Installation guide for Linux and MySQL

2c8 Server

Table of contents

1 Introduction

2c8 Server functionality is a component that runs in an application server called Payara. Payara is an open source software derived from Oracle's Glassfish. Data is stored in MySQL or Microsoft SQL Server[1].

1. It is possible to use the express versions if necessary.

2 Database configuration

If this is a new installation of 2c8 Server you will need to setup a database for it to run against. For existing installations the database will be upgraded when deploying the new server component.

2.1 Creating a new database

This chapter briefly describes how to set up a database for 2c8 Server functionality.

2.1.1 MySQL

Install MySQL 8 if you don't already have it installed. MySQL 5.7 is still supported but we recommend MySQL 8 for all new installations. The rest of this setup documentation assumes that you are running on MySQL 8, if you are running on MySQL 5.7 you will need to download a different JDBC driver and provide a different class namn for the driver.

Log on to mysql

```
# mysql –u root –p
```

Create a new database

```
mysql> CREATE DATABASE mt_server;
```

Add access to the database for user 'conciliate'

```
mysql> GRANT ALL PRIVILEGES ON mt_server.* TO ’conciliate’@’localhost’
IDENTIFIED BY ’choose a password’;
```

Force a reload of grant tables

```
mysql> FLUSH PRIVILEGES;
```

After logging off mysql, execute the script that populates the database

```
# mysql –u root –p mt_server < sql/mysql/mt_mysql_5.4.sql
```

2.2 Upgrading an existing database

The installation package contains patch scripts for upgrading existing databases to the current version. These will be applied automatically when the server component is upgraded in a later step.

3 Payara/GlassFish

2c8 Server functionality runs in an application server called Payara. Starting with 2c8 Modeling Tool 5.0 you will need version 5 of Payara. If you have an older version of Payara or are still running on the older Glassfish server you will need to uninstall this server and install Payara instead.

3.1 Prerequisites

To run Payara, JDK11 is required.

3.1.1 Debian/Ubuntu

Update and then install JDK 11:

```
# sudo apt-get update
```

```
# sudo apt-get install openjdk-11-jdk libmysql-java
```

3.2 Files needed

The following files are needed during the installation[2]:

Name	File
Payara Server (Full)	Installation files
Server component	MTServer-EAR-5.4-uX.ear

3.3 Installation

3.3.1 Debian/Ubuntu

Download the payara-zip linked above to the server. The file you have downloaded should be named something like »payara-5.x.zip« (depending on the version, the actual filename might differ).

2. The installation procedure require root-access to the server.

Unzip the file to an appropriate location (e.g /usr/share/):

```
# unzip payara-5.x.zip /usr/share/
```

3.4 JDBC Driver

Before configuring GlassFish you need to install the proper JDBC driver for your selected database.

3.4.1 MySQL Connector

Go to http://dev.mysql.com/downloads/connector/j/ and download the JDBC Driver, either as a zip or tar archive.

Unpack the downloaded content and copy the jar-file to the lib folder where Payara was installed. (/path/to/payara5/glassfish/lib/)

3.4.2 Restart Payara/Glassfish

After the jdbc-driver has been copied to the Payara/Glassfish installation a restart of the Payara service is required.

3.4.2.1 GNU/Linux

The Payara binaries are located at »/path/to/payara5/glassfish/bin/«. Change directory to that folder and run the asadmin tool to stop the service:

```
# sudo ./asadmin stop-domain domain1
```

When the service has stopped, start it again:

```
# sudo ./asadmin start-domain domain1
```

3.5 Database Connection

Open your server's admin console at http://{your server IP}:4848/ (alternatively http://localhost:4848 if you're accessing from within the server).

By default, there is no password set for the admin user to access the admin console. This is potentially a risk and we strongly encourage you to set a password.

Choose **Resources » JDBC » Connection Pools** from the tree on the left and click **New...** to create a new connection pool.

3.5.1 MySQL

Enter these values:

Name	mt-server
Resource type	javax.sql.DataSource
Database vendor	MySQL

Click **Next** and enter these values

General Settings

Datasource Classname	com.mysql.cj.jdbc.MysqlDataSource

Transaction

Transaction Isolation	read-committed

Additional Properties [3]

user	conciliate[4]
password	conciliate[5]
databaseName	mt_server[6]
serverName	localhost[7]
portNumber	3306
useSSL	false
allowPublicKeyRetrieval	true

Click finish.

A new connection pool named **mt-server** has been created. Click on it to edit it further. Navigate to the »Advanced« tab and edit the following properties:

Connection Validation	Required
Validation method	table
Table name	version

3. Other attributes should be deleted. To do this, select all except the mentioned above and press the Delete Properties.
4. The database user
5. The database password
6. The name of your selected database
7. The name of the server where your database is located

Click Save.

