Usando Docker para crear un servidor MySQL

por Rob Allen MVB · Nov. 03, 17 · Zona de base de datos

Navegar por las opciones actuales de escalamiento de la base de datos puede ser una pesadilla. Explore los compromisos involucrados tanto en las arquitecturas tradicionales como en las nuevas.

Cuando trabajo en código de prueba en mi computadora, suelo usar el servidor PHP incorporado (`php -S`), que funciona muy bien. De vez en cuando, necesito acceso a MySQL y utilizo Docker para crear temporalmente un servidor MySQL para mí. Así es como lo hago.

El comando mágico es:

```
1  $ docker run --name mysql \
     -e  MYSQL_USER = rob -e  MYSQL_PASSWORD = 12345
2
     -p  3306 : 3306 -d mysql / mysql-server: 5.7
3
```

Esto crea un contenedor acoplable llamada `mysql` en el puerto 3306. Pasamos tres variables de entorno: `MYSQL_USER` , `MYSQL_PASSWORD` y `MYSQL_DATABASE` , que son nuestras credenciales y el nombre de la base de datos.

Acceso del cliente a la base de datos

Si tiene instalado el cliente de línea de comandos de MySQL, puede acceder a su base de datos de esta manera:

```
$ mysql  --protocol = TCP  -u rob  -p123456
```

```
1 $ mysql --protocol = TCP -u rob -p123456
```

Necesitamos usar el protocolo TCP ya que no hay un socket en juego aquí. Si, como yo, eres demasiado perezoso para escribir `--protocol=TCP` cada vez, luego configúralo en tu `~/.my.cnf` archivo de la siguiente manera:

`~/.my.cnf`:

```
1 [cliente]
2 protocolo = TCP
```

Una de las formas más fáciles de obtener un cliente de línea de comandos MySQL en una Mac es instalar MySQL WorkBench y agregar `/Applications/MySQLWorkbench.app/Contents/MacOS` a la ruta, ya que Homebrew parece querer instalar el servidor también.

Controlando el contenedor

Detenga el uso del contenedor `docker stop mysql` y reinícielo de nuevo con `docker start mysql`. Cuando hayas terminado, puedes eliminarlo con `docker rm mysql`.

Restaurando un archivo de volcado

Como estas instancias de MySQL son temporales para mí, instalo un esquema de base de datos usando:

```
1 cat seed-mysql.sql | mysql --protocol = TCP -u ro
```

(¡Obviamente, usarías las mismas credenciales mientras corrías tu contenedor!)

Aleta

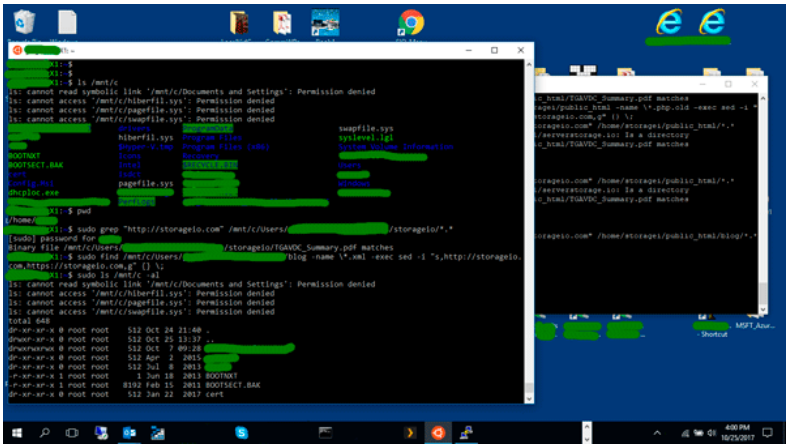
Eso es todo lo que hay para tener una instalación temporal de MySQL ejecutándose localmente para el desarrollo.

Temas: DOCKER, MYSQL, SERVIDOR, BASE DE DATOS, TUTORIAL

The Windows Subsystem for Linux

by Greg Schulz MVB · Nov 03, 17 · Integration Zone

Microsoft has been increasing their support of Linux across Azure public cloud, Hyper-V and Linux Integration Services (LIS) and Windows platforms including Windows Subsystem for Linux (WSL) as well as Server along with Docker support.



WSL with Ubuntu installed and open in a window on one of my Windows 10 systems.

