



# TRIFACTA

## Databases Guide

Version: 7.1.2  
Doc Build Date: 11/25/2020

**Copyright © Trifacta Inc. 2020 - All Rights Reserved. CONFIDENTIAL**

These materials (the “Documentation”) are the confidential and proprietary information of Trifacta Inc. and may not be reproduced, modified, or distributed without the prior written permission of Trifacta Inc.

EXCEPT AS OTHERWISE PROVIDED IN AN EXPRESS WRITTEN AGREEMENT, TRIFACTA INC. PROVIDES THIS DOCUMENTATION AS-IS AND WITHOUT WARRANTY AND TRIFACTA INC. DISCLAIMS ALL EXPRESS AND IMPLIED WARRANTIES TO THE EXTENT PERMITTED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE AND UNDER NO CIRCUMSTANCES WILL TRIFACTA INC. BE LIABLE FOR ANY AMOUNT GREATER THAN ONE HUNDRED DOLLARS (\$100) BASED ON ANY USE OF THE DOCUMENTATION.

For third-party license information, please select **About Trifacta** from the Help menu.

- 1. *Install Databases* . . 4
  - 1.1 *Install Databases for PostgreSQL* . . 6
  - 1.2 *Install Databases for MySQL* . . 9
  - 1.3 *Configure the Databases* . 14
  - 1.4 *Create Databases and Users* . 19
  - 1.5 *Upgrade Databases for PostgreSQL* . 20
  - 1.6 *Upgrade Databases for MySQL* . 28
  - 1.7 *Database Reference* 29
    - 1.7.1 *Change Database Passwords for MySQL* . 30
    - 1.7.2 *Change Database Passwords for PostgreSQL* . 32
    - 1.7.3 *Change Database Port* . 34
    - 1.7.4 *Database Parameter Reference* 35
    - 1.7.5 *Install Databases on Amazon RDS* . 40
    - 1.7.6 *Manual Database Install* 44

# Install Databases

The Trifacta® platform uses multiple SQL databases to manage platform metadata. This section describes how to install and initialize these databases.

## DB Installation Pre-requisites

- You must install a supported database distribution. For more information on the supported database versions, see *System Requirements* in the Planning Guide.
  - You must also acquire the database dependencies associated with the operating system distribution where the database is to be installed. Please see the database vendor for more information.
- Please verify that the ports used by the database are opened on the Trifacta node.
  - For more information on default ports, see *System Ports* in the Planning Guide.
  - If you need to use different ports, additional configuration is required. More instructions are provided later.
- Installation and configuration of the database cannot be completed until the Trifacta software has been installed. You should install the software on the Trifacta node first.
- If you are installing the databases on an instance of Amazon RDS, additional setup is required first. See *Install Databases on Amazon RDS*.

Other pre-requisites specific to the database distribution may be listed in the appropriate section below.

**If you are concerned about durability and disaster recovery of your Trifacta metadata, your enterprise backup procedures should include the Trifacta databases. See *Backup and Recovery* in the Admin Guide.**

## List of Databases

The Trifacta® platform requires access to the following databases. Below, you can review database names, descriptions and release in which it was introduced:

Database Name	Description	First Release
Trifacta database (Main)	Storage of users and metadata about your datasets, including completed jobs.	Release 1.0
Jobs database	Storage of job tracking information. Jobs are purged upon completion or job timeout. Failed jobs are purged periodically.	Release 3.2
Authorization Service database	Storage of object permissions.	Release 7.1
Orchestration Service database	Storage of plan execution information. Information is purged upon completion or run timeout. Failed runs are purged periodically.	Release 7.1
Configuration Service database	Storage of system-, tier-, and user-level configuration settings.	Release 6.0
Artifact Storage Service database	Storage for feature-specific usage data such value mappings.	Release 6.0
Job Metadata Service database	Storage of metadata on job execution.	Release 6.4
Scheduling database	Storage of schedules, including datasets to execute.	Release 4.1
Time-based Trigger database	Storage of triggering information.	Release 4.1



# Install Databases for PostgreSQL

## Contents:

- *Limitations*
- *Pre-requisites*
- *Select Configuration File*
- *Database Install*
- *Acquire Port Information*

This section describes how to install the PostgreSQL database server, after which you can create and initialize the databases and their users.

## Limitations

- You must install a supported version of the database. For more information on supported versions of this database type, see *System Requirements* in the Planning Guide.
- SSL connections are not supported.

## Pre-requisites

- The installing user must have write permissions to the directory from which the commands are executed.
- The installing user must have sudo privileges.

## Select Configuration File

By default, the Trifacta platform assumes that you are installing the databases in a PostgreSQL instance, which is reflected in the default platform configuration file. This configuration file is stored here:

```
/opt/trifacta/conf/trifacta-conf.json
```

No modifications to this file are required at this time.

## Database Install

**NOTE:** For this release, PostgreSQL 12.3 is supported on CentOS/RHEL 7 only. PostgreSQL 9.6 (default) is supported across all distributions.

O/S Distribution	URL	Package Name
CentOS/RHEL 7 - PostgreSQL 12.3	<a href="https://download.postgresql.org/pub/repos/yum/reporepms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm">https://download.postgresql.org/pub/repos/yum/reporepms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm</a>	postgresql12-server
CentOS 7	<a href="https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-redhat-repo-42.0-9.noarch.rpm">https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-redhat-repo-42.0-9.noarch.rpm</a>	postgresql96-server
CentOS 8	<a href="https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-8.1-x86_64/pgdg-redhat-repo-42.0-9.noarch.rpm">https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-8.1-x86_64/pgdg-redhat-repo-42.0-9.noarch.rpm</a>	postgresql96-server
RHEL 7	<a href="https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-redhat-repo-42.0-9.noarch.rpm">https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-redhat-repo-42.0-9.noarch.rpm</a>	postgresql96-server

RHEL 8	<a href="https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-8.1-x86_64/pgdg-redhat-repo-42.0-9.noarch.rpm">https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-8.1-x86_64/pgdg-redhat-repo-42.0-9.noarch.rpm</a>	postgresql96-server
--------	---	---------------------

### For CentOS/RHEL 7.x - PostgreSQL 12.3:

```
wget https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
sudo yum -y install pgdg-redhat-repo-latest.noarch.rpm
sudo yum -y install postgresql12-server
```

### For CentOS 7.x:

```
wget https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-redhat-repo-42.0-9.noarch.rpm
sudo yum -y install pgdg-redhat-repo-42.0-9.noarch.rpm
sudo yum -y install postgresql96-server
```

### For CentOS 8.x:

```
wget https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-8.1-x86_64/pgdg-redhat-repo-42.0-9.noarch.rpm
sudo yum -y install pgdg-redhat-repo-42.0-9.noarch.rpm
sudo yum -y install postgresql96-server
```

### For Red Hat Enterprise Linux 7.x:

```
wget https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-redhat-repo-42.0-9.noarch.rpm
sudo yum -y install pgdg-redhat-repo-42.0-9.noarch.rpm
sudo yum -y install postgresql96-server
```

### For Red Hat Enterprise Linux 8.x:

```
wget https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-8.1-x86_64/pgdg-redhat-repo-42.0-9.noarch.rpm
sudo yum -y install pgdg-redhat-repo-42.0-9.noarch.rpm
sudo yum -y install postgresql96-server
```

### For Ubuntu 16.04:

Add the repository's archive key to your apt-key keyring:

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
```

Create a file named `/etc/apt/sources.list.d/pgdg.list`, containing the following:

```
deb http://apt.postgresql.org/pub/repos/apt/xenial-pgdg main
```

Run the following command:

```
sudo apt-get update
sudo apt-get install -y postgresql-9.6
```

## For Ubuntu 18.04:

Add the repository's archive key to your apt-key keyring:

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -
```

Create a file named `/etc/apt/sources.list.d/pgdg.list`, containing the following:

```
deb http://apt.postgresql.org/pub/repos/apt/bionic-pgdg main
```

Run the following command:

```
sudo apt-get update  
sudo apt-get install -y postgresql-9.6
```

## Acquire Port Information

After you have completed the installation, you must acquire the port information for each database from the following locations on the Trifacta node. These port numbers need to be applied inside the Trifacta platform.

**NOTE:** By default, PostgreSQL and the platform use port 5432 for communication. If that port is not available at install/upgrade time, the next available port is used, which is typically 5433. This change may occur if a previous version of PostgreSQL is on the same server. When a non-default port number is used, the platform must be configured to use it. For more information, see *Change Database Port*.

## CentOS/RHEL 7 (PostgreSQL 12.3):

```
/var/lib/pgsql/12/data/postgresql.conf
```

## CentOS/RHEL (PostgreSQL 9.6):

```
/var/lib/pgsql/9.6/data/postgresql.conf
```

## Ubuntu (PostgreSQL 9.6) :

```
/etc/postgresql/9.6/main/postgresql.conf
```



# Install Databases for MySQL

## Contents:

- *Limitations*
  - *Pre-requisites*
    - *Acquire MySQL Java driver*
    - *MySQL log\_bin\_trust\_function\_creators is required for installation*
  - *Select Configuration File*
  - *Database Install*
  - *Acquire Port Information*
  - *Configure MySQL Timezone*
  - *Update MySQL Password Policy*
- 

This section describes how to install the MySQL database server, after which you can create and initialize the databases and their users.

