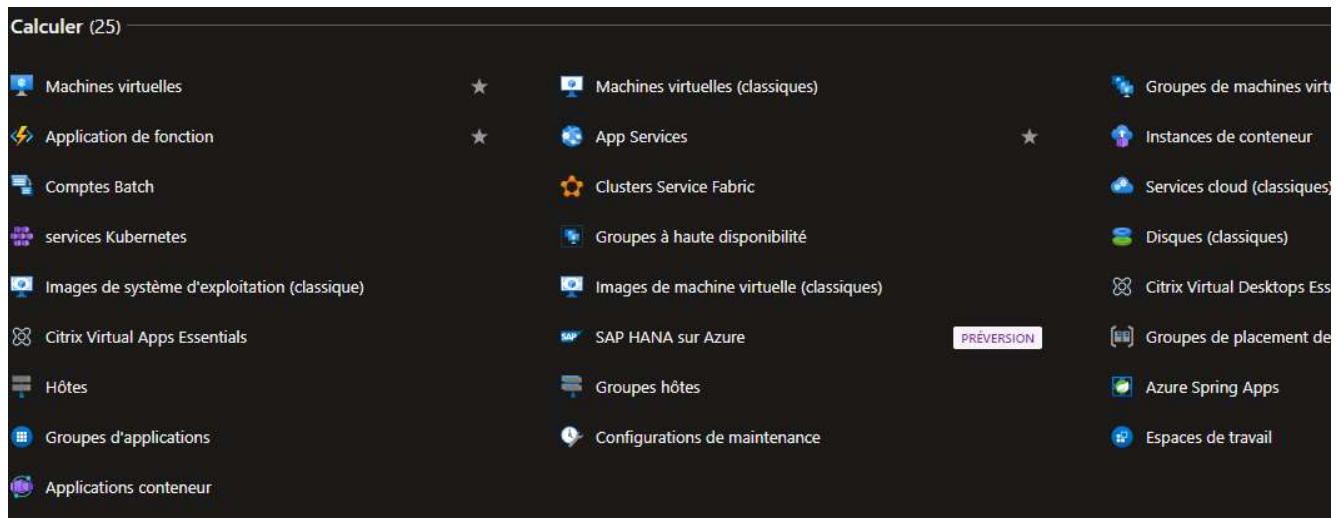


Azure – Plateforme de calcul (virtualisation) - Partie 1

mercredi 8 mars 2023 13:35

Formateur : Ihab ABADI

Voici la liste de tous les services de calcul (virtualisation) Azure :



Machine virtuelle :

Il y a plusieurs types de machine disponible pour répondre à divers besoins :



Création de la VM :

On définit l'abonnement et les groupes de ressources affecté :



[Créer nouveau](#)

On définit les détails de l'instance :

Détails de l'instance

Nom de la machine virtuelle *

vm-benoit-m2i-1

Région *

(US) East US

Options de disponibilité

Zone de disponibilité


Zone de disponibilité *

Zone 1

Type de sécurité

Standard

Image *

 Ubuntu Server 20.04 LTS - x64 Gen2

[Voir toutes les images](#) | [Configurer la génération de machine virtuelle](#)

On peut définir la taille de notre instance : c'est-à-dire sa configuration :

Taille *

Standard_B1s - 1 processeur virtuel, 1 Gio de mémoire (7,59 \$US/mois)

[Voir toutes les tailles](#)

Il faut faire très attention au coût de la taille demandé :

Sélectionner une taille de machine virtuelle

Rechercher par taille de ...

Afficher le coût : Mensuel

Processeurs virtuels : Tous

RAM (Gio) : Tous

Ajouter un filtre

Affichage de 707 tailles de machine virtuelle.

Abonnement : Abonnement Azure 1

Région : East US

Taille actuelle : Standard_B1s

Image : Ubuntu Server 20.04 LTS

[En savoir plus sur les tailles de machine virtuelle](#)

Grouper par série

Taille de la machine ...	Type	Processeurs virtuels	RAM (Gio)	Disques de donn...	Opérations d'E/S ...	Stockage temporaire...
Les plus courantes chez les utilisateurs Azure						
Série D v4						
Série B						
Série DC						
Série E v5						
Série E v4						
Série F v2						
Série N						
NV4as_v4	GPU	4	14	8	8x500	88
NV8as_v4	GPU	8	28	16	16x500	176
Série D v3						
Série E v3						
Série D v2						
Tailles de machine virtuelle de stockage non-Premium						
Tailles de génération précédente						
La taille n'est pas disponible						

Pour cet exemple, nous resteront avec l'authentification par mot de passe :

Compte d'administrateur

Type d'authentification

☐ Clé publique SSH

☒ Mot de passe

Nom d'utilisateur *

userazure

Mot de passe *

.....

Confirmer le mot de passe *

.....

On peut régler les ports d'entrée :

Règles des ports d'entrée

Sélectionnez les ports réseau de machine virtuelle accessibles publiquement à partir d'Internet. Vous pouvez spécifier un accès réseau plus limité ou granulaire sous l'onglet Mise en réseau.

Ports d'entrée publics * ⓘ ☐ Aucun ☒ Autoriser les ports sélectionnés

Sélectionner des ports d'entrée *

⚠ Cela permet à toutes les adresses IP d'accéder à votre machine virtuelle. Ceci est recommandé uniquement pour les tests. Utilisez les contrôles avancés de l'onglet Mise en réseau pour créer des règles afin de limiter le trafic entrant sur les adresses IP connues.

On créer le stockage :

De base **Disques** Mise en réseau Administration Monitoring Paramètres avancés Étiquettes Vérifier + créer

Les machines virtuelles Azure ont un disque de système d'exploitation et un disque temporaire pour le stockage à court terme. Vous pouvez attacher des disques de données supplémentaires. La taille de la machine virtuelle détermine le type de stockage que vous pouvez utiliser et le nombre de disques de données autorisés. [En savoir plus](#) ⓘ

VM disk encryption

Azure disk storage encryption automatically encrypts your data stored on Azure managed disks (OS and data disks) at rest by default when persisting it to the cloud.

Chiffrement sur l'hôte ⓘ ☐

Le chiffrement sur l'hôte n'est pas inscrit pour l'abonnement sélectionné. [En savoir plus sur l'activation de cette fonctionnalité](#) ⓘ

OS disk

Taille du disque du système d'exploitation * ⓘ

Type de disque de système d'exploitation * ⓘ

La taille de machine virtuelle sélectionnée prend en charge les disques Premium. Nous vous recommandons d'utiliser des disques SSD Premium pour les charges de travail avec des taux IOPS élevés. Les machines virtuelles équipées de disques SSD Premium remplissent les conditions de connectivité à 99,9 % du contrat SLA.

Supprimer avec la machine virtuelle ⓘ ☒

Key management ⓘ

Activer la compatibilité avec les disques Ultra ⓘ ☐

Disques de données pour vm-benoit-m2i-1

Vous pouvez ajouter et configurer des disques de données supplémentaires pour votre machine virtuelle ou attacher des disques existants. Cette machine virtuelle est également fournie avec un disque temporaire.

