Evolveum

MidPoint 4.4 – Upgrade and migration

Anton Tkáčik / January 2022

Agenda

- Short overview of changes

- How to upgrade to midPoint 4.4

- How to migrate to PostgreSQL native repository

Changes since 4.0 affecting upgrade

Ev⊙lveum

- Java 8 platform is no longer supported

  - Java 11 or Java 17 is recommended

- MySQL and MariaDB are no longer supported,

  - there is upgrade path to 4.4 and then migration to native repository

- Many deprecated elements were removed from schema

- Jasper-based reports are no longer supported
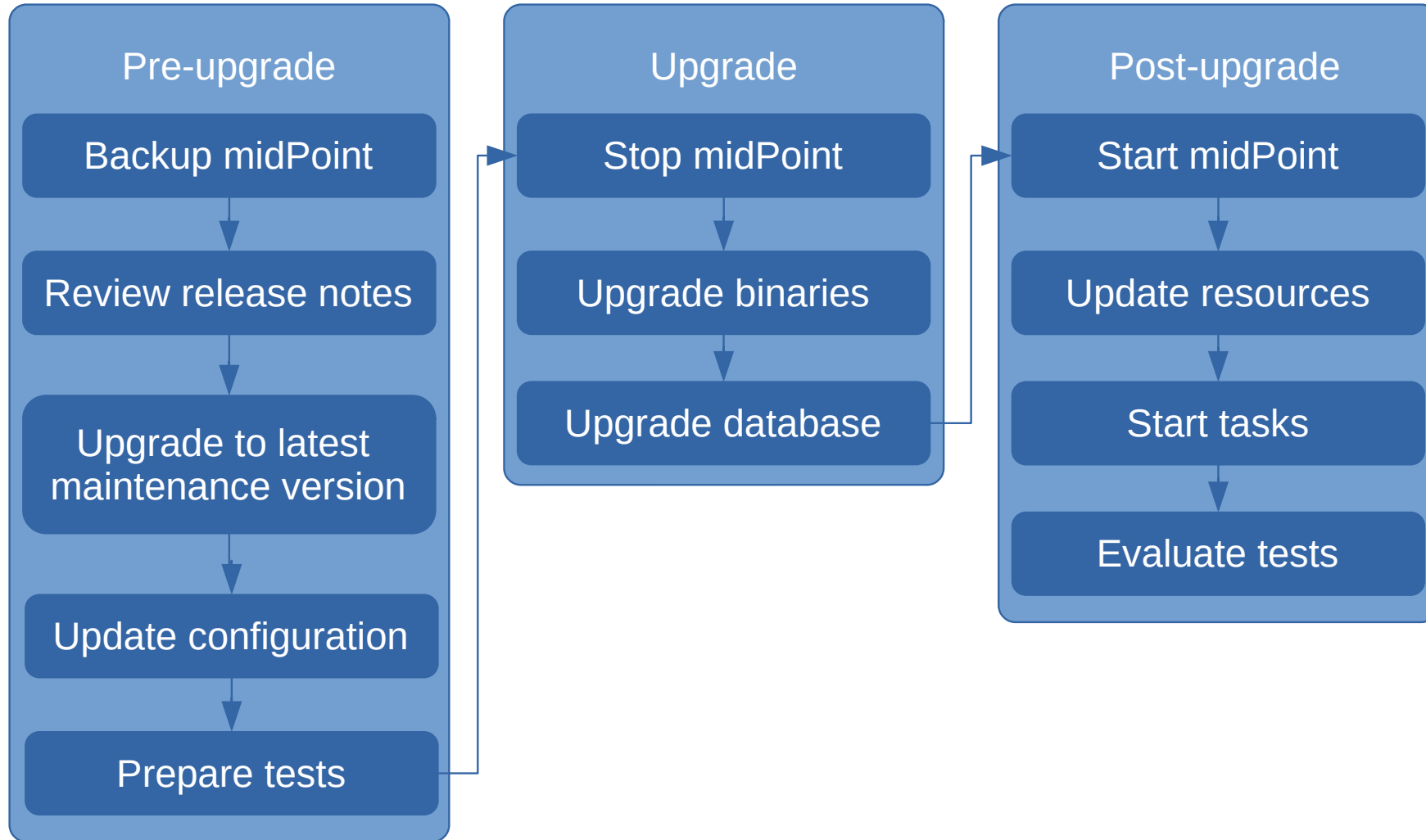
- Tasks should be reimported / updated

