TIBCO® Managed File Transfer Command Center

Container Deployment Guide

Version 8.7.0 | October 2025

Contents

Introduction to Container Deployment

TIBCO® Managed File Transfer Command Center can be container enabled. This means that TIBCO Managed File Transfer (MFT) Command Center can have the following capabilities:

- It can run in the cloud.
- It supports Docker and Kubernetes.
- It can be used for on-site or off-site cloud environments.
- It simplifies deployment: install, upgrade, and hotfixes.
- It can be used to deploy configuration changes.
- It runs on most 64-bit flavors of Linux that support Docker and Kubernetes.
- It supports Kubernetes scaling to start and stop TIBCO MFT Command Center instances as required.

The Docker container is the basis for most cloud implementations. Therefore, you must follow the steps to create TIBCO MFT Command CenterDocker images. Since each cloud implementation is different, the parameters supplied in the sample Kubernetes files do not work for every customer site. If you use container management software such as OpenShift, you need to tailor OpenShift to use the MFT Docker container.

The Docker file, provided in the <MFT-Install>/cloud/k8s_samples directory, shows a sample of how TIBCO MFT Command Center can be configured. You can use this document as a guideline to deploy TIBCO MFT Command Center using a Docker container.

Requirements and Recommendation for Container Deployment

You must check the requirements and follow the recommendation provided in this section for easy deployment.

Note: Instructions are provided to create and deploy TIBCO MFT Command Centercontainers. TIBCO MFT Command Center provides support for container-related issues. Sample YAML files are provided for Kubernetes deployment. However, providing support for using Kubernetes or other container orchestration tools is not in the scope of the support provided by TIBCO MFT Command Center. Contact TIBCO PSG for assistance with orchestrating TIBCO MFT Command Center containers.

You must have a solid understanding of Docker and Kubernetes. Before you start deployment, ensure that the following requirements are met:

- Current Docker and Kubernetes are installed and operational.
- The necessary firewall ports are opened to allow communications.
- The necessary Kubernetes port mapping is completed.

  Note: The FTP mode requires special Kubernetes mapping.

- Each TIBCO® Managed File Transfer Internet Server instance listens on a variety of ports. For example, Internet Server listens on port 46464 for Platform Server requests, port 7443 for HTTPS requests and port 2022 for SSH requests. Each MFT Internet Server instance should listen on the same set of ports.
- Each TIBCO® Managed File Transfer Command Center instance listens on the same HTTPS port. By default, MFT Command Center listens on HTTPS port 8443.
- All TIBCO MFT Command Center instances must have connectivity to the HTTPS port of the TIBCO MFT Internet Server images.

It is recommended that you create images for the base version and for each hotfix that is installed. By following this recommendation, you can easily deploy different versions of TIBCO MFT Command Centeracross your network.

For example, if you install version 8.5.0 and hotfixes 8.5.0_HF-001 and 8.5.0_HF-002, it is recommended that you create Docker images with the following versions:

- TIBCO MFT Command Center 8.5.0
- TIBCO MFT Command Center 8.5.0 with Hotfix 8.5.0_HF-001
- TIBCO MFT Command Center 8.5.0 with Hotfix 8.5.0_HF-002

Note: You can create images for TIBCO MFT Command Center and TIBCO MFT Internet Server independent of each other.

If you have already created a base TIBCO MFT Command Center image and want to install a hotfix, you can directly start with the Install TIBCO MFT Command Center Hotfixes step in the cloud deployment process.

Note: Kubernetes is actively being developed and significant changes are made across releases. Features are dropped and added. For example, the *hostport* support has been disabled in recent versions of Kubernetes.

Special Considerations for Command Center Services

When Kubernetes manages the host name of the TIBCO MFT Command Center instance, the host name changes when Kubernetes starts a container. The following TIBCO MFT Command Center services require you to run a server with a specific host name:

- **Collection Service**
- **Status Service**

Do not assign a specific host name to these TIBCO MFT Command Center services because the hostname can change. You can set **Eligible Collection Server Hosts** to **All Command Centers** in the **Configure Collection Service** admin page. This permits any TIBCO MFT Command Center to run **Collection Service**.

Likewise, you can set **Eligible Collection Server Hosts** to **All Command Centers** in the **Configure Status Service** admin page. This permits any TIBCO MFT Command Center to run **Status Service**.

Note: The option to select **All Command Centers** is only displayed when all the TIBCO MFT Command Center instances are at version 8.5.0 or higher.

For more information about the **Collection Server** and **Status Service** configuration, see "Appendix E: Collection and Status Service - High Availability" topic in the *TIBCO MFT Command Center User Guide*.

Container Deployment Process

To deploy TIBCO MFT Command Center using Docker container, you must perform the following steps that are a part of the process. Each step is documented in more detail in the sections that follow.

Step 1: Select the Base Operating System

You must install TIBCO MFT Command Center on a Linux environment to create a TIBCO MFT Command CenterDocker image. It is recommended that you use the same Linux environment for the base installation that is used for the Docker container.

Step 2: Create Directory Structure for Container Image

Run the following commands to create directories:

```
mkdir -p /mftdocker/mftcc
mkdir -p /mftdocker/mftcc/java
mkdir -p /mftdocker/mftcc/tibco
mkdir -p /mftdocker/mftcc/tibco/mftcc
```

Note: Ensure the logged-in user has `write` permission on these directories.

Step 3: Install Java JDK

TIBCO MFT Command Center requires a Java JDK.

Procedure

1. Install a current version of the Java JDK: Oracle, Coretto, or OpenJDK. The best practice is to use Long-Term Support (LTS) versions like Java 21.
   The Dockerfile assumes that Java is installed in `/mftdocker/mftcc/java`
   - To install Java from a tar or `tar.gz` file in this directory, run the following command:

     ```
     tar xvf javajdk.tar.gz -C /mftdocker/mftcc/java --strip-
     components=1
     ```

- If Java is installed using RPM, copy the Java installation directory using the following command:

```
cp -R /usr/java/jdk-21.0.xx/* /mftdocker/mftcc/java
```

2. Before proceeding further with the installation, set the following environment variables:

```
export JAVA_HOME=/mftdocker/mftcc/java/
export PATH=$JAVA_HOME/bin:$PATH
```

Tip: Add the above lines to your profile so that these environment variables are set when you log in.

3. Verify that `JAVA_HOME` is correct by issuing the following command:

```
ls -la $JAVA_HOME
$JAVA_HOME/bin/java -version
```

Step 4: Create a Base TIBCO MFT Command Center Installation

For a first-time installation or an upgrade, follow the instructions provided below. For a hotfix installation, see Step 5: TIBCO MFT Command Center Install Hotfixes.

