Getting to know

by Michelle Darling
mdarlingcmt@gmail.com
August 2013

Agenda:

- What is Cassandra?
- Installation, CQL3
- Data Modelling
- Summary

Only 15 min to cover these, so please hold questions til the end, or **email me** :-) and I'll summarize Q&A for everyone.

Unfortunately, no time for:

- DB Admin
  - Detailed Architecture
  - Partitioning / Consistent Hashing
  - Consistency Tuning
  - Data Distribution & Replication
  - System Tables
- App Development
  - Using Python, Ruby etc to access Cassandra
  - Using Hadoop to stream data into Cassandra

What is Cassandra?

“Fortuneteller of Doom”

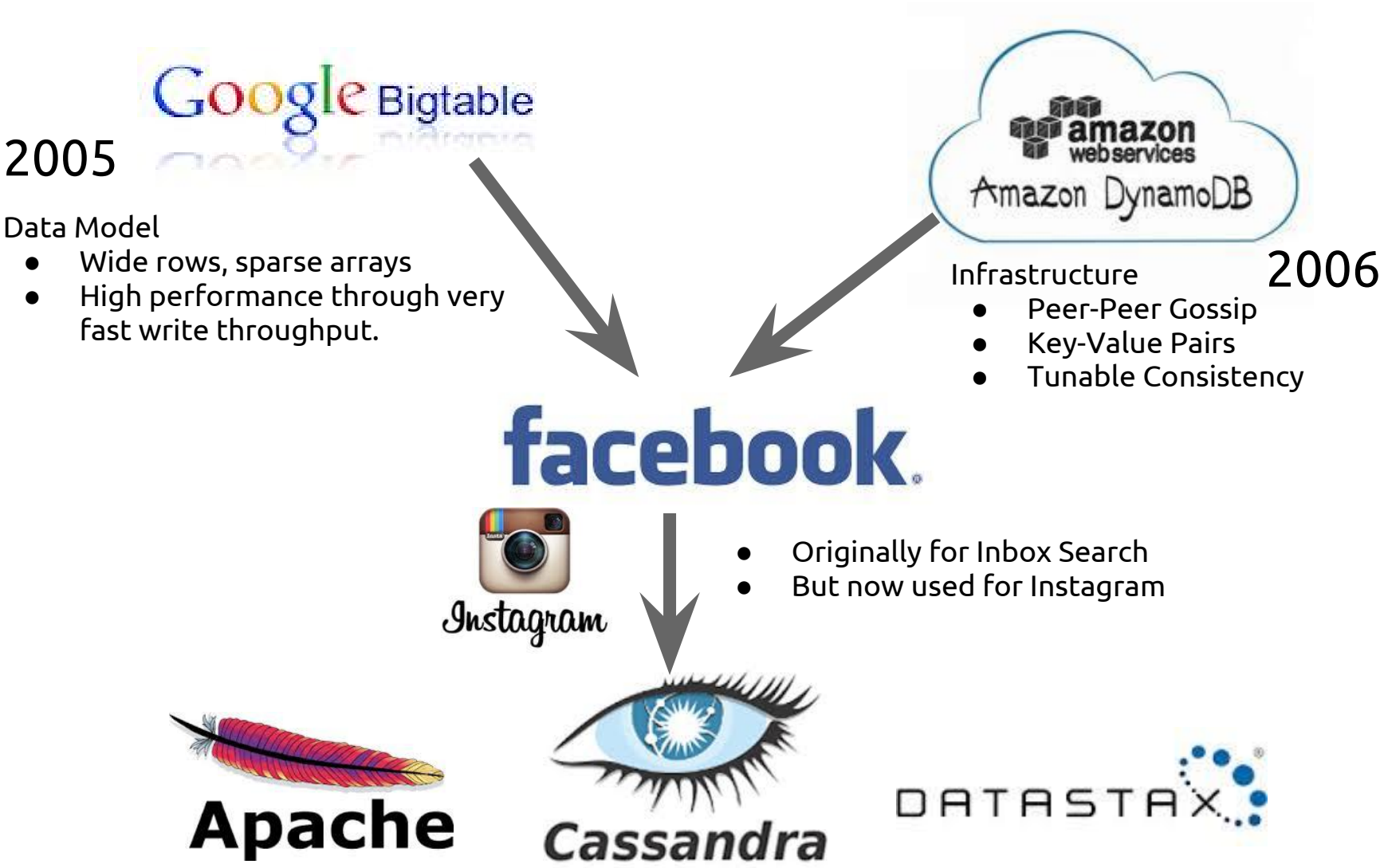
from Greek Mythology. Tried to warn others about future disasters, but no one listened. Unfortunately, she was 100% accurate.

NoSQL Distributed DB

- **Consistency** - A__ID
- **Availability** - High
- **Point of Failure** - none
- **Good for Event Tracking & Analysis**
  - Time series data
  - Sensor device data
  - Social media analytics
  - Risk Analysis
  - Failure Prediction

Rackspace: “Which servers are under heavy load and are about to crash?”

