**Azure Detection project**

**Topology:**

****

Many security positions particularly on the blue team side look for SIEM experience when evaluating potential candidates. **SIEM**stands for Security Information and Event Management System. This is a tool allows security professionals to analyze log data from a variety of sources within their network such as Firewalls, IDS/IPS, Identity solutions, and much more. This, allows the security professionals to monitor, prioritize and remediate potential threats in real-time. Azure Sentinel (now referred to as **Microsoft Sentinel**) is a cloud-based SIEM/SOAR solution providing capabilities such as security analytics, threat intelligence, threat response, and more in a single platform. First, we need to understand the basic principle of detection: **{Detection in cybersecurity is the practice of analyzing an environment for any anomalous activity that could possibly result in a network being compromised.}**

**Content:**

* [*Configure and Deploy Azure Resources such as Log Analytics Workspace, Virtual Machines, and Azure Sentinel*](#_Part_1:_Setup)
* [*Implement Network and Virtual Machine Security Best Practices*](#_Part_2:_Network)
* [*Utilize Data Connectors to bring data into Sentinel for Analysis*](#_PART_3:_Utilize)
* *[Understand Windows Security Event logs](#_Part_4:_Generating)*
* *[Configure Windows Security Policies](#_Part_4:_Generating)*
* [*Utilize KQL to query Logs*](#_Part_4:_Kusto)
* [*Write Custom Analytic Rules to detect Microsoft Security Events*](#_Part_5:_Writing)
* [*Utilize MITRE ATT&CK to map adversary tactics, techniques, detection and mitigation procedures*](#_Part_6:_MITRE)

## Part 1: Setup Lab Resources

**Step 1: Create Free Azure Account**

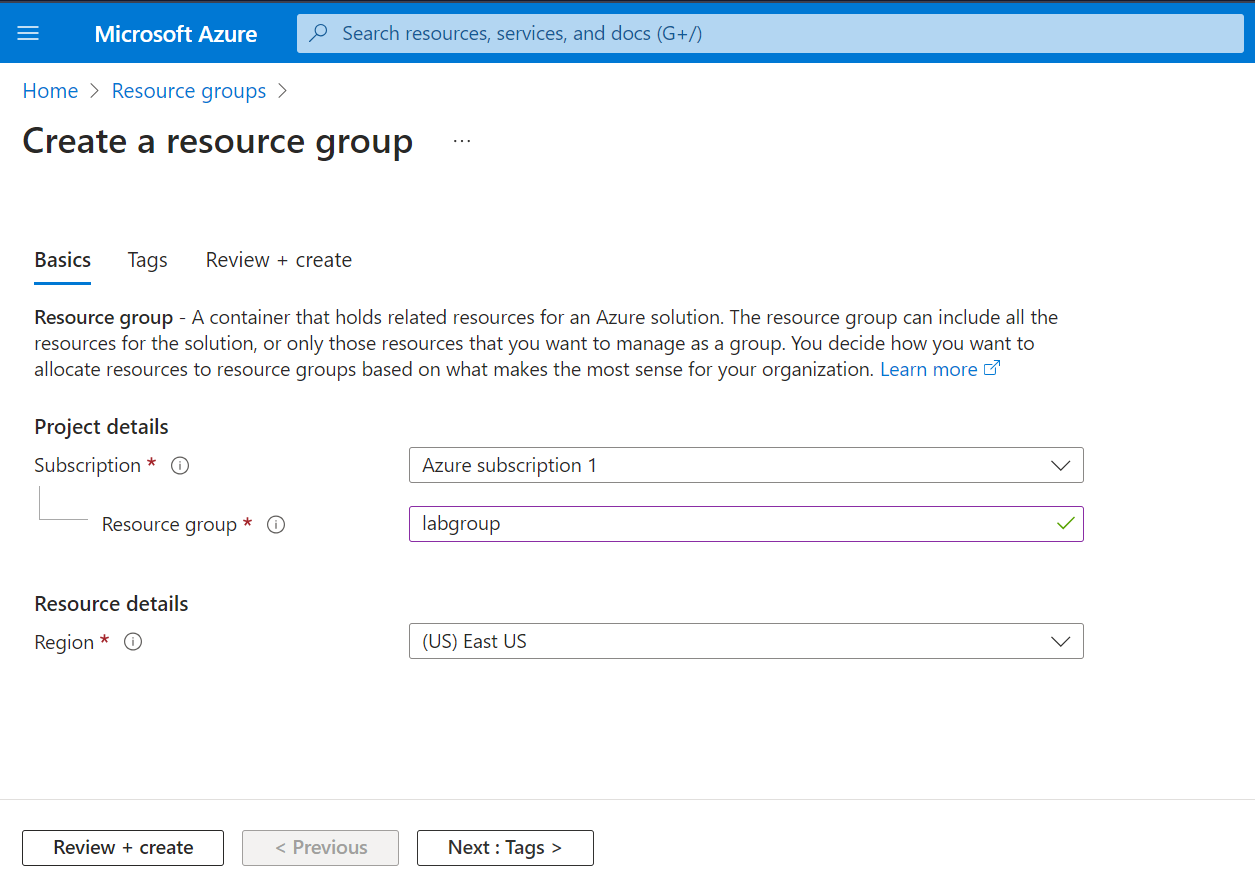
Use this link to create your Account. This process will automatically set up your account and associated Azure Subscription.

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

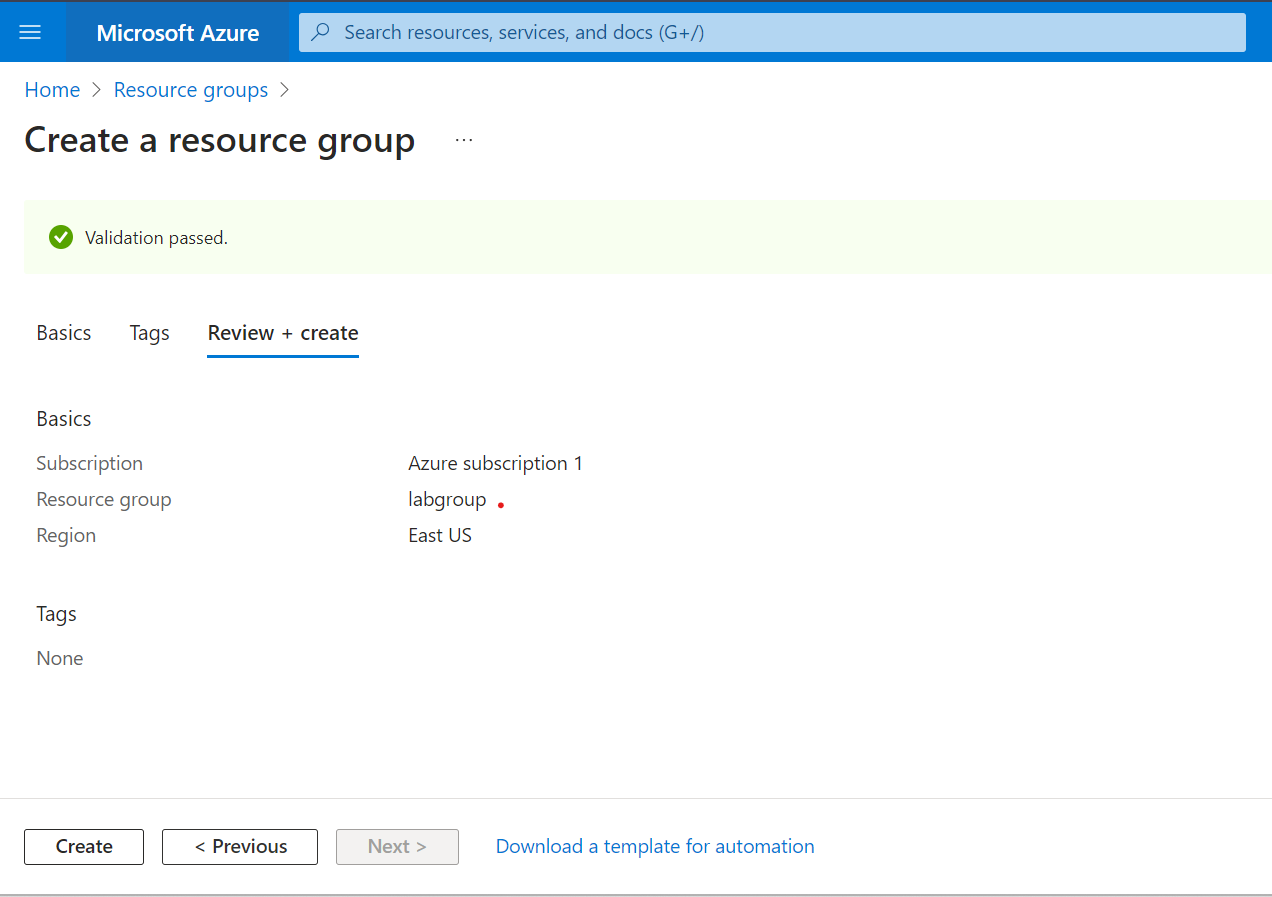
**Step 2: Deployment of Resource Group**

When working with Azure Resources, Resource groups are logical containers for all our resources. We will first create a resource group to group all the resources we will use for this lab. These resources will include a **Windows 10 VM**, **Log Analytics Workspace**, and an **Azure Sentinel Resource**.

Search “Resource Group” in the Azure Portal Search Bar and Follow the on-screen prompts for the basic tab.



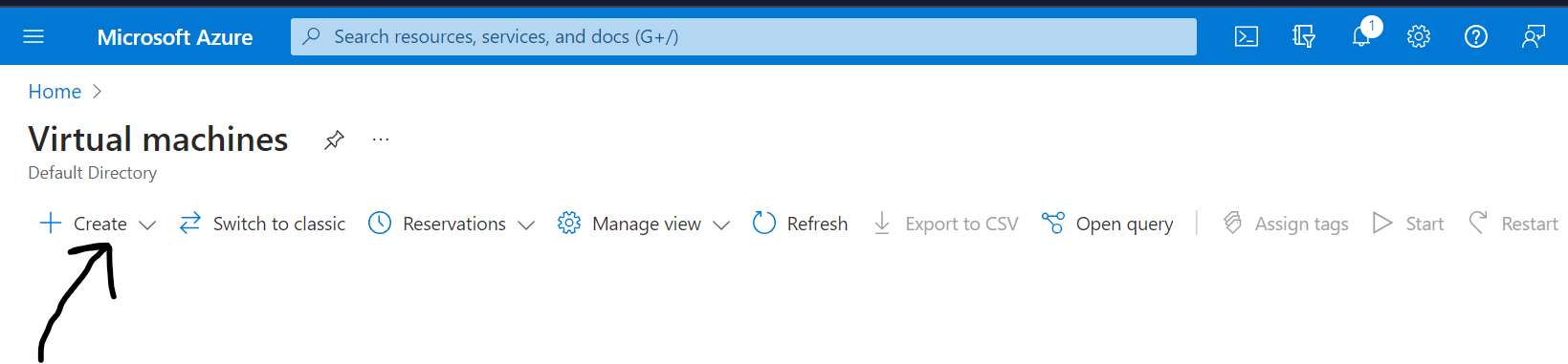
You can skip to “review and create” after basics and click the create.



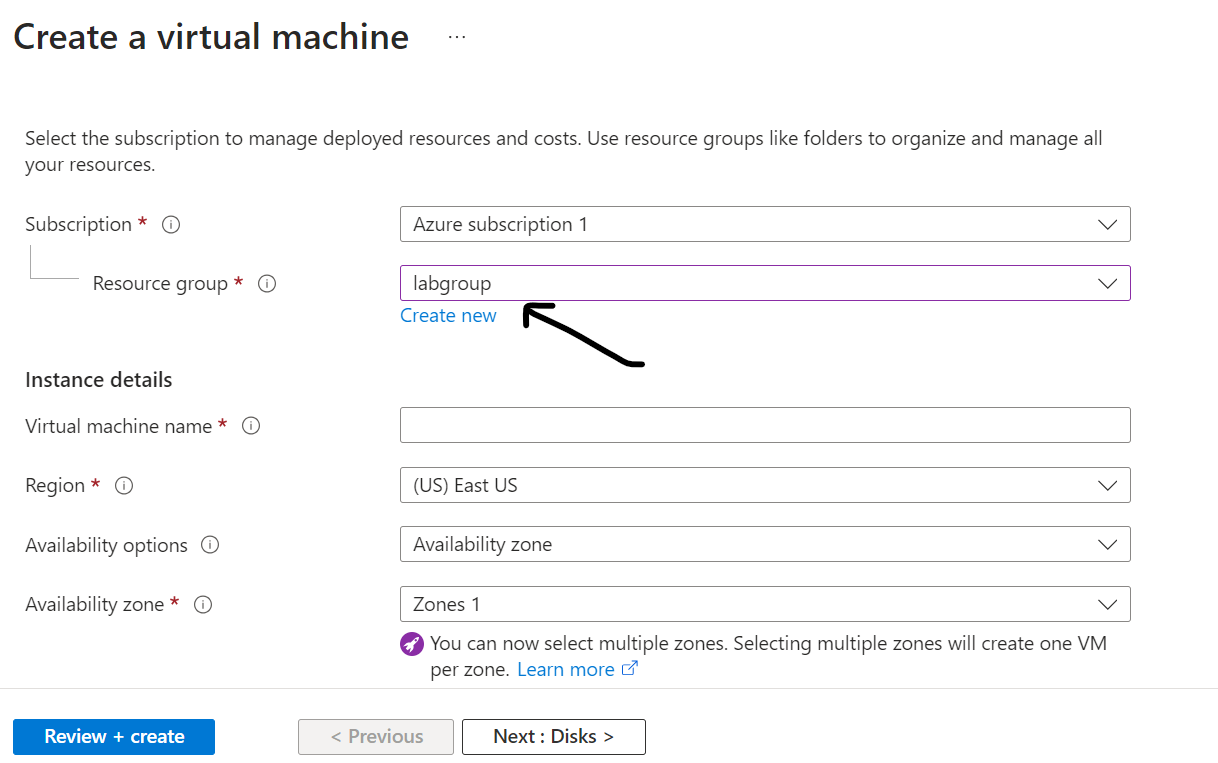
**Step 3: Deploy a Virtual Machine**

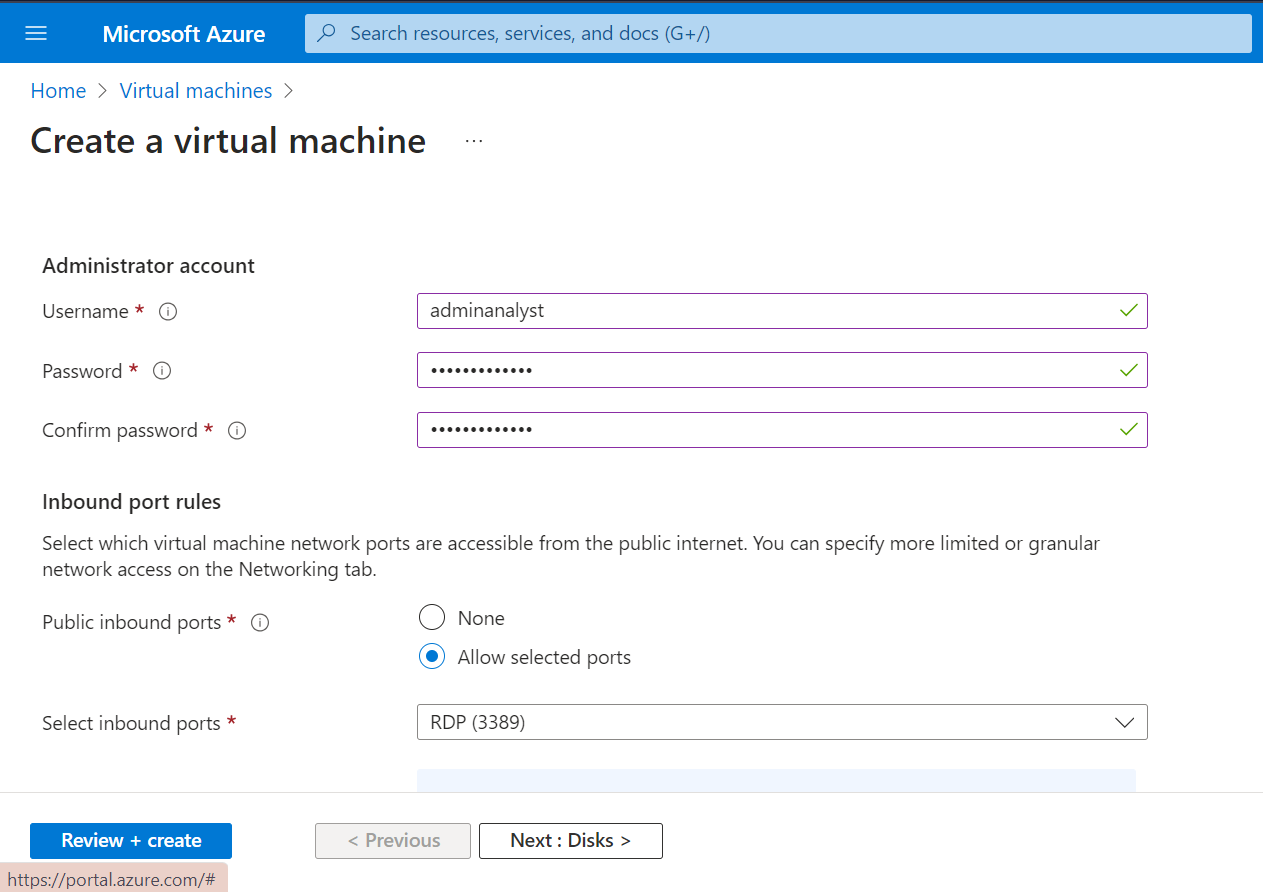
In this lab, we will be collecting our data from a Windows Virtual Machine. To deploy a Virtual Machine in Azure follow the following steps.