N...	Nom	Taille (...)	Type de disque	Mise en ca...	Supprimer avec la r
------	-----	--------------	----------------	---------------	---------------------

[Créer un disque et l'attacher](#) [Attacher un disque existant](#)

La partie mise en réseau qui va nous permettre configurer le réseau dans lequel va se retrouver notre VM :

De base Disques **Mise en réseau** Administration Monitoring Paramètres avancés Étiquettes Vérifier + créer

Définissez la connectivité réseau de votre machine virtuelle en configurant les paramètres de la carte d'interface réseau. Vous pouvez contrôler les ports et la connectivité entrante/sortante avec des règles de groupe de sécurité, ou placer derrière une solution d'équilibrage de charge existante. [En savoir plus](#) ⓘ

Interface réseau

Quand vous créez une machine virtuelle, une interface réseau est créée pour vous.

Réseau virtuel * ⓘ

[Créer](#)

Sous-réseau * ⓘ

[Créer](#)

Adresse IP publique ⓘ

[Créer](#)

Groupe de sécurité réseau de la carte réseau ⓘ ☐ Aucun ☒ De base ☐ Paramètres avancés

Ports d'entrée publics * ⓘ ☐ Aucun

☒ Autoriser les ports sélectionnés

Sélectionner des ports d'entrée SSH (22)

Cela permet à toutes les adresses IP d'accéder à votre machine virtuelle.
Ceci est recommandé uniquement pour les tests. Utilisez les contrôles avancés de l'onglet Mise en réseau pour créer des règles afin de limiter le trafic entrant sur les adresses IP connues.

Supprimer l'adresse IP publique et la carte réseau lors de la suppression de la machine virtuelle ☒

Activer les performances réseau accélérées ☐ La taille de machine virtuelle sélectionnée ne prend pas en charge la mise en réseau accélérée.

Équilibrage de charge

Vous pouvez placer cette machine virtuelle dans le pool de back-ends d'une solution d'équilibrage de charge Azure existante. [En savoir plus](#)

Placer cette machine virtuelle derrière une solution d'équilibrage de charge ☐

Les étiquettes permettent de définir le nom de nos ressources qui vont être créer :

Créer une machine virtuelle ...

De base Disques Mise en réseau Administration Monitoring Paramètres avancés Étiquettes Vérifier + créer

Les étiquettes sont des paires nom/valeur qui vous permettent de catégoriser les ressources et de voir une facturation centralisée en appliquant la même étiquette à plusieurs ressources et groupes de ressources. [En savoir plus sur les étiquettes](#)

Notez que si vous créez des étiquettes, puis que vous changez les paramètres de ressource sous d'autres onglets, vos étiquettes sont automatiquement mises à jour.

Nom	Valeur	Ressource
Name	vm-benoit-m2i-1	13 sélectionné
	vm-benoit-m2i-1	13 sélectionné

On vérifie notre configuration : apparemment tout est ok pour la création

Validation réussie

De base Disques Mise en réseau Administration Monitoring Paramètres avancés Étiquettes Vérifier + créer

Le coût donné ci-dessous est une estimation et non le prix final. Veuillez utiliser [Calculatrice de prix](#) pour tous vos besoins en matière de tarification.

Price

1 X Standard B1s
by Microsoft
[Terms of use](#) [Privacy policy](#)

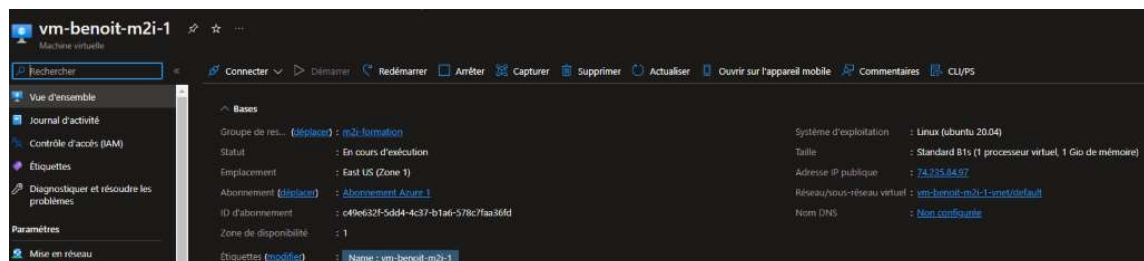
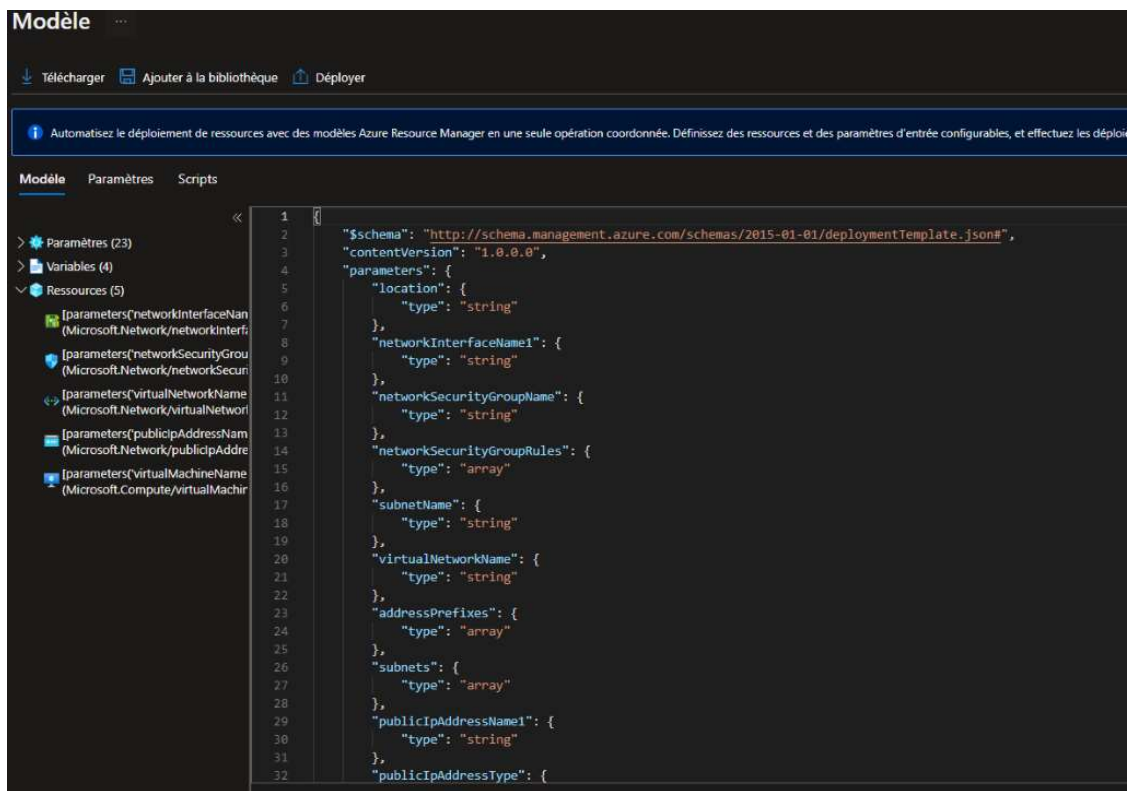
Subscription credits apply
0,0104 USD/hr
[Pricing for other VM sizes](#)

