



Behavioral Health Information Technology and Standards (BHITS) Project

Consent2Share Version 3.4.0 Deployment Guide

Sep 2017
Prepared by FEi Systems

Contents

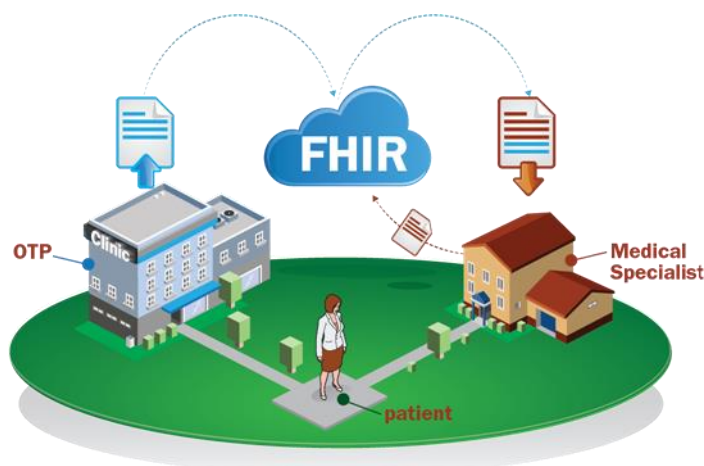
1	INTRODUCTION	4
1.1	Overview	4
1.2	Purpose	4
1.3	Organization of this Guide.....	5
1.4	Technology Stack.....	5
1.5	Prerequisites.....	6
1.6	Technical Support.....	6
2	DEPLOYMENT SERVER SETUP	6
2.1	Docker Installation.....	6
2.1.1	Prerequisites	6
2.1.2	Install Docker and Docker Compose	6
2.1.3	Add User Accounts to Docker Group	6
3	CONSENT2SHARE DEPLOYMENT.....	7
3.1	EHR Edition Setup.....	7
3.1.1	Configure Database Server.....	7
3.1.2	Configure Application Server	8
3.1.3	Configure FHIR Server(Optional)	10
3.1.4	Enable FHIR on App Server (Optional)	11
3.1.5	Compose Containers on Database Server	11
3.1.6	Compose Containers on Application Server.....	11
3.1.7	Compose Containers on FHIR Server (Optional)	11
3.2	HIE Edition Setup.....	12
3.2.1	Configure Database Server.....	12
3.2.2	Configure Application Server	12
3.2.3	Configure HIEOS Server	12
3.2.4	Enable HIEOS on App Server	13
3.2.5	Compose Containers on Database Server	14
3.2.6	Compose Containers on Application Server.....	14
3.2.7	Compose Containers on HIEOS Server	14
3.3	Populate Sample Data	14
3.3.1	Sample Providers (pls).....	14
3.3.2	Sample Value Sets (vss).....	14
3.3.3	Sample Data for Consent Management (pcm).....	15
3.3.4	Sample Data for User Management Service (ums)	15
3.3.5	Sample Document Type Code (phr)	15
3.4	Create Provider User Account	16
3.5	Possible Deployment Errors	16
3.6	UI URLs	16
3.7	Generate and Reconfigure UAA Public and Private Keys.....	17



1 Introduction

1.1 Overview

Specially protected health information (PHI) covered under the federal confidentiality regulation 42 CFR Part 2 (health information from federally assisted drug and alcohol treatment programs) has generally not been included in the electronic exchange of patient information between health care providers. One of the primary reasons is the lack of technology options for patients to share part of their health information while not sharing others.



To address this issue, the Federal Office of the National Coordinator for Health Information Technology (ONC) developed the Data Segmentation for Privacy (DS4P) initiative to allow patients to share portions of an electronic medical record while not sharing others. In collaboration with the ONC, the Substance Abuse and Mental Health Services Administration (SAMHSA) developed the Consent2Share application to address the specific privacy protections for substance use treatment patients covered by the federal confidentiality regulation 42 CFR Part 2.