How to upgrade to midPoint 4.4

How to upgrade to midPoint 4.4

- Supported upgrade paths are:

    - midPoint 4.0.4 to midPoint 4.4.1

    - midPoint 4.3.2 to midPoint 4.4.1

    - midPoint 4.4.1 with generic repository to midPoint 4.4.1 PostgreSQL Native repository

- Upgrade process may require changes in your tasks, mappings, etc. so it is strongly recommended to perform upgrade in testing / staging environment first to get a good estimate of changes required and time needed.

Upgrade Process Overview

Pre-upgrade
- Backup midPoint
- Review release notes
- Upgrade to latest maintenance version
- Update configuration
- Prepare tests

Upgrade
- Stop midPoint
- Upgrade binaries
- Upgrade database

Post-upgrade
- Start midPoint
- Update resources
- Start tasks
- Evaluate tests

Pre-upgrade steps

- Download 4.4.1 release

    https://evolveum.com/download/4-4-lts-tesla/

- Create fresh backup of your midPoint and database

- Review release notes for 4.4.1

    - Pay attention to changes, upgrade sections and deprecated / removed features

- Upgrade to latest maintenance version

    - Latest maintenance version contains pre-upgrade checks (e.g. deprecations)

Pre-upgrade steps

- All paths are relative to midPoint installation (home) directory

    - Usually this is /opt/midpoint

- Use ninja tool to check for deprecated / removed functionality

    ```
    bin/ninja.sh verify -w plannedRemoval
    ```

- If you are using functionality planned for removal adjust system configuration to not use that functionality (if possible)

    - Data in removed elements are not accessible after upgrade, if you need to remap them to other properties, this needs to happen before upgrade.

Ninja verify

- Some of removed schema elements

  - **objectType, employeeType, roleType, orgType and serviceType**. The functionality was replaced by **subtype** property, which was later replaced by archetypes.

  - **subtype** property still remains, however it is deprecated. It is strongly recommended migrating all object subtyping functionality to archetypes.

- More exhaustive list is present in release notes

Stop the system

- Suspend all scheduled and automatic tasks

  - this prevents any unexpected behavior which may happen after upgrade

- Shutdown midPoint server processes

  - in cluster deployment all nodes must be shut down

Upgrade binaries

- Usual location of midPoint installation is **/opt/midpoint**

- Replace binaries with new binaries from distribution package

    - **bin** and **lib** directories must be replaced

    - it is strongly recommended to upgrade **doc** and **README**

- **var** directory should be left untouched, since it contains deployment specific files

- Make sure all cluster nodes contains same upgraded binaries

Upgrade binaries


```
tony@tdm-box:/opt/midpoint$ mkdir backup
tony@tdm-box:/opt/midpoint$ ls
backup/      doc/         lib/           LICENSE.AL    NEWS        README        tmp/
bin/         INSTALL      LICENSE        LICENSE.EUPL  NOTICE      RELEASE-NOTES var/
tony@tdm-box:/opt/midpoint$ mv -t backup bin lib doc README RELEASE-NOTES
tony@tdm-box:/opt/midpoint$ ls
backup  INSTALL  LICENSE  LICENSE.AL  LICENSE.EUPL  NEWS  NOTICE  tmp  var
tony@tdm-box:/opt/midpoint$ mv -t . tmp/midpoint-4.4.1/{bin,lib,doc,README,RELEASE-NOTES}
tony@tdm-box:/opt/midpoint$ ls
backup  bin  doc  INSTALL  lib  LICENSE  LICENSE.AL  LICENSE.EUPL  NEWS  NOTICE  README  RELEASE-NOTES  tmp  var
tony@tdm-box:/opt/midpoint$
```

