

Instructions to create azure virtual machines from macos

Juan Casado Ballesteros

10/05/2019

Contents

DOWNLOAD AND LOGIN	2
CREATE AN UBUNTU VM	2
Open port 80 for web trafic	2
Intall webserver	2
Cleanup resources	2
Create Windows VM	3
Open port 80 for web trafic	3
Conect to the virtual machine	3
Cleanup Resources	5
Crear un proceso en powershell	5
Crear el proceso	5
Comprobar la creación de la máquina virtual	5
Conexión a la maquina virtual creada	5
Crear un recurso compartido	6
Crear una cuenta de almacenamiento	6
Creación del recurso	7
Conectarnos al recurso compartido desde Windows	7
Visual Studio	8
Escalabilidad horizontal y vertical	8
Creación de la aplicación	8
Subir la aplicación	9
Escalar la aplicación	9
Borrar recursos	10
Power BI con datos en Azure	10

DOWNLOAD AND LOGIN

```
brew update && brew install azure-cli
```

```
az login
```

CREATE AN UBUNTU VM

```
az group create --name vmUbuntu --location northeurope
```

```
az vm create \
  --resource-group vmUbuntu \
  --name vmUbuntu1 \
  --image UbuntuLTS \
  --admin-username juancasado \
  --generate-ssh-keys
```

Al ejecutar este comando debemos guardar la IP pública que se nos genera para usarla posteriormente.

Open port 80 for web trafic

```
az vm open-port --port 80 \
--resource-group vmUbuntu --name vmUbuntu1
```

Intall webserver

```
ssh juancasado@${publicIPUbuntu}
sudo apt-get -y update
sudo apt-get -y install nginx
```

With the public IP of the webserver you will be able to see from the browser the server

Cleanup resources

Esta instrucción está desactualizada en la documentación web, en ella se utiliza el tag -name para indicar el nombre del grupo de recursos, se debe utilizar --resource-group en su lugar

```
az group delete --resource-group vmUbuntu
```



Figure 1: vm_ubuntu_nginx_webserver

Create Windows VM

```
az group create --name vmWindows --location northeurope
```

```
az vm create \
  --resource-group vmWindows \
  --name vmWindows1 \
  --image win2016datacenter \
  --admin-username juancasado \
  --admin-password safePassword123
```

Open port 80 for web traffic

```
az vm open-port --port 80 \
  --resource-group vmWindows --name vmWindows1
```

Tenemos que cuadar la ip que obtengamos al ejecutar este comando.

Conect to the virtual machine

```
#Need to run in PowerShell
mstsc /v:${publicIPWindows}
```

Para hacer el login debemos seleccionar otra cuenta y usar las credenciales que hemos establecido al crear la máquina virtual.

