# CS300 Couchbase NoSQL Server Administration

# Lab 1 Exercise Manual


**Release: 4.5**

Revised: Sept 1st, 2016



# Lab #1: Installation and overview of Couchbase Server

**Objective:** This 1-hour lab will introduce you to Couchbase server, specifically its installation on one node and the web UI. Future labs will go much more in depth into the administrator and configuration components of Couchbase, so consider the coverage in this lab only your initial glance at Couchbase.

Please keep in mind that the major objective of these labs is not to build out a production-level cluster, but rather a prototype lab environment where various concepts and features can be explained and demonstrated. For example, in these labs we simply turn off the Linux firewall and don't follow the best practice to store the Couchbase data files and index files on separate devices/volumes. Also, note that the VMs in Amazon used for these labs are t2.medium with only 2 cores and 4GB memory. These VMs do not give a realistic representation of the performance you can get out of a Couchbase cluster. In other words, although these small VMs may show performance of 15,000 iops per second, in a larger Amazon VM, you will see 100,000 iops per second or even 200,000 iops per second in a physical data center.

Warning: Do not copy + paste commands from this lab into your PuTTY/Terminal session. Some commands, especially commands that span multiple lines. A multi-line command will break into 2 lines when you copy it as the PDF will insert a /n character after the first line. Instead, please type each command individually into the SSH session!

If you disregard this and insist on cutting and pasting please paste to a notepad or text file editor and then cut and paste again from there( this should strip out extraneous characters.

Please send any comments or corrections in this lab or future labs to Couchbase Learning Services at cls@couchbase.com

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

- Installation of Couchbase 4.5 EE on 1-node in Amazon Web Services (AWS)
- Explore the UI: Cluster overview, cluster summary, viewing buckets, viewing server nodes, viewing data buckets, logs
- Explore cmd line options
- How to start and stop Couchbase server
- look at Beer sample database
- Look at the Couchbase DB storage files in the Linux file system
- Initial glance at log files for Couchbase

# Couchbase Server Installation Introduction:

**Reference Documentation**

Below are some links to explore on your own time to learn more about Couchbase Server's installation and administration. The main, critical parts of these guides have been distilled into the abbreviated labs we will do in this class. However, for a deep dive into managing Couchbase, you must spend some time with these documents.

Here is a link to all of the available official Couchbase documentation in HTML format:
http://www.couchbase.com/documentation

The official Couchbase Server 4.X admin guide (note that you can choose topics from the blue dropdown in the top left that says "Couchbase Server"):
http://docs.couchbase.com/admin/admin/Whats-new-4.0.html

Couchbase's YouTube channel has many excellent videos from recent conferences and webinars:
https://www.youtube.com/channel/UCGUDXCRwJi-fuQp7sJyIZmg

About a dozen technical white papers on Couchbase can be found here (We recommend starting with "Couchbase Server Under the Hood: An Architectural Overview"):
http://www.couchbase.com/nosql-resources/nosql-whitepapers

Couchbase 101 – 105 webinar training series:
http://www.couchbase.com/nosql-resources/webinar

Couchbase presentations and slides from the Couchbase community:
http://www.couchbase.com/nosql-resources/presentations

You can follow the latest technical developments in Couchbase the official blog:
http://blog.couchbase.com

**Where to get help**

The Couchbase communities' website allows you to post technical questions:
http://www.couchbase.com/open-source

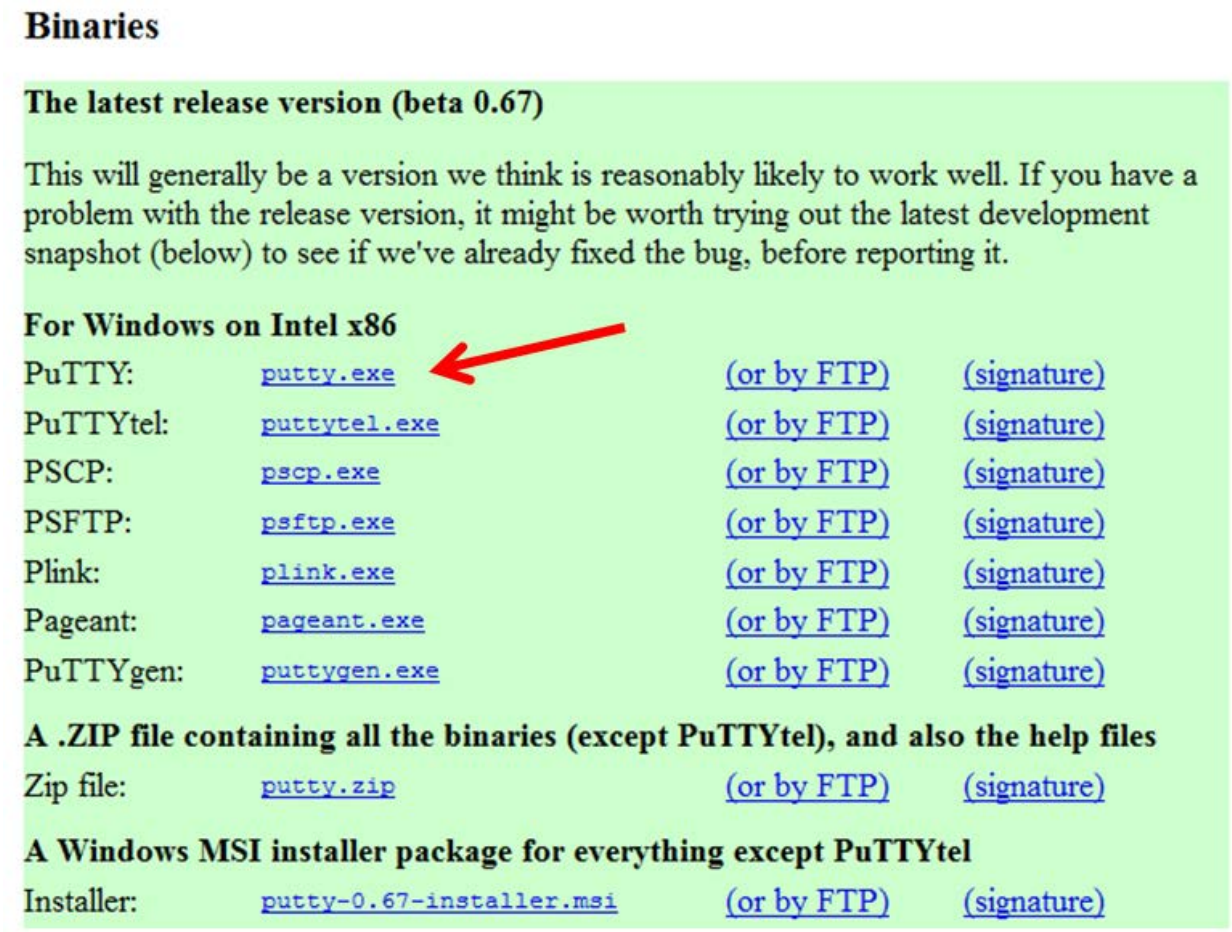
## Installing PuTTY & Connecting to the 1st VM (Windows Only):

If you are on a Mac, skip this step and go to the next bold, blue heading. You can use the built in Terminal or iTerm2 (a more feature-rich replacement for Terminal) to connect/SSH to the Amazon VM, so you don't need PuTTY (which is a Windows-only app).

If you're on Windows, it is highly recommended to install PuTTY, a free telnet/SSH client. With PuTTY, you can connect to the Amazon VM from a lightweight client in Windows and open multiple cmd-line sessions to the same VM.

Download PuTTY from:
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Look for the file named putty.exe under "Windows on Intel x86":



**Binaries**

**The latest release version (beta 0.67)**

This will generally be a version we think is reasonably likely to work well. If you have a problem with the release version, it might be worth trying out the latest development snapshot (below) to see if we've already fixed the bug, before reporting it.

**For Windows on Intel x86**

| | | | |
|---|---|---|---|
| PuTTY: | putty.exe | (or by FTP) | (signature) |
| PuTTYtel: | puttytel.exe | (or by FTP) | (signature) |
| PSCP: | pscp.exe | (or by FTP) | (signature) |
| PSFTP: | psftp.exe | (or by FTP) | (signature) |
| Plink: | plink.exe | (or by FTP) | (signature) |
| Pageant: | pageant.exe | (or by FTP) | (signature) |
| PuTTYgen: | puttygen.exe | (or by FTP) | (signature) |

**A .ZIP file containing all the binaries (except PuTTYtel), and also the help files**

| | | | |
|---|---|---|---|
| Zip file: | putty.zip | (or by FTP) | (signature) |

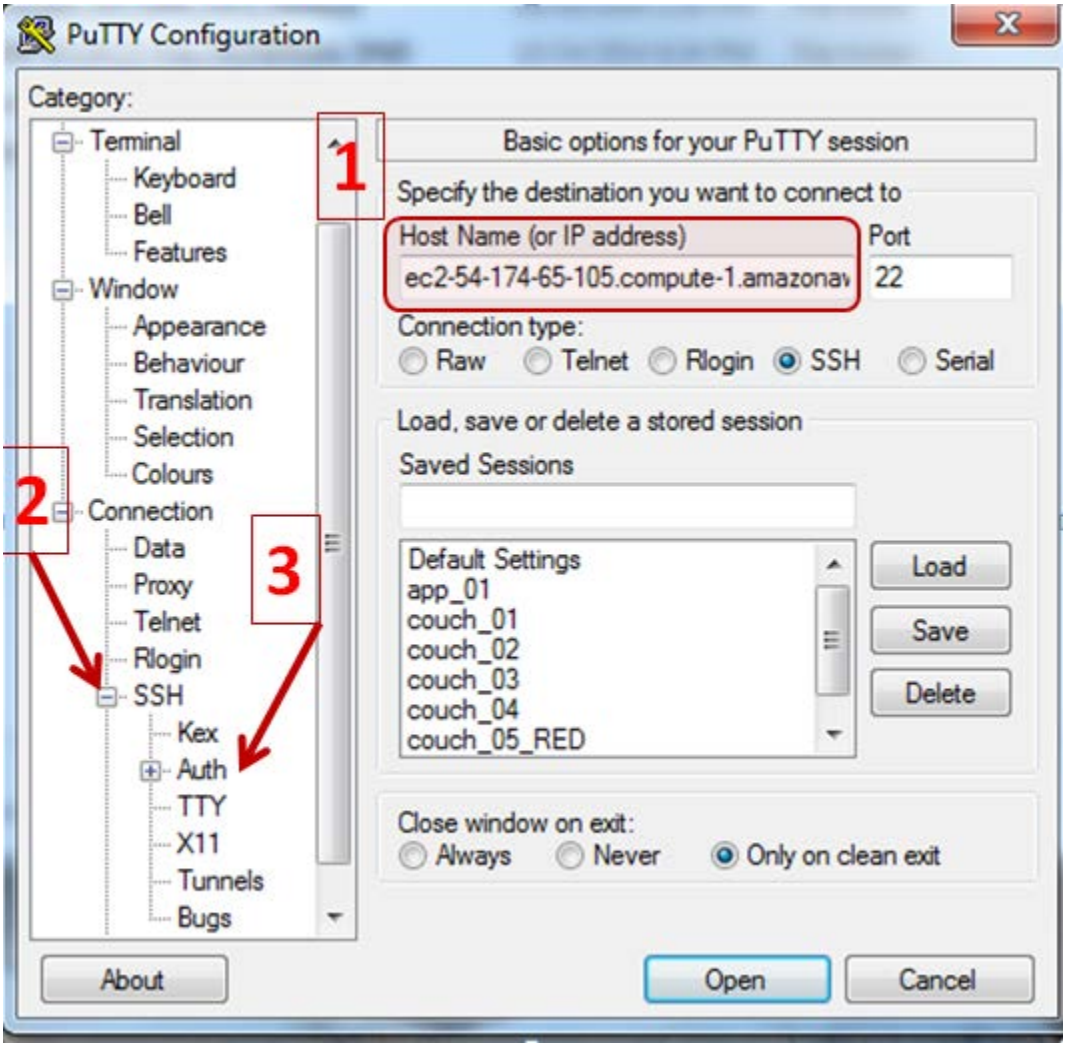
**A Windows MSI installer package for everything except PuTTYtel**

| | | | |
|---|---|---|---|
| Installer: | putty-0.67-installer.msi | (or by FTP) | (signature) |

