[ESB] Preproduction setup

Last edit: @Jul 19, 2019

1. Configure MariaDB in API monitoring backend server

1.1 Server preparation

Connect to ***UN-API-T-MONDS-001.edc.un.org*** using SSH.

Create the folder for installation

```
sudo su -
mkdir -p /opt/software/sources
```

1.2 Install and configure MariaDB Galera (standalone)

Import with WinSCP from production (***UN-API-P-MONDS-001.edcv.un.org***) to your home folder in ***UN-API-T-MONDS-001*** the following files and copy them to ***/opt/software/sources/*** folder:

```
/opt/software/mariadb-galera-5.5.46-linux-x86_64.tar.gz
```

```
cp /home/global.un.org/crodrigo/mariadb-galera-5.5.46-linux-x86_64.tar.gz /opt/software/sources/
```

Create the ***mysql*** user and group:

```
groupadd mysql
useradd -g mysql mysql
```

Copy the MariaDB Galera tar file you imported from production to ***/usr/local:***

```
cp /opt/software/sources/mariadb-galera-5.5.46-linux-x86_64.tar.gz /usr/local/
```

Decompress the MariaDB Galera file and create a symbolic link to ***mysql*** folder:

```
cd /usr/local
```

```
tar -zxvpf mariadb-galera-5.5.46-linux-x86_64.tar.gz
ln -s mariadb-galera-5.5.46-linux-x86_64 mysql
```

Change ownership of folders and create new ones for MariaDB datafiles and socket:

```
chown -R mysql:mysql mysql
cd mysql
chown -R mysql:mysql .
mkdir /var/run/mysqld
chown -R mysql:mysql /var/run/mysqld
mkdir /var/lib/mysql/
chown -R mysql:mysql /var/lib/mysql
```

Run the script to install the database:

```
./scripts/mysql_install_db --user=mysql --ldata=/var/lib/mysql/
```

Start the service:

```
./bin/mysqld --user=mysql --datadir=/var/lib/mysql/ --socket=/var/lib/mysql/mysql.sock &
```

Connect with ***mysql*** user to the database:

```
sudo su - mysql
cd /usr/local/mysql
./bin/mysqladmin -u root -password <password> --socket=/var/lib/mysql/mysql.sock
./bin/mysql -u root -p --socket=/var/lib/mysql/mysql.sock
```

Once connected, create the users that the application will use:

```
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'10.130.95.%' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'10.130.232.%' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'localhost' IDENTIFIED BY '<password>';

GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'un-api-t-gw-001.edc.un.org' IDENTIFIED BY '8m6zvopdRwEDc';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'10.130.131.155' IDENTIFIED BY '8m6zvopdRwEDc';

GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'un-api-t-gw-002.edc.un.org' IDENTIFIED BY '8m6zvopdRwEDc';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'10.130.131.156' IDENTIFIED BY '8m6zvopdRwEDc';

GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'un-api-t-mosrv-001.edc.un.org' IDENTIFIED BY '8m6zvopdRwEDc';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'10.130.131.191' IDENTIFIED BY '8m6zvopdRwEDc';

GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'10.130.131.191' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'inthubadmin'@'10.130.95.%' IDENTIFIED BY '<password>' WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'inthubadmin'@'10.130.232.%' IDENTIFIED BY '<password>' WITH GRANT OPTION;
```

Passwords are stored in Keepass

Stop MariaDB service

```
./bin/mysql -u root -p --socket=/var/lib/mysql/mysql.sock shutdown
```

Exit from ***mysql*** user session to ***root*** and start again the MariaDB service:

```
exit
```

```
/usr/local/mysql/bin/mysqld --user=mysql --datadir=/var/lib/mysql/ --socket=/var/lib/mysql/mysql.sock &
```

2. Configure MariaDB Galera cluster in API backend servers

2.1 Configure primary server (UN-API-T-DB-001)

2.1.1 Server preparation

Connect to ***UN-API-T-DB-001.edc.un.org***

Create the folder for installation

```
sudo su -
mkdir -p /opt/software/sources
```

2.1.2 Install and configure MariaDB Galera (cluster)

Import with WinSCP from production (***UN-API-P-DB-001.edcv.un.org***) to your home folder in ***UN-API-T-DB-001*** the following files and copy them to ***/opt/software/sources/*** folder:

```
/opt/software/mariadb-galera-5.5.46-linux-x86_64.tar.gz
/etc/my.cnf
```

```
cp /home/global.un.org/crodrigo/mariadb-galera-5.5.46-linux-x86_64.tar.gz /opt/software/sources/
cp /home/global.un.org/crodrigo/my.cnf /opt/software/sources/
```

Edit ***my.cnf*** and change hostnames and other variables in the WSREP section (see highlighted fields)

```
vi /opt/software/sources/my.cnf
```

```
# WSREP parameter

# wsrep_provider                = none                                # Start mysqld without Galera
wsrep_provider                = /usr/local/mariadb-galera-5.5.46-linux-x86_64/lib/libgalera_smm.so    # Location of Galera Plugin
wsrep_provider_options        = "gcache.size=300M; gcache.page_size=1G"                               # Depends on you workload, WS

wsrep_cluster_name            = "esb_galera_cluster"            # Same Cluster name for all nodes
wsrep_cluster_address         = "gcomm://"                          # Initial Cluster start
# wsrep_cluster_address         = "gcomm://192.168.0.2,192.168.0.3"   # Start other nodes like this

wsrep_node_name               = "nodeapidb11"                           # Unique node name
wsrep_node_address            = 10.130.132.92                         # Our address where replication is done
# wsrep_node_incoming_address   = 10.0.0.1                            # Our external interface where application comes from
# wsrep_causal_reads            = 1                                   # If you need realy full-synchronous replication (Galera 3.5
# wsrep_sync_wait               = 1                                   # If you need realy full-synchronous replication (Galera 3.6
# wsrep_slave_threads           = 1                                   # 4 - 8 per core, not more than wsrep_cert_deps_distance

wsrep_sst_method              = mysqldump                           # SST method (initial full sync): mysqldump, rsync, rsync_wan,
wsrep_sst_auth                = esbuser:<password>                          # Username/password for sst user
# wsrep_sst_receive_address     = 192.168.0.1                         # Our address where to receive SST
```

Passwords are stored in Keepass

Create the ***mysql*** user and group:

```
groupadd mysql
useradd -g mysql mysql
```

Copy the MariaDB Galera tar file you imported from production to ***/usr/local:***

```
cp /opt/software/sources/mariadb-galera-5.5.46-linux-x86_64.tar.gz /usr/local/
```

Decompress the MariaDB Galera file and create a symbolic link to ***mysql*** folder:

```
cd /usr/local
```

```
tar -zxvpf /opt/software/sources/mariadb-galera-5.5.46-linux-x86_64.tar.gz
ln -s mariadb-galera-5.5.46-linux-x86_64 mysql
```

Change ownership of folders and create new ones for MariaDB datafiles and socket:

```
chown -R mysql:mysql mysql
cd mysql
chown -R mysql:mysql .
mkdir /var/run/mysqld
chown -R mysql:mysql /var/run/mysqld
mkdir /var/lib/mysql/
chown -R mysql:mysql /var/lib/mysql
```

Run the script to install the database:

```
./scripts/mysql_install_db --user=mysql --ldata=/var/lib/mysql/
```

Start the service:

```
./bin/mysqld --user=mysql --datadir=/var/lib/mysql/ --socket=/var/lib/mysql/mysql.sock &
```

Connect with ***mysql*** user to the database:

```
sudo su - mysql
cd /usr/local/mysql
./bin/mysqladmin -u root password <password> --socket=/var/lib/mysql/mysql.sock
./bin/mysql -u root -p --socket=/var/lib/mysql/mysql.sock
```

Once connected, create the users that the application will use:

```
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'localhost' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'UN-API-T-GW-001.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'UN-API-T-GW-002.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'UN-API-T-CL-001.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'UN-API-T-CL-002.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'UN-API-T-CL-003.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'UN-API-T-DB-001.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'UN-API-T-DB-002.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'UN-API-T-DB-003.edc.un.org' IDENTIFIED BY '<password>';

GRANT ALL PRIVILEGES ON *.* TO 'inthubadmin'@'UN-API-T-CL-001.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'inthubadmin'@'UN-API-T-CL-002.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'inthubadmin'@'UN-API-T-CL-003.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'inthubadmin'@'10.130.95.%' IDENTIFIED BY '<password>' WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'inthubadmin'@'10.130.232.%' IDENTIFIED BY '<password>' WITH GRANT OPTION;

GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'localhost' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'un-api-t-gw-001.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'10.130.131.155' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'un-api-t-gw-002.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'10.130.131.156' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'un-api-t-cl-001.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'10.130.131.184' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'un-api-t-cl-002.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'10.130.131.185' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'un-api-t-cl-003.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'10.130.131.186' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'un-api-t-db-001.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'10.130.132.92' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'un-api-t-db-002.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'10.130.132.93' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'un-api-t-db-003.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'regadmin'@'10.130.132.94' IDENTIFIED BY '<password>';
```

Passwords are stored in Keepass

Stop MariaDB service

```
/usr/local/mysql/bin/mysqladmin -u root -p --socket=/var/lib/mysql/mysql.sock shutdown
/usr/local/mysql/bin/mysql -u root -p --socket=/var/lib/mysql/mysql.sock
```

Exit from ***mysql*** user session to ***root***:

```
exit
```

Backup the current ***my.cnf*** file and copy the one from production you have modified in the ***/etc*** folder (overwrite the existing file):

```
cp /etc/my.cnf /home/global.un.org/crodrigo/my.cnf.bak
cp /opt/software/sources/my.cnf /etc/
```

Start MariaDB cluster with the ***--wsrep-new-cluster*** option

```
cd /usr/local/mysql/
./bin/mysqld --wsrep-new-cluster --user=mysql --datadir=/var/lib/mysql/ &
```

NOTE: This step only has to be applied in the first node

2.2 Configure secondary cluster servers

2.2.1 Server preparation

Connect to ***UN-API-T-DB-002.edc.un.org***

Create the folder for installation:

```
sudo su -
mkdir -p /opt/software/sources
```

Repeat this step in ***UN-API-T-DB-003.edc.un.org***

2.2.2 Install and configure MariaDB Galera in UN-API-T-DB-002

Import with WinSCP from production (***UN-API-P-DB-002.edcv.un.org***) to your home folder in ***UN-API-T-DB-002*** the following files and copy them to ***/opt/software/sources/*** folder:

```
/opt/software/mariadb-galera-5.5.46-linux-x86_64.tar.gz
/etc/my.cnf
```

```
cp /home/global.un.org/crodrigo/mariadb-galera-5.5.46-linux-x86_64.tar.gz /opt/software/sources/
cp /home/global.un.org/crodrigo/my.cnf /opt/software/sources/
```

