**National University of Ho Chi Minh City UNIVERSITY OF INFORMATION TECHNOLOGY**

Logo, company name

Description automatically generated

**FINAL PROJECT REPORT**Subject: Database Management Systems

Semester II (2021-2022)

**TOPIC:**

COUCHDB

Student:

Tran Nhat Tan ID: 19522177

Instructor: Mr. Nguyen Thanh Binh  
 Mrs. Truong Thu Thuy

Class: IS210.M22.HTCL

**Ho Chi Minh city, June 2022**

**National University of Ho Chi Minh City UNIVERSITY OF INFORMATION TECHNOLOGY**

Logo, company name

Description automatically generated

**FINAL PROJECT REPORT**Subject: Database Management Systems

Semester II (2021-2022)

**TOPIC:**

COUCHDB

Student:

Tran Nhat Tan ID: 19522177

Instructor: Mr. Nguyen Thanh Binh  
 Mrs. Truong Thu Thuy

Class: IS210.M22.HTCL

**Ho Chi Minh city, June 2022**

Table of Contents

[**ACKNOWLEDGEMENT** 4](#_Toc106659379)

[**TEACHER’S COMMENTS** 5](#_Toc106659380)

[**CHAPTER I : INTRODUCTION** 6](#_Toc106659381)

[**1.1** **Introduce about CouchDB** 6](#_Toc106659382)

[**1.2** **Data model** 6](#_Toc106659383)

[**1.3** **History of formation** 9](#_Toc106659384)

[**1.4** **Installation instructions** 10](#_Toc106659385)

[**1.4.1 Computer configuration details:** 10](#_Toc106659386)

[**1.4.2 Install CouchDB on Windows** 10](#_Toc106659387)

[**CHAPTER II : INTERFACE AND MANIPULATION IN COUCHDB** 13](#_Toc106659388)

[**2.1** **Curl:** 13](#_Toc106659389)

[**2.2** **Futon:** 15](#_Toc106659390)

[**CHAPTER III : DISTRIBUTION IN COUCHDB** 20](#_Toc106659391)

[**3.1** **Distribution model** 20](#_Toc106659392)

[**3.2** **The CAP theorem** 20](#_Toc106659393)

[**3.3** **Data operation of 2 computers** 22](#_Toc106659394)

[**Single node setup:** 22](#_Toc106659395)

[**Cluster setup:** 25](#_Toc106659396)

[**CHAPTER IV : REPLICATION IN COUCHDB** 32](#_Toc106659397)

[**4.1** **Introduction to replication** 32](#_Toc106659398)

[**4.2** **Scattered architecture with replication** 32](#_Toc106659399)

[**4.3** **Replication mechanism** 34](#_Toc106659400)

[**REFERENCE** 39](#_Toc106659401)

[**The end.** 40](#_Toc106659402)

# **ACKNOWLEDGEMENT**

In fact, there is no success that is not tied to the support and help, whether more or less, directly or indirectly from others. With the deepest gratitude, first of all, our group would like to express our sincere thanks to the staff of the University of Information Technology - Vietnam National University, Ho Chi Minh City and the teachers of the Faculty of Information Systems. helped the group to have the basic knowledge as a basis to carry out this topic.

First of all, I would like to express our deepest gratitude to Mr. Nguyen Thanh Binh and Mrs. Truong Thu Thuy - Lecturer of Database management systems, University of Information Technology - National University of Ho Chi Minh City for their enthusiasm. guide, give valuable suggestions for us to complete the project well!

I would like to sincerely thank the lessons and materials provided by the two teachers to help us accumulate more knowledge and have a better and more complete insight into Database management systems.

Finally, thank you to all, Im worked at my best to complete the thesis well. Sincerely thank!

# **TEACHER’S COMMENTS**

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

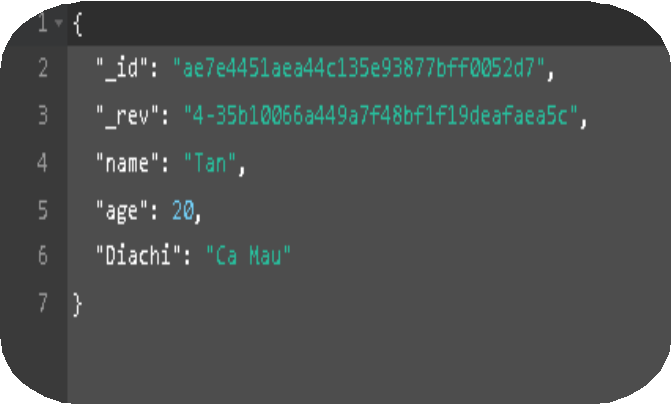
# **CHAPTER I : INTRODUCTION**

## **Introduce about CouchDB**

* It is an open source NoSQL database management system, whose data model is document store.
* Designed to be easy to use and cater to the web environment.
* JSON for storage .
* Javascript as the query language to convert documents.
* Ensure the ACIDity of the data.

## **Data model**

* The forms of nosql include: key - value, document, graph, hybrid,....



Picture: An example of a document.

* CouchDB is a NoSQL document database. A document is a unit of data (like an object of Javascript), each field has its own name that does not overlap, containing types of data such as letters, numbers, Boolean, lists ... There is no limit to the amount of text or field space in a document.
* CouchDB provides a RESTFul API for reading and writing (adding, editing, deleting documen

Diagram

Description automatically generated

* Data stored as B-Tree.
* When we add a document, it will initialization a new id as a key in B-Tree.

Diagram

Description automatically generated

* Map-reduce model help searching, filtering and generate the data easier.

## **History of formation**

* CouchDB is written in the programming languages Erlang, JS, C, C++.



* The CouchDB project was created in April 2015.

2005 by Damien Katz, a former playwright

Lotus at IBM. He self-funded the project for nearly two years and released it as an open source project under the GNU Public License.

Daminen Katz

* CouchDB became a project managed by the Apache Software Foundation in 2008.
* In early 2012, Katz left the project to focus on [couchbase server.](https://en.wikipedia.org/wiki/Couchbase_Server)
* Since Katz's departure, the ApacheCouchDB project has continued, releasing further versions 1.2 in April 2012 and 1.3 in April 2013  
  .
* Naitive cluster is supported in version 2.0.0.
* And the new Mango Query Server offers a simple JSON-based way to perform CouchDB queries without JavaScript or MapReduce.