## Limitations

**NOTE:** MySQL 5.7 Community is not supported on CentOS/RHEL 8.1.

- You must install a supported version of the database. For more information on supported versions of this database type, see *System Requirements* in the Planning Guide.

## Pre-requisites

- The installing user must have write permissions to the directory from which the commands are executed.
- The installing user must have sudo privileges.

## Acquire MySQL Java driver

The MySQL Java driver is not packaged with the Trifacta installer. If you are installing the Trifacta databases into MySQL, please acquire the following driver files.

### File 1:

```
mysql-connector-java-6.0.6.jar
```

This file can be downloaded from the following locations:

- <https://dev.mysql.com/downloads/connector/j/>
- <https://mvnrepository.com/artifact/mysql/mysql-connector-java/6.0.6>

This file needs to be installed in the following locations on the Trifacta node:

```
/opt/trifacta/services/artifact-storage-service/build/install/artifact-storage-service/lib/mysql-connector-  
java-6.0.6.jar  
/opt/trifacta/services/authorization-service/build/install/authorization-service/lib/mysql-connector-java-  
6.0.6.jar  
/opt/trifacta/services/configuration-service/build/install/configuration-service/lib/mysql-connector-java-  
6.0.6.jar  
/opt/trifacta/services/batch-job-runner/build/install/batch-job-runner/lib/mysql-connector-java-6.0.6.jar  
/opt/trifacta/services/job-metadata-service/build/install/job-metadata-service/lib/mysql-connector-java-6.0.6.  
jar  
/opt/trifacta/services/orchestration-service/build/install/orchestration-service/lib/mysql-connector-java-  
6.0.6.jar  
/opt/trifacta/services/scheduling-service/server/build/install/scheduling-service/lib/mysql-connector-java-  
6.0.6.jar  
/opt/trifacta/services/time-based-trigger-service/server/build/install/time-based-trigger-service/lib/mysql-  
connector-java-6.0.6.jar
```

## File 2:

```
mysql-connector-java-5.1.37.jar
```

This file can be downloaded from here:

- <https://mvnrepository.com/artifact/mysql/mysql-connector-java/5.1.37>

This file needs to be installed in the following location:

```
/opt/trifacta/services/secure-token-service/server/build/install/secure-token-service/lib/mysql-connector-  
java-5.1.37.jar
```

## MySQL `log_bin_trust_function_creators` is required for installation

When the Trifacta databases are hosted on MySQL, for the installation process, you must enable the `log_bin_trust_function_creators` flag. When enabled, this flag allows MySQL to trust the creators of stored functions to not write unsafe events to the binary log.

**NOTE:** This flag is a global flag in MySQL. It is only required during the installation process and can be disabled afterward.

For more information, see  
[https://dev.mysql.com/doc/refman/5.7/en/replication-options-binary-log.html#sysvar\\_log\\_bin\\_trust\\_function\\_creators](https://dev.mysql.com/doc/refman/5.7/en/replication-options-binary-log.html#sysvar_log_bin_trust_function_creators)

## Select Configuration File

If you are installing the database in a MySQL instance, a separate base configuration has been provided. This configuration file is stored in the following location:

```
/opt/trifacta/conf/trifacta-conf.json.MYSQL_DB
```

To use MySQL, you should back up the default configuration file and then copy the MySQL version in its place:

**If you have already applied configuration changes to `trifacta-conf.json` through the above file or the Admin Settings page, these changes are lost when the following steps are performed. You must manually migrate those changes over or apply the MySQL changes manually.**

```
cp /opt/trifacta/conf/trifacta-conf.json /opt/trifacta/conf/trifacta-conf.json.POSTGRES_DB
cp /opt/trifacta/conf/trifacta-conf.json.MYSQL_DB /opt/trifacta/conf/trifacta-conf.json
```

## Database Install

**NOTE:** The following distributions and commands are for MySQL Community Server 5.7.

O/S Distribution	URL	Package Name
CentOS 7	<a href="https://dev.mysql.com/get/mysql80-community-release-el7-1.noarch.rpm">https://dev.mysql.com/get/mysql80-community-release-el7-1.noarch.rpm</a>	mysql-community-server
CentOS 8	<a href="https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm">https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm</a>	mysql-community-server
RHEL 7	<a href="https://dev.mysql.com/get/mysql80-community-release-el7-1.noarch.rpm">https://dev.mysql.com/get/mysql80-community-release-el7-1.noarch.rpm</a>	mysql-community-server
RHEL 8	<a href="https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm">https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm</a>	mysql-community-server
Ubuntu 16.04	<a href="https://dev.mysql.com/get/mysql-apt-config_0.8.10-1_all.deb">https://dev.mysql.com/get/mysql-apt-config_0.8.10-1_all.deb</a>	
Ubuntu 18.04	<a href="https://dev.mysql.com/get/mysql-apt-config_0.8.10-1_all.deb">https://dev.mysql.com/get/mysql-apt-config_0.8.10-1_all.deb</a>	

### For CentOS 7.x:

```
# Install MySQL Repo List
sudo wget https://dev.mysql.com/get/mysql80-community-release-el7-1.noarch.rpm
sudo rpm -Uvh mysql80-community-release-el7-1.noarch.rpm
# Check list of available MySQL Repos; by default 8.0 is enabled, but we want 5.7
yum repolist all | grep mysql
# Disable 8.0 and enable 5.7
sudo yum-config-manager --disable mysql80-community
sudo yum-config-manager --enable mysql57-community
# Verify repo state
yum repolist all | grep mysql
# Install MySQL Server
sudo yum install mysql-community-server
# Start MySQL server
sudo systemctl start mysqld.service
# Verify status
sudo systemctl status mysqld.service
```

### For CentOS 8.x:

```
# Install MySQL Repo List
sudo wget https://dev.mysql.com/get/mysql80-community-release-el8-1.noarch.rpm
sudo rpm -Uvh mysql80-community-release-el7-1.noarch.rpm
# Check list of available MySQL Repos; by default 8.0 is enabled, but we want 5.7
yum repolist all | grep mysql
# Disable 8.0 and enable 5.7
sudo yum-config-manager --disable mysql80-community
sudo yum-config-manager --enable mysql57-community
# Verify repo state
yum repolist all | grep mysql
# Install MySQL Server
sudo yum install mysql-community-server
# Start MySQL server
sudo systemctl start mysqld.service
# Verify status
sudo systemctl status mysqld.service
```

**For Red Hat Enterprise Linux 7.x:** See CentOS 7 above.

**For Red Hat Enterprise Linux 8.x:** See CentOS 8 above.

#### **For Ubuntu 16.04:**

```
# Install and configure repo config package
sudo apt-get update && sudo apt-get install lsb-release
wget https://dev.mysql.com/get/mysql-apt-config_0.8.10-1_all.deb
sudo debconf-set-selections <<< "mysql-apt-config mysql-apt-config/select-server select mysql-5.7"
sudo DEBIAN_FRONTEND=noninteractive dpkg -i mysql-apt-config_0.8.10-1_all.deb
sudo apt-get update
# Set Installer configs Admin password and install MySQL Server package
sudo debconf-set-selections <<< "mysql-community-server mysql-community-server/root-pass password
<MYSQL_ADMIN_PASSWORD>"
sudo debconf-set-selections <<< "mysql-community-server mysql-community-server/re-root-pass password
<MYSQL_ADMIN_PASSWORD>"
sudo DEBIAN_FRONTEND=noninteractive apt-get install mysql-community-server
sudo service mysql start
```

#### **For Ubuntu 18.04:**

```
# Install and configure repo config package
sudo apt-get update && sudo apt-get install lsb-release
wget https://dev.mysql.com/get/mysql-apt-config_0.8.10-1_all.deb
sudo debconf-set-selections <<< "mysql-apt-config mysql-apt-config/select-server select mysql-5.7"
sudo DEBIAN_FRONTEND=noninteractive dpkg -i mysql-apt-config_0.8.10-1_all.deb
sudo apt-get update
# Set Installer configs Admin password and install MySQL Server package
sudo debconf-set-selections <<< "mysql-community-server mysql-community-server/root-pass password
<MYSQL_ADMIN_PASSWORD>"
sudo debconf-set-selections <<< "mysql-community-server mysql-community-server/re-root-pass password
<MYSQL_ADMIN_PASSWORD>"
sudo DEBIAN_FRONTEND=noninteractive apt-get install mysql-community-server
sudo service mysql start
```

## Acquire Port Information

After you have completed the installation, you must acquire the port information for each database from the following locations on the Trifacta node. These port numbers need to be applied inside the Trifacta platform.

### **CentOS/RHEL (MySQL 5.7):**

The default port is 3306.

**Ubuntu (MySQL 5.7) :** Not supported.

## Configure MySQL Timezone

If your MySQL databases are in a different timezone from the Trifacta node, you must configure the timezone value for each database, so that it can be inserted as part of the connection string.

**NOTE:** If the Trifacta node and your MySQL databases are co-located in the same timezone, you can skip this section.

In the Admin Settings page, these parameters are in the following form:

```
"*.database.mySqlServerTimezone": "",
```

For each database, insert the appropriate timezone value. For more information on supported values, see the documentation for your MySQL product.

See *Database Parameter Reference*.

