

Shibboleth IDPv3 Installation Guide For Centos7

1. Configure Firewall

```
sudo firewall-cmd --zone=public --add-service=https
sudo firewall-cmd --zone=public --permanent --add-service=https
sudo firewall-cmd --zone=public --add-service=http
sudo firewall-cmd --zone=public --permanent --add-service=http
sudo firewall-cmd --zone=public --permanent --add-port=8443/tcp
sudo firewall-cmd --zone=public --permanent --list-services
restart:
systemctl restart firewalld
```

2. Configure SELinux mode

Open vi /etc/sysconfig/selinux

```
SELINUX=permissive
```

Ensure your system is running in permissive mode.

Enter command on console:

```
setenforce permissive
```

3. Install packages

Apache with mod_ssl, Java8 (with javac):

```
yum install httpd mod_ssl java-1.8.0-openjdk java-1.8.0-openjdk-devel
sudo systemctl enable httpd.service
sudo apachectl configtest
sudo systemctl restart httpd.service
```

MariaDB Server:

```
yum install mariadb mariadb-server
yum install mariadb mariadb-server
systemctl enable mariadb
systemctl start mariadb
```

Tomcat Server:

```
sudo yum install tomcat
systemctl enable tomcat
chkconfig tomcat on
service tomcat start
```

Make all files under /opt/shibboleth-idp owned by Tomcat:

```
chown -R tomcat.tomcat /opt/shibboleth-idp
```

For extra security, you may consider making files where passwords to MySQL are stored readable only to the Tomcat users:

```
chmod 600 /opt/shibboleth-idp/conf/ldap.properties /opt/shibboleth-idp/conf/attribute-resolver.xml
```

4. Basic Shibboleth IdP installation

Define Environmental Variable:

vi /etc/profile.d/shib.sh

```
IDP_VERSION="3.3.3"
SHIB_HOME=/opt/shibboleth-idp
SHIB_INST_HOME=/root/inst/shibboleth-identity-provider-$IDP_VERSION

IDP_HOME=/opt/shibboleth-idp
JAVA_HOME=/usr/lib/jvm/java

export SHIB_HOME IDP_HOME JAVA_HOME SHIB_INST_HOME IDP_VERSION
```

Make it executable:

```
chmod 777 /etc/profile.d/shib.sh

source /etc/profile.d/shib.sh
```

Check for the most recent version of Shibboleth IdP at <https://shibboleth.net/downloads/identity-provider/>

Create an installation directory and download Shibboleth:

```
mkdir /root/inst
cd /root/inst

sudo curl -O https://shibboleth.net/downloads/identity-provider/latest/shibboleth-identity-provider-3.3.3.tar.gz

sudo tar -zxf shibboleth-identity-provider-3.3.3.tar.gz
cd $SHIB_INST_HOME

sh ./bin/install.sh
```

Answer the following questions:

- Source (Distribution) Directory: confirm the current directory
- Installation directory: accept the proposed value of /opt/shibboleth-idp if suitable.
- Hostname: enter the user-facing hostname of the IdP, typically idp.example.org
- SAML Entity ID: accept the value derived from the hostname, https://idp.example.org/idp/shibboleth
- Attribute Scope: set this to the domain name of your institution (example.org in the above example). The value offered based on the hostname should be already correct, but please check and adjust as needed.
- Enter the passphrase to protect the generated keystores (back-channel and cookie encryption). It is acceptable to use the phrase "changeit" (as the files are protected by filesystem permissions and the phrase itself is also stored in files on the same filesystem).
 - Note that the installer generates three separate certificates+keypairs for back-channel, signing, and encryption, respectively - and also an encryption key for cookie. Only the back-channel private key is encrypted, the other two private keys will be stored on disk unencrypted. The passphrase for the cookie encryption key is stored in the generated idp.properties file. Therefore, there is no need to choose a secure keystore password and instead, it is important to secure access to the system...
- This installs the Shibboleth IdP web application into /opt/shibboleth-idp/war/idp.war

