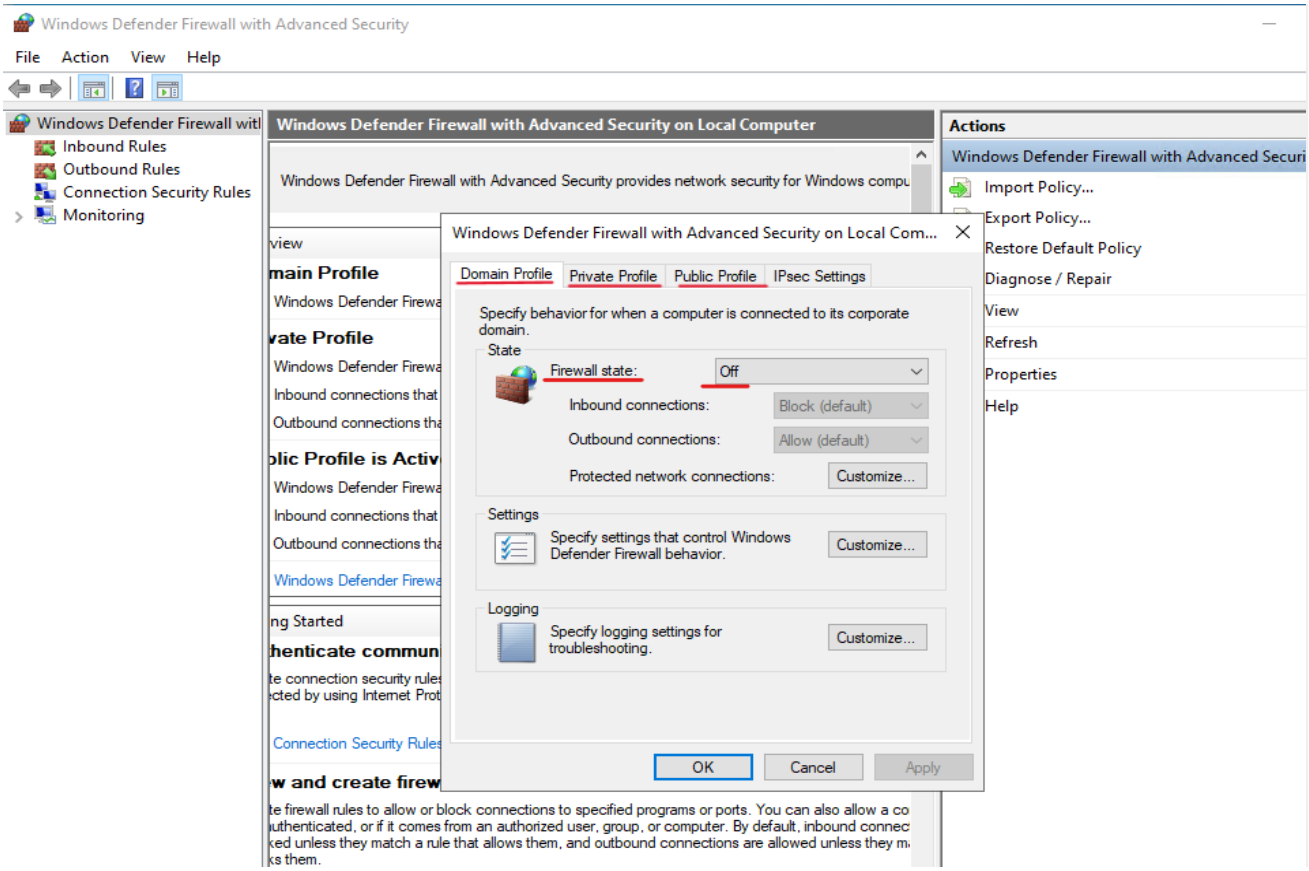
We are going to be analyzing any '**Audit Failure**' logs, under **Windows Logs > Security**



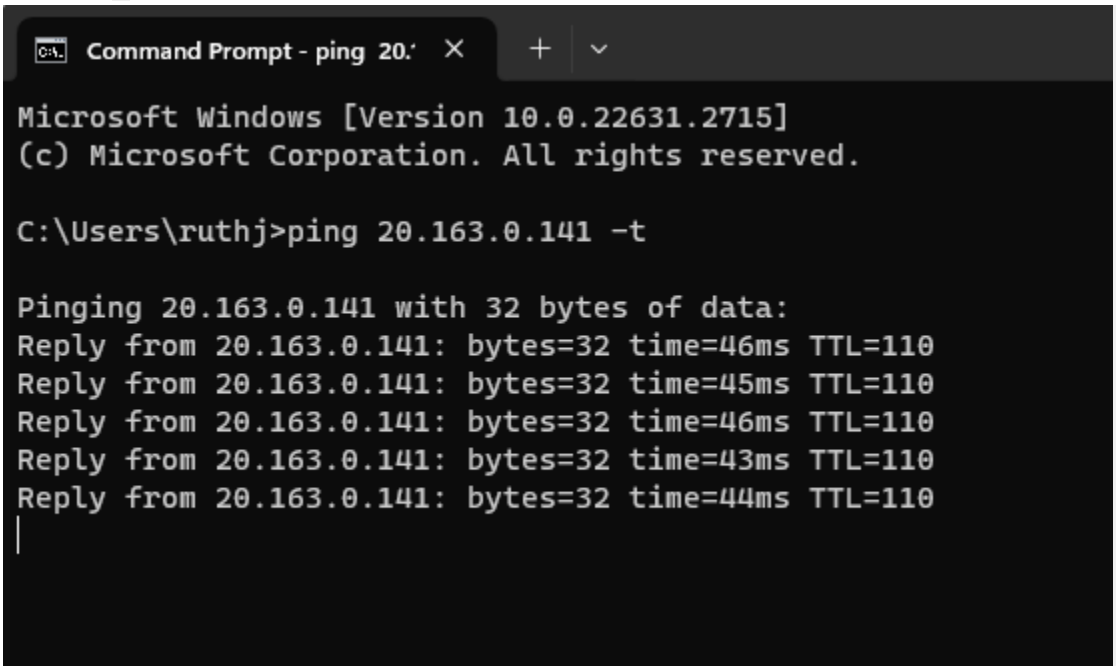
We can expand on these logs details and see important information such as the account name and failure reason, which was a bad username or password in this case. By monitoring and analyzing these failed RDP connection attempts, we gain insights into the persistence and determination of these attackers, ultimately strengthening our cybersecurity defenses.



Next, I must make sure our domain, private, and public profile's firewalls are disabled so that it can be pinged by anyone on the internet.



We can verify our work by going to our host machine's terminal and using the ping command with the -t flag to make sure the VM can be pinged without issues.



Then, I supported my project with a power shell script that will extract information from Windows Event Logs related to failed RDP login attempts, and create geographical data based on the source IP address. Here are the key components of the script:

1. **Importing Necessary Modules**:

   The script begins by importing the required PowerShell modules, including , , and .

2. **Setting Variables**:

   : This variable specifies the name of the Windows Event Log you want to target. In this case, it's set to 'Security' to access security-related events.

3. **Querying Event Logs**:

   The script queries the 'Security' event log for specific event IDs related to failed RDP login attempts. It uses the cmdlet to filter the events and stores the results in the variable.

4. **Iterating Through Events**:

   The script enters a loop to process each event in the array. It extracts relevant information from each event, such as the username, timestamp, source IP address, and more.

5. **Extracting Data**:

   Regular expressions are employed to extract specific details from the event description, which is stored in the property of the event. Information like the username, IP address, and other relevant data is extracted using regex patterns.