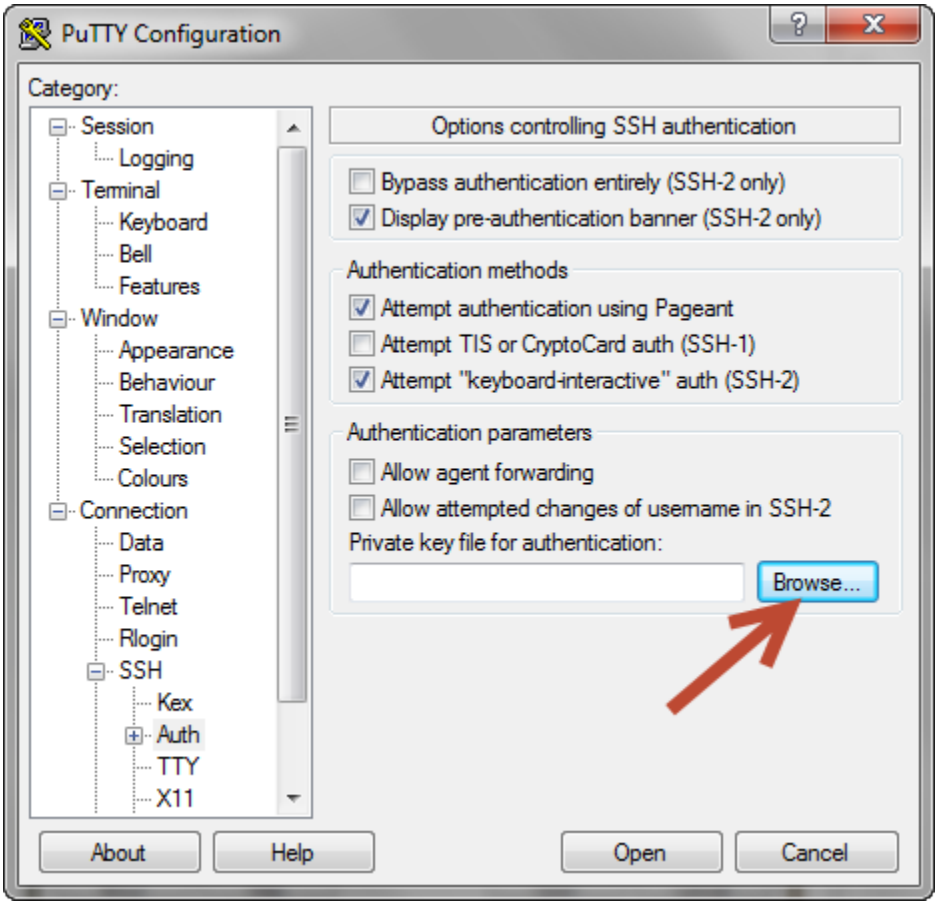
There is no installation for PuTTY. You can just run it from the downloaded `.exe` file.

After starting PuTTY, enter the public IP address of the first Amazon VM into PuTTY. You can get this IP from the `Cluster-IPs` spreadsheet that the instructor gave you along with this lab. The connection type will be SSH and the port will be 22.

**Type the public hostname** that the instructor gave you for your first Amazon VM into PuTTY and then **click on the + next to SSH** to expand its options and finally **select Auth**:

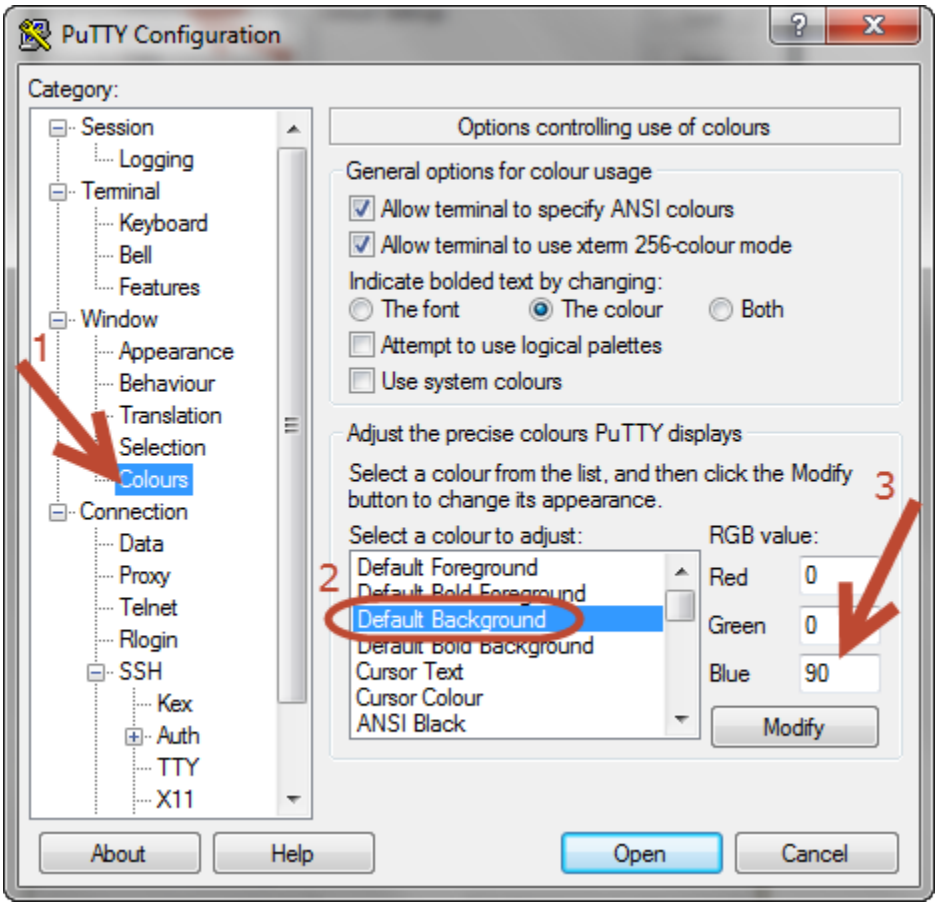


**Click Browse** to select the Private key file for authentication:

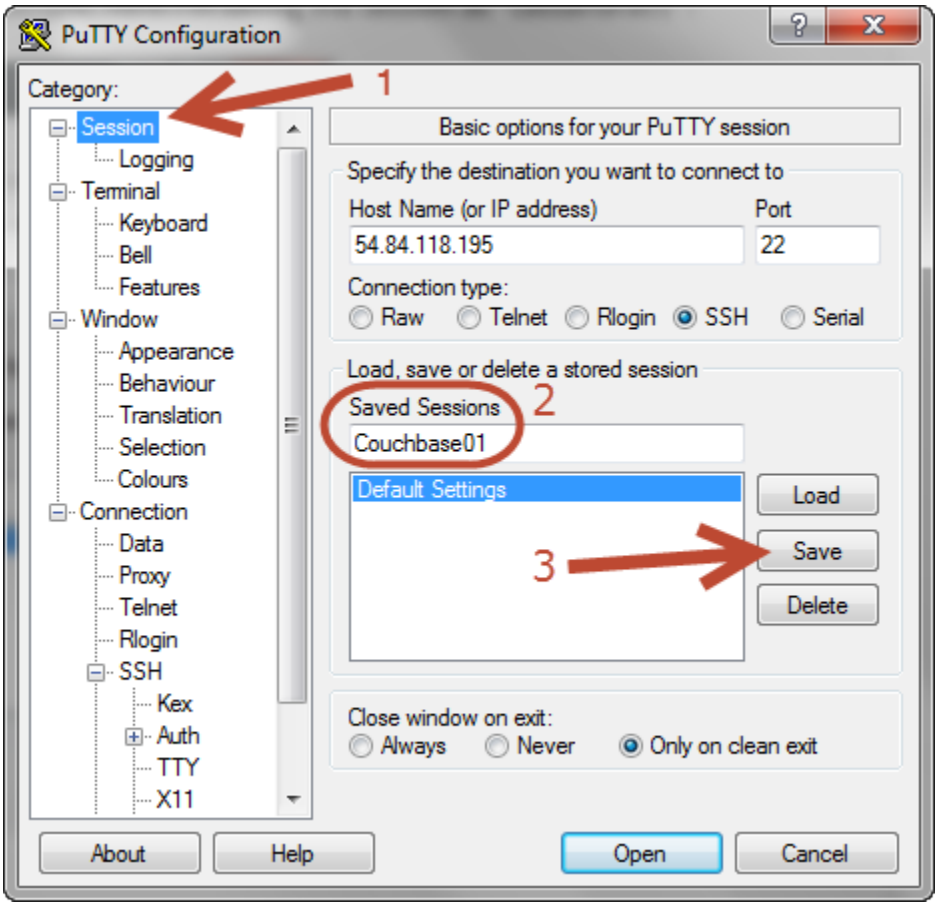



Choose the **"Amazon-Private-Key2.ppk"** file that the instructor provided you with.

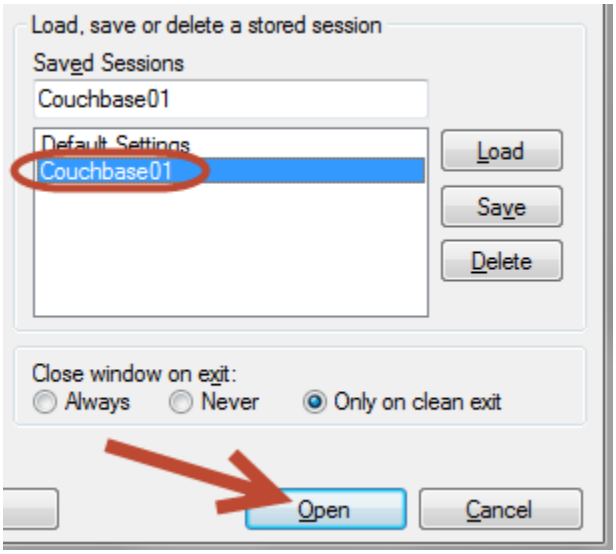
In the left pane, **click on Colors**, then under "Select a Colours to adjust" **choose Default Background** and alter the **Blue RGB value to 90**.



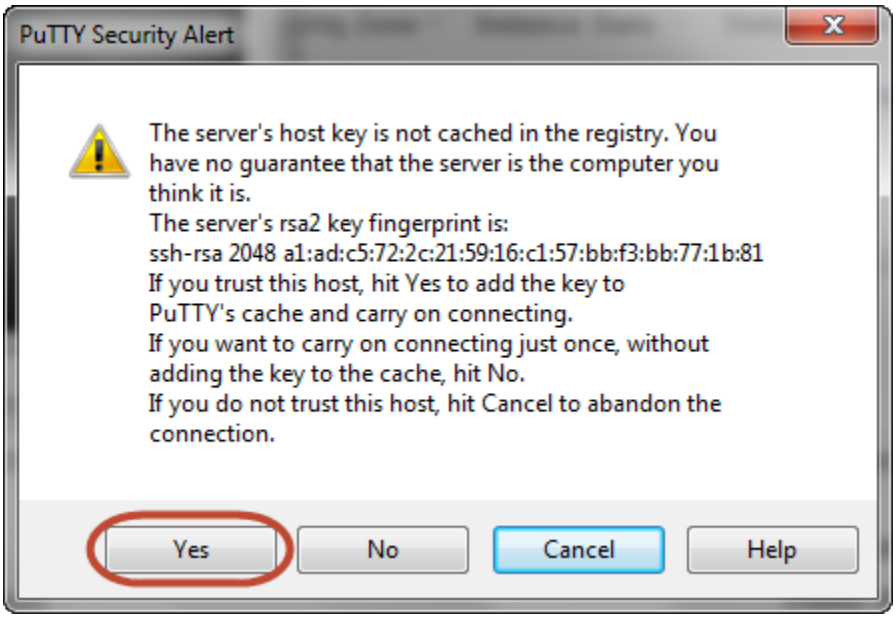
Next, **click on Session** and type to **save the session as "CouchbaseXX"**, where XX is the # of your node from the hostname. Then **click on Save**. For example, here the session is being saved as "Couchbase01":



Now highlight **Couchbase01** and click **Open** to connect to this VM:

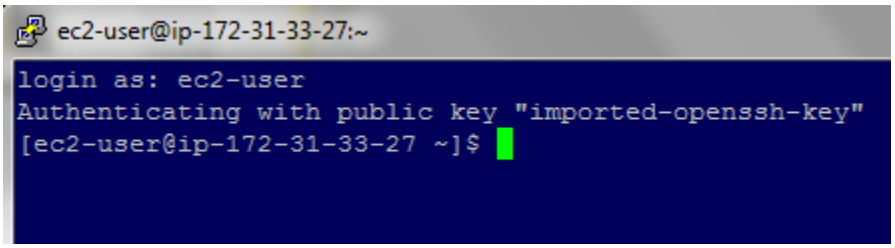



You will have to click **"Yes"** to a message about the server's rsa2 key before a successful connection.



