

ISBN :

## < 목 차 >

# Shibboleth Service Provider - OpenConext 연동 및 관리 매뉴얼

1 개요 및 구성도 .....	3
1.1 기술 개요 .....	3
1.2 시스템 구성도 .....	3
1.3 설치 S/W 목록 .....	3
1.4 H/W 사양 .....	4
2. Shibboleth SP 서버 구축 .....	6
2.1 서버 인증서 및 키 생성 .....	6
2.2 Shibboleth SP 설치 .....	9
2.3 Service Provider 설정 .....	13
3. OpenConext VM 서버 구축 및 Shibboleth SP 설정 .....	18
3.1 OpenConext VM 설치 .....	18
3.2 SP 등록하기 .....	20
4. OpenConext 연동 테스트 .....	26
4.1 SP의 서비스 페이지 작성 .....	26
4.2 SP-OpenConext-Mujina IdP 연동 .....	26
5. 참고 문헌 및 사이트 .....	29
5.1 SSL 설치 및 자가서명 인증서 설치 참조 .....	29
5.2 SP 설치 참조 .....	29
5.3 OpenConext 설치 참조 .....	29

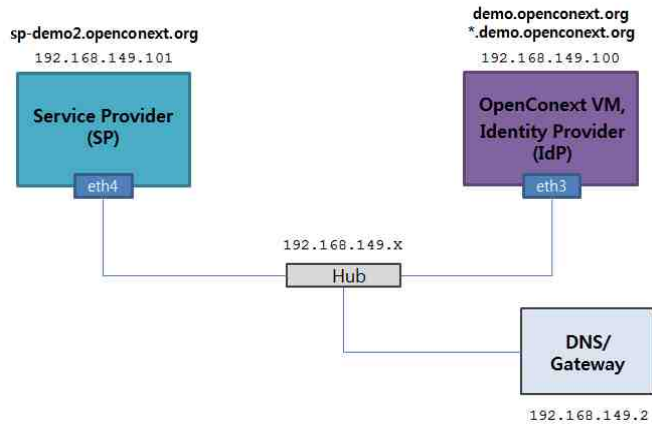
일 자 : 2014년 12월 8일  
부 서 : 첨단연구망센터/첨단연구망개발팀  
작성 자 : 장 희 진

# 1 개요 및 구성도

## 1.1 개요

Federation 기능은 사용자가 서비스 공급자(Service Provider, SP)의 웹서비스를 이용 시 소속된 기관 (Institution) 혹은 신원공급처(Identity Provider, IdP)의 익숙한 인증절차를 거쳐 로그인할 수 있도록 하는 편리한 서비스이다. 본 보고서에서는 오픈소스인 Shibboleth 프로그램을 이용하여 서비스 공급자(SP)와 Federation 기능을 제공하는 OpenConext VM, Mujina IdP (OpenConext VM에 built-in 된 Identity Provider)를 설치하고 각 구성요소 간 연동 방법에 대해 기술한다.

## 1.2 시스템 구성도



[그림 1] 테스트베드 전체 구성도

## 1.3 설치 S/W 목록

S/W 이름	SP	IdP	OC	설치 버전
Shibboleth SP	v			Shibboleth SP 2.5.3
OpenSSL	v	v	v	OpenSSL-1.0.1*

OpenConext VM			v	OpenConext-vm 74.4
Apache	v	v	v	Apache 2.2.15*
PHP	v	v	v	PHP 5.3.3*
CentOS Vmware 이미지	v	v	v	CentOS 6.2 VMware 이미지
JAVA	v	v	v	OpenJDK 1.7.0_65*
VMWare Workstation 또는 VMWare Player	v	v	v	VMWare Workstation 9.0.2

[표 1] 설치 S/W 목록

※ \* 표시는 OpenConext VM S/W에 포함되어 함께 설치되는 프로그램 목록

## 1.4 H/W 사양

### 1.4.1 H/W 사양

H/W	사양
Memory	16.0GB Dual-Channel DDR3 @ 802MHz
Processor	Intel Core i7 2700K @ 3.50GHz
Hard Disk	119GB SAMSUNG SSD 931GB Western Digital WDC
NIC	Intel(R) 82579V Gigabit Network Connection
OS	Windows 7 Professional K SP1 (64-bit)
Motherboard	ASUSTeK COMPUTER INC. P8Z77-V (LGA1155)
Graphics	1023MB NVIDIA GeForce GTX 560
Optical Drives	HL-DT-ST BDDVDRW CH10LS28

[표 2] 컴퓨터 H/W 사양

## 1.4.2

## H/W 사양

H/W	사양
Memory	2 G
Processor	1
Hard Disk	60 GB
Network Adapter	NAT
OS	CentOS 6.2 (kernel v2.6.32)

[표 3] 가상머신 H/W 사양

## 2

## Shibboleth SP 서버 구축

### 2.1 SP 인증서 및 키 생성

#### 2.1.1 Apache SSL 모듈 설치

SSL 암호화를 위해서는 OpenSSL과 mod\_ssl이 필요하다. SSL 설치 방법은 아파치 설치 시 함께 설치할 수도 있고, 기존에 아파치가 설치된 상태이면 별도로 추가적으로 설치할 수 있다.

※ 시스템 설정권한을 얻기 위해 root로 로그인 후 설치를 진행한다.

##### 1. SSL 설치 확인

- 아래 명령어로 mod\_ssl.c 파일이 있는지 확인한다.

```
# httpd -l
```

```
# apachectl -l
```

##### 2. 아파치 설치 시 SSL 설치

- 컴파일 시 옵션을 아래와 같이 설정 후 설치한다.

```
# ./configure --prefix=/home/apache2.2 \  
--enable-module=so \  
--enable-so \  
--enable-mods-shared=ssl \  
--with-ssl=/usr/local/ssl \  
--enable-ssl=shared  
  
# make  
# make install
```

configure 진행 중 openssl이 미설치 에러메시지가 출력되는 경우에는 아래와 같이 openssl-devel 데몬을 설치 후 컴파일 과정부터 다시 수행한다.

```

/* yum 으로 설치 */
# yum install -y openssl-devel

/* 소스로 직접 설치 */
# wget -c http://www.openssl.org/source/openssl-x.y.z.tar.gz
# tar xvfz openssl-x.y.z.tar.gz
# ./configure shared
# make
# make install

