구매회사 구축 프로젝트

Apache Tomcat 설치 보고서



2019-12-27

오픈나루(주)



Table of Contents

Та	ble of	Contents	ii		
Re	Revision Historyiv				
1.	개요		1		
	1.1	수행자 정보	1		
	1.2	고객 정보	1		
2.	설치 (인스턴스 정보 요약	2		
3.	시스턷	l 환경	2		
	3.1	운영체제 정보	2		
	3.1. 1	서버 정보 요약	2		
	3.1.2	. 서버 정보 : «\$property.host.trim()»(«\$property.ip.trim()»)	5		
	3.2	인스턴스 구성정보	7		
	3.3	인스턴스 접속 정보	8		
4.	Apach	ne Tomcat 설정 정보	10		
	4.1	인스턴스 구성	10		
	4.1.1	디렉터리 구성	10		
	4.1.2	인스턴스 이름 규칙	10		
	4.2	Tomcat 운영 스크립트	12		
	4.2.1	Tomcat 스크립트	12		
	4.3	환경 설정 파일	12		
5.	운영처	제 환경 설정	16		
	5.1	커널 파라미터	16		
	5.2	적용한 커널 파라미터 값	17		
	5.3	사용자 limit 값 설정	18		
6.	Apach	ne Tomcat 주요 설정값	19		
	6.1	JDK 설치	19		
	6.2	Native 모듈 설치	20		
	6.2.1	추가 설치 패키지	20		
	6.3	Thread Pool 설정	20		
7.	Tomc	at 운영 방법	22		



8.	. MBean 그래프 모니터링		
	8.1	hawtio Chrome Extension 설치	.22
	8.2	hawtio 실행하기	.23
	8.3	MBean 모니터링	.24
9.	도움0	ㅣ필요하십니까?	.27
10	10.References		



Revision History

Name	Date	Reason For Changes	Version
오픈나루	2014/5/12	Initial Version	1.0
(service@opennaru.com)			



1. 개요

1.1 수행자 정보

본 문서는 오픈나루(opennaru.com)의 자동 설치 제품인 OPENMARU Installer 을 이용하여 생성된 문서입니다. 웹 서버 / WAS(Tomcat) 미들웨어 자동 설치 제품에 대한 문의는 <u>service@opennaru.com</u> 으로 하시면 됩니다.

설치한 제품 및 설치 지원 회사의 정보는 다음과 같습니다.

항목	내용
설치 제품	Apache Tomcat 9.0.19
수행 일시	2019-12-27_13-35-20
설치지원 회사명	오픈나루(주)
수행자	한상진
이메일	hansj@opennaru.com
전화번호	010-4507-2165

1.2 고객 정보

구분	내용
고객사	구매회사
프로젝트명	구축 프로젝트
담당자	홍길동
고객 연락처	abcde@customer.co.kr (010-1234-1234)
수행시간	2019-12-27_13-35-20



2. 설치 인스턴스 정보 요약

호스트 IP	인스턴스명	포트 오프셋	프로파일
192.168.182.138	admin11	100	
192.168.182.138	front11	200	
192.168.182.139	admin21	100	
192.168.182.139	front21	200	

이후 시스템의 운영 중 발생하는 문제에 대해서는 "한국 레드햇 고객지원 서비스"의 전화나 고객지원 포탈을 통해서 기술지원을 받으실 수 있습니다.

- 고객지원 포탈 : http://support.opennaru.com
- 기술지원 이메일 : service@opennaru.com

3. 시스템 환경

설치한 시스템 기본환경에 대한 정보입니다.

3.1 운영체제 정보

3.1.1 서버 정보 요약

서버	정보	
	OS	x86_64
sm2	Mem	7990140 KB



(192.168.182.139)	CPU	4 개
	Core	4 개
	OS	x86_64
sm1	Mem	7990140 KB
(192.168.182.138)	CPU	4 개
	Core	4 개

3.1.2 서버 정보 : sm2(192.168.182.139)

