Azure Cloud Detection

Niteesh Deshmukh

Contents

Getting Started

In this project I documented the process of creating a detection lab setup in Azure using Microsoft Sentinel as a SIEM and creating a custom rule to detect scheduled tasks being launched on a system. Before Starting let's get acquainted with some important terms and goals of the project.

What is Detection ?

Detection in the context of cybersecurity is the process of monitoring and analyzing a security environment to find any malicious or abnormal activity or behavior which could compromise the secured environment or network.

What is a SIEM ?

SIEM stands for Security Information and Event Management. It is a security solution which helps organizations to detect, analyze and respond to security threats. SIEM tools collect, aggregate and analyze data in real time from different devices, applications and servers in an organization's network.

Microsoft Sentinel is a cloud based SIEM solution providing capabilities such as security analytics, threat intelligence, threat response in a single platform.

Goals

The goals of this project is to :

- Configure and deploy various Azure resources such as Virtual Machines, Log Analytics and Microsoft Sentinel.
- Implement best security practices for network and virtual machine configuration.
- Implement and utilize data connectors to feed data into Microsoft Sentinel for analysis.
- Understand and configure Windows Event logs and Windows Security Policies.
- Implement and utilize KQL (Kusto Query Language) to query for filter out logs.
- Create custom analytic rules to detect security events

Initial Setup

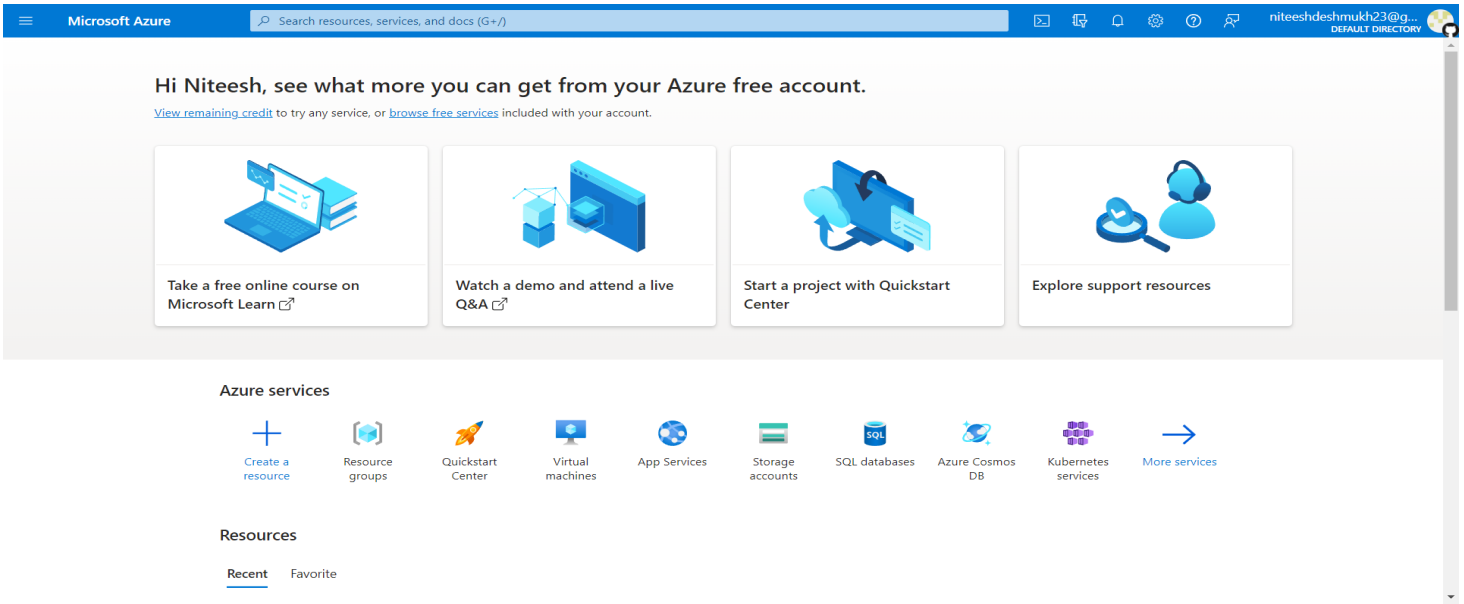
Setting up account

In order to start, an account must be created in Microsoft Azure. I will be using a free tier Azure account which provides a free trial for 30 days.

A free account can be setup here:

https://azure.microsoft.com/en-us/free/

After setting up the account and logging in we will land at Azure portal.



Now we will create a Resource Group.

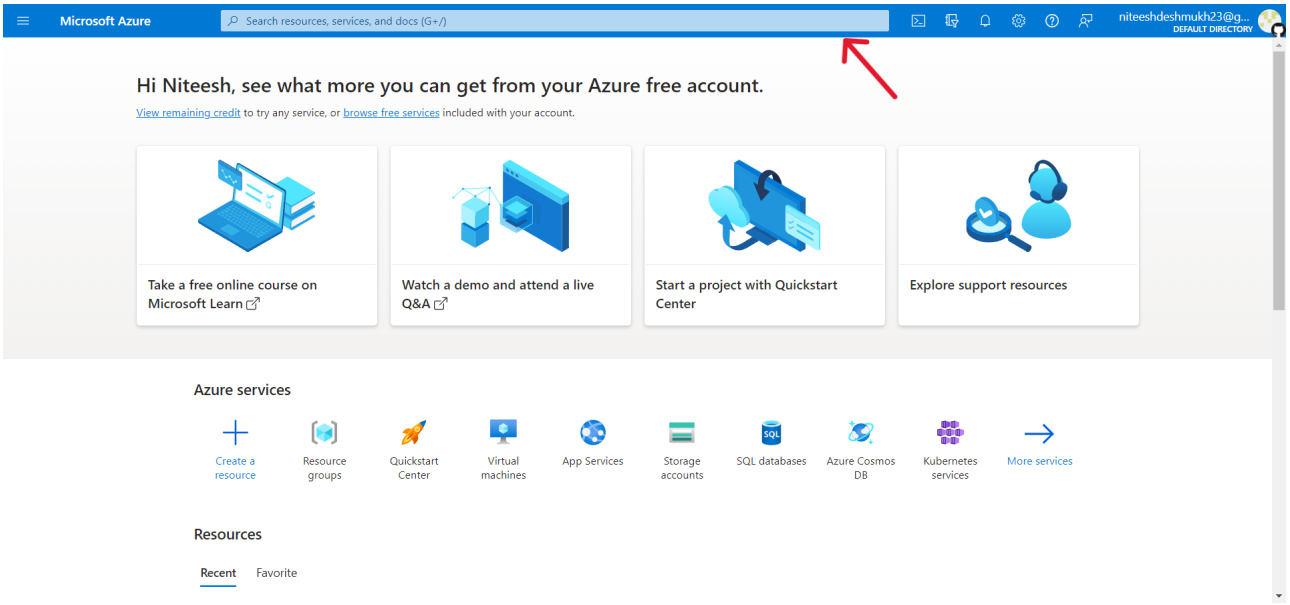
Creating a Resource Group

What is a Resource Group ?

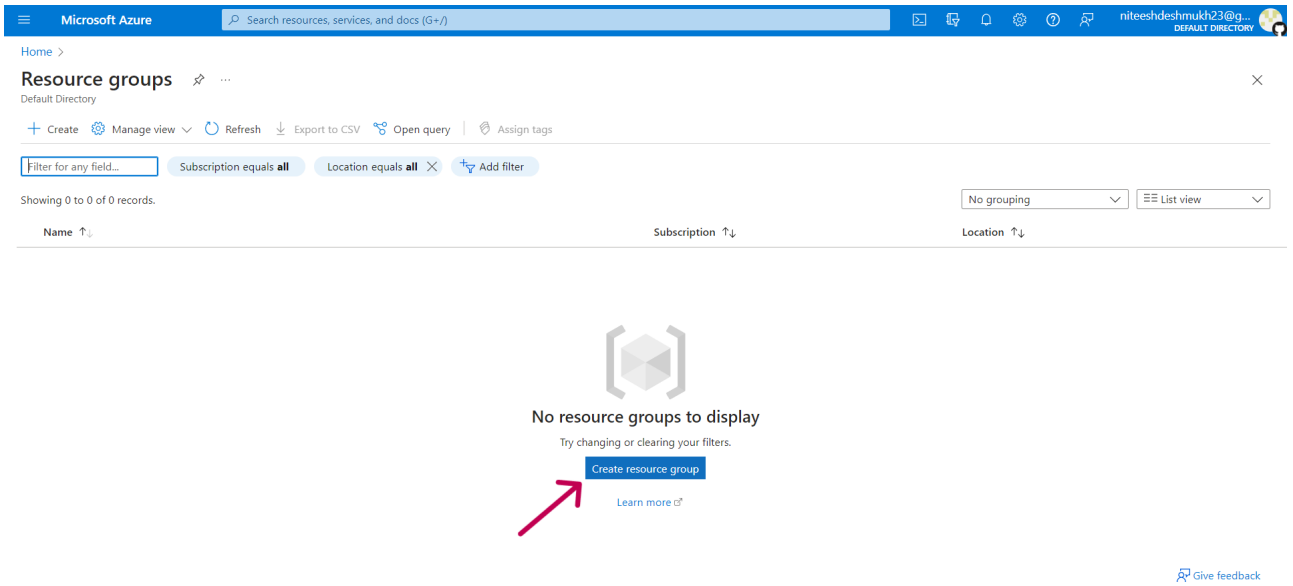
Resource Group is a logical container where you can deploy and manage Azure resources like web apps, databases, and storage accounts. Just as we keep our files and applications inside a folder or directory to manage them efficiently similarly we use a *Resource Group* to manage the Azure resources. We can create multiple *Resource Groups*.

Now we know what a *Resource Group* is, let's see how to create one.

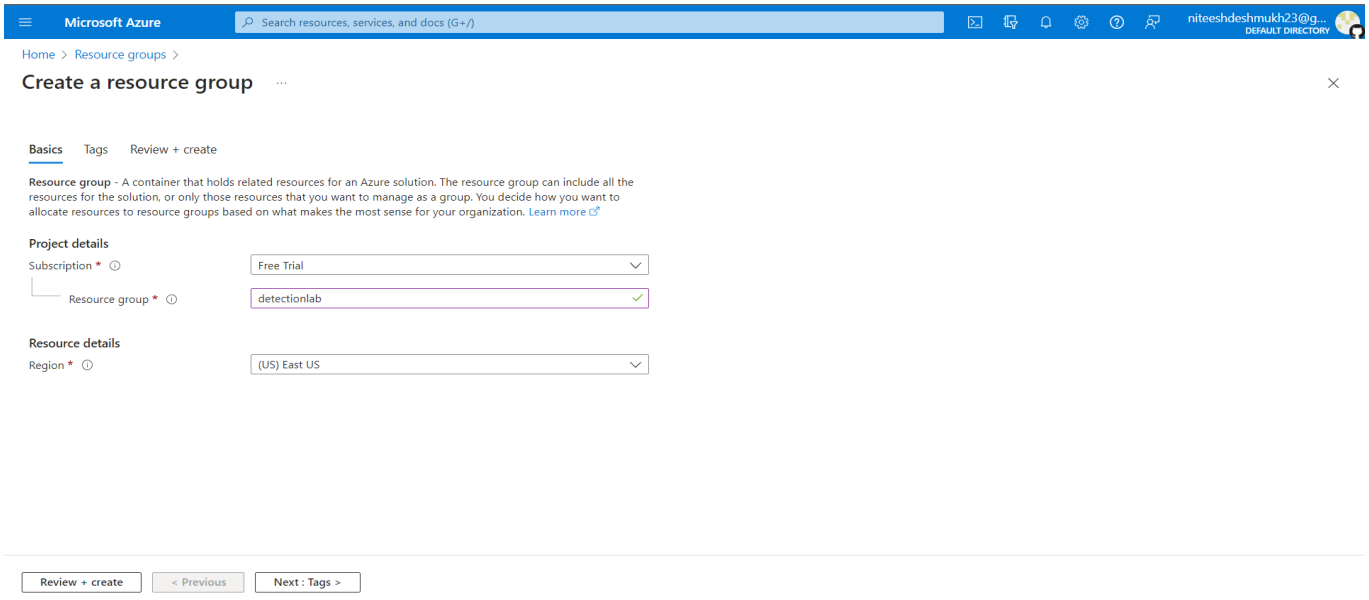
Head back to the Azure portal and search for *Resource Group* in the search bar which is at the top.



Now select *Create Resource Group*.



Fill in the necessary information, skip the *Tags* section and click *Review+Create*.

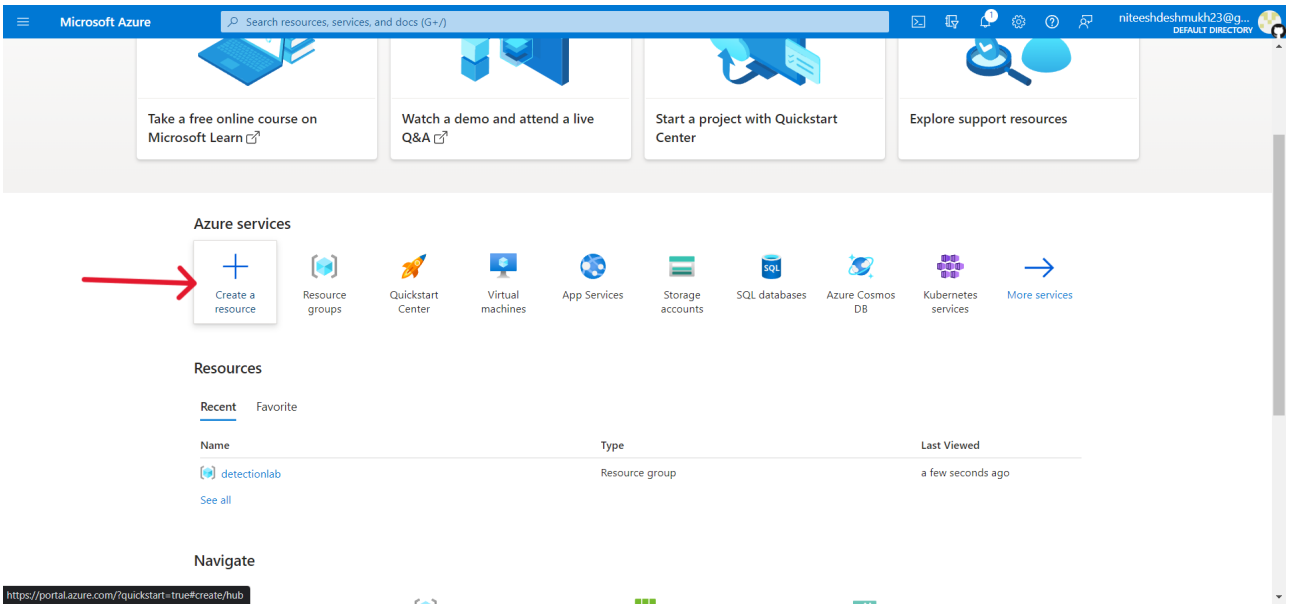


Review the information and click *Create*. We created a *Resource Group*.

