Docker depends on having a 64 bit kernel having the kernel module for handling containers. Docker is just a shell around that kernel module.

Follow the commands to set up Docker

Ready to build containers over it

sudo tee /etc/yum.repos.d/docker.repo <<-'EOF'

[dockerrepo]

name=Docker Repository

baseurl=https://yum.dockerproject.org/repo/main/centos/$releasever/

enabled=1

gpgcheck=1

gpgkey=https://yum.dockerproject.org/gpg

EOF

9 sudo yum install docker-engine

10 sudo service docker start

11 sudo docker run hello-world

12 docker version

13 sudo docker version

14 sudo usermod -aG docker ec2-user

15 exit

16 docker run hello-world

17 sudo chkconfig docker on

39 docker run -d -p 8091-8093:8091-8093 -p 11210:11210 couchbase

40 docker image

41 docker images

42 docker ps -a

43 sudo service docker start

44 service iptables status

45 chkconfig iptables off

46 sudo service docker start

47 docker images 48 docker ps -a

50 mkdir Node

51 cd Node

52 vi Dockerfile

126 docker ps

127 docker ps -a

128 docker images

129 ls

130 rm Dockerfile

131 vi dockerfile

132 docker build do/me .

133 docker build -t do/me .

134 vi dockerfile

135 docker build -t do/me .

136 docker run 72d7ff0207bc

137 docker run fe40b2695981

138 docker images

139 docker run -t -i do/me:latest /bin/bash

140 history

Couchbase Server is an open source distributed NoSQL document database that provides low latency data management for large scale, interactive online applications. It is designed to scale, especially horizontally, with ease and without performance degradation. Built with a strong emphasis on reliability, high availability, and simple management, Couchbase serves data non-stop with minimal human intervention.

Couchbase Server is a multi model, general purpose database that can be used as a managed cache tier, a key-value store, and a document database. Extending to embedded environments, Couchbase Lite runs native on device in mobile and IoT environments with managed sync.

**##Then follow this link to download couchbase servr**

http://developer.couchbase.com/documentation/server/4.1/getting-started/installing.html

**##This is the docker file needed to be made**

https://hub.docker.com/r/tleyden5iwx/couchbase-server-3.0.1/~/dockerfile/

**##The following link shows how is Docker file useful**

https://www.digitalocean.com/community/tutorials/docker-explained-using-dockerfiles-to-automate-building-of-images

**##This command can be used to get couchbase**

docker run -d -p 8091-8093:8091-8093 -p 11210:11210 couchbase

---and to open it in browser (public dns:8091)

**##FROM-http://blog.couchbase.com/2016/february/couchbase-docker-container**

**##Please note that you have to update your firewall configuration to**

allow connections to the following ports: 11211, 11210, 11209, 4369,

8091, 8092, 18091, 18092, 11214, 11215 and from 21100 to 21299.

**##Here are the commands to be kept in the docker file of couchbase**

FROM couchbase

COPY configure-cluster.sh /opt/couchbase

CMD ["/opt/couchbase/configure-cluster.sh"]

To set up Couchbase Server in a nonproduction environment, you can accept the default values provided on most of the set-up screens.

