**Giới thiệu về MongoDB**

**(Cơ sở dữ liệu NoSQL)**

1. **Khái niệm về cơ sở dữ liệu NoSQL và MongoDB**

**NoSQL:** thường được hiểu là ***Not Only SQL*** một dạng cơ sở dữ liệu cung cấp cơ chế lưu trữ và truy xuất dữ liệu theo mô hình khác với các cơ sở dữ liệu quan hệ.  NoSQL được đánh giá là có phương thức tiếp cận thiết kế đơn giản, dễ dàng mở rộng ngang và có độ sẵn sàng đáp ứng cao, dễ dàng kiểm soát. Cấu trúc dữ liệu của NoSQL được lưu trữ dưới dạng: key-value, document hoặc graph khác với cách lưu trữ mà các RDBMS đang sử dụng hiện nay. Tính chất lưu trữ đơn giản, không ràng buộc vì vậy hiệu suất hoạt động của NoSQL nhanh hơn RDBMS rất nhiều.

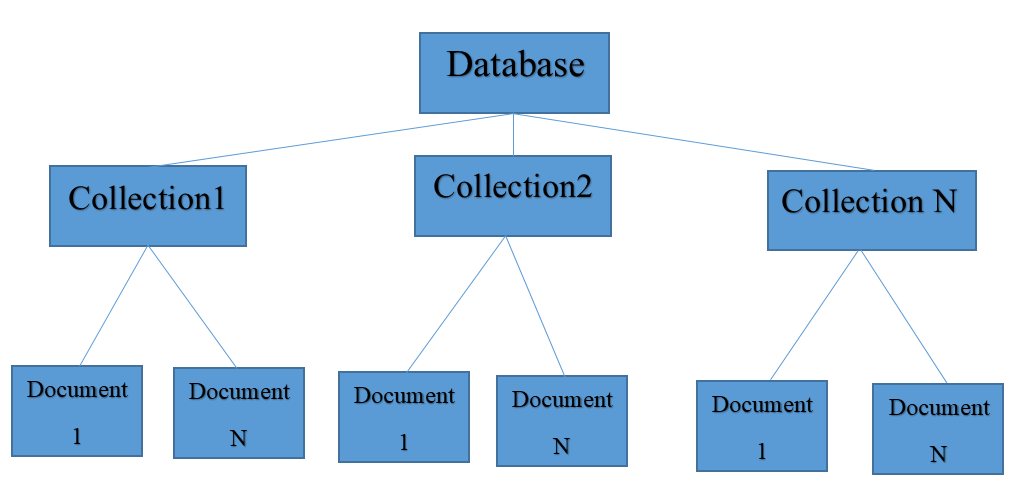


Được gọi là Not Only SQL có ý nghĩa những cơ sở dữ liệu thuộc họ NoSQL không những có thể hỗ trợ các truy vấn giống như SQL mà có thể làm được nhiều hơn thế. Hiện nay NoSQL đang được phát triển với tốc độ nhanh, được ứng dụng vào nhiều dự án ***Big Data*** (dữ liệu lớn) và các ứng dụng mạng xã hội yêu cầu đáp ứng thời gian thực.

**MongoDB:**

* MongoDB là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL hàng đầu, được hàng triệu người sử dụng. MongoDB được viết bằng C++. Mã cũng bởi vì được viết bằng C++ lên tốc độ tính toán sẽ nhanh hơn so với các CSDL khác.
* Ngoài ra, MongoDB là một cơ sở dữ liệu đa nền tảng, hoạt động trên các khái niệm Collection và Document, nó cung cấp hiệu suất cao, tính khả dụng cao và khả năng mở rộng dễ dàng.

1. **Những thành phần phần chính trong dữ liệu Mongo DB**

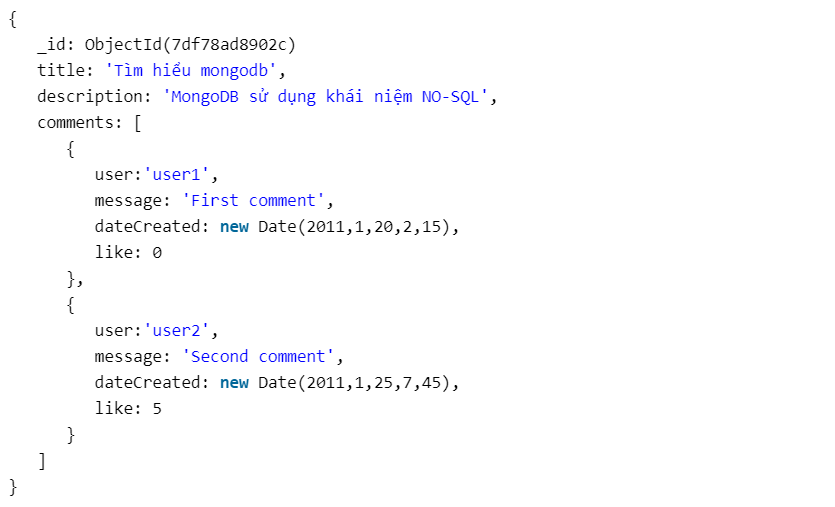


Sơ đồ thành phần cấu trúc của MongDB

|  |  |
| --- | --- |
| **RDBMS(SQL,MySQL,…)** | **MongoDB** |
| Database | Database |
| Table | Collection |
| Row | Document |
| Column | Field |
| Table Join | Embedded Documents |
| Primary Key | Primary Key (mặc định là \_id) |

**Bảng**: Đối chiếu thành phần dữ liệu của SQL với MongoDB

* **Database** là một Ô chứa dữ liệu mỗi database sẽ có nhiều collection và được thiết lập lưu trữ ở một nơi trong máy chủ máy tính. Một máy chủ MongoDB thường có thể tạo nhiều cơ sở dữ liệu.
* **Collection**: Collection là một nhóm các dữ liệu thuộc cùng loại do người dùng qui định, sự khác nhau giữa collection với table trong RDBMS đó là collection không có ràng buộc, quan hệ với collection khác. Mỗi dữ liệu trong collection không cố định về số trường như row trong table.
* **Document**: Document là một tài liệu lưu trữ dưới dạng json. Nó tương đương với row trong table của SQL nhưng dữ liệu trong document khá linh hoạt, nó không cố định có bao nhiêu trường, dữ liệu trong mỗi trường phải là cố định.
* **Embedded Documents**: Nếu trong SQL, chúng ta có kiểu join các bảng, ví dụ bảng person join với bảng address để biết một person có những address nào thì trong MongoDB, document của person sẽ chứa 1 array các document address.
* **Primary Key(\_id)**: Mặc định primary key trong MongoDB là field \_id, lúc insert nếu bạn không chỉ rõ trường \_id thì nó sẽ tự động sinh ra giá trị cho \_id.



document

Document được nhúng trong 1 document

\_id

Ví dụ về một **Collection**

**III. Các kiểu dự liệu trong MongoDB**

**String**(**chuỗi**): Đây là kiểu dữ liệu được sử dụng phổ biến nhất để lưu giữ dữ liệu. Chuỗi trong MongoDB phải là UTF-8 hợp lệ.

**Integer** (**Số nguyên**): Kiểu dữ liệu này được sử dụng để lưu một giá trị số. Số nguyên có thể là 32 bit hoặc 64 bit phụ thuộc vào Server của bạn.

**Boolean**: Kiểu dữ liệu này được sử dụng để lưu giữ một giá trị Boolean (true/false).

**Double**: Kiểu dữ liệu này được sử dụng để lưu các giá trị số thực .

**Min/ Max keys**: Kiểu dữ liệu này được sử dụng để so sánh một giá trị với các phần tử thấp nhất và cao nhất.

**Arrays** (**Mảng):** Kiểu dữ liệu này được sử dụng để lưu giữ các mảng hoặc danh sách hoặc nhiều giá trị vào trong một key.

**Timestamp(Dấu thời gian)**: Giúp thuận tiện cho việc ghi chép hoặc đánh dấu thời điểm một Document được sửa đổi hoặc được thêm vào.

**Object(Đối tượng)**: Kiểu dữ liệu này được sử dụng cho các Document được nhúng vào.

**Null** : Kiểu này được sử dụng để lưu trữ giá trị Null.

**Symbol(Biểu tượng):** Kiểu dữ liệu này được sử dụng giống nhau cho một chuỗi; tuy nhiên, nó thường dành riêng cho các ngôn ngữ sử dụng một loại ký hiệu cụ thể.

**Date(ngày)**: Kiểu dữ liệu này được sử dụng để lưu giữ date và time hiện tại trong định dạng UNIX time. Bạn có thể xác định date time riêng cho bạn bằng việc tạo đối tượng Date và truyền ngày, tháng, năm vào trong đó.