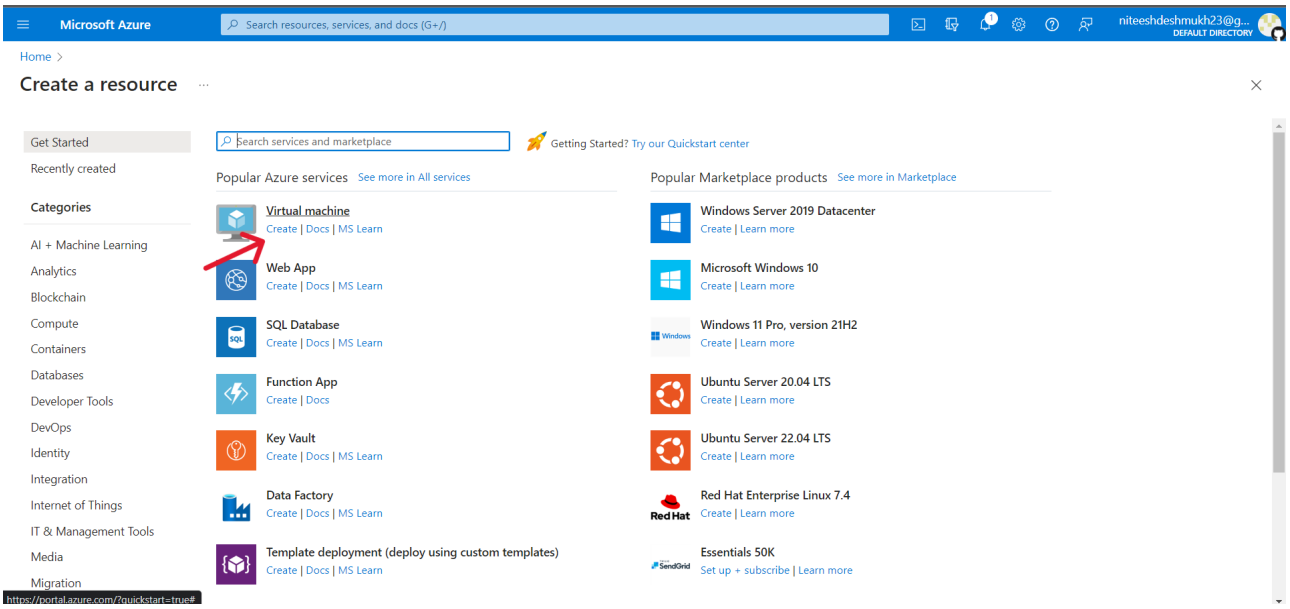
Creating a Virtual Machine

We will create a Windows 10 virtual machine, from where we will collect logs and send them to Microsoft Sentinel for analysis.

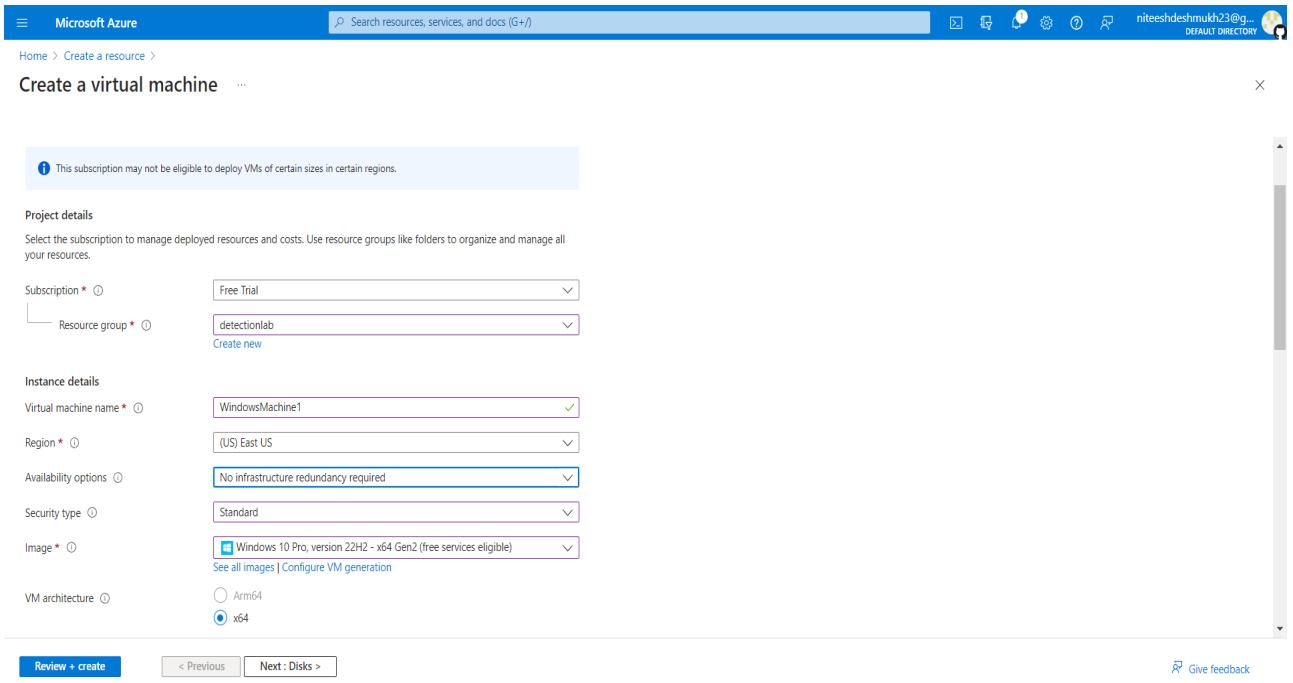
Head over to Azure portal and click on *Create a Resource*.



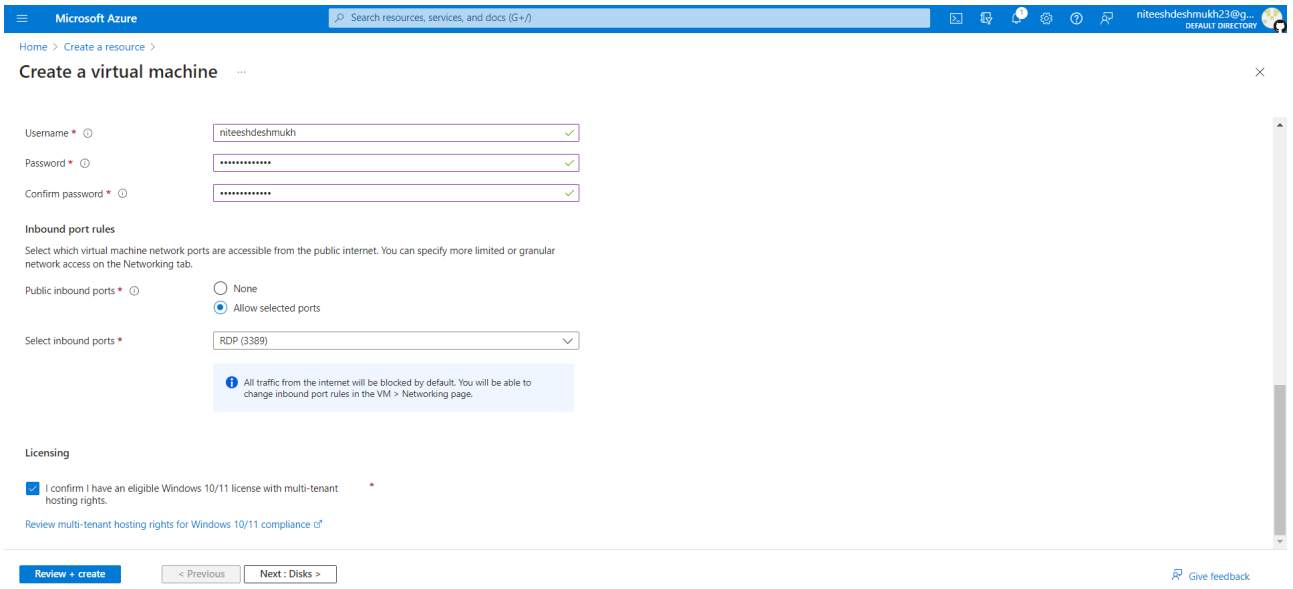
Then click *Create* under *Virtual machine*.



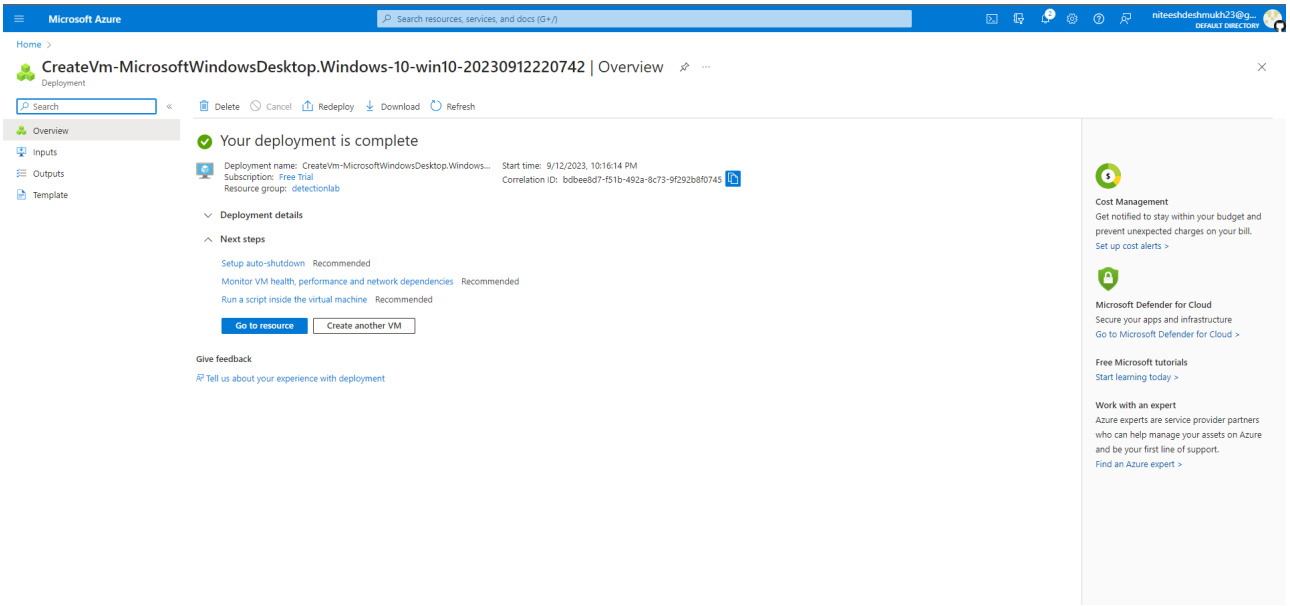
Fill in the details.



Create the username and password for this machine. Configure the *Inbound port rules* as shown. Remember the username and password.



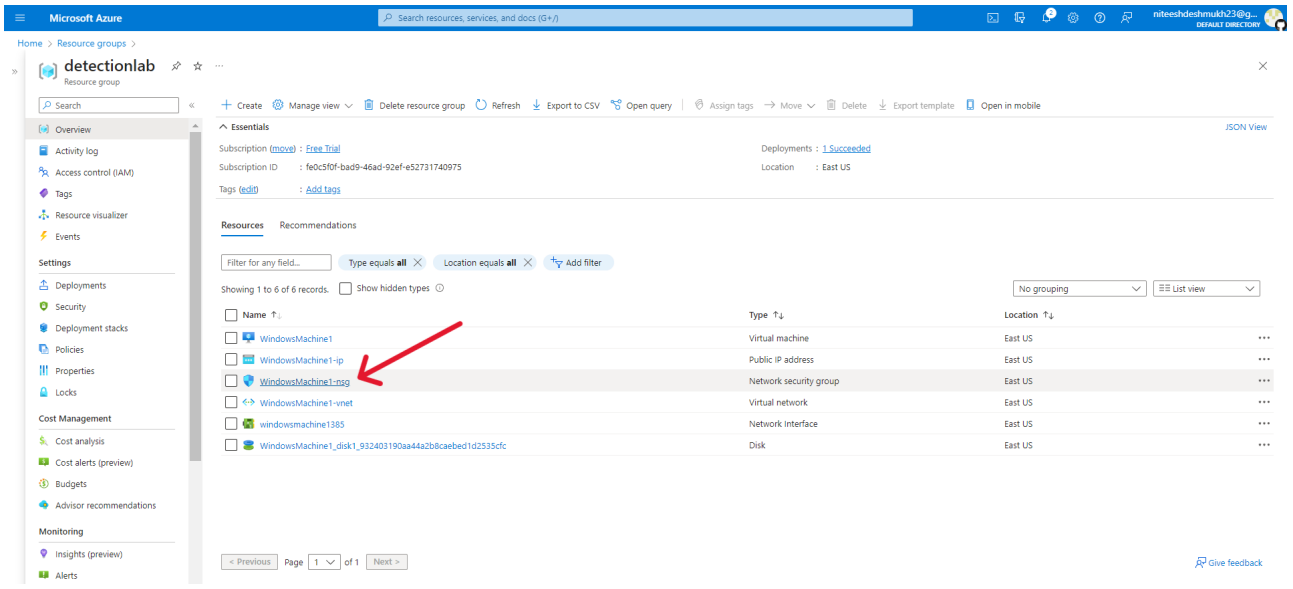
After reviewing click *Create*. The machine will begin to deploy. This can take a few minutes. After successful deployment, the following will be displayed.



