

Azure Cloud Detection

Niteesh Deshmukh

Contents

Getting Started

Initial Setup

Creating a Resource Group

Creating a virtual machine

Configuring Just In Time access

Creating Log Analytics Workspace

Adding Data Connectors

Configure VM and schedule a task

Create a detection rule

Generating Security Events

Utilizing Kusto Query Language (KQL)

Writing Analytic Rule and Creating Scheduled Task

Conclusion

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

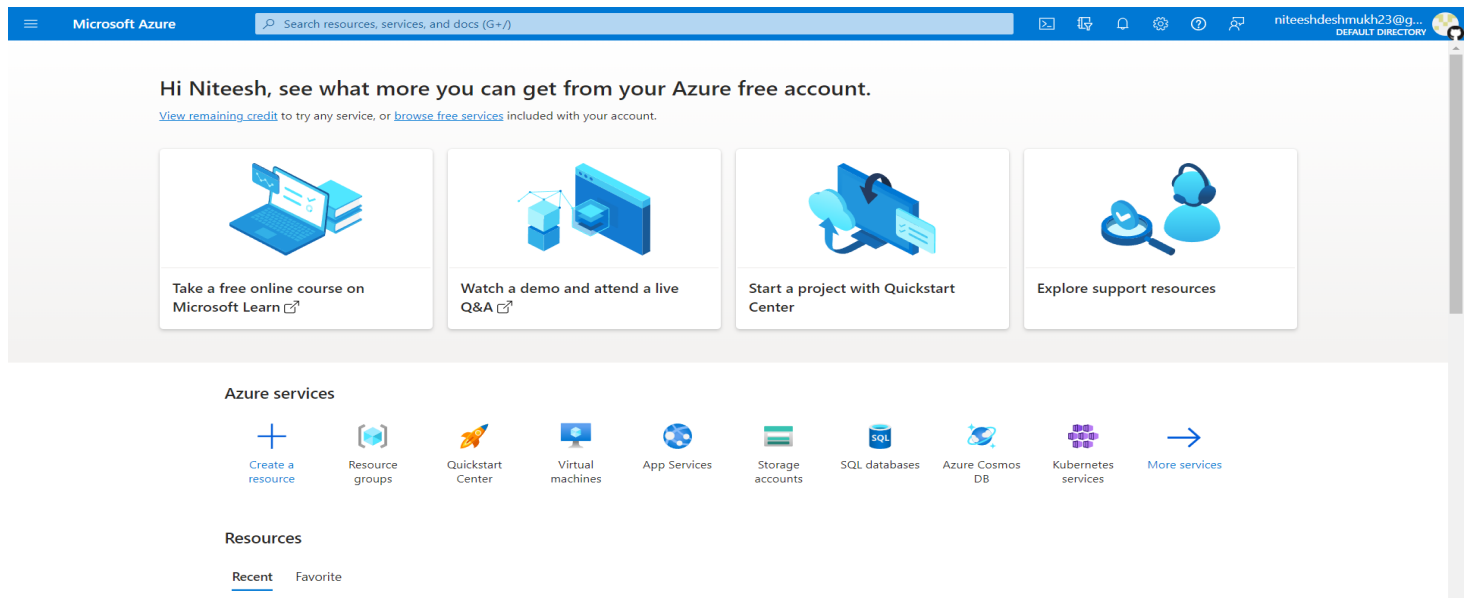
Setting up account

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.

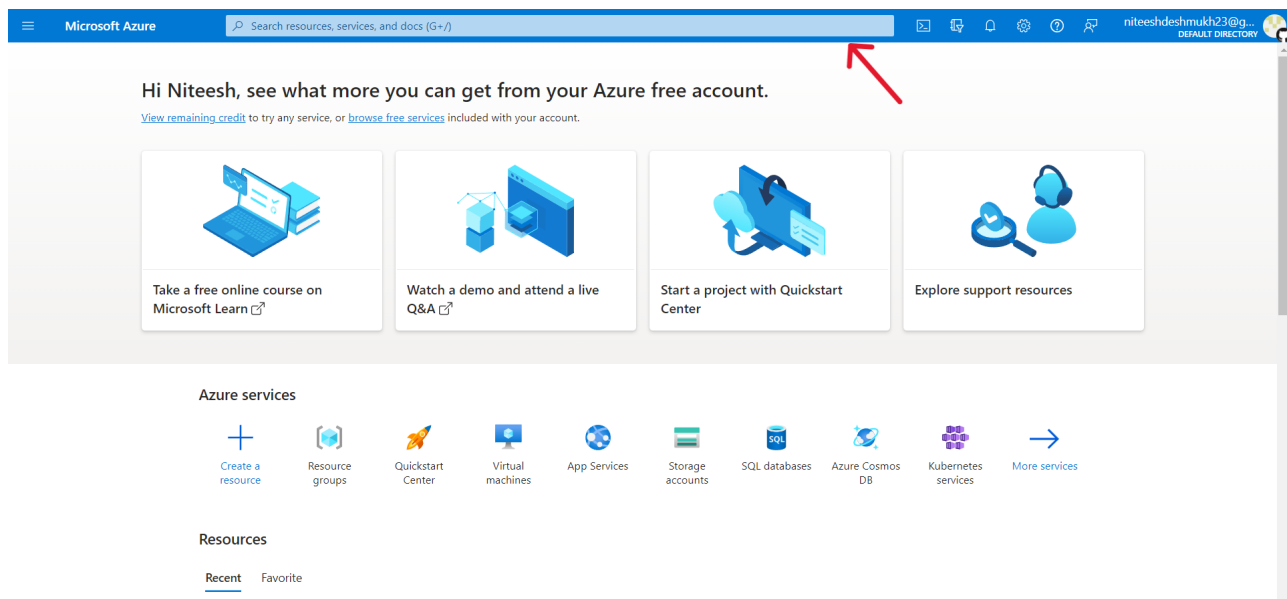
Creating a Resource Group

What is a Resource Group ?

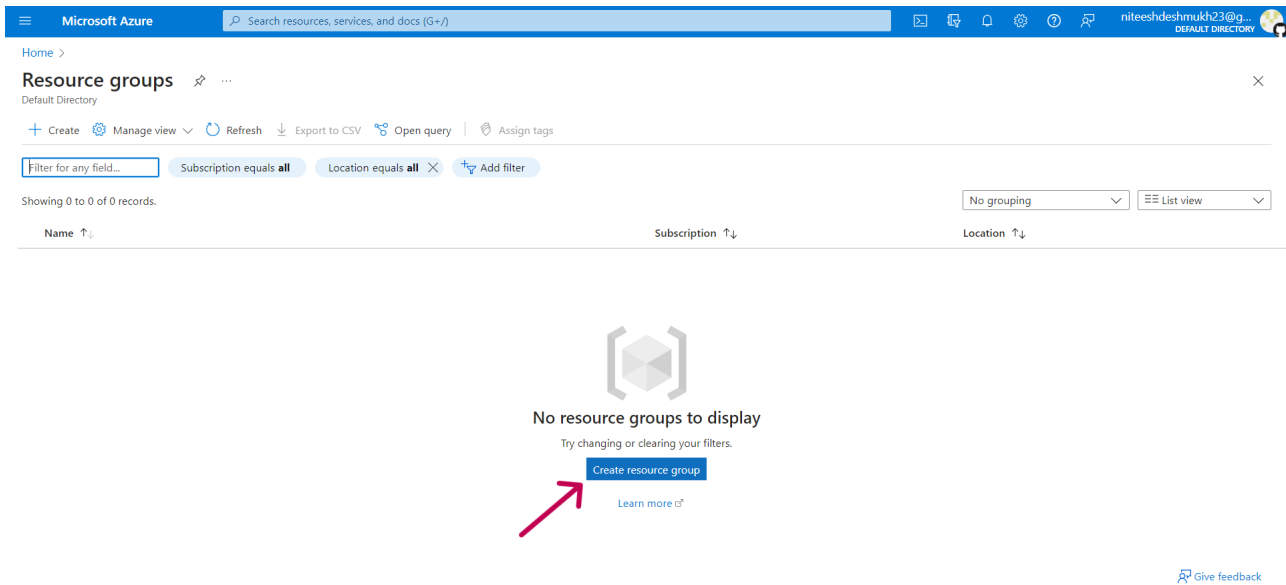
Resource Group is a logical container where you can deploy and manage Azure resources like web apps, databases, and storage accounts. Just as we keep our files and applications inside a folder or directory to manage them efficiently similarly we use a *Resource Group* to manage the Azure resources. We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