## Update MySQL Password Policy

**NOTE:** This section only applies to CentOS and RHEL platforms only where MySQL is the installed database.

By default, MySQL enforces a stricter password policy on database passwords. If you prefer to set your own passwords outside of this policy, you must lower the password policy. Please complete the following steps:

```
# Get temporary root password from mysql log
sudo grep 'temporary password' /var/log/mysqld.log
# Connect to server as root
mysql -uroot -p
# Update password
ALTER USER 'root'@'localhost' IDENTIFIED BY '<my_new_password>';
# Unless you plan to update all the User passwords to be meet MySQL Security requirements, you should set the
password policy to low
SET GLOBAL validate_password_policy=LOW;
```

# Configure the Databases

## Contents:

- *Initialize*
  - *PostgreSQL*
  - *MySQL*
  - *Set custom database parameters*
- *Next Steps*

---

## Initialize

Use the following steps to initialize the databases of the Trifacta® platform.

**NOTE:** These steps assume that the Trifacta node is the host of these databases. Please modify the following steps if you are connecting to databases on other nodes.

### Pre-requisites:

- The initializing user must have write permissions to the directory from which the commands are executed.
- The initializing user must have sudo privileges.

### PostgreSQL

**NOTE:** In the following steps, the default version is PostgreSQL 9.6.

#### 1. For CentOS 7.x, CentOS 8.x:

```
sudo /usr/pgsql-9.6/bin/postgresql96-setup initdb
```

#### 2. For CentOS 7.x - PostgreSQL 12.3:

```
sudo /usr/pgsql-12/bin/postgresql12-setup initdb
```

#### 3. For RHEL 7.x, RHEL 8.x:

```
sudo /usr/pgsql-9.6/bin/postgresql96-setup initdb
```

#### 4. For RHEL 7.x - PostgreSQL 12.3:

**NOTE:** This feature is in Beta release.

```
sudo /usr/pgsql-12/bin/postgresql12-setup initdb
```

#### 5. For Ubuntu 16.04 / 18.04:

```
pg_createcluster -d /var/lib/postgresql/9.6/main 9.6 main
```

## MySQL

No additional steps are required to initialize the databases in MySQL.

### Set custom database parameters

Use the following steps to set custom database names, usernames, and passwords in the Trifacta platform:

1. Edit `/opt/trifacta/conf/trifacta-conf.json`.
2. For each database, you can review the parameters in the listed area and make modifications as needed.

**NOTE:** For each database, you should change the default password. This change must also be applied on the database server. See *Change Database Passwords for PostgreSQL*. See *Change Database Passwords for MySQL*.

**NOTE:** The `type` is set to `POSTGRES` by default. Modify the value if you are installing the databases into a different database server.

Database	Parameter area
Main database	<code>webapp.database.*</code>
Jobs database	<code>batch-job-runner.database.*</code>
Scheduling database	<code>scheduling-service.database.*</code>
Time-Based Trigger database	<code>time-based-trigger-service.database.*</code>
Configuration Service database	<code>configuration-service.database.*</code>
Job Metadata Service database	<code>job-metadata-service.database.*</code>
Artifact Storage Service database	<code>artifact-storage-service.database.*</code>
Authorization Service database	<code>authorization-service.database.*</code>
Orchestration Service database	<code>orchestration-service.database.*</code>
Optimizer Service database	<code>optimizer-service.database.*</code>

For more information, see *Database Parameter Reference*.

3. Make changes in the file as needed and save.

### Apply customizations on upgrade

If you have customized database properties, you must apply the edits from the new sample file to the existing configuration file after you have upgrade the Trifacta platform.

If you are using all defaults, you can just overwrite the existing file with the new version's sample file.

### PostgreSQL:

1. Locate the sample Postgres configuration file:

```
/opt/trifacta/bin/setup-utils/db/pg_hba.conf.SAMPLE
```

2. If you are upgrading and have customizations in your existing version, you must apply the edits in the above to the following file. Otherwise, overwrite the following file with the above one based on your operating system:
  - a. CentOS/RHEL dir: `/var/lib/pgsql/9.6/data/pg_hba.conf`
  - b. Ubuntu dir: `/etc/postgresql/9.6/main/pg_hba.conf`
3. From the `SAMPLE` file, copy the following declarations and paste them into the production `pg_hba.conf` file **above any other declarations**:

**NOTE:** You can substitute different database usernames and groups for the ones listed below (`trifacta` and `trifacta`). These values may be needed for other configuration.

a. Trifacta database:

```
local  trifacta          trifacta          md5
host   trifacta          trifacta          127.0.0.1/32     md5
host   trifacta          trifacta          ::1/128          md5
```

b. Jobs database:

```
local  trifacta-activiti trifactaactiviti md5
host   trifacta-activiti trifactaactiviti 127.0.0.1/32     md5
host   trifacta-activiti trifactaactiviti ::1/128          md5
```

c. Scheduling database:

```
local  trifactaschedulingservice
trifactaschedulingservice          md5
host   trifactaschedulingservice    trifactaschedulingservice 127.0.0.1
/32          md5
host   trifactaschedulingservice    trifactaschedulingservice ::1
/128          md5
```

d. Time-based Trigger database:

```
local  trifactatimebasedtriggerservice
trifactatimebasedtriggerservice          md5
host   trifactatimebasedtriggerservice    trifactatimebasedtriggerservice
127.0.0.1/32          md5
host   trifactatimebasedtriggerservice    trifactatimebasedtriggerservice ::1
/128          md5
```

e. Configuration Service database:

```
local  trifactaconfiguration
trifactaconfiguration          md5
host   trifactaconfiguration    trifactaconfiguration 127.0.0.1
/32          md5
host   trifactaconfiguration    trifactaconfiguration ::1
/128          md5
```

f. Artifact Storage Service database:

```
local  trifactaartifactstorageservice
trifactaartifactstorageservice          md5
host   trifactaartifactstorageservice    trifactaartifactstorageservice 127.0.0.1
/32          md5
host   trifactaartifactstorageservice    trifactaartifactstorageservice ::1
/128          md5
```



```
/32          md5
host    trifactaartifactstorageservice    trifactaartifactstorageservice    :::1
/128          md5
```

g. Job Metadata Service database:

```
local    trifactajobmetadataservice
trifactajobmetadataservice          md5
host    trifactajobmetadataservice    trifactajobmetadataservice    127.0.0.1
/32          md5
host    trifactajobmetadataservice    trifactajobmetadataservice    :::1
/128          md5
```

h. Authorization Service database:

```
local    trifactaauthorizationsservice
trifactaauthorizationsservice          md5
host    trifactaauthorizationsservice    trifactaauthorizationsservice    127.0.0.1
/32          md5
host    trifactaauthorizationsservice    trifactaauthorizationsservice    :::1
/128          md5
```

i. Orchestration Service database:

j. Optimizer Service database:

k. Save the file.

4. Restart the databases:

- a. If you have also restarted the operating system, please execute the following first, followed by the O/S-specific commands:

**NOTE:** This command is valid only if the Postgres DB is also hosted in the Trifacta node.

```
chkconfig postgresql-9.6 on
```

For PostgreSQL 12.3 on CentOS 7:

```
chkconfig postgresql-12 on
```

b. CentOS/RHEL:

```
sudo service postgresql-9.6 start
```

c. CentOS/RHEL 7 (PostgreSQL 12.3):

```
sudo service postgresql-12 start
```

d. Ubuntu:

```
sudo service postgresql start
```

## MySQL:

Upgrading MySQL versions is not supported in this release.

## Next Steps

1. If the configuration files indicate that the databases are listening on a port other than the default, this port number must be applied within the Trifacta platform configuration. For more information, see *Change Database Port*.
2. If you are using non-default usernames and passwords, they must be applied within the Trifacta platform configuration. For more information, see *Change Database Passwords for PostgreSQL*.
3. When you have completed the above configuration, you can create the databases and their roles (users) and perform additional configuration. See *Create Databases and Users*.

# Create Databases and Users

## Contents:

- *Create DBs and Users*
  - *PostgreSQL*
  - *MySQL*
- *Backup*
- *Configure Non-Default Database Connections*

---

## Create DBs and Users

### PostgreSQL

Run the following script, which builds the four databases and specifies the appropriate roles (users) for each database, based on the parameters you have specified in `trifacta-conf.json` and in the `pg_hba.conf`:

**NOTE:** This script must be run as the root user or via sudo superuser.

```
/opt/trifacta/bin/setup-utils/db/trifacta-create-postgres-roles-dbs.sh
```

### MySQL

Run the following script, which builds the four databases and specifies the appropriate roles (users) for each database, based on the parameters you have specified in `trifacta-conf.json`:

**NOTE:** This script must be run as the root user or via sudo superuser.

```
/opt/trifacta/bin/setup-utils/db/trifacta-create-mysql-users-dbs.sh -u <MySQL_admin_username> -w <MySQL_admin_password>
```

where:

- `-u` = MySQL admin username, which is usually `root`.
- `-w` = MySQL admin password

## Backup

For more information on backup recommendations and commands, see *Backup and Recovery* in the Admin Guide.

## Configure Non-Default Database Connections

If you have used non-default values for the username, password, host, or port value for either database, you must update platform configuration. For more information, see *Database Parameter Reference*.

