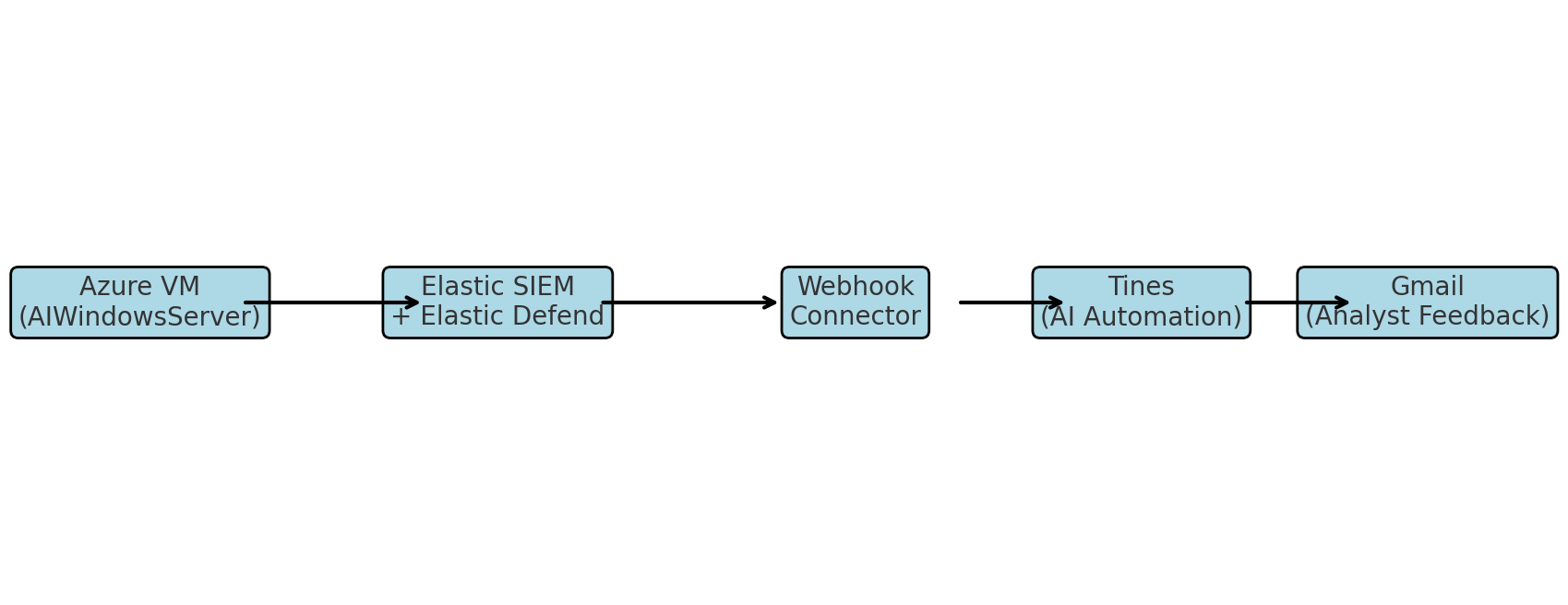
# SOC Project with AI-Driven Automation

This project demonstrates the integration of Elastic SIEM, Elastic Defend (EDR), and AI-driven automation to build a cloud-native SOC on Azure. The following flow diagram provides an overview of the architecture before walking through each implementation step in detail.  
  
POV: This project is an upgrade by adding an edge with AI, to existing SOC automation.   
You can check out my previous work here: <https://tejalbguttal1.wixsite.com/mysoclab>

