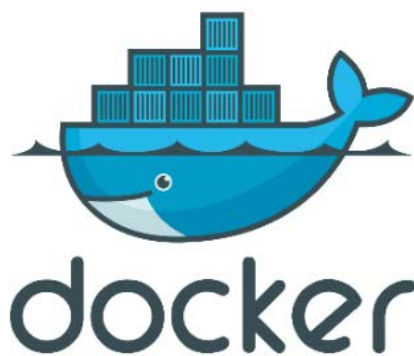


ATELIER 3



+



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

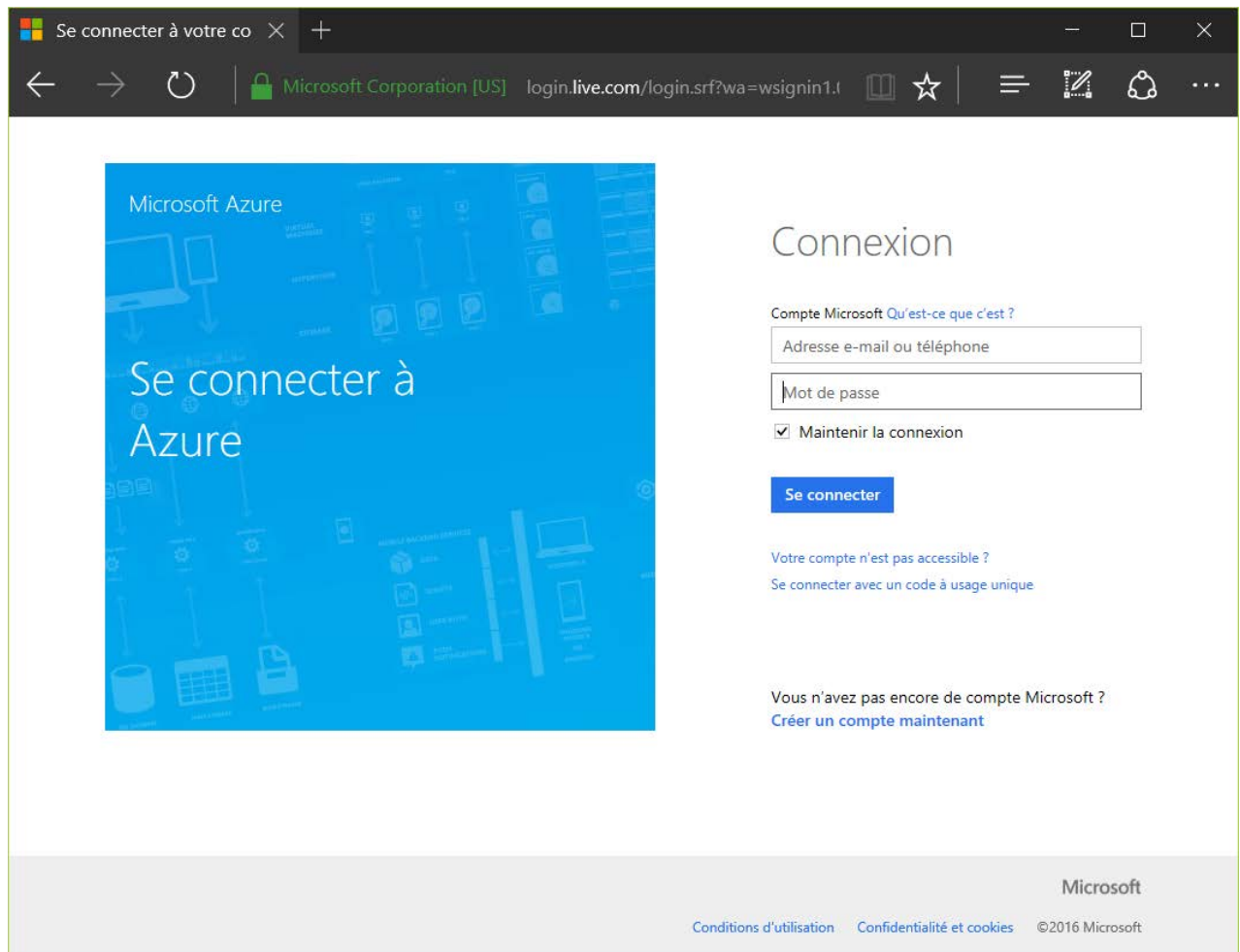
Dans ce premier acte, nous allons provisionner une machine virtuelle Windows Server 2016 avec les Windows Containers dans AZURE.

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

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

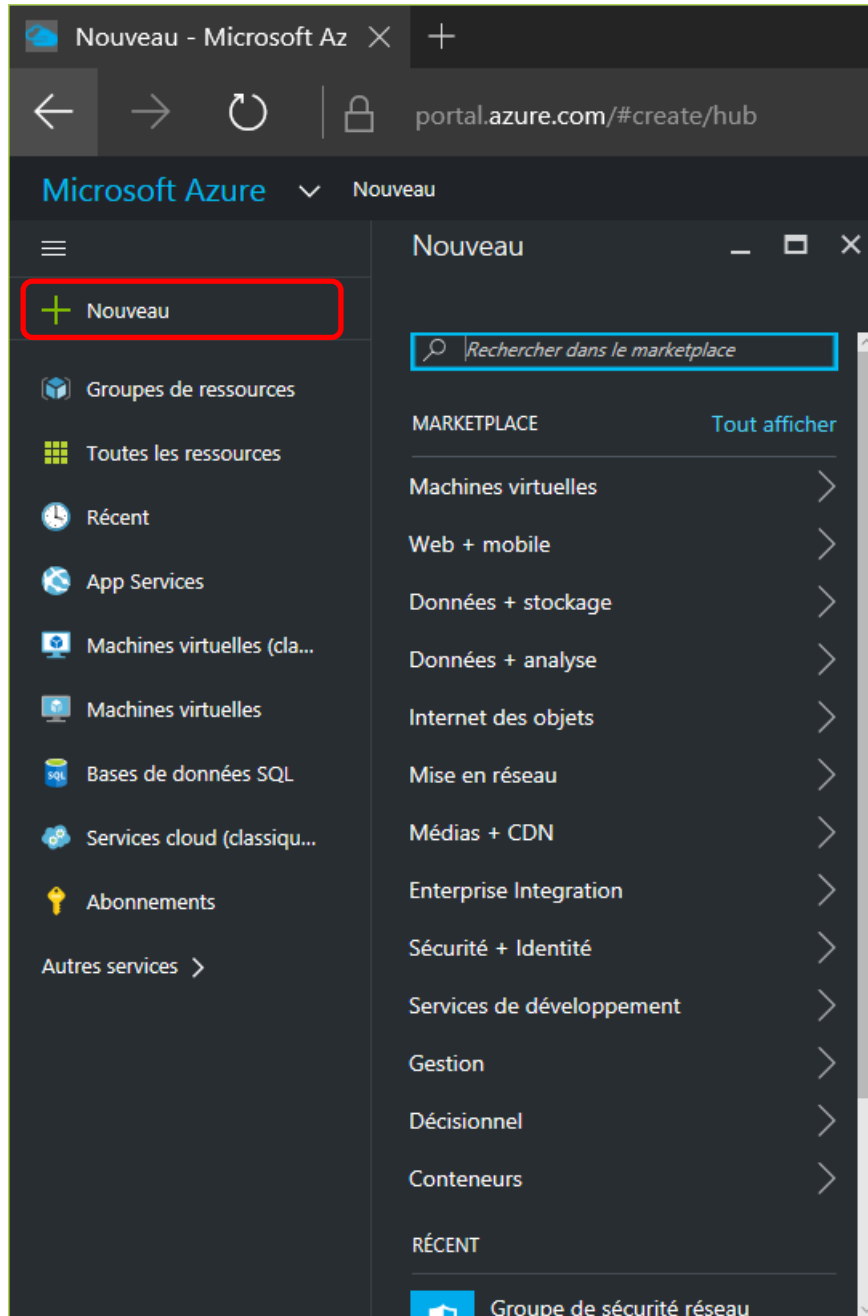
Vous serez redirigé sur une interface pour vous identifier si vous n'étiez pas précédemment connecté.

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



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

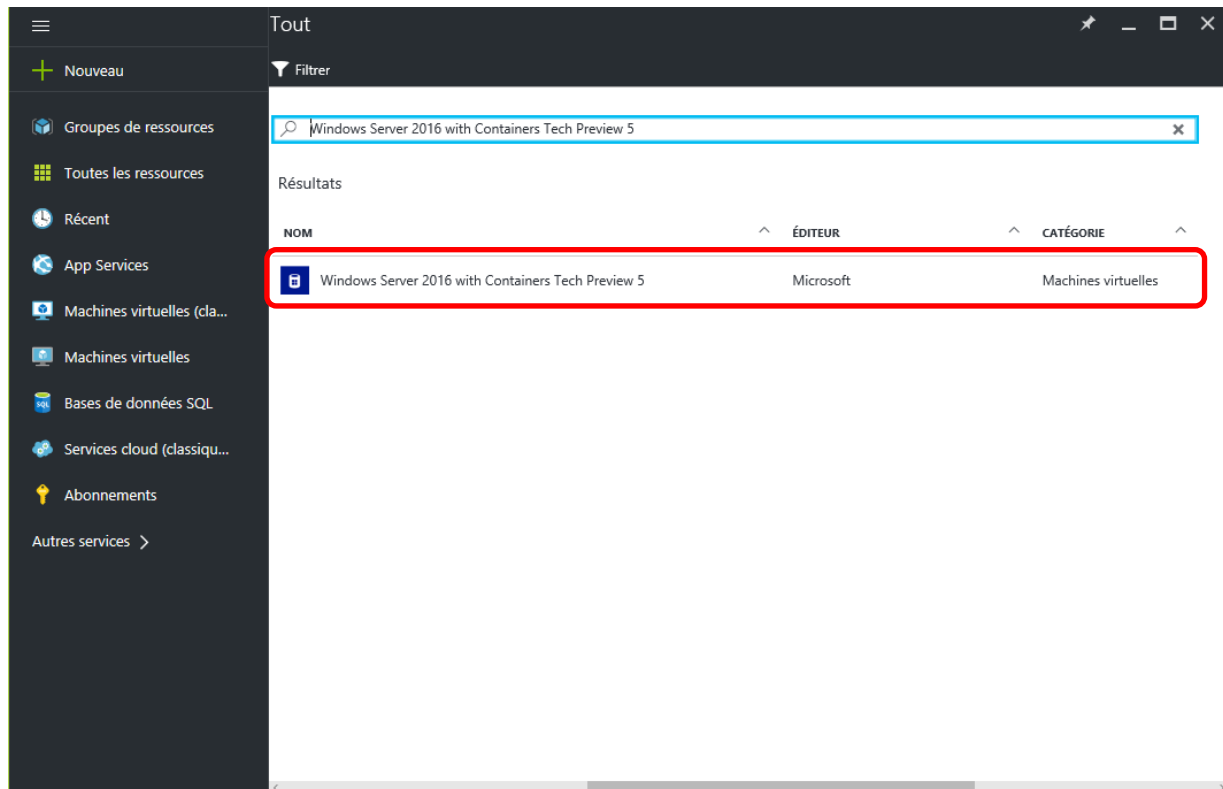
Une fois connecté, vous cliquez sur « Nouveau »



