Microsoft ADC Cybersecurity Skilling Program

Week 11 Lab Assignment

Student Name: Vincent Onchieku Collins

Student ID: ADC-CSS02-25052

- **Lab 08: Create a Log Analytics Workspace, Azure Storage Account, and Data Collection Rule (DCR)**
    - Exercise 1: Deploy an Azure virtual machine
    - Exercise 2: Create a Log Analytics workspace
    - Exercise 3: Create an Azure storage account
    - Exercise 4: Create a data colllection rule

- **Lab 09: Configuring Microsoft Defender for Cloud Enhanced Security Features for Servers**
    - Configure Microsoft Defender for Cloud Enhanced Security Features for Servers

- **Lab 10: Enable just-in-time access on VMs**
    - Exercise 1: Enable JIT on your VMs from the Azure portal.
    - Exercise 2: Request access to a VM that has JIT enabled from the Azure portal.

- **Lab 11: Microsoft Sentinel**
    - Task 1: On-board Microsoft Sentinel
    - Task 2: Connect Azure Activity to Sentinel
    - Task 3: Create a rule that uses the Azure Activity data connector.
    - Task 4: Create a playbook
    - Task 5: Create a custom alert and configure the playbook as an automated response.
    - Task 6: Invoke an incident and review the associated actions.

Tasks:

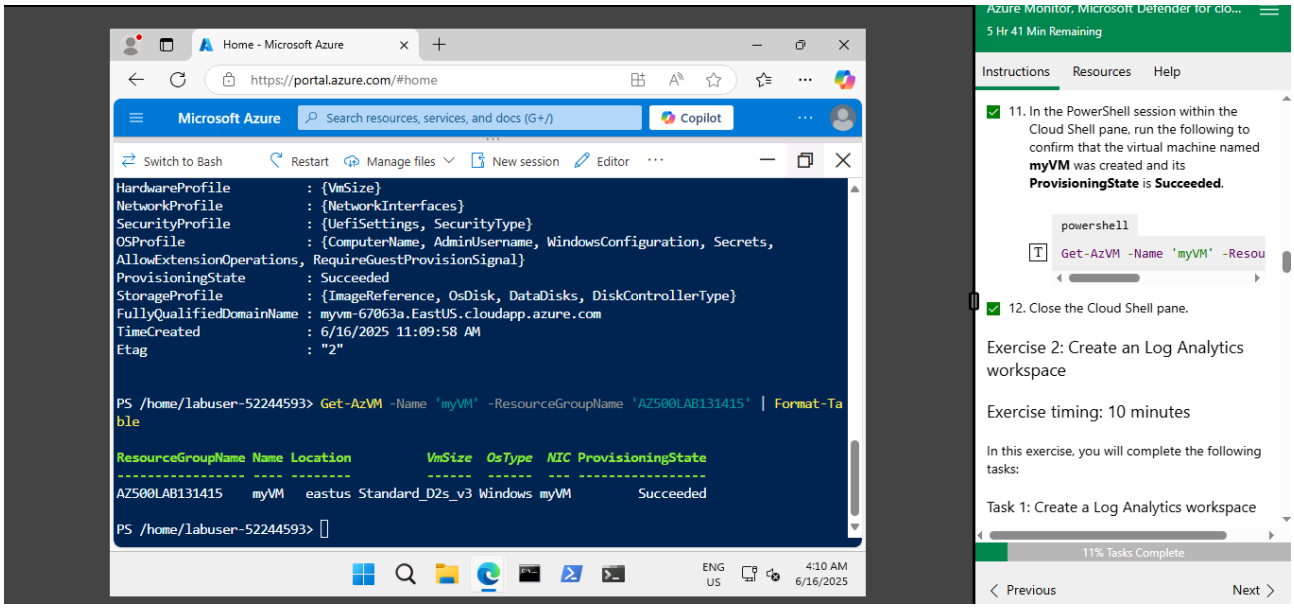
Lab 08: Create a Log Analytics Workspace, Azure Storage Account, and Data Collection Rule (DCR)

Introduction

In this lab, I assumed the role of an Azure Security Engineer tasked with implementing an efficient monitoring and data collection system across virtual machines processing financial transactions. The lab was centered on configuring Azure Monitor Agent (AMA) and Data Collection Rules (DCRs) to gather security events and performance data. To achieve this, I performed four key exercises: deploying a virtual machine, creating a Log Analytics workspace, setting up an Azure storage account, and configuring a data collection rule. These components collectively enable centralized monitoring and enhanced visibility into system behavior and potential security threats. The tasks you need to complete will include:

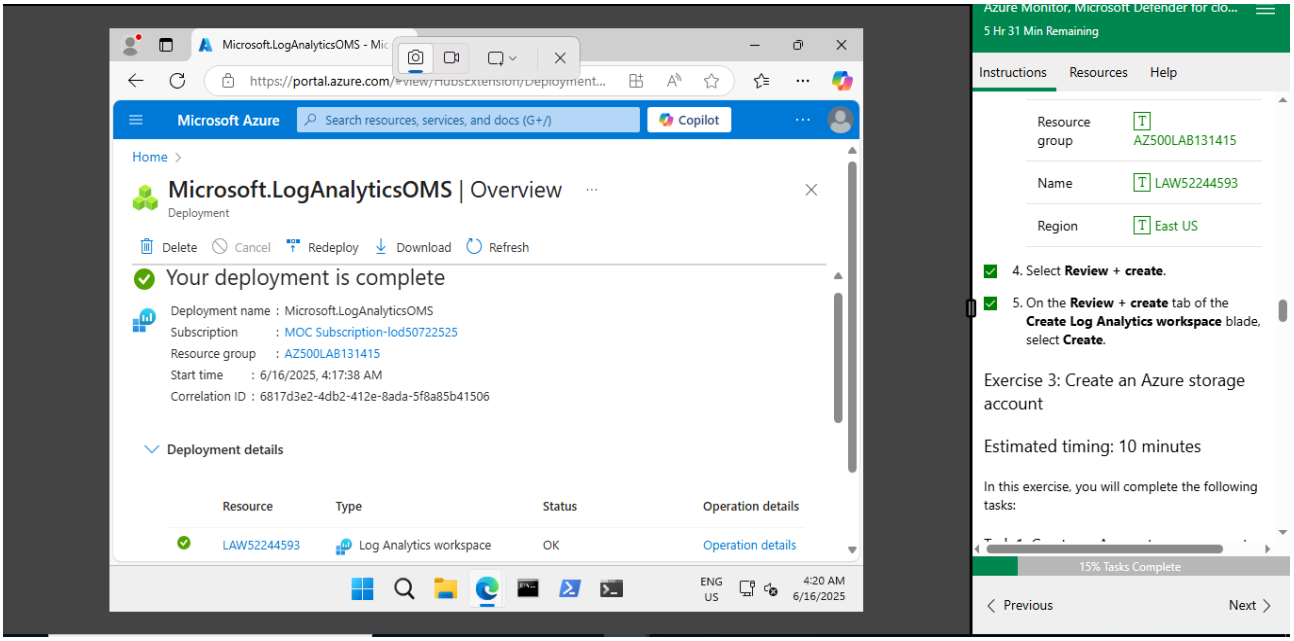
Exercise 1: Deploy an Azure Virtual Machine

The first exercise involved deploying an Azure virtual machine, which served as the foundational resource for the lab. Using PowerShell within Azure Cloud Shell, I created a new resource group and then provisioned a virtual machine named myVM in the East US region. During the deployment, I enabled Encryption at Host for added security, and opened necessary ports (80 and 3389) to support HTTP access and remote desktop connections. After setting up administrative credentials, I verified the provisioning state of the VM to ensure it was successfully created. This virtual machine would later be linked with monitoring tools to simulate data collection and performance tracking.