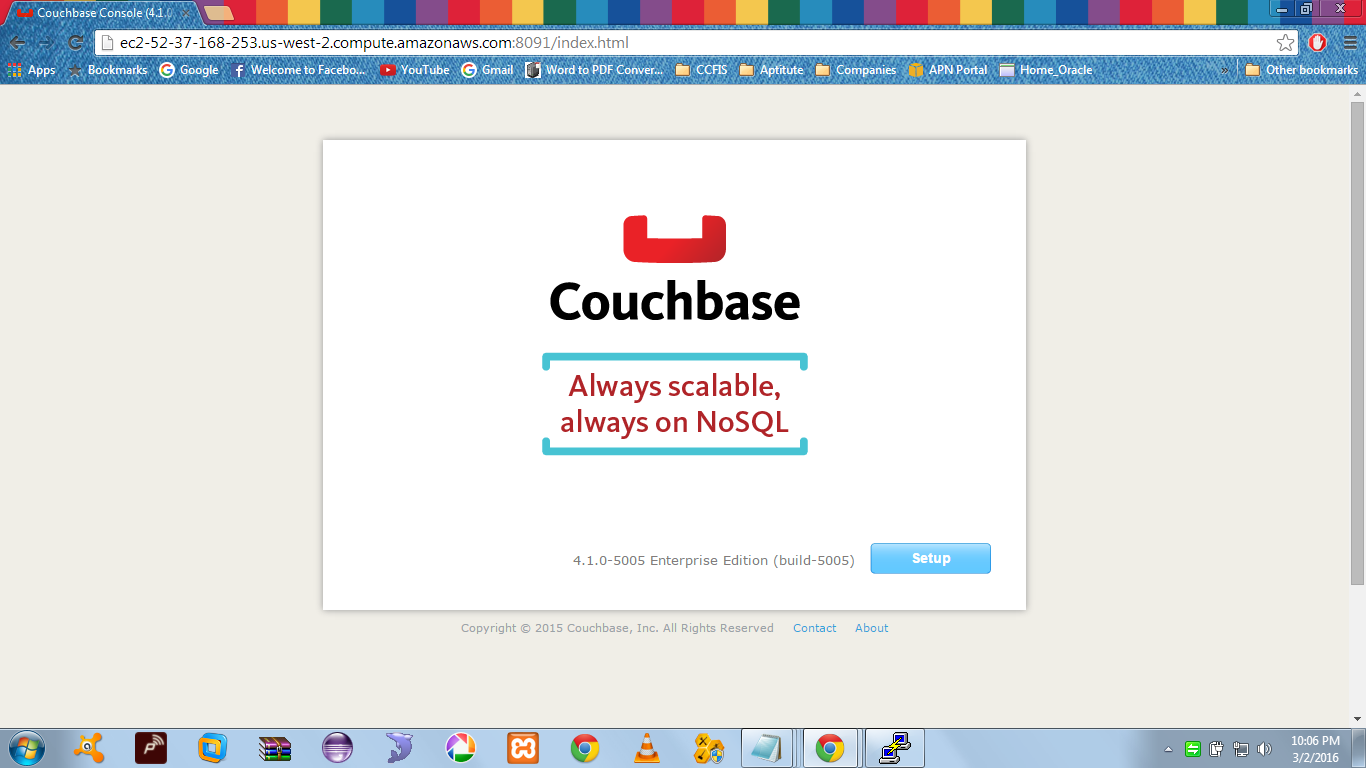
Couchbase Server is a distributed NoSQL database engineered for performance, scalability, and availability. It enables developers to build applications easier and faster by leveraging the power of SQL with the flexibility of JSON.

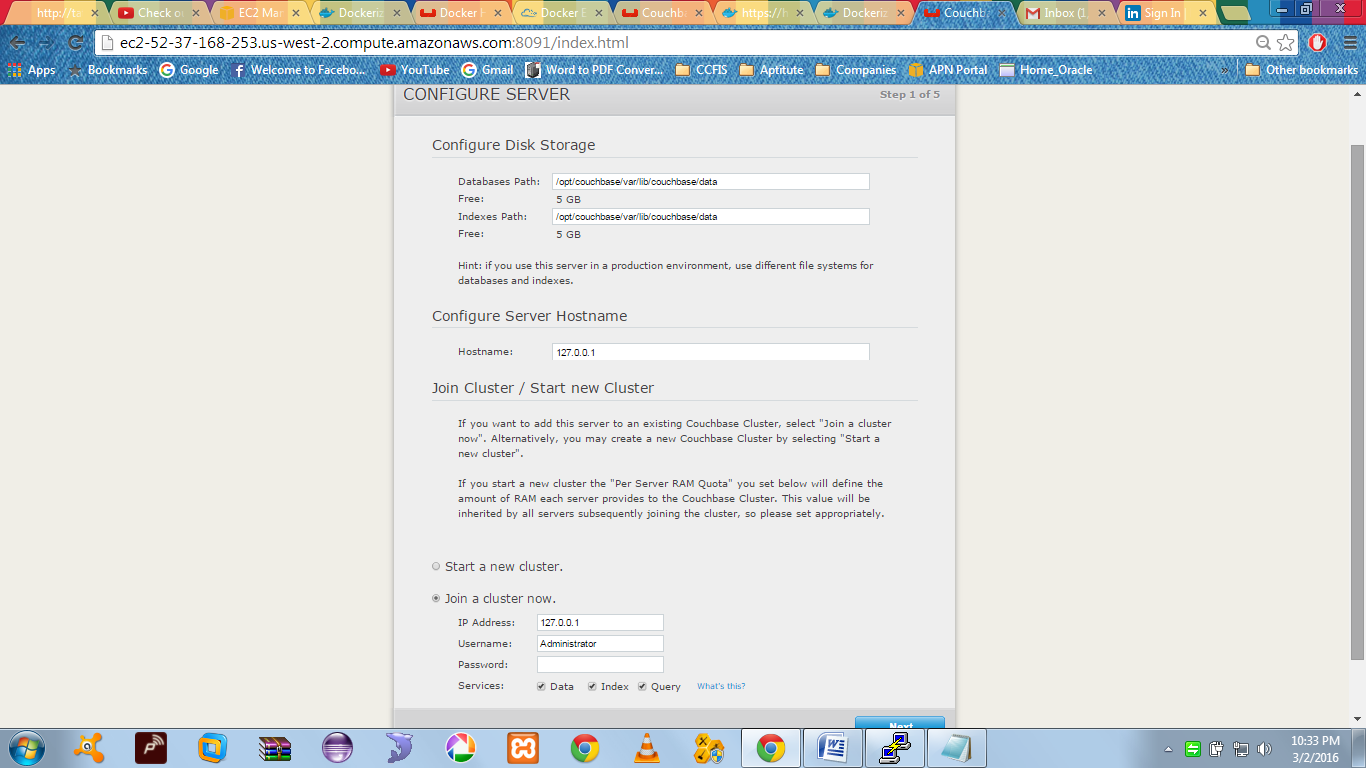
1. Open a browser and navigate to http://*hostname*:8091/.
2. In the URL, *hostname* represents the name or IP address of the computer that hosts Couchbase Server. If Couchbase Server is running locally, enter localhost for the host name.
3. Click **Setup**.
4. On the Configure Server screen (Step 1 of 5), click **Next** to accept the default values for a new cluster.
5. On the Sample Buckets screen (Step 2 of 5), under Available Samples select the two samples we will use later in this tutorial: **beer-sample** and **travel-sample** and click **Next**.
6. On the Create Default Bucket screen (Step 3 of 5), under Memory Size set the Per Node RAM Quota to 100 MB and click **Next**.

