

# **CS300 Couchbase NoSQL Server Administration**

## **Lab 3 Exercise Manual**



**Release: 4.5**

Revised: Sept 1st, 2016



## Lab #3: Grow Cluster to 6 Nodes

**Objective:** This 1-hour lab will walk you through the process of growing the Couchbase cluster to 6 nodes and rebalancing the cluster to distribute the vBuckets/partitions evenly across all 6 nodes. We will also create two server groups and place 2 nodes in each server group. Finally, you'll learn how to increase the number of replicas for a bucket to adjust its replication factor.

Please send any comments or corrections in this lab or future labs to [cls@couchbase.com](mailto:cls@couchbase.com)

**Overview:** The following high-level steps are involved in this lab:

- Grow cluster to 6 nodes in one data center (Amazon region)
- Grow the cluster using CMD line, REST API and using the web GUI
- Explore the Cluster Map
- Rebalance a cluster
- Adjust the number of replicas
- Use the Web UI to see performance metrics
- Run 'cbstats' command to see active and replica vBuckets

### Connect to the 2<sup>nd</sup>, 3<sup>rd</sup> and 4<sup>th</sup> VMs using PuTTY:

Refer to the `Cluster-IPs` spreadsheet that the instructor gave you to get your public IPs for the 2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup>, 5<sup>th</sup> & 6<sup>th</sup> VMs to be used in the Cluster. In this section, you will establish a connection to your VMs using PuTTY.

Remember that the second IP in your list is being used as an application server and that you will start with the next available IP in the list (IP address number 3).

The 1<sup>st</sup> and 2<sup>nd</sup> VMs are in their own simulated failure domain, and we will assign them to a separate server group. Similarly, the 3<sup>rd</sup> and 4<sup>th</sup> VMs will be assigned to their own server group. Note that if this were a physical environment in a physical data center, then all the servers on a rack would fall within the same failure domain and therefore would be in the same server group.

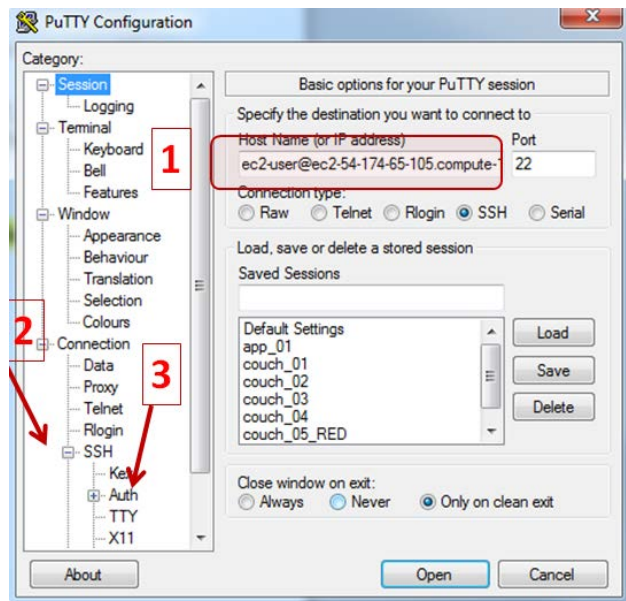
**Now, launch PuTTY and connect to the 2<sup>nd</sup> Couchbase Cluster VM(node 3 on your list).**

After starting PuTTY, enter the public hostname address of your 2<sup>nd</sup> Cluster VM into PuTTY. The connection type will be SSH and the port will be 22.

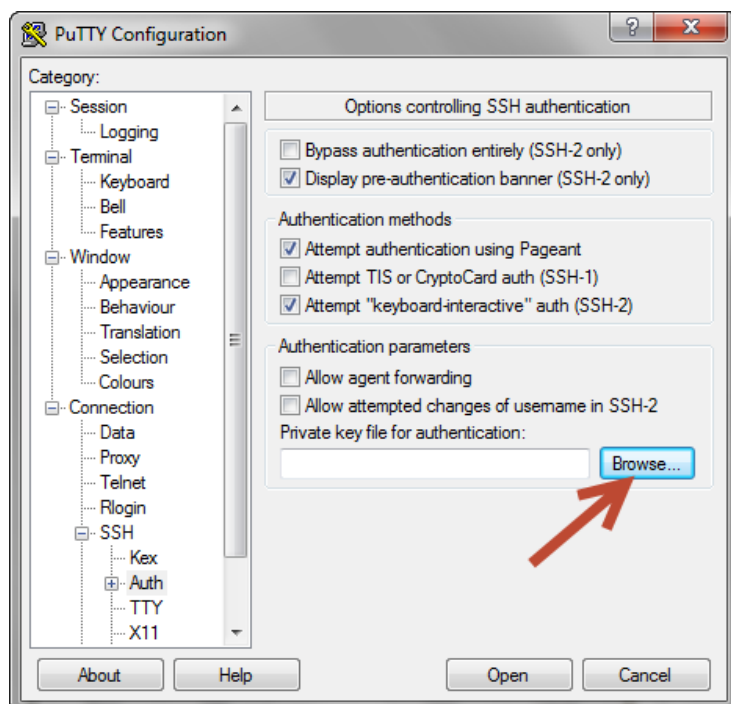


## Lab-3: Add 3 more nodes, page 3

Type **"ec2-user@public hostname"** with the public hostname that the instructor gave you for the 2<sup>nd</sup> Cluster VM into PuTTY and then **click on the + next to SSH** to expand its options and finally **select Auth**:



Click **Browse** to select the Private key file for authentication:

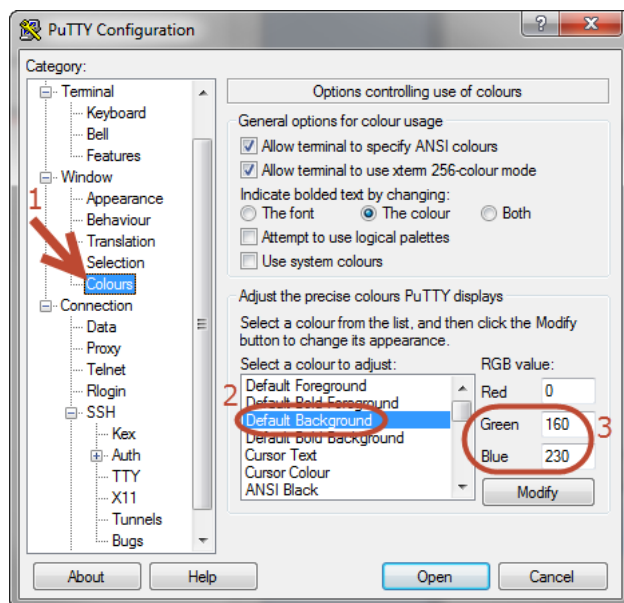




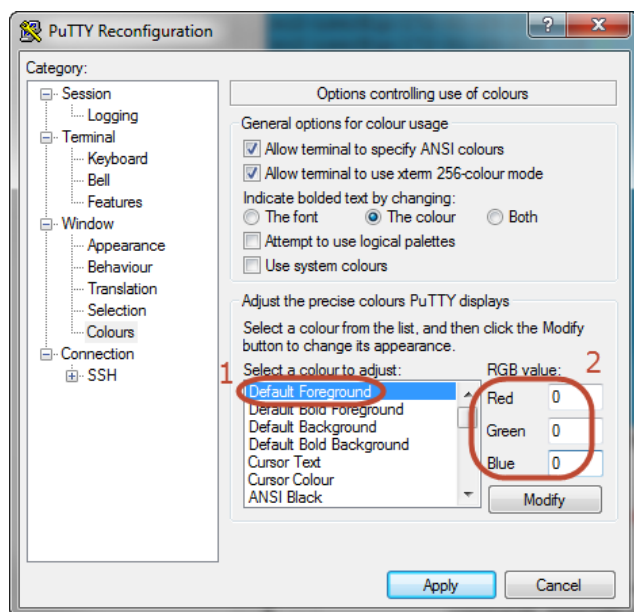
## Lab-3: Add 3 more nodes, page 4

Choose the **“Amazon-Private-Key.ppk”** file that the instructor provided you with.

In the left pane, **click on Colors**, then **choose Default Background** under “Select a colour to adjust” and alter the **Green RGB value to 160** and the **Blue RGB value to 230**.



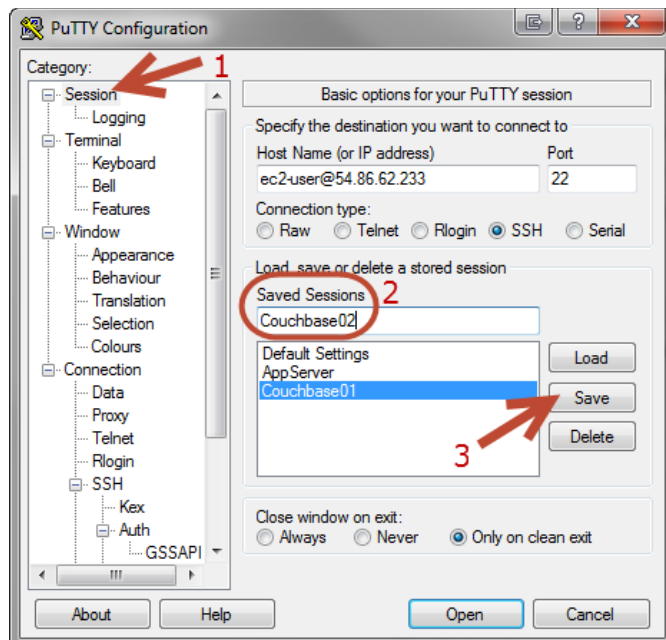
In the same window **choose Default Foreground** under “Select a colour to adjust” and alter the **Red, Green and Blue values to 0** to make the text color black.



