

Couchbase CheatSheet

TOOLS

- PDF Link: [cheatsheet-couchbase-A4.pdf](#), Category: tools
- Blog URL: <https://cheatsheet.dennyzhang.com/cheatsheet-couchbase-A4>
- Related posts: [Elasticsearch CheatSheet](#), [#denny-cheatsheets](#)

File me Issues or star this repo.

1.1 Couchbase Summary

Name	Command
Check couchbase version	<code>grep version /opt/couchbase/etc/runtime.ini, cat /opt/couchbase/VERSION.txt</code>
Get version	<code>cat /opt/couchbase/VERSION.txt</code>
Check couchbase status	<code>/etc/init.d/couchbase-server status</code>
couchbase web console	<code>http://localhost:8091 Administrator/password</code>
cbdump-config	<code>/opt/couchbase/bin/cbdump-config</code>
pre configuration	<code>/opt/couchbase/var/lib/couchbase/config/config.dat</code>
log files	<code>/opt/couchbase/var/lib/couchbase/logs</code>
Flush cb bucket by api	<code>flush-cb-bucket.sh</code>

1.2 Couchbase CLI

Name	Command
couchbase-cli	<code>/opt/couchbase/bin/couchbase-cli</code>
backup couchbase	<code>/opt/couchbase/bin/cbbackup http://localhost:8091 /backup-42</code>
List buckets	<code>/opt/couchbase/bin/couchbase-cli bucket-list -c localhost</code>
Check bucket RAM usage	<code>cbstats -b \$bucket_name 127.0.0.1:11210 memory</code>
Create bucket	See sample-couchbase-cli.sh
Load sample data	See sample-couchbase-cli.sh
Reset couchbase passwd	See sample-couchbase-cli.sh
Initialize couchbase cluster	See sample-couchbase-cli.sh

1.3 Couchbase REST API

Name	Command
List buckets	<code>curl -u \$username:\$passwd http://\$es_ip:8091/pools/default/buckets</code>
Get couchbase nodes	<code>curl -u \$username:\$passwd \$couchbase_ip:8091/pools/default</code>
Remove couchbase node	<code>curl -u \$username:\$passwd -d otpNode=ns_1@172.17.1.71 http://172.17.1.71:8091/control</code>

1.4 couchbase-cli Online Usage

```
> couchbase-cli help
couchbase-cli - command-line cluster administration tool
```

usage: couchbase-cli COMMAND CLUSTER [OPTIONS]

COMMAND:

server-list	list all servers in a cluster
server-info	show details on one server
server-add	add one or more servers to the cluster
server-readd	readd a server that was failed over
group-manage	manage server groups
rebalance	start a cluster rebalancing
rebalance-stop	stop current cluster rebalancing
rebalance-status	show status of current cluster rebalancing
failover	failover one or more servers
cluster-init	set the username,password and port of the cluster
cluster-edit	modify cluster settings

node-init	set node specific parameters
bucket-list	list all buckets in a cluster
bucket-create	add a new bucket to the cluster
bucket-edit	modify an existing bucket
bucket-delete	delete an existing bucket
bucket-flush	flush all data from disk for a given bucket
bucket-compact	compact database and index data
setting-compaction	set auto compaction settings
setting-notification	set notification settings
setting-alert	set email alert settings
setting-autofailover	set auto failover settings
setting-xdcr	set xdcr related settings
ssl-manage	manage cluster certificate
user-manage	manage read only user
xdcr-setup	set up XDCR connection
xdcr-replicate	xdcr operations
help	show longer usage/help and examples

CLUSTER:

```
--cluster=HOST[:PORT] or -c HOST[:PORT]
```

OPTIONS:

-u USERNAME, --user=USERNAME	admin username of the cluster
-p PASSWORD, --password=PASSWORD	admin password of the cluster
-o KIND, --output=KIND	KIND is json or standard
-d, --debug	

server-add OPTIONS:

--server-add=HOST[:PORT]	server to be added
--server-add-username=USERNAME	admin username for the server to be added
--server-add-password=PASSWORD	admin password for the server to be added
--group-name=GROUPNAME	group that server belongs

server-readd OPTIONS:

--server-add=HOST[:PORT]	server to be added
--server-add-username=USERNAME	admin username for the server to be added
--server-add-password=PASSWORD	admin password for the server to be added
--group-name=GROUPNAME	group that server belongs

rebalance OPTIONS:

--server-add*	see server-add OPTIONS
--server-remove=HOST[:PORT]	the server to be removed

group-manage OPTIONS:

--group-name=GROUPNAME	group name
--create	create a new group
--delete	delete an empty group
--list	show group/server relationship map
--rename=NEWGROUPNAME	rename group to new name
--add-servers=HOST[:PORT];HOST[:PORT]	add a list of servers to group
--move-servers=HOST[:PORT];HOST[:PORT]	move a list of servers from group
--from-group=GROUPNAME	group name that to move servers from
--to-group=GROUPNAME	group name tat to move servers to

failover OPTIONS:

```

--server-failover=HOST[:PORT]      server to failover

cluster-* OPTIONS:
--cluster-username=USER             new admin username
--cluster-password=PASSWORD         new admin password
--cluster-port=PORT                 new cluster REST/http port
--cluster-ramsize=RAMSIZEMB         per node ram quota in MB

node-init OPTIONS:
--node-init-data-path=PATH          per node path to store data
--node-init-index-path=PATH         per node path to store index

bucket-* OPTIONS:
--bucket=BUCKETNAME                bucket to act on
--bucket-type=TYPE                  memcached or couchbase
--bucket-port=PORT                  supports ASCII protocol and is auth-less
--bucket-password=PASSWORD          standard port, exclusive with bucket-port
--bucket-ramsize=RAMSIZEMB          ram quota in MB
--bucket-replica=COUNT             replication count
--enable-flush=[0|1]                enable/disable flush
--enable-index-replica=[0|1]        enable/disable index replicas
--wait                              wait for bucket create to be complete before returning
--force                             force to execute command without asking for confirmation
--data-only                         compact database data only
--view-only                         compact view data only

setting-compaction OPTIONS:
--compaction-db-percentage=PERCENTAGE at which point database compaction is triggered
--compaction-db-size=SIZE[MB]        at which point database compaction is triggered
--compaction-view-percentage=PERCENTAGE at which point view compaction is triggered
--compaction-view-size=SIZE[MB]      at which point view compaction is triggered
--compaction-period-from=HH:MM       allow compaction time period from
--compaction-period-to=HH:MM         allow compaction time period to
--enable-compaction-abort=[0|1]       allow compaction abort when time expires
--enable-compaction-parallel=[0|1]    allow parallel compaction for database and view
--metadata-purge-interval=DAYS        how frequently a node will purge metadata on deleted items

setting-notification OPTIONS:
--enable-notification=[0|1]           allow notification

setting-alert OPTIONS:
--enable-email-alert=[0|1]            allow email alert
--email-recipients=RECIPIENT          email recipients, separate addresses with , or ;
--email-sender=SENDER                 sender email address
--email-user=USER                     email server username
--email-password=PWD                  email server password
--email-host=HOST                     email server host
--email-port=PORT                     email server port
--enable-email-encrypt=[0|1]          email encrypt
--alert-auto-failover-node            node was auto failover
--alert-auto-failover-max-reached     maximum number of auto failover nodes was reached
--alert-auto-failover-node-down       node wasn't auto failover as other nodes are down at the same time
--alert-auto-failover-cluster-small   node wasn't auto fail over as cluster was too small
--alert-ip-changed                    node ip address has changed unexpectedly
--alert-disk-space                    disk space used for persistent storage has reached at least 90%
--alert-meta-overhead                 metadata overhead is more than 50%
--alert-meta-oom                      bucket memory on a node is entirely used for metadata
--alert-write-failed                  writing data to disk for a specific bucket has failed

