

Docker - Configuración de ASP.Net

ASP.Net es el marco de desarrollo web estándar proporcionado por Microsoft para desarrollar aplicaciones del lado del servidor. Dado que ASP.Net ha existido durante bastante tiempo para el desarrollo, Docker se ha asegurado de que tenga soporte para ASP.Net.

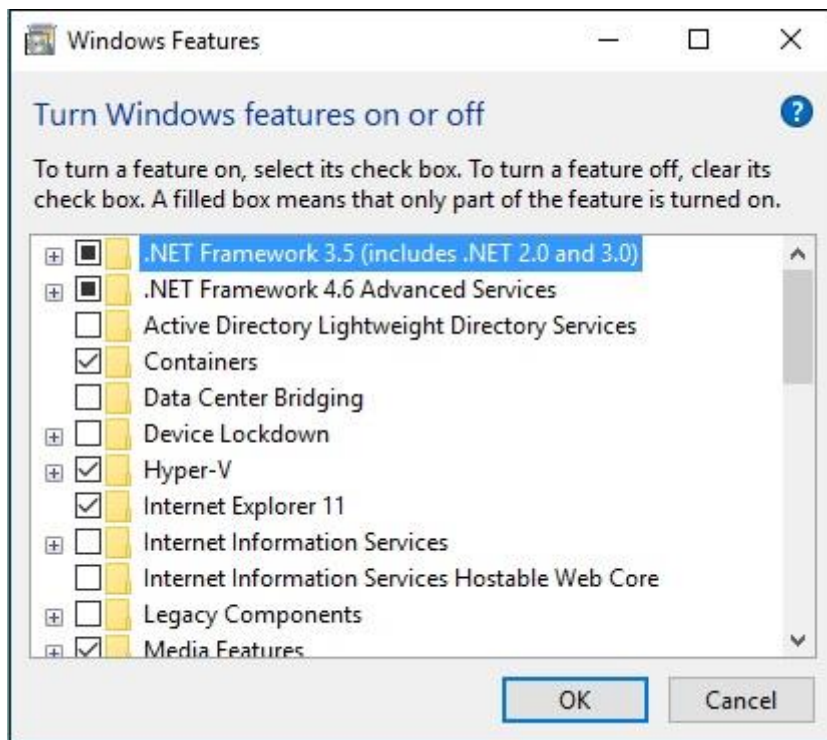
En este capítulo, veremos los distintos pasos para poner en funcionamiento el contenedor Docker para ASP.Net.

Prerrequisitos

Los siguientes pasos deben realizarse primero para ejecutar ASP.Net.

Paso 1 : dado que esto solo se puede ejecutar en sistemas Windows, primero debe asegurarse de tener Windows 10 o Windows Server 2016.

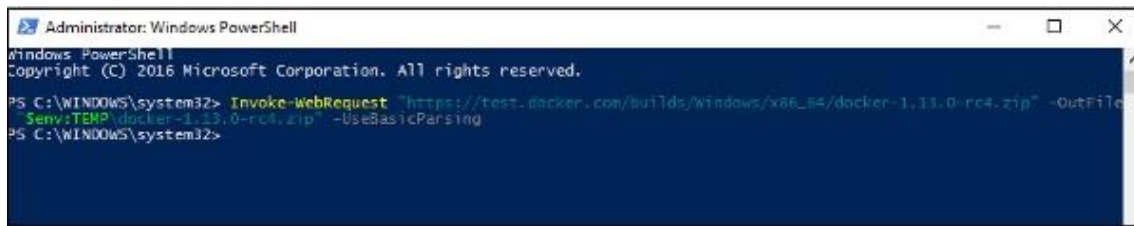
Paso 2 : a continuación, asegúrese de que Hyper-V y los contenedores estén instalados en el sistema Windows. Para instalar Hyper - V y contenedores, puede ir a Activar o desactivar las características de Windows. Luego, asegúrese de que la opción Hyper-V y los contenedores estén marcados y haga clic en el botón Aceptar.



