

L'écosystème Docker pour les Dev .NET et les Ops Windows

Julien Chable
Consultant – MVP

@Jchable
julien.chable@ncit.nc

Blog : <http://julien.chable.net/>



Speaker

Julien Chable

Société : NC IT

Consultant & Développeur
MVP Office Server



Compétences :

- Pilotage Agile
- Développement (.NET / Java & co)
- Mobilité
- Office 365, Azure & SharePoint

Me contacter :

- @JChable
- Julien.chable@ncit.nc
- +687 95.61.57
- <http://julien.chable.net>



Agenda de la session

- Technologies et Produits disponibles
- Support de Docker dans Visual Studio 2017
- Débogage d'un conteneur avec Visual Studio
- Conteneur Windows : c'est quoi ?
- Migration d'une application Windows legacy vers un conteneur Windows

Introduction

- Les conteneurs existaient depuis longtemps avant Docker, Inc
- Docker a joué un rôle important dans la démocratisation de cette technologie

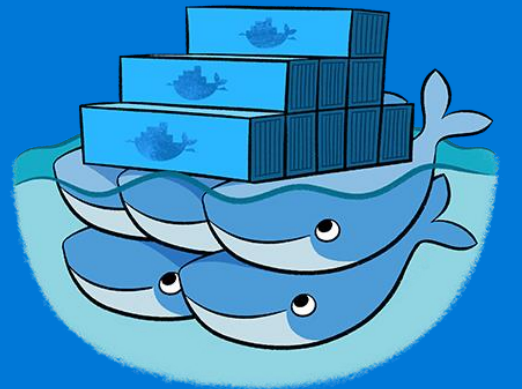
Comment ?

- Standardisation du format d'artefacts (image Docker/template, outils)
- Promesse aux Devs et Ops que l'on entendrait plus la phrase fatale : « Mais ça marche sur ma machine »

Introduction

- Promesse d'encapsuler une application et toutes les dépendances
- Pour cela il faut une standardisation du format pour assurer la portabilité tacite et totale (poste Dev, serveur, Cloud)
- Lorsque Microsoft a annoncé la création de conteneur dans le kernel de Windows Server 2016 ➔ il a annoncé la compatibilité avec le format et les outils Docker

Technologies et Produits Microsoft dans l'écosystème Docker



Technologies & Produits Docker chez MS

Systeme d'exploitation

- Windows Server 2016 : Qualification et production
- Windows 10 : Dev

Cloud

- Azure : support par de plus en plus de services
 - (Au-delà des serveurs clouds Ubuntu, CoreOS, Redhat, etc)
 - Orchestrateurs : Kubernetes, Docker Swarm, Mesosphere DC/OS ➔ pas de BEST tool, mais chacun supporte le format Docker et est disponible dans Azure