```

### 3. SSL 모듈을 추가 설치하는 경우

- 기존에 아파치가 설치된 경우라면 아래 명령어로 SSL 모듈을 추가 설치한다.

```

# yum install mod_ssl

/* openssl 설치 필요시 */
# yum install openssl

```

- modules/mod\_ssl.o 파일이 성공적으로 생성되었는지 확인한다.

```

# ls /usr/lib64/httpd/modules/

```

## 2.1.2 SP 자기서명된 인증서 (Self-signed Certificate) 및 개인키 생성

### 1. openssl 설치 확인

```

# rpm -qa openssl

```

### 2. 개인키 생성

```

# openssl genrsa -des3 -out server.key 2048

```

### 3. 인증서요청서 생성

- 진행 중 질문항목에 대해 적절히 입력한다.

```

# openssl req -new -key server.key -out server.csr

```

### 4. 개인키에서 암호 제거

- 개인키에 암호가 포함된 경우 아파치 구동 시마다 암호를 물으므로 암호를 제거한다. 암호가 제거된 파일은 용량이 약간 줄어든다.

```

# cp server.key server.key.org
# openssl rsa -in server.key.org -out server.key

```

### 5. 인증서 생성

- 개인키와 인증요청서로부터 인증서를 생성한다.

```

# openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt

```

### 6. 개인키와 인증서 설치

- 개인키와 인증서를 아래 위치에 복사한다.

```

# cp server.key /etc/ssl/certs/
# cp server.crt /etc/ssl/private/

```

## 2.1.3 Apache에 인증정보 설정

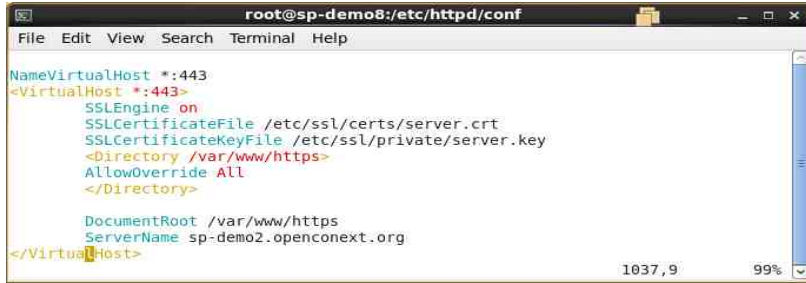
### 1. 아파치 데몬 설정 파일 설정

- 생성된 인증서와 키의 위치를 아파치 데몬 설정파일에 기재한다.

```

# vi /etc/httpd/conf/httpd.conf

```



[그림 2] SSL의 httpd.conf 설정

## 2. 아파치 SSL 설정파일 설정

```
# vi /httpd/conf.d/ssl.conf
```

### ssl.conf

```
SSLCertificateFile /etc/ssl/certs/server.crt
SSLCertificateKeyFile /etc/ssl/private/server.key
```

## 3. 아파치 재시작

```
# service httpd restart
```

## 2.2 Shibboleth SP

### 2.2.1 요구사항

- Shibboleth SP는 JAVA, PHP 등 다양한 언어를 지원하나 본 보고서에서는 PHP와 연동하는 방법에 대해 기술하므로 PHP를 설치한다.
- Apache 등 PHP 실행이 가능한 웹서버 설치되어 있어야 한다.

### 2.2.2 Shibboleth SP 설치

Shibboleth SP를 설치하는 방법에는 여러 가지 방법이 있는데 본 보고서에서는 yum 명령어를 이용해서 설치하는 방법에 대해 기술한다.

## 1. Repository 추가

- Shibboleth SP 소스를 다운로드 받을 수 있도록 Repository를 추가한다.

```
# cd /etc/yum.repos.d
```

```
# vi shibboleth.repo
```

### shibboleth.repo

```
[security_shibboleth]
name=Shibboleth (RHEL_6)
type=rpm-md
baseurl=http://download.opensuse.org/repositories/security:/shibboleth/RHEL_6/
gpgcheck=1
gpgkey=http://download.opensuse.org/repositories/security:/shibboleth/RHEL_6/repo
data/repomd.xml.key
enabled=1
```

## 2. Shibboleth RPM 설치

- 운영체제 환경에 따라 32-bit OS 버전 또는 64-bit OS 버전을 설치한다.

### 32-bit OS 인 경우

```
# yum install shibboleth
```

### 64-bit OS 인 경우

```
# yum install shibboleth.86_64
```

참고로 프로그램 소스를 다운로드받아 직접 설치할 경우 아래 사이트를 참고한다.

- 아래 사이트에서 Shibboleth SP 소스코드의 최신버전을 확인하고 다운로드 받는다.

```
https://shibboleth.net/downloads/service-provider/latest/
```

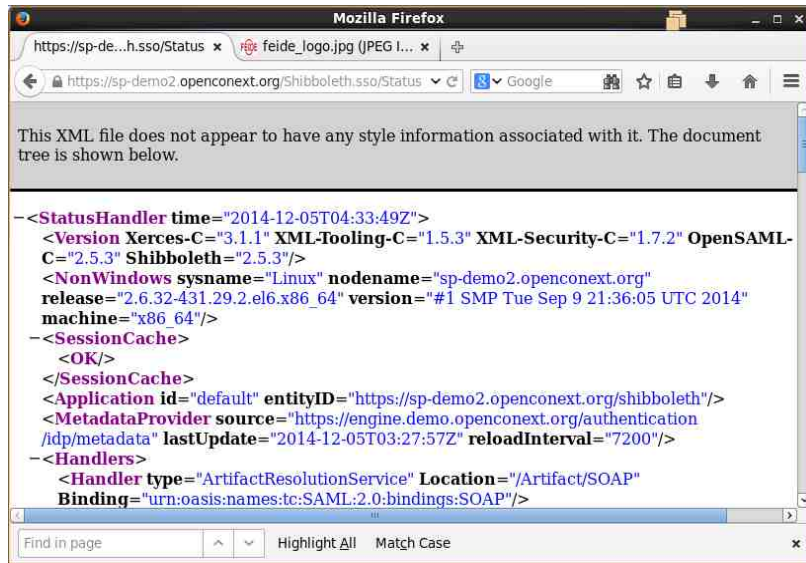
- 이 후 소스코드를 이용해 설치하는 과정은 아래 사이트를 참고한다.

