CAT – 2

CLOUD COMPUTING LAB

TEAM MEMBERS:

NARESH KUMAR R M - 19 34 025

SRIRAM S - 19 34 048

AATHEESWARAN M - 19 34 001

Problem statement:

Customers prefer messaging. Almost all mobile users are familiar with messaging apps such as WhatsApp, Telegram, Slack. Written and conversational communication over those applications is preferred especially by millennials. Banks are also testing using these popular messaging platforms for customer service. Demand for mobile banking is increasing. Citi points out that 91% of users have positive outcomes with mobile banking. 24/7 available chatbots integrated to mobile applications can offer users immediate solutions to their urgent problems that they cannot resolve via the app. Demand for automation increases in banking sector too, and the next form of automation is help desk chatbot in Banks

Goal:

Our aim is to build a chatbot for the replacement of help desk in banks. The chatbot runs on basic machine learning models such as K-NN and NB. Creating a minimalistic web UI for chatbot and connecting the model with the webpage.

After successful integration of the ML model with chatbot, the entire chatbot should be containerized into two parts and then should be deployed in Kube using Microsoft Azure.

Demonstration of Docker Features:

Image building:

- Docker images are built for the following chatbot files into two sections.
- Flask python file - app.py
- Webpage html file and other - index.html, style.css, get_post.js, chat_mod, vector mod.

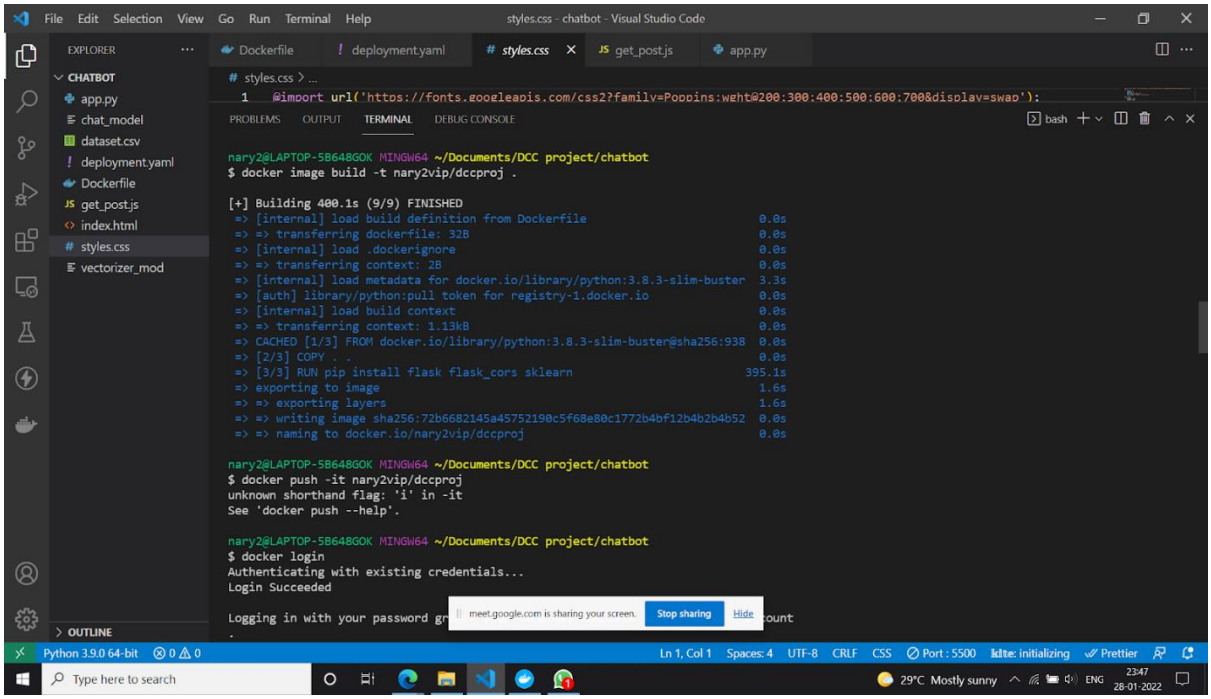
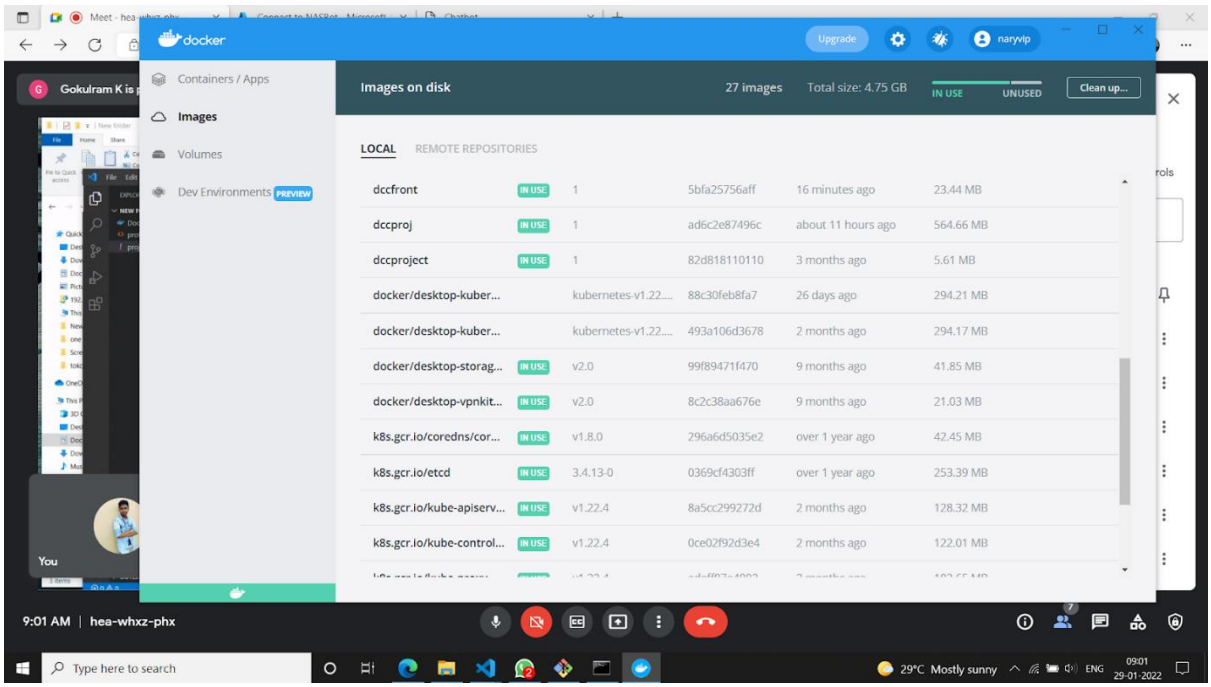