**Object ID**: Kiểu dữ liệu này được sử dụng để lưu giữ ID của Document.

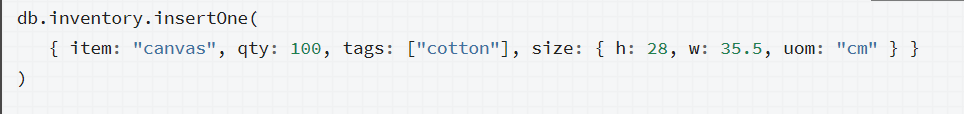
**Binary data**: Kiểu dữ liệu này được sử dụng để lưu giữ dữ liệu nhị phân

**IV. Các câu lệnh trong MongoDB**

**1.Chèn một document duy nhất**

Để chèn một document vào một Collection ta dùng **insertOne()**

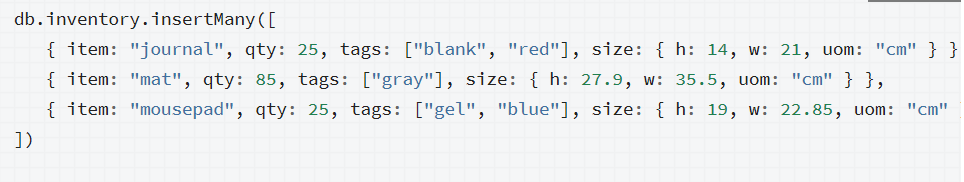
Ví dụ:thêm 1 document vào Collection inventory



**2.Chèn nhiều document vào 1 Collection**

Để chèn một document vào một Collection ta dùng [**insertMany**](https://docs.mongodb.com/manual/reference/method/db.collection.insertMany/#db.collection.insertMany)**()**

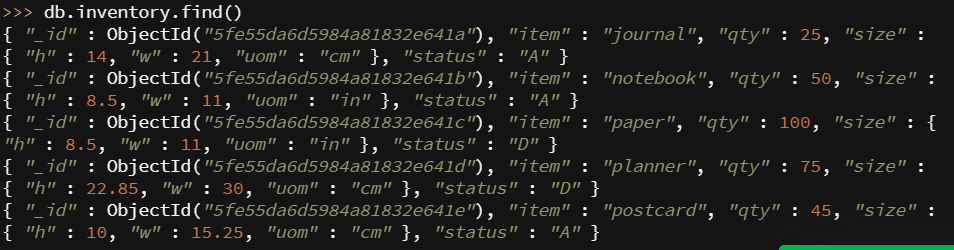
Ví dụ:thêm nhiều document vào Collection inventory



**3.Lấy toàn bộ dữ liệu từ một Collection**

Để lấy toàn bộ dữ liệu của một Collection ta dùng **find()**

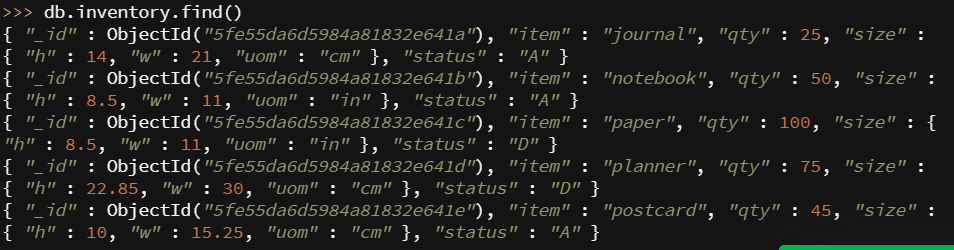
Ví dụ : Lấy toàn bộ dữ liệu của Collection inventory



**4.Lấy dữ liệu theo điều kiện**

Để lấy dữ liệu theo điều kiện của một Collection ta dùng **find({điều kiện})**

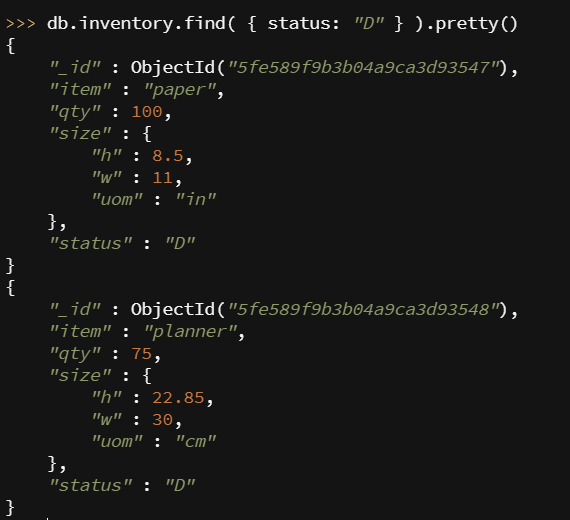
Dữ liệu của Collection **inventory**



4.1 : Cách viết điều kiện không có toán tử

Để chỉ định điều kiện bình đẳng, hãy sử dụng **<field>:<value>**

Ví dụ sau đây chọn từ inventorybộ sưu tập tất cả các tài liệu có giá trị statusbằng "D":



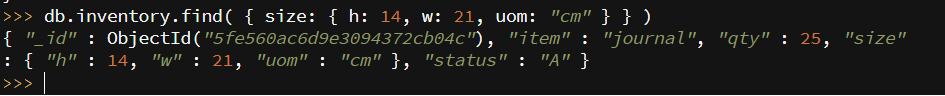
Value

Fied

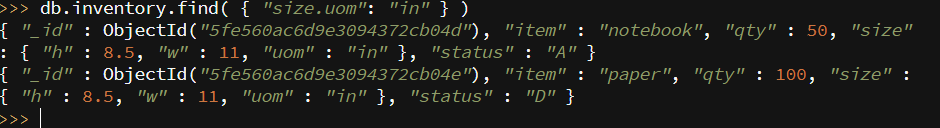
4.2 Các truy vấn tài liệu lồng nhau

Để truy vấn điều kiện lồng nhau ta sử dụng **{ <field>:<value>}<value>**

Ví dụ: truy vấn sau chọn tất cả các tài liệu có trường sizebằng tài liệu :{ h: 14, w: 21, uom: "cm" }



Ví dụ sau đây chọn tất cả các tài liệu có trường được uom lồng trong sizetrường bằng "in":



**5.Truy vấn có toán tử**

5.1:Sử dụng Và(AND) trong MongoDB

Cú pháp: Để truy vấn tài liệu dựa trên điều kiện AND, ta cần sử dụng $ và từ khóa. Sau đây là cú pháp cơ bản của AND



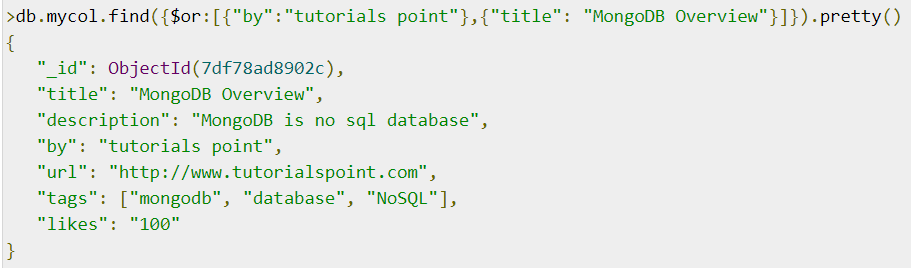
Ví dụ sau sẽ hiển thị tất cả các hướng dẫn được viết bởi **'tutorials point**' và có title là **'MongoDB Overview'**.



5.2 Sử dụng toán tử hoặc (or)

Cú pháp: Để truy vấn tài liệu dựa trên điều kiện OR, ta cần sử dụng **$ và**  từ khóa. Sau đây là cú pháp cơ bản của **OR**

Ví dụ sau sẽ hiển thị tất cả các hướng dẫn được viết bởi **'tutorials point**' hoặc có title là **'MongoDB Overview'**.