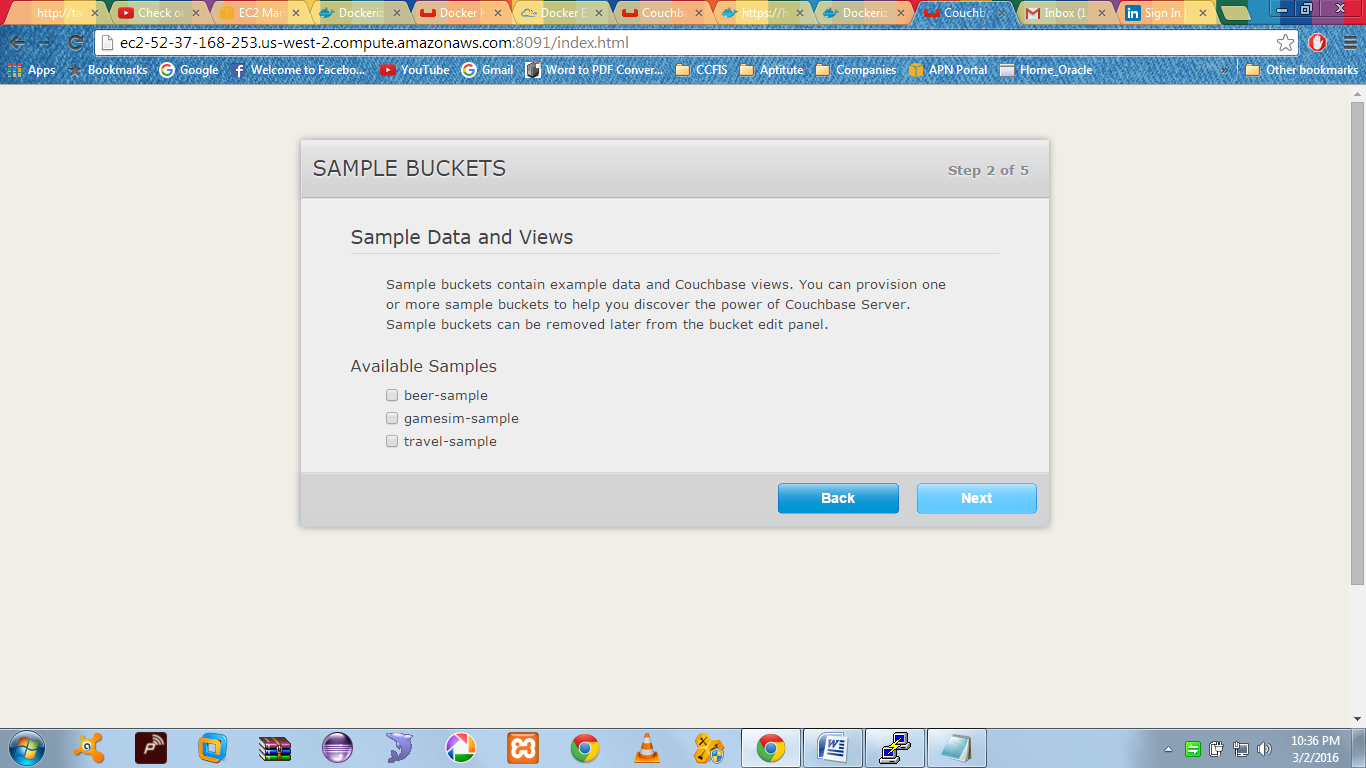
**(Have to give min 400 mb to data cluster ram so as to get proper per node division )**

