

How to install CloudGateway Application

- Purpose
- Environment
- Installing Shibboleth SP
- Installing Shibboleth IdP
- Installing and Setting Up MySQL
 - Setup /etc/my.cnf
 - Execute
 - Create Database
- Installing and Setting Up PHP
 - Setting Up Timezone
- Installing and Setting Up Mail Server
 - Install
 - Setup Postfix
 - Execute
- Installing Attribute Provider (SP)
 - Download required files from following URL.
 - Modify Setting
 - Copy Files
 - Metadata Deployment
 - Restart
- Installing Attribute Provider (IdP)
 - Download required file from following URL.
 - Modify Configuration
 - Overwrite the configuration files.
 - Metadata Deployment
 - Deployment of the MariaDB driver
 - Configuration for StoredID
 - Deployment of the trustany-ssl
 - Back Channel
 - Back Channel Port
 - Deployment
- Installing the Application
 - Download files from the following URL.
 - Deployment
 - Initial Setting
 - Configuration of httpd.conf
 - Restart
 - Database Configuration
 - Application Configuration File
 - Registration of organizations
 - Registration of SP administrators
 - Registration of SP
 - Registration of IdP groups
 - Registration of SP connectors
 - Notify the administrator via mail
- Validation
 - Authentication
 - SP
- Advanced Configuration
 - Clustering
 - Application Configuration File
 - Create sessions table
 - Shibboleth Configuration File
 - Add the SP manually
 - Register the SP which is not a member of the federation
 - Create SP Connector
 - Connection between SP Connector and SP.
 - Automatic Connection of SP Connector
 - Attribute Consent Setting for SP Connector
 - Importing the Account Data from the Existing Database

ファイル

変更日

PNG ファイル figure3-1.png	2015/01/05 by yamaji@nii.ac.jp
Microsoft Word 97 文書 local-map-install-manual.doc	2015/01/05 by yamaji@nii.ac.jp
Microsoft Powerpoint プレゼンテーション map-install-manual-figures.pptx	2015/01/05 by yamaji@nii.ac.jp

JavaScript ファイル embedded-wayf_config.js	2015/02/04 by Takeshi Nishimura
XML ファイル attribute-filter.xml	2015/02/04 by Takeshi Nishimura
ZIP アーカイブ map.zip	2015/02/05 by Takeshi Nishimura
ZIP アーカイブ cloudgateway-r1293.zip	2017/10/26 by Takeshi Nishimura
ZIP アーカイブ cloudgateway-api-r1293.zip	2017/10/26 by Takeshi Nishimura
ZIP アーカイブ cloudgateway-r1319.zip	2018/02/13 by Soshi Urakami
ZIP アーカイブ cloudgateway-api-r1319.zip	2018/02/13 by Soshi Urakami
XML ファイル attribute-resolver.xml	2018/06/11 by Soshi Urakami
PNG ファイル CGcompontns.png	2021/10/14 by Takeshi Nishimura
ZIP アーカイブ cloudgateway-r1662.zip	2022/03/30 by Toru Yamaha
ZIP アーカイブ cloudgateway-api-r1662.zip	2022/03/30 by Toru Yamaha
ZIP アーカイブ cloudgateway-r1694.zip	2022/10/12 by Yuta Sato
ZIP アーカイブ cloudgateway-api-r1694.zip	2022/10/12 by Yuta Sato
ZIP アーカイブ cloudgateway-r1725.zip	2023/03/10 by Soshi Urakami
ZIP アーカイブ cloudgateway-api-r1751.zip	2023/09/29 by Sota Kaneko
ZIP アーカイブ cloudgateway-r1751.zip	2023/09/29 by Sota Kaneko

[すべてダウンロード](#)

Purpose

The purpose of this document is to install and validate the (global) mAP environment.