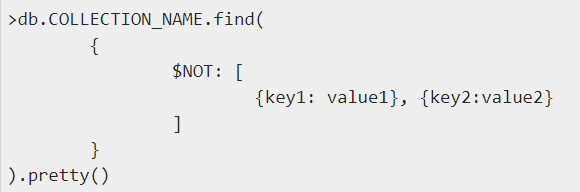
5.3 Sử dụng AND và OR cùng nhau

Ví dụ sau sẽ hiển thị các tài liệu có lượt thích lớn hơn 10 và có tiêu đề là 'Tổng quan về MongoDB' hoặc bằng 'điểm hướng dẫn'. Mệnh đề SQL tương đương where là **'where like> 10 AND (by =' tutorial point 'OR title =' MongoDB Overview')'**



5.4 Sử dụng NOT trong MongoDB

Cú pháp: Để truy vấn tài liệu dựa trên điều kiện NOT, bạn cần sử dụng từ khóa $NOT, sau đây là cú pháp cơ bản của **NOT**



Ví dụ truy vấn những người có tuổi không lớn hơn 25 tuổi



5.4 Các loại toán tử so sánh

**$lt** : Kiểu so sánh nhỏ hơn

**$lte** : Kiểu so sánh nhỏ hơn hoặc bằng

**$gt** : Kiểu so sánh lớn hơn

**$gte** : Kiểu so sánh lớn hơn hoặc bằng

**$ne** : Kiểu so sánh không bằng

**$size :** Truy vấn theo độ dài mảng

**6. Các cách xóa tài liệu**

6.1: Xóa tất cả các tài liệu

Để xóa hết các document trong một Collection ta dùng **deleteMany()**

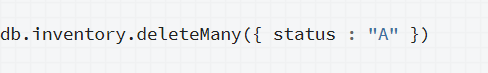
Cú pháp :



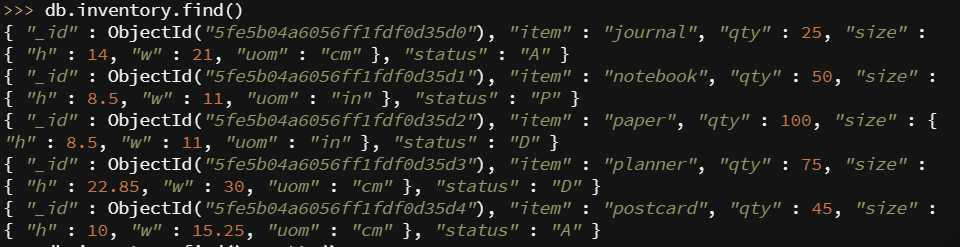
6.2 : Xóa tất các các điều kiện có phù hợp với điều kiện

Ví dụ sau đây xóa tất cả tài liệu khỏi inventory bộ sưu tập trong đó status trường bằng "A"

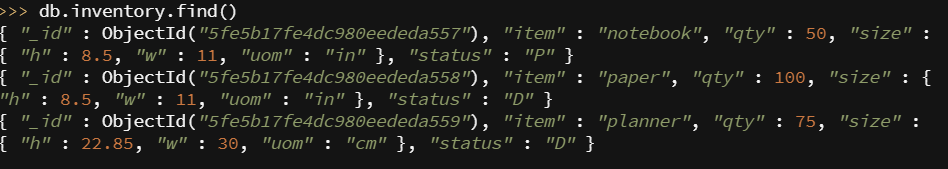
Cú pháp :



Collection khi chưa xóa



Collection khi xóa những **status : A**



**Sharding**

[Sharding](https://docs.mongodb.com/manual/reference/glossary/#term-sharding) là một phương pháp phân phối dữ liệu trên nhiều máy chủ. MongoDB sử dụng sharding để hỗ trợ triển khai với tập dữ liệu rất lớn và hoạt động thông lượng (tốc độ truyền dữ liệu thông qua mọi bộ phận trong hệ thống của máy tính) cao.

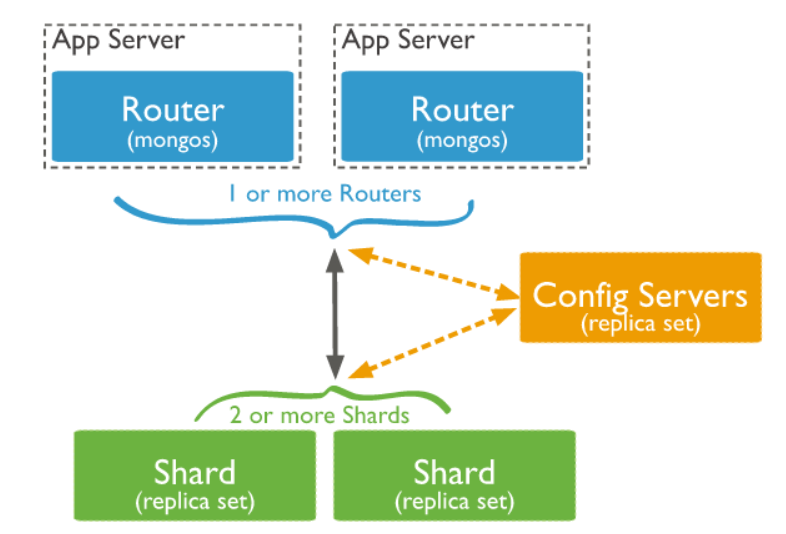
Có hai phương pháp để giải quyết sự phát triển của hệ thống: mở rộng theo chiều dọc và theo chiều ngang.

Vertical Scaling (chia tỷ lệ theo chiều dọc ) liên quan đến việc tăng dung lượng của một máy chủ, chẳng hạn như sử dụng CPU mạnh hơn, bổ sung thêm RAM hoặc tăng dung lượng lưu trữ.

*Horizontal scaling* (chia tỷ lệ theo chiều ngang )liên quan đến việc chia nhỏ dữ liệu của hệ thống và tải chúng lên nhiều server , thêm server để tăng công suất . Mặc dù tốc độ hoặc công suất chung của mỗi máy có thể vẫn thế nhưng mỗi máy xử lý một tập hợp con của khối lượng dữ liệu chung vì thế nó có khả năng mang lại hiệu quả tốt hơn so với một server công suất cao .\

Và MongoDB sử dụng phương pháp mở rộng theo chiều ngang

Một sharded cluster(cụm phân nhánh) bao gồm các thành phần sau:



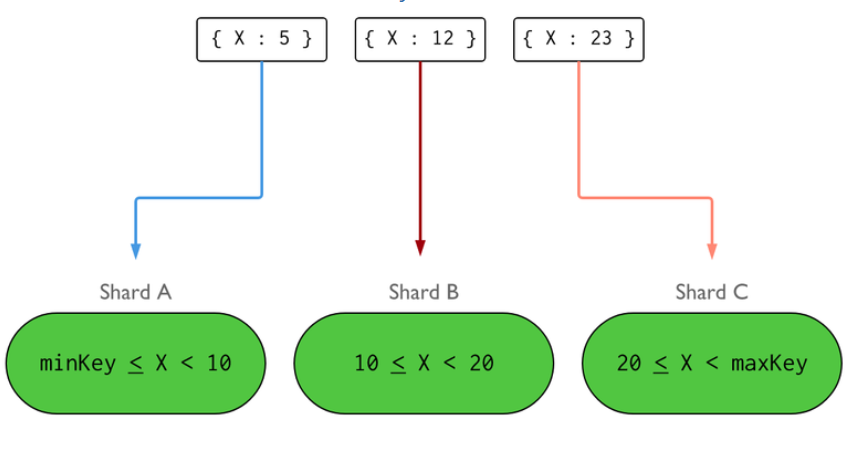
* **shard**: mọi người hãy hiểu shard là đơn vị quản lý dữ liệu của sharded cluster, mỗi shard thường là 1 replica set
* **mongos**: đóng vai trò như một bộ định tuyến, là interface đối với người dùng, người dùng chỉ giao tiếp với mongos và không cần quan tâm bên trong có cái gì
* **config server**: là một máy chủ mongo (mongod) chứa các thông tin cấu hình của cluster(cụm), như document(tài liệu) nào nằm ở shard nào chẳng hạn. Từ bản 3.0 trở đi, config server cũng là 1 replicaset

Vậy chính xác các shard chứa gì, các shard chứa các collection hoặc chứa 1 phần của collection

Mongodb chia sẻ dữ liệu ở tầng [**Collection**](https://www.hidemyass-freeproxy.com/proxy/vi-vn/aHR0cHM6Ly9kb2NzLm1vbmdvZGIuY29tL21hbnVhbC9yZWZlcmVuY2UvZ2xvc3NhcnkvI3Rlcm0tY29sbGVjdGlvbg)**(Tương ứng với Table trong hệ quản trị cơ sở dữ liệu quan hệ).**Dữ liệu trong Collection được phân tán trên các Shards trong cụm.

Để phân tán dữ liệu của Collection trên các Shards, MongoDb phân vùng dữ liệu thông qua **Shard Keys**. Shard Key là một cột hoặc một cụm các cột không thể thay đổi tồn tại ở mỗi [**Documents**](https://www.hidemyass-freeproxy.com/proxy/vi-vn/aHR0cHM6Ly9kb2NzLm1vbmdvZGIuY29tL21hbnVhbC9jb3JlL2RvY3VtZW50Lw) thuộc Collection. Chúng ta cần phải chọn một Shard Keys mỗi khi chúng ta phân chia dữ liệu của **Collection**.

Bạn phải chọn shard key khi sharding một collection. Việc lựa chọn shard key không thể thay đổi sau khi sharding. Một sharded collection có thể chỉ có một shard key duy nhất . Để tìm hiểu thêm về [cấu hình shard key](https://docs.mongodb.com/manual/core/sharding-shard-key/#sharding-shard-key-creation)



**hình minh họa vs shard key là x**

**Nhược điểm của Sharding :**

* Phụ thuộc nhiều vào kết nối giữa các máy chủ
* Độ trễ tăng lên khi truy vấn,  đặc biệt khi phải tìm kiếm nhiều hơn một phân đoạn.
* Dữ liệu hoặc chỉ mục thường chỉ được chia nhỏ theo một chiều, do đó, một số tìm kiếm là tối ưu, còn những tìm kiếm khác thì chậm hoặc không thể.
* Các vấn đề về tính nhất quán và độ bền do các chế độ lỗi phức tạp hơn của một tập hợp các máy chủ, thường dẫn đến hệ thống không đảm bảo về tính nhất quán hoặc độ bền của các phân đoạn chéo.

