



Rapport Programmation Distribuée

Fait Par

**Montassar Benabdallah
Sanessy Cisse**

**Master 2 MIAGE Groupe 1
Promotion 2024/2025**

Google LABS

Voici les captures des google Labs pour :

1)Montassar Benabdallah

Google Cloud

Tableau de bord

Explorer

Parcours

Profil

Abonnements

★ 140 pts

🏆 9th

🔍

?

🌐

👤

Montassar Benabdallah

Date d'abonnement : 2024

370 points

Votre profil n'est pas public ni accessible. [Rendre le profil public](#)

☰

Parcours de formation

☰

Activités

📊

Classement

🏆

Badges

Cours

Atelier

Quiz

Jeu

En cours

Terminée

Activité	Type	Date de début	Date de fin	Score	Réussie
Infrastructure as Code with Terraform	Atelier	il y a 28 minutes	il y a 2 minutes	Assessment: 100%	✓
Creating a Virtual Machine	Atelier	8 nov. 2024	8 nov. 2024	Assessment: 100%	✓
A Tour of Google Cloud Hands-on Labs	Atelier	27 sept. 2024	27 sept. 2024	Assessment: 100%	✓

2)Sanessy Cisse

← → ↺

cloudskillsboost.google/profile/activity

🔍 ☆ 📌

🇸🇩

Nouvelle version de Chrome disponible

Google Cloud

Tableau de bord

Explorer

Parcours

Profil

Abonnements

★ 0 pts


🏆

🔍

?

🌐

👤



Sanessy Cisse

Date d'abonnement : 2024

370 points

Votre profil n'est pas public ni accessible. [Rendre le profil public](#)

☰

Parcours de formation

☰

Activités

📊

Classement

🏆

Badges

Cours

Atelier

Quiz

Jeu

En cours

Terminée

Activité	Type	Date de début	Date de fin	Score	Réussie
Infrastructure as Code avec Terraform	Atelier	20 déc. 2024	20 déc. 2024	Assessment: 100%	✓
Creating a Virtual Machine	Atelier	8 nov. 2024	8 nov. 2024	Assessment: 100%	✓
A Tour of Google Cloud Hands-on Labs	Atelier	27 sept. 2024	27 sept. 2024	Assessment: 100%	✓

<https://www.cloudskillsboost.google/catalan>

Projet : Site de Fripperie

Description du Projet

Notre projet consiste en un **site de fripperie** où les utilisateurs peuvent consulter et acheter des vêtements d'occasion. Le site comprend les technologies suivantes :

- **Backend** : Node.js et Express
- **Base de données** : PostgreSQL
- **Frontend** : EJS
- **Design** : Interface professionnelle offrant une expérience utilisateur fluide

Le projet utilise **Docker**, **Kubernetes**, avec deux services et un Ingress pour la gestion de l'architecture.

Guide d'Installation et de Lancement

1. Cloner le dépôt Git

Clonez le dépôt Git en exécutant la commande suivante dans votre terminal :

```
git clone git@github.com:benabdam/Prog_Dist.git
```

2. Accéder au répertoire du projet

Accédez au dossier du projet avec la commande suivante :

```
cd "Prog_Dist/new"
```

3. Lancer le projet avec Docker

Utilisez Docker pour lancer le projet avec la commande suivante :

```
docker-compose up --build
```


Une fois l'application lancée, vous pouvez y accéder à **localhost:3000**.

4. Docker Hub


L'image Docker `montassar099/new-backend:v1.0` est disponible sur **Docker Hub**. Vous pouvez la retrouver et l'utiliser directement depuis Docker Hub.


montassar099 / [Repositories](#) / [new-backend](#) / [General](#) Using 0 of 1 private repositories.

General Tags Builds Collaborators Webhooks Settings

montassar099/new-backend 

Last pushed 14 days ago

[Add a description](#)  INCOMPLETE

[Add a category](#)  INCOMPLETE



Docker commands Public view

To push a new tag to this repository:

```
docker push montassar099/new-backend:tagname
```

Tags


This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
 v1.0		Image	an hour ago	14 days ago

[See all](#)

Automated builds

Manually pushing images to Docker Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#) .

Partie Kubernetes

Pré-requis

- **Docker Desktop** doit être lancé.
- **Minikube** doit être installé sur votre machine.

