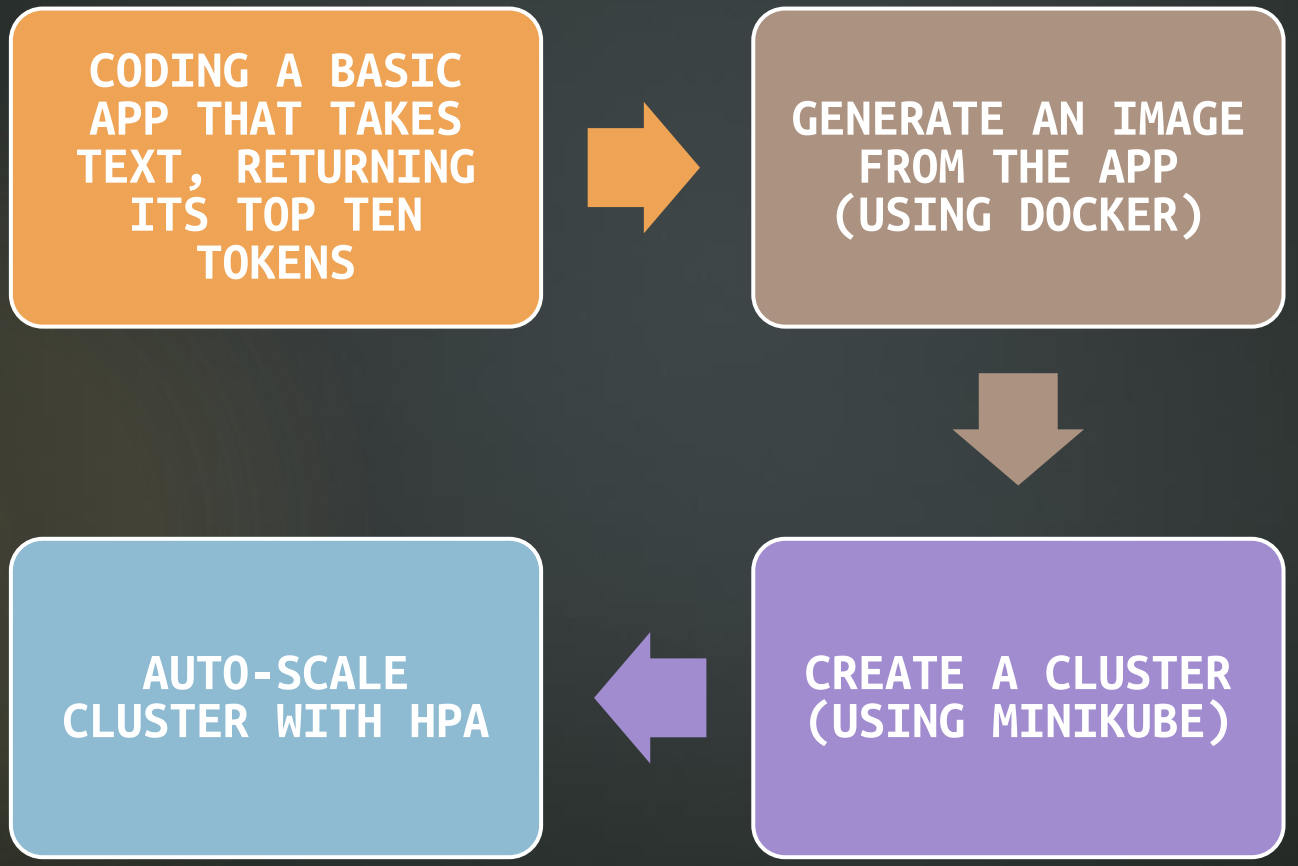
Tokens Generator Distributed System

Moscovitz Ori

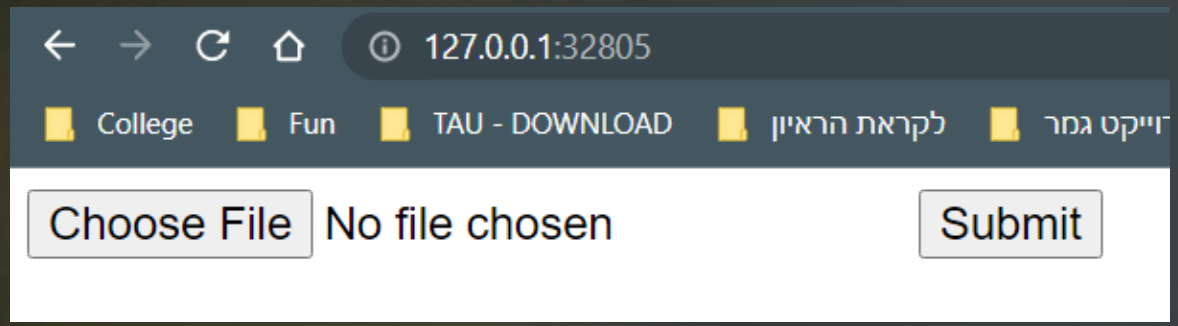
The Process

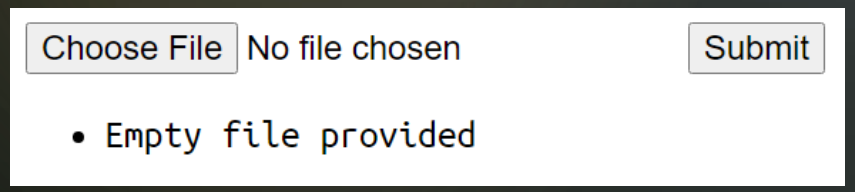

CODING A BASIC APP THAT TAKES TEXT, RETURNING ITS TOP TEN TOKENS
GENERATE AN IMAGE FROM THE APP (USING DOCKER)
CREATE A CLUSTER (USING MINIKUBE)
AUTO-SCALE CLUSTER WITH HPA

Tokens Generator App Build Blocks

```python
@app.route('/', methods=['GET'])
def index():
    return render_template('index.html')
```

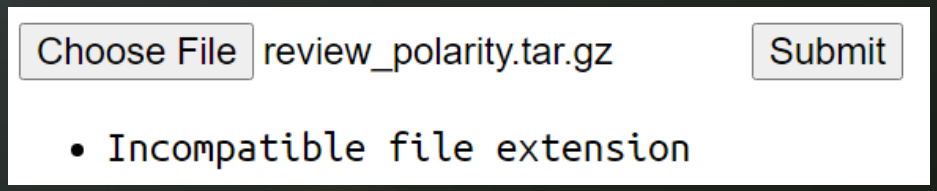
With Flask, returning a main page for GET requests





Empty Submission

User Indicators when upload isn't successful



Non-text file

Tokens Generator App Build Blocks

```python
# turn a doc into clean tokens
def clean_doc(doc):
    # split into tokens by white space
    tokens = doc.split()
    # prepare regex for char filtering
    re_punc = re.compile('[%s]' % re.escape(string.punctuation))
    # remove punctuation from each word
    tokens = [re_punc.sub('', w) for w in tokens]
    # remove remaining tokens that are not alphabetic
    tokens = [word for word in tokens if word.isalpha()]
    # filter out stop words
    stop_words = set(stopwords.words('english'))
    tokens = [w for w in tokens if w not in stop_words]
    # filter out short tokens
    tokens = [word for word in tokens if len(word) > 1]
    # Count occurrence
    c = Counter(tokens)
    # Keep 10 most common tokens
    tokens = []
    for val in c.most_common(10):
        tokens.append(val[0])
    return tokens
```

Text Cleaning function that removes punctuation, non alphabetic, stop words and short tokens then returns the ten most common out of that list

A proper upload returns a list of the ten most repeated tokens in the document

Top 10 tokens:

of

the

in

and

is

to

his

about

its

it

App → Docker Image

Using a Dockerfile that will:

- **Copy all apps content**
- **Install Python 3.9 and all relevant requirements needed for the app to run**

```
FROM python:3.9

WORKDIR /DS

COPY . .

RUN apt-get clean && apt-get -y update

RUN apt install python3.9

RUN python3 -m pip install -r requirements.txt

# exposing flask default port to the container
EXPOSE 5000


CMD ["python", "main.py"]
```

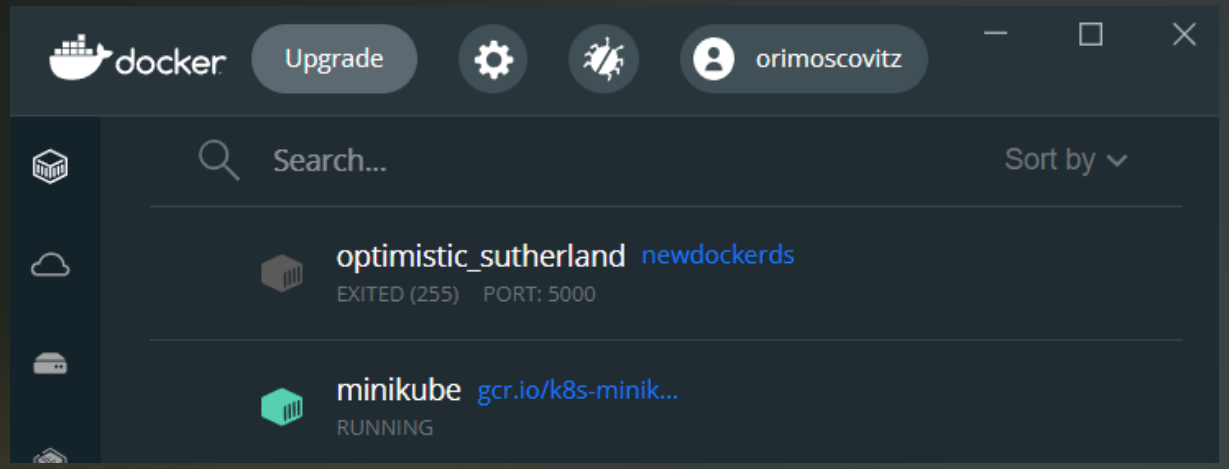
Dockerfile

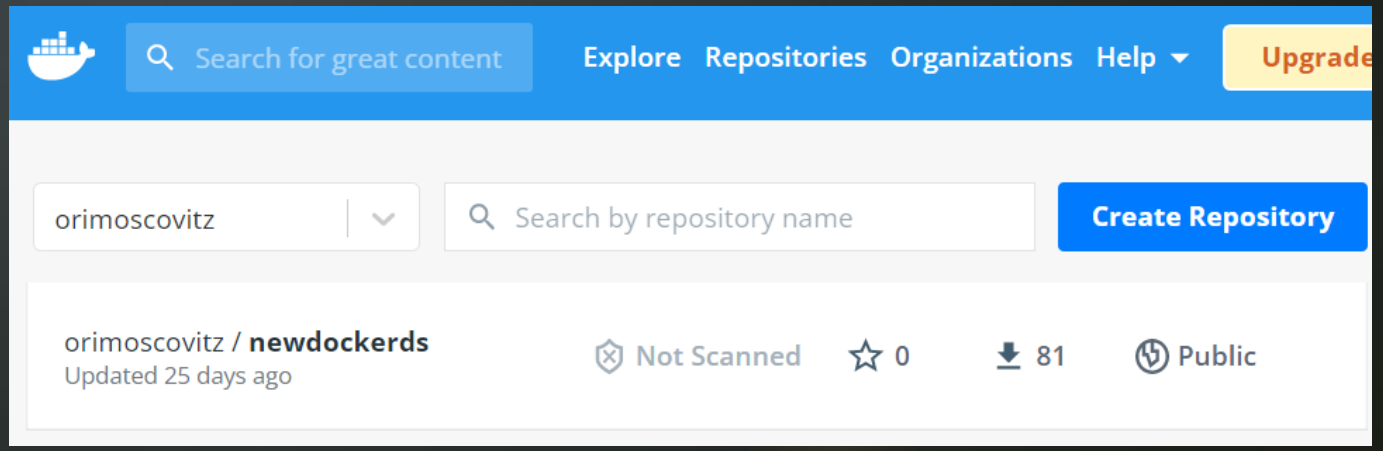
Requirements.txt

```
Flask==2.0.2
Jinja2==3.0.3
Werkzeug==2.0.2
regex==2021.11.10
nltk==3.6.5
pip~=21.3.1
wheel~=0.37.0
future~=0.18.2
```

```
click~=8.0.3
tqdm~=4.62.3
colorama~=0.4.4
MarkupSafe~=2.0.1
itsdangerous~=2.0.1
joblib~=1.1.0
altgraph~=0.17.2
pefile~=2021.9.3
setuptools~=58.5.3
pyinstaller~=4.7
```