# Upgrade Databases for PostgreSQL

## Contents:

- *Pre-requisites*
  - *Backup*
  - *Acquire Distribution and Port Information*
  - *Database Prep*
    - *Before You Begin*
    - *Using non-default ports*
  - *Upgrade PostgreSQL 9.6 to 12.3*
    - *PostgreSQL path references*
    - *Upgrade for CentOS/RHEL - PostgreSQL 9.6 to 12.3*
  - *Upgrade PostgreSQL 9.3 to 9.6*
    - *Upgrade for CentOS/RHEL - PostgreSQL 9.3 to 9.6*
    - *Upgrade for Ubuntu 16.04 (Xenial) - PostgreSQL 9.3 to 9.6*
    - *Upgrade for Ubuntu 18.04 (Bionic Beaver) - PostgreSQL 9.3 to 9.6*
    - *Transfer Settings*
    - *Verify and Cleanup*
- 

This section describes how to upgrade in-use instances of the Trifacta® databases for a working deployment of the Trifacta platform.

- When you upgrade to a new release of the platform, the underlying databases and their structures are automatically migrated to the new format. However, the database releases are not touched.

In this example, Postgres 9.3 is upgraded to Postgres 9.6.

## Pre-requisites

- The installing user must have write permissions to the directory from which the commands are executed.
- The installing user must have sudo privileges.
- Verify that you know the host and port number for each database.

## Backup

Before you begin, please verify that you have a valid backup of each Trifacta database. See *Backup and Recovery* in the Admin Guide.

## Acquire Distribution and Port Information

Before you begin, you must:

- Acquire the port information for the current database. Typically, this value is 5432.

**NOTE:** By default, PostgreSQL and the platform use port 5432 for communication. If that port is not available at install/upgrade time, the next available port is used, which is typically 5433. This change may occur if a previous version of PostgreSQL is on the same server. When a non-default port number is used, the platform must be configured to use it. For more information, see *Change Database Port*.

- Acquire the latest distribution for the database software.

For more information, see *Install Databases for PostgreSQL*.

For more information, see *Install Databases for MySQL*.

## Database Prep

This procedure describes the process for upgrading the PostgreSQL version in use by the Trifacta databases. This procedure assumes the following:

- All current databases are co-located with the software on the Trifacta node.
- Some downtime of the databases during the upgrade process is ok.

### Before You Begin

- If you haven't already, please back up each database and all PostgreSQL configuration files. See *Backup and Recovery* in the Admin Guide.
- Login to the node where the databases are hosted.

### Using non-default ports

The default port for these commands is port 5432.

If you are using a non-standard port for either the pre- or post-upgrade versions of PostgreSQL, you can use the following parameters with the `pg_upgrade` command:

Parameter	Description
<code>--old-port=&lt;pre-upgrade_port&gt;</code>	The port number of the pre-upgrade instance of PostgreSQL
<code>--new-port=&lt;post-upgrade_port&gt;</code>	The port number to use for the post-upgrade instance of PostgreSQL

For more information, see <https://www.postgresql.org/docs/9.6/pgupgrade.html>.

## Upgrade PostgreSQL 9.6 to 12.3

The following sections cover upgrading the versions listed below:

- Source version: PostgreSQL 9.6
- Upgrade version: PostgreSQL 12.3

Beginning in Release 7.1 Hot Fix 5, PostgreSQL 12.3 is supported on a limited basis. Important notes:

**NOTE:** Upgrade to PostgreSQL 12.3 is supported in this release for CentOS/RHEL 7 only.

**NOTE:** Upgrade to PostgreSQL 12.3 is supported in Release 7.1 Hot Fix 5 or later only. For more information on acquiring this build, please contact *Trifacta Support*.

**NOTE:** The database version upgrade must be performed after you have upgraded the software.

### PostgreSQL path references

In PostgreSQL 10 and later, directory references to versions of PostgreSQL paths have changed.

PostgreSQL 9.6 example path:

```
/var/lib/pgsql/9.6/data/pg_hba.conf
```

PostgreSQL 12.3 example path:

```
/var/lib/pgsql/12/data/pg_hba.conf
```

For more information, see [https://wiki.postgresql.org/wiki/YUM\\_Installation](https://wiki.postgresql.org/wiki/YUM_Installation).

## Upgrade for CentOS/RHEL - PostgreSQL 9.6 to 12.3

**NOTE:** For this release, PostgreSQL 12.3 is supported for supported versions of CentOS/RHEL 7 only. See *Product Support Matrix*.

1. Install the Postgres repo for yum. Below, PostgreSQL 12.3 is installed.

a. For CentOS 7.x:

```
wget https://download.postgresql.org/pub/repos/yum/repорpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
sudo yum -y install pgdg-redhat-repo-latest.noarch.rpm
```

b. For CentOS 8.x: Not supported

c. For Red Hat Enterprise Linux 7.x:

```
wget https://download.postgresql.org/pub/repos/yum/repорpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
sudo yum -y install pgdg-redhat-repo-latest.noarch.rpm
```

d. For Red Hat Enterprise Linux 8.x: Not supported

2. Install the upgraded version. Below, PostgreSQL 12.3 is installed.

```
sudo yum install postgresql12-server
```

3. Create a new data directory for the new database version:

a. For CentOS 7.x:

```
sudo /usr/pgsql-12/bin/postgresql-12-setup initdb
sudo systemctl enable postgresql-12
```

b. For RHEL 7.x:

```
sudo /usr/pgsql-12/bin/postgresql-12-setup initdb
sudo systemctl enable postgresql-12
```

4. Use the following commands to verify that the upgrade is possible:

```
su postgres
cd ~
/usr/pgsql-12/bin/pg_upgrade --old-bindir=/usr/pgsql-9.6/bin --new-bindir=/usr/pgsql-12/bin --old-datadir=/var/lib/pgsql/9.6/data/ --new-datadir=/var/lib/pgsql/12/data/ --check
```

**NOTE:** Please be sure to include the `--check` command line switch at the end of the above command.

If all is well, you should see a `Clusters are compatible` message.

5. Stop the old version of the database service as the root user:
  - a. CentOS/RHEL:

```
sudo service postgresql-9.6 stop
```

6. Run the upgrade as the `postgres` user:
  - a. CentOS/RHEL:

```
su postgres
/usr/pgsql-12/bin/pg_upgrade --old-bindir=/usr/pgsql-9.6/bin --new-bindir=/usr/pgsql-12/bin --
old-datadir=/var/lib/pgsql/9.6/data/ --new-datadir=/var/lib/pgsql/12/data/
```

Part of the output should include the following. These hashes will be re-indexed in a later step.

```
Your installation contains hash indexes. These indexes have different
internal formats between your old and new clusters, so they must be
reindexed with the REINDEX command. The file
  reindex_hash.sql
when executed by psql by the database superuser will recreate all invalid
indexes; until then, none of these indexes will be used.
```

```
Upgrade Complete
```

```
-----
Optimizer statistics are not transferred by pg_upgrade so,
once you start the new server, consider running:
  ./analyze_new_cluster.sh
```

```
Running this script will delete the old cluster's data files:
  ./delete_old_cluster.sh
```

7. Copy in `pg_hba.conf`:
  - a. Copy the default version from the upgraded software:

```
cp /opt/trifacta/bin/setup-utils/db/pg_hba.conf.SAMPLE /var/lib/pgsql/12/data/pg_hba.conf
```

**Tip:** You may find it easier to using a diff tool to compare your current version of this file (`/var/lib/pgsql/9.6/data/pg_hba.conf`) to the new file (`/var/lib/pgsql/12/data/pg_hba.conf`). If there are only a few updates, then you may find it easier to copy your current version into the target directory and then edit it to apply the differences.

- b. If you have made modifications to this configuration file:
    - i. Open this file:

```
/var/lib/pgsql/9.6/data/pg_hba.conf
```

ii. Apply custom values to this file:

```
/var/lib/pgsql/12/data/pg_hba.conf
```

8. Start new PostgreSQL:

```
systemctl start postgresql-12
```

9. Use the following command to reindex the hashes:

```
psql -f reindex_hash.sql
```

10. Verify that PostgreSQL is up and running:

```
systemctl status postgresql-12.service
```

## Upgrade PostgreSQL 9.3 to 9.6

The following sections cover upgrading the versions listed below:

- Source version: PostgreSQL 9.3
- Upgrade version: PostgreSQL 9.6

### Upgrade for CentOS/RHEL - PostgreSQL 9.3 to 9.6

1. Install the Postgres repo for yum. Below, PostgreSQL 9.6 is installed.
  - a. For CentOS 7.x:

```
wget https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-centos96-9.6-3.noarch.rpm
sudo yum -y install pgdg-centos96-9.6-3.noarch.rpm
```

- b. For CentOS 8.x:

```
wget https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-8-x86_64/pgdg-centos96-9.6-3.noarch.rpm
sudo yum -y install pgdg-centos96-9.6-3.noarch.rpm
```

- c. For Red Hat Enterprise Linux 7.x:

```
wget https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/pgdg-centos96-9.6-3.noarch.rpm
sudo yum -y install pgdg-redhat96-9.6-3.noarch.rpm
```

- d. For Red Hat Enterprise Linux 8.x:

```
wget https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-8-x86_64/pgdg-centos96-9.6-3.noarch.rpm
sudo yum -y install pgdg-redhat96-9.6-3.noarch.rpm
```



2. Install the upgraded version. Below, PostgreSQL 9.6 is installed.

```
sudo yum install postgresql96.x86_64 postgresql96-server.x86_64 postgresql96-contrib.x86_64 -y
```

3. Create a new data directory for the new database version:

- a. For CentOS 7.x, CentOS 8.x:

```
sudo /usr/pgsql-9.6/bin/postgresql96-setup initdb
```

- b. For RHEL 7.x, RHEL 8.x:

```
sudo /usr/pgsql-9.6/bin/postgresql96-setup initdb
```

4. Use the following commands to verify that the upgrade is possible:

```
su postgres
cd ~
/usr/pgsql-9.6/bin/pg_upgrade --old-bindir=/usr/pgsql-9.3/bin/ --new-bindir=/usr/pgsql-9.6/bin/ --old-
datadir=/var/lib/pgsql/9.3/data/ --new-datadir=/var/lib/pgsql/9.6/data/ --check
```

If all is well, you should see a `Clusters are compatible` message.

