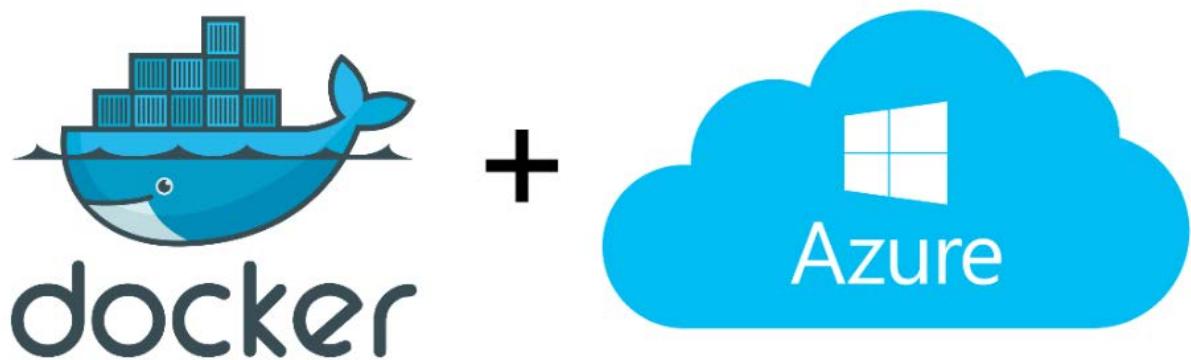


ATELIER 3



Déployer dans Azure
en mode « conteneur »

Pour réaliser cet atelier, vous devez disposer d'une machine équipée d'une connexion réseau et un compte Microsoft AZURE.

Si vous n'avez pas de compte Microsoft Azure, procurez-vous votre identifiant gratuit auprès des speakers.

La procédure pour enregistrer votre identifiant
<http://www.microsoftazurepass.com/howto>

ACTE 1

Provisionner une nouvelle VM dans AZURE

Dans ce premier acte, nous allons provisionner une machine virtuelle Windows Server 2016 avec les Windows Containers dans AZURE.

La première action est de se rendre sur le portail sur le portail AZURE.

<http://portal.azure.com/>

Vous serez redirigé sur une interface pour vous identifier si vous n'étiez pas précédemment connecté.

Si vous disposez d'un compte gratuit AZURE, c'est le moment de saisir les informations relatives à ce compte.

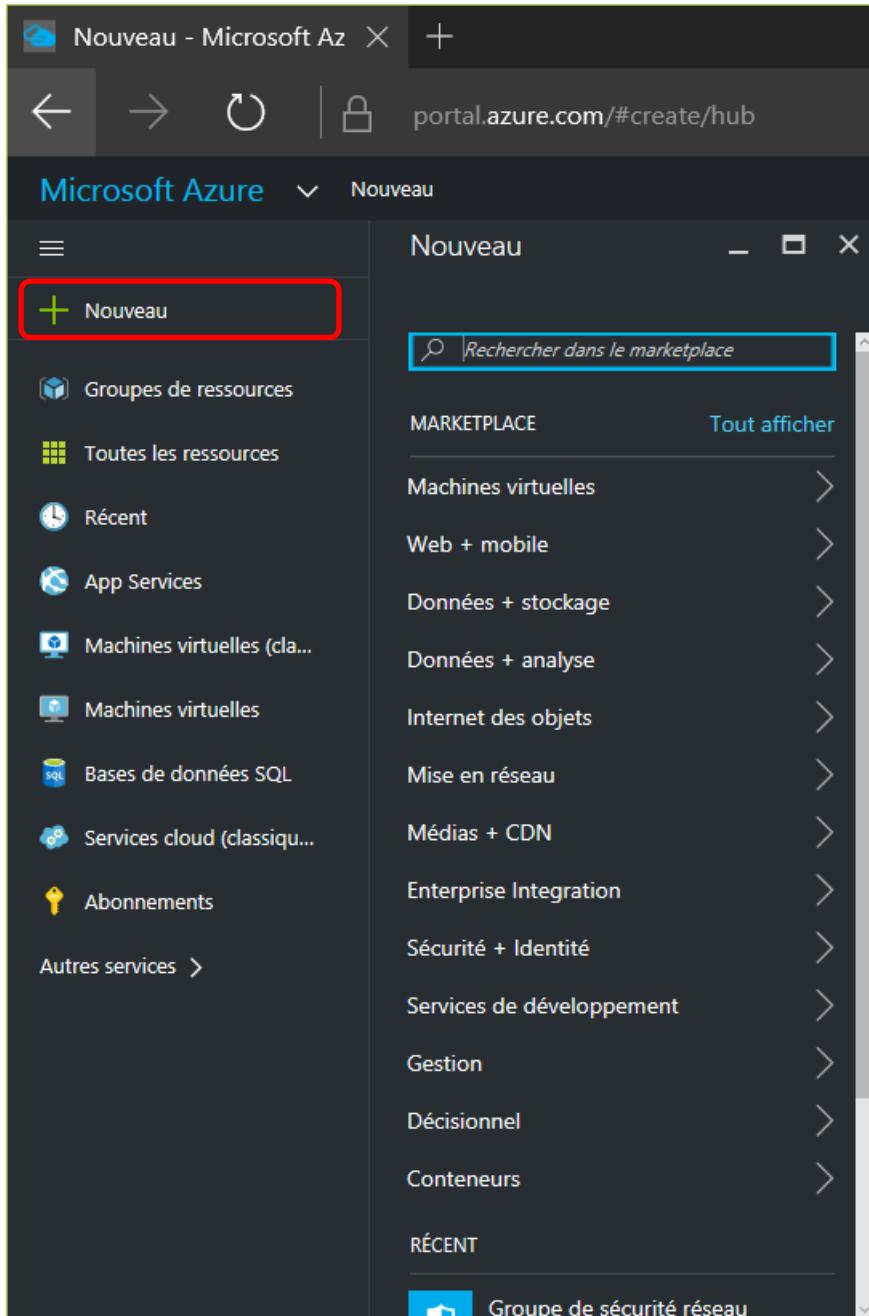


The screenshot shows a Microsoft Azure login page within a web browser. The address bar indicates the URL is login.live.com/login.srf?wa=wsignin1.0. The main content area features a blue background with the text "Microsoft Azure" at the top left and "Se connecter à Azure" in the center. To the right, there's a "Connexion" section with fields for "Adresse e-mail ou téléphone" and "Mot de passe", both currently empty. A checked checkbox labeled "Maintenir la connexion" is present. Below these fields is a large blue "Se connecter" button. Further down, links for "Votre compte n'est pas accessible ?" and "Se connecter avec un code à usage unique" are visible. At the bottom right of the page, the Microsoft logo and copyright information ("Conditions d'utilisation", "Confidentialité et cookies", "©2016 Microsoft") are displayed.

Vous avez le choix de vous connecter ou bien de créer un compte.



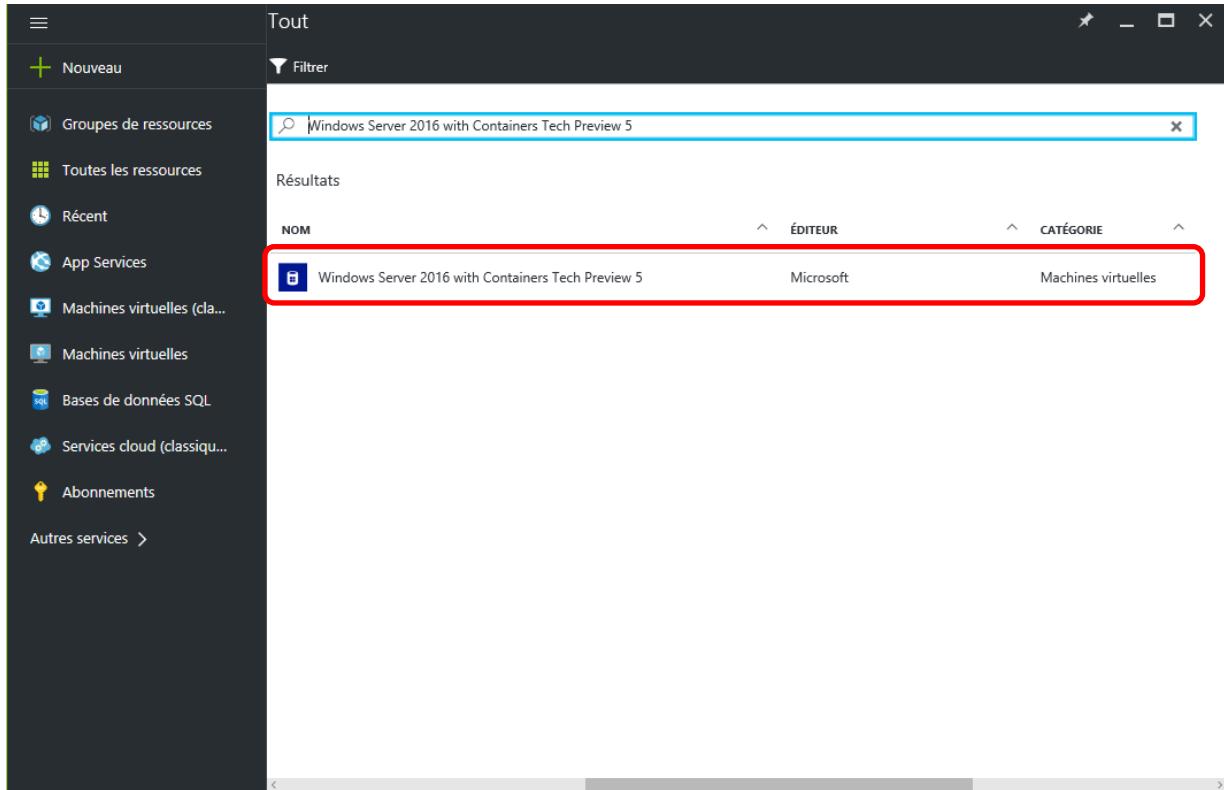
Une fois connecté, vous cliquez sur « Nouveau »



Et dans la partie recherche MARKETPLACE, saisissez le nom suivant :

Windows Server 2016 with Containers Tech Preview 5

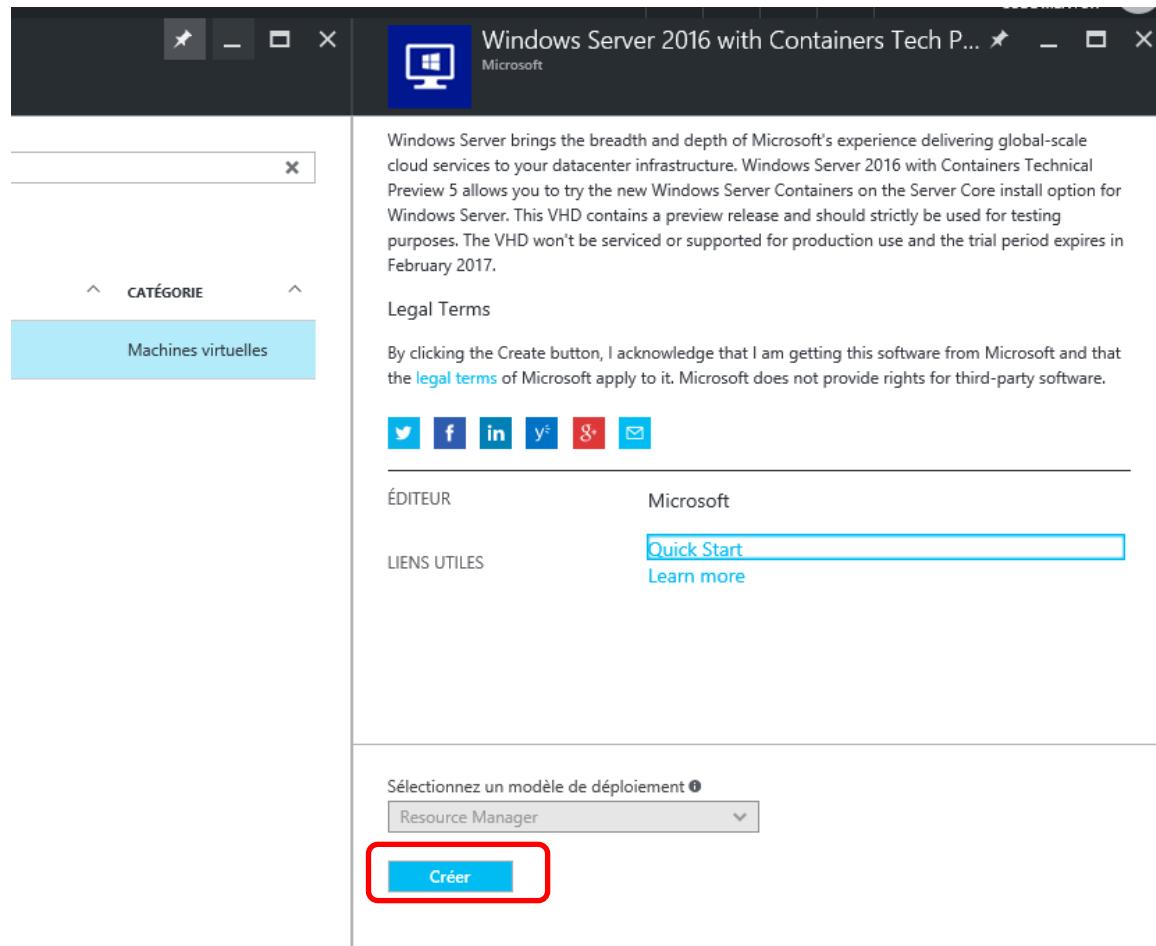
Vous devriez trouver immédiatement la bonne VM.

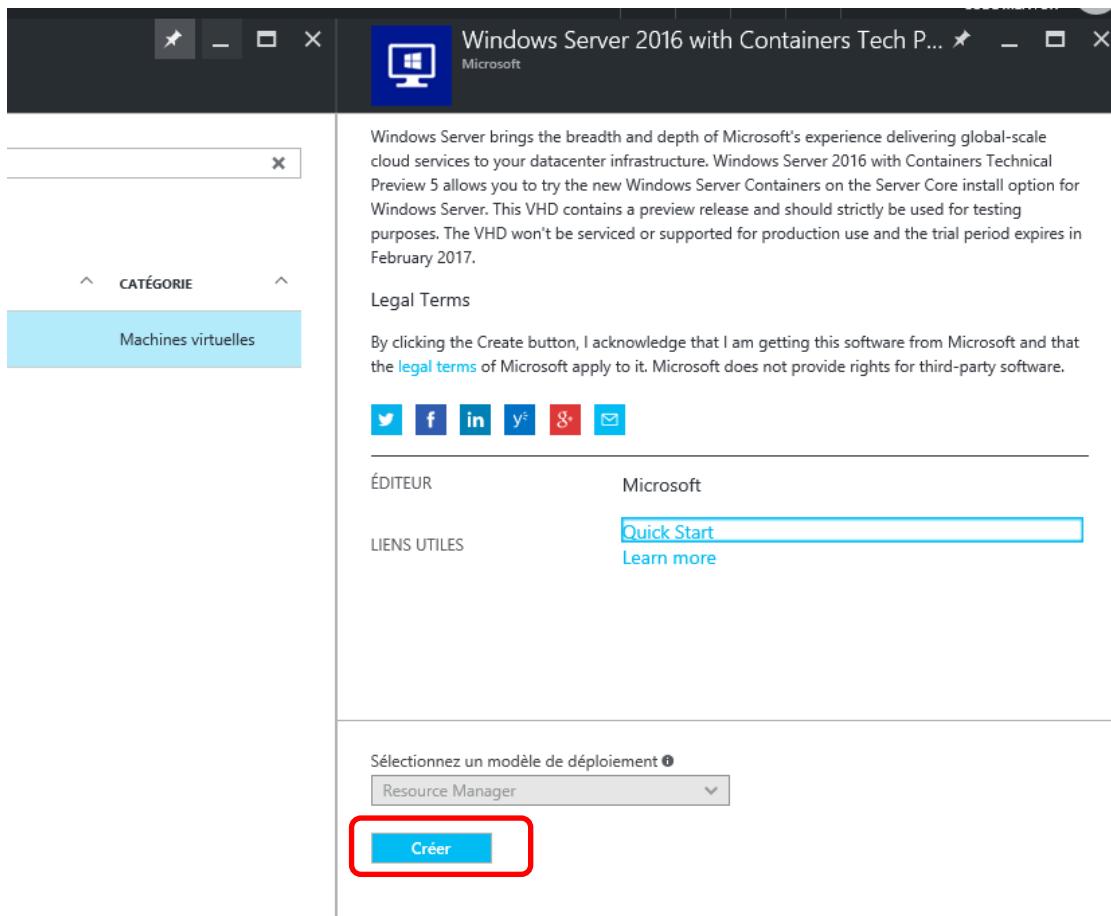


The screenshot shows the Azure portal interface. On the left, there's a sidebar with various service icons like Groups de ressources, Toutes les ressources, Récent, App Services, Machines virtuelles (classiques), Machines virtuelles, Bases de données SQL, Services cloud (classiques), and Abonnements. Below that is an 'Autres services >' link. The main area is titled 'Tout' and has a 'Filtrer' button. A search bar at the top contains the text 'Windows Server 2016 with Containers Tech Preview 5'. Below the search bar, the results are listed under 'Résultats'. There are three columns: 'NOM', 'ÉDITEUR', and 'CATÉGORIE'. The first result, 'Windows Server 2016 with Containers Tech Preview 5', is highlighted with a red box. It shows the icon of a server, the name 'Windows Server 2016 with Containers Tech Preview 5', 'Microsoft' as the editor, and 'Machines virtuelles' as the category. At the bottom of the results area, there are navigation arrows for more results.