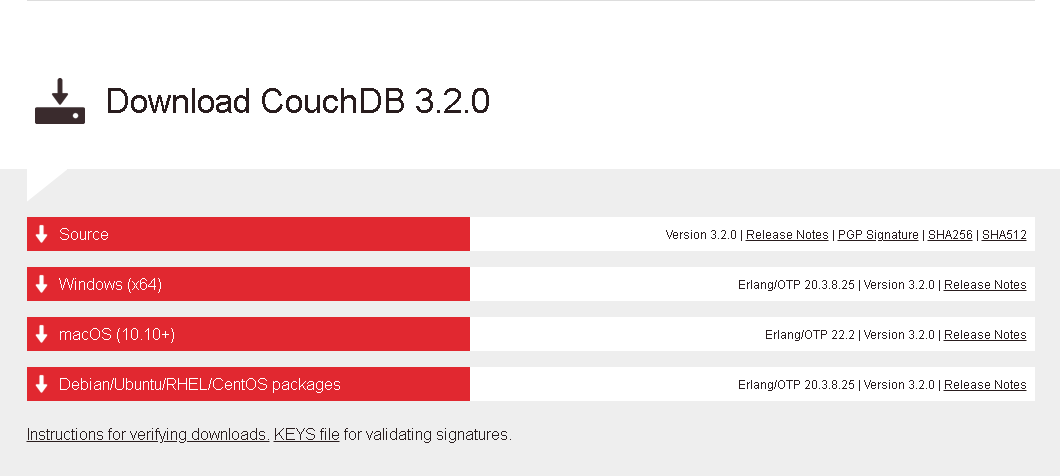
## **Installation instructions**

### **1.4.1 Computer configuration details:**

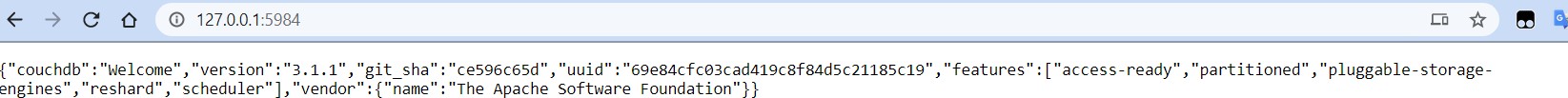
* Windows edition: Windows 10 Home Single © 2021 Microsoft Corporation System:
* Processor: Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
* Installed memory (RAM): 8.0 GB
* System type: 64-bit Operating System, x64-based processor

### **1.4.2 Install CouchDB on Windows**

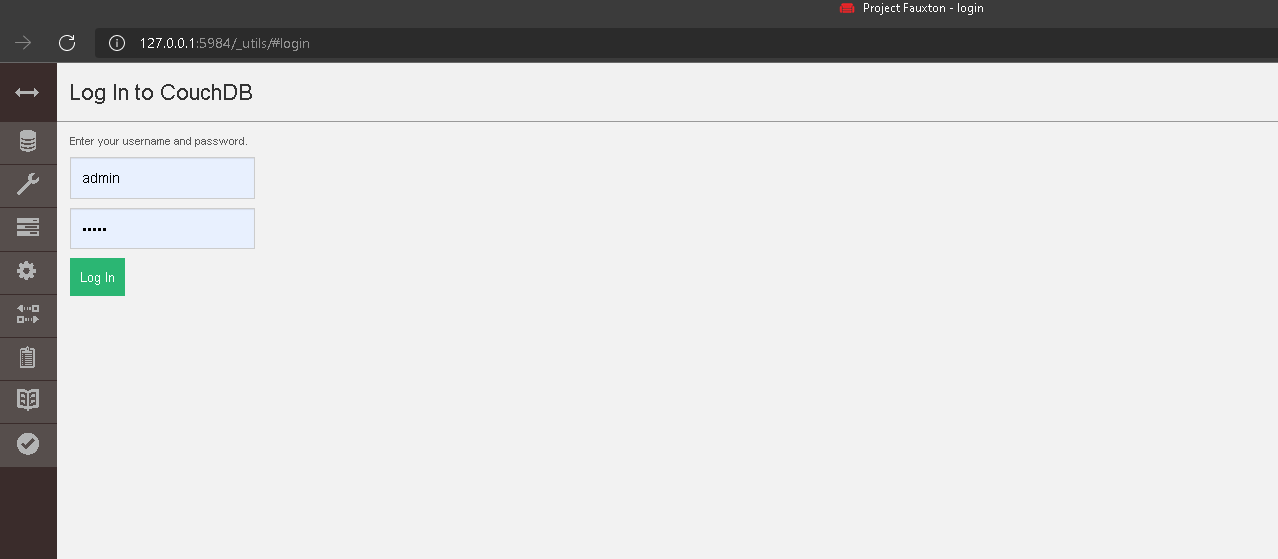
* Download CouchDB at link :<https://couchdb.apache.org/>.
* Then depending on the type of computer that chooses the version, here I choose the Windows version and proceed to install.



* Once installed, open CouchDB's built-in web interface by visiting http://127.0.0.1:5984/ .
* If all goes well, this will give you a website that has the following results.



* **-**
* You can interact with the CouchDB web interface using a urlhttp://127.0.0.1:5984/\_utils/



# **CHAPTER II : INTERFACE AND MANIPULATION IN COUCHDB**

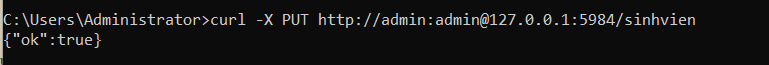
**Create and add new data**

CouchDB providestwo mechanisms of operation, curl (cmd) and futon (delivery).

## **Curl:**

**Syntax to create a database:**

curl -X PUT http://username:password@127.0.0.1:5984/database name



* Verify whether the database was created or not, by listing all the opportunities

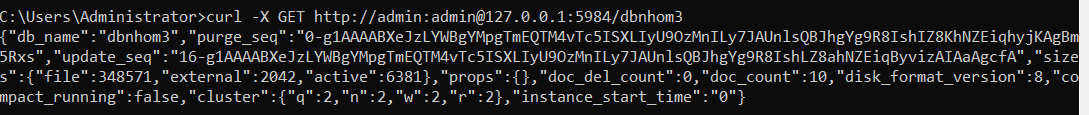
database with syntax:

curl -X GET http:// username:password@127.0.0.1:5984/\_all\_dbs

Ảnh có chứa văn bản, màu cam, tối  Mô tả được tạo tự động

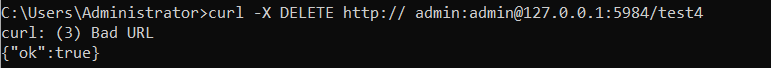
* View database information with the syntax:

curl -X GET http://username:password@127.0.0.1:5984/my\_database



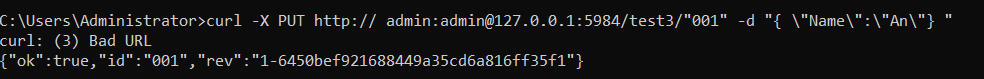
* Delete the data muscle with Curl

curl -X DELETE http:// username:password@127.0.0.1:5984/database name



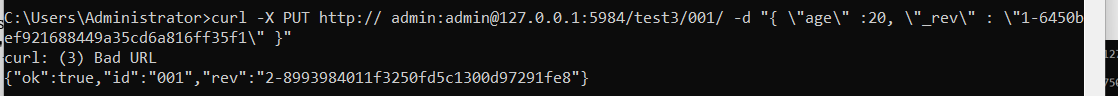
* Create a document with curl

curl -X PUT http:// username:password@127.0.0.1:5984/database name/"id" -d “{ document} “



