

Integrating AlertOps with ScienceLogic SL1

Introduction

This document describes how to integrate AlertOps with ScienceLogic SL1 to automatically forward events and streamline incident management. By following the steps in this guide, you can configure SL1 to send detailed alert payloads to AlertOps, enabling centralized alert handling and response.

This section covers the following topics:

- What is AlertOps?
 - What Does the AlertOps Integration Do?
 - Installing and Configuring the AlertOps Integration
 - **Experience: Using the Integration**
-

What is AlertOps?

AlertOps is an incident management and alerting platform designed to help organizations streamline their response to IT incidents and improve communication among teams. It centralizes alerts from various monitoring systems, automates workflows, and reduces alert fatigue, ultimately decreasing the mean time to resolution (MTTR) for critical issues.

What Does the AlertOps Integration Do?

The AlertOps integration for ScienceLogic SL1 uses a PowerPack to enable seamless communication between the two platforms.

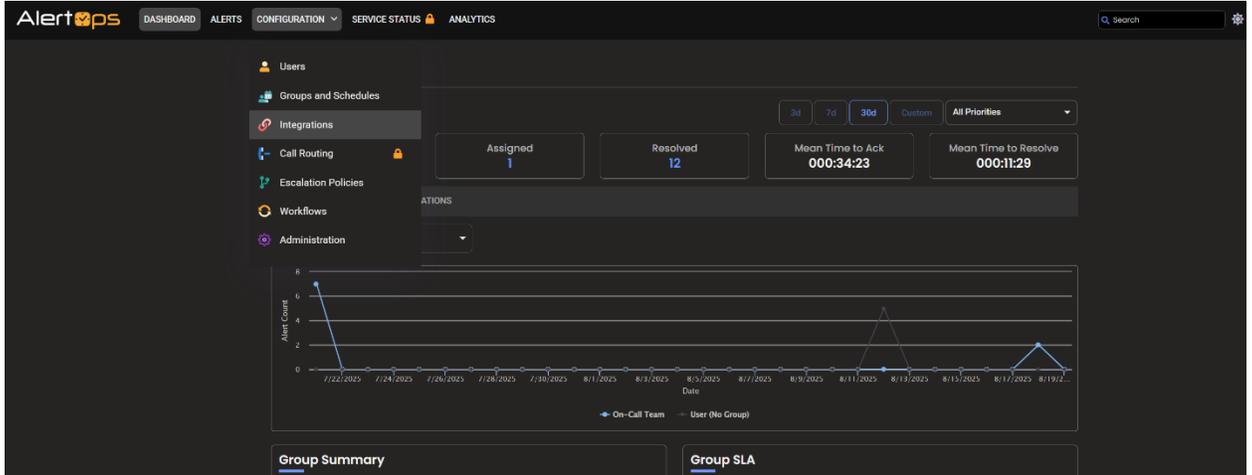
Features include:

- **Runbook Actions:** Two Python-based snippets that post alert payloads to the AlertOps API.
 - **Runbook Policies:** Two pre-configured policies to trigger the Runbook Actions for both critical events and event clearances.
 - **Custom API Integration:** A dedicated inbound integration in AlertOps to receive and process alerts from SL1.
-

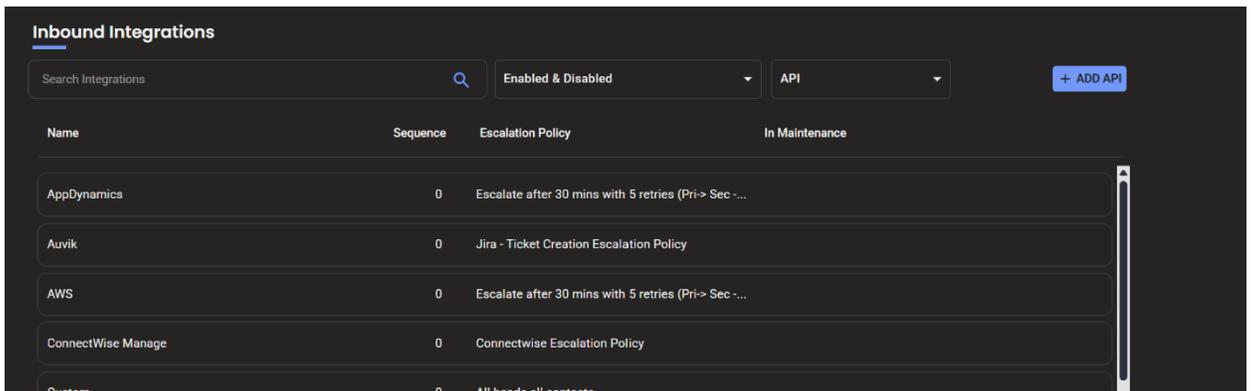
Installing and Configuring the AlertOps Integration