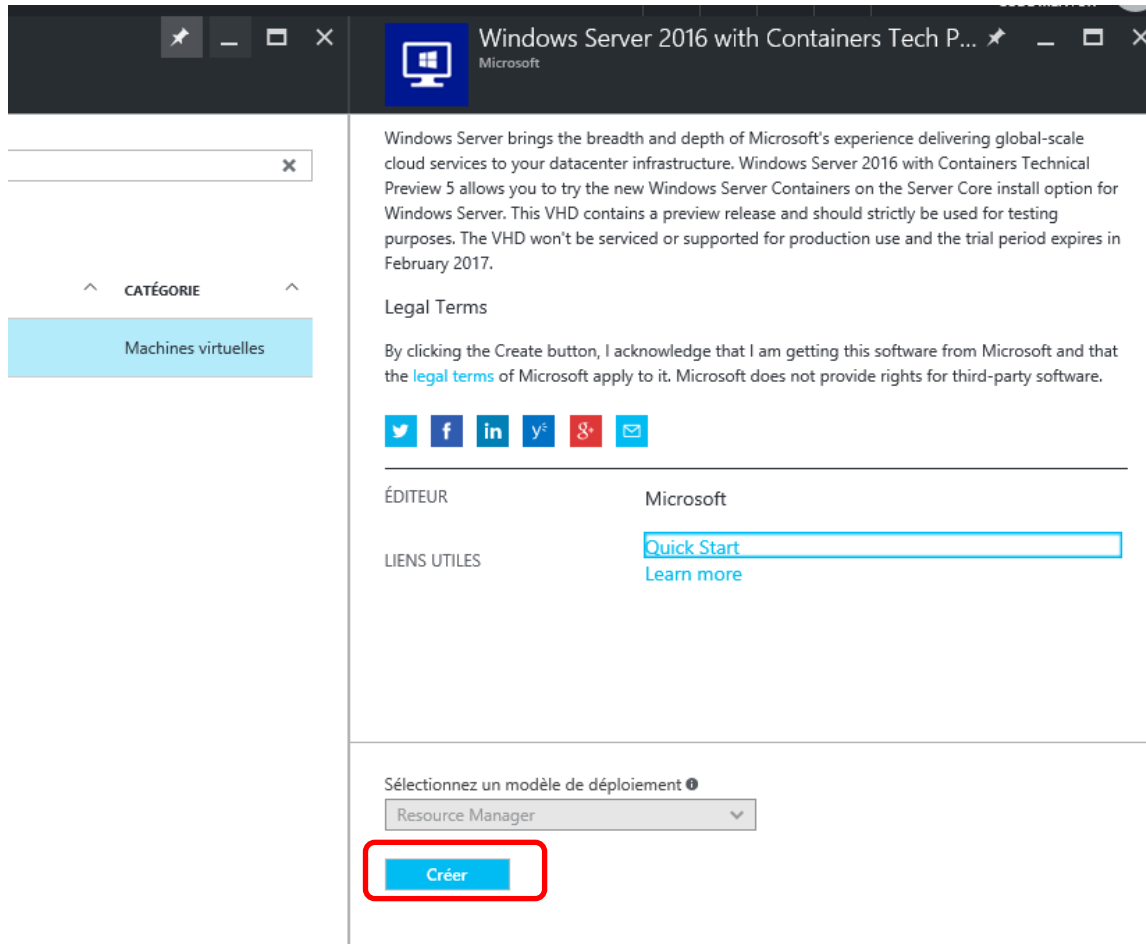
Et dans la partie recherche MARKETPLACE, saisissez le nom suivant :

Windows Server 2016 with Containers Tech Preview 5

Vous devriez trouver immédiatement la bonne VM.



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



Windows Server 2016 with Containers Tech P...

Windows Server brings the breadth and depth of Microsoft's experience delivering global-scale cloud services to your datacenter infrastructure. Windows Server 2016 with Containers Technical Preview 5 allows you to try the new Windows Server Containers on the Server Core install option for Windows Server. This VHD contains a preview release and should strictly be used for testing purposes. The VHD won't be serviced or supported for production use and the trial period expires in February 2017.

Legal Terms

By clicking the Create button, I acknowledge that I am getting this software from Microsoft and that the [legal terms](#) of Microsoft apply to it. Microsoft does not provide rights for third-party software.

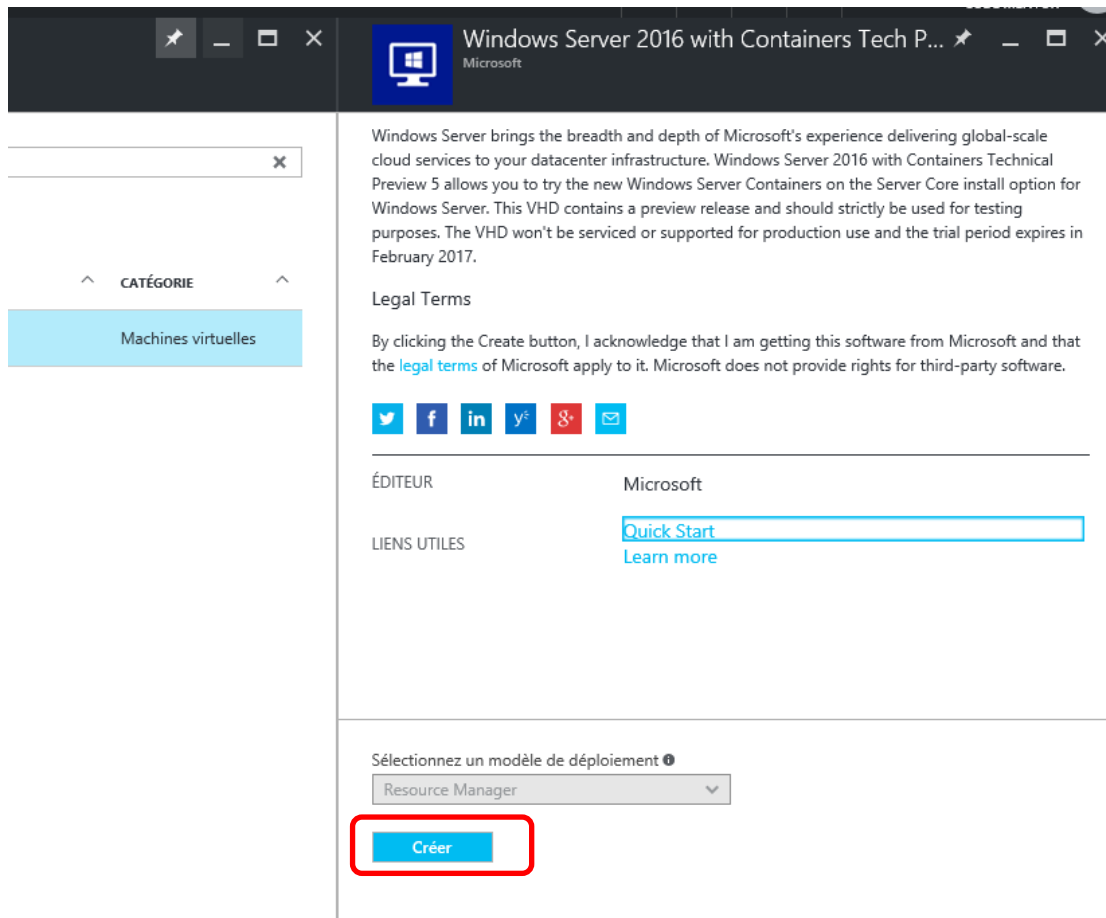
ÉDITEUR: Microsoft

LIENS UTILES: [Quick Start](#), [Learn more](#)

Sélectionnez un modèle de déploiement: Resource Manager

Créer





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

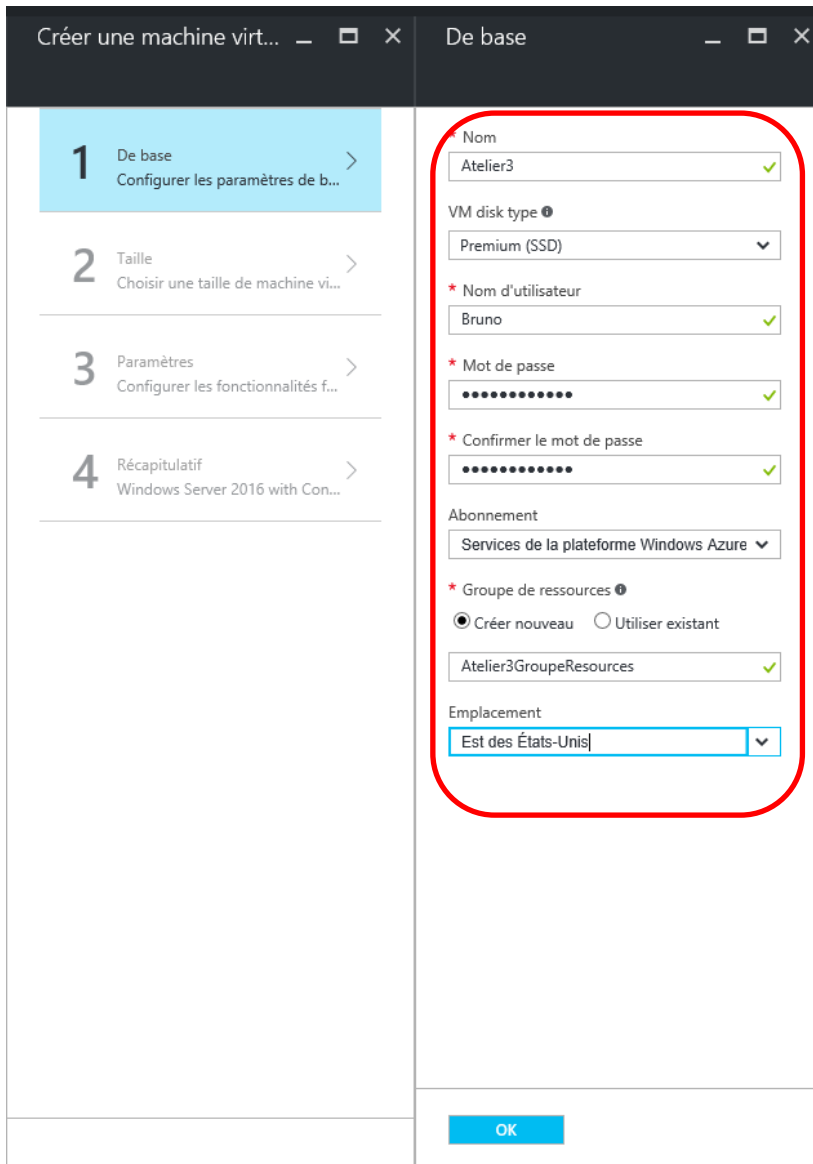


1 De base

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.