Es posible que sea necesario reiniciar el sistema después de esta operación.

Paso 3 : a continuación, debe usar el siguiente comando de Powershell para instalar la versión **1.13.0rc4** de Docker. El siguiente comando descargará esto y lo almacenará en la ubicación temporal.

```
Invoke-WebRequest  
"https://test.docker.com/builds/Windows/x86_64/docker-1.13.0-  
rc4.zip" -OutFile "$env:TEMP\docker-1.13.0-rc4.zip" -  
UseBasicParsing
```

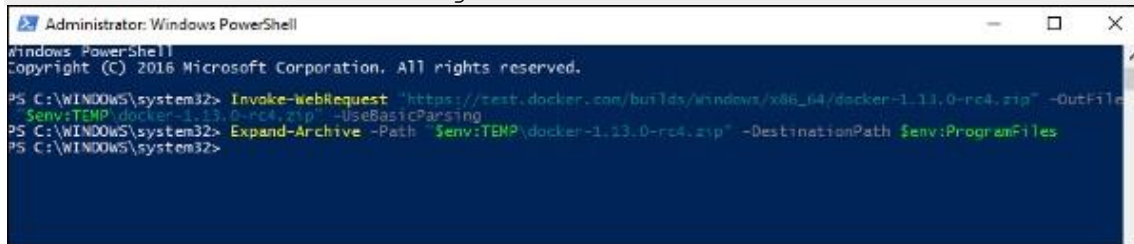


```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\WINDOWS\system32> Invoke-WebRequest "https://test.docker.com/builds/Windows/x86_64/docker-1.13.0-rc4.zip" -OutFile $env:TEMP\docker-1.13.0-rc4.zip -UseBasicParsing
PS C:\WINDOWS\system32>
```

Paso 4 : a continuación, debe expandir el archivo con el siguiente comando de **powershell** .

```
Expand-Archive -Path "$env:TEMP\docker-1.13.0-rc4.zip" -
DestinationPath $env:ProgramFiles
```



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

PS C:\WINDOWS\system32> Invoke-WebRequest "https://test.docker.com/builds/Windows/x86_64/docker-1.13.0-rc4.zip" -OutFile $env:TEMP\docker-1.13.0-rc4.zip -UseBasicParsing
PS C:\WINDOWS\system32> Expand-Archive -Path $env:TEMP\docker-1.13.0-rc4.zip -DestinationPath $env:ProgramFiles
PS C:\WINDOWS\system32>
```

Paso 5 : a continuación, debe agregar los archivos Docker a la variable de entorno utilizando el siguiente comando de **powershell** .

```
$env:path += ";$env:ProgramFiles\Docker"
```

Paso 6 : a continuación, debe registrar el servicio Docker Daemon utilizando el siguiente comando de **powershell** .

```
dockerd --register-service
```

Paso 7 : finalmente, puede iniciar **Docker Daemon** con el siguiente comando.