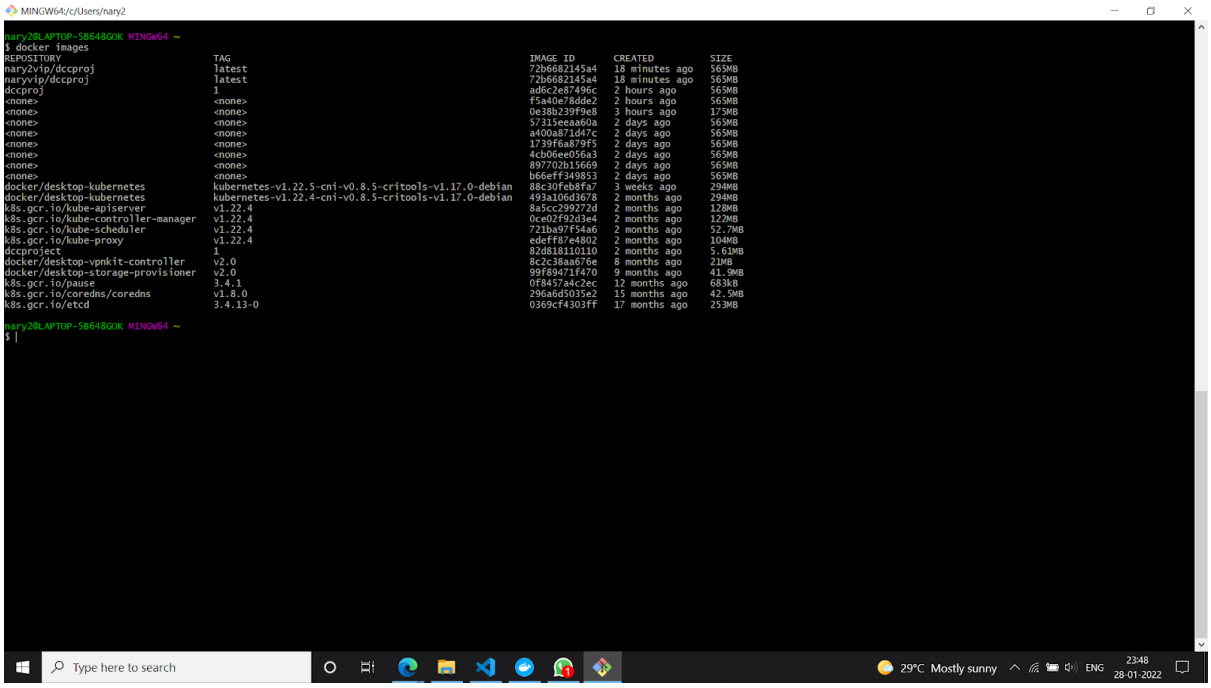
Image caching:

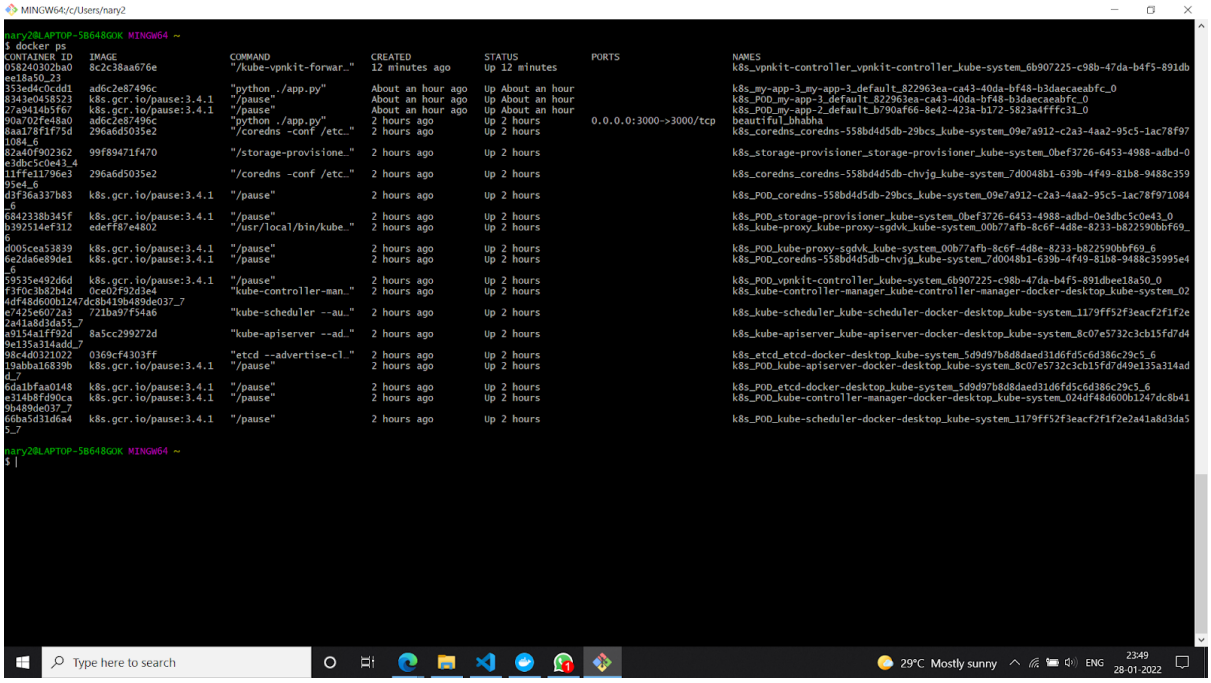


Container Deployment:

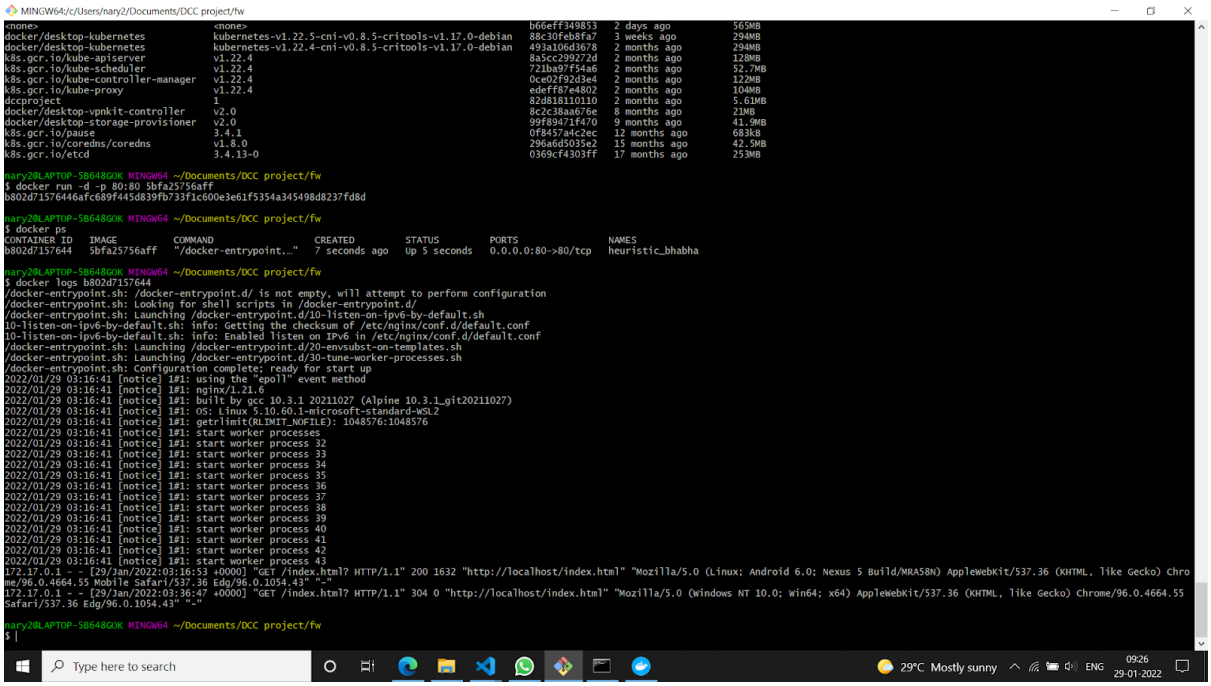
```
sriram@debian11:~/Documents/Cloud-computing/Chatbot$ sudo docker run -p 3000:3000 26bae27
5fb27
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://172.17.0.2:3000/ (Press CTRL+C to quit)
172.17.0.1 - - [28/Jan/2022 15:33:35] "POST / HTTP/1.1" 200 -
172.17.0.1 - - [28/Jan/2022 15:33:57] "POST / HTTP/1.1" 200 -
172.17.0.1 - - [28/Jan/2022 15:34:02] "POST / HTTP/1.1" 200 -
^CResult: ['hello']
Please withdraw your inconvenient words..How can I help you?
Result: ['hi']
Please withdraw your inconvenient words..How can I help you?
Result: ['deposit']
Fill up the details in the slip in the following way: First name of the Depositor, Accoun
t he/she is depositing to, the amount, Signature of the depositor.
  Created wheel for sklearn: filename=sklearn-0.0-py2.py3-none-any.whl size=1
72e9466947358cb37abe159a517c20787301d5995a878759dd333bf03b21b21
  Stored in directory: /root/.cache/pip/wheels/22/0b/40/fd3f795caaa1fb4c6cb73
1e57da95849bfc897
Successfully built sklearn
Installing collected packages: itsdangerous, MarkupSafe, Jinja2, click, Werkz
Six, flask-cors, numpy, scipy, joblib, threadpoolctl, scikit-learn, sklearn
Successfully installed Jinja2-3.0.3 MarkupSafe-2.0.1 Six-1.16.0 Werkzeug-2.0
3 flask-2.0.2 flask-cors-3.0.10 itsdangerous-2.0.1 joblib-1.1.0 numpy-1.22.1
-1.0.2 scipy-1.7.3 sklearn-0.0 threadpoolctl-3.0.0
WARNING: You are using pip version 20.1.1; however, version 21.3.1 is availab
You should consider upgrading via the '/usr/local/bin/python -m pip install
' command.
Removing intermediate container 2f4ffec9bd05
 ---> 05422253d6cc
Step 4/5 : EXPOSE 3000
 ---> Running in 2f891c1ae66f
Removing intermediate container 2f891c1ae66f
 ---> 16e33de5e976
Step 5/5 : cmd ["python", "./app.py"]
 ---> Running in 9b9a6175d175
Removing intermediate container 9b9a6175d175
 ---> 26bae275fb27
Successfully built 26bae275fb27
Successfully tagged my-app:3
```

Container Service Exposure:

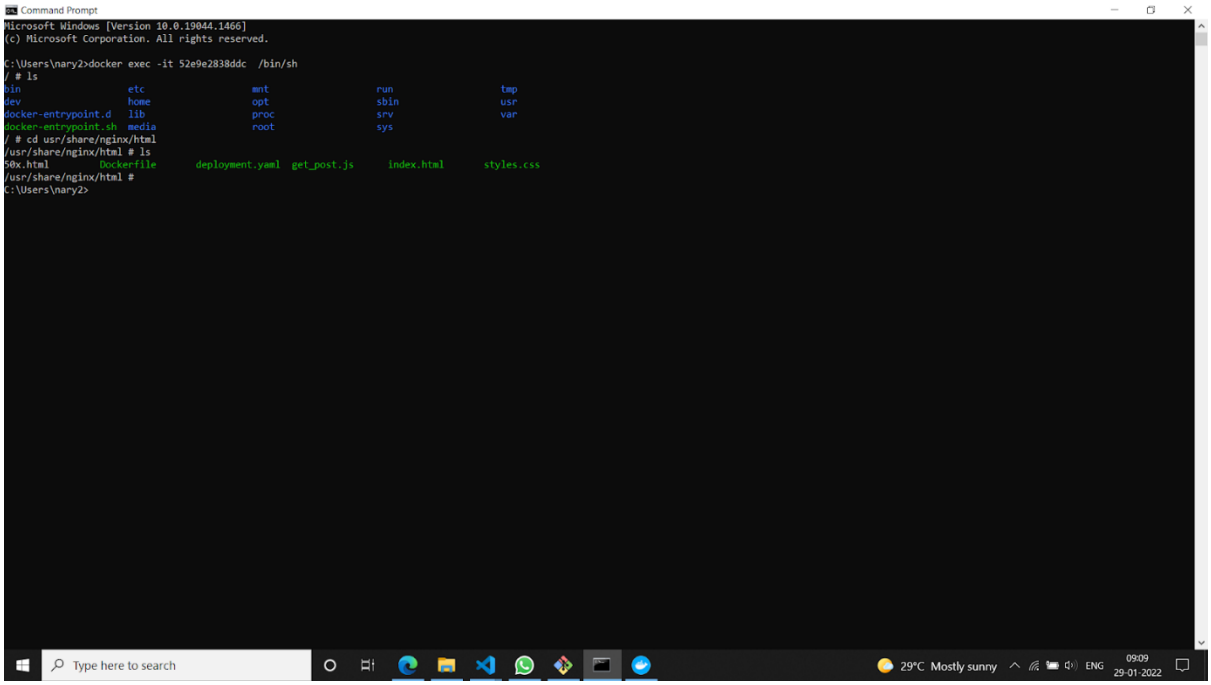




Logs Fetching:



Execute into the running Container:

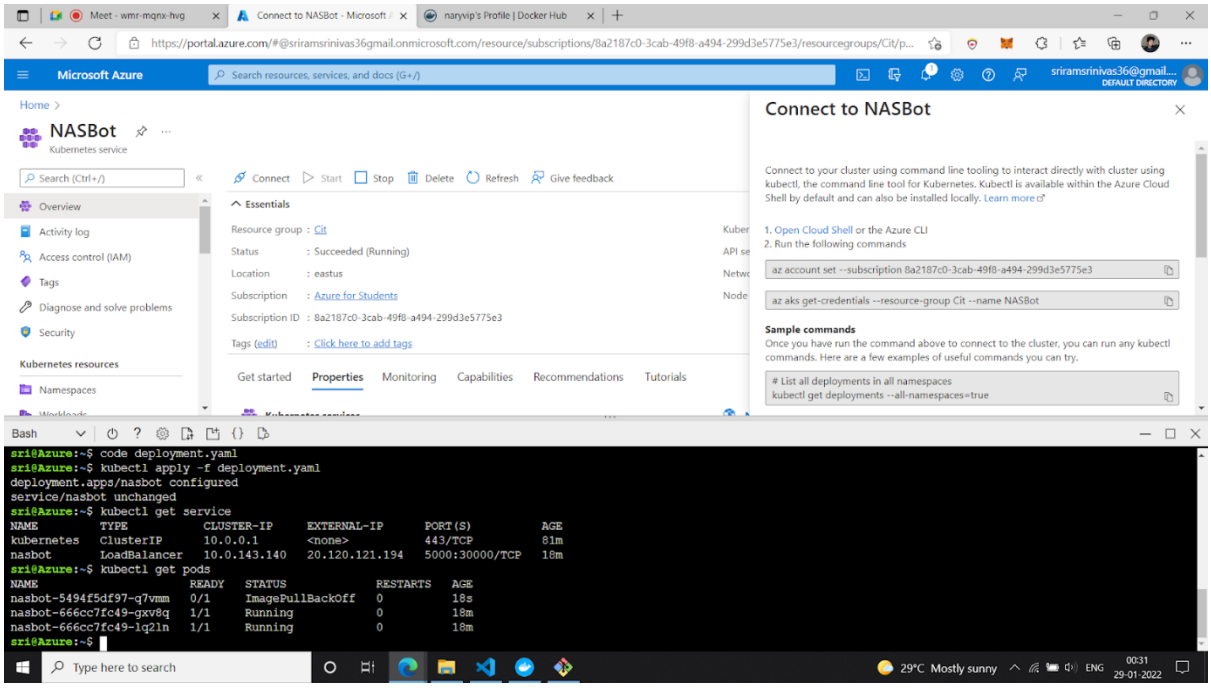


Demonstration of Kubernetes Features:

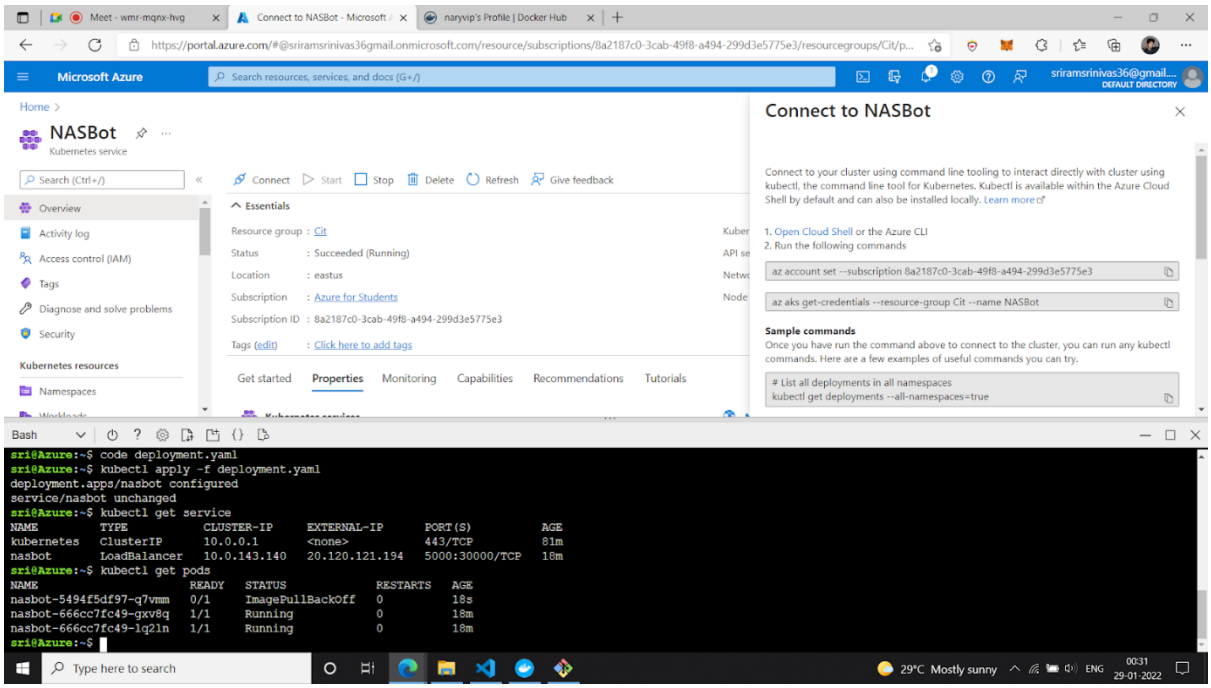
Creating Namespace:

We've been using the default namespace.

Deploying the pod in that namespace:

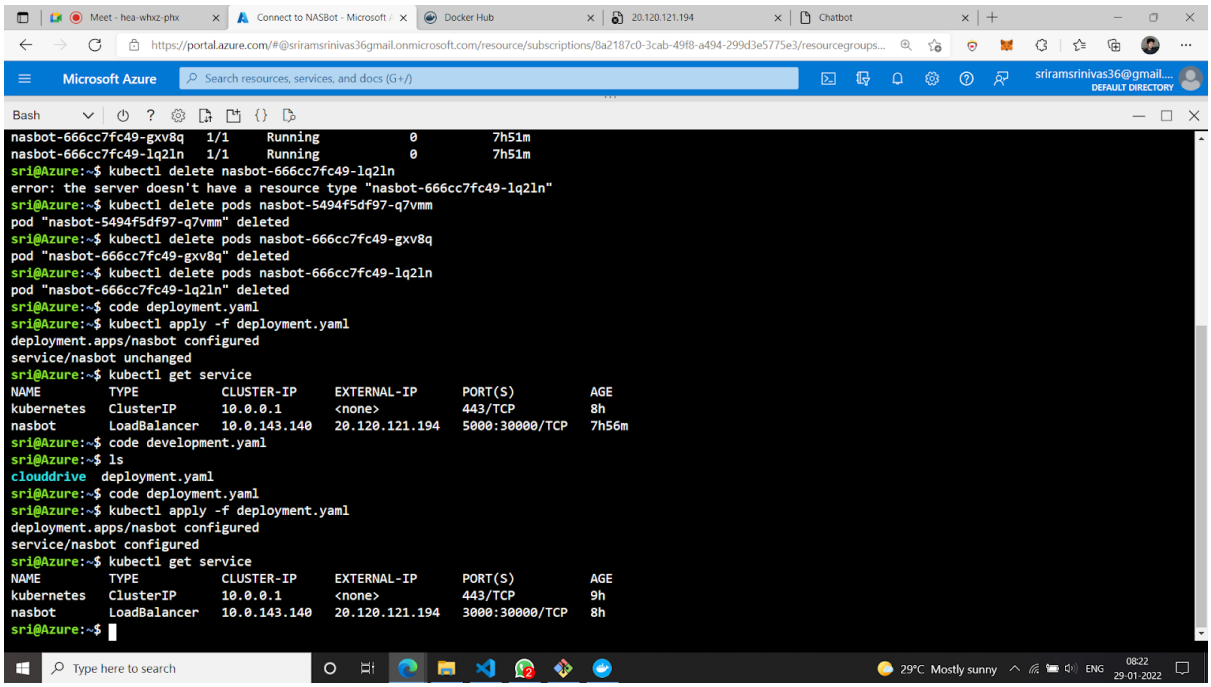


Setting the Replica factor:

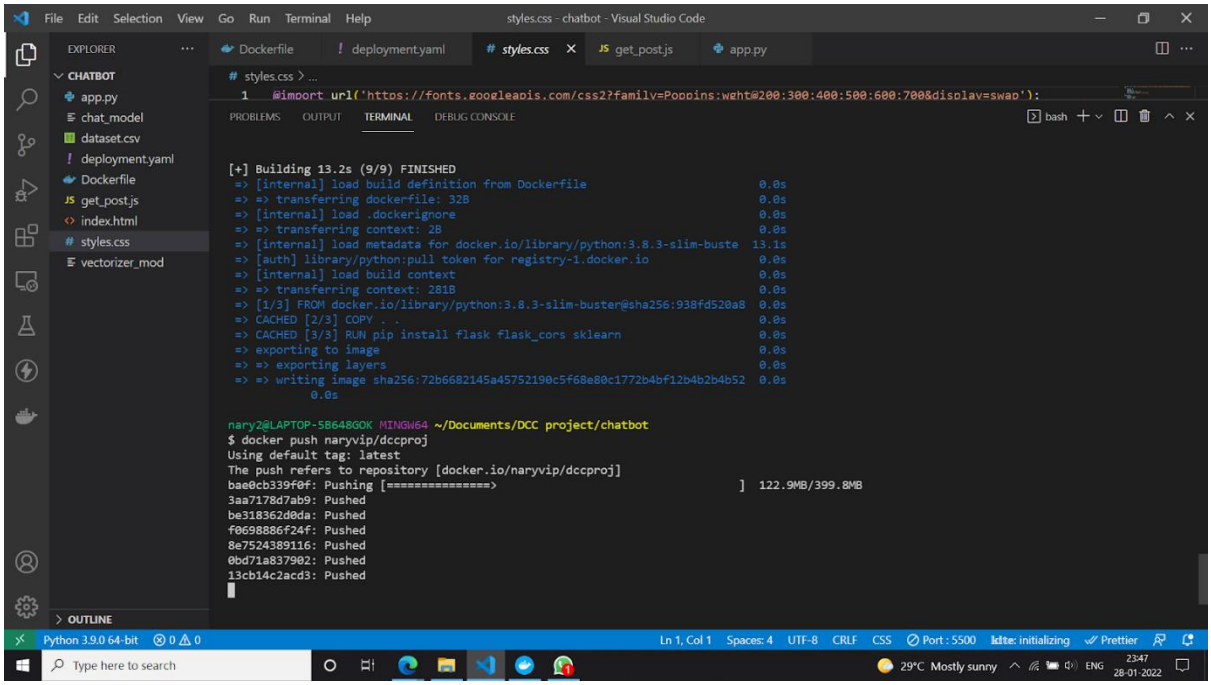


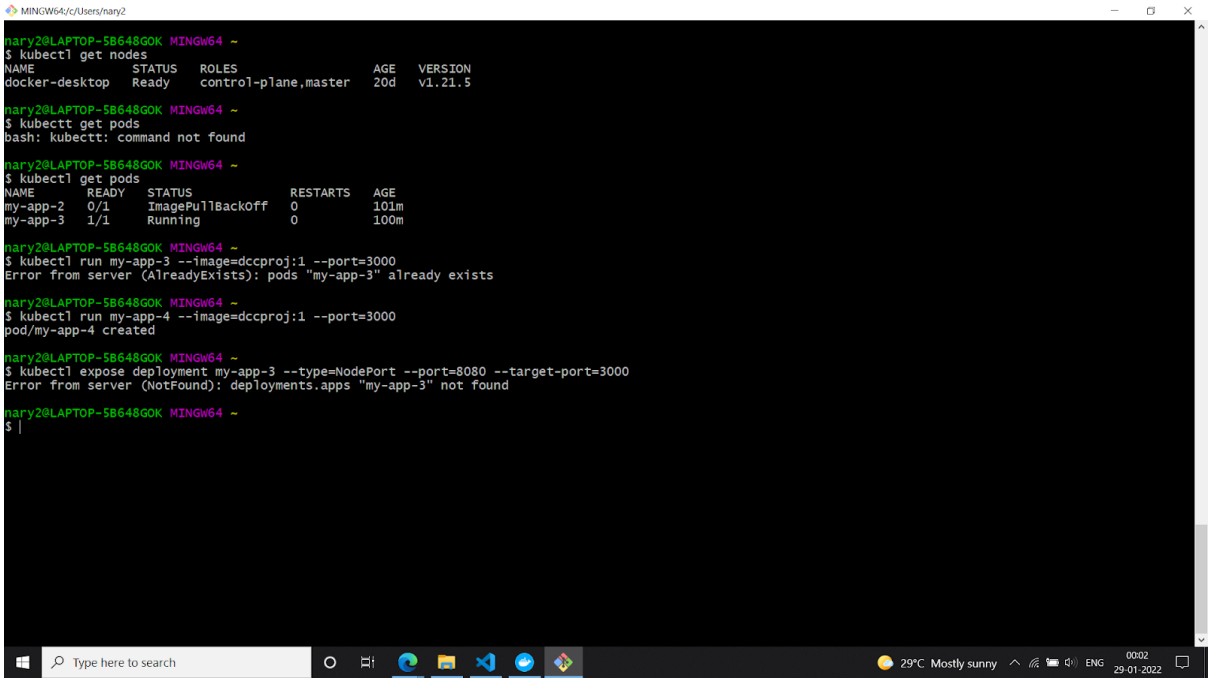
Replica Factor = 2

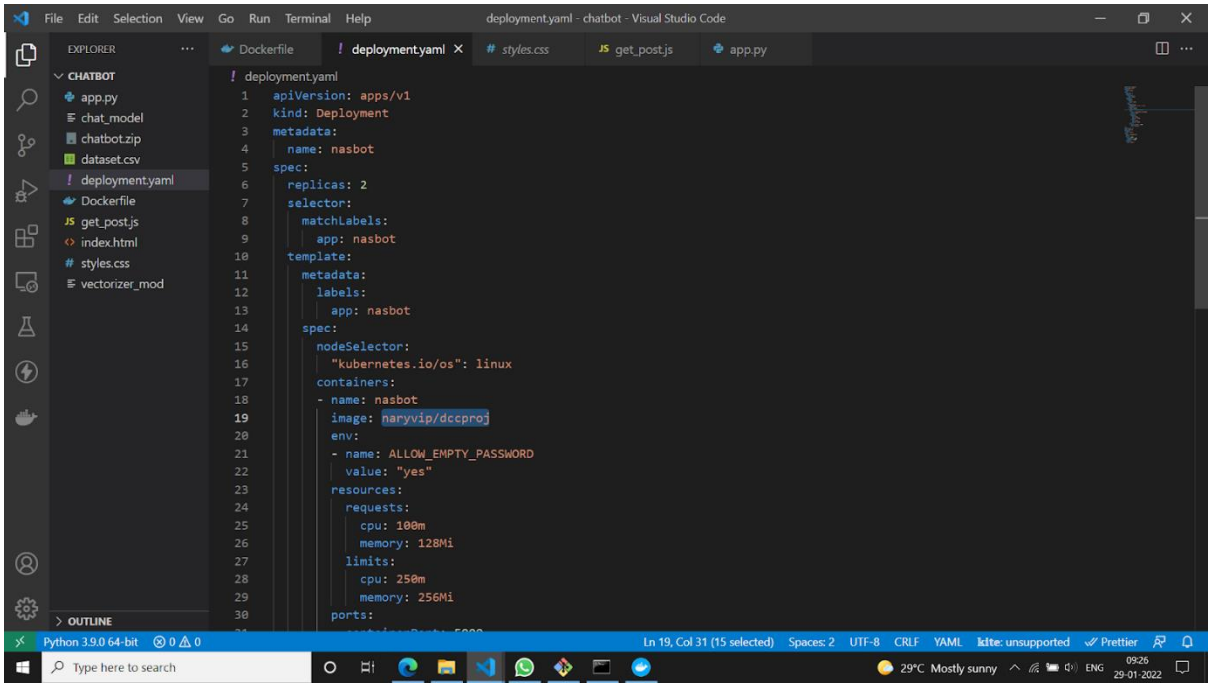
Expose service and make it accessible by other pods or from the external:

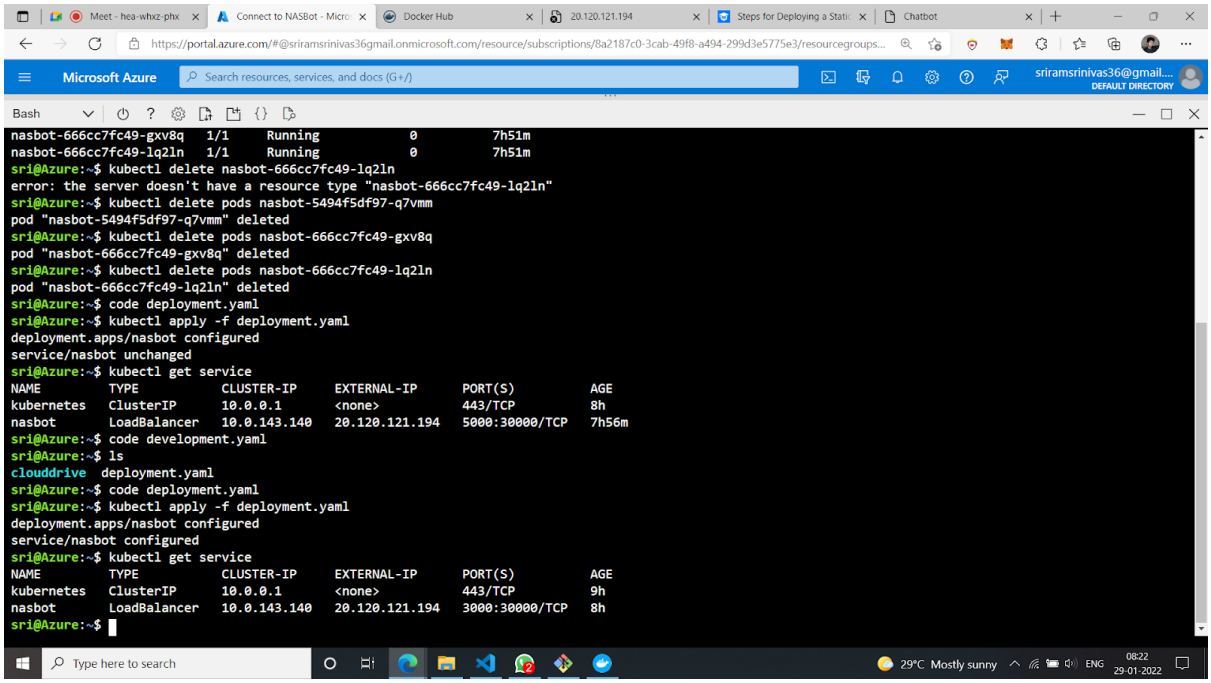


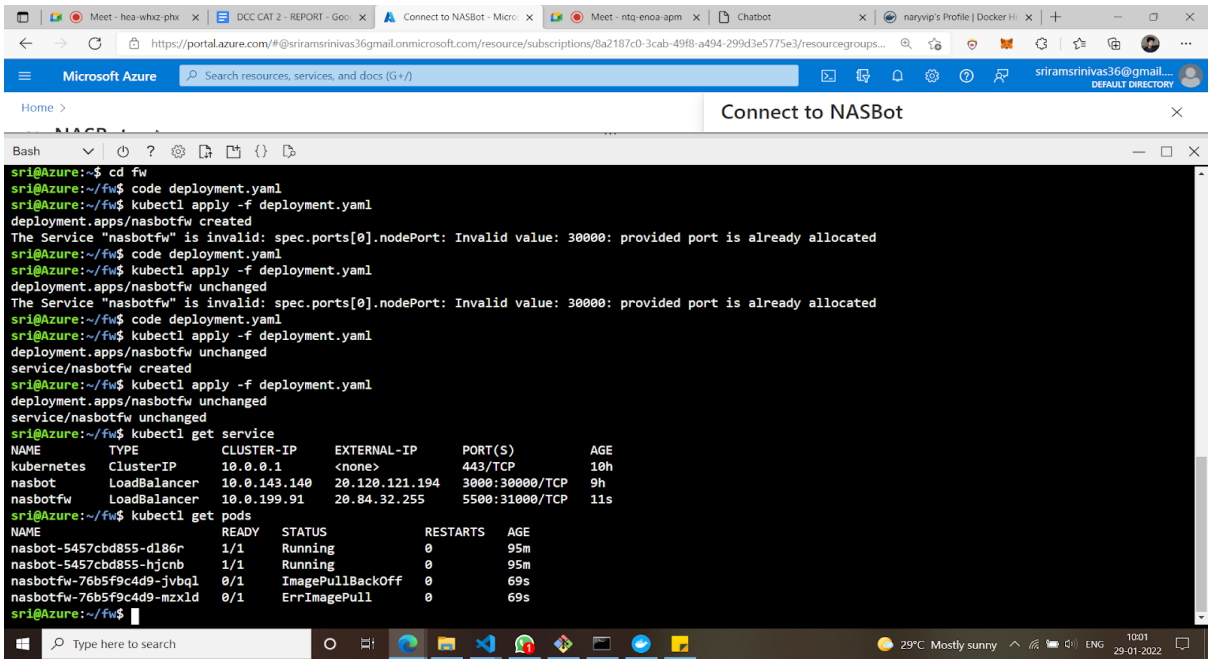
Describing the pod, deployment events and explain what happened in the system:

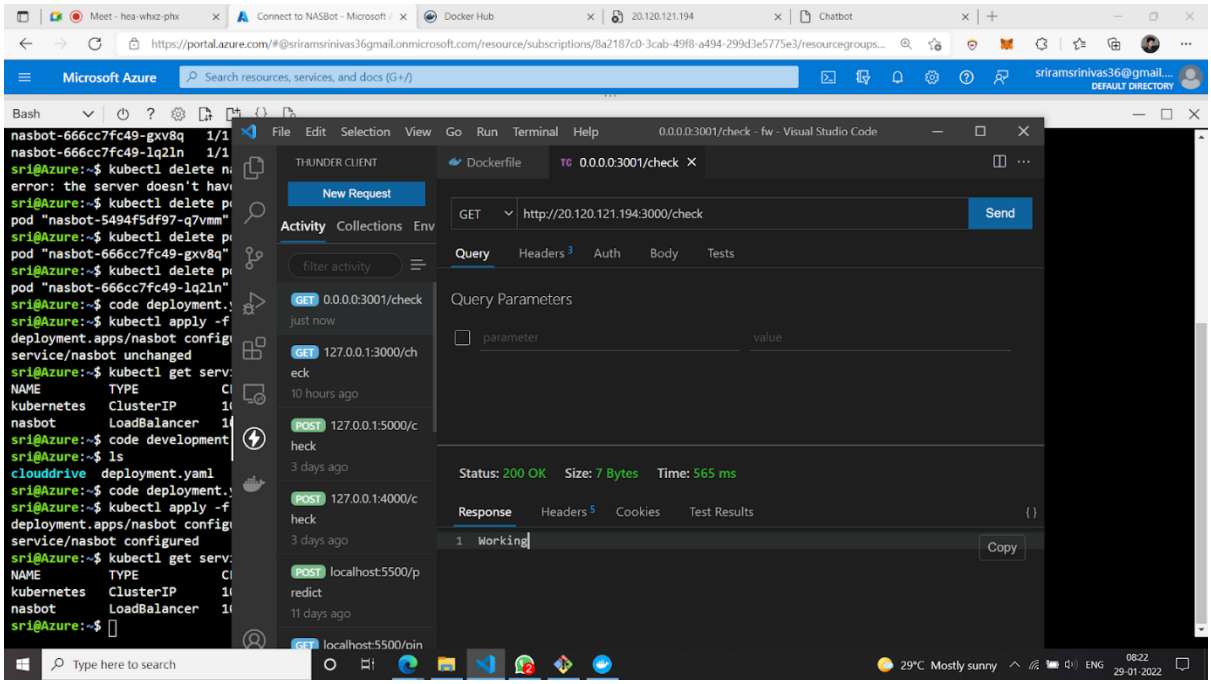


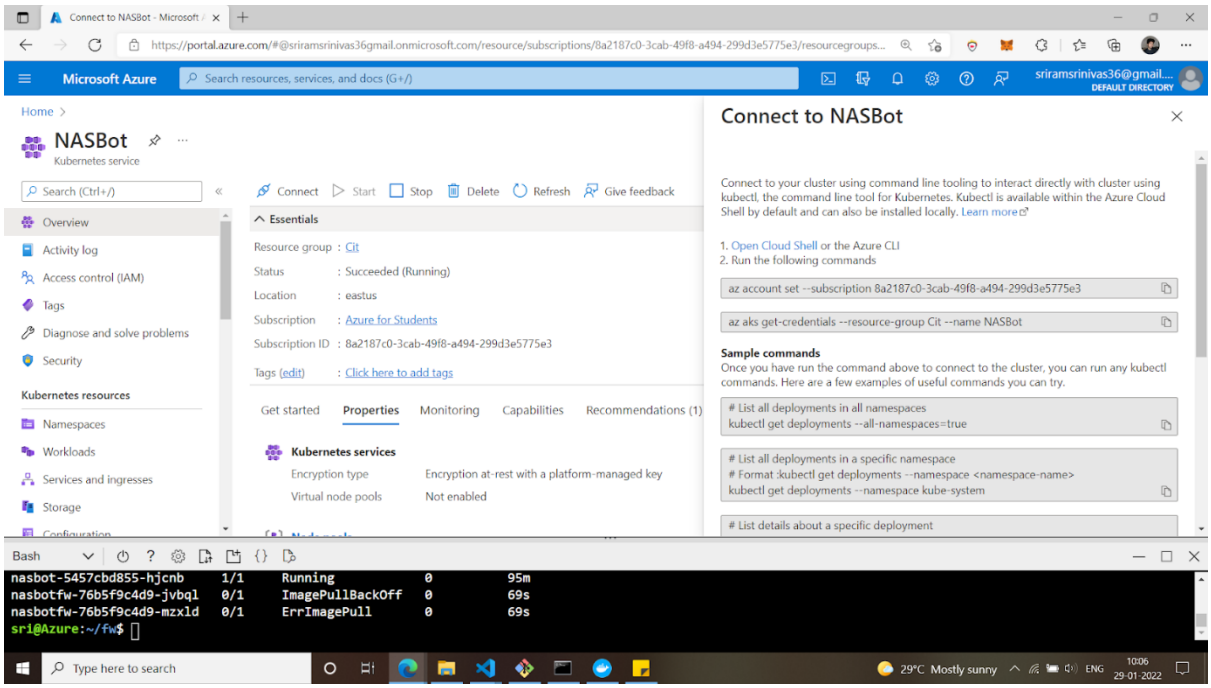

MINGW64:/c/Users/nary2
nary2@LAPTOP-5B648GOK MINGW64 ~
$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
docker-desktop Ready control-plane,master 20d v1.21.5
nary2@LAPTOP-5B648GOK MINGW64 ~
$ kubectt get pods
bash: kubectt: command not found
nary2@LAPTOP-5B648GOK MINGW64 ~
$ kubectl get pods
NAME READY STATUS RESTARTS AGE
my-app-2 0/1 ImagePullBackOff 0 101m
my-app-3 1/1 Running 0 100m
nary2@LAPTOP-5B648GOK MINGW64 ~
$ kubectl run my-app-3 --image=dccproj:1 --port=3000
Error from server (AlreadyExists): pods "my-app-3" already exists
nary2@LAPTOP-5B648GOK MINGW64 ~
$ kubectl run my-app-4 --image=dccproj:1 --port=3000
pod/my-app-4 created
nary2@LAPTOP-5B648GOK MINGW64 ~
$ kubectl expose deployment my-app-3 --type=NodePort --port=8080 --target-port=3000
Error from server (NotFound): deployments.apps "my-app-3" not found
nary2@LAPTOP-5B648GOK MINGW64 ~
$
Type here to search
29°C Mostly sunny
ENG
00:02
29-01-2022


File Edit Selection View Go Run Terminal Help
deployment.yaml - chatbot - Visual Studio Code
EXPLORER
CHATBOT
app.py
chat_model
chatbot.zip
dataset.csv
deployment.yaml
Dockerfile
get_post.js
index.html
styles.css
vectorizer_mod
OUTLINE
Dockerfile
deployment.yaml
styles.css
get_post.js
app.py
deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
name: nasbot
spec:
replicas: 2
selector:
matchLabels:
app: nasbot
template:
metadata:
labels:
app: nasbot
spec:
nodeSelector:
"kubernetes.io/os": linux
containers:
- name: nasbot
image: naryvip/dccproj
env:
- name: ALLOW_EMPTY_PASSWORD
value: "yes"
resources:
requests:
cpu: 100m
memory: 128Mi
limits:
cpu: 250m
memory: 256Mi
ports:
Python 3.9.0 64-bit
Ln 19, Col 31 (15 selected)
Spaces: 2
UTF-8
CRLF
YAML
kite: unsupported
Prettier
Type here to search
29°C Mostly sunny
ENG
09:26
29-01-2022

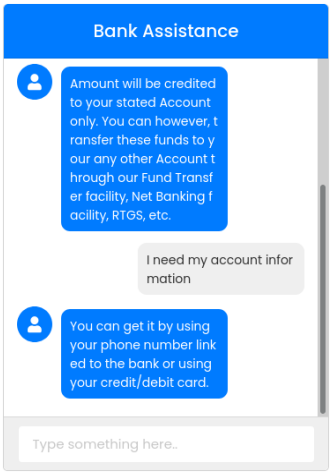

Microsoft Azure
Search resources, services, and docs (G+/)
sriramsrinivas36@gmail....
DEFAULT DIRECTORY
Bash
nasbot-666cc7fc49-gxv8q 1/1 Running 0 7h51m
nasbot-666cc7fc49-lq2ln 1/1 Running 0 7h51m
sri@Azure:~$ kubectl delete nasbot-666cc7fc49-lq2ln
error: the server doesn't have a resource type "nasbot-666cc7fc49-lq2ln"
sri@Azure:~$ kubectl delete pods nasbot-5494f5df97-q7vmm
pod "nasbot-5494f5df97-q7vmm" deleted
sri@Azure:~$ kubectl delete pods nasbot-666cc7fc49-gxv8q
pod "nasbot-666cc7fc49-gxv8q" deleted
sri@Azure:~$ kubectl delete pods nasbot-666cc7fc49-lq2ln
pod "nasbot-666cc7fc49-lq2ln" deleted
sri@Azure:~$ code deployment.yaml
sri@Azure:~$ kubectl apply -f deployment.yaml
deployment.apps/nasbot configured
service/nasbot unchanged
sri@Azure:~$ kubectl get service
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 8h
nasbot LoadBalancer 10.0.143.140 20.120.121.194 5000:30000/TCP 7h56m
sri@Azure:~$ code development.yaml
sri@Azure:~$ ls
clouddrive deployment.yaml
sri@Azure:~$ code deployment.yaml
sri@Azure:~$ kubectl apply -f deployment.yaml
deployment.apps/nasbot configured
service/nasbot configured
sri@Azure:~$ kubectl get service
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 9h
nasbot LoadBalancer 10.0.143.140 20.120.121.194 3000:30000/TCP 8h
sri@Azure:~$
Type here to search
29°C Mostly sunny
ENG
08:22
29-01-2022