6. **Storing Data**:

   The extracted data is stored in a custom log file named 'failed_rdp.log' located in the directory. The script appends this information to the log file.

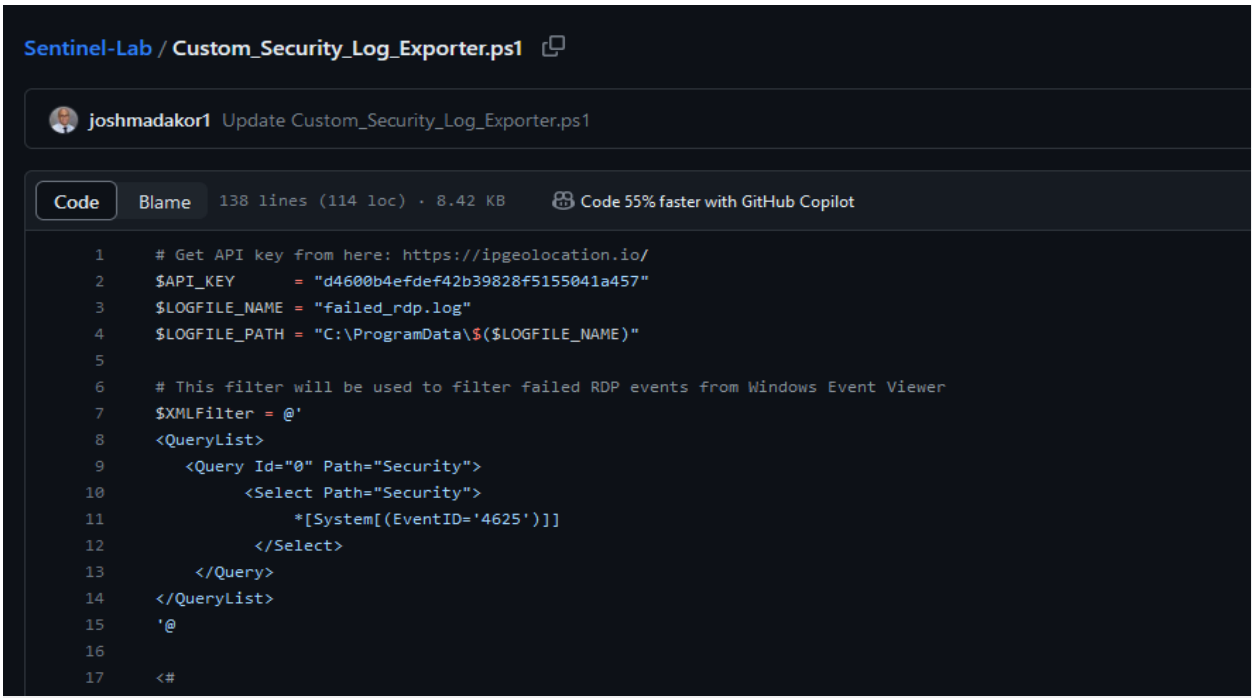
7. **Geolocation Data**:

   The script also includes logic for obtaining geolocation data based on the extracted IP address. It appears that this feature isn't implemented directly in the script, but it instructs users to obtain an API key from 'https://ipgeolocation.io/' and paste it into the script for future use. This API key would be used to determine the geographical location associated with each IP address.

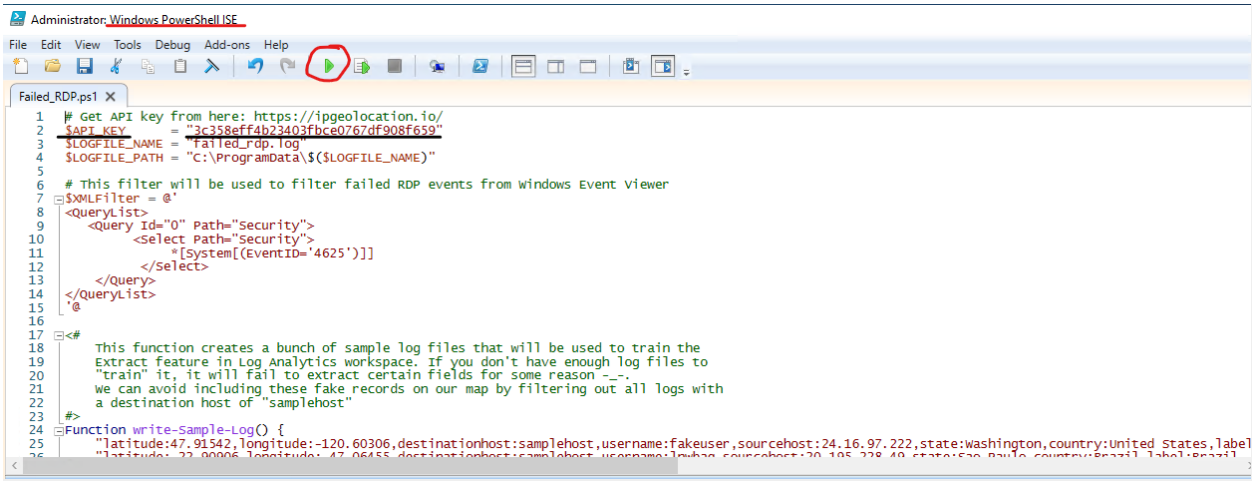
8. **Running the Script**:

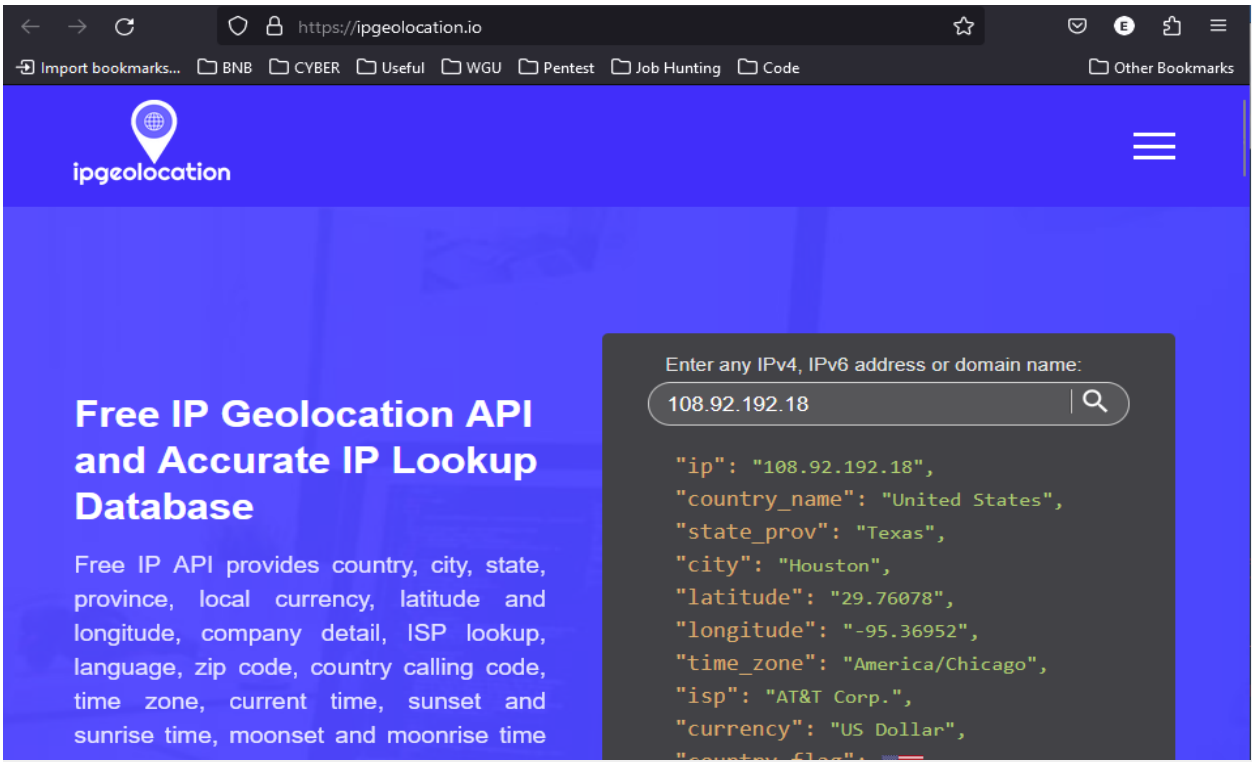
   To execute the script, users can save it as a '.ps1' file and run it in a PowerShell environment. The script runs through the defined event logs, extracts the necessary information, and appends it to the 'failed_rdp.log' file.