Environment

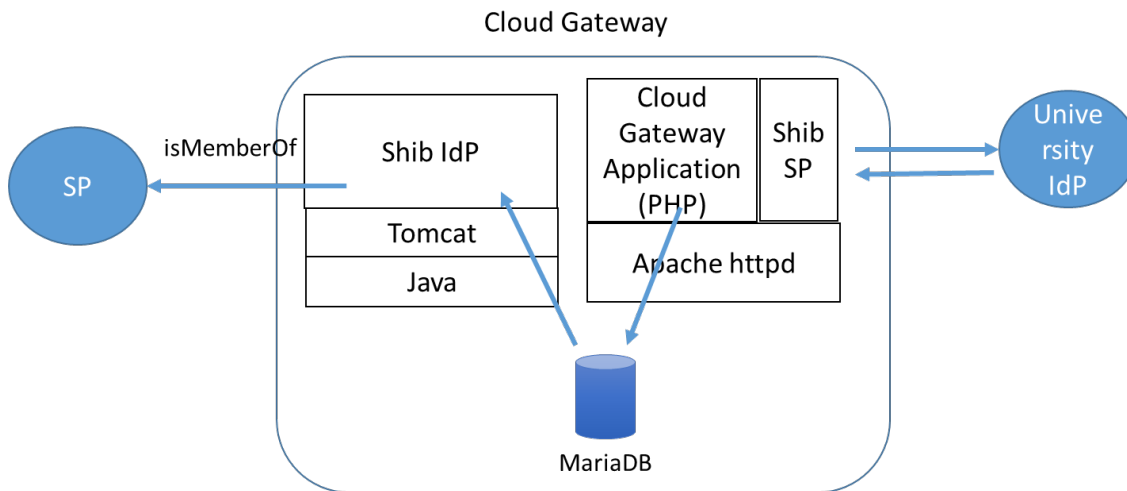
Following environment is required for installing mAP.

Table 2-1 Required Software

Software Name	Version	Notes
CentOS (64bit)	7.9	Operating System
Shibboleth-SP	3.4.1	Service Provider
Shibboleth-IdP	4.1.7	Identity Provider
Apache HTTP Server	2.4 (*1)	WEB Server
Java	OpenJDK 11.0.19 (*1)	IdP Executable Environment
Jetty	9.4.51	Servlet Container
MySQL	5.7.38	Relational Database
Postfix	2.10 (*1)	Mail Transfer Agent
PHP	5.4.16 (*1)	Programing Language

*1 Latest version of yum package as of 2023/05/18

These software will be used in the following structure:



Installing Shibboleth SP

Please install Shibboleth SP while referring to the install guide.

Shibboleth SP Install Guide

<https://meatwiki.nii.ac.jp/confluence/display/GakuNinShibInstallen/SP+Install>

In addition to Shibboleth SP, Apache, mod_ssl and NTP should be installed simultaneously. Please setup these environment as well.

Installing Shibboleth IdP

Please install Shibboleth IdP while referring to the install guide.

Shibboleth IdP Install Guide

<https://meatwiki.nii.ac.jp/confluence/display/GakuNinShibInstallen/IdP+Install+Manual>

In addition to Shibboleth IdP, Java and Jetty should be installed as well. Please setup these environment as well.

Installing and Setting Up MySQL

Install MariaDB by using following command.

```
$ sudo yum localinstall http://dev.mysql.com/get/mysql57-community-release-el7-7.noarch.rpm
$ sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022
$ sudo yum install mysql-community-server
```

* If the MySQL has already been installed as the initial component, you can skip this process.

Setup /etc/my.cnf

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
user=mysql
symbolic-links=0

old_passwords=1
character-set-server=utf8

[mysqld_safe] log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
[mysql]
default-character-set=utf8
```

Execute

```
$ sudo systemctl start mysqld.service
$ sudo systemctl enable mysqld.service
```

Create Database

```
$ mysql -u root
CREATE DATABASE vo;
GRANT INSERT, SELECT, UPDATE, DELETE ON vo.* TO 'vouser'@'localhost' IDENTIFIED BY 'YOUR OWN PASSWORD';
```

Installing and Setting Up PHP

Please install PHP by using following command.

```
$ sudo yum install php
$ sudo yum install php-devel php-gd php-mbstring php-pdo
$ sudo yum install php-mysql php-xml
$ sudo systemctl restart httpd.service
```

Setting Up Timezone

Add timezone in /etc/php.ini . Parameter “Asia/Tokyo” should be changed depending on your local time.

```
date.timezone = "Asia/Tokyo"
```

Installing and Setting Up Mail Server

Please install and setup mail server by using Postfix or Sendmail. Following instruction is for Postfix.

Install

```
$ sudo yum install postfix
```

Setup Postfix

Please modify parameters depending on your local environment.

```
/etc/postfix/main.cf
```

Execute

```
$ sudo systemctl start postfix.service  
$ sudo systemctl enable postfix.service
```

Installing Attribute Provider (SP)

Shibboleth SP which has been installed in section 4 will be configured for Attribute Provider.

Download required files from following URL.

Please contact GakuNin Office if authentication is requested.

[index.zip](#)
[embedded-wayf_config.js](#)

Modify Setting

- attribute-map.xml
Add the following line or make sure isMemberOf attribute is recognized.