1. Configure the Custom API Integration in AlertOps

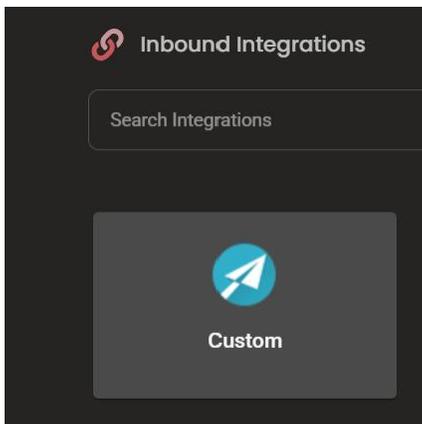
1. In AlertOps, navigate to **Configuration > Integrations**.



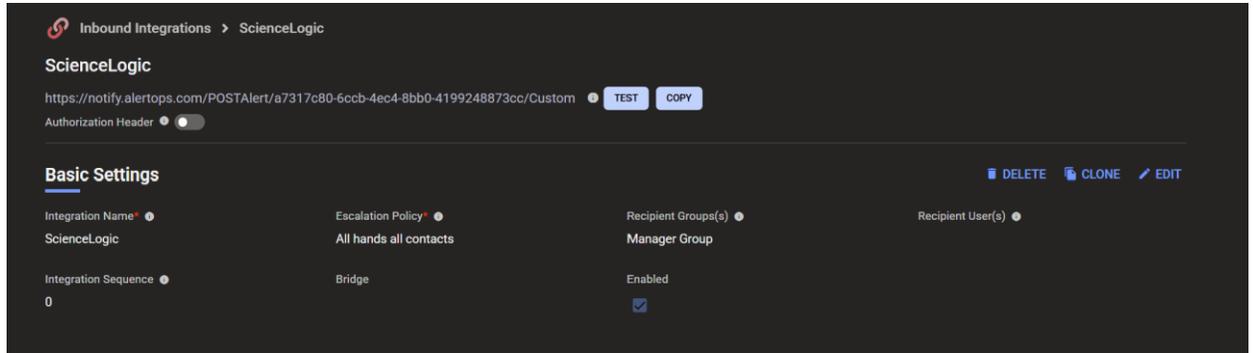
2. Select **Inbound Integrations**, then click **API** and choose **Add API**.



