**Work Plan: Gen-2 to Gen-3 Spring Batch Application Migration**

Work plan for migrating a Spring Batch application from a Gen‑2 stack (Java 8, Spring Boot 2.x, iBATIS) to a Gen‑3 stack (Java 17, Spring Boot 3.x, MyBatis).

**1. Prerequisites**

Before starting,

* **Access & Permissions**
  + Git repository access (read/write) for the Dvl‑Aws branch.
  + Access to the shared templates folder (contains BsPrdTmpltClodBtchDb & BsPrdTmpltClodBtchFile).
* **Environment Setup**
  + **Java 17** installed and configured (java -version outputs 17.x).
  + **IDE** (Eclipse/IntelliJ) configured to use Java 17 for both Build Path and Compiler Compliance.
  + **Gradle 8.3** wrapper available (will be configured in step 3).
* **Knowledge & Documentation**
  + Spring Boot 2→3 migration guides (Jakarta EE namespace changes).
  + Understanding of MyBatis vs iBATIS mappings.
  + Access to existing Gen‑2 application documentation and source.
* **CI/CD & Infrastructure**
  + Access to CI/CD pipeline

**2. High-Level Process**

The migration follows these phases:

1. **Preparation**: Branching, environment validation, and baseline verification.
2. **Gen‑2 Analysis & Setup**: Confirm current behavior, database/table structures, scheduled triggers.
3. **Gen‑3 Template Preparation**: Copy and configure the appropriate base template.
4. **Code Migration**: Upgrade build scripts, refactor DAOs, configure MyBatis, and batch jobs.
5. **Testing & Validation**: Unit, integration, and manual environment tests.
6. **CI/CD Integration**: Pipeline adjustments, staging deployments.

**3. Gen‑2 Application Setup**

**3.1 Checkout & Baseline**

# Clone and switch to the Gen‑2 branch

git clone git@github.com:xxxxx/SmnBtchApcnRpt.git

cd SmnBtchApcnRpt

git checkout Dvl-Aws

**3.2 Verify Baseline**

* Build and run the existing application under the dvl profile:
* ./gradlew clean build

java -jar build/libs/SmnBtchApcnRpt.jar --spring.profiles.active=dvl

* Confirm expected logs and database updates.

**3.3 Document Current Architecture**

* Identify:
  + Job names and schedules (cron, scheduler).
  + DAO classes, iBATIS XML mappers, stored procedures.
  + Domain classes and data flows.
  + External dependencies (REST calls, file I/O).
* Create a simple flow diagram or sequence chart (optional) to capture the existing batch logic.

**4. Migration to Gen‑3**

**4.1 Branch & Template Selection**

1. Create feature branch:

git checkout -b feature/migrate-to-gen3

1. Decide template:
   * **DB‑writing** jobs → BsPrdTmpltClodBtchDb.
   * **File‑based** jobs → BsPrdTmpltClodBtchFile.
2. Copy template locally:
3. cp -r ~/templates/BsPrdTmpltClodBtchDb ./SmnBtchApcnRpt-Gen3

cd SmnBtchApcnRpt-Gen3

**4.2 Build System Upgrade**

* **Gradle Wrapper** (gradle-wrapper.properties):

distributionUrl=https://services.gradle.org/distributions/gradle-8.3-bin.zip

* **build.gradle** modifications:
* - id 'org.springframework.boot' version '2.x.x'
* + id 'org.springframework.boot' version '3.2.0'
* - sourceCompatibility = '1.8'
* + java { sourceCompatibility = JavaVersion.VERSION\_17; targetCompatibility = JavaVersion.VERSION\_17 }
* - implementation 'com.ibatis:ibatis-sqlmap:2.x.x'

+ implementation 'org.mybatis.spring.boot:mybatis-spring-boot-starter:3.1.2'

* **Reload** the Gradle project and confirm compilation under JDK 17.

**4.3 Configuration Files**

* Under src/main/resources/config/, create:
  + SmnBtchApcnRpt-dvl.properties
  + SmnBtchApcnRpt-int.properties
  + SmnBtchApcnRpt-prd.properties
  + SmnBtchApcnRpt.properties (default)
  + log4j2-dvl.properties, etc.
* Populate with environment‑specific database URLs, credentials, file paths, and batch parameters (chunk size, step names).

**4.4 Domain & DAO Migration**

1. **Domain Classes**: Copy POJOs into com.delta.crew.domn.
2. **Mapper Interfaces** in com.delta.crew.dao:
3. **MyBatisConfig.java**:

**4.5 Batch Job Definition**

* **Tasklet (DB)**:

**4.6 Testing & Validation**

* **Unit Tests**
* **Integration Tests**
* **Manual Run**

**4.7 CI/CD & Deployment**

**5. Definition of Done**

The migration is considered complete when all of the following criteria are met:

|  |  |
| --- | --- |
| **Criteria** | **Verification Method** |
| **Local build & run** | ./gradlew clean build → success; bootRun logs. |
| **Unit & integration tests** | All tests pass with ≥ 80% coverage. |
| **Environment configs** | DVL/INT/PRD properties load correctly. |
| **Batch job functionality** | Pending records processed correctly in INT & PRD. |
| **CI pipeline green** | Jenkins/GitHub Actions build, test, and deploy pass. |
| **Documentation updated** | README/Confluence updated with new steps & diagrams. |
| **Stakeholder sign-off** | Email or Jira approval from batch owners. |