```
<Attribute name="urn:oid:1.3.6.1.4.1.5923.1.5.1.1" id="isMemberOf"/>
```

- attribute-policy.xml
Add the following rule before attributeID="*" line.

```
<!-- isMemberOf -->  
<afp:AttributeRule attributeID="isMemberOf">  
  <afp:PermitValueRule xsi:type="AttributeIssuerString"  
    value="https://[Host Name of the SP]/idp/shibboleth"/>  
</afp:AttributeRule>
```

Modify "[Host Name of the SP]" to this server host name.

- shibboleth2.xml
Add MetadataProvider.

```
<!-- Example of locally maintained metadata. -->  
<!-- Metadata of this IdP -->  
<MetadataProvider type="XML" file="/opt/shibboleth-idp/metadata/idp-metadata.xml"/>
```

And add SimpleAggregation AttributeResolver after <AttributeResolver type="Query" subjectMatch="true"/>line.

```
<!-- Uses eduPersonPrincipalName from IdP to query, and asks for isMemberOf. -->  
<AttributeResolver type="SimpleAggregation" attributeId="eppn" format="urn:oid:1.3.6.1.4.1.5923.1.1.1.6">  
  <Entity>https://[Host Name of the IdP]/idp/shibboleth</Entity>  
</AttributeResolver>
```

Modify "[Host Name of the IdP]" to this server host name.

- embedded-wayf_config.js
Modify "Host Name of the SP" to the name of this server.
Values in wayf_additional_idps also have to be changed.
Instead, you can use your own embedded DS in app/views/pages/home.ctp.

Copy Files

```
$ sudo mkdir /var/www/html/secure
$ unzip index.zip
$ sudo cp index.php /var/www/html/secure/attr.php
$ sudo mkdir /var/www/html/js
$ sudo cp embedded-wayf_config.js /var/www/html/js/.
```

Metadata Deployment

- Store the metadata of the SP of this server in the following location.
/etc/shibboleth/metadata/sp-metadata.xml
- Store the metadata of this IdP of this server in the following location.
/opt/shibboleth-idp/metadata/idp-metadata.xml
* shibd have to be restarted after the “10. Installing Attribute Provider (IdP)” since the metadata of the IdP is not yet generated at this moment.
- Federation Metadata
If required, federation metadata have to be included by the shibboleth2.xml configuration.

Restart

```
$ sudo systemctl restart shibd.service
$ sudo systemctl restart httpd.service
```

Installing Attribute Provider (IdP)

Shibboleth IdP which has been installed in section 5 will be configured for Attribute Provider.

Download required file from following URL.

Please contact GakuNin Office if authentication is required.

[attribute-resolver.xml](#)
[attribute-filter.xml](#)

Please download mariadb-java-client-1.3.x.jar from MariaDB site:

https://mariadb.com/my_portal/download/java-client

Please download trustany-ssl-1.0.x.jar from wiki.shibboleth.net:

<https://wiki.shibboleth.net/confluence/display/IDP30/ApacheTomcat8#ApacheTomcat8-SupportingSOAPEndpoints>

Modify Configuration

- attribute-resolver.xml
Modify “SALT” to appropriate random values.
Setup database password to be the same one with “6. Installing and Setting Up MariaDB.
Modify “Host name of the IdP” to this server host name.
- relying-party.xml
Add MetadataProvider for SP which utilize this mAP system.

```
<metadata:MetadataProvider id="SP" xsi:type="metadata:ResourceBackedMetadataProvider">
  <metadata:MetadataResource xsi:type="resource:FileSystemResource" file="/etc/shibboleth/metadata/sp-metadata.xml" />
</metadata:MetadataProvider>
```

Overwrite the configuration files.

```

Back Up First
$ cd /opt/shibboleth-idp/conf
$ sudo cp attribute-filter.xml attribute-filter.xml.bk
$ sudo cp attribute-resolver.xml attribute-resolver.xml.bk

Then Copy
$ cd
$ sudo cp attribute-filter.xml /opt/shibboleth-idp/conf/.
$ sudo cp attribute-resolver.xml /opt/shibboleth-idp/conf/.

```

Metadata Deployment

- The metadata of this SP has already been stored in the following location.
/etc/shibboleth/metadata/sp-metadata.xml
- Store the metadata of this IdP in the following location.
/opt/shibboleth-idp/metadata/idp-metadata.xml

Deployment of the MariaDB driver

```

$ sudo cp mariadb-java-client-1.3.x.jar ¥
[TOMCAT install directory]/webapps/idp/WEB-INF/lib/.

