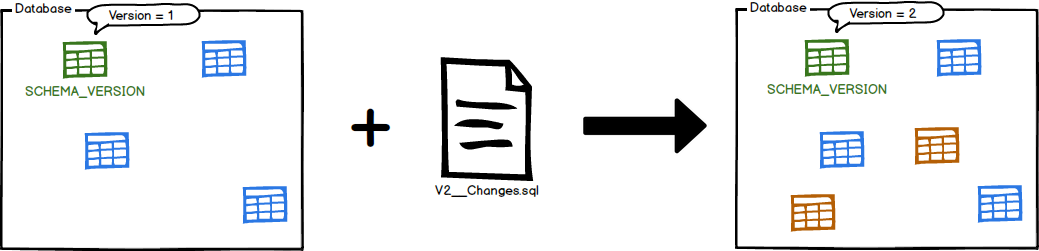
**It is based around just 6 basic commands:**

* Migrate,
* Clean,
* Info,
* Validate,
* Baseline and
* Repair.

# Migrate

Migrates the schema to the latest version. Flyway will create the metadata table automatically if it doesn't exist.

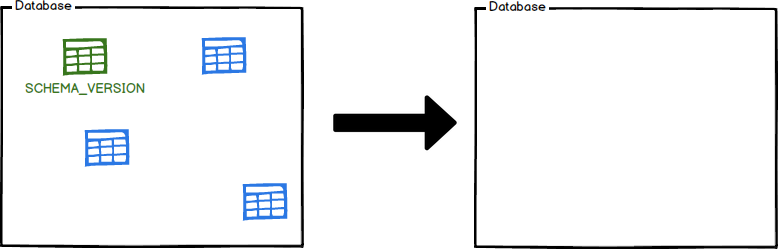


Migrate is the centerpiece of the Flyway workflow. It will scan the filesystem or your classpath for available migrations. It will compare them to the migrations that have been applied to the database. If any difference is found, it will migrate the database to close the gap.

Migrate should preferably be executed on application startup to avoid any incompatibilities between the database and the expectations of the code.

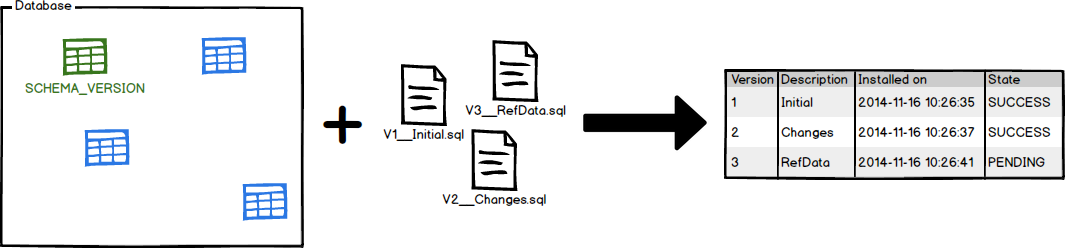
**Clean**

Drops all objects in the configured schemas.



# Info

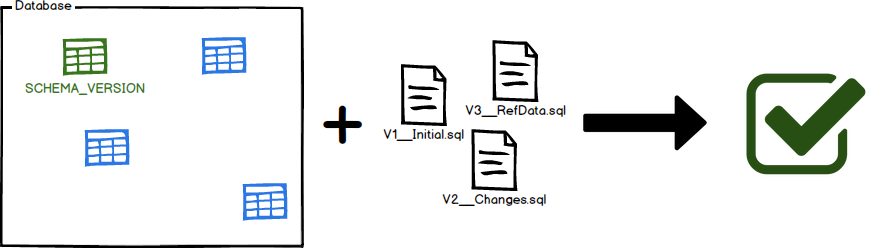
Prints the details and status information about all the migrations.



info lets you know where you stand. At a glance you will see which migrations have already been applied, which other ones are still pending, when they were executed and whether they were successful or not.

# Validate

Validates the applied migrations against the available ones.