* Update documents with curl

curl -X PUT http:// username:password@127.0.0.1:5984/database\_name/document\_id/ -d “{\"field\" : \"value\", \"\_rev\" : \"revision id\" }”



* Delete documents with curl

curl -X DELETE http : // username:password@127.0.0.1:5984 / database name/database id?\_rev id



## **Futon:**

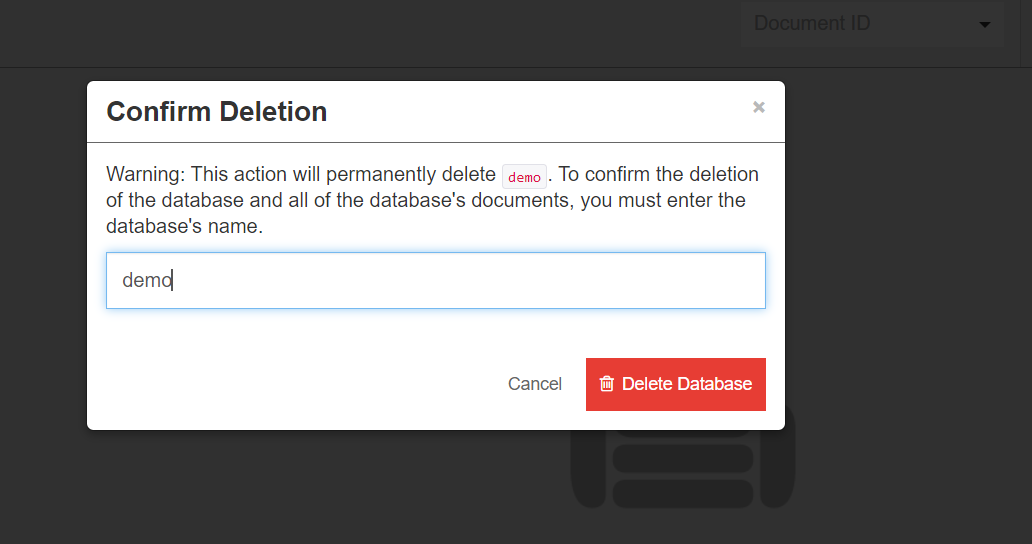
Create a database with Futon

Open **http://127.0.0.1:5984/utils/**

sign in username and password->act to Create Database -> type the name -> press create.

Graphical user interface, application

Description automatically generated

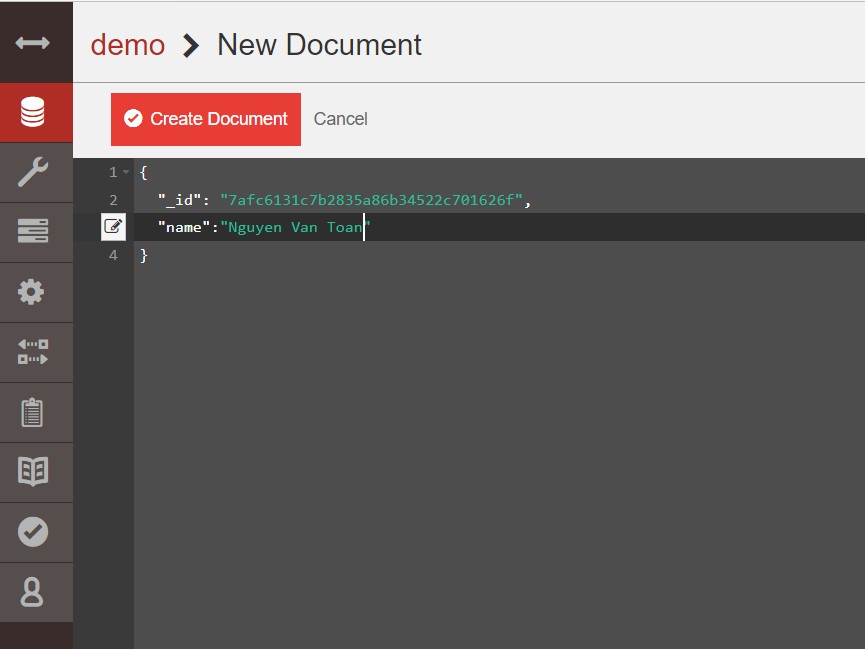


Delete data base with Futon:

Funnel chart

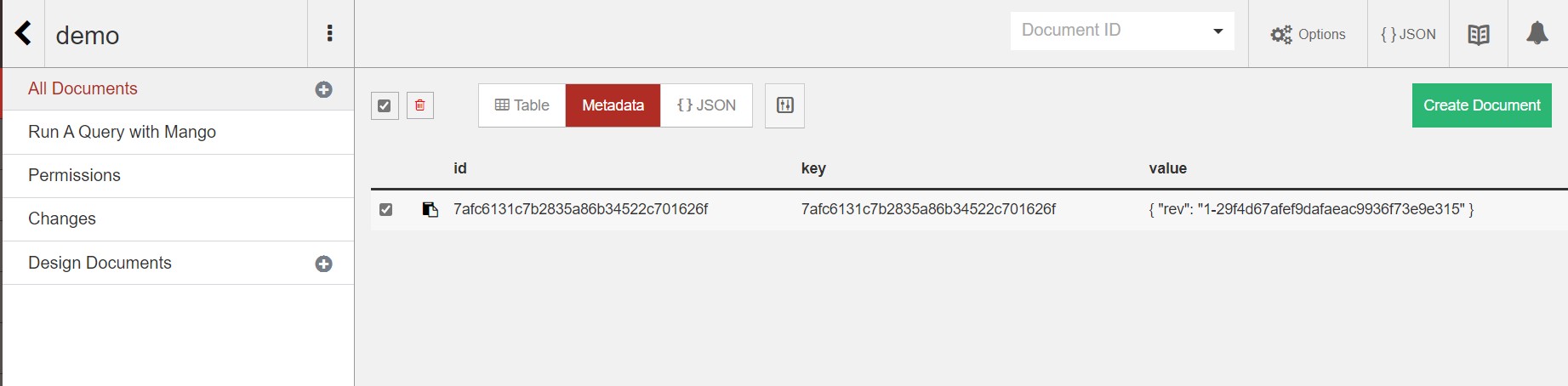
Description automatically generated with medium confidenceCreate a document using futon