```
Start-Service Docker
```

Use el comando **docker version** en **powershell** para verificar que el **Docker Daemon** esté funcionando

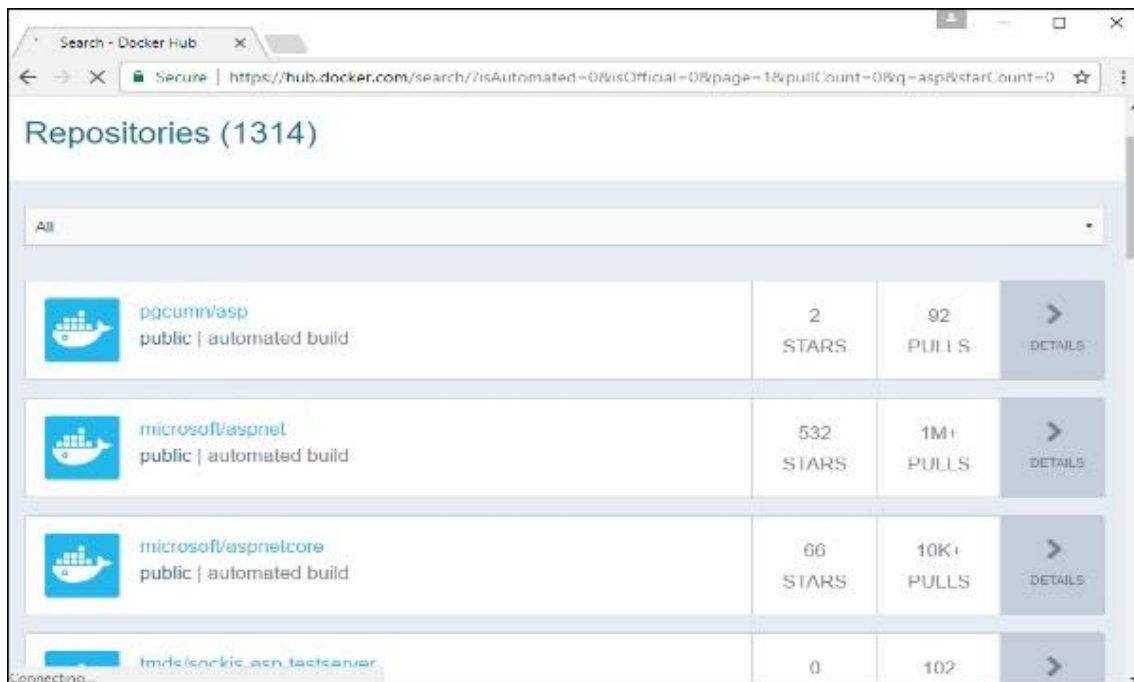