Consent2Share is an open source application for data segmentation and consent management. This Edition is designed to integrate with existing FHIR (Fast Health Interoperability Resources) systems. Initially, SAMHSA funded the Open Behavioral Health Information Technology Architecture (OBHITA) contract to develop the Consent2Share application. Subsequently, SAMHSA funded the Behavioral Health Information Technology and Standards (BHITS) contract to further develop and conduct pilot testing of Consent2Share. Through a process of electronic consent, the patient controls how his or her sensitive health data will be shared by selecting categories.

1.2 Purpose

This document was prepared by the Consent2Share application developers primarily to document key infrastructure setup, installation, configuration, and deployment technologies required to operationalize Consent2Share. This document is not a step-by-step software application development guide. Rather, this document is a reference guide to help developers and system administrators install, configure, deploy, and validate the key software components that operationalize Consent2Share. Since Consent2Share is designed to integrate with FHIR systems, this document assumes that the reader has in place FHIR Servers and is employing interoperability standards. However, the Consent2Share application can be configured to work without FHIR connections. The Consent2Share development team has an instance of HAPI FHIR Restful Server installed as the FHIR environment. Please refer to the Appendix 4.1 for more information.

1.3 Organization of this Guide

This Deployment Guide is divided into four Chapters:

- Chapter One provides an introduction to Consent2Share and the purpose of this guide
- Chapter Two provides information about setting up the deployment environment based on Docker
- Chapter Three provides instructions to deploy and configure Consent2Share using Docker on Linux servers
- Chapter Four is an appendix that provides information about FHIR Server setup

1.4 Technology Stack

The current version of Consent2Share (Consent2Share Version 3.3.0) employs [Docker](#). In short, Docker is an open-source tool that automates the deployment of applications inside software containers. Docker containers wrap up a piece of software in a complete filesystem that contains everything it needs to run. This includes code, runtime, system tools, and system libraries. This enables the ability that the application will always run the same, regardless of the environment within it is running.

Consent2Share is designed to integrate with existing FHIR systems via interoperability standards. To do so, the technology stack for Consent2Share configuration is as follows. The references for these technologies located in the Appendix 4.2.

- [AngularJS](#)
- [Angular Material](#)
- [Angular CLI](#)
- [Node.js](#)
- [npm](#)
- [MD2](#)
- [RxJS](#)
- [TypeScript](#)
- [JavaScript - ES6](#)
- [HTML5](#)
- [CSS3](#)
- [Oracle Java 8](#)
- [Spring Framework](#)
- [Spring Boot](#)
- [Spring Cloud](#)
- [Apache Maven](#)
- [Apache Tomcat](#)
- [MySQL](#)
- [Flyway](#)
- [Docker and Docker Compose](#)
- [Cloud Foundry User Account and Authentication \(UAA\) Server](#)

1.5 Prerequisites

This document is designed for developers and system administrators who install, configure, deploy, and maintain distributed applications. Familiarity with the following is recommended.

- Basic Linux system administration
- Basic knowledge of Docker and Docker-Compose
- Basic knowledge of Public Key Infrastructure (PKI) and creating SSL certificates

1.6 Technical Support

If you have specific questions about a specific API deployment, setup server environment, or anything related to the Consent2Share application, you should:

- Check the [Consent2Share Project site](#)
- Check the readme files available for each API in [Consent2Share GitHub repository](#).
- Check the [Issues](#) in the Consent2Share repository.

2 Deployment Server Setup

2.1 Docker Installation

The following provides instructions about how to install Docker on a Linux CentOS 7.X server.

2.1.1 Prerequisites

- Docker requires a 64-bit installation regardless of your CentOS version.
- Your kernel must be 3.10 at a minimum, which CentOS 7 runs. To check the CentOS version, run the command “uname -r” in the terminal.
- User account should have sudo or root privileges
- Ensure yum and curl are installed, and networking is operational.