$ sudo cp mariadb-java-client-1.3.x.jar /opt/shibboleth-idp/lib/.

```

Configuration for StoredID

Create table for StoredID in the MariaDB database.

<https://meatwiki.nii.ac.jp/confluence/display/GakuNinShibInstall/StoredID>

“4. Create table in the database (In case of MariaDB)”

```

$ mysql -u root vo
mysql> put SQL commands here.

```

Deployment of the trustany-ssl

```

$ sudo cp trustany-ssl-1.0.x.jar [TOMCAT install directory]/lib/.

```

Back Channel

Configure for back channel by referring to the following instruction.

Create credential

```

# cd /opt/shibboleth-idp/credentials
# UMASKORIG="`umask`" ; umask 0077
# openssl pkcs12 -export -out server.p12 -in idp.crt -inkey idp.key -name HOST-NAME-OF-THIS-SERVER
Enter Export Password: YOUR-OWN-PASSWRD
Verifying - Enter Export Password: YOUR-OWN-PASSWORD

# umask "$UMASKORIG"

```

* This instruction assumes IdP certificate and key as idp.crt and idp.key respectively.

Back Channel Port

Enable 8443 port in the server.xml of Tomcat configuration file.

Server.xml can be found in the following location if the Tomcat was installed by using yum

```
/usr/share/tomcat/conf/server.xml
```

Add following configuration.

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
  maxThreads="150"
  SSLEnabled="true"
  scheme="https"
  maxPostSize="100000"
  secure="true"
  clientAuth="want"
  sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
  keystoreFile="/opt/shibboleth-idp/credentials/server.p12"
  keystorePass="YOUR-OWN-PASSWORD"
  keystoreType="PKCS12"
  trustManagerClassName="net.shibboleth.utilities.ssl.TrustAnyCertificate" />
```

Deployment

```
$ sudo systemctl restart tomcat.service
```

Restart SP as well in order to include the metadata of newly configured IdP.

```
$ sudo systemctl restart shibd.service
$ sudo systemctl restart httpd.service
```

Installing the Application

Download files from the following URL.

```
cloudgateway-r1751.zip
```

Deployment

```
$ unzip cloudgateway-1.0.zip
$ sudo mv map /usr/local/.
```

Initial Setting


```
# Announce Information
$ cd /usr/local/map
$ mkdir -p app/webroot/tmp/
$ touch app/webroot/tmp/announce.txt
$ sudo chown -R apache.apache app/webroot/tmp

# Adjust permissions
$ sudo chown -R apache.apache /usr/local/map/app/tmp
$ sudo chmod +x /usr/local/map/cake/console/cake
$ sudo chmod +x /usr/local/map/app/vendors/shells/*.php
$ sudo chmod -R 777 /usr/local/map/app/tmp/cache

# Deletion of Log Files and Cache Files
$ sudo rm -rf app/tmp/cache/models/*
$ sudo rm -rf app/tmp/cache/persistent/*
$ sudo rm -rf app/tmp/cache/views/*
$ sudo rm -rf app/tmp/logs/*

# Copy of Configuration Files
$ cp app/config/database.template.php app/config/database.php
$ cp app/config/core.template.php app/config/core.php
# Put random data on 'Security.salt' and 'Security.cipherSeed' in core.php.
$ vi app/config/core.php

# Deletion of Files for Development (if exist)
$ rm app/config/local.php
```

Configuration of httpd.conf

Include following configuration in /etc/httpd/conf/httpd.conf

```
<VirtualHost _default_:80>
    Redirect permanent / https://[HOST-NAME-OF-THIS-SERVER]/
</VirtualHost>

Alias /map "/usr/local/map"
<Directory "/usr/local/map">
    Order allow,deny
    Allow from all
    Options ExecCGI FollowSymLinks
    AllowOverride All
</Directory>

<Location "/map">
    AuthType shibboleth
    ShibRequestSetting requireSession 0
    require shibboleth
</Location>
```

