

Software Engineering for the Cloud

1/ Start with a single local service

I use python for simplicity :

```
app.py 1 X
C: > Users > julie > OneDrive > Bureau > Software Engineering for the Cloud > app > app.py > ...
1  from flask import Flask
2
3  app = Flask(__name__)
4
5  @app.route('/')
6  def hello():
7      return 'Hello, this is your mini application!'
8
9  if __name__ == '__main__':
10     app.run(debug=True, host='0.0.0.0')
11
```

Then I create the Dockerfile :

```
C: > Users > julie > OneDrive > Bureau > Software Engineering for the Cloud > app > Dockerfile
1  # Use an official Python runtime as a parent image
2  FROM python:3.11.5
3
4  # Set the working directory to /app
5  WORKDIR /app
6
7  # Copy the current directory contents into the container at /app
8  COPY . /app
9
10 # Install any needed packages specified in requirements.txt
11 RUN pip install --trusted-host pypi.python.org flask
12
13 # Make port 80 available to the world outside this container
14 EXPOSE 80
15
16 # Define environment variable
17 ENV NAME World
18
19 # Run app.py when the container launches
20 CMD ["python", "app.py"]
21
```

I build and publish docker image :

```
Microsoft Windows [version 10.0.22631.3887]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\julie\OneDrive\Bureau\Software Engineering for the Cloud\app>docker build -t sl33pysl0ud/mini-app:latest .
[*] Building 4.5s (10/10) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 556B                                              0.0s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 28                                                  0.0s
=> [internal] load metadata for docker.io/library/python:3.11.5                1.5s
=> [auth] library/python:pull token for registry-1.docker.io                  0.0s
=> [internal] load build context                                                0.0s
=> => transferring context: 1.29kB                                              0.0s
=> [1/4] FROM docker.io/library/python:3.11.5@sha256:2e376999a11f1c1e037796d08db0e99c36eadb4bb6491372b227f1e53c34 0.0s
=> CACHED [2/4] WORKDIR /app                                                    0.0s
=> [3/4] COPY . /app                                                            0.1s
=> [4/4] RUN pip install --trusted-host pypi.python.org flask                  2.7s
=> exporting to image                                                            0.1s
=> => exporting layers                                                            0.1s
=> => writing image sha256:694511d529450dc75901f486cf6766c482d0454ef5eeb60fe78b309f5ae97bad 0.0s
=> => naming to docker.io/sl33pysl0ud/mini-app:latest                          0.0s

C:\Users\julie\OneDrive\Bureau\Software Engineering for the Cloud\app>docker push sl33pysl0ud/mini-app:latest
The push refers to repository [docker.io/sl33pysl0ud/mini-app]
c58513d807dd: Pushed
a47160e0694f: Pushed
931c6e5dd8d7: Pushed
db22e01d36b: Mounted from library/python
0d3f1aea6da4: Pushed
78dd9ecf8a6d: Mounted from library/python
c26432533a6a: Mounted from library/python
01d6cdeac539: Mounted from library/python
a981dddd4c65: Mounted from library/python
f6809995d5b5: Mounted from library/python
7c85cfa30cb1: Mounted from library/python
latest: digest: sha256:e8ba4dc9adab73246aef36bc6c788ace1d3a4667e6263a5b564ba1e62514129c size: 2632

C:\Users\julie\OneDrive\Bureau\Software Engineering for the Cloud\app>
```

I create the deployment file :

```
app.py 1 Dockerfile ! deployment.yaml X
C:\Users\julie> OneDrive > Bureau > Software Engineering for the Cloud > app > ! deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: mini-app
5  spec:
6    replicas: 3
7    selector:
8      matchLabels:
9        app: mini-app
10   template:
11     metadata:
12       labels:
13         app: mini-app
14     spec:
15       containers:
16       - name: mini-app
17         image: sl33pycloud/mini-app:latest
18         ports:
19         - containerPort: 5000
20
```