Azure Container Service

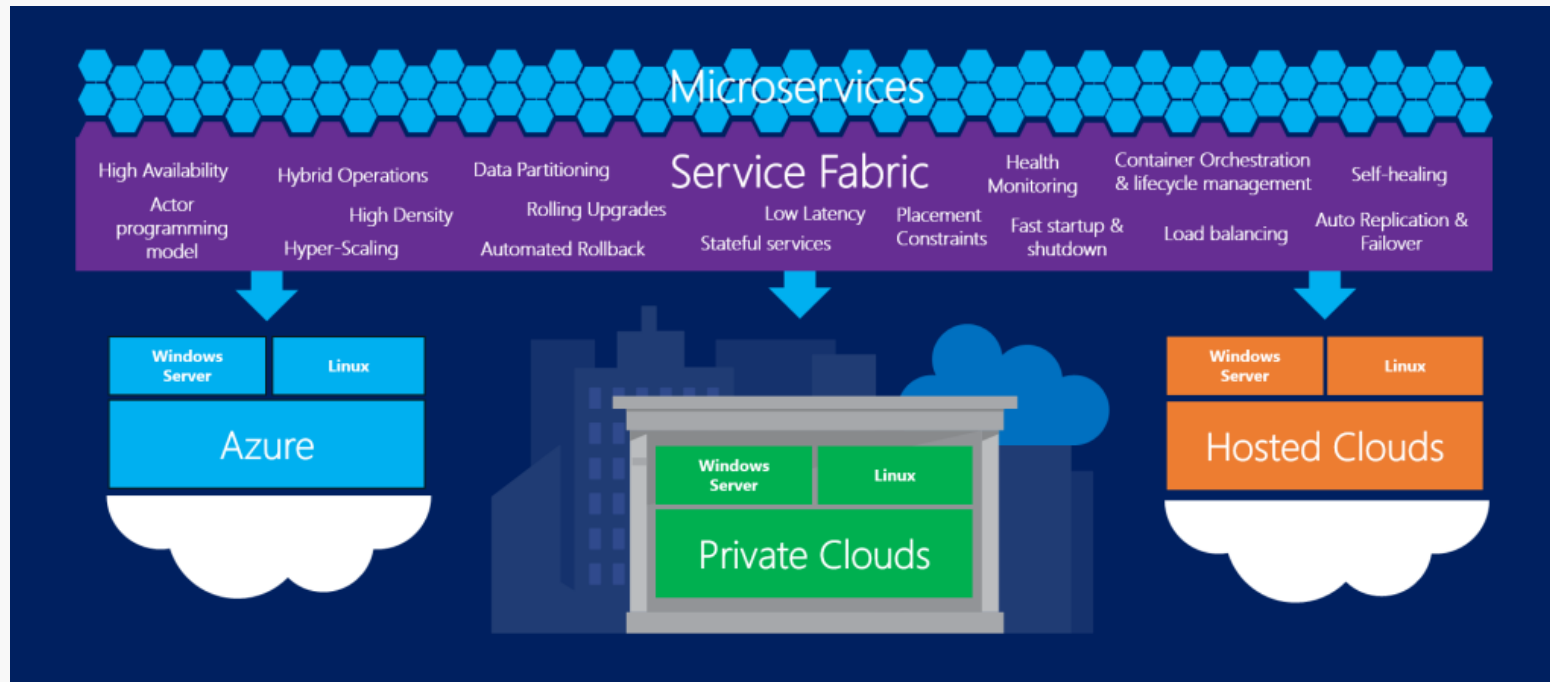
- Service d'infrastructure de standardisation de déploiement d'orchestrateurs de conteneurs Docker (version OSS non modifiée) :
 - Mesosphere DC/OS
 - Docker Swarm
 - Kubernetes
- Basé sur *Azure Resource Manager* qui se charge de déployer l'infra (réseau, VM, stockage, ...) et de configurer l'orchestrateur
- Utilisation des orchestrateurs comme On Premise (API REST, commande Docker, kubectl)
- Elasticité en augmentant les nœuds de cluster (portail Azure ou Azure CLI 2.0)

Azure Apps Service

- Platform as a service qui gère pour vous l'infrastructure
 - Echelle / Elasticité en auto-scaling (horizontal et vertical), slots de déploiement
 - Automatisation du déploiement depuis Github, Visual Studio team Services, etc
- 5 services
 - **Web Apps** : app web .NET, PHP, Java, Python, HTML/JS
 - **Mobile Apps** : basé sur web apps, déploiement de backoffice mobile (notification push, intégration AD, auto scaling, etc)
 - **API Apps** : authentication Azure AD, Oauth, CORS, Swagger
 - **Logic Apps** : applications de workflow / solution d'intégration des processus métier
 - **Azure Functions** : exécution de code « serverless »
- Depuis peu, ces services s'exécute sous Linux (sous format d'image et de conteneur Docker)

Azure Service Fabric

- Orchestrateur maison de MS pour certains services et apps micro-service sous forme de « guest executable »
- Supporte les conteneurs Docker, les nœuds Windows et Linux
- Plateforme dans Azure mais également Cloud privé et On Premises (mode standalone – Windows 2012 R2 ou 2016)