Microsoft Azure
Search resources, services, and docs (G+/)
sriramsrinivas36@gmail....
DEFAULT DIRECTORY
Home >
Connect to NASBot
Bash
sri@Azure:~$ cd fw
sri@Azure:~/fw$ code deployment.yaml
sri@Azure:~/fw$ kubectl apply -f deployment.yaml
deployment.apps/nasbotfw created
The Service "nasbotfw" is invalid: spec.ports[0].nodePort: Invalid value: 30000: provided port is already allocated
sri@Azure:~/fw$ code deployment.yaml
sri@Azure:~/fw$ kubectl apply -f deployment.yaml
deployment.apps/nasbotfw unchanged
The Service "nasbotfw" is invalid: spec.ports[0].nodePort: Invalid value: 30000: provided port is already allocated
sri@Azure:~/fw$ code deployment.yaml
sri@Azure:~/fw$ kubectl apply -f deployment.yaml
deployment.apps/nasbotfw unchanged
service/nasbotfw created
sri@Azure:~/fw$ kubectl apply -f deployment.yaml
deployment.apps/nasbotfw unchanged
service/nasbotfw unchanged
sri@Azure:~/fw$ kubectl get service
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 10h
nasbot LoadBalancer 10.0.143.140 20.120.121.194 3000:30000/TCP 9h
nasbotfw LoadBalancer 10.0.199.91 20.84.32.255 5500:31000/TCP 11s
sri@Azure:~/fw$ kubectl get pods
NAME READY STATUS RESTARTS AGE
nasbot-5457cbd855-d186r 1/1 Running 0 95m
nasbot-5457cbd855-hjcnb 1/1 Running 0 95m
nasbotfw-76b5f9c4d9-jvbql 0/1 ImagePullBackOff 0 69s
nasbotfw-76b5f9c4d9-mzxld 0/1 ErrImagePull 0 69s
sri@Azure:~/fw$
Type here to search
29°C Mostly sunny
ENG
10:01
29-01-2022


Microsoft Azure
Search resources, services, and docs (G+/)
sriramsrinivas36@gmail....
DEFAULT DIRECTORY
Bash
nasbot-666cc7fc49-gxv8q 1/1
nasbot-666cc7fc49-lq2ln 1/1
sri@Azure:~$ kubectl delete na
error: the server doesn't hav
sri@Azure:~$ kubectl delete po
pod "nasbot-5494f5df97-q7vmm"
sri@Azure:~$ kubectl delete po
pod "nasbot-666cc7fc49-gxv8q"
sri@Azure:~$ kubectl delete po
pod "nasbot-666cc7fc49-lq2ln"
sri@Azure:~$ code deployment.
sri@Azure:~$ kubectl apply -f
deployment.apps/nasbot config
service/nasbot unchanged
sri@Azure:~$ kubectl get serv
NAME TYPE
kubernetes ClusterIP
nasbot LoadBalancer
sri@Azure:~$ code development
sri@Azure:~$ ls
clouddrive deployment.yaml
sri@Azure:~$ code deployment.
sri@Azure:~$ kubectl apply -f
deployment.apps/nasbot config
service/nasbot configured
sri@Azure:~$ kubectl get serv
NAME TYPE
kubernetes ClusterIP
nasbot LoadBalancer
sri@Azure:~$
File Edit Selection View Go Run Terminal Help
0.0.0.0:3001/check - fw - Visual Studio Code
THUNDER CLIENT
New Request
Activity Collections Env
filter activity
GET 0.0.0.0:3001/check
just now
GET 127.0.0.1:3000/ch
eck
10 hours ago
POST 127.0.0.1:5000/c
heck
3 days ago
POST 127.0.0.1:4000/c
heck
3 days ago
POST localhost:5500/p
redict
11 days ago
Dockerfile
0.0.0.0:3001/check
GET
http://20.120.121.194:3000/check
Send
Query Headers 3 Auth Body Tests
Query Parameters
parameter
value
Status: 200 OK Size: 7 Bytes Time: 565 ms
Response Headers 5 Cookies Test Results
1 Working
Copy
Type here to search
29°C Mostly sunny
ENG
08:22
29-01-2022


Microsoft Azure
Search resources, services, and docs (G+/)
sriramsrinivas36@gmail....
DEFAULT DIRECTORY
Home
NASBot
Kubernetes service
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Security
Kubernetes resources
Namespaces
Workloads
Services and ingresses
Storage
Connect Start Stop Delete Refresh Give feedback
Essentials
Resource group : Cit
Status : Succeeded (Running)
Location : eastus
Subscription : Azure for Students
Subscription ID : 8a2187c0-3cab-49f8-a494-299d3e5775e3
Tags (edit) : Click here to add tags
Get started Properties Monitoring Capabilities Recommendations (1)
Kubernetes services
Encryption type Encryption at-rest with a platform-managed key
Virtual node pools Not enabled
Connect to NASBot
Connect to your cluster using command line tooling to interact directly with cluster using kubectl, the command line tool for Kubernetes. Kubectl is available within the Azure Cloud Shell by default and can also be installed locally. Learn more
1. Open Cloud Shell or the Azure CLI
2. Run the following commands
az account set --subscription 8a2187c0-3cab-49f8-a494-299d3e5775e3
az aks get-credentials --resource-group Cit --name NASBot
Sample commands
Once you have run the command above to connect to the cluster, you can run any kubectl commands. Here are a few examples of useful commands you can try.
List all deployments in all namespaces
kubectl get deployments --all-namespaces=true
List all deployments in a specific namespace
Format :kubectl get deployments --namespace <namespace-name>
kubectl get deployments --namespace kube-system
List details about a specific deployment
Bash
nasbot-5457cbd855-hjcnb 1/1 Running 0 95m
nasbotfw-76b5f9c4d9-jvbql 0/1 ImagePullBackOff 0 69s
nasbotfw-76b5f9c4d9-mzxld 0/1 ErrImagePull 0 69s
sri@Azure:~/fw$
Type here to search
29°C Mostly sunny
ENG
10:06
29-01-2022

OUR FINAL APPLICATION



CONCLUSION:

We successfully deployed both Backend API and Frontend docker containers in Microsoft Azure Kubernetes service and it can be now accessed anywhere in any device.

Further Improvement in Future:

- We will try to make the ML model more accurate and faster.
- Add rest API in Frontend Kubernetes.

Thank You