Then apply the deployment configuration :

```
C:\Users\julie\OneDrive\Bureau\Software Engineering for the Cloud\app>minikube start
* minikube v1.32.0 sur Microsoft Windows 11 Pro 10.0.22631.3007 Build 22631.3007
* Utilisation du pilote docker basé sur le profil existant
* Démarrage du noeud de plan de contrôle minikube dans le cluster minikube
* Extraction de l'image de base...
* Mise à jour du container docker en marche "minikube" ...
* Préparation de Kubernetes v1.28.3 sur Docker 24.0.7...
* Configuration de bridge CNI (Container Networking Interface)...
* Vérification des composants Kubernetes...
  - Utilisation de l'image gcr.io/k8s-minikube/storage-provisioner:v5
* Modules activés: storage-provisioner, default-storageclass

! C:\Program Files\Docker\Docker\resources\bin\kubectl.exe est la version 1.25.9, qui peut comporter des incompatibilités avec Kubernetes 1.28.3.
  - Vous voulez kubectl v1.28.3 ? Essayez 'minikube kubectl -- get pods -A'
* Terminé ! kubectl est maintenant configuré pour utiliser "minikube" cluster et espace de noms "default" par défaut.

C:\Users\julie\OneDrive\Bureau\Software Engineering for the Cloud\app>kubectl apply -f deployment.yaml
deployment.apps/mini-app unchanged
```

I create the service file :

```
C:\Users\julie> OneDrive > Bureau > Software Engineering for the Cloud > app > ! service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: mini-app
5  spec:
6    selector:
7      app: mini-app
8    ports:
9      - protocol: TCP
10        port: 5000
11        targetPort: 5000
12    type: LoadBalancer
13
14
```

Then apply the service configuration :

```
C:\Users\julie\OneDrive\Bureau\Software Engineering for the Cloud\app>kubectl apply -f service.yaml
service/mini-app unchanged
```

Checking the status :

```
C:\Users\julie\OneDrive\Bureau\Software Engineering for the Cloud\app>kubectl get service mini-app
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
mini-app  LoadBalancer  10.97.22.83   127.0.0.1     5000:31545/TCP   12d
```

2/ Add a local gateway

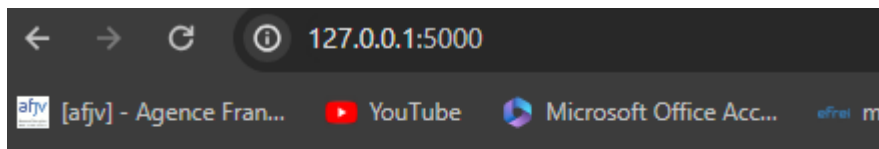
By using minikube tunnel, you create a local gateway that makes services accessible from your machine.

```
Microsoft Windows [version 10.0.22631.3007]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\julie\OneDrive\Bureau\Software Engineering for the Cloud>minikube tunnel
* Tunnel démarré avec succès

* REMARQUE : veuillez ne pas fermer ce terminal car ce processus doit rester actif pour que le tunnel soit accessible...
* Tunnel de démarrage pour le service mini-app.
```

Now let's open a web browser and navigate to the external IP. I should be able to see my mini application.



Hello, this is your mini application!

Working !

3 and 4/ Add a second service and add a local database

I repeat all the previous step to create a second service but this time with a local database in it :

```

1  from flask import Flask, request
2
3  from flask_sqlalchemy import SQLAlchemy
4
5  app = Flask(__name__)
6  app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///app2.db'
7  db = SQLAlchemy(app)
8
9  class Message(db.Model):
10     id = db.Column(db.Integer, primary_key=True)
11     content = db.Column(db.String(200), nullable=False)
12
13  @app.route('/')
14  def index():
15     messages = Message.query.all()
16     html_content = "<h1>Welcome to App2</h1><ul>"
17     for message in messages:
18         html_content += f"<li>{message.content}</li>"
19     html_content += "</ul>"
20     return html_content
21
22  @app.route('/add_message', methods=['POST'])
23  def add_message():
24     content = request.form.get('content')
25     new_message = Message(content=content)
26     db.session.add(new_message)
27     db.session.commit()
28     return 'Message added successfully!'
29
30  if __name__ == '__main__':
31     db.create_all()
32     app.run(debug=True, host='0.0.0.0')
33
34

```

```

1  # Use an official Python runtime as a parent image
2  FROM python:3.11.5
3
4  # Set the working directory to /app
5  WORKDIR /app2
6
7  # Copy the current directory contents into the container at /app
8  COPY . /app2
9
10 # Install any needed packages specified in requirements.txt
11 RUN pip install --trusted-host pypi.python.org flask flask_sqlalchemy
12
13 # Make port 80 available to the world outside this container
14 EXPOSE 80
15
16 # Define environment variable
17 ENV NAME World
18
19 # Run app2.py when the container launches
20 CMD ["python", "app2.py"]
21
22