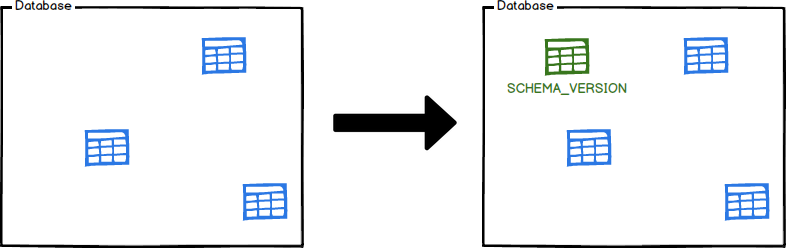


Validate helps you verify that the migrations applied to the database match the ones available locally.

This is very useful to detect accidental changes that may prevent you from reliably recreating the schema.

# Baseline

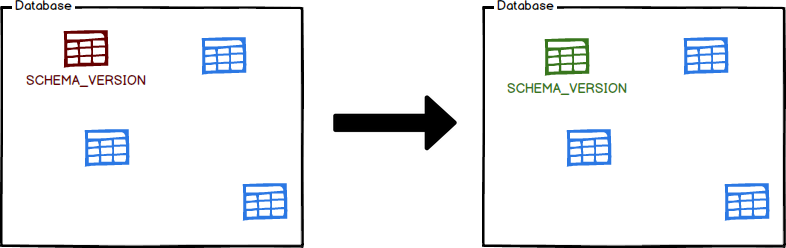
Baselines an existing database, excluding all migrations upto and including baselineVersion.



Baseline is for introducing Flyway to existing databases by baselining them at a specific version. The will causeMigrate to ignore all migrations upto and including the baseline version. Newer migrations will then be applied as usual.

# Repair

Repairs the metadata table



Repair is your tool to fix issues with the metadata table. It has two main uses:

* Remove failed migration entries (only for databases that do NOT support DDL transactions)
* Realign the checksums of the applied migrations to the ones of the available migrations

**First Steps: Maven**

**Prerequisites**

* Java 6+
* A working Maven 2 or 3 install

**Creating the project**

We're going to create our project using the Maven Archetype Plugin by issuing the following command:

> mvn archetype:generate -B

-DarchetypeGroupId=org.apache.maven.archetypes

-DarchetypeArtifactId=maven-archetype-quickstart

-DarchetypeVersion=1.1

-DgroupId=foo

-DartifactId=bar

-Dversion=1.0-SNAPSHOT

-Dpackage=foobar

We are now ready to get started. Let's jump into our project:

> cd bar

**Integrating Flyway**

**<plugin>**

**<groupId>org.flywaydb</groupId>**

**<artifactId>flyway-maven-plugin</artifactId>**

**<version>3.2.1</version>**

**<configuration>**

**<driver>com.mysql.jdbc.Driver</driver>**

**<url>jdbc:mysql://localhost:3306/mydb</url>**

**<user>root</user>**

**<password>root</password>**

**</configuration>**

**</plugin>**

**Creating the first migration**

We create the migration directory src/main/resources/db/migration.  
Followed by a first migration called src/main/resources/db/migration/V1\_\_Create\_person\_table.sql:

create table PERSON (

ID int not null,

NAME varchar(100) not null );

**Migrating the database**

It's now time to execute Flyway to migrate our database:

bar> mvn compile flyway:migrate

If all went well, you should see the following output:

[INFO] Creating Metadata table: "PUBLIC"."schema\_version"

[INFO] Current version of schema "PUBLIC": << Empty Schema >>

[INFO] Migrating schema "PUBLIC" to version 1

[INFO] Successfully applied 1 migration to schema "PUBLIC" (execution time 00:00.062s).

