Faire une démonstration technique de l'usage du service de conteneurs managés dans Azure

INTRODUCTION

Il s'agit de faire une démonstration simple du déploiement d'une application web Node-Azure- Doker via le service de conteneurs managés dans Azure.

Ce guide a pour but de servir de démonstration pour un collègue et indiquera la démarche à suivre pour déployer l'application Web.

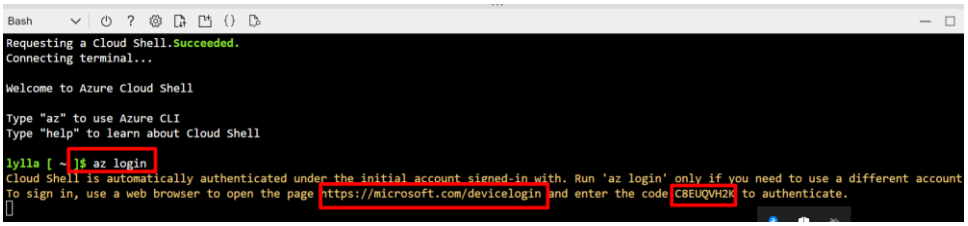
Les prérequis nécessaires pour la mise en place du déploiement de l'application web sont les suivants :

- Disposer d'un abonnement Azure
- Installer Azure CLI
- Installer l'extension "Azure Container Apps" pour l'interface CLI

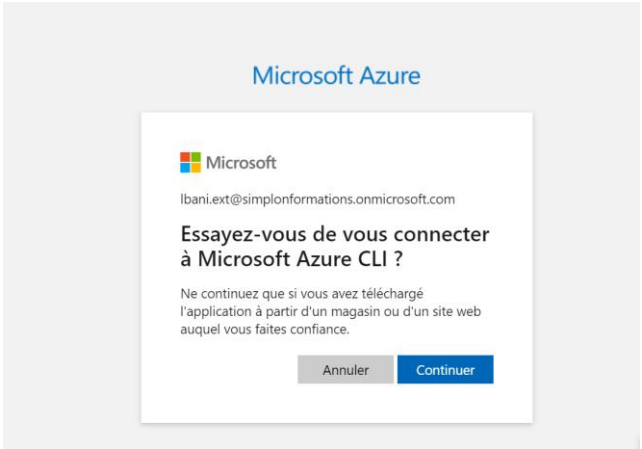
CONFIGURATION

Depuis le terminal azure, dans Bash, se connecter à son compte via la commande "az login".

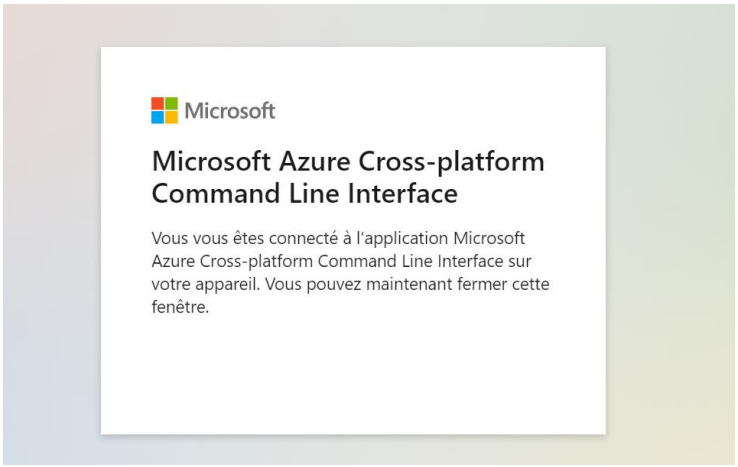
Vous devez vous connecter sur le lien http Microsoft avec le mot de passe donner pour être sur Azure CLI.



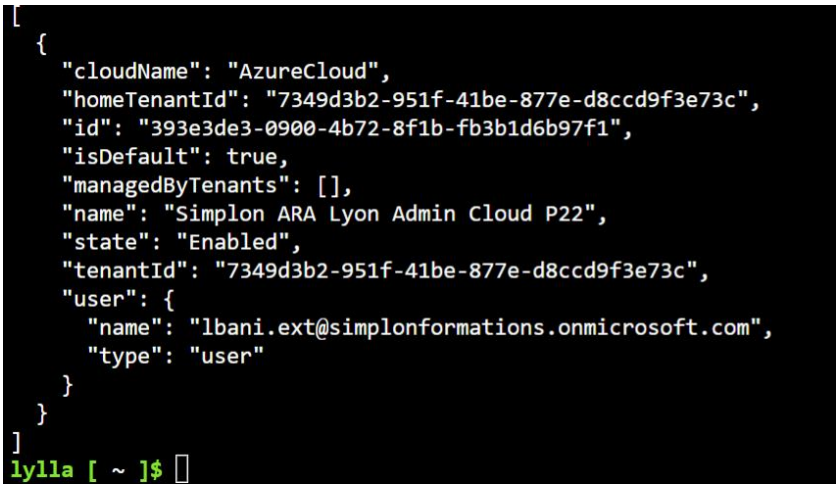
Suivez les instructions indiquées comme ci-dessous :