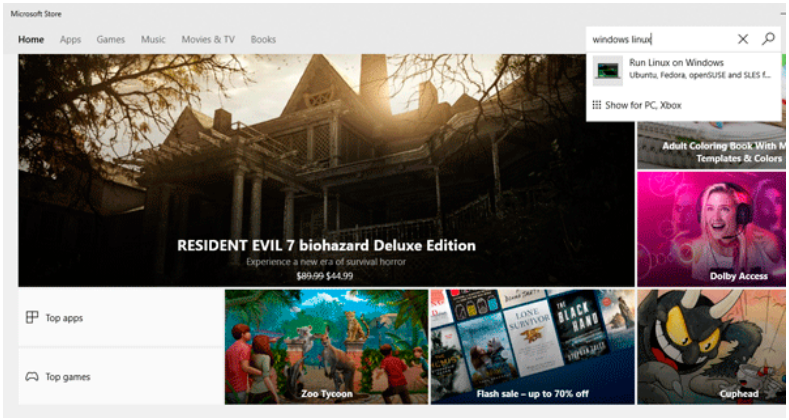
WSL is not a virtual machine (VM) running on Windows or Hyper-V, rather it is a subsystem that coexists next to win32 (read more about how it works and features, enhancements here). Once installed, WSL enables the use of Linux bash shell along with familiar tools (find, grep, sed, awk, and rsync, among others) as well as services such as SSH and MySQL, among others.

What this all means is that if you work with both Windows and Linux, you can do so on the same desktop, laptop, server or system using your preferred commands. For example, in one window you can be using Powershell or traditional Windows commands and tools, while in another window working with grep, find, and other tools eliminating the need to install things such as wingrep, among others.

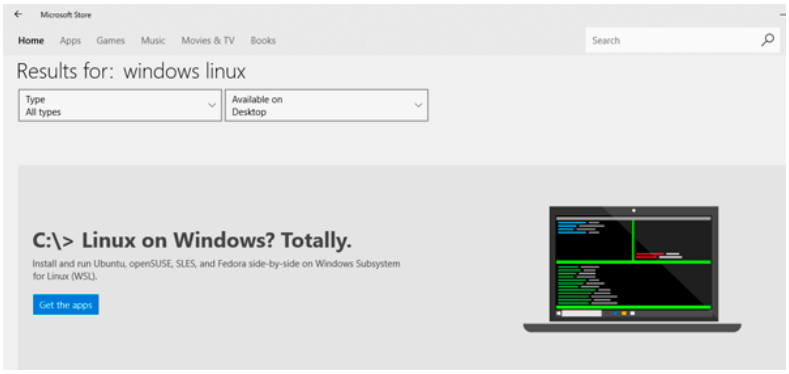
Installing WSL

Depending on which release of Windows desktop or server you are running, there are a couple of different install paths. Since my Windows 10 is the most recent release (e.g. 1709) I was able to simply go to the

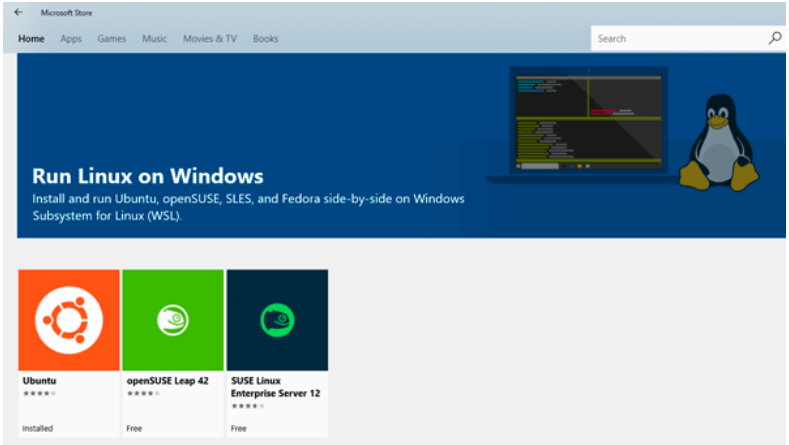
Microsoft Windows Store via the desktop, search for Windows Linux, select the distribution, install and launch. Microsoft has some useful information for installing WSL on different Windows version here, as well as for Windows Servers here.