Créer une machine virt... — □ × De base — □ ×

1 De base
Configurer les paramètres de b...

2 Taille
Choisir une taille de machine vi...

3 Paramètres
Configurer les fonctionnalités f...

4 Récapitulatif
Windows Server 2016 with Con...

Nom
Atelier3 ✓

VM disk type ⓘ
Premium (SSD) ▼

* Nom d'utilisateur
Bruno ✓

* Mot de passe
..... ✓

* Confirmer le mot de passe
..... ✓

Abonnement
Services de la plateforme Windows Azure ▼

* Groupe de ressources ⓘ
☒ Créer nouveau ☐ Utiliser existant
Atelier3GroupeResources ✓

Emplacement
Est des États-Unis ▼

OK



2 Taille

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

- 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... >

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](#)

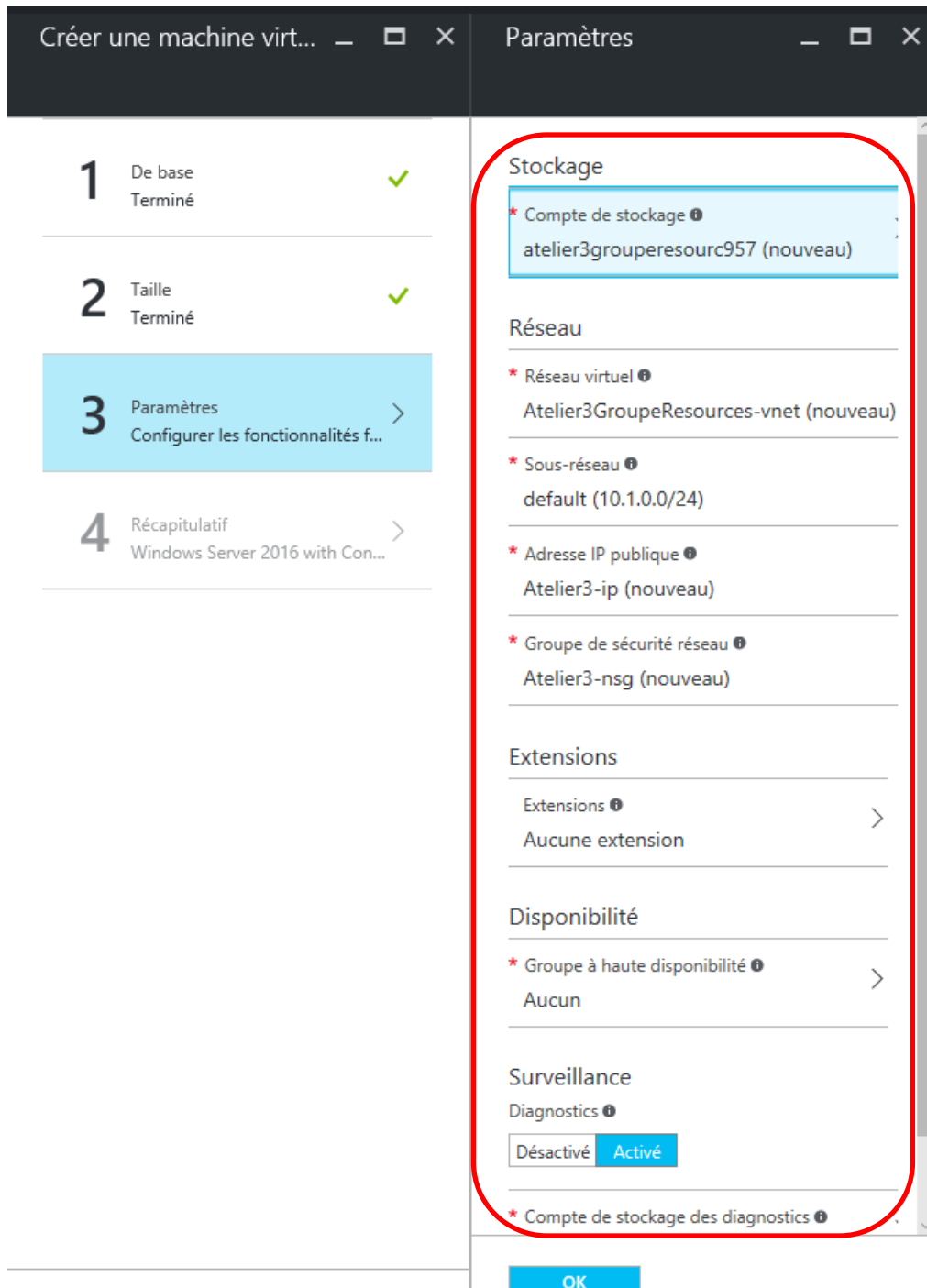
DS1_V2 Standard ★	DS2_V2 Standard ★	DS11_V2 Standard ★
1 Cœur	2 Cœurs	2 Cœurs
3.5 Go	7 Go	14 Go
2 Disques de données	4 Disques de données	4 Disques de données
3200 Opérations d'E/S par seco...	6400 Opérations d'E/S par seco...	6400 Opérations d'E/S par seco...
7 GB SSD local	14 GB SSD local	28 GB SSD local
Équilibrage de charge	Équilibrage de charge	Équilibrage de charge
Prise en charge de dis...	Prise en charge de dis...	Prise en charge de dis...
45,80 EUR/MOIS (ESTIMATION)	91,60 EUR/MOIS (ESTIMATION)	116,07 EUR/MOIS (ESTIMATION)

Sélectionner

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

3 Paramètres

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



Créer une machine virt... Paramètres

1 De base Terminé ✓

2 Taille Terminé ✓

3 Paramètres Configurer les fonctionnalités f... >

4 Récapitulatif Windows Server 2016 with Con... >

Stockage

* Compte de stockage ⓘ
atelier3grouperesourc957 (nouveau)

Réseau

* Réseau virtuel ⓘ
Atelier3GroupeResources-vnet (nouveau)

* Sous-réseau ⓘ
default (10.1.0.0/24)

* Adresse IP publique ⓘ
Atelier3-ip (nouveau)

* Groupe de sécurité réseau ⓘ
Atelier3-nsg (nouveau)

Extensions

Extensions ⓘ
Aucune extension >

Disponibilité

* Groupe à haute disponibilité ⓘ
Aucun >

Surveillance

Diagnostics ⓘ
Désactivé Activé

* Compte de stockage des diagnostics ⓘ

OK

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.

Cliquer 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.

The screenshot displays the Azure portal interface for configuring a Network Security Group (NSG). The left sidebar shows the configuration steps: 1. De base (Terminé), 2. Taille (Terminé), 3. Paramètres (Configurer les fonctionnalités f...), and 4. Récapitulatif (Windows Server 2016 with Cor...). The main area shows the 'Groupe de sécurité réseau' (Network Security Group) configuration. The 'Groupe de sécurité réseau' (Network Security Group) is selected, and the 'Ajouter une règle de trafic entrant' (Add an inbound rule) button is highlighted. The rule configuration is shown on the right, with fields for Name, Priority, Source, Service, Protocol, Port range, and Action.

Field	Value
Nom	Atelier3-nsg
Règles de trafic entrant	1000 default allow-ndp
Règles de trafic sortant	Aucun résultat
Nom	WEB
Priority	1200
Source	Any
Service	Personnalisé
Protocol	TCP
Port range	80
Action	Refuser



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

*** Nom**

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**

*** Priorité**

*** Source**

Any CIDR block Tag

Service

Personnalisé

*** Protocole**

Any TCP UDP

*** Port range**

*** 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

2 Taille Terminé

3 Paramètres Configurer les fonctionnalités f...

4 Récapitulatif Windows Server 2016 with Con...

Stockage

* Compte de stockage atelier3groupeResourc957 (nouveau)

Réseau

* Réseau virtuel Atelier3GroupeResourc-vinet (nouveau)

* Sous-réseau default (10.1.0.0/24)

* Adresse IP publique Atelier3-ip (nouveau)

* Groupe de sécurité réseau Atelier3-nsg (nouveau)

Extensions

Extensions Aucune extension

Disponibilité

* Groupe à haute disponibilité Aucun

Surveillance

Diagnostics Désactivé Activer

[Compte de stockage des diagnostics](#)

Créer

Aucun

*** Nom**

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**

*** Priorité**

*** Source**

Any CIDR block Tag

Service

Personnalisé

*** Protocole**

Any TCP UDP

*** Port range**

*** Action**

Refuser **Autoriser**

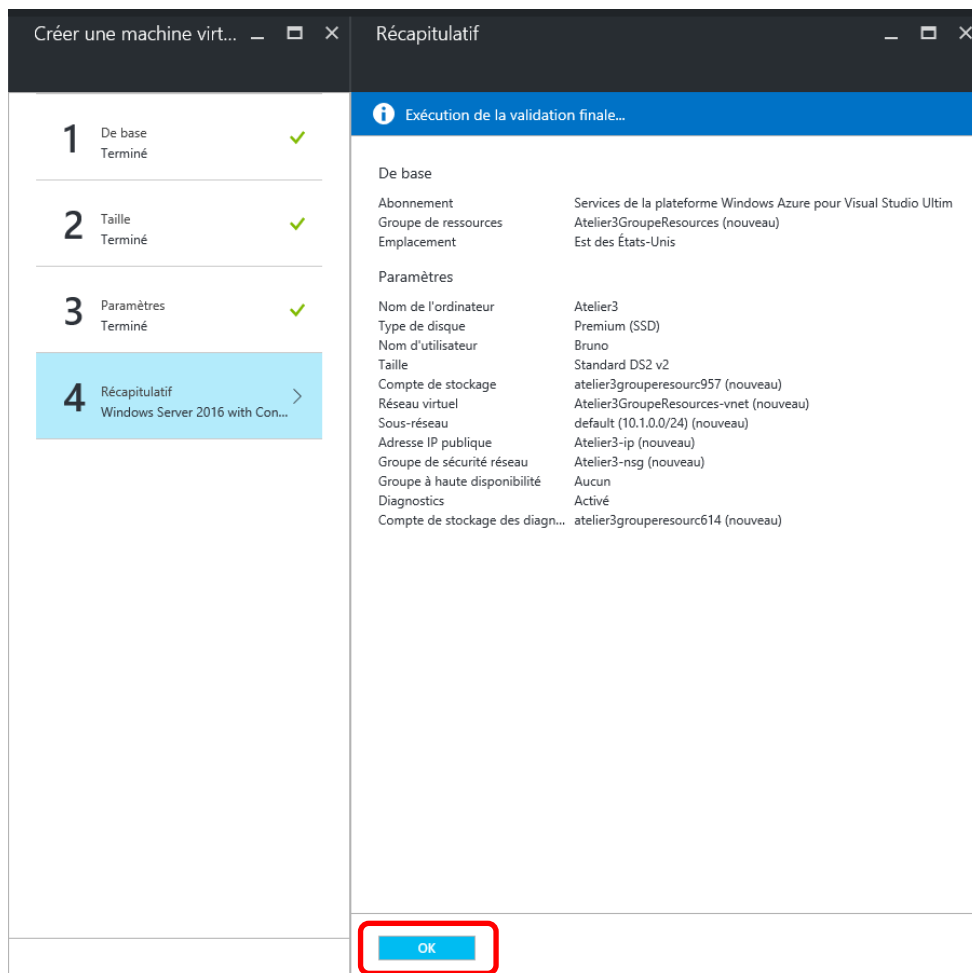
OK

OK

OK

4 Récapitulatif

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



Créer une machine virt... Récapitulatif

1 De base Terminé ✓

2 Taille Terminé ✓

3 Paramètres Terminé ✓

4 Récapitulatif Windows Server 2016 with Con... >

Exécution de la validation finale...