The installer has generated three separate certificates+keypairs for back-channel, signing, and encryption. The back-channel private key was only stored in a Java keystore, but for Apache, we need it converted to the PEM format. Run the following command (and when prompted for the keystore passphrase, enter the default passphrase "changeit"):

```
openssl pkcs12 -in $IDP_HOME/credentials/idp-backchannel.p12 -out
$IDP_HOME/credentials/idp-backchannel.key -nocerts -nodes
```

5. Configure Tomcat and deploy the IdP WAR

- Create `/etc/tomcat/Catalina/localhost/idp.xml` with the following content:

```
<Context docBase="/opt/shibboleth-idp/war/idp.war"
  privileged="true"
  antiResourceLocking="false"
  antiJARLocking="false"
  unpackWAR="false"
  swallowOutput="true" />
```

- Connectors: in `/etc/tomcat/server.xml`, define a new AJP connector at port 8009.

Disable already created connector

```
<!-- <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" /> -->
```

Create new AJP connector at port 8009

```
<Connector port="8009" address="127.0.0.1"
  enableLookups="false" redirectPort="443" protocol="AJP/1.3"
  tomcatAuthentication="false" />
```

- And comment out the existing http connector defined for port 8080:

```
<!--
<Connector port="8080" protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443" /> -->
```

- Edit `/etc/sysconfig/tomcat` and add (reasonably adjust according to the VM size):

```
JAVA_OPTS="-Xms768m -Xmx1536m"
```

The IdPV3 code relies on the web application container having support for JSTL, but Tomcat7 comes packaged without JSTL. Therefore, install JSTL (version 1.2.1, API and implementation jars) into `/usr/share/tomcat/lib`: download from (credits: <http://stackoverflow.com/tags/jstl/info>):

```
wget -O /usr/share/tomcat/lib/javax.servlet.jsp.jstl-api-1.2.1.jar
'http://search.maven.org/remotecontent?filepath=javax/servlet/jsp/jstl
/javax.servlet.jsp.jstl-api/1.2.1/javax.servlet.jsp.jstl-api-
1.2.1.jar'

wget -O /usr/share/tomcat/lib/javax.servlet.jsp.jstl-1.2.1.jar
'http://search.maven.org/remotecontent?filepath=org/glassfish/web/java
x.servlet.jsp.jstl/1.2.1/javax.servlet.jsp.jstl-1.2.1.jar'
```

6. Configure Apache

Apache needs to be configured to:

- Listen on ports 443 and 8443
- To make Apache listen at ports 443 and 8443,

Create `/etc/httpd/conf.d/ports.conf` with:

```
Listen 443
Listen 8443 https
```

- Make a backup of your `ssl.conf`

```
cd /etc/httpd/conf.d/
cp ssl.conf ssl.conf.dist
```

- Edit your `ssl.conf` and:

Delete the whole default `<VirtualHost>` section from `ssl.conf` (the definitions in `idpMain.conf` will be used instead).

Delete the entry for **Listen 443** (because we now have the directive in `ports.conf`).