**Adding a second migration**

If we now add a second migration called src/main/resources/db/migration/V2\_\_Add\_people.sql:

insert into PERSON (ID, NAME) values (1, 'Axel');

insert into PERSON (ID, NAME) values (2, ‘Mr. Foo’);

insert into PERSON (ID, NAME) values (3, ‘Ms. Bar');

and execute it by issuing:

> mvn compile flyway:migrate

We now get:

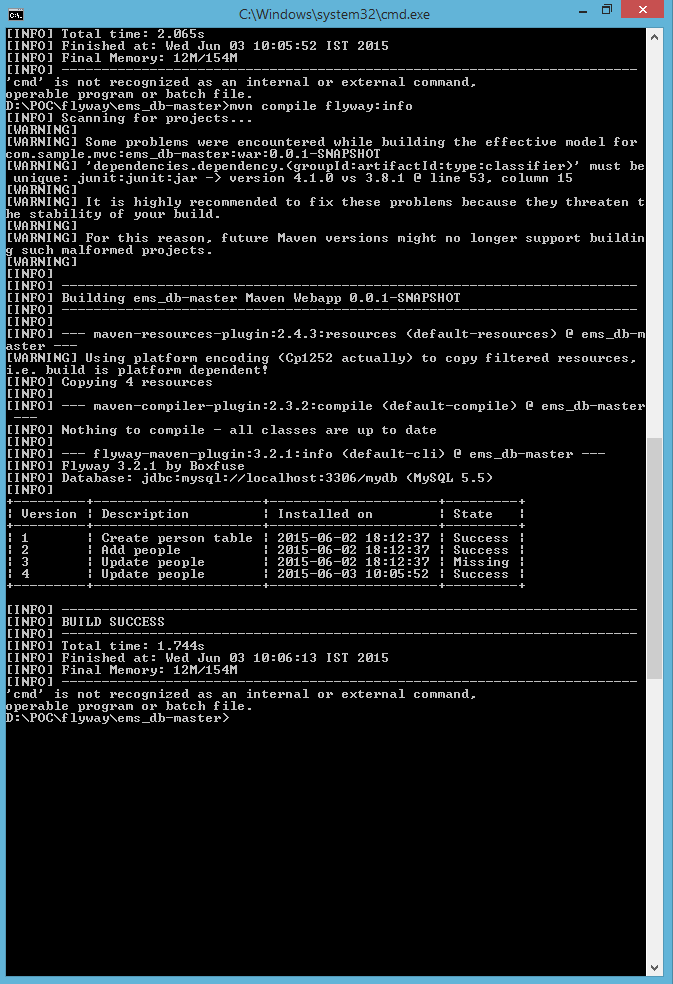
[INFO] Current version of schema "PUBLIC": 1

[INFO] Migrating schema "PUBLIC" to version 2

[INFO] Successfully applied 1 migration to schema "PUBLIC"

