

KUBI

TAKE THE HELM

EPITECH 2023

R A P P O R T T E C H N I Q U E

CLOUD 902

DURAMANA KALUMVUATI
TAHA IGUIMDRANE
AGHILAS AZZOUG
MAWUSE KOMI AIMÉ ALOGNIKOU
GAUTIER BLANDIN

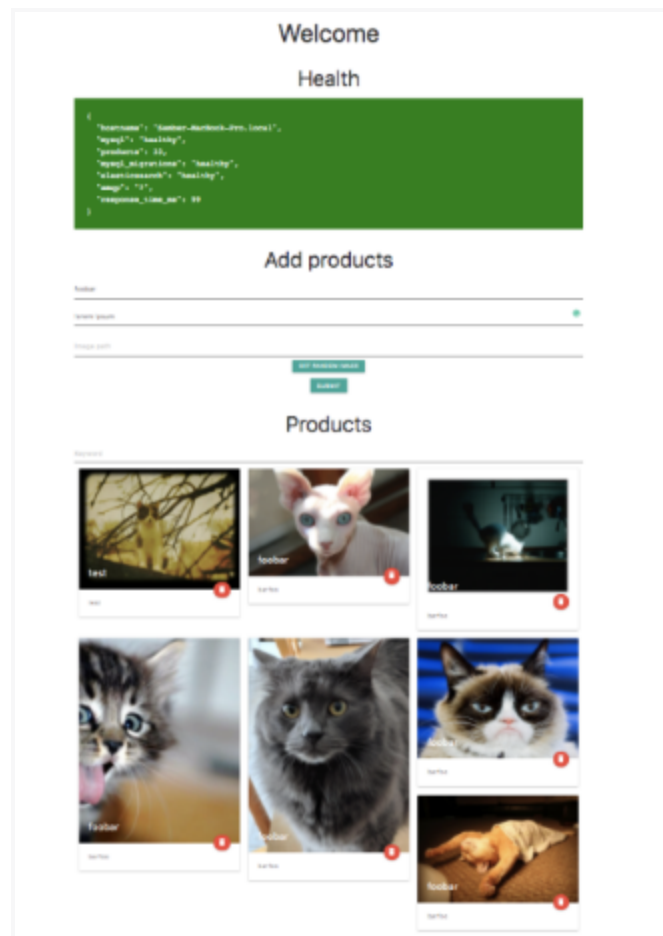
Introduction.....	2
Technologies utilisées:.....	3
L'architecture qu'on met en place.....	3
Structure du projet.....	4
Les Images docker dans notre Hub:.....	4
Spécifications techniques:	
I. Terraform.....	5
II. Activation d'AGIC.....	6
III. Installation de Helm.....	8
L'automatisation de l'exécution dans le Makefile:.....	10

Introduction

Ce document technique fournit des informations détaillées sur le projet, ainsi que sur les technologies utilisées.

Tout en mettant en illustration l'architecture construite, ainsi que nos composants.

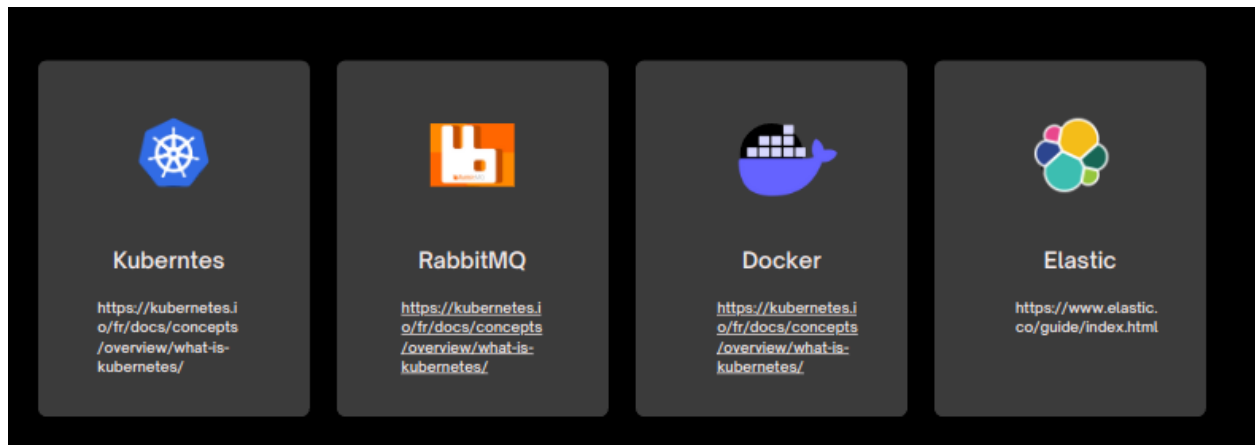
Le projet nous a permis de découvrir l'outil Kubernetes et l'univers du Cloud avec profondeur, tout en manipulant d'autres outils et frameworks.



