Soares, Olusegun Leo

Capstone Final Report - PGPCS-UTA, April 2025

1. **Introduction**

This report provides an in-depth implementation and analysis of Microsoft Sentinel's security operations, including its integration with Microsoft Defender for Cloud, advanced log analysis using KQL queries, threat intelligence integration, automated incident detection, and proactive threat hunting for improved cloud threat detection and response.

# **Objectives Completed**

This project is divided into two (2) parts.

**Part A: Protecting Azure resources with Sentinel**

* *Deployed a VM on Azure Environment.*
* *Deployed Microsoft Sentinel.*
* *Detected Failed Authentication events on Windows Server virtual machine.*
* *Detected File Access and Modifications events on Windows Server virtual machine.*

***Part B: Investigating and threat hunting with Sentinel***

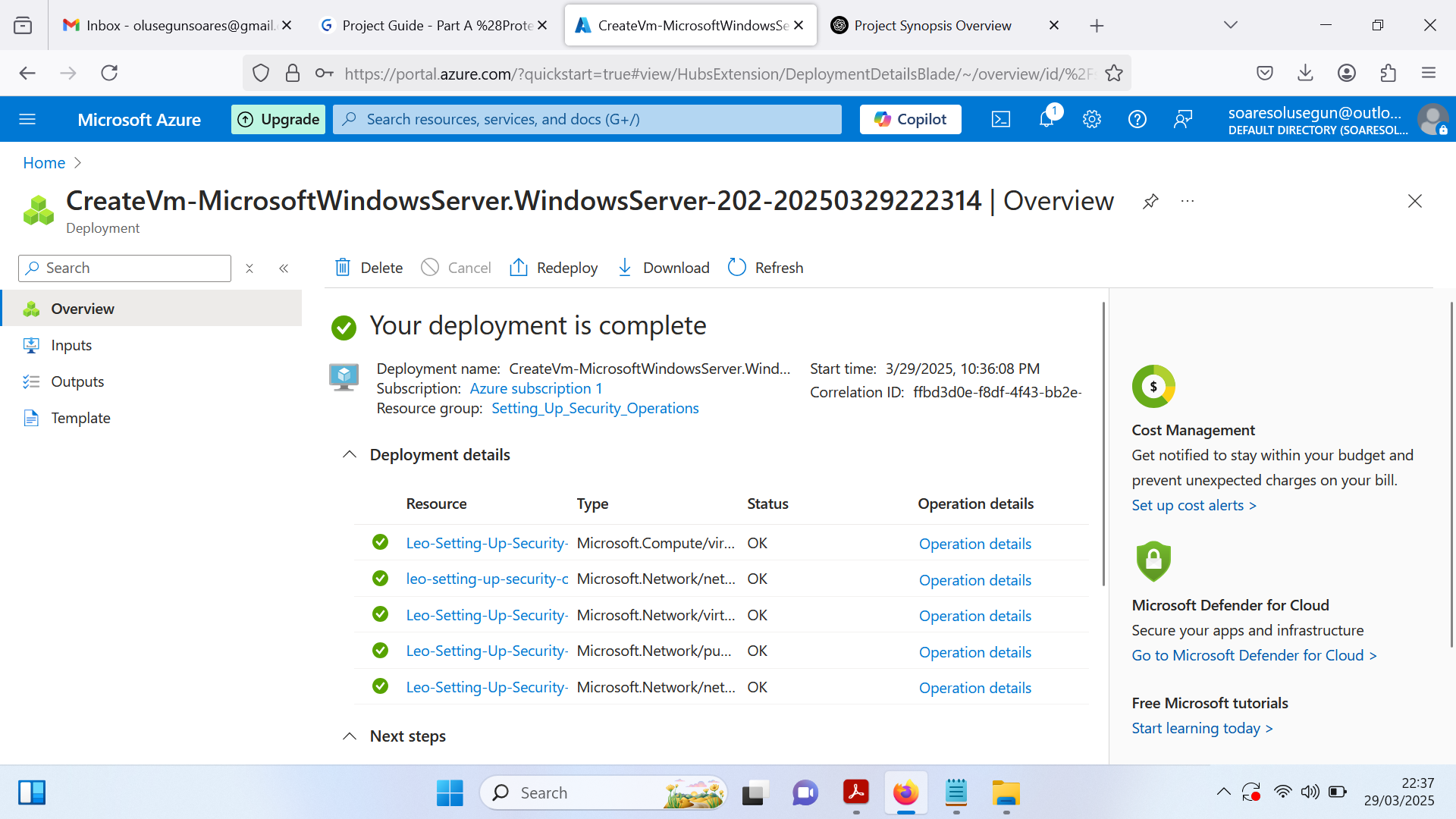
* *Connected Defender with Sentinel.*
* *Performed Incident Management with Sentinel.*
* *Performed Threat Hunting against a specific Tactic.*
* *Added Threat Intelligence data to Sentinel workspace.*
* *Launched a Hunt using Threat Intelligence Objects.*

1. **Implementation of Security Operations in Microsoft Sentinel**

**Part A**

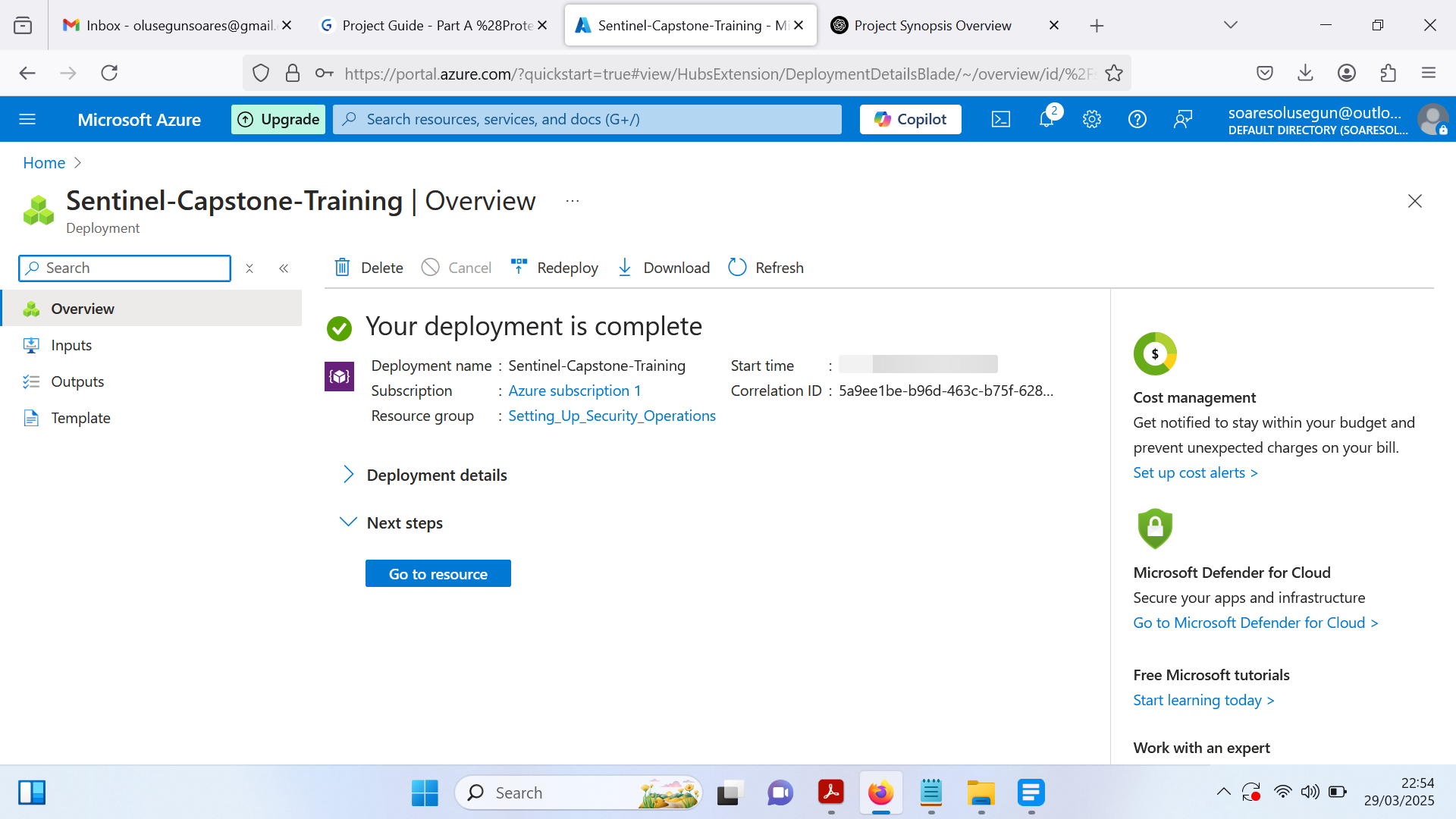
**Task 1: Deployment of a Virtual Machine (VM) on Azure Environment**

Deploy a virtual machine (VM) in Azure and integrate it into the Microsoft Azure environment, ensuring proper networking, security configurations, and resource group association.