Une fois l'authentification validé



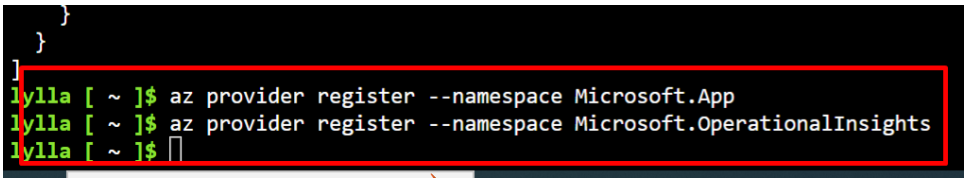
Bienvenue sur azure CLI. Voici ce que vous aurez sur votre écran côté terminal



Installez l'extension de "Azure Container app" pour CLI avec la commande suivante:

Az extension –name container app –upgrade

1. Puis, saisissez les 2 commandes ci-dessous "az provider register" afin de déterminer les 2 fournisseurs de votre espace de travail.



DEFINIR LES VARIABLES D'ENVIRONNEMENT

Comme dans l'exemple ci –dessous :

RESOURCE_GROUP="lyllabbrief10"

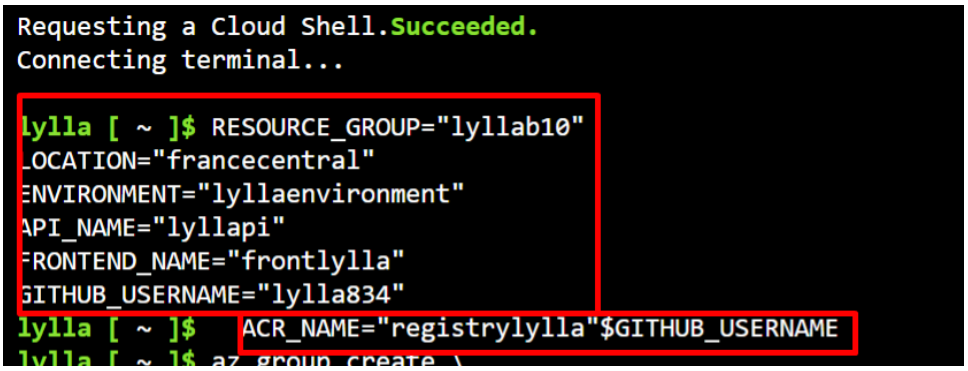
LOCATION="francecentral"

ENVIRONMENT="lyllaenvironment"

API_NAME="node-azure-docker"

FRONTEND_NAME="node-azure-dockerui"

GITHUB_USERNAME="lylla834"



Définir un nom de registre de conteneur avec la variable suivante “ACR_NAME” séparément des variables d’environnement.

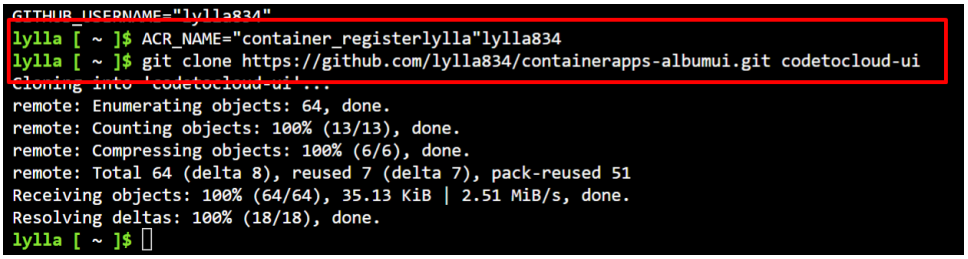
CREATION DU GROUPE DE RESSOURCE

Grâce aux variables créer plus tôt, vous allez pourvoir créer votre groupe de ressource via la commande “az group create” pour organiser vos services conteneurs.

