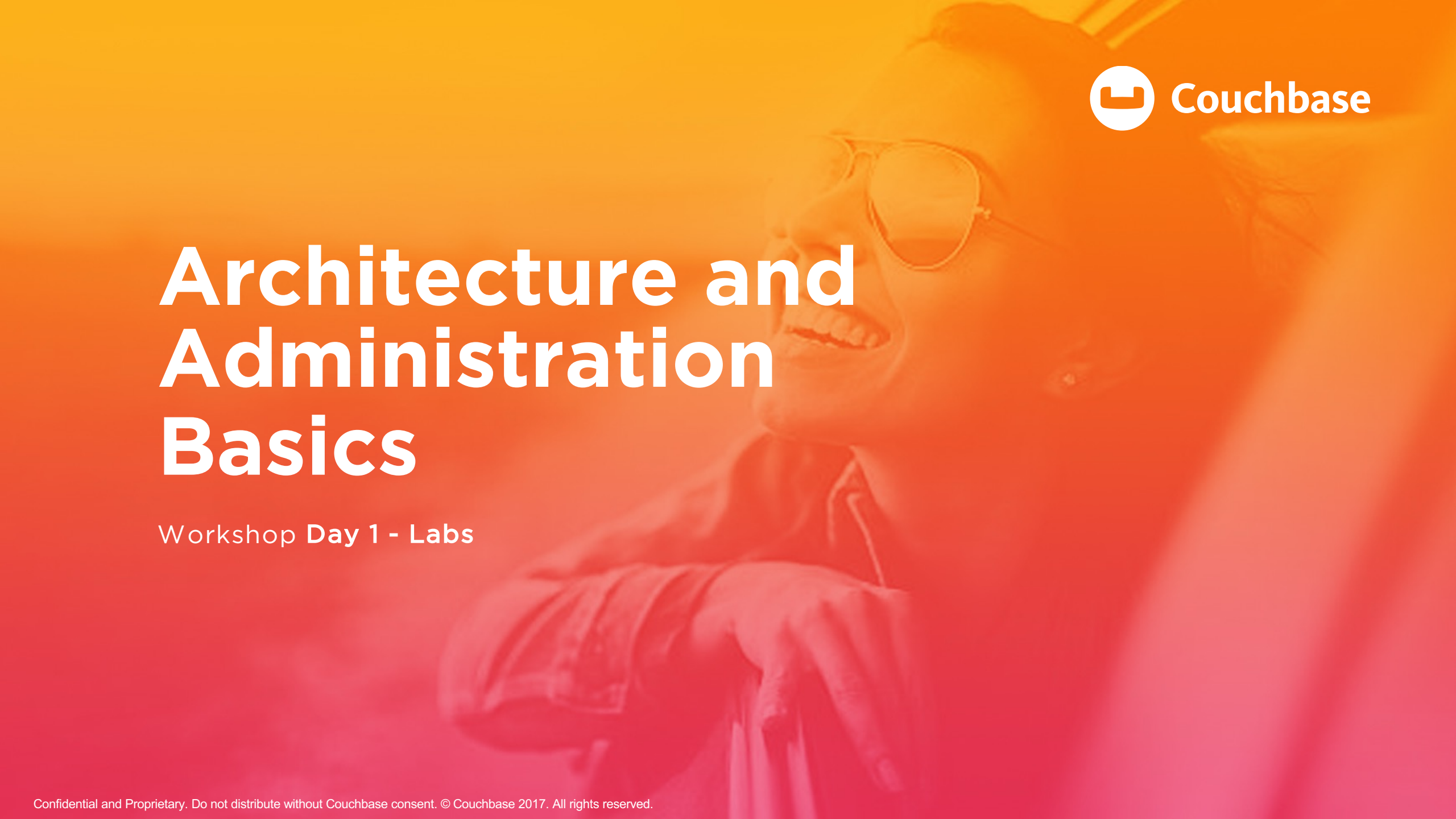


Couchbase

Architecture and Administration Basics

Workshop **Day 1 - Labs**

1 | Installation & Configuration

Installation - Swappiness

Perform the following steps in order install Couchbase Server on CentOS 7.x

- Disable Swappiness

```
# Set the value for the running system
sudo sh -c 'echo 0 > /proc/sys/vm/swappiness'

# Backup sysctl.conf
sudo cp -p /etc/sysctl.conf /etc/sysctl.conf.`date +%Y%m%d-%H:%M`

# This disables it permanently
# Set the value in /etc/sysctl.conf so it stays after reboot.
sudo sh -c 'echo "" >> /etc/sysctl.conf'
sudo sh -c 'echo "#Set swappiness to 0 to avoid swapping" >> /etc/sysctl.conf'
sudo sh -c 'echo "vm.swappiness = 0" >> /etc/sysctl.conf'
reboot
```

Installation - Firewall

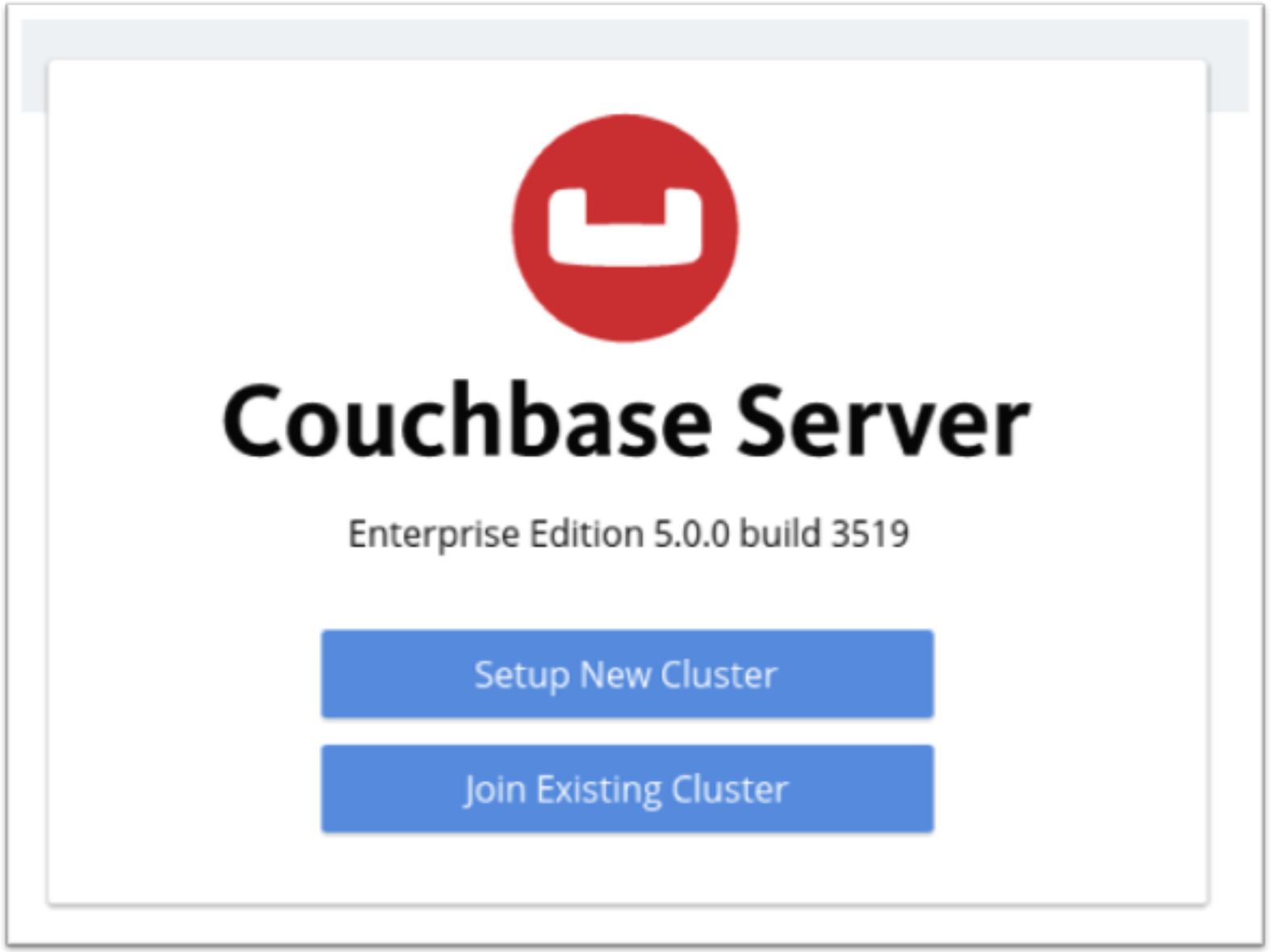
- Disable the Linux Firewall
  - May be configured in a production environment regarding http://developer.couchbase.com/documentation/server/current/install/install-ports.html

```
## Run as root or sudo
# Check the state
systemctl status firewalld

# Stop it
systemctl stop firewalld

# Disable it
systemctl disable firewalld
```