Before you begin

Ensure that the `JAVA_HOME` and the `PATH` variables are set. If they are not set, see Step 3: Install Java JDK.

Procedure

1. Copy or move the compressed TIBCO MFT Command Center installation file into the directory:

```
mv /tmp/TIB_mftcc.zip /mftdocker/mftcc/tibco/mftcc
```

2. To decompress the file, run the following commands:

```
cd /mftdocker/mftcc/tibco/mftcc
unzip TIB_mftcc.zip
```

3. Run the following command:

```
chmod +x expressinstall.sh
```

4. Run the Express Install script:

```
./expressinstall.sh
```

   Respond to the prompts to accept the End-User License Agreement.

   A preinstalled version is installed in the `./server` directory.

5. Make all scripts executable by running the following command:

```
find . -type f -name "*.sh" -print0 |xargs -0 chmod +x
```

6. To install a predefined keystore, follow the instructions in this step. Otherwise, skip to the next step. The preinstalled Command Center comes with a self-signed certificate with: `CN=localhost`

   Note: The HTTPS certificate is typically terminated at the load balancer, so that the Command Center can typically use the self-signed certificate.

   Run these commands to copy the new keystore:

```
cd /mftdocker/mftcc/tibco/mftcc/keystore

cp keystore.jks keystore.original.jks

cp /your/keystore.jks keystore.jks
```

   Verify the keystore password by running these commands:

```
cd /mftdocker/mftcc/tibco/mftcc/keystore
keytool -list -v -keystore keystore.jks
```

When prompted, enter the new keystore password.

If the new keystore password is not `changeit`, run these commands to set the keystore password:

```
cd /mftdocker/mftcc/tibco/mftcc/distribution/util/dbsettings
./keystorepwd.sh
```

When asked for the full path to the `server.xml` file, enter the following response:

```
/mftdocker/mftcc/tibco/mftcc/server/conf
```

7. Copy the required jar files to the installation directories.

   Copy the JDBC jar file to the following directories:

   - `/mftdocker/mftcc/tibco/mftcc/server/lib`
   - `/mftdocker/mftcc/tibco/mftcc/cloud/dbconfig/lib`

   Note: Optional: Copy JMS jar files to `/mftdocker/mftcc/tibco/mftcc/server/webapps/cfcc/WEB-INF/lib`

What to do next

Postinstallation tasks:

1. Customize Command Center logos and graphics as required.
2. Update web.xml parameters as needed.
3. See the Security Guide documentation to lock down your system.

Step 5: Install Current TIBCO MFT Command Center Hotfix

The `readme` file that accompanies a hotfix provides instructions to install the hotfix. You can install the hotfix directly over the base installation that you created in the previous step. It is a good practice to create a Docker image for each hotfix installation. That makes it easy for you to test and deploy different images.

Procedure

1. To install a hotfix, copy the required files to the base TIBCO MFT Command Center installation directory. Note that the hotfix installation `readme` file shows the files that should be copied to the TIBCO MFT Command Center directory. Optionally, you can unzip the compressed hotfix file directly into the installation directory: `/mftdocker/mftcc/tibco/mftcc`
2. Install the hotfix using the express installation:

```
cd /mftdocker/mftcc/tibco/mftcc
./install.sh 8.6.0_HF-001 express
```

Step 6: Test Current Installation

This is an optional step to test the installed Command Center. You can perform this to test the installation prior to creating the container image.

Note: This step requires connectivity to a database server and the database must be created according to the requirements of the MFT Command Center Installation Guide. The required MFT tables are created during the installation.

Procedure

1. Set the environment variables to point to the current database. These environment variables must be set:

```
export COM_TIBCO_MFT_CE_DB_TYPE
export COM_TIBCO_MFT_CE_DB_USER
export COM_TIBCO_MFT_CE_DB_PWD
export COM_TIBCO_MFT_CE_DB_URL
export COM_TIBCO_MFT_CE_DB_DRIVER
```

You can use the `clouddbconfig` script to display the environment variables. Run the following commands:

```
cd /mftdocker/mftcc/tibco/mftcc/cloud/dbconfig
./clouddbconfig.sh
```

Follow the instruction to **Add Database Server settings.**

Go to **Manage Database Server settings**.

Select **DB Config** and then select **Display Cloud Database Environment variables**. The environment variables are displayed.
You can use this script to display the encrypted password without creating the database entry:

```
cd /mftdocker/mftcc/tibco/mftcc/cloud/dbsettings
./clouddbconfig.sh encrypt abc123
```

Here is a sample script that you can use to set the environment variables and start the MFT Server.

Note: The following formats are valid for the COM_TIBCO_MFT_CE_DB_PWD environment variable:

- Encrypted
- Base 64 encoded
- Clear Text

```
# Export JAVA_HOME and PATH
export JAVA_HOME=/mftdocker/mftcc/java
export PATH=$JAVA_HOME/bin;$PATH
```

```
# Export the MFT database environment variables
export COM_TIBCO_MFT_CE_DB_TYPE=mysql
export COM_TIBCO_MFT_CE_DB_USER=cfcc
# If using Encrypted format
export COM_TIBCO_MFT_CE_DB_PWD=PWD:pz1clz83dAcdrJVBUYUu2P+vLyY=
# If using Base 64 encoded format
export COM_TIBCO_MFT_CE_DB_PWD=B64:YWJjMTIzCg==
# If using Clear Text format
export COM_TIBCO_MFT_CE_DB_PWD=CLR:abc123
# In this sample, we use the encrypted type format; you can use
either format for compatibility
export COM_TIBCO_MFT_CE_DB_PWD=/X5Zwxbjhabc%7-HJMy8eTVJfaVQpia/N1o=
export COM_TIBCO_MFT_CE_DB_
URL="jdbc:mysql://your.db.server:3306/mftdb?characterEncoding=UTF8&
useSSL=true&serverTimezone=UTC"
export COM_TIBCO_MFT_CE_DB_DRIVER=com.mysql.cj.jdbc.Driver
export COM_TIBCO_MFT_CE_KEYSTORE_PWD=uV9ZoAEILUIvQLwp9P3MMwjg2z0=
# Start MFT Server
server/bin/startup.sh
```

For more information about the environment variables, see Environment Variables section in the *Installation Guide*.

