


# 공개 S/W 기술지원 ETRI

한국소프트웨어진흥원  
공개SW기술지원센터

## <Revision 정보>

일자	VERSION	변경내역	작성자
2007. 8. 28	0.1	초기 작성	손명선

 <p>HTTP://WWW.OSS.ORG.KR</p>	공개SW 기술지원	
	구분 : 기술지원	단계:
	작성자: 손명선	작성일: 2007.8. 28
	검토자:	검토일:
	승인자:	승인일:

#### 1. 대상기업/기관 정보

구분	항목	내용	비고
기업/기관 정보	지역	대전광역시 유성구 가정로 138	
	기업/기관 명칭	한국전자통신연구원	
	부서	정보보호연구단 네트워크보안그룹	
	직책	선임연구원	
	담당자 이름	임재덕	
	전화번호 / 팩스번호	042-860-1522 / 042-860-5611	
	E-Mail	jdscol92@erti.re.kr	

#### 2. 대상기업/기관 지원사항

구분	항목	내용	비고
기업/기관 지원사항	접수내용	ETRI 정보보호연구단의 포털 사이트용 보안솔루션 개발을 위한 공개SW기반 테스트 포털 개발 시스템 구축 지원	
	지원내역	<p>1. 공개SW기반 포털 사이트 개발환경 구축 설계 분석 및 환경 파악</p> <ul style="list-style-type: none"> <li>- WAS : Apache, Tomcat, PHP, Java기반 총 5대</li> <li>- Mail : Sendmail 기반의 이중화 구조</li> <li>- DNS : Bind 기반의 이중화 구조</li> <li>- DB : MySQL 기반의 이중화 구조</li> <li>- 프로비저닝 : DHCP, PXE 기반</li> <li>- File : Hadoop 기반의 분산파일시스템 구조</li> </ul> <p>2. 하드웨어 납품 및 마운팅</p> <ul style="list-style-type: none"> <li>- TPM 지원 7대를 포함한 총 20대의 서버</li> </ul> <p>3. DHCP &amp; TFTP &amp; PXE 설치를 이용한 프로비저닝 환경 구축 및 네트워크 부팅을 통한 리눅스 자동 설치</p>	

		<pre> - dhcp 설치 및 설정  # cd /data1/os/rh4_x64_up2/RedHat/RPMS  # rpm -Uvh dhcp-3.0.1-12_EL.x86_64.rpm # rpm -Uvh dhcp-devel-3.0.1-12_EL.x86_64.rpm  # vi /etc/dhcpd.conf  -----  ### DHCP Server 설정 ##### # # 설치 환경에 맞게 네트워크 정보 수정 # ##### #####  ddns-update-style interim; ignore client-updates; default-lease-time 600; max-lease-time 7200; option subnet-mask 255.255.255.0; option broadcast-address 192.168.1.255; option routers 192.168.1.254; #option domain-name-servers 192.168.1.254; #option domain-name "cluster.bj";  # pxe 설정 시 아래 추가#####  allow booting; allow bootp;  class "pxeclients" {     match if substring (option vendor-class-identifier, 0, 9) = "pxeClient";     next-server 192.168.1.254;     filename "linux-install/pxelinux.0"; </pre>	
--	--	--	--

		<pre> }  ##### ##### ### subnet 192.168.1.0 netmask 255.255.255.0 {     range 192.168.1.100 192.168.1.253; }  -----  # /etc/rc.d/init.d/dhcpd restart  - NFS 설정 운영체제 패키지가 있는 디렉토리를 NFS로 서비스 해야한다. # vi /etc/exports ----- /data1/os/rh4_x64_up2                *(rw) -----  - tftp &amp; pxe 패키지 설치 # cd /data1/os/rh4_x64_up2/RedHat/RPMS/  # rpm -Uvh tftp-0.39-1.x86_64.rpm # rpm -Uvh tftp-server-0.39-1.x86_64.rpm # rpm -Uvh system-config-netboot-0.1.32-1_EL4.x86_64.rpm  - tftp 설정 # vi /etc/xinetd.d/tftp ----- service tftp {     disable = no    # disable = yes -&gt; no      socket_type      = dgram     protocol         = udp </pre>	
--	--	---	--