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

### 3. 설치 및 접근 확인

- 성공적으로 설치되고 접근권한에도 문제가 없는 경우 아래 사이트에 접속하면 Shibboleth status 페이지가 XML 형태로 표시된다.

<https://sp-demo2.openconext.org/Shibboleth.sso/Status>



[그림 3] Shibboleth Status 페이지

- 403 접근권한 거부 오류가 나면 아래와 같이 로그메시지를 통해 확인가능하다.

```
# vi /var/log/httpd/native.log
```



[그림 4] 허용되지 않은 IP 차단 에러 로그

- 접속하는 source IP가 차단된 경우 아래와 같이 source IP를 허용하도록 설정해야 한다.

```
# vi /etc/shibboleth/shibboleth2.xml
```

```
<Handler type="Status" Location="/Status" ac1="127.0.0.1 add.your.ip.address"/>
```



[그림 5] 접근 IP 허용 설정

### 4. Shibboleth 데몬 재구동

- Shibboleth 데몬을 재구동한다.

```
# /etc/init.d/shibd restart
```

## 2.3 Service Provider

### 2.3.1 및 키 생성

#### 1. 인증서 및 키 생성

- SSL을 통해 IdP, 즉 여기서는 OpenConext 측으로 전송하는 request/response 메시지 및 metadata에 서명 시 사용할 인증서 및 키를 생성한다.

```
# openssl req -newkey rsa:4096 -new -x509 -days 365 -nodes -text -out shib.crt  
-keyout shib.key  
  
# chmod 644 /etc/shibboleth/shib.key
```

### 2.3.2 SP 기본정보 설정

SP 기본정보를 아래파일의 다음 항목에 기입한다.

```
# vi /etc/shibboleth/shibboleth2.xml
```

#### 1. entityID 설정

```
shibboleth2.xml  
  
<ApplicationDefaults entityID=https://sp-demo2.openconext.org/shibboleth  
REMOTE_USER="eppn persistent-id targeted-id">
```

#### 2. 키 및 인증서 정보 설정

- 생성된 키와 인증서 정보를 기재한다.

```
shibboleth2.xml  
  
<CredentialResolver type="File" key="shib.key" certificate="shib.crt"/>
```

#### 3. OpenConext metadata 설정

- Shibboleth SP가 OpenConext metadata를 획득할 수 있는 URL을 기재한다.

```
shibboleth2.xml  
  
<MetadataProvider type="XML"  
    uri="https://engine.demo.openconext.org/authentication/idp/metadata"  
    backingFilePath="metadata-openconext.xml"  
    reloadInterval="7200">  
<MetadataFilter type="RequireValidUntil" maxValidityInterval="172800"/>  
</MetadataProvider>
```

#### 4. OpenConext SSO(Single Sign On) 설정

- SSO 정보를 설정한다.

```
shibboleth2.xml  
  
<SSO entityID="https://engine.demo.openconext.org/authentication/idp/metadata">  
SAML2</SSO>
```

#### 5. Shibboleth 데몬 재구동

- Shibboleth 데몬을 재구동한다.

```
# /etc/init.d/shibd restart
```



[그림 6] OpenConext IdP metadata 로드 성공 시 로그표시

```

root@sp-demo8:/etc/shibboleth
File Edit View Search Terminal Help

<ApplicationDefaults entityID="https://sp-demo2.openconext.org/shibboleth"
  REMOTE_USER="epmn persistent-id targeted-id">
  <Sessions lifetime="28800" timeout="3600" relayState="ss-mem"
    checkAddress="false" handlerSSL="false" cookieProps="http">

    <SSO entityID="https://engine.demo.openconext.org/authentication/idp/metadata">SAML2</SSO>

    <!-- SAML and local-only logout. -->
    <Logout>SAML2 Local</Logout>

    <!-- Extension service that generates "approximate" metadata based on SP configuration. -->
    <Handler type="MetadataGenerator" Location="/Metadata" signing="false"/>

    <!-- Status reporting service. -->
    <Handler type="Status" Location="/Status" acl="127.0.0.1 192.168.149.101 ::1"/>

    <!-- Session diagnostic service. -->
    <Handler type="Session" Location="/Session" showAttributeValues="false"/>

    <!-- JSON feed of discovery information. -->
    <Handler type="DiscoveryFeed" Location="/DiscoFeed"/>
  </Sessions>

  <Errors supportContact="root@localhost"
    helpLocation="/about.html"
    styleSheet="/shibboleth-sp/main.css"/>

  <MetadataProvider type="XML" uri="https://engine.demo.openconext.org/authentication/idp/metadata"
    backingFilePath="metadata-openconext.xml" reloadInterval="7200">
    <MetadataFilter type="RequireValidUntil" maxValidityInterval="172800"/>
  </MetadataProvider>

  <!-- Map to extract attributes from SAML assertions. -->
  <AttributeExtractor type="XML" validate="true" reloadChanges="false" path="attribute-map.xml"/>

  <!-- Use a SAML query if no attributes are supplied during SSO. -->
  <AttributeResolver type="Query" subjectMatch="true"/>

  <!-- Default filtering policy for recognized attributes, lets other data pass. -->
  <AttributeFilter type="XML" validate="true" path="attribute-policy.xml"/>

  <!-- Simple file-based resolver for using a single keypair. -->
  <CredentialResolver type="File" key="shib.key" certificate="shib.crt"/>
</ApplicationDefaults>

```

[그림 7] shibboleth2.xml 설정 사례

```

httpd.conf

<Location /secure>

    AuthType shibboleth

    ShibRequestSetting requireSession 1

    require valid-user

</Location>

```

```

root@sp-demo8:/etc (on sp-demo8.openconext.org)
File Edit View Search Terminal Help

#
#   Alias /simplesaml /var/simplesamlphp/www
#   ErrorLog logs/dummy-host.example.com-error_log
#   CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>

NameVirtualHost *:443
<VirtualHost *:443>
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/server.crt
    SSLCertificateKeyFile /etc/ssl/private/server.key
    <Directory /var/www/https>
        AllowOverride All
    </Directory>

    <Location /secure>
        AuthType shibboleth
        ShibRequestSetting requireSession 1
        require valid-user
    </Location>
    DocumentRoot /var/www/https
    ServerName sp-demo2.openconext.org
</VirtualHost>

"/etc/httpd/conf/httpd.conf" 1043L, 35202C          1037,2-9      Bot