3.5.2 Validate database connection

Choose **Resources » JDBC » Connection Pools » mt-server** from the tree. Click the Ping button. You should see a message saying Ping Succeeded if the application server connected successfully to the database.

3.5.3 Add JDBC Resource

Choose **Resources » JDBC » JDBC Resources** from the tree. Click **New...**to create a new JDBC Resource.

Enter these values

JNDI Name	jdbc/mt-server
Pool Name	mt-server

Make sure **Enabled** is checked, then click Ok.

3.6 Configuration account

Before users can login to the server using their 2c8 accounts the server needs to be registered and activated in the 2c8 Portal. Before performing this step you need to manually assign one or more 2c8 accounts that are allowed to configure the server. Note that this is only used for installation/activation of the server and will not be used in any way to allow access to the content on the server once it is activated.

Choose **Resources – JNDI - Custom Resources** and click on **New...**

Enter these values:

JNDI Name	configAccounts
Resource Type	java.lang.String

Under **Additional Properties** click **Add Property**. In the **Name** column, enter the string **value**. In the **Value** column, enter one or more valid 2c8 accounts that can be used to login, separated by comma characters, for example **user1@company.com,user2@company.com**.

Click **OK.**

3.7 Configure outgoing mailserver in 2c8 Server functionality (Optional)

This configuration step allows the 2c8 Server functionality to send emails to users and subscribers, which is a feature used by 2c8 Workflow and the notification system when publishing. If you plan on using one of these features to send automatic mail notifications an outgoing mail server must be configured in 2c8 Server functionality. **Note that this configuration is optional and only necessary if you want mail notifications.**

Choose **Resources – JavaMail Sessions** and click on **New...**.

Create a new JavaMail Session with the JNDI name **mail/workflow**. Enter the mandatory values:

Mail Host	Hostname for the outgoing mail server
Default User	User account for login to the mail server
Default Sender Address	Mail address to send mail from

The basic properties to use are:

mail.smtp.auth	true (Use authentication for mail server login)
mail.smtp.password	12345 (Password for authentication)
mail.smtp.starttls.enable	true (Use STARTTLS to handle encrypted communication channel)
mail.smtp.port	587 (Mailserver port to use)
mail.always.same.sender	true or false (Set to "true" if mail should appear to be sent from workflow@2c8.com, set to "false" if mail should appear to be sent from **Default Sender Address**)
mail.iso2lang	Language to use when sending notifications. Supported values are **en** for english and **sv** for swedish.

Please note that the values to use differ between different mail servers. The next sections describe some examples of settings to use for some common email services.

3.7.1 Microsoft Online

Demands that "Default Sender Address" is the same as "Default User".

Mail Host	smtp.office365.com
mail.smtp.auth	true
mail.smtp.password	Password for the account **Default User**
mail.smtp.starttls.enable	true
mail.smtp.port	587
mail.always.same.sender	true

3.7.2 Gmail

Demands that the Gmail account is accessible from “Less secure apps”.

Mail Host	smtp.gmail.com
mail.smtp.auth	true
mail.smtp.password	Password for the account **Default User**
mail.smtp.starttls.enable	true
mail.smtp.port	587
mail.always.same.sender	true or false

3.8 Deploy server component

Choose **Applications** from the tree. Click **Deploy** to install a new component (if you are upgrading, start by undeploying the previous component).

Click **Packaged file to be uploaded to the server**. Then click **Browse...** and choose the file »MTServer-EAR-5.4-uX.ear«.

Click **OK** to deploy the component. This will also upgrade the database to the current version so it could take a while for big databases. When it is done, restart the application server.

4 Activate server

The 2c8 server authenticates and authorizes users with their 2c8 accounts. For this to be possible the server has to be registered and activated in the 2c8 Portal. This step exhanges encryption keys that makes it possible to secure communications and make sure that only specific users at a given customer that has been given access through the portal can access the server.

Login to the 2c8 Portal with a portal administrator account. Under **Servers** click **Add Server**. Enter the server name, protocol, hostname and port for the server (the port is 8080 by default in Payara) and click **Save**.

If you have direct access to the server from your computer and are logged into the portal with one of the accounts that were previously set with the **configAccounts** property the rest of the activation will be performed automatically. Otherwise, use the link provided in the next step to manually login to the server configuration interface with one of the **configAccounts** accounts. Click the **Activate** button and follow the on-screen instructions to copy registration data from the server to the portal and then activation data from the portal back to the server.