		<pre> wait                = yes user                = root server              = /usr/sbin/in.tftpd server_args         = -s /tftpboot per_source          = 11 cps                 = 100 2 flags               = IPv4 } -----  - pxe 설정 # mkdir /tftpboot/linux-install/rhes4  # cd /data1/os/rh4_x64_up2/images/pxeboot/  # cp vmlinuz /tftpboot/linux-install/rhes4/ # cp initrd.img /tftpboot/linux-install/rhes4/  # vi /tftpboot/linux-install/pxelinux.cfg/default -----  default local timeout 10 prompt 1 display msgs/boot.msg F1 msgs/boot.msg F2 msgs/general.msg F3 msgs/expert.msg F4 msgs/param.msg F5 msgs/rescue.msg F7 msgs/snake.msg  LABEL local     localboot 1  LABEL node01 KERNEL rhes4/vmlinuz APPEND initrd=rhes4/initrd.img ramdisk_size=10000 ks=nfs:192.168.1.254:/data1/os/rh4_x64_up2/ks1.cfg </pre>	
--	--	---	--

		<pre>ksdevice=eth0  LABEL node02 KERNEL rhes4/vmlinuz APPEND initrd=rhes4/initrd.img ramdisk_size=10000 ks=nfs:192.168.1.254:/data1/os/rh4_x64_up2/ks2.cfg ksdevice=eth0  -----  - kickstart 설정 # vi /data1/os/rh4_x64_up2/ks1.cfg -----  # kickstart 자동 설치 install  # nfs 로 설치 이미지 다운 nfs --server=192.168.1.254 --dir=/data1/os/rh4_x64_up2  # 언어 설정 lang en_US.UTF-8  # 지원 언어 설정 langsupport --default=ko_KR.UTF-8 en_US.UTF-8 ko_KR.UTF-8  # 키보드 설정 keyboard us  # 네트워크 설정 , 네트워크 설정 시 해당 정보를 수정해야 한다.  network --device eth0 --bootproto static --ip 192.168.1.1 --netmask 255.255.255.0 --gateway 192.168.1.254 --hostname node01.cluster.bj # root 패스워드 ( no touch ) rootpw --iscrypted \$1\$P.9z.LGA\$MOrwcO86rCh2lOt71tqlq1</pre>	
--	--	--	--

		<pre># 방화벽 설정 firewall --disabled  # 보안 설정 selinux --disabled authconfig --enablesshadow --enablemd5  # 시간대 설정 timezone Asia/Seoul  # 설치 모드 설정 ( 만일 그래픽 모드에서 설치를 하실려면 아래 text를 삭제하면 된다.) text  skipx bootloader --location=mbr --append="rhgb quiet"  # 파티션 설정 clearpart --all part /boot --fstype ext3 --size=200 part /usr --fstype ext3 --size=10000 part swap --size=4000 part / --fstype ext3 --size=3000 --asprimary part /var --fstype ext3 --size=2000  # 패키지 설정 %packages @ compat-arch-development @ engineering-and-scientific @ admin-tools @ editors @ emacs @ system-tools @ korean-support @ gnome-software-development @ text-internet @ x-software-development @ legacy-network-server</pre>	
--	--	---	--



		<pre>@ dns-server @ gnome-desktop @ dialup @ ftp-server @ compat-arch-support @ legacy-software-development @ smb-server @ base-x @ server-cfg @ sound-and-video @ development-tools @ graphical-internet -evolution -rusers tetex-xdvi e2fsprogs pvm sysstat gftp -rwho rsh-server vnc iptraf -evolution-webcal kernel-devel kernel-smp-devel mc nmap-frontend thunderbird  %post -----</pre>	
		<p>4. 웹개발(Apache, PHP, MySQL) 환경 구축</p> <ul style="list-style-type: none"> <li>- Apache 설치</li> </ul> <pre>[root@node04 httpd-2.2.0]# ./configure --prefix=/usr/local/apache2</pre>	