The Evolution of Cassandra



2008: Open-Source Release / 2013: Enterprise & Community Editions

Other NoSQL vs. Cassandra

NoSQL Taxonomy:

- Key-Value Pairs
  - Dynamo, Riak, Redis
- Column-Based
  - BigTable, HBase, **Cassandra**
- Document-Based
  - MongoDB, Couchbase
- Graph
  - Neo4J

Big Data Capable

C* Differentiators:

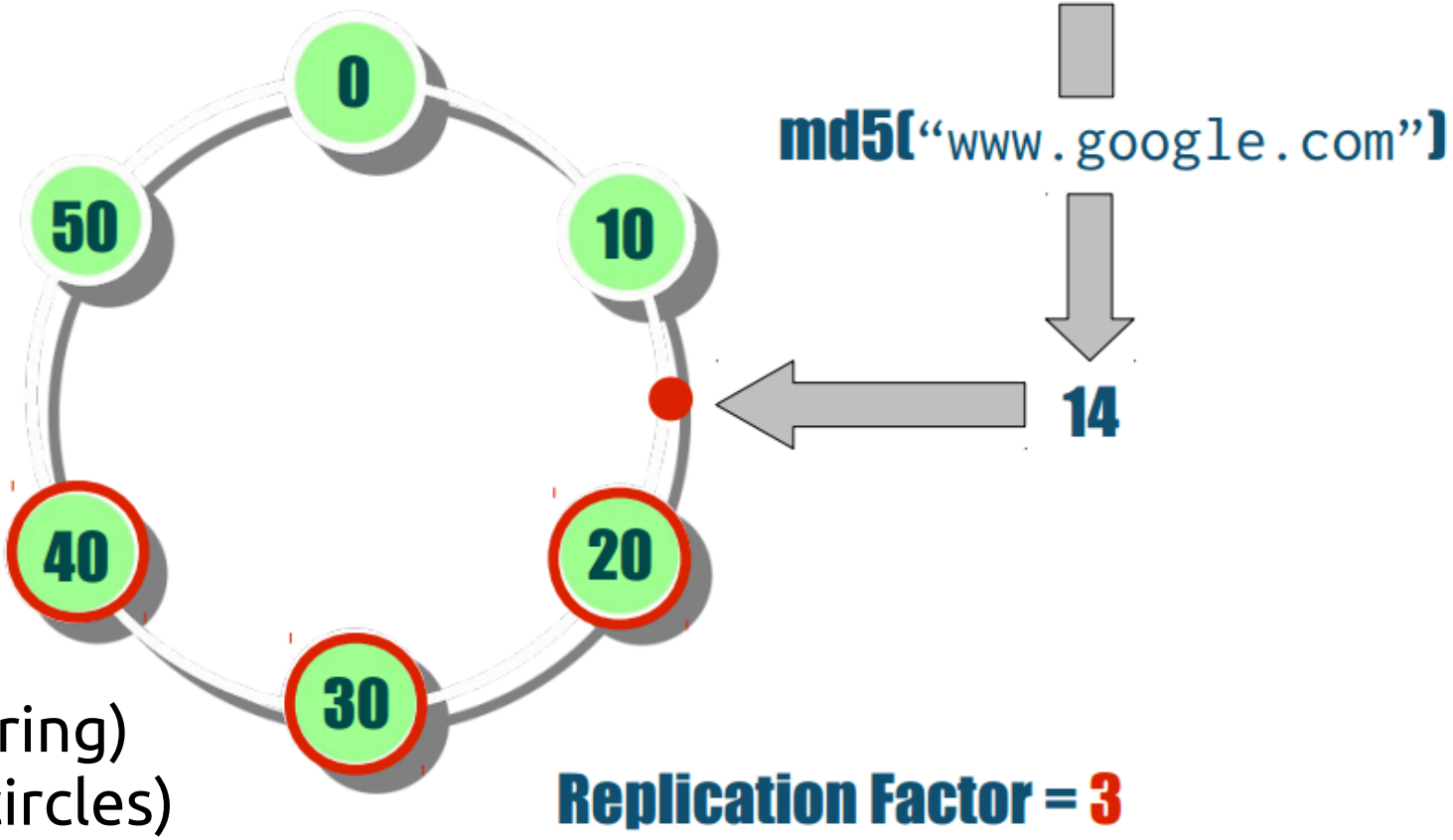
- Production-proven at Netflix, eBay, Twitter, 20 of Fortune 100
- "Clear Winner" in Scalability, Performance, Availability

Cassandra – affectionately **"C*"** – is an open source, distributed store for structured data that scales-out on cheap, commodity hardware and stays up **even when things get really bad.**