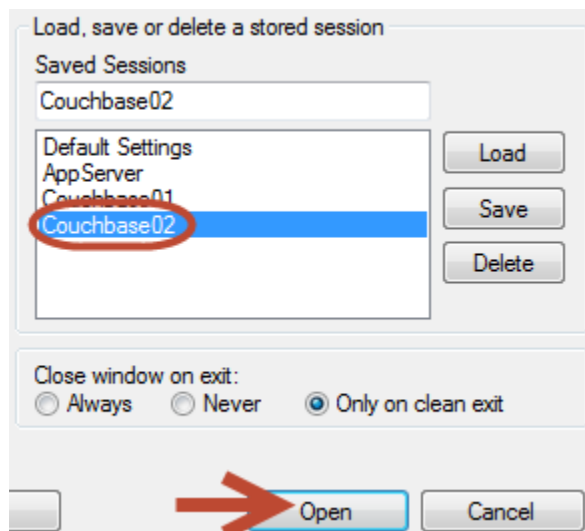


## Lab-3: Add 3 more nodes, page 5

Next, **click on Session** and type to **save the session as “CouchbaseXX”**, where XX is the # of your node from the hostname. Then **click on Save**. For example, here the session is being saved as “Couchbase02”:



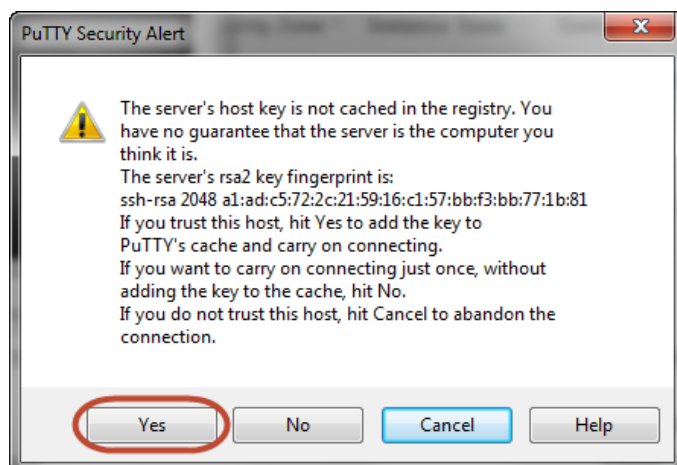
Now highlight the new server, for example here it is **Couchbase02** and click **Open** to connect to this VM:



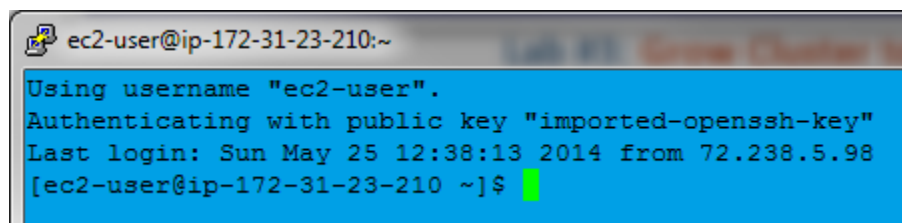
You will have to click **“Yes”** to a message about the server’s rsa2 key before a successful connection.



## Lab-3: Add 3 more nodes, page 6



You should now be successfully logged in to the 2<sup>nd</sup> (Light Blue) VM:



Become root and change the hostname of your VM to Couchbase02.

```
# sudo -i
```

```
# hostname Couchbase02
```

Log out of the putty session and log back in to have the putty terminal retitled for your session.

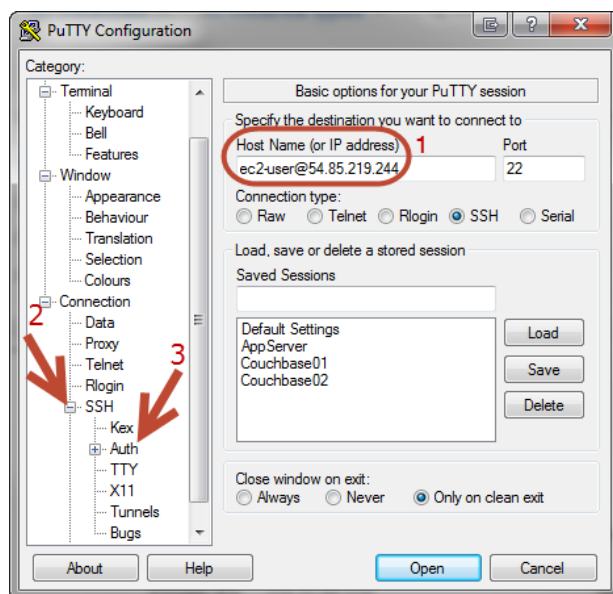
Repeat the same steps as above to connect to the 3<sup>rd</sup> VM, whose terminal color we will set to green.

Now, launch PuTTY and connect to the 3<sup>rd</sup> Couchbase VM.

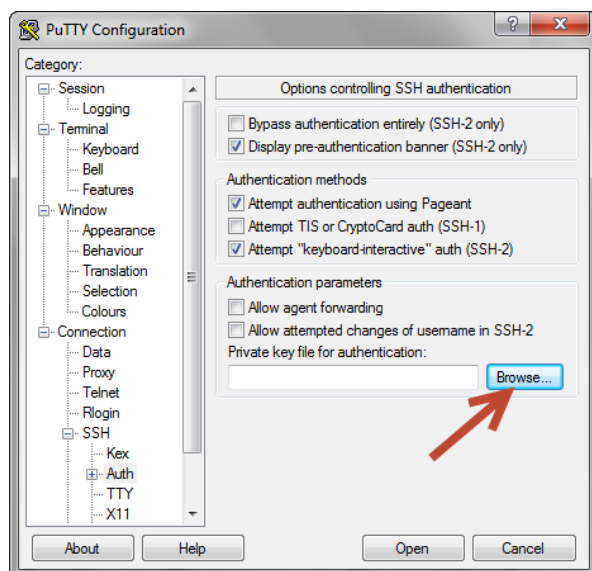
Type **“ec2-user@public hostname”** with the public hostname that the instructor gave you for the 3<sup>rd</sup> VM into PuTTY and then **click on the + next to SSH** to expand its options and finally **select Auth:**



## Lab-3: Add 3 more nodes, page 7



Click **Browse** to select the Private key file for authentication:

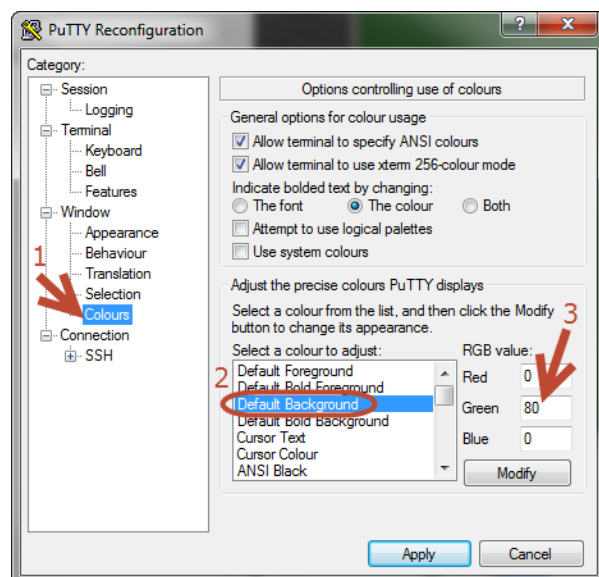


Choose the **“Amazon-Private-Key.ppk”** file that the instructor provided you with.

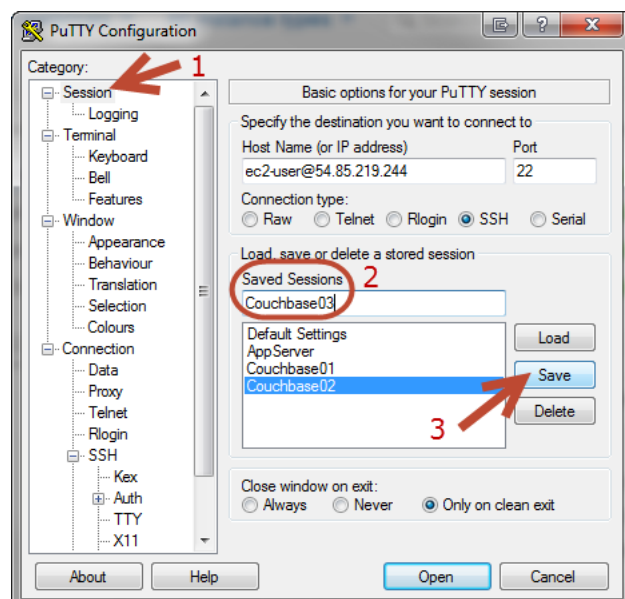
In the left pane, **click on Colors**, then **choose Default Background** under “Select a colour to adjust” and alter the **Green RGB value to 80**.



## Lab-3: Add 3 more nodes, page 8



Next, **click on Session** and type to **save the session as “CouchbaseXX”**, where XX is the # of your node from the hostname. Then **click on Save**. For example, here the session is being saved as “Couchbase03”:

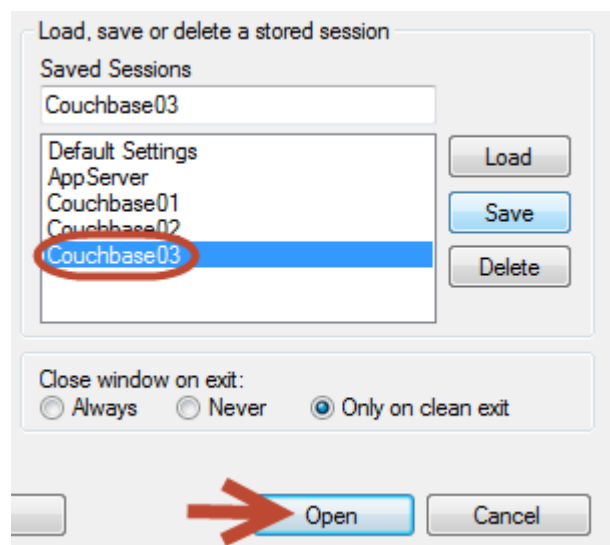


Now highlight the new server, for example here it is **Couchbase03** and click **Open** to connect to this VM:

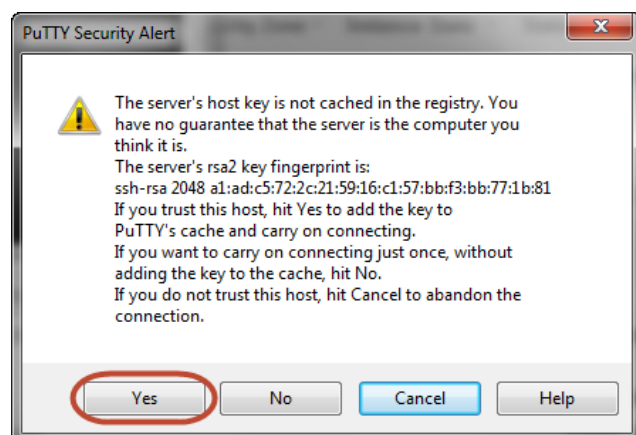




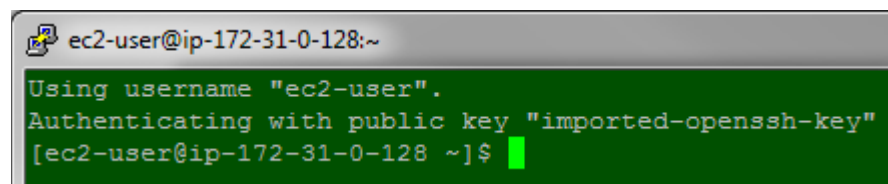
## Lab-3: Add 3 more nodes, page 9



You will have to click **"Yes"** to a message about the server's rsa2 key before a successful connection.



