## PART I – Background and Properties

### Four Parts of Cassandra Tutorial

- History and Background
- Installation
- Speed Contest *Vs* MongoDB
- Generate A Speed Layer

### History and Background

- Apache Cassandra - An open source, distributed /decentralized database, provides highly available service with no single point of failure.
- Notable Features:
  - **Elastic scalability** - Cassandra is highly scalable; it allows to add more hardware to accommodate more customers and more data as per requirement.
  - **Always on architecture -** Cassandra has no single point of failure and it is continuously available for business-critical applications that cannot afford a failure.
  - **Fast linear-scale performance** - Cassandra is linearly scalable. Therefore, it maintains a quick response time.
  - **Flexible data storage** - Dynamically accommodate structured, semi-structured, and unstructured according to need.
  - **Easy data distribution** - Cassandra provides the flexibility to distribute data where you need by replicating data across multiple data centers.
  - **Transaction support** - Cassandra supports properties like Atomicity, Consistency, Isolation, and Durability (ACID).
  - **Fast writes** - run on cheap commodity hardware. It performs blazingly fast writes and can store hundreds of terabytes of data, without sacrificing the read efficiency.

### Pros:

- It is scalable, fault-tolerant, and consistent.
- It is a column-oriented database.
- Its distribution design is based on Amazon’s Dynamo and its data model on Google’s Bigtable.
- Created at Facebook, it differs sharply from relational database management systems.
- Cassandra implements a Dynamo-style replication model with no single point of failure, but adds a more powerful “column family” data model.
- Cassandra is being used by some of the biggest companies such as Facebook, Twitter, Cisco, Rackspace, ebay, Twitter, Netflix, and more.

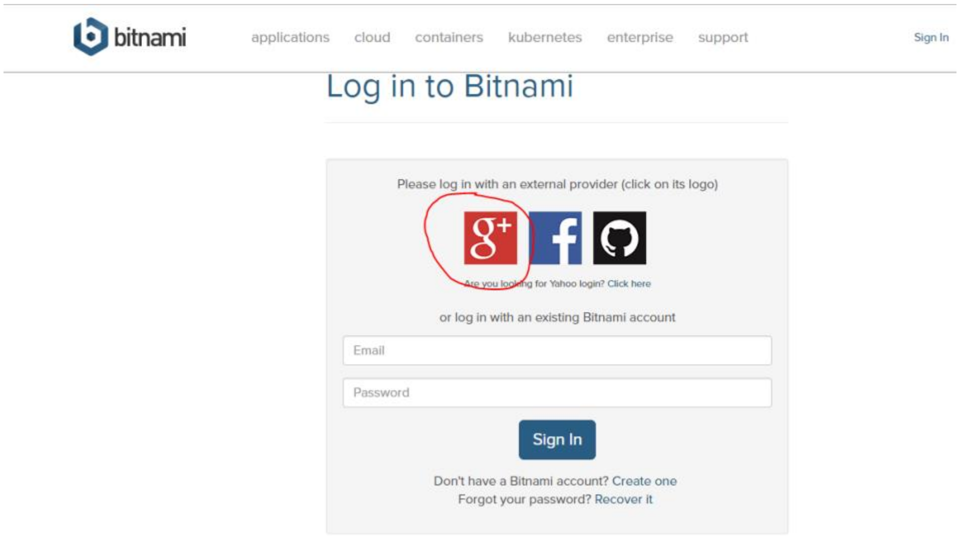
## PART II – Installation

### Cassandra Cloud Set Up Steps

- Create google account

- Open https://cloud.google.com
- Sign up for free trial
- Create a project on the google cloud console
- Open “https://google.bitnami.com” and sign in with Google
- Create a virtual machine

1. Go to the URL https://bitnami.com/sign_in and login with your own google account:



Log in to Bitnami

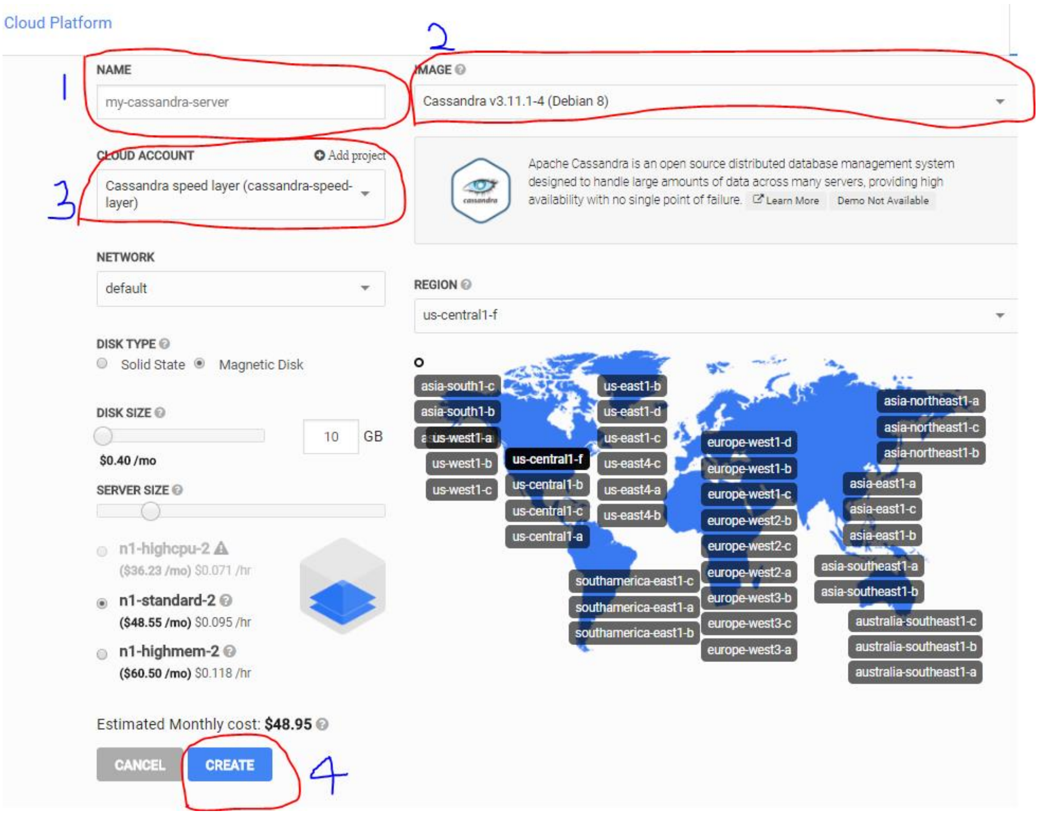
Please log in with an external provider (click on its logo)