```

```
C:\Users\julie\OneDrive\Bureau\Software Engineering for the Cloud\app>docker build -t sl33pycl0ud/mini-app2:latest .
[+] Building 1.3s (10/10) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 556B
=> [internal] load metadata for docker.io/library/python:3.11.5
=> [auth] library/python:pull token for registry-1.docker.io
=> [1/4] FROM docker.io/library/python:3.11.5@sha256:2e376990a11f1c1e03796d08db0e99c36eadb4bb6491372b227f1e53c3482914
=> [internal] load build context
=> => transferring context: 156B
=> CACHED [2/4] WORKDIR /app
=> CACHED [3/4] COPY . /app
=> CACHED [4/4] RUN pip install --trusted-host pypi.python.org flask
=> exporting to image
=> => exporting layers
=> => writing image sha256:694511d529450dc75901f486cf6766c482d0454ef5eeb60fe78b309f5ae97bad
=> => naming to docker.io/sl33pycl0ud/mini-app2:latest
```

```
C:\Users\julie\OneDrive\Bureau\Software Engineering for the Cloud\app>docker push sl33pycl0ud/mini-app2:latest
The push refers to repository [docker.io/sl33pycl0ud/mini-app2]
c58513d807dd: Mounted from sl33pycl0ud/mini-app
a47166e0694f: Mounted from sl33pycl0ud/mini-app
931c6e5dd0d7: Mounted from sl33pycl0ud/mini-app
db22e0d1d36b: Mounted from sl33pycl0ud/mini-app
0d3f1aea6da4: Mounted from sl33pycl0ud/mini-app
78dd9ecf8a6d: Mounted from sl33pycl0ud/mini-app
c26432533a6a: Mounted from sl33pycl0ud/mini-app
01d6cdeac539: Mounted from sl33pycl0ud/mini-app
a981ddddd4c65: Mounted from sl33pycl0ud/mini-app
f6589095d5b5: Mounted from sl33pycl0ud/mini-app
7c85cfa30cb1: Mounted from sl33pycl0ud/mini-app
latest: digest: sha256:e8ba4dc9adab73246aef36bc6c788ace1d3a4667e6263a5b564ba1e62514129c size: 2632
```

```
C: > Users > julie > OneDrive > Bureau > Software Engineering for the Cloud > app2 > ! deployment2.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: mini-app2-deployment
5  spec:
6    replicas: 3
7    selector:
8      matchLabels:
9        app: mini-app2
10   template:
11     metadata:
12       labels:
13         app: mini-app2
14     spec:
15       containers:
16       - name: mini-app2
17         image: sl33pycl0ud/mini-app2:latest
18         ports:
19         - containerPort: 5000
20
```

```
C:\Users\julie\OneDrive\Bureau\Software Engineering for the Cloud\app2>kubectl apply -f deployment2.yaml
deployment.apps/mini-app2-deployment created
```

```
C: > Users > julie > OneDrive > Bureau > Software Engineering for the Cloud > app2 > ! service2.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: mini-app2-service
5  spec:
6    selector:
7      app: mini-app2
8    ports:
9      - protocol: TCP
10        port: 5000
11        targetPort: 5000
12    type: LoadBalancer
13
```

```
C:\Users\julie\OneDrive\Bureau\Software Engineering for the Cloud\app2>kubectl apply -f service2.yaml
service/mini-app2-service created
```

```
C:\Users\julie\OneDrive\Bureau\Software Engineering for the Cloud\app2>kubectl get service mini-app2-service
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
mini-app2-service   LoadBalancer  10.107.41.78   127.0.0.1       5000:30157/TCP    2m
```