TERMS

By clicking "Créer", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

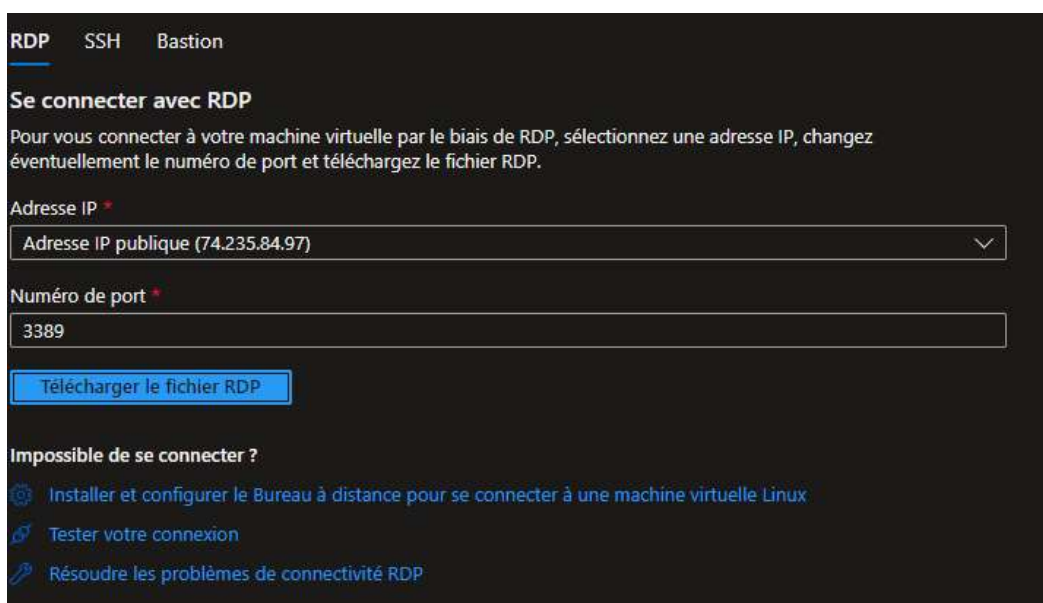
Azure génère le modèle JSON qu'on peut récupérer pour automatiser le déploiement de nos instances si on a besoin de reprendre cette configuration à l'avenir :

[View JSON](#)
[Download JSON](#)
[Copy and paste JSON](#)



Pour se connecter à notre instance, on peut utiliser le [RDP](#) ou [SSH](#) :

(Si on veut se connecter en RDP : il faut ouvrir le port **3389** et avoir une distribution Windows)



Fournir des commentaires

Faites-nous part de votre expérience de RDP

```
PS C:\Users\Administrateur> ssh userazure@74.235.84.97
The authenticity of host '74.235.84.97 (74.235.84.97)' can't be established.
```

```
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

userazure@vm-benoit-m2i-1:~$
```

Création script automatisé :



Machine Virtuelle

Commande Network

- Création d'un réseau virtuel avec Subnet

```
az network vnet create --name custom-vnet-benoit --resource-group m2i-formation --address-prefixes
10.0.0.0/16 --subnet-name custom-subvnet-benoit-1 --subnet-prefix 10.0.0.0/24
```

- Création d'une carte réseau

```
az network nic create --name custom-nic-benoit --resource-group m2i-formation --vnet-name custom-vnet-
benoit --subnet custom-subvnet-benoit-1
```

- Création VM

```
az vm create --name benoit-cli-1 --resource-group m2i-formation --image UbuntuLTS --size Standard_B1ls  
--admin-username benoit --admin-password Formation123456 --nics custom-nic-benoit
```

- Rattacher une adresse IP publique à un network

```
az network nic ip-config update --name ipconfig1 --resource-group m2i-formation --nic-name custom-nic-  
benoit --public-ip-address benoitcustompublicip
```

Exercice :

Exercice *Création d'une machine virtuelle*

Créer une machine virtuelle Ubuntu dans Azure avec les caractéristiques suivantes :

- Nom : vm-ubuntu
- Taille de la machine : Standard_DS2_v2
- Nom d'utilisateur : azureuser
- Mot de passe : Azure123456!
- Ressource de groupe : formation-m2i
- VNet : vnet-ubuntu
- Sous-réseau : subnet-ubuntu
- Adresse IP statique : 10.0.0.4

Code utilisé pour cet exercice :

```
#!/bin/bash  
### Création instance Ubuntu dans Azure ###  
  
#Création du vnet de notre instance avec subnet  
  
az network vnet create --name vnet-ubuntu-benoit \  
  --resource-group m2i-formation \  
  --address-prefixes 10.0.0.0/16 \  
  --subnet-name subnet-ubuntu-benoit \  
  --subnet-prefix 10.0.0.0/24 \  
  --location northcentralus
```

```
  #Création carte réseau  
  
az network nic create --name nic-ubuntu-benoit \  
  --resource-group m2i-formation \  
  --location northcentralus
```



```
--vnet-name vnet-ubuntu-benoit \  
--subnet subnet-ubuntu-benoit \  
--location northcentralus
```

#Création de l'instance

```
az vm create --name vm-ubuntu-benoit \  
--resource-group m2i-formation \  
--image UbuntuLTS \  
--size Standard_DS2_v2 \  
--private-ip-address 10.0.0.4 \  
--admin-username azureuser \  
--admin-password Azure123456! \  
--nics nic-ubuntu-benoit \  
--location westus
```

Petit rappel : ne pas oublier qu'une clé SSH doit être en mode lecture pour être utilisée : faire les modifications quand on récupère notre clé SSH

```
Administrateur@Salle_3_Z MINGW64 ~/Desktop/Azure  
$ chmod 400 cle_azure_ihab.pem  
  
Administrateur@Salle_3_Z MINGW64 ~/Desktop/Azure  
$ ssh -i cle_azure_ihab.pem azure@20.56.128.14  
The authenticity of host '20.56.128.14 (20.56.128.14)' can't be established.  
ED25519 key fingerprint is SHA256:gVea8n0MCa1p7MprGogK6J5b7x6f33Gi+aTQZo2wN+Y.  
This key is not known by any other names.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? ye  
Please type 'yes', 'no' or the fingerprint: yes  
Warning: Permanently added '20.56.128.14' (ED25519) to the list of known hosts.  
azure@20.56.128.14: Permission denied (publickey).
```

```
Administrateur@Salle_3_Z MINGW64 ~/Desktop/Azure  
$
```

TP Création d'une machine virtuelle

Objectif : Configurer une machine virtuelle Linux dans Microsoft Azure pour exécuter un serveur web, y accéder à distance et sécuriser l'accès à cette machine.

- Créez une machine virtuelle Linux (par exemple, Ubuntu) dans Azure avec une adresse IP publique statique.
- Connectez-vous à la machine virtuelle en utilisant SSH.
- Installez et configurez un serveur web Apache sur la machine virtuelle.
- Créez une page d'accueil pour le site web hébergé sur le serveur web Apache.
- Configurez le pare-feu Linux pour permettre l'accès au serveur web.
- Créez une règle de sécurité de groupe réseau pour autoriser l'accès HTTP à la machine virtuelle.
- Testez l'accès au site web en utilisant l'adresse IP publique de la machine virtuelle.
- Créez un groupe de sécurité pour permettre l'accès SSH à la machine virtuelle.
- Configurez le groupe de sécurité pour n'accepter les connexions SSH que depuis une plage d'adresses IP spécifique.
- Testez l'accès SSH à la machine virtuelle en utilisant le client SSH.
- Modifiez la configuration de la machine virtuelle pour désactiver l'accès SSH avec l'adresse IP publique.

