

목차

JBoss EAP 6.x - Standalone Session Clustering	3
설치환경	3
설치전 작업	3
Kernel/TCP Parameter 설정	3
즉시 적용	3
영구적용	3
의존 패키지 설치	4
사용자 및 그룹생성	4
다운로드	4
디렉토리 구조 생성	4
JBoss 구성	5
JBoss Server 설치	5
JBoss Clustering 기본 Node 구성	5
Standalone Node 복사	5
Shell Script 생성	5
설정	10
JBoss Clustering Node 추가	11
Standalone Node 복사	11
실행 스크립트 복사	11
환경설정	12
전체 Node 실행 Shell Script	13
all-run.sh - 전체 실행 기본 스크립트	13
all-start.sh - 전체 시작	13
all-stop.sh - 전체 종료	14
all-restart.sh - 전체 재시작	14
all-log.sh - 콘솔 로그 전체 보기	15
실행권한 부여	15
소유권 변경	15
Clustering Sample Application 생성	15
web.xml	15
jboss-web.xml	16
index.jsp	16
방화벽 설정	17
실행 확인	18
JBoss Cluster Node 실행	18
Clustering 동작 로그 확인	19
Clustering Sample Application 확인	20
Apache HTTPD Server 연동	20
연동 방법 1 - mod_jk	21
연동 방법 2 - mod_cluster 연동	21
오류관련	21
JBAS014612 : java.lang.RuntimeException: java.net.UnknownHostException: standalone01: standalone01	21

JBoss EAP 6.x - Standalone Session Clustering

원문출처 : <http://blog.beany.co.kr/archives/3971> 이상훈 (imfjbh@gmail.com)

설치환경

- CentOS 6.5
- JBoss EAP 6.2.0
- Sun JDK 7

설치전 작업

Kernel/TCP Parameter 설정

OS 에 설치된 기본 설정을 사용할 경우 JBoss 기동시 Socket Buffer 와 관련한 WARN 메시지가 나타날 수 있습니다.

즉시 적용

아래의 명령으로 TCP Parameter 값을 변경합니다. 재부팅시에는 원래의 기본 Parameter 값으로 돌아옵니다.

```
sysctl -w net.core.rmem_default=26214400
sysctl -w net.core.wmem_default=1048576
sysctl -w net.core.rmem_max=26214400
sysctl -w net.core.wmem_max=1048576
```

영구적용

아래의 명령을 실행하여 TCP Parameter 를 설정파일을 편집합니다.

```
net.core.wmem_max = 1048576
net.core.rmem_max = 26214400
net.core.rmem_default = 26214400
net.core.wmem_default = 1048576
```

의존 패키지 설치

아래의 명령을 실행하여 의존 패키지를 설치합니다.

```
yum -y install unzip
```

사용자 및 그룹생성

JBoss 기동시 사용할 사용자 (wasadmin) 계정과 그룹 (wasadmin) 을 생성합니다.

```
groupadd wasadmin  
useradd -M -g wasadmin wasadmin
```

다운로드

JBoss 사이트 (<http://www.jboss.org/jbossas/downloads>) 에 접속하여 JBoss EAP 6.2.0 GA 버전을 다운로드 받습니다.

디렉토리 구조 생성

JBoss 설치 디렉토리 구조는 아래와 같습니다.

```
/app/jboss-as └─ jboss-eap-6.2  
                  └─ nodes ┌─ node01  
                           └─ node02  
                           └─ ...  
/app/webapps/deployments  
/var/log/jboss-as
```

클러스터링을 위한 Standalone 노드를 설치할 폴더를 생성합니다.

```
mkdir -p /app/jboss-as/nodes
```

Web Application 배포 폴더를 생성합니다.

```
mkdir -p /app/webapps/deployments
```

JBoss 구성

JBoss Server 설치

다운로드한 JBoss 설치 압축 파일을 설치위치로 이동한 후 압축을 해제합니다.

```
cp jboss-eap-6.2.0.zip /app/jboss-as  
cd /app/jboss-as  
unzip jboss-eap-6.2.0.zip
```

JBoss 설치 파일을 삭제합니다.

```
rm -f /app/jboss-as/jboss-eap-6.2.0.zip
```

JBoss Clustering 기본 Node 구성

Node 별 로그를 중앙 집중적으로 관리하기 위한 로그 폴더를 생성합니다.

```
mkdir -p /var/log/jboss-as
```

Standalone Node 복사

압축을 해제한 JBoss 설치 디렉토리로 이동한 후 standalone 폴더를 복사한 후 폴더명을 변경합니다.

```
cp -R /app/jboss-as/jboss-eap-6.2/standalone /app/jboss-as/nodes  
mv /app/jboss-as/nodes/standalone /app/jboss-as/nodes/node01
```