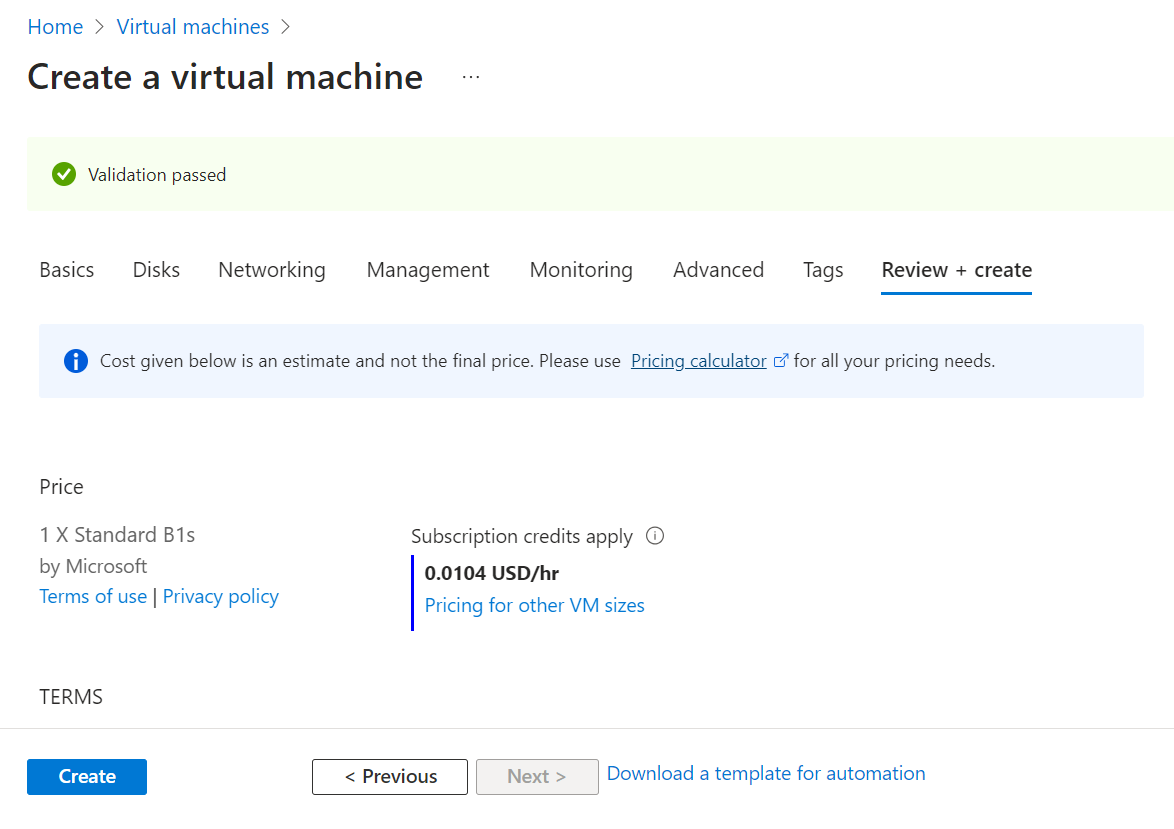
Click Create:



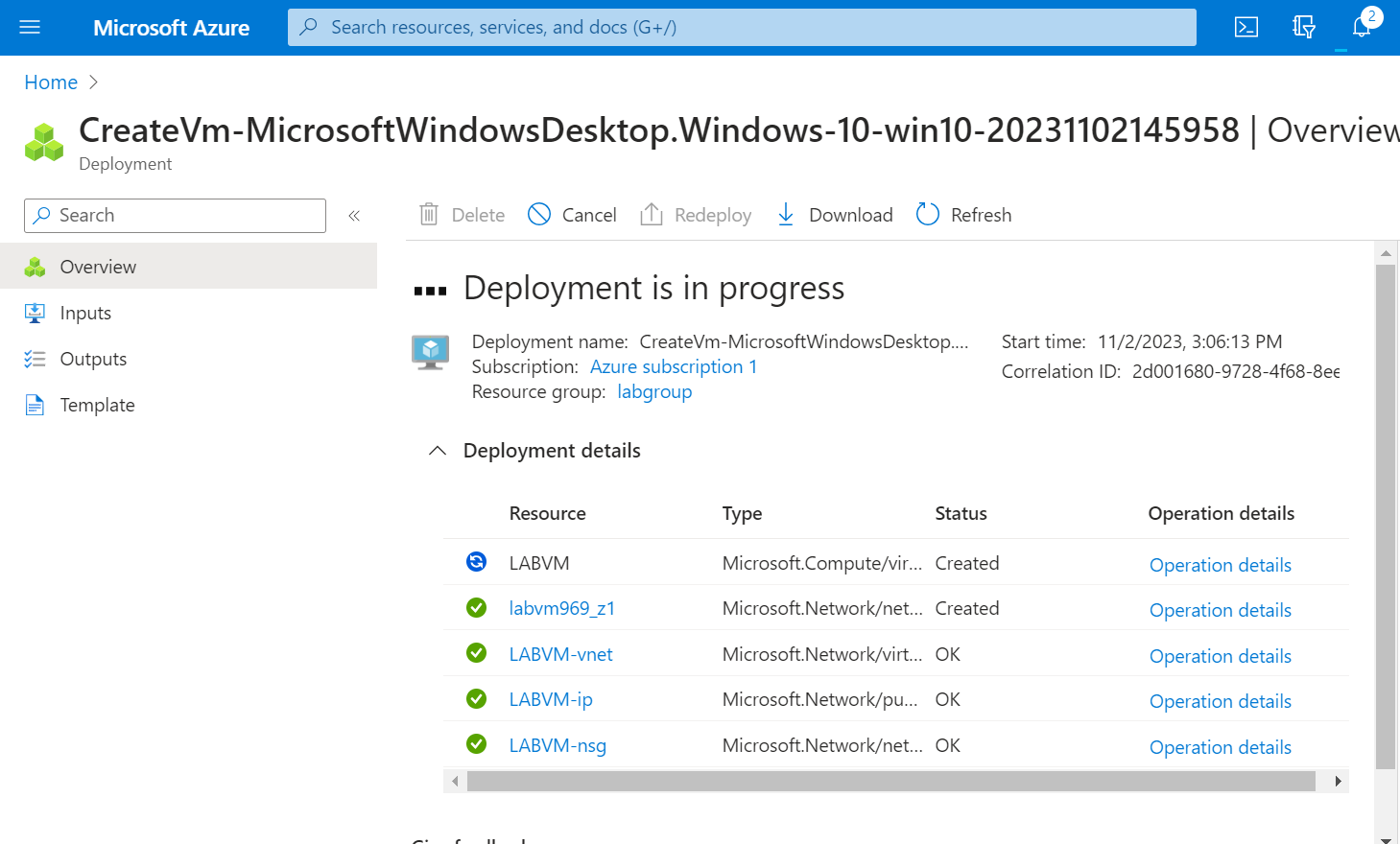
Use the resource group created in the first step and fill out the required field to create your virtual machine.







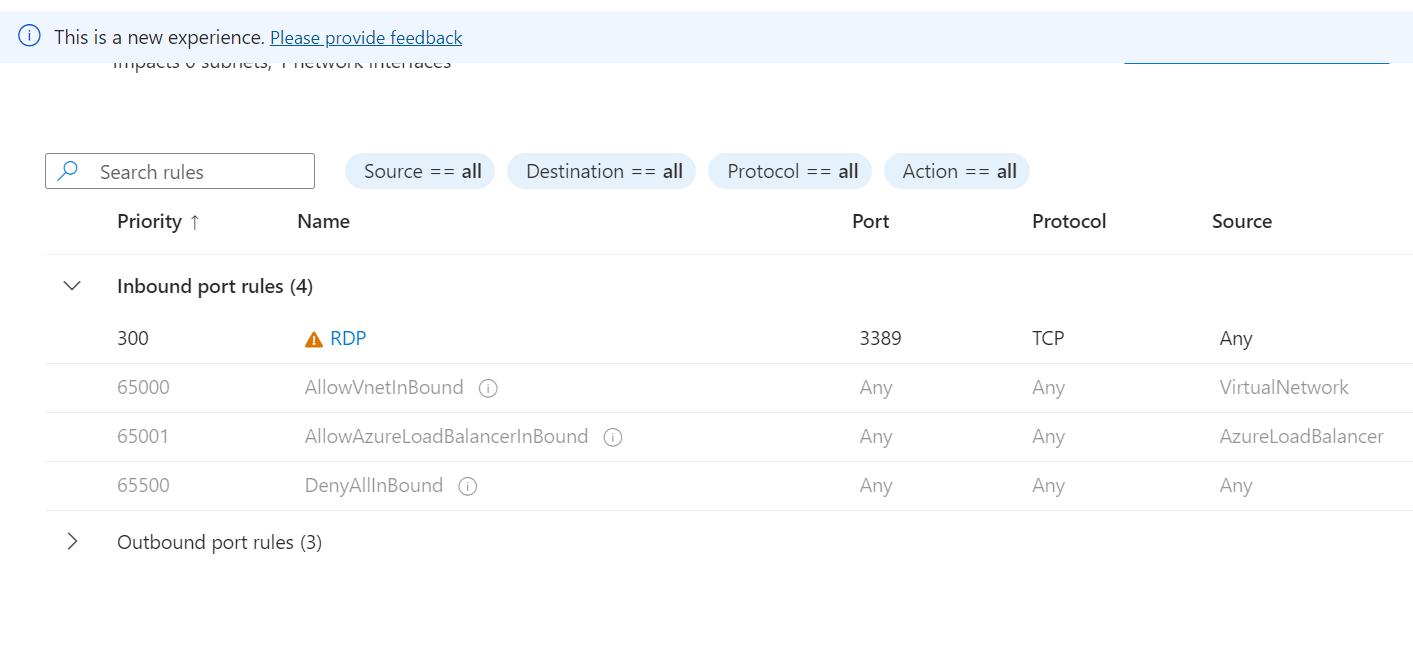
This is what occurs when deployment is succesfull



## Part 2: Network and Virtual Machine Security

When you deploy a virtual machine in Azure, that virtual machine is placed on a Virtual Network (vnet). Your Virtual Machine is assigned an IP address on that network as well as a network interface. Another Azure security feature that is implemented with the default settings we used are Network Security Groups (NSG). 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 deny.

If we go to back to our resource group we created earlier we can see the Virtual Network and NSG listed as resources. If we select our NSG we can see the default rules.



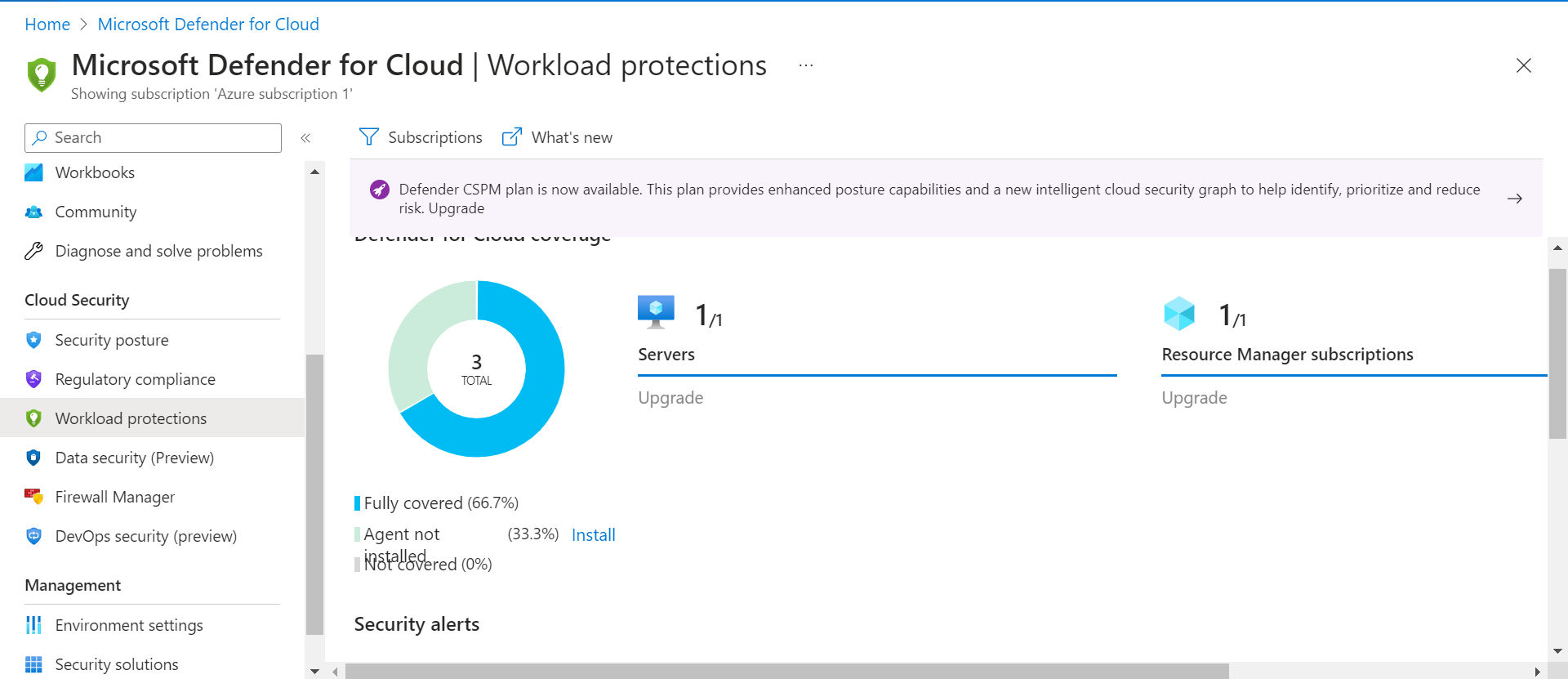
If you look at the first rule, you will see that inbound RDP traffic is allowed from any source to any destination. RDP is necessary to access our VM. However, with this current setting, anyone who obtains our public IP (which can be possibly be obtained via network scan) can potentially connect to the VM as this is public facing. This presents a security risk as it makes the VM vulnerable to possibly a brute force or password spray attack.

In order to reduce our attack surface, we need to enable a security feature called **Just in Time access**.

Essentially what this feature does is only provide access to our Virtual Machine when necessary via time-based restrictions as well as implements the **principle of least privilege**by giving the option to restrict access to certain IP’s as well as RBAC roles.

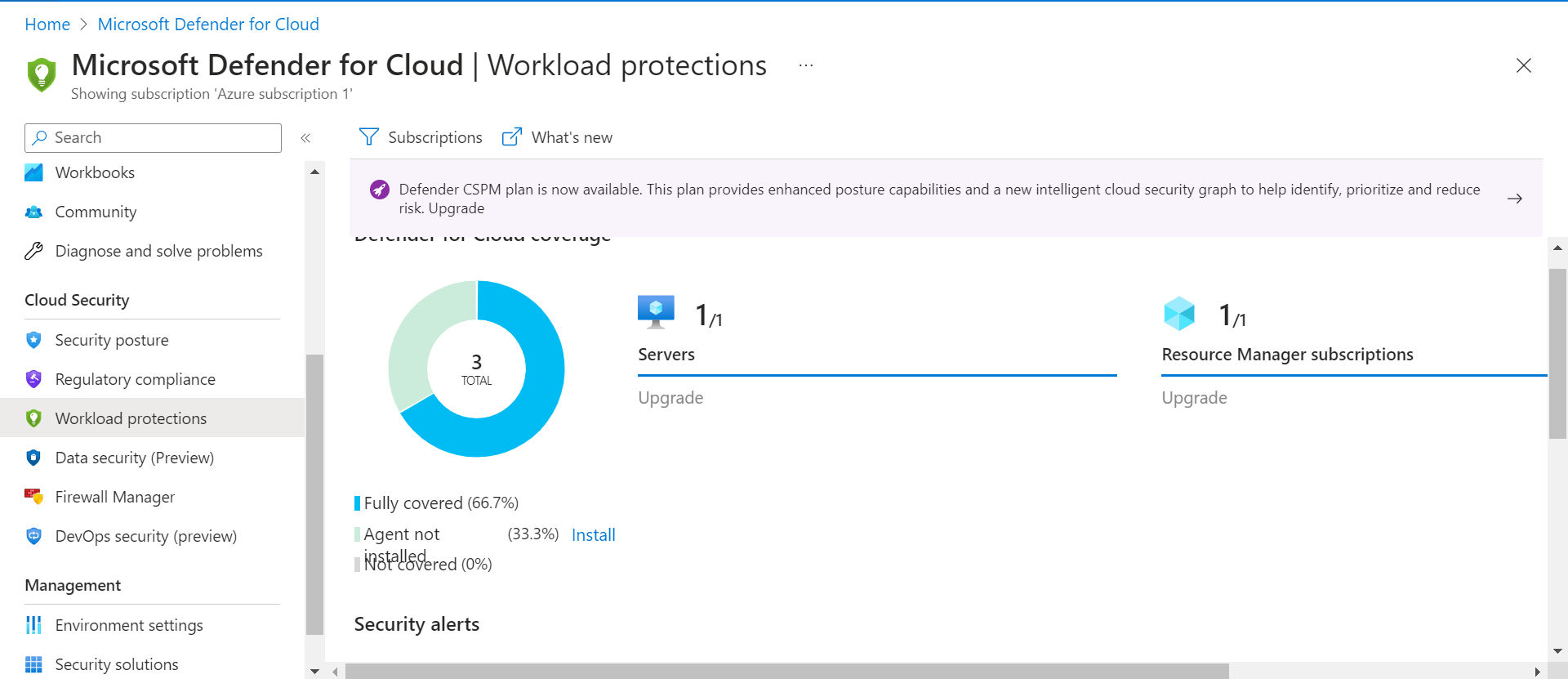
Anyone who wants access to the VM will need to request and based on their IP and assigned role they would be granted or denied access. By default when creating your Azure Account you are a Global Administrator so upon request you will be granted access to the VM. To set this up we will perform the following steps:

Search Microsoft Defender for Cloud in the search bar at the top of the Azure Portal and select the service. You will see a page similar to this. On the left pane select “environment settings”

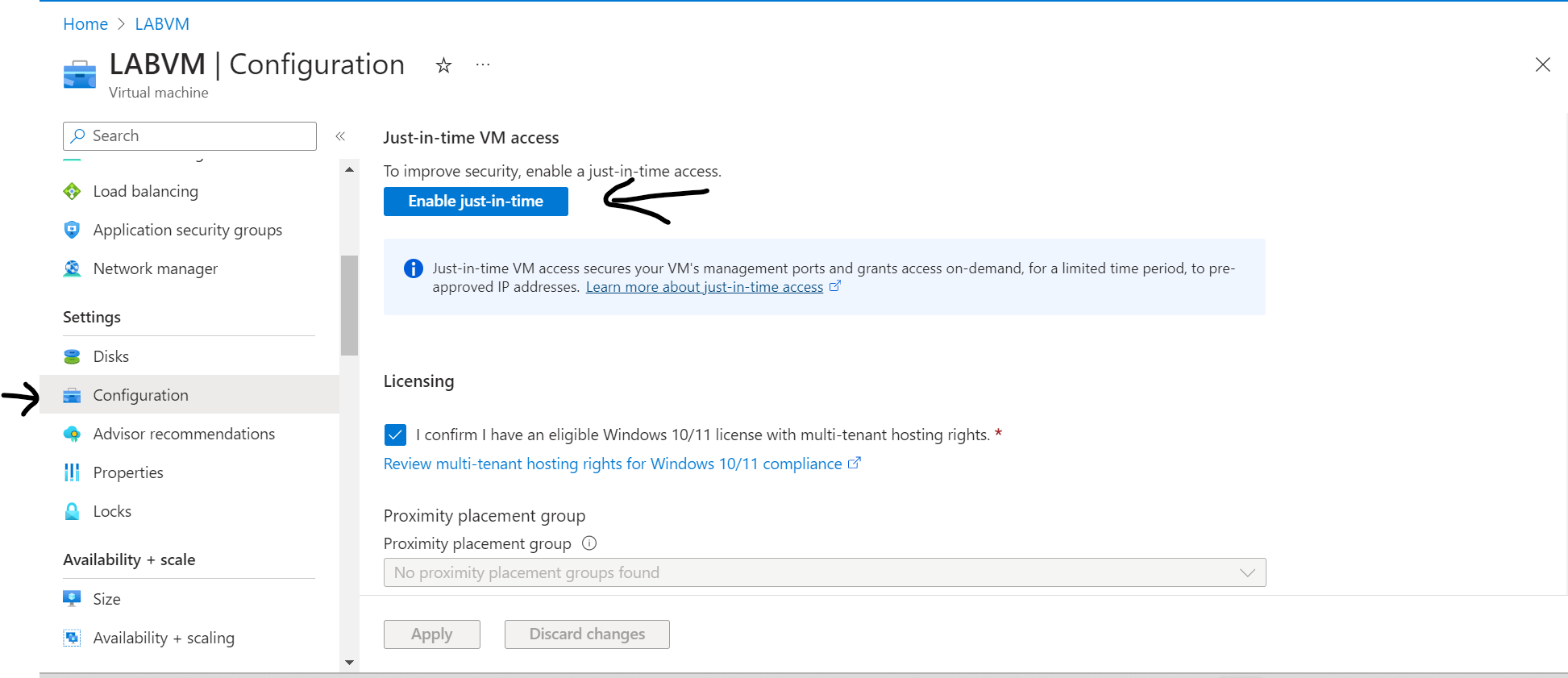


Select your Azure Subscription from the list provided. Upon selection, the following services will be seen on the screen. By default, Enhanced security is off but you will want to select the Enable All Microsoft Defender for Cloud Plans. You will be given a 30 free trial so be sure to disable when finished with the lab to avoid any cost. You can select then “Enable all option” and hit save.

After enabling the plan, Navigate back to the homepage for Defender for Cloud and select Workload Protections on the left pane. That will result in the following screen.

