**Apache Guide**



**■ Table of Contents**

**1. Apache configuration**

1.1 설치 디렉토리 구성

**2. Apache 서버 기동,정지**

2.1 Apache 기동, 정지

2.2 기동시 에러로그 확인

**3. WEB/WAS 연동**

* 1. Tomcat connector install (modjk)
  2. mod-jk.conf 작성

3.3 workers.properties 작성

**4. 자주 사용하는 Apache 설정**

4.1 가상호스트 설정

4.2 일반 계정으로 80 서비스 포트 사용하기

4.3 Apache X-forward 설정 적용

4.4 log 경로 및 로그레벨 변경하기

4.5 modsecurity 설정 적용

4.6 서비스 점검을 위한 프로세스 및 포트 체크

4.7 에러페이지 등록

**5. 운영체제 환경설정**

5.1 커널파라미터

5.2 적용한 커널파라미터 값

5.3 사용자 limit 값

**1. Apache configuration**

1.1 설치 디렉토리 구성

|  |  |
| --- | --- |
| 디렉토리 구성 | |
| 설치 디렉토리 | /usr/local/apache2\_hanacap |
| bin 디렉토리 | /usr/local/apache2\_hanacap/bin : Apache 실행 파일들이 위치한 디렉토리 |
| conf 디렉토리 | /usr/local/apache2\_hanacap/conf : Apache 설정 파일들이 위치한 디렉토리 |
| htdocs 디렉토리 | /usr/local/apache2\_hanacap/htdocs : Apache 설치 후 기본으로 설정되는 웹 파일 디렉토리 |
| logs 디렉토리 | /usr/local/apache2\_hanacap/logs : Apache 로그 파일 기본 디렉토리 |
| Modules 디렉토리 | /usr/local/apache2\_hanacap/modules : Apache에서 기본적으로 사용하는 모듈들이 위치한 디렉토리 |

**2. Apache 서버 기동,정지**

2.1 Apache 기동, 정지

|  |  |
| --- | --- |
| Apache 웹서버 구성 | |
| 핵심 설정 파일 | /usr/local/apache2\_hanacap/conf/httpd.conf |
| 기동스크립트 위치 | /usr/local/apache2\_hanacap/bin |

/usr/local/apache2\_hanacap/bin 해당 경로로 이동하여 아래의 스크립트를 실행한다.

아파치 syntax 체크 : # ./apachectl –t

아파치 기동 : # ./apachectl start

아파치 정지 : # ./apachectl stop

아파치 기동 확인 : # ps –ef | grep httpd 또는 netstat –na | grep 80

# pwd

/usr/local/apache2\_hanacap/bin

# ./apachectl -t

Syntax OK

# ps -ef | grep httpd

root 9426 1 0 16:47 ? 00:00:00 /usr/local/apache2\_hanacap/bin/httpd -k start

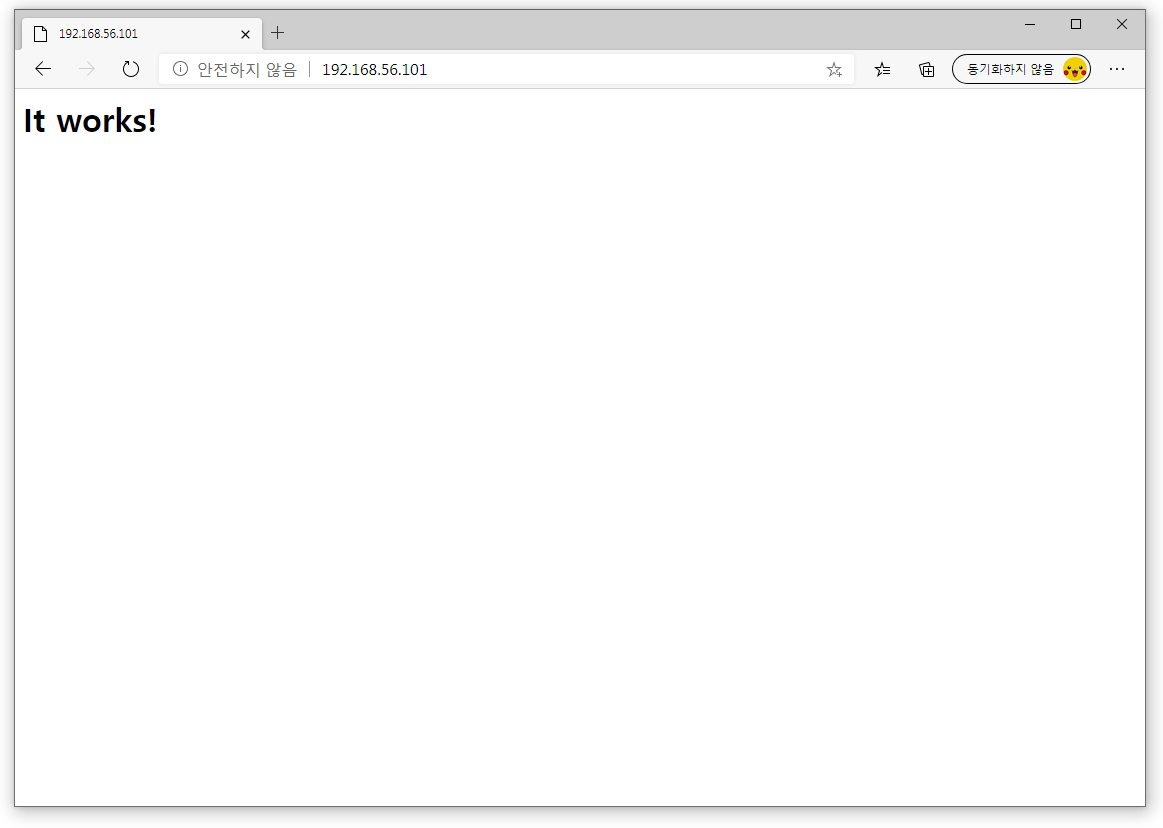
daemon 9430 9426 0 16:47 ? 00:00:00 /usr/local/apache2\_hanacap/bin/httpd -k start

daemon 9431 9426 0 16:47 ? 00:00:00 /usr/local/apache2\_hanacap/bin/httpd -k start

daemon 9432 9426 0 16:47 ? 00:00:00 /usr/local/apache2\_hanacap/bin/httpd -k start

root 9524 2923 0 16:47 pts/0 00:00:00 grep --color=auto httpd

▩ 기동 후 웹 호출 서비스 페이지 확인 (default http port : 80 )



2.2 기동 시 에러로그 확인

로그 내용 :

AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using localhost.localdomain. Set the 'ServerName' directive globally to suppress this message

서버네임이 지정되지 않아 나오는 로그이다. 서비스에는 지장이 없으나, 해당 로그가 보기 싫다면 httpd.conf 파일에서 ServerName을 지정해준다.

--------------------------------------------------------------------------------------------------------------------------------------

**3. WEB/WAS 연동**

3.1 Tomcat connector install (modjk.so)

WebServer와 WAS Server와 연동을 하기 위해서 추가 모듈 설치 및 적용을 한다.

Download URL : <https://tomcat.apache.org/download-connectors.cgi>

# tar -zxvf tomcat-connectors-1.2.46-src.tar.gz

# cd tomcat-connectors-1.2.46-src/native

# ./configure --with-apxs=/usr/local/apache2\_hanacap/bin/apxs

# make && make install

해당 경로의 파일이 존재하는지 확인한다.

/usr/local/apache2\_hanacap/modules/mod\_jk.so

다음과 같이 해당 모듈을 사용할 수 있게 설정을 해준다.

# vi /usr/local/apache2\_hanacap/conf/httpd.conf

(아래 내용 추가)

Include conf/mod-jk.conf

3.2 mod-jk.conf 생성

새로 컴파일한 mod\_jk.so 파일을 로드하면서 웹/와스 연동관련 로그 및 JKMount을 설정할 수 있다. 이 설정을 토대로 웹서버와 와스 서버의 연동이 가능해진다.

# vi /usr/local/apache2\_hanacap/conf/mod-jk.conf

# Load mod\_jk module

LoadModule jk\_module modules/mod\_jk.so

# Where to find workers.properties

JkWorkersFile conf/workers.properties

# Where to put jk shared memory

JkShmFile logs/jk.shm

JkOptions +ForwardLocalAddress

# Where to put jk logs

JkLogFile "|/usr/local/apache2\_hanacap/bin/rotatelogs /usr/local/apache2\_hanacap/logs/mod\_jk\_%Y%m%d%H.log 86400"

# Set the jk log level [debug/error/info]

JkLogLevel info

# Select the timestamp log format

JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "

# Send servlet for context /examples to worker named worker

JkMount /\* router

#JkMountFile conf/uriworkermap.properties

<Location /jkstatus>

JkMount status

Order deny,allow

Allow from 192.168.56.101

</Location>

* 1. workers.properties 생성

해당 파일은 웹/와스 연동 설정안에서 해당 웹서버가 어느 IP/Port 정보를 가지고 있는 와스 서버와 연동되어야 하는지 지정해 줄 수 있다. 아래의 설정파일은 Tomcat8a, Tomcat8b 라는 두대의 WAS를 연동하는 케이스이며, AJP Port를 통해서 통신한다. (default ajp port : 8009)

# vi workers.properties

# -------------------

worker.list=router,status

# -------------------

worker.template.type=ajp13

worker.template.ping\_mode=A

worker.template.socket\_connect\_timeout=10000

worker.template.connection\_pool\_size=15

# -------------------

# tomcat8a WAS Server

# -------------------

worker.tomcat8a.host=192.168.56.101

worker.tomcat8a.port=8009

worker.tomcat8a.reference=worker.template

# -------------------

# tomcat8b WAS Server

# -------------------

worker.tomcat8b.host=192.168.56.101

worker.tomcat8b.port=8109

worker.tomcat8b.reference=worker.template

# Define the LB worker

worker.router.type=lb

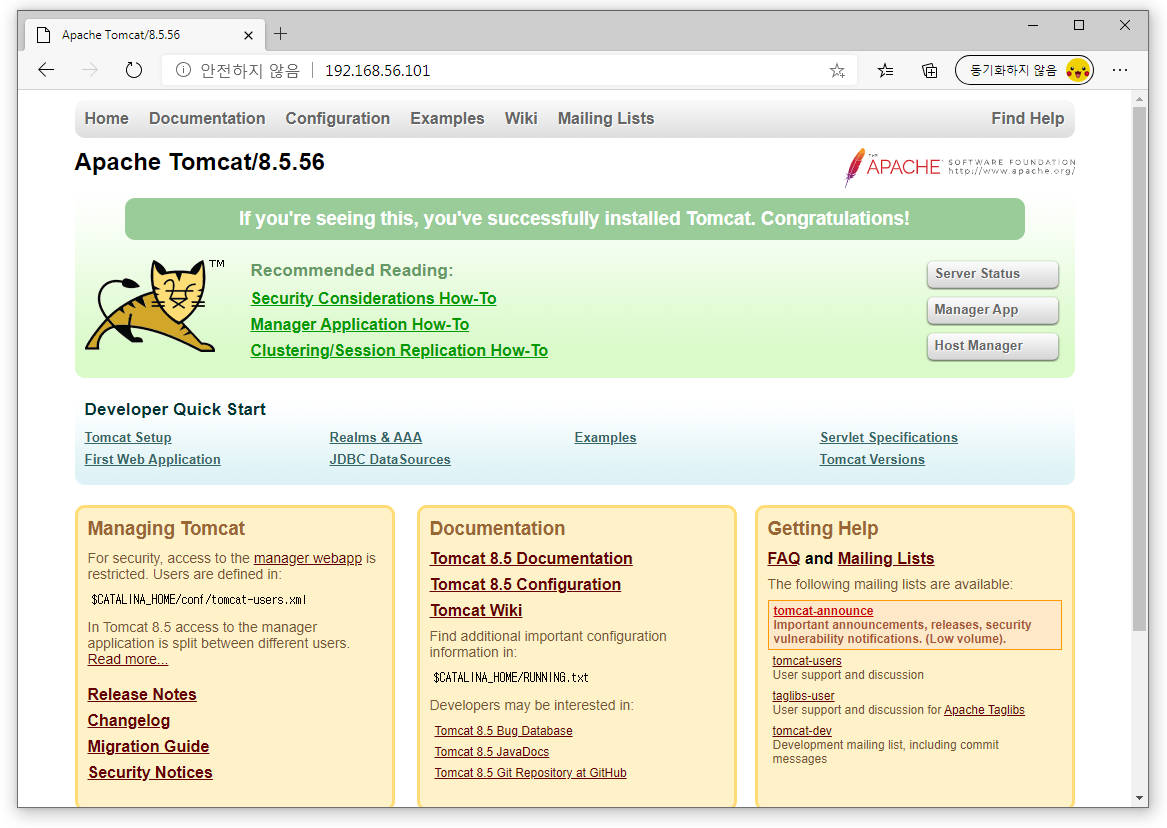
worker.router.balance\_workers=tomcat8a,tomcat8b

worker.status.type=status

▩ 기동 후 웹 호출 서비스 페이지 확인 (default http port : 80 )

웹/와스 연동이 되면 아파치 기본 페이지 호출 시 아래와 같이 톰캣 서비스 페이지가 확인된다.

이는 웹서버 호출을 통해 WAS가 가지고 있는 어플리케이션 서비스 페이지가 호출되는 것이다.



**4. 자주 사용하는Apache 설정**

4.1 가상호스트 설정

가상호스트를 사용하면 하나의 서버에서 여러 도메인의 웹사이트를 서비스가 가능하다.

아래는 간단하게 [www.test.com](http://www.test.com) 으로 가상호스트를 사용하는 예제이다.

# pwd

/usr/local/apache2\_hanacap/conf

# vi httpd.conf

Include conf/extra/httpd-vhosts.conf ← 해당 라인 주석 해제

# pwd

/usr/local/apache2\_hanacap/conf/extra

# vi httpd-vhosts.conf

<VirtualHost \*:80>

#ServerAdmin webmaster@dummy-host2.example.com

DocumentRoot "/usr/local/apache2\_hanacap/htdocs"

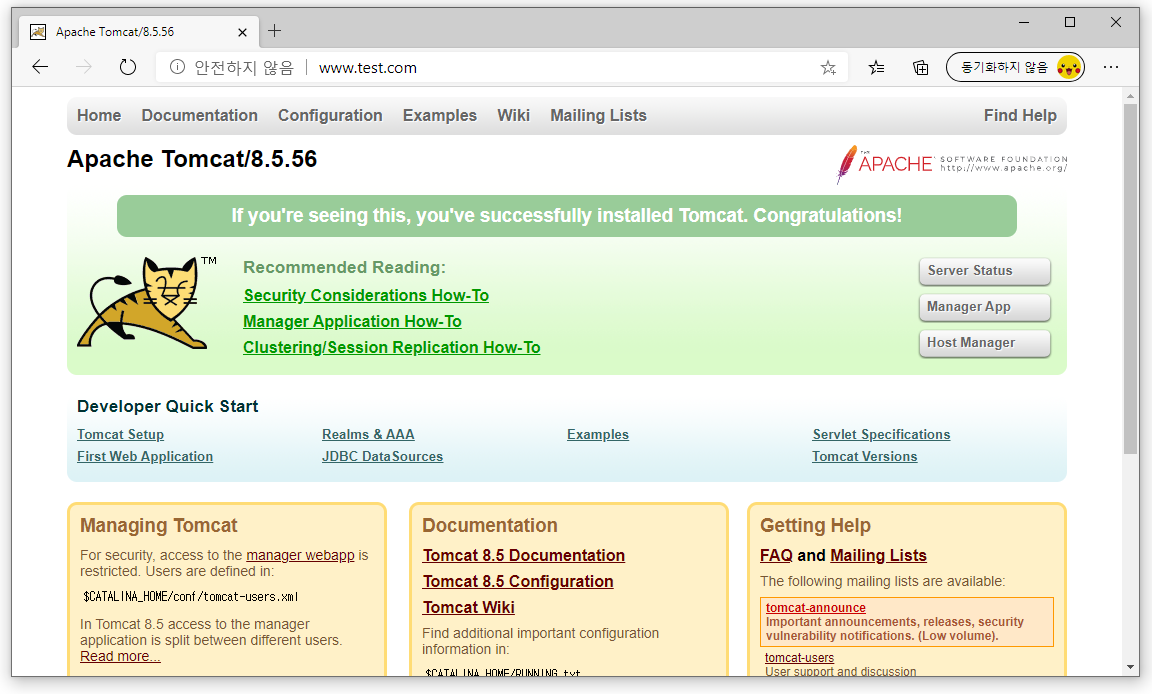
ServerName [www.test.com](http://www.test.com) ← 사용할 가상호스트 적용

JkMount /\* router ← mod-jk.conf 와 동일한 JKMount 사용

#ErrorLog "logs/dummy-host2.example.com-error\_log"

#CustomLog "logs/dummy-host2.example.com-access\_log" common

</VirtualHost>



4.2 일반 계정으로 80 서비스 포트 사용하기

▩ root 계정이 아닌 일반계정으로 Apache를 설치한 경우 well known port (0~1023) 를 LISTEN 할 수 없다. 80 포트로 서비스를 하고자 할 경우 아래와 같이 설정하면 된다.

아래는 “app” 라는 계정으로 변경하는 예제이다.

# cd /app/web

# chown -R app:app apache24

# cd /usr/local/apache2\_hanacap/conf

# vi httpd.conf

User app

Group app

# cd /usr/local/apache2\_hanacap/bin

# chmod +s httpd

# su - app

$ ./apachectl start

$ ps -ef | grep httpd

root 19978 1 0 14:09 ? 00:00:00 /usr/local/apache2\_hanacap/bin/httpd -k start

app 19982 19978 0 14:09 ? 00:00:00 /usr/local/apache2\_hanacap/bin/httpd -k start

app 19983 19978 0 14:09 ? 00:00:00 /usr/local/apache2\_hanacap/bin/httpd -k start

app 19984 19978 0 14:09 ? 00:00:00 /usr/local/apache2\_hanacap/bin/httpd -k start

app 20469 19900 0 14:16 pts/0 00:00:00 grep --color=auto httpd

4.3 Apache X-forward 설정 적용

설정 파일 : /usr/local/apache2\_hanacap/conf/httpd.conf

<IfModule log\_config\_module>

#

# The following directives define some format nicknames for use with

# a CustomLog directive (see below).

#

LogFormat " %{X-Forwarded-For}i %a %A %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined

LogFormat "%h %l %u %t \"%r\" %>s %b" common'

<IfModule logio\_module>

# You need to enable mod\_logio.c to use %I and %O

LogFormat " %{X-Forwarded-For}i %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O" combinedio

</IfModule>

적용 전

192.168.56.1 - - [21/Oct/2020:14:29:36 +0900] "GET / HTTP/1.1" 200 11394 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.80 Safari/537.36 Edg/86.0.622.43"

192.168.56.1 - - [21/Oct/2020:14:29:36 +0900] "GET /favicon.ico HTTP/1.1" 200 21630 "http://www.test.com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.80 Safari/537.36 Edg/86.0.622.43"

192.168.56.1 - - [21/Oct/2020:14:30:27 +0900] "-" 408 - "-" "-"

적용 후

- 192.168.56.1 192.168.56.101 - - [21/Oct/2020:14:48:31 +0900] "GET / HTTP/1.1" 200 11394 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.80 Safari/537.36 Edg/86.0.622.43"

- 192.168.56.1 192.168.56.101 - - [21/Oct/2020:14:48:31 +0900] "GET /favicon.ico HTTP/1.1" 200 21630 "http://www.test.com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.80 Safari/537.36 Edg/86.0.622.43"

- 192.168.56.1 192.168.56.101 - - [21/Oct/2020:14:49:12 +0900] "-" 408 - "-" "-"

4.4 log 경로 및 로그레벨 변경하기

Apache Path = /app/web/apache2/

Log Path = /app/logs/

해당 파일을 열어서

/usr/local/apache2\_hanacap/conf/httpd.conf

아래와 같이 경로를 지정한다. (재기동 이후 반영)

ErrorLog "|/usr/local/apache2\_hanacap/bin/rotatelogs /logs/error\_logs/error\_%Y%m%d%H.log 86400"

CustomLog "|/usr/local/apache2\_hanacap/bin/rotatelogs /logs/access\_logs/access\_%Y%m%d%H.log 86400" combined

- 로그레벨 설정

 # vi httpd.conf

..(중략)…

LogLevel Warn

..(중략)…

Apache 로그레벨의 종류 (아래로 갈수록 세부적으로 출력)

none

emerg

alert

crit

error

warn

notice

info

debug

\*

4.5 modsecurity 설정 적용

Modsecurity는 Apache 웹서버용 공개 웹방화벽으로 PHP Injection 공격등 Apache 웹서버의 주요 공격을 차단하는 기능을 가지고 있다.

Download URL : <https://www.modsecurity.org/download.html>

# wget <https://www.modsecurity.org/tarball/2.9.3/modsecurity-2.9.3.tar.gz>

# cd modsecurity-2.9.3/

# yum install curl\*

# yum install httpd-\*

# ./configure --with-apxs=/usr/local/apache2\_hanacap/bin/apxs

yum -y install libxml2-devel 추가

# make && make install

# vi /usr/local/apache2\_hanacap/conf/httpd.conf

LoadModule unique\_id\_module modules/mod\_unique\_id.so ← 해당 라인 주석 해제

LoadModule security2\_module modules/mod\_security2.so ← 해당 라인 추가

Include conf/extra/httpd-default.conf ← 해당 라인 주석 해제

# vi /usr/local/apache2\_hanacap/conf/extra/httpd-default.conf

ServerSignature On

SecServerSignature Hello ← 해당 라인 추가

적용 전

# curl -i http://localhost

HTTP/1.1 200 OK

Date: Thu, 22 Oct 2020 01:49:02 GMT

Server: Apache/2.4.43 (Unix)

Last-Modified: Thu, 09 Jul 2020 05:57:54 GMT

ETag: "2d-5a9fbe7c6bc80"

Accept-Ranges: bytes

Content-Length: 45

Content-Type: text/html

적용 후

# curl -i http://localhost

HTTP/1.1 200 OK

Date: Thu, 22 Oct 2020 01:49:02 GMT

Server: Hello

Last-Modified: Thu, 09 Jul 2020 05:57:54 GMT

ETag: "2d-5a9fbe7c6bc80"

Accept-Ranges: bytes

Content-Length: 45

Content-Type: text/html

4.6 서비스 점검을 위한 프로세스 및 포트 체크

Apache 설정 변경 후 재기동 또는 서비스 이상이 있는 경우 다양한 방법으로 웹서버 상태를 체크 할 수 있다. 주로 사용하는 방법은 프로세스 및 HTTP 포트 확인을 통해 서비스 기동 유무를 확인하며 에러로그 또는 modjk 로그를 통해 서비스 이상상태를 확인한다.

# ./apachectl -t

# ps -ef | grep httpd

# netstat -na | grep 80

4.7 에러페이지 등록

배포작업 및 정상적 에러페이지 호출 시 등록된 페이지 호출이 가능하다.

# vi /usr/local/apache2\_hanacap/conf/httpd.conf

(아래 추가)

ErrorDocument 404 /error.html

ex)

# Some examples:

#ErrorDocument 500 "The server made a boo boo."

#ErrorDocument 404 /missing.html

#ErrorDocument 404 "/cgi-bin/missing\_handler.pl"

#ErrorDocument 402 http://www.example.com/subscription\_info.html

2. httpd-vhosts.conf

<VirtualHost \*:80>

...

ErrorDocument 404 /error.html

ErrorDocument 401 /error.html

ErrorDocument 403 /error.html

ErrorDocument 405 /error.html

ErrorDocument 500 /error.html

ErrorDocument 501 /error.html

3. httpd-ssl.conf

<VirtualHost \*:443>

...

ErrorDocument 404 /error.html

ErrorDocument 401 /error.html

ErrorDocument 403 /error.html

ErrorDocument 405 /error.html

ErrorDocument 500 /error.html

ErrorDocument 501 /error.html

추가로 jsp, do 파일 등 동적으로 WAS 로 넘기는 파일들에 대해서는 WAS 에서도 404 error 처리를 넣어줘야 한다. Tomcat을 예를들어 설명하면 아래와 같은 설정이 추가로 필요하다.

예는 error.html이다.

error 페이지 연동법 (error 부분 추가)

파일 : web.xml

경로 : /WEB-INF

<error-page>

<error-code>404</error-code> <-- 500, 503등도 필요한 page 를 만들어서 연동하면 된다.

<location>/error.html</location> <-- 사용하고자 하는 error page 경로 및 파일명 적어준다.

</error-page> ( 디폴트 위치는 deploy 된 app 의 index 파일 위치이다. )

ex)

<filter>

<filter-name>Set Character Encoding</filter-name>

<filter-class>filters.SetCharacterEncodingFilter</filter-class>

<init-param>

<param-name>encoding</param-name>

<!--

<param-value>euc-kr</param-value>

-->

<param-value>UTF-8</param-value>

</init-param>

</filter>

<error-page>

<error-code>404</error-code>

<location>/error.html</location>

</error-page>

<filter-mapping>

<filter-name>Set Character Encoding</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

</web-app>

**5. 운영체제 환경 설정**

5.1 커널파라미터

웹 서버와 웹 기반 미들웨어 서버는 모두 네트워크를 통해 서비스를 제공하는 시스템이다. 네트워크를 통해 데이터를 전달하기 때문에, 운영체제의 TCP/IP에 대한 튜닝은 필수적이다. 아래 표에서 설명한 핵심적인 파라미터를 적용하는 것이 좋다. 특히 TCP의 수신, 송신 버퍼의 크기는 운영체제가 기본적으로 제공하는 것보다 크게 설정해야 서버의 성능을 향상할 수 있다. 다음 설정을 웹 서버와 JBoss 운영 서버에 대해 모두 적용한다.

|  |  |  |
| --- | --- | --- |
| **파라미터** | **권장값** | **설명** |
| net.ipv4.tcp\_keepalive\_time | 30 | keep-alive 시간을 줄인다. |
| net.ipv4.tcp\_fin\_timeout | 10 | FIN 타임아웃 시간을 줄여 FD를 빨리 확보할 수 있도록 한다. |
| net.core.netdev\_max\_backlog | 2500 | 백로그에 들어오는 소켓 개수를 늘린다. |
| net.ipv4.tcp\_retries1 | 3 | TCP 연결에 문제가 있을 때 연결을 재시도하는 횟수(최솟값은 3이다) |
| net.ipv4.tcp\_retries2 | 3 | TCP 연결을 끊기 전에 재시도하는 횟수를 줄인다. |
| net.ipv4.ip\_local\_port\_range | 1024 65000 | 사용할 수 있는 로컬 포트 범위를 늘린다. |
| net.core.rmem\_max | 56777216 | TCP 수신 버퍼크기 최댓값을 늘린다. |
| net.core.rmem\_default | 16777216 | TCP 수신 버퍼크기 기본값을 늘린다. |
| net.core.wmem\_max | 56777216 | TCP 전송 버퍼크기 최댓값을 늘린다. |
| net.core.wmem\_default | 16777216 | TCP 수신 버퍼크기 기본값을 늘린다. |
| net.ipv4.tcp\_window\_scaling | 1 | 65kb 이상의 큰 TCP 윈도우 스케일링을 사용한다. |
| net.ipv4.tcp\_orphan\_retries | 0 | 서버 측에서 닫은 TCP 연결을 끊기 전에 확인하는 횟수를 줄인다. 기본값은 7로 50초~16분 정도 걸린다. |
| net.ipv4.tcp\_sack | 0 | SYNC 패킷을 전송한 후 일부 ACK를 받지 못했을 경우 선택적으로 받지 못한 ACK 패킷을 받도록 설정할 수 있다. 0은 받지 않는 설정이다. 패킷 유실이 많은 네트워크에서는 1로 설정한다. |

6.2 적용한 커널파라미터 값

/etc/sysctl.conf

# Updates

net.ipv4.neigh.default.unres\_qlen=100

net.ipv4.tcp\_keepalive\_time = 30

net.ipv4.tcp\_fin\_timeout = 10

net.core.netdev\_max\_backlog = 2500

net.ipv4.tcp\_retries1 = 2

net.ipv4.tcp\_retries2 = 3

net.ipv4.ip\_local\_port\_range = 1024 65000

net.core.rmem\_max = 56777216

net.core.rmem\_default = 16777216

net.core.wmem\_max = 56777216

net.core.wmem\_default = 16777216

net.ipv4.tcp\_window\_scaling = 1

net.ipv4.tcp\_timestamps = 0

net.ipv4.tcp\_sack = 0

net.ipv4.tcp\_orphan\_retries = 0

6.3 사용자 limit 값

# /etc/security/limits.conf

#

#This file sets the resource limits for the users logged in via PAM.

#It does not affect resource limits of the system services.

#

#Each line describes a limit for a user in the form:

#

#<domain> <type> <item> <value>

#

#Where:

#<domain> can be:

# - an user name

# - a group name, with @group syntax

# - the wildcard \*, for default entry

# - the wildcard %, can be also used with %group syntax,

# for maxlogin limit

#

#<type> can have the two values:

# - "soft" for enforcing the soft limits

# - "hard" for enforcing hard limits

#

#<item> can be one of the following:

# - core - limits the core file size (KB)

# - data - max data size (KB)

# - fsize - maximum filesize (KB)

# - memlock - max locked-in-memory address space (KB)

# - nofile - max number of open files

# - rss - max resident set size (KB)

# - stack - max stack size (KB)

# - cpu - max CPU time (MIN)

# - nproc - max number of processes

# - as - address space limit (KB)

# - maxlogins - max number of logins for this user

# - maxsyslogins - max number of logins on the system

# - priority - the priority to run user process with

# - locks - max number of file locks the user can hold

# - sigpending - max number of pending signals

# - msgqueue - max memory used by POSIX message queues (bytes)

# - nice - max nice priority allowed to raise to values: [-20, 19]

# - rtprio - max realtime priority

#

#<domain> <type> <item> <value>

#

#\* soft core 0

#\* hard rss 10000

#@student hard nproc 20

#@faculty soft nproc 20

#@faculty hard nproc 50

#ftp hard nproc 0

#@student - maxlogins 4

**webAdmin hard nofile 65536**

**webAdmin soft nofile 65536**

**webAdmin soft nproc 2047**

**webAdmin hard nproc 16384**

# End of file