The username for your login is:

Login as: **ec2-user**




## Connecting to the 1st VM via Terminal/iTerm2 (Mac only):

The general instructions to log in via Mac Terminal are:

Open up your terminal app of choice and type in the following…

**Change the permissions of the .pem key file like this:**
**chmod 400 Amazon-Private-Key2.pem**

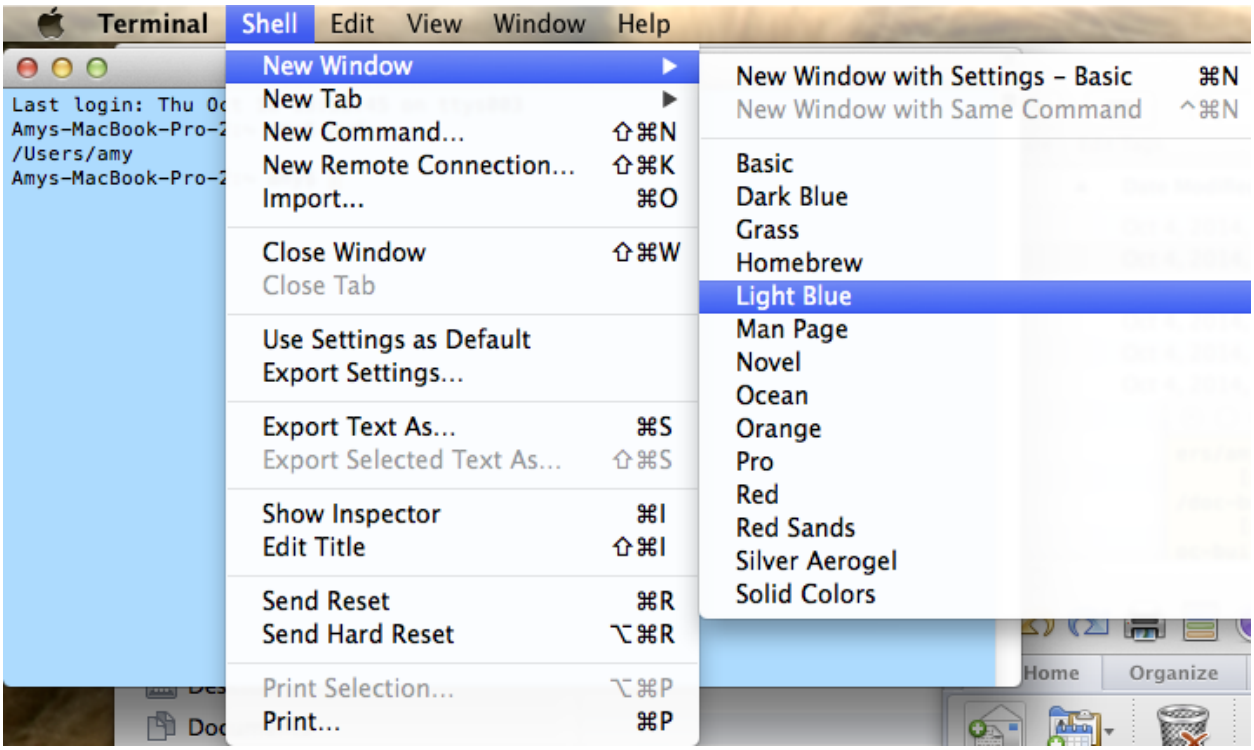
**SSH into the VM using this command:**
**ssh -i Amazon-Private-Key2.pem ec2-user@<public hostname of 1st VM>**

**Say Yes to this prompt:**

```
The authenticity of host 'ec2-198-51-100-x.compute-1.amazonaws.com
(10.254.142.33)'
can't be established.
RSA key fingerprint is
1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
Are you sure you want to continue connecting (yes/no)? yes
```

**Here are the official details on how to log in via Mac:**
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/

Setting screen colors can also be accomplished on a Mac.
Here is the screen shot of affecting color change on a Macbook.




Please ask the instructor for any further help needed with logging in from an Apple laptop!

## Exploring the 1st Amazon server:

Here are the specs for the Couchbase server VM that you just launched:

**Amazon AMI:**
Red Hat Enterprise Linux 7.2 (HVM) - ami-12663b7a (64-bit)
Root device type: ebs
Virtualization type: paravirtual

Amazon Instance Type: **t2.medium**
ECUs: **3**
vCPU: **2**
Memory: **4.0 GiB**
Storage: **10GB magnetic** *(Note, SSDs are available, but the labs will use magnetic storage)*
Network performance: **moderate**
CloudWatch Monitoring: **disabled**
Tenancy: **Shared tenancy** (multi-tenant hardware)
Cost: **$0.05 per hour**

Note that the above specs are not enough for a production-worthy Couchbase install! For production, you should have 4-6 CPU cores and at least 16 GB of RAM, but your VM's specs should be enough for a prototype lab environment.

We choose to run Red Hat Linux for these labs because RHEL is aimed at enterprise-level servers, which means that it is stable and handles heavy loads well. RHEL is also one of the supported OS's for Couchbase 4.X Enterprise Edition.

Here is a link to the supported OS platforms for Couchbase Server:
http://docs.couchbase.com/admin/admin/Install/install-platforms.html

Go to the PuTTY or Terminal window and…

**Check the hostname of your machine:**

```
[ec2-user@ip-172-31-46-176~]$ hostname
ip-172-31-19-30
```

*Note: this hostname is for inside Amazon resolution( with inside name server from amazon) all access for this class will use outside ec2-w-x-y-z-.amazon.com names.*

**Sudo to root and change the hostname to Couchbase01**

```
[ec2-user@ip-172-31-46-176 ~]$ sudo –i
[root@ip-172-31-46-176 ~]# hostname Couchbase01
[root@ip-172-31-46-176 ~]# hostname
Couchbase01
[root@ip-172-31-46-176 ~]#exit
Logout
```

**Now close the putty window and reopen a new one to verify that the hostname has been changed**

**First verify that this server has ~3.5GB of RAM and only 90-120 MB or so is currently being used. (*this could vary in your environment depending on how long the VM has been running for*):**

```
[ec2-user@Couchbase01 ~]$ free -mh
              total        used        free      shared  buff/cache   available
Mem:           3.5G         95M        3.1G         16M        251M        3.1G
Swap:            0B          0B          0B
```

**Verify that a ~10.0 GB data disk is showing up on the server:**

```
[ec2-user@Couchbase01~]$ sudo fdisk -l

Disk /dev/xvda: 10.7 GB, 10737418240 bytes, 20971520 sectors
```

```
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: gpt

#         Start          End    Size  Type            Name
 1         2048         4095      1M  BIOS boot parti
 2         4096     20971486     10G  Microsoft basic
```

**Check what type of file system is carved on the VM:**

```
[ec2-user@Couchbase01 ~]$ df -Th
```

```
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/xvda2     xfs        10G  927M  9.1G  10% /
devtmpfs       devtmpfs  1.9G     0  1.9G   0% /dev
tmpfs          tmpfs     1.8G     0  1.8G   0% /dev/shm
tmpfs          tmpfs     1.8G   17M  1.8G   1% /run
tmpfs          tmpfs     1.8G     0  1.8G   0% /sys/fs/cgroup
tmpfs          tmpfs     354M     0  354M   0% /run/user/1000
tmpfs          tmpfs     354M     0  354M   0% /run/user/0
```

Notice that the main file system is /dev/xvda2, which is of type xfs and size 10 GB with 927MB used (so 10% of the disk is full).

We will place both the Couchbase data files and the index files on this single disk. *However, in a production setup, it is recommended to configure 3 separate volumes on multiple disks, one for the Linux OS, one diskgroup per bucket for the data files and one diskgroup per index for the index files. For cost/time constraints, we will leave all 3 items on one volume in this lab.*

## Implement Best Practices for Couchbase:

### 1) Disable Swappiness

Swappiness levels tell the Linux virtual memory subsystem how much it should try and swap to disk. The problem is that the system will try to swap out items in memory even when there is plenty of RAM available to the system.

**Check what value your VM is set to by running:**

```
[ec2-user@Couchbase01 ~]$ cat /proc/sys/vm/swappiness
30
```

*The default setting of '30' is a bit aggressive. The value of 30 is a percentage; the higher the percentage, the higher the I/O cache and the faster that pages are swapped. You can gain performance by setting the swappiness value to 0. This tells the virtual memory subsystem of the OS to not swap items from RAM to disk unless it absolutely has to. A setting of 100 would have meant that programs will be swapped to disk almost immediately. If you have sized your nodes correctly, swapping should not be needed.*

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

```
[ec2-user@Couchbase01~]$ sudo -s
[root@Couchbase 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(DO NOT REBOOT!!) and exit root (*Note: all of these echo commands should be entered on ONE line, do not spread them across two lines in the CMD prompt!*):**

```
[root@Couchbase01 ec2-user]# echo '' >> /etc/sysctl.conf
[root@Couchbase01 ec2-user]# echo '#Set swappiness to 0 to avoid swapping' >> /etc/sysctl.conf
[root@Couchbase01 ec2-user]# echo 'vm.swappiness = 0' >> /etc/sysctl.conf
```

### 2) Turn off the Linux firewalld(may not be needed in amazon cloud)

```
#systemctl stop firewalld
Failed to issue method call: Unit firewalld.service not loaded.
```

*Note, you should not turn off the firewall like this in a production Couchbase cluster! In that case you should instead refer to this URL to see which ports need to be selectively opened up for Couchbase:*

### x) 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
```

For instructions on disabling THP permanently visit:
http://blog.couchbase.com/often-overlooked-linux-os-tweaks

Here is additional information about Transparent Huge Pages from Red Hat:
https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Performance_Tuning_Guide/s-memory-transhuge.html

## Install Couchbase:

```
# yum install wget
Loaded plugins: amazon-id, rhui-lb
rhui-REGION-client-config-server-7                                | 2.9 kB  00
rhui-REGION-rhel-server-releases                                  | 3.7 kB  00
rhui-REGION-rhel-server-rh-common                                 | 1.9 kB  00
(1/4): rhui-REGION-client-config-server-7/x86_64/primary_db       | 5.0 kB  00
(2/4): rhui-REGION-rhel-server-rh-common/7Server/x86_64/primary   |  30 kB  00
(3/4): rhui-REGION-rhel-server-rh-common/7Server/x86_64/updateinfo |  11 kB  00
(4/4): rhui-REGION-rhel-server-releases/7Server/x86_64/primary_db |  13 MB  00
(1/2): rhui-REGION-rhel-server-releases/7Server/x86_64/group_gz   | 133 kB  00
(2/2): rhui-REGION-rhel-server-releases/7Server/x86_64/updateinfo | 590 kB  00
rhui-REGION-rhel-server-rh-common
Resolving Dependencies
--> Running transaction check
---> Package wget.x86_64 0:1.14-10.el7_0.1 will be installed
```

```
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package              Arch                   Version                    Repository
================================================================================
Installing:
 wget                 x86_64                 1.14-10.el7_0.1            rhui-REGION-rhel-
server-releases

Transaction Summary
================================================================================
Install  1 Package

Total download size: 546 k
Installed size: 2.0 M
Is this ok [y/d/N]: y
Downloading packages:
wget-1.14-10.el7_0.1.x86_64.rpm
| 546 kB  00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : wget-1.14-10.el7_0.1.x86_64
  Verifying  : wget-1.14-10.el7_0.1.x86_64

Installed:
  wget.x86_64 0:1.14-10.el7_0.1

Complete!
```

**# exit**

**Download Couchbase 4.5 EE (*do not copy + paste this command!*):**

**[ec2-user@Couchbase01~]$ wget http://packages.couchbase.com/releases/4.5.0/couchbase-server-enterprise-4.5.0-centos7.x86_64.rpm**

```
--2016-04-26 13:31:47--  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)... 54.231.16.240
Connecting to packages.couchbase.com (packages.couchbase.com)|54.231.16.240|: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 58.0MB/s   in 1.7s

2016-04-26 13:31:48 (58.0 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@Couchbase01 ~]$ 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://Couchbase01:8091/ 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..
```

Congratulations! If you see the above message, you've successfully installed Couchbase. In the next section, we'll configure and start the cluster.

Note: in Couchbase 4.1 autostart was dependent on RH7.1 symbolic links that are not present in RH7.2 Please use the following to start couchbase. (this is remedied in Couchbase 4.5)

**/opt/couchbase/etc/couchbase_init.d start**
Starting couchbase-server

**/opt/couchbase/etc/couchbase_init.d stop**
Stopping couchbase-server

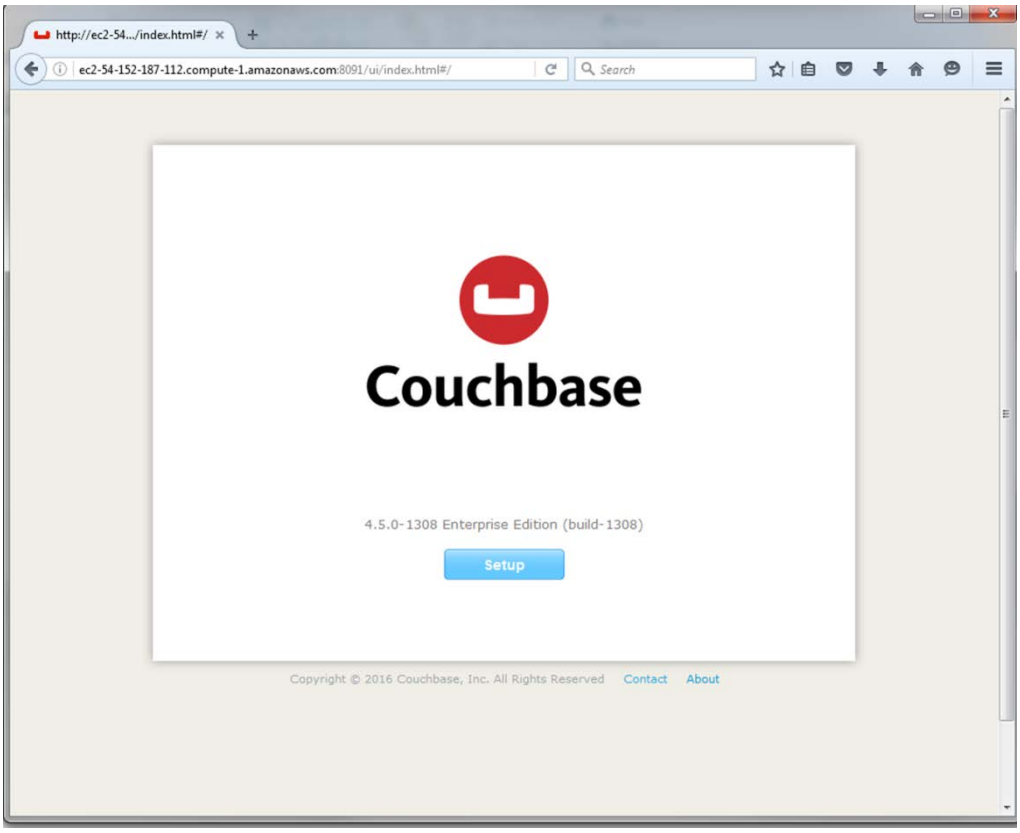
**/opt/couchbase/etc/couchbase_init.d status**
Obtaining system status

## Configure a 1-node Couchbase cluster:

Note that the URL provided in the output from the install will not work directly. You have to replace the AWS hostname/Couchbase01 from the installation output above (for example: ip-172-31-33-xx) with the public hostname of the VM from the excel spreadsheet (like: ec2-54-88-123-x.compute-1.amazonaws.com).