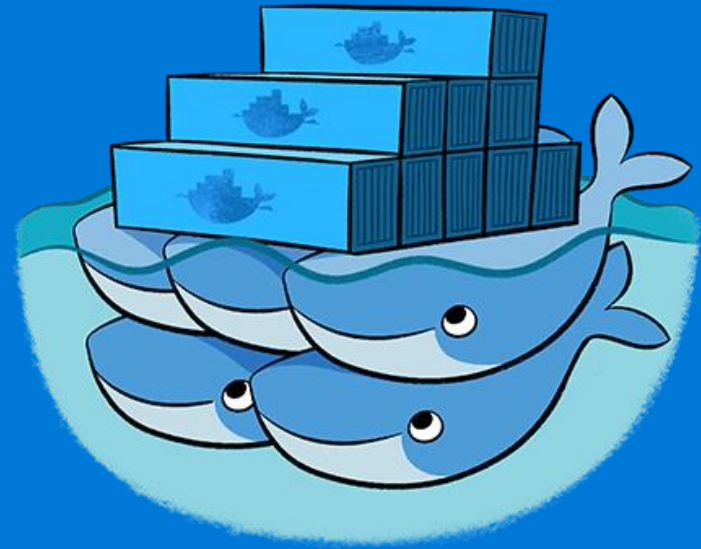
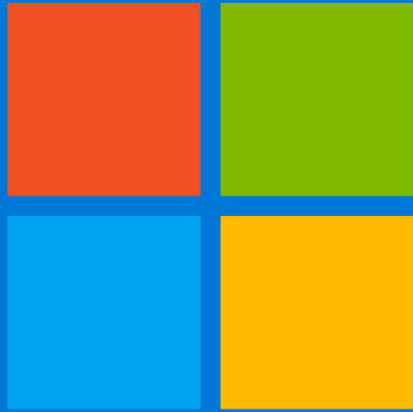
Azure Container Registry

- Implémentation « as a service » de la registry Docker ➔ Registry privée
- Utilise le registry Docker open source donc les opérations LOGIN, PUSH/PULL sont prises en compte
- Service gratuit, seul le stockage des images et la bande passante sont facturés
- Authentification possible avec Azure AD

Azure Market Place

- Permet aux éditeurs de proposer leur produit (gratuit ou licence facturable)
- Par exemple vous pourrez trouver :
 - Docker Enterprise Edition for Azure
 - Redhat OpenSwift (Paas RH basé sur Docker et Kubernetes : auto build & deploy, monitoring, etc)
 - Mesosphere DC/OS entreprise
 - Pivotal Cloud Foundry

Vous l'aurez compris :



Passons au Dev maintenant

Préparation de l'environnement du Dev

- Outils nécessaires :
 - **Visual Studio 2017** (Community Edition - gratuite)
 - DotNet Core 1.1 / 2.0 beta
 - Hyper-V (édition Professionnel et Entreprise – pour Docker ou VM WS 2016)
 - **Docker for Windows** (container Linux ou Windows)

Vous souhaitez tester une application Micro Service compatible VS 2017 / CLI :
<https://github.com/dotnet-architecture/eShopOnContainers>

DEMO : Support Docker de Visual Studio 2017

Débogage d'une app dans un conteneur

En temps normal, c'est quelque chose de trivial ...

(Pour une app web .NET (pas en PROD))

1. Ajouter à votre Dockerfile :
 - `EXPOSE 4020 4021`
 - `RUN Invoke-WebRequest -OutFile c:\rtools_setup_x64.exe -Uri http://download.microsoft.com/download/1/2/2/1225c23d-3599-48c9-a314-f7d631f43241/rtools_setup_x64.exe;`
 - `RUN & 'c:\rtools_setup_x64.exe' /install /quiet`
2. `docker build`
3. `docker run` avec `-p 4020:4020 -p 4021:4021`
4. Récupérer l'ID/nom du container
5. `docker exec -it <id/name> "C:\Program Files\Microsoft Visual Studio 14.0\Common7\IDE\Remote Debugger\x64\msvsmon.exe" /nostatus /silent /noauth /anyuser /nosecuritywarn`
6. Vous attacher Visual Studio au processus (connexion distante + w3wp.exe pour une app web IIS)