Shell Script 생성

JBoss Clustering Node 들에 대한 실행을 위한 스크립트 및 설정 파일을 생성합니다.

jboss-env.conf - 환경 설정 파일

아래의 명령을 실행하여 JBoss Node 별로 속성을 설정할 수 있는 jboss-env.conf 파일을 생성합니다.

```
vi /app/jboss-as/nodes/node01/jboss-env.conf
```

아래의 내용을 입력합니다.

```
JAVA_HOME=/opt/jdk1.7.0_55

JBOSS_NODE_NAME=node01
JBOSS_PORT_OFFSET=100

JBOSS_USER=wasadmin
JBOSS_HOME=/app/jboss-as/jboss-eap-6.2
JBOSS_NODE_BASE_DIR=/app/jboss-as/nodes

JBOSS_PIDFILE=/var/run/jboss-as/jboss-as-${JBOSS_NODE_NAME}.pid
JBOSS_CONFIG=standalone-ha.xml
JBOSS_BASE_LOG_DIR=/var/log/jboss-as
JBOSS_CONSOLE_LOG=${JBOSS_BASE_LOG_DIR}/${JBOSS_NODE_NAME}-console.log
JBOSS_GC_LOG=${JBOSS_BASE_LOG_DIR}/${JBOSS_NODE_NAME}-gc.log

JBOSS_BIND_ADDR=0.0.0.0
JBOSS_MANAGEMENT_ADDR=127.0.0.1
JBOSS_CONTROLLER_IP=${JBOSS_MANAGEMENT_ADDR}
JBOSS_MULTICAST_ADDR=230.0.0.4
JBOSS_JMS_MULTICAST_ADDR=231.7.0.1
JBOSS_MODCLUSTER_MULTICAST_ADDR=224.0.1.105

LAUNCH_JBOSS_IN_BACKGROUND=true

STARTUP_WAIT=30
SHUTDOWN_WAIT=30
```

jboss-env.sh - 환경 설정

공통환경 쉘 스크립트 파일을 생성합니다.

```
vi /app/jboss-as/nodes/node01/jboss-env.sh
```

아래의 내용으로 시작 쉘 스크립트를 입력한 후 저장합니다.

```
#!/bin/sh

DATE=`date +%Y%m%d%H%M%S`
BASE_DIR=$(dirname $0)

export JBOSS_CONF=$BASE_DIR/jboss-env.conf
```

```
[ -r "$JBASS_CONF" ] && . "${JBASS_CONF}"

let JBASS_CONTROLLER_PORT=9999+$JBASS_PORT_OFFSET
export JBASS_CONTROLLER_PORT

##### JBoss System module and User module directory #####
export JBASS_MODULEPATH=$JBASS_HOME/modules:$JBASS_HOME/modules/ext

# JVM Options : Server
export JAVA_OPTS="-server $JAVA_OPTS"

# JVM Options : Memory
export JAVA_OPTS=" $JAVA_OPTS -Xms1024m -Xmx1024m -XX:MaxPermSize=256m"
export JAVA_OPTS=" $JAVA_OPTS -XX:+PrintGCTimeStamps "
export JAVA_OPTS=" $JAVA_OPTS -XX:+PrintGCDetails "
export JAVA_OPTS=" $JAVA_OPTS -Xloggc:$JBASS_GC_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:-HeapDumpOnOutOfMemoryError "

# 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.jboss.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 -Djboss.modules.system.pkgs=org.jboss.byteman"
export JAVA_OPTS=" $JAVA_OPTS -Djava.awt.headless=true"
export JAVA_OPTS=" $JAVA_OPTS -
Djboss.server.base.dir=$JBASS_NODE_BASE_DIR/$JBASS_NODE_NAME"
export JAVA_OPTS=" $JAVA_OPTS -Djboss.server.log.dir=$JBASS_BASE_LOG_DIR"
export JAVA_OPTS=" $JAVA_OPTS -Djboss.socket.binding.port-
offset=$JBASS_PORT_OFFSET"
export JAVA_OPTS=" $JAVA_OPTS -Djboss.node.name=$JBASS_NODE_NAME"

export JAVA_OPTS=" $JAVA_OPTS -
Djboss.bind.address.management=$JBASS_MANAGEMENT_ADDR"
export JAVA_OPTS=" $JAVA_OPTS -Djboss.bind.address=$JBASS_BIND_ADDR"
#export JAVA_OPTS=" $JAVA_OPTS -Djboss.bind_addr=$JBASS_MULTICAST_ADDR"
#export JAVA_OPTS=" $JAVA_OPTS -Djboss.default.jgroups.stack=tcp"
export JAVA_OPTS=" $JAVA_OPTS -
Djboss.default.multicast.address=$JBASS_MULTICAST_ADDR"
export JAVA_OPTS=" $JAVA_OPTS -
Djboss.messaging.group.address=$JBASS_JMS_MULTICAST_ADDR"
export JAVA_OPTS=" $JAVA_OPTS -
Djboss.modcluster.multicast.address=$JBASS_MODCLUSTER_MULTICAST_ADDR"
export JAVA_OPTS=" $JAVA_OPTS -Dorg.jboss.as.logging.per-deployment=false"
export JAVA_OPTS=" $JAVA_OPTS -Dserver.mode=local"
```

```

export JAVA_OPTS=" $JAVA_OPTS -
Djboss.server.config.user.dir=$JBOSS_DOMAIN_BASE_DIR/$JBOSS_DOMAIN_NAME/conf
figuration"
export JAVA_OPTS=" $JAVA_OPTS -
Djboss.domain.config.user.dir=$JBOSS_DOMAIN_BASE_DIR/$JBOSS_DOMAIN_NAME/conf
figuration"

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH

echo "=====
echo "JAVA_HOME=$JAVA_HOME"
echo "JBOSS_HOME=$JBOSS_HOME"
echo "JBOSS_NODE_NAME=$JBOSS_NODE_NAME"
echo "JBOSS_NODE_BASE_DIR=$JBOSS_NODE_BASE_DIR"
echo "JBOSS_CONFIG=$JBOSS_CONFIG"

echo "JBOSS_BIND_ADDR=$JBOSS_BIND_ADDR"
echo "JBOSS_PORT_OFFSET=$JBOSS_PORT_OFFSET"
echo "JBOSS_MULTICAST_ADDR=$JBOSS_MULTICAST_ADDR"
echo "JBOSS_CONTROLLER=$JBOSS_CONTROLLER_IP:$JBOSS_CONTROLLER_PORT"

echo "JBOSS_PIDFILE=$JBOSS_PIDFILE"
echo "JBOSS_CONSOLE_LOG=$JBOSS_CONSOLE_LOG"
echo "JBOSS_GC_LOG=$JBOSS_GC_LOG"
echo "=====

