



Workshop IT en la Nube
9 de junio de 2015



Andres Felipe Rojas Parra
MVP SharePoint Server
Director de ShareCol

Correo: andres.rojas@geeks.ms

Móvil: +57 300 5906373

Blog: <http://arojaspa.blogspot.com>

Twitter: @arojaspa

YouTube: <http://www.youtube.com/arojaspa>

Correo: sharecol@outlook.com

Url: <http://www.sharecol.com>

Twitter: @sharecol

YouTube: <http://www.youtube.com/sharecol>

Enfoque del workshop

Este workshop está diseñado para aquellos administradores de SharePoint, Office 365 y Azure que en sus tareas de día a día utilizan PowerShell en operaciones de mantenimiento. El asistente aprenderá como instalar y configurar SharePoint 2013, realizar tareas de administración de la granja de servidores, configurar sitios de SharePoint Online de Office 365 y crear ambientes de DEV/QAS/PRD de SharePoint en Azure.

Audiencia

Este workshop está dirigido a administradores de SharePoint OnPremise y SharePoint Online de office 365. Deben tener experiencia en la administración de SharePoint 2013 y SharePoint Online. Adicionalmente, debe tener experiencia en Azure (IaaS y PaaS).

Objetivos

Los objetivos del workshop serán los siguientes:

1. Configurar un ambiente de desarrollo para SharePoint 2013 en Azure
2. Configurar un RunBook en Azure para automatizar el scheduling de activación de las máquinas virtuales del ambiente de desarrollo
3. Instalar y configurar una granja de SharePoint 2013 usando PowerShell
4. Administrar SharePoint 2013 usando PowerShell
5. Realizar tareas de mantenimiento de SharePoint 2013 usando PowerShell
6. Realizar tareas de mantenimiento de SharePoint online de Office 365 usando PowerShell

Prerrequisitos

Para recibir el workshop, el asistente deberá tener:

1. Experiencia trabajando con equipos de escritorio y servidores
2. Experiencia de trabajo en ambientes de directorio activo
3. Experiencia con SharePoint 2013, SharePoint Online y Azure
4. Conocimientos básicos de PowerShell

Software:

1. Licencia trial de Azure
2. Licencia trial de Office 365
3. Licencia trial de SharePoint 2013

AGENDA

1. Configurar un ambiente de desarrollo para SharePoint 2013 en Azure
2. Configurar un RunBook en Azure para automatizar el scheduling de activación de las máquinas virtuales del ambiente de desarrollo
3. Instalar y configurar una granja de SharePoint 2013 usando PowerShell
4. Administrar SharePoint 2013 usando PowerShell
5. Realizar tareas de mantenimiento de SharePoint 2013 usando PowerShell
6. Office 365 PowerShell

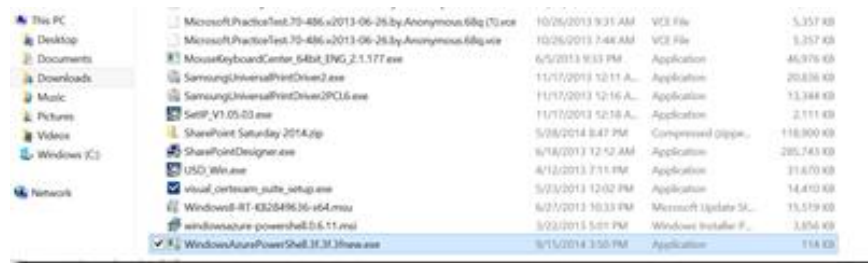
Ejercicio 1

Creación y Configuración del ambiente de desarrollo en Azure para SharePoint 2013 - Configuración de PowerShell para Azure

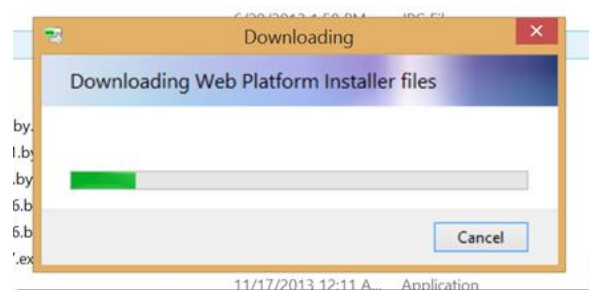
a. Configuración de PowerShell para Azure

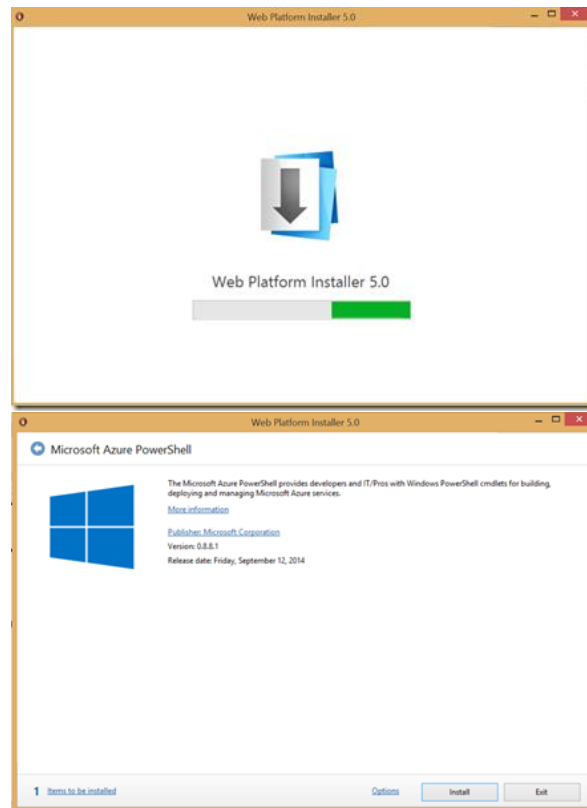
- I. Prerrequisitos de instalación de PowerShell:
 - i. Microsoft .NET Framework 4.5.
 - ii. Tener una suscripción a Microsoft Azure
- II. Instalar PowerShell para Azure

Se debe ejecutar el instalador de Microsoft Web Platform. El instalador lo pueden descargar de la siguiente url <http://go.microsoft.com/fwlink/p/?linkid=320376&clid=0x409>

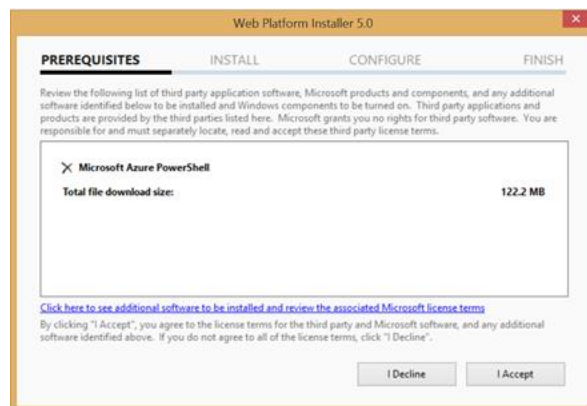


Una vez descargado, damos doble clic para iniciar el proceso.

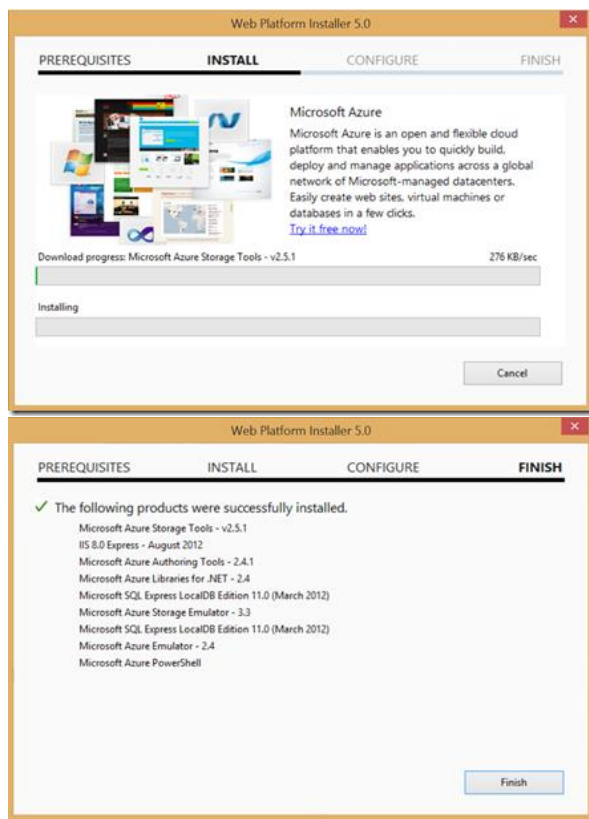




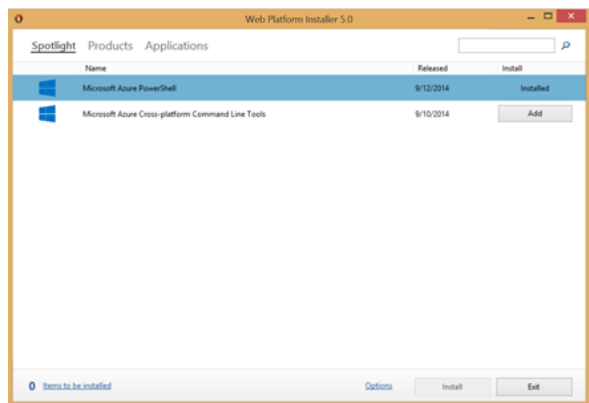
Damos clic en install y procedemos a la instalación de PowerShell para Microsoft Azure



Clic en "I Accept" para continuar.

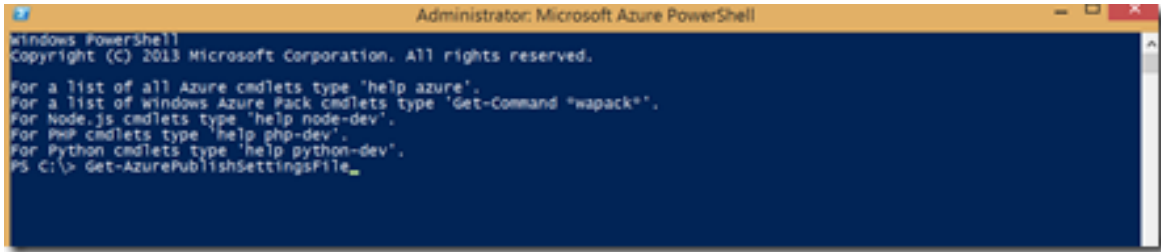


Para finalizar la instalación, damos clic en "Finish".



Una vez instalado el PowerShell para Microsoft Azure, necesitamos descargar la suscripción con la que nos vamos a conectar. Existen dos formas de hacerlo, la primera es accediendo directamente al servicio de directorios de Microsoft Azure con la instrucción "Add-AzureAccount" y algunos cmdlet mas, que a mi parecer es un poco más dispendioso. La segunda opción, es descargando el certificado de publicación de la suscripción. Éste es el proceso que vamos a ejecutar con los siguientes pasos:

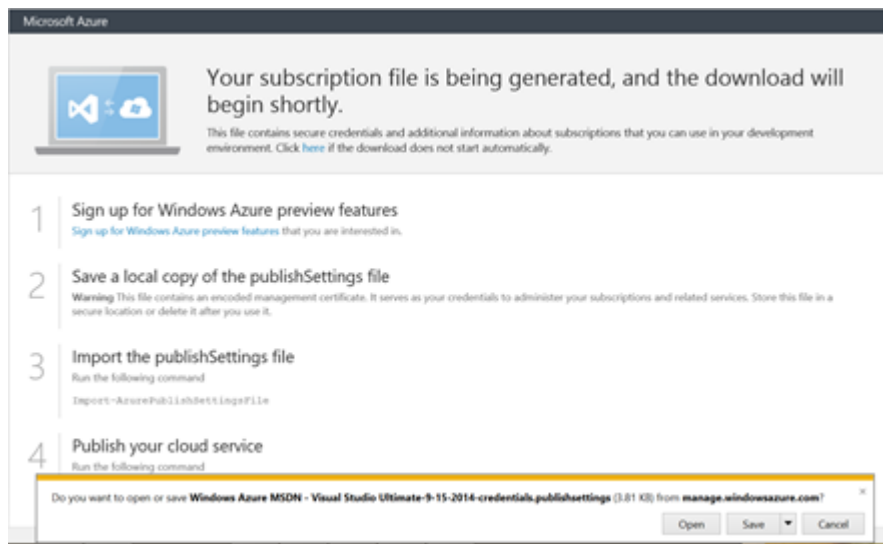
- Hacer login en el portal de administración de Microsoft Azure. Esto se hace para validar credenciales y que la consola de PowerShell para Microsoft Azure, tome dichas credenciales.
- Abrir una consola de PowerShell para Microsoft Azure. Mejor si la ejecutan como **administrador**.
- Sobre la consola de PowerShell para Microsoft Azure, ejecutamos el siguiente cmdlet: **Get-AzurePublishSettingsFile**



