CLOUD ENTERPRISE SYSTEMS

CSCI 5408:

Data Management, Warehousing, and Analytics

Prepared by: Suhaib Qaiser (suhaibqaiser@dal.ca)

Casandra

- What is Casandra
- Features
- Data Replication
- Components
- Query Language

Heroku

- What is Heroku
- Scaling
- Dynos
- Plugins
- Integration
- Metrics and Code deployment

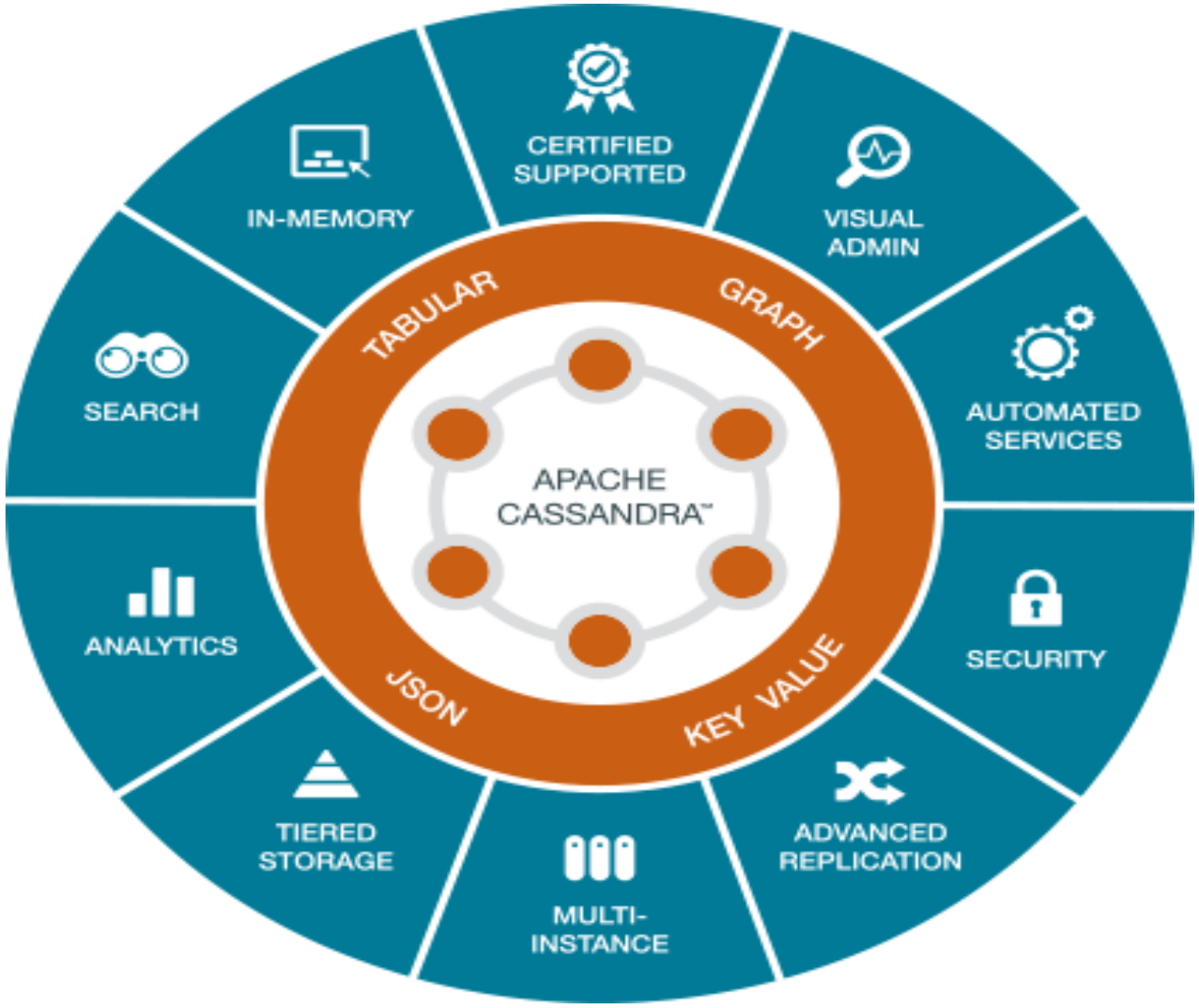
Salesforce

- What is Salesforce?
- Types of Sandboxes
- Types of Licenses
- Custom Objects
- Data Import / Export
- Custom Fields
- Lightning Framework
- App Exchange
- Trigger and Pages

Quiz

Casandra

Apache Cassandra

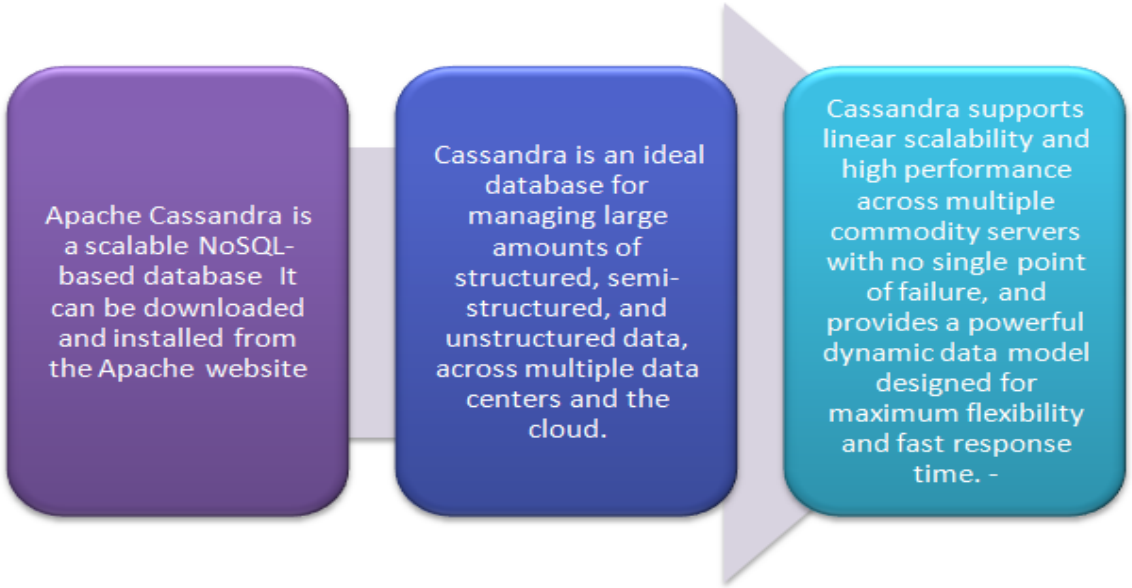


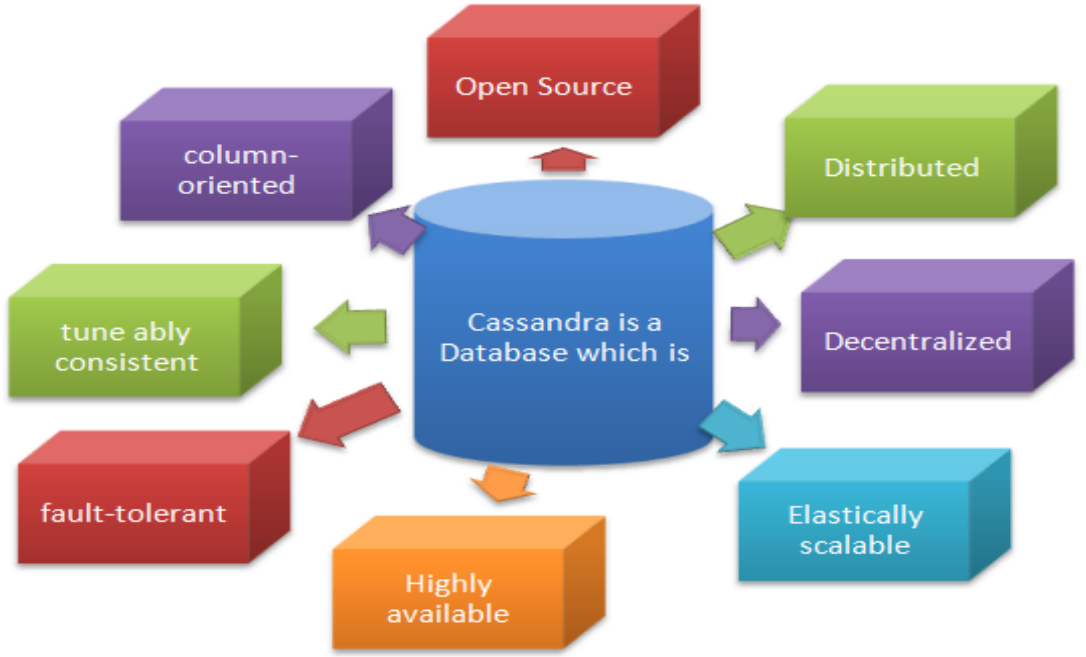
Casandra

What is Casandra

Apache **Cassandra** is a highly scalable, high-performance distributed database designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. It is a type of NoSQL database.

Apache Cassandra™, a top level Apache project born at Facebook and built on Amazon's Dynamo and Google's BigTable, is a distributed database for managing large amounts of structured data across many commodity servers, while providing highly available service and no single point of failure. Apache Cassandra™ offers capabilities that relational databases and other





https://academy.datastax.com/planet-cassandra/what-is-apache-cassandra

Casandra

Features

Cassandra architecture to manage large data volume

Masterless Architecture : means that all nodes are the same and there is no single node that controls other nodes

Cassandra : It is distributed on a number of nodes and it follows *'masterless'* architecture.

Database cluster: Cassandra automatically distributes data across all of the commodity nodes which form the 'ring' known as a database cluster

The redundant data is stored across multiple nodes in the Cassandra ring.

If there is any failure in any node, the same data is retrieved from other nodes having replicated data.

The Cassandra architecture is support for in-built and customizable replication.

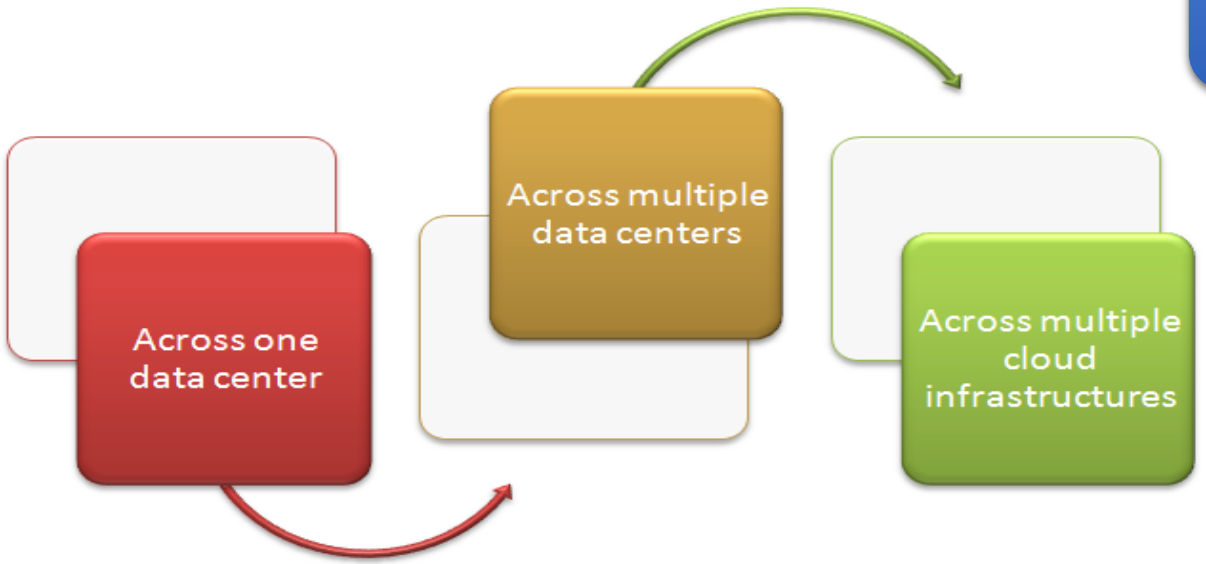
The data is automatically and transparently partitioned on the cluster, developers do not need to do anything programmatically

http://www.devx.com/opensource/managing-large-volumes-of-data-with-apache-cassandra-nosql.html

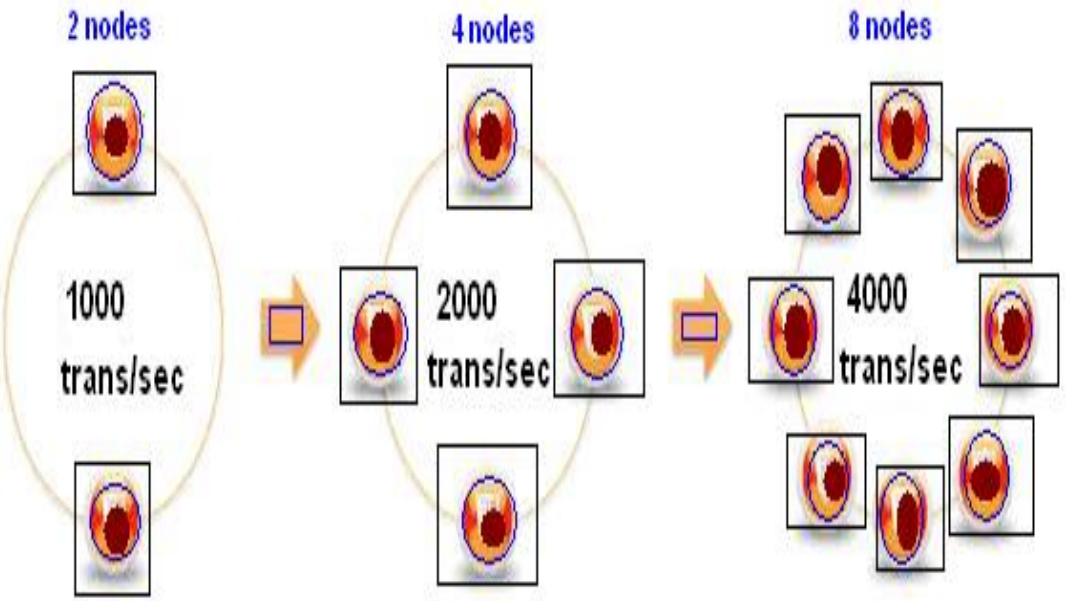
Casandra

Features :Cassandra architecture to manage large data volume

The replication can be configured in the following ways.



For example, if 2 nodes can handle 1000 transactions/sec, then 4 nodes will support 2000 transactions/sec and so on. Following picture shows the linear scalability of a Cassandra ring.



Cassandar ring showing sclability

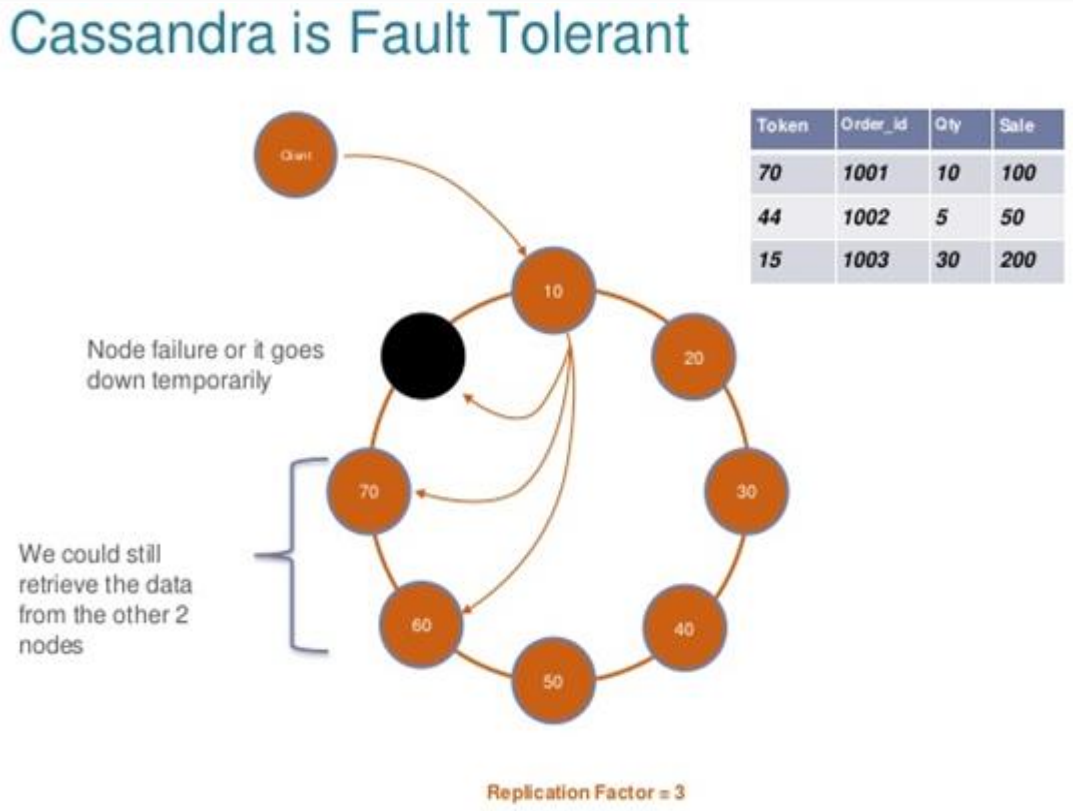
Another architectural feature is the support for linear scalability. It means the capacity or scalability can be increased by simply adding new nodes.

Casandra

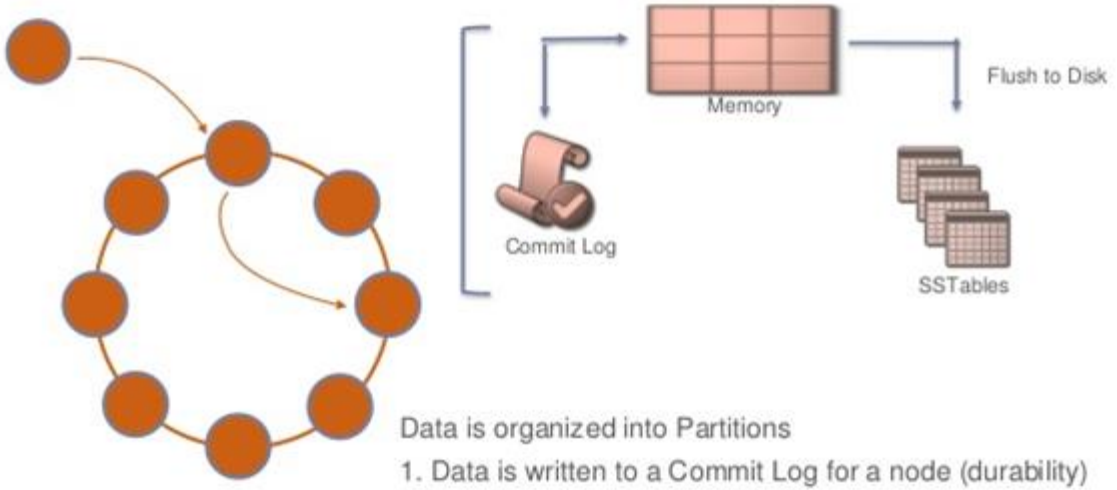
Features

Cassandra is Fault Tolerant

Writes in Cassandra



Token	Order_id	Qty	Sale
70	1001	10	100
44	1002	5	50
15	1003	30	200



Data is organized into Partitions

1. Data is written to a Commit Log for a node (durability)
2. Data is written to MemTable (in memory)
3. MemTables are flushed to disk in an SSTable based on size.

SSTables are immutable

http://www.slideshare.net/carolinedatastax/nyc-java-meetup-datastax-ca

Casandra

Data Replication

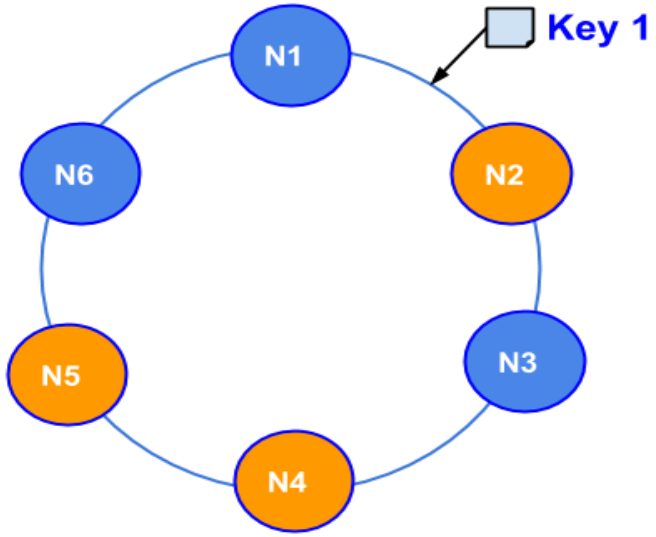
In a distributed system like **Cassandra**, **data replication** enables high availability and durability.

Cassandra replicates rows in a column family on to multiple endpoints based on the **replication** strategy associated to its keyspace.

The endpoints which store a row are called replicas or natural endpoints for that row

NetworkTopologyStrategy with Replication factor
{ DC1: 2, DC2: 2 }

Node	DC	RACK
N1	DC1	RACK2
N3	DC1	RACK1
N6	DC1	RACK1
N2	DC2	RACK1
N4	DC2	RACK1
N5	DC2	RACK2



Key 1 Replicas DC1 : {N3, N1} DC2: {N2, N5}

http://distributeddatastore.blogspot.com/2015/08/cassandra-replication.html

Casandra

Components

Node
- Node is the place where data is stored. It is the basic component of Cassandra.

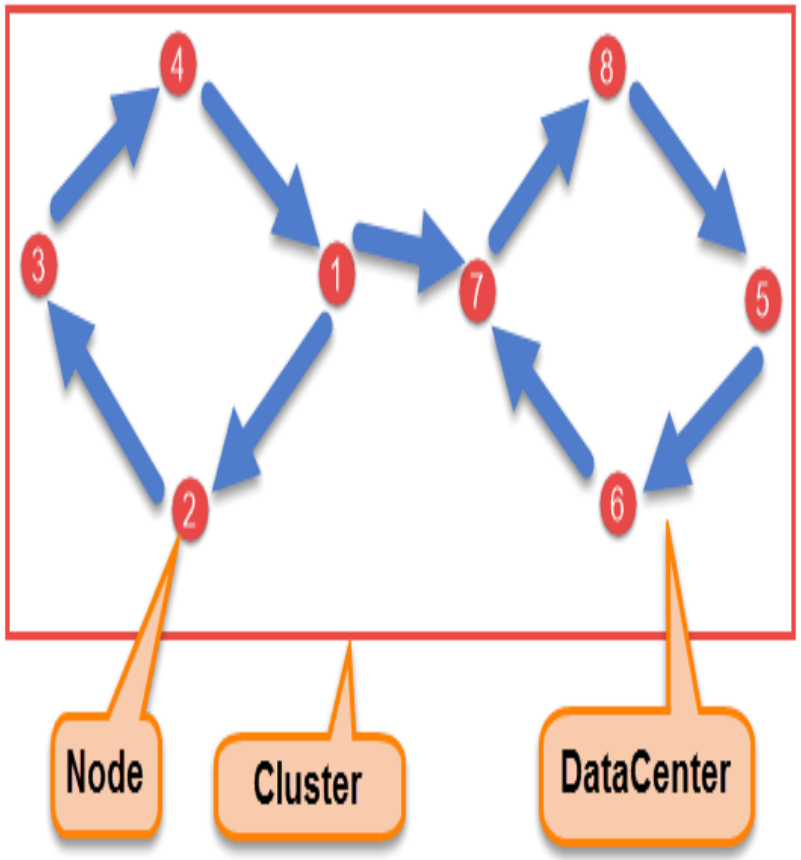
Data Center
- A collection of nodes are called data center. Many nodes are categorized as a data center.

Cluster
- The cluster is the collection of many data centers.

Commit Log
- Every write operation is written to Commit Log. Commit log is used for crash recovery.

Mem-table
- After data written in Commit log, data is written in Mem-table. Data is written in Mem-table temporarily.

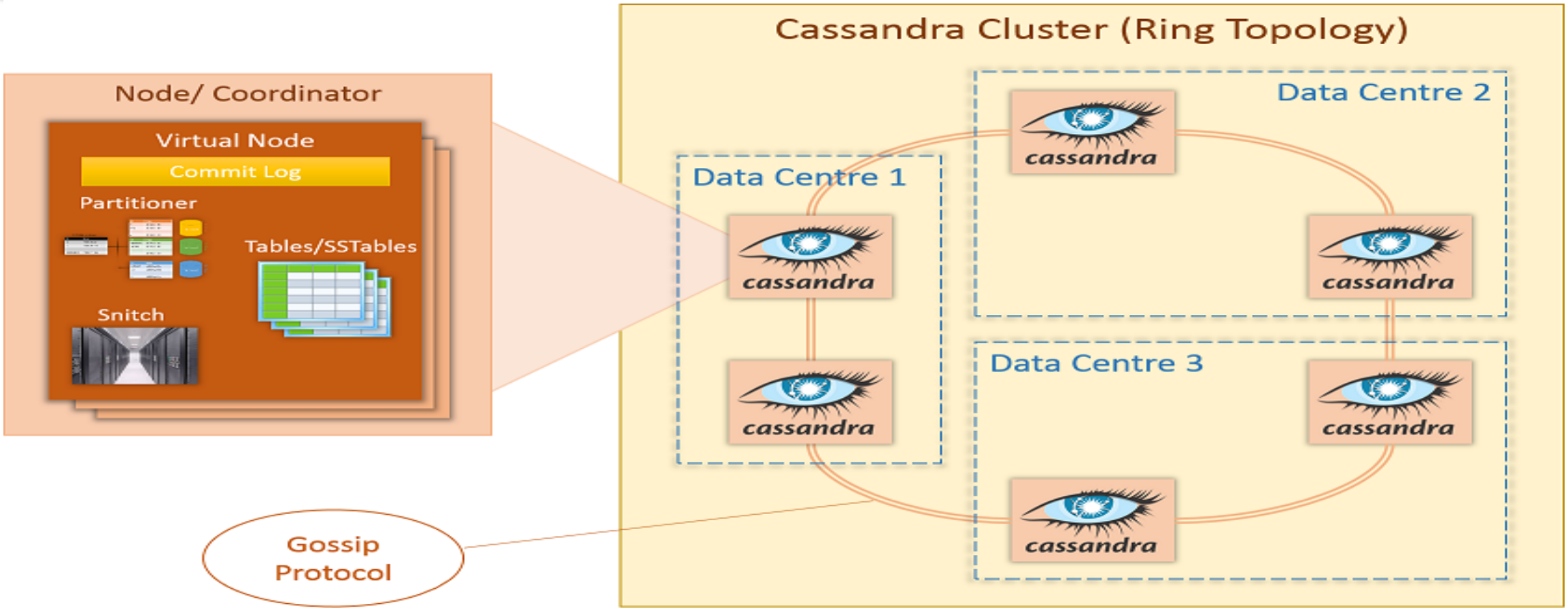
SSTable
- When Mem-table reaches a certain threshold, data is flushed to an SSTable disk file.



http://www.guru99.com/cassandra-architecture.html

Casandra

Components



http://www.allprogrammingtutorials.com/tutorials/getting-started-with-apache-cassandra.php

Casandra

Components

Node/Coordinator -

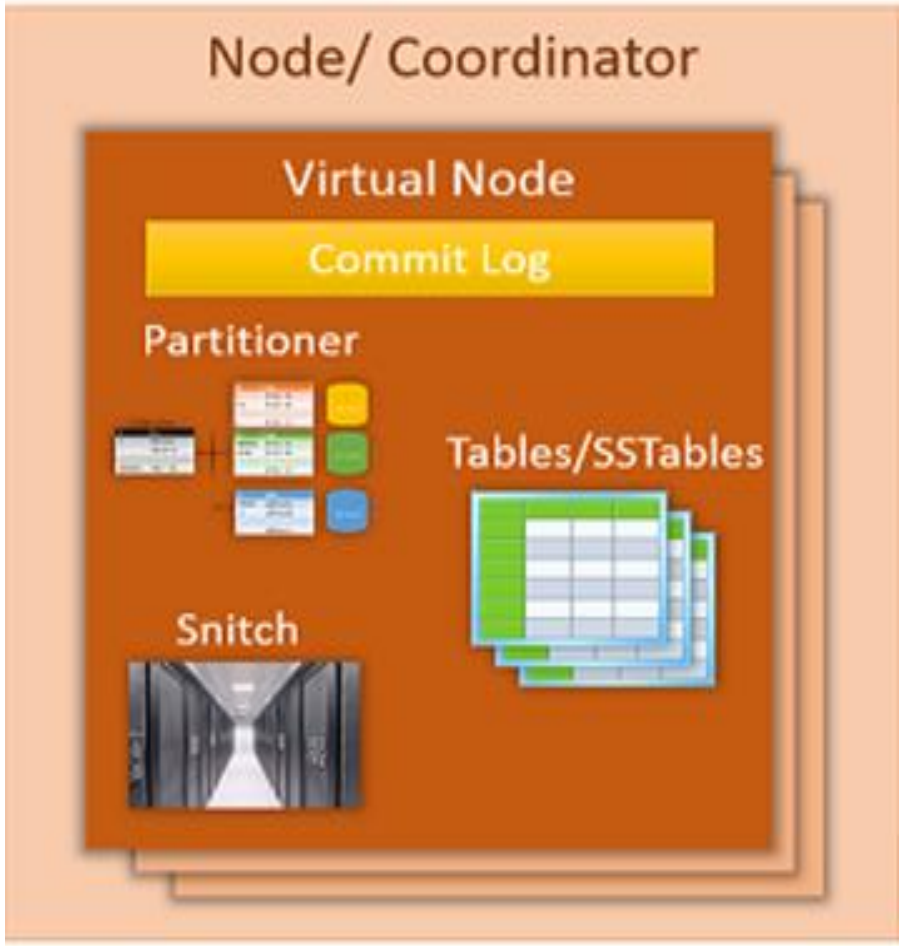
- This refers to a physical machine running Apache Cassandra instance. All the data is stored on these node machines. These nodes also act as coordinator as these pass on the client requirements and state information received from previous node to next node.

Virtual Nodes (Vnodes) -

- Virtual nodes is a recent innovation by Apache Cassandra in order to leverage heterogenous hardware and minimize movement of data when nodes are added and removed from cluster. A Cassandra Node normally has multiple Vnodes and each of these Vnodes is responsible for managing the data for a particular partition key value.

Commit Log -

- This is a temporary log maintained by Cassandra to store all the writes to tables. This ensures that there is no data loss if a machine is restarted due to some unexpected scenarion. Cassandra keeps the updates in memory (memtables) and flushes these to actual tables periodically as immutable records. Once this in-memory table data is flushed, corresponding data from commit log is also removed.



http://www.allprogrammingtutorials.com/tutorials/getting-started-with-apache-cassandra.php

Casandra

Components

Partitioner -

- A partitioner is responsible for assigning the data to particular nodes. It also takes care of assinging other machines for replication of data. Each row of data is uniquely identified by a primary key, that may be same as partition key. A partitioner is basically a hash function that dervies the tokens from partition key of a row. Partitioner uses these tokens to identify nodes as each of the nodes are responsible for managing the data for few token. Nodes manage the data of multiple tokens by assigning token to their virtual nodes. In order to configure this, we need to set num_tokens value while setting up partitioner in cluster.

There are following three types of partitioners in Cassandra:

- This partitioner uses Murmur hash function to derive token from partition key of a row. This partitioner uniformly distributes data across all the nodes of a cluster. This is also default partitioner and works well for almost all the scenarios.

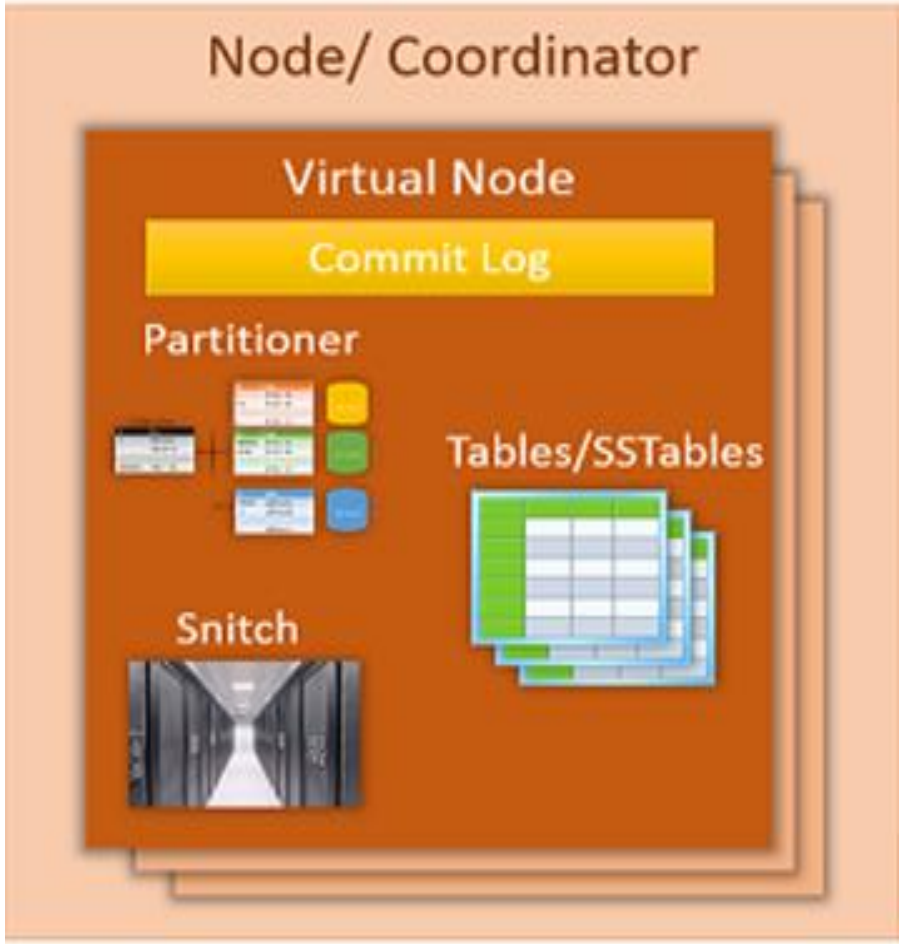
Murmur3Partitioner -

RandomPartitioner -

- This partitioner uses MD5 hash function but is comparatively slower than Murmur hash. Same as Murmur3 partitioner, this partitioner uniformly distributes data across all the nodes of a cluster.

- This partitioner makes it possible to keep an ordered distribution of data lexically by key bytes. However, this is difficult to manage and also does not evenly load balance and dstribute data uniformly in case of multiple tables.

ByteOrderedPartitioner -



http://www.allprogrammingtutorials.com/tutorials/getting-started-with-apache-cassandra.php

Casandra

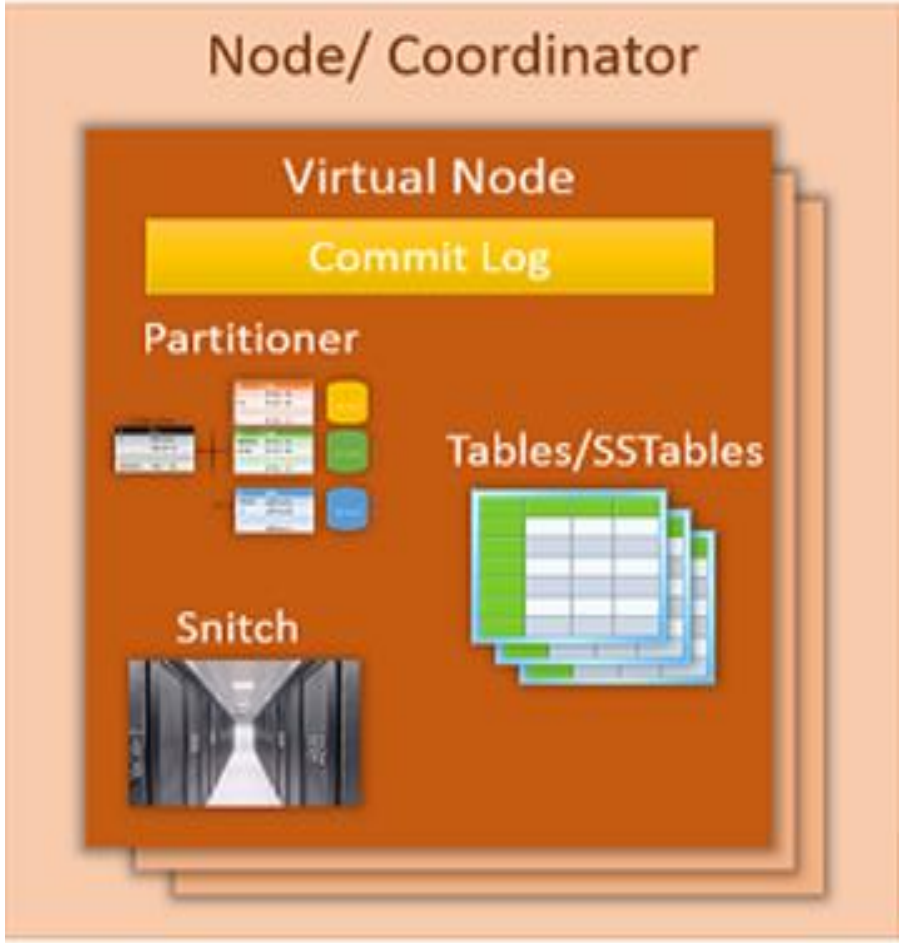
Components

Table and SSTable -

- Table is simply an ordered collection of columns for each row. Row typically consists of columns along with a primary key. Table is more of a logical structure that client works with. On the other hand, SSTable (Sorted String Table) represents data file of immutable records to which Apache Cassandra flushes in-memory tables periodically. SSTables are maintained for each of Cassandra tables.

Snitch -

- Snitch is a mechanism for Cassandra nodes to define which racks and data centers these belong to. This information is quite useful for replication strategy to place relicas in different racks and data centers.



http://www.allprogrammingtutorials.com/tutorials/getting-started-with-apache-cassandra.php

Casandra

Components

Data Center -

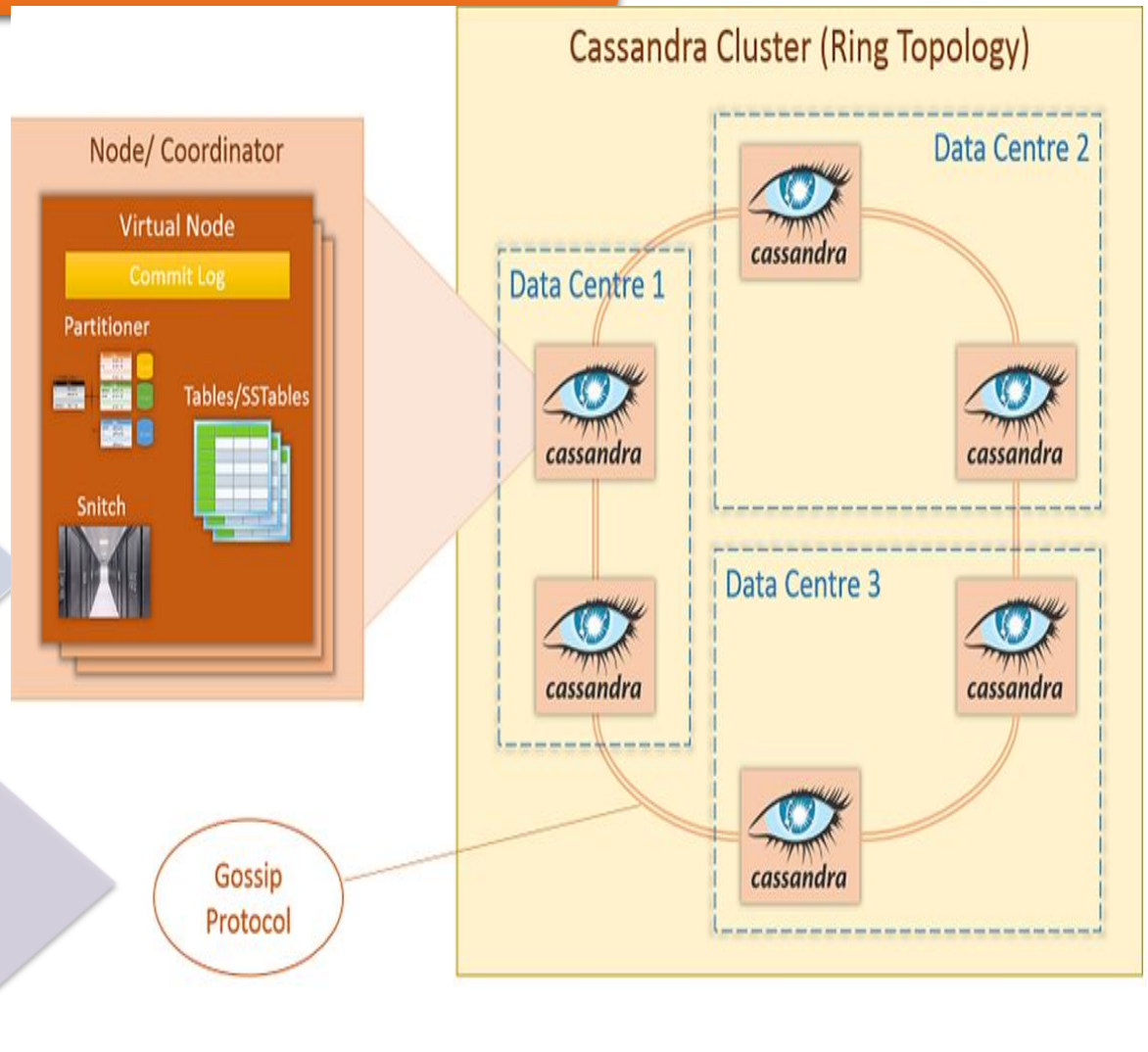
- This refers to physical place where computer and their storage systems are placed. It generally includes redundant or backup power supplies. In Cassandra context, a data center will host a collection of nodes containing the data.

Cluster and Ring Topology

- Cluster is a collection of multple nodes hosted in one or multiple data centers. Apache Cassandra cluster follows Ring topology as all nodes are connected to two other nodes forming a Ring as shown in above diagram.

Gossip -

- This is a protocol used by nodes to communicate rack, data center and state information. As the name suggests, it works like Gossip wherein each node passes this information to next node, eventually making this information available to all nodes in cluster.



Casandra

Query Language

Cassandra Query Language (**CQL**) is a query language for the Cassandra database. The Cassandra Query Language (**CQL**) is the primary language for communicating with the Cassandra database. The most basic way to interact with Cassandra is using the **CQL** shell, cqlsh.

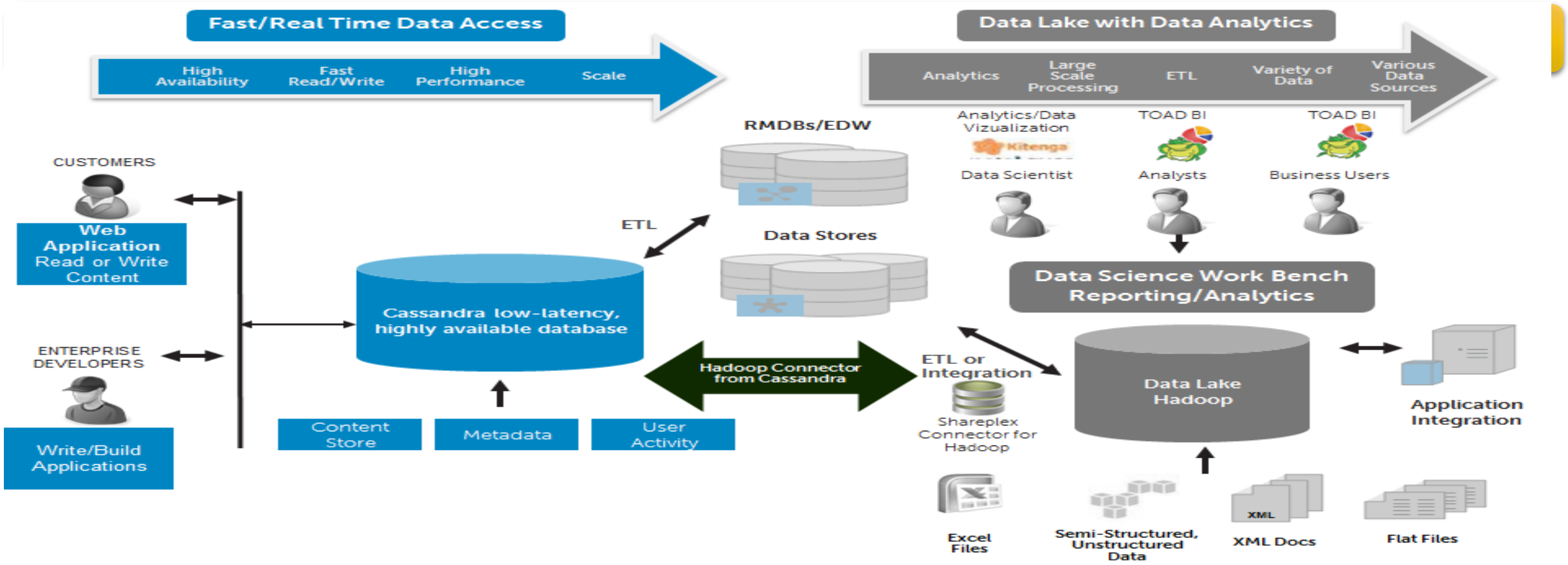
- **CQL: Cassandra Query Language**
- **Very similar to SQL**
- **But restrictions and limitations**
- JOIN requests are forbidden
- No subqueries
- String comparisons are limited (when not using SOLR)
  `select * from my_table where ~~mystring like '%tango%'~~`
- No OR operator
- Can only apply a WHERE condition on an indexed column (or primary key)

```
cqlsh:tutorialspoint> UPDATE ds_emp SET nationality = 'Dutch' where id =3;
cqlsh:tutorialspoint> select * from ds_emp;

 id | city      | name    | nationality | role
----+-----------+---------+-------------+------
  1 |   Utrecht |    Hary |       Greek |   DS
  2 |  Den Haag |   Joost |       Dutch |   DS
  4 | Rotterdam |  Gerard |     Mexican |   DS
  3 |  Den Haag | Rodrigo |       Dutch |   DS
```

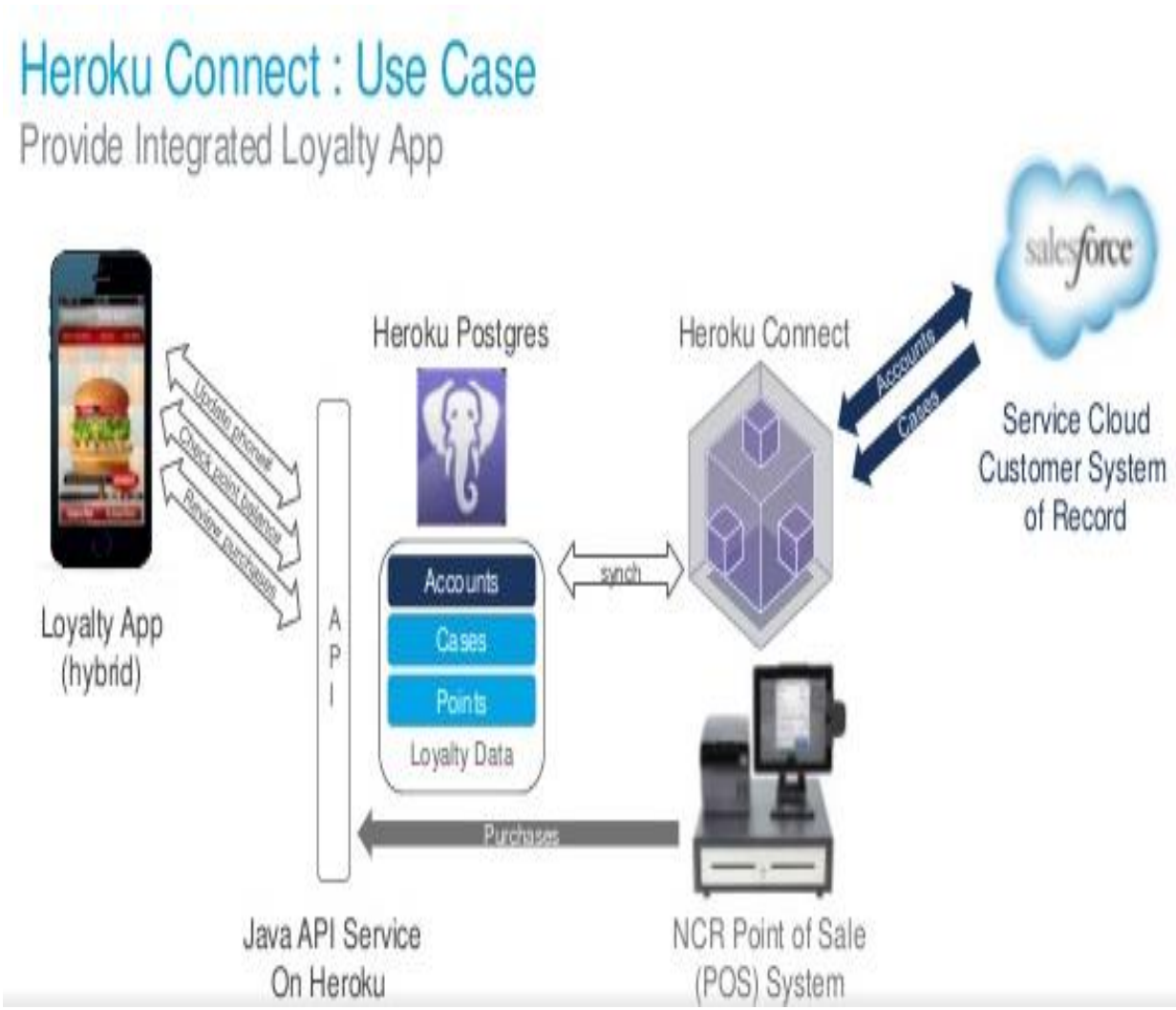
Casandra

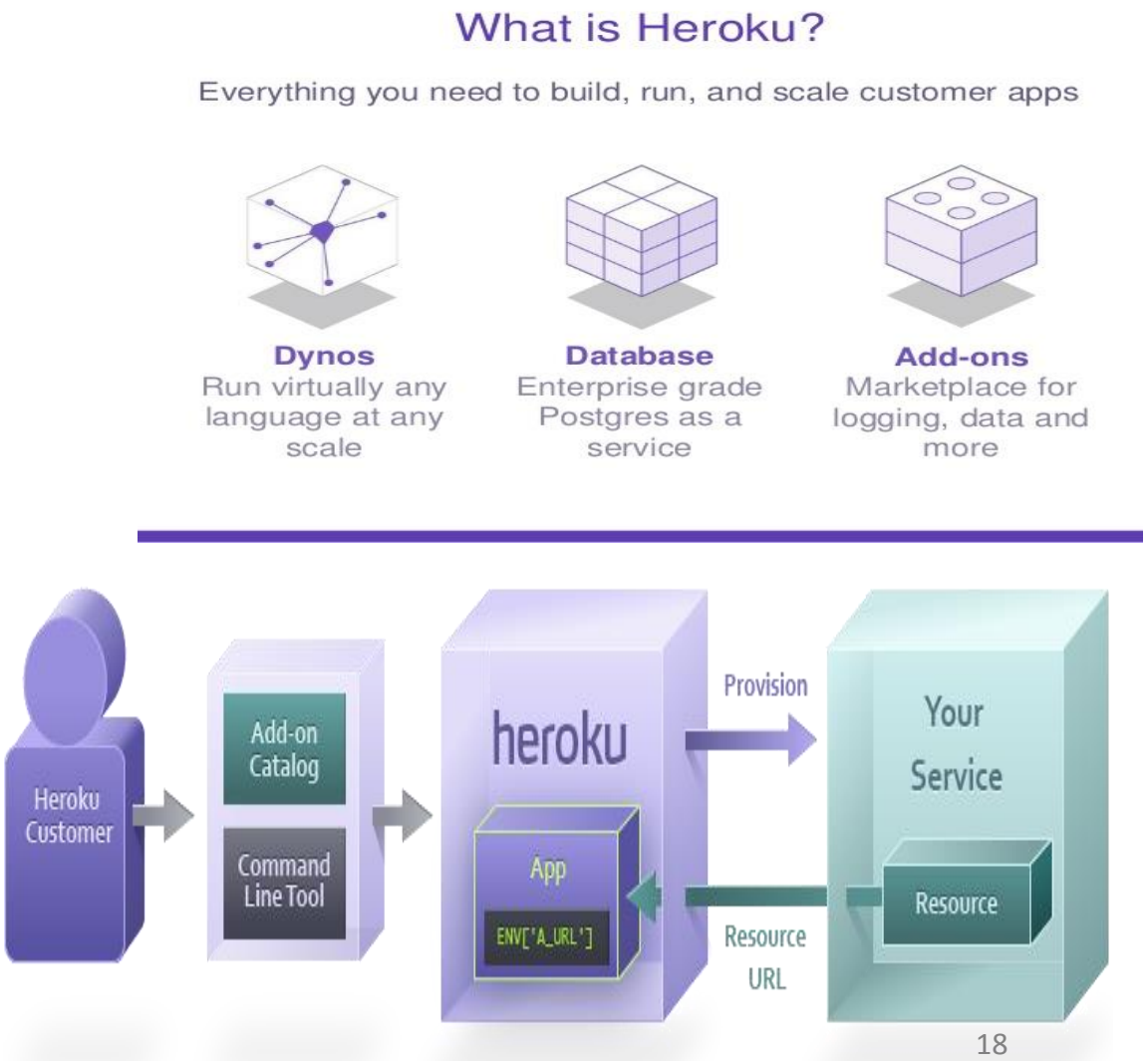
Overall Process



https://aws.amazon.com/ec2/

Heroku


Heroku Connect : Use Case
Provide Integrated Loyalty App
Heroku Postgres
Heroku Connect
salesforce
Accounts
Cases
Service Cloud
Customer System
of Record
Update phone#
Check point balance
Redeem purchases
Loyalty App
(hybrid)
API
Accounts
Cases
Points
Loyalty Data
synch
Purchases
Java API Service
On Heroku
NCR Point of Sale
(POS) System

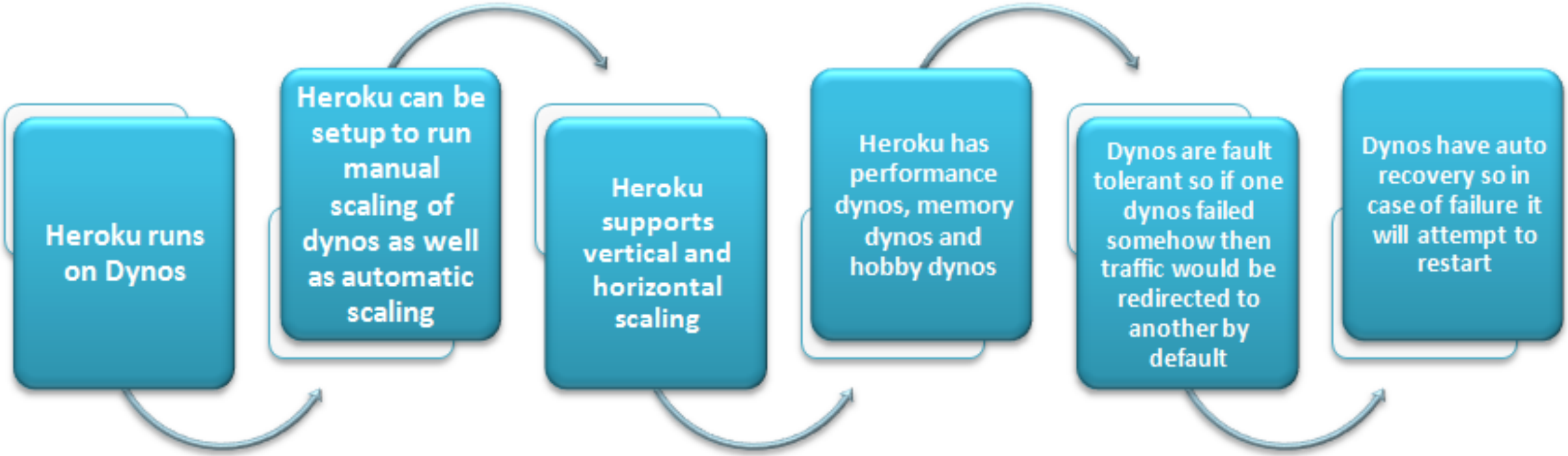

What is Heroku?
Everything you need to build, run, and scale customer apps
Dynos
Run virtually any language at any scale
Database
Enterprise grade Postgres as a service
Add-ons
Marketplace for logging, data and more
Heroku Customer
Add-on Catalog
Command Line Tool
heroku
App
ENV['A_URL']
Provision
Resource URL
Your Service
Resource

Heroku

What is Heroku

Heroku

Scaling



Heroku

Heroku CLI

The way of interaction with Heoku is with Heroku CLI

User will have to download CLI and run commands to deploy their applications

In order to run each command, user has to login first

- Heroku login
- Heroku release:info –a appname
- Heroku rollback – a appname
- Heroku git push origin master

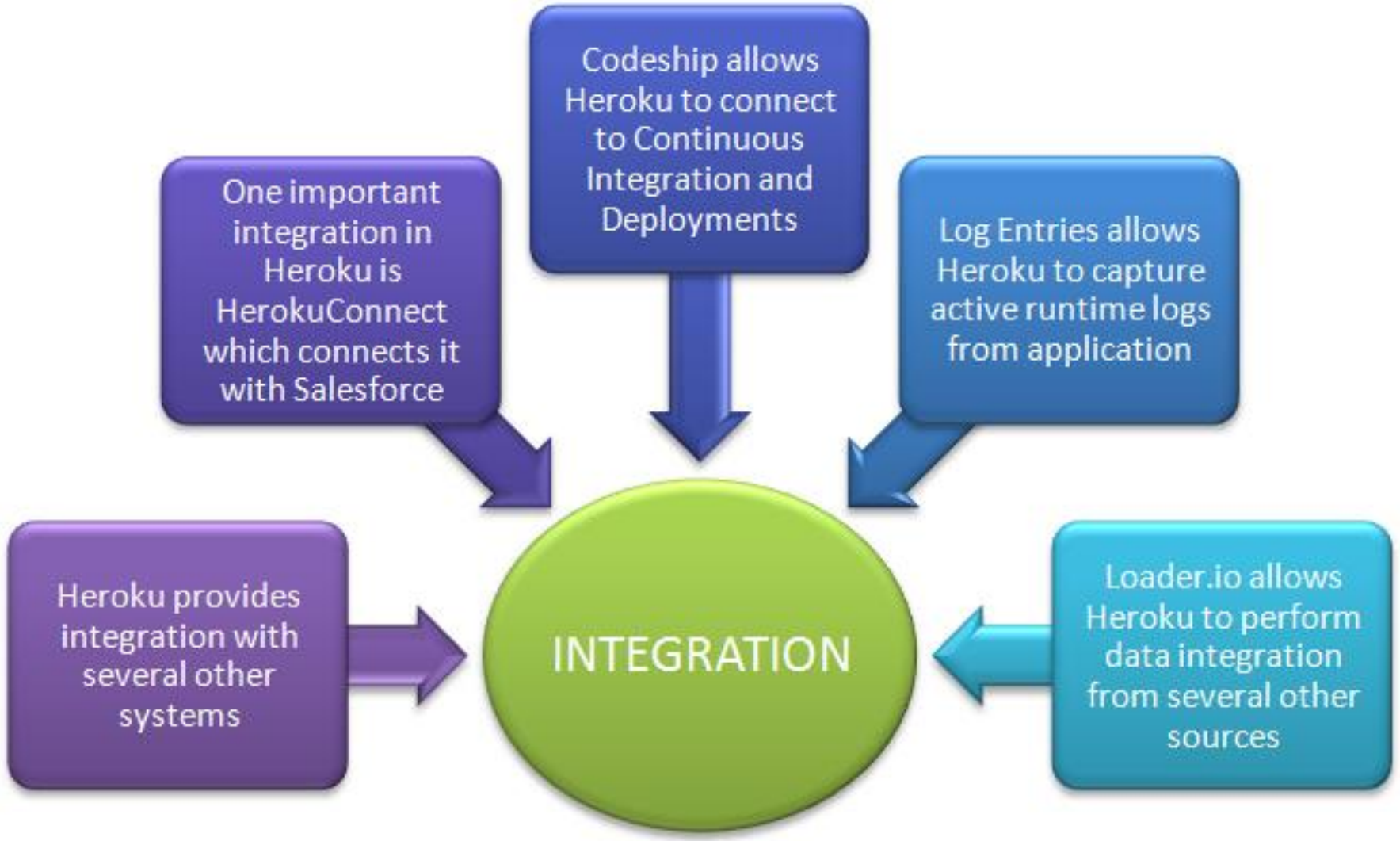
Heroku

Plug-in

There are many plugins available for heroku

Heroku

Integration



Heroku

Metrics and Code deployment

Metrics

Heroku has a popular Metrics dashboard where users can watch and see the current metrics

It contains error information, response times, dyno memory consumptions etc

Metrics also allows Heroku to capture timely stats on how application is behaving

Users can monitor peak times of application

Code deployment

Heroku allows three types of code integration

- Heroku CLI
- Github account
- Dropbox

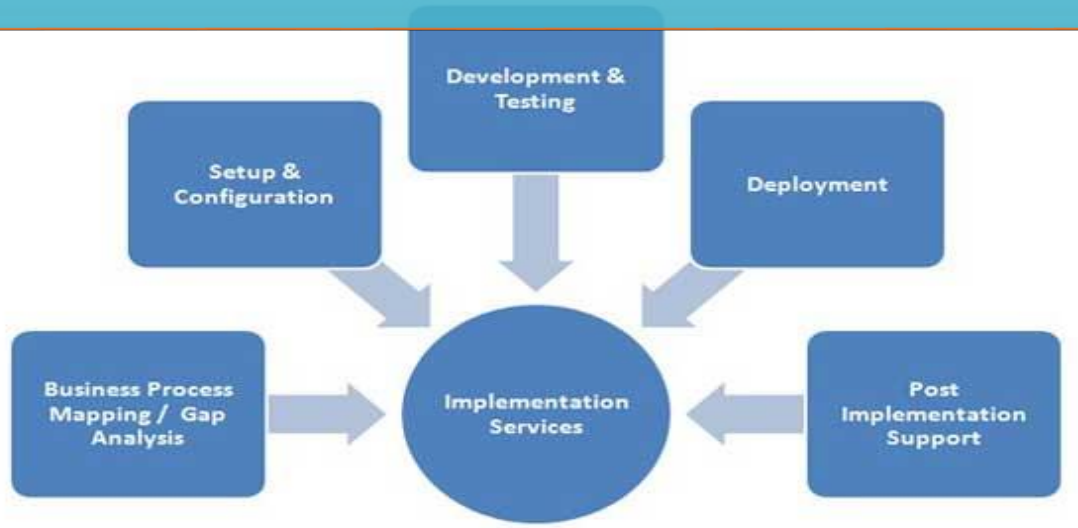
Users can upload code from any of these available code integrations

All the code is maintained in a Git repository managed internally in heroku

Once the code is pushed to repo it is deployed automatically

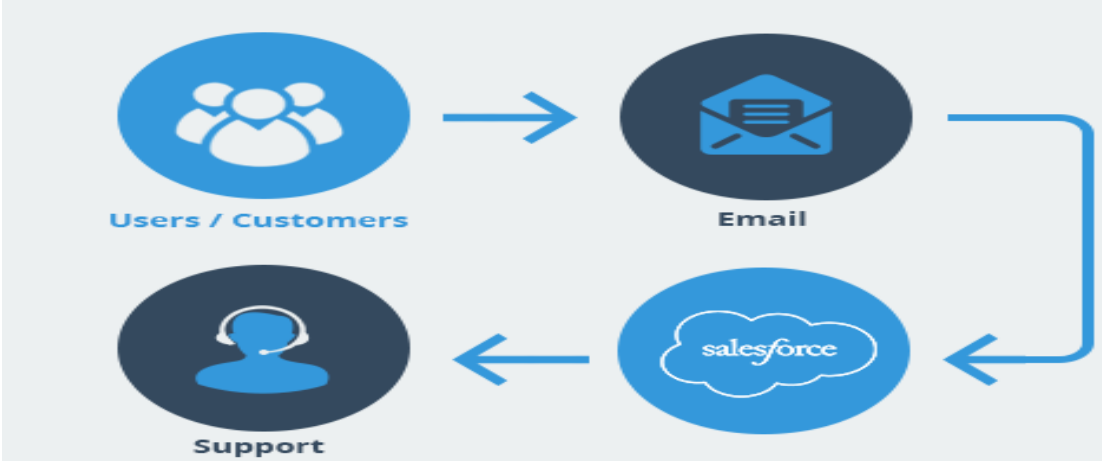
After deployment, users can restart dynos to restart application

Salesforce



salesforce.com

Implementation & Consulting



Salesforce

What is Salesforce?

- Salesforce is the primary enterprise offering within the Salesforce1 Platform.
- It provides companies with an interface for case management and task management, and a system for automatically routing and escalating important events.
- The Salesforce customer portal provides customers the ability to track their own cases, includes a social networking plug-in that enables the user to join the conversation about their company on social networking websites, provides analytical tools and other services including email alert, Google search, and access to customers' entitlement and contracts
- Salesforce has a mobile app and a lightning framework for responsive web design

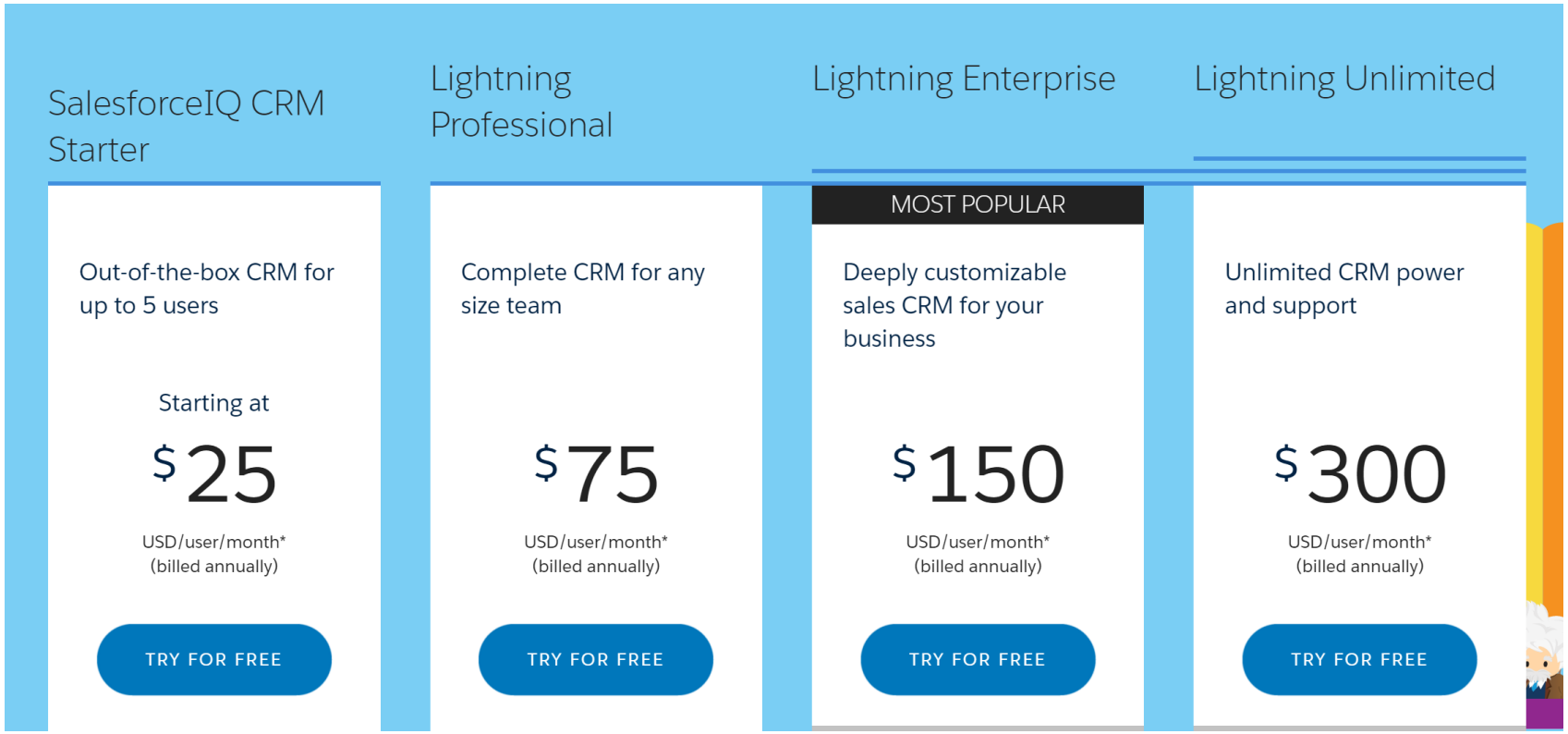
Salesforce

Types of Sandboxes

1. Development Org
2. Testing Org
3. Demo Org
4. Production Org

Salesforce

Types of Licenses



Salesforce

How to create a Custom Object

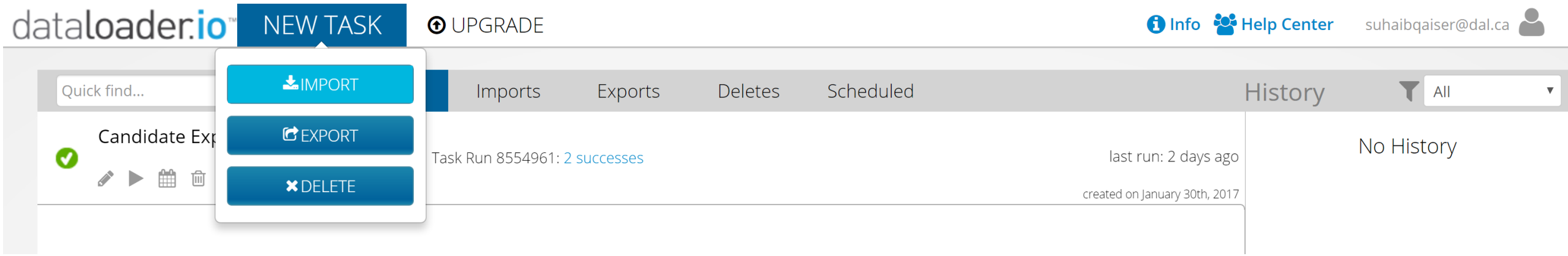
1. Go to Setup
2. Click Create on left side bar
3. Click Create Custom Object
4. Follow the steps in the process. Hit finish in the end

https://aws.amazon.com/ec2/

Salesforce

Data Import / Export

Click http://www.dataloader.io



https://aws.amazon.com/ec2/

Salesforce

Custom Fields

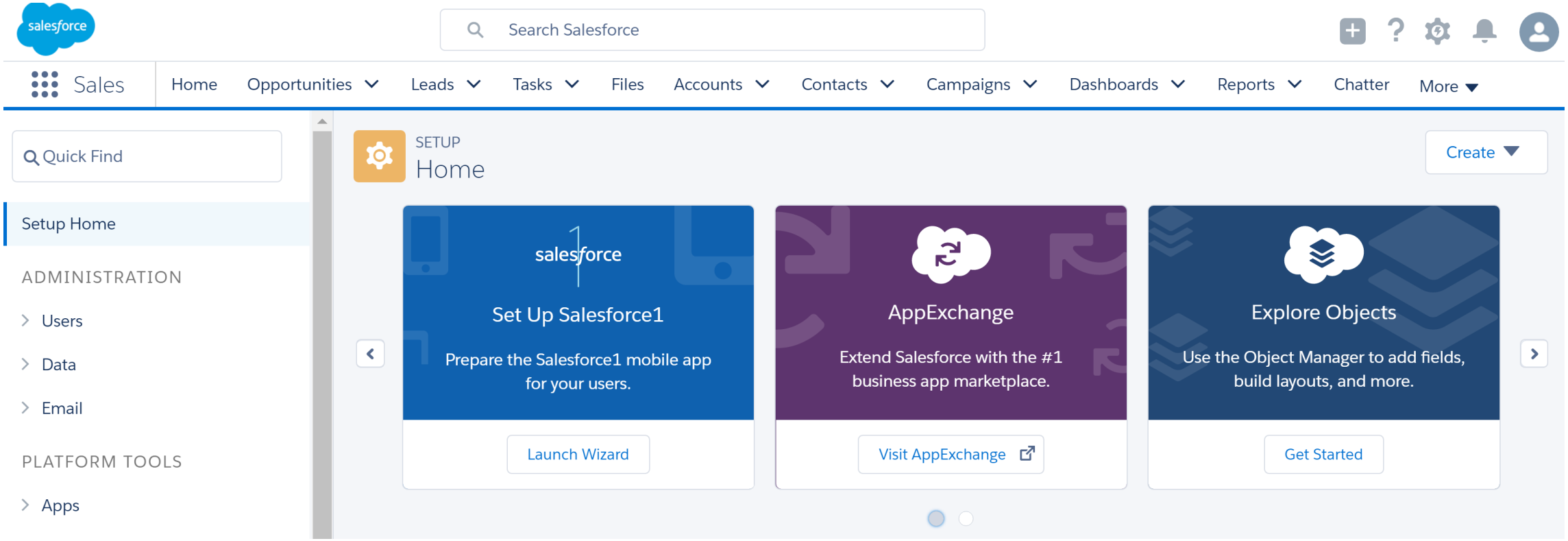
1. Go to Setup
2. Click Create on left side bar
3. Go to Objects and select the custom object
4. Click on Add Field and follow the steps in the process. Hit finish in the end

https://aws.amazon.com/ec2/

Salesforce

Lightning Framework

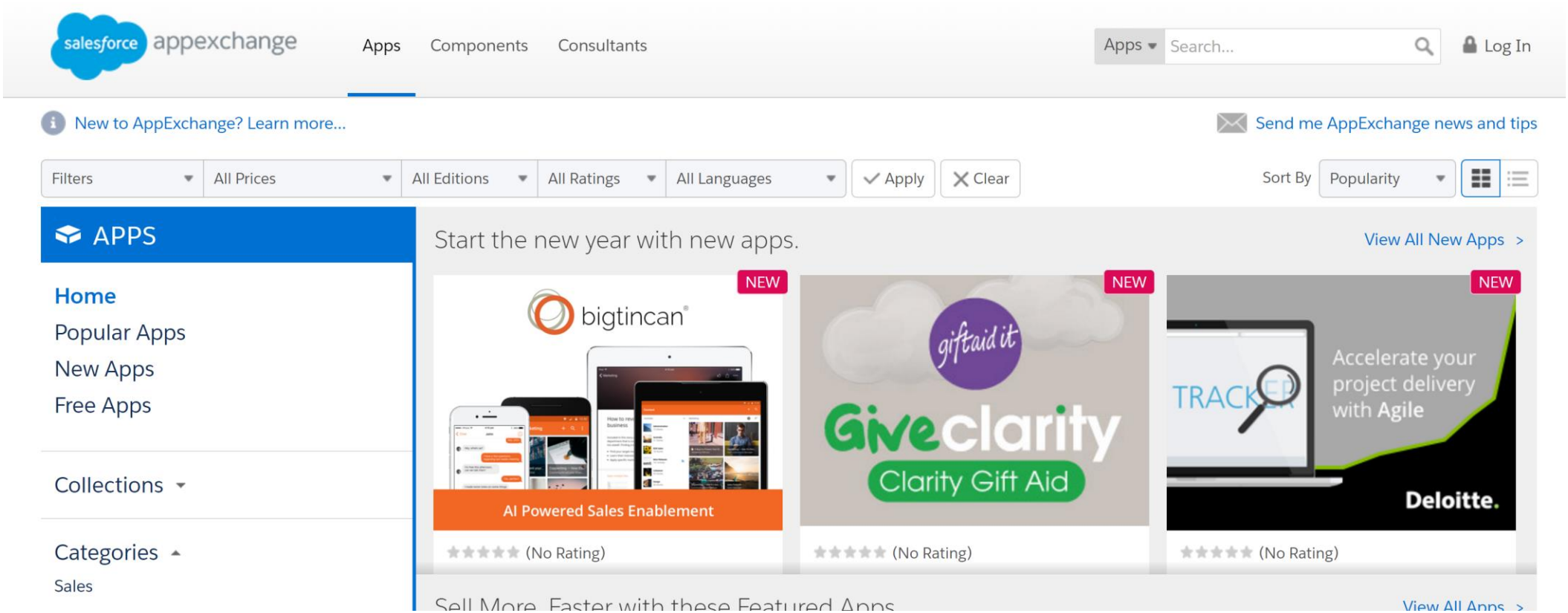
Responsive Web Interface of Salesforce UI. Read this: https://www.lightningdesignsystem.com/



Salesforce

App Exchange

Market place to buy Salesforce Apps: http://appexchange.salesforce.com



Salesforce

Triggers and Pages

1. Open Developer Console in Salesforce
2. Click NEW => Visualforce Pages/Trigger
3. Write APEX code for trigger and VF Page to execute on Salesforce UI
4. To attach VF Page to a custom object: Go to custom object buttons section
5. Click Edit and select Visual force Page that you created earlier

QUIZ

Recap of today and earlier lecture

Q1. How we create a Custom Object and Custom Tab in Salesforce?

Q2. How can we link Code Repository containing source code to Heroku?

Q3. What is Heroku CLI?

Q4. What is Heroku Connect?

Q5. Does Casandra support JOINS? WHY

Q5. Does Casandra support Replication? WHY

Scenario

ABC Insurance company has 300 Sales Agents that are responsible for selling insurance policies to different clients. After collecting data, all Sales rep needs to send application to head office for review. This data contains medical, financial and personal information of users. ABC company is looking for a solution to automate this process. Please suggest a system that can do following:

1. Online Quote

2. Online Applications

3. Cases

4. Claims

5. Accounts

6. Integration with their current financial system

Please suggest what particular licenses, configuration, environments, users, application they would need to purchase/install ?


ANY
QUESTIONS
?