Then enter the data in



Document update with futon: Similar to document creation step

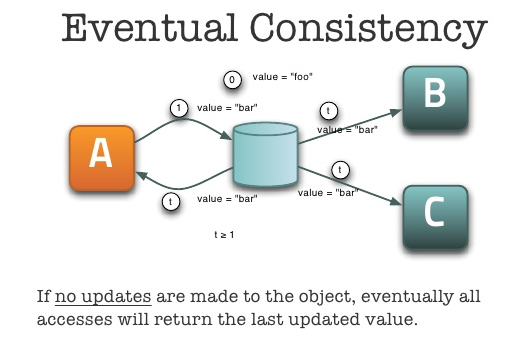
Delete the document with futon



# **CHAPTER III : DISTRIBUTION IN COUCHDB**

## **Distribution model**

* CouchDB uses eventual consistency model: The client can write data to a node without waiting for other nodes to agree. The change will then be sent to the remaining nodes and eventually the data between the nodes will be synchronized.



## **The CAP theorem**

* The CAP theorem describes a few different strategies for distributing application logic across networks. CouchDB’s solution uses replication to propagate application changes across participating nodes. This is a fundamentally different approach from consensus algorithms and relational databases, which operate at different intersections of consistency, availability, and partition tolerance.
* Consistency:

All database clients see the same data, even with concurrent updates.

* Availability:

All database clients are able to access some version of the data.

* Partition tolerance:

The database can be split over multiple servers.

Diagram, venn diagram

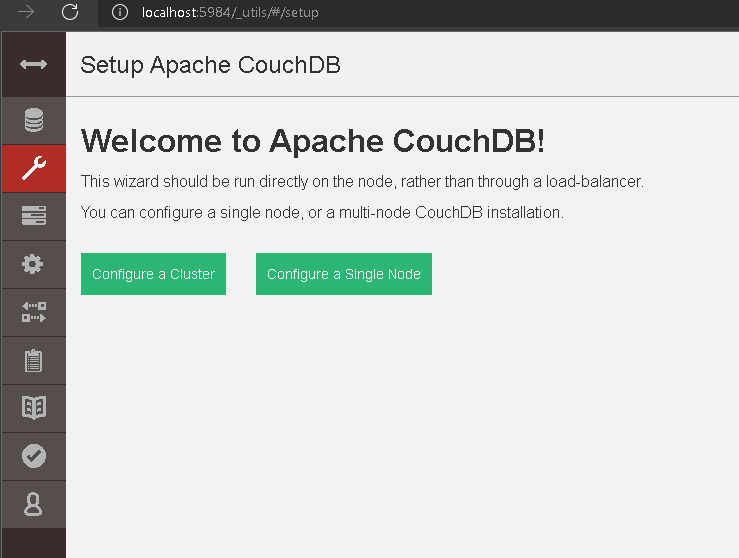
Description automatically generated

## **Data operation of 2 computers**

### **Single node setup:**

Enter localhost:5984/\_utils/#/setup.

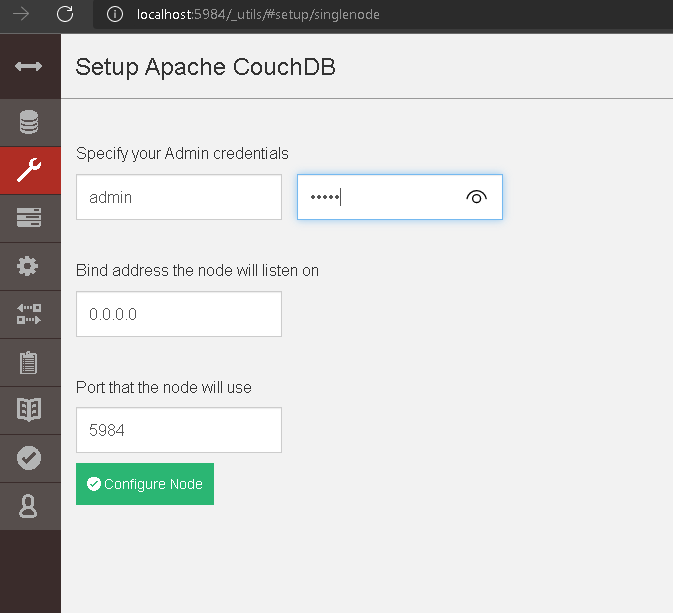
You'll be asked to set up CouchDB as a single-button version or set up a cluster of -> Click "Configure a Single Node".



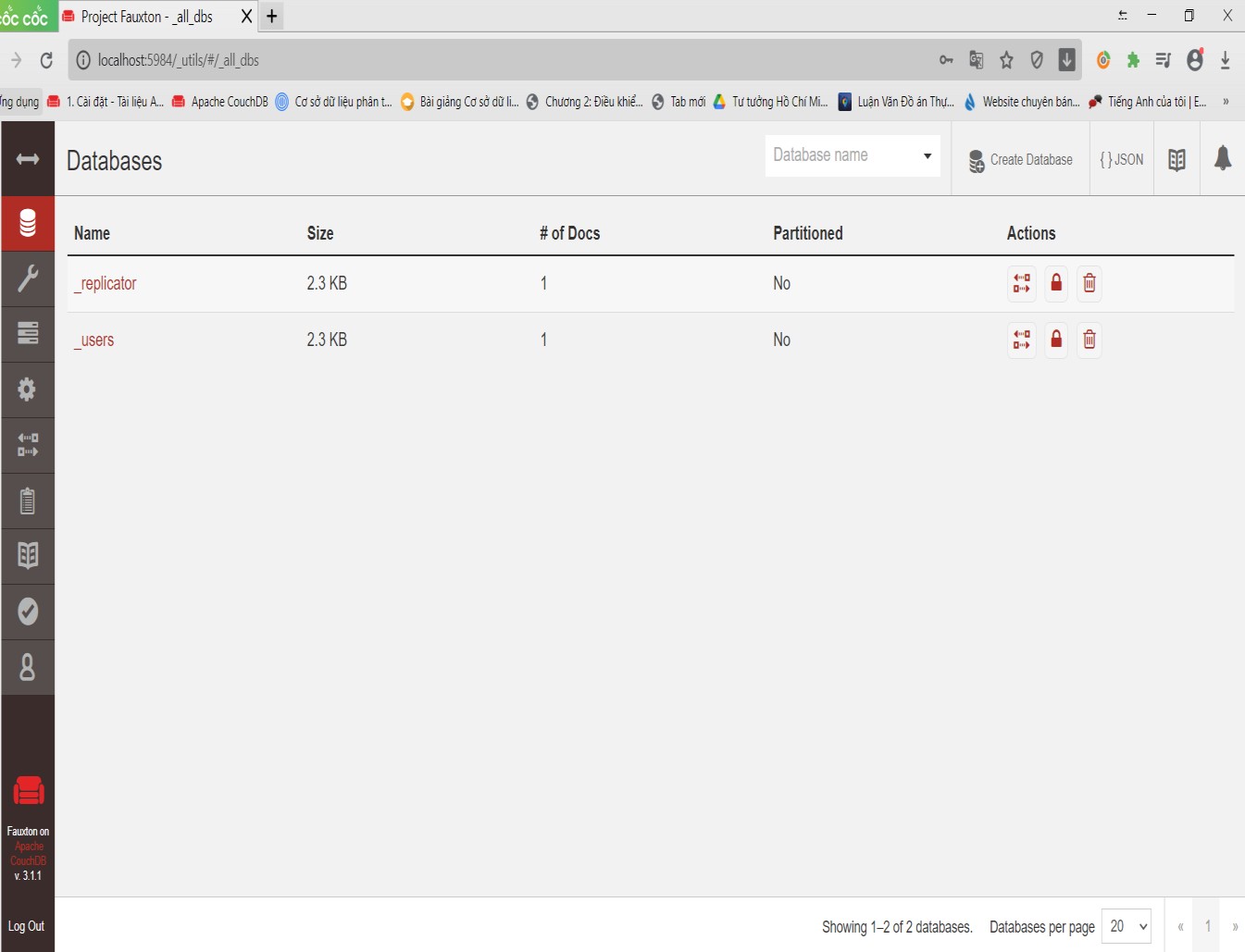
Enter the names tk and mk .