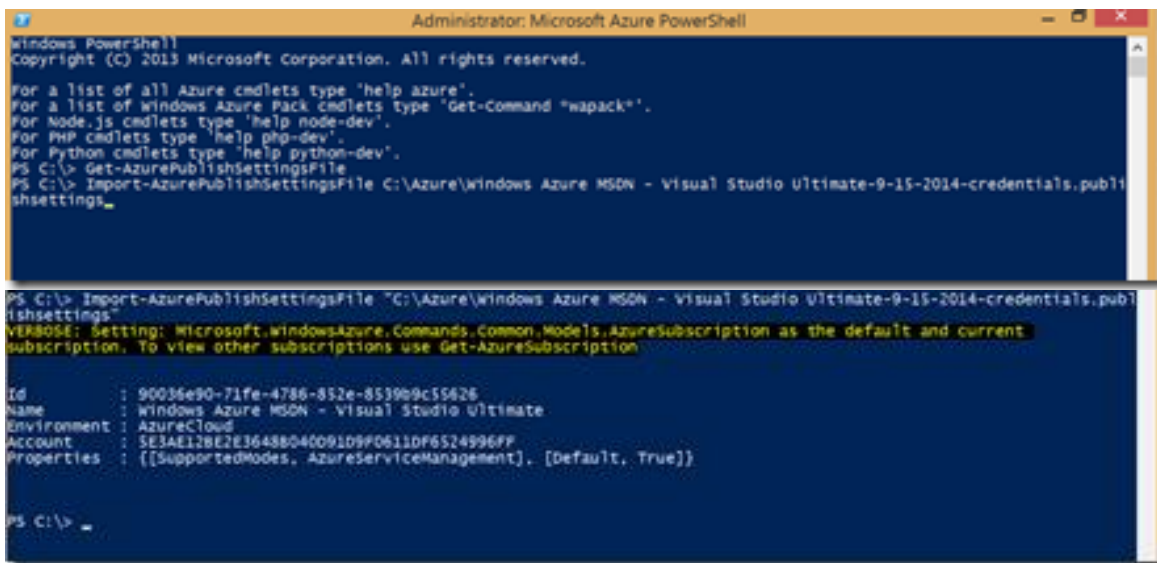
```
Administrator: Microsoft Azure PowerShell
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

For a list of all Azure cmdlets type 'help azure'.
For a list of Windows Azure Pack cmdlets type 'Get-Command *wapack*'.
For Node.js cmdlets type 'help node-dev'.
For PHP cmdlets type 'help php-dev'.
For Python cmdlets type 'help python-dev'.
PS C:\> Get-AzurePublishSettingsFile
```

- Se descarga el certificado en una carpeta del disco duro
"c:\azure\Nombredelcertificado.publishsettings"



- Para usar la información del certificado descargado en el paso anterior, se debe importar para que PowerShell de Microsoft Azure almacene las credenciales de conexión. Para ejecutar lo anterior se usa el cmdlet
- **Import-AzurePublishSettingsFile "c:\azure\nombredelcertificado.publishsettings"**.



```
Administrator: Microsoft Azure PowerShell
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

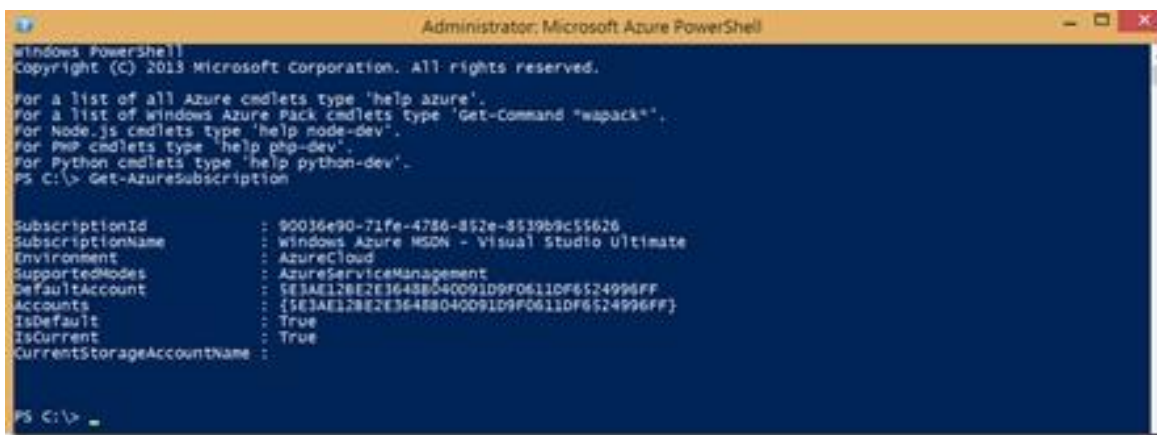
For a list of all Azure cmdlets type 'help azure'.
For a list of Windows Azure Pack cmdlets type 'Get-Command "wapack*"'
For Node.js cmdlets type 'help node-dev'.
For PHP cmdlets type 'help php-dev'.
For Python cmdlets type 'help python-dev'.
PS C:\> Get-AzurePublishSettingsFile
PS C:\> Import-AzurePublishSettingsFile C:\Azure\Windows Azure MSDN - Visual Studio Ultimate-9-15-2014-credentials.publishsettings_

PS C:\> Import-AzurePublishSettingsFile "C:\Azure\Windows Azure MSDN - Visual Studio Ultimate-9-15-2014-credentials.publishsettings"
VERBOSE: Setting: Microsoft.WindowsAzure.Commands.Common.Models.AzureSubscription as the default and current subscription. To view other subscriptions use Get-AzureSubscription

Id          : 90036e90-71fe-4786-852e-8539b9c55626
Name        : Windows Azure MSDN - Visual Studio Ultimate
Environment : AzureCloud
Account     : SE3AE128E2E3648B040D91D9F0611DF6524996FF
Properties  : {[SupportedModes, AzureServiceManagement], [Default, True]}

PS C:\>
```

- Una vez importado, ya tenemos configurado nuestro PowerShell para usar los cmdlets de Microsoft Azure. Para verificar que podemos acceder a nuestra cuenta de Microsoft Azure, cerramos la ventana de PowerShell y la abrimos nuevamente, pero esta vez, ejecutamos los siguientes comandos para verificación de nuestra conexión: **Get-AzureAccount** y **Get-AzureSubscription**. Con el último cmdlet, nos debe salir la siguiente información:



```
Administrator: Microsoft Azure PowerShell
Windows PowerShell
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

For a list of all Azure cmdlets type 'help azure'.
For a list of Windows Azure Pack cmdlets type 'Get-Command "wapack*"'
For Node.js cmdlets type 'help node-dev'.
For PHP cmdlets type 'help php-dev'.
For Python cmdlets type 'help python-dev'.
PS C:\> Get-AzureSubscription

SubscriptionId      : 90036e90-71fe-4786-852e-8539b9c55626
SubscriptionName    : Windows Azure MSDN - Visual Studio Ultimate
Environment         : AzureCloud
SupportedModes      : AzureServiceManagement
DefaultAccount      : SE3AE128E2E3648B040D91D9F0611DF6524996FF
Accounts            : {SE3AE128E2E3648B040D91D9F0611DF6524996FF}
IsDefault           : True
IsCurrent            : True
CurrentStorageAccountName :

PS C:\>
```

- Una vez revisada la instalación, procedemos a abrir una ventana de comandos en modo administrador y ejecutamos la siguiente instrucción para que nos permita ejecutar los scripts desde el ISE:

powershell set-executionpolicy remotesigned

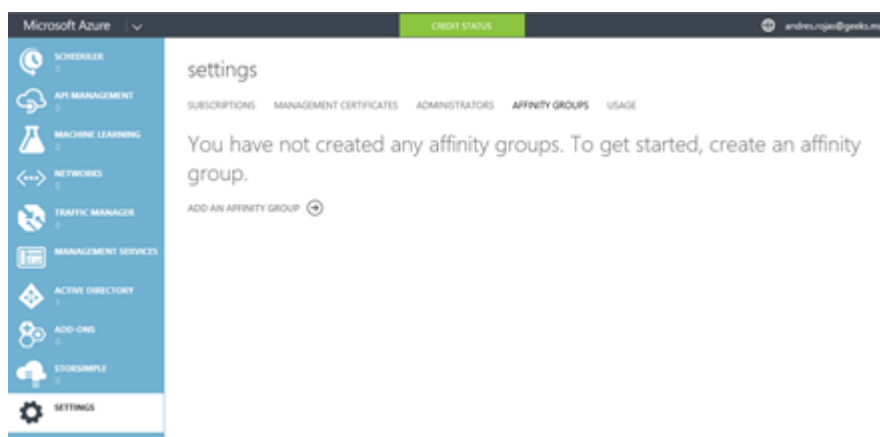
b. Creación de VM y Configuración de directorio activo

Vamos a crear una VM en Microsoft Azure para la configuración de un directorio activo en Windows Server 2012. **No vamos a hacer** uso del Directorio Activo de Microsoft Azure ya que, para instalar SharePoint 2013 en modo granja, puede llegar a ser bastante complicado el intentar usarlo. A continuación veremos cómo podemos crear una VM usando PowerShell e integrarla a la VM con el Directorio Activo que vamos a configurar.

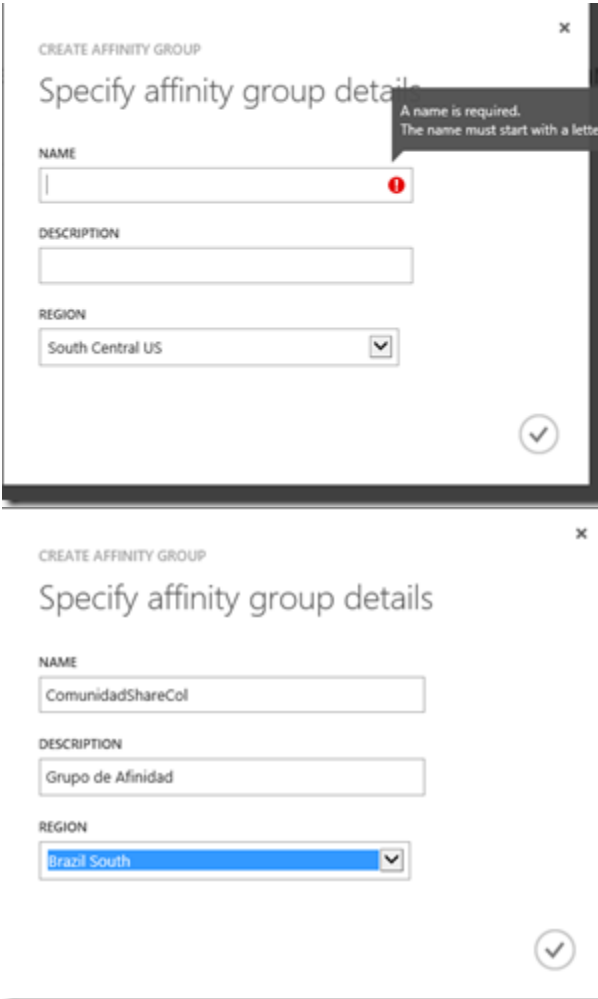
I. Pasos para la creación y configuración de una VM con Windows Server 2012:

Ingresamos a nuestro trial de Azure, sino lo han creado, pueden solicitarlo en la siguiente URL:
<http://azure.microsoft.com/en-us/pricing/free-trial/>

Una vez dentro de nuestro trial, lo primero que debemos crear es un "AFFINITY GROUP". Un Affinity Group es un grupo donde vamos a asociar todas las VM y otros objetos en Azure que hacen parte de este workshop. Para efectos del workshop, vamos a crear un Affinity Group llamado IberianSPConfAF.