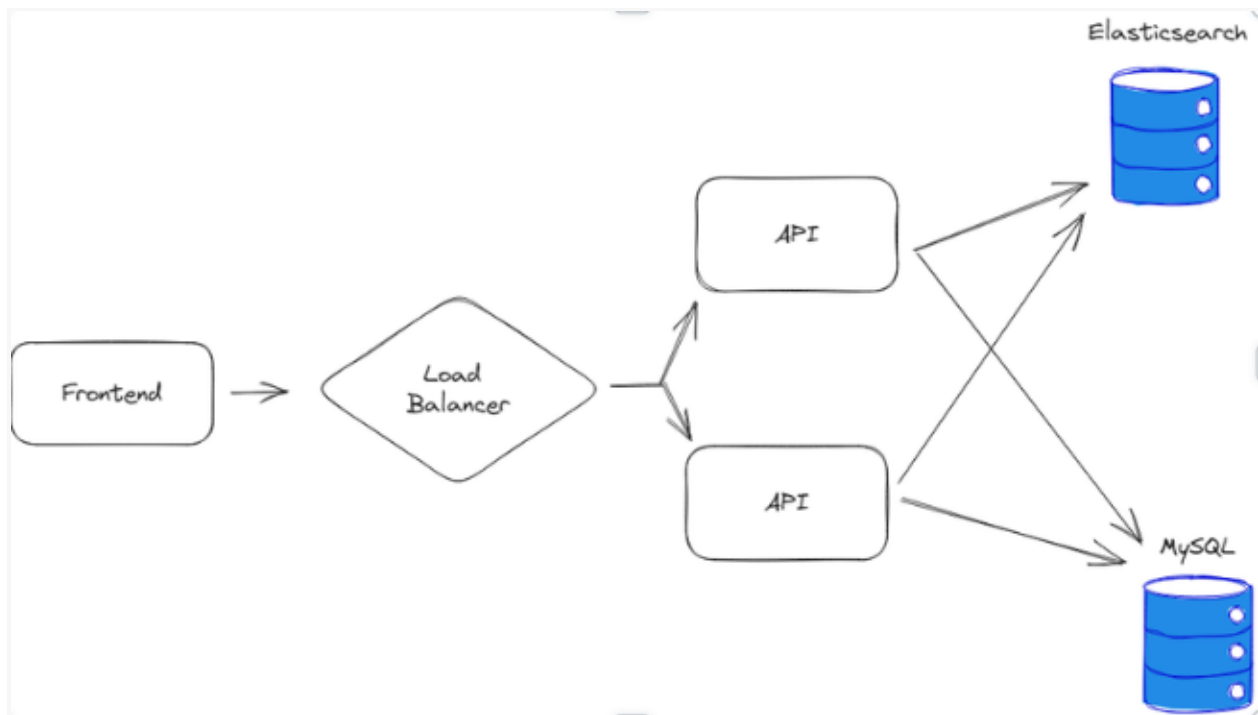
Présentation de projet

Le projet consiste à déployer une application d'administration d'e-commerce dans un cluster Kubernetes