PREPARATION DU REFERENTIEL GITHUB

Depuis votre compte GitHub, utiliser la fonction “Fork” afin de dupliquer l’application choisie. Cliquez sur “create fork” et donnez-lui un nom personnalisé dans votre repositorie dans votre compte GitHub. Ensuite, cloner l’application dans votre terminal avec la commande “git clone” suivit du lien https de l’emplacement de l’application sur GitHub.

Exemple de lien github : git clone https://github.com/lylla834/node-azure-docker.git



CREATION ACR

Vous devez créer un dépôt “Azure Container Registry” avec la commande “az acr create” pour pouvoir héberger l’image conteneur de l’application. La variable “–admin-enabled true” permettra d’activer Docker sur Azure Container Registry. Voir exemple ci –dessous :

az acr create \

--resource-group $RESOURCE_GROUP \

--name $ACR_NAME \

--sku Basic \

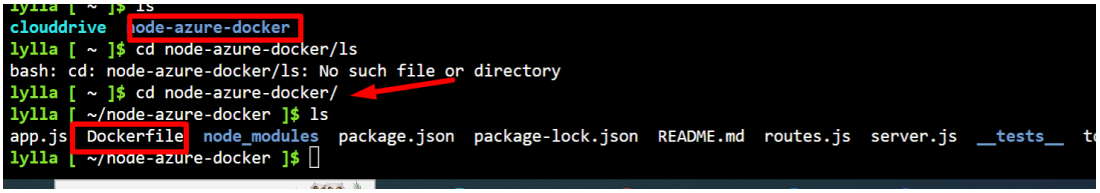
--admin-enabled true

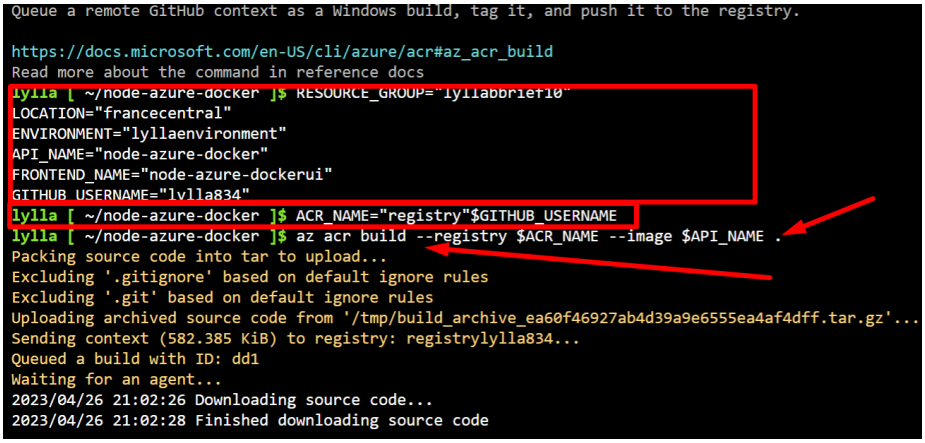
GENERER LE CONTENEUR AVEC ACR

Dans votre terminal azure, en utilisant les commandes “ls” pour voir l’ensemble des dossiers. Ici, on peut voir “clouddrive” et “node-azure-docke"r.

Puis pour aller dans dossier “cd node-azure-docke", enfin “ls”. Allez au niveau du dossier “ Dockerfile”, puis construire son image avec la commande “az acr build”.

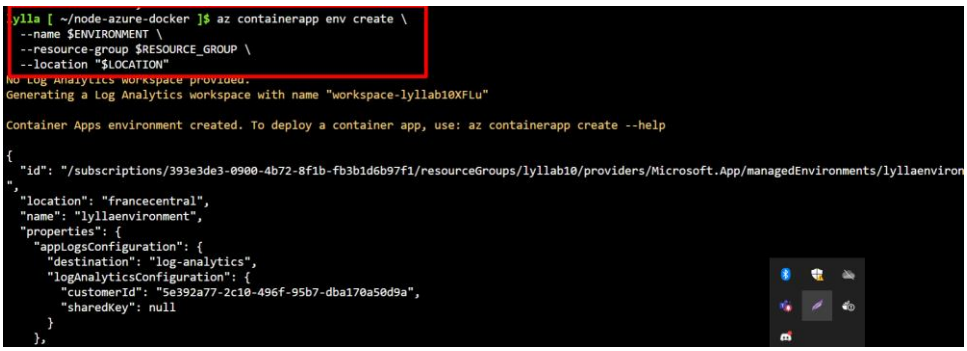
,





CREATION ENVIRONNEMENT CONTAINER APPS

Créer un environnement dans “Azure container apps”. Ce dernier viendra définir une limite sécurisée autour des groupes d’applications de conteneur.