Restart

```
$ sudo systemctl restart httpd.service
```

Database Configuration

```
$ mysql -u root vo < /usr/local/map/ddl/ddl.sql
$ mysql -u root vo < /usr/local/map/ddl/alter.sql
$ mysql -u root vo < /usr/local/map/ddl/index.sql
$ mysql -u root vo < /usr/local/map/ddl/init_system_admin.sql
$ mysql -u root vo
Open /usr/local/map/ddl/stored_procedure.sql and copy & paste the contents.
```

Application Configuration File

- /usr/local/map/app/config/database.php
In the 'password' => ", set the password defined at "6. Installing and Setting Up MariaDB"
- /usr/local/map/app/config/mail.php
In the 'host' => ", set the SMTP server.
In the \$from, \$this->from, set the FROM address of the email which will be send from this server.
In the \$footer_jp_map, \$footer_jp_test, \$footer_eng_map and \$footer_eng_test, modify to the appropriate sentences as you want.
- /usr/local/map/app/config/system.php
In the "host" and the host of "loginlink" in \$config["production"], \$config["test"], set the host name of this server.
- Registration of System Administrator
Create a new account by accessing to the following URL.
<https://HOTS-NAME-OF-THIS-SERVER/map/>
After selecting the IdP and then login, create a new account.

Register the Administrator of the Database.

```
$ mysql -u vouser vo -pYOUR-OWN-PASSWORD

mysql> select id,name,mail from accounts;
Find your ID by the above command, and then and execute the following SQL to your ID.
mysql> insert into mygroups(account_id, groupid, admin, created, modified) values(YOUR-ACCOUNT-ID, 1, 1, UTC_TIMESTAMP(),
UTC_TIMESTAMP());

Example)
mysql> insert into mygroups(account_id, groupid, admin, created, modified) values(1, 1, 1, UTC_TIMESTAMP(), UTC_TIMESTAMP());
```

Registration of organizations

By the following commands, organization which if a part of the federaton member will be registered.

```
$ cd /usr/local/map/cake/console
$ ./cake -app /usr/local/map/app organization /var/cache/shibboleth/federation-metadata.xml
```

Registration of SP administrators

By the following commands, will be registered SP administrators.

```
$ cd /usr/local/map/cake/console
$ ./cake -app /usr/local/map/app sp_administrator /path/to/somewhere/sp_administrator.tsv
```

/path/to/somewhere/sp_administrator.tsv must be created in the format below.

```
#Example
#eppn      eptid      entityID
XXX@nii.ac.jp      https://test-sp.gakunin.nii.ac.jp/shibboleth-sp
```

Registration of SP

By the following commands, SP information which if a part of the federaton member will be registered.

```
$ cd /usr/local/map/cake/console
$ ./cake -app /usr/local/map/app sp_host /var/cache/shibboleth/federation-metadata.xml
```

Registration of IdP groups

By the following commands, IdP groups which if a part of the federaton member will be registered.

```
$ cd /usr/local/map/cake/console
$ ./cake -app /usr/local/map/app idp_group_creator /var/cache/shibboleth/federation-metadata.xml
```

Registration of SP connectors

By the following commands, IdP groups which if a part of the federaton member will be registered.

```
$ cd /usr/local/map/cake/console
$ ./cake -app /usr/local/map/app sp_connector_creator /var/cache/shibboleth/federation-metadata.xml test-map
```

Notify the administrator via mail

Notify the administrator when applying to join the group.

- /usr/local/map/shell/map_inspect

Modify URL for your servers.

```
# example
wget --spider --no-check-certificate https://localhost/map/batch_inspects/batch?mapurl=https%3a%2f%2fcg%2egakunin%2eip%
2fmap%2f > /dev/null 2>&1
```

- Set crontab for /usr/cron.d/map_inspect

```
# example) Nofity every 10 minutes
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
01,11,21,31,41,51 * * * * root /usr/local/map/shell/map_inspect
```

Validation

Authentication

Confirm the authentication page after accessing to the following URL.

<https://HOST-NAME-OF-THIS-SERVER/secure/attr.php>

SP

After integrating the metadata of this IdP into the related SPs and enable SimpleAggregation AttributeResolver, you can check the connecting test.

Advanced Configuration

Clustering

If you want to run on multiple servers to set the following.

