**## 1. Deploy Scripts to Shared Location**

**### Option A: Network File Share**

**bash**

# Create shared directory structure

sudo mkdir -p /shared/amazonq-jenkins/{scripts,config,docs}

sudo chmod 755 /shared/amazonq-jenkins

sudo chown -R jenkins-admin:jenkins-users /shared/amazonq-jenkins

**### Option B: Git Repository (Recommended)**

🛠️ Using tool: fs\_write (trusted)

⋮

● Path: /Users/kombaths/deploy-shared-scripts.sh

+ 1: #!/bin/bash

+ 2:

+ 3: # Deployment script for Amazon Q Jenkins Integration

+ 4: # This script deploys the integration to a shared location

+ 5:

+ 6: SHARED\_PATH="/shared/amazonq-jenkins"

+ 7: REPO\_URL="https://github.com/your-org/amazonq-jenkins-integration.git"

+ 8:

+ 9: # Create directory structure

+ 10: create\_shared\_structure() {

+ 11: echo "Creating shared directory structure..."

+ 12: sudo mkdir -p ${SHARED\_PATH}/{scripts,config,docs,logs}

+ 13: sudo chmod -R 755 ${SHARED\_PATH}

+ 14: sudo chown -R jenkins-admin:jenkins-users ${SHARED\_PATH}

+ 15: }

+ 16:

+ 17: # Deploy scripts

+ 18: deploy\_scripts() {

+ 19: echo "Deploying scripts to shared location..."

+ 20:

+ 21: # Copy main scripts

+ 22: cp jenkins-trigger-script.sh ${SHARED\_PATH}/scripts/

+ 23: cp amazonq-jenkins-integration.sh ${SHARED\_PATH}/scripts/

+ 24:

+ 25: # Make scripts executable

+ 26: chmod +x ${SHARED\_PATH}/scripts/\*.sh

+ 27:

+ 28: # Create symlinks for easy access

+ 29: sudo ln -sf ${SHARED\_PATH}/scripts/jenkins-trigger-script.sh /usr/local/bin/trigger-jenkins

+ 30: sudo ln -sf ${SHARED\_PATH}/scripts/amazonq-jenkins-integration.sh /usr/local/bin/amazonq-jenkins

+ 31: }

+ 32:

+ 33: # Set up Git repository (if using version control)

+ 34: setup\_git\_repo() {

+ 35: if [ ! -d "${SHARED\_PATH}/.git" ]; then

+ 36: cd ${SHARED\_PATH}

+ 37: git init

+ 38: git remote add origin ${REPO\_URL}

+ 39: git add .

+ 40: git commit -m "Initial deployment of Amazon Q Jenkins integration"

+ 41: git push -u origin main

+ 42: fi

+ 43: }

+ 44:

+ 45: # Main execution

+ 46: main() {

+ 47: create\_shared\_structure

+ 48: deploy\_scripts

+ 49: setup\_git\_repo

+ 50: echo "✅ Deployment completed successfully!"

+ 51: echo "Scripts available at: ${SHARED\_PATH}/scripts/"

+ 52: }

+ 53:

+ 54: main "$@"

⋮

↳ Purpose: Create deployment script for shared Jenkins integration

Creating: /Users/kombaths/deploy-shared-scripts.sh

⋮

**● Completed in 0.0s**

> ## 2. Create Central Configuration

🛠️ Using tool: fs\_write (trusted)

⋮

● Path: /Users/kombaths/jenkins-jobs-config.yaml

+ 1: # Central Jenkins Jobs Configuration

+ 2: # This file defines all available Jenkins jobs and their configurations

+ 3:

+ 4: jenkins:

+ 5: base\_url: "https://ci.sms.us.caas.oneadp.com"

+ 6: default\_timeout: 300

+ 7: retry\_attempts: 3

+ 8:

+ 9: # Job definitions

+ 10: jobs:

+ 11: # Development Jobs

+ 12: - name: "OLP-2"

