

# Cassandra kezd lépések

Letöltjük, kicsomagoljuk és létrehozuk a szükséges könyvtárakat a megfelel jogokkal:

```
# adduser -g users -m cassandra
# su - cassandra
$ wget http://downloads.datastax.com/community/dsc-cassandra-2.1.2-bin.tar.gz
$ tar xzvf dsc-cassandra-2.1.2-bin.tar.gz
```

Egy kicsit szerkesztünk a konfigurációs állományon, ahol meg kell adni azokat a gépneveket vagy IP címeket, amelyeken a SEED szerverek fognak futni:

## apache-cassandra-2.0.9/conf/cassandra.yaml

```
$ diff cassandra.yaml.orig cassandra.yaml
10c10
< cluster_name: 'Test Cluster'
---
> cluster_name: 'gacivs'
261c261
<         - seeds: "127.0.0.1"
---
>         - seeds: "10.129.246.225,10.129.171.246,10.129.211.128,10.129.215.37,10.129.247.19"
367c367
< listen_address: localhost
---
> listen_address: 10.129.247.19
408c408
< rpc_address: localhost
---
> rpc_address: 10.129.247.19
```

Ezek után indítsuk el a Cassandra példányokat (egyelre eltérben, majd késbb faraghatunk hozzá indító scripteket):

```
$ apache-cassandra-2.0.9/bin/cassandra -f
INFO 15:39:19,534 Logging initialized
INFO 15:39:19,595 Loading settings from file:/home/cassandra/apache-cassandra-2.0.9/conf/cassandra.yaml
INFO 15:39:20,316 Data files directories: [/var/lib/cassandra/data]
INFO 15:39:20,317 Commit log directory: /var/lib/cassandra/commitlog
INFO 15:39:20,317 DiskAccessMode 'auto' determined to be mmap, indexAccessMode is mmap
INFO 15:39:20,318 disk_failure_policy is stop
INFO 15:39:20,318 commit_failure_policy is stop
INFO 15:39:20,327 Global memtable threshold is enabled at 59MB
INFO 15:39:20,539 Not using multi-threaded compaction
INFO 15:39:21,028 JVM vendor/version: OpenJDK 64-Bit Server VM/1.7.0_51
...
INFO 15:39:32,566 No gossip backlog; proceeding
INFO 15:39:32,789 Starting listening for CQL clients on gacivs-test02/10.129.215.37:9042...
INFO 15:39:32,919 Using TFRamedTransport with a max frame size of 15728640 bytes.
INFO 15:39:32,921 Binding thrift service to gacivs-test02/10.129.215.37:9160
INFO 15:39:32,933 Using synchronous/threadpool thrift server on gacivs-test02 : 9160
INFO 15:39:32,934 Listening for thrift clients...
```

Majd várjuk meg, amíg a SEED szervereken át az összes kliens csatlakozik, bár erre nem feltétlen van szükség, majd szinkronizálják a dolgaikat:

```
INFO 15:40:29,351 Handshaking version with gacivs-test01/10.129.216.43
INFO 15:40:30,239 Node /10.129.216.43 is now part of the cluster
INFO 15:40:30,255 Handshaking version with gacivs-test01/10.129.216.43
INFO 15:40:30,269 InetAddress /10.129.216.43 is now UP
```

Nézzük meg, hogy minden rendben van-e:

```
$ apache-cassandra-2.0.9/bin/nodetool -host gacivs02 -p 7199 status
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address          Load          Tokens   Owns (effective)  Host ID                               Rack
UN  10.129.215.37    40.85 KB      256      100.0%            db1a6512-a215-4fef-8f92-d858fb4cb8f2  rack1
UN  10.129.216.43    57.15 KB      256      100.0%            1aa61570-52f2-49e5-8a2a-dee9a9845638  rack1
```

Majd csatlakozunk a parancssoros CQL felületen és hozunk létre egy "test" nev kulcsterületet (megfelel replikációs szinttel), benne egy teszteléshez használható táblát:

```
$ apache-cassandra-2.0.9/bin/cqlsh gacivs-test01
Connected to gacivs at gacivs-test01:9160.
[cqlsh 4.1.1 | Cassandra 2.0.9 | CQL spec 3.1.1 | Thrift protocol 19.39.0]
Use HELP for help.
cqlsh>
cqlsh> CREATE KEYSPACE test WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 2 };
cqlsh>
cqlsh> USE test;
cqlsh:test>
cqlsh:test> CREATE TABLE test ( id int PRIMARY KEY, subject text, description text );
cqlsh:test>
```

