WildFly 8.2 アプリケーション サーバでの SSL の有効化

WildFly 11.1、11.1.1、および 12.0 での SSL の有効化

前提条件

- 次の Java 開発キットのいずれかがあることを確認します。
  - Oracle JDK 1.8.0_77 以上
  - サーバ内に Open JDK 1.8.0_77 以上があること。
- クライアント VM に JAVA_HOME 変数を設定する必要があります。
- Linux マシンおよび Windows マシン では、すべてのクラスタ管理スクリプト内の --connect 変数に次のプレフィックスを追加します。
  - --user=adminuser --password=newscale
- cisco.com で利用可能なファイルをダウンロードします。これらのダウンロードされたファイルの名前を次の表に記載するとおり既存のファイル名に変更し、既存のファイルを新しいファイルに置き換えます。次の表は、置き換えが必要なファイルのすべての詳細を示します。

(注) ユーザ名とパスワードが変更される場合は、必要な変更を行う必要があります。

インストール タイプ	データベース タイプ	既存ファイル	新しいファイル	コメント	新しいファイルのダウンロード リンク
標準的なスタンドアロン/カスタムのスタンドアロン	SQL	C:\InstallDirectory\wildfly-8.2.0.Final\ServiceCatalogServer\configuration**standalone-full.xml**	Standalone-full_RC_SQL.xml	新しいファイルを既存のパスに移動し、その名前を**太字**で記載する名前に置き換えます。	cisco.com
標準的なスタンドアロン/カスタムのスタンドアロン	SQL	C:\InstallDirectory\wildfly-8.2.0.Final\ServiceLinkServer\configuration**standalone-full.xml**	Standalone-full_SL_SQL.xml		
標準的なスタンドアロン/カスタムのスタンドアロン	Oracle	C:\InstallDirectory\wildfly-8.2.0.Final\ServiceCatalogServer\configuration**standalone-full.xml**	Standalone-full_RC_Oracle.xml		
標準的なスタンドアロン/カスタムのスタンドアロン	Oracle	C:\InstallDirectory\wildfly-8.2.0.Final\ServiceLinkServer\configuration**standalone-full.xml**	Standalone-full_SL_Oracle.xml		
2VM クラスタ	SQL	C:\InstallDirectory\wildfly-8.2.0.Final\domain\configuration**domain.xml**	2VM_SQL_domain.xml		
2VM クラスタ	Oracle	C:\InstallDirectory\wildfly-8.2.0.Final\domain\configuration**domain.xml**	2VM_Oracle_domain.xml		
4VM クラスタ	SQL	C:\InstallDirectory\wildfly-8.2.0.Final\domain\configuration**domain.xml**	4VM_SQL_domain.xml		

インストール タイプ	データベース タイプ	既存ファイル	新しいファイル	コメント	新しいファイルのダウンロード リンク
4VM クラスタ	Oracle	C:\InstallDirectory\wildfly-8.2.0.Final\domain\configuration**domain.xml**	4VM_Oracle_domain.xml		
標準的なスタンドアロン/カスタムのスタンドアロン/2VM クラスタ/4VM クラスタ	SQL/Oracle	同上	https-users.properties		
標準的なスタンドアロン/カスタムのスタンドアロン/2VM クラスタ/4VM クラスタ	SQL/Oracle	同上	https-roles.properties		
4VM クラスタ	SQL/Oracle	C:\InstallDirectory\wildfly-8.2.0.Final\domain\configuration**host1.xml**	4VM_host1_backup.xml		
2VM クラスタ	SQL/Oracle	C:\InstallDirectory\wildfly-8.2.0.Final\domain\configuration**hostva_backup.xml**	2VM_hostva_backup.xml		
4VM クラスタ	SQL/Oracle	C:\InstallDirectory\wildfly-8.2.0.Final\domain\configuration**host_default.xml**	4VM_host_default.xml		
2VM クラスタ	SQL/Oracle	C:\InstallDirectory\wildfly-8.2.0.Final\domain\configuration**host2.xml**	2VM_host2_backup.xml		
4VM クラスタ	SQL/Oracle	C:\InstallDirectory\wildfly-8.2.0.Final\domain\configuration**host2.xml**	4VM_host2_backup.xml		

標準的なスタンドアロン モードでの SSL の有効化

手順 1　C:\SSL パスで作成を行い、ディレクトリに移動して次の手順に従います。
これにより、サーバ キー（秘密および公開）とクライアント キーのペアが作成されます。

a. 次のコマンドを入力し、サーバの秘密キー（serverkey）とクライアントの秘密キー（clientkey）をそれぞれ作成します。

```
keytool -genkeypair -alias serverkey -keyalg RSA -keysize 2048 -validity 7360
-keystore server.keystore
keytool -genkeypair -alias clientkey -keyalg RSA -keysize 2048 -validity 7360
-keystore client.keystore
```

（注）　秘密キーと公開キーを作成するために入力する情報が、クライアントとサーバの両方で一致していることを確認します。

（注）　キーの作成に使用されるデフォルトのパスワードは *secret* です。新しいパスワードを作成する場合は、*rcjms.properties*、*integration-server.properties*、および *standalone-full.xml* ファイルにある *secret* という語を選択したパスワードに置き換えます。

（注）　名前と姓を要求されたら、ホスト コンピュータの IP アドレスを入力します。

b. 次のコマンドを入力し、キーを証明書にエクスポートします。

```
keytool -export -alias serverkey -keystore server.keystore -rfc -file server.crt
keytool -export -alias clientkey -keystore client.keystore -rfc -file client.crt
```

c. 次のコマンドを入力し、公開キーを証明書からエクスポートし、信頼ストアにインポートします。

```
keytool -import -file server.crt -keystore client.truststore
keytool -import -file client.crt -keystore server.truststore
```

d. **キーストア** ファイルおよび**信頼ストア** ファイルを以下の場所に配置します。

```
C:\Install_Dir\wildfly-8.2.0.Final\ServiceCatalogServer\configuration
C:\Install_Dir\wildfly-8.2.0.Final\ServiceLinkServer\configuration
C:\Install_Dir\bin
```

手順 2　**RequestCenter.war** の **rcjms.properties** および **ServiceLink.war** の **integration-server.properties** を編集します。

a. **Request Center** ファイルで、**rcjms.properties** セクションの次の変数を編集します。

– **http-remoting** という値をすべて **https-remoting** に置き換えます。

– ポート **6080** という値をすべてポート **6443** に置き換えます。

（注）　次の情報は、11.1 で SSL を有効にする場合は無視してください。

– 次のエントリを追加します。

```
## FOR SSL ##

BEEERequisitions.CLIENT_KEYSTORE=client.keystore
```

```
BEEERequisitions.CLIENT_TRUSTSTORE=client.truststore
BEEERequisitions.KEYSTORE_PASSWORD=secret
BEEERequisitions.TRUSTSTORE_PASSWORD=secret
BEEERequisitions.TRUSTSTORE_TYPE=JCEKS

## FOR SSL ###
```

b. **Service Link** ファイルで、**integration-server.properties** セクションの次の変数を編集します。
   - **http-remoting** という値をすべて **https-remoting** に置き換えます。
   - ポート **6080** という値をすべてポート **6443** に置き換えます。

