



| Date | Document | Author | Page |
|------------|-------------------------|--------|------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 1/34 |

Apache & Tomcat 설치 및 연동



| Date | Document | Author | Page |
|------------|-------------------------|--------|------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 2/34 |

Table of contents

| | | |
|----------|---|-----------|
| 1 | Introduction..... | 6 |
| 1.1 | System Construction | 6 |
| 2 | Java Development Kit (JDK)..... | 8 |
| 2.1 | JDK Install | 8 |
| 2.2 | JDK 환경 변수 | 8 |
| 3 | Apache Web Server..... | 10 |
| 3.1 | Apache MPM (Prefork, worker) | 10 |
| 3.2 | Forward Proxy & Reverse Proxy | 12 |
| 3.3 | Apache Install | 13 |
| 3.4 | mod_jk Install | 14 |
| 3.5 | cronolog install | 14 |
| 3.6 | httpd-modjk.conf..... | 15 |
| 3.7 | workers.properties..... | 15 |
| 3.8 | httpd.conf 설정 | 16 |
| 3.9 | SSL (Secure Socket Layer) 설정 | 18 |
| 3.10 | virtualhost 4가지 방법 | 19 |
| 3.11 | favicon.ico 설정..... | 20 |
| 3.12 | MaSiteInfo.ini | 20 |
| 3.13 | Apache 로그 파일에 image로그 빼기..... | 21 |
| 3.14 | mod_expires.so 모듈 설치 | 21 |
| 3.15 | 설치 후 확인 사항 | 22 |
| 4 | Tomcat WAS Server | 23 |
| 4.1 | Tomcat Construction | 23 |
| 4.2 | Tomcat Install | 24 |
| 4.3 | Native Library Install..... | 24 |
| 4.4 | Tomcat Accesslog 설정 | 25 |
| 4.5 | Tomcat JVM 메모리 설정..... | 25 |
| 4.6 | Tomcat JNDI 설정 | 26 |
| 4.7 | Tomcat thread pool 셋팅 | 27 |
| 4.8 | Catalina.sh 환경 변수 추가 및 catalina.out 로그 일자 별 분리..... | 27 |
| 4.9 | Tomcat Instance 생성 | 29 |
| 4.10 | 지시자 | 31 |



| Date | Document | Author | Page |
|------------|-------------------------|--------|------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 3/34 |

4.11 Hash DOS 공격 대비 32

5 Windows Apache & Tomcat 연동 32

5.1 Apache 설치 32

5.2 Tomcat 설치 33

5.3 Tomcat Connector 다운로드 33

5.4 Apache & Tomcat 연동 33



| Date | Document | Author | Page |
|------------|-------------------------|--------|------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 4/34 |

표 목차

| | |
|--|----|
| 표 1 Binary file name..... | 6 |
| 표 2 JDK install flow | 8 |
| 표 3 Apache MPM | 12 |
| 표 4 Apache install flow | 14 |
| 표 5 mod_jk install | 14 |
| 표 6 cronolog install | 15 |
| 표 7 httpd-modjk.conf..... | 15 |
| 표 8 workers.properties..... | 16 |
| 표 9 httpd.conf..... | 18 |
| 표 10 Image 로그 빼기..... | 21 |
| 표 11 mod_expires.so install..... | 22 |
| 표 12 Tomcat install flow..... | 24 |
| 표 13 Native library install flow | 25 |
| 표 14 Tomcat accesslog format..... | 25 |
| 표 15 JVM memory | 25 |
| 표 16 Tomcat JNDI..... | 26 |
| 표 17 Tomcat thread pool..... | 27 |
| 표 18 catalina.sh 환경 변수 추가..... | 28 |
| 표 19 catalina.log 일자 별 분리 | 29 |
| 표 20 Tomcat instance 생성 스크립트..... | 31 |
| 표 21 Tomcat instance 생성 디렉토리 구조..... | 31 |



| Date | Document | Author | Page |
|------------|-------------------------|--------|------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 5/34 |

그림 목차

| | |
|--------------------------------|----|
| 그림 1 System Construction | 7 |
| 그림 2 Tomcat Construction | 23 |



| Date | Document | Author | Page |
|------------|-------------------------|--------|------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 6/34 |

1 Introduction

해당 문서는 Java Development Kit 설치 및 Apache Web Server 와 Tomcat WAS Server 간의 연동 방법에 대해서 기술한다.

설치되는 서버의 환경은 Unix 기반의 Linux operating system이며, Install 되는 서버의 파일 버전은 다음의 표와 같다.

| <i>Binary file name</i> | <i>Binary version</i> |
|-------------------------|-------------------------------------|
| JDK | jdk-6u21-linux-i586.bin |
| Apache WEB Server | httpd-2.2.17.tar.gz |
| Tomcat WAS Server | apache-tomcat-6.0.29.tar.gz |
| Tomcat-connectors | tomcat-connectors-1.2.30-src.tar.gz |

표 1 Binary file name

1.1 System Construction

다음의 System Construction을 토대로 해당 기술문서를 작성한다.

Apache Server로의 HTTP Request 요청 시 header 정보의 도메인을 확인하여 어떤 Tomcat Instance에 게 요청 받은 HTTP Request를 전달할지에 대해서 판단한다.



| Date | Document | Author | Page |
|------------|-------------------------|--------|------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 7/34 |

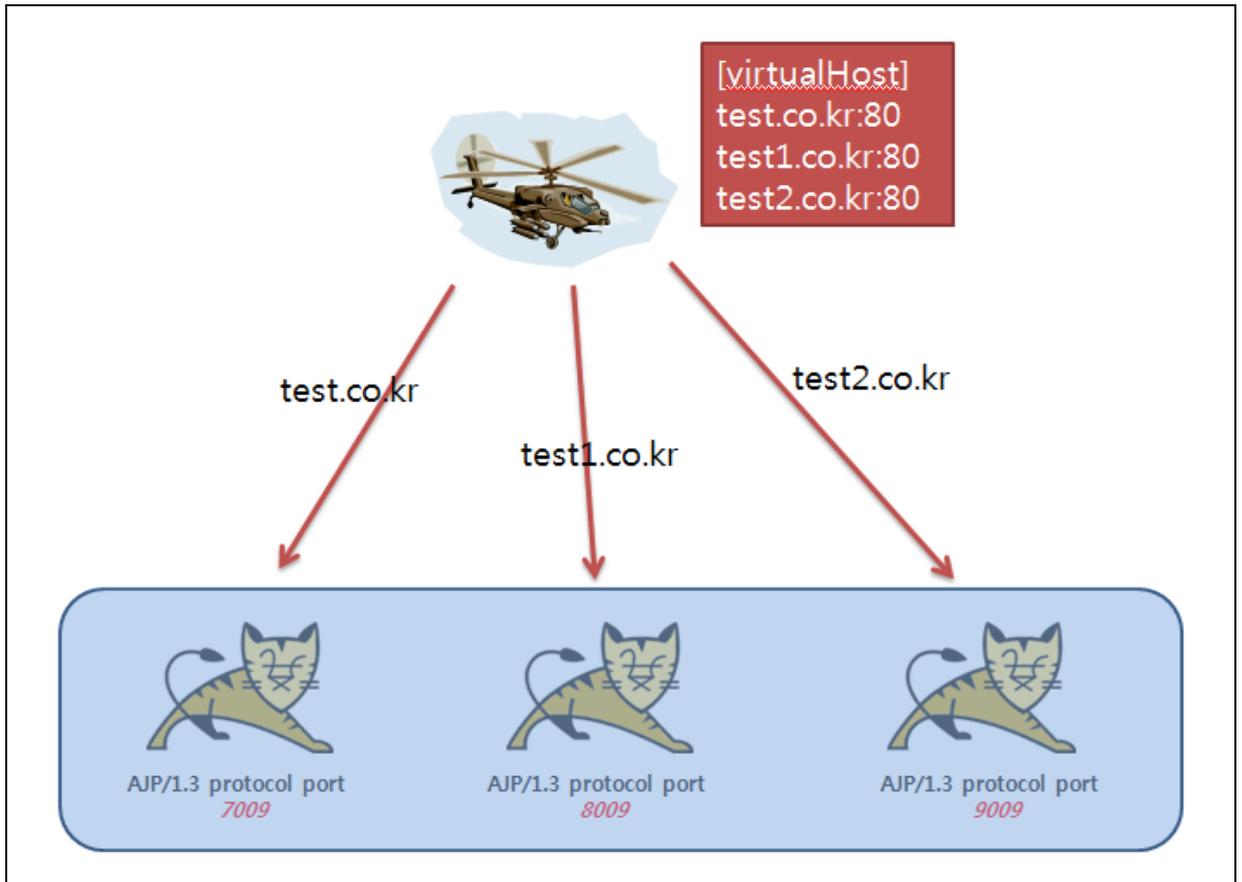


그림 1 System Construction



| Date | Document | Author | Page |
|------------|-------------------------|--------|------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 8/34 |

2 Java Development Kit (JDK)

JDK 는 Sun Microsystems 에서 나온 자바 소프트웨어 개발 환경이다.

자바 virtual machine(JVM)과 compiler, debugger, 그리고 java applet 및 응용프로그램 개발을 위한 도구들이 포함된다.

2.1 JDK Install

JDK 설치 시 몇 비트 운영체제의 리눅스를 사용하는지 먼저 확인을 해야 한다.

JDK binary 파일은 32bit 와 64bit 두 가지로 구분하여 다운로드를 제공하고 있기 때문이다..

root 계정으로 로그인 한 후, getconf LONG_BIT 명령어로 해당 운영체제가 몇 bit를 지원하는지 확인 한다.