5. Stop the old version of the database service as the root user:
  - a. CentOS/RHEL:

```
sudo service postgresql-9.3 stop
```

6. Run the upgrade as the postgres user:
  - a. CentOS/RHEL:

```
su postgres
/usr/pgsql-9.6/bin/pg_upgrade --old-bindir=/usr/pgsql-9.3/bin/ --new-bindir=/usr/pgsql-9.6/bin/
--old-datadir=/var/lib/pgsql/9.3/data/ --new-datadir=/var/lib/pgsql/9.6/data/
```

## Upgrade for Ubuntu 16.04 (Xenial) - PostgreSQL 9.3 to 9.6

### Steps:

1. Stop the platform:

```
service trifacta stop
```

2. Install the upgraded version. Below, PostgreSQL 9.6 is installed.

```
sudo apt-get install postgresql-9.6 postgresql-server-dev-9.6 postgresql-contrib-9.6 -y
```

3. Stop PostgreSQL:

```
sudo systemctl stop postgresql
service stop postgresql
service postgresql stop
```

4. Upgrade the PostgreSQL 9.3 version to PostgreSQL 9.6, using the newly installed version:

**NOTE:** Below, some values are too long for a single line. Single lines that overflow to additional lines are marked with a \. The backslash should not be included if the line is used as input.

```
sudo su - postgres -c '/usr/lib/postgresql/9.6/bin/pg_upgrade \  
-b /usr/lib/postgresql/9.3/bin -B /usr/lib/postgresql/9.6/bin \  
-d /var/lib/postgresql/9.3/main/ -D /var/lib/postgresql/9.6/main/ \  
-O "-c config_file=/etc/postgresql/9.6/main/postgresql.conf" \  
-o "-c config_file=/etc/postgresql/9.3/main/postgresql.conf"'
```

5. Remove the old version of PostgreSQL (9.3):

```
sudo apt-get remove postgresql-9.3 -y
```

6. Restart PostgreSQL and the platform:

```
service postgresql start  
service trifacta start
```

## Upgrade for Ubuntu 18.04 (Bionic Beaver) - PostgreSQL 9.3 to 9.6

PostgreSQL should not be on version 9.3 if you are using this version of the operating system.

### Transfer Settings

For all operating systems, you must transfer the settings from your old version of PostgreSQL to the new one.

1. Update the port number and any other settings in the platform configuration. To apply this configuration change, login as an administrator to the Trifacta node. Then, edit `trifacta-conf.json`. Some of these settings may not be available through the *Admin Settings Page*. For more information, see *Platform Configuration Methods*.
  - a. Replace all instances of the PostgreSQL port number for the earlier version and replace with the desired port number for the upgrade version.

**NOTE:** By default, PostgreSQL and the platform use port 5432 for communication. If that port is not available at install/upgrade time, the next available port is used, which is typically 5433. This change may occur if a previous version of PostgreSQL is on the same server. When a non-default port number is used, the platform must be configured to use it. For more information, see *Change Database Port*.

- b. Make the above changes and save the file.
2. Transfer settings from the configuration files for your old database version to the new one. Please review the old and new versions of these files:

**NOTE:** It is risky to perform a straight copy of these configuration files. Settings may change. New ones may be introduced. Setting values specific to the installation may be overwritten in a copy. Please retain a backup of both versions of each file before migrating settings.

Path to PostgreSQL 9.3 file	Path to PostgreSQL 9.6 file
<code>/var/lib/pgsql/9.3/data/pg_hba.conf</code>	<code>/var/lib/pgsql/9.6/data/pg_hba.conf</code>
<code>/var/lib/pgsql/9.3/data/postgresql.conf</code>	<code>/var/lib/pgsql/9.6/data/postgresql.conf</code>

3. Start the service:

a. CentOS/RHEL:

```
sudo service postgresql-9.6 start
```

b. Ubuntu:

```
sudo service postgresql start
```

c. When the service restarts, you can check the cluster status using the following script:

```
./analyze_new_cluster.sh
```

4. After you have completed the database installation, you must review the port number of the newly installed database, which may have changed between versions of the database software. That new port number must be applied through the Trifacta software. For more information, see *Change Database Port*.
5. If all is well, restart the platform. See *Start and Stop the Platform* in the Install Guide.

## Verify and Cleanup

1. Verify operations on all databases:
  - a. Login to the application.
  - b. Load a dataset from Flow View.
  - c. Run a job.
  - d. Schedule a job and execute it.
  - e. See *Verify Operations* in the Admin Guide.
2. If all of the above tests pass, you can use the following script to delete the old PostgreSQL version and its data directory:

```
./delete_old_cluster.sh
```

3. Restart the Trifacta platform. See *Start and Stop the Platform* in the Install Guide.

# Upgrade Databases for MySQL

Upgrading the databases for MySQL is not supported in this release. Please start with a clean install of the databases on MySQL.

# Database Reference

After installation, the following database topics may apply to your environment.

# Change Database Passwords for MySQL

You should change the passwords for the Trifacta® databases immediately after installing the software.

Please complete the following steps to change the passwords for the Trifacta platform users of the Trifacta databases.

## Change in Database Server

1. If you have not done so already, switch to the MySQL user and launch:

```
sudo su - mysql
mysql
```

2. For each database, change the default password. Default usernames are listed for each database. Please modify the command if you are using different roles and insert a more secure password for <new\_pwd>:

**NOTE:** In MySQL, the dash is a forbidden character in usernames. Please use underscores instead.

```
ALTER USER 'trifacta'@'localhost' IDENTIFIED BY '<new_pwd>';
ALTER USER 'trifactaactiviti'@'localhost' IDENTIFIED BY '<new_pwd>';
ALTER USER 'trifactaschedulingservice'@'localhost' IDENTIFIED BY '<new_pwd>';
ALTER USER 'trifactatimebasedtriggerservice'@'localhost' IDENTIFIED BY '<new_pwd>';
ALTER USER 'trifactaconfigurationsservice'@'localhost' IDENTIFIED BY '<new_pwd>';
ALTER USER 'trifactasartifactsstorageservice'@'localhost' IDENTIFIED BY '<new_pwd>';
ALTER USER 'trifactajobmetadataservice'@'localhost' IDENTIFIED BY '<new_pwd>';
ALTER USER 'trifactaauthorizationservice'@'localhost' IDENTIFIED BY '<new_pwd>';
ALTER USER 'trifactaorchestrationservice'@'localhost' IDENTIFIED BY '<new_pwd>';
```

3. Exit:

```
\q
exit
```

## Change in Trifacta configuration

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Modify the `user` and `password` properties for any of the following databases:

Database Name	Parameter area
Trifacta database	<code>webapp.database.*</code>
Jobs database	<code>batch-job-runner.database.*</code>
Scheduling database	<code>scheduling-service.database.*</code>
Time-based Trigger database	<code>time-based-trigger-service.database.*</code>

Configuration Service database	configuration-service.database.*
Artifact Storage Service database	artifact-storage-service.database.*
Job Metadata Service database	job-metadata-service.database.*
Authorization Service database	authorization-service.database.*
Orchestration Service database	orchestration-service.database.*

For more information, see *Database Parameter Reference*.

3. Save your changes and restart the platform.

# Change Database Passwords for PostgreSQL

You should change the passwords for the Trifacta® databases immediately after installing the software.

Please complete the following steps to change the passwords for the Trifacta platform users of the Trifacta databases.

## Change in Database Server

1. If you have not done so already, switch to the Postgres user and launch psql:

```
sudo su - postgres
psql
```

2. For each database, change the default password. Default usernames are listed for each database. Please modify the command if you are using different roles and insert a more secure password for <new\_pwd>:

```
ALTER ROLE trifacta WITH PASSWORD '<new_pwd>';
ALTER ROLE trifactaactiviti WITH PASSWORD '<new_pwd>';
ALTER ROLE trifactatimebasedtriggerservice WITH PASSWORD '<new_pwd>';
ALTER ROLE trifactaschedulingservice WITH PASSWORD '<new_pwd>';
ALTER ROLE trifactaconfigurationsservice WITH PASSWORD '<new_pwd>';
ALTER ROLE trifactaartifactstorageservice WITH PASSWORD '<new_pwd>';
ALTER ROLE trifactajobmetadataservice WITH PASSWORD '<new_pwd>';
ALTER ROLE trifactaauthorizationsservice WITH PASSWORD '<new_pwd>';
ALTER ROLE trifactaorchestrationservice WITH PASSWORD '<new_pwd>';
```

3. Exit psql:

```
\q
exit
```

## Change in Trifacta configuration

1. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
2. Modify the user and password properties for any of the following databases:

Database Name	Parameter area
Trifacta database	<code>webapp.database.*</code>
Jobs database	<code>batch-job-runner.database.*</code>
Scheduling database	<code>scheduling-service.database.*</code>
Time-based Trigger database	<code>time-based-trigger-service.database.*</code>
Configuration Service database	<code>configuration-service.database.*</code>
Artifact Storage Service database	<code>artifact-storage-service.database.*</code>
Job Metadata Service database	<code>job-metadata-service.database.*</code>
Authorization Service database	



	authorization-service.database.*
Orchestration Service database	orchestration-service.database.*

For more information, see *Database Parameter Reference*.