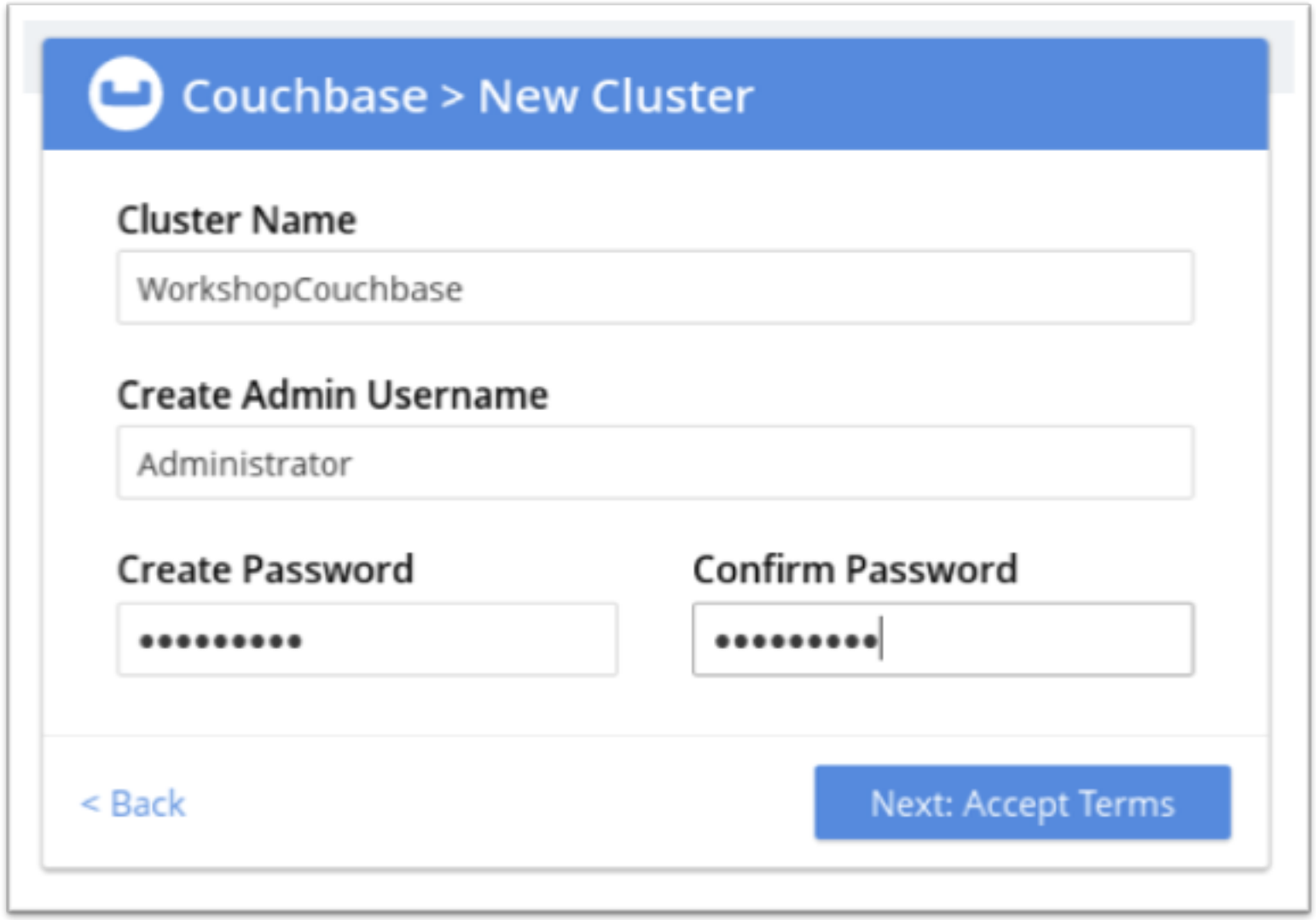
Installation – Download Couchbase 5.1

- Download the installation package from a browser or wget:
  - https://www.couchbase.com/downloads
  - `wget https://packages.couchbase.com/releases/5.1.1/couchbase-server-enterprise-5.1.1-centos7.x86_64.rpm`
- (Optional) SCP the .rpm to your local machine and then 'scp' the file to VMs.
  - `scp ${downloaded package}.rpm couchbase@://<public hostname of your VM>:/home/couchbase/Downloads/`
  - `pscp C:\Downloads\${downloaded package}.rpm couchbase@://<public hostname of your VM>:/home/couchbase/Downloads/`
- Perform the installation by using RPM
  - `sudo rpm --install ${downloaded package}.rpm`
- Open the Web UI Wizard
  - `http://<public hostname of your VM>:8091`

Installation– Setup New Cluster



Installation (password = couchbase)