L'assistant d'installation devrait vous ouvrir cette page. Vous cliquez sur la ligne, afin de sélectionner cette VM.





Vous n'avez plus qu'à cliquer sur “Créer” pour poursuivre la préparation de notre VM.

La création de la VM se déroule rapidement en 4 étapes :

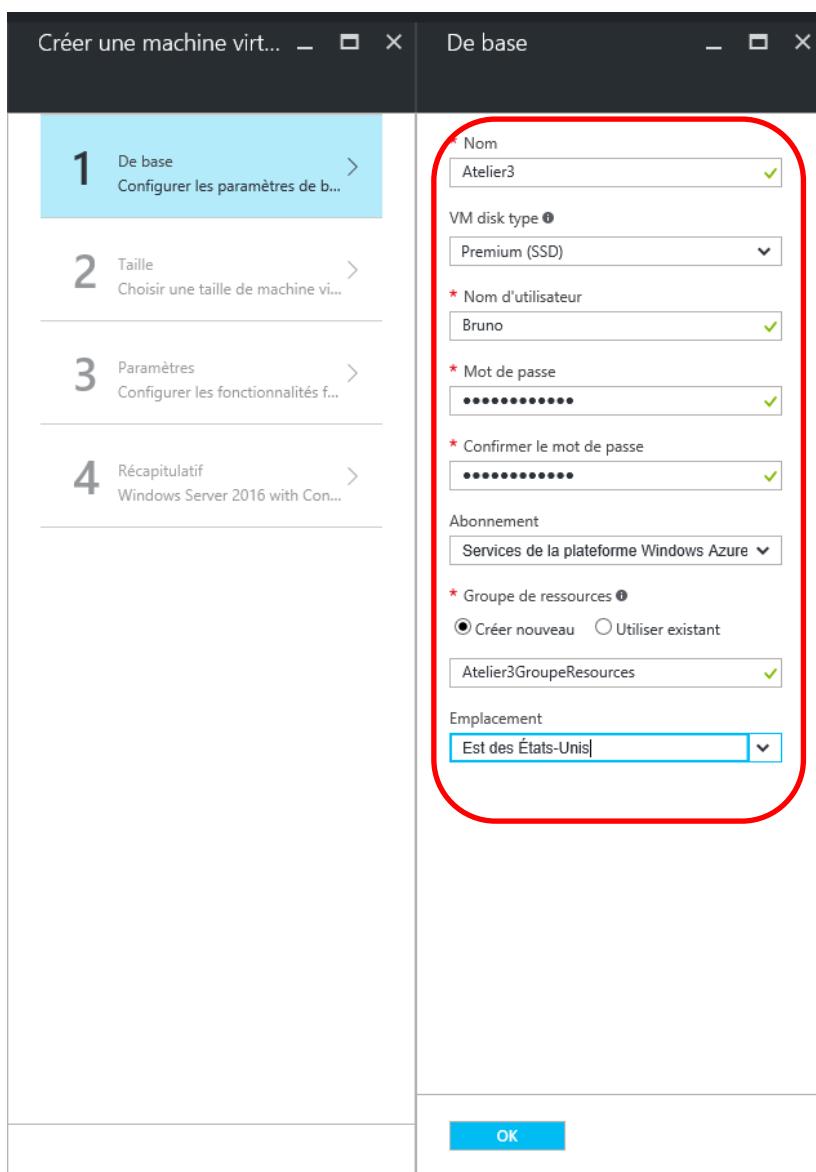
- 1 De base
- 2 Taille
- 3 Paramètres
- 4 Récapitulatif

1 De base

Dans l'écran suivant, vous allez devoir vous signer afin de préparer le compte administrateur de la VM. Vous devez aussi définir l'abonnement associé à la VM. Si vous disposez de plusieurs abonnements, vous pouvez choisir celui qui sera le plus adapté à votre usage.

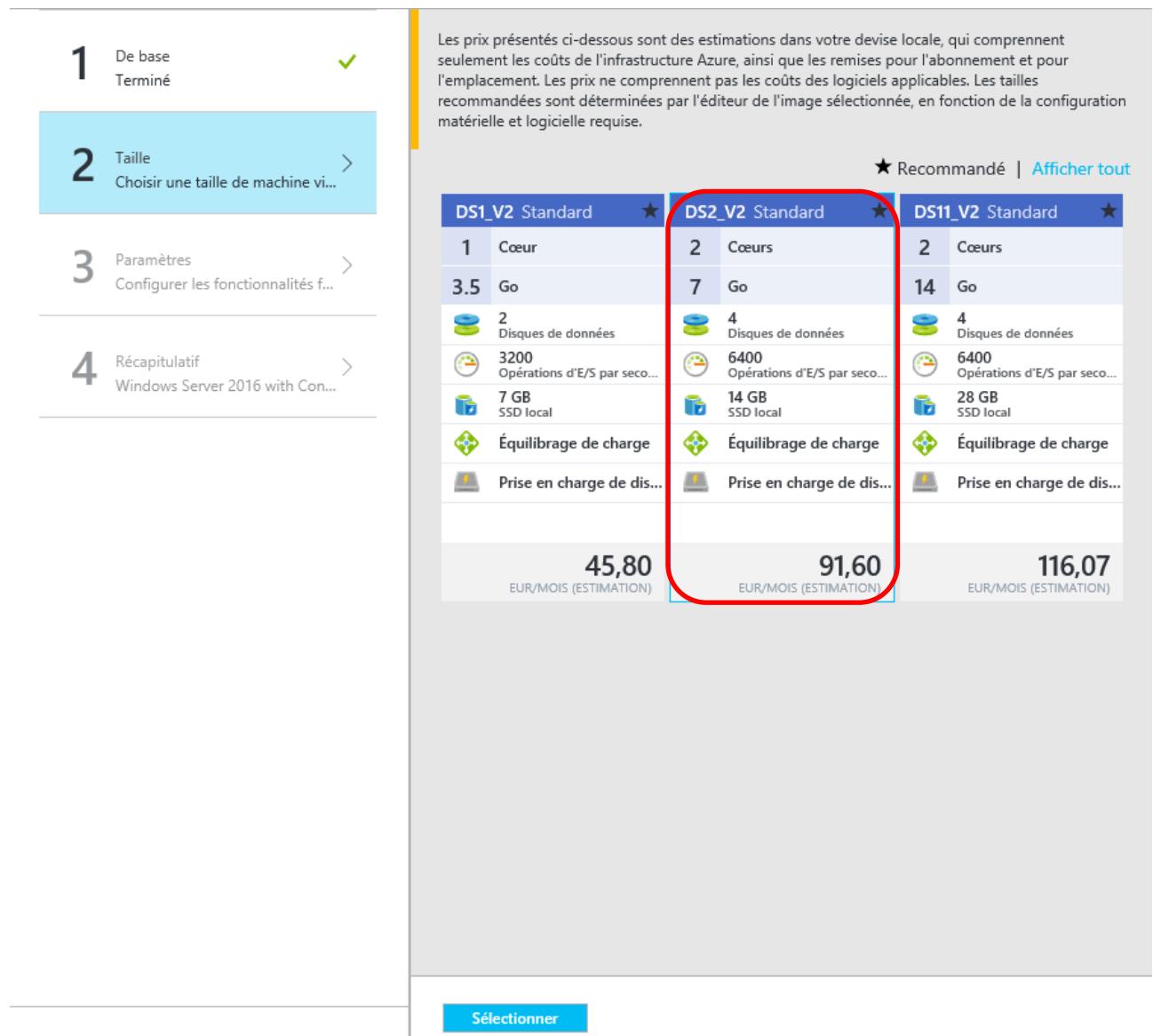
Vous devez définir le nom du groupe de ressource. Ce groupe contient tous les éléments d'infrastructure que votre VM utilisera.

Et enfin, vous définissez, l'emplacement physique où votre VM s'exécutera.



2 Taille

Puis vous passerez au choix de la taille de machine que vous comptez utiliser.



	DS1_V2 Standard	DS2_V2 Standard	DS11_V2 Standard
1	1 Cœur	2 Coeurs	2 Coeurs
	3,5 Go	7 Go	14 Go
	2 Disques de données	4 Disques de données	4 Disques de données
	3200 Opérations d'E/S par seco...	6400 Opérations d'E/S par seco...	6400 Opérations d'E/S par seco...
	7 GB SSD local	14 GB SSD local	28 GB SSD local
2	Équilibrage de charge	Équilibrage de charge	Équilibrage de charge
	Prise en charge de dis...	Prise en charge de dis...	Prise en charge de dis...
3			
4			

1 De base Terminé ✓

2 Taille Choisir une taille de machine vi... >

3 Paramètres Configurer les fonctionnalités f... >

4 Récapitulatif Windows Server 2016 with Con... >

Sélectionner

Les prix présentés ci-dessous sont des estimations dans votre devise locale, qui comprennent seulement les coûts de l'infrastructure Azure, ainsi que les remises pour l'abonnement et pour l'emplacement. Les prix ne comprennent pas les coûts des logiciels applicables. Les tailles recommandées sont déterminées par l'éditeur de l'image sélectionnée, en fonction de la configuration matérielle et logicielle requise.

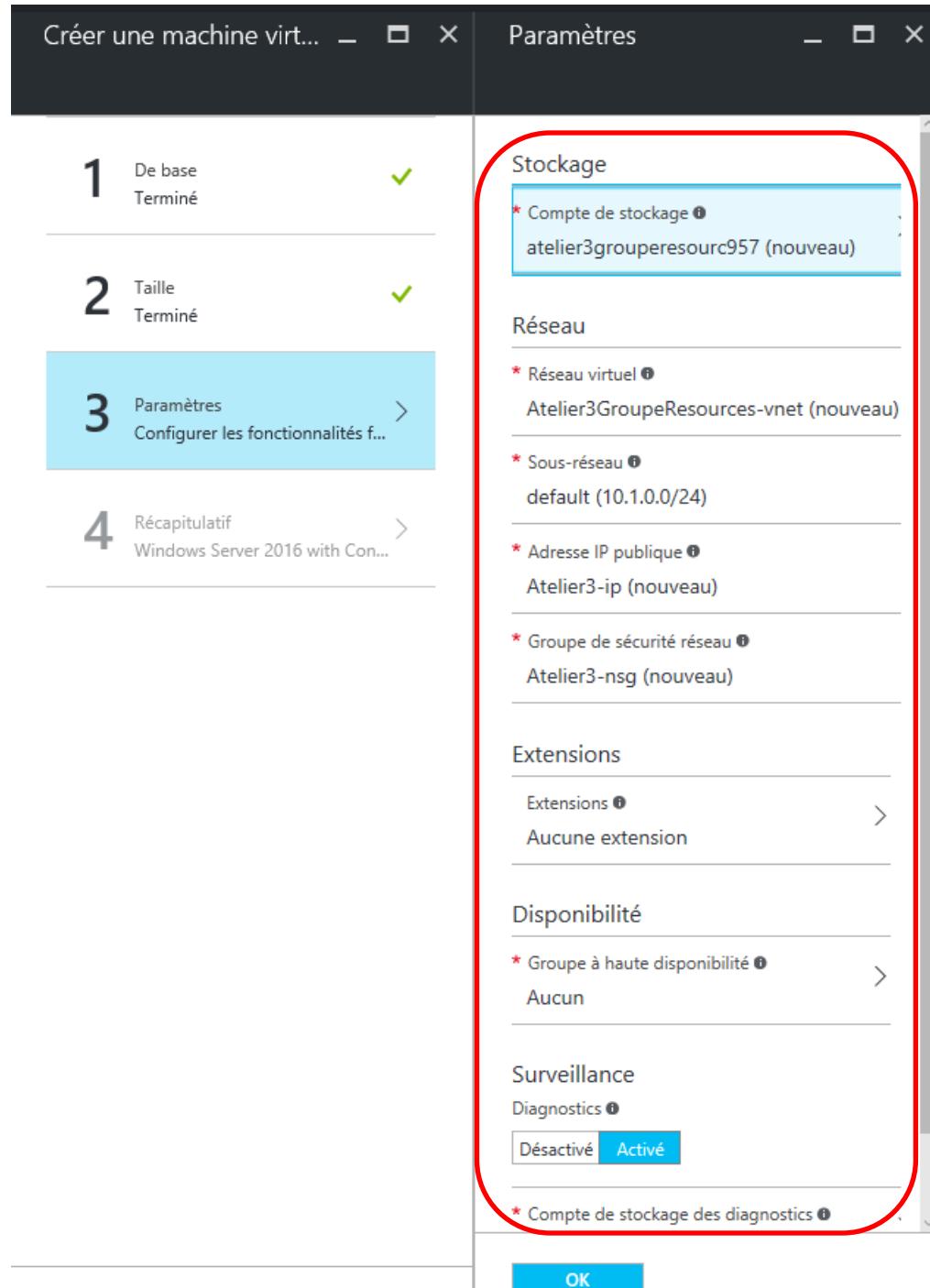
★ Recommandé | Afficher tout

Ici, on a choisi la sélection « DS2_V2 Standard » (2 coeurs, 7 Go RAM ...), bref une configuration raisonnable😊.



3 Paramètres

L'assistant, nous propose un nouvel écran contenant des informations importantes pour notre atelier.