（注）　次の情報は、11.1 で SSL を有効にする場合は無視してください。

   - 次のエントリを追加します。

```
## FOR SSL ##

ISEEOutbound.CLIENT_KEYSTORE=client.keystore
ISEEOutbound.CLIENT_TRUSTSTORE=client.truststore
ISEEOutbound.KEYSTORE_PASSWORD=secret
ISEEOutbound.TRUSTSTORE_PASSWORD=secret
ISEEOutbound.TRUSTSTORE_TYPE=JCEKS

## FOR SSL ###
```

手順 3　添付されている **https-users.properties** および **https-roles.properties** を次の場所にコピーします。

```
C:\Install_Dir\wildfly-8.2.0.Final\standalone\configuration
C:\Install_Dir\wildfly-8.2.0.Final\ServiceCatalogServer\configuration
C:\Install_Dir\wildfly-8.2.0.Final\ServiceLinkServer\configuration
```

手順 4　**standalone-full-RC.xml** を **Standalone-full_RC_SQL.xml** または **Standalone-full_RC_Oracle.xml** に、**standalone-full-SL.xml** を **Standalone-full_SL_SQL.xml** または **Standalone-full_SL_Oracle.xml** に置き換え、それらを次の場所に配置します。これらの xml ファイルはここからダウンロードします。

```
C:\Install_Dir\wildfly-8.2.0.Final\ServiceCatalogServer\configuration
C:\Install_Dir\wildfly-8.2.0.Final\ServiceLinkServer\configuration
```

（注）　インストールが Oracle である場合、データ ソースをそれに応じて変更する必要があります。

上記のファイルの名前を **standalone-full.xml** に変更し、**Service Link Oracle** に対し、次の操作を実行します。

a. **IP_ADDRESS** および **DB_NAME** を顧客が使用する IP アドレスおよびデータベースの名前に置き換えます。

```
<connection-url>jdbc:sqlserver://IP_ADDRESS:1433;DatabaseName=DB_NAME</connection-url>
```

b. **IP_ADDRESS** をアプリケーションがインストールされているマシンの IP アドレスに置き換えます。

```
<outbound-socket-binding name="my-http">
          <remote-destination host="IP_ADDRESS"port="${jboss.https.port:6443}"/>
</outbound-socket-binding>
```

手順 5　次のパスにあるモジュール ファイルに依存関係を追加します。
C:\Install_Dir\wildfly-8.2.0.Final\modules\system\layers\base\io\netty\main\module.xml

```
<dependencies>
     <module name="javax.api"/>
</dependencies>
```

手順 6　**newscale.properties** を編集し、**isee.base.url** 変数が **https://<ip-address>:6443** に設定されていることを確認します。

（注）　この手順 7 の使用が必要なのは、11.1 の場合のみです。

手順 7　次を **startServiceCatalog.conf.cmd/startServiceCatalog.conf.sh** にコピーします。

```
## For Windows START ##

set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.keyStore=client.keystore"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStore=client.truststore"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.keyStorePassword=secret"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStorePassword=secret"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStoreType=JCEKS"

## For Windows END ##

## For Linux START ##

JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStore=client.keystore"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStore=client.truststore"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStorePassword=secret"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStorePassword=secret"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStoreType=JCEKS"

## For Linux END ##
```

手順 8　サーバのログを消去して、Service Link、Request Center の順に再起動します。

カスタムのスタンドアロン モードでの SSL の有効化

手順 1　次のコマンドを入力し、カスタムのスタンドアロン用の秘密キーを作成します。

a. **RC** で次のコマンドを入力し、サーバの秘密キー（server1key）とクライアントの秘密キー（client1key）をそれぞれ作成します。

```
keytool -genkeypair -alias server1key -keyalg RSA -keysize 2048 -validity 7360
-keystore server1.keystore
keytool -genkeypair -alias client1key -keyalg RSA -keysize 2048 -validity 7360
-keystore client1.keystore
```

b. **SL** で次のコマンドを入力し、サーバの秘密キー（server2key）とクライアントの秘密キー（client2key）をそれぞれ作成します。

```
keytool -genkeypair -alias server2key -keyalg RSA -keysize 2048 -validity 7360
-keystore server2.keystore
keytool -genkeypair -alias client2key -keyalg RSA -keysize 2048 -validity 7360
-keystore client2.keystore
```

手順 2　次のコマンドを入力し、キーを証明書にエクスポートします。

a. RC で、

```
keytool -export -alias server1key -keystore server1.keystore -rfc -file server1.crt
keytool -export -alias client1key -keystore client1.keystore -rfc -file client1.crt
```

b. SL で、

```
keytool -export -alias server2key -keystore server2.keystore -rfc -file server2.crt
keytool -export -alias client2key -keystore client2.keystore -rfc -file client2.crt
```

手順 3　次のコマンドを入力し、公開キーを証明書からエクスポートし、信頼ストアにインポートします。

a. RC で、

server2.crt を SL から RC（C:\SSL）にコピーした後に、次のコマンドを入力します。

```
keytool -import -file server1.crt -keystore client1.truststore
keytool -import -file client1.crt -keystore server1.truststore
keytool -import -alias client1finaltrust -file server2.crt -keystore client1.truststore
```

b. SL で、

client1.crt を RC から SL（C:\SSL）にコピーした後に、次のコマンドを入力します。

```
keytool -import -file server2.crt -keystore client2.truststore
keytool -import -file client2.crt -keystore server2.truststore
keytool -import -alias server2finaltrust -file client1.crt -keystore server2.truststore
```

（注）　RC で、

- server1.keystore、server1.truststore、client1.keystore、client1.truststore を /INSTALL_DIR/wildfly-8.2.0.Final/ServiceCatalogServer/configuration にコピーします。
- server1.keystore、server1.truststore、client1.keystore、client1.truststore を RC の /INSTALL_DIR/bin にコピーします。

（注）　SL で、

- server2.keystore、server2.truststore、client2.keystore、および client2.truststore の各ファイルを /INSTALL_DIR/wildfly-8.2.0.Final/ServiceLinkServer/configuration にコピーします。
- server2.keystore、server2.truststore、client2.keystore、および client2.truststore の各ファイルを SL の /INSTALL_DIR/bin にコピーします。