3. Create a **Custom API Integration** for SL1.

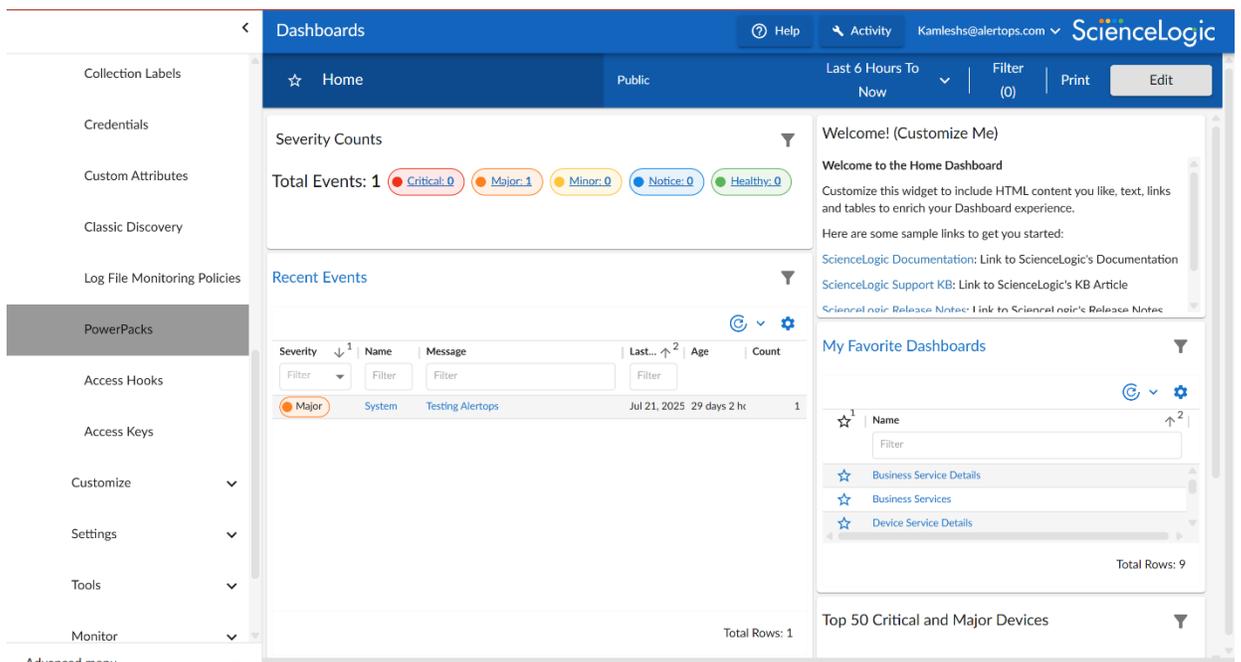


- Assign an **Escalation Policy** and a **Recipient Group**.
- Save the integration to generate a unique **API endpoint URL**. Copy this URL for use in the SL1 configuration.



2. Import the AlertOps PowerPack into ScienceLogic SL1

- Log into SL1.
- Navigate to **System > Manage > PowerPacks**.



- Choose **Import PowerPack**.
- Select the AlertOps PowerPack file and click **Import**.
- Click **Install** in the PowerPack Installer modal.

3. Configure the AlertOps Runbook Actions in SL1

1. In SL1, go to **Automation > Runbook Actions**.
2. Locate the two AlertOps snippets.
3. For each snippet, set the **ALERTOPS_WEBHOOK_URL** to the AlertOps API endpoint URL:
 1. **AlertOps – ACTIVE Event POST 2.0**
 2. **AlertOps – CLEAR Event POST 2.0**

The screenshot shows the 'Action Editor' window with the following configuration:

- Policy Editor | Editing Action [45]** (with a 'Reset' button)
- Action Name:** AlertOps - ACTIVE Event POST v2.0
- Action State:** [Enabled]
- Description:** POST Event Details to AlertOps
- Organization:** [System]
- Action Type:** Run a Snippet
- Snippet Credential:** [(None)]
- Action Run Context:** [Database]
- Execution Environment:** [-- Default Environment]
- Snippet Code:**

```
import json
import urllib.request
import urllib.parse
import ssl

# AlertOps webhook URL
ALERTOPS_WEBHOOK_URL = "https://notify.alertops.com/POSTAlert/abcd-3738-4794-b0fd-
e865888e8210/CUSTOM"

try:
    event_data = {
        "event_id": EM7_VALUES["%e"],
        "event_message": EM7_VALUES["%M"],
        "device_name": EM7_VALUES["%X"],
        "device_ip": EM7_VALUES["%a"],
        "severity": EM7_VALUES["%s"],
        "timestamp": EM7_VALUES["%t"],
        "event_source": "ScienceLogic SL1",
```
- Buttons:** Save, Save As

4. Configure the AlertOps Runbook Policies

1. Navigate to **Registry > Runbook > Automations**.
2. Enable the following:
 - **AlertOps Active Event Trigger v2.0**
 - **AlertOps Clear Event Trigger v2.0**
3. Review and adjust devices and events to fit your alerting strategy.

