

A Guide to CouchDB Installation Configuration and Monitoring

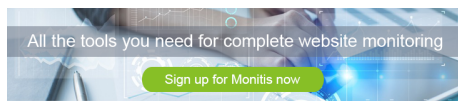
<https://www.monitis.com/blog/a-guide-to-couchdb-installation-configuration-and-monitoring/>

Introduction

There is huge amount of documentation about [CouchDB](#) on the web. This article is not a complete guide to CouchDB. It's just a very brief description of CouchDB with examples which could be useful if you decide to try cluster building.

A little about CouchDB:

Basically, CouchDB is a schema-less document oriented database with a lot of features:



- *Document Storage*
- ACID Semantics
- Map/Reduce Views and Indexes (JavaScript based)
- Distributed Architecture with Replication
- REST API
- Eventual Consistency
- *Mobile access*

It also has a great web based management console called Futon.

Installation

The easiest way to install CouchDB is to use apt-get: *apt-get install couchdb*.

Notice that the packages in the repositories are not always the latest version. So, when we install some packages with apt-get, we risk losing new features, bug fixes and improvements.

Installation from sources is not the best way to install, however in this case we'll install CouchDB in this way.

The latest stable version for Linux is available at: <http://www.apache.org/dyn/closer.cgi?path=/couchdb/releases/1.2.0/apache-couchdb-1.2.0.tar.gz>

Select your mirror and proceed!

Before installation, we have to install some dependency packages to make CouchDB work properly.

```
add-apt-repository ppa:launchpad/ppa #add the proper repository for libmozjs-dev and other packages
```

```
#install the dependencies
```

```
apt-get install libmozjs-dev
```

```
apt-get install libicu-dev
```

```
apt-get install libcurl4-openssl-dev
```

```
apt-get install build-essential erlang
```

```
# After all the dependencies successfully install, it is time to define where we will install from.
```

```
# There are 2 options
```

```
# The SVN repository or the last stable version from the official mirror.
```

```
# However, if you want to try installation from the svn repo, relax and type the commands below, otherwise do
```

```
wget -chttp://couchmirrorpath/<couch-version>.tar.gz
```

```
# cd <couch version>
```

```
# ./configure && make && sudo make install
```

```
svn co http://svn.apache.org/repos/asf/couchdb/trunk  
couchdb
```

```
cd couchdb
```

```
./bootstrap
```

```
./configure
```

```
make
```

```
sudo make install
```

```
make clean
```

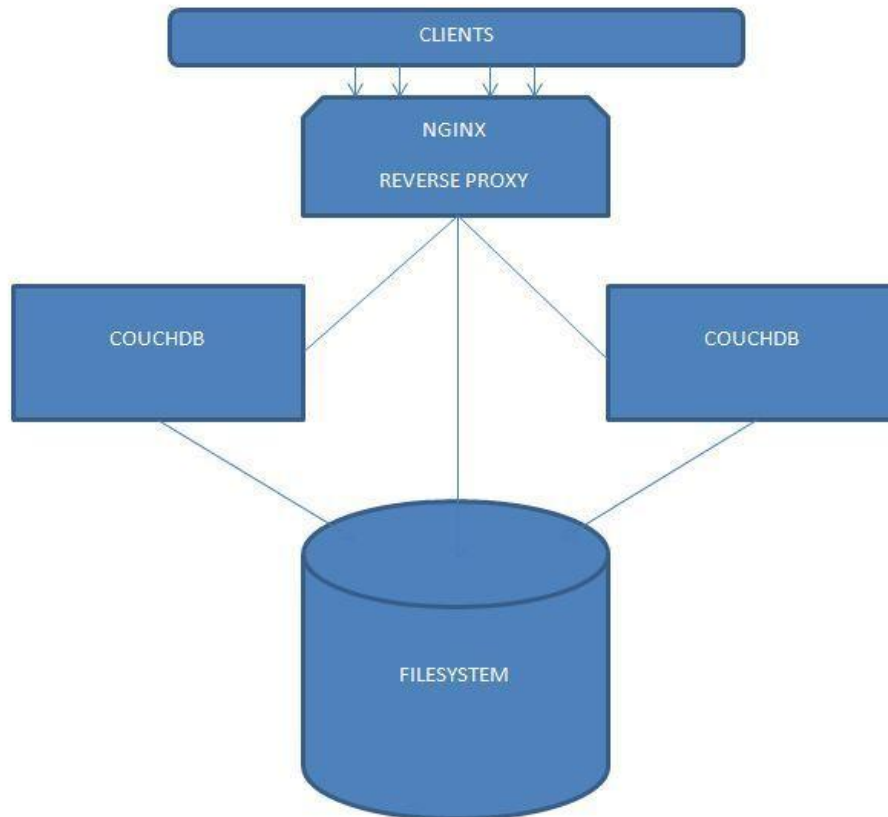
```
make distclean
sudo -i
# adding a user for couchdb
adduser --system --home /usr/local/var/lib/couchdb --
no-create-home --shell /bin/bash --group --gecos
"CouchDB Administrator" couchdb
# changing the owner to couchdb for couchdb's
directories
chown -R couchdb:couchdb /usr/local/var/lib/couchdb
chown -R couchdb:couchdb /usr/local/var/log/couchdb
chown -R couchdb:couchdb /usr/local/var/run
chown -R couchdb:couchdb /usr/local/etc/couchdb
# changing permissions for couchdb's directories
chmod -R 0770 /usr/local/var/lib/couchdb
chmod -R 0770 /usr/local/var/log/couchdb
chmod -R 0770 /usr/local/var/run
chmod -R 0770 /usr/local/etc/couchdb
# copying startup script to init.d/ to be available
within system boot process
cp /usr/local/etc/init.d/couchdb /etc/init.d/
update-rc.d couchdb defaults
# starting
/etc/init.d/couchdb start
```