Clic en "Add An Affinity Group"



CREATE AFFINITY GROUP

Specify affinity group details

A name is required.
The name must start with a letter

NAME

DESCRIPTION

REGION

South Central US

CREATE AFFINITY GROUP

Specify affinity group details

NAME

ComunidadShareCol

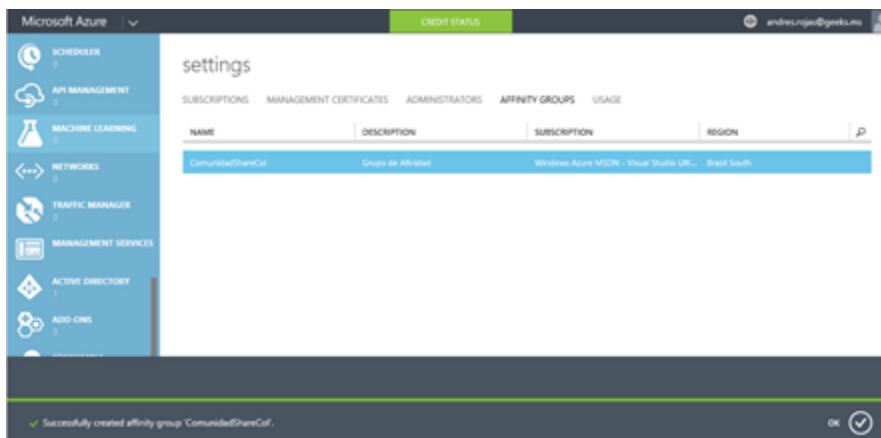
DESCRIPTION

Grupo de Afinidad

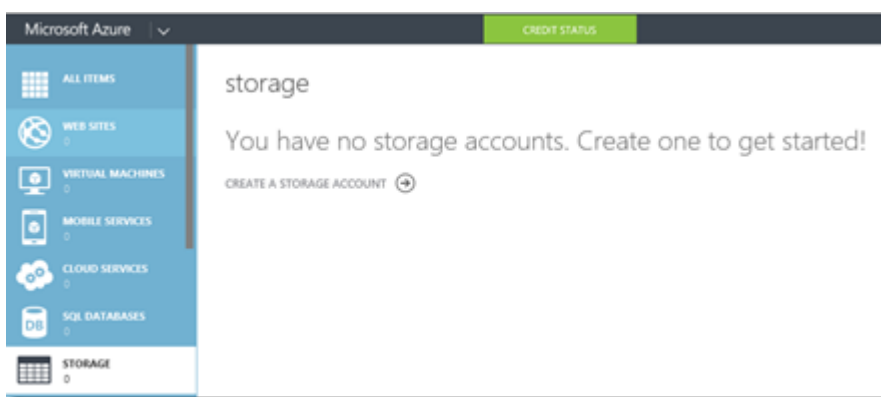
REGION

Brazil South

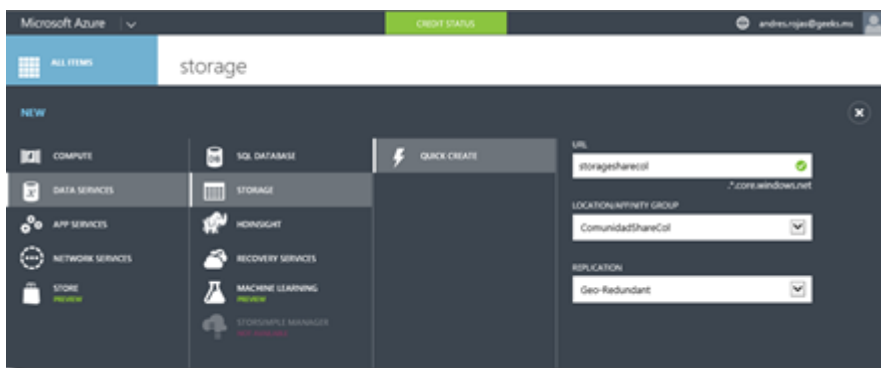
Por último, clic en el botón OK. Después de dar clic, Microsoft Azure iniciará el proceso de creación del "Affinity Group".



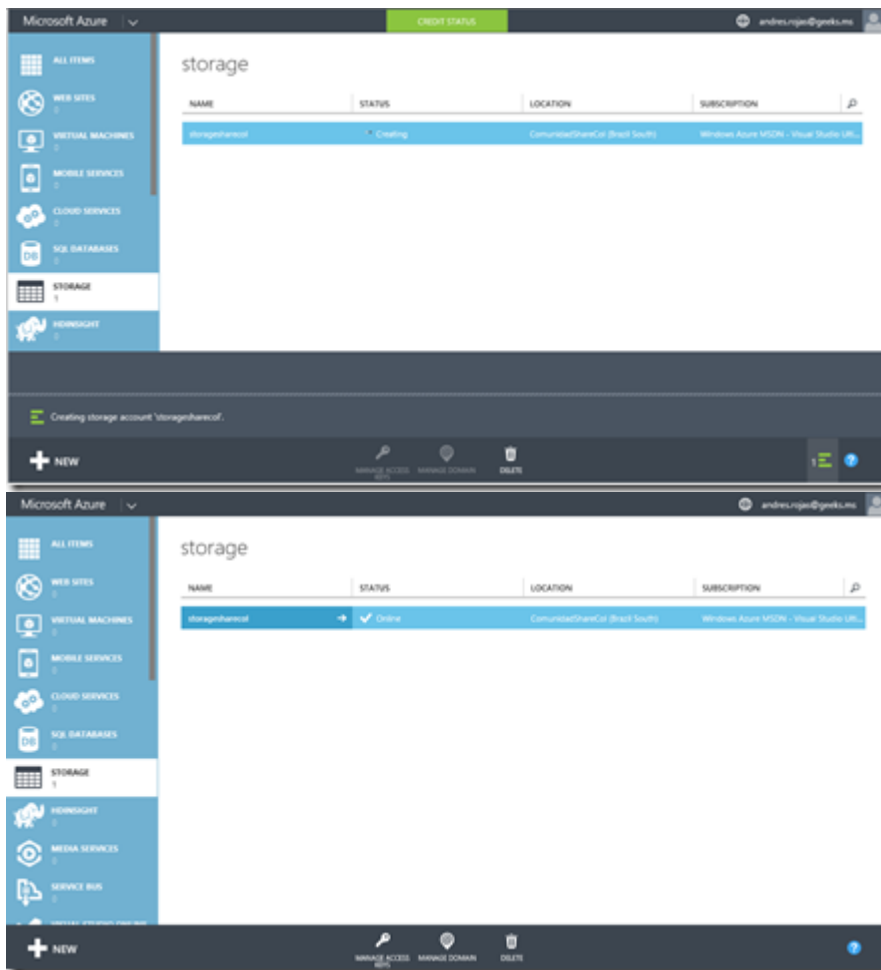
Estando en el portal de Microsoft Azure, si queremos hacer uso de un Storage ya creado en nuestra nube, solo basta buscar el nombre del storage y copiarlo al portapapeles. Para efectos del workshop, procedemos a crear uno donde las VM almacenaran sus discos.



Damos clic en el botón "+New" que se encuentra en la parte inferior y a la izquierda del portal, luego seleccionamos "Data Services\Storage\Quick create". Vamos a llamarlo **"storageiberianspconf"**. Seleccionamos el Affinity Group del paso anterior y seleccionamos que se replique de manera "Geo-Redundant". Luego dar clic en el botón "CREATE STORAGE ACCOUNT".



Microsoft Azure iniciará el proceso de creación del Storage y esperamos a que el storage quede en estado "Online".



Una vez realizados los pasos anteriores, vamos a desarrollar el script de PowerShell con el que vamos a crear la VM que va a operar como Controlador de Dominio. Lo primero que debemos hacer es abrir un notepad o el ISE de PowerShell o cualquier aplicación para crear scripts. Para instalar el ISE, se deben ejecutar los siguientes cmdlets en una ventana de Windows PowerShell:

- **En servidores:**

Import-Module ServerManager
Add-WindowsFeature PowerShell-ISE

- **En Windows 8 o 8.1:**
Seguir las instrucciones del siguiente enlace:
<https://technet.microsoft.com/en-us/library/hh847889.aspx>
- **En Windows 10:**
Por default, ya viene instalado con el sistema operativo. Solo hay que abrir las herramientas administrativas del panel de control y ahí se encuentra el icono de la aplicación.

Una vez instalado el ISE, abrimos una ventana para la creación del script con la Importación de los módulos con los que vamos a realizar la conexión a Microsoft Azure:

```
import-module "C:\Program Files (x86)\Microsoft  
SDKs\Azure\PowerShell\ServiceManagement\Azure\Azure.psd1"  
  
Import-AzurePublishSettingsFile "c:\azure\nombredelcertificado.publishsettings"  
  
Set-AzureSubscription -SubscriptionName "Windows Azure MSDN - Visual Studio  
Ultimate" -CurrentStorageAccountName "storageiberianspconf"  
  
Select-AzureSubscription -SubscriptionName "Windows Azure MSDN - Visual Studio  
Ultimate"
```

La primera línea importa el módulo de Azure para que se puedan usar los cmdlets desde la línea de comando. Sin este módulo, cualquier ejecución de un cmdlet de Microsoft Azure, no será reconocido. La segunda línea importa el certificado que generamos en el ejercicio "a". La tercera línea permite seleccionar nombre a la suscripción de Microsoft Azure que vamos a utilizar, configurando el Storage que vamos a utilizar. Como ya creamos un storage, usamos ese nombre. La cuarta línea nos permite seleccionar la suscripción que vamos a usar durante el proceso de creación de la VM.

Paso siguiente se configura el despliegue del controlador de dominio asignándole una IP fija para que no tome ninguna por defecto de las que genera Microsoft Azure y asignándole un nombre a la VM. Para efectos del workshop la IP será 127.0.0.1 y el nombre de la máquina será "iberianspconfdc01". Hasta el momento solo estamos asignando variables, todavía no vamos a ejecutar ningún cmdlet de creación de VM.

```
#specify the Domain Controller DNS IP to 127.0.0.1 and VM Name  
$myDNS = new-azuredns -Name "IberianSPConfDNS01" -IPAddress "127.0.0.1"  
$vmName = "IberianSPConfDC01"
```

Vamos a tener que solicitar la imagen con la que se instala el sistema operativo al momento de crear la máquina virtual. Para verificar que imagen queremos, hay que ejecutar el siguiente cmdlet en una ventana de PowerShell de azure, que es diferente a la ventana normal de PowerShell:

```
( Get-AzureVMImage | where-object { $_.Label -like "Windows Server 2012*" } )
```

Este comando nos mostrará las imágenes que existan para el sistema operativo. Para efectos del workshop, no usaremos el sistema operativo Windows Server 2012 R2, solo lo dejaremos en Windows Server 2012 Datacenter. Del resultado de la ejecución del cmdlet, copiamos la

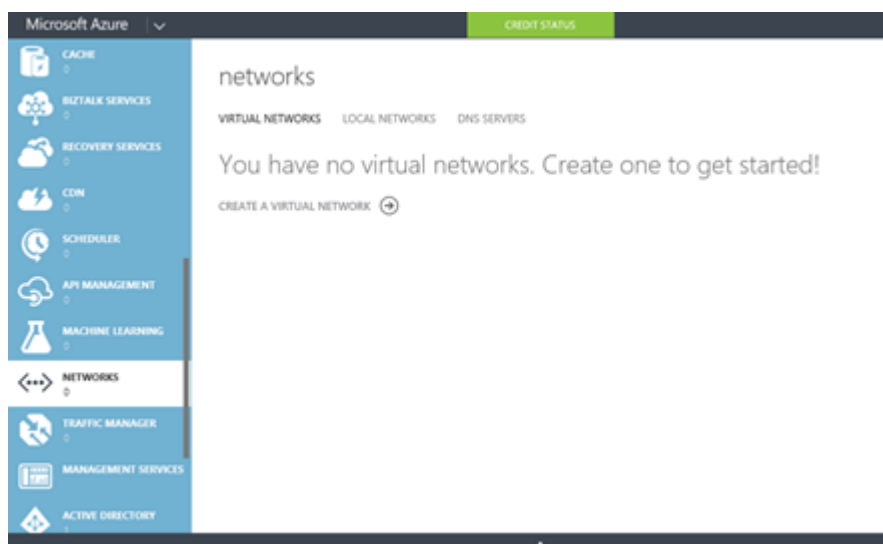
información que está en la misma línea de ImageName y que corresponde a la imagen de sistema operativo que vamos a desplegar con la VM:

"a699494373c04fc0bc8f2bb1389d6106__Windows-Server-2012-Datacenter-201505.01-en.us-127GB.vhd". Si hay una más reciente, por favor utilizar esa.