You should now be successfully logged in to the 3<sup>rd</sup> (Green) VM:



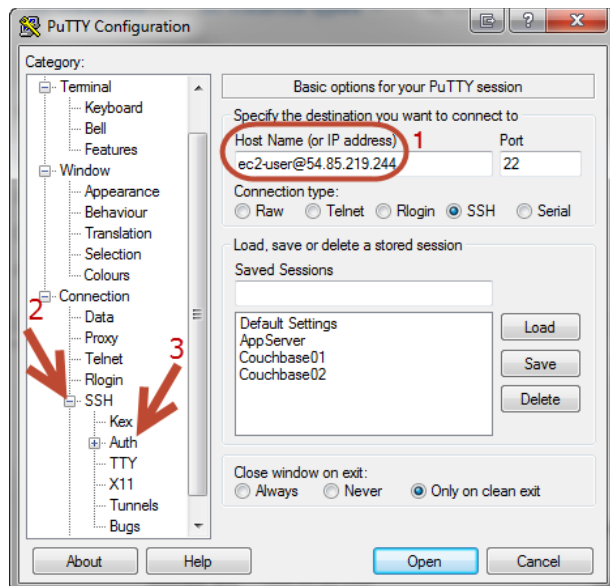
Repeat the same steps as above to connect to the 4<sup>th</sup> VM, whose terminal color we will set to yellow.

Now, launch PuTTY and connect to the 4<sup>th</sup> Couchbase VM.

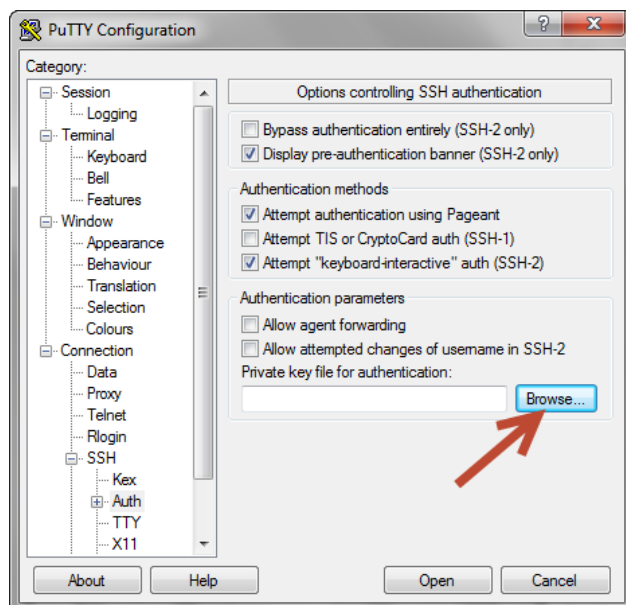


## Lab-3: Add 3 more nodes, page 10

Type **"ec2-user@public hostname"** with the public hostname that the instructor gave you for the 4th VM into PuTTY and then **click on the + next to SSH** to expand its options and finally **select Auth**:



Click **Browse** to select the Private key file for authentication:

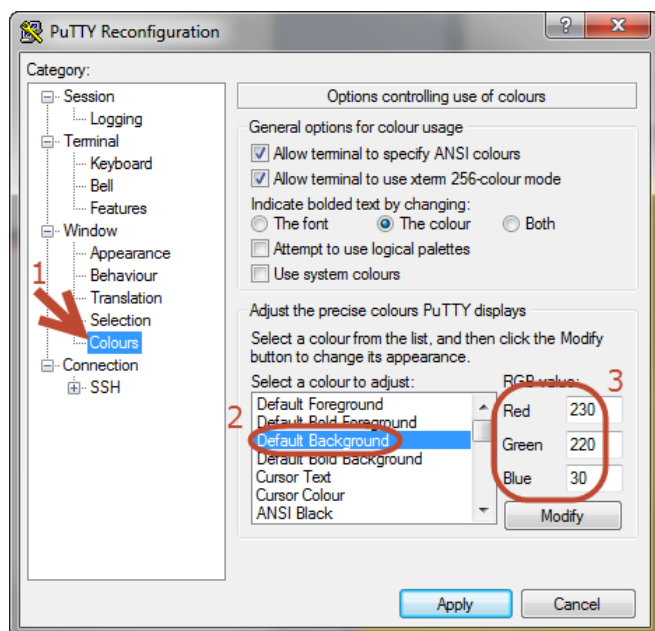


Choose the **"Amazon-Private-Key.ppk"** file that the instructor provided you with.

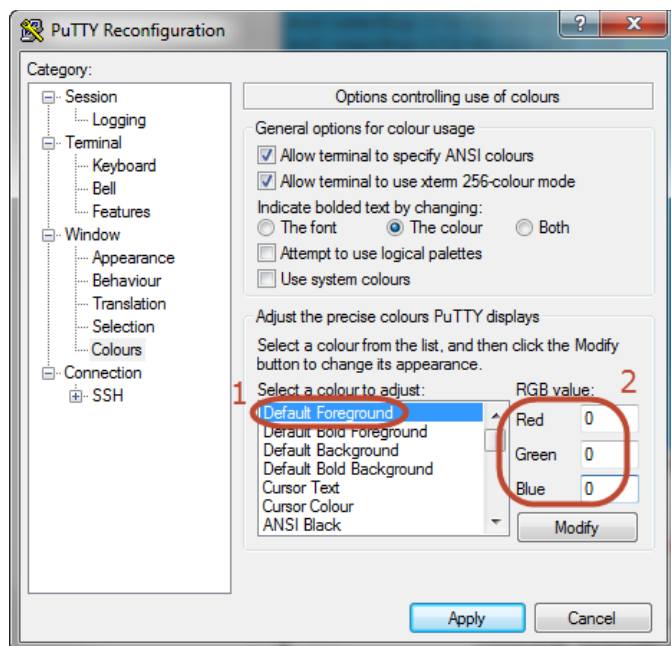


## Lab-3: Add 3 more nodes, page 11

In the left pane, **click on Colors**, then **choose Default Background** under “Select a colour to adjust” and alter the **Red, Green and Blue RGP values to 230, 220 and 30 respectively**.



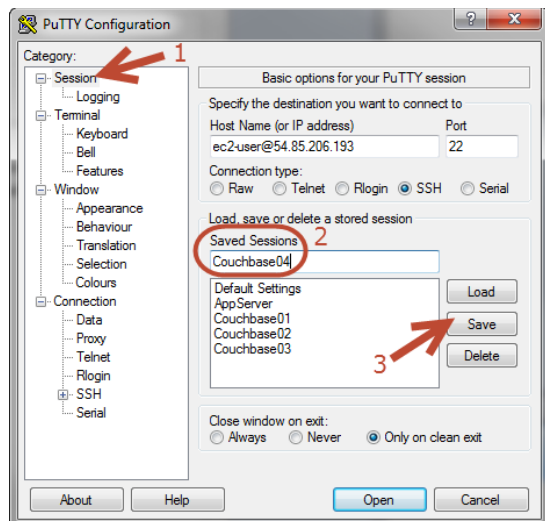
In the same window **choose Default Foreground** under “Select a color to adjust” and alter the **Red, Green and Blue values to 0** to make the text color black.



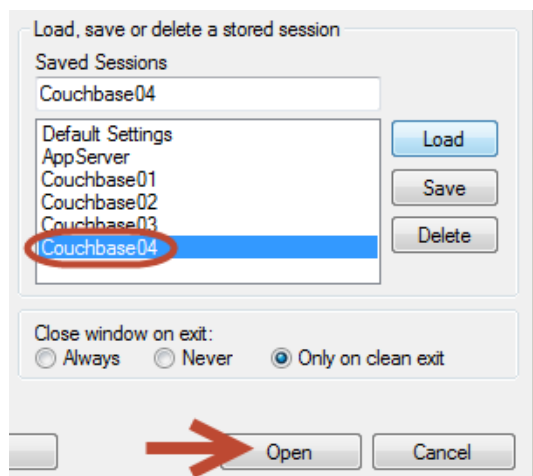


## Lab-3: Add 3 more nodes, page 12

Next, **click on Session** and type to **save the session as “CouchbaseXX”**, where XX is the # of your node from the hostname. Then **click on Save**. For example, here the session is being saved as “Couchbase04”:



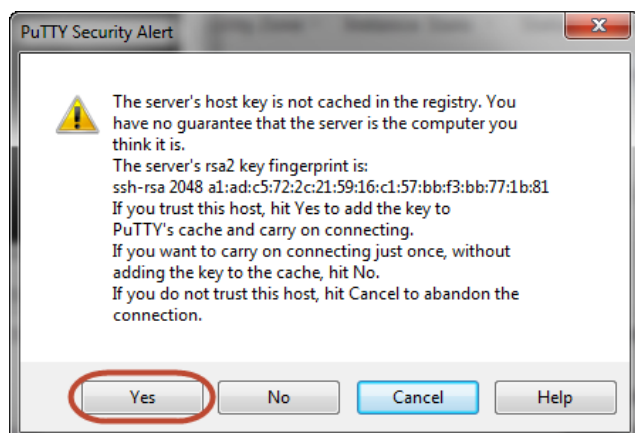
Now highlight the new server, for example here it is **Couchbase04** and click **Open** to connect to this VM:



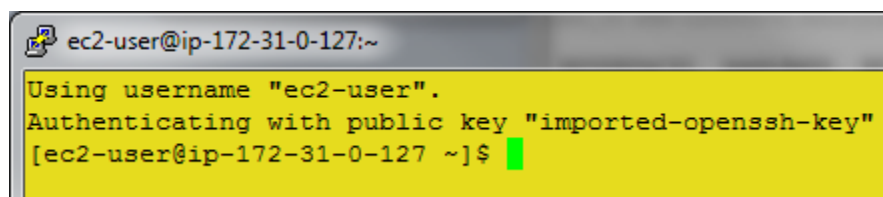
You will have to click **“Yes”** to a message about the server’s rsa2 key before a successful connection.



