**CONTAINERIZE MESSAGEBOARDAPP WITH DOCKER-COMPOSE. RUN DOCKER IMAGES IN DOCKER CONTAINERS.**

**Important files:**

1. **Dockerfile**
2. **docker-compose file**
3. **.env file**

**Observe: use the postgres database, postgres user, postgres password in .env file to run the django migrate jobs. Or set up database access with authorization GRANTED TO ALL in the db section of the docker-compose yaml file.**

**CLEANUP DOCKER IMAGES/PROCESSES/VOLUMES FIRST.**

PS C:\data\minikubetests\MessageBoardApp> **docker system prune**

PS C:\data\minikubetests\MessageBoardApp> **docker system prune -a**

PS C:\data\minikubetests\MessageBoardApp> **docker image prune**

PS C:\data\minikubetests\MessageBoardApp> **docker ps -a**

PS C:\data\minikubetests\MessageBoardApp> **docker rm $(docker ps -a -f status=exited -q)**

PS C:\data\minikubetests\MessageBoardApp> **docker stop $(docker ps -a -q)**

PS C:\data\minikubetests\MessageBoardApp> **docker volume ls**

PS C:\data\minikubetests\MessageBoardApp> **docker volume prune**

PS C:\data\minikubetests\MessageBoardApp> **docker volume ls -f dangling=true**

PS C:\data\minikubetests\MessageBoardApp>

PS C:\data\minikubetests\MessageBoardApp> **docker images -a**

**BUILD/RUN DOCKER IMAGES (WEB, POSTGRES, NGINX) WITH DOCKER-COMPOSE**

<https://dev.to/aduranil/10-docker-compose-and-docker-commands-that-are-useful-for-active-development-22f9>

PS C:\data\minikubetests\MessageBoardApp> **docker-compose build (builds the docker runnable images)**

**Before you run the docker-compose build command, make sure to use the correct docker-compose.yaml file and the correct .env file.**

**The command docker-compose build will create various images. You will need these 3 images to deploy them later to a kubernetes cluster with the docker-stack command:**

**messageboardapp\_nginx latest**

**messageboardapp\_web latest**

**postgres 12.0-alpine**

PS C:\data\minikubeTests\MessageBoardApp\.kompose-convert2> **docker-compose build**

db uses an image, skipping

Building web

Step 1/9 : FROM python:3.7

---> 11c6e5fd966a

Step 2/9 : WORKDIR /code

---> Running in 0686db1b59f0

Removing intermediate container 0686db1b59f0

---> bdf5a63ace69

Step 3/9 : ENV PYTHONDONTWRITEBYTECODE 1

---> Running in c0fcda7c3a0a

Removing intermediate container c0fcda7c3a0a

---> 30822e5fbdd4

Step 4/9 : ENV PYTHONUNBUFFERED 1

---> Running in 123c15018dd6

Removing intermediate container 123c15018dd6

---> ce79b70b46dd

Step 5/9 : RUN pip install --upgrade pip

---> Running in f751292b06cd

Requirement already up-to-date: pip in /usr/local/lib/python3.7/site-packages (20.2.3)

Removing intermediate container f751292b06cd

---> d2d56e36b891

Step 6/9 : RUN pip install pipenv

---> Running in aee8cfe593b9

Collecting pipenv

Downloading pipenv-2020.8.13-py2.py3-none-any.whl (3.9 MB)

Collecting virtualenv

Downloading virtualenv-20.0.31-py2.py3-none-any.whl (4.9 MB)

Collecting virtualenv-clone>=0.2.5

Downloading virtualenv\_clone-0.5.4-py2.py3-none-any.whl (6.6 kB)

Collecting certifi

Downloading certifi-2020.6.20-py2.py3-none-any.whl (156 kB)

Requirement already satisfied: pip>=18.0 in /usr/local/lib/python3.7/site-packages (from pipenv) (20.2.3)

Collecting distlib<1,>=0.3.1

Downloading distlib-0.3.1-py2.py3-none-any.whl (335 kB)

Collecting filelock<4,>=3.0.0

Downloading filelock-3.0.12-py3-none-any.whl (7.6 kB)