**Replication**

Replication là tiến trình đồng bộ hóa dữ liệu từ nhiều Server. Replication cung cấp sự dư thừa và tăng dữ liệu có tính khả dụng với nhiều bản sao dữ liệu trên nhiều Database Server khác nhau. Replication bảo vệ một cơ sở dữ liệu từ việc thất thoát của một Server nào đó. Replication cũng cho phép bạn phục hồi dữ liệu từ việc các lỗi ở phần cứng hoặc từ việc ngắt kết nối dịch vụ. Với các bản sao dữ liệu bổ sung, bạn có thể sử dụng cho việc phục hồi, báo cáo, hoặc backup

Quá trình replication khác với quá trình backup / sao lưu ở chỗ nó thực hiện realtime. Tức là khi thêm dữ liệu vào A thì nó sẽ lập tức đồng bộ sang B   quá trình backup thì thường thực hiện theo lịch và có thể phải xây dựng lại chỉ mục / index

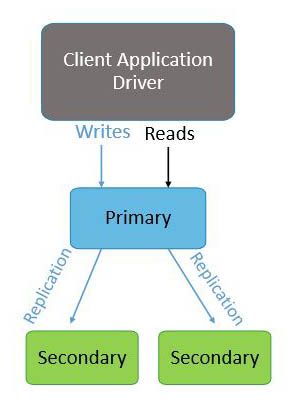
**Mô hình hoạt động của Replication trong MongoDB**

Replica Set là một nhóm các server MongoDB trong hệ Replication.

Mỗi server MongoDB trong Replica Set được coi là 1 node hoặc 1 thể hiện.

Trong một Replica Set, sẽ có một node chính (primary node) và nhiều node phụ (secondary node) + node trọng tài (arbiter node).

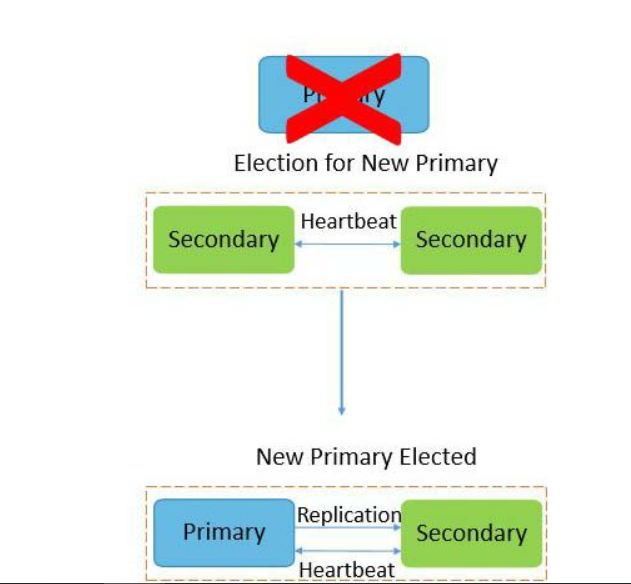
Primary node sẽ thực hiện nhận tất cả các write request để thực hiện thêm/sửa/xóa data, các thay đổi trên data này sẽ được ghi lại vào file oplog, khi file oplog thay đổi, nó sẽ dựa vào các thông tin thay đổi đó để đồng bộ dữ liệu sang các secondary node. (arbiter node sẽ không thực hiện ghi dữ liệu)



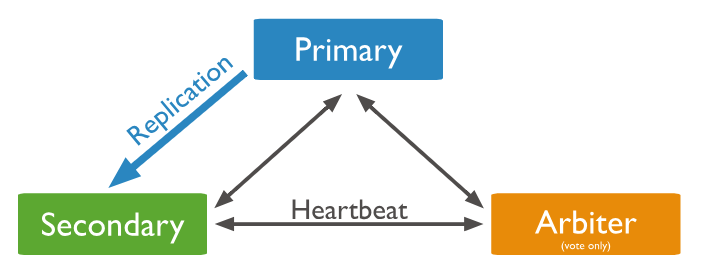
Sơ đồ đưa ra ở trên mô tả rằng thành viên sơ cấp của bộ sao chép mặc định nhận được tất cả các yêu cầu đọc và ghi

### **Tự động chuyển đổi dự phòng trong MongoDB**

Khi nút sơ cấp của bộ sao chép dừng liên lạc với các thành viên khác trong hơn 10s hoặc bị lỗi, bộ sao chép sẽ lựa chọn một thành viên khác như là một nút sơ cấp



Quá trình bầu primary node giống như bỏ phiếu vậy, node nào nhiều phiếu hơn thì trở thành primary node. Tuy nhiên sẽ có trường hợp có 2 node cùng số phiếu. Trong trường hợp này node arbiter sẽ quyết định node nào là primary node.



Khi primary node bị ngắt kết nối lúc trước được kết nối trở lại vào Replica Set thì nó sẽ trở thành secondary node.

Mặc định, Client sẽ đọc dữ liệu từ primary node. Tuy nhiên ta có thể cấu hình cho phép client đọc dữ liệu từ các secondary node để giảm tải áp lực cho primary node.