		<pre>[root@node04 httpd-2.2.0]# make &amp;&amp; make install</pre> <p>/usr/local/apache2 에 정상적인 설치 상태를 확인 한다.</p> <p>- MySQL 설치 및 설정</p> <pre>root@node04 mysql-4.0.24]# ./configure --prefix=/usr/local/mysql W --localstatedir=/usr/local/mysql/data --with-mysqld-user=mysql --with-charset=euc_kr</pre> <pre>[root@node04 mysql-4.0.24]# make &amp;&amp; make install</pre> <p>/usr/local/mysql 에 설치 상태를 확인 한다.</p> <pre>[root@node04 mysql-4.0.24]# /usr/local/mysql/bin/mysql_install_db [root@node04 mysql-4.0.24]# userdel -r mysql [root@node04 mysql-4.0.24]# groupdel mysql [root@node04 mysql-4.0.24]# groupadd mysql [root@node04 mysql-4.0.24]# adduser -g mysql -d /usr/local/mysql/data -s /bin/false mysql [root@node04 mysql-4.0.24]# chown -R mysql. /usr/local/mysql/data  [root@node04 mysql-4.0.24]# cp \$SRC_PATH/mysql-4.0.24/support-files/my-medium.cnf /etc/my.cnf  [root@node04 mysql-4.0.24]# vi /usr/local/mysql/share/mysql/mysql.server ----- \$bindir/mysqld_safe --datadir=\$datadir --pid-file=\$pid_file  위 행을 찾아서 뒤에 아래 옵션을 추가로 붙여준다.  --language=korean --safe-show-database &amp;</pre>	
--	--	--	--

		<pre> [root@node04 mysql-4.0.24]# cp /usr/local/mysql/share/mysql/mysql.server /etc/rc.d/init.d/mysqld [root@node04 mysql-4.0.24]# ln -s /etc/rc.d/init.d/mysqld /etc/rc.d/rc3.d/S90mysqld  [root@node04 mysql-4.0.24]# /etc/rc.d/init.d/mysqld start  mysql root password 설정한다.  초기에는 mysql root 패스워드가 Null 상태로 셋팅된다. 보안을 위해 Mysql 설치 후에 반드시 mysql root 암호를 변경해 주어야 한다.  [root@node04 mysql-4.0.24]# mysqladmin -u root password '패스워드' </pre> <p>- PHP 설치</p> <p>PHP에서 사용되어지는 기본 Library를 설치</p> <pre> [root@node04 libpng-1.2.12]# cp scripts/makefile.linux Makefile [root@node04 libpng-1.2.12]# make test &amp;&amp; make install [root@node04 libpng-1.2.12]# vi /etc/ld.so.conf  PHP를 이용하여 Imap 라이브러리 설치 및 연동 [root@node04 APMj]# tar xzvf imap-2004.tar.Z [root@node04 APMj]# mv imap-2004 /usr/local/imap [root@node04 APMj]# cd /usr/local/imap/ [root@node04 imap]# make slx SSLTYPE=none [root@node04 imap]# cp imapd/imapd /usr/sbin [root@node04 imap]# cp ipopd/ipop3d /usr/sbin  [root@node04 imap]# vi /etc/xinetd.d/imapd </pre> <pre> service imap </pre>	
--	--	---	--

		<pre> { disable = no socket_type = stream wait = no user = root server = /usr/sbin/imapd log_on_success += DURATION USERID log_on_failure += USERID nice = -2 }  -----  [root@node04 imap]# vi /etc/xinetd.d/ipop3d  -----  service pop3 { disable = no socket_type = stream wait = no user = root server = /usr/sbin/ipop3d log_on_success += USERID log_on_failure += USERID nice = -2 }  -----  [root@node04 imap]# /etc/rc.d/init.d/xinetd restart  PHP 설치 [root@node04 php-4.3.10]# ./configure --with-apxs2=/usr/local/apache2/bin/apxs W --with-mysql=/usr/local/mysql --with-jpeg --with-png-dir=/usr/local/lib W --with-gif-dir=/usr/lib --with-zlib --with-gd --with-freetype --with-xml W --with-mod_charset --with-language=korean </pre>	
--	--	---	--