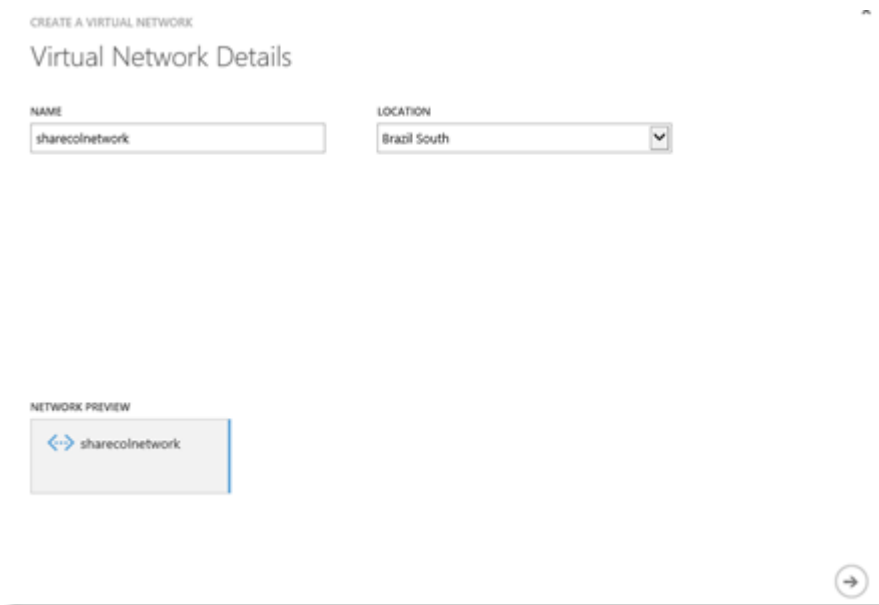
Continuamos el script con la configuración de la imagen de sistema operativo, el servicio de Microsoft Azure que vamos a asignar a la VM, el affinity group y la red que vamos a usar.

```
#OS Image to use
$image = "a699494373c04fc0bc8f2bb1389d6106__windows-server-2012-datacenter-201505.01-en.us-127GB.vhd"
$service = "IberianSPConfDC01Svc"
$ag = "IberianSPConfAF"
```

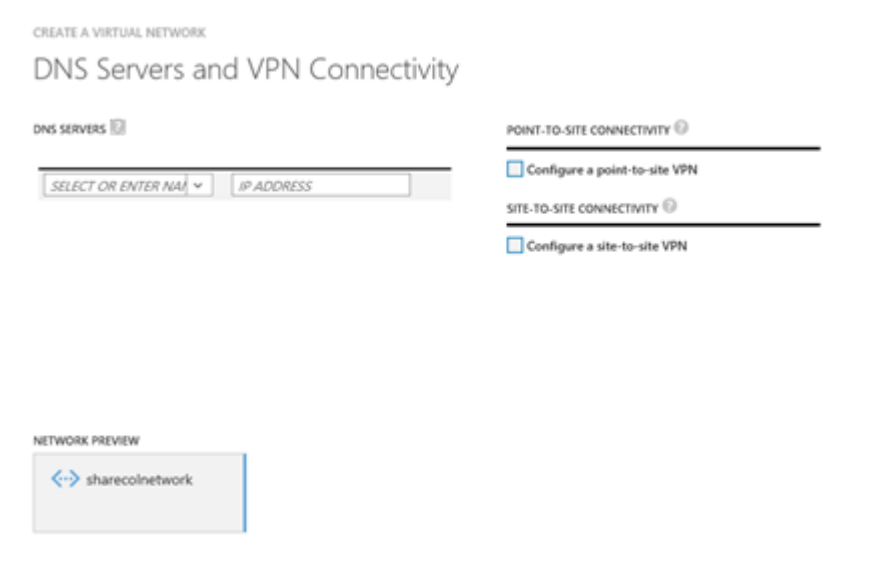
Como no hemos configurado el networking en azure, vamos a crear la configuración red. Dentro del sitio de administración de Microsoft Azure, damos clic en Networks y damos clic en "CREATE A VIRTUAL NETWORK".



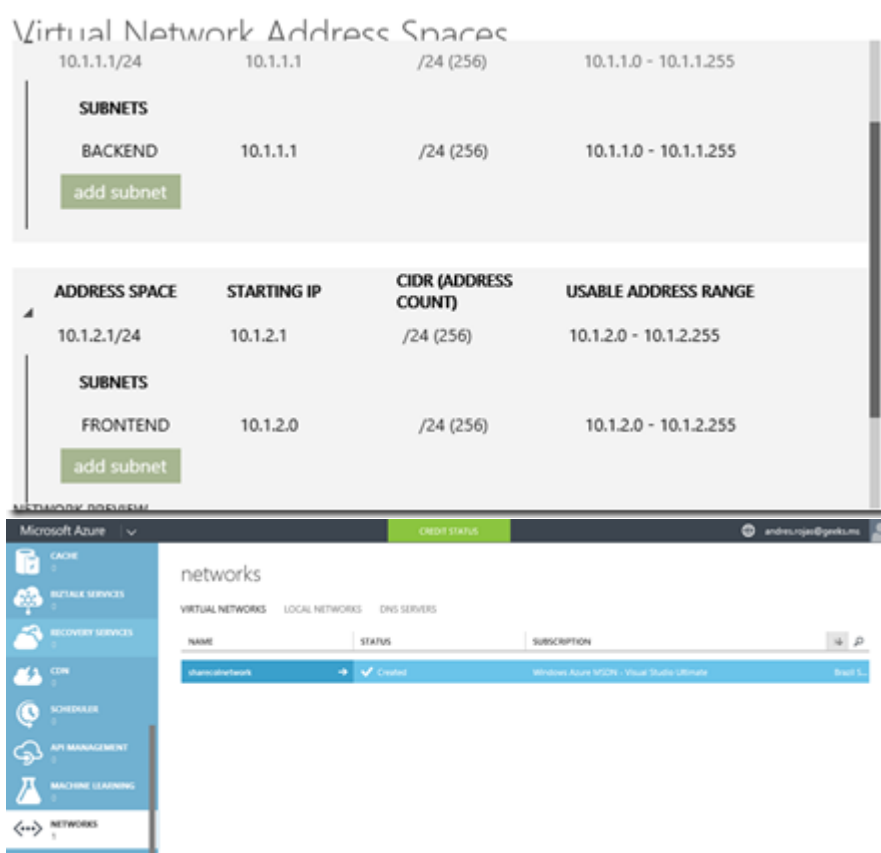
Llenamos los campos solicitados y mantenemos la misma Ubicación del Affinity group. Para efectos del workshop, la red la llamaremos "iberianspconfnet"



Como no vamos a conectar ninguna VLAN de nuestra red corporativa, dejamos los campos de la siguiente pantalla en blanco y damos clic en el botón siguiente.



En la pantalla siguiente, definimos nuestra red principal y subredes de ser el caso. Para efectos del Workshop vamos a crear dos redes, una llamada "BACKEND" y la otra "FRONTEND". Después de crearlas, damos clic en el botón finalizar para que se inicie el proceso de creación de las redes.



Una vez creada las redes, procedemos a complementar el script para powershell asi:

```
$vnet = "iberianspconfnet"
$localadmin = "iberianadmin"
$localadminpass = "iberianspconf123*."
$size = "A5"
$subnetback = "BACKEND"
```

Por último, se ejecuta el cmdlet de creamos la configuración de la VM y luego el cmdlet de creación de la VM en Microsoft Azure. El cmdlet de configuración de la VM requiere ciertos parámetros como el tamaño (RAM, Disco Duro entre otros - <http://msdn.microsoft.com/en-us/library/azure/dn197896.aspx>), el administrador local de Windows, el password del administrador y la subnet a la que va a pertenecer la VM.

Así quedaría el cmdlet de configuración de la VM:

```
$myDC = New-AzureVMConfig -Name $vmName -InstanceSize $size -ImageName $image | Add-
AzureProvisioningConfig -Windows -AdminUsername $localadmin -Password $localadminpass | Set-
AzureSubnet -SubnetNames $subnetback
```

El anterior es el cmdlet de creación de la VM que se está asignando a una variable de PowerShell.

Para ejecutar el cmdlet de creación de la VM se debe tener en cuenta la asignación de las variables creadas con anterioridad en todos los puntos:

New-AzureVM -ServiceName \$service -VMs \$myDC -AffinityGroup \$ag -DnsSettings \$myDNS -VNetName \$vnet

En resumen, así debería operar el script:

- Importar credenciales y el módulo de azure.psd1

```
VERBOSE: Importing alias 'New-IMPackSSNamespace'.
VERBOSE: Importing alias 'New-IMPackWebsite'.
VERBOSE: Importing alias 'Remove-IMPackEnvironment'.
VERBOSE: Importing alias 'Remove-IMPackSSNamespace'.
VERBOSE: Importing alias 'Remove-IMPackSubscription'.
VERBOSE: Importing alias 'Remove-IMPackWebsite'.
VERBOSE: Importing alias 'Restart-IMPackWebsite'.
VERBOSE: Importing alias 'Restart-IMPackWebsiteDeployment'.
VERBOSE: Importing alias 'Save-IMPackWebsiteLog'.
VERBOSE: Importing alias 'Select-IMPackSubscription'.
VERBOSE: Importing alias 'Set-IMPackEnvironment'.
VERBOSE: Importing alias 'Set-IMPackSubscription'.
VERBOSE: Importing alias 'Set-IMPackWebsite'.
VERBOSE: Importing alias 'Show-IMPackPartial'.
VERBOSE: Importing alias 'Show-IMPackWebsite'.
VERBOSE: Importing alias 'Start-CopyAzureStorageBlob'.
VERBOSE: Importing alias 'Start-IMPackWebsite'.
VERBOSE: Importing alias 'Stop-CopyAzureStorageBlob'.
VERBOSE: Importing alias 'Stop-IMPackWebsite'.
VERBOSE: Setting: Microsoft.WindowsAzure.Commands.Common.Models.AzureSubscription as the default and current subscription. To view other subscriptions use Get-AzureSubscription
```

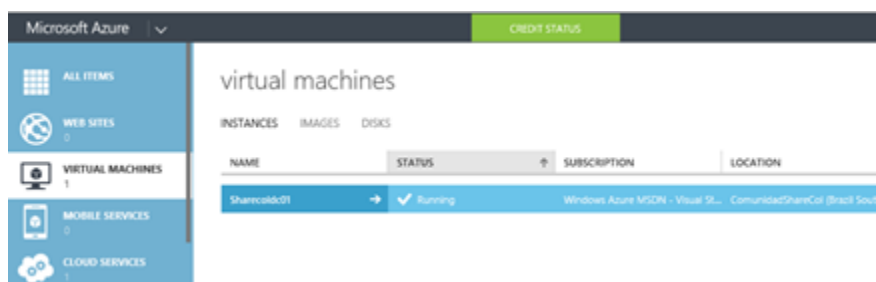
- Configuración y creación de la VM:

```
Id : 90036e90-71fe-4786-852e-8539b9c55626
Name : Windows Azure MSDN - Visual Studio Ultimate
Environment : AzureCloud
Account : [redacted]
Properties : ([Support@Modes, AzureServiceManagement])

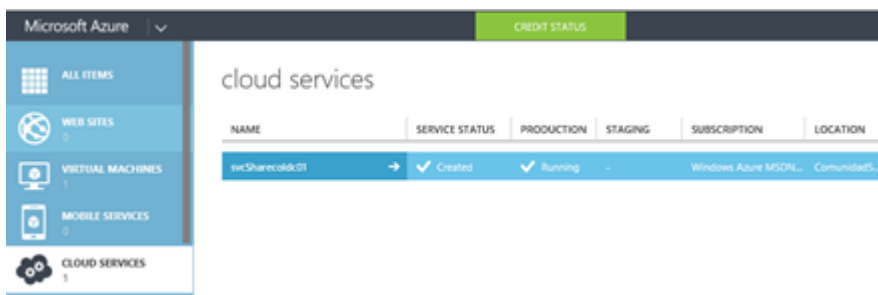
Id : 90036e90-71fe-4786-852e-8539b9c55626
Name : Windows Azure MSDN - Visual Studio Ultimate
Environment : AzureCloud
Account : [redacted]
Properties : ([Support@Modes, AzureServiceManagement, AzureResourceManager], [Tenants,
95af7b71-7137-43b7-99b7-acfd3ab3987], [Default, True], [StorageAccount, storagesharecol])

VERBOSE: 2:18:48 a. m. - Begin Operation: New-AzureVM - Create Deployment with VM Sharecolcd01
VERBOSE: 2:19:54 a. m. - Completed Operation: New-AzureVM - Create Deployment with VM Sharecolcd01
OperationDescription : New-AzureVM
OperationId : a0489c90-e281-17bd-a9ab-4a5b8575fbc8
OperationStatus : Succeeded
```

- Consola Microsoft Azure VM:



- Consola Microsoft Azure Cloud Services:



Una vez creada la máquina virtual, se debe proceder con la configuración del dominio "iberianspconf.local" ([https://technet.microsoft.com/en-us/library/cc755059\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc755059(v=ws.10).aspx))

c. Creación de VM con SharePoint 2013 desde PowerShell para Azure

Con este lab vamos a crear una VM en Microsoft Azure para la configuración de SharePoint Server 2013.

Una vez tenemos todo nuestro ambiente en azure configurado (Redes, Stogare, Affinity Group, VM de Directorio Activo), procederemos a crear la VM de SharePoint Trial que está creada como imagen en azure.