After the successful deployment, our virtual machine is now placed in a virtual network. It gets assigned with a network interface and private and public IP addresses; moreover another security feature known as Network Security Group (NSG) also gets implemented.

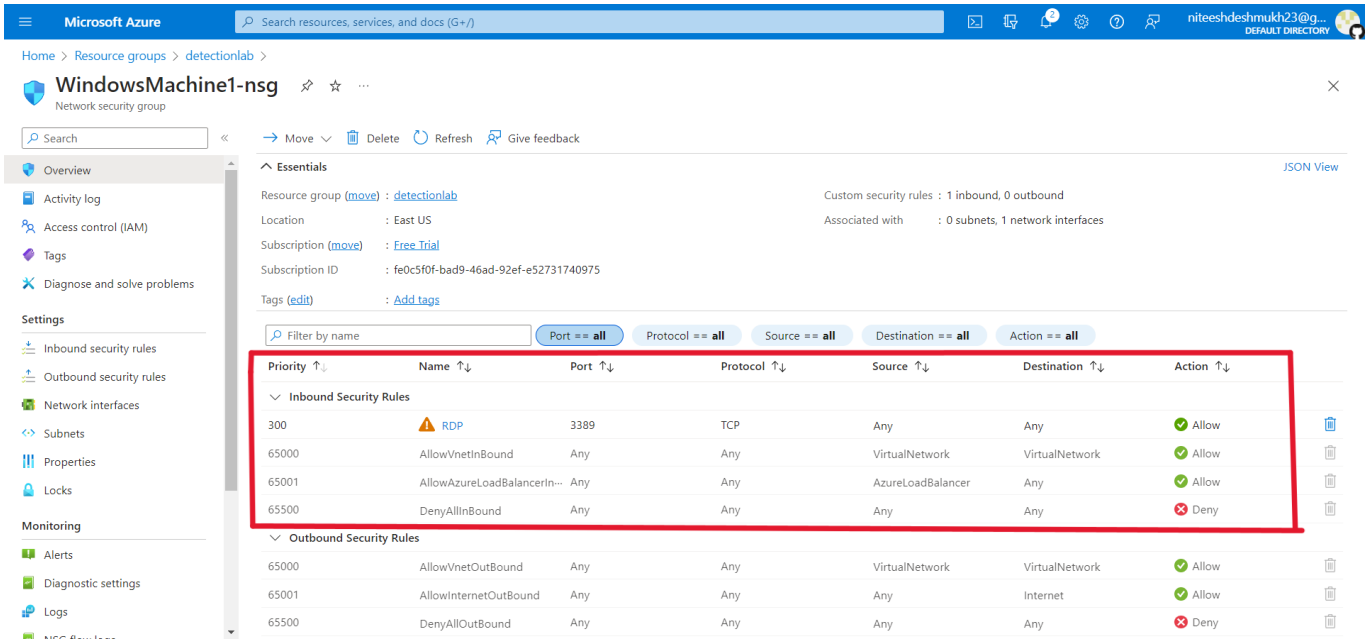
An NSG is used to filter network traffic to and from Azure resources. Similar to a firewall, filtering is based on rules that dictate source and destination ports and network protocols that are allowed or denied.

We can navigate to the NSG from the *Resource Group*.



It can be observed from the *Inbound port rules* that anyone can try to connect to the virtual machine from the internet as the RDP port 3389 is configured to accept incoming RDP traffic from anyone.

This can leave our machine vulnerable to brute force attacks or password spray attacks from across the internet.



Configuring Just In Time access (JIT)

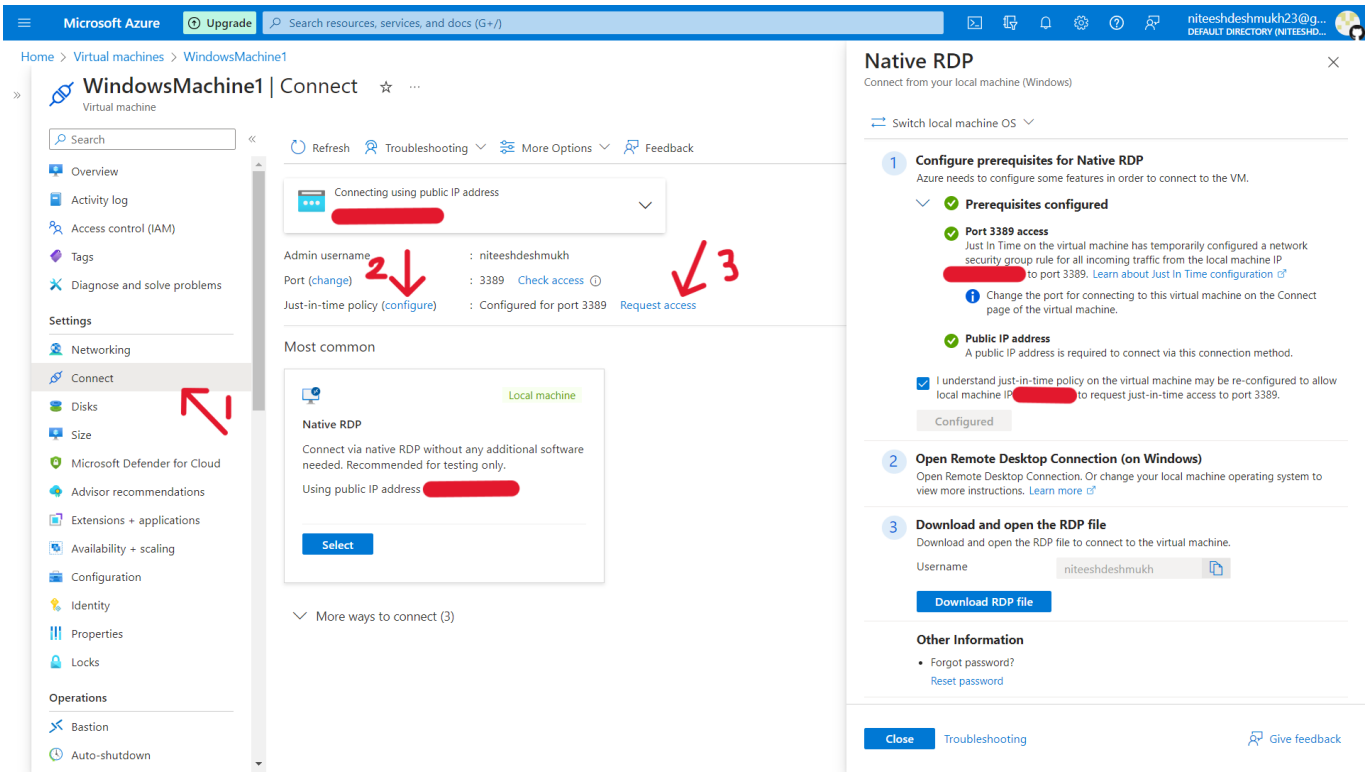
We saw that our machine is vulnerable to attacks. There is a security feature known as Just In Time access which comes under *Microsoft Defender for Cloud*. We can implement JIT access to reduce our attack surface.

What is Just In Time access ?

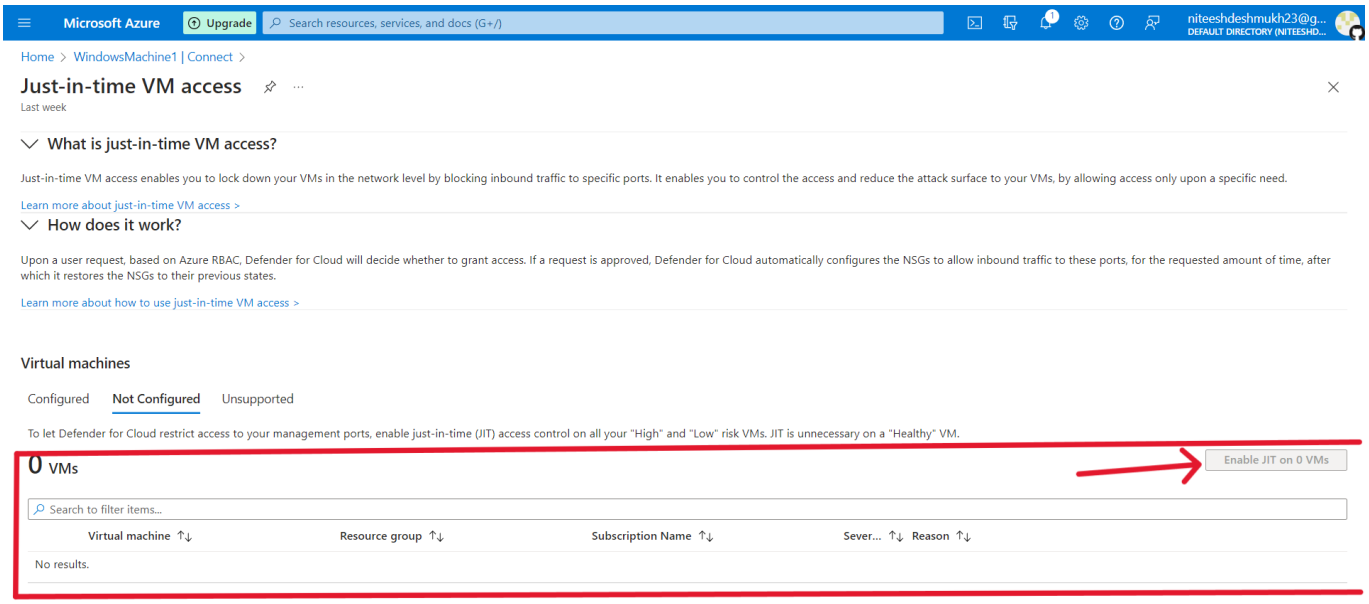
Microsoft Defender for Cloud's just-in-time (JIT) access protects Azure virtual machines (VMs) from unauthorized network access. Many times firewalls contain rules that leave the virtual machines vulnerable to attack. JIT allows access to the VMs only when the access is needed, on the ports needed, and for the period of time needed.

Head to the virtual machine tab.

1. Go to the *Connect* section.
2. Select *configure*. A new window will open. This is the JIT window. Select the virtual machine and click *Enable JIT*.
3. Select *Request Access*. This may take up to a few seconds or minutes. If nothing happens , just reload and repeat the steps.



The JIT window in step 2 will be like this.



In the *Not Configured* tab it is showing 0 because I already configured the JIT in my case. But in your case the virtual machine will be displayed. Select it and enable JIT.

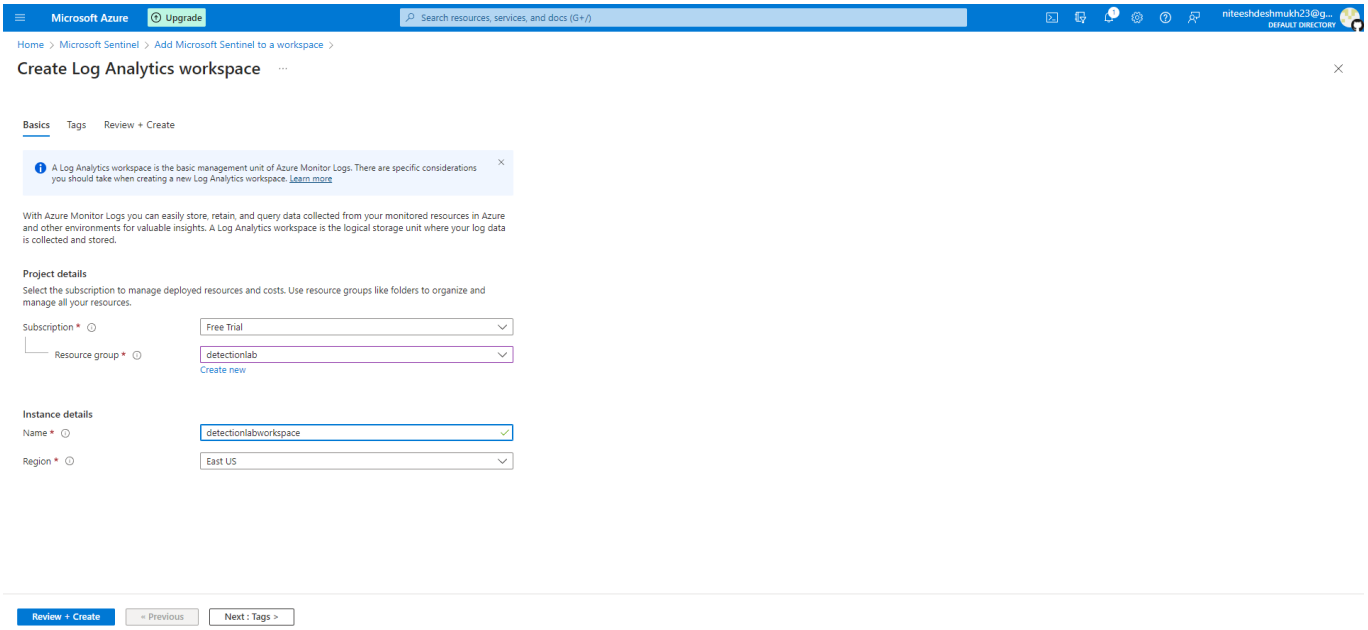
Creating Log Analytics Workspace

What is Log Analytics Workspace?

When working with Log Data in Azure we need somewhere to store/operate that data. Log Analytics workspace is used to collect and store log data from Azure Resources. A Log Analytics workspace is a unique environment for log data from Azure services, such as Microsoft Sentinel and Microsoft Defender for Cloud.

To configure Log Analytics Workspace, search Microsoft Sentinel in the search bar in Azure Portal, the following window will appear.

