

LinuxInstall

Shibboleth can be built on most versions of 32- and 64-bit Linux, but is officially supported only on the following distributions at this time, using the RPM packages available from the project's official mirrors:

- Red Hat Enterprise and CentOS 6, 7 (packages only provided for 7.4+)
- SUSE Linux Enterprise Server 12SP3, 12SP4

See the [SystemRequirements](#) page for more detail on this.

The official packages are built for, and integrate with, only the Apache (httpd) package that is supplied with the OS. When building from source or SRPM, you can accommodate any version of Apache (or its derivations) that is compatible.



RHEL 6+. Centos 6+

[Information about Red Hat Enterprise 6+ and CentOS 6+](#)



SELinux

Information about [SELinux](#)

If your distribution supports the RPM package manager, it is strongly suggested that you install using RPMs built for your specific distribution, or by rebuilding the SRPM source packages provided. If this is not possible, you can build from source.



Use the Correct RPM or not at all

Under no circumstances should you attempt to install a set of RPM packages built for/with a different OS or version from your own. This will usually lead to unpredictable problems and support issues. Instead, just rebuild the SRPM packages and make sure you have a repeatable process to incorporate security patch updates promptly.

- [Install from RPM](#)
- [Build from SRPM](#)
- [Build from Source](#)

Upgrades

Upgrading to new releases is handled automatically when RPMs are used, or in the case of a source build when "make install" is run using the same installation prefix. The system prevents configuration files from being overwritten and skips "initial install" tasks like generating keys. In the case of RPM upgrades, services should be appropriately restarted, but with source builds that step has to be done manually.

Initial Testing

You can test to ensure that the SP is running properly and the surrounding environment is correct by accessing

`https://localhost/Shibboleth.sso/Status`

from the actual web server machine itself. You **MUST** use "localhost" as the hostname or it WILL NOT WORK by default. If this test is successful, then the software is ready for further configuration.



You can also access the Status handler from other clients or using a non-localhost name, but **only** if you change the `ac1` parameter in the configuration to permit your client address or remove it entirely to open up access to anybody. The ACL is present by default because the Status handler can return some arguably sensitive information about your configuration.

Now you can progress to the [GettingStarted](#) material. Once you've actually configured the SP with its own settings and metadata from at least one IdP, in order to check that the SP is "working":

1. Protect a directory by requiring a Shibboleth session. Often, this is already done by default for the location "/secure".
2. Next, you typically place a script inside the protected directory that dumps the web server environment. With PHP for example you could in the easiest case just place a script there with the following:

```
<?php print_r($_SERVER) ?>
```

For an IIS deployment, an example ASP.NET script:

```
<% @ Page Language="C#" %>
<%
Response.Write("<h3>Server Variables</h3>");
foreach (string var in Request.ServerVariables)
{
    Response.Write(var + " = " + Request[var] + "<br>");
}
%>
```

3. Make sure that the Shibboleth-supplied variables are present. If there is a non-empty variable called **Shib-Application-ID**, you successfully authenticated and have a valid session. However, you also should check if there are other non-empty Shibboleth variables defined in the **attribute-map.xml** file. If there are no variables like **mail** or **givenName** or **surname**, the IdP probably released no attributes. In either case, have a look at the *shibd.log* and *transaction.log* files.