+ 13: display\_name: "OLP-2 Development Build"

+ 14: description: "Triggers OLP-2 development build and deployment"

+ 15: parameters: []

+ 16: keywords: ["olp-2", "olp2", "development", "dev build"]

+ 17: team: "development"

+ 18:

+ 19: - name: "QA-Host-File-Update"

+ 20: display\_name: "QA Host File Update"

+ 21: description: "Updates host files in QA environment"

+ 22: parameters:

+ 23: - name: "environment"

+ 24: type: "choice"

+ 25: choices: ["qa1", "qa2", "qa3"]

+ 26: default: "qa1"

+ 27: keywords: ["qa host", "host file", "qa update"]

+ 28: team: "qa"

+ 29:

+ 30: - name: "Production-Deploy"

+ 31: display\_name: "Production Deployment"

+ 32: description: "Deploys application to production"

+ 33: parameters:

+ 34: - name: "version"

+ 35: type: "string"

+ 36: required: true

+ 37: - name: "rollback\_enabled"

+ 38: type: "boolean"

+ 39: default: true

+ 40: keywords: ["prod deploy", "production", "release"]

+ 41: team: "devops"

+ 42: approval\_required: true

+ 43:

+ 44: # Infrastructure Jobs

+ 45: - name: "Infrastructure-Provision"

+ 46: display\_name: "Infrastructure Provisioning"

+ 47: description: "Provisions AWS infrastructure using Terraform"

+ 48: parameters:

+ 49: - name: "environment"

+ 50: type: "choice"

+ 51: choices: ["dev", "staging", "prod"]

+ 52: - name: "action"

+ 53: type: "choice"

+ 54: choices: ["plan", "apply", "destroy"]

+ 55: default: "plan"

+ 56: keywords: ["infra", "terraform", "provision", "infrastructure"]

+ 57: team: "platform"

+ 58:

+ 59: # Testing Jobs

+ 60: - name: "Automated-Tests"

+ 61: display\_name: "Run Automated Test Suite"

+ 62: description: "Executes full automated test suite"

+ 63: parameters:

+ 64: - name: "test\_suite"

+ 65: type: "choice"

+ 66: choices: ["unit", "integration", "e2e", "all"]

+ 67: default: "all"

+ 68: keywords: ["test", "testing", "automated tests", "test suite"]

+ 69: team: "qa"

+ 70:

+ 71: # Team configurations

+ 72: teams:

+ 73: development:

+ 74: jenkins\_folder: "Development"

+ 75: notification\_channel: "#dev-notifications"

+ 76:

+ 77: qa:

+ 78: jenkins\_folder: "QA"

+ 79: notification\_channel: "#qa-notifications"

+ 80:

+ 81: devops:

+ 82: jenkins\_folder: "DevOps"

+ 83: notification\_channel: "#devops-notifications"

+ 84:

+ 85: platform:

+ 86: jenkins\_folder: "Platform"

+ 87: notification\_channel: "#platform-notifications"

+ 88:

+ 89: # Amazon Q prompt patterns

+ 90: prompt\_patterns:

+ 91: trigger\_job:

+ 92: - "trigger {job\_name}"

+ 93: - "run {job\_name}"

+ 94: - "start {job\_name}"

+ 95: - "execute {job\_name}"

+ 96:

+ 97: status\_check:

+ 98: - "status of {job\_name}"

+ 99: - "check {job\_name}"

+ 100: - "how is {job\_name} doing"

+ 101:

+ 102: list\_jobs:

+ 103: - "list jobs"

+ 104: - "show available jobs"

+ 105: - "what jobs can I run"

⋮

↳ Purpose: Create central Jenkins jobs configuration file

Creating: /Users/kombaths/jenkins-jobs-config.yaml

⋮

**● Completed in 0.0s**

🛠️ Using tool: fs\_write (trusted)

⋮

● Path: /Users/kombaths/config-parser.sh

+ 1: #!/bin/bash

+ 2:

+ 3: # Configuration Parser for Jenkins Jobs

+ 4: # Reads the YAML configuration and provides job information

+ 5:

+ 6: CONFIG\_FILE="/shared/amazonq-jenkins/config/jenkins-jobs-config.yaml"

+ 7:

+ 8: # Function to parse YAML (requires yq tool)

+ 9: parse\_job\_config() {

+ 10: local job\_name="$1"

+ 11:

+ 12: if ! command -v yq &> /dev/null; then

+ 13: echo "Error: yq tool is required. Install with: brew install yq"

+ 14: exit 1

+ 15: fi

+ 16:

+ 17: # Get job details

+ 18: local job\_info=$(yq eval ".jobs[] | select(.name == \"$job\_name\")" "$CONFIG\_FILE")

+ 19:

+ 20: if [ -z "$job\_info" ]; then

+ 21: echo "Job '$job\_name' not found in configuration"

+ 22: return 1

+ 23: fi

+ 24:

+ 25: echo "$job\_info"

+ 26: }

+ 27:

+ 28: # Function to list all available jobs

+ 29: list\_all\_jobs() {

+ 30: echo "Available Jenkins Jobs:"

+ 31: echo "======================"

+ 32:

+ 33: yq eval '.jobs[] | "- " + .name + " (" + .display\_name + ")"' "$CONFIG\_FILE"

+ 34: }

+ 35:

+ 36: # Function to find job by keyword

+ 37: find\_job\_by\_keyword() {

+ 38: local keyword="$1"

+ 39:

+ 40: yq eval ".jobs[] | select(.keywords[] | contains(\"$keyword\")) | .name" "$CONFIG\_FILE"

+ 41: }

+ 42:

+ 43: # Function to get team jobs

+ 44: get\_team\_jobs() {

+ 45: local team="$1"

+ 46:

+ 47: yq eval ".jobs[] | select(.team == \"$team\") | .name" "$CONFIG\_FILE"

+ 48: }

+ 49:

+ 50: # Main execution

+ 51: case "$1" in

+ 52: "parse")

+ 53: parse\_job\_config "$2"

+ 54: ;;

+ 55: "list")

+ 56: list\_all\_jobs

+ 57: ;;

+ 58: "find")

+ 59: find\_job\_by\_keyword "$2"

+ 60: ;;

+ 61: "team")

+ 62: get\_team\_jobs "$2"

+ 63: ;;

+ 64: \*)

+ 65: echo "Usage: $0 {parse|list|find|team} [argument]"

+ 66: echo " parse <job\_name> - Get job configuration"

+ 67: echo " list - List all jobs"

+ 68: echo " find <keyword> - Find job by keyword"

+ 69: echo " team <team\_name> - Get jobs for team"

+ 70: ;;

+ 71: esac

⋮

↳ Purpose: Create configuration parser script for Jenkins jobs

Creating: /Users/kombaths/config-parser.sh

⋮

**● Completed in 0.0s**

> ## 3. Set Up Environment Variables for Team Members

🛠️ Using tool: fs\_write (trusted)

⋮

● Path: /Users/kombaths/setup-user-env.sh

+ 1: #!/bin/bash

+ 2:

+ 3: # User Environment Setup for Amazon Q Jenkins Integration

+ 4: # This script sets up individual user credentials and configurations

+ 5:

+ 6: SHARED\_CONFIG\_PATH="/shared/amazonq-jenkins/config"

+ 7: USER\_CONFIG\_PATH="$HOME/.amazonq-jenkins"

+ 8:

+ 9: # Function to create user configuration directory

+ 10: setup\_user\_config() {

+ 11: echo "Setting up user configuration directory..."