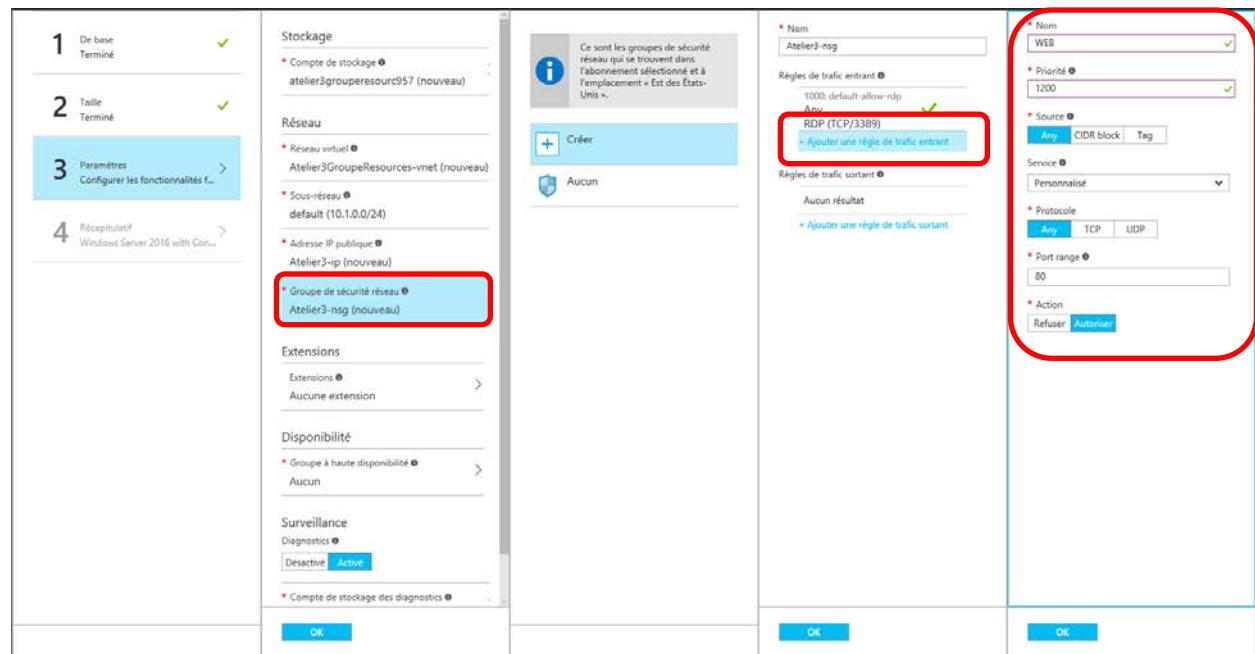


Permettre l'accès à un site Web contenu dans notre VM

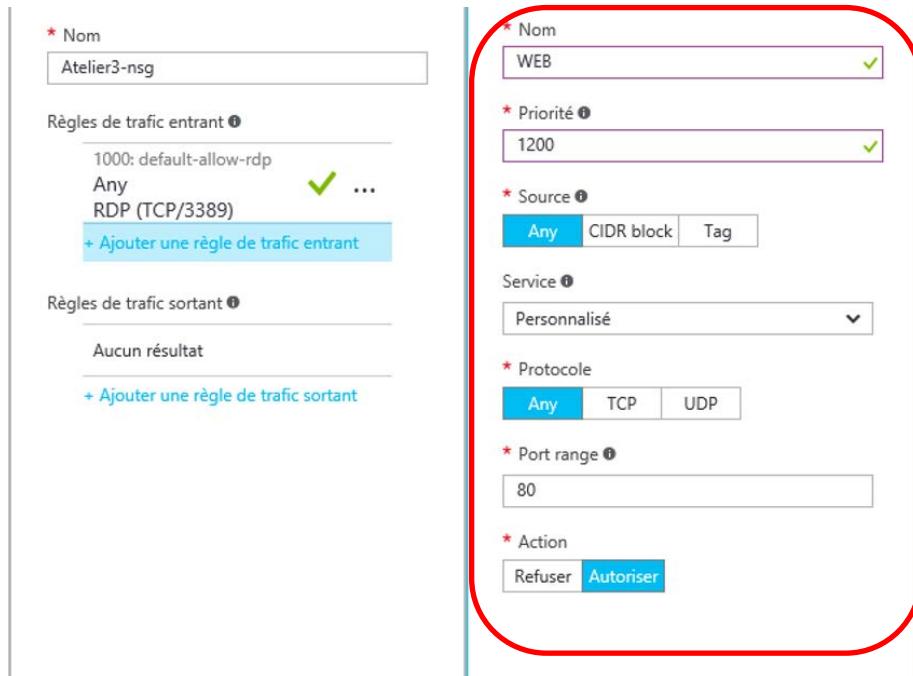
Par défaut, une nouvelle VM dans Azure, ne peut pas recevoir de connexion entrante exceptée pour les entrées RDP (service de connexions Bureau distant) afin de se connecter à la machine.

Nous allons donc légèrement modifier la configuration par défaut, afin d'autoriser les connexions Web.

Cliquez sur l'item **Groupe de sécurité réseau**, afin de déplier un menu. Vous cliquerez sur **Ajouter une règle de trafic entrant** afin d'ajouter une nouvelle règle.



Saisissez les champs suivants : Nom : WEB, Priorité 1200, Source : Any, Service : Personnalisé, Protocole : Any, Action : Autoriser



* Nom
Atelier3-nsg

Règles de trafic entrant ⓘ
1000: default-allow-rdp
Any RDP (TCP/3389) ✓ ...
+ Ajouter une règle de trafic entrant

Règles de trafic sortant ⓘ
Aucun résultat
+ Ajouter une règle de trafic sortant

* Nom
WEB

* Priorité ⓘ
1200

* Source ⓘ
Any CIDR block Tag

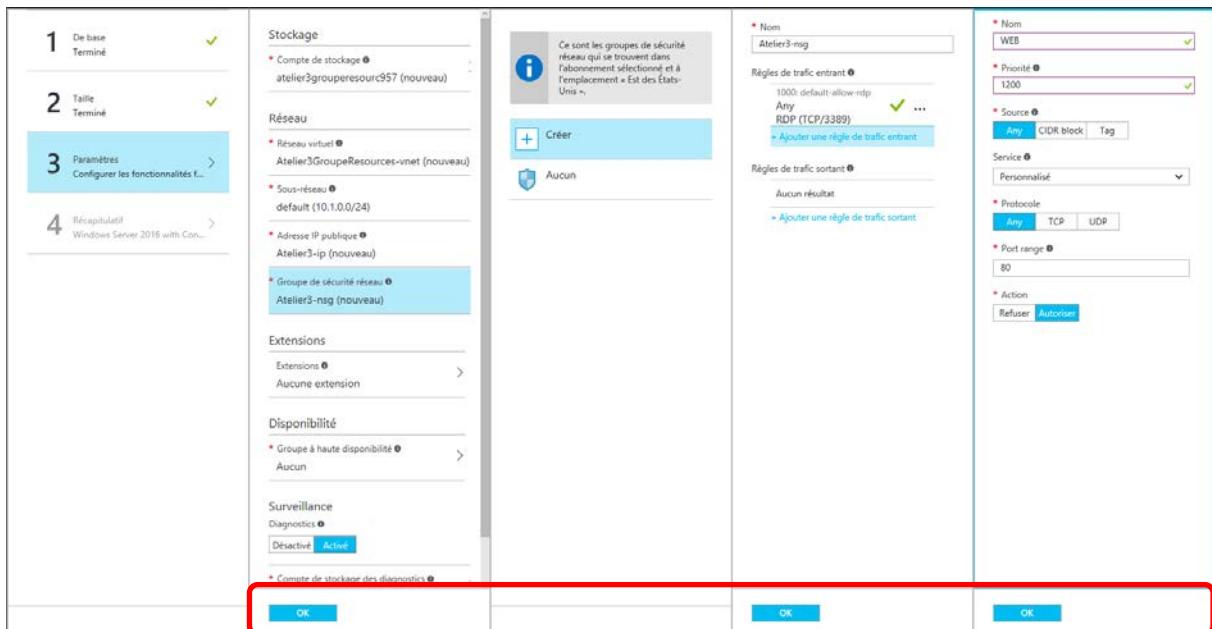
* Service ⓘ
Personnalisé

* Protocole
Any TCP UDP

* Port range ⓘ
80

* Action
Refuser Autoriser

Pour terminer, vous cliquez de la droite vers la gauche sur tous les boutons « OK » afin de terminer la définition de notre VM.

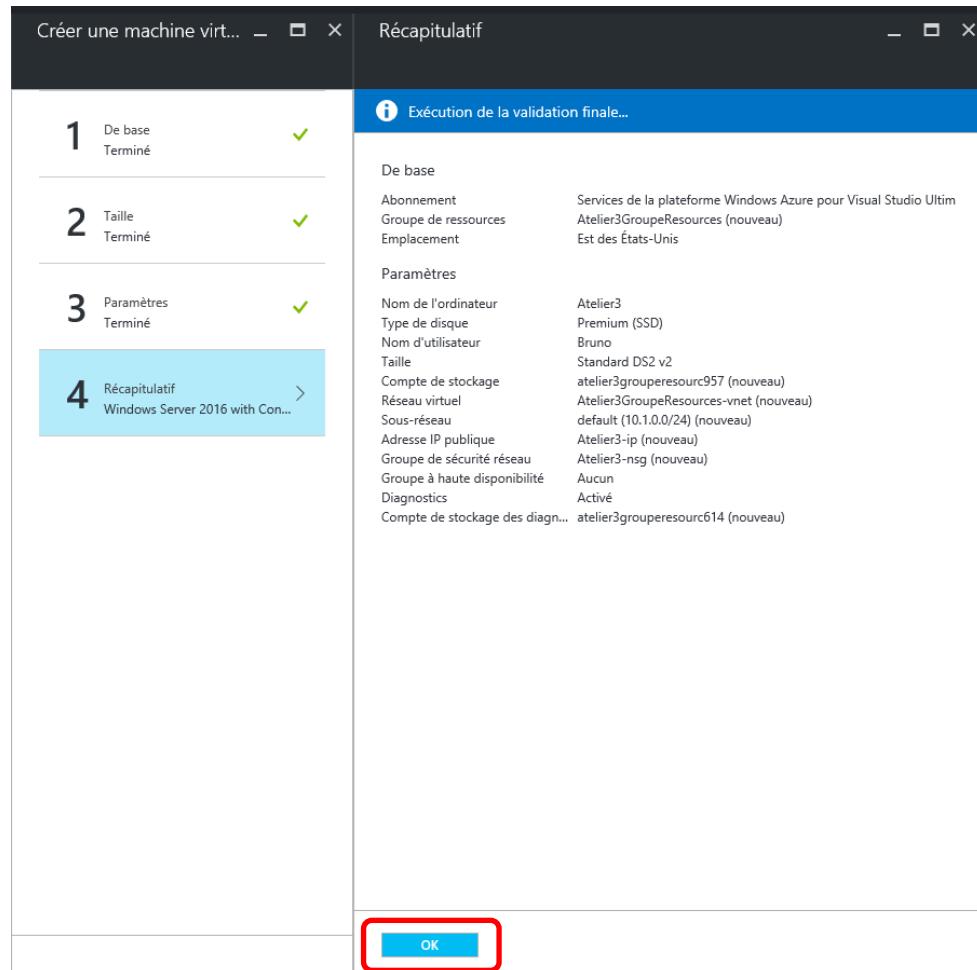


1 De base Terminé	✓	Stockage * Compte de stockage ⓘ atelier3grouperesourcsg957 (nouveau)	Ce sont les groupes de sécurité réseau qui se trouvent dans l'abonnement sélectionné et à l'emplacement « Est des États-Unis ».	* Nom Atelier3-nsg
2 Taille Terminé	✓	Réseau * Réseau virtuel ⓘ Atelier3GroupeResources-vnet (nouveau)	+ Créer	* Priorité ⓘ 1200
3 Paramètres Configurer les fonctionnalités t...		* Sous-réseau ⓘ default (10.1.0.0/24)	Aucun	* Source ⓘ Any CIDR block Tag
4 Récapitulatif Windows Server 2016 with Con...		* Adresse IP publique ⓘ Atelier3-ip (nouveau)		* Service ⓘ Personnalisé
		* Groupe de sécurité réseau ⓘ Atelier3-nsg (nouveau)		* Protocole Any TCP UDP
		Extensions * Extensions ⓘ Aucune extension		* Port range ⓘ 80
		Disponibilité * Groupe à haute disponibilité ⓘ Aucun		* Action Refuser Autoriser
		Surveillance Diagnostics ⓘ Désactivé Activer		
		* Contente de stockage des diagnostics ⓘ		
		OK	OK	OK



4 Récapitulatif

Le dernier écran est une forme de synthèse de la définition de notre VM afin de lancer sa création.



Pour valider, cliquez « Ok » et le déploiement de notre VM commence immédiatement.



Se libérer par ses compétences

<http://www.42skillz.io>

The screenshot shows the 42skillz platform interface. At the top left, there's a "MES RESSOURCES" section with a world map showing green checkmarks at various locations. Below it is a "Marketplace" icon (blue shopping bag with a white cloud) and a "Aide + support" icon (blue person with a speech bubble). On the right, a blue box displays a grid icon and the text "Déploiement de Windows Server 2016 with ...".

Après environ **15 minutes** notre VM sera prête à être utilisée.

Le binaire c'est pour les machines. Le Code, c'est pour les gens. Nous, on s'occupe des gens.

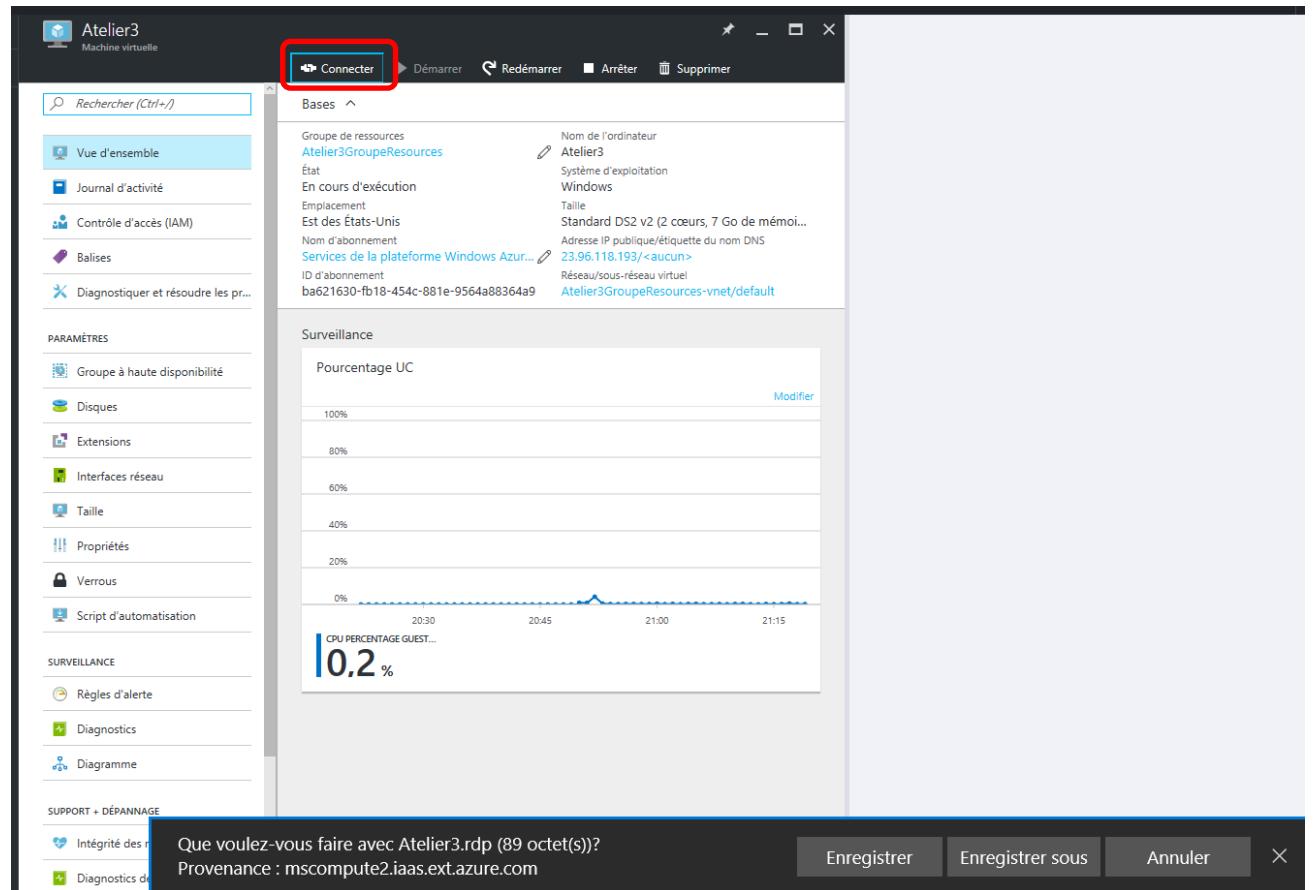


Acte 2

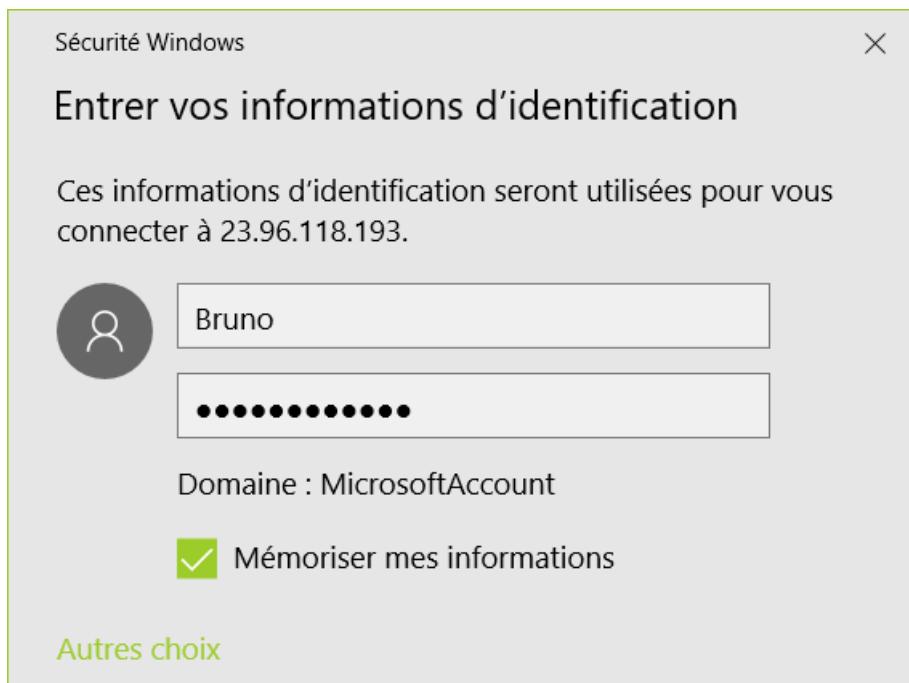
Découverte de docker via les Windows Containers

