JWT-SERVERインストールマニュアル

内容

[tomcat のインストール 3](#_Toc97725164)

[JDKのインストール 3](#_Toc97725165)

[javaのファイルにパスを通す 3](#_Toc97725166)

[tomcatを起動するユーザーを作成 3](#_Toc97725167)

[tomcatのダウンロード＆展開 3](#_Toc97725168)

[tomcatのサービス登録 3](#_Toc97725169)

[tomcatサービスに権限を与えて起動できるようにし、起動 4](#_Toc97725170)

[apache のインストール 4](#_Toc97725171)

[tomact と apache の連携 4](#_Toc97725172)

[proxyの設定 4](#_Toc97725173)

[ポートの設定 4](#_Toc97725174)

[apache&tomcatの再起動と自動起動設定 5](#_Toc97725175)

[Maven （3.2以上）のインストール 5](#_Toc97725176)

[パスの設定 5](#_Toc97725177)

[jwt-server のインストール 5](#_Toc97725178)

[環境設定 6](#_Toc97725179)

[ビルド＆デプロイ 6](#_Toc97725180)

[mysql のインストール 6](#_Toc97725181)

[mysqlの起動と自動起動設定 6](#_Toc97725182)

[初期設定 7](#_Toc97725183)

[パスワードの変更 7](#_Toc97725184)

[環境設定 7](#_Toc97725185)

[ユーザ作成 7](#_Toc97725186)

[データベース作成 7](#_Toc97725187)

[Keycloakの設定 8](#_Toc97725188)

[realmnの作成 8](#_Toc97725189)

[clientの作成 8](#_Toc97725190)

[ユーザの作成 10](#_Toc97725191)

# tomcat のインストール

## JDKのインストール

# yum -y install java-1.8.0-openjdk-devel

## javaのファイルにパスを通す

# echo "export JAVA\_HOME=$(readlink -e $(which java)|sed 's:/bin/java::')" > /etc/profile.d/java.sh

# echo "PATH=\$PATH:\$JAVA\_HOME/bin" >> /etc/profile.d/java.sh

# source /etc/profile.d/java.sh

## tomcatを起動するユーザーを作成

# useradd -s /sbin/nologin tomcat

## tomcatのダウンロード＆展開

# mkdir -p /usr/local/src/tomcat

# cd /usr/local/src/tomcat

# curl -OL https://dlcdn.apache.org/tomcat/tomcat-8/v8.5.76/bin/apache-tomcat-8.5.76.tar.gz

# tar zxvf apache-tomcat-8.5.76.tar.gz

# mv apache-tomcat-8.5.76 /opt/

# chown -R tomcat. /opt/apache-tomcat-8.5.76

# ln -s /opt/apache-tomcat-8.5.76 /opt/tomcat

# echo 'export CATALINA\_HOME=/opt/tomcat' > /etc/profile.d/tomcat.sh

# source /etc/profile.d/tomcat.sh

## tomcatのサービス登録

# vi /etc/systemd/system/tomcat.service

（sample）

[Unit]

Description=Apache Tomcat 8

After=syslog.target network.target

[Service]

User=tomcat

Group=tomcat

Type=oneshot

PIDFile=/opt/tomcat/tomcat.pid

RemainAfterExit=yes

ExecStart=/opt/tomcat/bin/startup.sh

ExecStop=/opt/tomcat/bin/shutdown.sh

ExecReStart=/opt/tomcat/bin/shutdown.sh;/opt/tomcat/bin/startup.sh

[Install]

WantedBy=multi-user.target

## tomcatサービスに権限を与えて起動できるようにし、起動

# chmod 755 /etc/systemd/system/tomcat.service

# systemctl start tomcat

# apache のインストール

# yum -y install httpd mod\_ssl

# systemctl start httpd

# tomact と apache の連携

## proxyの設定

# vi /etc/httpd/conf.d/proxy-ajp.conf

（追加）

ProxyPass /jwt-server ajp://localhost:8009/jwt-server secret=password

ProxyPassReverse /jwt-server ajp://localhost:8009/jwt-server secret=password

## ポートの設定

# vi /opt/tomcat/conf/server.xml

※8080ポートアクセスの不許可

<!--

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

-->

※8009ポートからのアクセス許可

<Connector protocol="AJP/1.3"

