République Tunisienne

*****

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

****

Université de Monastir

*****

Institut Supérieur d'Informatique et de Mathématiques de Monastir

Rapport de projet Devops

Deuxième Année Cycle Ingénieur en Informatique

Spécialité :

Génie Logiciel

Elaboré par :

Khawla Grissa
Malak Letaief
Samar Mestiri

Déploiement d'une application web avec pipeline CI/CD en local

---

Encadré par : ***Mr. Lazhar HAMEL***
Mme. Dorra DHAOU

Année Universitaire : 2024/2025

Objectif du Projet

L'objectif principal de ce projet est de créer une application web simple, de la gérer avec un système de versioning (Git), de configurer une pipeline CI/CD pour l'automatisation des tests et de la mise à jour du code, puis de déployer l'application localement à l'aide de Docker.
Ce projet se concentre sur les aspects de configuration, intégration et déploiement, et non sur le développement fonctionnel de l'application.

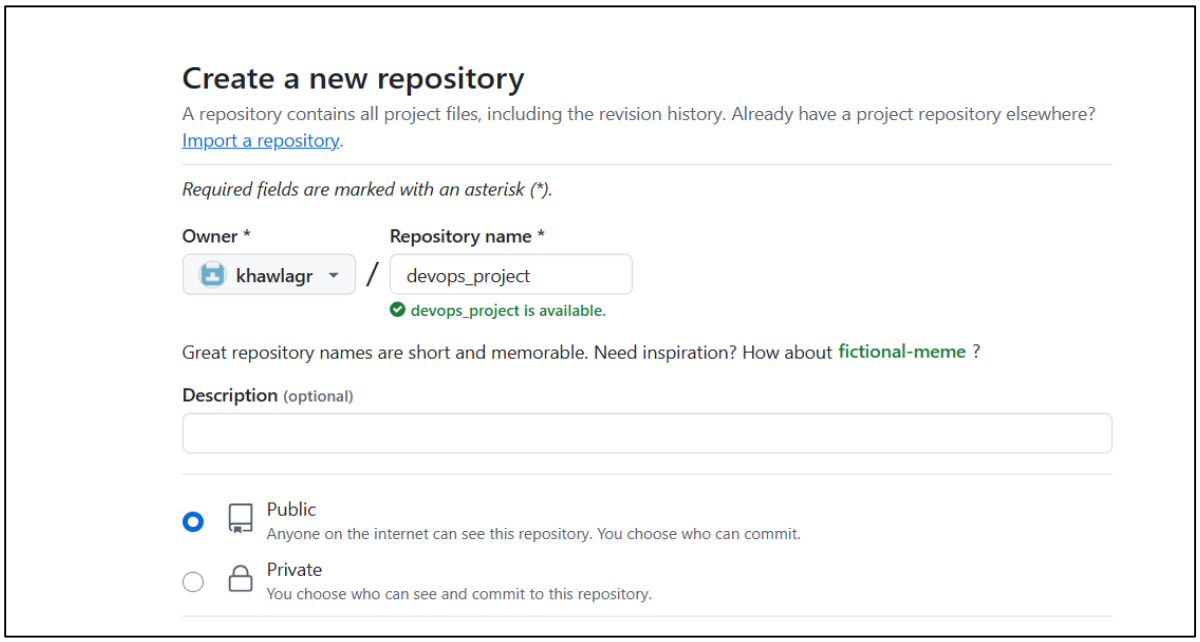
Outils et Environnement Utilisés

- **Express.js** (framework backend Node.js)
- **HTML/CSS, EJS** (frontend)
- **MongoDB** (base de données NoSQL)
- **Git et GitHub** (versionnage)
- **Docker** (conteneurisation)
- **GitHub Actions** (CI/CD)
- **Terraform** pour l'infrastructure-as-code

Étapes Réalisées

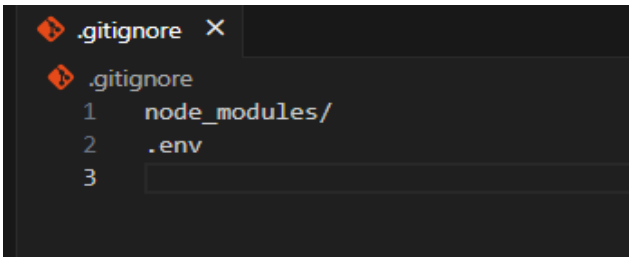
1. Initialisation du Version Control

   Création d'un dépôt GitHub nommé `devops-project`.



Clonage local : git clone https://github.com/khawlagr/devops-project.git

Ajout d'un fichier **.gitignore**



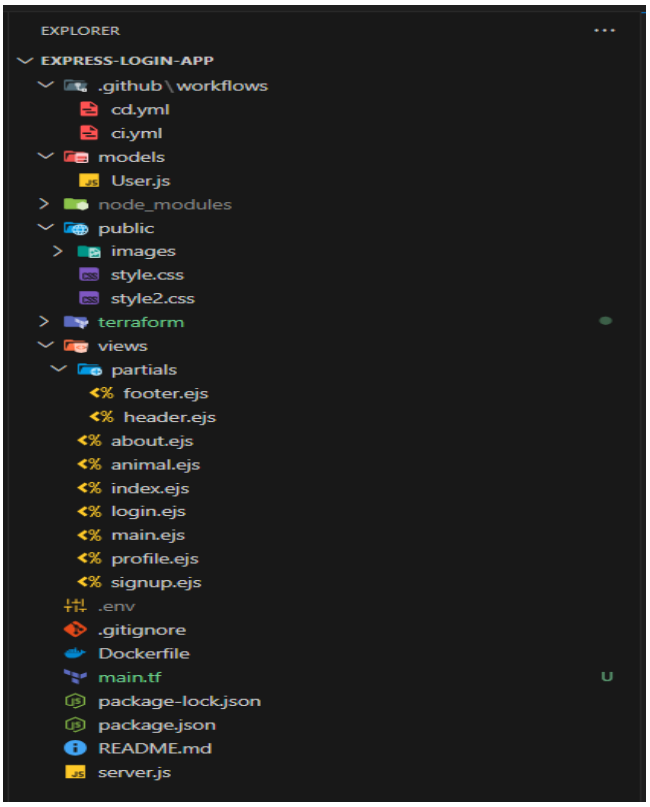
Commit et push initial :

```
git add .
git commit -m "Initial commit"
git push origin main
```

2. Développement de l'Application

Structure 3 couches :

- **Model** : Connexion MongoDB
- **View** : Template EJS
- **Controller** : Express.js



index.ejs

server.js | index.ejs

views > index.ejs > ?

```ejs
<%- include('./partials/header') %>

<section class="hero">
  <div class="hero-content">
    <h1>Discover the Amazing World of Animals</h1>
    <p>Learn about different species from around the globe</p>
  </div>
</section>

<section class="animal-grid">
  <h2>Featured Animals</h2>

  <div class="animals">
    <% animals.forEach(animal => { %>
      <div class="animal-card">
        <img src="<%= animal.image %>" alt="<%= animal.name %>">
        <div class="animal-info">
          <h3><%= animal.name %></h3>
          <p class="species"><%= animal.species %></p>
          <p><%= animal.description.substring(0, 100) %>...</p>
          <a href="/animal/<%= animal.id %>" class="btn">Learn More</a>
        </div>
      </div>
    <% }); %>
  </div>
</section>

<%- include('./partials/footer') %>
```

Server.js

server.js

server.js > ...

```js
require('dotenv').config();
const express = require('express');
const mongoose = require('mongoose');
const session = require('express-session');
const bcrypt = require('bcrypt');
const User = require('./models/User');
const path=require('path');

const app = express();

app.set('view engine', 'ejs');
app.use(express.static('public'));
app.use(express.urlencoded({ extended: true }));

app.use(session({
  secret: process.env.SESSION_SECRET,
  resave: false,
  saveUninitialized: true
}));

mongoose.connect(process.env.MONGO_URI)
  .then(() => console.log("MongoDB connected"))
  .catch(err => console.error(err));


// Accueil
app.get('/', (req, res) => {
  res.render('main', { user: req.session.user });
});

```

Installer **nodemon :**

```
PS C:\express-login-app> npm install --save-dev nodemon
>>

up to date, audited 293 packages in 2s

51 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\express-login-app>
```

Modifier le fichier **package.json** :

package.json

```
{
  "name": "express-login-app",
  "version": "1.0.0",
  "main": "server.js",
  Debug
  "scripts": {
    "dev": "nodemon server.js"
  },
  "dependencies": {
    "bcrypt": "^5.1.0",
    "dotenv": "^16.0.3",
    "ejs": "^3.1.9",
    "express": "^4.18.2",
    "express-session": "^1.17.3",
    "mongodb": "^6.16.0",
    "mongoose": "^7.2.0"
  },
  "devDependencies": {
    "mocha": "^11.2.2",
    "nodemon": "^3.1.10"
  }
}
```

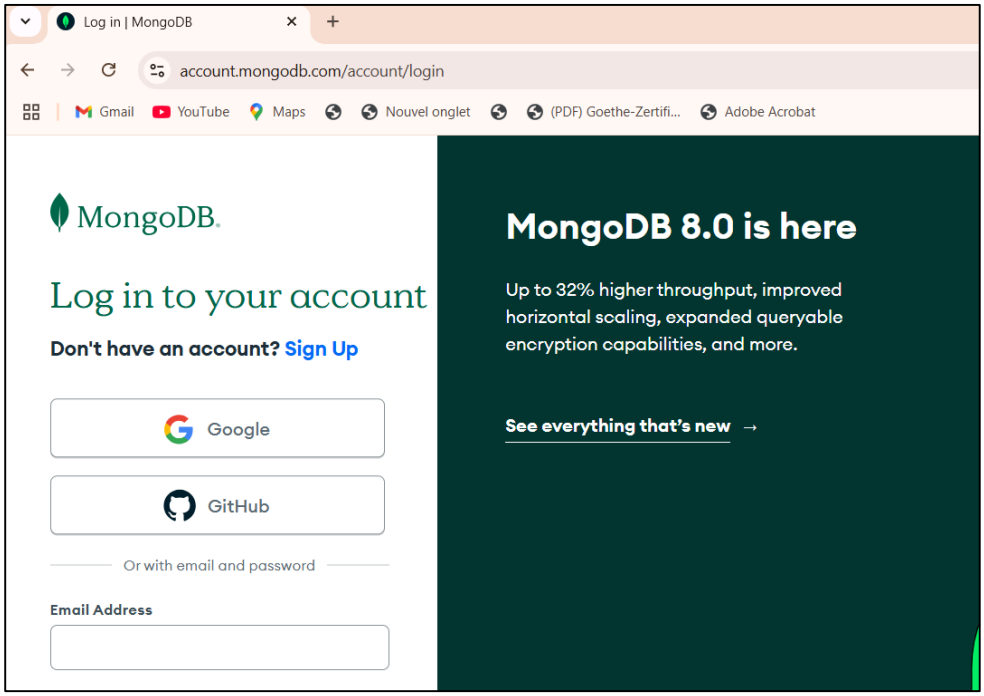
Lancer le projet en mode développement :

Accéder à : http://localhost:3000

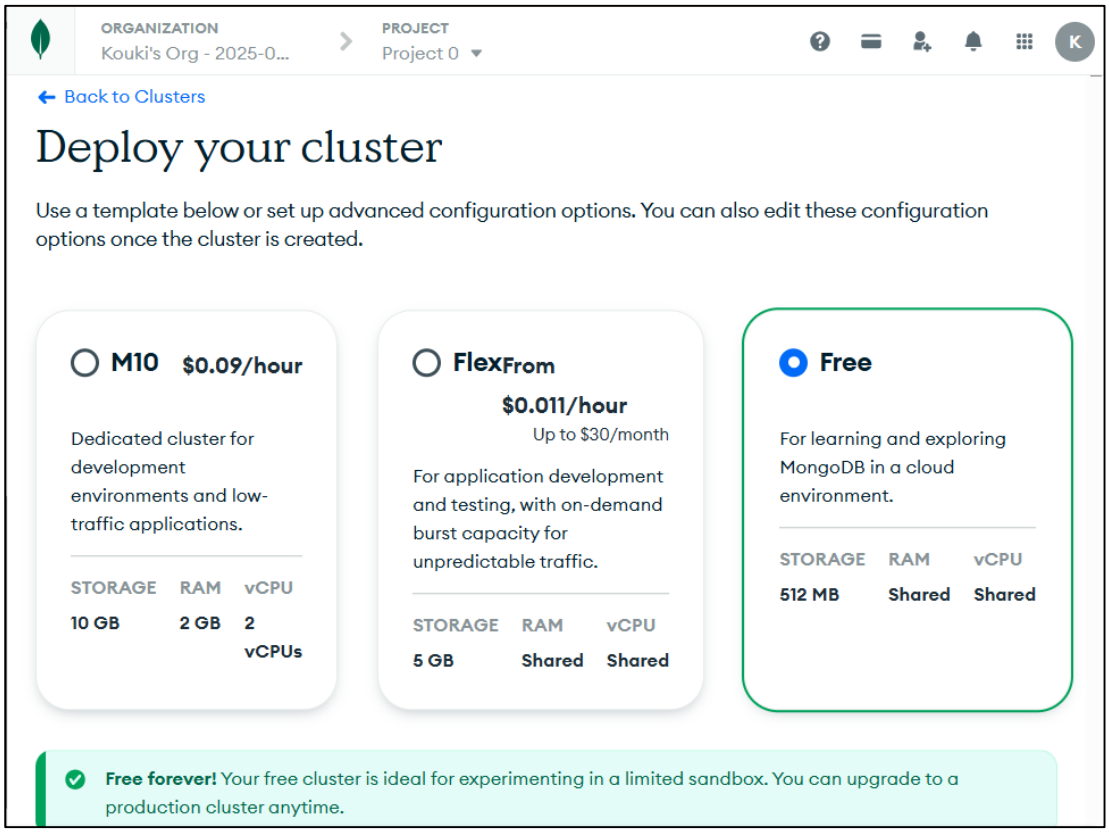
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\express-login-app> npm run dev