Application Configuration File

- /usr/local/map/app/config/database.php
In the 'host' => ", set the host of database server.

database.php

```
@@ -76,9 +76,9 @@
    var $default = array(
        'driver' => 'mysql',
        'persistent' => false,
-       'host' => 'localhost',
+       'host' => 'YOUR-DATABASE-SERVER',
        'login' => 'vouser',
        'password' => 'xxxxx',
        'database' => 'vo',
```

- /usr/local/map/app/config/core.php

Configure to use a database to store the session.

core.php

```
@@ -124,7 +124,7 @@
 * the cake shell command: cake schema create Sessions
 *
 */
-    Configure::write('Session.save', 'map');
+    Configure::write('Session.save', 'database');
/**
 * The model name to be used for the session model.
@@ -133,7 +133,7 @@
 *
 * The model name set here should *not* be used elsewhere in your application.
 */
-    //Configure::write('Session.model', 'Session');
+    Configure::write('Session.model', 'Session');
/**
 * The name of the table used to store CakePHP database sessions.
@@ -147,14 +147,14 @@
 *
 * [Note: Session.table is deprecated as of CakePHP 1.3]
 */
-    //Configure::write('Session.table', 'cake_sessions');
+    Configure::write('Session.table', 'cake_sessions');
/**
 * The DATABASE_CONFIG::$var to use for database session handling.
 *
 * 'Session.save' must be set to 'database' in order to utilize this constant.
 */
-    //Configure::write('Session.database', 'default');
+    Configure::write('Session.database', 'default');
```

- /usr/local/map/cake/libs/cake_session.php
Modify value of 'session.cookie_lifetime' to 0 if deleting cookie when closed browser.

cake_session.php

```
@@ -513,7 +513,7 @@ class CakeSession extends Object {
        ini_set('session.serialize_handler', 'php');
        ini_set('session.use_cookies', 1);
        ini_set('session.name', Configure::read('Session.cookie'));
-        ini_set('session.cookie_lifetime', $this->cookieLifeTime);
+        ini_set('session.cookie_lifetime', 0);
        ini_set('session.cookie_path', $this->path);
        ini_set('session.auto_start', 0);
    }
```

Create sessions table

```
$ cd /usr/local/map/cake/console
$ ./cake schema create sessions
```

Shibboleth Configuration File

- /opt/shibboleth-idp/conf/attribute-resolver.xml
Modify the host part of the jdbcURL in <dc:ApplicationManagedConnection> elements.
- /etc/shibboleth/shibboleth2.xml
Add the <TCPLListener> element. Please see below URL for more information of <TCPLListener>.

<https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPTCPLListener>

Add the SP manually

Register the SP which is not a member of the federation

If you would like to register the SP which is not a member of the federation, it's easy to add the SP in the `/var/cache/shibboleth/federation-metadata.xml` and then execute the above command.

Register SP Administrator in the Database.

```
$ mysql -u vouser vo -pYOUR-OWN-PASSWORD

mysql> insert into sp_administrators(eppn, host_name, entityid, created)
values('YOUR-ePPN', 'HOST-NAME-OF-UTILIZED-SP',
'ENTITY-ID-OF-UTILIZED-SP', UTC_TIMESTAMP());

Example1) Registration for researchmap and kyouindb
mysql> insert into sp_administrators(eppn, host_name, entityid, created) ¥
values('xxxx@kyoto-u.ac.jp', 'researchmap.jp', 'https://researchmap.jp/shibboleth-sp', UTC_TIMESTAMP());

mysql> insert into sp_administrators(eppn, host_name, entityid, created) ¥
values('xxxx@kyoto-u.ac.jp', 'kyouindb.iimc.kyoto-u.ac.jp', ¥
'https://kyouindb.iimc.kyoto-u.ac.jp/shibboleth-sp', UTC_TIMESTAMP());

Exmaple2) Registration for test-meatmail.nii.ac.jp and test-map-sp1.nii.ac.jp
mysql> insert into sp_administrators(eppn, host_name, entityid, created) ¥
values('xxxxxx@ebook-idp.nii.ac.jp', 'test-meatmail.nii.ac.jp', ¥
'https://test-meatmail.nii.ac.jp/shibboleth-sp', UTC_TIMESTAMP());

mysql> insert into sp_administrators(eppn, host_name, entityid, created) ¥
values('xxxxxx@ebook-idp.nii.ac.jp', 'test-map-sp1.nii.ac.jp', ¥
'https://test-map-sp1.nii.ac.jp/shibboleth-sp', UTC_TIMESTAMP());
```