address="localhost"

port="8009"

redirectPort="8443"

secret="password" />

## apache&tomcatの再起動と自動起動設定

# systemctl restart tomcat

# systemctl restart httpd

# systemctl enable tomcat

# systemctl enable httpd

# Maven （3.2以上）のインストール

# curl -OL https://archive.apache.org/dist/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz

# tar -xzvf apache-maven-3.8.4-bin.tar.gz

# mv apache-maven-3.8.4 /opt/

# cd /opt

# ln -s /opt/apache-maven-3.8.4 apache-maven

## パスの設定

% vi ~/.bash\_profile

（追加）

JAVA\_HOME=/usr/java/default/

PATH=$PATH:/opt/apache-maven/bin

% source ~/.bash\_profile

# jwt-server のインストール

% git clone https://github.com/oss-tsukuba/jwt-server.git

% cd jwt-server

## 環境設定

% vi src/main/resources/application.properties

（設定例)

# change keycloak settings

keycloak.enabled=true

keycloak.auth-server-url=https://can3.canaly.co.jp:8443/auth

keycloak.realm=hpci

keycloak.resource=jwt-server

keycloak.public-client=false

keycloak.credentials.secret=8a5a0480-1d7a-420c-b3d9-93b65439d321

user-claim=hpci.id

#change MySQL settings

spring.datasource.url=jdbc:mysql://${MYSQL\_HOST:localhost}:3306/gfarmdb

spring.datasource.username=dbuser

spring.datasource.password=passworD12@

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.database=MYSQL

spring.jpa.hibernate.ddl-auto=update

各設定値の内容は以下の通りである。

* keycloak.enabled

Keycloak Spring Boot アダプターの有効/無効を設定する。

* keycloak.auth-server-url

KeycloakサーバのURLを設定する。

* keycloak.realm

レルムを設定する。

* keycloak.resource

クライアントIDを設定する。

* keycloak.public-client

クライアントがpublicか否かを設定する。

* keycloak.credentials.secret

クライアントのシークレットを設定する。クライアントがpublicの場合は不要である。

* user-claim

ユーザIDとして利用するトークンのクレームを設定する。

* spring.datasource.url

データベースのURLを設定する。MySQLを利用する想定である。

* spring.datasource.username

データベース名を設定する。

* spring.datasource.password

データベースにアクセスするユーザを設定する。

* spring.datasource.driver-class-name

ユーザのパスワードを設定する。

* spring.jpa.database

データベース（DBMS）の種類を設定する。

* spring.jpa.hivernate.ddl-auto

データベースのスキーマの生成方法を設定する。

## ビルド＆デプロイ

% mvn package spring-boot:repackage

% sudo cp target/jwt-server.war /opt/tomcat/webapps/

# mysql のインストール

# yum remove -y mariadb-libs

# rm -rf /var/lib/mysql/

# yum localinstall -y http://dev.mysql.com/get/mysql57-community-release-el7-7.noarch.rpm

# yum install -y mysql-community-server

(\*) 鍵が古い場合は更新する。

# rpm --import <https://repo.mysql.com/RPM-GPG-KEY-mysql-2022>

## mysqlの起動と自動起動設定

# systemctl start mysqld

# systemctl enable mysqld

## 初期設定

# cat /var/log/mysqld.log | grep root

2019-01-24T17:10:57.681293Z 1 [Note] A temporary password is generated for root@localhost: [ここにパスワードの文字が記載されているはず]

# mysql -u root -p //上記のパスワードでログイン

Enter password:

## パスワードの変更

mysql> SET PASSWORD = PASSWORD('任意のパスワード');