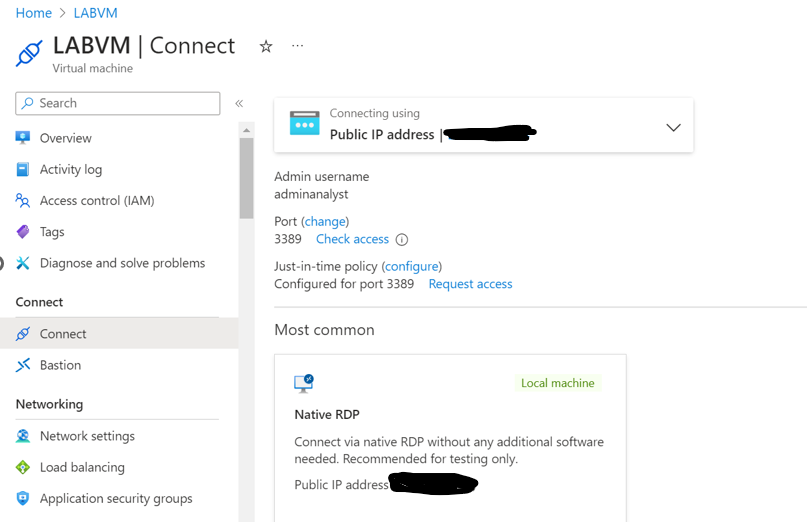


In the Advanced protection section select “Just in time VM access” and select “Enable JIT on VM”. Make sure to select VM you created from earlier steps.

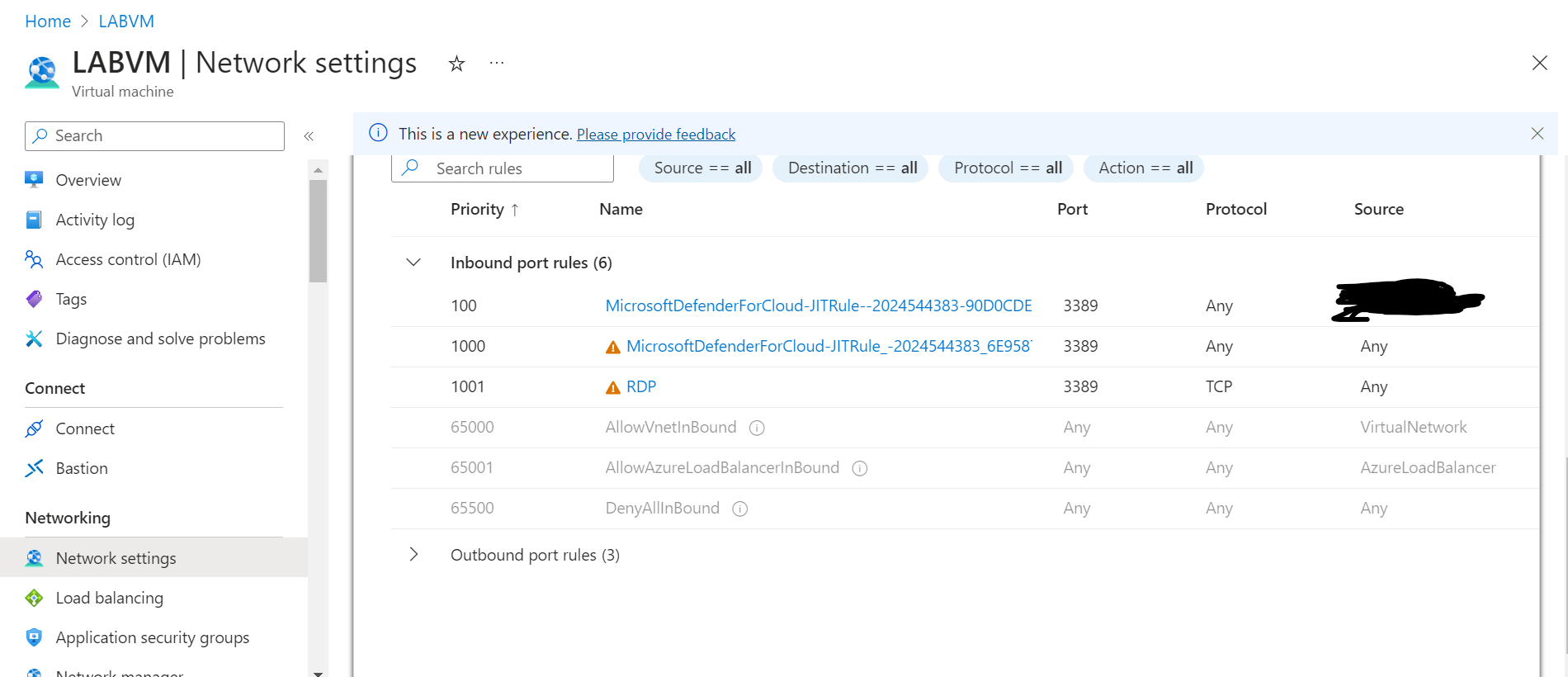


Click save. Just in Time Access should now be enabled.

Go to your VM settings and click Connect on the left pane. Select “Request access”.



If we go to the networking tab for our VM we can see our rules have been updated. Now RDP traffic is allowed for a certain amount of time only from the IP of your computer. Anyone else who attempts to establish and RDP connection will be blocked via our Just in Time Access rules.



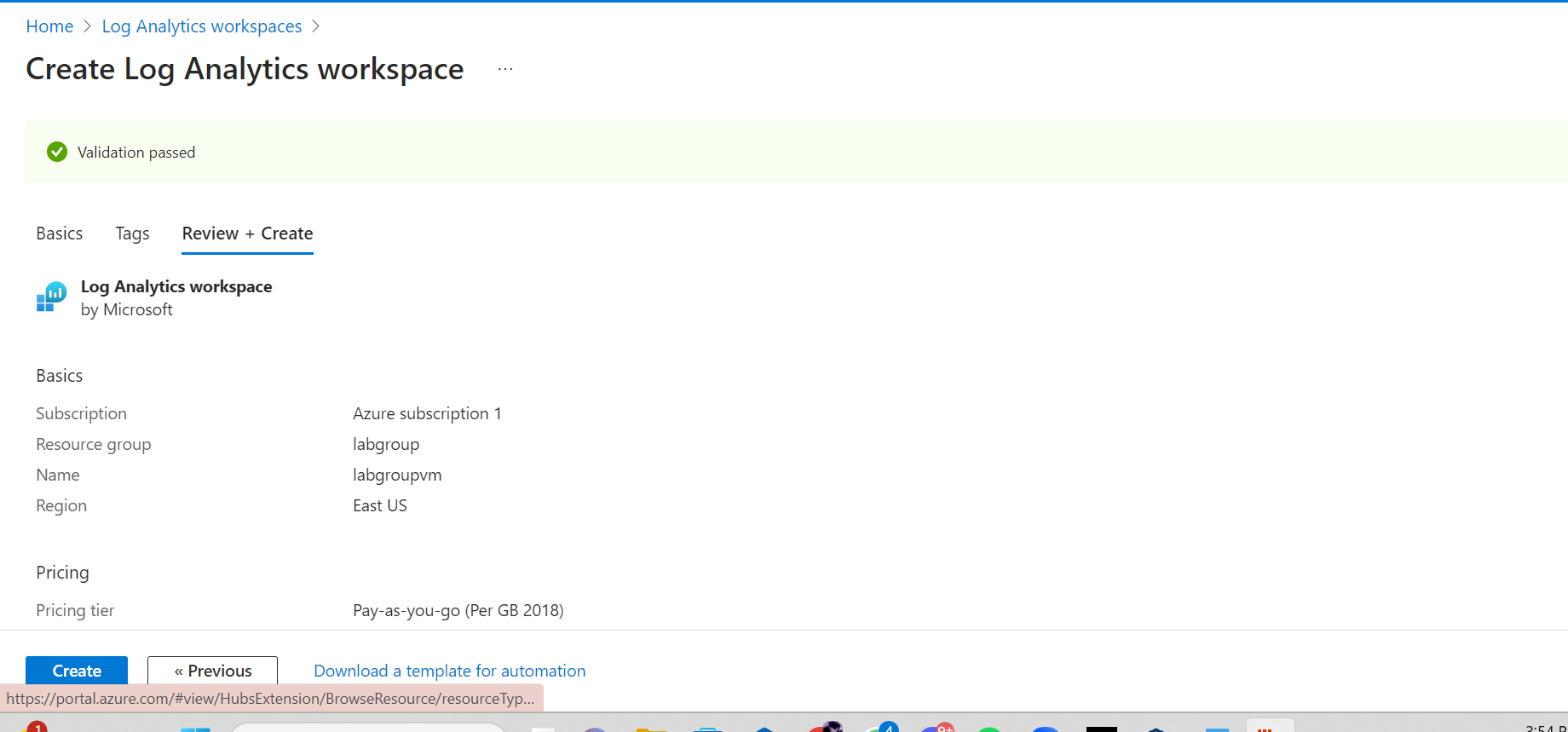
**Step 4: Create Log Analytics Workspace and Deploy Sentinel**

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.

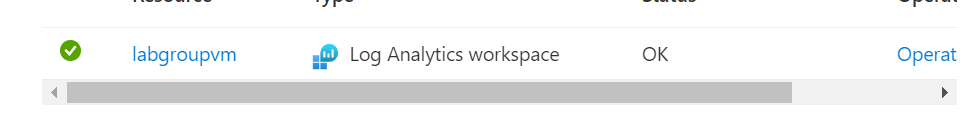
To configure a Log Analytic workspace:

Search “Microsoft Sentinel” in Search Bar in Azure Portal. This will prompt you to create a Log Analytics Workspace.

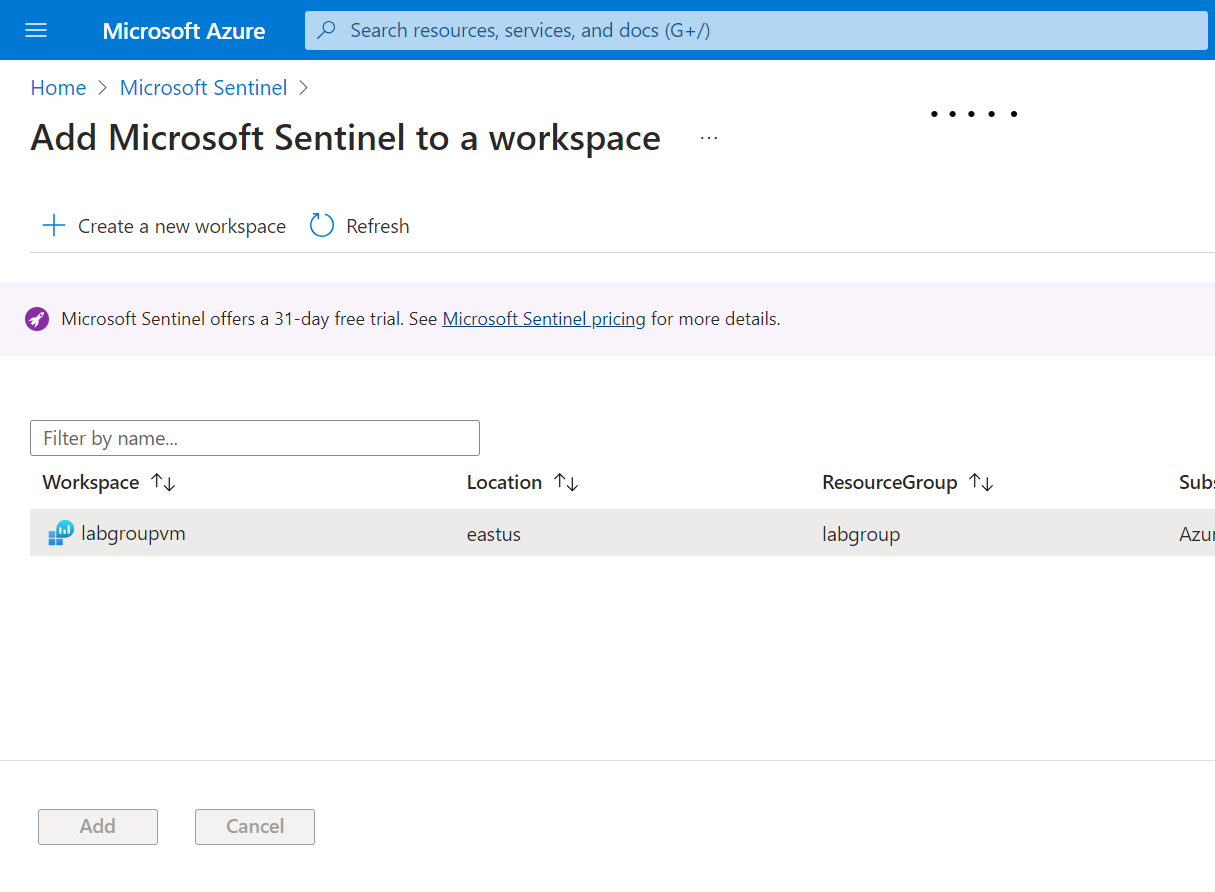
Use the same resource group used for the Azure Virtual Machine you created in the previous step when filling out the necessary fields to create your Log Analytics workspace.



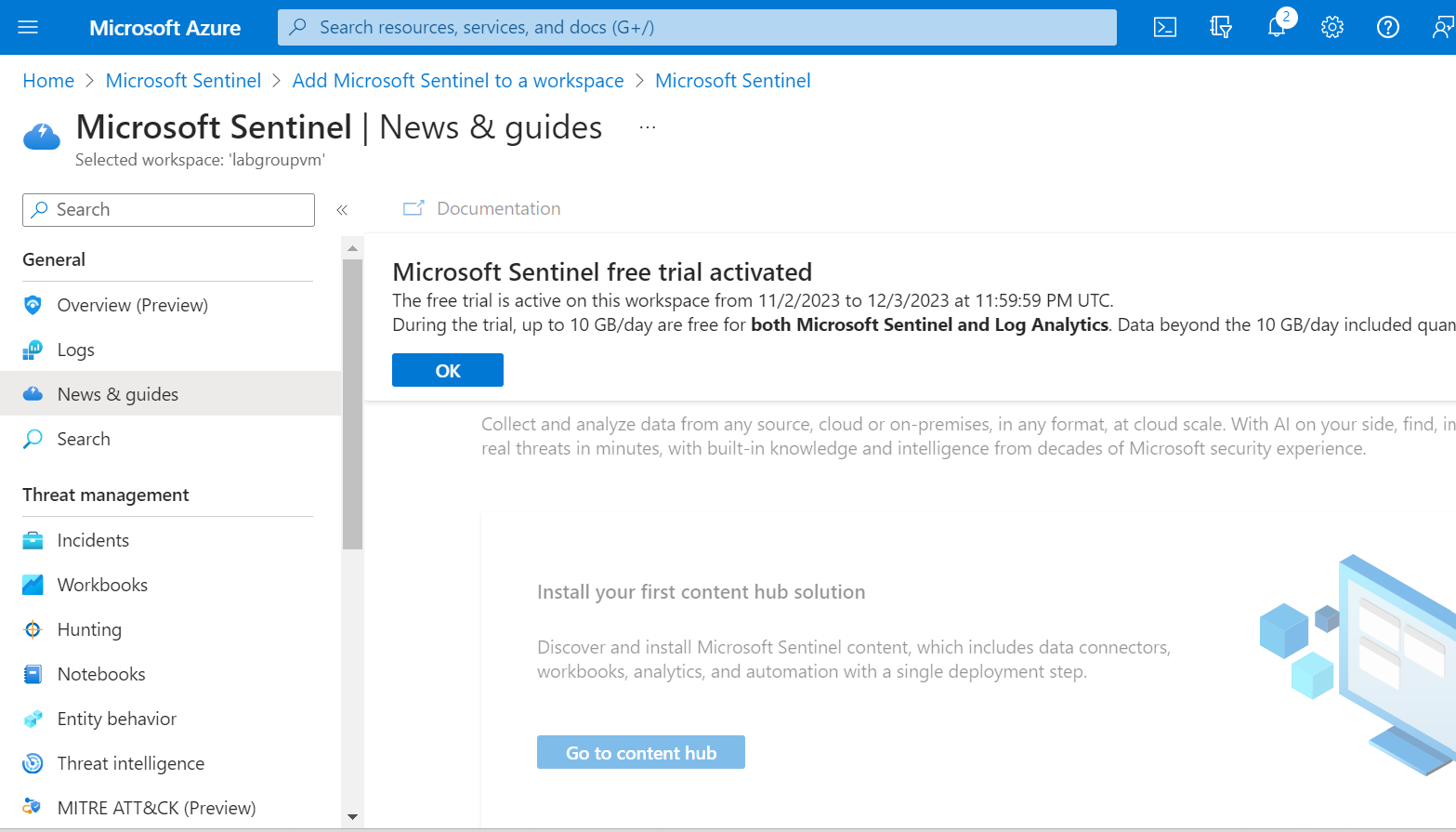
After creating the Log Analytics Workspace search Sentinel in the search bar



Scroll to the bottom of the page and select Add.

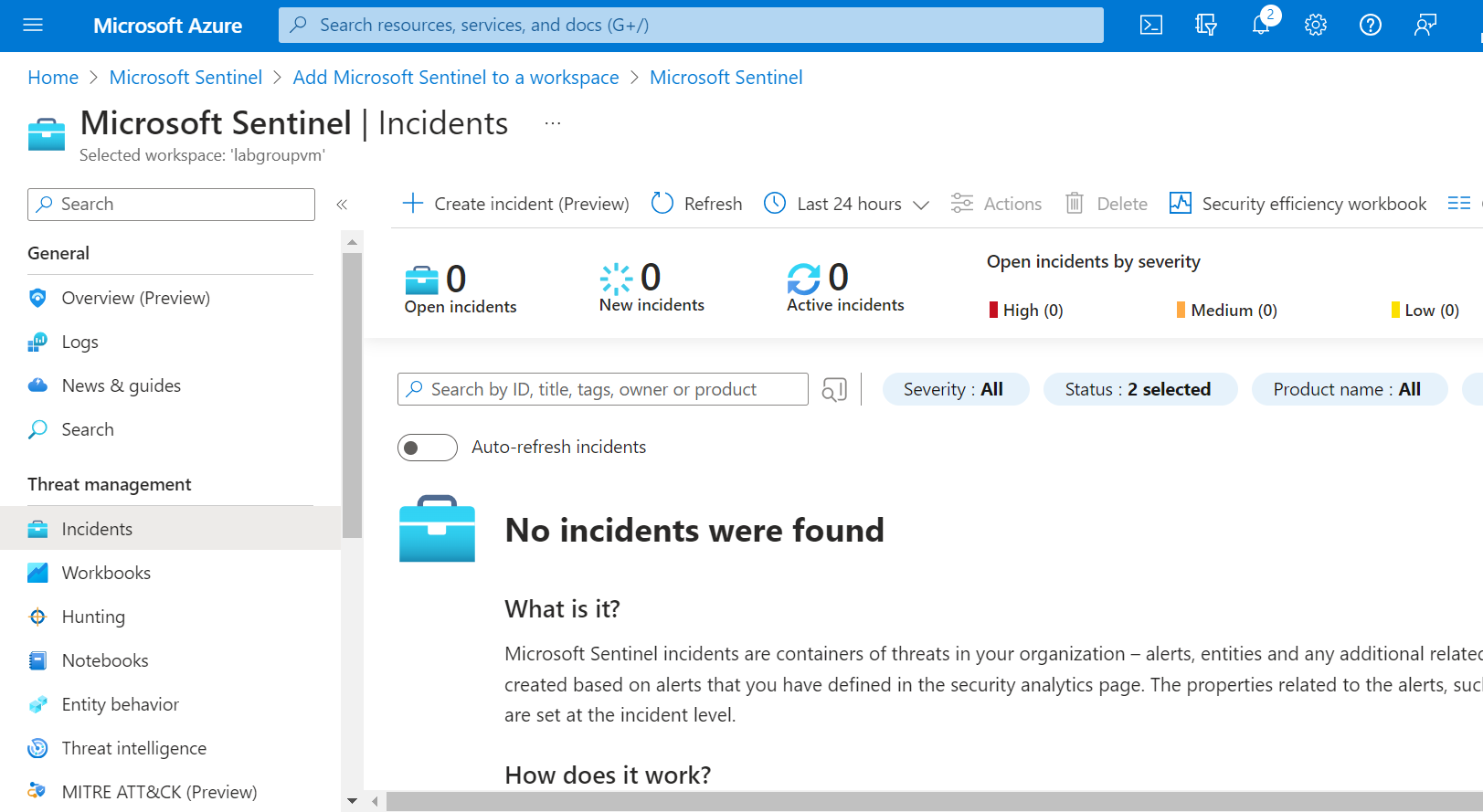


After clicking create, you should see this screen upon deployment :



# ***PART 3: Utilize Data Connectors to bring data into Sentinel for Analysis***

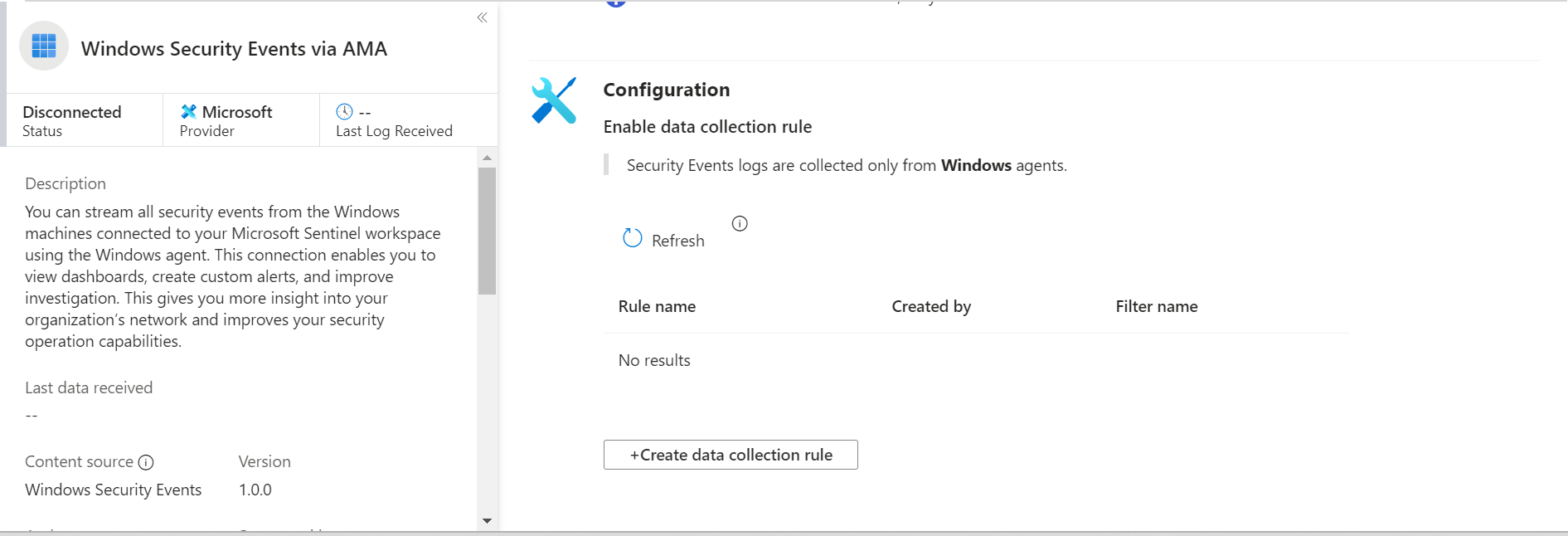
After the Sentinel Deployment, if we go to the incidents tab on the left we see we don’t have any incidents currently as there is no data being fed into sentinel. We are next going to utilize data connectors and create a data collection rule to bring in data from our Windows 10 VM.



