CouchDB is used as the database in the current project. It is setup as a 3-node cluster with Instance 1 (115.146.95.146) as the master node and instance 2 (115.146.95.150) and instance 3 (115.146.94.217) are the respective slaves.

Installation:

CouchDB is setup as containers using Docker, Hence, installation of Docker followed by CouchDB images are required.

Note: The below installation steps are required to be executed on instances 1,2 and 3 .

Step1: Remove older installations of Docker

sudo apt-get remove docker docker-engine docker.io

Step2: Update Repo and install software-properties-common

sudo apt-get update

sudo apt-get install \

apt-transport-https \

ca-certificates \

curl \

software-properties-common

Step3: Install Latest Stable Docker Release:

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo apt-key fingerprint 0EBFCD88

pub 4096R/0EBFCD88 2017-02-22

Key fingerprint = 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88

uid Docker Release (CE deb) <docker@docker.com>

sub 4096R/F273FCD8 2017-02-22

sudo add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) \

stable"

sudo apt-get update

sudo apt-get install docker-ce

Step4: Test Docker Installation by running the hello-world image.

sudo docker run hello-word

Step5: Install CouchDB Image on docker

docker pull couchdb:2.1.1

Runs and Configuration:

Step1: Setup a Security Group and Create Rules

Instances should be able to communicate with each other, in order to do this we have to setup a security group and create rules that allows the instances in the same group to communicate with each other. Once the group is ready we have to attach it to instances.

Step2: Connect to Instances 1,2,3 and run docker containers and also make sure you expose the port to the host and we can test this by executing ‘curl localhost:5984’ on one of the VMs that installed Couchdb.

Instance1:

sudo docker run --name couch\_master -e COUCHDB\_USER=admin -e COUCHDB\_PASSWORD=admin -e NODENAME=115.146.95.146 -v /database:/opt/couchdb/data -p 5984:5984 -p 4369:4369 -p 5986:5986 -p 9100-9200:9100-9200 -d couchdb:2.1.1

Instance2:

sudo docker run --name couch\_master -e COUCHDB\_USER=admin -e COUCHDB\_PASSWORD=admin -e NODENAME=115.146.95.150 -v /database:/opt/couchdb/data -p 5984:5984 -p 4369:4369 -p 5986:5986 -p 9100-9200:9100-9200 -d couchdb:2.1.1

Instance3:

sudo docker run --name couch\_master -e COUCHDB\_USER=admin -e COUCHDB\_PASSWORD=admin -e NODENAME=115.146.94.217 -v /database:/opt/couchdb/data -p 5984:5984 -p 4369:4369 -p 5986:5986 -p 9100-9200:9100-9200 -d couchdb:2.1.1

After executing ‘curl localhost:5984’ after running the containers we would see something as below,

*{"couchdb":"Welcome","version":"2.1.1","features":["scheduler"],"vendor":{"name":"The Apache Software Foundation"}}*

Make sure we can reach the Couchdb from another instance in the cluster, to do this, try `curl <remote-instance-ip>:5984`, do this is all 3 instances.

Once the above steps are completed and we see the required outputs , then we start to setup the cluster. Run these commands to add each node to the cluster (except for Instance1 "master" node):

curl -X POST -H ‘Content-Type: application/json’ http://admin:admin@127.0.0.1:5984/\_cluster\_setup -d “{\”action\”: \”enable\_cluster\”, \”bind\_address\”:\”0.0.0.0\”, \”username\”: \”admin\”, \”password\”:\”admin\”, \”port\”: 5984, \”node\_count\”: \”3\”, \”remote\_node\”: \”115.146.95.146\”, \”remote\_current\_user\”: \”admin\”, \”remote\_current\_password\”: \”admin\”}”

curl -X POST -H 'Content-Type: application/json' http://admin:admin@127.0.0.1:5984/\_cluster\_setup -d "{\"action\": \"add\_node\", \"host\":\"115.146.95.146\", \"port\": 5984, \"username\": \”admin\”, \"password\”:\”admin\”}”

Connect to Instance1 i.e. the master node and execute the below command to complete clustering.

curl -X POST "http://admin:admin@localhost:5984/\_cluster\_setup" -H 'Content-Type: application/json' -d '{"action": "finish\_cluster"}'

To check whether all the nodes are successfully installed and are in a cluster, we execute the below command and check the members of the cluster,

curl -X GET http://admin:admin@localhost:5984/\_membership