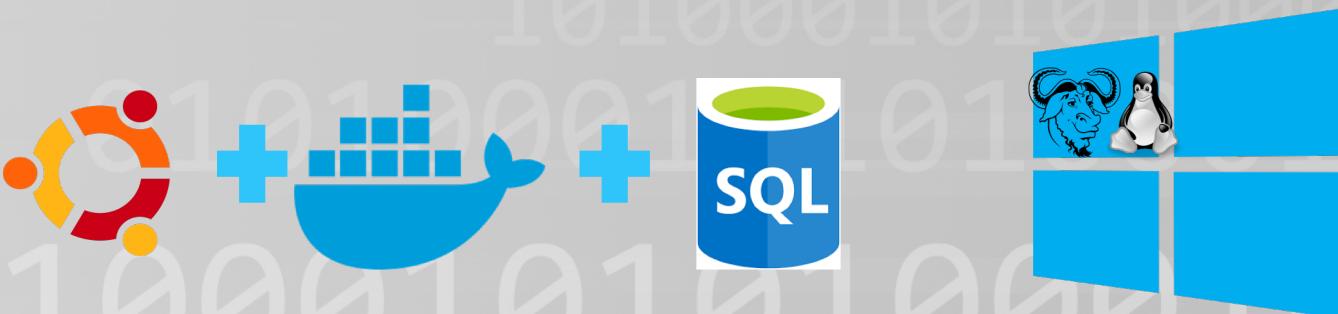


1010001010100010101

# Getting started with database development with WSL2





# Carlos Lopez

Microsoft MVP MCP Data Platform



/carlos-lopez-taks



@CarlosLopezDBA



carlosarturo.lopeztaks



caltls@gmail.com

## Experience

Microsoft Certified Professional 2012/2014,  
2016-2017

More 10 years of experience

Multi-platform DBA

## Community

Guatemala SQL Server User Group – board  
member

Blogger, Speaker

## Fields of Experience

RDBMS: MS SQL Server, MySQL

Oracle 10-11g, NoSQL: Mongo

Linux Distros





# Agenda

## WSL2 Insights

What is it? – Features

## Performance Facts & History

Performance tests and Benchmarking

## Preparing your Environment

Configuring WSL2 , Requirements & scenarios (Containers), Preparing your SQL, IDE's and Environment.

## Demo

Checking the Environment Containers – DB - OS

01

02

03

04



# WSL2 Insights

What is it? – Features

# What is WSL2?

Windows Subsystem for Linux v2



**Complete GNU subsystem over windows**

GNU/Linux distros in Microsoft Store



Executes Bash shell scripts **transparently**



Full Software install within subsystem GNU/Linux

Executes RPM packages from GNU/Linux on Windows

# What is WSL2?

Windows Subsystem for Linux v2



Executes Linux binaries on Windows

Performance improvement on File System.