3. Save your changes and restart the platform.

# Change Database Port

If needed, you can change the default port used by the database instance for the Trifacta® platform. Please complete the following commands to avoid conflicts between versions.

**If you are changing the port within the database, those changes must be applied through your database system by an administrator. To limit downtime, port number changes should be done at the same time in the platform and the database. For more information, please see the documentation included with your database product.**

**NOTE:** By default, PostgreSQL and the platform use port 5432 for communication. If that port is not available at install/upgrade time, the next available port is used, which is typically 5433. This change may occur if a previous version of PostgreSQL is on the same server. When a non-default port number is used, the platform must be configured to use it. For more information, see *Change Database Port*.

**NOTE:** By default, the ports for all databases are set to the default database port for PostgreSQL. For more information on the default database ports, see *Database Parameter Reference*.

## Steps:

1. Stop the Trifacta services. For more information, see *Start and Stop the Platform* in the Install Guide.
2. Stop the database server.
3. Change the port number in the database administration console. For more information, see the documentation provided with your database distribution.
4. Start the database server. Verify that it is listening on the new port number.
5. Restart the platform.
6. Login to the application.
7. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
8. Modify the `port` properties for any of the following databases:

Database Name	Parameter area
Trifacta database	<code>webapp.database.*</code>
Jobs database	<code>batch-job-runner.database.*</code>
Scheduling database	<code>scheduling-service.database.*</code>
Time-based Trigger database	<code>time-based-trigger-service.database.*</code>
Configuration Service database	<code>configuration-service.database.*</code>
Artifact Storage Service database	<code>artifact-storage-service.database.*</code>
Job Metadata Service database	<code>job-metadata-service.database.*</code>
Authorization Service database	<code>authorization-service.database.*</code>
Orchestration Service database	<code>orchestration-service.database.*</code>

For more information, see *Database Parameter Reference*.

9. Save your changes and restart the platform.
10. Verify that you can login and run a simple job. See *Verify Operations* in the Admin Guide.

# Database Parameter Reference

## Contents:

- *Trifacta database*
- *Jobs database*
- *Jobs database thread pool size*
- *Scheduling Service database*
- *Time-based Trigger Service database*
- *Configuration Service database*
- *Artifact Storage Service database*
- *Job Metadata Service database*
- *Authorization Service database*
- *Orchestration Service database*
- *Parameter Reference*

This section contains reference information on the relevant Trifacta platform parameters that can be modified for each database.

**NOTE:** Defaults are listed below. The default database server is PostgreSQL. If you have installed databases on a different server, updates are required.

**NOTE:** Parameters that you specify through the application must match the values that were applied to the database at the time of creation or that have been set through the database administration console.

**Tip:** You should modify the default password for each database.

Parameters can be modified through:

- **Admin Settings Page:** If the application is working, then you are able to connect to the Trifacta database. Parameters can be modified by an administrator from within the application. See *Admin Settings Page*.

**Tip:** This method is recommended.

- `trifacta-conf.json`: If the application is not working, then an administrator must modify through the following file on the Trifacta node:  
`/opt/trifacta/conf/trifacta-conf.json`
- For more information, see *Platform Configuration Methods* in the Configuration Guide.

## Trifacta database

```
"webapp.database.username": "trifacta",
"webapp.database.logging": false,
"webapp.database.name": "trifacta",
"webapp.database.host": "localhost",
"webapp.database.password": "<pwd_trifactaDB>",
"webapp.database.type": "postgresql",
"webapp.database.port": 5432,
```

```
"webapp.database.mySqlServerTimezone": "",
"webapp.database.pool.maxIdleTimeInMillis": 30000,
"webapp.database.pool.maxConnections": 10,
```

The following parameters apply to the Trifacta database only:

Parameter	Description
logging	Set this value to true to enable logging on the Trifacta database.
pool.maxIdleTimeInMillis	Specifies the maximum permitted idle time for a database connection before it is automatically closed.
pool.maxConnections	Defines the maximum permitted database connections for the Trifacta database.

## Jobs database

Modify the batch-job-runner.database settings:

```
"batch-job-runner.database.username": "trifactaactiviti",
"batch-job-runner.database.name": "trifacta-activiti",
"batch-job-runner.database.type": "postgresql",
"batch-job-runner.database.driver": "org.postgresql.Driver",
"batch-job-runner.database.host": "localhost",
"batch-job-runner.database.password": "<pwd_trifactaactivitiDB>",
"batch-job-runner.database.port": 5432,
"batch-job-runner.database.mySqlServerTimezone": "",
"batch-job-runner.database.verifyServerCertificate": false,
"batch-job-runner.database.poolMaxSize": 20,
```

## Jobs database thread pool size

You can modify the following settings to specify minimum and maximum permitted thread pools for the Jobs database:

```
"batch-job-runner.database.minPoolSize": 3,
"batch-job-runner.database.initialPoolSize": 3,
"batch-job-runner.database.maxPoolSize": 50,
```

Parameter Name	Description
batch-job-runner.database.minPoolSize	Integer representing the minimum size of the database connection pool
batch-job-runner.database.initialPoolSize	Integer representing the initial size of the database connection pool
batch-job-runner.database.maxPoolSize	Integer representing the maximum size of the database connection pool

## Scheduling Service database

```
"scheduling-service.database.type": "POSTGRESQL",
"scheduling-service.database.host": "localhost",
"scheduling-service.database.port": "5432",
"scheduling-service.database.name": "trifactascheduling-service",
"scheduling-service.database.username": "trifactascheduling-service",
"scheduling-service.database.password": "<pwd_scheduling-serviceDB>",
"scheduling-service.database.mySqlServerTimezone": "",
"scheduling-service.database.verifyServerCertificate": false,
"scheduling-service.database.poolMaxSize": 20,
```

## Time-based Trigger Service database

```
"time-based-trigger-service.database.type": "POSTGRESQL",
"time-based-trigger-service.database.host": "localhost",
"time-based-trigger-service.database.port": "5432",
"time-based-trigger-service.database.name": "trifactatimebasedtriggerservice",
"time-based-trigger-service.database.username": "trifactatimebasedtriggerservice",
"time-based-trigger-service.database.password": "<pwd_triggerserviceDB>",
"time-based-trigger-service.database.mySqlServerTimezone": "",
"time-based-trigger-service.database.verifyServerCertificate": false,
"time-based-trigger-service.database.poolMaxSize": 20,
```

## Configuration Service database

```
"configuration-service.database.type": "POSTGRESQL",
"configuration-service.database.host": "localhost",
"configuration-service.database.port": "5432",
"configuration-service.database.name": "trifactaconfigurationservice",
"configuration-service.database.username": "trifactaconfigurationservice",
"configuration-service.database.password": "<pwd_trifactaconfigurationserviceDB>",
"configuration-service.database.mySqlServerTimezone": "",
"configuration-service.database.verifyServerCertificate": false,
"configuration-service.database.poolMaxSize": 20,
```

## Artifact Storage Service database

```
"artifact-storage-service.database.type": "POSTGRESQL",
"artifact-storage-service.database.host": "localhost",
"artifact-storage-service.database.port": "5432",
"artifact-storage-service.database.name": "trifactaartifactstorageservice",
"artifact-storage-service.database.username": "trifactaartifactstorageservice",
"artifact-storage-service.database.password": "<pwd_trifactaartifactstorageserviceDB>",
"artifact-storage-service.database.mySqlServerTimezone": "",
"artifact-storage-service.database.verifyServerCertificate": false,
"artifact-storage-service.database.poolMaxSize": 20,
```

## Job Metadata Service database

```
"job-metadata-service.database.type": "POSTGRESQL",
"job-metadata-service.database.host": "localhost",
"job-metadata-service.database.port": "5432",
"job-metadata-service.database.name": "trifactajobmetadataservice",
"job-metadata-service.database.username": "trifactajobmetadataservice",
"job-metadata-service.database.password": "<pwd_trifactajobmetadataserviceDB>",
"job-metadata-service.database.mySqlServerTimezone": "",
"job-metadata-service.database.verifyServerCertificate": false,
"job-metadata-service.database.poolMaxSize": 20,
```

## Authorization Service database

```
"authorization-service.database.type": "POSTGRESQL",
"authorization-service.database.host": "localhost",
"authorization-service.database.port": "5432",
"authorization-service.database.name": "trifactaauthorizationservice",
"authorization-service.database.username": "trifactaauthorizationservice",
"authorization-service.database.password": "<pwd_trifactaauthorizationserviceDB>",
"authorization-service.database.mySqlServerTimezone": "",
"authorization-service.database.verifyServerCertificate": false,
"authorization-service.database.poolMaxSize": 20,
```

## Orchestration Service database

```
"orchestration-service.database.type": "POSTGRESQL",
"orchestration-service.database.host": "localhost",
"orchestration-service.database.port": "5432",
"orchestration-service.database.name": "trifactaorchestrationservice",
"orchestration-service.database.username": "trifactaorchestrationservice",
"orchestration-service.database.password": "<pwd_trifactaorchestrationserviceDB>",
"orchestration-service.database.mySqlServerTimezone": "",
"orchestration-service.database.verifyServerCertificate": false,
"orchestration-service.database.poolMaxSize": 20,
```

## Parameter Reference

The following generalized parameters apply to one or more of the databases.