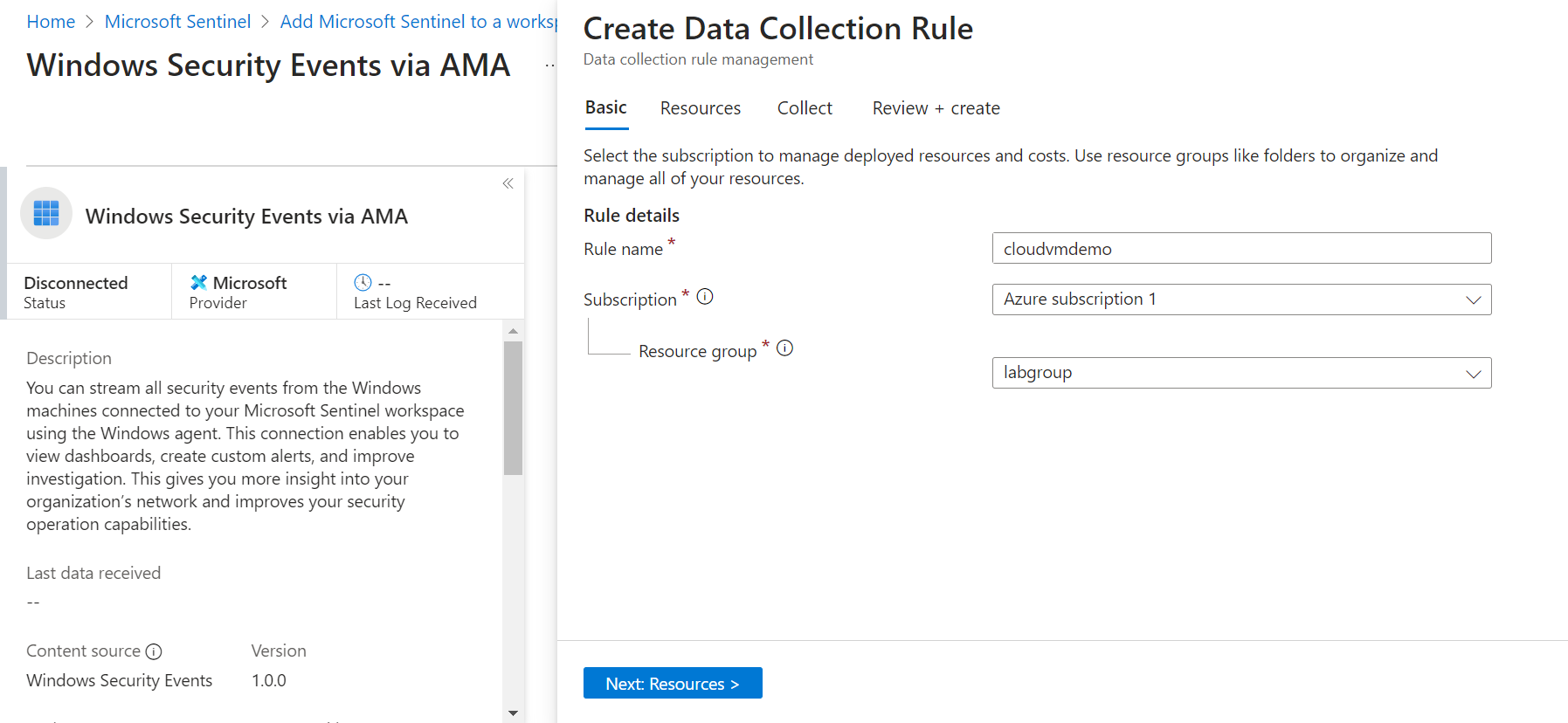
Go to the Data connectors Tab. This is where we can select the type of data that we want to bring into our SIEM.

In the Search bar type in “Windows” and you will see Windows Security Events via AMA. Select that option and click “Open Connector Page”.

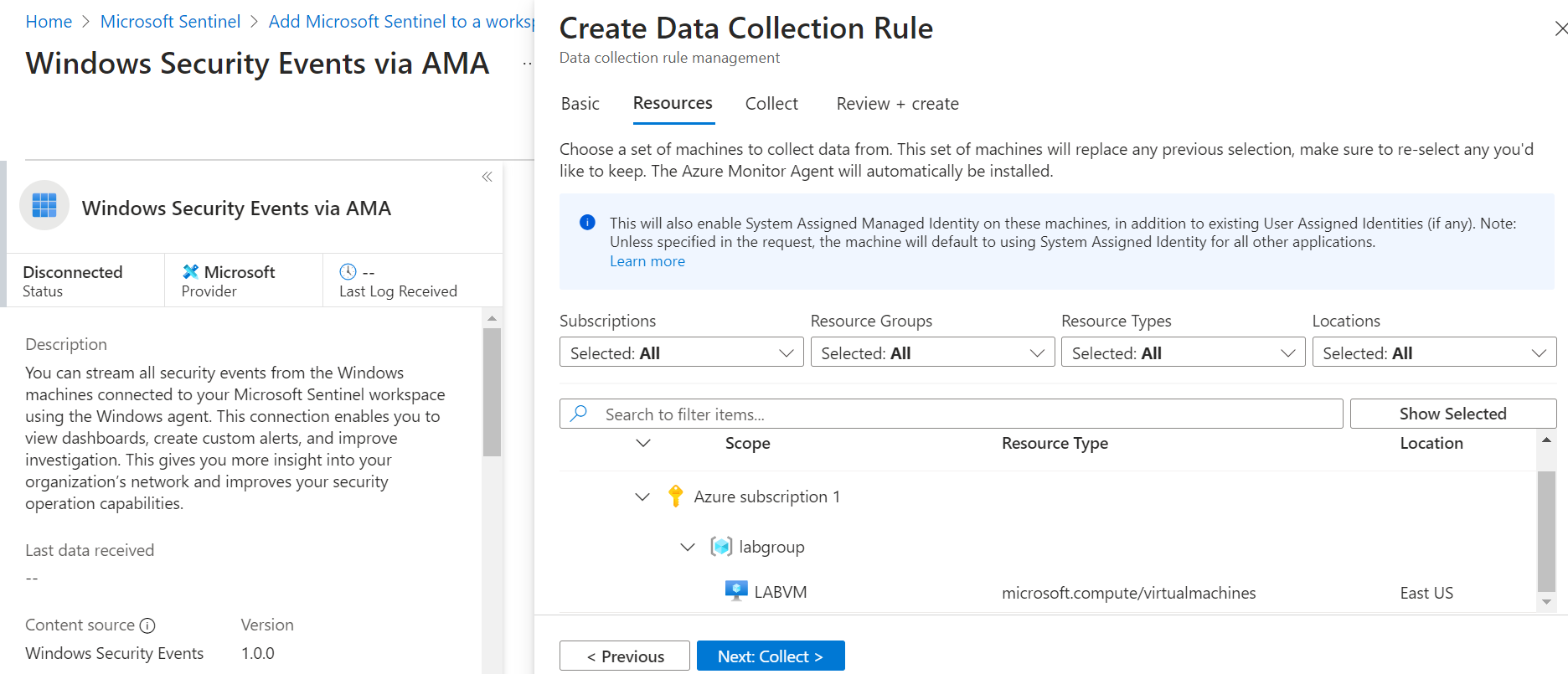
Click “Add Data collection rule.”



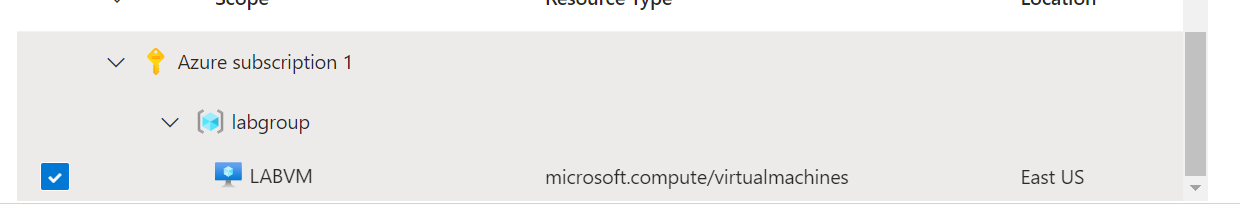
Give your rule a name and connect to it your resource group we have used for all resources thus far.



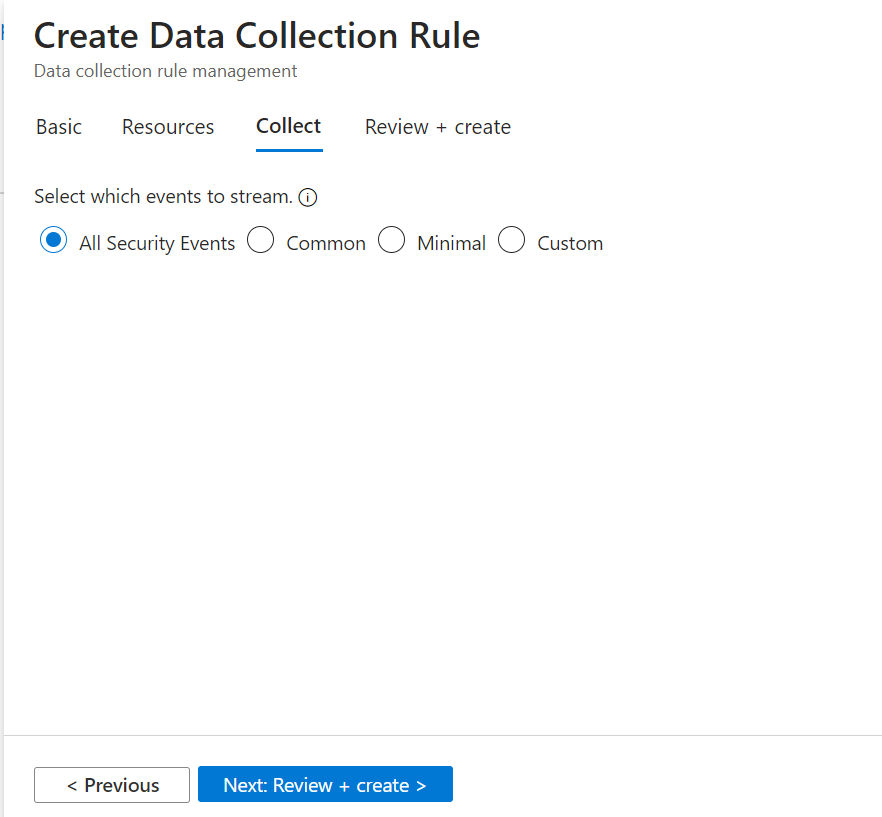
Click “Add resources”.



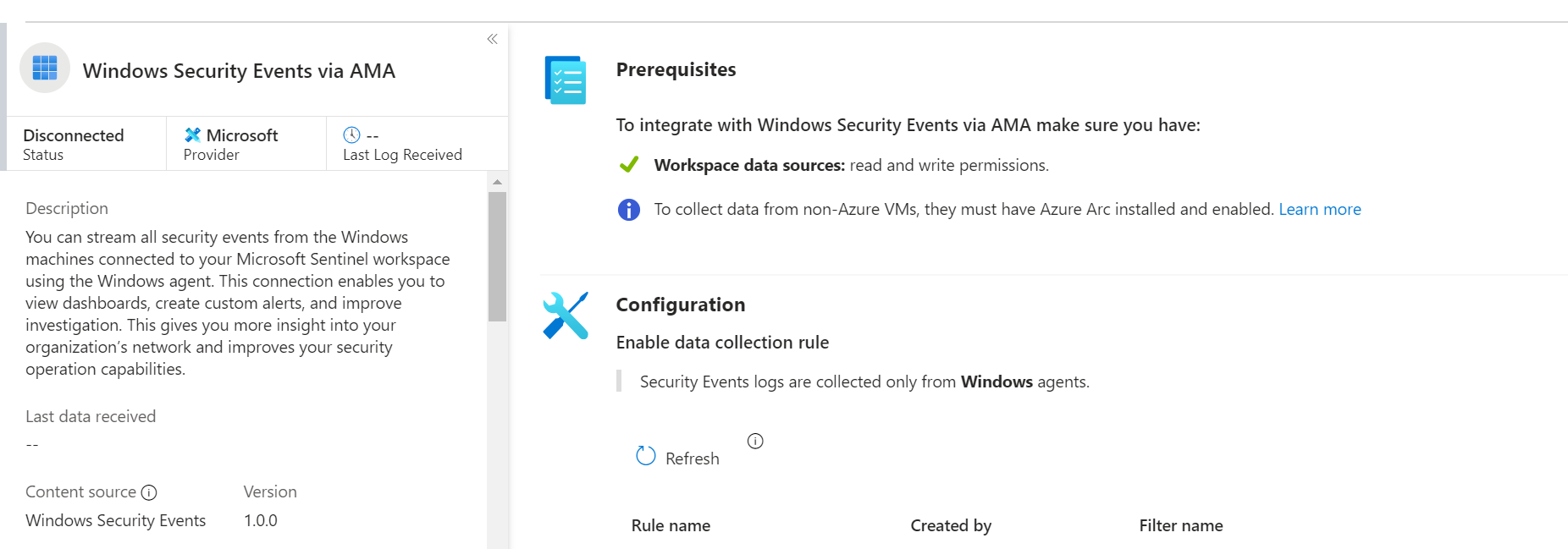
Select the Virtual Machine created in Step 2.



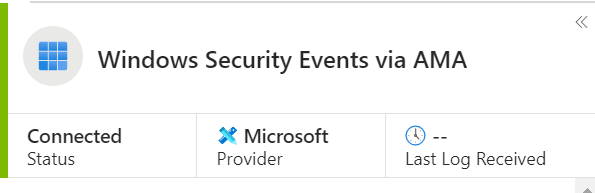
Select “All Security Events”



The data collection rule should have a “Validation Passed” screen.



Refresh the page and it should say “Connected”



## Part 4: 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.

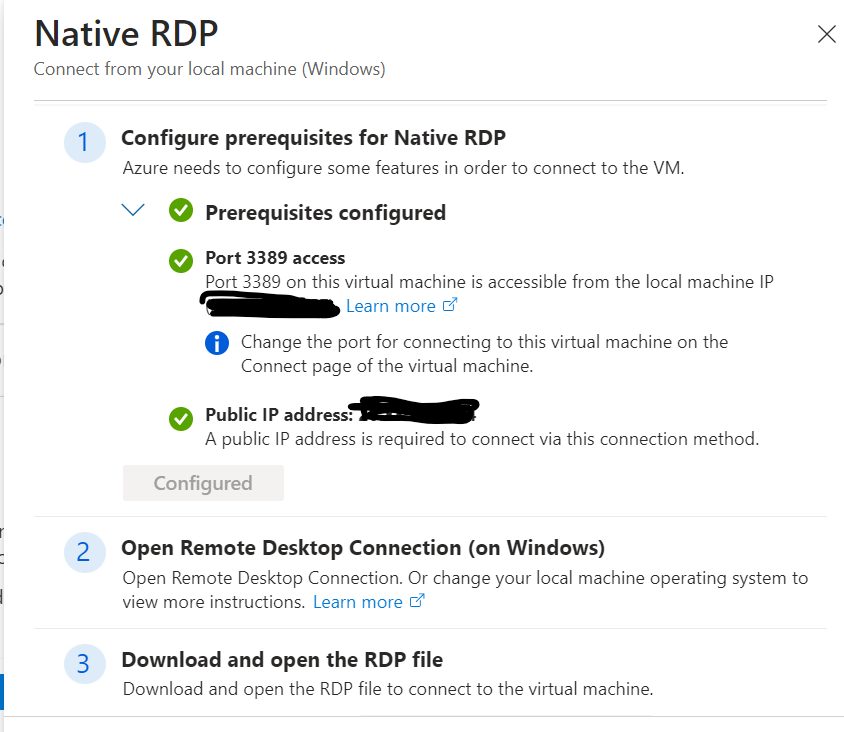
Utilize the Azure Portal to navigate to the VM created earlier in the lab.

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

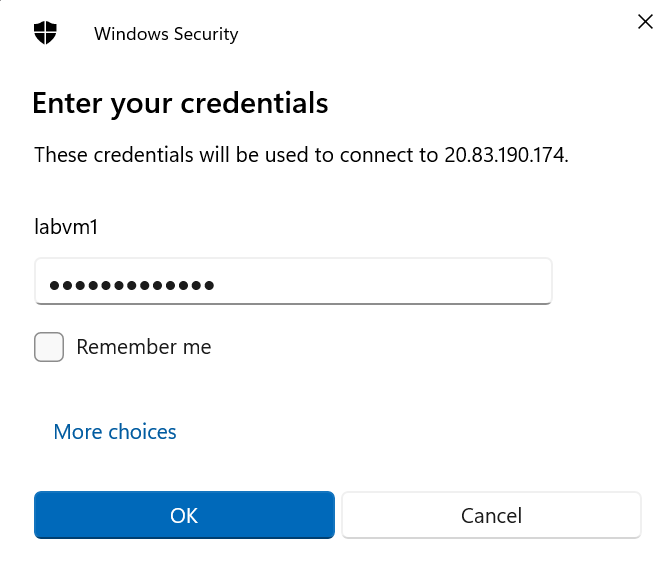


Under Networking, you are given a public IP. Use an RDP on your PC Client such as Remote Desktop Connection to access your VM by entering in the public IP address.