**Cài đặt và kết nối MongoDB**

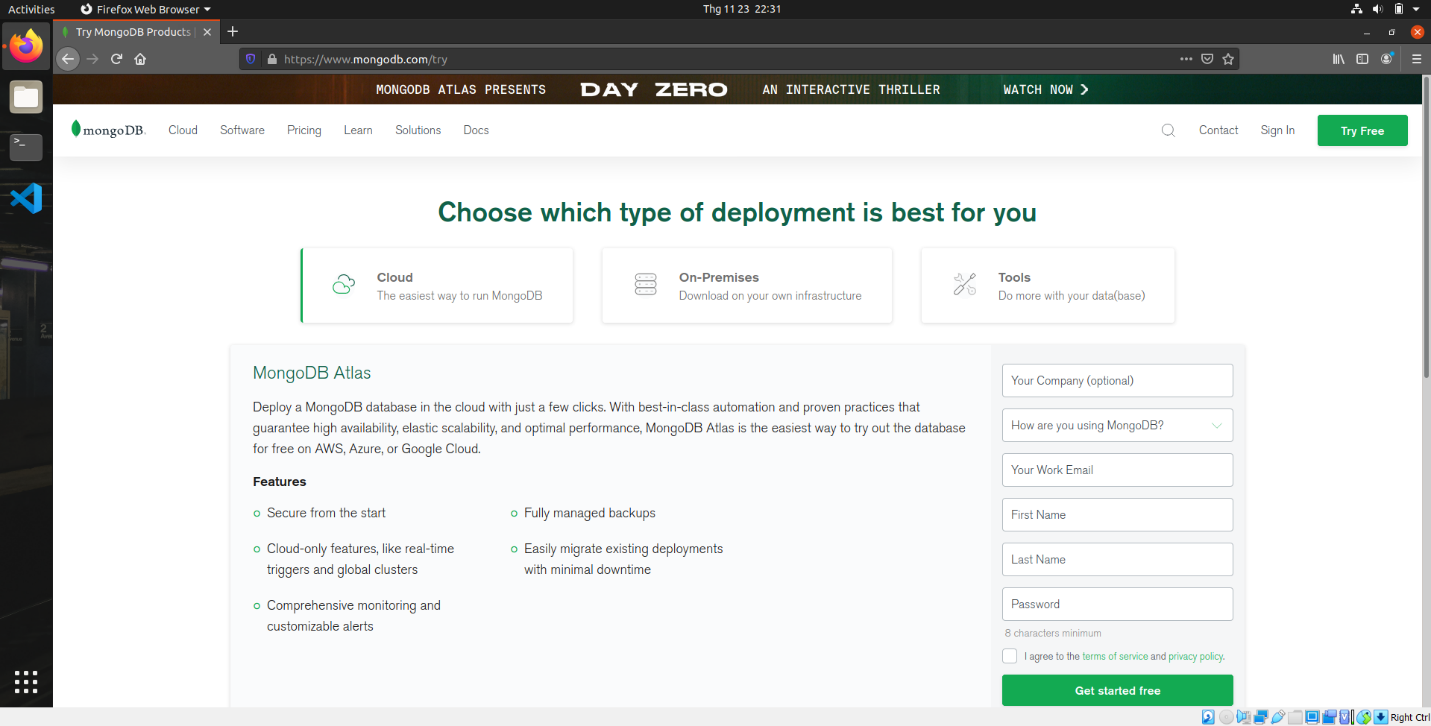
Cài đặt mongodb và kết nối môngdb

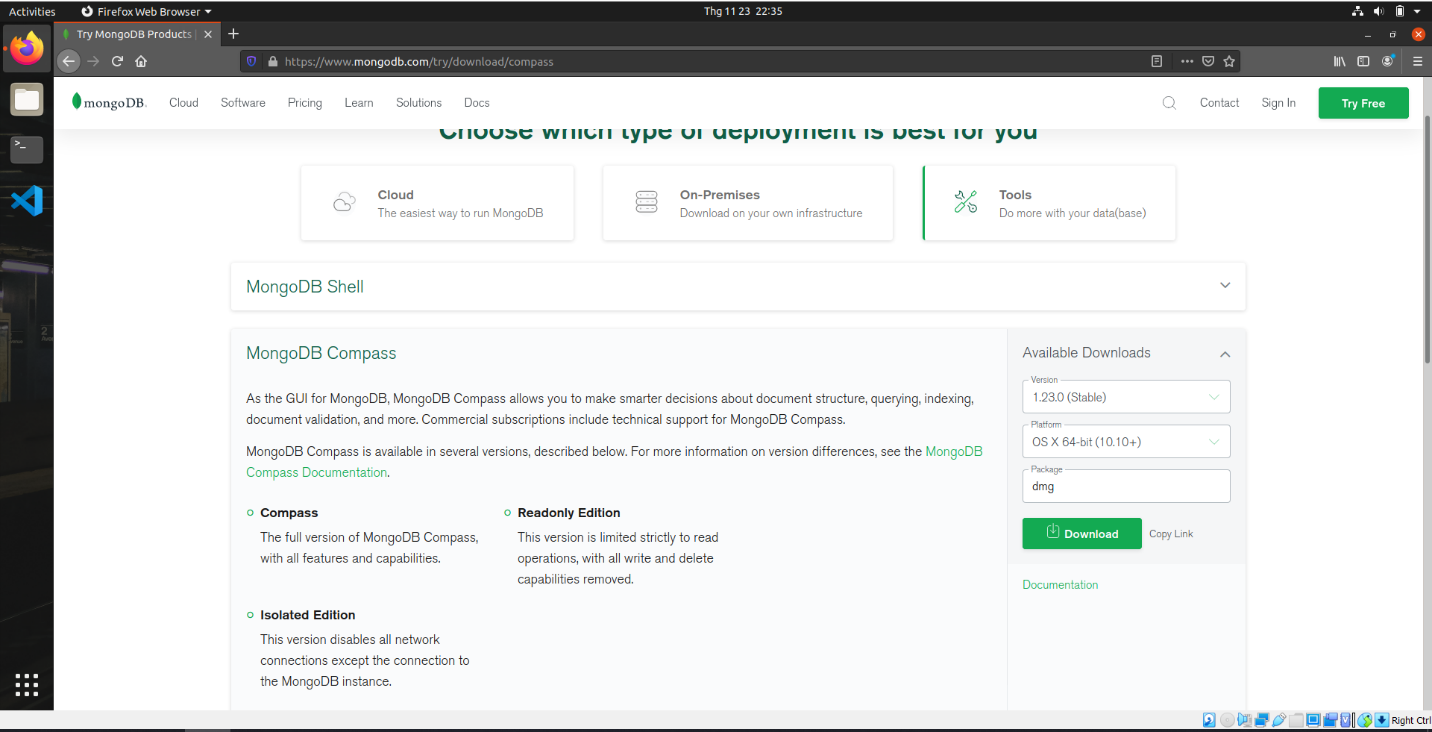
Mở trình duyệt web gõ mongodb.com -- Có 3 lựa chọn

Cloud: cái này tóm lại bạn không phải cái mongo mà sử dụg trên cloud,cloud tự cấp cloud quảm lý cái d

on-pre: tự cài lên máy

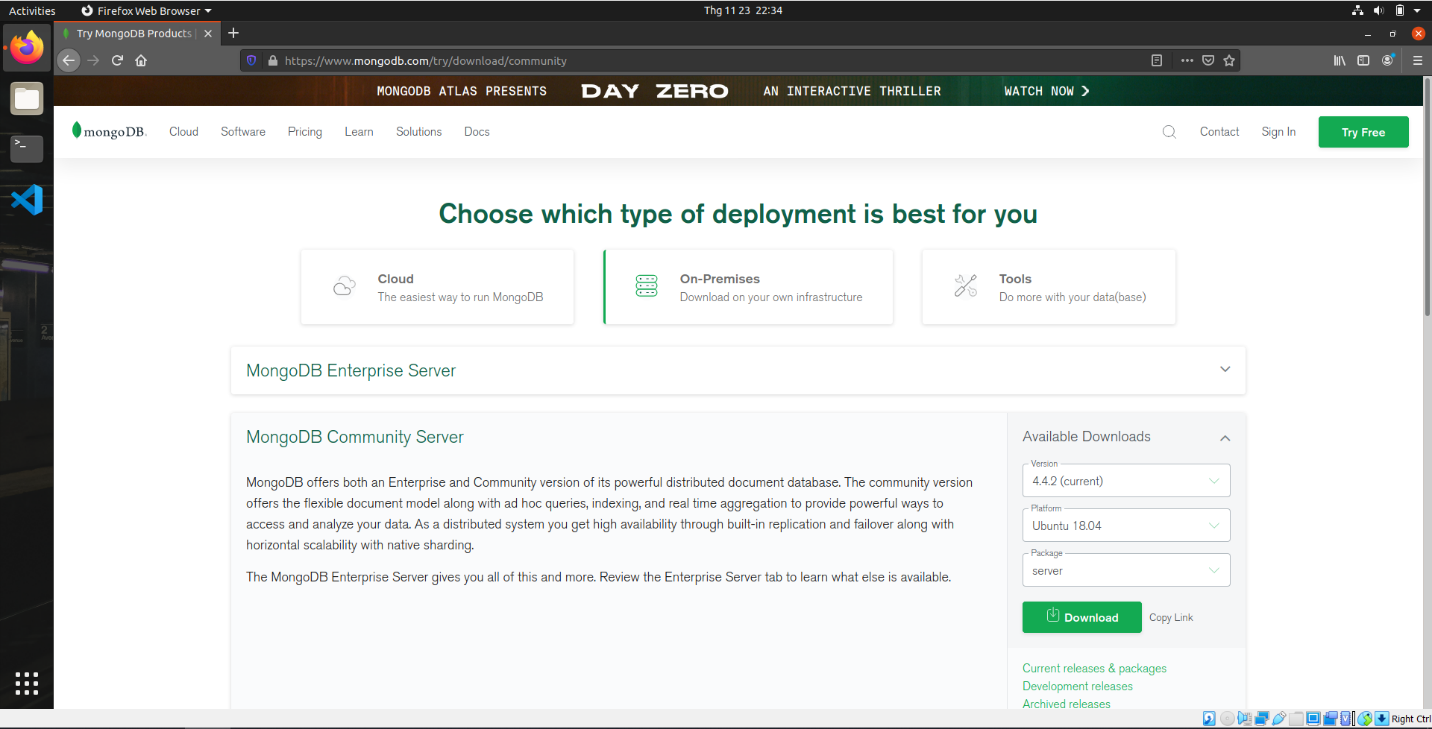
Tools: công cụ hỗ trợ khi chúng ta lm vc vs mongo