```

[그림 8] httpd.conf에 federation 서비스가 필요한 SP 페이지 주소 설정

### 2.3.3 Apache

#### 1. httpd 설정

- 아파치 설정파일에 인증 federation 서비스 (Shibboleth SP - OpenConext VM 연동)가 적용될 URI 경로를 설정한다.

```
# vi /etc/init.d/conf.d/httpd.conf
```

#### 2. 아파치 데몬 재시작

```
# service httpd restart
```

#### 3. SP 페이지 접속

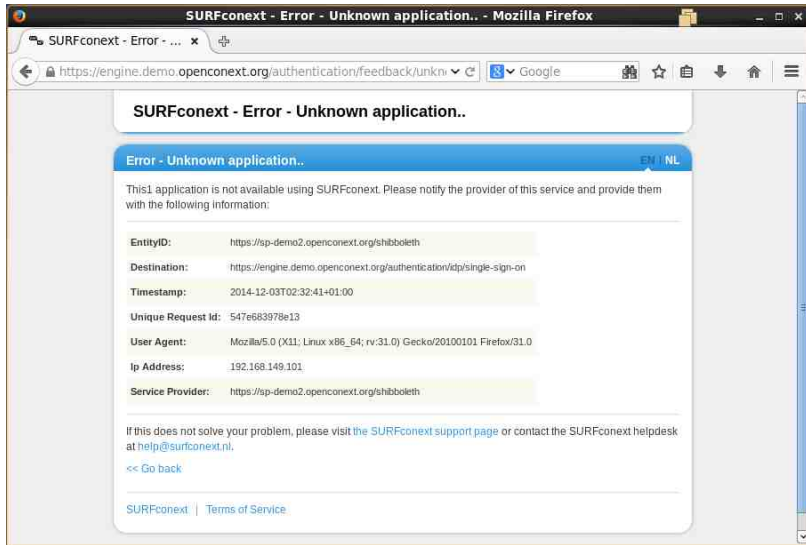
- federation 서비스가 설정된 SP 페이지를 접속한다.

```
http://sp-demo2.openconext.org/secure/
```

- Shibboleth SP가 SP측에서 성공적으로 설정되어 OpenConext와 연동되나 아직 SP가 OpenConext



에 등록되지 않는 경우 아래와 같은 화면이 출력된다.



[그림 9] OpenConext 연동화면

### 3 OpenConext VM 서버 구축 및 SP 설정

#### 3.1 OpenConext VM

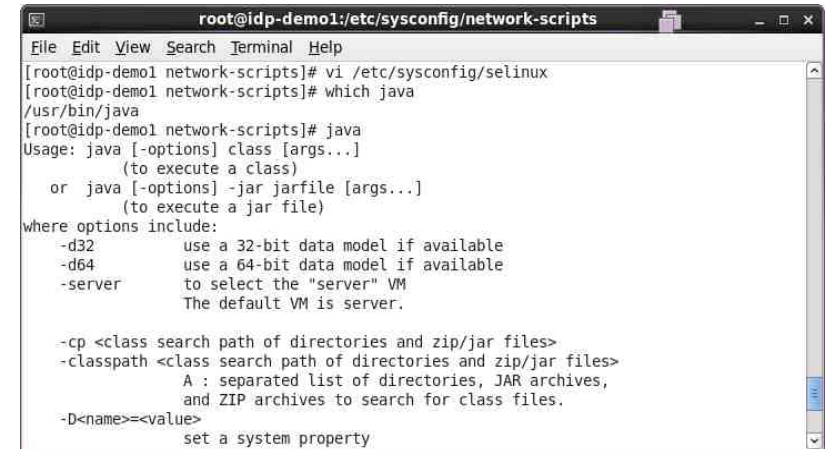
##### 3.1.1 요구사항

1. OpenConext VM 설치 요구사항

- 리눅스 가상머신으로 CentOS 6 또는 RedHat EL 6 이상 설치

- JDK 1.4 이상 설치 권고

※ **JAVA\_HOME** 변수가 적절히 설정되어 아래와 같이 **java** 명령 실행이 가능해야 한다.



[그림 10] java 명령어 실행 화면

##### 3.1.2 OpenConext VM 다운로드 및 설치

1. 설치파일 다운로드

- 아래 명령어를 사용해 OpenConext VM 설치파일을 다운로드 받고 압축을 해제한다.

```
# curl https://codeload.github.com/OpenConext/OpenConext-vm/tar.gz/v74.4 | tar zx
```

2. OpenConext VM 설정변경

- oc\_config.sh 파일 원본을 아래와 같이 복사해 두고, 스크립트 내의 코멘트를 참조해 OpenConext-vm-75/scripts/oc\_config.sh 파일을 가상머신에 맞게 설정한다.

```
# install -d /etc/openconext
# cp OpenConext-vm-75/scripts/oc_config.sh /etc/openconext/.
```

- 이번 설치과정에서는 기본설정을 유지하였다.

※ host name은 기본적으로 [componentname].demo.openconext.org으로 설정된다. 변경을 원하는 경우 초기 설치 시 설정파일에서 변경한다.