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

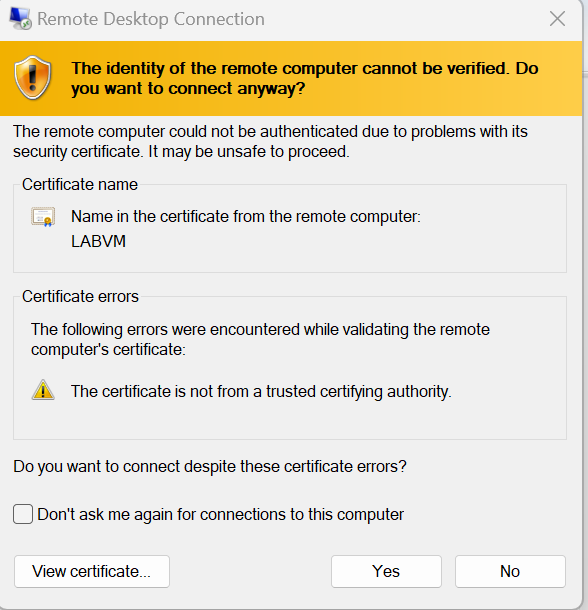






After entering the username and password of the vm you created at step 2

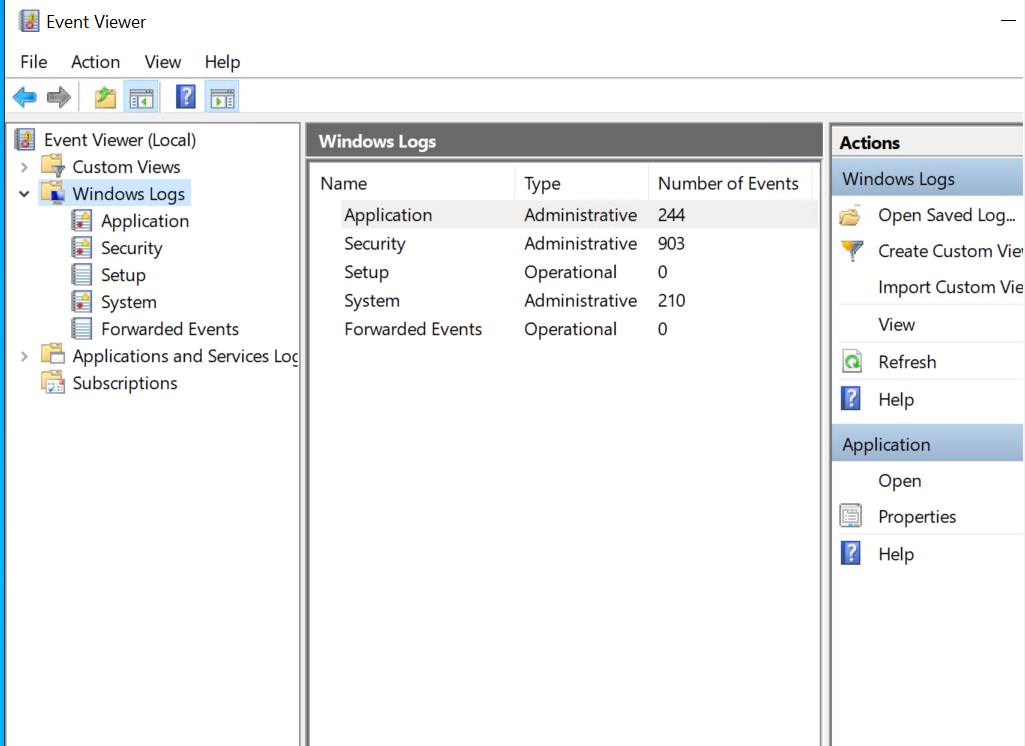
You will be prompted to this screen



Click yes

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

As you can see there are several types of logs Windows Collects. **Application logs**,**Security Logs**, **Setup**, **System,**and **Forwarded Events**.

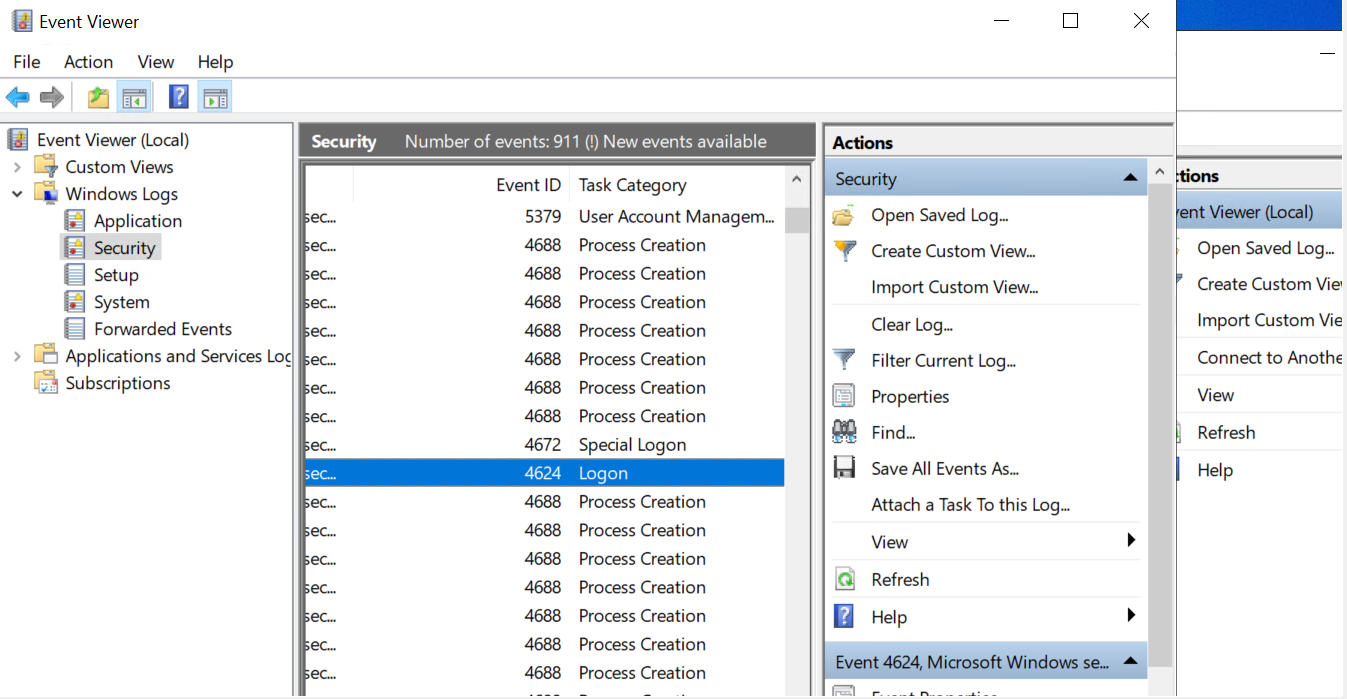


Our focus in this lab will be on Windows Security events.

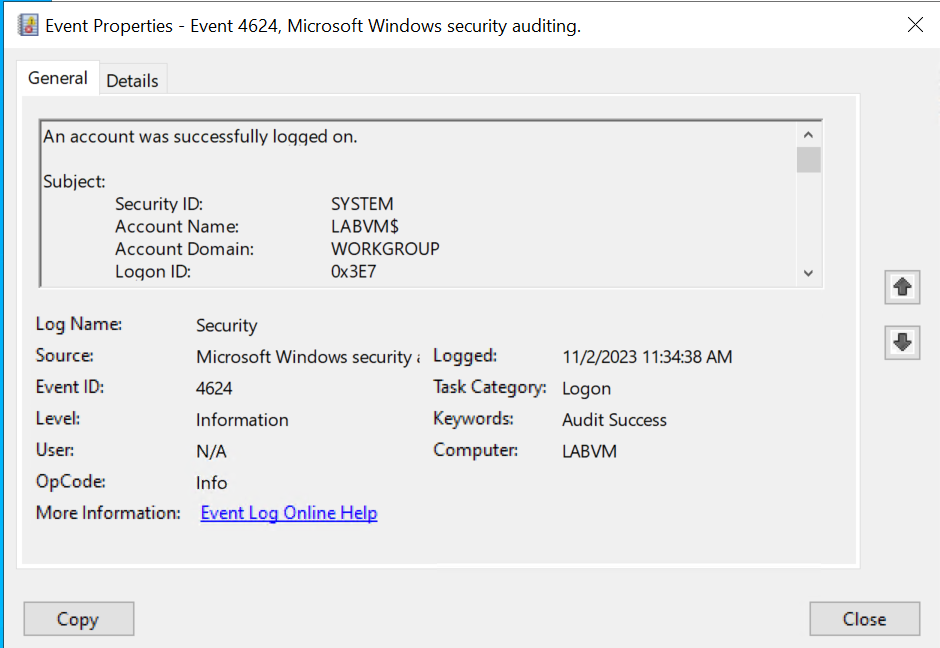
Click “**Security**” and observe the events.

As you can see there are several security events in event viewer. Let’s drill into one of these events.

Use the find option and search for ***4624***



When we select event 4624 we see that 4624 ID is indicative of a successful logon. We can also examine more detailed information about the logon if need be.



## Part 5: Kusto Query Language

The purpose of a SIEM such as Azure Sentinel would be to bring data like this into a centralized location. In an enterprise, we would want data coming from all our endpoints and virtual machines to make it easier for an analyst to get the information that is needed quickly.

Let’s go back to Azure Sentinel and pull this event from our Sentinel Logs.

In the Sentinel, Main Page click “Logs”

In the section where it says “Type your query” we are going to use the following KQL (Kusto Query Language) Logic:

SecurityEvent

| where EventID == 4624

| project TimeGenerated, Computer, AccountName

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**. While there are many different syntax rules and ways to construct queries in KQL we will be using a few basic KQL commands to extract the data and write our analytics rule later in the lab.

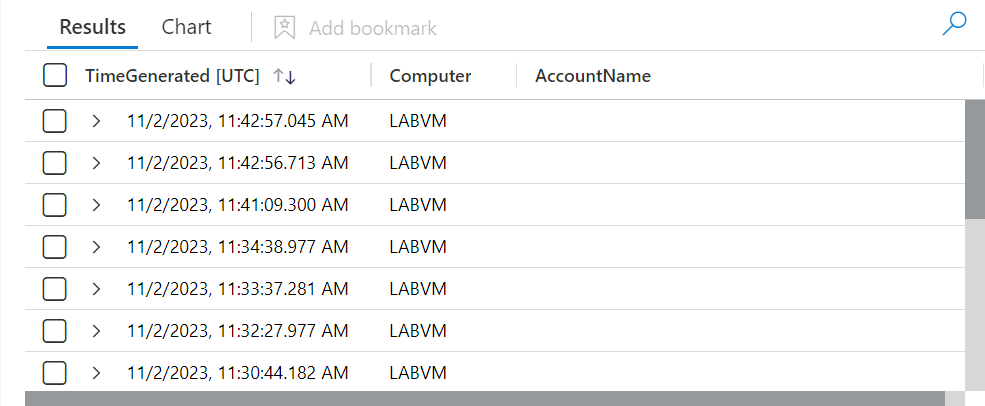
Let’s break down the meaning of this query

**Security Event** 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.

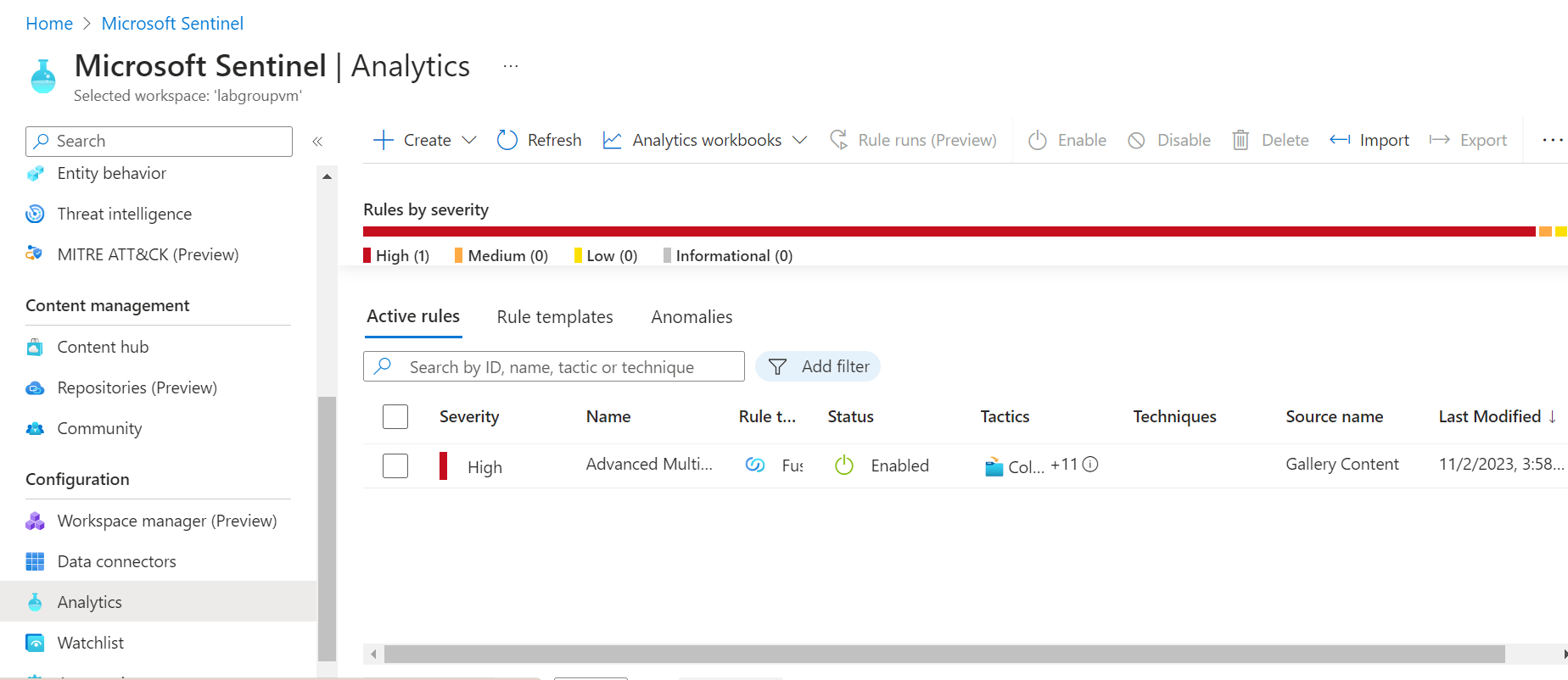
When the query is run we get this result.



As you can see, 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 a sentinel is not automatically putting that data into that field. We will go over how to populate that field later in the lab when we create our analytic rule.

## Part 6: Writing Analytic Rule and Generating 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 so 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.

## Scheduled Task and Persistence Techniques

The final part of this lab is to create our own custom rule to detect potentially malicious activity on our VM. In 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 .

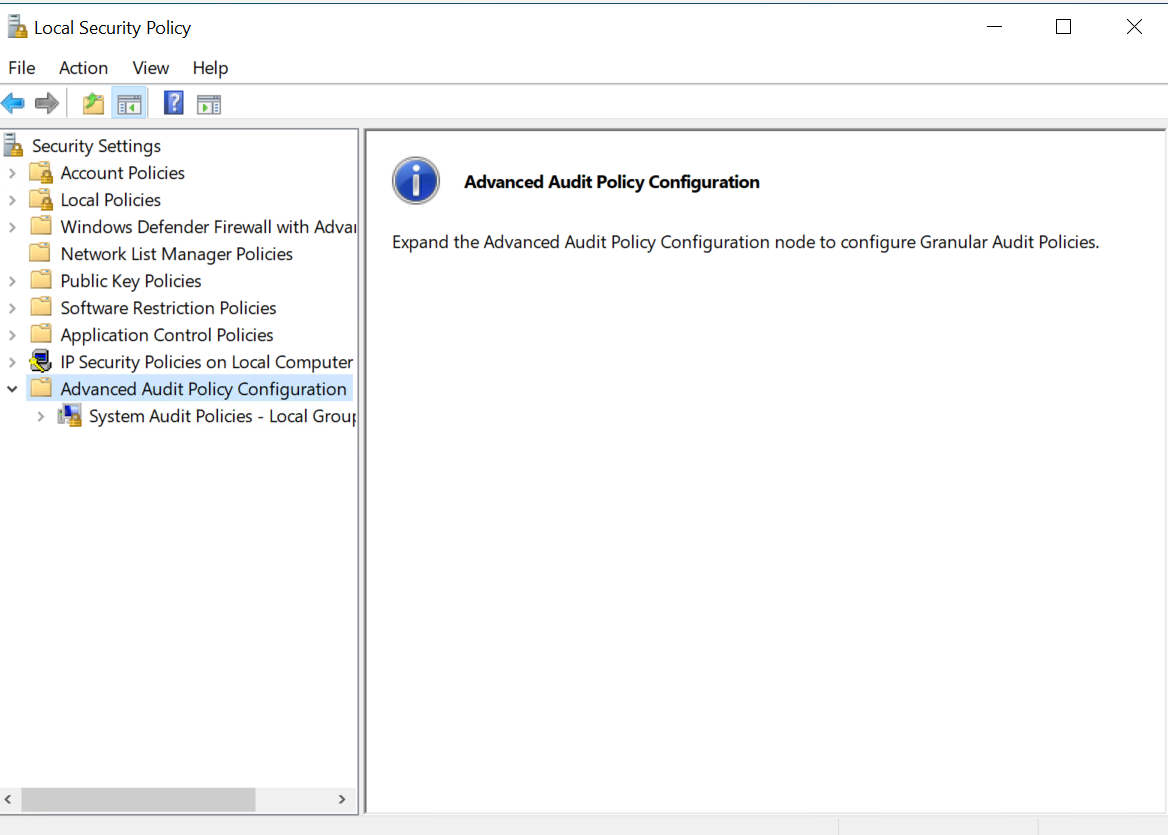
For instance, you could set up a scheduled task that opens google chrome at a certain time every day. While many times scheduled task can be a harmless event this can also be used as a persistence technique for malicious actors.

According to the [**MITRE Attack Framework**](https://attack.mitre.org/techniques/T1053/)**,** “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 lab, 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.

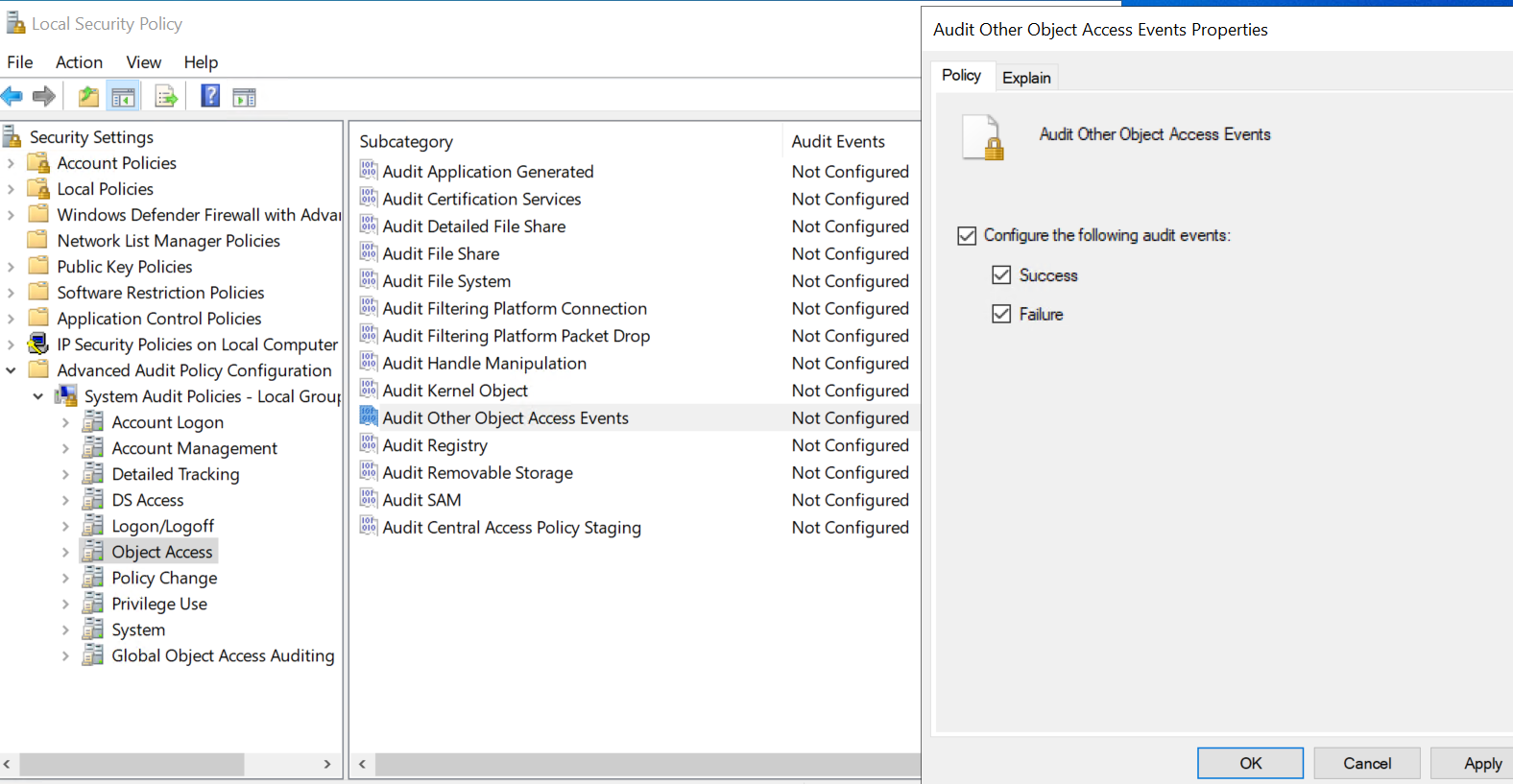
The Windows Security Event ID that corresponds to scheduled task creation is 4698. However, these events are not logged by default in the Windows event viewer. To enable logging for this event we need to make some changes to the Windows Security policy in our VM.

