**Backend**

FROM oraclelinux:7-slim

RUN yum update -y && \

yum install -y oracle-release-el7 && \

yum install -y oracle-nodejs-release-el7 && \

yum install -y oracle-instantclient19.3-basic.x86\_64 && \

yum install -y nodejs && \

yum clean all

WORKDIR /app/

COPY . /app/

RUN npm install

EXPOSE 3600

CMD ["npm", "start"]

#docker run -d --restart unless-stopped --name containerName -p 3500:3500 imageid

#docker build -t imageName:Version .

**Front end**

FROM node:alpine AS builder

WORKDIR /usr/angular-workdir

COPY . .

RUN npm install && \

npm run build

FROM nginx:alpine

RUN rm -rf /usr/share/nginx/html/\*

COPY nginx.conf /etc/nginx/nginx.conf

COPY --from=builder /usr/angular-workdir/dist/\* /usr/share/nginx/html

EXPOSE 4200

**Nginx frontend**

user nginx;

worker\_processes 1;

error\_log /var/log/nginx/error.log warn;

pid /var/run/nginx.pid;

events {

worker\_connections 1024;

}

http {

include /etc/nginx/mime.types;

default\_type application/octet-stream;

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for"';

access\_log /var/log/nginx/access.log main;

server {

listen 4200;

server\_name localhost;

root /usr/share/nginx/html;

index index.html index.htm;

include /etc/nginx/mime.types;

gzip on;

gzip\_min\_length 1000;

gzip\_proxied expired no-cache no-store private auth;

gzip\_types text/plain text/css application/json application/javascript application/x-javascript text/xml application/xml application/xml+rss text/javascript;

location / {

add\_header 'Access-Control-Allow-Credentials' 'true';

try\_files $uri $uri/ /index.html;

}

}

sendfile on;

#tcp\_nopush on;

keepalive\_timeout 65;

#gzip on;

include /etc/nginx/conf.d/\*.conf;

}

***To redirect url through nginx***

user nginx;

worker\_processes auto;

error\_log /var/log/nginx/error.log;

pid /run/nginx.pid;

include /usr/share/nginx/modules/\*.conf;

events {

worker\_connections 1024;

}

http {

log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request" '

'$status $body\_bytes\_sent "$http\_referer" '

'"$http\_user\_agent" "$http\_x\_forwarded\_for"';

access\_log /var/log/nginx/access.log main;

sendfile on;

tcp\_nopush on;

tcp\_nodelay on;

keepalive\_timeout 65;

types\_hash\_max\_size 2048;

include /etc/nginx/mime.types;

default\_type application/octet-stream;

# Settings for a TLS enabled server.

server {

listen 443 ssl http2 default\_server;

listen [::]:443 ssl http2 default\_server;

server\_name localhost;

ssl\_certificate "/etc/pki/nginx/server.crt";

ssl\_certificate\_key "/etc/pki/nginx/private/server.key";

ssl\_session\_cache shared:SSL:1m;

ssl\_session\_timeout 10m;

ssl\_ciphers HIGH:!aNULL:!MD5;

ssl\_prefer\_server\_ciphers on;

location / {

proxy\_pass hostname:3978;

add\_header X-Frame-Options SAMEORIGIN;

add\_header X-XSS-Protection "1; mode=block";

add\_header Strict-Transport-Security "max-age=31536000; includeSubdomains; preload";

add\_header X-Content-Type-Options nosniff;

add\_header Content-Security-Policy "plugin-types 'application/pdf'";

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

location ^~ /url {

rewrite ^/url/(.\*)$ /$1 break;

proxy\_pass hostname:3500;

add\_header X-Frame-Options SAMEORIGIN;

add\_header X-XSS-Protection "1; mode=block";

add\_header Strict-Transport-Security "max-age=31536000; includeSubdomains; preload";

add\_header X-Content-Type-Options nosniff;

add\_header Content-Security-Policy "plugin-types 'application/pdf'";

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

}  
  
**SSL - Certificate**  
  
#openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout mysitename.key -out mysitename.crt

**NGINX-LB**

user nginx*;*

worker\_processes 1*;*

error\_log /var/log/nginx/error.log warn*;*

pid /var/run/nginx.pid*;*

events {

worker\_connections 1024*;*

}

http {

upstream myapp {

server http://localhost:4200/*;*

server http://172.22.127.225:3103/ weight 3*;*

}

server {

listen 2003*;*

location / {

proxy\_pass http://myapp*;*

}

}

}