***Install Docker-CE***

vi /etc/yum.conf --> add proxy for yum [proxy=http://10.133.12.181:80]

***Install Docker-CE***

sudo yum install -y yum-utils device-mapper-persistent-data lvm2

sudo yum-config-manager --add-repo <https://download.docker.com/linux/centos/docker-ce.repo>

sudo yum install docker-ce

***Enable & Start Docker service:-***

sudo systemctl start docker

sudo systemctl enable docker

sudo systemctl status docker

***Disable Firewalld:-***

sudo systemctl stop firewalld.service

sudo systemctl disable firewalld.service

***IPtables:-***

sudo iptables -A INPUT -i docker0 -j ACCEPT

***Configure Docker Registry & partition:-***

sudo mkdir –p /docker/docker

sudo vi /etc/docker/daemon.json:-

{

"graph": "/docker/docker",

"storage-driver": "overlay",

~~// "hosts": ["unix:///var/run/docker.sock", "tcp://0.0.0.0:2375"],~~

"insecure-registries" : [ "~~\*\*\*currentMACHINE IP\*\*\*~~:5000"]

}

***Enable Docker to be accessible from remote machine:-***

sudo vi /usr/lib/systemd/system/docker.service :-

ExecStart=/usr/bin/dockerd -H unix:// -H tcp://0.0.0.0:2375

***Proxy setting for Docker:-***

sudo mkdir -p /etc/systemd/system/docker.service.d

sudo vi /etc/systemd/system/docker.service.d/http-proxy.conf :-

[Service]

Environment="HTTP\_PROXY=http://10.133.12.181:80/" "NO\_PROXY=~~\*current MACHINE IP\*~~"

***Reload Docker Daemon:-***

sudo systemctl daemon-reload

sudo systemctl restart docker

***Base Containers: ( HaProxy, Redis, Registry, DB, RabbitMQ, KeyCloak )***

***//copy haproxy from 114 & paste current machine (INP44XDAPP2998)***

[pmurugan@INP44XDAPP2523 ~]$ sudo scp -P 5522 -r /opt/haproxy/haproxy.cfg pmurugan@10.133.208.187:/tmp

[pmurugan@INP44XDAPP2998 opt]$ sudo mkdir haproxy

[pmurugan@ INP44XDAPP2998 opt]$ sudo chmod 777 haproxy/

[pmurugan@ INP44XDAPP2998 opt]$ sudo cp /tmp/haproxy.cfg /opt/haproxy/

[pmurugan@ INP44XDAPP2998 opt]$ sudo ls -l /opt/haproxy/

[pmurugan@ INP44XDAPP2998 opt]$ sudo chmod 777 /opt/haproxy/haproxy.cfg

***//Find INP44XDAPP2523’s IP address and replace with INP44XDAPP2998’s IP except “Key cloak server”***

***(refer : sample haproxy.cfg)***

sudo vi /opt/haproxy/haproxy.cfg

sudo docker run --name redis -p 6379:6379 --restart=always -v /docker/redis-data:/data -d redis:4.0.9

sudo docker run -d -p 5000:5000 --restart=always --name registry -v /docker/registry:/var/lib/registry registry:2

sudo docker run --name mariadb -p 3306:3306 --restart=always -v /docker/mysql/datadir:/var/lib/mysql -e MYSQL\_ROOT\_PASSWORD=adminpwd -e MYSQL\_USER=optimus -e MYSQL\_PASSWORD=OPTimus12#$ -d mariadb:10.2.17

sudo docker run -d --cap-add NET\_ADMIN --restart=always -p 443:443 -p 80:80 -v /opt/haproxy/haproxy.cfg:/etc/haproxy/haproxy.cfg million12/haproxy -n 10000

sudo docker run -d --restart=always -p 5672:5672 -p 15672:15672 --hostname my-rabbit -e RABBITMQ\_DEFAULT\_PASS=tata -e RABBITMQ\_DEFAULT\_USER=web --name rabbitmq rabbitmq:3-management

sudo docker run --name keyclock -e KEYCLOAK\_USER=admin -e KEYCLOAK\_PASSWORD=admin -e MYSQL\_ADDR=10.133.208.140 -e MYSQL\_PORT=3306 -e MYSQL\_USER=optimus -e MYSQL\_PASSWORD=OPTimus12#$ -e KEYCLOAK\_HTTPS\_PORT=8443 -p 5080:8080 -p 4080:8443 -p 9990:9990 -d jboss/keycloak

***@Jenkins Server to be Configured [for every new Registry]:***

Create a new item in Jenkins with Pipeline option

Update Jenkinsfile (replace old ip with new ip)(refer: sample jenkinsfile)

***Add CURRENT MACHINE’s IP address in below files on Jenkins machine:***

***[****pmurugan@INP44XDAPP2556 ~]$ sudo vi /etc/systemd/system/docker.service.d/http-proxy.conf*

*[pmurugan@INP44XDAPP2556 ~]$ sudo vi /etc/docker/daemon.json*

***Restart the servers on both Jenkins and current machine:***

[pmurugan@INP44XDAPP2556 ~]$ sudo systemctl daemon-reload

[pmurugan@INP44XDAPP2556 ~]$ sudo systemctl restart docker

[pmurugan@ INP44XDAPP2998 ~]$ sudo systemctl daemon-reload

[pmurugan@ INP44XDAPP2998 ~]$ sudo systemctl restart docker

***Add Database to CURRENT MACHINE***:

Tmp# sudo yum install mysql

Tmp# sudo mysqldump -uoptimus -pOPTimus12#$ -h 10.133.208.140 --single-transaction --databases customer\_uat location\_uat oms\_uat product\_catalog\_uat camunda notification\_nfw service\_inventory\_uat> optimus.sql

tmp# sudo docker cp /tmp/optimus.sql 05643df38d10:/root

#sudo docker exec -it 05643df38d10 mysql -u optimus -p

>use mysql

>GRANT ALL PRIVILEGES ON \*.\* TO 'optimus'@'localhost' IDENTIFIED BY '<password>';

>GRANT ALL PRIVILEGES on \*.\* to 'optimus'@'ip address' IDENTIFIED BY '<password>';

>FLUSH PRIVILEGES

>source /root/optimus.sql

>quit

#sudo docker restart ***<<mysql container id>>***

***Keycloak configuration***

<http://10.133.208.187/auth/>

username : admin

password : admin

clients-> optimus-> settings -> Valid redirect URI -> add Current machine ip addresss both http and https

