To deploy the application into any environment of your choice, please follow the below steps.

Our environment:

- Our solution is hosted on AWS cloud on Linux Red Hat 7.2.1-2 operating system

Getting the environment ready

- Connect to Linux Red Hat 7.2.1-2
  ssh -i ../aws/dockertest.pem ec2-user@13.57.242.90

- Run "sudo yum update" to apply all updates.
  sudo yum update -y

- Install Git on the server
  sudo yum install -y git

- Install Docker on the server
  sudo yum install -y docker

- Check the Docker status by running below command

```
[ec2-user@ip-172-31-23-165 ~]$ sudo service docker status
docker (pid  8414) is running...
[ec2-user@ip-172-31-23-165 ~]$
```

- Start the docker daemon
  sudo service docker start
- Check the docker version with below command

```
[ec2-user@ip-172-31-23-165 ~]$ sudo docker version
Client:
 Version:      17.09.1-ce
 API version:  1.32
 Go version:   go1.8.4
 Git commit:   3dfb8343b139d6342acfd9975d7f1068b5b1c3d3
 Built:        Mon Jan  8 22:44:09 2018
 OS/Arch:      linux/amd64

Server:
 Version:      17.09.1-ce
 API version:  1.32 (minimum version 1.12)
 Go version:   go1.8.4
 Git commit:   402dd4a/17.09.1-ce
 Built:        Mon Jan  8 22:44:42 2018
 OS/Arch:      linux/amd64
 Experimental: false
[ec2-user@ip-172-31-23-165 ~]$
```

- Build Jenkins Containers using below command

  sudo docker run --rm -d --user root -p 8080:8080 -p 50000:50000 --name mydockerjenkins --env JAVA_OPTS="-Xmx512m" -v jenkins_home:/var/jenkins_home -v /var/run/docker.sock:/var/run/docker.sock -v $(your docker):/bin/docker mydockerjenkins

- Build wso2 API Manager and wso2 dss container with below instructions

        - Use wget to pull the DSS 3.5.1 ZIP from a S3 bucket into the container /opt folder.
        - Install zip.
        - Unzip the DSS ZIP.
        - Remove the DSS ZIP.
        - Expose the container port 9443, 9763, 8243, 8280.
        - Set the wso2server.sh start-up script as the container entrypoint.
- Build NGINX with below instructions
        - $ docker run --name tmp-nginx-container -d nginx
        - $ docker cp tmp-nginx-container:/etc/nginx/nginx.conf /host/path/nginx.conf
        - $ docker rm -f tmp-nginx-container

Running the containers:

Before running the all the containers, please perform the following steps

1) Open ports for all above specified containers 8280, 8080, 5432, 22, 9763, 9443, 8244, 8281, 8088, 9444
2) Once these ports are opened then run below commands to run these containers

    - docker run --name my-custom-nginx-container -d custom-nginx
    - sudo docker run -d --user root -p 9443:9443 -p 9763:9763 -p 52000:52000 --name wso2dss xfusiontech/kmtrepo:wso2dss
    - sudo docker run -d --user root -p 8281:8281 -p 8244:8244 -p 9444:9444 -p 51000:51000 --name wso2am xfusiontech/kmtrepo:wso2am
    - sudo docker run --rm -d --user root -p 8080:8080 -p 50000:50000 --name mydockerjenkins --env JAVA_OPTS="-Xmx512m" -v jenkins_home:/var/jenkins_home -v /var/run/docker.sock:/var/run/docker.sock -v $(which docker):/bin/docker mydockerjenkins

3) You can also see the docker images status by running the below commands.

```
[ec2-user@ip-172-31-23-165 ~]$ sudo docker images
REPOSITORY            TAG                 IMAGE ID            CREATED             SIZE
xfusiontech/kmtrepo   kmtweb              7fbfae9677fc        17 minutes ago      128MB
nginx                 1.13                73acd1f0cfad        44 hours ago        109MB
xfusiontech/kmtrepo   latest              3f8b2a7c261d        3 days ago          124MB
wso2dss               3.5.1               56447056c141        6 days ago          1.48GB
xfusiontech/kmtrepo   wso2dss             56447056c141        6 days ago          1.48GB
wso2am                2.1.0               e01abfaa8ea9        8 days ago          9.24GB
xfusiontech/kmtrepo   wso2am              e01abfaa8ea9        8 days ago          9.24GB
node                  8.9                 672002a50a0b        3 weeks ago         676MB
nginx                 <none>              e548f1a579cf        3 weeks ago         109MB
mydockerjenkins       latest              11611158ec76        4 weeks ago         815MB
wso2/wso2base         latest              03b324ed733e        20 months ago       232MB
[ec2-user@ip-172-31-23-165 ~]$
```