Configuring CouchDB

After installation, CouchDB is ready to use. The default configuration is usually sufficient. But what if you want to make your database available from an external network? Or have to change the listen port because another instance of CouchDB is already using the default one? It's possible, when more than one instance of CouchDB is running on the same machine.

Let discuss an example.

Assume we need to create a simple CouchDB failover cluster of 2 nodes.



Nginx (a very popular lightweight web server) will act as a reverse proxy. It's the front end for our small cluster, that is, it will handle user requests and redirect them to one of the CouchDB servers, depending on settings or an algorithm defined in the Nginx configuration file.

We have to create a configuration file for each instance of CouchDB:

Copy the *default.ini* to *couch_alpha.ini* and *couch_beta.ini*

```
cp /usr/local/etc/couchdb/default.ini /usr/local/etc/couchdb/couch_alpha.ini
```

```
cp /usr/local/etc/couchdb/default.ini /usr/local/etc/couchdb/couch_beta.ini
```

Replace the port and ssl variable values in both files.

couch_alpha.ini must look like this (you can set other values for port and/or database_dir, this is just an example):

```
[httpd]
port = 5980
[ssl]
port = 6984
```

```
[couchdb]
database_dir = /usr/local/var/lib/couchdb/alpha
And couch_beta.ini must look like this:
[httpd]
port = 5981
[ssl]
port = 6985
[couchdb]
database_dir = /usr/local/var/lib/couchdb/beta
Create database directories for alpha and beta:
mkdir /usr/local/var/lib/couchdb/{alpha,beta}
Change the owner:
chown -R couchdb:couchdb /usr/local/var/lib/couchdb/
Start servers from the command line manually or modify the
startup script in /etc/init.d/ to do it automatically every time the
system boots
# Starting couchdb from command line
# couchdb -a /usr/local/etc/couchdb/couch_alpha.ini &
# couchdb -a /usr/local/etc/couchdb/couch_beta.ini &
```

```
root@virtual: /usr/local/etc/couchdb
```

```
root@virtual:/usr/local/etc/couchdb# curl http://localhost:5980
[info] [<0.126.0>] 127.0.0.1 - - GET / 200
{"couchdb":"Welcome","version":"1.2.0"}
root@virtual:/usr/local/etc/couchdb# curl http://localhost:5981
[info] [<0.177.0>] 127.0.0.1 - - GET / 200
{"couchdb":"Welcome","version":"1.2.0"}
root@virtual:/usr/local/etc/couchdb#
```

Okay, both servers are up and running.

