Dynatrace Workflows – Detailed Guide

# What Is Dynatrace Workflow?

Dynatrace Workflows are event-driven, no-code/low-code automation pipelines that enable users to:  
- Respond to incidents (e.g., self-healing)  
- Orchestrate external tools (Jira, ServiceNow, Slack, AWS, etc.)  
- Automate monitoring and remediation tasks

# Key Components

• Trigger: Starts the workflow (problem, schedule, webhook)  
• Input: Optional data  
• Steps: Logical blocks (if, loop, action)  
• Actions: Tasks (notify, API, script)  
• Davis CoPilot: AI assistant for building steps  
• Custom logic: Python/JS/Bash scripting

# Supported Triggers

- Event-based: Problems, events, thresholds  
- Schedule-based: Every X minutes/hours  
- Manual trigger: UI or API  
- External: Webhook from external systems

# Example Use Cases

• Auto-remediate high CPU  
• Create ServiceNow ticket  
• Slack alert for log anomaly  
• Scale AWS resources  
• Run synthetic test + notify QA

# Hands-On: Create a Basic Workflow

Goal: Notify Slack on critical problem  
  
1. Open Workflows → Create workflow  
2. Add Trigger: Problem severity = critical  
3. Add Slack step:  
 Message: 🚨 Critical problem detected: {{problemTitle}} / {{entityName}}  
4. Test + Activate

# Advanced Logic

- Conditions (if problem tag = X)  
- Loops (over affected entities)  
- JSON input/output chaining  
- Run scripts (Python, Bash)

# Integration Options

• ITSM: Jira, ServiceNow, PagerDuty  
• Messaging: Slack, Teams, Email  
• Cloud: AWS, Azure, GCP APIs  
• CI/CD: Jenkins, GitHub, GitLab  
• Observability: Fetch metrics, logs, DQL

# Security & Governance

• RBAC for workflow access  
• Logs of executions  
• Input/output audit  
• Secure credentials in Vault

# Monitoring Workflow Runs

• Execution history & logs  
• Retry failed steps  
• Visual flow and outputs

# Summary

Dynatrace Workflows enable:  
- Drag-and-drop automation  
- Problem/metric triggers  
- 3rd-party integrations  
- Custom logic with scripts  
- Davis AI assistance