2.1.2 Install Docker and Docker Compose

- Get the [c2s_docker_install.sh](#) and run the file.

```
sh c2s_docker_install.sh
```

- Verify the Docker installation.

```
sudo docker version
```

```
sudo docker run hello-world
```

Output message will contain the following:

Hello from Docker!

This message shows that your installation appears to be working correctly.

- Verify Docker compose installation.

```
sudo docker-compose --version
```

Note: if docker-compose gives the command “not found” try with following:

```
/usr/local/bin/docker-compose --version
```

2.1.3 Add User Accounts to Docker Group

The user accounts that need to run Docker and Docker Compose commands must be added to the Docker group. Run the following command by replacing the *** with the actual username to add a user to the Docker group

```
sudo usermod -aG docker ***
```

3 Consent2Share Deployment

E edition deployment and HIE edition deployment are provided to run the Consent2Share application on Linux. Here we use CentOS 7.X as an example to describe the setups.

Consent2Share Docker images will be downloaded from [Dockerhub BHITS](#) public registry.

3.1 EHR Edition Setup

This option is designed to run all Consent2Share services, UIs on app server, databases on database server, and optional to setup FHIR server.

3.1.1 Configure Database Server

3.1.1.1 Database Server Info

DB Server	MAX	MIN
Memory	20GB	10GB
Storage	80GB	10GB
CPU	2	1

3.1.1.2 Configure Database Server

- Get the [c2s_config.sh](#) and run the file.

```
curl https://raw.githubusercontent.com/bhitsu-dev/consent2share/master/infrastructure/scripts/c2s_config.sh > c2s_config.sh
```

```
sh c2s_config.sh
```

Enter 2 when console prompted "Please select a server to setup:"

```
care/scripts (master)
$ sh c2s_config.sh
This script is used to setup server configuration for Consent2Share.
1. EHR Edition App Server
2. EHR Edition DB Server
3. EHR Edition FHIR Server
4. HIE Edition App Server
5. HIE Edition Db Server
6. HIE Edition Hieos Server
Please select a server to setup:2
```

Expected Results:

The following subfolders and the Consent2Share configurations will be created under '/usr/local/' folder.

- Java
 - ✓ [docker-compose-db-server](#) file : docker-compose.yml
- java /C2S_PROPS/pcm
 - ✓ [insert_consent_attestation_term.sql](#) file:
 - /pcm/ insert_consent_attestation_term.sql
 - ✓ [insert_consent_revocation_term.sql](#) file:
 - /pcm/ insert_consent_revocation_term.sql
 - ✓ [insert_purposes.sql](#) file:

- /pcm/ insert_purposes.sql
 - java /C2S_PROPS/phr
 - ✓ [Document Type](#) file: /phr/insert_document_type_codes.sql
 - java /C2S_PROPS/pls
 - ✓ [Pls sample provider data](#) file: /pls/pls_db_sample.sql
 - ✓ [State Code Lookup Data](#): /pls/insert_state_code_lookup_data.sql
 - java /C2S_PROPS/vss
 - ✓ [VSS sample provider data](#) file: /vss/vss_db_sample.sql
 - java /C2S_PROPS/ums
 - ✓ [Administrative Gender Code](#) file: /ums/insert_administrative_gender_code_lookup_data.sql
 - ✓ [Country Code file](#) : /ums/ insert_country_code_lookup_data.sql
 - ✓ [Locale file](#): /ums/ insert_locale_lookup_data.sql
 - ✓ [Identifier file](#): /ums/insert_npi_identifier_system.sql
 - ✓ [Role sample file](#): /ums/insert_role_scopes_lookup_data.sql
 - ✓ [State Code file](#) : /ums/insert_state_code_lookup_data.sql
 - The [c2s_db_env.sh](#) file will be placed as c2s_env.sh file in '/etc/profile.d/' folder.
- Re-login to the server in order for the file c2s_env.sh file to run automatically during the login.
- Verify by checking any variables mentioned in the file
Ex: echo \${C2S_BASE_PATH} should show the value set in the file

