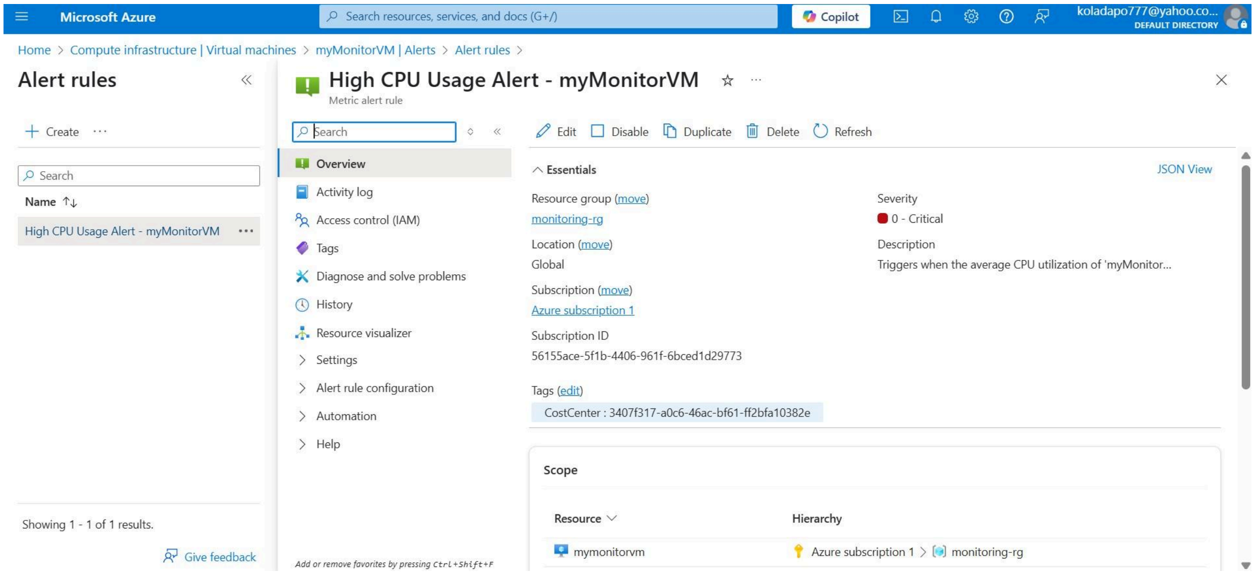
Project Report: Proactive Monitoring with Azure Monitor Alerts

This project focused on implementing proactive monitoring for an Azure Virtual Machine using Azure Monitor alerts. My primary objectives were to create a metric-based alert rule, configure multi-channel notifications, and validate the end-to-end alert workflow through controlled testing.

Steps Taken to Create and Configure Alerts

To begin, I ensured I had a running Azure Virtual Machine available in my subscription: “myMonitorVM”, which served as the target resource for monitoring. This VM was crucial as it provided the platform on which I would simulate performance issues to test my alert system.

My first key step was to **define an alert rule for resource metrics**. I navigated to my chosen Virtual Machine in the Azure portal and accessed the "Alerts" section to create a new alert rule. For the alert condition, I selected the **'Percentage CPU'** signal, as high CPU utilisation is a critical indicator of potential performance bottlenecks. I configured the alert to trigger when the **'Average' CPU** exceeded **'80%'** over a **'5 minute'** period, with the evaluation occurring **'Every 1 minute'**. This aggressive monitoring frequency ensures the timely detection of sustained high CPU. I named my alert rule `High CPU Usage Alert - myMonitorVM` and provided a clear description outlining its purpose. Crucially, I set the **Severity to '0 (Critical)'** to ensure immediate attention for such a significant performance indicator. I also made sure to **enable the 'Common alert schema'** for standardised alert payloads, which I recognised as a best practice for future integrations.