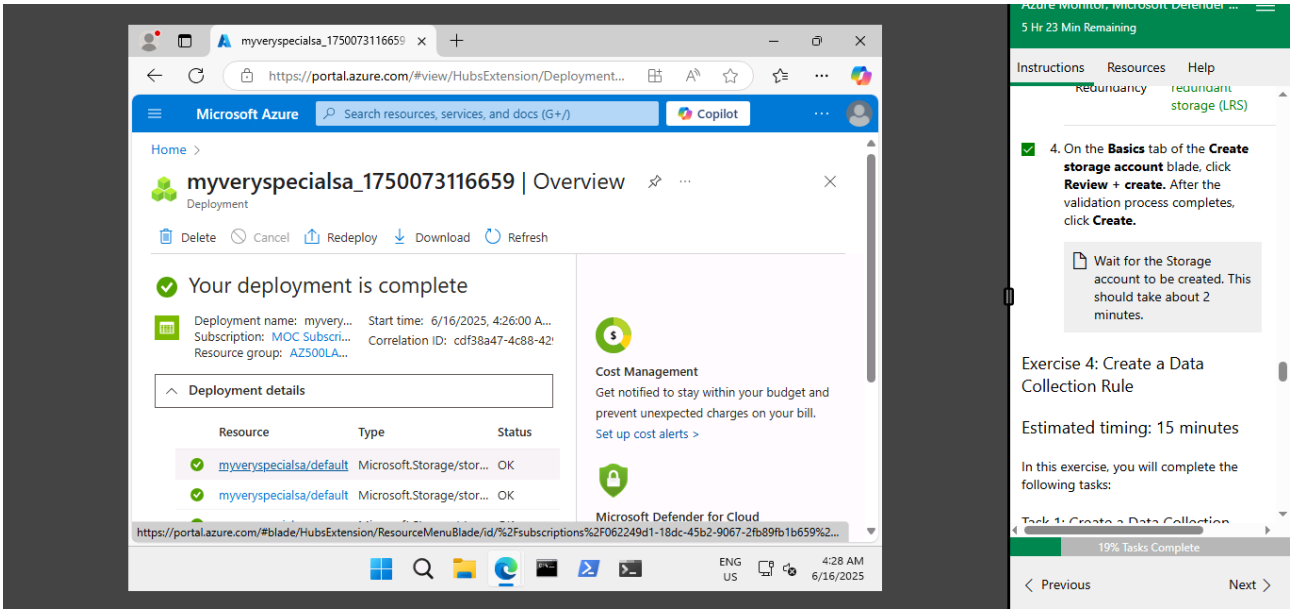
Exercise 2: Create a Log Analytics Workspace

In the second exercise, I created a Log Analytics workspace, which acts as the central repository for logs collected from Azure resources. Through the Azure portal, I created a workspace named LAW52243405 under the same resource group (AZ500LAB131415) used for the virtual machine. I ensured the workspace was deployed in the East US region to maintain consistency across resources. This workspace would later be integrated with the Data Collection Rule to receive telemetry and performance data from the virtual machine, enabling real-time log analytics and insights.



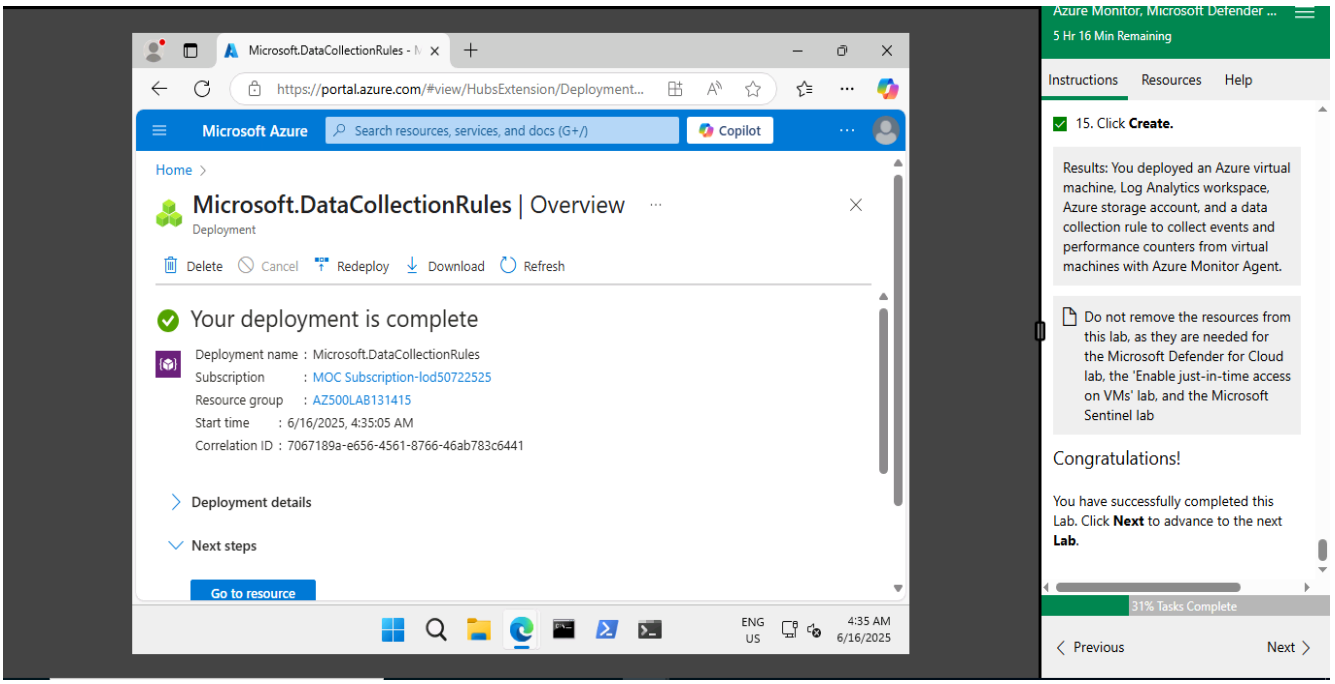
Exercise 3: Create an Azure Storage Account

The third exercise focused on creating a storage account for storing large volumes of collected data. I accessed the Storage accounts blade and initiated the creation of a new storage account under the same resource group and region as previous resources. The account was configured as a general-purpose v2 account with locally redundant storage (LRS), using either Azure Blob Storage or Azure Data Lake Gen2 as the primary service. This storage account is essential for storing diagnostic logs, backups, or exporting log data for extended retention and offline analysis.



Exercise 4: Create a Data Collection Rule

The final exercise involved setting up a Data Collection Rule (DCR) to define how telemetry is collected and where it is delivered. I created a rule named DCR1 through the Azure Monitor interface, targeting the previously deployed virtual machine. I specified performance counters such as CPU, memory, disk, and network with a sample rate of 60 seconds each. The rule was then linked to the Log Analytics workspace created in Exercise 2, ensuring that all collected data from the VM would be sent and centralized there. This rule is key to enabling granular monitoring, alerting, and future analysis of system health and behavior.



