DevPipeline Notifications and Webhooks Guide

Document ID: PRD-DP-035 **Last Updated:** 2024-03-01 **Owner:** DevPipeline Product Team **Classification:** Public

---

Overview

DevPipeline provides comprehensive notification and webhook capabilities to keep your team informed about pipeline events and integrate with external tools.

Notification Channels

Built-in Channels

Channel	Use Case
Email	Individual or digest notifications
Slack	Team notifications
Microsoft Teams	Team notifications
In-app	Dashboard notifications
Mobile push	DevPipeline mobile app

Webhook Channels

Channel	Use Case
Generic webhook	Custom integrations
PagerDuty	Incident management
Opsgenie	Alert management
Jira	Issue tracking
Custom HTTP	Any HTTP endpoint

Notification Configuration

Project-Level Settings

```yaml
# .devpipeline/notifications.yaml
notifications:
  channels:
    - type: slack
      webhook: ${SLACK_WEBHOOK_URL}
      channel: "#builds"
    - type: email
      recipients:
        - team@novatech.com

  events:
    pipeline_failed:
      - slack
      - email
    pipeline_success:
      - slack  # Only Slack for success
    deployment_completed:
      - slack
      - email
```

User-Level Preferences

Users can configure personal notifications at **Settings** → **Notifications**:

Notification	Default	Options
My pipeline failed	On	Email, Slack, Push
My pipeline succeeded	Off	Email, Slack, Push
Mentioned in comment	On	Email, Slack, Push
Review requested	On	Email, Slack, Push
Deployment completed	On	Email, Slack, Push

Slack Integration

Setup

1. Go to **Settings** → **Integrations** → **Slack**
2. Click **Connect Slack Workspace**

3. Authorize DevPipeline
4. Select default channel

Configuration

```yaml
notifications:
  slack:
    workspace: novatech
    default_channel: "#engineering-builds"

    channels:
      production:
        channel: "#production-deploys"
        events: [deployment_started, deployment_completed, deployment_failed]

      builds:
        channel: "#builds"
        events: [pipeline_failed]

    formatting:
      include_commit_message: true
      include_author: true
      include_duration: true
      mention_on_failure: "@channel"
```

Message Format

```
 Pipeline Failed: myapp/main #1234

Commit: Fix authentication bug
Author: jane@novatech.com
Branch: main
Duration: 5m 23s

Failed stage: test
Error: 3 tests failed

[View Pipeline] [View Logs]
```

Slack Commands

```
/devpipeline status myapp           # Get project status
/devpipeline run myapp              # Trigger pipeline
/devpipeline logs myapp 1234        # Get pipeline logs
```

```
/devpipeline subscribe #channel     # Subscribe channel
```

Email Notifications

Configuration

```
notifications:
  email:
    recipients:
      - team@novatech.com
      - leads@novatech.com

    digest:
      enabled: true
      frequency: daily
      time: "09:00"
      timezone: America/Chicago

    templates:
      failure:
        subject: "  Pipeline Failed: {{project}}/{{branch}}"
        include_logs: true
        max_log_lines: 50
```

Email Types

Type	Trigger	Default
Instant	Each event	On for failures
Digest	Scheduled summary	Off
Mention	@mentioned in comment	On

Webhooks

Generic Webhook

```
webhooks:
  - name: custom-integration
```

```yaml
    url: https://api.example.com/devpipeline-events
    events:
      - pipeline_started
      - pipeline_completed
      - pipeline_failed
      - deployment_started
      - deployment_completed
    headers:
      Authorization: Bearer ${WEBHOOK_TOKEN}
      Content-Type: application/json
    retry:
      max_attempts: 3
      backoff: exponential
```

Webhook Payload

```json
{
  "event": "pipeline_completed",
  "timestamp": "2024-07-25T14:30:00Z",
  "project": {
    "id": "proj_123",
    "name": "myapp",
    "url": "https://devpipeline.novatech.com/myapp"
  },
  "pipeline": {
    "id": "pipe_456",
    "number": 1234,
    "status": "success",
    "duration_seconds": 323,
    "url": "https://devpipeline.novatech.com/myapp/pipelines/1234"
  },
  "commit": {
    "sha": "abc123def",
    "message": "Fix authentication bug",
    "author": {
      "name": "Jane Doe",
      "email": "jane@novatech.com"
    },
    "branch": "main"
  },
  "triggered_by": {
    "type": "push",
    "user": "jane@novatech.com"
  }
}
```

Event Types

Event	Description
`pipeline_started`	Pipeline execution began
`pipeline_completed`	Pipeline finished (any status)
`pipeline_failed`	Pipeline failed
`pipeline_success`	Pipeline succeeded
`pipeline_canceled`	Pipeline was canceled
`stage_started`	Stage execution began
`stage_completed`	Stage finished
`job_started`	Job execution began
`job_completed`	Job finished
`deployment_started`	Deployment began
`deployment_completed`	Deployment finished
`deployment_failed`	Deployment failed
`approval_required`	Manual approval needed
`approval_granted`	Approval was granted
`comment_added`	Comment on pipeline

PagerDuty Integration

Setup

```yaml
webhooks:
  - name: pagerduty
    type: pagerduty
    routing_key: ${PAGERDUTY_ROUTING_KEY}
    events:
      - pipeline_failed
      - deployment_failed
    severity_mapping:
      pipeline_failed: warning
      deployment_failed: critical
    dedup_key: "{{project}}-{{branch}}"
```