**Mvn compile flyway:migrate**



**Mvn compile flyway:info** 

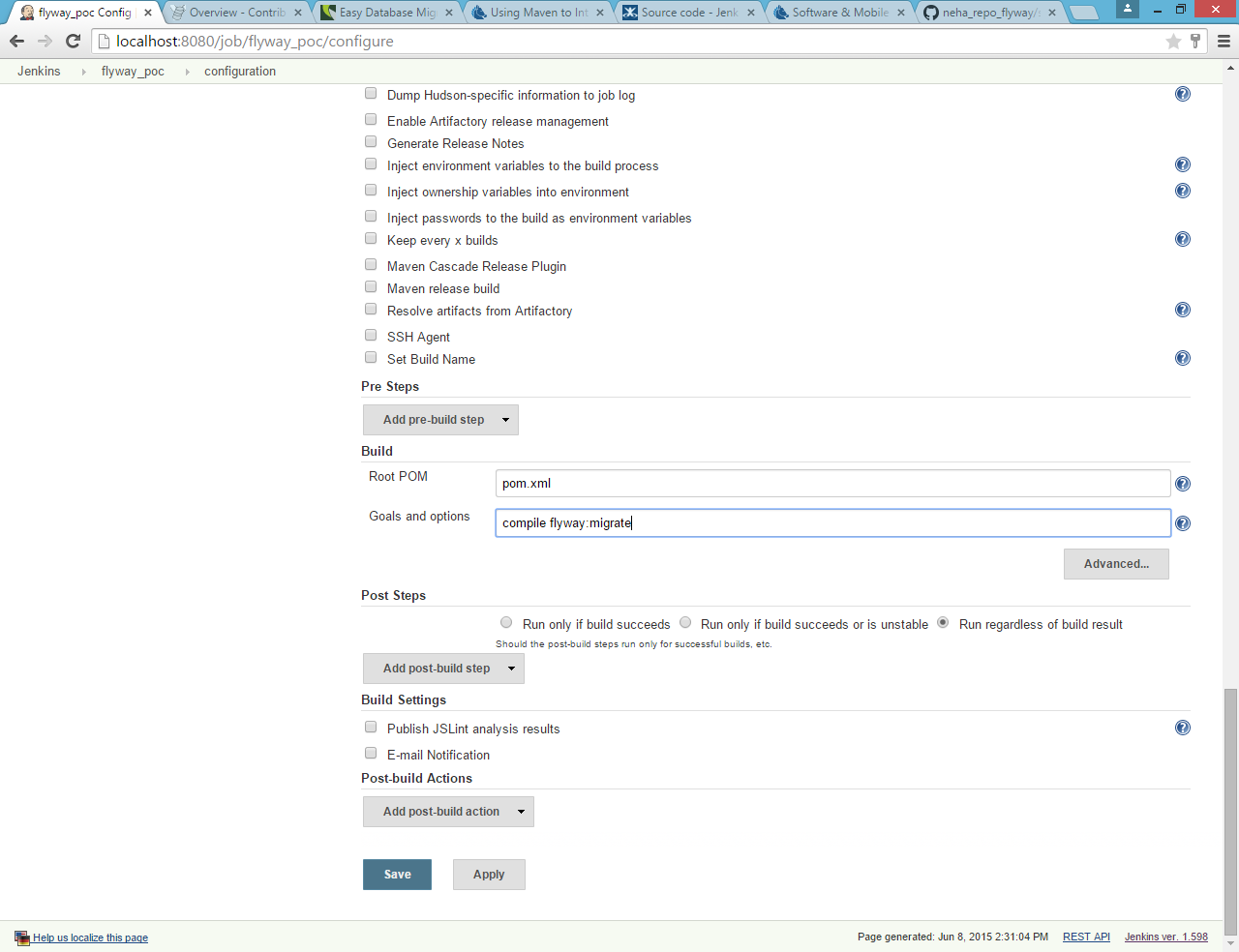
## How do you repair the database after a failed migration?

If your database supports DDL transactions, Flyway does the work for you.