구분	정보
호스트 이름	sm2
IP 주소	192.168.182.139
OS 버전	
Kernel 버전	3.10.0-1062.el7.x86_64
아키텍쳐(bit 수)	x86_64
CPU 정보	Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz
CPU 개수	4
Core 개수	4
CPU 당 Core 수	1
메모리(KB)	7990140 KB
Disk 사용량	Filesystem Size Used Avail Use% Mounted on
	devtmpfs 3.8G 0 3.8G 0% /dev
	tmpfs 3.9G 0 3.9G 0% /dev/shm
	tmpfs 3.9G 12M 3.8G 1% /run
	tmpfs 3.9G 0 3.9G 0% /sys/fs/cgroup
	/dev/mapper/rhel-root 17G 1.8G 16G 11% /



	/dev/sda1 1014M 150M 865M 15% /boot				
	tmpfs 781M 0 781M 0% /run/user/1104				
	tmpfs 781M 0 781M 0% /run/user/0				
Disk 정보	Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 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: 0x00008747				
	Device Boot Start End Blocks Id System				
	/dev/sda1 * 2048 2099199 1048576 83 Linux				
	/dev/sda2 2099200 41943039 19921920 8e Linux LVM				
	Disk /dev/mapper/rhel-root: 18.2 GB, 18249416704 bytes, 35643392 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 /dev/mapper/rhel-swap: 2147 MB_2147483648 bytes_4194304 sectors				
	Units = sectors of $1 \times 512 = 512$ bytes				
	Sector size (logical/physical): 512 bytes / 512 bytes				
	I/O size (minimum/optimal): 512 bytes / 512 bytes				
네트워크 섴정	ens33: flags=4163 <up,broadcast,running,multicast> mtu 1500</up,broadcast,running,multicast>				
	inet 192.168.182.139 netmask 255.255.255.0 broadcast 192.168.182.255				
	ether 00:0c:29:69:42:c8 txqueuelen 1000 (Ethernet)				
	RX packets 4528 bytes 1329412 (1.2 MiB)				
	RX errors 0 dropped 0 overruns 0 frame 0				
	TX packets 2316 bytes 277020 (270.5 KiB)				
	TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0				
	lo: flags=73 <up,loopback,running> mtu 65536</up,loopback,running>				
	inet 127.0.0.1 netmask 255.0.0.0				



	loop txqueuelen 1000 (Local Loopback)					
	RX packets 7829 bytes 801782 (782.9 KiB)					
	RX errors 0 dropped 0 overruns 0 frame 0					
	TX packets 7829 bytes 801782 (782.9 KiB)					
	TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0					
라우팅 정보	Kernel IP routing table					
	Destination Gateway Genmask Flags MSS Window irtt Iface					
	0.0.0.0 192.168.182.2 0.0.0.0 UG 0 0 0 ens33					
	192.168.182.0 0.0.0.0 255.255.255.0 U 0 0 0 ens33					

3.1.3 서버 정보 : sm1(192.168.182.138)

구분	정보
호스트 이름	sm1
IP 주소	192.168.182.138
OS 버전	
Kernel 버전	3.10.0-1062.el7.x86_64
아키텍쳐(bit 수)	x86_64
CPU 정보	Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz
CPU 개수	4
Core 개수	4
CPU 당 Core 수	1
메모리(KB)	7990140 KB
Disk 사용량	Filesystem Size Used Avail Use% Mounted on
	devtmpfs 3.8G 0 3.8G 0% /dev
	tmpfs 3.9G 152K 3.9G 1% /dev/shm
	tmpfs 3.9G 12M 3.8G 1% /run
	tmpfs 3.9G 0 3.9G 0% /sys/fs/cgroup
	/dev/mapper/rhel-root 17G 3.9G 14G 23% /



	/dev/sda1 1014M 150M 865M 15% /boot			
	tmpfs 781M 0 781M 0% /run/user/0			
Disk 정보	Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 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: 0x0007f37f			
	Device Boot Start End Blocks Id System			
	/dev/sda1 * 2048 2099199 1048576 83 Linux			
	/dev/sda2 2099200 41943039 19921920 8e Linux LVM			
	Disk /dev/mapper/rhel-root: 18.2 GB, 18249416704 bytes, 35643392 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 /dev/mapper/rhel-swap: 2147 MB, 2147483648 bytes, 4194304 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			
네트워크 설정	ens33: flags=4163 <up,broadcast,running,multicast> mtu 1500</up,broadcast,running,multicast>			
	inet 192.168.182.138 netmask 255.255.255.0 broadcast 192.168.182.255			
	ether 00:0c:29:9e:29:16 txqueuelen 1000 (Ethernet)			
	RX packets 776535 bytes 1063841117 (1014.5 MiB)			
	RX errors 0 dropped 0 overruns 0 frame 0			
	TX packets 125289 bytes 48328210 (46.0 MiB)			
	TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0			
	lo: flags=73 <up,loopback,running> mtu 65536</up,loopback,running>			
	inet 127.0.0.1 netmask 255.0.0.0			
	loop txqueuelen 1000 (Local Loopback)			