2. If Command Center has not been started by the script above, run the following commands to start Command Center.

```
cd /mftdocker/mftcc/tibco/mftcc
server/bin/startup.sh
```

Step 7: Prepare TIBCO MFT Command Center Docker Image

The container image is now ready. This step prepares the directory for containerization.

Procedure

1. Run the following commands to remove files to save space on the image.

```
rm /mftdocker/mftcc/tibco/mftcc/cfcc.jar
rm /mftdocker/mftcc/tibco/mftcc/server.jar
rm -R -f /mftdocker/mftcc/tibco/mftcc/distribution
rm -R -f /mftdocker/mftcc/tibco/mftcc/META-INF
rm -f /mftdocker/mftcc/tibco/mftcc/server/logs/*
rm -f /mftdocker/mftcc/tibco/mftcc/logs/trace/*
rm -f /mftdocker/mftcc/tibco/mftcc/logs/audit/*
rm -f /mftdocker/mftcc/tibco/mftcc/logs/webAdmin/*
rm -f /mftdocker/mftcc/tibco/mftcc/logs/webAdmin/*
rm -f /mftdocker/mftcc/tibco/mftcc/express.jar
rm -f /mftdocker/mftcc/tibco/mftcc/8.6.0_HF*.jar
```

2. Copy the Dockerfile from the distribution Cloud directory.
   MFT distributes the non-root and root DockerFiles.
   - To copy a non-root Dockerfile:
     `cp /mftdocker/mftcc/tibco/mftcc/cloud/k8s_samples/nonRoot/DockerFile.mftcc/mftdocker/mftccis`
   - To copy root Dockerfile:
     `cp /mftdocker/mftcc/tibco/mftcc/cloud/k8s_samples/DockerFile.mftcc` /mftdocker/mft`ccis`
3. Verify the contents of the Dockerfile.

   You can update the Linux version that you use. If you did not use the suggested directory structure, you need to update this file with your directory structure details.

Step 8: Build TIBCO MFT Command Center Docker Image

The Docker `build` command creates the Docker image using the Dockerfile.

Procedure

1. Change to the directory where the Dockerfile is located.

   `cd /mftdocker/mftcc`
2. Run the following command to build the Docker image:

```
sudo docker build -f Dockerfile.mftcc -t library/mftcc_8.6.0:v1 .
```

> **Note:** The previous command has a dot (.) at the end of the command. This is required when running the command.

3. Confirm if the image is present.

```
sudo docker images
```

Step 9: Test TIBCO MFT Command Center Docker Image

Now that the Docker image is created, you must test the Docker image using the `docker run` command. This step shows how to test the Docker image. Note that by default, the TIBCO MFT Command Center server connects to the database used in the initial TIBCO MFT Command Centerinstallation. The database connectivity can be changed later to point to a different database.

The Docker `run` command allows you to redirect the container ports to the ports on the Docker host.

> **Note:** Make sure that the ports defined by the `-p` parameter are unique and available on the Docker host when running multiple containers on the same Docker host.

Procedure

1. To test the HTTPS and HTTP ports, run the following command. In the Internet Server, if you want to test other protocols such as SFTP, you must add the port mapping to the following command:

```
docker run -it --rm -p 8443:8443 -p 8080:8080 -e \ COM_TIBCO_MFT_
TRACEDIR='/tmp' library/mftcc:v.v.v (where v.v.v is the product version)
```

> **Note:** The above command is for Internet Server only.

The following table describes the parameters in the `run` command:

Parameter	Description
`-p 8443:8443 -p 8080:8080`	Maps the ports used by the container to the ports accessible by the network.
`-e \ COM_TIBCO_MFT_TRACEDIR='/tmp -library/mftcc:v.v.v`	Defines the location where TIBCO MFT Command Centerlog files are created. TIBCO MFT Command Centercreates a directory for the machine name. Under this directory, TIBCO MFT Command Center writes MFT log and Server log files.

Note: If you have changed the Docker image that you want to make permanent, you can use the Docker `commit` command to commit the changes to an image. However, make sure to do this before the Docker image is removed.

Step 10: Execute TIBCO MFT Command Center Docker Image

Now that the Docker image is tested, you must run the Docker image using the Docker `run` command.

When you want to run a Docker image, you must perform the following steps:

1. Configure the database.
2. Map additional ports.
3. Run the Docker command to start the TIBCO MFT Command Centerserver container.

Procedure

1. To configure the database, see Step 10: Configure the Database.

- It prepares any database tables necessary for TIBCO MFT Command Center to run.
- It creates an XML file that contains the parameters necessary for TIBCO MFT Command Center to connect to the database. You must either pass the path to this XML file to the TIBCO MFT Command Center server in the `COM_TIBCO_MFT_CE_DBCFG` environment variable, or pass the environment variables defined in the file to docker.

2. To map additional ports, see the tables below.

   The following ports are used by Command Center and can be configured by the Internet Server and Command Center management tools. Ports shown are sample ports that can be changed.

Protocol	Command Center ports	Comments
Platform Server	46465	CC and IS must use different ports. On CC, this port can only run a Platform Server job.
FTP	N/A	Control port
FTP/S	N/A	Data ports
HTTP	8080	
HTTPS	8443	
Shutdown	8005	Internally used only

   The following ports are used by Internet Server and can be configured by the Internet Server and Command Center management tools. Ports shown are sample ports that can be changed.

Protocol	Internet Server ports	Comments
HTTP	7080	A load balancer typically redirects to this port.
HTTPS	7443	
Shutdown	7005	Internally used only
SFTP	22	The SFTP port is defined by the customer.
Platform Server	46464	Clear text port is defined by the customer.
	56565	SSL port is defined by the customer.
	58585	TLS tunnel port is defined by the customer.
FTP	21	The FTP control port is defined by the customer.
FTP/S	990	The FTP Implicit SSL control port is defined by the customer.
FTP/FTPS data Ports	30000-30100	The FTP data ports are defined by the customer.
OFTP2	6609	OFTP2 TLS port. **Note:** We do not recommend using FTP or FTPS when running in the cloud. The FTP requirement for separate Control and Data connections makes getting FTP to work in the cloud difficult. We recommend using SFTP instead.

3. To run the Docker command and start the TIBCO MFT Command Centerserver container, run the following command.

```
docker run -it --rm -p 8443:8443 -p 8080:8080 -p 4646546465 -e \
```

```
COM_TIBCO_MFT_TRACEDIR='/persist/CC_ServerA/MFTLogs' -e\

COM_TIBCO_MFT_CE_DBCFG='/persist/mftDBProd.xml' library/mftcc:v.v.v (where
v.v.v is the product version)
```

Step 11: Save TIBCO MFT Command Center Docker Image to a Repository

Now that the Docker image has been built, tested, and run, you can save it to a Docker registry. You must save the Docker image to a file or push the image to a repository to make the image available to all nodes.