Bind address set to 0.0.0.0 .

Then click Configure Node.



In the database tab, there are 2 dbs: replica and user.

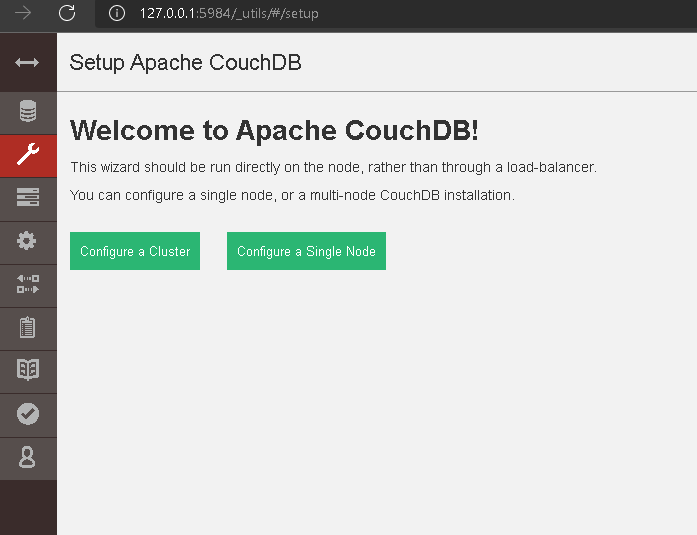


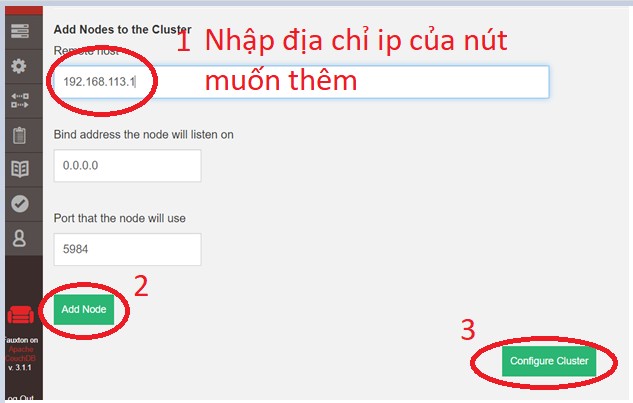
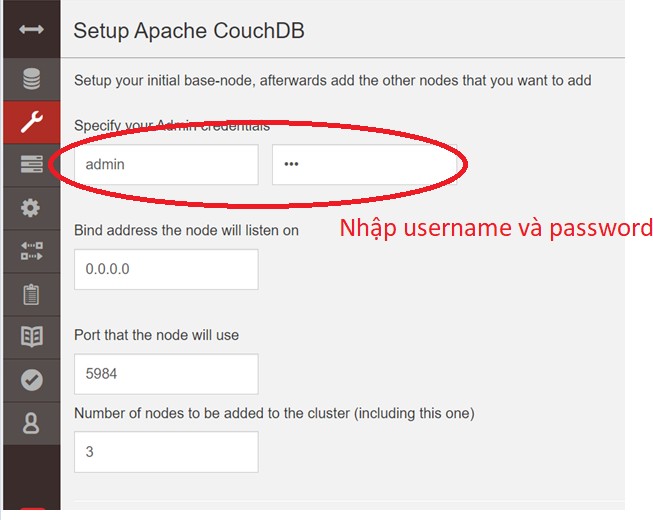
### **Cluster setup:**

At the Setup interface, select Configure a Cluster.

A screenshot of a computer

Description automatically generated with low confidence





Added a successful node.

***Get the data of the connected computer:***

Data retrieval syntax:

curl -X GET http://admin1:admin1@26.170.181.39:5984/tendb/\_all\_docs

curl -X PUT http://username:password@ipnode:5984/database name

Text

Description automatically generated

**See the databases of the node:**

curl -X GET http://username:password@ipnode:5984/\_all\_dbs

CouchDB

Text

Description automatically generated

**View database information:**

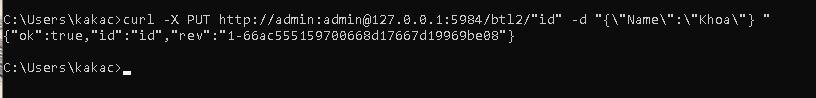
curl -X GET <http://username:password@ipnode:5984/_my_db>

Text

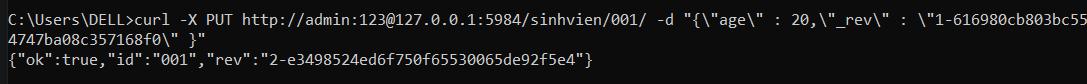
Description automatically generated

**Add a document on the node:**

curl -X PUT http://username:password@ipnode:5984/database name/"id" -d “{ document} “



**Update document on node:** curl -X PUT http://username:password@127.0.0.1:5984/database\_name/document\_id / -d "{ \"field\" : \"value\", \"\_rev\" : \"revision id\" }"

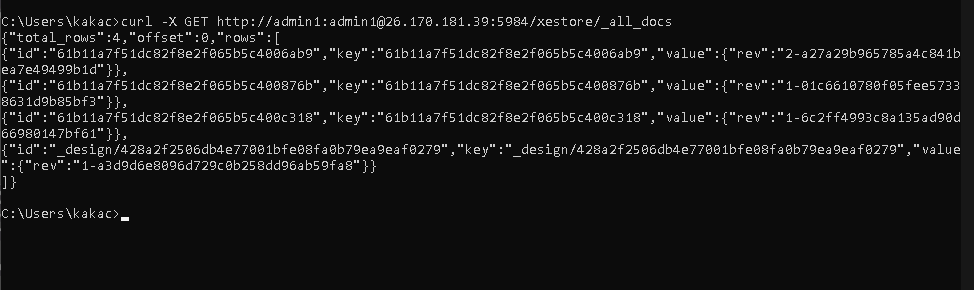


**Delete the document on the node:**

curl -X DELETE http://username:password@ipnode:5984 / database name/database id?\_rev=?