Resource group (move): monitoring-rg

Location (move): Global

Subscription (move): Azure subscription 1

Subscription ID: 56155ace-5f1b-4406-961f-6bced1d29773

Tags (edit): CostCenter : 3407f317-a0c6-46ac-bf61-ff2bfa10382e

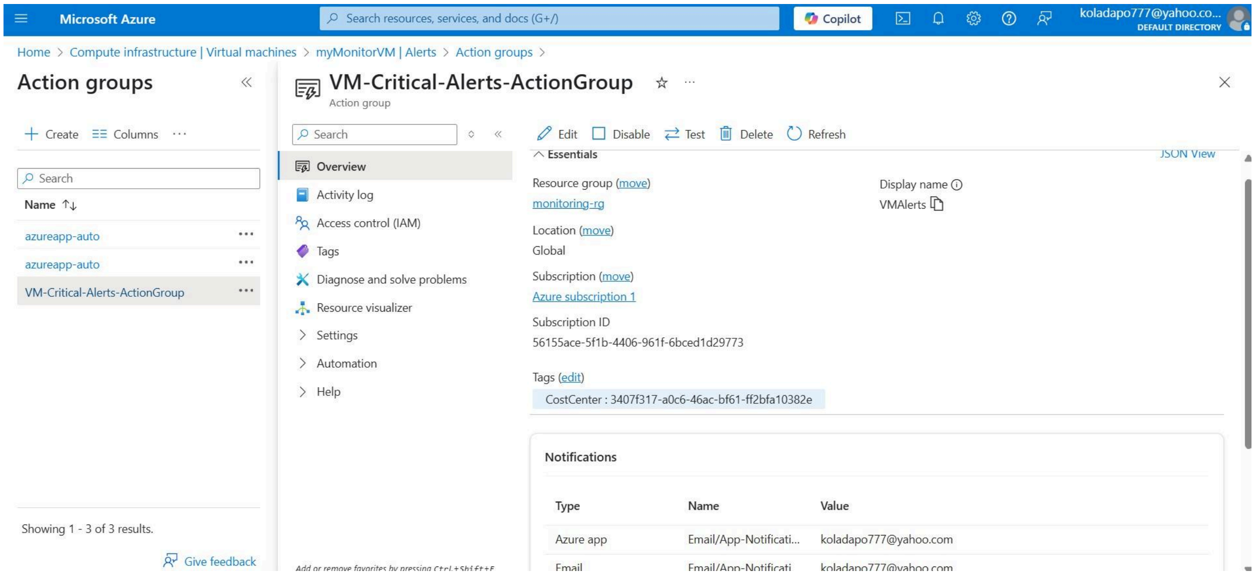
Severity: 0 - Critical

Description: Triggers when the average CPU utilization of 'myMonitor...

Resource	Hierarchy
mymonitorvm	Azure subscription 1 > monitoring-rg

Screenshot of your alert rule settings, showing the condition.

Next, I focused on **setting up an action group for notifications**. This component is vital for defining how I would be alerted when the CPU threshold is breached. Within the alert rule creation workflow, I created a new action group, naming it `VM-Critical-Alerts-ActionGroup` with the display name “VMAlerts”. For notifications, I configured multiple channels to ensure robust communication: I added my **email address**, enabled **Azure Mobile App Push notifications** for immediate alerts on my mobile device.


Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
koladapo777@yahoo.co...
DEFAULT DIRECTORY
Home > Compute infrastructure | Virtual machines > myMonitorVM | Alerts > Action groups >
Action groups
+ Create
Columns
Search
Name
azureapp-auto
azureapp-auto
VM-Critical-Alerts-ActionGroup
Showing 1 - 3 of 3 results.
Give feedback
VM-Critical-Alerts-ActionGroup
Action group
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Settings
Automation
Help
Add or remove favorites by pressing Ctrl+Shift+F
Edit
Disable
Test
Delete
Refresh
Essentials
JSON View