Collecting appdirs<2,>=1.4.3

Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)

Collecting importlib-metadata<2,>=0.12; python\_version < "3.8"

Downloading importlib\_metadata-1.7.0-py2.py3-none-any.whl (31 kB)

Collecting six<2,>=1.9.0

Downloading six-1.15.0-py2.py3-none-any.whl (10 kB)

Collecting zipp>=0.5

Downloading zipp-3.1.0-py3-none-any.whl (4.9 kB)

Installing collected packages: distlib, filelock, appdirs, zipp, importlib-metadata, six, virtualenv, virtualenv-clone, certifi, pipenv

Successfully installed appdirs-1.4.4 certifi-2020.6.20 distlib-0.3.1 filelock-3.0.12 importlib-metadata-1.7.0 pipenv-2020.8.13 six-1.15.0 virtualenv-20.0.31 virtualenv-clone-0.5.4 zipp-3.1.0

Removing intermediate container aee8cfe593b9

---> a966922f1b0d

Step 7/9 : COPY Pipfile Pipfile.lock /code/

---> da31b97b0b28

Step 8/9 : RUN pipenv install --system

---> Running in 57ddcd8413ce

Installing dependencies from Pipfile.lock (bbff23)…

Removing intermediate container 57ddcd8413ce

Successfully built e310ec692606

**Successfully tagged messageboardapp\_web:latest**

Building nginx

Step 1/3 : FROM nginx:1.19.0-alpine

---> 7d0cdcc60a96

Step 2/3 : RUN rm /etc/nginx/conf.d/default.conf

---> Running in c0c04807d3e1

Removing intermediate container c0c04807d3e1

---> 64d24bd1f158

Step 3/3 : COPY nginx.conf /etc/nginx/conf.d

---> fc4bd340a5ff

Successfully built fc4bd340a5ff

**Successfully tagged messageboardapp\_nginx:latest**

PS C:\data\minikubeTests\MessageBoardApp> docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

**messageboardapp\_nginx latest**  fc4bd340a5ff 57 minutes ago 21.3MB

**messageboardapp\_web latest** e310ec692606 57 minutes ago 959MB

python 3.7 11c6e5fd966a 7 days ago 876MB

monicamarshall/helloproject\_web latest c4004f86e8f2 3 weeks ago 946MB

**nginx 1.19.0-alpine**  7d0cdcc60a96 3 months ago 21.3MB

kubernetesui/dashboard v2.0.0 8b32422733b3 4 months ago 222MB

kubernetesui/metrics-scraper v1.0.4 86262685d9ab 5 months ago 36.9MB

docker/desktop-storage-provisioner v1.1 e704287ce753 5 months ago 41.8MB

docker/desktop-vpnkit-controller v1.0 79da37e5a3aa 6 months ago 36.6MB

docker/desktop-kubernetes kubernetes-v1.16.5-cni-v0.7.5-critools-v1.15.0 a86647f0b376 8 months ago 279MB

k8s.gcr.io/kube-controller-manager v1.16.5 441835dd2301 8 months ago 151MB

k8s.gcr.io/kube-apiserver v1.16.5 fc838b21afbb 8 months ago 159MB

k8s.gcr.io/kube-scheduler v1.16.5 b4d073a9efda 8 months ago 83.5MB

k8s.gcr.io/kube-proxy v1.16.5 0ee1b8a3ebe0 8 months ago 82.7MB

docker/kube-compose-controller v0.4.25-alpha1 129151cdf35f 10 months ago 35.6MB

docker/kube-compose-api-server v0.4.25-alpha1 989749268895 10 months ago 50.7MB

docker/kube-compose-installer v0.4.25-alpha1 2a71ac5a1359 10 months ago 42.3MB

**postgres 12.0-alpine** 5b681acb1cfc 11 months ago 72.8MB

k8s.gcr.io/etcd 3.3.15-0 b2756210eeab 12 months ago 247MB

k8s.gcr.io/coredns 1.6.2 bf261d157914 13 months ago 44.1MB

kubernetesui/metrics-scraper v1.0.0 44390ebe2b73 16 months ago 36.8MB

k8s.gcr.io/pause 3.1 da86e6ba6ca1 2 years ago 742kB

PS C:\data\minikubetests\MessageBoardApp> **docker-compose up (runs the docker images in undetached mode)**