***Sample haproxy.cfg***

[pmurugan@INP44XDAPP2998 ~]$ cat /opt/haproxy/haproxy.cfg

#---------------------------------------------------------------------

# Example configuration for a possible web application. See the

# full configuration options online.

#

# http://haproxy.1wt.eu/download/1.4/doc/configuration.txt

#

#---------------------------------------------------------------------

#---------------------------------------------------------------------

# Global settings

#---------------------------------------------------------------------

global

# to have these messages end up in /var/log/haproxy.log you will

# need to:

#

# 1) configure syslog to accept network log events. This is done

# by adding the '-r' option to the SYSLOGD\_OPTIONS in

# /etc/sysconfig/syslog

#

# 2) configure local2 events to go to the /var/log/haproxy.log

# file. A line like the following can be added to

# /etc/sysconfig/syslog

#

# local2.\* /var/log/haproxy.log

#

log 127.0.0.1 local2

chroot /var/lib/haproxy

pidfile /var/run/haproxy.pid

maxconn 4000

user haproxy

group haproxy

daemon

# turn on stats unix socket

stats socket /var/lib/haproxy/stats

#---------------------------------------------------------------------

# common defaults that all the 'listen' and 'backend' sections will

# use if not designated in their block

#---------------------------------------------------------------------

defaults

mode http

#mode tcp

log global

option httplog

option dontlognull

option http-server-close

option forwardfor except 127.0.0.0/8

option redispatch

retries 3

timeout http-request 10s

timeout queue 1m

timeout connect 10s

timeout client 1m

timeout server 1m

timeout http-keep-alive 10s

timeout check 10s

maxconn 3000

#---------------------------------------------------------------------

# main frontend which proxys to the backends

#---------------------------------------------------------------------

frontend main #0.0.0.0:80

bind \*:443 ssl crt /etc/ssl/dummy.pem alpn h2,http/1.1 #crt #/etc/haproxy/sample.pem

bind \*:80

mode http

#mode tcp

#acl delcookie hdr(Location) -m sub realms

#http-request del-header Cookie #if delcookie

#rspirep ^Location:\ http:(.\*) Location:\ https:\1

#acl sensitive\_folder path\_beg /optimus

#acl https ssl\_fc

#redirect scheme https if !https sensitive\_folder

http-response set-header Cache-Control no-cache

#http-response set-header Access-Control-Allow-Origin optimus-uat.tatacommunications.com

http-response set-header Access-Control-Allow-Methods POST,GET

http-response set-header Access-Control-Max-Age 3600

http-response set-header Access-Control-Allow-Headers x-auth-token,Origin,Accept,X-Requested-With,Content-Type,Access-Control-Request-Method,Access-Control-Request-Headers

http-response set-header X-Frame-Options SAMEORIGIN

#http-response set-header Strict-Transport-Security "max-age=16000000; includeSubDomains; preload;"

http-response set-header Referrer-Policy no-referrer-when-downgrade

http-response set-header X-Content-Type-Options nosniff

#http-response set-header X-XSS-Protection: 1;mode=block

compression algo gzip

compression type text/css text/html text/javascript application/javascript text/plain text/xml application/json