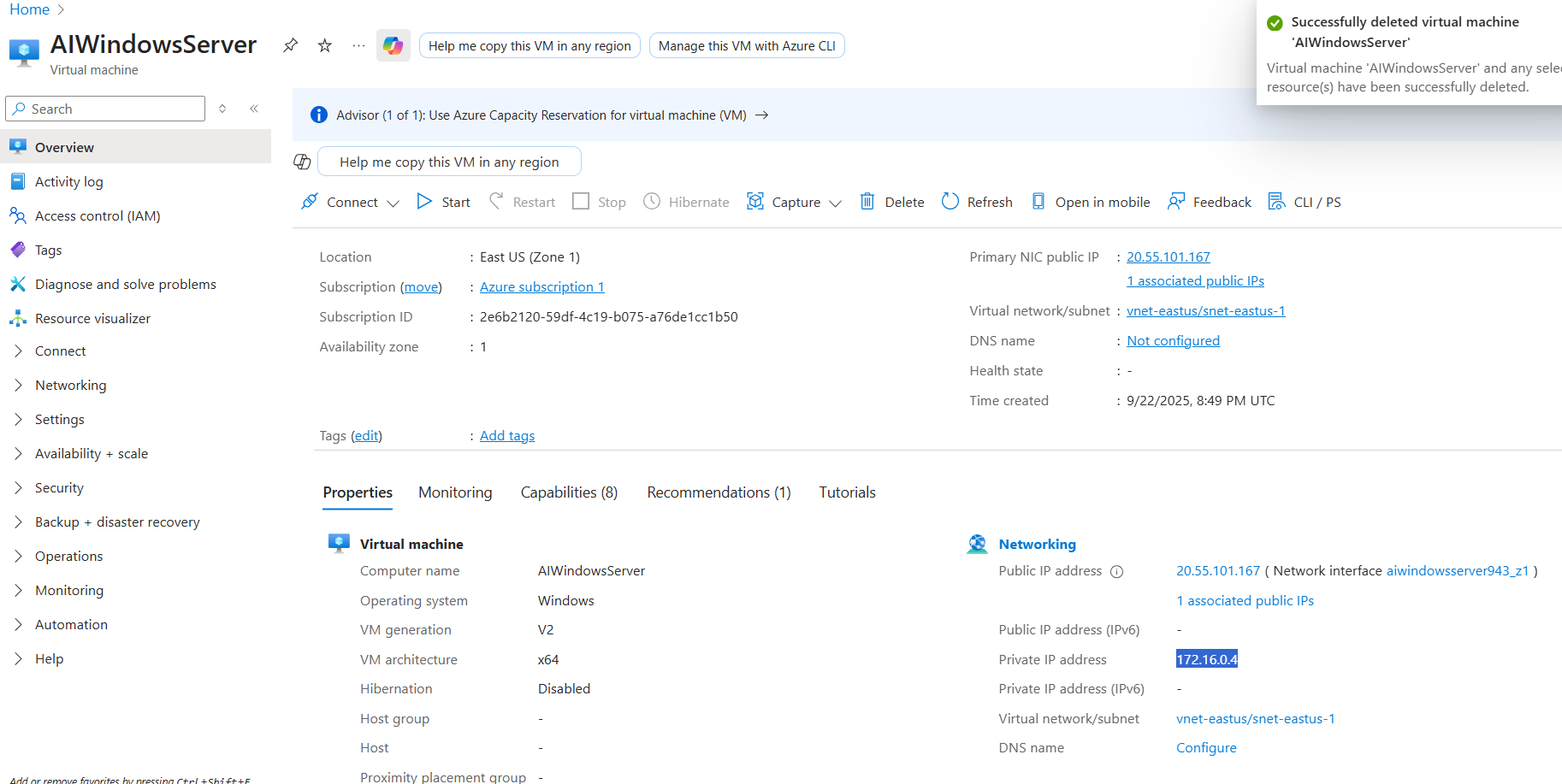
## Architecture Flow Diagram

The diagram below illustrates the flow of security events: from the Azure-hosted Windows VM to Elastic SIEM/EDR, then through the webhook connector into Tines for AI enrichment and finally delivered as actionable feedback via Gmail.