Conclusion

Completing this lab provided hands-on experience with building a comprehensive monitoring infrastructure using Azure-native tools. By deploying a virtual machine, creating a Log Analytics workspace, setting up a storage account, and defining a data collection rule, I established a secure and efficient pipeline for collecting and analyzing performance and security data. These steps are essential for organizations needing to monitor critical workloads and maintain high standards of compliance and operational insight. The configured resources will serve as a foundation for subsequent labs focused on security enhancement using Microsoft Defender for Cloud and Microsoft Sentinel.

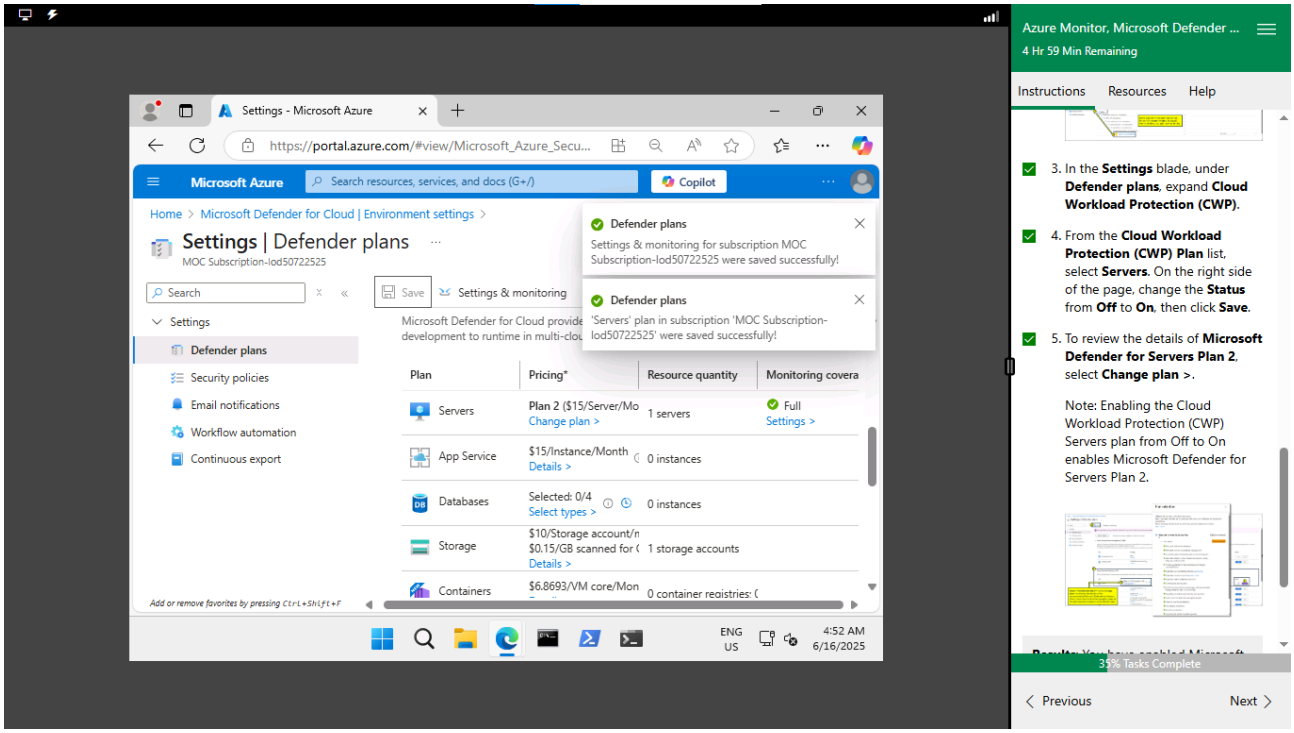
Lab 09: Configuring Microsoft Defender for Cloud Enhanced Security Features for Servers

Introduction

In today's evolving cybersecurity landscape, proactive defense strategies are essential for safeguarding cloud infrastructure. In this lab, I acted as an Azure Security Engineer for a global e-commerce organization with a critical dependency on Azure virtual machines and hybrid servers. My task was to strengthen the security posture of these resources by enabling advanced threat protection using Microsoft Defender for Servers within Microsoft Defender for Cloud. This configuration helps detect and respond to threats, reduce vulnerabilities, and ensure continuous compliance across the server environment.

Exercise: Configure Microsoft Defender for Cloud Enhanced Security Features for Servers

During the exercise, I accessed Microsoft Defender for Cloud via the Azure portal and navigated to the Environment settings of my Azure subscription. Under the Cloud Workload Protection (CWP) section, I located the plan settings for Servers and switched the protection status from Off to On, thereby activating Microsoft Defender for Servers Plan 2. This plan includes a robust suite of security features such as threat detection, vulnerability assessments, file integrity monitoring, and integration with Microsoft Sentinel. I also reviewed the details of Plan 2 to understand the full capabilities offered, including just-in-time access control and endpoint detection and response (EDR). This configuration provides continuous monitoring and advanced defenses for both cloud-based and on-premises workloads.



Conclusion

Completing this lab enabled me to implement enterprise-grade security enhancements for server infrastructure in Azure. By activating Microsoft Defender for Servers Plan 2, I established a solid foundation for identifying threats, reducing risks, and ensuring regulatory compliance. This hands-on experience reinforced the importance of leveraging Microsoft Defender for Cloud as a centralized tool for protecting critical workloads in dynamic, hybrid environments.

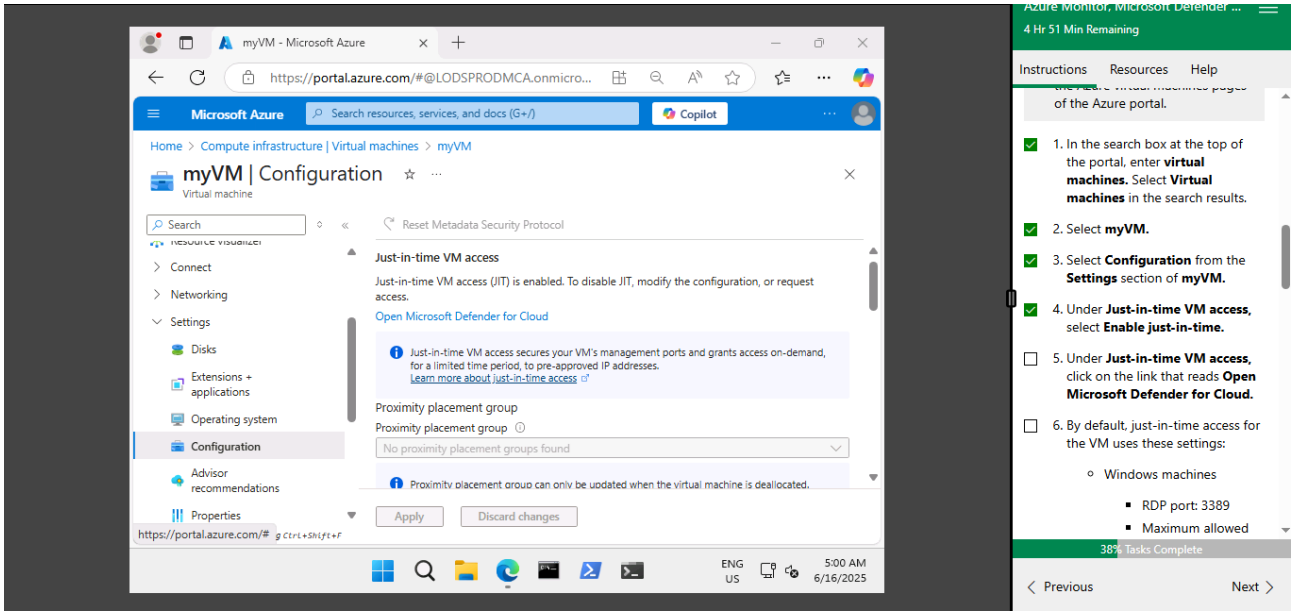
Lab 10: Enable just-in-time access on VMs