> express-login-app@1.0.0 dev
> nodemon server.js

[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
(node:18756) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
http://localhost:3000
MongoDB connected
```

Installation MongoDB

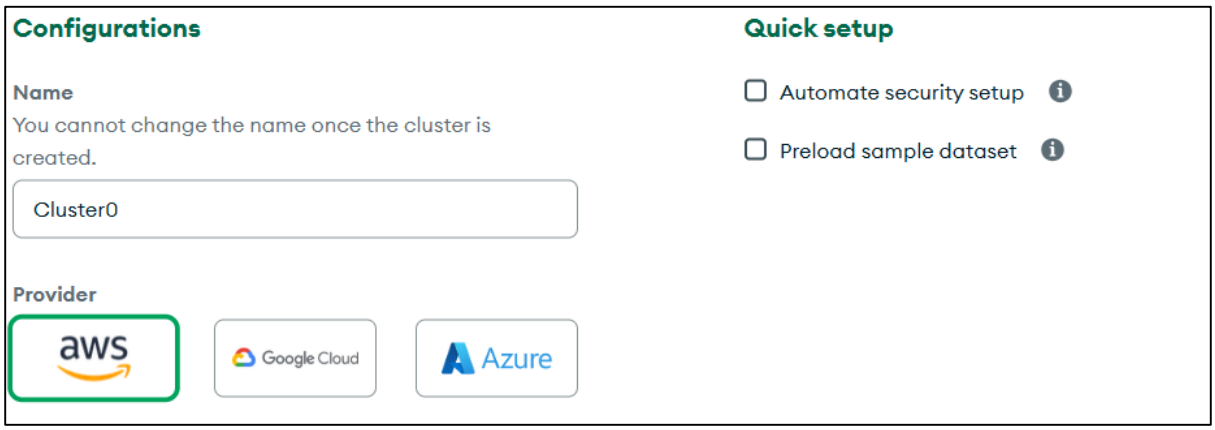
Créer un compte sur https://www.mongodb.com/cloud/atlas



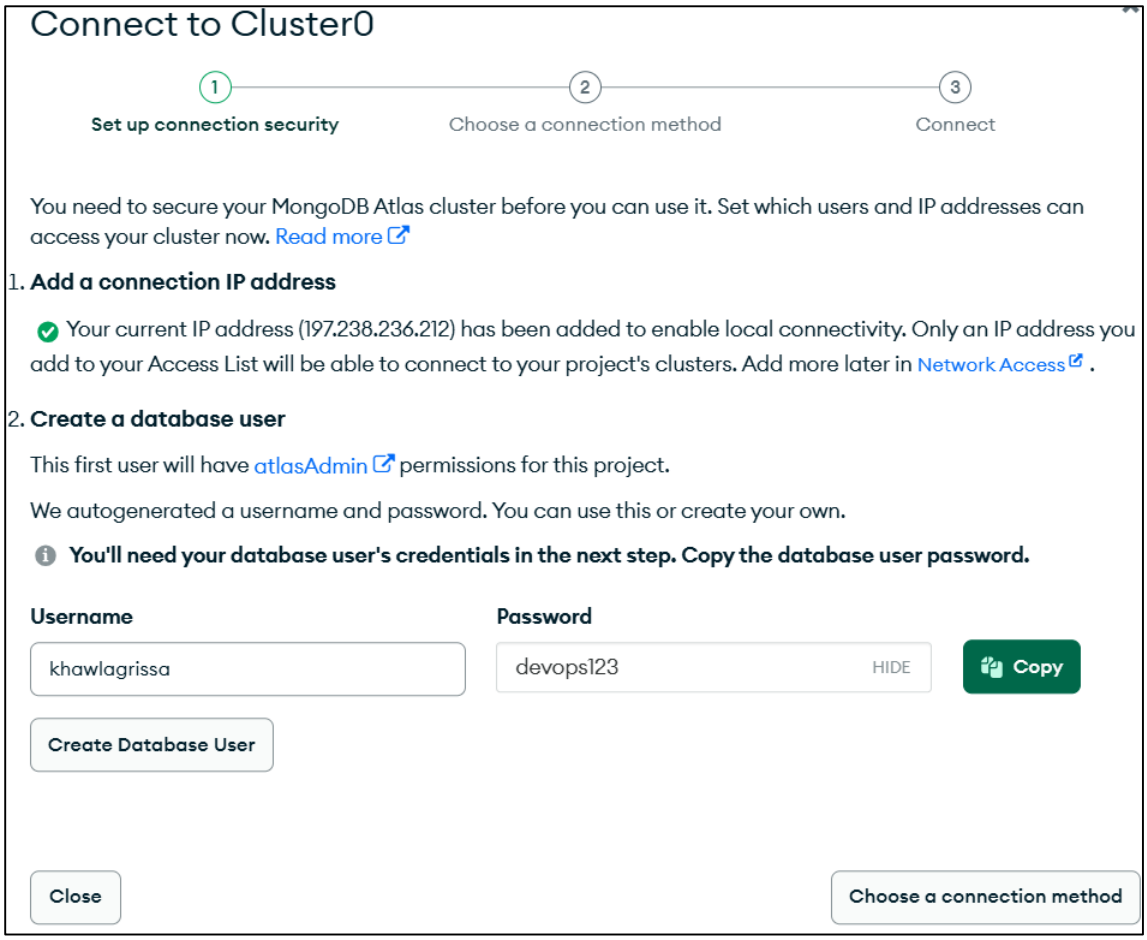
Créer un **cluster gratuit**



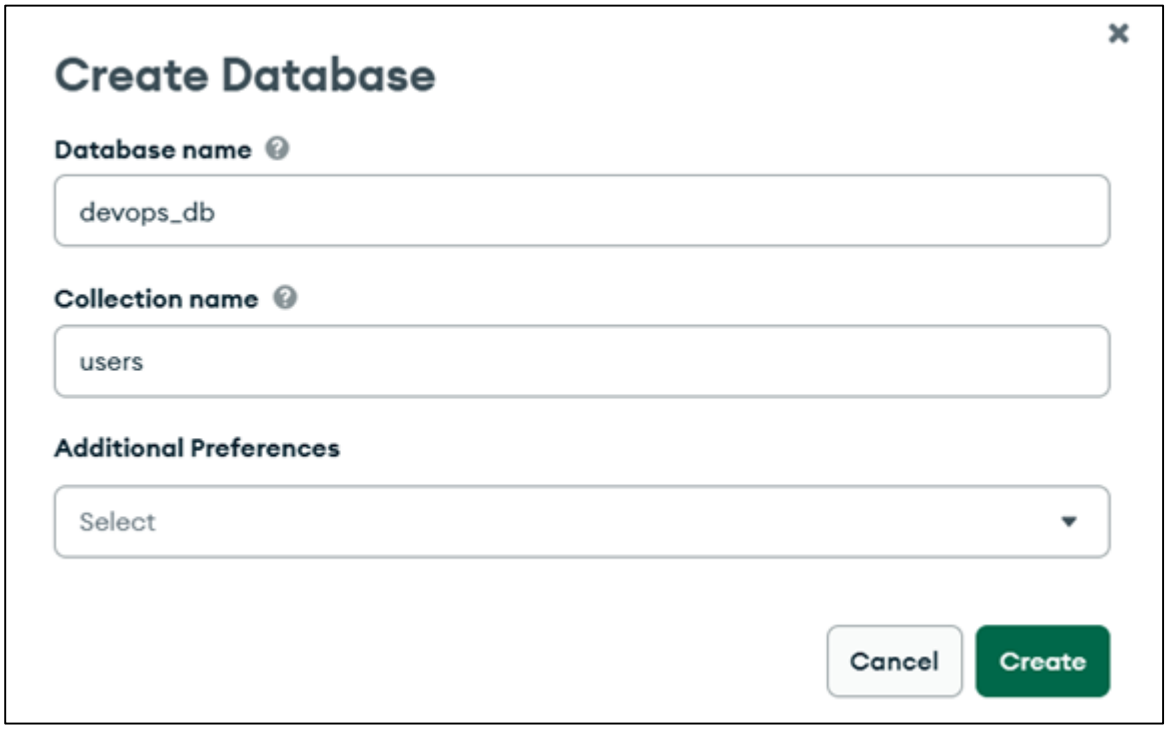
Sélectionner le fournisseur cloud : **AWS**
Nommer le cluster : **Cluster0**

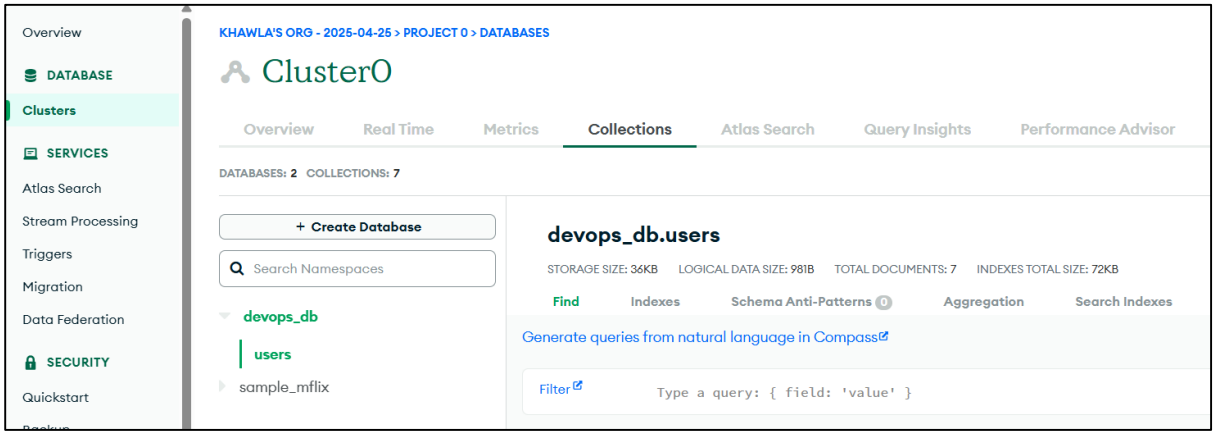


Créer un utilisateur pour la base



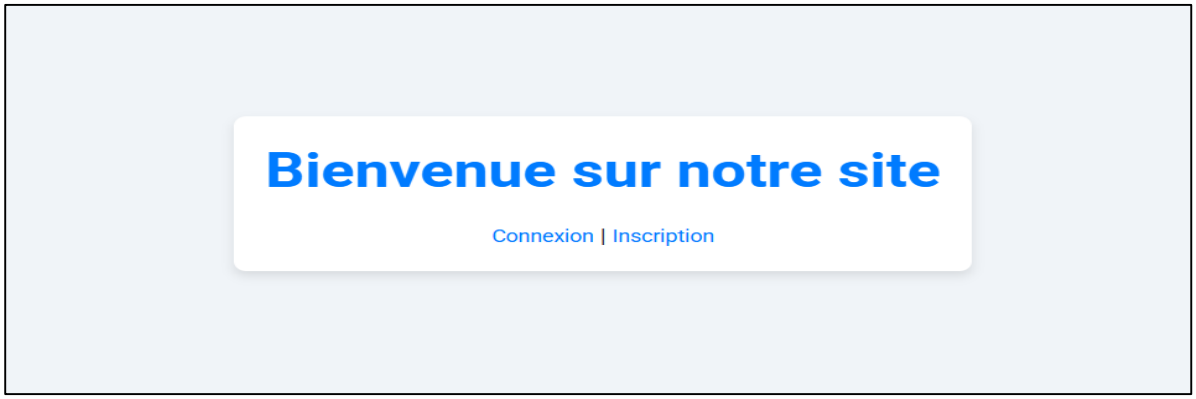
Création d'une base de données **devops_db**





Application :

Page d'accueil

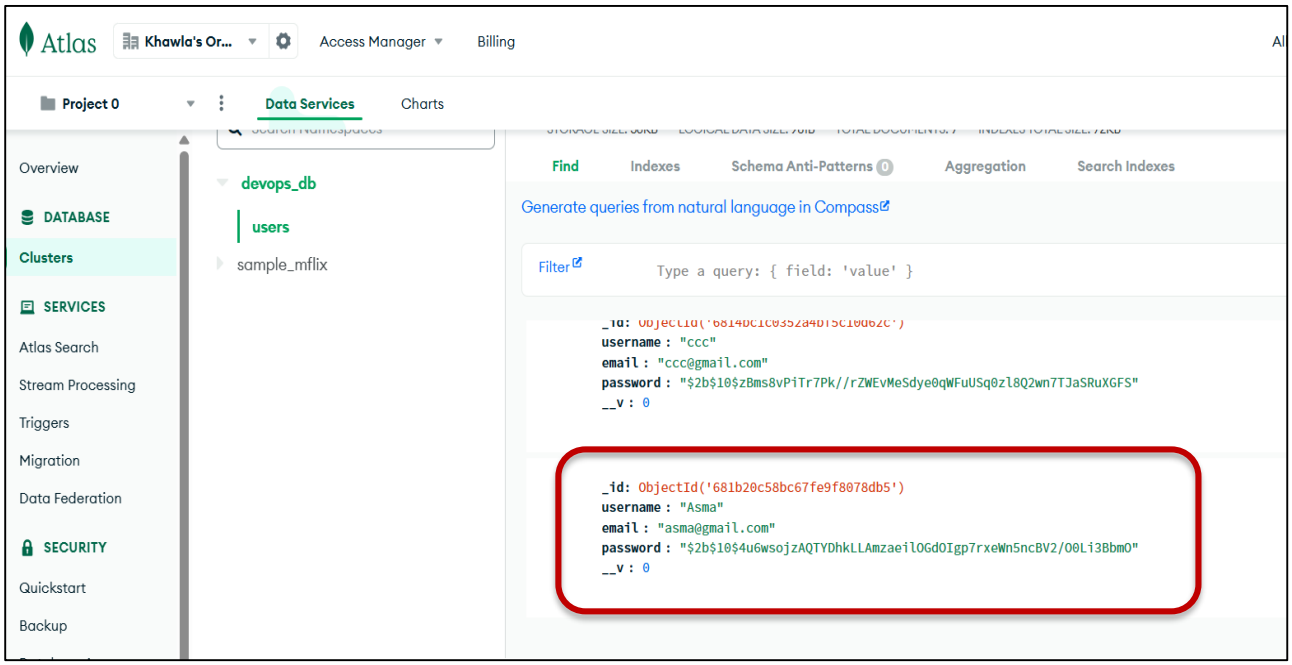


Page d’inscription

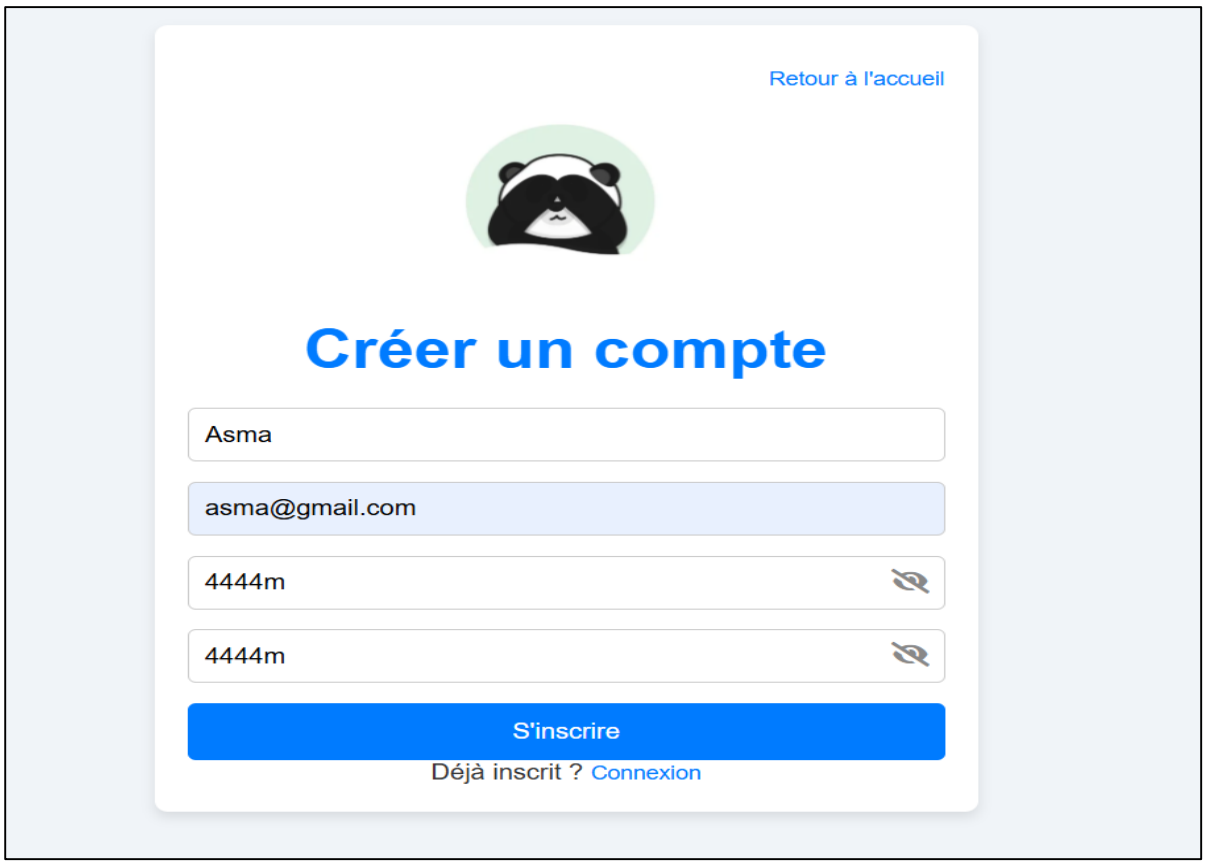
Création d'un nouveau compte

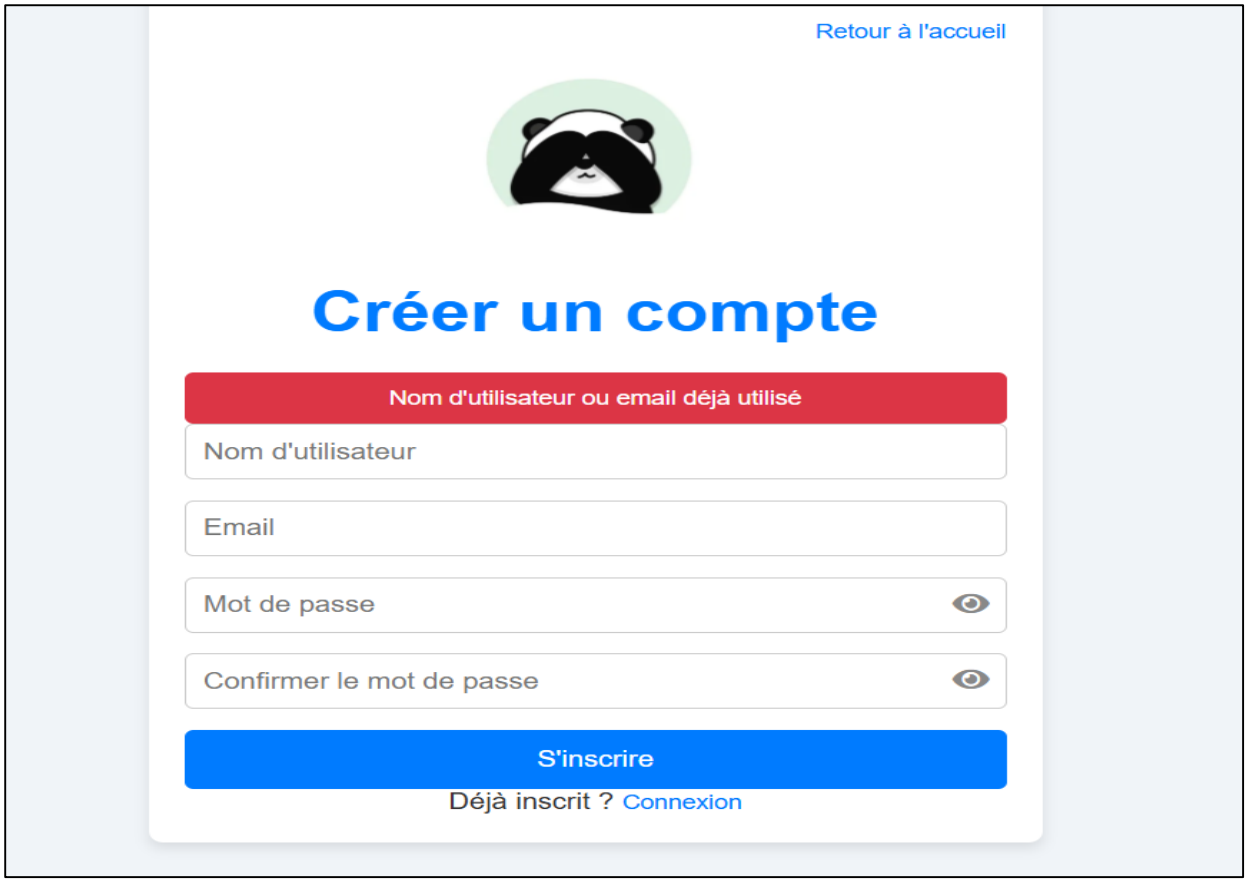
Étude de cas : Création d’un nouveau compte Asma

Utilisateur ajouté à la base de données

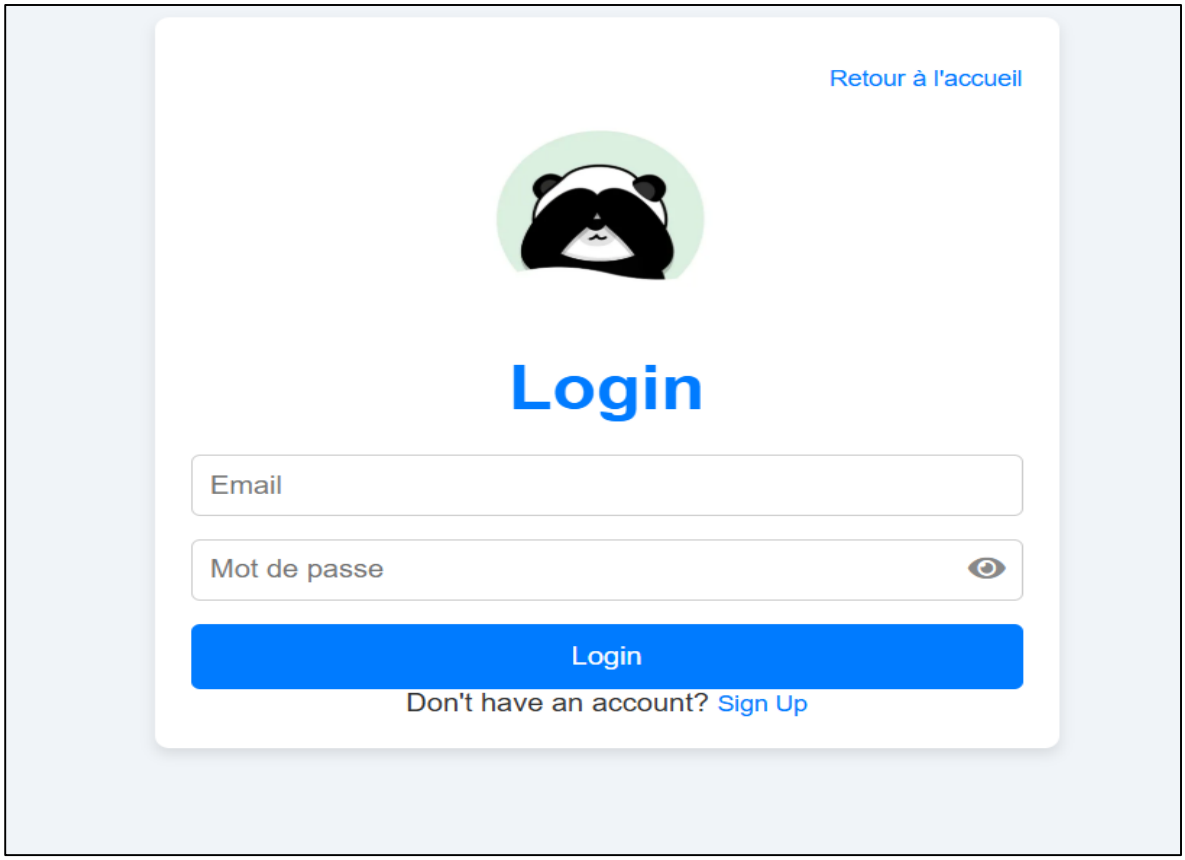


Un compte avec cette adresse e-mail existe déjà.

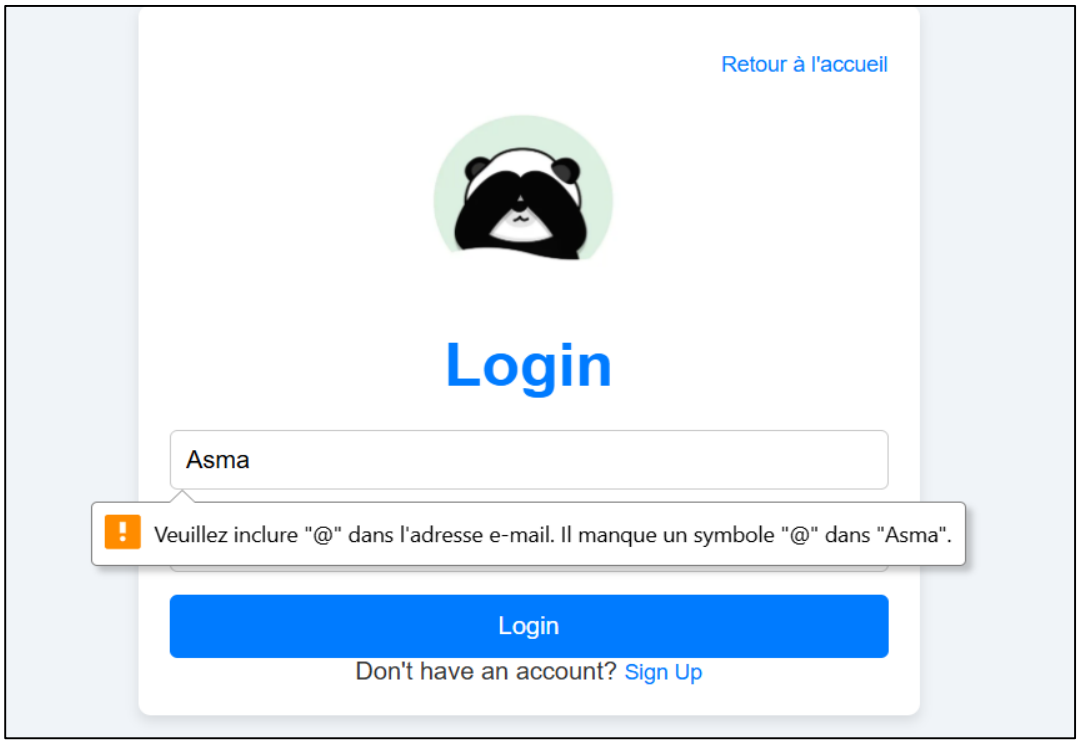




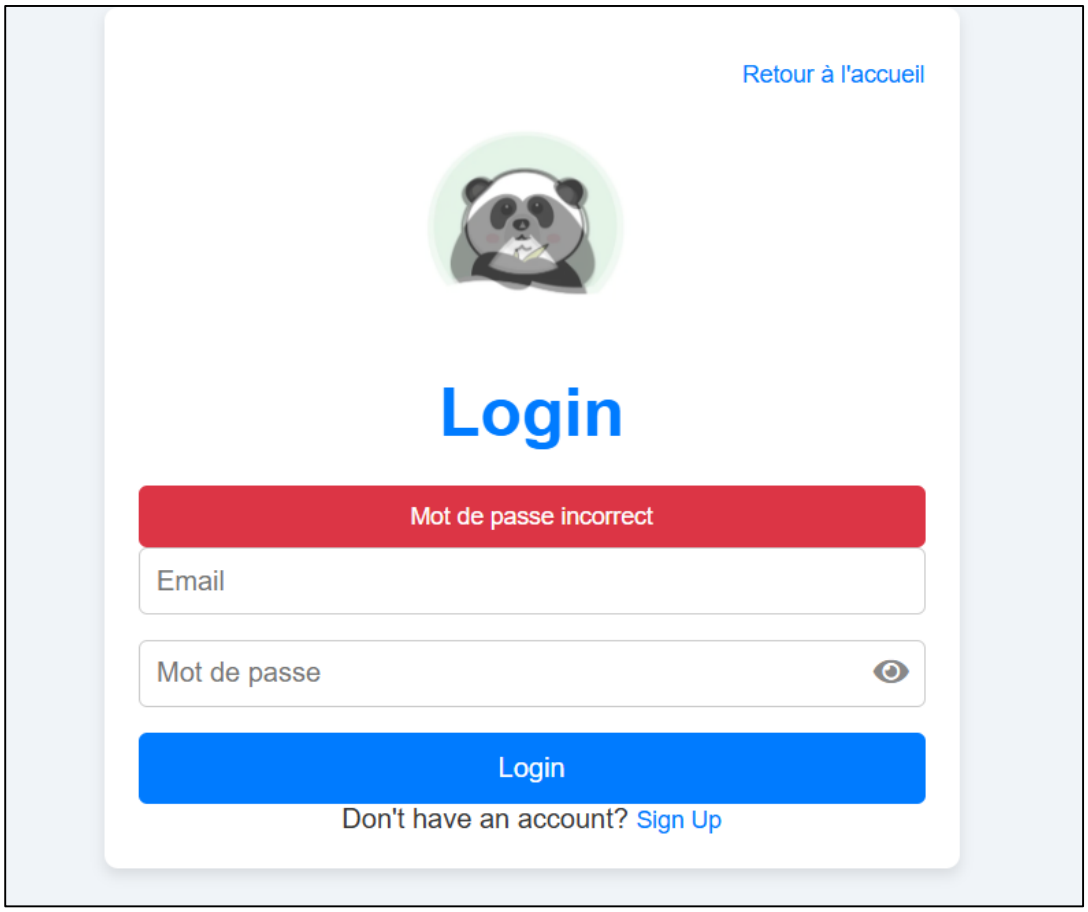
Page de connexion



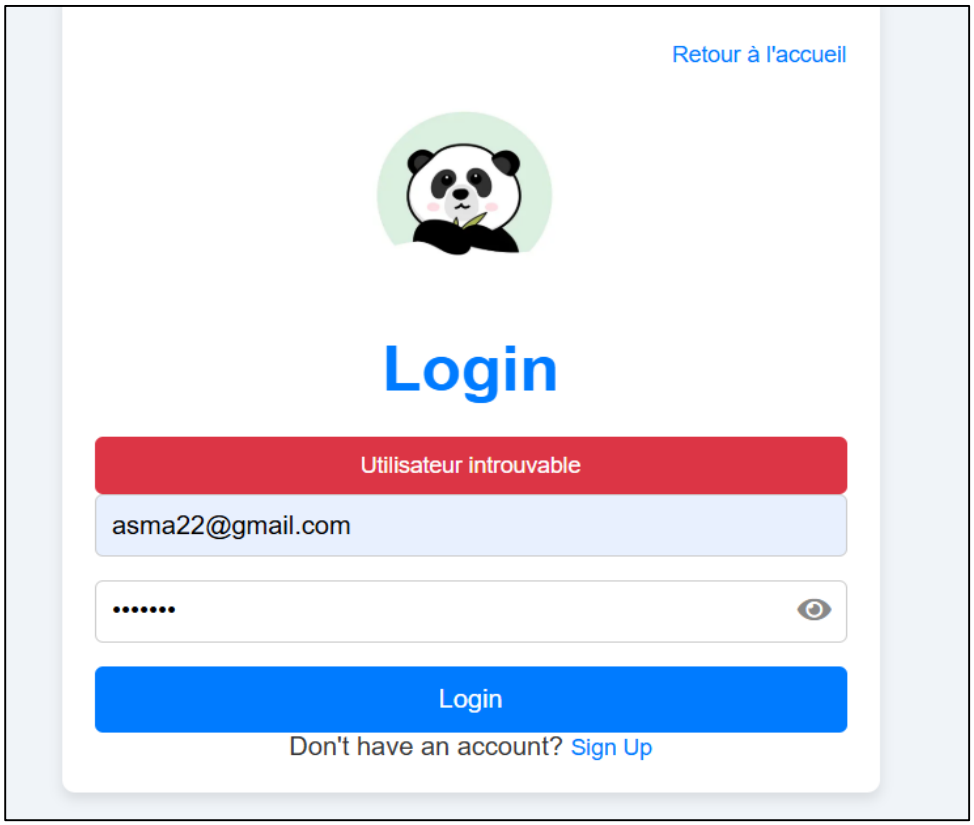
Cas 1 : Adresse e-mail invalide. Veuillez entrer une adresse valide.