Connexion à la VM

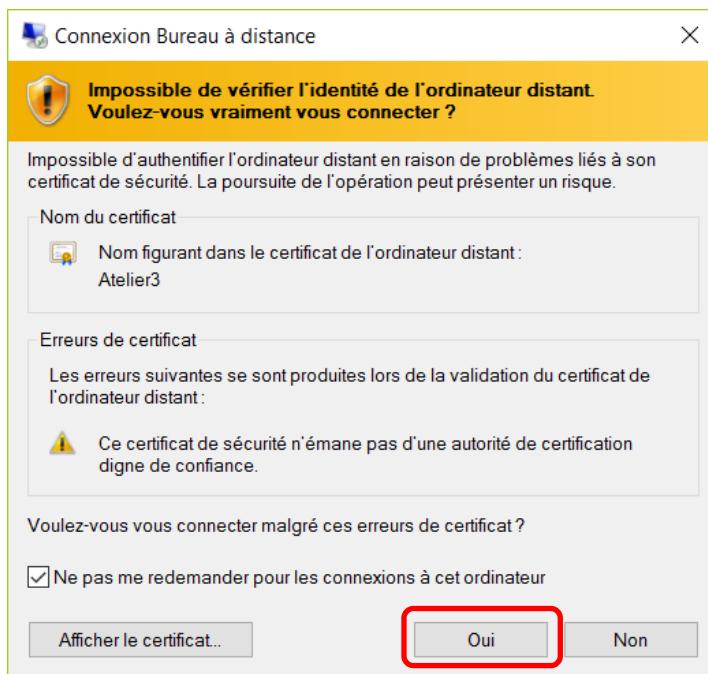
Pour vous connecter, il vous suffit de vous rendre sur le portail, sur la vue d'ensemble de votre machine virtuelle, puis de cliquer sur « Connecter »



Un fichier de configuration pour se connecter avec le bureau à distante sur la machine virtuelle, est téléchargé. Il vous suffit de l'ouvrir pour vous signer et vous connecter.



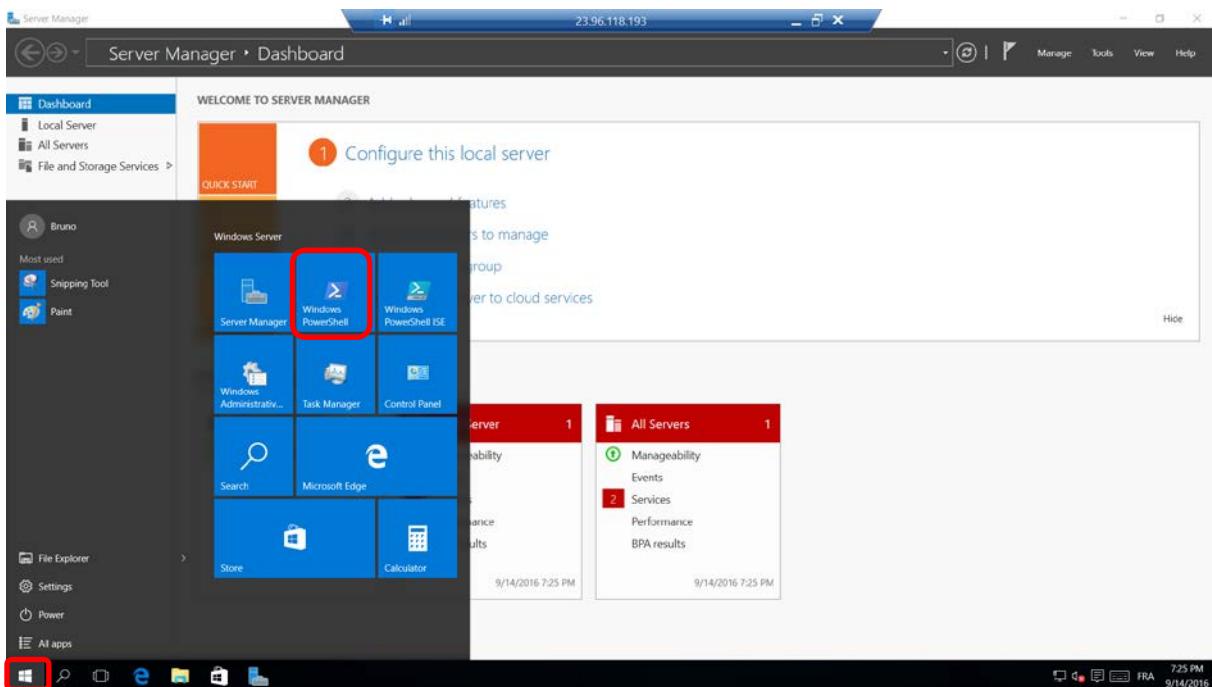
Même si le certificat de votre machine n'est pas vraiment vérifiable, je vous invite à accepter pour poursuivre votre connexion



Une fois connecté, vous devrez autoriser cette machine à être visible par d'autres.



Puis cliquer sur le menu Windows, pour ensuite lancer une session PowerShell.

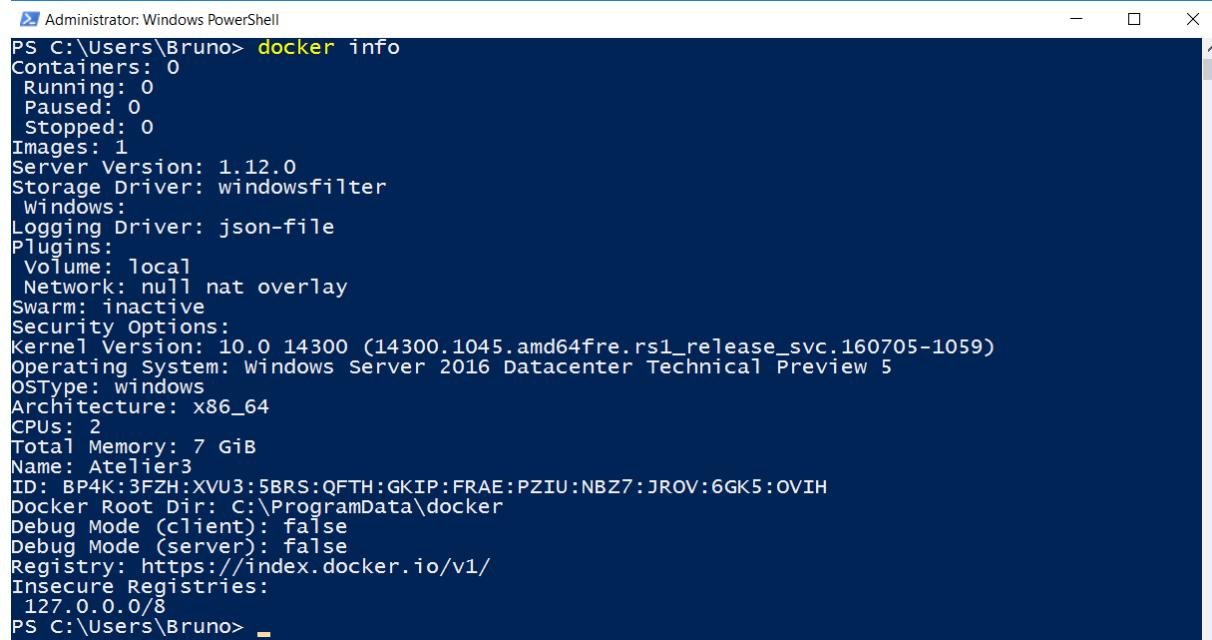


Jouer avec les commandes docker de bases

Arrivé à ce stade, vous êtes fin prêt pour commencer votre apprentissage de docker.

La première commande est sans aucun doute :

docker info

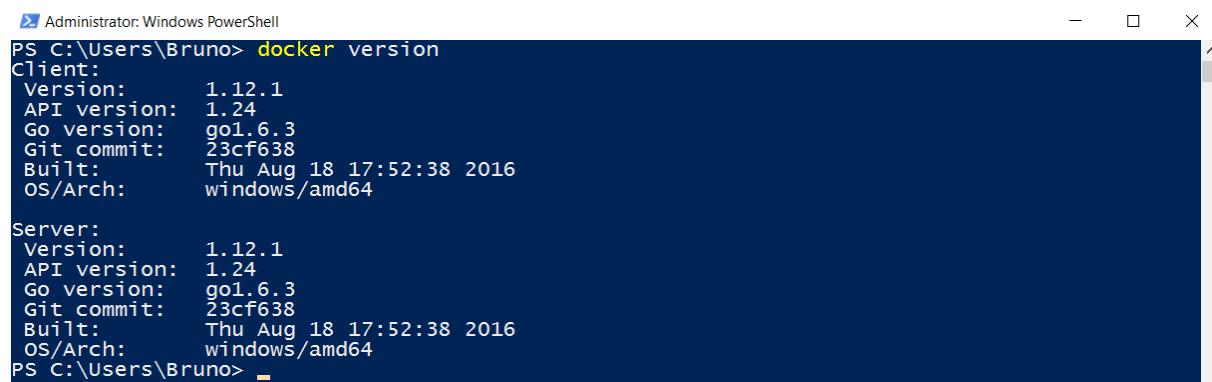


```
PS C:\Users\Bruno> docker info
Containers: 0
Running: 0
Paused: 0
Stopped: 0
Images: 1
Server Version: 1.12.0
Storage Driver: windowsfilter
Windows:
Logging Driver: json-file
Plugins:
Volume: local
Network: null nat overlay
Swarm: inactive
Security Options:
Kernel Version: 10.0 14300 (14300.1045.amd64fre.rs1_release_svc.160705-1059)
Operating System: Windows Server 2016 Datacenter Technical Preview 5
OSType: windows
Architecture: x86_64
CPUs: 2
Total Memory: 7 GiB
Name: Atelier3
ID: BP4K:3FZH:XVU3:5BRS:QFTH:GKIP:FRAE:PZIU:NBZ7:JROV:6GK5:OVIH
Docker Root Dir: C:\ProgramData\docker
Debug Mode (client): false
Debug Mode (server): false
Registry: https://index.docker.io/v1/
Insecure Registries:
127.0.0.0/8
PS C:\Users\Bruno> -
```

Elle vous permet de prendre connaissance avec la configuration système de la plateforme docker installée.

La seconde commande permet d'obtenir les détails du client et du serveur installés :

docker version



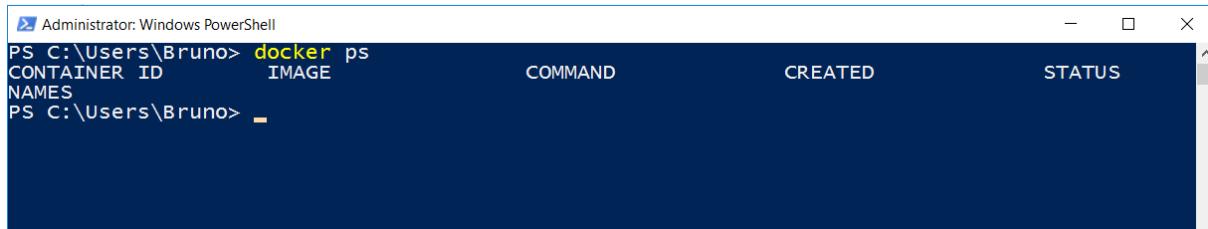
```
PS C:\Users\Bruno> docker version
Client:
Version: 1.12.1
API version: 1.24
Go version: go1.6.3
Git commit: 23cf638
Built: Thu Aug 18 17:52:38 2016
OS/Arch: windows/amd64

Server:
Version: 1.12.1
API version: 1.24
Go version: go1.6.3
Git commit: 23cf638
Built: Thu Aug 18 17:52:38 2016
OS/Arch: windows/amd64
PS C:\Users\Bruno> -
```



La commande suivante est :

docker ps

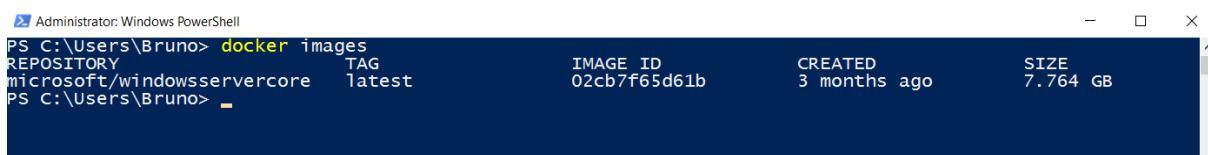


```
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker ps
CONTAINER ID        IMAGE               COMMAND       CREATED          STATUS
NAMES
PS C:\Users\Bruno>
```

À l'instar de la commande ps qui affiche les processus en cours d'exécutions sous UNIX, cette commande affiche les conteneurs en cours d'exécution. Alors, rien d'étonnant de ne rien observer puisque nous venons d'installer la machine.