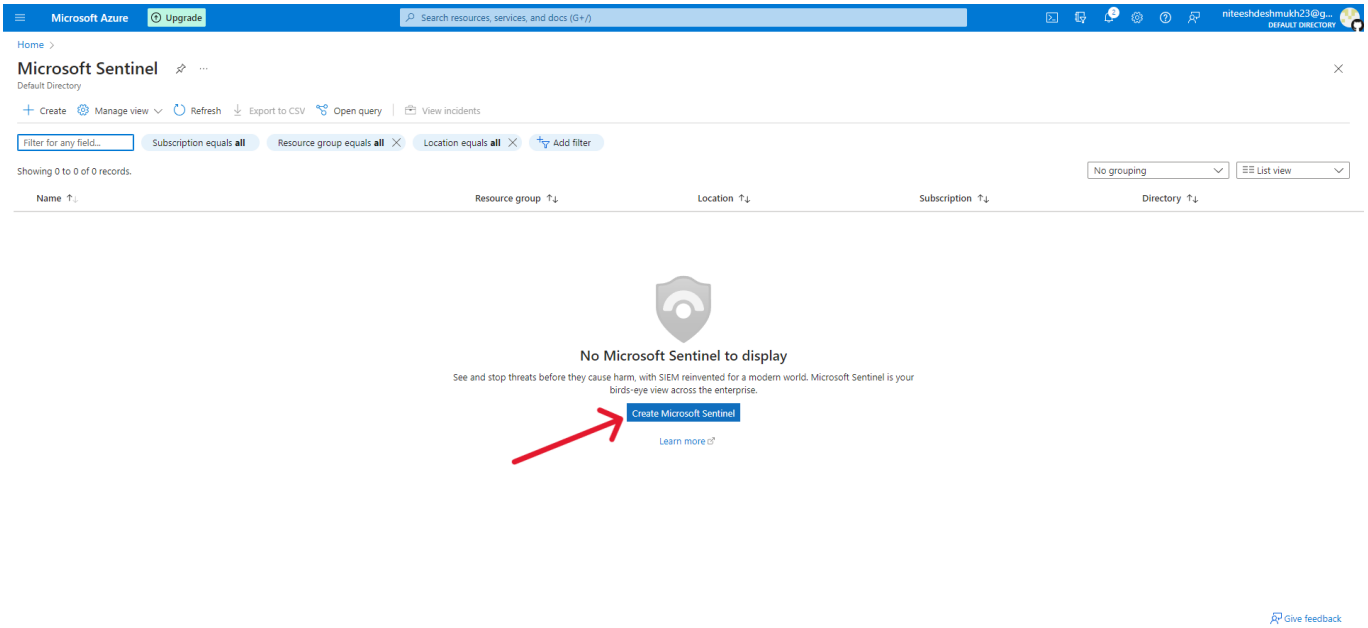
Fill in the required information.



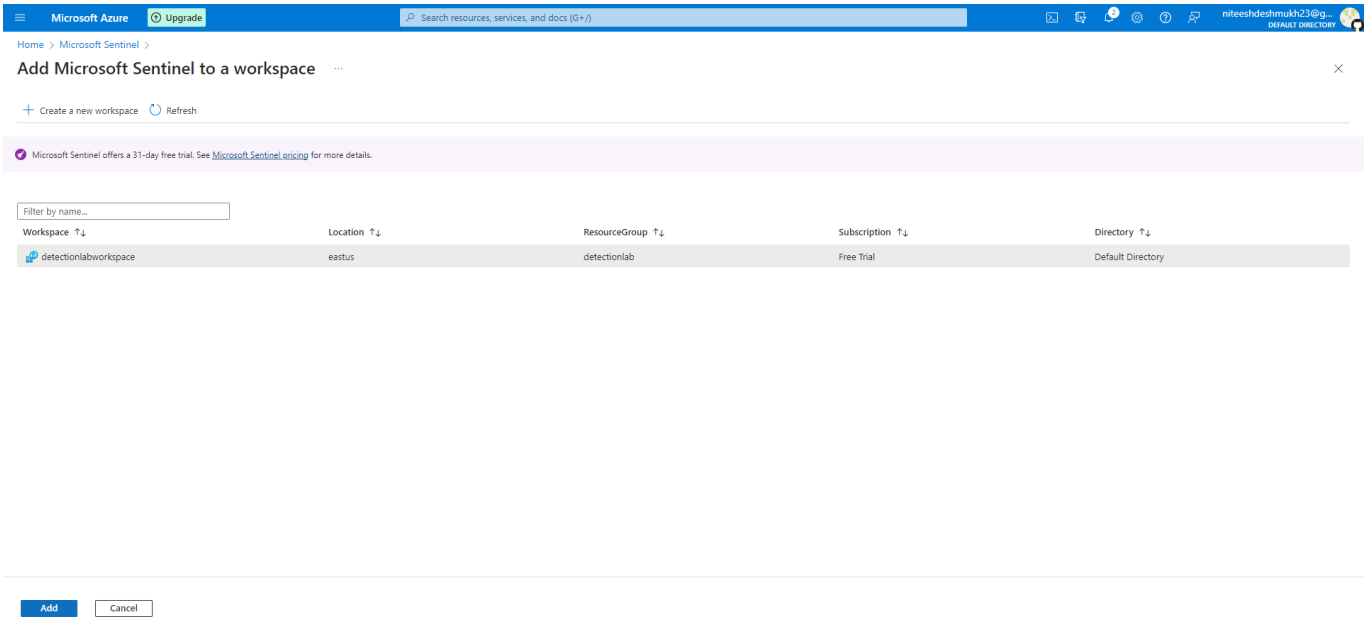
Click *Review and Create*. Workspace will be created.

Now we need to add Microsoft Sentinel to this workspace.

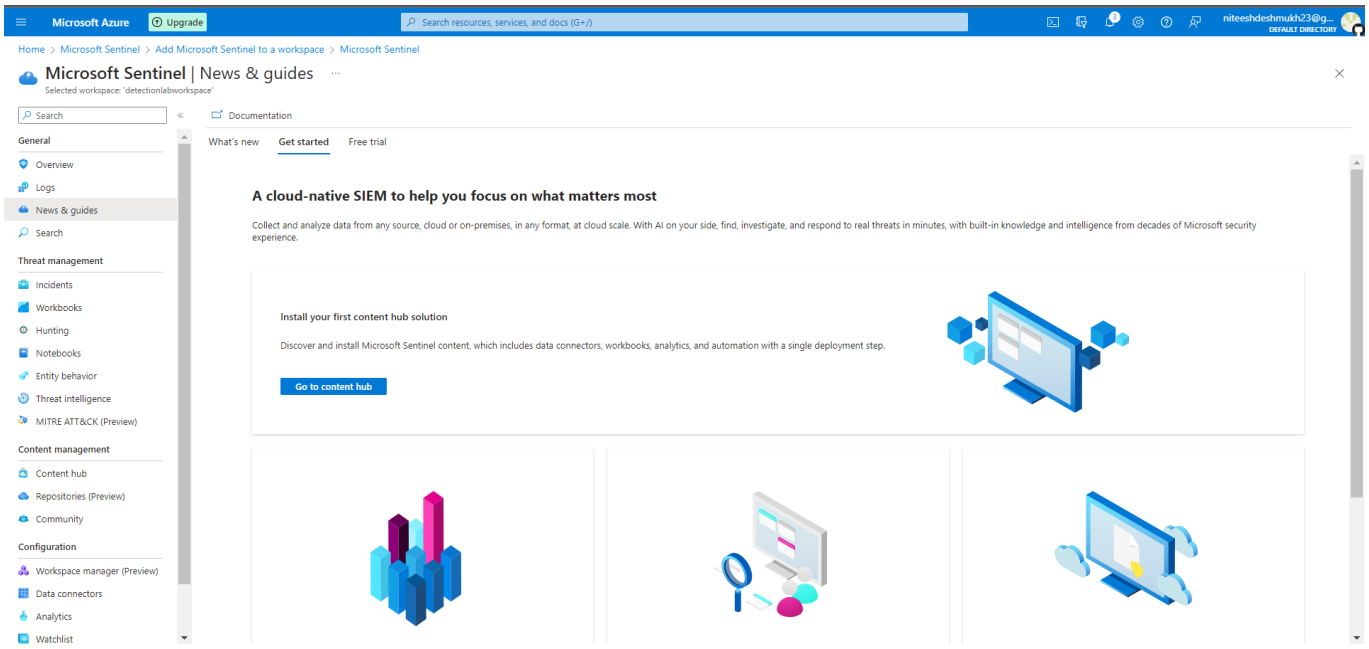
Search again for Microsoft Sentinel and click *Create Microsoft Sentinel*.



Select the workspace and click *Add*.



Following window shall open.



We have successfully deployed Microsoft Sentinel.

But in order to analyze logs we must somehow get the logs from the Windows 10 virtual machine and send them to Microsoft Sentinel.

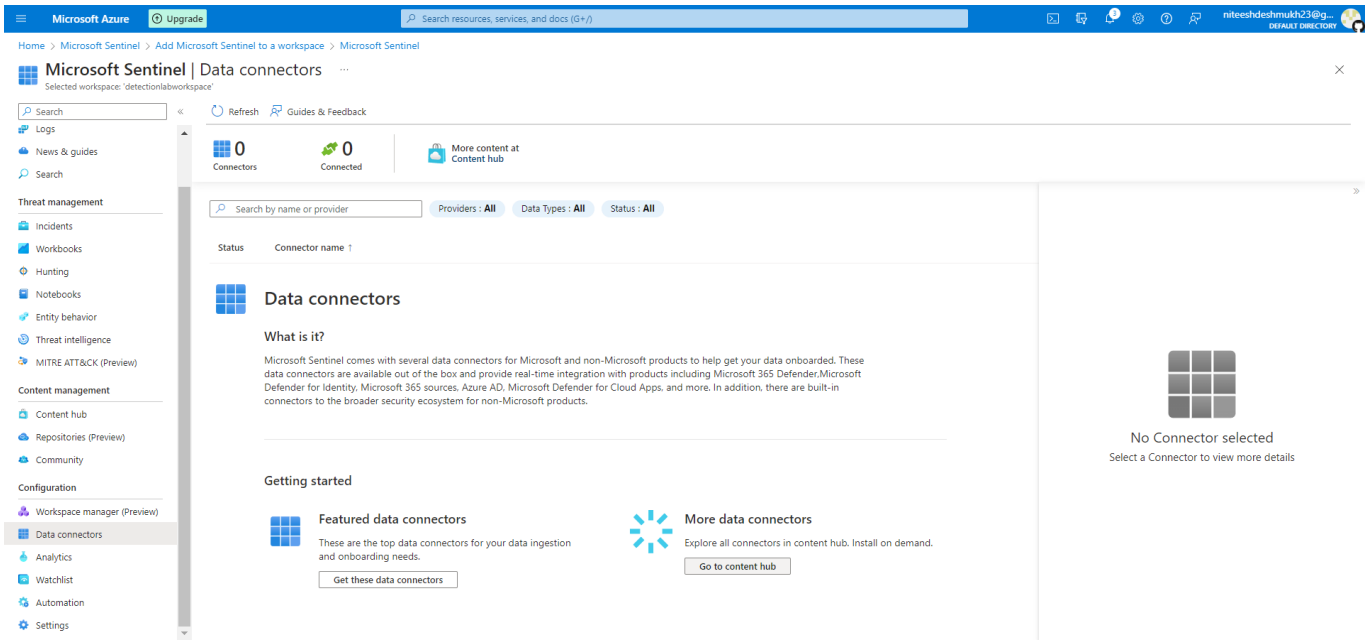
In order to do so we will use *Data Connectors*.

Adding Data Connectors

What are Data Connectors ?

Data connectors help Microsoft Sentinel to ingest data from various sources. Data collection rules are utilized to specify the data to be ingested.

Go to the *Data Connectors* section in the Microsoft Sentinel.

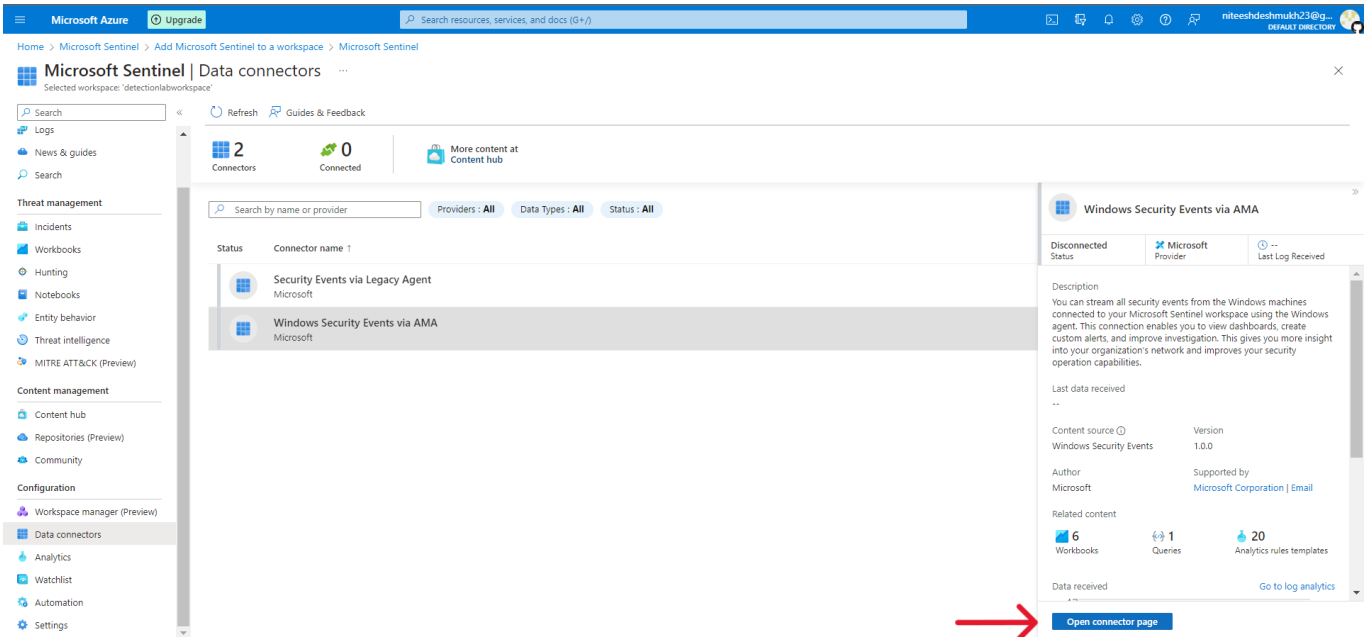


It can be observed that there are currently no data connectors installed.