1. Démarrer Minikube

Lancez Minikube avec la commande suivante :

`minikube start`

```
C:\Users\Montassar\Documents>cd Prog_Dist
C:\Users\Montassar\Documents\Prog_Dist>cd new
C:\Users\Montassar\Documents\Prog_Dist\new>minikube start
* minikube v1.34.0 sur Microsoft Windows 10 Home 10.0.19045.5371 Build 19045.5371
* Choix automatique du pilote docker
* Utilisation du pilote Docker Desktop avec le privilège root
* Démarrage du nœud "minikube" primary control-plane dans le cluster "minikube"
* Extraction de l'image de base v0.0.45...
* Création de docker container (CPU=2, Memory=2200Mo) ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* Pour extraire de nouvelles images externes, vous devrez peut-être configurer un proxy : https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Préparation de Kubernetes v1.31.0 sur Docker 27.2.0...
  - Génération des certificats et des clés
  - Démarrage du plan de contrôle ...
  - Configuration des règles RBAC ...
* 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
* Terminé ! kubectl est maintenant configuré pour utiliser "minikube" cluster et espace de noms "default" par défaut.
```

2. Créer le Volume Persistant (PV)

Appliquez le fichier de création du **Persistent Volume** avec la commande :

```
kubectl apply -f pv.yaml
```

```
C:\Users\Montassar\Documents\Prog_Dist\new>kubectl apply -f pv.yaml
persistentvolume/postgres-pv created
persistentvolume/session-pv created
```

3. Créer le Persistent Volume Claim (PVC)

Ensuite, appliquez le fichier de création du **Persistent Volume Claim** :

```
kubectl apply -f PVC.yaml
```

```
C:\Users\Montassar\Documents\Prog_Dist\new>kubectl apply -f PVC.yaml
persistentvolumeclaim/postgres-pvc created
persistentvolumeclaim/session-pvc created
```

Vérifiez l'état du PV et du PVC avec la commande :

```
kubectl get pv
```

Assurez-vous que l'état du PV est "**Bound**".

```
C:\Users\Montassar\Documents\Prog_Dist\new>kubectl get pv
```

NAME	CAPACITY	ACCESS MODES	RECLAIM POLICY	STATUS	CLAIM	STORAGECLASS	VOLUMEATTRIBUTES
CLASS	REASON	AGE					
postgres-pv	1Gi	RWO	Retain	Bound	default/postgres-pvc	hostpath	<unset>
	93s						
session-pv	1Gi	RWX	Retain	Bound	default/session-pvc	hostpath	<unset>
	93s						

4. Configurer la Base de Données

Appliquez le **ConfigMap** de la base de données avec la commande suivante :

```
kubectl apply -f init-configmap.yaml
```

```
C:\Users\Montassar\Documents\Prog_Dist\new>kubectl apply -f init-configmap.yaml
configmap/init-sql-configmap created
```

5. Déployer la Base de Données PostgreSQL

Appliquez le fichier de déploiement de la base de données PostgreSQL avec la commande :

```
kubectl apply -f db-deployment.yaml
```

6. Créer le Service pour la Base de Données

Appliquez le fichier de création du service pour la base de données avec la commande suivante :

```
kubectl apply -f db-service.yaml
```

Vérifiez que l'état du pod PostgreSQL est "**Running**" avec la commande :

kubectl get pods

```
C:\Users\Montassar\Documents\Prog_Dist\new>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
db-deployment-6f59869597-g6cqd     1/1     Running   0           2m14s
```

7. Déployer le Backend

Appliquez le fichier de déploiement du backend avec la commande :

kubectl apply -f backend-deployment.yaml

8. Créer le Service pour le Backend

Appliquez le fichier de création du service pour le backend avec la commande suivante :

kubectl apply -f backend-service.yaml

```
C:\Users\Montassar\Documents\Prog_Dist\new>kubectl apply -f db-deployment.yaml
deployment.apps/db-deployment created

C:\Users\Montassar\Documents\Prog_Dist\new>kubectl apply -f db-service.yaml
service/db-service created
```

Vérifiez que l'état du pod backend est "**Running**" avec la commande :

kubectl get pods

```
C:\Users\Montassar\Documents\Prog_Dist\new>kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
backend-deployment-56bf4796f-j8lwg  1/1     Running   0           4m37s
db-deployment-6f59869597-g6cqd     1/1     Running   0           7m30s
```

9. Appliquer Ingress

Appliquez la configuration de l'Ingress avec la commande suivante :

kubectl apply -f <https://raw.githubusercontent.com/kubernetes/ingress-nginx/main/deploy/static/provider/cloud/deploy.yaml>