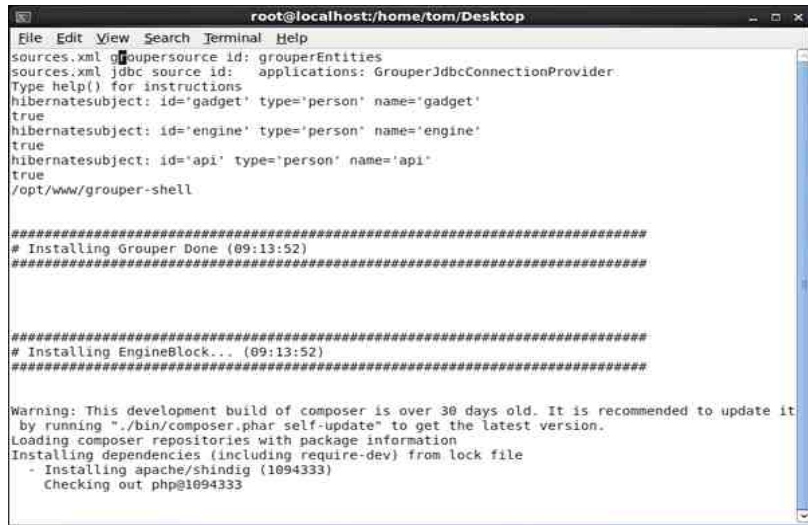
### 3. 설치

- 설치 스크립트를 실행하여 설치를 진행한다.

```
# bash OpenConext-vm-74.4/scripts/install_openconext.sh
```

※ 설치 과정 중 중간에 멈춘 듯 보여도 기다리면 설치가 진행되니 인내를 가지고 진행한다.

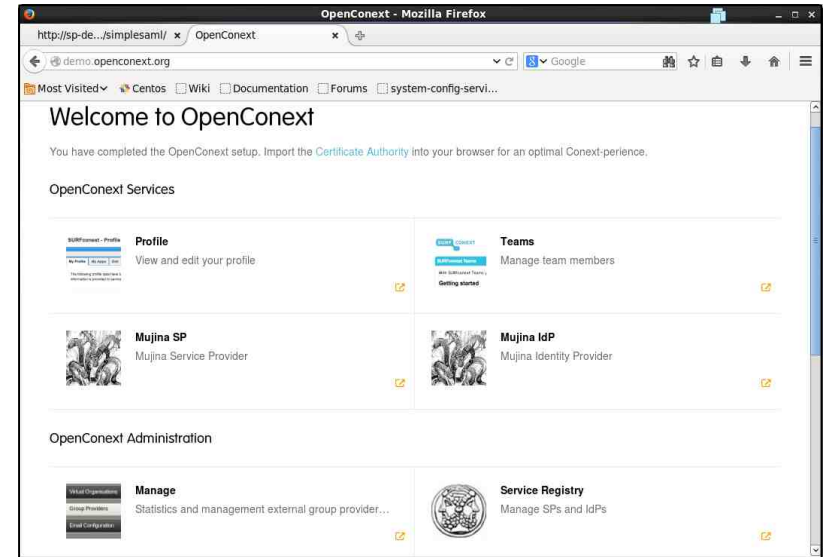
※ Github OpenConext VM 설치 매뉴얼(<https://github.com/OpenConext/OpenConext-vm/>)에 따르면 30분 정도 소요된다고 언급되어있으나 실제 설치 시 2시간 정도 소요되었다.



[그림 11] OpenConext VM 설치 화면

### 4. 구동 확인

- 브라우저를 실행해서 OpenConext 웹사이트를 확인한다.



[그림 12] OpenConext VM 구동 화면

### 5. 각 컴포넌트 실행

- 사이트 내 각 메뉴는 SP로부터 공급되는 서비스처럼 동작하므로 로그인하려면 OpenConext에 기본적으로 설치된 (build-in) Mujina IdP로부터 인증 절차를 거쳐야한다. Mujina IdP를 선택하고 아래 인증 정보를 이용해 로그인 한다.

```
User : admin
Password : secret
```

## 3.2 SP

### 3.2.1 Connection

#### 1. Service Registry 로그인

- Mujina IdP(admin/secret)로 로그인한다.

## 2. SP Connection 생성

- 'Connection' 탭에서 'Create connection' 클릭 후 Connection ID를 입력한다.

```
https://sp-demo2.openconext.org/shibboleth
```

- 'SAML 2.0 SP'를 선택한다.

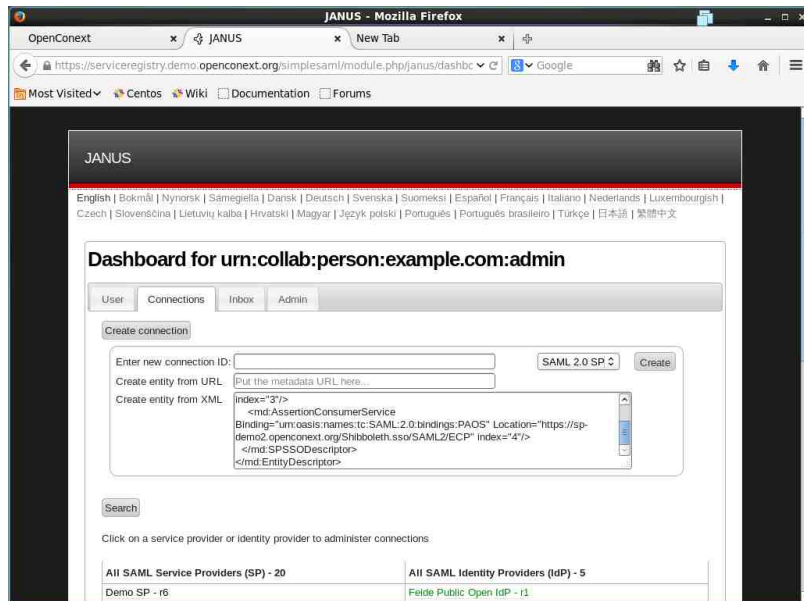
### 3.2.2 SP Metadata

#### 1. SP metadata 설정

- 아래 Shibboleth SP metadata URL 주소로부터 SP metadata를 다운받는다.

