

- MSSQL
Crossplatform y
Portable con
Docker



Hola!

Soy Christian Melendez

Senior Solutions Architect @ Autoweb
Arquitectura + Cloud + .NET + Docker
Twitter: @Christianhxc

● Agenda

○ Docker
MSSQL Linux + Docker
Casos de Uso
Demo



1

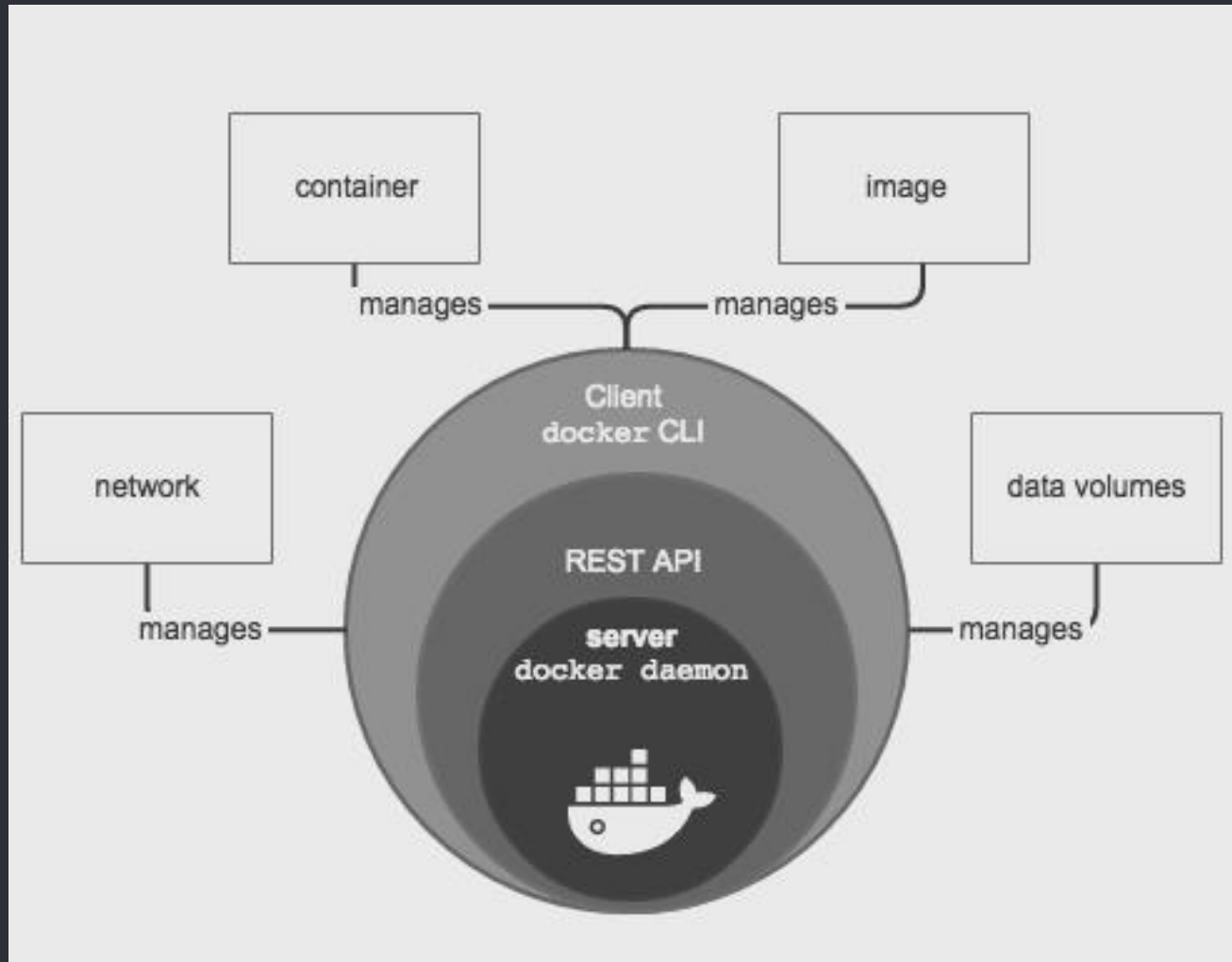
Docker

Plataforma para desarrollar, ejecutar y entregar apps.
Windows, Linux, Mac y Cloud.