đsscdc

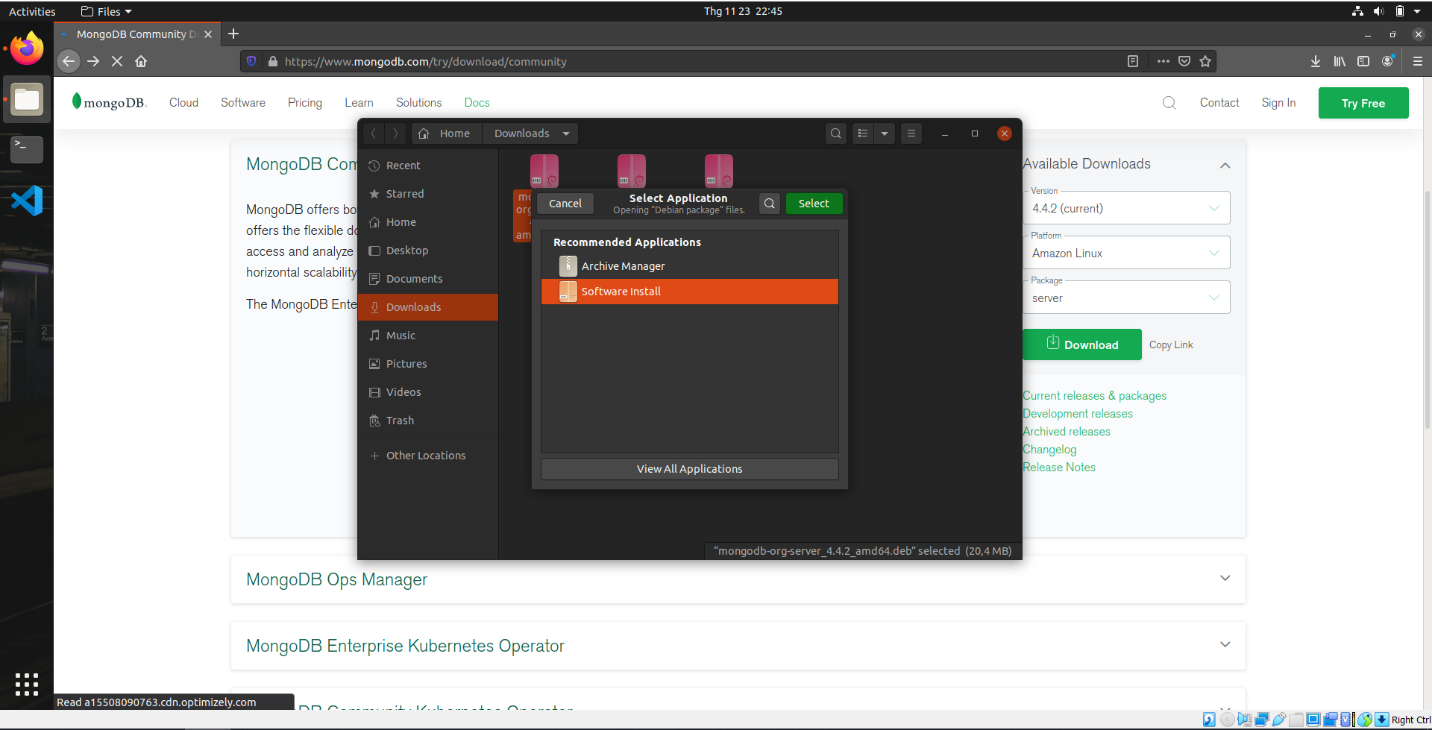
Tải mongodb compass – chọn hệ điều hành

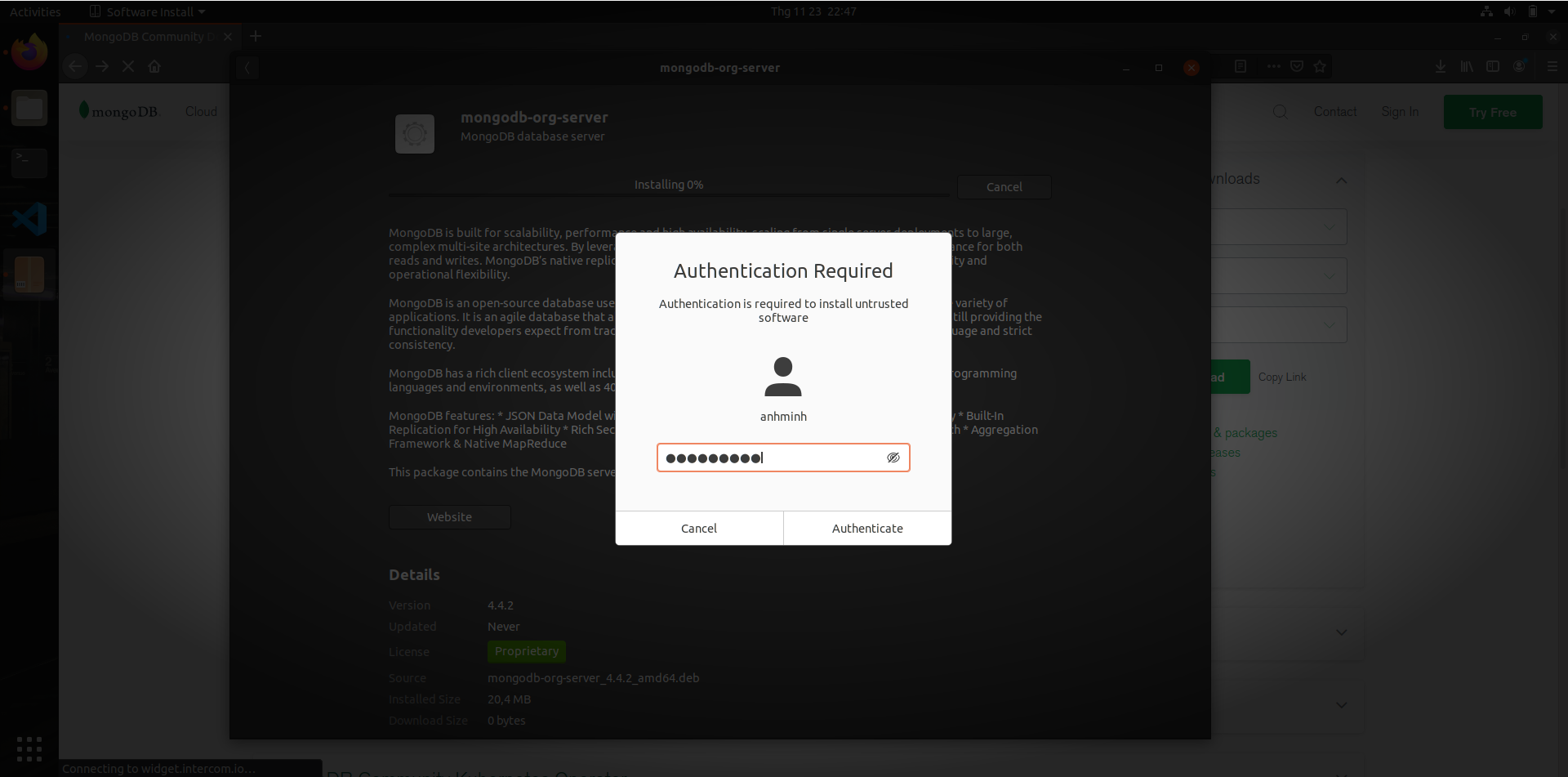
Chọn mongodb community – chọn hệ diều hành





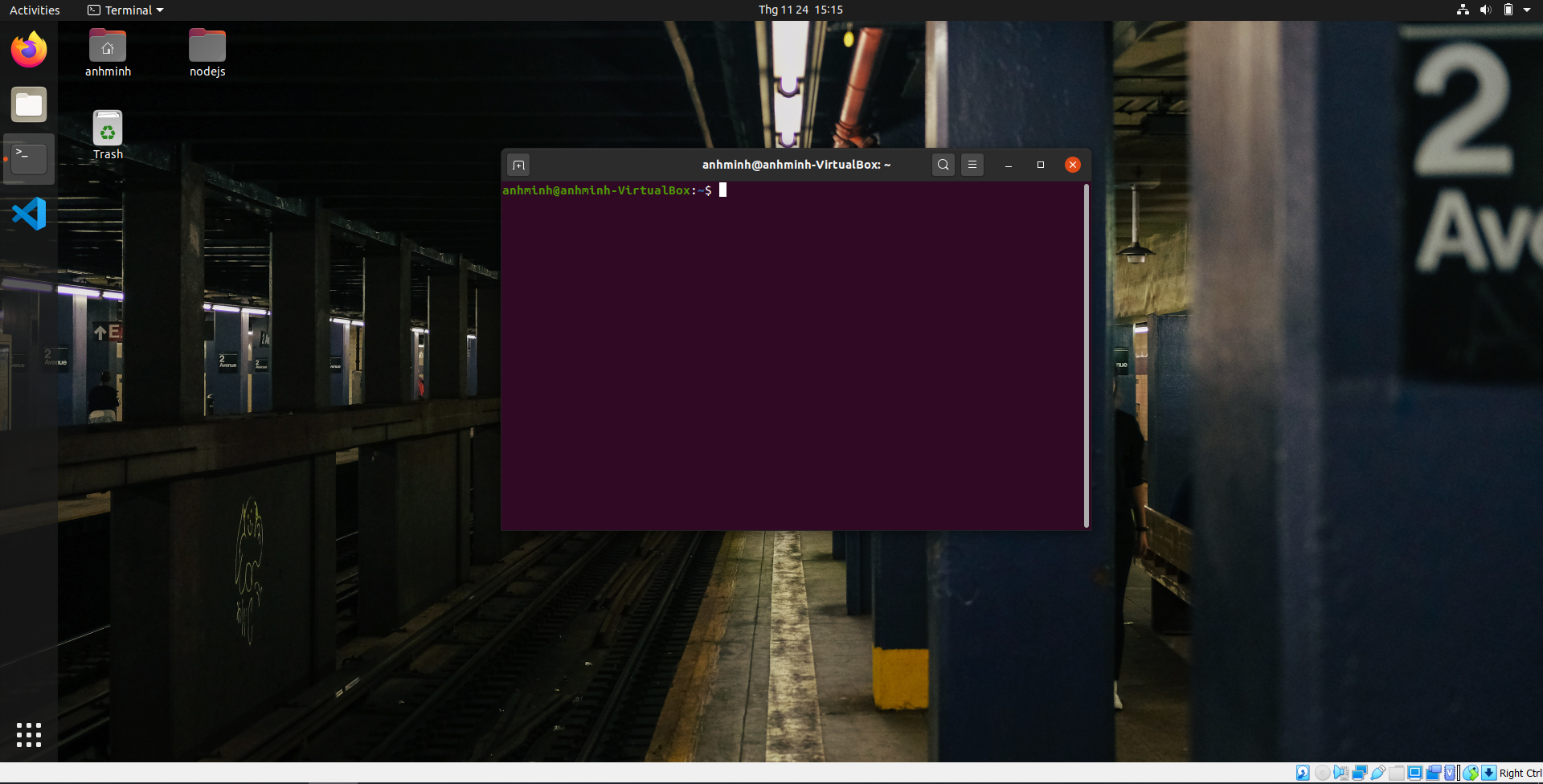
Dùng trên Ubuntu chúng ta có thể install luôn. Bấm chuột phải và intall