## Lab-3: Add 3 more nodes, page 13



You should now be successfully logged in to the 4<sup>th</sup> (Yellow) VM:



Repeat this set of procedures for nodes 5 and 6 using the following colors

Node 5- Pink/ Couchbase VM #5

Node 6- Watermelon / Couchbase VM #6

At this point, you should have 7 open PuTTY/Terminal sessions:

- Dark Blue / Couchbase01 VM #1
- Light Blue / Couchbase02 VM #2
- Green / Couchbase03 VM #3
- Yellow / Couchbase04 VM #4
- Pink/ Couchbase05 VM #5
- Watermelon / Couchbase06 VM #6
- Black / AppServer



## Configure the 6 new nodes and install Couchbase Server on them:

In this section, we will do the following on the 6 new nodes (light blue, green and yellow PuTTY windows):

Install Couchbase Server

The steps below should be repeated five times. Once on the 2<sup>nd</sup> node and once on the 3<sup>rd</sup> node and once on the 4<sup>th</sup> node and fifth nodes and sixth nodes. First run through it on the 2<sup>nd</sup> node, then come back around and implement this on the 3<sup>rd</sup> and 4<sup>th</sup> node 5<sup>th</sup> and 6<sup>th</sup>.

So, let's get going. Switch to the 2<sup>nd</sup> node (light blue window) and implement the following steps. Be careful not to typo or hit the wrong key!

(Loop starts here)

Turn off swapping for the running system, but first switch to root user:

```
[ec2-user@Couchbase02 ~]$ sudo -s
[root@ Couchbase02 ec2-user]# echo 0 > /proc/sys/vm/swappiness
```

Then permanently make this change in the sysctl.conf file, so the change persists after a reboot and exit root:

```
[root@ Couchbase02 ec2-user]# echo '' >> /etc/sysctl.conf
[root@ Couchbase02 ec2-user]# echo '#Set swappiness to 0 to avoid swapping' >> /etc/sysctl.conf
[root@ Couchbase02 ec2-user]# echo 'vm.swappiness = 0' >> /etc/sysctl.conf
```

Disable Transparent Huge Pages

In a production Couchbase cluster, it is very important to disable Transparent Huge pages on each node.

Disable THP on a running system

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
echo never > /sys/kernel/mm/transparent_hugepage/defrag
```

```
# yum install wget
```

```
# exit
```

Download Couchbase 4.5.0 EE:

```
[ec2-user@Couchbase0X ~]$ wget
http://packages.couchbase.com/releases/4.5.0/couchbase-server-enterprise-4.5.0-centos7.x86_64.rpm
```



## Lab-3: Add 3 more nodes, page 15

```
--2016-05-10 14:32:24-- http://packages.couchbase.com/releases/4.5.0-DP1/couchbase-server-
enterprise-4.5.0-DP1-centos7.x86_64.rpm
Resolving packages.couchbase.com (packages.couchbase.com)... 54.231.97.227
Connecting to packages.couchbase.com (packages.couchbase.com)|54.231.97.227|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 100473860 (96M) [application/x-rpm]
Saving to: 'couchbase-server-enterprise-4.5.0-DP1-centos7.x86_64.rpm'

100%[=====>] 100,473,860 56.4MB/s in 1.7s

2016-05-10 14:32:26 (56.4 MB/s) - 'couchbase-server-enterprise-4.5.0-DP1-centos7.x86_64.rpm'
saved [100473860/100473860]
```

**Install Couchbase (note, this command might take 1-2 minutes to complete):**

```
[ec2-user@Couchbase0X ~]$ sudo rpm --install couchbase-server-
enterprise-4.5.0-centos7.x86_64.rpm
```

```
Minimum RAM required : 4 GB
System RAM configured : 3.45 GB
```

```
Minimum number of processors required : 4 cores
Number of processors on the system : 2 cores
```

```
Reloading systemd: [ OK ]
Starting couchbase-server (via systemctl): [ OK ]
```

You have successfully installed Couchbase Server.  
Please browse to <http://Couchbase0X:8091/> to configure your server.  
Please refer to <http://couchbase.com> for additional resources.

Please note that you have to update your firewall configuration to allow connections to the following ports: 11211, 11210, 11209, 4369, 8091, 8092, 8093, 9100 to 9105, 9998, 18091, 18092, 11214, 11215 and from 21100 to 21299.

By using this software you agree to the End User License Agreement.  
See </opt/couchbase/LICENSE.txt>...

**Note: in Couchbase 4.1.0 autostart was dependent on RH7.1 symbolic links that are not present in RH7.2 Please use the following to start couchbase. (this will be remedied in Couchbase 4.1.1 and above)**

**[/opt/couchbase/etc/couchbase\\_init.d](/opt/couchbase/etc/couchbase_init.d) start**

Starting couchbase-server

**[/opt/couchbase/etc/couchbase\\_init.d](/opt/couchbase/etc/couchbase_init.d) stop**

Stopping couchbase-server

**[/opt/couchbase/etc/couchbase\\_init.d](/opt/couchbase/etc/couchbase_init.d) status**

Obtaining system status

**After the install finishes, wait 30 seconds, then check the status of the Couchbase Server:**

```
[root@ Couchbase0X ec2-user]# sudo /etc/init.d/couchbase-server status
couchbase-server is running
```



## Lab-3: Add 3 more nodes, page 16

If you just finished working on the 2<sup>nd</sup> (light blue) node, then please scroll up a few pages and repeat these steps on the 3<sup>rd</sup> (green) or 4<sup>th</sup> (yellow) node from the part that says : (Loop starts here)

## Add the 2<sup>nd</sup> node into the Couchbase Cluster using the Web UI:

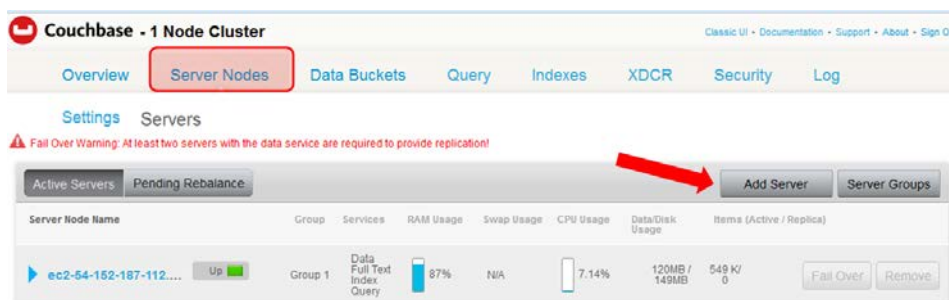
The Rack Awareness feature in the Enterprise Edition of Couchbase allows logical groupings of servers on a cluster where each server group physically belongs to a rack or Amazon availability zone. When configured properly, this feature allows administrators to specify that active and corresponding replica partitions be created on servers that are part of a separate rack or zone. In the event that a whole rack goes down, since the replica partitions are on separate racks, data will remain available.

It is recommended to keep the number of servers the same between the server groups. If an imbalance occurs where there is an unequal number of servers in one server group, the rebalance operation performs a “best effort” of evenly distributing the replica vBuckets across the cluster.

In this section, we will use the Web UI to add a 2nd node to the cluster under Group 1. In the section after this one, we will add the 3<sup>rd</sup> and 4<sup>th</sup> nodes to the cluster under Group 2 using the couchbase-cli tool.

**Connect to the Couchbase Server Web UI** running on node #1, click on **Server Nodes** at the top and then **Add Server**:

`http://<public hostname of 1st VM>:8091`



In the Server IP Address field, **type in the public hostname of Node #2 from the Cluster-IPs spreadsheet.**

**NOTE: make sure there are no leading spaces if you have cut and pasted.**





## Lab-3: Add 3 more nodes, page 17

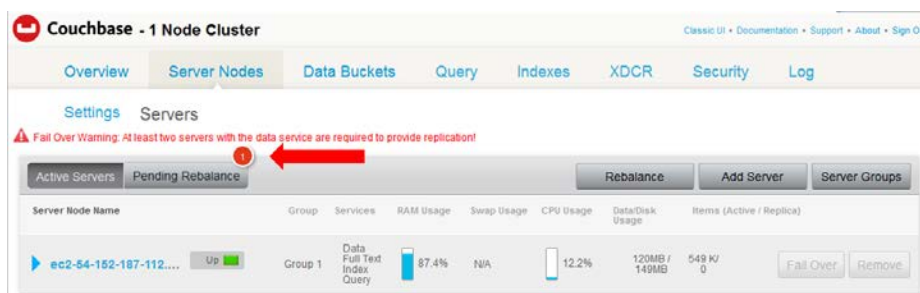
Also enter “**Administrator**” for the username and “**couchbase**” for the password:

Finally check the box for Data service

Click Add Server :

*Note: Warning about all previous data on this node will be lost when it joins the cluster.*

If you see a red #1 by “Pending Rebalance”, then the server has been successfully added:



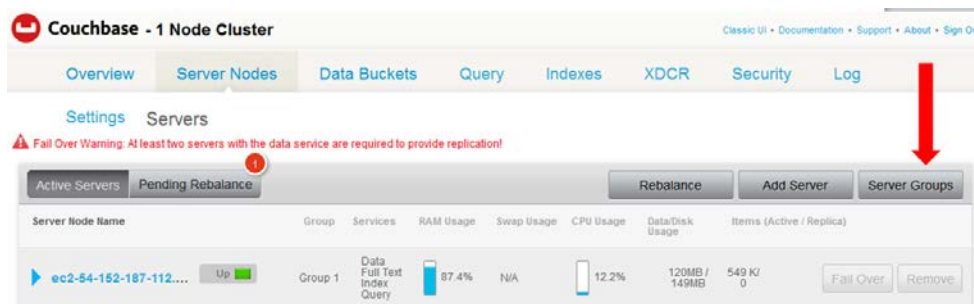
## Add the 3<sup>rd</sup> and 4<sup>th</sup> nodes into the Couchbase Cluster using the Couchbase-cli tool:

In this section, we will use the Couchbase-cli to add two additional nodes to the cluster. So, we will add the 3<sup>rd</sup> and 4<sup>th</sup> nodes to a new Server Group #2.



## Lab-3: Add 3 more nodes, page 18

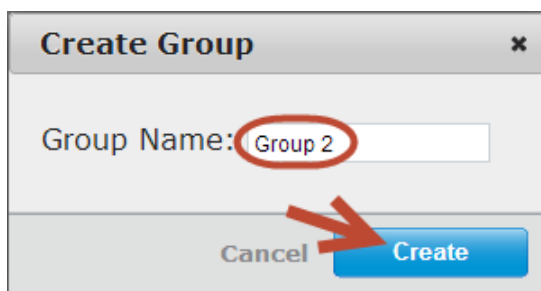
While you are still logged into the Couchbase Web UI, first create a new Server Group by clicking on **“Server Groups”**:



On the next page, click on **“Create Group”**:



On the popup, enter **“Group 2”** and click **“Create”**:



You should now see an empty Group 2:



## Lab-3: Add 3 more nodes, page 19

Server Groups

Apply Changes Create Group Server Nodes

Group 1 Rename Group

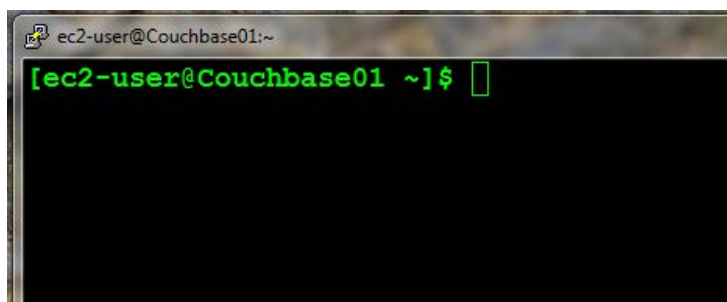
ec2-54-85-43-128.compute-1.amazonaws.com

ec2-54-86-106-120.compute-1.amazonaws.com

Group 2 Rename Group

This group is empty, [click to delete](#).

Go to the 1<sup>st</sup> node's PuTTY shell (dark blue window):