```

setting-autofailover OPTIONS:

```
--enable-auto-failover=[0|1]      allow auto failover
--auto-failover-timeout=TIMEOUT (>=30)  specify timeout that expires to trigger auto failover
```

setting-xdcr OPTIONS:

```
--max-concurrent-reps=[32]      maximum concurrent replications per bucket, 8 to 256.
--checkpoint-interval=[1800]    intervals between checkpoints, 60 to 14400 seconds.
--worker-batch-size=[500]      doc batch size, 500 to 10000.
--doc-batch-size=[2048]KB      document batching size, 10 to 100000 KB
--failure-restart-interval=[30] interval for restarting failed xdcr, 1 to 300 seconds
--optimistic-replication-threshold=[256] document body size threshold (bytes) to trigger optimistic replica
```

xdcr-setup OPTIONS:

```
--create      create a new xdcr configuration
--edit        modify existed xdcr configuration
--delete      delete existed xdcr configuration
--xdcr-cluster-name=CLUSTERNAME  cluster name
--xdcr-hostname=HOSTNAME        remote host name to connect to
--xdcr-username=USERNAME        remote cluster admin username
--xdcr-password=PASSWORD        remote cluster admin password
--xdcr-demand-encryption=[0|1]  allow data encrypted using ssl
--xdcr-certificate=CERTIFICATE  pem-encoded certificate. Need be present if xdcr-demand-encryption i
```

xdcr-replicate OPTIONS:

```
--create      create and start a new replication
--delete      stop and cancel a replication
--list        list all xdcr replications
--xdcr-from-bucket=BUCKET      local bucket name to replicate from
--xdcr-clucter-name=CLUSTERNAME  remote cluster to replicate to
--xdcr-to-bucket=BUCKETNAME    remote bucket to replicate to
```

user-manage OPTIONS:

```
--set      create/modify a read only user
--list      list any read only user
--delete    delete read only user
--ro-username=USERNAME  readonly user name
--ro-password=PASSWORD  readonly user password
```

ssl-manage OPTIONS:

```
--retrieve-cert=CERTIFICATE  retrieve cluster certificate AND save to a pem file
--regenerate-cert=CERTIFICATE regenerate cluster certificate AND save to a pem file
```

The default PORT number is 8091.

EXAMPLES:

Set data path for an unprovisioned cluster:

```
couchbase-cli node-init -c 192.168.0.1:8091 \
  --node-init-data-path=/tmp/data \
  --node-init-index-path=/tmp/index \
  -u Administrator -p password
```

List servers in a cluster:

```
couchbase-cli server-list -c 192.168.0.1:8091
```

Server information:

```
couchbase-cli server-info -c 192.168.0.1:8091
```

Add a node to a cluster, but do not rebalance:

```
couchbase-cli server-add -c 192.168.0.1:8091 \
```

```
--server-add=192.168.0.2:8091 \  
--server-add-username=Administrator1 \  
--server-add-password=password1 \  
--group-name=group1 \  
-u Administrator -p password
```

Add a node to a cluster and rebalance:

```
couchbase-cli rebalance -c 192.168.0.1:8091 \  
--server-add=192.168.0.2:8091 \  
--server-add-username=Administrator1 \  
--server-add-password=password1 \  
--group-name=group1 \  
-u Administrator -p password
```

Remove a node from a cluster and rebalance:

```
couchbase-cli rebalance -c 192.168.0.1:8091 \  
--server-remove=192.168.0.2:8091 \  
-u Administrator -p password
```

Remove and add nodes from/to a cluster and rebalance:

```
couchbase-cli rebalance -c 192.168.0.1:8091 \  
--server-remove=192.168.0.2 \  
--server-add=192.168.0.4 \  
--server-add-username=Administrator1 \  
--server-add-password=password1 \  
--group-name=group1 \  
-u Administrator -p password
```

Stop the current rebalancing:

```
couchbase-cli rebalance-stop -c 192.168.0.1:8091 \  
-u Administrator -p password
```

