

# Using External Databases

## Overview

JFrog Mission Control uses several databases for different features of its operation.

- **Elasticsearch v5.5.2** stores analytics data used to create [Graphs](#).
- **PostgreSQL v9.6** is used for continuous data import and scheduling
- **MongoDB v3.2.6** is used for configuration management

Until version 2.1, Mission Control installed an instance of all of these databases dedicated for its own use.

From version 2.1, Mission Control gives you the option of using your own **Elasticsearch**, **PostgreSQL** or **MongoDB** databases if you have these already installed and in use in your organization.

It is up to you to choose which, if any of these databases to externalize when you install Mission Control.

During the installation process, the Mission Control installation script will first ask if you want to perform a standard installation:

```
Perform a standard Installation? [Y/n]:
```

If you respond "Y", the installation process will automatically run to completion and install internal databases for Mission Control to use.

If you respond "n", then for each of the three databases, Mission Control will ask whether you want to use the internal database or an external one you are already using in your organization.



### You take full responsibility for your own databases

If you choose to have Mission Control use any of your own databases for its operation, you take full responsibility for the maintenance, monitoring, backup and correct functioning of these databases.

If you do externalize any of the databases, Mission Control will also ask if you want it to seed the database or prefer to do it manually.

## Page Contents

- [Overview](#)
- [Externalizing Elasticsearch](#)
  - [Manually Seeding Elasticsearch](#)
- [Externalizing MongoDB](#)
  - [Manually Seeding MongoDB](#)
- [Externalizing PostgreSQL](#)
  - [Manually Seeding PostgreSQL](#)
- [Externalizing Databases on an Existing Installation](#)
- [Changing Externalized Databases](#)

## Externalizing Elasticsearch

To externalize the Elasticsearch database, respond to the prompts as described below:

| Prompt  | Response  |
|---|---|
| Install Elasticsearch? [Y/n]:n  | "n"   |
| Please enter the Elasticsearch URL<br>[http://docker.for.mac.localhost:9200]: | Provide the URL to your Elasticsearch database or accept the default if that is correct.  |
| Does this Elasticsearch instance need credentials?                            | "Y" if your Elasticsearch database requires credentials, or "n" if it allows anonymous access.  |
| Please enter the User ID:<br>Please enter the Password:                       | Provide your Elasticsearch user ID and password if required.<br>If you want the installer to seed your database automatically, make sure this user has privileges to create templates, aliases and indices. |

Attempt to seed Elasticsearch?  
[y/N]:

- If you respond with "y", the installer will attempt to perform the following actions to seed the database automatically:
  - create the necessary templates and indices
  - copy all files required to seed the Elasticsearch database to a separate folder to enable manual seeding at a later time should that be necessary

If the automatic seeding operation fails, the installer will display a prompt asking if you want to **retry**, **abort** or **skip**.

- If you respond with "N", the installer will copy all files required to seed the Elasticsearch database to a separate folder to enable manual seeding

## Manually Seeding Elasticsearch



### Seed the database before starting JFrog Mission Control

If you choose to seed your database manually, make sure you do so BEFORE starting up JFrog Mission Control .

To use an external Elasticsearch database, Mission Control requires connectivity to the database and the presence of certain templates and aliases. These are created by the `createIndices.sh` script provided in the installation package using the following process:

- Make the `createIndices.sh` file executable (`chmod +x createIndices.sh`)
- Create the following environment variables (with appropriate values)
  - `ELASTIC_SEARCH_URL`
  - `ELASTIC_SEARCH_USERNAME` (optional)
  - `ELASTIC_SEARCH_PASSWORD` (optional)
- Execute the file (`./createIndices.sh`)

## Externalizing MongoDB

To externalize the MongoDB database, respond to the prompts as described below:

| Prompt   | Response  |
|--|---|
| Install MongoDB? [Y/n]:n                                     | "n"   |
| Please enter the MongoDB Host<br>[docker.for.mac.localhost]: | Enter the host that MongoDB is available on. (Usually, 'localhost'). NOTE: Do NOT include a protocol. |
| Please enter the MongoDB Port [27017]:                       | Enter the port that Mission Control can use to access MongoDB. (Usually, 27017)                       |

|   |   |
|---|---|
| <p> <b>Docker installation does not try to automatically seed the database</b></p> <p>In a Docker installation, the installer does not attempt to seed the external MongoDB database. Instead, it copies the files needed for manual seeding as described in Manually Seeding MongoDB below and the automatic installation process ends here.</p> <p>Attempt to seed MongoDB? [y/N]:</p> | <p>To have the installer seed your MongoDB automatically, respond with "y"</p> <div data-bbox="1013 436 1463 680" style="border: 1px solid #ffc107; padding: 10px; margin: 10px 0;"> <p> <b>MongoDB must be on the same machine for automatic seeding</b></p> <p>The installer can only seed a MongoDB database automatically if it is on the same machine as the installer.</p> <p>If your MongoDB is installed on another machine, respond with "N".</p> </div> <p>If you respond with "N", the installer copies the files you will need to manually seed the database and ends the installation process.</p> <p>If you respond with "y" the installer continue and prompt you with the following questions.</p> |
| <p>Is this a fresh installation with no users? (If you choose 'y', the installer will create an admin user):</p>  | <p>If you respond "y", the installer will prompt you for an admin user and password</p>   |
| <p>Please enter the MongoDB admin user ID:</p> <p>Please enter the MongoDB admin user password:</p>   | <p>Provide an admin user ID and password</p> <p>The installer will attempt to perform the following actions to seed the MongoDB automatically:</p> <ul style="list-style-type: none"> <li>• create the necessary databases and users</li> <li>• copy all files required to seed the MongoDB database to a separate folder to enable manual seeding at a later time should that be necessary</li> </ul> <p>If the automatic seeding operation fails, the installer will display a prompt asking if you want to <b>retry</b>, <b>abort</b> or <b>skip</b>.</p>  |