JDK 설치 방법에 대해서는 아래의 설치 flow을 확인 한다.

| No. | Install flow |
|-----|---|
| 1 | root 계정으로 로그인 |
| 2 | getconf LONG_BIT 명령어로 몇 비트의 운영체제인지 확인 <pre>[root@imad1 imad]# getconf LONG_BIT 32</pre> |
| 3 | JDK 다운로드 <ul style="list-style-type: none"> - http://java.sun.com/javase/download/index.jsp 웹 페이지로 이동 - 최신 버전의 JDK 다운로드 링크 클릭 - “Platform” 아래 select box 가 있고, 설치하고자 하는 운영체제를 선택한다. - Continue 클릭 - 로그인 팝업 창이 나오게 되는데 계정이 존재하지 않는다면 하단 영역의 “Skip the Step” 버튼을 클릭 한다. - 다운로드 링크되어 있는 jdk-6u21-linux-i586.bin 파일을 다운로드 받는다. - 다운로드 받은 JDK bin 파일을 설치하려고 하는 리눅스 운영체제에 ftp upload 한다. |
| 4 | 설치 진행 <ul style="list-style-type: none"> - 설치할 directory에 해당 JDK bin 파일을 이동시켜 설치하도록 한다. (prefix 옵션을 이용하여 설치하고자 하는 directory를 지정할 수 있다.) - chmod 755 jdk-6u21-linux-i586.bin (파일 권한 설정) - ./jdk-6u21-linux-i586.bin (설치 진행) |

표 2 JDK install flow

2.2 JDK 환경 변수

JDK 환경 변수를 setting 하기에 앞서 해당 계정이 어떤 shell 을 사용하는지에 대해서 확인해야 한다.

확인 방법은 /etc/passwd 파일을 열어 해당 계정이 어떤 shell 을 사용하는지 확인하며 해당 기술 문서에서는 bash shell 을 사용한다고 가정하여 환경 변수를 추가하도록 하겠다.



| Date | Document | Author | Page |
|------------|-------------------------|--------|------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 9/34 |

추가할 계정의 home directory로 이동하여 .bash_profile 을 vi 명령어로 open 한 후, 아래의 환경 변수 정보를 추가 한다.

```
JAVA_HOME=/usr/local/java  
PATH=$JAVA_HOME/bin:$PATH  
export JAVA_HOME
```

export를 해주는 이유는 콘솔에서 다음과 같이 사용 가능하게 하기 위함
echo \$JAVA_HOME
/usr/local/java

위의 환경 변수 추가 후, .bash_profile 을 컴파일 한다.

```
source .bash_profile
```



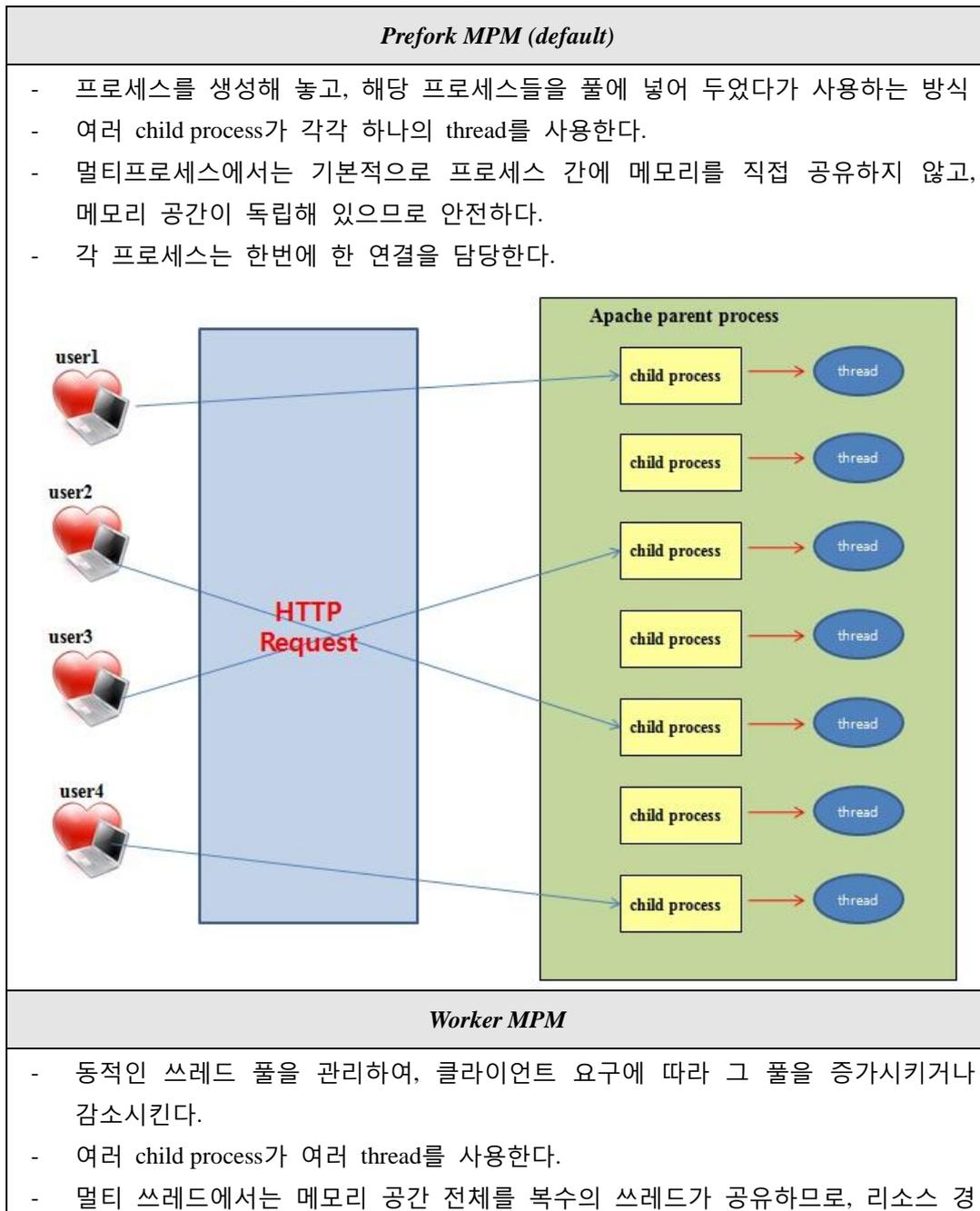
| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 10/34 |

3 Apache Web Server

Apache Web Server 는 웹 브라우저에서 요청하는 HTML 문서나 이미지, 파일 등을 HTTP(Hyper Text Transfer Protocol)을 통해 전송해주는 서비스 프로그램이다.

3.1 Apache MPM (Prefork, worker)

Apache가 받아들인 Request 요청을 처리 하기 위해 child processes 에게 분배하는 방식 분배 방식으로는 Unix 계열에서 사용되는 Prefork, Worker 방식이 있다.

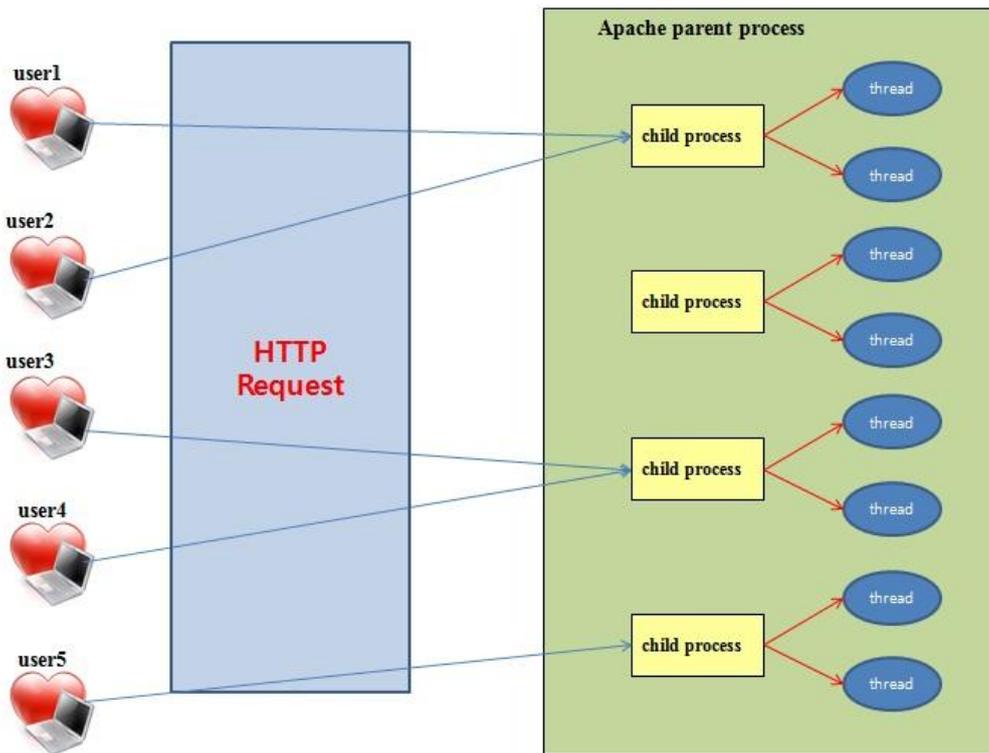




| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 11/34 |

합이 발생하지 않도록 주의할 필요가 있다. 이것이 멀티쓰레드 프로그래밍이 복잡하다고 하는 이유이다.

- 각 쓰레드는 한번에 한 연결을 담당한다.
- 하나의 프로세스 내에 복수의 쓰레드를 생성하고, 쓰레드 하나로 클라이언트 하나를 처리한다.



Install flow

[MPM 설치]

```
[kyu-qcdev@app/kyu/test:208] % ./configure --prefix=/home/test/apache2 --with-mpm=worker
```

아파치 설치 후, `./httpd -l` 명령어를 이용하여 선택한 MPM을 알 수 있다.

➔ `httpd` 실행 파일은 `APACHE_HOME/bin` 디렉토리에 있음.