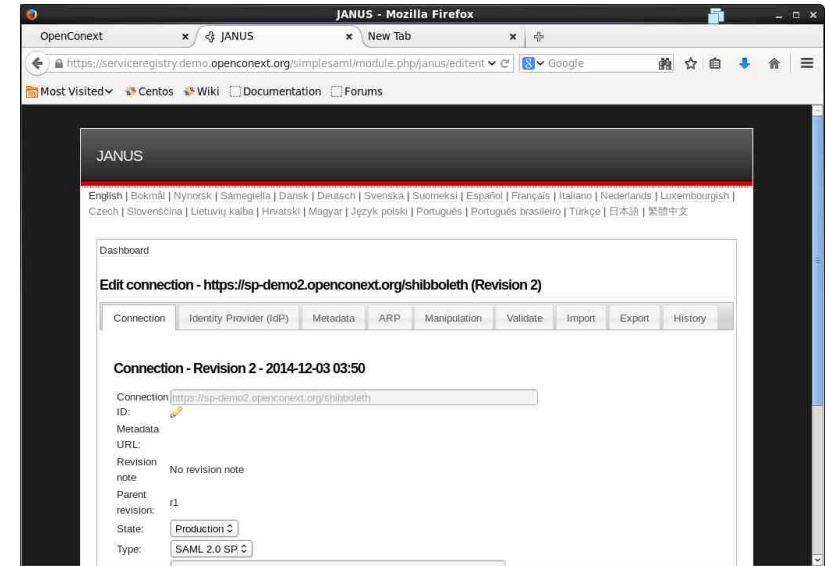
```
https://sp-demo2.openconext.org/Shibboleth.sso/Metadata
```

- SP metadata를 'create entity from XML'에 기입하고 'Create' 버튼을 클릭한다.



[그림 13] SP Connection 생성 화면 1

- Connection 탭에서 'State'를 'Production'으로 설정한다



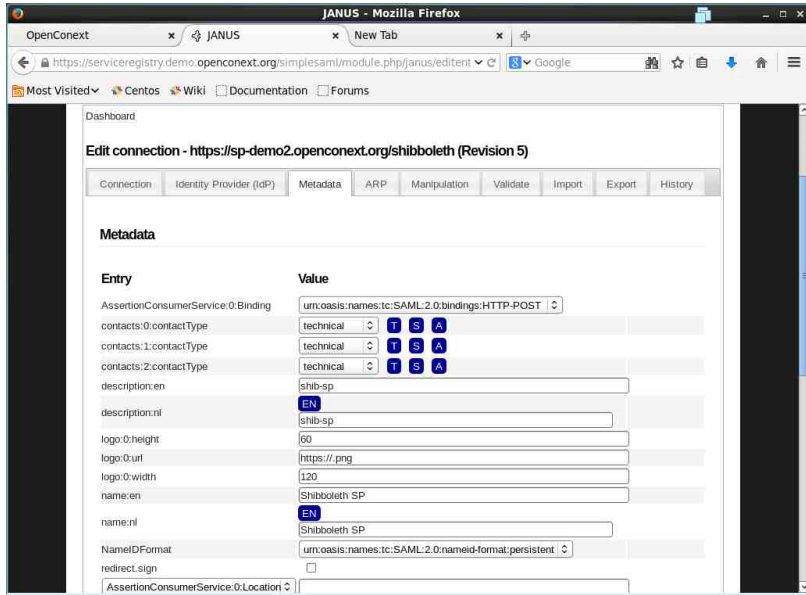
[그림 14] SP Connection 생성 화면 2

#### 2. SP metadata 수정

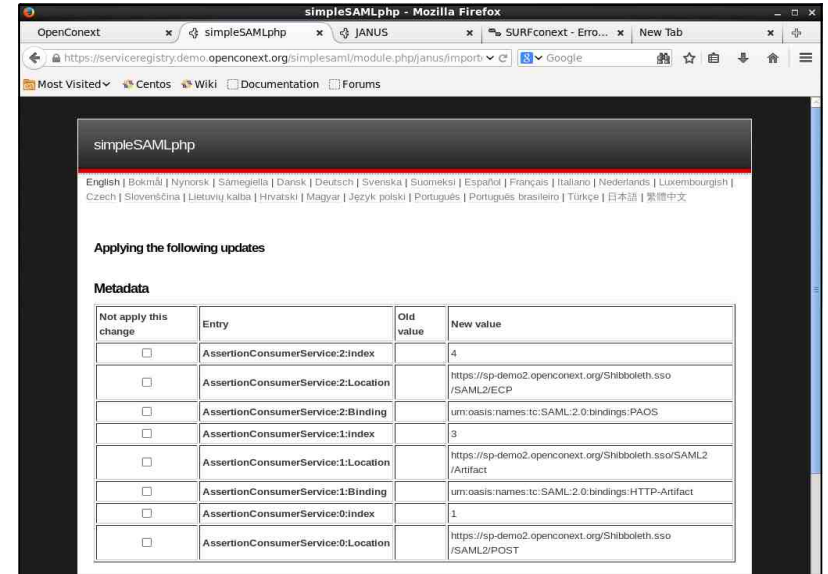
- 1의 과정 후 metadata 탭을 클릭하여 metadata를 확인한다.

\* metadata가 connection 생성 시 제대로 생성되지 않으면 URL로부터 import하여 다시 metadata를 얻어온다. 이 경우 'Import from URL' 항목에 SP metadata URL 주소를 입력하여 SP metadata를 읽어 들여야한다.

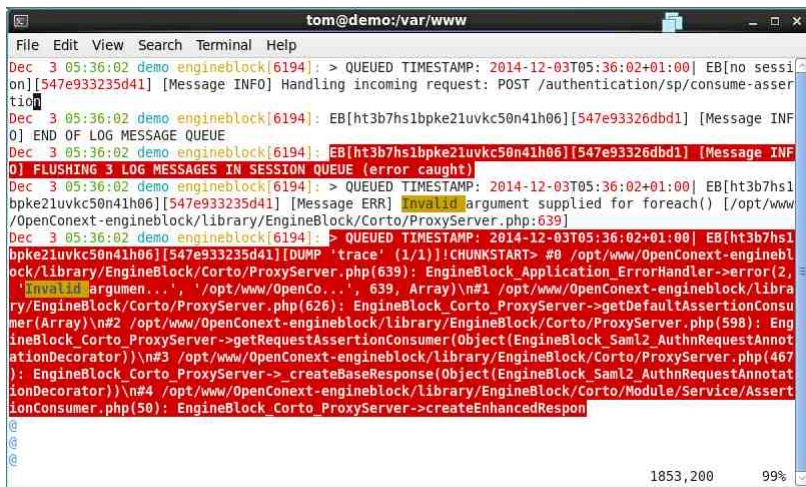
```
https://sp-demo2.openconext.org/Shibboleth.sso/Metadata
```