		<pre>--with-charset=euc_kr W --enable-ftp --enable-sockets --disable-debug --enable-system W --enable-track-vars --enable-calendar --enable-magic-quotes W --with-imap=/usr/local/imap  [root@node04 php-4.3.10]# make &amp;&amp; make install</pre>	
		<p>5. WAS(Tomcat 설치)</p> <p>- java 환경 구축</p> <pre>[root@node04 APMj_new]# ./jdk-1_5_0_06-linux-i586.rpm.bin . . For inquiries please contact: Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. (LFI#141623/Form ID#011801)  Do you agree to the above license terms? [yes or no] yes . . Creating jdk1.5.0_06/lib/tools.jar Creating jdk1.5.0_06/jre/lib/rt.jar Creating jdk1.5.0_06/jre/lib/jsse.jar Creating jdk1.5.0_06/jre/lib/charsets.jar Creating jdk1.5.0_06/jre/lib/ext/localedata.jar Creating jdk1.5.0_06/jre/lib/plugin.jar Creating jdk1.5.0_06/jre/lib/javaws.jar Creating jdk1.5.0_06/jre/lib/deploy.jar  Done.</pre>	

		<pre>[root@node04 APMj_new]# mv jdk1.5.0_06/ /usr/local</pre> <pre>[root@node04 APMj_new]# cd /usr/local</pre> <pre>[root@node04 local]# ln -sf jdk1.5.0_06/ java</pre> <pre>[root@node04 local]# ln -sf /usr/local/jdk1.5.0_06/jre/ jre</pre> <p>/etc/profile 에 java 환경 설정을 추가한다.</p> <pre>[root@node04 local]# vi /etc/profile</pre> <pre>-----</pre> <pre>. export JAVA_HOME=/usr/local/java export PATH=/usr/local/java/bin:\$PATH</pre> <pre>-----</pre> <p>같은 쉘환경에서 바로 작업을 진행하기 위해 java 환경 설정을 적용한다.</p> <pre>[root@node04 ]# export JAVA_HOME=/usr/local/java</pre> <pre>[root@node04 ]# export PATH=/usr/local/java/bin:\$PATH</pre> <pre>[root@node04 ]# java -version</pre> <pre>java version "1.5.0_06"</pre> <pre>Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_06-b05)</pre> <pre>Java HotSpot(TM) Server VM (build 1.5.0_06-b05, mixed mode)</pre> <p>- tomcat5 설치</p> <pre>[root@node04 APMj_new]# tar xzvf</pre> <pre>apache-tomcat-5.5.12.tar.gz</pre> <pre>[root@node04 APMj_new]# mv apache-tomcat-5.5.12 /usr/local/tomcat</pre> <pre>[root@node04 APMj_new]# vi /etc/profile</pre> <pre>-----</pre> <pre>.</pre>	
--	--	---	--

		<pre>export CATALINA_HOME=/usr/local/tomcat export PATH=\$PATH:\$CATALINA_HOME/bin  -----  - jakarta-tomcat-connectors-jk2 설치  [root@node04 APMj_new]# tar xzvf jakarta-tomcat-connectors-jk2-2.0.4-src.tar.gz [root@node04 APMj_new]# cd jakarta-tomcat-connectors-jk2-2.0.4-src/jk/native2/ [root@node04 native2]# ./configure --with-apxs2=/usr/local/apache2/bin/apxs [root@node04 native2]# make [root@node04 native2]# cd ../build/jk2/apache2/ [root@node04 apache2]# /usr/local/apache2/bin/apxs -n jk2 -i mod_jk2.so [root@node04 ~]# ls /usr/local/apache2/modules/mod_jk2.so /usr/local/apache2/modules/mod_jk2.so  - httpd.conf 설정  [root@node04 ~]# vi /usr/local/apache2/conf/httpd.conf  -----  . . # Example: # LoadModule foo_module modules/mod_foo.so #  LoadModule jk2_module modules/mod_jk2.so LoadModule php4_module      modules/libphp4.so . .  ServerName 192.168.123.64 . .</pre>	
--	--	--	--