```
[ec2-user@Couchbase01 ~]# cd ~
```

Next, we need to construct the couchbase-cli command to add the 3<sup>rd</sup> node. The command you need to run will look identical to the command below, except you will have to modify the green and blue highlighted sections.

```
# /opt/couchbase/bin/couchbase-cli server-add --server-add=ec2-54-173-46-5.compute-1.amazonaws.com --server-add-username=Administrator --server-add-password=couchbase --group-name="Group 2" --services="data" --cluster=ec2-54-174-65-105.compute-1.amazonaws.com:8091 --user=Administrator --password=couchbase
```

The green part will be the public hostname of the 3<sup>rd</sup> server you want to add.

The blue part will be the public hostname of the 1<sup>st</sup> server.



The **purple** part will be the services you intend to offer on this node.

*Note that in the above command we are adding the 3<sup>rd</sup> node to Server Group 2.*

**Run the command that you constructed above (note the public hostnames in your command will be different!):**

```
[ec2-user@Couchbase01 ~]$ /opt/couchbase/bin/couchbase-cli server-add
--server-add=ec2-54-173-46-5.compute-1.amazonaws.com --server-add-
username=Administrator --server-add-password=couchbase --group-
name="Group 2" --services="data" --cluster=ec2-54-174-65-105.compute-
1.amazonaws.com:8091 --user=Administrator --password=couchbase
```

```
SUCCESS: add server 'ec2-54-173-46-5.compute-1.amazonaws.com:8091' to
group 'Group 2'
```

Upon a successful run of the command, you will see the success message above.

From the same dark blue/1<sup>st</sup> node window, run the server-add command again to add the 4<sup>th</sup> node to the cluster under Group 2.

**First construct the command:**

```
/opt/couchbase/bin/couchbase-cli server-add --server-add=ec2-54-174-
67-95.compute-1.amazonaws.com --server-add-username=Administrator --
server-add-password=couchbase --group-name="Group 2" --
services="data" --cluster=ec2-54-174-65-105.compute-
1.amazonaws.com:8091 --user=Administrator --password=couchbase
```

The **yellow** part will be the public hostname of the 4<sup>th</sup> server you want to add.

The **blue** part will be the public hostname of the 1<sup>st</sup> server.

The **purple** part will be the services you intend to offer on this node

**Run the command:**

```
[ec2-user@Couchbase01 ~]$ /opt/couchbase/bin/couchbase-cli server-add
--server-add=ec2-54-174-67-95.compute-1.amazonaws.com --server-add-
username=Administrator --server-add-password=couchbase --group-
name="Group 2" --services="data" --cluster=ec2-54-174-65-105.compute-
1.amazonaws.com:8091 --user=Administrator --password=couchbase
```

```
SUCCESS: add server 'ec2-54-174-67-95.compute-1.amazonaws.com:8091' to
group 'Group 2'
```



Another success message will mean that the 4<sup>th</sup> node is successfully added to the cluster.

*Note that it is also possible to add nodes to the cluster using the REST API. See the following documentation for details:*

<http://docs.couchbase.com/admin/admin/REST/rest-cluster-joinnode.html>

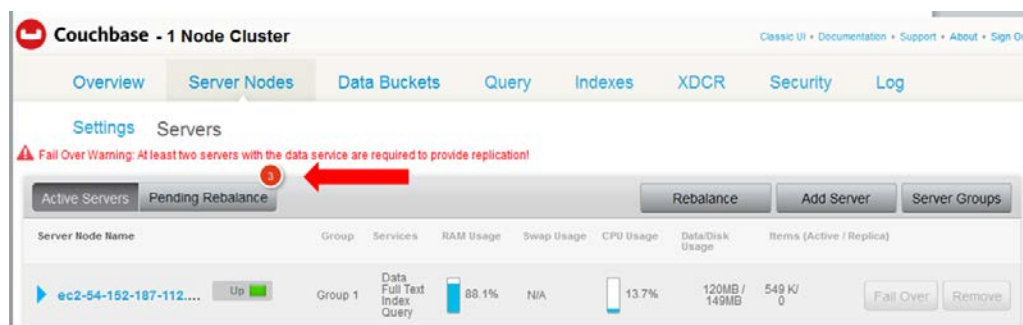
When using the REST API, the join command looks like this **(do not type this command anywhere, it's just for reference)**:

```
curl -u admin:password -d clusterMemberHostIp=192.168.0.1 \
-d clusterMemberPort=8091 \
-d user=admin -d password=password \
http://localhost:8091/node/controller/doJoinCluster
```

You'd have to replace the admin, password, and 192.168.0.1 values in the above example with your actual values.

## Rebalance the cluster:

In the Couchbase Web UI, you should now see the number 3 under Pending Rebalance:



This means that there are 3 nodes pending rebalance.

If you click on **Pending Rebalance**, you should see the 3 new nodes that you added with status of Pending Add. Also, notice that 1 server will join Group 1, while the other 2 will join Group 2. And that the services being offered by each node are Data service.



## Lab-3: Add 3 more nodes, page 22

Servers

⚠ Fail Over Warning: At least two servers are required to provide replication!

Server Node Name	Group	Services	RAM Usage	Swap Usage	CPU Usage	Data/Disk Usage	Items (Active / Replica)	
ec2-54-174-42-8.co...	Group 1	Data	9.88%	N/A	32.4%	N/A	0/0	Pending Add Cancel
ec2-54-173-46-5.co...	Group 2	Data	9.32%	N/A	0.99%	N/A	0/0	Pending Add Cancel
ec2-54-174-67-95.co...	Group 2	Data	9.23%	N/A	1%	N/A	0/0	Pending Add Cancel

Adding new nodes to Couchbase is actually a two-phase process. First you add the new nodes, then you rebalance the information across all the nodes. So far, we have only completed the first phase.

Simply adding 3 new nodes like we have done in this lab, merely configures the new nodes into the cluster. No actual changes are made to the cluster or data when configuring new nodes.

Next we will rebalance the cluster. During rebalance:

- vbuckets are moved from node to node to match the newly calculated vbucket map.
- So, the vBucket map is updated during the rebalance operation; the updated map is then sent to all the cluster participants, including the other nodes, any connected “smart” clients
- Rebalancing moves both the data stored in RAM, and the data stored on disk for each bucket, and for each node, within the cluster.
  - The cluster remains up, and continues to service and handle client requests.
  - The current vBucket map, used to identify which nodes in the cluster are responsible for handling client requests, is updated incrementally as each vBucket is moved. The updated vBucket map is communicated to Couchbase client libraries and enabled smart clients (such as Moxi), and allows clients to use the updated structure as the rebalance completes.

There are four primary reasons that you perform a rebalance operation:

1. Adding nodes to expand the size of the cluster.
2. Removing nodes to reduce the size of the cluster.
3. Reacting to a failover situation, where you need to bring the cluster back to a healthy state.
4. You need to temporarily remove one or more nodes to perform a software, operating system or hardware upgrade.

**Before starting the rebalance operation, run the `cbstats` command against the beer-sample bucket and `grep` for the `active_num` and `replica_num` variables to verify that the 1<sup>st</sup> node is responsible for all 1024 vBuckets for the bucket. Note, run this command on the 1<sup>st</sup> node of the cluster (dark blue Couchbase01) with the public hostname of the first node:**

```
[ec2-user@Couchbase01 ~]$ /opt/couchbase/bin/cbstats ec2-54-174-65-105.compute-1.amazonaws.com:11210 all -b beer-sample | egrep -i "active_num|replica_num"
```



## Lab-3: Add 3 more nodes, page 23