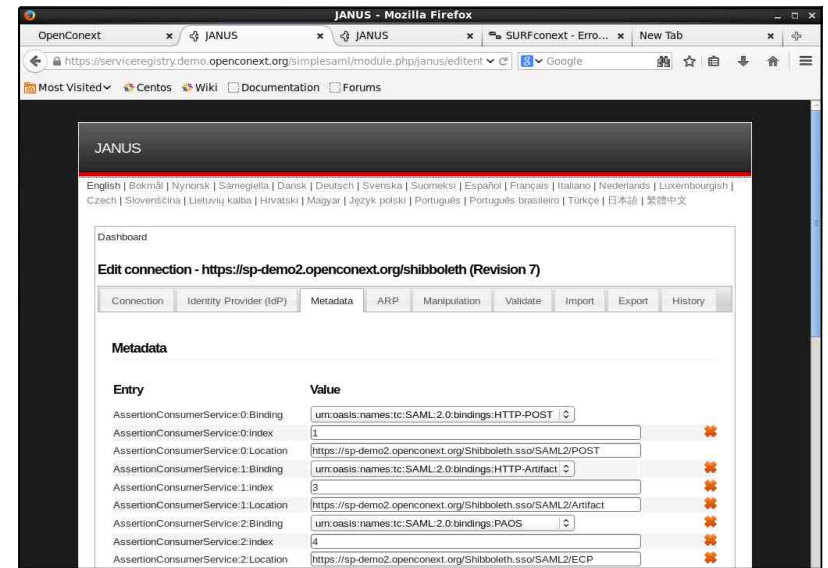
[그림 15] SP metadata import 로드 실패 시 ServiceRegistry 화면



[그림 17] SP metadata import 화면



[그림 16] SP metadata import 로드 실패 시 OpenConext 에러 로그



[그림 18] SP metadata 표시 화면

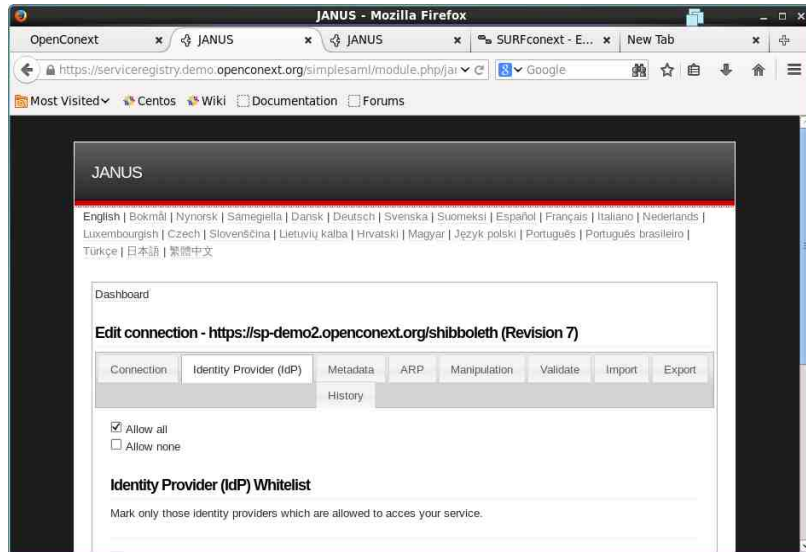
## 2. name 정보 설정

- Metadata 탭에서 'name:en' 및 'name:nl' 항목을 'Shibboleth SP'로 기재하고 save 한다. 이후부터 Dashboard의 SP list에 name(en)으로 표시된다.

### 3.2.3 정보 설정

#### 1. Identity Provider 설정

- Identity Provider 탭에서 기본적인 설정('Allow All')을 선택하여 모든 IdP를 허용한다. (현재 설치에서는 기본설정 유지)



[그림 19] Identity Provider 설정

#### 2. Manipulation 설정

- Manipulation 메뉴는 SP에 공개되는 사용자의 Attribute 정보를 UI를 통해 수정/삭제/추가할 수 있는 기능이다. (현재 설치에서는 기본설정 유지)

#### 3. 기타 설정

- 기타 설정은 OpenConext 설치사이트 메뉴얼을 참조하여 적절히 설정한다. (현재 설치에서는 기본 설정을 유지)

## 4 OpenConext 연동 테스트

### 4.1 SP 서비스 페이지 작성

1. 아래와 같이 SP 웹페이지 방문 시 처음 게시되는 대표 페이지(index.php)를 작성한다.

```
# vi /var/www/https/secure/index.php
```

#### index.php

```
<html>
<head><title>Shibboleth test</title></head>
<body><pre><?php print_r($_SERVER); ?></pre></body>
</html>
```

- 출력되는 속성정보는 아래 파일에 설정된 정보이다.

```
# vi /etc/shibboleth/attribute-map.xml
```

### 4.2 SP-OpenConext-Mujina IdP 연동

#### 1. SP 접속

- sp-demo2.openconext.org에 접속한다.

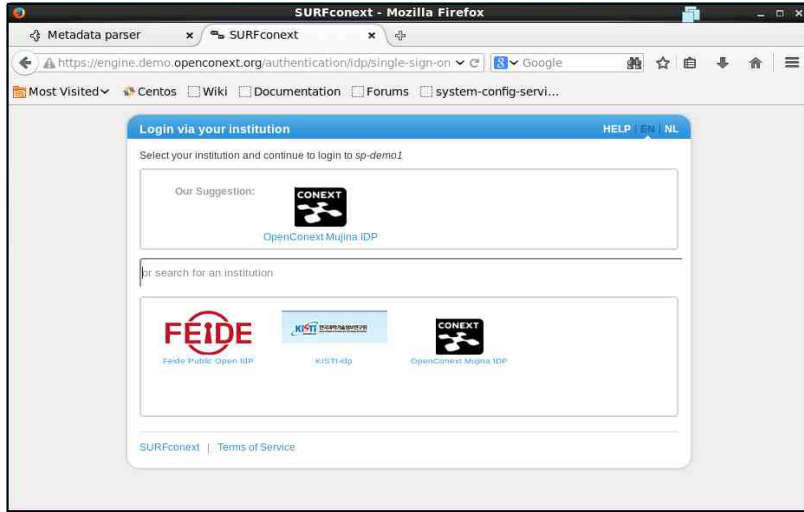
```
http://sp-demo2.openconext.org/secure/
```