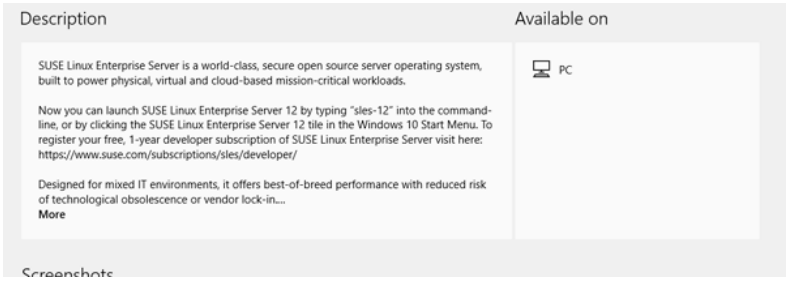
Get WSL from the Windows Store or more information and options here.

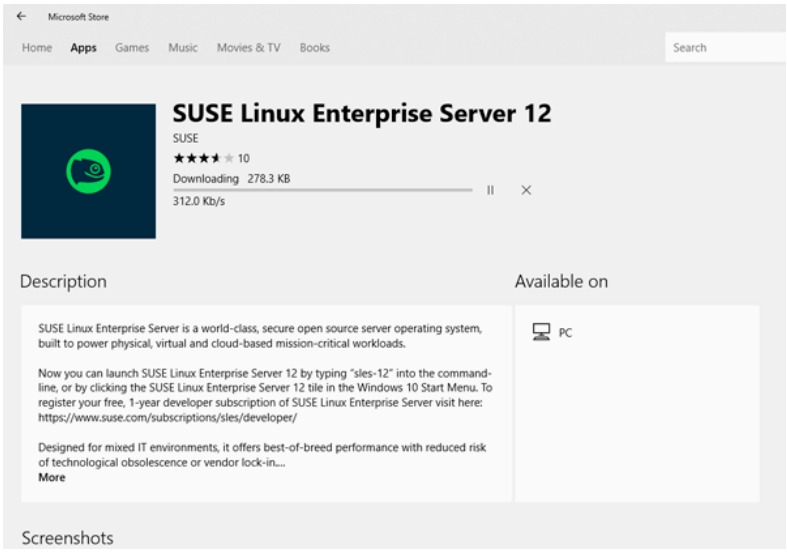


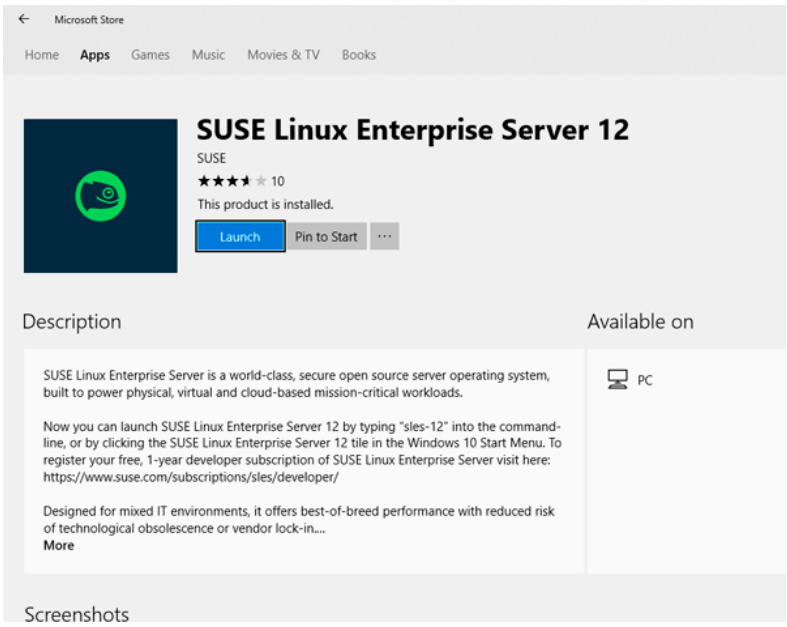
Click on "Get the app."

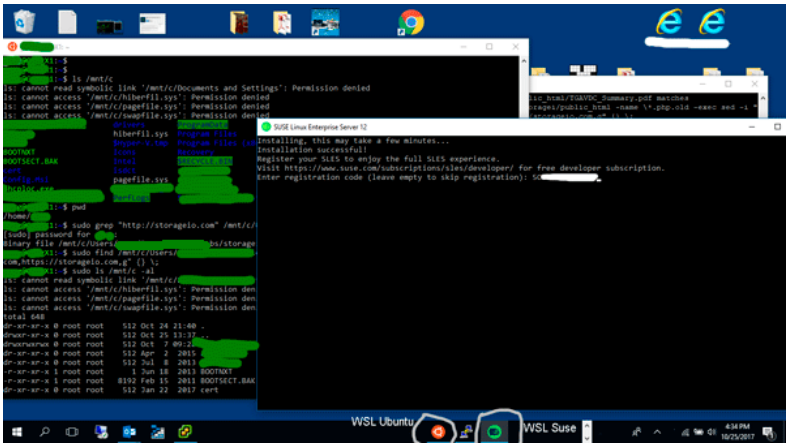


Select the desired WSL distribution.