Edit ***my.cnf*** and change hostnames and other variables in the WSREP section

```
vi /opt/software/sources/my.cnf
```

```
# WSREP parameter

# wsrep_provider                = none                                # Start mysqld without Galera
wsrep_provider                = /usr/local/mariadb-galera-5.5.46-linux-x86_64/lib/libgalera_smm.so    # Location of Galera Plugin
wsrep_provider_options        = "gcache.size=300M; gcache.page_size=1G"                      # Depends on you workload, WS kept fo

wsrep_cluster_name            = "esb_galera_cluster"            # Same Cluster name for all nodes
# wsrep_cluster_address         = "gcomm://"                          # Initial Cluster start
wsrep_cluster_address         = "gcomm://10.130.132.92"   # Start other nodes like this

wsrep_node_name               = "nodeapidb12"                           # Unique node name
wsrep_node_address            = 10.130.132.93                       # Our address where replication is done
# wsrep_node_incoming_address   = 10.0.0.1                            # Our external interface where application comes from
# wsrep_causal_reads            = 1                                   # If you need realy full-synchronous replication (Galera 3.5
# wsrep_sync_wait               = 1                                   # If you need realy full-synchronous replication (Galera 3.6
# wsrep_slave_threads           = 1                                   # 4 - 8 per core, not more than wsrep_cert_deps_distance

wsrep_sst_method              = mysqldump                          # SST method (initial full sync): mysqldump, rsync, rsync_wan,
wsrep_sst_auth                = esbuser:<password>                        # Username/password for sst user
# wsrep_sst_receive_address     = 192.168.0.1                         # Our address where to receive SST
```

Passwords are stored in Keepass

Create the ***mysql*** user and group:

```
groupadd mysql
useradd -g mysql mysql
```

Copy the MariaDB Galera tar file you imported from production to ***/usr/local:***

```
cp /opt/software/sources/mariadb-galera-5.5.46-linux-x86_64.tar.gz /usr/local/
```

Decompress the MariaDB Galera file and create a symbolic link to ***mysql*** folder:

```
cd /usr/local
```

```
tar -zxvpf mariadb-galera-5.5.46-linux-x86_64.tar.gz
ln -s mariadb-galera-5.5.46-linux-x86_64 mysql
```

Change ownership of folders and create new ones for MariaDB datafiles and socket:

```
chown -R mysql:mysql mysql
cd mysql
chown -R mysql:mysql .
mkdir /var/run/mysqld
chown -R mysql:mysql /var/run/mysqld
mkdir /var/lib/mysql/
chown -R mysql:mysql /var/lib/mysql
```

Run the script to install the database:

```
./scripts/mysql_install_db --user=mysql --ldata=/var/lib/mysql/
```

Start the service:

```
/usr/local/mysql/bin/mysqld --user=mysql --datadir=/var/lib/mysql/ --socket=/var/lib/mysql/mysql.sock &
```

Connect with ***mysql*** user to the database:

```
sudo su - mysql
cd /usr/local/mysql
/usr/local/mysql/bin/mysqladmin -u root password Gu3SPBOG81Lio --socket=/var/lib/mysql/mysql.sock
/usr/local/mysql/bin/mysql -u root -pGu3SPBOG81Lio --socket=/var/lib/mysql/mysql.sock
```

Gu3SPBOG81Lio

Once connected, create the users that the application will use:

```
GRANT ALL PRIVILEGES ON *.* TO 'inthubadmin'@'10.130.95.%' IDENTIFIED BY '<password>' WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'inthubadmin'@'10.130.232.%' IDENTIFIED BY '<password>' WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'UN-API-T-DB-001.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'UN-API-T-DB-002.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'UN-API-T-DB-003.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'10.130.132.92' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'10.130.132.93' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'10.130.132.94' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'localhost' IDENTIFIED BY '<password>';
```

Passwords are stored in Keepass

Stop MariaDB service

```
./bin/mysql -u root -p --socket=/var/lib/mysql/mysql.sock shutdown
```

Exit from ***mysql*** user session to ***root***:

```
exit
```

Backup the current ***my.cnf*** file and copy the one from production you have modified in the ***/etc*** folder (overwrite the existing file):

```
cp /etc/my.cnf /home/global.un.org/crodrigo/my.cnf.bak
cp /opt/software/sources/my.cnf /etc/
```

Start MariaDB adding the node to the current cluster:

```
cd /usr/local/mysql/
./bin/mysqld --wsrep_cluster_address=gcomm://10.130.132.92 --user=mysql --datadir=/var/lib/mysql/ &
```

2.2.3 Install and configure MariaDB Galera in UN-API-T-DB-003

Import with WinSCP from production (***UN-API-P-DB-003.edcv.un.org***) to your home folder in ***UN-API-T-DB-003*** the following files and copy them to ***/opt/software/sources/*** folder:

```
/opt/software/mariadb-galera-5.5.46-linux-x86_64.tar.gz
/etc/my.cnf
```

```
cp /home/global.un.org/crodrigo/mariadb-galera-5.5.46-linux-x86_64.tar.gz /opt/software/sources/
cp /home/global.un.org/crodrigo/my.cnf /opt/software/sources/
```

Edit ***my.cnf*** and change hostnames and other variables in the WSREP section

```
vi /opt/software/sources/my.cnf
```

```
# WSREP parameter

# wsrep_provider                = none                                # Start mysqld without Galera
wsrep_provider                = /usr/local/mariadb-galera-5.5.46-linux-x86_64/lib/libgalera_smm.so    # Location of Galera Plugin
wsrep_provider_options        = "gcache.size=300M; gcache.page_size=1G"                        # Depends on you workload, WS kept

wsrep_cluster_name            = "esb_galera_cluster"            # Same Cluster name for all nodes
# wsrep_cluster_address         = "gcomm://"                          # Initial Cluster start
wsrep_cluster_address         = "gcomm://10.130.132.92"   # Start other nodes like this

wsrep_node_name               = "nodeapidb13"                           # Unique node name
wsrep_node_address            = 10.130.132.94                       # Our address where replication is done
# wsrep_node_incoming_address   = 10.0.0.1                            # Our external interface where application comes from
# wsrep_causal_reads            = 1                                   # If you need realy full-synchronous replication (Galera 3.5
# wsrep_sync_wait               = 1                                   # If you need realy full-synchronous replication (Galera 3.6
# wsrep_slave_threads           = 1                                   # 4 - 8 per core, not more than wsrep_cert_deps_distance

wsrep_sst_method              = mysqldump                          # SST method (initial full sync): mysqldump, rsync, rsync_wan,
wsrep_sst_auth                = esbuser:<password>                        # Username/password for sst user
# wsrep_sst_receive_address     = 192.168.0.1                         # Our address where to receive SST
```

Passwords are stored in Keepass

Create the ***mysql*** user and group:

```
groupadd mysql
useradd -g mysql mysql
```

Copy the MariaDB Galera tar file you imported from production to ***/usr/local:***

```
cp /opt/software/sources/mariadb-galera-5.5.46-linux-x86_64.tar.gz /usr/local/
```

Decompress the MariaDB Galera file and create a symbolic link to ***mysql*** folder:

```
cd /usr/local
```

```
tar -zxvpf mariadb-galera-5.5.46-linux-x86_64.tar.gz
ln -s mariadb-galera-5.5.46-linux-x86_64 mysql
```

Change ownership of folders and create new ones for MariaDB datafiles and socket:

```
chown -R mysql:mysql mysql
cd mysql
chown -R mysql:mysql .
mkdir /var/run/mysqld
chown -R mysql:mysql /var/run/mysqld
mkdir /var/lib/mysql/
chown -R mysql:mysql /var/lib/mysql
```

Run the script to install the database:

```
./scripts/mysql_install_db --user=mysql --ldata=/var/lib/mysql/
```

Start the service:

```
./bin/mysqld --user=mysql --datadir=/var/lib/mysql/ --socket=/var/lib/mysql/mysql.sock &
```

Connect with ***mysql*** user to the database:

```
sudo su - mysql
cd /usr/local/mysql
/usr/local/mysql/bin/mysqladmin -u root password <password> --socket=/var/lib/mysql/mysql.sock
/usr/local/mysql/bin/mysql -u root -p --socket=/var/lib/mysql/mysql.sock
```

Once connected, create the users that the application will use:

```
GRANT ALL PRIVILEGES ON *.* TO 'inthubadmin'@'10.130.95.%' IDENTIFIED BY '5XbfbTSRsvXfR' WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'inthubadmin'@'10.130.232.%' IDENTIFIED BY '5XbfbTSRsvXfR' WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'UN-API-T-DB-001.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'UN-API-T-DB-002.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'UN-API-T-DB-003.edc.un.org' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'10.130.132.92' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'10.130.132.93' IDENTIFIED BY '<password>';
GRANT ALL PRIVILEGES ON *.* TO 'esbuser'@'10.130.132.94' IDENTIFIED BY '<password>';
```

Passwords are stored in Keepass

Stop MariaDB service

```
./bin/mysql -u root -p --socket=/var/lib/mysql/mysql.sock shutdown
```

Exit from ***mysql*** user session to ***root***:

```
exit
```

Backup the current ***my.cnf*** file and copy the one from production you have modified in the ***/etc*** folder (overwrite the existing file):

```
cp /etc/my.cnf /home/global.un.org/crodrigo/my.cnf.bak
cp /opt/software/sources/my.cnf /etc/
```

Start MariaDB adding the node to the current cluster:

```
cd /usr/local/mysql/
./bin/mysqld --wsrep_cluster_address=gcomm://10.130.132.92 --user=mysql --datadir=/var/lib/mysql/ &
```

2.3 Import and restore database backups in PRE

2.3.1 Restore MariaDB cluster databases

Import with WinSCP from production (***UN-API-P-DB-001.edcv.un.org***) to your home folder in ***UN-API-T-DB-001*** the following backup files:

```
/opt/backup/activemq.gz
/opt/backup/esb.gz
/opt/backup/esbtest.gz
/opt/backup/regdbam.gz
/opt/backup/regdbcore.gz
```

In ***UN-API-T-DB-001*** restore the dump files by running:

```
/usr/local/mysql/bin/mysql -u root -p --socket=/var/lib/mysql/mysql.sock -e "CREATE DATABASE activemq;"
/usr/local/mysql/bin/mysql -u root -pGu3SPBOG81Lio --socket=/var/lib/mysql/mysql.sock -e "CREATE DATABASE esb;"
/usr/local/mysql/bin/mysql -u root -pGu3SPBOG81Lio --socket=/var/lib/mysql/mysql.sock -e "CREATE DATABASE esbtest;"
/usr/local/mysql/bin/mysql -u root -pGu3SPBOG81Lio --socket=/var/lib/mysql/mysql.sock -e "CREATE DATABASE regdbam;"
/usr/local/mysql/bin/mysql -u root -pGu3SPBOG81Lio --socket=/var/lib/mysql/mysql.sock -e "CREATE DATABASE regdbcore;"

zcat activemq.gz | /usr/local/mysql/bin/mysql -u root -p --socket=/var/lib/mysql/mysql.sock activemq
zcat esb.gz | /usr/local/mysql/bin/mysql -u root -p --socket=/var/lib/mysql/mysql.sock esb
zcat esbtest.gz | /usr/local/mysql/bin/mysql -u root -p --socket=/var/lib/mysql/mysql.sock esbtest
zcat regdbam.gz | /usr/local/mysql/bin/mysql -u root -p --socket=/var/lib/mysql/mysql.sock regdbam
zcat regdbcore.gz | /usr/local/mysql/bin/mysql -u root -p --socket=/var/lib/mysql/mysql.sock regdbcore
```

Do this step only in DB-001 server, the Galera cluster will propagate the changes to the rest of the nodes.