Installation

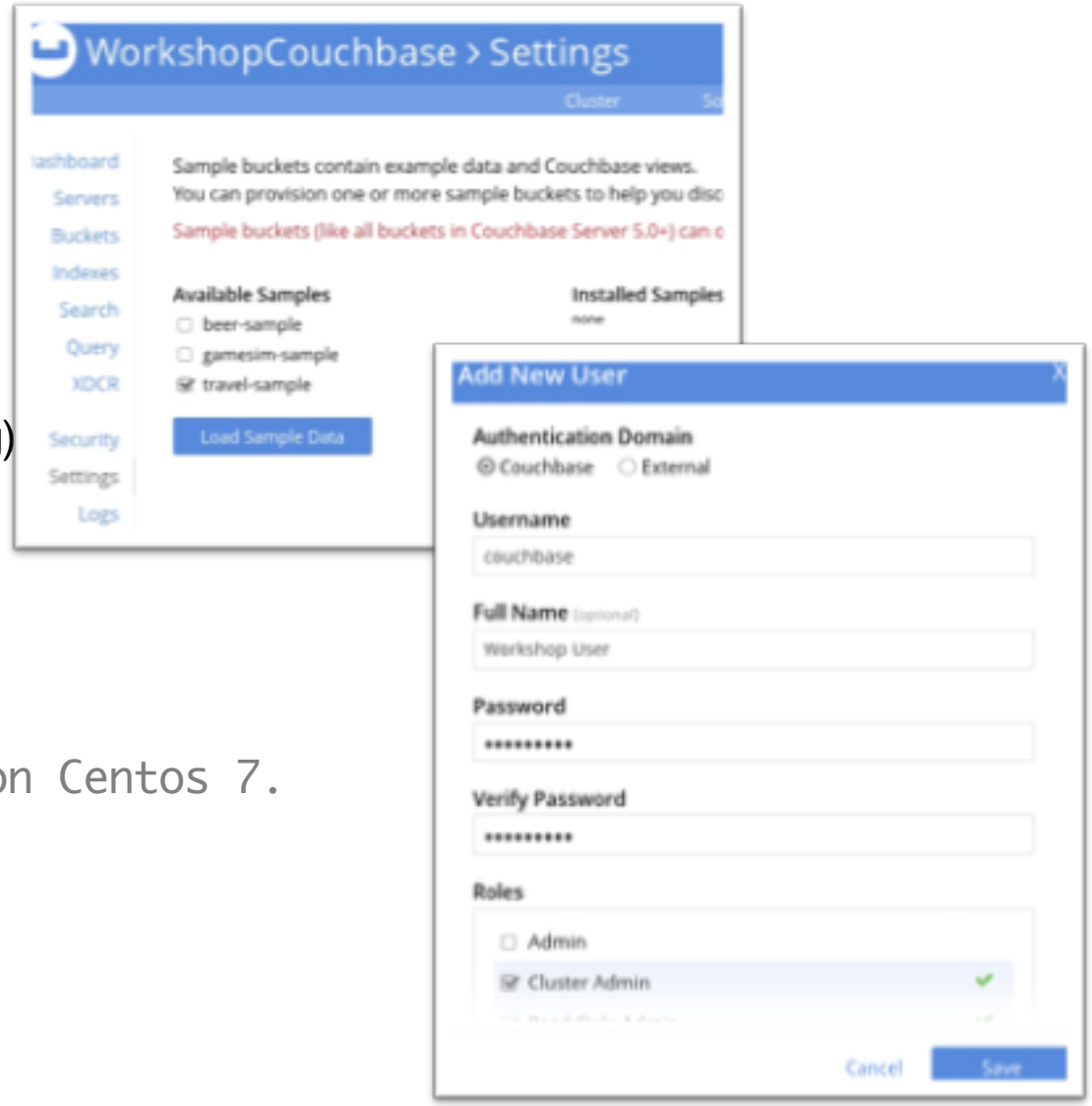
- **Hostname**: Your public IP or localhost.
- **Data Disk Path**: /opt/couchbase/var/lib/couchbase/data
- **Indexes Disk Path**: /opt/couchbase/var/lib/couchbase/index
- **Data Service**: 512Mo
- **Index Service**: 512Mo
- **Search Service**: 256Mo

- **Index Storage Setting**: Standard Global Secondary (Plasma)

Installation – Sample buckets

- Perform further steps in the Wizard
  - Add the travel-sample bucket
  - Edit the Travel-Sample configuration and remove the replicas.
  - Create an Administrator in Settings.

    User: couchbase
    Password: couchbase
  - Browse the UI and check Statistics. (while travel-sample is loading)



- Check that Couchbase Server has started.

```
sudo systemctl status couchbase-server

#Note & Optional: How to stop & start Couchbase on Centos 7.
sudo systemctl stop couchbase-server
sudo systemctl start couchbase-server
```

- Set up the command line environment in the Path.

```
export PATH=$PATH:$HOME/bin:/opt/couchbase/bin
```

2 | Testing Installation

Testing the Installation – REST API

Perform the following steps with the REST API

- *Install a plug-in in Browser to format JSON*
  Ex: Beautiful JSON {J}
- *Install a CLI plug-in to format JSON*
  Ex: yum install jq

- Check some statistics

- Cluster Status:
  http://<Your Public IP>:8091/nodeStatuses
  curl -u ${admin user}:${password} http://${Your Public IP}:8091/nodeStatuses | jq
- System Statistics:
  http://<Your Public IP>:8091/pools
- Cluster Details:
  http://<Your Public IP>:8091/pools/default
- Bucket Monitoring:
  http://<Your Public IP>:8091/pools/default/buckets/travel-sample
- Tasks running:
  http://<Your Public IP>:8091/pools/default/tasks
- Performance on Queries:
  http://<Your Public IP>:8093/admin/vitals
- Statistics on Indexes (check storage mode):
  http://<Your Public IP>:9102/stats

Testing the Installation - CLI

Perform the following steps in order to test your installation

- List the nodes & buckets of your current cluster

```
couchbase-cli server-list --cluster=${ip}:8091 -u=${admin user} -p=${password}
couchbase-cli bucket-list --cluster=${ip}:8091 -u=${admin user} -p=${password}
```

- Investigate the data and index directory

```
# You should see approximately 1030 files in this directory.
# So one file per vBucket + some extra files.
sudo ls -al /opt/couchbase/var/lib/couchbase/data/travel-sample

#List index files
sudo ls -al /opt/couchbase/var/lib/couchbase/index/@2i/…
```

Testing the Installation - Telnet

- Get some data & info from a vBucket file (here vbucket = 0)

- `couch_dbdump /opt/couchbase/var/lib/couchbase/data/travel-sample/0.couch.1`
- `couch_dbinfo /opt/couchbase/var/lib/couchbase/data/travel-sample/0.couch.1`

- Create an ephemeral moxi bucket & check connectivity with Telnet.

- `couchbase-cli bucket-create -c <IP>:8091 --username Administrator \`
  `--password couchbase --bucket test --bucket-type couchbase \`
  `--bucket-port 11252` `--bucket-ramsize 128`
- `sudo yum install telnet`
- `telnet <IP host> 11252`

```
[couchbase@localhost bin]$ telnet 192.168.56.102 11252
Trying 192.168.56.102...
Connected to 192.168.56.102.
Escape character is '^]'.
```

Testing the Installation - Telnet

- Retrieve some statistics of the default bucket

```
stats
```

- Set a new key with value (set $key $flags $exptime $numbytes $value)

```
set test_key 0 300 4
<Enter>
data
```

- Get the key.

```
get test_key
```

- Quit

```
quit
```

Testing the Installation – Cbworkloadgen

- Generate a workload on the bucket “test”

```
cd /opt/couchbase/bin
```

- Generate a workload with cbworkloadgen
  - 50% write & 50% read (r = % of write workload => 1 means 100% writes, 0 means 100% reads)
  - Size = 100 bytes
  - Number of items = 250 000
  - Number of threads = 2
  - Json documents

```
cbworkloadgen –n <IP>:8091 -u Administrator -p couchbase -b test -i 250000 -r .5
-s 100 -t 2 -j
```

- Observe the Metrics on the UI.

https://developer.couchbase.com/documentation/server/current/cli/cbworkloadgen-tool.html