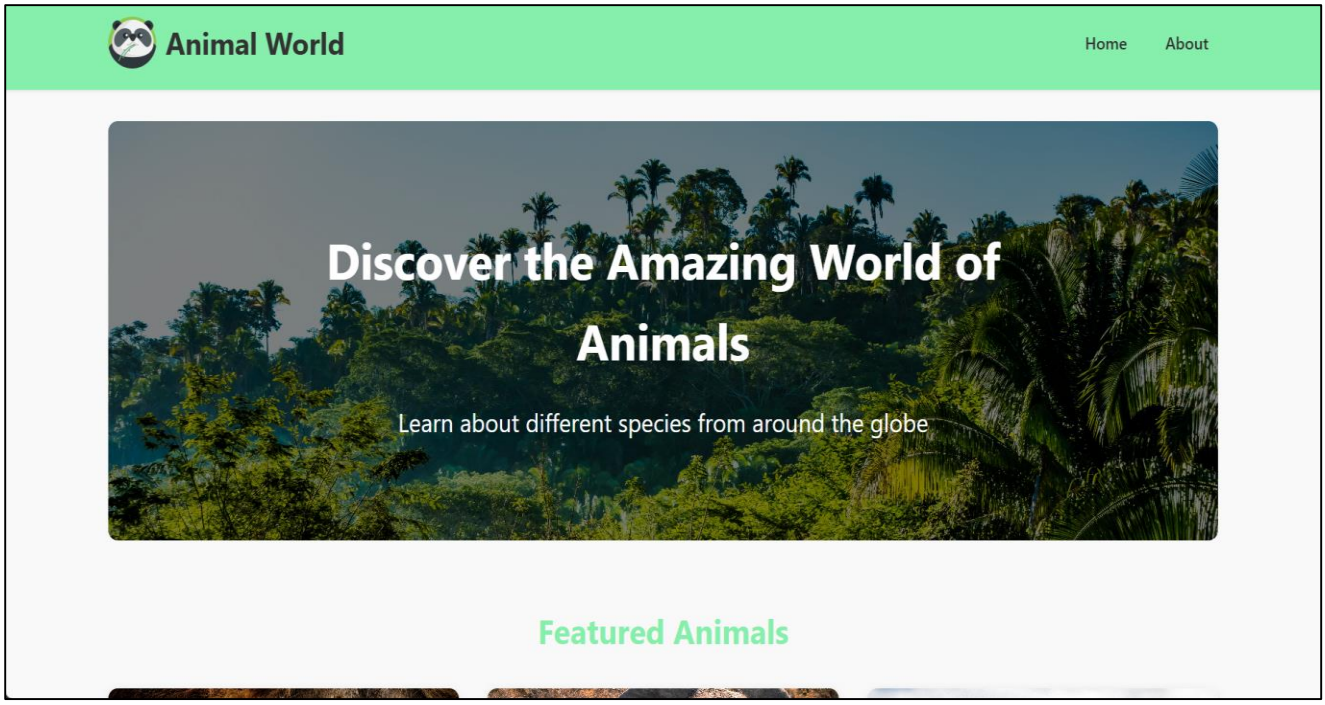
Cas 2 : Mot de passe incorrect. Veuillez réessayer



Cas 3 : Utilisateur introuvable. Veuillez vérifier vos informations de connexion.

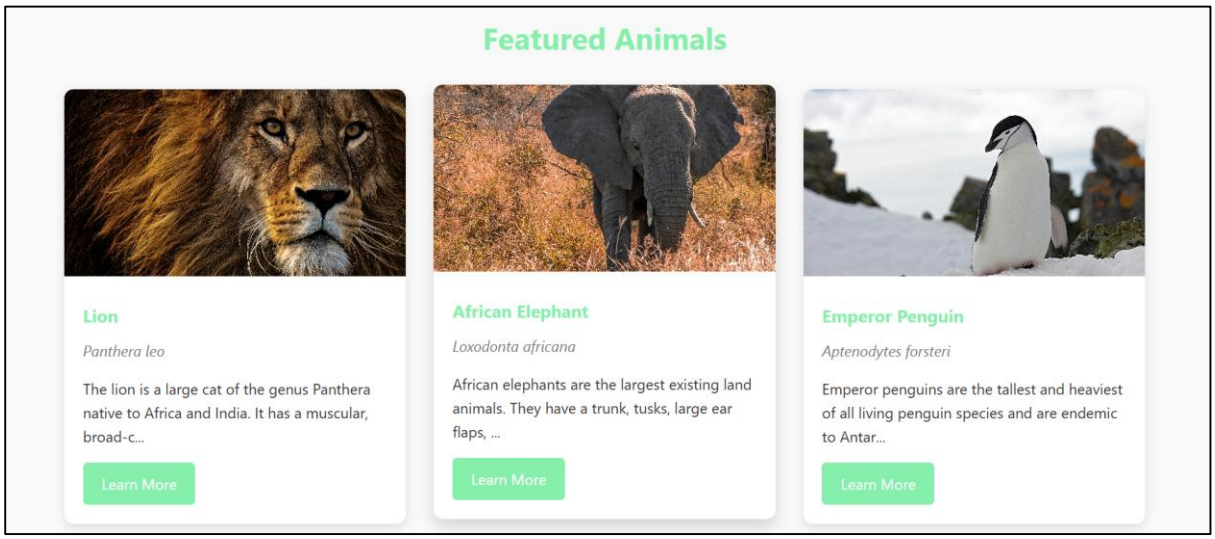


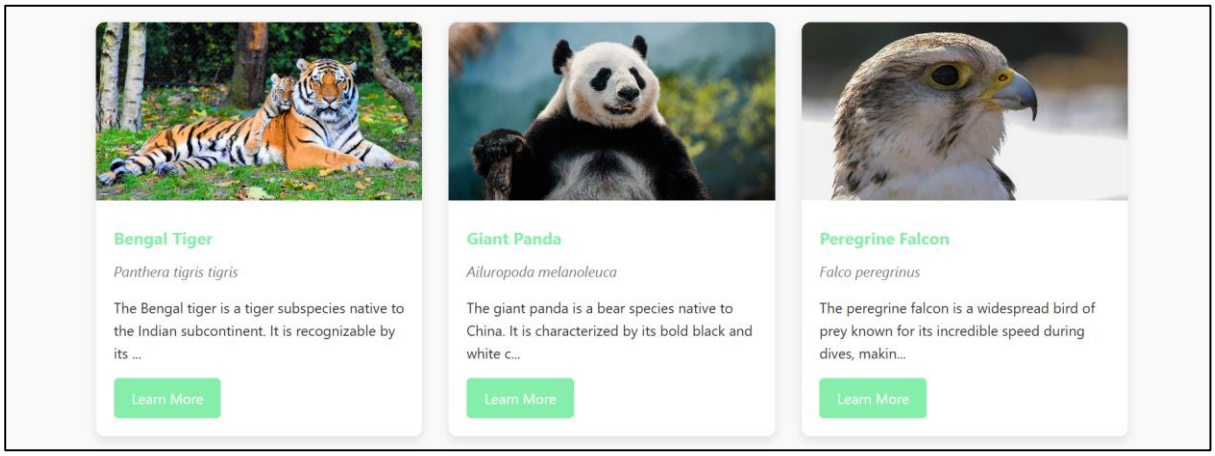
Cas 4 : Compte ouvert avec succès. Bienvenue dans votre espace **"Animals World".**

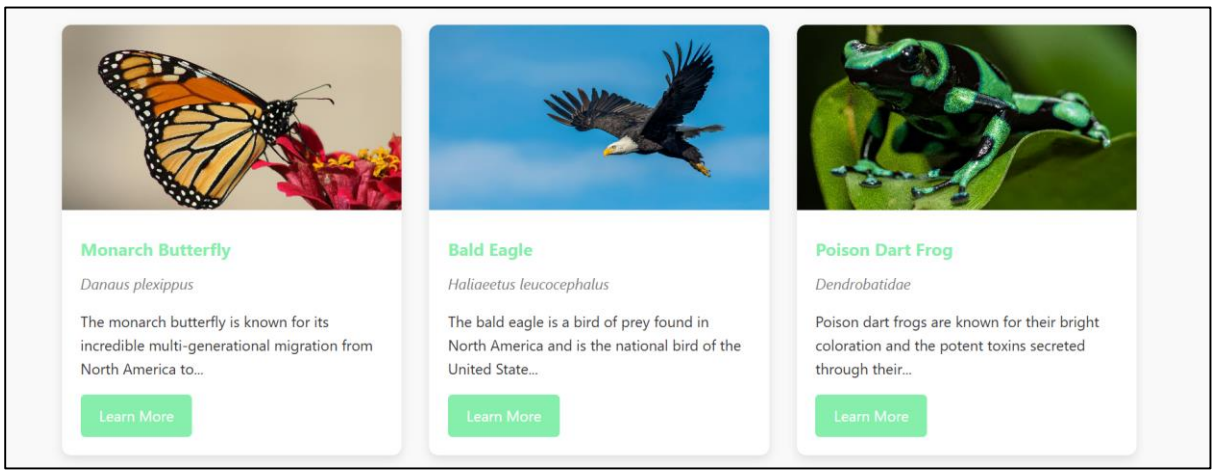


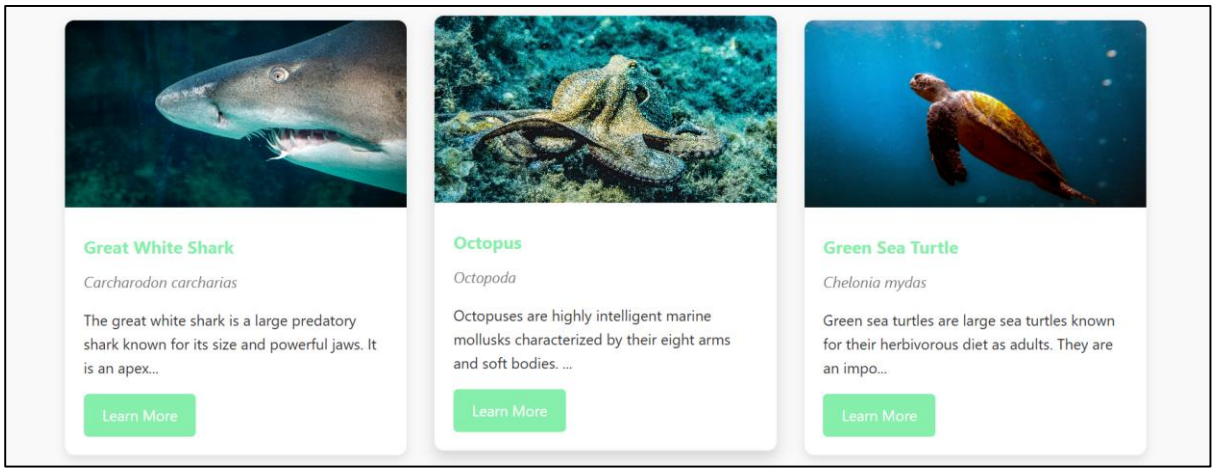
Le concept de notre site web est de permettre aux utilisateurs de **découvrir différents animaux** à travers une interface simple et interactive.
Le site propose plusieurs fonctionnalités, notamment :

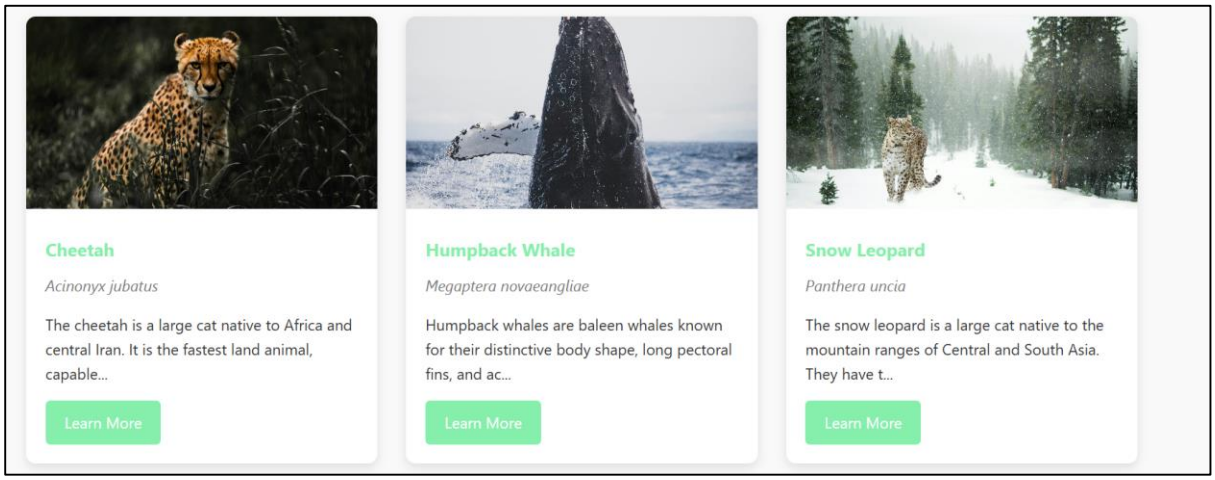
- Un bouton **"Learn more"** pour en savoir plus sur chaque animal.
- Deux boutons de navigation en haut de la page : **"About"** (pour en apprendre davantage sur le site) et **"Home"** (pour revenir à la page d'accueil).

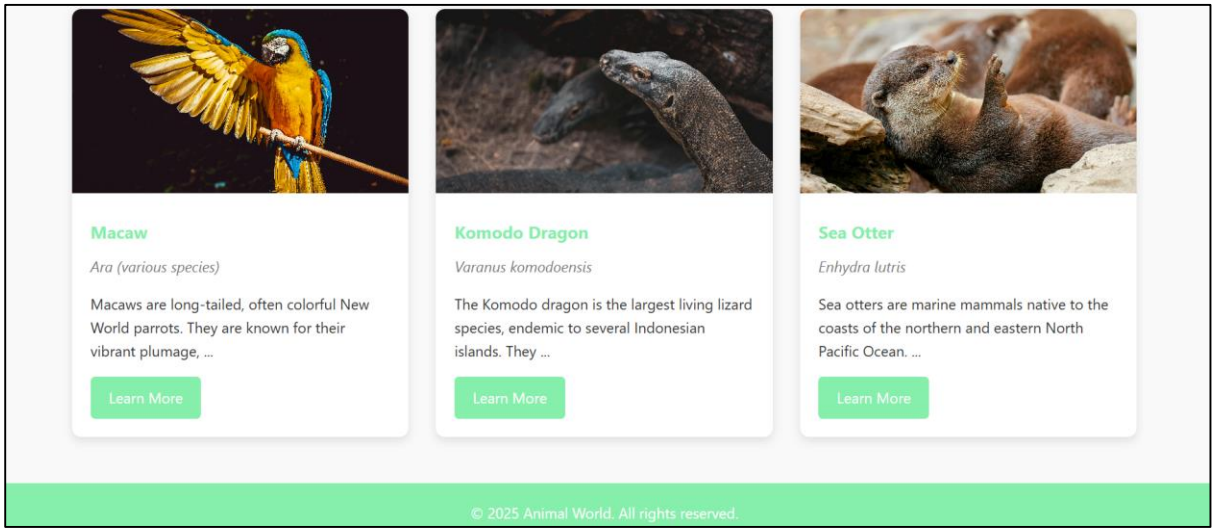







Great White Shark
Carcharodon carcharias
The great white shark is a large predatory shark known for its size and powerful jaws. It is an apex...
Learn More
Octopus
Octopoda
Octopuses are highly intelligent marine mollusks characterized by their eight arms and soft bodies. ...
Learn More
Green Sea Turtle
Chelonia mydas