There are many ways to make images available to all nodes. The following instructions show how this is done by saving the image to a file. If your company has a repository, you can use your repository to hold the TIBCO MFT Command Centerimages.

Procedure

1. Run the following Docker command:

   ```
   docker save -o mftccv.v.v.tar library/mftcc:v.v.v (where v.v.v is the
   product version)
   ```

   Note: If you do not specify a version, all version images are saved. Otherwise, only the defined image is saved. The command creates a file called `mftcc`*`v.v.v`*`.tar`.

2. (Optional) Copy `mftcc`*`v.v.v`*`.tar` to each Kubernetes node.

   Note: If you create a hotfix, use the format: `mftcc:`*`v.v.v`*`_HF-`*`nnn`* (where *nnn* is the hotfix number). When using a repository, you do not need to run this step.

   The image is now available to all the Kubernetes nodes.

3. (Optional) From the Kubernetes nodes, load the Docker image files using the

following command.

```
docker load -i mftccv.v.v.tar
```

Step 12: Configure the TIBCO MFT Command CenterDatabase

Each time you install a new version, upgrade an existing version or install a hotfix, update the database with any new tables or columns.
When the Docker image is created, it points to a test database. However, on running the image, it must point to a different database. There are three instances where you should update the TIBCO MFT Command Center database:

1. When you install MFT for the first time on a database. The necessary tables and records are created in the database.
2. When you upgrade the database to a new version or release. New tables and columns may be created.
3. When you install an MFT hotfix on a previously installed version. New columns might be created.

To configure the TIBCO MFT Command Center database, see the following sections:

Procedure

1. Scripts to Update the Database
2. Sample Script Execution
3. Execute Commands to Update the Database
4. Database Migration Utility

Scripts to Update the TIBCO MFT Command Center Database

You can create the database tables used by MFT in the following ways:

Note: Before creating the database tables, you must create the MFT database. See*TIBCO Managed File Transfer Command Center Installation* manual for details to create a database.

- Each time an Internet Server or Command Center instance starts, the instance checks if the tables are created and at the correct release or hotfix level. If the tables are not created or not at the proper release or hotfix level, they are created or upgraded respectively.
  This is the preferred method to install or upgrade the MFT Database tables.
- You can use the clouddbsettings utility to update the tables. The instructions for using this method are documented in this section. This is documented for compatibility with prior versions. We suggest using the previous option.

TIBCO MFT Command Center provides two scripts to allow you to create or update database tables for the Docker image (one for UNIX and another for Windows).

Two scripts are provided with the MFT distribution to prepare the database. These scripts are located in the `<MFT-Install>/cloud/dbsetup` folder.

- Windows: `clouddbconfig.bat`
- UNIX: `./clouddbconfig.sh`

Note: When you run the script to update the database, the computer where the script is executing must have connectivity to the database. Additionally, you can only use the same database type (Oracle, MS SQL, mySQL, among other databases) and the JDBC driver that was used when you created the initial Docker image.

These scripts allow you to perform the following actions:

1. Define the properties of the TIBCO MFT Command Center database connection.
2. Save the database connection properties in a file.
3. Update the database connection properties.
4. Update the database with new tables and columns required by this MFT release or hotfix.
5. Test the database connection.

6. Generate the cloud database XML file required by TIBCO MFT Command Center containers to connect to the correct database.

New Command-Line Arguments

Procedure

1. Create the following entry in the xml file for the database connection information.

```
clouddbconfig action:create arg:value arg:value ...
```

The following table lists the supported arguments:

Argument	Argument Type	Valid Values	Required
db	Database type	mssql|mysql|postgresql|oracle	Yes
name	Name for this entry in the xml file.		Yes
url	Database connection URL		Required if dbhost and dbname are not specified.
dbname	Database name		Required if url is not specified.
userid	ID of user with access to database		Yes

Argument	Argument Type	Valid Values	Required
password	Password for the user ID		Yes

driver	Database JDBC driver class name	Oracle:
		`oracle.jdbc.driver.OracleDriver`
		Postgres:
		`org.postgresql.Driver`
		SQLServer:
		`com.microsoft.sqlserver.jdbc.SQLServerDriver`(Microsoft JDBC Driver)
		`net.sourceforge.jtds.jdbc.Driver`(JTDS JDBC Driver)
		MySQL:
		• `com.mysql.cj.jdbc.Driver` (version 8 and later)
		• `com.mysql.jdbc.Driver`(version 5 and earlier)
		• `org.mariadb.jdbc.Driver`(MariaDB)
			Required if not default.

drivertype	The type of JDBC driver if using Microsoft SQL server.	microsoft|jtds	Required for Microsoft SQL server only.
dbhost	The host name of the database server.		Required if URL is not specified.

Argument	Argument Type	Valid Values	Required
dbport	The port the database is listening on.		Required if URL is not specified and is not using the default port.

2. Run the following script to create or update the database for an entry in the xml file:

```
clouddbconfig action:updatedb name:<xml file entry name>
```

3. Run the following script to delete an entry in the xml file:

```
clouddbconfig action:delete name:<xml file entry name>
```

4. Run the following script to generate the database connection file used by cloud version of MFT:

```
clouddbconfig action:generatexml name:<xml file entry name>
file:<name of file to write entry>
```

> **Note:** If the file argument is not specified, then the file name is the name of the entry.

5. Run the following script to display an entry in the xml file:

```
clouddbconfig action:list name:<xml file entry name>
```

6. Run the following script to list all the entries in the xml file:

```
clouddbconfig action:listall
```

7. Run the following script to encrypt a password:

```
clouddbconfig encrypt <password>
```

> **Note:** The encrypted password can be piped into an export command to set the `DBPass` environment variable.

Sample Script Execution

Before you use the cloudconfig script, you must copy the jdbc jar file to the following directory: *`MFT Install`*`/cloud/dbconfig/lib`

This script can run for UNIX or Windows, regardless of where TIBCO MFT Command Center is installed. So even if TIBCO MFT Command Center is installed on Linux in the DMZ, you can update the database in Windows, as long as Windows has connectivity to the database. This script also creates a log file of all commands entered. The file created is called:

`MFT-Install``/cloud/dbsettings/db-install.YYYY.MM.DD.HH.MM.SS.log`

To get help information on this script, enter the following command:

```
clouddbsetup.sh help
```

The clouddbconfig script contains the following formats:

1. Create and manage database connections with an interactive menu.
   `clouddbconfig`
2. Run a database update of an MFT database (with confirmations).
   `clouddbconfig config name`
3. Run a database update of an MFT database in silent mode (with no confirmations).
   `clouddbconfig` *`config name`* `silent`

> **Note:** When performing a database update, the database is updated to the level associated with the MFT installation where the dbcloudconfig files are located.