The script can be found here: https://github.com/joshmadakor1/Sentinel-Lab/blob/main/Custom_Security_Log_Exporter.ps1

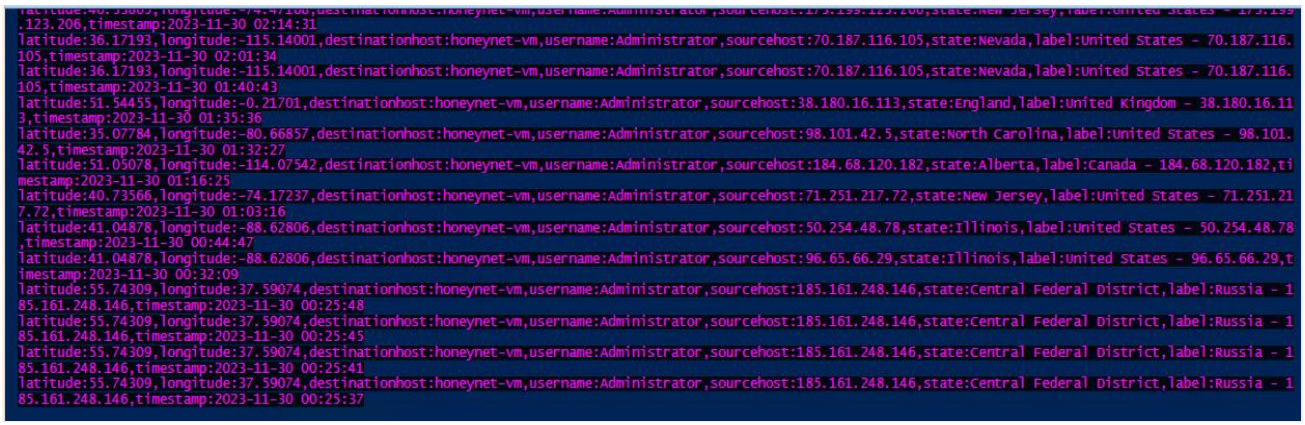


I copied this code and pasted it into a new .ps1 file inside my VM's PowerShell ISE. Then, I went over to https://ipgeolocation.io and signed up for a free account to get my API key to insert into the code for proper IP geolocation.

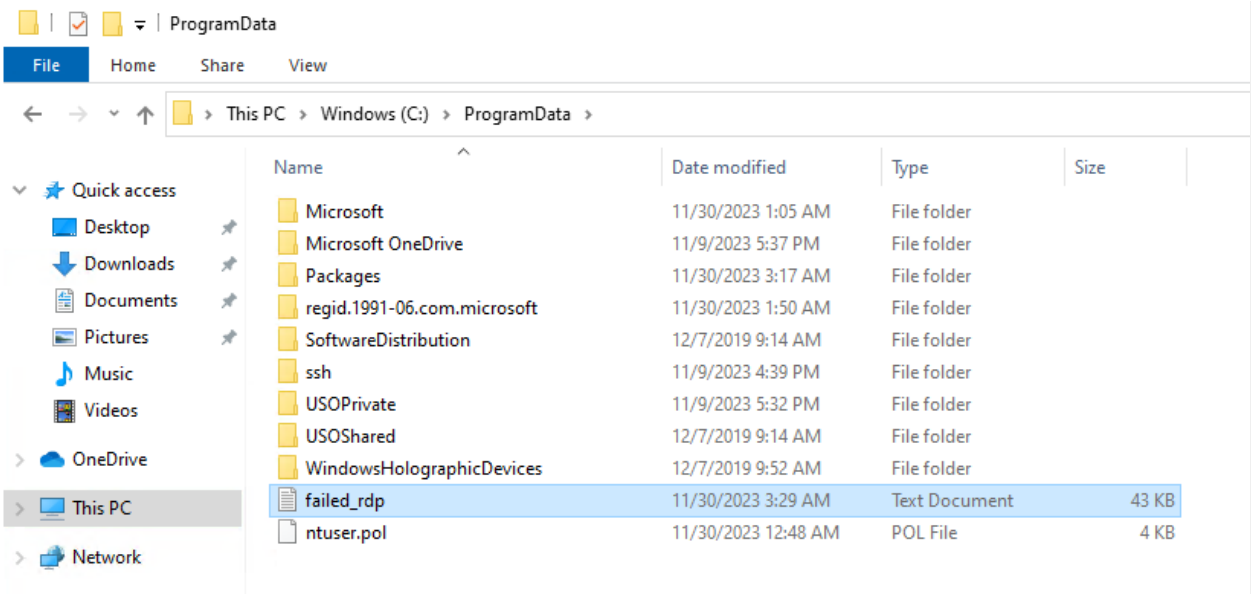




The next step will be to simply run my code, and after only a few minutes I could already start to see the script in action inside of the VM's PowerShell CLI:



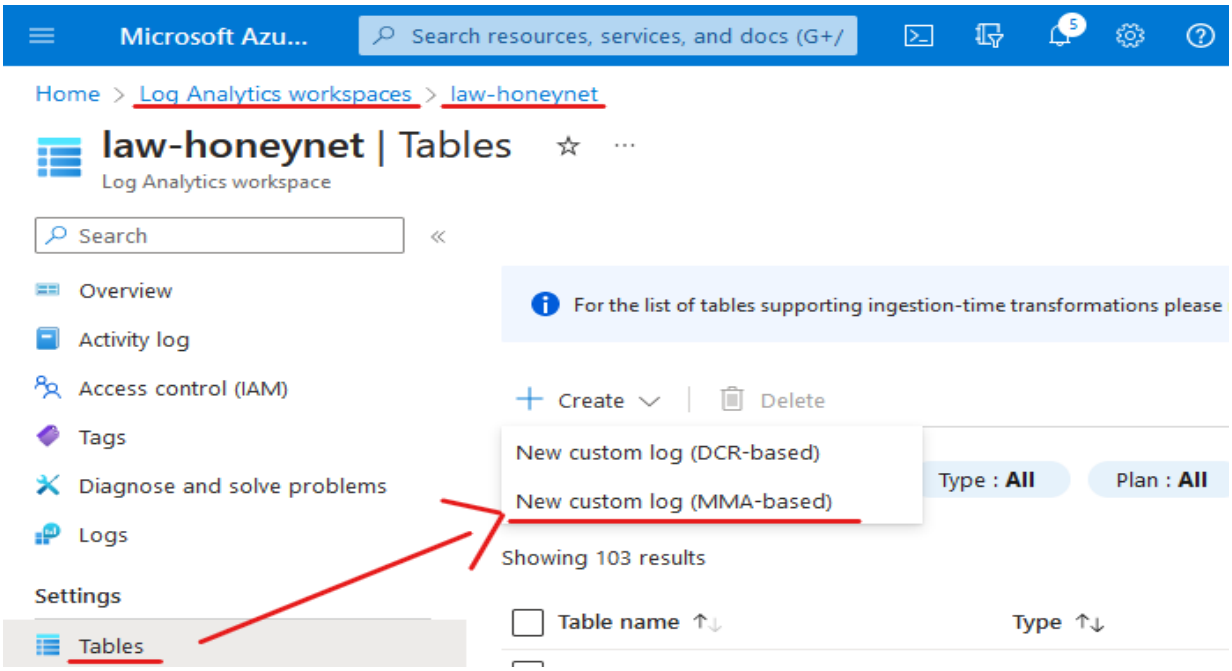
The script saves the desired logs into a hidden folder C:\ProgramData\failed_rdp.log