Green sea turtles are large sea turtles known for their herbivorous diet as adults. They are an impo...
Learn More


Cheetah
Acinonyx jubatus
The cheetah is a large cat native to Africa and central Iran. It is the fastest land animal, capable...
Learn More
Humpback Whale
Megaptera novaeangliae
Humpback whales are baleen whales known for their distinctive body shape, long pectoral fins, and ac...
Learn More
Snow Leopard
Panthera uncia
The snow leopard is a large cat native to the mountain ranges of Central and South Asia. They have t...
Learn More


Macaw
Ara (various species)
Macaws are long-tailed, often colorful New World parrots. They are known for their vibrant plumage, ...
Learn More
Komodo Dragon
Varanus komodoensis
The Komodo dragon is the largest living lizard species, endemic to several Indonesian islands. They ...
Learn More
Sea Otter
Enhydra lutris
Sea otters are marine mammals native to the coasts of the northern and eastern North Pacific Ocean. ...
Learn More
© 2025 Animal World. All rights reserved.

Lion

Panthera leo

Classification

Category: Mammal

Species: Panthera leo

Habitat

Savanna, grasslands

About

The lion is a large cat of the genus Panthera native to Africa and India. It has a muscular, broad-chested body, short, rounded head, round ears, and a hairy tuft at the end of its tail. Known for their social structure called prides.