Nhập mật khẩu mấy tính của của bạn vào và intall

Mở terminal (lưu ý : nếu các bạn mở terminal trước khi cài thì hãy tắt đi và mở lại khi đó nó mới vào được cái biến pass ở biến môi trường mongodb

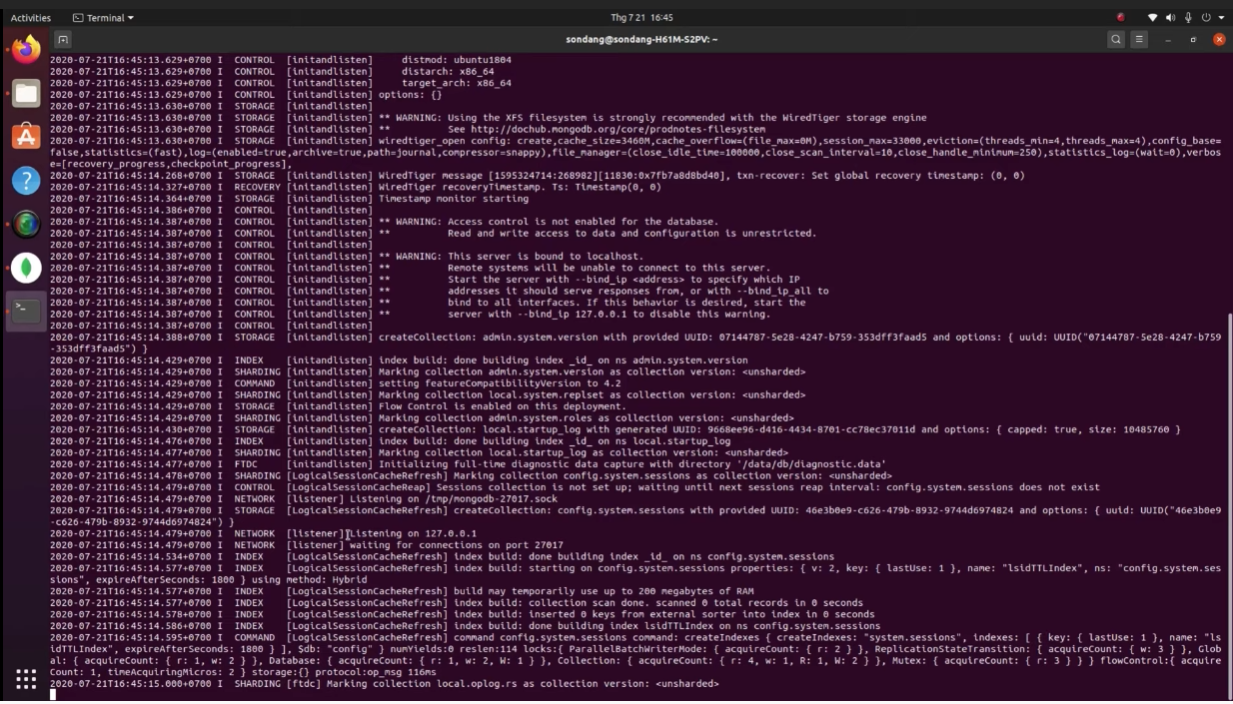
Kiểm tra: gõ mongod



Nếu chưa thi nó sẽ báo là mongodb not’t false

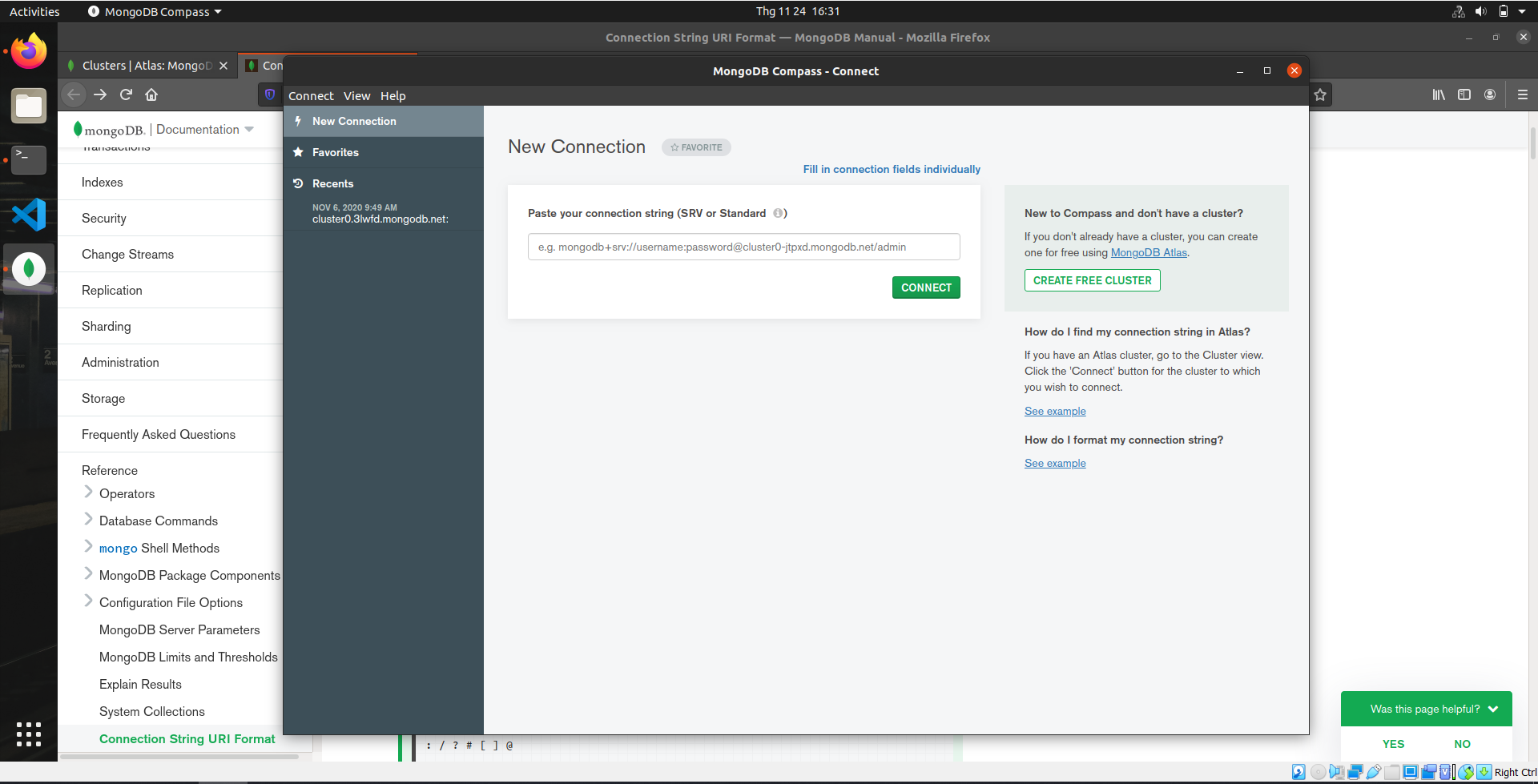
Các bạn sẽ đọc bên này sẽ thấy nó đang chạy ở 127.0.0.1 chính là cái localhost và nó sẽ open ra cái port là : 27017

Bây giờ các bạn connec sever mongodb qua địa chỉ ip là 127.0.0.1 và port là 27017