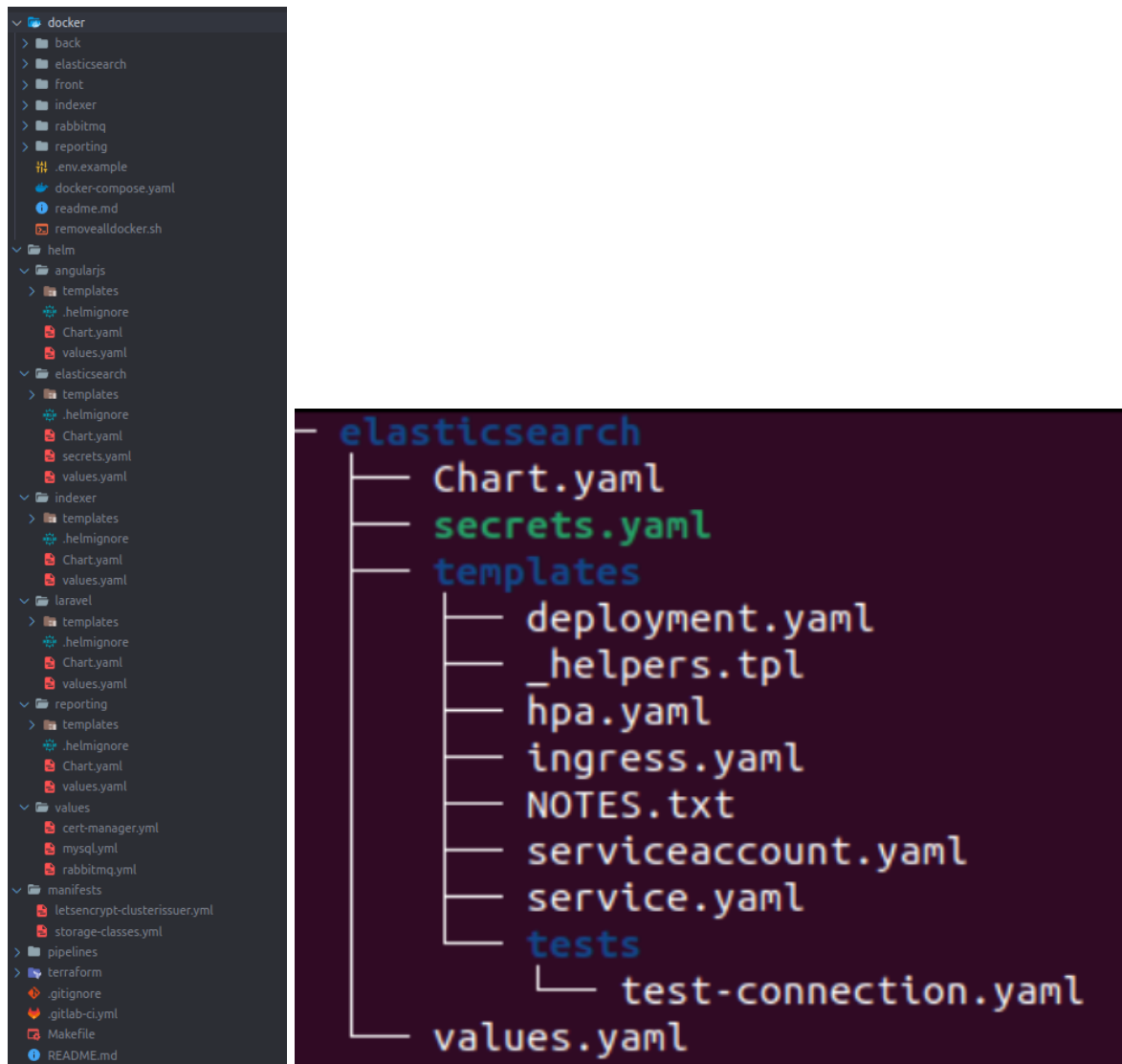
Technologies utilisées:



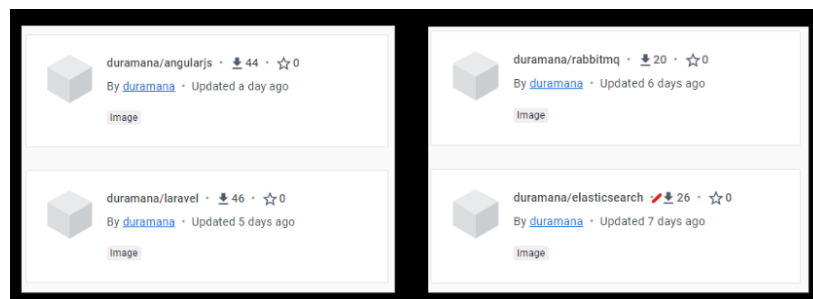
L'architecture qu'on met en place



Structure du projet



Les Images docker dans notre Hub:



Les Spécifications Techniques:

Ce document de spécification technique fournit des informations détaillées sur les commandes et les étapes techniques nécessaires pour exécuter et configurer le projet.

I. Terraform

Initialisation de Terraform:

Pour initialiser Terraform, exécutez la commande suivante dans le répertoire "terraform":

```
cd terraform && terraform init
```

Planification de Terraform:

Pour planifier les modifications Terraform, exécutez la commande suivante :

```
terraform plan -out plan.tfplan
```

Cette commande génère un plan d'exécution Terraform qui sera utilisé ultérieurement.

Application de Terraform:

Pour appliquer les modifications Terraform et créer l'infrastructure, exécutez la commande suivante :

```
terraform apply plan.tfplan -auto-approve
```

Cette commande génère un plan d'exécution Terraform qui sera utilisé ultérieurement.