10. Activer Ingress dans Minikube

Activez Ingress avec la commande suivante :

minikube addons enable ingress

```
C:\Users\Montassar\Documents\Prog_Dist\new>minikube addons enable ingress
* ingress est un addon maintenu par Kubernetes. Pour toute question, contactez minikube sur GitHub.
Vous pouvez consulter la liste des mainteneurs de minikube sur : https://github.com/kubernetes/minikube/blob/master/OWNERS
* Après que le module est activé, veuillez exécuter "minikube tunnel" et vos ressources ingress seront disponibles à "127.0.0.1"
  - Utilisation de l'image registry.k8s.io/ingress-nginx/controller:v1.11.2
  - Utilisation de l'image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.3
  - Utilisation de l'image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.3
* Vérification du module ingress...
* Le module 'ingress' est activé
```

11. Appliquer la configuration Ingress

Appliquez le fichier d'Ingress avec la commande :

```
kubectl apply -f ingress.yaml
```

```
C:\Users\Montassar\Documents\Prog_Dist\new>kubectl apply -f ingress.yaml
ingress.networking.k8s.io/app-ingress created
```

12. Modifier le fichier /etc/hosts

Ajoutez la ligne suivante dans le fichier /etc/hosts pour mapper l'URL de l'application localement :

```
127.0.0.1 mafripperie.info
```