```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> docker version
Client:
Version:      1.13.0-rc4
API version:  1.25
Go version:   gol.7.3
Git commit:   88862e7
Built:        Sat Dec 17 01:34:17 2016
OS/Arch:      windows/amd64
Server:
Version:      1.13.0-rc4
API version:  1.25 (minimum version 1.24)
Go version:   gol.7.3
Git commit:   88862e7
Built:        Sat Dec 17 01:34:17 2016
OS/Arch:      windows/amd64
Experimental: false
PS C:\WINDOWS\system32>
```

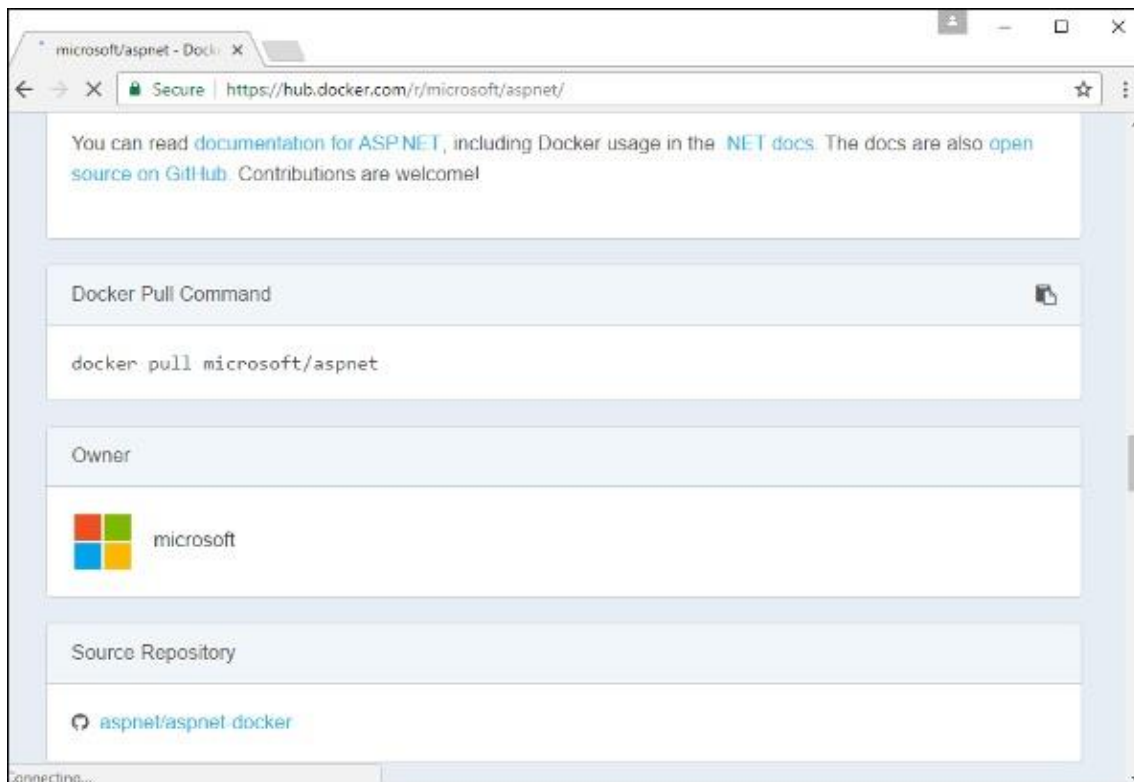
Instalación del contenedor ASP.Net

Veamos cómo instalar el contenedor ASP.Net.

Paso 1 : el primer paso es extraer la imagen de Docker Hub. Cuando inicie sesión en Docker Hub, podrá buscar y ver la imagen de **Microsoft / aspnet** como se muestra a continuación. Simplemente escriba **asp** en el cuadro de búsqueda y haga clic en el enlace Microsoft / aspnet que aparece en los resultados de búsqueda.



Paso 2 : verá que el comando de **extracción de** Docker para ASP.Net en los detalles del repositorio en Docker Hub.



Paso 3 : vaya al host de Docker y ejecute el comando de **extracción de** Docker para la imagen de microsoft / aspnet. Tenga en cuenta que la imagen es bastante grande, cercana a los 4,2 GB.

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> docker pull microsoft/aspnet
Using default tag: latest
latest: Pulling from microsoft/aspnet
3889bb8d808b: Pull complete
04ee5d718c7a: Pull complete
c0931dd15237: Pull complete
61784b745c20: Pull complete
d05122f129ca: Pull complete
c31417523ee9: Pull complete
Digest: sha256:48e6b45a43bc58e71da3d20d73839c29cc1c7ce87ed06ad257c4150e5286ade3
Status: Downloaded newer image for microsoft/aspnet:latest
PS C:\WINDOWS\system32>
```

Paso 4 : ahora vaya a la siguiente ubicación <https://github.com/Microsoft/aspnet-docker> y descargue todo el repositorio de Git.

Paso 5 : cree una carpeta llamada **Aplicación** en su unidad C. Luego copie el contenido de la carpeta **4.6.2 / sample** a su unidad C. Vaya al archivo Docker en el directorio de muestra y ejecute el siguiente comando:

```
docker build -t aspnet-site-new -build-arg site_root=/
```

Los siguientes puntos deben tenerse en cuenta sobre el comando anterior:

- **Crea** una nueva imagen llamada **aspnet-site-new** desde el archivo Docker.
- La ruta raíz se establece en la carpeta de ruta local.

```
Administrator: Windows PowerShell
-a----          11/3/2016  11:54 AM                72 Dockerfile

PS C:\App> cd 4.6.2
PS C:\App\4.6.2> dir

Directory: C:\App\4.6.2

Mode                LastWriteTime         Length Name
----                -
d-----          1/15/2017   8:52 AM             sample
-a----          11/3/2016  11:54 AM             216 Dockerfile

PS C:\App\4.6.2> cd sample
PS C:\App\4.6.2\sample> dir

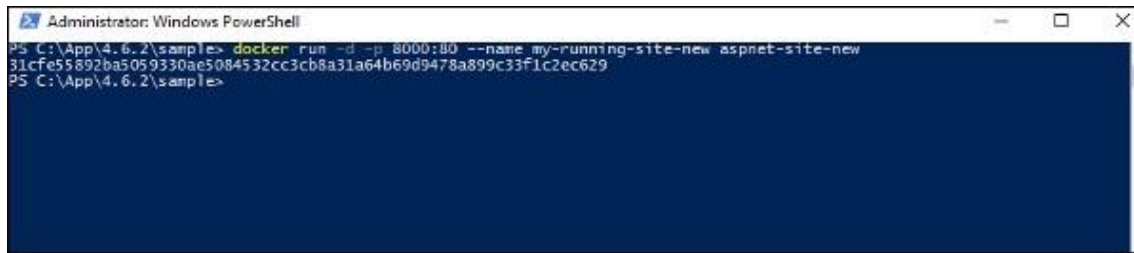
Directory: C:\App\4.6.2\sample

Mode                LastWriteTime         Length Name
----                -
-a----          11/3/2016  11:54 AM             72 Dockerfile

PS C:\App\4.6.2\sample> docker build -t aspnet-site-new --build-arg site_root=/.
Sending build context to Docker daemon 2.048 kB
Step 1/3 : FROM microsoft/aspnet
--> 08897a3b116a
Step 2/3 : ARG site_root=.
--> Using cache
--> 4c89c01bef2d
Step 3/3 : ADD ${site_root} /inetpub/wwwroot
--> d1ce3a0ca1bc
Removing intermediate container 8652c844e52a
Successfully built d1ce3a0ca1bc
PS C:\App\4.6.2\sample>
```

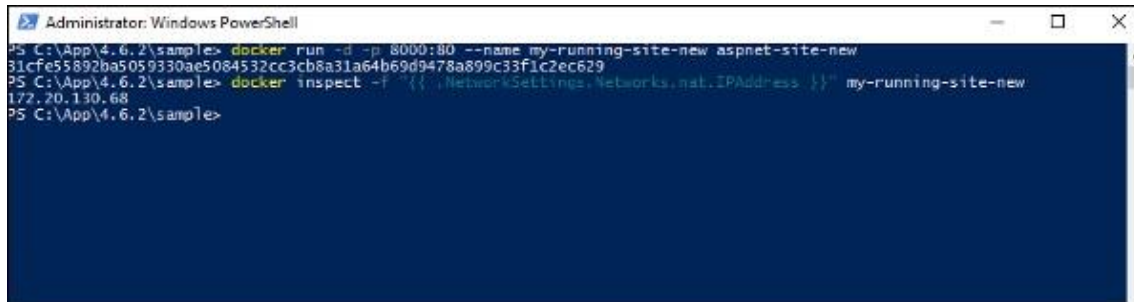
Paso 6 : ahora es el momento de ejecutar el contenedor. Se puede hacer usando el siguiente comando:

```
docker run -d -p 8000:80 -name my-running-site-new aspnet-site-new
```



```
Administrator: Windows PowerShell
PS C:\App\4.6.2\sample> docker run -d -p 8000:80 --name my-running-site-new aspnet-site-new
31cfe55892ba5059330ae5084532cc3cb8a31a64b69d9478a899c33f1c2ec629
PS C:\App\4.6.2\sample>
```

Paso 7 : ahora tendrá IIS ejecutándose en el contenedor de Docker. Para encontrar la dirección IP del contenedor de Docker, puede ejecutar el comando de **inspección de** Docker como se muestra a continuación.



```
Administrator: Windows PowerShell
PS C:\App\4.6.2\sample> docker run -d -p 8000:80 --name my-running-site-new aspnet-site-new
31cfe55892ba5059330ae5084532cc3cb8a31a64b69d9478a899c33f1c2ec629
PS C:\App\4.6.2\sample> docker inspect -f "{{ .NetworkSettings.Networks.nat.IPAddress }}" my-running-site-new
172.20.130.68
PS C:\App\4.6.2\sample>
```