```

jboss-run.sh - 실행 스크립트 (시작&종료)

JBoss Node 실행 쉘 스크립트 파일을 생성합니다.

```
vi /app/jboss-as/nodes/node01/jboss-run.sh
```

아래의 내용으로 실행 쉘 스크립트를 입력한 후 저장합니다.

```

#!/bin/sh

BASE_DIR=$(dirname $0)

. $BASE_DIR/jboss-env.sh

if [ "x$1" == "xstart" ]; then
    if [ -e $JBOSS_CONSOLE_LOG ]; then
        mv $JBOSS_CONSOLE_LOG $JBOSS_CONSOLE_LOG.$DATE
    fi

    if [ -e $JBOSS_GC_LOG ]; then

```

```
mv $JBOSS_GC_LOG $JBOSS_GC_LOG.$DATE
fi
fi

$JBOSS_HOME/bin/init.d/jboss-as-standalone.sh $1

#tail -f $JBOSS_CONSOLE_LOG
```

jboss-jconsole.sh - JConsole

JConsole 셸 스크립트 파일을 생성합니다.

```
vi /app/jboss-as/nodes/node01/jboss-jconsole.sh
```

아래의 내용을 JConsole 셸 스크립트를 입력한 후 저장합니다.

```
#!/bin/sh

BASE_DIR=$(dirname $0)

. $BASE_DIR/jboss-env.sh

echo "=====
echo " JMX URL : service:jmx:remoting-
jmx://$JBOSS_MANAGEMENT_ADDR:$JBOSS_CONTROLLER_PORT"
echo "=====

$JBOSS_HOME/bin/jconsole.sh
```

jboss-cli.sh - CLI

CLI 셸 스크립트 파일을 생성합니다.

```
vi /app/jboss-as/nodes/node01/jboss-cli.sh
```

아래의 내용을 CLI 셸 스크립트를 입력한 후 저장합니다.

```
#!/bin/sh

BASE_DIR=$(dirname $0)

. $BASE_DIR/jboss-env.sh
```

```
export JAVA_OPTS=" -Djava.awt.headless=false $JAVA_OPTS"

$JBOSS_HOME/bin/jboss-cli.sh --
controller=$JBOSS_CONTROLLER_IP:$JBOSS_CONTROLLER_PORT --connect @$
```

실행권한 부여

셸 스크립트에 실행 권한을 부여합니다.

```
chmod 755 /app/jboss-as/nodes/node01/*.sh
```

설정

Standalone 기반의 Clustering 구성을 위하여 standalone-ha.xml 파일을 수정합니다.

```
vi /app/jboss-as/nodes/node01/configuration/standalone-ha.xml
```

아래의 항목을 다음과 같이 수정또는 추가합니다.