2.3.2 Restore API Monitor MariaDB databases

Import with WinSCP from production (***UN-API-P-MONDS-001.edcv.un.org***) to your home folder in ***UN-API-T-MONDS-001*** the following backup files:

```
/opt/backup/regdbam.gz
/opt/backup/regdbcarbon.gz
/opt/backup/regdbstats.gz
```

In ***UN-API-T-MONDS-001*** restore the dump files by running:

```
zcat regdbam.gz | mysql -u root -p regdbam
zcat regdbcarbon.gz | mysql -u root -p regdbcarbon
zcat regdbstats.gz | mysql -u root -p regdbstats
```

3. WSO2 Business Activity Monitor

Connect to ***UN-API-T-MOSRV-001.edc.un.org***

Import with WinSCP from production (***UN-API-P-MOSRV-001.edcv.un.org***) to your home folder in ***UN-API-T-MOSRV-001*** the following files:

```
In production server:
cd /opt/software/sources
tar -cvzf prod_sources.tar.gz *
cp prod_sources.tar.gz /home/global.un.org/crodrigo2/

In pre-production server:
cp prod_sources.tar.gz /opt/software/sources/
cd /opt/software/sources
tar -xvzf prod_sources.tar.gz *
```

Create the folder for installation:

```
sudo su -
mkdir /opt/software
```

Create the OICT admin user:

```
useradd oictadmin
```

Copy JDK and Apache Maven TAR files to ***/opt/software***

```
cp /home/global.un.org/crodrigo/jdk-7u71-linux-x64.tar.gz /opt/software/
cp /home/global.un.org/crodrigo/apache-maven-3.2.3-bin.tar.gz /opt/software/
```

Install the pre-requirements (Java and Apache Maven) and move the source tar files to sources folder

```
cd /opt/software
tar -xvf jdk-7u71-linux-x64.tar.gz
```

```
tar -xvf apache-maven-3.2.3-bin.tar.gz
```

```
mv jdk-7u71-linux-x64.tar.gz sources/
mv apache-maven-3.2.3-bin.tar.gz sources/
```

Edit the `.bashrc` file for ***oictadmin*** user:

```
sudo su - oictadmin
vi ~/.bashrc
```

Add the following lines at the end of the file:

```
export M2_HOME=/opt/software/apache-maven-3.2.3
export M2=$M2_HOME/bin
export PATH=$M2:$PATH
export JAVA_HOME=/opt/software/jdk1.7.0_71
export PATH=$JAVA_HOME/bin:/$PATH
```

This lines allow the system to point to the installation folders of JDK and Maven when logged as ***oictadmin*** user.

Save and exit the file.

Unzip WSO2 BAM zip file (as ***root***):

```
exit
cd /opt/software/
unzip wso2bam-2.5.0.zip
```

If ***unzip*** software is not installed in the machine, install it by running:

```
yum install unzip
```

Change the owner of Java and Maven folders

```
chown oictadmin:oictadmin -R /opt/software/jdk1.7.0_71
```

```
chown oictadmin:oictadmin -R /opt/software/apache-maven-3.2.3
```

Give execution permissions to the Java folder

```
chmod +x -R /opt/software/jdk1.7.0_71/bin
```

Go to ***repository/conf*** folder for changing the BAM configuration:

```
cd /opt/software/wso2bam-2.5.0/repository/conf/
```

Make a backup of the current ***user-mgt.xml*** file:

```
cp /opt/software/wso2bam-2.5.0/repository/conf/user-mgt.xml /home/global.un.org/crodrigo/user-mgt.xml.bak
```

Copy the ***user-mgt.xml*** file previously uploaded from production to WSO2 BAM config folder:

```
cp /home/global.un.org/crodrigo/user-mgt.xml /opt/software/wso2bam-2.5.0/repository/conf/user-mgt.xml
```

At this point you can edit the file if needed and change username and password. **In this case we leave it as it is.**

Make a backup of the current ***master-datasources.xml*** file:

```
cp /opt/software/wso2bam-2.5.0/repository/conf/datasources/master-datasources.xml /opt/software/wso2bam-
2.5.0/repository/conf/datasources/master-datasources.xml.bak
```

Copy ***master-datasources.xml*** file previously uploaded from production to WSO2 BAM repository folder:

```
cp /home/global.un.org/crodrigo/master-datasources.xml /opt/software/wso2bam-2.5.0/repository/conf/datasources/master-
datasources.xml
```

Edit the file and change the lines under ***<configuration>*** tag:

```
vi /opt/software/wso2bam-2.5.0/repository/conf/datasources/master-datasources.xml
```

```
<datasource>
    <name>WSO2_CARBON_DB</name>
    <description>The datasource used for registry and user manager</description>
    <jndiConfig>
        <name>jdbc/WSO2CarbonDB</name>
    </jndiConfig>
    <definition type="RDBMS">
        <configuration>
            <url>jdbc:mysql://un-api-t-monds-001.edc.un.org:3306/regdbcarbon?autoReconnect=true&amp;relaxAutoCommit=true</url>
            <username>regadmin</username>
            <password>password</password>
            <driverClassName>com.mysql.jdbc.Driver</driverClassName>
            <maxActive>80</maxActive>
            <maxWait>60000</maxWait>
            <minIdle>5</minIdle>
            <testOnBorrow>true</testOnBorrow>
            <validationQuery>SELECT 1</validationQuery>
            <validationInterval>30000</validationInterval>
        </configuration>
    </definition>
</datasource>
```

Passwords are stored in Keepass

These lines define the connection from BAM application to ***regdbcarbon*** database in ***UN-API-T-MONDS-001***.

Copy the java connector to the component library

`cp /home/global.un.org/spati/mysql-connector-java-5.1.35-bin.jar /opt/software/wso2bam-2.5.0/repository/components/lib/`

Connect now to MariaDB in ***UN-API-T-MONDS-001*** (you can use HeidiSQL or connect directly on the local server) and create the database ***regdbcarbon***:

`CREATE DATABASE IF NOT EXISTS regdbcarbon;`

Come back to ***UN-API-T-MOSRV-001*** and install the MariaDB client

`yum install mariadb`

In ***UN-API-T-MONDS-001*** connect to the database and grant permissions to ***inthubadmin*** user to access the server:

`GRANT ALL PRIVILEGES ON *.* TO 'inthubadmin'@'10.130.131.191' IDENTIFIED BY '5XbfbTSRsvXfR ' WITH GRANT OPTION;`

`FLUSH PRIVILEGES;`

populate the database using the predefined script `mysql.sql`

`mysql -u root -p -h UN-API-T-MONDS-001.edc.un.org regdbcarbon < /opt/software/wso2bam-2.5.0/dbscripts/mysql.sql`

Change to oictadmin user and execute the ***wsoserver.sh*** script

`sudo su - oictadmin`

`cd /opt/software/wso2bam-2.5.0/bin`

`sh wso2server.sh &`

You can test the installation by connecting to:

https://10.130.131.191:9443/carbon

4. JBoss Fuse Cluster

4.1 Configure first cluster node (UN-API-T-CL-001)

Import with WinSCP from production (***UN-API-P-CL-001.edcv.un.org***) to your home folder in ***UN-API-T-CL-001*** the following files:

```
/opt/software/jdk-7u71-linux-x64.tar.gz
/opt/software/apache-maven-3.2.3-bin.tar.gz
/opt/software/jboss-fuse-full-6.2.0.redhat-123.zip
/home/oictadmin/.m2/repository.tar.gz
/home/oictadmin/.m2/settings-prod.xml
```

Create the folder for installation:

```
sudo su -
mkdir /opt/software
```

Create the ***oictadmin*** user

```
useradd oictadmin
```

Copy files from your home folder to ***/opt/software/***:

```
cp /home/global.un.org/crodrigo/jdk-7u71-linux-x64.tar.gz /opt/software/
cp /home/global.un.org/crodrigo/apache-maven-3.2.3-bin.tar.gz /opt/software/
cp /home/global.un.org/crodrigo/jboss-fuse-full-6.2.0.redhat-123.zip /opt/software/
cp /home/global.un.org/crodrigo/settings-prod.xml /opt/software/
cp /home/global.un.org/crodrigo/repository.tar.gz /opt/software/
```

Copy the pre-requirement software (Java and Apache Maven) to ***/opt/software***:

```
cd /opt/software
tar -xvf jdk-7u71-linux-x64.tar.gz
tar -xvf apache-maven-3.2.3-bin.tar.gz
```

Edit the `.bashrc` file for ***oictadmin*** user

```
sudo su - oictadmin
vi ~/.bashrc
```

Add the following lines at the end of the file:

```
export M2_HOME=/opt/software/apache-maven-3.2.3
export M2=$M2_HOME/bin
export PATH=$M2:$PATH
export JAVA_HOME=/opt/software/jdk1.7.0_71
export PATH=$JAVA_HOME/bin:/$PATH
```

Uncompress the Jboss zip file:

```
cd /opt/software
unzip jboss-fuse-full-6.2.0.redhat-123.zip
```

If ***unzip*** software is not installed in the machine, install it by running:

```
yum install unzip
```

Copy the settings file for Maven:

```
sudo su - oictadmin
mkdir /home/oictadmin/.m2
cd /home/oictadmin/.m2
cp /opt/software/settings-prod.xml settings.xml
```

Edit the ***settings.xml*** file and change the admin password:

```
vi settings.xml
```

```
<servers>
    <server>
        <id>fabric8.console</id>
        <username>admin</username>
        <password>password</password>
    </server>
    <server>
        <id>fabric8.upload.repo</id>
        <username>admin</username>
        <password>password</password>
    </server>
</servers>
```

Passwords are stored in Keepass

Copy the repository tar file to ***.m2*** folder:

```
cp /opt/software/repository.tar.gz .
```

Decompress the file:

```
tar -xvf repository.tar.gz
```

Exit to ***root*** user and edit the ***user.properties*** file:

```
exit
cd /opt/software/jboss-fuse-6.2.0.redhat-123
vi etc/users.properties
```

Uncomment the last line:

```
#admin=admin,admin,manager,viewer,Monitor,Operator, Maintainer, Deployer, Auditor, Administrator, SuperUser
```

Edit the ***system.properties*** file and change the main container name:

```
vi etc/system.properties
```

```
karaf.name=root0012
```

We will be using the number based on the machine name, as per list below:

UN-API-T-GW-001.edc.un.org - BRVM0010
UN-API-T-GW-002.edc.un.org - BRVM0011
UN-API-T-CL-001.edc.un.org - BRVM0012
UN-API-T-CL-002.edc.un.org - BRVM0013
UN-API-T-CL-003.edc.un.org - BRVM0014
UN-API-T-CASCL-001.edc.un.org - BRVM0015
UN-API-T-CASCL-002.edc.un.org - BRVM0016
UN-API-T-MOQUE-001.edc.un.org - BRVM0017
UN-API-T-MOQUE-002.edc.un.org - BRVM0018
UN-API-T-MOSRV-001.edc.un.org - BRVM0020
UN-API-T-DB-001.edc.un.org - BRVM0021
UN-API-T-DB-002.edc.un.org - BRVM0022
UN-API-T-DB-003.edc.un.org - BRVM0023
UN-API-T-MONDS-001.edc.un.org - BRVM0019

Start Fuse standalone:

```
chown oictadmin:oictadmin -R /opt/software/jboss-fuse-6.2.0.redhat-123
sudo su - oictadmin
cd /opt/software/jboss-fuse-6.2.0.redhat-123/bin
./fuse
```

Create a new Fuse Fabric

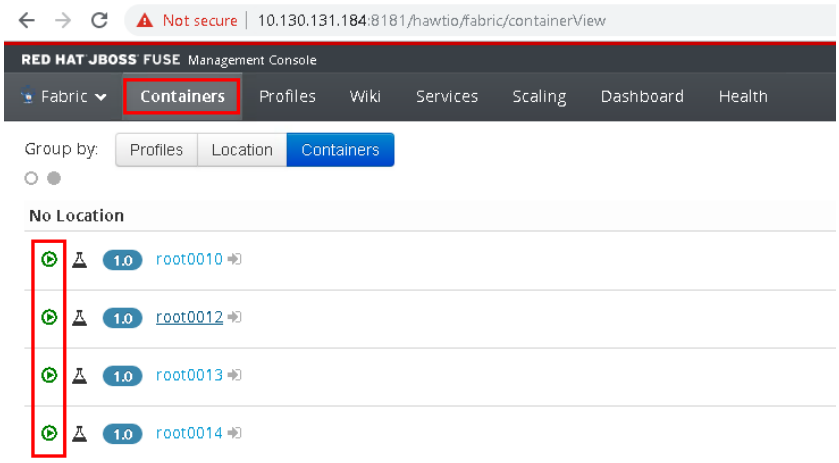
```
JBossFuse:admin@root0012> fabric:create --clean --new-user admin --new-user-password admin --zookeeper-password <password> --
zookeeper-server-port 3181 --min-port 3000 --max-port 3999 --wait-for-provisioning
```

To verify and check the status of the fabric execute the command:

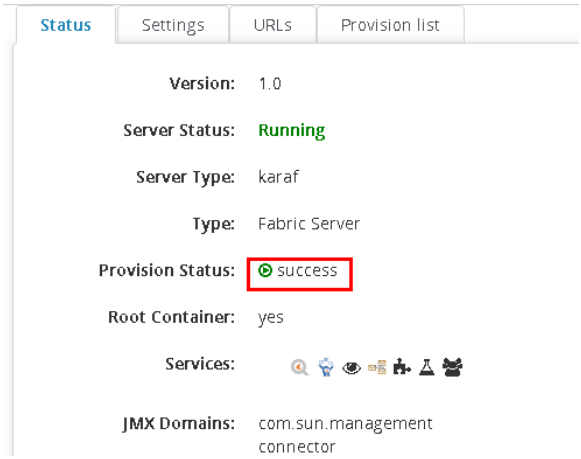
```
JBossFuse:admin@root0012> fabric:container-list
```

```
[id]       [version]  [type]  [connected]  [profiles]              [provision status]
root0010   1.0        karaf   yes          fabric                  success
root0012*  1.0        karaf   yes          fabric                  success
                                           fabric-ensemble-0000-1
                                           jboss-fuse-full
root0013   1.0        karaf   yes          fabric                  success
root0014   1.0        karaf   yes          fabric                  success
```

You can also connect to http://10.130.131.184:8181/hawtio/login and login as admin, there you can see the status of the containers:



You can click on any of the containers and check that the status is ***success***



Quit Karaf console and start Fuse server without console

```
karaf> exit
./start
```

4.2 Configure second cluster node (UN-API-T-CL-002)

Import with WinSCP from production (***UN-API-P-CL-002.edcv.un.org***) to your home folder in ***UN-API-T-CL-002*** the following files:

```
/opt/software/jdk-7u71-linux-x64.tar.gz
/opt/software/apache-maven-3.2.3-bin.tar.gz
/opt/software/jboss-fuse-full-6.2.0.redhat-123.zip
/home/oictadmin/.m2/repository.tar.gz
/home/oictadmin/.m2/settings-prod.xml
```

Create the folder for installation:

```
sudo su -
mkdir /opt/software
```

Create the ***oictadmin*** user

```
useradd oictadmin
```

Copy files from your home folder to ***/opt/software/***:

```
cp /home/global.un.org/crodrigo/jdk-7u71-linux-x64.tar.gz /opt/software/
cp /home/global.un.org/crodrigo/apache-maven-3.2.3-bin.tar.gz /opt/software/
cp /home/global.un.org/crodrigo/jboss-fuse-full-6.2.0.redhat-123.zip /opt/software/
cp /home/global.un.org/crodrigo/settings-prod.xml /opt/software/
cp /home/global.un.org/crodrigo/repository.tar.gz /opt/software/
```

Install the pre-requirements (Java and Apache Maven):

```
cd /opt/software
tar -xvf jdk-7u71-linux-x64.tar.gz
tar -xvf apache-maven-3.2.3-bin.tar.gz
```

Edit the `.bashrc` file for ***oictadmin*** user

```
sudo su - oictadmin
vi ~/.bashrc
```

Add the following lines at the end of the file:

```
export M2_HOME=/opt/software/apache-maven-3.2.3
export M2=$M2_HOME/bin
export PATH=$M2:$PATH
export JAVA_HOME=/opt/software/jdk1.7.0_71
export PATH=$JAVA_HOME/bin:/$PATH
```

Uncompress the Jboss zip file:

```
cd /opt/software
unzip jboss-fuse-full-6.2.0.redhat-123.zip
```

If ***unzip*** software is not installed in the machine, install it by running:

```
yum install unzip
```

Copy the settings file for Maven:

```
sudo su - oictadmin
mkdir /home/oictadmin/.m2
cd /home/oictadmin/.m2
cp /opt/software/settings-prod.xml settings.xml
```

Edit the ***settings.xml*** file and change the admin password:

```
vi settings.xml
```

```
<servers>
    <server>
        <id>fabric8.console</id>
        <username>admin</username>
        <password>password</password>
    </server>
    <server>
        <id>fabric8.upload.repo</id>
        <username>admin</username>
        <password>password</password>
    </server>
</servers>
```

Passwords are stored in Keepass

Copy the repository tar file to ***.m2*** folder:

```
cp /opt/software/repository.tar.gz .
```

Decompress the file:

```
tar -xvf repository.tar.gz
```

Exit to ***root*** user and edit the ***user.properties*** file:

```
exit
cd /opt/software/jboss-fuse-6.2.0.redhat-123
vi etc/users.properties
```

Uncomment the last line:

```
#admin=admin,admin,manager,viewer,Monitor,Operator, Maintainer, Deployer, Auditor, Administrator, SuperUser
```

Edit the ***system.properties*** file and change the main container name:

```
vi etc/system.properties
```

```
karaf.name=root0013
```

Start Fuse standalone:

```
chown oictadmin:oictadmin -R /opt/software/jboss-fuse-6.2.0.redhat-123
sudo su - oictadmin
cd /opt/software/jboss-fuse-6.2.0.redhat-123/bin
./fuse
```

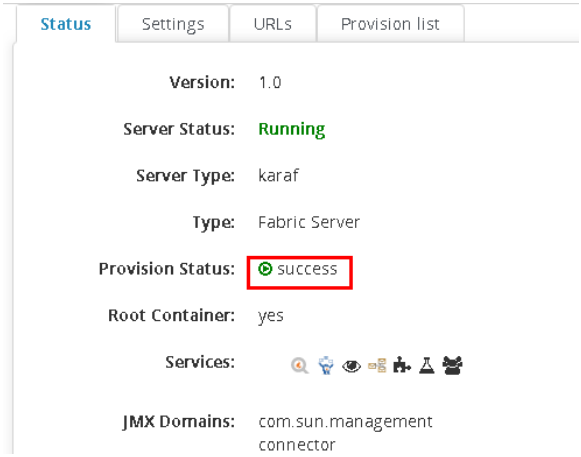
Add to existing Fuse Fabric

```
karaf> fabric:join --zookeeper-password <password> 10.130.131.184:3181 root0013
```

Quit Karaf console and start Fuse server without console

```
karaf> exit
./start
```

To check if the container was correctly added to the fabric and started connect to http://10.130.131.184:8181/hawtio/login, click on the name of the root container and check that the status is ***success***



4.3 Configure third cluster node (UN-API-T-CL-003)

Import with WinSCP from production (***UN-API-P-CL-003.edcv.un.org***) to your home folder in ***UN-API-T-CL-003*** the following files:

```
/opt/software/jdk-7u71-linux-x64.tar.gz
/opt/software/apache-maven-3.2.3-bin.tar.gz
/opt/software/jboss-fuse-full-6.2.0.redhat-123.zip
/home/oictadmin/.m2/repository.tar.gz
/home/oictadmin/.m2/settings-prod.xml
```

Create the folder for installation:

```
sudo su -
mkdir /opt/software
```

Create the ***oictadmin*** user

```
useradd oictadmin
```

Copy files from your home folder to ***/opt/software/***:

```
cp /home/global.un.org/crodrigo/jdk-7u71-linux-x64.tar.gz /opt/software/
cp /home/global.un.org/crodrigo/apache-maven-3.2.3-bin.tar.gz /opt/software/
cp /home/global.un.org/crodrigo/jboss-fuse-full-6.2.0.redhat-123.zip /opt/software/
cp /home/global.un.org/crodrigo/settings-prod.xml /opt/software/
cp /home/global.un.org/crodrigo/repository.tar.gz /opt/software/
```

Install the pre-requirements (Java and Apache Maven):

```
cd /opt/software
tar -xvf jdk-7u71-linux-x64.tar.gz
tar -xvf apache-maven-3.2.3-bin.tar.gz
```

Edit the `.bashrc` file for ***oictadmin*** user

```
sudo su - oictadmin
vi ~/.bashrc
```

Add the following lines at the end of the file:

```
export M2_HOME=/opt/software/apache-maven-3.2.3
export M2=$M2_HOME/bin
export PATH=$M2:$PATH
export JAVA_HOME=/opt/software/jdk1.7.0_71
export PATH=$JAVA_HOME/bin:/$PATH
```

Uncompress the Jboss zip file:

```
cd /opt/software
unzip jboss-fuse-full-6.2.0.redhat-123.zip
```

If ***unzip*** software is not installed in the machine, install it by running:

```
yum install unzip
```

Copy the settings file for Maven:

```
sudo su - oictadmin
mkdir /home/oictadmin/.m2
cd /home/oictadmin/.m2
cp /opt/software/settings-prod.xml settings.xml
```