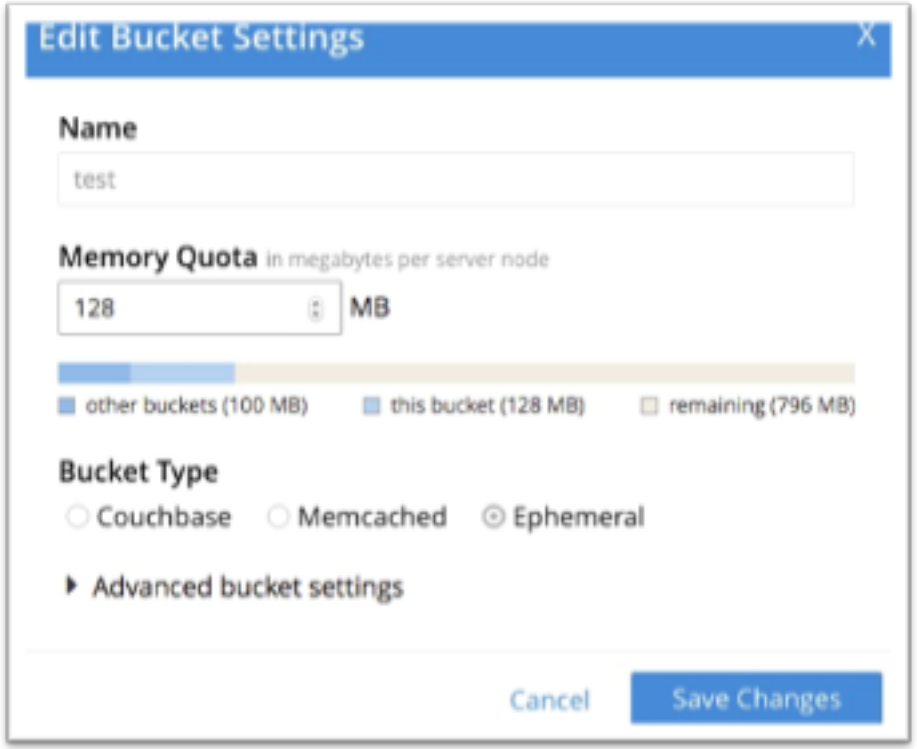
3 | Buckets Operations

Edit a Bucket

Perform the following steps in order to edit a bucket

- Open the Web Admin UI and go to the ‘Buckets’ tab

```
http://<public hostname of your VM>:8091
```



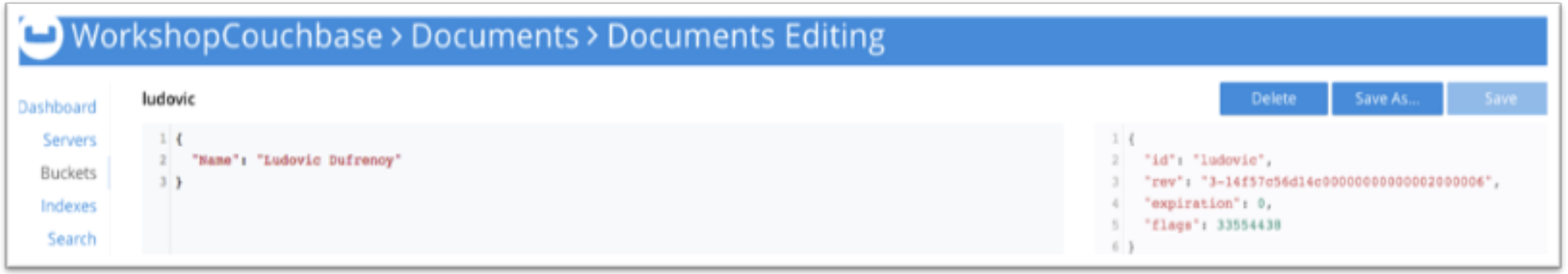
- Edit the bucket ‘test’ and configure the following:
  - Update the Memory quota to 256 MB RAM
  - Enable one Replica (Why do you get a Warning?)
  - Enable Flush

- Insert a new document in the bucket & search for it from the UI with .

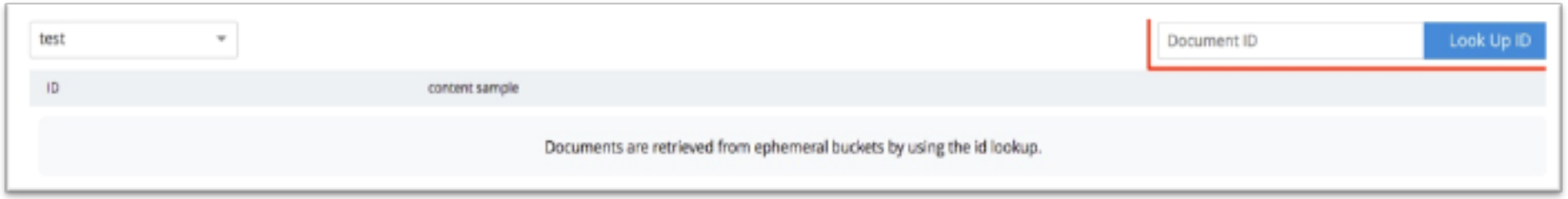
Create a document in the Bucket

Perform the following steps in order to add a document

- Open the Web Admin UI and go to the Buckets.
- Insert a new document in the bucket (check the Metadata)



- Search for it with the name of the key.



4 | Cluster Operations

Cluster Operations: Start a Cluster with Docker

Perform the following steps:

- Stop the local Couchbase instance again. (At each VM restart also)

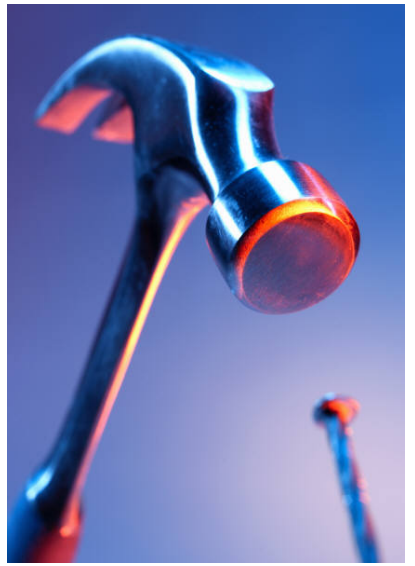
```
sudo systemctl stop couchbase-server
```

- Start 3 Docker containers with Couchbase already installed.

```
sudo docker run -d --name couchbase-1 -p 8091-8094:8091-8094\
-p 11210-11211:11210-11211 couchbase

sudo docker run -d --name couchbase-2 couchbase
sudo docker run -d --name couchbase-3 couchbase
```

- Get the IP of your first node with Docker.

```
sudo docker inspect couchbase-$i | grep IPAddress
```



Couchbase Docker Repository:
https://hub.docker.com/_/couchbase/

Cluster Operations: Start a Cluster with Docker

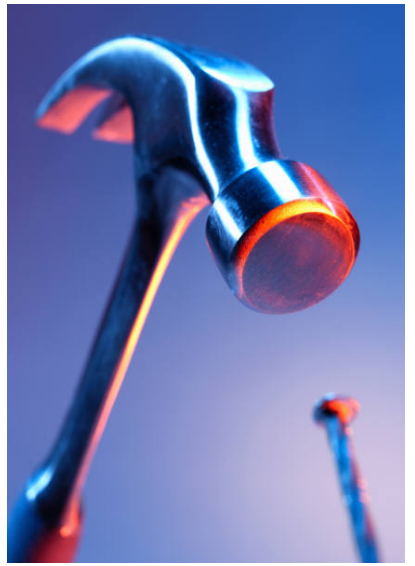
Perform the following steps:

- Check you can access the Couchbase CLI

```
sudo docker exec -it couchbase-1 bin/bash
```

- Test if all nodes are reachable

```
curl http://<IP couchbase-1>:8091/pools
curl http://<IP couchbase-2>:8091/pools
curl http://<IP couchbase-3>:8091/pools
```



- You should get something like:

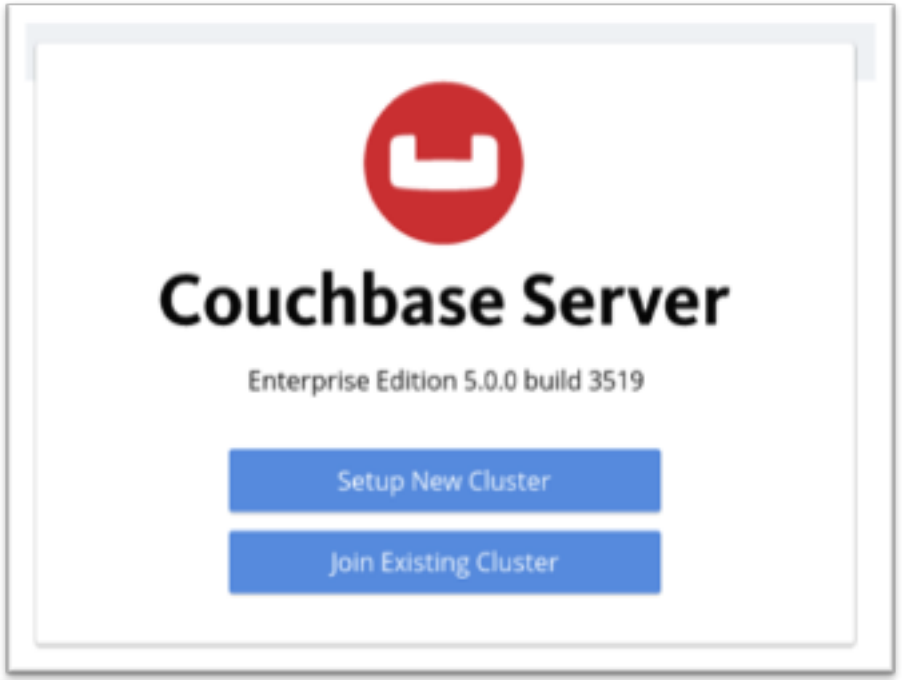

```
{"isAdminCreds":true,"isROAdminCreds":false,"isEnterprise":true,"pools":[],"settings":[],"uuid":[
],"implementationVersion":"5.0.0-3519-
enterprise","componentsVersion":{"lhttpc":"1.3.0","os_mon":"2.2.14","public_key":"0.21","asn1":"2
.0.4","kernel":"2.16.4","ale":"5.0.0-3519-enterprise","inets":"5.9.8","ns_server":"5.0.0-3519-
enterprise","crypto":"3.2","ssl":"5.3.3","sasl":"2.3.4","stdlib":"1.19.4"}}
```

Cluster Operations: Start a New Cluster with Docker

Perform the following steps:

- Setup a **New Cluster** via the UI
  - ClusterName = Cluster_3_Nodes
  - User: Administrator (pwd = couchbase)
  - HostName = <IP couchbase-1>
    (You get the IP with docker inspect and it should 172.17.0.2)
  - RAM Data Service = 1024 MB
  - RAM Index Service = 256 MB
  - RAM FTS Service = 256 MB

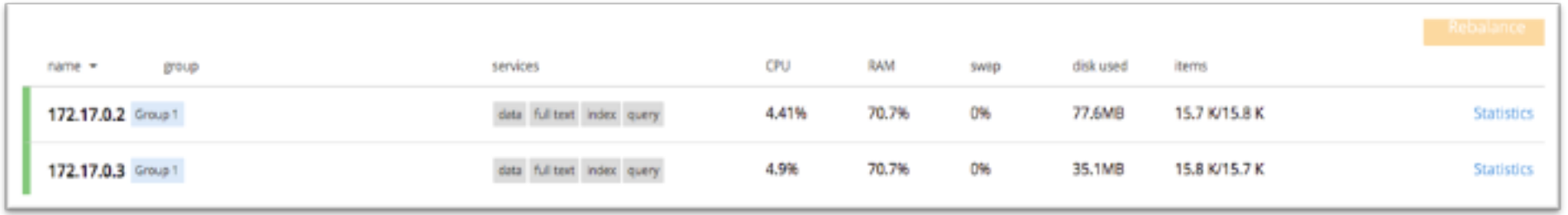
- Load the travel-sample bucket.
  Settings => Sample buckets.



Cluster Operations: Add the 2nd Node via UI

Perform the following steps:

- Add the 2nd node via the UI with IP of couchbase-2
  - IP from docker inspect on couchbase-2
  - Username = Administrator
  - Password = couchbase
  - Load all Services
- Add Server
- Rebalance.



Cluster Operations: Add the 3rd node

- On the 3rd node execute the following command

(you can log-in by using docker exec -it couchbase-3 /bin/bash)

```
/opt/couchbase/bin/couchbase-cli server-add --server-add=<IP couchbase-3>
--server-add-username=Administrator --server-add-password=couchbase
--group-name="Group 1" --cluster=<IP couchbase-1>:8091
--user=Administrator --password=couchbase
```

- Don't forget to rebalance!
  - Perform the Rebalance again via the UI
  - BTW: The CLI command 'couchbase-cli rebalance' can be used to invoke it from the command line
  - Which Service role was enabled on the 3rd node?

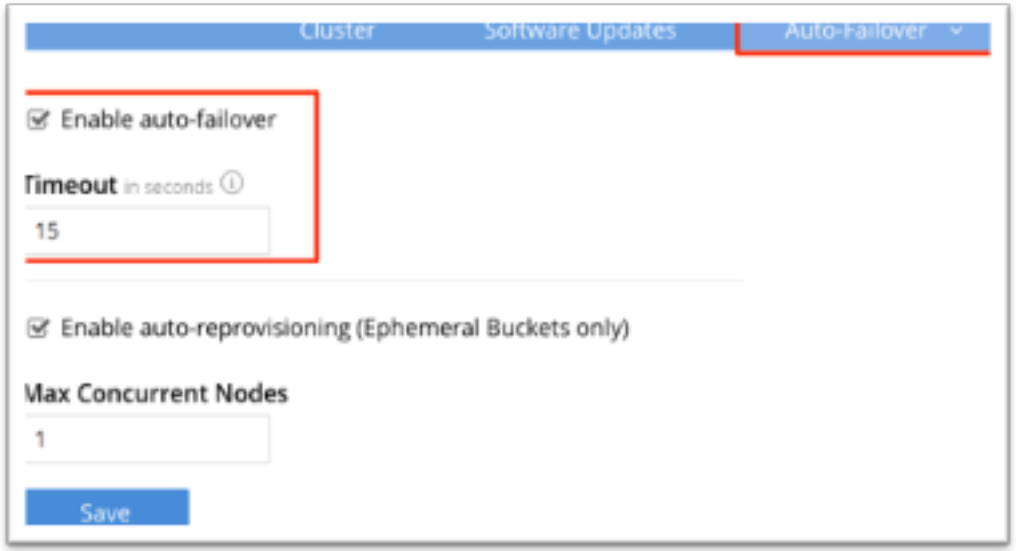
- *Optional: To remove node 3 from the cluster (to be added back after)*

```
couchbase-cli rebalance -c <IP couchbase-1>:8091 --server-remove=<IP couchbase-3>
--user=${admin user} --password=${password}
```

Cluster Operations: Auto-Failover

Perform the following steps:

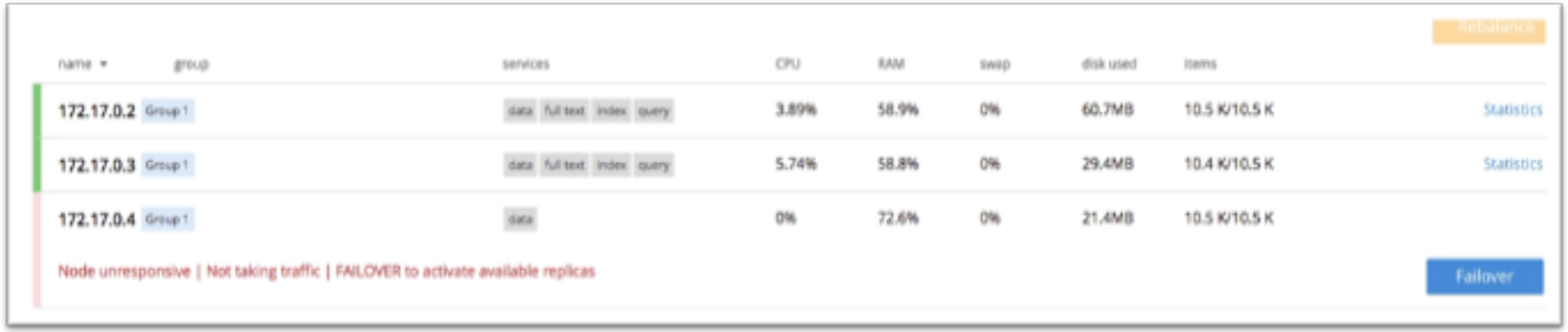
- Enable Auto-Failover in the Cluster to 15s



- Stop couchbase service on Node 3 to simulate a failure.

```
sudo docker stop couchbase-3
```

- Monitor the console on the tab Servers. (Is the bucket 100% available?)



Cluster Operations: Recover from a Failure

Perform the following steps:

- Restart the failing node

```
sudo docker start couchbase-3
```

- Monitor the UI “Servers” tab.
- Couchbase should be back as reachable.
- You have 2 options:
  - Full Recovery (Erase RAM and restore from replicas)
  - Delta Recovery (Compare and recover mutations which happened after failures.

- Rebalance

Cluster Operations: Stop Couchbase on Docker

Perform the following steps:

- Stop Couchbase on the 3 containers

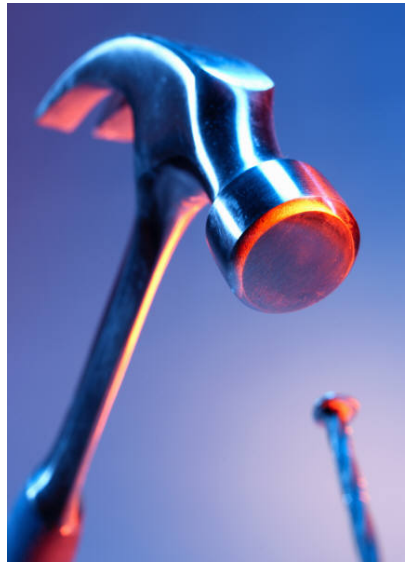
```
sudo docker stop couchbase-1
sudo docker stop couchbase-2
sudo docker stop couchbase-3
```

- Delete the containers

```
sudo docker rm couchbase-1
sudo docker rm couchbase-2
sudo docker rm couchbase-3
```

- Check the containers are not anymore running

```
sudo docker ps
```



Couchbase Docker Repository:
https://hub.docker.com/_/couchbase/

5 | Security

Security: Create a User with limited permissions

Perform the following steps:

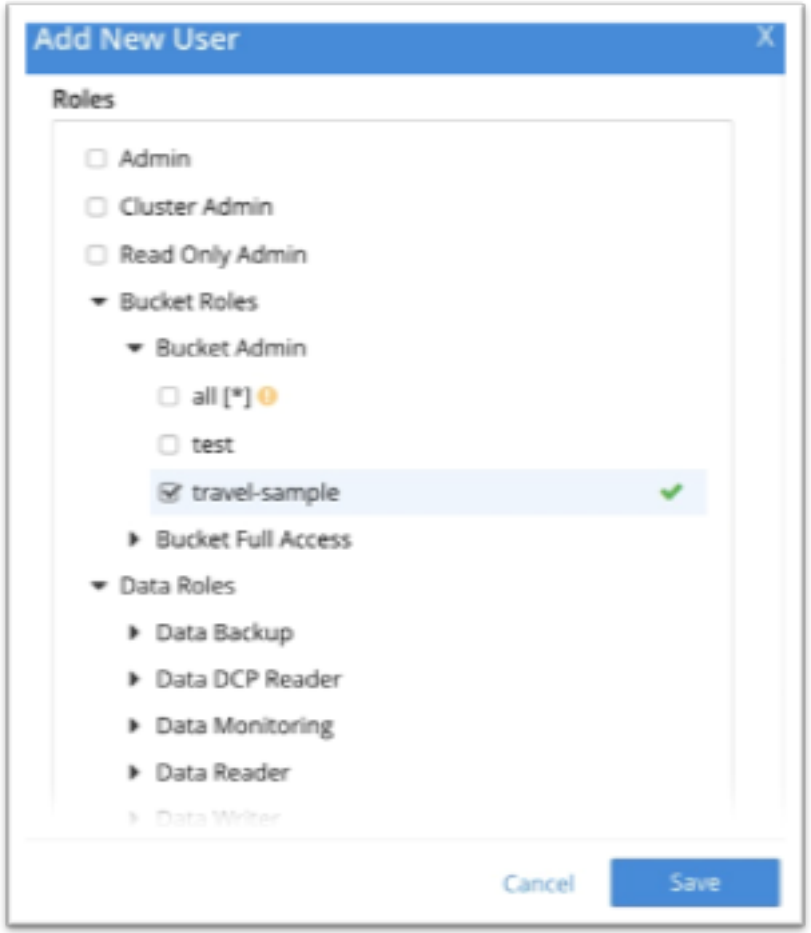
- Start Couchbase

```
sudo systemctl start couchbase-server
```

- Create a User with “Bucket Admin” role on travel-sample.
- Logout and Login with the new user.
  - Can you change the settings of test bucket?

- Log back as Administrator.
- Grant more permissions to your user.

```
GRANT Cluster_Admin TO `Ludo`
SELECT * FROM system:user_info
```



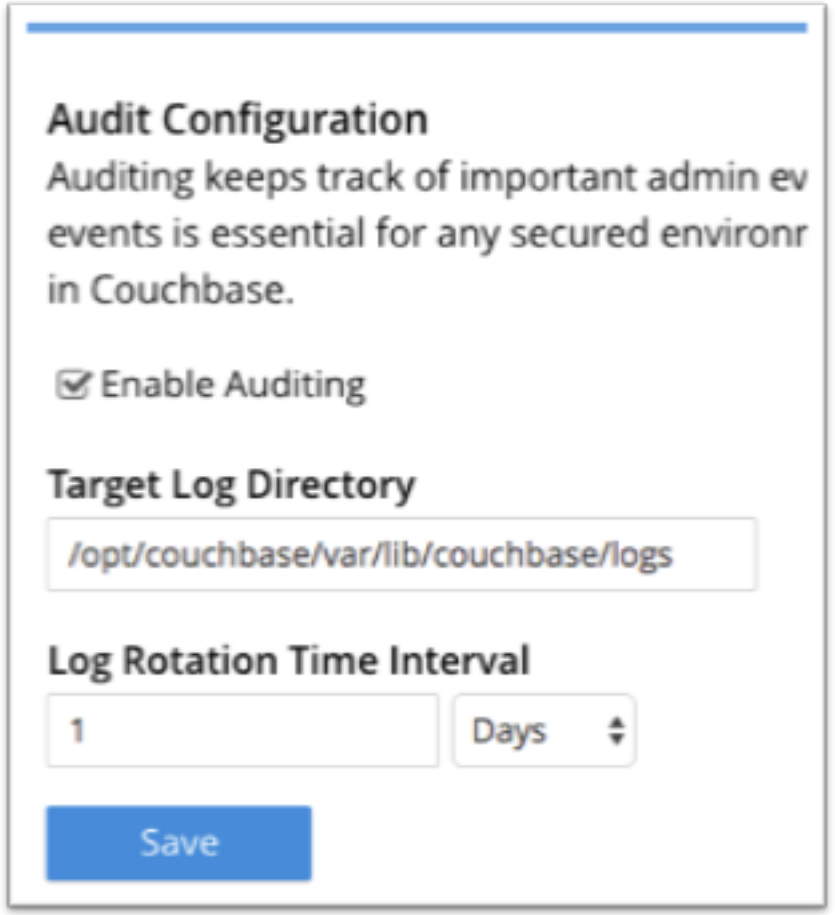
https://developer.couchbase.com/documentation/server/current/security/concepts-rba-for-apps.html

