

# WildFly - Session Clustering 구축

## 1. Wildfly node 2개 생성

<https://linuxtechlab.com/creating-wildfly-cluster-linux/>

```
[ec2-user@ip-172-31-19-53 ~]$ cd /opt/wildfly
[ec2-user@ip-172-31-19-53 wildfly]$ ls
appclient      docs           LICENSE.txt    standalone
bin            domain        modules       welcome-content
copyright.txt  jboss-modules.jar  README.txt
```

```
$ cd /opt/wildfly
```

```
$ sudo mkdir standalone2
```

```
$ sudo cp -rv standalone/* standalone2/
```

```
$ sudo mv standalone standalone1
```

```
[ec2-user@ip-172-31-19-53 wildfly]$ ls
appclient      docs           LICENSE.txt    standalone1
bin            domain        modules       standalone2
copyright.txt  jboss-modules.jar  README.txt    welcome-content
```

## 2. Make script1.sh file

```
[ec2-user@ip-172-31-19-53 bin]$ pwd
/opt/wildfly/bin
[ec2-user@ip-172-31-19-53 bin]$ sudo vim script1.sh
```

```
#!/bin/sh
./standalone.sh -Djboss.server.base.dir=/opt/wildfly-
16.0.0.Final/standalone1 -Djboss.node.name=node0
```

- 생성한 script1.sh 파일에 위 박스의 내용을 넣는다.

**script1.sh** : Wildfly instance를 시작하는데 쓰이는 script.

**-Djboss.server.base.dir** : location with all instance configuration & deployed packages.

**-Djboss.node.name=node** : instance의 이름

위 두 설정 값은 상황마다 다르니 주의하자.

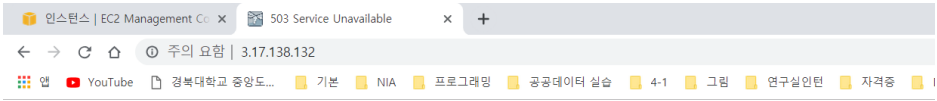
- 이제 WildFly를 **\$ ./script1.sh** 명령어를 사용해 실행시킬 수 있다.

(권한 관련 에러 발생 시 \$ sudo su 를 통해 root 계정(# ...)에서 실행시키면 가능하다.)

- Script1,2를 실행키시면 80번 포트로 접속 시 wildfly 화면이 뜬다.

Script 둘 중 하나를 중지시켜도 80번 포트로 접속 가능하며 둘 다 중지시키면

Service Unavailable 페이지가 나온다.



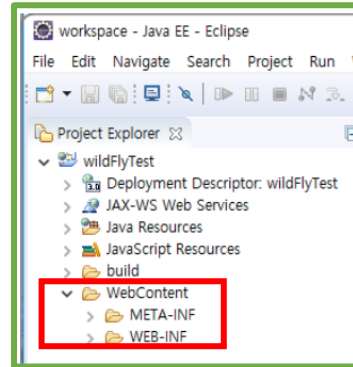
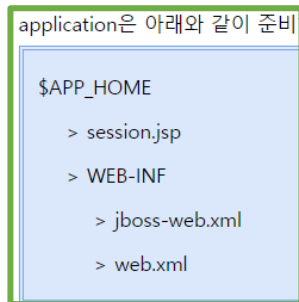
## Service Unavailable

The server is temporarily unable to service your request due to maintenance downtime or capacity problems. Please try again later.

# WildFly - Session Clustering Test

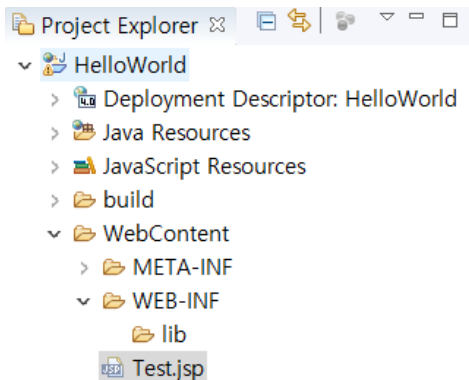
## 1. Make application project

<https://waspro.tistory.com/203>



- 이 파일이 보통 이렇게 웹 프로젝트 안에 위치함을 알게 되어 우선 웹 프로젝트를 생성하려 한다.
- 위의 **Web Application** 생성을 보고 프로젝트를 생성한다.
- 참고 : <https://gmlwjd9405.github.io/2018/10/29/web-application-structure.html>
- 나의 경우 **Eclipse IDE for Enterprise Java Developers**를 설치해 Web application을 만들고 그 프로젝트 파일을 통째로 AWS에 가져왔다.