**Open a Chrome or Firefox browser and go to the following URL:**

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




**Click on the SETUP button in the bottom area to continue.**
**Some of the settings on the "CONFIGURE SERVER" page will need to be altered. Specifically, the items in red need to be changed:**

**Choose:** Start a new cluster

**Services select:** Data, Index, Query & Full Text

**Per Server DATA RAM Quota: 2120** MB *Accept the default calculation if your memory value is different than shown*

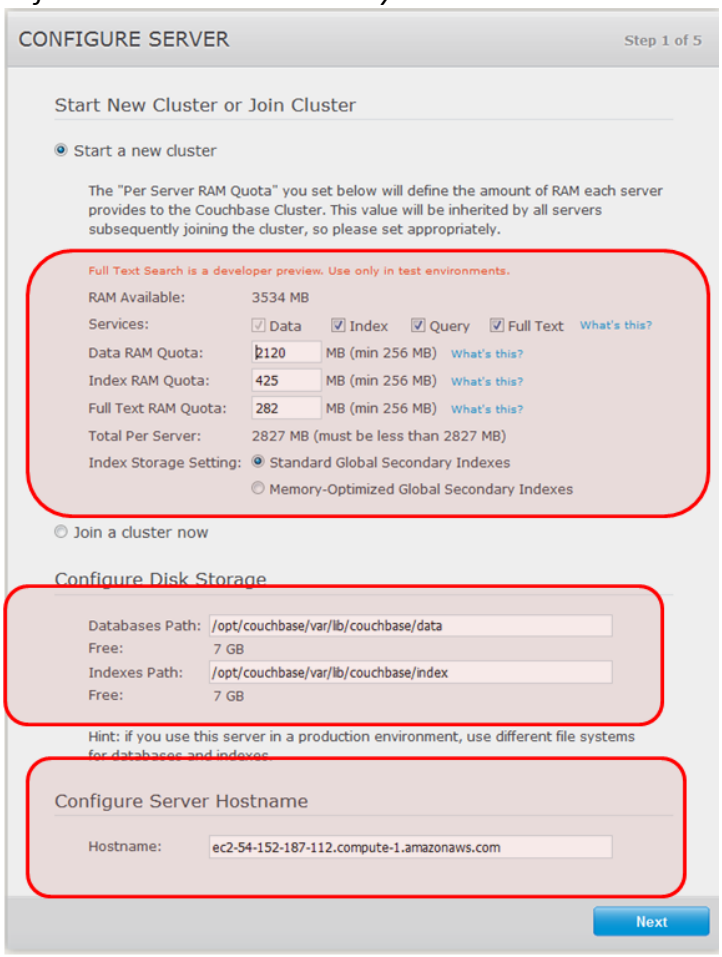
**Per Server Index RAM Quota: 425** MB *(Min ram value shown to right)*
*Accept the default calculation if your memory value is different than shown or you are configuring a dedicated INDEX service with more RAM available.*

***Full text RAM Quota:282 MB*** *Accept the default calculation if your memory value is different than shown or you are configuring a dedicated INDEX service with more RAM available*
*Databases Path: /opt/couchbase/var/lib/couchbase/data*

Index Storage Setting:
Leave the Standard Global Secondary Indexes radio button selected

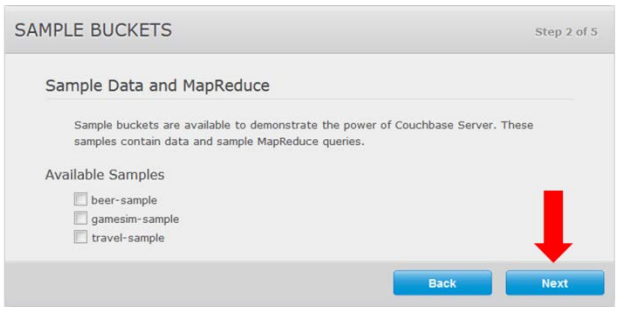
*Indices Path: /opt/couchbase/var/lib/couchbase/index*

*Hostname: <Public hostname of VM, retrieve this from the Cluster-IPs spreadsheet>*
*Or from the browser URL entry line. i.e. <ec2-54-174-65-105.compute-1.amazonaws.com>*

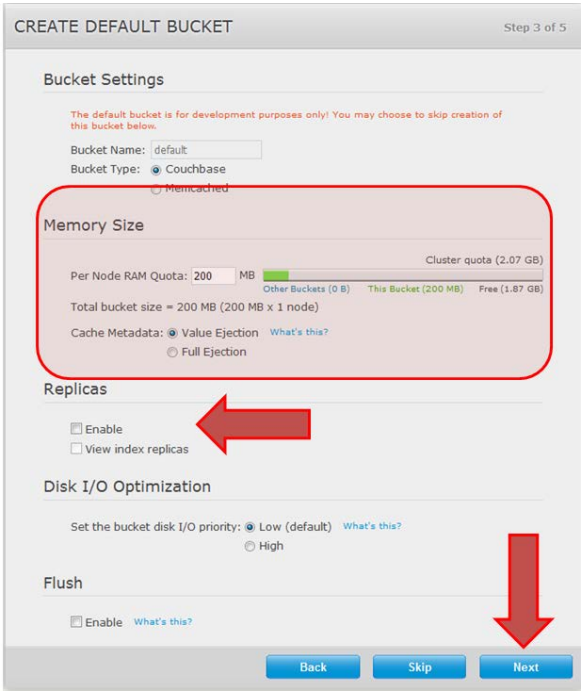



**Click on Next to continue:**

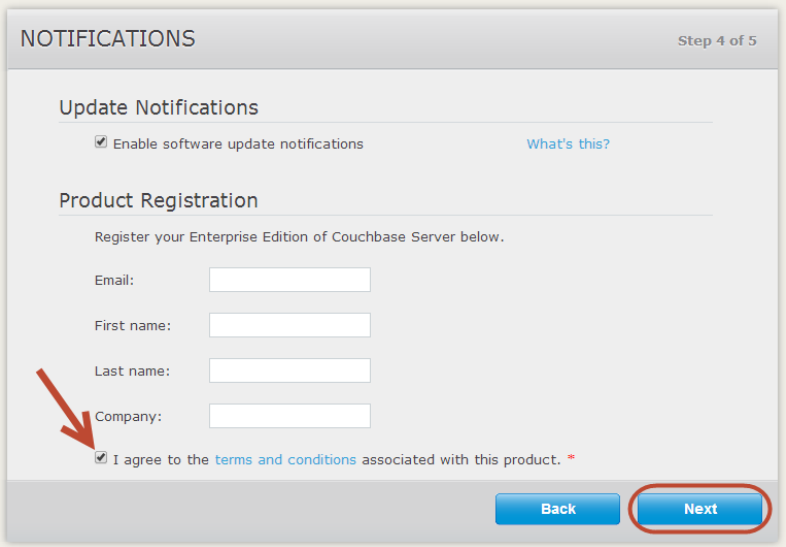
**Leave the sample buckets to add later and click on Next to continue:**



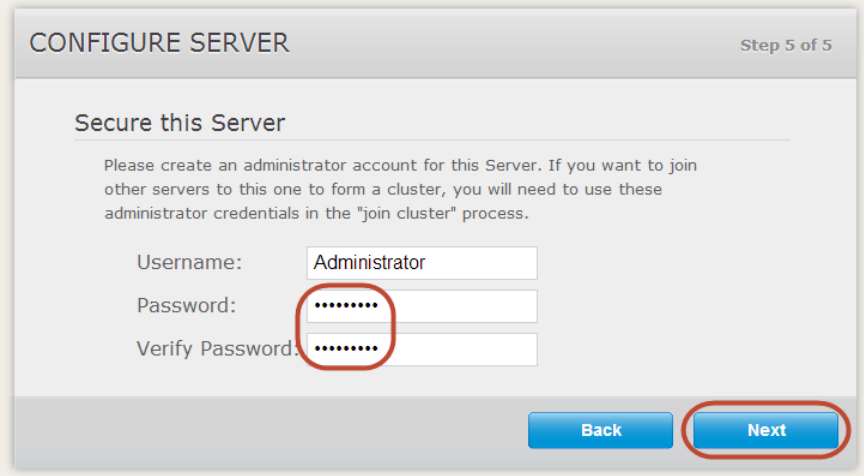

Next screen configures the default bucket for Couchbase. 3 changes to make on Step 3:
- **Reduce the Per Node RAM Quota to 200 MB**
- **Cache Metadata to value ejection (default )**
- **Uncheck 'Enable' under Replicas to disable creating an extra copy of the data. Since we are only configuring a 1-node cluster for this lab, it doesn't make sense to have a duplicate copy of the data, as that requires a 2nd server in the cluster.**