CouchDB





# **CHAPTER IV : REPLICATION IN COUCHDB**

## **Introduction to replication**

* One of CouchDB’s strengths is the ability to synchronize two copies of the same database. This enables users to distribute data across several nodes or data centers, but also to move data more closely to clients.
* Replication involves a source and a destination database, which can be on the same or on different CouchDB instances. The aim of replication is that at the end of the process, all active documents in the source database are also in the destination database and all documents that were deleted in the source database are also deleted in the destination database (if they even existed).

## **Scattered architecture with replication**

* Replication from Master to Slave computer
* Synchronize replication between 2 Master computers together
* Filter Replication.
* Incremental and 2-dimensional replication (only text updates change from 2 dimensions)
* Managing conflicts.

**Text, letter

Description automatically generated**

* Support the master-master replication, easy to clone, share, sync the data between database and pc.
* Master-master replication allow read and write data at once time in 2 server. At the moment, the 2 server are the master role and slave role.

**A picture containing chart

Description automatically generated**

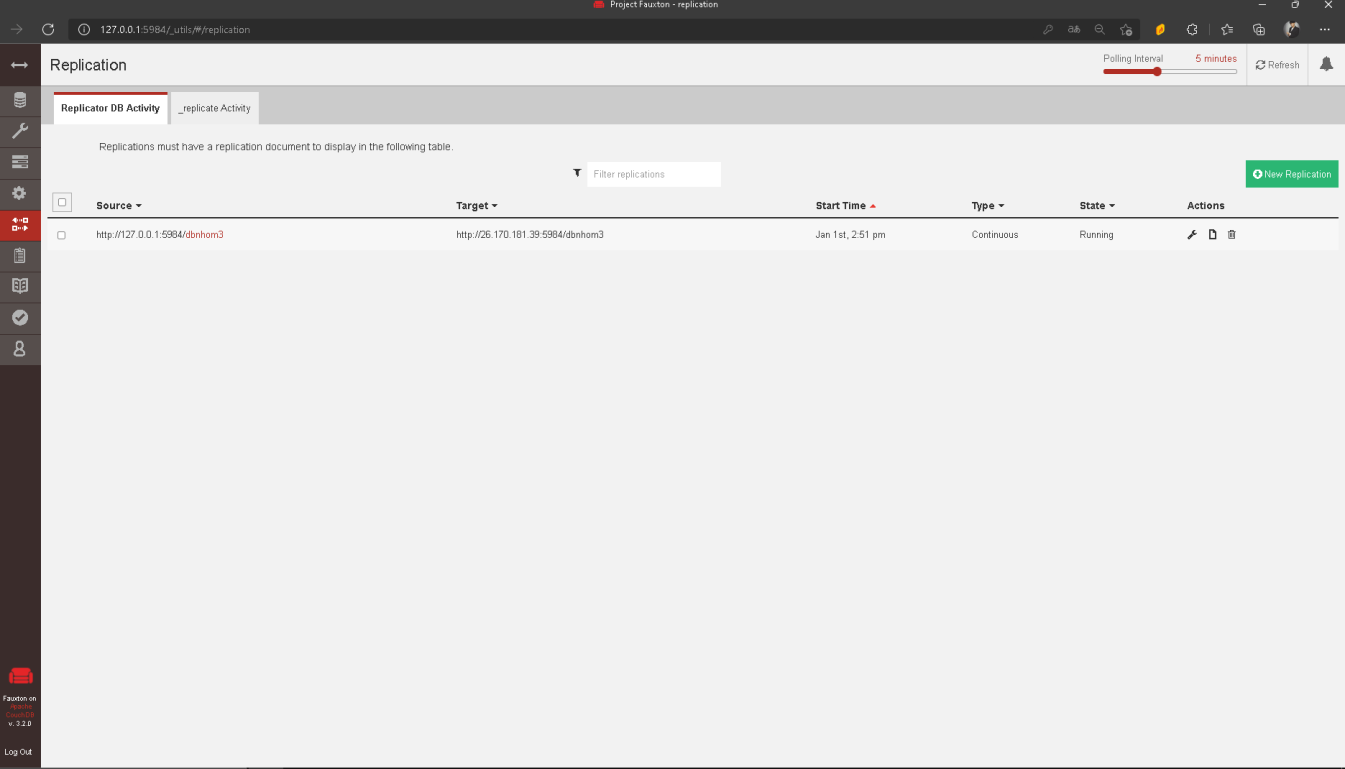
* CouchDB’s operations take place within the context of a single document. As CouchDB achieves eventual consistency between multiple databases by using incremental replication you no longer have to worry about your database servers being able to stay in constant communication. Incremental replication is a process where document changes are periodically copied between servers. We are able to build what’s known as a shared nothing cluster of databases where each node is independent and self-sufficient, leaving no single point of contention across the system.
* Support the master-master replication, easy to clone, share, sync the data between database and pc.
* Master-master replication allow read and write data at once time in 2 server. At the moment, the 2 server are the master role an slave role.

## **Replication mechanism**

***At the server (master):***

Enter the http://127.0.0.1:5984/\_utils/#/replication path

Will appear the interface as shown below, click new replication.



