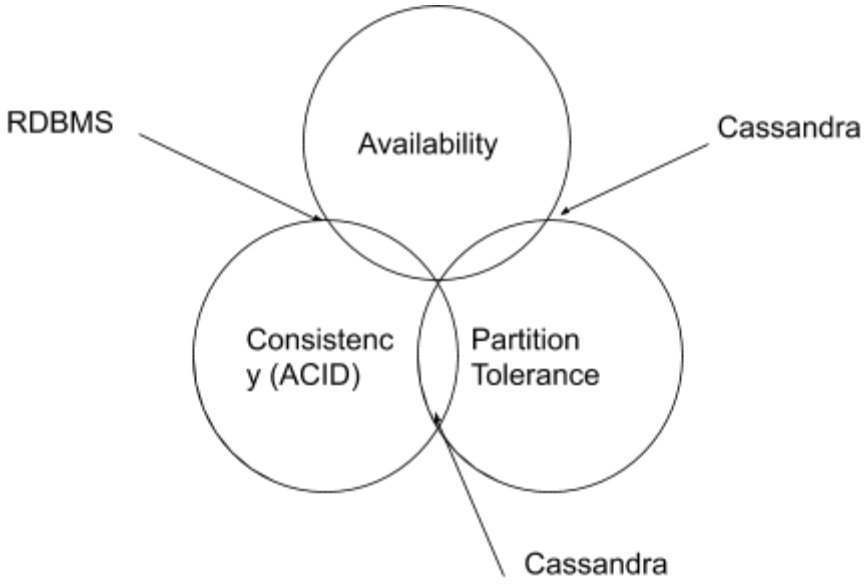
Lab 11: Cassandra

Agenda:

1. HW review
2. Local Cassandra setup
3. working with Cassandra from Java
4. working with Cassandra from Python
5. Cassandra tools
6. Cassandra cluster setup with Docker

Cassandra:

CAP Theorem



1. Fast Distributed (Column Family NoSQL)
2. Database High availability Linear Scalability High Performance
3. Fault tolerant
4. Multi-Data Center Support
5. Easy to operate

Cassandra Data Modeling

1. Access Pattern data modeling
2. Spread data evenly across the cluster
3. Use denormalization

Local Cassandra setup

Download from: http://cassandra.apache.org/download/

Start: **./bin/cassandra**
CQLSH: **./bin/cqlsh localhost**

Stop Cassandra process on tarball installation:

Find the Cassandra Java process ID (PID), and then kill the process using its PID number:
$ ps auwx | grep CassandraDaemon
$ kill pid

Cassandra with Python:

pip install cassandra-driver

Use driver:
http://datastax.github.io/python-driver/getting_started.html#asynchronous-queries

Cassandra tools

./bin/nodetool

./bin/nodetool info - quick summary of the node

./bin/nodetool version - version of cassandra

./bin/nodetool status <keyspace_name> - status of cluster/keyspace

./bin/nodetool cfstats <keyspace_name.table_name> - info on the table

CREATE KEYSPACE lab10 WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };

```
CREATE TABLE IF NOT EXISTS lab10.sensor_metrics (
        sensor_id UUID,
        time_hour timestamp,
        sensor_type text,
        reading_time timestamp,
        metric float,
        PRIMARY KEY ((sensor_type, time_hour), reading_time)
) WITH CLUSTERING ORDER BY (reading_time asc);
```

```
INSERT INTO lab10.sensor_metrics
  (sensor_type, time_hour, reading_time, sensor_id, metric)
  VALUES ('p_type1', '2019-11-11 21:00:00', '2019-11-11 21:05:00', uuid(), 1.0);
```

```
SELECT * from lab10.sensor_metrics WHERE sensor_type='p_type1' AND time_hour =
'2019-11-11 21:00:00';
```

Docker setup for Cassandra cluster

docker run --name cassandra-node1 -d -p 9042:9042 -e CASSANDRA_CLUSTER_NAME=Lab10 -e CASSANDRA_ENDPOINT_SNITCH=GossipingPropertyFileSnitch -e CASSANDRA_DC=datacenter1 cassandra

docker inspect --format='{{ .NetworkSettings.IPAddress }}' cassandra-node1

docker run --name cassandra-node2 -d -e CASSANDRA_SEEDS="$(docker inspect --format='{{ .NetworkSettings.IPAddress }}' cassandra-node1)" -e CASSANDRA_CLUSTER_NAME=Lab10 -e CASSANDRA_ENDPOINT_SNITCH=GossipingPropertyFileSnitch -e CASSANDRA_DC=datacenter1 cassandra

Execute below statement to check nodetool status to check the status of cluster . Don't proceed to node3 untill you see second node in UN
docker exec cassandra-node1 nodetool status

docker run --name cassandra-node3 -d -e CASSANDRA_SEEDS="$(docker inspect --format='{{ .NetworkSettings.IPAddress }}' cassandra-node1)" -e CASSANDRA_CLUSTER_NAME=Lab10 -e CASSANDRA_ENDPOINT_SNITCH=GossipingPropertyFileSnitch -e CASSANDRA_DC=datacenter2 cassandra

Execute below statement to check nodetool status to check the status of cluster .Don't proceed until you see all three nodes in UN

docker exec cassandra-node1 nodetool status

Execute below statement to connect to node-1 and connect to cqlsh session
docker exec -it cassandra-node1 cqlsh

```
CREATE KEYSPACE lab10
WITH replication = {
        'class' : 'NetworkTopologyStrategy',
        'datacenter1' : 2,
        'datacenter2' : 1
};

CREATE TABLE lab10.person (
        id int primary key,
        name text
);
```

docker stop cassandra-node1 cassandra-node2 cassandra-node3

CREATE KEYSPACE lab10 WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };

```
CREATE TABLE IF NOT EXISTS lab10.sensor_metrics (
        sensor_id UUID,
        time_hour timestamp,
        sensor_type text,
        reading_time timestamp,
        metric float,
        PRIMARY KEY ((sensor_type, time_hour), reading_time)
) WITH CLUSTERING ORDER BY (reading_time asc);
```

```
INSERT INTO lab10.sensor_metrics
  (sensor_type, time_hour, reading_time, sensor_id, metric)
  VALUES ('p_type1', '2019-11-11 21:00:00', '2019-11-11 21:05:00', uuid(), 1.0);
```

```
SELECT * from lab10.sensor_metrics WHERE sensor_type='p_type1' AND time_hour =
'2019-11-11 21:00:00';
```