Next, add a `VirtualHost` configuration for the IdP by creating a file named `/etc/httpd/conf.d/idpMain.conf` and populating it with the following contents:

```
<VirtualHost *:80>
ServerName idp3.hec.gov.pk:80

RedirectMatch 301 (.*) https://idp3.hec.gov.pk$1
RewriteEngine on
RewriteCond %{SERVER_NAME} =idp3.hec.gov.pk
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI}
[END,NE,R=permanent]
</VirtualHost>

<VirtualHost _default_:443>
# General setup for the virtual host, inherited from global
configuration
#DocumentRoot "/var/www/html"

ServerName idp3.hec.gov.pk:443

# Use separate log files for the SSL virtual host; note that LogLevel
# is not inherited from httpd.conf.
ErrorLog logs/ssl_error_log
TransferLog logs/ssl_access_log
```

```
LogLevel warn
SSLEngine on
# NOTE: We recomend disabling TLS 1.0 and TLS 1.1, using only TLS 1.2:
SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
# If you MUST support legacy outdated clients:
#Android<=4.3, IE<=10, Java <=6u45 or 7u25, Safari 5 or 6, or openssl
<=0.9.8y
# use instead (commented out):
#SSLProtocol all -SSLv2 -SSLv3
SSLCipherSuite
HIGH:MEDIUM:!aNULL:!MD5:!SEED:!IDEA:!RC4:!LOW:!3DES:!kRSA
SSLHonorCipherOrder on
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
#SSLCertificateChainFile /etc/pki/tls/certs/server-chain.crt

Header always set "Strict-Transport-Security" "max-age=31536000"

# Only needed if doing REMOTE_USER Auth
<Location /idp/Authn/RemoteUser>
# block access to RemoteUser authentication - use forms instead
order deny,allow
deny from all
#AuthType Basic
#AuthName "Please Login"
#AuthzLDAPAuthoritative Off
#AuthBasicProvider ldap
#AuthLDAPURL ldap://ldap.example.org/ou=People,dc=example,dc=org?uid
#Require valid-user
</Location>

ProxyRequests Off

<Proxy ajp://localhost:8009>
Allow from all
</Proxy>

ProxyPass /idp ajp://localhost:8009/idp retry=5

<Files ~ "\.(cgi|shtml|phtml|php3?)$" >
    SSLOptions +StdEnvVars
</Files>

<Directory "/var/www/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>

SetEnvIf User-Agent ".*MSIE.*" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0

CustomLog logs/ssl_request_log \
    "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\"
    %b"

</VirtualHost>
```

```

<VirtualHost _default_:8443>

    # General setup for the virtual host, inherited from
    global configuration
    #DocumentRoot "/var/www/html"
    ServerName idp3.hec.gov.pk:8443

    # Use separate log files for the SSL virtual host; note
    that LogLevel
    # is not inherited from httpd.conf.
    ErrorLog logs/ssl_error_log
    TransferLog logs/ssl_access_log
    LogLevel warn

    SSLEngine on
    SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
    SSLCipherSuite
HIGH:MEDIUM:!aNULL:!MD5:!SEED:!IDEA:!RC4:!LOW:!3DES:!kRSA
    SSLHonorCipherOrder on
    SSLCertificateFile /opt/shibboleth-idp/credentials/idp-
backchannel.crt
    SSLCertificateKeyFile /opt/shibboleth-
idp/credentials/idp-backchannel.key
    SSLVerifyClient optional_no_ca
    SSLVerifyDepth 10

    ProxyRequests Off

    <Proxy ajp://localhost:8009>
        Allow from all
    </Proxy>

    ProxyPass /idp ajp://localhost:8009/idp retry=5

    <Files ~ "\.(cgi|shtml|phtml|php3?)$" >
        SSLOptions +StdEnvVars
    </Files>
    <Directory "/var/www/cgi-bin">
        SSLOptions +StdEnvVars
    </Directory>

    SetEnvIf User-Agent ".*MSIE.*" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0
    CustomLog logs/ssl_request_log \
        "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\"
%b"

    SSLOptions -StdEnvVars +ExportCertData

</VirtualHost>

```

7. Testing

IdP Website

[https://\[IdP URL\]/idp](https://[IdP URL]/idp)

IdP Status

[https://\[IdP URL\]/idp/status](https://[IdP URL]/idp/status)

IdP MetaData

[https://\[IdP URL\]/idp/shibboleth](https://[IdP URL]/idp/shibboleth)

IdP Login URL

[https://\[IdP URL\]/idp/profile/Shibboleth/SSO](https://[IdP URL]/idp/profile/Shibboleth/SSO)

IdP Logout URL

[https://\[IdP URL\]/idp/profile/Logout](https://[IdP URL]/idp/profile/Logout)

8. Login form customization

In version 3 of the Shibboleth IdP, the default mechanism for rendering the login and error pages has been changed from JSP to [Velocity](#) templates, with the benefit that modifications to the login page do no longer require a restart of the Servlet container (Tomcat), but become active immediately after the editing the respective `.vm` file.