#acl httponly\_cookie res.hdr(Set-Cookie),lower -m sub httponly

#rspirep ^(set-cookie:.\*) \1;\ HttpOnly if !httponly\_cookie

acl missing\_slash path\_end /optimus

redirect code 301 prefix / append-slash if missing\_slash

#acl token\_app url\_beg /authentication

#rspirep ^Location:\ http:(.\*) Location:\ https:\1 if token\_app

# acl url\_static path\_beg -i /static /images /javascript /stylesheets

# acl url\_static path\_end -i .jpg .gif .png .css .js

#use\_backend static if url\_static

#default\_backend app

default\_backend ceph\_server

acl piwik path\_beg -i /piwik

use\_backend piwik\_server if piwik

acl gitlab path\_beg -i /gitlab

use\_backend gitlab\_server if gitlab

acl jenkins path\_beg -i /jenkins

use\_backend jenkins\_server if jenkins

acl cas path\_beg -i /sso

use\_backend cas\_server if cas

acl saml2 path\_beg -i /saml2

use\_backend cas\_server if saml2

acl utils path\_beg -i /utils/

use\_backend cas\_server if utils

acl key\_cloak path\_beg -i /auth/

use\_backend key\_cloak\_server if key\_cloak

acl optimus\_product path\_beg -i /optimus-product

use\_backend optimus\_product\_server if optimus\_product

acl optimus\_serviceinventory path\_beg -i /optimus-serviceinventory

use\_backend optimus\_serviceinventory\_server if optimus\_serviceinventory

acl optimus\_quote path\_beg -i /optimus-quote

use\_backend optimus\_quote\_server if optimus\_quote

acl optimus\_auth path\_beg -i /authentication/

use\_backend optimus\_auth\_server if optimus\_auth

acl optimus\_ui path\_beg -i /optimus/

use\_backend optimus\_ui\_server if optimus\_ui

acl optimus\_ui6 path\_beg -i /optimus\_ui6/

use\_backend optimus\_ui6\_server if optimus\_ui6

acl optimus\_notification path\_beg -i /optimus-notification/

use\_backend optimus\_notification\_server if optimus\_notification

acl optimus\_userservice path\_beg -i /optimus-userservice/

use\_backend optimus\_userservice\_server if optimus\_userservice

acl optimus\_location path\_beg -i /optimus-location/

use\_backend optimus\_location\_server if optimus\_location

acl optimus\_oms path\_beg -i /optimus-oms/

use\_backend optimus\_oms\_server if optimus\_oms

acl optimus\_customer path\_beg -i /optimus-customer/

use\_backend optimus\_customer\_server if optimus\_customer

acl optimus\_service path\_beg -i /optimus-service/

use\_backend optimus\_service\_server if optimus\_service

acl pdf path\_beg -i /optimus-pdf

use\_backend pdf\_server if pdf

acl optimus-template path\_beg -i /optimus-template

use\_backend html\_templates if optimus-template

acl soap path\_beg -i /orders

use\_backend soap\_server if soap

# acl rcode path\_beg -i /api

# use\_backend rcode\_server if rcode

acl rcode\_price path\_beg -i /pricing/api

use\_backend rcode\_price\_server if rcode\_price

acl rcode\_onnet\_wireline path\_beg -i /onnet-wireline

use\_backend rcode\_onnet\_wireline\_server if rcode\_onnet\_wireline

acl rcode\_offnet\_wireless path\_beg -i /offnet-wireless

use\_backend rcode\_offnet\_wireless\_server if rcode\_offnet\_wireless

acl rcode\_onnet\_wireless path\_beg -i /onnet-wireless

use\_backend rcode\_onnet\_wireless\_server if rcode\_onnet\_wireless

acl rcode\_price\_new path\_beg -i /new/pricing

use\_backend rcode\_price\_newserver if rcode\_price\_new

acl rcode\_onnet\_wireline\_new path\_beg -i /new/onnet-wireline

use\_backend rcode\_onnet\_wireline\_newserver if rcode\_onnet\_wireline\_new

acl rcode\_offnet\_wireless\_new path\_beg -i /new/offnet-wireless

use\_backend rcode\_offnet\_wireless\_newserver if rcode\_offnet\_wireless\_new

acl rcode\_onnet\_wireless\_new path\_beg -i /new/onnet-wireless

use\_backend rcode\_onnet\_wireless\_newserver if rcode\_onnet\_wireless\_new

#---------------------------------------------------------------------

# static backend for serving up images, stylesheets and such

#---------------------------------------------------------------------

backend static

balance roundrobin

server static 127.0.0.1:4331 check

backend gitlab\_server

server gitlab1 10.133.208.116:7080 check

backend ceph\_server

server ceph1 10.133.208.122:7000 check

backend jenkins\_server

server jenkins1 10.133.208.133:8080 check

backend cas\_server

server cas1 10.133.208.116:8080 check

backend piwik\_server

server piwik 10.133.208.133:75 check

backend jira\_server

server jira 10.133.208.116:9090 check

backend optimus\_product\_server

server optimus\_product\_server 10.133.208.187:9091 check

backend optimus\_serviceinventory\_server

server optimus\_serviceinventory\_server 10.133.208.187:9018 check

backend optimus\_auth\_server

server optimus\_auth\_server 10.133.208.187:6060 check

#server optimus\_auth\_server 10.133.208.133:5050 check

backend optimus\_ui\_server

server optimus\_ui\_server 10.133.208.187:8095 check

backend optimus\_ui6\_server

server optimus\_ui6\_server 10.133.208.187:8096 check

backend optimus\_notification\_server

server optimus\_notification\_server 10.133.208.187:8090 check

backend optimus\_userservice\_server

server optimus\_userservice\_server 10.133.208.187:8085 check

backend key\_cloak\_server

#mode tcp

#server key\_cloak\_server 10.133.208.116:8443 check ssl verify none

#server key\_cloak\_server 10.133.208.142:443 check ssl verify none

#server key\_cloak\_server 10.133.208.142:80 check #LAST\_WORKING

#http-request add-header X-Forwarded-Proto https

server key\_cloak\_server 10.133.208.114:5080 check

#server key\_cloak\_server 10.133.208.116:7080 check

#server key\_cloak\_server 10.133.208.116:5080 check

backend optimus\_quote\_server

server optimus\_quote\_server 10.133.208.187:8070 check

backend optimus\_location\_server

server optimus\_location\_server 10.133.208.187:7071 check

backend optimus\_oms\_server

server optimus\_oms\_server 10.133.208.187:7072 check

backend optimus\_customer\_server

server optimus\_customer\_server 10.133.208.187:7073 check

backend optimus\_service\_server

server optimus\_service\_server 10.133.208.187:7074 check

backend pdf\_server

#server node-pdf 10.133.208.124:8000 check

server node-pdf 10.133.208.187:8000 check

backend html\_templates

server optimus-template 10.133.208.187:2080 check

backend soap\_server

server soap 10.133.208.133:7075 check

backend rcode\_offnet\_wireless\_server

server rcode\_offnet\_wireless 10.133.208.120:7000 check

backend rcode\_price\_server

server rcode\_price 10.133.208.120:2100 check

backend rcode\_onnet\_wireless\_server

server rcode\_onnet\_wireless 10.133.208.120:9000 check

backend rcode\_onnet\_wireline\_server

server rcode\_onnet\_wireline 10.133.208.120:6000 check

backend rcode\_offnet\_wireless\_newserver

server rcode\_offnet\_wireless\_new 10.133.208.120:7100 check

backend rcode\_price\_newserver

server rcode\_price\_new 10.133.208.120:2100 check

backend rcode\_onnet\_wireless\_newserver

server rcode\_onnet\_wireless\_new 10.133.208.120:9100 check

backend rcode\_onnet\_wireline\_newserver

server rcode\_onnet\_wireline\_new 10.133.208.120:6200 check

#---------------------------------------------------------------------

# round robin balancing between the various backends

#---------------------------------------------------------------------

backend app

balance roundrobin

server app1 127.0.0.1:5001 check

server app2 127.0.0.1:5002 check

server app3 127.0.0.1:5003 check

server app4 127.0.0.1:5004 check

***Sample Jenkinsfile backend:***

node {

// Export proxy

stage('set proxy'){

sh 'export http\_proxy="http://10.133.12.181:80"'

sh 'export https\_proxy="http://10.133.12.181:80"'

}

//NPL - Environment Variables

stage('export environment variables'){

sh 'DB\_UNAME=optimus'

sh "DB\_PWD=OPTimus12#\$"

sh 'MQ\_HOST=10.133.208.133'

sh 'MQ\_PORT=5672'

sh 'MQ\_UNAME=web'

sh 'MQ\_PWD=tata'

sh 'AUTH\_URL=http://10.133.208.114/auth/'

sh 'REDIS\_HOST=10.133.208.114'

sh 'REDIS\_PORT=6379'

sh 'REDIS\_UNAME='

sh 'REDIS\_PWD='

sh 'LDAP\_URL=ldap://10.133.208.116:389'

sh 'LDAP\_BASE=dc=TCL'

sh 'LDAP\_UNAME=cn=Manager,dc=TCL'

sh 'LDAP\_PWD=AzXXb5lqpw1YgiHK5t'

sh 'KEYCLOAK\_PROXY='

//## ML F&P ##

sh 'R\_ONNET\_WIRELINE=http://10.133.208.121/new/onnet-wireline/api'

sh 'R\_ONNET\_WIRELESS=http://10.133.208.121/new/onnet-wireless/api'

sh 'R\_OFFNET\_WIRELESS=http://10.133.208.121/new/offnet-wireless/api'

sh 'R\_PRICING=http://10.133.208.121/new/pricing/api'

sh 'R\_GVPN\_PRICING=http://10.133.208.121/new/gvpn/pricing/api'

sh 'R\_GVPN\_IND\_PRICING=http://10.133.208.121/new/gvpn/ind-int/pricing/api'

sh 'R\_NPL\_PRICING=http://10.133.208.120:2500/new/npl/pricing/api'

sh 'R\_NPL\_ONNET\_POP=http://10.133.208.121/new/npl/pop/api'

sh 'R\_GVPN\_CQ\_POP=http://10.133.208.121/new/npl/pop/api'

sh 'R\_GSC\_PRICING=http://10.133.208.121/new/gsc/pricing/api'

sh 'R\_GVPN\_INT\_PRICING=http://10.133.208.121/new/gvpn/ind-int/pricing/api'

sh 'R\_NPL\_ONNET\_POP=http://10.133.208.121/new/npl/pop/api'

sh 'R\_GVPN\_INT\_PRICING=http://10.133.208.121/new/gvpn/international/pricing/api'

sh 'R\_GSC\_PRICING=http://10.133.208.121/new/gsc/pricing/api'

sh 'R\_GVPN\_CQ\_POP=http://10.133.208.121/new/gvpn/cqrank/api'