Two different files can be updated or created by this script:

1. `db-settings.xml`. An entry is created in this file for each config entry created. This entry can be used to update the MFT database, either through the interactive script or the automated scripts (silent and with confirmations)
2. An xml file generated by selecting "Generate Cloud Database XML file". This file is fed into the Docker Container through the COM_TIBCO_MFT_CE_DBCFG environment variable. This environment variable can be defined in one of the following ways:
    a. Through the environment variable when starting a Docker Container
    b. Through the spec:env parameter defined in the Kubernetes YAML file

TIBCO MFT Command Center detects this environment variable at startup, reads this file, and substitutes the DB setup parameters defined in the `web.xml` file with the parameters defined in this file.

Note:

- This file must be saved to the persistent storage accessible by the MFT container.
- If you want to use a PostgreSQL database, the CITEXT extension must be installed. To enable the CITEXT data type, you must install the PostgreSQL CITEXT extension and run the following SQL command as a superadmin for the PostgreSQL MFT database:
  `CREATE EXTENSION IF NOT EXISTS CITEXT;`

See the following information to create a database server configuration entry:

```
clouddbconfig.bat

Main Menu
=========
1. Add Database Server settings
2. Manage Database Server settings
3. Save your changes
4. Exit

Please enter the number corresponding to your selection: 1

1. Add database server configuration entry
==========================================

Enter a name for the database configuration settings entry: mftqadb
```

```
Select database server type:
Enter 1 for MSSQL
Enter 2 for MySQL Enterprise Server or Community Server
Enter 3 for Oracle
Enter 4 for PostgreSQL
: 2

MySQL selected as database server type.

Enter the DNS name or IP Address of the database server...
[localhost]:mysql.qasvr
Enter the database port number.....................................[3306]:
Enter the database name.....................................
[cfcc]:mftqadb
Enter the database UserID.....................................[cfcc]:
Enter the database Password.....................................[cfcc]:
Enter if you want database connections to use TLS? y/n.....[n]:
Please confirm the password:

Use database URL:
[jdbc:mysql://mysql.qasvr:3306/mftqadb?characterEncoding=UTF8&useSSL=fal
se]? y/n [y] :

Verifying database connection using the following URL:

jdbc:mysql://mysql.qasvr:3306/mftqadb?characterEncoding=UTF8&useSSL=fals
e

Connected successfully to mftqadb

mftqadb added successfully.

Press enter for menu options

Main Menu
=========
1. Add Database Server settings
2. Manage Database Server settings
3. Save your changes
4. Exit

Please enter the number corresponding to your selection: 3

Changes saved successfully.
```

See the information below to create a Cloud Database XML file:

```
clouddbconfig.bat
Main Menu
=========
1. Add Database Server settings
2. Manage Database Server settings
3. Save your changes
4. Exit
Please enter the number corresponding to your selection: 2
Manage Servers
==============
1. mft830  [ mysql,
jdbc:mysql://localhost:3306/mft830?characterEncoding=UTF8&useSSL=false&s
erverTimezone=UTC ]
Please enter the number of the server to view details and manage or 'X'
to return to main menu: 1
Server Details [mft830]
==============
Name:                  mft830
DB Type:               mysql
DB User Id:            root
DB Password:           *****
DB Connection URL:
jdbc:mysql://localhost:3306/mft830?characterEncoding=UTF8&useSSL=false&s
erverTimezone=UTC
DB Driver ClassName:  com.mysql.jdbc.Driver
Press enter for menu options
Server Detail Menu [mft830]
===================
1. View Entry
2. Update Entry
3. Copy Entry
4. Delete Entry
5. Test Database Connection
6. Generate Cloud Database XML file
7. Display Cloud Database Environment variables
8. Update the Database
9. Save your changes
10. Back to Main Menu
11. Exit
Please enter the number corresponding to your selection: 6
Enter the name of the file to write the cloud entry to:
c:\temp\mftqa.xml
c:\temp\mftqa.xml created successfully.
```

The `mftqadb.xml` file had the following contents:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<para>
    <!--the DB driver name-->
    <DBDriver>com.mysql.jdbc.Driver</DBDriver>
    <!--DB URL connection string-->

<DBConn>jdbc:mysql://localhost:3306/mft830?characterEncoding=UTF8&amp;us
eSSL=false&amp;serverTimezone=UTC</DBConn>
    <!--DB user-->
    <DBUser>root</DBUser>
    <!--DB password, encrypted with MFT install routine. For test, can
use clear text password by using clear:pwd-->
    <DBPass>YGimk+0kDKwvl3sih4AgOveXicg=</DBPass>
    <!--DB Type, valid values: mysql, mssql, oracle, or postgresql-->
    <!--If a valid value is specified, MFT will try to create tables for
an empty DB during start up time. Otherwise, MFT will not try to create
tables during start up time.-->
    <DBType>mysql</DBType>
</para>
```

Execute Commands to Update the TIBCO MFT Command CenterDatabase

Procedure

1. You can update the database by using one of the following ways:
    - Run the command with confirmations

      ```
      clouddbconfig mftqadb
      ```

    - Run the silent command with no confirmations

      ```
      clouddbconfig mftqadb silent
      ```

    - Run the following interactive command

```
(Note: The first few menus are not displayed)
Manage Servers
==============
1. mft830  [ mysql,
jdbc:mysql://localhost:3306/mft830?characterEncoding=UTF8&useSS
L=false&serverTimezone=UTC ]
Please enter the number of the server to view details and
manage or 'X' to return to main menu: 1
Server Details [mft830]
==============
Name:                 mft830
DB Type:              mysql
DB User Id:           root
DB Password:          *****
DB Connection URL:
jdbc:mysql://localhost:3306/mft830?characterEncoding=UTF8&useSS
L=false&serverTimezone=UTC
DB Driver ClassName:  com.mysql.jdbc.Driver
Press enter for menu options
Server Detail Menu [mft830]
==================
1. View Entry
2. Update Entry
3. Copy Entry
4. Delete Entry
5. Test Database Connection
6. Generate Cloud Database XML file
7. Display Cloud Database Environment variables
8. Update the Database
9. Save your changes
10. Back to Main Menu
11. Exit
Please enter the number corresponding to your selection: 8
Are you sure you want to update the database for mft830?: y/n
[n] y
Database will be modified for new features.  Please backup
database before proceeding.
Do you wish to continue? y/n [y]
setupSIFTDB running directory: C:\MFT830\cc\cloud\dbconfig
distribution directory: C:\MFT830\cc\cloud
installation directory: C:\MFT830\cc
Install directory: C:\MFT830\cc
```