The screenshot shows the 'Automation Policy Editor' for 'AlertOps Active Event Trigger v2.0'. The interface is divided into several sections:

- Policy Configuration:** Policy Name: AlertOps Active Event Trigger v2.0; Policy Type: Active Events; Policy State: Enabled; Policy Priority: Default; Organization: System.
- Criteria Logic:** Severity >= Minor; Match Logic: Text search; Repeat Time: Only once; Align With: Devices.
- Available Devices:** A.Ops, AWS: Service: Test Device 1, AWS: Service: TestDevice2.
- Aligned Devices:** (All devices).
- Available Events:** A list of 15 critical events including AC/DC voltage sensor issues, smoke detector alerts, and AWS network/connection failures.
- Aligned Events:** (All events).
- Available Actions:** Send Email, SNMP Trap, Snippet [5]: AlertOps - ACTIVE Event POST v2.0, Snippet [5]: AlertOps - CLEAR Event POST v2.0, Snippet [5]: AO-AUTO-EventWebhook-AlertOpsIntegration, Snippet [5]: AO-PostWebhook2, Snippet [5]: AWS: Account Creation, Snippet [5]: AWS: Account Write Back, Snippet [5]: AWS: Disable Instance By Tag, Snippet [5]: AWS: Disable from EC2 ID.
- Aligned Actions:** 1. Snippet [5]: AlertOps - ACTIVE Event POST v2.0.

Buttons for 'Save' and 'Save As' are visible at the bottom.

Experience: Using the Integration

Once configured, customers experience a **seamless event-to-resolution pipeline**.

1. Event Triggered in SL1

- A critical issue (e.g., CPU overload, service failure) is detected.
- The RBA Policy executes, sending event details to AlertOps.

The screenshot shows the ScienceLogic Events dashboard. The main content area displays an event overview for 'Test AlertOps High CPU Utilization detected on server: web-prod-01.example.com'. The event details include:

| Organization | Device Name | Severity | Event Type | Event Source | First Detected | Occurrence Count | Last Detected | Event ID | External Ticket ID |
|--------------|---------------|----------|------------|--------------|--|------------------|--|----------|--------------------|
| A_Ops | Test Device 1 | Major | Device | API | 8 months 28 days Jan 1, 2025, 7:20 AM | 1 | 8 months 28 days Jan 1, 2025, 7:20 AM | 52 | |

Below the event overview, there is an 'Event Policy Information' section showing the policy name 'AlertOps v2.0' and a 'Device Details' section showing the device name 'Test Device 1' and its IP address.

2. Alert Created in AlertOps

- AlertOps parses the payload, applies templates, and creates a grouped alert (if relevant).

The screenshot shows the AlertOps Inbound Log for an inbound message. The message details are as follows:

| Date Sent | Date Received | Date Processed | Status |
|------------------------|------------------------|------------------------|---------------|
| 10/01/2025 06:14:16 AM | 10/01/2025 06:14:16 AM | 10/01/2025 06:14:16 AM | Mapped Opened |

Integration: ScienceLogic
Source: SL1Events
Source Name: ScienceLogic
Source Id: 52

Alert Id: 85076005
Subject: MAJOR | Test AlertOps High CPU Utilization detected on server: web-prod-01.example.com

Body:

```
{
  "application": "",
  "event_url": "http://em7.mydomain.com/events/detail/52",
  "vendor": "Test AlertOps 95% High CPU Util",
  "description": "SL1 Event: Test AlertOps High CPU Utilization detected on server: web-prod-01.example.com on Test Device 1",
  "timestamp": "0",
  "device_ip": "",
  "event_message": "Test AlertOps High CPU Utilization detected on server: web-prod-01.example.com",
  "severity_name": "MAJOR",
  "device_class": "",
  "interface": "",
  "threshold_value": "90%",
  "..."
}
```

- The alert is enriched with metadata (time, source, severity, device).

Alert Insights

Agent FixIt

1. Check CPU usage on the server using the top command.
2. Identify any processes consuming excessive CPU resources.
3. If necessary, restart the problematic service using systemctl.
4. Monitor the server for stability post-fix.
5. Document the findings and actions taken in the incident log.

Agent SignalCheck

False Positive: False
Confidence: High

Agent RootCause

Likely Cause: High CPU utilization
Explanation: The alert indicates that the server web-prod-01.example.com is experiencing high CPU usage, likely due to a runaway process or insufficient resources.