**Click Next to continue onto step #4.**
**Place a check next to "I agree" and click Next:**



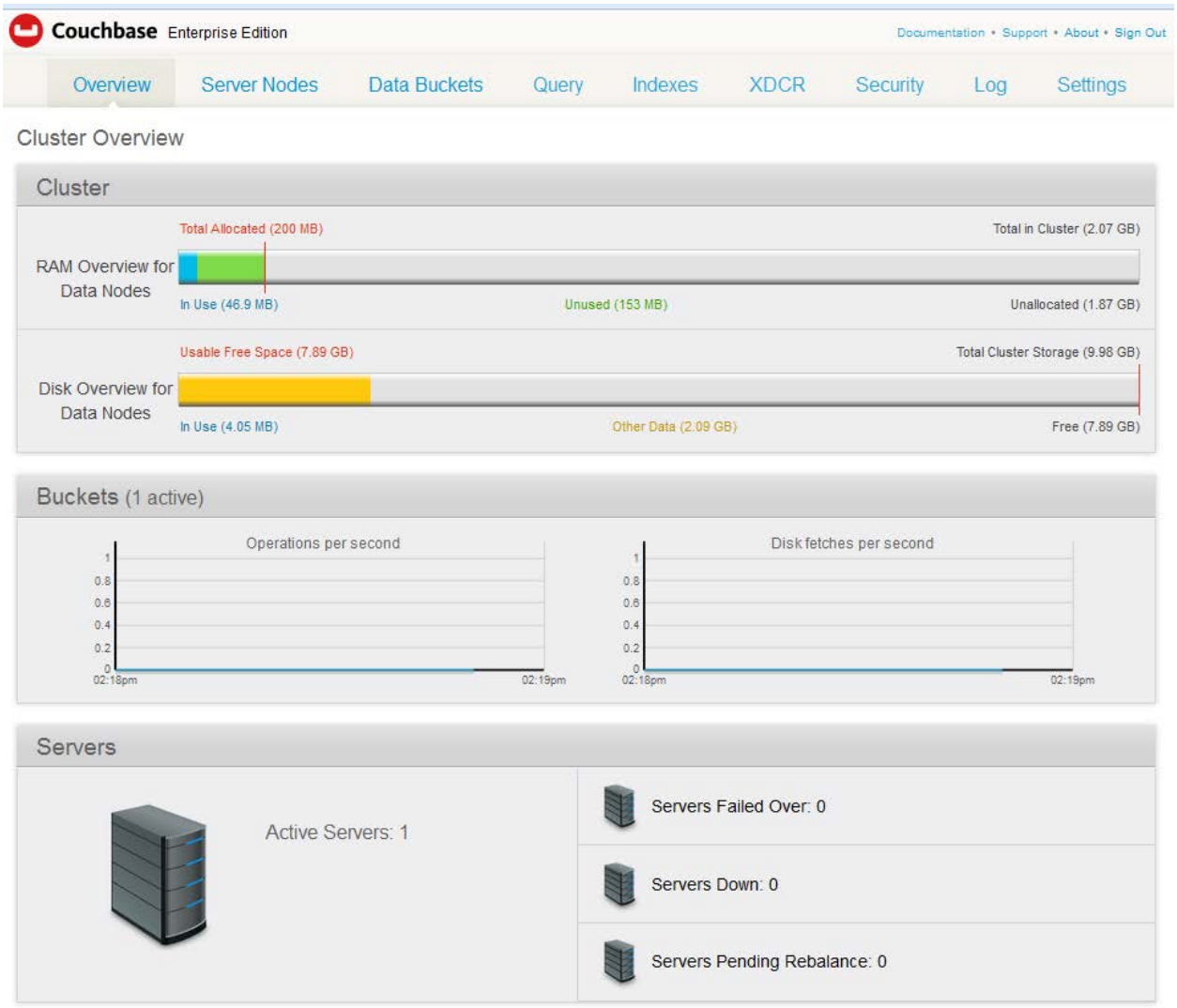

**Type 'couchbase' in lower case twice as the password and click Next:**




**In a few moments, you will see the Couchbase cluster dashboard:**

Couchbase

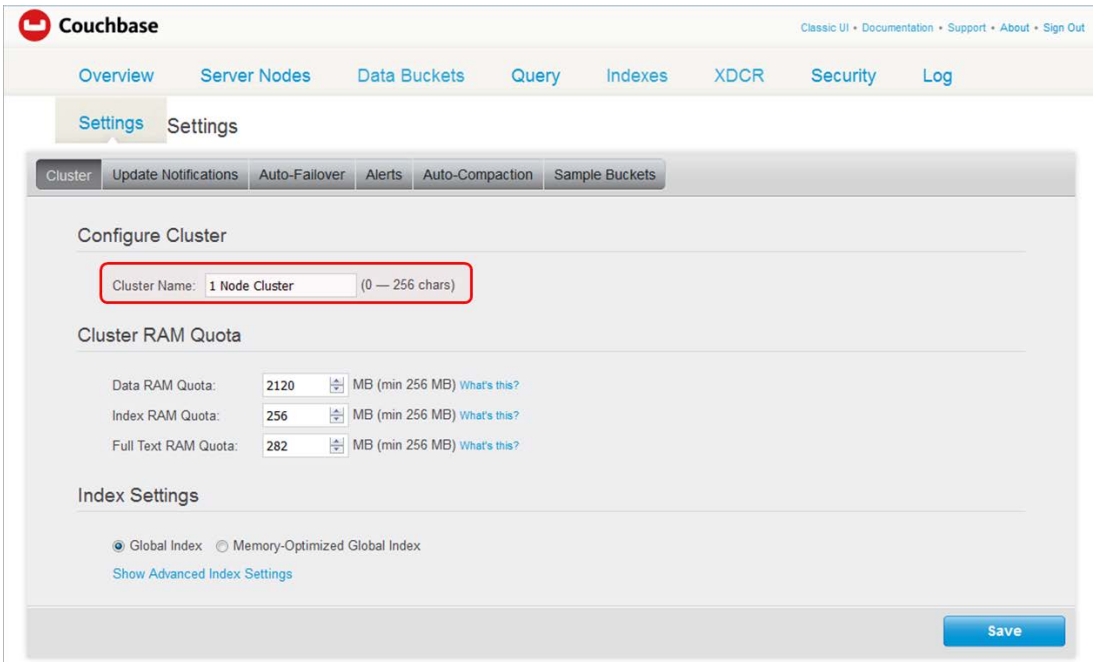




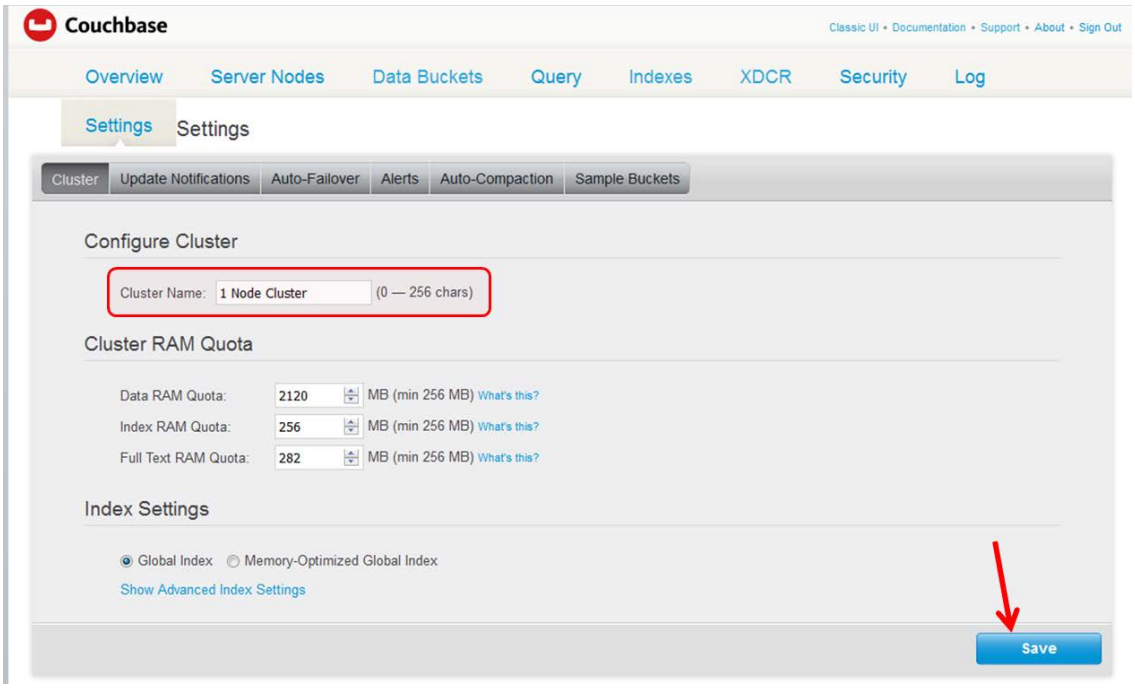

**Go to the settings tab**

Couchbase





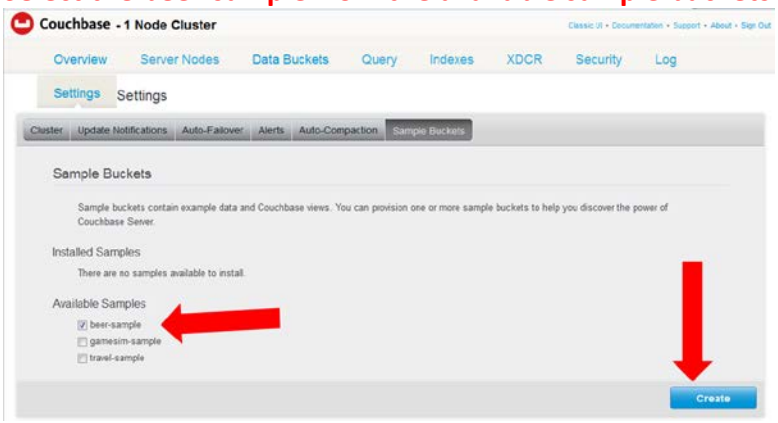

**Name your cluster "1 Node Cluster"**



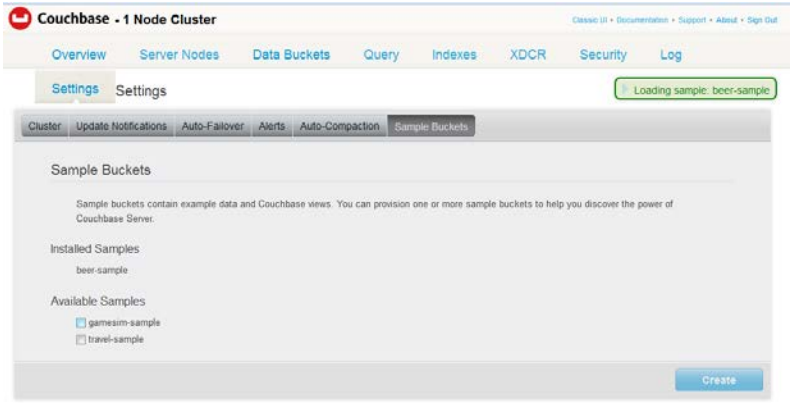

**Click Save**

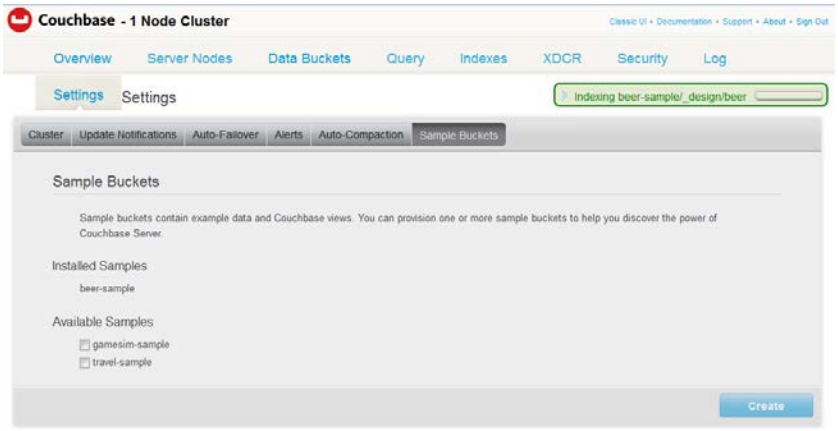
**Now click the "Sample buckets" sub-tab**

**Select the beer-sample from the available sample buckets and click Create**



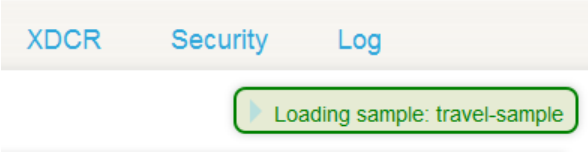

**Notice that the beer-sample databases are being loaded and indexed:**

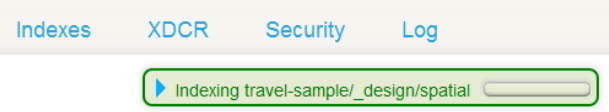


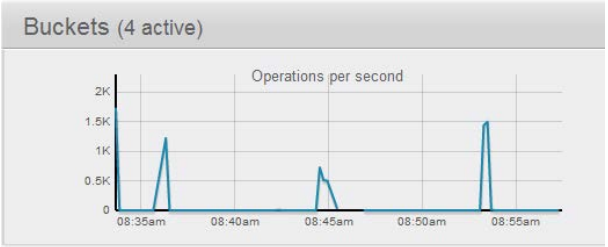
**Now repeat the steps necessary to load gamesim-sample and travel-sample**

**Observe loading and indexing**

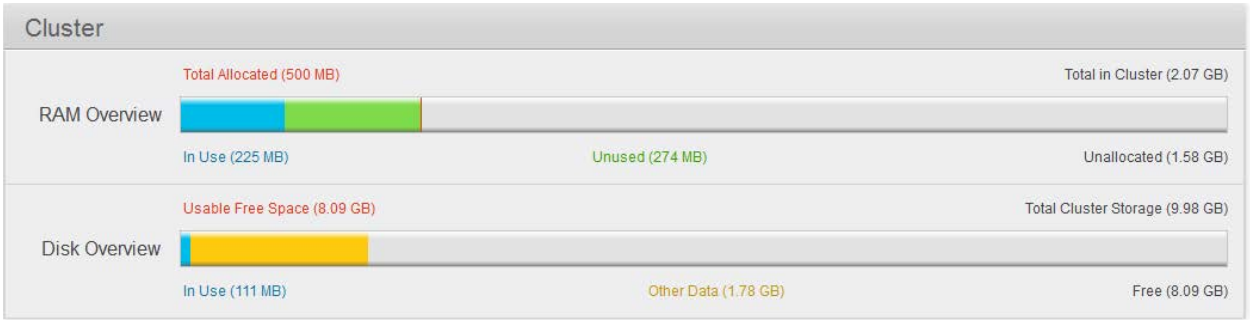

**You will start to see some Operations per second occurring as the sample databases are loaded:**



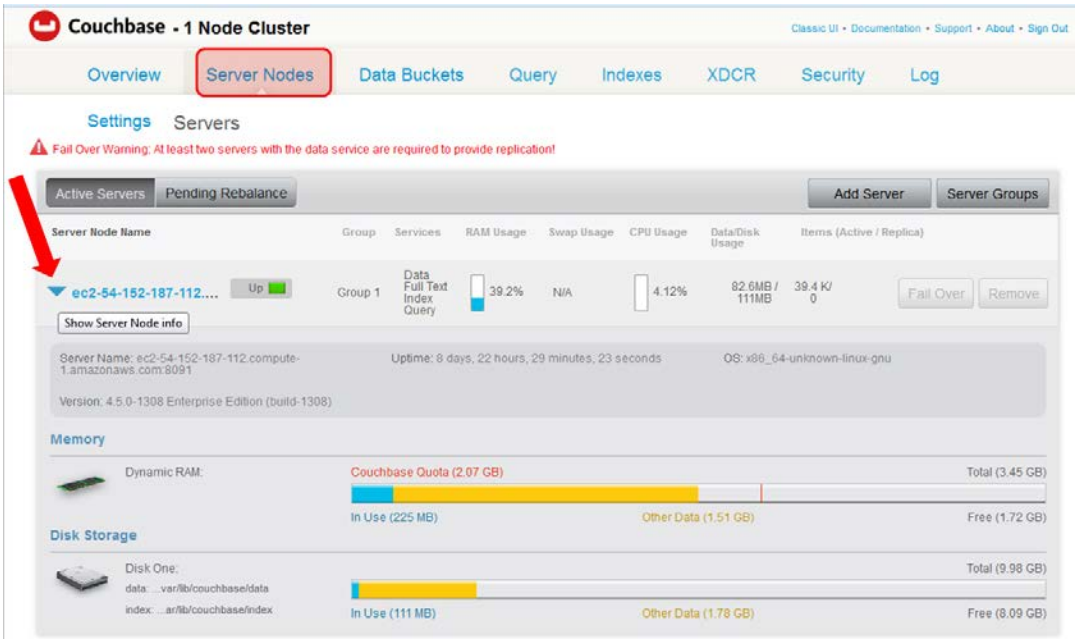

## Exploring the Couchbase Web UI:

Now that Couchbase is installed, let's do a quick tour of the Couchbase Web User Interface. On the main dashboard, notice that the total allocated RAM is 500MB and the total available RAM in the 1-node cluster is approximately 2.07 GB, but only 225MB is in use.

