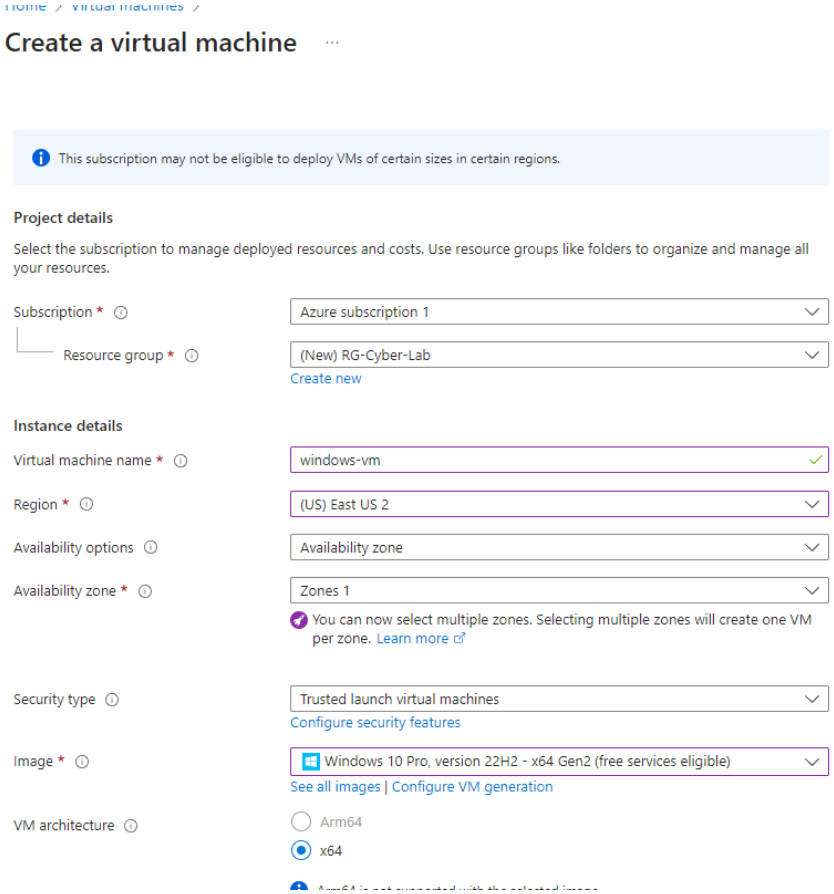
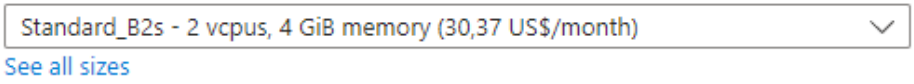
We will first create out windows 10 VM that will be used in our honeynet

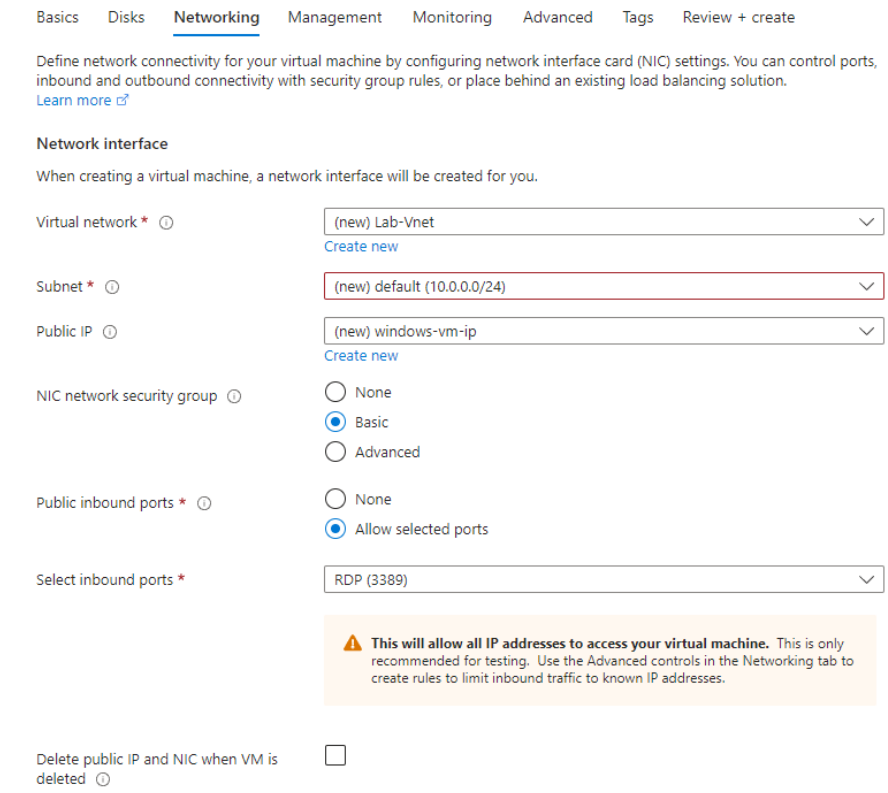


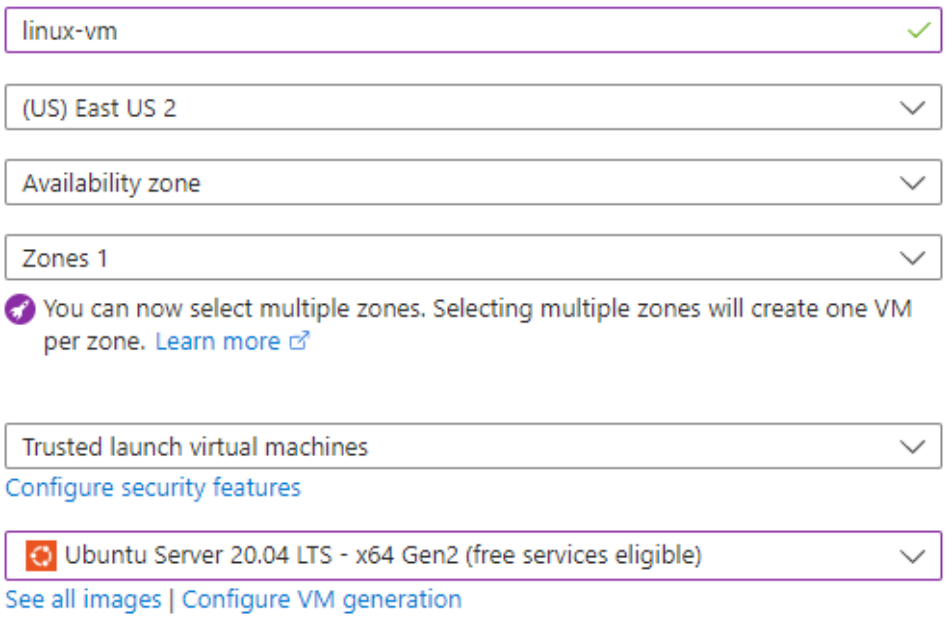
We ended up choosing a bit of a beefier RIG because otherwise we would get DDOS'd and out logs would stop feeding into our SIEM. Here is what I choose:



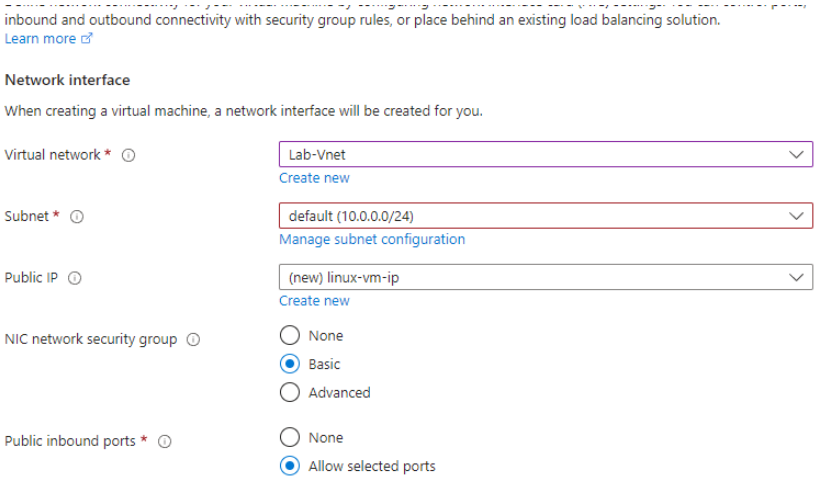
See all sizes

We pretty much left everything else as stock but made a new VNET as you can see here

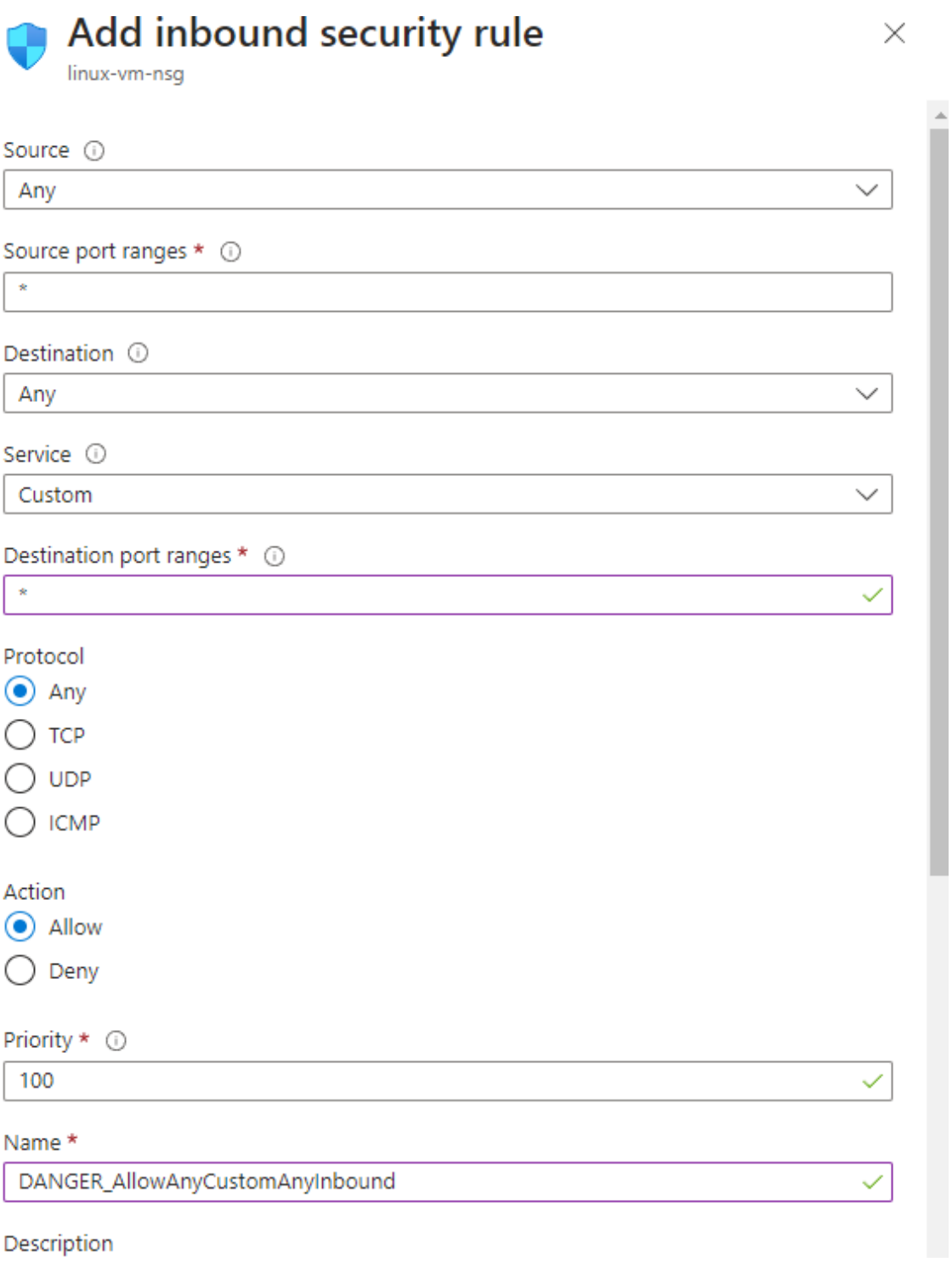




We will now create another VM but this time a linux machine in the same resource group as the other.

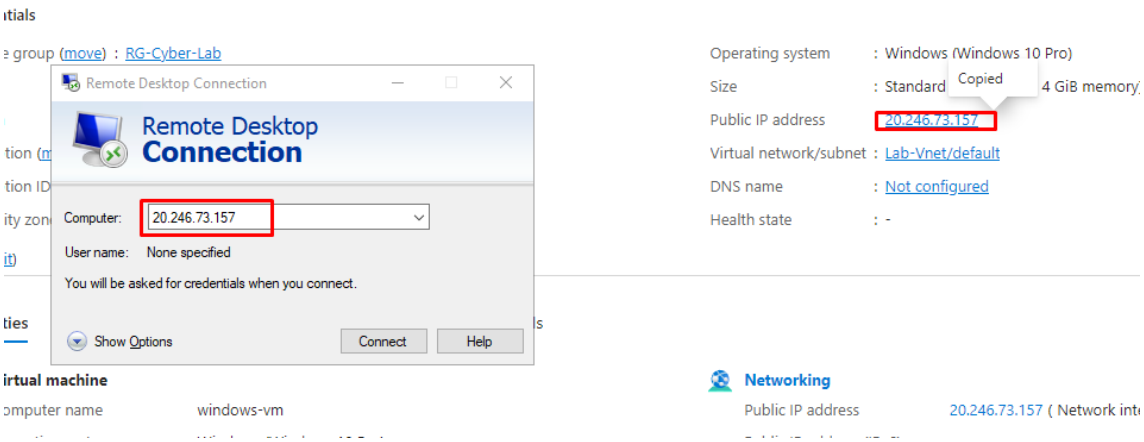


We will put it in the same VNET and the other machine as well so they can communicate with one another!



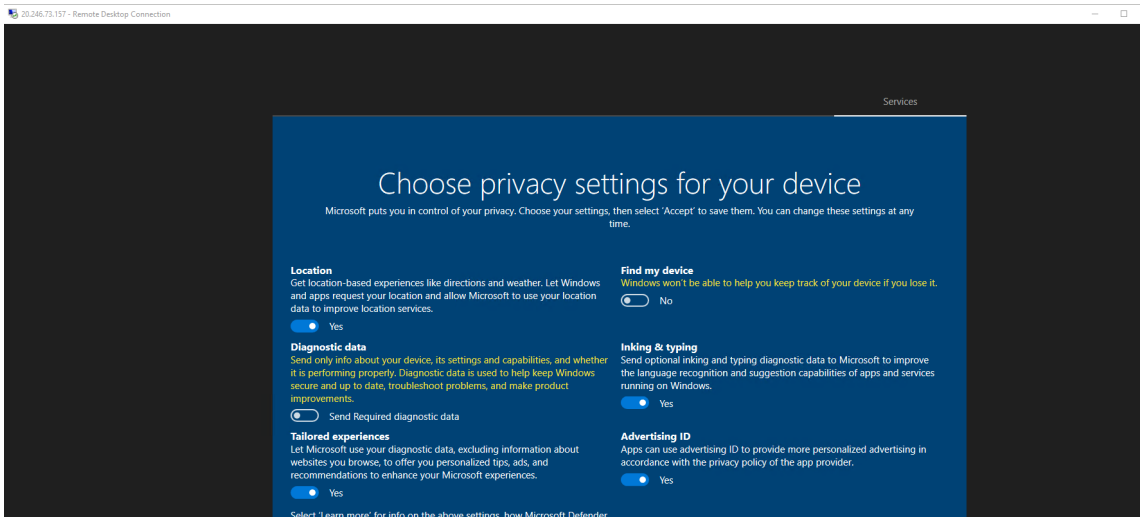
We will now also add an inbound rule to both of the VMs that allow any traffic to reach to them from the internet!

What we will do now is install the SQL Database onto the windows VM.



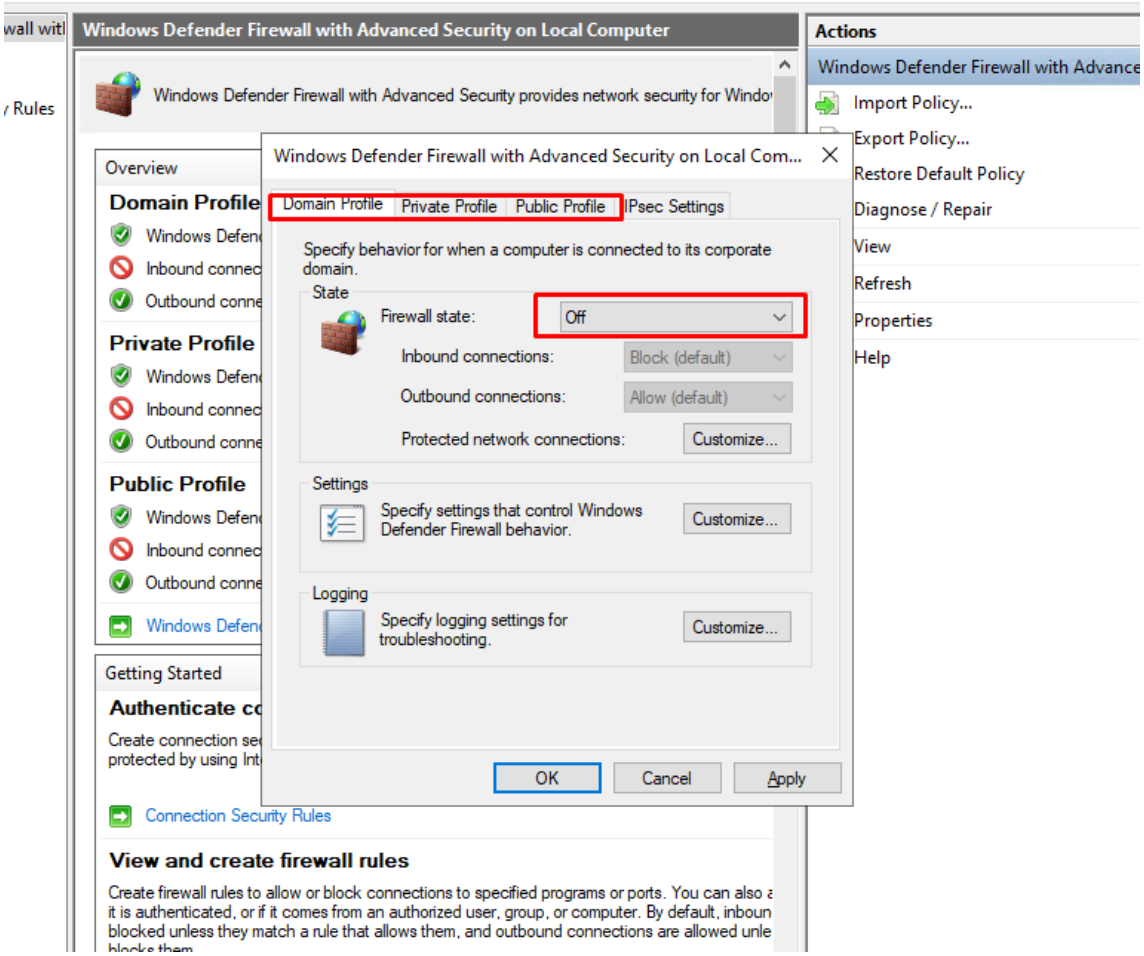
We will first RDP into the Windows VM like this.

We will use the User and Password we set before and we should be right here:



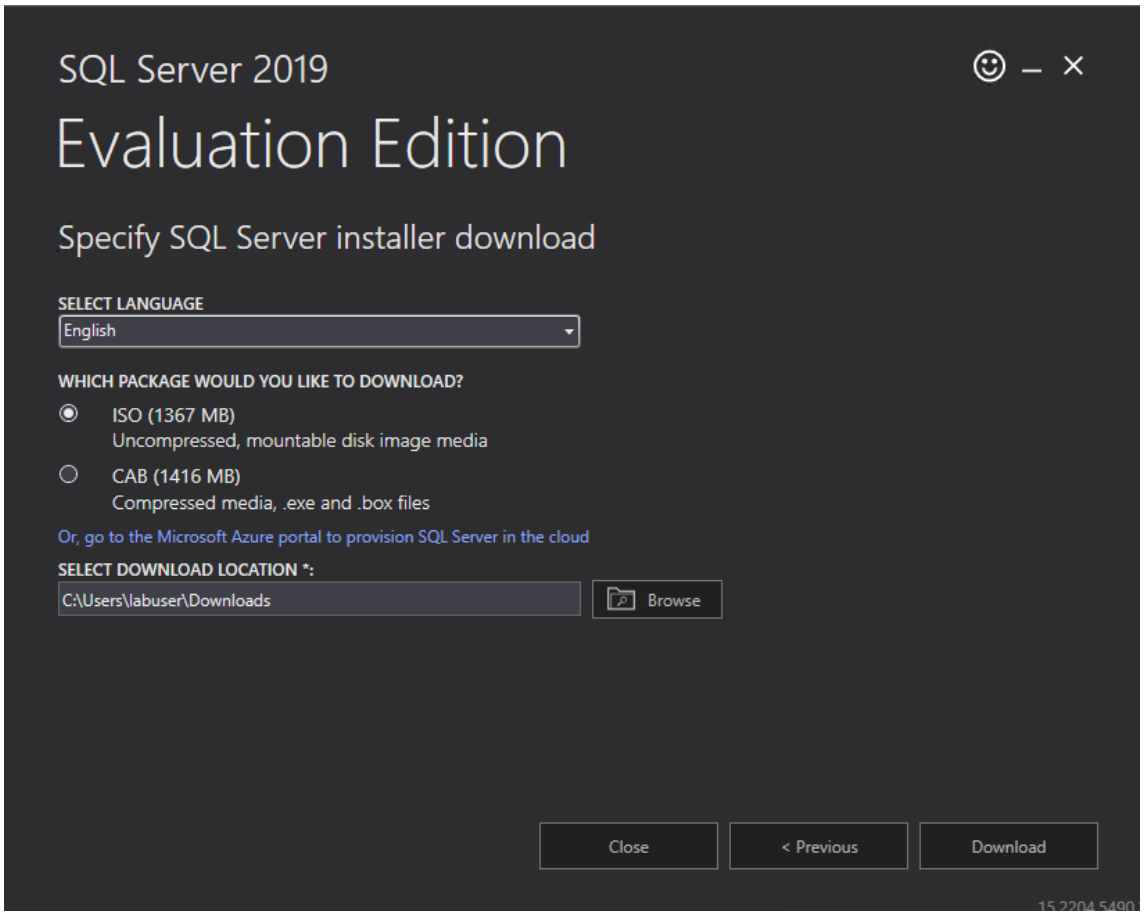
We will now proceed to Turn off the windows Firewall. (So, it can be easier to find online)

For example, with ping requests.

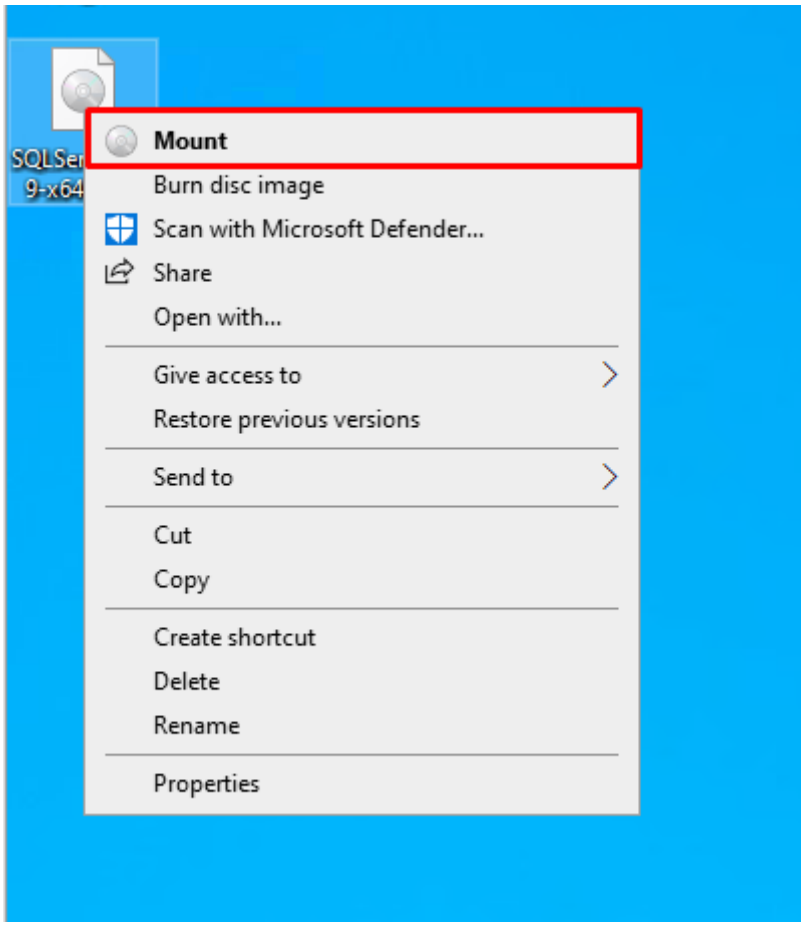


We will now proceed to install Install SQL Server Evaluation

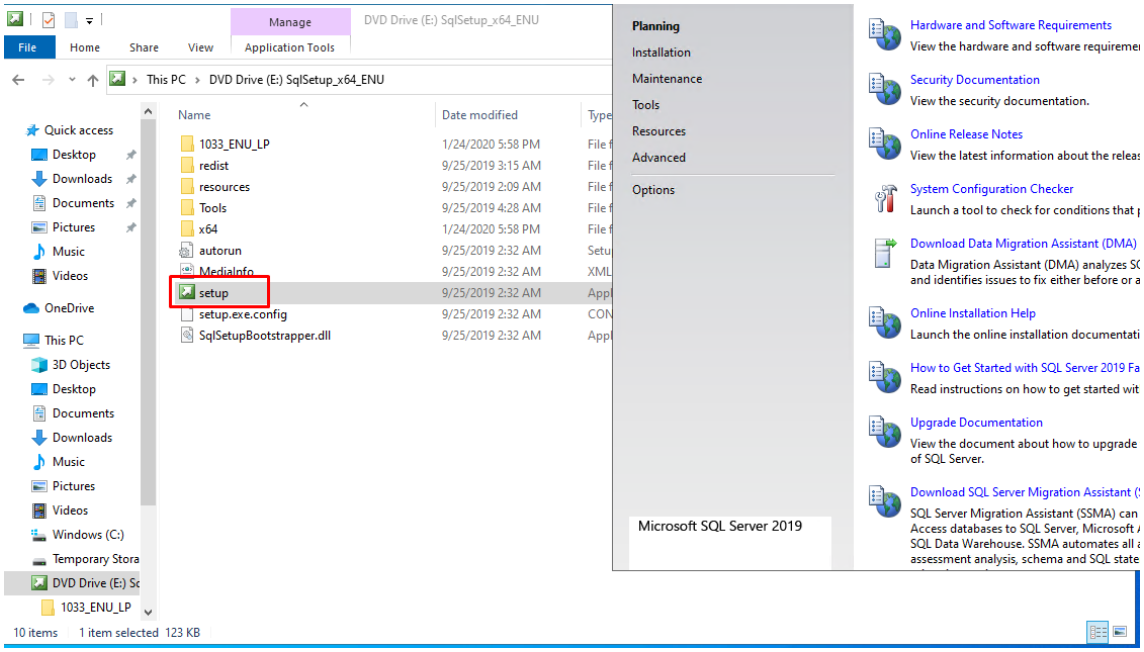
Please select your SQL Server 2019 download



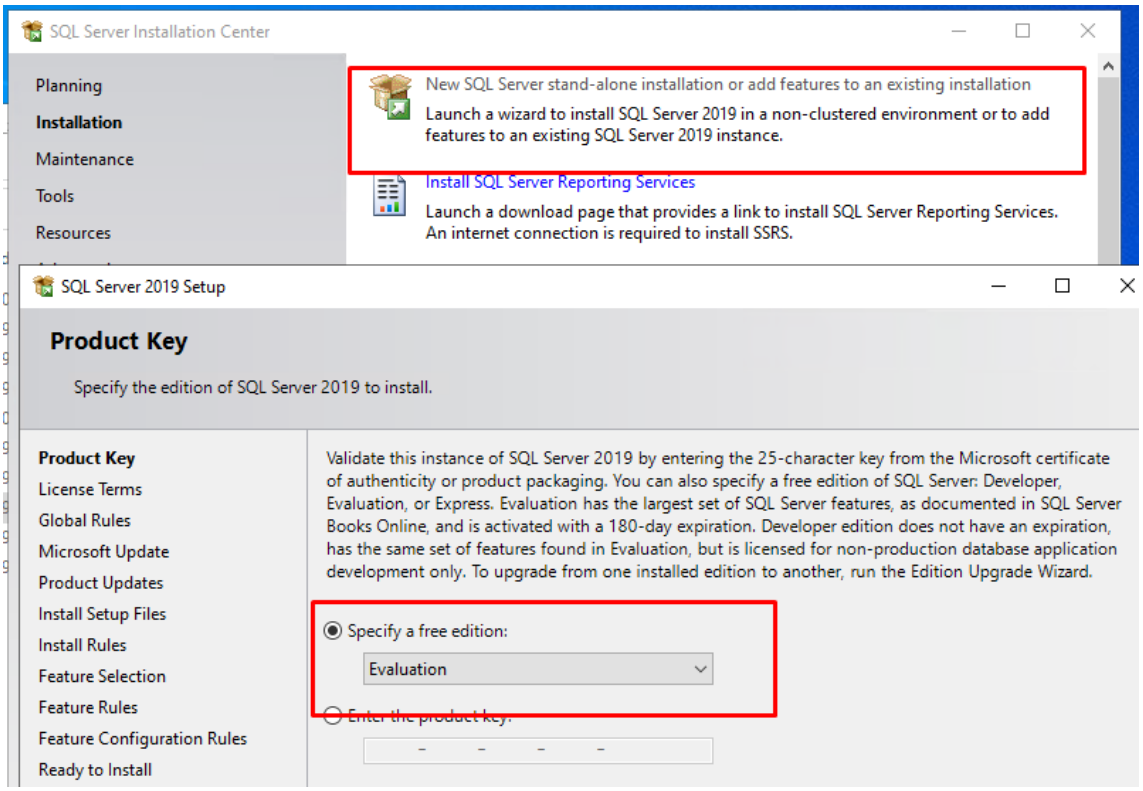
We will keep going with the installation.



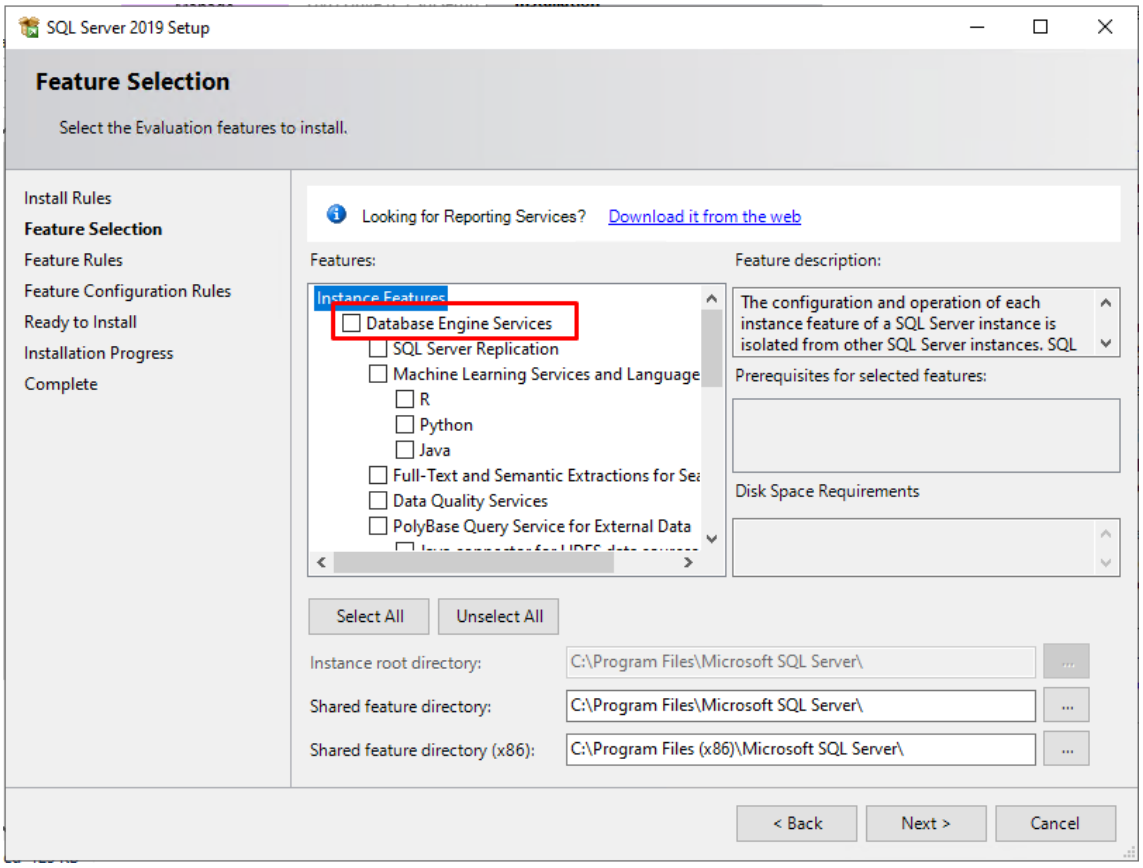
Since this is a ISO file we need to right click it and MOUNT!