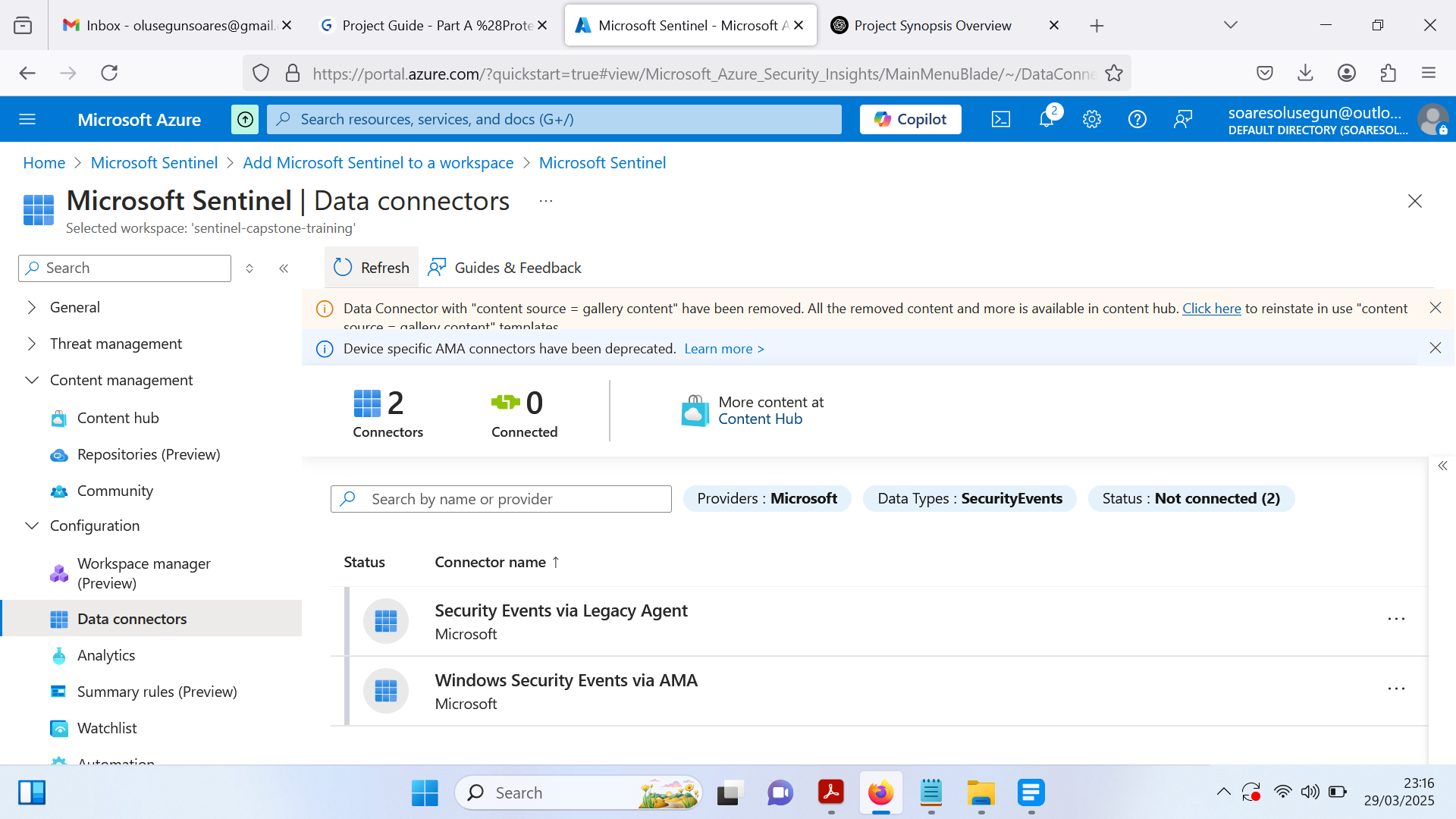
**Task 2: Deploying Microsoft Sentinel**

Create a new workspace in Microsoft Sentinel, select the appropriate subscription, and link it to the resource group used during the VM deployment.

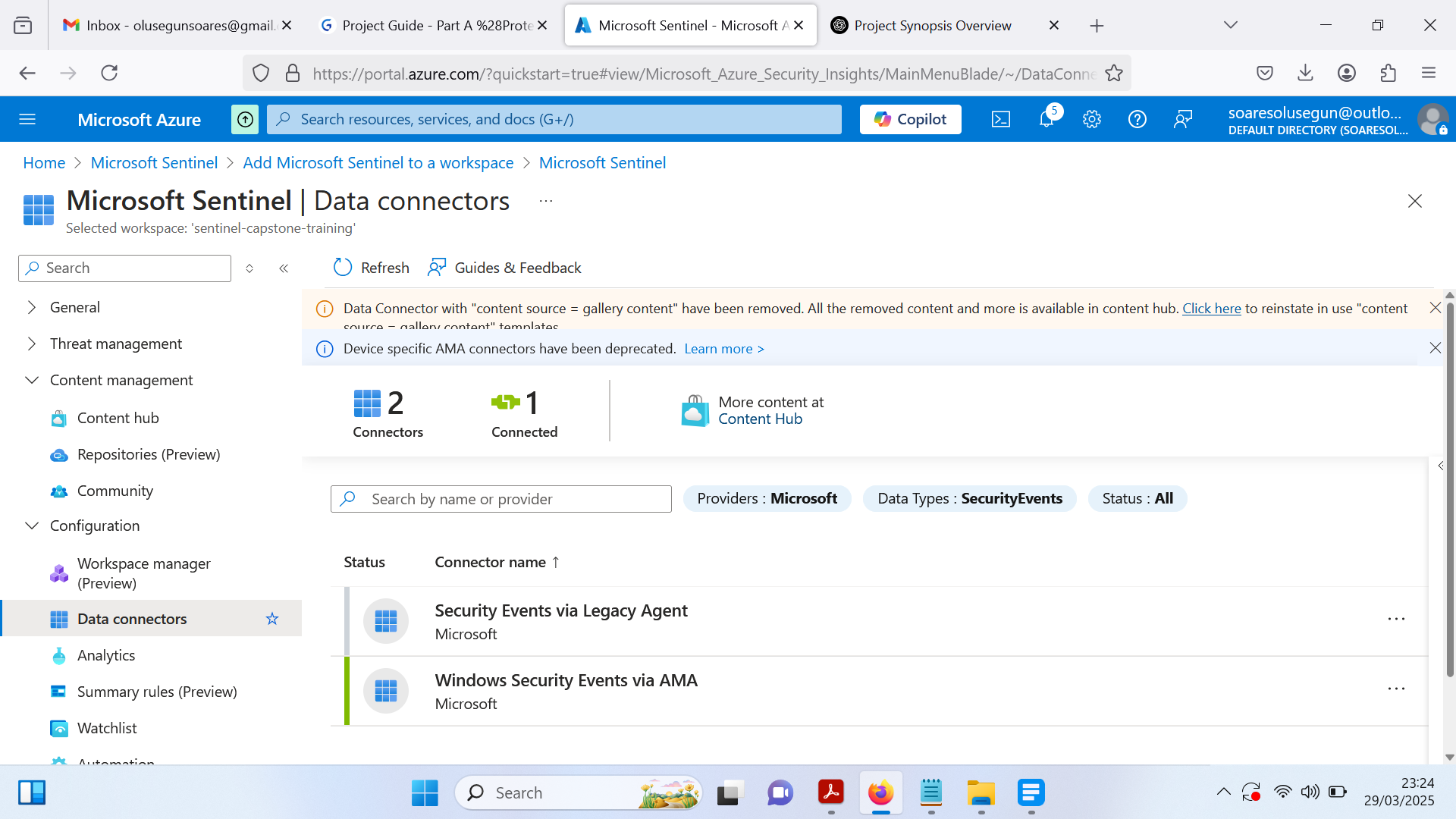


**Task 3: Setting up Data Connector on Sentinel**

To set up a Data Connector in Microsoft Sentinel, go to Sentinel > Data Connectors and install the required connector via the Content Hub (e.g., Windows Security Events). Then, navigate to Data Connectors, where two connectors will appear, but none will be in the connected state.