- deployment-scanner → path : Web Application 배포 디렉토리를 지정합니다.
- deployment-scanner → scan-interval : path (Web Application 배포 디렉토리) 의 파일들에 대한 변경 감지 시간을 설정합니다. (1/1000 초) 0 으로 설정시 변경 감지를 하지 않습니다.
- deployment-scanner → auto-deploy-exploded : true 로 설정했을 경우 war 파일이 아닌 디렉토리로 설정할 수 있습니다.
- instance-id : jvmRoute 기능을 하기 위한 설정이며, 설정값 \${jboss.node.name} 는 외부 설정값으로 대체됩니다.
- enable-welcome-root : ROOT Context 를 사용할 경우 false 로 설정합니다. true 로 설정이 되어있을 경우 Root Context 는 JBoss 에 기본으로 내장되어 있는 Content 를 표시하게 됩니다.

```
<?xml version='1.0' encoding='UTF-8'?>

<server xmlns="urn:jboss:domain:1.5">

    ~~~~~

    <profile>

        ~~~~~

        <subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">
            <!--
            <deployment-scanner path="deployments" relative-
to="jboss.server.base.dir" scan-interval="5000"/>
            -->
            <deployment-scanner path="/app/webapps/deployments" scan-
```

```

interval="0" auto-deploy-exploded="true" />

    </subsystem>

    <subsystem xmlns="urn:jboss:domain:web:1.5" default-virtual-
server="default-host" instance-id="${jboss.node.name}" native="false">
        <connector name="http" protocol="HTTP/1.1" scheme="http" socket-
binding="http"/>
        <connector name="ajp" protocol="AJP/1.3" scheme="http" socket-
binding="ajp"/>
        <virtual-server name="default-host" enable-welcome-root="false">
            <alias name="localhost"/>
            <alias name="example.com"/>
        </virtual-server>
    </subsystem>

    ~~~~~

</profile>

~~~~~

</server>

```

JBoss Clustering Node 추가

하나의 물리적인 서버에 Clustering Node 추가 작업에 대한 설명입니다.

Standalone Node 복사

아래의 명령을 실행하여 기존의 JBoss Home 폴더의 standalone 폴더를 복사한 후 해당 Node 의 이름을 변경합니다.

현재는 기존의 /app/jboss-as/nodes/node01 폴더가 존재하기 때문에 /app/jboss-as/nodes/node02 로 설정합니다.

```

cp -R /app/jboss-as/jboss-eap-6.2/standalone /app/jboss-as/nodes
mv /app/jboss-as/nodes/standalone /app/jboss-as/nodes/node02

```

실행 스크립트 복사

/app/jboss-as/nodes/node01 에서 생성한 실행 스크립트를 신규로 생성하는 Node 로 복사합니다.

```

cp /app/jboss-as/nodes/node01/*.sh /app/jboss-as/nodes/node01/*.conf
/app/jboss-as/nodes/node02

```

환경설정

아래의 명령을 실행하여 jboss-env.conf 파일을 편집합니다.

```
vi /app/jboss-as/nodes/node02/jboss-env.conf
```

아래의 항목을 현재 설정에 맞도록 수정한 후 저장합니다.

```
JAVA_HOME=/opt/jdk1.7.0_55

JBOSS_NODE_NAME=node02
JBOSS_PORT_OFFSET=200

JBOSS_USER=wasadmin
JBOSS_HOME=/app/jboss-as/jboss-eap-6.2
JBOSS_NODE_BASE_DIR=/app/jboss-as/nodes

JBOSS_PIDFILE=/var/run/jboss-as/jboss-as-${JBOSS_NODE_NAME}.pid
JBOSS_CONFIG=standalone-ha.xml
JBOSS_BASE_LOG_DIR=/var/log/jboss-as
JBOSS_CONSOLE_LOG=${JBOSS_BASE_LOG_DIR}/${JBOSS_NODE_NAME}-console.log
JBOSS_GC_LOG=${JBOSS_BASE_LOG_DIR}/${JBOSS_NODE_NAME}-gc.log

JBOSS_BIND_ADDR=0.0.0.0
JBOSS_MANAGEMENT_ADDR=127.0.0.1
JBOSS_CONTROLLER_IP=${JBOSS_MANAGEMENT_ADDR}
JBOSS_MULTICAST_ADDR=230.0.0.4
JBOSS_JMS_MULTICAST_ADDR=231.7.0.1
JBOSS_MODCLUSTER_MULTICAST_ADDR=224.0.1.105

LAUNCH_JBOSS_IN_BACKGROUND=true

STARTUP_WAIT=30
SHUTDOWN_WAIT=30
```

기본 Node 에서 standalone-ha.xml 파일을 복사합니다.