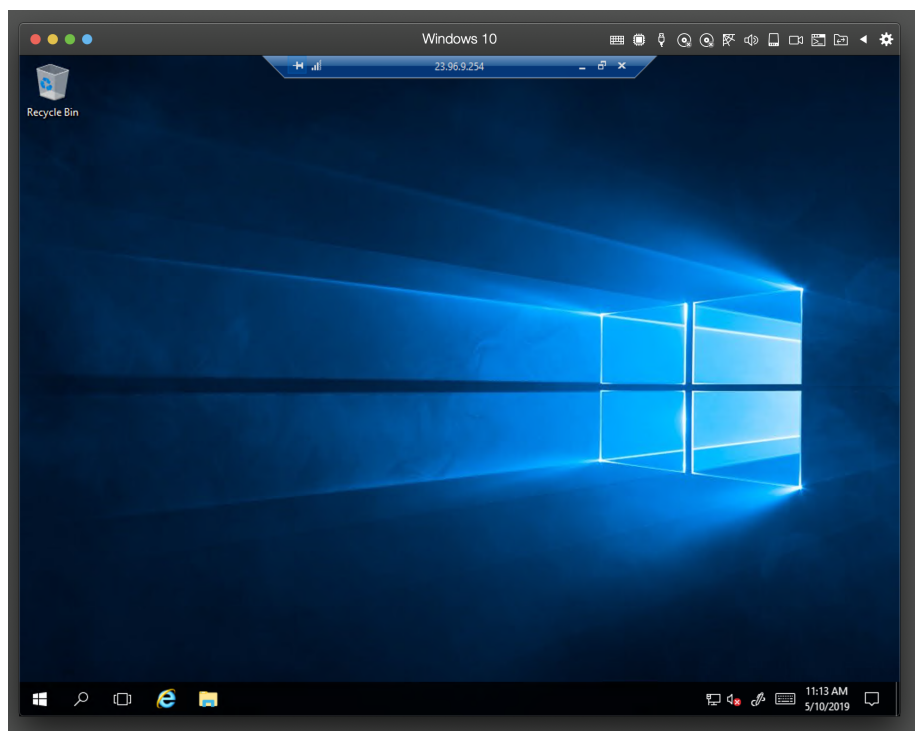


Figure 2: vm_windows_server

Cleanup Resources

```
az group delete --name vmWindows
```

Crear un proceso en powershell

Para ellos desde la página de azure activamos el uso de power shell en la barra superior a la derecha. Se nos solicitará crear un cloud drive para poderla usar lo cual se hará de forma automática.

Crear el proceso

El proceso que vamos a crear desde esta powershell será una máquina virtual windows en azure. Debemos usar una contraseña de más de 8 caracteres con números, mayúsculas y letras y al menos un caracter especial.

```
New-AzResourceGroup -Name powerShellProcess -Location northeurope
$cred = Get-Credential -Message "username and password for the vm"
$vmParams = @{
    ResourceGroupName = 'powerShellProcess'
    Name = 'wVMprocess'
    Location = 'northeurope'
    ImageName = 'Win2016Datacenter'
    PublicIpAddressName = 'wVMprocessIP'
    Credential = $cred
    OpenPorts = 3389
}
$newVM1 = New-AzVM @vmParams
```

Comprobar la creación de la máquina virtual

Podremos ver las características de la vm.

```
$newVM1
$newVM1.OSProfile | Select-Object ComputerName,AdminUserName
$newVM1 | Get-AzNetworkInterface |
Select-Object -ExpandProperty IpConfigurations |
    Select-Object Name,PrivateIpAddress
```

Conexión a la maquina virtual creada

Obtenemos la IP de nuestra máquina virtual

```
$publicIp = Get-AzPublicIpAddress -Name wVMprocessIP |
    -ResourceGroupName powerShellProcess
$publicIp | Select-Object Name,IpAddress,@{label='FQDN';expression={$_.DnsSettings.Fqdn}}
```

Ahora desde el panel de azure buscamos nuestra máquina virtual, la seleccionamos y hacemos click en *connect*. Se nos descargará un archivo **.rdp** que podremos usar para entrar en nuestra máquina virtual con las credenciales que indicamos anteriormente.

```
Your cloud drive has been created in:

Subscription Id: 87bcc685-8894-47ca-b1f6-f1c63d730e07
Resource group:  cloud-shell-storage-westeuropa
Storage account: csb87bcc6858894x47caxb1f
File share:      cs-jcb7777777-gmail-com-1003200044b177df

Initializing your account for Cloud Shell...-
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

MOTD: Download files or directories from the cloudshell: Export-File

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
Azure:/
PS Azure:\> █
```

Figure 3: pws_process

Crear un recurso compartido

Crearemos el recurso desde azure cli.

Crear una cuenta de almacenamiento

Primero deberemos crearnos una cuenta de almacenamiento para poder crear el recurso compartido en ella.

```
az group create \
--name crearRecursoCompartido \
--location westus
az storage account create \
--name cuentarcompartido \
--resource-group crearRecursoCompartido \
```

```
--location westus \
--sku Standard_RAGRS \
--kind StorageV2
```

Creación del recurso

Crearemos el recurso desde azure cli

```
current_env_conn_string=$(az storage account show-connection-string \
-n cuentarcompartido -g crearRecursoCompartido \
--query 'connectionString' -o tsv)
if [[ $current_env_conn_string == "" ]]; then
    echo "Couldn't retrieve the connection string."
fi
az storage share create --name sharedfile --quota 2048 \
--connection-string ${current_env_conn_string}
```