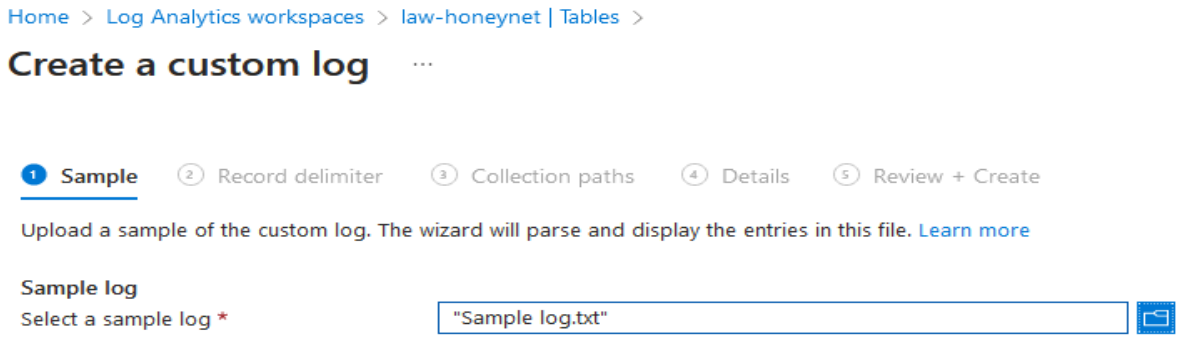
Step 4: Set up Custom Log Files and Query

Up next, before creating our log query, I must create a custom log file from the failed_rdp.log file in our VM for Azure to collect our data in an organized way.
To do this, I first need to go back to **Home > Log Analytics Workspaces** in my Azure account and click the LAW we created previously. Select **Tables** on the left, the **Create > New Custom Log (MMA-based)**

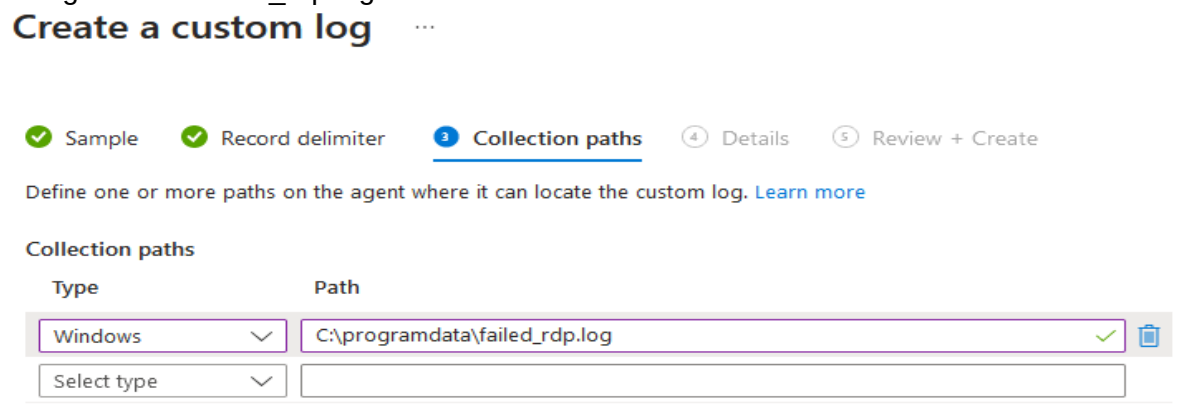
Esteban E. Valverde



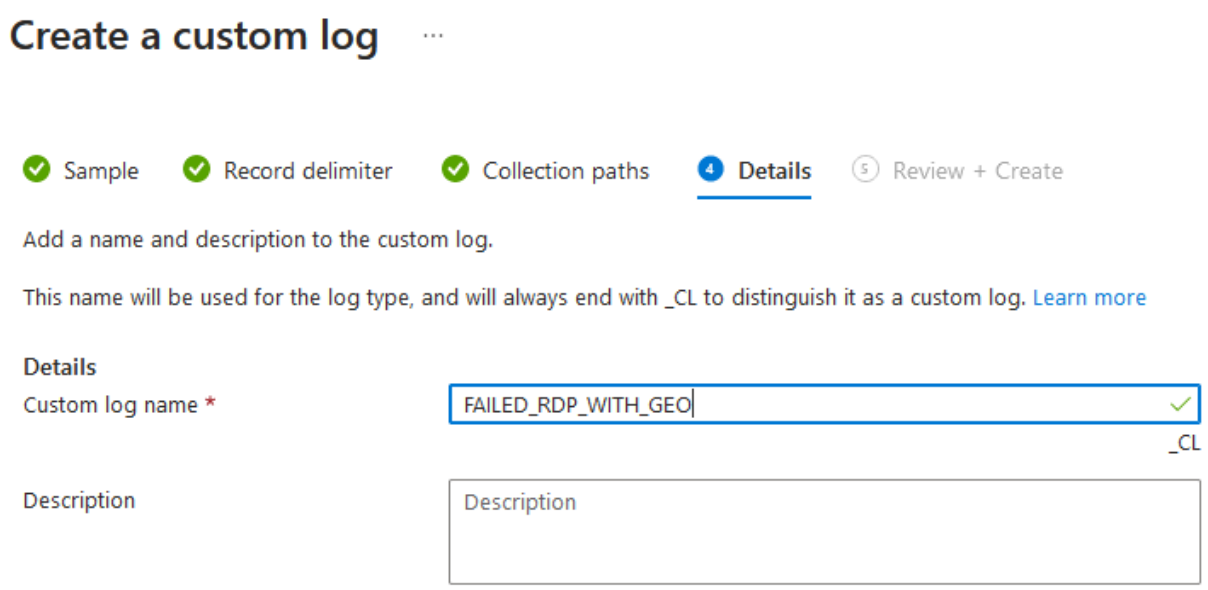
It's going to ask us for a custom log file on our local host. Since we are looking to use the data from our Virtual Machine's failed RDP Logs, I am going to have to go back into my VM, copy the contents of the failed RDP Log, create a file on my local host, and then upload that file to Azure.