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

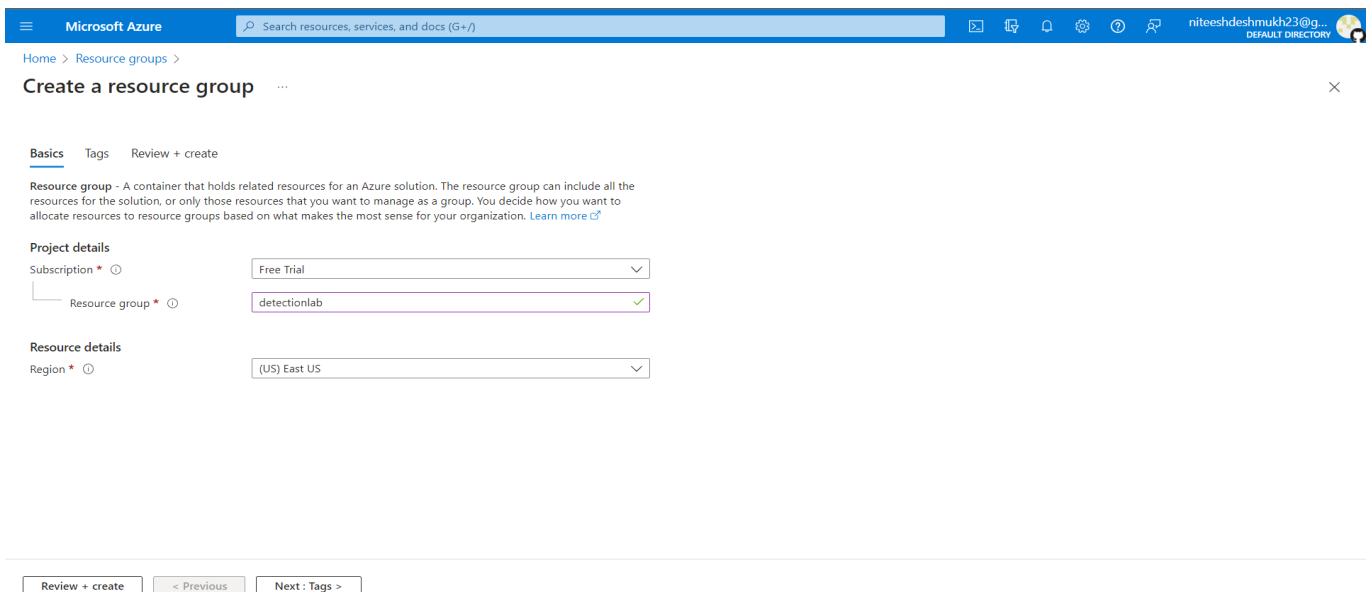
Head back to the Azure portal and search for *Resource Group* in the search bar which is at the top.



Now select *Create Resource Group*.



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

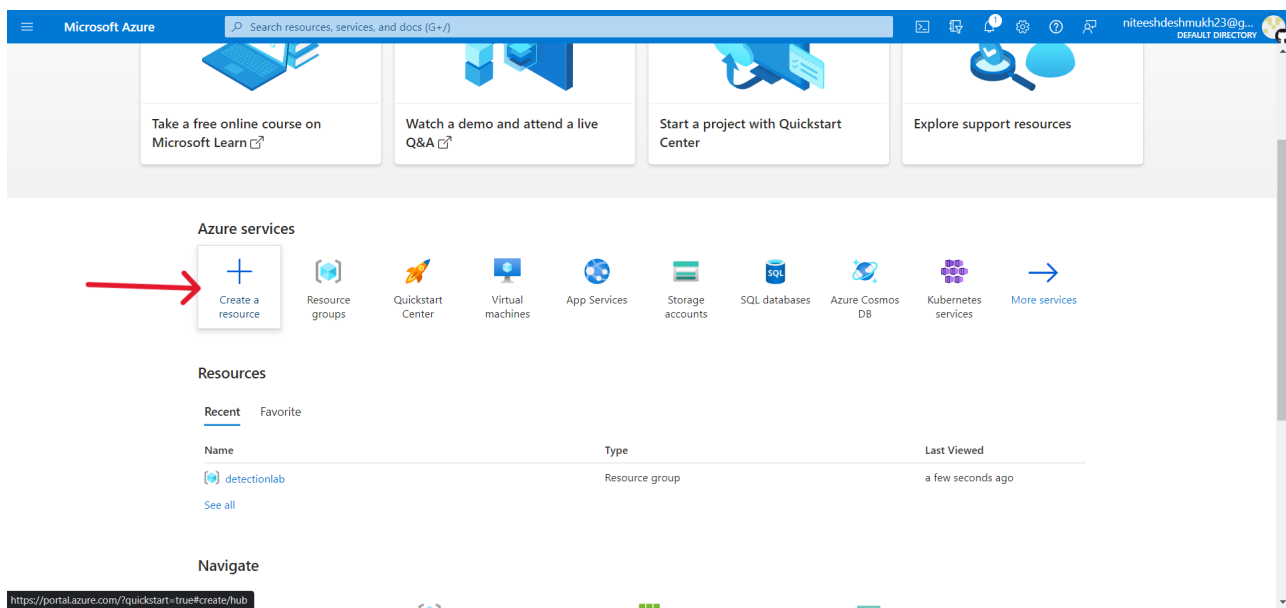


Review the information and click *Create*. We created a *Resource Group*.

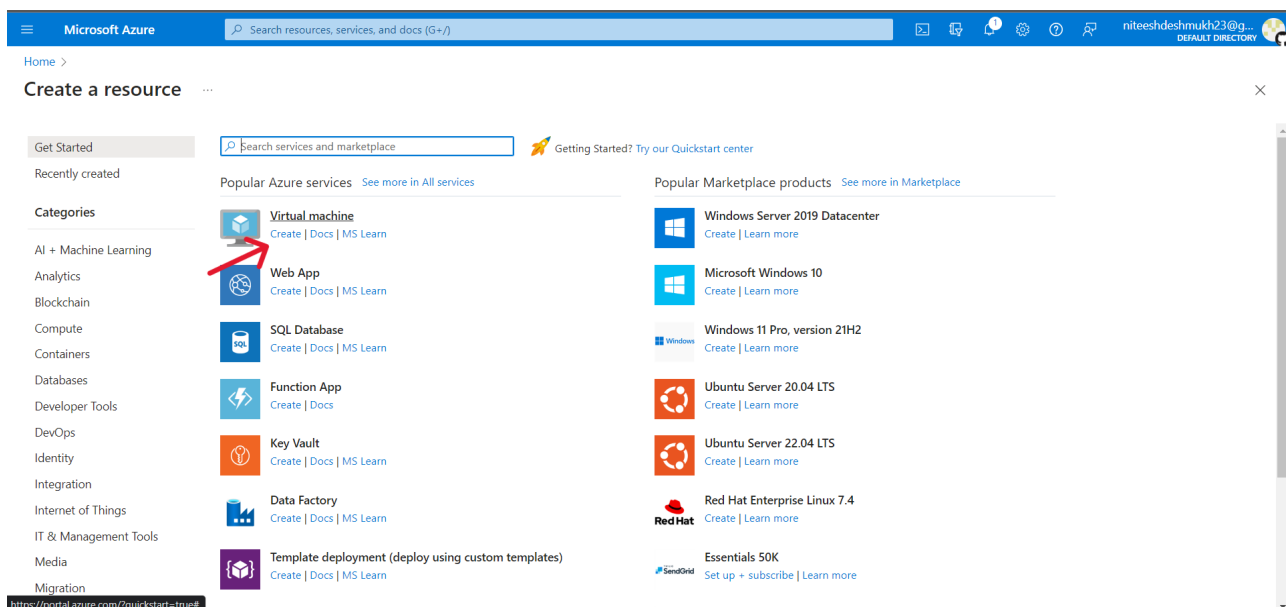
Creating a Virtual Machine

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.

Microsoft Azure

Search resources, services, and docs (G+)

niteeshdeshmukh23@g...
DEFAULT DIRECTORY

Home > Create a resource >

Create a virtual machine

This subscription may not be eligible to deploy VMs of certain sizes in certain regions.

Project details

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

Subscription *

Free Trial

Resource group *

detectionlab

Create new

Instance details

Virtual machine name *

WindowsMachine1

Region *

(US) East US

Availability options

No infrastructure redundancy required

Security type

Standard

Image *

Windows 10 Pro, version 22H2 - x64 Gen2 (free services eligible)

See all images | Configure VM generation

VM architecture

Arm64

☒ x64

Review + create

< Previous

Next : Disks >

Give feedback

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

Microsoft Azure

Search resources, services, and docs (G+)

niteeshdeshmukh23@g...
DEFAULT DIRECTORY

Home > Create a resource >

Create a virtual machine

Username *

niteeshdeshmukh

Password *

Confirm password *

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports *

None

☒ Allow selected ports

Select inbound ports *

RDP (3389)

All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.

Licensing

☒ I confirm I have an eligible Windows 10/11 license with multi-tenant hosting rights.