-- DataStax

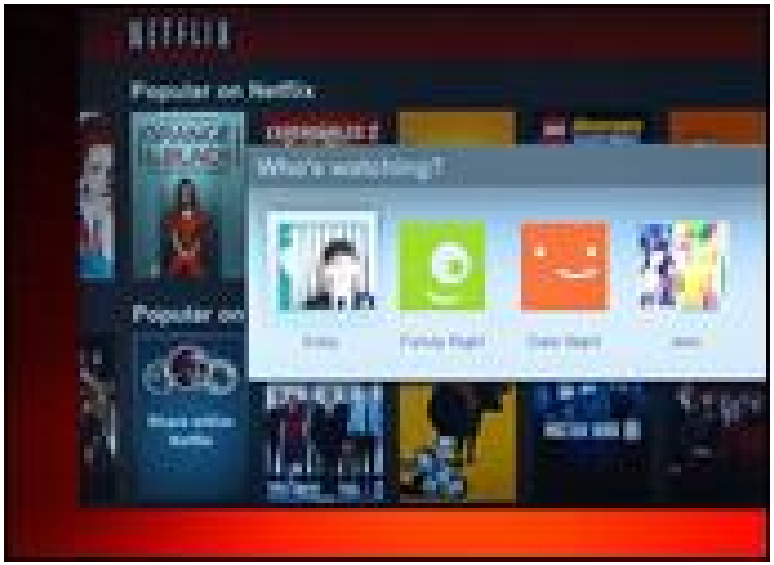
Architecture

Partitioner:
Consistent Hashing



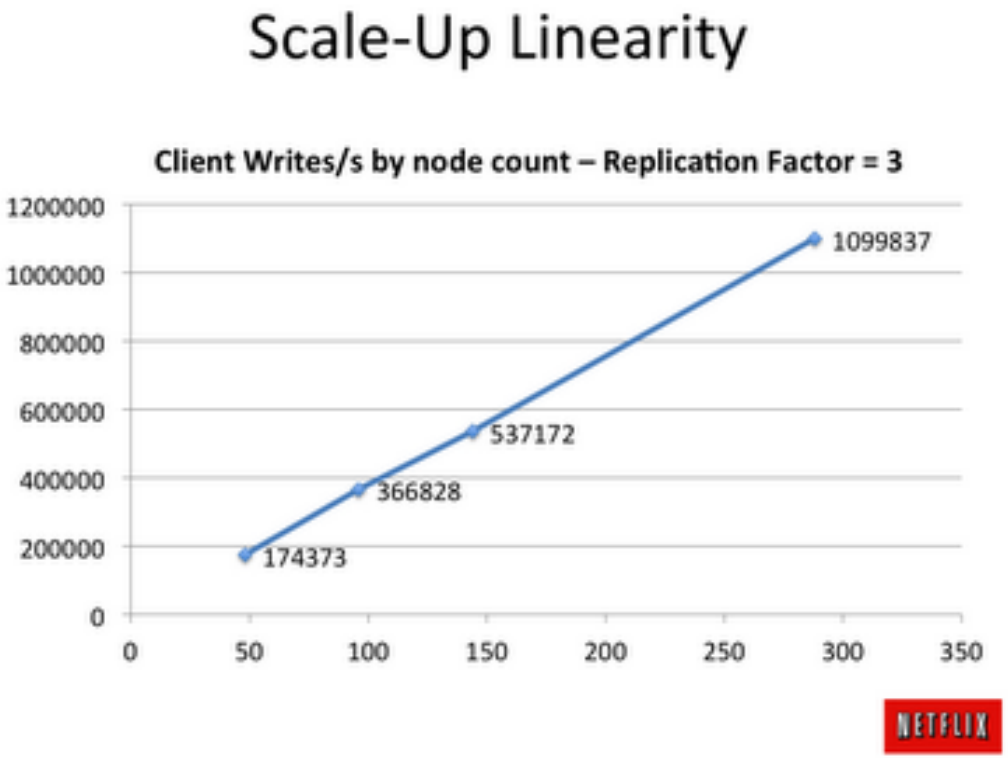
- Cluster (ring)
- Nodes (circles)
- Peer-to-Peer Model
- Gossip Protocol

“Netflix foretells 'House of Cards' success with Cassandra big data engine [Big data algorithms helped Netflix map the viewing patterns of its customers”]



Netflix
Streaming Video

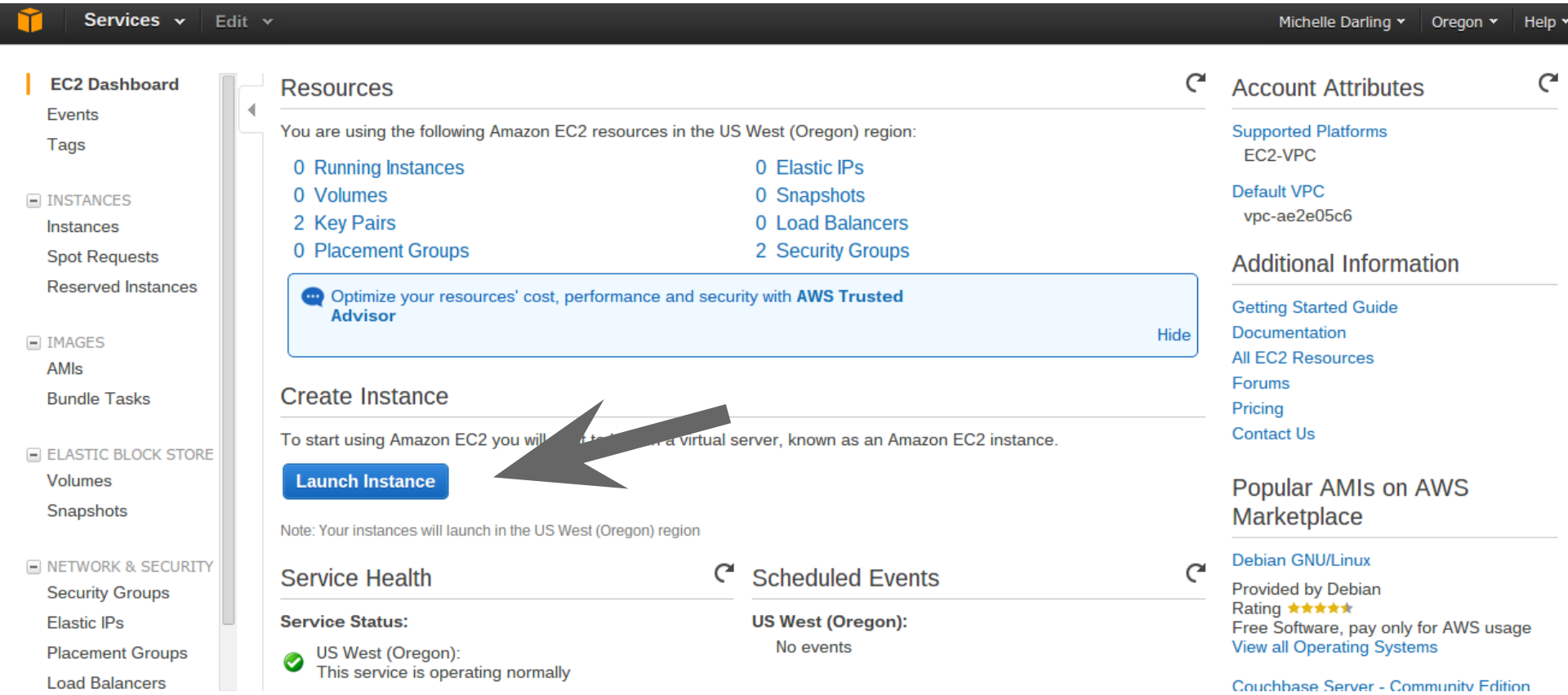
- Personalized Recommendations per family member
- Built on Amazon Web Services (AWS) + Cassandra