PS C:\data\minikubetests\MessageBoardApp> **docker-compose up –d –build (builds the images and runs them in detached mode)**

PS C:\data\minikubetests\MessageBoardApp> **docker-compose up -d (runs the images in detached mode without building them)**

PS C:\data\minikubetests\MessageBoardApp> **docker-compose down –v (brings images down and also removes volumes too)**

PS C:\data\minikubetests\MessageBoardApp> **docker-compose logs –f (prints logs to console)**

**SHELL INTO THE DOCKER CONTAINER TO RUN PYTHON MANAGE.PY <ACTION>**

PS C:\data\minikubetests\MessageBoardApp> **docker-compose exec web python manage.py collectstatic**

PS C:\data\minikubetests\MessageBoardApp> **docker-compose exec web python manage.py migrate**

PS C:\data\minikubetests\MessageBoardApp> **docker-compose exec web python manage.py migrate --run-syncdb**

PS C:\data\minikubetests\MessageBoardApp> **docker-compose exec web python manage.py createsuperuser**

Username (leave blank to use 'root'): monica

Email address: monica@gmail.com

Password (again):

….

**IF YOU DON’T HAVE ACCESS TO THE DOCKER SERVICE, in this case <web>, use docker ps to find the container id, then use the container id to shell into the container:**

PS C:\data\minikubeTests\MessageBoardApp> **docker ps**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

1acc0f1bb4ea messageboardapp2\_nginx "/docker-entrypoint.…" 33 minutes ago Up 33 minutes 0.0.0.0:1338->80/tcp messageboardapp2\_nginx\_1

f024b4653ec7 **messageboardapp2\_web** "gunicorn messageboa…" 33 minutes ago Up 33 minutes 8000/tcp messageboardapp2\_web\_1

27ddc24c5b83 postgres:11 "docker-entrypoint.s…" 33 minutes ago Up 33 minutes 5432/tcp messageboardapp2\_db\_1

…

PS C:\data\minikubeTests\MessageBoardApp> **docker exec -it f024b4653ec7 bash**

root@f024b4653ec7:/code# python

Python 3.7.9 (default, Sep 10 2020, 17:09:36)

[GCC 8.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> quit()

root@f024b4653ec7:/code# exit

exit

root@f024b4653ec7:/code# **python manage.py collectstatic**

root@f024b4653ec7:/code# **python manage.py makemigrations**

root@f024b4653ec7:/code# **python manage.py migrate**

root@f024b4653ec7:/code# **python manage.py migrate –run-syncdb**

root@f024b4653ec7:/code# **python manage.py createsuperuser**

**PUSH DOCKER IMAGES TO DOCKERHUB FOR public USE (optional step)**

**USE CONTAINER ID (docker commit, docker push)** Run docker ps –a to find the CONTAINER ID

1. Log into Dockerhub (create an account on dockerhub if you don’t have it)
2. Run docker ps –a to retrieve the container id.
3. Run **docker commit** command
4. Run **docker push** command
5. Make sure to follow the dockerhub docker images naming conventions: username/imagename:versionNumber. Image name may not contain underscores.

**USE IMAGE ID (docker tag, docker push)**

1. Run docker images –a to find the IMAGE ID
2. Log into Dockerhub (create an account on dockerhub if you don’t have it)
3. Run **docker tag** command
4. Run **docker push** command
5. Make sure to follow the dockerhub docker images naming conventions: username/imagename:versionNumber. Image name may not contain underscores.