Back to All Animals

About Animal World

Our Mission

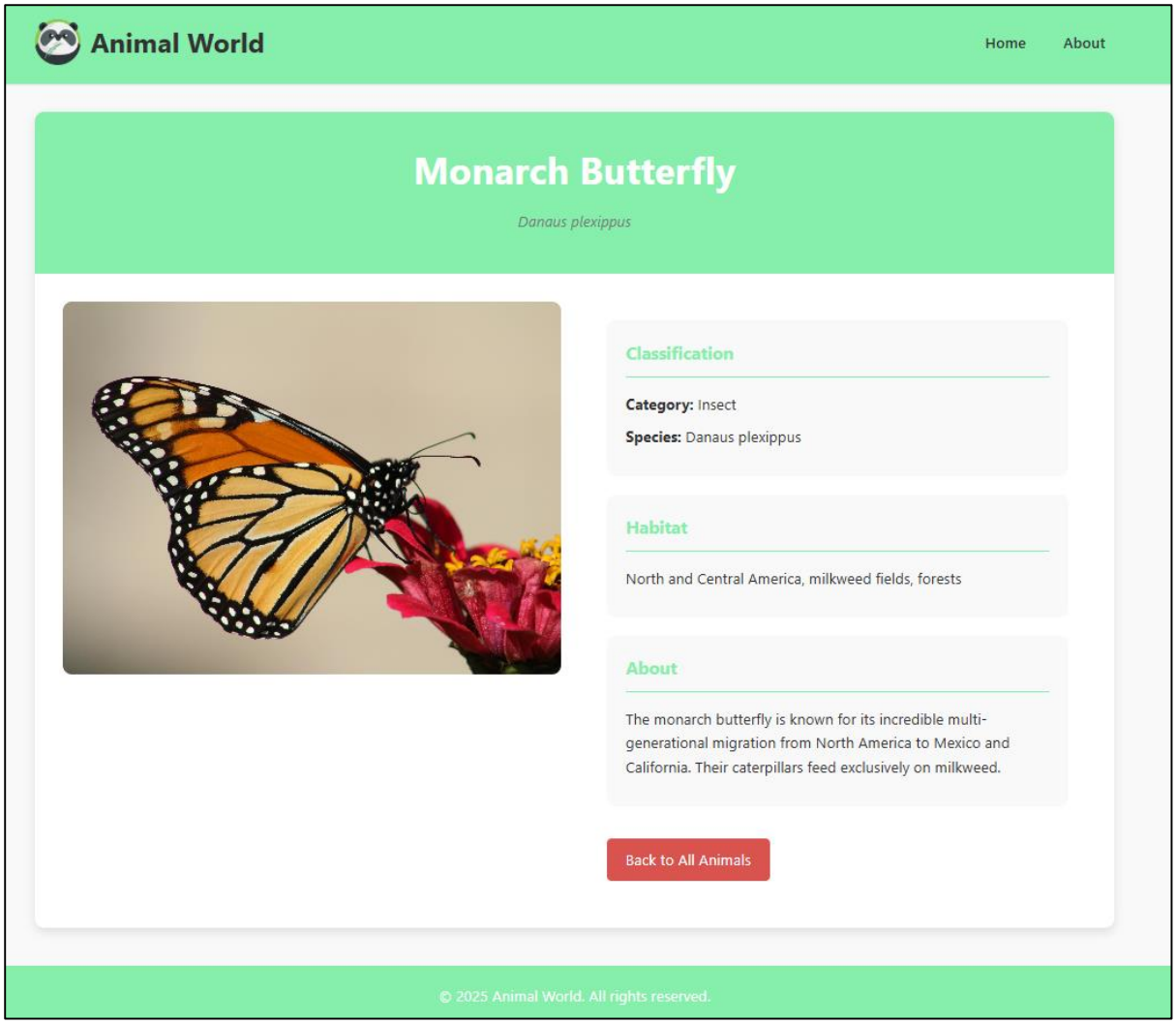
Animal World is dedicated to educating people about the diverse wildlife that inhabits our planet. Our mission is to promote understanding and appreciation of animals from all ecosystems, fostering a sense of responsibility toward wildlife conservation.

What We Do

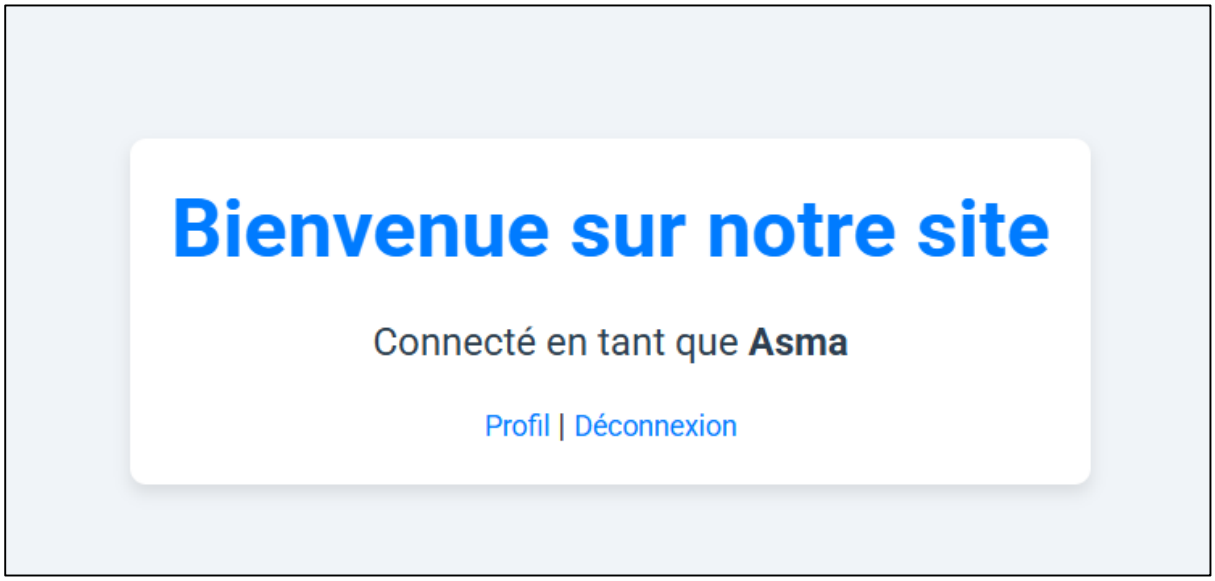
We collect and present information about various animal species, their habitats, behaviors, and conservation status. Our team of experts works diligently to ensure that all information is accurate, up-to-date, and presented in an engaging manner.

Join Us

Whether you're a wildlife enthusiast, a student, or simply curious about the animal kingdom, Animal World provides valuable resources to expand your knowledge. Explore our collection of animal profiles and discover the wonders of wildlife.



Un **lien "Déconnexion"** est disponible pour permettre à l'utilisateur de quitter sa session. Lorsqu'il clique sur ce lien, il est redirigé vers la page de connexion ou la page d'accueil publique, selon la configuration du site.



3. Containeriser l'application

Préparer un Dockerfile pour Express

Dockerfile

Fichier Modifier Affichage