DEPLOYER NOTRE IMAGE

Vous allez créer l’application conteneur via la commande “azcontainerapp create” dans l’environnement définir plus tôt.

```
Run ID: dd1 was successful after 35s
lylla [ ~/node-azure-docker ]$ az containerapp create \
  --name $API_NAME \
  --resource-group $RESOURCE_GROUP \
  --environment $ENVIRONMENT \
  --image $ACR_NAME.azurecr.io/$API_NAME \
  --target-port 3500 \
  --ingress 'external' \
  --registry-server $ACR_NAME.azurecr.io \
  --query properties.configuration.ingress.fqdn
No credential was provided to access Azure Container Registry. Trying to look up credentials...
Adding registry password as a secret with name "registrylylla834azurecrio-registrylylla834"

Container app created. Access your app at https://node-azure-docker.greenwave-83cf3b61.francecentral.azurecontai

"node-azure-docker.greenwave-83cf3b61.francecentral.azurecontainerapps.io"
lylla [ ~/node-azure-docker ]$
```

Faites attention à la ligne “target-port” afin de renseigner le bon numéro de port. On peut trouver le port exposé dans le Dockerfile via la commande “cat Dockerfile” ou modifier via le portail Azure.

Exécutez les commandes suivantes :

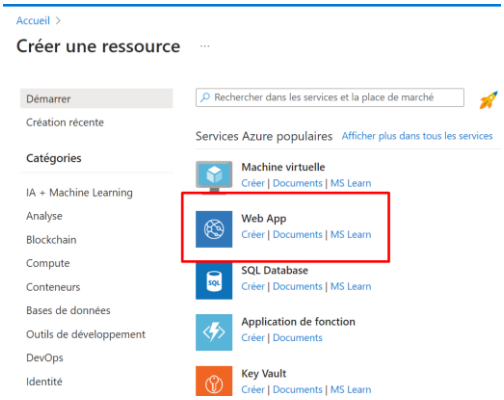
```
lylla [ ~ ]$ npm install
npm ERR! code ENOENT
npm ERR! syscall open
npm ERR! path /home/lylla/package.json
npm ERR! errno -2
npm ERR! enoent ENOENT: no such file or directory, open '/home/lylla/package.json'
npm ERR! enoent This is related to npm not being able to find a file.
npm ERR! enoent

npm ERR! A complete log of this run can be found in:
npm ERR!     /home/lylla/.npm/_logs/2023-04-26T21_11_32_735Z-debug-0.log
lylla [ ~ ]$
```

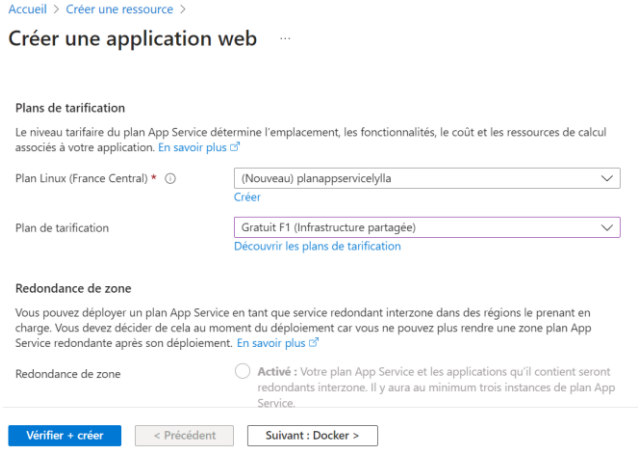
```
  npm audit fix

Run `npm audit` for details.
lylla [ ~/node-azure-docker ]$ npm start

> azure-node-api@1.0.0 start
> node server.js

Server running on localhost:5000
```

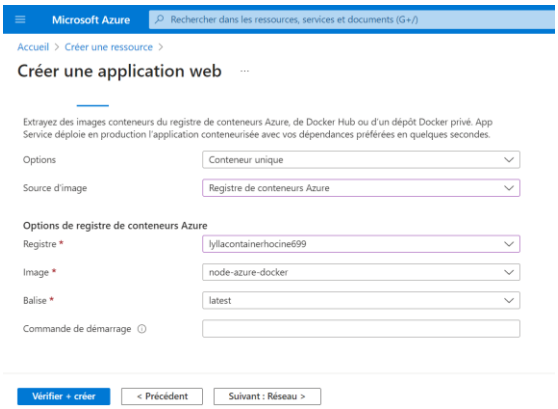
Une fois le conteneur créé, on peut vérifier le déploiement de notre application sur un navigateur web.