Upgrade database structure

- Upgrade scripts are located in **doc/config/sql/generic-old**

  - Naming pattern is database type, original version – target version
    (eg. postgresql-upgrade-4.0-4.4.sql)

- Use appropriate tool for your database to run these scripts

  ```
  PostgreSQL:
  psql --host=localhost --username=midpoint
  midpoint -f doc/config/sql/postgresql-upgrade-
  4.0-4.4.sql
  ```

Upgrade database structure

Evolveum


```
tony@tdm-box:/opt/midpoint$ cd doc/config/sql/generic-old/
tony@tdm-box:/opt/midpoint/doc/config/sql/generic-old$ ls
h2-upgrade-4.0-4.4.sql     mysql-upgrade-4.0-4.4-utf8mb4.sql  oracle-upgrade-4.3-4.4.sql     postgresql-4.4-all.sql
h2-upgrade-4.3-4.4.sql     mysql-upgrade-4.3-4.4.sql          oracle-4.4-all.sql             sqlserver-upgrade-4.0-4.4.sql
h2-4.4-all.sql            mysql-upgrade-4.3-4.4-utf8mb4.sql   postgresql-upgrade-4.0-4.4.sql  sqlserver-upgrade-4.3-4.4.sql
mysql-upgrade-4.0-4.4.sql  oracle-upgrade-4.0-4.4.sql         postgresql-upgrade-4.3-4.4.sql  sqlserver-4.4-all.sql
tony@tdm-box:/opt/midpoint/doc/config/sql/generic-old$ psql --host=localhost --username=midpoint webinar40 -f postgresql-upgrade-4.0
-4.4.sql
Password for user midpoint: 
```
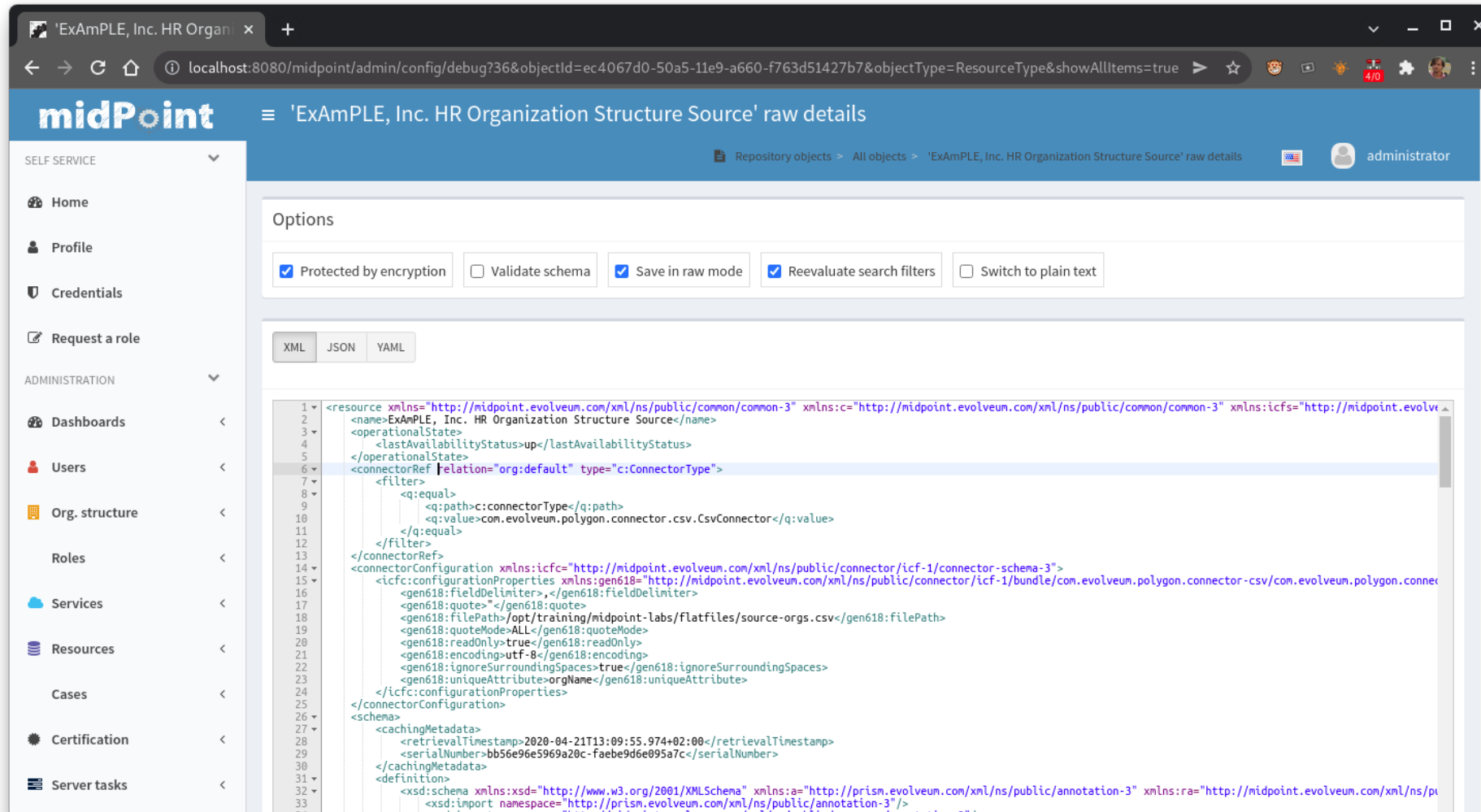
Post-upgrade steps