	RX packets 1861 bytes 1137226 (1.0 MiB)			
	RX errors 0 dropped 0 overruns 0 frame 0			
	TX packets 1861 bytes 1137226 (1.0 MiB)			
	TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0			
라우팅 정보	Kernel IP routing table			
	Destination Gateway Genmask Flags MSS Window irtt Iface			
	0.0.0.0 192.168.182.2 0.0.0.0 UG 0 0 ens33			
	192.168.182.0 0.0.0.0 255.255.255.0 U 0 0 0 ens33			

3.2 인스턴스 구성정보

호스트 IP	인스턴스명		포트 정보
192.168.182.138	admin11	HTTP 포트	8180
		AJP 포트	8109
		VirtualHost 도메인명	admin.opennaru.co.kr
192.168.182.138	front11	HTTP 포트	8280
		AJP 포트	8209
		VirtualHost 도메인명	front.opennaru.co.kr
192.168.182.139	admin21	HTTP 포트	8180
		AJP 포트	8109



		VirtualHost 도메인명	admin.opennaru.co.kr
192.168.182.139	front21	HTTP 포트	8280
		AJP 포트	8209
		VirtualHost 도메인명	front.opennaru.co.kr

3.3 인스턴스 접속 정보

호스트 IP	인스턴스명	포트 정보	
192.168.182.138	admin11	관리 콘솔	http://192.168.182.138:10090/manag
			er
		세션 테스트	http://192.168.182.138:8180/session
		테스트	http://192.168.182.138:8180/
		애플리케이션	
192.168.182.138	front11	관리 콘솔	http://192.168.182.138:10190/manag
			er
		세션 테스트	http://192.168.182.138:8280/session
		테스트	http://192.168.182.138:8280/
		애플리케이션	
192.168.182.139	admin21	관리 콘솔	http://192.168.182.139:10090/manag
			er
		세션 테스트	http://192.168.182.139:8180/session
		테스트	http://192.168.182.139:8180/
		애플리케이션	



192.168.182.139	front21	관리 콘솔	http://192.168.182.139:10190/manag
			er
		세션 테스트	http://192.168.182.139:8280/session
		테스트 애플리케이션	http://192.168.182.139:8280/



4. Apache Tomcat 설정 정보

4.1 인스턴스 구성

4.1.1 디렉터리 구성

스탠드얼론 모드 Tomcat 인스턴스 구성		
설치 디렉토리	/svc/test/tomcat	
도메인 디렉토리	/svc/test/tomcat/servers	
서버 인스턴스	/svc/test/tomcat/servers /\$SERVER_NAME	
Log 디렉토리	/svc/test/logs/tomcat	

4.1.2 인스턴스 이름 규칙

\$SERVER_NAME 명명 규칙은 아래와 같은 Rule 을 사용하여 구성한다.

/svc/test/tomcat/servers 이라는 디렉터리 하위에 인스턴스 들을 구성한다. domains 하위에 서버 인스턴스 이름으로 디렉터리를 만든다. 인스턴스의 이름은 다음 그림과 같이 서버의 이름 + 머신번호 + 인스턴스 번호와 같은 규칙을 사용하는 것이 좋다.

실제 서버의 이름은 서비스를 나타내는 고유명칭으로 변경하여 사용하면 된다. 머신 번호는 Tomcat 에 사용할 머신에 1 번부터 차례대로 붙인 번호를 사용하고, 인스턴스 번호는 같은 머신 내의 인스턴스 개수를 1 번부터 번호를 붙인다.

