

Guida installazione Idp di Piattaforma

Prerequisiti:

Ubuntu 14.04 – 4 VCPU – 8 RAM - 20 GB

Certificato SSL per il dominio

Security group per porta 80, 443, 8080, 8443

Installazione JAVA JDK 7:

Installare jdk 7 dal sito di Oracle come da guida al seguente link:

<http://www.webupd8.org/2012/01/install-oracle-java-jdk-7-in-ubuntu-via.html>

```
sudo add-apt-repository ppa:webupd8team/java  
sudo apt-get update  
sudo apt-get install oracle-java7-installer
```

accettare le licenze oracle.

Una volta installato verificarne il risultato e la versione con il comando:

```
java -version
```

Prendere la path di dove è installato java... ad esempio:

```
/usr/lib/jvm/java-7-oracle
```

Copy the path from your preferred installation and then edit the file `/etc/environment`:

```
vi /etc/environment
```

In this file, add the following line (replacing YOUR_PATH by the just copied path):

```
JAVA_HOME="YOUR_PATH"
```

That should be enough to set the environment variable. Now reload this file:

```
source /etc/environment
```

Test it by executing:

```
echo $JAVA_HOME
```

If it returns the just set path, the environment variable has been set successfully. If it doesn't, please make sure you followed all steps correctly.

Installazione Tomcat:

Scaricare la versione 8.0.9 da <http://tomcat.apache.org/>

```
cd /opt
```

```
wget https://archive.apache.org/dist/tomcat/tomcat-8/v8.0.9/bin/apache-tomcat-8.0.9.zip
```

```
apt-get install unzip
```

```
unzip apache-tomcat-8.0.9.zip
```

```
chmod -R 777 apache-tomcat-8.0.9
```

```
cd /opt/apache-tomcat-8.0.9/bin
```

```
opt/apache-tomcat-8.0.9/bin# ./startup.sh
```

Using CATALINA_BASE: /opt/apache-tomcat-8.0.9

Using CATALINA_HOME: /opt/apache-tomcat-8.0.9

Using CATALINA_TMPDIR: /opt/apache-tomcat-8.0.9/temp

Using JRE_HOME: /usr/lib/jvm/java-7-oracle

Using CLASSPATH: /opt/apache-tomcat-8.0.9/bin/bootstrap.jar:/opt/apache-tomcat-8.0.9/bin/tomcat-juli.jar

Tomcat started.

Ricordarsi di aprire il security group per la porta http 8080 sulla vm

<http://MYIP:8080/>

Per stoppare tomcat

```
cd /opt/apache-tomcat-8.0.9/bin#
```

dare il seguente comando

```
./shutdown.sh
```

IMPORTANTE

Creare il file setenv.sh sotto bin folder:

```
/opt/apache-tomcat-8.0.9/bin# vi setenv.sh
```

Inserire la seguente configurazione:

```
JAVA_OPTS="$JAVA_OPTS -Djava.security.egd=file:/dev/./urandom"
```

Si evita così di avere problemi con la generazione del file di entropia da parte di Tomcat e quindi tempi di deploy estremamente lenti.

Installazione apache2:

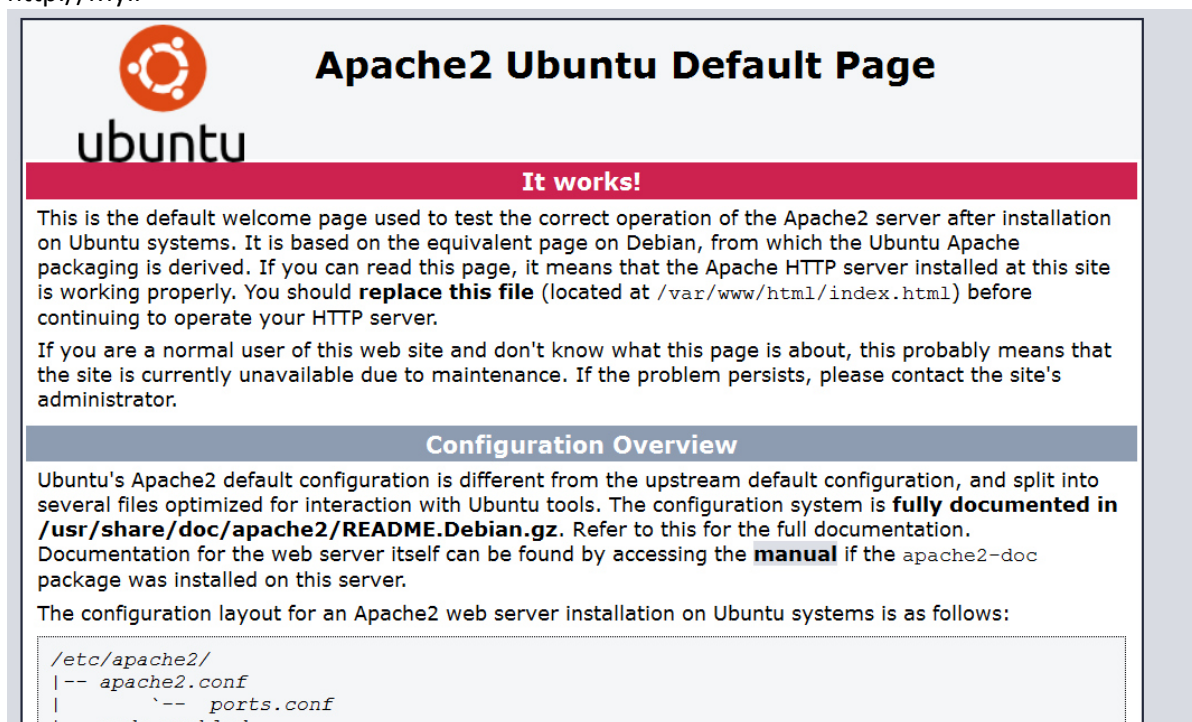
```
sudo apt-get update
sudo apt-get install apache2
```

Se ci sono dei warning si possono togliere andando a settare in apache2.conf

```
ServerName localhost
```

Per visualizzare la pagina di apache da browser assicurarsi di aver abilitato sulla VM la porta relativa su http.

http://MyIP



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
```

Installazione MYSQL

Installare MySQL server eseguendo:

```
apt-get install mysql-server
```

During the installation, MySQL will ask you to set a root password. If you miss the chance to set the password while the program is installing, it is very easy to set the password later from within the MySQL shell.

impostando le credenziali

```
username: root
password: XXXXXXXXXXXXXXXXX
```

Riavviare il servizio di MySQL Server

```
service mysql restart
```

Testare l'accesso al DB da remoto eseguendo in una macchina esterna

```
mysql -h <IP> -u root -pXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Testare l'accesso in localhost :

```
mysql -uroot -pXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Installazione PHP

```
sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt
(in caso ci siano problemi provare ad installare anche libapache2-mod-auth-mysql)
```

```
vi /etc/apache2/mods-enabled/dir.conf
```

Add index.php to the beginning of index files. The page should now look like this:

```
<IfModule mod_dir.c>

    DirectoryIndex index.php index.html index.cgi index.pl index.php
index.xhtml index.htm

</IfModule>
```

Installazione phpMyAdmin

```
sudo apt-get install phpmyadmin
```

scegliere apache2 (la selezione dovrebbe già essere su quella voce e dare invio)

Dare NO.

Immettere utente root e pwd usata nell'installazione di MySQL.

Al termine dell'installazione dovrete accedere tramite il link: (sostituire con IP del server)

<http://localhost/phpmyadmin>



Benvenuto in phpMyAdmin

Lingua - Language

Italiano - Italian

Connetti 

Nome utente:

Password:

Esegui

 Da questo punto in poi, i cookie devono essere abilitati.

In caso di errori creare il seguente link simbolico:

```
sudo ln -s /usr/share/phpmyadmin /var/www/phpmyadmin
```

questo comando creerà un link simbolico nella cartella di apache che punterà al percorso reale dell'applicazione, infatti può succedere che lo script di installazione non riesca ad effettuare automaticamente questa operazione.