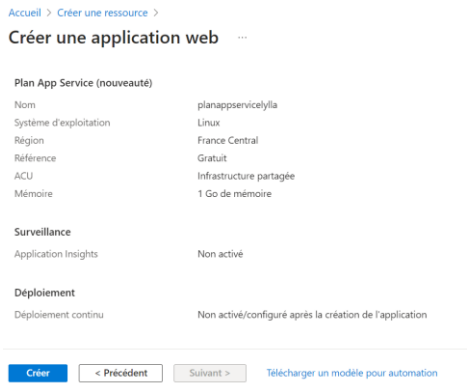
CREATION APPLICATION WEB

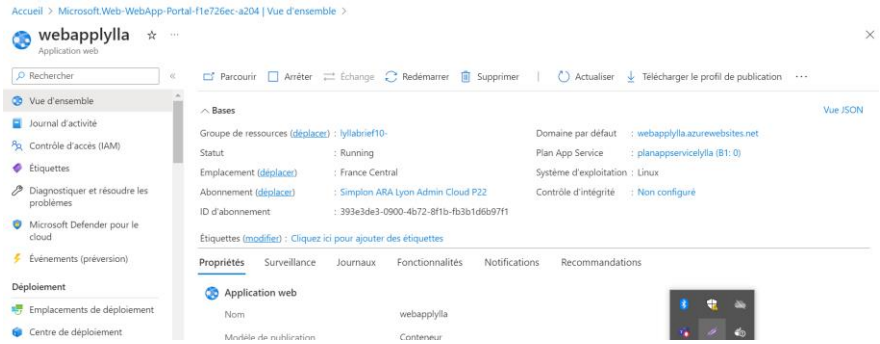
Depuis le portail Azure, dans la barre de recherche, chercher “Web App”, puis cliquez sur “Créer”. Remplissez les champs obligatoires dans les onglets “basics” et “Docker” puis cliquer sur “Vérifier + créer”.


Accueil
Créer une ressource
Démarrer
Création récente
Catégories
IA + Machine Learning
Analyse
Blockchain
Compute
Conteneurs
Bases de données
Outils de développement
DevOps
Identité
Rechercher dans les services et la place de marché
Services Azure populaires
Afficher plus dans tous les services
Machine virtuelle
Créer | Documents | MS Learn
Web App
Créer | Documents | MS Learn
SQL Database
Créer | Documents | MS Learn
Application de fonction
Créer | Documents
Key Vault
Créer | Documents | MS Learn

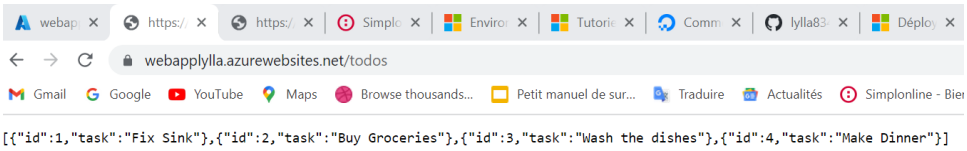

Accueil > Créer une ressource >
Créer une application web
Plans de tarification
Le niveau tarifaire du plan App Service détermine l'emplacement, les fonctionnalités, le coût et les ressources de calcul associés à votre application. En savoir plus
Plan Linux (France Central) *
(Nouveau) planappservicelylla
Créer
Plan de tarification
Gratuit F1 (Infrastructure partagée)
Découvrir les plans de tarification
Redondance de zone
Vous pouvez déployer un plan App Service en tant que service redondant interzone dans des régions le prenant en charge. Vous devez décider de cela au moment du déploiement car vous ne pouvez plus rendre une zone plan App Service redondante après son déploiement. En savoir plus
Redondance de zone
Activé : Votre plan App Service et les applications qu'il contient seront redondants interzone. Il y aura au minimum trois instances de plan App Service.
Vérifier + créer
< Précédent
Suivant : Docker >