이렇게 인스턴스 이름을 지정하면 모든 인스턴스가 고유한 이름을 갖게 된다. 이 고유한 인스턴스 이름은 클러스터링에서 세션 ID, 로그 파일의 이름, nohup 로그 파일의 이름 등에 사용하게 된다. 장애 상황이나 로그 파일를 확인하는 등 어떤 상황에서도 서버 이름만 보면 어떤 머신의 어떤 인스턴스의 것인지 금방 구분할 수 있게 된다.





또, 포트 오프셋 번호는 '인스턴스의 번호 * 100'을 사용하면, 포트를 구분하기 쉽다. 기본 포트셋 + 포트오프셋으로 사용하는 포트가 지정되는데, 사용하는 대부분의 포트 번호가 100 번대가 0 이기 때문에 해당 인스턴스가 사용하는 번호를 기억하기 쉽다.

예를 들어 인스턴스 이름이 'frontSvr24'이라면, 2 번째 Tomcat 머신의 4 번째 인스턴스를 의미하며, 이 인스턴스가 사용하는 HTTP 포트는 8480(8080 + 400) 포트를 사용하고, AJP 포트는 8409(8009 + 400) 포트를 사용한다.



4.2 Tomcat 운영 스크립트

4.2.1 Tomcat 스크립트

Tomcat 인스턴스 구성	
운영 스크립트 위치	/svc/test/tomcat/servers /\$SERVER_NAME/*.sh

4.3 환경 설정 파일

4.3.1.1 주요 설정 파일

스크립트 파일	
env.sh	Tomcat 운영환경 주요 환경 설정 스크립트
start.sh	Tomcat 인스턴스 실행 스크립트
shutdown.sh	Tomcat 인스턴스 정상 종료 스크립트
kill.sh	Tomcat 강제 종료 스크립트
nohup.sh	Tomcat 로그의 tail 보기 스크립트
tail.sh	Tomcat server.log 파일 tail 보기 스크립트
status.sh	Tocmat 인스턴스가 실행중인지 체크하는 스크립트
dump.sh	인스턴스 장애시 Java Process 의 Thread Dump 를 받기 위한
	스크립트

서버 마다 설정이 다른 부분은 KHAN [provisioning]에서 자동으로 설정을 변경하여 구성한다.

● env.sh 파일

env.sh 는 Tomcat 환경설정을 위한 파일이다. 각 인스턴스의 모든 설정은 이 파일에서 하면 된다. Tomcat 인스턴스마다 각각의 인스턴스의 이름, 디렉터리의 이름, 포트 오프셋 등을 설정하면 된다.



또 이 파일에는 JVM 의 메모리 옵션, GC 옵션, OufOfMemory 오류가 발생할 때 Heapdump 를 출력하는 옵션 등이 설정되어 있다.

#!/bin/sh #
 # KHAN [provisioning] http://www.opennaru.com/ # Tomcat 7.0.42 #
 # contact : service@opennaru.com # Copyright(C) 2013, opennaru.com, All Rights Reserved. #
DATE=`date +%Y%m%d%H%M%S`
Tomcat Directory export SERVER_NAME={{ INSTANCES[seq int][item int-1]['instance']['name'] }} export CATALINA_HOME={{ SVC_WAS_HOME }}/{{ EWS_DIR_NAME }}/tomcat export CATALINA_BASE={{ SVC_DOMAIN_HOME}}/\$SERVER_NAME
export PORT_OFFSET={{ INSTANCES[seq int][item int-1]['instance']['portOffset'] }}
export TOMCAT_USER={{ TOMCAT_USER }} export TOMCAT_LOGDIR=\$CATALINA_BASE/logs
Port Configuration let HTTP_PORT=8080+\$PORT_OFFSET export HTTP_PORT
let AJP_PORT=8009+\$PORT_OFFSET export AJP_PORT
let SSL_PORT=8443+\$PORT_OFFSET export SSL_PORT
let SHUTDOWN_PORT=8005+\$PORT_OFFSET export SHUTDOWN_PORT
let JMX_PORT=8086+\$PORT_OFFSET export JMX_PORT



JVM Options : Server export JAVA_OPTS="-server \$AGENT_OPTS \$JAVA_OPTS"