Are you looking for Yahoo login? Click here

or log in with an existing Bitnami account

Don't have a Bitnami account? Create one
Forgot your password? Recover it

2. a). Type in the name you want to name your Cassandra vm (Step 1)
   b). Select Cassandra from the list of virtual machine instances (Step 2)
   c). Select the google cloud project (project we just created) (Step 3)
   d) Select create to create the project (Step 4)



NAME
my-cassandra-server

IMAGE
Cassandra v3.11.1-4 (Debian 8)

CLOUD ACCOUNT ⊕ Add project
Cassandra speed layer (cassandra-speed-layer)

Apache Cassandra is an open source distributed database management system designed to handle large amounts of data across many servers, providing high availability with no single point of failure. Learn More Demo Not Available

NETWORK
default

REGION
us-central1-f

DISK TYPE
○ Solid State ● Magnetic Disk

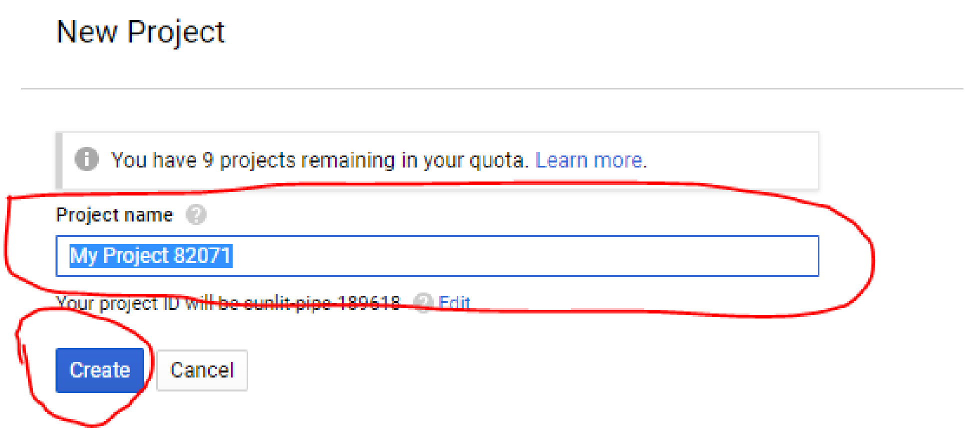
DISK SIZE
10 GB
$0.40 /mo

SERVER SIZE

○ n1-highcpu-2 ($36.23 /mo) $0.071 /hr
● n1-standard-2 ($48.55 /mo) $0.095 /hr
○ n1-highmem-2 ($60.50 /mo) $0.118 /hr

Estimated Monthly cost: **$48.95**

3. The Cloud will give a default test cluster, you can also create your own clusters and name it as your personal need:

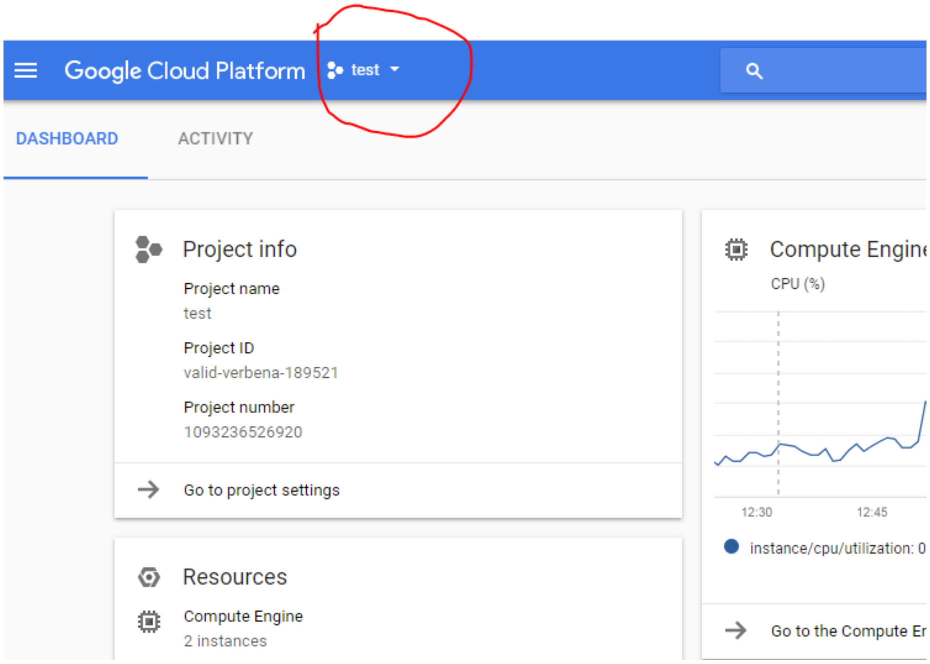


New Project

You have 9 projects remaining in your quota. Learn more.