1. At Source: Type: local database Name: select database
2. At Target:

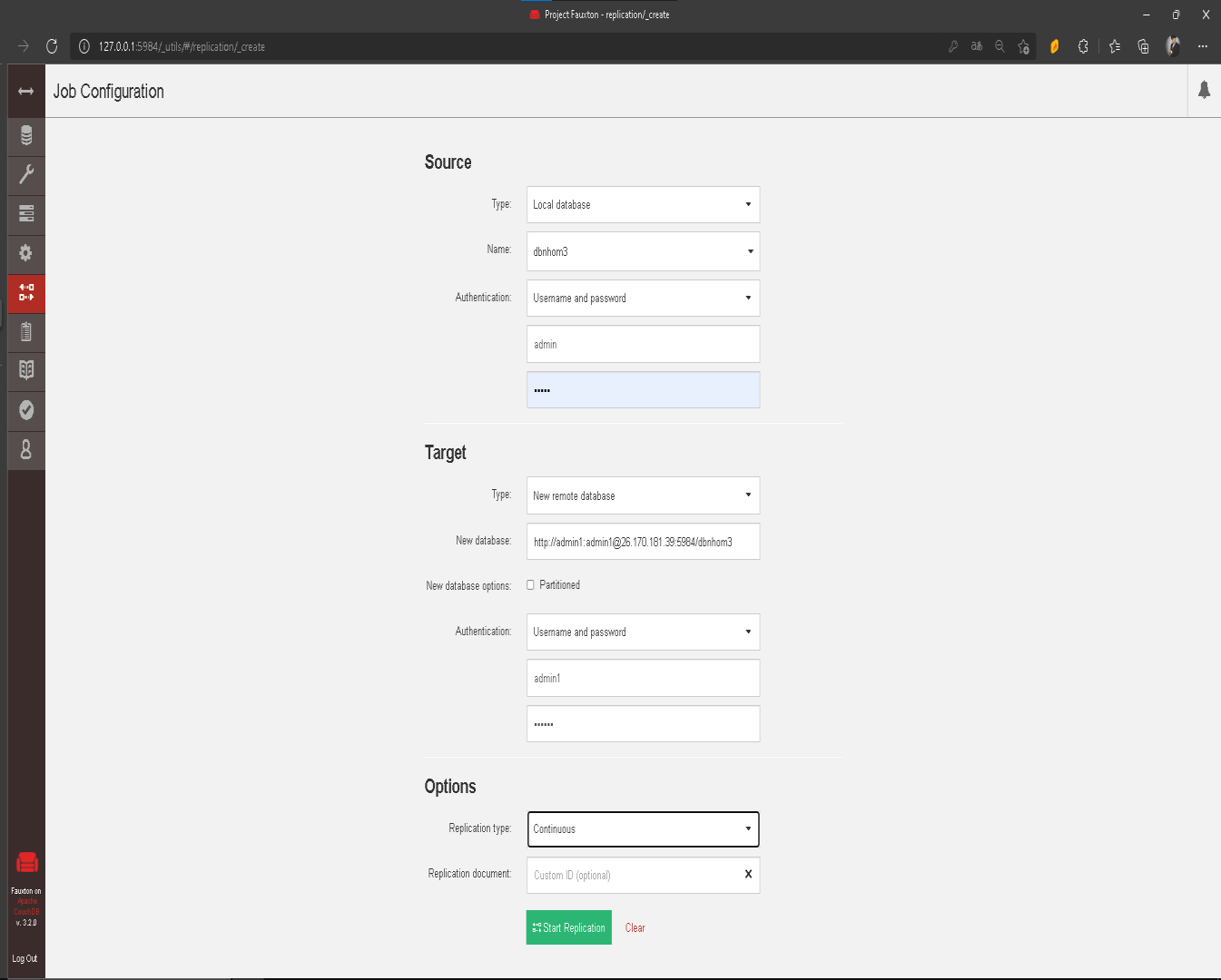
Type: New remote database

New db: install the path to the replicating computer

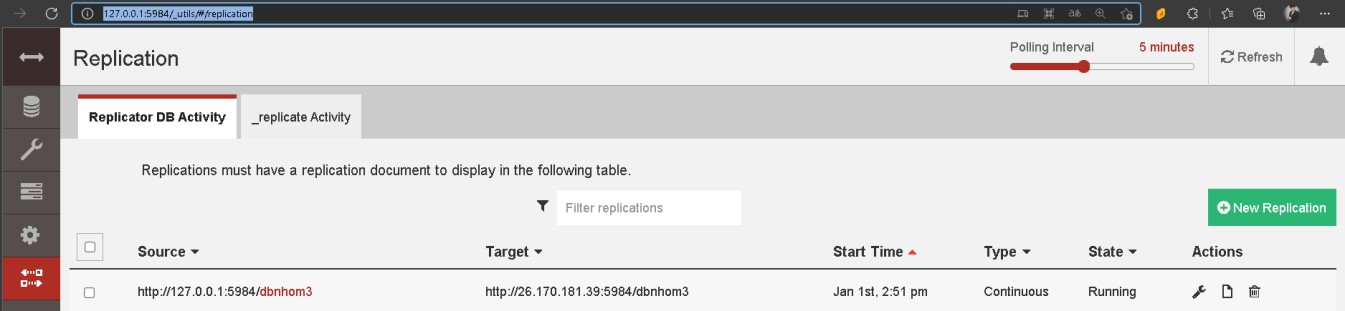
1. At Option:

Rep type: continues so that the data is synchronized after the update.

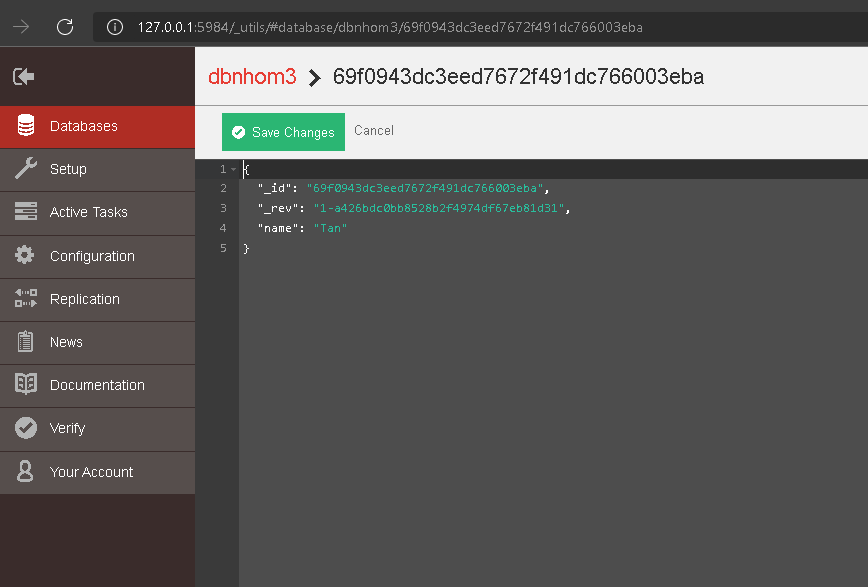
1. Here we clone 1 db named dbnhom3 into another computer.