+ 12: mkdir -p "$USER\_CONFIG\_PATH"

+ 13: chmod 700 "$USER\_CONFIG\_PATH"

+ 14: }

+ 15:

+ 16: # Function to create user credentials file

+ 17: create\_user\_credentials() {

+ 18: local username="$1"

+ 19: local api\_token="$2"

+ 20: local team="$3"

+ 21:

+ 22: cat > "$USER\_CONFIG\_PATH/credentials" << EOF

+ 23: # Amazon Q Jenkins Integration - User Credentials

+ 24: # DO NOT SHARE THIS FILE

+ 25:

+ 26: JENKINS\_USER="$username"

+ 27: JENKINS\_TOKEN="$api\_token"

+ 28: USER\_TEAM="$team"

+ 29: USER\_EMAIL="$username@company.com"

+ 30:

+ 31: # Jenkins Configuration

+ 32: JENKINS\_BASE\_URL="https://ci.sms.us.caas.oneadp.com"

+ 33: JENKINS\_TIMEOUT=300

+ 34:

+ 35: # Notification Settings

+ 36: SLACK\_WEBHOOK\_URL=""

+ 37: EMAIL\_NOTIFICATIONS=true

+ 38:

+ 39: # Security Settings

+ 40: REQUIRE\_APPROVAL\_FOR\_PROD=true

+ 41: LOG\_ALL\_ACTIONS=true

+ 42: EOF

+ 43:

+ 44: chmod 600 "$USER\_CONFIG\_PATH/credentials"

+ 45: echo "✅ User credentials created at $USER\_CONFIG\_PATH/credentials"

+ 46: }

+ 47:

+ 48: # Function to create team-specific environment

+ 49: setup\_team\_environment() {

+ 50: local team="$1"

+ 51:

+ 52: case "$team" in

+ 53: "development"|"dev")

+ 54: cat > "$USER\_CONFIG\_PATH/team-config" << EOF

+ 55: ALLOWED\_JOBS="OLP-2,Development-Build,Unit-Tests"

+ 56: DEFAULT\_ENVIRONMENT="dev"

+ 57: NOTIFICATION\_CHANNEL="#dev-notifications"

+ 58: APPROVAL\_REQUIRED=false

+ 59: EOF

+ 60: ;;

+ 61: "qa")

+ 62: cat > "$USER\_CONFIG\_PATH/team-config" << EOF

+ 63: ALLOWED\_JOBS="QA-Host-File-Update,Automated-Tests,QA-Deploy"

+ 64: DEFAULT\_ENVIRONMENT="qa"

+ 65: NOTIFICATION\_CHANNEL="#qa-notifications"

+ 66: APPROVAL\_REQUIRED=false

+ 67: EOF

+ 68: ;;

+ 69: "devops"|"platform")

+ 70: cat > "$USER\_CONFIG\_PATH/team-config" << EOF

+ 71: ALLOWED\_JOBS="\*"

+ 72: DEFAULT\_ENVIRONMENT="all"

+ 73: NOTIFICATION\_CHANNEL="#devops-notifications"

+ 74: APPROVAL\_REQUIRED=true

+ 75: EOF

+ 76: ;;

+ 77: \*)

+ 78: echo "Unknown team: $team"

+ 79: exit 1

+ 80: ;;

+ 81: esac

+ 82:

+ 83: echo "✅ Team configuration created for: $team"

+ 84: }

+ 85:

+ 86: # Function to create shell aliases

+ 87: create\_shell\_aliases() {

+ 88: cat > "$USER\_CONFIG\_PATH/aliases.sh" << 'EOF'

+ 89: # Amazon Q Jenkins Integration Aliases

+ 90:

+ 91: # Source user credentials

+ 92: if [ -f "$HOME/.amazonq-jenkins/credentials" ]; then

+ 93: source "$HOME/.amazonq-jenkins/credentials"

+ 94: fi

+ 95:

+ 96: # Main aliases

+ 97: alias aq-jenkins='amazonq-jenkins'

+ 98: alias trigger-job='jenkins-trigger-script.sh'

+ 99: alias list-jobs='config-parser.sh list'

+ 100: alias job-status='jenkins-status.sh'

+ 101:

+ 102: # Quick job triggers

+ 103: alias trigger-olp2='jenkins-trigger-script.sh "OLP-2"'

+ 104: alias trigger-qa-host='jenkins-trigger-script.sh "QA-Host-File-Update"'

+ 105: alias run-tests='jenkins-trigger-script.sh "Automated-Tests"'

+ 106:

+ 107: # Team-specific functions

+ 108: dev-jobs() {

+ 109: config-parser.sh team development

+ 110: }

+ 111:

+ 112: qa-jobs() {

+ 113: config-parser.sh team qa

+ 114: }

+ 115:

+ 116: my-jobs() {

+ 117: if [ -n "$USER\_TEAM" ]; then

+ 118: config-parser.sh team "$USER\_TEAM"

+ 119: else

+ 120: echo "USER\_TEAM not set in credentials"

+ 121: fi

+ 122: }

+ 123:

+ 124: # Export functions

+ 125: export -f dev-jobs qa-jobs my-jobs

+ 126: EOF

+ 127:

+ 128: echo "✅ Shell aliases created"

+ 129: echo "Add this to your ~/.bashrc or ~/.zshrc:"

+ 130: echo "source $USER\_CONFIG\_PATH/aliases.sh"

+ 131: }

+ 132:

+ 133: # Interactive setup function

+ 134: interactive\_setup() {

+ 135: echo "Amazon Q Jenkins Integration - User Setup"

+ 136: echo "========================================"

+ 137:

+ 138: read -p "Enter your Jenkins username: " username

+ 139: read -s -p "Enter your Jenkins API token: " api\_token

+ 140: echo

+ 141: read -p "Enter your team (development/qa/devops): " team

+ 142:

+ 143: setup\_user\_config

+ 144: create\_user\_credentials "$username" "$api\_token" "$team"

+ 145: setup\_team\_environment "$team"

+ 146: create\_shell\_aliases

+ 147:

+ 148: echo ""

+ 149: echo "🎉 Setup completed successfully!"

+ 150: echo ""

+ 151: echo "Next steps:"

+ 152: echo "1. Add to your shell profile: source $USER\_CONFIG\_PATH/aliases.sh"

+ 153: echo "2. Reload your shell: source ~/.bashrc (or ~/.zshrc)"

+ 154: echo "3. Test with: aq-jenkins --help"

+ 155: }

+ 156:

+ 157: # Main execution

+ 158: case "$1" in

+ 159: "interactive"|"")

+ 160: interactive\_setup

+ 161: ;;

+ 162: "credentials")

+ 163: create\_user\_credentials "$2" "$3" "$4"

+ 164: ;;

+ 165: "team")

+ 166: setup\_team\_environment "$2"

+ 167: ;;

+ 168: "aliases")

+ 169: create\_shell\_aliases

+ 170: ;;

+ 171: \*)

+ 172: echo "Usage: $0 [interactive|credentials|team|aliases]"

+ 173: echo " interactive - Run interactive setup"

+ 174: echo " credentials <user> <token> <team> - Create credentials"

+ 175: echo " team <team\_name> - Setup team environment"

+ 176: echo " aliases - Create shell aliases"

+ 177: ;;

+ 178: esac

⋮

↳ Purpose: Create environment setup script for team members

Creating: /Users/kombaths/setup-user-env.sh

⋮

**● Completed in 0.0s**

> ## 4. Document Prompt Patterns

🛠️ Using tool: fs\_write (trusted)

⋮

● Path: /Users/kombaths/AMAZONQ\_JENKINS\_GUIDE.md

+ 1: # Amazon Q Jenkins Integration Guide

+ 2:

+ 3: ## Overview

+ 4: This guide explains how to use Amazon Q to trigger Jenkins jobs using natural language prompts.