Sur notre VM :

Installer apache :

Sudo apt install apache2

Modification de la page web de base


```
GNU nano 4.8 index.html Mod
<html>
<head>
  <meta charset='utf-8'>
  <meta http-equiv='X-UA-Compatible' content='IE=edge'>
  <title>Page Title</title>
  <meta name='viewport' content='width=device-width, initial-scale=1'>
  <link rel='stylesheet' type='text/css' media='screen' href='main.css'>
  <script src='main.js'></script>
</head>
<body>
  <div>
    <h1>Hello From Ihab VM</h1>
  </div>
</body>
</html>
Save modified buffer?
```

```
azureuser@vm-ubuntu-ihab:/var/www/html$ sudo ufw allow http
Rules updated
Rules updated (v6)
azureuser@vm-ubuntu-ihab:/var/www/html$
```

Priorité ↑↓	Nom ↑↓	Port ↑↓	Protocole ↑↓	Source ↑↓	Destination ↑↓	Action ↑
▼ Règles de sécurité de trafic entrant						
300	SSH	22	TCP	N'importe lequel	N'importe lequel	✓ Allow
320	HTTP	80	TCP	N'importe lequel	185.31.149.99	✓ Allow
340	HTTPS	443	TCP	N'importe lequel	N'importe lequel	✓ Allow

Même exercice en powershell :

TP 2 Création d'une machine virtuelle azure powershell

Objectif : Dans ce TP, vous allez apprendre à créer et gérer des machines virtuelles Azure en utilisant Azure PowerShell. Vous allez créer une machine virtuelle, vous allez la démarrer et l'arrêter, vous allez vous y connecter via SSH et vous allez supprimer la machine virtuelle.

- Ecrire un script en powershell qui permet de créer les différentes ressources nécessaires à la création d'une VM .

```
> tp1.ps1
1  # Objet avec les informations de l'adresse IP
2  $ip = @{
3      Name = 'public-ip-az-powershell'
4      ResourceGroupName = 'm2i-formation'
5      AllocationMethod = 'Static'
6      IpAddressVersion = 'IPv4'
7  }
8  # Création de l'adresse ip
9  $ipConfig = New-AzPublicIpAddress @ip
10
```

```
# Objet avec les informations du groupe de sécurité

$netSecurityGroup = @{
    Name = "security-group-az-powershell-ihab"
    ResourceGroupName = "m2i-formation"
```

```
}
ResourceGroupName = 'm2i-formation'
```

```
# Création d'un security group

$nsc = New-AzNetworkSecurityGroup @netSecurityGroup
```

Création des règles de sécurité

```
des règles de sécurité

New-AzNetworkSecurityRuleConfig -Name web-rule -Description "Allow HTTP" `
-Access Allow -Protocol Tcp -Direction Inbound -Priority 101 -SourceAddressPrefix
Internet -SourcePortRange * -DestinationAddressPrefix * -DestinationPortRange 80

New-AzNetworkSecurityRuleConfig -Name web-rule -Description "Allow HTTP" `
-Access Allow -Protocol Tcp -Direction Inbound -Priority 101 -SourceAddressPrefix
Internet -SourcePortRange * -DestinationAddressPrefix * -DestinationPortRange 80
```

```
$nsc | Add-AzNetworkSecurityRuleConfig -NetworkSecurityRuleConfig $httpRule
$nsc | Add-AzNetworkSecurityRuleConfig -NetworkSecurityRuleConfig $sshRule

Set-AzNetworkSecurityGroup -NetworkSecurityGroup $nsc
```

tp1.ps1

```
1 # Objet avec les informations de l'adresse IP
2 $ip = @{
3     Name = 'public-ip-az-powershell-2'
4     ResourceGroupName = 'm2i-formation'
5     AllocationMethod = 'Static'
6     IpAddressVersion = 'IPv4'
7     Location = "eastus"
8 }
9 # Création de l'adresse ip
10 $ipConfig = New-AzPublicIpAddress @ip
11
12 # Objet avec les informations du groupe de sécurité
13
14 $netSecurityGroup = @{
15     Name = "security-group-az-powershell-ihab-2"
16     ResourceGroupName = "m2i-formation"
17     Location = "eastus"
18 }
19
20 # Création d'un security group
21 $nsc = New-AzNetworkSecurityGroup @netSecurityGroup

23 # Création des règles de sécurité
24 $nsc | Add-AzNetworkSecurityRuleConfig -Name web-rule-2 -Description "Allow HTTP" `
25     -Access Allow -Protocol Tcp -Direction Inbound -Priority 102 -SourceAddressPrefix `
26     Internet -SourcePortRange * -DestinationAddressPrefix * -DestinationPortRange 80 | Set-AzNetworkSecurityGr
27
28 $nsc | Add-AzNetworkSecurityRuleConfig -Name ssh-rule-2 -Description "Allow SSH" `
29     -Access Allow -Protocol Tcp -Direction Inbound -Priority 101 -SourceAddressPrefix `
30     Internet -SourcePortRange * -DestinationAddressPrefix * -DestinationPortRange 22 | Set-AzNetworkSecurityGr
31
```

```

32 # Création d'une vm
33
34 $infoVm = @{
35     Name = 'vm-az-powershell-ihab'
36     ResourceGroupName = 'm2i-formation'
37     Location = "eastus"
38     SecurityGroupName = $nsc
39     Image = "UbuntuLTS"
40     PublicIpAddressName = $ipConfig
41 }
42
43 $vm = New-AzVM @infoVm -Credential (Get-Credential)

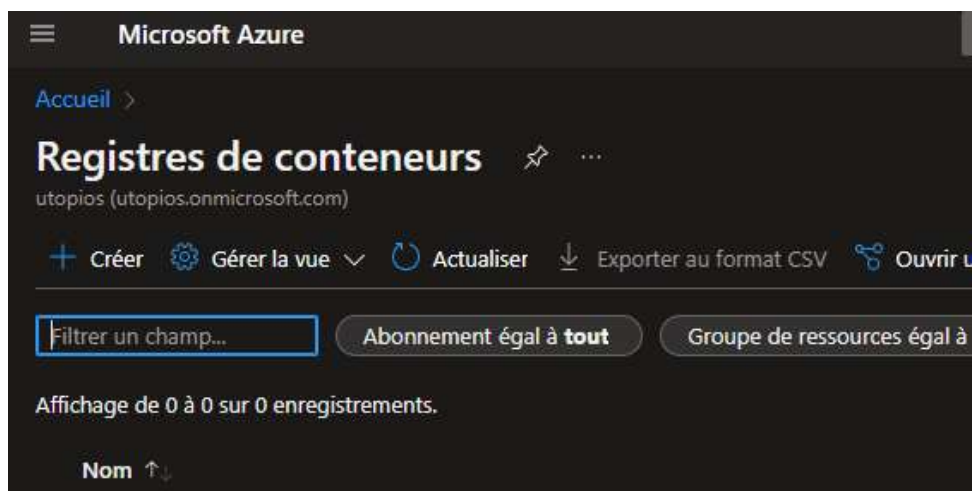
```

Infrastructure conteneurisé avec Azure (équivalent ECR sur AWS) :

ECR -> Serveur de Registre

Docker est composé de 3 blocs : API – Run - DockerHub

Pour créer un ECR : Il faut aller dans le menu Registres de conteneurs



Créer un Registre de conteneurs

Fonctions de base Réseau Chiffrement Étiquettes Vérifier + créer

Azure Container Registry vous permet de générer, stocker et gérer les artefacts et images conteneur dans un registre privé pour tous les types de déploiement de conteneurs. Utilisez les registres de conteneurs Azure avec vos pipelines de développement et de déploiement de conteneurs existants. Utilisez Azure Container Registry Tasks pour générer des images conteneur dans Azure à la demande, ou pour automatiser les builds déclenchées par les mises à jour du code source, les mises à jour de l'image de base d'un conteneur ou les minuteurs. [En savoir plus](#)