手順 4　**RequestCenter.war**（VM1）の **rcjms.properties** および **ServiceLink.war**（VM2）の **integration-server.properties** を編集します。

a. **Request Center** ファイルで、**rcjms.properties** セクションの次の変数を編集します。
   - **http-remoting** という値をすべて **https-remoting** に置き換えます。
   - ポート **6080** という値をすべてポート **6443** に置き換えます。

（注）　次の情報は、11.1 で SSL を有効にする場合は無視してください。

– 次のエントリを追加します。

```
## FOR SSL ##

BEEERequisitions.CLIENT_KEYSTORE=client.keystore
BEEERequisitions.CLIENT_TRUSTSTORE=client.truststore
BEEERequisitions.KEYSTORE_PASSWORD=secret
BEEERequisitions.TRUSTSTORE_PASSWORD=secret
BEEERequisitions.TRUSTSTORE_TYPE=JCEKS

## FOR SSL for VM1###
```

b. **Service Link** ファイルで、**integration-server.properties** セクションの次の変数を編集します。

– **http-remoting** という値をすべて **https-remoting** に置き換えます。

– ポート **6080** という値をすべてポート **6443** に置き換えます。

（注）　次の情報は、11.1 で SSL を有効にする場合は無視してください。

– 次のエントリを追加します。

```
## FOR SSL for VM2##

ISEEOutbound.CLIENT_KEYSTORE=client.keystore
ISEEOutbound.CLIENT_TRUSTSTORE=client.truststore
ISEEOutbound.KEYSTORE_PASSWORD=secret
ISEEOutbound.TRUSTSTORE_PASSWORD=secret
ISEEOutbound.TRUSTSTORE_TYPE=JCEKS

## FOR SSL ###
```

手順 5　添付されている **https-users.properties** および **https-roles.properties** を次の場所にコピーします。

```
C:\Install_Dir\wildfly-8.2.0.Final\standalone\configuration
C:\Install_Dir\wildfly-8.2.0.Final\ServiceCatalogServer\configuration
C:\Install_Dir\wildfly-8.2.0.Final\ServiceLinkServer\configuration
```

手順 6　**standalone-full-RC.xml** を **Standalone-full_RC_SQL.xml** または **Standalone-full_RC_Oracle.xml** に、**standalone-full-SL.xml** を **Standalone-full_SL_SQL.xml** または **Standalone-full_SL_Oracle.xml** に置き換え、それらを次の場所に配置します。これらの xml ファイルはここからダウンロードします。

```
C:\Install_Dir\wildfly-8.2.0.Final\ServiceCatalogServer\configuration
C:\Install_Dir\wildfly-8.2.0.Final\ServiceLinkServer\configuration
```

（注）　インストールが Oracle である場合、データ ソースをそれに応じて変更する必要があります。

上記のファイルの名前を **standalone-full.xml** に変更し、**Service Link Oracle** に対し、次の操作を実行します。

a. **IP_ADDRESS** および **DB_NAME** を顧客が使用する IP アドレスおよびデータベースの名前に置き換えます。

```
<connection-url>jdbc:sqlserver://IP_ADDRESS:1433;DatabaseName=DB_NAME</connection-url>
```

b. **IP_ADDRESS** をアプリケーションがインストールされているマシンの IP アドレスに変更します。

```
<outbound-socket-binding name="my-http">
          <remote-destination host="IP_ADDRESS"port="${jboss.https.port:6443}"/>
</outbound-socket-binding>
```

手順 7　次のパスにあるモジュール ファイルに依存関係を追加します。
C:\Install_Dir\wildfly-8.2.0.Final\modules\system\layers\base\io\netty\main\module.xml

```
<dependencies>
     <module name="javax.api"/>
</dependencies>
```

(注)　手順 5 から手順 7 までを VM1 と VM2 の両方に対して実行します。

手順 8　VM1 の **newscale.properties** を編集し、**isee.base.url** 変数が **https://<ip-address>:6443** に設定されていることを確認します。

(注)　この手順 9 の使用が必要なのは、11.1 の場合のみです。

手順 9　次を **startServiceCatalog.conf.cmd/startServiceCatalog.conf.sh** にコピーします。

```
## For Windows START ##

set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.keyStore=client.keystore"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStore=client.truststore"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.keyStorePassword=secret"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStorePassword=secret"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStoreType=JCEKS"

## For Windows END ##

## For Linux START ##

JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStore=client.keystore"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStore=client.truststore"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStorePassword=secret"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStorePassword=secret"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStoreType=JCEKS"

## For Linux END ##
```

手順 10　サーバのログを消去して、Service Link、Request Center の順に再起動します。

2VM クラスタ トポロジでの SSL の有効化

手順 1　C:\SSL パスで作成を行い、ディレクトリに移動して次の手順に従います。
これにより、サーバ キー（秘密および公開）とクライアント キーのペアが作成されます。

(注)　秘密キーと公開キーを作成するために入力する情報が、クライアントとサーバの両方で一致していることを確認します。

(注)　キーの作成に使用されるデフォルトのパスワードは *secret* です。新しいパスワードを作成する場合は、*rcjms.properties*、*integration-server.properties*、および *domain.xml* ファイルにある *secret* という語を選択したパスワードに置き換えます。

（注）　名前と姓を要求されたら、ホスト コンピュータの IP アドレスを入力します。

a. 次のコマンドを入力し、サーバの秘密キー（serverkey）とクライアントの秘密キー（clientkey）をそれぞれ作成します。

VM1（RC および SL）で、

```
keytool -genkeypair -alias server1key -keyalg RSA -keysize 2048 -validity 7360 -keystore
server1.keystore
keytool -genkeypair -alias client1key -keyalg RSA -keysize 2048 -validity 7360 -keystore
client1.keystore
```

RC2 で、

```
keytool -genkeypair -alias server2key -keyalg RSA -keysize 2048 -validity 7360 -keystore
server2.keystore
keytool -genkeypair -alias client2key -keyalg RSA -keysize 2048 -validity 7360 -keystore
client2.keystore
```

b. 次のコマンドを入力し、キーを証明書にエクスポートします。

VM1（RC および SL）で、

```
keytool -export -alias server1key -keystore server1.keystore -rfc -file server1.crt
keytool -export -alias client1key -keystore client1.keystore -rfc -file client1.crt
```

RC2 で、

```
keytool -export -alias server2key -keystore server2.keystore -rfc -file server2.crt
keytool -export -alias client2key -keystore client2.keystore -rfc -file client2.crt
```

c. 次のコマンドを入力し、公開キーを証明書からエクスポートし、信頼ストアにインポートします。

VM1 で、

server2.crt と client2.crt を VM2 から VM1 にコピーします。

```
Copy the server2.keystore, client2.keystore and server2.truststore, client2.truststore to
/<INSTALL_DIR>/wildfly-8.2.0.Final
Copy the server2.keystore,client2.keystore and server2.truststore, client2.truststore to
/<INSTALL_DIR>/wildfly-8.2.0.Final/domain/configuration
```