```
FROM node:14

RUN apt-get update && apt-get install -y \
  build-essential \
  python3 \
  && rm -rf /var/lib/apt/lists/*

WORKDIR /app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3000

CMD ["node", "server.js"]
```

Construire l'image Docker : **docker build -t myapp .**

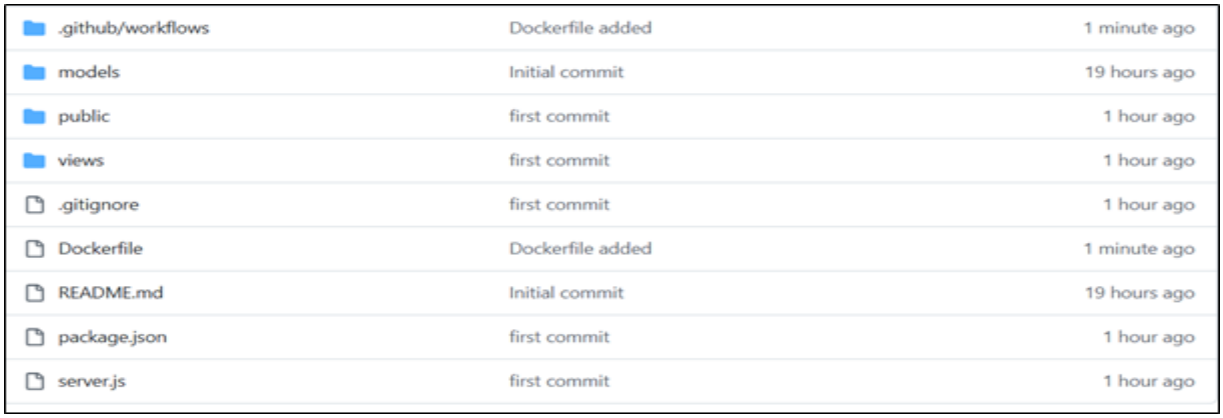
```
C:\express-login-app>docker build -t myapp .
[+] Building 855.6s (10/10) FINISHED                                                    docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                  0.1s
 => => transferring dockerfile: 174B                                                                  0.0s
 => [internal] load metadata for docker.io/library/node:14                                            3.5s
 => [internal] load .dockerignore                                                                     0.2s
 => => transferring context: 2B                                                                       0.0s
 => [1/5] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461a  783.5s
 => => resolve docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa    0.2s
 => => sha256:0d27a8e86132900757 4c6766fba946d48e20d2c8e964e873de352603f22c4ceb 450B / 450B                0.3s
 => => sha256:0c8cc2f24a4dcb64e602e086fc9446b0a541e8acd9ad72d2e90df3ba22f158b3 2.29MB / 2.29MB            19.5s
 => => sha256:6f51ee005deac0d99898e41b8ce60ebf250ebe1a31a0b03f613aec6bbc9b83d8 4.19kB / 4.19kB            0.9s
 => => sha256:5f32ed3c3f278edda4fc571c880b5277355a29ae8f52b52cdf865f058378a590 35.24MB / 35.24MB          324.8s
 => => sha256:d9a8df5894511ce28a05e2925a75e8a4acbd0634c39ad734fdfba8e23d1b1569 191.85MB / 191.85MB       765.7s
 => => sha256:1de76e268b103d05fa8960e0f77951ff54b912b63429c34f5d6adfd09f5f9ee2 51.88MB / 51.88MB         397.6s
 => => sha256:3d2201bd995cccf12851a50820de03d34a17011dcbb9ac9fdf3a50c952cbb131 10.00MB / 10.00MB          84.8s
 => => sha256:b253aeafeaa7e0671bb60008df01de101a38a045ff7bc656e3b0fbfc7c05cca5 7.86MB / 7.86MB            81.9s
 => => sha256:2ff1d7c41c74a25258bfa6f0b8adb0a727f84518f55f65ca845ebc747976c408 50.45MB / 50.45MB          313.0s
 => => extracting sha256:2ff1d7c41c74a25258bfa6f0b8adb0a727f84518f55f65ca845ebc747976c408                  2.8s
 => => extracting sha256:b253aeafeaa7e0671bb60008df01de101a38a045ff7bc656e3b0fbfc7c05cca5                  0.4s
 => => extracting sha256:3d2201bd995cccf12851a50820de03d34a17011dcbb9ac9fdf3a50c952cbb131                  0.3s
 => => extracting sha256:1de76e268b103d05fa8960e0f77951ff54b912b63429c34f5d6adfd09f5f9ee2                  3.5s
 => => extracting sha256:d9a8df5894511ce28a05e2925a75e8a4acbd0634c39ad734fdfba8e23d1b1569                  9.8s
 => => extracting sha256:6f51ee005deac0d99898e41b8ce60ebf250ebe1a31a0b03f613aec6bbc9b83d8                  0.1s
 => => extracting sha256:5f32ed3c3f278edda4fc571c880b5277355a29ae8f52b52cdf865f058378a590                  5.7s
 => => extracting sha256:0c8cc2f24a4dcb64e602e086fc9446b0a541e8acd9ad72d2e90df3ba22f158b3                  0.3s
 => => extracting sha256:0d27a8e86132900757 4c6766fba946d48e20d2c8e964e873de352603f22c4ceb                  0.1s
 => [internal] load build context                                                                        55.9s
 => => transferring context: 55.57MB                                                                     55.7s
 => [2/5] WORKDIR /app                                                                                    0.9s
 => [3/5] COPY package*.json ./                                                                           0.4s
 => [4/5] RUN npm install                                                                                47.1s
 => [5/5] COPY . .                                                                                        3.5s
 => exporting to image                                                                                   15.8s
 => => exporting layers                                                                                   7.1s
 => => exporting manifest sha256:a33ea7d60bea4e4174053cb1a87b5ea4a75382bc69e67b2f8ccc2a5a8a2ec8d4         0.1s
 => => exporting config sha256:324d0e23f466b5164adfbf8228dc11c8e6a7a366ed8b251bd4ecaa99a41ac5a4           0.0s
 => => exporting attestation manifest sha256:c464a0166e10489b2f8509b6c8ad773737d746ba986e1eaedd7b9bfcefda415f  0.1s
 => => exporting manifest list sha256:f2026260cfceddfeb4243b98496401af2d07c2951dc5649577387d0a285a7715    0.1s
 => => naming to docker.io/library/myapp:latest                                                           0.0s
 => => unpacking to docker.io/library/myapp:latest                                                        8.2s
```

Exécuter le conteneur Docker : **docker run -p 3000:3000 myapp**