Download `pkifed-template.zip` and follow the below instructions.

[pkifed-login-template.rar](#)

Please keep the following `.vm` files in `/opt/shibboleth-idp/views` directory.

[login.vm](#)

[error.vm](#)

[logout.vm](#)

Please keep the following `main.css` files in `/opt/shibboleth-idp/edit-webapp/css` directory.

[main.css](#)

- **Logo configuration:**

Configure the following parameters in `/opt/shibboleth-idp/system/messages/messages.properties`.

Insert the new parameter `idp.logo.target.url` since it is not included in the default configuration.

- `idp.logo = /images/logo.png`
- `idp.logo.alt-text = Friendly name of your organization`
- `idp.logo.target.url = http://www.yourdomain.org`

please keep the following images at location `/opt/shibboleth-idp/edit-webapp/images/`

[pkifedlogo](#) [logo](#)

You also need to place a suitable PNG image with your organization's logo under the location referenced by `idp.logo`, i.e. `/opt/shibboleth-idp/edit-webapp/images/logo.png`, with a maximum suggested size of 200 by 100 pixel.

- Configure three target link parameters:

Add three lines which define the target for the *Forgot your password?*, *Change your password* and *Need Help?* links:

- `idp.login.forgotPassword.url = https://support.yourdomain.org/password-assistance`
- `idp.login.needHelp.url = https://support.yourdomain.org/help`
- `idp.login.changePassword.url = https://support.yourdomain.org/password-assistance`

Rebuild the Shibboleth Idp war file:

```
/opt/shibboleth-idp/bin/build.sh
```

9. SSL Configuration

Step 1 — Installing the Required Software

Enable the EPEL repository first by typing:

- `sudo yum install epel-release`

Now that you have access to the extra repository, install all of the required packages by typing:

- `sudo yum install httpd mod_ssl python-certbot-apache`

Step 2 — Configuring Access to Apache

Before we can request a certificate, we need to make sure Apache is running on our server and is accessible to the outside world.

To make sure Apache is up and running, type:

- `sudo systemctl start httpd`

Verify that Apache is running by checking the status of the service:

- `systemctl status httpd`

Step 3 — Requesting an SSL Certificate from Let's Encrypt

To execute the interactive installation and obtain a certificate that covers only a single domain, run the `certbot` command with:

- `sudo certbot --apache -d idp.example.org`

Step 4 — Setting Up Auto Renewal

To trigger the renewal process for all installed domains, you should run:

- `sudo certbot renew`

A practical way to ensure your certificates will not get outdated is to create a cron job that will periodically execute the automatic renewal command for you. Since the renewal first checks for the expiration date and only executes the renewal if the certificate is less than 30 days away from expiration, it is safe to create a cron job that runs every week or even every day, for instance.

Let's edit the crontab to create a new job that will run the renewal command every day. To edit the crontab for the root user, run:

- `sudo crontab -e`

Include the following content, all in one line:

```
crontab
.
.
.
30 2 * * * /usr/bin/certbot renew >> /var/log/le-renew.log
```

Save and exit. This will create a new cron job that will execute the `certbot renew` command every day at 2:30 am. The output produced by the command will be piped to a log file located at `/var/log/le-renew.log`. Since the command checks the expiration on the certificates locally, this does not stress the Let's Encrypt infrastructure, but allows you to update your certificates when they are within 30 days of expiring. Apache will be automatically restarted if any certificates are renewed.

10. References

1. <https://tuakiri.ac.nz/confluence/display/Tuakiri/Installing+a+Shibboleth+3.x+IdP#Installinga+Shibboleth3.xIdP-Testing>
2. <https://parichay.inflibnet.ac.in/idp.php>
3. <https://www.digitalocean.com/community/tutorials/how-to-secure-apache-with-let-s-encrypt-on-centos-7>
4. <https://www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-using-firewalld-on-centos-7>