De base

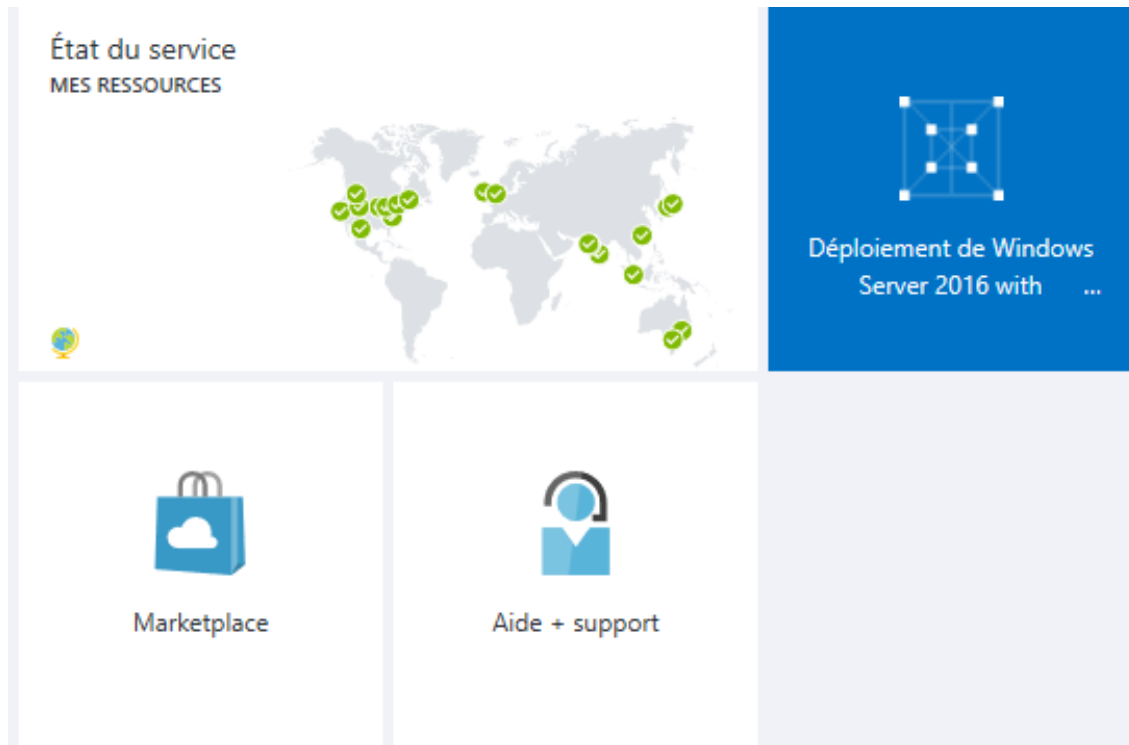
Abonnement	Services de la plateforme Windows Azure pour Visual Studio Ultim
Groupe de ressources	Atelier3GroupeResources (nouveau)
Emplacement	Est des États-Unis

Paramètres

Nom de l'ordinateur	Atelier3
Type de disque	Premium (SSD)
Nom d'utilisateur	Bruno
Taille	Standard DS2 v2
Compte de stockage	atelier3grouperesourc957 (nouveau)
Réseau virtuel	Atelier3GroupeResources-vnet (nouveau)
Sous-réseau	default (10.1.0.0/24) (nouveau)
Adresse IP publique	Atelier3-ip (nouveau)
Groupe de sécurité réseau	Atelier3-nsg (nouveau)
Groupe à haute disponibilité	Aucun
Diagnostics	Activé
Compte de stockage des diagn...	atelier3grouperesourc614 (nouveau)

OK

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



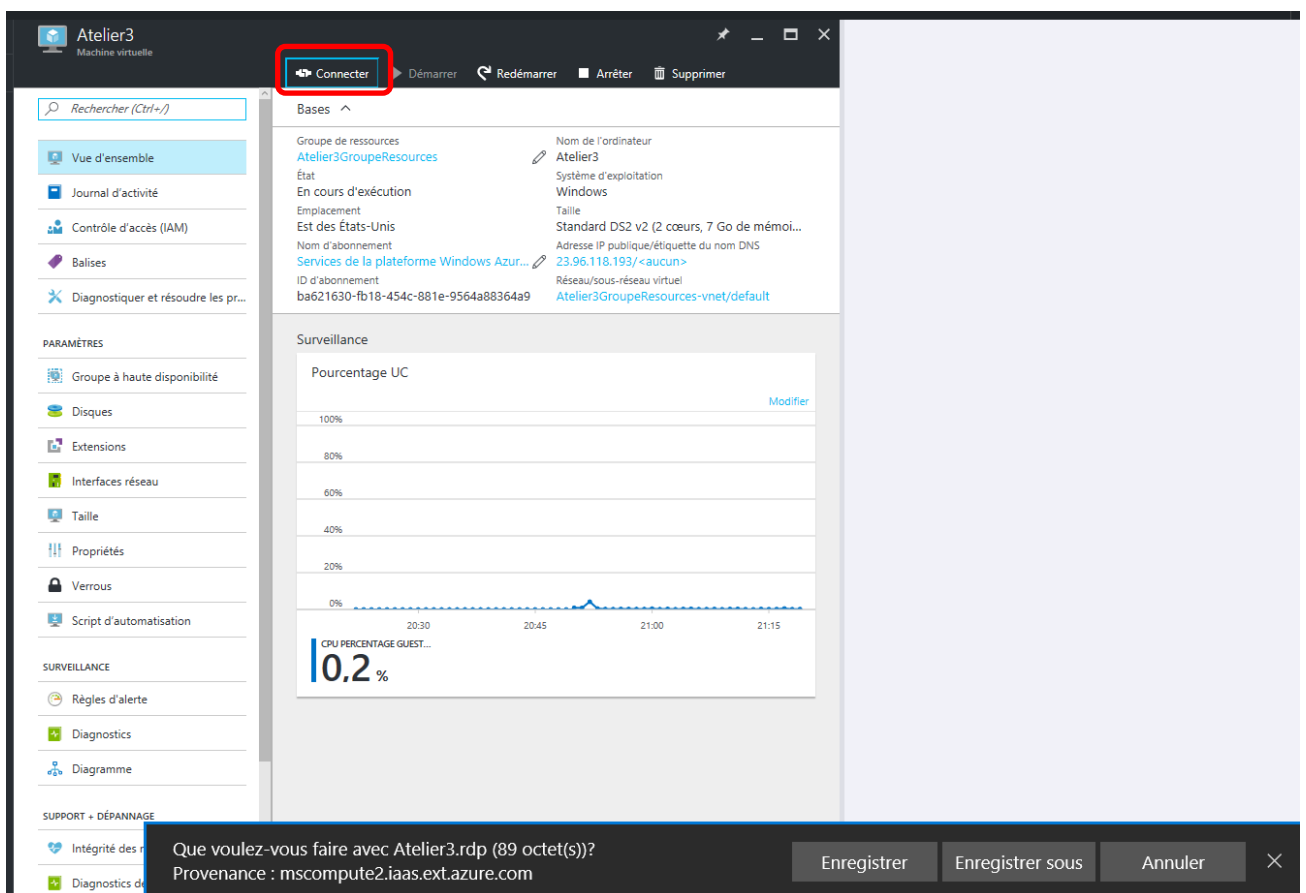
Après environ 15 minutes, notre VM sera prête à être utilisée.

Acte 2

Découverte de docker via les Windows Containers

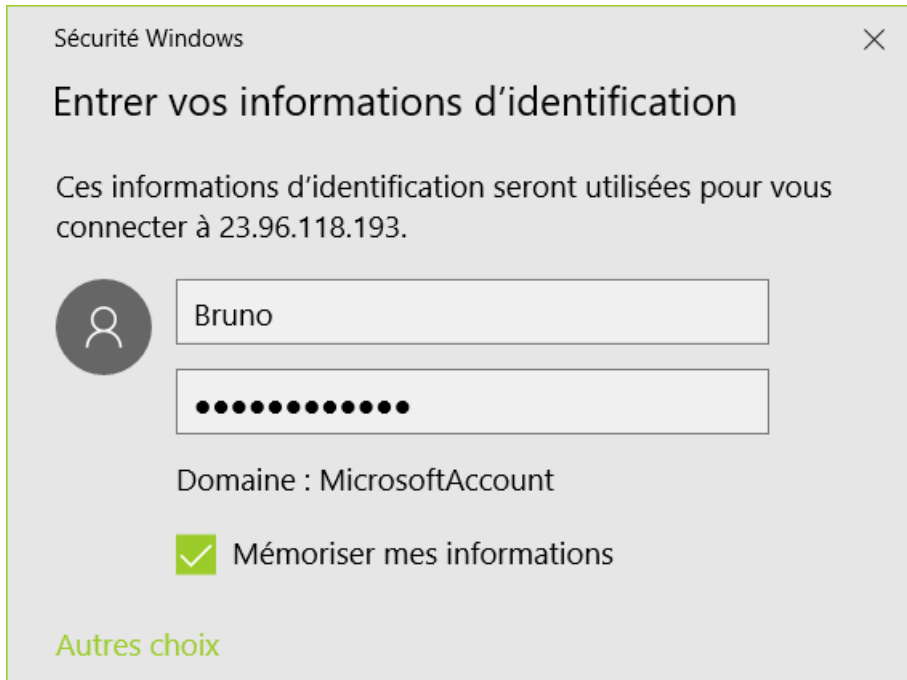
Connexion à la VM

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 »



The screenshot shows the Azure portal interface for a virtual machine named 'Atelier3'. The 'Connect' button is highlighted with a red rectangle. The interface includes a sidebar with navigation options like 'Vue d'ensemble', 'Journal d'activité', and 'Contrôle d'accès (IAM)'. The main area displays the VM's status as 'En cours d'exécution' and provides details such as the operating system (Windows), size (Standard DS2 v2), and public IP address (23.96.118.193). A 'Surveillance' section shows a 'Pourcentage UC' graph with a current value of 0.2%.


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



Sécurité Windows

Entrer vos informations d'identification

Ces informations d'identification seront utilisées pour vous connecter à 23.96.118.193.

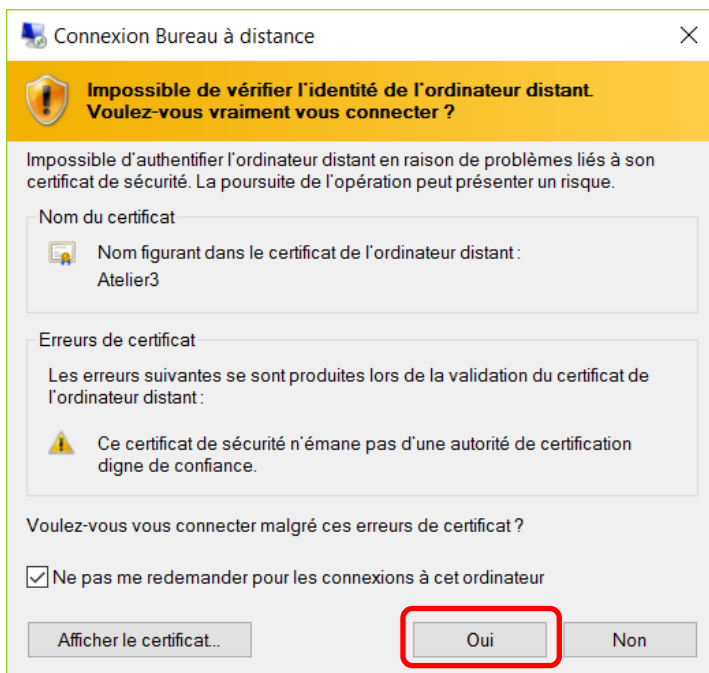


Domaine : MicrosoftAccount

☒ Mémoriser mes informations

[Autres choix](#)

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




Connexion Bureau à distance

Impossible de vérifier l'identité de l'ordinateur distant. Voulez-vous vraiment vous connecter ?


Impossible d'authentifier l'ordinateur distant en raison de problèmes liés à son certificat de sécurité. La poursuite de l'opération peut présenter un risque.

Nom du certificat

 Nom figurant dans le certificat de l'ordinateur distant :
Atelier3

Erreurs de certificat

Les erreurs suivantes se sont produites lors de la validation du certificat de l'ordinateur distant :

 Ce certificat de sécurité n'émane pas d'une autorité de certification digne de confiance.