//## SFDC ##

sh 'SFDC\_UNAME=huzefa.tarwala@tatacommunications.com.tcluat'

sh "SFDC\_PWD=7ujm\\&UJMKmmcdUX7JprK1kwkoyOm8LSXx"

sh 'SFDC\_GRANT\_TYPE=password'

sh "SFDC\_AUTH=Basic Og=="

sh "SFDC\_CLIENT=3MVG9rKhT8ocoxGkPdSEUBFzU\_VZC0HhknudUIGBqUgoTG6mlBwbhnihq74XgyvhW5.fEOubuJg1t6LVDW04X"

sh 'SFDC\_SECRET=4585614469580292015'

sh 'SFDC\_AUTH\_URL=https://test.salesforce.com/services/oauth2/token'

sh 'SFDC\_GEN\_HOST=https://cs80.salesforce.com/services/apexrest/'

sh 'SFDC\_OPT\_CREATE\_PATH=GSC\_REST\_V9\_2/OpportunityService/'

sh 'SFDC\_PRD\_CREATE\_REQ\_URL=GSC\_REST\_V9\_2/Products\_Services\_Service/'

sh 'SFDC\_PRD\_DELETE\_URL=DeleteProductService'

sh 'SFDC\_STAGE\_UPDATE\_URL=GSC\_REST\_V9\_2/OpportunityService/'

sh 'SFDC\_SITE\_UPDATE\_URL=CreateSiteLocation'

//## SMTP ##

sh 'SMTP\_HOST=p44relay'

sh 'SMTP\_UNAME='

sh 'SMTP\_PWD='

sh 'SMTP\_PORT=25'

//## Internet Proxy ##

sh 'SYSTEM\_PROXY\_HOST=10.133.12.181'

sh 'SYSTEM\_PROXY\_PORT=80'

sh 'BUILD\_VERSION=Hydrogen1.0'

sh 'APP\_HOST=http://10.133.208.114'

sh 'APP\_ENV=DEV'

sh 'DOCUSIGN\_URL=https://demo.docusign.net/restapi'

//#DOCUSIGN\_URL=https://docusign.net/restapi

sh 'DOCUSIGN\_UNAME=support.customer@tatacommunications.com'

sh 'DOCUSIGN\_PWD=OPTimus12$'

sh 'DOCUSIGN\_KEY=73580ce7-ee53-4010-9f98-293a12fb9588'

}

withMaven(maven:'maven') {

stage('Checkout') {

//git url: 'http://10.133.208.121/gitlab/root/optimus.git', credentialsId: 'daf9c129-a7f0-4ca0-97c5-ed4e7900c652', branch: 'Hydrogen'

git url: 'http://10.133.208.121/gitlab/root/optimus.git', credentialsId: 'daf9c129-a7f0-4ca0-97c5-ed4e7900c652', branch: 'dev/izopc'

//git url: 'http://10.133.208.133/gitlab/root/optimus.git', credentialsId: '167ea1b4-c98b-41b1-9bd0-047282460d7f', branch: 'feature/GSC-Hydrogen'

}

stage('Build Backend Microservices') {

sh 'mvn clean install'

}

stage('Image') {

dir ('optimus-auth') {

sh 'sudo cp /opt/optimus/optimus-auth/Dockerfile .'

sh 'sudo docker build -t optimus-auth .'

sh 'sudo docker tag optimus-auth:latest 10.133.208.187:5000/optimus-auth:latest'

sh 'sudo docker push 10.133.208.187:5000/optimus-auth:latest'

}

dir ('optimus-customer') {

sh 'sudo cp /opt/optimus/optimus-customer/Dockerfile .'

sh 'sudo docker build -t optimus-customer .'

sh 'sudo docker tag optimus-customer:latest 10.133.208.187:5000/optimus-customer:latest'

sh 'sudo docker push 10.133.208.187:5000/optimus-customer:latest'

}

dir ('optimus-location') {

sh 'sudo cp /opt/optimus/optimus-location/Dockerfile .'

sh 'sudo docker build -t optimus-location .'

sh 'sudo docker tag optimus-location:latest 10.133.208.187:5000/optimus-location:latest'

sh 'sudo docker push 10.133.208.187:5000/optimus-location:latest'

}

dir ('optimus-notification') {

sh 'sudo cp /opt/optimus/optimus-notification/Dockerfile .'

sh 'sudo docker build -t optimus-notification .'

sh 'sudo docker tag optimus-notification:latest 10.133.208.187:5000/optimus-notification:latest'

sh 'sudo docker push 10.133.208.187:5000/optimus-notification:latest'

}

dir ('optimus-oms') {

sh 'sudo cp /opt/optimus/optimus-oms/Dockerfile .'

sh 'sudo docker build -t optimus-oms .'

sh 'sudo docker tag optimus-oms:latest 10.133.208.187:5000/optimus-oms:latest'

sh 'sudo docker push 10.133.208.187:5000/optimus-oms:latest'

}

dir ('optimus-product') {

sh 'sudo cp /opt/optimus/optimus-product/Dockerfile .'

sh 'sudo docker build -t optimus-product .'

sh 'sudo docker tag optimus-product:latest 10.133.208.187:5000/optimus-product:latest'

sh 'sudo docker push 10.133.208.187:5000/optimus-product:latest'

}

dir ('optimus-service') {

sh 'sudo cp /opt/optimus/optimus-service/Dockerfile .'

sh 'sudo docker build -t optimus-service .'

sh 'sudo docker tag optimus-service:latest 10.133.208.187:5000/optimus-service:latest'

sh 'sudo docker push 10.133.208.187:5000/optimus-service:latest'

}

dir ('optimus-wfe') {

sh 'sudo cp /opt/optimus/optimus-wfe/Dockerfile .'

sh 'sudo docker build -t optimus-wfe .'

sh 'sudo docker tag optimus-wfe:latest 10.133.208.187:5000/optimus-wfe:latest'

sh 'sudo docker push 10.133.208.187:5000/optimus-wfe:latest'