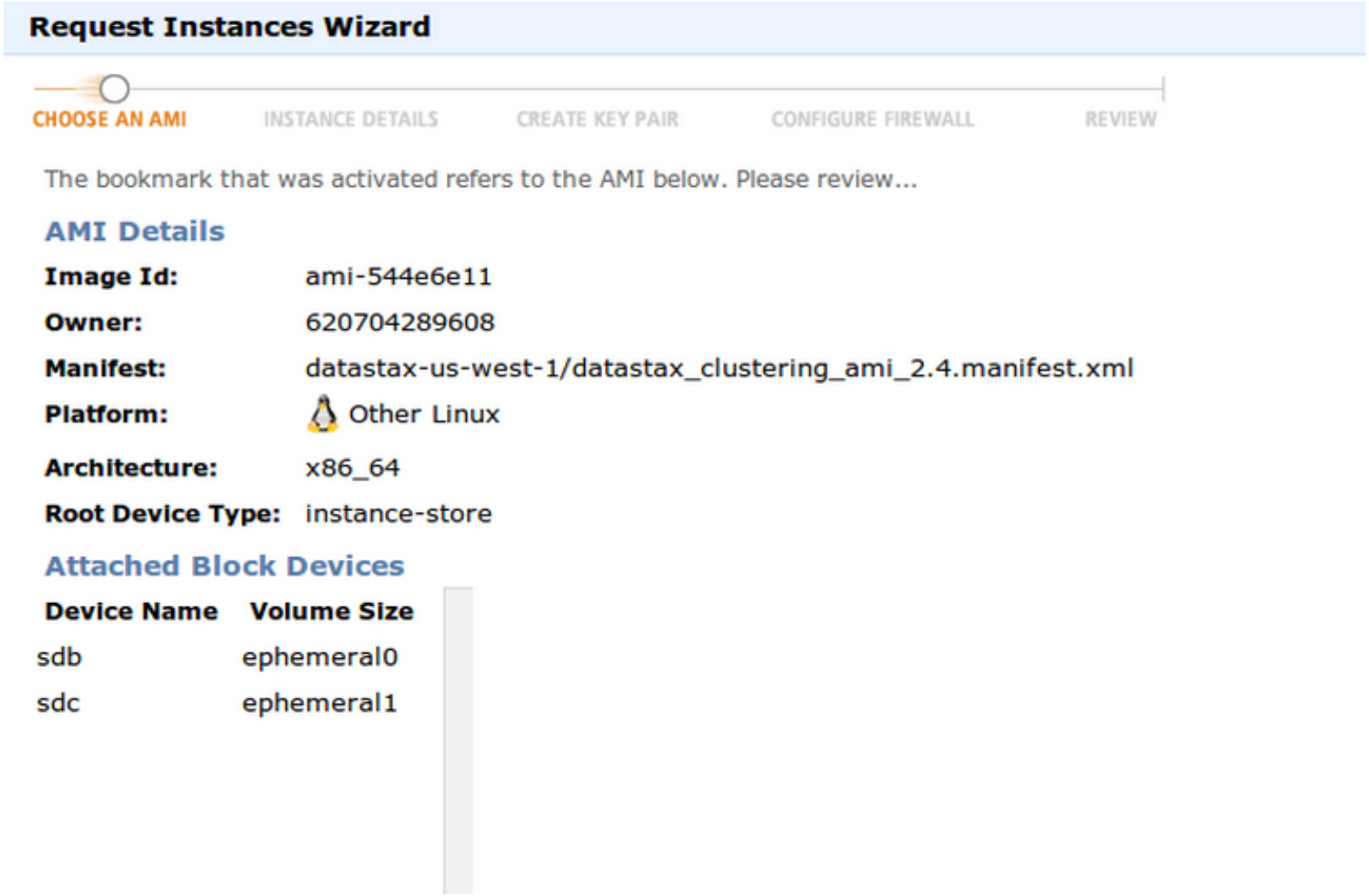
Cloud installation using 

- **Amazon Web Services (AWS)**
- **Elastic Compute Cloud (EC2)**
  - Free for the 1st year! Then pay only for what you use.
  - Sign up for AWS EC2 account: Big Data University Video 4:34 minutes,
- **Amazon Machine Image (AMI)**
  - Preconfigured installation template
  - Choose: **"DataStax AMI for Cassandra Community Edition"**
  - Follow these *very good* step-by-step instructions from DataStax.
  - AMIs also available for CouchBase, MongoDB (make sure you pick the free tier community versions to avoid monthly charge$$!!!).