Review multi-tenant hosting rights for Windows 10/11 compliance

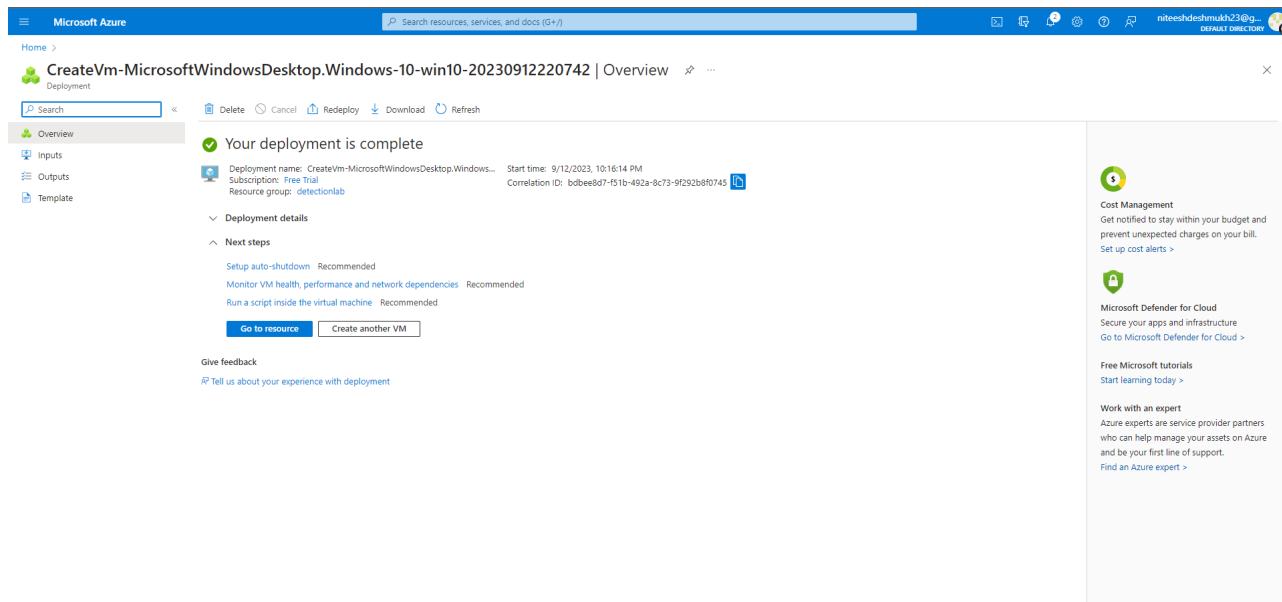
Review + create

< Previous

Next : Disks >

Give feedback

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*.

The screenshot shows the Microsoft Azure portal interface. The left sidebar contains navigation options like Overview, Activity log, Access control (IAM), Tags, Resource visualizer, Events, Settings, Deployments, Security, Deployment stacks, Policies, Properties, Locks, Cost Management, Cost analysis, Cost alerts (preview), Budgets, Advisor recommendations, Monitoring, Insights (preview), and Alerts. The main content area displays the 'detectionlab' resource group. Under the 'Resources' tab, a list of resources is shown. A red arrow points to 'WindowsMachine1-nsg', which is a Network security group.

Name	Type	Location
WindowsMachine1	Virtual machine	East US
WindowsMachine1-ip	Public IP address	East US
WindowsMachine1-nsg	Network security group	East US
WindowsMachine1-vnet	Virtual network	East US
WindowsMachine1385	Network Interface	East US
WindowsMachine1_disk1_932403190aa44a2b8caebed1d2535cfc	Disk	East US

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.

The screenshot shows the 'WindowsMachine1-nsg' Network security group page. The left sidebar has options like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Inbound security rules, Outbound security rules, Network interfaces, Subnets, Properties, Locks, Monitoring, Alerts, Diagnostic settings, Logs, and NSG Rules. The main content area shows the 'Essentials' tab with details about the resource group, location, subscription, and tags. Below this, there's a table of security rules. A red box highlights the 'Inbound Security Rules' section.

Priority	Name	Port	Protocol	Source	Destination	Action
300	RDP	3389	TCP	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerIn...	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

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.

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

Home > Virtual machines > WindowsMachine1

WindowsMachine1 | Connect

Virtual machine

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Networking
Connect
Disks
Size
Microsoft Defender for Cloud
Advisor recommendations
Extensions + applications
Availability + scaling
Configuration
Identity
Properties
Locks
Operations
Bastion
Auto-shutdown

Connecting using public IP address

Admin username : niteshdeshmukh
Port (change) : 3389 Check access
Just-in-time policy (configure) : Configured for port 3389 Request access

Most common

Native RDP
Connect via native RDP without any additional software needed. Recommended for testing only.
Using public IP address
Select

More ways to connect (3)

Native RDP
Connect from your local machine (Windows)
Switch local machine OS

1 Configure prerequisites for Native RDP
Azure needs to configure some features in order to connect to the VM.
Prerequisites configured
Port 3389 access
Just In Time on the virtual machine has temporarily configured a network security group rule for all incoming traffic from the local machine IP to port 3389. Learn about Just In Time configuration
Change the port for connecting to this virtual machine on the Connect page of the virtual machine.
Public IP address
A public IP address is required to connect via this connection method.
I understand just-in-time policy on the virtual machine may be re-configured to allow local machine IP to request just-in-time access to port 3389.
Configured

2 Open Remote Desktop Connection (on Windows)
Open Remote Desktop Connection. Or change your local machine operating system to view more instructions. Learn more

3 Download and open the RDP file
Download and open the RDP file to connect to the virtual machine.
Username niteshdeshmukh
Download RDP file

Other Information
Forgot password?
Reset password

Close Troubleshooting Give feedback

The JIT window in step 2 will be like this.

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

Home > WindowsMachine1 | Connect >

Just-in-time VM access Last week

What is just-in-time VM access?
Just-in-time VM access enables you to lock down your VMs in the network level by blocking inbound traffic to specific ports. It enables you to control the access and reduce the attack surface to your VMs, by allowing access only upon a specific need.
Learn more about just-in-time VM access >

How does it work?
Upon a user request, based on Azure RBAC, Defender for Cloud will decide whether to grant access. If a request is approved, Defender for Cloud automatically configures the NSGs to allow inbound traffic to these ports, for the requested amount of time, after which it restores the NSGs to their previous states.
Learn more about how to use just-in-time VM access >

Virtual machines
Configured Not Configured Unsupported

To let Defender for Cloud restrict access to your management ports, enable just-in-time (JIT) access control on all your "High" and "Low" risk VMs. JIT is unnecessary on a "Healthy" VM.

0 VMs
Enable JIT on 0 VMs

Search to filter items...

Virtual machine	Resource group	Subscription Name	Sever...	Reason
No results.				

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.

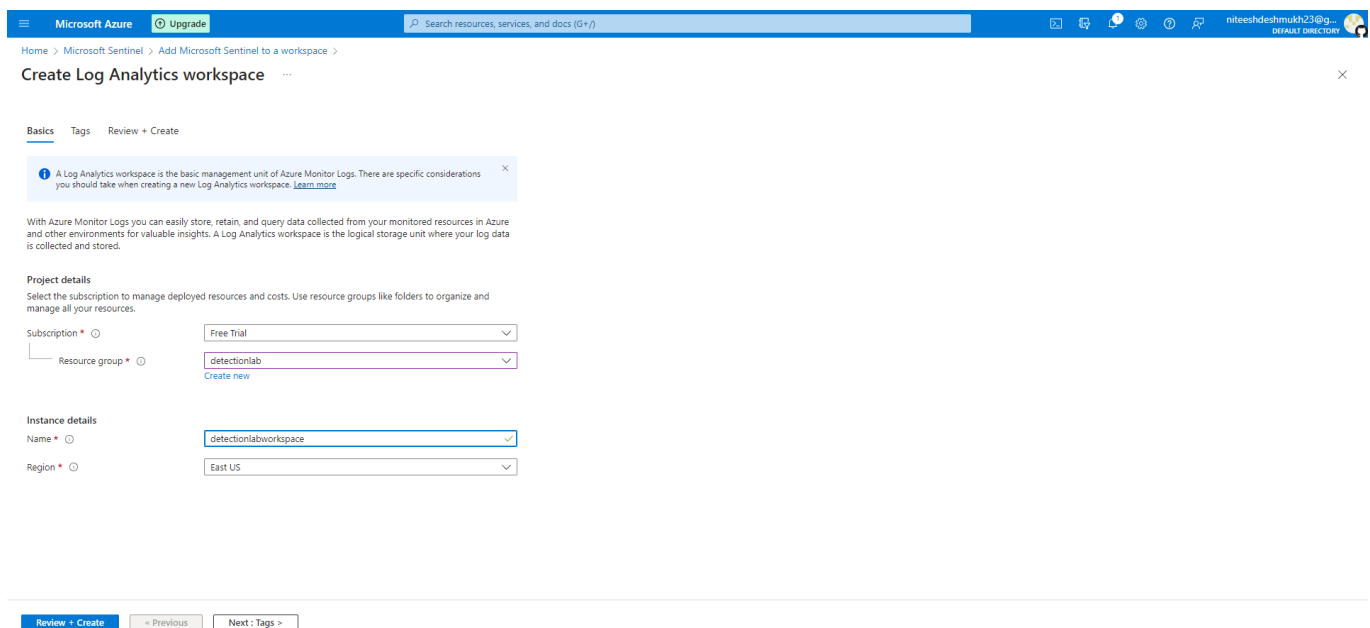
Creating Log Analytics Workspace

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.



The screenshot shows the 'Create Log Analytics workspace' page in the Azure Portal. The page has a blue header with the Microsoft Azure logo, an 'Upgrade' button, and a search bar. Below the header, the breadcrumb trail is 'Home > Microsoft Sentinel > Add Microsoft Sentinel to a workspace > Create Log Analytics workspace'. The page is divided into three tabs: 'Basics', 'Tags', and 'Review + Create'. The 'Basics' tab is active. A blue information box at the top states: 'A Log Analytics workspace is the basic management unit of Azure Monitor Logs. There are specific considerations you should take when creating a new Log Analytics workspace. [Learn more](#)'. Below this, a paragraph explains that with Azure Monitor Logs, you can easily store, retain, and query data collected from your monitored resources in Azure and other environments for valuable insights. The 'Project details' section asks to select the subscription to manage deployed resources and costs, and to use resource groups like folders to organize and manage all your resources. It has two dropdown menus: 'Subscription' (set to 'Free Trial') and 'Resource group' (set to 'detectionlab', with a 'Create new' link below it). The 'Instance details' section has two dropdown menus: 'Name' (set to 'detectionlabworkspace') and 'Region' (set to 'East US'). At the bottom, there are three buttons: 'Review + Create' (highlighted in blue), '< Previous', and 'Next : Tags >'. The user's profile 'niteeshdeshmukh23@... DEFAULT DIRECTOR' is visible in the top right corner.

