**BUILD/RUN/DEPLOY DOCKER IMAGES IN A KUBERNETES CLUSTER WITH KUBECTL.**

Use pertinent files in k8s folder. Deploy the web app to a single node kubernetes cluster with kubernetes deployment yaml files. In this scenario 3 different pods will be created. 1 pod runs the web app exposed at port 8000 via the gunicorn wsgi channel. The 8000 port is not accessible from outside the cluster, it is only exposed internally to the nginx container. The web app application is accessible from outside the kubernetes cluster via NodePort 3100. This port is specified in the nginx-deployment file. 1 pod runs the nginx server exposing the web app on port 31000 via the nginx-ingress-controller. 1 pod runs the postgres database on port 5432. This port is only accessible internally to the web application. The postgres database is accessible from outside the kubernetes cluster on NodePort selected: 32000. The 32000 port selection is done in the db service yaml file. All NodePorts are allocated in the kubernetes services yaml files and may be changed. The range of allowable NodePorts is from 30000-32767. Make sure to select a free port within the range.

**KUBERNETES YAML DEPLOYMENT FILES USE IMAGES THAT HAVE ALREADY BEEN CREATED. TO CREATE THESE IMAGES FIRST BY RUNNING DOCKER-COMPOSE BUILD COMMAND USING THE APPROPRIATE DOCKER-COMPOSE YAML FILE. FIND INSTRUCTIONS IN THE DOCUMENT ‘MessageBoardApp\_Docker-compose\_Docker-StackDeployment’ IN THE DOCUMENTATION FOLDER ON GITHUB FOR BUILDING THE IMAGES NEEDED.**

Before you apply the kubernetes yaml files for deployment of the webapp, postgres, and nginx, the nginx ingress controller must be deployed.

Instructions for deploying the nginx ingress controller are in the document ‘nginx-ingress-deploy-command.txt’ in the documentation folder.

**First enable ingress addons**, then run the yaml file below:

PS C:\data\minikubeTests\MessageBoardApp3> **kubectl apply -f** [**https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.30.0/deploy/static/mandatory.yaml**](https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.30.0/deploy/static/mandatory.yaml)

**1. enable ingress using admin powershell/cmd**

**2. apply mandatory.yaml with kubectl apply**

**3. verify the nginx-ingress-controller is running in the 'ingress-nginx' namespace on kubernetes cluster.**

**4. To remove the nginx-ingress controller run kubectl delete -f <filename>**

**PS C:\WINDOWS\system32> minikube addons enable ingress**

**\* The 'ingress' addon is enabled**

**PS C:\WINDOWS\system32>**

**kubectl apply -f** [**https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.30.0/deploy/static/mandatory.yaml**](https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.30.0/deploy/static/mandatory.yaml)

**PS C:\eclipseWorkspace2020-09\MessageBoardApp> kubectl get pods -n ingress-nginx**

**NAME READY STATUS RESTARTS AGE**

**nginx-ingress-controller-7f74f657bd-4kskh 1/1 Running 0 3h45m**

**kubectl delete -f** [**https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.30.0/deploy/static/mandatory.yaml**](https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.30.0/deploy/static/mandatory.yaml)

**USE KUBECTL APPLY –f <yamlFileName> TO DEPLOY THE IMAGES PREVIOUSLY CREATED TO THE KUBERNETES CLUSTER.**

PS C:\data\minikubeTests\MessageBoardApp3> **kubectl apply -f .\db-configMap.yaml**

PS C:\data\minikubeTests\MessageBoardApp3> **kubectl apply -f .\db-credentials.yaml**

PS C:\data\minikubeTests\MessageBoardApp3> **kubectl apply -f .\env-configmap.yaml**

PS C:\data\minikubeTests\MessageBoardApp3> **kubectl apply -f .\db-deployment.yaml**

PS C:\data\minikubeTests\MessageBoardApp3> **kubectl apply -f .\nginx-deployment.yaml**

PS C:\data\minikubeTests\MessageBoardApp3> **kubectl apply -f .\web-deployment.yaml**

PS C:\data\minikubeTests\MessageBoardApp3> **kubectl apply -f .\nginx-ingress.yaml**

**ACCESS THE WEBAPP AT:**

<http://localhost:31000> or <http://messageboard.com:31000>

Check pods, services for port allocation:

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

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

**All the python commands (collectstatic, migrate, migrate –run-syndb, createsuperuser) can be run inside the kubernetes dashboard using the pod’s shell option on the left, 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:**
7. **Python manage.py collectstatic**
8. **Python manage.py migrate**
9. **Python manage.py migrate –run-syndb**
10. **Python manage.py createsuperuser**

**python commands (collectstatic, migrate, , createsuperuser) can be run also using the kubectl exec –it command on the command line. Python migrate –run-syndb will not execute with kubectl exec –it.**

PS C:\data\minikubetests\MessageBoardApp> **kubectl get pods (to find the pod name)**

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>

**CONFIGURE/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 PUBLIC 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

**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**