VM1 で次のコマンドを入力します。

```
keytool -import -file server2.crt -keystore client1.truststore
keytool -import -file client2.crt -keystore server1.truststore
keytool -import -alias client1finaltrust -file server1.crt -keystore client1.truststore
keytool -import -alias server1finaltrust -file client1.crt -keystore server1.truststore
```

VM2 で、

server1.crt と client1.crt を VM1 から VM2 にコピーします。

```
Copy the server1.keystore, client1.keystore and server1.truststore, client1.truststore to
/<INSTALL_DIR>/wildfly-8.2.0.Final
Copy the server1.keystore, client1.keystore and server1.truststore to
/<INSTALL_DIR>/wildfly-8.2.0.Final/domain/configuration
```

VM2 で次のコマンドを入力します。

```
keytool -import -file server1.crt -keystore client2.truststore
keytool -import -file client1.crt -keystore server2.truststore
keytool -import -alias client2finaltrust -file server2.crt -keystore client2.truststore
keytool -import -alias server2finaltrust -file client2.crt -keystore server2.truststore
```

手順 2　**client.keystore** および **client.truststore** へのパスを **password** および **truststoretype** とともに入力します。

（注）　HC2 の場合、物理ファイル client2.keystore、client2.truststore、server2.keystore、および server2.truststore の名前を VM1 のものと同じ名前に変更します。これは、VM1 から VM2 に rcjms.properties を持つバイナリが展開されるためです。

VM1 および VM2 で、

startServiceCatalogCluster.conf.sh または startServiceCatalogCluster.conf.bat に次のコードを入力します。

```
rem # Properties for SSL over Wildfly
## For Windows START ##

set "JAVA_OPTS=%JAVA_OPTS% \-Djavax.net.ssl.keyStore=client1.keystore"
set "JAVA_OPTS=%JAVA_OPTS% \-Djavax.net.ssl.trustStore=client1.truststore"
set "JAVA_OPTS=%JAVA_OPTS% \-Djavax.net.ssl.keyStorePassword=secret"
set "JAVA_OPTS=%JAVA_OPTS% \-Djavax.net.ssl.trustStorePassword=secret"
set "JAVA_OPTS=%JAVA_OPTS% \-Djavax.net.ssl.trustStoreType=JCEKS"

## For Windows End##

## For Linux START ##

JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStore=client.keystore"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStore=client.truststore"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStorePassword=secret"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStorePassword=secret"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStoreType=JCEKS"

## For Linux END ##
```

手順 3　**/content/RequestCenter.war** および **/content/ISEE.war** に次の変更を加えます。
次に、**/INSTALL_DIR/dist** にある **RequestCenter.war** および **ISEE.war** を解凍し、war ファイルを再作成します。

a. **RC** については **rcjms.properties** に、**SL** については **integrationserver.properties** に、次の値を入力します。
   - **http-remoting** という値をすべて **https-remoting** に置き換えます。
   - ポート **6080** という値をすべてポート **6443** に置き換えます。

（注）　次の情報は、11.1 で SSL を有効にする場合は無視してください。

   - **rcjms.properties** および **integrationserver.properties** に次のエントリを追加します。

```
## FOR RC in rcjms.properties ###
BEEERequisitions.CLIENT_KEYSTORE=client1.keystore
BEEERequisitions.CLIENT_TRUSTSTORE=client1.truststore
BEEERequisitions.KEYSTORE_PASSWORD=secret
BEEERequisitions.TRUSTSTORE_PASSWORD=secret
```

```
BEEERequisitions.TRUSTSTORE_TYPE=JCEKS
## END OF FOR SSL ##

## FOR SL in integrationserver.properties ###
ISEEOutbound.CLIENT_KEYSTORE=client2.keystore
ISEEOutbound.CLIENT_TRUSTSTORE=client2.truststore
ISEEOutbound.KEYSTORE_PASSWORD=secret
ISEEOutbound.TRUSTSTORE_PASSWORD=secret
ISEEOutbound.TRUSTSTORE_TYPE=JCEKS
## END OF FOR SSL ##
```

b. **newscale.properties** を編集し、**isee.base.url** 変数が **https://<ip-address>:6443** に設定されていることを確認します。

手順 4　このページに添付されている **https-users.properties** および **https-roles.properties** をコピーするか、次の手順に従って独自に作成します。

a. 添付されている **https-users.properties** および **https-roles.properties** を次の場所に移動します。

JBOSS_HOME/domain/configuration

（注）　*JBOSS_HOME* は、ユーザによってクライアント VM に設定される変数です。

（注）　ユーザ名およびパスワードを新規作成するか既存のものを変更し、次の手順に従います。

b. 正しい **16 進数のパスワード**を、httpRealm で次のコマンドを使用して作成した **https-user.properties ファイル**に作成します。

```
<INSTALL_DIR\wildfly-8.2.0.Final\modules\system\layers\base\org\jboss\sasl\main>java
-classpath jboss-sasl-1.0.4.Final.jar org.jboss.sasl.util.UsernamePasswordHashUtil
jmsuser httpsRealm newscale

The above command will generate the below line:
jmsuser=b8b4553774410fded0a8c8f317217f44


<INSTALL_DIR\wildfly-8.2.0.Final\modules\system\layers\base\org\jboss\sasl\main>java
-classpath jboss-sasl-1.0.4.Final.jar org.jboss.sasl.util.UsernamePasswordHashUtil
adminuser httpsRealm admin123

The above command will generate the below line:
adminuser=ad61acb853db393c03846f3670b999b5


<INSTALL_DIR\wildfly-8.2.0.Final\modules\system\layers\base\org\jboss\sasl\main>java
-classpath jboss-sasl-1.0.4.Final.jar org.jboss.sasl.util.UsernamePasswordHashUtil
HOST2 httpsRealm HOST2
The above command will generate the below line:
HOST2=43e484d6af217b513fccf36c8b13cb27

<INSTALL_DIR>\wildfly-8.2.0.Final\modules\system\layers\base\org\jboss\sasl\main>java
-classpath jboss-sasl-1.0.4.Final.jar org.jboss.sasl.util.UsernamePasswordHashUtil
HOST1 httpsRealm HOST1HOST1=91e12144aab41d877d778aff1b1921bb
```

手順 5　**JMS クレデンシャル**、**管理者クレデンシャル**、**HOST1 のクレデンシャル**、および **HOST2 のクレデンシャル** を JBOSS_HOME/domain/configuration の **https-user.properties** にコピーします。