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*.

Microsoft Azure

Upgrade

Search resources, services, and docs (G+/I)

niteeshdeshmukh23@g...
DEFAULT DIRECTORY

Home > Microsoft Sentinel

Default Directory

Create

Manage view

Refresh

Export to CSV

Open query

View incidents

Filter for any field...

Subscription equals all

Resource group equals all

Location equals all

Add filter

Showing 0 to 0 of 0 records.

No grouping

List view

Name	Resource group	Location	Subscription	Directory
<div><div></div><div>No Microsoft Sentinel to display</div><div>See and stop threats before they cause harm, with SIEM reinvented for a modern world. Microsoft Sentinel is your birds-eye view across the enterprise.</div><div>Create Microsoft Sentinel</div><div>Learn more</div></div>				

Give feedback

Select the workspace and click *Add*.

Microsoft Azure

Upgrade

Search resources, services, and docs (G+/I)

niteeshdeshmukh23@g...
DEFAULT DIRECTORY

Home > Microsoft Sentinel >

Add Microsoft Sentinel to a workspace

Create a new workspace

Refresh

Microsoft Sentinel offers a 31-day free trial. See [Microsoft Sentinel pricing](#) for more details.

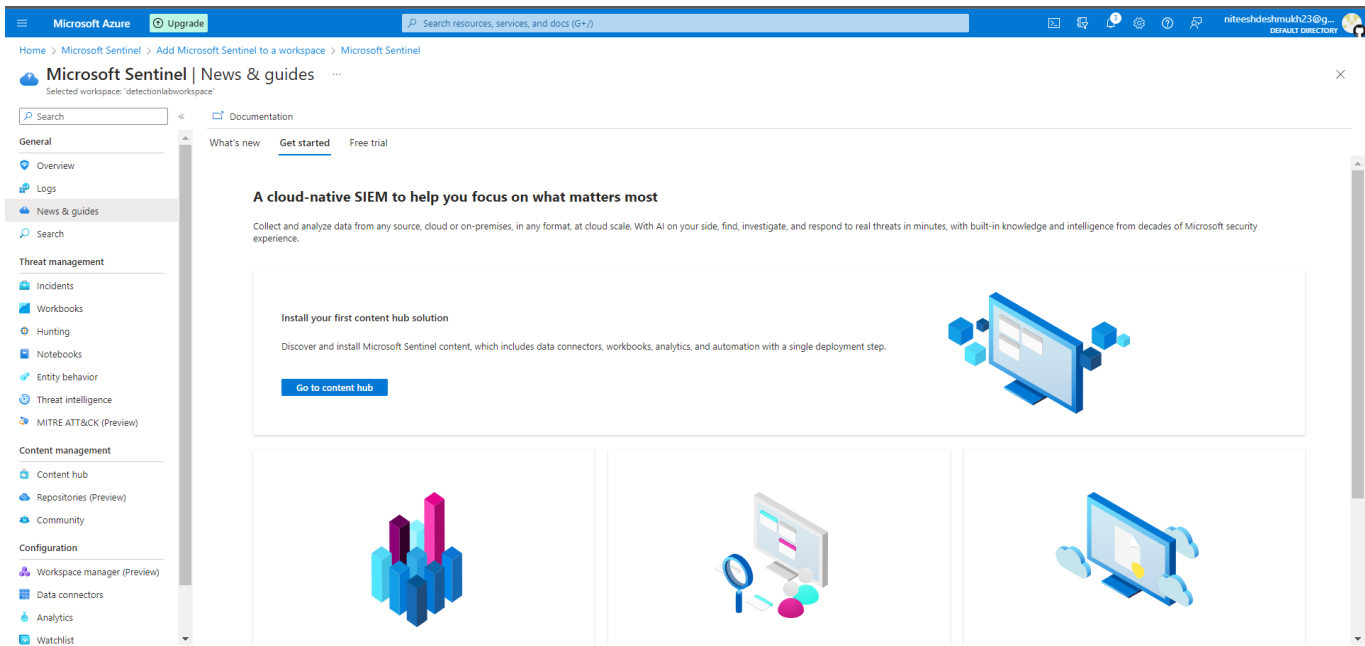
Filter by name...

Workspace	Location	ResourceGroup	Subscription	Directory
detectionlabworkspace	eastus	detectionlab	Free Trial	Default Directory

Add

Cancel

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.

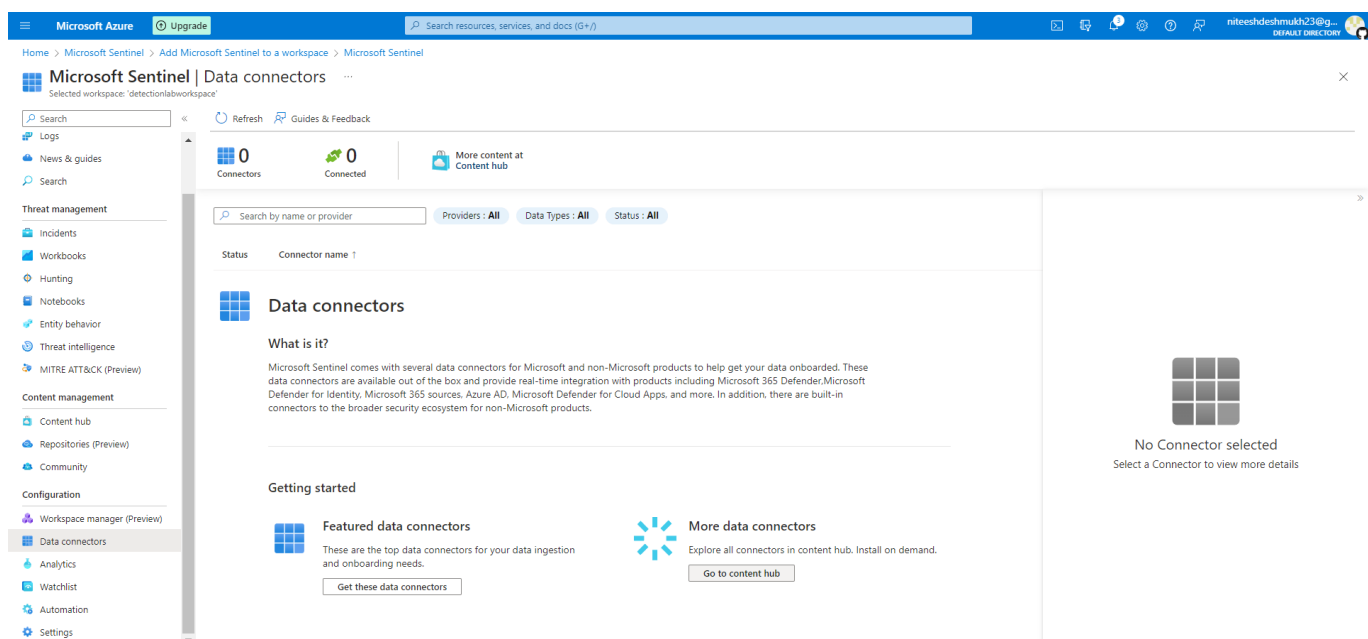
In order to do so we will use *Data Connectors*.

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.

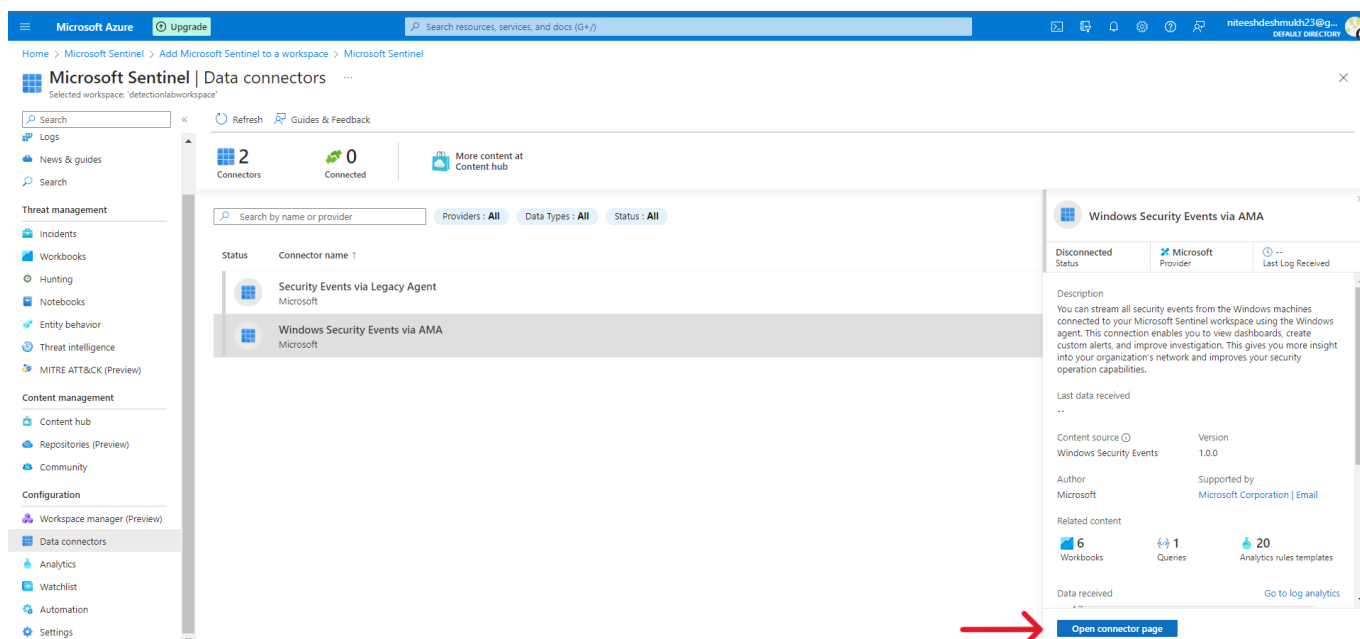


It can be observed that there are currently no data connectors installed.

