# The Installation and Configuration of the Couchdb

## Opening port for external web server access

Couchdb is one kind of NoSQL database systems (DBS) for distributed system as well as abilities of this database are aligned with project’s requirements because of global twittering analysis and distribution-based architecture. Based on this conceptual idea, couchdb can be implemented in this project and installed in a virtual machine by the command line tool. Also, the couchdb cannot be accessed by external server so the bind\_address is required to modify to 0.0.0.0 for client access. Regarding to the couchdb configuration file for modifications of bind\_address, there are two files, storing in etc directory: local and default.ini files. The local file needs to be configured only. After that, the couchdb is restarted for acquiring new configuration so the cocudb can be accessed by external connection. The diagram 1.1 illustrates the installation of the couchdb.

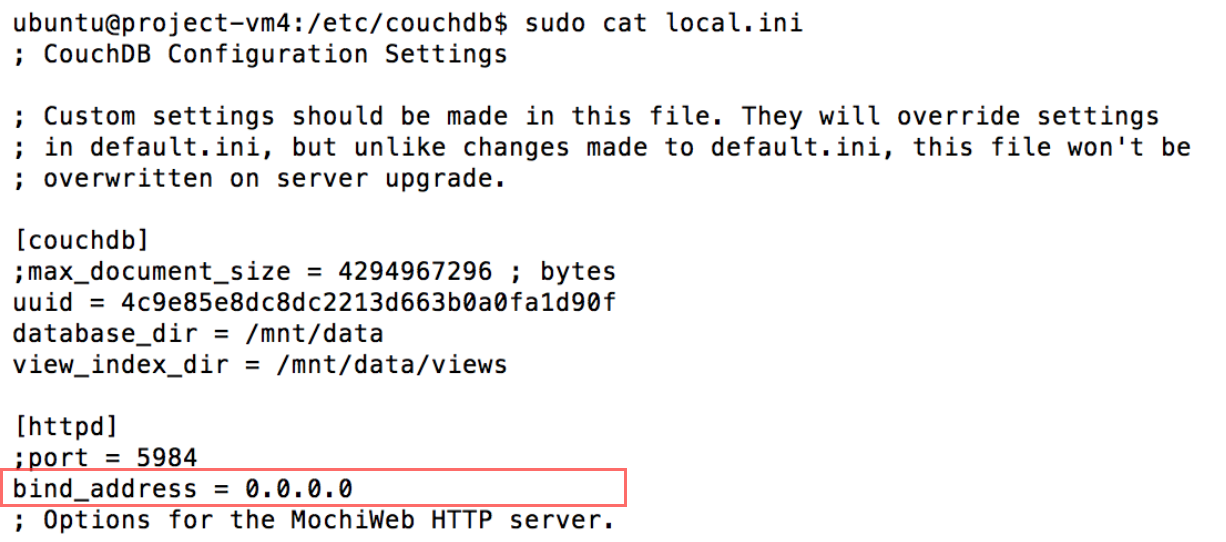


Diagram 1.1: Changing bind\_address for client access.

## Accessing to the Couchdb

According to the REST approach, the request is communicated by HTTP of GET,POST,PUT,DELETE [1]. The Couchdb supports REST protocol so that executing “curl –X Get <http://yourdatabaseip:5984/_all_dbs>” to the command line is able to gain all of database systems in the couchdb. Another approach the database information can be acquired is that using web browser with <http://115.146.95.53:5984/_utils/>, as a Futon webpage, can retrieve all of database information and verify whether the couchdb is working or not. The diagram 1.2 and 1.3 depict the process of gathering information from the Couchdb.