Agent Prioritize

Urgency: High
Impact: Medium
Reason: High CPU usage can affect application performance and user experience.

Agent Context

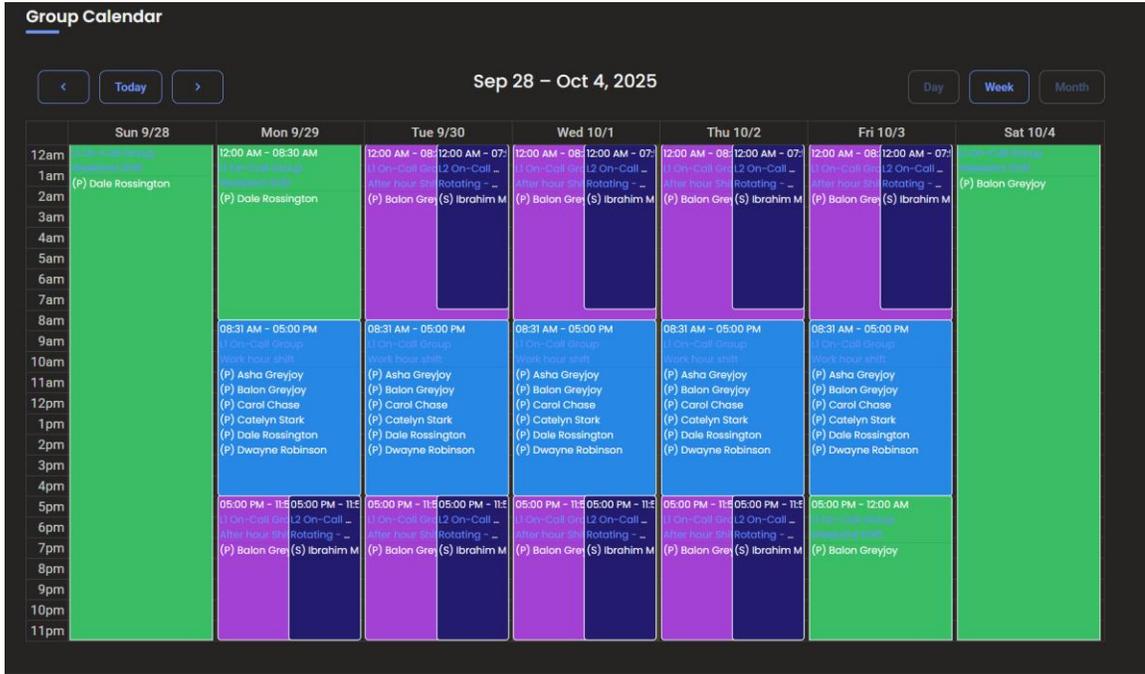
Summary: High CPU utilization detected on server.
Service/System: web-prod-01.example.com
Why It Matters: This could lead to degraded performance or downtime for users.