Also, the total cluster disk storage capacity is approximately 9.98 GB, and about 111 MB of the disk is being used by Couchbase data (this section should appear in blue in the 2nd bar, but it may be too small to show up) and there is approximately 1.78 GB of "other data" (yellow bar) on the server (probably OS files, etc).
Numbers may vary slightly due to packages loaded.



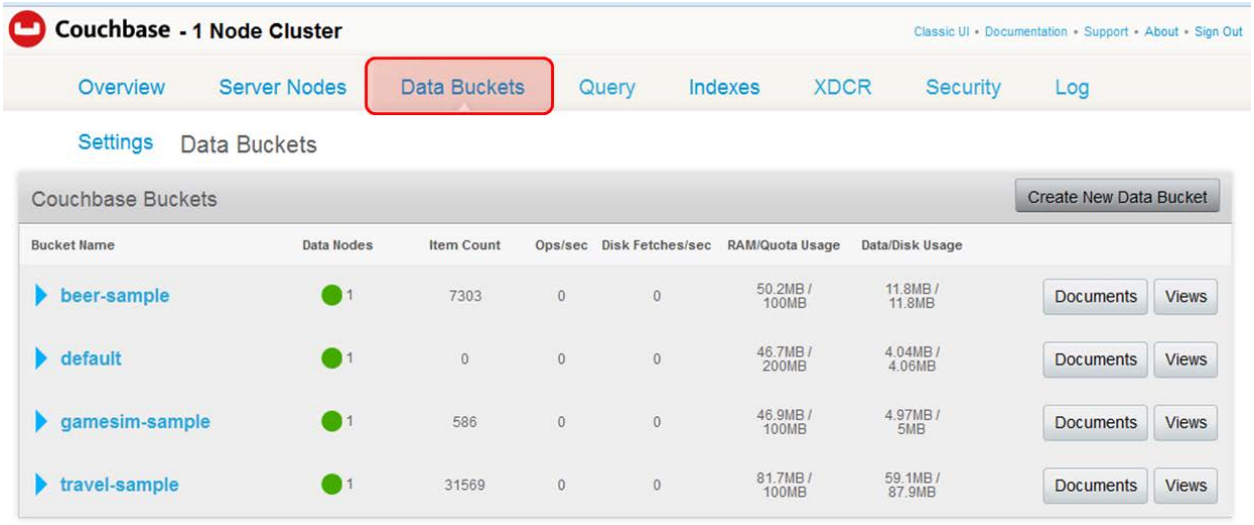

**Click on "Server Nodes" at the top and then expand the Blue Triangle dropdown next to the Server Node Name.**




Notice a few things there. The RAM usage on this single node is about 30-40% and the CPU usage is 3-5%. There is an "Add Server" button at the top right that we'll explore in a future lab to grow the cluster.

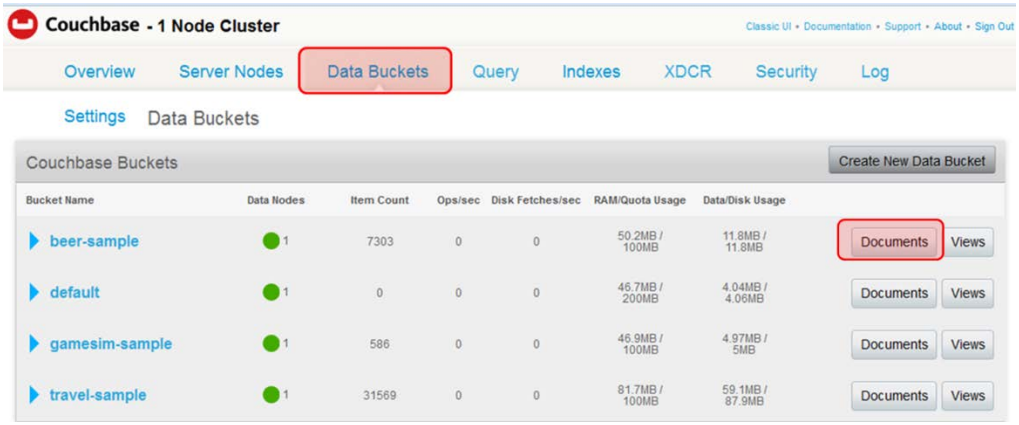
On the far right, you can also see that there are a 39.4k active items on the server and 0 replica items.

**Click on "Data Buckets" on the top menu:**

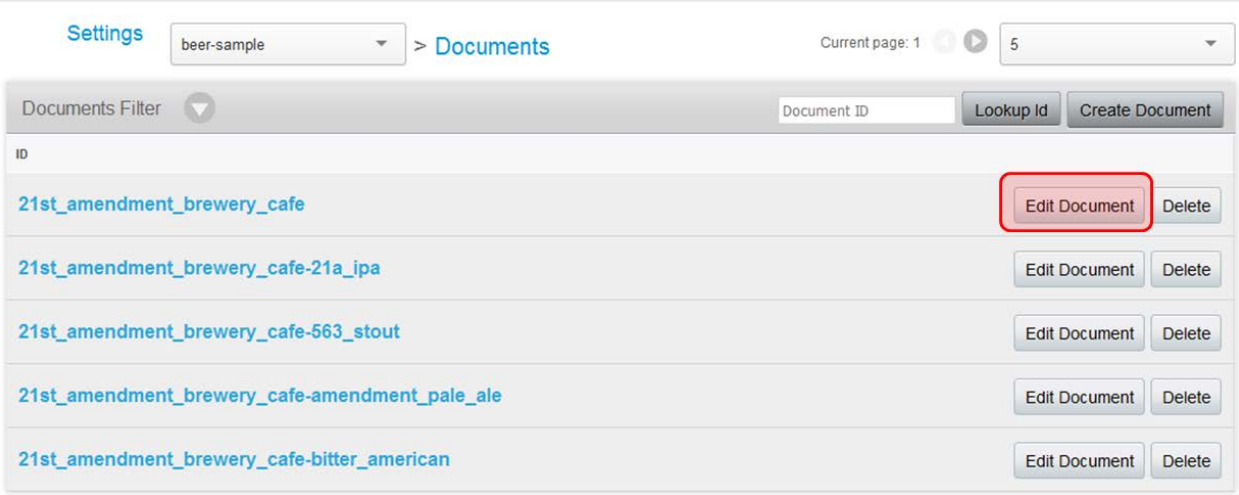



Here you will find 4 buckets. Three are sample buckets and one is the default bucket that gets installed with Couchbase.

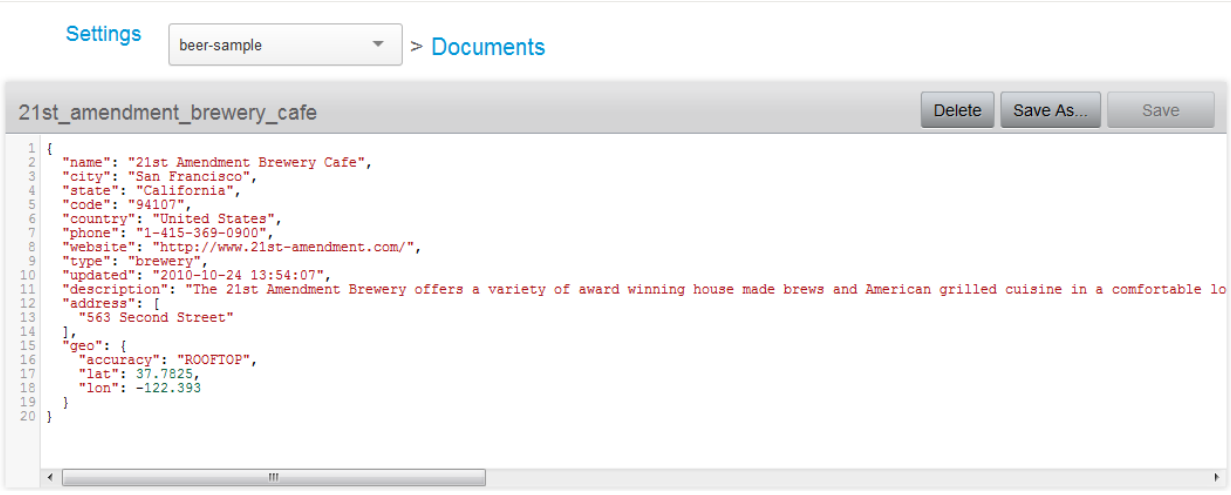
**Click on "Documents" next to the beer-sample bucket.**



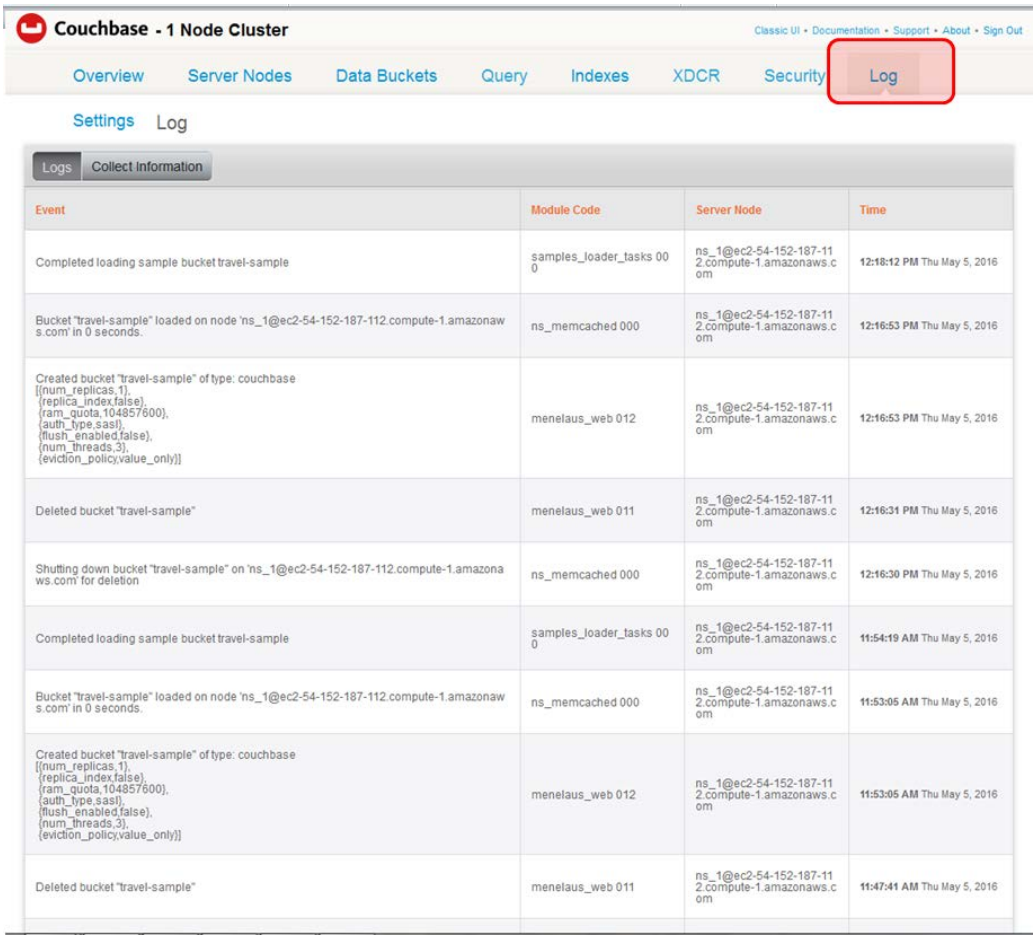

You will see the first 5 documents in this sample database displayed.
**Click on the first "Edit Document" button to see the full first document.**




The raw JSON document displays the first brewery listed in the database along with metadata like address, phone #, website, etc.