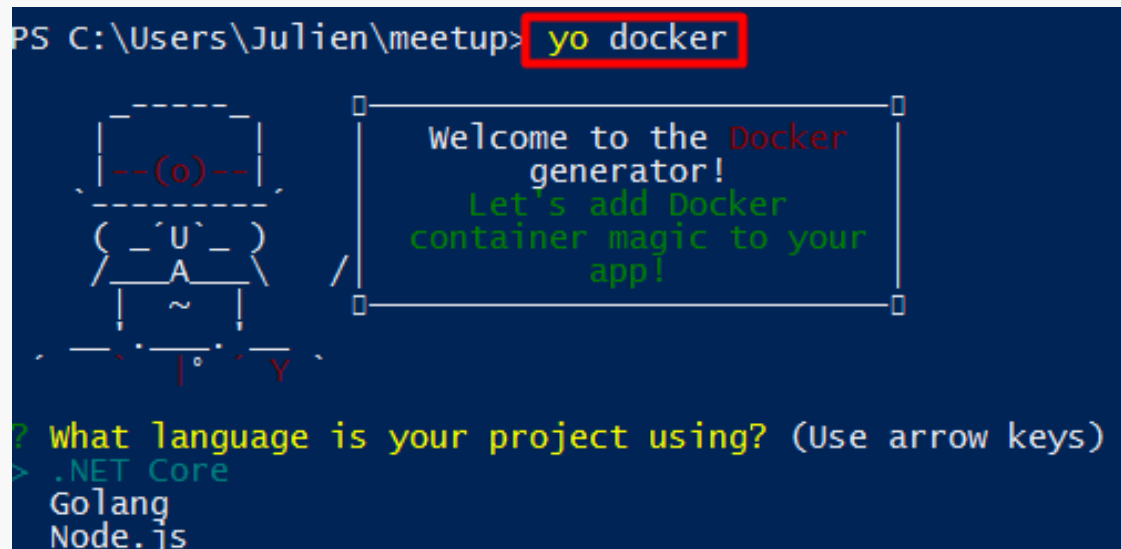
DEMO : Debug dans un container avec Visual Studio 2017

Support de Docker dans Visual Studio Code

Prérequis :

- Visual Studio Code
- Dotnet Core
- NodeJS
- Yeoman (*npm install -g yo*)
- Yeoman Docker Generator (*npm install -g generator-docker*)

```
PS C:\Users\Julien\meetup> yo docker
```



```

  ____
 /--(o)--\
 |  _U_  |
 |  A   |
 |  ~   |
 |_____|
 |  .   |
 |  .   |
 |  .   |
 |_____|

Welcome to the Docker
generator!
Let's add Docker
container magic to your
app!

? What language is your project using? (Use arrow keys)
> .NET Core
  Go lang
  Node.js
```

Conteneur sous Windows

Et migration d'app web legacy

Rappel

- *« Docker for Windows » est différent des conteneurs sous Windows, c'est :*
- *« Docker for Windows »*
 - *est un outil développé par Docker, Inc. Permet d'avoir un environnement de développement prêt à l'emploi pour tester des workloads conteneurisées sur une machine locale !*
 - *permet de travailler avec les conteneurs Docker sous Windows 10 mais aussi avec des conteneurs Docker sous Linux, via une machine virtuelle déployée sur l'Hyper-V local.*
- *Docker sous Windows*
 - Technologie de conteneurisation présente dans le kernel de Windows Server 2016
 - Permet d'exécuter des applications Windows, comme on le fait sous Linux avec les conteneurs Linux.
 - Utilise le format Docker et les outils Docker pour les piloter

Conteneur sous Windows

- Toute la suite d'outils présentés précédemment fonctionne à l'identique sous Windows : format d'image Docker, registry, Docker CLI, Dockerfile...