- Start upgraded midPoint and monitor midPoint logs for any errors

- 4.4.1 upgrade provides newer versions of connector, so resources are probably not working

- In 4.4.1 some of the initial objects changed,but this changes are not applied automatically if you have older version

Post-upgrade steps

- Upgrade Resources to point to new connector versions

    - delete old connector objects in midPoint repository

- Review changes in initial objects and default system configuration

    - Initial objects are located in doc/config/initial-objects

- If your upgrade path was upgrade to 4.4, switch to native repository, then upgrade to 4.4.1 with Native repository, you **need to run reindex task** to ensure shadow attributes indexes are constructed properly.

Upgrade connectors to point to new connector versions

Review changes in existing imported initial objects

Update of server tasks

- Deleting working state of server tasks is recommended

- Update server tasks definitions if necessary

  - Apply server task archetypes if missing

  - Configuration of advanced distribution features was changed and needs to be migrated

    - e.g. buckets, worker tasks, and task partitions

    - See https://docs.evolveum.com/midpoint/reference/tasks/activities/migration/

      and midPoint Studio, which provides tooling for tasks conversion.

Post-upgrade testing

- Before testing we recommend to restart system

- Evaluate updated system and run your post-upgrade tests

  - Monitor logs for possible errors

- If there are no critical errors, your system is ready for use.

Follow-up steps

Evolveum

- Review release notes again and use **ninja verify** to determine which functionality is deprecated and you may need to plan migration to newer mechanisms.

- Consider and plan for migration to PostgreSQL native repository.

How to migrate to PostgreSQL Native repository

Migration to PostgreSQL Native Repository

- Prerequisites

    - midPoint updated to 4.4.1

    - PostgreSQL database of version 13 and later

- Familiarize yourself with Native Repository design

- Decide if midPoint and midPoint audit database is same or will be two different databases

- Repository migration process should be treated similar as upgrade process with separate testing in test environment, planned downtime