<https://www.eclipse.org/downloads/packages/>



```
ec2-user@ip-172-31-19-53 WebContent]$ pwd
/home/ec2-user/APP_HOME/HelloWorld/WebContent
ec2-user@ip-172-31-19-53 WebContent]$ ls
META-INF  test.jsp  WEB-INF
ec2-user@ip-172-31-19-53 WebContent]$
```

## 2. WildFly에 Web Application 반영하기

<https://waspro.tistory.com/263?category=827007>

<https://doosiks-programming.blogspot.com/2017/09/wildfly.html>

<https://support.eapps.com/index.php?/Knowledgebase/Article/View/513/52/wildfly-8-application-deployment-in-standalone-mode>

- Standalone.xml 파일을 수정해 web application 프로젝트가 있는 경로를 연결한다.

```
[root@ip-172-31-19-53 configuration]# pwd
/opt/wildfly-16.0.0.Final/standalone/configuration
[root@ip-172-31-19-53 configuration]# vim standalone.xml
[root@ip-172-31-19-53 configuration]#
```

```
<subsystem xmlns="urn:jboss:domain:undertow:8.0" default-server="default-server" ...<생략>... >

    <buffer-cache name="default"/>

    <server name="default-server">

        <ajp-listener name="ajp" scheme="http" socket-binding="ajp"/>

        <http-listener name="default" socket-binding="http" redirect-socket="https" enable-http2="true"/>

        <https-listener name="https" socket-binding="https" security-realm="ApplicationRealm" enable-
http2="true"/>

        <host name="default-host" alias="localhost">

            <location name="/web" handler="WebContent"/>

            <http-invoker security-realm="ApplicationRealm"/>

        </host>

    </server>

    <servlet-container name="default">

        <jsp-config/>

        <websockets/>

    </servlet-container>

    <handlers>

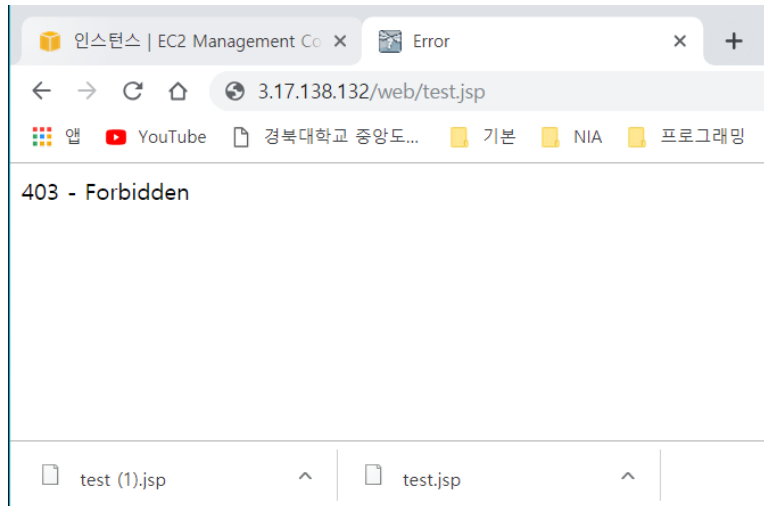
        <file name="WebContent" path="/home/ec2-user/APP_HOME/HelloWorld/WebContent"/>

    </handlers>

</subsystem>
```

### 3. 접속 확인

- <http://3.17.138.132/web/test.jsp> 에 접속한다.



현재는 다음과 같은 에러창이 뜨면서 test.jsp 파일이 다운받아진다.

Web application 프로젝트 경로를 찾을 수는 있다는 말이므로 이제 application 프로젝트가 제대로 실행만 되면 될 듯...