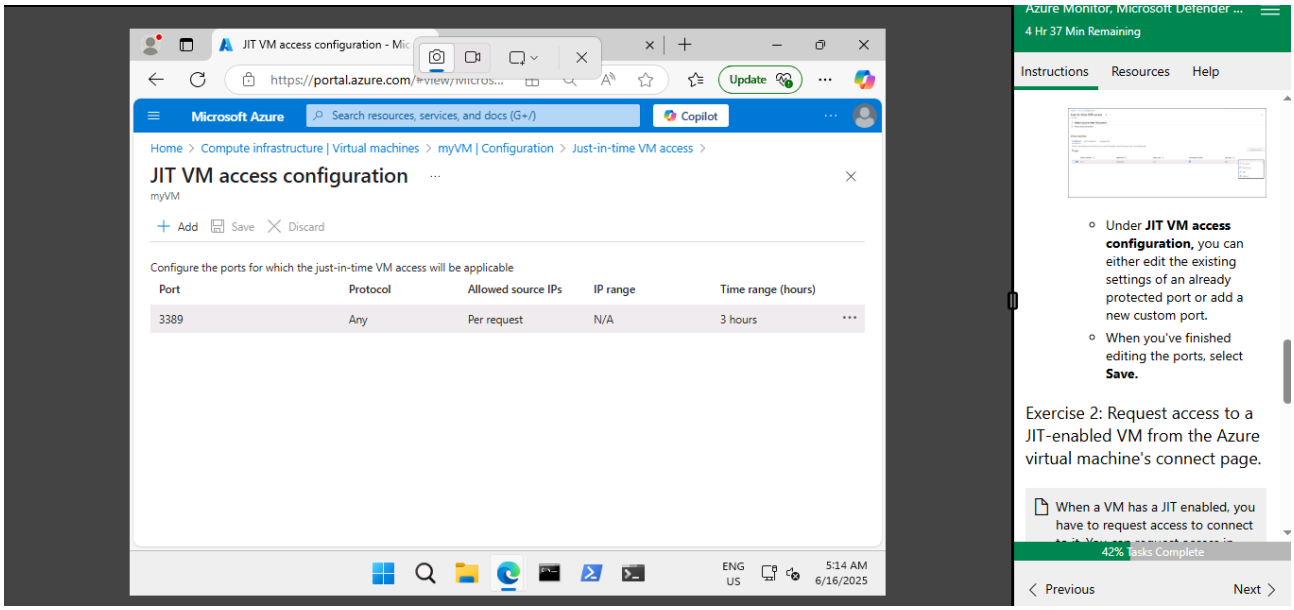
Introduction

In today's cybersecurity environment, minimizing exposure to attack surfaces is a crucial defense strategy. In this lab, acting as an Azure Security Engineer for a financial services company, I focused on reducing the risk of unauthorized access and brute-force attacks on Azure virtual machines (VMs). The objective was to implement Just-in-Time (JIT) VM access, a security feature in Microsoft Defender for Cloud that restricts access to VMs by only allowing time-limited, approved access. This lab consisted of two key exercises: enabling JIT access and requesting access to a protected VM.

Exercise 1: Enable JIT on Your VMs from the Azure Portal

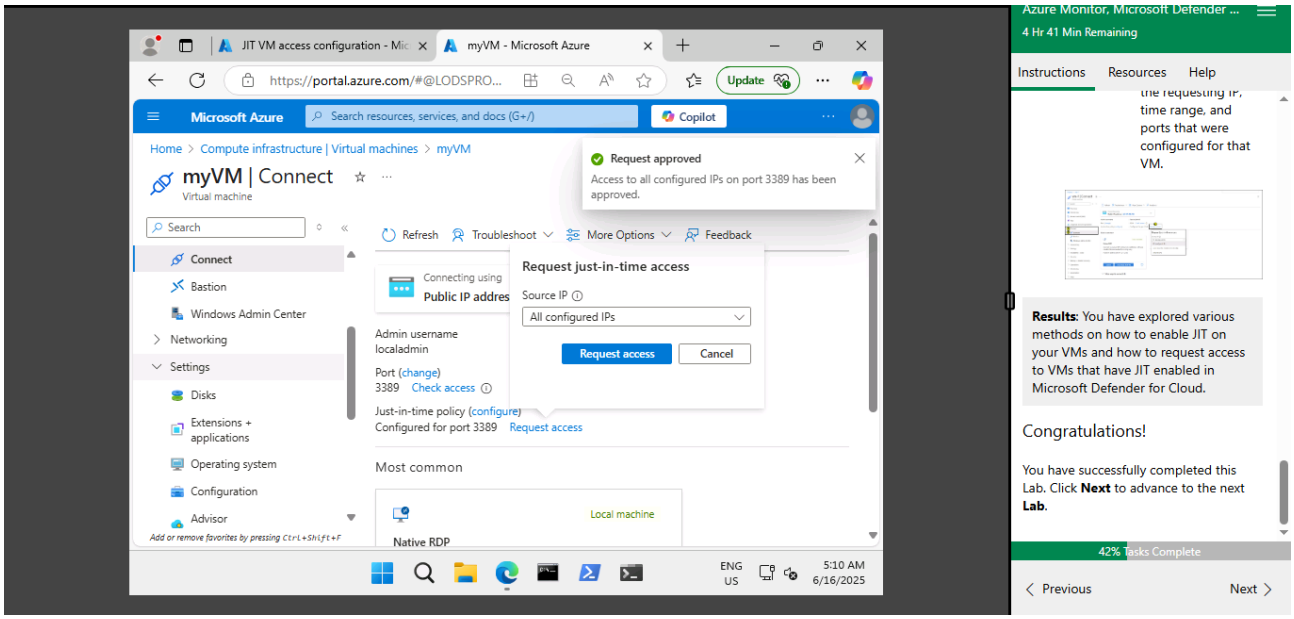
In this exercise, I enabled Just-in-Time access on a virtual machine named myVM using the Azure portal. I navigated to the VM's configuration settings and accessed the Just-in-Time access option through Microsoft Defender for Cloud. The default settings allowed RDP access (port 3389) for up to three hours from any IP address. I reviewed and confirmed these default parameters and saved the configuration. This setup ensures that remote access to the VM is only granted when explicitly requested, thereby reducing the risk of constant exposure to potential cyber threats.





Exercise 2: Request access to a VM that has JIT enabled from the Azure portal.

In the second exercise, I simulated a real-world scenario by requesting temporary access to the JIT-enabled VM. Through the VM's Connect page in the Azure portal, I verified that JIT was active and proceeded to request access. I specified the IP address, access duration, and port (RDP 3389) as required. Azure validated the request and granted timed access, demonstrating how JIT effectively controls and monitors who can access the VM and when. This request-based approach ensures that VMs remain protected when idle and accessible only when necessary.