Détails du projet

Abonnement * Abonnement Azure 1

Groupe de ressources * m2i-formation
[Créer nouveau](#)

Détails de l'instance

Nom du Registre * benoitECR [.azurecr.io](#)

Emplacement * North Central US

Zones de disponibilité [?](#) ☐ Activé

[i](#) Les zones de disponibilité sont activées sur les registres premium et dans les régions qui les prennent en charge. [En savoir plus](#)

SKU * [?](#) Standard

A savoir que l'ECR azure est disponible de n'importe où



Créer un Registre de conteneurs

Fonctions de base Réseau Chiffrement Étiquettes Vérifier + créer

Connectivité réseau

Vous pouvez vous connecter à ce registre de manière publique via des adresses IP publiques, ou de manière privée via un point de terminaison privé. [En savoir plus](#)

Configuration de la connectivité

☒ Accès public (tous réseaux)

☐ Accès privé (Recommandé)

⚠ L'accès privé (recommandé) n'est disponible que pour le SKU Premium.

Créer un Registre de conteneurs

Fonctions de base Réseau Chiffrement Étiquettes Vérifier + créer

Le chiffrement du service Azure Container Registry protège vos données au repos. Azure Container Registry chiffre vos images et autres artefacts quand ils sont envoyés (push) vers votre registre, et les déchiffre automatiquement quand vous les tirez (pull). [En savoir plus](#)

Clé gérée par le client

☐ Activé ☒ Désactivé

⚠ La clé gérée par le client est disponible uniquement pour la référence SKU Premium.

✓ Validation réussie

Fonctions de base Réseau Chiffrement Étiquettes Vérifier + créer

Détails du registre

Fonctions de base	
Nom du Registre	benoitECR
Abonnement	Abonnement Azure 1
Groupe de ressources	m2i-formation
Emplacement	North Central US
Zones de disponibilité	Désactivé
SKU	Standard

Réseau	
Accès au réseau public	Oui

Chiffrement	
Clé gérée par le client	Désactivé
Identité	Aucun
Key Vault	Aucun
Clé de chiffrement	Aucun
Version	Aucun

Étiquettes	
Name	benoitECR

🗑 Supprimer ⏸ Annuler 📶 Redéployer ⬇ Télécharger 🔄 Actualiser

✓ **Votre déploiement a été effectué**

🏠	Nom du déploiement : Microsoft.ContainerRegistry_1	Heure de début : 09/03/2023 15:34:28
	Abonnement : Abonnement Azure 1	ID de corrélation : 34675a17-e6de-4318-9a74-387cc8ebb572 📄
	Groupe de ressources : m2i-formation	

⌵ **Détails du déploiement**

⌶ **Étapes suivantes**

Accéder à la ressource

Envoyer des commentaires

Partagez votre expérience avec le déploiement

→ Déplacer	Supprimer	Mettre à jour
Bases		
Groupe de res... (déplacer) : m2i-formation	Serveur de connexion	: benoitcr.azurecr.io
Emplacement : North Central US	Date de création	: 09/03/2023 15:34 UTC+1
Abonnement (déplacer) : Abonnement Azure 1	SKU	: Standard
ID d'abonnement : c49e632f-5dd4-4c37-b1a6-578c7faa36fd	État de l'approvisionnem...	: Opération réussie
	Suppression réversible (p...	: Désactivé

Important à savoir : sur un conteneur -> le root du conteneur peut effectuer des requêtes sur l'hôte si le volume est partagé

différence cmd et entrypoint : <https://aws.amazon.com/fr/blogs/france/demystifier-entrypoint-et-cmd-dans-docker/#:~:text=Ils%20peuvent%20tous%20deux%20%C3%AAtre,arguments%20sont%20converties%20en%20listes.>

Quand devrais-je utiliser ENTRYPOINT? Qu'en est-il de CMD?

Supposons que nous construisons notre propre `Dockerfile` pour un projet. À ce stade, nous comprenons les mécanismes de fonctionnement des instructions `ENTRYPOINT` et comment ils fonctionnent ensemble pour construire une liste d'arguments par défaut pour un conteneur. Mais maintenant, nous devons savoir lequel choisir : quand est-il préférable d'utiliser `ENTRYPOINT` et quand est-il préférable d'utiliser `CMD` ?

Le choix que vous faites est essentiellement "artistique" et dépendra beaucoup de votre cas d'utilisation. Notre expérience, cependant, est que l'entrée `ENTRYPOINT` convient tous les cas que nous avons rencontrés. Considérez les cas d'utilisation suivants:

- Wrappers

Certaines images contiennent un «wrapper» qui décore un programme existant ou le prépare autrement à une utilisation dans un environnement conteneurisé. Par exemple, si votre service ait été écrit pour lire sa configuration à partir d'un fichier plutôt qu'à partir de variables d'environnement. Dans une telle situation, vous pouvez inclure un script d'encapsulation qui génère le fichier de configuration de l'application à partir des variables d'environnement, puis lance l'application en appelant `exec /path/to/app` à la fin. Déclarer un pointeur `ENTRYPOINT` dans le wrapper est un excellent moyen de s'assurer que le wrapper est toujours exécuté, quels que soient les arguments passés `docker run`.

- Images à usage unique

Si votre image est conçue pour ne faire qu'une chose – par exemple, exécuter un serveur Web – utilisez l'instruction `ENTRYPOINT` pour spécifier le chemin d'accès au fichier binaire et tous les arguments obligatoires. Un exemple classique est l'image `nginx`, dont le seul but est d'exécuter le serveur Web nginx. Cela se prête à une ligne de commande agréable et plus naturelle pour l'invocation: `docker run nginx`. Ensuite, vous pouvez ajouter des arguments de programme intuitivement sur la ligne de commande, par exemple `docker run nginx -c /test.conf`, comme vous le feriez si vous exécutiez nginx sans Docker.

- Images multi-mode

Il s'agit également un modèle commun pour les images qui prennent en charge plusieurs « modes » d'utiliser le premier argument pour spécifier un verbe qui mappe au mode d'exécution, par exemple shell, migrate ou debug. Pour de tels cas d'utilisation, nous recommandons de définir avec `ENTRYPOINT` le pointeur vers un script qui analyse l'argument et fait le bon choix en fonction de sa valeur:

```
ENTRYPOINT ["/bin/parse_container_args"]
```

Les arguments seront transmis au point d'entrée sur l'invocation avec `ARGV[1..n]`, ou `$1`, `$2`, etc.