ScienceLogic

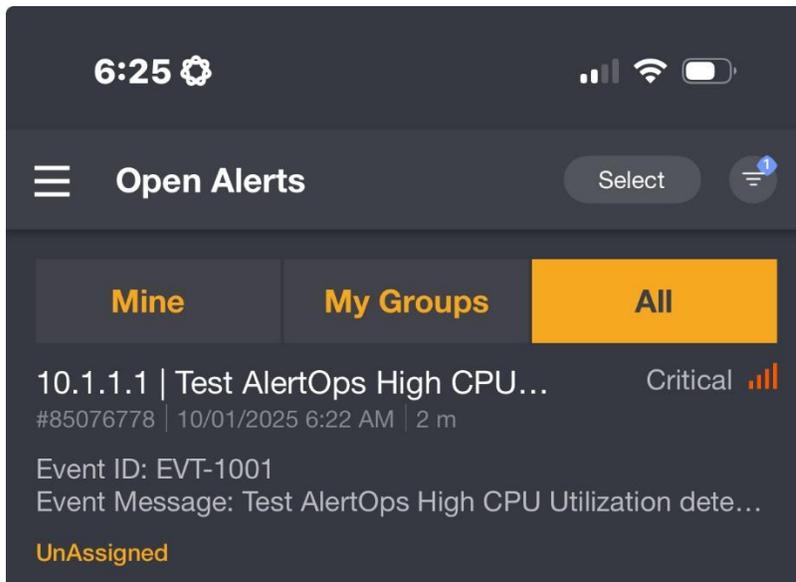
| | | |
|--|-----------------|----------------------------------|
| Event Type | Model | Vendor |
| Performance | Catalyst 9500 | Test AlertOps 95% High CPU Utili |
| Location | Service | Application |
| Chicago Data Center 1 | Core Networking | N/A |
| Interface | Port | Device Class |
| GigabitEthernet0/2 | Gig0/2 | Network Switch |
| Event Class | Current Value | Threshold Value |
| Interface Utilization | 79% | 90% |
| Collector | Organization | Timestamp |
| Chicago-DC1 | Acme Corp | 8/8/2025 2:10:23 PM |
| Alarm Severity | Device IP | Device Name |
| 3 | 10.1.1.1 | Test Device 1 |
| Event Message | Event ID | |
| Test AlertOps High CPU Utilization detected on server: web-prod-01.example.com | EVT-1001 | |

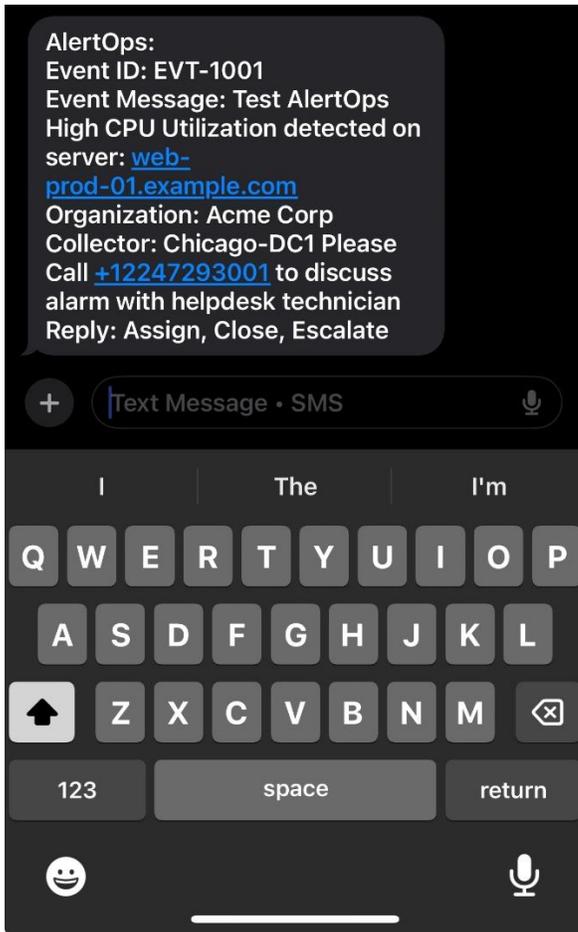
3. Multi-Channel Notifications

- AlertOps determines who the appropriate On-Call Member would be from the shifts.



- Primary on-call receives a notification through their preferred contacts such as **SMS**, **E-mail**, **Push Notification**, and **voice call**





Alert ID: 85076778 10.1.1.1 | Test AlertOps High CPU Utilization detected on server: web-prod-01.example.com Inbox x