**IMAGE ID (docker tag, docker push)**

PS C:\data\minikubeTests\MessageBoardApp> **docker images -a**

REPOSITORY TAG IMAGE ID CREATED SIZE

<none> <none> **b1a708c9bdd5** 2 days ago 958MB

messageboardapp\_web latest 91f1bccc9fd7 2 days ago 958MB

<none> <none> **f45708c07ddd** 2 days ago 21.3MB

messageboardapp\_nginx latest 322ea88a3adc 2 days ago 21.3MB

PS C:\data\minikubeTests\MessageBoardApp> **docker tag 91f1bccc9fdmonicamarshall/messageboardapp-web:v1.0**

PS C:\data\minikubeTests\MessageBoardApp> **docker push monicamarshall/messageboardapp-web:v1.0**

The push refers to repository [docker.io/monicamarshall/messageboardapp-web]

cb2d17c547b6: Pushed 97f008807a81: Pushed e396e39bb994: Pushed a57e42ced3a5: Pushed 55efe7d2369d: Pushed 712649607894: Pushed a92828556ea3: Mounted from library/python 0843bbf2ef8c: Mounted from library/python a3080e907f05: Mounted from library/python 0fb2e27dc3b8: Mounted from library/python a995c5106335: Mounted from library/python 17bdf5e22660: Mounted from library/python d37096232ed8: Mounted from library/python 6add0d2b5482: Mounted from library/python 4ef54afed780: Mounted from library/python v1.0: digest: sha256:d985980a8f362d4f73605c8edce0482d19b185417886b1607ebb5327d5acfc5b size: 3476

PS C:\data\minikubeTests\MessageBoardApp> **docker tag 322ea88a3adc monicamarshall/messageboardapp-nginx:v1.0**

PS C:\data\minikubeTests\MessageBoardApp> **docker push monicamarshall/messageboardapp-nginx:v1.0**  The push refers to repository [docker.io/monicamarshall/messageboardapp-nginx]

21fcab763b9a: Pushed 8264f53ade92: Pushed a181cbf898a0: Pushed 570fc47f2558: Pushed 5d17421f1571: Pushed 7bb2a9d37337: Mounted from library/nginx 3e207b409db3: Mounted from library/nginx v1.0: digest: sha256:192be263cd1df0a071f4259ef6fa82be234de920d142a128092a546436bb76fa size: 1774

**USE CONTAINER ID (docker commit, docker push)**

PS C:\data\minikubetests\MessageBoardApp> **docker ps -a**

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

7e9006ff9497 nginx "/docker-entrypoint.…" 3 minutes ago Up 3 minutes 0.0.0.0:1338->80/tcp messageboardapp\_nginx\_1

81115c5d1f1b **messageboardapp-web:latest** "gunicorn messageboa…" 3 minutes ago Up 3 minutes 8000/tcp messageboardapp\_web\_1

07d6d99a56e3 postgres:11 "docker-entrypoint.s…" 3 minutes ago Up 3 minutes 5432/tcp messageboardapp\_db\_1

14ed61afe245 86262685d9ab "/metrics-sidecar" 2 hours ago Up 2 hours k8s\_dashboard-metrics-scraper\_dashboard-metrics-scraper-c79c65bb7-zf7sq\_kubernetes-dashboard\_e5541675-5984-494e-821a-b61520b2f0d9\_5

a6c812e8d456 44390ebe2b73 "/metrics-sidecar" 2 hours ago Up 2 hours k8s\_kubernetes-metrics-scraper\_kubernetes-metrics-scraper-6b97c6d857-m4trk\_kubernetes-dashboard\_063ee72a-698b-4a52-834d-eb9dbc144ae6\_5

ashboard\_857e1be1-0663-4089-bcd5-3dfac972be2c\_6

PS C:\data\minikubetests\MessageBoardApp> **docker commit 81115c5d1f1b monicamarshall/messageboardapp-web:latest**

PS C:\data\minikubetests\MessageBoardApp> **docker push monicamarshall/messageboardapp-web:latest**

**USE DOCKER STACK TO DEPLOY THE IMAGES PREVIOUSLY CREATED TO THE KUBERNETES CLUSTER. DEPLOY EXISTING LOCAL/PUBLIC IMAGES.**

**Important files:**

1. **Dockerfile**
2. **docker-compose file (Observe: build instruction replaced with image name, local or public). If you don’t have a local or public image available, build it first with the docker-compose command. Instructions above.**
3. **.env file**