```
vb_active_num: 1024
vb_active_num_non_resident: 0
vb_replica_num: 0
vb_replica_num_non_resident: 0
```

Before rebalancing the cluster we will now flush the default bucket.  
Click blue triangle next the word default on the buckets tab.

The screenshot shows the Couchbase Data Buckets tab. The 'default' bucket is selected, and the 'Edit' button is highlighted with a red arrow. The table below shows the details for the buckets:

Bucket Name	Data Nodes	Item Count	Ops/sec	Disk Fetches/sec	RAM/Quota Usage	Data/Disk Usage
beer-sample	1	7303	0	0	40.4MB / 100MB	11.8MB / 14.9MB
default	1	510000	0	0	103MB / 200MB	44.8MB / 44.8MB
gamesim-sample	1	586	0	0	35.8MB / 100MB	4.97MB / 4.99MB
travel-sample	1	31569	0	0	72.6MB / 100MB	59.1MB / 84.5MB

Then click the edit button. On the pop up click enable Flush and click the save button.

The screenshot shows the 'Configure Bucket' dialog box. The 'Flush' checkbox is checked, and the 'Save' button is highlighted with a red arrow. The dialog box contains various settings for the bucket, including Memory Size, Access Control, Replicas, Disk I/O Optimization, and Auto-Compaction.





## Lab-3: Add 3 more nodes, page 24

and now click the newly enabled flush button

The screenshot shows the Couchbase Buckets management interface. At the top, there's a 'Create New Data Bucket' button. Below it, a table lists buckets: 'beer-sample' and 'default'. The 'default' bucket is selected, showing details like Item Count (510000), Ops/sec (0), and RAM/Quota Usage (103MB / 200MB). On the right side of the bucket details, there are buttons for 'Documents', 'Views', 'Compact', 'Edit', 'Flush', and 'Delete'. A red arrow points to the 'Flush' button. Below the bucket details, there are sections for 'Cache Size' and 'Storage Size' with progress bars and usage information.

From the Couchbase Web UI, on the server tab, on the pending rebalance sub-tab, click on “Rebalance” to begin the rebalancing process:

## Servers

**Fail Over Warning:** At least two servers are required to provide replication!

The screenshot shows the Couchbase Servers management interface. At the top, there's a 'Fail Over Warning' message. Below it, there are tabs for 'Active Servers' and 'Pending Rebalance'. The 'Pending Rebalance' tab is selected, and a red arrow points to the 'Rebalance' button. Below the tabs, there's a table listing servers with columns: Server Node Name, Group, Services, RAM Usage, Swap Usage, CPU Usage, Data/Disk Usage, and Items (Active / Replica). Three servers are listed, all with a status of 'Up' and 'Pending Add'.

You can monitor the rebalancing progress in the Web UI:

The screenshot shows the Couchbase Servers management interface during a rebalancing process. At the top, there's a 'Couchbase - 1 Node Cluster' header. Below it, there are tabs for 'Overview', 'Server Nodes', 'Data Buckets', 'Query', 'Indexes', 'XDCR', 'Security', and 'Log'. The 'Server Nodes' tab is selected, and a green box highlights the 'Rebalancing 4 nodes' progress bar. Below the tabs, there's a 'Settings' section with a 'Fail Over Warning' message. Below that, there are tabs for 'Active Servers' and 'Pending Rebalance'. The 'Pending Rebalance' tab is selected, and a red arrow points to the 'Stop Rebalance' button. Below the tabs, there's a table listing servers with columns: Server Node Name, Group, Services, RAM Usage, Swap Usage, CPU Usage, Data/Disk Usage, and Items (Active / Replica). Four servers are listed, all with a status of 'Up' and 'Pending Add'. The progress bar for each server shows the percentage of completion: 0.762 % Complete, 1.66 % Complete, 1.27 % Complete, and 0.879 % Complete.

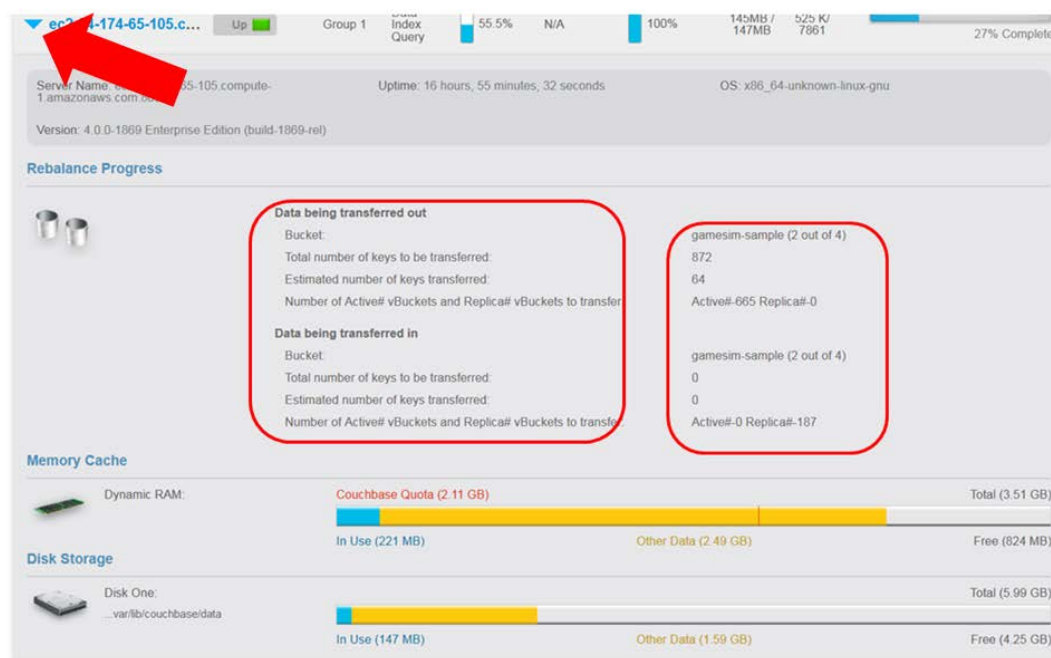




## Lab-3: Add 3 more nodes, page 25

During this process, you may notice some servers (especially the 1<sup>st</sup> VM in your lab with all the data on it) at an elevated CPU usage.

Try clicking the drop-down arrow on the 1<sup>st</sup> of the nodes to see detailed statistics about it:



Notice that there are two main sections here, Data being transferred out and Data being transferred in. In the sample screenshot above, it is the gamesim-sample bucket that is being transferred out from the 1<sup>st</sup> node to the other nodes. There is no data being transferred into the 1<sup>st</sup> node. After the gamesim-sample bucket is finished rebalancing, the nodes will work on rebalancing the beer-sample bucket and then the default bucket and then the travel-sample bucket.

**The entire rebalancing process should take about 4-6 minutes.**

While the rebalance is going on, you can also get rebalance statistics using the cbstats tool.

**Warning: The following commands have to be run when the beer-sample bucket is being rebalanced only! Please verify this from the above web UI!**

Wait about 3 – 4 minutes for the beer-sample database to start getting rebalanced. You can verify that the rebalance operation is working on the beer-sample bucket by watching the “Data being transferred out” section under the details of the 1<sup>st</sup> node:

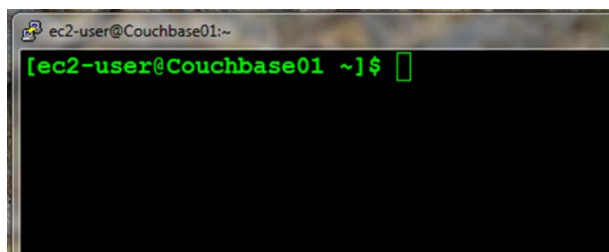


Lab-3: Add 3 more nodes, page 26

Data being transferred out	
Bucket:	beer-sample (2 out of 3)
Total number of keys to be transferred:	12470
Estimated number of keys transferred:	727
Number of Active# vBuckets and Replica# vBuckets to transfer:	Active#-727 Replica#-0

Quickly run through the steps on the next 5 - 6 pages WHILE the beer-sample bucket is being rebalanced (which takes about 2 minutes)! After the beer-sample bucket is finished rebalancing, the output of the following commands will not be the same.

Go to the 1<sup>st</sup> node's(Couchbase01) PuTTY shell (dark blue window):



```
[ec2-user@Couchbase01 ~]$ cd ~
```

The next command will return a list of DCP backfill processes and whether they are still running (true) or done (false).

The DCP protocol is an internal part of the Couchbase Server system and is used in a number of different areas to exchange data throughout the system. DCP provides a stream of data of the changes that are occurring within the system.

DCP is used during replication, to copy data between vBuckets used for replicas. It is also used during the rebalance procedure to move data between vBuckets and redistribute the information across the system.

Run the following command using the public hostname of the 1<sup>st</sup> node:

```
[ec2-user@Couchbase01 ~]$ cbstats ec2-54-85-43-x.compute-1.amazonaws.com:11210 -b beer-sample dcp | grep backfill
ep_dcp_max_running_backfills:
102
ep_dcp_num_running_backfills:
0
ep_dcp_queue_backfillremaining:
0
```



## Lab-3: Add 3 more nodes, page 27