Now, it's time to configure Nginx.

Edit the nginx.conf file (usually it's located in */etc/nginx*) and add the following in the *http* section:

```
upstream couchdb_cluster {
    server 127.0.0.1:5980;
    server 127.0.0.1:5981;
}
server {
    listen 80;
    server_name master.couch.local;
    location / {
        proxy_pass http://couchdb_cluster;
        break;
    }
}
```

```
}  
}
```

Add to your /etc/hosts file: `'127.0.0.1 master.couch.local'`
`echo '127.0.0.1 master.couch.local' >> /etc/hosts`
Restart Nginx.

Now let's create databases on both servers:

```
curl -X PUT http://localhost:5980/netangels  
curl -X PUT http://localhost:5981/netangels
```

Start continuous replication between databases:

```
curl -X POST -H 'Content-Type: application/json'  
http://localhost:5980/_replicate -d  
'{"source":"netangels", "target":"http://  
localhost:5981/netangels", "continuous":true}'
```

```
curl -X POST -H 'Content-Type: application/json'  
http://localhost:5981/_replicate -d  
'{"source":"netangels", "target":"http://  
localhost:5980/netangels", "continuous":true}'
```

We have completed basic installation.

Let's check it!

Create a document on alpha server:

```
#curl -X POST -H 'Content-Type: application/json'  
http://localhost:5980/netangels/ -d '{"color":"red"}
```

```
{"ok":true,"id":"06b419bd3cdfdcda2472718756000ffb","rev  
":"1-5f9b73300433277490f800eae6fd321d"}
```

Get the same document from beta server:

```
@virtual:/usr/local/etc/couchdb# curl http://localhost:5981/netangels/06b419bd3cdfdcda2472718756000ffb  
[<0.1603.0>] 127.0.0.1 - - GET /netangels/06b419bd3cdfdcda2472718756000ffb 200  
{"id":"06b419bd3cdfdcda2472718756000ffb","_rev":"1-5f9b73300433277490f800eae6fd321d","color":"red"}  
@virtual:/usr/local/etc/couchdb#
```

Replication is working.

Notice that continuous replication currently does not survive server restart. Maybe in a future release this will be fixed.

Monitoring CouchDB

Your server is ready to serve! What now?

Now we are going to monitor our server.

There are many monitoring tools for monitoring CouchDB, but in this article we will look at only Monitis tools, as they are easy to use and offer good functionality.

Basic metrics

CouchDB itself provides necessary statistical data via the REST interface, enough to make an educated guess about the server's health. By requesting http://server/_stats, we get information about the open databases count, request failures, etc., and all in JSON format!

Following is a list of the metrics available via the REST:

`http_codes` and `http_methods` are not actually metrics, but they provide useful information such as invalid queries count, internal errors count, non-existent documents count and detailed statistics for each HTTP method.

`open_databases` – number of open databases since last restart

`open_os_files` – number of file descriptors CouchDB has open at the moment

`database_writes` – number of times a database was changed since last restart

`database_reads` – number of times a document was read from a database since last restart

`request_time` – the time elapsed from the start of the request, after MochiWeb has passed it to CouchDB

`clients_requesting_changes` – number of clients for continuous changes since last restart

`requests` – number of HTTP requests since last restart

`view_reads` – number of view reads since last restart

`temporary_view_reads` – number of temporary view reads since last restart

`bulk_requests` – number of bulk requests since last restart

We will use [Monitis M3](#) to process all of these metrics.

Installing Monitis M3

First of all clone the [github repository](#):

```
git clone git@github.com:monitisexchange/Monitis-Linux-Scripts.git
```

This will clone the whole repository. We need only M3v3 part.

