# External repository demo

This demo shows how to run midPoint container with externally provided repository.

We will assume that the repository is Oracle-based, hosted at **192.168.56.101**. We have chosen Oracle mainly because it requires adding custom drivers to midPoint.

## Starting

### Driver installation

First of all we have to copy Oracle driver to appropriate place. Custom drivers reside in `lib` subdirectory of midPoint home directory. The home directory is available as a Docker volume `extrepo_midpoint_home` (in this particular composition). After downloading the Oracle driver you can copy it e.g. using the following commands:

```
$ docker volume create extrepo_midpoint_home
$ sudo mkdir /var/lib/docker/volumes/extrepo_midpoint_home/_data/lib
$ sudo cp ~/Downloads/ojdbc7.jar /var/lib/docker/volumes/extrepo_midpoint_home/_data/lib
```

These commands assume the volume `extrepo_midpoint_home` does not exist yet. If it exists, you can skip the first two commands.

### Password setup

Password used to connect to the database is stored in `configs-and-secrets/midpoint/application/database_password.txt` file. So, before starting midPoint, we have to make sure it's correct.

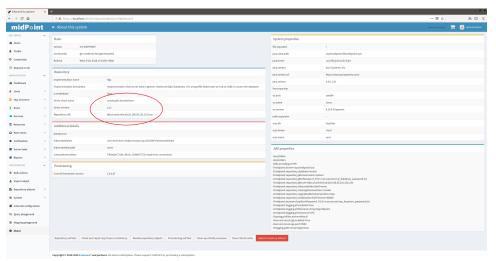In our case the user name is `system` and the password is `oracle`, so let's execute this command:

```
$ cd demo/extrepo
$ echo oracle > configs-and-secrets/midpoint/application/database_password.txt
```

### Running midPoint

Finally, let us start midPoint. We have to provide all the necessary information needed for repository connection:

```
$ cd demo/extrepo
$ env REPO_DATABASE_TYPE=oracle REPO_HOST=192.168.56.101 REPO_USER=system docker-compose up
```

After successful start of midPoint you can login and check that the database is really an Oracle one.



## Containers

The `demo/extrepo` composition contains a single container:

| Container name | Description |
| --- | --- |
| `extrepo_midpoint_s erver_1` | This is the standard container providing midPoint functionality. It contains standalone Tomcat running midPoint application, reverse Apache proxy, and TIER Beacon. |

# Communication

The containers publish the following TCP ports. (*Port mapped to localhost* denotes the mapping of container port to the host port where it can be reached from the outside.)

| Container | Port number | Port mapped to localhost | Description |
| --- | --- | --- | --- |
| `extrepo_midpoint_server_1` | 443 | 8443 | HTTPS port to be used to connect to midPoint application |
| | 80 | - | HTTP port to be used to connect to midPoint application |
| | 9090 | - | Tomcat AJP port used for Apache httpd  Tomcat communication |

# Docker volumes

The following volumes are created to persist data and other relevant files.

| Volume name | Description | Used by container |
| --- | --- | --- |
| `extrepo_midpoint _home` | The midPoint home directory. Contains schema extensions, logs, custom libraries, custom ConnId connectors, and so on. | `extrepo_midpoint_se rver_1` |

# Configuring the composition

The following configuration properties are supported. Please refer to the [main documentation page](#) for their explanation.

| Property | Default value |
| --- | --- |
| `ENV` | `demo` |
| `USERTOKEN` | |
| `REPO_DATABASE_TYPE` | |
| `REPO_JDBC_URL` | `default` |
| `REPO_HOST` | |
| `REPO_PORT` | `default` |
| `REPO_DATABASE` | |
| `REPO_USER` | |
| `REPO_MISSING_SCHEMA_ACTION` | `create` |
| `REPO_UPGRADEABLE_SCHEMA_ACTION` | `stop` |
| `REPO_SCHEMA_VERSION_IF_MISSING` | |
| `REPO_SCHEMA_VARIANT` | |
| `MP_MEM_MAX` | `2048m` |
| `MP_MEM_INIT` | `1024m` |
| `MP_JAVA_OPTS` | |
| `TIER_BEACON_OPT_OUT` | |
| `TIMEZONE` | `UTC` |

You can tailor these to your needs.

The following Docker secrets are used:

| Secret | Location |
|---|---|
| mp_database_password.txt | configs-and-secrets/midpoint/application/database_password.txt |
| mp_keystore_password.txt | configs-and-secrets/midpoint/application/keystore_password.txt |
| mp_host-key.pem | configs-and-secrets/midpoint/httpd/host-key.pem |

The following configuration files are used:

| Target file | Source location |
|---|---|
| /etc/pki/tls/certs/host-cert.pem | configs-and-secrets/midpoint/httpd/host-cert.pem |
| /etc/pki/tls/certs/cachain.pem | configs-and-secrets/midpoint/httpd/host-cert.pem |

You can modify or replace these files as needed.

# Outdated schema detection and upgrade

Here we will demonstrate outdated schema detection and upgrade functionality.