#### 2. Mujina IdP 로그인

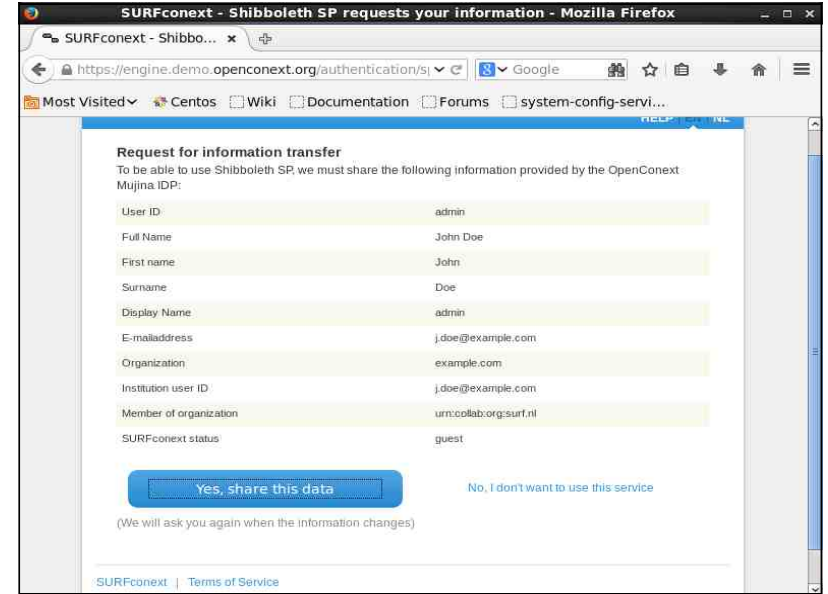
- Mujina IdP에 아래 사용자 정보로 로그인 한다.

```
User : admin
Password : secret
```

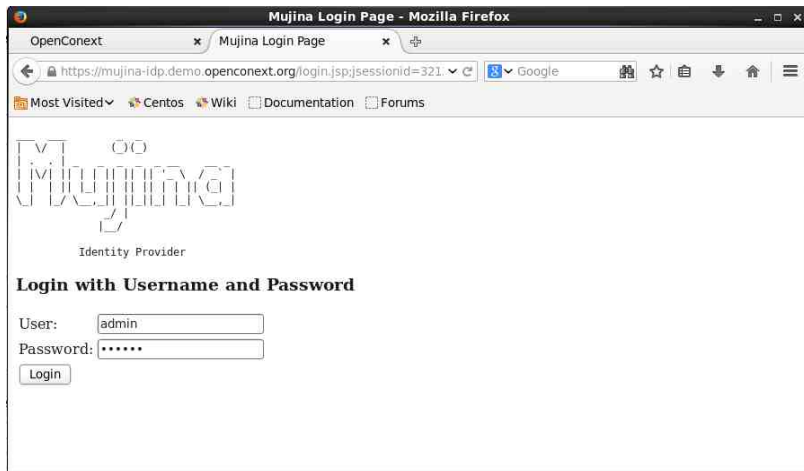




[그림 21] OpenConext 연동화면



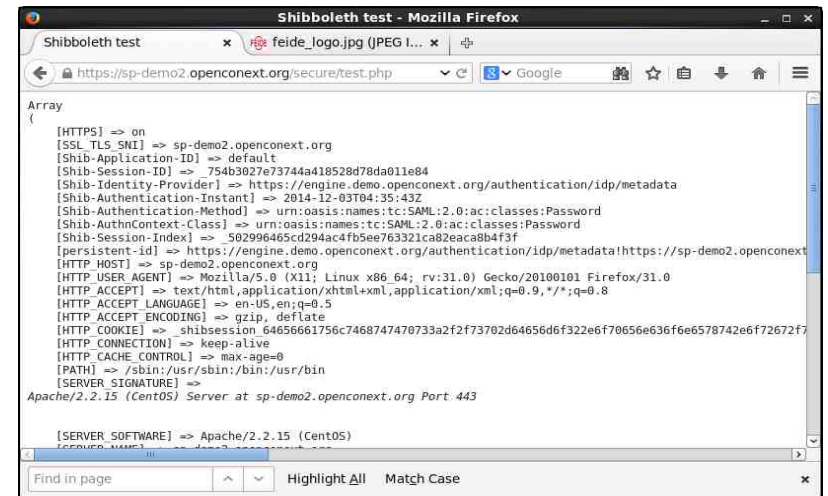
[그림 23] 사용자 첫 로그인 시 attribute 전송 요청 승인 화면



[그림 22] Mujina IdP 로그인

3. 사용자가 첫 로그인 시 아래와 같이 IdP 측에 저장된 사용자 정보를 SP 제공에 동의를 요청하는 화면이 표시된다.

4. 아래와 같이 SP의 서비스페이지가 표시된다.



[그림 24] OpenConext 연동 및 인증 성공 후 SP 표시 화면

## 5 참고 문헌 및 사이트

### 5.1 SSL 및 자가서명 인증서 설치 참조

- **Setting up an SSL secured Webserver with CentOS**

<http://wiki.centos.org/HowTos/Https>

- **CentOS 6.4 SSL 구축 문서**

<http://starkapin.tistory.com/501>

- **리눅스 개인서명 SSL 인증서 생성 가이드**

[http://jmnote.com/wiki/%EC%82%AC%EC%84%A4\\_SSL\\_%EC%9D%B8%EC%A6%9D%EC%84%9C\\_%EC%83%9D%EC%84%B1](http://jmnote.com/wiki/%EC%82%AC%EC%84%A4_SSL_%EC%9D%B8%EC%A6%9D%EC%84%9C_%EC%83%9D%EC%84%B1)

### 5.2 Shibboleth SP 설치 참조

- **OpenConext Shibboleth SP Installation**

<https://wiki.surfnet.nl/display/OpenConext/OpenConext+SP+-+Shibboleth>

- **Shibboleth NativeSPLinuxRPM Installation**

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

### 5.3 OpenConext 설치 참조

- **OpenConext wiki 사이트**

<https://wiki.surfnet.nl/display/OpenConext/Installing+an+OpenConext+%28VM%29+instance+for+development>

- **OpenConext Github Project 사이트**

<https://github.com/OpenConext/OpenConext-vm/>