Resource group (move)
monitoring-rg

Location (move)
Global

Subscription (move)
Azure subscription 1

Subscription ID
56155ace-5f1b-4406-961f-6bced1d29773

Tags (edit)
CostCenter : 3407f317-a0c6-46ac-bf61-ff2bfa10382e

Display name
VMAlerts

Notifications

Type	Name	Value
Azure app	Email/App-Notificati...	koladapo777@yahoo.com
Email	Email/App-Notificati	koladapo777@yahoo.com

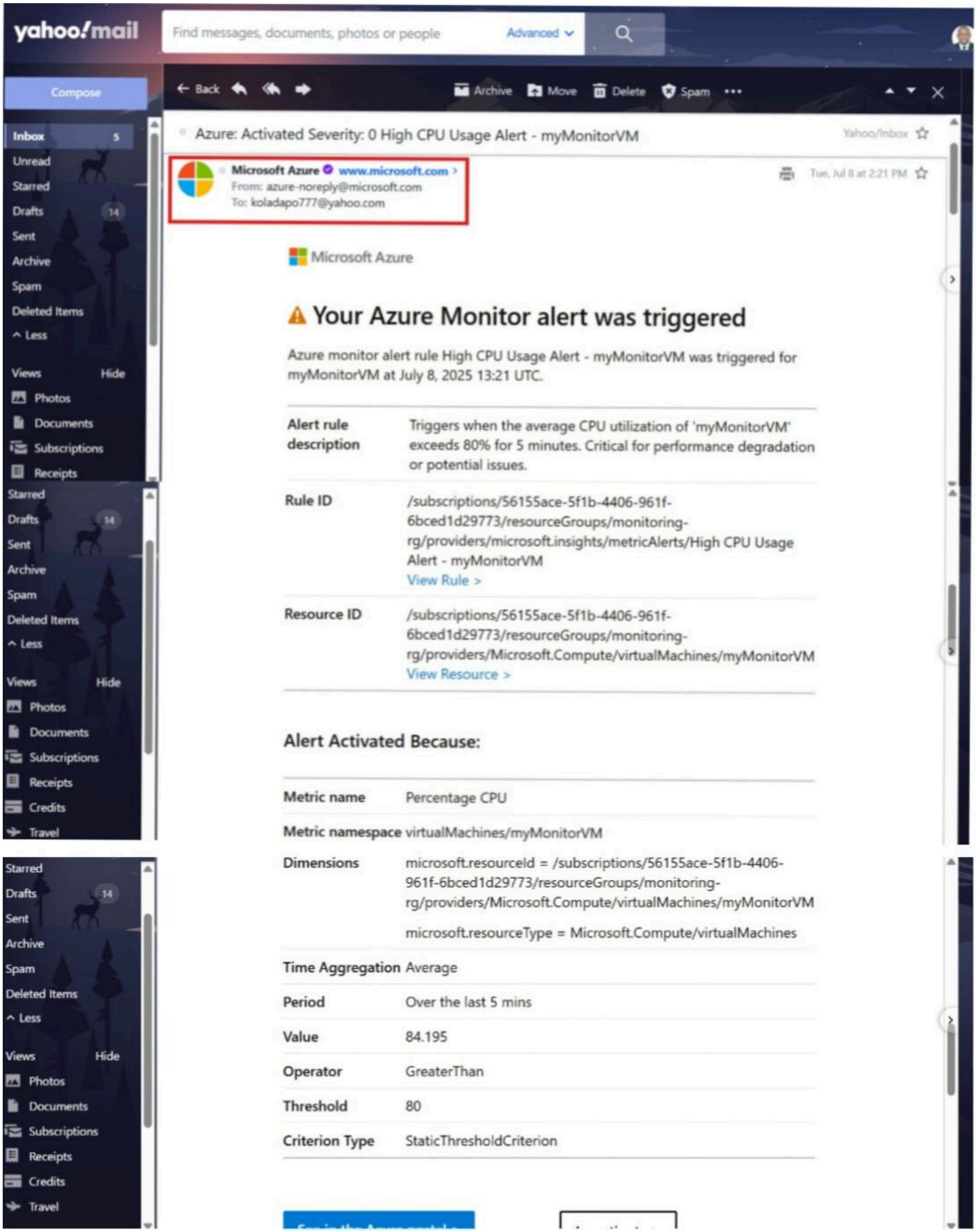
Screenshot of configured action group, with notification type of Email, Azure App.