[`httpd.conf` 수정]

```
#Include conf/extra/httpd-mpm.conf 주석 제거
```

[`httpd-mpm-conf` 수정]

```
<IfModule mpm_worker_module>
    StartServers      10
    MaxClients        2000
    MinSpareThreads   75
    MaxSpareThreads   250
    ThreadsPerChild   25
    MaxRequestsPerChild 10000
</IfModule>
```

MaxClients: 동시에 접속할 수 있는 클라이언트의 상한, 즉 프로세스 수 * 쓰레드 수



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 12/34 |

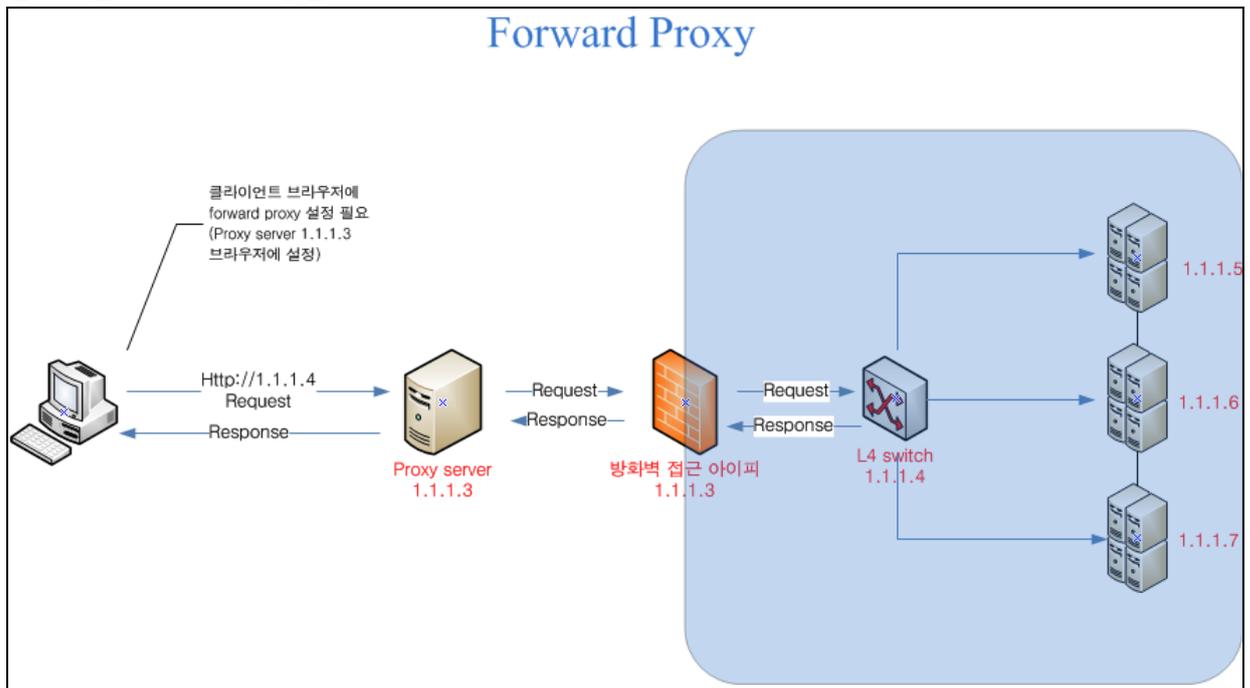
| | |
|---|------|
| ServerLimit: 프로세스 수의 상한 | |
| ThreadLimit: 프로세스당 쓰레드 수의 상한 | |
| ThreadsPerChild: 프로세스당 쓰레드 수(ThreadLimit와 거의 같은 의미) | |
| MaxClients 4096 / ThreadsPerChild 128 = 32 프로세스 | |
| ServerLimit | 32 |
| ThreadLimit | 64 |
| MaxClients | 4096 |
| ThreadsPerChild | 64 |

☞ 3 Apache MPM

3.2 Forward Proxy & Reverse Proxy

Forward Proxy 서버는 클라이언트와 실제 서버 사이에 존재하는 중개 서버이다.

방화벽에 의해 제한된 서버에 Internet Access를 제공하며, cache 설정을 이용하여 network traffic을 감소시킬 목적으로 사용된다.

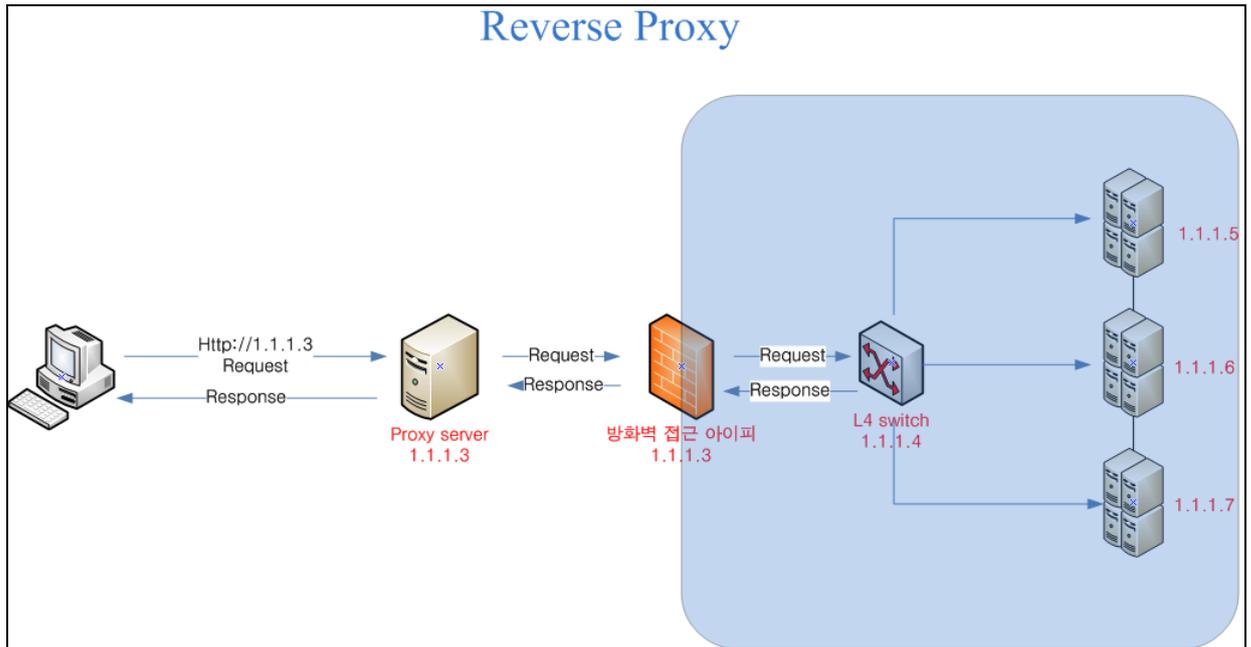


Reverse Proxy는 클라이언트 측면에서 볼 때 실제 서버처럼 나타남.

클라이언트에서 웹 브라우저 주소 창에 <http://1.1.1.3>을 치면 Proxy Server에게 요청이 되고, Proxy Server에서는 L4 switch에 전달 받은 요청 정보를 넘긴다.



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 13/34 |



3.3 Apache Install

아파치 설치하는 아래의 flow 와 같이 진행한다.

| No. | Install flow |
|-----|---|
| 1 | http://httpd.apache.org 웹 사이트에 접근 |
| 2 | 왼쪽 메뉴에 “Download!” 라벨이 보이며, 하단에 “from a mirror” 링크가 존재한다. 해당 링크를 클릭하여 서브 페이지로 이동 한다. |
| 3 | 리눅스에 아파치 웹 서버를 설치할 것이므로 Current Releases 의 2.2.17 (released 2010-**-**) 를 다운로드 한다. |
| 4 | 2.2.17 링크 부분에 마우스 커서를 올리고 마우스 오른쪽 버튼을 누른 후, 속성에 들어간다. “속성 (URL)” 메뉴의 HTTP URL을 복사하여 리눅스 command 창에 wget 명령어를 이용해 파일 다운로드 받는다. <pre>[moa-qcdev@/app/moa/test:205] % wget http://apache.tt.co.kr/httpd/httpd-2.2.17.tar.gz --15:11:01-- http://apache.tt.co.kr/httpd/httpd-2.2.17.tar.gz => `httpd-2.2.17.tar.gz' Resolving apache.tt.co.kr... 121.125.79.185 Connecting to apache.tt.co.kr[121.125.79.185]:80... connected. HTTP request sent, awaiting response... 200 OK Length: 6,597,991 (6.3M) [application/x-gzip] 100%[=====] 15:11:09 (11.08 MB/s) - `httpd-2.2.17.tar.gz' saved [6597991/6597991]</pre> |
| 5 | 압축을 풀어준다. (tar 명령어 이용) <pre>[kyu-qcdev@/app/kyu/test:208] % tar xvfz httpd-2.2.17.tar.gz</pre> |
| 6 | Configure 진행 압축 해제된 directory로 이동하여 아래의 명령어를 실행한다. --prefix 옵션은 설치 경로를 지정하며, --prefix 옵션을 사용하지 않는다면 default 로 /usr/local/ 경로에 설치된다. <pre>[kyu-qcdev@/app/kyu/test:208] % ./configure --prefix=/service/test/web/apache2 --with-mpm=worker --enable-module=ssl --with-ssl=/usr/bin/openssl --enable-ssl=shared --enable-rewrite</pre> |



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 14/34 |

| | |
|---|---|
| 7 | 컴파일 진행 [kyu-qcdev@/app/kyu/test/httpd-2.2.17:211] % make |
| 8 | Install 진행 [kyu-qcdev@/app/kyu/test/httpd-2.2.17:211] % make install |

표 4 Apache install flow

3.4 mod_jk Install

mod_jk 는 Apache 와 Tomcat 연동 모듈이다.

| No. | Install flow |
|-----|---|
| 1 | http://tomcat.apache.org/download-connectors.cgi 웹 페이지 이동 |
| 2 | archive download site link 클릭 |
| 3 | jk > source > jk-1.2.30 (해당 파일 URL 경로 복사) |
| 4 | 리눅스 command 라인에서 다음과 같이 wget 명령어 실행 wget http://archive.apache.org/dist/tomcat/tomcat-connectors/jk/source/jk-1.2.30/tomcat-connectors-1.2.30-src.tar.gz |
| 5 | 압축 풀기 tar xvzf tomcat-connectors-1.2.30-src.tar.gz |
| 6 | cd /home/kyu/tomcat-connectors-1.2.30-src/native 디렉토리로 이동 |
| 7 | ./configure --with-apxs=/home/kyu/apache2/bin/apxs apxs 옵션에는 apache home directory 의 bin/apxs 파일 경로를 지정한다. |
| 8 | make |
| 9 | make install |
| 10 | 컴파일이 성공적으로 진행된다면 mod_jk.so 파일이 Apache home directory의 /modules로 자동 복사 된다. |

표 5 mod_jk install

3.5 cronolog install

아파치의 accesslog 및 errorlog를 날짜 별로 생성 또는 폴더 별로 생성 가능

아파치 설치 후, 서버 기동 시 logs 디렉토리에 access_log 및 error_log 파일이 생성되는데 이 각각의 파일에는 HTTP Request 요청에 대한 로그 정보가 누적되어 몇 일 몇 달이 지나버리면 어마어마한 파일 용량을 가지게 된다.

파일 용량이 커지다 보면 log write 하는데 적지 않은 시간이 소요 되고, 이로 인하여 웹 서버 성능이 저하되는 현상이 발생하게 된다.