A Cassandra naplóban látjuk, hogy létrehozta a táblát:

```
INFO 15:45:08,722 Loading org.apache.cassandra.config.CFMetaData@f4870e6[cfId=05a671c6-6aef-3a12-8cc0-8b76ea6d30bb,ksName=test,cfName=test,cfType=Standard,comparator=org.apache.cassandra.db.marshall.CompositeType(org.apache.cassandra.db.marshall.UTF8Type),comment=,readRepairChance=0.1,dcllocalReadRepairChance=0.0,replicateOnWrite=true,gcGraceSeconds=864000,defaultValidator=org.apache.cassandra.db.marshall.BytesType,keyValidator=org.apache.cassandra.db.marshall.Int32Type,minCompactionThreshold=4,maxCompactionThreshold=32,column_metadata={java.nio.HeapByteBuffer[pos=0 lim=2 cap=2]=ColumnDefinition{name=6964, validator=org.apache.cassandra.db.marshall.Int32Type, type=PARTITION_KEY, componentIndex=null, indexName=null, indexType=null}, java.nio.HeapByteBuffer[pos=0 lim=11 cap=11]=ColumnDefinition{name=6465736372697074696f6e, validator=org.apache.cassandra.db.marshall.UTF8Type, type=REGULAR, componentIndex=0, indexName=null, indexType=null}, java.nio.HeapByteBuffer[pos=0 lim=7 cap=7]=ColumnDefinition{name=7375626a656374, validator=org.apache.cassandra.db.marshall.UTF8Type, type=REGULAR, componentIndex=0, indexName=null, indexType=null}},compactionStrategyClass=class org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy,compactionStrategyOptions={},compressionOptions={sstable_compression=org.apache.cassandra.io.compress.LZ4Compressor},bloomFilterFpChance=0.01,memtable_flush_period_in_ms=0,caching=KEYS_ONLY,defaultTimeToLive=0,speculative_retry=99.0PERCENTILE,indexInterval=128,populateIoCacheOnFlush=false,droppedColumns={},triggers={}]
INFO 15:45:08,726 Initializing test.test
```

Ezek után töltünk bele adatokat, majd kérdezzük vissza:

```
cqlsh:test> INSERT INTO test (id, subject, description) VALUES(1, 'test1', 'test1 description');
cqlsh:test> INSERT INTO test (id, subject, description) VALUES(2, 'test2', 'test2 description');
cqlsh:test> INSERT INTO test (id, subject, description) VALUES(3, 'test3', 'test3 description');
cqlsh:test> SELECT * FROM test;
 id | description          | subject
-----+-----+-----
  1 | test1 description    | test1
  2 | test2 description    | test2
  3 | test3 description    | test3
(3 rows)
cqlsh:test>
```

Kíváncsiságból csinálhatunk tesztek, hogy leállítgatjuk a Cassandra példányokat, miközben írunk és olvasunk, de ez jól fog mködni.

## Csatlakozás Java programból

Szükséges egy Maven projekt, benne egy darab Maven függőség:

```
<dependency>
  <groupId>com.datastax.cassandra</groupId>
  <artifactId>cassandra-driver-core</artifactId>
  <version>2.0.1</version>
</dependency>
```

Majd próbáljunk meg csatlakozni és adatokat lekérdezeni:

```

public class Main {
    private final Cluster cluster;

    public Main(final String[] nodes) {
        try {
            this.cluster = Cluster.builder().addContactPoints(nodes).build();
        } catch (DriverException ex) {
            System.out.println(ex.toString());
            throw ex;
        }
    }

    public final void printMetadata() {
        final Metadata metadata = this.cluster.getMetadata();
        System.out.printf("Connected to cluster: %s\n", metadata.getClusterName());
        for (final Host host : metadata.getAllHosts()) {
            System.out.printf("Datacenter: %s; Host: %s; Rack: %s\n",
                host.getDatacenter(), host.getAddress(), host.getRack());
        }
    }

    public final void close() {
        cluster.close();
    }

    public final ResultSet query(final String cql) {
        try (final Session session = cluster.connect("test")) {
            return session.execute(cql);
        }
    }

    public static void main(final String[] args) {
        Main main = new Main(args);
        main.printMetadata();
        for (Row row : main.query("SELECT * FROM test;")) {
            System.out.println(row.getInt("id"));
            System.out.println(row.getString("subject"));
            System.out.println(row.getString("description"));
        }

        main.close();
    }
}

```

Az eredmény a cluster neve, a hozzá csatlakozott node-ok száma és paraméterei, illetve a test nev táblában lévő adatok:

```

Connected to cluster: gacivs
Datacenter: datacenter1; Host: gacivs-test01/10.129.216.43; Rack: rack1
Datacenter: datacenter1; Host: gacivs-test02/10.129.215.37; Rack: rack1
1
test1a
test1 description
2
test2a
test2 description
3
test3a
test3 description

```