Observations from Triggered Alerts

With the alert rule and action group fully configured, the final phase was to **trigger and verify the alerts**. I connected to my Azure Virtual Machine via SSH to intentionally simulate high CPU load.

- For my **Linux VM**, I used the `stress` tool with the command `stress -c 2 -t 300` to utilise 2 CPU cores for 5 minutes.

I allowed the stress tool to run for approximately 5-10 minutes, giving Azure Monitor sufficient time to evaluate the sustained high CPU usage. Soon after, I began receiving notifications exactly as configured. My inbox quickly showed an email with the customised subject line, `Azure: Activated Severity: High CPU Usage Alert-myMonitorVM`, clearly indicating the critical status and dynamic details. Simultaneously, I received a notification on my phone, and a push notification appeared on my Azure Mobile App.



Azure: Activated Severity: 0 High CPU Usage Alert - myMonitorVM

Microsoft Azure www.microsoft.com
From: azure-noreply@microsoft.com
To: koladapo777@yahoo.com

Tue, Jul 8 at 2:21 PM

Microsoft Azure

⚠ Your Azure Monitor alert was triggered

Azure monitor alert rule High CPU Usage Alert - myMonitorVM was triggered for myMonitorVM at July 8, 2025 13:21 UTC.

Alert rule description	Triggers when the average CPU utilization of 'myMonitorVM' exceeds 80% for 5 minutes. Critical for performance degradation or potential issues.
Rule ID	/subscriptions/56155ace-5f1b-4406-961f-6bced1d29773/resourceGroups/monitoring-rg/providers/microsoft.insights/metricAlerts/High CPU Usage Alert - myMonitorVM View Rule >
Resource ID	/subscriptions/56155ace-5f1b-4406-961f-6bced1d29773/resourceGroups/monitoring-rg/providers/Microsoft.Compute/virtualMachines/myMonitorVM View Resource >

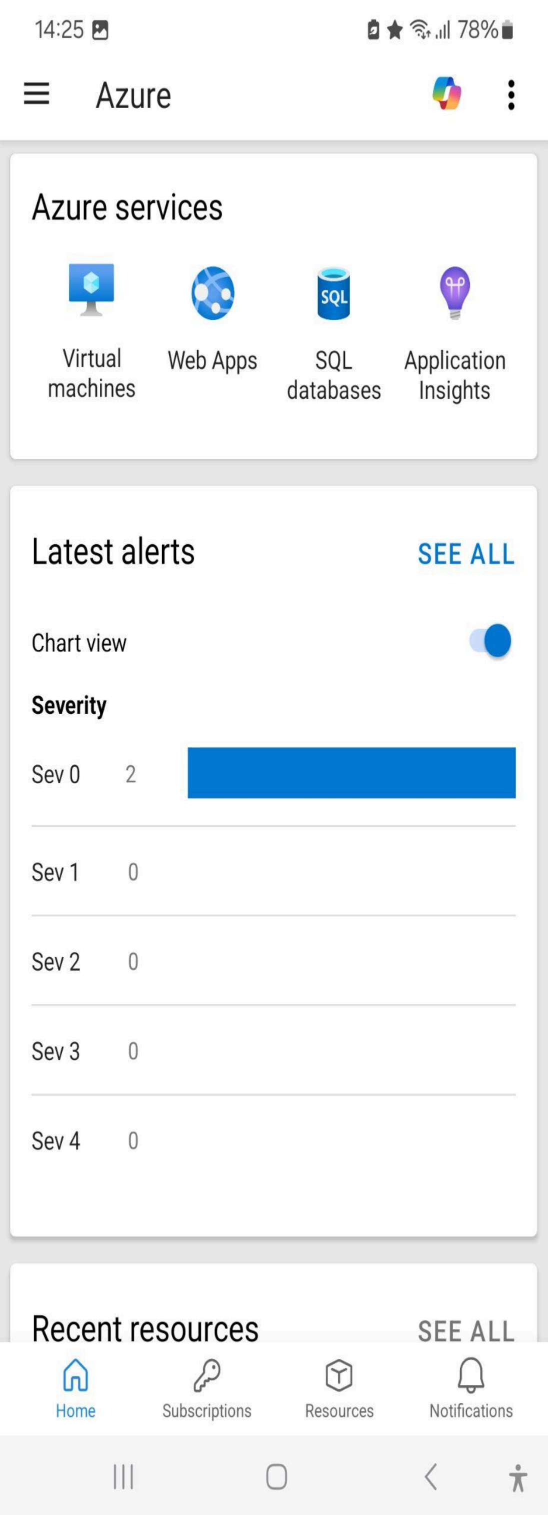
Alert Activated Because:

Metric name	Percentage CPU
Metric namespace	virtualMachines/myMonitorVM

Dimensions	microsoft.resourceId = /subscriptions/56155ace-5f1b-4406-961f-6bced1d29773/resourceGroups/monitoring-rg/providers/Microsoft.Compute/virtualMachines/myMonitorVM
	microsoft.resourceType = Microsoft.Compute/virtualMachines

Time Aggregation	Average
Period	Over the last 5 mins
Value	84.195
Operator	GreaterThan
Threshold	80
Criterion Type	StaticThresholdCriterion

Screenshot of received alert, Email notification



Azure services

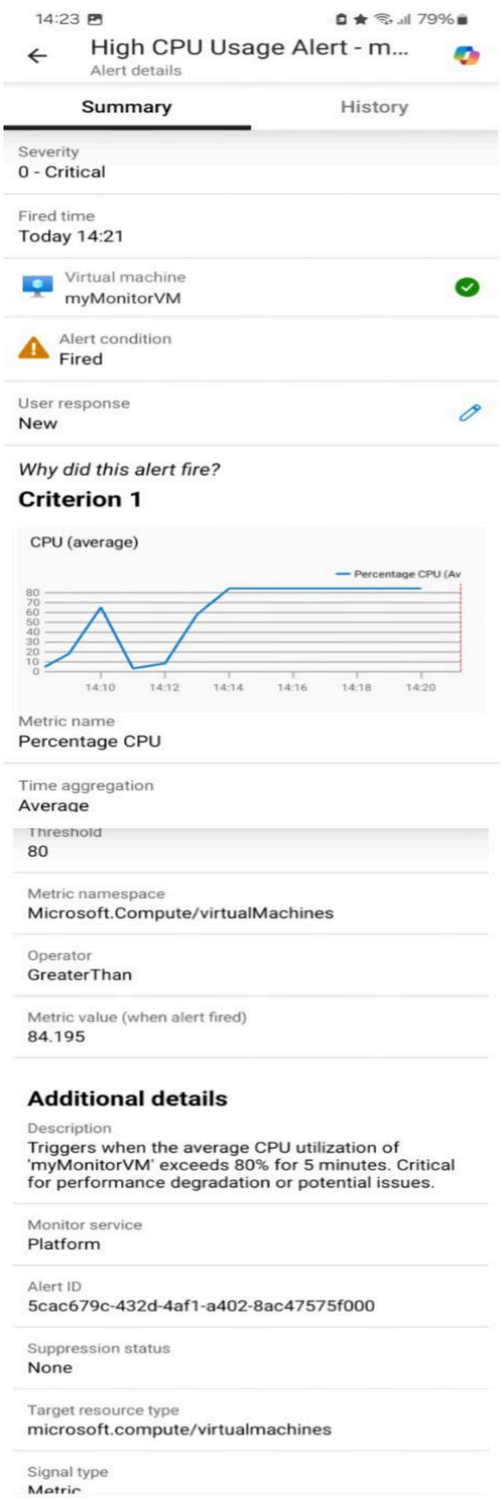
Latest alerts

Chart view

Severity

Sev 0	2	▇▇▇▇▇▇▇▇
Sev 1	0	
Sev 2	0	
Sev 3	0	
Sev 4	0	

Recent resources



Severity
0 - Critical

Fired time
Today 14:21

Virtual machine
myMonitorVM ✓

Alert condition
Fired

User response
New

Why did this alert fire?