하나의 파일에 로그 누적을 방지하기 위해서는 날짜 별로 로그 파일을 분리해주는 cronolog를 설치하여야 한다.

(아파치에 내장되어 있는 bin/rotatelogs 유틸이 존재하지만 이보다 더 개선된 로그 유틸이 cronolog 이다.)

| No. | Install flow |
|-----|--|
| 1 | http://cronolog.org 해당 사이트로 이동하여 DOWNLOAD 링크를 클릭합니다. |
| 2 | 최신 버전의 cronolog 위에 마우스 커서를 올려 놓으신 후, 오른쪽 마우스 클릭하여 해당 파일에 대한 주소를 복사 |



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 15/34 |

| | |
|---|--|
| 3 | 리눅스 command 라인에 다음의 명령어 실행 \$ wget http://cronolog.org/download/cronolog-1.6.2.tar.gz |
| 4 | 압축을 풀어준다. \$ tar xvfz cronolog-1.6.2.tar.gz |
| 5 | Cronolog-1.6.2 디렉토리로 이동하여 다음의 명령어 실행 \$./configure --prefix=/home/test/cronolog \$ make \$ make install |
| 6 | httpd.conf 수정 ErrorLog "/home/test/cronolog/sbin/cronolog /home/test/apache2/logs/error_log.%Y-%m-%d" CustomLog "/home/test/cronolog/sbin/cronolog /home/test/apache2/logs/access_log.%Y-%m-%d" combined |
| 7 | 아파치 재 기동 후, 로그 디렉토리로 이동하여 access_log 및 error_log 파일에 날짜가 붙어있는지 확인 |

표 6 cronolog install

3.6 httpd-modjk.conf

tomcat connector 설정 파일이며 설치 시 APACHE_HOME/conf/extra directory에 파일을 생성한다. 파일 생성 후, 다음과 같이 설정 정보 추가

| <i>httpd-modjk.conf</i> |
|---|
| <pre><IfModule !mod_jk.c> LoadModule jk_module modules/mod_jk.so </IfModule> <IfModule mod_jk.c> JkWorkersFile conf/extra/workers.properties JkLogFile "/app/service/test/web/apache2.2.17/bin/rotatelogs /app/service/test/web/apache2.2.17/logs/mod_jklogs/modjk_log.%Y%m%d%H 86400" JkLogLevel error JkLogStampFormat "[%Y %a %b %d %H:%M:%S]" JKRequestLogFormat "%R %w %V %T %U" JkOptions +ForwardKeySize +ForwardURICompatUnparsed -ForwardDirectories JkShmFile logs/mod_jklogs/mod-jk.shm </IfModule></pre> |

표 7 httpd-modjk.conf

httpd-modjk.conf 파일 생성을 한 후, httpd.conf 파일에서 해당 파일을 include 시켜준다.
`Include conf/extra/httpd-modjk.conf`

3.7 workers.properties

APACHE_HOME/conf/extra 디렉토리에 workers.properties 파일을 생성한다. 파일 생성 후, 다음과 같이 설정 정보 추가

| <i>workers.properties</i> |
|---|
| <pre>worker.list=devajp13, manajp13, testajp13 worker.devajp13.port=7009 worker.devajp13.host=localhost worker.devajp13.type=ajp13 worker.devajp13.lbfactor=1 worker.devajp13.socket_timeout=0 worker.devajp13.socket_keepalive=true</pre> |



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 16/34 |

```

worker.devajp13.recovery_options=4
worker.devajp13.ping_mode=A
worker.devajp13.ping_timeout=10000
worker.devajp13.connection_pool_size=150
worker.devajp13.connection_pool_minsize=25
worker.devajp13.connection_pool_timeout=60

worker.manajp13.port=8009
worker.manajp13.host=localhost
worker.manajp13.type=ajp13
worker.manajp13.lbfactor=1
worker.manajp13.socket_timeout=0
worker.manajp13.socket_keepalive=true
worker.manajp13.recovery_options=4
worker.manajp13.ping_mode=A
worker.manajp13.ping_timeout=10000
worker.manajp13.connection_pool_size=150
worker.manajp13.connection_pool_minsize=25
worker.manajp13.connection_pool_timeout=60

worker.testajp13.port=9009
worker.testajp13.host=localhost
worker.testajp13.type=ajp13
worker.testajp13.lbfactor=1
worker.testajp13.socket_timeout=0
worker.testajp13.socket_keepalive=true
worker.testajp13.recovery_options=4
worker.testajp13.ping_mode=A
worker.testajp13.ping_timeout=10000
worker.testajp13.connection_pool_size=150
worker.testajp13.connection_pool_minsize=25
worker.testajp13.connection_pool_timeout=60

```

Tomcat Home directory의 server.xml 에서 AJP/1.3 protocol을 사용하는 포트들 workers.properties 파일에 명시되어 있는 port로 변경되어야지만 해당 port로 TCP/IP연동이 이루어진다.

**connection_pool_timeout=60 은 톰캣 server.xml의 connectionTimeout과 동기화를 시켜준다.
60이면 connectionTimeout=60000**

표 8 workers.properties

3.8 httpd.conf 설정

APACHE_HOME/conf 디렉토리에 httpd.conf 파일이 만들어져 있으며, 다음과 같이 설정 정보를 추가 한다.

| <i>httpd.conf</i> |
|--|
| <pre> NameVirtualHost *:80 #MAN <VirtualHost *:80> ServerName test.co.kr DocumentRoot /app/service/test/web/apache2.2.17/htdocs/test Alias /banner /app/service/test/content/banner <Directory /app/service/test/content> Options -Indexes -FollowSymLinks Order allow,deny Allow from all </Directory> </pre> |



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 17/34 |

```
CustomLog                "/app/service/test/web/apache2.2.17/bin/rotatelogs
/app/service/test/web/apache2.2.17/logs/accesslogs/man/man_access_log.%Y%m%d 86400" combined
env=!image-request

    JkMount /*.do manajp13
    JkMount /*.jsp manajp13
</VirtualHost>

#DEV
<VirtualHost *:80>
    ServerName test1.co.kr
    ServerAlias www.test1.co.kr test1.co.kr
    DocumentRoot /app/service/test/web/apache2.2.17/htdocs/test1
    Alias /banner /app/service/test/content/banner

    <Directory /app/service/test/content>
        Options -Indexes -FollowSymLinks
        Order allow,deny
        Allow from all
    </Directory>

    CustomLog                "/app/service/test/web/apache2.2.17/bin/rotatelogs
/app/service/test/web/apache2.2.17/logs/accesslogs/dev/dev_access_log.%Y%m%d 86400" combined
env=!image-request

    JkMount /*.do devajp13
    JkMount /*.jsp devajp13
</VirtualHost>

#TEST
<VirtualHost *:80>
    ServerName test2.co.kr
    DocumentRoot /app/service/test/web/apache2.2.17/htdocs/test2

    CustomLog                "/app/service/test/web/apache2.2.17/bin/rotatelogs
/app/service/test/web/apache2.2.17/logs/accesslogs/test/test_access_log.%Y%m%d 86400" combined
env=!image-request

    JkMount /*.do testajp13
    JkMount /*.jsp testajp13
    JkMount /*.html testajp13
</VirtualHost>

주석 해제
Include conf/extra/httpd-default.conf
Include conf/extra/httpd-mpm.conf

httpd-default.conf 설정

Timeout 300
클라이언트의 요청에 의해 서버와 연결된 후 클라이언트와 서버 간에 아무런 메시지가 발생하
지 않았을 때 오류로 처리될 시간을 초 단위로 설정하는 항목

KeepAlive On
접속한 채로 특별한 요청이 없이 지속적인 연결을 허용할 것인지 설정
3way handshaking 에 의한 TCP 연결 과정 성립 후, 서버에서 text/html 문서를 response 받았을
때, 해당 html 문서에서  라는 태그를 만나면 서버에 재 요청을 보내게 된
다.
만약, KeepAlive 상태가 On인 경우에는 3way handshaking 과정을 재시도하지 않고 기존의
session을 이용하여 서버의 test.jpg 자원을 가져오게 된다.
```



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 18/34 |

MaxKeepAliveRequests 500

클라이언트가 접속된 시간 동안 아파치 서버에 요청할 수 있는 최대 개수를 지정. 0으로 지정하면 제한이 없고, 서버의 성능향상을 위해 가능한 높은 값이 좋다.

즉, 하나의 클라이언트가 html 을 response 받고 해당 문서에 존재하는 또 다른 자원의 서버 요청 최대 개수 이다.

KeepAliveTimeout 5

아파치 서버는 접속상태의 클라이언트에서 지정한 초만큼의 요청이 없었을 때 접속을 끊게 된다. (KeepAlive가 On일 때만 유용함)

UseCanonicalName Off**AccessFileName .htaccess**

특정페이지 암호화 패스워드로 로그인할 수 있도록 하기 위한 설정으로 접근 제어할 정보파일명을 기입해준다.

ServerTokens Prod

아파치의 버전 정보를 나타낸다.

Prod: Product Only의 약어로서 웹 서버의 종류만을 제공 (Server: Apache)

Min: Minimal의 약어로서 웹 서버의 종류와 버전을 제공 (Server: Apache/2.2.15)

OS: 웹 서버의 종류와 버전, 그리고 운영체제의 종류를 제공 (Server: Apache/2.2.15 (Unix)

Full: 웹 서버의 종류와 버전, 그리고 운영체제의 종류, 설치된 모듈정보 등을 제공 (Server: Apache/2.2.15 (Unix) DAV/2 mod_jk/1.2.28 PHP/5.33

Major: 웹 서버 종류, 메이저버전 제공 (Server: Apache/2)

ServerSignature On

On 상태인 경우 브라우저의 404 페이지에 아파치 버전 및 서버 IP, port가 표시된다.

HostnameLookups On

apache accesslog 파일에 방문자에 대한 주소를 ip주소로 기록 할 것인가(off) 또는 도메인명(on)으로 기록할 것인가를 결정하는 지시자

httpd-mpm.conf 설정

worker 방식으로 설치하였다면 다음과 같이 적절하게 변경해 준다.

```
<IfModule mpm_worker_module>
  StartServers      10
  MaxClients        300
  MinSpareThreads   75
  MaxSpareThreads   150
  ThreadsPerChild   25
  MaxRequestsPerChild 1000
</IfModule>
```

표 9 httpd.conf

위의 virtual host 분류 방법은 이름기반으로 분류한 것이며, ServerName에 명시되어 있는 도메인에 따라서 분류된다.