Création image avec DockerFile

AZURE	Dockerfile > Python > app.py
✓ Dockerfile \ Python	1 print('Hello, world!')
app.py	
Dockerfile	

Dockerfile > Python > Dockerfile > ...
1 FROM python:3.9
2 WORKDIR /app
3 COPY app.py app.py
4 CMD ["python", "app.py"]

```
PS C:\Atelier\Azure\Dockerfile\Python> docker build -t image-test-benoit-azure .
[+] Building 11.7s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 31B
```

```
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/python:3.9
```

```
Administrateur@LIL-JTJ3KN3 MINGW64 /c/Atelier/Azure
$ docker login benoitcr.azurecr.io --username benoitcr -password BqVDNuAJKpejJ0fxCIIdNPB7iNI2p8fI24Q9KhTQCH6+ACRBo3rmJ
accepts at most 1 arg(s), received 2
```

```
PS C:\Users\Administrateur\Desktop\Azure\dockerfile\python> docker push demo-python-image
Using default tag: latest
The push refers to repository [docker.io/library/demo-python-image]
9c119468e5e3: Preparing
4c8a37c5a72f: Preparing
28fa86444e63: Preparing
85158601c190: Preparing
0cdca02a73c8: Preparing
0baa05faf31a: Waiting
ac504d030134: Waiting
52ebb9a789db: Waiting
86d774dafc20: Waiting
da68f13652fc: Waiting
cf2e8433dbf2: Waiting
denied: requested access to the resource is denied
PS C:\Users\Administrateur\Desktop\Azure\dockerfile\python> docker tag demo-python-image m2informationhab.azurecr.io/demo-python-image:1
```

Push de l'image sur ECR Azure :

```
is not allowed access. Refer https://aka.ms/acr/firewall to grant access.
PS C:\Users\Administrateur\Desktop\Azure\dockerfile\python> docker push m2informationhab.azurecr.io/demo-python-image:1
The push refers to repository [m2informationhab.azurecr.io/demo-python-image]
9c119468e5e3: Pushing [=====] 2.56kB
4c8a37c5a72f: Pushing [=====] 2.048kB
28fa86444e63: Pushing [=====] 3.907MB/10.16MB
85158601c190: Pushing [=====] 4.608kB
0cdca02a73c8: Pushing [====] 3.396MB/46.45MB
0baa05faf31a: Waiting
```

Résultat de l'envoi :

Dépôts

Actualiser

Filtrage des référentiels...

Dépôts ↑↓ Cache rule

demo-python-image

demo-python-image

Référéntiel

Actualiser Gérer les artefacts supprimés Supprimer le dépôt

Vue JSON

Bases

demo-python-image

Nombre d'étiquettes: 1

Date de la dernière mise à jour: 09/03/2023 16:40 UTC+1

Nombre de manifestes: 1

Filtrage des étiquettes...

Étiquettes ↑↓	Digest ↑↓	Dernière modification
1	sha256:f52750e40f102d83f375d1...	09/03/2023 16:40 UTC+1

Résumé des commandes :

Commande Docker

Commande pour ajouter un alias avec l'host à une image

```
docker tag <nom_image_local> <host>/<nom_image>:<tag>
```

Connexion à l'host

```
docker login <host_acr> --username <username> --password <password>
```

Push de l'image sur docker hub


```
docker push <alias_avec_host>
```

```
Administrateur@LIL-JT3KN3 MINGW64 /c/Atelier/Azure/Dockerfile/Python
$ docker login benoitecr.azurecr.io --username BenoitECR --password YVtzOuwPqoBechhHuhYQj0zpSNwn/WjBCatplz2CtX+ACRD/ttuT
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
Login Succeeded

Administrateur@LIL-JT3KN3 MINGW64 /c/Atelier/Azure/Dockerfile/Python
$ docker tag image-test-benoit-azure benoitecr.azurecr.io/ecr-better-windows

Administrateur@LIL-JT3KN3 MINGW64 /c/Atelier/Azure/Dockerfile/Python
$ docker push benoitecr.azurecr.io/ecr-better-windows
Using default tag: latest
The push refers to repository [benoitecr.azurecr.io/ecr-better-windows]
1f6f6d88b27a: Pushed
e029d074b8ca: Pushed
28fa86444e63: Pushed
85158601c190: Pushed
0cdca02a73c8: Pushed
0baa05faf31a: Pushed
ac504d030134: Pushed
52ebb9a789db: Pushed
86d774dafc20: Pushed
da68f13652fc: Pushed
cf2e8433dbf2: Pushed
latest: digest: sha256:02845293bc90290b857a8ea02e10ae474a1597ed3be0df7e864e146ace28f8a7 size: 2631

Administrateur@LIL-JT3KN3 MINGW64 /c/Atelier/Azure/Dockerfile/Python
$
```

```
9 docker login benoitecr.azurecr.io --username BenoitECR --password YVtzOuwPqoBech
10 docker tag image-test-benoit-azure benoitecr.azurecr.io/ecr-better-windows
11 docker push benoitecr.azurecr.io/ecr-better-windows
```

Azure ACI :

Container Instances

Lancez des conteneurs avec l'isolation de l'hyperviseur.

[Essayez Azure gratuitement](#) [Créez un compte avec paiement à l'utilisation](#)

[Vue d'ensemble](#) [Fonctionnalités](#) [Sécurité](#) [Prix](#) [Prise en main](#) [Ressources](#)

Développez des applications rapidement sans gérer de machines virtuelles ni apprendre à vous servir de nouveaux outils. Vous avez juste besoin de votre application, dans un conteneur, qui s'exécute dans le cloud.

Création Instance de conteneur :

Créer une instance de conteneur ...

[De base](#) [Réseau](#) [Avancé](#) [Étiquettes](#) [Vérifier + créer](#)

Azure Container Instances (ACI) vous permet d'exécuter rapidement et facilement des conteneurs sur Azure sans avoir à gérer de serveurs ni à apprendre de nouveaux outils. ACI offre une facturation par seconde pour minimiser le coût d'exécution des conteneurs sur le cloud. [En savoir plus sur Azure Container Instances](#)

Détails du projet

Sélectionnez l'abonnement pour gérer les coûts et les ressources déployées. Utilisez les groupes de ressources comme les dossiers pour organiser et gérer toutes vos ressources.

Abonnement * ⓘ Abonnement Azure 1 ▼

Groupe de ressources * ⓘ m2i-formation ▼
[Créer nouveau](#)

Détails du conteneur

Nom de conteneur * ⓘ benoit-first-conteneur ✓

Région * ⓘ (US) North Central US ▼

Zones de disponibilité ⓘ None ▼

La région sélectionnée ne prend pas en charge Zones de disponibilité.

SKU Standard ▼

Standard SKU is available for all regions. Confidential SKU is only available for specific regions. [Learn more](#)

Source d'image * ⓘ

☐ Images de démarrage rapide

☒ Azure Container Registry

☐ Autre registre

☐ Autre registre

Registre * ⓘ BenoitECR ▼

Image * ⓘ ecr-better-windows ▼

Étiquette d'image * ⓘ latest ▼

Type de système d'exploitation Linux

Taille * ⓘ 1 processeur virtuel, 1.5 Gio de mémoire, 0 processeurs graphiques
[Changer la taille](#)

Configuration du réseau :

De base **Réseau** Avancé Étiquettes Vérifier + créer

Choisissez entre trois options de réseau pour votre instance de conteneur :

- « **Public** » crée une adresse IP publique pour votre instance de conteneur.
- « **Privée** » vous permet de choisir un réseau virtuel nouveau ou existant pour votre instance de conteneur. Cette option n'est pas encore disponible pour les conteneurs Windows.
- « **Aucun** » ne crée pas d'adresse IP publique ni de réseau virtuel. Vous pouvez toujours accéder à vos journaux de conteneur à l'aide de la ligne de commande.

Type de réseau ☒ Public ☐ Privé ☐ Aucun

Étiquette du nom DNS ⓘ

Réutilisation de l'étendue de l'étiquette de nom DNS ⓘ Toute réutilisation (non sécurisée) ▼

Ports ⓘ

Ports	Protocole de ports	
80	TCP	
<input type="text"/>	▼	

On peut personnalisé le comportement de notre conteneur dans les options avancées :

Créer une instance de conteneur ...