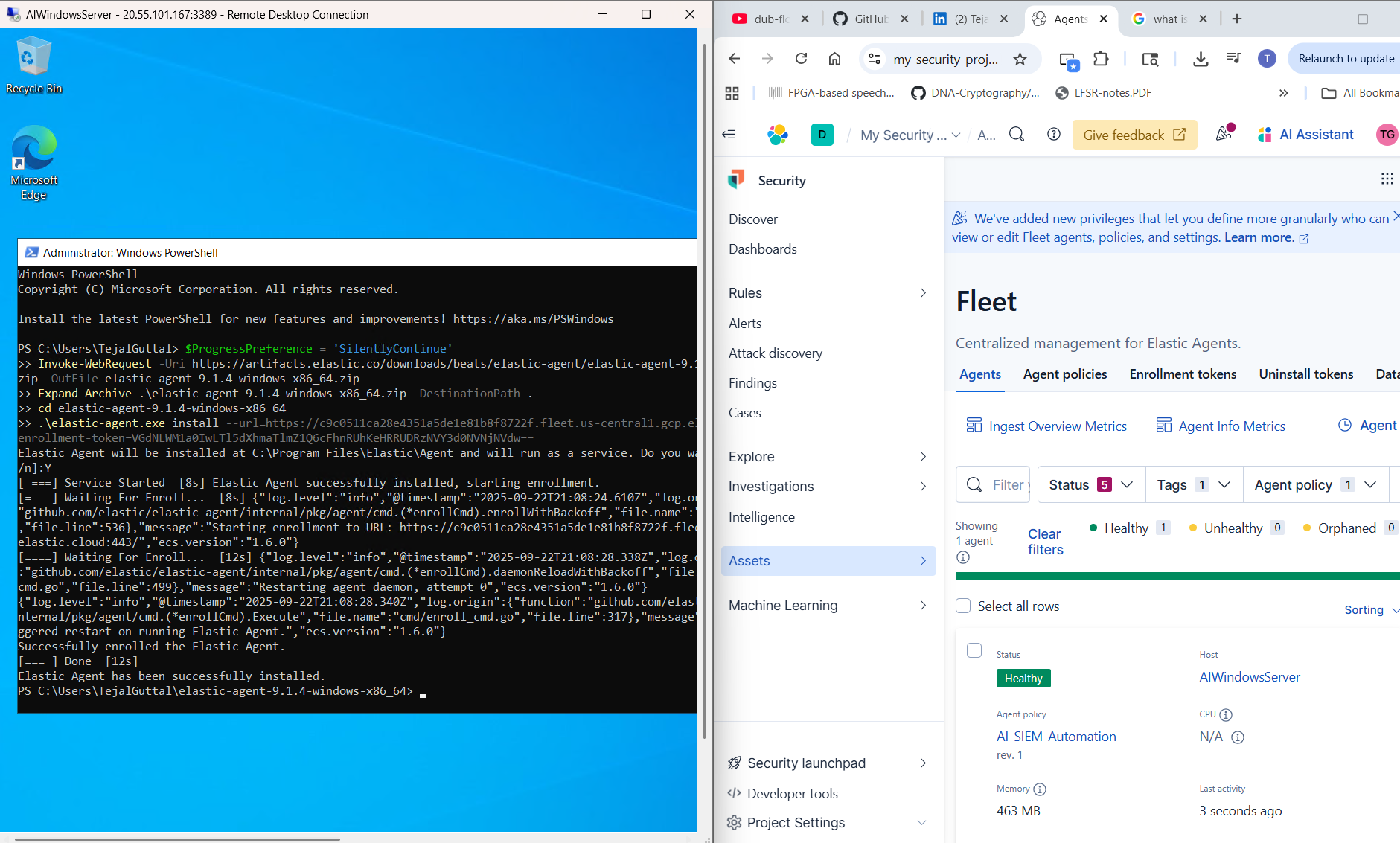


# Step-by-Step Walkthrough

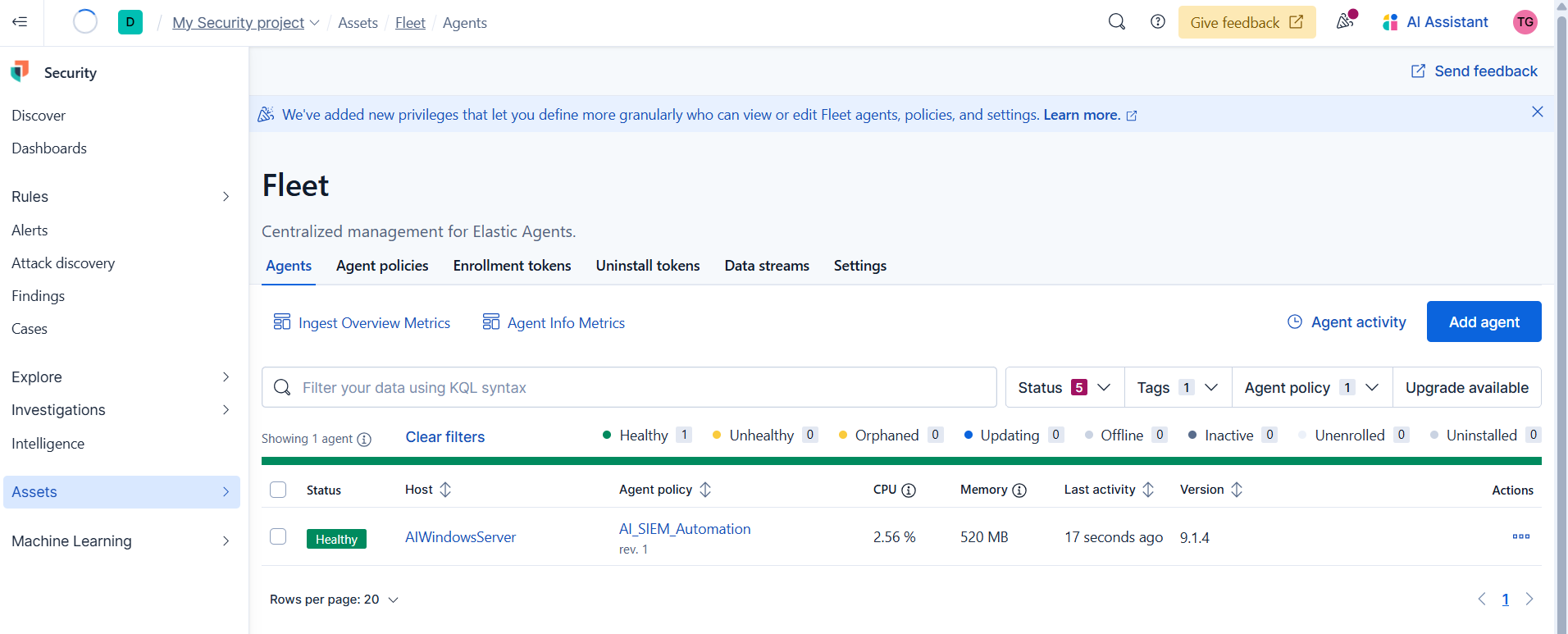
## Creating windows virtual machine

A Windows Server virtual machine is created in Azure (you may choose any cloud provider). This VM serves as the monitored endpoint for Elastic Agent deployment and security testing.  
Make sure your VM has a public IP address as we are connecting over the internet.  


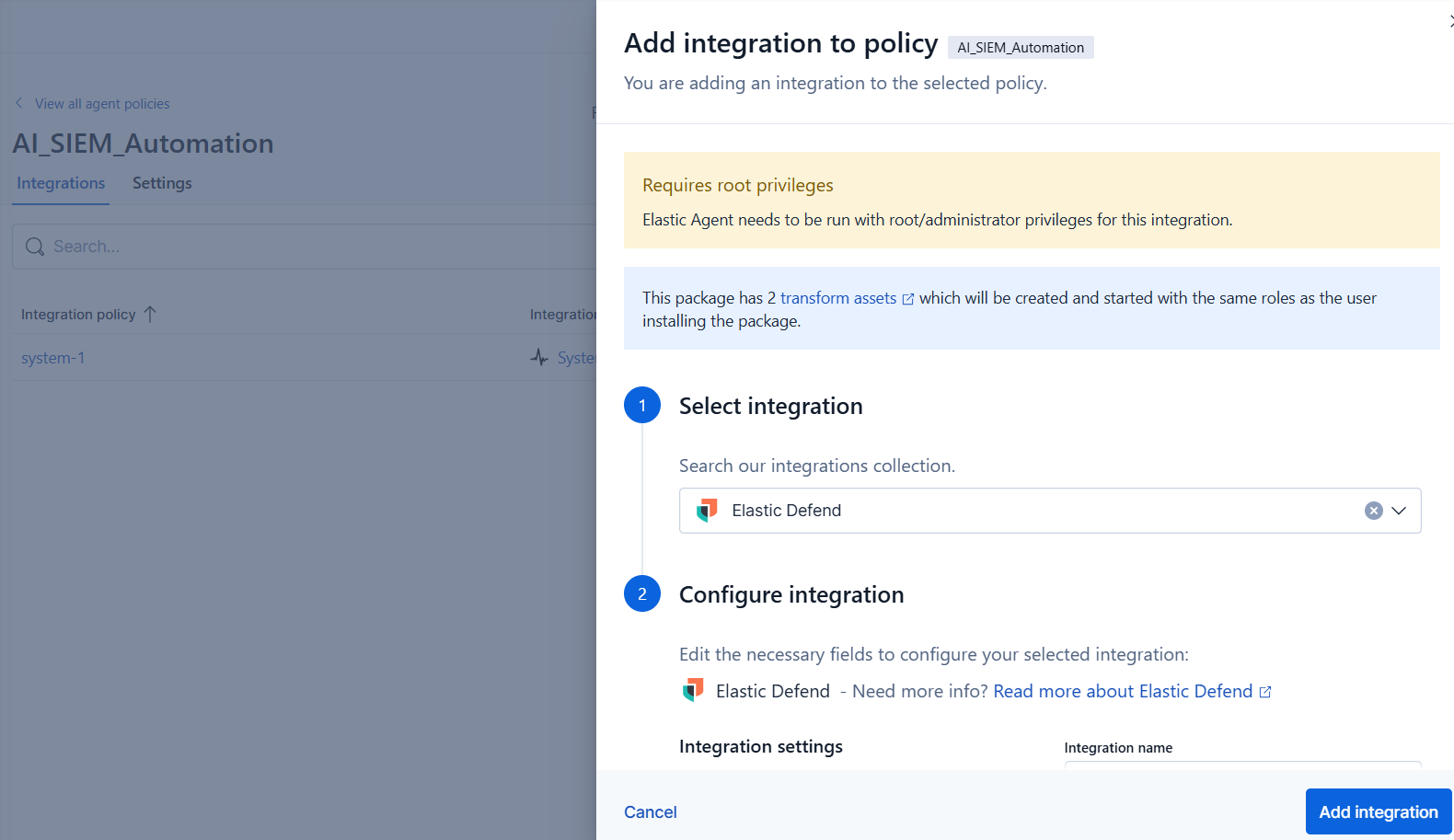
## Integrating Elastic Fleet for real time monitoring