3.1.2 Configure Application Server

3.1.2.1 App Server Info

APP Server	MAX	MIN
Memory	40 GB	20GB
Storage	80GB	15GB
CPU	2	1

3.1.2.2 Configure App Server

- Get the [c2s_config.sh](#) and run the file.
- ```
curl https://raw.githubusercontent.com/bhits-dev/consent2share/master/infrastructure/scripts/c2s_config.sh > c2s_config.sh
sh c2s_config.sh
```
- Enter 1 when console prompted "Please select a server to setup:"



```
$ sh c2s_config.sh
This script is used to setup server configuration for Consent2Share.
1. EHR Edition App Server
2. EHR Edition DB Server
3. EHR Edition FHIR Server
4. HIE Edition App Server
5. HIE Edition Db Server
6. HIE Edition Hieos Server
Please select a server to setup:1|
```

### Expected Results:

The following subfolders and the Consent2Share configurations will be created under '/usr/local/' folder.

- Java
  - ✓ [docker-compose-app-server](#) file : docker-compose.yml
- Application Files
  - java /C2S\_PROPS/uua
    - ✓ [uua configuration](#) file: /uua/uua.yml
  - java /C2S\_PROPS/[c2s-config-data](#)
    - ✓ [Consent2share configuration](#) files: /c2s-config-data

Note: If not present clone it by using the following command

```
git clone https://github.com/bhits-dev/c2s-config-data.git
```
  - java /keystore
  - java/C2s\_PROPS/ums
    - Create and activate provider user account
      - [create activate provider user.sh](#)
    - Activate Provider user account file :
      - [activate user account.sh](#)
    - The [c2s\\_app\\_env.sh](#) file will be placed as c2s\_env.sh file in '/etc/profile.d/' folder.

### ➤ Edge-server security:

- Create/Obtain a valid SSL certificate
- Export the public and private keys from the SSL certificate to a JKS formatted keystore file named 'edge-server.keystore'
- upload the 'edge-server.keystore' file into '/usr/local/java/keystore' folder
- Add the following in the
 

```
'/usr/local/java/C2S_PROPS/c2s-config-data/edge-server.yml' file.
```

```
spring.profiles: your_app_Server_specific_profile
server:
 ssl:
 key-store: /java/keystore/edge-server.keystore
 key-store-password: "keystore password"
```

- Modify the ENV\_APP\_PROFILE= your\_app\_Server\_specific\_profile in appServerConfig() function in the '/etc/profile.d/ c2s\_env.sh' file.
- Modify the following configuration files:
  - Set edge server, config server, and SMTP variables to the server specific values in the '/etc/profile.d/ c2s\_env.sh' file.
  - There are many Consent2Share API codes. The configurations in these codes can be overridden using the corresponding API YAMLS that are available in the c2s-config-data folder. The structure of the API YAMLS should be similar to the corresponding application.yml mentioned in the each API code. They can be found in the 'src/main/resources' folder.
    - ✓ For instance, to override database variables for PCM API. The following can be added in the pcm.yml as below
 

```
spring.profiles: your_app_Server_specific_profile
spring:
 datasource:
 url: "database url"
 username: "database user name"
 password: "{cipher} encrypted database pwd"
```
    - ✓ Follow the instructions mentioned in the [config server api](#) to encrypt/decrypt server specific sensitive configurations.
- Re-login to the server in order for the file `c2s\_env.sh` to run automatically during the login
  - Verify by checking any variable mentioned in the file
 

Ex: echo \${C2S\_BASE\_PATH} should show the value set in the file

### 3.1.3 Configure FHIR Server(Optional)

#### 3.1.3.1 FHIR Server Info

| APP Server | MAX   | MIN  |
|------------|-------|------|
| Memory     | 40 GB | 20GB |
| Storage    | 80GB  | 15GB |
| CPU        | 2     | 1    |