App → Docker Image





Create a cluster with K8S

Using WSL2 to run ubuntu to deploy the app with K8S (via minikube)

“kubectl create deployment...”

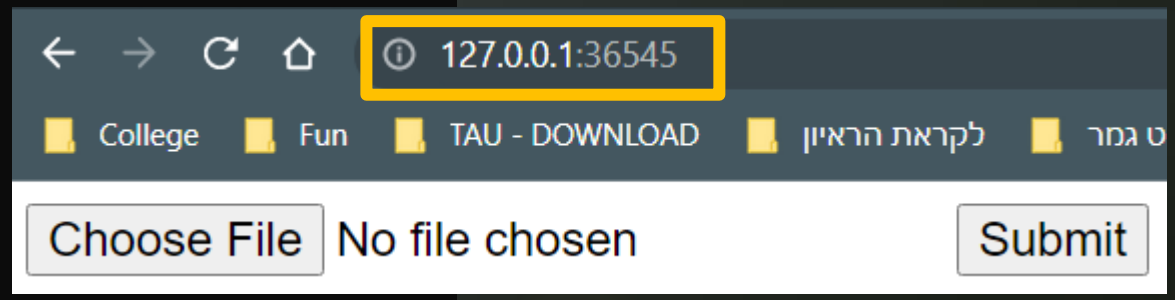
```
ori@MSI:~$ kubectl get pods
NAME                          READY   STATUS    RESTARTS      AGE
dsdeploy-6598f5ddc7-gttpp     1/1     Running   1 (56m ago)   83m
```

Main node (pod)

Expose Deployment and listen to the service with

“kubectl expose deployment...” &

```
ori@MSI:~$ minikube service dsdeploy --url
🏃  Starting tunnel for service dsdeploy.
|-----------|----------|-------------|------------------------|
| NAMESPACE |   NAME   | TARGET PORT |          URL           |
|-----------|----------|-------------|------------------------|
| default   | dsdeploy |             | http://127.0.0.1:36545 |
|-----------|----------|-------------|------------------------|
http://127.0.0.1:36545
❗  Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```



Horizontal Auto-scale cluster

```
ori@MSI:~$ kubectl get hpa
NAME       REFERENCE              TARGETS          MINPODS   MAXPODS   REPLICAS   AGE
dsdeploy   Deployment/dsdeploy    <unknown>/10%    1         10        0          3s
```