Security: Enable Auditing

Perform the following steps:

- Enable Auditing in the Security tab
- Perform some Administration tasks
  - Change Auto-Compaction to 20%.
- Check the Audit.log file.


```
{"timestamp":"2017-11-
10T17:33:26.373190+01:00","real_userid":{"source":"internal","us
er":"couchbase"},"auditd_enabled":true,"descriptors_path":"/opt/
couchbase/etc/security","hostname":"localhost.localdomain","log_
path":"/opt/couchbase/var/lib/couchbase/logs","rotate_interval":
86400,"version":1,"id":4096,"name":"configured audit
daemon","description":"loaded configuration file for audit
daemon"}
```

https://developer.couchbase.com/documentation/server/current/security/security-auditing.html

6 | Backup & Restore

Backup Couchbase

Perform the following steps in order to backup some data

- Create a target folder

```
cd /tmp
mkdir cb-backup
cd /opt/couchbase/bin
```

- Prepare the backup archive

```
/opt/couchbase/bin/cbbackupmgr config  --archive /tmp/cb-backup --repo workshop
```

- Backup the data twice and then use the list command to list the increments!

```
cbbackupmgr backup -a /tmp/cb-backup -r workshop -c http://localhost:8091
-u Administrator -p couchbase

cbbackupmgr list -archive /tmp/cb-backup --repo workshop
```

Restore Couchbase

Perform the following steps in order to restore some data

- Delete a document in the bucket travel-sample via the UI
- Get the count of document – 31590 (after delete)

- Restore the database.

```
cbbackupmgr list --archive /tmp/cb-backup --repo workshop

cbbackupmgr restore --archive /tmp/cb-backup --repo workshop -c http://localhost:8091 -u
Administrator -p couchbase --start 2017-11-10T18_07_25.462463124+01_00 --end 2017-11-
10T18_07_25.462463124+01_00
```

- Does the document come back? => Try again with --force-updates

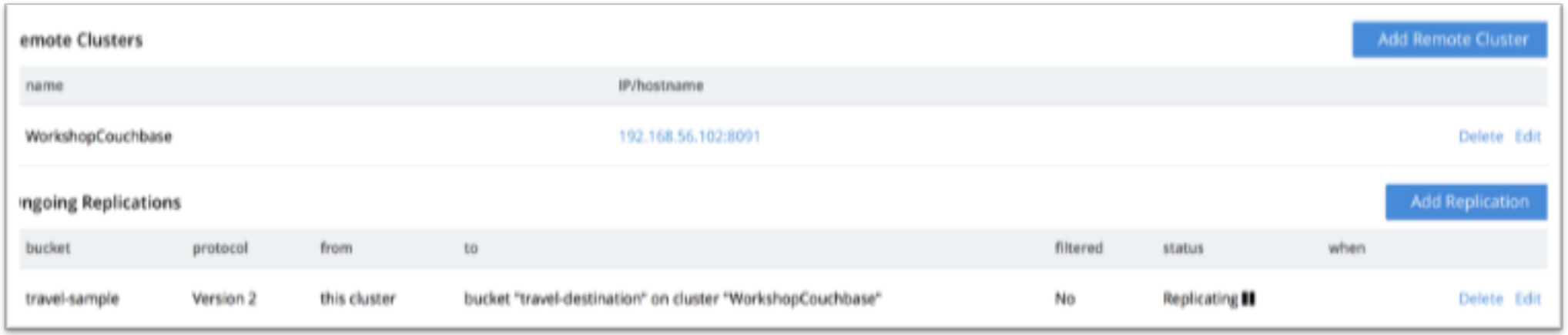
```
cbbackupmgr restore --archive /tmp/cb-backup --repo workshop -c http://localhost:8091 -u
Administrator -p couchbase --start 2017-11-10T18_07_25.462463124+01_00 --end 2017-11-
10T18_07_25.462463124+01_00 --force-updates
```

7 | XDCR

XDCR: Replicate a Bucket

Let's XDCR the travel-sample bucket to a new bucket "travel-destination"

- Create a new bucket "travel-destination"
  - RAM Quota = 100MB
  - No Replica
  - Conflict Resolution: Sequence Number
  - Flush: Enable
- Add a remote cluster (the local one)
  - Name of the Cluster: WorkshopCouchbase
  - IP of the local cluster.
- Add replication
  from "travel-sample" to "travel-destination"
  - Default Settings



XDCR: Replicate a Bucket – Update Source

Let's update a document in the source cluster.

- Update 1 document in travel-sample
  - Select 1 document "airline_10"
  - Check the metadata this document.
  - Save the revision (CAS) id.

- Create a new document in travel-sample
  - ID = airline_XX

```
{
  "id": "airline_10",
  "rev": "1-14f33c01fd1f00000000000002000000",
  "expiration": 0,
  "flags": 33554432
}
```

```
{
  "id": "airline_10",
  "rev": "2-14f671d35bd20000000000002000000",
  "expiration": 0,
  "flags": 33554432
}
```

- Check the "airline_10" document in the "travel-destination" bucket.
- Check the document count on both buckets.

- Bonus: Play with bi-directional replication.

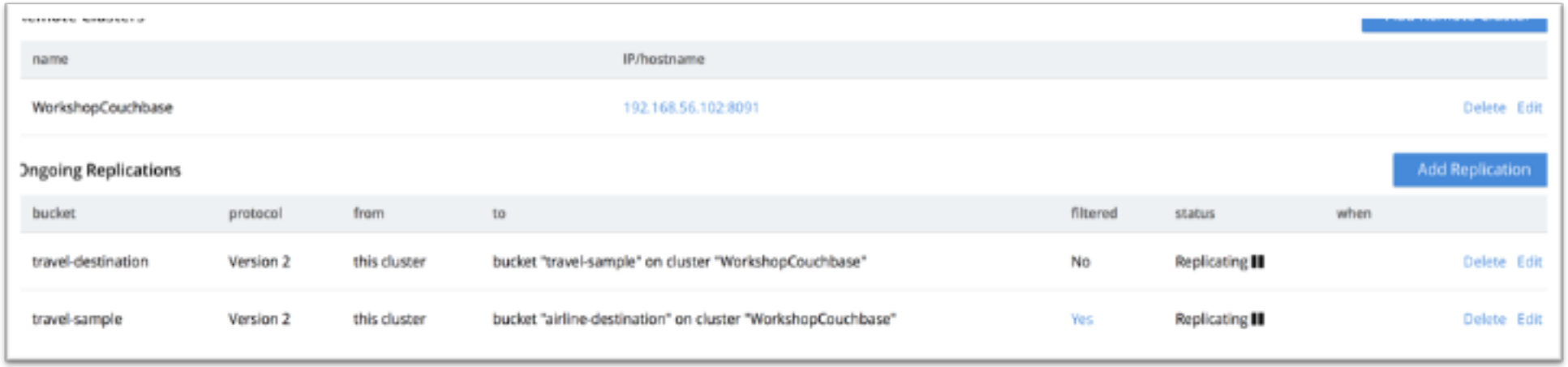
XDCR: Replicate a Bucket => Filtering on Airlines