Migration to PostgreSQL Native Repository

Evolveum

Pre-migration

- Initialize database
- Verify correct OIDs
- Prepare tests
- Shutdown midPoint

Migration

- Export data & audit
- Change configuration
- Import data & audit

Post-migration

- Start midPoint
- Start tasks
- Evaluate tests
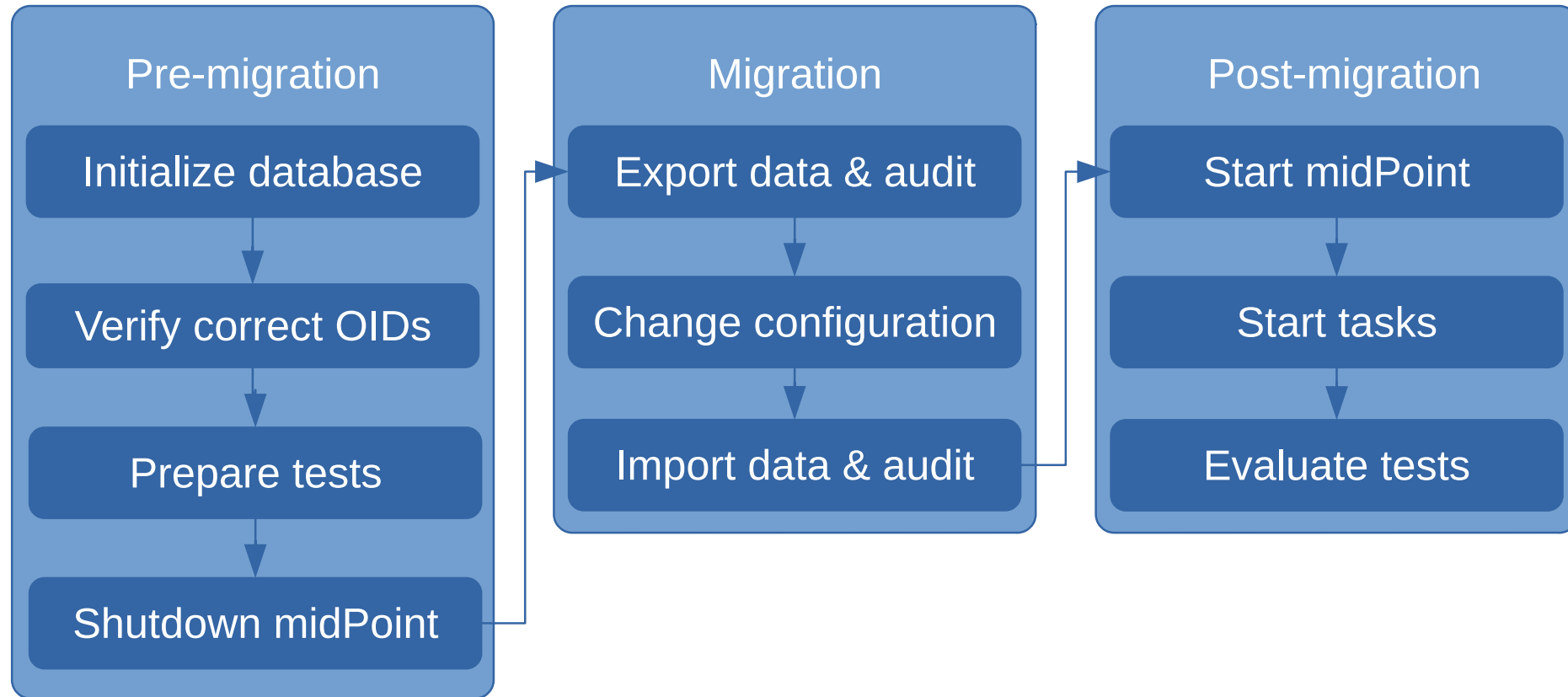
Initialize Native PostgreSQL database

- Database schema and creation scripts are present in **doc/config/sql/native-new/**

- For midPoint database use following scripts:

  - postgres-new.sql

  - postgres-new-quartz.sql

- For midPoint audit database (could be same as midPoint DB) use following scripts

  - postgres-new-audit.sql

Initialize Native PostgreSQL database

Evolveum

tony@tdm-box: /opt/midpoint/doc/config/sql/native-new

```
tony@tdm-box:/opt/midpoint$ cd doc/config/sql/native-new/
tony@tdm-box:/opt/midpoint/doc/config/sql/native-new$ psql --host=localhost --username=midpoint webinar44 -f postgres-new.sql -f postgres-new-quartz.sql -f
postgres-new-upgrade.sql
Password for user midpoint:
tony@tdm-box:/opt/midpoint/doc/config/sql/native-new$ psql --host=localhost --username=midpoint webinar44 -f postgres-new-audit.sql -f postgres-new-upgrade-
audit.sql
```