Edit the ***settings.xml*** file and change the admin password:

```
vi settings.xml
```

```
<servers>
    <server>
        <id>fabric8.console</id>
        <username>admin</username>
        <password>password</password>
    </server>
    <server>
        <id>fabric8.upload.repo</id>
        <username>admin</username>
        <password>password</password>
    </server>
</servers>
```

Passwords are stored in Keepass

Copy the repository tar file to ***.m2*** folder:

```
cp /opt/software/repository.tar.gz .
```

Decompress the file:

```
tar -xvf repository.tar.gz
```

Exit to ***root*** user and edit the ***user.properties*** file:

```
exit
cd /opt/software/jboss-fuse-6.2.0.redhat-123
vi etc/users.properties
```

Uncomment the last line:

```
#admin=admin,admin,manager,viewer,Monitor,Operator, Maintainer, Deployer, Auditor, Administrator, SuperUser
```

Edit the ***system.properties*** file and change the main container name:

```
vi etc/system.properties
```

```
karaf.name=root0014
```

Start Fuse standalone:

```
chown oictadmin:oictadmin -R /opt/software/jboss-fuse-6.2.0.redhat-123
sudo su - oictadmin
cd /opt/software/jboss-fuse-6.2.0.redhat-123/bin
./fuse
```

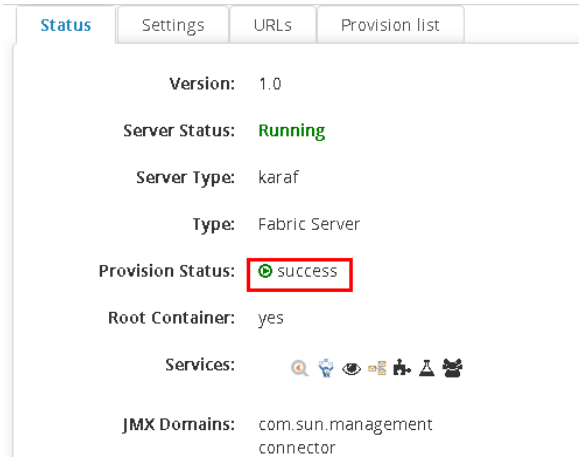
Add to existing Fuse Fabric

```
karaf> fabric:join --zookeeper-password <password> 10.130.131.184:3181 root0014
```

Quit Karaf console and start Fuse server without console

```
karaf> exit
./start
```

To check if the container was correctly added to the fabric and started connect to http://10.130.131.184:8181/hawtio/login, click on the name of the root container and check that the status is ***success***



5. API Manager

5.1 Configure JBoss Fuse in first frontend node (UN-API-T-GW-001)

Import with WinSCP from production (***UN-API-P-GW-001.edcv.un.org***) to your home folder in ***UN-API-T-GW-001*** the following files:

```
/opt/software/jdk-7u71-linux-x64.tar.gz
/opt/software/apache-maven-3.2.3-bin.tar.gz
/opt/software/jboss-fuse-full-6.2.0.redhat-123.zip
/opt/software/wso2am-1.8.0.zip
/opt/software/wso2am-1.8.0/repository/conf/api-manager.xml
/opt/software/wso2am-1.8.0/repository/conf/datasources/master-datasources.xml
/opt/software/wso2am-1.8.0/repository/deployment/server/synapse-configs/default/registry.xml
/opt/software/wso2am-1.8.0/repository/conf/user-mgt.xml
/home/oictadmin/.m2/repository.tar.gz
/home/oictadmin/.m2/settings-prod.xml
```

Create the ***oictadmin*** user

```
useradd oictadmin
```

Copy installation files from your home folder to ***/opt/software/***:

```
cp /home/global.un.org/crodrigo/jdk-7u71-linux-x64.tar.gz /opt/software/
cp /home/global.un.org/crodrigo/apache-maven-3.2.3-bin.tar.gz /opt/software/
cp /home/global.un.org/crodrigo/jboss-fuse-full-6.2.0.redhat-123.zip /opt/software/
cp /home/global.un.org/crodrigo/wso2am-1.8.0.zip /opt/software/
```

Install the pre-requirements (Java and Apache Maven):

```
cd /opt/software
tar -xvf jdk-7u71-linux-x64.tar.gz
tar -xvf apache-maven-3.2.3-bin.tar.gz
```

Edit the ***.bashrc*** file for ***oictadmin*** user

```
sudo su - oictadmin
vi ~/.bashrc
```

Add the following lines at the end of the file:

```
export M2_HOME=/opt/software/apache-maven-3.2.3
export M2=$M2_HOME/bin
export PATH=$M2:$PATH
export JAVA_HOME=/opt/software/jdk1.7.0_71
export PATH=$JAVA_HOME/bin:/$PATH
```

Uncompress the Jboss Fuse zip file:

```
cd /opt/software
unzip jboss-fuse-full-6.2.0.redhat-123.zip
```

If ***unzip*** software is not installed in the machine, install it by running:

```
yum install unzip
```

Copy the settings file for Maven imported from production:

```
sudo su - oictadmin
mkdir /home/oictadmin/.m2
cd /home/oictadmin/.m2
cp /opt/software/settings-prod.xml settings.xml
```

Edit the settings.xml file and change the admin password:

```
vi settings.xml
```

Copy the ***repository.tar.gz*** file to ***.m2*** folder

```
cp /opt/software/repository.tar.gz .
```

Decompress the file

```
tar -xvf repository.tar.gz
```

Return to ***root*** user session and edit the ***user.properties*** file:

```
exit
cd /opt/software/jboss-fuse-6.2.0.redhat-123
vi etc/users.properties
```

Uncomment the last line and save:

```
#admin=admin,admin,manager,viewer,Monitor,Operator, Maintainer, Deployer, Auditor, Administrator, SuperUser
```

Edit the ***system.properties*** file and change the main container name:

```
vi etc/system.properties
```

```
karaf.name=root0010
```

Start Fuse standalone

```
chown oictadmin:oictadmin -R /opt/software/jboss-fuse-6.2.0.redhat-123
sudo su - oictadmin
cd /opt/software/jboss-fuse-6.2.0.redhat-123/bin
./fuse
```

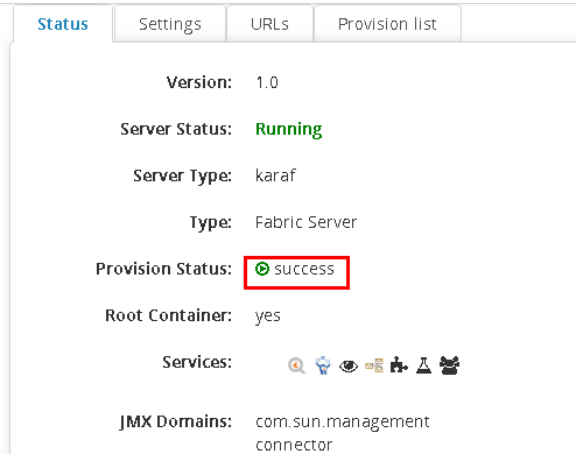
Add to existing Fuse Fabric

```
karaf> fabric:join --zookeeper-password <password> 10.130.131.184:3181 root0010
```

Quit Karaf console and start Fuse server without console

```
karaf> exit
./start
```

To check if the container was correctly added to the fabric and started connect to http://10.130.131.184:8181/hawtio/login, click on the name of the root container and check that the status is ***success***



5.2 Configure WSO2 API Gateway in first frontend node (UN-API-T-GW-001)

Unzip WSO2 zip file (as ***root*** user):

```
cd /opt/software/
unzip wso2am-1.8.0.zip
```

Make a backup of the current ***api-manager.xml*** file

```
cp /opt/software/wso2am-1.8.0/repository/conf/api-manager.xml /home/global.un.org/crodrigo/api-manager.xml.bak
```

Copy ***api-manager.xml*** file to ***wso2am*** config folder:

```
cp /home/global.un.org/crodrigo/api-manager.xml /opt/software/wso2am-1.8.0/repository/conf/api-manager.xml
```

Edit ***api-manager.xml*** and change hostname, port number, user name and password for BAM server as appropriate

```
vi /opt/software/wso2am-1.8.0/repository/conf/api-manager.xml
```

```
[...]
<BAMServerURL>tcp://un-api-t-mosrv-001.edc.un.org:7614/</BAMServerURL>
[...]
<Access-Control-Allow-Origin>http://un-api-t-gw-001.edc.un.org:9763/store,
https://un-api-t-gw-001.edc.un.org:9443/store</Access-Control-Allow-Origin>
```

Make a backup of the current ***master-datasources.xml*** file

```
cp /opt/software/wso2am-1.8.0/repository/conf/datasources/master-datasources.xml /home/global.un.org/crodrigo/master-datasources.xml.bak
```

Copy ***master-datasources.xml*** file to ***wso2am*** repository folder and edit it:

```
cp /home/global.un.org/crodrigo/master-datasources.xml /opt/software/wso2am-1.8.0/repository/conf/datasources/master-datasources.xml
vi /opt/software/wso2am-1.8.0/repository/conf/datasources/master-datasources.xml
```

Edit IPs and user credentials for all ***<datasource>*** sections

```
<datasource>
    <name>WSO2_CARBON_DB</name>
    <description>The datasource used for registry and user manager</description>
    <jndiConfig>
        <name>jdbc/WSO2CarbonDB</name>
    </jndiConfig>
    <definition type="RDBMS">
        <configuration>
            <url>jdbc:mysql:loadbalance://10.130.132.92,10.130.132.93/regdbcore?autoReconnect=true&amp;relaxAutoCommit=true</url>
            <username>regadmin</username>
            <password>password</password>
            <driverClassName>com.mysql.jdbc.Driver</driverClassName>
            <maxActive>80</maxActive>
            <maxWait>60000</maxWait>
            <minIdle>5</minIdle>
```

```
            <testOnBorrow>true</testOnBorrow>
            <validationQuery>SELECT 1</validationQuery>
            <validationInterval>30000</validationInterval>
        </configuration>
    </definition>
</datasource>

<datasource>
    <name>WSO2AM_DB</name>
    <description>The datasource used for API Manager database</description>
    <jndiConfig>
        <name>jdbc/WSO2AM_DB</name>
    </jndiConfig>
    <definition type="RDBMS">
        <configuration>
            <url>jdbc:mysql:loadbalance://10.130.132.92,10.130.132.93/regdbam?autoReconnect=true&amp;relaxAutoCommit=true</url>
            <username>regadmin</username>
            <password>password</password>
            <driverClassName>com.mysql.jdbc.Driver</driverClassName>
            <maxActive>80</maxActive>
            <maxWait>60000</maxWait>
            <minIdle>5</minIdle>
            <testOnBorrow>true</testOnBorrow>
            <validationQuery>SELECT 1</validationQuery>
            <validationInterval>30000</validationInterval>
        </configuration>
    </definition>
</datasource>

<datasource>
    <name>WSO2AM_STATS_DB</name>
    <description>The datasource used for getting statistics to API Manager</description>
    <jndiConfig>
        <name>jdbc/WSO2AM_STATS_DB</name>
    </jndiConfig>
    <definition type="RDBMS">
        <configuration>
            <url>jdbc:mysql://un-api-t-monds-001.edc.un.org:3306/regdbstats?autoReconnect=true&amp;relaxAutoCommit=true</url>
            <username>regadmin</username>
            <password>password</password>
            <driverClassName>com.mysql.jdbc.Driver</driverClassName>
            <maxActive>80</maxActive>
            <maxWait>60000</maxWait>
            <minIdle>5</minIdle>
            <testOnBorrow>true</testOnBorrow>
            <validationQuery>SELECT 1</validationQuery>
            <validationInterval>30000</validationInterval>
        </configuration>
    </definition>
</datasource>
```