Project name: My Project 82071

Your project ID will be sunlit-pipe-189618 Edit



Project info

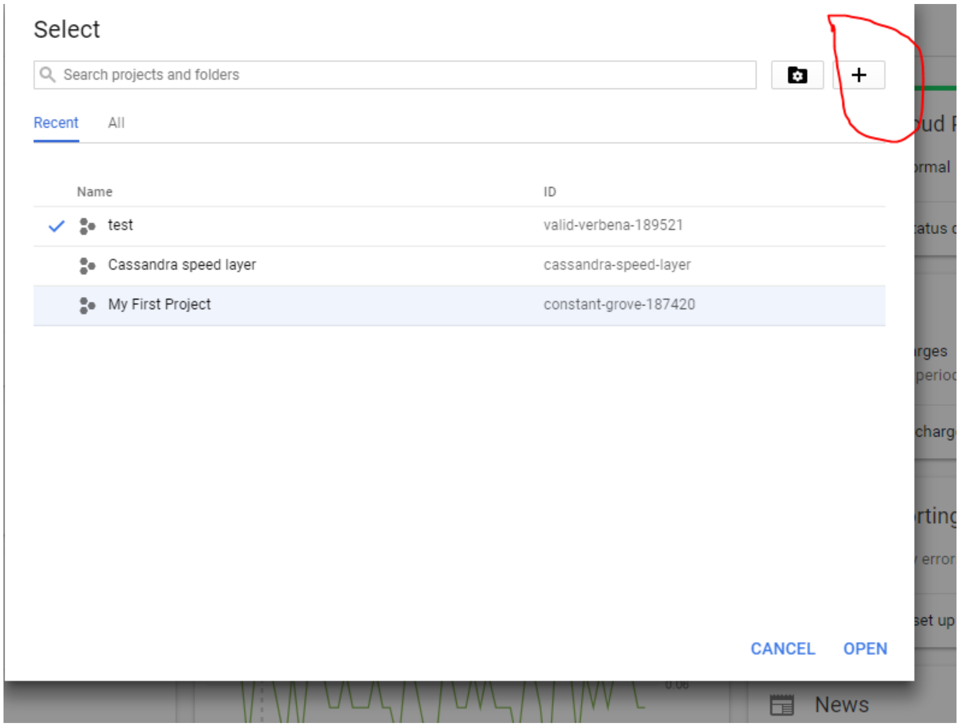
Project name: test

Project ID: valid-verbena-189521

Project number: 1093236526920

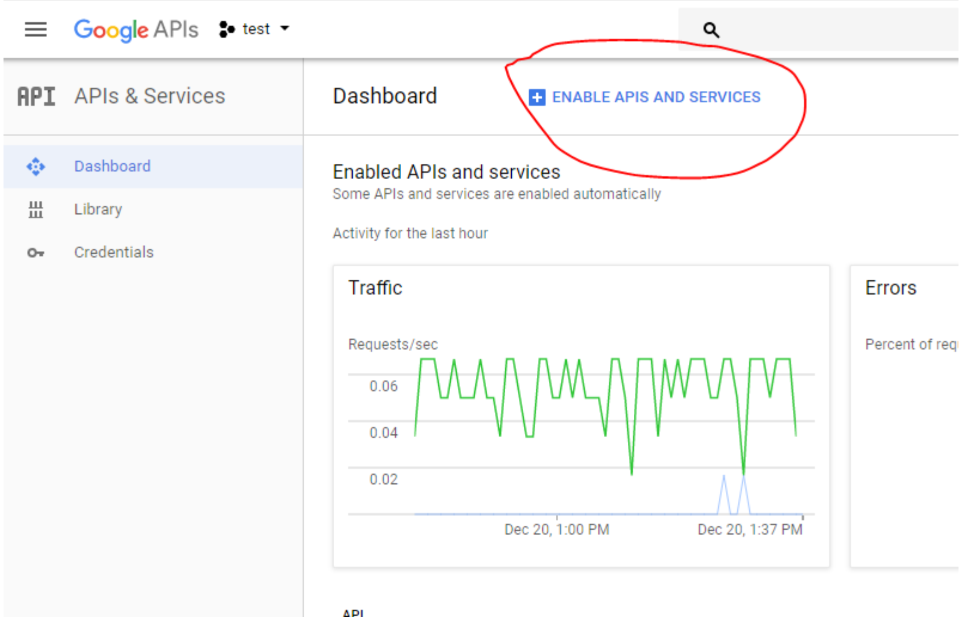
Resources

Compute Engine: 2 instances



| | Name | ID |
|---|---|---|
| ✓ | test | valid-verbena-189521 |
| | Cassandra speed layer | cassandra-speed-layer |
| | My First Project | constant-grove-187420 |

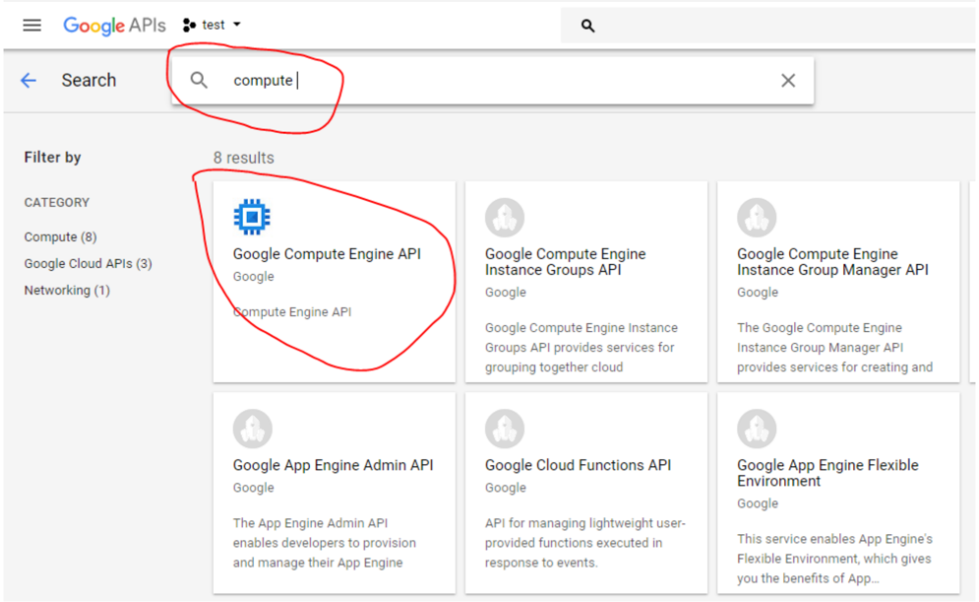
4. Enable APIs and Services as following steps:



Enabled APIs and services

Some APIs and services are enabled automatically

Activity for the last hour



Filter by

CATEGORY

Compute (8)

Google Cloud APIs (3)

Networking (1)

8 results

Google Compute Engine API
Google
Compute Engine API

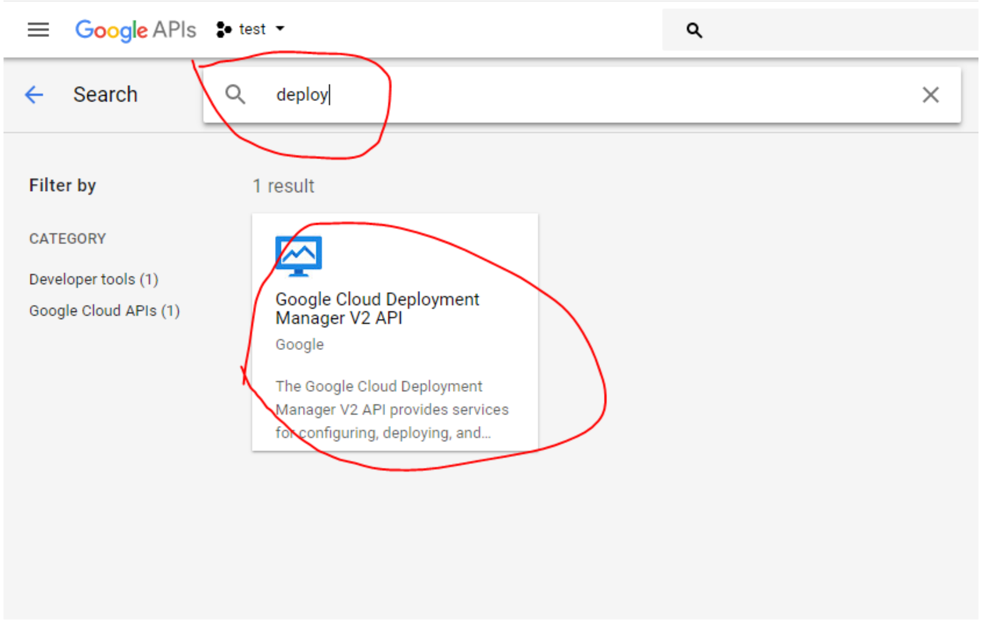
Google Compute Engine Instance Groups API
Google
Google Compute Engine Instance Groups API provides services for grouping together cloud

Google Compute Engine Instance Group Manager API
Google
The Google Compute Engine Instance Group Manager API provides services for creating and

Google App Engine Admin API
Google
The App Engine Admin API enables developers to provision and manage their App Engine

Google Cloud Functions API
Google
API for managing lightweight user-provided functions executed in response to events.

Google App Engine Flexible Environment
Google
This service enables App Engine's Flexible Environment, which gives you the benefits of App...



Filter by

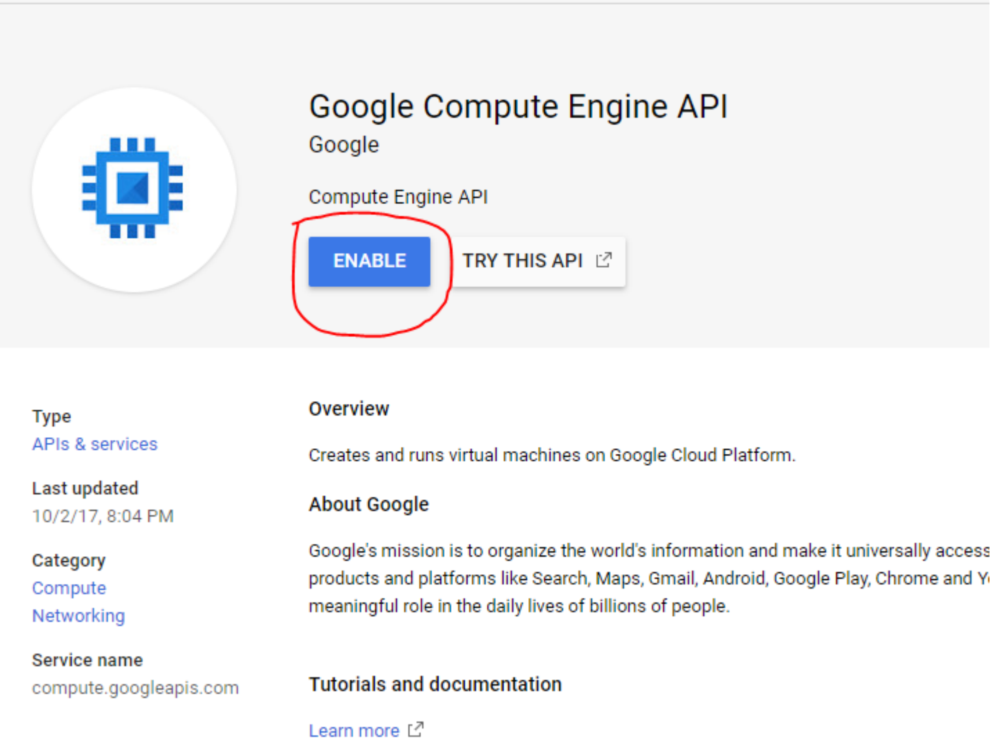
CATEGORY

Developer tools (1)

Google Cloud APIs (1)

1 result

Google Cloud Deployment Manager V2 API
Google
The Google Cloud Deployment Manager V2 API provides services for configuring, deploying, and...