```
cp /app/jboss-as/nodes/node01/configuration/standalone-ha.xml /app/jboss-as/nodes/node02/configuration/
```

전체 Node 실행 Shell Script

JBoss Node 기본 디렉토리로 이동합니다.

```
cd /app/jboss-as/nodes
```

all-run.sh - 전체 실행 기본 스크립트

아래의 명령을 실행하여 전체 실행시 공통으로 사용할 run-all.sh 파일을 생성합니다.

```
vi /app/jboss-as/nodes/all-run.sh
```

아래와 같이 Shell Script 작성 후 저장합니다.

```
#!/bin/sh

if [ "x$1" == "x" ]; then
    exit 0
fi

BAKIFS=$IFS;
IFS=' ';

BASE_NODE_DIR=$(dirname $0)
WAIT=3

nodes="node01,node02";
nodeArray=( $nodes );

for ((i=0; i<${#nodeArray[@]}; ++i));
do
    echo ">>> Run Node : ${nodeArray[$i]}";

    runScript="${BASE_NODE_DIR}/${nodeArray[$i]}/jboss-run.sh $1";
    eval $runScript

    sleep $WAIT
done

IFS=$BAKIFS;
```

all-start.sh - 전체 시작

아래의 명령을 실행하여 전체 시작 Shell Script 파일을 생성합니다.

```
vi /app/jboss-as/nodes/all-start.sh
```

아래와 같이 Shell Script 작성 후 저장합니다.

```
#!/bin/sh

BASE_NODE_DIR=$(dirname $0)

$BASE_NODE_DIR/all-run.sh start
```

all-stop.sh - 전체 종료

아래의 명령을 실행하여 전체 종료 Shell Script 파일을 생성합니다.

```
vi /app/jboss-as/nodes/all-stop.sh
```

아래와 같이 Shell Script 작성 후 저장합니다.

```
#!/bin/sh

BASE_NODE_DIR=$(dirname $0)

$BASE_NODE_DIR/all-run.sh stop
```

all-restart.sh - 전체 재시작

아래의 명령을 실행하여 전체 재시작 Shell Script 파일을 생성합니다.

```
vi /app/jboss-as/nodes/all-restart.sh
```

아래와 같이 Shell Script 작성 후 저장합니다.

```
#!/bin/sh

BASE_NODE_DIR=$(dirname $0)

$BASE_NODE_DIR/all-run.sh restart
```

all-log.sh - 콘솔 로그 전체 보기

아래의 명령을 실행하여 콘솔로그 전체 보기 Shell Script 파일을 생성합니다.

```
vi /app/jboss-as/nodes/all-log.sh
```

아래와 같이 Shell Script 작성 후 저장합니다.

```
#!/bin/sh  
  
tail -n 50 -f /var/log/jboss-as/node*-console.log
```

실행권한 부여

셸 스크립트에 실행권한을 부여합니다.

```
chmod 755 /app/jboss-as/nodes/*.sh
```

소유권 변경

아래의 명령을 실행하여 JBoss 설치 관련 디렉토리의 소유권을 변경합니다.

```
chown -R wasadmin:wasadmin /app/jboss-as /app/webapps /var/log/jboss-as
```

Clustering Sample Application 생성

Sample Application 배포 폴더를 생성합니다.

```
mkdir -p /app/webapps/deployments/clustering.war/WEB-INF
```

web.xml

web.xml 파일을 생성합니다.

```
vi /app/webapps/deployments/clustering.war/WEB-INF/web.xml
```

web.xml 의 설정을 다음과 같이 입력한 후 저장합니다.

- <distributable/> : Web Application 을 Session Clustering 에 참여시킵니다.

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
version="3.0">

    <display-name>JBoss Sample Web App</display-name>
    <welcome-file-list>
        <welcome-file>/index.jsp</welcome-file>
    </welcome-file-list>

    <distributable/>

</web-app>
```

jboss-web.xml

jboss-web.xml 파일을 생성합니다.

```
vi /app/webapps/deployments/clustering.war/WEB-INF/jboss-web.xml
```

jboss-web.xml 을 아래와 같이 설정한 후 저장합니다.