We will install *Windows Security Event* connector to enable Sentinel to ingest log data from Windows 10 virtual machine.

Go to the *Content Hub* menu and search for *Windows Security Event* and install it.

The package will contain two connectors and after successful installation you will see the following screen.



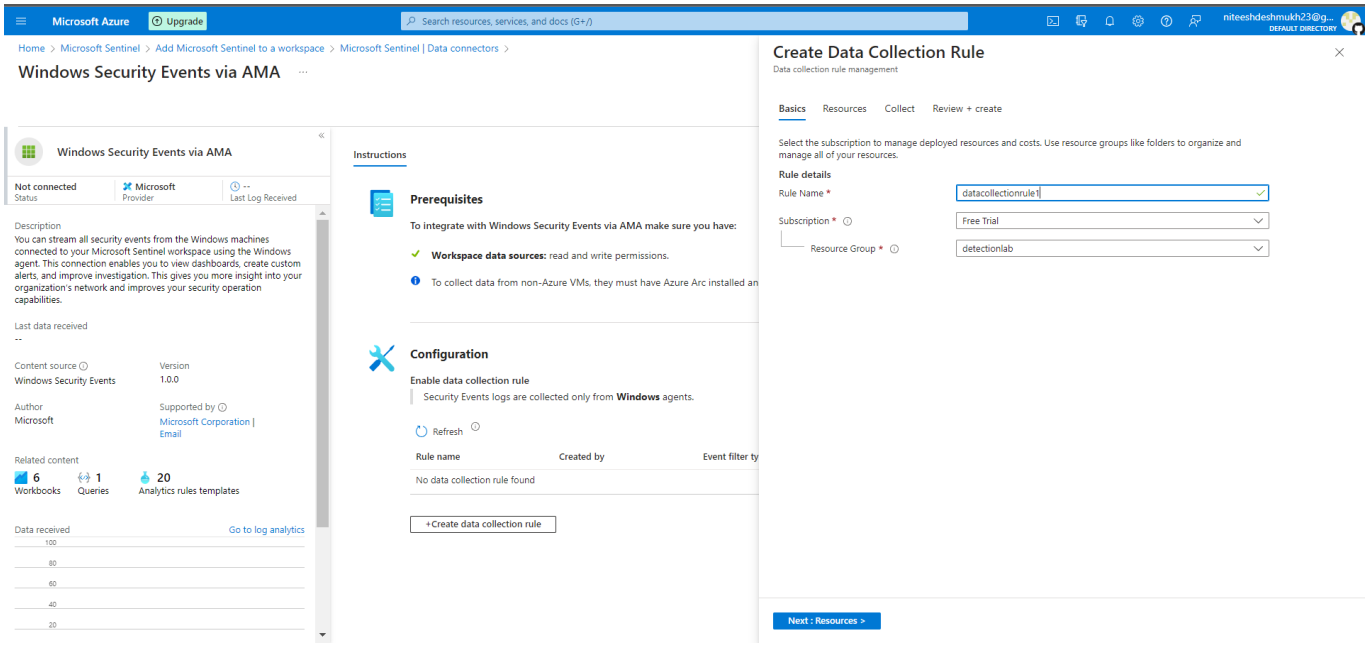
Now we will create a Data Collection Rule.

In the *Data Connector* select *Windows Security via AMA*.

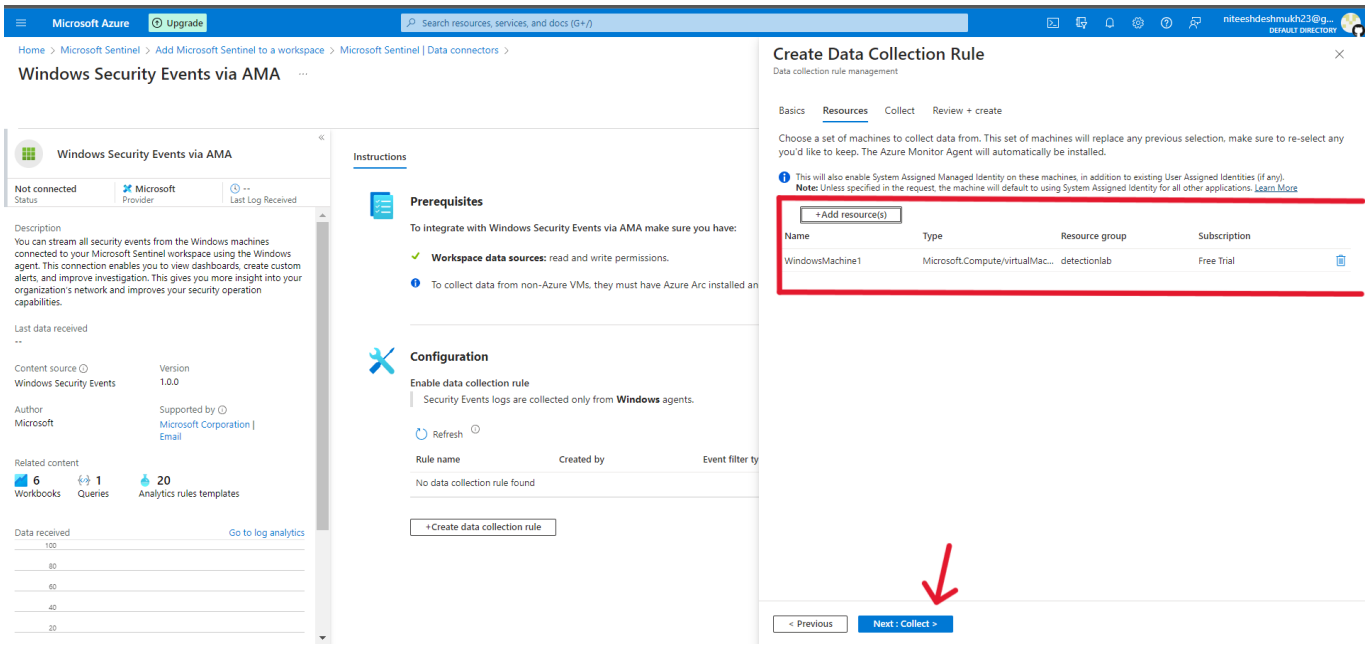
Brief description of the connector will be shown on the right.

Select *Open connector page* as shown in the image above and a new window will open with a detailed description of the connector.

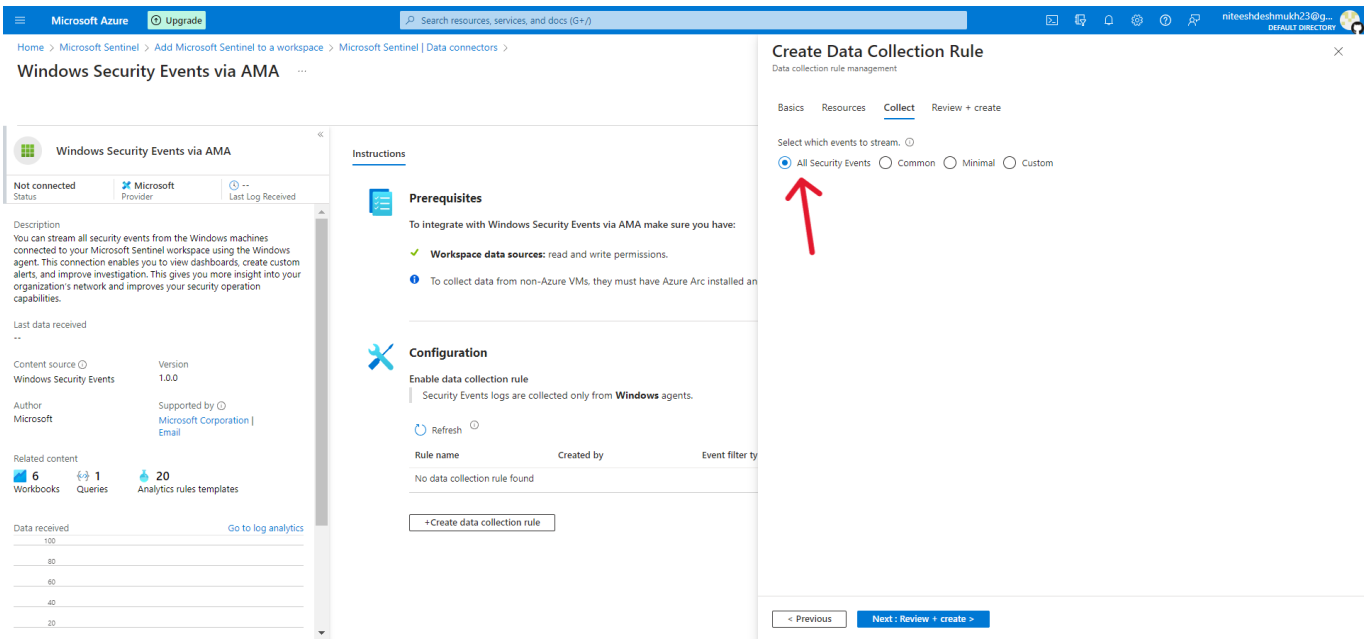
We will get this window. On the right side fill in the details and click *Resources*.



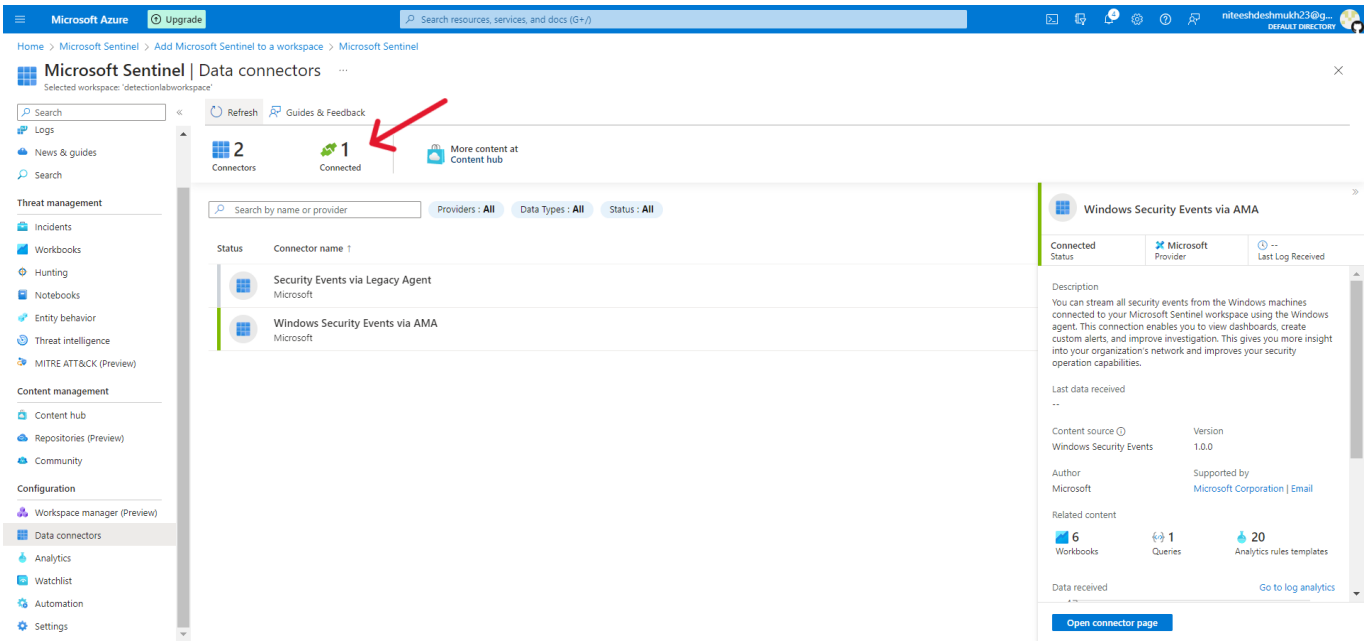
In the *Resources* click on *Add Resources* and add the Windows 10 virtual machine and then move to the *Collect* tab.



In the *Collect* tab select *All security events*. And then click *Review+Create*.



After successful creation the Data Connector will look like this. Here it shows that a connector is successfully connected.



Generating Security Events

Now that our VM is connected to Sentinel and our Log Analytics Workspace we need to transport data from our Logs. To do this we need to simply need to perform some action on the Windows 10 events that will generate security alerts.

Windows keeps a record of several types of security events. These events cover several potential scenarios such as privileged use, Logon events, processes, policy changes, and much more.

We will now observe some Windows security events on our Virtual Machine.

Utilize the Azure Portal to navigate to the VM created earlier in the lab.