Diagram 1.2: A command script to check all of DBS in the couchdb

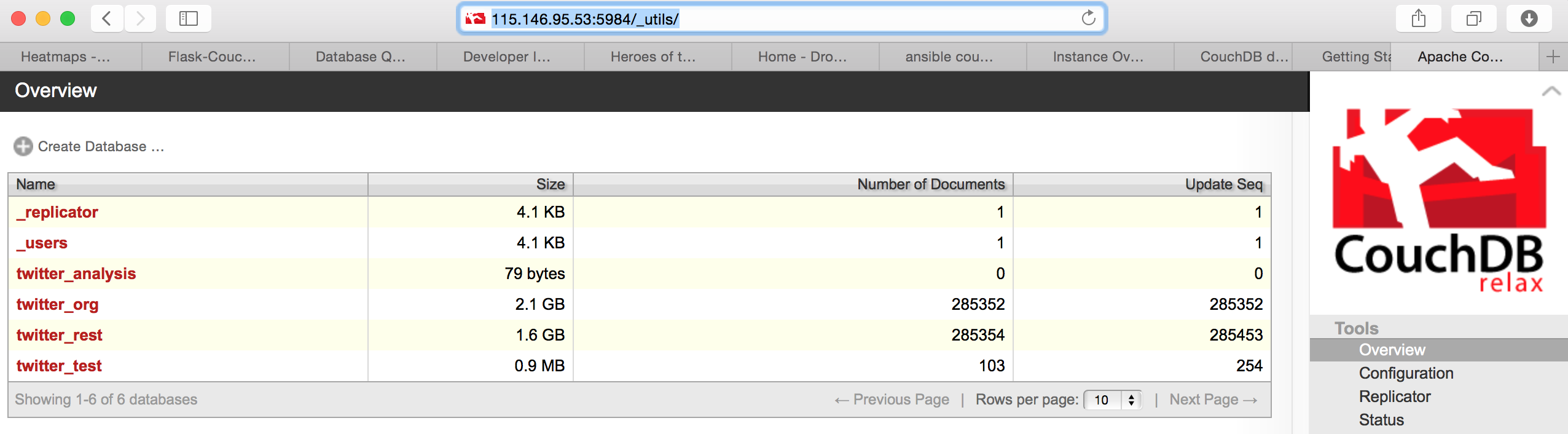


Diagram 1.3: Futon webpage for verifying all of DbS in the couchdb

## The attachment of volumes

The system also attaches a volume to the DBS because it may be increased incrementally due to inserting the tweet data on a daily basis for purposes of data and sentiment analysis. Based on the NeCTAR research cloud service, the volume can be created as well as attached to the target the VM; however, this attachment has a limitation that is only for the same availability zone. The availability zone of the VM and volumes are created in the same zone because of that limitation. Once the volume has been attached to the VM, the volume need to be configured to connect to the location of the database files by command. By doing this implementation, the volume has been attached to VM finally. The diagram 1.4 shows a volume has been attached in couchdb partition.

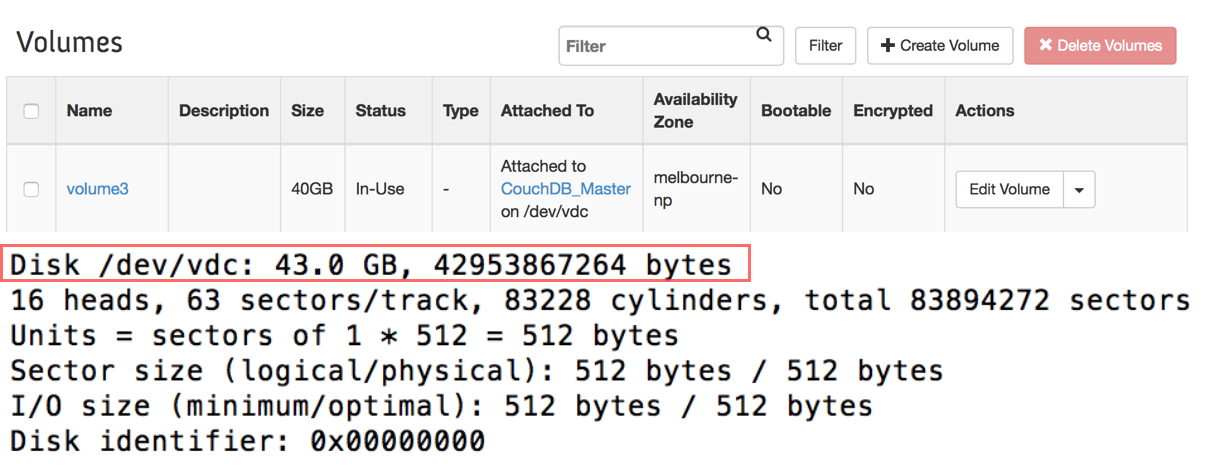


Diagram 1.4: The volume for the DBS partition

## Configurations of the replication

## Replication is essential for data backup as well as avoids accidental disaster. This disaster recovery in Couchdb can be implemented by web configuration as a continuous mode so that this recovery is executed constantly. The tweet data is replicated to another couchdb as a backup database. This plan provides an alternative approach that the couchdb is crashed by unknown reasons and unable to recover. By doing this implementation, the users have a alternative database for performing the daily job. The diagram 1.5 depicts the status of this recovery.





Diagram 1.5: the status of the replication

[1] http://www.ibm.com/developerworks/library/ws-restful/