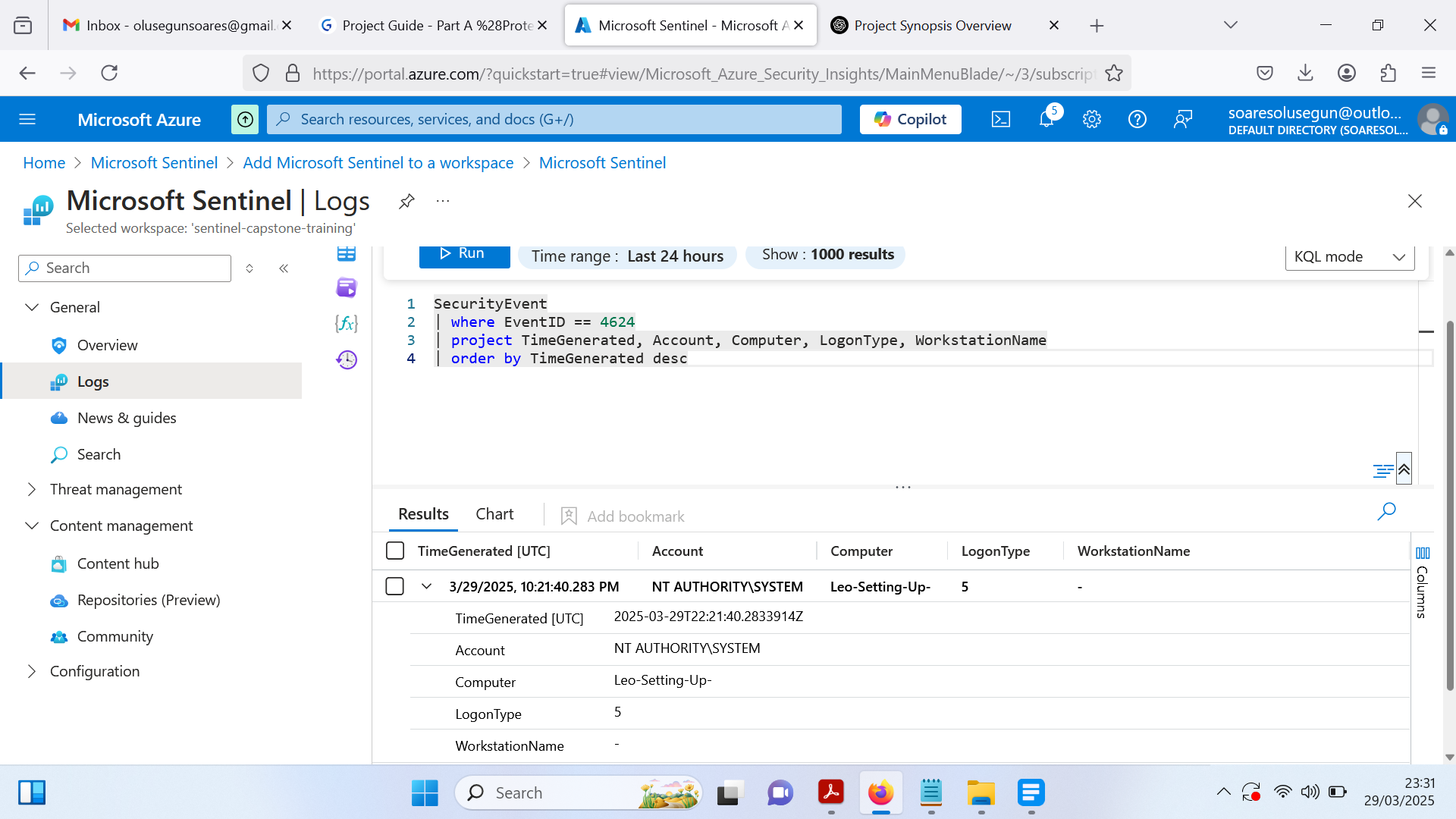


Next, connect the Windows Security Events via AMA connector to the previously deployed Windows Server VM. Once connected, the status should update to Enabled, confirming successful integration.

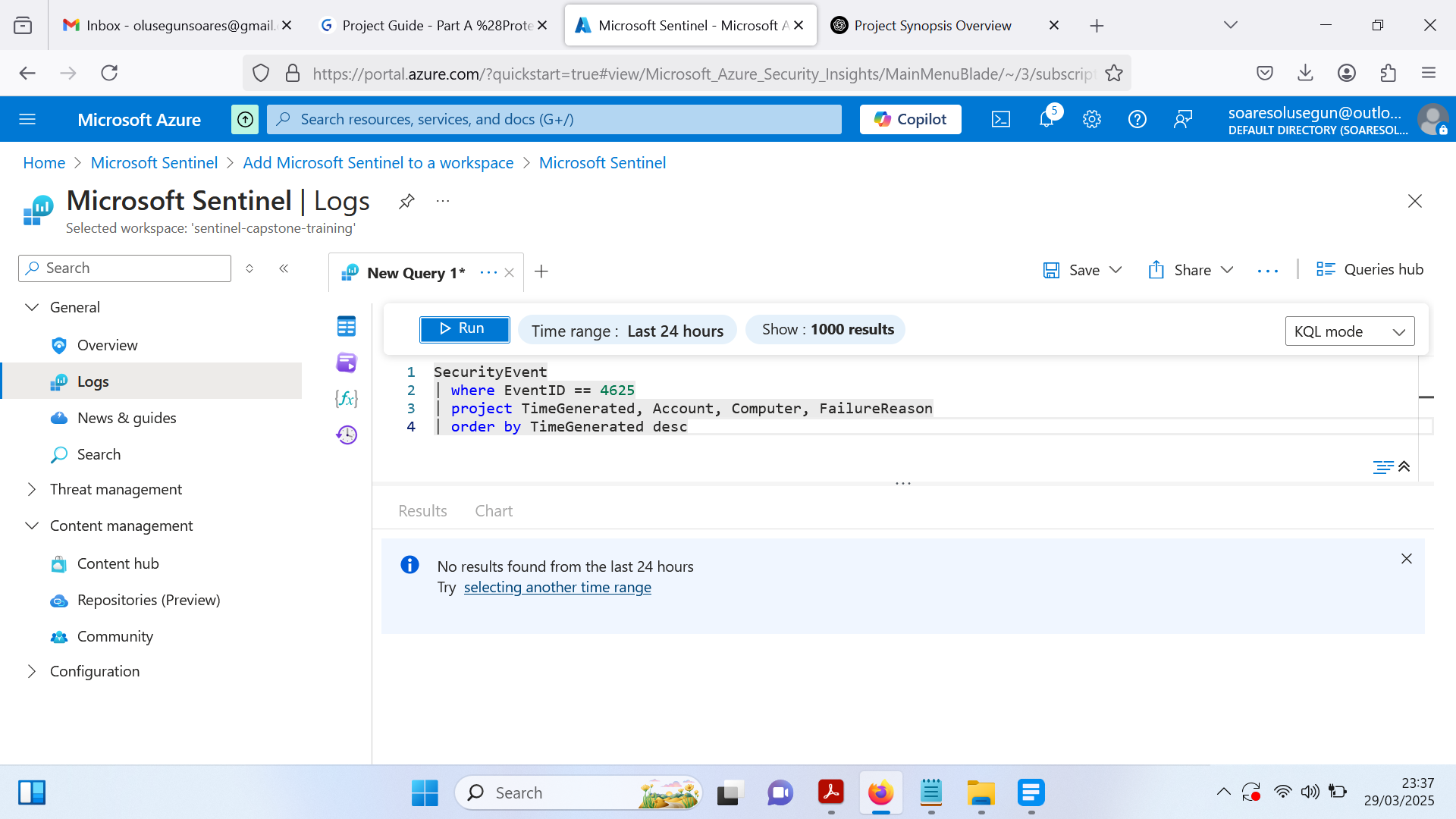


**Task 4: Detecting Failed Authentication Events on Windows Server Virtual Machine**

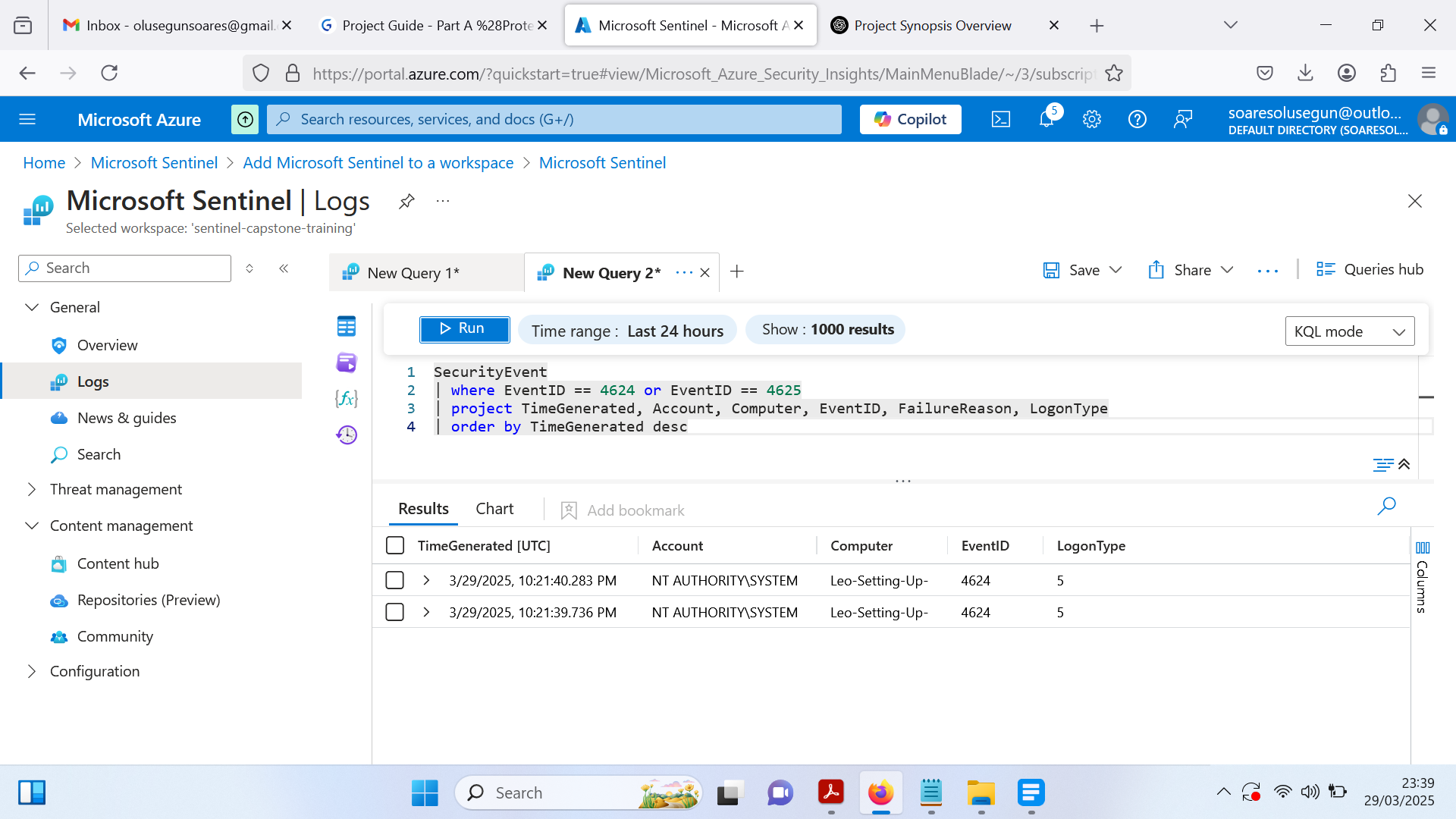
In Microsoft Sentinel, navigate to the Logs section to access captured logs from the Windows Server VM, then execute the KQL query below to filter and display recent successful logon events (EventID 4624) from the SecurityEvent table, sorted by timestamp for easy analysis.



