Vw-angular

Docker:

**FROM** node:9 as builder

*# Install Google Chrome*

**RUN** wget -q -O - https://dl-ssl.google.com/linux/linux\_signing\_key.pub | apt-key add -

**RUN** sh -c 'echo "deb [arch=amd64] http://dl.google.com/linux/chrome/deb/ stable main" >> /etc/apt/sources.list.d/google.list'

**RUN** apt update && apt install -y google-chrome-stable

*# Fixes permission error when install sass compiler*

**RUN** npm config set user 0

**RUN** npm config set unsafe-perm true

**RUN** npm i -g @angular/cli

**WORKDIR** /client

*# Copy and install packages first to take advantage of Docker cache*

**COPY** package.json package-lock.json ./

**RUN** npm set progress=false && npm config set depth 0 && npm cache verify

**RUN** npm install --silent

*# Copy local directory to container working directory*

**COPY** . .

*# To be used for teamcity lint & unit tests*

**RUN** npm run teamlint && npm run teamtest

*# Allow option to not run build. This is useful for cases where we just want to run tests (feature branches)*

ARG **RUN**\_BUILD=true

*# Build for when in production.*

**RUN** if [ "$RUN\_BUILD" = true ]; then ng build --prod; fi

*# Build the runtime container*

**FROM** nginx:alpine

**RUN** apk update && apk add jq

**COPY** --from=builder /client/dist /usr/share/nginx/html

**COPY** nginx/default.conf /etc/nginx/conf.d/default.conf

**WORKDIR** /client

**COPY** nginx/environments/ .

**COPY** nginx/set-env.sh .

**RUN** chmod +x set-env.sh

**CMD** ./set-env.sh; nginx -g "daemon off;"

--

Set-env.sh

#!/bin/sh

NGINX\_PATH=/usr/share/nginx/html/assets/

ENV\_FILE\_NAME=environment.json

transform\_file()

{

FILE=$1

if [ ! -z "$APP\_CONFIG" ]; then #If config variable is set

# Replace values in environment file with values from variable

echo "Replacing config..."

echo $APP\_CONFIG | jq --slurpfile env $FILE '$env[0] \* .' > $ENV\_FILE\_NAME

else

cp $FILE $ENV\_FILE\_NAME

fi

}

copy\_file()

{

cp $ENV\_FILE\_NAME $NGINX\_PATH$ENV\_FILE\_NAME

}

if [ -z "$APP\_ENVIRONMENT" ]; #If environment is not set

then

echo "Using default config INT"

transform\_file environment-int.json

copy\_file

else

if [ "$APP\_ENVIRONMENT" = "int" ]; then

echo "Using config INT"

transform\_file environment-int.json

copy\_file

elif [ "$APP\_ENVIRONMENT" = "qa" ]; then

echo "Using config QA"

transform\_file environment-qa.json

copy\_file

fi

fi

--Bonkkkers Angular

**FROM** node:9 as builder

*# Fixes permission error when install sass compiler (will fix later)*

**RUN** npm config set user 0

**RUN** npm config set unsafe-perm true

**RUN** npm i -g @angular/cli

**COPY** package.json package-lock.json ./

**RUN** npm set progress=false && npm config set depth 0 && npm cache verify

**RUN** npm install --silent && mkdir /portal-client && cp -R ./node\_modules ./portal-client

**WORKDIR** /portal-client

**COPY** . .

**RUN** npm run teamlint && npm run teamtest

*# Build each environemnt for when we have environtmental variables in the environment.ts files*

*# for now just target prod*

**RUN** ng build --prod --build-optimizer --output-path dist/int

*#RUN ng build --sqa --build-optimizer --output-path dist/sqa*

*#RUN ng build --uat --build-optimizer --output-path dist/uat*

*#RUN ng build --prod --build-optimizer --output-path dist/prod*

**FROM** nginx

**RUN** mkdir -p /app/consumerportal/int

**RUN** mkdir -p /app/consumerportal/sqa

**RUN** mkdir -p /app/consumerportal/uat

**RUN** mkdir -p /app/consumerportal/prod

*# Copy custom config file*

**COPY** proxy/nginx.conf /etc/nginx/conf.d/default.conf

*# Copy built source to each environemnt*

*# for now since we don't have multiple environments we will just copy int to the other envs*

*# Don't do symlink as that's not how it will be once other envs are built*

**COPY** --from=builder /portal-client/dist/int /app/consumerportal/int

**COPY** --from=builder /portal-client/dist/int /app/consumerportal/sqa

**COPY** --from=builder /portal-client/dist/int /app/consumerportal/uat

**COPY** --from=builder /portal-client/dist/int /app/consumerportal/prod

**EXPOSE** 80