```
vi /etc/apache2/apache2.conf
```

Add the following to the end of the file:

```
# phpMyAdmin Configuration
Include /etc/phpmyadmin/apache.conf
```

Then exit and save the file.

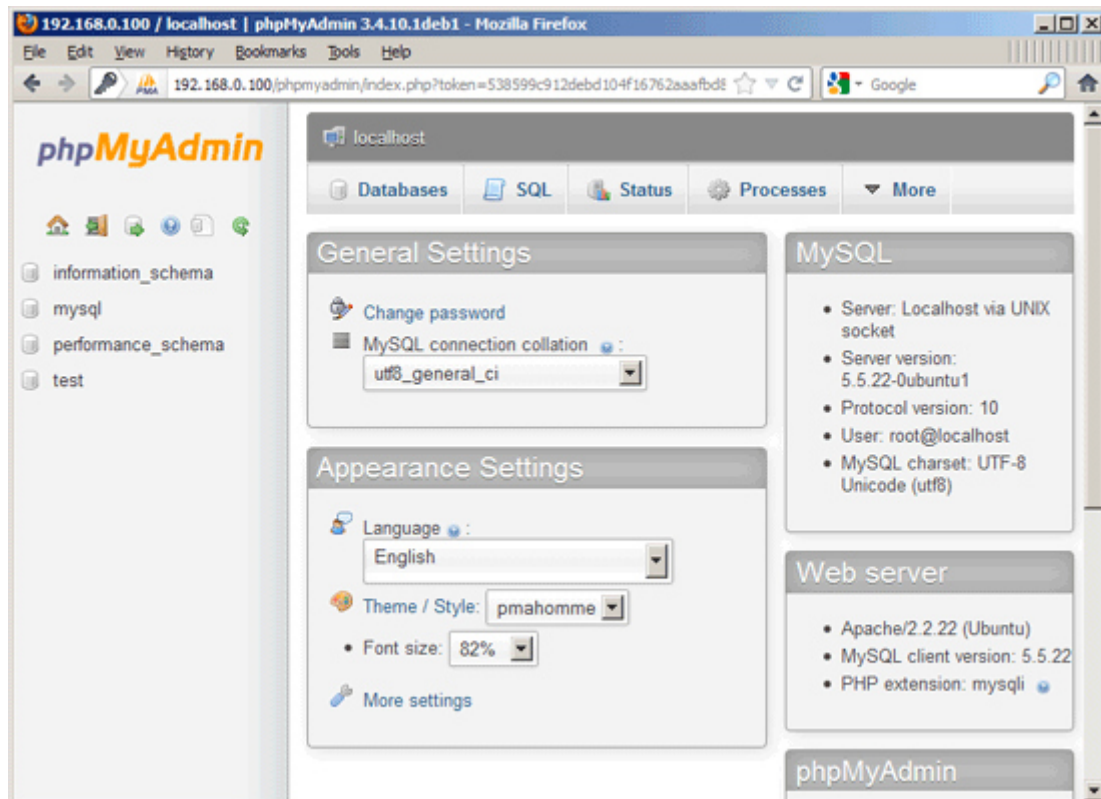
And, restart Apache 2 with the following command:

```
service apache2 restart
```

Ritestare il link e dovrebbe apparire l'interfaccia di phpmyAdmin.

<http://MYIP/phpmyadmin/>

Una volta loggati si entra nella seguente home page:



CONFIGURAZIONE DATABASE

Scaricare il dump relativo all'idp.

Loggarsi su phpmyadmin, creare un nuovo schema di tipo *InnoDB* → *utf8_general_ci* con nome a piacere.

Creare uno user (diverso da root) che abbia tutti i privilegi su quello schema. Esso verrà poi utilizzato nel file authsources.php di simplesamlphp. Queste credenziali saranno anche inserite anche nel file di properties del codice war prima del deploy. Vedi per questo il paragrafo dedicato al Deploy, in fondo a questo documento.

INSTALL simplesaml

Scaricare il pacchetto già modificato di simplesamlphp dal repository di Progetto, metterlo sotto /var e scompattarlo.

Abilitare i permessi **777** con `chmod` della subfolder log :

```
/var/simplesamlphp/log
```

Configurare i file di apache2

```
cd /etc/apache2/sites-available
```

```
vi 000-default.conf
```

Modificare su esempio della parte selezionata in verde:

<VirtualHost *:80>

```
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com
```

```
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
```

ServerName myname.dominio.it

```
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn
```

```
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
```

Redirect permanent /simplesaml https://myname.dominio.it /simplesaml

</VirtualHost>

```
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

HTTPS

Abilitare il modulo ssl su apache2

```
sudo a2enmod ssl
```

```
sudo service apache2 restart
```

Andare sotto /etc/apache2/site

```
etc/apache2/sites-available# ls
000-default.conf default-ssl.conf ..
```