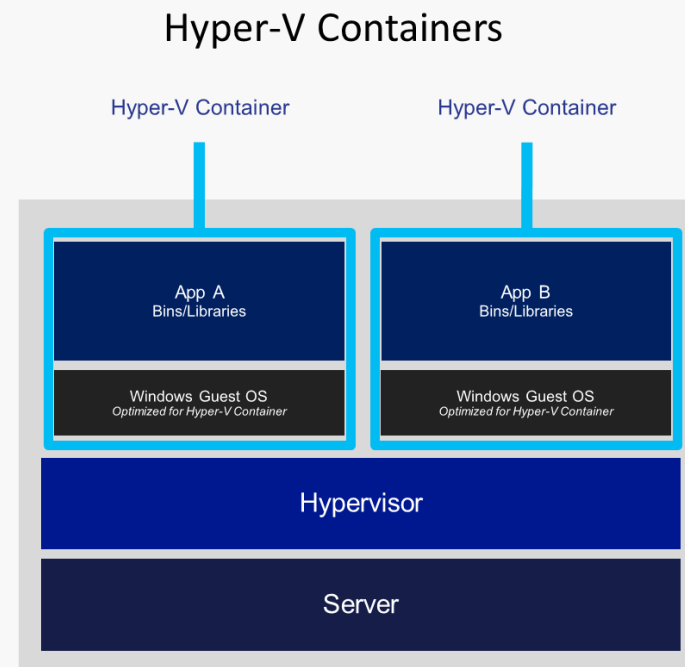
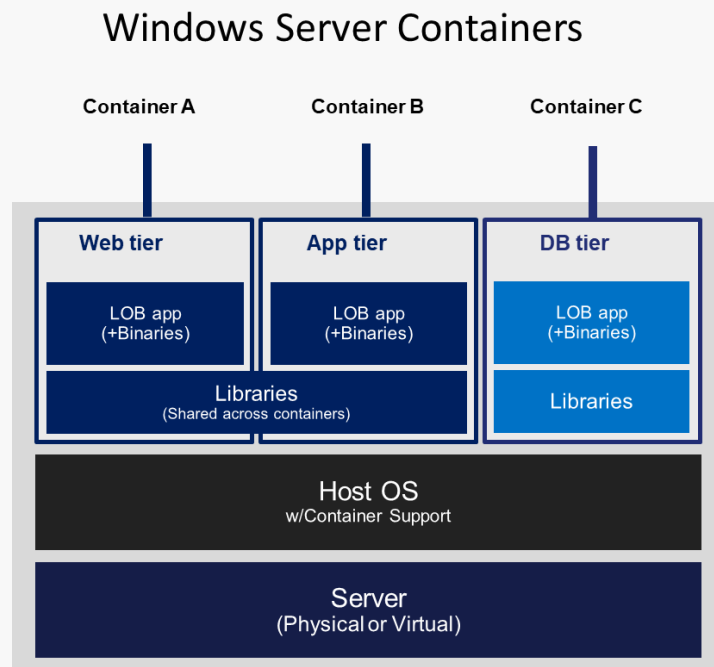
```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> docker version
Client:
Version:      17.05.0-ce
API version:  1.29
Go version:   go1.7.5
Git commit:   89658be
Built:        Thu May  4 21:43:09 2017
OS/Arch:      windows/amd64

Server:
Version:      17.05.0-ce
API version:  1.29 (minimum version 1.24)
Go version:   go1.7.5
Git commit:   89658be
Built:        Thu May  4 21:43:09 2017
OS/Arch:      windows/amd64
Experimental: true
PS C:\Windows\system32> docker images
REPOSITORY              TAG                IMAGE ID            CREATED
rebuild2017.azurecr.io/demo/iis  latest            ee24470ca0c7       3 weeks ago
microsoft/windowsservercore      10.0.14393.1066   590c0c2590e4       4 weeks ago
microsoft/windowsservercore      latest            590c0c2590e4       4 weeks ago
microsoft/nanoserver             10.0.14393.1066   a943c29f0046       4 weeks ago
microsoft/nanoserver             latest            a943c29f0046       4 weeks ago
PS C:\Windows\system32>
```