Install MonitisMonitorManager according to the [installation recommendations](#).

```
cd Monitis-Linux-Scripts/M3v3/MonitisMonitorManager
perl Makefile.PL
```

make

sudo make install

Edit M3Templates.pm and add your APIKEY and SECRETKEY

Check your template by running:

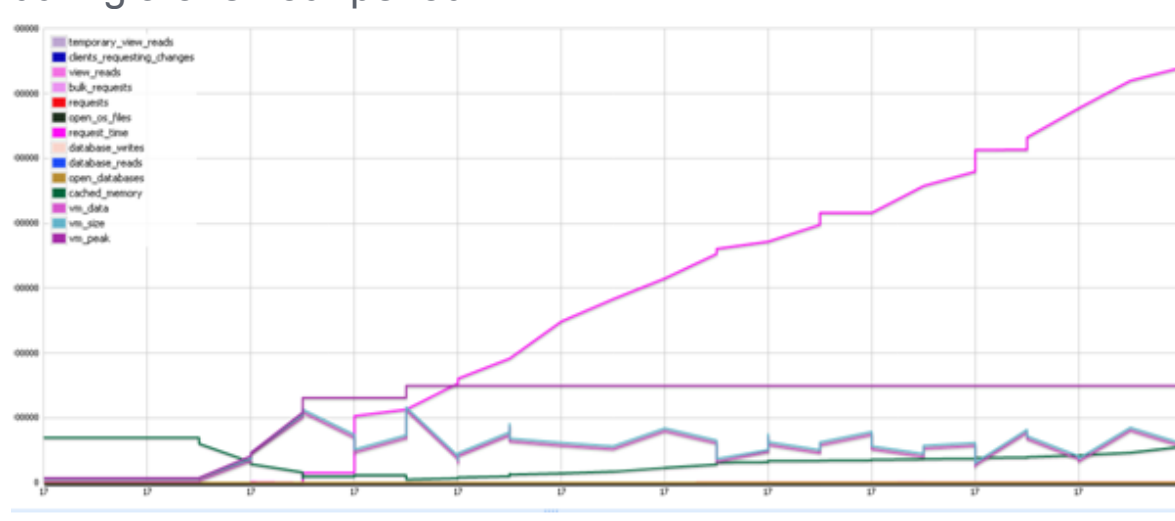
monitis-m3 --dry-run --once /path/to/monitis/
couchdb_monitor.xml

Ensure all is okay, then do:

monitis-m3 /path/to/monitis/couchd_template.xml

Now you should get data in your Monitis.com dashboard.

The graphics below show the variation in several metric's values during a one hour period:



virtual memory peak, view reads, cached_memory, open os files, vm_data values variation graphical representation