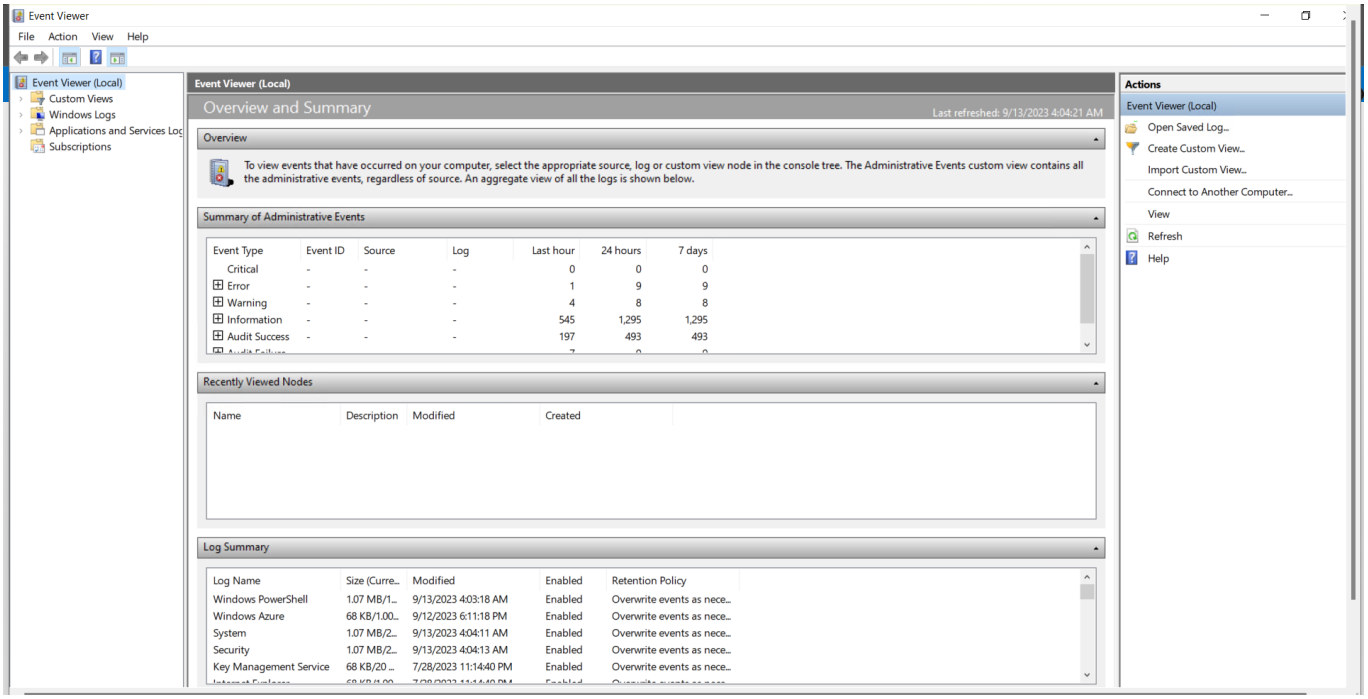
Click “Start” at the top page to turn on the VM if it’s not on already . Enable Just in time Access if necessary.

Under Networking, you are given a public IP. Use an RDP on your PC Client such as Remote Desktop Connection to access your VM by entering in the public IP address.

(you might need to refresh after starting the virtual machine to have the public IP show up).

From here you will be prompted to enter the username and password created when you made the VM.

Once you successfully authenticate to the virtual machine and are logged in, search for Event Viewer and open the program.



There are several types of logs Windows Collects. Application logs, Security Logs, Setup, System, and Forwarded Events.

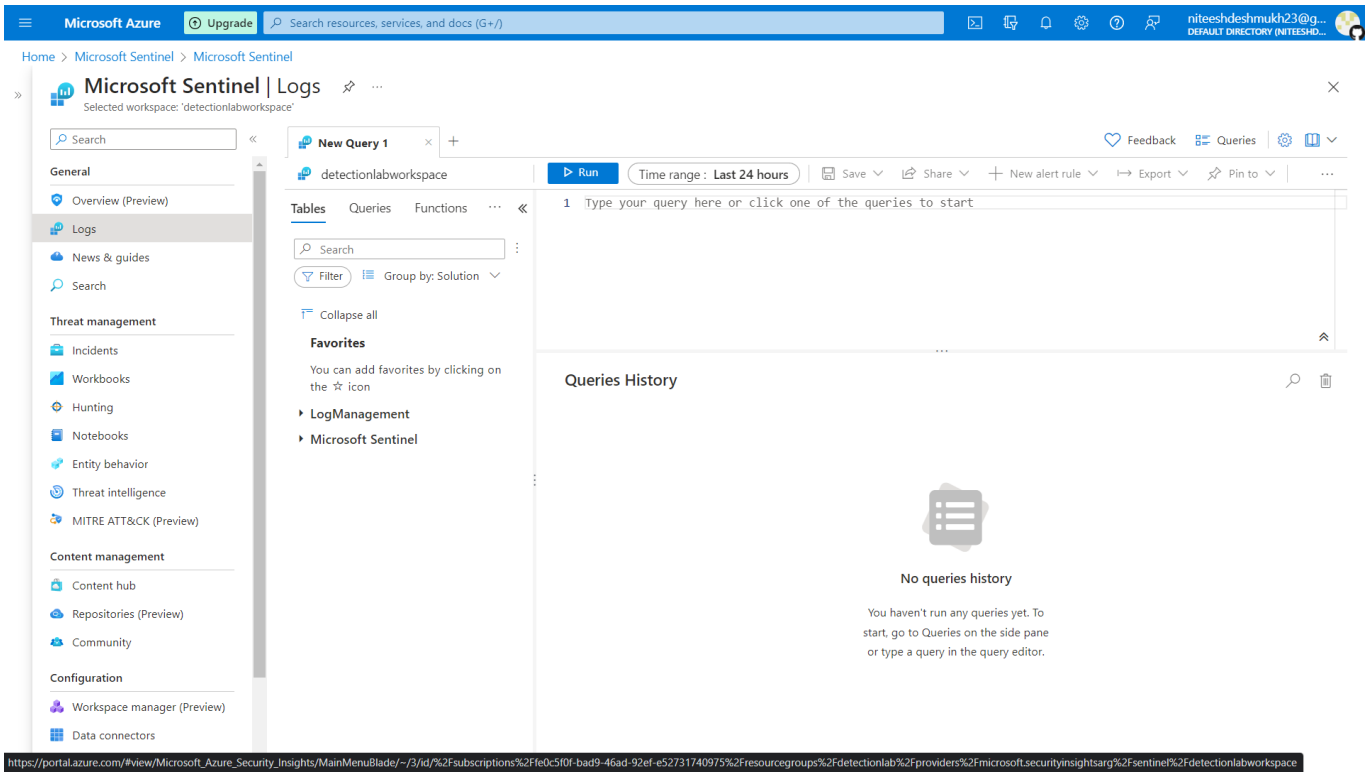
We will focus on a specific security log which indicates successful logon into the system. It has an id assigned to it which is 4624.

Utilizing Kusto Query Language(KQL)

What is KQL ?

Kusto Query Language (KQL) is a powerful tool to explore data and discover patterns, identify anomalies and outliers, create statistical modeling, and more. The query uses schema entities that are organized in a hierarchy similar to SQLs: databases, tables, and columns. KQL can be used to pull specific logs.

Head over to the *Logs* section in Microsoft Sentinel. It should look like this.



We will now write a KQL query to pull some logs.

In the *Type your query* section write the following query.

SecurityEvent | where EventID == 4628 | project Computer, TimeGenerated, AccountName

Let's break down the meaning of this query

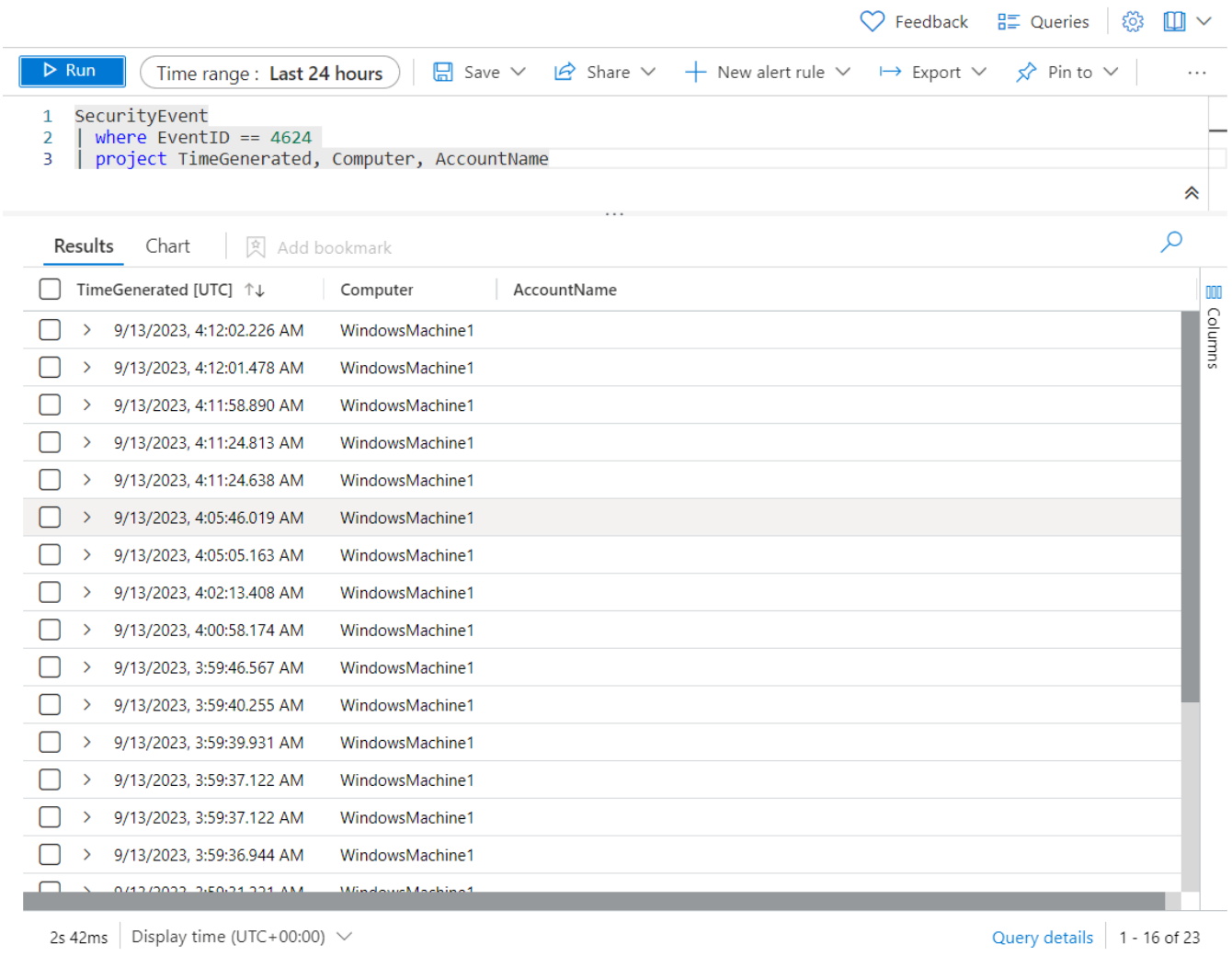
SecurityEvent refers to the event table we are pulling the data from. All the events we observed in the event viewer are stored there.

Where command filters on a specific category. In this case, we only want events that correspond to successful logins.

Project command will specify what data to display when the query is run so, in this specific scenario, we want to only see the time the logon event occurred, what computer it came from and what account on this computer-generated the event.

Note that every SIEM has a search language that makes it simple to extract data from Logs. In Sentinel, that language is called KQL or Kusto Query Language.

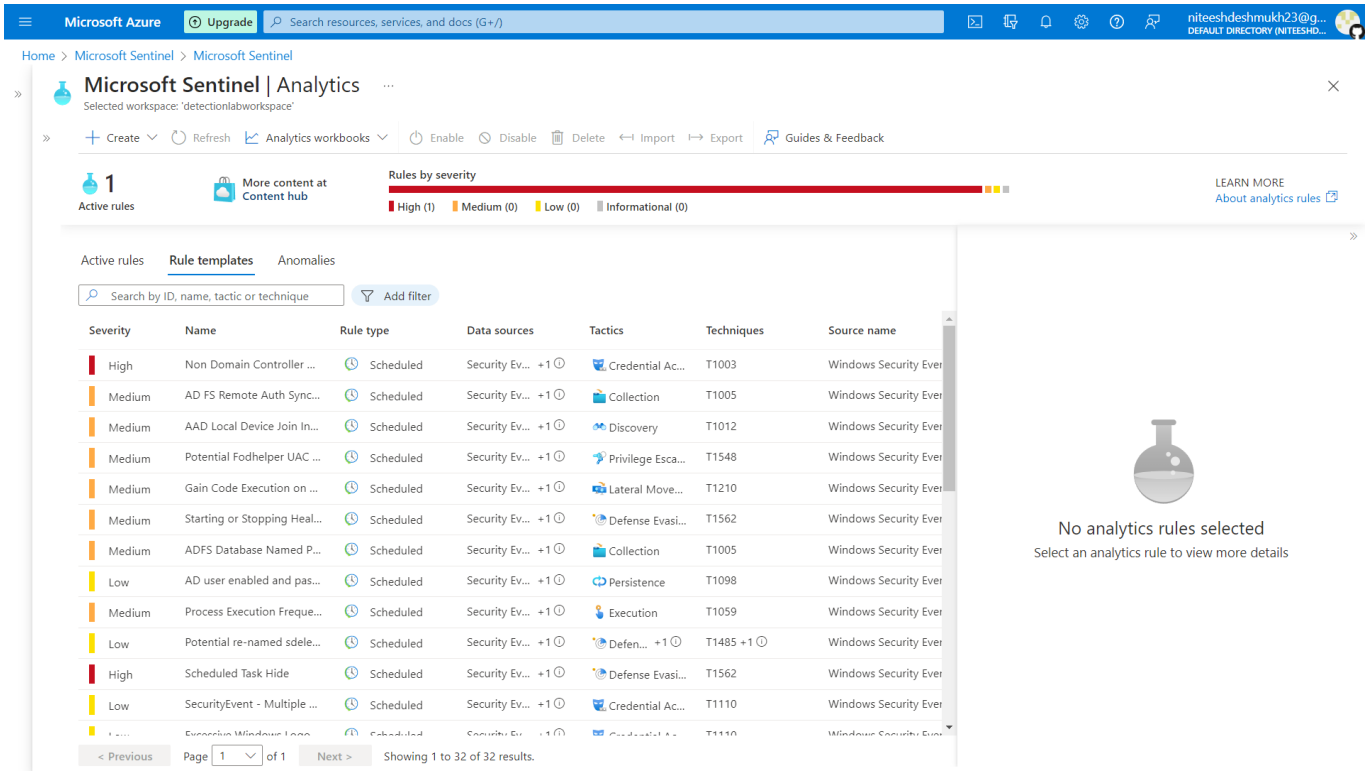
When the query is run we get this result.



We have a list of all the times we have had a successful login on our VM. However, as you can see the Account Name field is empty as Sentinel is not automatically putting that data into that field. We will go over how to populate that field later.