Elastic Agent is installed on the Azure VM through PowerShell. Successful enrollment confirms the server is connected to Elastic Fleet for real-time monitoring.  


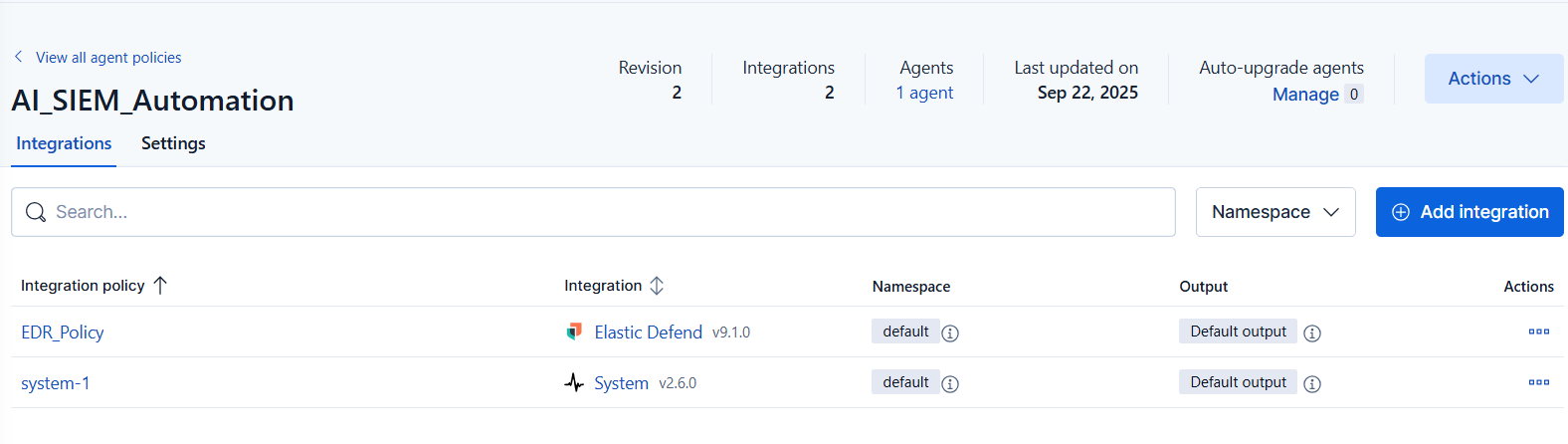
## VM Connection is established with Elastic Fleet

The Elastic Fleet dashboard shows the agent (AIWindowsServer) as Healthy, validating secure enrollment and continuous telemetry streaming to the SIEM.  


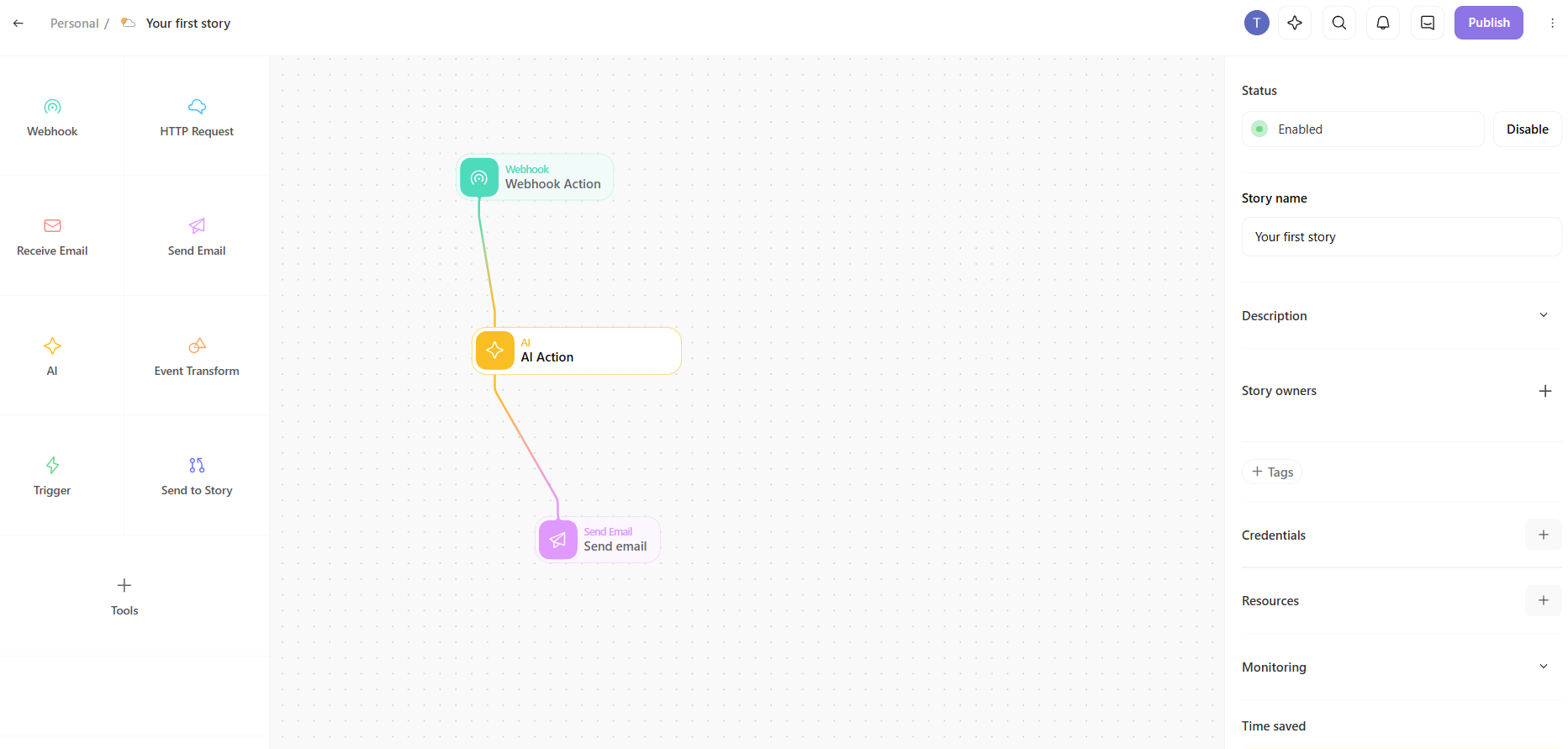
## Integrating Elastic Defend into the VM