Set the username, password, port and ram quota:

```
couchbase-cli cluster-init -c 192.168.0.1:8091 \  
--cluster-init-username=Administrator \  
--cluster-init-password=password \  
--cluster-init-port=8080 \  
--cluster-init-ramsize=300
```

change the cluster username, password, port and ram quota:

```
couchbase-cli cluster-edit -c 192.168.0.1:8091 \  
--cluster-username=Administrator1 \  
--cluster-password=password1 \  
--cluster-port=8080 \  
--cluster-ramsize=300 \  
-u Administrator -p password
```

Change the data path:

```
couchbase-cli node-init -c 192.168.0.1:8091 \  
--node-init-data-path=/tmp \  
-u Administrator -p password
```

List buckets in a cluster:

```
couchbase-cli bucket-list -c 192.168.0.1:8091
```

Create a new dedicated port couchbase bucket:

```
couchbase-cli bucket-create -c 192.168.0.1:8091 \  
--bucket=test_bucket \  
--bucket-type=couchbase \  

```

```
--bucket-port=11222 \  
--bucket-ramsize=200 \  
--bucket-replica=1 \  
-u Administrator -p password
```

Create a couchbase bucket and wait for bucket ready:

```
couchbase-cli bucket-create -c 192.168.0.1:8091 \  
--bucket=test_bucket \  
--bucket-type=couchbase \  
--bucket-port=11222 \  
--bucket-ramsize=200 \  
--bucket-replica=1 \  
--wait \  
-u Administrator -p password
```

Create a new sasl memcached bucket:

```
couchbase-cli bucket-create -c 192.168.0.1:8091 \  
--bucket=test_bucket \  
--bucket-type=memcached \  
--bucket-password=password \  
--bucket-ramsize=200 \  
--enable-flush=1 \  
-u Administrator -p password
```

Modify a dedicated port bucket:

```
couchbase-cli bucket-edit -c 192.168.0.1:8091 \  
--bucket=test_bucket \  
--bucket-port=11222 \  
--bucket-ramsize=400 \  
--enable-flush=1 \  
-u Administrator -p password
```

Delete a bucket:

```
couchbase-cli bucket-delete -c 192.168.0.1:8091 \  
--bucket=test_bucket
```

Flush a bucket:

```
couchbase-cli bucket-flush -c 192.168.0.1:8091 \  
--force \  
-u Administrator -p password
```

Compact a bucket for both data and view

```
couchbase-cli bucket-compact -c 192.168.0.1:8091 \  
--bucket=test_bucket \  
-u Administrator -p password
```

Compact a bucket for data only

```
couchbase-cli bucket-compact -c 192.168.0.1:8091 \  
--bucket=test_bucket \  
--data-only \  
-u Administrator -p password
```

Compact a bucket for view only

```
couchbase-cli bucket-compact -c 192.168.0.1:8091 \  
--bucket=test_bucket \  
--view-only \  
-u Administrator -p password
```

Create a XDCR remote cluster

```
couchbase-cli xdcr-setup -c 192.168.0.1:8091 \  
--create \  
--xdc-cluster-name=test \  
--xdc-hostname=10.1.2.3:8091 \  
--xdc-username=Administrator1 \  
--xdc-password=password1 \  
--xdc-demand-encryption=1 \  
--xdc-certificate=/tmp/test.pem \  
-u Administrator -p password
```

Delete a XDCR remote cluster

```
couchbase-cli xdcr-setup -delete -c 192.168.0.1:8091 \  
--xdc-cluster-name=test \  
-u Administrator -p password
```

Start a replication stream in memcached protocol

```
couchbase-cli xdcr-replicate -c 192.168.0.1:8091 \  
--create \  
--xdc-cluster-name=test \  
--xdc-from-bucket=default \  
--xdc-to-bucket=default1 \  
--xdc-replication-mode=xmem \  
-u Administrator -p password
```