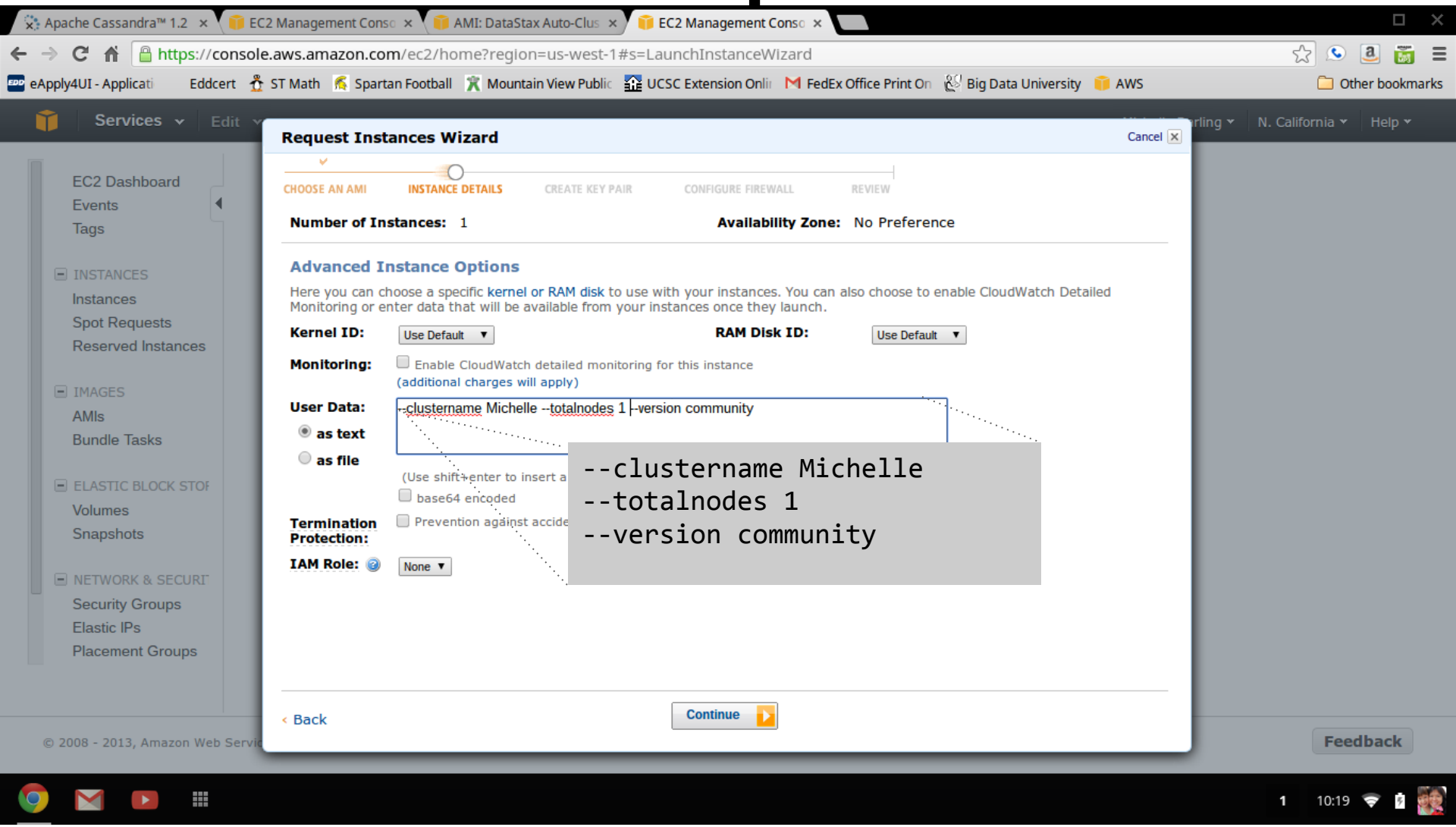
AWS EC2 Dashboard



DataStax AMI Setup



DataStax AMI Setup



“Roll your Own” Installation

DataStax Community Edition

- **Install instructions**
  For Linux, Windows, MacOS:
  http://www.datastax.com/2012/01/getting-started-with-cassandra
- **Video:** “Set up a 4-node Cassandra cluster in under 2 minutes”
  http://www.screenr.com/5G6

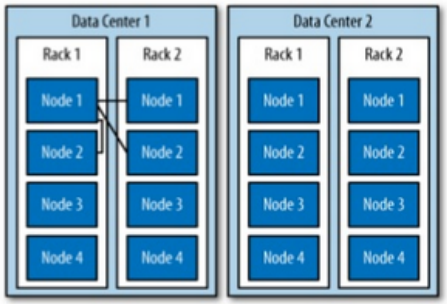
Invoke CQLSH, CREATE KEYSPACE

```
./bin/cqlsh

cqlsh> CREATE KEYSPACE big_data
... with strategy_class = ‘org.apache.cassandra.
locator.SimpleStrategy’
... with strategy_options:replication_factor=‘1’;

cqlsh> use big_data;
cqlsh:big_data>
```

- SimpleStrategy - returns the nodes that are next to each other on the ring.



Tip: Skip Thrift -- use CQL3

Thrift RPC

```
// Your Column
Column col = new Column(ByteBuffer.wrap("name".
getBytes()));
col.setValue(ByteBuffer.wrap("value".getBytes()));
col.setTimestamp(System.currentTimeMillis());
// Don't ask
ColumnOrSuperColumn cosc = new ColumnOrSuperColumn();
cosc.setColumn(col);
// Prepare to be amazed
Mutation mutation = new Mutation();
mutation.setColumnOrSuperColumn(cosc);
List<Mutation> mutations = new ArrayList<Mutation>();
mutations.add(mutation);
Map mutations_map = new HashMap<ByteBuffer, Map<String,
List<Mutation>>>();
Map cf_map = new HashMap<String, List<Mutation>>();
cf_map.set("Standard1", mutations);
mutations_map.put(ByteBuffer.wrap("key".getBytes()),
cf_map);
cassandra.batch_mutate(mutations_map,
consistency_level);
```

CQL3

- Uses `cqlsh`
- “SQL-like” language
- Runs on top of Thrift RPC
- Much more user-friendly.

Thrift code on left
equals this in CQL3:

```
INSERT INTO (id, name)
VALUES ('key',
'value');
```

CREATE TABLE


```
cqlsh:big_data> create table user_tags (
... user_id varchar,
... tag varchar,
... value counter,
... primary key (user_id, tag)
...):
```

- **TABLE user_tags**: “How many times has a user mentioned a hashtag?”
- **COUNTER** datatype - Computes & stores counter value at the time data is written. This optimizes query performance.