Next, we'll explore the Event Log for Couchbase. **At the top bar, click on "Log" and scroll down to the end of the page:**




After scrolling to the end of the page, you can see some of the first log messages generated by this Couchbase server as it started up. Notice the 2nd message: “I'm the only node, so I'm the master.” This essentially means that Node #1 is the Couchbase Orchestrator. The orchestrator node in Couchbase manages the rebalance process.

This examines the current vBucket map and then combines that information with the node additions and removals in order to create a new vBucket map.

The orchestrator starts the process of moving the individual vBuckets from the current vBucket map to the new vBucket structure. The process is only started by the orchestrator - the nodes themselves are responsible for actually performing the movement of data between the nodes. The aim is to make the newly calculated vBucket map match the current situation.

| | | | |
|---|---|---|---|
| Renamed node. New name is 'ns_1@ec2-54-172-130-66.compute-1.amazonaws.com'. | ns_cluster000 | ns_1@ec2-54-172-130-66.compute-1.amazonaws.com | **14:32:39** - Tue Oct 14, 2014 |
| Change of address to "ec2-54-172-130-66.compute-1.amazonaws.com" is requested. | ns_cluster000 | ns_1@127.0.0.1 | **14:32:39** - Tue Oct 14, 2014 |
| Changing address to "ec2-54-172-130-66.compute-1.amazonaws.com" due to client request | ns_cluster000 | ns_1@127.0.0.1 | **14:32:39** - Tue Oct 14, 2014 |
| Couchbase Server has started on web port 8091 on node 'ns_1@127.0.0.1'. Version: "3.0.0-1209-rel-enterprise". | menelaus_sup001 | ns_1@127.0.0.1 | **14:32:38** - Tue Oct 14, 2014 |
| I'm the only node, so I'm the master. | mb_master000 | ns_1@127.0.0.1 | **14:32:38** - Tue Oct 14, 2014 |
| Setting database directory path to /opt/couchbase/var/lib/couchbase/data and index directory path to /opt/couchbase/var/lib/couchbase/index | ns_storage_conf000 | ns_1@127.0.0.1 | **14:32:36** - Tue Oct 14, 2014 |
| Couchbase Server has started on web port 8091 on node 'ns_1@127.0.0.1'. Version: "3.0.0-1209-rel-enterprise". | menelaus_sup001 | ns_1@127.0.0.1 | **14:31:21** - Tue Oct 14, 2014 |
| Changed cluster compat mode from undefined to [3,0] | ns_orchestrator000 | ns_1@127.0.0.1 | **14:31:17** - Tue Oct 14, 2014 |
| I'm the only node, so I'm the master. | mb_master000 | ns_1@127.0.0.1 | **14:31:17** - Tue Oct 14, 2014 |
| Initial otp cookie generated: iutrhyacxacaygev | ns_cookie_manager003 | ns_1@127.0.0.1 | **14:31:16** - Tue Oct 14, 2014 |

## Exploring Couchbase cmd-line interface:

The couchbase-cli tool provides various management operations for Couchbase clusters, nodes and buckets.

The tool is located in the following directory location on linux:
/opt/couchbase/bin/couchbase-cli

**Add the /opt/couchbase/bin directory to your Linux PATH so that you can run the couchbase-cli tool (and other tools) by simply typing 'couchbase-cli' without providing the full directory path into the command.**

Note, you will need to use a linux text editor for the following section. You can use either nano, vi, vim or emacs to open the XML file and all future files. If you are unfamiliar with the arcane vi/vim or emacs syntax, I recommend using nano, one of the simplest text editors to use on Linux. My preference is vi & vim, so you will see me opening all files with these 2 editors for the rest of the labs, but feel free to replace the word 'vi' or 'vim' with 'nano' or 'emacs' on your end.

If you want a 3 min crash course in vi/vim, go to this link and graduate levels 1 and 2 and then come back: http://yannesposito.com/Scratch/en/blog/Learn-Vim-Progressively/

**Edit the .bash_profile file:**

```
[ec2-user@Couchbase01 ~]$ cd ~
[ec2-user@Couchbase01 ~]$ vi .bash_profile
```

Line 10 should currently show the following:

```
PATH=$PATH:$HOME/.local/bin:$HOME/bin
```

**Edit line 10 by appending the couchbase tools path to the end of the line, like so:**

```
PATH=$PATH:$HOME/.local/bin:$HOME/bin:/opt/couchbase/bin
```

**Save and quit the vi or nano session.**

**Source the .bash_profile file so that the changes you made take effect in the current bash session:**

```
[ec2-user@Couchbase01 ~]$ source ~/.bash_profile
```

```
[ec2-user@Couchbase01 ~]$ couchbase-cli
couchbase-cli - command-line cluster administration tool
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
  -s, --ssl             uses SSL for communication with secure servers
NOTE:
  USERNAME can be set in environment variable CB_REST_USERNAME and/or
  PASSWORD can be set in environment variable CB_REST_PASSWORD instead

usage: couchbase-cli COMMAND CLUSTER [OPTIONS]

COMMANDs include
  bucket-compact        compact database and index data
  bucket-create         add a new bucket to the cluster
  bucket-delete         delete an existing bucket
  bucket-edit           modify an existing bucket
  bucket-flush          flush all data from disk for a given bucket
  bucket-list           list all buckets in a cluster
  cluster-edit          modify cluster settings
  cluster-init       set the username,password and port of the cluster
  collect-logs-start    start a cluster-wide log collection
  collect-logs-status   show the status of cluster-wide log collection
  collect-logs-stop     stop a cluster-wide log collection
  failover              failover one or more servers
  group-manage          manage server groups
  help                  show longer usage/help and examples
  node-init             set node specific parameters
  rebalance             start a cluster rebalancing
  rebalance-status      show status of current cluster rebalancing
```

```
rebalance-stop          stop current cluster rebalancing
recovery                recover one or more servers
server-add              add one or more servers to the cluster
server-info             show details on one server
server-list             list all servers in a cluster
server-readd            readd a server that was failed over
setting-alert           set email alert settings
setting-audit           set audit settings
setting-autofailover    set auto failover settings
setting-cluster         set cluster settings
setting-compaction      set auto compaction settings
setting-index           set index settings
setting-ldap            set ldap settings
setting-notification    set notification settings
setting-xdcr            set xdcr related settings
ssl-manage              manage cluster certificate
user-manage             manage read only user
xdcr-replicate          xdcr operations
xdcr-setup              set up XDCR connection
```

**Test out a few common couchbase-cli commands to get familiar with the interface.**

**First, get your Couchbase hostname from the Cluster-IPs spreadsheet. You will need this for the next few commands. This will look something like: ec2-54-85-43-128.compute-1.amazonaws.com**

The basic usage format for CLI commands is:
`couchbase-cli COMMAND [BUCKET_NAME] CLUSTER [OPTIONS]`

We will provide the CLUSTER specification for the command using the long form syntax:
--cluster=HOST[:PORT]

The OPTIONS part includes the username and password to invoke the command as.

**Run the 'server-list' command. In this command, you will have to mention your public hostname and port.**

**Warning: you must change the hostname below to match your specific server's public hostname!**

```
[ec2-user@Couchbase01 ~]$ couchbase-cli server-list --cluster=ec2-54-174-65-105.compute-1.amazonaws.com:8091 --user=Administrator --password=couchbase
ns_1@ec2-54-174-65-105.compute-1.amazonaws.com ec2-54-174-65-105.compute-1.amazonaws.com:8091 healthy active
```

**Run the 'server-info' command, which requires the username and password also:**

**Note: If you followed the directions in the beginning of this lab, then your username should be 'Administrator' and password should be 'couchbase', all lowercase.**

## Exploring Couchbase index and data directories:

**Let's take a look at what's in the index and data directories on disk:**

```
[ec2-user@Couchbase01 ~]$ sudo ls -alh
/opt/couchbase/var/lib/couchbase/index
total 8.0K
drwxrwx---.  6 couchbase couchbase   56 May  5 11:33 .
drwxr-xr-x.  8 couchbase couchbase 4.0K May  5 14:14 ..
drwxr-x---. 10 couchbase couchbase 4.0K May  5 12:18 @2i
drwxrwx---.  4 couchbase couchbase   84 May  5 12:17 .delete
drwxrwx---.  2 couchbase couchbase   22 May  5 12:16 @fts
drwxrwx---.  5 couchbase couchbase   65 May  5 12:17 @indexes
```

```
[ec2-user@Couchbase01 ~]$ sudo ls -alh
/opt/couchbase/var/lib/couchbase/index/@indexes
total 8.0K
drwxrwx---. 5 couchbase couchbase   65 May  5 12:17 .
drwxrwx---. 6 couchbase couchbase   56 May  5 11:33 ..
drwxrwx---. 3 couchbase couchbase 4.0K May  5 11:36 beer-sample
drwxrwx---. 2 couchbase couchbase   57 May  5 11:42 gamesim-sample
drwxrwx---. 3 couchbase couchbase 4.0K May  5 12:17 travel-sample
```

```
[ec2-user@Couchbase01 ~]$ sudo ls -alh
/opt/couchbase/var/lib/couchbase/index/@indexes/beer-sample
total 756K
drwxrwx---. 3 couchbase couchbase 4.0K May  5 11:36 .
drwxrwx---. 5 couchbase couchbase   65 May  5 12:17 ..
-rw-rw----. 1 couchbase couchbase 752K May  5 11:36 main_5a222b8c920aa5e3a28b51ee7eb609a0.view.1
drwxrwx---. 2 couchbase couchbase    6 May  5 11:36 tmp_5a222b8c920aa5e3a28b51ee7eb609a0_main
```

Views within Couchbase process the information stored in the Couchbase database, allowing you to index and query your data. A view creates an index on the stored information according to the format and structure defined within the view. Views in Couchbase will be covered in depth later in the course.

**Next, explore the data directory, specifically the beer-sample database files:**

```
[ec2-user@Couchbase01 ~]$ sudo ls -alh
/opt/couchbase/var/lib/couchbase/data

total 228K
drwxrwx---. 7 couchbase couchbase 4.0K May  5 12:16 .
drwxr-xr-x. 8 couchbase couchbase 4.0K May  5 14:17 ..
drwxrwx---. 2 couchbase couchbase  28K May  5 14:17 beer-sample
drwxrwx---. 2 couchbase couchbase  28K May  5 14:18 default
drwxrwx---. 2 couchbase couchbase    6 May  5 12:16 .delete
drwxrwx---. 2 couchbase couchbase  28K May  5 14:17 gamesim-sample
-rw-rw----. 1 couchbase couchbase   34 Apr 26 13:35 _replicator.couch.1
drwxrwx---. 2 couchbase couchbase  28K May  5 14:18 travel-sample
-rw-rw----. 1 couchbase couchbase 4.1K Apr 26 13:34 _users.couch.1
```

```
[ec2-user@Couchbase01 ~]$ sudo ls -alh
/opt/couchbase/var/lib/couchbase/data/beer-sample
```

```
total 16M
drwxrwx---. 2 couchbase couchbase  28K May  5 14:18 .
drwxrwx---. 7 couchbase couchbase 4.0K May  5 12:16 ..
-rw-rw----. 1 couchbase couchbase  17K May  5 11:36 0.couch.1
-rw-rw----. 1 couchbase couchbase 8.1K May  5 11:36 1000.couch.1
-rw-rw----. 1 couchbase couchbase 8.1K May  5 11:36 1001.couch.1
-rw-rw----. 1 couchbase couchbase 8.1K May  5 11:36 1002.couch.1
-rw-rw----. 1 couchbase couchbase  13K May  5 11:36 1003.couch.1
-rw-rw----. 1 couchbase couchbase 8.1K May  5 11:36 1004.couch.1
-rw-rw----. 1 couchbase couchbase 8.1K May  5 11:36 1005.couch.1
-rw-rw----. 1 couchbase couchbase 8.1K May  5 11:36 1006.couch.1
-rw-rw----. 1 couchbase couchbase  13K May  5 11:36 1007.couch.1
-rw-rw----. 1 couchbase couchbase 8.1K May  5 11:36 1008.couch.1
-rw-rw----. 1 couchbase couchbase 8.1K May  5 11:36 1009.couch.1
-rw-rw----. 1 couchbase couchbase  13K May  5 11:36 100.couch.1
-rw-rw----. 1 couchbase couchbase  13K May  5 11:36 1010.couch.1
-rw-rw----. 1 couchbase couchbase 8.1K May  5 11:36 1011.couch.1
-rw-rw----. 1 couchbase couchbase 8.1K May  5 11:36 1012.couch.1
-rw-rw----. 1 couchbase couchbase 8.1K May  5 11:36 1013.couch.1
-rw-rw----. 1 couchbase couchbase 8.1K May  5 11:36 1014.couch.1
-rw-rw----. 1 couchbase couchbase  13K May  5 11:36 1015.couch.1
 <output truncated>
```