#### 3.1.3.2 Configure FHIR Server

- Get the [c2s\\_config.sh](#) and run the file.
 

```
curl https://raw.githubusercontent.com/bhits-dev/consent2share/master/infrastructure/scripts/c2s_config.sh > c2s_config.sh
sh c2s_config.sh
```

Enter 3 when console prompted "Please select a server to setup:"

```
$ sh c2s_config.sh
This script is used to setup server configuration for Consent2Share.
1. EHR Edition App Server
2. EHR Edition DB Server
3. EHR Edition FHIR Server
4. HIE Edition App Server
5. HIE Edition Db Server
6. HIE Edition Hieos Server
Please select a server to setup:3]
```

### Expected Results:

The following subfolders and the Consent2Share configurations will be created under '/usr/local/' folder.

- Java
  - ✓ [docker-compose-fhir-server](#) file : docker-compose.yml
- The [c2s\\_fhir\\_env.sh](#) file will be placed as c2s\_env.sh file in '/etc/profile.d/' folder.

#### 3.1.4 Enable FHIR on App Server (Optional)

By default, FHIR publishing is disabled. To enable Consent2Share to publish patient and consent resource to fhir server, go to app server:

- Add following lines in ums.yml under /user/local/java/C2S\_PROPS/c2s-config-data/c2s.ums.fhir.publish:
  - enabled: true
  - serverUrl: [http://fhir\\_server\\_ip/hapi-fhir-jpaserver/baseStu3](http://fhir_server_ip/hapi-fhir-jpaserver/baseStu3)
- Replacing fhir\_server\_ip in above serverUrl properties by FHIR server IP address
- Add following lines in pcm.yml under /user/local/java/C2S\_PROPS/c2s-config-data/c2s.pcm.consent.publish:
  - enabled: true
  - serverUrl: [http://fhir\\_server\\_ip/hapi-fhir-jpaserver/baseStu3](http://fhir_server_ip/hapi-fhir-jpaserver/baseStu3)
- Replacing fhir\_server\_ip in above serverUrl properties by FHIR server IP address

#### 3.1.5 Compose Containers on Database Server

Run the following command from the '/usr/local/java' folder to start up all databases:

```
docker-compose up -d
```

Run 'docker ps -a' to verify all the containers are up running except data-only containers.

#### 3.1.6 Compose Containers on Application Server

Run the following command from the '/usr/local/java' folder to start up all Consent2Share services, UIs:

```
docker-compose up -d
```

Run 'docker ps -a' to verify all the containers are up running.

#### 3.1.7 Compose Containers on FHIR Server (Optional)

Run the following command from the '/usr/local/java' folder to start up FHIR services and DB:

```
docker-compose up -d
```

Run 'docker ps -a' to verify all the containers are up running except data-only containers.

## 3.2 HIE Edition Setup

This option is designed to run all Consent2Share services, UIs on app server, databases on database server, and hie services and databases on hieos server.

### 3.2.1 Configure Database Server

Please refer 3.1.1 to setup database server.

- Type “5” when run the c2s\_config.sh
- [docker-compose-db-server](#) file placed as /usr/local/java/docker-compose.yml
- The [c2s\\_db\\_env.sh](#) file will be placed as c2s\_env.sh file in ‘/etc/profile.d/’ folder.

### 3.2.2 Configure Application Server

Please refer 3.1.1 to setup application server

- Type “4” when run the c2s\_config.sh
- [docker-compose-app-server](#) file will be placed as /usr/local/java/docker-compose.yml
- The [c2s\\_app\\_env.sh](#) file will be placed as c2s\_env.sh file in ‘/etc/profile.d/’ folder.
- [IExHub.properties](#) file will be placed as /usr/local/java/C2S\_PROPS/iexhub/config/IExHub.properties

### 3.2.3 Configure HIEOS Server

#### 3.2.3.1 HIEOS Server Info

| APP Server | MAX   | MIN  |
|------------|-------|------|
| Memory     | 40 GB | 20GB |
| Storage    | 80GB  | 15GB |
| CPU        | 2     | 1    |