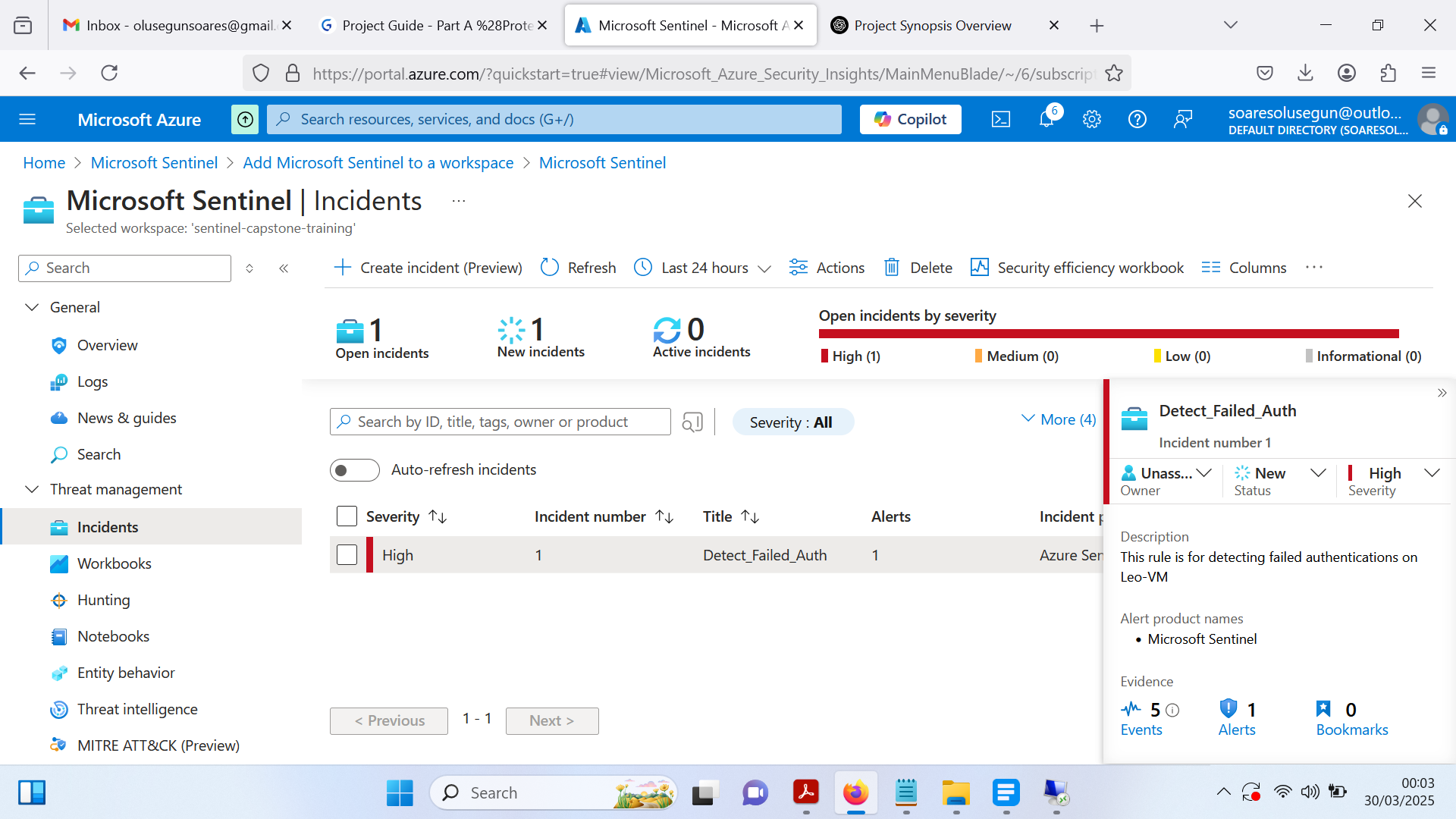
Execute the KQL query in Microsoft Sentinel’s Logs section to filter and display failed logon events (EventID 4625) from the SecurityEvent table, showing relevant details such as timestamp, account, computer, and failure reason, though no failed logon events were found.



Run the KQL query in Microsoft Sentinel’s Logs section to retrieve both successful (EventID 4624) and failed (EventID 4625) logon events from the SecurityEvent table, displaying key details such as timestamp, account, computer, event ID, failure reason, and logon type in descending order.

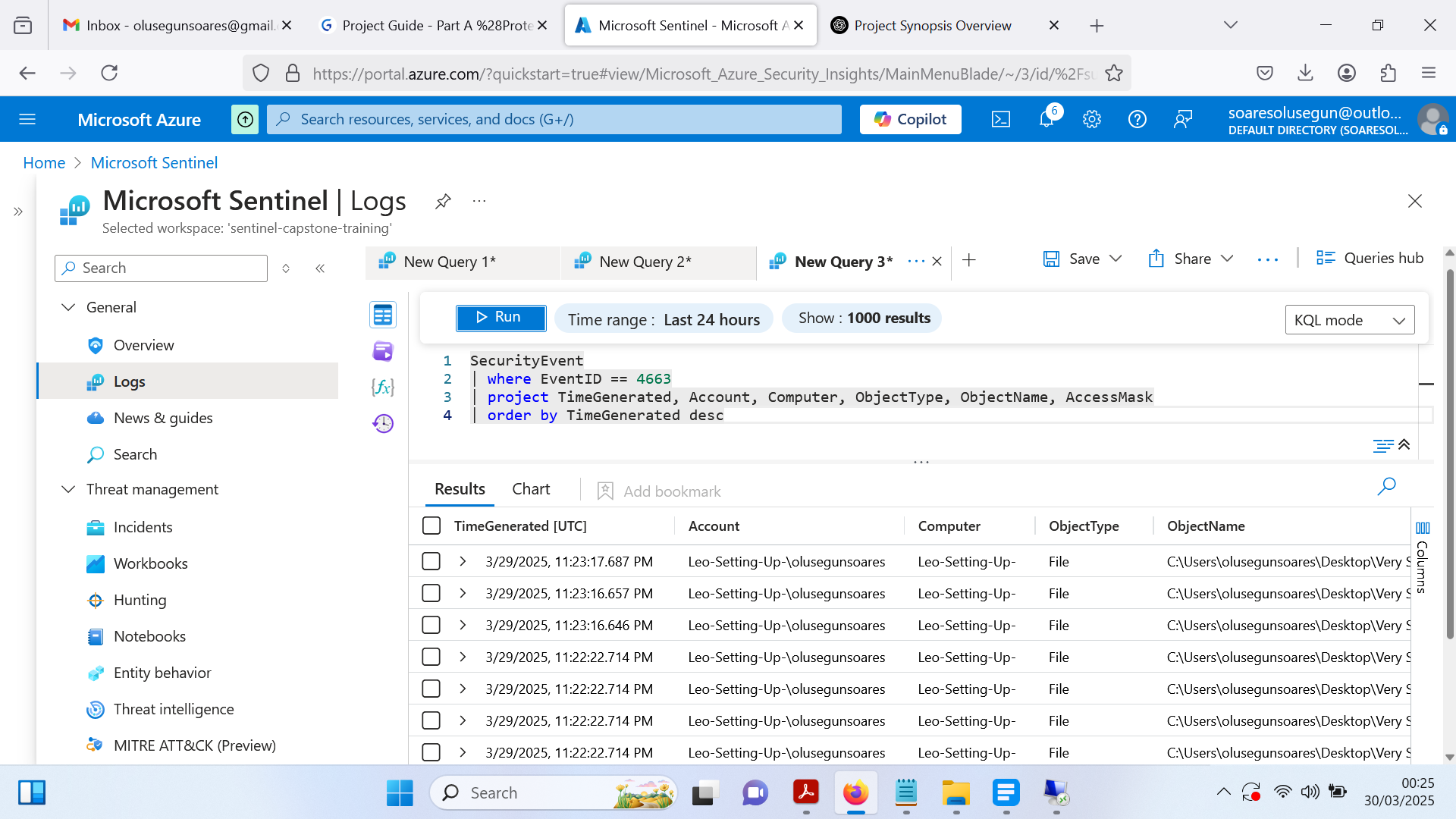


Create an alert in Microsoft Sentinel to detect failed authentication events on the VM by naming it **Detect\_failed\_auth**, setting Severity to High, and mapping it to MITRE ATT&CK T1110 – Brute Force; configure the rule logic using a KQL query to filter EventID 4625, then attempt multiple failed RDP logins to trigger and verify the incident in the Incidents section.