}

dir ('optimus-batch') {

sh 'sudo cp /opt/optimus/optimus-batch/Dockerfile .'

sh 'sudo docker build -t optimus-batch .'

sh 'sudo docker tag optimus-batch:latest 10.133.208.187:5000/optimus-batch:latest'

sh 'sudo docker push 10.133.208.187:5000/optimus-batch:latest'

}

dir ('optimus-serviceinventory') {

sh 'sudo cp /opt/optimus/optimus-serviceinventory/Dockerfile .'

sh 'sudo docker build -t optimus-serviceinventory .'

sh 'sudo docker tag optimus-serviceinventory:latest 10.133.208.187:5000/optimus-serviceinventory:latest'

sh 'sudo docker push 10.133.208.187:5000/optimus-serviceinventory:latest'

}

}

// unset proxy

stage('unset proxy'){

sh 'unset http\_proxy'

sh 'unset https\_proxy'

}

// delete deployments

stage('Delete all deployments'){

try {

sh 'sudo docker -H 10.133.208.187 rm -f optimus-wfe'

sh 'sudo docker -H 10.133.208.187 rm -f optimus-service'

sh 'sudo docker -H 10.133.208.187 rm -f optimus-oms'

sh 'sudo docker -H 10.133.208.187 rm -f optimus-customer'

sh 'sudo docker -H 10.133.208.187 rm -f optimus-location'

sh 'sudo docker -H 10.133.208.187 rm -f optimus-notification'

sh 'sudo docker -H 10.133.208.187 rm -f optimus-product'

sh 'sudo docker -H 10.133.208.187 rm -f optimus-auth'

sh 'sudo docker -H 10.133.208.187 rm -f optimus-batch'

} catch (err) {

echo "Error while deleting Deployments"

}

try {

sh 'sudo docker -H 10.133.208.187 rm -f optimus-serviceinventory'

} catch (err) {

echo "Error while deleting Service Inventory"

}

}

//delete images

stage('Delete Local Images'){

try {

sh 'sudo docker -H 10.133.208.187 rmi \$(sudo docker -H 10.133.208.187 images | grep 500 | awk {\'print \$3\'})'

} catch (err) {

echo "Error while deleting Local Images"

}

}

stage ('Run') {

withEnv(['DB\_UNAME=optimus',"DB\_PWD=OPTimus12#\$",'MQ\_HOST=10.133.208.187','MQ\_PORT=5672','MQ\_UNAME=web','MQ\_PWD=tata','AUTH\_URL=http://10.133.208.187/auth/','REDIS\_HOST=10.133.208.187','REDIS\_PORT=6379','REDIS\_UNAME=','REDIS\_PWD=','LDAP\_URL=ldap://10.133.208.116:389','LDAP\_BASE=dc=TCL','LDAP\_UNAME=cn=Manager,dc=TCL','LDAP\_PWD=AzXXb5lqpw1YgiHK5t','KEYCLOAK\_PROXY=','R\_ONNET\_WIRELINE=http://10.133.208.121/new/onnet-wireline/api','R\_NPL\_ONNET\_POP=http://10.133.208.121/new/npl/pop/api','R\_ONNET\_WIRELESS=http://10.133.208.121/new/onnet-wireless/api','R\_OFFNET\_WIRELESS=http://10.133.208.121/new/offnet-wireless/api','R\_PRICING=http://10.133.208.121/new/pricing/api','R\_GVPN\_PRICING=http://10.133.208.121/new/gvpn/pricing/api','R\_GSC\_PRICING=http://10.133.208.121/new/gsc/pricing/api','R\_GVPN\_INT\_PRICING=http://10.133.208.121/new/gvpn/ind-int/pricing/api','R\_GVPN\_IND\_PRICING=http://10.133.208.121/new/gvpn/ind-int/pricing/api','R\_NPL\_PRICING=http://10.133.208.120:2500/new/npl/pricing/api','SFDC\_UNAME=huzefa.tarwala@tatacommunications.com.tcluat',"SFDC\_PWD=7ujm&UJMKmmcdUX7JprK1kwkoyOm8LSXx",'SFDC\_GRANT\_TYPE=password',"SFDC\_AUTH=Basic Og==","SFDC\_CLIENT=3MVG9rKhT8ocoxGkPdSEUBFzU\_VZC0HhknudUIGBqUgoTG6mlBwbhnihq74XgyvhW5.fEOubuJg1t6LVDW04X",'SFDC\_SECRET=4585614469580292015','SFDC\_AUTH\_URL=https://test.salesforce.com/services/oauth2/token','SFDC\_GEN\_HOST=https://cs80.salesforce.com/services/apexrest/','SFDC\_OPT\_CREATE\_PATH=GSC\_REST\_V9\_2/OpportunityService/','SFDC\_PRD\_CREATE\_REQ\_URL=GSC\_REST\_V9\_2/Products\_Services\_Service/','SFDC\_PRD\_DELETE\_URL=DeleteProductService','SFDC\_STAGE\_UPDATE\_URL=GSC\_REST\_V9\_2/OpportunityService/','SFDC\_SITE\_UPDATE\_URL=CreateSiteLocation','SMTP\_HOST=p44relay','SMTP\_UNAME=','SMTP\_PWD=','SMTP\_PORT=25','SYSTEM\_PROXY\_HOST=10.133.12.181','SYSTEM\_PROXY\_PORT=80','BUILD\_VERSION=Hydrogen1.0','APP\_HOST=http://10.133.208.187','APP\_ENV=DEV','KEYCLOAK\_UNAME=admin','KEYCLOAK\_PWD=admin','R\_GVPN\_CQ\_POP=http://10.133.208.121/new/gvpn/cqrank/api','R\_GVPN\_INT\_PRICING=http://10.133.208.121/new/gvpn/international/pricing/api','R\_GSC\_PRICING=http://10.133.208.121/new/gsc/pricing/api','DOCUSIGN\_URL=https://demo.docusign.net/restapi','DOCUSIGN\_UNAME=support.customer@tatacommunications.com','DOCUSIGN\_PWD=OPTimus12$','DOCUSIGN\_KEY=73580ce7-ee58-4010-9f98-293a12fb9488']) {

