**To get started, you can start the nodes by just doing:**

Go to the directory

**Cd /ncrvms/**

**Vagrant up**

**NOTE:** If you prefer a different Vagrant box, you can set the DEMO\_BOX\_NAME environment variable before starting vagrant like this: agent1.vm.box = "centos/7"" vagrant up

Once it is finished, type the following:

**Vagrant status**

**Accessing the Consul Web UI**

Once the cluster is up-and-running you will be able to access the from a browser running on your host workstation by going to the following

Consul Web UI -URL: <http://172.20.20.40:8500/ui/>.

Docker apache agent1 -URL: <http://172.20.20.10:8080/sample/>.

Docker apache agent2 -URL: <http://172.20.20.20:8080/sample/>.

Docker apache agent3 -URL: <http://172.20.20.30:8080/sample/>.

**Vagrant Consul cluster/Redis/docker**

This vagrant file creates three Consul Server agent nodes and webuiportal server for test purpose.

All are running a standard Debian Ubuntu distribution.

**In this vagrant file I have created four virtual-machines namely:**

Agent1-Primary (172.20.20.10)

Agent 2 (172.20.20.20)

Agent 3 (172.20.20.30)

Webuiportal (172.20.20.40)

**Procedure:**

In Vagrant file I have invoked bootstrap.sh script and I passed the arguments in all three agents virtual- machines.

**--> Bootstrap.sh**

In this script I have installed and configured the services like consul, Redis and docker

**--> Webportal.sh**

In this webportal I wrote separate script for testing the consul.

I have installed the consul cluster and added all agents to it

Webportal: Cd ncrvms\webuiportal

**--> Dockerfile**

I have created a script for running the docker container with tomcat application server and I deployed the sample.war application in apache tomcat

**Cd /ncrvms/dockerfile**

**--> Config.json:**

For every agent there is a separate consul Config.json file in that I have done clustering

Agent1: **cd ncrvms\agent1\Config.json**

Agent2: **cd ncrvms\agent2\Config.json**

Agent3: **cd ncrvms\agent3\Config.json**