Google Compute Engine API

Google

Compute Engine API

Type

APIs & services

Last updated

10/2/17, 8:04 PM

Category

Compute

Networking

Service name

compute.googleapis.com

Overview

Creates and runs virtual machines on Google Cloud Platform.

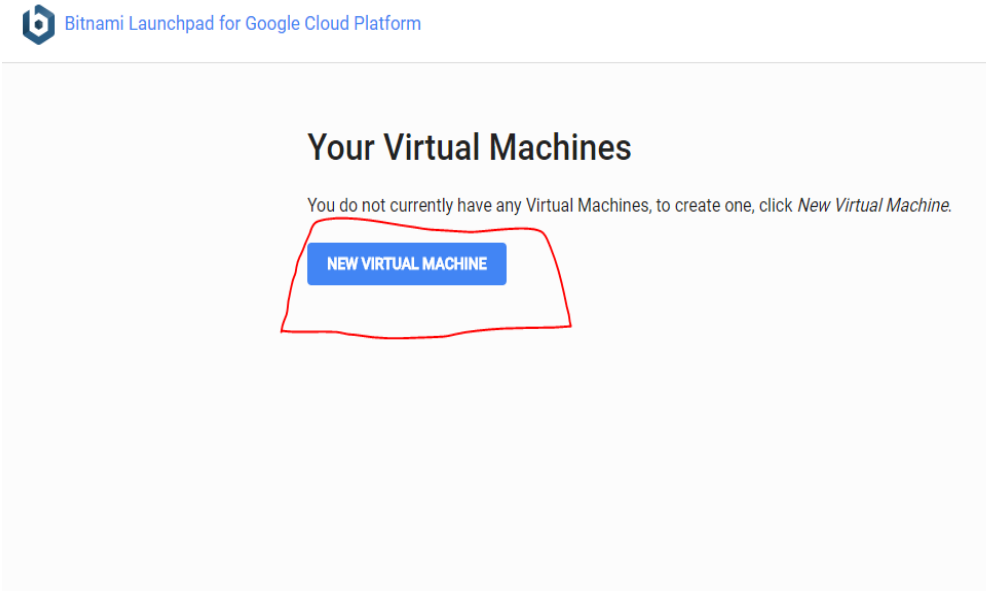
About Google

Google's mission is to organize the world's information and make it universally access products and platforms like Search, Maps, Gmail, Android, Google Play, Chrome and Y meaningful role in the daily lives of billions of people.

Tutorials and documentation

Learn more

5. Once logged in accept the license agreement. Follow the image step below:



Bitnami Launchpad for Google Cloud Platform

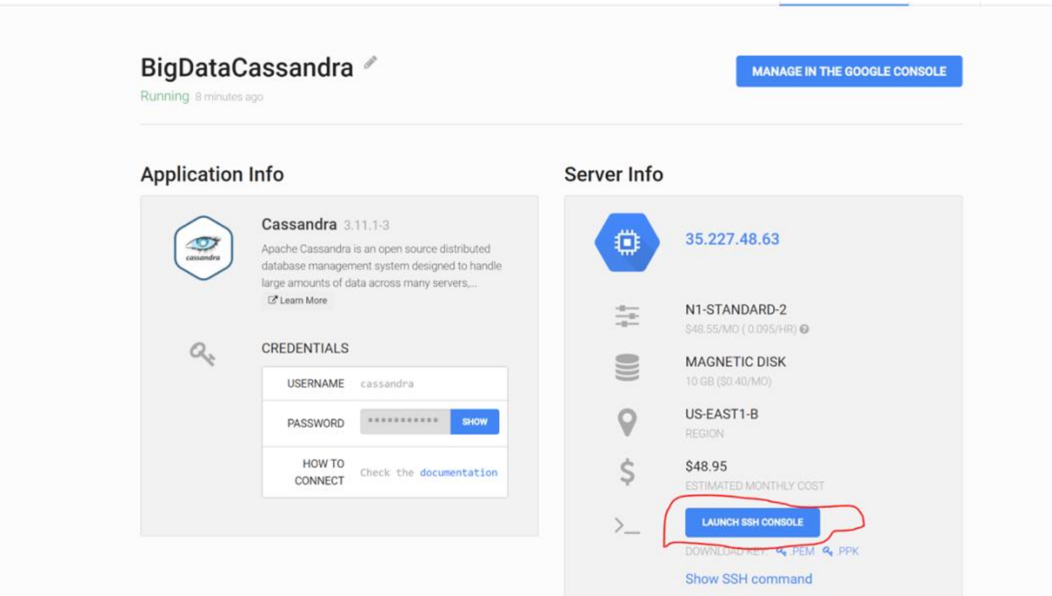
Your Virtual Machines

You do not currently have any Virtual Machines, to create one, click *New Virtual Machine.*

6. Multiple node cluster:

   1). Go to the bitnami console using "https://google.bitnami.com

2). Sign in and select virtual machines from the top right of the screen
3). Select the cassandra virtual machine you have setup and the screen below will show up\



BigDataCassandra

Running 8 minutes ago

Application Info

Cassandra 3.11.1-3

Apache Cassandra is an open source distributed database management system designed to handle large amounts of data across many servers,...

