

[별첨2]

Cassandra 테스트 환경 구축 가이드

2012. 7.

본 설치 가이드는 공개SW 역량프라자에서 분산 데이터베이스 (NoSQL) 기반 기술 중 Cassandra에 대한 테스트 결과 보고서 외에 테스트 환경에 대한 이해를 돕고자 작성되었습니다.

모든 테스트 환경 구성에 대한 내용을 포함되어 있지 않으며, 주의가 필요하거나 참고해야 할 내용을 기반으로 작성되었습니다.

1. 설치환경

☐ Cassandra 환경

모듈	Version
Cassandra	1.1.1

☐ Stack 환경

구성	OS	JDK
A Stack	CentOS 6.2 (64bit)	1.7.0_04
B Stack	Windows Server 2008 R2 Standard Edition (64bit)	1.7.0_04

☐ HW 환경

제조사	모델명	CPU	MEM	Disk	NIC
IBM	X3850M2	2.40GHz (Six Core) x 4, 12MB L3 90w	8GB	146G x 4	Gigabit
HP	DL360G6	2.40 GHz (Quad-Core) x 2	8GB	146G x 5	Gigabit

2. JDK 설치

□ Linux

리눅스 서버를 설치(설치 옵션에 따라 다름)할 경우 기본적으로 리눅스 버전에 따라 jdk버전을 다르지만, OpenJDK가 설치되며, 해당 jdk버전을 사용해도 문제가 없을 경우 아래의 내용과 같이 별도로 jdk를 설치하지 않아도 되며, 본 테스트에서는 오라클에서 배포하는 jdk를 사용하였음

1) jdk 다운로드

자바 다운로드 사이트에서 리눅스 최신 버전(jdk-7u4-linux-x64.gz)을 설치 서버에 다운로드한다.

```
#cd /opt
#wget download_url/jdk-7u4-linux-x64.gz
```

2) 다운로드 받은 파일을 /opt 디렉터리에 압축을 해제한다.

```
#tar xvf jdk-7u4-linux-x64.gz
```

3) java 경로 설정

```
#vi /etc/profile.d/java.sh
//아래의 내용을 입력 후 저장
export JAVA_HOME=/opt/jdk1.7.0_04/jre
export PATH=$JAVA_HOME/bin:$PATH
```

4) java 경로 등록

```
#source /etc/profile.d/java.sh
#echo $JAVA_HOME //경로가 정상적으로 설정되었는지 확인
```

5) 시스템에서 사용할 java 설정

```
#/usr/sbin/alternatives --install /usr/bin/java java /opt/jdk1.7.0_04/jre/bin/java 2
```

```
#/usr/sbin/alternatives --config java
```

-> 설치된 java 환경이 보여지며, 그 중 현재 설치한 경로의 java정보의 선택번호를 입력하여 시스템에서 사용할 java를 선택한다.

6) java 설정 확인

```
#/usr/sbin/alternatives --display java
```

```
#java -version
```

☐ Windows

윈도우 환경에서 jdk 설치 정보는 웹 검색시 많은 자료가 있으므로 별도의 설명을 하지 않음

3. Cassandra 설치

☐ 공통(Linux, Windows)

※ 기본적으로 JDK는 정상적으로 설치되었다고 가정함

1) Cassandra 다운로드

Cassandra 홈페이지(<http://cassandra.apache.org>)에서 최신 안정화 버전(`apache-cassandra-1.1.1-bin.tar.gz`)을 다운로드한다.

2) 압축 해제

```
#tar xvf apache-cassandra-1.1.1-bin.tar.gz
```

3) 기본 디렉터리 생성 및 권한 설정

```
#mkdir -p /var/log/cassandra
#chown -R 'whoami' /var/log/cassandra //리눅스만 해당
#mkdir -p /var/lib/cassandra
#chown -R 'whoami' /var/lib/cassandra //리눅스만 해당
```

4) Cassandra 실행

```
# bin/cassandra -f
```

- 위의 명령어를 실행하여 정상적으로 cassandra 서버가 구동되는지 확인

5) Cassandra 클라이언트 접속

```
# bin/cassandra-cli -h
```

4. Cassandra 설정

- ☐ Linux 환경에서 cassandra 서버 구동시 Stack Size 오류

- o 오류현상

```
The stack size specified is too small, Specify at least 160k
Error: Could not create the Java Virtual Machine.
Error: A fatal exception has occurred. Program will exit.
```

- o 해결방안

- /conf/cassandra-env.sh 파일에서 아래의 내용으로 수정

```
if [ "uname" = "Linux" ] ; then
    # reduce the per-thread stack size to minimize the impact of Thrift
    # thread-per-client. (Best practice is for client connections to
    # be pooled anyway.) Only do so on Linux where it is known to be
    # supported.
    #JVM_OPTS="$JVM_OPTS -Xss128k" //수정 전 정보
    JVM_OPTS="$JVM_OPTS -Xss160k" //수정 후 정보
fi
```

□ Cluster 설정하기

본 테스트 환경에서는 서버 2대를 이용하여 Cluster를 구성하였음

o A(node1) 서버 설정

- IP : 192.168.0.1
- seed 서버로 설정

1) token 생성

- 아래의 URL을 참조하여 token 생성

http://www.datastax.com/docs/1.1/initialize/token_generation#token-gen-cassandra

2) edit conf/cassandra.yaml

- initial_token: 0
- seeds : 192.168.0.1 (A서버 IP 설정)
- listen_address: 192.168.0.1
- rpc_address: 0.0.0.0

o B(node2) 서버 설정

- IP : 192.168.0.2

1) edit conf/cassandra.yaml

- initial_token: 85070591730234615865843651857942052864

(tokentool을 이용하여 생성 된 값)

- seeds: "192.168.0.1" <- A서버 IP
- listen_address: 192.168.0.2
- rpc_address: 192.168.0.2

o Cluster 동작 확인

1) A 서버 실행

bin/cassandra -f

2) B 서버 실행

bin/cassandra -f

3) 클러스터 확인

A노드 실행화면에 아래와 같은 메시지 출력

INFO 17:30:39,765 Node /192.168.0.2 is now part of the cluster

INFO 17:30:39,766 InetAddress /192.168.0.2 is now UP

4) nodetool을 통한 확인

Address	DC	Rack	Status	State	Load	Effective-Ownership		Token
85070591730234615865843651857942052864								
192.168.0.1	162	249		Up	Normal	35.43	MB	60.88%
18510561381408809963308084773099337758								
192.168.0.2	162	249		Up	Normal	58.67	MB	39.12%
85070591730234615865843651857942052864								