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*.

The screenshot shows the Microsoft Azure portal interface. On the left, the 'Windows Security Events via AMA' connector is displayed with its description, version (1.0.0), and related content (6 Workbooks, 1 Query, 20 Analytics rules templates). The main area shows the 'Instructions' tab with prerequisites and configuration steps. On the right, the 'Create Data Collection Rule' window is open, showing the 'Basics' tab. The 'Rule details' section has the following values: Rule Name: 'datacollectionrule1', Subscription: 'Free Trial', and Resource Group: 'detectionlab'. The 'Next: Resources >' button is visible at the bottom of the right pane.

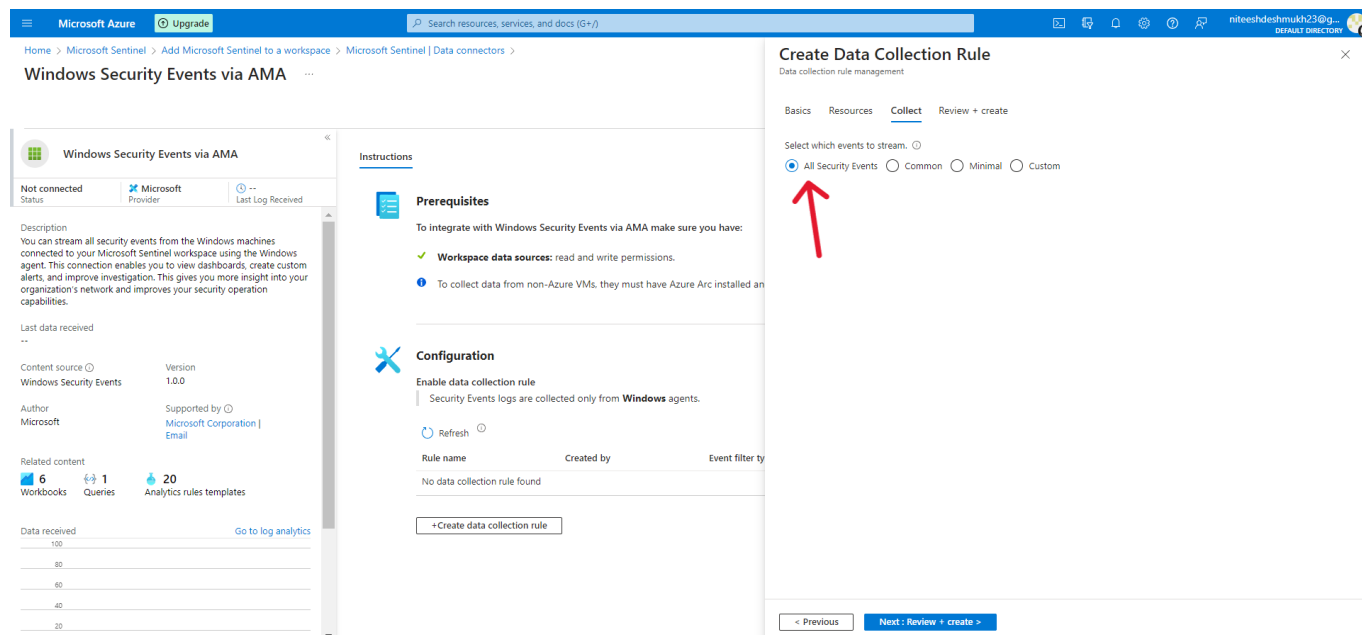
In the *Resources* click on *Add Resources* and add the Windows 10 virtual machine and then move to the *Collect* tab.

This screenshot shows the 'Create Data Collection Rule' window with the 'Resources' tab selected. A red box highlights the '+Add resource(s)' button and a table of added resources. The table has the following data:

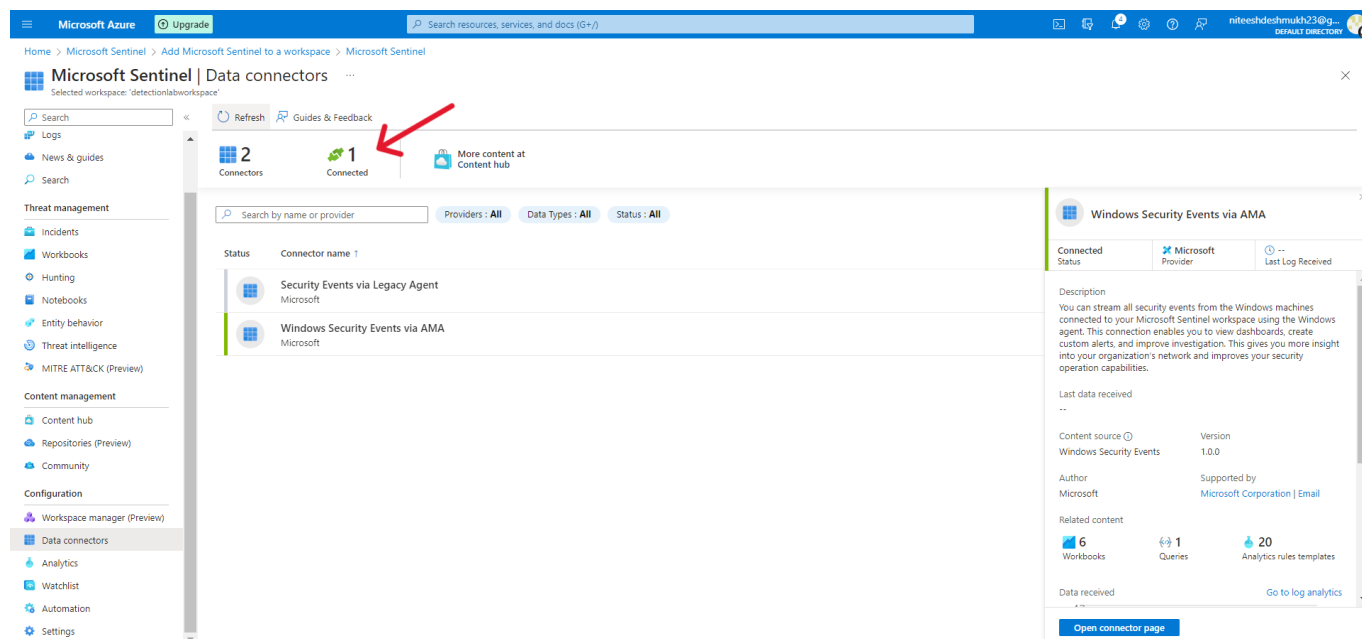
Name	Type	Resource group	Subscription
WindowsMachine1	Microsoft.Compute/virtualMac...	detectionlab	Free Trial

Below the table, a red arrow points to the 'Next: Collect >' button, indicating the next step in the process.

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.