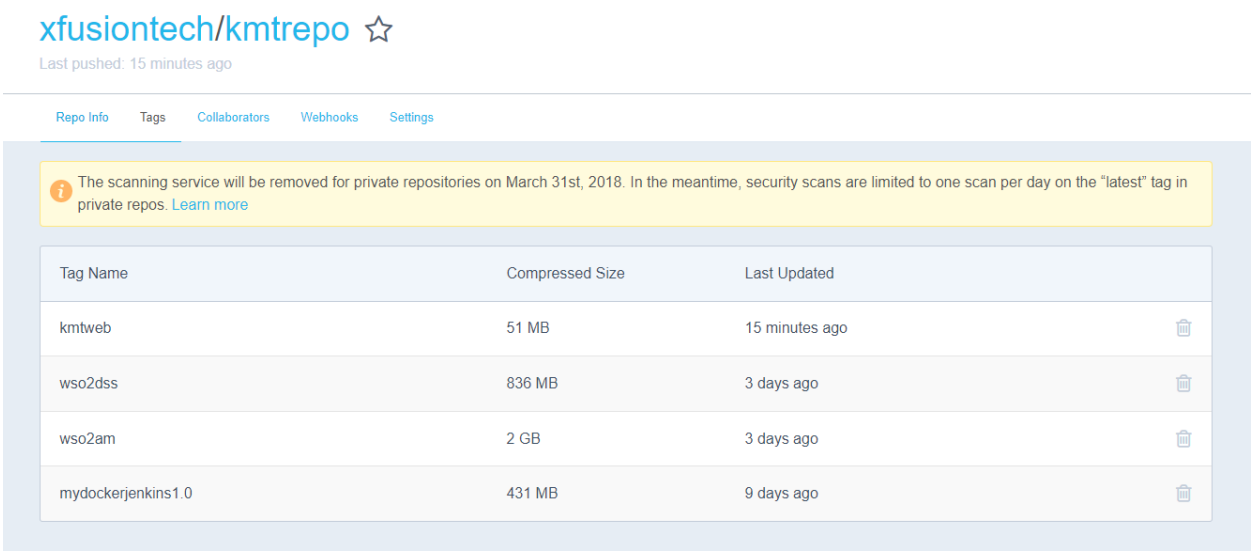
4) You can also see the container status by running below command

```
[ec2-user@ip-172-31-23-165 ~]$ sudo docker container ls -as
CONTAINER ID        IMAGE                          COMMAND                  CREATED             STATUS              PORTS
                                                                            NAMES               SIZE
5d967500536a        xfusiontech/kmtrepo:wso2am     "/usr/local/bin/in..."   6 days ago          Up 6 days           8243/tcp, 0.0.0.0:8244->8244/tcp, 8280/tcp, 0.0.0.0:8281->8281/tcp, 9443/tcp,
 9763/tcp, 0.0.0.0:9444->9444/tcp, 10397/tcp, 0.0.0.0:51000->51000/tcp   wso2am              3.75GB (virtual 13GB)
7ef9f9c6e1d9        xfusiontech/kmtrepo:wso2dss    "/usr/local/bin/in..."   6 days ago          Up 6 days           0.0.0.0:9443->9443/tcp, 0.0.0.0:9763->9763/tcp, 0.0.0.0:52000->52000/tcp
                                                                            wso2dss             41.2MB (virtual 1.52GB)
b7bcaccec6fa        mydockerjenkins                "/sbin/tini -- /us..."   11 days ago         Up 3 days           0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp
                                                                            mydockerjenkins     146MB (virtual 961MB)
[ec2-user@ip-172-31-23-165 ~]$
```

5) Use below commands to push the containers to private repository

- sudo docker login
- sudo docker tag 11611158ec76 xfusiontech/kmtrepo:jenkins1.0
- sudo docker push xfusiontech/kmtrepo

6) Once images are pushed then they will be available



The deployments to web and app servers are complete. Now you can test by going to the URL of the web server, for example, http://ec2-18-144-75-92.us-west-1.compute.amazonaws.com:8088/.