Parameter	Description
host	Host of the database. Default value is <code>localhost</code> , meaning the database is hosted on the Trifacta node.
port	Port number for the database. Default value is <code>5432</code> for all databases, assuming that they are PostgreSQL. Modify this value accordingly if you have installed the databases onto a different type of database server.
name	Name of the database. This value should match what was used during installation.
user or username	The username to use to connect to the database.
password	<p>Password to use to connect to the database.</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p><b>NOTE:</b> For each database, you should change the default password.</p> </div> <p>This change must also be applied on the database server.</p> <ul style="list-style-type: none"> <li>• See <i>Change Database Passwords for PostgreSQL</i>.</li> <li>• See <i>Change Database Passwords for MySQL</i>.</li> </ul>
type	<p>The default database type is <code>POSTGRESQL</code>.</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p><b>NOTE:</b> Modify this value accordingly if you have installed the databases onto a different type of database server.</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p><b>NOTE:</b> H2 database type is used for internal testing. It is not a supported database.</p> </div>
driver	Name of the database. Do not modify.
mySqlServerTimezone	<p>(MySQL DBs only) If populated, the MySQL connection string for the database is augmented with the specified timezone, which overrides the default mapping/detection of the timezone.</p> <p>If empty, the default MySQL timezone settings are used for the database.</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p><b>NOTE:</b> Leave this value as empty if the Trifacta node and your MySQL databases are located in the same timezone.</p> </div> <p>For more information, see <i>Install Databases for MySQL</i>.</p>
verifyServerCertificate	When <code>true</code> , the local database client does not check the validity of the database server certificate when connecting.

	<p>Default is false.</p> <p><b>NOTE:</b> Do not modify unless necessary for connectivity.</p>
poolMaxSize	<p>Maximum number of threads permitted in the connection pool for the database. Default is 20.</p> <p><b>NOTE:</b> Do not modify unless necessary.</p>

# Install Databases on Amazon RDS

## Contents:

- *Limitations*
- *Pre-requisites*
- *Initialize RDS instance*
- *Configure the Trifacta platform for RDS*
- *Create the databases*
- *Configure non-default connections*
- *Logging*

---

As needed, the Trifacta® databases can be installed as PostgreSQL DBs on Amazon RDS. Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud.

## Limitations

**NOTE:** If you are installing or upgrading a deployment of Trifacta Wrangler Enterprise that uses or will use a remote database service, such as Amazon RDS, for hosting the Trifacta databases, please contact *Trifacta Customer Success Services*. For this release, additional configuration may be required.

- In Amazon RDS, databases must be installed in a supported version of PostgreSQL.
- SSL connectivity is not supported.

## Pre-requisites

**NOTE:** You can use the suggested defaults below for sizing your RDS instance. If you have questions or concerns about sizing recommendations, please contact *Trifacta Support*.

- Admin access to an Amazon RDS account

## Initialize RDS instance

### Steps:

1. In your RDS dashboard, click **Launch a DB instance**.

**NOTE:** The RDS instance must be launched in the same Amazon region as the Trifacta node.

2. For Select Engine: Select **PostgreSQL**.
3. For Production?: Choose **Yes** if you are deploying the database for a production instance of the Trifacta platform. Otherwise, select **No**.
4. DB Engine: `postgres`
5. For the DB details, see below:

**NOTE:** Except as noted below, properties should be specified according to your enterprise requirements.

#### a. Instance Specifications:



- i. License Model: `postgresql-license`
  - ii. DB Engine Version: For more information on the supported versions of PostgreSQL, see *System Requirements* in the Planning Guide.
  - iii. Allocated Storage: at least 10 GB
- 6. For Advanced Settings, please apply the following settings:
  - a. **Network and Security:**
    - i. VPC security group must allow for access from the Trifacta platform.
  - b. **Database Options:**
    - i. Database Name: `trifacta`
    - ii. Database Port: `5432`
      - 1. The port number can be changed as needed. See *System Ports* in the Planning Guide.
- 7. Populate other properties according to your enterprise requirements.
- 8. To complete the set up click **Launch DB Instance**.
- 9. Create a master username and password for the DB instance.

**NOTE:** In the configuration instructions listed below, this master username is referenced as: `trifacta_rds`.

- 10. When the RDS DB instance is up and running, please collect the following information, which is used later:
  - a. Public DNS
  - b. Port Number
  - c. Admin username
  - d. Admin password

## Configure the Trifacta platform for RDS

Please complete the following steps to integrate the Trifacta platform with the DB instance you just created.

### Steps:

1. In the RDS console, you must find the Public DNS endpoint for the RDS instance you created:
  - a. Under Instances, expand the name of the instance you created.
  - b. The DNS endpoint should be listed under the name in the Endpoint section.
2. Set the `host` for each database to the Public DNS endpoint for the RDS instance:

Database	Parameter
Main database	<code>webapp.database.host</code>
Jobs database	<code>batch-job-runner.database.host</code>
Scheduling database	<code>scheduling-service.database.host</code>
Time-based Trigger database	<code>time-based-trigger-service.database.host</code>
Configuration Service database	<code>configuration-service.database.host</code>
Artifact Storage Service database	<code>artifact-storage-service.database.host</code>
Job Metadata Service database	<code>job-metadata-service.database.host</code>
Authorization Service database	<code>authorization-service.database.host</code>
Orchestration Service database	<code>orchestration-service.database.host</code>

For more information, see *Database Parameter Reference*.

3. To set custom database names, usernames, and passwords:

- a. Edit `/opt/trifacta/conf/trifacta-conf.json`.
- b. For each database, you can review the database name, username, and password.

Database	Parameter area
Main database	<code>webapp.database.*</code>
Jobs database	<code>batch-job-runner.database.*</code>
Scheduling database	<code>scheduling-service.database.*</code>
Time-Based Trigger database	<code>time-based-trigger-service.database.*</code>
Configuration Service database	<code>configuration-service.database.*</code>
Artifact Storage Service database	<code>artifact-storage-service.database.*</code>
Job Metadata Service database	<code>job-metadata-service.database.*</code>
Authorization Service database	<code>authorization-service.database.*</code>
Orchestration Service database	<code>orchestration-service.database.*</code>

- c. Make changes in the file as needed and save.

## Create the databases

**NOTE:** For the host, port, username, and password values in the following configuration, use the public DNS value for the RDS instance.

From the Trifacta node, switch to the Postgres user and run the following commands against the RDS Host for the master username:

```
su - postgres
psql -h <endpoint of RDS Instance> -U trifacta_rds postgres
```

**NOTE:** `postgres` is the name of the existing/default database.

Please do the following for each database, which initializes the DB and assigns the `trifacta` role to the `trifacta_rds` master username:

### Trifacta database:

```
CREATE ROLE trifacta LOGIN ENCRYPTED PASSWORD '<pwd_trifacta>';
GRANT trifacta TO trifacta_rds;
CREATE DATABASE "trifacta" WITH OWNER trifacta;
```

### Jobs database:

```
CREATE ROLE trifactaactiviti LOGIN ENCRYPTED PASSWORD '<pwd_trifactaactiviti>';
GRANT trifactaactiviti TO trifacta_rds;
CREATE DATABASE "trifacta-activiti" WITH OWNER trifactaactiviti;
```

### Time-based trigger service database:

```
CREATE ROLE trifactatimebasedtriggerservice LOGIN ENCRYPTED PASSWORD '<pwd_trifactatimebasedtriggerservice>';
GRANT trifactatimebasedtriggerservice TO trifacta_rds;
CREATE DATABASE "trifactatimebasedtriggerservice" WITH OWNER trifactatimebasedtriggerservice;
```

### Scheduling service database:

```
CREATE ROLE trifactaschedulingservice LOGIN ENCRYPTED PASSWORD '<pwd_trifactaschedulingservice>';
GRANT trifactaschedulingservice TO trifacta_rds;
CREATE DATABASE "trifactaschedulingservice" WITH OWNER trifactaschedulingservice;
```

### Configuration Service database:

```
CREATE ROLE trifactaconfigurationservice LOGIN ENCRYPTED PASSWORD '<pwd_trifactaconfigurationservice>';
GRANT trifactaconfigurationservice TO trifacta_rds;
CREATE DATABASE trifactaconfigurationservice WITH OWNER trifactaconfigurationservice;
```

### Artifact Storage Service database:

```
CREATE ROLE trifactaartifactstorageservice LOGIN ENCRYPTED PASSWORD '<pwd_trifactaartifactstorageservice>';
GRANT trifactaartifactstorageservice TO trifacta_rds;
CREATE DATABASE trifactaartifactstorageservice WITH OWNER trifactaartifactstorageservice;
```

### Job Metadata Service database:

```
CREATE ROLE trifactajobmetadataservice LOGIN ENCRYPTED PASSWORD '<pwd_trifactajobmetadataservice>';
GRANT trifactajobmetadataservice TO trifacta_rds;
CREATE DATABASE trifactajobmetadataservice WITH OWNER trifactajobmetadataservice;
```

### Authorization Service database:

```
CREATE ROLE trifactaauthorizationservice LOGIN ENCRYPTED PASSWORD '<pwd_trifactaauthorizationservice>';
GRANT trifactaauthorizationservice TO trifacta_rds;
CREATE DATABASE trifactaauthorizationservice WITH OWNER trifactaauthorizationservice;
```

### Orchestration Service database:

```
CREATE ROLE trifactaorchestrationservice LOGIN ENCRYPTED PASSWORD '<pwd_trifactaorchestrationservice>';
GRANT trifactaorchestrationservice TO trifacta_rds;
CREATE DATABASE trifactaorchestrationservice WITH OWNER trifactaorchestrationservice;
```

## Configure non-default connections

If you have used non-default values for the username, password, host, or port value for either database, you must update platform configuration. For more information, see *Database Parameter Reference*.

