

# Azure hosting for **ASP.Net & SQL on Linux?**

Peter Willmot

[peter@Qri.co.za](mailto:peter@Qri.co.za)

# Before we start .....

Disclaimers and important notices:

All opinions expressed are those of the presenter and/or **Qri (Pty) Ltd**

We are a completely independent entity

We do not represent the interests of any other organisation or product/offering

Any demonstrations we include ...

were designed specifically to illustrate key concepts under discussions

are often dramatically simplified constructs that are often more complex in practice

are not intended for direct deployment in any production context

# Why would we want do this stuff?

Let's pretend that .....

We have an existing ASP.Net **MVC** and/or **webAPI** app

Which we built and tested on Windows server

It is a **low-volume / small-scale** app

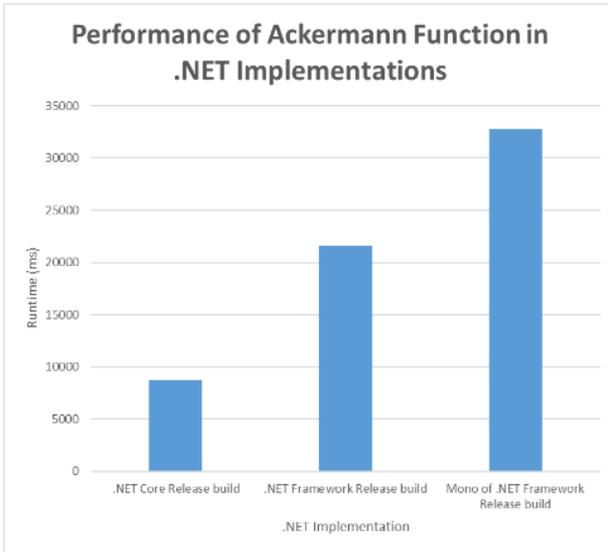
Notably with very constrained budgets (e.g. maybe a startup division or similar)

But it needs to be secure, reliable, etc

We want to **minimise running costs** .....

Hosting **Ubuntu** on **Azure** is ~60% of the fee for hardware equivalent Windows server

# What are our options for .Net?



Source: CODING GORILLA, 2016

	<b>.Net Fx</b> (Windows only)	<b>Mono</b> (Win/Mac/Linux)	<b>.Net Core</b> (Win/Mac/Linux)
<b>ASP.Net</b>	WebForms (ASPX) MVC webAPI Web Pages <b>WCF</b>	WebForms (ASPX) MVC webAPI Web Pages <b>WCF (?)</b>	<b>*** NOPE ***</b> MVC webAPI Web Pages <b>Conn. Svc. only?</b>
Custom Assembly (DLL)	Yes	Yes	Yes
<b>Entity Framework</b>	Yes	Most	<b>EF Core (Basics)</b>
Console App	Yes	Yes	Yes
Windows Service	Yes	<b>daemon</b>	<b>daemon</b>
<b>Windows Forms</b>	Yes	<b>Kinda (sorta)</b>	<b>*** NOPE ***</b>
<b>WPF / WWF</b>	Yes	<b>*** NOPE ***</b>	<b>*** NOPE ***</b>

More info: .NET Core, .NET Framework, Xamarin – “WHAT and WHEN”  
Cesar de la Torre, Microsoft

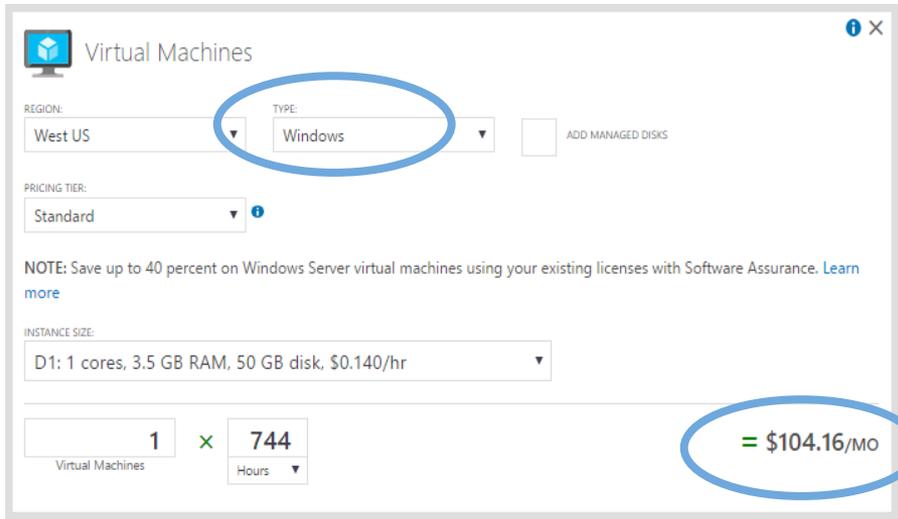
# What about our MS SQL DB if we move to Linux?