De base Réseau **Avancé** Étiquettes Vérifier + créer

Configurez des variables et des propriétés de conteneur supplémentaires.

Stratégie de redémarrage ① En cas d'échec ▼

Variables d'environnement

Marquer comme sécurisé	Clé	Valeur
Non ▼		

Remplacement de commande ① []
Exemple : ["/bin/bash", "-c", "echo hello; sleep 100000"]

Gestion des clés ①

☒ Clés gérées par Microsoft (MMK)
☐ Clés gérées par le client (CMK)

Créer une instance de conteneur ...

✓ Validation réussie

De base	
Abonnement	Abonnement Azure 1
Groupe de ressources	m2i-formation
Région	North Central US
Nom de conteneur	benoit-first-conteneur
SKU	Standard
Type d'image	Private
Serveur de connexion du registre d'images	benoitcr.azurecr.io
Image	benoitcr.azurecr.io/ecr-better-windows:latest
Nom d'utilisateur du registre d'images	BenoitECR
Type de système d'exploitation	Linux
Mémoire (Gio)	1.5
Nombre de cœurs du processeur	1
Type de processeur graphique (préversion)	None
Nombre de processeurs graphiques	0
Réseau	
Type de réseau	Public
Ports	80 (TCP)
Réutilisation de l'étendue de l'étiquette de nom DNS	Toute réutilisation (non sécurisée)

Résultat des manipulations :

▶ Démarrer
 ↶ Redémarrer
 □ Arrêter
 🗑 Supprimer
 ↻ Actualiser

^ Bases

Groupe de res...	(déplacer) : m2i-formation	SKU	: Standard
Statut	: Réussite	Type de système d'explo...	: Linux
Emplacement	: North Central US	Adresse IP (Public)	: ---
Abonnement (déplacer)	: Abonnement Azure 1	FQDN	: ---
ID d'abonnement	: c49e632f-5dd4-4c37-b1a6-578c7faa36fd	Nombre de conteneurs	: 1
Étiquettes (modifier)	: Name : benoit-first-conteneur		

Exercice : Cloner une application (https://gitlab.com/mohamed_formation_test/projet-react-app-kube-aws) dans un nouvel instance de conteneur :

Création Registre de Conteneur :


 Créer un Registre de conteneurs ...

Fonctions de base Réseau Chiffrement Étiquettes Vérifier + créer

Azure Container Registry vous permet de générer, stocker et gérer les artefacts et images conteneur dans un registre privé pour tous les types de déploiement de conteneurs. Utilisez les registres de conteneurs Azure avec vos pipelines de développement et de déploiement de conteneurs existants. Utilisez Azure Container Registry Tasks pour générer des images conteneur dans Azure à la demande, ou pour automatiser les builds déclenchées par les mises à jour du code source, les mises à jour de l'image de base d'un conteneur ou les minuteurs. [En savoir plus](#)

Détails du projet

Abonnement * Abonnement Azure 1

Groupe de ressources * m2i-formation
[Créer nouveau](#)

Détails de l'instance

Nom du Registre * benoitReactRegistry .azurecr.io

Emplacement * Japan East

Zones de disponibilité ⓘ ☐ Activé

SKU * ⓘ Premium

Configuration réseau : pour cet exercice : on reste en Accès Public

Fonctions de base Réseau Chiffrement Étiquettes Vérifier + créer

Connectivité réseau

Vous pouvez vous connecter à ce registre de manière publique via des adresses IP publique

Configuration de la connectivité

☒ Accès public (tous réseaux)

☐ Accès privé (Recommandé)

Créer un Registre de conteneurs ...

Fonctions de base Réseau Chiffrement Étiquettes Vérifier + créer

Nom ⓘ	Valeur ⓘ	Ressource
Name	benoitReactRegistry	Registre de conteneurs

Créer un Registre de conteneurs ...

✓ Validation réussie

Fonctions de base Réseau Chiffrement Étiquettes Vérifier + créer

Détails du registre

Fonctions de base

Nom du Registre	benoitReactRegistry
Abonnement	Abonnement Azure 1
Groupe de ressources	m2i-formation
Emplacement	Japan East
Zones de disponibilité	Désactivé
SKU	Premium

Sur le registry : activation des droits administrateurs :

Nom du Registre benoitReactRegistry

Serveur de connexion	benoitreactregistry.azurecr.io
Utilisateur administrateur ⓘ	<input checked="" type="checkbox"/> Activé
Nom d'utilisateur	benoitReactRegistry

Build de mon image en local :

```
21 docker build -t project_react_utopios .
```

```
26 docker tag project_react_utopios benoitreactregistry.azurecr.io/project_react_utopios:v1
27 docker push benoitreactregistry.azurecr.io/project_react_utopios:v1
```

Création de mon container :

Déploiement de plusieurs containers : nécessaire d'avoir un fichier de déploiement à indiquer (ici, en yaml)

```
Dockerfile > demo-aci > ! demo-aci-deploy.yml > {} properties > [] containers > {} 1 > {} properties > [] ports > {} 0 > # ports
1  apiVersion: 2019-12-01
2  location: japaneast
3  properties:
4    containers:
5      - name: api-todolist-1
6        properties:
7          image: benoitreactregistry.azurecr.io/project_react_utopios:v1
8          ressources:
9            requests:
10              cpu: 1
11              memoryInGb: 2
12          ports:
13            - ports: 8080
14      - name: api-todolist-2
15        properties:
16          image: benoitreactregistry.azurecr.io/project_react_utopios:v1
17          ressources:
18            requests:
19              cpu: 1
20              memoryInGb: 2
21          ports:
22            - ports: 8090
```

YAML deployment

Create a `secure-env.yaml` file with the following snippet.

```
YAML Copy
apiVersion: 2019-12-01
location: eastus
name: securetest
properties:
  containers:
    - name: mycontainer
      properties:
        environmentVariables:
          - name: 'NOTSECRET'
            value: 'my-exposed-value'
          - name: 'SECRET'
            secureValue: 'my-secret-value'
        image: mcr.microsoft.com/oss/nginx/nginx:1.15.3-alpine
        ports: []
        resources:
          requests:
            cpu: 1.0
            memoryInGB: 1.5
        osType: Linux
        restartPolicy: Always
      tags: null
      type: Microsoft.ContainerInstance/containerGroups
```

Envoie des images vote/result/worker dans le dépôt

Script possible pour automatiser cette étape :