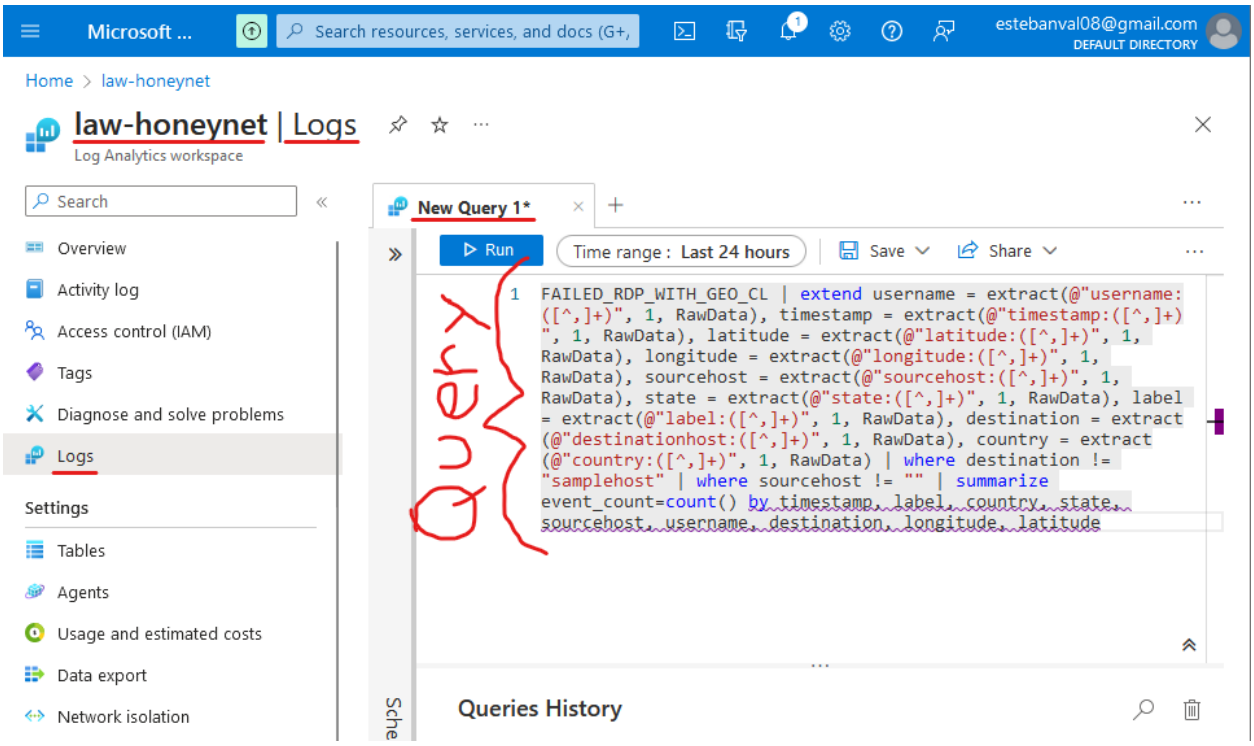
Just like in the script (unless it was modified), the file will be saved under C:\ProgramData\failed_rdp.log:



I then proceeded to name the custom log, and then reviewed and submitted the changes.



After creating the log, we want to go back to the main log page, select logs from the left menu, and we'll be met with the query field.
We're going to build a query that will parse our log files to output the appropriate data to be able to plot to the map in sentinel.
The query we are going to run is:



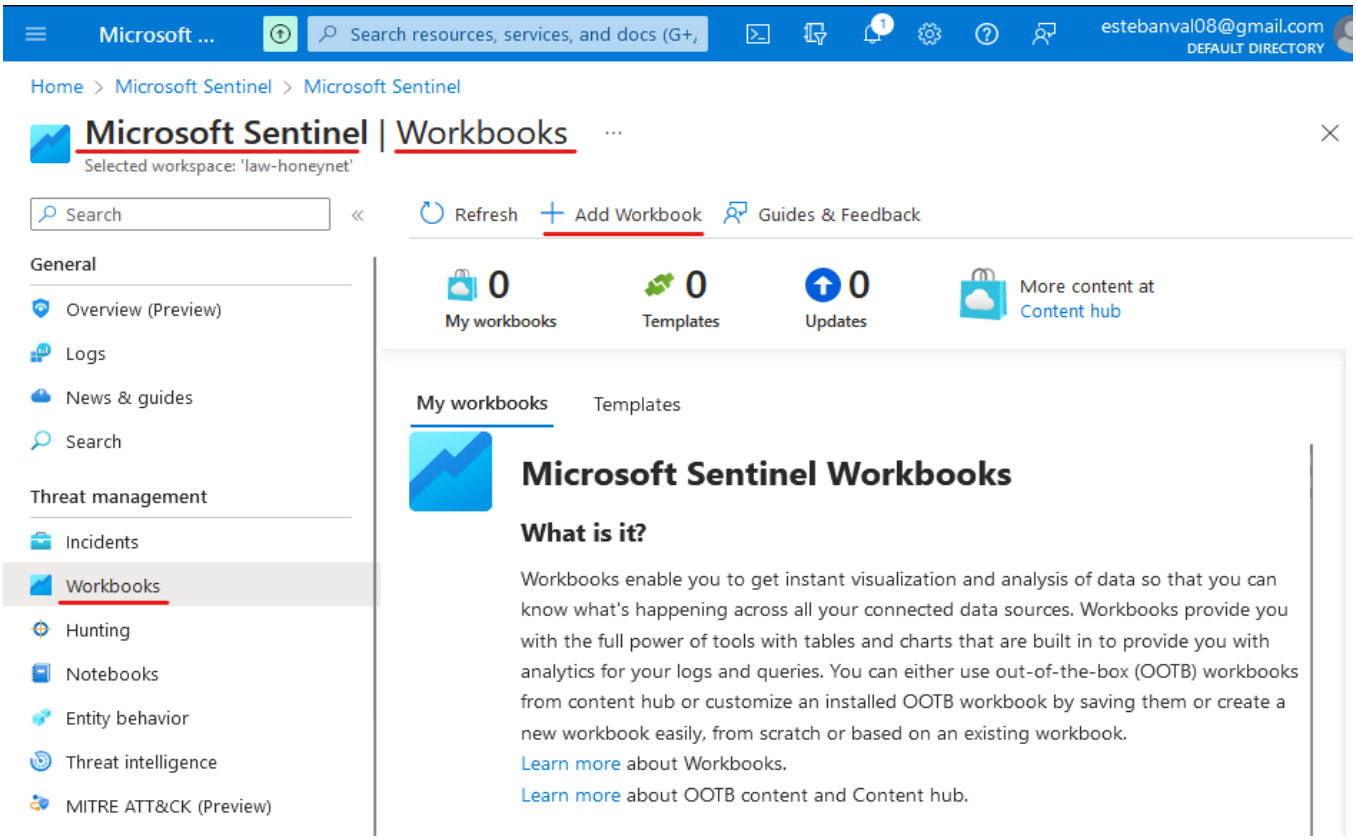
In detail, we start the query by choosing the custom log table we created, in this case "**FAILED_RDP_WITH_GEO_CL**", then we will use "extend" to extend the data by extracting the username field from the RawData field ("**extract(@"username:([^,]+)", 1, RawData)**". We will repeat the section we used to extract the username field, and we are also going to extract the timestamp field, latitude, longitude, sourcehost, state, label, destination, and country.

After extracting all these fields, we will instruct the query to exclude any results where the 'destination' field is equal to 'samplehost', since the PowerShell script I used earlier contains some sample logs that we do not want to illustrate. We also want to exclude any results where 'sourcehost' is an empty field to avoid any false positives. Finally, we will summarize the event count for better illustration.