| CustomMonitor CouchDB_Monitor_debian | | | | | | | | | | | | | | | | |
|--|------------------|--------------------|------------|---------------|---------------|---------------------|--------------------|------------------|------------------|----------------|------------------|-----------------|------------------|---------|--|--|
| Last 24 hou | | | | | | | | | | | | | | | | |
| Time | temporary vie... | clients for con... | view reads | bulk requests | HTTP requests | file descriptors... | a request insid... | times a datab... | times a docum... | open databases | File System S... | Data Segment... | Virtual Memor... | Virtual | | |
| 05-31 17:38 | 0 | 0 | 0 | 1200 | 1463 | 1 | 3202685.569 | 1200 | 0 | 1 | 277576 | 278628 | 302532 | 74709 | | |
| 05-31 17:37 | 0 | 0 | 0 | 1130 | 1392 | 1 | 3100210.642 | 1130 | 0 | 1 | 231732 | 401388 | 425292 | 74709 | | |
| 05-31 17:36 | 0 | 0 | 0 | 1061 | 1322 | 1 | 2887252.103 | 997 | 0 | 1 | 211744 | 162988 | 186968 | 74709 | | |
| 05-31 17:36 | 0 | 0 | 0 | 997 | 1321 | 1 | 2887023.103 | 993 | 0 | 1 | 211684 | 181420 | 205324 | 74709 | | |
| 05-31 17:35 | 0 | 0 | 0 | 993 | 1252 | 1 | 2662271.401 | 993 | 0 | 1 | 196264 | 336180 | 360084 | 74709 | | |
| 05-31 17:35 | 0 | 0 | 0 | 993 | 1251 | 1 | 2662140.401 | 993 | 0 | 1 | 196264 | 336292 | 360196 | 74709 | | |
| 05-31 17:35 | 0 | 0 | 0 | 992 | 1249 | 1 | 2568412.401 | 992 | 0 | 1 | 194408 | 384948 | 408852 | 74709 | | |
| 05-31 17:34 | 0 | 0 | 0 | 992 | 1248 | 1 | 2566716.401 | 926 | 0 | 1 | 192896 | 133964 | 154280 | 74709 | | |
| 05-31 17:34 | 0 | 0 | 0 | 0 | 0 | 1 | 2565677.401 | 924 | 0 | 1 | 192912 | 123960 | 147864 | 74709 | | |
| 05-31 17:34 | 0 | 0 | 0 | 924 | 1178 | 1 | 2399374.088 | 924 | 0 | 1 | 186064 | 283748 | 307852 | 74709 | | |
| 05-31 17:33 | 0 | 0 | 0 | 923 | 1176 | 1 | 2286394.088 | 923 | 0 | 1 | 183796 | 262384 | 286288 | 74709 | | |
| 05-31 17:33 | 0 | 0 | 0 | 923 | 1176 | 1 | 2286394.088 | 923 | 0 | 1 | 183688 | 257264 | 282412 | 74709 | | |
| 05-31 17:33 | 0 | 0 | 0 | 923 | 1174 | 1 | 2287977.088 | 857 | 0 | 1 | 181340 | 199576 | 223292 | 74709 | | |
| 05-31 17:32 | 0 | 0 | 0 | 855 | 1105 | 1 | 2080765.647 | 855 | 0 | 1 | 177492 | 257988 | 281892 | 74709 | | |
| 05-31 17:32 | 0 | 0 | 0 | 855 | 1105 | 1 | 2080751.647 | 855 | 0 | 1 | 173836 | 369396 | 393300 | 74709 | | |
| 05-31 17:31 | 0 | 0 | 0 | 855 | 1103 | 1 | 2080729.647 | 855 | 0 | 1 | 171528 | 288276 | 312180 | 74709 | | |
| 05-31 17:31 | 0 | 0 | 0 | 854 | 1101 | 1 | 1989667.647 | 854 | 0 | 1 | 170176 | 250064 | 273968 | 74709 | | |
| 05-31 17:31 | 0 | 0 | 0 | 0 | 1100 | 1 | 1989432.647 | 810 | 0 | 1 | 170028 | 228636 | 252572 | 74709 | | |
| 05-31 17:30 | 0 | 0 | 0 | 786 | 1031 | 1 | 1859375.776 | 786 | 0 | 1 | 169472 | 250972 | 314700 | 74709 | | |
| 05-31 17:30 | 0 | 0 | 0 | 786 | 1030 | 1 | 1859339.776 | 786 | 0 | 1 | 163236 | 351212 | 375116 | 74709 | | |
| 05-31 17:30 | 0 | 0 | 0 | 785 | 1029 | 1 | 1859307.776 | 785 | 0 | 1 | 159480 | 238400 | 253972 | 74709 | | |
| 05-31 17:29 | 0 | 0 | 0 | 785 | 1027 | 1 | 1803678.776 | 717 | 0 | 1 | 156252 | 160248 | 184152 | 74709 | | |
| 05-31 17:29 | 0 | 0 | 0 | 717 | 958 | 1 | 1764243.085 | 717 | 0 | 1 | 140592 | 302980 | 326884 | 74709 | | |
| 05-31 17:28 | 0 | 0 | 0 | 579 | 814 | 1 | 1575494.343 | 579 | 0 | 1 | 118512 | 396520 | 420474 | 74709 | | |

all metric's values variation table view

| CustomMonitor CouchDB_Monitor_debian | | | | | | | | | | | | | | | | |
|--|------------|-------------|--------------|-------------|-------------|--------------|-------------|--------|----------------|--------------|-------------|-------------|------|-----|------|-----|
| Last 24 hou | | | | | | | | | | | | | | | | |
| #12 Prec... | 304 Not... | 401 Unau... | 500 Inter... | 405 Meth... | 301 Move... | 404 Not F... | 202 Acco... | 200 OK | 409 Conflic... | 403 Forbi... | 201 Crea... | 400 Bad ... | COPY | GET | MOVE | PUT |
| 0 | 0 | 0 | 31 | 0 | 0 | 0 | 0 | 301 | 0 | 0 | 1131 | 0 | 0 | 301 | 0 | 0 |

additional data set for a single row.