#### 3.2.3.2 Configure HIEOS Server

- Get the [c2s\\_config.sh](#) and run the file.

```
curl https://raw.githubusercontent.com/bhits-dev/consent2share/master/infrastructure/scripts/c2s_config.sh > c2s_config.sh
sh c2s_config.sh
```

Enter 6 when console prompted “Please select a server to setup:”

```
$ sh c2s_config.sh
This script is used to setup server configuration for Consent2Share.
1. EHR Edition App Server
2. EHR Edition DB Server
3. EHR Edition FHIR Server
4. HIE Edition App Server
5. HIE Edition Db Server
6. HIE Edition Hieos Server
Please select a server to setup:6
```

#### Expected Results:

The following subfolders and the Consent2Share configurations will be created under ‘/usr/local/’ folder.

- Java
  - ✓ [docker-compose-hieos-server](#) file : docker-compose.yml

- java /C2S\_PROPS/openempi-db  
[create\\_new\\_database\\_schema-2.2.0.sql](#) file:  
 create\_new\_database\_schema-2.2.0.sql  
[update\\_database\\_schema-2.2.1.sql](#) file:  
 update\_database\_schema-2.2.1.sql  
[update\\_database\\_schema-2.2.3.sql](#) file:  
 update\_database\_schema-2.2.3.sql  
[update\\_database\\_schema-2.2.4.sql](#) file:  
 update\_database\_schema-2.2.4.sql  
[update\\_database\\_schema-2.2.6.sql](#) file:  
 update\_database\_schema-2.2.6.sql  
[update\\_database\\_schema-2.2.7.sql](#) file:  
 update\_database\_schema-2.2.7.sql
- java /C2S\_PROPS/hieos-db/adt  
[createadtdddl.sql](#) file:  
 createadtdddl.sql
- java /C2S\_PROPS/hieos-db/log  
[createlogddl.sql](#) file:  
 createadtdddl.sql
- java /C2S\_PROPS/hieos-db/registry  
[createregistryddl.sql](#) file:  
 createregistryddl.sql
- java /C2S\_PROPS/hieos-db/repository  
[createrepoddddl.sql](#) file:  
 createrepoddddl.sql
- The [c2s\\_hieos\\_env.sh](#) file will be placed as c2s\_env.sh file in '/etc/profile.d/' folder.

### 3.2.4 Enable HIEOS on App Server

To enable Consent2Share to publish patient and consent resource to hieos server, go to app server:

- Add following lines in ums.yml under /user/local/java/C2S\_PROPS/c2s-config-data/  
 c2s.ums.fhir.publish:  
 enabled: true  
 serverUrl: http://iexhub.c2s.com:8080/iexhub/services
- Add following lines in pcm.yml under /user/local/java/C2S\_PROPS/c2s-config-data/  
 c2s.pcm.consent.publish:  
 enabled: true  
 serverUrl: <http://iexhub.c2s.com:8080/iexhub/services>
- Replacing following properties in IExHub.properties under  
 /usr/local/java/C2S\_PROPS/iexhub/config/
  - ✓ XdsBRegistryEndpointURI=http://hieos\_server\_ip:82/axis2/services/xdsregistryb

- ✓ XdsBRepositoryEndpointURI=http:// hieos\_server\_ip :82/axis2/services/xdsrepositoryb
- ✓ PIXManagerEndpointURI=http:// hieos\_server\_ip/openempi-admin/services/PIXManager\_Port\_Soap12
- ✓ PDQManagerEndpointURI=http:// hieos\_server\_ip/openempi-admin/services/PDQSupplier\_Port\_Soap12

### 3.2.5 Compose Containers on Database Server

Run the following command from the '/usr/local/java' folder to start up all databases:

```
docker-compose up -d
```

Run 'docker ps -a' to verify all the containers are up running except data-only containers.