JVM Options : Memory export JAVA_OPTS=" \$JAVA_OPTS -Xms{{ jvm_heap_min }} -Xmx{{ jvm_heap_max }} -XX:MaxPermSize={{ jvm_permgen_max }} -Xss256k"

export JAVA_OPTS=" \$JAVA_OPTS -XX:+PrintGCTimeStamps " export JAVA_OPTS=" \$JAVA_OPTS -XX:+PrintGCDetails " export JAVA_OPTS=" \$JAVA_OPTS -Xloggc:\$TOMCAT_LOGDIR/gclog/gc_\$DATE.log " export JAVA_OPTS=" \$JAVA_OPTS -XX:+UseParallelGC " #export JAVA_OPTS=" \$JAVA_OPTS -XX:+UseConcMarkSweepGC " export JAVA_OPTS=" \$JAVA_OPTS -XX:+ExplicitGCInvokesConcurrent " export JAVA_OPTS=" \$JAVA_OPTS -XX:+ExplicitGCInvokesConcurrent " export JAVA_OPTS=" \$JAVA_OPTS -XX:+HeapDumpOnOutOfMemoryError " export JAVA_OPTS=" \$JAVA_OPTS -XX:HeapDumpPath=\$TOMCAT_LOGDIR/heapdump "

Linux Large Page Setting
#export JAVA_OPTS = \$JAVA_OPTS -XX:+UseLargePages "

#export JAVA_OPTS=" \$JAVA_OPTS -verbose:gc"
export JAVA_OPTS=" \$JAVA_OPTS -Djava.net.preferIPv4Stack=true"
export JAVA_OPTS=" \$JAVA_OPTS -Dorg.tomcat.resolver.warning=true"
export JAVA_OPTS=" \$JAVA_OPTS -Dsun.rmi.dgc.client.gcInterval=3600000 "
export JAVA_OPTS=" \$JAVA_OPTS -Dsun.rmi.dgc.server.gcInterval=3600000"
export JAVA_OPTS=" \$JAVA_OPTS -Djava.awt.headless=true"
export JAVA_OPTS=" \$JAVA_OPTS -Djava.awt.headless=true"

export JAVA_OPTS="\$JAVA_OPTS -DSERVER=\$SERVER_NAME" export JAVA_OPTS="\$JAVA_OPTS -Dhttp.port=\$HTTP_PORT" export JAVA_OPTS="\$JAVA_OPTS -Dajp.port=\$AJP_PORT" export JAVA_OPTS="\$JAVA_OPTS -Dssl.port=\$SSL_PORT" export JAVA_OPTS="\$JAVA_OPTS -Dshutdown.port=\$SHUTDOWN_PORT" export JAVA_OPTS="\$JAVA_OPTS -Djava.library.path=\$CATALINA_HOME/lib/"



echo "AJP_PORT=\$AJP_PORT"



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 Tomcat 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.
0	서버 측에서 닫은 TCP 연결을 끊기 전에
	확인하는 횟수를 줄인다. 기본값은 7 로
	50 초~16 분 정도 걸린다.
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.
	0

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 prio# - rtprio - max realtime	rity allowed to rai e priority	se to values: [-20, 19]
#		
# <domain> <type> <ite< td=""><td>em> <value< td=""><td>></td></value<></td></ite<></type></domain>	em> <value< td=""><td>></td></value<>	>
#		
#* soft core	0	
#* hard rss	10000	
#@student hard npro	oc 20	
#@faculty soft nproc	20	
#@faculty hard nproc	50	
#ftp hard nproc	0	
#@student - maxlo	gins 4	
tomcat hard	nofile	<mark>65536</mark>
tomcat soft	nofile	<mark>65536</mark>
tomcat soft	nproc	<mark>2047</mark>
tomcat hard	nproc	<mark>16384</mark>
# End of file		

6. Apache Tomcat 주요 설정값

6.1 JDK 설치

Oracle JDK openjdk.1.8 rpm 버전을 설치하였다. JDK 는 alternatives 명령을 이용하여 리눅스 system 의 모든 사용자가 기본으로 설치된 java openjdk.1.8 을 사용할 수 있도록 설정하였다. JDK 는 현재 가장 최신 버전을 사용하였다.