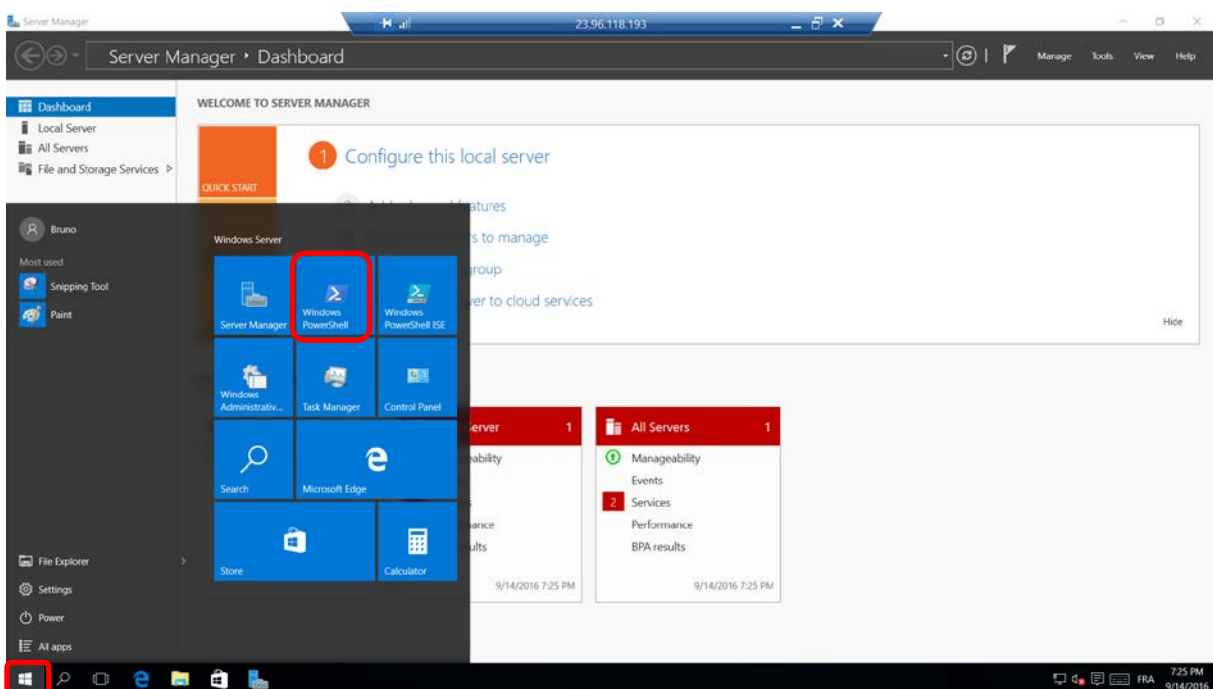
Voulez-vous vous connecter malgré ces erreurs de certificat ?

☒ Ne pas me redemander pour les connexions à cet ordinateur

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

```
Administrator: Windows PowerShell
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 :

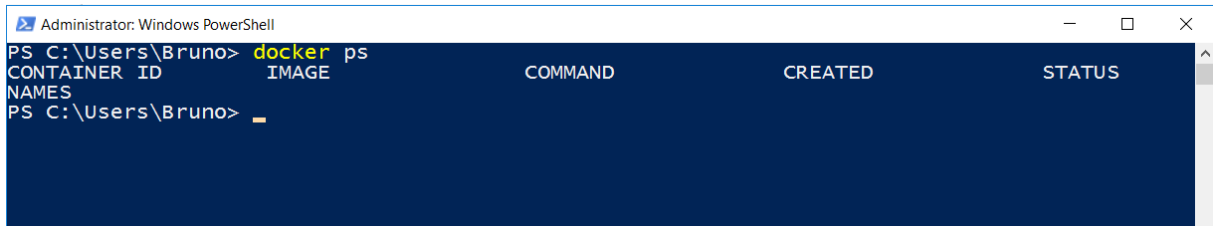
docker version

```
Administrator: Windows PowerShell
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

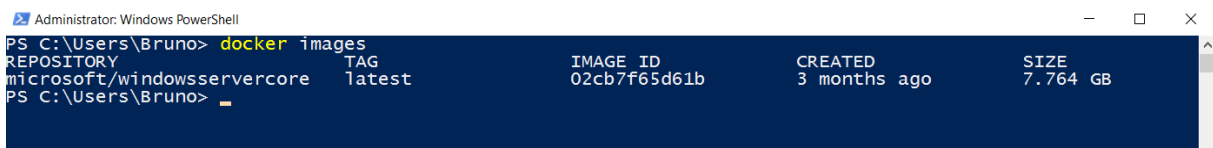


```
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
NAMES
```

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 :

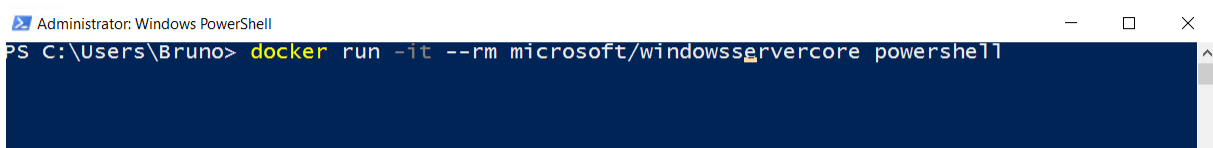
docker images



```
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
```

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

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   :

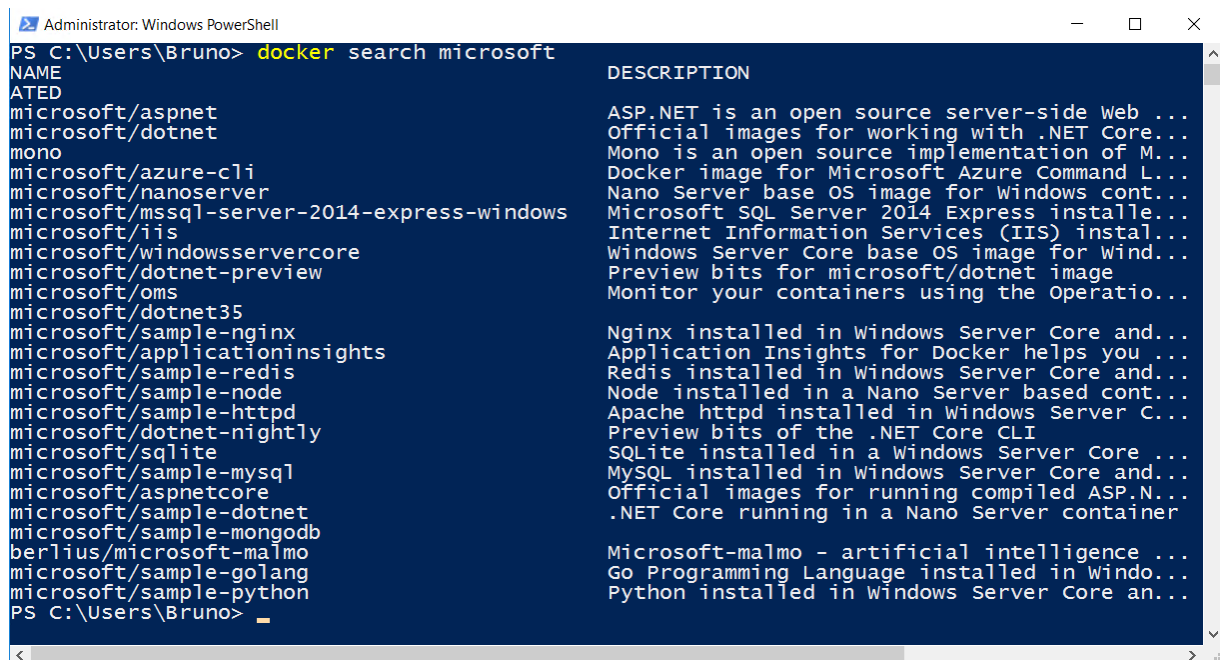
PS C:\Users\Bruno>
```

Ce premier contact vous permet de comprendre qu'un conteneur est bien plus restreint qu'un système complet.



Si votre 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`

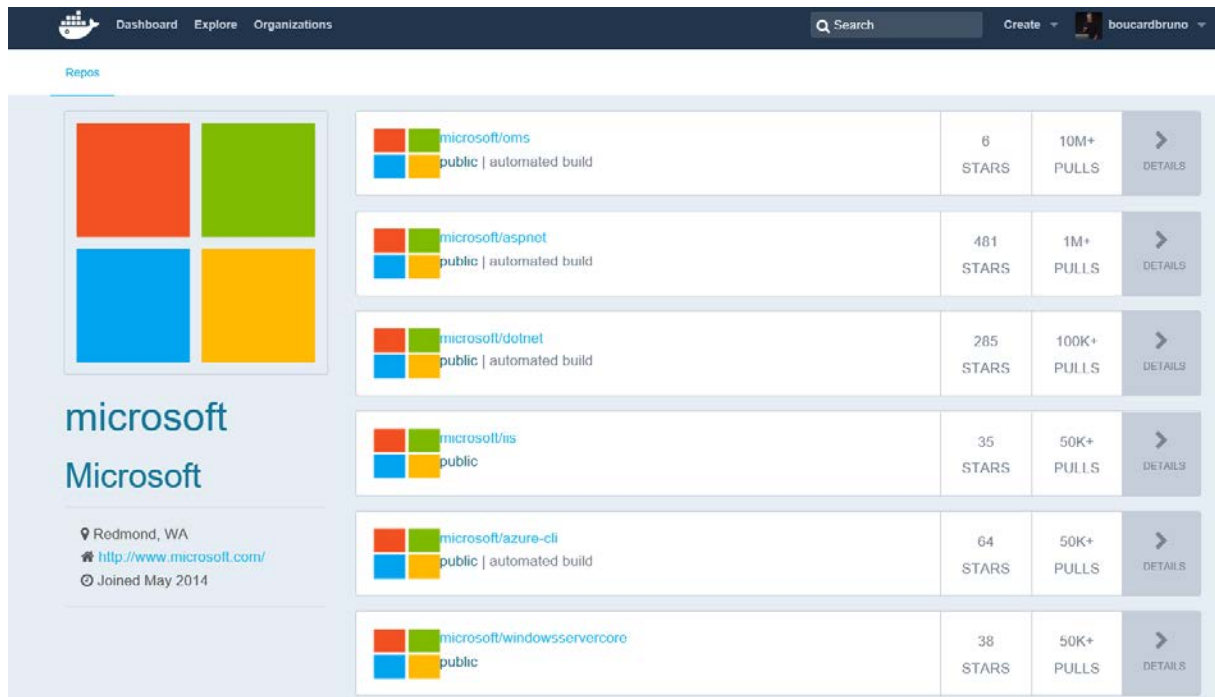


NAME	DESCRIPTION
microsoft/aspnet	ASP.NET is an open source server-side Web ...
microsoft/dotnet	Official images for working with .NET Core...
mono	Mono is an open source implementation of M...
microsoft/azure-cli	Docker image for Microsoft Azure Command L...
microsoft/nanoserver	Nano Server base OS 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	
microsoft/sample-nginx	Nginx installed in windows Server Core and...
microsoft/applicationinsights	Application Insights for Docker helps you ...
microsoft/sample-redis	Redis installed in Windows Server Core and...
microsoft/sample-node	Node installed in a Nano Server based cont...
microsoft/sample-httpd	Apache httpd installed in Windows Server C...
microsoft/dotnet-nightly	Preview bits of the .NET Core CLI
microsoft/sqlite	SQLite installed in a Windows Server Core ...
microsoft/sample-mysql	MySQL installed in Windows Server Core and...
microsoft/aspnetcore	Official images for running compiled ASP.N...
microsoft/sample-dotnet	.NET Core running in a Nano Server container
berlious/microsoft-malmo	Microsoft-malmo - artificial intelligence ...
microsoft/sample-golang	Go Programming Language installed in Windo...
microsoft/sample-python	Python installed in Windows Server Core an...

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/>





Repository	Stars	Pulls	Details
microsoft/oms	6	10M+	DETAILS
microsoft/aspnet	481	1M+	DETAILS
microsoft/dotnet	285	100K+	DETAILS
microsoft/iis	35	50K+	DETAILS
microsoft/azure-cli	64	50K+	DETAILS
microsoft/windowsservercore	38	50K+	DETAILS

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.

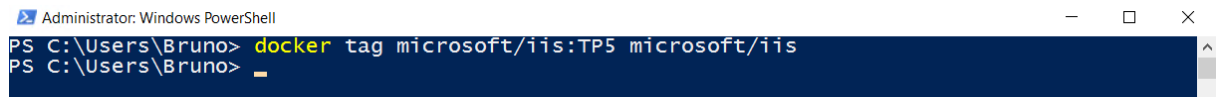
```
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker pull microsoft/iis:TP5
TP5: Pulling from microsoft/iis
1239394e5a8a: Already exists
847199668046: Pull complete
4b1361d2706f: Pull complete
Digest: sha256:1d64cc22fbc56abc96e4b7df1b51e6f91b0da1941aa155f545f14dd76ac522fc
Status: Downloaded 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

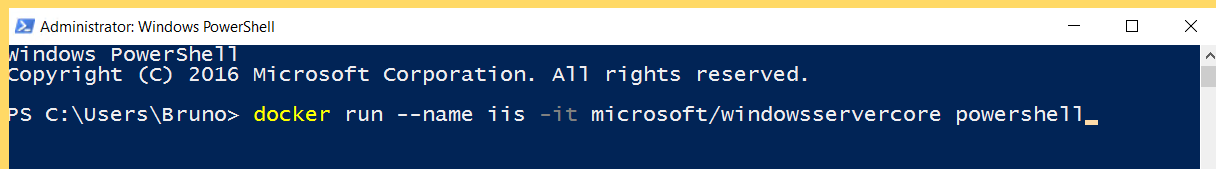


```
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker tag microsoft/iis:TP5 microsoft/iis
PS C:\Users\Bruno> _
```

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.