- <context-root> : Context Path 을 지정하는 옵션입니다.

```
<?xml version="1.0" encoding="UTF-8"?>

<jboss-web>
    <context-root>clustering</context-root>
</jboss-web>
```

index.jsp

index.jsp 파일을 생성합니다.

```
vi /app/webapps/deployments/clustering.war/index.jsp
```

아래의 내용을 입력한 후 저장합니다.

```
<%@ page session="true" %>

<html>
<head>
    <title>Session Tracking Test</title>
</head>
<body>
<h1>Session Tracking Test</h1>
Session tracking with JSP is easy
<p>
<%
// Get the session data value
Integer ival = (Integer) session.getValue ("counter");

if (ival == null) ival = new Integer (1);
else ival = new Integer (ival.intValue() + 1);
session.putValue ("counter", ival);
%>
<h4>Server Name : <%=session.getId().substring(session.getId().indexOf(".")
+ 1) %></h4>
You have hit this page <%= ival %> times.<br>
<%
out.println("Your Session ID is " + session.getId() + "<br>");
System.out.println("session=" + session.getId() + ", counter=" + ival);
%>
</body>
</html>
```

방화벽 설정

아래의 명령을 실행하여 방화벽 정책 파일을 편집합니다.

```
vi /etc/sysconfig/iptables
```

방화벽 정책은 JBoss 가 구성될 네트워크의 환경에 맞도록 수정한 저장합니다.

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8180 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8280 -j ACCEPT

-A INPUT -m iprange --src-range 192.168.122.21-192.168.122.23 -p tcp -m tcp
--dport 7600 -j ACCEPT
-A INPUT -m iprange --src-range 192.168.122.21-192.168.122.23 -p tcp -m tcp
--dport 54200 -j ACCEPT
-A INPUT -m iprange --src-range 192.168.122.21-192.168.122.23 -p tcp -m tcp
--dport 57600 -j ACCEPT
```

```
-A INPUT -s 230.0.0.4/32 -p udp -m udp --dport 45688 -j ACCEPT
-A INPUT -s 230.0.0.4/32 -p udp -m udp --dport 45700 -j ACCEPT
-A INPUT -s 224.0.1.105/32 -p udp -m udp --dport 23364 -j ACCEPT
-A INPUT -m iprange --src-range 192.168.122.21-192.168.122.23 -p udp -j
ACCEPT
#-A INPUT -s 192.168.122.0/24 -p udp -j ACCEPT
```

아래의 명령을 실행하여 방화벽 정책을 적용합니다.

```
service iptables restart
```

실행 확인

JBoss Cluster Node 실행

아래의 명령을 실행하여 JBoss Cluster Node 를 모두 실행합니다.

```
/app/jboss-as/nodes/all-start.sh
```

정상적으로 실행되면 아래와 같이 메시지가 출력이 됩니다.

```
>>> Run Node : node01
=====
JAVA_HOME=/opt/jdk1.7.0_55
JBOSS_HOME=/app/jboss-as/jboss-eap-6.2
JBOSS_NODE_NAME=node01
JBOSS_NODE_BASE_DIR=/app/jboss-as/nodes
JBOSS_CONFIG=standalone-ha.xml
JBOSS_BIND_ADDR=0.0.0.0
JBOSS_PORT_OFFSET=100
JBOSS_MULTICAST_ADDR=230.0.0.4
JBOSS_CONTROLLER=127.0.0.1:10099
JBOSS_PIDFILE=/var/run/jboss-as/jboss-as-node01.pid
JBOSS_CONSOLE_LOG=/var/log/jboss-as/node01-console.log
JBOSS_GC_LOG=/var/log/jboss-as/node01-gc.log
=====
Starting jboss-as: [ OK ]
>>> Run Node : node02
=====
JAVA_HOME=/opt/jdk1.7.0_55
JBOSS_HOME=/app/jboss-as/jboss-eap-6.2
JBOSS_NODE_NAME=node02
JBOSS_NODE_BASE_DIR=/app/jboss-as/nodes
```

```
JBoss_CONFIG=standalone-ha.xml
JBoss_BIND_ADDR=0.0.0.0
JBoss_PORT_OFFSET=200
JBoss_MULTICAST_ADDR=230.0.0.4
JBoss_CONTROLLER=127.0.0.1:10199
JBoss_PIDFILE=/var/run/jboss-as/jboss-as-node02.pid
JBoss_CONSOLE_LOG=/var/log/jboss-as/node02-console.log
JBoss_GC_LOG=/var/log/jboss-as/node02-gc.log
=====
Starting jboss-as: [ OK ]
```

Clustering 동작 로그 확인

아래의 명령을 실행하여 “node01” 서버의 로그를 출력합니다.