+ 5:

+ 6: ## Quick Start

+ 7:

+ 8: ### Basic Job Triggering

+ 9: Use these natural language patterns with Amazon Q:

+ 10:

+ 11: ```

+ 12: "Please trigger OLP-2"

+ 13: "Run the QA host file update"

+ 14: "Start automated tests"

+ 15: "Execute production deployment"

+ 16: ```

+ 17:

+ 18: ### Job Status Checking

+ 19: ```

+ 20: "What's the status of OLP-2?"

+ 21: "Check if the QA job is running"

+ 22: "Show me the last build results"

+ 23: ```

+ 24:

+ 25: ### Listing Available Jobs

+ 26: ```

+ 27: "What jobs can I run?"

+ 28: "List all available Jenkins jobs"

+ 29: "Show me development jobs"

+ 30: ```

+ 31:

+ 32: ## Detailed Prompt Patterns

+ 33:

+ 34: ### 1. Job Triggering Patterns

+ 35:

+ 36: | Pattern | Example | Description |

+ 37: |---------|---------|-------------|

+ 38: | `trigger <job-name>` | "trigger OLP-2" | Direct job trigger |

+ 39: | `run <job-name>` | "run automated tests" | Alternative trigger |

+ 40: | `start <job-name>` | "start production deploy" | Another trigger option |

+ 41: | `execute <job-name>` | "execute QA host update" | Formal trigger |

+ 42:

+ 43: ### 2. Parameterized Jobs

+ 44:

+ 45: For jobs that require parameters:

+ 46:

+ 47: ```

+ 48: "Trigger OLP-2 with environment=qa"

+ 49: "Run tests with test\_suite=integration"

+ 50: "Deploy to production with version=1.2.3"

+ 51: ```

+ 52:

+ 53: ### 3. Team-Specific Jobs

+ 54:

+ 55: #### Development Team

+ 56: ```

+ 57: "Run OLP-2 development build"

+ 58: "Trigger unit tests"

+ 59: "Start development deployment"

+ 60: ```

+ 61:

+ 62: #### QA Team

+ 63: ```

+ 64: "Update QA host files"

+ 65: "Run automated test suite"

+ 66: "Deploy to QA environment"

+ 67: ```

+ 68:

+ 69: #### DevOps Team

+ 70: ```

+ 71: "Provision infrastructure for staging"

+ 72: "Deploy to production with rollback enabled"

+ 73: "Run infrastructure tests"

+ 74: ```

+ 75:

+ 76: ## Available Jobs by Team

+ 77:

+ 78: ### Development Jobs

+ 79: - \*\*OLP-2\*\*: Development build and deployment

+ 80: - \*\*Unit-Tests\*\*: Run unit test suite

+ 81: - \*\*Development-Deploy\*\*: Deploy to dev environment

+ 82:

+ 83: ### QA Jobs

+ 84: - \*\*QA-Host-File-Update\*\*: Update host files in QA

+ 85: - \*\*Automated-Tests\*\*: Full automated test suite

+ 86: - \*\*QA-Deploy\*\*: Deploy to QA environment

+ 87:

+ 88: ### DevOps Jobs

+ 89: - \*\*Production-Deploy\*\*: Production deployment (requires approval)

+ 90: - \*\*Infrastructure-Provision\*\*: Terraform infrastructure provisioning

+ 91: - \*\*Backup-Restore\*\*: Database backup and restore operations

+ 92:

+ 93: ## Advanced Usage

+ 94:

+ 95: ### Conditional Triggers

+ 96: ```

+ 97: "If the last build passed, trigger production deployment"

+ 98: "Run tests only if code changed in the last hour"

+ 99: ```

+ 100:

+ 101: ### Batch Operations

+ 102: ```

+ 103: "Trigger all QA jobs"

+ 104: "Run development pipeline end-to-end"

+ 105: "Execute full deployment sequence"

+ 106: ```

+ 107:

+ 108: ### Scheduled Operations

+ 109: ```

+ 110: "Schedule OLP-2 to run at 2 PM"

+ 111: "Set up nightly automated tests"

+ 112: ```

+ 113:

+ 114: ## Error Handling

+ 115:

+ 116: ### Common Issues and Solutions

+ 117:

+ 118: | Error | Cause | Solution |

+ 119: |-------|-------|----------|

+ 120: | "Job not found" | Incorrect job name | Use `list jobs` to see available jobs |

+ 121: | "Permission denied" | Insufficient permissions | Contact your team lead |

+ 122: | "Build failed" | Job execution error | Check Jenkins logs |

+ 123: | "Timeout" | Job took too long | Check Jenkins queue |

+ 124:

+ 125: ### Troubleshooting Commands

+ 126: ```

+ 127: "Show me the last error for OLP-2"

+ 128: "What went wrong with the QA deployment?"

+ 129: "Check Jenkins queue status"

+ 130: ```

+ 131:

+ 132: ## Security and Permissions

+ 133:

+ 134: ### Team Permissions

+ 135: - \*\*Development\*\*: Can trigger dev and test jobs

+ 136: - \*\*QA\*\*: Can trigger QA and test jobs

+ 137: - \*\*DevOps\*\*: Can trigger all jobs including production

+ 138: - \*\*Platform\*\*: Full access to infrastructure jobs

+ 139:

+ 140: ### Approval Requirements

+ 141: Some jobs require approval:

+ 142: - Production deployments

+ 143: - Infrastructure changes

+ 144: - Database operations

+ 145:

+ 146: ### Audit Trail

+ 147: All job triggers are logged with:

+ 148: - User who triggered the job

+ 149: - Timestamp

+ 150: - Parameters used

+ 151: - Amazon Q conversation context

+ 152:

+ 153: ## Best Practices

+ 154:

+ 155: ### 1. Use Descriptive Prompts

+ 156: ✅ Good: "Trigger OLP-2 development build for feature branch"

+ 157: ❌ Bad: "run job"

+ 158:

+ 159: ### 2. Specify Environment When Needed

+ 160: ✅ Good: "Deploy to QA environment"

+ 161: ❌ Bad: "Deploy"

+ 162:

+ 163: ### 3. Check Status Before Retriggering

+ 164: ✅ Good: "Check OLP-2 status, then retrigger if failed"

+ 165: ❌ Bad: Multiple rapid triggers

+ 166:

+ 167: ### 4. Use Team Context

+ 168: ✅ Good: "Run our team's standard test suite"

+ 169: ❌ Bad: "Run tests" (ambiguous)

+ 170:

+ 171: ## Integration Examples

+ 172:

+ 173: ### Example 1: Development Workflow

+ 174: ```

+ 175: User: "I just pushed code to feature-branch. Can you run the development pipeline?"

+ 176: Amazon Q: "I'll trigger the OLP-2 development build for you."

+ 177: [Executes: jenkins-trigger-script.sh "OLP-2"]

+ 178: ```

+ 179:

+ 180: ### Example 2: QA Workflow

+ 181: ```

+ 182: User: "The dev build passed. Please update QA host files and run tests."

+ 183: Amazon Q: "I'll update the QA host files first, then run the automated tests."

+ 184: [Executes: jenkins-trigger-script.sh "QA-Host-File-Update"]

+ 185: [Then: jenkins-trigger-script.sh "Automated-Tests"]

+ 186: ```

+ 187:

+ 188: ### Example 3: Production Deployment

+ 189: ```

+ 190: User: "We're ready for production deployment of version 2.1.0"

+ 191: Amazon Q: "Production deployment requires approval. I'll trigger the job with version 2.1.0 and notify the DevOps team."

