
LF
NETWORKING
LFN Developer & Testing Forum

LF NETWORKING

LFN Developer & Testing Forum

Policy Framework

Database Upgrade and Rollback

Overview

- Overview
  - Issues we were trying to solve
  - Research into database schema creation options
  - Policy DB Migrator components
  - Policy DB Migrator features
  - Demo

Issues we were trying to solve

- This functionality will allow the end user to easily upgrade or downgrade the policy framework version they want to use without the need to manually alter database objects.
- It also prevents errors caused by java connection issues.

Research into database migration tools

- Licensing issues with Liquibase prevented us from using it in ONAP
- Similar migration tool was already available in Policy drools.
- This formed the base of Policy DB migrator.

Policy DB migrator components

- Policy DB migrator is made up of 2 main components
  - Shell script
  - SQL files

Shell script

- The main engine of Policy DB migrator is a shell script which connects to the database and runs the SQL files
- It also creates a log record of each action performed.

SQL files

- SQL files are divided into directories specific to the release they belong to
- Further sub divided into upgrade and downgrade directories depending on the script function.
- SQL files are sequenced to ensure correct running order

Automatic Schema detection

- Policy DB Migrator Features:
  - Automatic schema detection
  - Point of failure detection and resolution
  - Logging

Automatic Schema detection

- Script will automatically detect if the policy admin schema is empty.
- If it is, it will run a full install
- Otherwise it will set the version to 0800 (Honolulu) and upgrade to 0900 (Istanbul)

Partial Upgrade/downgrade

- Db migrator allows for partial upgrades or downgrades in the case of failure
- Users can fix the problem that caused the failure and continue with the upgrade or downgrade or just rollback their changes

Logging

- Policy DB migrator script output can be seen in the k8s logs. This will include the output from each SQL file run and a summary report.
- Policy DB migrator also persists logging information in the database. There are 2 tables in the migration schema.
  - schema_versions table stores the current schema version.
  - policyadmin_schema_changelog table stores a sequential list of the SQL files run, their timestamp and status.

How It Works

MariaDB is started and required databases /users are created if necessary

Database Initialization

Policy DB Migrator

Policy DB migrator script starts

Check the version to be installed

Current state of policyadmin schema is checked to determine which scripts need to be run

- If tables already exist in the policy schema the current version is set to 0800
- Check the policy db migrator log table to see if the previous run failed. This means a partial install is required
- If the target version is less than or equal to the current version exit

Run the required SQL files

All files are run unless an error is encountered

Success/Failure code is returned to k8s

Benefits

- Separating database table creation will allow for more control of the schema layout. This will enable us to implement a more normalized data model going forward.
- This should help us with transaction processing and data consistency.

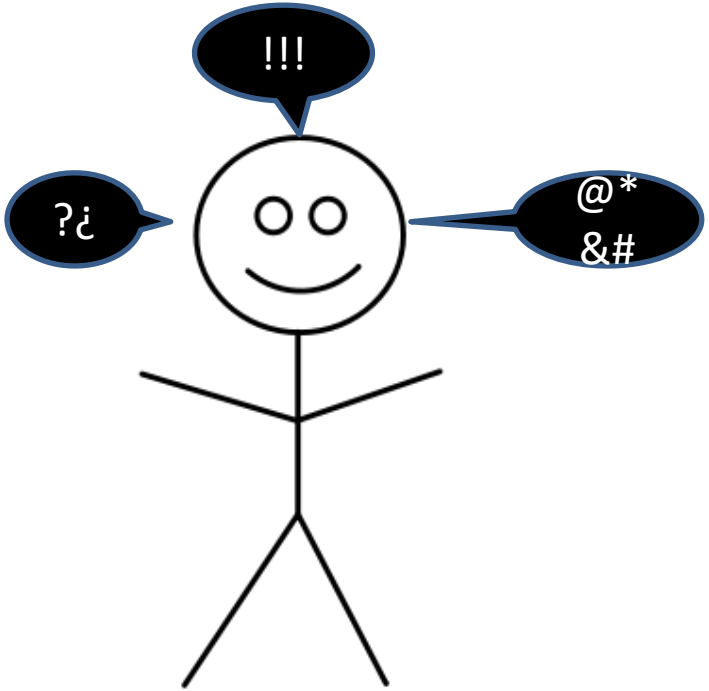
Looking ahead

- More database related work planned in future releases:
  - making control loop/policy framework database agnostic to allow user to choose between mariadb and postgres
  - Investigating high availability options
  - Integrating back up and restore

Links

- Using Policy DB Migrator — onap master documentation

Question Or Comments



Thank You

LFN NETWORKING
LFN Developer & Testing Forum

Thanks to:

Policy Framework Committers:
Jim Hahn
Pamela Dragosh
Jorge Hernandez Herrero
Ram Krishna Verma
Ajith Sreekumar

Developers:
Bruno Militzer
Francesco Fiora
Liam Fallon
Ramesh Murugan Iyer
Saul Gill
Sirisha Manchikanti
Wayne Dunican
Francesco Lapenta
Adheli Tavares
Kevin Timoney