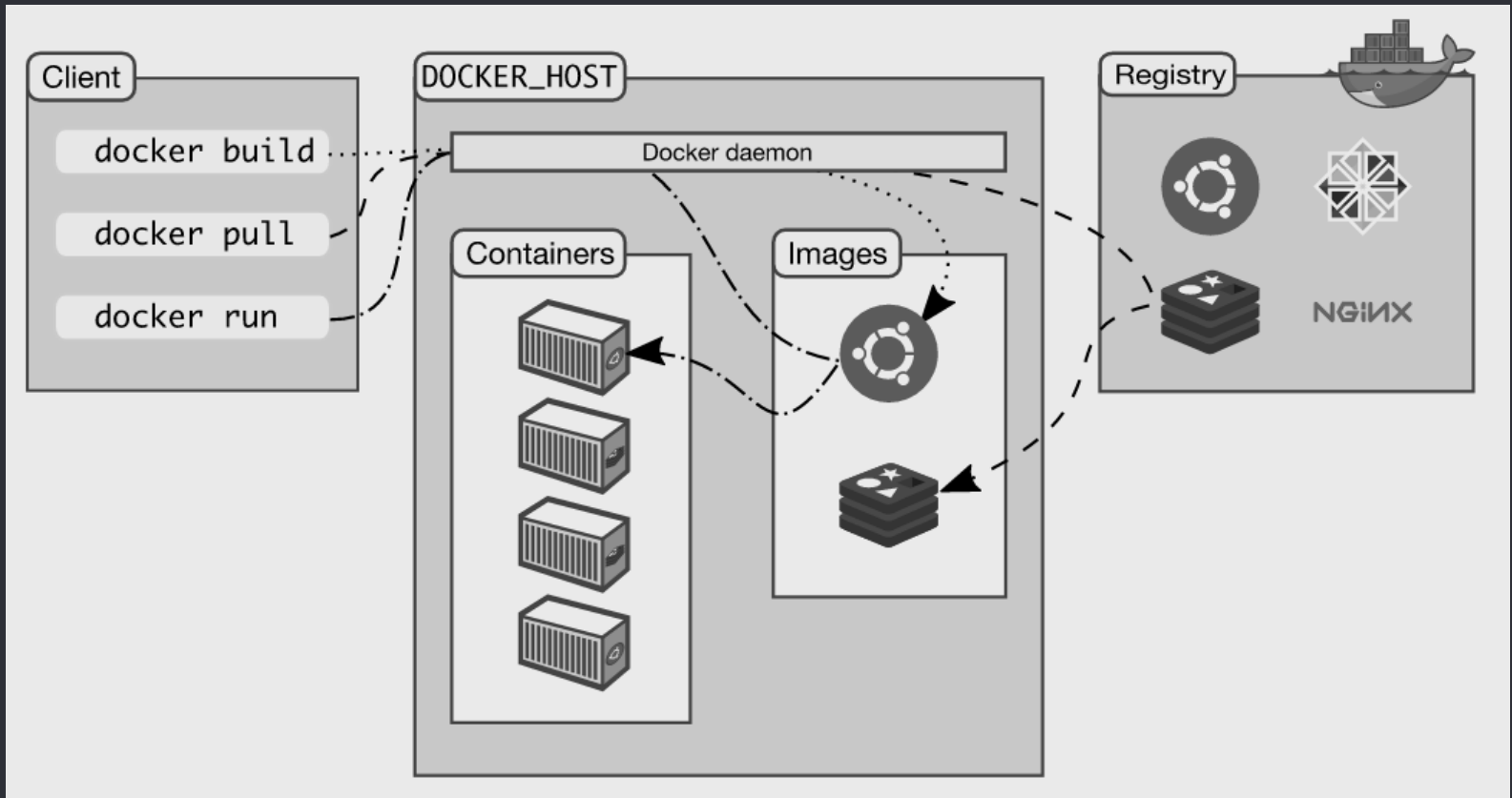
“

*Docker te permite separar tus **aplicaciones de tu infraestructura** para que puedas entregar software **rápidamente**. Con Docker, puedes administrar tu infraestructura de la misma manera en que administras tus aplicaciones.*