13. Lancer Minikube Tunnel

Lancez Minikube Tunnel pour exposer les services :

```
minikube tunnel
```

```
C:\Users\Montassar\Documents\Prog_Dist\new>minikube tunnel
Tunnel démarré avec succès

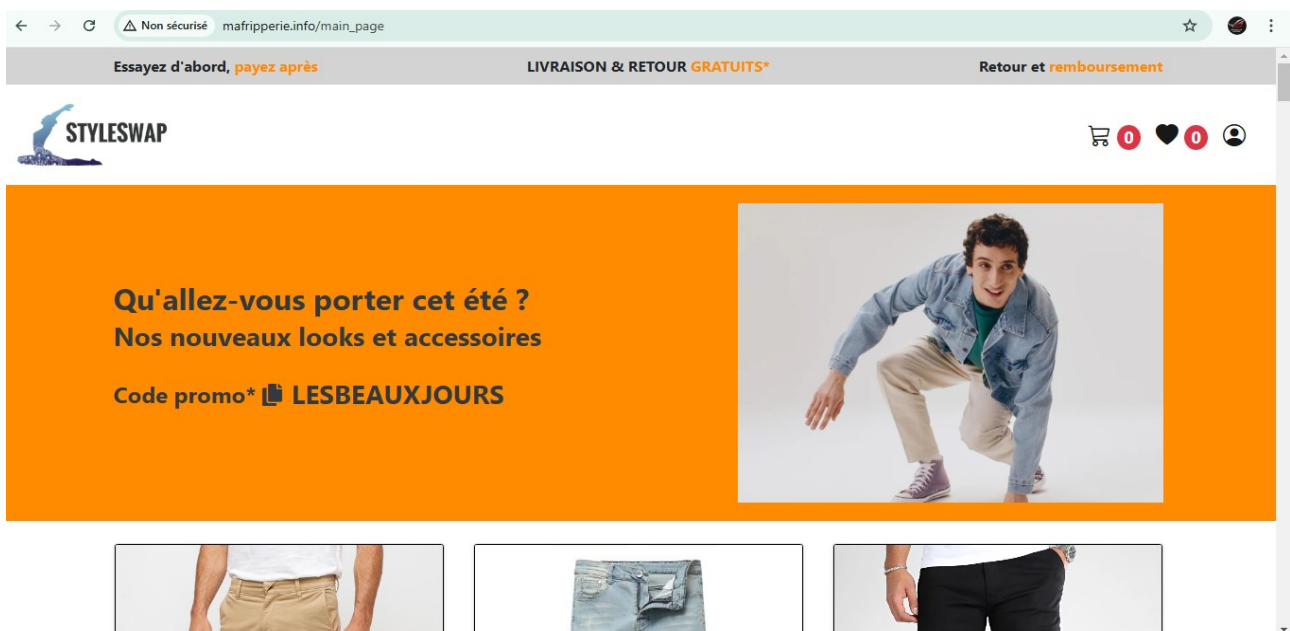
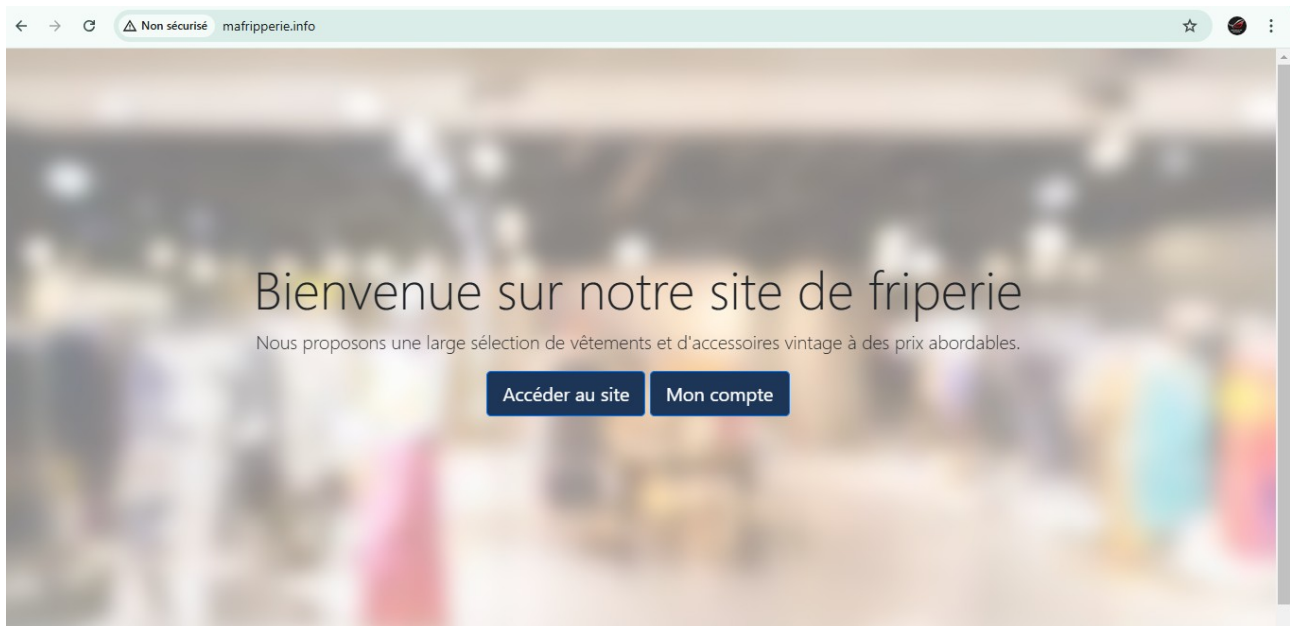
REMARQUE : veuillez ne pas fermer ce terminal car ce processus doit rester actif pour que le tunnel soit accessible...

Accéder aux ports inférieurs à 1024 peut échouer sur Windows avec les clients OpenSSH antérieurs à v8.1. Pour plus d'information, voir: https://minikube.sigs.k8s.io/docs/handbook/accessing/#access-to-ports-1024-on-windows-requires-root-permission
Tunnel de démarrage pour le service app-ingress.
```

14. Accéder au Site

Une fois Minikube Tunnel lancé, accédez au site en ouvrant votre navigateur à l'adresse suivante :