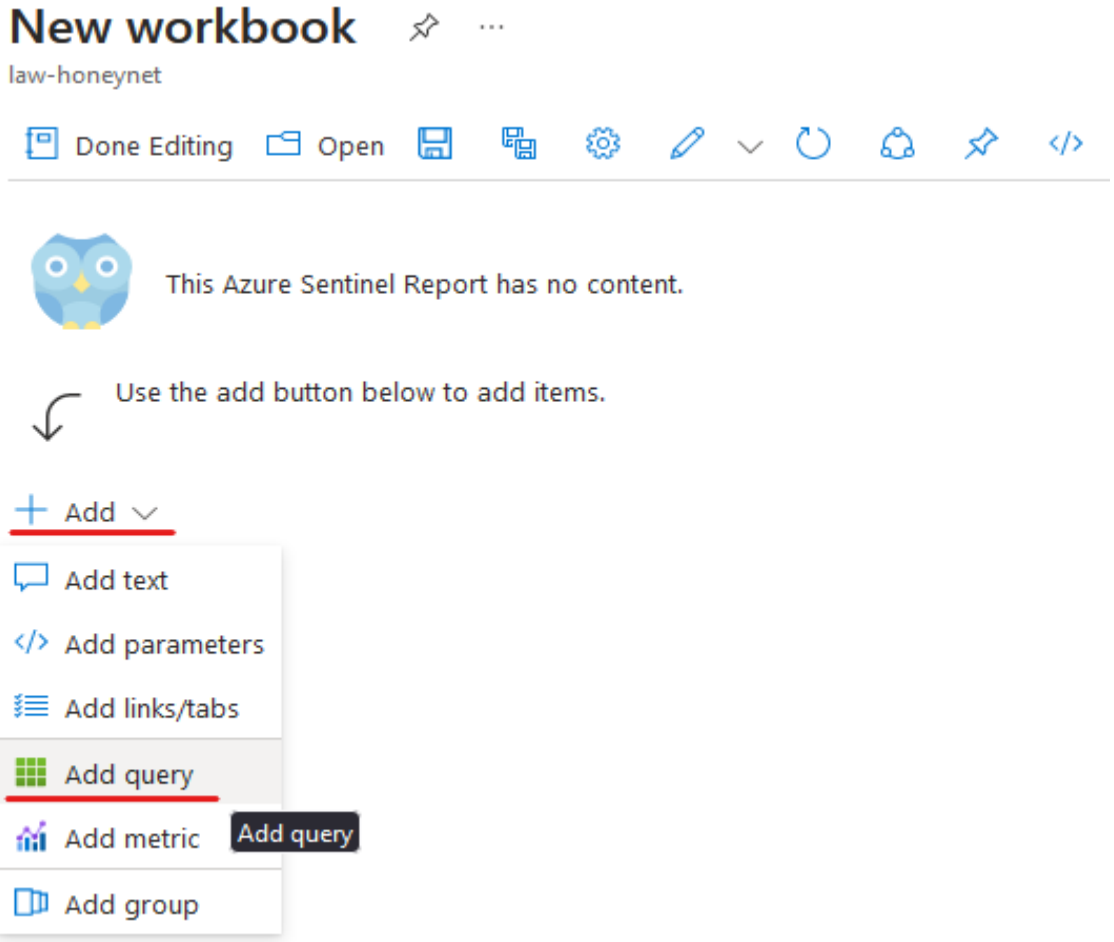
The final output provides structured data that can be used for visualization and analysis, particularly for plotting geographical information related to failed RDP connection attempts on a map in Microsoft Sentinel.

Step 5: Construct Sentinel Map

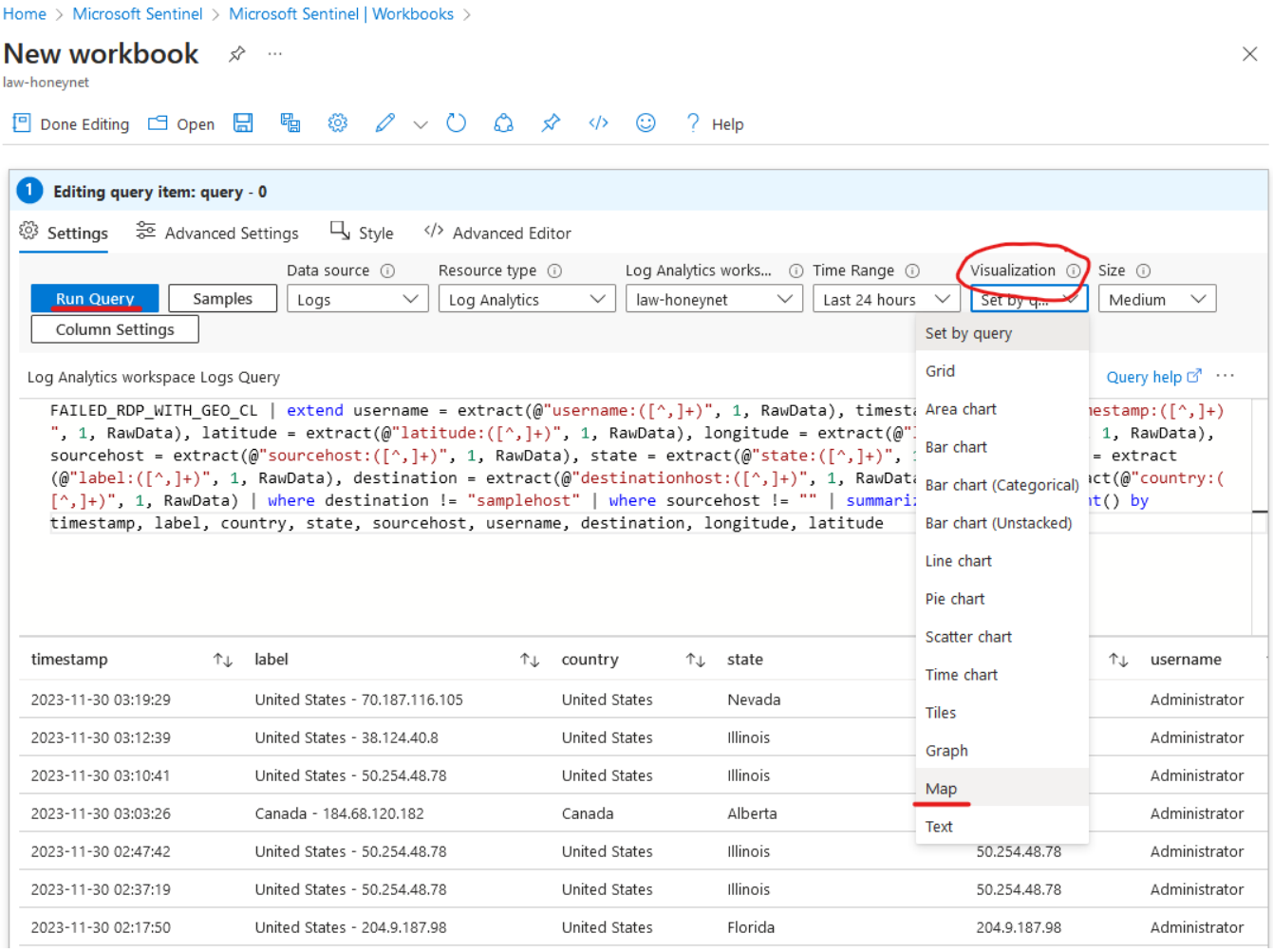
It's time to start putting all our work together in Sentinel! I will start by going to Microsoft Sentinel in my Azure, clicking on '**Workbooks**' on the left menu, and then I'll hit the **Add Workbook** button.



