

# Introduction to



# with PyMongo

# Content

- Who am I ?
- Introduction to databases
- Introduction to MongoDB
- Workshop with PyMongo
- Practical exercise

# Who am I ?

Love technology and passionate to implement it

Product manager background

Learning programming and IT architectures since May'17

Started with Python in Nov '17

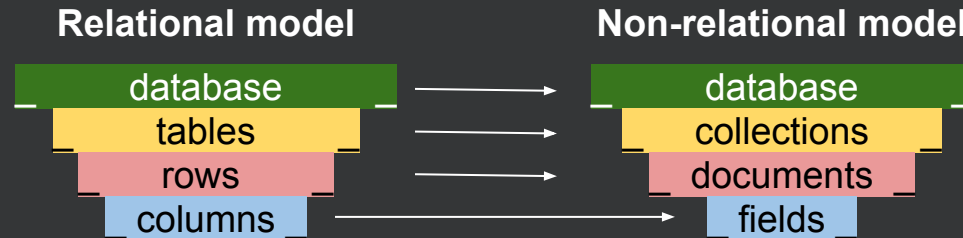
# Introduction to databases

**Database** is an organized collection of data, generally stored and accessed electronically from a computer. From wikipedia

## Database models

**Relational model** organized in tables with rows and columns, with relations between them.

**Non-relational model** organized in collections with documents and fields ( key : value ).



# SQL vs No-SQL comparison

## **SQL** (Structured Query Language)

- Data uses Schema
- Relational
- Data is distributed across multiple tables
- Vertical scaling is possible
- Horizontal scaling is difficult
- Limitations for lots of read write queries per second
- MySQL, SQLite, MS SQL, Oracle, ..

## **No SQL**

- Schema-less
- No relational oriented (few)
- Data is merged in a collection or few collections
- Vertical scaling is possible
- Horizontal scaling is also possible
- Great performance for massive read and write
- MongoDB, Cassandra, Dynamo, ..

# What is MongoDB ?

- MongoDB was founded in 2007, by DoubleClick guys (400,000 ads/second)
- It is free and open-source
- Document-oriented, Non relational database NoSQL
  - Hash-based, schema-less database
    - Keys are a basic data type but in reality stored as strings
    - Document Identifiers (\_id) will be created for each document, field name reserved by system
    - Uses BSON format
      - Based on JSON – B stands for Binary
  - Written in C++
  - Supports APIs in many computer languages
    - JavaScript, Python, Ruby, Perl, Java, Java Scala, C#, C++, ..

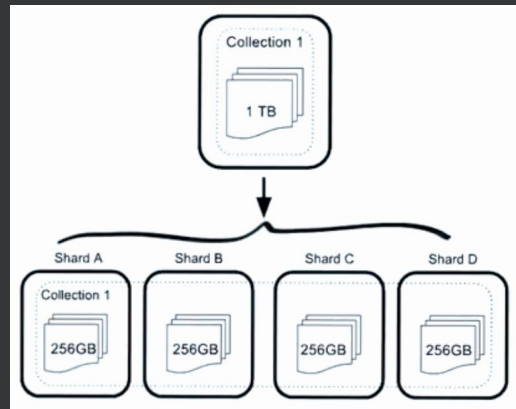
# Why MongoDB ?

- Doesn't require a lot of memory
  - No preallocated buffer pools (except for WiredTiger)
  - Makes use of the filesystem cache to cache data
  - Indexes are loaded in memory
- Flexible data model / schema-less
- Allows high levels of concurrency
- Strong consistency
- Replication ( High availability, data redundancy and failover ) : Easy for scaling reads
- Sharding of data ( Data distribution across machines ) : Easy for scaling writes