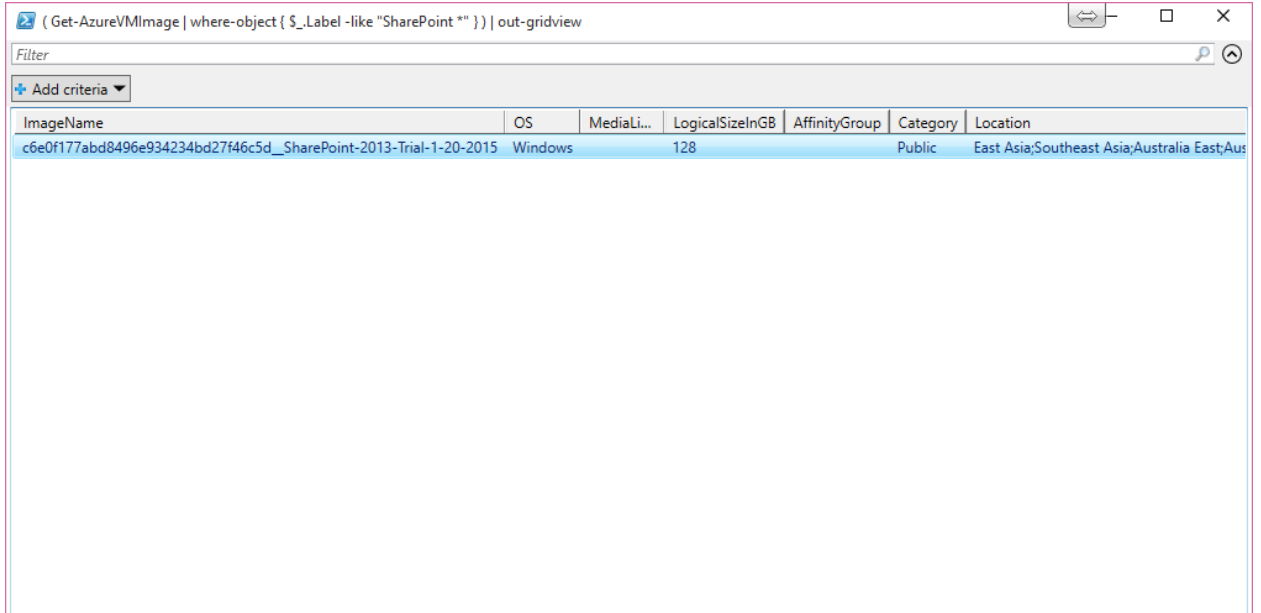
Para buscar la imagen que debemos aplicar a nuestro script, solo basta ejecutar el siguiente cmdlet:

```
( Get-AzureVMImage | where-object { $_.Label -like "SharePoint *" } )
```

Este comando nos devolverá la siguiente información:

```
ImageName      : c6e0f177abd8496e934234bd27f46c5d__SharePoint-2013-Trial-1-20-2015
OS             : Windows
MediaLink      :
LogicalSizeInGB : 128
AffinityGroup  :
Category       : Public
Location       : East Asia;Southeast Asia;Australia East;Australia Southeast;Brazil South;North Europe;West Europe;Japan East;Japan West;Central US;East US;East US 2;North Central US;South Central US;West US
Label          : SharePoint Server 2013 Trial
Description    : Microsoft SharePoint Server 2013 Trial on Windows Server 2012 Datacenter. Virtual Machines created with this trial image will expire on July 18, 2015. This image includes a complete installation of SharePoint Server 2013. Some SharePoint Server 2013 components require additional setup and configuration. You can set-up Active Directory and SQL Server required for your SharePoint farm by provisioning additional virtual machines. Minimum recommended virtual machine size for this image is Large. To evaluate the advanced capabilities of SharePoint Server 2013, we recommend that you use a virtual machine size of A4.
Eula           : http://www.microsoft.com/en-us/download/details.aspx?id=38417
ImageFamily    : Windows Server Datacenter 2012 R2 Datacenter
PublishedDate  : 20/01/2015 9:00:00 a. m.
IsPremium      : False
IconUri        :
SmallIconUri   :
PrivacyUri     :
RecommendedVMSize : Standard_D12
PublisherName  : Microsoft SharePoint Group
IOType         : Standard_LRS
ShowInGui      :
IconName       :
SmallIconName  :
OperationDescription : Get-AzureVMImage
OperationId     : ec50f80b-3bd8-a5e9-9c6c-2025f0105554
OperationStatus : Succeeded
```

Quizás sea un poco confusa la información vista de esa manera, si al final del cmdlet agregamos "`| out-gridview`", quizás lo veamos diferente:

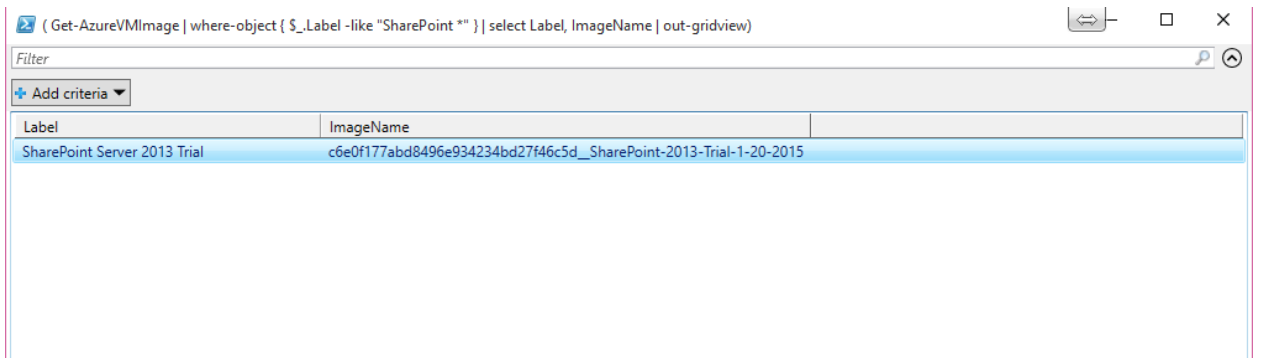


ImageName	OS	MediaLi...	LogicalSizeInGB	AffinityGroup	Category	Location
c6e0f177abd8496e934234bd27f46c5d_SharePoint-2013-Trial-1-20-2015	Windows		128		Public	East Asia;Southeast Asia;Australia East;Aus

Si aun así, sigue siendo confusa por la cantidad de información que nos muestra, lo bueno que tienen los cmdlets es que puedes ir de lo general a lo particular, en este caso, sólo nos interesa el Nombre de la Imagen, la descripción y qué aplicación es. Para este caso, filtramos el cmdlet de la siguiente forma:

(Get-AzureVMImage | where-object { \$_.Label -like "SharePoint *" } | select Label, ImageName | out-gridview)

El resultado es:



Label	ImageName
SharePoint Server 2013 Trial	c6e0f177abd8496e934234bd27f46c5d_SharePoint-2013-Trial-1-20-2015

Una vez tenemos el nombre de la imagen detectada, iniciamos la creación del script de PowerShell.

El primer paso es la creación del script con la Importación de los módulos con los que vamos a realizar la conexión a Microsoft Azure:

```
#import modules to connect azure instance
#-----
Import-module "C:\Program Files (x86)\Microsoft
SDKs\Azure\PowerShell\ServiceManagement\Azure\Azure.psd1"

Import-AzurePublishSettingsFile "c:\azure\windows Azure MSDN - Visual Studio
Ultimate-1-30-2015-credentials.publishsettings"

Set-AzureSubscription -SubscriptionName "windows Azure MSDN - Visual Studio
Ultimate" -CurrentStorageAccountName "storageiberianspconf"

Select-AzureSubscription -SubscriptionName "windows Azure MSDN - Visual Studio
Ultimate"
```

- La primera línea, importa el módulo de Azure para que se puedan usar los cmdlets desde la línea de comando. Sin este módulo, cualquier ejecución de un cmdlet de Microsoft Azure, no será reconocido.
- La segunda línea importa el certificado que generamos en el punto anterior para obtener las credenciales de conexión a nuestra suscripción de Microsoft Azure.
- La tercera línea permite ponerle nombre a la suscripción de Microsoft Azure que vamos a utilizar, configurando el Storage que vamos a utilizar. Como ya creamos un storage, usamos ese nombre.
- La cuarta línea nos permite seleccionar la suscripción que vamos a usar durante el proceso de creación de la VM.

El segundo paso, es la configuración del despliegue de la máquina virtual. Se configura el DNS, en nuestro caso es el Controlador de Dominio creado en el punto anterior, al que se va a asociar esta máquina virtual. El script que vamos a desarrollar, también vinculará la máquina virtual de SharePoint dentro del dominio "iberianspconf.local" que fue configurado como parte del proceso de creación del controlador de dominio. Adicionalmente, también configuramos la variable que contendrá el nombre de la máquina virtual dentro de nuestra infraestructura de directorio activo.

```
#Deploy the virtual machine in a virtual network
#-----
#specify the Domain Controller DNS IP to 10.1.1.4 and VM Name
$myVMDNS = new-azuredns -Name "IberianSPConfDNS01" -IPAddress "10.1.1.4"
$vmName = "IberianSPConfSP01"
```

El tercer paso es la configuración del nombre de la imagen, del affinity group, de la red a la que nos vamos a conectar, el nombre del servicio que administrará la VM en azure, los usuarios administradores local y del dominio, del tamaño de la máquina virtual (para efectos del workshop vamos a usar A5) y por último la configuración del nombre del dominio a donde vamos a vincular la máquina virtual.

De esta forma, esta parte del script queda así:

```
#OS Image to use
$image = "c6e0f177abd8496e934234bd27f46c5d__SharePoint-2013-Trial-1-20-2015"
$service = "IberianSPConfSP01Svc"
$ag = "IberianSPConfAF"
$vnet = "iberianspconfnet"
$adminlocal = "iberianadmin"
$domainuser = "iberianadmin"
$password = "iberianspconf123*."
$size = "A5"
$netbiosname = "iberianspconf"
$domainname = "iberianspconf.local"
$subnetback = "BACKEND"
```

Por último, creamos la variable que tendrá toda la información anterior para realizar la creación de la máquina virtual.

```
$myVM = (New-AzureVMConfig -Name $vmName -InstanceSize $size -ImageName $image
| Add-AzureProvisioningConfig -windowsDomain -AdminUsername $adminlocal -
Password $password -Domain $netbiosname -DomainUserName $domainuser -
DomainPassword $password -JoinDomain $domainname | Set-AzureSubnet -SubnetNames
$subnetback);
```

Una vez tenemos todas las variables listas, la última línea del script, es la que da la orden a PowerShell que inicie la creación de la máquina virtual. Este proceso tardará de 3 a 5 minutos

```
New-AzureVM -ServiceName $service -VMS $myVM -AffinityGroup $ag -DnsSettings
$myVMDNS -VNetName $vnet;
```

El script vincula la máquina virtual automáticamente al dominio.

Esperamos a que la maquina esta up & running y nos conectamos para verificar que todo esté funcionando y que SharePoint 2013 esté instalado. Hay que tener en cuenta que todavía no podemos crear la granja de SharePoint 2013 debido a que no hemos creado la máquina virtual de SQL Server para tal fin. Ese será el siguiente laboratorio del workshop.

d. Creación de VM con SQL Server 2012 SP2 desde PowerShell para Azure

Con este lab vamos a crear una VM en Microsoft Azure para la configuración de SQL Server 2012 SP2.

Para buscar la imagen que debemos aplicar a nuestro script, solo basta ejecutar el siguiente cmdlet:

```
( Get-AzureVMImage | where-object { $_.Label -like "SQL Server 2012 *" } ) | select Label, ImageName | out-gridview
```

Este comando nos devolverá la siguiente información con el listado de todas las imágenes que tiene azure respecto a SQL Server 2012:

```
Administrator: Microsoft Azure PowerShell
OperationId      : e09571e8-2164-a3dd-b3d3-3de00b43a0f0
OperationStatus  : Succeeded
ImageName       : fb83b3509582419d99629ce476bcb5c8_SQL-Server-2012-SP2-11.0.5569.0-OLTP-ENU-Win2012R2-cy15su04
Label           : SQL Server 2012 SP2 Enterprise Optimized for Transactional Workloads on Windows Server 2012 R2
Category        : Public
Description     : This Enterprise Edition image is optimized for OLTP workloads and is intended for VM sizes including A4, A7, A8, A9, D4, D13, D14, G3, G4 and G5. Once deployed, the VM comes with Windows Storage Spaces pre-configured. This image has been pre-configured for Windows Azure, including enabling CEIP which can be disabled, for more info see [here](http://msdn.microsoft.com/en-us/library/windowsazure/dn133151.aspx#database_blank).
OS              : Windows
OSDiskConfiguration : Microsoft.WindowsAzure.Commands.ServiceManagement.Model.OSDiskConfiguration
DataDiskConfigurations : {SQL2012SP2OLTPENUWin2012R2.IaaSVMImageDisk1-1, SQL2012SP2OLTPENUWin2012R2.IaaSVMImageDisk10-1, SQL2012SP2OLTPENUWin2012R2.IaaSVMImageDisk11-1, SQL2012SP2OLTPENUWin2012R2.IaaSVMImageDisk12-1...}
ServiceName     :
DeploymentName  :
RoleName       :
AffinityGroup   :
CreatedTime     : 1/01/1801 1:00:00 a. m.
ModifiedTime    : 1/01/1801 1:00:00 a. m.
Language        : English
ImageFamily     : SQL Server 2012 SP2 Enterprise Optimized for Transactional Workloads on Windows Server 2012 R2
RecommendedVMSize : A7
IsPremium       : True
Eula            : http://go.microsoft.com/fwlink/?LinkID=285687
IconUri         : SqlServer2012_100.png
SmallIconUri    : SqlServer2012_45.png
PrivacyUri      : http://www.microsoft.com/en-us/download/details.aspx?id=29067
PublisherName   : Microsoft SQL Server Group
PublishedDate   : 15/04/2015 2:00:00 a. m.
ShowInGui       : True
PricingDetailLink : http://go.microsoft.com/fwlink/?LinkID=271259
MediaLink       :
LogicalSizeInGB :
Location        : East Asia;Southeast Asia;Australia East;Australia Southeast;Brazil South;North Europe;West Europe;Japan East;Japan West;Central US;East US 2;North Central US;South Central US;West US
IOType          :
IconName        : SqlServer2012_100.png
SmallIconName   : SqlServer2012_45.png
OperationDescription : Get-AzureVMImage
OperationId      : e09571e8-2164-a3dd-b3d3-3de00b43a0f0
OperationStatus  : Succeeded
```