### 3.2.6 Compose Containers on Application Server

Run the following command from the '/usr/local/java' folder to start up all Consent2Share services, UIs:

```
docker-compose up -d
```

Run 'docker ps -a' to verify all the containers are up running.

### 3.2.7 Compose Containers on HIEOS Server

Run the following command from the '/usr/local/java' folder to start up FHIR services and DB:

```
docker-compose up -d
```

Run 'docker ps -a' to verify all the containers are up running except data-only containers.

## 3.3 Populate Sample Data

Perform the following in the database docker containers

### 3.3.1 Sample Providers (pls)

- Login to docker pls database container
 

```
docker ps | grep pls-db
```

```
docker exec -it <<pls-db container id>> bash
```
- Run pls\_db\_sample.sql from container directory `/java/C2S\_PROPS/pls`.
  - `cd /java/C2S_PROPS/pls`
  - `mysql -p`  
Enter root pwd(default is admin)
  - `use pls;`
  - `source insert_state_code_lookup_data.sql;`
  - `source pls_db_sample.sql;`

### 3.3.2 Sample Value Sets (vss)

- Login to docker vss database container
 

```
docker ps | grep vss-db
```

```
docker exec -it <<vss-db container id>> bash
```
- Run vss\_db\_sample.sql from container directory `/java/C2S\_PROPS/vss`.
  - `cd /java/C2S_PROPS/vss`
  - `mysql -p`

Enter root pwd(default is admin)

- use vss;
- source vss\_db\_sample.sql;

### 3.3.3 Sample Data for Consent Management (pcm)

- Login to docker pcm database container  
docker ps | grep pcm-db  
docker exec -it <<pcm-db container id>> bash
- Run pcm sample files from container directory `/java/C2S\_PROPS/pcm`.
  - cd /java/C2S\_PROPS/pcm
  - mysql -p  
Enter root pwd(default is admin)
  - use pcm;
  - source insert\_consent\_attestation\_term.sql;
  - source insert\_consent\_revocation\_term.sql;
  - source insert\_purposes.sql;

### 3.3.4 Sample Data for User Management Service (ums)

- Login to docker ums database container  
docker ps | grep ums-db  
docker exec -it <<ums-db container id>> bash
- Run pcm sample files from container directory `/java/C2S\_PROPS/ums`.
  - cd /java/C2S\_PROPS/ums
  - mysql -p  
Enter root pwd(default is admin)
  - use ums;
  - source insert\_administrative\_gender\_code\_lookup\_data.sql;
  - source insert\_country\_code\_lookup\_data.sql;
  - source insert\_locale\_lookup\_data.sql;
  - source insert\_role\_scopes\_lookup\_data.sql;
  - source insert\_state\_code\_lookup\_data.sql;
  - Add sample identifier system  
insert into identifier\_system (display, oid, system, reassignable)  
values ('BHITS EHR System', null, 'http://github.com/bhits', 0);

### 3.3.5 Sample Document Type Code (phr)

- Login to docker pls database container  
docker ps | grep phr-db  
docker exec -it <<phr-db container id>> bash
- Run phr\_db\_sample.sql from container directory `/java/C2S\_PROPS/phr`.
  - cd /java/C2S\_PROPS/phr
  - mysql -p  
Enter root pwd(default is admin)
  - use phr;
  - source insert\_document\_type\_codes.sql;

### 3.4 Create Provider User Account

Perform the following in the Application Server

- Login to docker ums application container  
docker ps | grep ums  
docker exec -it <<ums container id>> bash
- Run create\_activate\_provider\_user.sh from container directory `/java/C2S\_PROPS/ums`.
  - cd /java/C2S\_PROPS/ums
  - chmod 700 create\_activate\_provider\_user.sh
  - ./create\_activate\_provider\_user.sh

Note: If the script errors out with following error  
bash: ./create\_activate\_provider\_user.sh: /bin/bash^M: bad interpreter: No such file or directory  
Convert line separator of the file to unix and run again

  - Enter sample provider account details. These credentials are used to login to the provider ui.