Elastic Defend integration is added to the agent policy on our VM. This enables endpoint detection and response features, providing deeper visibility into host processes and threats.|  


## Policies are integrated

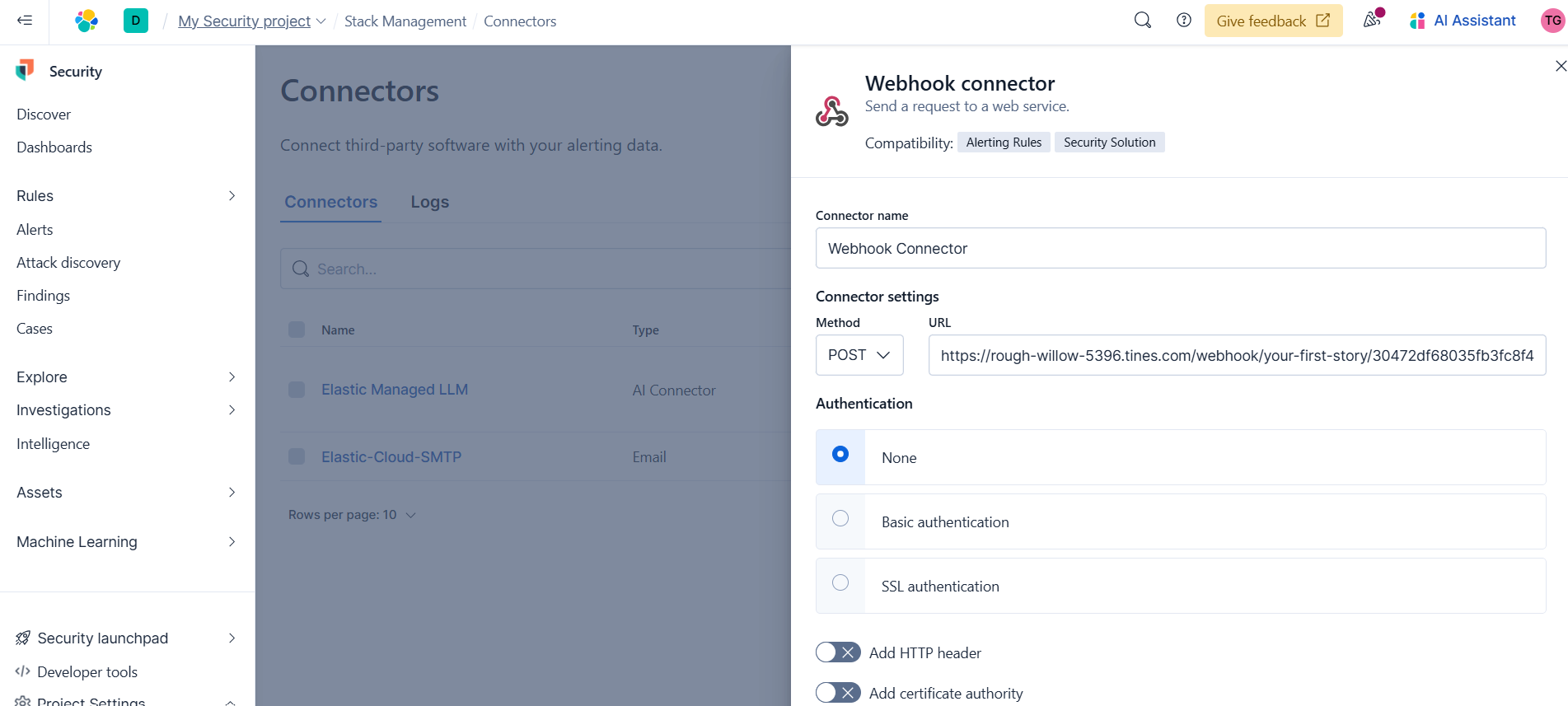
The agent policy now includes Elastic Defend (for EDR) and System (for telemetry). This combination ensures strong endpoint protection and host-level monitoring.  


## Tines – The no code automation platform

A Tines automation workflow is designed with a Webhook trigger, AI action, and email notification. This workflow applies AI to enrich alerts and forward results to analysts.  