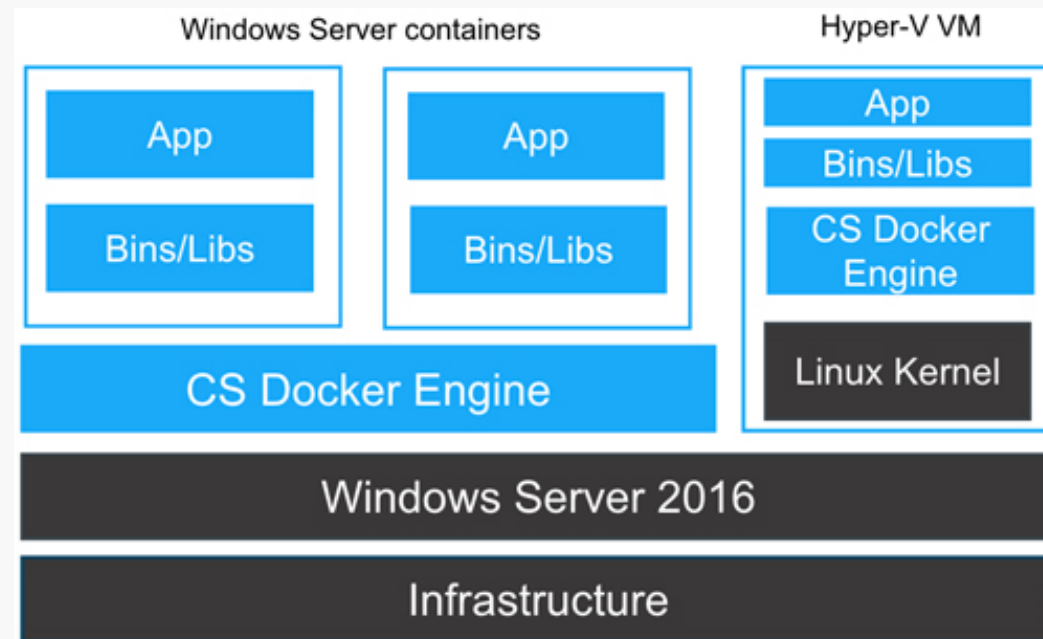
Windows Container / Hyper-V Container

- Maintenant que les choses sont claires, on va pouvoir complexifier 😊
- Il existe deux types de conteneurs Windows : les Windows Containers et les Hyper-V Containers



Windows Container / Hyper-V Container

- Maintenant que les choses sont claires, on va pouvoir complexifier 😊
- Il existe deux types de conteneurs Windows : les Windows Containers et les Hyper-V Containers

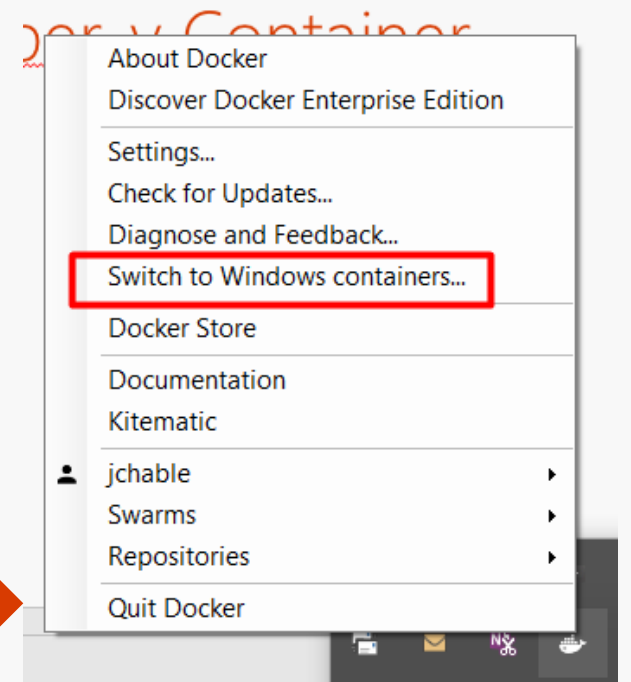


Les images Windows

- Deux images :
 - [Windows Server Core](#) (taille > 10 Go)
 - [Nano Server](#) (une version minimaliste de Windows Server 2016, qui devrait finir sa course par du exclusivement conteneur)
- Disponibles dans le hub Docker
- Exécution sous Windows container ou Hyper-V Container