Impostare in `default-ssl.conf` la porta 443, il nome macchina (.....dominio.it), i certificati, alias per `simplesaml`.. vedi parte evidenziata in giallo:

```
<IfModule mod_ssl.c>
```

```
<VirtualHost *:443>
```

```
ServerAdmin webmaster@localhost
```

```
DocumentRoot /var/www/html
```

```
ServerName myname.dominio.it
```

```
# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn
```

```
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

```
# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
```

```
# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on
```

```
# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
```

```
SSLCertificateFile /root/keys/Mycert.cer
```

```
SSLCertificateKeyFile /root/keys/Mykey.key
```

....

```
BrowserMatch "MSIE [2-6]" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
# MSIE 7 and newer should be able to use keepalive
BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
```

```
Alias /simplesaml /var/simplesamlphp/www
```

```
<Directory /var/simplesamlphp/www/>
```



```
Require all granted
</Directory>
```

```
</VirtualHost>
</IfModule>
```

Salvare e poi riavviare apache2:

```
service apache2 restart
```

Andare su browser a:
<https://myIP/simplesaml>

Di seguito un esempio della pagina ufficiale del pacchetto simpleSAMLphp non modificata
https://myIP/simplesaml/module.php/core/frontpage_welcome.php



Le pagine php di progetto sono state modificate con altra grafica. <https://myIP/simplesaml>

```
service apache2 stop
```

CONFIGURARE SIMPLESAMPLPHP

Andare a modificare il file di configurazione di simplesaml:

Editare il main configuration file, `config.php`, che si trova sotto la folder
`/var/simplesamlphp/config`:

```
vi config.php
```

Set administrator password. This is needed to access some of the pages in your simpleSAMLphp installation web interface. Solitamente la password di default dell'admin è 123 cambiarla

```

*/
'auth.adminpassword' => '123',
'admin.protectindexpage' => false,
'admin.protectmetadata' => false,

```

```
'auth.adminpassword' => 'setnewpasswordhere',
```

- Set a secret salt. This should be a random string. Some parts of the simpleSAMLphp needs this salt to generate cryptographically secure hashes. SimpleSAMLphp will give an error if the salt is not changed from the default value. The command below can help you to generated a random string on (some) unix systems:

```
tr -c -d '0123456789abcdefghijklmnopqrstuvwxyz' </dev/urandom | dd bs=32
count=1 2>/dev/null;echo
```

Here is an example of the config option:

```
'secretsalt' => 'randombytesinsertedhere',
```

- Set technical contact information. This information will be available in the generated metadata. The e-mail address will also be used for receiving error reports sent automatically by simpleSAMLphp. Here is an example:

```
'technicalcontact_name' => 'AdministratorName',
'technicalcontact_email' => 'na@example.org',
```

- If you use simpleSAMLphp in a country where english is not widespread, you may want to change the default language from english to something else:

```
'language.default' => 'no',
Lasciare anche Inglese con 'en'
```

Set the timezone which you use:

- 'timezone' => 'Europe/Rome',
Dovrebbe essere già impostato *Europe/Rome*, in caso si voglia un altro Timezone vedere il seguente sito web: List of Supported Timezones at php.net
- Abilitare l'IDP SAML

```
'enable.saml20-idp' => true,
'enable.shib13-idp' => false,
'enable.adfs-idp' => false,
'enable.wsfed-sp' => false,
'enable.authmemcookie' => false,
```

Configurare il data source al database:

```
cd /var/simplesamlphp/config
```

```
vi authsources.php
```

Modificare la parte in giallo, con i dati del proprio host database, schema, utente e password.

```
'sql-idp-autenticazione' => array(
    'sqlauth:SQL',
    'dsn' => 'mysql:host=127.0.0.1;dbname=XXXXXXX',
    'username' => 'XXXXXXXXXX',
    # 'certSSL' => '/opt/simplesamlphp/.../MY-cert.pem', Insert here your CAcert when DB is
    host in another machine

    'password' => 'XXXXXXXXXX',
    'query' => 'SELECT username,password,firstName,email FROM User WHERE username =
:username AND password = sha1(:password) AND isActive=TRUE',
),
```

CONFIGURAZIONE METADATI

GENERAZIONE CHIAVI E CERTIFICATO

Generare la coppia di chiavi pubblica e privata e il certificato; entrare nella cartella `simplesamlphp/cert/` e eseguire il comando