PS C:\data\minikubetests\MessageBoardApp> **docker stack deploy -c docker-compose.yml messageboardapp-stack**

PS C:\data\minikubeTests\MessageBoardApp\.docker-stack> docker stack deploy -c docker-compose.yml messageboardapp-stack

service "db": env\_file are ignored

service "web": depends\_on are ignored

service "web": env\_file are ignored

service "nginx": depends\_on are ignored

Waiting for the stack to be stable and running...

nginx: Failed [pod status: 0/1 ready, 0/1 pending, 0/1 failed]

web: Ready [pod status: 1/1 ready, 0/1 pending, 0/1 failed]

db: Ready [pod status: 1/1 ready, 0/1 pending, 0/1 failed]

PS C:\data\minikubeTests\MessageBoardApp\.docker-stack> **kubectl get services**

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

db ClusterIP None <none> 55555/TCP 3m19s

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 28h

nginx ClusterIP None <none> 55555/TCP 3m19s

**nginx-published LoadBalancer 10.109.15.6 localhost 1338:31815/TCP 3m19s**

web ClusterIP None <none> 55555/TCP 3m19s

web-published LoadBalancer 10.103.65.84 localhost 8000:31691/TCP 3m19s

PS C:\data\minikubetests\MessageBoardApp> **kubectl get pods**

PS C:\data\minikubetests\MessageBoardApp> **docker stack remove messageboardapp-stack**

PS C:\data\minikubetests\MessageBoardApp> **kubectl get services**

PS C:\data\minikubetests\MessageBoardApp> **kubectl get pods**

**All the python commands below can be run inside the kubernetes dashboard, instead of running them in the powershell.**

1. **Access the kubernetes dashboard**
2. **Select the Pods view on the left colum**
3. **Select the web Pod.**
4. **Click on the 3 dots at the end of the row (the 3 dots are aligned in a column)**
5. **Select EXEC to shell into the pod**
6. **Enter the python commands directly in the bash shell.**

PS C:\data\minikubetests\MessageBoardApp> **kubectl exec -it web-59766f5cd4-2vtnn python manage.py collectstatic**

PS C:\data\minikubetests\MessageBoardApp> **kubectl exec -it web-59766f5cd4-2vtnn python manage.py createsuperuser**

Username (leave blank to use 'root'): monica

Email address: monica@gmail.com

Password:

PS C:\data\minikubetests\MessageBoardApp> **kubectl exec -it web-59766f5cd4-2vtnn python manage.py migrate**

Operations to perform:

Apply all migrations: admin, auth, contenttypes, sessions

Running migrations:

Applying contenttypes.0001\_initial... OK

Applying auth.0001\_initial... OK

Applying admin.0001\_initial... OK

Applying admin.0002\_logentry\_remove\_auto\_add... OK

Applying admin.0003\_logentry\_add\_action\_flag\_choices... OK

Applying auth.0002\_alter\_permission\_name\_max\_length... OK

Applying auth.0003\_alter\_user\_email\_max\_length... OK

Applying auth.0004\_alter\_user\_username\_opts... OK

Applying auth.0005\_alter\_user\_last\_login\_null... OK

Applying auth.0006\_require\_contenttypes\_0002... OK

Applying auth.0007\_alter\_validators\_add\_error\_messages... OK

Applying auth.0008\_alter\_user\_username\_max\_length... OK

Applying auth.0009\_alter\_user\_last\_name\_max\_length... OK

Applying sessions.0001\_initial... OK

PS C:\data\minikubetests\MessageBoardApp> kubectl exec -it web-59766f5cd4-2vtnn python manage.py migrate --run-syncdb

Error: unknown flag: --run-syncdb

**LOG INTO THE POD SHELL TO SYNC THE DB**

PS C:\data\minikubetests\MessageBoardApp> **kubectl get pods**

NAME READY STATUS RESTARTS AGE

db-0 1/1 Running 0 15m

nginx-68866666f-ztdwt 1/1 Running 1 15m

web-59766f5cd4-2vtnn 1/1 Running 0 15m

PS C:\data\minikubetests\MessageBoardApp> **kubectl exec --stdin --tty web-59766f5cd4-2vtnn -- /bin/bash**