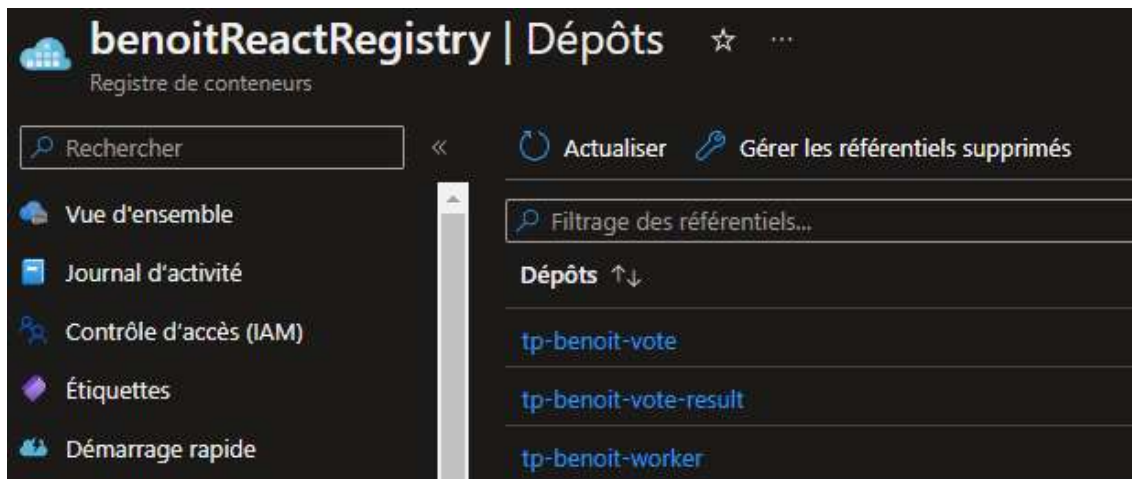
```
# Automatisation du build, tag et push
# Coller le fichier dans le dossier de destination
# Changer la valeur de la variable
# Exécuter le script en ligne de commande dans le dossier

$ImageName = "arnaud"

docker build -t $ImageName .
docker tag worker-arnaud m2informationarnaud.azurecr.io/$ImageName
docker push m2informationarnaud.azurecr.io/$ImageName
```

```
# Automatisation du build, tag et push
# Coller le fichier dans le dossier de destination
# Changer la valeur de la variable
# Exécuter le script en ligne de commande dans le dossier
```

```
$ImageName = "ma_variable"
docker build -t $ImageName .
docker tag worker-arnaud m2informationarnaud.azurecr.io/$ImageName
docker push m2informationarnaud.azurecr.io/$ImageName
```



Vérification du code .yaml qui sera utilisé pour push nos conteneurs dans notre instance :

```
apiVersion: 2019-12-01
location: japaneast
type: Microsoft.ContainerInstance/ContainerGroups
properties:
  containers:
    - name: tp-benoit-vote-result
      properties:
        image: benoitreactregistry.azurecr.io/tp-benoit-vote-result:v1
        resources:
          requests:
            cpu: 0.5
            memoryInGb: 0.5
        ports:
          - port: 4000
```

```
- name: tp-benoit-vote
  properties:
    image: benoitreactregistry.azurecr.io/tp-benoit-vote:v1
    resources:
      requests:
        cpu: 0.5
```



```

      memoryInGb: 0.5
    ports:
      - port: 80

- name: tp-benoit-worker
  properties:
    image: benoitreactregistry.azurecr.io/tp-benoit-worker:v1
    resources:
      requests:
        cpu: 0.5
        memoryInGb: 0.5
    ports:
      - port: 8081

```

Ne pas oublier les variables d'environnement pour postgres : ici user et password, sinon le conteneur postgres ne démarrera pas

```

- name: tp-benoit-postgres
  properties:
    image: postgres
    environmentVariables:
      - name: 'POSTGRES_USER'
        value: 'postgres'
      - name: 'POSTGRES_PASSWORD'
        value: 'postgres'
    resources:
      requests:
        cpu: 2
        memoryInGb: 2
    ports:
      - port: 5432

```

Une fois les images push sur le registry -> exécution du code .yaml pour déployer les conteneurs :

```

90 az container create --name benoit-apply-vote --resource-group m2i-formation --file web_apply
91 az container create --name benoit-apply-vote --resource-group m2i-formation --file web_apply

```

Une fois l'exécution effectué -> vérification de l'instances et de l'état des conteneurs

▶ Démarrer
↺ Redémarrer
□ Arrêter
🗑 Supprimer
🔄 Actualiser

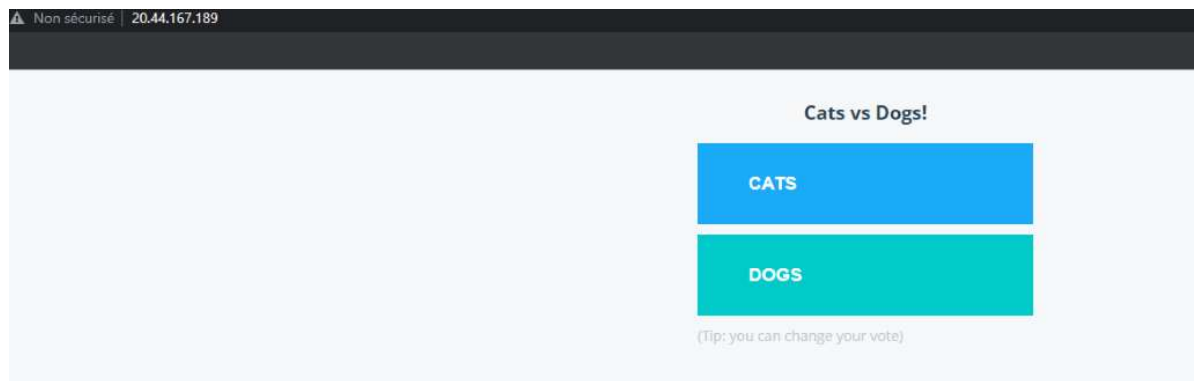
^ Bases

Groupe de res...	(déplacer) : m2i-formation	SKU	: Sta
Statut	: En cours d'exécution	Type de système d'explo...	: Lin
Emplacement	: Japan East	Adresse IP (Public)	: 20.
Abonnement (déplacer)	: Abonnement Azure 1	FQDN	: ---
ID d'abonnement	: c49e632f-5dd4-4c37-b1a6-578c7faa36fd	Nombre de conteneurs	: 5
Étiquettes (modifier)	: Cliquez ici pour ajouter des étiquettes		

🔄 Actualiser

5 conteneurs et 0 conteneurs init		
Nom	Image	État
tp-benoît-vote-result	benoitreactregistry.azurecr.io/tp-benoît-...	Running
tp-benoît-vote	benoitreactregistry.azurecr.io/tp-benoît-...	Running
tp-benoît-worker	benoitreactregistry.azurecr.io/tp-benoît-...	Running
tp-benoît-postgres	postgres	Running
tp-benoît-redis	redis	Running

Test d'accès à l'adresse publique de l'instance :



Code utilisé pour deploy les applications :

```

apiVersion: 2019-12-01
location: japaneast
type: Microsoft.ContainerInstance/ContainerGroups
properties:
  containers:
    - name: tp-benoit-vote-result
      properties:
        image: benoitreactregistry.azurecr.io/tp-benoit-vote-result:v1
        resources:
          requests:
            cpu: 0.5
            memoryInGb: 0.5
        ports:
          - port: 4000
    - name: tp-benoit-vote
      properties:
        image: benoitreactregistry.azurecr.io/tp-benoit-vote:v1
        resources:
          requests:
            cpu: 0.5
            memoryInGb: 0.5
        ports:
          - port: 80
    - name: tp-benoit-worker
      properties:
        image: benoitreactregistry.azurecr.io/tp-benoit-worker:v1
        resources:
          requests:
            cpu: 0.5
            memoryInGb: 0.5
        ports:
          - port: 8081
    - name: tp-benoit-postgres
      properties:
        image: postgres
        environmentVariables:
          - name: 'POSTGRES_USER'
            value: 'postgres'
          - name: 'POSTGRES_PASSWORD'
            value: 'postgres'
        resources:
          requests:

```

```
      cpu: 2
      memoryInGb: 2
    ports:
      - port: 5432
- name: tp-benoit-redis
  properties:
    image: redis
    resources:
      requests:
        cpu: 0.5
        memoryInGb: 0.5
    ports:
      - port: 6379
osType: linux
ipAddress:
  type: public
  ports:
    - protocol: tcp
      port: 80
```