Verify that only UUIDs are used

- Native repository requires oids to be valid UUIDs

  - objects generated by midPoint always use UUIDs, only manually created objects may use invalid UUIDs

- Use ninja verify for detecting incorrect oids

  bin/ninja.sh verify -w uuid

- midPoint does not allow for oid modifications, you need to export objects with broken oids, remove them from midPoint, fix oids to be UUIDs and reimport them

- You also need to update references to use new correct oids

Verify that only UUIDs are used


```
tony@tdm-box:/opt/midpoint$ bin/ninja.sh verify -w uuid
Initializing using midpoint home; with repository connection
Starting verify
Processed: 0, error: 0, skipped: 0, avg processed: 0obj/s
WARNING user:tony(tony-OID)  OID 'tony' is not valid UUID
Finished verify in 2,84s. Processed: 422, error: 0, skipped: 0, avg processed: 148,64obj/s
tony@tdm-box:/opt/midpoint$
```

Shutting down midPoint

- It's recommended to suspend all server tasks and shutdown midPoint instances before migration / exporting data to simplify process

- It is possible to migrate midPoint to new repository with old instances running, but you need to carefully plan migration and its execution.

  - Exporting data could be done while midPoint is running, but you may need reexport any changes, that happened after initial export

  - midPoint with old repository and midPoint with new repository using same data should never be running at same time

Exporting existing data

- Migration does not support direct database operations and must be done using ninja.

- MidPoint data needs to be exported using **ninja** to file system

```
bin/ninja.sh -v export -r -z -O export.zip
```

- Depending on your deployment complete export may take few minutes to few hours and may result in file size of several GB

- Audit data needs to be exported using **ninja** to file system

```
bin/ninja.sh -v exportAudit -z -O audit-export.zip
```

Exporting existing data

Evolveum


```
tony@tdm-box:/opt/midpoint$ bin/ninja.sh -v export -r -z -O export.zip
2022-01-18 19:23:59,563 [main] INFO  (SYSOUT): Initializing using midpoint home; with repository connection
2022-01-18 19:24:05,429 [main] INFO  (SYSOUT): Starting export
2022-01-18 19:24:05,429 [main] DEBUG (SYSOUT): Operation: started
2022-01-18 19:24:07,452 [pool-1-thread-2] INFO  (SYSOUT): Processed: 0, error: 0, skipped: 0, avg processed: 0obj/s
2022-01-18 19:24:08,397 [pool-1-thread-1] DEBUG (SYSOUT): Operation: producer finished
2022-01-18 19:24:08,475 [pool-1-thread-3] DEBUG (SYSOUT): Operation: finished
2022-01-18 19:24:09,455 [main] INFO  (SYSOUT): Finished export in 3,05s. Processed: 421, error: 0, skipped: 0, avg processed: 138,21obj/s
tony@tdm-box:/opt/midpoint$
tony@tdm-box:/opt/midpoint$ bin/ninja.sh -v exportAudit -z -O audit-export.zip
2022-01-18 19:24:26,333 [main] INFO  (SYSOUT): Initializing using midpoint home; with repository connection
2022-01-18 19:24:32,119 [main] INFO  (SYSOUT): Starting exportAudit
2022-01-18 19:24:32,120 [main] DEBUG (SYSOUT): Operation: started
2022-01-18 19:24:34,149 [pool-1-thread-2] INFO  (SYSOUT): Processed: 0, error: 0, skipped: 0, avg processed: 0obj/s
2022-01-18 19:24:34,384 [pool-1-thread-1] DEBUG (SYSOUT): Operation: producer finished
2022-01-18 19:24:34,539 [pool-1-thread-3] DEBUG (SYSOUT): Operation: finished
2022-01-18 19:24:36,152 [main] INFO  (SYSOUT): Finished exportAudit in 2,42s. Processed: 217, error: 0, skipped: 0, avg processed: 89,71obj/s
tony@tdm-box:/opt/midpoint$
```