UPDATE TABLE
SELECT FROM TABLE

```
cqlsh:big_data> UPDATE user_tags SET
value=value+1 WHERE user_id = ‘paul’ AND tag =
‘cassandra’


cqlsh:big_data> SELECT * FROM user_tags

user_id | tag       | value
--------+-----------+----------
paul    | cassandra | 1
```

DATA MODELING
A Major Paradigm Shift!

RDBMS	Cassandra
Structured Data, Fixed Schema	Unstructured Data, Flexible Schema
“Array of Arrays” 2D: ROW x COLUMN	“Nested Key-Value Pairs” 3D: ROW Key x COLUMN key x COLUMN values
DATABASE	KEYSPACE
TABLE	TABLE a.k.a COLUMN FAMILY
ROW	ROW a.k.a PARTITION. Unit of replication.
COLUMN	COLUMN [Name, Value, Timestamp]. a.k.a CLUSTER. Unit of storage. Up to 2 billion columns per row.
FOREIGN KEYS, JOINS, ACID Consistency	Referential Integrity not enforced, so A_CID. BUT relationships represented using COLLECTIONS.

Rows

Columns

Column names

	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7329	SMITH	CLERK	7902	17–DEC–88	800.00	300.00	20
7499	ALLEN	SALESMAN	7698	20–FEB–88	1600.00	300.00	30
7521	WARD	SALESMAN	7698	22–FEB–88	1250.00	500.00	30
7566	JONES	MANAGER	7839	02–APR–88	2975.00		20

Column not allowing nulls

Column allowing nulls

RDBMS
2D: Rows x columns

Cassandra
3D+: Nested Objects

(Column Family and Table mean the same.)

Example: “Twissandra” Web App

Twitter-Inspired sample application written in Python + Cassandra.

- Play with the app: twissandra.com
- Examine & learn from the code on GitHub.

Features/Queries:

- **Sign In, Sign Up**
- **Post Tweet**
- **Userline** (User’s tweets)
- **Timeline** (All tweets)
- **Following** (Users being followed by user)
- **Followers** (Users following this user)

Twissandra.com vs Twitter.com

Twissandra

Home Public Find Friends Sign out of tom

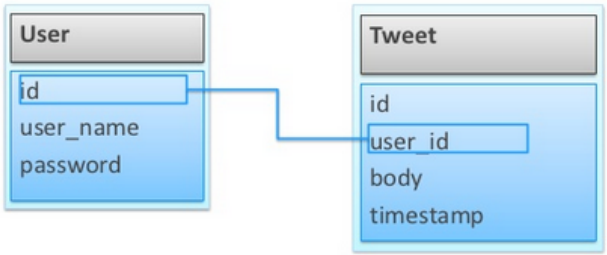
Post Tweet

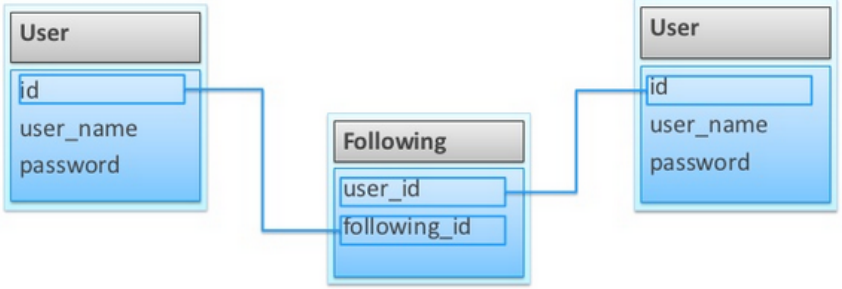
tom	Try Twissandra/you will learn a lot!
tom	Shout out to UCSC Big Data class!
tom	Hello
jim	Sorry test not interested.
tom	Hello Every!
tom	Hello, How are you today?
tom	hello from aj
tom	hi
tom	Hi Boston.
tom	Hello Tim
tom	hi

Twissandra is an example project, created to learn and demonstrate how to use Cassandra. Running the project will present a website that has similar functionality to Twitter.

View the code for this site on GitHub

Twissandra - RDBMS Version



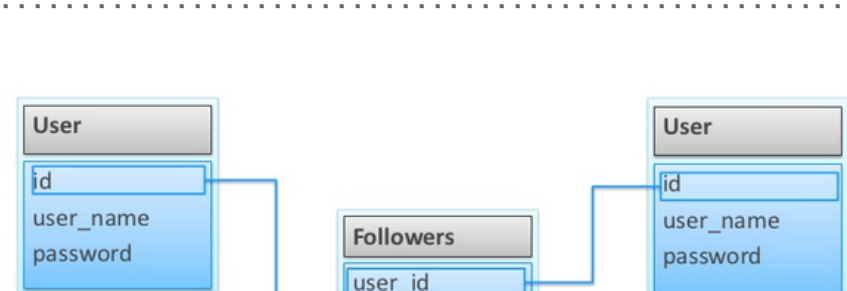


Entities

- USER, TWEET
- FOLLOWER, FOLLOWING
- FRIENDS

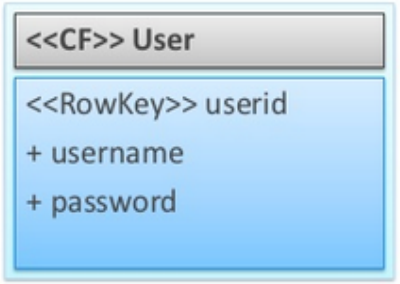
Relationships:

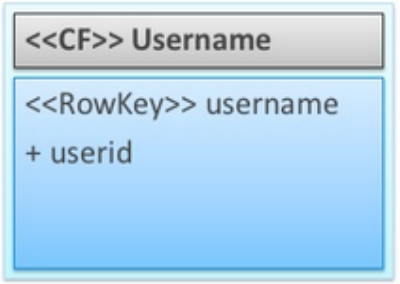
- USER has many TWEETs.
- USER is a FOLLOWER of many USERs.
- Many USERs are FOLLOWING USER.