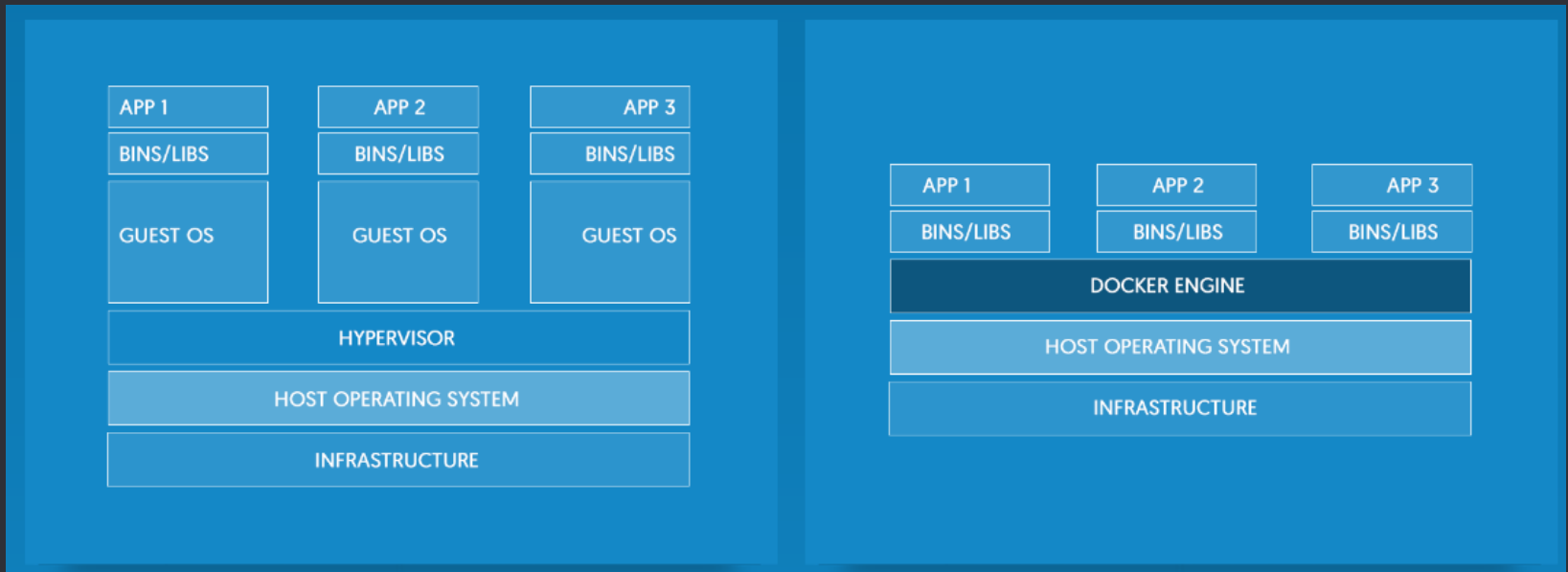
● Docker: Engine



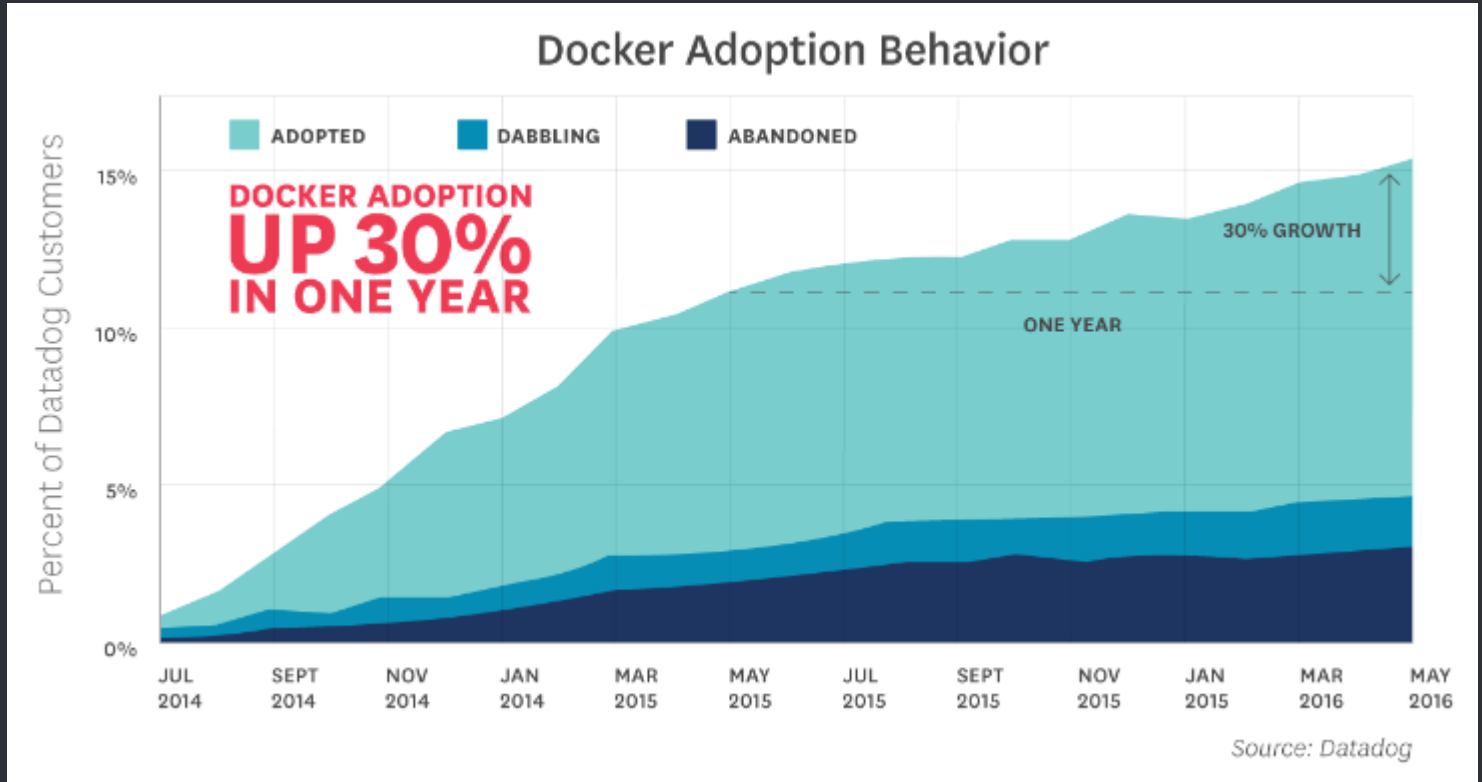
Docker: Arquitectura



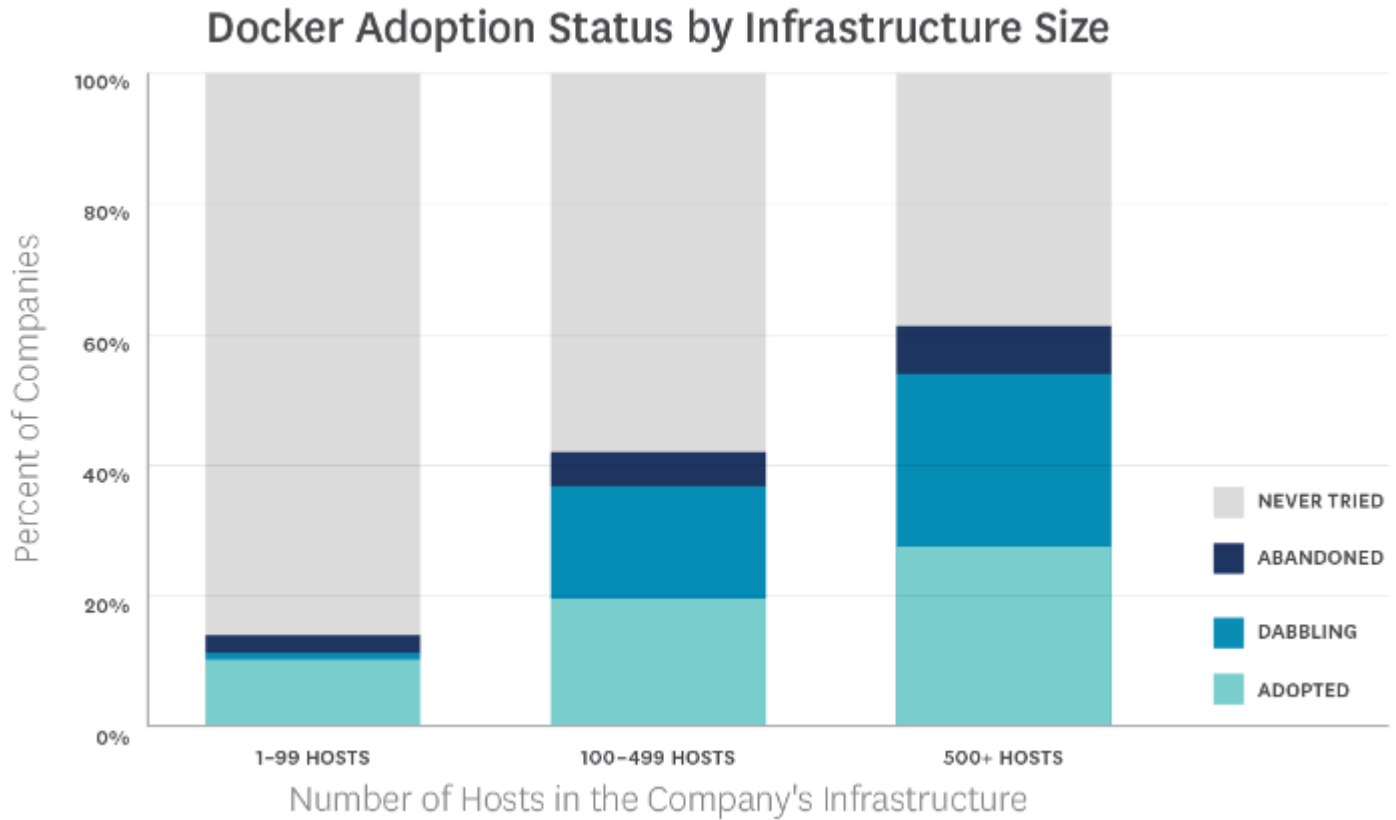
Docker: Containers vs VMs



Docker: Adopción

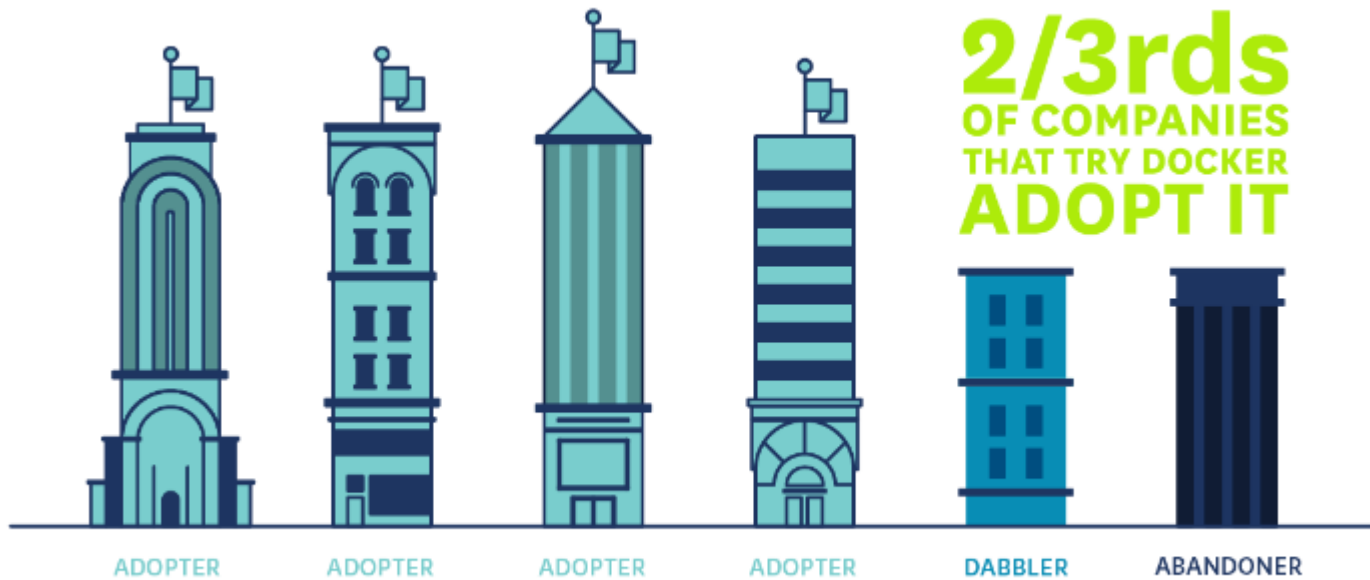


Docker: Adopción



Source: Datadog

Docker: Adopción



Source: Datadog



2

MSSQL Linux + Docker

● MSSQL Linux + Docker



● MSSQL Linux + Docker: Requerimientos

○ Docker Engine 1.8+

Como minimo 3.25 GB de RAM

Variables de ambiente:

- ACCEPT_EULA=Y
- SA_PASSWORD=<YourStrongPassword>