Conclusion

Completing this lab enhanced my practical understanding of Just-in-Time VM access as a proactive security measure. By enabling and managing JIT through Microsoft Defender for Cloud, I successfully reduced potential attack vectors while maintaining controlled administrative access. This feature is essential for organizations handling sensitive data, as it strengthens the security posture of critical virtual machines against common threats and unauthorized intrusion attempts.

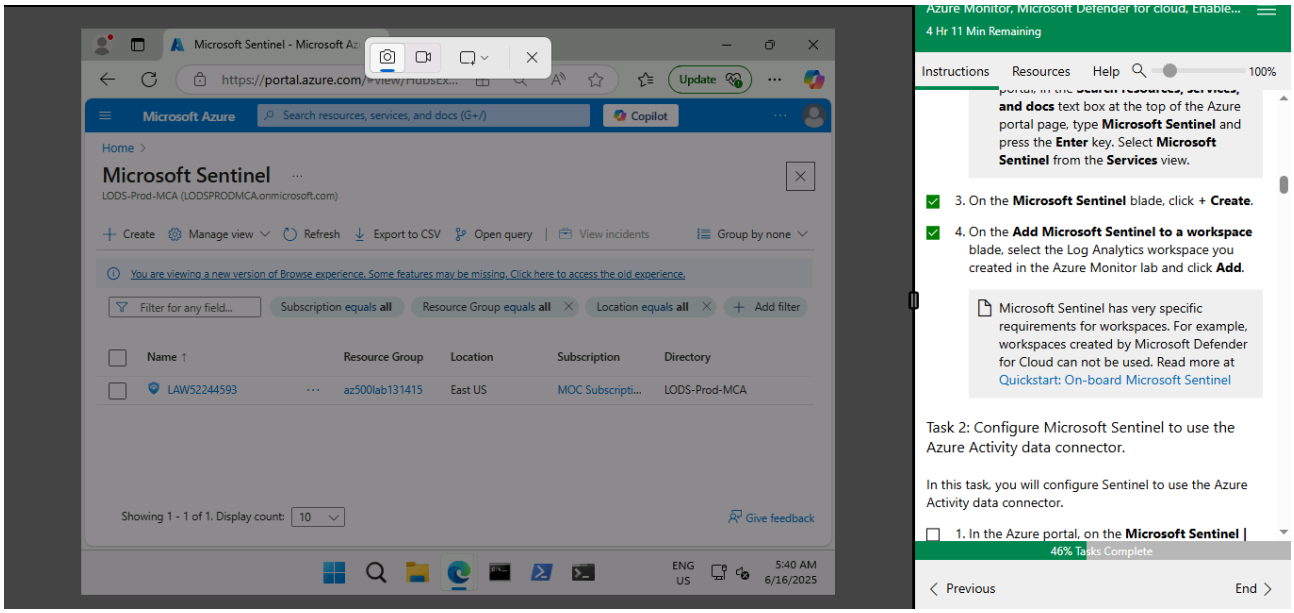
Lab 11: Microsoft Sentinel

Introduction:

In Lab 11, I explored the core functionalities of **Microsoft Sentinel**, Microsoft's cloud-native SIEM (Security Information and Event Management) solution, by performing a series of practical tasks designed to simulate real-world security monitoring and incident response. The lab began by onboarding Microsoft Sentinel to a Log Analytics workspace, establishing a centralized location for ingesting and analyzing security data. From there, I connected Azure Activity logs, created analytic rules to detect suspicious behavior, and developed a security playbook using Logic Apps. These foundational steps enabled the setup of automated detection and response workflows that enhance security visibility and operational efficiency within an Azure environment.