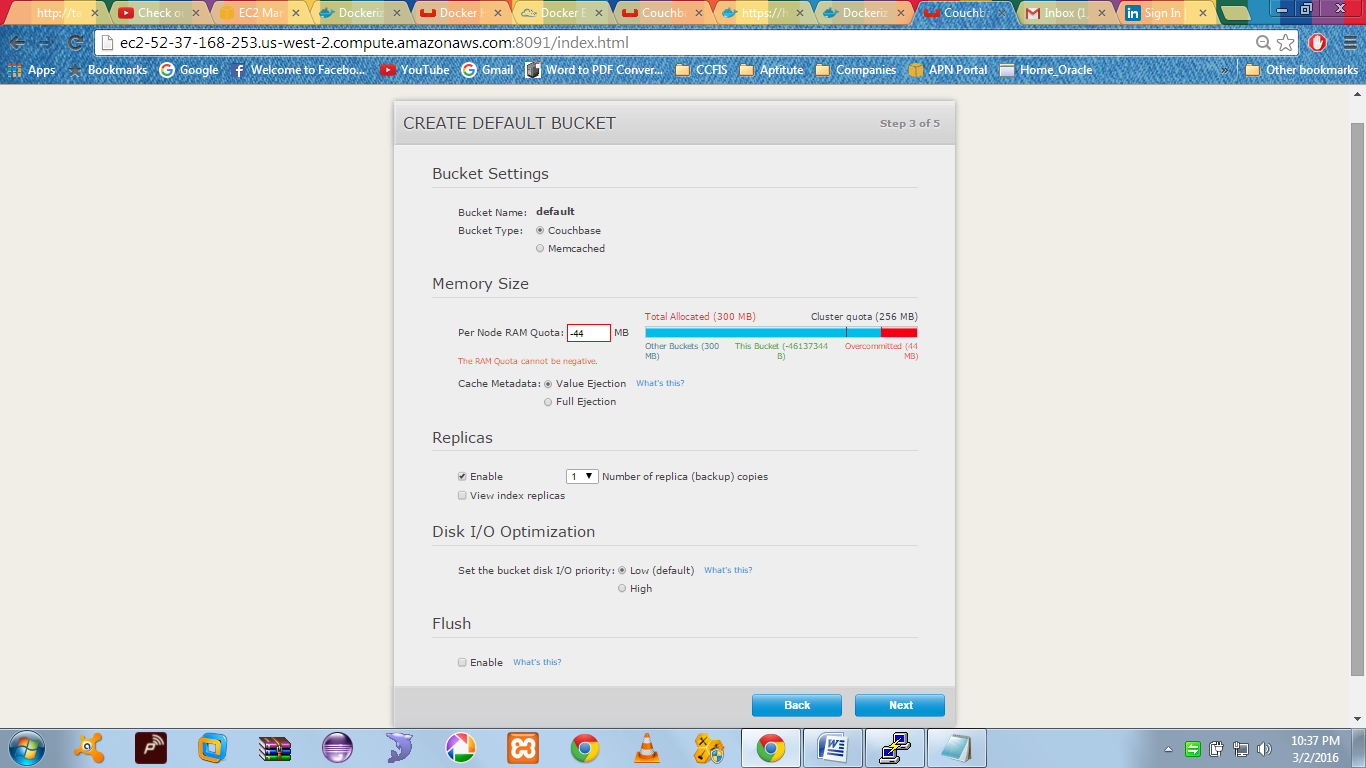
1. On the Notifications screen (Step 4 of 5), enter your registration information, agree to the terms and conditions, and click **Next**.
2. On the Configure Server screen (Step 5 of 5), enter and verify a password for the administrator account, and click**Next**.

The Couchbase Web Console opens and displays the Cluster Overview.