```
(many DB Update messages are displayed, followed by these
messages:)

Successfully populated DB tables with default information.
Database updated successfully.
```

Database Migration Utility

The Database Migration Utility migrates table data from one MFT database to another. It supports Oracle, MySQL, MS-SQL, and PostgreSQL databases. You can migrate between similar databases (for example, from a source Oracle database to a destination Oracle database) or different databases (for example, from a source Oracle database to a destination PostgreSQL database). This utility simplifies moving MFT from an on-premises installation to a cloud installation by allowing you to copy an on-premises database to a cloud database.

Restrictions

- Database Migration Utility is designed to migrate only MFT databases.
- Database Migration Utility can only migrate entire tables, not individual rows.
- You must create the destination database before migrating data. The utility supports creating the destination database.
- The destination MFT installation must either match or be of a later release than the source MFT installation.

Functionality

- **Create Database Tables:** Database Migration Utility can create MFT DB tables for the selected tables.
- **Purge Database Tables:** Database Migration Utility can delete all rows from the selected tables.
- **Migrate Database Data:** Database Migration Utility can migrate data from the selected tables.
- **Encrypt Config File Passwords:** Database Migration Utility can encrypt clear text passwords in the config file.

Configuration

The `db-migrate.properties` configuration file defines the source database and destination database connection properties.

Config file

The db-migrate.properties configuration file is located in the `<Installation Directory>/cloud/dbconfig/config` directory. Lines beginning with the **#** character are treated as comments.

The MFT distribution has the following sample configuration file provided with it:

```
#
# Source Database
Settings#source.url=jdbc:oracle:thin:@<hostname>:1521/<service name>
#
source.driver=oracle.jdbc.driver.OracleDriver
source.userid=<user id>
source.password=<optional plaintext password>
source.type=oracle
source.mft.version=8.6
#
# Destination Database Settings
#
destination.url=jdbc:postgresql://<hostname>:5432/<database
name>?stringtype=unspecified
destination.driver=org.postgresql.Driver
destination.userid=<user id>
destination.password=<optional plaintext password>
destination.type=postgres
destination.mft.version=8.6
#
# Source Tables
#
```

```
source.tables=alertaudits,Alerts,As2SrvCfg,Audits,AuditFTs,AuditsExt,Aud
itsExtAs2,AuditsExtOFTP2,
Authenticators,CfSrvCfg,Checkpt,CollectorSrvCfg,Config,ConfigU2U,ConnMgr
Node,Dept,Devices,DocDeleted,
DocRevisions,DocShare,Documents,EncryptKeys,ErrorEvents,Events,Files,FSE
vents,FtpSrvCfg,FTTransfers,
HostStats,JMSSrvCfg,JMSSrvCfgExt,LicenseKey,LoginAudit,Messages,MFAConfi
g,MFAMFTCfg,MFTStatistics,
NodeFTs,Nodes,NodesExt,Notifications,OFTP2SrvCfg,OIDCAuthServer,OIDCMFTC
fg,OnDemandSecurity,
OnDemandUsers,PCILog,PCILogDetail,Profiles,QRTZ_JOB_DETAILS,QRTZ_
TRIGGERS,QRTZ_BLOB_TRIGGERS,QRTZ_CALENDARS,QRTZ_CRON_TRIGGERS,QRTZ_
FIRED_TRIGGERS,QRTZ_LOCKS,QRTZ_PAUSED_TRIGGER_GRPS,QRTZ_SCHEDULER_
STATE,QRTZ_SIMPLE_TRIGGERS,QRTZ_SIMPROP_
TRIGGERS,Roles,SAMLConfig,SchedulerSrvCfg,SearchAudits,ServiceLock,Sessi
ons,SrvStatusCfg,
SSHAlgorithmGroup,SshSrvCfg,U2URequests,Attachments,Recipients,UserProfi
les,Users,UsersMap