Quizás sea un poco confusa la información vista de esa manera, si al final del cmdlet agregamos “| out-gridview”, lo vemos diferente:

(Get-AzureVMImage | where-object { \$_.Label -like "SQL Server 2012 SP2 Enterprise *" }) | out-gridview

ImageName	OS	MediaL...	LogicalSizeInGB	AffinityGroup	Category	Location
fb83b3509582419d99629ce476bcb5c8_SQL-Server-2012-SP2-11.0.5537.0-Ent-ENU-Win2012-cy14su08	Windows		127		Public	East Asia/Si
fb83b3509582419d99629ce476bcb5c8_SQL-Server-2012-SP2-11.0.5548.0-Ent-ENU-Win2012R2-cy14...	Windows		127		Public	East Asia/Si
fb83b3509582419d99629ce476bcb5c8_SQL-Server-2012-SP2-11.0.5569.0-Ent-ENU-Win2012-cy15su02	Windows		127		Public	East Asia/Si
fb83b3509582419d99629ce476bcb5c8_SQL-Server-2012-SP2-11.0.5569.0-Ent-ENU-Win2012-cy15su04	Windows		127		Public	East Asia/Si
fb83b3509582419d99629ce476bcb5c8_SQL-Server-2012-SP2-11.0.5569.0-Ent-ENU-Win2012R2-cy15...	Windows		127		Public	East Asia/Si
fb83b3509582419d99629ce476bcb5c8_SQL-Server-2012-SP2-11.0.5537.0-DW-ENU-Win2012-cy14su...	Windows		127		Public	East Asia/Si
fb83b3509582419d99629ce476bcb5c8_SQL-Server-2012-SP2-11.0.5537.0-OLTP-ENU-Win2012-cy14su...	Windows		127		Public	East Asia/Si
fb83b3509582419d99629ce476bcb5c8_SQL-Server-2012-SP2-11.0.5548.0-DW-ENU-Win2012R2-cy1...	Windows		127		Public	East Asia/Si
fb83b3509582419d99629ce476bcb5c8_SQL-Server-2012-SP2-11.0.5569.0-DW-ENU-Win2012-cy15su...	Windows		127		Public	East Asia/Si
fb83b3509582419d99629ce476bcb5c8_SQL-Server-2012-SP2-11.0.5569.0-DW-ENU-Win2012-cy15su...	Windows		127		Public	East Asia/Si
fb83b3509582419d99629ce476bcb5c8_SQL-Server-2012-SP2-11.0.5569.0-DW-ENU-Win2012-cy15su...	Windows		127		Public	East Asia/Si
fb83b3509582419d99629ce476bcb5c8_SQL-Server-2012-SP2-11.0.5569.0-OLTP-ENU-Win2012-cy15s...	Windows		127		Public	East Asia/Si
fb83b3509582419d99629ce476bcb5c8_SQL-Server-2012-SP2-11.0.5569.0-OLTP-ENU-Win2012R2-cy...	Windows		127		Public	East Asia/Si

Si aún así, sigue siendo confuso por la cantidad de información que nos muestra, lo bueno que tienen los cmdlets es que puedes ir de lo general a lo particular, en este caso, sólo nos interesa el Nombre de la Imagen, la fecha de publicación y que aplicación es. Para este caso, filtramos el cmdlet de la siguiente forma:

**(Get-AzureVMImage | where-object { \$_.Label -like "SQL Server 2012 *" } | select Label, ImageName
PublishedDate | out-gridview)**

El resultado es:

Filter

Get-AzureVMImage | where-object { \$_.Label -like "SQL Server 2012 Enterprise *" } | select Label, ImageName, PublishedDate | out-gridview

Label	ImageName	PublishedDate
SQL Server 2012 SP2 Enterprise on Windows Server 2012	fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5537.0-Ent-ENU-Win2012-cy14su08	31/08/2014 9:00:00 a. m.
SQL Server 2012 SP2 Enterprise on Windows Server 20...	fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5548.0-Ent-ENU-Win2012R2-cy14...	1/12/2014 9:00:00 a. m.
SQL Server 2012 SP2 Enterprise on Windows Server 2012	fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5569.0-Ent-ENU-Win2012-cy15su02	10/03/2015 8:00:00 a. m.
SQL Server 2012 SP2 Enterprise on Windows Server 20...	fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5569.0-Ent-ENU-Win2012-cy15su04	15/04/2015 9:00:00 a. m.
SQL Server 2012 SP2 Enterprise on Windows Server 20...	fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5569.0-Ent-ENU-Win2012R2-cy15...	11/09/2014 9:00:00 a. m.
SQL Server 2012 SP2 Enterprise Optimized for DataWar...	fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5537.0-DW-ENU-Win2012-cy14su...	11/09/2014 9:00:00 a. m.
SQL Server 2012 SP2 Enterprise Optimized for Transact...	fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5537.0-OLTP-ENU-Win2012-cy14su...	11/09/2014 9:00:00 a. m.
SQL Server 2012 SP2 Enterprise Optimized for DataWar...	fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5548.0-DW-ENU-Win2012R2-cy1...	1/12/2014 1:00:00 a. m.
SQL Server 2012 SP2 Enterprise Optimized for Transact...	fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5548.0-OLTP-ENU-Win2012R2-cy...	1/12/2014 1:00:00 a. m.
SQL Server 2012 SP2 Enterprise Optimized for DataWar...	fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5569.0-DW-ENU-Win2012-cy15su...	10/03/2015 8:00:00 a. m.
SQL Server 2012 SP2 Enterprise Optimized for DataWar...	fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5569.0-DW-ENU-Win2012-cy15su...	15/04/2015 2:00:00 a. m.
SQL Server 2012 SP2 Enterprise Optimized for DataWar...	fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5569.0-DW-ENU-Win2012R2-cy1...	15/04/2015 2:00:00 a. m.
SQL Server 2012 SP2 Enterprise Optimized for Transact...	fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5569.0-OLTP-ENU-Win2012-cy15...	15/04/2015 2:00:00 a. m.
SQL Server 2012 SP2 Enterprise Optimized for Transact...	fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5569.0-OLTP-ENU-Win2012R2-cy...	15/04/2015 2:00:00 a. m.

Ok, una vez tenemos el nombre de la imagen detectada, iniciamos la creación del script de powershell. Para nuestro caso, la imagen que vamos a usar es:

"fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5569.0-Ent-ENU-Win2012-cy15su04"

SQL Server 2012 SP2 Enterprise on Windows Server 2012

El primer paso es la creación del script con la Importación de los módulos con los que vamos a realizar la conexión a Microsoft Azure:

```
#import modules to connect azure instance
#-----
import-module "C:\Program Files (x86)\Microsoft
SDKs\Azure\Powershell\ServiceManagement\Azure\Azure.psd1"

Import-AzurePublishSettingsFile
"D:\MVPEvents\IberianSharePointConference2015\workshop\Desarrollo-pruebas de
MSDN - Pago por uso-Windows Azure MSDN - Visual Studio Ultimate-6-7-2015-
credentials.publishsettings"

Set-AzureSubscription -SubscriptionName "windows Azure MSDN - Visual Studio
Ultimate" -CurrentStorageAccountName "storageiberianspconf"

Select-AzureSubscription -SubscriptionName "windows Azure MSDN - Visual Studio
Ultimate"
```

- La primera línea, importa el módulo de Azure para que se puedan usar los cmdlets desde la línea de comando. Sin este módulo, cualquier ejecución de un cmdlet de Microsoft Azure, no será reconocido.

- La segunda línea importa el certificado que generamos en el punto anterior para obtener las credenciales de conexión a nuestra suscripción de Microsoft Azure.
- La tercera línea permite ponerle nombre a la suscripción de Microsoft Azure que vamos a utilizar, configurando el Storage que vamos a utilizar. Como ya creamos un storage, usamos ese nombre.
- La cuarta línea nos permite seleccionar la suscripción que vamos a usar durante el proceso de creación de la VM.

El segundo paso, es la configuración del despliegue de la máquina virtual. Se configura el DNS, en nuestro caso es el Controlador de Dominio creado en el punto anterior, al que se va a asociar esta máquina virtual. El script que vamos a desarrollar, también vinculará la máquina virtual de SQL Server 2012 SP2 dentro del dominio "iberianspconf.local" que fue configurado como parte del proceso de creación del controlador de dominio. Adicionalmente, también configuramos la variable que contendrá el nombre de la máquina virtual dentro de nuestra infraestructura de directorio activo.

```
#specify the Domain Controller DNS IP to 10.1.1.4 and VM Name
$myVMDNS = new-azuredns -Name "IberianSPConfDNS01" -IPAddress "10.1.1.4"
$vmName = "IberianSPConfSQL01"
```

El tercer paso es la configuración del nombre de la imagen, del affinity group, de la red a la que nos vamos a conectar, el nombre del servicio que administrará la VM en azure, los usuarios administradores local y del dominio, del tamaño de la máquina virtual (para efectos del workshop vamos a usar A5) y por último la configuración del nombre del dominio a donde vamos a vincular la máquina virtual.

De esta forma, esta parte del script queda así:

```
#OS Image to use
$image = "fb83b3509582419d99629ce476bcb5c8__SQL-Server-2012-SP2-11.0.5569.0-Ent-ENU-win2012-cy15su04"
$service = "IberianSPConfSQL01Svc"
$ag = "IberianSPConfAF"
$vnet = "iberianspconfnet"
$adminlocal = "iberianadmin"
$domainuser = "iberianadmin"
$password = "iberianspconf123*."
$size = "A5"
$netbiosname = "iberianspconf"
$domainname = "iberianspconf.local"
$subnetback = "BACKEND"
```

Por último, creamos la variable que tendrá toda la información anterior para realizar la creación de la máquina virtual.

```
$myVM = (New-AzureVMConfig -Name $vmName -InstanceSize $size -ImageName $image |
Add-AzureProvisioningConfig -windowsDomain -AdminUsername $adminlocal -
Password $password -Domain $netbiosname -DomainUserName $domainuser -
DomainPassword $password -JoinDomain $domainname | Set-AzureSubnet -SubnetNames
$subnetback);
```


Una vez tenemos todas las variables listas, la última línea del script, es la que da la orden a PowerShell que inicie la creación de la máquina virtual. Este proceso tardará de 3 a 5 minutos

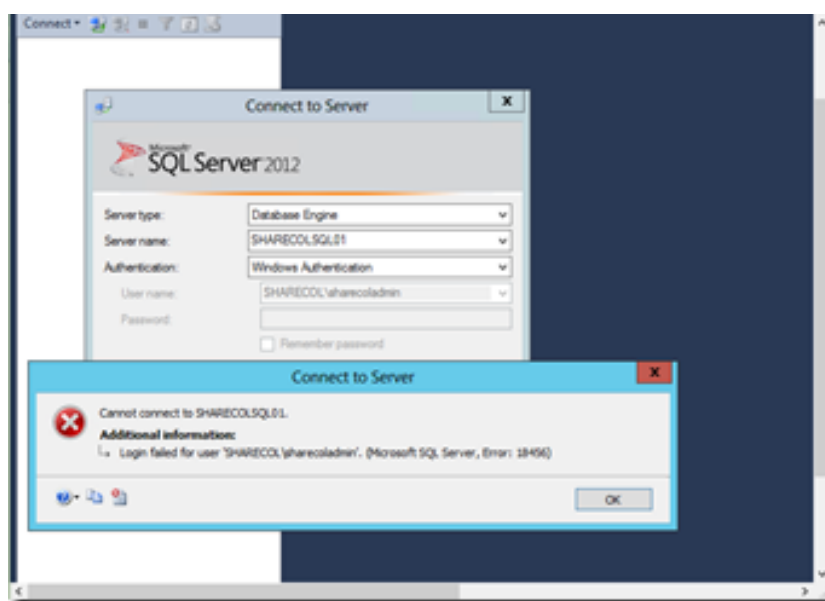
```
New-AzureVM -ServiceName $service -VMs $myVM -AffinityGroup $ag -DnsSettings  
$myVMDNS -VNetName $vnet;
```

El script vincula la máquina virtual automáticamente al dominio.