```

eq_dcpq:mapreduce_view: beer-sample _design/beer (prod/main):backfill_buffer_bytes_read:
0
eq_dcpq:mapreduce_view: beer-sample _design/beer (prod/main):backfill_buffer_full:
false
eq_dcpq:mapreduce_view: beer-sample _design/beer (prod/main):backfill_buffer_max_bytes:
20971832
eq_dcpq:mapreduce_view: beer-sample _design/beer (prod/main):backfill_num_active:
0
eq_dcpq:mapreduce_view: beer-sample _design/beer (prod/main):backfill_num_pending:
0
eq_dcpq:mapreduce_view: beer-sample _design/beer (prod/main):backfill_num_snoozing:
0
eq_dcpq:replication:ns_1@ec2-54-174-65-105.compute-1.amazonaws.com->ns_1@ec2-54-173-46-
5.compute-1.amazonaws.com:beer-sample backfill_buffer_bytes_read: 0
eq_dcpq:replication:ns_1@ec2-54-174-65-105.compute-1.amazonaws.com->ns_1@ec2-54-173-46-
5.compute-1.amazonaws.com:beer-sample backfill_buffer_full: false
eq_dcpq:replication:ns_1@ec2-54-174-65-105.compute-1.amazonaws.com->ns_1@ec2-54-173-46-
5.compute-1.amazonaws.com:beer-sample backfill_buffer_max_bytes: 20971832
eq_dcpq:replication:ns_1@ec2-54-174-65-105.compute-1.amazonaws.com->ns_1@ec2-54-173-46-
5.compute-1.amazonaws.com:beer-sample backfill_num_active: 0
eq_dcpq:replication:ns_1@ec2-54-174-65-105.compute-1.amazonaws.com->ns_1@ec2-54-173-46-
5.compute-1.amazonaws.com:beer-sample backfill_num_pending: 0
eq_dcpq:replication:ns_1@ec2-54-174-65-105.compute-1.amazonaws.com->ns_1@ec2-54-173-46-
5.compute-1.amazonaws.com:beer-sample backfill_num_snoozing: 0
eq_dcpq:replication:ns_1@ec2-54-174-65-105.compute-1.amazonaws.com->ns_1@ec2-54-174-67-
95.compute-1.amazonaws.com:beer-sample:backfill_buffer_bytes_read: 0
eq_dcpq:replication:ns_1@ec2-54-174-65-105.compute-1.amazonaws.com->ns_1@ec2-54-174-67-
95.compute-1.amazonaws.com:beer-sample:backfill_buffer_full: false
eq_dcpq:replication:ns_1@ec2-54-174-65-105.compute-1.amazonaws.com->ns_1@ec2-54-174-67-
95.compute-1.amazonaws.com:beer-sample:backfill_buffer_max_bytes: 20971832
eq_dcpq:replication:ns_1@ec2-54-174-65-105.compute-1.amazonaws.com->ns_1@ec2-54-174-67-
95.compute-1.amazonaws.com:beer-sample:backfill_num_active: 0
eq_dcpq:replication:ns_1@ec2-54-174-65-105.compute-1.amazonaws.com->ns_1@ec2-54-174-67-
95.compute-1.amazonaws.com:beer-sample:backfill_num_pending: 0
eq_dcpq:replication:ns_1@ec2-54-174-65-105.compute-1.amazonaws.com->ns_1@ec2-54-174-67-
95.compute-1.amazonaws.com:beer-sample:backfill_num_snoozing: 0

```

In the above command output, notice the backfill for dcp queues

While the rebalance is occurring on the **beer-bucket** (overall rebalance of all the buckets should be at about 50% completed across all the nodes by now), **click on the 1<sup>st</sup> VM's Server Name under "Server Nodes" to see detailed statistics about the 1<sup>st</sup> node while the rebalance is going on:**

Servers

Active Servers Pending Rebalance

Stop Rebalance Rebalance Server Groups

Server Node Name	Group	Services	RAM Usage	Swap Usage	CPU Usage	Data/Disk Usage	Items (Active / Replica)
<b>ec2-54-174-42-8.co...</b>	Group 1	Data	13.1%	N/A	63.1%	16MB / 10MB	7626 / 7937

25% Complete

Server Name: ec2-54-174-42-8.compute-1.amazonaws.com:9091

Uptime: 45 minutes, 42 seconds

OS: x86\_64-unknown-linux-gnu

Version: 4.0.0-1060 Enterprise Edition (build-1060-rc0)

Rebalance Progress

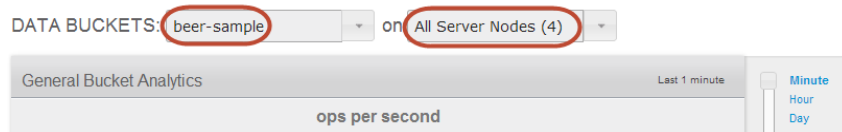
Data being transferred in

Bucket:	travel-sample (1 out of 4)
Total number of keys to be transferred:	15863
Estimated number of keys transferred:	15863
Number of Active vBuckets and Replica vBuckets to transfer:	Active# 0 Replica# 0



## Lab-3: Add 3 more nodes, page 28

Under the Data Buckets dropdown, **switch to the beer-sample bucket and change the second dropdown to All Server Nodes:**



You should see lots of activity under the Server Resources graphs:



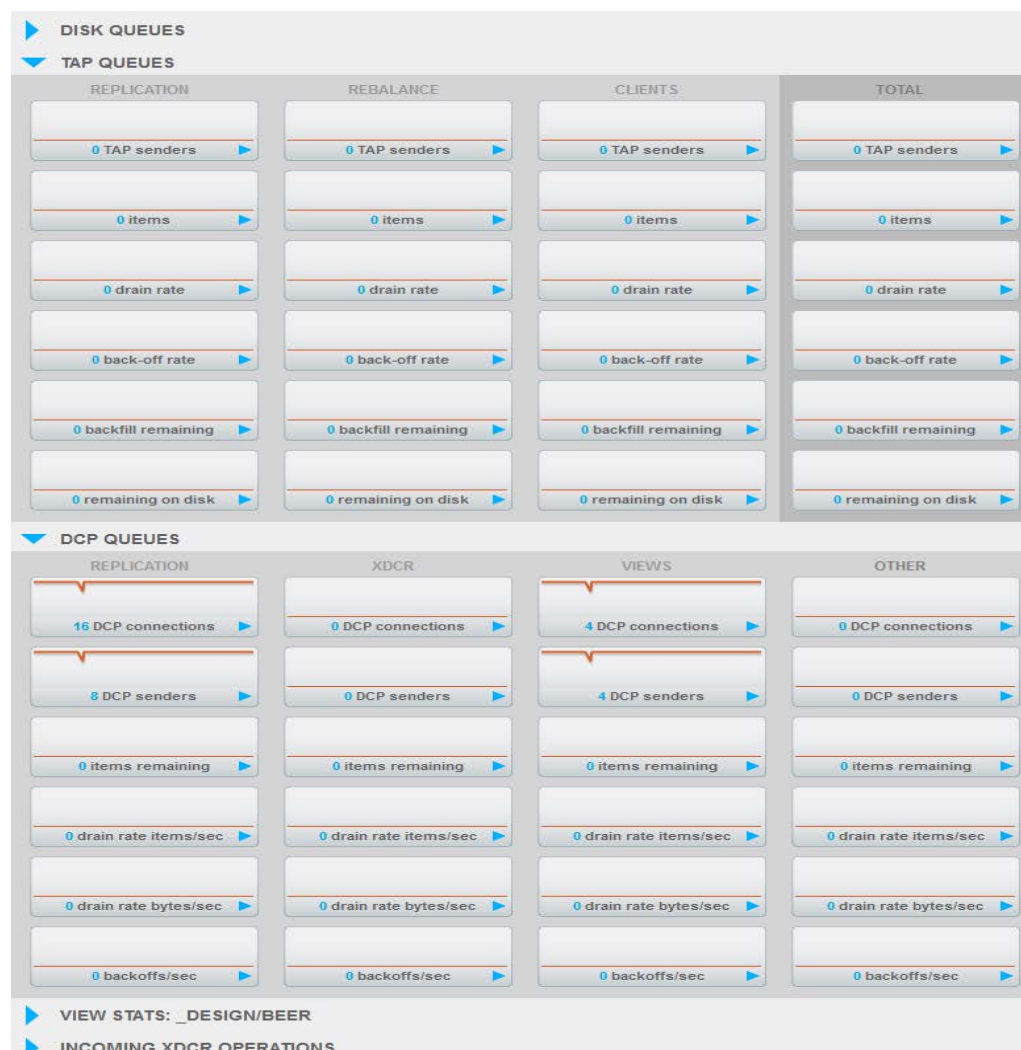
Scroll down and expand vBucket resources:



Scroll down to also look at the DCP queues(4.5.0):



## Lab-3: Add 3 more nodes, page 29



Note, in the above screenshot, in the first column, notice that there are 8 DCP senders, 2 on each server. You can also see that the drain rate is about 15 items per second. You can hover the mouse over any of the tiny graphs to understand more about what the graph is portraying:

**In Couchbase 4.5.0 (3.0 and above) the intracluster communication is now being handled by DCP and you can observe DCP related statistics.**

**Note:** During an upgrade from 2.5.2 to 4.5.0 you may observe statistics continuing to use TAP or if using a dedicated custom client application, once upgrade is complete and all nodes have DCP then further cluster communications will use DCP.



## Lab-3: Add 3 more nodes, page 30



More details about what these graphs and charts mean will be covered in the performance lab later on in the class.

**Note that by default the beer-sample bucket has 1 replica configured for it, so there will be a duplicate copy of every key/value created. You can verify this in the GUI by clicking on “Data Buckets” in the top menu and expanding beer-sample:**

The screenshot shows the Couchbase GUI for a 1 Node Cluster. The 'Data Buckets' tab is selected. The 'beer-sample' bucket is expanded, showing the following details:

- Bucket Name:** beer-sample
- Data Nodes:** 4
- Item Count:** 7303
- Ops/sec:** 0
- Disk Fetches/sec:** 0
- RAM/Quota Usage:** 102MB / 400MB
- Data/Disk Usage:** 27.8MB / 45.5MB
- Access Control:** Authentication
- Cache Metadata:** Value Ejection
- Replicas:** 1 replica copy
- Compaction:** Not active
- Cache Size:** Dynamic RAM Quota: 100MB
- Storage Size:** Persistence Enabled: Yes, Disk Usage: 45.5MB, Data Usage: 27.8MB

Below the expanded bucket, a table lists other buckets:

Bucket Name	Data Nodes	Item Count	Ops/sec	Disk Fetches/sec	RAM/Quota Usage	Data/Disk Usage	Documents	Views
default	4	0	0	0	50.6MB / 800MB	9.61MB / 9.68MB	Documents	Views
gamesim-sample	4	586	0	0	93.9MB / 400MB	19.6MB / 30.8MB	Documents	Views
travel-sample	4	31569	0	0	167MB / 400MB	86.1MB / 129MB	Documents	Views





## Lab-3: Add 3 more nodes, page 31

While the beer-sample bucket is being rebalanced, from the 1<sup>st</sup> (dark blue) node, run the following command again with the public hostname of the 1<sup>st</sup> node to see # of active vBuckets on this node reduce from the original number of 1024 to something lower (depending on how far along the rebalance operation is):

```
[ec2-user@Couchbase01 ~]$ /opt/couchbase/bin/cbstats ec2-54-174-65-105.compute-1.amazonaws.com:11210 all -b beer-sample | egrep -i "active_num|replica_num"
vb_active_num: 399
vb_active_num_non_resident: 0
vb_replica_num: 157
vb_replica_num_non_resident: 0
```

Once the rebalance is complete for the beer-sample bucket, you should see the same command return results showing 256 active vBuckets and 256 replica vBuckets on the 1<sup>st</sup> node:

```
[ec2-user@ Couchbase01 ~]$ /opt/couchbase/bin/cbstats ec2-54-174-65-105.compute-1.amazonaws.com:11210 all -b beer-sample | egrep -i "active_num|replica_num"
vb_active_num: 256
vb_active_num_non_resident: 0
vb_replica_num: 256
vb_replica_num_non_resident: 0
```

Rebalance for the default bucket might still be going on. After the default bucket is rebalanced, you will see a total of 256 active vBuckets on the 1<sup>st</sup> node, but zero replica vBuckets (since there are no replicas configured for the default bucket):