Passwords are stored in Keepass

Make a backup of the current ***registry.xml*** file:

```
cp /opt/software/wso2am-1.8.0/repository/deployment/server/synapse-configs/default/registry.xml
/home/global.un.org/crodrigo/registry.xml.bak
```

Copy ***registry.xml*** file to ***wso2am*** repository folder:

```
cp /home/global.un.org/crodrigo/registry.xml /opt/software/wso2am-1.8.0/repository/deployment/server/synapse-
configs/default/registry.xml
```

Make a backup of the current ***user-mgt.xml*** file

```
cp /opt/software/wso2am-1.8.0/repository/conf/user-mgt.xml /home/global.un.org/crodrigo/user-mgt.xml.bak
```

Copy ***user-mgt.xml*** file to ***wso2am*** repository folder

```
cp /home/global.un.org/crodrigo/user-mgt.xml /opt/software/wso2am-1.8.0/repository/conf/user-mgt.xml
```

At this point you can edit the file if needed and change username and password. **In this case we leave it as it is.**

Change owner of API Manager folder:

```
chown -R oictadmin:oictadmin wso2am-1.8.0
```

Execute the ***wsoserver.sh*** script

```
sudo su - oictadmin
cd /opt/software/wso2am-1.8.0/bin
sh wso2server.sh &
```

To test if the application works connect to:

https://10.130.131.155:9443/

5.3 Configure JBoss Fuse in second frontend node (UN-API-T-GW-002)

Import with WinSCP from production (***UN-API-P-GW-002.edcv.un.org***) to your home folder in ***UN-API-T-GW-002*** the following files:

```
/opt/software/jdk-7u71-linux-x64.tar.gz
/opt/software/apache-maven-3.2.3-bin.tar.gz
/opt/software/jboss-fuse-full-6.2.0.redhat-123.zip
/opt/software/wso2am-1.8.0.zip
/opt/software/wso2am-1.8.0/repository/conf/api-manager.xml
/opt/software/wso2am-1.8.0/repository/conf/datasources/master-datasources.xml
/opt/software/wso2am-1.8.0/repository/deployment/server/synapse-configs/default/registry.xml
/opt/software/wso2am-1.8.0/repository/conf/user-mgt.xml
/home/oictadmin/.m2/repository.tar.gz
/home/oictadmin/.m2/settings-prod.xml
```

Create the ***oictadmin*** user

```
useradd oictadmin
```

Copy installation files from your home folder to ***/opt/software/***:

```
cp /home/global.un.org/crodrigo/jdk-7u71-linux-x64.tar.gz /opt/software/
cp /home/global.un.org/crodrigo/apache-maven-3.2.3-bin.tar.gz /opt/software/
cp /home/global.un.org/crodrigo/jboss-fuse-full-6.2.0.redhat-123.zip /opt/software/
cp /home/global.un.org/crodrigo/wso2am-1.8.0.zip /opt/software/
```

Install the pre-requirements (Java and Apache Maven):

```
cd /opt/software
tar -xvf jdk-7u71-linux-x64.tar.gz
tar -xvf apache-maven-3.2.3-bin.tar.gz
```

Edit the ***.bashrc*** file for ***oictadmin*** user

```
sudo su - oictadmin
vi ~/.bashrc
```

Add the following lines at the end of the file:

```
export M2_HOME=/opt/software/apache-maven-3.2.3
export M2=$M2_HOME/bin
export PATH=$M2:$PATH
export JAVA_HOME=/opt/software/jdk1.7.0_71
export PATH=$JAVA_HOME/bin:/$PATH
```

Uncompress the Jboss Fuse zip file:

```
cd /opt/software
unzip jboss-fuse-full-6.2.0.redhat-123.zip
```

If ***unzip*** software is not installed in the machine, install it by running:

```
yum install unzip
```

Copy the settings file for Maven imported from production:

```
sudo su - oictadmin
mkdir /home/oictadmin/.m2
cd /home/oictadmin/.m2
cp /opt/software/settings-prod.xml settings.xml
```

Edit the settings.xml file and change the admin password:

```
vi settings.xml
```

Copy the ***repository.tar.gz*** file to ***.m2*** folder

```
cp /opt/software/repository.tar.gz .
```

Decompress the file

```
tar -xvf repository.tar.gz
```

Return to ***root*** user session and edit the ***user.properties*** file:

```
exit
cd /opt/software/jboss-fuse-6.2.0.redhat-123
vi etc/users.properties
```

Uncomment the last line and save:

```
#admin=admin,admin,manager,viewer,Monitor,Operator, Maintainer, Deployer, Auditor, Administrator, SuperUser
```

Edit the ***system.properties*** file and change the main container name:

```
vi etc/system.properties
```

```
karaf.name=root0011
```

Start Fuse standalone

```
chown oictadmin:oictadmin -R /opt/software/jboss-fuse-6.2.0.redhat-123
sudo su - oictadmin
cd /opt/software/jboss-fuse-6.2.0.redhat-123/bin
./fuse
```

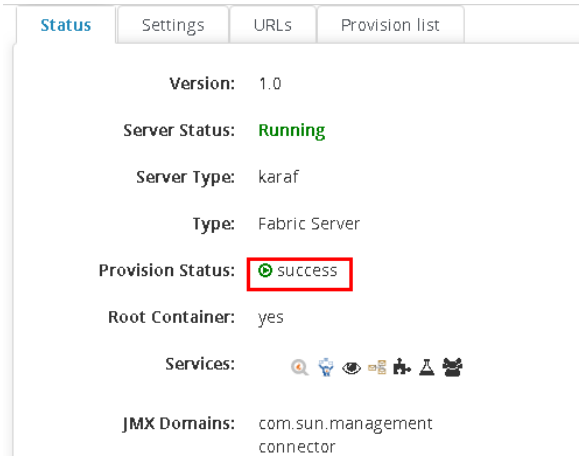
Add to existing Fuse Fabric

```
karaf> fabric:join --zookeeper-password <password> 10.130.131.184:3181 root0011
```

Quit Karaf console and start Fuse server without console

```
karaf> exit
./start
```

To check if the container was correctly added to the fabric and started connect to http://10.130.131.184:8181/hawtio/login, click on the name of the root container and check that the status is ***success***



5.4 Configure WSO2 API Gateway in second frontend node (UN-API-T-GW-002)

Unzip WSO2 zip file (as ***root*** user):

```
cd /opt/software/
unzip wso2am-1.8.0.zip
```

Make a backup of the current ***api-manager.xml*** file

```
cp /opt/software/wso2am-1.8.0/repository/conf/api-manager.xml /home/global.un.org/crodrigo/api-manager.xml.bak
```

Copy ***api-manager.xml*** file to ***wso2am*** config folder:

```
cp /home/global.un.org/crodrigo/api-manager.xml /opt/software/wso2am-1.8.0/repository/conf/api-manager.xml
```

Edit ***api-manager.xml*** and change hostname, port number, user name and password for BAM server as appropriate

```
vi /opt/software/wso2am-1.8.0/repository/conf/api-manager.xml
```

```
[...]
<BAMServerURL>tcp://un-api-t-mosrv-001.edc.un.org:7614/</BAMServerURL>
[...]
<Access-Control-Allow-Origin>http://un-api-t-gw-002.edc.un.org:9763/store,
https://un-api-t-gw-002.edc.un.org:9443/store</Access-Control-Allow-Origin>
```

Make a backup of the current ***master-datasources.xml*** file

```
cp /opt/software/wso2am-1.8.0/repository/conf/datasources/master-datasources.xml /home/global.un.org/crodrigo/master-datasources.xml.bak
```

Copy ***master-datasources.xml*** file to ***wso2am*** repository folder and edit it:

```
cp /home/global.un.org/crodrigo/master-datasources.xml /opt/software/wso2am-1.8.0/repository/conf/datasources/master-datasources.xml
vi /opt/software/wso2am-1.8.0/repository/conf/datasources/master-datasources.xml
```

Edit IPs and user credentials for all ***<datasource>*** sections

```
<datasource>
    <name>WSO2_CARBON_DB</name>
    <description>The datasource used for registry and user manager</description>
    <jndiConfig>
        <name>jdbc/WSO2CarbonDB</name>
    </jndiConfig>
    <definition type="RDBMS">
        <configuration>
            <url>jdbc:mysql:loadbalance://10.130.132.92,10.130.132.93/regdbcore?autoReconnect=true&amp;relaxAutoCommit=true</url>
            <username>regadmin</username>
            <password>password</password>
            <driverClassName>com.mysql.jdbc.Driver</driverClassName>
            <maxActive>80</maxActive>
            <maxWait>60000</maxWait>
            <minIdle>5</minIdle>
            <testOnBorrow>true</testOnBorrow>
            <validationQuery>SELECT 1</validationQuery>
            <validationInterval>30000</validationInterval>
        </configuration>
    </definition>
</datasource>

<datasource>
    <name>WSO2AM_DB</name>
    <description>The datasource used for API Manager database</description>
    <jndiConfig>
        <name>jdbc/WSO2AM_DB</name>
    </jndiConfig>
    <definition type="RDBMS">
        <configuration>
            <url>jdbc:mysql:loadbalance://10.130.132.92,10.130.132.93/regdbam?autoReconnect=true&amp;relaxAutoCommit=true</url>
            <username>regadmin</username>
            <password>password</password>
            <driverClassName>com.mysql.jdbc.Driver</driverClassName>
            <maxActive>80</maxActive>
            <maxWait>60000</maxWait>
            <minIdle>5</minIdle>
            <testOnBorrow>true</testOnBorrow>
            <validationQuery>SELECT 1</validationQuery>
            <validationInterval>30000</validationInterval>
        </configuration>
    </definition>
</datasource>

<datasource>
    <name>WSO2AM_STATS_DB</name>
    <description>The datasource used for getting statistics to API Manager</description>
    <jndiConfig>
        <name>jdbc/WSO2AM_STATS_DB</name>
    </jndiConfig>
    <definition type="RDBMS">
        <configuration>
            <url>jdbc:mysql://un-api-t-monds-001.edc.un.org:3306/regdbstats?autoReconnect=true&amp;relaxAutoCommit=true</url>
            <username>regadmin</username>
            <password>password</password>
            <driverClassName>com.mysql.jdbc.Driver</driverClassName>
            <maxActive>80</maxActive>
            <maxWait>60000</maxWait>
            <minIdle>5</minIdle>
```

```
            <testOnBorrow>true</testOnBorrow>
            <validationQuery>SELECT 1</validationQuery>
            <validationInterval>30000</validationInterval>
        </configuration>
    </definition>
</datasource>
```