```
C:\express-login-app>docker build -t myapp .
[+] Building 57.3s (11/11) FINISHED                                                                    docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                                  0.1s
 => => transferring dockerfile: 283B                                                                                  0.0s
 => [internal] load metadata for docker.io/library/node:14                                                            2.4s
 => [internal] load .dockerignore                                                                                     0.1s
 => => transferring context: 2B                                                                                       0.0s
 => [1/6] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa      0.2s
 => => resolve docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa      0.1s
 => [internal] load build context                                                                                     1.1s
 => => transferring context: 277.79kB                                                                                 1.1s
 => CACHED [2/6] RUN apt-get update && apt-get install -y   build-essential   python3   && rm -rf /var/lib/apt/list  0.0s
 => CACHED [3/6] WORKDIR /app                                                                                         0.0s
 => [4/6] COPY package*.json ./                                                                                       0.2s
 => [5/6] RUN npm install                                                                                            37.1s
 => [6/6] COPY . .                                                                                                    1.1s
 => exporting to image                                                                                               14.8s
 => => exporting layers                                                                                               7.1s
 => => exporting manifest sha256:c3d3660f24fa14d277ed6206a800a8d9948968b32b601311089996e24678b787                     0.0s
 => => exporting config sha256:10b6477b0b48f539c8429d038b839888750305f281258a3a7f1dba350fe24aaf                       0.0s
 => => exporting attestation manifest sha256:fe5343be3b6e3659e6788f0a4d8a0fb57b8873f0584588230cf5f553b6af745c         0.1s
 => => exporting manifest list sha256:49f456845d561fa794858e12283af5e3ca106a329ba47cd52c58534069a42283               0.1s
 => => naming to docker.io/library/myapp:latest                                                                       0.0s
 => => unpacking to docker.io/library/myapp:latest                                                                    7.2s

C:\express-login-app>docker run -p 3000:3000 myapp
http://localhost:3000
MongoDB connected
```

Valider et pousser le Dockerfile :

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\express-login-app> git add .
PS C:\express-login-app> git commit -m "Dockerfile added"
[main 5e0cbb1] Dockerfile added
 4 files changed, 67 insertions(+), 2189 deletions(-)
 create mode 100644 .github/workflows/cd.yml
 create mode 100644 .github/workflows/ci.yml
 create mode 100644 Dockerfile
 delete mode 100644 package-lock.json
PS C:\express-login-app> git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 937 bytes | 234.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/khawlagr/devops-project.git
   7c988b2..5e0cbb1  main -> main
PS C:\express-login-app>
```



.github/workflows	Dockerfile added	1 minute ago
models	Initial commit	19 hours ago
public	first commit	1 hour ago
views	first commit	1 hour ago
.gitignore	first commit	1 hour ago
Dockerfile	Dockerfile added	1 minute ago
README.md	Initial commit	19 hours ago
package.json	first commit	1 hour ago
server.js	first commit	1 hour ago

4. Configurer l'Intégration Continue (CI)

Installer **Mocha** comme dépendance de développement pour les tests dans votre projet.

```
PS C:\express-login-app> npm install --save-dev mocha

up to date, audited 293 packages in 1s

51 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\express-login-app>
```

Enregistrer le workflow CI/CD avec un message de commit, puis le publie sur la branche principale du dépôt distant.

```
git commit -m "Ajout du workflow CI/CD"
git push origin main
```

```
PS C:\express-login-app> git commit -m "Add CI/CD Workflow"
[main 9c2909a] Add CI/CD Workflow
 10 files changed, 1547 insertions(+)
 create mode 100644 main.tf
 create mode 100644 terraform/.terraform.lock.hcl
 create mode 100644 terraform/.terraform/providers/registry.terraform.io/kreuzwerker/docker/2.25.0/windows_amd64/CHANGELO
 create mode 100644 terraform/.terraform/providers/registry.terraform.io/kreuzwerker/docker/2.25.0/windows_amd64/LICENSE
 create mode 100644 terraform/.terraform/providers/registry.terraform.io/kreuzwerker/docker/2.25.0/windows_amd64/README.m
 create mode 100644 terraform/.terraform/providers/registry.terraform.io/kreuzwerker/docker/2.25.0/windows_amd64/terrafor
exe
 create mode 100644 terraform/main.tf
 create mode 100644 terraform/terraform.tfstate
 create mode 100644 terraform/terraform.tfstate.backup
PS C:\express-login-app> git push origin main
>>
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 8 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (21/21), 32.99 MiB | 441.00 KiB/s, done.
Total 21 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
remote: warning: See https://gh.io/lfs for more information.
remote: warning: File terraform/bin/terraform.exe is 88.41 MB; this is larger than GitHub's recommended maximum file size
remote: warning: GH001: Large files detected. You may want to try Git Large File Storage - https://git-lfs.github.com.
To https://github.com/khawlagr/devops-project.git
   2d4435c..9c2909a  main -> main
PS C:\express-login-app> git push origin main
>>
Everything up-to-date
```

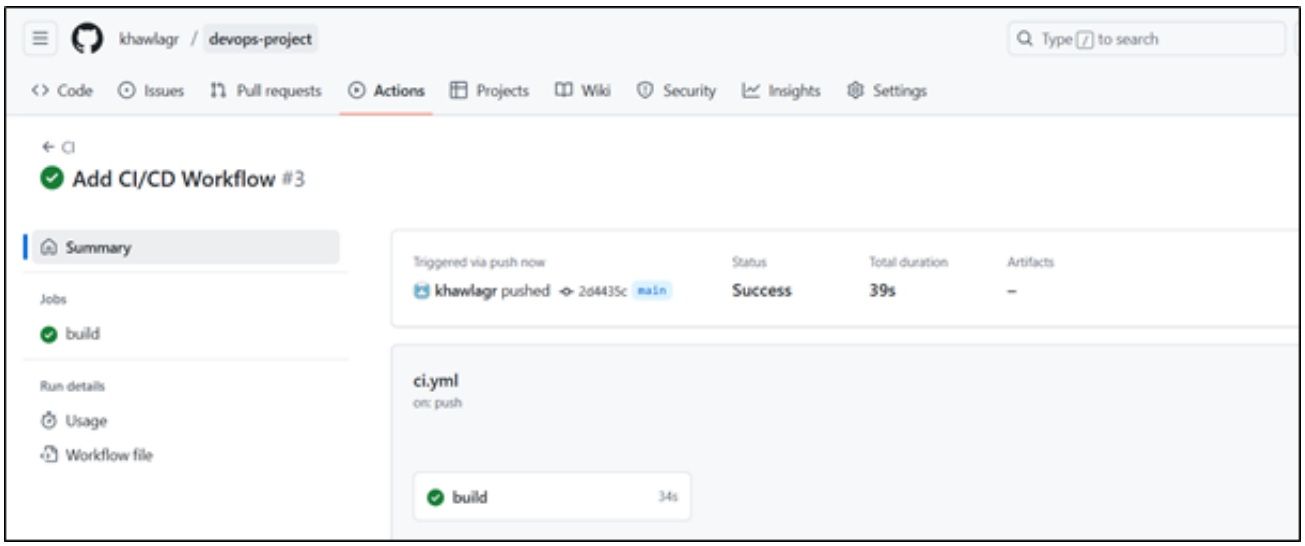
Créer un fichier de configuration **ci.yml**

ci.yml

.github > workflows > ci.yml

```yaml

on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest

    steps:
    - name: Checkout code
      uses: actions/checkout@v2

    - name: Set up Docker Buildx
      uses: docker/setup-buildx-action@v1

    - name: Build Docker image
      run: docker build -t myapp .

    - name: Run tests
      run: docker run myapp npm test

```

CI ajouté avec succès.



5. Configurer le Déploiement Continu (CD)

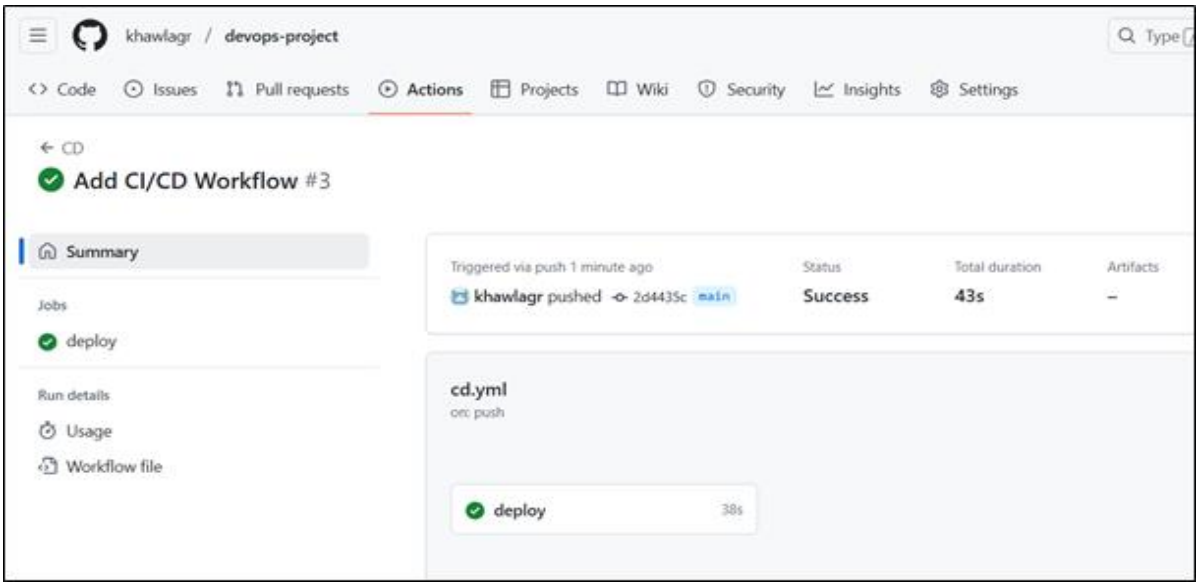
Créer un fichier de configuration **cd.yml**

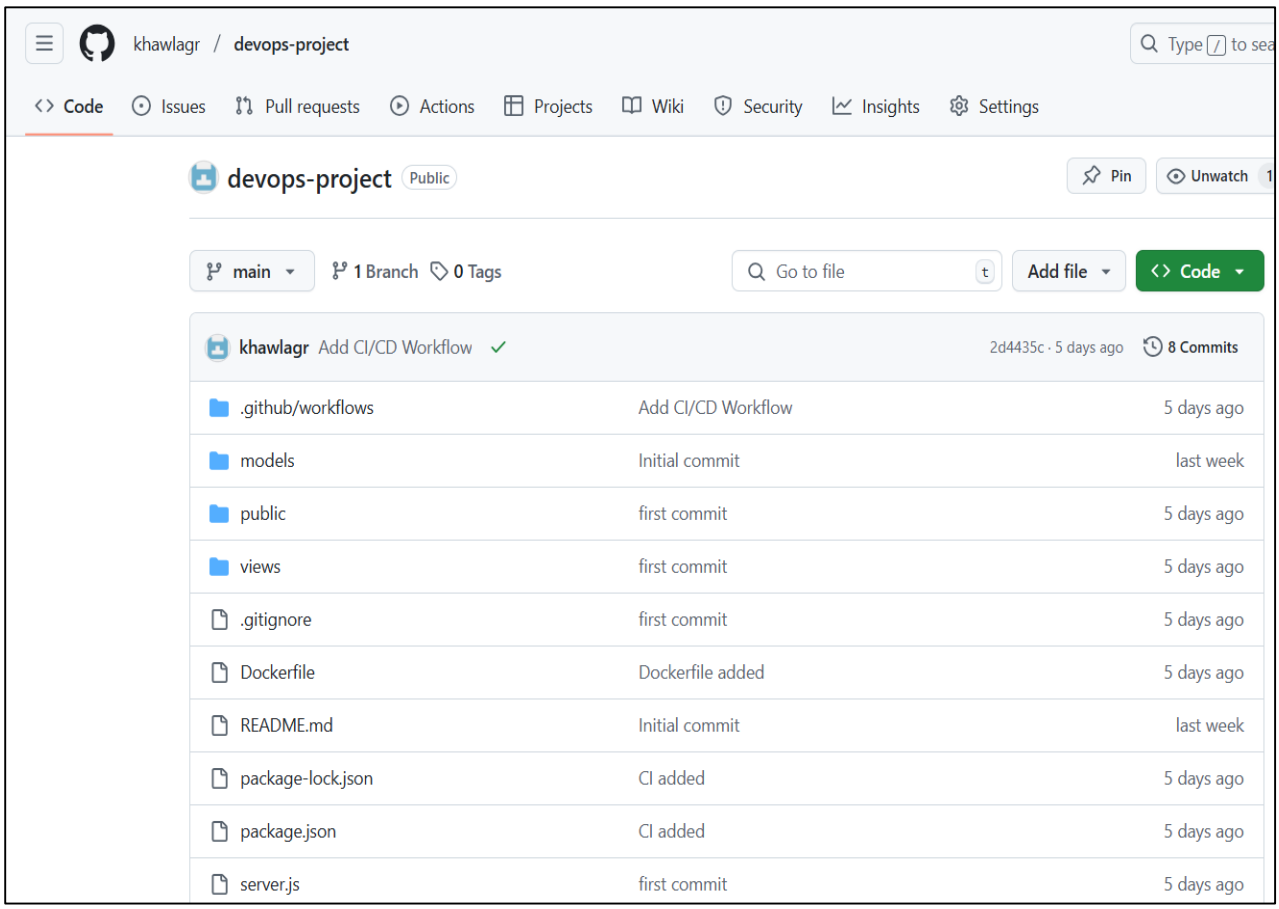
cd.yml

.github > workflows > cd.yml

```yaml
name: CD

on:
  push:
    branches:
      - main

jobs:
  deploy:
    runs-on: ubuntu-latest

    steps:
    - name: Checkout code
      uses: actions/checkout@v2

    - name: Set up Docker Buildx
      uses: docker/setup-buildx-action@v1

    - name: Build Docker image
      run: docker build -t myapp .

    - name: Deploy Docker container
      run: |
        docker stop myapp || true
        docker rm myapp || true
        docker run -d -p 3000:3000 --name myapp myapp

```

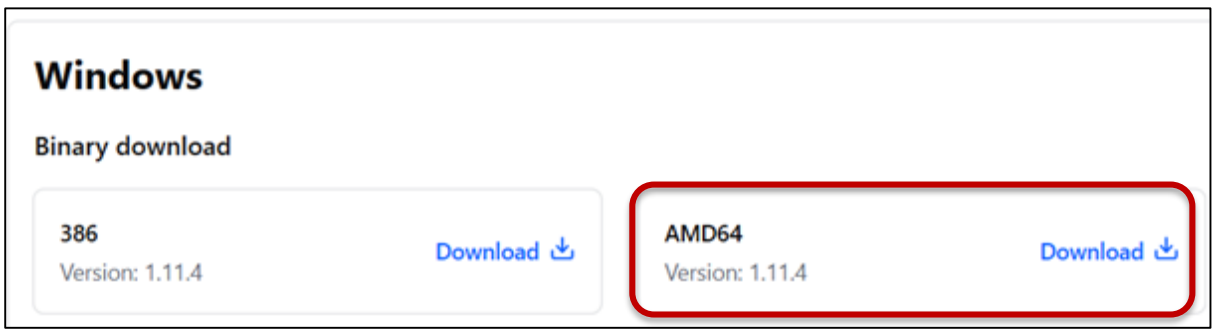
CD ajouté avec succès.