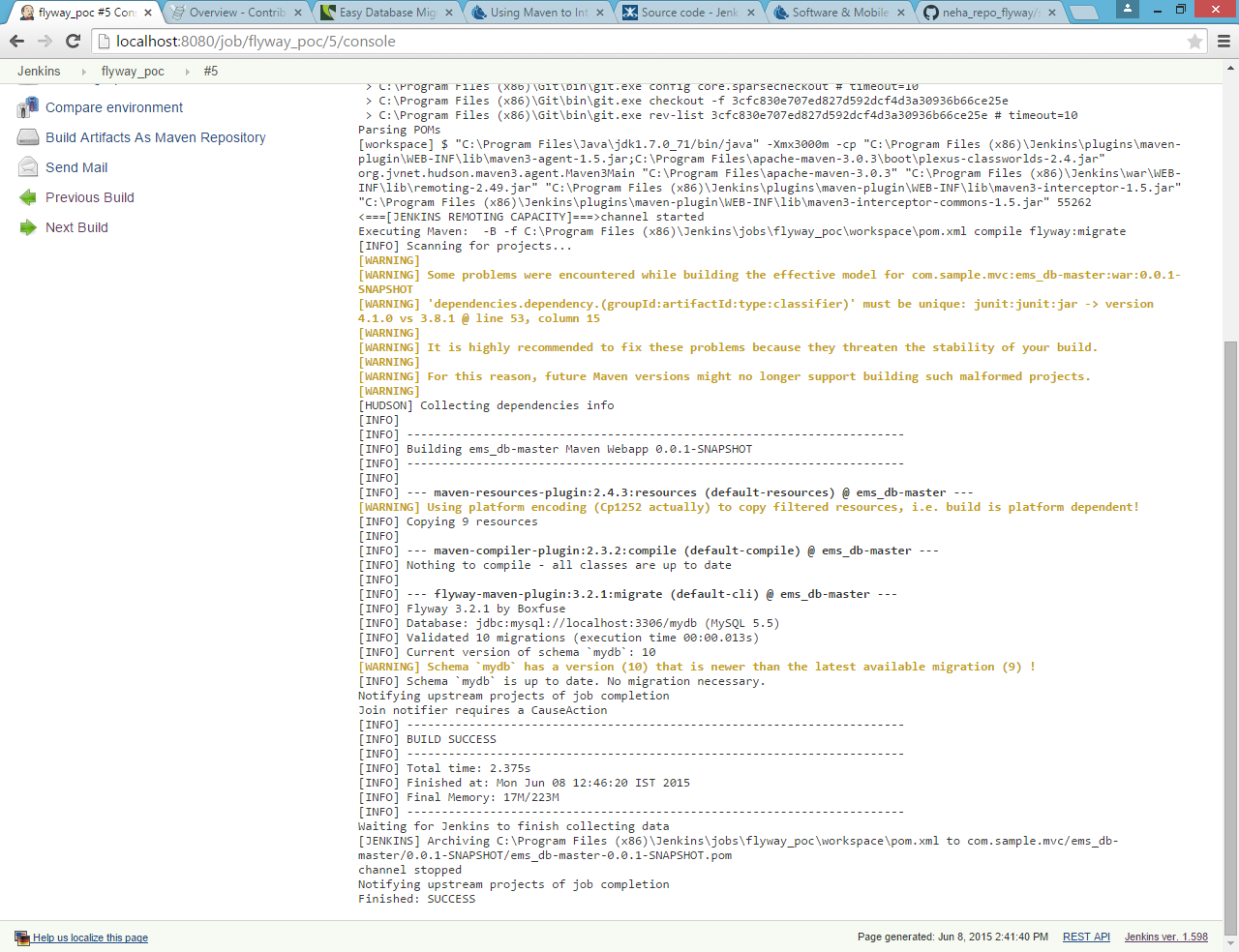
If your database doesn't, these are the steps to follow:

1. Manually undo the changes of the migration
2. Invoke the repair command
3. Fix the failed migration
4. Try again

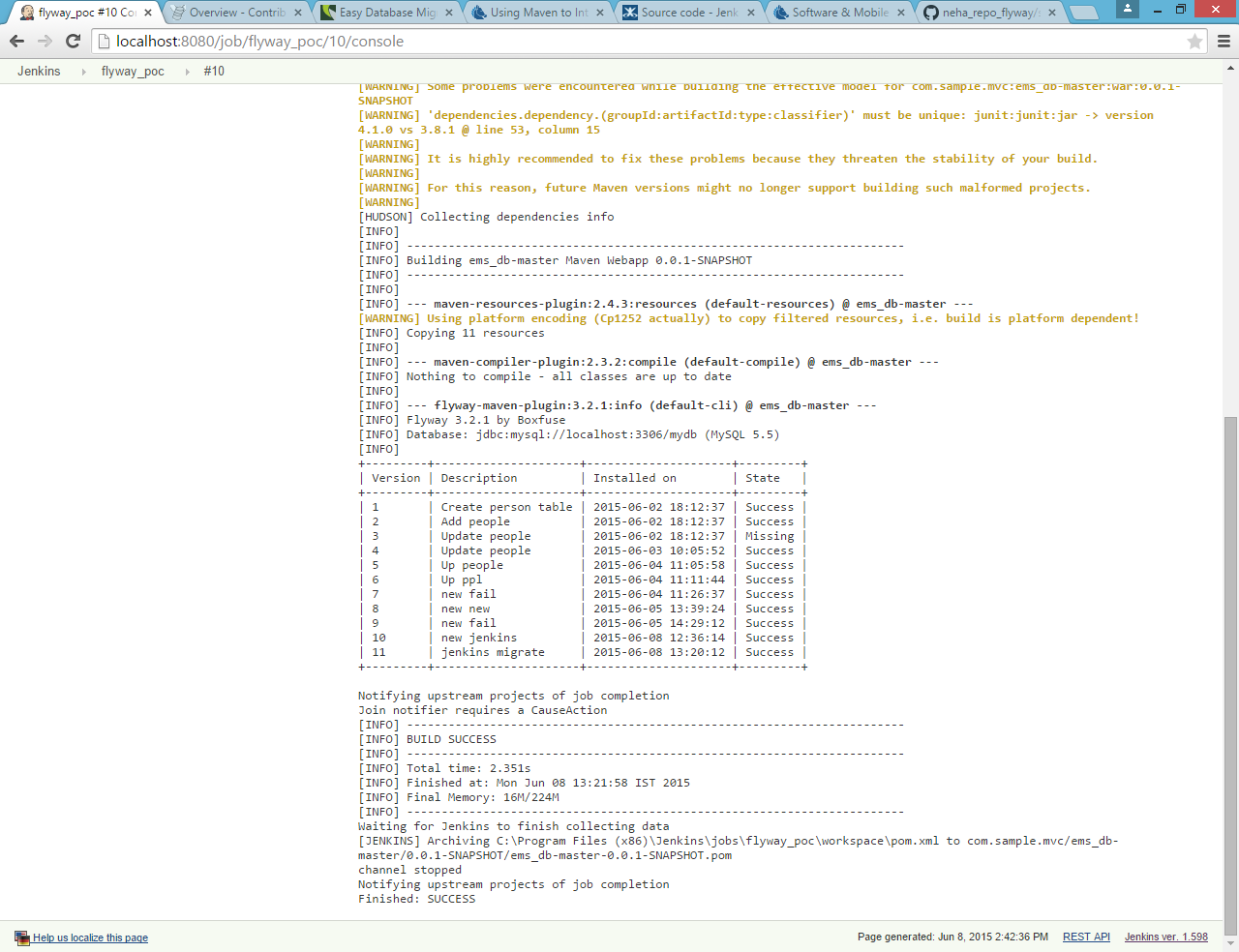
**Integration with Jenkins:**



**Run the build**



**Mvn compile flyway:info**



**Advantages**

Developers no longer need to remove the entire test database in order to create a new test database from scratch (e.g. using schema creation scripts from DDL generation tools). Further, if generation of test data costs a lot of time, developers can avoid regenerating test data for small, non-destructive changes to the schema.

## Available Tools

* **Flyway** - database migration tool (for Windows, OSX, Linux, Android and the JVM) where migrations are written in SQL or Java
* **LiquiBase** - cross platform tool where migrations are written in XML, YAML, JSON or SQL.
* **Datical** - Enterprise commercial version of Liquibase.
* **Active Record (Migrations)** - schema migration tool for Ruby on Rails projects based on Active Record.
* **Ruckusing**-migrations - schema migration tool for PHP projects.
* **Phinx** - another framework-independent PHP migration tool.
* **MyBatis Migrations** - seeks to be the best migration tool of its kind.
* **Ragtime** - a SQL database schema migration library written in Clojure
* **Lobos** - a SQL database schema manipulation and migration library written in Clojure.
* **Alembic** - a lightweight database migration tool for usage with the SQLAlchemy Database Toolkit for Python.
* **RoundhousE** - a SQL database versioning and change management tool written in C#.
* **XMigra** - a SQL database evolution management tool written in Ruby that generates scripts without communicating with the database.

**Source:**

<http://flywaydb.org>

<http://en.wikipedia.org/wiki/Flyway>

<http://techieindescribable.blogspot.in/2013/08/comparison-between-flyway-and-liquibase.html>

<http://www.lordofthejars.com/2011/12/far-away-long-ago-glowing-dim-as-ember.html>