

Startup Procedure Document

Startup Procedure Document

Project: Identity Governance Lifecycle Management (IGLM)

Version: 1.0

Date: April 28, 2025

Owner: Platform Engineering Team

Objective

This document outlines the structured and repeatable startup sequence to initialize the Identity Governance Lifecycle Management (IGLM) platform services, prepare underlying databases, execute required database migrations, seed configuration data, and start the application server.

Prerequisites

- Docker and Docker Compose installed and configured on the host system.
- Node.js and `npm` utilities available in the execution environment.
- Project repository cloned locally and dependencies installed (`npm install`).

Startup Procedure Document

- Network access and sufficient permissions to connect to PostgreSQL, Redis, RabbitMQ, and Consul services.

Startup Procedure

1. Shutdown Any Existing Containers

Command:

```
docker compose -f deploy/docker-compose.yml down -v --remove-orphans
```

2. Start Core Infrastructure Services

Command:

```
docker compose -f deploy/docker-compose.yml up -d hrms-db config-db joiner-db mover-db leaver-db provisioning-db  
redis rabbitmq consul
```

3. Initialize PostgreSQL Extensions

Command:

```
docker exec -it iglm-config-db psql -U postgres -d idcs_dev
```

Once connected, run:

```
CREATE EXTENSION IF NOT EXISTS "uuid-oss";
```

Startup Procedure Document

4. Database Migration Management

Commands:

```
npx sequelize-cli db:migrate:undo --env development
```

```
npx sequelize-cli db:migrate --env development
```

```
npx sequelize-cli db:migrate --env development_idcs
```

5. Seed Configuration Data

Commands:

```
node scripts/setupConfigs.js
```

```
node scripts/setupApplicationDiscoveryConfigs.js
```

6. Reset Redis Cache

Command:

```
docker exec -it iglm-redis redis-cli -a Secured3211 FLUSHALL
```

7. Launch the Application Server

Command:

Startup Procedure Document

node src/server.js

Post-Startup Validation Checklist

- All Docker containers are healthy and operational (docker ps).
- Database connections established successfully.
- Redis cache cleared and accessible.
- RabbitMQ queues and exchanges initialized (validate via RabbitMQ Management Console).
- Consul services registered and visible.
- Application server logs indicate successful startup.

Additional Notes

- Any failure during migration or script execution must be remediated before proceeding to application startup.
- Always validate environment-specific `.env`` files prior to startup.
- Ensure time synchronization across services to prevent token/session validation issues.

Startup Procedure Document

End of Document