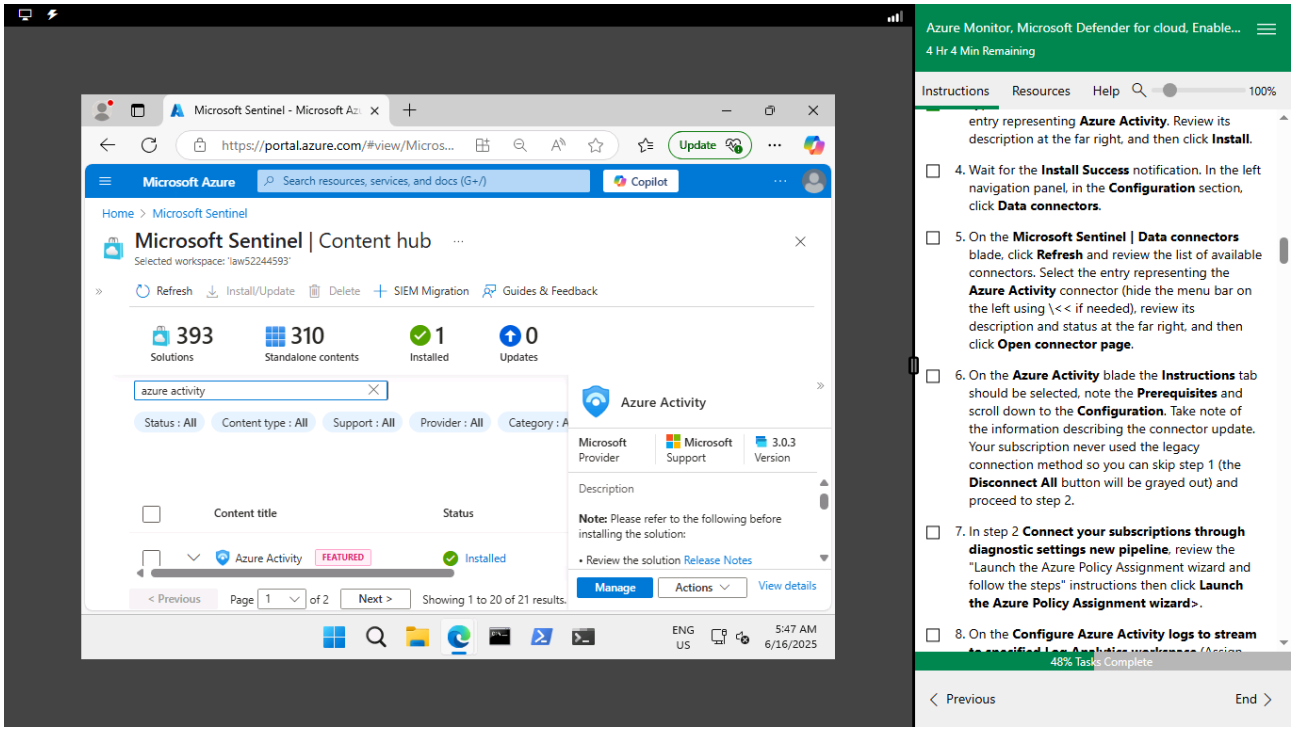
Task 1: On-board Microsoft Sentinel

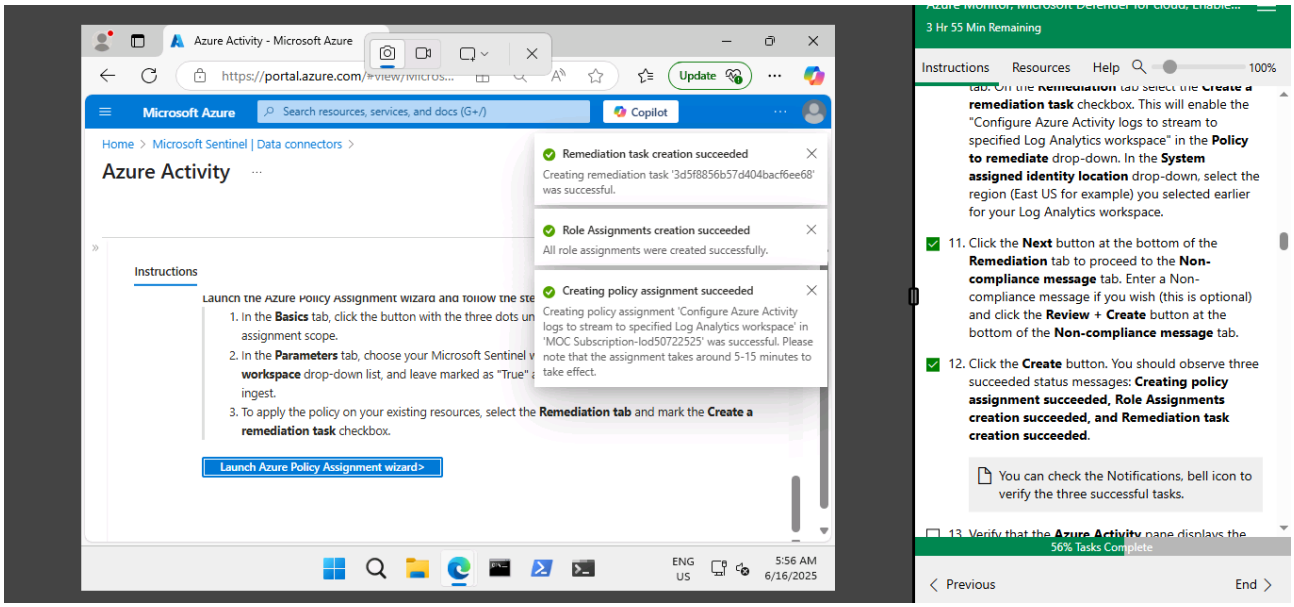
In the first task of the lab, I began by onboarding Microsoft Sentinel to my Azure environment. This involved selecting an appropriate Log Analytics workspace and adding Microsoft Sentinel to it. This setup is crucial, as it allows Sentinel to ingest data and begin monitoring for potential security threats across connected resources. By linking the Sentinel instance with a properly configured workspace, I laid the foundation for collecting and analyzing security-related logs in a centralized location.



Task 2: Connect Azure Activity to Sentinel

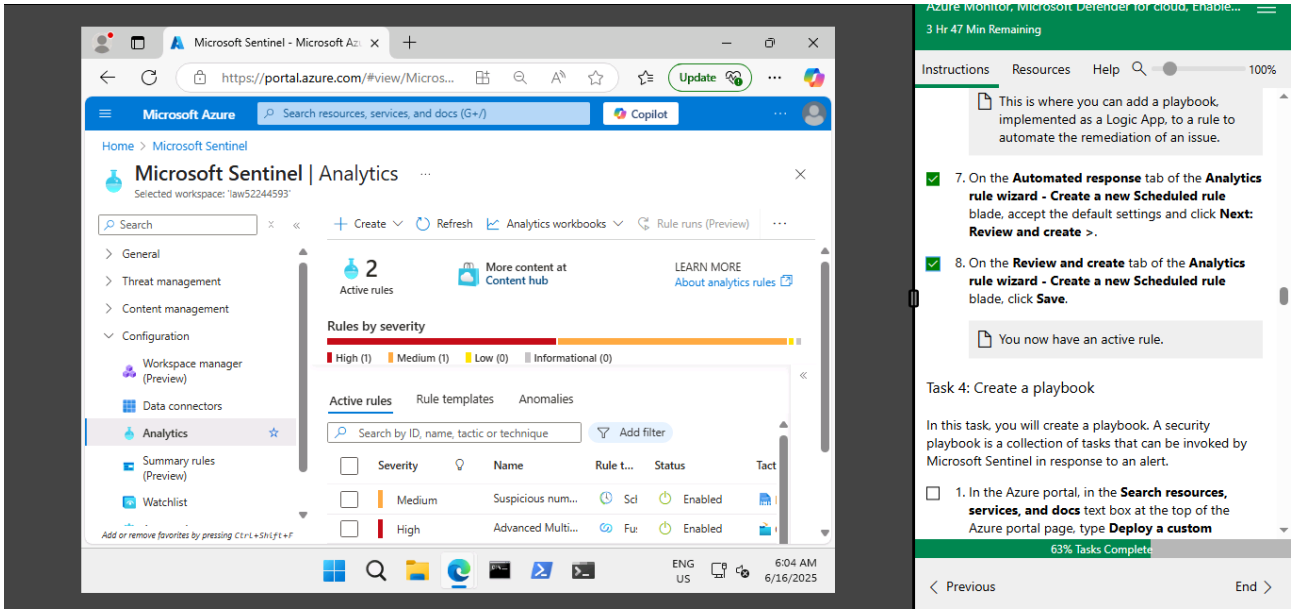
In the second task, I configured Microsoft Sentinel to connect to the Azure Activity data connector. This step involved navigating to the Content Hub, installing the Azure Activity content, and assigning a policy to stream activity logs to the Log Analytics workspace. By doing this, I ensured that all relevant activity within the Azure environment would be captured and analyzed by Sentinel. This stream of activity data is essential for detecting suspicious behaviors and generating alerts based on anomalous or unauthorized actions within the cloud infrastructure.





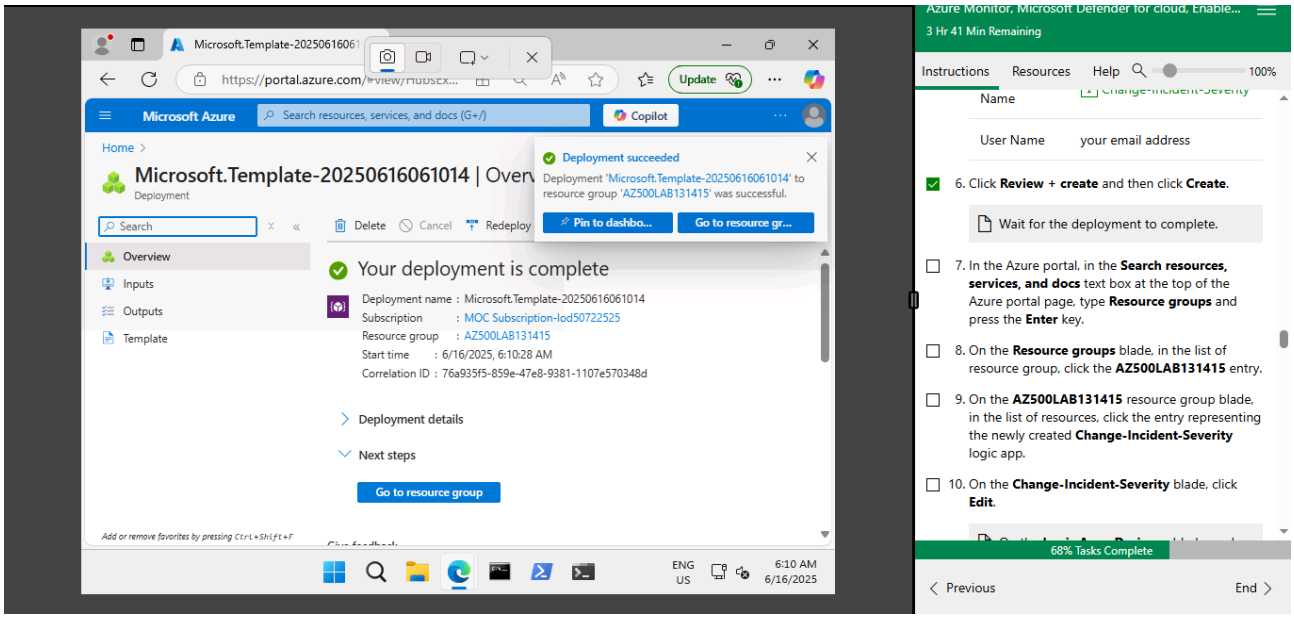
Task 3: Create a rule that uses the Azure Activity data connector.