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”

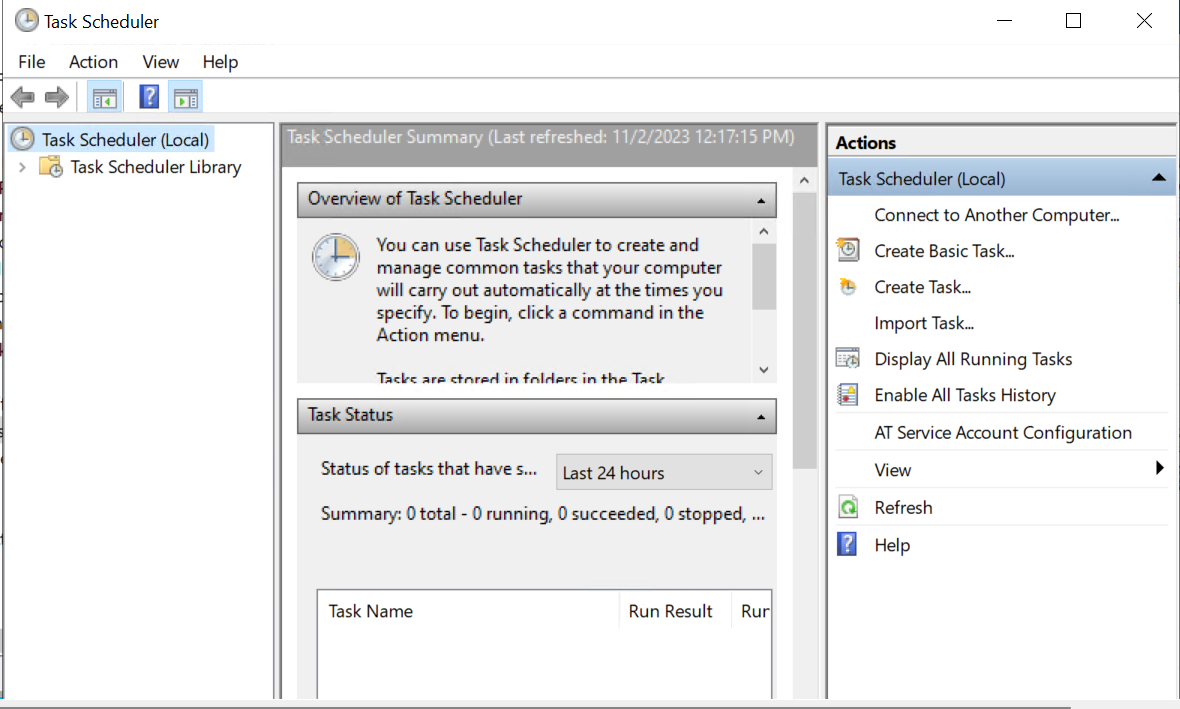
****

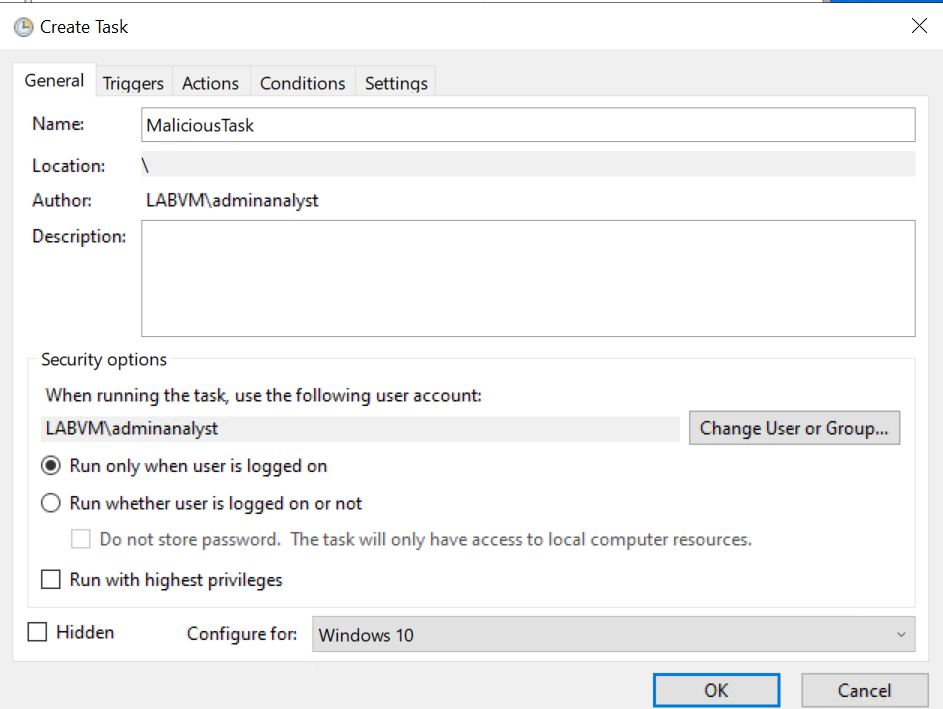
Logging is now enabled for the scheduled task event.

**Creating our scheduled task**

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.

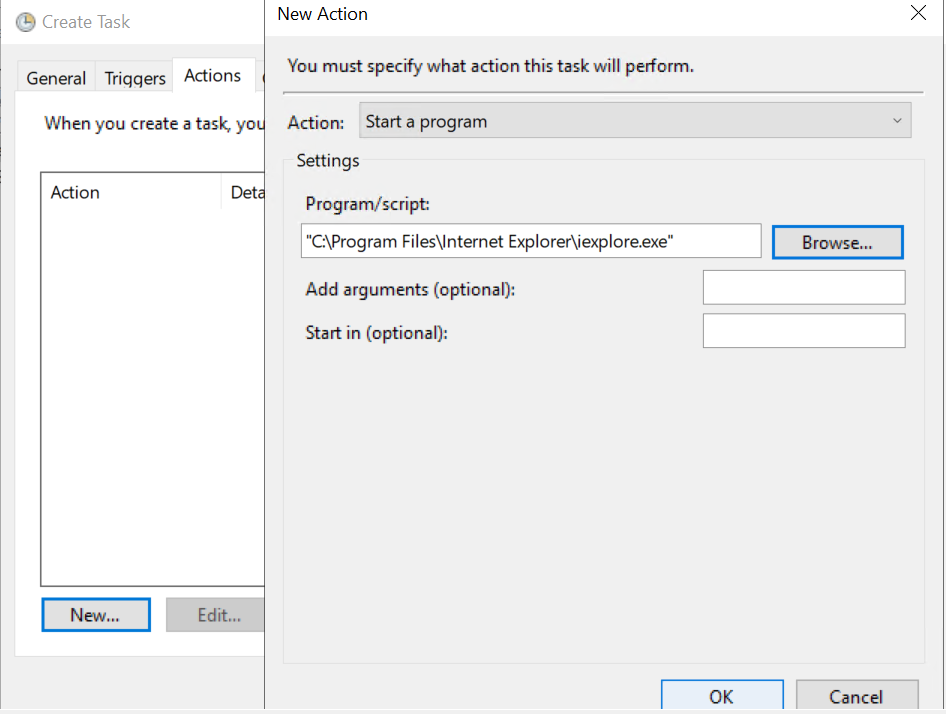
****

****

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

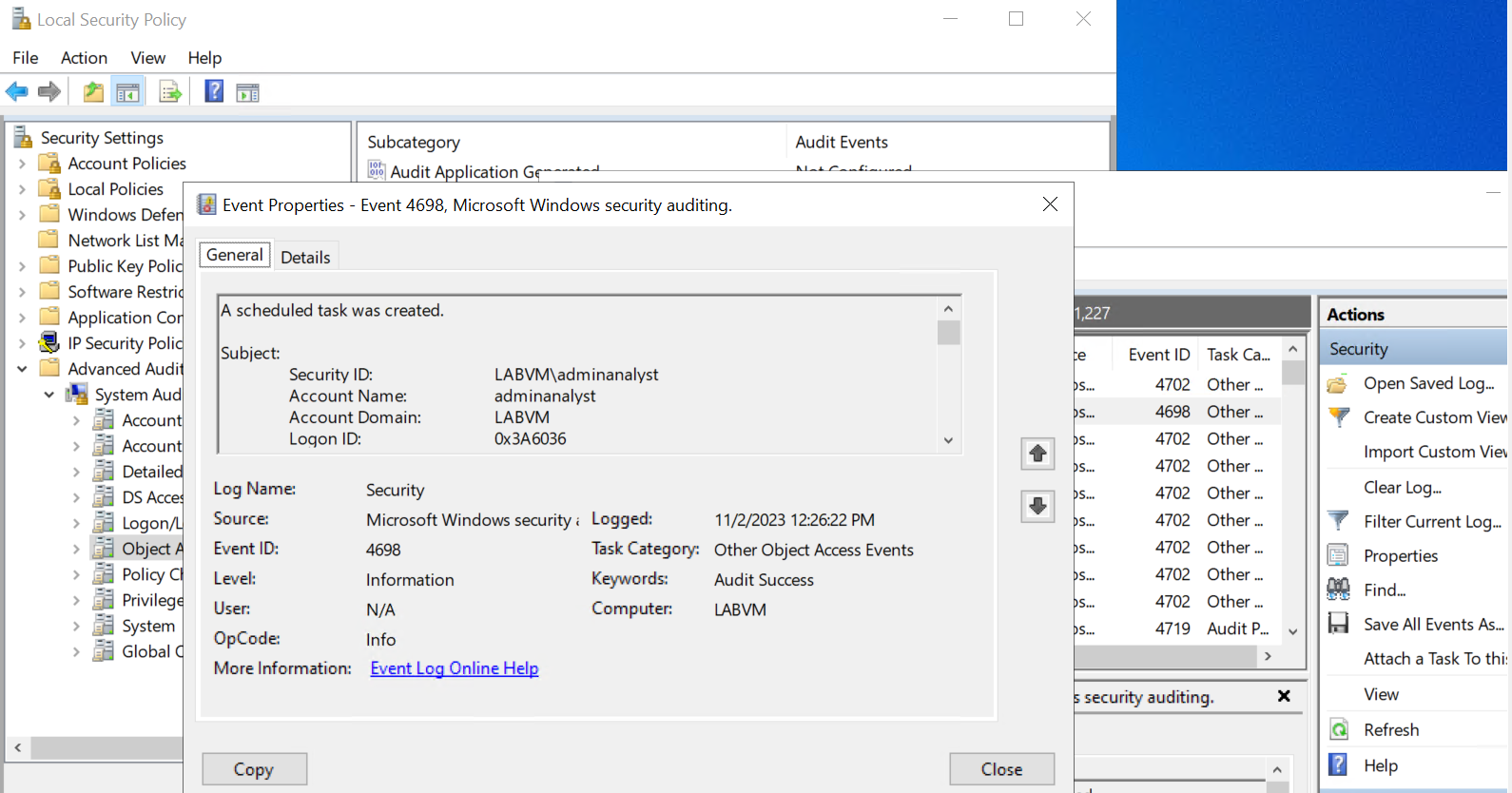
Navigate to the action tap and select start a program.

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

****

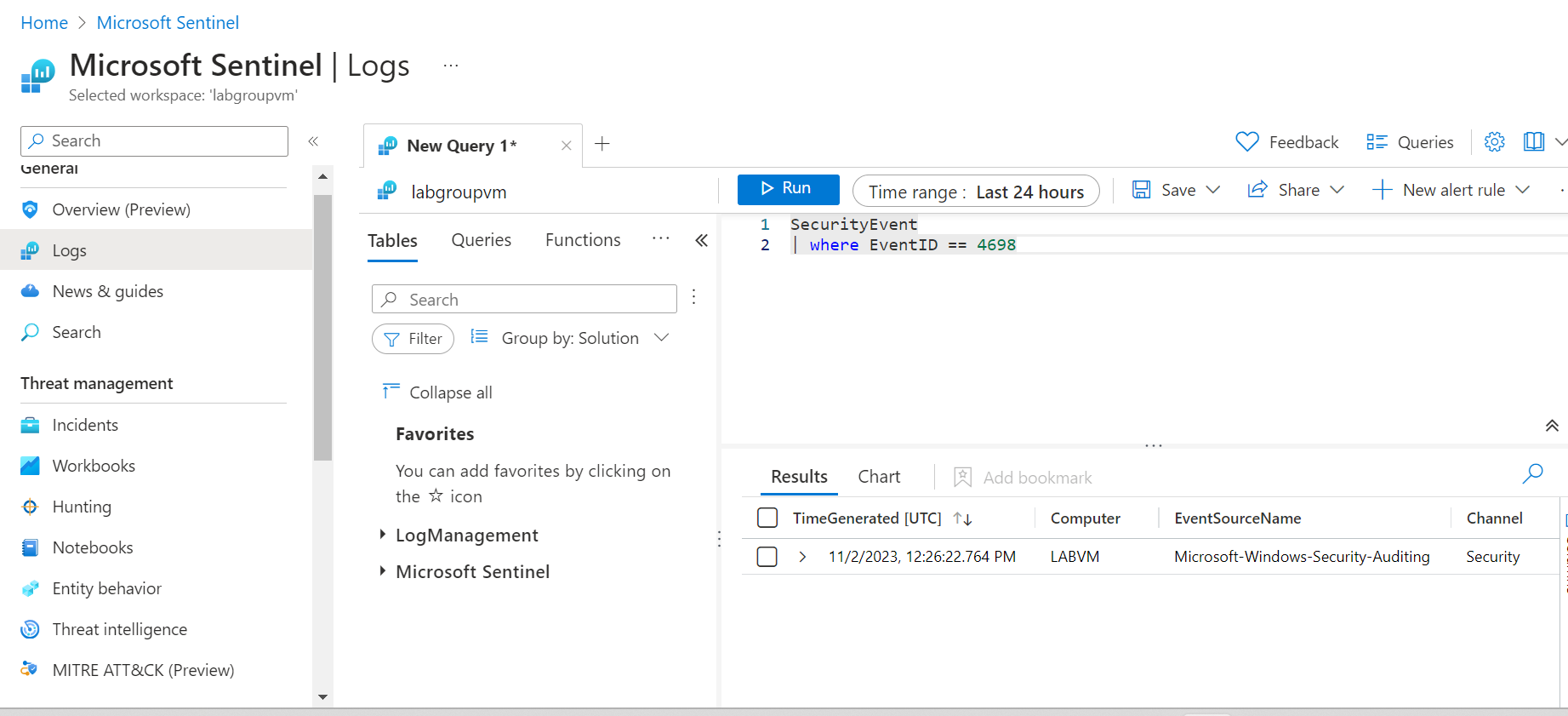
After this do not worry about the conditions and settings do not change any of the settings and click “OK”.

This will create your scheduled task you can now go to event viewer and search for that event id 4698 in the security logs. You can see here there are several instances of this event.

****

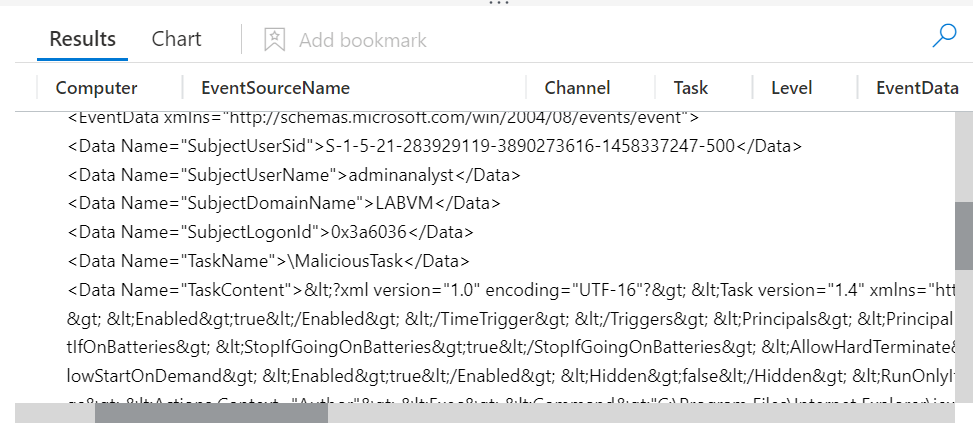
We go to sentinel and type a query:  
**SecurityEvent  
| where EventID == 4698**

And we can see the result



Expand the log for the scheduled task you created in the previous step and look at the Event Data category.

You should see something like this:



There is a lot of useful data in here such as the name of the scheduled task, the Task Name field, the ClientProcessID, the username of the account that created the scheduled task amongst other info.

However, if you use the **project**command to display these data fields as columns when you run the query, we need to add this to our logic.