Microsoft Azure
Rechercher dans les ressources, services et documents (G+/)
Accueil > Créer une ressource >
Créer une application web
Extrayez des images conteneurs du registre de conteneurs Azure, de Docker Hub ou d'un dépôt Docker privé. App Service déploie en production l'application conteneurisée avec vos dépendances préférées en quelques secondes.
Options
Conteneur unique
Source d'image
Registre de conteneurs Azure
Options de registre de conteneurs Azure
Registre *
lyllacontainerhocine699
Image *
node-azure-docker
Balise *
latest
Commande de démarrage
Vérifier + créer
< Précédent
Suivant : Réseau >


Accueil > Créer une ressource >
Créer une application web
Plan App Service (nouveauté)
Nom
planappservicelylla
Système d'exploitation
Linux
Région
France Central
Référence
Gratuit
ACU
Infrastructure partagée
Mémoire
1 Go de mémoire
Surveillance
Application Insights
Non activé
Déploiement
Déploiement continu
Non activé/configuré après la création de l'application
Créer
< Précédent
Suivant >
Télécharger un modèle pour automation

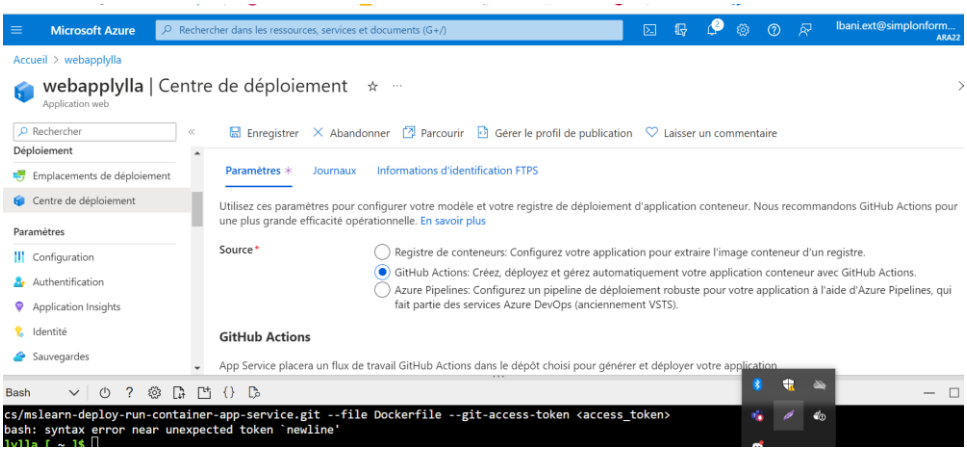


Si votre déploiement a bien fonctionné, vous aurez votre image sur le navigateur web.

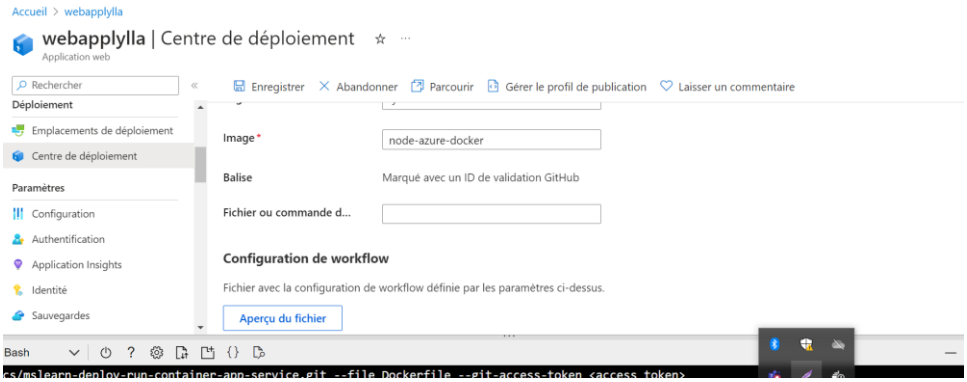


DEPLOIEMENT CONTINU

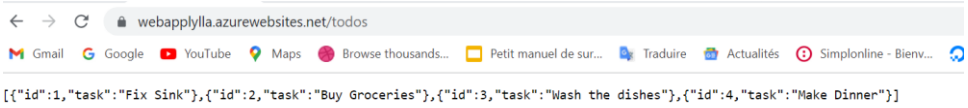
Depuis le portail azure, en passant par “centre de déploiement” sur votre gauche, sélectionnez “GitHub action” à droite, remplissez.



Enregistrez.



Vérifiez votre image sur le navigateur web, pour vérifier le déploiement continu.



[{"id":1,"task":"Fix Sink"},{"id":2,"task":"Buy Groceries"},{"id":3,"task":"Wash the dishes"},{"id":4,"task":"Make Dinner"}]