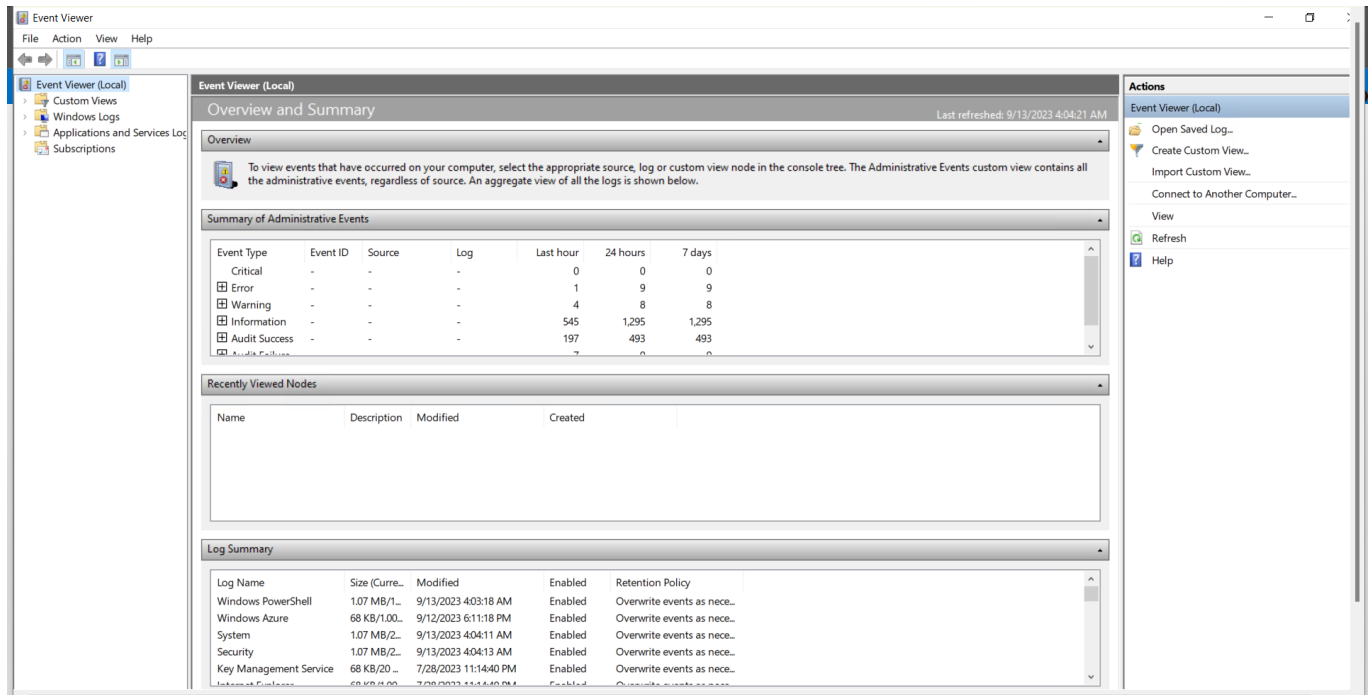
Click “Start” at the top page to turn on the VM if it’s not on already . Enable Just in time Access if necessary.

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.

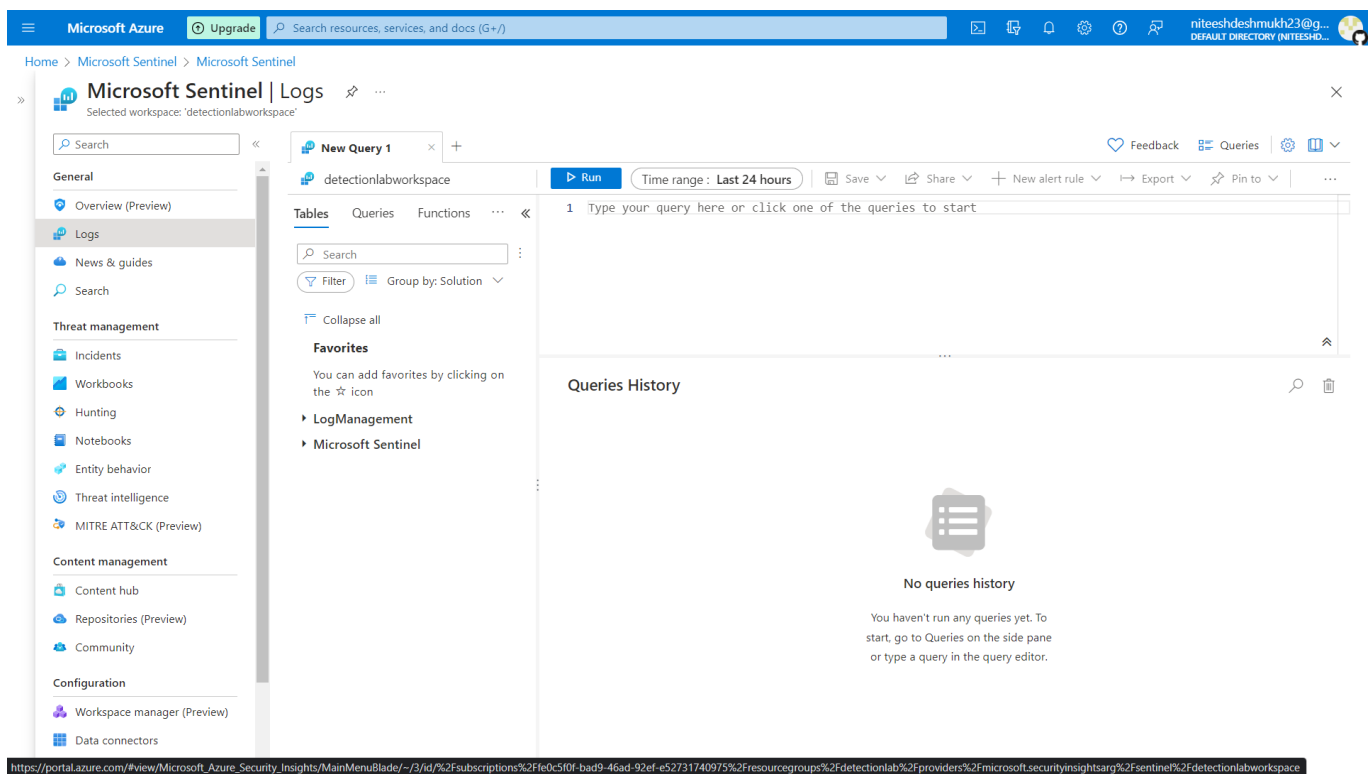
We will focus on a specific security log which indicates successful logon into the system. It has an id assigned to it which is 4624.

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.

The screenshot shows a query interface with a top toolbar containing 'Run', 'Time range: Last 24 hours', 'Save', 'Share', 'New alert rule', 'Export', and 'Pin to'. Below the toolbar is a query editor with the following code:

```
1 SecurityEvent
2 | where EventID == 4624
3 | project TimeGenerated, Computer, AccountName
```

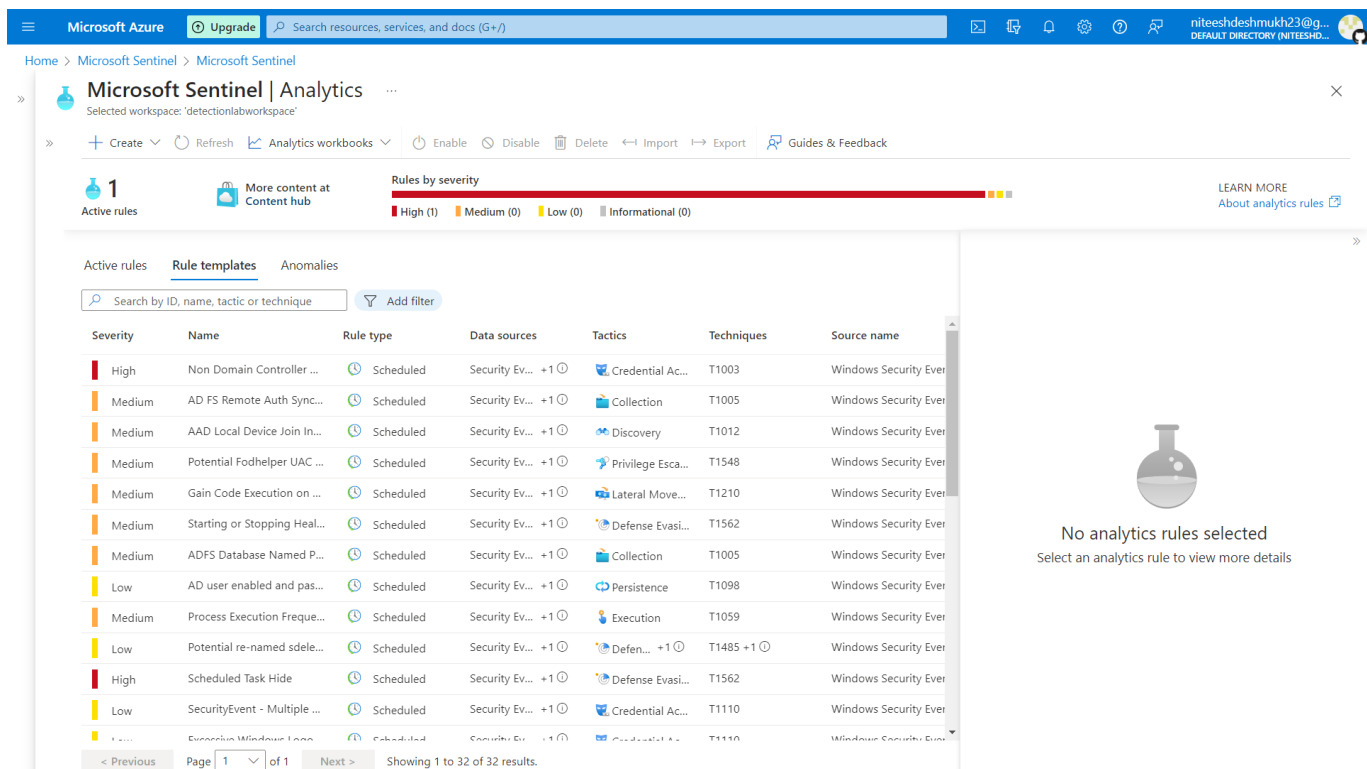
The results are displayed in a table with columns: TimeGenerated [UTC] ↑↓, Computer, and AccountName. The table contains 16 rows of data, all from 'WindowsMachine1'. The 'AccountName' column is empty for all entries. The interface also includes a 'Results' tab, a 'Chart' tab, an 'Add bookmark' button, and a 'Columns' sidebar. At the bottom, it shows '2s 42ms | Display time (UTC+00:00)' and a 'Query details' link.