Changing midPoint configuration

- Shutdown your midPoint instance

- Modify var/config.xml to enable new native repository and configure its connection to PostgreSQL

  - Replace repository and audit sections with configuration for new native repository

  - See **doc/config/config-native.xml** for quick reference and https://docs.evolveum.com/midpoint/reference/repository/native-postgresql/ for more details

- **Do not start midPoint after configuration change** – it would initialize as fresh install

Example native repository section

```xml
<?xml version="1.0"?>
<configuration>
    <midpoint>
        <repository>
          <type>native</type>
          <jdbcUrl>jdbc:postgresql://localhost:5432/midpoint</jdbcUrl>
          <jdbcUsername>midpoint</jdbcUsername>
          <jdbcPassword>password</jdbcPassword>
        </repository>
        <audit>
            <auditService>
                <auditServiceFactoryClass>com.evolveum.midpoint.audit.impl.LoggerAuditServiceFactory</auditServiceFactoryClass>
            </auditService>
            <auditService>
                <auditServiceFactoryClass>com.evolveum.midpoint.repo.sqale.audit.SqaleAuditServiceFactory</auditServiceFactoryClass>
            </auditService>
        </audit>
```

Importing existing data

- Use ninja import to import production data to new repository

  ```
  bin/ninja.sh -v import -l 4 -z -i export.zip
  ```

- Import will fail fast on schema problems (missing definitions for extensions)

- Observe / review output of command for potential problems

  - Objects with invalid UUIDs will not be imported and will be reported

- Use ninja importAudit to import audit data to new repository

  ```
  bin/ninja.sh -v importAudit -l 4 -z -i audit-export.zip
  ```

- Run ninja import multithreaded using -l option.

Importing existing data (with invalid OIDs)

Post-migration steps

- Migration of data to native PostgreSQL repository is finished

- Make sure same configuration changes are applied to all nodes in cluster deployment

- You may start your migrated instance

- You should observe logs and resume all suspended tasks

- Verify that migration happened succesfully.

Audit Migration

- Audit data may be too large to be migrated at single planned downtime window

- It's possible to migrate audit at later date, but you need to plan for that from beginning of migration

- See Native Audit section in documentation for more information about postponed audit migration
  https://docs.evolveum.com/midpoint/reference/repository/native-audit/#audit-migration-from-other-database

Migration performance numbers

- midPoint with 20k users, 40k shadows, 100k audit records

- Test machine
  - Intel i8700 6-core
  - 32GB RAM
  - HDD WD Blue 5400rpm
  - PostgreSQL running on same machine

Operation	Average objects / sec	Total time
Export data	859	73s
Import data (12 threads)	196	320s
Export audit	380	286s
Import audit (12 threads)	309	350s

Resources

- midPoint 4.4 Release notes

  https://docs.evolveum.com/midpoint/release/4.4/

- midPoint Upgrade Guide

  https://docs.evolveum.com/midpoint/reference/upgrade/upgrade-guide/

- PostgreSQL native repository

  https://docs.evolveum.com/midpoint/reference/repository/native-postgresql/

- Native PostgreSQL Audit Trail

  https://docs.evolveum.com/midpoint/reference/repository/native-audit/

MidScale

- Target:

  - Tens of millions of identities

- Key results:

  - Improved scalability of midPoint

  - Improved visibility, diagnostic and reliability of midPoint

  - Improved performance and user experience of midPoint user interface

Webinars

- Tasks, February 3, 2022 by Pavol Mederly

- Customizing GUI, February 10, 2022 by Katarína Bolemant

- Native reports, February 16, 2022 by Lukáš Škublík

Thank you for your time!

See other talks at https://docs.evolveum.com

Also **follow us** on our social media for further information!

/Evolveum   /Evolveum   /Evolveum   @Evolveum   /Evolveum

Evolveum