Create SP Connector

Create SP Connector of the utilized SP by executing the following SQL.

```
$ mysql -u vouser vo -pYOUR-OWN-PASSWORD

mysql> insert into groups(group_key,name,introduction,active,public,openmember,¥
inspectjoin,inspectquit,down_permission,up_permission,group_response,sp,¥
created,modified) values('GROUP-KEY', 'SERVICE-NAME', '', 0, 1, 1, 1, 0, 2, 0, 2, 1, UTC_TIMESTAMP(), UTC_TIMESTAMP());

Example1) Researchmap and kyouindb
mysql> insert into groups(group_key,name,introduction,active,public,openmember,¥
inspectjoin,inspectquit,down_permission,up_permission,group_response,sp,created,modified) ¥
values('researchmap', 'Researchmap', '', 0, 1, 1, 1, 0, 2, 0, 2, 1, UTC_TIMESTAMP(), UTC_TIMESTAMP());

mysql> insert into groups(group_key,name,introduction,active,public,openmember,¥
inspectjoin,inspectquit,down_permission,up_permission,group_response,sp,created,modified ¥
) values('kyouindb', 'kyouindb', '', 0, 1, 1, 1, 0, 2, 0, 2, 1, UTC_TIMESTAMP(), UTC_TIMESTAMP());

Example2) test-meatmail.nii.ac.jp and test-map-sp1.nii.ac.jp
mysql> insert into groups(group_key,name,introduction,active,public,openmember,¥
inspectjoin,inspectquit,down_permission,up_permission,group_response,sp,created,modified) ¥
values('test-meatmail.nii.ac.jp', 'test-meatmail.nii.ac.jp', '', ¥
0, 1, 1, 1, 0, 2, 0, 2, 1, UTC_TIMESTAMP(), UTC_TIMESTAMP());

mysql> insert into groups(group_key,name,introduction,active,public,openmember,¥
inspectjoin,inspectquit,down_permission,up_permission,group_response,sp,created,modified) ¥
values('test-map-sp1.nii.ac.jp', 'test-map-sp1.nii.ac.jp', '', ¥
0, 1, 1, 1, 0, 2, 0, 2, 1, UTC_TIMESTAMP(), UTC_TIMESTAMP());
```

Register the administrator of SP Connector to the database.

```
$ mysql -u vouser vo -pYOUR-OWN-PASSWORD
mysql> select id,name,mail from accounts;
mysql> select id,group_key,name from groups where sp=1;
```

Find your ID and Group table ID by the search command above and then put it in the "YOUR-ACCOUNT-ID" and "GroupID" in the following SQL.

```
mysql> insert into mygroups(account_id, groupid, admin, created, modified) values(YOUR-ACCOUNT-ID, GroupID, 1, UTC_TIMESTAMP(),
UTC_TIMESTAMP());
```

Example) In case of creating 3 SP Connectors

```
mysql> insert into mygroups(account_id, groupid, admin, created, modified) values(1, 2, 1, UTC_TIMESTAMP(), UTC_TIMESTAMP());
mysql> insert into mygroups(account_id, groupid, admin, created, modified) values(1, 3, 1, UTC_TIMESTAMP(), UTC_TIMESTAMP());
mysql> insert into mygroups(account_id, groupid, admin, created, modified) values(1, 4, 1, UTC_TIMESTAMP(), UTC_TIMESTAMP());
```

Connection between SP Connector and SP.

Connect SP Connector and SP by executing the following SQL.

```
$ mysql -u vouser vo -pYOUR-OWN-PASSWORD
```

Search utilizes SP Connector ID.

```
mysql> select id, name from groups where sp = 1;
```

Search utilizes SP ID

```
mysql> select id, name from sp_hosts;
```

Based on the search result, register connecting information between SP Connector and SP.

```
insert into group_sphosts(group_id,sp_id,lead_url,created,modified,service_name) ¥
values(SP-CONNECTOR-ID, SP-ID, 'SERVICE-URL', UTC_TIMESTAMP(),UTC_TIMESTAMP(), 'SERVICE-NAME');
```

- SP Connector ID : ID of groups table
- SP ID : ID of sp_hosts table

Example1) Researchmap and kyouindb

```
mysql> insert into group_sphosts(group_id,sp_id,lead_url,created,modified,service_name) ¥
values(2, 3, 'http://researchmap.jp/', UTC_TIMESTAMP(),UTC_TIMESTAMP(), 'Researchmap');
```

```
mysql> insert into group_sphosts(group_id,sp_id,lead_url,created,modified,service_name) ¥
values(3, 15, 'http://kyouindb.iimc.kyoto-u.ac.jp/', UTC_TIMESTAMP(),UTC_TIMESTAMP(), 'kyouindb');
```

