

PostgreSQL demo

This demo shows how to run midPoint container with alternative repository implementation.

Starting

```
$ cd demo/postgresql
$ docker-compose up
```

After `docker-compose up` command successfully finishes you should see something like this on the console:

```
midpoint_server_1 | midpoint:midpoint.log;demo;;2018-09-20 16:25:22,191 [] [main] INFO (org.springframework.
boot.web.embedded.tomcat.TomcatWebServer): Tomcat started on port(s): 8080 (http) 9090 (http) with context path
'/midpoint'
midpoint_server_1 | midpoint:midpoint.log;demo;;2018-09-20 16:25:22,209 [] [main] INFO (com.evolveum.midpoint.
web.boot.MidPointSpringApplication): Started MidPointSpringApplication in 60.512 seconds (JVM running for
61.688)
```

Now you can log into midPoint using <https://localhost:8443/midpoint> URL, with an user of administrator and a password of 5ecr3t.

You can safely ignore console messages like this:

```
midpoint_data_1 | ERROR: could not serialize access due to read/write dependencies among transactions
midpoint_data_1 | DETAIL: Reason code: Canceled on identification as a pivot, during write.
midpoint_data_1 | HINT: The transaction might succeed if retried.
```

This is a part of standard midPoint conflict resolution process. The mentioned transactions are really retried and they succeed eventually.

Containers

The `demo/postgresql` composition contains the following containers:

Container name	Description
<code>postgresql_midpoint_server_1</code>	This is the standard container providing midPoint functionality. It contains standalone Tomcat running midPoint application, reverse Apache proxy, and TIER Beacon.
<code>postgresql_midpoint_data_1</code>	This container hosts midPoint repository; this time it is implemented on PostgreSQL 9.5 database.

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
<code>postgresql_midpoint_server_1</code>	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
<code>postgresql_midpoint_data_1</code>	5432	5432	Port used to connect to the PostgreSQL database

Docker volumes

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

Volume name	Description	Used by container
-------------	-------------	-------------------

postgresql_midpoint_home	The midPoint home directory. Contains schema extensions, logs, custom libraries, custom ConnId connectors, and so on.	postgresql_midpoint_server_1
postgresql_midpoint_data	Volume hosting PostgreSQL database used by midPoint.	postgresql_midpoint_data_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_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.