TimeGenerated [UTC] ↑↓	Computer	AccountName
> 9/13/2023, 4:12:02.226 AM	WindowsMachine1	
> 9/13/2023, 4:12:01.478 AM	WindowsMachine1	
> 9/13/2023, 4:11:58.890 AM	WindowsMachine1	
> 9/13/2023, 4:11:24.813 AM	WindowsMachine1	
> 9/13/2023, 4:11:24.638 AM	WindowsMachine1	
> 9/13/2023, 4:05:46.019 AM	WindowsMachine1	
> 9/13/2023, 4:05:05.163 AM	WindowsMachine1	
> 9/13/2023, 4:02:13.408 AM	WindowsMachine1	
> 9/13/2023, 4:00:58.174 AM	WindowsMachine1	
> 9/13/2023, 3:59:46.567 AM	WindowsMachine1	
> 9/13/2023, 3:59:40.255 AM	WindowsMachine1	
> 9/13/2023, 3:59:39.931 AM	WindowsMachine1	
> 9/13/2023, 3:59:37.122 AM	WindowsMachine1	
> 9/13/2023, 3:59:37.122 AM	WindowsMachine1	
> 9/13/2023, 3:59:36.944 AM	WindowsMachine1	
> 9/13/2023, 3:59:31.231 AM	WindowsMachine1	

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.



The screenshot displays the Microsoft Sentinel Analytics interface. At the top, there's a navigation bar with 'Home > Microsoft Sentinel > Microsoft Sentinel'. Below this, the 'Microsoft Sentinel | Analytics' header is visible, along with a search bar and various action buttons like 'Create', 'Refresh', 'Analytics workbooks', 'Enable', 'Disable', 'Delete', 'Import', 'Export', and 'Guides & Feedback'. A 'Rules by severity' bar shows counts for High (1), Medium (0), Low (0), and Informational (0). The main content area is divided into 'Active rules', 'Rule templates', and 'Anomalies'. The 'Active rules' tab is selected, showing a table of rules. The table has columns for Severity, Name, Rule type, Data sources, Tactics, Techniques, and Source name. The first rule is 'Non Domain Controller ...' with a High severity and Scheduled rule type. The second rule is 'AD FS Remote Auth Sync...' with a Medium severity and Scheduled rule type. The third rule is 'AAD Local Device Join In...' with a Medium severity and Scheduled rule type. The fourth rule is 'Potential Fodhelper UAC ...' with a Medium severity and Scheduled rule type. The fifth rule is 'Gain Code Execution on ...' with a Medium severity and Scheduled rule type. The sixth rule is 'Starting or Stopping Heal...' with a Medium severity and Scheduled rule type. The seventh rule is 'ADFS Database Named P...' with a Medium severity and Scheduled rule type. The eighth rule is 'AD user enabled and pas...' with a Low severity and Scheduled rule type. The ninth rule is 'Process Execution Freque...' with a Medium severity and Scheduled rule type. The tenth rule is 'Potential re-named sdele...' with a Low severity and Scheduled rule type. The eleventh rule is 'Scheduled Task Hide' with a High severity and Scheduled rule type. The twelfth rule is 'SecurityEvent - Multiple ...' with a Low severity and Scheduled rule type. The thirteenth rule is 'Executive Windows Logon' with a Low severity and Scheduled rule type. The table is paginated, showing 1 of 1 results. A 'No analytics rules selected' message is displayed on the right side of the interface.

Severity	Name	Rule type	Data sources	Tactics	Techniques	Source name
High	Non Domain Controller ...	Scheduled	Security Ev... +1	Credential Ac...	T1003	Windows Security Ever
Medium	AD FS Remote Auth Sync...	Scheduled	Security Ev... +1	Collection	T1005	Windows Security Ever
Medium	AAD Local Device Join In...	Scheduled	Security Ev... +1	Discovery	T1012	Windows Security Ever
Medium	Potential Fodhelper UAC ...	Scheduled	Security Ev... +1	Privilege Esca...	T1548	Windows Security Ever
Medium	Gain Code Execution on ...	Scheduled	Security Ev... +1	Lateral Move...	T1210	Windows Security Ever
Medium	Starting or Stopping Heal...	Scheduled	Security Ev... +1	Defense Evasi...	T1562	Windows Security Ever
Medium	ADFS Database Named P...	Scheduled	Security Ev... +1	Collection	T1005	Windows Security Ever
Low	AD user enabled and pas...	Scheduled	Security Ev... +1	Persistence	T1098	Windows Security Ever
Medium	Process Execution Freque...	Scheduled	Security Ev... +1	Execution	T1059	Windows Security Ever
Low	Potential re-named sdele...	Scheduled	Security Ev... +1	Defen... +1	T1485 +1	Windows Security Ever
High	Scheduled Task Hide	Scheduled	Security Ev... +1	Defense Evasi...	T1562	Windows Security Ever
Low	SecurityEvent - Multiple ...	Scheduled	Security Ev... +1	Credential Ac...	T1110	Windows Security Ever
Low	Executive Windows Logon	Scheduled	Security Ev... +1	Credential Ac...	T1110	Windows Security Ever

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

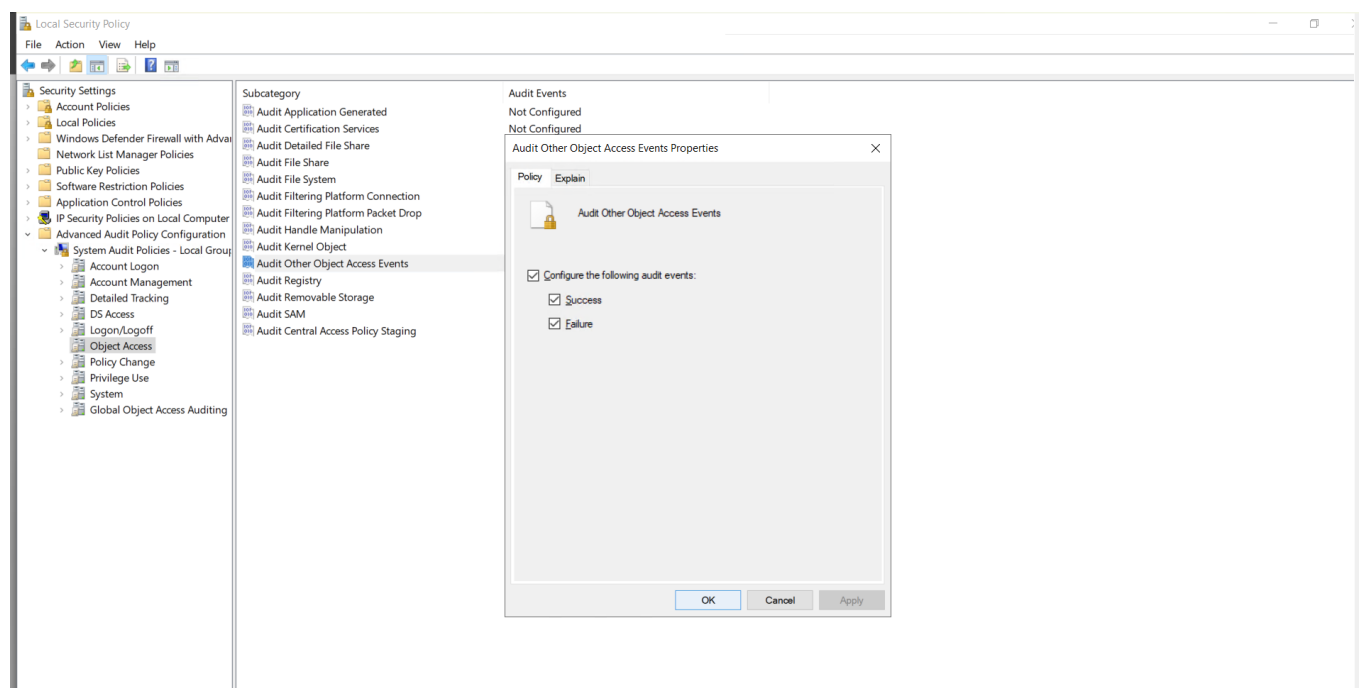
to monitor for that specific action so we can be alerted to the to the activity in our SIEM in order to simulate the scenario.

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*.