Conectarnos al recurso compartido desde Windows

Montamos el recurso en un disco nuevo de modo que podemos acceder a él creando un mapa de red a la dirección que se nos proporciona al haberlo creado. Para poder acceder a él debemos de estar fuera de la red de eduroam.

```
Test-NetConnection -ComputerName \
cuentarcompartido.file.core.windows.net -Port 445
# Save the password so the drive will persist on reboot
Invoke-Expression -Command "cmdkey /add:cuentarcompartido.file.core.windows.net \
/user:Azure\cuentarcompartido /pass:sxSZrZt1Eizk38a2+oQvaAUV5/zsrD+5YrCfIAW4xWGdXIOWyY/m8hql
# Mount the drive
New-PSDrive -Name A -PSProvider FileSystem\
-Root "\\cuentarcompartido.file.core.windows.net\sharedfile"
```

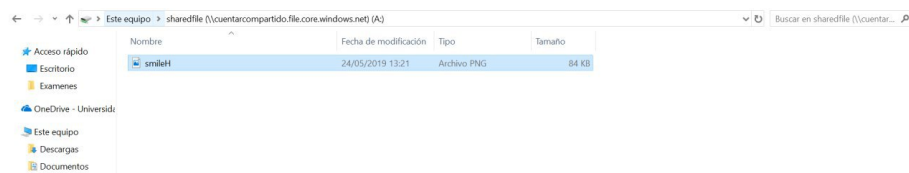


Figure 4: shared_resource

Visual Studio

Necesitamos crear un grupo de recursos con una máquina virtual en él. En la máquina habilitamos los puertos HTTP y RDP para poder acceder. Posteriormente creamos una virtual Network con una IP pública para poder acceder a ella desde fuera.

Una vez la máquina virtual ha sido creado y se ha instalado el visual studio 2017 accedemos a ella por rmd y comprobamos la instalación abriendo visual studio.

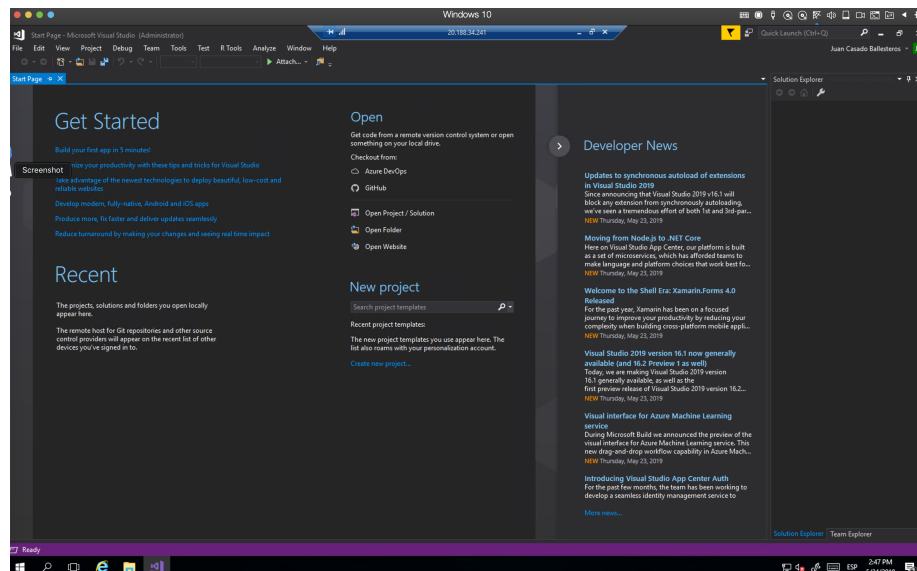


Figure 5: visual_studio

Escalabilidad horizontal y vertical

La realizaremos desde App Services.

Creación de la aplicación

Para ello descargamos el zip con una aplicación de ejemplo desde azure <https://github.com/Azure-Samples/nodejs-docs-hello-world/archive/master.zip> Probamos la aplicación con

```
npm start
```

Posteriormente creamos un .zip con los archivos.


```
zip -r myAppFiles.zip .
```

Subir la aplicación

Creamos un grupo de recursos y un plan desde el que servir la aplicación.

```
az group create --name appScaleResources \  
--location "South Central US"  
az appservice plan create --name appScale \  
--resource-group appScaleResources --sku FREE  
az webapp create --resource-group appScaleResources \  
--plan appScale --name appScaleApp  
az webapp config appsettings set --resource-group appScaleResources \  
--name appScaleApp --settings WEBSITE_NODE_DEFAULT_VERSION=10.14.1
```

Accedemos ahora a <https://appscaleapp.scm.azurewebsites.net/ZipDeployUI> donde podremos subir el zip que hemos creado previamente. Accediendo ahora a <http://appscaleapp.azurewebsites.net> podremos ver nuestra aplicación.