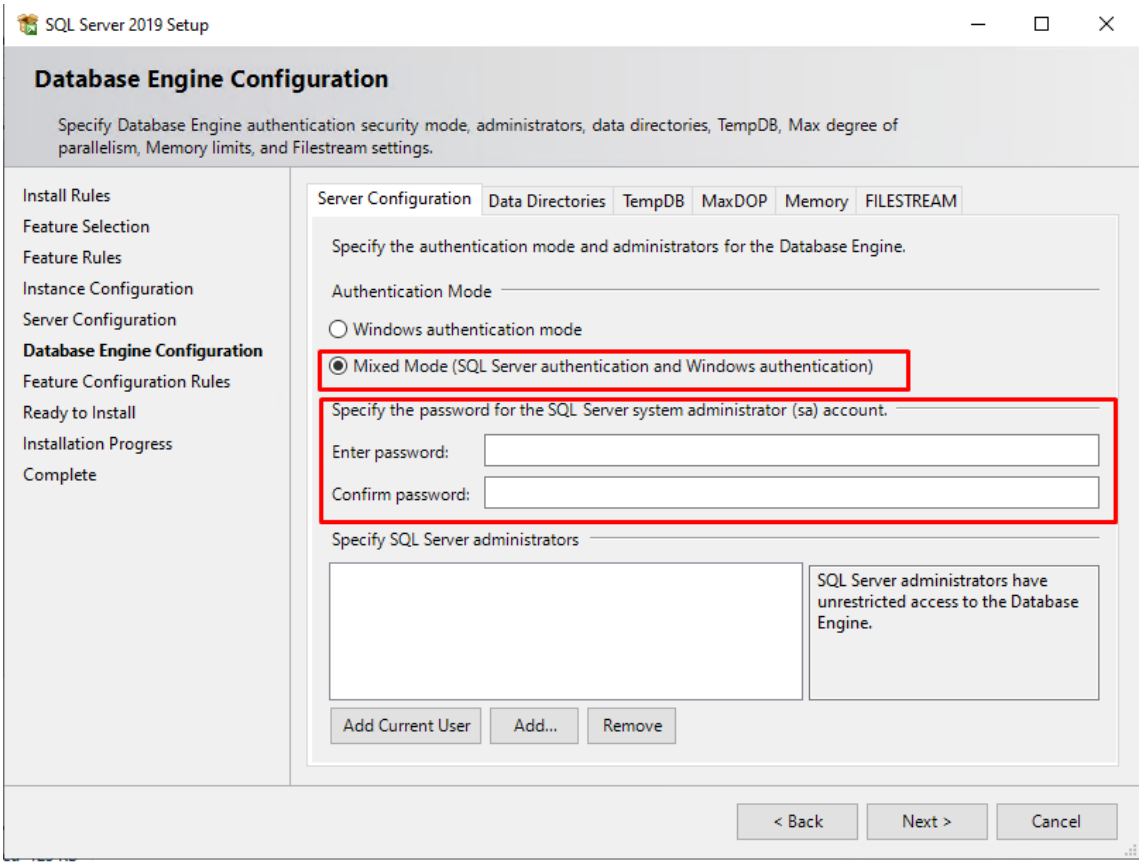
Now we run the setup.



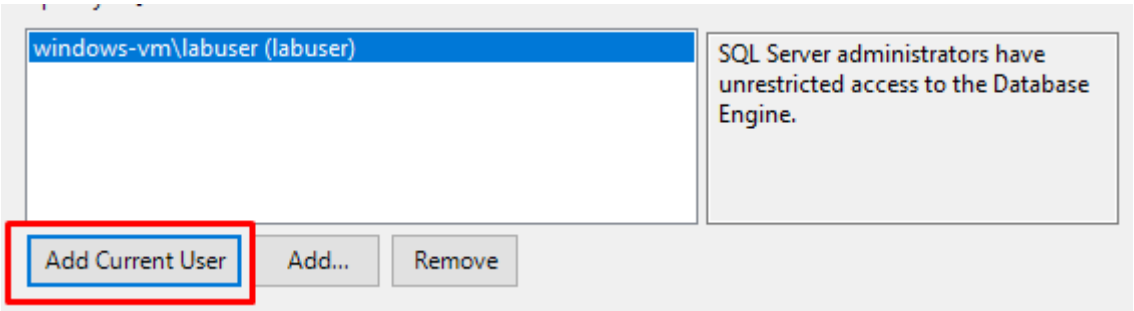
We will pick the first option and just leave it as evaluation.



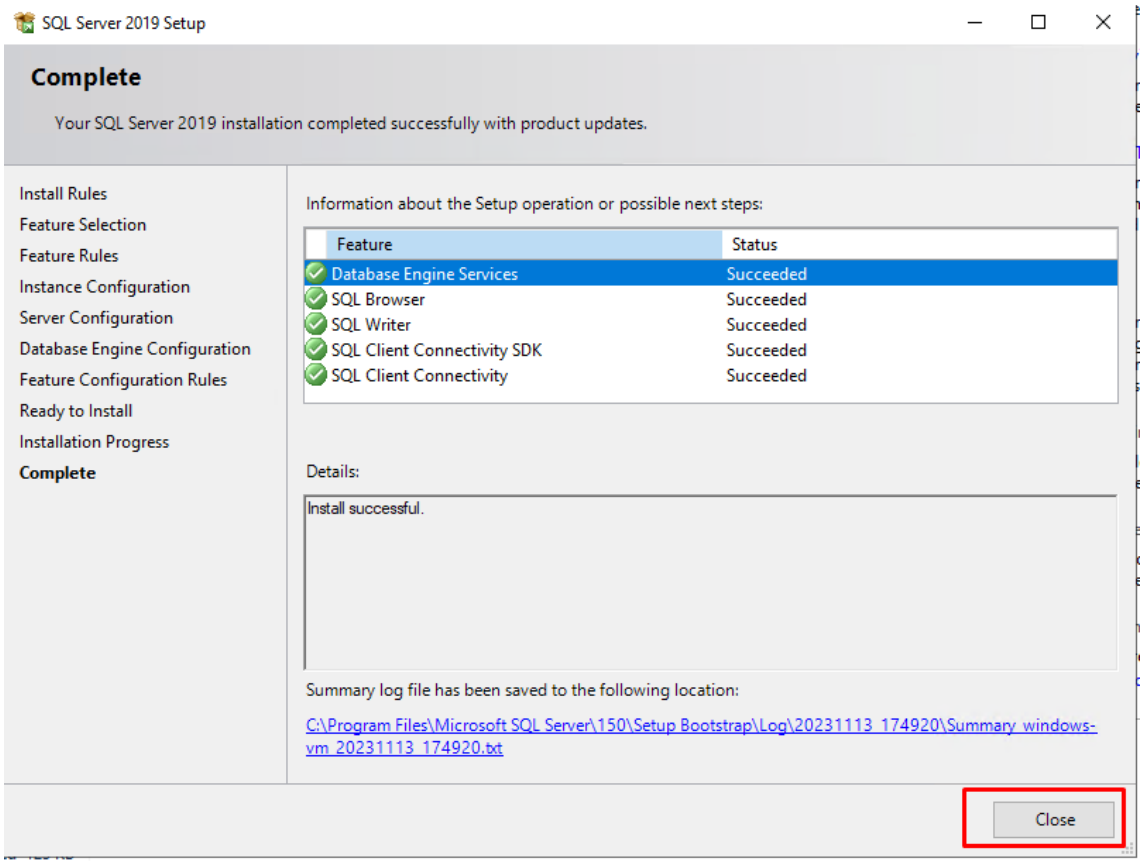
When we get to here we check the first option.



When we get to here, we will pick the second option and I will now input the admin password I want.



We will also add ourselves as an admin.

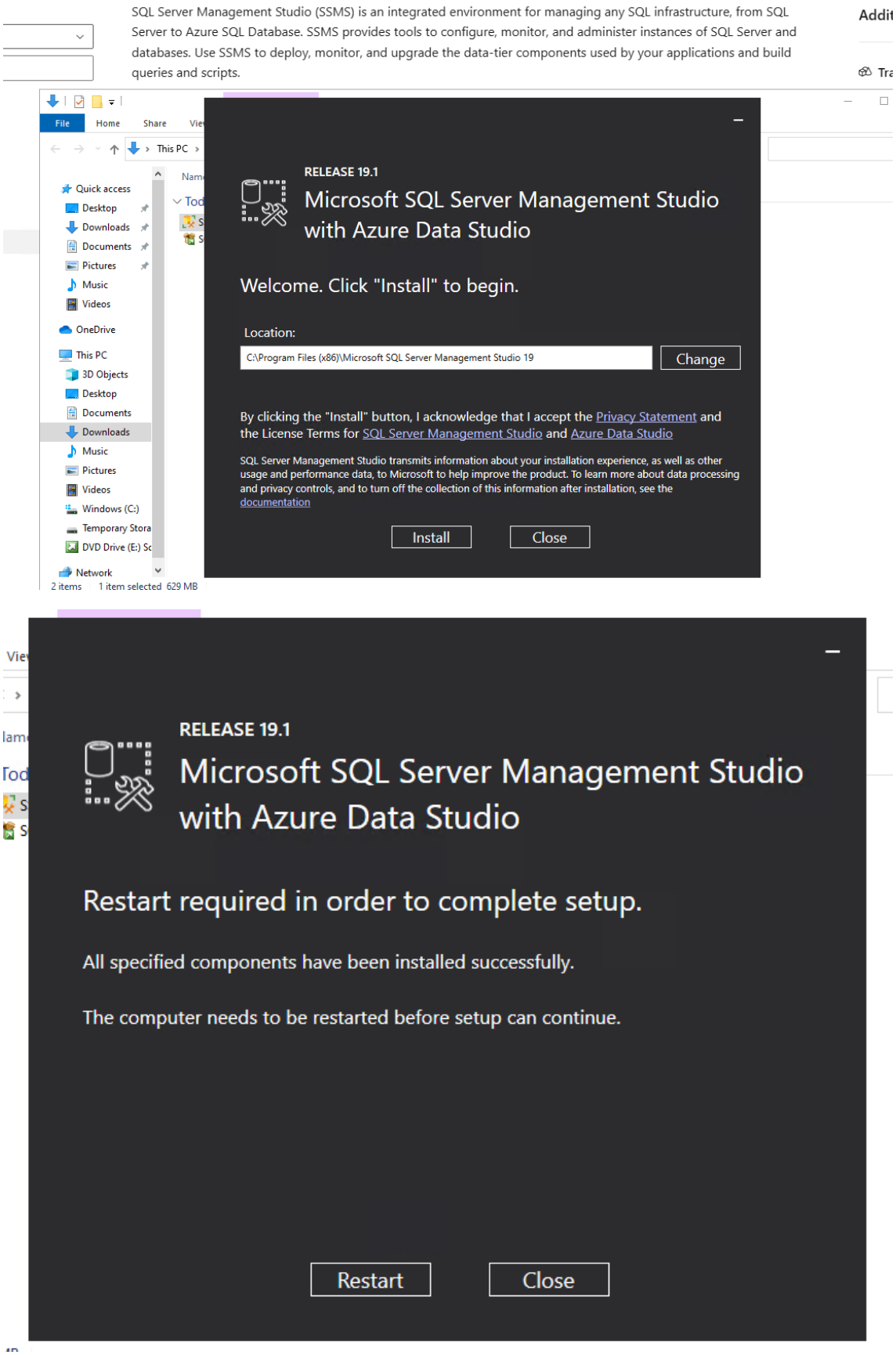


We keep going and when we get here, we just close out.

We will now install:

- SSMS (SQL Server Management Studio): https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms

Which we will use to login and generate logs for our SQL database.



We will now restart the VM to complete the install.

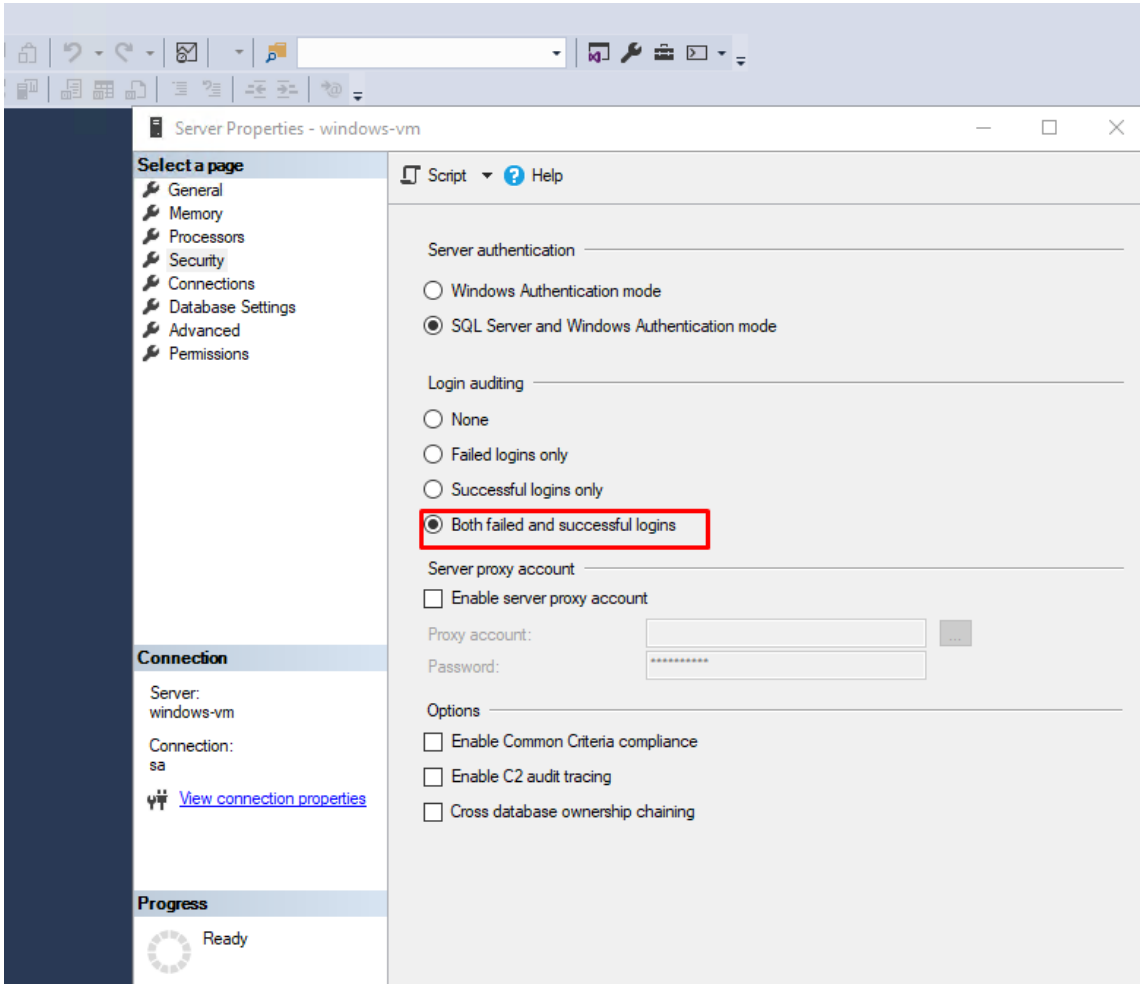
Next what we want is to

Enable logging for SQL Server to be ported into Windows Event Viewer

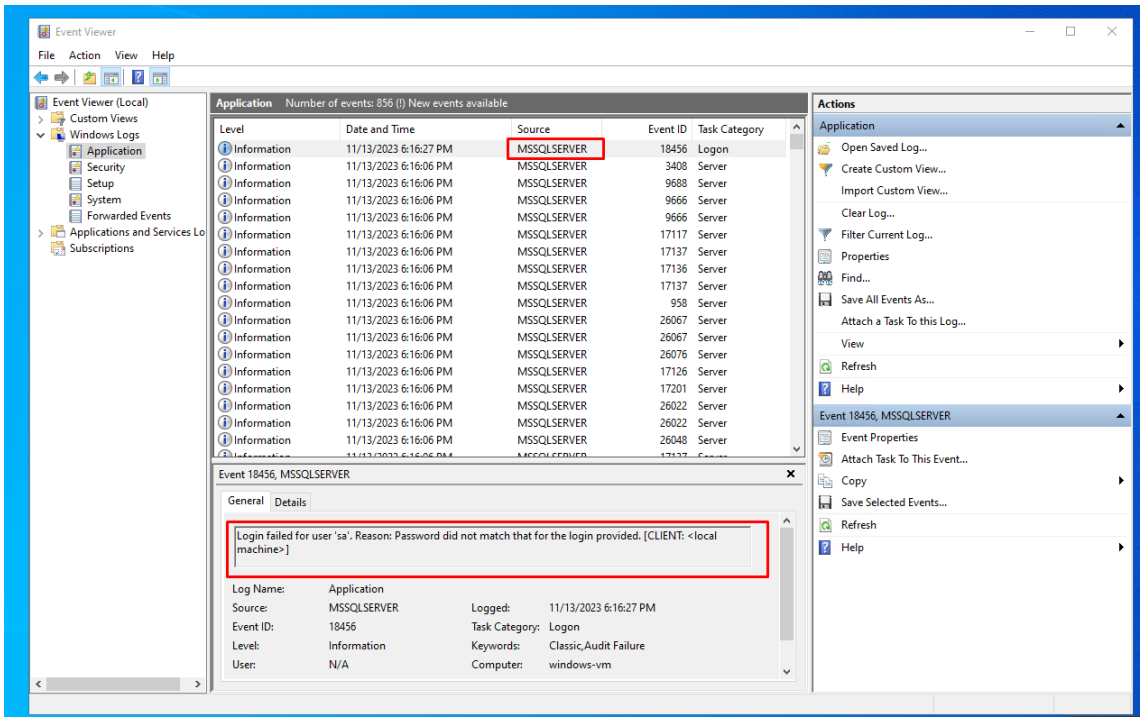
So that we can keep track of who is trying to login, etc.… Just so that we can use our SIEM later to practice incident response.

I will follow these steps:

https://learn.microsoft.com/en-us/sql/relational-databases/security/auditing/write-sql-server-audit-events-to-the-security-log?view=sql-server-ver16



After doing all that we want to make sure as well that we want to log both failed and successful login attempts.

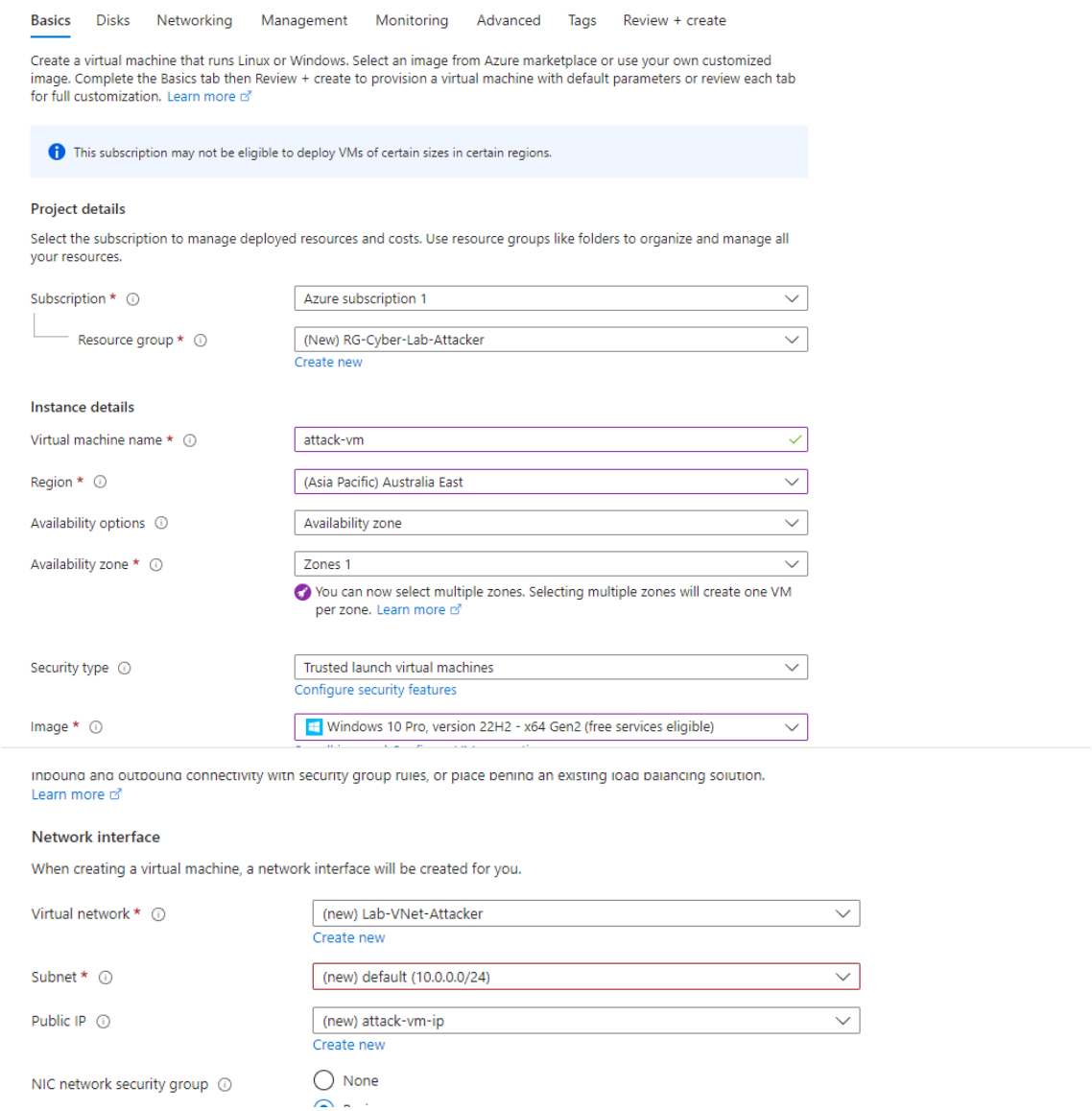


After trying a wrong password, we can indeed see that it works and out events are now being logged into the event viewer!

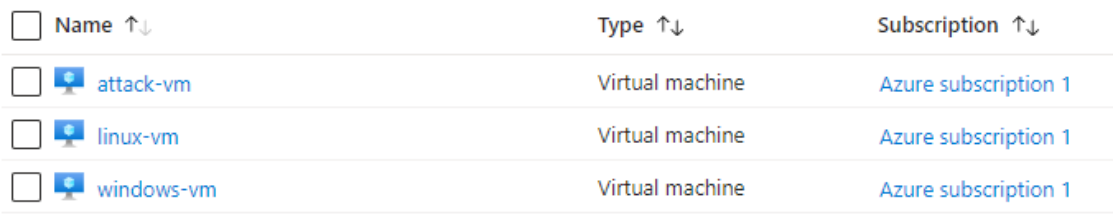
I will also test the Linux vm by logging into it and pinging it from the windows VM!

We will now proceed to create a third VM, this will be out ATTACK vm where we will try to connect to the SQL database over the internet as well as the SSH login on the linux machine.

And after the fact we will observer the logs.



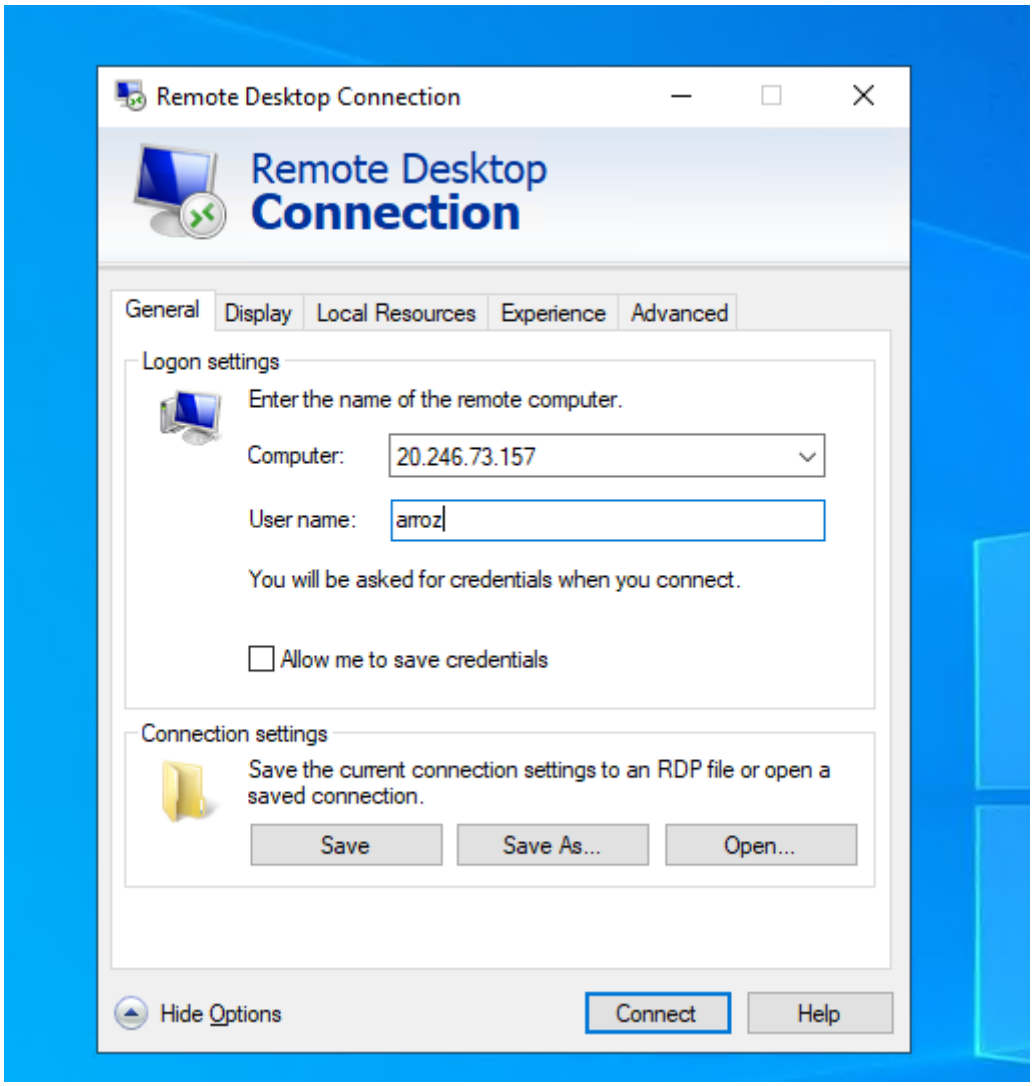
We will create another vnet so we can make sure they are segmented.



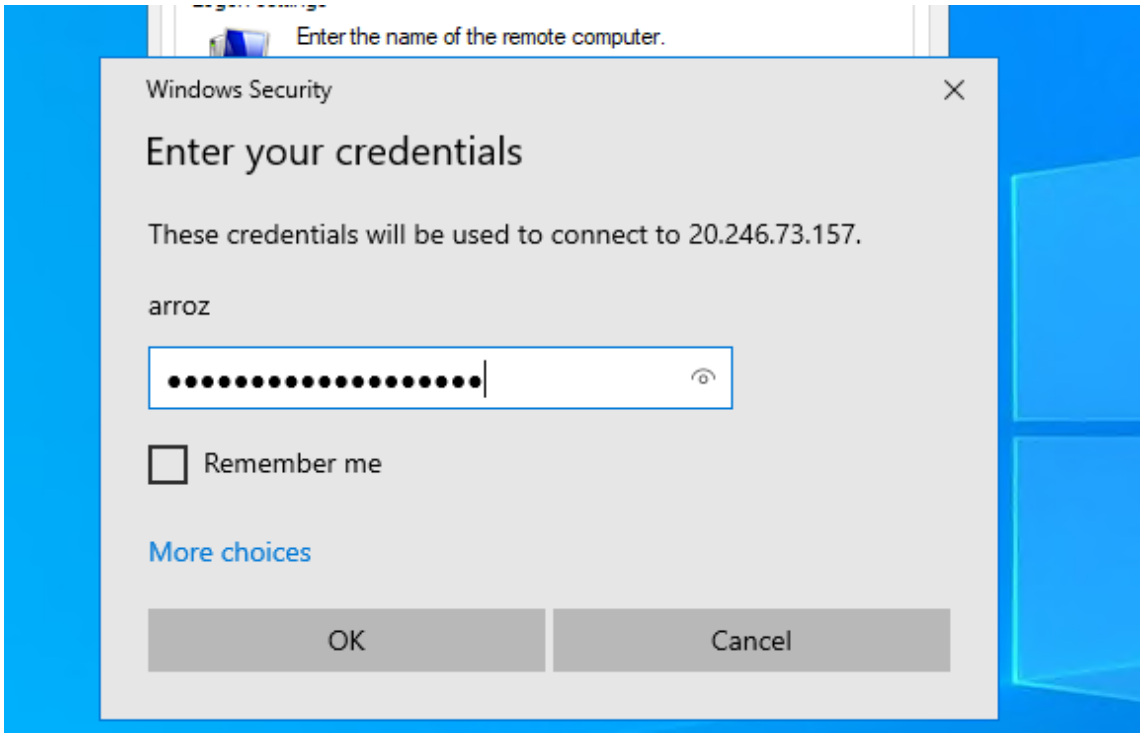
Here is what we have so far.

We will now login to the attack VM using RDP.

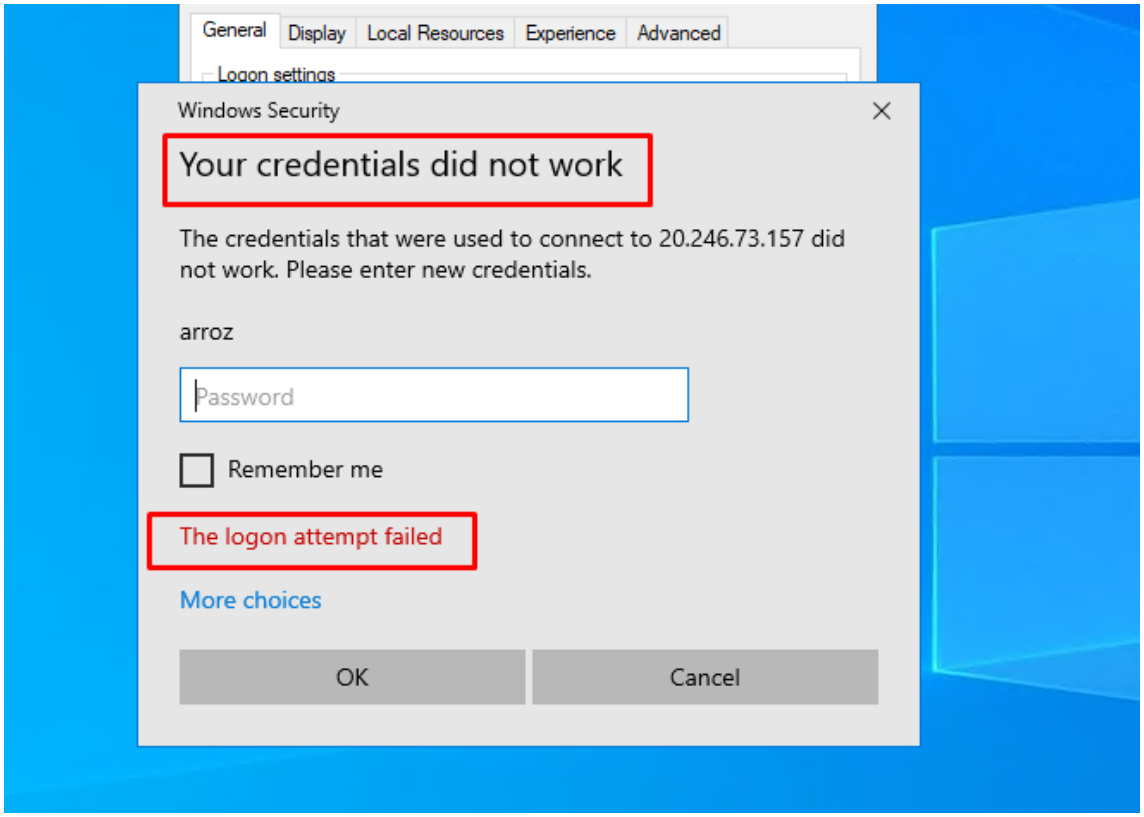
What will we do now is FROM WITHIN the ATTACK vm we will try and RDP into the windows VM (sql database) and input the wrong info on purpose to generate some logs.