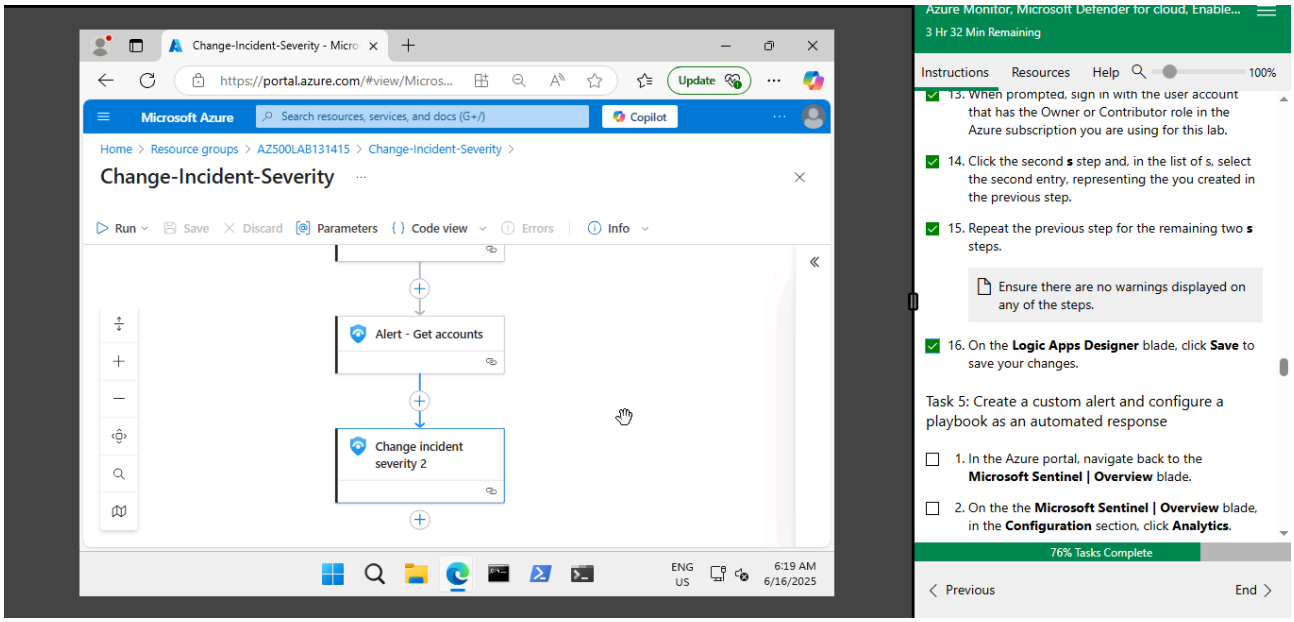
The third task focused on creating an analytics rule using the Azure Activity data source. I selected a rule template titled "Suspicious number of resource creation or deployment", which is designed to flag potentially malicious or unusual provisioning activity. I went through the configuration wizard, accepted the default logic settings, and enabled the rule to actively monitor incoming data. Though I did not add automation at this stage, the rule provided the capability to detect and raise incidents based on the defined logic, enhancing my environment's ability to identify threats in real time.



Task 4: Create a playbook

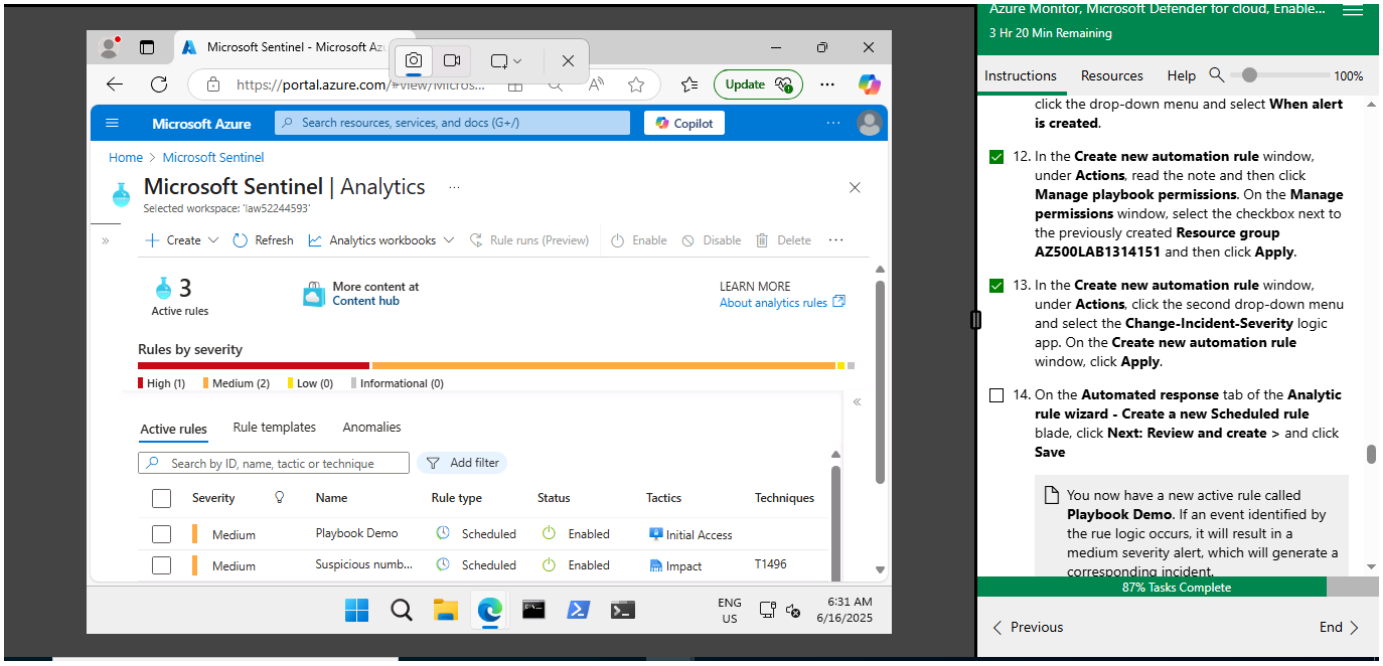
I then created a security playbook to automate responses to security incidents. Using a JSON template, I deployed a Logic App called Change-Incident-Severity in the East US region. After deployment, I edited the Logic App steps within the designer to authenticate connections and finalize the configuration. This playbook was designed to automatically respond to triggered alerts by modifying incident severity—showcasing the power of Microsoft Sentinel's automated incident handling. By the end of the lab, I had successfully set up end-to-end threat detection and response capabilities, including data collection, rule creation, and automated remediation, forming a complete security operations workflow.