Passwords are stored in Keepass

Make a backup of the current ***registry.xml*** file:

```
cp /opt/software/wso2am-1.8.0/repository/deployment/server/synapse-configs/default/registry.xml /home/global.un.org/crodrigo/registry.xml.bak
```

Copy ***registry.xml*** file to ***wso2am*** repository folder:

```
cp /home/global.un.org/crodrigo/registry.xml /opt/software/wso2am-1.8.0/repository/deployment/server/synapse-configs/default/registry.xml
```

Make a backup of the current ***user-mgt.xml*** file

```
cp /opt/software/wso2am-1.8.0/repository/conf/user-mgt.xml /home/global.un.org/crodrigo/user-mgt.xml.bak
```

Copy ***user-mgt.xml*** file to ***wso2am*** repository folder

```
cp /home/global.un.org/crodrigo/user-mgt.xml /opt/software/wso2am-1.8.0/repository/conf/user-mgt.xml
```

At this point you can edit the file if needed and change username and password. **In this case we leave it as it is.**

Change owner of API Manager folder:

```
chown -R oictadmin:oictadmin wso2am-1.8.0
```

Execute the ***wsoserver.sh*** script

```
sudo su - oictadmin
cd /opt/software/wso2am-1.8.0/bin
sh wso2server.sh &
```

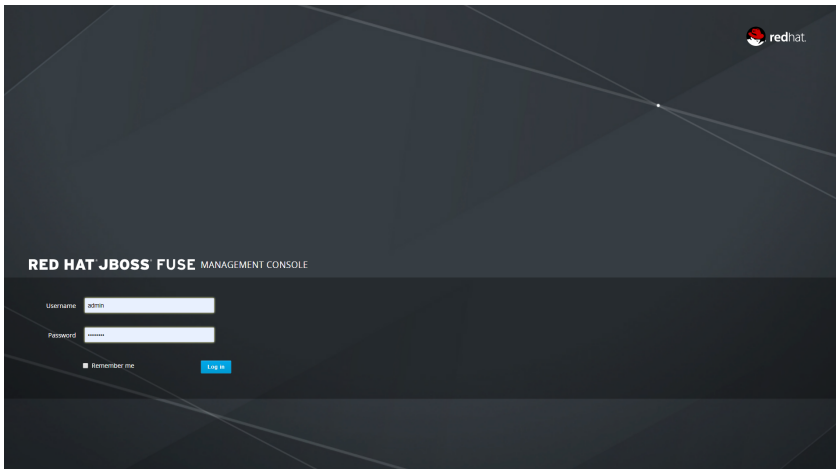
To test if the application works connect to:

https://10.130.131.156:9443/

6. Import profiles

6.1 Create child containers

Login in https://10.130.131.184:8181



Create a new child containers in Fuse console. This is the list of child containers to create:

- On UN-API-T-CL-001 (root0012):

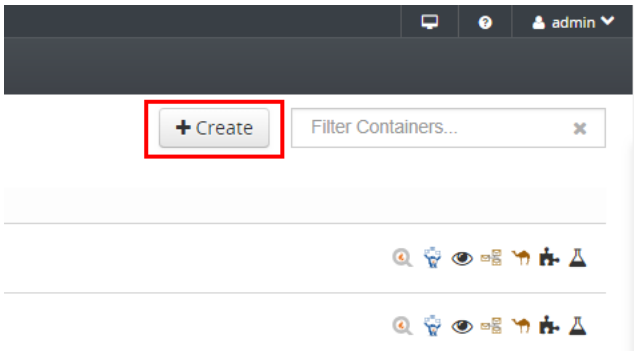
child1-0012
child2-0012
child3-0012-eacvn-events
child4-0012-uniteid
child5-0012-udcorres

- On UN-API-T-CL-002 (root0013):

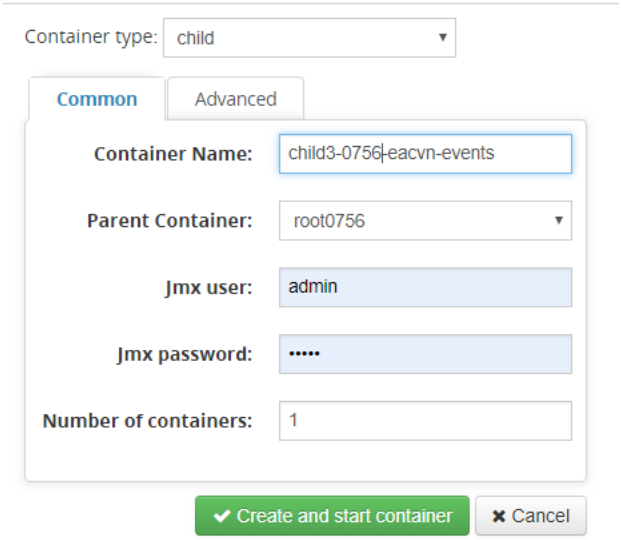
  child1-0013
  child2-0013-udocs
  child3-0013-eacvn-events
  child4-0013-uniteid
  child5-0013-udcorres

- On UN-API-T-CL-003 (root0014):
  child1-0014
  child2-0014-udocs
  child3-0014-eacvn-events
  child4-0014-uniteid
  child5-0014-udcorres

To create a child container, click on ***Create*** on the top right corner.



Chose the type ***child*** and introduce the name of the container. Finally click on ***Create and start container***:



Repeat for the rest of the child containers

6.2 Copy child container files to preproduction

Copy the necessary configuration files in `/opt/software/jboss-fuse-6.2.0.redhat-123/instances/<container-name>/etc` folder.

For example, for ODS Update API, `org.un.integration.ods.doc.api.props.v2.cfg` and `org.un.integration.esb.common.util.v2.props.cfg` configuration files are required. See in the next step the list of files to import.

6.2.1 Compress the files in production environment

Connect to the following production servers and compress the necessary files stored in the child container folders.

Note: We are already creating the tar files with the name of the container in destination (*eg. 0012 instead of 0757*)

In ***UN-API-P-CL-001***:

```
tar -cvzf /home/global.un.org/crodrigo2/child1-0012.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0757/etc/org.un.integration.*
tar -cvzf /home/global.un.org/crodrigo2/child2-0012-udocs.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0757-udocs/etc/org.un.integration.*
tar -cvzf /home/global.un.org/crodrigo2/child3-0012-eacvn-events.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child3-0757-eacvn-events/etc/org.un.integration.*
tar -cvzf /home/global.un.org/crodrigo2/child4-0012-uniteid.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0757-uniteid/etc/org.un.integration.*
tar -cvzf /home/global.un.org/crodrigo2/child5-0012-udcorres.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child5-0757-udcorres/etc/org.un.integration.*
```

In ***UN-API-P-CL-002***:

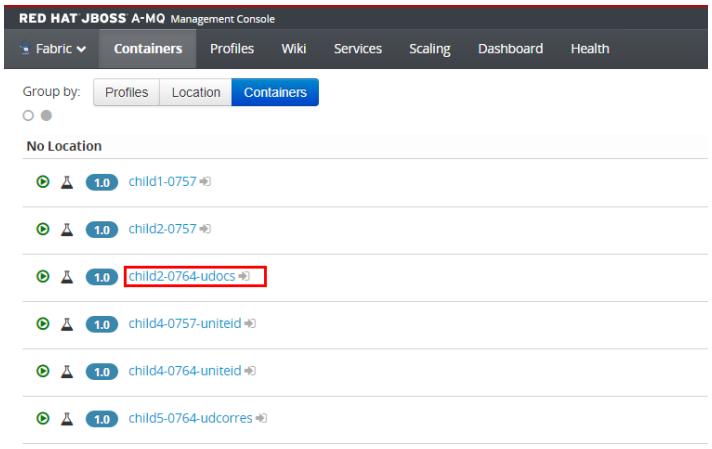
```
tar -cvzf /home/global.un.org/crodrigo2/child1-0013.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0763/etc/org.un.integration.*
tar -cvzf /home/global.un.org/crodrigo2/child2-0013-udocs.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0763-udocs/etc/org.un.integration.*
tar -cvzf /home/global.un.org/crodrigo2/child3-0013-eacvn-events.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child3-0763-eacvn-events/etc/org.un.integration.*
tar -cvzf /home/global.un.org/crodrigo2/child4-0013-uniteid.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0763-uniteid/etc/org.un.integration.*
tar -cvzf /home/global.un.org/crodrigo2/child5-0013-udcorres.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child5-0763-udcorres/etc/org.un.integration.*
```

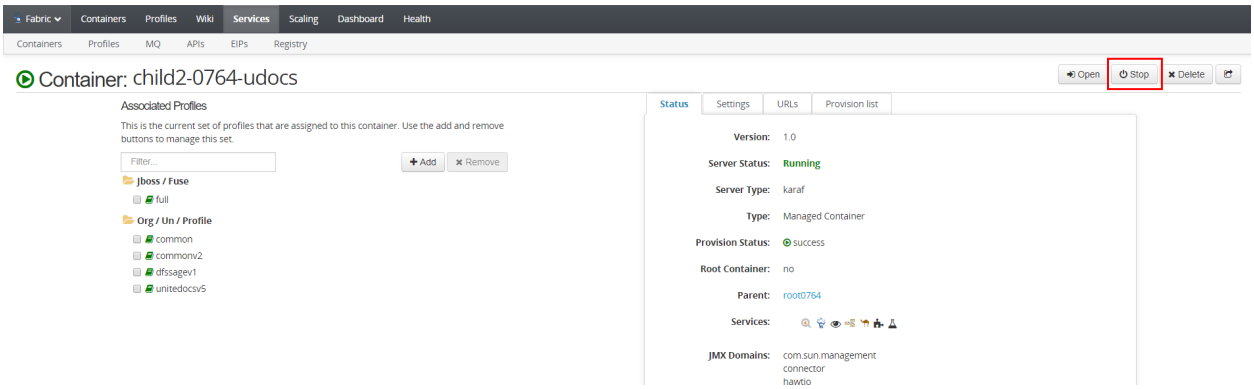
In ***UN-API-P-CL-003***:

```
tar -cvzf /home/global.un.org/crodrigo2/child1-0014.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0764/etc/org.un.integration.*
tar -cvzf /home/global.un.org/crodrigo2/child2-0014-udocs.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0764-udocs/etc/org.un.integration.*
tar -cvzf /home/global.un.org/crodrigo2/child3-0014-eacvn-events.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child3-0764-eacvn-events/etc/org.un.integration.*
tar -cvzf /home/global.un.org/crodrigo2/child4-0014-uniteid.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0764-uniteid/etc/org.un.integration.*
tar -cvzf /home/global.un.org/crodrigo2/child5-0014-udcorres.tar.gz /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child5-0764-udcorres/etc/org.un.integration.*
```

6.2.2 Extract the files in preproduction

Back to preproduction environment, stop the containers you have created in web console by clicking on the child container's name and then in ***Stop***.





Repeat for all child containers.

Use WinSCP to copy the tar files to correspondent preproduction servers.