So we will put a random username for example



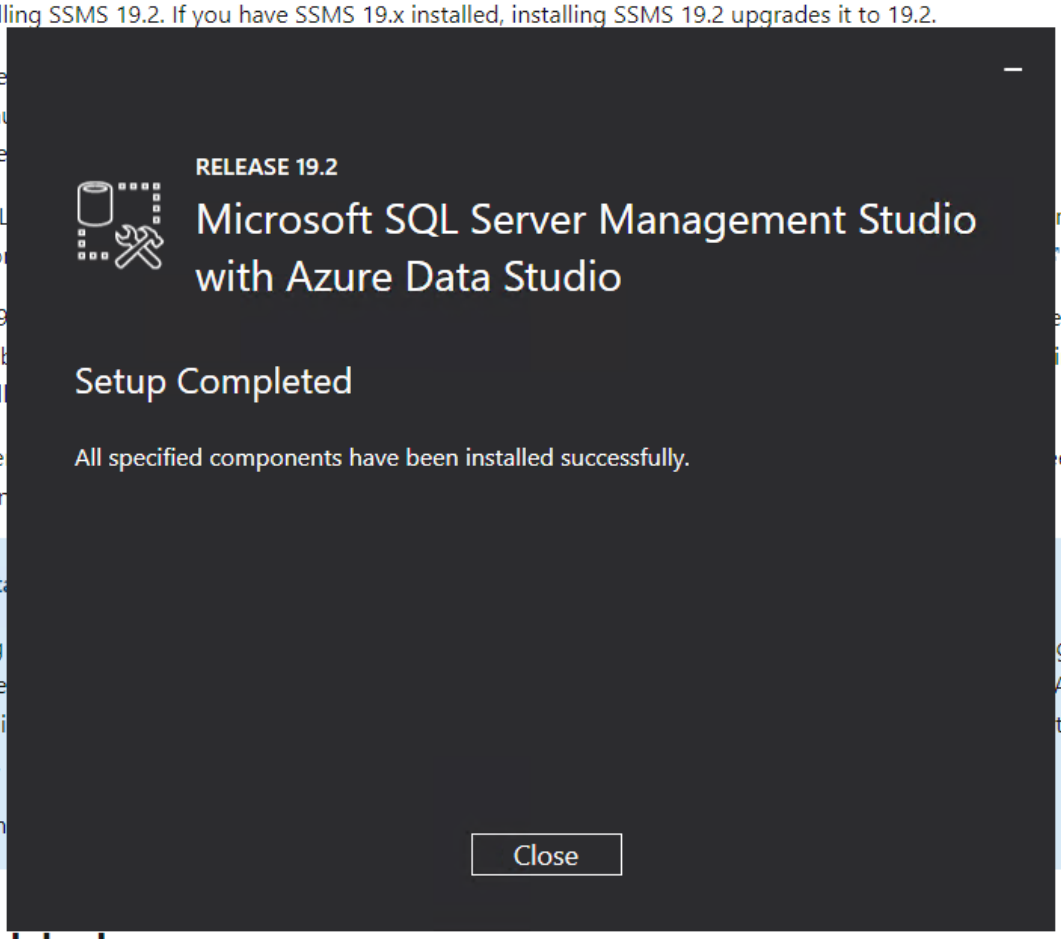
And random pass



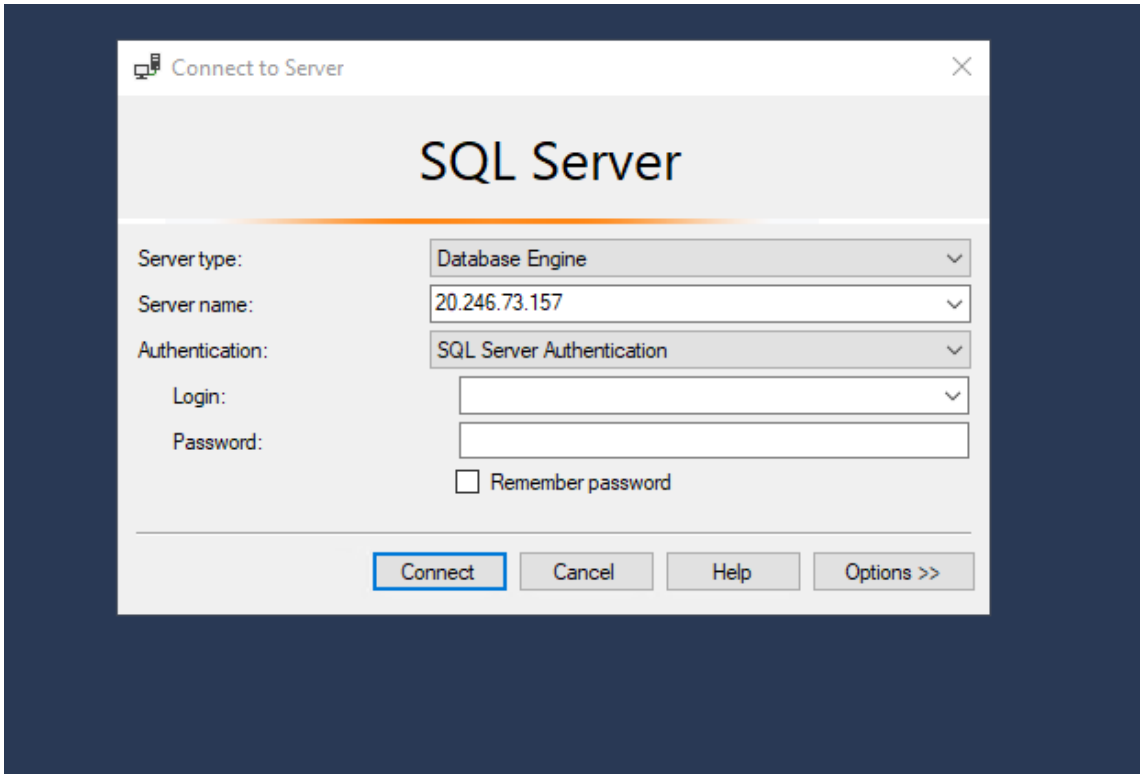
And this is what we want.

What we will do now is install SSMS and we will generate some SQL logs instead of the windows Logs. Let's generate some logs for that.

2 is the latest general availability (GA) version. If you have a *preview* version of SSMS 19 installed, uninstall i
stalling SSMS 19.2. If you have SSMS 19.x installed, installing SSMS 19.2 upgrades it to 19.2.



lable languages



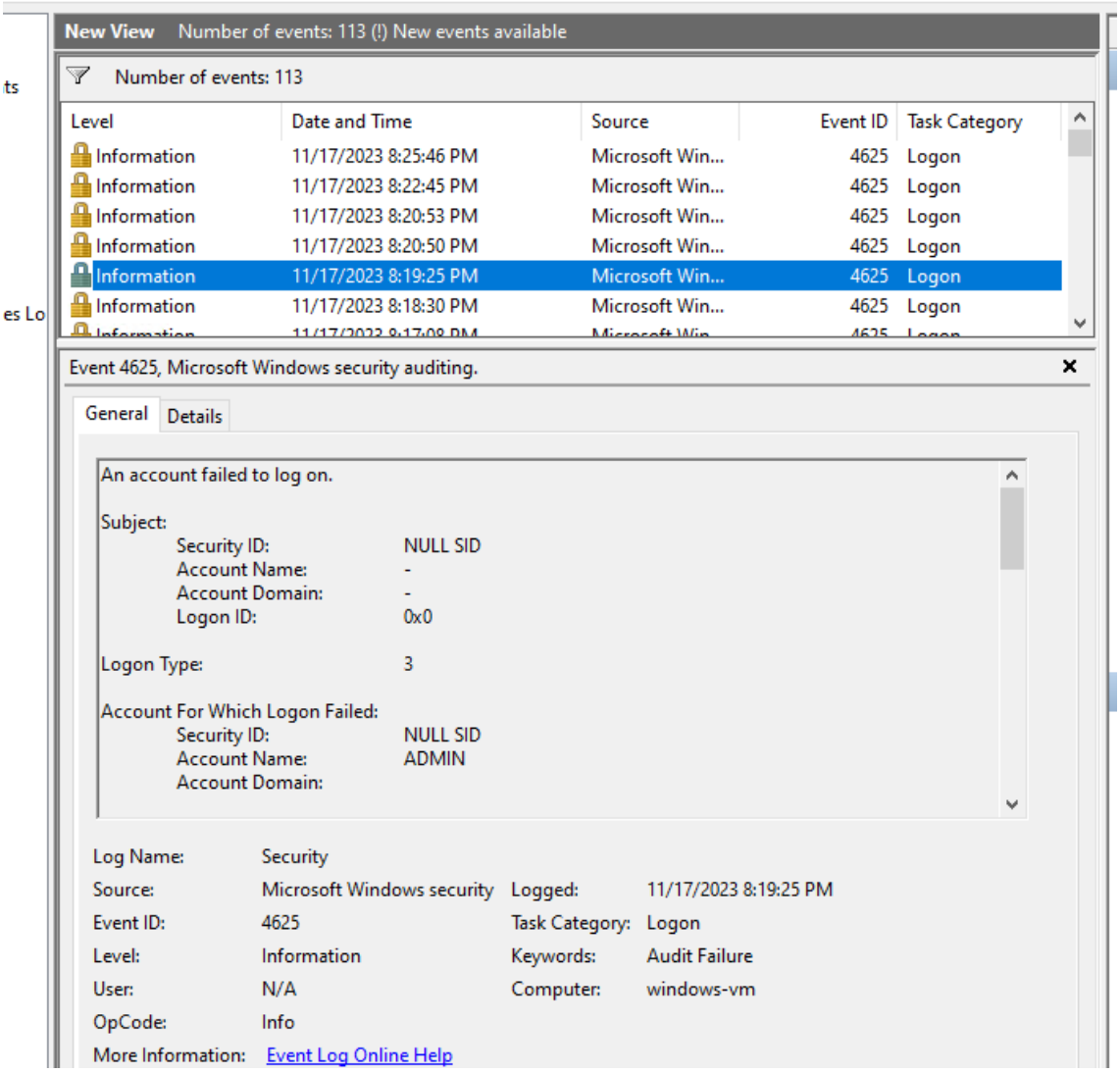
For server name we input the machine IP and we will now generate some failed logs.

Now we will try and SSH into the linux VM still using the ATTACK VM.

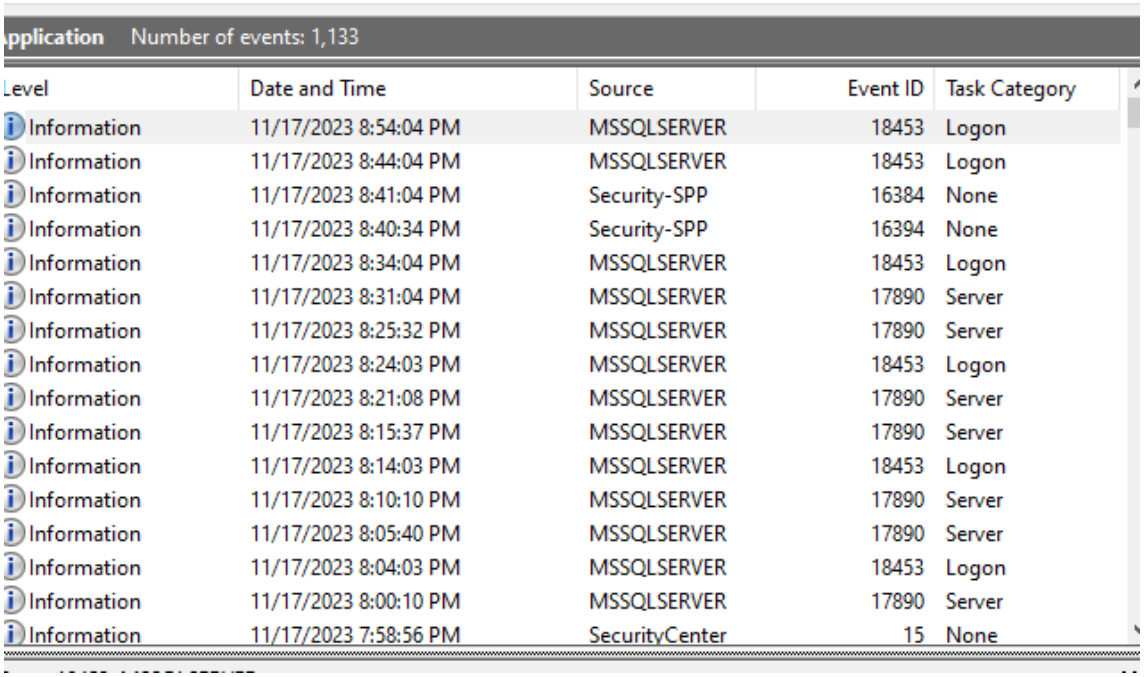
```
PS C:\Users\labuser> ssh josh@20.246.73.205
The authenticity of host '20.246.73.205 (20.246.73.205)' can't be established.
ECDSA key fingerprint is SHA256:oTvAzKVJoByZegATQcQUi8WD86LreR72HM8SCu+9aQc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '20.246.73.205' (ECDSA) to the list of known hosts.
josh@20.246.73.205's password:
Permission denied, please try again.
josh@20.246.73.205's password:
```

Just like this.

We will now take the perspective of an admin and login to the windows Vm to check the logs as well as the linux VM.



Let's just take a note of these.

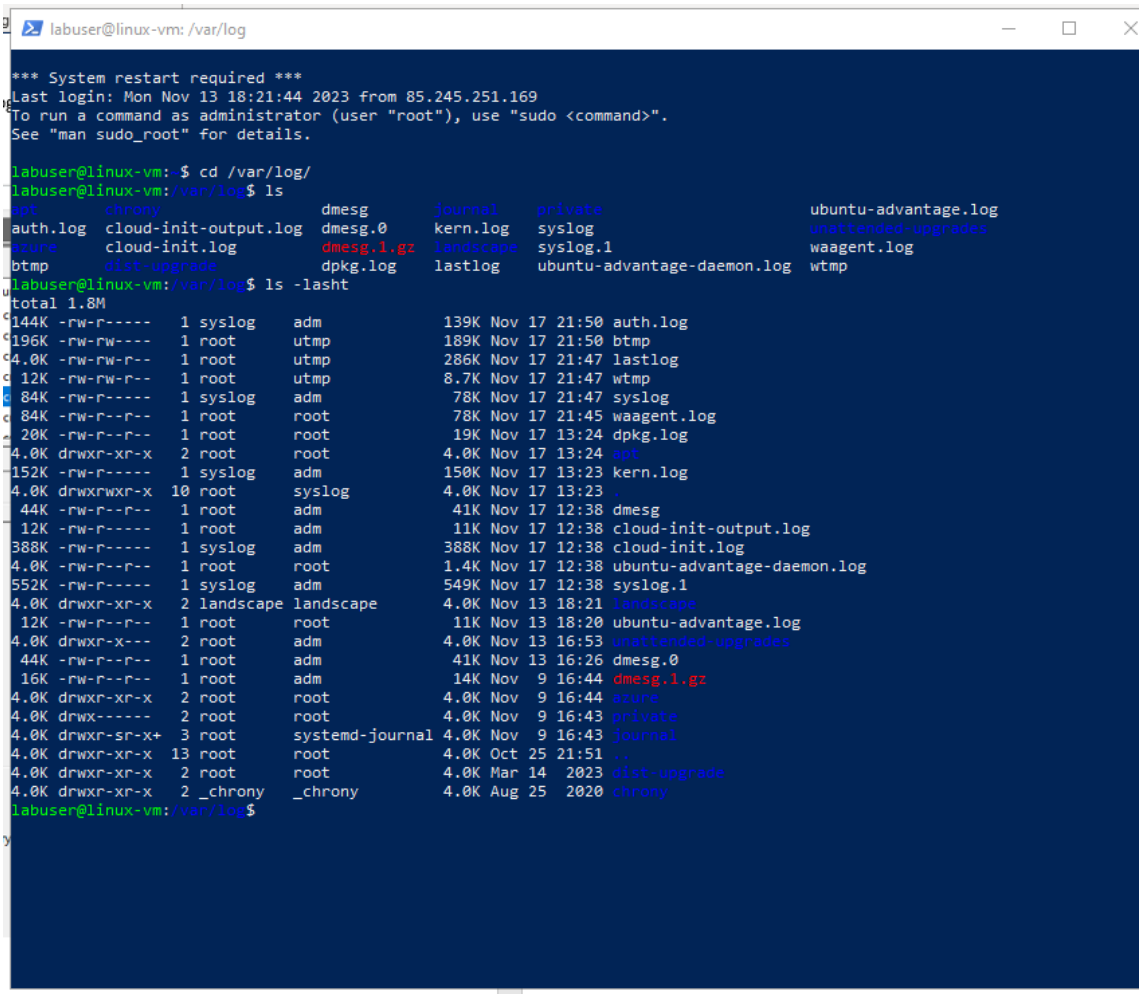


pplication Number of events: 1,133

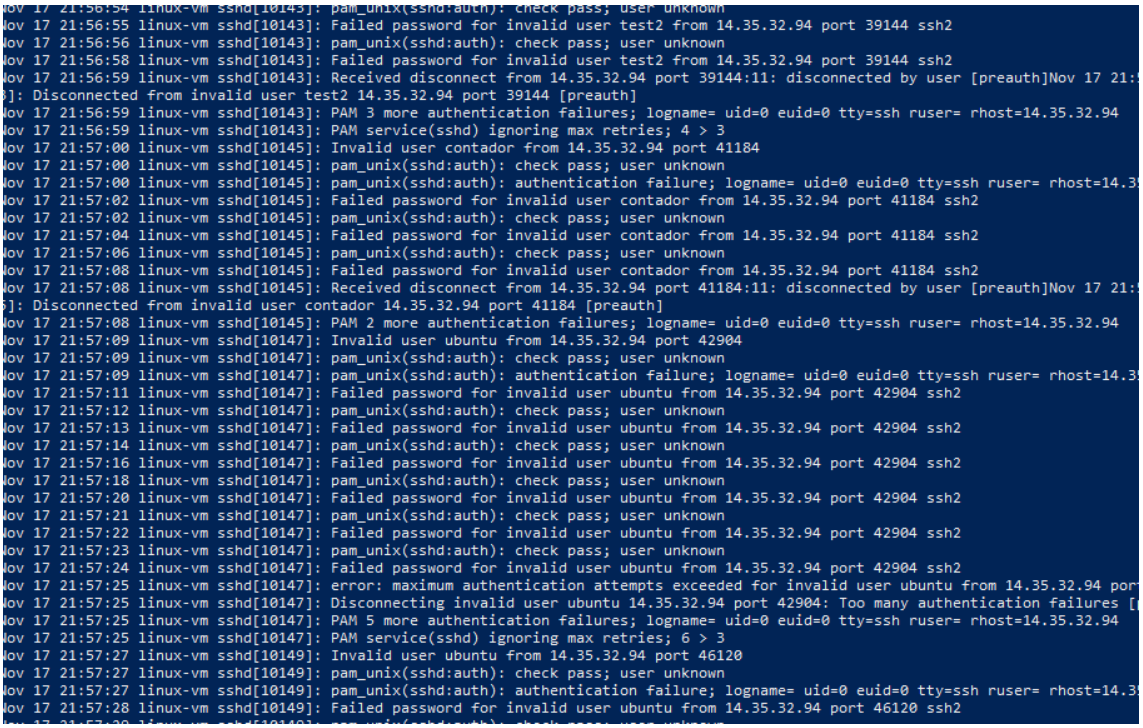
Level	Date and Time	Source	Event ID	Task Category
Information	11/17/2023 8:54:04 PM	MSSQLSERVER	18453	Logon
Information	11/17/2023 8:44:04 PM	MSSQLSERVER	18453	Logon
Information	11/17/2023 8:41:04 PM	Security-SPP	16384	None
Information	11/17/2023 8:40:34 PM	Security-SPP	16394	None
Information	11/17/2023 8:34:04 PM	MSSQLSERVER	18453	Logon
Information	11/17/2023 8:31:04 PM	MSSQLSERVER	17890	Server
Information	11/17/2023 8:25:32 PM	MSSQLSERVER	17890	Server
Information	11/17/2023 8:24:03 PM	MSSQLSERVER	18453	Logon
Information	11/17/2023 8:21:08 PM	MSSQLSERVER	17890	Server
Information	11/17/2023 8:15:37 PM	MSSQLSERVER	17890	Server
Information	11/17/2023 8:14:03 PM	MSSQLSERVER	18453	Logon
Information	11/17/2023 8:10:10 PM	MSSQLSERVER	17890	Server
Information	11/17/2023 8:05:40 PM	MSSQLSERVER	17890	Server
Information	11/17/2023 8:04:03 PM	MSSQLSERVER	18453	Logon
Information	11/17/2023 8:00:10 PM	MSSQLSERVER	17890	Server
Information	11/17/2023 7:58:56 PM	SecurityCenter	15	None

Here are the SQL logs.

Let’s take notes of a couple of things like Ips, event IDs and more.



And finally, here is the linux logs.



If we CAT the auth.logs we get the authorization logs from the machine.

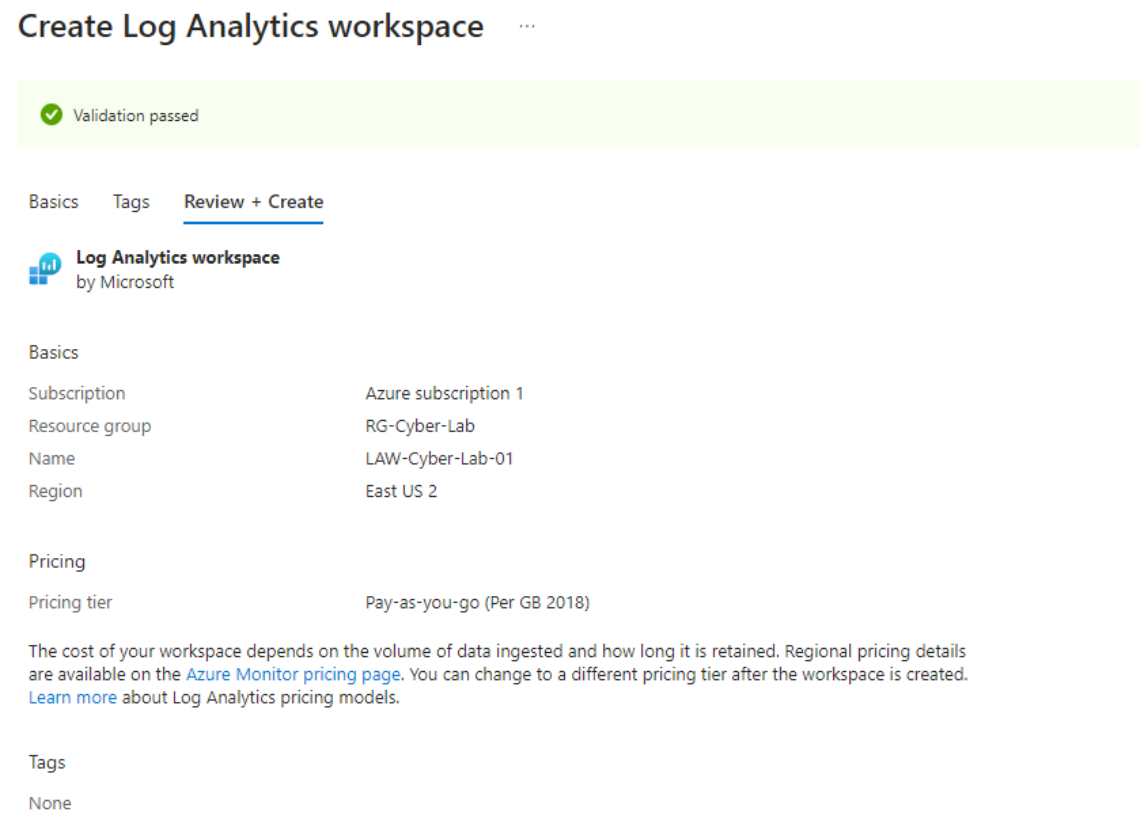
Log	Layer	Description
AAD logs	Azure Tenant	Contain the history of sign-in activity and audit trail of changes made in Azure AD for a particular tenant.
Activity log	Azure Subscription	Provides insight into the operations on each Azure resource in the subscription from the outside (the ***management plane***) in addition to updates on Service Health events. Use the Activity log to determine the *what*, *who*, and *when* for any write operations (PUT, POST, DELETE) taken on the resources in your subscription. There's a single activity log for each Azure subscription.
Resource logs	Azure Resources	Provide insight into operations that were performed within an Azure resource (the ***data plane***). Examples might be getting a secret from a key vault or making a request to a database. The content of resource logs varies by the Azure service and resource type. *Resource logs were previously referred to as diagnostic logs.*

Here is an overview on how logs worn in azure, I am showing this because we will work mostly on logging for right now.

We will start by downloading a file to our pc, this is a list with IPs and locations that we can use to map our attackers later on a map. Here is what I used:

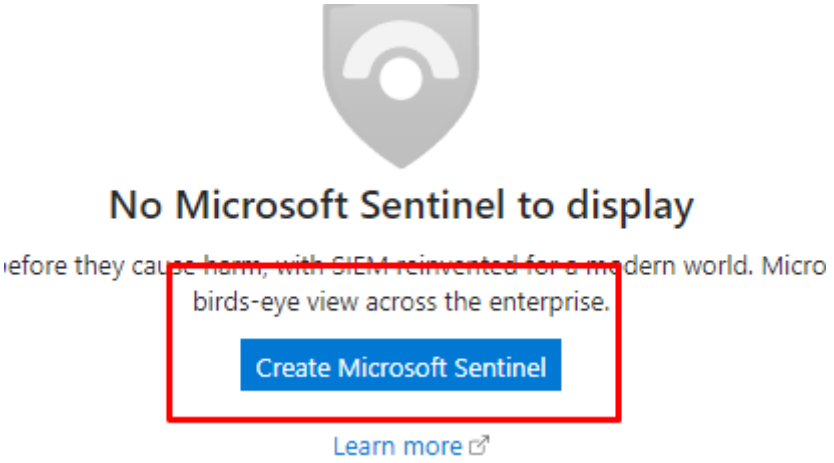
	A	B	C	D	E	F	G	H
1	network	latitude	longitude	cityname	countryname			
2	1.0.0.0/16	-33.494	143.2104		Australia			
3	1.1.0.0/16	17.8148	103.3386	Ban Chan	Thailand			
4	1.2.0.0/16	13.8667	100.1917	Nakhon P	Thailand			
5	1.3.0.0/16	13.8679	100.1891	Nakhon P	Thailand			
6	1.4.0.0/16	13.6687	100.579	Bangkok	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	1.10.0.0/1	17.8842	102.7394	Nong Khai	Thailand			
13	1.11.0.0/1	37.5841	127.0616	Dongdaen	South Korea			
14	1.12.0.0/1	37.5917	127.069	Dongdaen	South Korea			
15	1.13.0.0/1	37.5814	127.0663	Dongdaen	South Korea			
16	1.14.0.0/1	37.58	127.0609	Dongdaen	South Korea			
17	1.15.0.0/1	37.5786	127.0678	Dongdaen	South Korea			
18	1.16.0.0/1	37.5802	127.0644	Dongdaen	South Korea			
19	1.17.0.0/1	37.5838	127.0617	Dongdaen	South Korea			
20	1.18.0.0/1	37.5872	127.0651	Dongdaen	South Korea			
21	1.19.0.0/1	37.5757	127.0658	Dongdaen	South Korea			
22	1.20.0.0/1	13.5052	102.1872	Wang Nan	Thailand			
23	1.21.0.0/1	35.6384	139.6288	Kamiogi	Japan			
24	1.22.0.0/1	12.9634	77.5855	Bengaluru	India			
25	1.23.0.0/1	19.1963	72.9675	Thane	India			
26	1.24.0.0/1	19.1954	72.9646	Thane	India			
27	1.25.0.0/1	19.195	72.969	Thane	India			
28	1.26.0.0/1	19.1874	72.965	Thane	India			
29	1.27.0.0/1	19.2048	72.9588	Thane	India			
30	1.28.0.0/1	19.1879	72.9592	Thane	India			
31	1.29.0.0/1	19.2007	72.963	Thane	India			
32	1.30.0.0/1	19.2026	72.9631	Thane	India			
33	1.31.0.0/1	19.1989	72.9773	Thane	India			
34	1.32.0.0/1	3.0995	101.6054	Petaling J	Malaysia			
35	1.33.0.0/1	35.6887	139.745	Chiyoda-k	Japan			
36	1.34.0.0/1	24.9466	121.586	New Taipe	Taiwan			
37	1.35.0.0/1	24	121		Taiwan			

We will now create a Log analytics workspace (log aggregator)

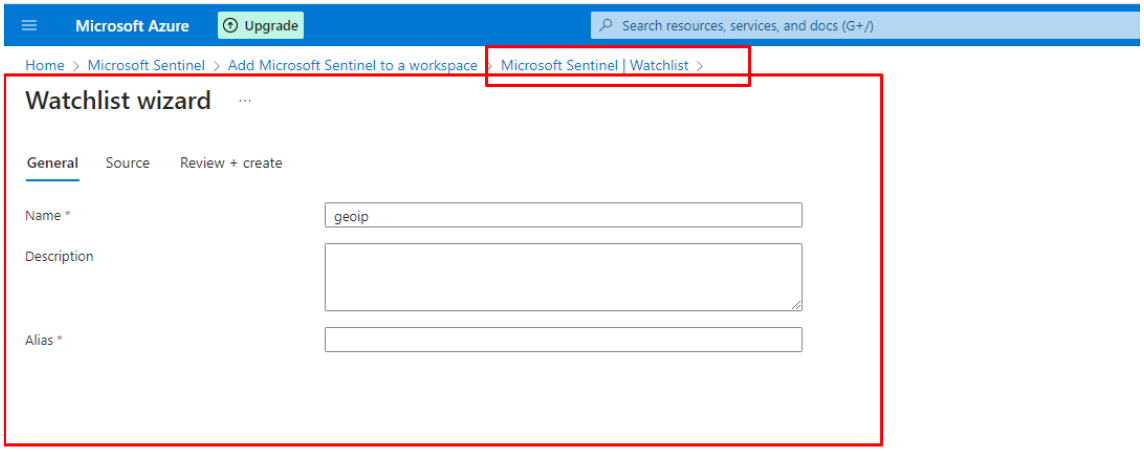


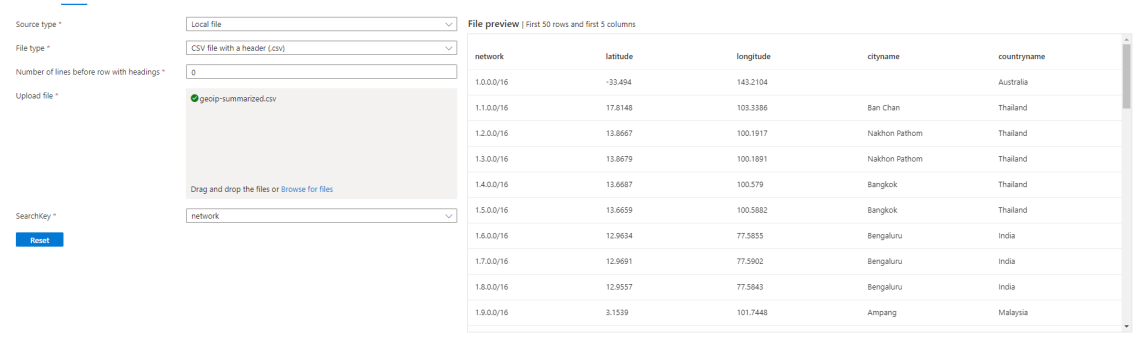
Here it is created.

We will now setup Microsoft Sentinel (SIEM) and we will connect it to out analytics workspace.

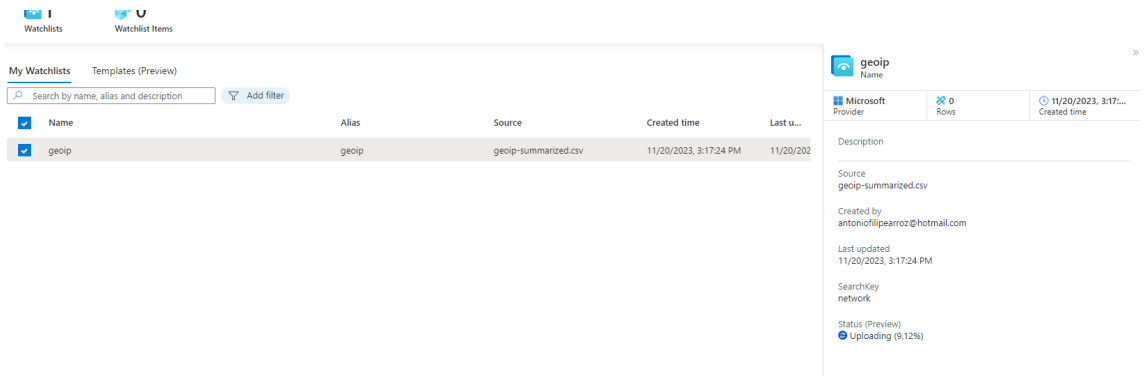


After sentinel is created, we will create the geoip watchlist.





