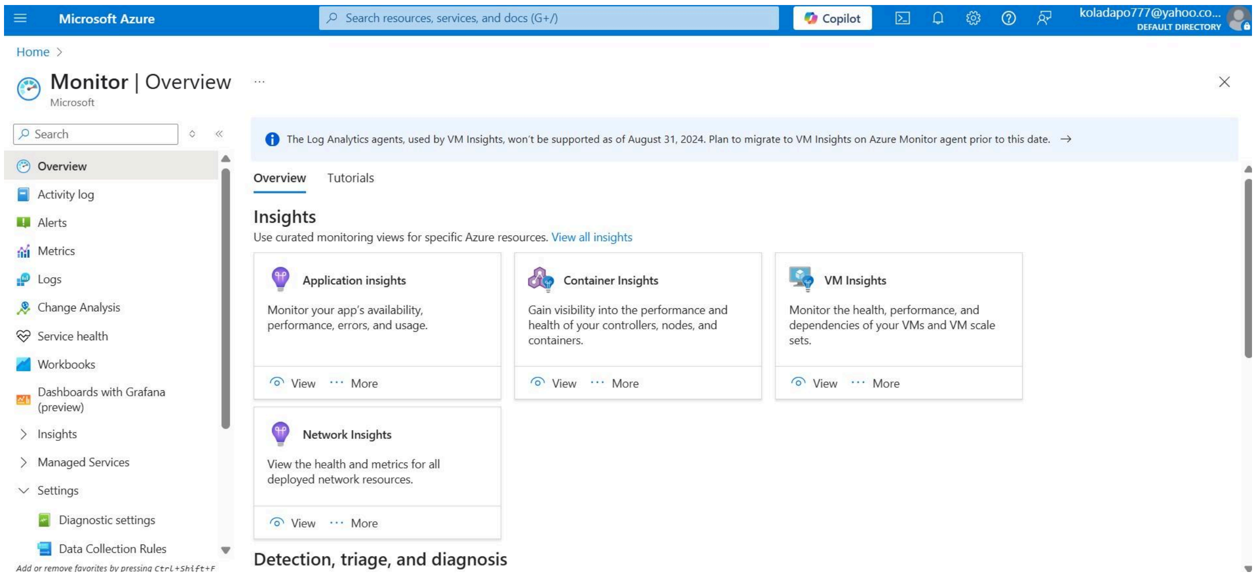
Project Report: Mastering Observability with Azure Monitor

Introduction: I undertook the task of implementing robust monitoring for my Azure Linux Virtual Machine (VM), `MonitorVM1`, leveraging Azure Monitor services. My goal was to gain insights into its performance and security events, ensuring proactive management and operational efficiency.

TASK 1- Access Azure Monitor

Azure Monitor Dashboard: I began by familiarising myself with the Azure Monitor dashboard, which provides a centralised view of all my monitoring data. This dashboard serves as my primary hub for observing the health and performance of my Azure resources.



The Log Analytics agents, used by VM Insights, won't be supported as of August 31, 2024. Plan to migrate to VM Insights on Azure Monitor agent prior to this date.

Insights

Use curated monitoring views for specific Azure resources. View all insights

Application insights: Monitor your app's availability, performance, errors, and usage.

Container Insights: Gain visibility into the performance and health of your controllers, nodes, and containers.

VM Insights: Monitor the health, performance, and dependencies of your VMs and VM scale sets.

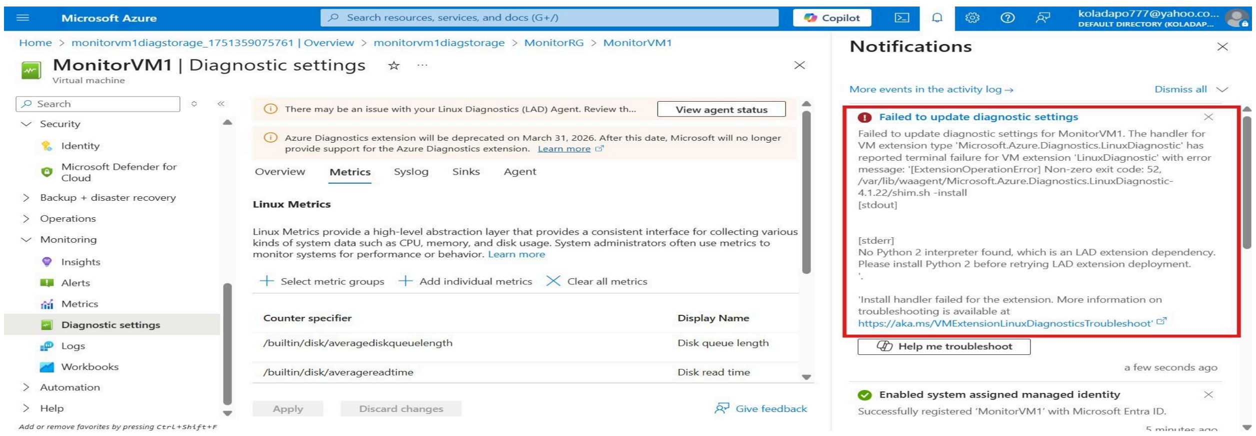
Network Insights: View the health and metrics for all deployed network resources.