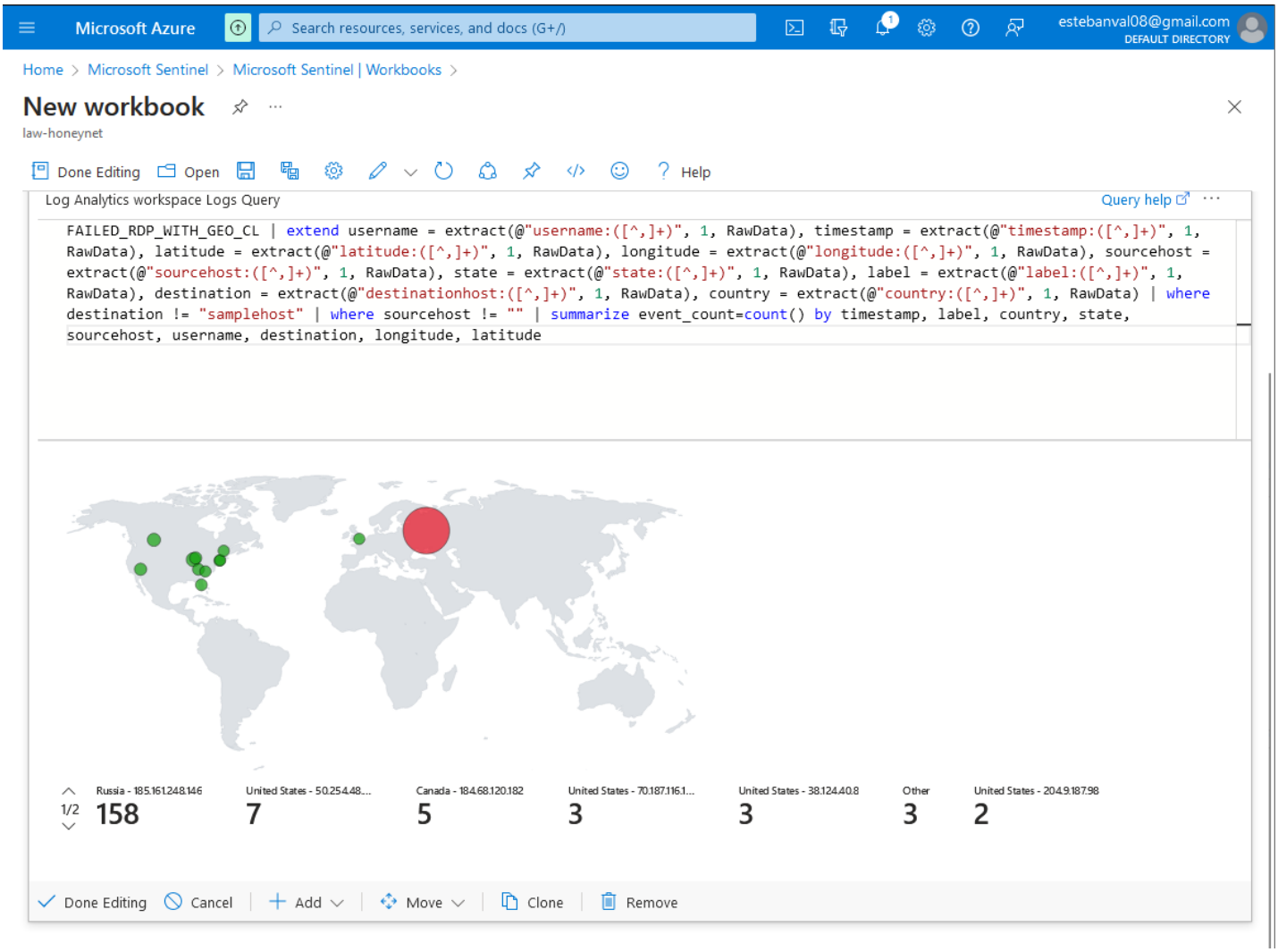
Before adding my desired widget, I need to delete any default ones by clicking '**Edit**' and deleting the widgets. Then I will click '**Add**' and '**Add query**'



I am going to paste the same query used earlier for the Custom Log Table, and then I am going to run it. After running it I will click '**Map**' under "Visualization", and then we will be able to see our map!

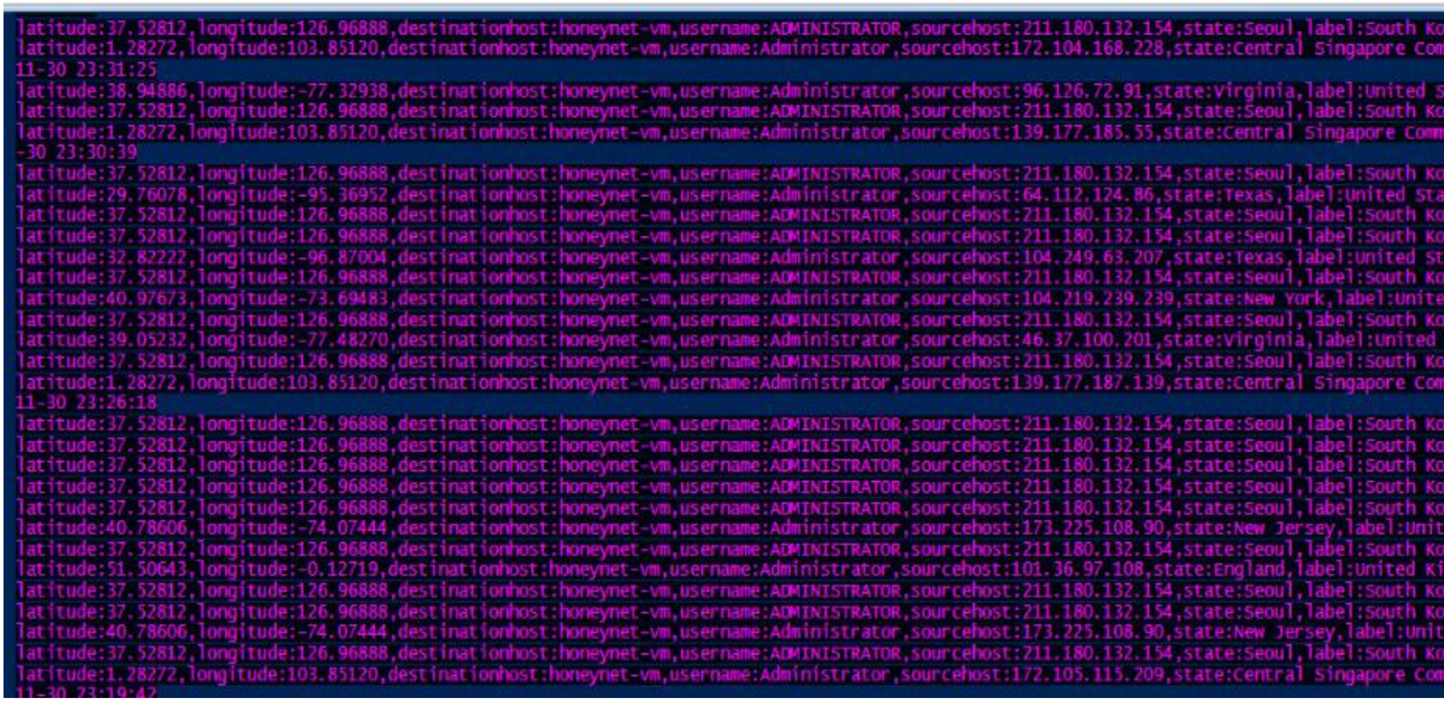


After approximately 1 hour we can see very intriguing log data illustrated in our Sentinel map about failed RDP login attempts to our honeypot VM. We can infer that the IP address in Russia is performing some sort of brute force attack due to the amount of login attempts in such a short period of time.