Pour engendrer un conteneur, il nous faut au moins une image. Et pour connaître les images disponibles dans notre entrepôt local. Nous lançons la commande suivante :

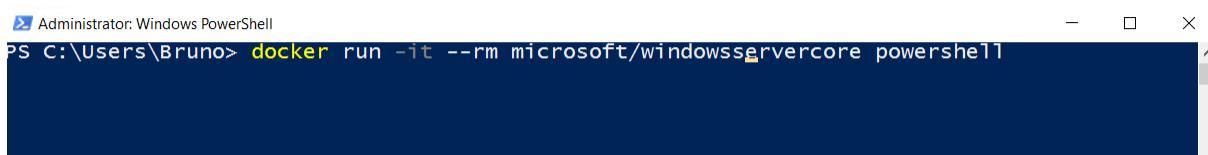
docker images



```
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker images
REPOSITORY          TAG      IMAGE ID      CREATED          SIZE
microsoft/windowsservercore   latest   02cb7f65d61b   3 months ago   7.764 GB
PS C:\Users\Bruno>
```

Par défaut, notre machine arrive avec une image de base, microsoft/windowsservercore. Pour faire connaissance avec les conteneurs voici une première commande pour lancer votre premier conteneur :

docker run -it --rm microsoft/windowsservercore powershell

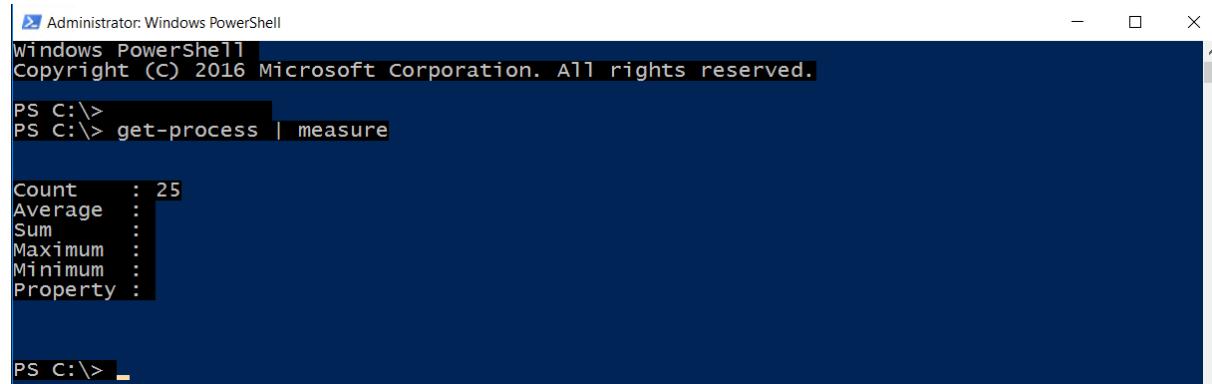


```
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker run -it --rm microsoft/windowsservercore powershell
```

Une fois dans le conteneur, lancer la commande :

get-process | measure





```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\>
PS C:\> get-process | measure

Count      : 25
Average    :
Sum        :
Maximum   :
Minimum   :
Property  :

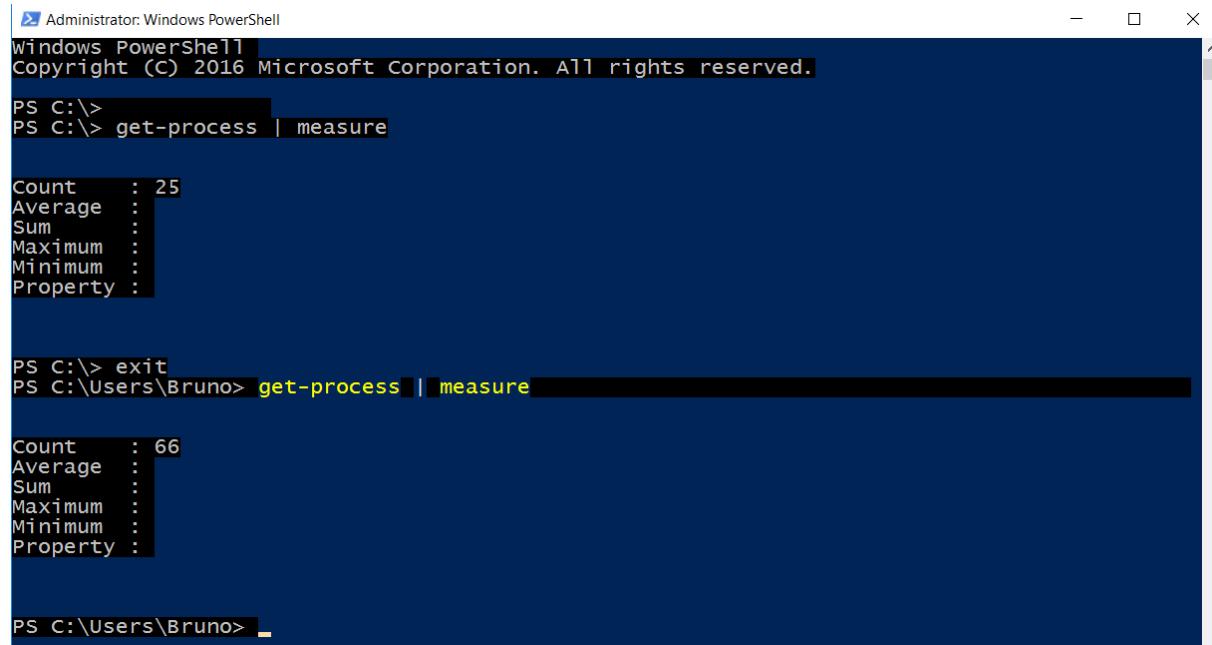
PS C:\>
```

Sortez du conteneur avec la commande :

exit

Puis refaire la même commande :

get-process | measure



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\>
PS C:\> get-process | measure

Count      : 25
Average    :
Sum        :
Maximum   :
Minimum   :
Property  :

PS C:\> exit
PS C:\Users\Bruno> get-process | measure

Count      : 66
Average    :
Sum        :
Maximum   :
Minimum   :
Property  :

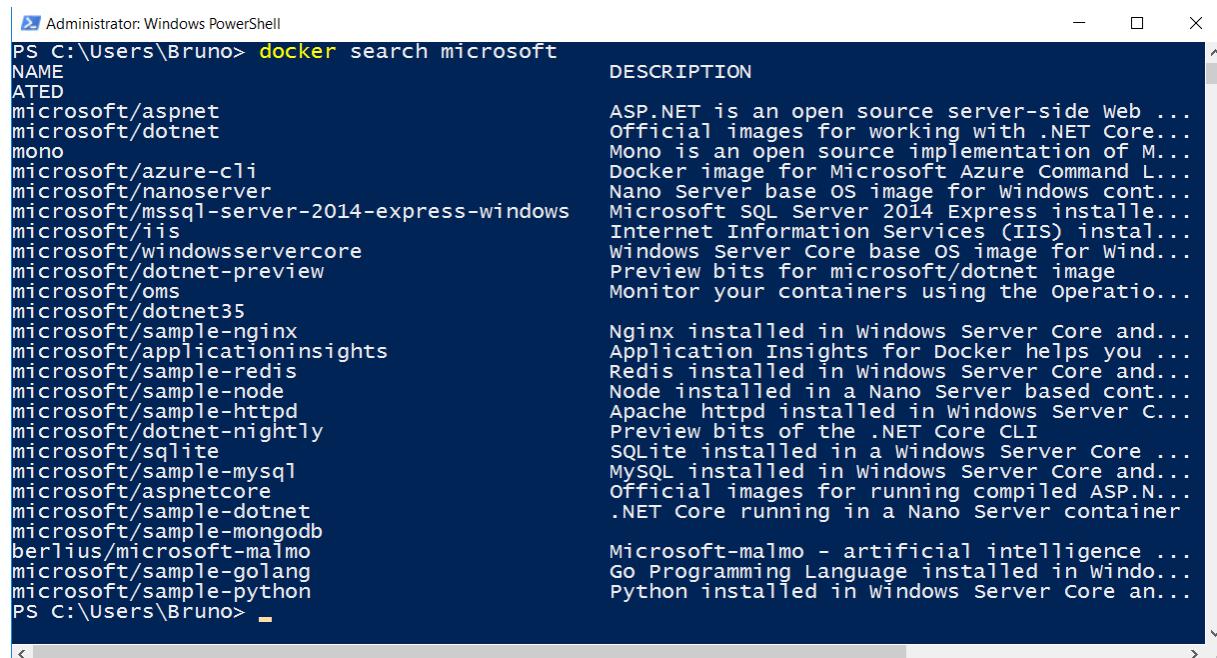
PS C:\Users\Bruno>
```

Ce premier contact vous permet de comprendre qu'un conteneur est bien plus restreint qu'un système complet (grosse différence avec une VM).



Si la connexion réseau est bonne, nous pouvons aller chercher des images provenant d'autres entrepôts. Le Hub Docker est un entrepôt public contenant de nombreuses images. Nous pouvons obtenir la liste des images microsoft en lançant la commande suivante :

docker search microsoft



```

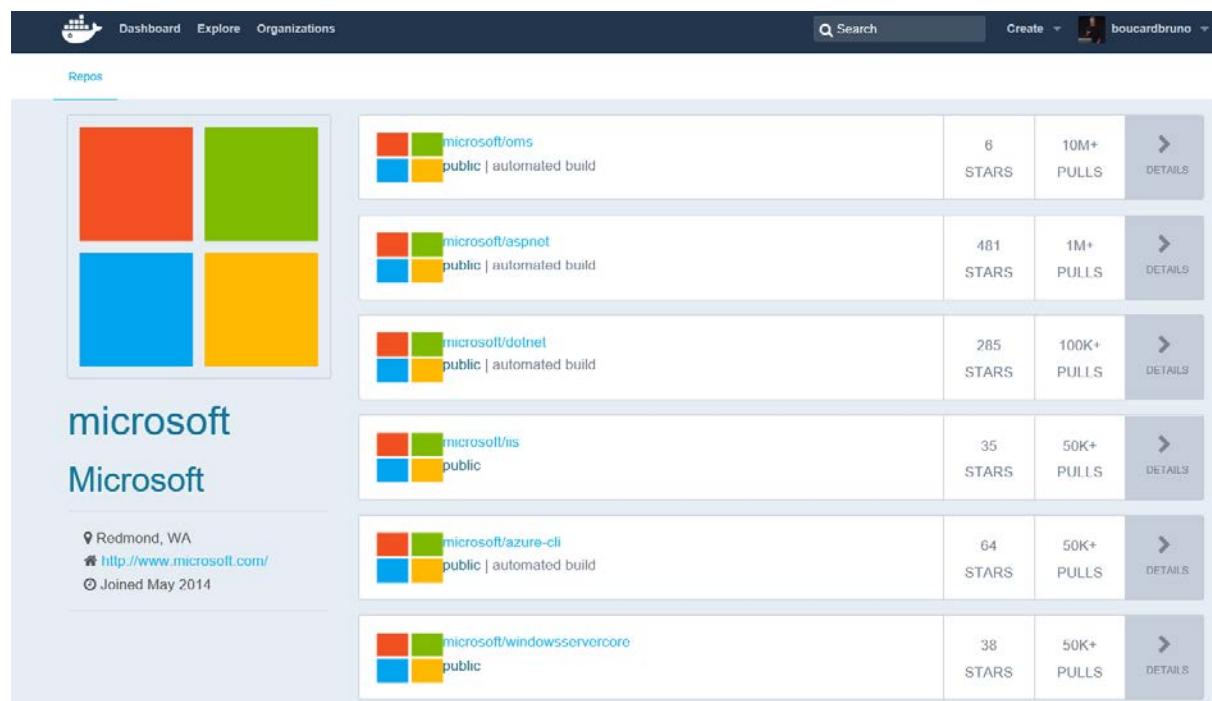
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker search microsoft
NAME                                            DESCRIPTION
microsoft/aspnet                                ASP.NET is an open source server-side Web ...
microsoft/dotnet                                 Official images for working with .NET Core...
mono                                              Mono is an open source implementation of M...
microsoft/azure-cli                               Docker image for Microsoft Azure Command L...
microsoft/nanoserver                            Nano Server base OS image for Windows cont...
microsoft/mssql-server-2014-express-windows     Microsoft SQL Server 2014 Express installe...
microsoft/iis                                    Internet Information Services (IIS) instal...
microsoft/windowsservercore                      Windows Server Core base OS image for Wind...
microsoft/dotnet-preview                         Preview bits for microsoft/dotnet image
microsoft/oms                                     Monitor your containers using the Operatio...
microsoft/dotnet35                               Nginx installed in Windows Server Core and...
microsoft/sample-nginxx                          Application Insights for Docker helps you ...
microsoft/applicationinsights                   Redis installed in Windows Server Core and...
microsoft/sample-redis                           Node installed in a Nano Server based cont...
microsoft/sample-node                           Apache httpd installed in windows Server C...
microsoft/sample-httdp                          Preview bits of the .NET Core CLI
microsoft/dotnet-nightly                        SQLite installed in a Windows Server Core ...
microsoft/sqlite                                MySQL installed in Windows Server Core and...
microsoft/sample-mysql                           Official images for running compiled ASP.N...
microsoft/aspnetcore                           .NET Core running in a Nano Server container
microsoft/sample-dotnet                         Microsoft-malmo - artificial intelligence ...
microsoft/sample-mongodbc                      Go Programming Language installed in Windo...
microsoft/sample-golang                         Python installed in Windows Server Core an...
microsoft/sample-python                         PS C:\Users\Bruno>

```

Vous pouvez aussi observer les images spécifiques Microsoft sur le hub. Sur votre machine, dans un browser tapez l'url suivante :

<https://hub.docker.com/u/microsoft/>





microsoft
Microsoft

Redmond, WA
http://www.microsoft.com/
Joined May 2014

Nous sommes intéressés par une image en particulier, l'image contenant le serveur Microsoft IIS. Nous lançons la commande :

docker pull microsoft/iis:TP5

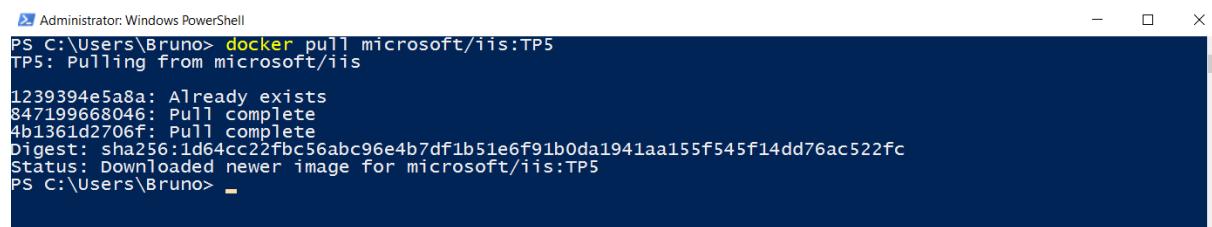
(Explication : nous précisons le tag TP5 car l'entrepôt dispose par défaut d'images pour la version RTM des Windows Containers qui n'est pas actuellement disponible sur Microsoft Azure)



```
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker pull microsoft/iis:TP5
TP5: Pulling from microsoft/iis
1239394e5a8a: Already exists
847199668046: Pull complete
4b1361d2706f: Extracting 20.55 MB/35.33 MB
```

Le téléchargement d'une image peut se révéler plus ou moins long en fonction de la connectivité réseau et la taille de l'image. Par nature une image docker est constituée de couches téléchargeables indépendamment. C'est une caractéristique importante dans la compréhension générale de Docker. Par nature une image est une composition de couches.



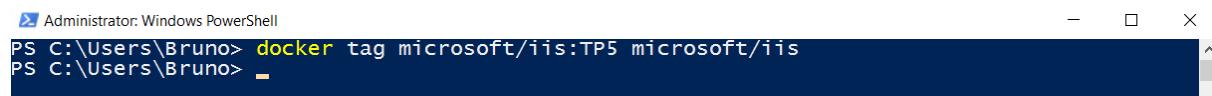


```
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker pull microsoft/iis:TP5
TP5: Pulling from microsoft/iis
1239394e5a8a: Already exists
847199668046: Pull complete
4b1361d2706f: Pull complete
Digest: sha256:1d64cc22fb56abc96e4b7df1b51e6f91b0da1941aa155f545f14dd76ac522fc
Status: Downloaded newer image for microsoft/iis:TP5
PS C:\Users\Bruno>
```

Après quelques minutes, la nouvelle image est installée.

Nous allons ajouter un nouveau tag sur cette nouvelle image, afin de s'affranchir du suffixe TP5.

docker tag microsoft/iis:TP5 microsoft/iis

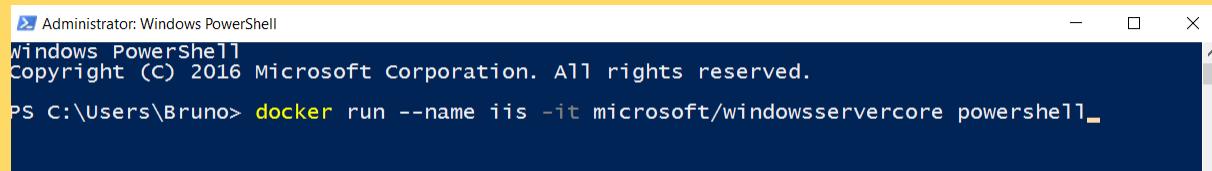


