

Admin Infra : Labo 8

Préliminaires

Pour cette séance, nous continuons d'utiliser notre Debian sur Azure.

Lorsqu'un mot de passe est demandé, nous vous conseillons de mettre « azerty1.» par facilité.

Exercice 1 : Création App Service

Nous continuons à construire l'infrastructure de notre société ITDEV. Celle-ci souhaite utiliser Terraform afin de déployer des applications sur Azure. Nous allons déployer le site nodejs des exoplanètes dans une App Service Azure. **Nous allons dans un premier temps réaliser la création de l'App Service via l'interface Web d'Azure à sa voir**

<https://portal.azure.com>

Instructions

Nous allons créer une App Service (PaaS) sur le site d'Azure pour le site des exoplanètes.

1. Allez sur le site <https://portal.azure.com>
 - a. Tapez App Services dans la zone de recherche
 - b. Créez une Application Web
 - c. Donnez un nom à votre App Service
 - d. Pile d'exécution : node 16 LTS
 - e. OS: Linux
 - f. Région : France Central
 - g. **Plan de tarification : gratuit !!!**
2. Dans la vue d'ensemble(overview) de votre App Service
 - a. Allez dans le Centre de déploiement
 - b. Configurez le déploiement pour utiliser un git externe avec l'URL <https://github.com/olivierchoquet/exoplanets>
 - c. En image ci-dessous

Microsoft Azure

Accueil > Microsoft Web-WebApp-Portal-1bb681fb-9ce2 | Vue d'ensemble > exoplanets-och

exoplanets-och | Centre de déploiement

Rechercher

Enregistrer Abandonner Parcourir Gérer le profil de publication Synchroniser Laisser un commentaire

Paramètres Journaux Informations d'identification FTPS

Vous êtes maintenant dans l'emplacement de production, ce qui n'est pas recommandé pour la configuration de CI/CD. En savoir plus

Déployez et générez du code à partir de votre fournisseur de build et source par défaut. En savoir plus

Source * Git externe

Génération avec le service de build App Service. Changez le fournisseur.

Git externe

Si votre code n'est pas sur GitHub ou BitBucket, vous pouvez utiliser cette option pour synchroniser manuellement votre code à partir du dépôt. Lorsque vous synchronisez votre dépôt, App Service extrait votre code, génère votre application et le déploie sur votre application web.

Dépôt * //github.com/olivierchoquet/exoplanets

Branche * main

Type de dépôt ☒ Public ☐ Privé

d. Retournez dans la vue d'ensemble et cliquez sur le lien de l'application

Microsoft Azure

Accueil > exoplanets-och

exoplanets-och | Application web

Rechercher

Parcourir Arrêter Échange Redémarrer Supprimer Actualiser

Vue d'ensemble

Journal d'activité

Contrôle d'accès (IAM)

Étiquettes

Diagnostiquer et résoudre les problèmes

Microsoft Defender pour le cloud

Événements (préversion)

Vue JSON

⚠ Votre application cible un runtime qui est obsolète. Mettez à jour votre application pour continuer à recevoir des correctifs de sécurité et de bogues.

Bases

Groupe de ressources (déplacer) [debian12-och-group](#)

Statut En cours d'exécution

Emplacement (déplacer) France Central

Domaine par défaut [exoplanets-och.azurewebsites.net](#)

Plan App Service ASP-debian12ochgroup-99b6 (F1: 1)

Système d'exploitation Linux

e. Attendez un peu, rechargez la page et vous devriez voir le site des exoplanètes

Voilà vous avez réussi à déployer une application nodejs en PaaS.

Exercice 2 : Découverte Terraform

Exécutez simplement les instructions ci-dessous pour découvrir Terraform et répondez ensuite aux questions :

Référez-vous à la section 13.6 Cloud (Terraform) du syllabus.

Instructions

L'objectif est donc de déployer le site nodejs des exoplanètes depuis le repository public <https://github.com/olivierchoquet/exoplanets> dans une App Service sous Azure via Terraform.

1. Installez Terraform sur votre VM Azure
 - a. Voir installation de Terraform dans le syllabus
2. Installez la console Azure CLI sur votre VM Azure
 - a. Voir utilisation de Terraform dans le syllabus
3. Récupérez le fichier terraform **main.tf** présent sur MooVin
4. Créez un répertoire **terraform_exoplanets** sur votre VM Azure
5. Transférez le fichier main.tf sur votre VM Azure dans le répertoire terraform_exoplanets
6. Lancez la commande **terraform init** dans le répertoire terraform_exoplanets
7. Lancez la commande **terraform plan** dans le répertoire terraform_exoplanets
8. Lancez la commande **terraform apply** dans le répertoire terraform_exoplanets

Répondez aux questions suivantes :

Vous trouverez les réponses dans <https://portal.azure.com> et/ou dans le fichier main.tf

Quelle est l'adresse de déploiement de votre application ? Fonctionne-t-elle correctement ?

Quel est le nom de votre App Service ?

A quoi correspond le paramètre sku_name dans le fichier main.tf ?

A quoi correspond un groupe de ressources et à quoi correspond un plan de service ?

Pouvez-vous retrouver sur <https://portal.azure.com> l'information sur le runtime de l'application (node_version = "16-lts") et le déploiement du code ?

Exercice 3 : Terraform site PHP

Essayez maintenant de déployer via Terraform une App Service avec le site PHP des bonnes nouvelles.

1. Récupérez le site des bonnes nouvelles sur MooVin
2. Ne vous préoccupez pas de la base de données
3. Essayez de déployer le site des bonnes nouvelles manuellement « au clic » via <https://portal.azure.com> dans un premier temps
4. Ensuite inspirez-vous de l'exercice précédent et de ce que vous avez fait manuellement pour déployer le site PHP des bonnes nouvelles via Terraform

Exercice 4 (BONUS) : Terraform DockerHub

Déployer une App Service qui contiendra le syllabusHTML via Terraform. Le syllabusHTML sera déployé à partir d'une image Docker présente sur le DockerHub à savoir l'image : `olivierchoquet/syllabushtml:alpine`.