## Manually Seeding MongoDB



### Seed the database before starting JFrog Mission Control

If you choose to seed your database manually, make sure you do so BEFORE starting up JFrog Mission Control .

MongoDB is used to store metadata about Mission Control's microservices, so the script attempts to create the necessary databases and users. If you are familiar with MongoDB or do not have access to the Mongo instance, you can review the `createMongoUsers.js` file and create these yourself using the appropriate database client. If not and if you have access to the instance where MongoDB is installed, follow the instructions below:

- Copy the files `createMongoUsers.sh` and `createMongoUsers.js` to the system where MongoDB is running.
- Make the shell file executable (`chmod +x createMongoUsers.sh`)
- Execute the file `./createMongoUsers.sh` and follow the prompts on screen.

## Externalizing PostgreSQL

To externalize the PostgreSQL database, respond to the prompts as described below:

| Prompt | Response |
|--------|----------|
|--------|----------|

|   |  |
|---|--|
| Install Postgres? [Y/n]:n   | "n"  |
| Please enter the Postgres Host<br>[docker.for.mac.localhost]:   | Enter the host that Postgres is available on.<br>NOTE: Do NOT include a protocol.  |
| Please enter the Postgres Port [5432]:  | Enter the port that Mission Control can use to access Postgres. (Usually, 5432)  |
| <div style="border: 1px solid green; padding: 10px; margin-bottom: 10px;">  <b>Docker installation does not try to automatically seed the database</b><br/>           In a Docker installation, the installer does not attempt to seed the external Postgres database. Instead, it copies the files needed for manual seeding as described in Manually Seeding Postgres below and the automatic installation process ends here.         </div> <p>Attempt to seed Postgres? [y/N]:</p> | <p>To have the installer seed your Postgres automatically, respond with "y"</p> <div style="border: 1px solid orange; padding: 10px; margin-bottom: 10px;">  <b>Postgres DB must be on the same machine for automatic seeding</b><br/>           The installer can only seed a Postgres database automatically if it is on the same machine as the installer.<br/>           If your Postgres database is installed on another machine, respond with "N".         </div> <p>If you respond with "N", the installer copies the files you will need to manually seed the database and ends the installation process.</p> <p>If you respond with "y" the installer continue and prompt you with the following questions.</p> |
| Please enter the path where Postgres executable (psql) is available:  | Provide the path to your psql executable   |
| Please enter the ID of the root user:<br><br>Please enter the password of the root user:  | Provide a root user ID and password<br>The installer will attempt to perform the following actions to seed the Postgres automatically: <ul style="list-style-type: none"> <li>• create the necessary databases and users</li> <li>• copy all files required to seed the Postgres database to a separate folder to enable manual seeding at a later time should that be necessary</li> </ul> If the automatic seeding operation fails, the installer will display a prompt asking if you want to <b>retry</b> , <b>abort</b> or <b>skip</b> .   |

## Manually Seeding PostgreSQL



### Seed the database before starting JFrog Mission Control

If you choose to seed your database manually, make sure you do so BEFORE starting up JFrog Mission Control .

Postgres is used to store jobs by Mission Control's scheduling service, so the script attempts to create a database (default value: **quartzdb**), a user and password (default values: **quartzdb:password**).

The script then attempts to create several tables and indices.

If you are familiar with Postgres or the externalized instance is managed by another user, you can use any database client to create the necessary database and users and then create the tables in the file **quartz\_postgres.sql**.

If not, and if you have access to the instance where Postgres is installed, follow the instructions below:

- Copy the files `createPostgresUsers.sh` and `quartz_postgres.sql` to the system where Postgres is running.
  - Make the shell file executable (`chmod +x createPostgresUsers.sh`)
  - Execute the file `./createPostgresUsers.sh` and follow the prompts on screen
- 

## Externalizing Databases on an Existing Installation

You can externalize the databases Mission Control uses on an existing installation at any time by simply running the installer again.

When prompted with `Perform a standard upgrade? [Y/n]`, select `"N"`.

Now just continue with the process in the same way you would during a new installation as described in [UsingExternalDatabases](#) above.

---

## Changing Externalized Databases

Mission Control offers you the flexibility to decide when to externalize its databases, and even to switch externalized databases in case you have more than one instance of any particular database installed in your system.

When you do externalize a database, the installer creates a file in the installation log folder (`$JFMC_DATA/installer`) to indicate if the database has been seeded. If, when changing the externalized database, you want the installer to try seeding the new database, make sure to delete the corresponding file before running the installer.