手順 6　**https-users.properties** および **https-roles.properties** を JBOSS_HOME/domain/configuration から次の場所にコピーします。

```
<INSTALL_DIR>/wildfly-8.2.0.Final/domain/servers/server-host1-RC/configuration
<INSTALL_DIR>/wildfly-8.2.0.Final/domain/servers/server-host1-SL/configuration
```

（注）　インストールが Oracle である場合、データ ソースをそれに応じて変更する必要があります。

手順 7　**domain.xml** および **hostva_backup.xml** または **hostva.xml** を、添付されている **VM1** の **2VM_SQL_domain.xml** または **4VM_Oracle_domain.xml** および **2VM_hostva_backup.xml** に置き換えます。また、VM2 内の **host2_backup.xml** または **host2.xml** を添付されている **2VM_host2_backup.xml** に置き換えます。これらの xml ファイルはここからダウンロードします。

a. **IP_ADDRESS** および **DB_NAME** を顧客が使用する IP アドレスおよびデータベースの名前に置き換えます。

```
<connection-url>jdbc:sqlserver://IP_ADDRESS:1433;DatabaseName=DB_NAME</connection-url>
```

b. **IP_ADDRESS** をアプリケーションがインストールされているマシンの IP アドレスに置き換えます。

```
<outbound-socket-binding name="my-http">
        <remote-destination host="IP_ADDRESS" port="${jboss.https.port:6443}"/>
</outbound-socket-binding>
```

手順 8　次のパスにあるモジュール ファイルに依存関係を追加します。**VM1 と VM2 両方の <INSTALL_DIR>\wildfly-8.2.0.Final\modules\system\layers\base\io\netty\main\module.xml**

```
<dependencies>
<module name="javax.api"/>
</dependencies>
```

手順 9　サーバのログを消去して、Service Link、Request Center の順に再起動します。

4VM クラスタ トポロジでの SSL の有効化

手順 1　C:\SSL パスで作成を行い、ディレクトリに移動して次の手順に従います。
これにより、サーバ キー（秘密および公開）とクライアント キーのペアが作成されます。

VM1（ドメイン コントローラ）について、

a. 次のコマンドを入力し、サーバの秘密キー（server.keystore）とクライアントの秘密キー（client.keystore）をそれぞれ作成します。

```
keytool -genkeypair -alias serverkey -keyalg RSA -keysize 2048 -validity 7360
-keystore server.keystore
keytool -genkeypair -alias clientkey -keyalg RSA -keysize 2048 -validity 7360
-keystore client.keystore
```

（注）　秘密キーと公開キーを作成するために入力する情報が、クライアントとサーバの両方で一致していることを確認します。

(注)　キーの作成に使用されるデフォルトのパスワードは *secret* です。新しいパスワードを作成する場合は、*rcjms.properties*、*integration-server.properties*、および *domain.xml* ファイルにある *secret* という語を選択したパスワードに置き換えます。

(注)　名前と姓を要求されたら、ホスト コンピュータの IP アドレスを入力します。

b. 次のコマンドを入力し、キーを証明書にエクスポートします。

```
keytool -export -alias serverkey -keystore server.keystore -rfc -file server.crt
keytool -export -alias clientkey -keystore client.keystore -rfc -file client.crt
```

VM2（ホスト コントローラ - HOST1）について、

a. 次のコマンドを入力し、サーバの秘密キー（server1.keystore）とクライアントの秘密キー（client1.keystore）をそれぞれ作成します。

```
keytool -genkeypair -alias server1key -keyalg RSA -keysize 2048 -validity 7360
-keystore server1.keystore
keytool -genkeypair -alias client1key -keyalg RSA -keysize 2048 -validity 7360
-keystore client1.keystore
```

(注)　秘密キーと公開キーを作成するために入力する情報が、クライアントとサーバの両方で一致していることを確認します。

(注)　キーの作成に使用されるデフォルトのパスワードは *secret* です。新しいパスワードを作成する場合は、*rcjms.properties*、*integration-server.properties*、および *domain.xml* ファイルにある *secret* という語を選択したパスワードに置き換えます。

(注)　名前と姓を要求されたら、ホスト コンピュータの IP アドレスを入力します。

b. 次のコマンドを入力し、キーを証明書にエクスポートします。

```
keytool -export -alias server1key -keystore server1.keystore -rfc -file server1.crt
keytool -export -alias client1key -keystore client1.keystore -rfc -file client1.crt
```

VM3（Service Link – スタンドアロン）について、

a. 次のコマンドを入力し、サーバの秘密キー（serverSL.keystore）とクライアントの秘密キー（clientSL.keystore）をそれぞれ作成します。

```
keytool -genkeypair -alias serverSLkey -keyalg RSA -keysize 2048 -validity 7360
-keystore serverSL.keystore
keytool -genkeypair -alias clientSLkey -keyalg RSA -keysize 2048 -validity 7360
-keystore clientSL.keystore
```

(注)　秘密キーと公開キーを作成するために入力する情報が、クライアントとサーバの両方で一致していることを確認します。

（注）　キーの作成に使用されるデフォルトのパスワードは *secret* です。新しいパスワードを作成する場合は、*rcjms.properties*、*integration-server.properties*、および *domain.xml* ファイルにある *secret* という語を選択したパスワードに置き換えます。

（注）　名前と姓を要求されたら、ホスト コンピュータの IP アドレスを入力します。

b. 次のコマンドを入力し、キーを証明書（serverSL.crt および clientSL.crt）にエクスポートします。

```
keytool -export -alias serverSLkey -keystore serverSL.keystore -rfc -file serverSL.crt
keytool -export -alias clientSLkey -keystore clientSL.keystore -rfc -file clientSL.crt
```

手順 2　全 VM のキーを生成し、次のコマンドを入力して、公開キーを証明書からエクスポートし、信頼ストアにインポートします。

a. server.crt と client.crt を VM1 から VM2 にコピーします。VM2 で次のコマンドを入力します。

```
keytool -import -alias serverTrustclient1 -file server.crt -keystore
client1.truststore
keytool -import -alias clientTrustserver1 -file client.crt -keystore
server1.truststore
```

b. serverSL.crt と clientSL.crt を VM3 から VM2 にコピーします。VM2 で次のコマンドを入力します。

```
keytool -import -alias serverSLTrustclient1 -file serverSL.crt -keystore
client1.truststore
keytool -import -alias clientSLTrustserver1 -file clientSL.crt -keystore
server1.truststore
```

c. server1.crt と client1.crt を VM2 から VM1 にコピーします。VM1 で次のコマンドを入力します。

```
keytool -import -alias server1Trustclient -file server1.crt -keystore
client.truststore
keytool -import -alias client1Trustserver -file client1.crt -keystore
server.truststore
```

d. server1.crt と client1.crt を VM2 から VM3 にコピーします。VM3 で次のコマンドを入力します。

```
keytool -import -alias server1TrustclientSL -file server1.crt -keystore
clientSL.truststore
keytool -import -alias client1TrustserverSL -file client1.crt -keystore
serverSL.truststore
```

e. serverSL.crt および clientSL.crt が VM3 にあることを確認します。VM3 で次のコマンドを入力します。

```
keytool -import -alias serverSLTrustclientSL -file serverSL.crt -keystore
clientSL.truststore
keytool -import -alias clientSLTrustserverSL -file clientSL.crt -keystore
serverSL.truststore
```