3.9 SSL (Secure Socket Layer) 설정

다음에서 생성하는 인증서는 공인인증이 아니라 자체 서명이므로 보안 경고가 뜨거나 접속되지



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 19/34 |

않을 수 있다.

정상적으로 SSL 서비스를 운영하려면 국내 공인인증서 판매업체 <http://www.anycert.co.kr>에서 공인 인증서를 구입해야 한다.

생성한 CSR 정보를 공인인증 업체에 제출하고 계약하면 CRT 공인인증서 파일을 받을 수 있으며 업체로부터 받은 CRT 파일을 httpd-ssl.conf의 SSLCertificateFile 설정에 연결해주고, ServerName 설정에는 'Common Name'에 입력했던 도메인을 넣어주면 된다.

| No. | Install flow |
|-----|---|
| 1 | 인증서를 보관할 디렉토리에서 명령어를 입력한다. 개인키 생성 [root@localhost ~] openssl genrsa -des3 -out server.key 1024 암호 입력 |
| 2 | 개인키로 CSR(Certificate Signing Request) 생성 [root@localhost ~] openssl req -new -key server.key -out server.csr 개인키 생성 시 입력했던 암호를 입력 Country Name (2 letter code) [GB]:KR State or Province Name (full name) [Berkshire]:Seoul Locality Name (eg, city) [Newbury]:songpa Organization Name (eg, company) [My Company Ltd]:incross Organizational Unit Name (eg, section) []:developer Common Name (eg, your name or your server's hostname) []:localhost Email Address []:lng1982@in-cross.co.kr A challenge password : enter An optional company name : enter Key 파일로 csr 파일 추출 |
| 3 | 인증서 생성 [root@localhost ~] openssl x509 -in server.csr -out server.crt -req -signkey server.key -days 3650 개인키 생성 시 입력했던 암호를 입력 server.crt 인증서 생성 인증서의 유효기간을 10년으로 설정 |
| 4 | httpd-ssl.conf 파일에 생성한 key 파일 및 crt 파일 지정 SSLCertificateFile "/home/lng1982/apache2/conf/server.crt" SSLCertificateKeyFile "/home/lng1982/apache2/conf/server.key" |

3.10 virtualhost 4가지 방법

| 분류 | 설명 |
|-------------------------|---|
| Name Based Virtual host | 도메인 네임 기반 가상 호스트 하나의 IP에 여러 개의 도메인을 부여하여 각각 다른 호스트로 운용 |
| IP Based Virtual host | IP 주소 기반 가상 호스트 가상 호스트 각각에 다른 IP 주소를 부여해서 운용 |



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 20/34 |

| | |
|-------------------------|--|
| Port Based Virtual host | 포트 기반 가상 호스트 하나의 동일한 호스트에 포트만 다르게 지정해서 운용 |
| Default Virtual host | 디폴트 가상호스트 위의 세 가지에 해당사항이 없는 경우 기본적으로 응답하는 호스트 |

가상호스트 설정이 되어 있지 않은 요청을 받았을 경우 가장 최 상단에 설정된 가상호스트에 자동 연결된다.

같은 포트를 이용하는 가상호스트를 2개 이상 생성 시 NameVirtualHost를 작성해야 한다.

```
NameVirtualHost *:80

<VirtualHost *:80>
    ServerName localhost

<VirtualHost *:80>
    ServerName other.test.co.kr
```

만약 NameVirtualHost를 작성하지 않았을 경우에는 최 상단에 설정되어 있는 가상호스트로 연결된다.

가상호스트 연결이 되어 있지 않은 요청이 올 경우 에러페이지로 연결하고 싶을 때에는 `_default_` 지시자를 사용한다.

```
<VirtualHost _default_:80>
    DocumentRoot /service/test
</VirtualHost>
```

3.11 favicon.ico 설정

favicon.ico 이미지를 client web browser에게 내려주기 위해서는 httpd.conf 파일에 다음의 정보를 추가시켜줘야 한다.

Vi 에디트 모드로 httpd.conf 파일을 열어 AddType이 있는 라인 하단에 추가
`AddType image/x-icon .ico`

favicon.ico 이미지 파일은 Document Root 에 넣어주면 된다.

ex> test.com 이라는 도메인의 현재 Document Root directory가 /usr/local/test 라고 하였을 때, 해당 디렉토리에 넣어주면 된다.

3.12 MaSiteInfo.ini

Apache 로그에 /MarkAny/Websafer/MaSiteInfo.init 404 File Not found 에러가 나는 이유는 IE 브라우저에서 해당 파일을 Request 하기 때문이다.

MarkAny WebSafer DRM이 설치된 IE 브라우저가 사이트에 접속할 경우 접속하는 모든 사이트에 대해 해당 URL을 요청한다.



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 21/34 |

왜냐하면, 접속 사이트가 MarkAny사의 DRM솔루션을 이용해 보호하고 싶은 콘텐츠가 있을 수 있고, 이럴 경우 해당 콘텐츠의 복사, 프린트, 화면 캡처 등을 막아줘야 하기 때문이다.

결국 MarkAny WebSafer ActiveX가 설치되어 있는 IE 브라우저에서는 서버로의 HTTP Request 요청 시 해당 서버에 MarkAny WebSafer가 적용되어 있는지를 확인하기 위한 것을 뿐, 공격은 아니다.

MaSiteInfo.ini 에러 로그를 남기지 않으려면 다음과 같이 설정 정보를 추가해준다.

[에러 로그 남기지 않기](#)

3.13 Apache 로그 파일에 image로그 빼기

아파치 accesslog 및 errorlog 파일에 이미지 접속 및 에러 로그를 기록하지 않기 위해서는 다음과 같이 설정 정보를 추가해 준다.

| <i>Image 로그 빼기</i> | |
|--|--|
| Apache 설정 파일인 httpd.conf 파일에 추가 SetEnvIf Request_URI "\.(gif jpg png css js swf ini)\$" image-request | |
| CustomLog 지시자에는 다음과 같이 combined 속성을 추가해 준다. CustomLog "/app/service/test/web/apache2.2.17/bin/rotatelog" -1 /app/service/test/web/apache2.2.17/logs/accesslogs/man/man_access_log.%Y%m%d 86400" combined env=!image-request | |

표 10 Image 로그 빼기

3.14 mod_expires.so 모듈 설치

mod_expires 모듈을 통해 Expires HTTP header를 설정할 수 있다.

이 모듈을 사용하게 되면 웹 페이지 방문자가 response 받는 문서나 이미지들을 브라우저에 캐싱 하여 트래픽을 감소시킬 수 있다.

| No. | <i>Install flow</i> |
|-----|---|
| 1 | /usr/local/apache/bin/apxs -aic /usr/local/src/httpd-2.2.14/modules/metastata/mod_expires.c 위의 명령문을 실행하게 되면 APACHE_HOME의 modules 디렉토리에 mod_expires.so 파일이 생성된다. |
| 2 | httpd.conf 내용 추가 LoadModule expires_module libexec/mod_expires.so <IfModule mod_expires.c> ExpiresActive On ExpiresDefault "access plus 1 month" ExpiresByType application/javascript "access plus 1 month" ExpiresByType text/css "access plus 1 month" ExpiresByType image/jpeg "access plus 1 month" ExpiresByType image/gif "access plus 1 month" ExpiresByType image/png "access plus 1 month" <Directory "/usr/local/apache/htdocs"> ExpiresActive Off </Directory> </IfModule> <VirtualHost *> |



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 22/34 |

| | |
|---|--|
| | <pre> ServerName localhost DocumentRoot /home/test/www #이미지(gif/jpeg) 캐싱 한 달로 설정 <Directory "/home/test/img"> ExpiresActive On ExpiresByType image/jpeg "access plus 1 month" ExpiresByType image/gif "access plus 1 month" </Directory> </VirtualHost> </pre> |
| 3 | <p>작동 확인</p> <p>Response header의 Cache-Control 을 확인 한다. ex> Cache-Control: max-age=2592000 2592000 값은 30일을 뜻 한다.</p> |
| 4 | <p>ExpiresDefault는 모든 문서에 대한 지시자이며 ExpiresByType은 만기 시간을 자세히 설명할 수 있는 지시자이다.</p> |

표 11 mod_expires.so install

3.15 설치 후 확인 사항

| <i>progress</i> |
|---|
| CustomLog "logs/access_log" common → 주석 처리 |
| <pre> ErrorLog "logs/error_log"를 다음과 같이 변경 ErrorLog "/service/test/web/apache2/bin/rotatelogs -1 /service/test/web/apache2/logs/errorlogs/error_log.%Y%m%d 86400" </pre> |
| ServerName localhost → localhost로 설정 |
| Port Based Virtual host로 셋팅되어 있다면 httpd.conf 파일의 Listen에 open되는 포트를 등록한다. |
| <p>netstat 명령어를 이용하여 virtualhost port, AJP port, tomcat port, tomcat shutdown port 사용 유무를 확인 한다.</p> <p>만약 이미 사용중인 포트를 할당하여 apache 및 tomcat을 구동하게 된다면 연동이 안 되는 현상이 발생</p> |
| <p>VirtualHost에서 다음과 같이 NAS 디렉토리를 접근하기 위한 설정을 하였더라면 빨간색 line을 추가해 줘야 한다. (디렉토리 리스팅 방지)</p> <pre> <Directory /data> Options -Indexes -FollowSymLinks Order allow,deny Allow from all </Directory> </pre> |



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 23/34 |

4 Tomcat WAS Server

인터넷 상에서 HTTP을 통해 사용자 컴퓨터나 장치에 애플리케이션을 수행해주는 미들웨어(소프트웨어 엔진)

웹 서버와 데이터베이스 서버 사이에 위치하며 동적 트랜잭션을 처리하고 비즈니스 로직 등을 수행한다.

4.1 Tomcat Construction

Tomcat Server 는 다음과 같이 총 3개의 instance를 생성하며, 각 instance 별로 Application이 하나씩 등록되어 있다.

한 장비에 여러 개의 Tomcat instance 생성 시 shutdown port 및 AJP/1.3 port가 서로 충돌되지 않게 적절히 변경해줘야 한다. 또한, AJP/1.3 port는 Apache 연동 시 사용되는 port 이므로 workers.properties 파일에 정의되어 있는 port와 동일하게 맞춰주도록 한다.

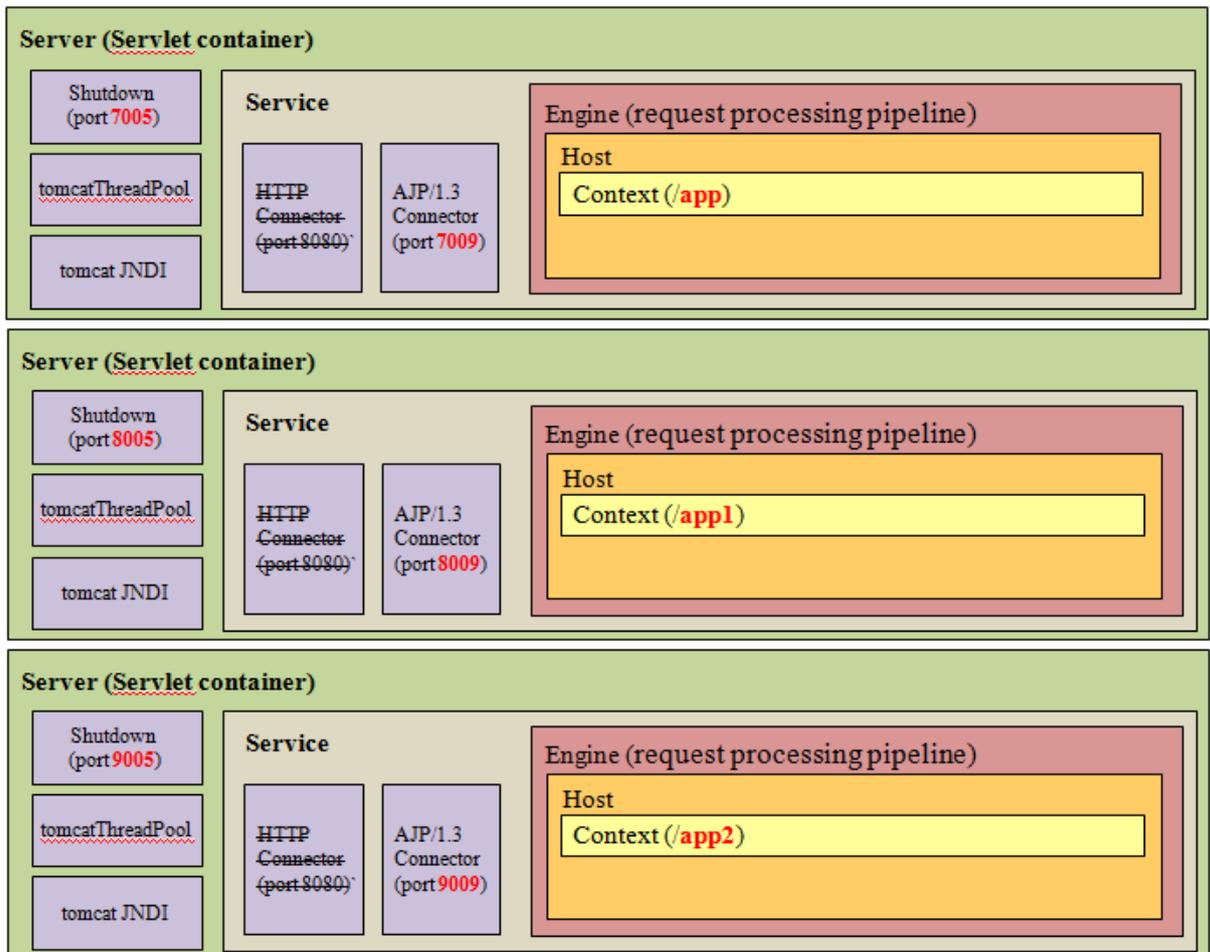


그림 2 Tomcat Construction



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 24/34 |

4.2 Tomcat Install

Tomcat 설치에 아래의 flow와 같이 진행한다.

| No. | Install flow |
|-----|--|
| 1 | http://apache.org/ 웹 페이지에 접근 |
| 2 | 이동한 페이지 하단에 보면 “Binary Distributions”와 Source Code Distributions이 있다. 리눅스 서버에 톰캣을 설치하기 위한 목적이기때 “Binary Distributions”의 Core에 있는 tar.gz (pgp, md5)를 다운로드 받는다. |
| 3 | tar.gz (pgp, md5)에 마우스 커서를 올려 놓은 후, 오른쪽 마우스를 클릭 속성창이 뜨면, 주소(URL) 부분을 복사한다. |
| 4 | 리눅스 command 창에서 wget을 이용하여 tomcat binary 파일 다운로드 wget http://apache.mirror.cdnetworks.com/tomcat/tomcat-6/v6.0.28/bin/apache-tomcat-6.0.28.tar.gz |
| 5 | 압축을 푼다. tar xvfz apache-tomcat-6.0.28.tar.gz |

표 12 Tomcat install flow

4.3 Native Library Install

Tomcat 5.5 버전 이상부터 설치한 후 실행시키면 CATALINA_HOME/logs/catalina.out에 아래와 같은 메시지를 확인할 수 있다.

정보: The ARP based Apache Tomcat Native library which allows optimal performance in production environments was not found on the java.library.path..

위 메시지는 톰캣 Native library를 사용하지 않았을 때 나오는 문구이며 아래와 같이 Tomcat Native library를 설치해 주면 된다.

Native library를 사용하게 되면 다음과 같이 성능 향상이 된다.

1. TCP/IP 커넥션 핸들링 향상
2. 파일 I/O 향상
3. SSL 처리 향상
4. Apache 나 IIS을 깔지 않아도 된다. (기능 향상)

| No. | Install flow |
|-----|---|
| 1 | http://tomcat.apache.org/download-native.cgi 웹 페이지 이동 |
| 2 | Tomcat Native Connector Caption 부분에 링크되어 있는 tar.gz 파일 URL 복사 |
| 3 | 적당한 폴더에 해당 tomcat native library를 받고 압축을 풀어 준다. |
| 4 | jni/native 디렉토리에서 아래의 명령어를 실행해 준다. apr 옵션은 아파치 HOME 디렉토리를 지정해주며 java 옵션은 자바 HOME 디렉토리를 prefix 옵션에는 TOMCAT HOME 디렉토리를 지정 한다. ./configure --with-apr=/app/service/test/web/apache2.2.17 --with-java-home=/usr/local/jdk1.6.0_21 -- prefix=/app/service/test/was/tomcat6029 \$ make \$ make install |



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 25/34 |

| | |
|---|--|
| 5 | catalina.sh 파일에 아래의 환경 변수 추가 LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/test/was/tomcat6029/lib export LD_LIBRARY_PATH |
| 6 | 설치 완료 후, Tomcat 재 기동하면 아래와 같이 native library가 정상적으로 Load되었다고 나온다. 정보: Loaded APR based Apache Tomcat Native library .1.1.16 |

표 13 Native library install flow

4.4 Tomcat Accesslog 설정

Tomcat 서버의 앞 단에 Apache가 없을 시 사용하면 유용하며 HTTP protocol 을 이용한 Tomcat 서버 접근 시 해당 자원에 대해서 기록 한다.

다음의 표에 나오는 설정은 기본적으로 server.xml 의 Context element 의 자식 요소 부분에 위치 시키면 된다.

| <i>Accesslog format</i> |
|---|
| <pre><Valve className="org.apache.catalina.valves.AccessLogValve" directory="../accesslogs" prefix="test_dev_access_log." suffix=".log" pattern="common" resolveHosts="false" /></pre> <p>User-Agent를 로그에 남기기 위해서는 다음과 같이 pattern 정의 pattern="%h %u %t %r %b % {Referer}i % {User-Agent}i [%s]"</p> |

표 14 Tomcat accesslog format

4.5 Tomcat JVM 메모리 설정

Tomcat 설치 후 process를 기동하면 기본적으로 JVM 메모리 영역을 64M 할당하여 실행 시킨다. 이로 인하여 어플리케이션에서의 과도한 메모리 사용으로 인하여 Out of memory error 가 발생하게 되며, 이에 대한 error를 방지하기 위해서는 적절한 JVM 메모리 할당이 필요하다.

기본적으로 다음과 같이 catalina.sh 파일에 설정을 추가한다.

| <i>JVM memory</i> |
|---|
| <pre>JAVA_HOME=/usr/local/java export JAVA_HOME JAVA_OPTS="-D\$SERVER_NAME -Xms1024m -Xmx1024m -XX:MaxPermSize=512m" JAVA_OPTS="\${JAVA_OPTS} -XX:+PrintGCDetails -XX:+PrintGCTimeStamps " JAVA_OPTS="\${JAVA_OPTS} -verbose:gc -Xloggc:../gclogs/"\$SERVER_NAME"_gc.log -XX:+HeapDumpOnOutOfMemoryError" JAVA_OPTS="\${JAVA_OPTS} -Dfile.encoding=EUC-KR" export JAVA_OPTS</pre> |

표 15 JVM memory

Xms : 최초 JVM이 로드 될 때 부여하는 메모리 영역

Xmx : 최대 JVM이 가질 수 있는 메모리 영역



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 26/34 |

MaxPermSize : JVM 내의 클래스 정보가 담길 최대 메모리

Xloggc : 해당 옵션을 지정해 줌으로써 -verbose:gc 의 출력 값을 파일로 출력할 수 있다.

4.6 Tomcat JNDI 설정

JNDI (Java Naming and Directory Interface)는 JDBC 드라이버를 이름으로 콜 하는 방식이며, 어플리케이션에서 DB연결 정보를 가지지 않고 이름으로만 접속을 하기 위함이다.

즉, Tomcat에서 제공하는 DBCP (Database Connection Pooling)을 사용하기 위해 JNDI을 사용하는 것이다.

다음은 설정 사항들에 대한 정보이다. (DB: oracle)

| <i>Tomcat JNDI</i> |
|---|
| Tomcat server.xml 파일안의 GlobalNamingResources element 하위에 아래의 정보 기입 <pre><Resource name="jdbc/testDS" auth="Container" type="javax.sql.DataSource" driverClassName="oracle.jdbc.driver.OracleDriver" initialSize="20" maxActive="100" maxIdle="-1" maxWait="-1" username="test" password="test" url="jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=172.18.167.231)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=172.18.167.231)(PORT=1521))(LOAD_BALANCE off)(failover=on)(CONNECT_DATA=(SERVER = DEDICATED)(SERVICE_NAME = TESTDB)))" validationQuery="SELECT 1 FROM DUAL" testOnBorrow="true" poolPreparedStatements="true" maxOpenPreparedStatements="10" removeAbandoned="true" removeAbandonedTimeout="60" logAbandoned="true" /></pre> |
| Context element 하위에 아래의 정보 기입 <pre><ResourceLink name="jdbc/testDS" global="jdbc/testDS" type="javax.sql.DataSource" /></pre> |
| 스프링에서 아래와 같이 설정하여 Tomcat 의 JNDI 사용 <pre><bean id="dataSource" class="org.springframework.jndi.JndiObjectFactoryBean"> <property name="jndiName" value="jdbc/testDS"/> <property name="resourceRef" value="true" /> </bean></pre> |
| database driver jar 파일은 TOMCA HOME의 lib 디렉토리에 넣어줘야만 driver not found Exception이 발생하지 않고 정상적으로 구동 된다. |

☞ 16 Tomcat JNDI

위의 방법 외에 다음과 같은 방법을 이용하여 JNDI 설정이 가능하다.

```
context.xml 파일에 위의 ResourceLink 정보 등록
웹 어플리케이션의 web.xml 에 아래 정보 추가
<resource-ref>
  <description>DB Connection</description>
  <res-ref-name>jdbc/xxxxDS</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 27/34 |

4.7 Tomcat thread pool 셋팅

| <i>Tomcat thread pool</i> |
|---|
| <p>server.xml 파일의 Executor element 주석 해제한 후, 다음과 같이 설정 한다.</p> <pre><!--The connectors can use a shared executor, you can define one or more named thread pools--> <Executor name="tomcatThreadPool" namePrefix="catalina-exec-" maxThreads="150" minSpareThreads="50" maxIdleTime="60000" /></pre> |
| <p>tomcat의 server.xml 설정 부분에 http protocol 및 AJP/1.3 protocol 이 두 가지 설정 부분이 있는데 stand alone으로 tomcat을 사용하면 http protocol 설정 부분에 아래와 같이 executor 추가한다.</p> <pre><Connector port="8080" executor="tomcatThreadPool" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" /></pre> |
| <p>Apache와 tomcat을 연동하여 AJP/1.3 protocol을 사용한다면 다음과 같이 추가한다.</p> <pre><Connector port="8009" executor="tomcatThreadPool" protocol="AJP/1.3" redirectPort="8443" /></pre> |

표 17 Tomcat thread pool

Executor element 요소의 속성 값에 대한 설명은 다음과 같다.

namePrefix: Thread 각각의 접두사이며, 이름이다. (namePrefix + threadNumber와 같이 조합)

maxThreads: 최대 수용할 수 있는 Thread 개수며, default는 200이다.

minSpareThreads: 언제나 살아 있는 Thread의 최소 개수며, default는 25이다.

maxIdleTime: Idle thread가 살아 있는 시간(ms)이며, default는 60000(1분) 이다.

4.8 Catalina.sh 환경 변수 추가 및 catalina.out 로그 일자 별 분리

Catalina.sh 파일에 아래의 코드를 추가해 준다.

왼쪽(수정 전) 오른쪽(수정 후): WinMerge 프로그램 사용

| | |
|---|--|
| <pre>86 # 87 # OS specific support. \$var _must_ be set to either true or false. 88 cygwin=false 89 os400=false</pre> | <pre>89 # 90 # 91 JAVA_HOME=/usr/java/jdk1.6.0_21 92 export JAVA_HOME 93 # 94 JAVA_OPTS="-D\$SERVER_NAME -Xms1024m -Xmx1024m -XX:MaxPermSize=512m" 95 JAVA_OPTS=\${JAVA_OPTS} -XX:+PrintGCDetails -XX:+PrintGCTimeStamps 96 JAVA_OPTS=\${JAVA_OPTS} -verbose:gc -Xloggc:../gclogs/"\$SERVER_NAME".gc.log 97 # -XX:+HeapDumpOnOutOfMemoryError 98 JAVA_OPTS=\${JAVA_OPTS} -Dfile.encoding=EUC-KR 99 # 100 export JAVA_OPTS 101 PATH="\$JAVA_HOME/bin:\$PATH" 102 export PATH 103 # 104 CATALINA_HOME="/app/svc/imad/was/tomcat6029" 105 export CATALINA_HOME 106 # 107 #LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/app/svc/imad/was/tomcat6029/lib 108 #export LD_LIBRARY_PATH 109 # 110 # OS specific support. \$var _must_ be set to either true or false. 111 cygwin=false 112 os400=false</pre> |
| <pre>JAVA_HOME=/usr/java/jdk1.6.0_21 export JAVA_HOME JAVA_OPTS="-D\$SERVER_NAME -Xms1024m -Xmx1024m -XX:MaxPermSize=512m" JAVA_OPTS=\${JAVA_OPTS} -XX:+PrintGCDetails -XX:+PrintGCTimeStamps " JAVA_OPTS=\${JAVA_OPTS} -verbose:gc -Xloggc:../gclogs/"\$SERVER_NAME".gc.log</pre> | |



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 28/34 |

```

XX:+HeapDumpOnOutOfMemoryError"
JAVA_OPTS="{JAVA_OPTS} -Dfile.encoding=EUC-KR"

export JAVA_OPTS

PATH="{JAVA_HOME}/bin:$PATH"
export PATH

CATALINA_HOME="/app/service/test/was/tomcat6029"
export CATALINA_HOME

#LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/app/service/test/was/tomcat6029/lib
#export LD_LIBRARY_PATH

```

표 18 catalina.sh 환경 변수 추가

Catalina.log을 일자 별로 분리하기 위해서는 다음과 같이 변경 하다.

| | |
|---|---|
| <pre> 110 # Set standard environment variables 111 PRGDIR= dirname "\$PRG" 112 113 # Only set CATALINA_HOME if not already set 114 [-z "\$CATALINA_HOME"] && CATALINA_HOME=cd "\$PRGDIR/.." >/dev/null; pwd 115 116 # Ensure that any user defined CLASSPATH variables are not used on startup, 117 # but allow them to be specified in setenv.sh, in rare case when it is needed. </pre> | <pre> 124 PRGDIR= dirname "\$PRG" 124 PRGDIR= dirname "\$PRG" 124 PRGDIR= dirname "\$PRG" 127 [-z "\$CATALINA_HOME"] && CATALINA_HOME=cd "\$PRGDIR/.." ; pwd 128 129 # Ensure that any user defined CLASSPATH variables are not used on startup, 130 # but allow them to be specified in setenv.sh, in rare case when it is needed. </pre> |
| <pre> 320 shift 321 touch "\$CATALINA_OUT" </pre> | <pre> 330 shift 334 prefix=`date +%Y%m%d%H%M` 335 CATALINA_OUT=\$CATALINA_HOME/logs/\$SERVER_NAME.out 336 CATALINA_OUT_BAK=\$CATALINA_HOME/logs/backup/\$SERVER_NAME.out.\$prefix 337 mv \$CATALINA_OUT \$CATALINA_OUT_BAK </pre> |
| <pre> 334 org.apache.catalina.startup.Bootstrap "\$@" start ? 335 >> "\$CATALINA_OUT" 2>&1 & 336 337 else 338 "\$_RUNJAVA" "\$LOGGING_CONFIG" \$JAVA_OPTS \$CATALINA_OPTS # 339 -Djava.endorsed.dirs="\$JAVA_ENDORSED_DIRS" -classpath "\$CLASSPATH" # 340 -Dcatalina.base="\$CATALINA_BASE" # 341 -Dcatalina.home="\$CATALINA_HOME" # </pre> | <pre> 350 org.apache.catalina.startup.Bootstrap "\$@" start ? 351 >> "\$CATALINA_OUT" 2>&1 & 352 353 else 354 "\$_RUNJAVA" \$JAVA_OPTS "\$LOGGING_CONFIG" \$CATALINA_OPTS # 355 -Djava.endorsed.dirs="\$JAVA_ENDORSED_DIRS" -classpath "\$CLASSPATH" # 356 -Dcatalina.base="\$CATALINA_BASE" # 357 -Dcatalina.home="\$CATALINA_HOME" # </pre> |
| <pre> 354 shift 355 SLEEP=5 356 357 if [! -z "\$@"]; then 358 echo \$! grep "[0-9]" > /dev/null 2>&1 359 if [\$? -gt 0]; then 360 SLEEP=\$! 361 shift </pre> | <pre> 370 shift 371 SLEEP=5 372 373 prefix=`date +%Y%m%d%H%M` 374 GC_LOG=\$CATALINA_HOME/gclogs/\${SERVER_NAME}_gc.log 375 GC_LOG_BAK=\$CATALINA_HOME/gclogs/backup/\${SERVER_NAME}_gc.log.\$prefix 376 mv \$GC_LOG \$GC_LOG_BAK 377 378 if [! -z "\$@"]; then 379 echo \$! grep "[0-9]" > /dev/null 2>&1 380 if [\$? -eq 1]; then 381 SLEEP=\$! 382 shift </pre> |
| <pre> prefix=`date +%Y%m%d%H%M` GC_LOG=\$CATALINA_HOME/gclogs/\${SERVER_NAME}_gc.log GC_LOG_BAK=\$CATALINA_HOME/gclogs/backup/\${SERVER_NAME}_gc.log.\$prefix mv \$GC_LOG \$GC_LOG_BAK if [\$? -eq 1]; then </pre> | |



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 29/34 |

```
373 kill -0 "cat $CATALINA_PID" >/dev/null 2>&1
374 if [ $? -gt 0 ]; then
375     echo "PID file ($CATALINA_PID) found but no matching process was found."
376     Stop aborted.
377     exit 1
378 fi
379 else
380     echo "#$CATALINA_PID was set ($CATALINA_PID) but the specified file does
381     not exist. Is Tomcat running? Stop aborted."
382     exit 1
383 fi
384 fi
385 "$_RUNJAVA" $JAVA_OPTS #
386 -Djava.endorsed.dirs="$JAVA_ENDORSED_DIRS" -classpath "$CLASSPATH" #
387 -Dcatalina.base="$CATALINA_BASE" #
388 -Dcatalina.home="$CATALINA_HOME" #
389 -Djava.io.tmpdir="$CATALINA_TMPDIR" #
390 org.apache.catalina.startup.Bootstrap "$@" stop
391
392 if [ ! -z "$CATALINA_PID" ]; then
393     if [ ! -f "$CATALINA_PID" ]; then
394         while [ $SLEEP -ge 0 ]; do
395             kill -0 "cat $CATALINA_PID" >/dev/null 2>&1
396             if [ $? -eq 1 ]; then
397                 rm $CATALINA_PID
398                 break
399             fi
400         done
401     fi
402 fi
403 fi
404 "$_RUNJAVA" $JAVA_OPTS #
405 -Djava.endorsed.dirs="$JAVA_ENDORSED_DIRS" -classpath "$
406 -Dcatalina.base="$CATALINA_BASE" #
407 -Dcatalina.home="$CATALINA_HOME" #
408 -Djava.io.tmpdir="$CATALINA_TMPDIR" #
409 org.apache.catalina.startup.Bootstrap "$@" stop
410
411 if [ ! -z "$CATALINA_PID" ]; then
412     if [ ! -f "$CATALINA_PID" ]; then
413         while [ $SLEEP -ge 0 ]; do
414             kill -0 "cat $CATALINA_PID" >/dev/null 2>&1
415             if [ $? -eq 1 ]; then
416                 rm $CATALINA_PID
417                 break
418             fi
419         done
420     fi
421 fi
422 fi
```

표 19 catalina.log 일자 별 분리

4.9 Tomcat Instance 생성

TOMCAT_HOME/bin 디렉토리에 다음과 같이 star, stop shell 스크립트를 생성한다.

```
StartTestDevSvr.sh

#!/bin/sh

# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# -----
# Start Script for the CATALINA Server
#
# $Id: startup.sh 562770 2007-08-04 22:13:58Z markt $
# -----

SERVER_NAME=testDevSvr
CATALINA_BASE=/app/service/test/was/tomcat6029/testDevSvr
export SERVER_NAME CATALINA_BASE

# Better OS/400 detection: see Bugzilla 31132
os400=false
darwin=false
case "`uname`" in
CYGWIN*) cygwin=true;;
OS400*) os400=true;;
Darwin*) darwin=true;;
```



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 30/34 |

```
esac

# resolve links - $0 may be a softlink
PRG="$0"

while [ -h "$PRG" ] ; do
  ls=`ls -ld "$PRG"`
  link=`expr "$ls" : '.*-> \(.*\)${`}'`
  if expr "$link" : '/.*' > /dev/null; then
    PRG="$link"
  else
    PRG=`dirname "$PRG"/"$link"`
  fi
done

PRGDIR=`dirname "$PRG"`
EXECUTABLE=catalina.sh

# Check that target executable exists
if $os400; then
  # -x will Only work on the os400 if the files are:
  # 1. owned by the user
  # 2. owned by the PRIMARY group of the user
  # this will not work if the user belongs in secondary groups
  eval
else
  if [ ! -x "$PRGDIR"/"$EXECUTABLE" ]; then
    echo "Cannot find $PRGDIR/$EXECUTABLE"
    echo "This file is needed to run this program"
    exit 1
  fi
fi

exec "$PRGDIR"/"$EXECUTABLE" start "$@"
```

stopTestDevSvr.sh

```
#!/bin/sh

# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements. See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License. You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# -----
# Stop script for the CATALINA Server
#
# $Id: shutdown.sh 562770 2007-08-04 22:13:58Z markt $
# -----

SERVER_NAME=testDevSvr
CATALINA_BASE=/app/service/test/was/tomcat6029/testDevSvr
export SERVER_NAME CATALINA_BASE
# resolve links - $0 may be a softlink
PRG="$0"
```



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 31/34 |

```

while [ -h "$PRG" ]; do
  ls=`ls -ld "$PRG"`
  link=`expr "$ls" : '.*-> (.*)$'`
  if expr "$link" : '/.*' > /dev/null; then
    PRG="$link"
  else
    PRG=`dirname "$PRG"/"$link"`
  fi
done

PRGDIR=`dirname "$PRG"`
EXECUTABLE=catalina.sh

# Check that target executable exists
if [ ! -x "$PRGDIR"/"$EXECUTABLE" ]; then
  echo "Cannot find $PRGDIR/$EXECUTABLE"
  echo "This file is needed to run this program"
  exit 1
fi

exec "$PRGDIR"/"$EXECUTABLE" stop "$@"

```

표 20 Tomcat instance 생성 스크립트

빨간색으로 강조된 CATALINA_BASE의 지정 경로에 다음과 같이 디렉토리를 생성한다.

conf, logs, work, temp 디렉토리를 복사하여 testDevSvr 디렉토리에 붙여 넣기 한다.

표 21 Tomcat instance 생성 디렉토리 구조

4.10 지시자

| 지시자 | 설명 |
|----------|---|
| jvmRoute | JSESSIONID 쿠키 값의 맨 끝부분에 jvmRoute 이름이 붙는다. (단순 구분자 역할) ex> 6E343D454D2332FD.v1_devSvr |



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 32/34 |

| | |
|--------------|--|
| redirectPort | http를 8080포트를 이용하여 서비스 하고 있는 상황에서 https로 들어왔을 때 8443으로 redirect하는 기능 https://172.18.16.44:8080로 접속하였을 경우 8443 포트로 redirect |
| allowLinking | “true”, “false” true로 지정하면 symbolic link로 되어 있는 디렉토리 접근이 가능하다. <Context path="/mgr" docBase="/app/test/service/test/ADNP_MAN" antiResourceLocking="false" antiJARLocking="false" privileged="true" allowLinking="true" /> |
| | |

4.11 Hash DOS 공격 대비

Hash Dos 공격은 많은 HTTP Request Parameter를 만들어 서버 CPU에 부하를 준다.
약 2MB 정도의 POST HTTP Request 한 번 이면 i7 core 서버를 약 2분간 CPU 100%로 만든다.

[방안]

tomcat 파라미터 개수 제한

tomcat 5.5.35, 6.0.35, 7.0.23에서만 지원 (옵션을 사용하지 않았을 때 default 값은 10,000)

```
<Connector port="8009" protocol="AJP/1.3"
  maxParameterCount="xxx"
  .../>
```

5 Windows Apache & Tomcat 연동

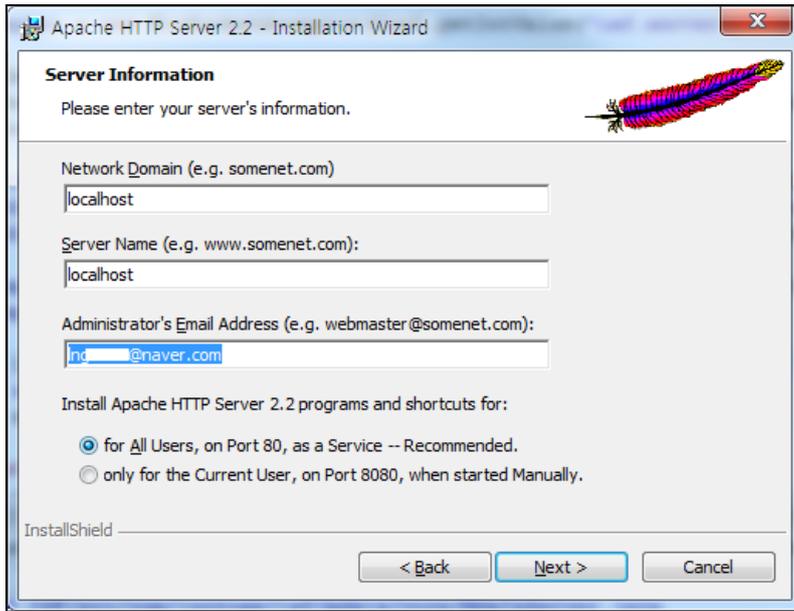
Windows 에서의 Apache + Tomcat 연동에 대해서 기술한다.

5.1 Apache 설치

- ① 윈도우용 아파치를 다운로드 받는다.
<http://httpd.apache.org/download.cgi>
- ② 다운로드 받는 installer 파일을 실행하여 설치 진행
- ③ Server information 창에서는 다음과 같이 서버 정보를 입력해야 다음 페이지로 이동할 수 있다.



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 33/34 |



④ 설치 완료

5.2 Tomcat 설치

Tomcat 설치는 <http://tomcat.apache.org/download-60.cgi> 에서 다운로드 받아 진행 한다.

5.3 Tomcat Connector 다운로드

<http://archive.apache.org/dist/tomcat/tomcat-connectors/jk/binaries/win32/jk-1.2.30/> 페이지에서 apache 버

전과 호환되는 mod_jk.so 파일 다운로드

apache 2.2 설치를 했다면 mod_jk-1.2.30-httpd-2.2.3.so 를 다운로드 하면 된다.

다운로드 완료 후 APACHE_HOME/modules 디렉토리 경로에 파일을 옮긴다.

5.4 Apache & Tomcat 연동

APACHE_HOME/conf 디렉토리에 workers.properties 파일 생성

worker.worker1.port는 tomcat server.xml의 AJP/1.3 protocol port번호와 일치시킨다.

```
worker.list=worker1
worker.worker1.port=8009
worker.worker1.host=localhost
worker.worker1.type=ajp13
```

APACHE_HOME/conf 디렉토리에 mod_jk.conf 파일 생성

```
LoadModule jk_module      modules/mod_jk.so
<IfModule mod_jk.c>
    JkWorkersFile          conf/workers.properties
    JkLogFile              logs/mod_jk.log
    JkLogLevel             info
    JkLogStampFormat       "[%a %b %d %H:%M:%S %Y]"
    JkRequestLogFormat     "%w %V %T"
</IfModule>
```



| Date | Document | Author | Page |
|------------|-------------------------|--------|-------|
| 2012-02-24 | Apache & Tomcat 설치 및 연동 | 이남규 | 34/34 |

httpd.conf 파일에 다음의 코드 삽입

```
Include conf/mod_jk.conf

NameVirtualHost *:80

<VirtualHost *:80>
    ServerName localhost
    DocumentRoot C:\Apache2.2\htdocs\test

    JkMount /*.do worker1
    JkMount /*.jsp worker1
</VirtualHost>
```

* tomcat server에 적재되어 있는 application이 context-path별로 분리가 되어 있다면 DocumentRoot 디렉토리 하위에 context-path 이름 별로 디렉토리를 생성해 줘야 한다.

Context path="/study1" → study1 디렉토리를 DocumentRoot하위에 생성

Context path="/study2" → study2 디렉토리를 DocumentRoot하위에 생성

생성하지 않으면 web.xml 파일에 정의되어 있는 welcome-file을 호출하지 못한다.