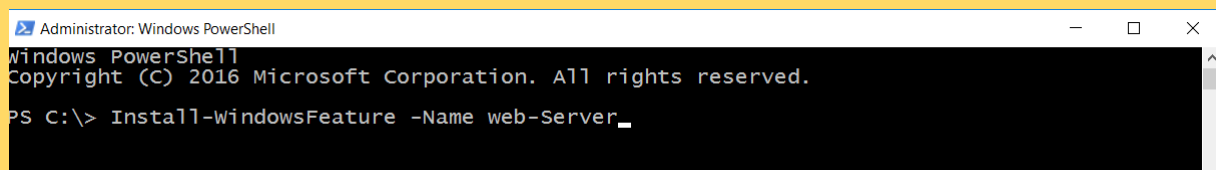
docker run --name iis -it microsoft/windowsservercore powershell



```
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_
```

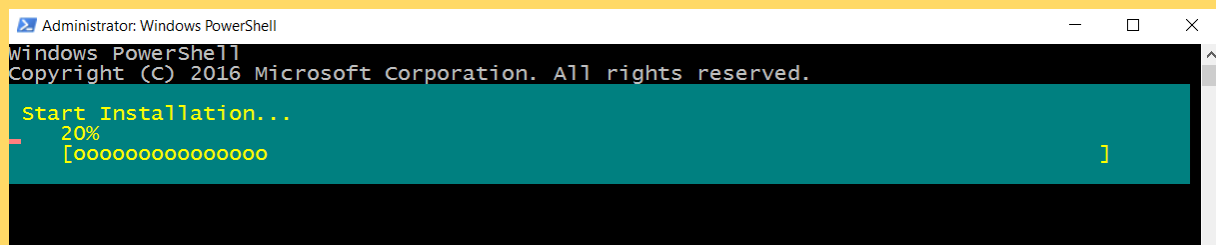
Puis une fois dans le conteneur taper la commande:

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



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

Start Installation...
20%
[ooooooooooooooooooooo]
```



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

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

Success Restart Needed Exit Code      Feature Result
-----
True     No                Success      {Common HTTP Features, Default Document, D...

PS C:\> _
```

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

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

PS C:\Users\Bruno> docker commit iis microsoft/iis_
```

Après quelques secondes, une nouvelle image est engendrée.

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

PS C:\Users\Bruno> docker commit iis microsoft/iis
sha256:ef0f73c037923c70abb9861f543567ffe7d722a86ee371236b58ac3bdaa1bdd4
PS C:\Users\Bruno> _
```

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 de 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).

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

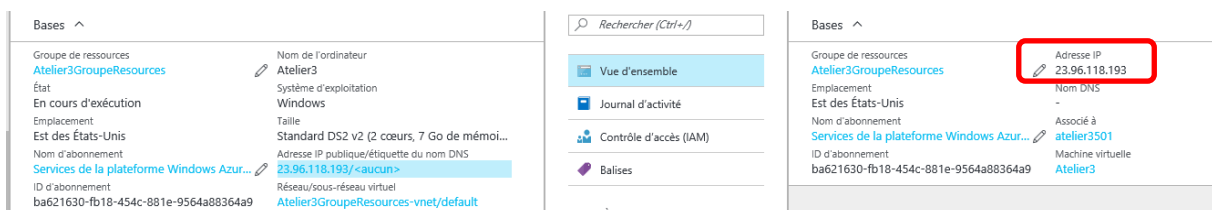
PS C:\Users\Bruno> docker run -it -p 80:80 microsoft/iis cmd_
```

Après quelques secondes, vous êtes dans le conteneur.

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

C:\>
```

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



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



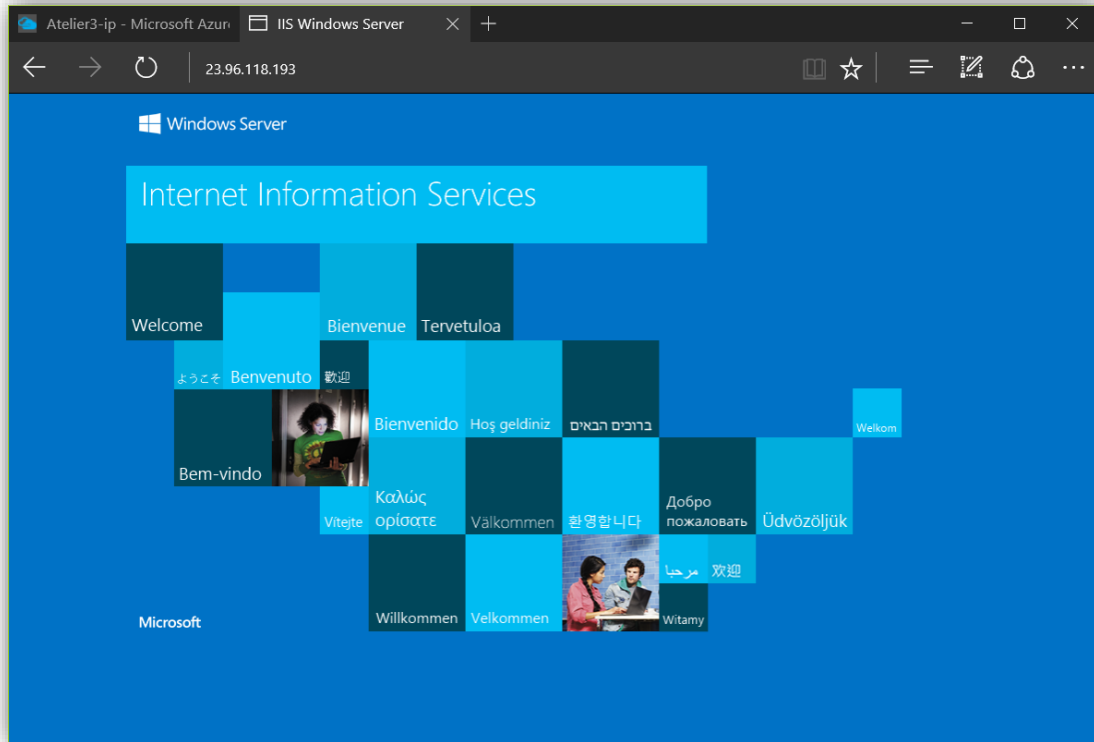
Se libérer par ses compétences

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

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



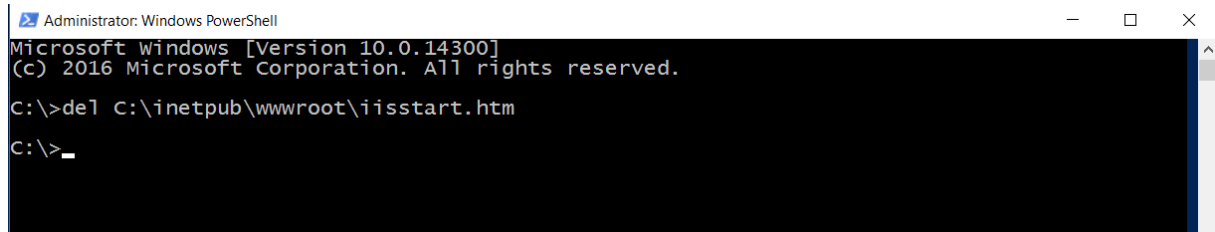
Depuis votre poste travail, taper l'IP dans un browser.



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

Retournez dans votre conteneur, puis taper la commande:

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

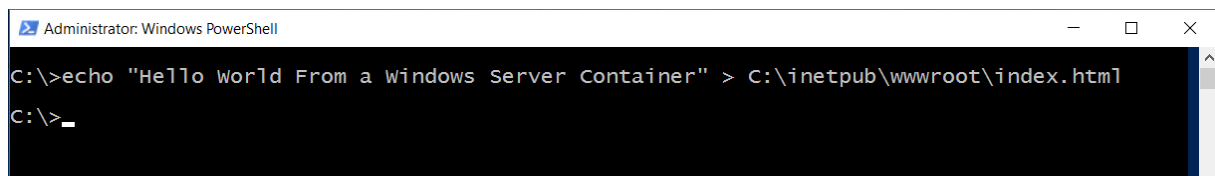


```
Administrator: Windows PowerShell
Microsoft Windows [Version 10.0.14300]
(c) 2016 Microsoft Corporation. All rights reserved.

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

Puis la commande:

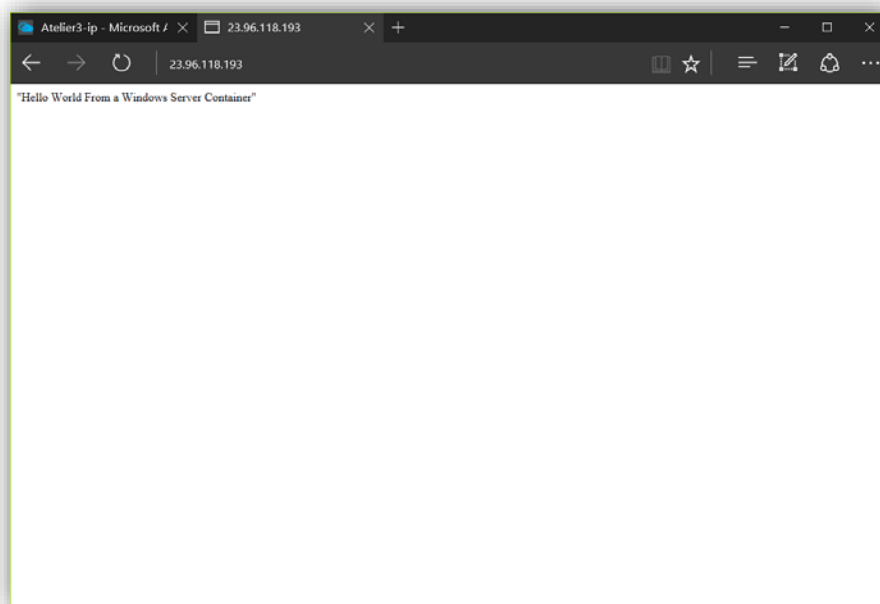
```
echo "Hello World From a Windows Server Container"
> C:\inetpub\wwwroot\index.html
```



```
Administrator: Windows PowerShell

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

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 conteurs 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