		<pre> &lt;Directory "/usr/local/tomcat/webapps"&gt;     Options Indexes FollowSymLinks     AllowOverride None     Order allow,deny     Allow from all &lt;/Directory&gt;  . .  &lt;IfModule dir_module&gt;     DirectoryIndex index.html index.jsp index.php &lt;/IfModule&gt;  . .      AddType application/x-httpd-php .html .php3 .php4     .inc .phtml .php .ph     AddType application/x-httpd-php-source .phps . . ----- </pre>	
		<p>- httpd-autoindex.conf 설정</p> <pre> [root@node04 ~]# vi /usr/local/apache2/conf/extra/httpd-autoindex.conf ----- . . # We include the /icons/ alias for FancyIndexed directory listings.  If # you do not use FancyIndexing, you may comment this out. # Alias /icons/ "/usr/local/apache2/icons/" Alias /jsp-examples/ "/usr/local/tomcat/webapps/jsp-examples/" Alias /servlets-examples/ "/usr/local/tomcat/webapps/servlets-examples/" </pre>	



		<pre> . . -----  - workers2.properties 설정 [root@node04 ~]# vi /usr/local/apache2/conf/workers2.properties -----  [channel.socket:localhost:8009] port=8009 host=localhost  [ajp13:localhost:8009] channel=channel.socket:localhost:8009  [uri:/*.jsp] worker=ajp13:localhost:8009 ----- </pre> <p>6. Apache, PHP, MySQL, Tomcat 구동 테스트</p> <p>- 서비스 시작</p> <pre> [root@node04 ~]# /usr/local/apache2/bin/apachectl start [root@node04 ~]# /usr/local/tomcat/bin/startup.sh </pre> <p>httpd.conf 의 Document Root 에 정의된 디렉토리에 html, php 파일을 생성 후 브라우저로 아래와 같이 테스트 실시</p> <p>http://192.168.123.64/test.html -&gt; httpd.conf 의 Document Root 의 test.html 파일 로딩 (Apache2 엔진에서 처리)</p> <p>http://192.168.123.64/test.php -&gt; httpd.conf 의 Document Root 의 test.php 파일 로딩 (Apache2 엔진에서 처리)</p> <p>http://192.168.123.64/index.jsp -&gt; /usr/local/tomcat/webapps/ROOT 의 index.jsp 파일</p>	
--	--	---	--

		<p>로딩 (Tomcat5 엔진에서 처리)</p> <p>http://192.168.123.64:8080 -&gt; /usr/local/tomcat/webapps/ROOT 의 index.jsp 파일 로딩 (Tomcat5 엔진에서 처리)</p> <p>7. DNS(Bind) 설치 -&gt; 설정파일 내용은 삭제</p> <ul style="list-style-type: none"> <li>- 패키지 설치 : 리눅스 설치 CD를 이용한 설치</li> </ul> <pre># rpm -Uvh ./cd1/RedHat/RPMS/bind-9.2.1-16.i386.rpm # rpm -Uvh ./cd1/RedHat/RPMS/bind-utils-9.2.1-16.i386.rpm # rpm -Uvh ./cd2/RedHat/RPMS/bind-devel-9.2.1-16.i386.rpm</pre> <ul style="list-style-type: none"> <li>- /etc/rndc.conf 파일 설정</li> <li>- /etc/named.conf 파일 설정 (Primary/Secondary DNS)</li> <li>- /var/named/&lt;domain&gt;.zone 파일 설정</li> <li>- /var/named/&lt;domain&gt;.rev 파일 설정</li> <li>- /etc/resolv.conf 파일 설정</li> <li>- nslookup을 통한 DNS 서버 동작 확인</li> </ul> <p>8. 분산 파일 시스템(Hadoop) 설치</p> <ul style="list-style-type: none"> <li>- 소스파일 다운로드 및 빌딩</li> </ul> <pre># cd /usr/local/ # wget http://ftp.apache-kr.org/lucene/hadoop/hadoop-0.13.1.tar.gz # tar zxvf hadoop-0.13.1.tar.gz # mv hadoop-0.13.1 /home/nutch</pre>	
--	--	---	--