Twissandra - Cassandra Version

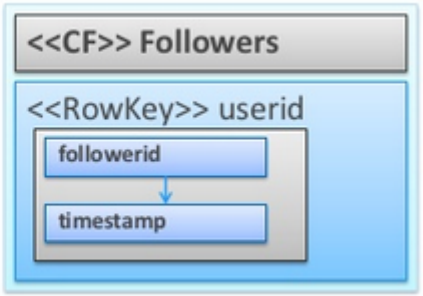
Tip: Model tables to mirror queries.





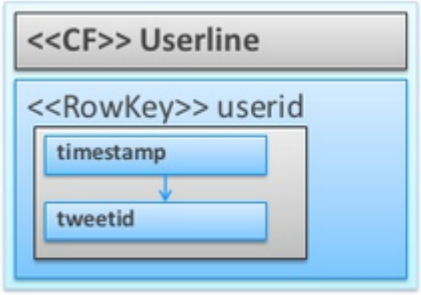
<u>TABLES or CFs</u>

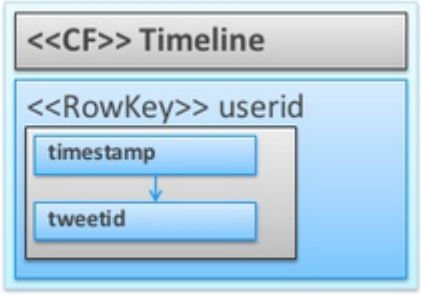
- TWEET
- USER, USERNAME
- FOLLOWERS, FOLLOWING
- USERLINE, TIMELINE



<u>Notes:</u>

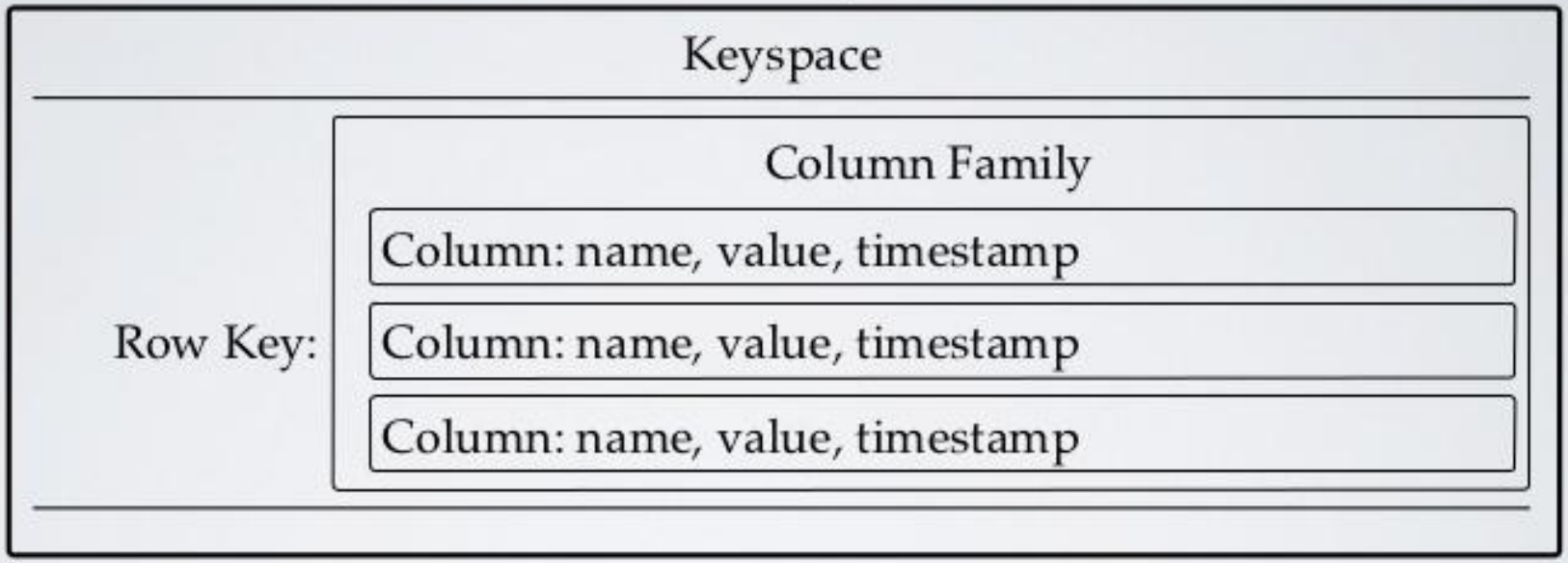
- Extra tables mirror queries.
- Denormalized tables are “pre-formed”for faster performance.