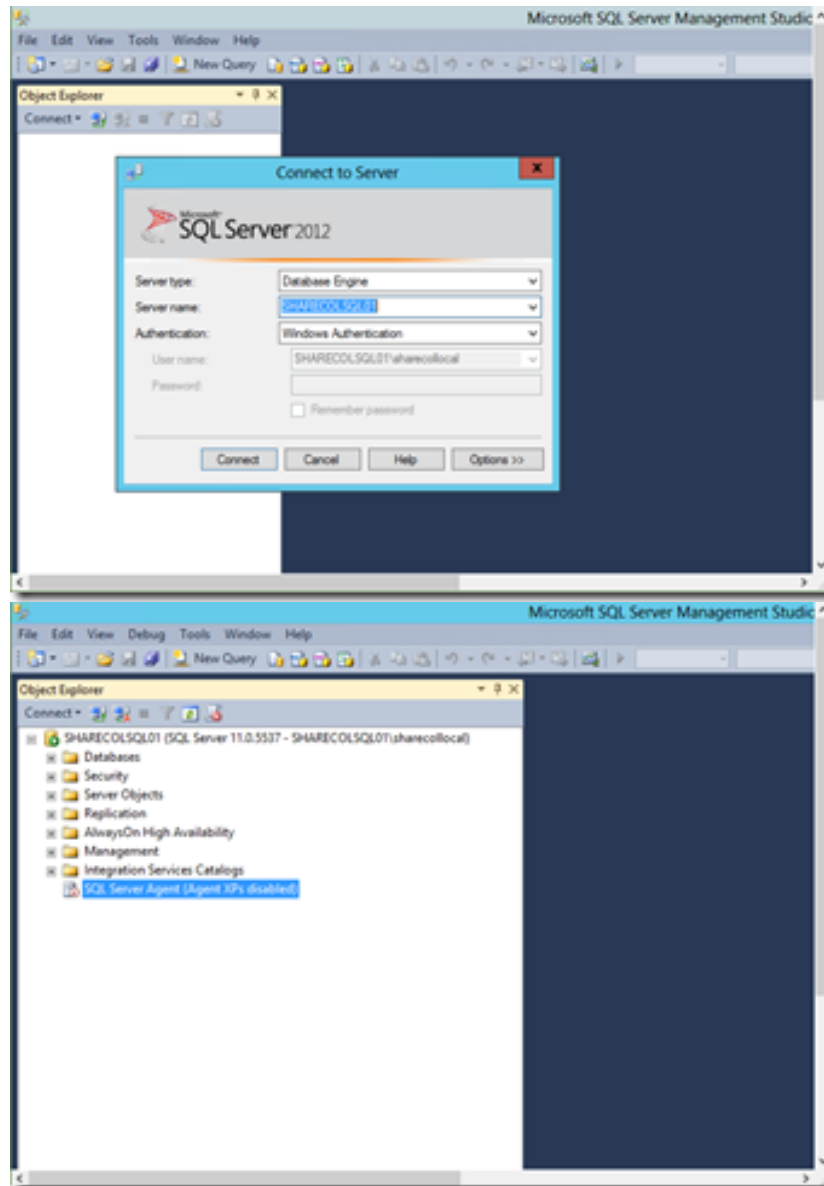
Esperamos a que la maquina esta up & running y nos conectamos para verificar que todo esté funcionando. y que SQL Server 2012. Verificamos que SQL Server 2012 SP2 esté correctamente instalado, como la instancia de SQL Server 2012 SP2 se instala en modo stand alone, debemos entrar al servidor con el usuario local de la máquina y configurar los usuarios de dominio que hayamos creado para ser sysadmin. SharePoint Requiere que los usuarios administradores de la granja y de sql server sean sysadmin.

Veamos el proceso:

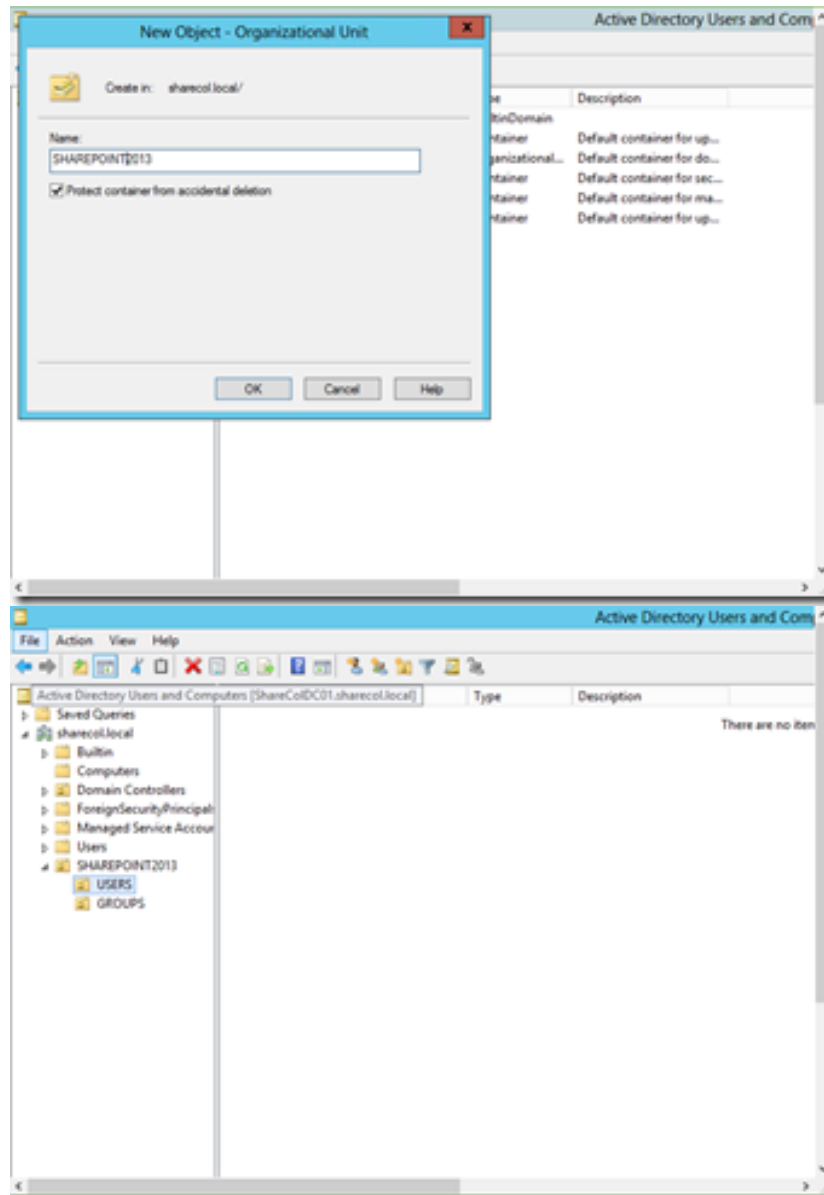
- Cuando ingreso en modo usuario del directorio activo, esto es lo que pasa:



- Ingresamos en modo usuario local, y la consola abre normalmente:



- Ahora procederemos a crear los usuarios que vamos a necesitar para la configuración de SharePoint 2013. Entramos a la VM donde está configurado el Controlador de Domino y abrimos la consola "Active Directory Users and Computers" y creamos una OU (Organizational Unit) en la raíz del AD (Active Directory). Para efectos del Workshop vamos a llamarla "SHAREPOINT2013". Dentro de la OU, crearemos dos OU más, una que se llame USERS y otra GROUPS.

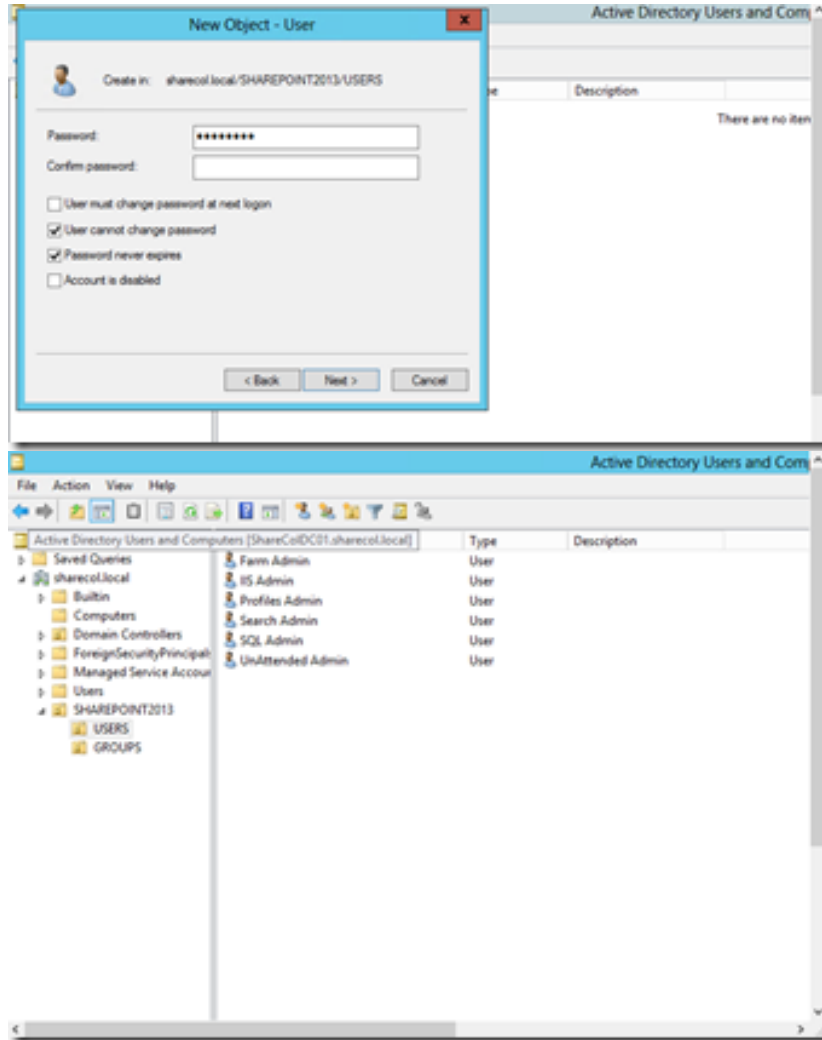


Estando en USERS, vamos a crear la tabla de usuarios que a continuación relaciono. El nombre de los usuarios si es a la libre disposición según las normas que tengan en su empresa. Una ayuda para conocer los usuarios que deberíamos crear para SharePoint 2013, la encuentran en la siguiente URL <http://social.technet.microsoft.com/wiki/contents/articles/14500.sharepoint-2013-service-accounts.aspx>.

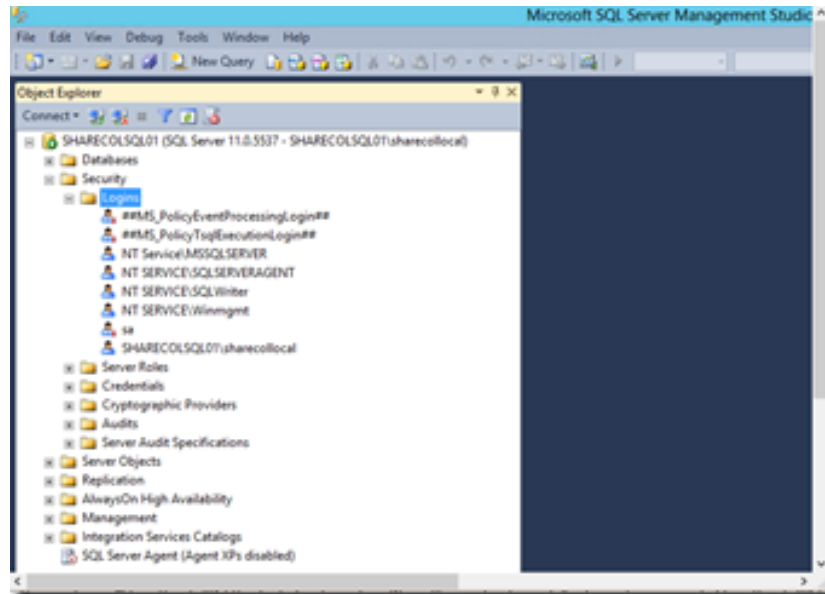
Para efectos del workshop, solo crearemos 6 usuarios y son los siguientes:

Usuario	Descripción del Rol
svcfarmadmin	<ul style="list-style-type: none">• usuario administrador de la granja
svcsqladmin	<ul style="list-style-type: none">• usuario administrador de SQL
svcsearchadmin	<ul style="list-style-type: none">• usuario administrador de la búsqueda empresarial
svciispool	<ul style="list-style-type: none">• usuario que administrará el pool de aplicaciones de todas las aplicaciones incluyendo las de servicio.
svcprofilesnc	<ul style="list-style-type: none">• usuario que se deberá configurar con permisos especiales para realizar la sincronización de perfiles del directorio activo.
svcunattended	<ul style="list-style-type: none">• usuario para configurar los permisos de acceso a las aplicaciones que requieren seguridad a nivel de sharepoint y que se configuran en la secure store service application.

