# Chapter: Setting Up Alerts in Freshservice Alert Management

Setting up alerts is a cornerstone of Freshservice Alert Management, enabling IT teams to respond proactively to system events and potential disruptions. This chapter explores the process of configuring alert sources, with detailed examples for email-based alerts and integration with SolarWinds. Additionally, it demonstrates how alert-based workflow automation can seamlessly trigger incident workflows, ensuring a streamlined response to IT issues.

## Understanding Alert Sources in Freshservice

Alert sources are external systems or applications configured to send notifications to Freshservice when predefined conditions are met. These can include monitoring tools, email systems, or custom integrations using APIs. Properly configuring these sources ensures that Freshservice captures all critical events, enabling IT teams to act swiftly and effectively.

## Example 1: Setting Up Email as an Alert Source

Email remains one of the most versatile and widely used methods for transmitting alerts. Many monitoring tools and systems are capable of sending email notifications when specific conditions occur. Here's how to configure email as an alert source in Freshservice:

1. Navigate to the \*\*Admin Settings\*\* in Freshservice and select \*\*Alert Sources\*\*.  
2. Choose \*\*Email\*\* as the alert source and specify an email address where Freshservice will receive notifications.  
3. Configure parsing rules to extract critical information such as severity, category, and the source of the alert from the email subject or body.  
4. Map the parsed data to Freshservice fields for actionable insights.  
5. Send a test email to ensure that the parsing rules and field mappings work correctly.

## Example 2: Setting Up SolarWinds as an Alert Source

SolarWinds is a comprehensive monitoring tool widely used for network and system management. To set up SolarWinds as an alert source in Freshservice:

1. In Freshservice, navigate to \*\*Admin Settings\*\* and select \*\*Alert Sources\*\*.  
2. Choose \*\*SolarWinds\*\* from the list of supported integrations and provide the required API credentials, including the API URL, username, and password.  
3. Define alert criteria in SolarWinds to specify which events should trigger notifications. For example, alerts for high CPU usage or network outages.  
4. Test the integration by generating a sample alert in SolarWinds and ensuring it appears correctly in Freshservice.  
5. Use Freshservice to validate that the alert details, such as severity and source, are accurately recorded.

## Using Alert-Based Workflow Automation

Freshservice's workflow automation capabilities extend the power of alert management by enabling automatic responses to specific events. When an alert is received, predefined workflows can be triggered to perform actions such as creating incidents, notifying stakeholders, and initiating resolution tasks.

For example, consider the following scenario: An alert from SolarWinds indicates a critical server outage. A workflow automation is triggered by the alert, performing the following actions:  
1. Automatically create a high-priority incident ticket in Freshservice, linking it to the original alert.  
2. Assign the incident to the appropriate support team based on predefined criteria, such as server location or type.  
3. Notify IT managers and stakeholders via email and SMS, ensuring they are informed of the situation.  
4. Launch a predefined remediation plan, which could include dispatching a technician or initiating a server restart.  
5. Log all actions for audit and post-incident analysis.

## Conclusion

Setting up alerts in Freshservice Alert Management and integrating them with workflow automation creates a robust framework for proactive IT management. Whether using email or advanced monitoring tools like SolarWinds, organizations can ensure timely and efficient responses to system events. By leveraging automation, IT teams not only reduce manual effort but also enhance their ability to maintain system reliability and minimize downtime.