sh 'sudo docker -H 10.133.208.187 run -e DB\_URL="jdbc:mysql://10.133.208.187:3306/oms\_uat?zeroDateTimeBehavior=convertToNull&autoReconnect=true&failOverReadOnly=false&maxReconnects=10&useSSL=false" -e CAS\_HOST="optimus-uat.tatacommunications.com" -e APP\_HOST="optimus-uat.tatacommunications.com" -e DB\_UNAME=$DB\_UNAME -e DB\_PWD=$DB\_PWD -e MQ\_HOST=$MQ\_HOST -e MQ\_PORT=$MQ\_PORT -e MQ\_UNAME=$MQ\_UNAME -e MQ\_PWD=$MQ\_PWD -e REDIS\_HOST=$REDIS\_HOST -e REDIS\_PORT=$REDIS\_PORT -e REDIS\_UNAME=$REDIS\_UNAME -e REDIS\_PWD=$REDIS\_PWD -e AUTH\_URL=$AUTH\_URL -e LDAP\_URL=$LDAP\_URL -e LDAP\_BASE=$LDAP\_BASE -e LDAP\_UNAME=$LDAP\_UNAME -e LDAP\_PWD=$LDAP\_PWD -e KEYCLOAK\_PROXY=$KEYCLOAK\_PROXY -e SYSTEM\_PROXY\_HOST=$SYSTEM\_PROXY\_HOST -e SYSTEM\_PROXY\_PORT=$SYSTEM\_PROXY\_PORT -e KEYCLOAK\_UNAME=$KEYCLOAK\_UNAME -e KEYCLOAK\_PWD=$KEYCLOAK\_PWD -e CUSTOMER\_SUPPORT\_EMAIL=$CUSTOMER\_SUPPORT\_EMAIL -d --name optimus-auth -p 6060:6060 -t 10.133.208.187:5000/optimus-auth:latest'

sh 'sudo docker -H 10.133.208.187 run -e DB\_URL="jdbc:mysql://10.133.208.187:3306/product\_catalog\_uat?zeroDateTimeBehavior=convertToNull&autoReconnect=true&failOverReadOnly=false&maxReconnects=10&useSSL=false" -e DB\_UNAME=$DB\_UNAME -e DB\_PWD=$DB\_PWD -e MQ\_HOST=$MQ\_HOST -e MQ\_PORT=$MQ\_PORT -e MQ\_UNAME=$MQ\_UNAME -e MQ\_PWD=$MQ\_PWD -e REDIS\_HOST=$REDIS\_HOST -e REDIS\_PORT=$REDIS\_PORT -e REDIS\_UNAME=$REDIS\_UNAME -e REDIS\_PWD=$REDIS\_PWD -e SYSTEM\_PROXY\_HOST=$SYSTEM\_PROXY\_HOST -e SYSTEM\_PROXY\_PORT=$SYSTEM\_PROXY\_PORT -e KEYCLOAK\_PROXY=$KEYCLOAK\_PROXY -e AUTH\_URL=$AUTH\_URL -d --name optimus-product -p 9091:7070 -t 10.133.208.187:5000/optimus-product:latest'

sh '#sudo docker -H 10.133.208.187 run -e APP\_HOST="optimus-uat.tatacommunications.com" -e DB\_UNAME=$DB\_UNAME -e DB\_PWD=$DB\_PWD -e MQ\_HOST=$MQ\_HOST -e MQ\_PORT=$MQ\_PORT -e MQ\_UNAME=$MQ\_UNAME -e MQ\_PWD=$MQ\_PWD -e REDIS\_HOST=$REDIS\_HOST -e REDIS\_PORT=$REDIS\_PORT -e REDIS\_UNAME=$REDIS\_UNAME -e REDIS\_PWD=$REDIS\_PWD -e KEYCLOAK\_PROXY=$KEYCLOAK\_PROXY -e AUTH\_URL=$AUTH\_URL -d --name optimus-userservice -p 8085:8085 -t 10.133.208.187:5000/optimus-userservice:latest'

sh 'sudo docker -H 10.133.208.187 run -e DB\_URL="jdbc:mysql://10.133.208.187:3306/notification\_db?zeroDateTimeBehavior=convertToNull&autoReconnect=true&failOverReadOnly=false&maxReconnects=10&useSSL=false" -e DB\_UNAME=$DB\_UNAME -e DB\_PWD=$DB\_PWD -e MQ\_HOST=$MQ\_HOST -e MQ\_PORT=$MQ\_PORT -e MQ\_UNAME=$MQ\_UNAME -e MQ\_PWD=$MQ\_PWD -e REDIS\_HOST=$REDIS\_HOST -e REDIS\_PORT=$REDIS\_PORT -e REDIS\_UNAME=$REDIS\_UNAME -e REDIS\_PWD=$REDIS\_PWD -e APP\_ENV=$APP\_ENV -e SMTP\_HOST=$SMTP\_HOST -e SMTP\_UNAME=$SMTP\_UNAME -e SMTP\_PWD=$SMTP\_PWD -e SMTP\_PORT=$SMTP\_PORT -e SYSTEM\_PROXY\_HOST=$SYSTEM\_PROXY\_HOST -e SYSTEM\_PROXY\_PORT=$SYSTEM\_PROXY\_PORT -e KEYCLOAK\_PROXY=$KEYCLOAK\_PROXY -e AUTH\_URL=$AUTH\_URL -e DOCUSIGN\_URL=$DOCUSIGN\_URL -e DOCUSIGN\_UNAME=$DOCUSIGN\_UNAME -e DOCUSIGN\_PWD=$DOCUSIGN\_PWD -e DOCUSIGN\_KEY=$DOCUSIGN\_KEY -d --name optimus-notification -p 8090:8090 -t 10.133.208.187:5000/optimus-notification:latest'

sh 'sudo docker -H 10.133.208.187 run -e DB\_URL="jdbc:mysql://10.133.208.187:3306/location\_uat?zeroDateTimeBehavior=convertToNull&autoReconnect=true&failOverReadOnly=false&maxReconnects=10&useSSL=false" -e DB\_UNAME=$DB\_UNAME -e DB\_PWD=$DB\_PWD -e MQ\_HOST=$MQ\_HOST -e MQ\_PORT=$MQ\_PORT -e MQ\_UNAME=$MQ\_UNAME -e MQ\_PWD=$MQ\_PWD -e REDIS\_HOST=$REDIS\_HOST -e REDIS\_PORT=$REDIS\_PORT -e REDIS\_UNAME=$REDIS\_UNAME -e REDIS\_PWD=$REDIS\_PWD -e SYSTEM\_PROXY\_HOST=$SYSTEM\_PROXY\_HOST -e SYSTEM\_PROXY\_PORT=$SYSTEM\_PROXY\_PORT -e KEYCLOAK\_PROXY=$KEYCLOAK\_PROXY -e AUTH\_URL=$AUTH\_URL -d --name optimus-location -p 7071:7071 -t 10.133.208.187:5000/optimus-location:latest'