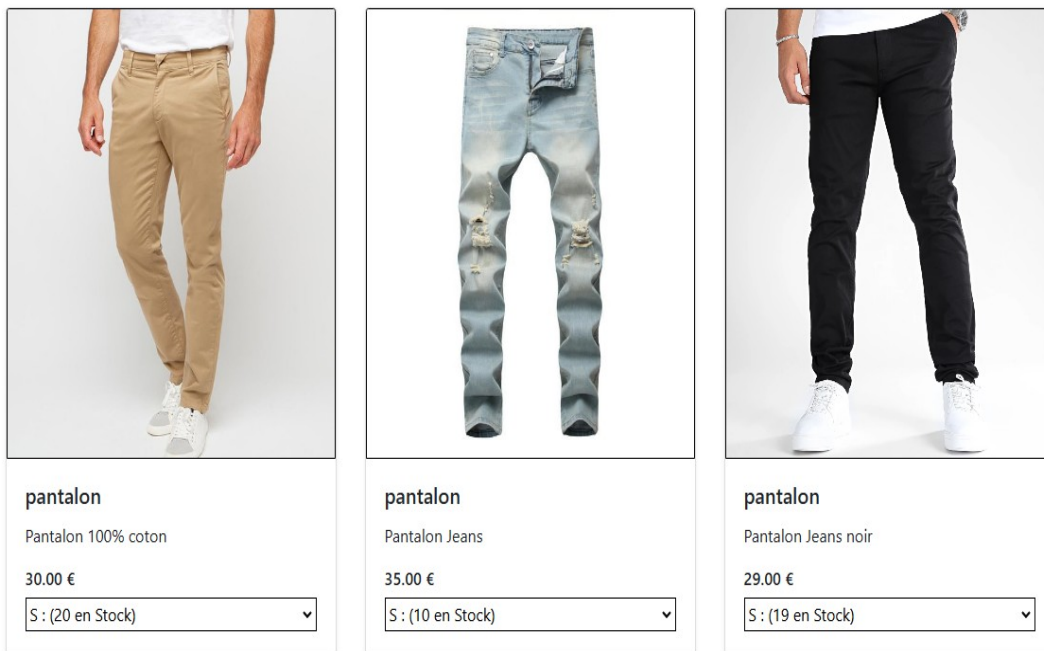
<http://mafripperie.info>



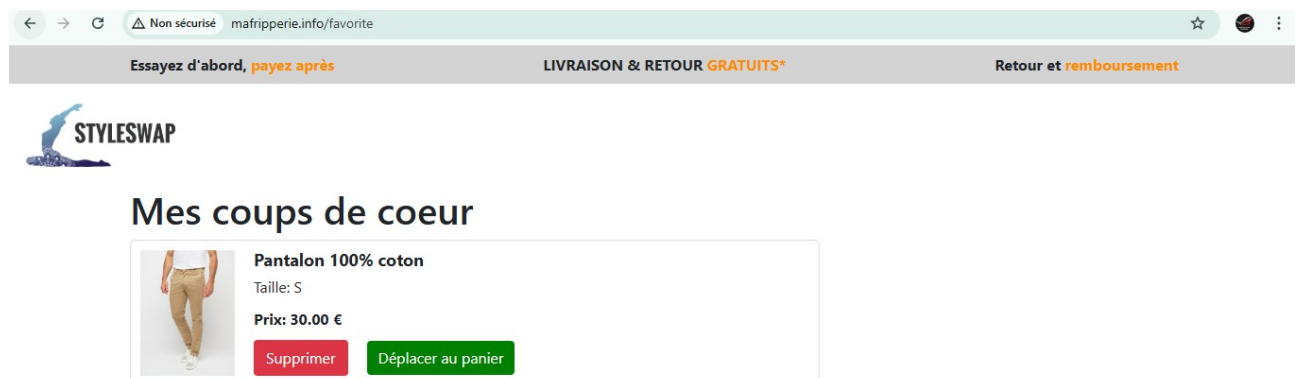
Fonctionnalités du Site

Notre site de friperie permet aux utilisateurs de réaliser plusieurs actions une fois sur la plateforme :

- **Ajouter un article** aux favoris ou au panier en fonction de la taille et du stock disponible.



- **Consulter la liste des favoris**, supprimer un article ou déplacer un favori dans le panier.



- **Consulter le panier**, ajuster la quantité des articles, appliquer un code promo et valider la commande.



Pantalon 100% coton

Taille: S

Prix: 30.00 €

Supprimer

1

Récapitulatif du panier

Nombre d'article	1
Sous-total	30.00 €
Reduction	0.00 €
Livraison	0 €
Total	30.00 €

Code promo (facultatif)

Entrer le code de réduction:

LESBEAUXJOURS

Appliquer/Supprimer code

Valider ma commande

Essayez d'abord, payez aprèsLIVRAISON & RETOUR GRATUITS*Retour et remboursement



Validation de commande

example@gmail.com

45 rue des saints pères

75006


paris

0125467389

Confirmer

← → ↺ Non sécurisé mafripperie.info/final☆🍪⋮


Essayez d'abord, payez aprèsLIVRAISON & RETOUR GRATUITS*Retour et remboursement



Votre commande a été confirmée!

Merci pour votre confiance.

Livraison estimée dans 3 jours ouvrés sauf weekend et jours fériés.



🛒 0

❤️ 0

👤

- Le site propose également un **espace gérant** où il peut gérer le stock et les commandes :
 - Ajouter ou enlever des articles du stock.
 - Supprimer une commande après livraison.