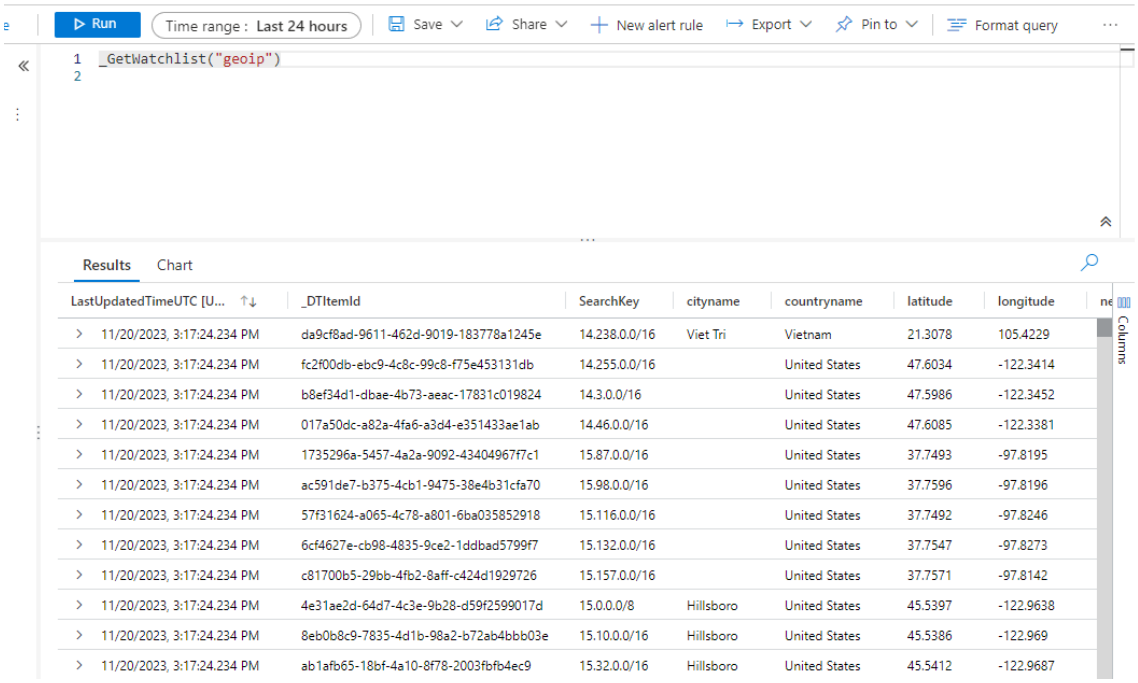
And here it is.



We now must wait for the csv to upload!

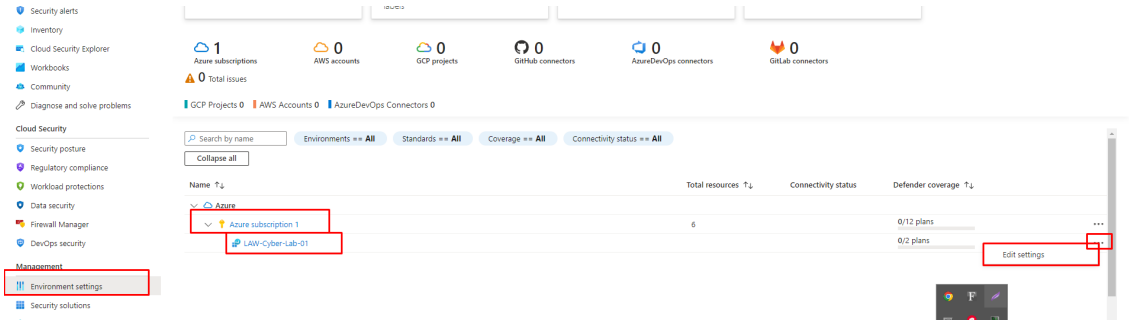
Next, we will go to our log analytics, and we will make sure something comes out when we query _GetWatchlist("geoip")

Something like this:



LastUpdatedTimeUTC [U...	_DTItemId	SearchKey	cityname	countryname	latitude	longitude
11/20/2023, 3:17:24.234 PM	da9cf8ad-9611-462d-9019-183778a1245e	14.238.0.0/16	Viet Tri	Vietnam	21.3078	105.4229
11/20/2023, 3:17:24.234 PM	fc2f00db-ebc9-4c8c-99c8-f75e453131db	14.255.0.0/16		United States	47.6034	-122.3414
11/20/2023, 3:17:24.234 PM	b8ef34d1-dbae-4b73-aeac-17831c019824	14.3.0.0/16		United States	47.5986	-122.3452
11/20/2023, 3:17:24.234 PM	017a50dc-a82a-4fa6-a3d4-e351433ae1ab	14.46.0.0/16		United States	47.6085	-122.3381
11/20/2023, 3:17:24.234 PM	1735296a-5457-4a2a-9092-43404967f7c1	15.87.0.0/16		United States	37.7493	-97.8195
11/20/2023, 3:17:24.234 PM	ac591de7-b375-4cb1-9475-38e4b31cfa70	15.98.0.0/16		United States	37.7596	-97.8196
11/20/2023, 3:17:24.234 PM	57f31624-a065-4c78-a801-6ba035852918	15.116.0.0/16		United States	37.7492	-97.8246
11/20/2023, 3:17:24.234 PM	6cf4627e-cb98-4835-9ce2-1ddbad5799f7	15.132.0.0/16		United States	37.7547	-97.8273
11/20/2023, 3:17:24.234 PM	c81700b5-29bb-4fb2-8aff-c424d1929726	15.157.0.0/16		United States	37.7571	-97.8142
11/20/2023, 3:17:24.234 PM	4e31ae2d-64d7-4c3e-9b28-d59f2599017d	15.0.0.0/8	Hillsboro	United States	45.5397	-122.9638
11/20/2023, 3:17:24.234 PM	8eb0b8c9-7835-4d1b-98a2-b72ab4bbb03e	15.10.0.0/16	Hillsboro	United States	45.5386	-122.969
11/20/2023, 3:17:24.234 PM	ab1afb65-18bf-4a10-8f78-2003fbfb4ec9	15.32.0.0/16	Hillsboro	United States	45.5412	-122.9687

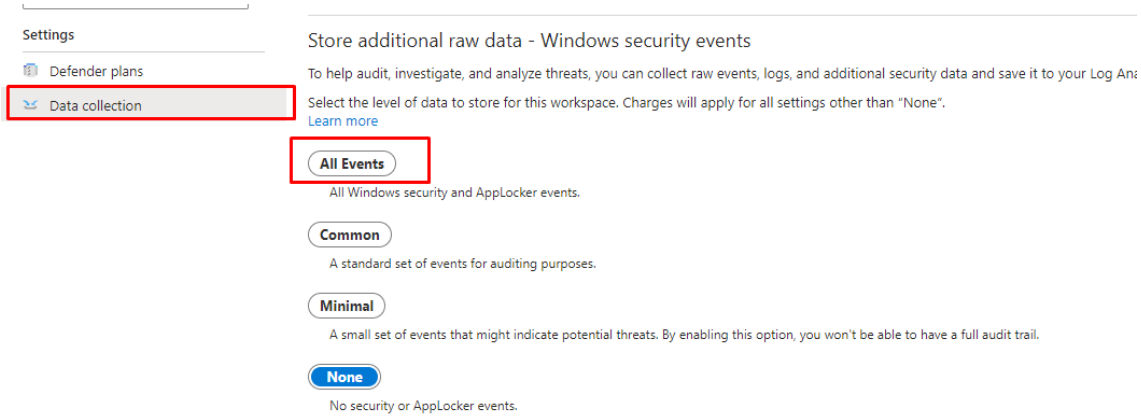
Next, we will enable Microsoft defender for cloud, it allows us to take logs from virtual machines and setwork security groups and ingest them in out log analytics workspace!



We will go here after getting to the azure page.

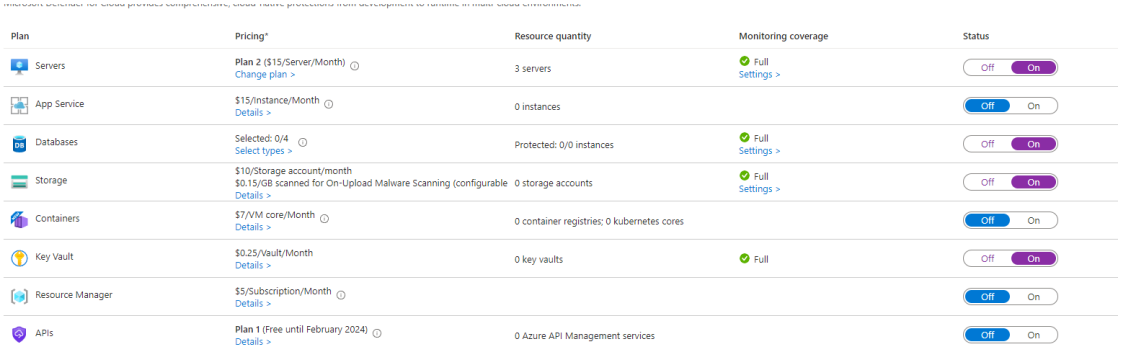
Plan	Pricing*	Resource quantity	Plan
Foundational CSPM	Free		Off On
Servers	$15/Server/Month	0 servers	Off On
SQL servers on machines	$15/Server/Month $0.015/Core/Hour	0 servers	Off On

We need these 2 on!

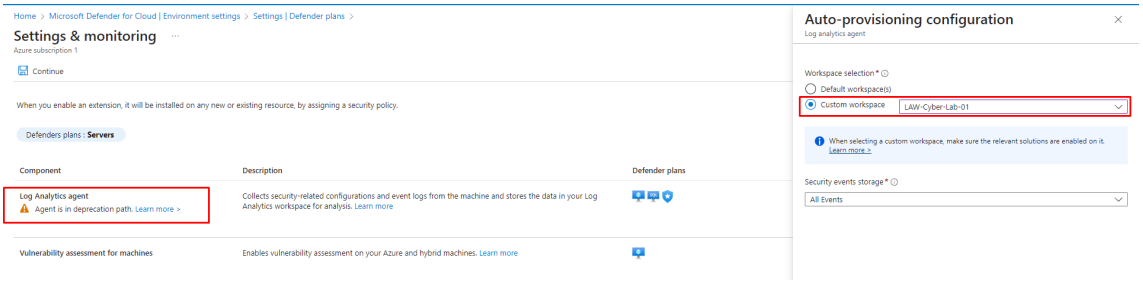


We also want to click on All events on this TAB.

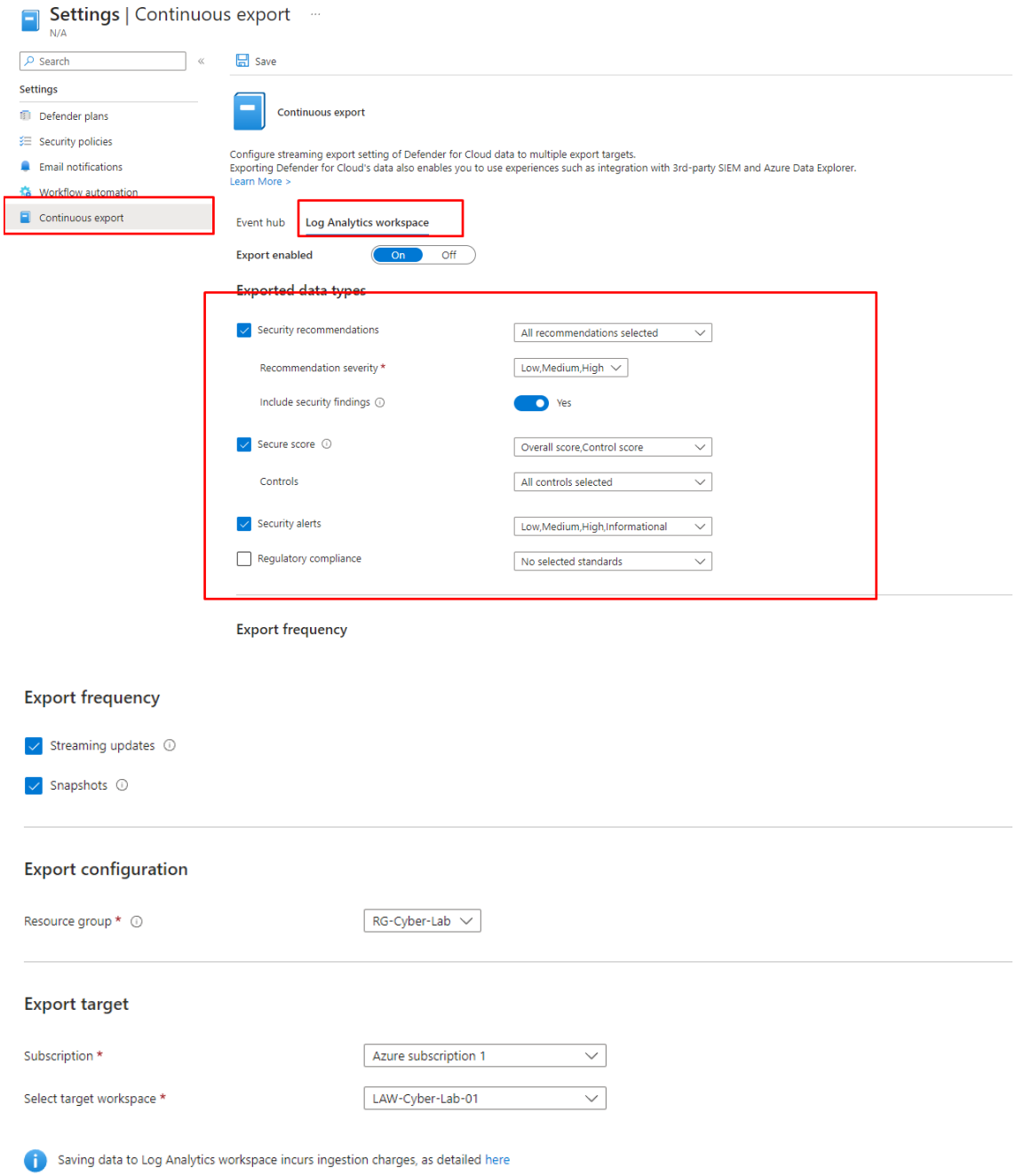
We also want to make sure it is set for the subscription as well!



We want to make sure all of these are turned on because we will be creating some more resources like a key vault in the future.

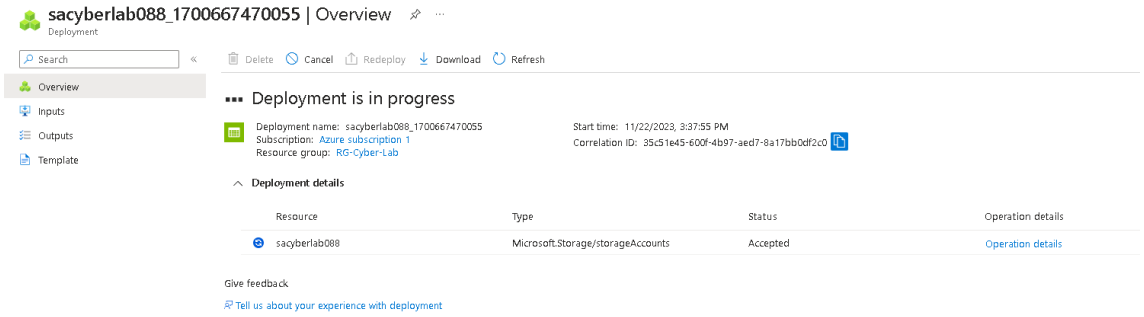


We also want to click settings under the servers and select for the logs to be sent to our custom log aggregation space and not a default one.

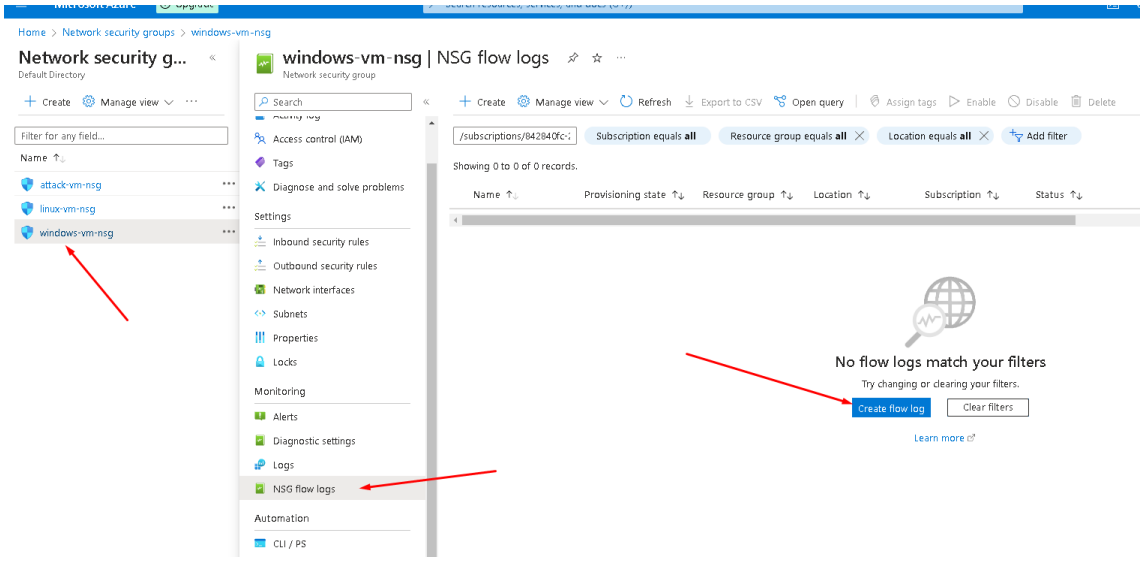


Next, we will send the logs using continuous export like shown above.

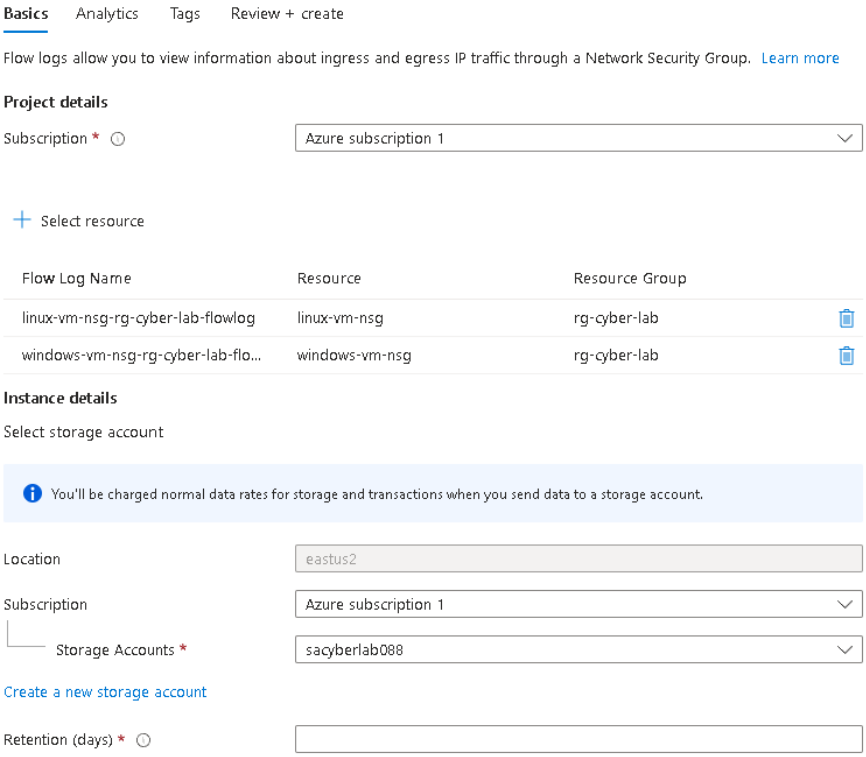
We will now create an azure storage account, it must be in the same region as the VMs, this will be used to store the NSG flow logs which we will create shortly!



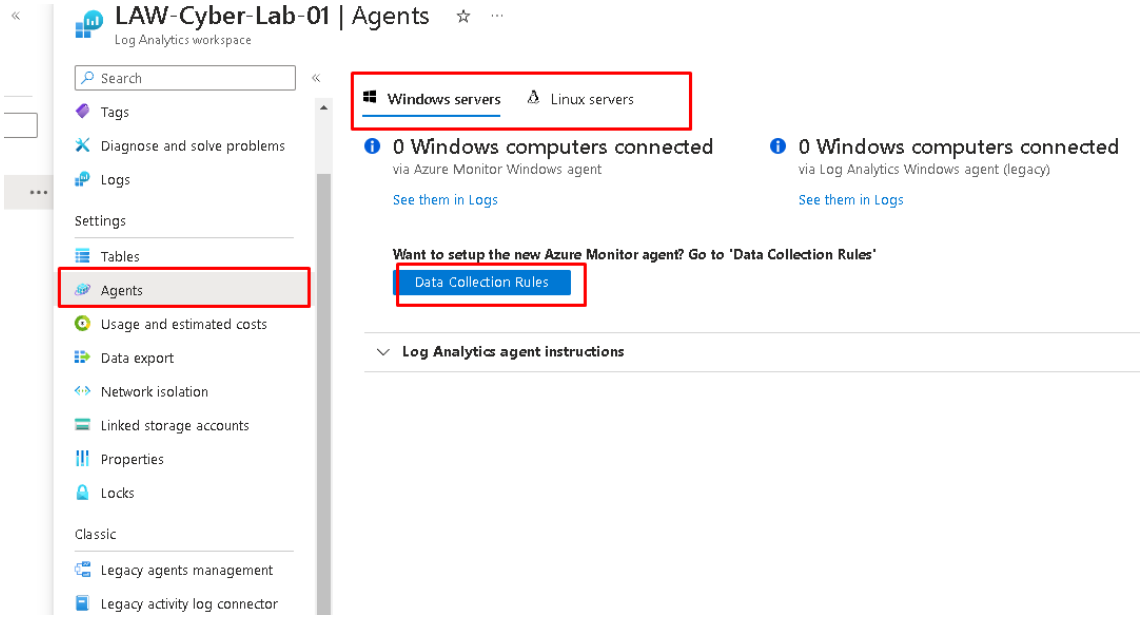
Created, next we will enable flow logs for both NSGs.

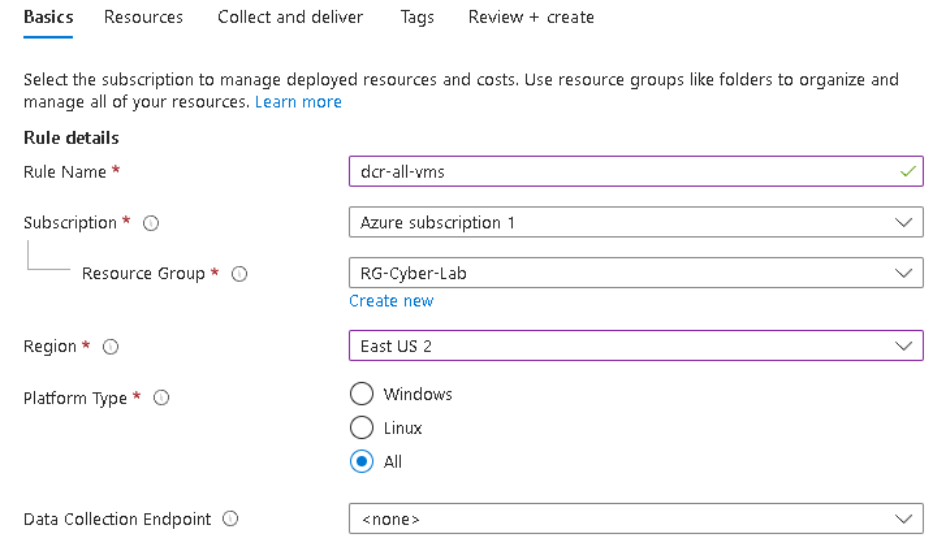


This is how we create the flow logs (:

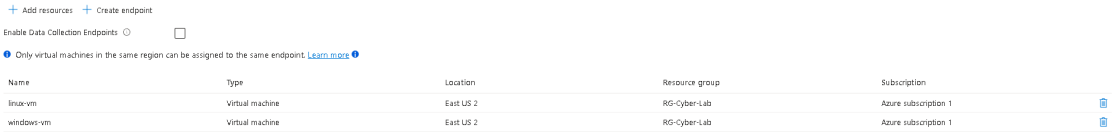


We will now configure the data collection rules within our log analytics workspace.

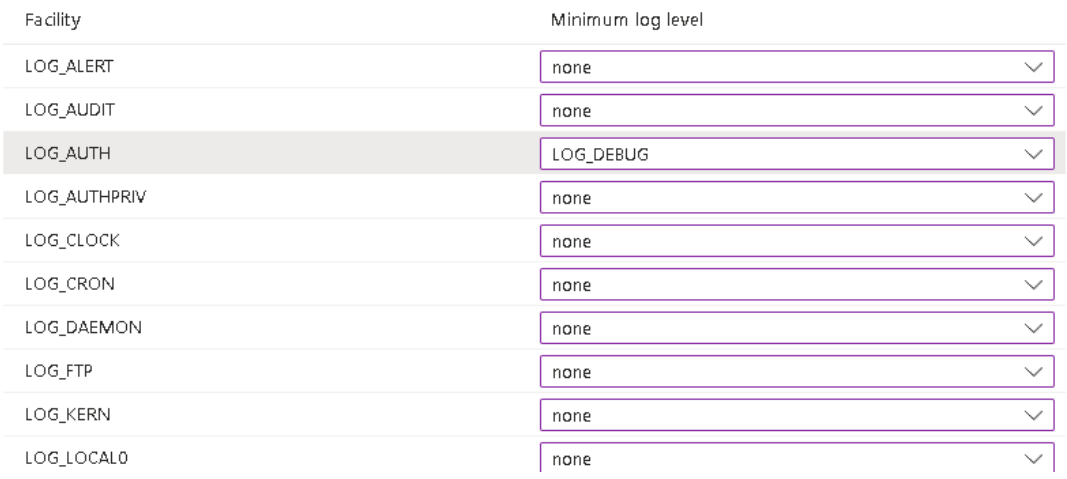




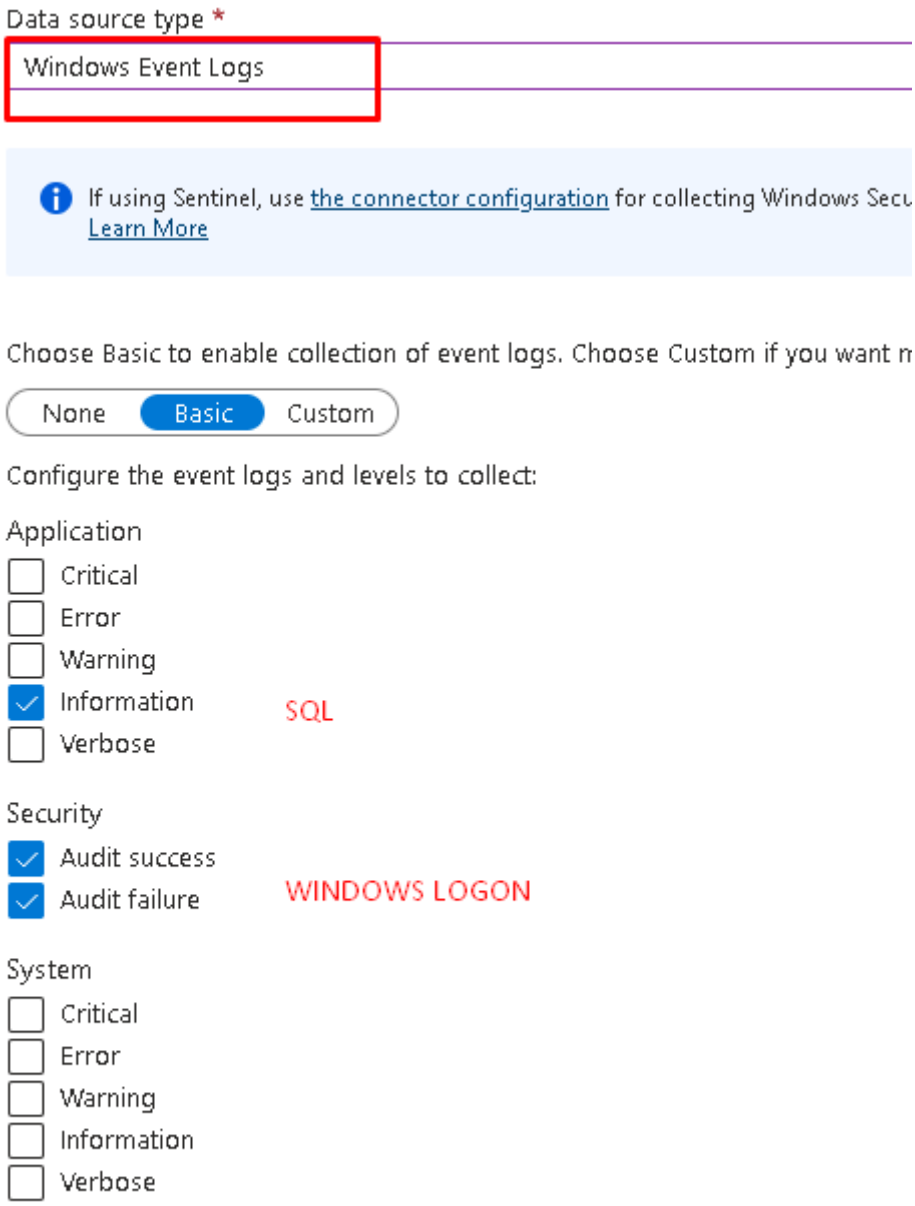
this will be the data collection rule for all our VMs so here is what we input.



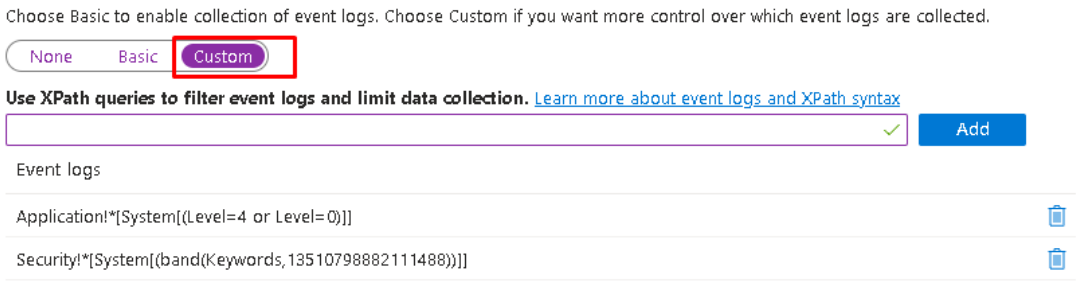
We will add both vms here.



For this lab we will only collect AUTH logs so rest should be set to none.



For windows we only need these.



We will actually set this to custom after creating and changing it to custom since we also want to record logs of people messing with the windows defender and more and we need to input 2 more lines here, here they are:

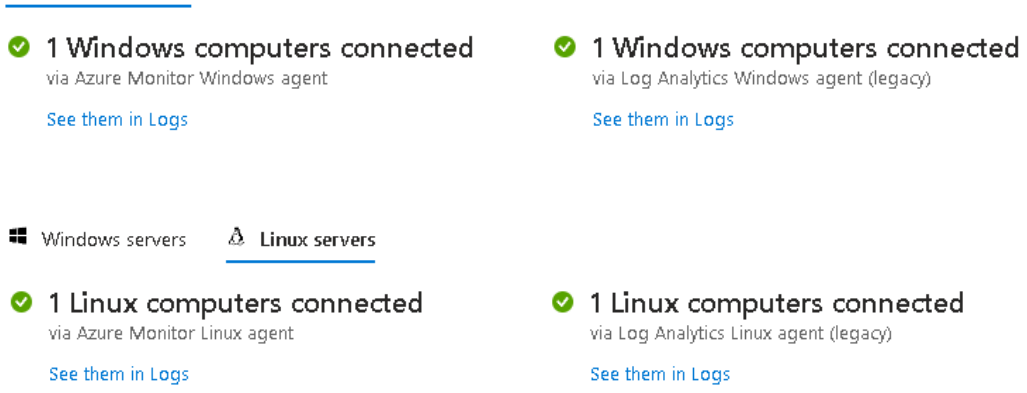
One of these is for malware and the other it for tempering detection!

