Below are the details of the test. The test is divided into 6 steps, beginning with setting up the infrastructure to installing software and configuring the servers. Our preferred infrastructure automation tools are Terraform and Ansible, but it's not mandatory to use them. Feel free to use the tools of your choice.

1) Create two EC2 Instances in AWS Cloud using,

   Additional Information

   - Instance Type of both instance is t2.micro
   - Operating System for both instances Ubuntu Server 16.04 LTS
   - Hostname of Instance 1 : MSR-test-Instance-1
   - Hostname of Instance 2 : MSR-test-Instance-2

   Preferred tools but not mandatory – Terraform

2) Once these two servers are provisioned, ensure the below following software packages are installed using configuration management tool in both the provisioned instances.

   Additional Information

   - NVM – Version 0.33.2

```
sudo apt-get update
sudo apt-get install nodejs
sudo apt-get install npm
nodejs -v
npm -v
```

   - Node – 8.12.0

```
sudo apt-get install curl python-software-properties
curl -sL https://deb.nodesource.com/setup_11.x | sudo -E bash -

sudo apt-get install curl python-software-properties
curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -

sudo apt-get install nodejs

node -v
```

- Docker – 18.06 or latest

```
sudo apt-get remove docker docker-engine docker.io containerd runc
```

```
sudo apt-get update

sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg-agent \
    software-properties-common

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

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io

systemctl start docker

systemctl enable docker

systemctl status docker
```

- Docker Compose – 1.13 or latest

```
sudo curl -L
"https://github.com/docker/compose/releases/download/1.23.2/docker-compose-$(uname
-s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose

sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
docker-compose --version
```

- Openssl – latest version

```
cd /tmp

wget https://www.openssl.org/source/openssl-1.1.1.tar.gz

tar xvf openssl-1.1.1.tar.gz
```

```
cd openssl-1.1.1

sudo ./config -Wl,--enable-new-dtags,-rpath,'$(LIBRPATH)'

sudo make

sudo make install
```

- Git – latest version

```
sudo apt-get update

sudo apt-get install git
```

git --version

Installed Manually with above commands

Jenkins Install:-

```
sudo apt update
sudo apt install openjdk-8-jdk

wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -

sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ >
/etc/apt/sources.list.d/jenkins.list'

sudo apt update
sudo apt install jenkins
systemctl start jenkins
systemctl enable jenkins
systemctl status jenkins
```

3) Create a Docker Container in MSR-test-Instance-1 using Docker Compose file and ensure apache webserver is installed. Try to use configuration management tools to automate the entire installation of apache and deploy a sample html file from a GitHub repository.

Additional Information

- You can create your own GitHub repository with a sample html file.

Preferred tools – Chef / Puppet / Salt stack / Ansible. (Note – Ansible is Preferred)

4) Create a Docker Container in MSR-test-Instance-2 using Docker Compose file and ensure CouchDB Database is installed. Try to use any configuration management tool to automate the entire installation processes.
5) http://13.234.94.122:4000/
6) http://13.234.88.195:4000/

Additional Information

- We should be able to access the Futon – web GUI of CouchDB, from the external system.

Preferred tools – Chef / Puppet / Salt stack / Ansible. (Note – Ansible is Preferred)

7) http://13.234.94.122:8080/
8) http://13.234.88.195:8080/
9)

10) Commit all the code/files to GitHub and write your explanations and documentations into the readme.
11) It will be an added advantage if you can draft the step by step procedure in performing the above activities and how to execute the code.