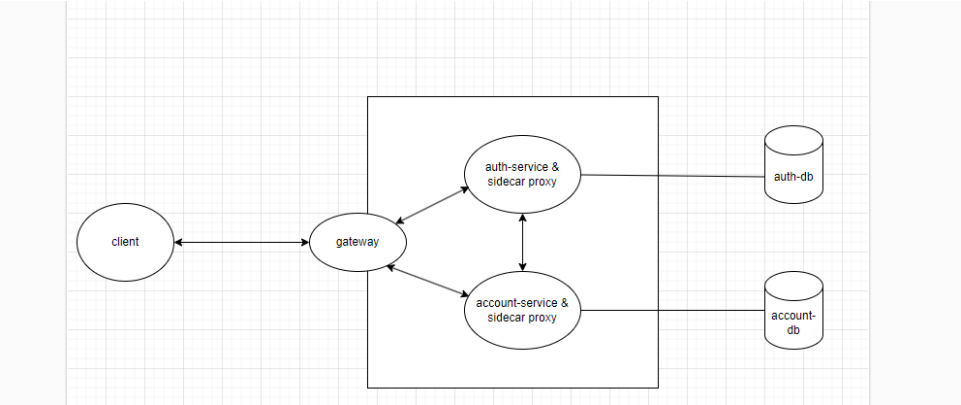
This project serves as an illustrative example of creating a basic authentication application using JWT tokens and one-time passwords (OTP) for enhanced security measures using microservices architecture with the service mesh.

Here is the designed architecture that we want to deploy. It composed of two services : auth and account. The account service is designed to manage the personal data of users. The auth service is designed to manage the authentification of users.



Each services has his own database following the “One database per service” pattern to Keep each microservice’s persistent data private to that service and accessible only via its API.

To add more control between services, we use the service mesh to facilitate communication, observability, and control between services in a microservices architecture. We inject a proxy in all services, a ‘sidecar proxy’ to control the flow of data in each services.

When clone the repository, on the k8s folder, you should see helm chart with subdependencies : auth, account and gateway.

On the root k8s folder, to deploy the infrastructure, type helm install ‘name-of-deployment' :

```
nasty@nasty ~/Documents/SimpleAuthJWT/k8s  dev  helm install deploy .
NAME: deploy
LAST DEPLOYED: Mon Jan 22 21:11:12 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
NOTES:
1. Get the application URL by running these commands:
  export POD_NAME=$(kubectl get pods --namespace default -l "app.kubernetes.io/name=simple-auth-deploy,app.kubernetes.io/instance=deploy" -o jsonpath="{.
items[0].metadata.name}")
  export CONTAINER_PORT=$(kubectl get pod --namespace default $POD_NAME -o jsonpath="{.spec.containers[0].ports[0].containerPort}")
  echo "Visit http://127.0.0.1:8080 to use your application"
  kubectl --namespace default port-forward $POD_NAME 8080:$CONTAINER_PORT
nasty@nasty ~/Documents/SimpleAuthJWT/k8s  dev  kubectl get pods
```

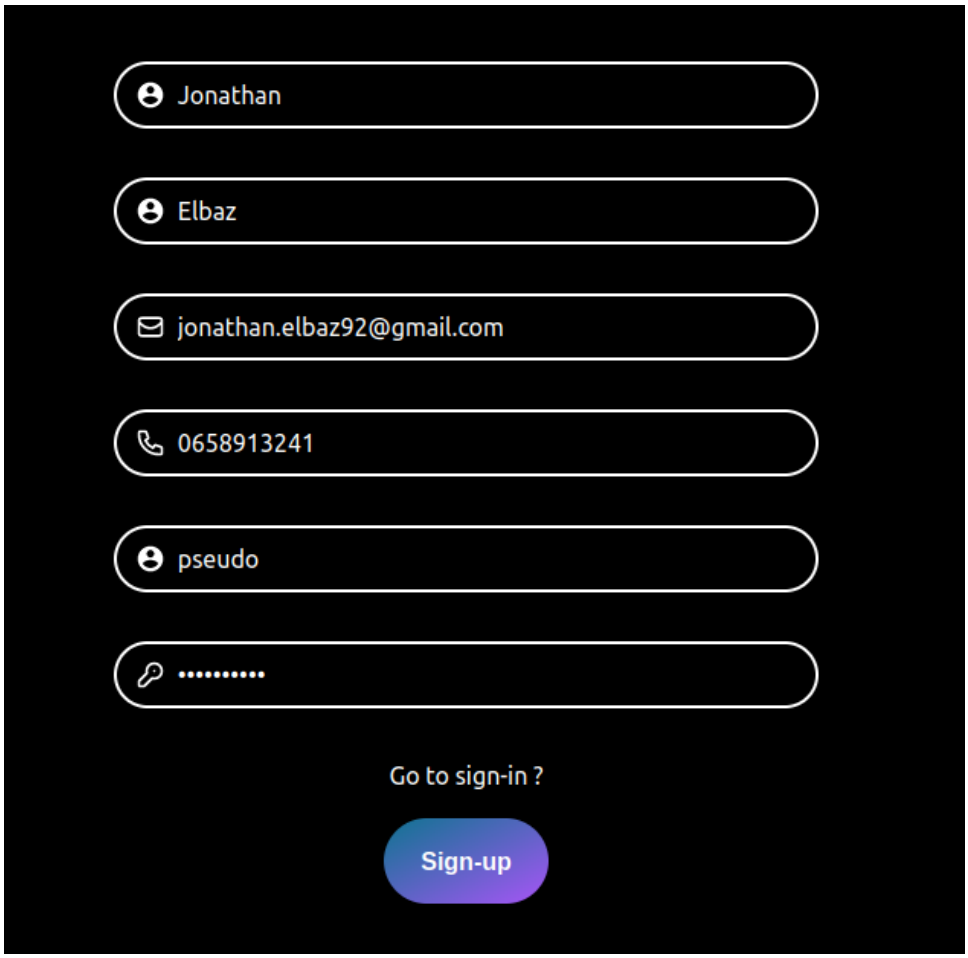
After that, wait some times to see the pods running with this command : kubectl get pods.

```
nasty@nasty ~/Documents/SimpleAuthJWT/k8s ⊿ dev kubectl get pods
NAME                                    READY   STATUS    RESTARTS          AGE
account-db-deployment-7b45bc54bc-p65mq  2/2     Running   0                 26s
account-deployment-6664847c6b-ghvlk     2/2     Running   0                 26s
auth-db-deployment-5988ccf996-jmpmn     2/2     Running   0                 26s
auth-deployment-98f6ddbfc-58rx9         2/2     Running   0                 26s
dnsutils                                1/1     Running   19 (4h20m ago)    44d
nasty@nasty ~/Documents/SimpleAuthJWT/k8s ⊿ dev
```