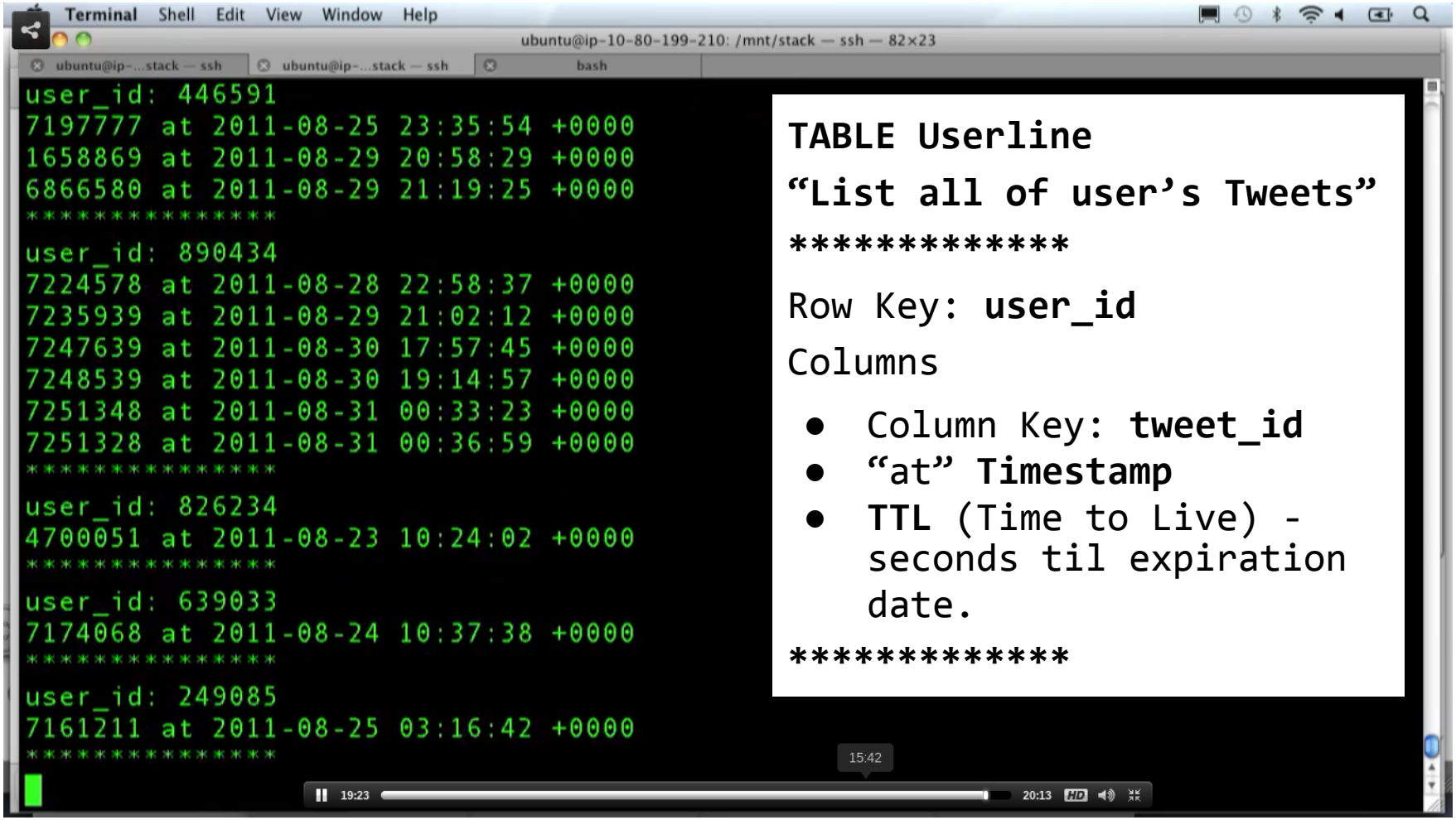
Tip: Remember, Skip Thrift -- use CQL3

Inside the ~~Column Family~~ TABLE

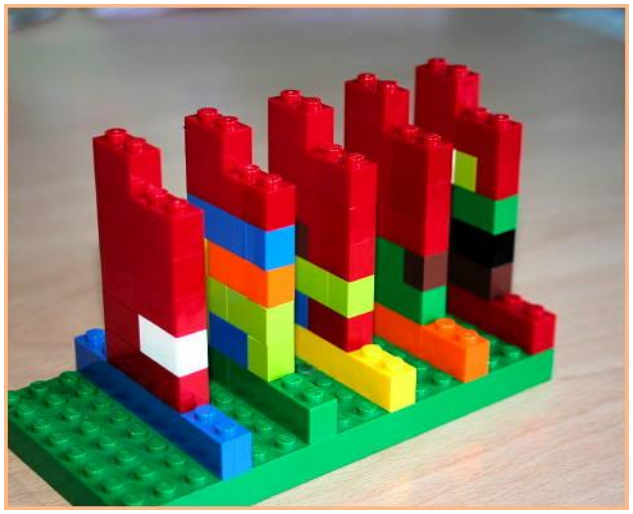


(Also TTL Columns)

What does C* data look like?



Cassandra Data Model = LEGOs?



Key1	Column	Column	Column

Key2	Column	Column

Flexible Schema

Stored sorted by column key/name →

Stored sorted by row key ↓

Row key1	Column Key1	Column Key2	Column Key3	…
	Column Value1	Column Value2	Column Value3	
⋮				

Summary:

- **Go straight from SQL to CQL3**; skip Thrift, Column Families, SuperColumns, etc
- **Denormalize tables** to mirror important queries. Roughly 1 table per impt query.
- **Choose wisely:**
  - Partition Keys
  - Cluster Keys
  - Indexes
  - TTL
  - Counters
  - Collections

See DataStax **Music Service** Example

- **Consider hybrid approach:**
  - 20% - RDBMS for highly structured, OLTP, ACID requirements.
  - 80% - Scale Cassandra to handle the rest of data.

Remember:

- **Cheap:** storage, servers, OpenSource software.
- **Precious:** User AND Developer Happiness.

Resources

C* Summit 2013:

- Slides
- Cassandra at eBay Scale (slides)
- Data Modelers Still Have Jobs - Adjusting For the NoSQL Environment (Slides)
- Real-time Analytics using Cassandra, Spark and Shark slides
- Cassandra By Example: Data Modelling with CQL3 Slides
- DATASTAX C*OLLEGE CREDIT: DATA MODELLING FOR APACHE CASSANDRA slides

I wish I found these 1st:

- How do I Cassandra? slides
- Mobile version of DataStax web docs (link)