```
openssl req -new -x509 -days 3652 -nodes -out saml.crt -keyout saml.pem
```

per generare il certificate per l'IdP.

Configurazione metadata

Questa configurazione è configurata per la Piattaforma, in altri casi fare riferimento alla guida di simplesaml.

Modificare `simplesamlphp/metadata/saml20-idp-hosted.php` inserendo il nome della chiave e del certificate appena creato

```
'privatekey' => 'saml.pem',
```

```
'certificate' => 'saml.crt',
```

Vanno quindi aggiunti all'IDP i metadati dei portali in cui si vuole abilitare il login tramite questo IDP;

Bisogna modificare quindi il file `simplesamlphp/metadata/saml20-sp-remote.php` aggiungendo al file, per ogni portale, un valore alla variabile globale (array associativo) `$metadata`, inserendo questi valori ricavati dal metadata.xml ricevuto dal gestore del portale

```
$metadata['ENTITY_ID_DEL_PORTALE'] = array (
    'entityid' => 'ENTITY_ID_DEL_PORTALE',
    'contacts' =>
    array (
    ),
    'metadata-set' => 'saml20-sp-remote',
    'AssertionConsumerService' =>
    array (
    0 =>
```

```

array (
  'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST',
  'Location' => 'https://DOMAIN_NAME_DEL_PORTALE:443/saml/SSO/alias/defaultAlias',
  'index' => 0,
  'isDefault' => true,
),
1 =>
array (
  'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact',
  'Location' => 'https://DOMAIN_NAME_DEL_PORTALE:443/saml/SSO/alias/defaultAlias',
  'index' => 1,
),
),
'SingleLogoutService' =>
array (
  0 =>
array (
  'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST',
  'Location' => 'https://DOMAIN_NAME_DEL_PORTALE:443/saml/SingleLogout/alias/defaultAlias',
),
1 =>
array (
  'Binding' => 'urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect',
  'Location' => 'https://DOMAIN_NAME_DEL_PORTALE:443/saml/SingleLogout/alias/defaultAlias',
),
),
'NameIDFormat' => 'urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress',
'keys' =>
array (
  0 =>
array (
    'encryption' => false,
    'signing' => true,
    'type' => 'X509Certificate',
    'X509Certificate' => 'CERTIFICATO X509 contenuto nel tag <ds:X509Certificate> contenuto a sua volta
nel tag <md:KeyDescriptor use="signing">'
  ),
  1 =>
array (
    'encryption' => true,
    'signing' => false,
    'type' => 'X509Certificate',
    'X509Certificate' => 'CERTIFICATO X509 contenuto nel tag <ds:X509Certificate> contenuto a sua volta
nel tag <md:KeyDescriptor use="encryption">'
  ),
),
'validate.authnrequest' => true,
'saml20.sign.assertion' => true,
);

```

APACHE TOMCAT

Abilitare https in Tomcat sulla porta 8443 inserendo (o modificando se già presenti) le seguenti righe nel file `/conf/server.xml`

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol"
    maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS"
    keystoreFile="/root/keys/Mycert.pfx"
    keystorePass="PASSWORD_KEYSTORE"
    keystoreType="PKCS12"
    keyAlias="ALIAS_CHIAVE"
    keyPass="PASSWORD_CHIAVE" />
```

DEPLOY IDP

Scaricare il war relativo al progetto idp dal repository.

Modificare il file `jdbcIdp.properties` contenuto nel war in questo modo:

```
driverDB=com.mysql.jdbc.Driver
connectionDB=jdbc:mysql://127.0.0.1:3306/IDP_SCHEMA
userDB=USERNAME_DB
passwordDB=PASSWORD_DB
```

Abilitare il deploy del war aggiungendo le seguenti righe in `/conf/server.xml`

```
<Context path="" docBase="NOME_DEL_WAR_SENZA_.WAR">
    <!-- Default set of monitored resources -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
</Context>
```

Deployare il war su tomcat.

Ricordarsi di modificare nel dump generico di progetto, ossia in `prisma_paas` i seguenti dati:
aggiunto campo `prismaIdP` nella tabella `IdentityProvider`. Settarlo a `true` solo per l'IdP di Piattaforma.