Récupération du fichier kubeconfig

Pour obtenir le fichier kubeconfig permettant de se connecter au cluster AKS, exécutez la commande suivante :

```
# Cette commande vous aidera à utiliser OpenLens pour vous connecter au cluster
# az = azure cli
az aks get-credentials --resource-group rg-kubi --name aks-kubi
```

Suppression de Terraform

Pour supprimer l'infrastructure créée par Terraform, exécutez la commande suivante :

```
terraform destroy -auto-approve
```

Connexion au cluster AKS

Pour vous connecter au cluster AKS, exécutez la commande suivante :

```
az aks get-credentials --resource-group rg-kubi --name aks-kubi
```

II. Activation d'AGIC

Pour activer l'AGIC (Application Gateway Ingress Controller), veuillez suivre les instructions fournies dans le lien suivant : [Lien AGIC](#)

```
az aks enable-addons -n aks-kubi -g rg-kubi --subscription
"394e6431-b5e0-4004-b173-bc15e0379ebf" -a ingress-appgw --appgw-id
"/subscriptions/394e6431-b5e0-4004-b173-bc15e0379ebf/resourceGroups/rg-k
ubi/providers/Microsoft.Network/applicationGateways/agic-appgw"
```

III. Installation de Helm

Installation de Angular

Pour installer Angular à l'aide de Helm, exécutez la commande suivante :

```
helm install angular -n applications --create-namespace  
./values/angularjs
```

Installation de Laravel

Pour installer Laravel à l'aide de Helm, exécutez la commande suivante :

```
helm install laravel -n applications --create-namespace ./values/laravel
```

Installation d'Elasticsearch

Pour installer Elasticsearch à l'aide de Helm, exécutez la commande suivante :

```
helm install elasticsearch -n applications --create-namespace  
./values/elasticsearch
```

Installation de l'indexeur

Pour installer l'indexeur à l'aide de Helm, exécutez la commande suivante :

```
helm install indexer -n applications --create-namespace ./values/indexer
```

Installation de Reporting

Pour installer Reporting à l'aide de Helm, exécutez la commande suivante :

```
helm install reporting -n applications --create-namespace  
./values/reporting
```

N'oubliez pas de modifier le domaine IP (DNS) dans le fichier hosts.

Assurez-vous d'ajouter l'accès du contributeur à AppGW et vérifiez les journaux AGIC.

L'automatisation de l'exécution dans le Makefile:

Le fichier Makefile fournit des commandes pour automatiser certaines tâches dans le projet. Voici une explication technique des principales commandes du fichier Makefile :

```
# Dockers
run:
    echo "Running docker-compose"
    cd docker && \
    docker-compose up --build
run_d:
    echo "Running docker-compose in detached mode"
    cd docker && \
    docker-compose up --build -d
...
```

Voici un petit manuel qui explique les commandes utilisées dans le [Makefile](#) .

Ces commandes automatisent certaines étapes du déploiement et de la gestion des conteneurs Docker, ainsi que des opérations liées à Terraform et à Helm dans le projet :

Commandes pour les conteneurs Docker

- `run` : Cette commande exécute `docker-compose up --build` pour démarrer les conteneurs Docker en utilisant les fichiers de configuration de docker-compose.
- `run_d` : Cette commande exécute `docker-compose up --build -d` pour démarrer les conteneurs Docker en mode détaché.
- `down` : Cette commande exécute `docker-compose down` pour arrêter les conteneurs Docker.
- `down_v` : Cette commande exécute `docker-compose down -v` pour arrêter les conteneurs Docker et supprimer les volumes associés.
- `remove` : Cette commande exécute un script `removealldocker.sh` pour supprimer tous les conteneurs Docker.

Commandes pour les prérequis et Helm

- `preinstall` : Cette commande exécute `terraform init` et `terraform plan` dans le

répertoire "terraform" pour initialiser et planifier les modifications Terraform.

- ``helm_install`` : Cette commande exécute ``terraform apply -auto-approve`` pour appliquer les modifications Terraform, puis installe les charts Helm à l'aide de la commande ``helm install`` dans le répertoire "helm".
- ``helm_upgrade`` : Cette commande met à niveau les charts Helm existants à l'aide de la commande ``helm upgrade`` dans le répertoire "helm". Elle inclut également des mises à niveau spécifiques pour des composants tels que RabbitMQ, MySQL et Cert-Manager.
- ``helm_uninstalled`` : Cette commande désinstalle les charts Helm à l'aide de la commande ``helm uninstall`` dans le répertoire "helm".
- ``helm_delete`` : Cette commande supprime les charts Helm à l'aide de la commande ``helm delete``.