手順 3　キーストア ファイルおよび信頼ストア ファイルを以下の場所に配置します。

a. VM1 と VM2 の信頼ストアおよびキーストアを同じマシンの次の場所にコピーします。

```
C:\Install_Dir
C:\Install_Dir\bin
C:\Install_Dir\wildfly-8.2.0.Final
C:\Install_Dir\wildfly-8.2.0.Final\domain\configuration
C:\Install_Dir\wildfly-8.2.0.Final\domain\servers\configuration
```

b. VM3 の信頼ストアおよびキーストアを次の場所にコピーします。

```
C:\Install_Dir
C:\Install_Dir\bin
C:\Install_Dir\wildfly-8.2.0.Final
C:\Install_Dir\wildfly-8.2.0.Final\ServiceLinkServer\configuration
C:\Install_Dir\wildfly-8.2.0.Final\standalone\configuration
```

手順 4　起動スクリプトの変更:

a. **VM1** の **startServiceCatalogCluster.conf.cmd** または **startServiceCatalogCluster.conf.sh** に次のスニペットを追加します。

```
## For Windows START ##

set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.keyStore=client.keystore"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStore=client.truststore"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.keyStorePassword=secret"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStorePassword=secret"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStoreType=JCEKS"

## For Windows END ##


## For Linux START ##

JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStore=client.keystore"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStore=client.truststore"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStorePassword=secret"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStorePassword=secret"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStoreType=JCEKS"

## For Linux END ##
```

b. **VM2** の **startServiceCatalogCluster.conf.cmd** または **startServiceCatalogCluster.conf.sh** に次をコピーします。

```
## For Windows START ##

set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.keyStore=client1.keystore"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStore=client1.truststore"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.keyStorePassword=secret"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStorePassword=secret"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStoreType=JCEKS"

## For Windows END ##


## For Linux START ##

JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStore=client1.keystore"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStore=client1.truststore"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStorePassword=secret"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStorePassword=secret"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStoreType=JCEKS"

## For Linux END ##
```

c. **VM3** の **startServiceLink.conf.cmd** または **startServiceLink.conf.sh** に次をコピーします。

```
## For Windows START ##
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.keyStore=clientSL.keystore"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStore=clientSL.truststore"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.keyStorePassword=secret"
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStorePassword=secret"
```

```
set "JAVA_OPTS=%JAVA_OPTS% -Djavax.net.ssl.trustStoreType=JCEKS"
## For Windows END ##


## For Linux START ##
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStore=clientSL.keystore"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStore=clientSL.truststore"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.keyStorePassword=secret"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStorePassword=secret"
JAVA_OPTS="$JAVA_OPTS -Djavax.net.ssl.trustStoreType=JCEKS"

## For Linux END ##
```

手順 5　全 VM の **C:\4VM_SSL_Domain\wildfly-8.2.0.Final\bin\jboss-cli.xml** に次の変更を加えます。

（注）　次のコード スニペット内のタグ IP_ADDRESS を各 VM の IP アドレスに変更します。

```
<default-controller>
   <protocol>https-remoting</protocol>
   <host>IP_ADDRESS</host>
   <port>9993</port>
</default-controller>
```

手順 6　添付されている 4VM-https-roles.properties および 4VM-https-users.properties を（それぞれ **https-roles.properties** および **https-users.properties** として）、VM1 および VM2 の **C:\4VM_SSL_Domain\wildfly-8.2.0.Final\domain\configuration** にコピーします。

手順 7　設定ファイルの変更:

a. 添付されている **4VM_SQL_domain.xml または 4VM_Oracle_domain.xml** を（**domain.xml** として）VM1 にコピーします。

- **<outbound-socket-binding name="remote-http">** セクションで、**IP アドレス**（デフォルトでは 10.76.82.36）を VM1 の IP アドレスに変更します。
- **<interfaces>** セクションで、**IP アドレス**（デフォルトでは 10.76.82.36）を VM1 の IP アドレスに置き換えます。
- **<datasource jndi-name="java:/REQUESTCENTERDS" pool-name="REQUESTCENTERDS" enabled="true">** セクションで、**IP アドレス**（デフォルトでは 10.76.81.198）とデータベースの名前を、（データベース サーバがある）マシンの IP アドレスと現在使用されているデータベースの名前に置き換えます。

b. 添付されている 4VM_host_default.xml を（**host_default.xml** として）VM1 にコピーします。

- **<interface name="unsecure">** セクションで、**IP アドレス**（デフォルトでは 10.76.82.36）を VM1 の IP アドレスに置き換えます。

c. VM2 で、添付されている 4VM_host1_backup.xml を（**host1.xml または host1_backup.xml** として）コピーし、次の変更を加えます。

- **<interfaces>** セクションで、**IP アドレス**（デフォルトでは 10.76.82.38）を VM2 の IP アドレスに置き換えます。
- **<domain-controller>** セクションで、**IP アドレス**（デフォルトでは 10.76.82.36）を VM1 の IP アドレスに置き換えます。

d. VM3 で、添付されている 4VM-standalone-SL.xml を（**standalone-full.xml** として）コピーし、以下の変更を行います。

- **<outbound-socket-binding name="my-http">** セクションで、**IP アドレス**（デフォルトでは 10.76.82.37）を VM3 の IP アドレスに置き換えます。

手順 8　WAR ファイルの変更:

a. **VM2** で、展開されているバージョンの **RequestCenter.war** に次の変更を加えます。

- **Request Center** ファイルで、**rcjms.properties** セクションの次の変数を編集します。
  - **http-remoting** という値をすべて **https-remoting** に置き換えます。
  - ポート **6080** という値をすべてポート **6443** に置き換えます。

（注）　次の情報は、11.1 で SSL を有効にする場合は無視してください。

  - 次のコードを追加します。

```
BEEERequisitions.CLIENT_KEYSTORE=client1.keystore
BEEERequisitions.CLIENT_TRUSTSTORE=client1.truststore
BEEERequisitions.KEYSTORE_PASSWORD=secret
BEEERequisitions.TRUSTSTORE_PASSWORD=secret
BEEERequisitions.TRUSTSTORE_TYPE=JCEKS
```