AlertOps Alerts <mspdemos-Alerts@email.alertops.com>

6:22 AM (6 minutes ago) ☆ 😊 ↶

AlertOps

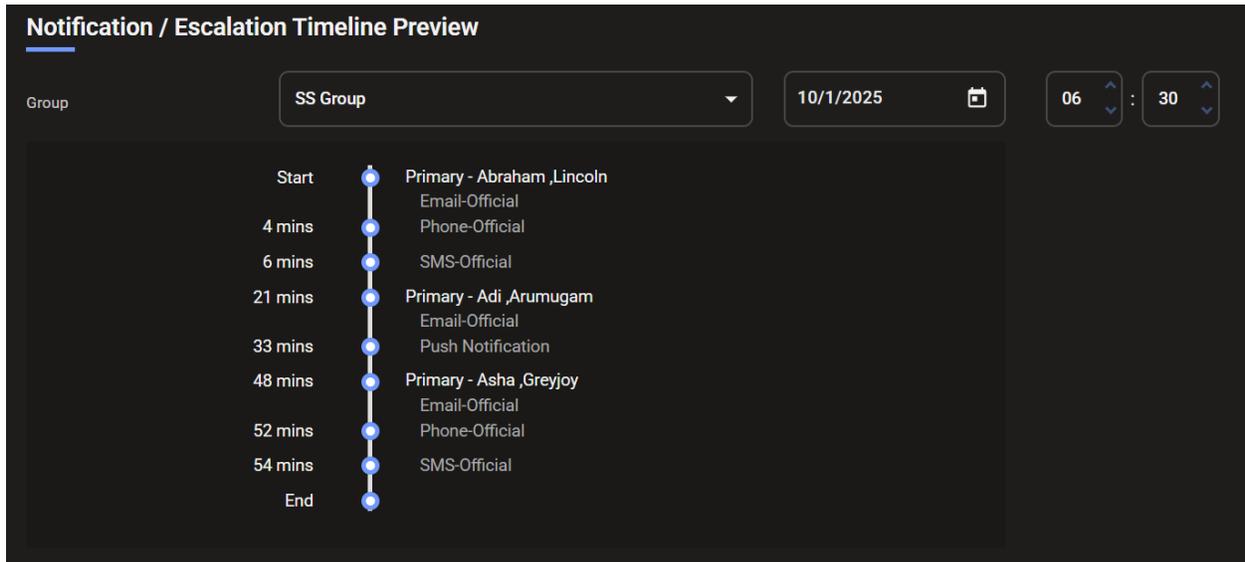
Event ID: EVT-1001 Event Message: Test AlertOps High CPU Utilization detected on server: [web-prod-01.example.com](#) Organization: Acme Corp Collector: Chicago-DC1 Please Call +12247293001 to discuss alarm with helpdesk technician

[Assign](#) [Escalate](#) [Close](#)

4. Escalation in Action

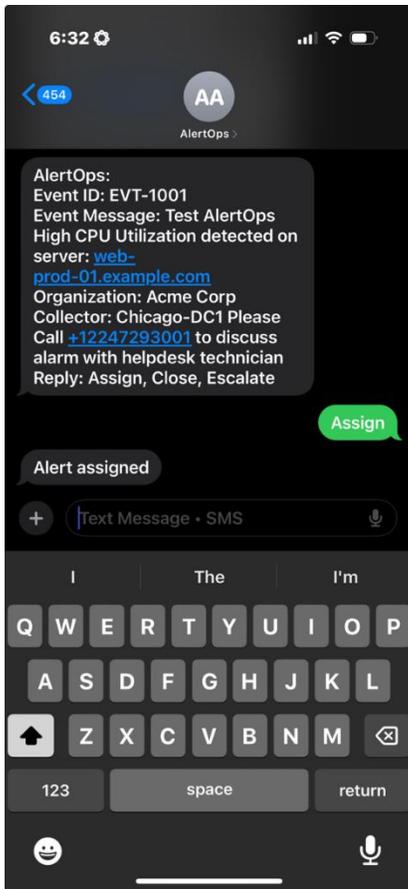
- If Primary does not acknowledge in X minutes, escalation automatically triggers.
- Secondary receives a **phone call + email**.

- Manager is notified if escalation reaches final stage.



5. Acknowledgement & Closure

- On-call acknowledges via **mobile app swipe**, **reply to SMS**, or **click in Slack/Teams** to prevent further escalation
- Once resolved in SL1 the AlertOps Alert will be Auto-Closed



Customer Benefits in Daily Use

- **Confidence:** Teams know every SL1 event reaches the right person.
- **Speed:** Escalations cut MTTA dramatically.
- **Clarity:** Centralized alert feed in AlertOps eliminates noise.
- **Mobility:** On-call can acknowledge directly from mobile or chat apps.
- **Accountability:** Audit trails show exactly who responded and when.