Extract the files imported into the child container's folder:

In ***UN-API-T-CL-001***:

```
tar -xvzf /home/global.un.org/crodrigo/child2-0012.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0012/etc/ --strip-components 6
tar -xvzf /home/global.un.org/crodrigo/child2-0012-udocs.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0012-udocs/etc/ --strip-components 6
tar -xvzf /home/global.un.org/crodrigo/child3-0012-eacvn-events.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child3-0012-eacvn-events/etc/ --strip-components 6
tar -xvzf /home/global.un.org/crodrigo/child4-0012-uniteid.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0012-uniteid/etc/ --strip-components 6
tar -xvzf /home/global.un.org/crodrigo/child5-0012-udcorres.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child5-0012-udcorres/etc/ --strip-components 6
```

In ***UN-API-T-CL-002***:

```
tar -xvzf /home/global.un.org/crodrigo/child1-0013.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0013/etc/ --strip-components 6
tar -xvzf /home/global.un.org/crodrigo/child2-0013-udocs.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0013-udocs/etc/ --strip-components 6
tar -xvzf /home/global.un.org/crodrigo/child3-0013-eacvn-events.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child3-0013-eacvn-events/etc/ --strip-components 6
tar -xvzf /home/global.un.org/crodrigo/child4-0013-uniteid.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0013-uniteid/etc/ --strip-components 6
tar -xvzf /home/global.un.org/crodrigo/child5-0013-udcorres.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child5-0013-udcorres/etc/ --strip-components 6
```

In ***UN-API-T-CL-003***:

```
tar -xvzf /home/global.un.org/crodrigo/child1-0014.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0014/etc/ --strip-components 6
tar -xvzf /home/global.un.org/crodrigo/child2-0014-udocs.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0014-
```

```
udocs/etc/ --strip-components 6
tar -xvzf /home/global.un.org/crodrigo/child3-0014-eacvn-events.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child3-0014-eacvn-events/etc/ --strip-components 6
tar -xvzf /home/global.un.org/crodrigo/child4-0014-uniteid.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0014-uniteid/etc/ --strip-components 6
tar -xvzf /home/global.un.org/crodrigo/child5-0014-udcorres.tar.gz -C /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child5-0014-udcorres/etc/ --strip-components 6
```

6.2.3 Edit credentials

Edit all ***org.un.integration.**** files and change credentials pointing to production services to preproduction ones.

In ***UN-API-T-CL-001***:

```
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0012/etc/org.un.integration.esb.common.util.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0012/etc/org.un.integration.ods.doc.api.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0012/etc/org.un.integration.ods.doc.api.props.v2.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0012/etc/org.un.integration.ods.doc.scheduler.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0012-udocs/etc/org.un.integration.dfs.sage.api.v1.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0012-udocs/etc/org.un.integration.esb.common.util.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0012-udocs/etc/org.un.integration.esb.common.util.v2.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0012-udocs/etc/org.un.integration.esb.unitedocs.api.v1.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0012-udocs/etc/org.un.integration.esb.unitedocs.api.v1.props.cfg.old
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0012-udocs/etc/org.un.integration.udocs.correspondence.api.v1.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child3-0012-eacvn-events/etc/org.un.integration.accesscontrol.amq.routes.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child3-0012-eacvn-events/etc/org.un.integration.esb.common.util.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0012-uniteid/etc/org.un.integration.eidms.userdata.api.v3.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0012-uniteid/etc/org.un.integration.esb.common.util.v2.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0012-uniteid/etc/org.un.integration.ineed.sr.api.props.v2.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0012-uniteid/etc/org.un.integration.o365.services.api.v1.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child5-0012-udcorres/etc/org.un.integration.esb.common.util.v2.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child5-0012-udcorres/etc/org.un.integration.udocs.correspondence.api.v1.props.cfg
```

In ***UN-API-T-CL-002***:

```
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0013/etc/org.un.integration.esb.common.util.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0013/etc/org.un.integration.ods.doc.api.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0013/etc/org.un.integration.ods.doc.api.props.v2.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0013/etc/org.un.integration.ods.doc.scheduler.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0013-udocs/etc/org.un.integration.dfs.sage.api.v1.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0013-udocs/etc/org.un.integration.esb.common.util.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0013-udocs/etc/org.un.integration.esb.common.util.v2.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0013-udocs/etc/org.un.integration.esb.unitedocs.api.v1.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0013-udocs/etc/org.un.integration.esb.unitedocs.api.v1.props.cfg.old
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0013-udocs/etc/org.un.integration.udocs.correspondence.api.v1.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child3-0013-eacvn-events/etc/org.un.integration.accesscontrol.amq.routes.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child3-0013-eacvn-events/etc/org.un.integration.esb.common.util.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0013-uniteid/etc/org.un.integration.eidms.userdata.api.v3.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0013-uniteid/etc/org.un.integration.esb.common.util.v2.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0013-uniteid/etc/org.un.integration.ineed.sr.api.props.v2.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0013-uniteid/etc/org.un.integration.o365.services.api.v1.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child5-0013-udcorres/etc/org.un.integration.esb.common.util.v2.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child5-0013-udcorres/etc/org.un.integration.udocs.correspondence.api.v1.props.cfg
```

In ***UN-API-T-CL-003***:

```
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0014/etc/org.un.integration.esb.common.util.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0014/etc/org.un.integration.ods.doc.api.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0014/etc/org.un.integration.ods.doc.api.props.v2.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child1-0014/etc/org.un.integration.ods.doc.scheduler.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0014-udocs/etc/org.un.integration.dfs.sage.api.v1.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0014-udocs/etc/org.un.integration.esb.common.util.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0014-udocs/etc/org.un.integration.esb.common.util.v2.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0014-udocs/etc/org.un.integration.esb.unitedocs.api.v1.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0014-udocs/etc/org.un.integration.esb.unitedocs.api.v1.props.cfg.old
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child2-0014-udocs/etc/org.un.integration.udocs.correspondence.api.v1.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child3-0014-eacvn-events/etc/org.un.integration.accesscontrol.amq.routes.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child3-0014-eacvn-events/etc/org.un.integration.esb.common.util.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0014-uniteid/etc/org.un.integration.eidms.userdata.api.v3.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0014-uniteid/etc/org.un.integration.esb.common.util.v2.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0014-uniteid/etc/org.un.integration.ineed.sr.api.props.v2.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child4-0014-uniteid/etc/org.un.integration.o365.services.api.v1.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child5-0014-udcorres/etc/org.un.integration.esb.common.util.v2.props.cfg
vi /opt/software/jboss-fuse-6.2.0.redhat-123/instances/child5-0014-udcorres/etc/org.un.integration.udocs.correspondence.api.v1.props.cfg
```

Start again all the child containers one by one, **waiting until the status show a *success* status**.

6.3 Export profiles from the existing installation

Log in to production server ***UN-API-P-CL-001***

- Go to ***/opt/software/jboss-fuse-6.2.0.redhat-123/bin*** and login to the Fuse client:

```
cd /opt/software/jboss-fuse-6.2.0.redhat-123/bin
./client -u admin
```

- In the Fuse shell, export the profiles to your home folder

```
profile-export /home/global.un.org/crodrigo2/profiles.zip org
```

Copy the ***profiles.zip*** file in the target server ***UN-API-T-CL-001*** using WinSCP

6.4 Import profiles into the target installation

- Connect to the preproduction server ***UN-API-T-CL-001***
- Go to ***/opt/software/jboss-fuse-6.2.0.redhat-123/bin*** and login to the Fuse client:

```
cd /opt/software/jboss-fuse-6.2.0.redhat-123/bin
./client -u admin
```

- In the Fuse shell, import the profiles from the file you copied from production

```
profile-import file:/<path-to-file>/profiles.zip
```

- Now log in to the Fuse web console of the target installation

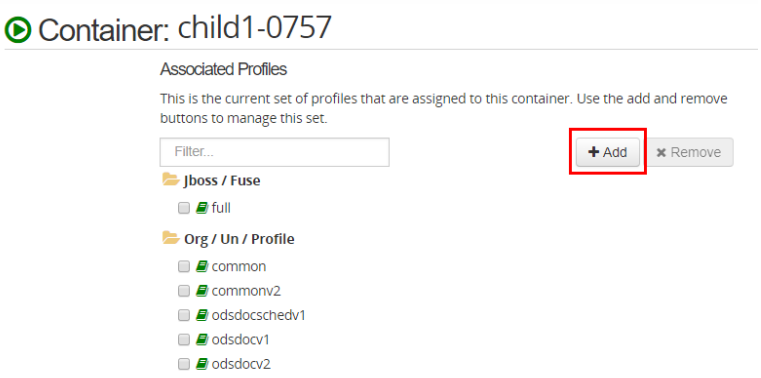
https://10.130.131.184:8181

- Deploy the appropriate profiles for the component (for example, for ODS Update API, deploy *jboss-fuse-full*, *commonv2* and *unodscamelv2* profiles). This is the list of profiles for each container:

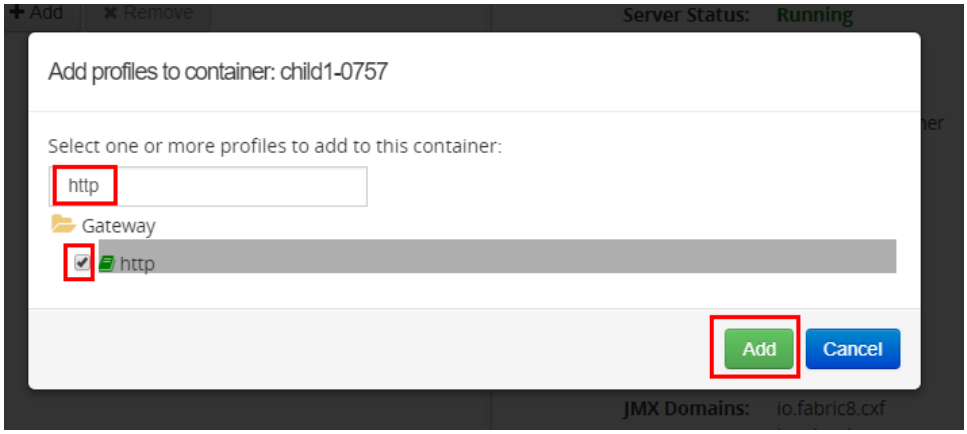
child1-0012	**child1-0013**	**child1-0014**	**child2-0012**	**child2-0013/0014-udocs**
- full	- full	- full	default	full
- common	- common	- common	full	common
- commonv2	- odsdocv2	- odsdocv2	common	commonv2
- odsdocschedv1			commonv2	dfssagev1
- odsdocv1			dfssagev1	unitedocsv5
- odsdocv2			unitedocsv5	

child4-0012/0013/0014-uniteid	**child5-0012/0013/0014-udcorres**	**child6-0012/0013/0014-inspira**
full	http	full
commonv2	full	commonv2
eidmsuserdatav3	commonv2	onehrv1
eidmsuserdatav4	corresv1	
esbineedv6		
o365v1		

To deploy a profile go to the child container and click on ***Add***:



Then search the profile name, select it and click on ***Add***:



For each import wait until the profile is deployed and the container is in ***success*** Provision status: