

구매회사 구축 프로젝트

KHAN [a p m]

Apache HTTPD 설치 보고서



2018-05-22

오픈나루

Table of Contents

Table of Contents.....	ii
Revision History.....	iv
1. 개요.....	1
1.1 수행자 정보.....	1
1.2 고객 정보.....	1
2. 설치 서버 정보.....	2
3. 시스템 환경.....	2
3.1 운영체제 정보.....	2
3.1.1 서버 정보 요약.....	2
3.1.2 서버 정보 : TOMCAT88(192.168.23.88).....	2
4. Apache HTTPD 설정 정보.....	5
4.1 설치 디렉터리 구성.....	5
5. 운영체제 환경 설정.....	6
5.1 커널 파라미터.....	6
5.2 적용한 커널 파라미터 값.....	7
5.3 사용자 limit 값 설정.....	8
6. Apache HTTPD 환경 설정.....	10
6.1 Apache HTTPD 환경 설정.....	10
6.1.1 MPM 모듈.....	10
6.1.2 httpd.conf 설정.....	10
6.2 Connector 모듈 설정.....	11
6.2.1 mod_jk.conf.....	12
6.2.2 workers.properties(예시).....	12
7. JBoss 운영 방법.....	14
7.1 Apache 운영방법.....	14
7.1.1 Apache 확인 방법.....	14
7.1.2 Apache 운영방법.....	14
7.1.3 Apache 서비스 등록 방법.....	15
7.2 Apache HTTPD, Tomcat 테스트를 위한 웹 페이지.....	15

7.2.1 Apache HTTPD 자동 생성 웹 페이지.....	16
7.2.2 Apache HTTPD jkstatus 페이지.....	16
8. 도움이 필요하십니까?	18
9. References	19

Revision History

Name	Date	Reason For Changes	Version
오픈나루 (service@opennaru.com)	2014/1/22	Initial Version	1.0

1. 개요

1.1 수행자 정보

본 문서는 오픈나루(opennaru.com)의 자동 설치 제품인 KHAN [provisioning]을 이용하여 생성된 문서입니다. 웹 서버 / APM(Apache PHP MySQL) / WAS(Tomcat, JBoss) 미들웨어 자동 설치 제품에 대한 문의는 service@opennaru.com 으로 하시면 됩니다.

설치한 제품 및 설치 지원 회사의 정보는 다음과 같습니다.

항목	내용
설치 제품	Apache HTTPD 2.4.33
수행 일시	2018-05-22_21-34-33
설치지원 회사명	오픈나루
수행자	오픈나루㈜
이메일	support@opennaru.com
전화번호	02-469-5426

1.2 고객 정보

구분	내용
고객사	구매회사
프로젝트명	구축 프로젝트
담당자	홍길동
고객 연락처	abcde@customer.co.kr (010-1234-1234)
수행시간	2018-05-22_21-34-33

2. 설치 서버 정보

호스트 IP	서비스 URL
192.168.23.88	http://192.168.23.88/

3. 시스템 환경

설치한 시스템 기본환경에 대한 정보입니다.

3.1 운영체제 정보

3.1.1 서버 정보 요약

서버	정보	
TOMCAT88 (192.168.23.88)	운영체제	Red Hat Enterprise Linux Server release 7.3 (Maipo) x86_64
	메모리	2048988 KB
	CPU	1 개
	Core	1 개

3.1.2 서버 정보 : TOMCAT88(192.168.23.88)

구분	정보
----	----

호스트 이름	TOMCAT88
IP 주소	192.168.23.88
OS 버전	Red Hat Enterprise Linux Server release 7.3 (Maipo)
Kernel 버전	3.10.0-514.el7.x86_64
아키텍처(bit 수)	x86_64
CPU 정보	Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz
CPU 개수	1
Core 개수	1
CPU 당 Core 수	1
메모리(KB)	2048988 KB
Disk 사용량	<pre>Filesystem Size Used Avail Use% Mounted on /dev/sda3 90G 3.6G 87G 4% / devtmpfs 991M 0 991M 0% /dev tmpfs 1001M 0 1001M 0% /dev/shm tmpfs 1001M 17M 984M 2% /run tmpfs 1001M 0 1001M 0% /sys/fs/cgroup /dev/sda1 2.0G 115M 1.9G 6% /boot tmpfs 201M 0 201M 0% /run/user/1106 tmpfs 201M 0 201M 0% /run/user/0</pre>
Disk 정보	<pre>Disk /dev/sda: 107.4 GB, 107374182400 bytes, 209715200 sectors Units = sectors of 1 * 512 = 512 bytes Sector size (logical/physical): 512 bytes / 512 bytes I/O size (minimum/optimal): 512 bytes / 512 bytes Disk label type: dos Disk identifier: 0x000ac226 Device Boot Start End Blocks Id System /dev/sda1 * 2048 4196351 2097152 83 Linux /dev/sda2 4196352 20973567 8388608 82 Linux swap / Solaris /dev/sda3 20973568 209715199 94370816 83 Linux</pre>
네트워크 설정	<pre>ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500 inet 192.168.23.88 netmask 255.255.255.0 broadcast 192.168.23.255 ether 00:50:56:bc:af:e9 txqueuelen 1000 (Ethernet) RX packets 2580053 bytes 570410244 (543.9 MiB) RX errors 0 dropped 538028 overruns 0 frame 0 TX packets 1216033 bytes 323766351 (308.7 MiB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0</pre>

	<pre> lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536 inet 127.0.0.1 netmask 255.0.0.0 loop txqueuelen 1 (Local Loopback) RX packets 1470854 bytes 2096843748 (1.9 GiB) RX errors 0 dropped 0 overruns 0 frame 0 TX packets 1470854 bytes 2096843748 (1.9 GiB) TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0 </pre>
라우팅 정보	<pre> Kernel IP routing table Destination Gateway Genmask Flags MSS Window irtt Iface 0.0.0.0 192.168.23.1 0.0.0.0 UG 0 0 0 ens192 192.168.23.0 0.0.0.0 255.255.255.0 U 0 0 0 ens192 </pre>

4. Apache HTTPD 설정 정보

4.1 설치 디렉터리 구성

Apache HTTPD	
버전	2.4.33
MPM 형식	worker
APR 버전	1.6.2
APR Util 버전	1.6.0
mod_jk 버전	1.2.42
설치 디렉토리	/svc/test/web/apache2
설정파일 디렉토리	/svc/test/web/apache2/conf
Log 디렉토리	/svc/test/logs/apache2

5. 운영체제 환경 설정

5.1 커널 파라미터

웹 서버와 웹 기반 미들웨어 서버는 모두 네트워크를 통해 서비스를 제공하는 시스템이다. 네트워크를 통해 데이터를 전달하기 때문에, 운영체제의 TCP/IP 에 대한 튜닝은 필수적이다. 아래 표에서 설명한 핵심적인 파라미터를 적용하는 것이 좋다. 특히 TCP 의 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 FD 를 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 3 이다)
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 7 로 50 초~16 분 정도 걸린다.
net.ipv4.tcp_sack	0	SYNC 패킷을 전송한 후 일부 ACK 를 받지 못했을 경우 선택적으로 받지 못한 ACK 패킷을 받도록 설정할 수 있다. 0 은 받지 않는 설정이다. 패킷 유실이 많은 네트워크에서는 1 로 설정한다.

5.2 적용한 커널 파라미터 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
```

5.3 사용자 limit 값 설정

```
# /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

apache      hard   nofile      65536
apache      soft   nofile      65536

apache      soft   nproc       2047
apache      hard   nproc       16384

# End of file

```

6. Apache HTTPD 환경 설정

6.1 Apache HTTPD 환경 설정

6.1.1 MPM 모듈

Apache HTTPD 는 prefork, worker, event 방식을 모두 지원한다. KHAN [provisioning]을 이용한 설치시에는 'worker' 을 사용하도록 설치하였다.

6.1.2 httpd.conf 설정

Worker MPM 방식을 사용할 경우 아래 설정 값이 적용된다.

```
# worker MPM
# StartServers: initial number of server processes to start
# MaxClients: maximum number of simultaneous client connections
# MinSpareThreads: minimum number of worker threads which are kept spare
# MaxSpareThreads: maximum number of worker threads which are kept spare
# ThreadsPerChild: constant number of worker threads in each server process
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule mpm_worker_module>
    StartServers      10
    MaxClients        960
    MinSpareThreads   75
    MaxSpareThreads   150
    ThreadsPerChild   64
    MaxRequestsPerChild 250
</IfModule>
ListenBackLog 1000
```

Prefork 일 경우 아래 설정 값이 적용된다.

```
# worker MPM
# StartServers: initial number of server processes to start
# MaxClients: maximum number of simultaneous client connections
# MinSpareThreads: minimum number of worker threads which are kept spare
```

```
# MaxSpareThreads: maximum number of worker threads which are kept spare
# ThreadsPerChild: constant number of worker threads in each server process
# MaxRequestsPerChild: maximum number of requests a server process serves
<IfModule mpm_prefork_module>
    StartServers          5
    MinSpareServers       5
    MaxSpareServers       10
    MaxRequestWorkers     250
    MaxConnectionsPerChild 0
</IfModule>
```

그외 속도를 빠르게 하기 위해 KeepAlive 를 On 으로 설정하였다.

```
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On

#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 1000

.....

ListenBackLog 1000

.....

#LoadModule proxy_balancer_module modules/mod_proxy_balancer.so

.....
#
# ExtendedStatus controls whether Apache will generate "full" status
# information (ExtendedStatus On) or just basic information (ExtendedStatus
# Off) when the "server-status" handler is called. The default is Off.
#
ExtendedStatus On
```

6.2 Connector 모듈 설정

Apache HTTPD 는 WAS 서버와 mod_jk 방식을 사용하도록 설정하였다.

\$khanConfig.webProperty.property.SVC_WEB_HOME/httpd/conf.d 디렉토리에 mod_jk.conf, workers.properties 파일에 설정이 저장되어 있다.

6.2.1 mod_jk.conf

```
# Load mod_jk module
# Update this path to match your modules location
LoadModule jk_module          modules/mod_jk.so

# Where to find workers.properties
# Update this path to match your conf directory location (put workers.properties next to
httpd.conf)
JkWorkersFile conf.d/workers.properties

# Where to put jk logs
# Update this path to match your logs directory location (put mod_jk.log next to access_log)
JkLogFile      /svc/test/logs/web/mod_jk.log
JkShmFile      /svc/test/logs/web/mod_jk.shm

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

# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "

# JkOptions indicate to send SSL KEY SIZE,
JkOptions      +ForwardKeySize +ForwardURICompat -ForwardDirectories
+ForwardURICompatUnparsed

# JkRequestLogFormat set the request format
JkRequestLogFormat "%w %V %T"

# Send everything for context /examples to worker named worker1 (ajp13)
JkMount  /*.jsp lb
JkMount  /*.do lb
JkMount  /*.mvc lb
JkMount  /jkstatus* jkstatus
```

6.2.2 workers.properties(예시)


```
# Define load balancer worker using ajp13

worker.list=lb,jkstatus

# Templates
worker.template.type=ajp13
worker.template.maintain=60
worker.template.lbfactor=1
worker.template.ping_mode=A
worker.template.ping_timeout=2000
worker.template.prepost_timeout=2000
worker.template.socket_timeout=60
worker.template.socket_connect_timeout=2000
worker.template.socket_keepalive=true
worker.template.connection_pool_timeout=60
worker.template.connect_timeout=10000
worker.template.recovery_options=7

# Set properties for server11 (ajp13)
worker.server11.reference=worker.template
worker.server11.host=192.168.0.20
worker.server11.port=8109

# Set properties for server12 (ajp13)
worker.server12.reference=worker.template
worker.server12.host=192.168.0.20
worker.server12.port=8209

# Set properties for server13 (ajp13)
worker.server13.reference=worker.template
worker.server13.host=192.168.0.20
worker.server13.port=8309

worker.lb.type=lb
worker.lb.balance_workers=server11,server12,server13,
worker.lb.method=Session
worker.lb.sticky_session=True

worker.jkstatus.type=status
```

7. JBoss 운영 방법

7.1 Apache 운영방법

7.1.1 Apache 확인 방법

확인 항목	명령어
컴파일 옵션 확인	\$ /svc/test/web/apache2/httpd/sbin/httpd -V
HTTPD 버전 확인	\$ /svc/test/web/apache2/httpd/sbin/httpd -v
컴파일된 모듈 확인	\$ /svc/test/web/apache2/httpd/sbin/httpd -l
config 문법 체크	\$ /svc/test/web/apache2/httpd/sbin/httpd -t

7.1.2 Apache 운영방법

\$ cd /svc/test/web/apache2/httpd /sbin/

명령	명령어
시작	\$./apachectl start
종료	\$./apachectl stop
재시작	\$./apachectl restart
상태체크	\$./apachectl status
프로세스 확인	\$ ps -ef grep httpd

7.1.3 Apache 서비스 등록 방법

RHEL 시작시 Apache HTTPD 가 자동으로 시작될 수 있도록, apache-httpd 서비스 등록 스크립트가 설치되어 있다. 리눅스 시작시 자동으로 시작되도록 설정되어 있지는 않다.

다음 명령으로 Apache HTTPD 를 서비스로 등록할 수 있다.

```

$ chkconfig apache-httpd on
$ chkconfig -list apache-httpd
apache-httpd      0:off 1:off 2:on 3:on 4:on 5:on 6:off
    
```

서비스로 등록한 후에는 다음과 같이 service 명령을 사용하여 Apache HTTPD 를 관리할 수 있다.

명령	명령어
시작	\$ service apache-httpd start
종료	\$ service apache-httpd stop
재시작	\$ service apache-httpd restart
상태체크	\$ service apache-httpd status

7.2 Apache HTTPD, Tomcat 테스트를 위한 웹 페이지

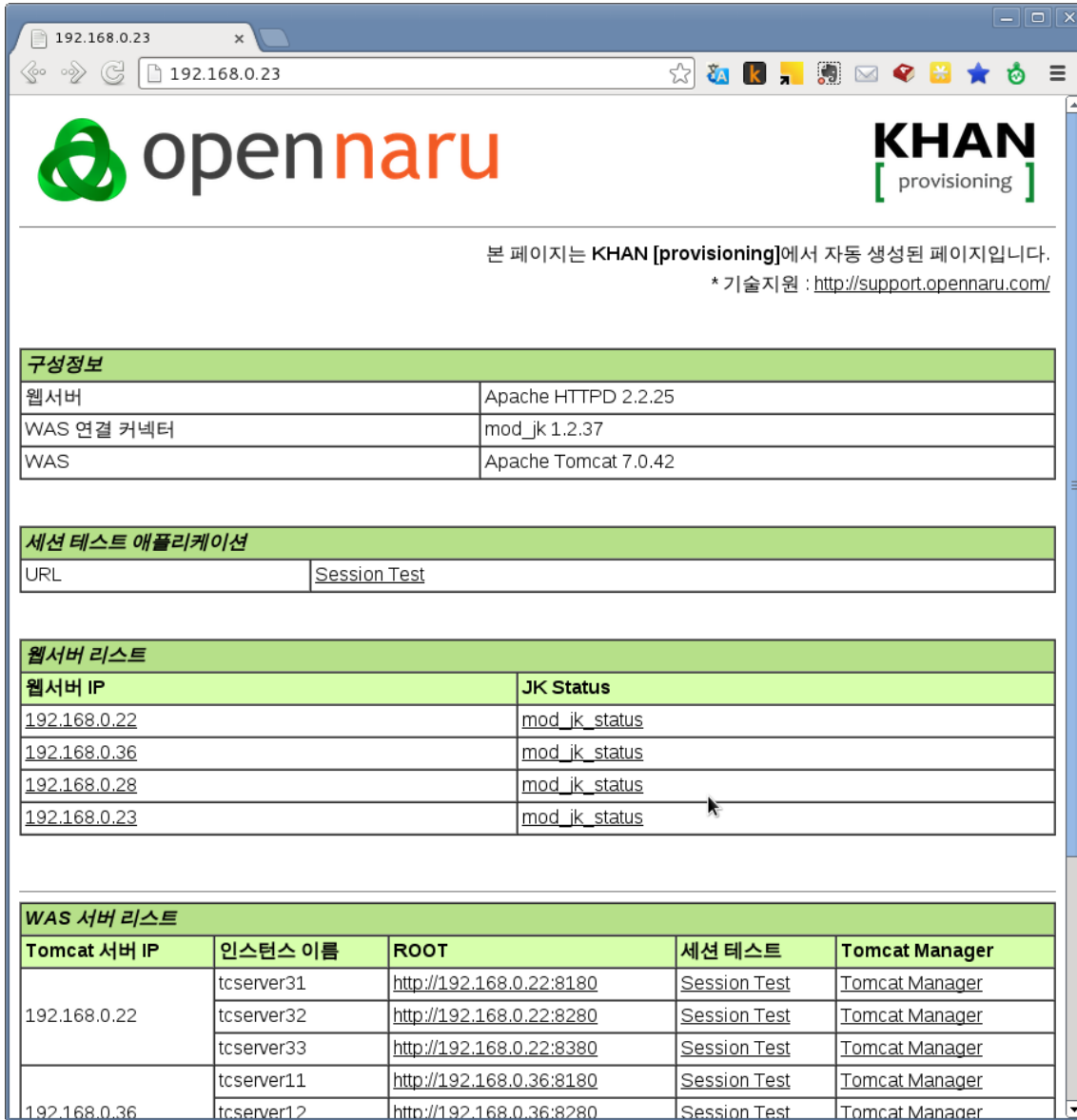
설치된 Apache HTTPD, Tomcat 서버들에 접근할 수 있는 HTML 페이지를 자동으로 생성하여 설치하였다.

Apache HTTPD 서버들의 목록과 각 서버들의 mod_jk 상태 정보를 확인해 볼 수 있는 페이지(jkstatus)들에 접근할 수 있는 링크들을 제공하고 있다.

또한, Tomcat 인스턴스들의 목록(도메인 모드와 Standalone 모드)과 Tomcat 의 웹 관리 콘솔에 접근할 수 있도록 콘솔 접근 URL 들을 출력한다.

Tomcat 인스턴스에 직접 접근할 수 있는 URL 과 Apache 를 통하여 접근하는 URL 들을 제공하고 있다.

7.2.1 Apache HTTPD 자동 생성 웹 페이지



본 페이지는 KHAN [provisioning]에서 자동 생성된 페이지입니다.
* 기술지원 : <http://support.opennaru.com/>

구성정보	
웹서버	Apache HTTPD 2.2.25
WAS 연결 커넥터	mod_jk 1.2.37
WAS	Apache Tomcat 7.0.42

세션 테스트 애플리케이션	
URL	Session Test

웹서버 리스트	
웹서버 IP	JK Status
192.168.0.22	mod_jk_status
192.168.0.36	mod_jk_status
192.168.0.28	mod_jk_status
192.168.0.23	mod_jk_status

WAS 서버 리스트				
Tomcat 서버 IP	인스턴스 이름	ROOT	세션 테스트	Tomcat Manager
192.168.0.22	tcserver31	http://192.168.0.22:8180	Session Test	Tomcat Manager
	tcserver32	http://192.168.0.22:8280	Session Test	Tomcat Manager
	tcserver33	http://192.168.0.22:8380	Session Test	Tomcat Manager
192.168.0.36	tcserver11	http://192.168.0.36:8180	Session Test	Tomcat Manager
	tcserver12	http://192.168.0.36:8280	Session Test	Tomcat Manager

7.2.2 Apache HTTPD jkstatus 페이지

JK Status Manager for 192.168.0.36:80

Server Version: Apache/2.2.25 (Unix) mod_jk/1.2.37 Server Time: Mon, 26 Aug 2013 19:56:18 KST
 JK Version: mod_jk/1.2.37 Unix Seconds: 1377514578

Start auto refresh (every seconds) | Change format

[\[Read Only\]](#) [\[Dump\]](#) [S=Show only this worker, E=Edit worker, R=Reset worker state, T=Try worker recovery]

Listing Load Balancing Worker (1 Worker) [\[Hide\]](#)

[\[S|E|R\]](#) **Worker Status for lb**

Type	Sticky Sessions	Force Sticky Sessions	Retries	LB Method	Locking	Recover Wait Time	Error Escalation Time	Max Reply Timeouts	[Hide]
lb	True	False	2	Sessions	Optimistic	60	30	0	

Good Degraded Bad/Stopped Busy Max Busy Next Maintenance Last Reset [\[Hide\]](#)