Start a replication stream in capi protocol

```
couchbase-cli xdcr-replicate -c 192.168.0.1:8091 \  
--create \  
--xdc-cluster-name=test \  
--xdc-from-bucket=default \  
--xdc-to-bucket=default1 \  
--xdc-replication-mode=capi \  
-u Administrator -p password
```

Delete a replication stream

```
couchbase-cli xdcr-replicate -c 192.168.0.1:8091 \  
--delete \  
--xdc-replicator=f4eb540d74c43fd3ac6d4b7910c8c92f/default/default \  
-u Administrator -p password
```

List all xdcr replication streams

```
couchbase-cli xdcr-replicate -c 192.168.0.1:8091 \  
--list \  
-u Administrator -p password
```

List read only user in a cluster:

```
couchbase-cli user-manage --list -c 192.168.0.1:8091 \  
-u Administrator -p password
```

Delete a read only user in a cluster

```
couchbase-cli user-manage -c 192.168.0.1:8091 \  
--delete --ro-username=readonlyuser \  
-u Administrator -p password
```

create/modify a read only user in a cluster

```
couchbase-cli user-manage -c 192.168.0.1:8091 \  
--set --ro-username=readonlyuser --ro-password=readonlypassword \  
-u Administrator -p password
```

Create a new group

```
couchbase-cli group-manage -c 192.168.0.1:8091 \
--create --group-name=group1 -u Administrator -p password
```

Delete an empty group

```
couchbase-cli group-manage -c 192.168.0.1:8091 \
--delete --group-name=group1 -u Administrator -p password
```

Rename an existed group

```
couchbase-cli group-manage -c 192.168.0.1:8091 \
--rename=newgroup --group-name=group1 -u Administrator -p password
```

Show group/server map

```
couchbase-cli group-manage -c 192.168.0.1:8091 \
--list -u Administrator -p password
```

Add a server to a group

```
couchbase-cli group-manage -c 192.168.0.1:8091 \
--add-servers="10.1.1.1:8091;10.1.1.2:8091" \
--group-name=group1 \
--server-add-username=Administrator1 \
--server-add-password=password1 \
-u Administrator -p password
```

Move list of servers from group1 to group2

```
couchbase-cli group-manage -c 192.168.0.1:8091 \
--move-servers="10.1.1.1:8091;10.1.1.2:8091" \
--from-group=group1 \
--to-group=group2 \
-u Administrator -p password
```

Download a cluster certificate

```
couchbase-cli ssl-manage -c 192.168.0.1:8091 \
--retrieve-cert=/tmp/test.pem \
-u Administrator -p password
```

Regenerate AND download a cluster certificate

```
couchbase-cli ssl-manage -c 192.168.0.1:8091 \
--regenerate-cert=/tmp/test.pem \
-u Administrator -p password
```

```
root@testmini1:/opt/couchbase/bin#
```

```
root@testmini1:/cloudpass/backend/build/bin# cd /opt/couchbase/bin
```

```
root@testmini1:/opt/couchbase/bin# ls
```

cbbackup	cbworkloadgen	dump-guts	install
cbbrowse_logs	couchbase-cli	ebucketmigrator	makeconv
cbcollect_info	couchbase-server	epmd	memcached
cbdump-config	couch_compact	erl	moxi
cbenable_core_dumps.sh	couchdb	erlc	reports
cbepctl	couch_dbdump	escript	sigar_port
cbhealthchecker	couch_dbinfo	genbrk	sqlite3
cbrecovery	couchjs	gencfu	to_erl
cbreset_password	couch_viewgen	gencnval	tools
cbrestore	curl	genctd	typer
cbstats	curl-config	generate_cert	uconv
cbtransfer	derb	genrb	vbmap
cbvbucketctl	dialyzer	icu-config	

1.5 More Resources

<http://www.couchbase.com/#usecases>

<https://www.digitalocean.com/community/tutorials/how-to-install-couchbase-from-source-with-git-and-make-o>

License: Code is licensed under MIT License.