**# AIM Jenkins Pipeline Documentation**

**## Overview**

This document describes the automated testing stages in the AIM (Access Identity Management) Jenkins pipeline. The pipeline includes three main automation stages that can be triggered either manually with parameters or automatically via timer.

**## Pipeline Stages**

**### 1. AIM Smoke SCIM Services**

**\*\*Purpose\*\***: Executes smoke tests for SCIM (System for Cross-domain Identity Management) services.

**\*\*Trigger Conditions\*\***:

- Manual: When `SCIM\_APIs` parameter is set to `true`

- Automatic: Timer-based execution

**\*\*Parameters\*\***:

- `Pod`: Target pod for execution (dynamic for timer triggers)

- `Env`: Environment to test against (dynamic for timer triggers)

- `Notification\_Email`: Email address for notifications

**\*\*Execution Flow\*\***:

1. Determines if triggered by timer or manual execution

2. Sets pod and environment values dynamically for timer triggers

3. Calls downstream job: `AIM\_QA/jenkins-ScimServicesSmoke-docker`

4. Captures build result and URL for reporting

**\*\*Notifications\*\***:

- **\*\*Skipped\*\***: When `SCIM\_APIs` is false

- **\*\*Success\*\***: Includes execution ID and build URL

- **\*\*Failure\*\***: Includes error message and execution ID

**### 2. Platoon Newman Automation**

**\*\*Purpose\*\***: Runs API automation tests using Newman (Postman CLI) for various services.

**\*\*Trigger Conditions\*\***:

- Manual: When `CG\_Newman\_APIs` parameter is set to `true`

- Automatic: Timer-based execution

**\*\*Supported Services\*\***:

- `enterprise-search`

- `adapt`

- `organization-scim`

- `ALL` (runs all available tests)

**\*\*Environment Mapping\*\***:

```

FIT → na-fit

DIT → na-dit

IAT → na-dc1-iat

UAT → na-dc2-uat

PROD → na-dc1-prod

```

**\*\*Environment Restrictions\*\***:

- **\*\*na-dit, na-fit, na-dc1-iat, na-dc1-prod\*\***: All services allowed

- **\*\*na-dc2-uat\*\***: Only `adapt`, `organization-scim`, and `ALL` allowed

**\*\*Parameters\*\***:

- `Env`: Target environment

- `CG\_Newman\_API\_Tests`: Comma-separated list of tests to run (default: `enterprise-search`)

- `Notification\_Email`: Email address for notifications

**\*\*Execution Flow\*\***:

1. Validates test selection against environment restrictions

2. Maps environment to Platoon naming convention

3. Calls downstream job: `AIM/aim-newman-e2e-automation/feature%2Faim\_qa`

4. Passes environment, test selection, and trigger information

**\*\*Notifications\*\***:

- **\*\*Skipped\*\***: When `CG\_Newman\_APIs` is false

- **\*\*Success/Failure\*\***: Includes executed by, test type, and environment information

**### 3. FedSSO Consent Page Automation**

**\*\*Purpose\*\***: Executes automated tests for FedSSO (Federated Single Sign-On) consent page functionality.

**\*\*Trigger Conditions\*\***:

- Manual: When `FedSSO\_CONSENT\_PAGE\_APIs` parameter is set to `true`

- Automatic: Timer-based execution

**\*\*Parameters\*\***:

- `Env`: Target environment (dynamic for timer triggers)

- `RECORD\_VIDEO`: Boolean flag to enable video recording during tests

- `Notification\_Email`: Email address for notifications

**\*\*Repository Configuration\*\***:

- **\*\*Repository\*\***: `https://bitbucket.es.ad.adp.com/scm/nexo/fedsso-myadpid-consentpage-automation.git`

- **\*\*Branch\*\***: `master`

- **\*\*Credentials\*\***: `aim-sre-cicd-user-bitbucket-token-credentials`

**\*\*Execution Flow\*\***:

1. Determines environment based on trigger type

2. Calls `consentpage()` function with configuration parameters

3. Handles video recording option if enabled

**## Common Features**

**### Timer-Based Execution**

All stages support automatic execution via Jenkins timer triggers. When triggered by timer:

- Environment and pod values are determined dynamically using `envNow()` and `podNow()` functions

- No manual parameter input required

**### Error Handling**

Each stage uses `catchError` blocks to:

- Continue pipeline execution even if individual stages fail

- Set appropriate build and stage results

- Ensure proper cleanup and notification

**### Dynamic Parameter Resolution**

The pipeline intelligently switches between:

- **\*\*Manual execution\*\***: Uses provided parameters

- **\*\*Timer execution\*\***: Uses dynamic functions to determine current values

**## Best Practices**

1. **\*\*Environment Validation\*\***: Always validate test compatibility with target environments

2. **\*\*Notification Management\*\***: Ensure notification emails are properly configured

3. **\*\*Error Monitoring\*\***: Review failure notifications and build logs for troubleshooting

4. **\*\*Resource Management\*\***: Monitor pod and environment resource usage during execution

**## Troubleshooting**

**### Common Issues**

- **\*\*Environment Restrictions\*\***: Verify test selection is allowed for target environment

- **\*\*Credential Issues\*\***: Ensure repository access credentials are valid

- **\*\*Resource Availability\*\***: Check pod and environment availability before execution

**### Log Locations**

- Build logs available at: `${env.BUILD\_URL}`

- Downstream job logs available at respective job URLs

- Email notifications contain direct links to failed builds