Trying to build a sample app over couchbase

<http://developer.couchbase.com/documentation/server/4.1/travel-app/index.html>

<https://hub.docker.com/_/node/>

**Dockerfile to download node.js**

|  |
| --- |
| FROM buildpack-deps:wheezy |
|  |  |
|  | # gpg keys listed at https://github.com/nodejs/node |
|  | RUN set -ex \ |
|  | && for key in \ |
|  | 9554F04D7259F04124DE6B476D5A82AC7E37093B \ |
|  | 94AE36675C464D64BAFA68DD7434390BDBE9B9C5 \ |
|  | 0034A06D9D9B0064CE8ADF6BF1747F4AD2306D93 \ |
|  | FD3A5288F042B6850C66B31F09FE44734EB7990E \ |
|  | 71DCFD284A79C3B38668286BC97EC7A07EDE3FC1 \ |
|  | DD8F2338BAE7501E3DD5AC78C273792F7D83545D \ |
|  | B9AE9905FFD7803F25714661B63B535A4C206CA9 \ |
|  | C4F0DFFF4E8C1A8236409D08E73BC641CC11F4C8 \ |
|  | ; do \ |
|  | gpg --keyserver ha.pool.sks-keyservers.net --recv-keys "$key"; \ |
|  | done |
|  |  |
|  | ENV NPM\_CONFIG\_LOGLEVEL info |
|  | ENV NODE\_VERSION 5.7.1 |
|  |  |
|  | RUN curl -SLO "https://nodejs.org/dist/v$NODE\_VERSION/node-v$NODE\_VERSION-linux-x64.tar.xz" \ |
|  | && curl -SLO "https://nodejs.org/dist/v$NODE\_VERSION/SHASUMS256.txt.asc" \ |
|  | && gpg --batch --decrypt --output SHASUMS256.txt SHASUMS256.txt.asc \ |
|  | && grep " node-v$NODE\_VERSION-linux-x64.tar.xz\$" SHASUMS256.txt | sha256sum -c - \ |
|  | && tar -xJf "node-v$NODE\_VERSION-linux-x64.tar.xz" -C /usr/local --strip-components=1 \ |
|  | && rm "node-v$NODE\_VERSION-linux-x64.tar.xz" SHASUMS256.txt.asc SHASUMS256.txt |
|  |  |
|  | CMD [ "node" ] |

| **What's hard for a relational database...** | **...is easy for Couchbase Server.** |
| --- | --- |
| Read and write data online and offline: Relational databases do not offer an easy solution that supports seamless offline and online operation | Couchbase Lite, an embedded NoSQL database, works in conjunction with Couchbase Server and Sync Gateway to ensure data is always accessible, with or without a network connection |
| Enable rapid development and enhancement of mobile apps:Relational databases, with their rigid schemas that require rewriting an app when there are changes to the data model, slow down the app development and update process | With its flexible JSON data model and pre-built synchronization via Sync Gateway, Couchbase Mobile is engineered to support rapid and easy app development and fast time to market |
| Support multiple device types and platforms: With relational databases, developing apps that support multiple devices and platforms is highly complex | Couchbase Mobile natively supports all the major mobile platforms (iOS, Android, .NET) and its flexible JSON data model makes it easy to support any new data type without having to rewrite the app |

**About Node.js**

Node.js is a software platform for scalable server-side and networking applications. Node.js applications are written in JavaScript and can be run within the Node.js runtime on Mac OS X, Windows, and Linux without changes.

Node.js applications are designed to maximize throughput and efficiency, using non-blocking I/O and asynchronous events. Node.js applications run single-threaded, although Node.js uses multiple threads for file and network events. Node.js is commonly used for real-time applications due to its asynchronous nature.

Node.js internally uses the Google V8 JavaScript engine to execute code; a large percentage of the basic modules are written in JavaScript. Node.js contains a built-in, asynchronous I/O library for file, socket, and HTTP communication. The HTTP and socket support allows Node.js to act as a web server without additional software such as Apache.

**Sync Gateway:**

**Maintains up-to-date copies of documents** where users need them: on mobile devices for instant access and on servers in data centers for reasons such as synchronizing documents, sharing documents, and loss-protection. Mobile apps create, update, and delete files locally. Sync Gateway takes care of the rest.

**Provides access control,** ensuring that users can only access documents to which they should have access.

**Ensures that only relevant documents are synced.** Sync Gateway accomplishes this by examining documents and by applying business logic to decide whether to assign the documents to channels.

Access control and ensuring that only relevant documents are synced are **achieved through use of channels and the sync function.**

Command to pull the repo and get sync gateway is :

docker run -p 4984:4984 -p 4985:4985 couchbase/sync-gateway http://git.io/vfQpe