Migrer une application .NET vers un conteneur Windows Server Core

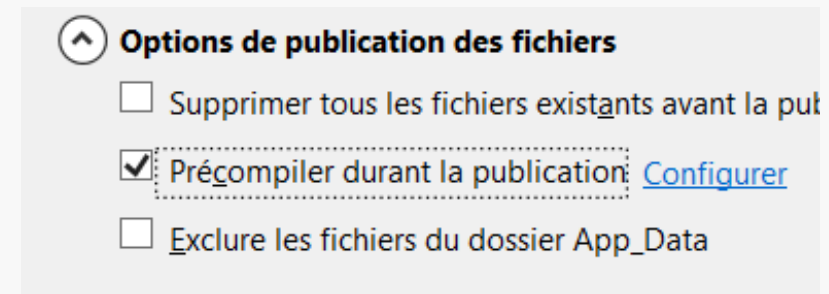
- Aucun changement de l'application n'est nécessaire
- Pour les apps web IIS, le plus simple est d'avoir les sources ou les binaires publiés
- Prérequis :
 - Une bonne connexion
 - Win 10 avec Docker 4 Windows ou Windows Server 2016
 - Pour travailler avec des conteneurs Windows, switcher Docker for Windows vers les conteneurs Windows



Migrer une application .NET vers un conteneur Windows Server Core

1. Créer le script de publication (profile dans VS)

Demandez une pré compilation



2. Créer un fichier Dockerfile

FROM microsoft/aspnet

COPY ./bin/Release/PublishOutput/ /inetpub/wwwroot

3. Construire l'image

docker build -t mvcmeetupdemoiis .

Migrer une application .NET vers un conteneur Windows Server Core

4. Exécuter l'image

```
docker run -d --name meetupdemoiis mvcmeetupdemoiis
```

5. Tester l'application

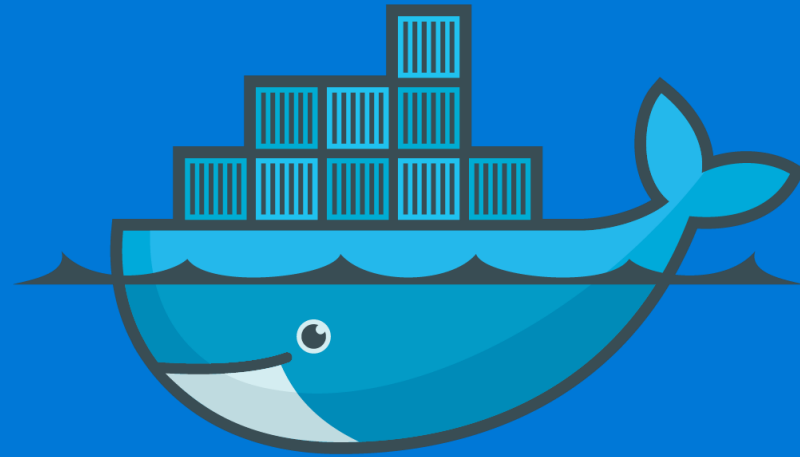
```
docker inspect -f "{{ .NetworkSettings.Networks.nat.IPAddress }}" meetupdemoiis
```

DEMO : Utilisation d'un conteneur Windows

Synthèse

- Constat : Microsoft à embrasser la containerization des applications dès le début de Docker
- Vous pouvez aujourd'hui densifier vos serveurs avec des applications Windows (et pas seulement Linux) en utilisant Windows Server 2016

Merci ! Des questions ?



docker

Ressources

Pour plus d'informations

Github du Meetup
Docker @Noumea

<https://github.com/docker-noumea>

ASP.NET Core avec
Docker

<http://julien.chable.net/deployer-un-projet-web-asp-net-core-1-0-rc2-sur-docker-en-15-minutes>

ASP.NET Core avec
Nano Server

<http://julien.chable.net/tour-dhorizon-de-nano-server-partie-1-creation-dune-image>
<http://julien.chable.net/tour-dhorizon-de-nano-server-partie-2-dployer-un-projet-asp-net-5-avec-iis>

First Step avec
ASP.NET Core

<http://julien.chable.net/dployer-asp-net-5-dans-un-container-docker>

L'écosystème Docker pour les Dev .NET et les Ops Windows

Julien Chable
Consultant – MVP

@Jchable
julien.chable@ncit.nc

Blog : <http://julien.chable.net/>