Detection, triage, and diagnosis

Screenshot of Azure Monitor dashboard (General overview)

TASK 2 - Enable Diagnostics

Diagnostic Settings: To ensure comprehensive data collection for my VM, I configured diagnostic settings. This step was crucial for directing various types of logs and metrics from the Azure platform level to my Log Analytics Workspace, allowing for detailed analysis of the VM's operational health and activity. Unfortunately, I was unable to configure diagnostic settings after trying several times with consistent failed attempts. I then noticed the notification on top of the page, which warns: “Azure Diagnostics extension will be deprecated on March 31, 2026.” Also, another warning: “There may be an issue with the LAD agent”.



There may be an issue with your Linux Diagnostics (LAD) Agent. Review th...

Azure Diagnostics extension will be deprecated on March 31, 2026. After this date, Microsoft will no longer provide support for the Azure Diagnostics extension. Learn more

Linux Metrics

Linux Metrics provide a high-level abstraction layer that provides a consistent interface for collecting various kinds of system data such as CPU, memory, and disk usage. System administrators often use metrics to monitor systems for performance or behavior. Learn more

Counter specifier	Display Name
/builtin/disk/averagediskqueuelength	Disk queue length
/builtin/disk/averagereadtime	Disk read time

Failed to update diagnostic settings

```
Failed to update diagnostic settings for MonitorVM1. The handler for
VM extension type 'Microsoft.Azure.Diagnostics.LinuxDiagnostic' has
reported terminal failure for VM extension 'LinuxDiagnostic' with error
message: '[ExtensionOperationError] Non-zero exit code: 52,
/var/lib/waagent/Microsoft.Azure.Diagnostics.LinuxDiagnostic-
4.1.22/shim.sh -install
[stdout]


[stderr]
No Python 2 interpreter found, which is an LAD extension dependency.
Please install Python 2 before retrying LAD extension deployment.
'.

'Install handler failed for the extension. More information on
troubleshooting is available at
https://aka.ms/VMExtensionLinuxDiagnosticsTroubleshoot'
```

a few seconds ago

Enabled system assigned managed identity

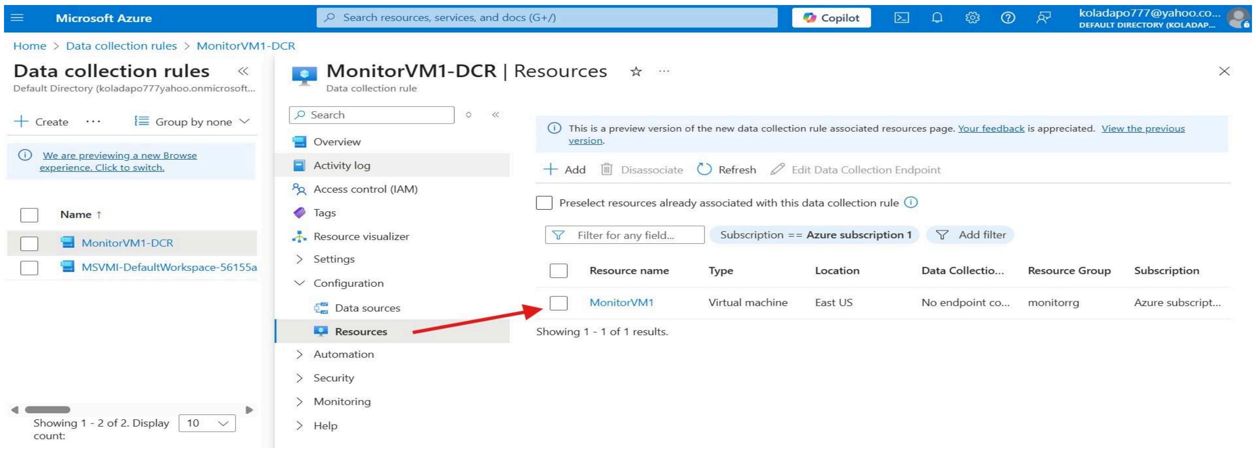
Successfully registered 'MonitorVM1' with Microsoft Entra ID.