Let's XDCR all document with type "airlines" from the travel-sample bucket to a new bucket "airline-destination"

- Create a new bucket "airline-destination"
  - RAM Quota = 100MB
  - No Replica
  - Conflict Resolution: Sequence Number
  - Flush: Enable

- Add replication
  from "travel-sample" to "airline-destination"
  - Enable **Advance Filtering**.
  - Enter a regular expression **"airline*"**
  - **Start Replication**
  - Run the **following query** & compare the count with the number of documents in "airline-destination"

```
SELECT COUNT(*) FROM `travel-sample` WHERE meta().id LIKE "airline%"
```

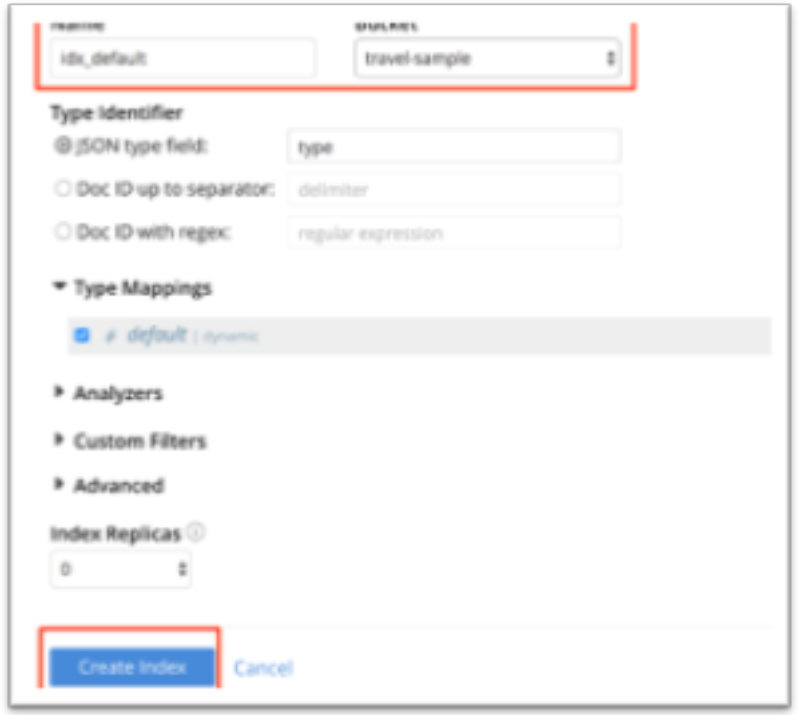


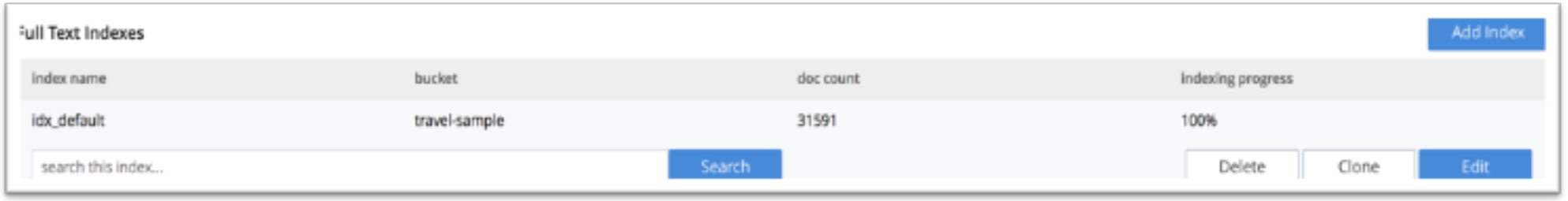
8 | FTS

FTS: Create a default Index

Let's build the default Index on `travel-sample` bucket.

- Build a new default Index
  - Give index name "idx_default"
  - Select bucket "travel-sample"
  - Create Index
- Observe time to create default index. (~60s)
- Check the size on disk of the default index. (~ 915Mb)
- Search default index & review search results

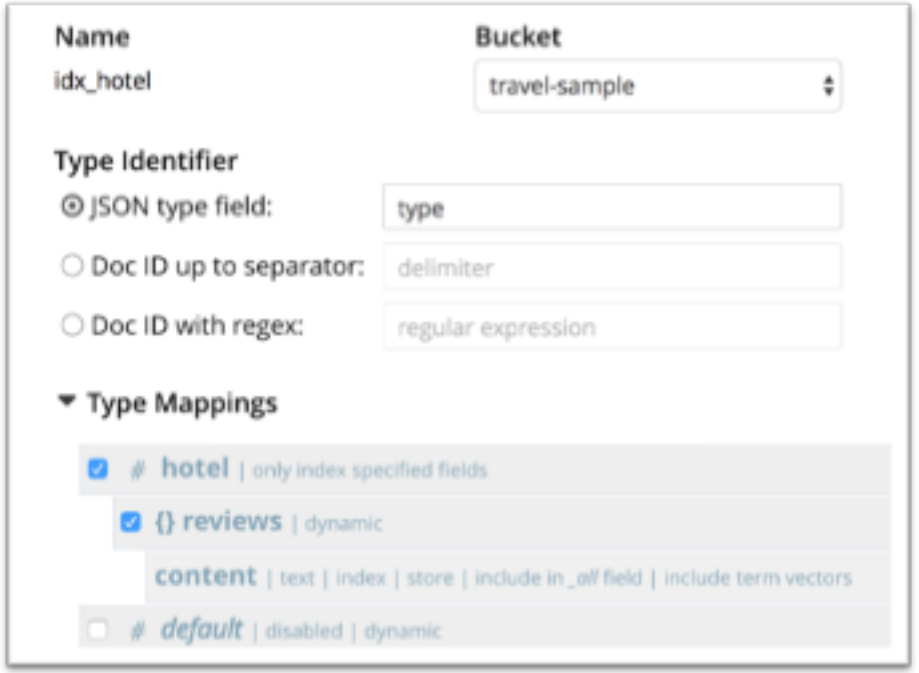




Note: Default index are not recommended in a production environment

FTS: Create a custom Index on type = hotel

Let's build a new Index on `travel-sample` bucket.



- Build a new Index
  - Give index name "idx_hotel"
  - Select bucket "travel-sample"
  - Add a Type mapping = hotel (only index specified fields)
  - Disable the default mapping.
  - Add a child mapping on reviews (array)
  - Add a field mapping on content
    (only index specified fields and store)
- Observe time to create default index. (~5s)
- Check the size on disk of the default index. (~ 79Mb)
- Search "friendly" in the UI with the hotel index & review search results

Bonus: Run the same search with the REST API. (you can find the curl command in the UI)

Thank you

Couchbase