IP-ADDRESS を **VM3** の IP アドレスに、**newscale.properties** 内の**ポート 6443** を次のキーと値のペアに置き換えます。**isee.base.url=https://<IP-ADDRESS>:6443**

b. VM3 の展開ディレクトリにある **ServiceLink.war** に次の変更を加えます。

- **Service Link** ファイルで、**integration-server.properties** セクションの次の変数を編集します。
  - **http-remoting** という値をすべて **https-remoting** に置き換えます。
  - ポート **6080** という値をすべてポート **6443** に置き換えます。

c. **VM1** に次の変更を加えます。

- **INSTALL_DIR/dist** フォルダ内の **RequestCenter.war** および **ISEE.war** を解凍する必要があります。
  Request Center ファイルで、**rcjms.properties** セクションの次の変数を編集します。
  - **http-remoting** という値をすべて **https-remoting** に置き換えます。
  - ポート **6080** という値をすべてポート **6443** に置き換えます。

（注）　次の情報は、11.1 で SSL を有効にする場合は無視してください。

  - 次のコードを追加します。

```
BEEERequisitions.CLIENT_KEYSTORE=client1.keystore
BEEERequisitions.CLIENT_TRUSTSTORE=client1.truststore
BEEERequisitions.KEYSTORE_PASSWORD=secret
BEEERequisitions.TRUSTSTORE_PASSWORD=secret
BEEERequisitions.TRUSTSTORE_TYPE=JCEKS
```

（注）　CLIENT_KEYSTORE および CLIENT_TRUSTSTORE については、VM1 のキーを参照してください。

  - **IP-ADDRESS** を **VM3** の IP アドレスに、newscale.properties 内のポート **6443** を次のキーと値のペアに置き換えます。**isee.base.url=https://<IP-ADDRESS>:6443**
  - 上述の変更が完了したら、RequestCenter.war を再度圧縮し、展開スクリプトを実行します。展開スクリプトを実行する際に、ユーザは、証明書を永久的に受け入れるか、一時的に受け入れるかを選択できます。

手順 9　次のパスにあるモジュール ファイルに以下の依存関係を追加します。全 VM の **C:\Install_Dir\wildfly-8.2.0.Final\modules\system\layers\base\io\netty\main\module.xml**

```
<dependencies>
     <module name="javax.api"/>
</dependencies>
```

手順 10　**Linux 環境**で **4VM クラスタ** をセットアップするには、**jboss-cli.sh** を実行し、証明書を永久的に受け入れます。

11.1 Fix でのクラスタのセットアップ

この項では、11.1 fix のみのクラスタ セットアップの情報を提供します。

(注)　Linux システムの場合、端末でコマンドを実行する際は、バック スラッシュ (\) をスラッシュ (/) に置き換えてください。

(注)　次の表記を使用します。VM1 = DC、VM2 = SL、VM3 = HC1、VM4 = HC2。

2VM でのクラスタのセットアップ

手順 1　VM1 の Internet Explorer ブラウザで、VM1 の **RequestCenter** URL を入力します。

(注)　クラスタ内で 2 つの VM サーバが起動していることを確認します。

手順 2　表示されたウィンドウから [このサイトの閲覧を続行する (推奨されません)。(Continue to this website (not recommended))] を選択します。

手順 3　証明書情報ウィンドウの下部のペインで [証明書(Certificates)] をクリックします。

手順 4　表示されたウィンドウの [詳細(Details)] タブを選択して、[ファイルへコピー(Copy to File)] をクリックします。

手順 5　[証明書のエクスポート ウィザード(Certificate Export Wizard)] で [次へ(Next)] をクリックします。

手順 6　[証明書のエクスポート ウィザード(Certificate Export Wizard)] で **DER encoded binary X.509.cer** を選択します。

手順 7　[コピー(Copy)] をクリックして、たとえば次のような証明書をエクスポートするパスを参照します。

```
c:\root.cer
```

手順 8　VM1 で、**JAVA_LOCATION\jdk\jre\bin** ディレクトリから次のコマンドを入力し、cacerts ファイル内のエントリを決定します。

```
C:\Java\jdk1.7.0_71\jre\bin>keytool -list -keystore ..\lib\security\cacerts
```

手順 9　キーストアのパスワードとして **changeit** を入力します。

（注）　cacerts ファイル内のエントリの数と cacerts ファイルのサイズを書き留めておいてください。

手順 10　次のコマンドを実行して、サーバ証明書を追加します。

（注）　Java ランタイム環境（JRE）は、サーバ証明書がキーストアに追加されるまで、その存在を認識しません。

```
keytool -import -file C:/root.cer -keystore C:\Java\jdk1.7.0_71\jre\lib\security\cacerts
```

手順 11　キーストアのパスワードとして **changeit** を入力します。

手順 12　**bin** ディレクトリからの次のコマンドを入力し、cacerts ファイル内のエントリを決定します。

```
C:\Java\jdk1.7.0_71\jre\bin>keytool -list -keystore ..\lib\security\cacerts
```

手順 13　キーストアのパスワードとして **changeit** を入力します。

（注）　cacerts 内のエントリの数と cacerts ファイルのサイズを確認します。エントリの数は 1 増えており、cacerts ファイルのサイズは 1 KB 増えている必要があります。

これにより、プライベート ルート証明書が信頼された認証局としてエクストラネット サーバの cacerts キーストアに追加されたことが確認できます。

手順 14　2VM クラスタについて、

VM1 から VM2 に証明書をコピーし、同じ手順を繰り返します。次に例を示します。

```
VM1 = DC+HC1 and VM2 = HC2
```

手順 15　サーバを再起動し、クラスタ操作を実行します。

4VM でのクラスタのセットアップ

前提条件

- 使用する JDK のデフォルトのバージョンは、1.7.0_79 です。
- serverSL.crt および clientSL.crt は、SSL モードの 4VM クラスタでスタンドアロンの Service Link をセットアップする際に生成される証明書です。

4VM でクラスタをセットアップするには、次の手順に従います。VM2 で生成される serverSL.crt と clientSL.crt は、次の手順で cacerts ファイルに追加されます。

手順 1　コマンド プロンプトで、現在のディレクトリを指す java の bin フォルダで次のコマンドを入力します。

（注）　Linux の場合は、*Terminal* を使用する必要があります。

```
cd C:\Java\jdk1.7.0_79\jre\bin
```

手順 2　次のコマンドを入力し、cacerts ファイル内のエントリを決定します。
要求された場合、キーストアのパスワードとして **changeit** を入力します。

（注）　検証のため、ファイルのサイズを書き留めておきます。

```
keytool -list -keystore ..\lib\security\cacerts
```

手順 3　次のコマンドを入力し、**clientSL.crt** を cacerts ファイルに追加します。

（注）　.crt ファイルは C:\SSL ディレクトリなどにあります。

```
keytool -import -alias clientSL -file C:\SSL\clientSL.crt -keystore
C:\Java\jdk1.7.0_79\jre\lib\security\cacerts
```