Going back to our PowerShell script inside of our VM we can observe real-time log recollection of these events.

Esteban E. Valverde

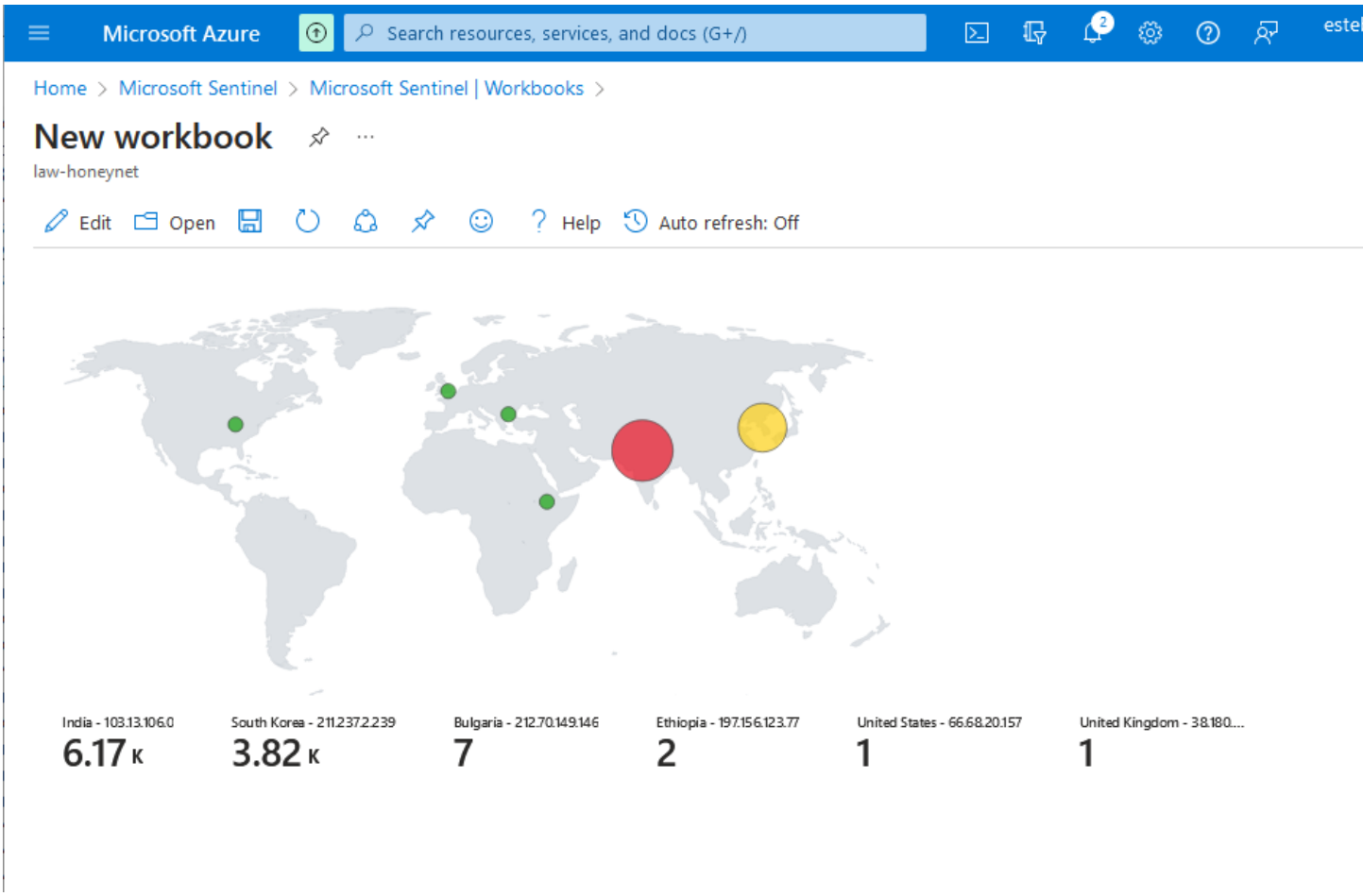


```
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:1.28272,longitude:103.85120,destinationhost:honeynet-vm,username:Administrator,sourcehost:172.104.168.228,state:Central Singapore Com
11-30 23:31:25
latitude:38.94886,longitude:-77.32938,destinationhost:honeynet-vm,username:Administrator,sourcehost:96.126.72.91,state:Virginia,label:United S
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:1.28272,longitude:103.85120,destinationhost:honeynet-vm,username:Administrator,sourcehost:139.177.185.55,state:Central Singapore Comm
-30 23:30:39
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:29.76078,longitude:-95.36952,destinationhost:honeynet-vm,username:Administrator,sourcehost:64.112.124.86,state:Texas,label:United Sta
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:32.82222,longitude:-96.87004,destinationhost:honeynet-vm,username:Administrator,sourcehost:104.249.63.207,state:Texas,label:United St
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:40.97673,longitude:-73.69483,destinationhost:honeynet-vm,username:Administrator,sourcehost:104.219.239.239,state:New York,label:Unite
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:39.05232,longitude:-77.48270,destinationhost:honeynet-vm,username:Administrator,sourcehost:46.37.100.201,state:Virginia,label:United
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:1.28272,longitude:103.85120,destinationhost:honeynet-vm,username:Administrator,sourcehost:139.177.187.139,state:Central Singapore Com
11-30 23:26:18
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:40.78606,longitude:-74.07444,destinationhost:honeynet-vm,username:Administrator,sourcehost:173.225.108.90,state:New Jersey,label:Unit
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:51.50643,longitude:-0.12719,destinationhost:honeynet-vm,username:Administrator,sourcehost:101.36.97.108,state:England,label:United Ki
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:40.78606,longitude:-74.07444,destinationhost:honeynet-vm,username:Administrator,sourcehost:173.225.108.90,state:New Jersey,label:Unit
latitude:37.52812,longitude:126.96888,destinationhost:honeynet-vm,username:ADMINISTRATOR,sourcehost:211.180.132.154,state:Seoul,label:South Ko
latitude:1.28272,longitude:103.85120,destinationhost:honeynet-vm,username:Administrator,sourcehost:172.105.115.209,state:Central Singapore Com
11-30 23:19:42
```

After roughly 12 hours we can observe a few different IP’s that tried to login, but the two that definitely attract the most attention are the IP’s from India and South Korea, since we can observe that they definitely employed an extensive brute force dictionary attack in an attempt to gain access to my honeypot.

Note my API key malfunctioned and I had to issue a new one, hence why we might observe a bit different data than the previous map picture.

Esteban E. Valverde



Summary

In conclusion, the project aimed to bolster the security team's proactive measures by creating and monitoring a honeypot in Microsoft Azure. Focused on understanding attack techniques and detecting threats, the initiative involved building an Azure Virtual Machine, creating a Log Analytics Workspace, and connecting and configuring the virtual machine for monitoring failed RDP login attempts.

The deployment of a custom PowerShell script allowed for the extraction of valuable information from Windows Event Logs, enabling the creation of geolocation data based on source IP addresses. The integration of this data into Azure Sentinel facilitated the construction of a comprehensive map illustrating the geographical distribution of potential attackers and the frequency of unsuccessful attempts on the honeypot.

Through meticulous steps, including setting up custom log files, crafting queries, and constructing a Sentinel map, the project showcased the effectiveness of honeypots in attracting and analyzing global threat actor traffic. Real-time observations revealed diverse and persistent

attackers engaging in brute force attempts, emphasizing the project's success in providing actionable insights to strengthen cybersecurity defenses.