6.2 Native 모듈 설치

Tomcat Native (TC-Native)는 JNI(Java Native Interface)를 이용하여 Tomcat 내의 Tomcat 모듈의 코어기능을 자동 설치 과정에서 libaio, apr, apr-util, openssl RPM 모듈이 설치되었다. Java 가 아닌 Native 코드를 사용할 수 있도록 하여, 전체적으로 서버 성능을 향상시킬 수 있다. 또, Tomcat 설치시 Tomcat Native Component 들이 설치되었다. 설치 확인은 \$TOMCAT_HOME/lib 에 libtcnative-1.so 파일이 설치되었는지 확인하면 된다.

6.2.1 추가 설치 패키지

yum install apr apr-util openssl

6.3 Thread Pool 설정

HTTP/AJP 연결에 대한 Thread Pool 을 아래와 같이 400 개 스레드를 사용할 수 있도록 설정하였다.

<Connector port="\${http.port}" protocol="HTTP/1.1" redirectPort="\${ssl.port}" enableLookups="false" URIEncoding="UTF-8" acceptCount="200" maxThreads="400" maxKeepAliveRequests="200" backlog="120" connectionTimeout="10000"/>

<!-- Define an AJP 1.3 Connector on port 8009 --> <Connector port="\${ajp.port}" protocol="AJP/1.3" redirectPort="8443" enableLookups="false" URIEncoding="UTF-8" maxThreads="400" maxKeepAliveRequests="200" backlog="120" connectionTimeout="10000" acceptCount="200" minSpareThreads="25" />





7. Tomcat 운영 방법

Tomcat 관리자인 'tomcat' 계정에 alias 를 설정하였기 때문에, s1, s2 만 입력하면 인스턴스 디렉터리로 이동한다.

다음과 같이 스크립트파일들을 실행하여 Tomcat 인스턴스를 관리할 수 있다.

스크립트	설명
\$./start.sh	인스턴스 시작
\$./stop.sh	인스턴스 정상 종료
\$./kill.sh	인스턴스 강제 종료
\$./status.sh	인스턴스 수행 중인지 체크
\$./nohup.sh	인스턴스 nohup 로그를 tail 로 출력
\$./tail.sh	인스턴스의 서버 로그를 tail 로 출력
\$./dump.sh	인스턴스의 스레드 덤프 출력

8. MBean 그래프 모니터링

KHAN [provisioning]으로 설치한 인스턴스들은 hawt.io Chrome Extension 을 사용하여 MBean 정보를 그래프로 모니터링 할 수 있다.

Chrome 브라우저의 Extension 을 설치하여 KHAN [provisioning]으로 설치한 인스턴스에 접속하여 모니터링한다.

8.1 hawtio Chrome Extension 설치

https://chrome.google.com/webstore/에 접속하여 hawtio 를 검색하여 설치한다.



الله المراجع ال مع المراجع المرا	awtio ★★★ (০) <u>유ছিলল</u> আক্র	: hawtio			+ 무료 <
개요	세부정보	리뷰	관련 프로그램		8+1
O	x t Dashboard Health Infinispan jet Dashboard Health Infinispan jet Dashboard Health Infinispan jet personelikecerd personelikecerd personelikecerd personelikecerd personelikecerd personelikecerd personelikecerd personelikecerd personelikecerd personelikecerd personelikecerd personelikecerd personelikecerd	apache.camei-camei-1-routes-K22fox y JMX Juhit Logs Quarz ag Trace @ Profile ⊞ Attribute to to to to to to to to to to	622 Threads Wild Is # Operations @ Chart		Don't cha wish your console was hawt like me? Hawtio is a modular web console for managing your Java stuff. Once installed it can connect to any Java process which has a jolokia agent installed (http://jolokia.org/) and view its JMX MBeans and work with its various services such as Apache Camel, Apache ActiveMQ or fabric8. hawtio has lots of plugins for working with various Java frameworks and tools: http://hawt.io/plugins/index.html for more help check out the documentation: http://hawt.io/docs/index.html
맵을 설치한 사용자의	의 다른 설치 항목				
Quick Note	온라인 ****	음막 알람 시계 🛛 🕡	속도 테스트 ★★★★★★ (614)	Speed Tes	u i i p 주소 ★★★★★ (217)

구글 Web Store 에서 설치되지 않는다면, <u>http://hawt.io/getstarted/index.html</u> 에서 Chrome Extension 을 다운로드 하여 압축을 해제한 뒤, 브라우저에서 주소창에 chrome://extensions/ 입력한다.