#

# Commit Count

#

commit.count=500
```

Explanation of Properties

The following table provides the list of allowed parameters for Database Migration Utility.

Parameter Name	Parameter Description
source.url	Defines the DB connection URL for the source database. See the following examples for some sample DB URLs: Oracle • Service name format: Oracle jdbc:oracle:thin:@host:1521/orclpdb • SID format: jdbc:oracle:thin:@host:1521:orcl PostgreSQL • jdbc:postgresql://host:5432/testdb?stringtype=unspecified&currentSchema=public

Parameter Name	Parameter Description
	MySQL • jdbc:mysql://host:3307/testdb?characterEncoding=UTF8&useSSL=false&serverTimezone=UTC&defaultFetchSize=500&useCursorFetch=true MS SQL • jdbc:sqlserver://host:1433;databaseName=testdb
source.driver	Defines the class name of the source DB Driver. See the following examples for some sample drivers: Oracle • oracle.jdbc.driver.OracleDriver PostgreSQL • org.postgresql.Driver MS SQL: • com.microsoft.sqlserver.jdbc.SQLServerDriver MySQL: • com.mysql.cj.jdbc.Driver (for MySQL version 8 and above). • com.mysql.jdbc.Driver (for MySQL version 5).
source.userid	A user with rights to access the MFT database. The source user must have read rights to the tables.
source.password	The password associated with the user ID. You have three options for entering the password: • Leave the password field blank and be prompted to enter it when required. • Enter the password in plain text, run the utility, and select the **Encrypt Config File Passwords** option to encrypt the passwords. • Enter the password in plain text and delete it after completing

Parameter Name	Parameter Description
	the migration.
source.type	Defines the source database type. The following are the valid values for the database type: • oracle • mysql • postgres • mssql **Important:** Ensure that you enter the value in lowercase.
source.tables	Defines the tables to be processed. The order is important and should not be changed as some tables depend on other tables. You can remove tables if you only want to process certain tables.
source.mft.version	Defines the version of MFT for the source tables. The valid MFT version values are 8.3, 8.4, 8.5, or 8.6.
destination.url	Defines the DB connection URL for the destination database. See the following examples for some sample DB URLs: Oracle • Service name format: jdbc:oracle:thin:@host:1521/orclpdb • SID format: jdbc:oracle:thin:@host:1521:orcl PostgreSQL • jdbc:postgresql://host:5432/testdb?stringtype=unspecified&currentSchema=public MySQL • jdbc:mysql://host:3307/testdb?characterEncoding=UTF8&useSSL=

Parameter Name	Parameter Description
	false&serverTimezone=UTC&rewriteBatchedStatements=true MS SQL • jdbc:sqlserver://host:1433;databaseName=testdb
destination.driver	Defines the class name of the destination DB driver. See the following examples for some sample drivers: Oracle • oracle.jdbc.driver.OracleDriver PostgreSQL • org.postgresql.Driver MS SQL: • com.microsoft.sqlserver.jdbc.SQLServerDriver MySQL: • com.mysql.cj.jdbc.Driver (for MySQL version 8 and above). • com.mysql.jdbc.Driver (for MySQL version 5).
destination.userid	A user with rights to access the MFT database. If using the Create Database function, the destination user must have the rights to create tables. Otherwise, only write access is required.
destination.password	The password associated with the user ID. You have three options for entering the password: • Leave the password field blank and be prompted to enter it when required. • Enter the password in plain text, run the utility, and select the **Encrypt Config File Passwords** option to encrypt the passwords. • Enter the password in plain text and delete it after completing the migration.

Parameter Name	Parameter Description
destination.type	Defines the destination database type. The following are the valid values for the database type: • oracle • mysql • postgres • mssql **Important:** Ensure that you enter the value in lowercase.
destination.mft.version	Defines the version of MFT for the destination tables. The valid MFT version values are 8.3, 8.4, 8.5, or 8.6.
commit.count	Defines the number of records to update before committing. Increasing this number may improve performance, but the best practice is to use 500 as the value.

Setup

Complete the following steps before running the utility:

1. Copy the database driver JAR files for the source and destination databases to the `<MFT Install>/cloud/dbconfig/lib` directory. If both databases are of the same type, copy only one JAR file. If both databases are different, copy both the JAR files.
2. Edit the db-migrate.properties file in the `<MFT Install>/cloud/dbconfig/config` directory.
   See the configuration file parameters provided in the *Explanation of Properties* section.

The DB Migration Utility is located in the `<Installation Directory>/cloud/dbconfig` directory.

Two scripts are available:

```
db-migrate.bat for Windows
```

```
db-migrate.sh for Unix/Linux
```

When running on Unix or Linux, you may need to run the following command to make the script executable.

```
chmod +x db-migrate.sh
```

Migrating Data

The utility uses a text-based menu. The following is a sample of the main menu:

```
MFT Database Migration Utility
------------------------------
Main Menu
=========
1. Create Database Tables
2. Purge Database Tables
3. Migrate Database Data
4. Encrypt Config File Passwords
5. Help
6. Exit
Please enter the number corresponding to your selection:
```

Explanation of Options

Option	Description
Create Database Tables	Use this option if the destination database does not have the necessary tables. This step creates all the required tables for the current version of MFT and purges the default rows. You are prompted to confirm before proceeding. All tables are created, regardless of the tables defined by the **`source.tables`** config parameter.

Option	Description
Purge Database Tables	This option removes all records from the database tables, which is necessary for Database Migration Utility to work. **Note:** When migrating to an existing destination database, remove all the data from the destination tables to avoid migration failure. Similarly, if a database migration partially completes, remove all the data from the destination tables before you retry. You are prompted to confirm before proceeding. Only tables defined by the **`source.tables`** config parameter are purged.
Migrate Database Data	This option copies data from the source database to the destination database. Only tables defined by the **`source.tables`** config parameter are migrated. You are prompted to confirm before proceeding. The migration duration depends on the number of records.
Encrypt Config File Passwords	This option encrypts the source password and destination password. If a password is already encrypted, it is not encrypted again.
Help	Displays helpful information needed to configure the utility.
Exit	Exits the utility. When you exit the utility, a log file is generated. The name and location of the file is displayed on the console.

Step 13: Prepare the Kubernetes Environment

Procedure

1. Set up an NFS (Network File System) for TIBCO MFT Command Centerfiles that is persistent and not deleted when the Docker container is terminated. Make sure that all Kubernetes nodes and the 172.0.0.0/8 network can access the file share with write

permissions.

> **Note:** If you do not require persisting MFT traces and logs, avoid setting up the persistent storage.

- Add the following variables to the existing variables in the environment section of your Kubernetes YAML file. For more information about these variables, see Important YAML Parameters.

```
name: COM_TIBCO_MFT_CE_DB_TYPE
value: "mysql"
name: COM_TIBCO_MFT_CE_DB_URL
value:
"jdbc:mysql://localhost:3306/gwlmft821?characterEncoding=UTF8&u
seSSL=false&serverTimezone=UTC"
name: COM_TIBCO_MFT_CE_DB_USER
value: "root"
name: COM_TIBCO_MFT_CE_DB_PWD
value: "KIt6zmvY8sfpxerqN6HGlj21CPo="
name: COM_TIBCO_MFT_CE_DB_DRIVER
value: "com.mysql.jdbc.Driver"
```

- Alternatively, you can run the following Docker commands. For more information about these environment variables, see Important YAML Parameters.

```
-e COM_TIBCO_MFT_CE_DB_TYPE='mysql'
-e COM_TIBCO_MFT_CE_DB_
URL='jdbc:mysql://localhost:3306/gwlmft821?characterEncoding=UT
F8&useSSL=false&serverTimezone=UTC'
-e COM_TIBCO_MFT_CE_DB_USER='root'
-e COM_TIBCO_MFT_CE_DB_PWD='KIt6zmvY8sfpxerqN6HGlj21CPo='
-e COM_TIBCO_MFT_CE_DB_DRIVER='com.mysql.jdbc.Driver'

For example:

docker run -it --rm -p 7443:7443 -p 7080:7080 -p -e COM_TIBCO_
MFT_CE_DB_TYPE='mysql' -e
COM_TIBCO_MFT_CE_DB_
URL='jdbc:mysql://localhost:3306/gwlmft821?characterEncoding=UT
F8&useSSL=false&serverTimezone=UTC' -e
```

```
COM_TIBCO_MFT_CE_DB_USER='root' -e COM_TIBCO_MFT_CE_DB_
PWD='KIt6zmvY8sfpxerqN6HGlj21CPo=' -e
COM_TIBCO_MFT_CE_DB_DRIVER='com.mysql.jdbc.Driver'
library/mftis8.3.0:v5
```

Note: You can use the following option in the clouddbconfig Server Detail Menu to display the environment variables in Kubernetes and Docker formats:

```
7. Display Cloud Database Environment Variables
```

2. Where necessary, update the following files with NFS server information:

   MFT-Install/cloud/k8s_samples/yaml_default/mftcc_svc.yaml

3. Update the YAML files.

   TIBCO MFT Command Center provides the following YAML file in the *MFT-Install*/cloud/k8s_samples/yaml_default/ directory. You must review this and update the file where necessary.

   mftcc_svc.yaml Service and deployment for mftcc (only use this if you want to create service for mftcc)

   Default settings have been made for the ports.

   We suggest that you review the following parameters in the `mftis_svc.yaml`, and `mftcc_svc.yaml` files.

   Different Kubernetes environments have different requirements. You must configure your Command Centers and Internet Servers as services or pods based on the requirements of your installation.

   For more information, see Important YAML Parameters.

4. Copy the Docker YAML files to a directory on the Kubernetes master server. Make a directory to save the YAML files.

   ```
   mkdir /mftk8s
   ```

   ```
   cp MFT-Install/cloud/k8s_samples/yaml_default/mftcc_svc.yaml /mftk8s
   ```

5. Copy the file defined by COM_TIBCO_MFT_CE_DBCFG to the proper place in persistent storage.

Important YAML Parameters

The YAML files provided are sample files designed to provide an example of a working Kubernetes network. You must update these files to meet the needs of your environment.

Parameter	Description
spec: image:	Defines the name of the Docker image that is loaded. Replace the default image name with the required value.
spec: ports	Defines the ports that are used. In the service (_svc) yaml files, the ports are defined in two places. **Note:** It is critical that the Command Center and Internet Server ports are different. Otherwise, Internet Server requests may be incorrectly forwarded to the Command Center and vice versa. Ports should be the same across all TIBCO MFT Command Centercontainers. But the ports used cannot be used by TIBCO MFT Command Center.
spec: replicas	Defines how many instances are started.
spec: env	Defines environment variables that are set by Kubernetes. These environment variables provide the information for TIBCO MFT Command Centerand to connect to the database and define persistent storage that is used for log files. **Note:** These parameters only need to be updated if the persistent storage directory is changed. The following environment parameters are supported: • COM_TIBCO_MFT_CE_DBCFG: Defines the location of the file that defines the connectivity information to the MFT database. This file was created in "Step 10: Configure the TIBCO MFT Command Center Database" by executing "Generate Cloud Database XML file". This file must then be copied to persistent

Parameter	Description
	storage so that TIBCO MFT Command Center can access this file at initialization. This parameter must be defined if you want TIBCO MFT Command Center to connect the TIBCO MFT Command CenterServer to a specific database. If this parameter is not defined, the database settings in the `web.xml` file is used. • COM_TIBCO_MFT_CE_TEMPLATENAME: Defines the name of the template used to create Internet Server FTP, Platform, AS2, and SFTP Services. If this parameter is not defined, the default template name is used: "*DefaultTemplate". If you want to have different settings for the Command Center containers, you can define a new template name. If the template name defined to Command Center is not in the database, it is created when TIBCO MFT Command Centerstarts. • COM_TIBCO_MFT_TRACEDIR: Defines the directory where all log files are saved. This is generally set to a directory in the persistent storage. TIBCO MFT Command Center attempts to create the directory. Make sure that TIBCO MFT Command Center has the necessary rights to create and update files in this directory. • COM_TIBCO_MFT_CE_DB_TYPE: Defines the type of cloud database used. • COM_TIBCO_MFT_CE_DB_URL: Defines the URL of the cloud database. • COM_TIBCO_MFT_CE_DB_USER: Defines the cloud database user. • COM_TIBCO_MFT_CE_DB_PWD: Defines the cloud database password. • COM_TIBCO_MFT_CE_DB_DRIVER: Defines the cloud database driver.

Parameter	Description
	Note: If the following 5 variables are defined, the COM_TIBCO_MFT_CE_DBCFG variable is ignored. COM_TIBCO_MFT_CE_DB_TYPE; COM_TIBCO_MFT_CE_DB_URL; COM_TIBCO_MFT_CE_DB_USER; COM_TIBCO_MFT_CE_DB_PWD; COM_TIBCO_MFT_CE_DB_DRIVER Use the clouddbconfig utility to generate these values. Or on the Server Detail Menu, enter 7 for the option **Display Cloud Database Environment variables** These environment variables should be set in the Kubernetes YAML config file. See step 1 in Prepare the Kubernetes Environment.
spec: volumeMounts	Defines the persistent storage directory that will be mounted.
spec: volumes	Defines the information required to mount an NFS share.

External Ports

If you are using auto scaling, update file `mftis-hpa.yaml` to define the scaling values. The scaling values in the default file are very low to facilitate testing this feature.

The following list provides the external ports defined in the YAML files. If you want to use different ports, then you must update the ports in the YAML files.

Command Center

- 31443 HTTPS
- 31480 HTTP
- 31464 Platform Server
- 31465 Platform Server SSL
- 31466 Platform Server Tunnel
- 30122 SFTP/SSH

Note:
- The ports should be the same across all TIBCO MFT Command Center instances.
- The ports defined as the FTP Data Ports must match the ports defined in the **Admin System Configuration > Global FTP Settings > Starting Port** and **Number of Ports to Use**.

Health Check Page

The health check page for TIBCO MFT Command Centerdoes not require authentication.

TIBCO MFT Command Center issues a simple SQL request to check if the database connection is working. The HTTP response code indicates the status of TIBCO MFT Command Center.

- 200 HTTP response code: Successful status
- 500 HTTP response code: Failure status

TIBCO MFT Command Center retains the status of a database connection for 60 seconds, so it only verifies the database connection at most, every 60 seconds.

Step 14: Deploy the Container

To deploy the container, perform the following steps:

Procedure

1. Run the following command to start instances:

```
kubectl create -f mftcc_svc.yaml
```

2. Run the following command to stop instances:

```
kubectl delete -f mfcc_svc.yaml
```

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the Product Documentation website, mainly in HTML and PDF formats.

The Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The documentation for this product is available on the TIBCO® Managed File Transfer Command Center Documentation page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our product Support website.
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the product Support website. If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature

requests from within the TIBCO Ideas Portal. For a free registration, go to TIBCO Community.

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

TIBCO, the TIBCO logo, the TIBCO O logo, and Slingshot are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (https://www.cloud.com/legal) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: https://scripts.sil.org/OFL

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.