Writing Analytic Rule and Creating Scheduled Task

We can have the option to be alerted to certain events by setting up analytic rules. The Analytic rule will check our VM for the activity that matches the rule logic and generate an alert any time that activity is observed. There will be some details provided in the alert that can help an analyst start their investigation into determining whether the event in the alert is a false positive or true positive.



These are a list of alerts that we can enable our SIEM to monitor that come out of the box. If you wish, you can expand some and see what they are comprised of by clicking create a rule and

following the onscreen tasks to see the rule logic and as well as enabling the rule. However, the rules will only fire if the logic is met by a security event on your VM.

We will create our own custom rule to detect potentially malicious activity on our VM. In the Windows Task scheduler you have the option to create a scheduled task. A scheduled task is essentially a way to automate certain activities on your machine .

For instance, you could set up a scheduled task that opens google chrome at a certain time every day. While many times scheduled tasks can be a harmless event this can also be used as a persistence technique for malicious actors.

According to the MITRE Attack Framework, “Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time”.

In this project, our scheduled task will not be associated with any malicious activity as we will set up a scheduled task that opens Internet Explorer at a certain time but we will create an analytic rule

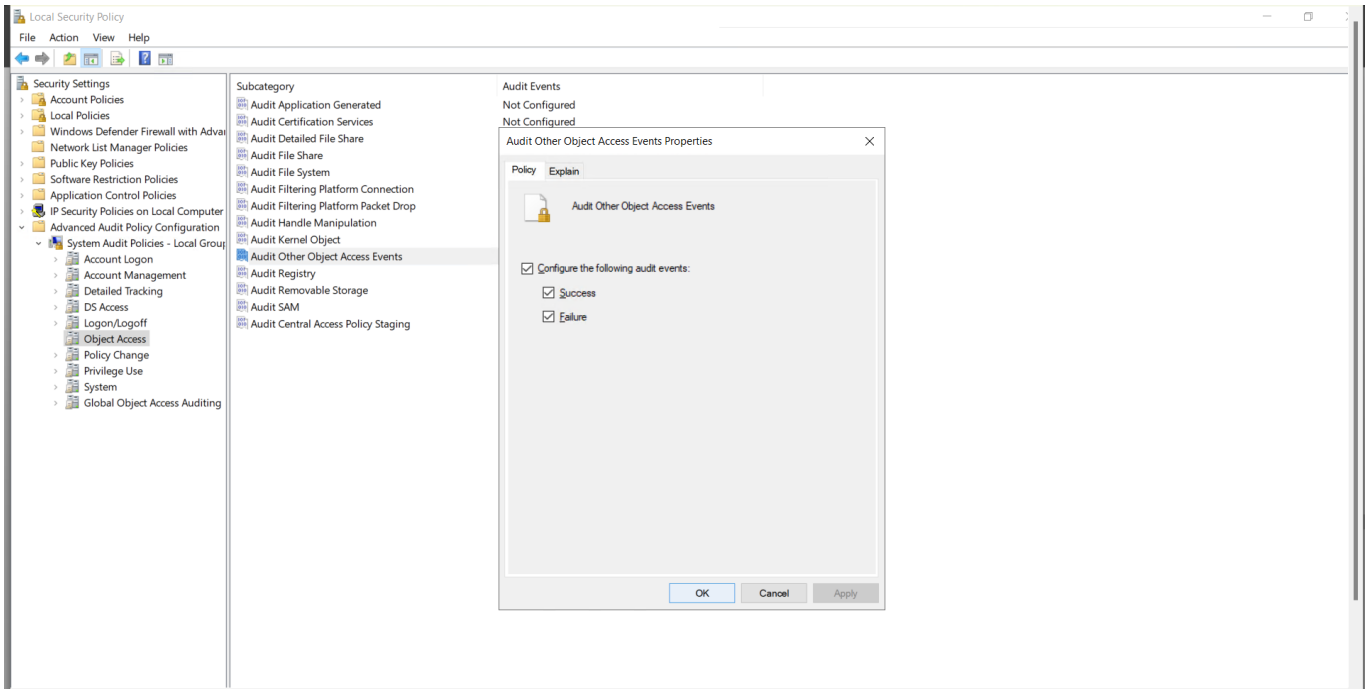
to monitor for that specific action so we can be alerted to the to the activity in our SIEM in order to simulate the scenario.

The Windows Security Event ID that corresponds to scheduled task creation is 4698. However, these events are not logged by default in the Windows event viewer. To enable logging for this event we need to make some changes to the Windows Security policy in our VM.

Search for *Local Security Policy* in Windows 10 VM and expand *Advanced Audit Policy Configuration*.

Expand *System Audit Policies* and *Select Object Access*. Then select the *Audit Other Object Access*.

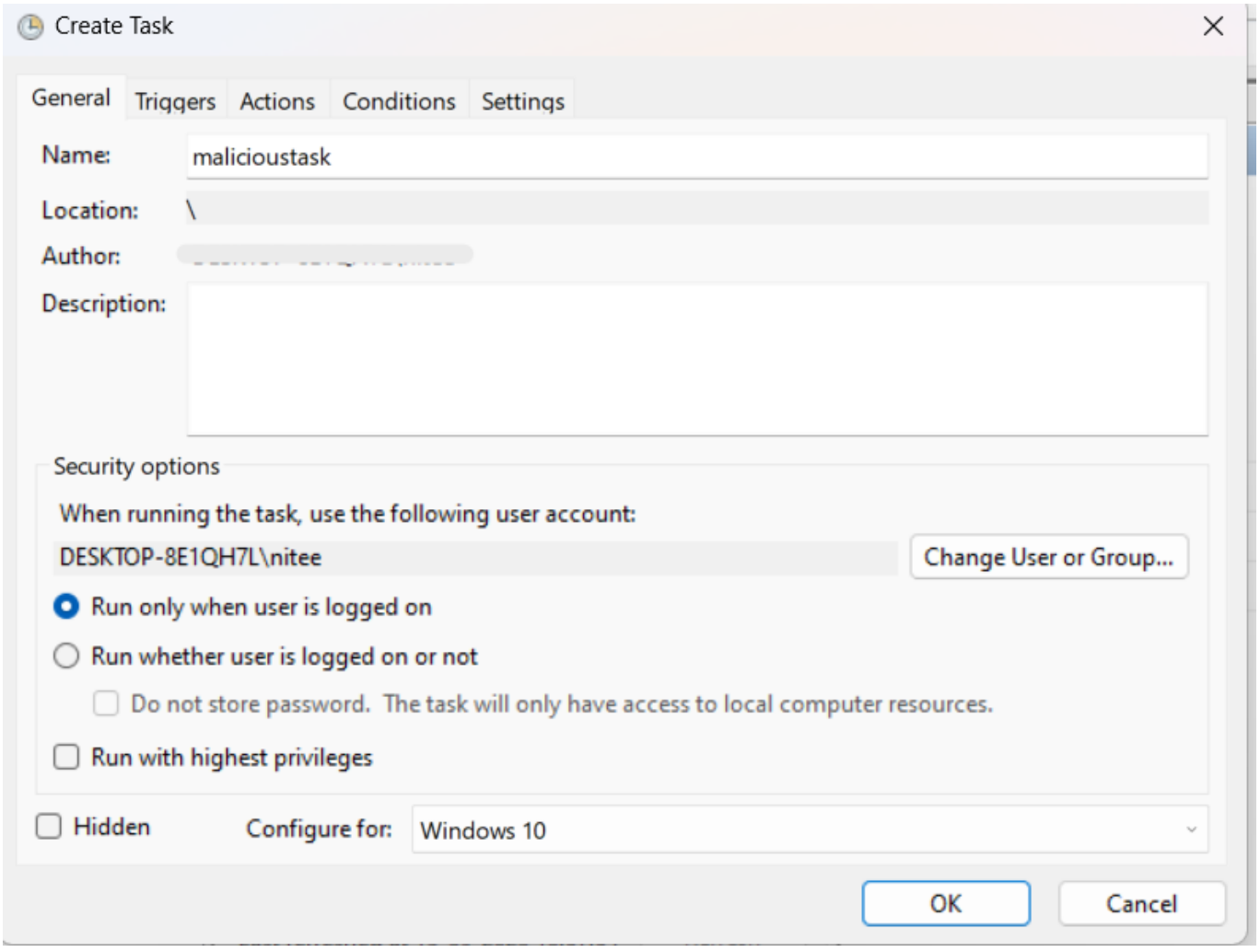
Enable *Success* and *Failure*.



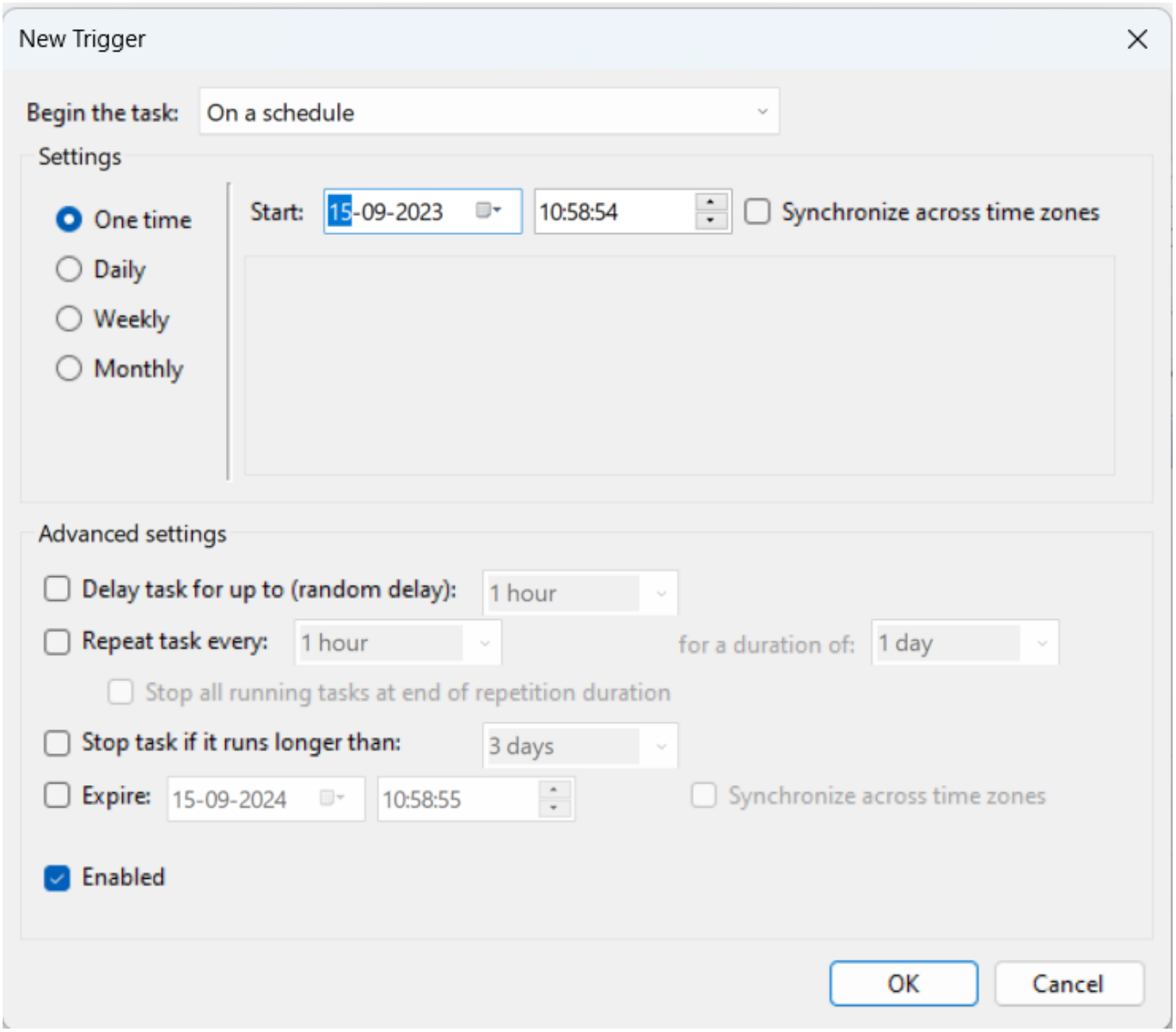
Logging is now enabled for the scheduled task event.

To detect a scheduled task creation, we need to generate some activity in our VM.

Open Windows Task Scheduler and navigate to “Create Task” . Add a name and change the “Configure For” Operating system to Windows 10.