Example2) test-meatmail.nii.ac.jp and test-map-sp1.nii.ac.jp

```
mysql> insert into group_sphosts(group_id,sp_id,lead_url,created,modified,service_name) ¥
values(2, 80, 'https://test-meatmail.nii.ac.jp/', UTC_TIMESTAMP(),UTC_TIMESTAMP(), 'Test-MeatMail');
mysql> insert into group_sphosts(group_id,sp_id,lead_url,created,modified,service_name) ¥
values(3, 175, 'https://test-map-sp1.nii.ac.jp/', UTC_TIMESTAMP(),UTC_TIMESTAMP(), 'Test-mAP-SP1');
```

Automatic Connection of SP Connector

This enables users to utilize SP (ex. Researchmap, kyouindb) by connecting SP connector automatically when the user create new group.

```
$ mysql -u vouser vo -pYOUR-OWN-PASSWORD
```

Obtain ID by searching SP Connector

```
mysql> select id, name from groups where sp = 1;
```

Set the found ID in the following SQL and then execute.

```
mysql> insert into sp_auto_connectors(groupid,created) values(FOUND-ID, UTC_TIMESTAMP());
```

Example)

```
mysql> insert into sp_auto_connectors(groupid,created) values(2, UTC_TIMESTAMP());
mysql> insert into sp_auto_connectors(groupid,created) values(3, UTC_TIMESTAMP());
```

Attribute Consent Setting for SP Connector

Set the consent information which will be utilized by the SP Connector

```
$ mysql -u vouser vo -pYOUR-OWN-PASSWORD
```

Obtain ID by searching SP Connector

```
mysql> select id, name from groups where sp = 1;
```

Set ID for SP Connector in the following SQL and then execute.

```
mysql> insert into provide_attributes(group_id,ismemberof,eptid,name,mail,idp,introduction,¥  
language,organization,created,modified) values(FOUND-ID, 1, 1, 1, 1, 1, 1, 1, 1, UTC_TIMESTAMP(),UTC_TIMESTAMP());
```

Example)

```
select id, name from groups where sp = 1;
```

```
+---+-----+  
| id | name          |  
+---+-----+  
| 2  | xxxxxx       |  
| 3  | yyyyyy       |  
| 4  | zzzzzz       |
```

```
mysql> insert into provide_attributes(group_id,ismemberof,eptid,name,mail,idp,introduction,¥  
language,organization,created,modified) values(2, 1, 1, 1, 1, 1, 1, 1, 1, UTC_TIMESTAMP(),UTC_TIMESTAMP());  
mysql> insert into provide_attributes(group_id,ismemberof,eptid,name,mail,idp,introduction,¥  
language,organization,created,modified) values(3, 1, 1, 1, 1, 1, 1, 1, 1, UTC_TIMESTAMP(),UTC_TIMESTAMP());  
mysql> insert into provide_attributes(group_id,ismemberof,eptid,name,mail,idp,introduction,¥  
language,organization,created,modified) values(4, 1, 1, 1, 1, 1, 1, 1, 1, UTC_TIMESTAMP(),UTC_TIMESTAMP());
```

Importing the Account Data from the Existing Database

If there exist the data in the existing database, it can be imported by means of TSV file.

Format of the TSV is as follows

Note that display name have to be within 50 characters.

```
LOCAL-ID(SPS-ID)<<TAB>>ePPN<<TAB>>DISPLAY-NAME(NAME)  
...
```

Import command is as follows.

```
$ export TERM=vt100  
$ cd /usr/local/map/cake/console  
$ ./cake -app /usr/local/map/app kyoto_u_ac_jp_import [TSV-FILE-NAME] [IdP-ENTITY-ID]
```

Example 1) Normal Execution

```
$ export TERM=vt100  
$ cd /usr/local/map/cake/console  
$ ./cake -app /usr/local/map/app kyoto_u_ac_jp_import ¥  
/var/local/map/tmp/kyoto-u.ac.jp.sample.tsv ¥  
https://authidp1.iimc.kyoto-u.ac.jp/idp/shibboleth
```

Example 2) Send Email after Execution

```
$ export TERM=vt100  
$ cd /usr/local/map/cake/console  
$ ./cake -app /usr/local/map/app kyoto_u_ac_jp_import ¥  
/var/local/map/tmp/kyoto-u.ac.jp.sample.tsv ¥  
https://authidp1.iimc.kyoto-u.ac.jp/idp/shibboleth | mail -s "Import Result" EMAIL-ADDRESS
```