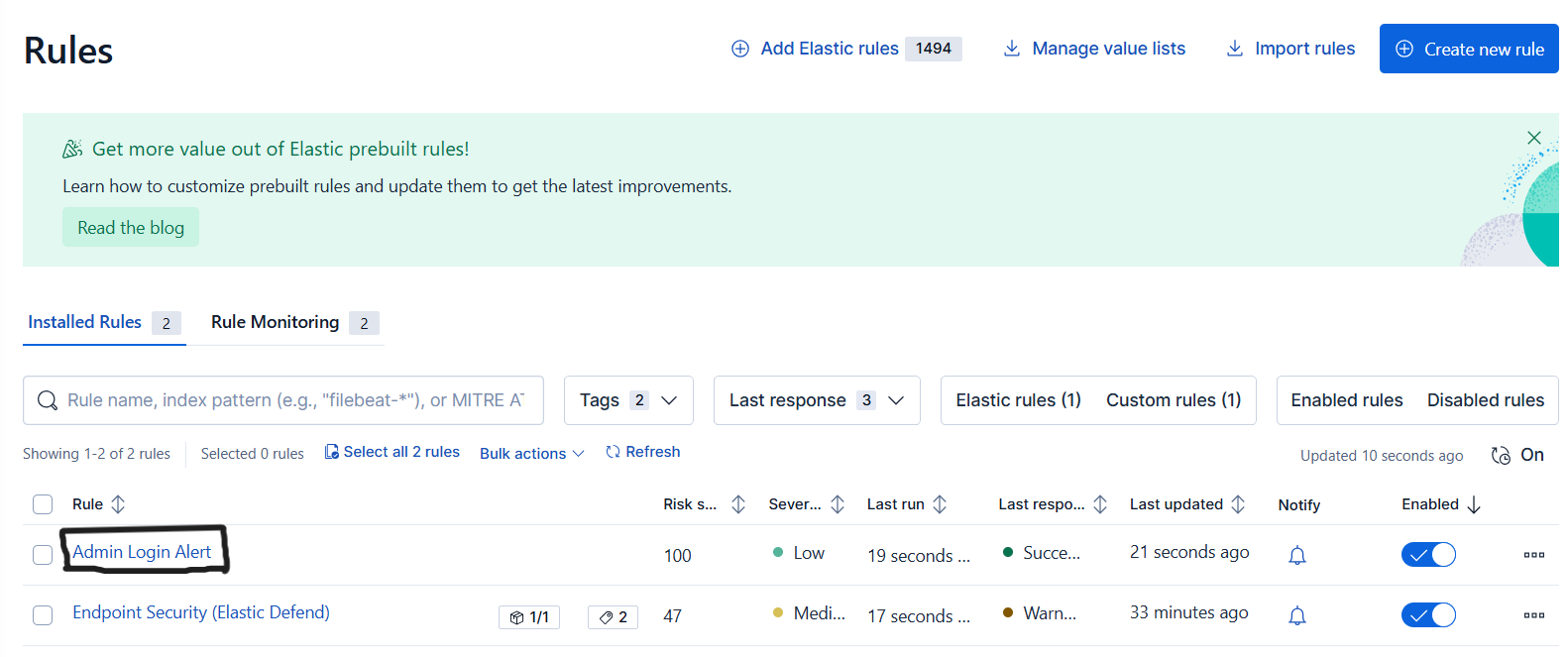
## 

## Web Hook Connector – The middleman!

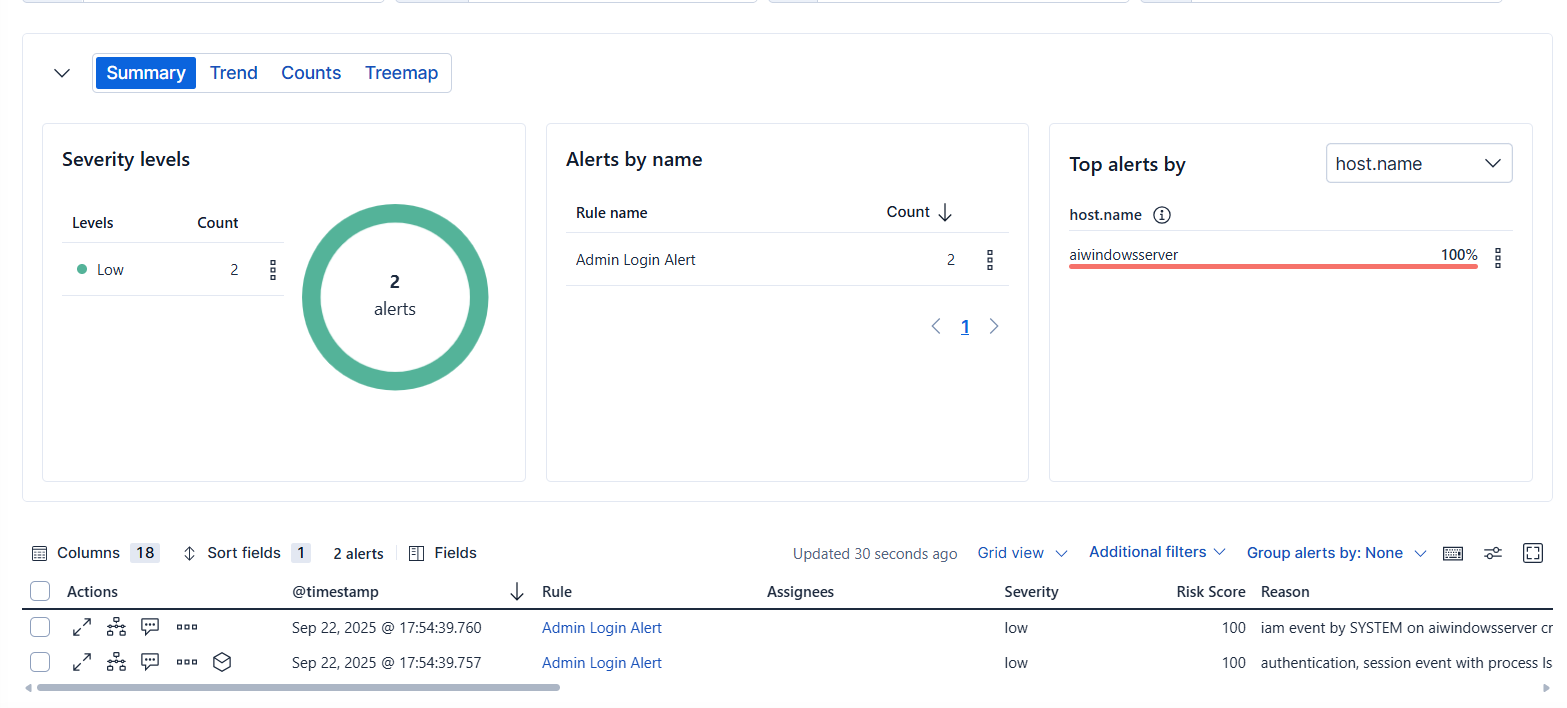
Elastic Security is connected to Tines via a Webhook connector. This bridges detection alerts from SIEM to the AI automation pipeline.  
The POST Uri is supposed to be mentioned in the custom rule which carries the communication further..  