CREDENTIALS

| | |
|---|---|
| USERNAME | cassandra |
| PASSWORD | ************ |
| HOW TO CONNECT | Check the documentation |

Server Info

35.227.48.63

N1-STANDARD-2
$48.55/MO ( 0.095/HR)

MAGNETIC DISK
10 GB ($0.40/MO)

US-EAST1-B
REGION

$48.95
ESTIMATED MONTHLY COST

DOWNLOAD KEY: .PEM .PPK

Show SSH command

6. In the online console, follow these steps:
   1). Download the personal key from Bitnami.
   2). Open local terminal, change permission to public by type in:

```
chmod 600 bitnami-google-cassandra-speed-layer.pem.
```

   3). Run:    ssh -i bitnami-google-cassandra-speed-layer.pem
   4). Connect to cluster:

```
cqlsh -u cassandra 23.251.158.183 -p N4pKBV8AjKiU
```

   5). You can change password to yours:

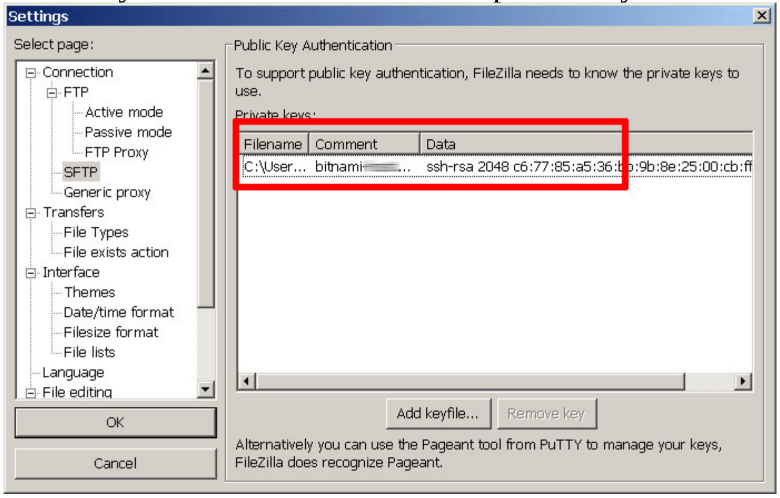
```
cqlsh> ALTER USER cassandra with PASSWORD 'NEWPASSWORD';
```

   6). Restart, all set!

**Get connected to the FileZilla (**To upload local files to the server with SFTP.)

- Download and install FileZilla, link: https://filezilla-project.org/
- Launch FileZilla and use the "Edit -> Settings" command to bring up FileZilla's configuration settings.

- Use the "Add keyfile" command to select the private key file for the server:



Public Key Authentication

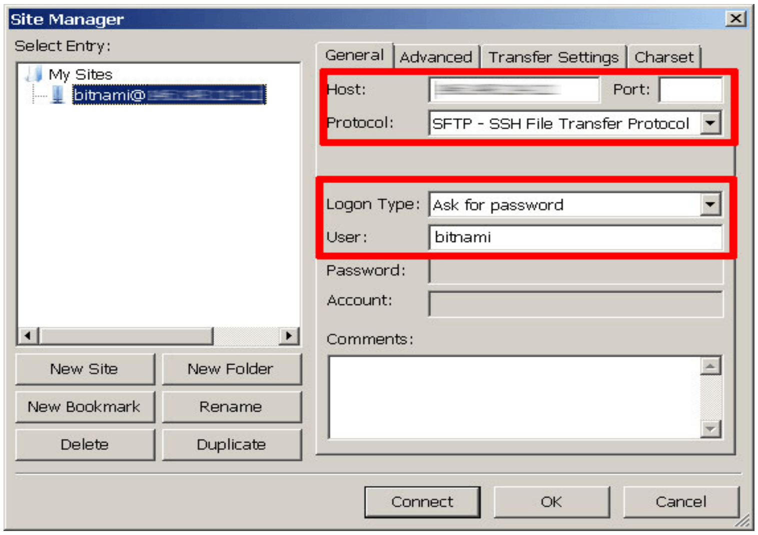
To support public key authentication, FileZilla needs to know the private keys to use.

Private keys:

| Filename | Comment | Data |
|---|---|---|
| C:\User... | bitnami-[redacted]... | ssh-rsa 2048 c6:77:85:a5:36:b:9b:8e:25:00:cb:ff |

Alternatively you can use the Pageant tool from PuTTY to manage your keys, FileZilla does recognize Pageant.

- Use the "File -> Site Manager -> New Site" command to bring up the FileZilla Site Manager.
- Enter your server host name and specify *bitnami* as the user name.
- Select "SFTP" as the protocol and "Ask for password" as the logon type.
- Use the "Connect" button to connect to the server and begin an SFTP session. You might need to accept the server key, by clicking "Yes" or "OK" to proceed.



bitnami@[redacted]

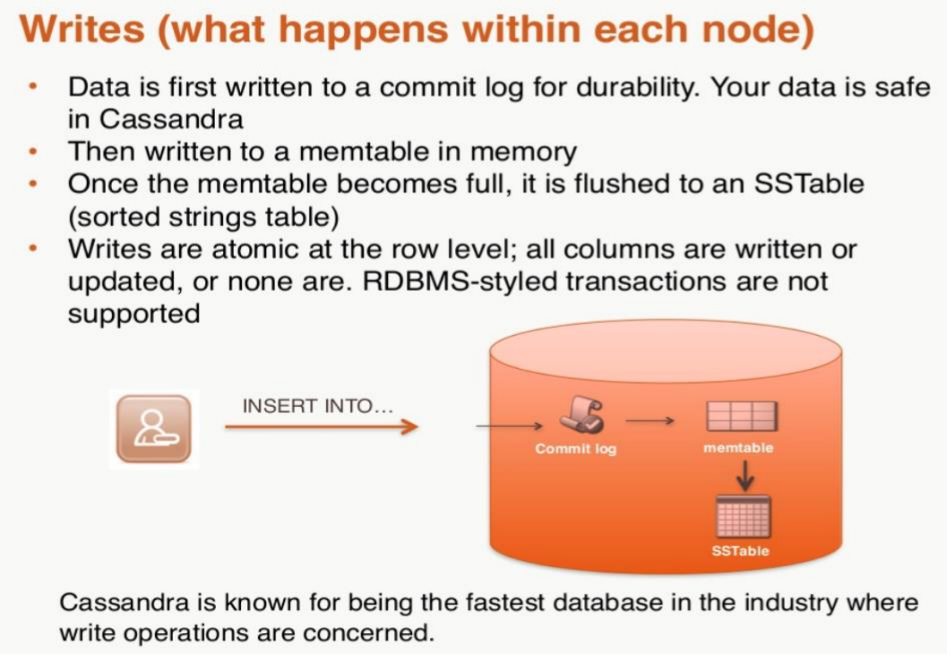
Host: [redacted]
Port:
Protocol: SFTP - SSH File Transfer Protocol
Logon Type: Ask for password
User: bitnami
Password:
Account:
Comments:

## Generate Speed Views

Nodes and Clusters

- Every node is a peer
- Ring structure (not master/slave)
- Tokens and hashing

- Elastic scaling (remove or add node from (to) runnung cluster,   upgrade when it's running, …)



## Writes (what happens within each node)

- Data is first written to a commit log for durability. Your data is safe in Cassandra
- Then written to a memtable in memory
- Once the memtable becomes full, it is flushed to an SSTable (sorted strings table)
- Writes are atomic at the row level; all columns are written or updated, or none are. RDBMS-styled transactions are not supported

INSERT INTO…
Commit log
memtable
SSTable

Cassandra is known for being the fastest database in the industry where write operations are concerned.

**Consistency**

- Every node acts as a coordinator
- Tunable consistency levels
- Strong consistence (R + W > N)
- Eventually consistence (R + W < N)

**Cassandra Data Model**

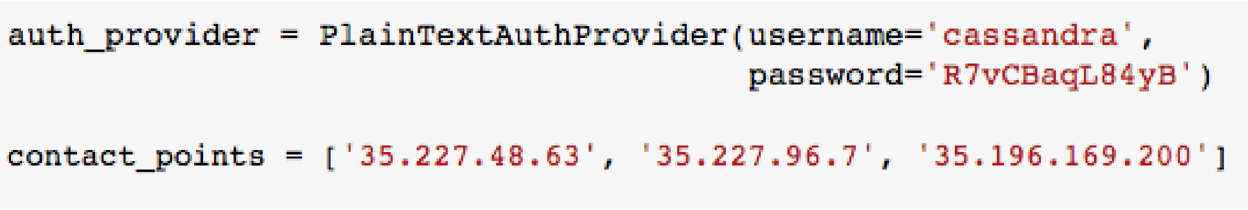
- Query language (CQL) looks like SQL
- Data model is tabular (like relational database)
  - Tall, but narrow (6-10 columns)
  - Scalable → can keep millions or billions of rows
- Differences:
  - Cassandra do not do JOIN tables
  - Denormalize a lot (at scale we do not JOIN, it slows the read down)
  - Keyspace is namespace (container) for tables

**Cassandra Keys**

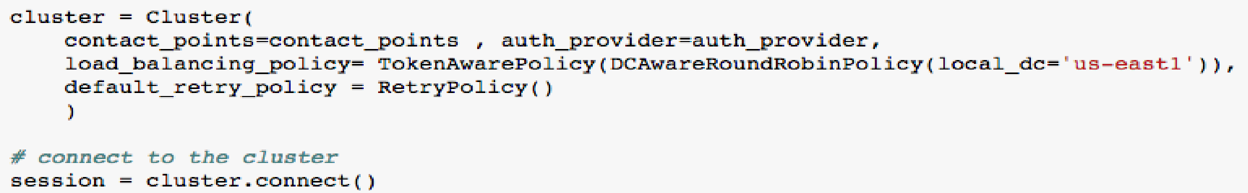
- Record ID:
  - Primary key:
  - Uuid:
    - surrogate primary key
    - why not just integer?
- Keys:

- Primary keys uniquely identify rows (like RB)
- Each key has 2 parts:
  - Partition key: group of rows in the table is guaranteed is located in the same node
  - Clustering key: provide ordering to the rows in the table
- You need to put a column in a key so that it can participate in the query

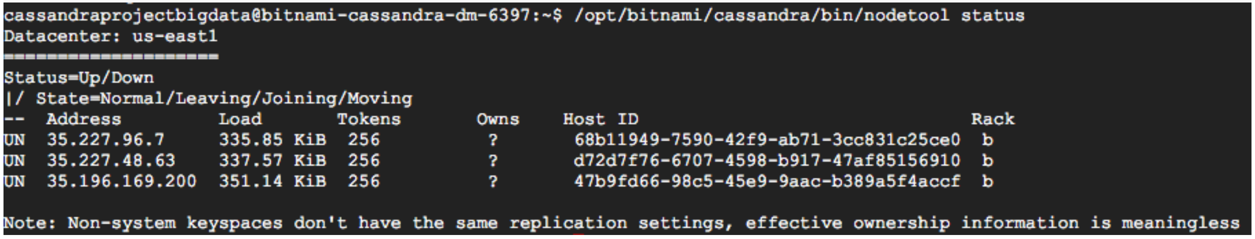
**Connected to Clusters:**



```
auth_provider = PlainTextAuthProvider(username='cassandra',
                                      password='R7vCBaqL84yB')

contact_points = ['35.227.48.63', '35.227.96.7', '35.196.169.200']
```

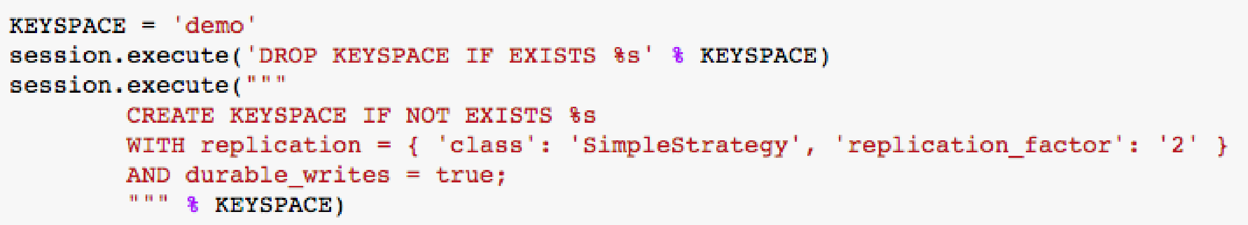


```
cluster = Cluster(
    contact_points=contact_points , auth_provider=auth_provider,
    load_balancing_policy= TokenAwarePolicy(DCAwareRoundRobinPolicy(local_dc='us-east1')),
    default_retry_policy = RetryPolicy()
    )

# connect to the cluster
session = cluster.connect()
```