```
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker tag microsoft/iis:TP5 microsoft/iis
PS C:\Users\Bruno>
```

VOUS DISPOSEZ D'UNE CONNEXION RESEAU LENTE

Si votre connexion réseau ne vous permet pas de télécharger rapidement cette image, voici une autre façon d'obtenir une image identique.

docker run --name iis -it microsoft/windowsservercore powershell

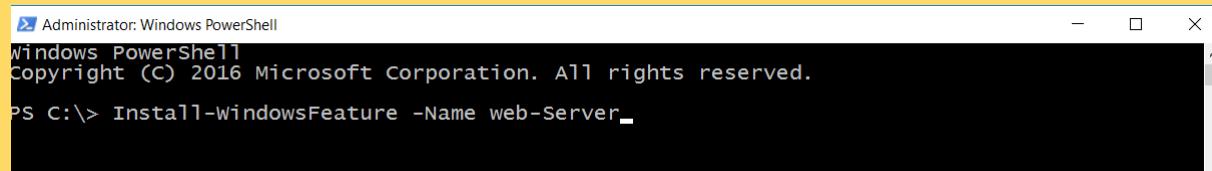


```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Bruno> docker run --name iis -it microsoft/windowsservercore powershell
```

Puis une fois dans le conteneur taper la commande:

Install-WindowsFeature -Name Web-Server

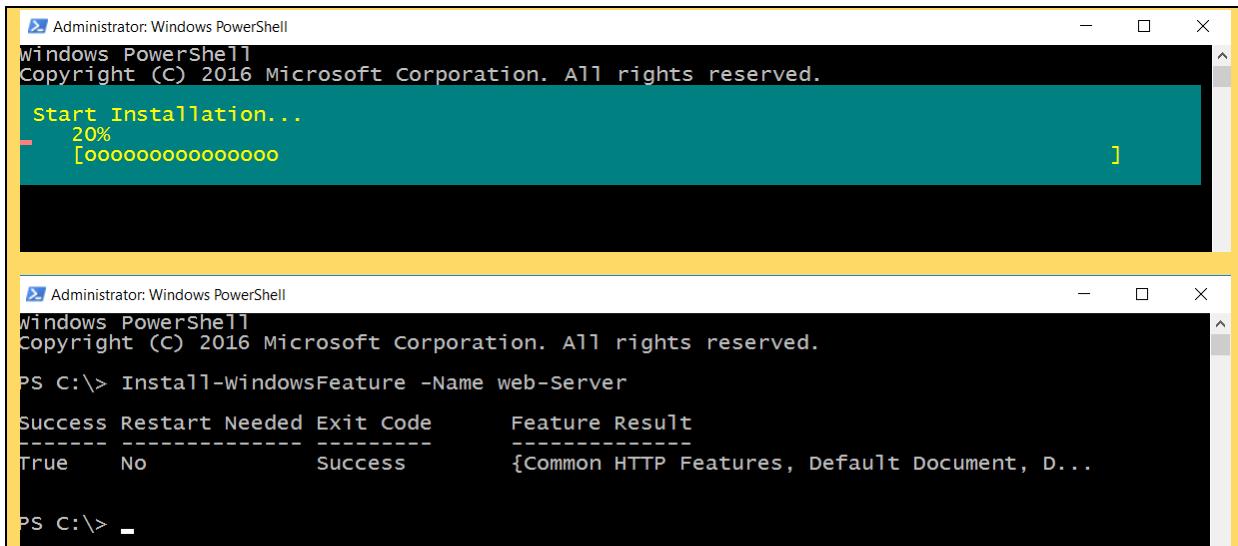


```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\> Install-WindowsFeature -Name web-Server
```

L'installation de serveur web IIS ne prend que quelques secondes





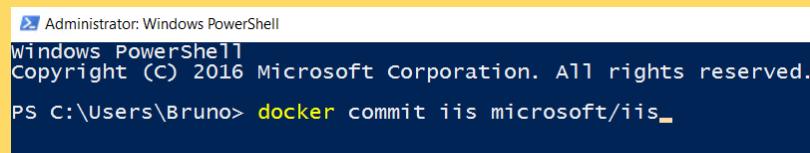
The screenshot shows two stacked PowerShell windows. The top window is titled 'Administrator: Windows PowerShell' and displays a progress bar indicating a 'Start Installation...' process at 20% completion. The bottom window is also titled 'Administrator: Windows PowerShell' and shows the command 'Install-WindowsFeature -Name web-Server' being run. A table output shows the success of the operation, a restart needed (No), and the exit code (Success). The feature installed is listed as 'Common HTTP Features, Default Document, D...'. The command 'PS C:\> -' is shown at the bottom.

L'installation est terminée, nous pouvons sortir du conteneur :

exit

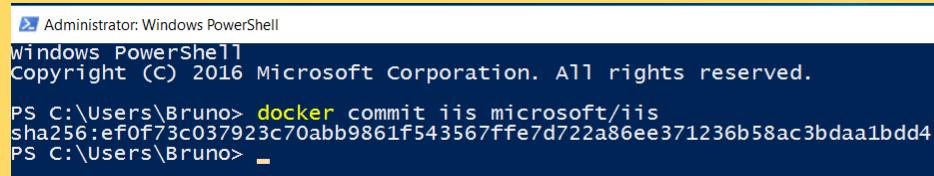
Nous pouvons créer une nouvelle image sur la base du conteneur « iis » :

docker commit iis microsoft/iis



The screenshot shows a single PowerShell window with the command 'docker commit iis microsoft/iis' entered. The command is highlighted in yellow.

Après quelques secondes, une nouvelle image est engendrée.



The screenshot shows a single PowerShell window displaying the output of the 'docker commit' command. It shows the SHA-256 hash of the new image: 'sha256:ef0f73c037923c70abb9861f543567ffe7d722a86ee371236b58ac3bdaa1bdd4'. The command is highlighted in yellow.

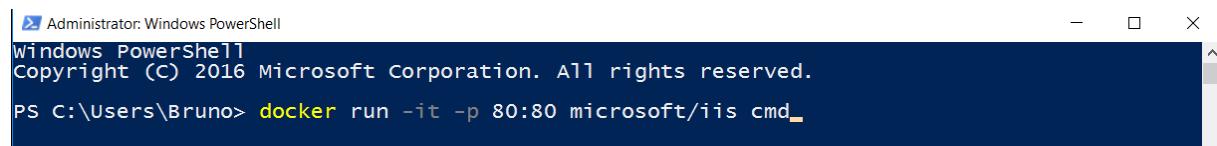
A ce stade, vous pouvez reprendre la suite de l'atelier.



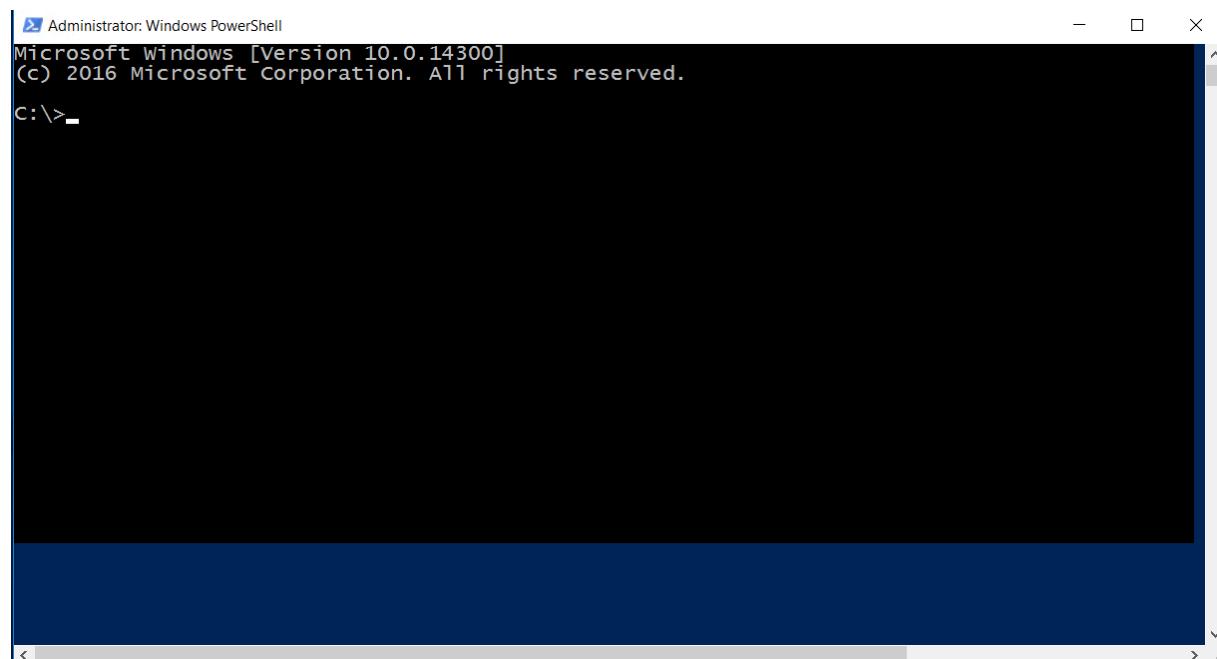
Nous pouvons lancer notre second conteneur en vous donnant plus d'explications.

```
docker run -it -p 80:80 microsoft/iis cmd
```

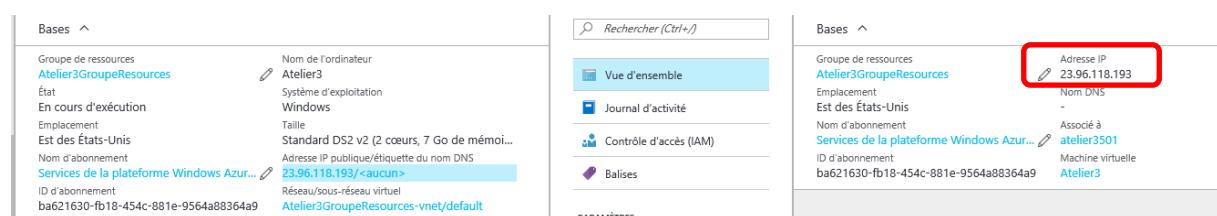
La ligne de commande permet d'exécuter (run) un conteneur basé sur une image contenant le serveur IIS (microsoft/iis) en mode interactif (-it) en précisant la correspondance des ports entre la machine host et le conteneur (-p 80 :80). Enfin nous précisons la commande qui sera exécuté au démarrage du conteneur (cmd).



Après quelques secondes, vous êtes dans le conteneur.



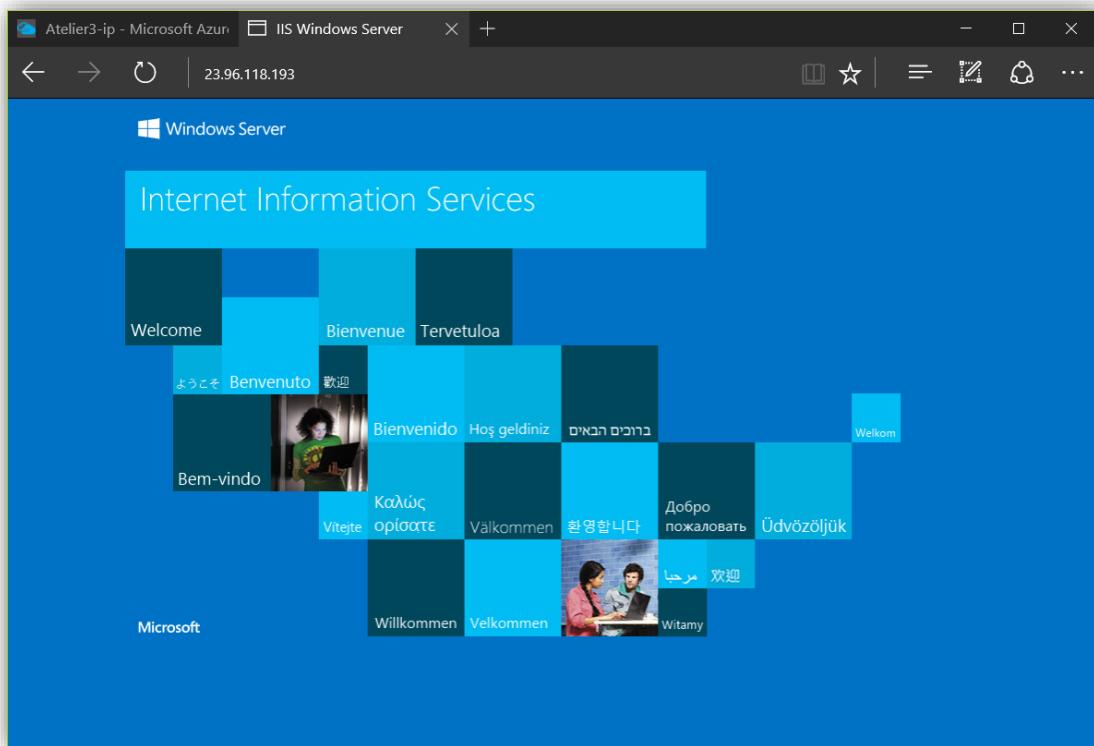
Sur le portail Azure, récupérez l'adresse IP publique virtuelle.



Informations	Valeur
Groupe de ressources	Atelier3GroupeResources
État	En cours d'exécution
Emplacement	Est des États-Unis
Nom d'abonnement	Services de la plateforme Windows Azur...
ID d'abonnement	ba621630-fb18-454c-881e-9564a88364a9
Nom de l'ordinateur	Atelier3
Système d'exploitation	Windows
Taille	Standard DS2 v2 (2 coeurs, 7 Go de mémoi...)
Adresse IP publique/étiquette du nom DNS	23.96.118.193</aucun>
Réseau/sous-réseau virtuel	Atelier3GroupeResources-vnet/default



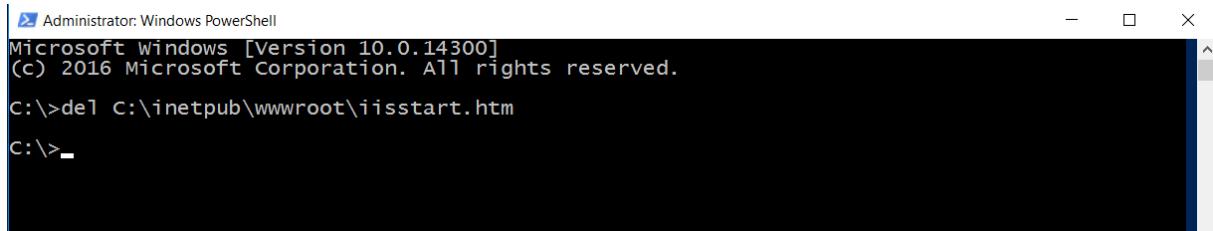
Depuis votre poste travail, taper l'IP dans un browser.



Vous devriez observer l'écran d'accueil du serveur IIS.

Retournez dans votre conteneur, puis taper la commande:

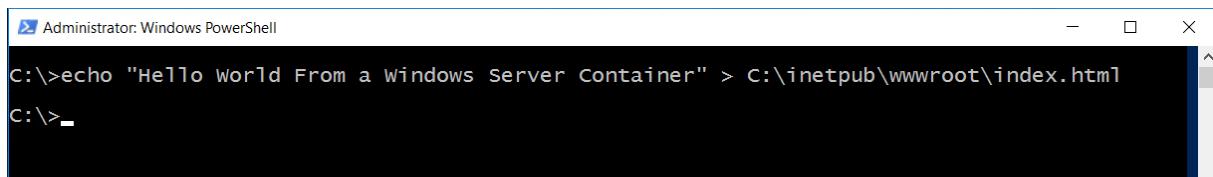
```
del C:\inetpub\wwwroot\iisstart.htm
```



A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The window shows the command "del C:\inetpub\wwwroot\iisstart.htm" being run, followed by a new line prompt "c:\>". The background of the window is black.

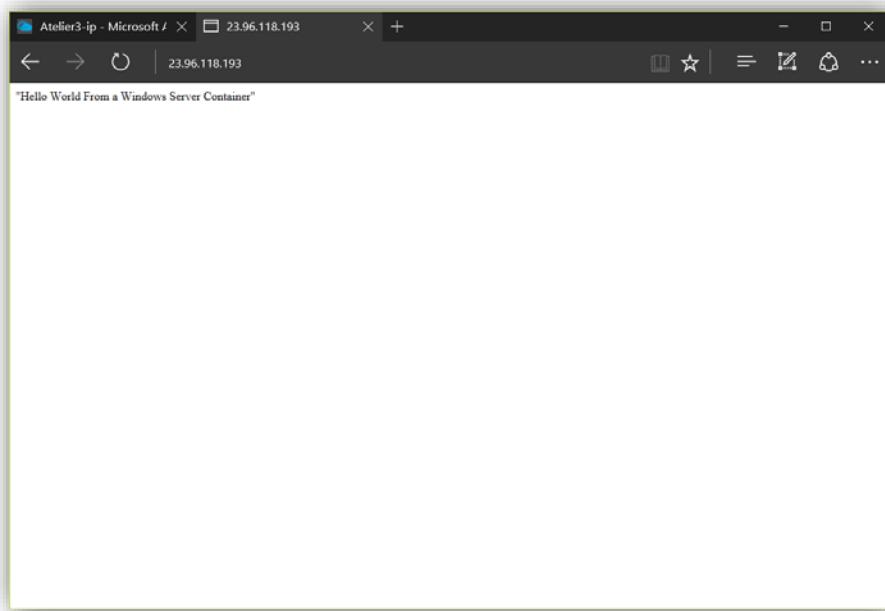
Puis la commande:

```
echo "Hello World From a Windows Server Container"  
> C:\inetpub\wwwroot\index.html
```



A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The window shows the command "echo "Hello World From a Windows Server Container" > c:\inetpub\wwwroot\index.html" being run, followed by a new line prompt "c:\>". The background of the window is black.

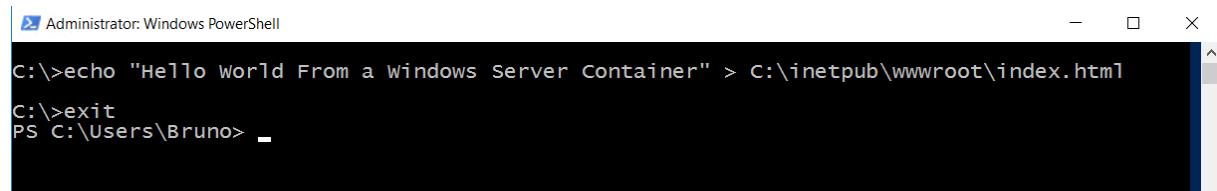
Depuis votre poste travail, dans votre browser, rafraîchissez la page.



Vous devriez observer une page similaire.

Retournez dans votre conteneur, puis lancer la commande suivante :

exit

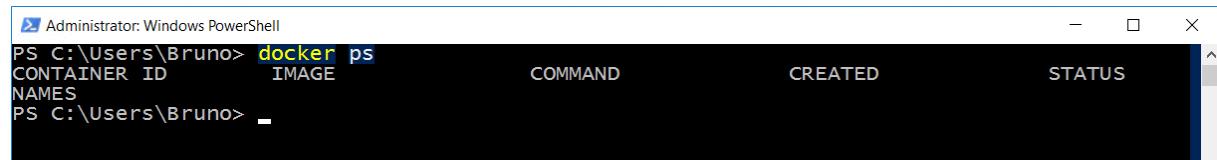


```
Administrator: Windows PowerShell
C:\>echo "Hello World From a Windows Server Container" > C:\inetpub\wwwroot\index.html
C:\>exit
PS C:\Users\Bruno>
```

Vous êtes sorti du conteneur et revenu sous l'host, ici la session PowerShell de départ.

Nous avons déjà lancé la commande :

docker ps

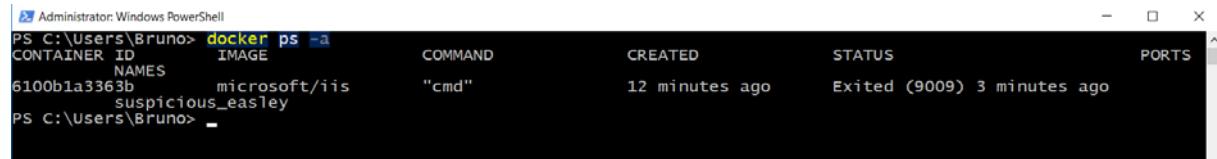


```
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker ps
CONTAINER ID        IMAGE               COMMAND       CREATED          STATUS
NAMES
PS C:\Users\Bruno>
```

Cette commande permet d'observer les conteneurs en cours d'exécution. Comme nous sommes sortis de notre premier conteneur, il n'y a plus de conteneur actif.

Cependant, lorsque nous faisons cette commande :

docker ps -a



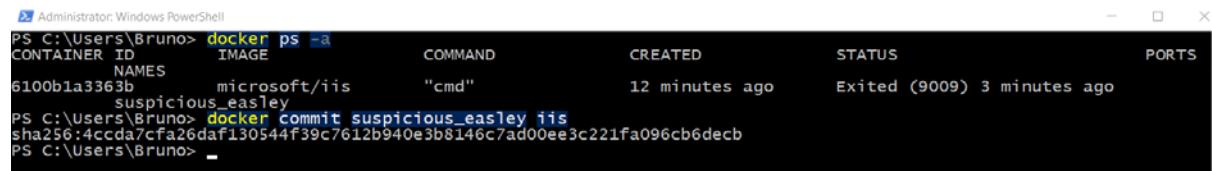
```
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker ps -a
CONTAINER ID        IMAGE               COMMAND       CREATED          STATUS            PORTS
NAMES
6100b1a3363b        microsoft/iis      "cmd"         12 minutes ago   Exited (9009) 3 minutes ago
suspicuous_easley
PS C:\Users\Bruno>
```

Nous observons la trace de l'exécution de notre conteneur. Il y a deux moyens pour identifier un conteneur : son ID et son nom.



Par nature les images docker sont immuables (c'est une notion importante). Si vous souhaitez modifier une image, vous devez en créer une nouvelle. Lancer la commande :

```
docker commit suspicious_easley iis
```



```
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker ps -a
CONTAINER ID        IMAGE               COMMAND       CREATED          STATUS           PORTS
 NAMES              microsoft/iis      "cmd"         12 minutes ago   Exited (9009) 3 minutes ago
6100b1a3363b        suspicious_easley
PS C:\Users\Bruno> docker commit suspicious_easley iis
sha256:4ccda7cfa26daf130544f39c7612b940e3b8146c7ad00ee3c221fa096cb6decb
PS C:\Users\Bruno>
```

Nous avons engendré une nouvelle image que nous avons appelée « iis » sur la base d'état du conteneur précédent, appelé « suspicious_easley ». Par défaut docker engendre automatiquement un nom à conteneur anonyme. Vous pouvez imposer un nom avec option --name.

Nous pouvons donc démarrer un nouveau conteneur sur la base de cette nouvelle image. Lancez la commande :

```
docker run -it --name iis -p 80:80 iis cmd
```

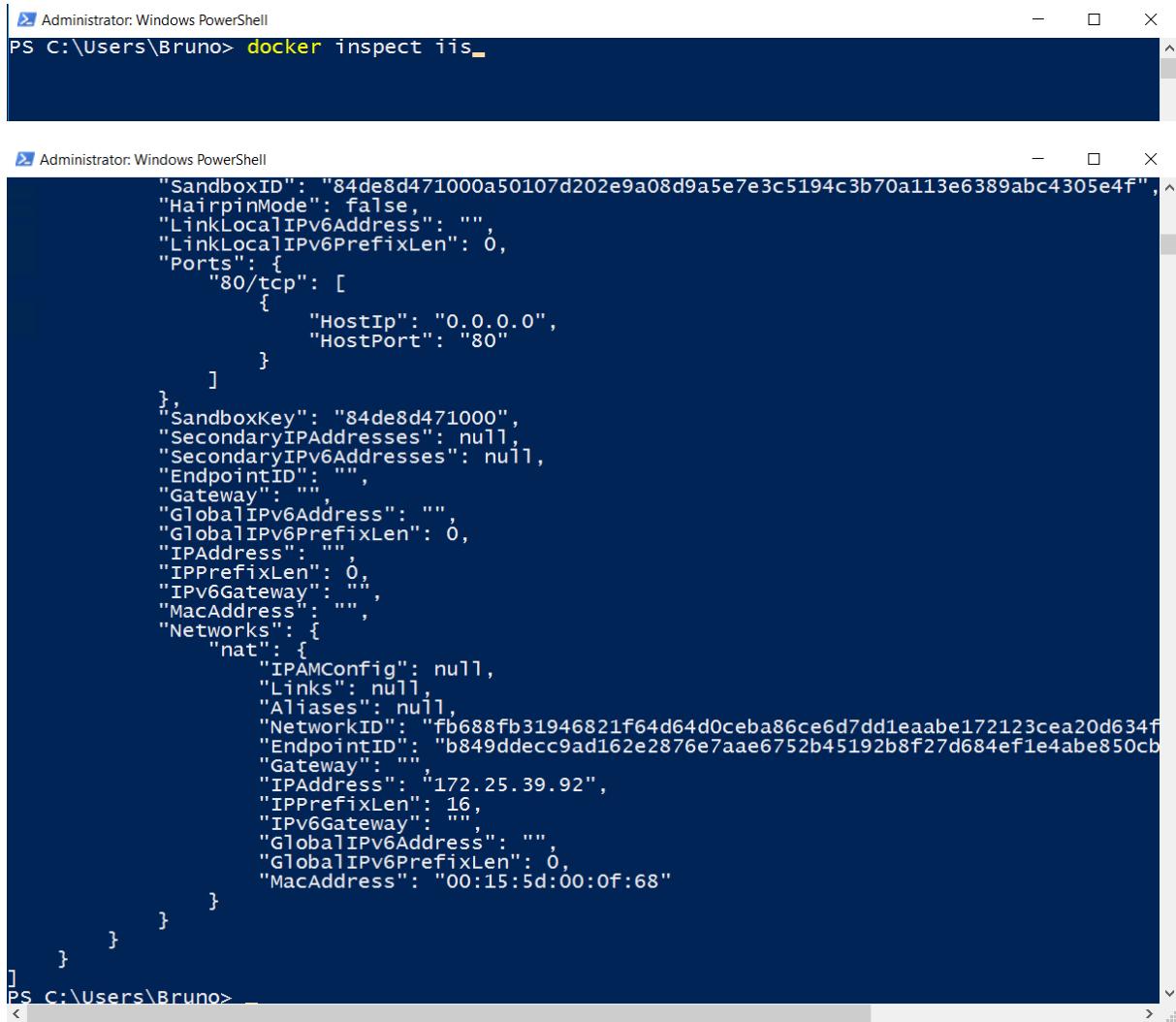


```
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker run -it -p 80:80 iis cmd
```



Une fois dans votre conteneur. Lancer une autre fenêtre powershell

docker inspect iis



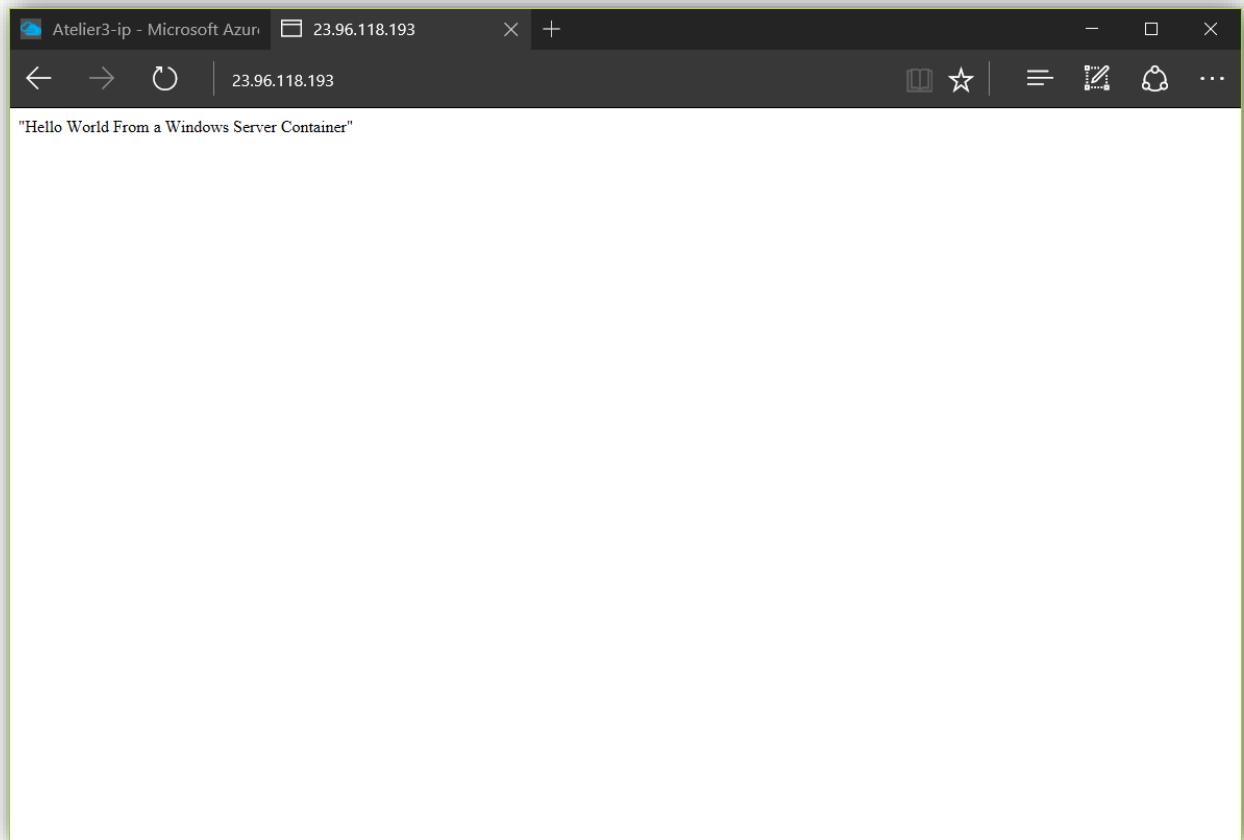
```