## Creating a custom rule for triggering an event

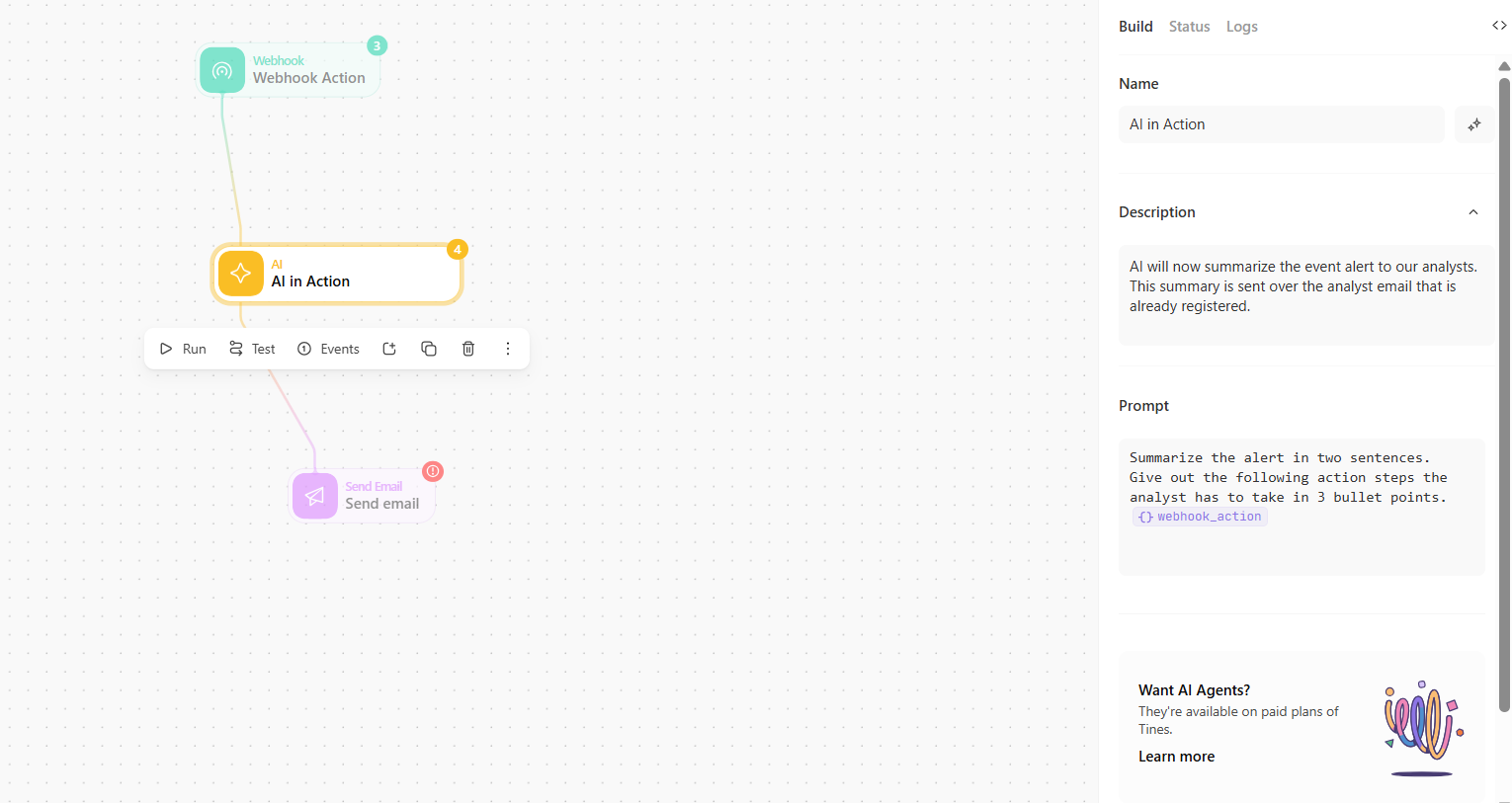
Custom alert rules are created in Elastic SIEM, such as Admin Login Alert. These rules ensure critical login events are detected and forwarded for analysis.