Escalar la aplicación

Podremos acceder ahora desde el panel de azure a las opciones de escalado de nuestra aplicación las cuales estarán restringidas en función del plan que hayamos elegido para ella.

- Scale up realiza el escalado vertical: dar más poder de cómputo a nuestra aplicación.
- Scale out realiza el escalado horizontal: crear más instancias que se balanceen de forma automática a nuestra aplicación.

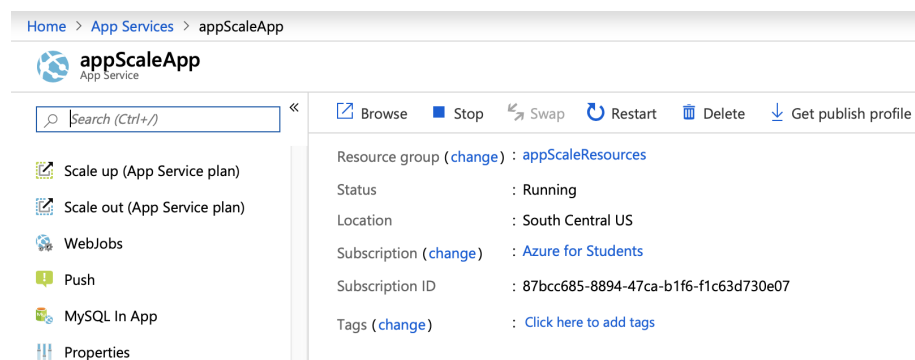


Figure 6: appscale

Borrar recursos

Podremos borrar los recursos que hemos creado borrando el grupo de recursos.

```
az group delete --name appScaleResources
```

Power BI con datos en Azure

Descargamos la aplicación de Power BI desktop para Windows.

Creamos una base de datos SQL desde azure en un nuevo grupo de recursos. Desde power BI nos conectamos al servidor en modo base de datos SQL azure. Nos proporcionará una ip que debemos aplicar a la base de datos en azure.

Tomamos los datos que se crean por defecto en la base de datos para crear una representación gráfica de ellos. En PowerBI seleccionamos algunos de esos datos y con ellos creamos una representación gráfica.

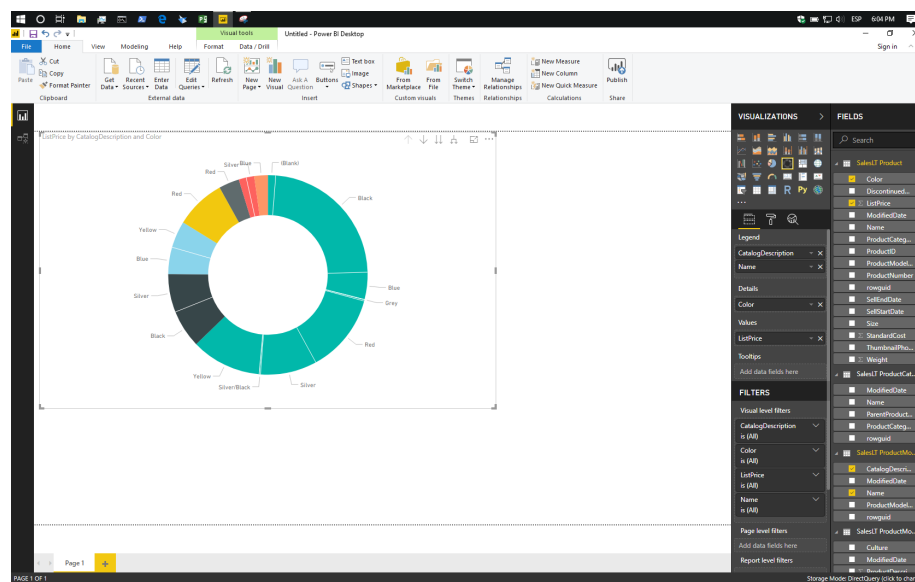


Figure 7: powerbi