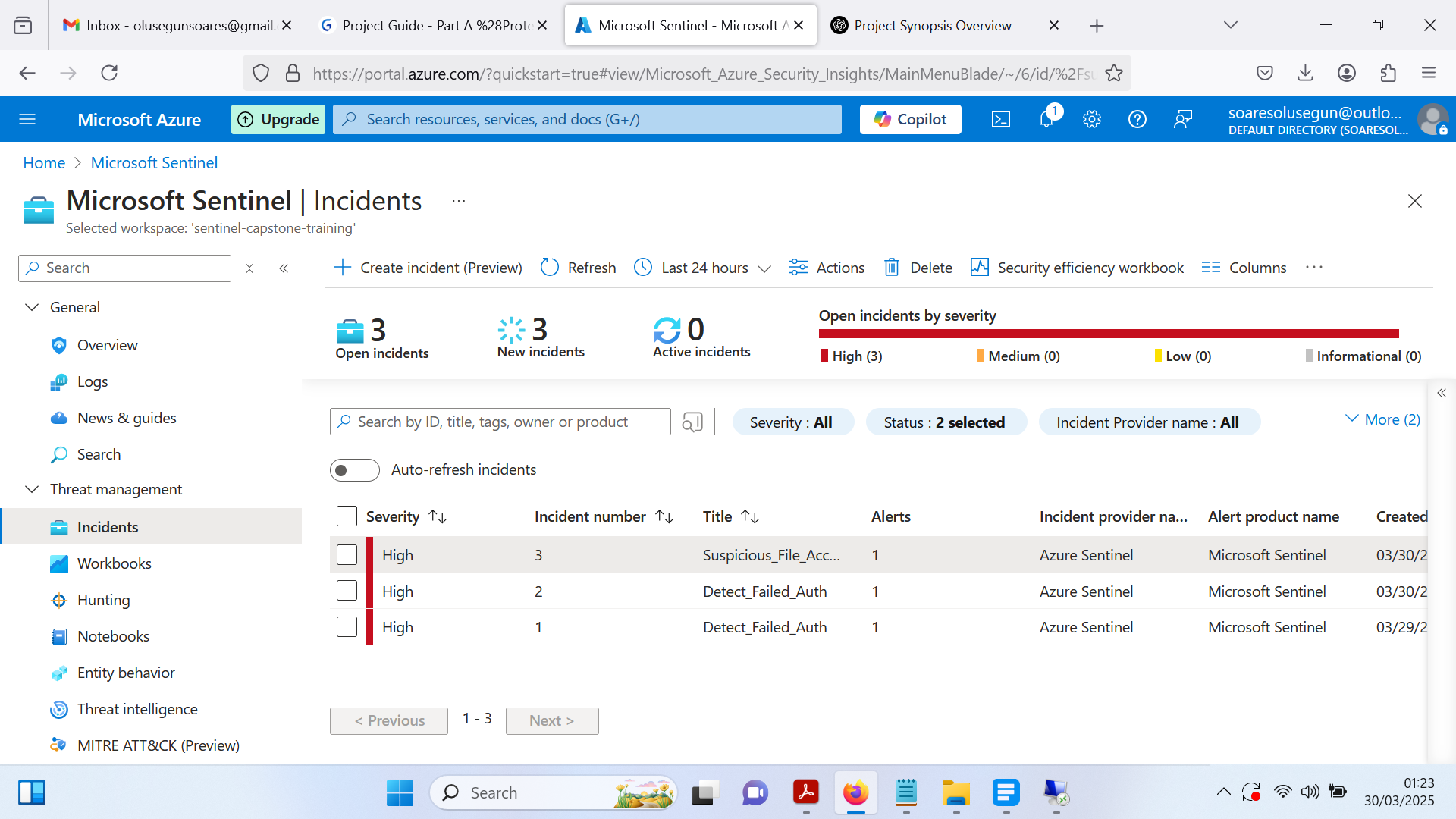


**Task 5: Detecting File Access and Modifications events on Windows Server Virtual Machine**

To detect file access or modification (Event ID 4663), enable the Audit Policy for File Access via gpedit.msc, navigate to Advanced Audit Policy Configuration → Object Access, and enable Audit File System for both success and failure events; then create and configure auditing for a monitored folder (Very Secret) by setting Testadmin as the principal, applying full auditing permissions, and ensuring logs capture access attempts; finally, return to the Azure portal, open Logs in Microsoft Sentinel, and run a KQL query to check for file access and modification logs.



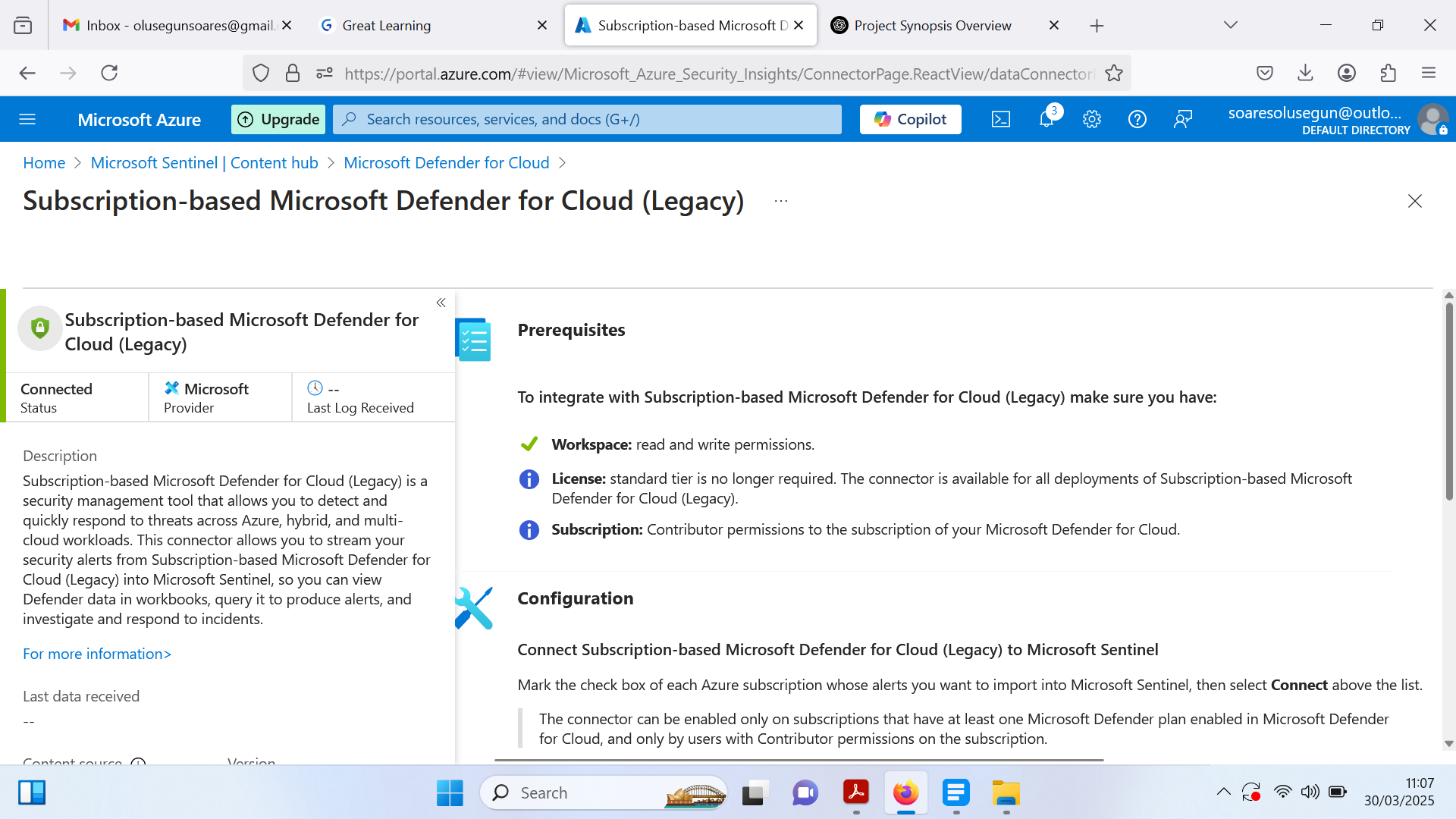
To monitor modifications in the Very Secret folder, run a KQL query in Microsoft Sentinel Logs to filter for Event ID 4663, specifically targeting file access events within the designated path; then, create an alert rule for file deletion events by navigating to + New alert rule > Create Microsoft Sentinel alert, configure it in the Analytics rule wizard under General configuration with the name **Suspicious\_file\_access**, set the Severity to High, assign the MITRE ATT&CK technique (T1041 – Exfiltration over C2 Channel), and define the KQL query to detect deletion attempts based on AccessMask 0x10000; once saved, verify the rule creation in the Analytics section, delete Confidential.txt from the Very Secret folder on the VM, wait for approximately 10-11 minutes, and check the Incidents section in Microsoft Sentinel for a new alert.