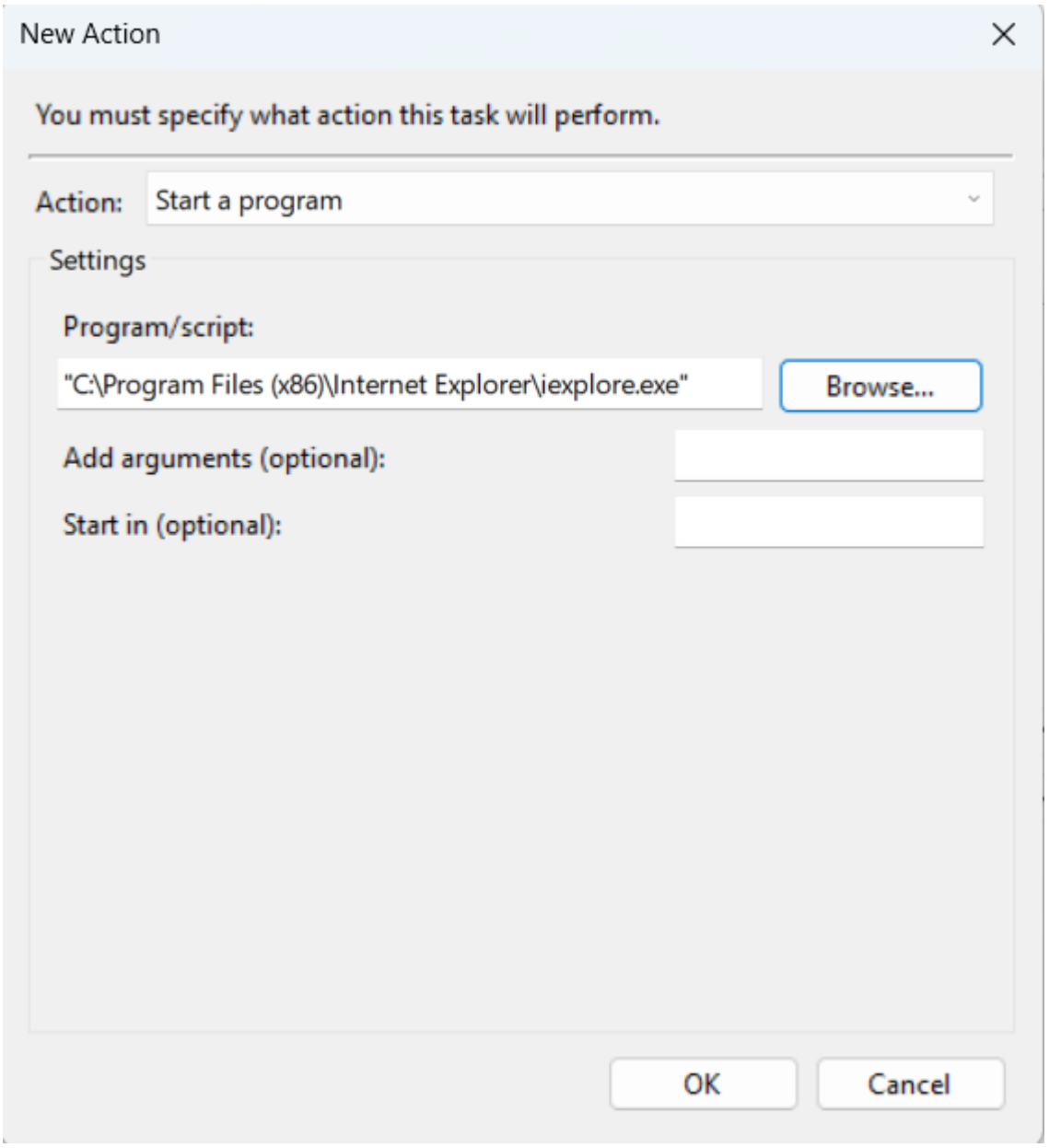


Navigate to triggers and click “new” and schedule the task for a time close to your current time. Then select “OK”.



Navigate to the action tap and select start a program.

Then open a program or script and select a program to run every time this task runs. I will select Internet Explorer.

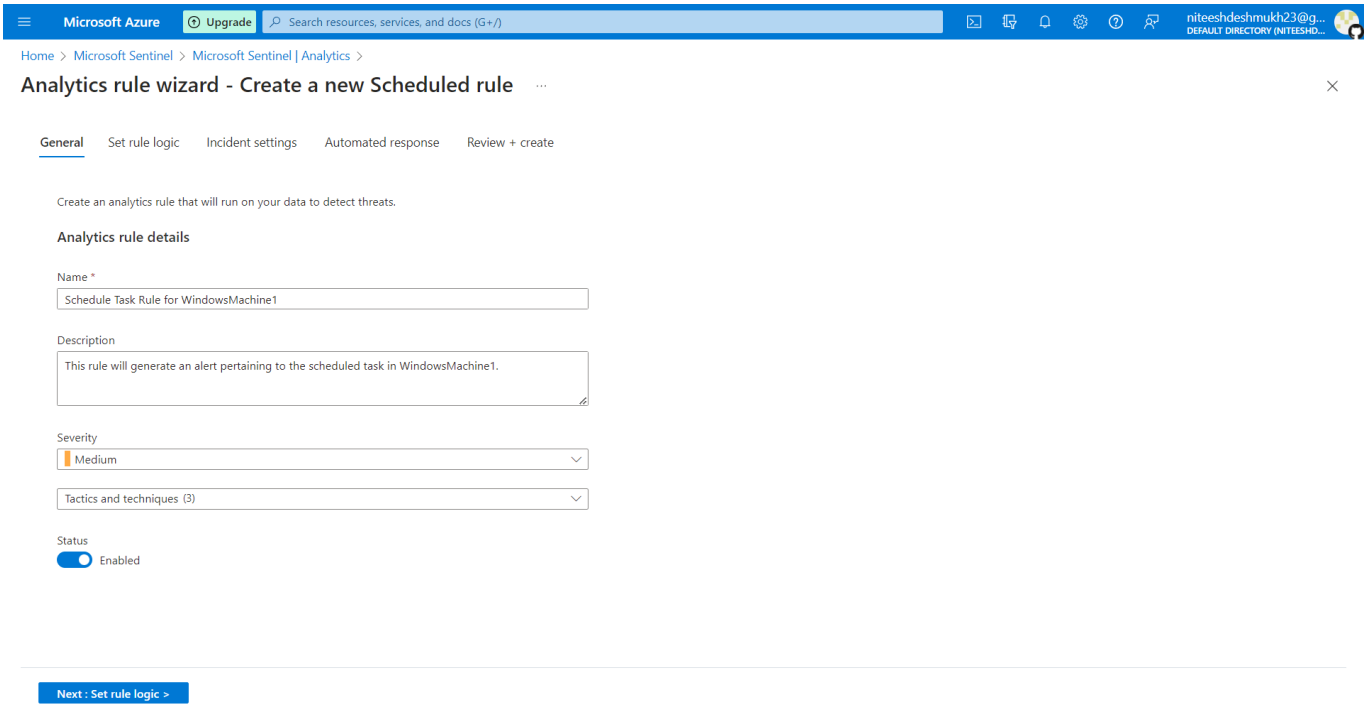


Skip the conditions and settings and click “OK”.

This will create your scheduled task and you can now go to event viewer and search for that event id 4698 in the security logs.

We need to write some KQL logic to alert us when a scheduled task is created.

Go to the sentinel Home Page and click “Analytics Rules” and click create at the top of the page and select the scheduled query option.



Next, we will come up with the alert logic that causes our alert to fire. Most of the logic will be like the KQL Query that we created earlier for logon event.

```
SecurityEvent | where EventID == 4698
```

This query will pull instances of scheduled tasks.

If we want to display the information pulled in a more visually pleasing way we can use the following query.

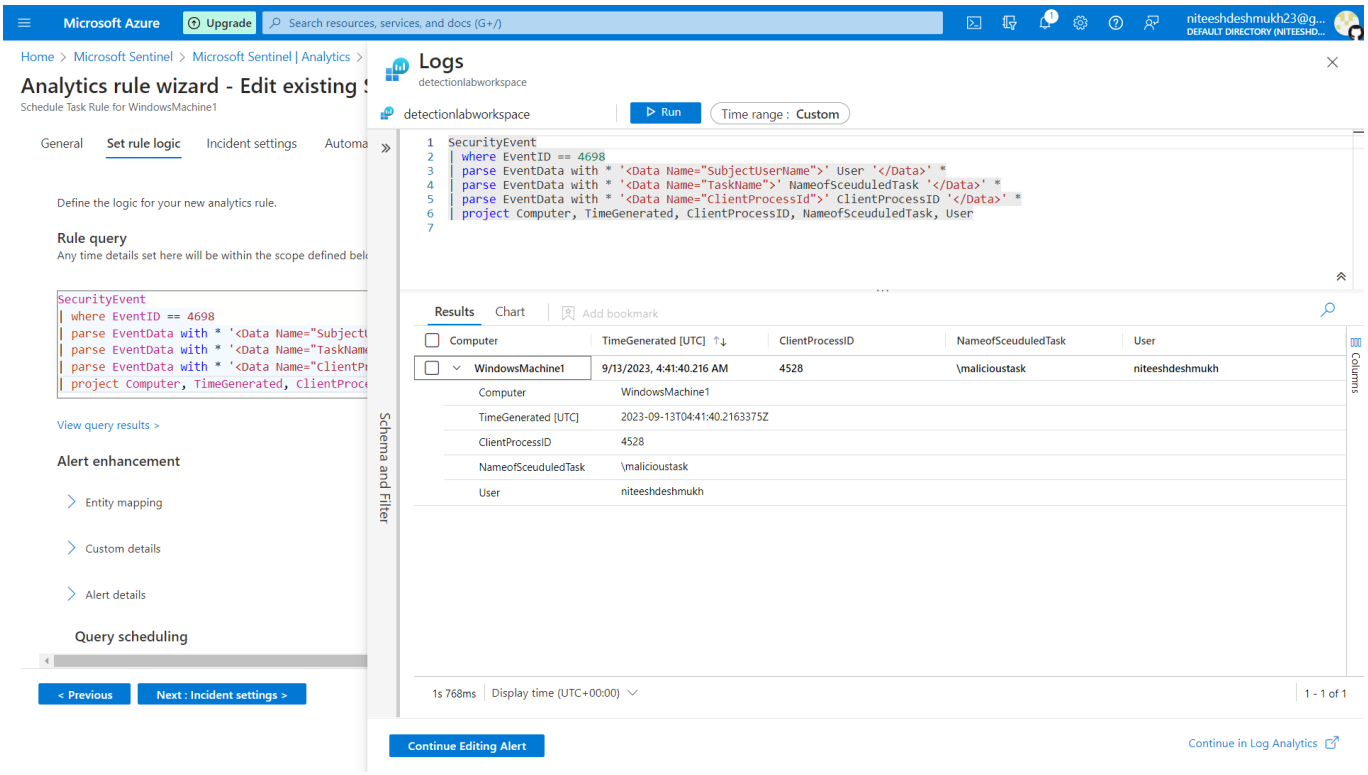
```
SecurityEvent
 | where EventID == 4698
 | parse EventData with * '<Data Name="SubjectUserName">' User '</Data>' *
| parse EventData with * '<Data Name="TaskName">' NameofSceuduledTask '</Data>' *
 | parse EventData with * '<Data Name="ClientProcessId">' ClientProcessID '</Data>' *
 | project Computer, TimeGenerated, ClientProcessID, NameofSceuduledTask, User
```

The **parse** command will allow us to extract data from the Event Data Field that we find important.

This extracted the SubjectUserName , TaskName, ClientProcessID (Computer automatically displays) .

The above logic allows us to assign those to new categories such as User, NameofScheduledTask, and ClientProcessID respectively.

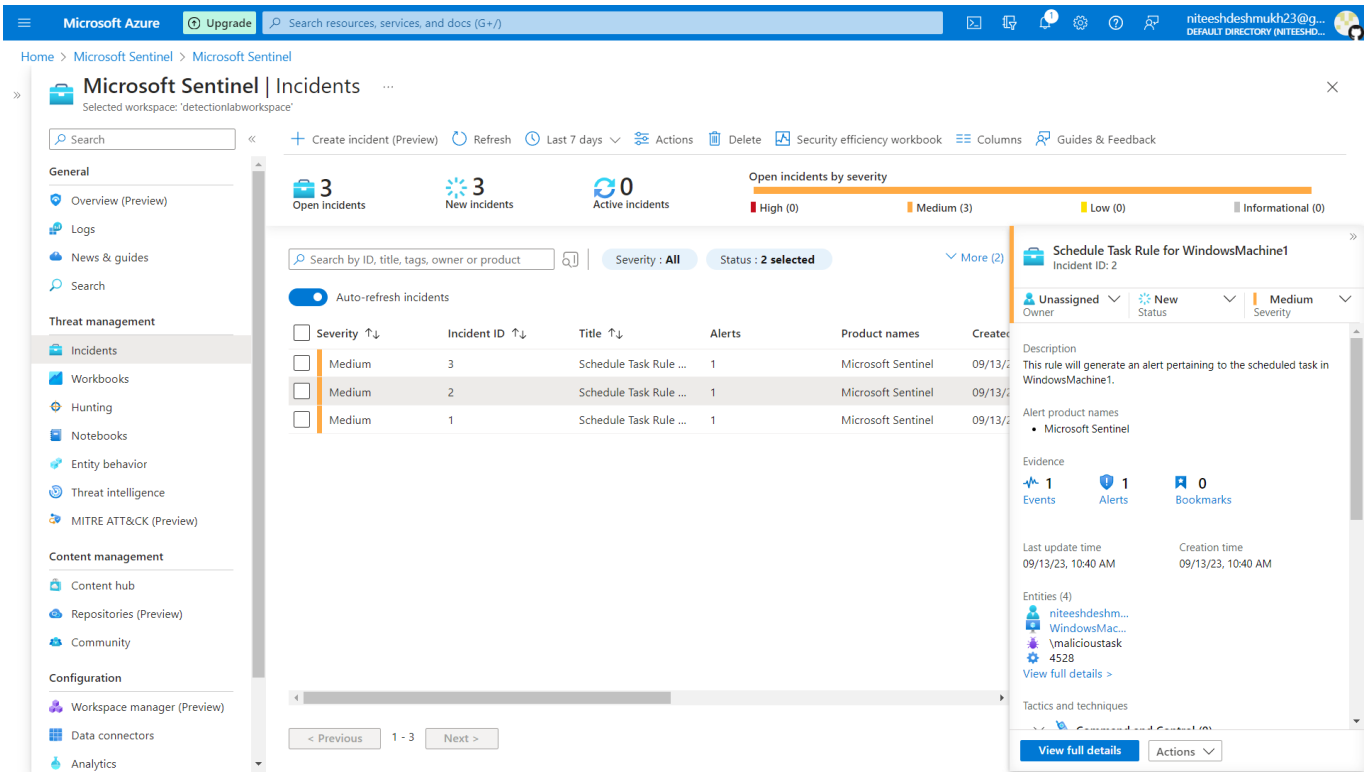
It is analogous to giving aliases for our ease of understanding.



Incident Settings and Automated Response are not necessary to alter for this time so you can go ahead to *Review and Create* to make the Analytic Rule.

Go to the *Incident* section.

On the right pane we see all the necessary information we would need to begin investigating the alert such as the host machine, user account, process ID of the task, and the name of the scheduled task.



While in this case, the scheduled task is non-malicious, in the event it was, from here an analyst would investigate the entities such as the user account, the scheduled task, host, etc. with other tools such as an EDR solution and other security tools to decide if this is a false positive or true positive.

Conclusion

The MITRE ATT&CK tactic which can be observed in this project is Persistence.

Scheduled Task/Job is a sub technique which comes under Persistence.

User Account Management is a mitigation method defined for this tactic which suggests that user account privileges should be limited to only authorize admins to create scheduled tasks on remote systems.