**Notice that there is 16 MB of data in this directory. (*Note: on your machine this can range between 15 – 35 MB*)**

**Try counting the number of files in this directory:**

```
[ec2-user@Couchbase01 ~]$ sudo ls -al
/opt/couchbase/var/lib/couchbase/data/beer-sample | wc -l
1030
```

**You should see about 1030 items. There will be 1024 couchbase partitions (vBucket files) along with a back index, metadata files, etc.**

**Next, run the couch_dbdump command to inspect one of the vbucket files in the beer-sample database:**

```
[ec2-user@Couchbase01 ~]$ sudo /opt/couchbase/bin/couch_dbdump
/opt/couchbase/var/lib/couchbase/data/beer-sample/0.couch.1 | head -20

Dumping "/opt/couchbase/var/lib/couchbase/data/beer-sample/0.couch.1":
Doc seq: 1
     id: lafayette_brewing-black_angus_oatmeal_stout
     rev: 1
     content_meta: 128
     size (on disk): 230
     cas: 1438282944544964608, expiry: 0, flags: 0, datatype: 1, conflict_resolution_mode: 0
     size: 233
     data: (snappy) {"name":"Black Angus Oatmeal
Stout","abv":0.0,"ibu":0.0,"srm":0.0,"upc":0,"type":"beer","brewery_id":"lafayette_brewing","upda
```

```
ted":"2010-07-22 20:00:20","description":"","style":"American-Style Stout","category":"North
American Ale"}
Doc seq: 2
     id: el_toro_brewing_company
     rev: 1
     content_meta: 128
     size (on disk): 1496
     cas: 1438282944545030144, expiry: 0, flags: 0, datatype: 1, conflict_resolution_mode: 0
     size: 1826
     data: (snappy) {"name":"El Toro Brewing Company","city":"Morgan
Hill","state":"California","code":"95037","country":"United States","phone":"408-782-
2739","website":"http://www.eltorobrewing.com/","type":"brewery","updated":"2010-07-22
20:00:20","description":"Geno and Cindy Acevedo founded El Toro Brewing Company in the summer of
1992. After much planning, research and construction business was begun in March of 1994. The
brewery is a small 17-barrel (527 gallon) batch system and operates on the Acevedo's rural
residential property.  Within seven months of opening El Toro Brewing Company received its first
national recognition at the 1994 Great American Beer Festival (GABF). One of its flagships El
Toro Oro Golden Ale won a coveted Gold Medal for English Pale Ale. At the 1996 GABF Poppy Jasper
Amber Ale, the brewery's other flagship beer, won a Silver Medal for English Brown Ale. It is a
mild yet robust brown ale. The 1997 GABF saw El Toro garner another Gold Medal for the American
styled wheat beer named after the brewer's father-in-law, William Jones Wheat Beer.  After 16
years of planning, Geno and Cindy Acevedo of Morgan Hill finally opened a brewpub on November 25
2006. El Toro Brewpub is a two floored building with patio. Inside is the world's only Poppy
Jasper Bar. Featuring over 45 feet of gorgeously inlaid and polished Poppy Jasper rock into its
surface. To the back of the large mirrored bar you will find over 25 beers and handcrafted sodas
on tap. We also have a gleaming copper Pub Brewing system 3 BBL (100 gallon) brewery. Live music
and dancing will also be a regular nighttime happening at the brewpub consisting of mostly local
bands playing cover type and original Rock, Blues, Jazz and Reggae.","address":["17605 Monterey
Road"],"geo":{"accuracy":"RANGE_INTERPOLATED","lat":37.1553,"lon":-121.676}}
Doc seq: 3
     id: st_austell_brewery-hsd_hicks_special_draught
     rev: 1
```

**Notice that there are 3(partial) JSON documents in this specific data file.**

**You can also print information about the data file with the couch_dbinfo command:**

```
[ec2-user@Couchbase01 ~]$ sudo /opt/couchbase/bin/couch_dbinfo
/opt/couchbase/var/lib/couchbase/data/beer-sample/0.couch.1

DB Info (/opt/couchbase/var/lib/couchbase/data/beer-sample/0.couch.1)
- header at 16384
   file format version: 12
   update_seq: 9
   purge_seq: 0
   crc: CRC-32C
   doc count: 9
   deleted doc count: 0
   data size: 6.6 kB
   B-tree size: 1.2 kB
   total disk size: 16.1 kB
```

**Notice that the above output shows the doc count in this data file along with the actual data size and the total disk file size (which includes metadata). (*The total disk file size in your specific environment can vary between 24 kB and 40 kB*) The # of deleted doc count is also shown, but in our sample database there are currently no deleted docs in the data file.**

## Startup and shutdown on Linux:

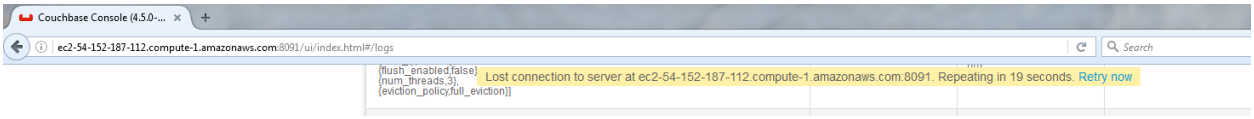
On Linux, Couchbase Server is installed as a standalone application with support for running as a background (daemon) process during startup through the use of a standard control script, /etc/init.d/couchbase-server. The startup script is automatically installed during installation.

By default, Couchbase Server is configured to be started automatically at run levels 2, 3, 4, and 5, and explicitly shutdown at run levels 0, 1 and 6.

**To manually stop Couchbase Server using the shutdown script:**

```
[ec2-user@Couchbase01 ~]$ sudo /etc/init.d/couchbase-server stop
Stopping couchbase-server (via systemctl):                 [  OK  ]
```

You may notice that the web UI will now start reporting that it has lost connection to the server:




**Before starting Couchbase back up, take a look at the info log file for Couchbase. Promote yourself to root and continue:**

```
[ec2-user@Couchbase01 ~]$ sudo -s

[root@Couchbase01 ec2-user]# cd /opt/couchbase/var/lib/couchbase/logs

[root@Couchbase01 logs]# ls -alh
total 24M
drwxr-xr-x. 2 couchbase couchbase  4.0K Apr 19 22:44 .
drwxr-xr-x. 8 couchbase couchbase  4.0K Apr 19 22:44 ..
-rw-rw----. 1 couchbase couchbase  824K Apr 19 22:44 babysitter.log
-rw-rw----. 1 couchbase couchbase  726K Apr 19 22:44 couchdb.log
-rw-rw----. 1 couchbase couchbase    13 Apr 19 22:44 crash_log.bin
-rw-rw----. 1 couchbase couchbase     0 Apr 19 22:44 crash_log.bin.tmp
-rw-rw----. 1 couchbase couchbase  5.2M Apr 19 22:44 debug.log
-rw-rw----. 1 couchbase couchbase  8.9K Apr 19 22:44 error.log
-rw-rw----. 1 couchbase couchbase   19K Apr 19 22:44 goxdcr.log
-rw-rw----. 1 couchbase couchbase  529K Apr 19 22:44 http_access.log
-rw-rw----. 1 couchbase couchbase 1001K Apr 19 22:44 indexer.log
-rw-rw----. 1 couchbase couchbase  3.1M Apr 19 22:44 info.log
-rw-rw--    1 couchbase couchbase 0 Apr 19 17:35 mapreduce_errors.log
-rw-rw---   1 couchbase couchbase 2.4M Apr 19 22:44 memcached.log.0.txt
-rw-rw----. 1 couchbase couchbase  1.5M Apr 19 22:44 ns_couchdb.log
-rw-rw----. 1 couchbase couchbase  615K Apr 19 22:44 projector.log
-rw-rw----. 1 couchbase couchbase   637 Apr 19 22:44 query.log
-rw-rw----. 1 couchbase couchbase  276K Apr 19 22:44 reports.log
-rw-rw----. 1 couchbase couchbase  8.5K Apr 19 18:31 ssl_proxy.log
-rw-r--r--. 1 root      root          9 Apr 19 17:35 start.log
-rw-rw----. 1 couchbase couchbase  7.3M Apr 19 22:44 stats.log
-rw-rw----. 1 couchbase couchbase  107K Apr 19 21:45 views.log
```

```
-rw-rw----. 1 couchbase couchbase     0 Apr 19 17:35 xdcr_errors.log
-rw-rw----. 1 couchbase couchbase  1.2K Apr 19 18:31 xdcr.log
-rw-rw----. 1 couchbase couchbase     0 Apr 19 17:35 xdcr_trace.log
```

**Print the last 21 lines of the info log file using the tail command:**

```
[root@Couchbase01 logs]# tail -21 info.log
[ns_server:info,2015-07-30T16:26:58.053-04:00,ns_1@ec2-52-6-74-39.compute-
1.amazonaws.com:<0.3478.0>:compaction_new_daemon:spawn_scheduled_views_compactor:493]Start
compaction of indexes for bucket gamesim-sample with config:
[{database_fragmentation_threshold,{30,undefined}},
 {view_fragmentation_threshold,{30,undefined}}]
[ns_server:info,2015-07-30T16:26:58.056-04:00,ns_1@ec2-52-6-74-39.compute-
1.amazonaws.com:<0.3486.0>:compaction_new_daemon:spawn_scheduled_views_compactor:493]Start
compaction of indexes for bucket beer-sample with config:
[{database_fragmentation_threshold,{30,undefined}},
 {view_fragmentation_threshold,{30,undefined}}]
[ns_server:info,2015-07-30T16:26:58.059-04:00,ns_1@ec2-52-6-74-39.compute-
1.amazonaws.com:<0.3494.0>:compaction_new_daemon:spawn_scheduled_views_compactor:493]Start
compaction of indexes for bucket default with config:
[{database_fragmentation_threshold,{30,undefined}},
 {view_fragmentation_threshold,{30,undefined}}]
[ns_server:info,2015-07-30T16:26:58.757-04:00,ns_1@ec2-52-6-74-39.compute-
1.amazonaws.com:<0.3554.0>:compaction_new_daemon:spawn_scheduled_kv_compactor:467]Start
compaction of vbuckets for bucket travel-sample with config:
[{database_fragmentation_threshold,{30,undefined}},
 {view_fragmentation_threshold,{30,undefined}}]
[ns_server:info,2015-07-30T16:26:58.760-04:00,ns_1@ec2-52-6-74-39.compute-
1.amazonaws.com:<0.3558.0>:compaction_new_daemon:spawn_scheduled_kv_compactor:467]Start
compaction of vbuckets for bucket gamesim-sample with config:
[{database_fragmentation_threshold,{30,undefined}},
 {view_fragmentation_threshold,{30,undefined}}]
[ns_server:info,2015-07-30T16:26:58.762-04:00,ns_1@ec2-52-6-74-39.compute-
1.amazonaws.com:<0.3562.0>:compaction_new_daemon:spawn_scheduled_kv_compactor:467]Start
compaction of vbuckets for bucket beer-sample with config:
[{database_fragmentation_threshold,{30,undefined}},
 {view_fragmentation_threshold,{30,undefined}}]
[ns_server:info,2015-07-30T16:26:58.764-04:00,ns_1@ec2-52-6-74-39.compute-
1.amazonaws.com:<0.3566.0>:compaction_new_daemon:spawn_scheduled_kv_compactor:467]Start
compaction of vbuckets for bucket default with config:
[{database_fragmentation_threshold,{30,undefined}},
 {view_fragmentation_threshold,{30,undefined}}]

[root@Couchbase01 logs]# exit
exit
```

**We will explore the rest of the logs in this directory in future labs, but if you're interested in what the rest of the logs contain, go here:**
http://docs.couchbase.com/admin/admin/Misc/Trbl-logs.html

**Manually start Couchbase Server back up:**

```
[ec2-user@Couchbase01 ~]$ sudo /etc/init.d/couchbase-server start
Starting couchbase-server (via systemctl):                 [  OK  ]
```

There may be cases where you want to explicitly shutdown a server and then restart it. Typically the server had been running for a while and has data stored on disk when you restart

it. In this case, the server needs to undergo a warmup process before it can again serve data requests.

Warmup is a process a restarted server must undergo before it can serve data. During this process the server loads items persisted on disk into RAM. One approach to load data is to do sequential loading of items from disk into RAM; however it is not necessarily an effective process because the server does not take into account whether the items are frequently used. In Couchbase Server, additional optimizations are provided during the warmup process to make data more rapidly available, and to prioritize frequently-used items in an access log. The server pre-fetches a list of most-frequently accessed keys and fetches these documents before it fetches any other items from disk.

Warmup will be discussed in more depth in a future lab. For now, you should know that the server can also switch into a ready mode before it has actually retrieved all documents for keys into RAM, and therefore can begin serving data before it has loaded all stored items. This is also a setting you can configure so that server warmup is faster.

**Wait a full 40 seconds to allow Couchbase to start up, before running the status command:**

```
[ec2-user@Couchbase01 ~]$ sudo /etc/init.d/couchbase-server status
couchbase-server is running
```

**This concludes the first lab.**