우측 상단의 '개발자 모드'를 선택한 후, '압축해제된 확장 프로그램 로드..' 버튼을 클릭하여 압축을 해제한 디렉터리를 입력하면 설치된다.

8.2 hawtio 실행하기

브라우저에 chrome://apps 을 입력하면 hawtio 애플리케이션을 클릭하면 실행된다.

서버 접속 정보에 KHAN [provisioning]으로 설치한 인스턴스의 IP 주소와 포트 정보, Path 는 jolokia 를 입력하고, 설치한 인스턴스의 관리자 계정 정보를 입력하면 새로운 탭이 열리며 서버에 접속된다.

서버 접속 정보는 저장하여 관리할 수 있다.



다음과 같이 서버의 접속정보를 입력하여 접속한다.

Saved Connections		
Connections:	test 🔹	2 - +
Connection Settings		
Connection name:	test	
Scheme:	http 🔹	
Host:	192.168.0.23	
Port:	8180	
Path:	jolokia	
User name:	wasadm	
Password:		
Co	nnect to remote server Save	

8.3 MBean 모니터링

이제 접속된 인스턴스의 정보를 모니터링 할 수 있다. Tomcat 에서 제공하는 모든 MBean 에 대한 정보를 모니터링하고 MBean 메소드를 실행하고 관리할 수 있다. 또, 다음과 같이 서버 인스턴스에 대한 대시보드를 확인할 수 있다.



🔄 📎 🥝 🗋 ch	rome-extension://nlaokfcnhjbpikooiddimhlni	abjhpmk/ind	ex.html?url=http%3A%2F%2F192.168	3.0.22%3A8180%2Fjolokia#/dashboard/	id/4e9d11617:숬 🝇	関 📒 🤏 🖂 🎯 🔛	★ 👌 🛎 ≡
💘 hawtio						Ģ	🛛 🍐 wasadm 🕯
Connect Dashbo	ard JBoss JMX Threads						
Monitor							🕼 Manage
Operating System		x	System CPU Load	Java Heap Memory	× Jav	va Non Heap Memory	×
Property	Value	۲	1.0 • • • • • • • • • • • •				
Arch	i386		0.9 - <u>F</u>				17.7% Used (56576048)
Available proce	1		0.8 - 0				
Committed virtu	821960704		0.6-	44.5%			
Free physical m	126431232		0.5 -	06215504).			
Free swap spac	2089750528		0.4 -		55.5% Used (132)		
Max file descript	65536		0.3 -				
Name	Linux		0.2-			82.3%	
Object name	java.lang:type=OperatingSystem		0.0-		Fre	e (262191056)	
Open file descri	344		.540				
Process cpu load	0.0893970893970894						×
Process cpu time	136980000000		10:27 10:28 10:29 10:30	10:31 10:32 10:33 10:34 10	:35 10:36 10:37 1	LO:38 10:39 10:40	10:41 10:42
System cpu load	0		Peak thread count				430
System load av	0.09		Current thread cpu time	n is a company and choor this and	eteritik (hand bester bittik)	illean an the UNA Child Albert when	onini, 11, 11, 180M
Total physical m	1587994624		Daemon thread count				78
Total swap spac	2113921024		Total started thread count				510
Version	2.6.32-431.1.2.0.1.el6.i686		Current thread user time	المراجع والمراجع والمراجع والمراجع والمراجع والمراجع والمراجع والمراجع والمراجع والمراجع	adath na dhuadh	anter a sur i Milel i liticitador	70M
			Thread count				430
			10:27 10:28 10:29 10:30	10:31 10:32 10:33 10:34 10	35 10:36 10:37 1	0:38 10:39 10:40	10:41 10:42
			TOUT TOUTO TOUTO	TOIDT TOIDT TOIDD TOID1 TO	100 20100 20101 2	10100 10100 10110	10112

KHAN [provisioning]으로 설치하면 함께 설치되는 Stuck Thread 정보도 모니터링 할 수 있다.