5 minutes ago

A screenshot showing diagnostic settings warnings and a failed update

This is why Azure Monitor Agent (AMA) was Used. In my monitoring setup, I strategically chose to deploy the **Azure Monitor Agent (AMA)** on my Linux Virtual Machine for data collection, rather than the older Linux Diagnostic Extension (LAD). This decision was based on several key advantages offered by AMA.

AMA represents the modern and unified approach to monitoring in Azure. It utilises Data Collection Rules (DCRs) to provide granular control over what specific data is collected from the VM's guest operating system and where that data is sent. This allows for a highly customised and efficient data collection pipeline. For instance, I configured the DCR to collect both `Microsoft-Perf` data for performance metrics and `Linux Syslog` data for system and security logs, and it then sends the data collected to the Azure Monitor Logs destination.


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
koladapo777@yahoo.co...
DEFAULT DIRECTORY (KOLADAP...
Home > Data collection rules > MonitorVM1-DCR
Data collection rules
Default Directory (koladapo777yahoo.onmicrosoft...
Create
Group by none
We are previewing a new Browse experience. Click to switch.
Name ↑
MonitorVM1-DCR
MSVMI-DefaultWorkspace-56155a
Showing 1 - 2 of 2. Display count: 10
MonitorVM1-DCR | Resources
Data collection rule
Search
Overview
Activity log
Access control (IAM)
Tags
Resource visualizer
Settings
Configuration
Data sources
Resources
Automation
Security
Monitoring
Help
Add
Disassociate
Refresh
Edit Data Collection Endpoint
Filter for any field...
Subscription == Azure subscription 1
Add filter

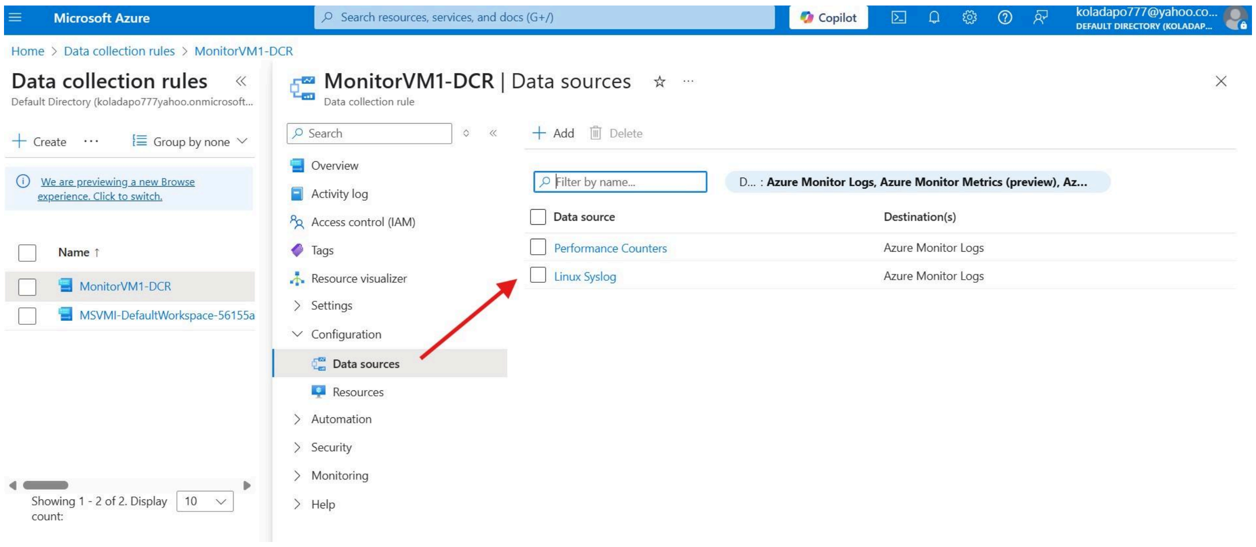
This is a preview version of the new data collection rule associated resources page. Your feedback is appreciated. View the previous version.