```
[ec2-user@ Couchbase01 ~]$ /opt/couchbase/bin/cbstats ec2-54-174-65-105.compute-1.amazonaws.com:11210 all -b default | egrep -i "active_num|replica_num"
vb_active_num: 256
vb_active_num_non_resident: 0
vb_replica_num: 0
vb_replica_num_non_resident: 0
```

When the entire rebalance operation across 4 nodes and 3 buckets is complete, the Server Nodes screen will refresh as follows:

The screenshot shows the Couchbase Server Nodes screen with the following data:

Server Node Name	Group	Services	RAM Usage	Swap Usage	CPU Usage	Data/Disk Usage	Items (Active / Replicas)	Fail Over	Remove
ec2-54-152-187-112...	Group 1	Data, Full Text Index, Query	62.5%	N/A	14.5%	39.7MB / 50.1MB	9811 / 9870	Fail Over	Remove
ec2-54-175-147-154...	Group 1	Data	19.8%	N/A	3.03%	34.9MB / 52MB	9924 / 9953	Fail Over	Remove
ec2-54-208-111-134...	Group 2	Data	20.9%	N/A	2.52%	34.9MB / 52.1MB	9943 / 9959	Fail Over	Remove
ec2-54-209-38-55.co...	Group 2	Data	21%	N/A	13.1%	33.6MB / 50.7MB	9980 / 9999	Fail Over	Remove



## Lab-3: Add 3 more nodes, page 32

Full details about the cbstats command may be found here:

<http://docs.couchbase.com/admin/admin/CLI/cbstats-intro.html>

For more information on how the Rebalance technology works in Couchbase, refer to the following 2 blog posts:

<http://blog.couchbase.com/rebalancing-couchbase-part-i>

<http://blog.couchbase.com/rebalancing-couchbase-part-ii>

In general, there should be no impact to a running application's availability while the rebalance operation is running.

**This command in the lab will count the number of files in the beer-sample bucket's data directory in linux. Run this command from the 1<sup>st</sup> node (dark blue):**

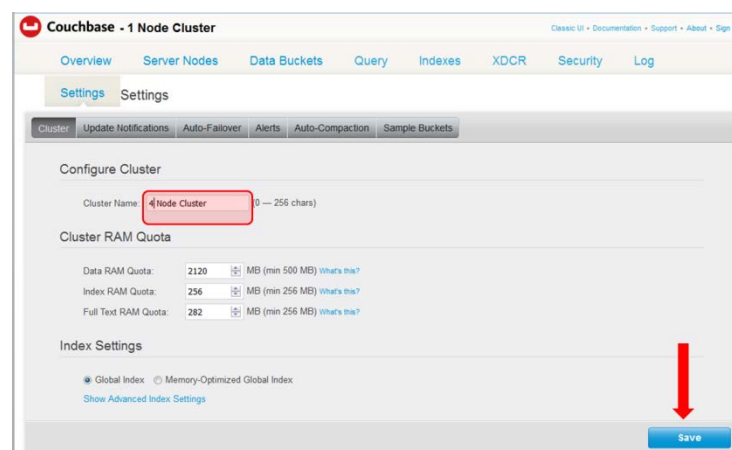
```
[ec2-user@Couchbase01 ~]$ sudo ls -al  
/opt/couchbase/var/lib/couchbase/data/beer-sample | wc -l  
518
```

The command output shows that there are 518 files in this directory. Remember that each Bucket has a total of 1024 vBuckets/partitions. So across a 4-node cluster, each node will hold about 256 vBuckets. So, on the 1<sup>st</sup> node, for the beer-sample bucket there will be 256 files for the active vBuckets, 256 files for the replica vBucket and a few extra metadata files totaling approximately **518**.

**Since the default bucket has zero replicas configured, it will have closer to 256 files on the 1<sup>st</sup> node:**

```
[ec2-user@ Couchbase01 ~]$ sudo ls -al  
/opt/couchbase/var/lib/couchbase/data/default | wc -l  
262
```

**Now go to the setting tab and change the name of the cluster to "4 Node Cluster"**







## Lab-3: Add 3 more nodes, page 33

Click “save”

## Adding index service node and query service nodes to the cluster

Click on the Server tab and click the Add server button

**Couchbase - 4 Node Cluster**

Classic UI • Documentation • Support • About • Sign Out

Overview **Server Nodes** Data Buckets Query Indexes XDCR Security Log

Settings Servers

Active Servers Pending Rebalance **Add Server** Server Groups

Server Node Name	Group	Services	RAM Usage	Swap Usage	CPU Usage	Data/Disk Usage	Items (Active / Replica)	
<a href="#">ec2-54-152-187-112...</a> Up	Group 1	Data Full Text Index Query	63%	N/A	10.1%	39.7MB / 60.1MB	9811 / 9870	Fail Over Remove
<a href="#">ec2-54-175-147-194...</a> Up	Group 1	Data	19.6%	N/A	2.01%	34.9MB / 52MB	9924 / 9853	Fail Over Remove
<a href="#">ec2-54-208-111-134...</a> Up	Group 2	Data	20%	N/A	9.59%	34.9MB / 52.1MB	9843 / 9869	Fail Over Remove
<a href="#">ec2-54-209-38-55.co...</a> Up	Group 2	Data	19.6%	N/A	1.51%	33.6MB / 50.7MB	9880 / 9866	Fail Over Remove

Fill in the information you have been provided for your 5<sup>th</sup> node IP address.

Add the cluster password “couchbase”

Select the Index service, Query service and FullText with a checked box. Deselect data service.

And hit the “Add Server” button

**Add Server**

**Warning** – Adding a server to this cluster means all data on that server will be removed.

Server IP Address\*:  [What's this?](#)

Server Group:

Security [What's this?](#)

Username:

Password:

Services:

☐ Data ☒ Index ☒ Query ☒ Full Text

[What's this?](#)



## Lab-3: Add 3 more nodes, page 34

Repeat this process for your 6<sup>th</sup> node and select Index service , Query service and **FullText** for the service. Include group 2 on node 6.

You should now have to pending rebalance operations.  
Click the “Rebalance” button to add them to the cluster

Server Node Name	Group	Services	RAM Usage	Swap Usage	CPU Usage	Data/Disk Usage	Items (Active / Replica)	
ec2-54-152-187-112....	Group 1	Data Full Text Index Query	57.6%	N/A	7.65%	39.7MB / 60.1MB	9811 / 9870	Fail Over Remove
ec2-54-175-147-194....	Group 1	Data	20.1%	N/A	1.01%	34.9MB / 52MB	9924 / 9853	Fail Over Remove
ec2-54-208-111-134....	Group 2	Data	19.7%	N/A	3.5%	34.9MB / 52.1MB	9843 / 9869	Fail Over Remove
ec2-54-209-38-55.co...	Group 2	Data	20%	N/A	2.53%	33.6MB / 50.7MB	9880 / 9866	Fail Over Remove

In this case the nodes are added but since they are nodes that offer index and query service and Full Text service they are not involved in the vbucket migrations that characterize data service rebalancing.



## Lab-3: Add 3 more nodes, page 35

While the nodes are rebalancing.....

Go to the settings tab and rename the cluster “6 Node Cluster”

The screenshot shows the Couchbase management console for a 4 Node Cluster. The 'Settings' tab is selected, and the 'Cluster' sub-tab is active. The 'Configure Cluster' section shows the 'Cluster Name' as '6 Node Cluster'. Below this, the 'Cluster RAM Quota' section shows three quotas: Data RAM Quota (2120 MB), Index RAM Quota (256 MB), and Full Text RAM Quota (282 MB). The 'Index Settings' section shows the 'Global Index' option selected. A 'Save' button is located at the bottom right of the settings panel.

Click the “save” button

The screenshot shows the Couchbase management console for a 6 Node Cluster. The 'Settings' tab is selected, and the 'Cluster' sub-tab is active. The 'Configure Cluster' section shows the 'Cluster Name' as '6 Node Cluster'. Below this, the 'Cluster RAM Quota' section shows three quotas: Data RAM Quota (2120 MB), Index RAM Quota (256 MB), and Full Text RAM Quota (282 MB). The 'Index Settings' section shows the 'Global Index' option selected. A 'Save' button is located at the bottom right of the settings panel.



## Lab-3: Add 3 more nodes, page 36

If you click on the blue triangle next to the rebalance progress bar it will open to display all operations being rebalanced. In the below screen shot it is rebuilding indexes that were constructed in a test cluster.

The screenshot shows the Couchbase 6 Node Cluster management interface. The 'Servers' tab is active, displaying a table of server nodes. A dropdown menu is open for the first node, showing the progress of rebalancing 4 nodes and rebuilding indexes.

Server Node Name	Group	Services	RAM Usage	Swap Usage	CPU Usage	Disk Usage	Buttons
ec2-54-152-187-112....	Group 1	Data, Full Text, Index, Query	57.2%	N/A	8.9%	39.7MB / 58.5MB, 9811 / 9870	Fail Over, Remove
ec2-54-175-147-194....	Group 1	Data	21.1%	N/A	4.04%	34.9MB / 52MB, 9924 / 9853	Fail Over, Remove
ec2-54-208-111-134....	Group 2	Data	21.2%	N/A	2.01%	34.9MB / 52.1MB, 9843 / 9869	Fail Over, Remove
ec2-54-209-38-55.co...	Group 2	Data	20.9%	N/A	1.51%	33.6MB / 50.7MB, 9880 / 9866	Fail Over, Remove
ec2-54-210-234-108....	Group 1	Full Text, Index, Query	20.4%	N/A	2.01%	N/A, 0 / 0	Fail Over, Remove
ec2-54-210-235-65.c...	Group 1	Full Text, Index, Query	21.1%	N/A	2.01%	N/A, 0 / 0	Fail Over, Remove

Rebalancing 4 nodes

- Indexing beer-sample/\_design/beer
- Indexing beer-sample/\_design/ddl\_primary
- Indexing beer-sample/\_design/ddl\_Beer\_abv
- Indexing beer-sample/\_design/ddl\_Beer\_by\_phone
- Indexing beer-sample/\_design/ddl\_Beer\_by\_state
- Indexing beer-sample/\_design/ddl\_Beer\_style

This completes lab 3.