※8文字以上かつ英大文字・小文字・数字・記号を含める

## 環境設定

# vi /etc/my.cnf

（追記）

character\_set\_server=utf8

skip-character-set-client-handshake

# systemctl restart mysqld

## ユーザ作成

% mysql -u root -p

mysql> CREATE USER 'dbuser'@'localhost' IDENTIFIED BY 'passworD12@';

mysql> GRANT ALL PRIVILEGES ON \* . \* TO 'dbuser'@'localhost';

mysql> CREATE DATABASE gfarmdb;

mysql> FLUSH PRIVILEGES;

mysql> quit

## データベース作成

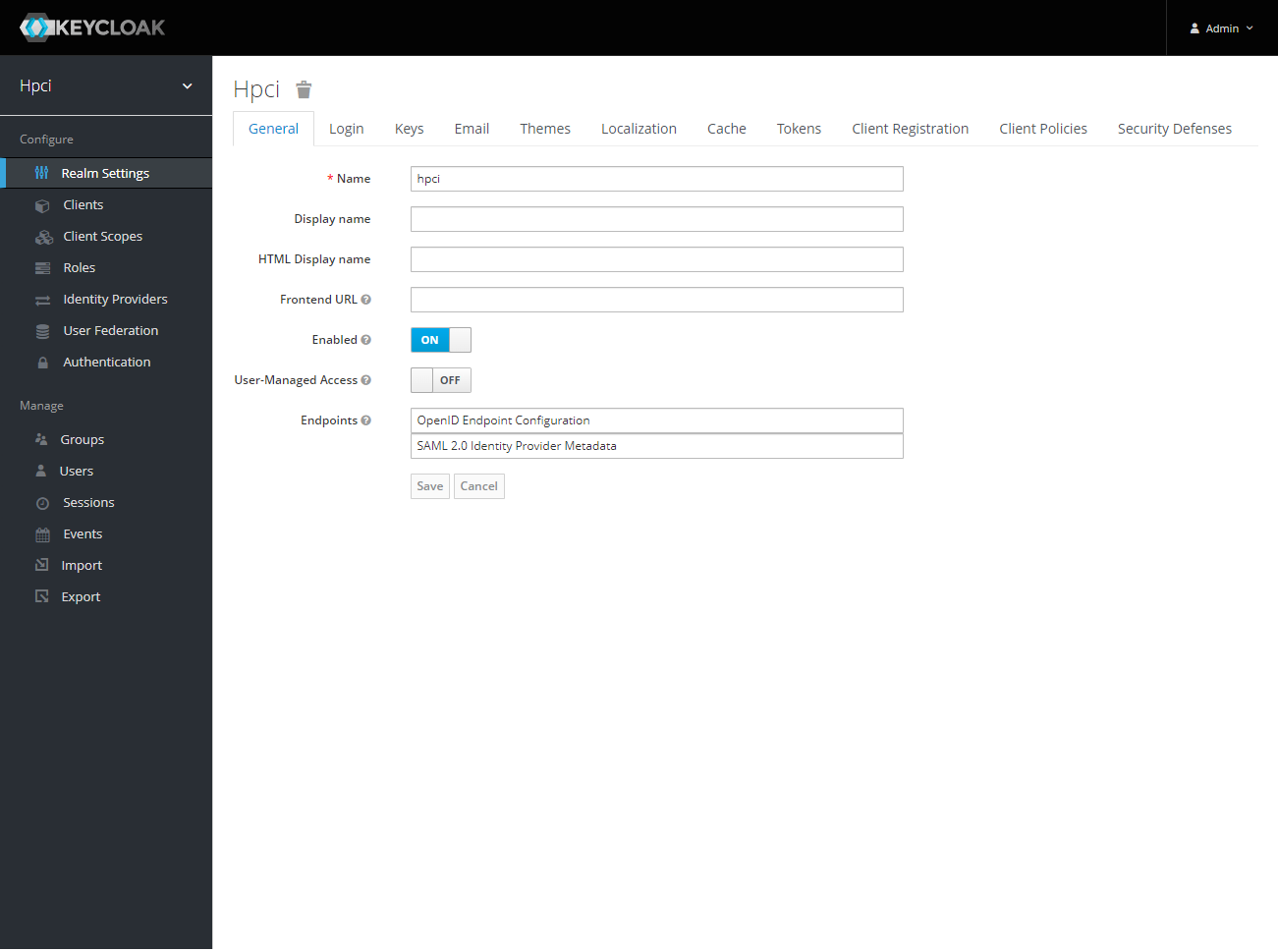
% mysql -u dbuser -p

% mysql -u dbuser -p gfarmdb < jwt-server/ddl/jwt-server.ddl

# Keycloakの設定

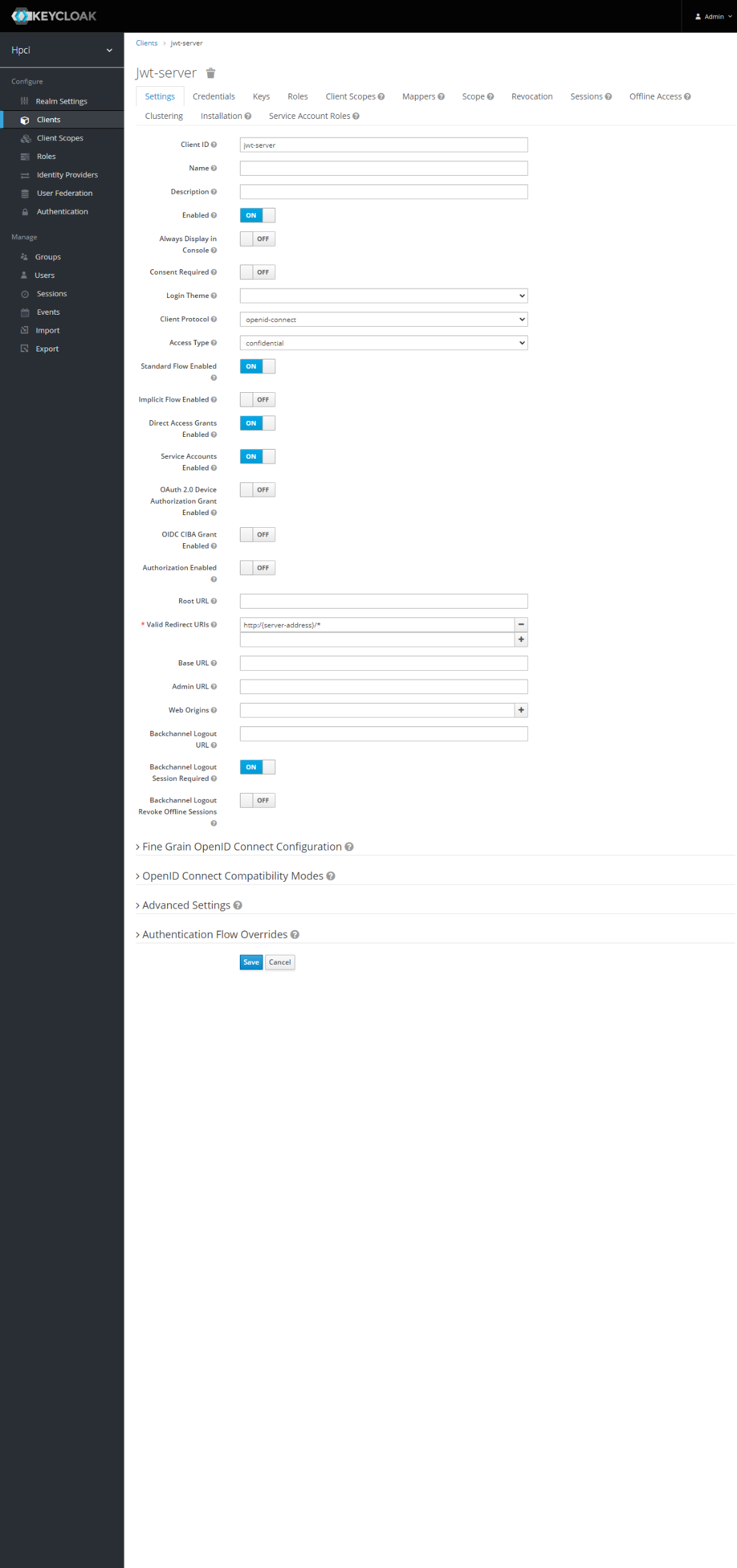
## realmnの作成

レルム「hpci」を作成する。



## clientの作成

hcpi配下にクライアント「jwt-server」を作成する。トークンの認証を行うため、「Access Type」はconfidentialにする必要がある。



## ユーザの作成

必要に応じてユーザを作成する。