Alert Format

```json
{
  "routing_key": "...",
  "event_action": "trigger",
  "dedup_key": "myapp-main",
  "payload": {
    "summary": "Pipeline failed: myapp/main #1234",
    "severity": "warning",
    "source": "devpipeline",
    "custom_details": {
      "project": "myapp",
      "branch": "main",
      "pipeline_id": "1234",
      "failed_stage": "test",
      "commit_sha": "abc123"
    }
  },
  "links": [
    {
      "href": "https://devpipeline.novatech.com/myapp/pipelines/1234",
      "text": "View Pipeline"
    }
  ]
}
```

Jira Integration

Setup

```
webhooks:
  - name: jira
    type: jira
    url: https://novatech.atlassian.net
    auth:
      type: api_token
      email: ${JIRA_EMAIL}
      token: ${JIRA_API_TOKEN}

    actions:
      pipeline_failed:
        action: create_issue
        project: ENG
        issue_type: Bug
        summary: "Pipeline failure: {{project}}/{{branch}}"
        labels: [pipeline-failure, automated]

      deployment_completed:
```

```
    action: transition_issue
    issue_key: "{{commit_message | extract_jira_key}}"
    transition: Done
```

Custom Webhook Examples

GitHub Status Check

```
webhooks:
  - name: github-status
    url: https://api.github.com/repos/novatech/{{project}}/statuses/{{commit_sha}}
    method: POST
    headers:
      Authorization: Bearer ${GITHUB_TOKEN}
    events:
      - pipeline_started
      - pipeline_completed
    payload:
      state: "{{status | github_state}}"
      target_url: "{{pipeline_url}}"
      description: "DevPipeline: {{status}}"
      context: "devpipeline/{{stage}}"
```

DataLens Metrics

```
webhooks:
  - name: datalens-metrics
    url: https://api.datalens.novatech.com/v1/metrics
    events:
      - pipeline_completed
    payload:
      metric: devpipeline.build.duration
      value: "{{duration_seconds}}"
      tags:
        project: "{{project}}"
        branch: "{{branch}}"
        status: "{{status}}"
```

Custom Deployment Tracker

```
webhooks:
  - name: deployment-tracker
```

```
    url: https://deployments.internal.novatech.com/api/v1/record
    events:
      - deployment_completed
    payload:
      service: "{{project}}"
      environment: "{{environment}}"
      version: "{{commit_sha}}"
      deployed_by: "{{triggered_by.user}}"
      deployed_at: "{{timestamp}}"
      success: "{{status == 'success'}}"
```

Filtering and Conditions

Branch Filtering

```
notifications:
  slack:
    channel: "#builds"
    events:
      - pipeline_failed
    filters:
      branches:
        include: [main, release/*]
        exclude: [feature/*]
```

Status Filtering

```
notifications:
  email:
    recipients:
      - team@novatech.com
    events:
      - pipeline_completed
    filters:
      status: [failed, canceled]  # Only notify on failure
```

Environment Filtering

```
webhooks:
  - name: production-alerts
    url: https://alerts.internal.novatech.com
```

```
events:
  - deployment_completed
filters:
  environments: [production, staging]
```

Templating

Available Variables

Variable	Description
{{project}}	Project name
{{branch}}	Branch name
{{commit_sha}}	Full commit SHA
{{commit_short_sha}}	Short commit SHA
{{commit_message}}	Commit message
{{author_name}}	Commit author name
{{author_email}}	Commit author email
{{pipeline_id}}	Pipeline ID
{{pipeline_number}}	Pipeline number
{{pipeline_url}}	Pipeline URL
{{status}}	Pipeline status
{{duration_seconds}}	Duration in seconds
{{duration}}	Human-readable duration
{{environment}}	Deployment environment
{{timestamp}}	Event timestamp

Template Functions

```
# Truncate long messages
"{{commit_message | truncate(50)}}"

# Extract Jira key
"{{commit_message | extract_jira_key}}"

# Format date
"{{timestamp | date('YYYY-MM-DD')}}"

# Conditional
"{{if status == 'success'}} {{else}} {{endif}}"
```

Testing Webhooks

Test Endpoint

```
# Send test webhook
devpipeline webhook test \
  --name my-webhook \
  --event pipeline_failed

# View recent deliveries
devpipeline webhook deliveries --name my-webhook

# Retry failed delivery
devpipeline webhook retry --delivery-id del_123
```

Webhook Logs

View webhook delivery history at **Settings** → **Webhooks** → **Deliveries**: - Request payload - Response status - Response body - Retry attempts

---

Best Practices

Notification Fatigue

1. **Use digests** for non-critical notifications
2. **Filter by branch** to reduce noise
3. **Different channels** for different severity
4. **Let users customize** their preferences

Webhook Reliability

1. **Implement idempotency** in your handlers
2. **Handle retries** gracefully
3. **Validate signatures** for security
4. **Monitor delivery** success rates

Security

1. **Use secrets** for webhook URLs and tokens
2. **Validate payloads** with HMAC signatures

3. **Use HTTPS** for all endpoints
4. **Rotate tokens** regularly

Related Documents: Getting Started (PRD-DP-001), Pipeline Configuration (PRD-DP-010), Integrations (PRD-DP-040)