	/nlaokfcnhjbpi	kooiddimhlnia	bjhpmk/ind	dex.html?	url=http;	63A%2F%	2公 ն	k ,	۹ 🖂	. 🛛 🤤	a 🚖 1	ى ق	≝ ≣
💘 hawtio										Ģ	0	å v	vasadm 🗸
Connect Dashboard JBoss	JMX Thre	eads											
JMX Tree	I Attrib	utes 💋 Op	erations	Jil Charl	Ed	it Chart						C	8
 com.opennaru.khan.thread stuckThreads com.oracle.idbc 	.0:30 10 Active three):31 10:32 ead count	10:33	10:34	10:35	10:36	10:37	10:38	10:39	10;40	10:41	10	:42 2.0
 > com.sun.management > iava.lang 	Hogging t	hreshold											88 0.0
> 는 java.nio > 는 java.util.logging	Stuck thre	ad count											0.0
> 📄 jboss.as	Stuck thre	shold											800
 jboss.as.expr jboss.infinispan jboss.jsr77 jboss.jta jboss.modules jboss.msc 	.0:30 10	.31 10:32	10:33	10:34	10:35	10:36	10:37	10:38	10:39	10:40	10:41	10	:42

현재 실행 중인 스레드에 대한 Stack Trace 정보도 모니터링 할 수 있다.



📎 🌀 🗈 chrome-extension://nlaokfcnhjbpikooiddimhlniabjhpmk/index.html?url=http%3A%2F%2☆ 🝇 閣 茾 🍕 🖂 🍳 🎼 ★ 1	å ७ ≡
hawtio 🖓 🔍	🛔 wasadm 🗸
nnect Dashboard JBoss JMX Threads	
Total: 430 Runnable: 26 Timed waiting: 21 Waiting: 383 (Object Monitor Usage) (Synchronizer Usage) (Thread Contention Monitoring	
12 III Incoming-11,shared=udp	
Thread-29 (HornetQ-server-HornetQServerImpl::serverUUID=cfc3b26b-b5af-11e3-8686-f3d41053c72d-22170601)	
10 III http:-thread-pool-threads - 250	
09 III http:-thread-pool-threads - 249	
08 III http:-thread-pool-threads - 248	
07 III http:/thread-pool-threads - 247	
i06 III http:/thread-pool-threads - 246	
505 III http-thread-pool-threads - 245	
04 III http:thread-pool-threads - 244	
03 III http:thread-pool-threads - 243	
02 III http:/thread-pool-threads - 242	
01 III http:thread-pool-threads - 241	
00 III http-thread-pool-threads - 240	
199 III http-thread-pool-threads - 239	
198 III http:thread-pool-threads - 238	
197 III http:thread-pool-threads - 237	
196 💵 http-thread-pool-threads - 236	

KHAN [provisioning]으로 설치할 때 자동으로 설치된 JVM Memory MBean 을 사용하면 다음과 같이 Java 메모리의 사용량 %로 모니터링 할 수 있다.

💘 hawtio	ų	Θ	🛔 wasadm 🕻
Connect Dashboard JBoss	JMX Threads		
JMX Tree	I≣ Attributes Ø Operations III Chart ✿ Edit Chart	~	C X
🗸 📂 com.opennaru.khan.memory	10:49 10:50 10:51 10:52 10:53 10:54 10:55 10:56 10:57	10:58	10:59
🔅 java V 📂 com.opennaru.khan.thread	Heap used	iiiil	81M
🔅 stuckThreads	Heap max		240M
com.oracle.jdbc	Noniheao max		320M
com.sun.management	Non neap used		76M
java.nio	Non heap used bercent		24
java.util.logging	Heap used percent		34
🔉 🚞 jboss.as			
🔉 🚞 jboss.as.expr	10:49 10:50 10:51 10:52 10:53 10:54 10:55 10:56 10:57	10:58	10:59
🔉 📄 jboss.infinispan			



9. 도움이 필요하십니까?

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





10. References

- Red Hat Documentation
- http://docs.redhat.com/
- 오픈나루 고객지원 포탈
- <u>http://support.opennaru.com</u>
- 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 OPENMARU Installer 을 이용하여 생성된 문서입니다. 본 문서에 대한 저작권은 오픈나루 주식회사에 있습니다.