```
tail -n 100 /var/log/jboss-as/node01-console.log
```

아래와 같이 Cluster member 개수와 하단에 node02 가 등록되었다는 메시지를 확인 하실수 있습니다. 현재 Cluster member 가 1 개로 표시된 것은 node01 이 처음에 실행이 되어서 현재 등록 되어있는 Cluster member 가 존재하지 않기 때문입니다.

```
17:16:06,477 INFO [org.infinispan.factories.GlobalComponentRegistry]
(ServerService Thread Pool -- 53) ISPN000128: Infinispan version: Infinispan
'Delirium' 5.2.7.Final
17:16:06,507 INFO [org.jboss.as.clustering] (MSC service thread 1-3)
JBAS010238: Number of cluster members: 1
17:16:06,525 INFO [org.infinispan.factories.TransactionManagerFactory]
(ServerService Thread Pool -- 54) ISPN000161: Using a batchMode transaction
manager

~~~~~
~~~~~

17:16:06,972 INFO [org.jboss.as] (Controller Boot Thread) JBAS015961: Http
management interface listening on http://127.0.0.1:10090/management
17:16:06,973 INFO [org.jboss.as] (Controller Boot Thread) JBAS015951: Admin
console listening on http://127.0.0.1:10090
17:16:06,973 INFO [org.jboss.as] (Controller Boot Thread) JBAS015874: JBoss
EAP 6.2.0.GA (AS 7.3.0.Final-redhat-14) started in 7154ms - Started 218 of
348 services (129 services are passive or on-demand)
17:16:15,553 INFO [org.jboss.as.clustering] (Incoming-1,shared=udp)
JBAS010225: New cluster view for partition web (id: 1, delta: 1, merge:
false) : [node01/web, node02/web]
17:16:15,553 INFO
[org.infinispan.remoting.transport.jgroups.JGroupsTransport]
(Incoming-1,shared=udp) ISPN000094: Received new cluster view:
```

```
[node01/web|1] [node01/web, node02/web]
```

아래의 명령을 실행하여 node02 서버의 로그를 출력합니다.

```
tail -n 100 /var/log/jboss-as/node02-console.log
```

node01 이 실행되고 난 후 실행된 node02 에서는 아래와 같이 Cluster member 로 등록되었다는 메시지가 나타나게 됩니다.

```
17:16:15,582 INFO
[org.infinispan.remoting.transport.jgroups.JGroupsTransport] (ServerService
Thread Pool -- 57) ISPN000094: Received new cluster view: [node01/web|1]
[node01/web, node02/web]
17:16:15,686 INFO
[org.infinispan.remoting.transport.jgroups.JGroupsTransport] (ServerService
Thread Pool -- 57) ISPN000079: Cache local address is node02/web, physical
addresses are [0.0.0.0:55400]
17:16:15,694 INFO [org.infinispan.factories.GlobalComponentRegistry]
(ServerService Thread Pool -- 57) ISPN000128: Infinispan version: Infinispan
'Delirium' 5.2.7.Final
17:16:15,709 INFO [org.jboss.as.clustering] (MSC service thread 1-2)
JBAS010238: Number of cluster members: 2
17:16:15,752 INFO [org.infinispan.factories.TransactionManagerFactory]
(ServerService Thread Pool -- 53) ISPN000161: Using a batchMode transaction
manager
17:16:15,752 INFO [org.infinispan.factories.TransactionManagerFactory]
(ServerService Thread Pool -- 54) ISPN000161: Using a batchMode transaction
manager
17:16:15,756 INFO [org.infinispan.factories.TransactionManagerFactory]
(ServerService Thread Pool -- 55) ISPN000161: Using a batchMode transaction
manager
```

Clustering Sample Application 확인

브라우저를 통하여 각 노드를 직접 연결하여 확인을 합니다. “Node01” 과 “Node02” 를 번갈아 가면서 새로 고침을 했을 경우 Page Hit 수가 연결되어 증가하는지 확인 합니다.

Apache HTTPD Server 연동

Apache HTTPD Server 의 연동은 mod_jk, mod_cluster, mod_proxy 등을 이용하여 할 수 있습니다. 다음의 연동방법 중 한 개를 선택하셔서 구성하시면 됩니다.

연동 방법 1 - mod_jk

JBoss EAP 6.x - Apache HTTPD 2.2.x + mod_jk 1.2.x Session Clustering 구성 문서를 참고합니다.

연동 방법 2 - mod_cluster 연동

JBoss EAP 6.x - Apache HTTPD 2.2.x + mod_cluster 1.2.x Session Clustering 구성 문서를 참고합니다.

오류관련

JBAS014612 : java.lang.RuntimeException: java.net.UnknownTion: standalone01: standalone01

오류 상세 메시지

JBoss Node 시작시 console 로그에 아래와 같은 오류 메시지 출력됩니다.