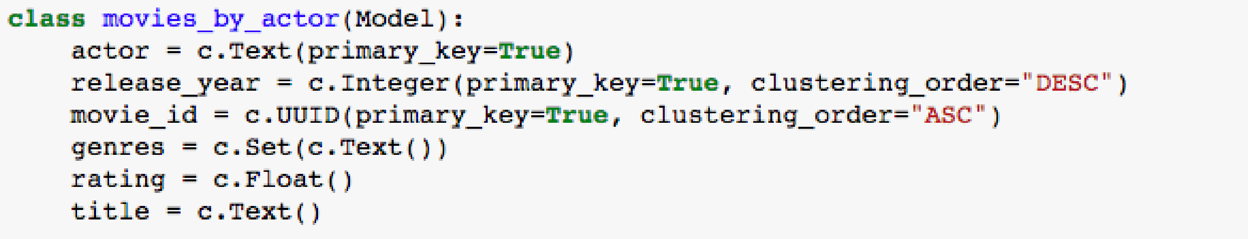


```
cassandraprojectbigdata@bitnami-cassandra-dm-6397:~$ /opt/bitnami/cassandra/bin/nodetool status
Datacenter: us-east1
====================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address         Load       Tokens       Owns    Host ID                               Rack
UN  35.227.96.7     335.85 KiB  256          ?       68b11949-7590-42f9-ab71-3cc831c25ce0  b
UN  35.227.48.63    337.57 KiB  256          ?       d72d7f76-6707-4598-b917-47af85156910  b
UN  35.196.169.200  351.14 KiB  256          ?       47b9fd66-98c5-45e9-9aac-b389a5f4accf  b

Note: Non-system keyspaces don't have the same replication settings, effective ownership information is meaningless
```

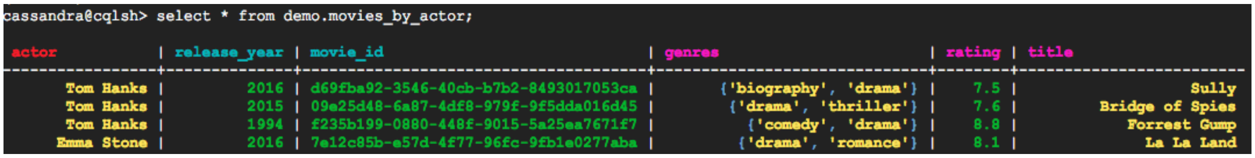
**To Create Tables:**



```
KEYSPACE = 'demo'
session.execute('DROP KEYSPACE IF EXISTS %s' % KEYSPACE)
session.execute("""
        CREATE KEYSPACE IF NOT EXISTS %s
        WITH replication = { 'class': 'SimpleStrategy', 'replication_factor': '2' }
        AND durable_writes = true;
        """ % KEYSPACE)
```



```
class movies_by_actor(Model):
    actor = c.Text(primary_key=True)
    release_year = c.Integer(primary_key=True, clustering_order="DESC")
    movie_id = c.UUID(primary_key=True, clustering_order="ASC")
    genres = c.Set(c.Text())
    rating = c.Float()
    title = c.Text()
```



```
cassandra@cqlsh> select * from demo.movies_by_actor;

 actor           | release_year | movie_id                             | genres                       | rating | title
-----------------+--------------+--------------------------------------+------------------------------+--------+------------------------
       Tom Hanks |         2016 | d69fba92-3546-40cb-b7b2-8493017053ca |       {'biography', 'drama'} |    7.5 |                   Sully
       Tom Hanks |         2015 | 09e25d48-6a87-4df8-979f-9f5dda016d45 |        {'drama', 'thriller'} |    7.6 |         Bridge of Spies
       Tom Hanks |         1994 | f235b199-0880-448f-9015-5a25ea7671f7 |          {'comedy', 'drama'} |    8.8 |            Forrest Gump
      Emma Stone |         2016 | 7e12c85b-e57d-4f77-96fc-9fb1e0277aba |         {'drama', 'romance'} |    8.1 |              La La Land
```

### Speed Test: Cassandra vs MongoDB

- According to the Cassandra website (Apache, 2016): “Cassandra consistently outperforms popular NoSQL alternatives in benchmarks and real applications”
- Datastax (Datastax, 2017): “For mixed operational and analytic workloads typical to modern Web, Mobile and IOT applications, Cassandra performed six times faster than HBase and 195 times faster than MongoDB.”

### Test: Cassandra vs MongoDB – Procedure

- Cassandra and MongoDB node (one each) set up on Google Cloud Platform
- Three trials: In each, create 1000 pretend movies (fictitious films that have the same fields as that from the movie database) and insert them into the respective database.
- Trials differ in the amount of times we repeat the process (1, 100, 1000).
- Statistics collected displaying the time (in seconds)
  - Total time for # of inserts
  - Average time for # of inserts

### Speed Test: Cassandra vs MongoDB – Procedure

- Cassandra and MongoDB node (one each) set up on Google Cloud Platform
- Two trials: Create and insert 100 or 1000 movies.
- Statistics collected displaying the time (in seconds)
- Each also had some specific tests (1 movie insert for MongoDB, adding movies by actor for Cassandra; see GitHub for full results)

### References:

1. Apache, 2016. http://cassandra.apache.org/
2. Datastax,  2017. https://www.datastax.com/apache-cassandra-leads-nosql-benchmark
3. Arunkumar U, June 2017. https://medium.com/@arun_74827/what-is-apache-cassandra-what-are-the-features-of-it-a4b26b860d07
4. Bitnami Docs, https://docs.bitnami.com/google/infrastructure/cassandra/#description