+ 192: [Executes: jenkins-trigger-script.sh "Production-Deploy" "version=2.1.0"]

+ 193: ```

+ 194:

+ 195: ## Setup for New Team Members

+ 196:

+ 197: ### 1. Initial Setup

+ 198: ```bash

+ 199: # Run the setup script

+ 200: ./setup-user-env.sh interactive

+ 201:

+ 202: # Add aliases to shell

+ 203: echo "source ~/.amazonq-jenkins/aliases.sh" >> ~/.bashrc

+ 204: source ~/.bashrc

+ 205: ```

+ 206:

+ 207: ### 2. Test Your Setup

+ 208: ```bash

+ 209: # List available jobs

+ 210: list-jobs

+ 211:

+ 212: # Test a simple trigger

+ 213: trigger-olp2

+ 214:

+ 215: # Check your team jobs

+ 216: my-jobs

+ 217: ```

+ 218:

+ 219: ### 3. Configure Amazon Q

+ 220: Add these environment variables to your Amazon Q configuration:

+ 221: ```bash

+ 222: export AMAZONQ\_JENKINS\_CONFIG="$HOME/.amazonq-jenkins"

+ 223: export JENKINS\_INTEGRATION\_ENABLED=true

+ 224: ```

+ 225:

+ 226: ## Support and Troubleshooting

+ 227:

+ 228: ### Getting Help

+ 229: - Use `aq-jenkins --help` for command help

+ 230: - Check logs in `/shared/amazonq-jenkins/logs/`

+ 231: - Contact DevOps team for permission issues

+ 232: - Refer to Jenkins documentation for job-specific help

+ 233:

+ 234: ### Common Commands

+ 235: ```bash

+ 236: # Check configuration

+ 237: cat ~/.amazonq-jenkins/credentials

+ 238:

+ 239: # View recent logs

+ 240: tail -f /shared/amazonq-jenkins/logs/triggers.log

+ 241:

+ 242: # Test connection

+ 243: curl -u $JENKINS\_USER:$JENKINS\_TOKEN $JENKINS\_BASE\_URL/api/json

+ 244: ```

+ 245:

+ 246: ## Feedback and Improvements

+ 247:

+ 248: To suggest improvements or report issues:

+ 249: 1. Create a ticket in the DevOps Jira project

+ 250: 2. Join the #amazonq-jenkins Slack channel

+ 251: 3. Submit a pull request to the integration repository

+ 252:

+ 253: ---

+ 254:

+ 255: \*\*Last Updated\*\*: July 2025

+ 256: \*\*Version\*\*: 1.0

+ 257: \*\*Maintained by\*\*: DevOps Team

⋮

↳ Purpose: Create comprehensive documentation for Amazon Q Jenkins integration

Creating: /Users/kombaths/AMAZONQ\_JENKINS\_GUIDE.md

⋮

**● Completed in 0.0s**

> ## Summary of Organization-wide Setup

I've created a complete organization-wide setup with these components:

**### 📁 Files Created:**

1. **deploy-shared-scripts.sh** - Deployment script for shared location

2. **jenkins-jobs-config.yaml** - Central job configuration

3. **config-parser.sh** - Configuration parser utility

4. **setup-user-env.sh** - Individual user environment setup

5. **AMAZONQ\_JENKINS\_GUIDE.md** - Comprehensive documentation

**### 🚀 Deployment Steps:**

1. **Deploy to Shared Location**:

bash

chmod +x deploy-shared-scripts.sh

./deploy-shared-scripts.sh

2. **Set Up Central Configuration**:

bash

cp jenkins-jobs-config.yaml /shared/amazonq-jenkins/config/

cp config-parser.sh /shared/amazonq-jenkins/scripts/

3. **Team Member Setup**:

bash

./setup-user-env.sh interactive

4. **Install Documentation**:

bash

cp AMAZONQ\_JENKINS\_GUIDE.md /shared/amazonq-jenkins/docs/

**### 🔧 Key Features:**

• \*\*Role-based access control\*\* for different teams

• \*\*Centralized job configuration\*\* with YAML

• \*\*Natural language prompt patterns\*\*

• \*\*Individual user credential management\*\*

• \*\*Comprehensive documentation\*\*

• \*\*Audit logging and security\*\*

Now your team members can use prompts like:

• "Please trigger OLP-2"

• "Run QA host file update"

• "What jobs can I run?"