Criterion 1

Metric name
Percentage CPU

Time aggregation
Average

Threshold
80

Metric namespace
Microsoft.Compute/virtualMachines

Operator
GreaterThan

Metric value (when alert fired)
84.195

Additional details

Description
Triggers when the average CPU utilization of 'myMonitorVM' exceeds 80% for 5 minutes. Critical for performance degradation or potential issues.

Monitor service
Platform

Alert ID
5cac679c-432d-4af1-a402-8ac47575f000

Suppression status
None

Target resource type
microsoft.compute/virtualmachines

Signal type
Metric

Screenshot of received alert, Azure Mobile App Notification

Once I verified the successful alert delivery, I promptly stopped the CPU stress tool on my VM. Observing the metrics in the Azure portal, I could see the CPU utilisation returning to normal. After a short period, the alert automatically resolved in Azure Monitor, demonstrating the full lifecycle of a proactive monitoring alert from trigger to resolution.

Troubleshooting Insights

During this project, the most critical aspect was ensuring that all configurations – from the alert conditions' thresholds and evaluation frequency to the action group's notification details – were precisely set. I found that patience was key when waiting for the alert to trigger; allowing enough time for the "average over 5 minutes" condition to be met was essential. Each step, from VM deployment to the final alert verification, reinforced the systematic approach required for robust cloud monitoring.

Deeper insight into its real-world applications and use cases:

- **Proactive Problem Resolution:** Beyond just CPU, similar alerts can monitor memory, disk I/O, network traffic, or application-specific metrics (e.g., failed requests, queue depth). This allows operations teams to identify and address performance bottlenecks or resource exhaustion *before* they lead to service outages or user impact, shifting from reactive firefighting to proactive problem-solving.
- **Service Level Agreement (SLA) Adherence:** By setting alerts on key performance indicators (KPIs) like application response times, error rates, or database connection pools, organisations can ensure their services meet defined SLAs. Alerts provide early warnings when these metrics approach critical thresholds, enabling timely intervention to prevent SLA breaches.
- **Cost Optimisation through Resource Management:** Alerts aren't just for problems. You can set alerts for *low* utilisation (e.g., CPU consistently below 10%). This identifies over-provisioned VMs or App Services, prompting right-sizing or scaling down, leading to significant cloud cost savings.
- **Enhanced Security Posture:** Alerts can be configured for unusual or suspicious activities, such as:
    - Spikes in failed login attempts.
    - Outbound network traffic from internal systems to known malicious IPs.
    - Unexpected changes to critical Azure resources (using Activity Log Alerts). This helps detect and respond to potential security breaches or misconfigurations.
- **Automated Incident Response:** Integrating action groups with webhooks, Azure Functions, or Logic Apps transforms alerts from mere notifications into automated responses. For example:
    - A critical CPU alert could automatically scale out the VMSS.
    - A security alert could automatically block a malicious IP address at the network security group (NSG) level.

    - Any alert can automatically create an incident ticket in an ITSM system (e.g., ServiceNow, Jira).
- **Capacity Planning & Trend Analysis:** By reviewing historical alert data and associated metrics, organisations can gain insights into resource usage patterns. This data is invaluable for predicting future capacity needs, planning infrastructure upgrades, and making informed decisions about scaling strategies.
- **Auditing and Compliance:** Alerts on specific configuration changes or resource deletions can help maintain compliance with internal policies and external regulations. For instance, alerting on changes to network security group rules or critical database access permissions.