手順 4　次のコマンドを入力し、**serverSL.crt** を cacerts ファイルに追加します。

（注）　.crt ファイルは C:\SSL ディレクトリなどにあります。

```
keytool -import -alias serverSL -file C:\SSL\serverSL.crt -keystore
C:\Java\jdk1.7.0_79\jre\lib\security\cacerts
```

手順 5　次のコマンドを入力し、cacerts ファイル内のエントリを決定します。
要求された場合、キーストアのパスワードとして **changeit** を入力します。

```
keytool -list -keystore ..\lib\security\cacerts
```

（注）　cacerts ファイル内のエントリの数が 2 つ増えていることを確認します。また、cacerts ファイルのサイズは 2 KB 増加しています。

手順 6　**変更された cacerts** ファイルを VM2 からコピーし、VM1、VM3、VM4 のそれぞれ該当する場所にある cacerts ファイルと置き換えます。

（注）　変更された cacerts ファイルに置き換える前に、必要に応じ、cacerts ファイルのバックアップを取ります。

手順 7　すべてのサーバを再起動し、クラスタ スクリプトを実行してクラスタを起動します。

2VM クラスタ上の SSL が有効な Wildfly アプリケーション サーバに接続するための SSL が有効な Apache Httpd

手順 1　**2VM の SSL** が有効化されたクラスタをセットアップします。「2VM でのクラスタのセットアップ」*を参照してください。*

手順 2　host1 と host2 でサーバを起動し、展開プロセスを完了します。

手順 3　WildFly クラスタとは別の VM に **Apache httpd バージョン 2.4.18** をダウンロードおよびインストールします。*次の Web サイトを参照してください。*
https://www.apachehaus.com/cgi-bin/download.plx?dli=gWy82MONVWy0kej9SWYZFbJVlUGRVYSZ1YxIUN

- C:\Apache24\bin>httpd -version
- Server version: Apache/2.4.18 (Win32)
- Server built: Jan 28 2016 09:58:25

手順 4　Windows 上の **Apache** で SSL を有効化します。

a. 必要なもの:
- SSL サポートを含む Apache のコピー。
- OpenSSL のコピー。
- openssl.cnf ファイル。

(注)　ダウンロードされた Apache には SSL サポートがあるので、その他のモジュールを別途ダウンロードしないでください。

b. openSSL の **openssl.cnf** ファイルをディレクトリ **C:\Apache\bin** にコピーします。

手順 5　自己署名証明書を作成します。

(注)　証明書のために作成される各種のファイルは、同じ名前に異なる拡張子が付いたものになります。

(注)　次のコマンド例で使用されている *bob* という名前は、必要に応じて置き換えてください。

a. 新しい証明書を作成するには、コマンド プロンプトで OpenSSL を含むディレクトリ（たとえば *C:\Apache\bin*）に切り替えた後、次のコマンドを入力します。

```
openssl req -config openssl.cnf -new -out bob.csr -keyout bob.pem
```

b. 多くの質問に答えるよう要求されますが、次以外は空白のままでかまいません。
- PEM パスフレーズ:生成する秘密キー（bob.pem）に関連付けられるパスワードです。次の手順でのみ使用されます。お好きなように設定してください。
- 共通名:この証明書に関連付けられる完全修飾ドメイン名です。この例では、IP アドレスを使用しています。

コマンドが完了すると、フォルダには **bob.csr** および **bob.pem** という 2 つのファイルが作成されます。

c. Apache が使用するためのパスワード保護のないキーを作成します。

```
openssl rsa -in bob.pem -out bob.key
```

上記で作成したパスワードを要求されます。その後、フォルダには **bob.key** というファイルが作成されます。

d. 次のコマンドを入力し、Apache にも必要な **X.509** 証明書を作成します。

```
openssl x509 -in bob.csr -out bob.cert -req -signkey bob.key -days 365
```

e. Apache が SSL を有効にするために使用する自己署名証明書が作成されます。ファイルは、たとえば次のようなパスに追加されています。
   - C:\Apache24\bin
   - C:\Apache24\conf
   - C:\Apache24\conf\ssl.

手順 6　ローカルにある **C:\Apache24\conf\httpd.conf** および **C:\Apache24\conf\extra\httpd-ssl.conf** を、このページに添付されている対応するファイルに置き換えます。

（注）　*bob* という名前は、証明書/キーに与えた名前に変更できます。

手順 7　**httpd.conf** と **httpd-ssl.conf** の両方で、**IP アドレス**をお使いの環境に応じて置き換えます。

手順 8　次のコマンドを入力し、Apache Httpd を起動します。

```
C:\Apache24\bin>httpd -k start
```

手順 9　次の URL でテストします。https://<ip>/RequestCenter

手順 10　Apache httpd Web サーバを停止するには、次を入力します。

```
C:\Apache24\bin>httpd -k stop
```

手順 11　Linux 環境では、添付で追加されたファイルを置き換えます。

スクリプトの変更

スタンドアロン モード（標準およびカスタム）の Wildfly アプリケーション サーバ上で SSL を有効化したら、動作が可能になるよう、次の変更を加えます。

手順 1　パス **C:\Installation Directory\wildfly-8.2.0.Final\ServiceCatalogServer\configuration\standalone-full.xml** にある **standalone-full.xml** の次のコード スニペットを置き換えます。

```
<management-interfaces>
     <http-interface security-realm="httpsRealm">
               <socket interface="management" port="9990" secure-port="9993"/>
     </http-interface>
</management-interfaces>
```

置き換え後:

```
<management-interfaces>
     <http-interface security-realm="ManagementRealm" http-upgrade-enabled="true">
               <socket-binding http="management-https"/>
     </http-interface>
</management-interfaces>
```

手順 2　パス **C:\Installation Directory\wildfly-8.2.0.Final\ServiceLinkServer\configuration\standalone-full.xml**. にある **standalone-full.xml** の次のコード スニペットを置き換えます。

```
<management-interfaces>
     <http-interface security-realm="httpsRealm">
               <socket interface="management" port="7990" secure-port="7443"/>
     </http-interface>
</management-interfaces>
```

置き換え後:

```
<management-interfaces>
     <http-interface security-realm="ManagementRealm" http-upgrade-enabled="true">
             <socket-binding http="management-https"/>
     </http-interface>
</management-interfaces>
```

手順 3　**stopServiceCatalog.cmd** または **stopServiceCatalog.sh** で、**CONTROLLER_PORT** 変数を **9993** に設定します。

手順 4　**stopServiceLink.cmd** または **stopServiceLink.sh** で、**CONTROLLER_PORT** 変数を **7443** に設定します。