As at **CTP 1.4** (March, 2017) on **Ubuntu**:

<b>MS SQL Server Component/Option</b>	<b>Missing</b>
SQL <b>DB</b> Engine  <b>NB:</b> Standardised code base since SQL 2016	<b>Replication</b> Polybase Dist. Query <b>DB Mirror</b> <b>AD/Win AuthN</b> UNSAFE CLR
SQL <b>Management Studio</b>	<b>All (use SQLCMD)</b>
SQL <b>Agent</b>	<b>Some</b>
SQL <b>Reporting</b> Services	<b>All</b>
SQL <b>Analysis</b> Services	<b>All</b>
SQL <b>Integration</b> Services	<b>All</b>
Other (e.g. Master Data, ....)	<b>All</b>

**NB:** license/cost implications  
(expected release late 2017?)  
is **unknown** at present!

# Will the move be worth the effort?



Virtual Machines

REGION: West US

TYPE: Windows

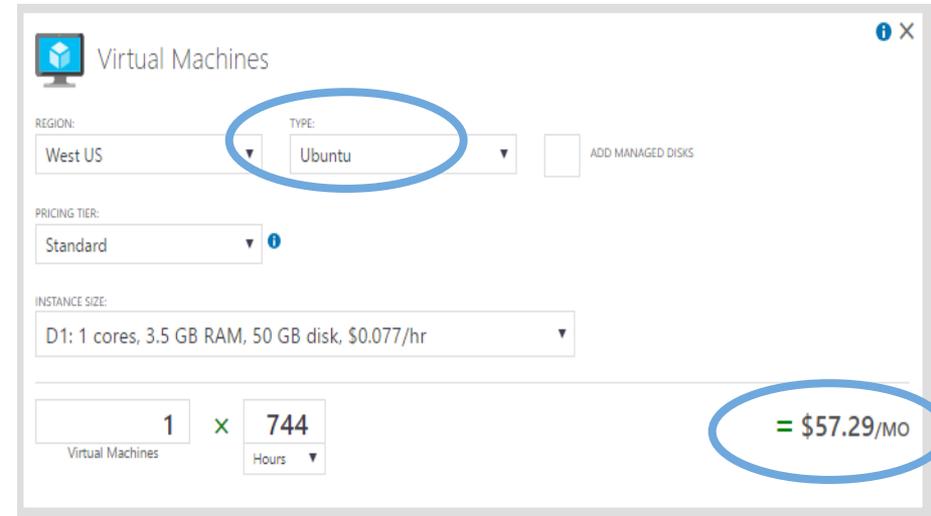
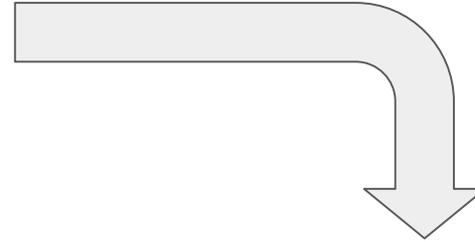
PRICING TIER: Standard

NOTE: Save up to 40 percent on Windows Server virtual machines using your existing licenses with Software Assurance. [Learn more](#)

INSTANCE SIZE: D1: 1 cores, 3.5 GB RAM, 50 GB disk, \$0.140/hr

1 Virtual Machines × 744 Hours = \$104.16/mo

This screenshot shows the configuration for a Windows virtual machine. The 'TYPE' dropdown is set to 'Windows' and is circled in blue. The total cost is calculated as 1 VM multiplied by 744 hours, resulting in a monthly cost of \$104.16, which is also circled in blue.



Virtual Machines

REGION: West US

TYPE: Ubuntu

PRICING TIER: Standard

INSTANCE SIZE: D1: 1 cores, 3.5 GB RAM, 50 GB disk, \$0.077/hr

1 Virtual Machines × 744 Hours = \$57.29/mo

This screenshot shows the configuration for an Ubuntu virtual machine. The 'TYPE' dropdown is set to 'Ubuntu' and is circled in blue. The total cost is calculated as 1 VM multiplied by 744 hours, resulting in a monthly cost of \$57.29, which is also circled in blue.

# Our approach .... assuming we started with .Net Fx

1. Confirm that the **SQL DB** will move
2. Replace **EF6 EDMX** model with **EF Core**
3. **Convert** the ASP.Net app from **.Net Framework** to **.Net Core**

A good starting point:

<https://docs.microsoft.com/en-us/aspnet/core/migration/mvc>

1. Make sure that the **ops** personnel can use the new toolset
2. **Migrate** the converted .Net Core app and SQL DB to the new host platform
3. Bask in glory

# Hosted Ubuntu deployment considerations (1)

## Bye-byte GUI desktop

Connect using **ssh** (on Windows 10 use **bash** otherwise try **PuTTY** or **OpenSSH** ... others)

Azure Ubuntu host enables remote **ssh** by default

## MSSQL

Microsoft recommend a **minimum of 3.25 GB RAM** at present for MS SQL

Use **apt-get** to install **Server** and **Tools**

Use **SQL Server Authentication** instead of Windows Integrated for all access

Provision **Network Interface** for external/remote access to **TDS** port (TCP 1433)

[Qri.co.za/learn](https://qri.co.za/learn) Use **rcp** (or similar) to copy DB backup files to **/var/opt/mssql/data** folder

# Hosted Ubuntu deployment considerations (2)

## .Net Core

```
sudo apt-get install dotnet-dev-1.0.1
```

```
sudo apt-get install npm
```

## Web Server

Enable remote **Port 80** for the Azure **Network Interface**

Internet Information Server (IIS) - does not exist on Ubuntu

```
var host = new WebHostBuilder()  
    .UseKestrel()  
    .UseUrls("http://*:80")  
    .....
```

ASP .Net Core uses **kestrel** as an inbuilt basic web server platform

Defaults to listen on **Port 5000**

You can change **Program.cs**:

# Hosted Ubuntu deployment considerations (3)

## Create and Test a template web app

```
dotnet new web
```

```
dotnet restore
```

```
dotnet run
```

ZIP and migrate your ASP.Net app folders/files

Setup client-side script/content provisioning

```
sudo npm install gulp
```

```
sudo npm install bower
```



Please visit [Qri.co.za/learn](https://Qri.co.za/learn) for more about the training workshops and resources we provide

**Peter Willmot**

[peter@Qri.co.za](mailto:peter@Qri.co.za)