End User License Agreement (EULA)

Politica de Password:

- Longitud de 8 caracteres
- No Alfanumerico (\$, <, >, etc.)
- Mayuscula(s)
- Minuscula(s)
- [0-9]

● MSSQL Linux + Docker: Imágenes

○ Linux

mssql-server-linux:latest (370 MB)

Windows

microsoft/mssql-server-windows (8 GB)

microsoft/mssql-server-windows-express (5 GB)

MSSQL Linux + Docker: Iniciar

1. Descargar la Imagen

```
docker pull microsoft/mssql-server-linux:ctp1-2
```

2. Correr MSSQL

```
docker run -e 'ACCEPT_EULA=Y' -e  
'SA_PASSWORD=YourStrong!Passw0rd' -p 1433:1433  
-d microsoft/mssql-server-linux:ctp1-2
```

```
docker run -e 'ACCEPT_EULA=Y' -e  
'SA_PASSWORD=YourStrong!Passw0rd' -p 1433:1433  
-v <host directory>:/var/opt/mssql -d  
microsoft/mssql-server-linux
```

3. Obtener Docker logs

```
docker logs <ContainerID>
```




3

Casos de Uso



Casos de Uso

1. Prototipos (Rapidez)
2. Ambiente local para desarrolladores (PROD)
3. Ambiente de pruebas locales para DBAs
4. Ensayo de scripts de migracion (versiones)
5. Ensayo para restaurar backups
6. Separar capas de procesamiento y almacenamiento (1:1)
7. Facilidad de migracion a otro ambiente.
Copiar/Pegar archivos y reiniciar el contenedor.
8. Arquitectura de Microservicios
9. Multiples DBs pequeñas



4

Demo



Demo

1. Configurar el ambiente
2. Crear una aplicacion en C# + MSSQL
3. 100x mas rapido con Columnstore
4. Extension mssql en Visual Studio Code

Referencia:

<https://www.microsoft.com/en-us/sql-server/developer-get-started/csharp-ubuntu>

Gracias!

Preguntas?

Twitter: @Christianhxc

Blog: cmelendeztech.com

Github: <https://github.com/christianhxc/sqlsaturday586>