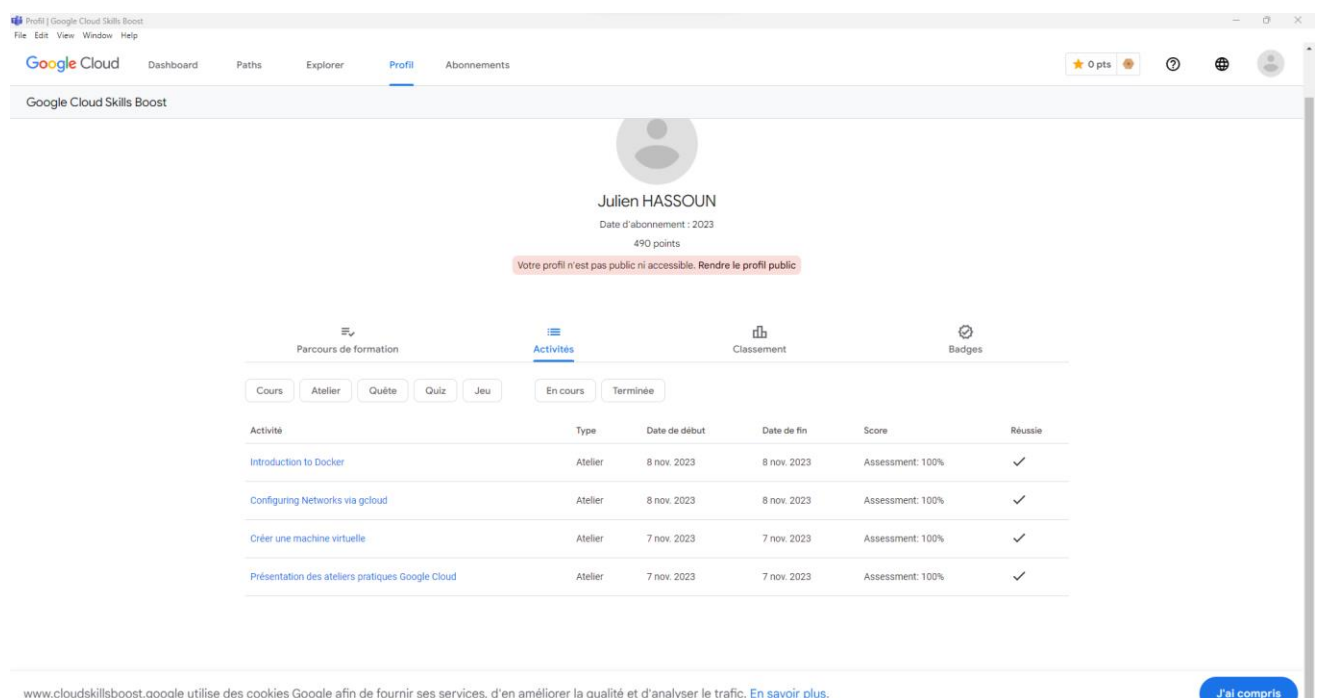
Now I use again the minikube tunnel :

```
C:\Users\julie\OneDrive\Bureau\Software Engineering for the Cloud>minikube tunnel
* Tunnel démarré avec succès

* REMARQUE : veuillez ne pas fermer ce terminal car ce processus doit rester actif pour que le tunnel soit accessible...
* Tunnel de démarrage pour le service mini-app.
* Tunnel de démarrage pour le service mini-app2-service.
```

Unfortunately, I didn't get the time to do the Deployment in a cloud infrastructure.

5/ Google labs



Google Cloud Skills Boost

Julien HASSOUN
Date d'abonnement : 2023
490 points
Votre profil n'est pas public ni accessible. Rendre le profil public

Parcours de formation

Activités

Activité	Type	Date de début	Date de fin	Score	Réussie
Introduction to Docker	Atelier	8 nov. 2023	8 nov. 2023	Assessment: 100%	✓
Configuring Networks via gcloud	Atelier	8 nov. 2023	8 nov. 2023	Assessment: 100%	✓
Créer une machine virtuelle	Atelier	7 nov. 2023	7 nov. 2023	Assessment: 100%	✓
Présentation des ateliers pratiques Google Cloud	Atelier	7 nov. 2023	7 nov. 2023	Assessment: 100%	✓

www.cloudskillsboost.google utilise des cookies Google afin de fournir ses services, d'en améliorer la qualité et d'analyser le trafic. [En savoir plus.](#)

J'ai compris

I couldn't do all of them because I have no credit 😞 (Thanks Efrei !)

LINK FOR THE GITHUB : <https://github.com/Sl33pyCl0ud/Software-Engineering-for-the-Cloud.git>