// Windows Defender Malware Detection XPath Query

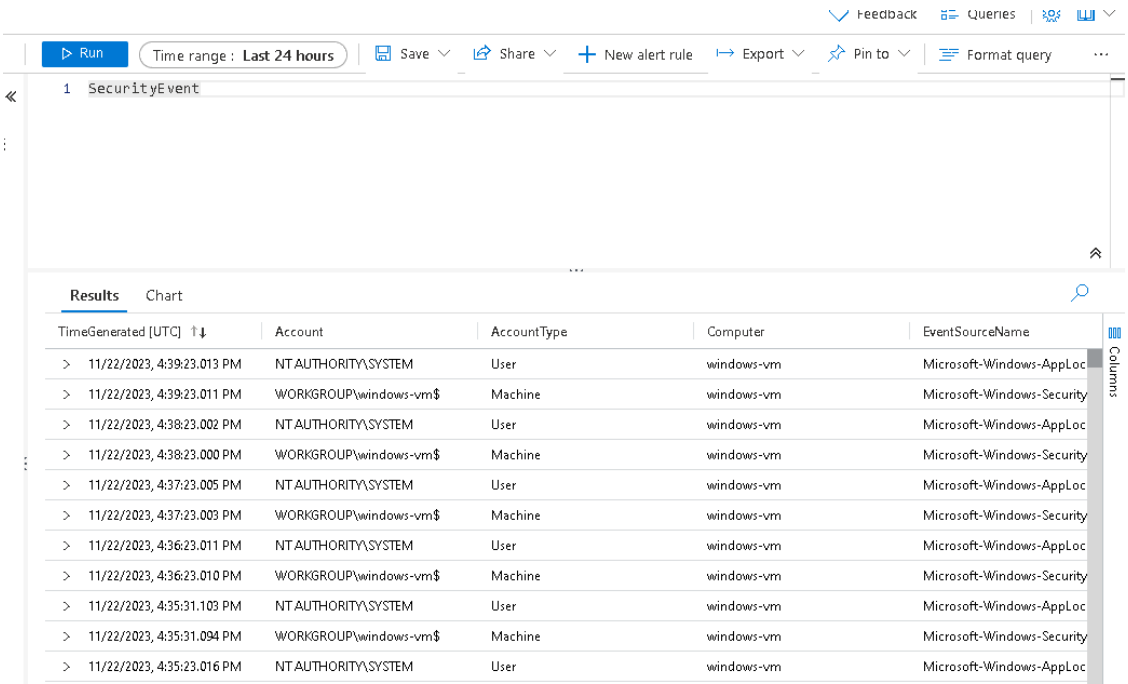
Microsoft-Windows-Windows Defender/Operational!*[System[(EventID=1116 or EventID=1117)]]

// Windows Firewall Tampering Detection XPath Query

Microsoft-Windows-Windows Firewall With Advanced Security/Firewall!*[System[(EventID=2003)]]



And now both machines should be sending logs!

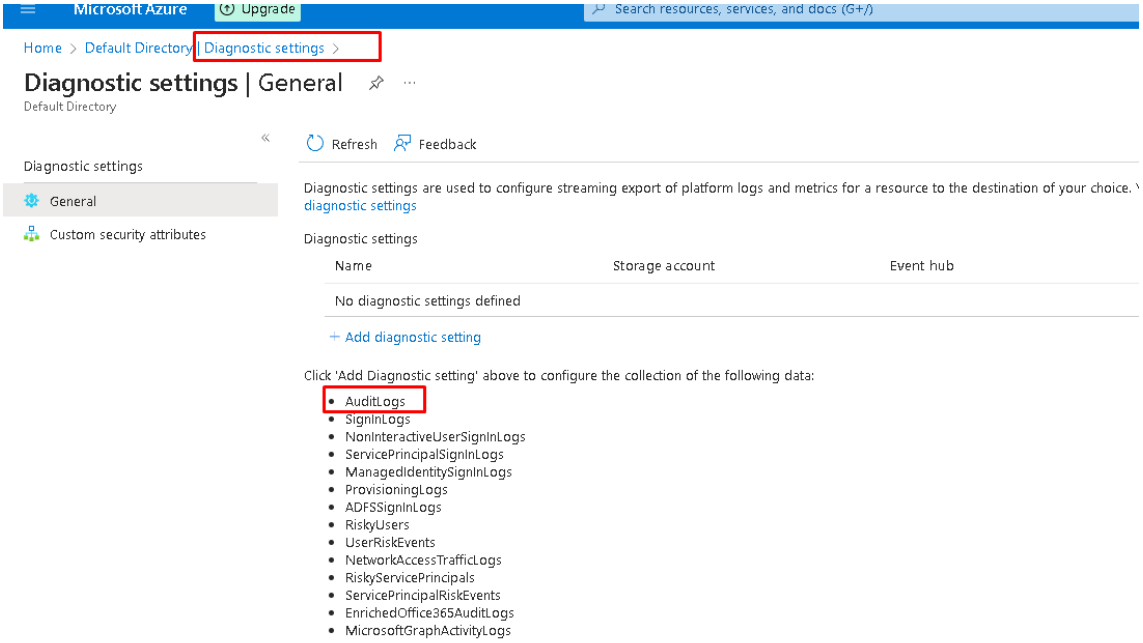


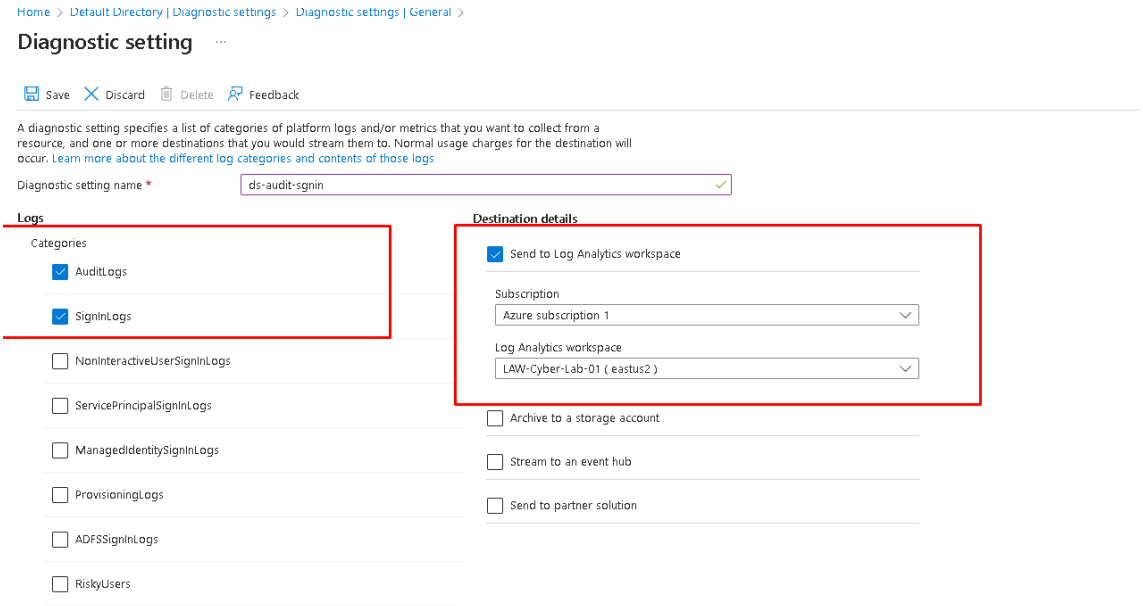
TimeGenerated [UTC]	Account	AccountType	Computer	EventSourceName
11/22/2023, 4:39:23.013 PM	NT AUTHORITY\SYSTEM	User	windows-vm	Microsoft-Windows-AppLoc
11/22/2023, 4:39:23.011 PM	WORKGROUP\windows-vm$	Machine	windows-vm	Microsoft-Windows-Security
11/22/2023, 4:38:23.002 PM	NT AUTHORITY\SYSTEM	User	windows-vm	Microsoft-Windows-AppLoc
11/22/2023, 4:38:23.000 PM	WORKGROUP\windows-vm$	Machine	windows-vm	Microsoft-Windows-Security
11/22/2023, 4:37:23.005 PM	NT AUTHORITY\SYSTEM	User	windows-vm	Microsoft-Windows-AppLoc
11/22/2023, 4:37:23.003 PM	WORKGROUP\windows-vm$	Machine	windows-vm	Microsoft-Windows-Security
11/22/2023, 4:36:23.011 PM	NT AUTHORITY\SYSTEM	User	windows-vm	Microsoft-Windows-AppLoc
11/22/2023, 4:36:23.010 PM	WORKGROUP\windows-vm$	Machine	windows-vm	Microsoft-Windows-Security
11/22/2023, 4:35:31.103 PM	NT AUTHORITY\SYSTEM	User	windows-vm	Microsoft-Windows-AppLoc
11/22/2023, 4:35:31.094 PM	WORKGROUP\windows-vm$	Machine	windows-vm	Microsoft-Windows-Security
11/22/2023, 4:35:23.016 PM	NT AUTHORITY\SYSTEM	User	windows-vm	Microsoft-Windows-AppLoc

And just to make sure we can query some logs and we can see that they are actually being forwarded!

Now that we have logs from out 2 VMs we will start ingesting logs from our Microsoft active directory as well!

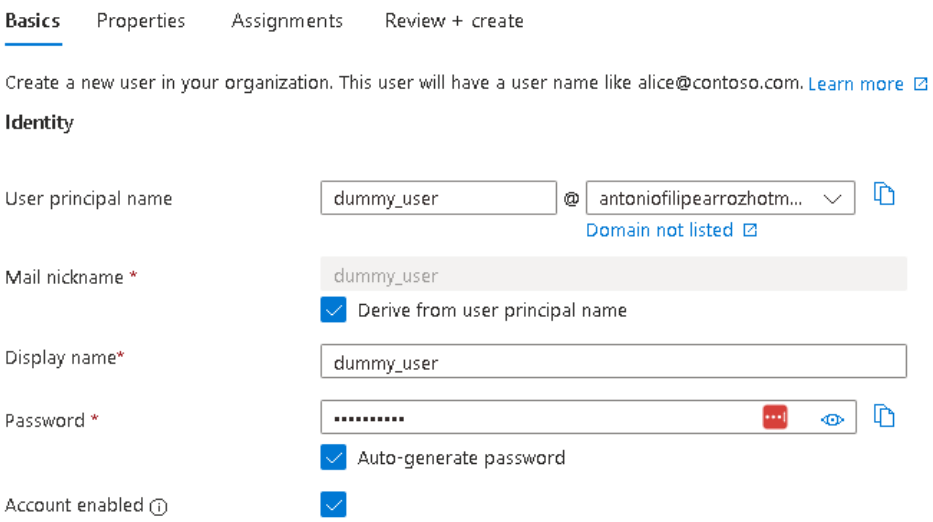
Let's start by:



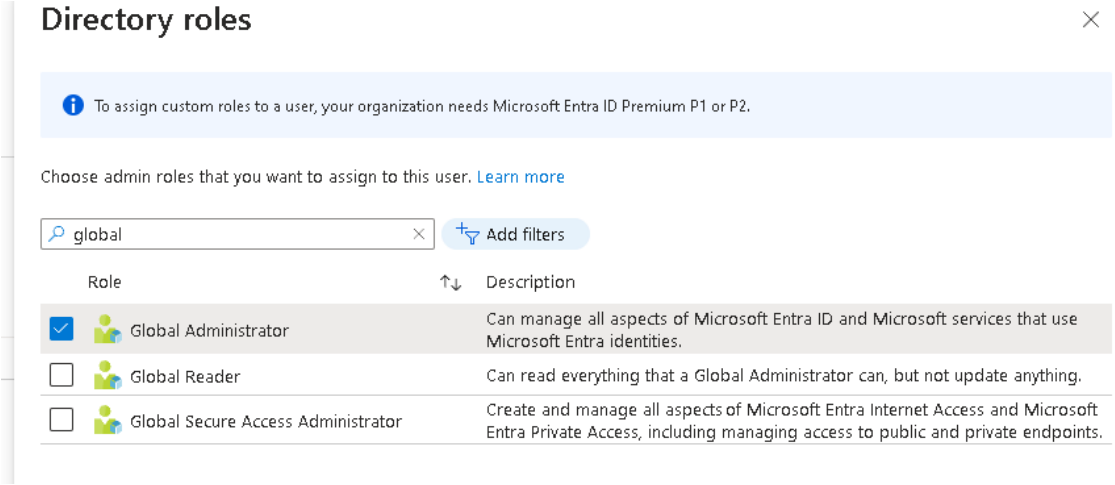


And this is all we need.

We will now create a dummy user in active directory and see if we can generate some logs!



We logged it to this user using a private tab and this should already generate a log, we will now assign the role of global admin to this user and this should generate another log as well.



We will now delete the dummy user and this should generate one more log.

And Boom,

```
1  AuditLogs
2
```

Results | Chart

TimeGenerated [UTC]	ResourceId	OperationName	OperationVersion	Cat
11/23/2023, 2:36:00.706 PM	/tenants/e50924e0-82a7-4a71-93cd-6d6af0c55bc8/providers...	Change password (self-service)	1.0	Use
11/23/2023, 2:36:00.688 PM	/tenants/e50924e0-82a7-4a71-93cd-6d6af0c55bc8/providers...	Update StsRefreshTokenValidFrom Timestamp	1.0	Use
11/23/2023, 2:36:00.687 PM	/tenants/e50924e0-82a7-4a71-93cd-6d6af0c55bc8/providers...	Change user password	1.0	Use
11/23/2023, 2:34:59.282 PM	/tenants/e50924e0-82a7-4a71-93cd-6d6af0c55bc8/providers...	Add user	1.0	Use
11/23/2023, 2:33:53.194 PM	/tenants/e50924e0-82a7-4a71-93cd-6d6af0c55bc8/providers...	Delete user	1.0	Use
11/23/2023, 2:33:53.193 PM	/tenants/e50924e0-82a7-4a71-93cd-6d6af0c55bc8/providers...	Delete user	1.0	Use
11/23/2023, 2:33:53.185 PM	/tenants/e50924e0-82a7-4a71-93cd-6d6af0c55bc8/providers...	Delete user	1.0	Use

Columns

Many logs!

Adding user, deleting user, everything is in there and more should come in the next few minutes.

Run | Time range : Last 48 hours | Save | Share | New alert rule | Export | Pin to | Format query

```
1  AuditLogs
2  | where OperationName == "Add member to role" and Result == "success"
3  | where TargetResources[0].modifiedProperties[1].newValue == '"Global Administrator"' or TargetResources[0].
   modifiedProperties[1].newValue == '"Company Administrator"'
4  | order by TimeGenerated desc
5  | project TimeGenerated, OperationName, AssignedRole = TargetResources[0].modifiedProperties[1].newValue, Status =
   Result, TargetResources
6
```

Results | Chart

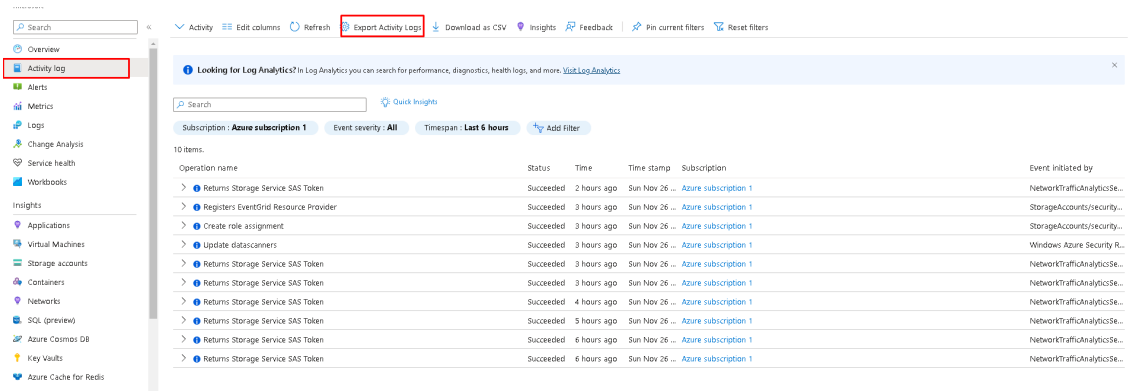
TimeGenerated [UTC]	OperationName	AssignedRole	Status	TargetResources
11/23/2023, 2:38:06.143 PM	Add member to role	"Global Administrator"	success	[{"id":"25163d24-79c5-44bf-b72

Columns

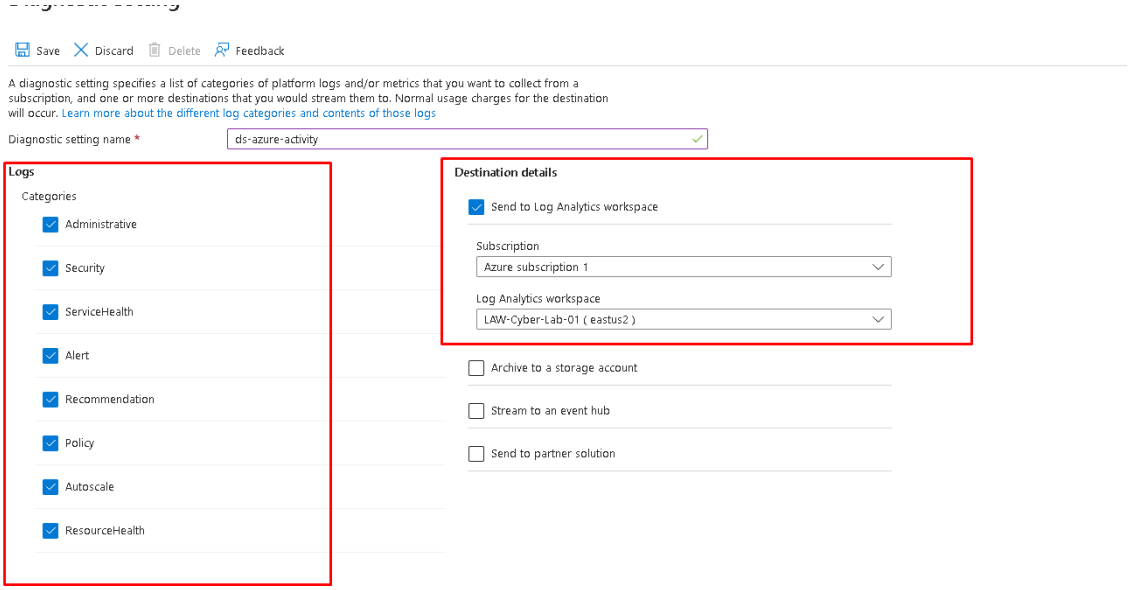
This query searches for adding someone as a global admin and like we expected we get a result!

All we need now for logging is subscription level logging, let's start with activity logs!

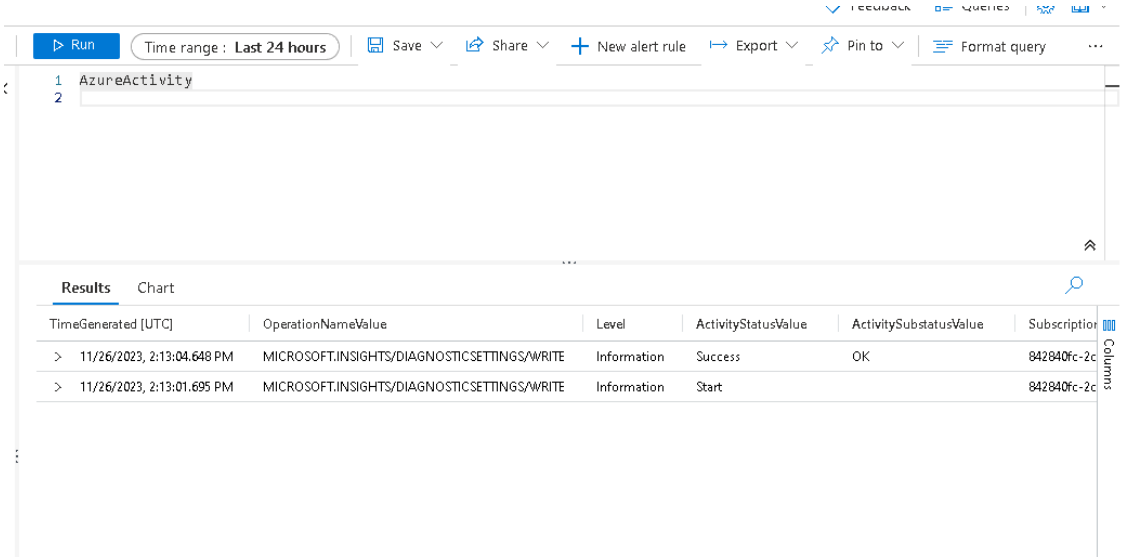
Like creating resources, deleting resources, etc.…



We first went to azure monitor, after that we need to click activity log and we will export those logs.

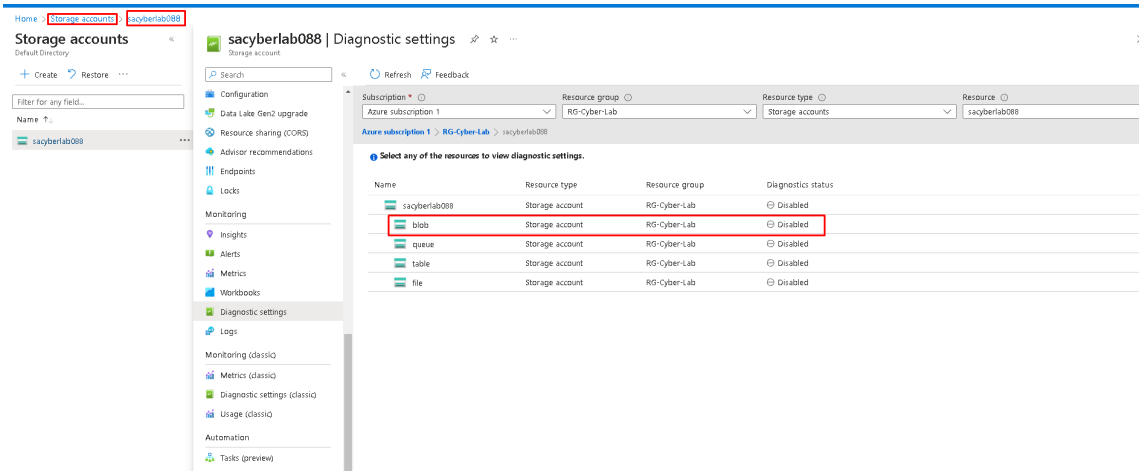


We will send everything to out log analytics workspace!

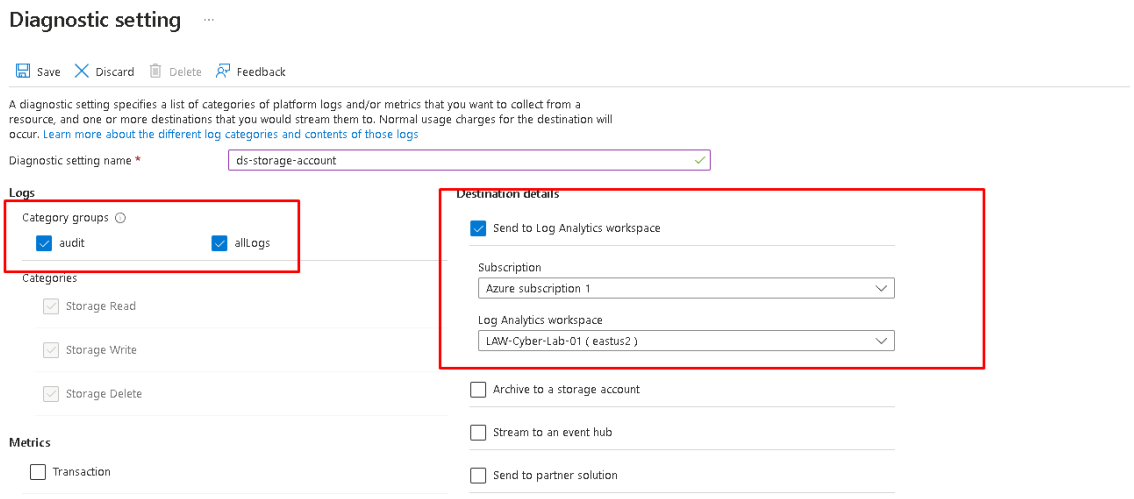


If we test, we have some logs already!

Lastly, we will setup resource logging, we will create a key vault and a blob storage!

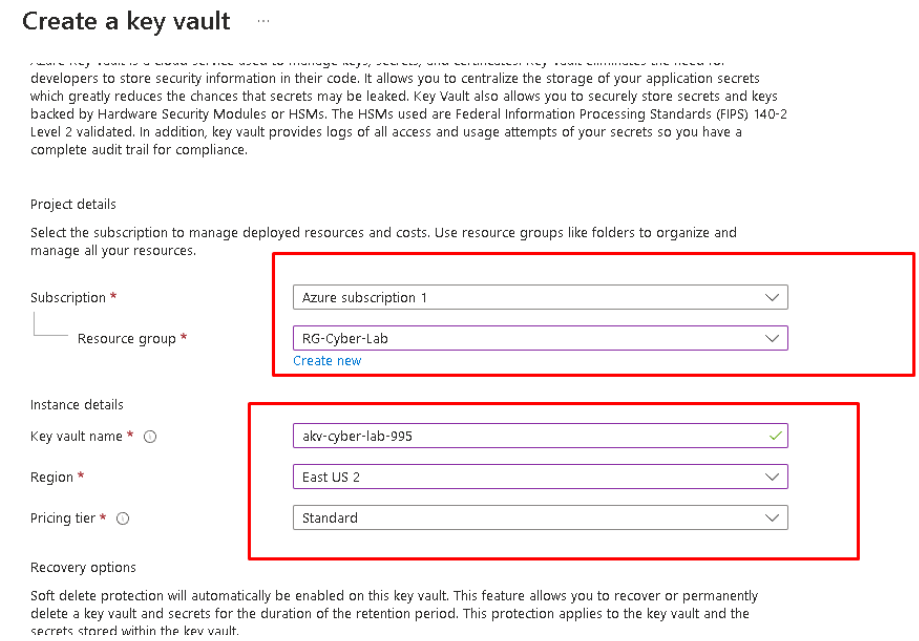


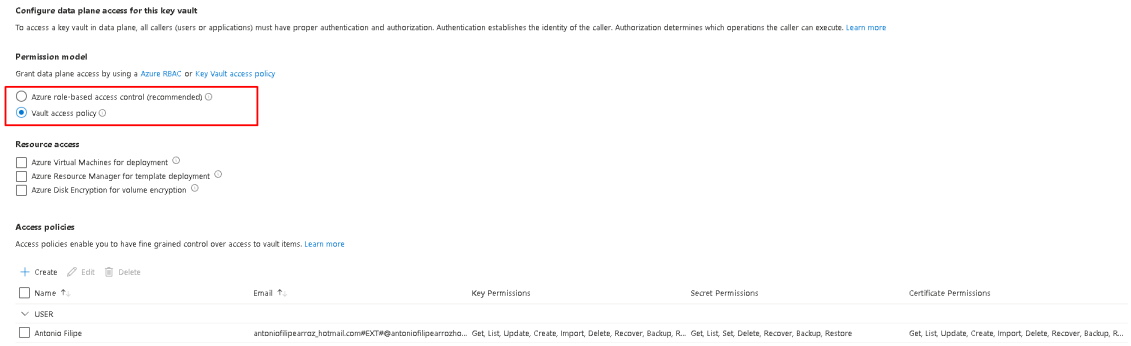
We now have to go to our storage account, we scroll down to diagnostic settings and next we will enable logging for the blob portion of the storage account.



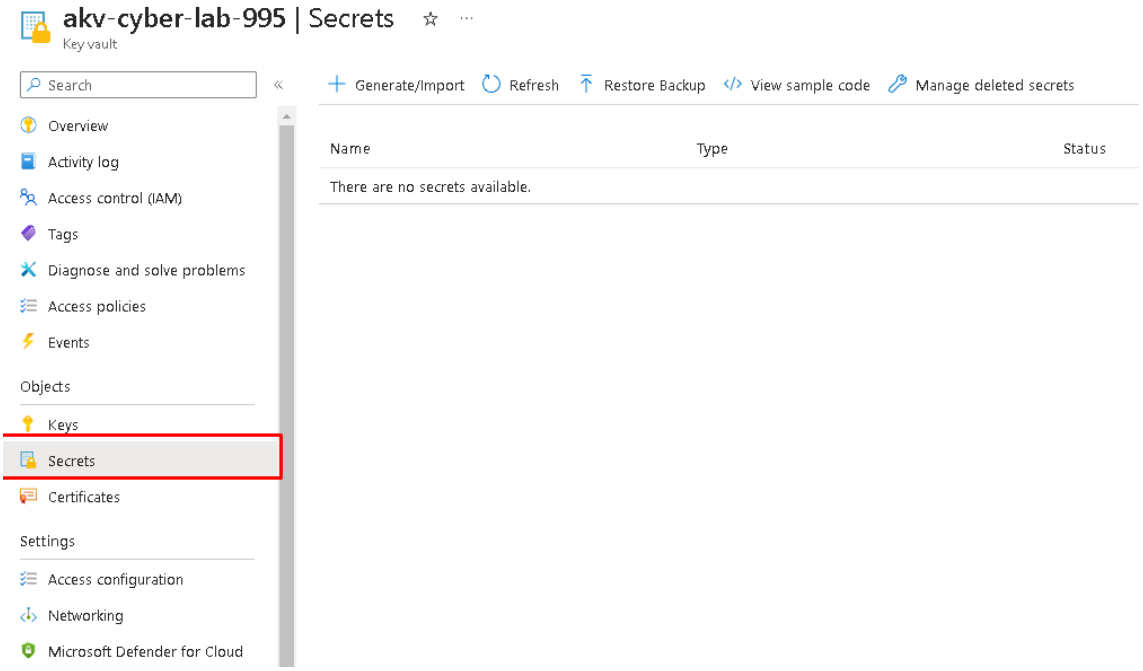
Same things as the others, we will send all logs including audit logs to our analytics workspace!

Let’s create our key vault now!

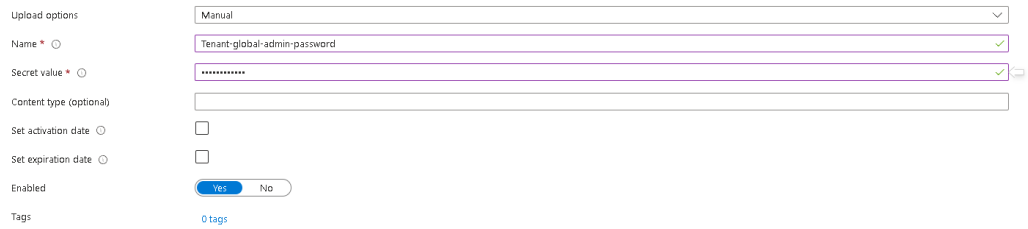




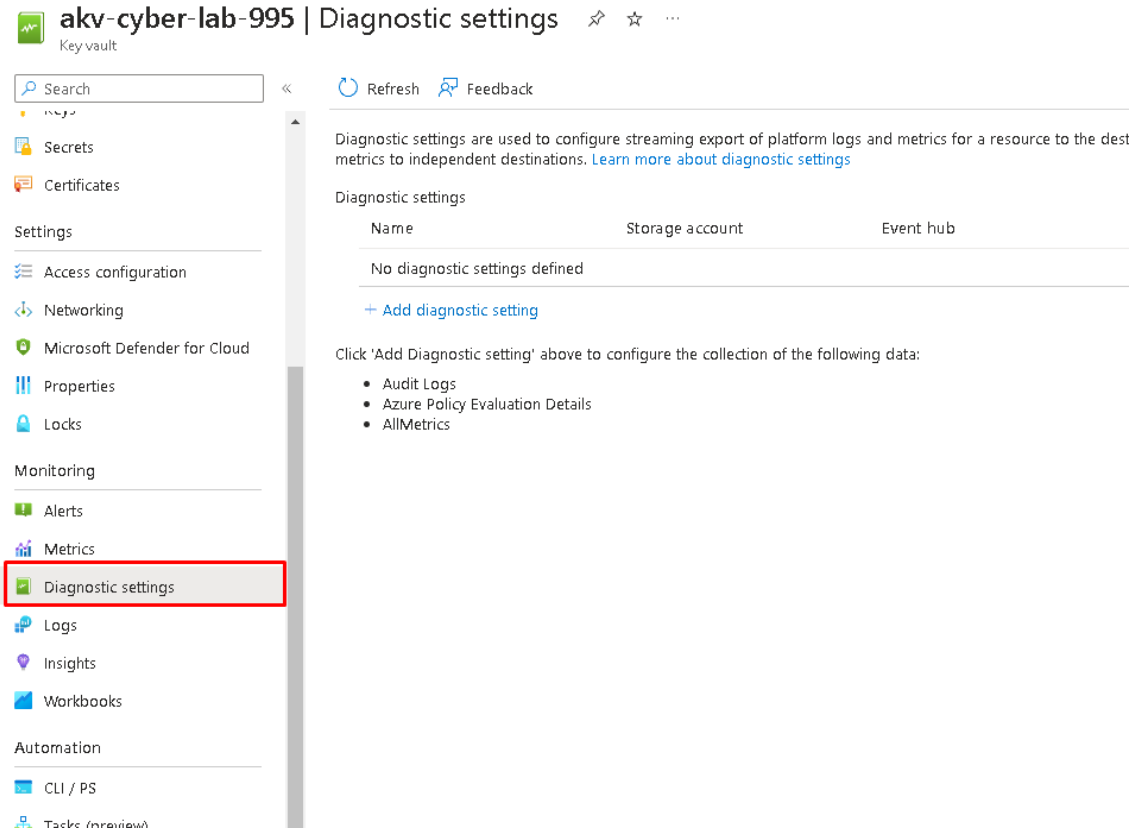
We want to change our vault policy to the one above!