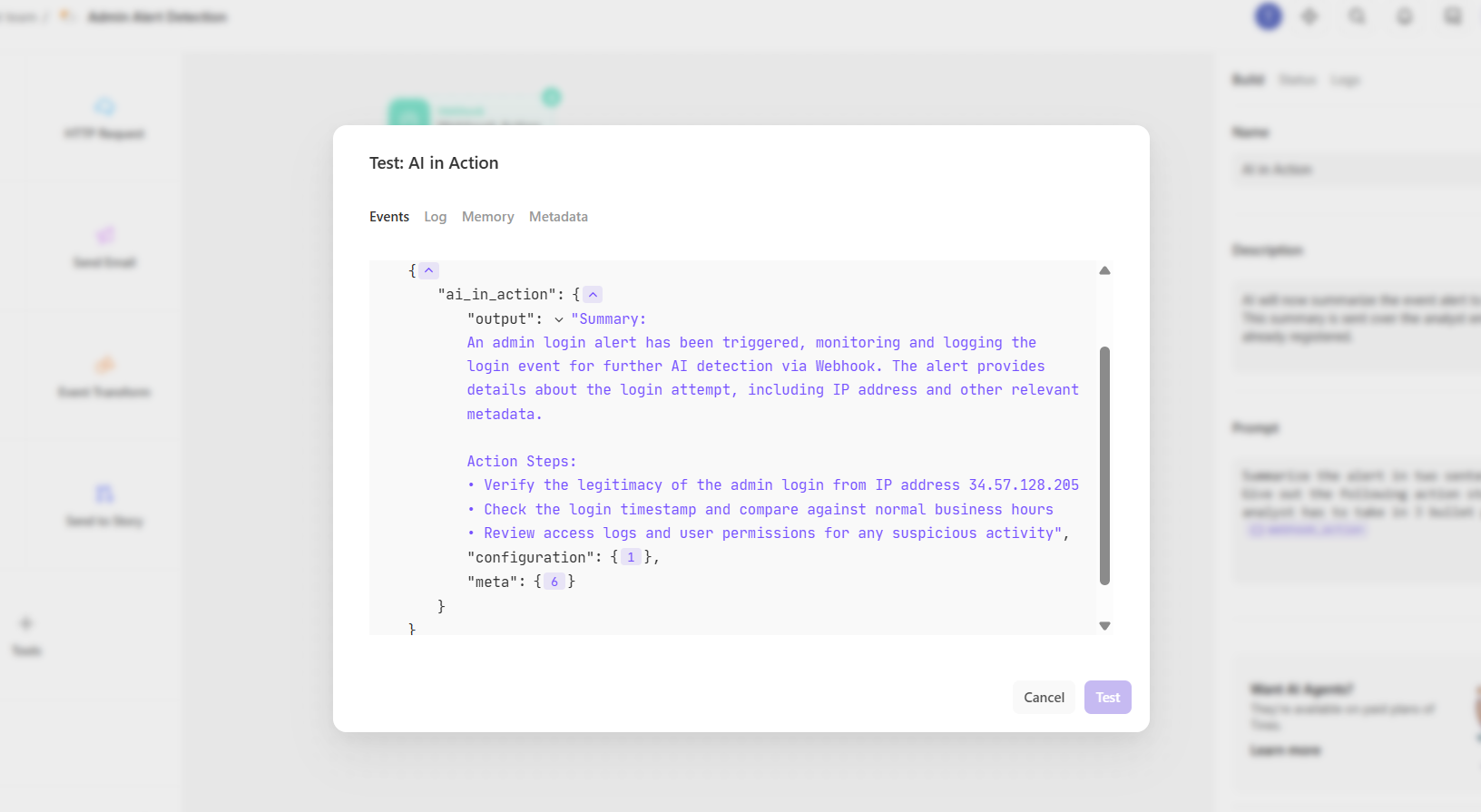
## Admin login alert is captured on SIEM

An Admin Login Alert is triggered and captured in the SIEM. It provides detailed information about login events on the AIWindowsServer, ensuring visibility into access attempts.  


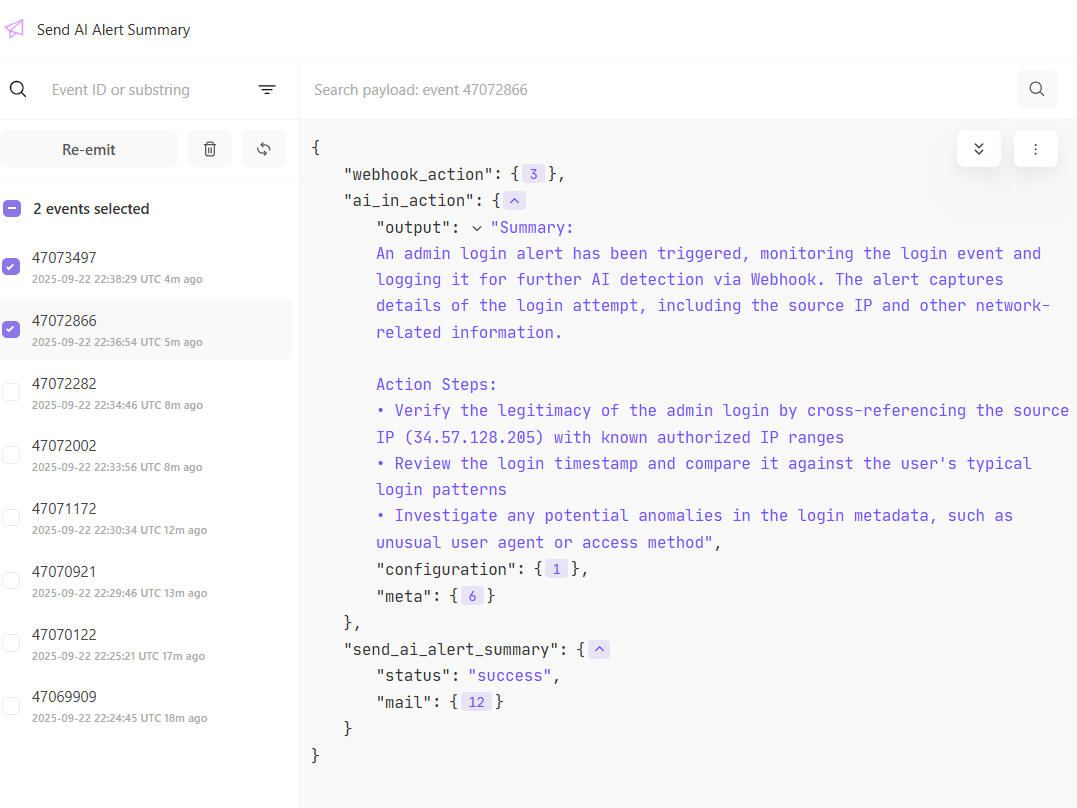
## Configuring AI’s prompt

Within Tines, an AI action is configured to summarize alerts and suggest action steps. The AI processes raw alert data and converts it into concise analyst instructions.  


## Sample of AI functioning

Testing the AI action shows its output: a summary of the alert and a list of action steps e.g., verifying login legitimacy, checking timestamps, reviewing logs.  


## Mail recieved by analysts!

The enriched AI-generated alert summary is successfully delivered to analysts via email. This closes the loop, ensuring actionable insights reach analysts in real time.  
  
  
  
Thank you for viewing my page!