## Replication

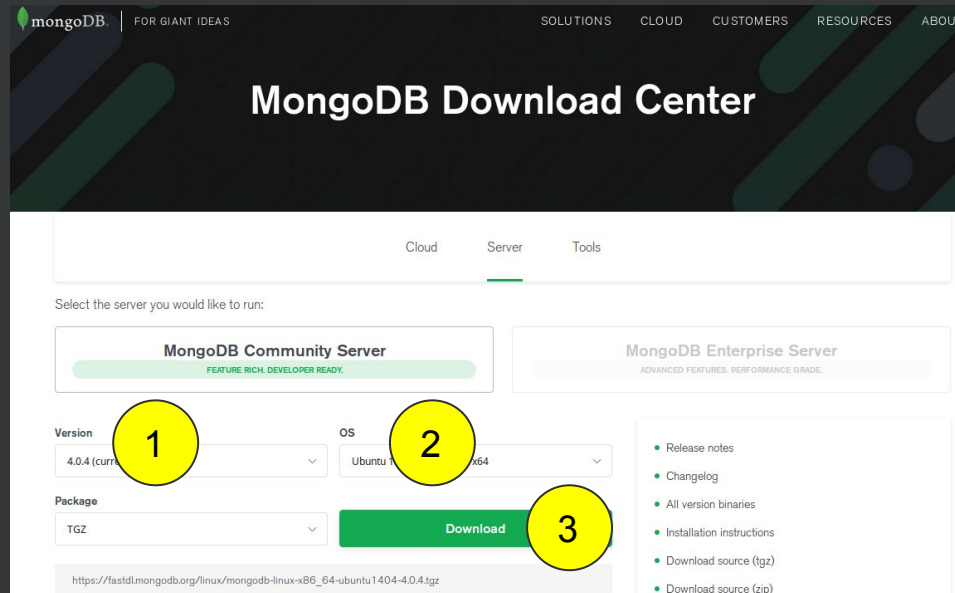


## Sharding



# Installing MongoDB server (not required for workshop)

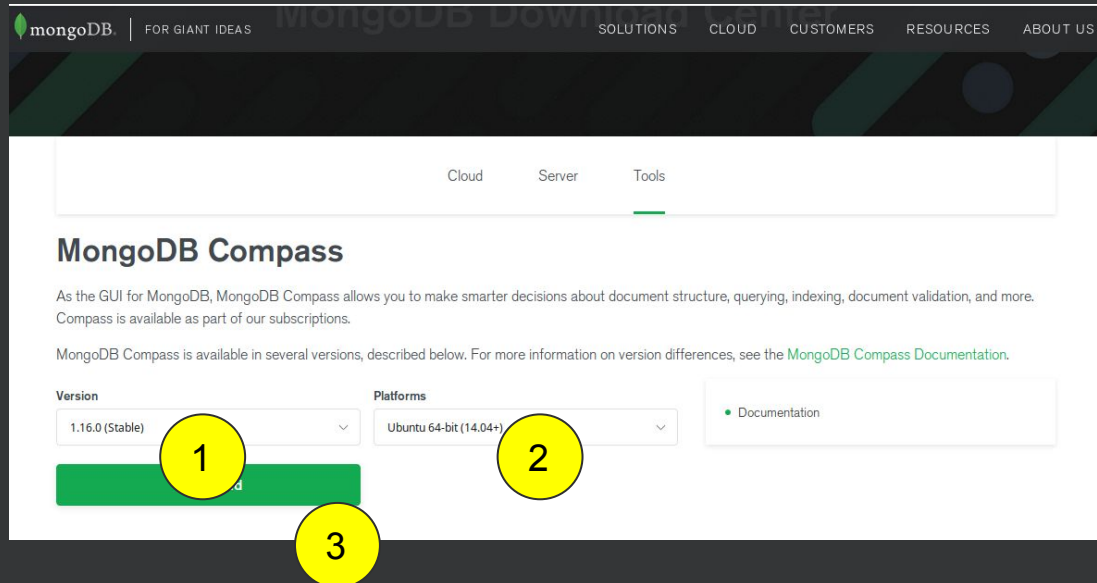
- Go to MongoDB community server [link](#) and select Version and Operating System
- Create a folder called data to store the database. (Win c:/data )





# Optional, install Mongo GUI

- MongoDB offers an graphic interface to facilitate the database analysis
- Go to MongoDB Compass [link](#) and select Version and Platform.



# Reference

- [Mongo University](#) many official free courses and exams.
- [MongoDB - Tutorials and Tech](#)
- [Getting started with Python and MongoDB](#)
- [SQL to MongoDB Mapping Chart comparison](#)
- How to install Mongo [MongoDB en 20 minutos](#) (first 2 minutes of video )
- Practice mongo queries in [mongoplayground.net](#)

# Workshop

- Go to workshop

[https://github.com/Giffy/MongoDB\\_PyMongo\\_Tutorial](https://github.com/Giffy/MongoDB_PyMongo_Tutorial)