Full compatibility on windows kernel operative system

# Performance Facts & History

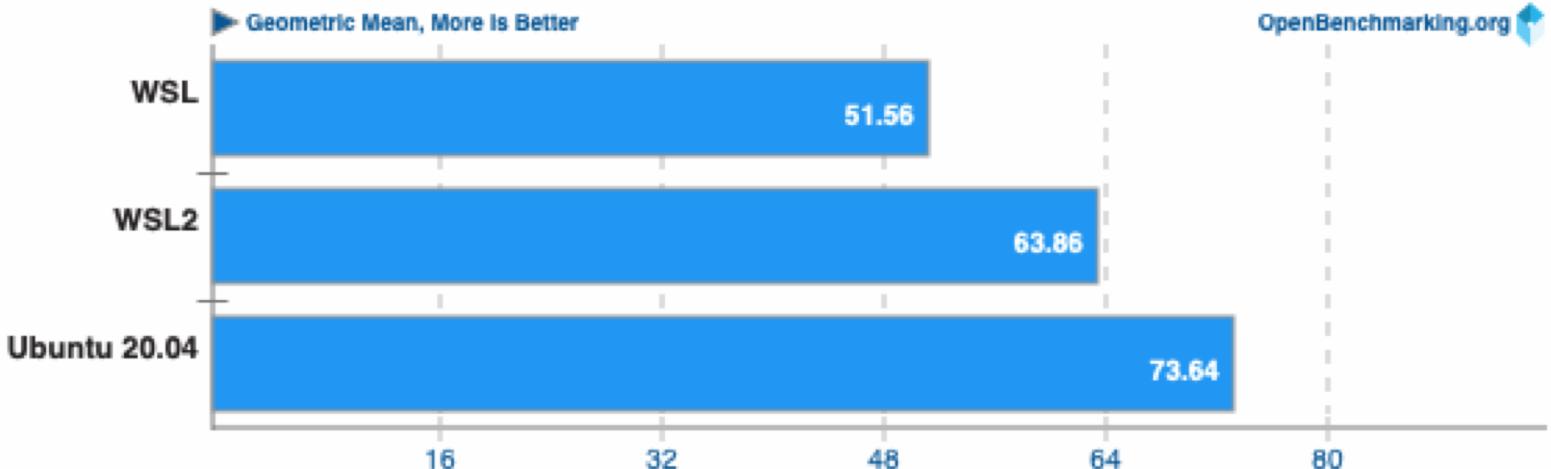
Benchmarking Tests made - Docker on Windows Story So Far...

# Performance Benchmark Tests

Overall tests

## Geometric Mean Of All Test Results

Result Composite



# Performance Benchmark Tests

RDBMS Test

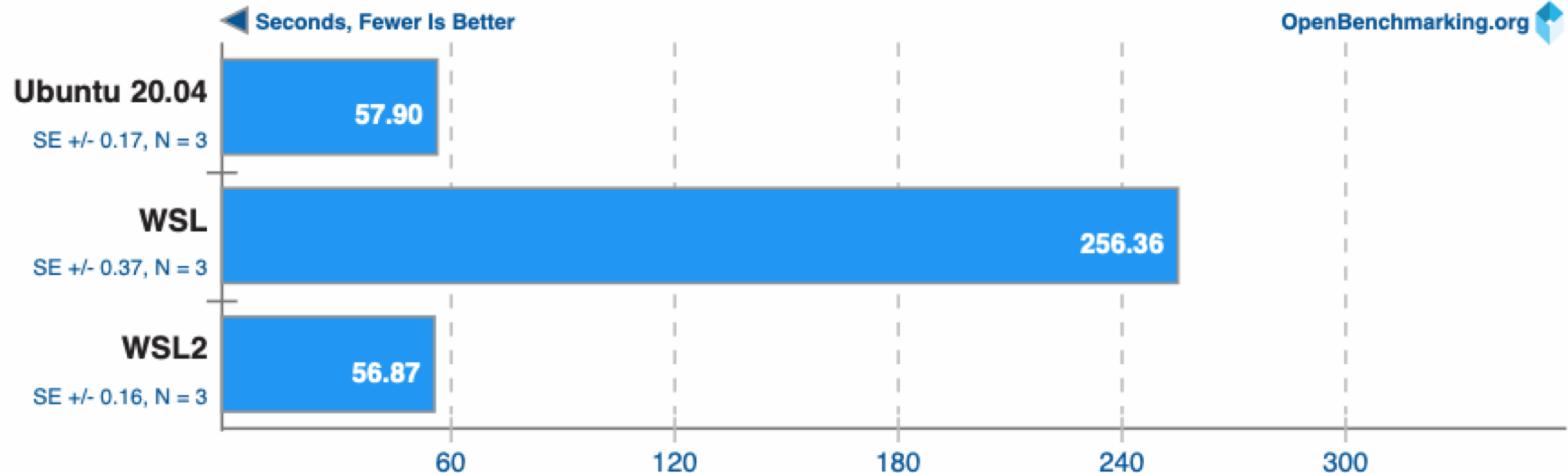
## SQLite Speedtest

### SQLite Speedtest v3.30

Timed Time - Size 1,000



OpenBenchmarking.org



# Performance Benchmark Tests

NoSQL Tests

Redis

## Redis v5.0.5

Test: SADD



# **Preparing your environment**

Configuring WSL2 , Requirements (Containers),  
Preparing your SQL, IDE's and Environment