## Logging

1. To review database logs in RDS, locate the Instance details page in the RDS console.
2. Click **Recent Events and Logs**.
3. If your account has the appropriate permissions, all Trifacta database logs are available here.

# Manual Database Install

## Contents:

- *Pre-requisites*
  - *Required special database permissions*
- *Manual DB Install for PostgreSQL*
  - *Launch PostgreSQL*
  - *Trifacta database*
  - *Jobs database*
  - *Scheduling database*
  - *Time-based Trigger database*
  - *Configuration Service database*
  - *Artifact Storage Service database*
  - *Job Metadata Service database*
  - *Authorization Service database*
  - *Orchestration Service database*
- *Manual DB Install for MySQL*
  - *Launch MySQL*
  - *Trifacta database*
  - *Jobs database*
  - *Scheduling database*
  - *Time-based Trigger database*
  - *Configuration Service database*
  - *Artifact Storage Service database*
  - *Job Metadata Service database*
  - *Authorization Service database*
  - *Orchestration Service database*

---

As needed, you can perform manual creation of users and individual databases used by the Trifacta® platform.

**Tip:** In most installation scenarios, manual database installs are not needed.

## Pre-requisites

1. The database server must be installed and available:
  - a. See *Install Databases for PostgreSQL*.
  - b. See *Install Databases for MySQL*.
2. The databases must be initialized.

**NOTE:** If you are using non-standard database names, users, passwords, port numbers, that information must be applied during installation and afterwards in the Trifacta platform configuration.

For more information, see *Configure the Databases*.

## Required special database permissions

If you are manually installing any of the following databases, special database permissions are required for the user performing the installation and access. When the Trifacta platform is initialized for the first time, a cross-migration of data is performed between these databases, which is required for successful installation and enablement of the related services. This cross-migration process requires special database privileges.

To execute the cross-migration, the database user for the following databases requires special privileges:

- trifacta database
- trifactaauthorizationservice database

The privileges required for cross-migration depend on the database server on which these databases are being installed:

Database Type	Required permissions for database user
PostgreSQL	SUPERUSER
MySQL	FILE

**If the cross-migration is run as part of the normal install or upgrade script, then these permissions are already set for the above databases. If for some reason, you are manually installing or upgrading these databases, the above privileges must be applied.**

After install or upgrade, the above privileges are not required for normal operations on these databases. These privileges must be re-applied if cross-migrations need to be performed in the future.

## Manual DB Install for PostgreSQL

Use the following steps to manually create the users and databases required by the platform in PostgreSQL.

### Launch PostgreSQL

#### Pre-requisites for PostgreSQL:

- The installing user must have write permissions to the directory from which the commands are executed.
- The installing user must have sudo privileges.

1. Switch to the Postgres user. Launch psql.

```
sudo su - postgres
```

**NOTE:** Unless the port number for postgres has been modified, it should be listening at the default value: 5432.

2. Launch psql using the following command, applying the admin password when prompted:

```
psql --host=${POSTGRES_HOST_NAME} --port=${POSTGRES_HOST_PORT} --username=${POSTGRES_ADMIN_USERNAME} --password=${POSTGRES_ADMIN_PASSWORD}
```

where:

Parameter	Description
host	Public DNS value for database server instance
port	Port number value for database server instance
username	Admin username for database server instance

password | Password for admin username

3. Execute the following commands using the `postgres` user.

**NOTE:** The values in platform configuration must match the values that you use below. Below are the default values.

### Trifacta database

```
CREATE ROLE trifacta LOGIN ENCRYPTED PASSWORD '<pwd_trifacta>';
CREATE DATABASE trifacta WITH OWNER trifacta;
```

### Jobs database

```
CREATE ROLE trifactaactiviti WITH LOGIN ENCRYPTED PASSWORD '<pwd_trifactaactiviti>';
CREATE DATABASE "trifacta-activiti" WITH OWNER trifactaactiviti;
```

### Scheduling database

```
CREATE ROLE trifactaschedulingservice LOGIN ENCRYPTED PASSWORD '<pwd_trifactaschedulingservice>';
CREATE DATABASE trifactaschedulingservice WITH OWNER trifactaschedulingservice;
```

### Time-based Trigger database

```
CREATE ROLE trifactatimebasedtriggerservice LOGIN ENCRYPTED PASSWORD '<pwd_trifactatimebasedtriggerservice>';
CREATE DATABASE trifactatimebasedtriggerservice WITH OWNER trifactatimebasedtriggerservice;
```

### Configuration Service database

```
CREATE ROLE trifactaconfigurationservice LOGIN ENCRYPTED PASSWORD '<pwd_trifactaconfigurationservice>';
CREATE DATABASE trifactaconfigurationservice WITH OWNER trifactaconfigurationservice;
```

### Artifact Storage Service database

```
CREATE ROLE trifactaartifactstorageservice LOGIN ENCRYPTED PASSWORD '<pwd_trifactaartifactstorageservice>';
CREATE DATABASE trifactaartifactstorageservice WITH OWNER trifactaartifactstorageservice;
```

### Job Metadata Service database

```
CREATE ROLE trifactajobmetadataservice LOGIN ENCRYPTED PASSWORD '<pwd_trifactajobmetadataservice>';
CREATE DATABASE trifactajobmetadataservice WITH OWNER trifactajobmetadataservice;
```

### Authorization Service database

```
CREATE ROLE trifactaauthorizationservice LOGIN ENCRYPTED PASSWORD '<pwd_trifactaauthorizationservice>';
CREATE DATABASE trifactaauthorizationservice WITH OWNER trifactaauthorizationservice;
```

### Orchestration Service database

```
CREATE ROLE trifactaorchestrationservice LOGIN ENCRYPTED PASSWORD '<pwd_trifactaorchestrationservice>';
CREATE DATABASE trifactaorchestrationservice WITH OWNER trifactaorchestrationservice;
```

## Manual DB Install for MySQL

Use the following steps to manually initialize the databases required by the platform in MySQL.

### Launch MySQL

#### Pre-requisites for MySQL:

- The installing user must have write permissions to the directory from which the commands are executed.
- The installing user must have sudo privileges.

1. Login to MySQL as the root user. When prompted, enter the password for `root` user:

```
mysql -u root -p
```

2. Execute the following commands using the `root` user.

**NOTE:** The values in platform configuration must match the values that you use below. Below are the default values.

#### Trifacta database

```
Create USER 'trifacta' IDENTIFIED BY '<pwd_trifacta>';  
CREATE DATABASE trifacta;  
Grant all on trifacta.* to trifacta;
```

#### Jobs database

```
Create USER 'trifactaactiviti' IDENTIFIED BY '<pwd_trifactaactiviti>';  
CREATE DATABASE trifactaactiviti;  
Grant all on trifactaactiviti.* to trifactaactiviti;
```

#### Scheduling database

```
Create USER 'trifactaschedulingservice' IDENTIFIED BY '<pwd_trifactaschedulingservice>';  
CREATE DATABASE trifactaschedulingservice;  
Grant all on trifactaschedulingservice.* to trifactaschedulingservice;
```

#### Time-based Trigger database

```
Create USER 'trifactatimebasedtriggerservice' IDENTIFIED BY '<pwd_trifactatimebasedtriggerservice>';  
CREATE DATABASE trifactatimebasedtriggerservice;  
Grant all on trifactatimebasedtriggerservice.* to trifactatimebasedtriggerservice;
```

#### Configuration Service database

```
Create USER 'trifactaconfigurationservice' IDENTIFIED BY '<pwd_trifactaconfigurationservice>';  
CREATE DATABASE trifactaconfigurationservice;  
Grant all on trifactaconfigurationservice.* to trifactaconfigurationservice;
```

## Artifact Storage Service database

```
Create USER 'trifactaartifactstorageservice' IDENTIFIED BY '<pwd_trifactaartifactstorageservice>';  
CREATE DATABASE trifactaartifactstorageservice;  
Grant all on trifactaartifactstorageservice.* to trifactaartifactstorageservice;
```

## Job Metadata Service database

```
Create USER 'trifactajobmetadataservice' IDENTIFIED BY '<pwd_trifactajobmetadataservice>';  
CREATE DATABASE trifactajobmetadataservice;  
Grant all on trifactajobmetadataservice.* to trifactajobmetadataservice;
```

## Authorization Service database

```
Create USER 'trifactaauthorizationservice' IDENTIFIED BY '<pwd_trifactaauthorizationservice>';  
CREATE DATABASE trifactaauthorizationservice;  
Grant all on trifactaauthorizationservice.* to trifactaauthorizationservice;
```

## Orchestration Service database

```
Create USER 'trifactaorchestrationservice' IDENTIFIED BY '<pwd_trifactaorchestrationservice>';  
CREATE DATABASE trifactaorchestrationservice;  
Grant all on trifactaorchestrationservice.* to trifactaorchestrationservice;
```





Copyright © 2020 - Trifacta, Inc.  
All rights reserved.