**Or**

PS C:\data\minikubetests\MessageBoardApp> **kubectl exec –it web-59766f5cd4-2vtnn -- /bin/bash**

root@web-59766f5cd4-2vtnn:/code# **ls**

Dockerfile db-deployment.yaml docker-compose.yml manage.py nginx-service.yaml templates

Dockerfile.bk2 db-service.yaml env-configmap.yaml messageboardapp postgres-data-persistentvolumeclaim.yaml web-claim0-persistentvolumeclaim.yaml

Pipfile db.sqlite3 job.yaml nginx posts web-deployment.yaml

Pipfile.lock docker-compose.bk.yml kubedashboard-adminuser.yaml nginx-claim0-persistentvolumeclaim.yaml requirements.txt web-service.yaml

README.md docker-compose.bk2.yml kubedashboard-clusterrolebinding.yaml nginx-deployment.yaml static

root@web-59766f5cd4-2vtnn:/code# **python manage.py migrate --run-syncdb**

Operations to perform:

Synchronize unmigrated apps: messages, posts, staticfiles

Apply all migrations: admin, auth, contenttypes, sessions

Synchronizing apps without migrations:

Creating tables...

**Creating table posts\_post**

Running deferred SQL...

Running migrations:

No migrations to apply.

root@web-59766f5cd4-2vtnn:/code# exit

exit

PS C:\data\minikubetests\MessageBoardApp>

**ACCESS KUBERNETES DASHBOARD:**

1. Create 2 kubernetes dashboard admin files: kubedahsboard-adminuser.yaml and kubedashboard-clusterrolebinding.yaml and apply both files with the command: **kubectl apply –f kubedahsboard-adminuser.yaml** and **kubectl apply –f kubedashboard-clusterrolebinding.yaml.** Content of files listed below.
2. Open powershell
3. Enter kubectl proxy on the command line. The command will start servicing the kubectl dashboard on localhost:8001

PS C:\data\minikubeTests\MessageBoardApp> **kubectl proxy**

Starting to serve on 127.0.0.1:8001

1. Open browser, enter: <http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/overview?namespace=default>

**kubedahsboard-adminuser.yaml**

apiVersion: v1

kind: ServiceAccount

metadata:

name: admin-user

namespace: kubernetes-dashboard

**kubedashboard-clusterrolebinding.yaml**

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: admin-user

roleRef:

apiGroup: rbac.authorization.k8s.io

kind: ClusterRole

name: cluster-admin

subjects:

- kind: ServiceAccount

name: admin-user

namespace: kubernetes-dashboard

**CREATE A REGCRED (DOCKERHUB REGISTRY CREDENTIALS) SECRET TO ALLOW KUBERNETES TO RUN IMAGES STORED IN DOCKERHUB. THIS IS NECESSARY WHEN THE IMAGES USED IN THE KUBERNETES MANIFEST FILES ARE STORED IN DOCKERHUB.**