First let us delete any midPoint-specific data from our Oracle database. Then we manually install there midPoint schema from version 3.8, e.g. by applying this script: https://github.com/Evolveum/midpoint/blob/v3.8/config/sql/_all/oracle-3.8-all.sql.

Then let us start the container as above:

```
$ cd demo/extrepo
$ env REPO_DATABASE_TYPE=oracle REPO_HOST=192.168.56.101 REPO_USER=system docker-compose up
```

Unfortunately, midPoint refuses to start. There's something like this on the console (not quite at the end, because there's a large exception with tons of stack trace afterwards):

```
midpoint_server_1  | midpoint;midpoint.log;demo;;2018-10-03 17:59:23,732 [] [main] ERROR (com.evolveum.midpoint.
repo.sql.schemacheck.SchemaChecker):
midpoint_server_1  | midpoint;midpoint.log;demo;;
midpoint_server_1  | midpoint;midpoint.log;demo;;
*******************************************************************************
midpoint_server_1  | midpoint;midpoint.log;demo;;
***                                                                        ***
midpoint_server_1  | midpoint;midpoint.log;demo;;***      Couldn't start midPoint because of a database schema
issue.       ***
midpoint_server_1  | midpoint;midpoint.log;demo;;
***                                                                        ***
midpoint_server_1  | midpoint;midpoint.log;demo;;
*******************************************************************************
midpoint_server_1  | midpoint;midpoint.log;demo;;
midpoint_server_1  | midpoint;midpoint.log;demo;;Database schema is not compatible with the executing code;
however, an upgrade path is available.
midpoint_server_1  | midpoint;midpoint.log;demo;;Current version cannot be determined. The metadata table
(m_global_metadata) is missing or inaccessible. This indicates the version is below 3.9.
midpoint_server_1  | midpoint;midpoint.log;demo;;Required version is: 3.9
midpoint_server_1  | midpoint;midpoint.log;demo;;
midpoint_server_1  | midpoint;midpoint.log;demo;;For more information about the upgrade process please see
https://wiki.evolveum.com/display/midPoint/Release+3.9
midpoint_server_1  | midpoint;midpoint.log;demo;;
midpoint_server_1  | midpoint;midpoint.log;demo;;
```

(The URL mentioned will be functional after 3.9 is released.)

We can now upgrade the database manually.

Or we can try the automatic upgrade process. (Note: better **do not** do this on a production system!)

We need to do two things:

1. Because before midPoint 3.9 there is no explicit information about the schema version in the repository, we have to tell midPoint we have 3.8. Otherwise it would not know. This is done by setting REPO_SCHEMA_VERSION_IF_MISSING to 3.8.
2. We tell midPoint to automatically upgrade the schema. This is done by setting REPO_UPGRADEABLE_SCHEMA_ACTION to upgrade.

```
$ cd demo/extrepo
$ env REPO_DATABASE_TYPE=oracle REPO_HOST=192.168.56.101 REPO_USER=system REPO_SCHEMA_VERSION_IF_MISSING=3.8
REPO_UPGRADEABLE_SCHEMA_ACTION=upgrade docker-compose up
```

Then we see this in the console (among other startup messages):

```
midpoint_server_1  | midpoint;midpoint.log;demo;;2018-10-03 18:10:28,933 [] [main] INFO (com.evolveum.midpoint.
repo.sql.schemacheck.SchemaChecker): Attempting to upgrade database tables using file 'oracle-upgrade-3.8-3.9.
sql'.
midpoint_server_1  | midpoint;midpoint.log;demo;;2018-10-03 18:10:29,057 [] [main] INFO (com.evolveum.midpoint.
repo.sql.schemacheck.SchemaChecker): Validating database tables after upgrading.
midpoint_server_1  | midpoint;midpoint.log;demo;;2018-10-03 18:10:30,264 [] [main] INFO (com.evolveum.midpoint.
repo.sql.schemacheck.SchemaChecker):
midpoint_server_1  | midpoint;midpoint.log;demo;;
midpoint_server_1  | midpoint;midpoint.log;demo;;
**********************************************************************
midpoint_server_1  | midpoint;midpoint.log;demo;;
***                                                      ***
midpoint_server_1  | midpoint;midpoint.log;demo;;***           Database schema upgrade was
successful           ***
midpoint_server_1  | midpoint;midpoint.log;demo;;
***                                                      ***
midpoint_server_1  | midpoint;midpoint.log;demo;;
**********************************************************************
midpoint_server_1  | midpoint;midpoint.log;demo;;
midpoint_server_1  | midpoint;midpoint.log;demo;;Schema was successfully upgraded from 3.8 to 3.9 using script
'oracle-upgrade-3.8-3.9.sql'.
midpoint_server_1  | midpoint;midpoint.log;demo;;Please verify everything works as expected.
midpoint_server_1  | midpoint;midpoint.log;demo;;
midpoint_server_1  | midpoint;midpoint.log;demo;;
```

and midPoint starts successfully.