**SecurityEvent**

**| where EventID == 4698**

**| parse EventData with \* '<Data Name="SubjectUserName">' User '</Data>' \***

**| parse EventData with \* '<Data Name="TaskName">' NameofSceuduledTask '</Data>' \***

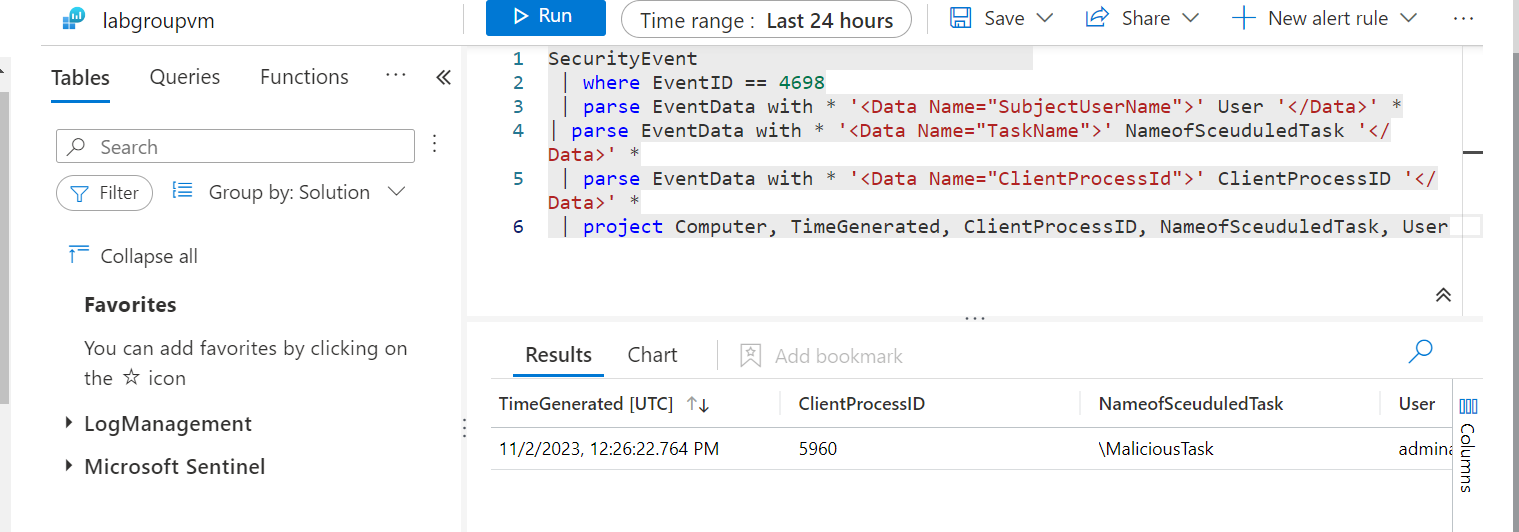
**| parse EventData with \* '<Data Name="ClientProcessId">' ClientProcessID '</Data>' \***

**| project Computer, TimeGenerated, ClientProcessID, NameofSceuduledTask, User**

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

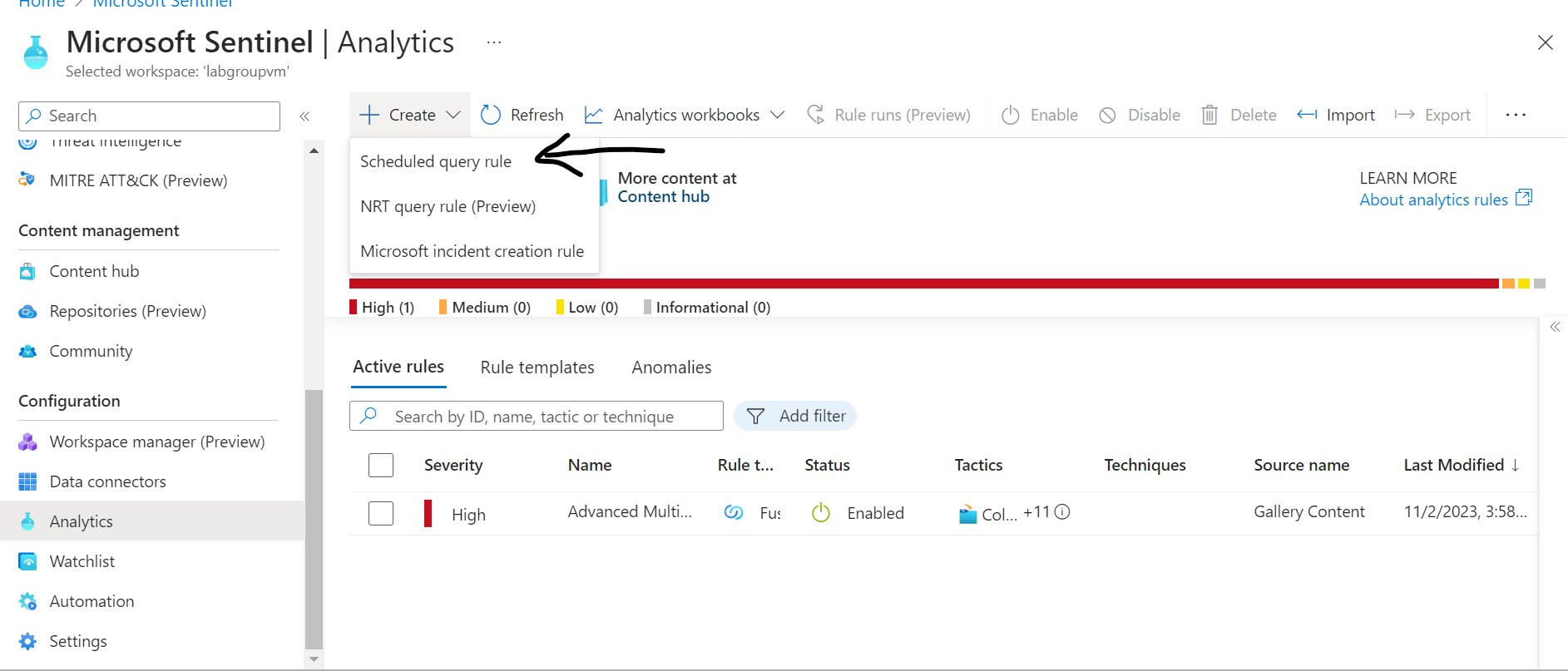
This extracted the SubjectUserName , TaskName, ClientProcessID (Computer automatically displays) .

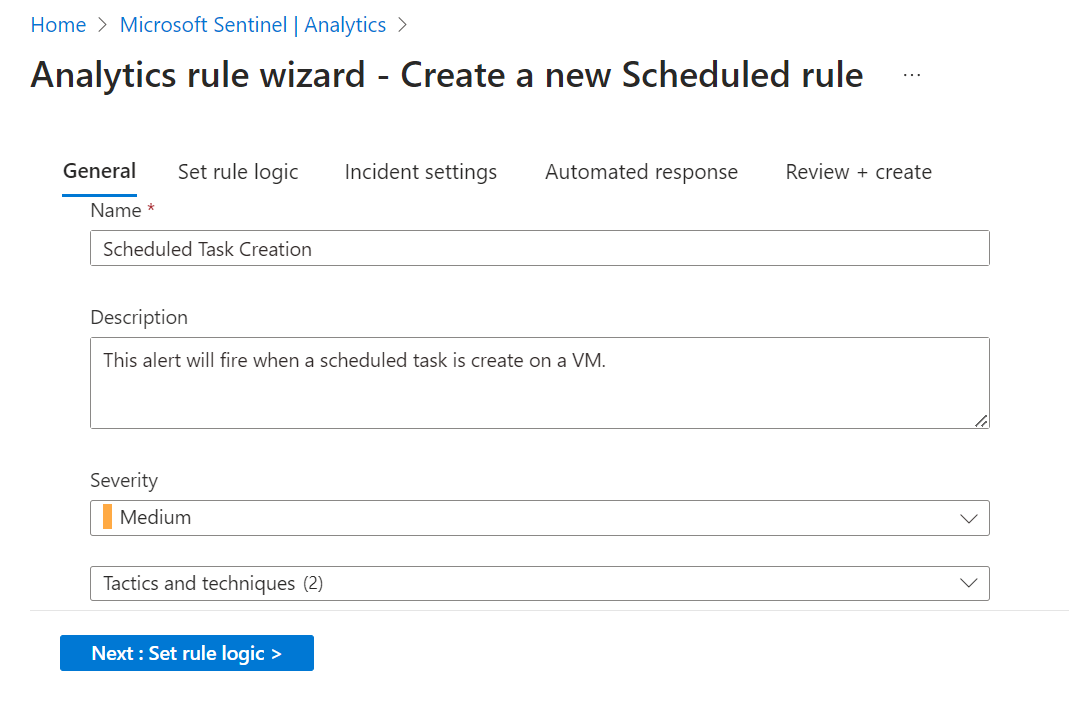
The above logic allows me to assign those to new categories such as User, NameofScheduledTask, and ClientProcessIDrespectively. When we project our new fields, the output is the following:



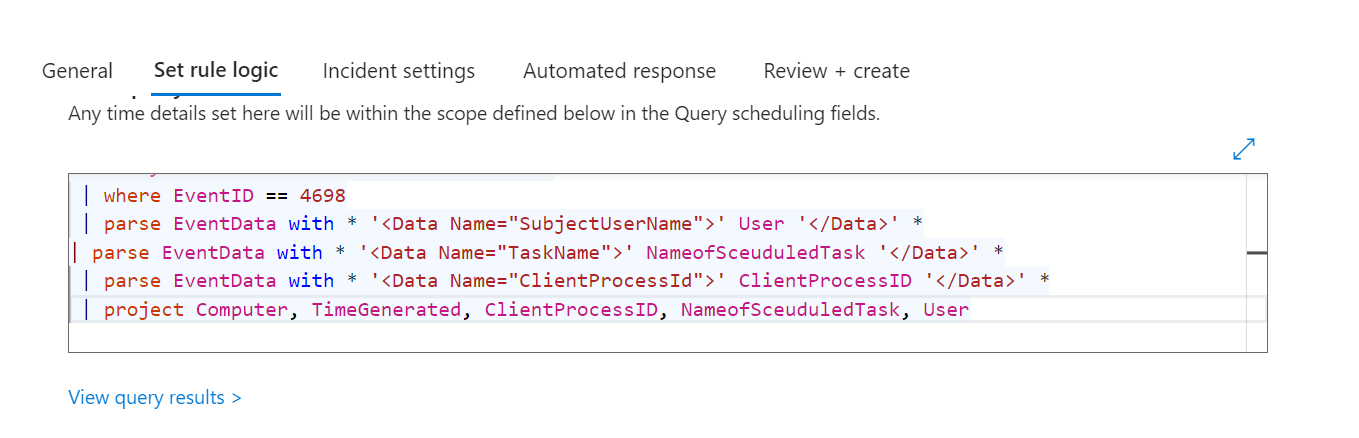
As you can see we’re able to generate Event Data and place it into its own category for readability.

Copy and Paste the above query into the editor under the “Set Rule Logic” tab





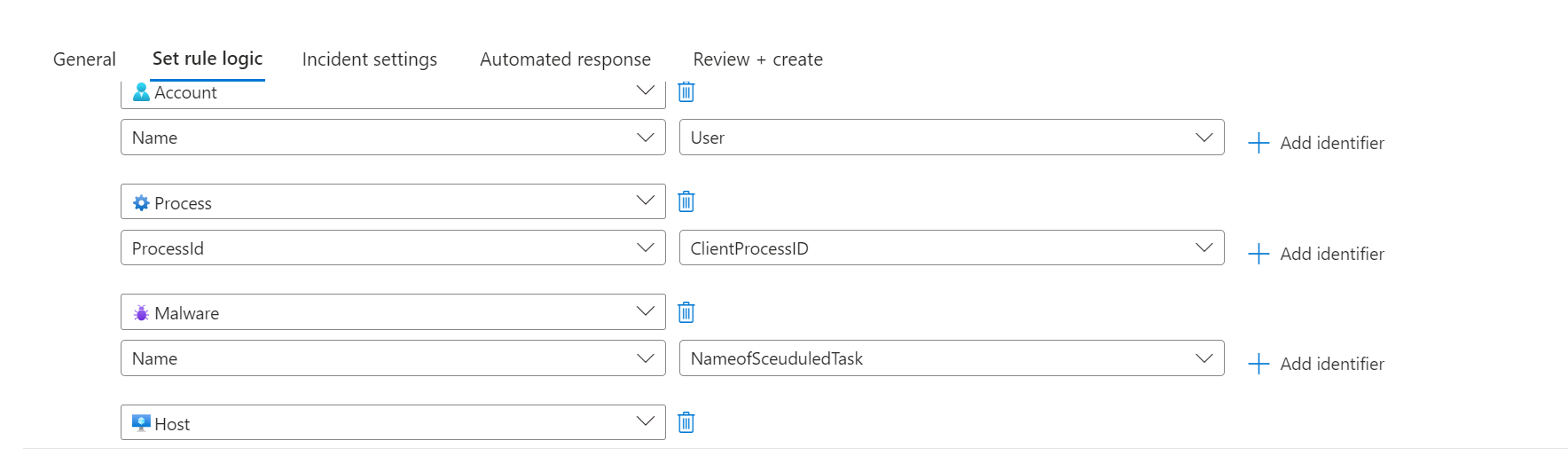
Copy and Paste the above query into the editor under the “Set Rule Logic” tab

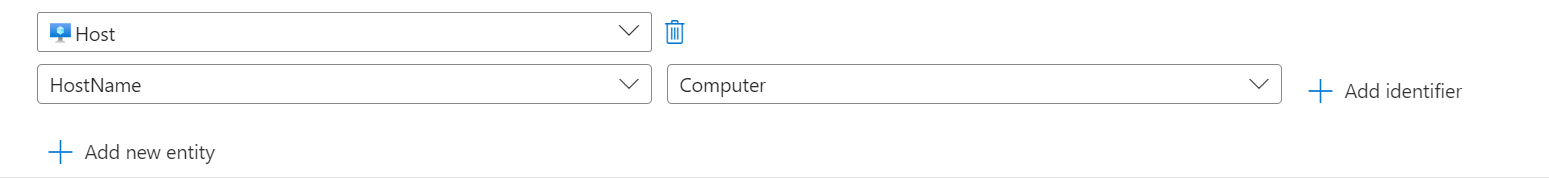


The Alert Enrichment section is below this. **Enriching**an alert simply is the process of adding context to the alert to make it easier for the analyst to investigate.

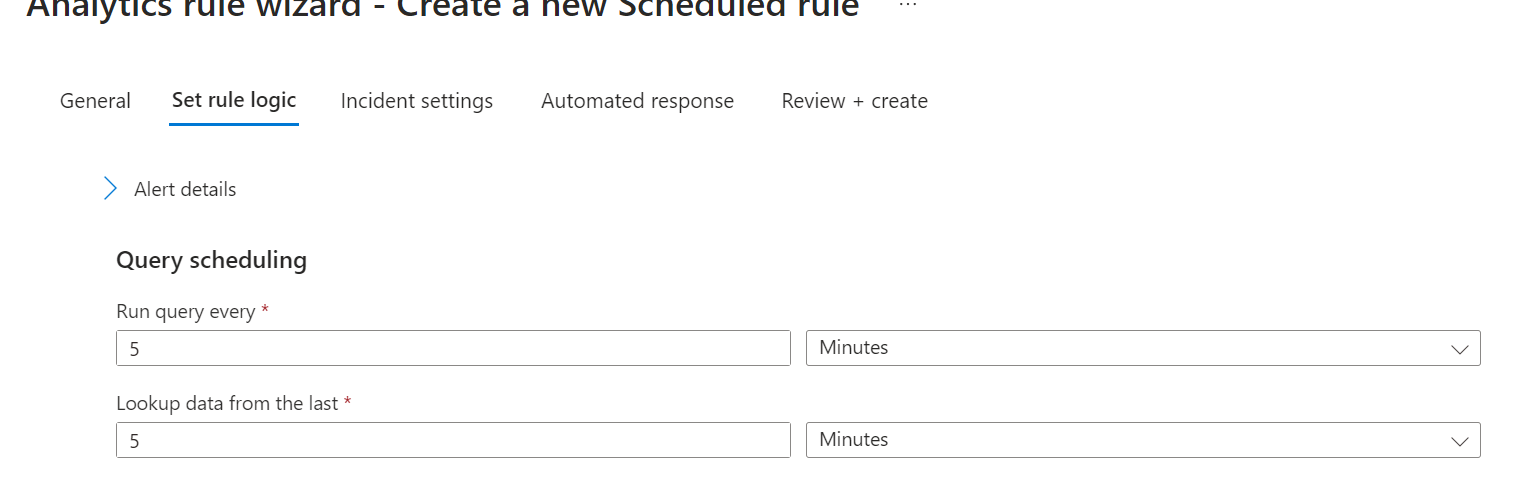
As opposed to having to query this data as we did in the analytic rule Alert Enrichment allows us to put the necessary data into the alert details as Entities so the analyst can begin investigating those specific components.

Use the following settings for entity mapping.

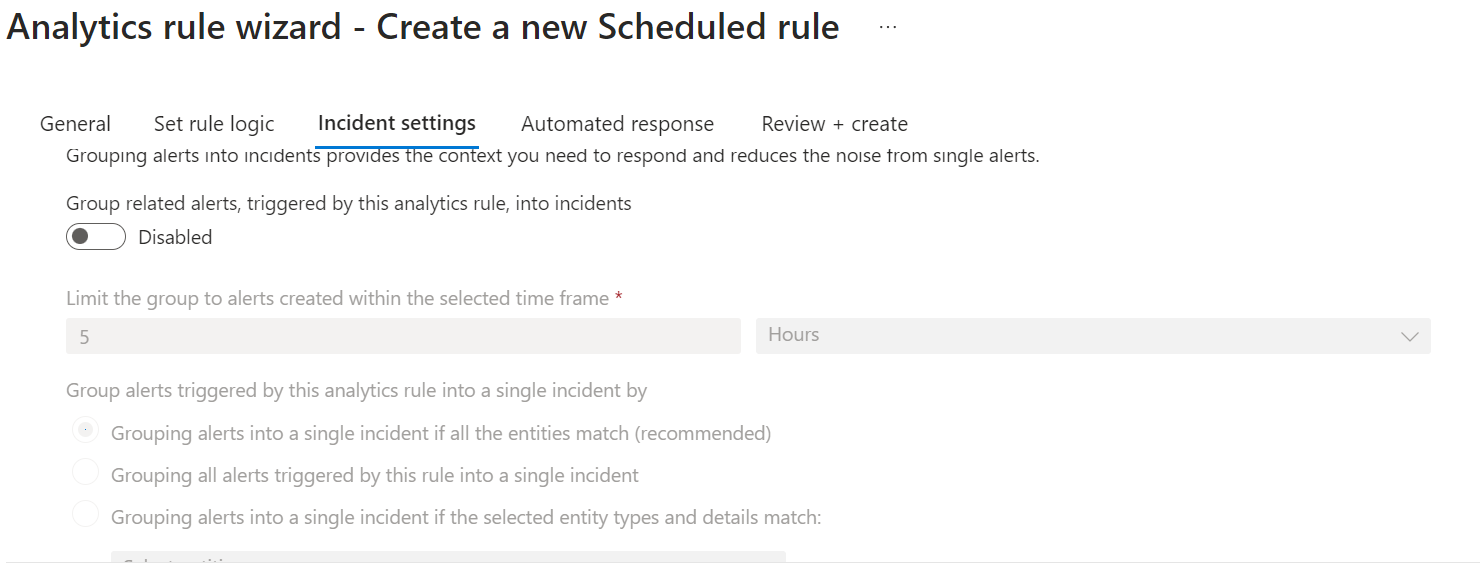


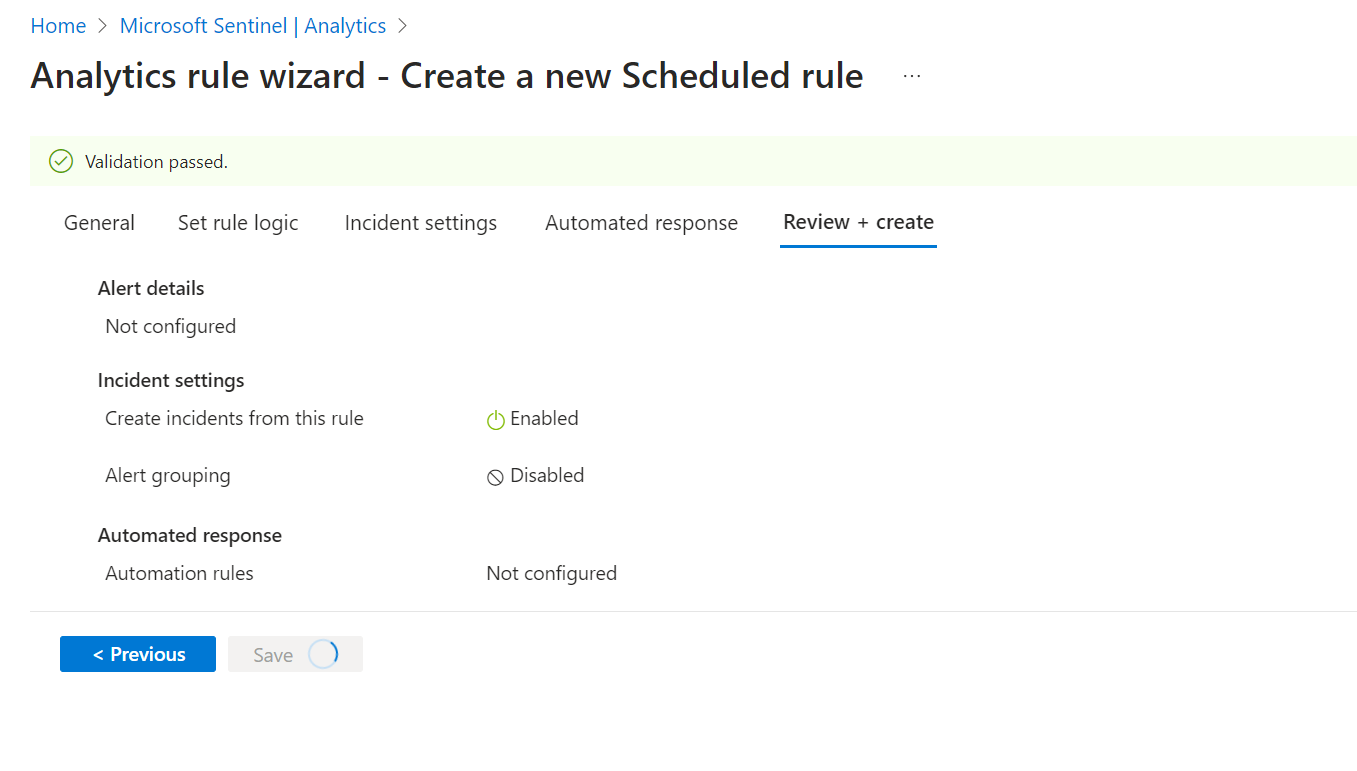


The only other setting that need to be changed for the purposes of this lab is how often the query is run and when to look up the data. Change defaults to the following:



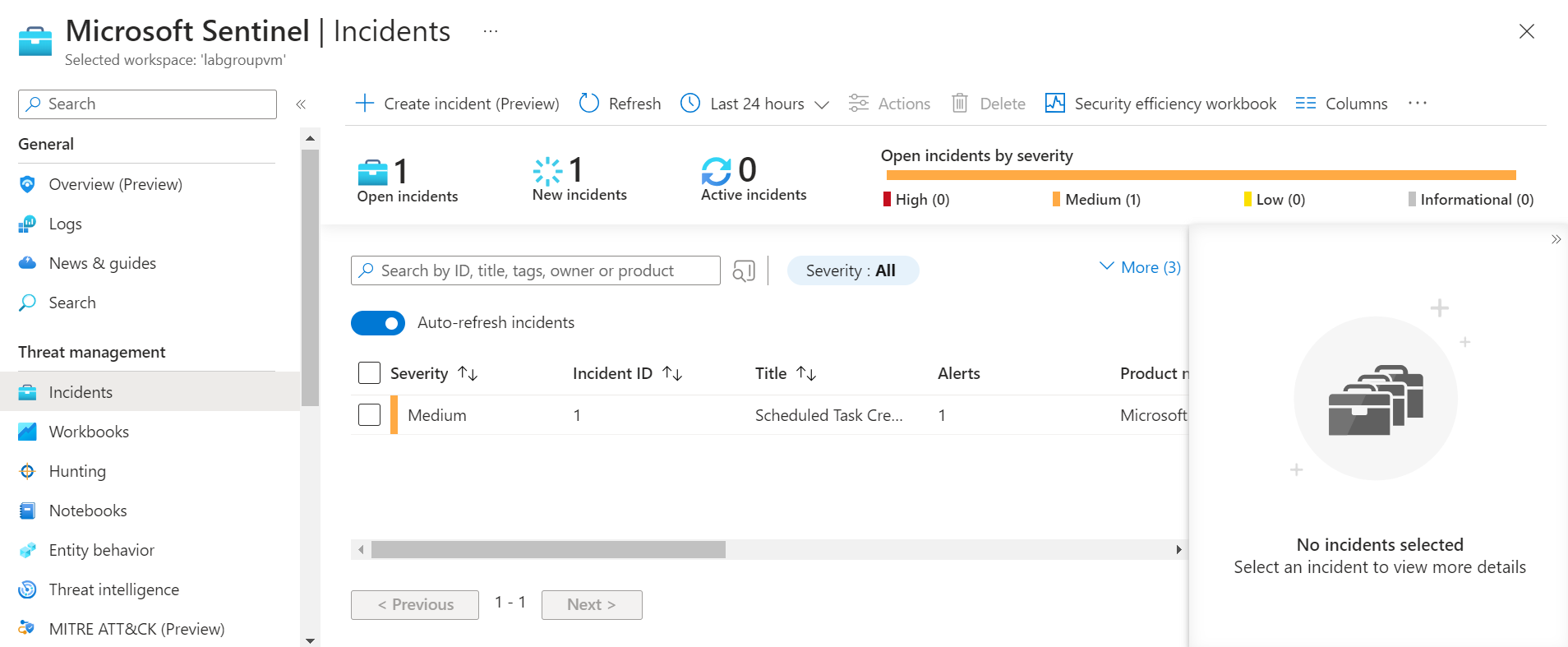
Incident Settings and Automated Response are not necessary to alter for this lab so you can go ahead to “review and create” to make the Analytic Rule.

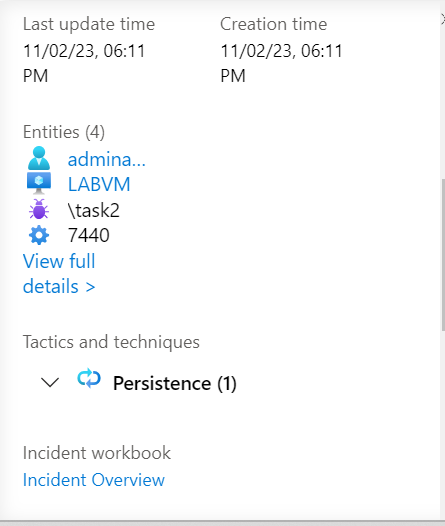




Once you create your analytic rule the final step is to create another scheduled task in your Windows VM and wait for the alert to trigger in Sentinel

\*Note this might take up to 10 minutes. Refresh the Incidents page periodically until the Alert triggers. When the alert fires this is what you should see.



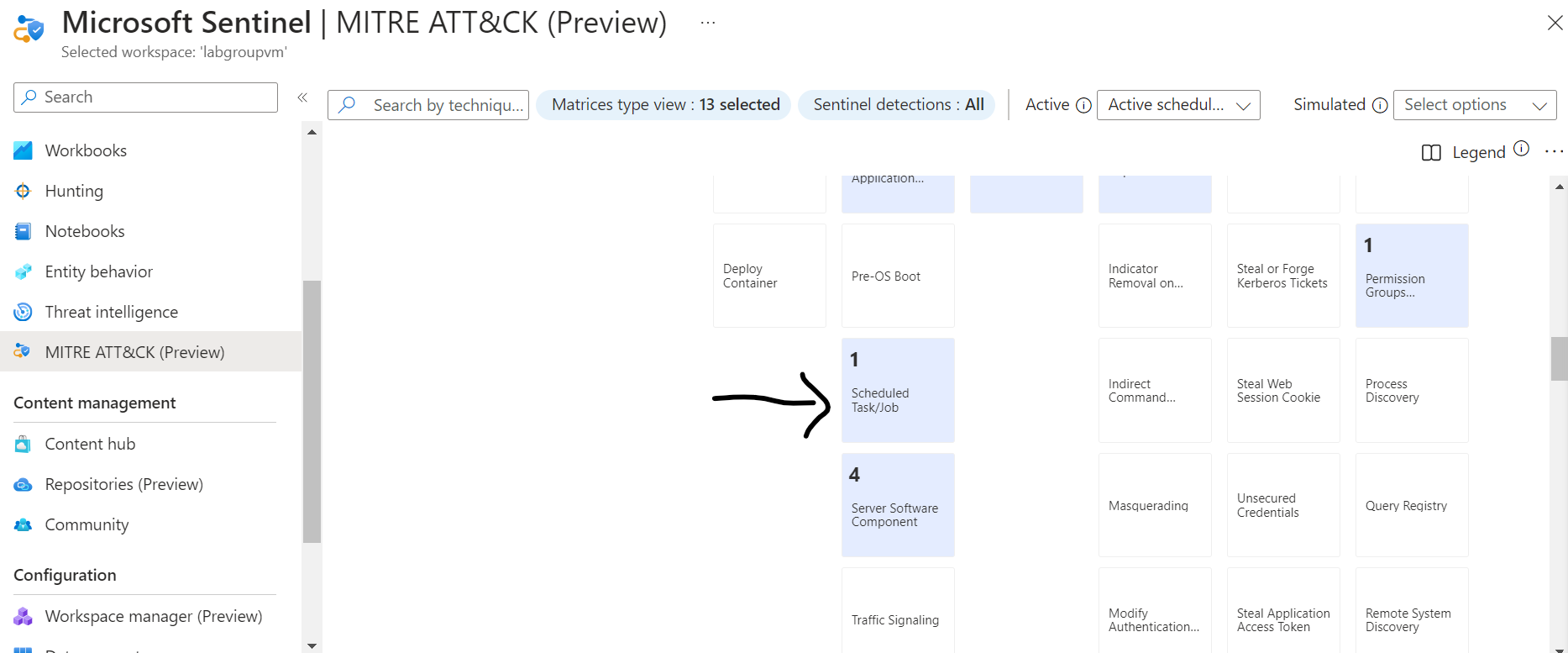


On the right pane we see all the necessary information we would need to begin investigating the alert such as the host machine, user account, process ID of the task, and the name of the scheduled task.

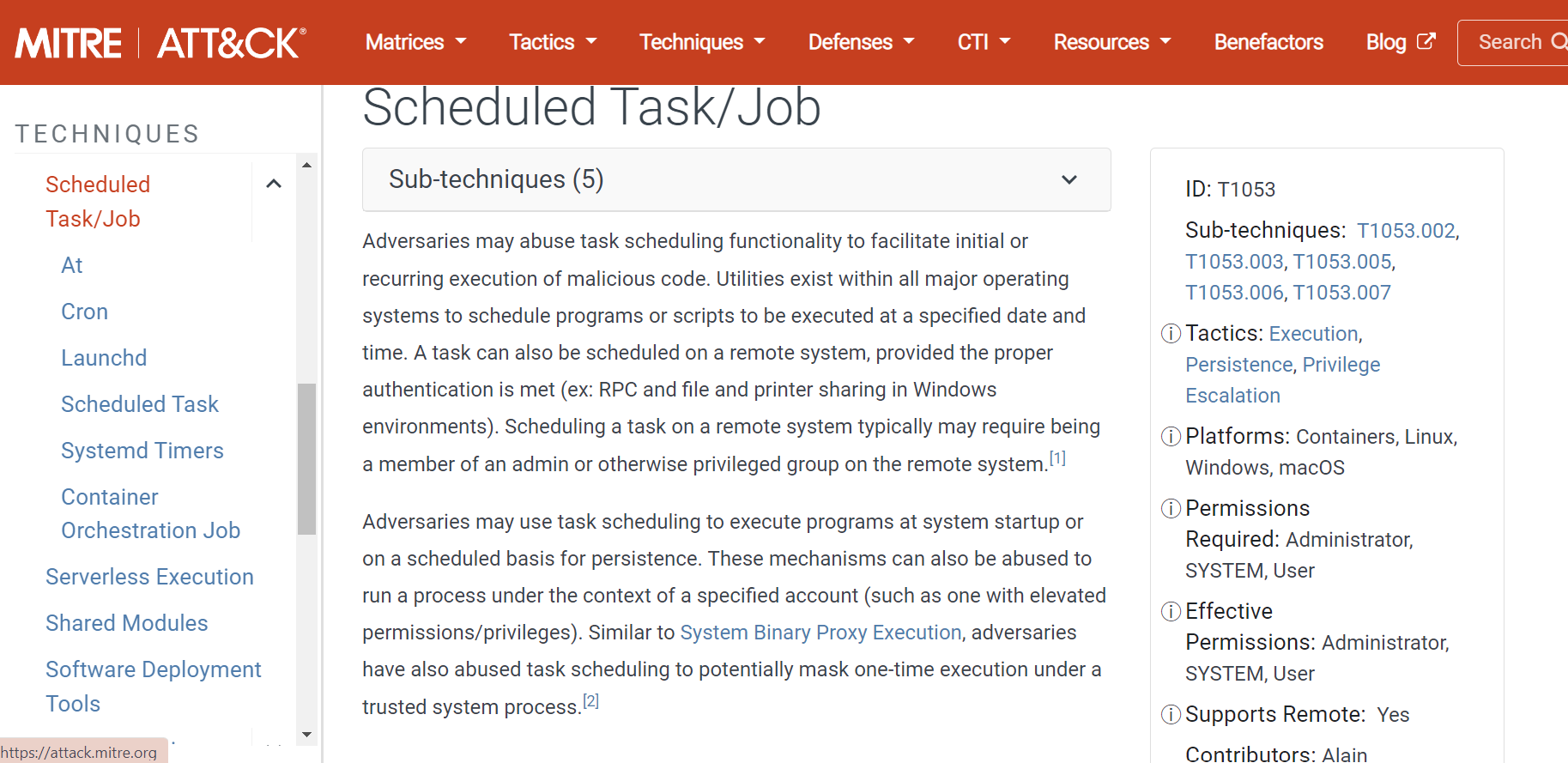
While in this case, the scheduled task is non-malicious, in the event it was, from here an analyst would investigate the entities such as the user account, the scheduled task, host, etc. with other tools such as an EDR solution and other security tools to decide if this is a false positive or true positive.

## [Part 7: MITRE ATT&CK](#_Part_6:_MITRE)

The observed MITRE ATT&CK tactic used in this lab is [TA0003 **Persistence**](https://attack.mitre.org/tactics/TA0003/)which essentially allows a malicious actor to maintain a foothold in an environment.

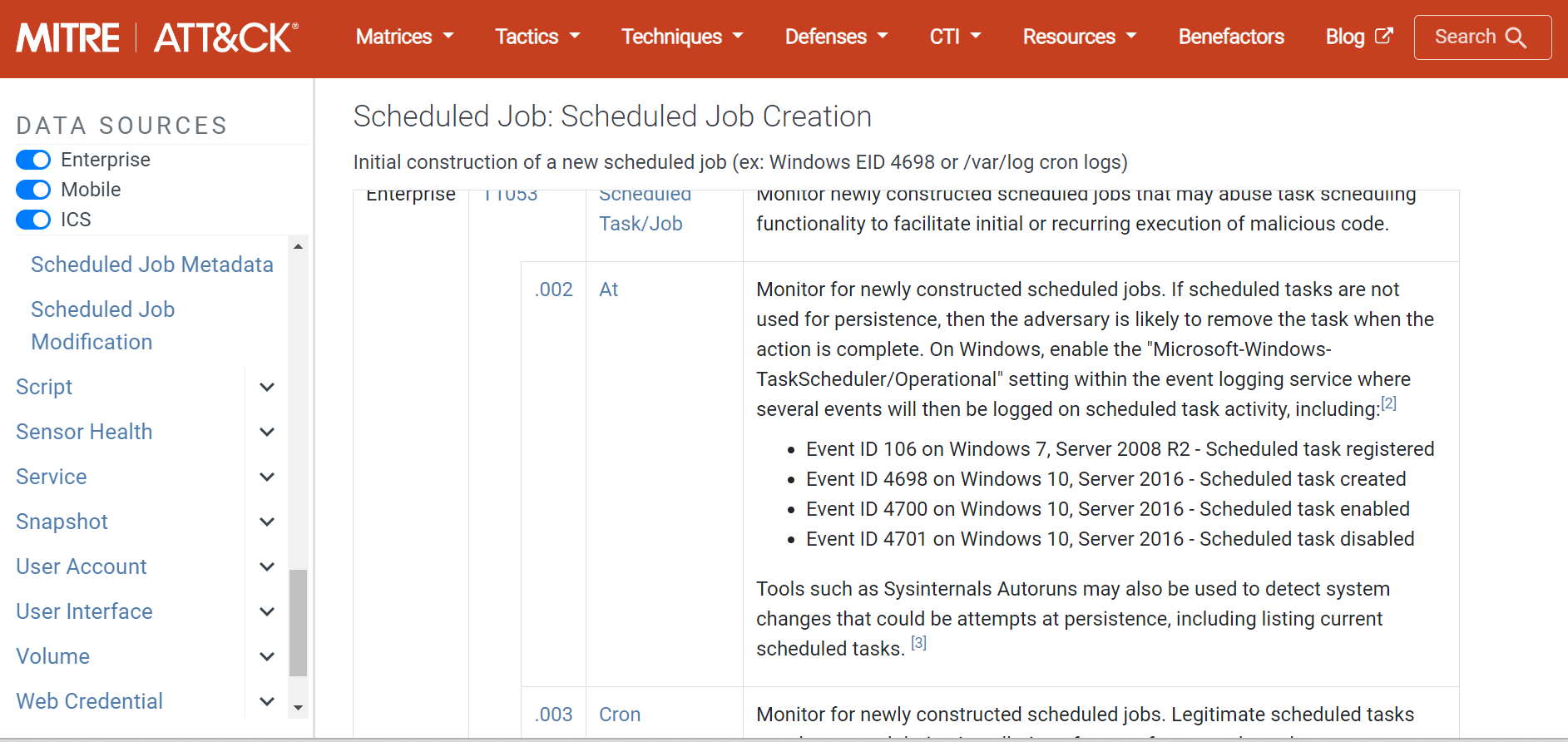
****

Even further, we can dig further into the specific sub technique of [T1053.005](https://attack.mitre.org/techniques/T1053/005/)

****

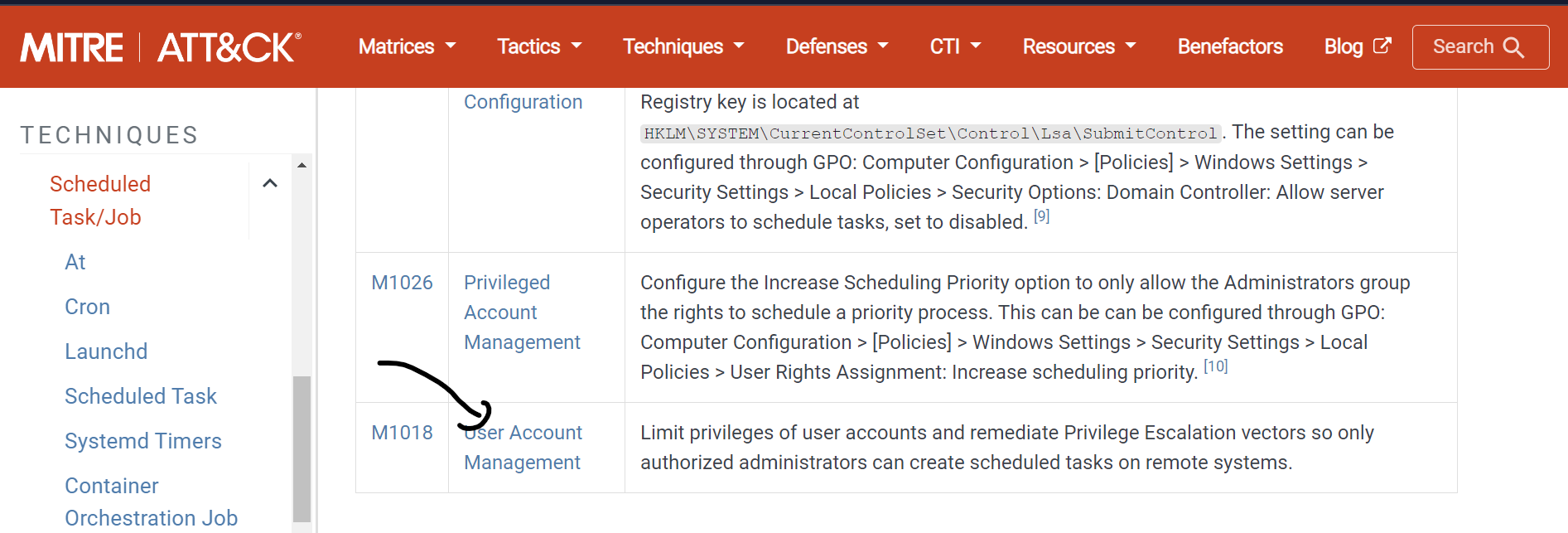
### Detection

As observed. monitoring and logging of specific windows event id was used to detect this activity. However, MITRE also has more recommendations for detection.

****

### Mitigation

[MITRE ID M1019](https://attack.mitre.org/mitigations/M1018/) – **User Account Management** suggests that user account privileges should be limited to only authorize admins to create scheduled tasks on remote systems.

****

**This concludes the lab. Thank you.**