☐ Preselect resources already associated with this data collection rule

Resource name	Type	Location	Data Collectio...	Resource Group	Subscription
MonitorVM1	Virtual machine	East US	No endpoint co...	monitorrg	Azure subscript...

Showing 1 - 1 of 1 results.

Screenshot of DCR associated with the 'MonitorVM1'


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
koladapo777@yahoo.co...
DEFAULT DIRECTORY (KOLADAP...
Home > Data collection rules > MonitorVM1-DCR
Data collection rules
Default Directory (koladapo777yahoo.onmicrosoft...
Create
Group by none
We are previewing a new Browse experience. Click to switch.
Name ↑
MonitorVM1-DCR
MSVMI-DefaultWorkspace-56155a
Showing 1 - 2 of 2. Display count: 10
MonitorVM1-DCR | Data sources
Data collection rule
Search
Overview
Activity log
Access control (IAM)
Tags
Resource visualizer
Settings
Configuration
Data sources
Resources
Automation
Security
Monitoring
Help
Add
Delete
Filter by name...
D... : Azure Monitor Logs, Azure Monitor Metrics (preview), Az...

Data source	Destination(s)
Performance Counters	Azure Monitor Logs
Linux Syslog	Azure Monitor Logs

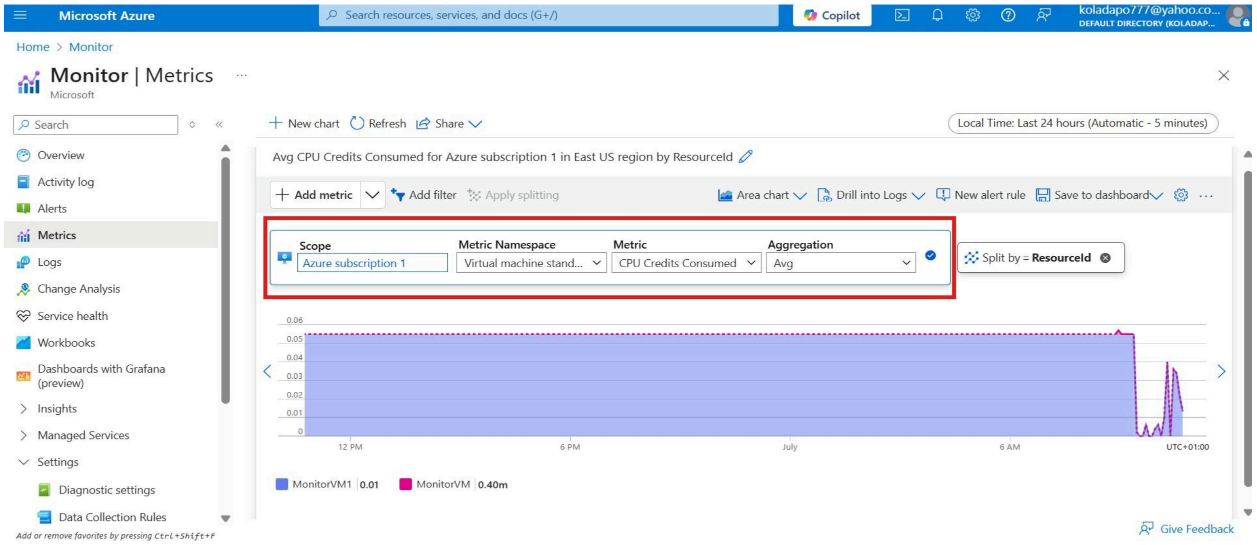
Screenshot of DCR Data Sources: ‘performance counters’ and ‘Linux Syslog’

In contrast, the Linux Diagnostic Extension (LAD), while functional, is an older solution. One significant consideration is LAD's reliance on a Python dependency for certain functionalities, which can introduce potential compatibility challenges and additional maintenance overhead in managing Python environments. AMA, on the other hand, is designed for broader compatibility and a more streamlined deployment experience, without such specific external dependencies. AMA also offers enhanced security features and improved performance, making it the recommended choice for new Azure monitoring deployments and ensuring my setup aligns with current best practices. The successful provisioning of AMA on my VM was a key factor in achieving reliable data flow.