sh 'sudo docker -H 10.133.208.187 run -v "/docker/customer/optimus-files:/opt/optimus-files" -e DB\_URL="jdbc:mysql://10.133.208.187:3306/customer\_uat?zeroDateTimeBehavior=convertToNull&autoReconnect=true&failOverReadOnly=false&maxReconnects=10&useSSL=false" -e DB\_UNAME=$DB\_UNAME -e DB\_PWD=$DB\_PWD -e MQ\_HOST=$MQ\_HOST -e MQ\_PORT=$MQ\_PORT -e MQ\_UNAME=$MQ\_UNAME -e MQ\_PWD=$MQ\_PWD -e REDIS\_HOST=$REDIS\_HOST -e REDIS\_PORT=$REDIS\_PORT -e REDIS\_UNAME=$REDIS\_UNAME -e REDIS\_PWD=$REDIS\_PWD -e SYSTEM\_PROXY\_HOST=$SYSTEM\_PROXY\_HOST -e SYSTEM\_PROXY\_PORT=$SYSTEM\_PROXY\_PORT -e KEYCLOAK\_PROXY=$KEYCLOAK\_PROXY -e AUTH\_URL=$AUTH\_URL -d --name optimus-customer -p 7073:7078 -t 10.133.208.187:5000/optimus-customer:latest'

sh 'sudo docker -H 10.133.208.187 run -v "/docker/customer/optimus-files:/opt/optimus-files" -e DB\_URL="jdbc:mysql://10.133.208.187:3306/oms\_uat?zeroDateTimeBehavior=convertToNull&autoReconnect=true&failOverReadOnly=false&maxReconnects=10&useSSL=false" -e DB\_UNAME=$DB\_UNAME -e DB\_PWD=$DB\_PWD -e MQ\_HOST=$MQ\_HOST -e MQ\_PORT=$MQ\_PORT -e MQ\_UNAME=$MQ\_UNAME -e MQ\_PWD=$MQ\_PWD -e REDIS\_HOST=$REDIS\_HOST -e REDIS\_PORT=$REDIS\_PORT -e REDIS\_UNAME=$REDIS\_UNAME -e REDIS\_PWD=$REDIS\_PWD -e R\_PRICING=$R\_PRICING -e SYSTEM\_PROXY\_HOST=$SYSTEM\_PROXY\_HOST -e SYSTEM\_PROXY\_PORT=$SYSTEM\_PROXY\_PORT -e APP\_HOST=$APP\_HOST -e APP\_ENV=$APP\_ENV -e CUSTOMER\_SUPPORT\_EMAIL=$CUSTOMER\_SUPPORT\_EMAIL -e KEYCLOAK\_PROXY=$KEYCLOAK\_PROXY -e AUTH\_URL=$AUTH\_URL -e R\_GVPN\_PRICING=$R\_GVPN\_PRICING -e R\_GVPN\_IND\_PRICING=$R\_GVPN\_IND\_PRICING -e R\_GVPN\_INT\_PRICING=$R\_GVPN\_INT\_PRICING -e R\_NPL\_PRICING=$R\_NPL\_PRICING -e R\_GSC\_PRICING=$R\_GSC\_PRICING -d --name optimus-oms -p 7072:7072 -t 10.133.208.187:5000/optimus-oms:latest'

sh 'sudo docker -H 10.133.208.187 run -e DB\_UNAME=$DB\_UNAME -e DB\_PWD=$DB\_PWD -e MQ\_HOST=$MQ\_HOST -e MQ\_PORT=$MQ\_PORT -e MQ\_UNAME=$MQ\_UNAME -e MQ\_PWD=$MQ\_PWD -e REDIS\_HOST=$REDIS\_HOST -e REDIS\_PORT=$REDIS\_PORT -e REDIS\_UNAME=$REDIS\_UNAME -e REDIS\_PWD=$REDIS\_PWD -e APP\_HOST="optimus-uat.tatacommunications.com" -e SFDC\_UNAME=$SFDC\_UNAME -e SFDC\_PWD=$SFDC\_PWD -e SFDC\_GRANT\_TYPE=$SFDC\_GRANT\_TYPE -e "SFDC\_AUTH=$SFDC\_AUTH" -e SFDC\_CLIENT=$SFDC\_CLIENT -e SFDC\_SECRET=$SFDC\_SECRET -e SFDC\_AUTH\_URL=$SFDC\_AUTH\_URL -e SFDC\_GEN\_HOST=$SFDC\_GEN\_HOST -e SFDC\_OPT\_CREATE\_PATH=$SFDC\_OPT\_CREATE\_PATH -e SFDC\_PRD\_CREATE\_REQ\_URL=$SFDC\_PRD\_CREATE\_REQ\_URL -e SFDC\_PRD\_DELETE\_URL=$SFDC\_PRD\_DELETE\_URL -e SFDC\_STAGE\_UPDATE\_URL=$SFDC\_STAGE\_UPDATE\_URL -e SFDC\_SITE\_UPDATE\_URL=$SFDC\_SITE\_UPDATE\_URL -e SYSTEM\_PROXY\_HOST=$SYSTEM\_PROXY\_HOST -e SYSTEM\_PROXY\_PORT=$SYSTEM\_PROXY\_PORT -e BUILD\_VERSION=$BUILD\_VERSION -e KEYCLOAK\_PROXY=$KEYCLOAK\_PROXY -e AUTH\_URL=$AUTH\_URL -d --name optimus-service -p 7074:7074 -t 10.133.208.187:5000/optimus-service:latest'

sh 'sudo docker -H 10.133.208.187 run -e DB\_URL="jdbc:mysql://10.133.208.187:3306/camunda?zeroDateTimeBehavior=convertToNull&autoReconnect=true&failOverReadOnly=false&maxReconnects=10&useSSL=false" -e DB\_UNAME=$DB\_UNAME -e DB\_PWD=$DB\_PWD -e MQ\_HOST=$MQ\_HOST -e MQ\_PORT=$MQ\_PORT -e MQ\_UNAME=$MQ\_UNAME -e MQ\_PWD=$MQ\_PWD -e REDIS\_HOST=$REDIS\_HOST -e REDIS\_PORT=$REDIS\_PORT -e REDIS\_UNAME=$REDIS\_UNAME -e REDIS\_PWD=$REDIS\_PWD -e R\_ONNET\_WIRELINE=$R\_ONNET\_WIRELINE -e R\_ONNET\_WIRELESS=$R\_ONNET\_WIRELESS -e R\_OFFNET\_WIRELESS=$R\_OFFNET\_WIRELESS -e SYSTEM\_PROXY\_HOST=$SYSTEM\_PROXY\_HOST -e SYSTEM\_PROXY\_PORT=$SYSTEM\_PROXY\_PORT -e KEYCLOAK\_PROXY=$KEYCLOAK\_PROXY -e AUTH\_URL=$AUTH\_URL -e R\_NPL\_ONNET\_POP=$R\_NPL\_ONNET\_POP -e R\_GVPN\_CQ\_POP=$R\_GVPN\_CQ\_POP -d --name optimus-wfe -p 7019:7019 -t 10.133.208.187:5000/optimus-wfe:latest'