kubectl create secret docker-registry regcred \

--docker-server=<your registry> \

--docker-username=<your username> \

--docker-password=<your password> \

--docker-email=<your email>

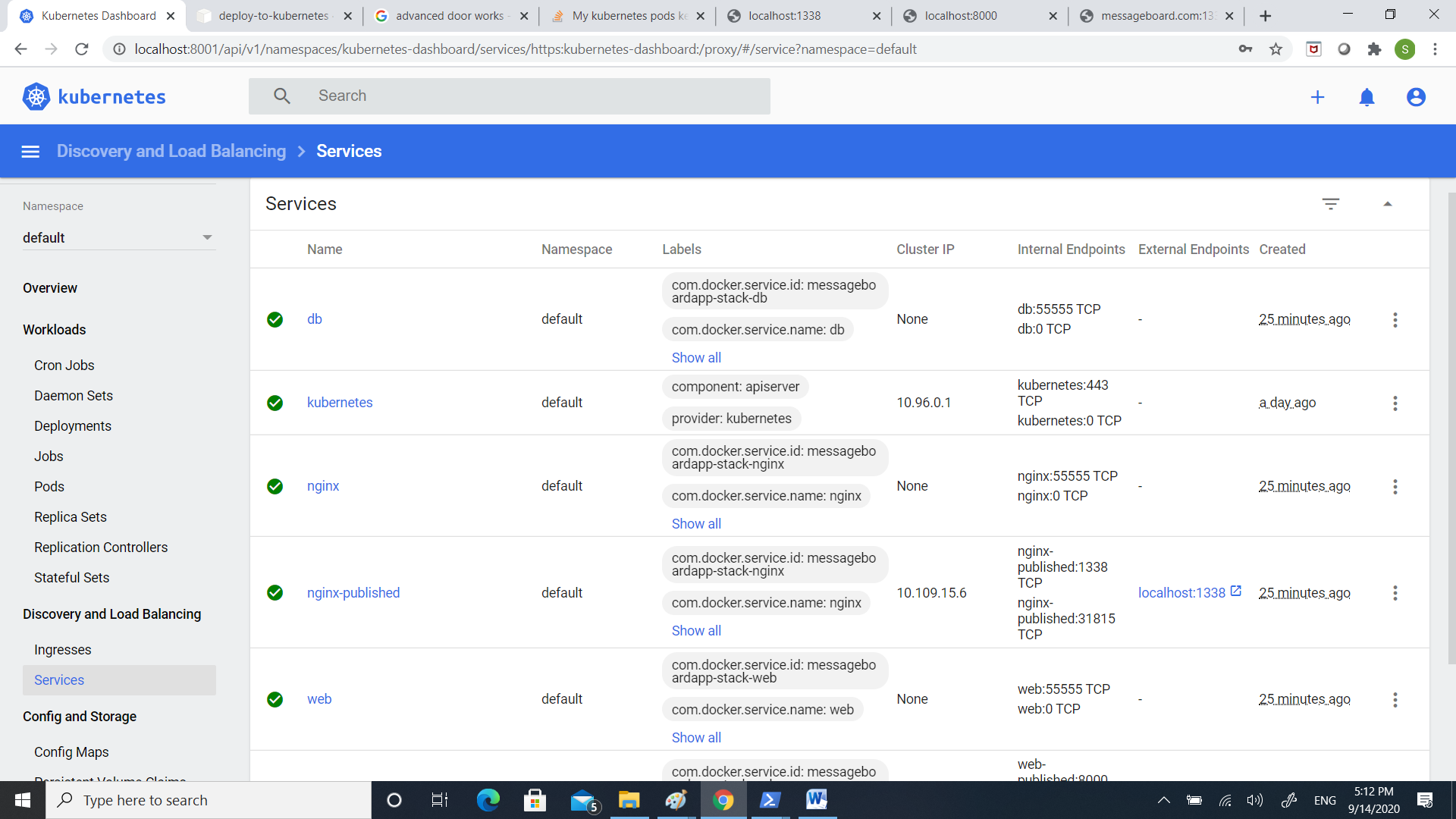
1. Open Powershell

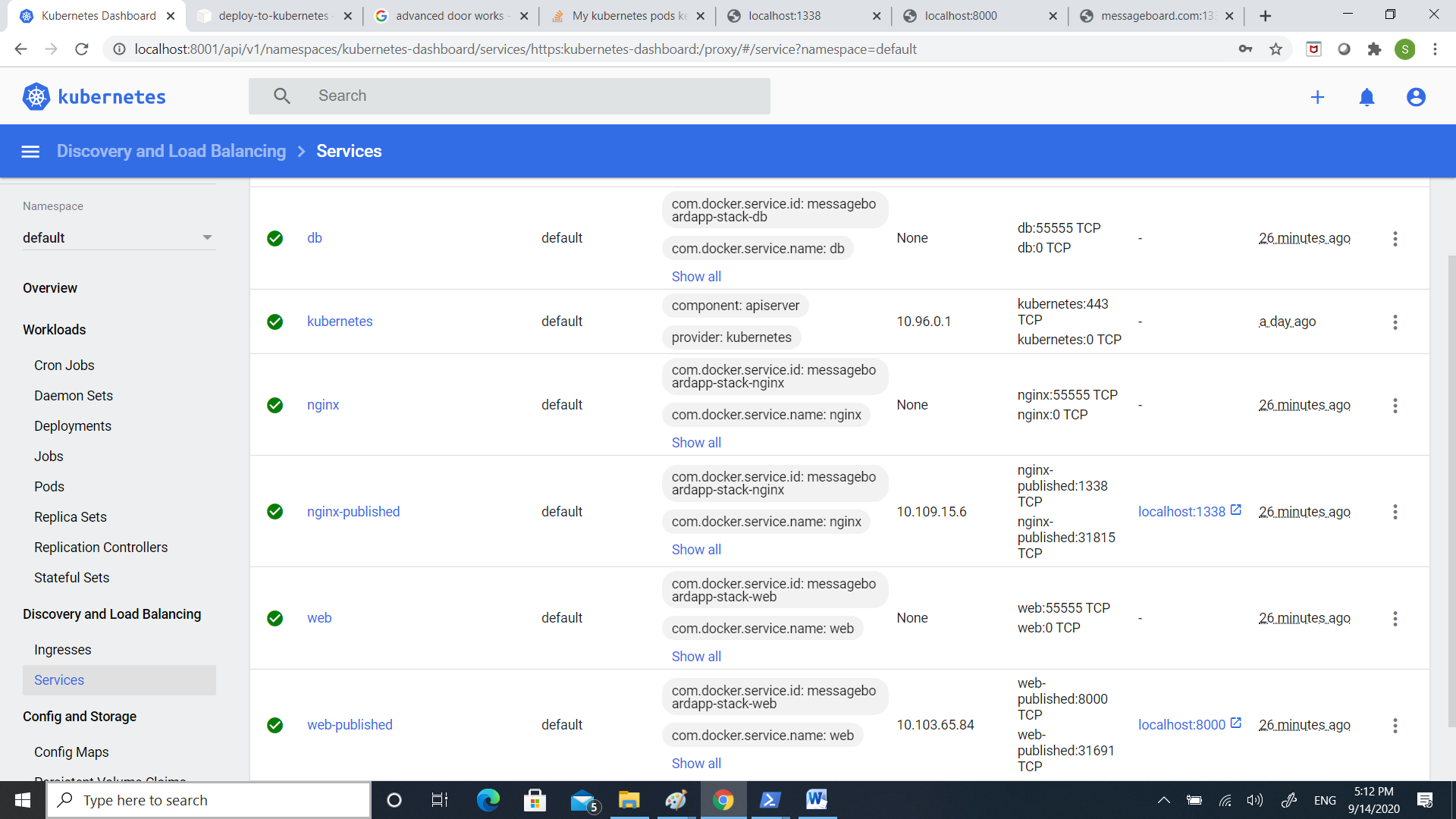
2. Set up account in dockerhub

3. Enter command to store dockerhub credentials in kubernetes:

**kubectl create secret docker-registry regcred** --docker-server=https://index.docker.io/v1/ --docker-username=<Yourdockerhubusername> --docker-password=<yourpassword> [--docker-email=marshallmonica@yahoo.com](mailto:--docker-email=marshallmonica@yahoo.com)

**KUBERNETES DASHBOARD SCREENSHOTS SHOWING SERVICES DEPLOYED:**





**USE KUBECTL COMMAND TO CREATE A DEPLOYMENT OF AN EXISTING PUBLIC IMAGE (DOCKERHUB) AND EXPOSE IT FOR EXTERNAL USE:**

kubectl create deployment nginx-deployment --image=monicamarshall/messageboardapp-nginx:v1.0

kubectl expose deployment nginx-deployment --port=1338 --target-port=80 --type LoadBalancer

kubectl expose deployment nginx-deployment --port=1339 --target-port=80 --type LoadBalancer

kubectl create deployment web-deployment --image=monicamarshall/messageboardapp-web:v1.0

kubectl expose deployment web-deployment --port=80 --target-port=8000 --type NodePort