		<p>- hadoop-env.sh 편집</p> <pre># vi ./conf/hadoop-env.sh export JAVA_HOME=/usr/java/jdk_\$version export HADOOP_HOME=/home/nutch export HADOOP_SLAVES=\${HADOOP_HOME}/conf/slaves export HADOOP_LOG_DIR=\${HADOOP_HOME}/logs</pre> <p>- hadoop-site.xml 편집</p> <pre># vi conf/hadoop-site.xml  &lt;configuration&gt;  &lt;property&gt;   &lt;name&gt;fs.default.name&lt;/name&gt;   &lt;value&gt;server1:9000&lt;/value&gt;   &lt;description&gt;     The name of the default file system. Either the literal string     "local" or a host:port for NDFS.   &lt;/description&gt; &lt;/property&gt;  &lt;property&gt;   &lt;name&gt;mapred.job.tracker&lt;/name&gt;   &lt;value&gt;server1:9001&lt;/value&gt;   &lt;description&gt;     The host and port that the MapReduce job tracker runs at. If     "local", then jobs are run in-process as a single map and     reduce task.   &lt;/description&gt; &lt;/property&gt;  &lt;property&gt;   &lt;name&gt;mapred.map.tasks&lt;/name&gt;   &lt;value&gt;2&lt;/value&gt;   &lt;description&gt;</pre>	
--	--	--	--

		<pre> define mapred.map tasks to be number of slave hosts &lt;/description&gt; &lt;/property&gt;  &lt;property&gt;   &lt;name&gt;mapred.reduce.tasks&lt;/name&gt;   &lt;value&gt;2&lt;/value&gt;   &lt;description&gt;     define mapred.reduce tasks to be number of slave     hosts   &lt;/description&gt; &lt;/property&gt;  &lt;property&gt;   &lt;name&gt;dfs.name.dir&lt;/name&gt;   &lt;value&gt;/home/nutch/filesystem/name&lt;/value&gt; &lt;/property&gt;  &lt;property&gt;   &lt;name&gt;dfs.data.dir&lt;/name&gt;   &lt;value&gt;/home/nutch/filesystem/data&lt;/value&gt; &lt;/property&gt;  &lt;property&gt;   &lt;name&gt;mapred.system.dir&lt;/name&gt;    &lt;value&gt;/home/nutch/filesystem/mapreduce/system&lt;/value&gt; &lt;/property&gt;  &lt;property&gt;   &lt;name&gt;mapred.local.dir&lt;/name&gt;   &lt;value&gt;/home/nutch/filesystem/mapreduce/local&lt;/value&gt; &lt;/property&gt;  &lt;property&gt;   &lt;name&gt;dfs.replication&lt;/name&gt;   &lt;value&gt;1&lt;/value&gt; </pre>	
--	--	--	--

		<pre> &lt;/property&gt;  &lt;/configuration&gt;  - 방화벽 설정 master node 설정  # vi /etc/sysconfig/iptables  -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 9000 -j ACCEPT -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 9001 -j ACCEPT -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 50010 -j ACCEPT -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 50030 -j ACCEPT -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 50040 -j ACCEPT -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 50050 -j ACCEPT -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 50060 -j ACCEPT -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 50070 -j ACCEPT  # /etc/init.d/iptables restart  slave node 설정  # vi /etc/sysconfig/iptables  -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 50010 -j ACCEPT -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 50040 -j ACCEPT </pre>	
--	--	--	--

		<pre>-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 50050 -j ACCEPT -A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 50060 -j ACCEPT  # /etc/init.d/iptables restart</pre> <p>- ssh key 생성</p> <pre># scp ~/.ssh/authorized_keys who@server2:~/.ssh/authorized_keys</pre> <p>ssh key를 scp를 이용해 각 slave 노드로 복사 (먼저 복사될 파일 시스템이 각 slave노드에 존재해야 한다.</p> <p>- 설정 파일 편집</p> <p>master node의 slaves 파일 편집. 한줄에 노드이름 하나씩</p> <pre># cd /home/nutch/search/conf # vi slaves server2</pre> <p>hadoop-site.xml 파일 편집. mapred.map.tasks, mapred.reduce.tasks 두 가지 항목은 각 slave 노드에서도 수정해 줘야 함</p> <pre>ser1&gt; vi hadoop-site.xml mapred.map.tasks : 4 mapred.reduce.tasks : 2 dfs.replication : 2</pre> <p>- Namenode 포맷</p> <pre># /home/nutch/bin/hadoop namenode -format</pre> <p>- 데몬 실행</p> <pre># /home/nutch/bin/start-all.sh</pre> <p>마스터와 슬레이브의 모든 데몬을 한꺼번에 실행</p> <p>- master-slave config test</p> <pre># /home/nutch/bin/slaves.sh uptime</pre>	
--	--	--	--