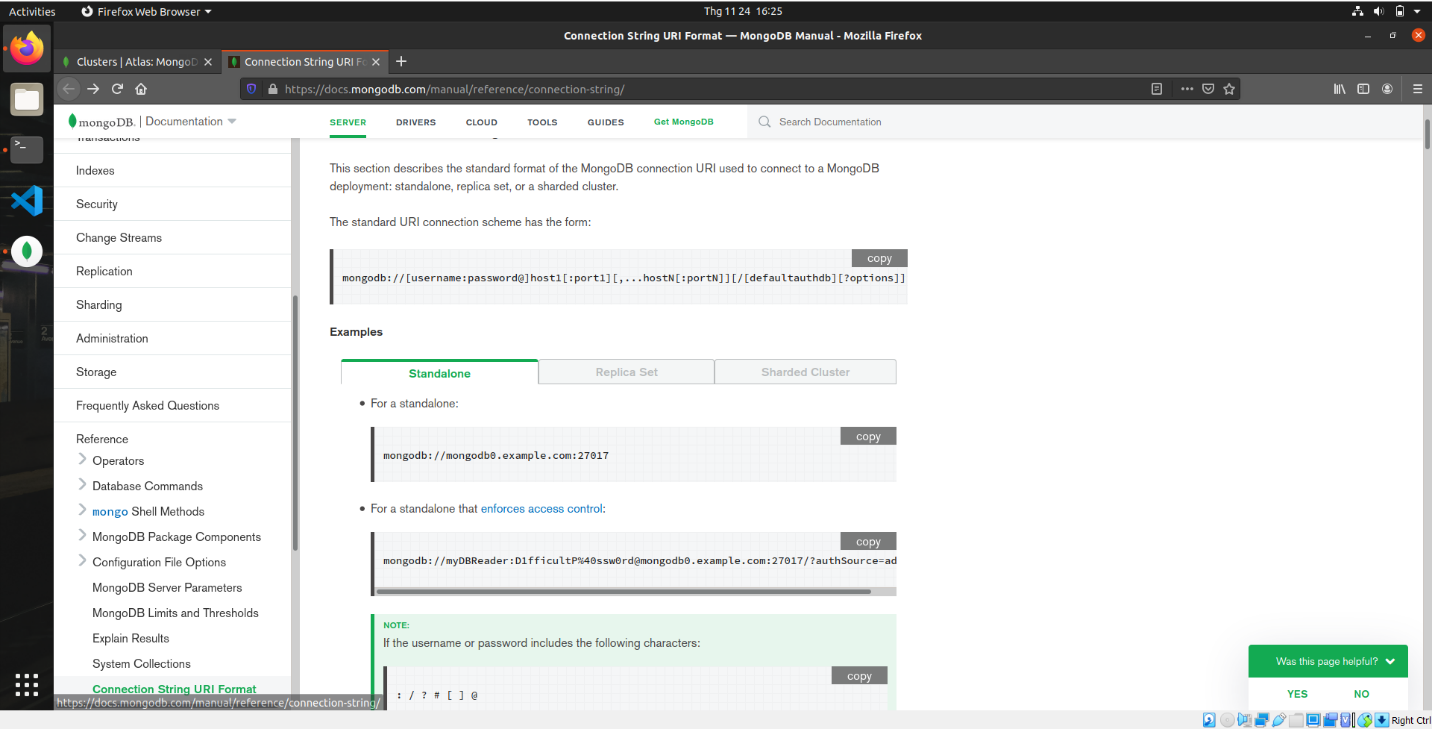


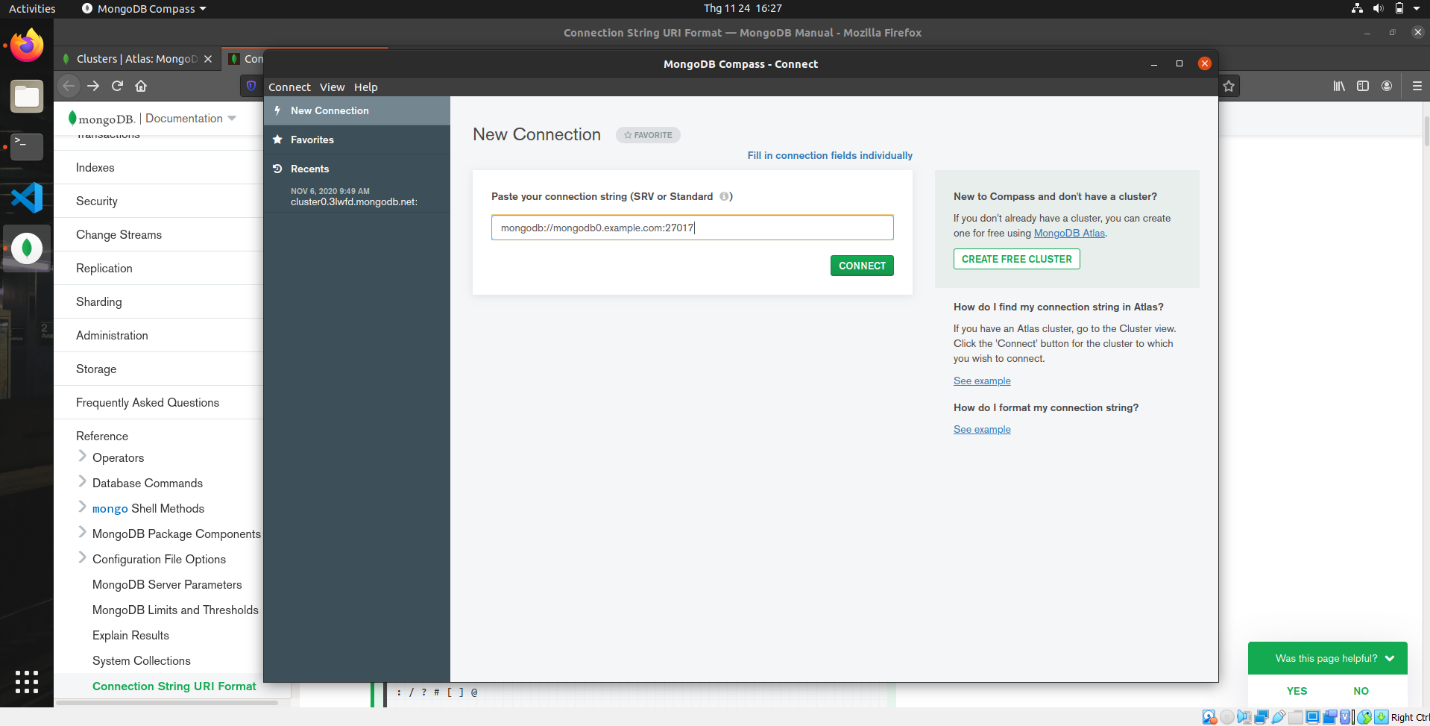
Sang compasss nhập connection string là kết nối được.

Sau đso các bạn bấm vào see exemple để xem connec string cái mẫu của nó.

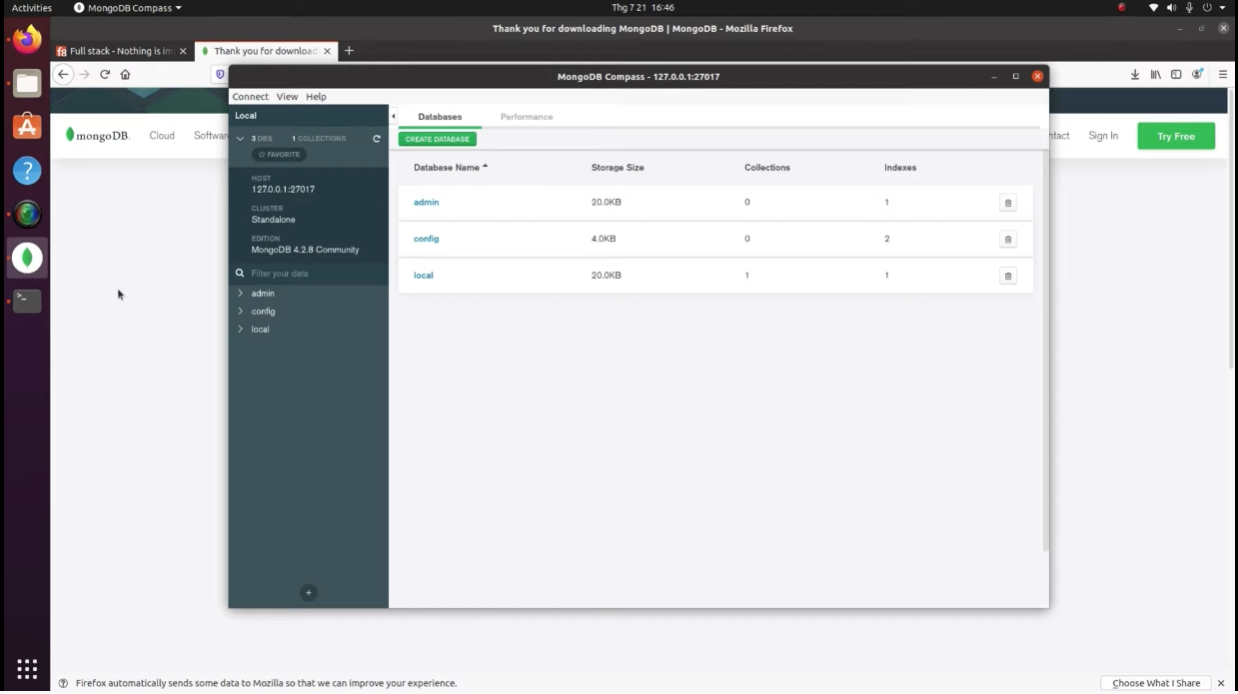


Coppy dòng lệnh standalone và paste



Thay hotname nhập vào 127.0.0.1 hoặc locahosst cũng được bấm connec.

Thành công!



Kết nối mongodb.