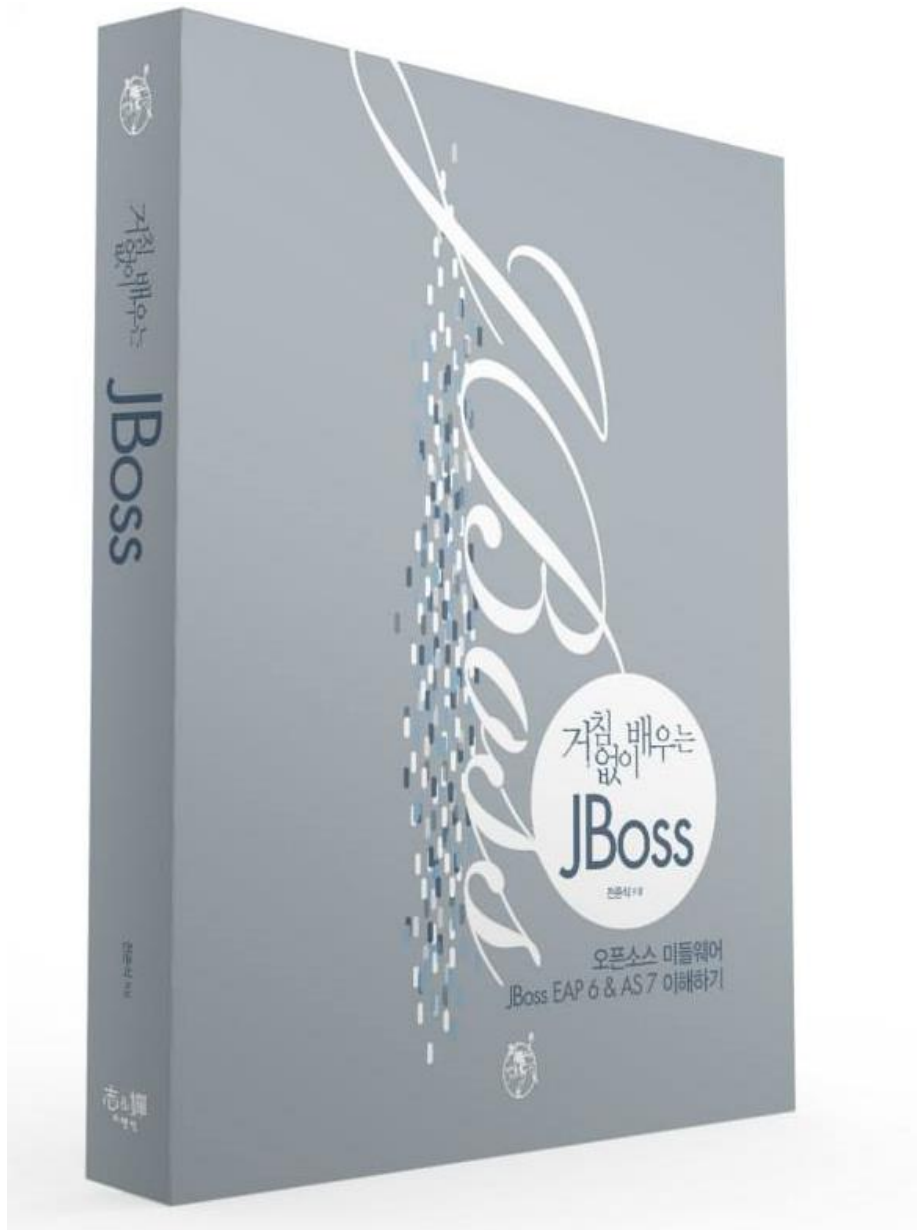
3	0	0	0	0	58/120	7006			
---	---	---	---	---	--------	------	--	--	--

Balancer Members [\[Hide\]](#)

Name	Type	Hostname	Address:Port	Connection Pool	Connect Timeout	Prepost Timeout	Reply Timeout	Retries	Recovery Options	Max Packet Size	[Hide]
tcserver11	ajp13	192.168.0.36	192.168.0.36:8109	60	10000	2000	0	2	7	8192	
tcserver12	ajp13	192.168.0.36	192.168.0.36:8209	60	10000	2000	0	2	7	8192	
tcserver13	ajp13	192.168.0.36	192.168.0.36:8309	60	10000	2000	0	2	7	8192	

8. 도움이 필요하십니까?

만약 이 문서에 설명된 절차를 수행할 때 문제를 겪는다면, 오픈나루 고객 포털(<http://support.opennaru.com>)을 방문하십시오.



9. References

- **Red Hat Documentation**
 - <http://docs.redhat.com/>

- **오픈나루 고객지원 포탈**
 - <http://support.opennaru.com>

- **오픈나루 기술 Blog**
 - <http://opennaru.tistory.com>

- **오픈나루 Facebook Page**
 - <https://www.facebook.com/opennaru>



t : +82-2-469-5426

f : +82-2-469-7247

e : service@opennaru.com, sales@opennaru.com

h : <http://www.opennaru.com>

본 문서는 오픈나루(opennaru.com)의 자동 설치 제품인 KHAN [provisioning]을 이용하여 생성된 문서입니다. 본 문서에 대한 저작권은 오픈나루 주식회사에 있습니다.