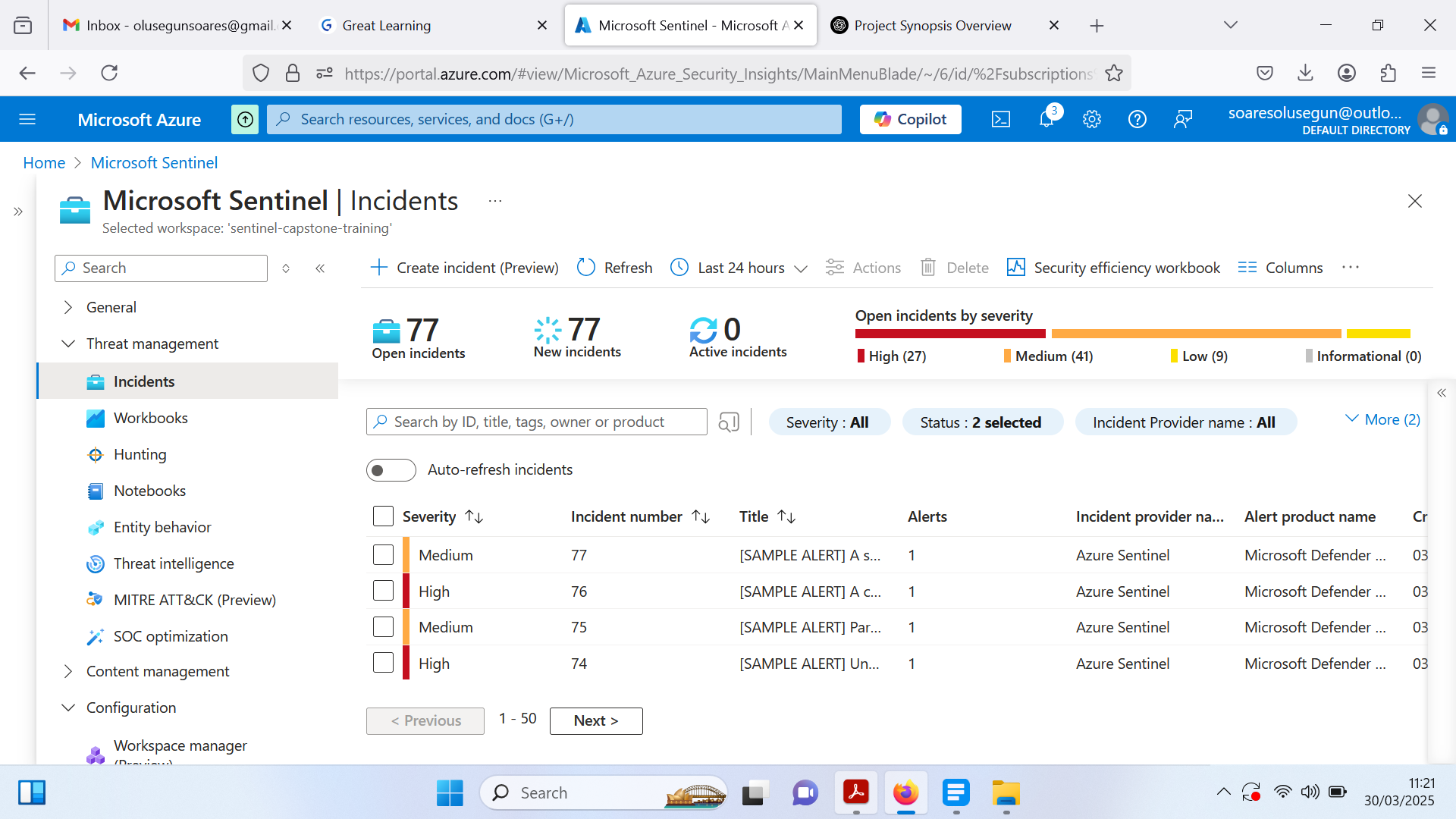
**Part B**

**Task 1: Connecting Defender with Sentinel**

The process begins with the deployment of Microsoft Defender for Cloud, followed by Microsoft Sentinel to enable advanced security monitoring. If Sentinel is already deployed, proceed to install the Microsoft Training Lab solution from the Marketplace. To establish a connection between Defender for Cloud and Sentinel, install the relevant Data Connector via the Content Hub, search for Microsoft Defender for Cloud, select it, and proceed with the installation. Once installed, navigate to Manage > Configuration Page, choose Subscription-based Microsoft Defender for Cloud (Legacy), and toggle the connection status to Connected. Verify the successful connection by checking the Data Connector page in Sentinel, where the status should be highlighted in Green.

****

With the connection established, create an Analytics Rule to ensure security alerts from Defender for Cloud are logged as Incidents in Sentinel. Navigate to Analytics > Create > Microsoft Incident Creation Rule, provide a Name and Description, enable the rule, and set the Microsoft Security Service to Microsoft Defender for Cloud, filtering severity as Any before saving the configuration. To validate the setup, generate sample alerts in Defender for Cloud by selecting Sample Alerts, ensuring the correct subscription and plans are selected, then create the alerts. Finally, navigate to the Microsoft Sentinel Incidents Page, where the generated sample alerts should now be reflected as Incidents on the dashboard, confirming successful integration.

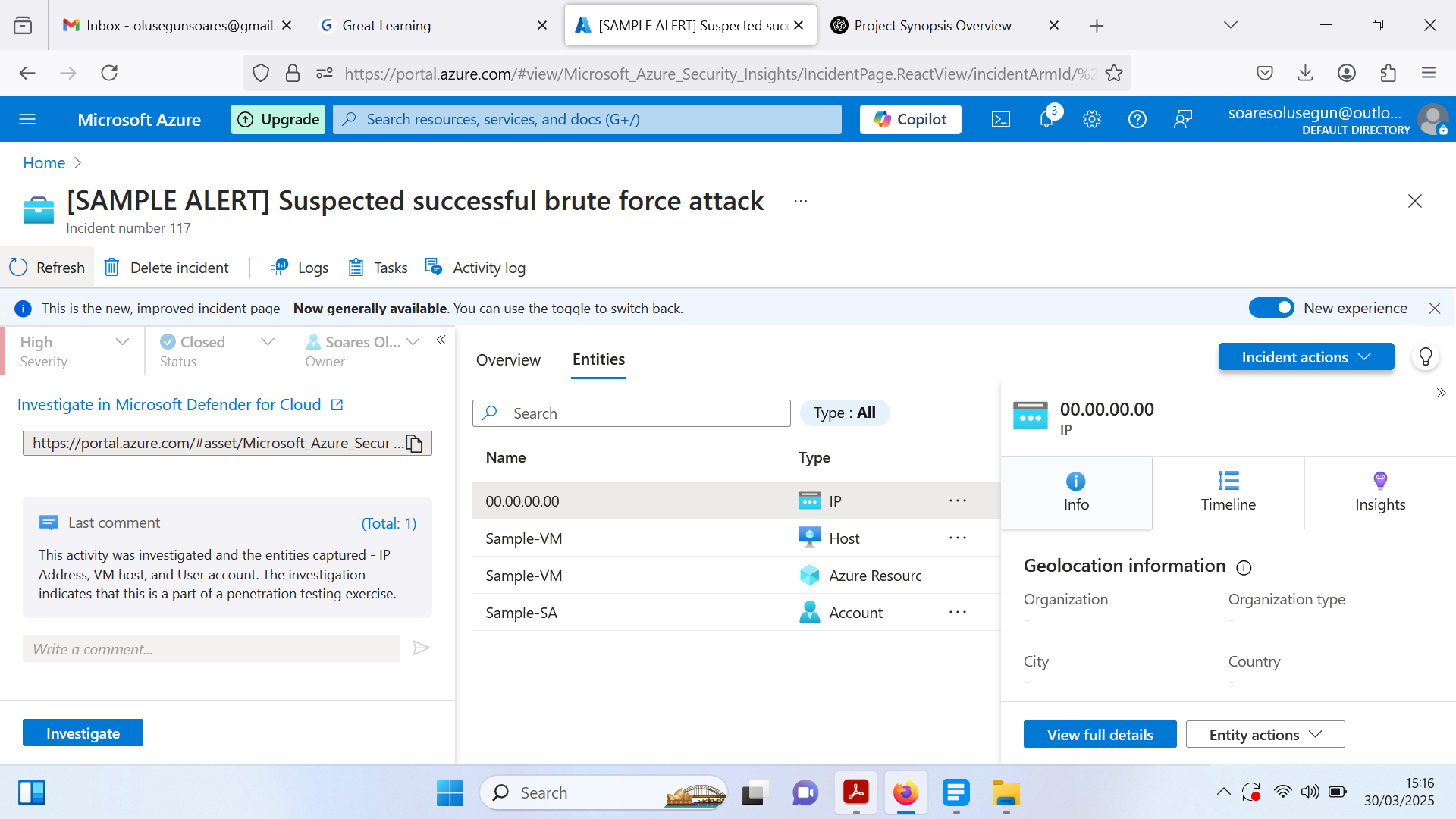
****

**Task 2: Incident Management with Sentinel**

Locate the Suspected Successful Brute Force Attack incident in Microsoft Sentinel and assign ownership to yourself using the Assign to me option. Open the incident details to investigate and identify the four entities involved. Since this is a controlled Red Teaming exercise, mark the incident as Benign to indicate no actual threat.