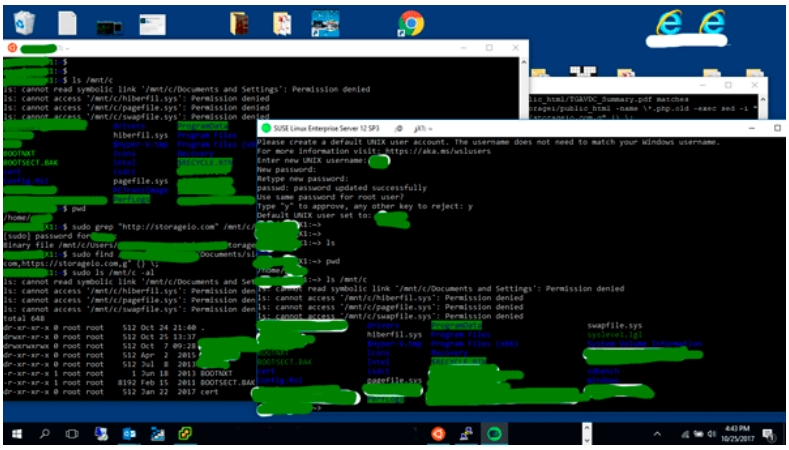








Let's select SUSE, as I already have Ubuntu installed (I have both).



SUSE WSL is in the process of downloading. Note that SUSE needs an access code (free) that you get from https://www.suse.com/subscriptions/sles/developer/; while waiting for the download and installation is a good time to get that code.

Launching WSL with SUSE, you will be prompted to enter the code mentioned above; if you do not have a code, get it here from SUSE.

The WSL installation is very straightforward: enter the SUSE code (Ubuntu did not need a code). Note the Ubuntu and SUSE WSL task bar icons circled in the bottom center.

Provide a username for accessing the WSL bash shell along with a password, confirm how root and sudo are to be applied, and that is it. Seriously, the install for WSL- at least with Windows 10 1709- is that fast and easy. Note that in the above image, I have WSL with Ubuntu open in a window on the left, WSL with SUSE on the right, and their taskbar icons bottom center.

Using WSL

Once you have WSL installed, try something simple, such as viewing your present directory:

```
1  pwd
```

Then look at the Windows C: drive location.

```
1  ls /mnt/c -al
```

In case you did not notice the above, you can use Windows files and folders from the bash shell by placing /mnt in front of the device path. Note that you need to be case-sensitive such as User vs. user or Documents vs. documents.

As a further example, I needed to change several .htm, .html, .php and .xml files on a Windows system whose contents had not yet changed from http://storageio.com to https://storageio.com. Instead of installing wingrep or some tools, using WSL such as with Ubuntu finding files can be done with grep such as:

```
grep "http://storageio.com" /mnt/c/Users/*.xml
```

And then making changes using find and sed such as:

```
find /mnt/c/Users -name \*.xml -exec sed  -i "s,h
```

Note that not all Linux apps and tools can use file via /mnt, in which case a solution is to create a symbolic link.

For example:

```
ln -s "/mnt/c/Users/Test1/Documents"  /home/Test1
```

What This All Means

If you primarily work on (or have a preference for) Linux systems and need to do some functions from development to the administration or other activity on a Windows system, Windows Subsystem for Linux (WSL) provides a bash shell to do familiar tasks. Likewise, if you are primarily a Windows person and need to brush up on your Linux skills, WSL can help. If you need to run Linux server applications or workloads, put those into a Docker container, Hyper-V instance or Azure VM.

Overall I like WSL for what it is, a tool that eliminates the need of having to install several other tools to do common tasks, plus makes it easier to work across various Linux and Windows systems including bare metal, virtual and cloud-based. Now that you have been introduced to Windows Subsystems for Linux WSL and an overview including install as well as using, add it to your data infrastructure toolbox.

Topics: DATA INFRASTRUCTURE, DATACENTER TECHNOLOGIES, CLOUD