# Using WSL2

Requirement

Windows 10 ver. 2004 (now on Windows Home)

## 1. Enable PowerShell CLI

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

## 2. Install Docker Desktop

## 3. Configure WSL ver 2

```
wsl -l -v
```

Upgrade V2:

```
wsl --set-version (distro name) 2
```

```
wsl --set-default-version 2
```

NAME	STATE	VERSION
* Ubuntu-20.04	Running	2
docker-desktop-data	Running	2
docker-desktop	Running	2

```
PS C:\Users\Beralios> wsl -l -v
NAME          STATE    VERSION
* Ubuntu-20.04 Running   2
docker-desktop-data Running   2
docker-desktop  Running   2

PS C:\Users\Beralios> wsl
Welcome to Ubuntu 20.04 LTS (GNU/Linux 4.19.104-microsoft-standard x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 System information as of Sat Aug 29 20:09:18 GMT 2020

 System load: 0.15           Processes:           11
 Usage of /: 1.4% of 250.98GB Users logged in:      0
 Memory usage: 6%            IPv4 address for eth0: 172.21.250.9
 Swap usage:  0%
```

# Facts

Containers / No Containers

Pros	Cons
Containerized Environment	Specialized knowledge
Microservices Technology	Complexity between layers
Dev / Ops culture	Stand-Alone isolated stage coupled vs decoupled
IaC Search	

# Scenario

Identified Problem

Problem:

```
ForceFlush is enabled for this instance.  
ForceFlush feature is enabled for log durability.  
System has not been booted with systemd as init system (PID 1). Can't operate.  
Failed to connect to bus: Host is down  
Attempting to start the Microsoft SQL Server service failed.
```

Force “systemd” in Ubuntu

Set adequate Run-level

Grant permissions to the binaries

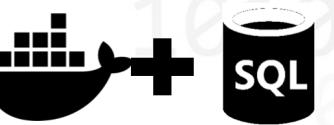
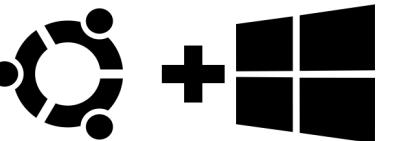
# Scenario

What can I do with it?

Installed Ubuntu LTS 20.04

set Docker with Ubuntu

Install SQL Server 2017 - 19 on  
Ubuntu



# Prepare your SQL

- Define your SQL Container on a Dockerfile
- Create Volumes dedicated for SQL container for your data.
- Build and **test** your Docker Container SQL Image
- Configure your Git Repository
- Run your image as a container
- Use ACI feature pull your container into your repository



# Tools and IDE's Environment



Visual Studio Code WSL2 Extension



SQL Server VSC Extension



Azure CLI Tools



Docker VSC Extension



Azure Data Studio



SSMS



# Demo

Hovering the tool

# Important Notes

WSL2 it's a best choice to manage Docker SQL Containers Windows

WSL2 file system management improves substantially the SQL Containers I/O on Windows

Always dedicate volumes for your SQL Container Data



# Material



## Github

<https://github.com/Muppity/Presentations-Material>

## Docker, Kubernetes Resources

[https://hub.docker.com/\\_/microsoft-mssql-server](https://hub.docker.com/_/microsoft-mssql-server)

[Ubuntu on WSL2](#)

[Microsoft WSL2 Guide](#)

[WSL2 Complete Benchmark Report](#)

[phoronix.com Performance for WSL vs WSL2](#)

Simon Ferquel, wsl2 blog



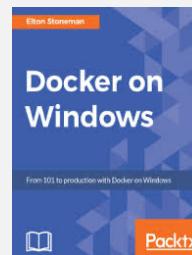
## Social Networks



/carlos-lopez-taks



@CarlosLopezDBA



**Thank you!**