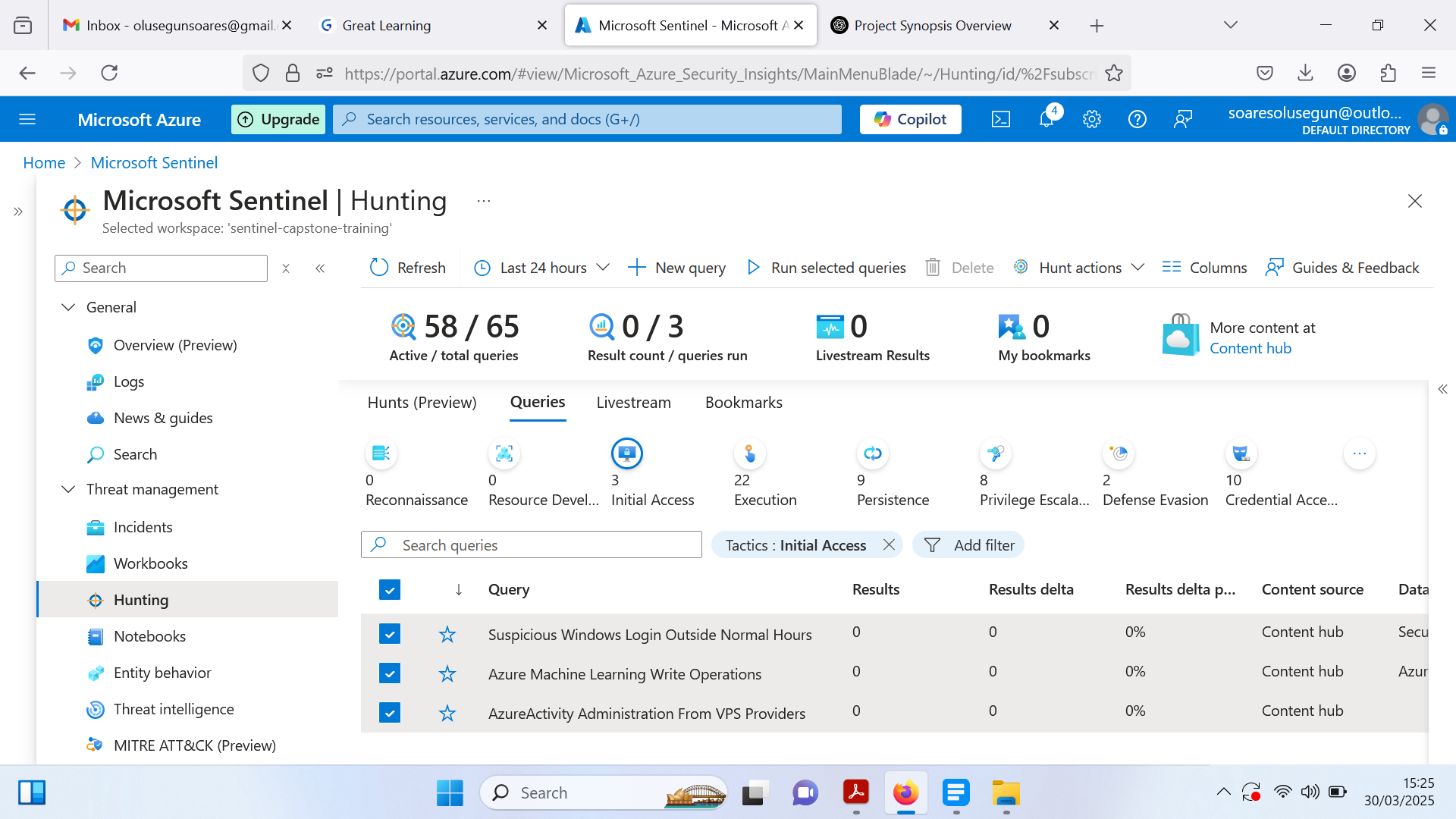
****

Next, investigate the IP address and associated entities to gather geographic details and assess its origin. Upon discovering that the IP is used by the organization's penetration testing team as part of a structured Red Teaming exercise, classify the incident as ***Benign***. To prevent unnecessary alerts, create an automation rule by selecting ***Create Automation Rule*** from the ***Incident Actions*** dropdown, setting the expiration date to match the engagement's timeline. On the ***Incident*** Page, change the status to ***Closed*** and classify it as ***Benign Positive*** – Suspicious but expected. Finally, document the findings in the comments, noting that the activity involved an IP address, VM host, and user account as part of a controlled security test.

****

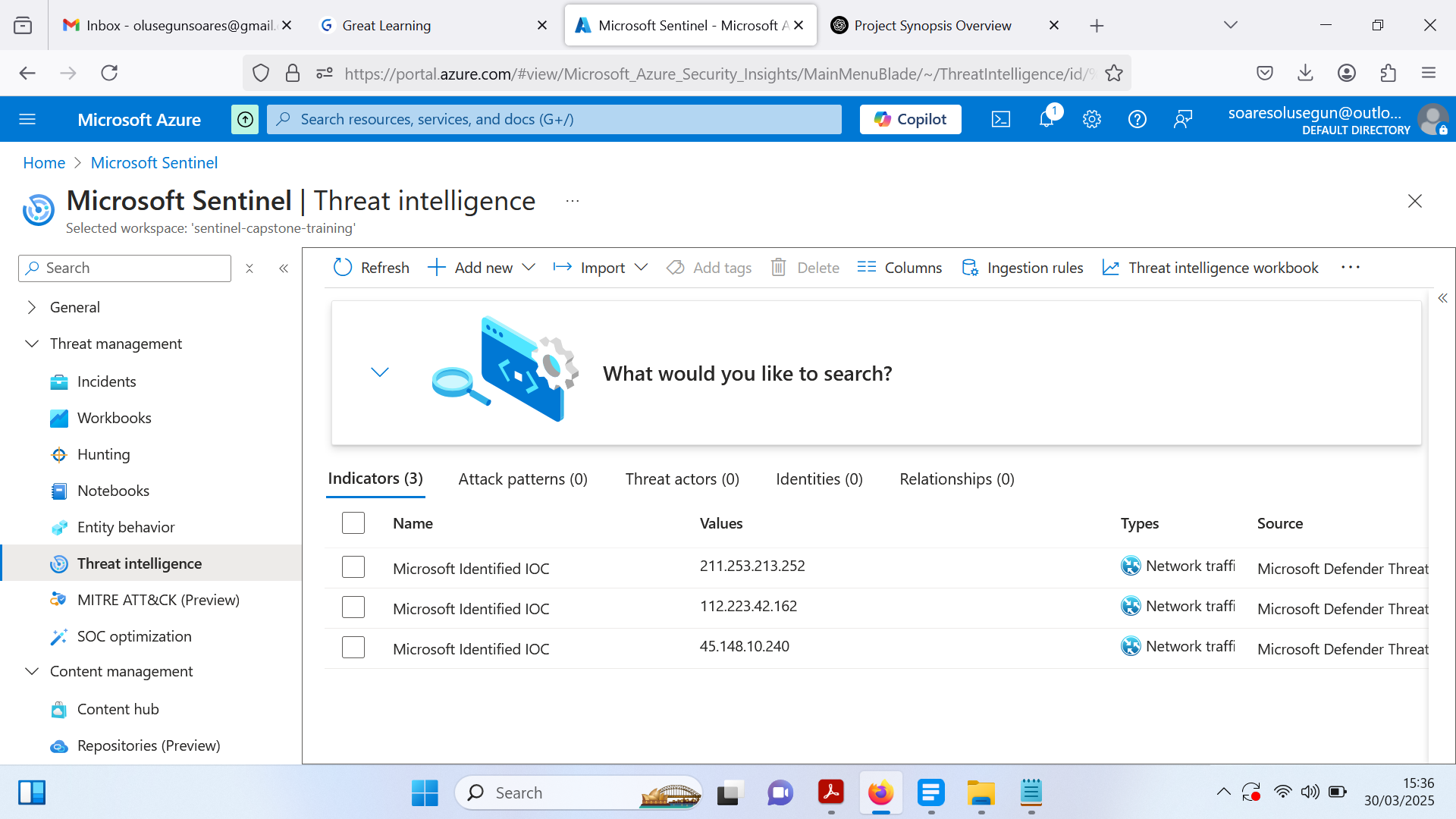
**Task 3: Threat Hunting Against A Specific Tactic**

Evaluate your cloud instance for potential malicious Initial Access attempts by navigating to Microsoft Sentinel > Incidents. In the Hunting section, under Queries, install new queries from the Content Hub by searching for Azure Activity and completing the installation. Once installed, filter queries by the Initial Access tactic (Add Filter > Tactics > Initial Access), then analyze the results. If no results appear, it indicates no logged attempts of unauthorized access via phishing, drive-by compromise, or exploitation of public-facing applications.