Administrator: Windows PowerShell
PS C:\Users\Bruno> docker inspect iis_
[Output is too long to show here, but it shows a detailed JSON object representing the Docker container configuration, including network settings, ports, and host IP mapping for port 80/tcp.]
```

Ceci vous permet d'observer toutes les caractéristiques d'un conteneur en cours d'exécution.



Depuis votre poste travail, rafraîchissez la page.



Vous devriez retrouver une page web similaire.

Puis vous sortez du conteneur et lancer les commandes similaires :

exit

docker ps -a

docker rm id_conteneur1

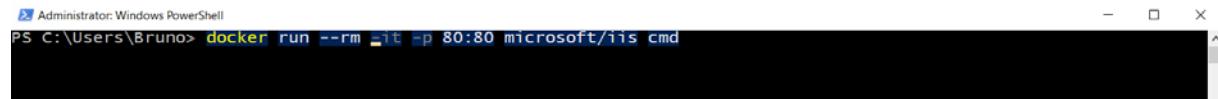
docker rm id_conteneur2

docker ps -a



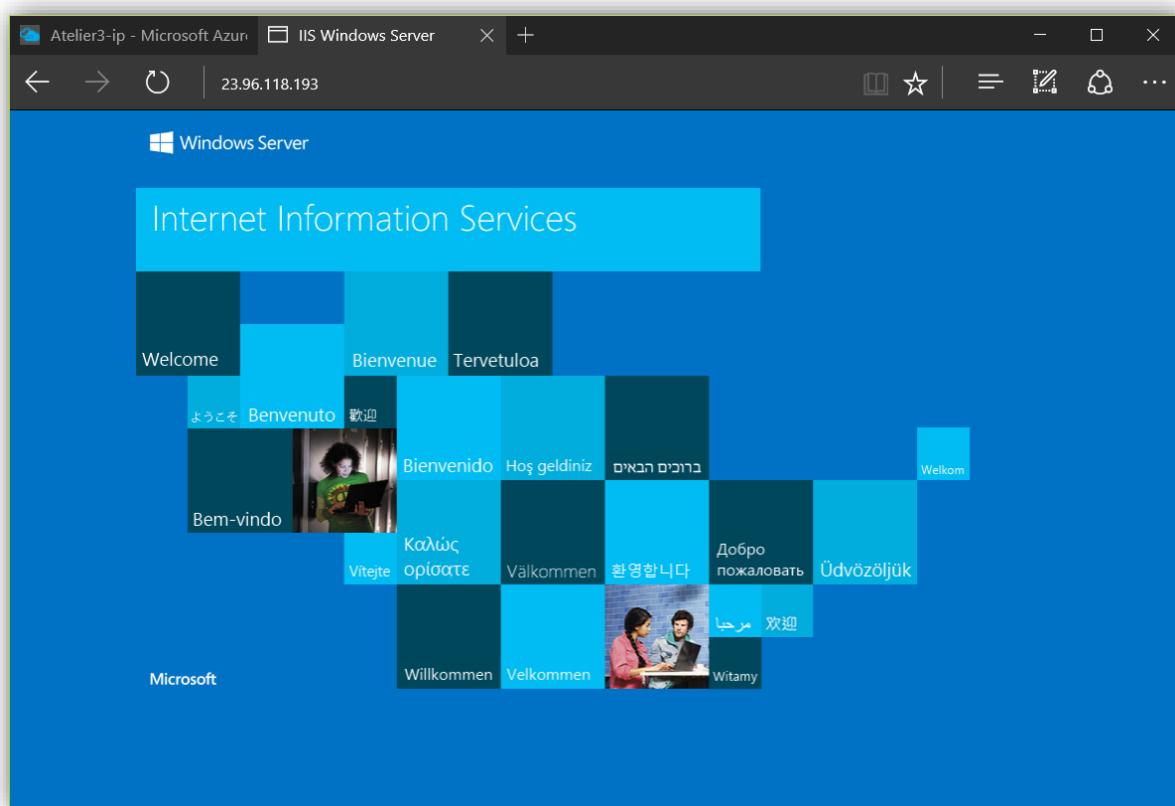
La commande docker rm {id conteneur | nom}, permet de supprimer un conteneur. Une fois que vous avez supprimé la trace de tous les conteneurs, vous lancez la commande :

```
docker run --rm -it -p 80:80 microsoft/iis cmd
```



Vous avez relancé un conteneur dont la configuration est presque identique au tout premier, car nous avons précisé l'option --rm qui permet de nettoyer la trace du conteneur une fois terminé.

Depuis votre poste travail, rafraîchissez la page



On retrouve la page d'accueil du serveur IIS.

Maintenant, retournez dans votre conteneur, puis taper les commandes:

exit

```
docker ps -a
```

```
[Administrator: Windows PowerShell]
Microsoft Windows [Version 10.0.14300]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\>exit
PS C:\Users\Bruno> docker ps -a
CONTAINER ID        IMAGE               COMMAND       CREATED          STATUS          PORTS
PS C:\Users\Bruno>
```

On note que l'option `--rm` a permis de ne pas laisser de trace d'exécution de notre conteneur.



Automatiser avec un fichier dockerfile

Dans l'activité précédente, nous avons principalement frappé des commandes à la main. C'est une excellente méthode, lorsque nous souhaitons découvrir les bases de docker.

Cependant, il existe un moyen simple d'automatiser des commandes : le Dockerfile. À l'instar du makefile qui automatise la compilation. Le dockerfile permet d'automatiser la création d'une image.

Commençons par créer un répertoire de travail.

```
md c:\build
```

```
Administrator: Windows PowerShell
PS C:\Users\Bruno> md c:\build

Directory: C:\

Mode                LastWriteTime         Length Name
----                -----          ----- ----
d----- 9/15/2016   6:18 PM            build

PS C:\Users\Bruno> -
```



Création d'un fichier dockerfile via powershell :

```
powershell new-item c:\build\Dockerfile -Force
```

```
Administrator: Windows PowerShell
PS C:\Users\Bruno> powershell new-item c:\build\Dockerfile -Force

Directory: C:\build

Mode          LastWriteTime
----          -----
-a---  9/15/2016  6:21 PM

Length Name
----- -
          0 Dockerfile

PS C:\Users\Bruno> -
```

Lancer notepad pour éditer le fichier Dockerfile

```
Administrator: Windows PowerShell
PS C:\Users\Bruno> notepad c:\build\Dockerfile_
```

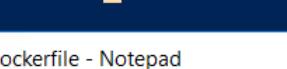
Tapez les trois lignes suivantes :

```
FROM microsoft/iis

RUN del C:\inetpub\wwwroot\iisstart.htm

RUN echo "Hello World - Dockerfile" > c:\inetpub\wwwroot\index.html
```

 Administrator: Windows PowerShell
PS C:\build> notePad .\Dockerfile
PS C:\build> -



Dockerfile - Notepad

File Edit Format View Help

```
FROM microsoft/iis
RUN del C:\inetpub\wwwroot\iisstart.htm
RUN echo "Hello World - Dockerfile" > c:\inetpub\wwwroot\index.html
```



Génération d'une nouvelle image iis-dockerfile avec la commande docker build. Placez-vous dans le répertoire C:\build puis lancez la commande, sans oublier le point à la fin :

```
docker build -t iis-dockerfile .
```

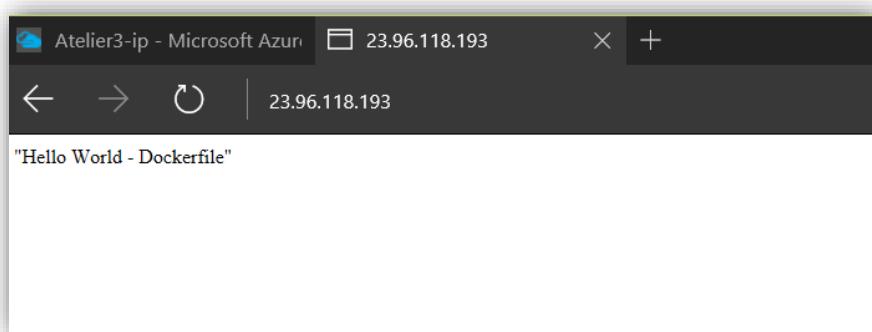
```
Administrator: Windows PowerShell
PS C:\build> docker build -t iis-dockerfile .
Sending build context to Docker daemon 2.048 kB
Step 1 : FROM microsoft/iis
--> accd044753c1
Step 2 : RUN del C:\inetpub\wwwroot\iisstart.htm
--> Running in fe6bdf9bba1
--> 8818049efdd2
Removing intermediate container fe6bdf9bba1
Step 3 : RUN echo "Hello World - Dockerfile" > c:\inetpub\wwwroot\index.html
--> Running in ebfbd689a392
--> cbf6b9cc9db6
Removing intermediate container ebfbd689a392
Successfully built cbf6b9cc9db6
PS C:\build> -
```

Lancer la commande suivante :

```
docker run -d --rm -p 80:80 iis-dockerfile ping -t localhost
```

On a lancé un nouveau conteneur sur la base de l'image générée par la commande build sur la base du fichier dockerfile. On utilise le mode -d (mode démon) pour rendre la main dès le lancement du conteneur.

Depuis votre poste travail, rafraîchissez la page



Lancer les commandes suivantes afin d'arrêter le conteur

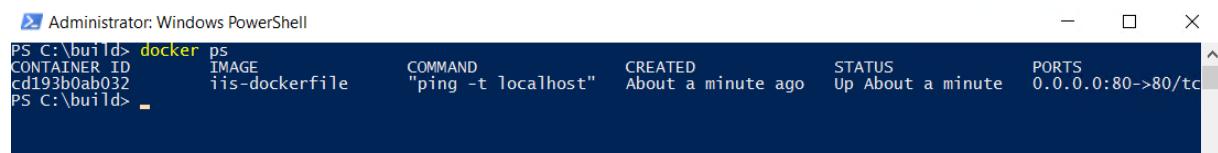
```
Administrator: Windows PowerShell
PS C:\build> docker ps
CONTAINER ID        IMAGE               COMMAND       CREATED          STATUS           PORTS
2f1514982b06        iis-dockerfile      "ping -t localhost"   4 minutes ago   Up 4 minutes   0.0.0.0:80->80/tcp
PS C:\build> docker rm 2f1514982b06 -f
```

Pour terminer le conteneur, nous récupérons son nom, via la commande :

```
docker ps
```

Puis nous le supprimons en mode force car le conteneur est cours d'exécution.

```
docker rm <id_conteneur/nom> -f
```

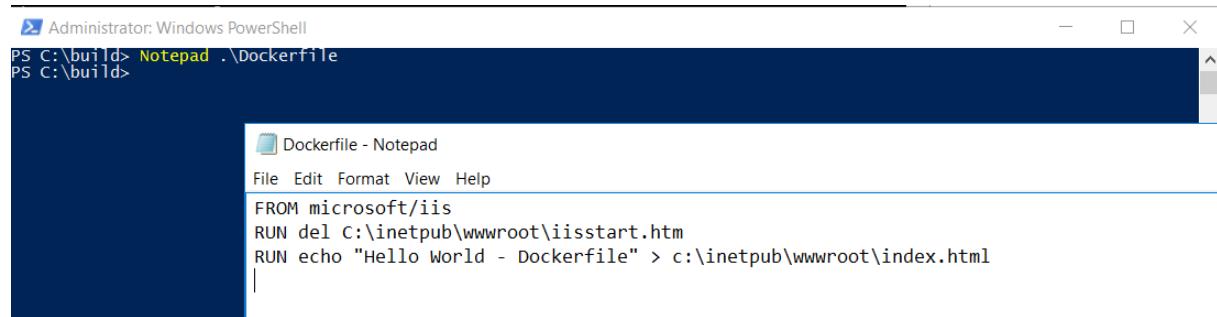


```
Administrator: Windows PowerShell
PS C:\build> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
cd193b0ab032        iis-dockerfile      "ping -t localhost"   About a minute ago   Up About a minute   0.0.0.0:80->80/tcp
PS C:\build>
```



Nous pourrions automatiser la commande ping. C'est une action que nous pouvons ajouter dans notre Dockerfile.

Lancer Notepad en ouvrant le Dockerfile :



```
Administrator: Windows PowerShell
PS C:\build> Notepad .\Dockerfile
PS C:\build>
```

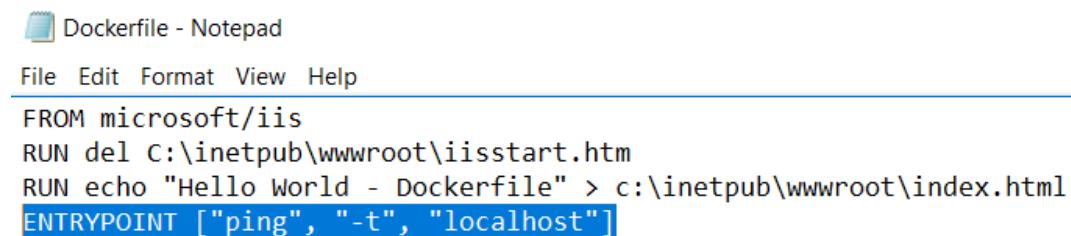
Dockerfile - Notepad

File Edit Format View Help

```
FROM microsoft/iis
RUN del C:\inetpub\wwwroot\iisstart.htm
RUN echo "Hello World - Dockerfile" > c:\inetpub\wwwroot\index.html
```

Ajouter la ligne :

```
ENTRYPOINT ["ping", "-t", "localhost"]
```



Dockerfile - Notepad

File Edit Format View Help

```
FROM microsoft/iis
RUN del C:\inetpub\wwwroot\iisstart.htm
RUN echo "Hello World - Dockerfile" > c:\inetpub\wwwroot\index.html
ENTRYPOINT ["ping", "-t", "localhost"]
```

Ceci permet de définir un point d'entrée au démarrage du conteneur. Vous sauvez le fichier Dockerfile et vous relancez la construction d'une nouvelle image :

```
docker build -t iis-dockerfile .
```

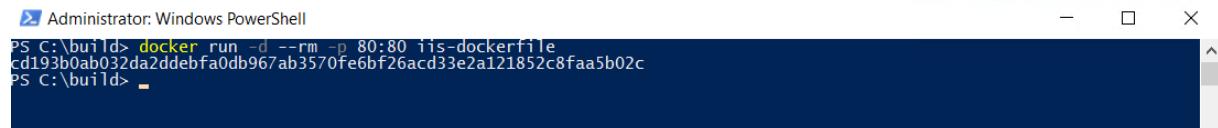


```
Administrator: Windows PowerShell
PS C:\build> docker build -t iis-dockerfile .
Sending build context to Docker daemon 2.048 kB
Step 1/4 : FROM microsoft/iis
--> e1e2cde45cd9
Step 2/4 : RUN del C:\inetpub\wwwroot\iisstart.htm
--> Using cache
--> 97d846cb4a08
Step 3/4 : RUN echo "Hello World - Dockerfile" > c:\inetpub\wwwroot\index.html
--> Using cache
--> 86dbb5703cca
Step 4/4 : ENTRYPOINT ping -t localhost
--> Using cache
--> 1bbd5edfb90
Successfully built 1bbd5edfb90
PS C:\build>
```

Vous sauvez lancez le conteneur en mode « démon » (n'attend pas la fin d'exécution du conteneur). Lancer la commande suivante :

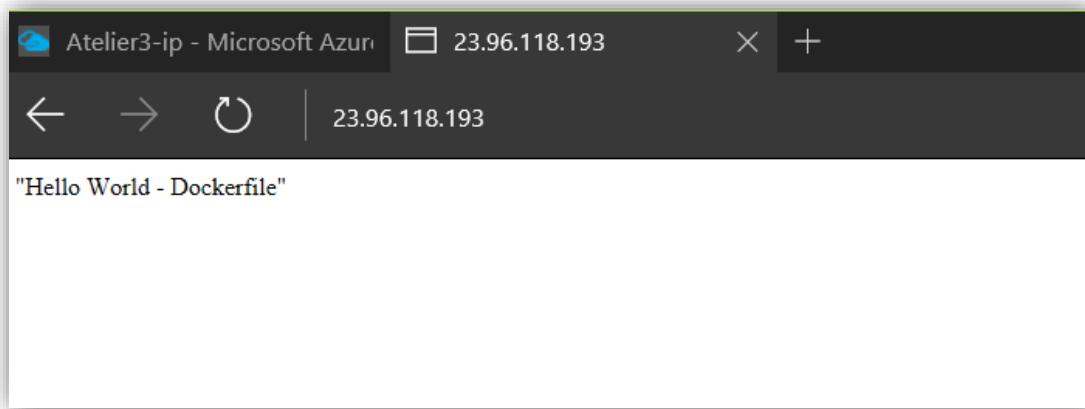


```
docker run -d --rm -p 80:80 iis-dockerfile
```



```
Administrator: Windows PowerShell
PS C:\build> docker run -d --rm -p 80:80 iis-dockerfile
cd193b0ab032da2ddebf0a0db967ab3570fe6bf26acd33e2a121852c8faa5b02c
PS C:\build> -
```

Depuis votre poste travail, rafraîchissez à nouveau la page :



Arrivez à ce stade vous accompli de nombreuses actions. Cette introduction aux Windows Containers vous a sans doute éclairé sur les bases de docker. N'hésitez pas à nous contacter si vous souhaitez plus d'information sur les Windows Containers.

