# Chapter: Creating Workflow Automations for Alerts in Freshservice

Workflow automation is a key feature of Freshservice Alert Management, enabling IT teams to handle alerts effectively and efficiently. By automating predefined actions in response to specific alert conditions, organizations can minimize manual intervention and ensure consistent incident handling. This chapter provides a detailed guide to creating workflow automations for alerts and illustrates their applications through five example workflows.

## The Process of Creating a Workflow Automation for Alerts

Creating a workflow automation in Freshservice involves the following steps:  
1. \*\*Access the Workflow Automator\*\*: Log in to Freshservice, navigate to the Admin settings, and select Workflow Automator.  
2. \*\*Define the Trigger\*\*: Choose 'Alert' as the trigger event. Specify the conditions under which the workflow will activate, such as severity, category, or source of the alert.  
3. \*\*Add Conditions\*\*: Use conditional branches to refine the workflow's behavior based on alert attributes like priority or team assignment.  
4. \*\*Define Actions\*\*: Specify the actions to be taken when the workflow is triggered. Examples include creating incidents, notifying stakeholders, or updating alert statuses.  
5. \*\*Test and Publish\*\*: Test the workflow with sample alerts to verify functionality. Once satisfied, publish the workflow for live use.

## Example Workflow 1: Critical to Major Incident Workflow

Scenario: An alert with 'Critical' severity is received, requiring immediate attention and escalation.  
Steps:  
1. Trigger: Alert received with severity 'Critical'.  
2. Actions:  
 - Create a new incident with priority set to 'High'.  
 - Assign the incident to the 'Critical Incident Response Team'.  
 - Notify IT leadership via email and SMS.  
 - Update the alert status to 'In Progress'.

## Example Workflow 2: Error Workflow Automation to High Priority

Scenario: An alert with 'Error' severity is received, indicating a potential issue that requires prompt resolution.  
Steps:  
1. Trigger: Alert received with severity 'Error'.  
2. Actions:  
 - Create a new incident with priority set to 'High'.  
 - Assign the incident to the technical support team responsible for the affected service.  
 - Send an email notification to the service owner with details of the issue.  
 - Automatically log the alert as an activity in the incident record.

## Example Workflow 3: Warning to Incident Workflow

Scenario: A 'Warning' alert is received, requiring further investigation but not immediate action.  
Steps:  
1. Trigger: Alert received with severity 'Warning'.  
2. Actions:  
 - Create a new incident with priority set to 'Low'.  
 - Assign the incident to the 'Monitoring Team' for analysis.  
 - Notify the team lead with the alert details for review.  
 - Add the alert to a queue for periodic review.

## Example Workflow 4: Resolving Workflow for "OK" Alerts

Scenario: An alert switches to 'OK' status, indicating that the issue has been resolved.  
Steps:  
1. Trigger: Alert status changes to 'OK'.  
2. Actions:  
 - Locate and resolve any open incidents linked to the alert.  
 - Notify stakeholders that the issue has been resolved.  
 - Log the resolution in the system for audit purposes.  
 - Update the alert status to 'Resolved'.

## Example Workflow 5: "UP" or "Active" Severity Workflow

Scenario: An alert switches to 'UP' or 'Active' status after a service recovery.  
Steps:  
1. Trigger: Alert status changes to 'UP' or 'Active'.  
2. Actions:  
 - Confirm service recovery by cross-referencing monitoring tools.  
 - Resolve all related incidents.  
 - Notify the operations team of service restoration.  
 - Update the alert status to 'Closed' and log the recovery.

## Conclusion

Workflow automation in Freshservice enables IT teams to respond to alerts in a structured and efficient manner. By implementing tailored workflows for various alert severities and statuses, organizations can reduce response times, ensure consistency in incident handling, and maintain operational reliability. The examples provided in this chapter serve as a foundation for designing workflows that meet specific organizational needs.