### 3.5 Possible Deployment Errors

If you encounter an error in the deployment:

- *ERROR: for dss.c2s.com UnixHTTPConnectionPool(host='localhost', port=None): Read timed out. (read timeout=60)*

Follow the steps below to resolve the error:

1. Restart the Docker service: sudo service docker restart
2. Check for all Docker containers that are running: docker ps -a  
If you notice any containers that are exited or down except the data-only containers based on `busybox` image, follow the next steps
3. For instance, if MySQL containers are not running
  - a. Go to /usr/local/java and then remove all containers : docker-compose down
  - b. Go to /usr/local/java and then remove mysql folder : sudo rm -rf mysql/
4. Start up all containers: Re-run from `/usr/local/java` folder: docker-compose up -d

- *ERROR: for any issues while mounting volumes to the containers from `/usr/local/java`*

If SELinux is enabled, run the command below to assign the relevant SELinux policy type as a workaround to prevent mounting issues.

```
sudo chcon -Rt svirt_sandbox_file_t /usr/local/java
```

### 3.6 UI URLs

- Consent2Share Provider UI: [https://<application\\_server>/provider-ui](https://<application_server>/provider-ui)
  - Use the credentials that were created as part of the [Create Provider User Account](#) section.
  - Follow the [Consent2Share Provider User Guide](#) to verify Consent2Share provider user features



By default, Consent2Share comes with sample providers as documented in Section#3.3. Please refer to [Consent2Share Patient User Guide](#) to add providers and then create consents

- Consent2Share Staff UI: [https://<application\\_server>/staff-ui](https://<application_server>/staff-ui)
  - By default, Consent2Share comes with a staff admin user c2s-admin@mailinator.com.
  - Login to Consent2Share staff UI as a staff using username 'c2s-admin@mailinator.com' and password 'AAA#aaa1'
  - Follow the [Consent2Share Staff User Guide](#) to verify Consent2Share staff admin features

By default, Consent2Share comes with sample providers as documented in Section#3.3. Please refer to [Consent2Share Patient User Guide](#) to add providers and then create consents

- Consent2Share UI: [https://<application\\_server>/c2s-ui](https://<application_server>/c2s-ui)
  - Create and activate a patient using provider-ui mentioned in [Consent2Share provider User Guide](#).
  - Follow the [Consent2Share Patient User Guide](#) to verify Consent2Share patient features
    - ✓ By default, Consent2Share comes with sample providers as documented in Section#3.3. Please refer to [Consent2Share Patient User Guide](#) to add providers and then create consents.

### 3.7 Generate and Reconfigure UAA Public and Private Keys

By default, Consent2Share has created public and private keys that are available in the configuration files. It is recommended to change them, especially in production environment.

- Create a temporary folder uaa-keystore under /usr/local/java
- Go to uaa-keystore folder, Run following in command line to generate a pair of public and private key. Enter pass phrase when promoted.
  - openssl genrsa -des3 -out uaa\_token\_key\_private.pem 2048
  - openssl rsa -in uaa\_token\_key\_private.pem -outform PEM -pubout -out uaa\_token\_key\_public.pem
  - openssl rsa -in uaa\_token\_key\_private.pem -out uaa\_token\_key\_private\_unencrypted.pem -outform PEM
- Update uaa.yml under /usr/local/java/C2S\_PROPS/uaa.
  - Replace jwt.token.verification-key with public key in uaa\_token\_key\_public.pem.
  - Replace jwt.token.signing-key with private key in uaa\_token\_key\_private\_unencrypted.pem.
- Update c2s\_docker.sh under /etc/profile.d
  - Replace UAA\_PUBLIC\_KEY with public key in uaa\_token\_key\_public.pem.
    - ✓ Ensure no spaces in the key value, check the default value for reference
  - Re-login to the server for latest UAA\_PUBLIC\_KEY to be effective.