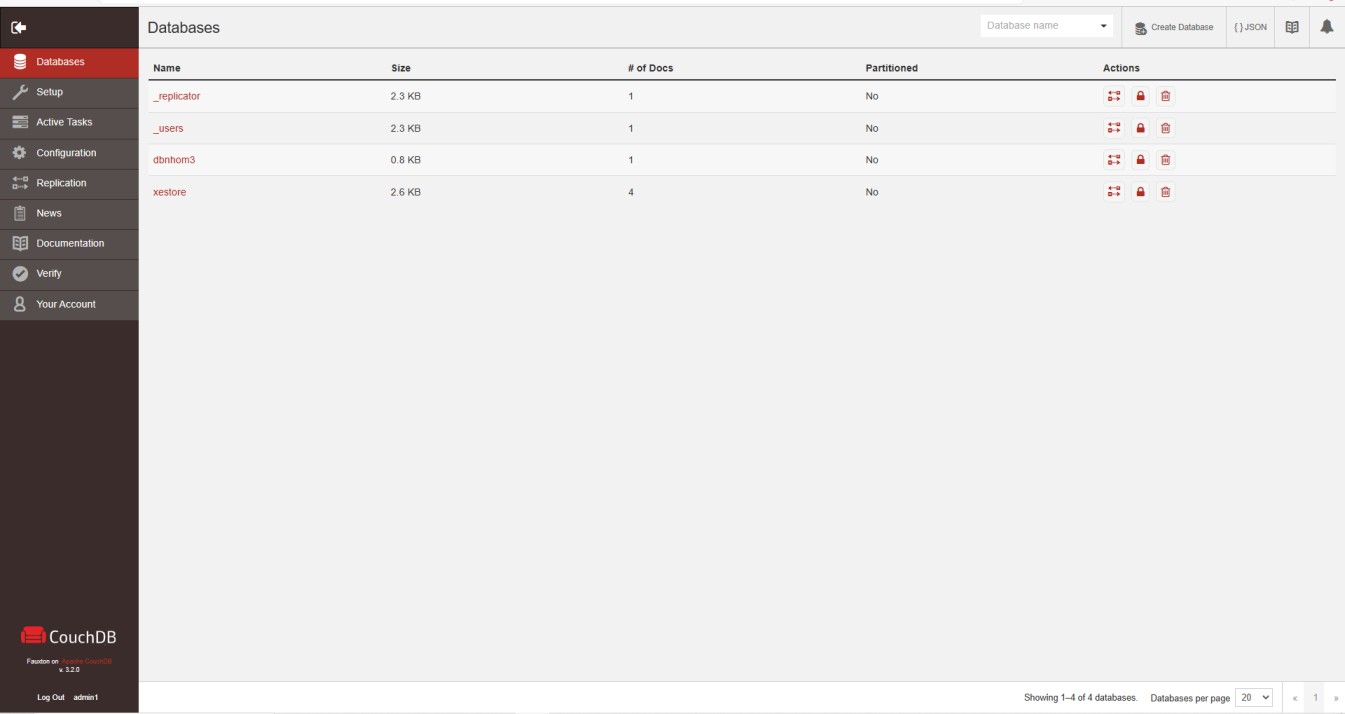
Successful replicating.



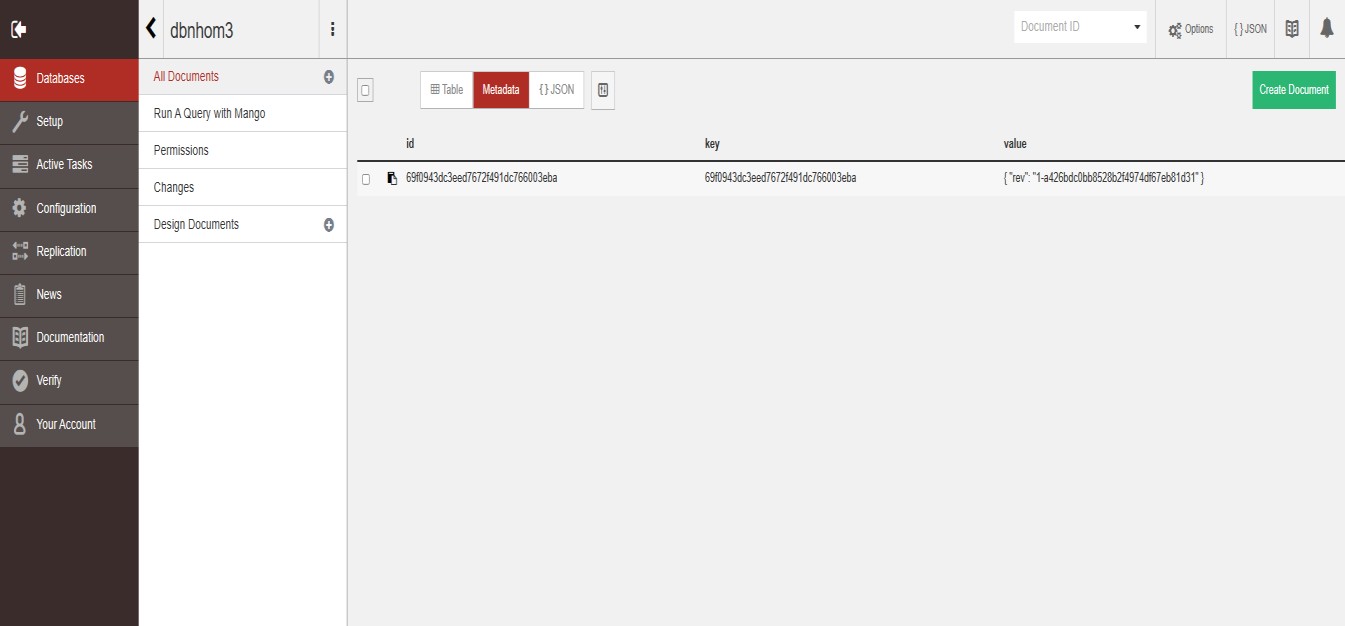
Proceed to add 1 doc to db after replicating



And at the cloned computer: dbnhom3- database is replicated.



In db ap



1. The data in db clone is one-dimensional, only the server is entitled to edit, add, delete synchronously to the slave computer. When the slave computer performs more repair deletion, the master computer is not in sync. To synchronize, it is necessary to replicate the reverse master computer.

# **REFERENCE**

Slide Database management systems of University of Information Technology – VNU HCMC.

Website official of CouchDB [Eventual Consistency (couchdb.org)](https://guide.couchdb.org/draft/consistency.html).

Youtube CouchDB Replication [BZAN 6356 Lecture 8.3: Replication in CouchDB - YouTube](https://www.youtube.com/watch?v=U6SlST9tQ6k)

Orginal guide: [2.1. Introduction to Replication — Apache CouchDB® 3.2 Documentation](https://docs.couchdb.org/en/3.2.0/replication/intro.html#:~:text=3.-,Replication%20Procedure,the%20documents%20to%20the%20destination.)

Link demo : https://drive.google.com/drive/folders/12disnU4UIytu4Cy2CZC35LuclMvgGKG0?usp=sharing

# **The end.**