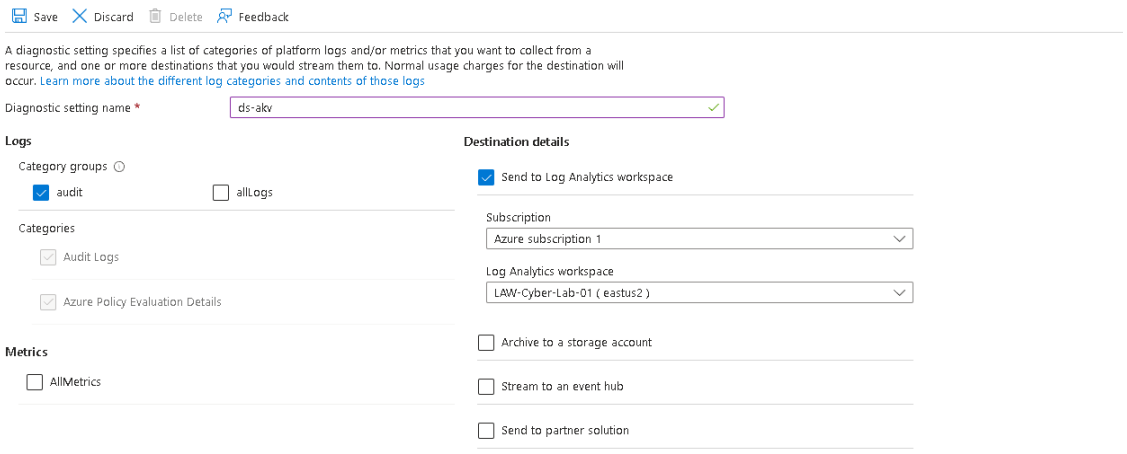
After creating the key vault, we will create a “secret” which is like a password we want to store.



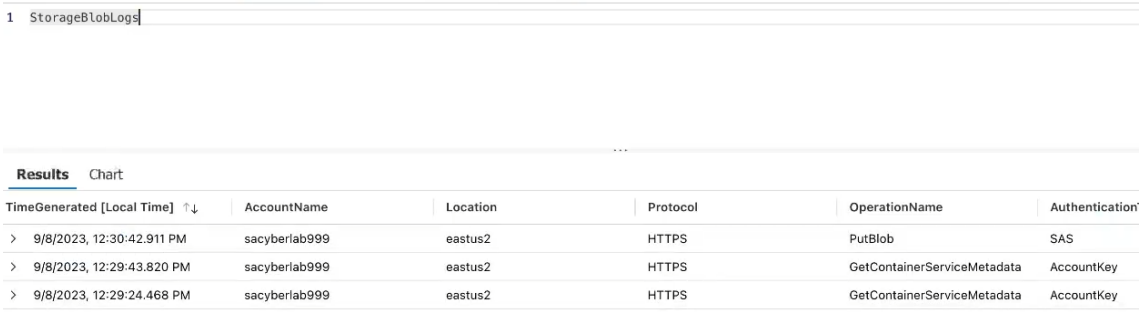
Just create a random secret.



And like always, let’s create Diagnostic settings so we can start generating some logs.



Like this.



And everything is being logged, the blob storage as well as the key vault.

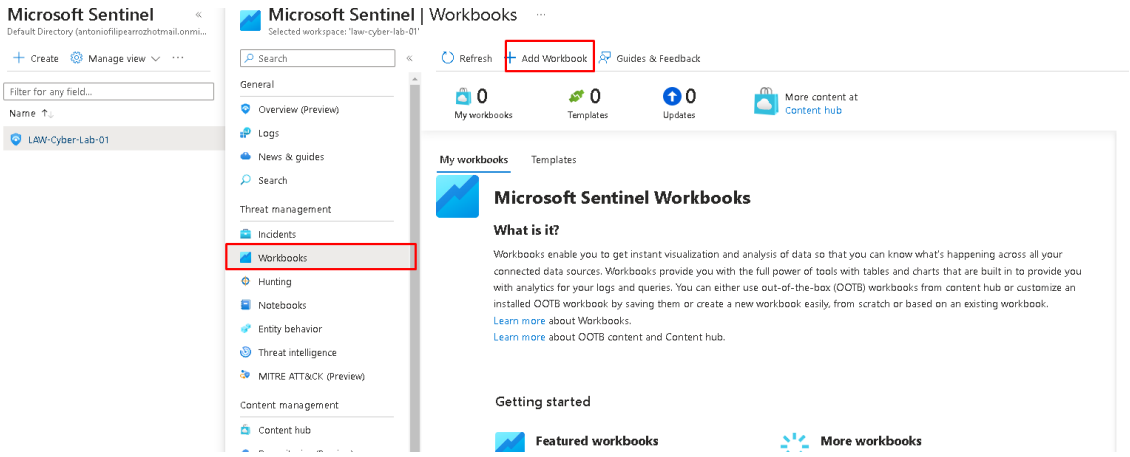
We will finally start working on our SIEM, creating incidents, creating a threat map, harden our environment, work on incidents,etc…

We will start by creating 4 MAPS, one for windows VMs: RDP, SMB / GENERAL AUTHENTICATION FAILURES

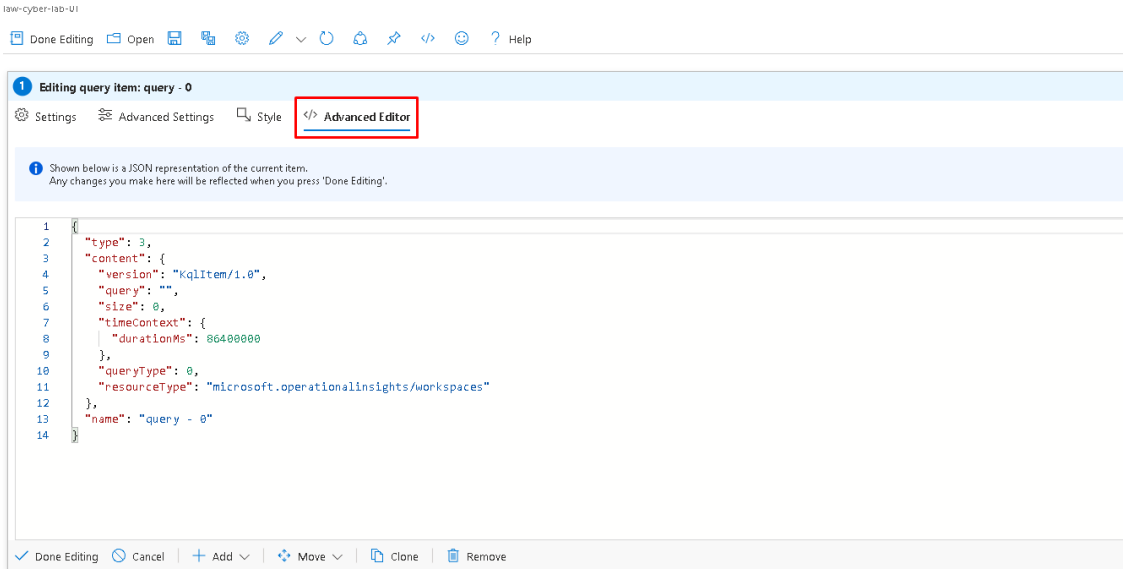
LINUX VMs: SSH authentication FAILURES.

AZURE SQL SERVER: authentication FAILURES.

NETWORK SECURITY GROUP: MALICIOUS FLOWS



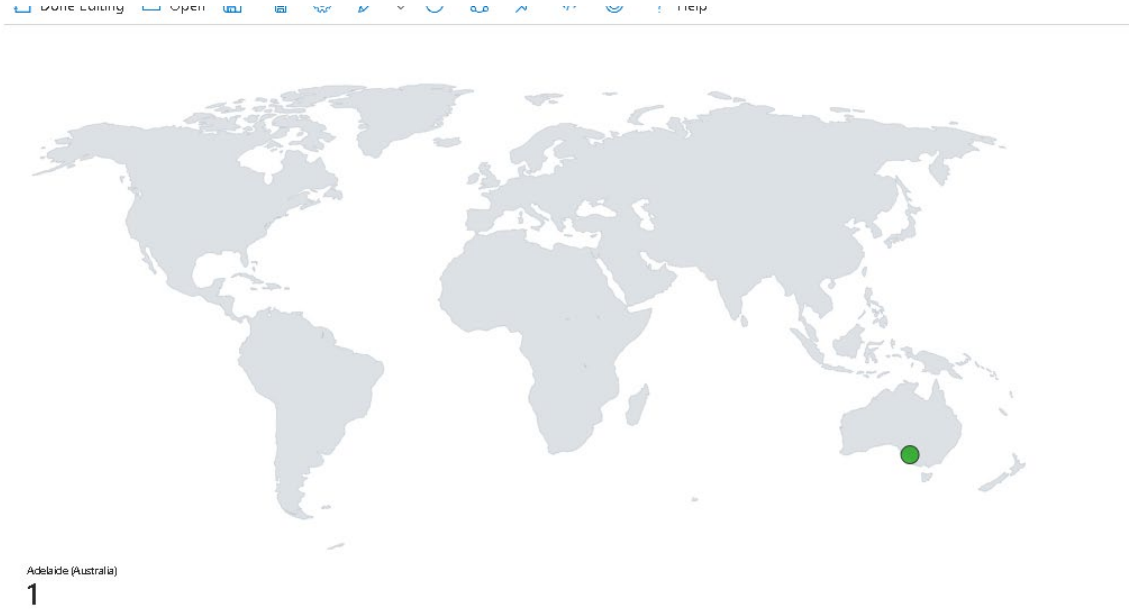
In sentinel, maps are workbooks, so let's create one.



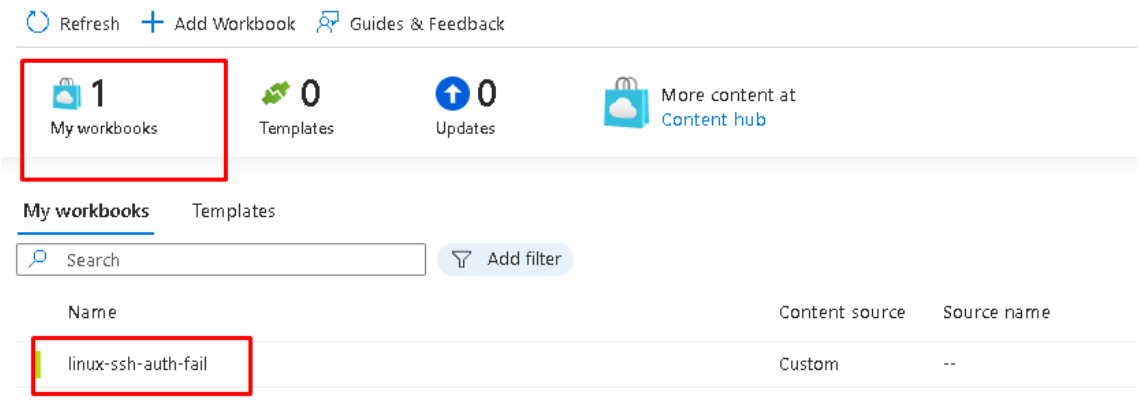
We will edit the default workbook that comes with sentinel and use the advanced feature to create our maps.

```
{
  "type": 3,
  "content": {
    "version": "KqlItem/1.0",
    "query": "let GeoIPDB_FULL = _GetWatchlist(\"geoip\");\nlet IpAddress_REGEX_PATTERN = @\"\\b\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}\\b\";\nSyslog\n| wh
    "size": 3,
    "timeContext": {
      "durationMs": 2592000000
    },
    "queryType": 0,
    "resourceType": "microsoft.operationalinsights/workspaces",
    "visualization": "map",
    "mapSettings": {
      "locInfo": "LatLong",
      "locInfoColumn": "countryname",
      "latitude": "latitude",
      "longitude": "longitude",
      "sizeSettings": "latitude",
      "sizeAggregation": "Count",
      "opacity": 0.8,
      "labelSettings": "friendly_location",
      "legendMetric": "friendly_location",
      "legendAggregation": "Count",
      "itemColorSettings": {
        "nodeColorField": "latitude",
        "colorAggregation": "Count",
        "type": "heatmap",
        "heatmapPalette": "greenRed"
      }
    }
  },
  "name": "query - 0"
}
```

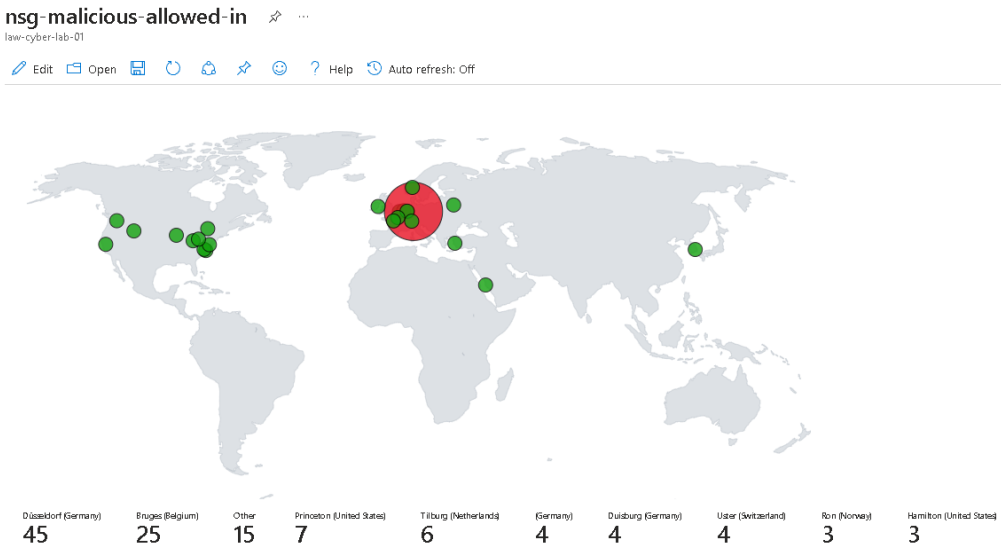

Here is a snipper of what we will be using for the SSH logs/MAP.



And boom! Here is the map, since my VM has been off most of the time, almost nobody has tried to break in yet.



Now that we have 1 created, I will go ahead and create the other 3 Maps.



For example, there is a lot of malicious traffic from our NSG.

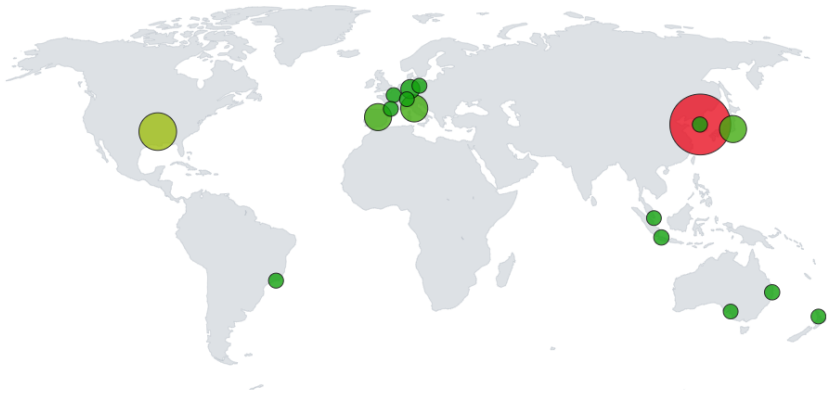
24 hours later and here is how the graphs are looking:

Home > Microsoft Sentinel | Workbooks >

linux-ssh-auth-fail

law-cyber-lab-01

Edit Open Help Auto refresh: Off



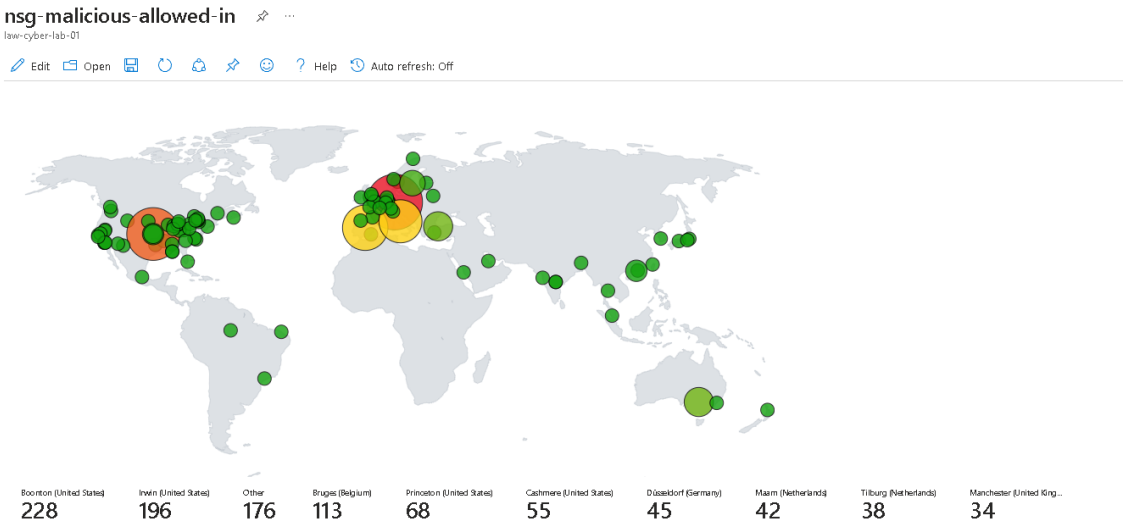
Gangdong-gu (South Kor...	North Little Rock (United ...	Rokucho (Japan)	Madrid (Spain)	San Mauro (Italy)	(Germany)	Other	Jakarta (Indonesia)	Brisbane (Australia)	Seoul (South Korea)
963	344	153	153	153	45	8	6	6	5

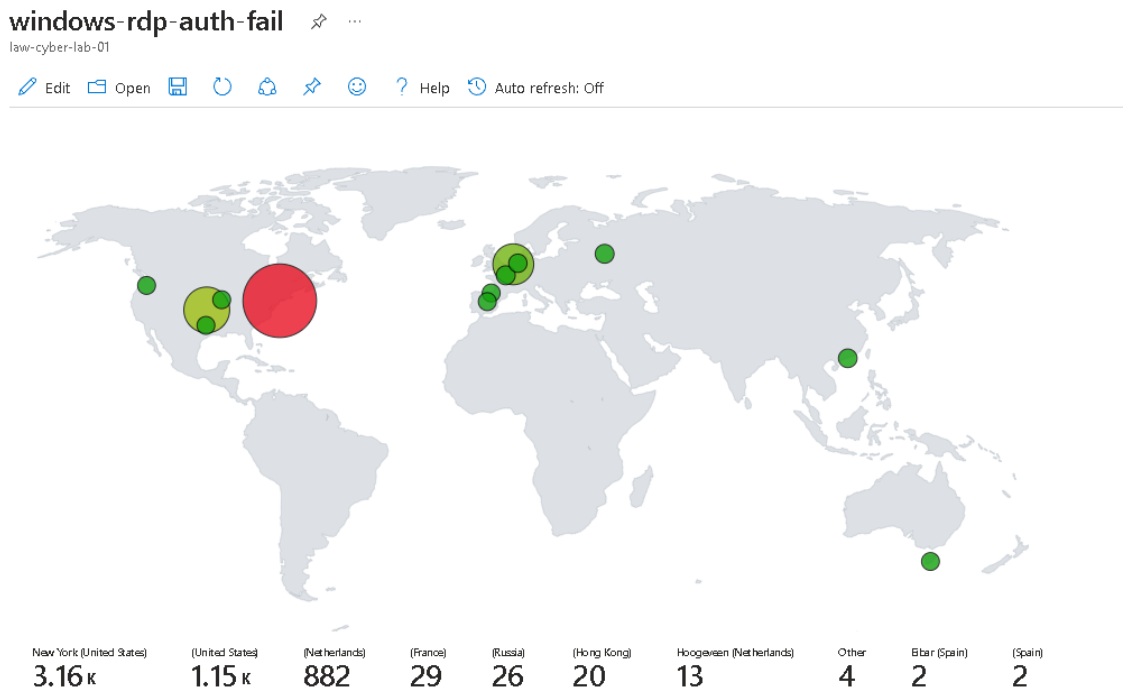
mssql-auth-fail

law-cyber-lab-01

Edit Open Help Auto refresh: Off

Cheongju-si (South Korea)
82





Definitely a lot of malicious traffic and a lot of attack attempts, let's go ahead and create some alerts in our SIEM!

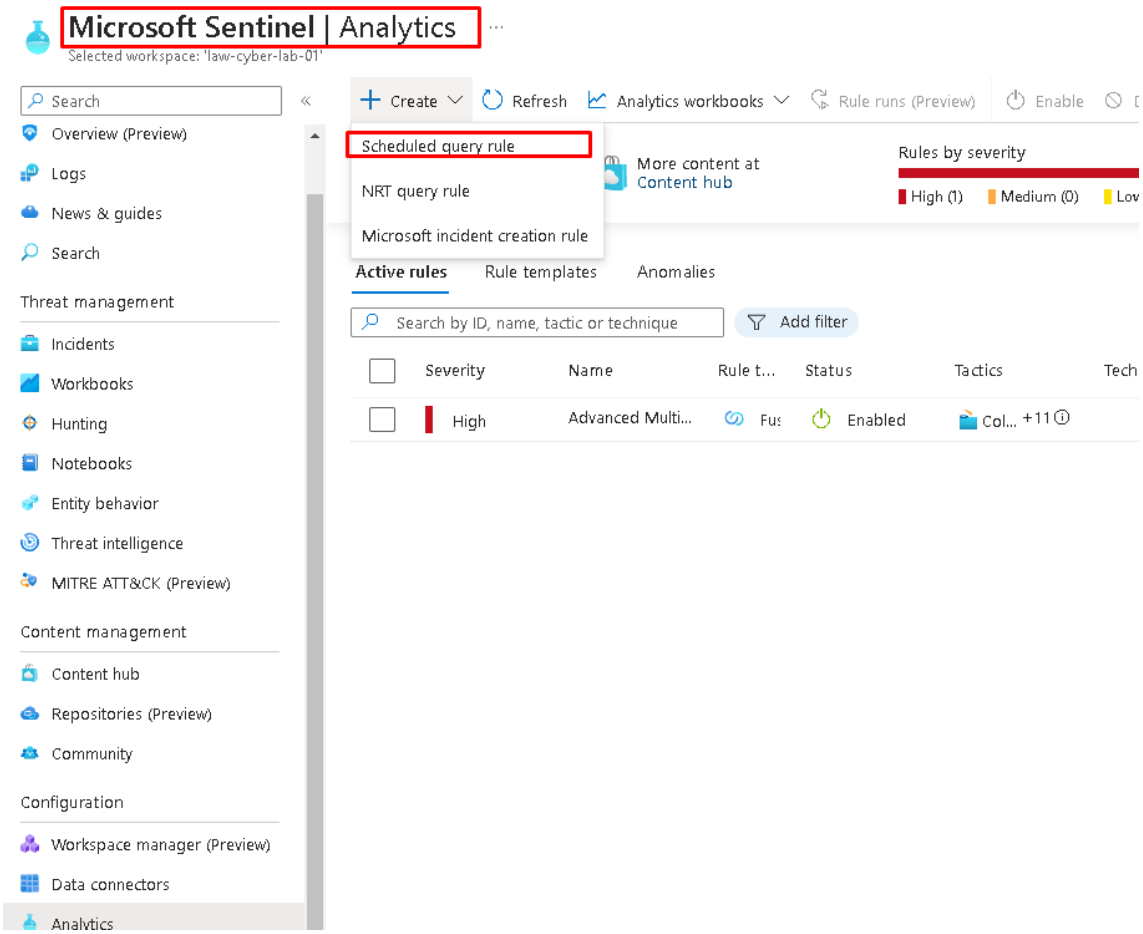
- ```
  SecurityEvent
  | where EventID == 4625
  | where TimeGenerated > ago(60m)
  | summarize FailureCount = count() by AttackerIP =
  IpAddress, EventID, Activity, DestinationHostName =
  Computer
  | where FailureCount >= 10
  ```