sh 'sudo docker -H 10.133.208.187 run -e DB\_URL="jdbc:mysql://10.133.208.187:3306/oms\_uat?zeroDateTimeBehavior=convertToNull&autoReconnect=true&failOverReadOnly=false&maxReconnects=10&useSSL=false" -e DB\_UNAME=$DB\_UNAME -e DB\_PWD=$DB\_PWD -e MQ\_HOST=$MQ\_HOST -e MQ\_PORT=$MQ\_PORT -e MQ\_UNAME=$MQ\_UNAME -e MQ\_PWD=$MQ\_PWD -e REDIS\_HOST=$REDIS\_HOST -e REDIS\_PORT=$REDIS\_PORT -e REDIS\_UNAME=$REDIS\_UNAME -e REDIS\_PWD=$REDIS\_PWD -e SYSTEM\_PROXY\_HOST=$SYSTEM\_PROXY\_HOST -e SYSTEM\_PROXY\_PORT=$SYSTEM\_PROXY\_PORT -e KEYCLOAK\_PROXY=$KEYCLOAK\_PROXY -e AUTH\_URL=$AUTH\_URL -d --name optimus-batch -p 9016:9016 -t 10.133.208.187:5000/optimus-batch:latest'

sh 'sudo docker -H 10.133.208.187 run -e DB\_URL="jdbc:mysql://10.133.208.187:3306/service\_inventory\_uat?zeroDateTimeBehavior=convertToNull&autoReconnect=true&failOverReadOnly=false&maxReconnects=10&useSSL=false" -e DB\_UNAME=$DB\_UNAME -e DB\_PWD=$DB\_PWD -e MQ\_HOST=$MQ\_HOST -e MQ\_PORT=$MQ\_PORT -e MQ\_UNAME=$MQ\_UNAME -e MQ\_PWD=$MQ\_PWD -e REDIS\_HOST=$REDIS\_HOST -e REDIS\_PORT=$REDIS\_PORT -e REDIS\_UNAME=$REDIS\_UNAME -e REDIS\_PWD=$REDIS\_PWD -e SYSTEM\_PROXY\_HOST=$SYSTEM\_PROXY\_HOST -e SYSTEM\_PROXY\_PORT=$SYSTEM\_PROXY\_PORT -e KEYCLOAK\_PROXY=$KEYCLOAK\_PROXY -e AUTH\_URL=$AUTH\_URL -d --name optimus-serviceinventory -p 9018:9018 -t 10.133.208.187:5000/optimus-serviceinventory:latest'

}

}

}

stage('clean workspace'){

cleanWs()

}

}

***Sample Jenkinsfile for UI:***

node {

// Export proxy

stage('set proxy'){

sh 'export http\_proxy="http://10.133.12.181:80"'

sh 'export https\_proxy="http://10.133.12.181:80"'

}

stage('Checkout') {

try {

sh 'mkdir optimus'

} catch (err) {

echo "Error while Creating dir"

}

try {

dir ('optimus') {

git url: 'http://10.133.208.121/gitlab/root/optimus-ui.git', credentialsId: 'daf9c129-a7f0-4ca0-97c5-ed4e7900c652', branch: 'Hydrogen'

//git url: 'http://10.133.208.133/gitlab/root/optimus-ui.git', credentialsId: '167ea1b4-c98b-41b1-9bd0-047282460d7f', branch: 'feature/GSC-Hydrogen'

}

} catch (err) {

echo "Error while checking out files"

}

}

stage('Enabling Permissions') {

sh 'sudo chmod -R og+rw optimus'

sh 'sudo chown -R pmurugan:pmurugan optimus'

}

stage('Build'){

dir ('optimus') {

sh 'sudo cp -a /docker/node\_modules .'

sh 'sudo npm config set strict-ssl false'

sh 'sudo npm config set proxy http://10.133.19.36:80'

sh 'sudo npm config set https-proxy http://10.133.19.36:80'

sh 'sudo ng build --prod --base-href="/optimus/"'

//sh 'sudo ng build --base-href="/optimus/"'

sh 'sudo chown -R pmurugan:pmurugan ../optimus'

sh 'sudo chmod -R go+rx dist/assets/images'

}

}

// unset proxy

stage('unset proxy'){

sh 'unset http\_proxy'

sh 'unset https\_proxy'

}

stage('Image') {

// writeFile file: 'Dockerfile', text: 'FROM nginx\nCOPY nginx.conf /etc/nginx/nginx.conf\nRUN mkdir /usr/share/nginx/html/optimus\nCOPY optimus/dist /usr/share/nginx/html/optimus\nRUN chmod -R go+rx /usr/share/nginx/html/optimus'

writeFile file: 'Dockerfile', text: 'FROM httpd:2.4\nCOPY httpd.conf /usr/local/apache2/conf/httpd.conf\nRUN mkdir /usr/local/apache2/htdocs/optimus\nCOPY optimus/dist /usr/local/apache2/htdocs/optimus/\nRUN chmod -R go+rx /usr/local/apache2/htdocs/optimus'

sh 'sudo cp /docker/httpd.conf .'

sh 'sudo docker build -t optimus-ui .'

sh 'sudo docker tag optimus-ui:latest 10.133.208.187:5000/optimus-ui:latest'

sh 'sudo docker push 10.133.208.187:5000/optimus-ui:latest'

}

// delete deployments

stage('Delete all deployments'){

try {

sh 'sudo docker -H 10.133.208.187 rm -f optimus-ui'

} catch (err) {

echo "Error while deleting Deployments"

}

}

//delete images

stage('Delete Local Images'){

try {

sh 'sudo docker -H 10.133.208.187 rmi \$(sudo docker -H 10.133.208.187 images | grep 500 | grep ui | awk {\'print \$3\'})'

} catch (err) {

echo "Error while deleting Local Images"

}

}

stage ('Run') {

sh 'sudo docker -H 10.133.208.187 run -d --name optimus-ui -p 8095:80 -t 10.133.208.187:5000/optimus-ui:latest'

}

stage('clean workspace'){

cleanWs()

}

}