```
ori@MSI:~$ kubectl get hpa --watch
NAME       REFERENCE              TARGETS    MINPODS   MAXPODS   REPLICAS   AGE
dsdeploy   Deployment/dsdeploy    8%/10%     1         10        1          115s
dsdeploy   Deployment/dsdeploy    9%/10%     1         10        1          2m1s
dsdeploy   Deployment/dsdeploy    9%/10%     1         10        1          2m31s
dsdeploy   Deployment/dsdeploy    10%/10%    1         10        1          2m46s
dsdeploy   Deployment/dsdeploy    7%/10%     1         10        1          3m1s
dsdeploy   Deployment/dsdeploy    6%/10%     1         10        1          3m16s
dsdeploy   Deployment/dsdeploy    6%/10%     1         10        1          3m31s
dsdeploy   Deployment/dsdeploy    10%/10%    1         10        1          3m46s
dsdeploy   Deployment/dsdeploy    8%/10%     1         10        1          4m25s
dsdeploy   Deployment/dsdeploy    18%/10%    1         10        1          4m38s
dsdeploy   Deployment/dsdeploy    18%/10%    1         10        2          4m53s
dsdeploy   Deployment/dsdeploy    9%/10%     1         10        2          5m8s
dsdeploy   Deployment/dsdeploy    8%/10%     1         10        2          5m38s
dsdeploy   Deployment/dsdeploy    7%/10%     1         10        2          5m54s
dsdeploy   Deployment/dsdeploy    7%/10%     1         10        2          6m19s
dsdeploy   Deployment/dsdeploy    19%/10%    1         10        2          6m33s
dsdeploy   Deployment/dsdeploy    7%/10%     1         10        4          6m48s
dsdeploy   Deployment/dsdeploy    6%/10%     1         10        4          7m3s
```

Defined at 10%, when service exceeds this, auto-scaling will come into play, creating up to 10 replicas (arbitrary), to handle the requests

Horizontal Auto-scale cluster

```
ori@MSI:~$ kubectl get hpa --watch
NAME       REFERENCE                TARGETS   MINPODS   MAXPODS   REPLICAS   AGE
dsdeploy   Deployment/dsdeploy      8%/10%    1         10        1          115s
dsdeploy   Deployment/dsdeploy      9%/10%    1         10        1          2m1s
dsdeploy   Deployment/dsdeploy      9%/10%    1         10        1          2m31s
dsdeploy   Deployment/dsdeploy      10%/10%   1         10        1          2m46s
dsdeploy   Deployment/dsdeploy      7%/10%    1         10        1          3m1s
dsdeploy   Deployment/dsdeploy      6%/10%    1         10        1          3m16s
dsdeploy   Deployment/dsdeploy      6%/10%    1         10        1          3m31s
dsdeploy   Deployment/dsdeploy      10%/10%   1         10        1          3m46s
dsdeploy   Deployment/dsdeploy      8%/10%    1         10        1          4m25s
dsdeploy   Deployment/dsdeploy      18%/10%   1         10        1          4m38s
dsdeploy   Deployment/dsdeploy      18%/10%   1         10        2          4m53s
dsdeploy   Deployment/dsdeploy      9%/10%    1         10        2          5m8s
dsdeploy   Deployment/dsdeploy      8%/10%    1         10        2          5m38s
dsdeploy   Deployment/dsdeploy      7%/10%    1         10        2          5m54s
dsdeploy   Deployment/dsdeploy      7%/10%    1         10        2          6m10s
dsdeploy   Deployment/dsdeploy      19%/10%   1         10        2          6m33s
dsdeploy   Deployment/dsdeploy      7%/10%    1         10        4          6m48s
dsdeploy   Deployment/dsdeploy      6%/10%    1         10        4          7m3s
```

Added nine new replicas to handle the excessive amount of requests

```
ori@MSI:~$ kubectl get pods --watch
NAME                        READY   STATUS              RESTARTS      AGE
dsdeploy-6598f5ddc7-gttpp   1/1     Running             1 (73m ago)   100m
dsdeploy-6598f5ddc7-k6scf   1/1     Running             0             20s
dsdeploy-6598f5ddc7-jldth   0/1     Pending             0             0s
dsdeploy-6598f5ddc7-2tqrh   0/1     Pending             0             0s
dsdeploy-6598f5ddc7-jldth   0/1     Pending             0             0s
dsdeploy-6598f5ddc7-2tqrh   0/1     Pending             0             0s
dsdeploy-6598f5ddc7-jldth   0/1     ContainerCreating   0             0s
dsdeploy-6598f5ddc7-2tqrh   0/1     ContainerCreating   0             0s
dsdeploy-6598f5ddc7-2tqrh   1/1     Running             0             14s
dsdeploy-6598f5ddc7-jldth   1/1     Running             0             14s
```