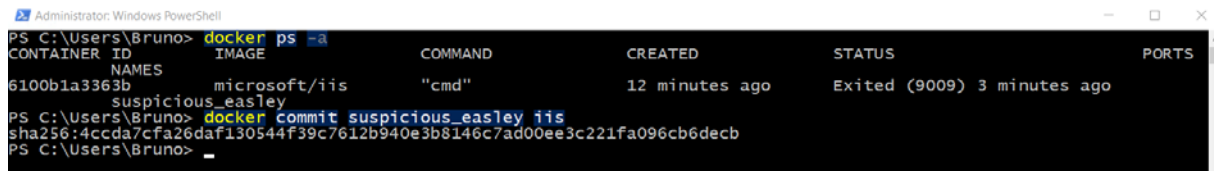
```
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
NAMES
6100b1a3363b       microsoft/iis       "cmd"              12 minutes ago     Exited (9009) 3 minutes ago
suspicious_easley
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. Si vous souhaitez modifier une image, vous devez en créer une nouvelle. Lancer la commande :

```
docker commit suspicious_easley iis
```



CONTAINER ID	NAMES	IMAGE	COMMAND	CREATED	STATUS	PORTS
6100b1a3363b	suspicious_easley	microsoft/iis	"cmd"	12 minutes ago	Exited (9009) 3 minutes ago	

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.

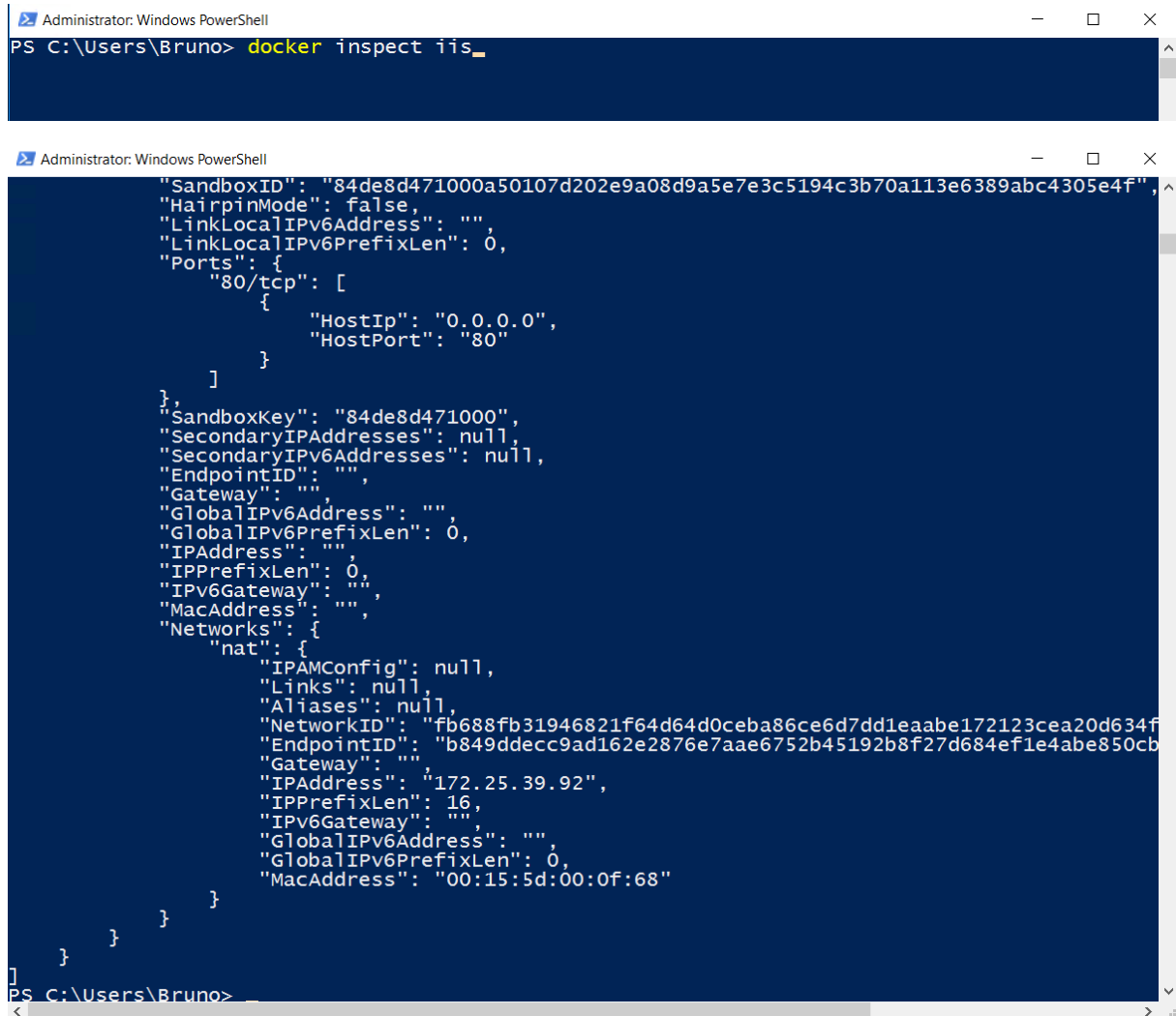
Nous pouvons donc démarrer un nouveau conteneur sur la base de cette nouvelle image. Lancez la commande :

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



Une fois dans votre conteneur. Lancer une autre fenêtre powershell

`docker inspect iis`



```

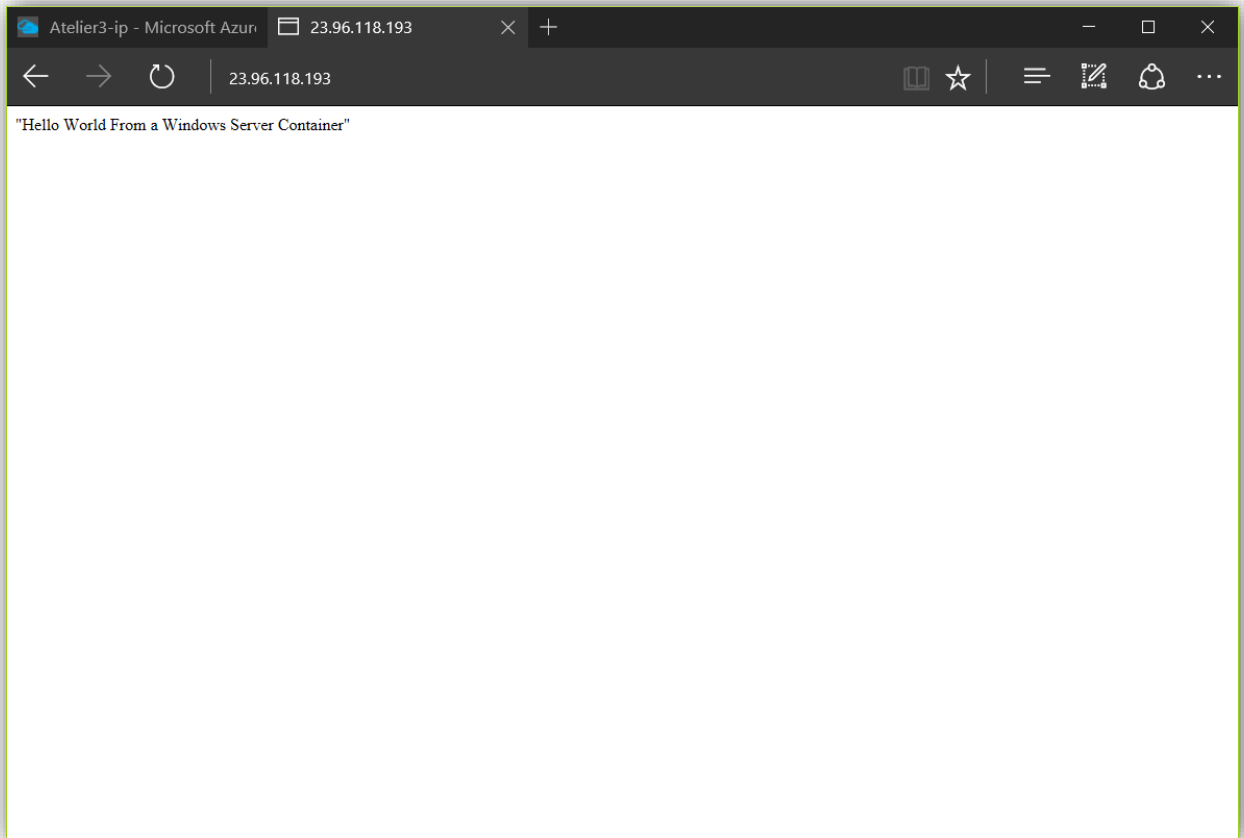
Administrator: Windows PowerShell
PS C:\Users\Bruno> docker inspect iis

Administrator: Windows PowerShell
{
  "SandboxID": "84de8d471000a50107d202e9a08d9a5e7e3c5194c3b70a113e6389abc4305e4f",
  "HairpinMode": false,
  "LinkLocalIPv6Address": "",
  "LinkLocalIPv6PrefixLen": 0,
  "Ports": {
    "80/tcp": [
      {
        "HostIp": "0.0.0.0",
        "HostPort": "80"
      }
    ]
  },
  "SandboxKey": "84de8d471000",
  "SecondaryIPAddresses": null,
  "SecondaryIPv6Addresses": null,
  "EndpointID": "",
  "Gateway": "",
  "GlobalIPv6Address": "",
  "GlobalIPv6PrefixLen": 0,
  "IPAddress": "",
  "IPPrefixLen": 0,
  "IPv6Gateway": "",
  "MacAddress": "",
  "Networks": {
    "nat": {
      "IPAMConfig": null,
      "Links": null,
      "Aliases": null,
      "NetworkID": "fb688fb31946821f64d64d0ceba86ce6d7dd1eaabe172123cea20d634f",
      "EndpointID": "b849ddecc9ad162e2876e7aae6752b45192b8f27d684ef1e4abe850cb",
      "Gateway": "",
      "IPAddress": "172.25.39.92",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "MacAddress": "00:15:5d:00:0f:68"
    }
  }
}
PS C:\Users\Bruno>

```

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

Depuis votre poste travail, rafraîchissez la page.



Vous devriez retrouver une page web similaire.

Puis vous sortez du conteneur et lancer les commandes similaires :

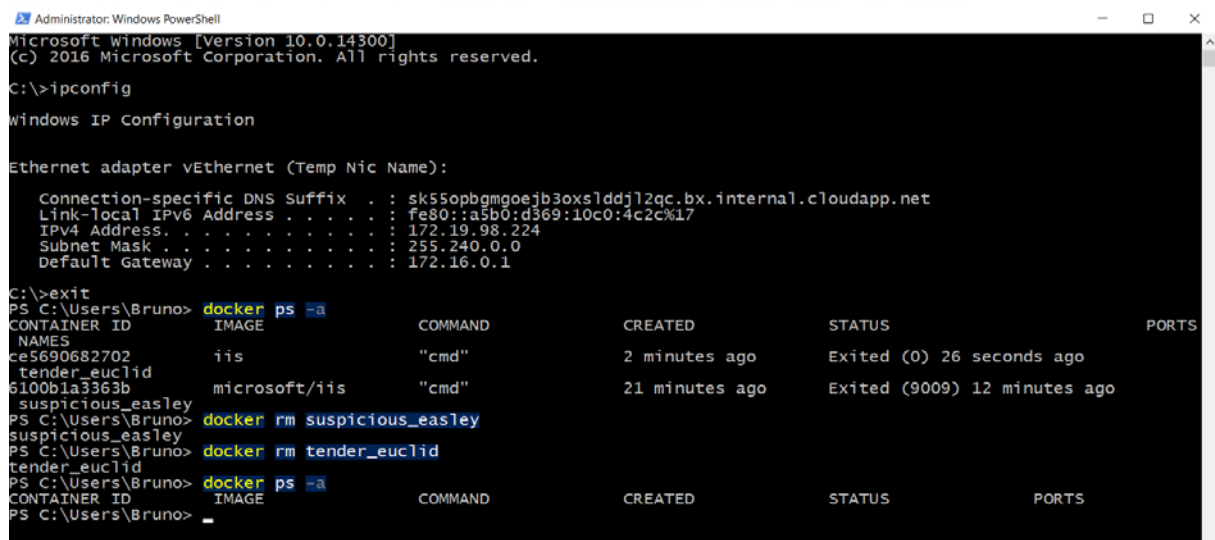
`exit`

`docker ps -a`

`docker rm id_conteneur1`

`docker rm id_conteneur2`

`docker ps -a`