THIS WILL BE OUR TEST ALERT!
  ```

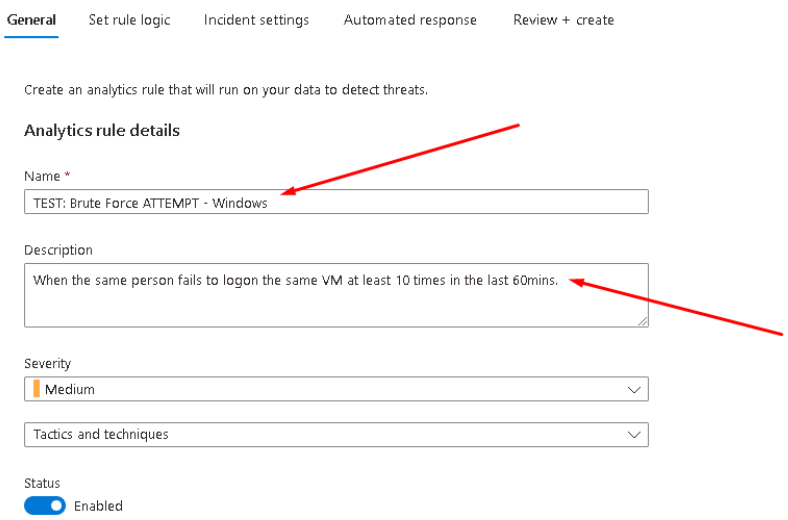
And pretty much what this is checking is for a windows brute force attack, like CHATGPT puts it:

1. `SecurityEvent`: This specifies the data source that the query will operate on, which in this case is a table or dataset containing security events.
2. `| where EventID == 4625`: This filters the events to include only those where the `EventID` is `4625`, which typically represents a failed login attempt in Windows security event logs.
3. `| where TimeGenerated > ago(60m)`: This further filters the events to only include those generated in the last 60 minutes.
4. `| summarize FailureCount = count() by AttackerIP = IpAddress, EventID, Activity, DestinationHostName = Computer`: This aggregates the data by counting the number of failed login attempts (`count()`) and groups them by the IP address of the attacker (`IpAddress`), the `EventID`, the type of activity (`Activity`), and the host name of the target machine (`Computer`). The result is aliased to `FailureCount` for the count, `AttackerIP` for the `IpAddress`, and `DestinationHostName` for the `Computer`.
5. `| where FailureCount >= 10`: This final filter includes only the aggregated results where the count of failed login attempts is 10 or more.

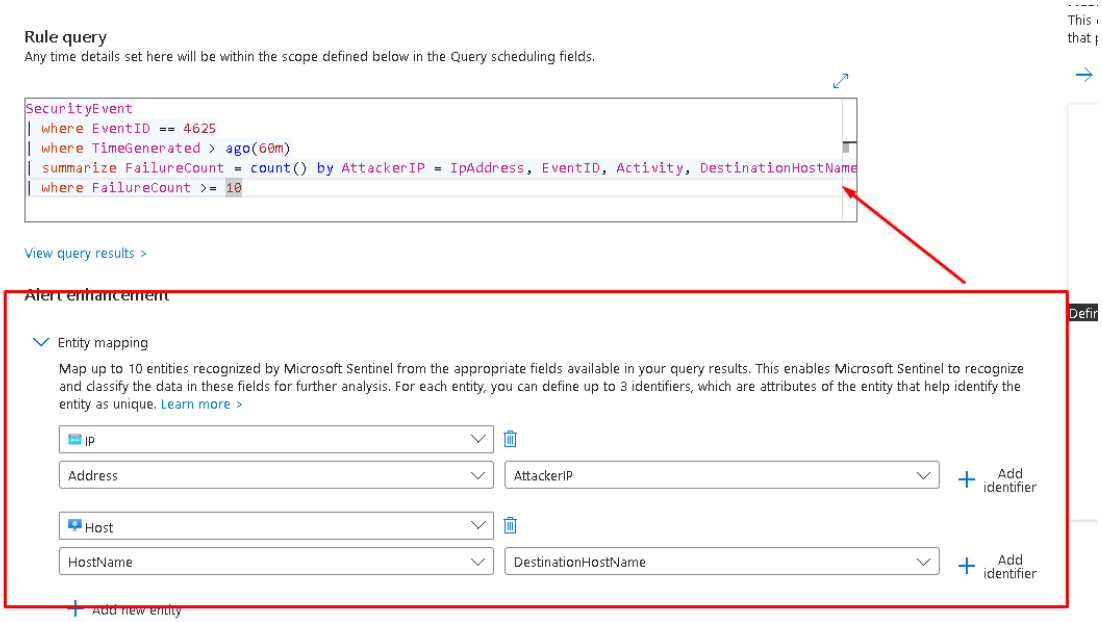
The query is useful for identifying potential brute-force attack attempts by detecting multiple failed login attempts from the same IP address within the last hour. It helps in pinpointing which attackers (by IP address) are frequently trying to log in to which host machines, allowing for a targeted security response.



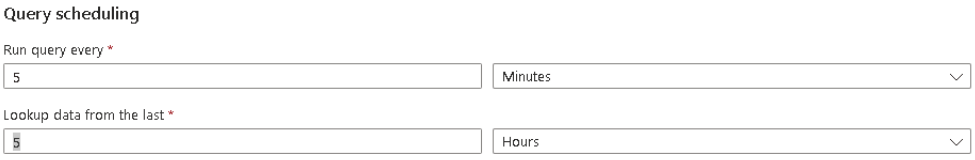
We will go to sentinel and create a scheduled query rule!



We will enter a name of our choosing and a description.

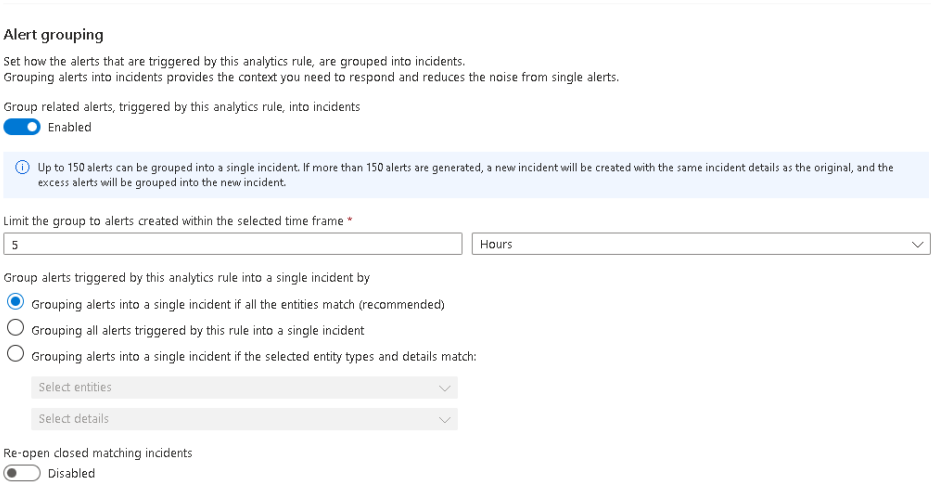


We will put our KQL query on the top and here in entity mapping we will map an IP address as the attacker IP and the host, this pretty much let's sentinel correlate different alerts with one another (let's say the same IP is attacking another HOST, sentinel will know that and alert us as well that it's the same IP).



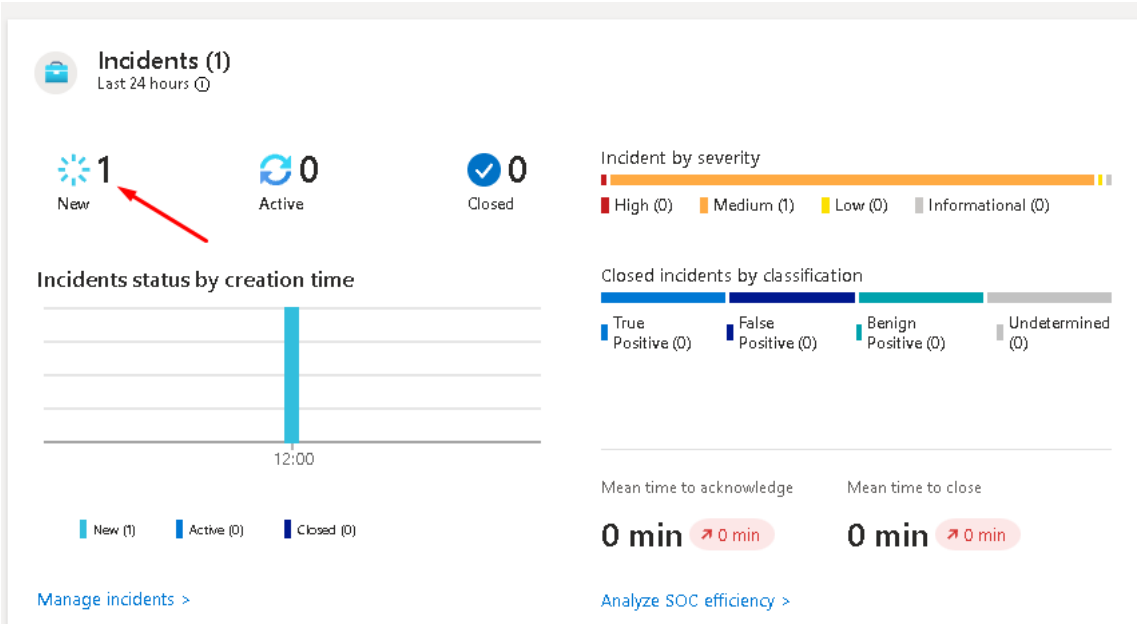
We will make sure this query is running every 5 mins.

Everything else is default!



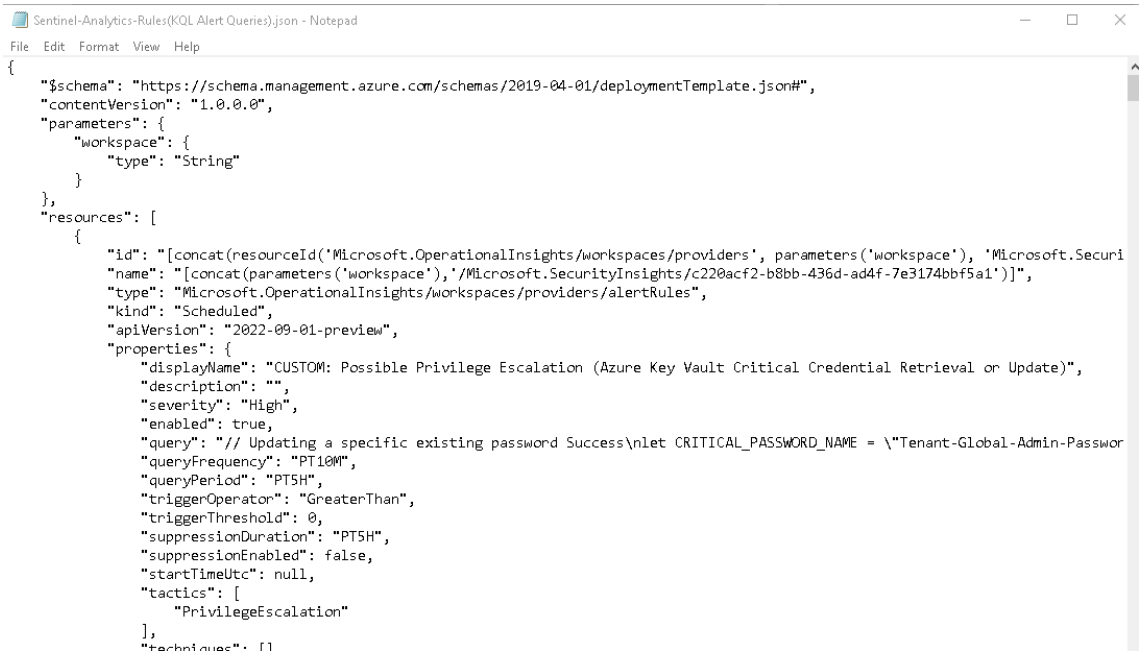
We will enable alert grouping and leave the settings as default.

We don’t want to automate anything because later we will respond to these incidents according to the NIST incident response framework.

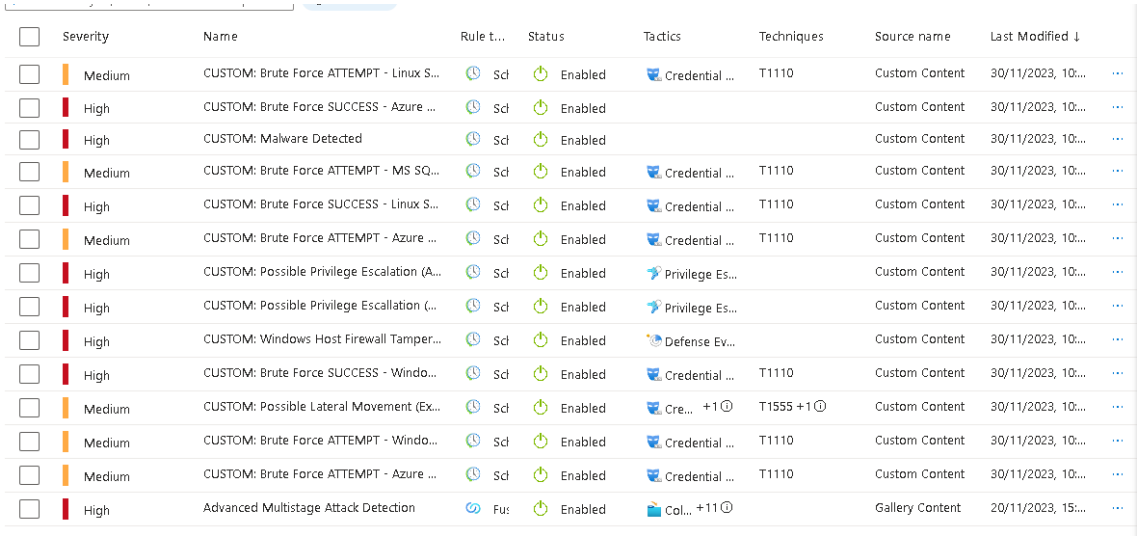


We got “lucky” since the query came back positive without us having to do anything, if the query were to come back with no results, we would have needed to try and “attack” our own windows RDP login for us to get an alert.

Since this rule was mostly for testing, we will now import a whole bunch of detection queries for this project.



We will import this big file with a bunch of detection rules already setup.



Severity	Name	Rule t...	Status	Tactics	Techniques	Source name	Last Modified ↓
Medium	CUSTOM: Brute Force ATTEMPT - Linux S...	Scl	Enabled	Credential ...	T1110	Custom Content	30/11/2023, 10:...
High	CUSTOM: Brute Force SUCCESS - Azure ...	Scl	Enabled			Custom Content	30/11/2023, 10:...
High	CUSTOM: Malware Detected	Scl	Enabled			Custom Content	30/11/2023, 10:...
Medium	CUSTOM: Brute Force ATTEMPT - MS SQ...	Scl	Enabled	Credential ...	T1110	Custom Content	30/11/2023, 10:...
High	CUSTOM: Brute Force SUCCESS - Linux S...	Scl	Enabled	Credential ...	T1110	Custom Content	30/11/2023, 10:...
Medium	CUSTOM: Brute Force ATTEMPT - Azure ...	Scl	Enabled	Credential ...	T1110	Custom Content	30/11/2023, 10:...
High	CUSTOM: Possible Privilege Escalation (A...	Scl	Enabled	Privilege Es...		Custom Content	30/11/2023, 10:...
High	CUSTOM: Possible Privilege Escallation (...	Scl	Enabled	Privilege Es...		Custom Content	30/11/2023, 10:...
High	CUSTOM: Windows Host Firewall Tamper...	Scl	Enabled	Defense Ev...		Custom Content	30/11/2023, 10:...
High	CUSTOM: Brute Force SUCCESS - Windo...	Scl	Enabled	Credential ...	T1110	Custom Content	30/11/2023, 10:...
Medium	CUSTOM: Possible Lateral Movement (Ex...	Scl	Enabled	Cre... +1	T1555 +1	Custom Content	30/11/2023, 10:...
Medium	CUSTOM: Brute Force ATTEMPT - Windo...	Scl	Enabled	Credential ...	T1110	Custom Content	30/11/2023, 10:...
Medium	CUSTOM: Brute Force ATTEMPT - Azure ...	Scl	Enabled	Credential ...	T1110	Custom Content	30/11/2023, 10:...
High	Advanced Multistage Attack Detection	Fus	Enabled	Col... +11		Gallery Content	20/11/2023, 15:...

Everything worked like expected!

And we will have 13 custom detection rules and 1 default rule that comes already with sentinel.

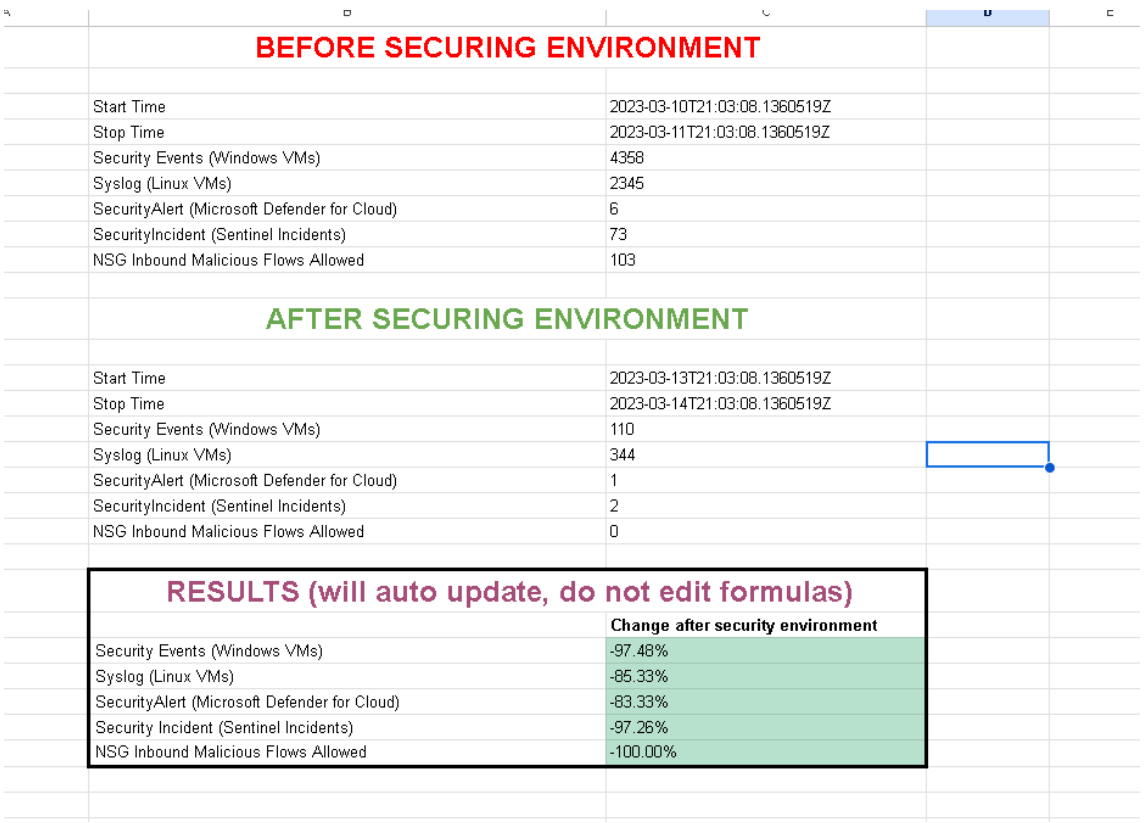
```
// Brute Force Success Windows
let FailedLogons = SecurityEvent
| where EventID == 4625 and LogonType == 3
| where TimeGenerated > ago(1h)
| summarize FailureCount = count() by AttackerIP = IpAddress, EventID,
Activity, LogonType, DestinationHostName = Computer
| where FailureCount >= 5;
let SuccessfulLogons = SecurityEvent
| where EventID == 4624 and LogonType == 3
| where TimeGenerated > ago(1h)
| summarize SuccessfulCount = count() by AttackerIP = IpAddress,
LogonType, DestinationHostName = Computer, AuthenticationSuccessTime =
TimeGenerated;
SuccessfulLogons
| join kind = inner FailedLogons on DestinationHostName, AttackerIP,
LogonType
| project AuthenticationSuccessTime, AttackerIP, DestinationHostName,
FailureCount, SuccessfulCount
```

For example, this is one of our rules that detects brute force success on windows, let’s break it down.

This KQL query is composed of two subqueries that interact with each other, and its primary function is to detect brute force login attempts that have eventually succeeded on Windows systems. Here is a breakdown of the query:

1. **Define `FailedLogons` Subquery**:
    - `SecurityEvent`: This selects the security events to be analyzed.
    - `| where EventID == 4625 and LogonType == 3`: Filters for events with an ID of 4625, which indicates a failed logon attempt, and where the logon type is 3, which usually means a network logon (often used for remote access).
    - `| where TimeGenerated > ago(1h)`: Limits the events to those generated in the last hour.
    - `| summarize FailureCount = count() by AttackerIP = IpAddress, EventID, Activity, LogonType, DestinationHostName = Computer`: Aggregates the number of failed logon attempts, grouping them by IP address of the attacker, event ID, activity type, logon type, and the host name of the target machine.
    - `| where FailureCount >= 5`: Filters the aggregated results to only include records where there have been five or more failed attempts.
2. **Define `SuccessfulLogons` Subquery**:
    - `SecurityEvent`: Again, selects the security events to be analyzed.
    - `| where EventID == 4624 and LogonType == 3`: Filters for events with an ID of 4624, which indicates a successful logon, again where the logon type is 3.
    - `| where TimeGenerated > ago(1h)`: Limits the events to those generated in the last hour.
    - `| summarize SuccessfulCount = count() by AttackerIP = IpAddress, LogonType, DestinationHostName = Computer, AuthenticationSuccessTime = TimeGenerated`: Aggregates the number of successful logons, grouped by attacker IP, logon type, destination host name, and the time the authentication succeeded.
3. **Combine `FailedLogons` and `SuccessfulLogons`**:
    - `SuccessfulLogons`: This initiates the join operation with the results of the `SuccessfulLogons` subquery.
    - `| join kind = inner FailedLogons on DestinationHostName, AttackerIP, LogonType`: Performs an inner join between the two subqueries on the destination host name, attacker IP, and logon type. This join will match records from `SuccessfulLogons` with records from `FailedLogons` where the logon type, attacker IP, and destination host name are the same.
    - `| project AuthenticationSuccessTime, AttackerIP, DestinationHostName, FailureCount, SuccessfulCount`: Projects (selects) the final columns to display in the output, which includes the time of successful authentication, IP address of the attacker, the destination host name, the count of failed attempts, and the count of successful logons.

The resulting output from this query would give you the instances where an IP address had multiple failed logon attempts (suggesting a brute force attack) but eventually managed to successfully log in within the past hour. This could indicate that the brute force attack was successful and may warrant further investigation or immediate action.

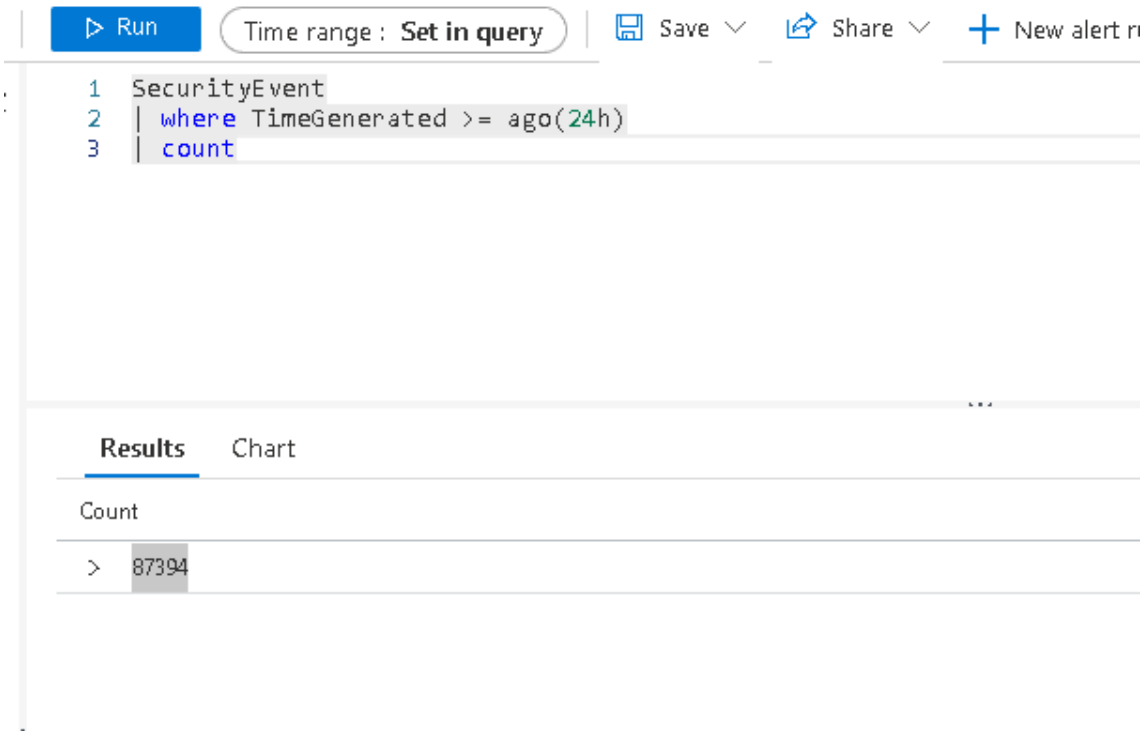


BEFORE SECURING ENVIRONMENT	
Start Time	2023-03-10T21:03:08.1360519Z
Stop Time	2023-03-11T21:03:08.1360519Z
Security Events (Windows VMs)	4358
Syslog (Linux VMs)	2345
SecurityAlert (Microsoft Defender for Cloud)	6
SecurityIncident (Sentinel Incidents)	73
NSG Inbound Malicious Flows Allowed	103

AFTER SECURING ENVIRONMENT	
Start Time	2023-03-13T21:03:08.1360519Z
Stop Time	2023-03-14T21:03:08.1360519Z
Security Events (Windows VMs)	110
Syslog (Linux VMs)	344
SecurityAlert (Microsoft Defender for Cloud)	1
SecurityIncident (Sentinel Incidents)	2
NSG Inbound Malicious Flows Allowed	0

RESULTS (will auto update, do not edit formulas)	
	Change after security environment
Security Events (Windows VMs)	-97.48%
Syslog (Linux VMs)	-85.33%
SecurityAlert (Microsoft Defender for Cloud)	-83.33%
Security Incident (Sentinel Incidents)	-97.26%
NSG Inbound Malicious Flows Allowed	-100.00%

Before we keep on going, we will record our stats.



We will record all the numbers in the spreadsheet.

mssql-auth-fail

law-cyber-lab-01

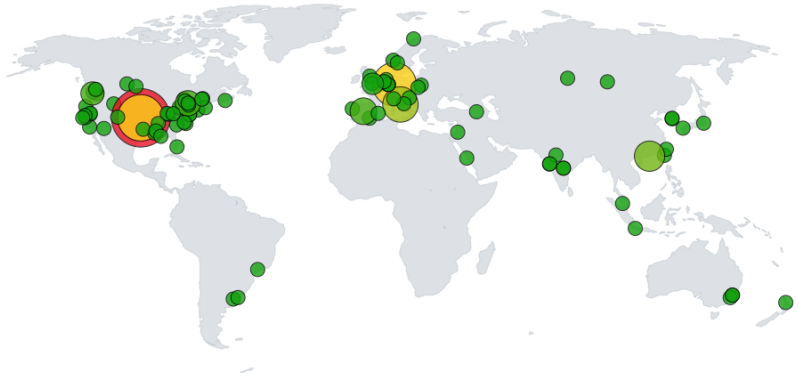
Done Editing | Open | Help

Düsseldorf (Germany)	Eibar (Spain)
7.73 K	2.27 K

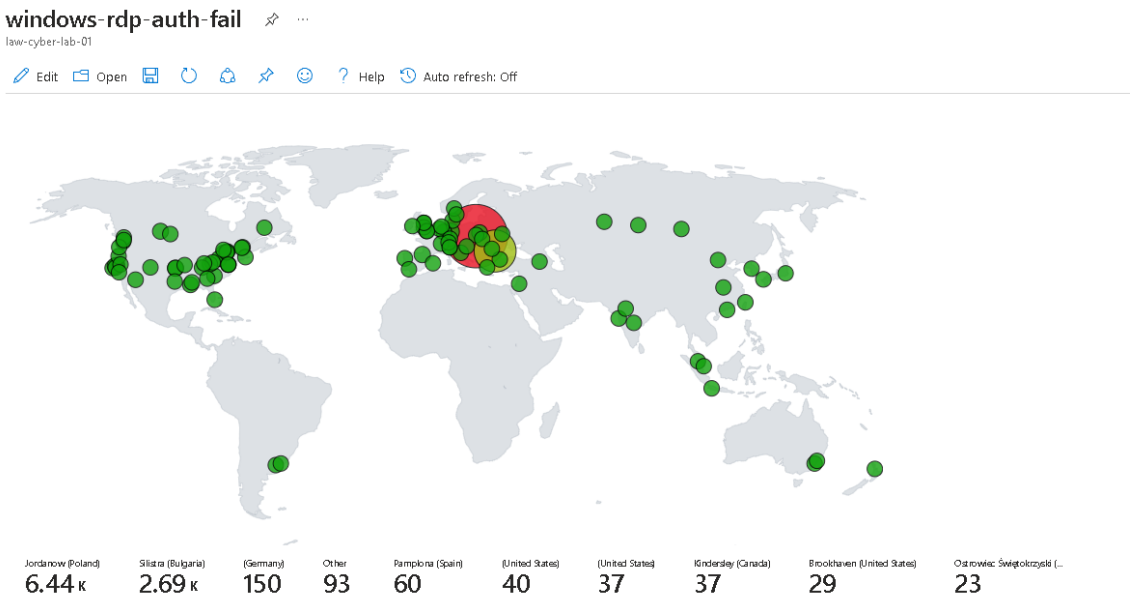
nsg-malicious-allowed-in

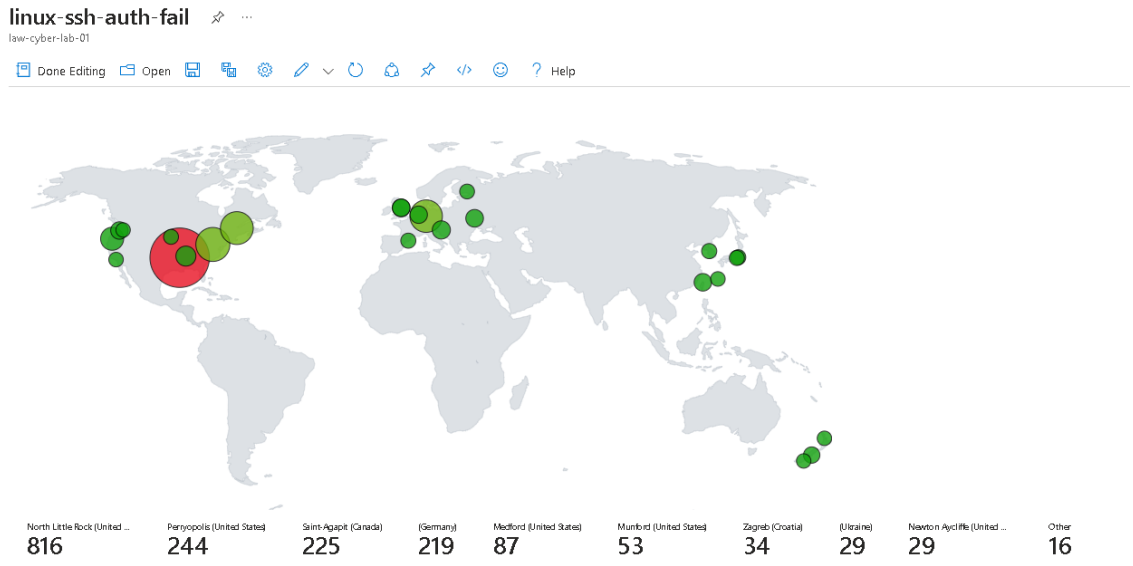
law-cyber-lab-01

Edit | Open | Help | Auto refresh: Off



Other	Boonton (United States)	Irwin (United States)	Tilburg (Netherlands)	(United States)	Bruges (Belgium)	Princeton (United States)	Kindersley (Canada)	(United States)	Sydney (Australia)
269	241	212	138	123	80	78	76	76	68

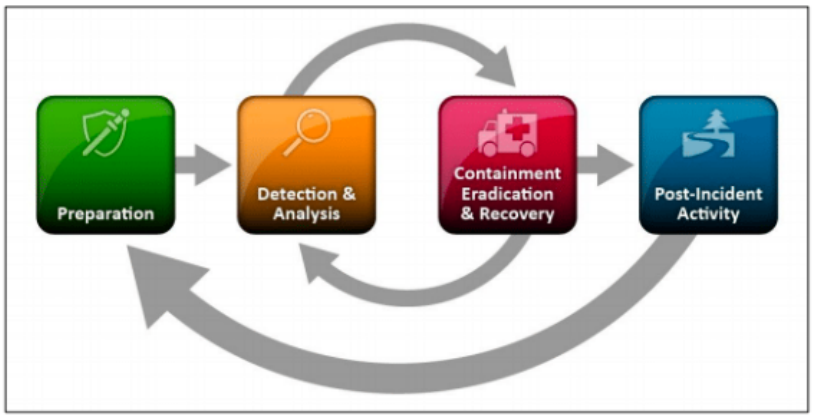




Here is all the malicious traffic in the last 24 hours.

We will now adhere to NIST 800-61 to practice incident response.

It goes something like this:



Step 1: Preparation

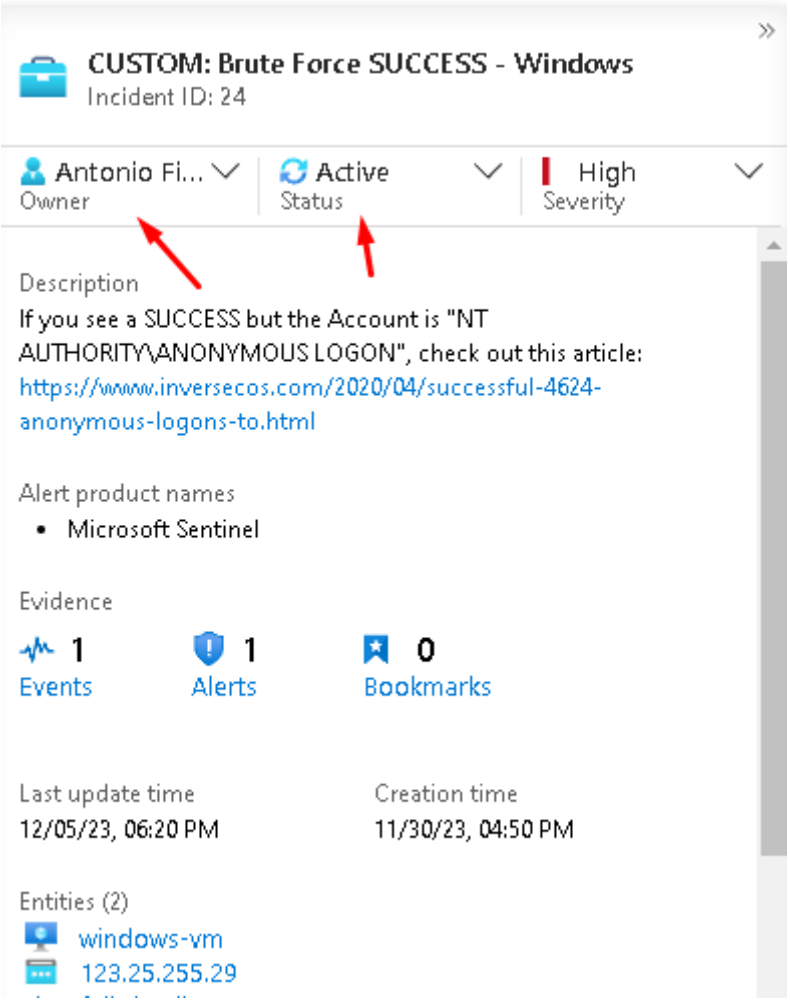
- (We initiated this already by ingesting all of the logs into Log Analytics Workspace and Sentinel and configuring alert rules)

Step 2: Detection & Analysis (You may have different alerts/incidents)

1. Set Severity, Status, Owner
2. View Full Details (New Experience)
3. Observe the Activity Log (for history of incident)
4. Observe Entities and Incident Timelines (are they doing anything else?)
5. "Investigate" the incident and continue trying to determine the scope
6. Inspect the entities and see if there are any related events
7. Determine legitimacy of the incident (True Positive, False Positive, etc.)
8. If True Positive, continue, if False positive, close it out

Step 3: Containment, Eradication, and Recovery

- Use the simple Incident Response PlayBook



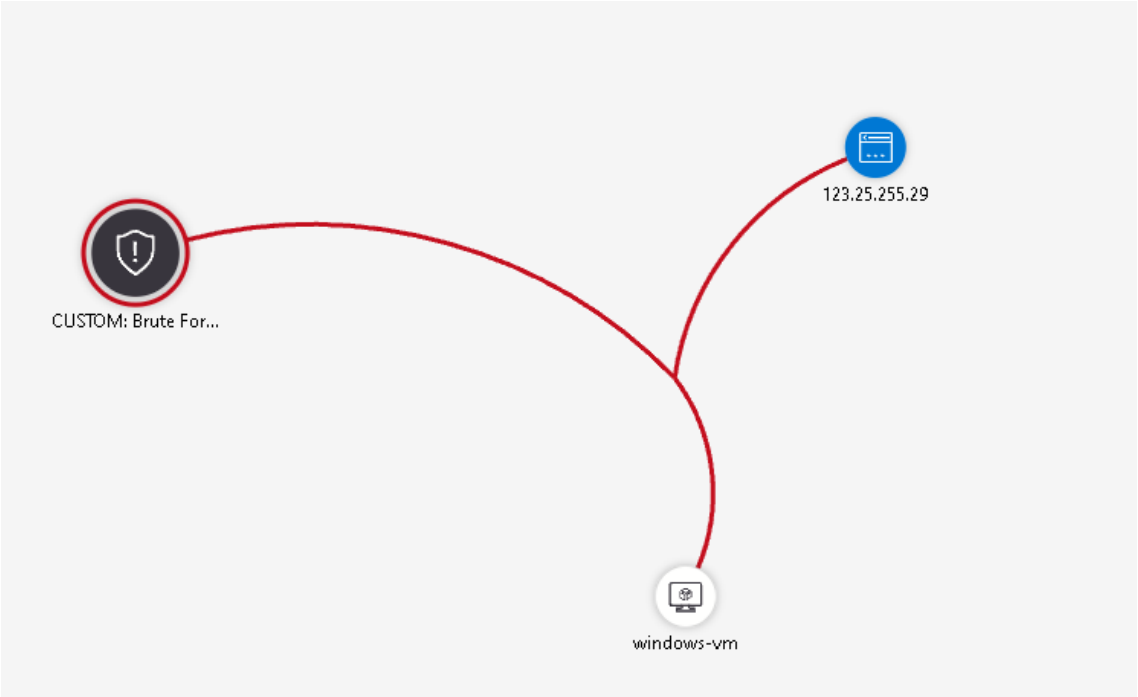
Geolocation information

Organization Vietnam Posts And Telecommunications Group	Organization type Telecommunications
City Dong Da	Country Vietnam
State Ha Noi	Continent Asia

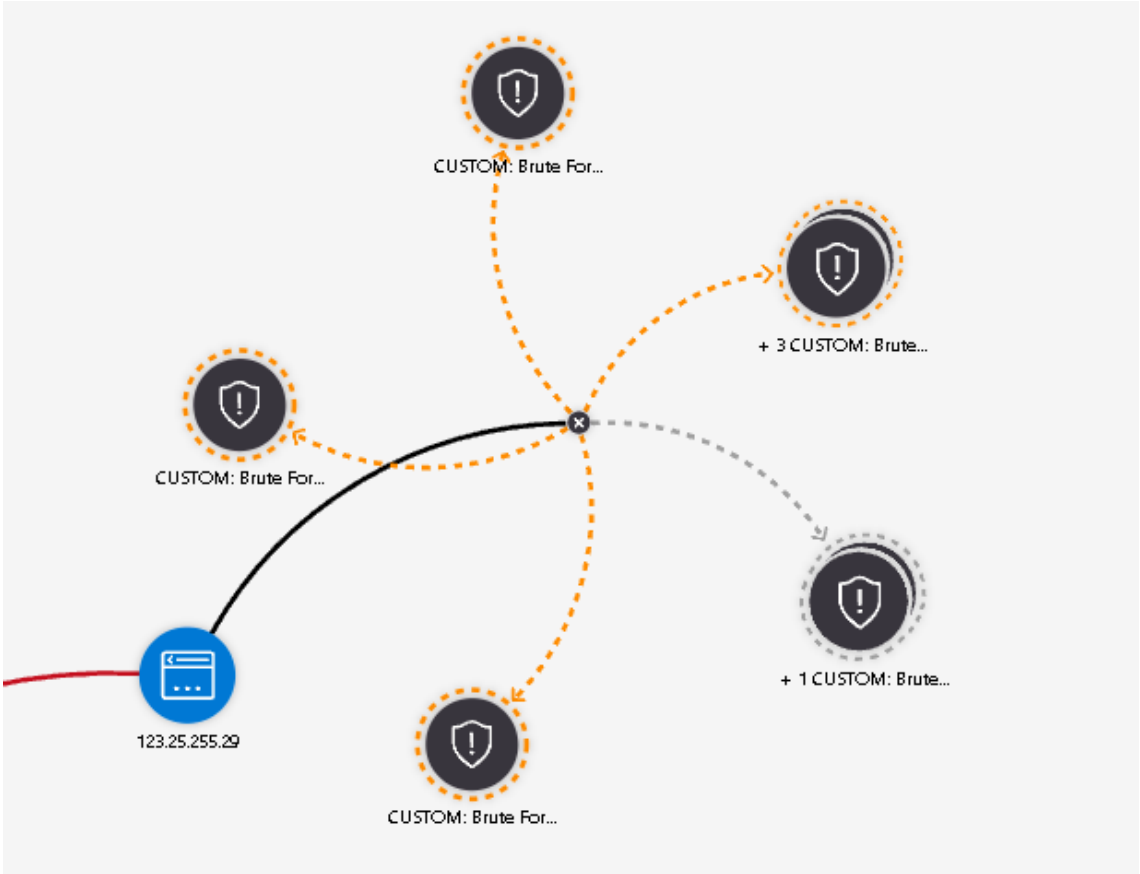
Log activity

First seen 30/11/2023, 16:11:43	Last seen 30/11/2023, 17:26:43

We have info that this person is from Vietnam.

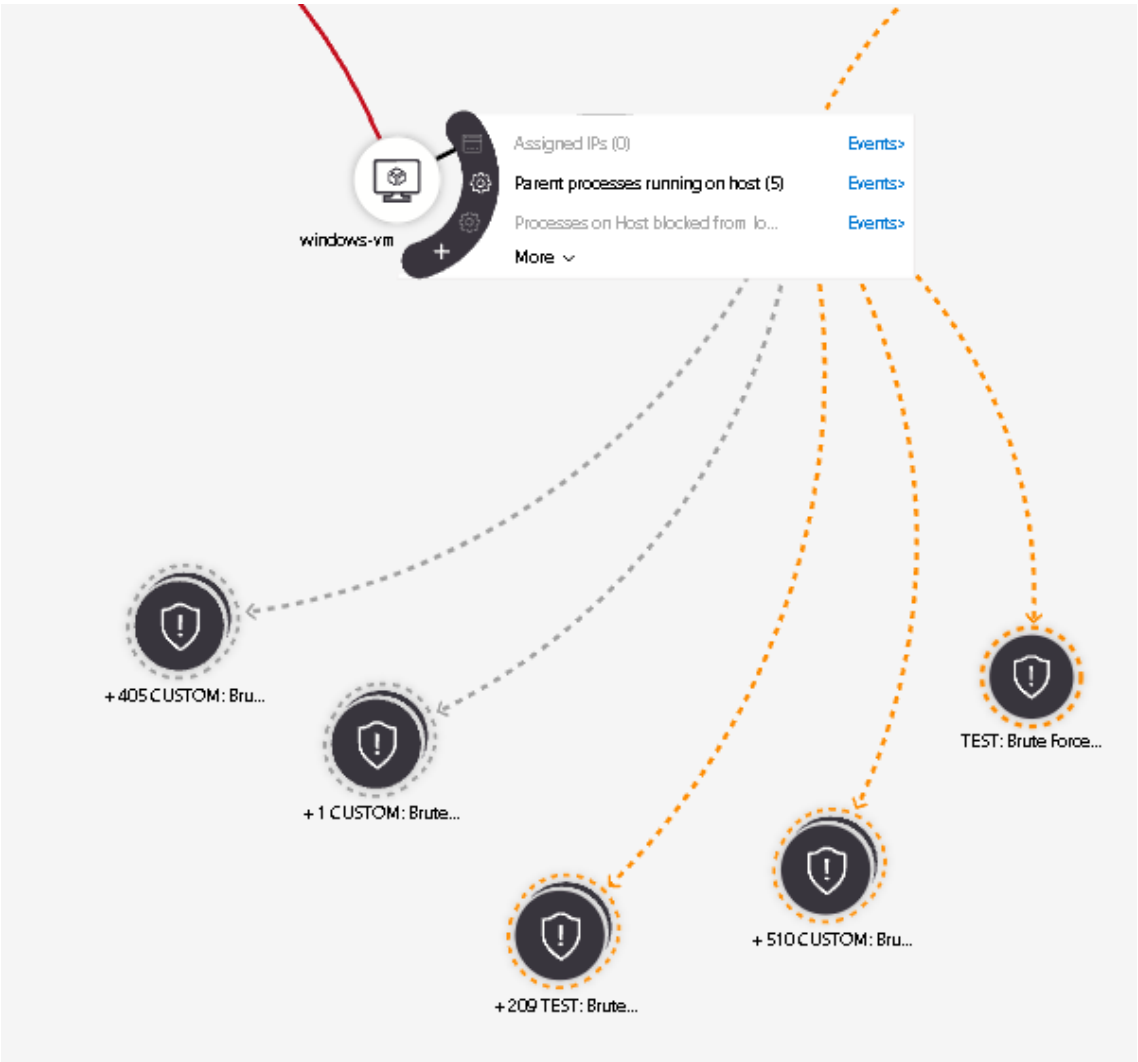


We can see that this attacker was not able to attack other hosts.



There are other events associated with this same attacker though, brute force attempts…