6. Infrastructure en tant que Code avec Terraform

Télécharger et installer Terraform depuis le site officiel :
https://developer.hashicorp.com/terraform/downloads



Vérifier l'installation de Terraform avec la commande **terraform -version**

```
Invite de commandes
Microsoft Windows [version 10.0.22000.3260]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Lenovo>terraform -version
Terraform v1.11.4
on windows_amd64
```

main.tf file:

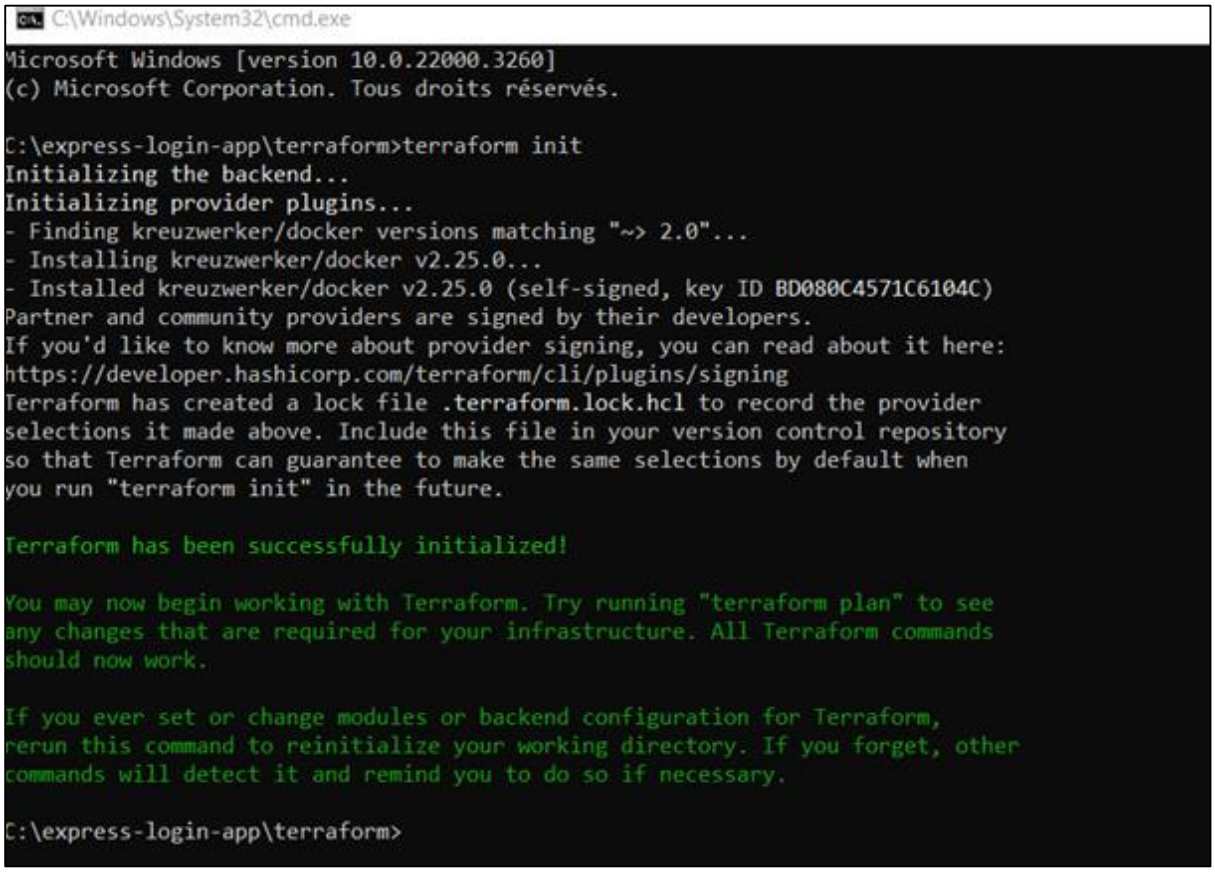
main.tf U

main.tf

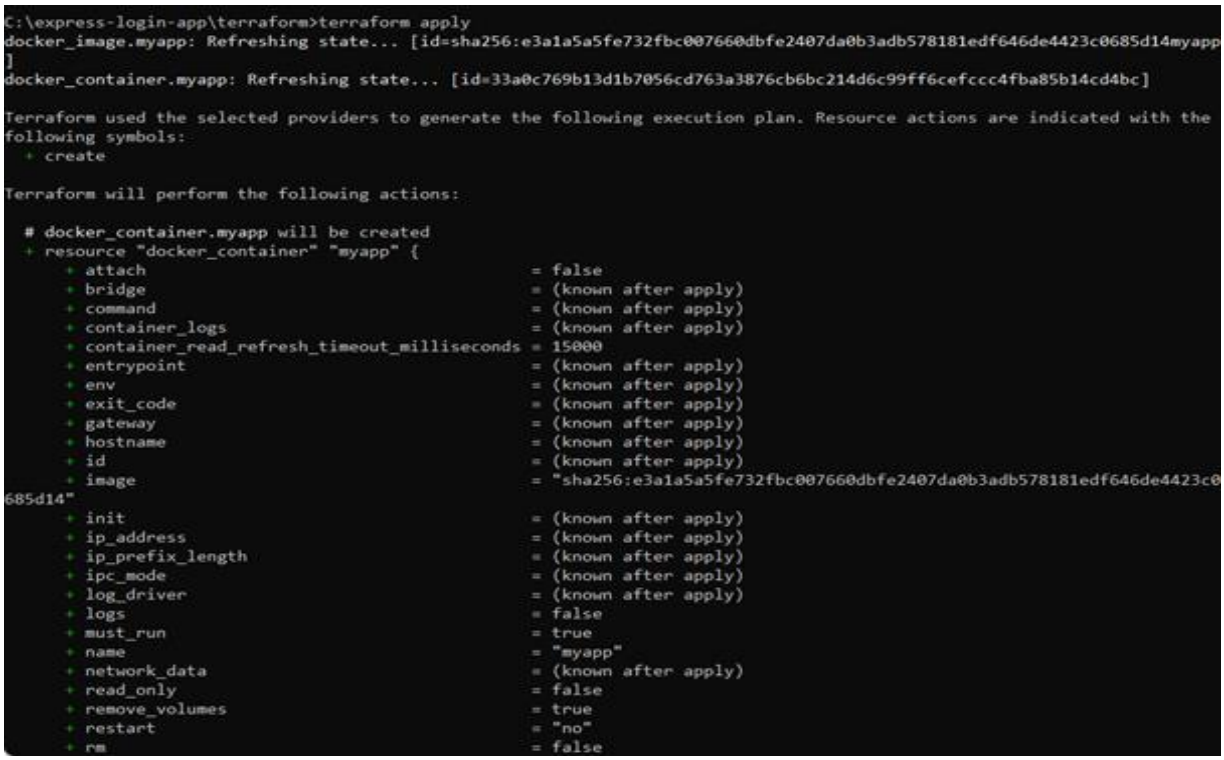
```
terraform {
  required_providers {
    docker = {
      source = "kreuzwerker/docker"
      version = "~> 2.0"
    }
  }
}

provider "docker" {}



resource "docker_image" "myapp" {
  name = "myapp"
  build {
    context = "${path.module}/.."
  }
}

resource "docker_container" "myapp" {
  name  = "myapp"
  image = docker_image.myapp.latest

  ports {
    internal = 3000
    external = 3001
  }
}

```

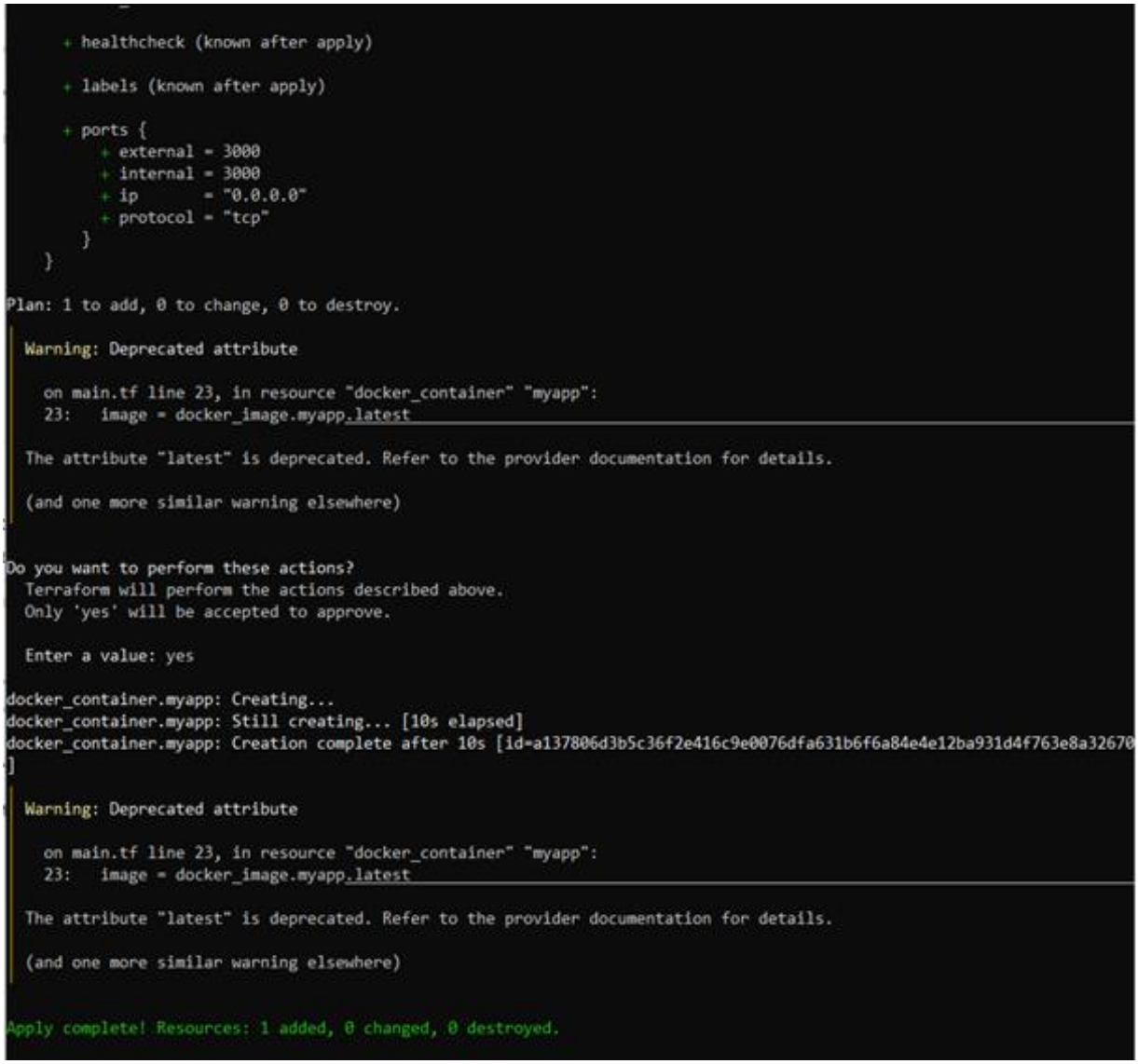
Ce fichier déclare un provider Docker, construit une image Docker, puis déploie un conteneur exposant le port 3000 en interne et le port 3001 en externe.

Initialiser Terraform avec la commande : **terraform init**



Appliquer la configuration avec la commande : **terraform apply**





Nous vous remercions pour votre attention ☺