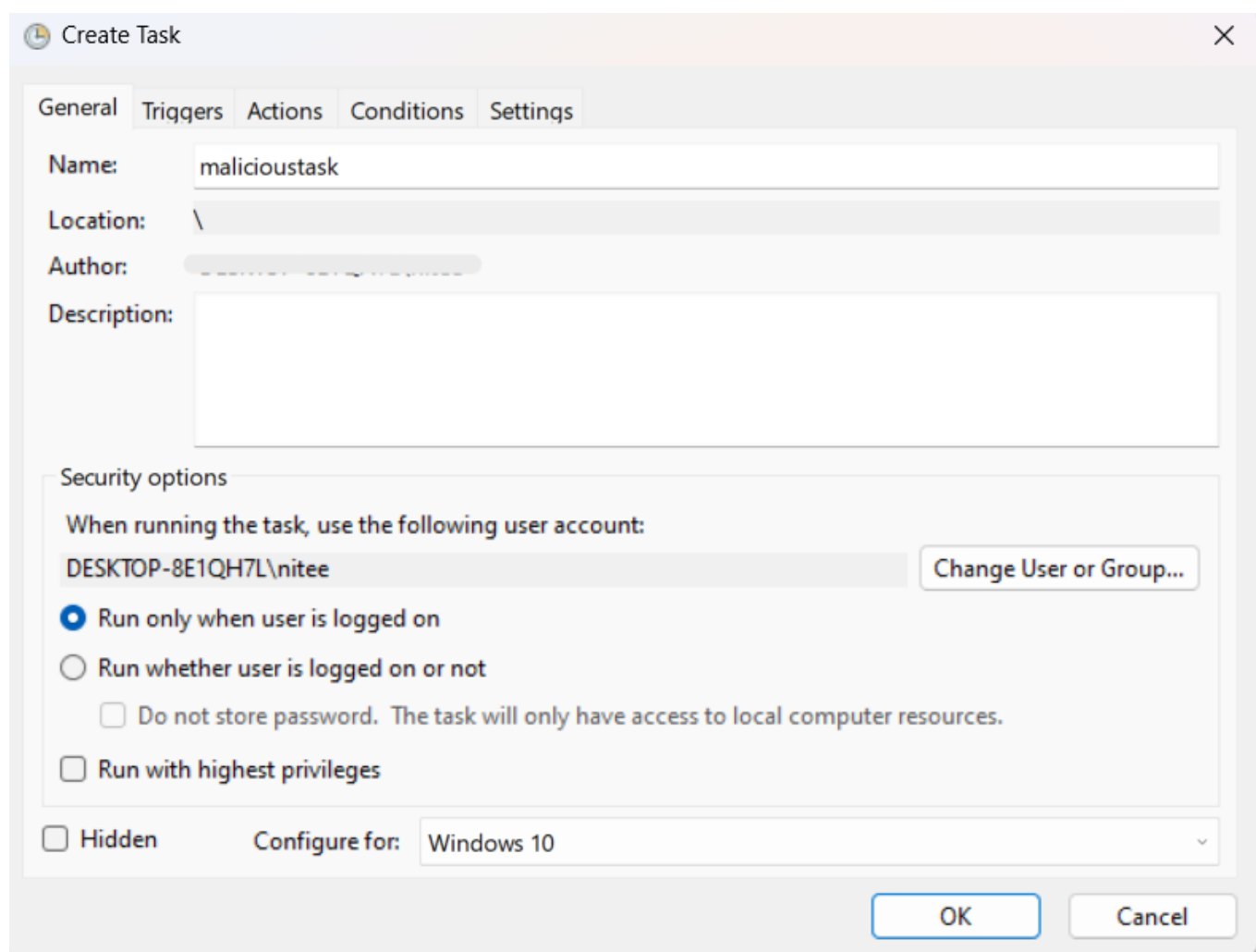
Enable *Success* and *Failure*.



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.



The screenshot shows the 'Create Task' dialog box in Windows Task Scheduler. The 'General' tab is selected. The 'Name' field contains 'malicioustask'. The 'Location' field is empty. The 'Author' field is empty. The 'Description' field is empty. Under 'Security options', the text 'When running the task, use the following user account:' is followed by a text box containing 'DESKTOP-8E1QH7L\nitee' and a 'Change User or Group...' button. Below this, there are three radio buttons: 'Run only when user is logged on' (selected), 'Run whether user is logged on or not', and 'Do not store password. The task will only have access to local computer resources.' There are also two checkboxes: 'Run with highest privileges' and 'Hidden'. At the bottom, the 'Configure for:' dropdown menu is set to 'Windows 10'. The 'OK' and 'Cancel' buttons are at the bottom right.

Create Task

General Triggers Actions Conditions Settings

Name: malicioustask

Location: \

Author:

Description:

Security options

When running the task, use the following user account:

DESKTOP-8E1QH7L\nitee Change User or Group...

☒ Run only when user is logged on

☐ Run whether user is logged on or not

☐ Do not store password. The task will only have access to local computer resources.

☐ Run with highest privileges

☐ Hidden

Configure for: Windows 10

OK Cancel

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

The screenshot shows the 'New Trigger' dialog box in Windows Task Scheduler. The 'Begin the task' dropdown is set to 'On a schedule'. Under the 'Settings' section, 'One time' is selected. The 'Start' date is '15-09-2023' and the time is '10:58:54'. The 'Synchronize across time zones' checkbox is unchecked. The 'Advanced settings' section includes several options: 'Delay task for up to (random delay): 1 hour', 'Repeat task every: 1 hour for a duration of: 1 day', 'Stop all running tasks at end of repetition duration' (unchecked), 'Stop task if it runs longer than: 3 days', 'Expire: 15-09-2024 10:58:55', and 'Synchronize across time zones' (unchecked). The 'Enabled' checkbox is checked. At the bottom right are 'OK' and 'Cancel' buttons.

New Trigger

Begin the task: On a schedule

Settings

☒ One time
☐ Daily
☐ Weekly
☐ Monthly

Start: 15-09-2023 10:58:54 ☐ Synchronize across time zones

Advanced settings

☐ Delay task for up to (random delay): 1 hour

☐ Repeat task every: 1 hour for a duration of: 1 day

☐ Stop all running tasks at end of repetition duration

☐ Stop task if it runs longer than: 3 days

☐ Expire: 15-09-2024 10:58:55 ☐ Synchronize across time zones

☒ Enabled

OK Cancel

Navigate to the action tab 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.

The image shows a 'New Action' dialog box with a title bar containing a close button (X). The main text reads 'You must specify what action this task will perform.' Below this is a section titled 'Settings' which contains a dropdown menu for 'Action:' set to 'Start a program'. Underneath, the 'Program/script:' field contains the path 'C:\Program Files (x86)\Internet Explorer\iexplore.exe' and a 'Browse...' button. Below that are two optional fields: 'Add arguments (optional):' and 'Start in (optional):', both with empty text boxes. At the bottom right are 'OK' and 'Cancel' buttons.

New Action

You must specify what action this task will perform.

Action: Start a program

Settings

Program/script:

"C:\Program Files (x86)\Internet Explorer\iexplore.exe" Browse...

Add arguments (optional):

Start in (optional):

OK Cancel

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.

The screenshot shows the 'Analytics rule wizard - Create a new Scheduled rule' interface in Microsoft Sentinel. The 'General' tab is selected, displaying the following details:

- Name:** Schedule Task Rule for WindowsMachine1
- Description:** This rule will generate an alert pertaining to the scheduled task in WindowsMachine1.
- Severity:** Medium
- Tactics and techniques:** (3)
- Status:** Enabled

At the bottom, there is a blue button labeled 'Next : Set rule logic >'.

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.

The screenshot displays the Microsoft Sentinel 'Analytics rule wizard - Edit existing' interface. The left pane shows the 'Set rule logic' tab with a KQL query. The right pane shows the 'Logs' view with a table of results.

Rule 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
```

Results

Computer	TimeGenerated [UTC]	ClientProcessID	NameofSceuduledTask	User
WindowsMachine1	9/13/2023, 4:41:40.216 AM	4528	\malicioustask	niteeshdeshmukh
Computer	WindowsMachine1			
TimeGenerated [UTC]	2023-09-13T04:41:40.2163375Z			
ClientProcessID	4528			
NameofSceuduledTask	\malicioustask			
User	niteeshdeshmukh			

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.

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

Home > Microsoft Sentinel > Microsoft Sentinel

Microsoft Sentinel | Incidents

Selected workspace: 'detectionlabworkspace'

Search

General

- Overview (Preview)
- Logs
- News & guides
- Search

Threat management

- Incidents
- Workbooks
- Hunting
- Notebooks
- Entity behavior
- Threat intelligence
- MITRE ATT&CK (Preview)

Content management

- Content hub
- Repositories (Preview)
- Community

Configuration

- Workspace manager (Preview)
- Data connectors
- Analytics

3 Open Incidents | 3 New Incidents | 0 Active Incidents

Open incidents by severity: High (0) | Medium (3) | Low (0) | Informational (0)

Search by ID, title, tags, owner or product

Severity: All | Status: 2 selected | More (2)

Auto-refresh incidents

Severity	Incident ID	Title	Alerts	Product names	Created
Medium	3	Schedule Task Rule ...	1	Microsoft Sentinel	09/13/23
Medium	2	Schedule Task Rule ...	1	Microsoft Sentinel	09/13/23
Medium	1	Schedule Task Rule ...	1	Microsoft Sentinel	09/13/23

< Previous | 1 - 3 | Next >

Schedule Task Rule for WindowsMachine1

Incident ID: 2

Unassigned | Owner: niteeshdeshm... | New | Status: New | Medium | Severity: Medium

Description: This rule will generate an alert pertaining to the scheduled task in WindowsMachine1.

Alert product names: Microsoft Sentinel

Evidence: 1 Events | 1 Alerts | 0 Bookmarks

Last update time: 09/13/23, 10:40 AM | Creation time: 09/13/23, 10:40 AM

Entities (4): niteeshdeshm..., WindowsMac..., \malicioustask, 4528

View full details >

Tactics and techniques: Scheduled Task (TA0002)

View full details | Actions

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.