#Dockerfile basev3 (building on top of basev1)

FROM docker.io/showpopulous/chatscrum\_img\_base2:basev2

#FROM docker.io/showpopulous/chatscrum\_img\_base3:basev3

######################## HOLD ON ##########################################

# If all you need to do is to run chatscrum,

# then you do not need to build a new image, just run existing image with the following

# docker pull docker.io/showpopulous/chatscrum\_img\_base3:basev3

# NOTE: edit chatscrum\_img:basevi to fit the latest version for your use

# If you don't already have a database, install mysql and create a database called chatscrum

# mkdir /web && cd /web

# copy chatscrum artifacts to /web, untar it there

# mkdir /web/www && cd /web/www

# git init .

# git remote add origin https://gitlab.com/showpopulous/scrumastr.git

# git pull

# cd /web/www/Django/ScrumMaster

# python3.6 manage.py makemigrations

# python3.6 manage.py migrate

# run chatscrum with this: docker run -p 5000:5000 --name chatscrum\_ci local/chatscrum\_img\_base2

# type the ip address of your server into your browser and you should see chatscrum

####### Troubleshooting Tips

# to test mysql connection problems if you are using mysql.connect.django:

# -- from container, import mysql.connector, then conn = mysql.connector.connect(host='xx',database='xx',password='xx')

#

####### If you are rebuilding the chatscrum image

# your machine must be minimum 12G

# create extra swap of minimum 640Mb

# In this dockerfile, suppose your IP is 2.2.2.2, replace ci.chatscrum.com with your ip. You should have 2.2.2.2:5000 below

# pull and run a redis image. For example:

# in the even that you have rebuilt the image, make sure db user linuxjobber can access your database, then run the command below:

# docker run -d -p 5000:5000 -p 5100:5100 -e "DATABASE\_URL=mysql://linuxjobber:8iu7\*IU&@lj\_db" -e "SESSION\_DEFAULTS=database" --link lj\_db --name history\_image

MAINTAINER The CentOS Project <cloud-ops@centos.org>

LABEL Vendor="CentOS" \

License=GPLv2 \

Version=2.4.6-40

RUN mkdir -p /web/

COPY www/ /web/www/

COPY nginx.conf /etc/nginx/

COPY package.json /web

COPY start.sh /start.sh

COPY settings.py /web/www/Django/ScrumMaster/ScrumMaster/settings.py

RUN chmod +x /start.sh

RUN rm -rf /web/Chatscrum-Angular

RUN yum install -y nodejs && yum install -y gcc-c++ make

RUN cd /web && npm install

RUN git config --global user.email "joseph.showunmi@linuxjobber.com"

RUN git config --global user.name "joseph.showunmi"

RUN cd /web && . $HOME/.nvm/nvm.sh && ng new Chatscrum-Angular --routing

RUN pip3.6 install Pillow channels\_redis

RUN . $HOME/.nvm/nvm.sh && yes | cp -r /web/www/Angular/\* /web/Chatscrum-Angular/src

RUN cd /web/Chatscrum-Angular/ && sed -i '26s/.\*/"src\/styles.css","node\_modules\/materialize-css\/dist\/css\/materialize.min.css"/' angular.json;

RUN cd /web/Chatscrum-Angular/ && sed -i '28s/.\*/"scripts": ["node\_modules\/jquery\/dist\/jquery.min.js","node\_modules\/materialize-css\/dist\/js\/materialize.min.js"]/' angular.json; sed -i '19s/.\*/],"types": ["jquery","materialize-css"]/' tsconfig.json;

RUN cd /web/Chatscrum-Angular/ && sed -i 's/127.0.0.1:8000/int.chatscrum.com:81/' src/app/data.service.ts;

#RUN ls /web/Chatsrum-Angular

#RUN cat /web/Chatscrum-Angular/src/app/profile/profile.component.html

RUN cd /web/Chatscrum-Angular && . $HOME/.nvm/nvm.sh && npm install ngx-materialize materialize-css@next ng2-dragula rxjs && ng build --prod --aot

RUN yes | cp -r /web/Chatscrum-Angular/dist/Chatscrum-Angular/assets/ /web/Chatscrum-Angular/dist/Chatscrum-Angular/src/

RUN yes | cp -r /web/Chatscrum-Angular/dist/Chatscrum-Angular/\* /usr/share/nginx/web/Chatscrum-Angular

#RUN cd /web/www/Django/ScrumMaster/ && /bin/python3.6 manage.py makemigrations && /bin/python3.6 manage.py migrate

#RUN cd /web/www/Django/ScrumMaster/ && /bin/python3.6 manage.py runserver 0.0.0.0:5000

RUN touch /etc/uwsgi.d/chatscrum.ini

RUN echo "[uwsgi]" > /etc/uwsgi.d/chatscrum.ini

RUN echo "socket = /run/chatscrumuwsgi/uwsgi.sock" >> /etc/uwsgi.d/chatscrum.ini

RUN echo "chmod-socket = 775" >> /etc/uwsgi.d/chatscrum.ini

RUN echo "chdir = /web/www/Django/ScrumMaster" >> /etc/uwsgi.d/chatscrum.ini

RUN echo "master = true" >> /etc/uwsgi.d/chatscrum.ini

RUN echo "module = ScrumMaster.wsgi:application" >> /etc/uwsgi.d/chatscrum.ini

RUN echo "uid = uwsgi" >> /etc/uwsgi.d/chatscrum.ini

RUN echo "gid = uwsgi" >> /etc/uwsgi.d/chatscrum.ini

RUN echo "processes = 1" >> /etc/uwsgi.d/chatscrum.ini

RUN echo "threads = 1" >> /etc/uwsgi.d/chatscrum.ini

RUN echo "plugins = python36u,logfile" >> /etc/uwsgi.d/chatscrum.ini

RUN mkdir -p /run/chatscrumuwsgi/

RUN chgrp nginx /run/chatscrumuwsgi

RUN chmod 2775 /run/chatscrumuwsgi

RUN touch /run/chatscrumuwsgi/uwsgi.sock

RUN rm -rf /web/www/Django/ScrumMaster/Scrum/migrations/00\*

RUN chgrp -R 0 /start.sh /run /run/chatscrumuwsgi/\* /etc /usr/share/nginx /var/lib /var/log /usr/sbin/uwsgi \

&& chmod -R g=u /start.sh /run /run/chatscrumuwsgi/\* /etc /usr/share/nginx /usr/sbin/uwsgi /var/lib /var/log

#for basev2, container nginx should be running on port 5000 so that host nginx can run on 80

EXPOSE 5000 5100

CMD ["/start.sh"]