```
Potentially compromised system "windows-vm" involved in several other incidents/alerts.
```



Seems like a true positive as the KQL query is very specific.

This KQL (Kusto Query Language) query is designed to identify potential brute force attack attempts on Windows systems, specifically by detecting multiple failed logon attempts followed by a successful logon within a certain time frame. The query is divided into several parts:

1. **Failed Logons**: This section of the query filters `SecurityEvent` logs for failed logon events (EventID 4625) with LogonType 3, which indicates network logon attempts. It only considers events generated in the last hour (`ago(1h)`). It then summarizes these events, counting the number of failures (`FailureCount`) by the attacker's IP address (`AttackerIP`), along with other details like EventID, Activity, LogonType, and the destination host name (`DestinationHostName`). It filters for instances where there have been five or more failed attempts from the same IP.

2. **Successful Logons**: Similar to the first part, this section filters for successful logon events (EventID 4624) with LogonType 3, also within the last hour. It summarizes these events by counting the successful attempts (`SuccessfulCount`) by the IP address, LogonType,

destination host name, and the time of successful authentication (`AuthenticationSuccessTime`).

3. **Joining and Projecting Data**: The final part joins the two datasets (`SuccessfulLogons` and `FailedLogons`) using an inner join on shared columns (`DestinationHostName`, `AttackerIP`, `LogonType`). This means it only includes records that have matching values in both datasets for these columns. The `project` statement then specifies which columns to include in the final output: the time of successful authentication, the attacker's IP address, the destination host name, the number of failed attempts, and the number of successful attempts.

The purpose of this query is to identify IP addresses that have had multiple failed logon attempts (suggesting a brute force attack) followed by a successful logon, which could indicate that the attacker eventually guessed the correct credentials. This can be a useful tool for security analysts monitoring for potential security breaches.

Though the alert was a true positive, this type of traffic should not be allowed to reach our VM in the first place, possible NSG configuration issue.

Will close out this incident but will look into NSG.

CUSTOM: Brute Force SUCCESS - Windows and Linux

Incident Description

- This incident involves observation of potential brute force attempts against a Windows VM.

Initial Response Actions

- Verify the authenticity of the alert or report.
- Immediately isolate the machine and change the password of the affected user
- Identify the origin of the attacks and determine if they are attacking or involved with anything else
- Determine how and when the attack occurred
  - Are the NSGs not being locked down? If so, check other NSGs
- Assess the potential impact of the incident.
  - What type of account was it? Permissions?

Containment and Recovery

- Lock down the NSG assigned to that VM/Subnet, either entirely, or to allow only necessary traffic
- Reset the affected user's password
- Enable MFA

Document Findings and Close out Incident

[Back to Top]

This is the playbook for this incident.

We will now lock down NSG.

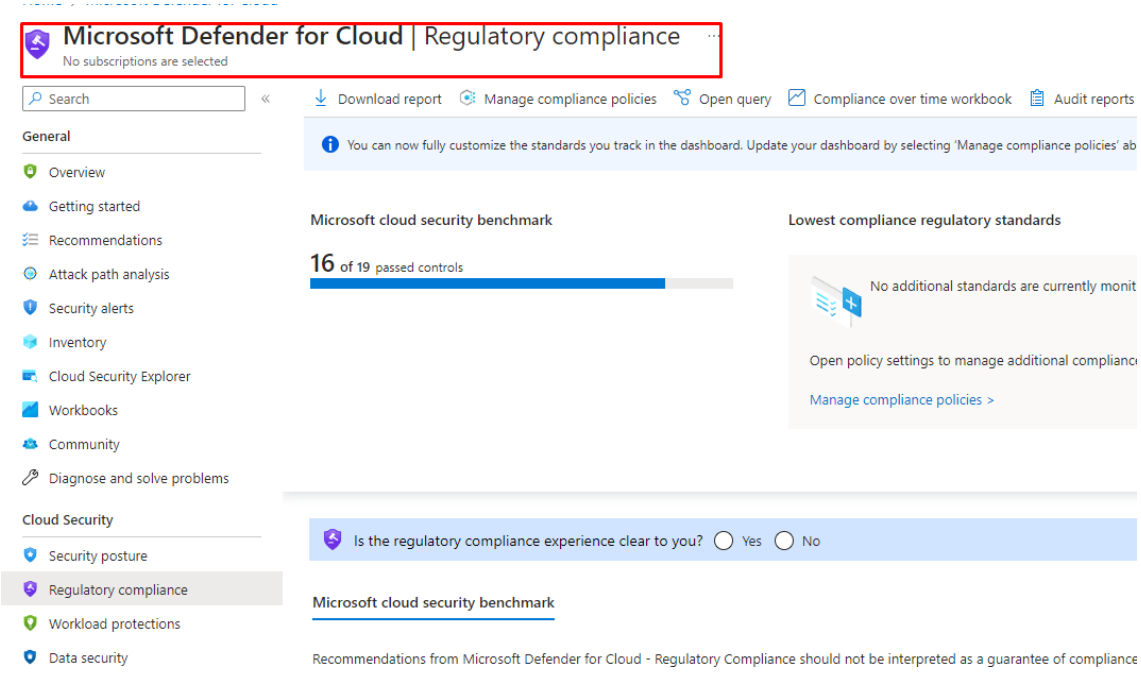
We just put so that only our IP address can access this VM.

So now this attack will not happen anymore as only we can access these VMs.

We will enable MFA later.

I will keep practicing incident response using the incident management playbook that states (reset password, enable mfa, lock down nsg, etc…)

After that I will now Enable NIST 800-53, we will lock down our environment more and more until it becomes complacent with NIST.



We will go to defender for cloud and enable nist 800-53 controls so we can have a checklist on how to harden out environment, security recommendations,etc…

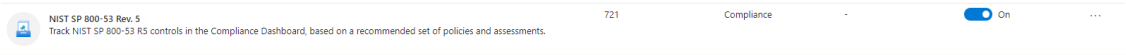
CPRT / SP 800-53 Rev 5

Expand Entire Reference Dataset

Control Families

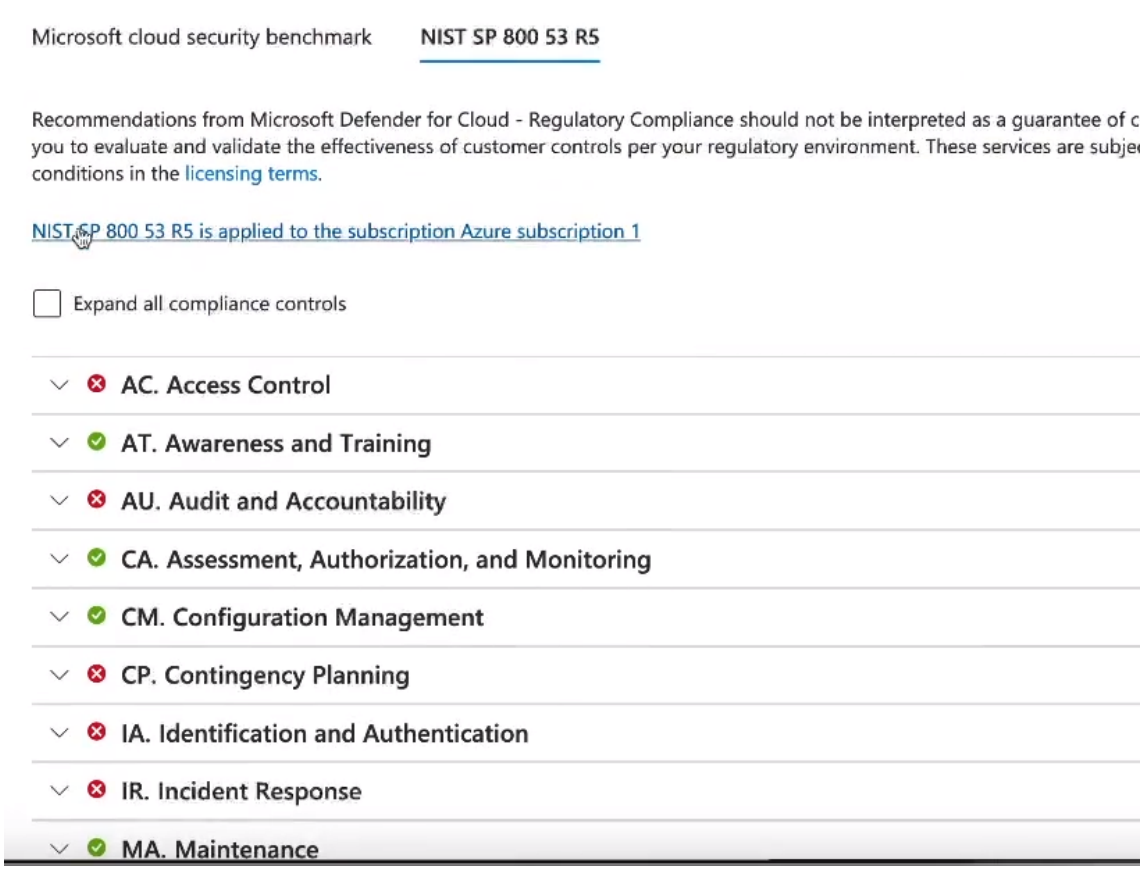
AC	ACCESS CONTROL
AT	AWARENESS AND TRAINING
AU	AUDIT AND ACCOUNTABILITY
CA	ASSESSMENT, AUTHORIZATION, AND MONITORING
CM	CONFIGURATION MANAGEMENT
CP	CONTINGENCY PLANNING
IA	IDENTIFICATION AND AUTHENTICATION
IR	INCIDENT RESPONSE
MA	MAINTENANCE
MP	MEDIA PROTECTION
PE	PHYSICAL AND ENVIRONMENTAL PROTECTION
PL	PLANNING
PM	PROGRAM MANAGEMENT
PS	PERSONNEL SECURITY
PT	PERSONALLY IDENTIFIABLE INFORMATION PROCESSING AND TRANSPARENCY
RA	RISK ASSESSMENT
SA	SYSTEM AND SERVICES ACQUISITION

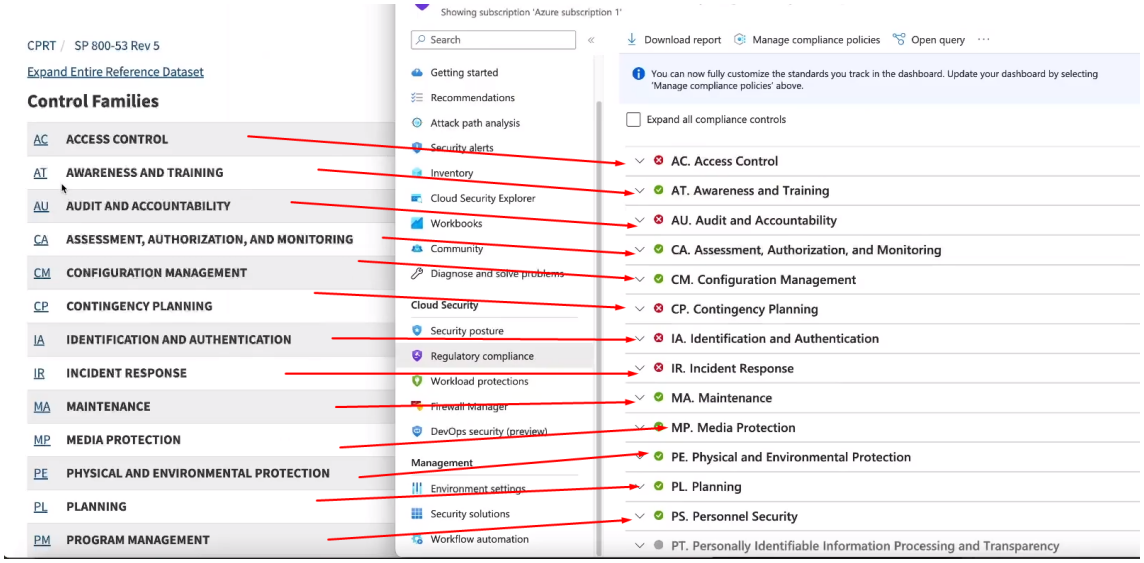
Nist 800-53 family controls.



We just added revision 5!

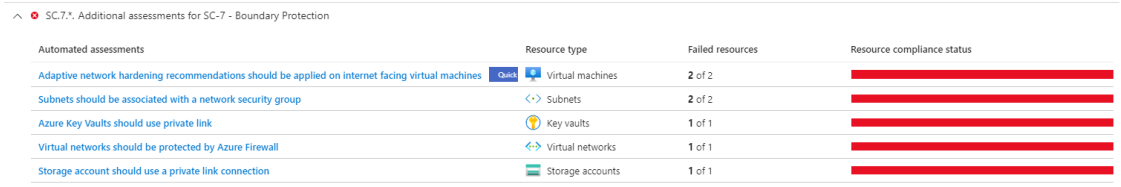
Now that nist 800-53 family controls are enabled, we can go ahead and look at this:



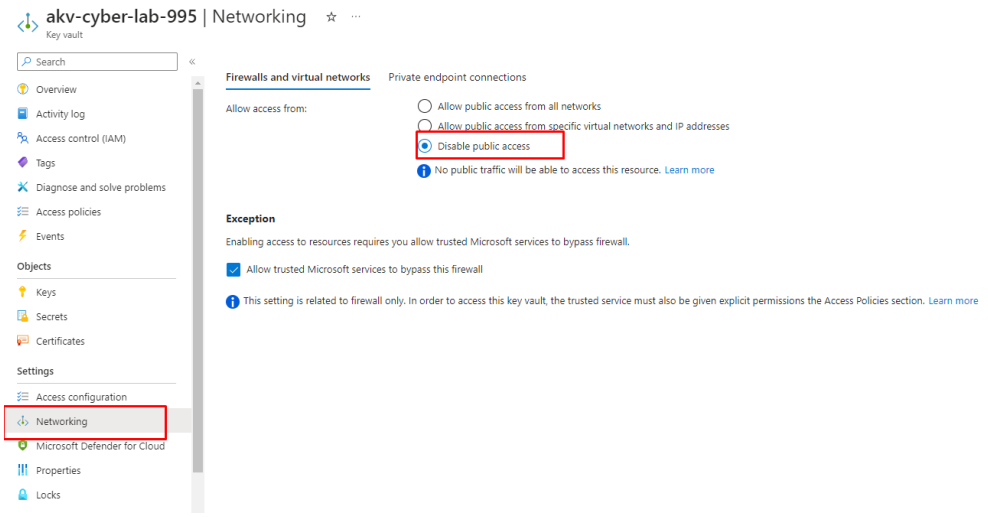


And we can see it does indeed match.

We will implement NIST 800-53: SC-7: BOUNDRY PROTECTION

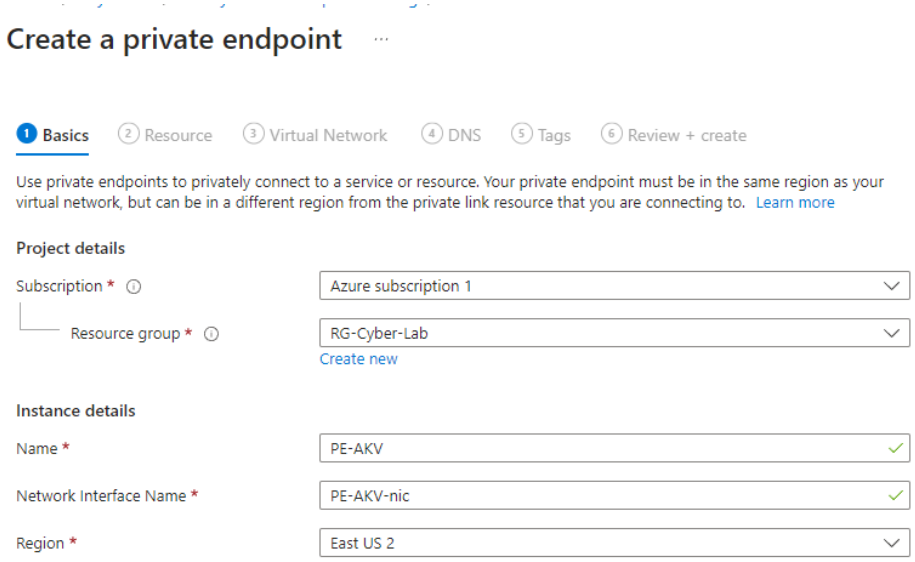


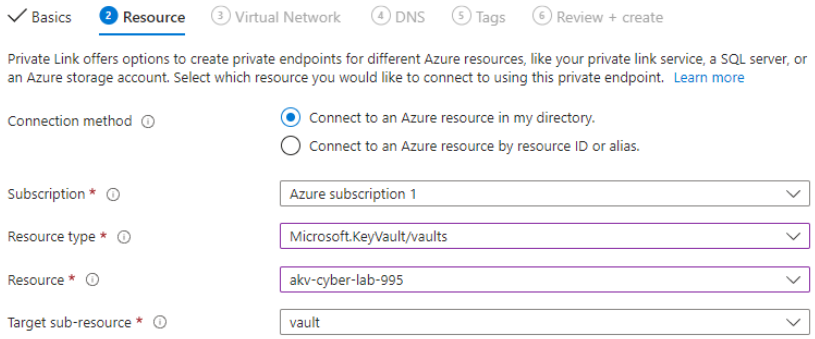
We will start first by configuring a private link and firewall for our key vault.



This is what we must do.

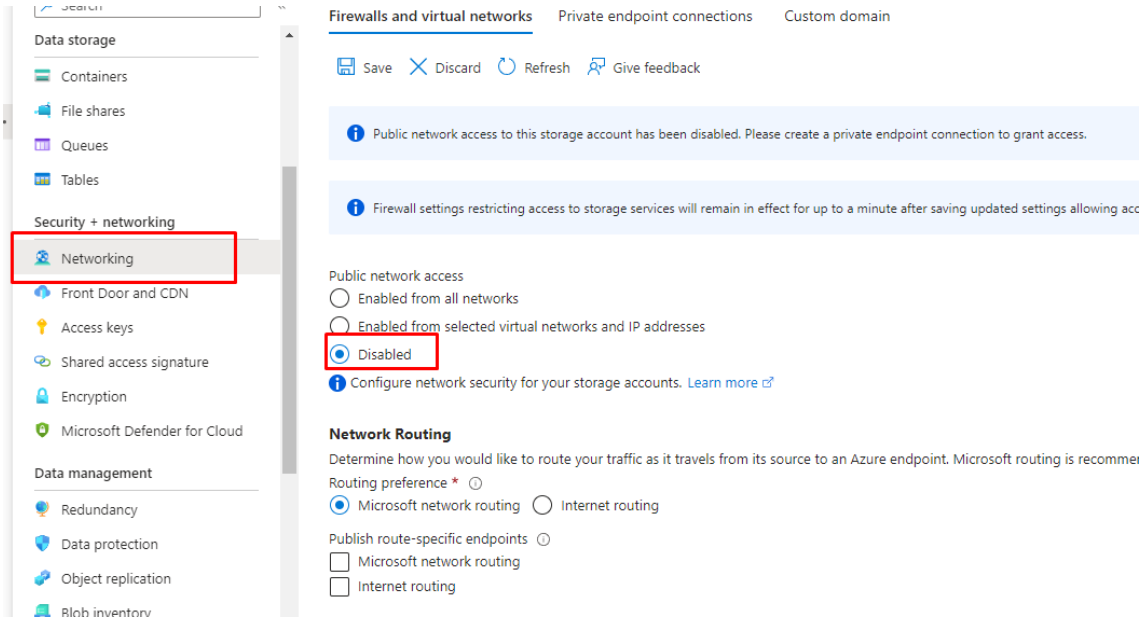
Next to that tab we will also create our private endpoint:

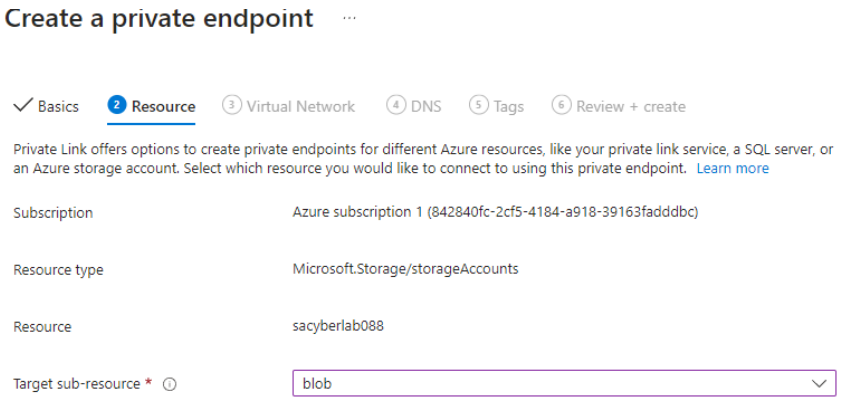




Rest is left at default.

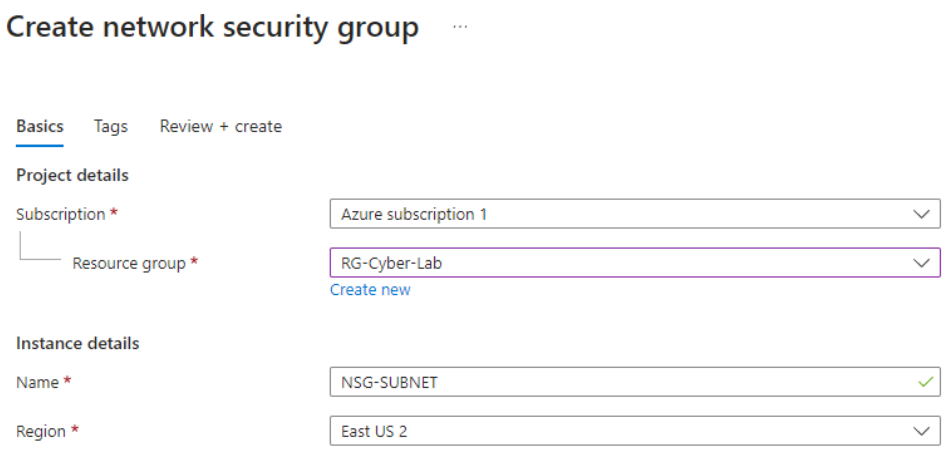
We will now do the same for our storage accounts.



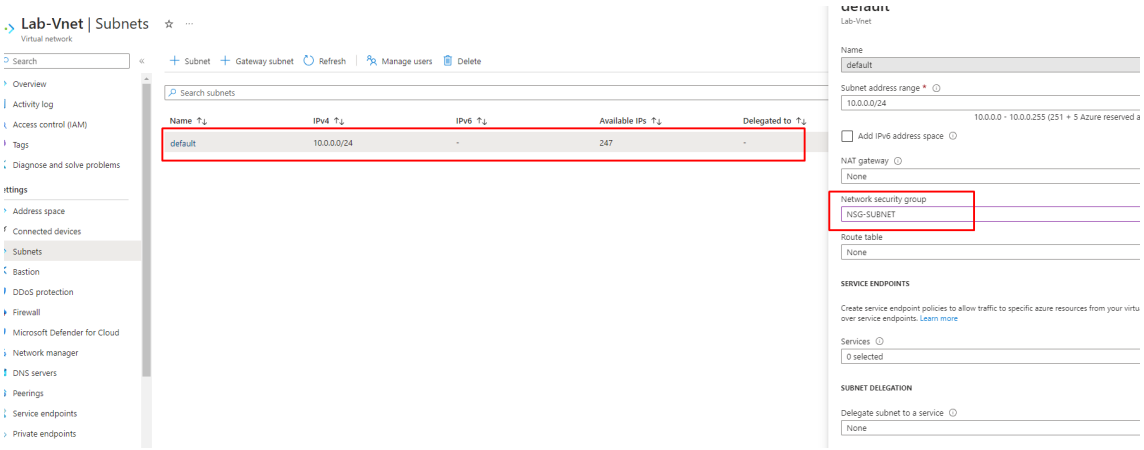


The rest is just like the previous one, but we want to make sure blob is selected here since we are creating our private endpoint for a blob storage account.

Next, we will attach the network security group to the subnet.



Now that the nsg is created we will attach it to our subnet. This is mostly to satisfy the recommendation that defender for cloud gives us since we won't have any rules in this nsg. Our lab machines are already secure so there is really no point in adding the same rules again here. We are doing this to practice.



Now that these nist 800-53 controls are in place we will again run our environment for 24 hours and see if there is any difference in the number of attacks compared to before.

I am back after roughly 24 hours and now we will look at the results!

	B	C	D
	BEFORE SECURING ENVIRONMENT		
	Start Time	12/4/2023, 5:03:24 PM	
	Stop Time	12/5/2023, 5:03:24 PM	
	Security Events (Windows VMs)	87394	
	Syslog (Linux VMs)	5221	
	SecurityAlert (Microsoft Defender for Cloud)	5	
	SecurityIncident (Sentinel Incidents)	384	
	NSG Inbound Malicious Flows Allowed	3423	
	AFTER SECURING ENVIRONMENT		
	Start Time	12/21/2023, 5:21:59 PM	
	Stop Time	12/22/2023, 5:21:59 PM	
	Security Events (Windows VMs)		
	Syslog (Linux VMs)		
	SecurityAlert (Microsoft Defender for Cloud)		
	SecurityIncident (Sentinel Incidents)		
	NSG Inbound Malicious Flows Allowed		

These are the before numbers and now let's get the after.

We will use these queries to get the information we want:

Security Events (Windows VMs)	SecurityEvent | where TimeGenerated >= ago(24h) | count
Syslog (Linux VMs)	Syslog | where TimeGenerated >= ago(24h) | count
SecurityAlert (Microsoft Defender for Cloud)	SecurityAlert | where DisplayName !startswith "CUSTOM" and DisplayName !startswith "TEST" | where TimeGenerated >= ago(24h) | count
Security Incident (Sentinel Incidents)	SecurityIncident | where TimeGenerated >= ago(24h) | count
NSG Inbound Malicious Flows Allowed	AzureNetworkAnalytics_CL | where FlowType_s == "MaliciousFlow" and AllowedInFlows_d > 0 | where TimeGenerated >= ago(24h) | count
NSG Inbound Malicious Flows Blocked	AzureNetworkAnalytics_CL | where FlowType_s == "MaliciousFlow" and DeniedInFlows_d > 0 | where TimeGenerated >= ago(24h) | count

```
1  Syslog
2  | where TimeGenerated >= ago(24h)
3  | count
4
```

Results | Chart

Count
1

Let’s keep going here is what we have so far:

BEFORE SECURING ENVIRONMENT	
Start Time	12/4/2023, 5:03:24 PM
Stop Time	12/5/2023, 5:03:24 PM
Security Events (Windows VMs)	87394
Syslog (Linux VMs)	5221
SecurityAlert (Microsoft Defender for Cloud)	5
SecurityIncident (Sentinel Incidents)	384
NSG Inbound Malicious Flows Allowed	3423

AFTER SECURING ENVIRONMENT	
Start Time	12/21/2023, 5:21:59 PM
Stop Time	12/22/2023, 5:21:59 PM
Security Events (Windows VMs)	11288
Syslog (Linux VMs)	1
SecurityAlert (Microsoft Defender for Cloud)	
SecurityIncident (Sentinel Incidents)	
NSG Inbound Malicious Flows Allowed	

RESULTS (will auto update, do not edit formulas)	
	Change after security environment
Security Events (Windows VMs)	-87.08%
Syslog (Linux VMs)	-99.98%
SecurityAlert (Microsoft Defender for Cloud)	-100.00%
Security Incident (Sentinel Incidents)	-100.00%
NSG Inbound Malicious Flows Allowed	-100.00%

```
1  SecurityIncident
2  | where TimeGenerated >= ago(24h)
3  | count
```

Results | Chart

Count
0

We can see a HUGE difference already…

And here is the result:

BEFORE SECURING ENVIRONMENT	
Start Time	12/4/2023, 5:03:24 PM
Stop Time	12/5/2023, 5:03:24 PM
Security Events (Windows VMs)	87394
Syslog (Linux VMs)	5221
SecurityAlert (Microsoft Defender for Cloud)	5
SecurityIncident (Sentinel Incidents)	384
NSG Inbound Malicious Flows Allowed	3423

AFTER SECURING ENVIRONMENT	
Start Time	12/21/2023, 5:21:59 PM
Stop Time	12/22/2023, 5:21:59 PM
Security Events (Windows VMs)	11288
Syslog (Linux VMs)	1
SecurityAlert (Microsoft Defender for Cloud)	0
SecurityIncident (Sentinel Incidents)	0
NSG Inbound Malicious Flows Allowed	0

RESULTS (will auto update, do not edit formulas)	
	Change after security environment
Security Events (Windows VMs)	-87.08%
Syslog (Linux VMs)	-99.98%
SecurityAlert (Microsoft Defender for Cloud)	-100.00%
Security Incident (Sentinel Incidents)	-100.00%
NSG Inbound Malicious Flows Allowed	-100.00%

End.