```
21:28:41,857 ERROR [org.jboss.as.controller.management-operation]
(ServerService Thread Pool -- 50) JBAS014612: Operation ("add") failed -
address: ([("subsystem" => "webservices"))]: java.lang.RuntimeException:
java.net.UnknownTion: standalone01: standalone01
    at
org.jboss.as.webservices.dmr.WSSubsystemAdd.createServerConfig(WSSubsystemAd
d.java:103)
    at
org.jboss.as.webservices.dmr.WSSubsystemAdd.performBoottime(WSSubsystemAdd.j
ava:88)
    at
org.jboss.as.controller.AbstractBoottimeAddStepHandler.performRuntime(Abstra
ctBoottimeAddStepHandler.java:57) [jboss-as-controller-7.3.0.Final-
redhat-14.jar:7.3.0.Final-redhat-14]
    at
org.jboss.as.controller.AbstractAddStepHandler$1.execute(AbstractAddStepHand
ler.java:76) [jboss-as-controller-7.3.0.Final-redhat-14.jar:7.3.0.Final-
redhat-14]
    at
org.jboss.as.controller.AbstractOperationContext.executeStep(AbstractOperati
onContext.java:607) [jboss-as-controller-7.3.0.Final-
redhat-14.jar:7.3.0.Final-redhat-14]
    at
org.jboss.as.controller.AbstractOperationContext.doCompleteStep(AbstractOper
ationContext.java:485) [jboss-as-controller-7.3.0.Final-
redhat-14.jar:7.3.0.Final-redhat-14]
    at
org.jboss.as.controller.AbstractOperationContext.completeStepInternal(Abstra
```

```

ctOperationContext.java:282) [jboss-as-controller-7.3.0.Final-
redhat-14.jar:7.3.0.Final-redhat-14]
    at
org.jboss.as.controller.AbstractOperationContext.executeOperation(AbstractOp
erationContext.java:277) [jboss-as-controller-7.3.0.Final-
redhat-14.jar:7.3.0.Final-redhat-14]
    at
org.jboss.as.controller.ParallelBootOperationStepHandler$ParallelBootTask.ru
n(ParallelBootOperationStepHandler.java:343) [jboss-as-
controller-7.3.0.Final-redhat-14.jar:7.3.0.Final-redhat-14]
    at
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:11
45) [rt.jar:1.7.0_55]
    at
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:6
15) [rt.jar:1.7.0_55]
        at java.lang.Thread.run(Thread.java:745) [rt.jar:1.7.0_55]
        at org.jboss.threads.JBossThread.run(JBossThread.java:122) [jboss-
threads-2.1.1.Final-redhat-1.jar:2.1.1.Final-redhat-1]
Caused by: java.net.UnknownHostException: standalone01: standalone01
    at java.net.InetAddress.getLocalHost(InetAddress.java:1473)
[rt.jar:1.7.0_55]
    at
org.jboss.ws.common.management.AbstractServerConfig.setWebServiceHost(Abstra
ctServerConfig.java:111)
    at
org.jboss.as.webservices.dmr.WSSubsystemAdd.createServerConfig(WSSubsystemAd
d.java:101)
    ... 12 more
Caused by: java.net.UnknownHostException: standalone01
    at java.net.Inet4AddressImpl.lookupAllHostAddr(Native Method)
[rt.jar:1.7.0_55]
    at java.net.InetAddress$1.lookupAllHostAddr(InetAddress.java:901)
[rt.jar:1.7.0_55]
    at
java.net.InetAddress.getAddressesFromNameService(InetAddress.java:1293)
[rt.jar:1.7.0_55]
        at java.net.InetAddress.getLocalHost(InetAddress.java:1469)
[rt.jar:1.7.0_55]
        ... 14 more

```

원인

JBoss 시작시 오류에 찍힌 호스트명 (standalone01) 을 인식하지 못하여 발생하는 문제입니다. hosts 파일에 해당 호스트명을 등록해야 합니다.

해결

아래의 명령을 실행하여 “/etc/hosts” 파일을 편집 합니다.

```
vi /etc/hosts
```

아래와 같이 해당 호스트명을 등록한 후 저장합니다.

```
127.0.0.1    localhost localhost.localdomain localhost4  
localhost4.localhost4  
192.168.0.11 standalone01 # 해당 서버의 아이피
```

JBoss Server 을 재실행합니다.

From:

<https://atl.kr/dokuwiki/> - AllThatLinux!

Permanent link:

https://atl.kr/dokuwiki/doku.php/jboss_eap_6.x_standalone_session_clustering

Last update: **2016/05/20 05:42**