TASK 3 - Analyse Data

1. Metrics Explorer

Visualised Metric: A key aspect of performance monitoring is visualising crucial metrics to quickly identify trends and anomalies. I focused on CPU utilisation, a fundamental indicator of VM performance. By collecting and querying the performance data, I was able to generate a clear visualisation of my VM's processor time over a specific period. This graphical representation allows for immediate assessment of the VM's workload.



Avg CPU Credits Consumed for Azure subscription 1 in East US region by ResourceId

Scope	Metric Namespace	Metric	Aggregation
Azure subscription 1	Virtual machine stand...	CPU Credits Consumed	Avg

Split by = ResourceId

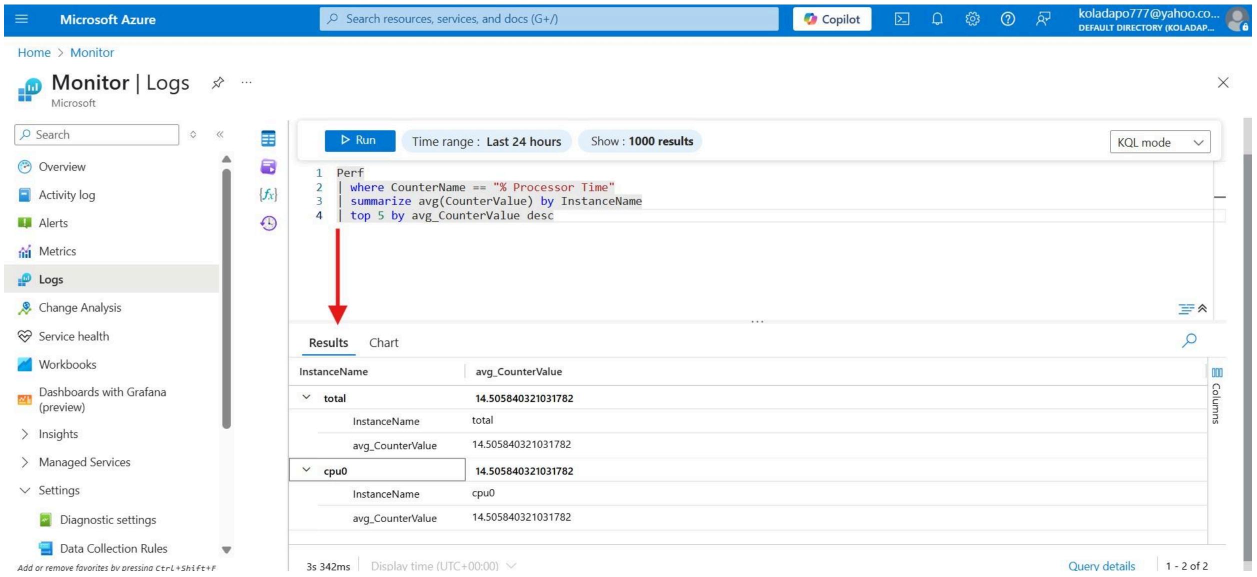
MonitorVM1 0.01
MonitorVM 0.40m

Screenshot of visualised metric, specifically a chart showing CPU utilisation over time, derived from `Perf` data in Log Analytics.

2. Log Analytics (KQL Queries)

Log Analytics Query and Results: To gain deeper insights from the collected data, I utilised Log Analytics and executed specific KQL (Kusto Query Language) queries.

Firstly, I successfully queried for performance data, specifically focusing on processor time to understand CPU consumption on my VM. The results helped me confirm that performance metrics were being correctly collected and analysed.