Microsoft.Template-20250616061
https://portal.azure.com/
Update
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home
Microsoft.Template-20250616061014 | Overv
Deployment
Deployment succeeded
Deployment 'Microsoft.Template-20250616061014' to resource group 'AZ500LAB131415' was successful.
Pin to dashbo...
Go to resource gr...
Search
Delete
Cancel
Redeploy
Overview
Inputs
Outputs
Template
Your deployment is complete
Deployment name : Microsoft.Template-20250616061014
Subscription : MOC Subscription-lod50722525
Resource group : AZ500LAB131415
Start time : 6/16/2025, 6:10:28 AM
Correlation ID : 76a935f5-859e-47e8-9381-1107e570348d
Deployment details
Next steps
Go to resource group
Add or remove favorites by pressing Ctrl+Shift+F
ENG
US
6:10 AM
6/16/2025
Azure Monitor, Microsoft Defender for cloud, Enable...
3 Hr 41 Min Remaining
Instructions
Resources
Help
100%
Name
User Name
your email address
6. Click Review + create and then click Create.
Wait for the deployment to complete.
7. In the Azure portal, in the Search resources, services, and docs text box at the top of the Azure portal page, type Resource groups and press the Enter key.
8. On the Resource groups blade, in the list of resource group, click the AZ500LAB131415 entry.
9. On the AZ500LAB131415 resource group blade, in the list of resources, click the entry representing the newly created Change-Incident-Severity logic app.
10. On the Change-Incident-Severity blade, click Edit.
68% Tasks Complete
Previous
End


Change-Incident-Severity - Micro
https://portal.azure.com/#view/Micros...
Update
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > Resource groups > AZ500LAB131415 > Change-Incident-Severity
Change-Incident-Severity
Run
Save
Discard
Parameters
Code view
Errors
Info
Alert - Get accounts
Change incident severity 2
ENG
US
6:19 AM
6/16/2025
Azure Monitor, Microsoft Defender for cloud, Enable...
3 Hr 32 Min Remaining
Instructions
Resources
Help
100%
13. When prompted, sign in with the user account that has the Owner or Contributor role in the Azure subscription you are using for this lab.
14. Click the second s step and, in the list of s, select the second entry, representing the you created in the previous step.
15. Repeat the previous step for the remaining two s steps.
Ensure there are no warnings displayed on any of the steps.
16. On the Logic Apps Designer blade, click Save to save your changes.
Task 5: Create a custom alert and configure a playbook as an automated response
1. In the Azure portal, navigate back to the Microsoft Sentinel | Overview blade.
2. On the the Microsoft Sentinel | Overview blade, in the Configuration section, click Analytics.
76% Tasks Complete
Previous
End

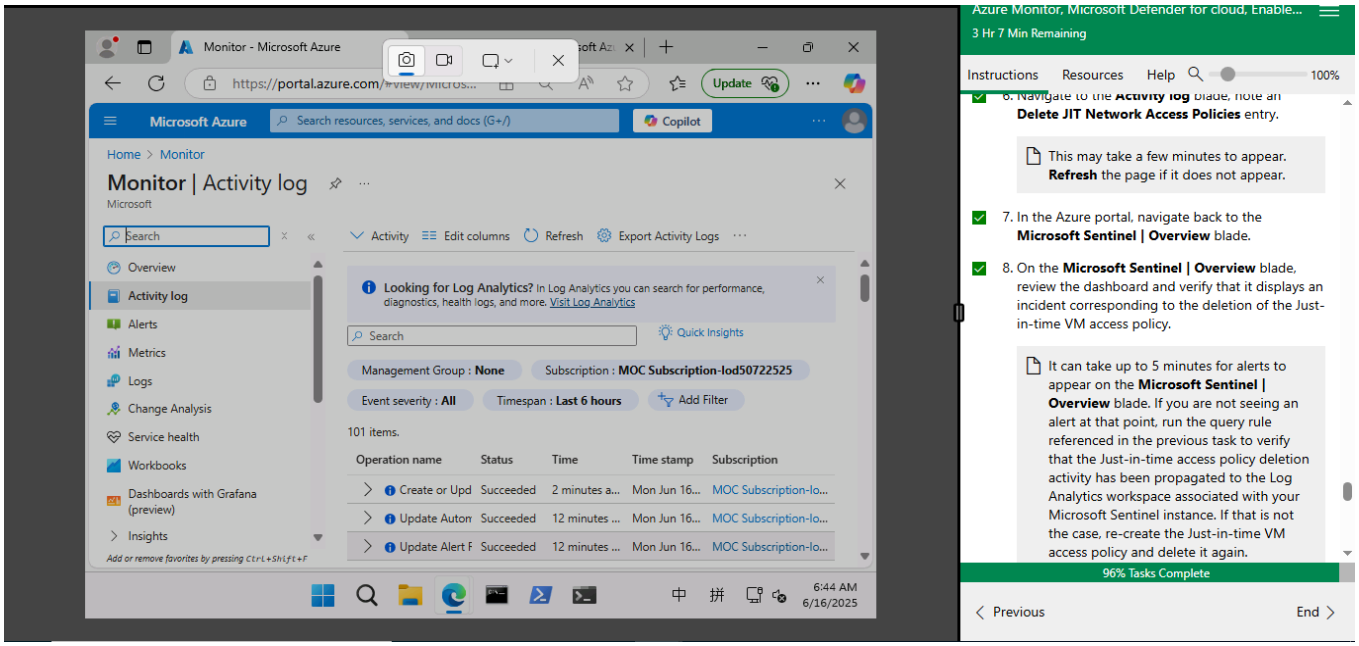
Task 5: Create a custom alert and configure the playbook as an automated response.

In Task 5 of the lab, I created a custom alert in Microsoft Sentinel and configured it to trigger a playbook as an automated response. This began with setting up a Scheduled query rule named Playbook Demo, designed to detect the deletion of Just-in-time (JIT) VM access policies—a potential sign of unauthorized or malicious activity. The rule used a KQL (Kusto Query Language) script to query Azure Activity logs for specific operations associated with JIT policy deletions. I scheduled the rule to run every five minutes and linked it to a playbook titled Change-Incident-Severity, which was configured as an automation rule that runs when an alert is created. This automation helps enforce consistent, immediate responses to security incidents without manual intervention.



Task 6: Invoke an incident and review the associated actions.

In Task 6, I validated the effectiveness of the custom alert and playbook by simulating a security event. I accessed Microsoft Defender for Cloud and manually removed a JIT access policy from a virtual machine named myVM. This simulated activity triggered the alert configured in Task 5. I then reviewed the Activity log to confirm the deletion was logged and returned to Microsoft Sentinel to verify that a corresponding incident was generated. The incident appeared on the Sentinel dashboard with the expected severity, confirming the rule and playbook were functioning as intended. Finally, I checked the Playbooks section to confirm the playbook executed successfully, demonstrating that Microsoft Sentinel can automate threat detection and response workflows effectively.



Conclusion:

By the end of Lab 11, I had successfully configured Microsoft Sentinel to act as a proactive security operations center (SOC) tool. The lab demonstrated how to onboard Sentinel, connect critical data sources, create actionable analytics rules, and automate incident response through playbooks. I simulated a real-world security event by deleting a Just-in-time VM access policy and verified that Sentinel not only detected the activity but also automatically triggered a predefined response. This hands-on experience highlighted Sentinel's powerful capabilities in monitoring, detecting, and responding to security threats, solidifying my understanding of how to operationalize cloud-native security in Microsoft Azure.