```

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

C:\>ipconfig

Windows IP Configuration

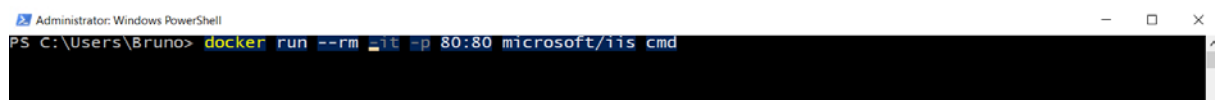
Ethernet adapter vEthernet (Temp Nic Name):

    Connection-specific DNS Suffix  . : sk55opbgmgoejb3oxs1ddj12qc.bx.internal.cloudapp.net
    Link-local IPv6 Address . . . . . : fe80::a5b0:d369:10c0:4c2c%17
    IPv4 Address. . . . . : 172.19.98.224
    Subnet Mask . . . . . : 255.240.0.0
    Default Gateway . . . . . : 172.16.0.1

C:\>exit
PS C:\Users\Bruno> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
ce5690682702      iis                 "cmd"              2 minutes ago       Exited (0) 26 seconds ago
tender_euclid      microsoft/iis       "cmd"              21 minutes ago      Exited (9009) 12 minutes ago
6100b1a3363b
suspicious_easley
PS C:\Users\Bruno> docker rm suspicious_easley
suspicious_easley
PS C:\Users\Bruno> docker rm tender_euclid
tender_euclid
PS C:\Users\Bruno> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
PS C:\Users\Bruno>
  
```

La commande `docker rm id`, 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`



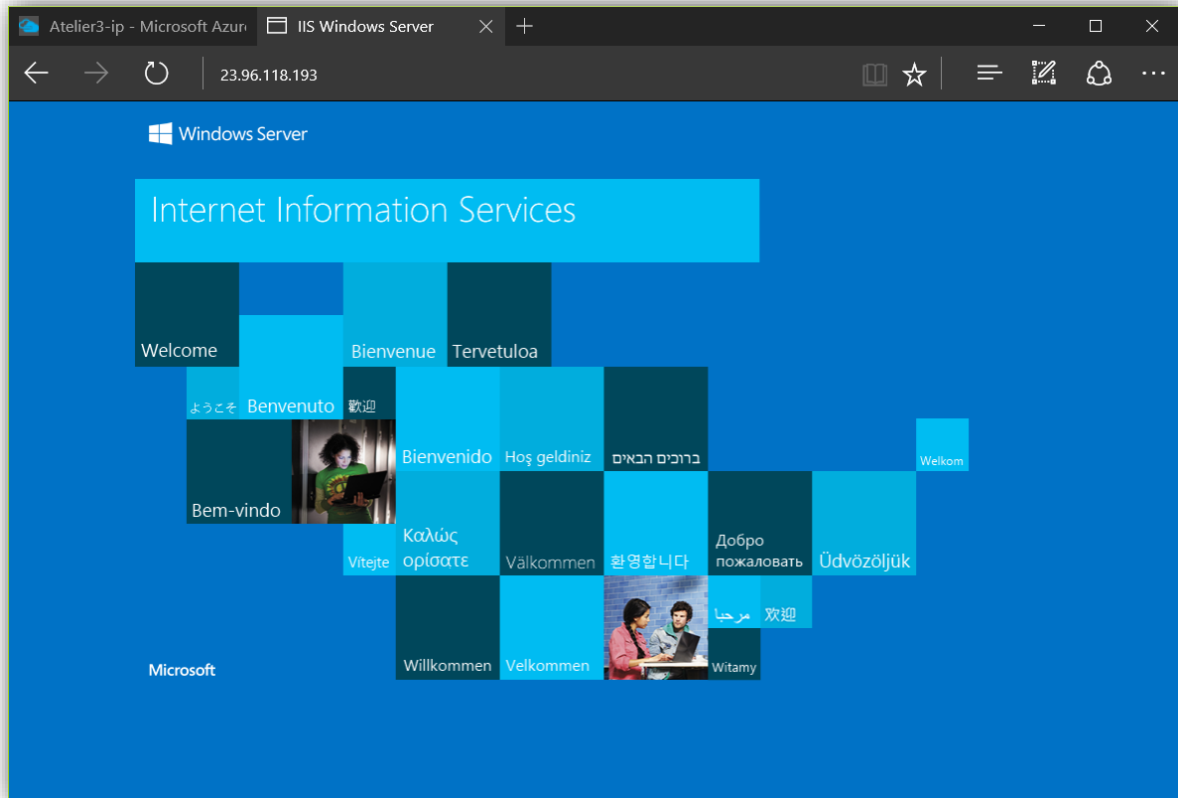
```

Administrator: Windows PowerShell
PS C:\Users\Bruno> 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

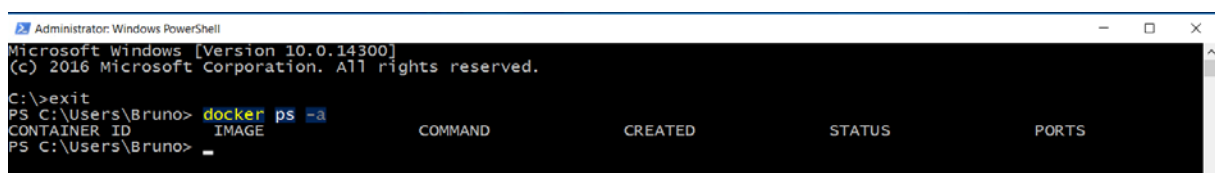


On retrouve la page d'accueil du serveur IIS.

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

`exit`

`docker ps -a`



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

Automatiser avec un fichier dockerfile

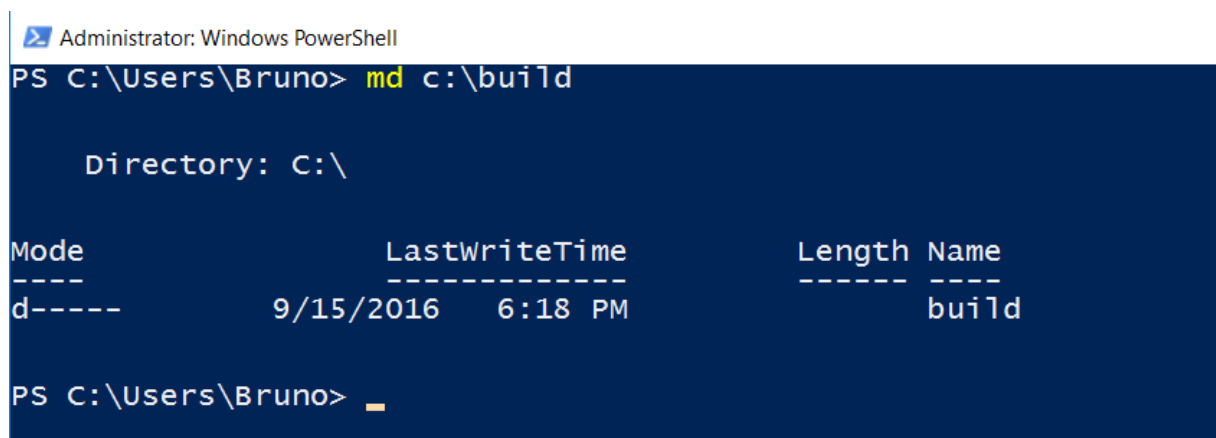
Dans l'activité précédente, nous avons principalement frappé des commandes à la main. C'est une excellente méthode, lorsque nous souhaitons découvrir les bases de docker.

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         Length Name
----                -
-a----             9/15/2016   6:21 PM             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 ebfb689a392
---> cbf6b9cc9db6
Removing intermediate container ebfb689a392
Successfully built cbf6b9cc9db6
PS C:\build> _
```

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.

Pour maintenir le conteneur en vie, nous lançons la commande ping en locale dans le conteneur.

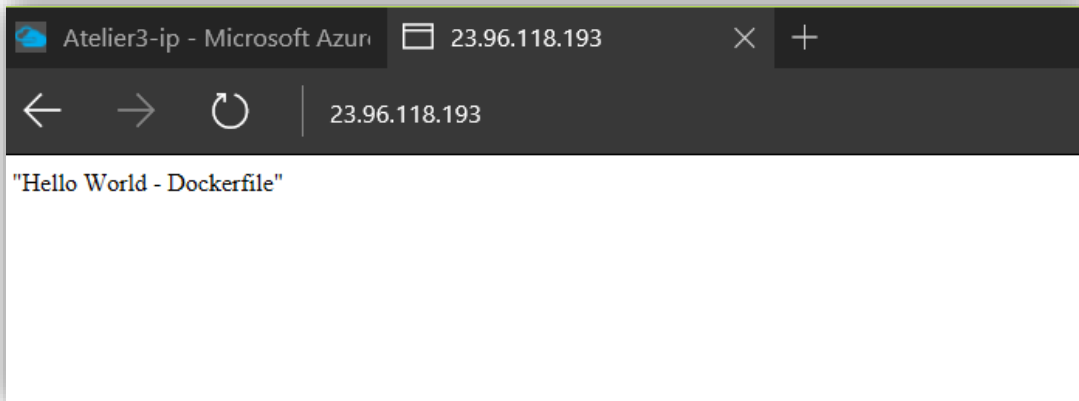
Lancer la commande :

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

```
Administrator: Windows PowerShell
PS C:\build> docker run -d -p 80:80 iis-dockerfile ping -t localhost
cd6f6d13177e4d4ed197e235e3517541c60a29589dfac30fdfa5fc6e1fb3dd75
PS C:\build> _
```



Depuis votre poste travail, rafraîchissez la page



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
```

Nous faisons une dernière commande :

```
docker ps -a
```

Afin de confirmer qu'il n'y a plus trace de conteneur sur notre machine.

```
Administrator: Windows PowerShell
PS C:\build> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
NAMES
cd6f6d13177e       iis-dockerfile     "ping -t localhost" 23 minutes ago
elated_meninsky
PS C:\build> docker rm elated_meninsky -f
elated_meninsky
PS C:\build> docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
NAMES
PS C:\build> _
```



Durant cet atelier, nous avons créé trois nouvelles images : iis-dockerfile, iis, microsoft/iis. Lancez la commande :

docker images

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

PS C:\Users\Bruno> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
iis-dockerfile       latest             41b30d697ce6       34 seconds ago     7
iis                  latest             6fe5d1611de2       5 minutes ago      7
microsoft/iis        TP5                accd044753c1       7 weeks ago        7
microsoft/iis        latest             accd044753c1       7 weeks ago        7
microsoft/windowsservercore latest             02cb7f65d61b       3 months ago       7
PS C:\Users\Bruno>
```

Nous allons les retirer de l'entrepôt local via la commande :

docker rmi <id|nom>

La commande rmi <id_image|nom>, permet de détruire une image dans votre entrepôt d'images locales.

docker rmi <id|nom> -f

Permet de forcer la destruction si besoin.

À ce stade, nous retrouvons l'entrepôt d'images dans l'état initial du début d'atelier.

docker images

```
Administrator: Windows PowerShell
PS C:\build> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
microsoft/windowsservercore latest             02cb7f65d61b       3 months ago       7
PS C:\build>
```

