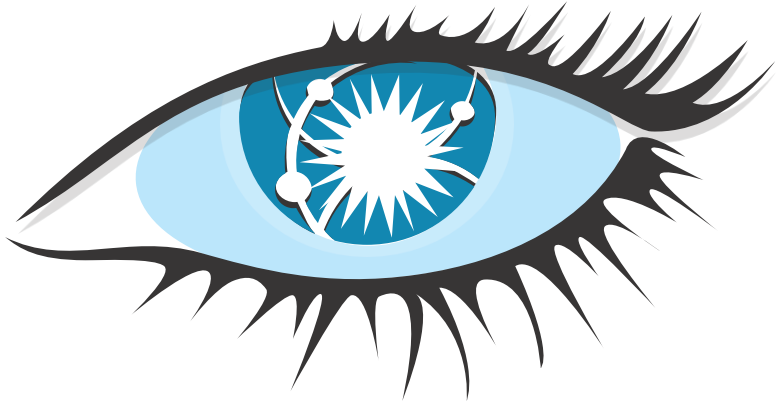
Apache Cassandra

The goal is to create Apache Cassandra clusters of about 5 nodes each. Each cluster will be run by one team and each team member will run one cluster node.



Please choose your team members now.

Apache Cassandra

- Boot up your CIP-Pool Host and choose Ubuntu (not Windows!)
- Download Apache Cassandra "Latest GA Version" as tar.gz
- Start a terminal (e.g. **konsole**). Tip: Switch to a monospaced font.
- Unpack it into your home directory (tar xvfz filename.tar.gz)
- Determine the lowest and highest IP address in your team. This will be the **seed hosts**
- Edit the file conf/cassandra.yaml.
  - Comment out the **listen_address** directive
  - Comment in the **listen_interface** directive and set it to **eno1**
  - Set the value of the **seeds** key to the IPs of the seed hosts determined earlier
  - Agree on a cluster name and set the value of the **cluster_name** key accordingly
- Change into the **bin/** directory and start the **./cassandra** script one node after another, starting with the seed nodes. Wait at least a minute between starts.
- Watch out for error messages

Apache Cassandra

Checking the cluster:

- Open another terminal window or tab
- Change into the **bin/** directory
- Start the script **./nodetool status**
- The output should look something like:

```
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address        Load        Tokens  Owns (effective)  Host ID
UN  172.20.38.87   79.2 KiB    16      60.3%             720a6a7e-2615-4f77-8e4a-652870ea
UN  172.20.38.88   140.66 KiB  16      60.9%             54b90b8d-ab37-4abe-b1eb-1da24ec7
UN  172.20.38.85   79.28 KiB   16      60.0%             6f0a4343-97ed-49c5-8255-b9f94cab
UN  172.20.38.86   79.2 KiB    16      58.7%             e12d7735-4b94-4207-8c8c-d652930(
UN  172.20.38.89   199.63 KiB  16      60.1%             5956d51f-9ab4-448e-aa2b-abf9c55e
```

- You might have to wait a bit. If it still doesn't work after some time:
  - Check if your java processes still run (ps ax), restart cassandra if not
  - Watch out for error messages
  - Cry for help

Apache Cassandra

Change into the **bin/** directory and fire up the cli client with **./cqlsh**

Create a keyspace (similar to a database)

Every team member can create his own keyspace, use different names

Use a replication factor of 3 at first

The syntax is:

```
CREATE KEYSPACE <name>
       WITH REPLICATION = {
                            'class' : 'SimpleStrategy',
                            'replication_factor' : <factor>
                          };
```

Apache Cassandra

Now create a table and insert some data. The syntax is similar to SQL.

Example:

```
CREATE TABLE people (id INT PRIMARY KEY, name TEXT);
```

```
INSERT INTO people (id, name) VALUES (1,'Annie');
INSERT INTO people (id, name) VALUES (2,'Bernhard');
INSERT INTO people (id, name) VALUES (3,'Charlotte');
INSERT INTO people (id, name) VALUES (4,'David');
INSERT INTO people (id, name) VALUES (5,'Esmeralda');
     ...
```

Insert at least 10 rows

Discuss with your team mates, how the data is probably distributed

Apache Cassandra

Some handy commands:

cqlsh: CONSISTENCY (ONE|TWO|THREE|QUORUM|ALL) -- sets the consistency level for all following commands

Linux shell: killall -(STOP|CONT) java -- pause or continue the local node

Linux shell: ./nodetool stopdaemon -- stop the local node

Linux shell (in bin/): ./cassandra -- start the local node

Linux shell (in bin/): ./nodetool status -- check the cluster status

Apache Cassandra

- Set the consistency level ALL
- Now stop the two cluster nodes with the highest IP addresses
- Ask for some of the rows by their id (works like an SQL select)
- Try to explain what you observe
- What changes with a consistency level of TWO? (first discuss, then try)

Apache Cassandra

- Set the consistency level to ALL again, leave the two nodes stopped.
- Try to insert some data
- Use different primary keys for your attempts
- Again, try to explain what you observe

Apache Cassandra

- We will try to simulate a network partition now:
- First, we have to find out some data records, that is also located on the two stopped nodes
- Set the consistency level to TWO, leave the two nodes stopped.
- Try to select random data records by id, until an error occurs
- Now set the consistency level to ONE and update the record by inserting a different name for the same id

Apache Cassandra

- Stop the three running nodes now (wait until they have really shut down), then start the other two nodes again.
- Set the consistency level to TWO and try to retrieve the ID you have updated. What happens?
- Set the consistency level to ONE and try again. Which value do you get?
- Now start the three stopped nodes one by one, after each start try to see what happens to the value on the node of the previous step.
- Discuss what has happened

Apache Cassandra

- There is a nodetool-function how to find out, which nodes contain a specific data record by primary key. Find out which one this is and test it. Make sure the information is correct by using CONSISTENCY and shutdown nodes.
- Develop an own test scenario to test replication or sharding behaviour of Apache Cassandra. Perform the test and evaluate the results.