****

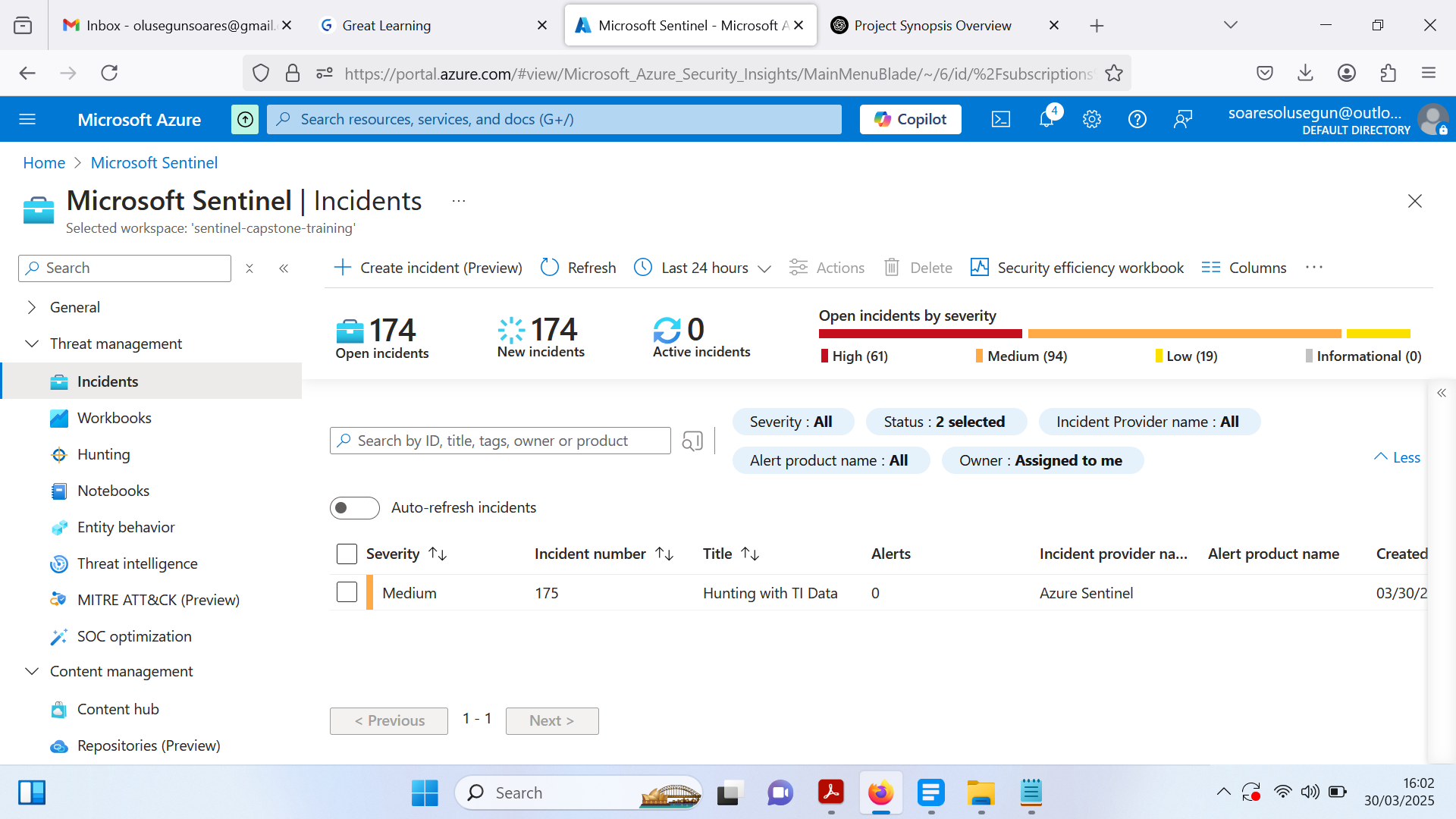
**Task 4: Adding Threat Intelligence Data To Sentinel Workspace**

Configure Microsoft Sentinel with Threat Intelligence (TI) data by navigating to the Threat Intelligence blade. If no TI objects are present, select Visit TI solution in Content Hub and install the Microsoft Defender Threat Intelligence solution. Once installed, open the Connector page, select Connect, and refresh the Threat Intelligence page to verify the newly added objects. These TI objects enhance incident analysis by enriching alerts with contextual data, supporting KQL-based threat hunting, and integrating into automation and detection rules.

****

**Task 5: Launching A Hunt Using Threat Intelligence Objects**

Initiate a Threat Intelligence (TI) Hunt in Microsoft Sentinel by navigating to Hunting, selecting + New Hunt, and configuring details such as Hunt Name, Description, Owner, and Hypothesis. Once created, add relevant TI queries by filtering for Threat Intelligence and selecting all available queries. Execute the hunt by running all queries and reviewing results within five minutes. If logs match TI objects, create an incident by selecting Create Incident from the Actions dropdown, confirm pre-filled details, assign ownership, and finalize the creation. Verify the incident under the Incidents blade by filtering for those assigned to you.

****

1. ***Analysis of the Implemented Security Operations in Microsoft Sentinel***

*Going through these implementations, I can see how each step contributes to building a strong and proactive security monitoring system in Microsoft Sentinel. By setting up Sentinel and integrating it with Microsoft Defender for Cloud, I established a strong foundation for centralizing security event management and incident response.*

*One thing that stood out to me was how effective log analysis with KQL queries can be in detecting real threats, like failed login attempts, file modifications, and brute force attacks. By fine-tuning these queries, I was able to filter out noise and focus on meaningful security events. Setting up alert rules and automation workflows added another layer of efficiency, making incident detection and response more seamless.*

*I also found the Threat Intelligence integration to be a game-changer. Bringing in external threat feeds and correlating them with Sentinel data makes it easier to identify suspicious activity early. Plus, mapping incidents to MITRE ATT&CK techniques helped add structure to the analysis, making investigations more precise and informed.*

*The threat hunting and incident response exercises reinforced the importance of being proactive rather than just reactive. Setting up hunts, analyzing security events, and creating custom incident rules showed me how a well-configured SIEM can truly enhance security visibility.*

*This entire process has been an eye-opener for me. It’s clear that building a strong cloud security posture goes beyond simply setting up tools, it requires continuous refinement of detection rules, responding quickly to threats, and staying ahead of attackers. Through these implementations, I've gained a solid, hands-on understanding of how Microsoft Sentinel actively enhances security monitoring, making it an invaluable tool for real-time cyber threat detection, investigation, and response.*

1. **Technical Challenges and Resolutions**

During the implementation of Microsoft Sentinel’s security operations, I encountered several technical challenges that required troubleshooting and optimization. Below are some key obstacles and how I resolved them:

* **Data Connector Integration Issues**

**Challenge:** After integrating Microsoft Defender for Cloud with Sentinel, the data connector did not immediately reflect as “Connected”.

**Resolution:** I manually refreshed the Data Connector page, verified my subscription settings, and confirmed that the necessary permissions in Defender for Cloud were properly assigned. This resolved the issue, ensuring logs were ingested correctly.

* **Delayed Threat Intelligence Data**

**Challenge:** After enabling the Threat Intelligence (TI) connector, TI objects didn’t appear in Sentinel as expected.

**Resolution:** I refreshed the Threat Intelligence page and allowed time for data ingestion. Additionally, I confirmed that the TI connector was properly linked to my Sentinel workspace.

* **Automation Rule Not Triggering**

**Challenge:** Despite setting up an automation rule for incident classification, alerts were not automatically categorized.

**Resolution:** I reviewed the rule conditions and realized the filters were too broad. Adjusting the severity parameters and refining entity mappings resolved the issue.

1. **Key Learnings from this Capstone Project Implementation**

* **Unified Security Monitoring is a Game-Changer:** Integrating Sentinel with Defender for Cloud demonstrated the efficiency of centralized security event management, streamlining analysis and response.
* **Threat Intelligence Enhances Decision-Making:** Bringing in external threat feeds provided critical context, making it easier to correlate threats with real-world attack patterns and respond proactively.
* **KQL Queries are Powerful but Require Optimization:** Mastering KQL allowed me to filter out noise and focus on meaningful security events, significantly improving detection accuracy.
* **Threat Hunting Elevates Security Posture:** Instead of passively waiting for alerts, actively hunting for threats helped me detect and mitigate risks earlier in the attack chain.
* **Automation is Crucial for Efficiency:** Automating incident detection and response saved time, minimized human error, and improved overall security operations.
* **Configuration Issues Can Slow You Down:** Challenges with permissions and misconfigurations reinforced the importance of proper access control and system setup from the beginning.
* **Incident Classification Must Follow Industry Frameworks:** Aligning investigations with frameworks like MITRE ATT&CK added structure to threat analysis and response strategies.
* **Security is an Ongoing Process:** Detection rules, automation scripts, and threat intelligence integrations need regular updates to keep up with evolving cyber threats.
* **SIEM Customization Enhances Effectiveness:** Tuning Sentinel’s built-in rules and creating custom detections tailored to specific environments improved its effectiveness in detecting advanced threats.

Finally, this project reinforced that cybersecurity is not just about tools but about strategy, continuous learning, and proactive defense. Microsoft Sentinel provides a robust platform, but its true power lies in how well it is configured, monitored, and adapted to emerging threats.