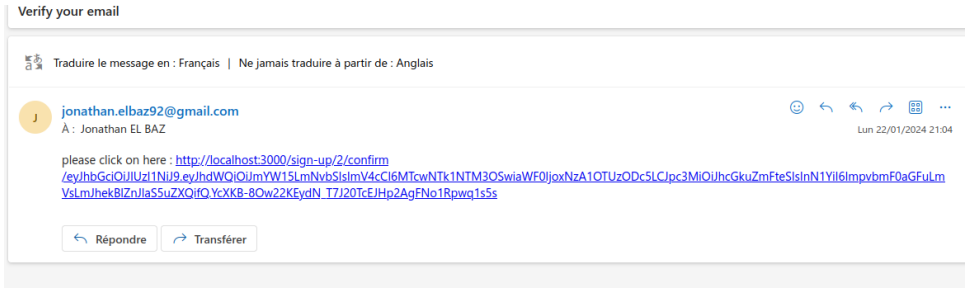
For now, the infrastructure is deployed and the pods containing services is running. To forwarding port of ingress and so, use it at localhost, you should see a bash script named ingress-forward.sh. This script containes the requirements to forward the port of the ingress. Just execute this script :

```
nasty@nasty ~/Documents/SimpleAuthJWT/k8s ⊿ dev sudo -E ./ingress-forward.sh
[sudo] password for nasty:
Open Ingress at http://localhost:80/
Forwarding from 127.0.0.1:80 -> 8080
Forwarding from [::1]:80 -> 8080
```

For now, the infrasctucture is deployed, running and binded to the localhost http port. To test it, I recommand you to run and build the client website. Just go to the client folder, type yarn, yarn build and finally yarn dev to run the website. You should arrive at sign-up page, just register your account and see what happens.

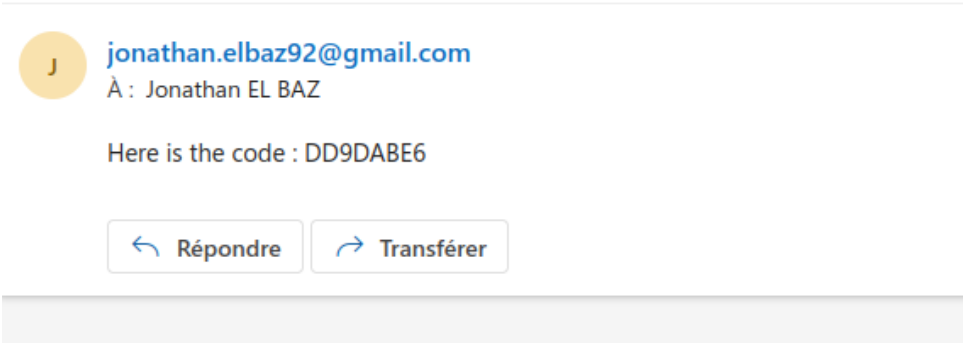


After that, like another webiste, an email is sent with a link, go to this link to be registered and verificated:



For now, you can sign-in, go to the /sign-in page and follow the instructions

You will receive a One Time password, just type it on the screen and you will be redirected to /connected url to illustrate the success of the app.



Links:

https://microservices.io/patterns/data/database-per-service.html