Time range : Last 24 hours
Show : 1000 results

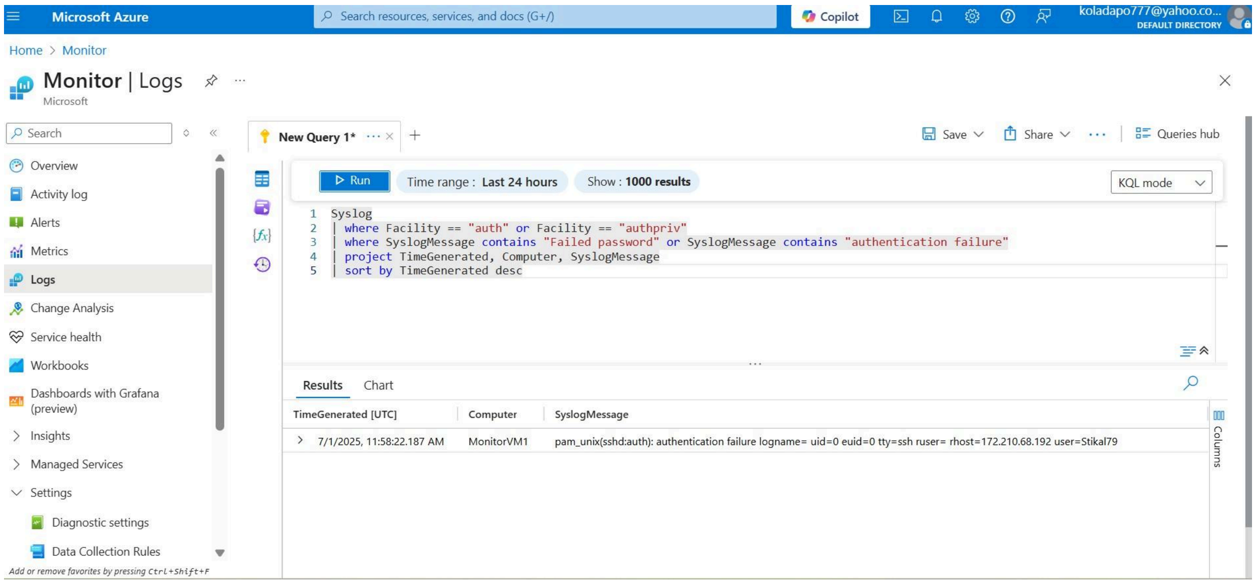
```
Perf
| where CounterName == "% Processor Time"
| summarize avg(CounterValue) by InstanceName
| top 5 by avg_CounterValue desc
```

InstanceName	avg_CounterValue
total	14.505840321031782
InstanceName	total
avg_CounterValue	14.505840321031782
cpu0	14.505840321031782
InstanceName	cpu0
avg_CounterValue	14.505840321031782

3s 342ms | Display time (UTC+00:00) | Query details | 1 - 2 of 2

(Part 1) A Screenshot of the KQL query and results for CPU performance of Log Analytics

Secondly, for security auditing and to monitor for unauthorised access attempts, I executed a `Syslog` query to identify failed login attempts. This demonstrated the successful collection of Linux system logs and their availability for security analysis.



```
Syslog
| where Facility == "auth" or Facility == "authpriv"
| where SyslogMessage contains "Failed password" or SyslogMessage contains "authentication failure"
| project TimeGenerated, Computer, SyslogMessage
| sort by TimeGenerated desc
```

TimeGenerated [UTC]	Computer	SyslogMessage
7/1/2025, 11:58:22.187 AM	MonitorVM1	pam_unix(sshd:auth): authentication failure logname= uid=0 euid=0 tty=ssh ruser= rhost=172.210.68.192 user=Stikal79

(Part 2) A screenshot of the KQL query for failed login attempts in Log Analytics

Deep Insight into Real-world Applications and Use Cases of Findings

The capabilities demonstrated in this project have direct and significant real-world applications:

- **Performance Optimisation:** Continuous monitoring of CPU utilisation allows for proactive identification of performance bottlenecks, enabling me to scale resources up or out before they impact application performance or user experience.
- **Security Incident Detection:** Tracking failed login attempts via Syslog is a fundamental security practice. It helps detect brute-force attacks, unauthorised access attempts, and anomalous user behaviour, enabling rapid response to potential security breaches.
- **Capacity Planning:** Analysing historical performance trends, such as average CPU usage, provides valuable data for informed capacity planning, ensuring that infrastructure resources are adequately provisioned to meet current and future demands without over-provisioning.
- **Troubleshooting and Root Cause Analysis:** Having detailed `Perf` and `Syslog` data readily available in Log Analytics accelerates troubleshooting by providing a comprehensive timeline of events and performance metrics leading up to an incident, aiding in faster root cause analysis.

- **Compliance and Auditing:** The ability to collect, store, and query system logs is essential for meeting regulatory compliance requirements and for internal auditing purposes, providing an immutable record of system activities.

Conclusion Through these steps, I successfully implemented a robust monitoring solution for my Azure Linux VM, enabling comprehensive insights into its performance and security. The ability to collect and analyse `Perf` data for CPU utilisation and `Syslog` data for failed login attempts provides invaluable visibility, allowing me to maintain a secure and efficient operational environment.