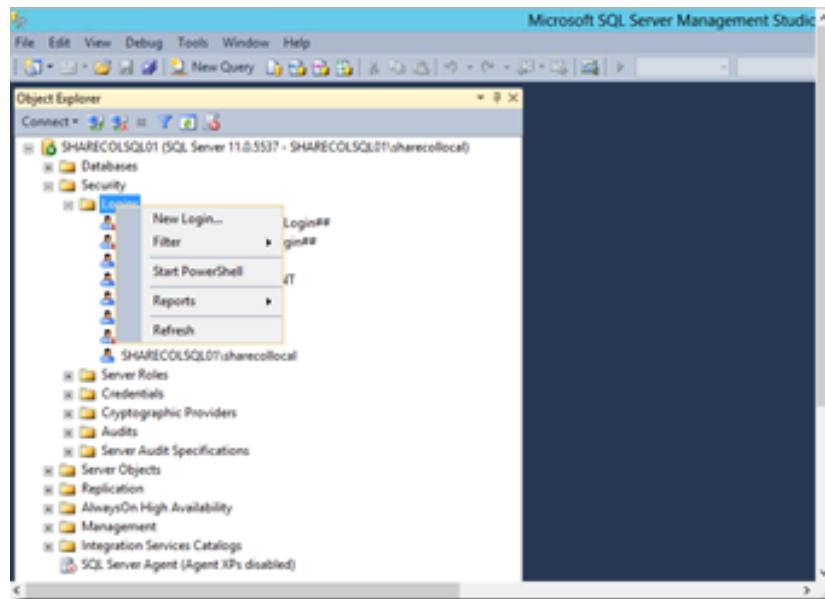
Como consideración especial, para que un usuario sea usuario de servicio, deben habilitar las siguientes dos opciones una vez estén creando el usuario:



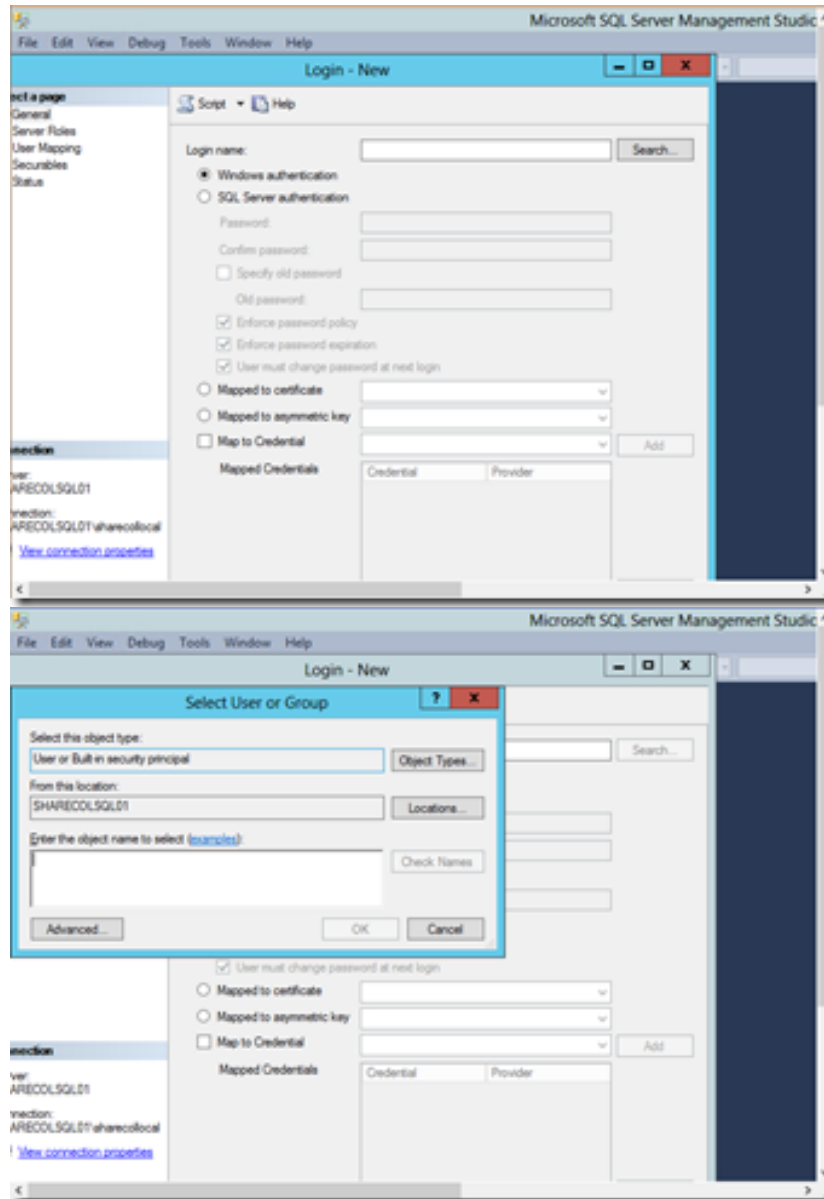
Una vez los usuarios estén creados en el Directorio Activo, procedemos a la creación de los logins en la consola de SQL Server 2012 SP2. Vamos a crear dos inicialmente, uno para el administrador de la granja y le daremos el rol de Sysadmin y el otro el administrador de SQL Server también con rol de Sysadmin.



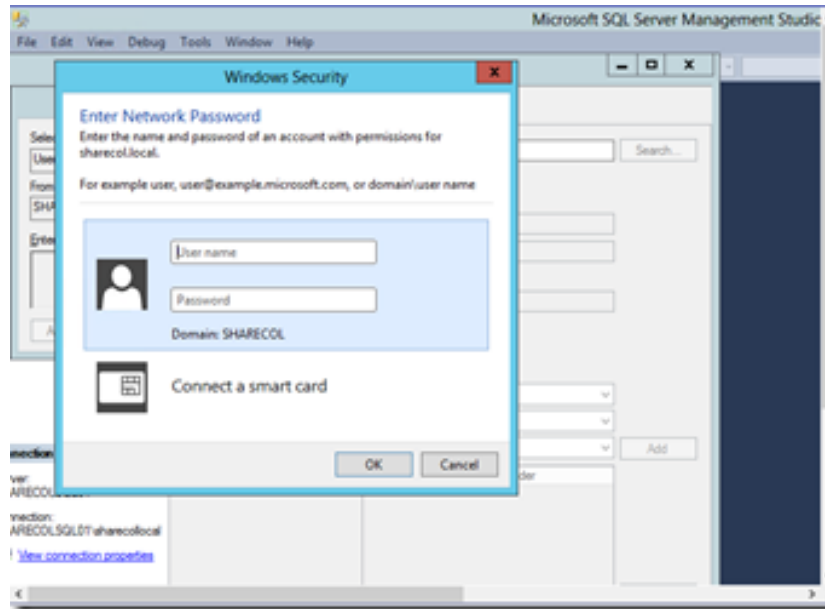
Estando en logins, damos clic derecho y clic sobre "new login".



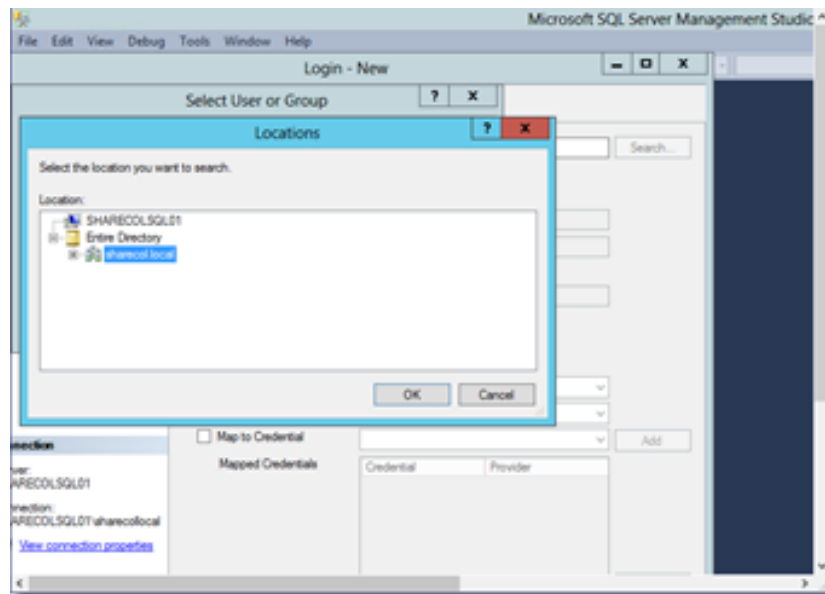
Se abre la ventana de new login y luego damos clic en el botón "Search" para determinar en donde se van a buscar los usuarios para que ingresen a SQL Server.



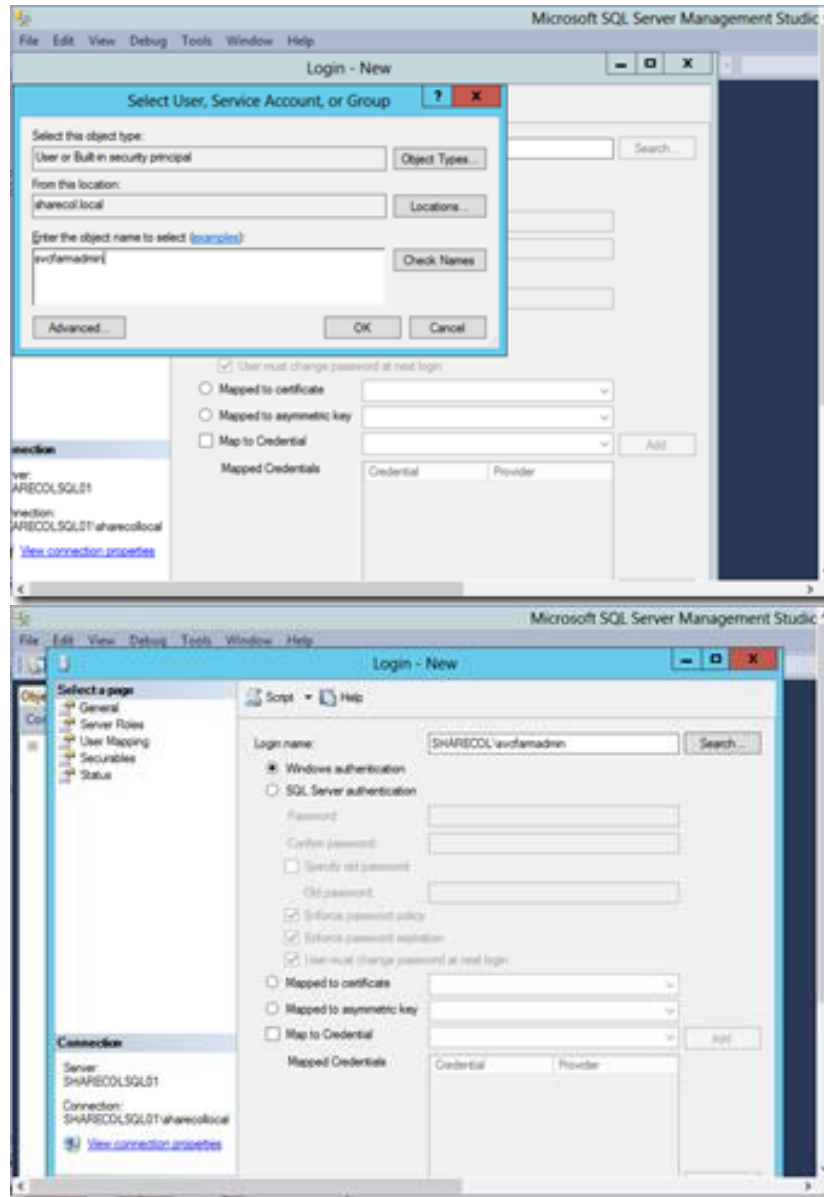
Punto de aclaración, como son usuarios del dominio, y estamos logueados como usuario local, es importante decirle a SQL Server que busque la información de los usuarios en el dominio. Por eso, al darle clic en el botón "locations", se abre la siguiente pantalla solicitando credenciales del directorio activo. Ahí, debemos ingresar las credenciales del administrador del dominio que creamos en el primer Lab.



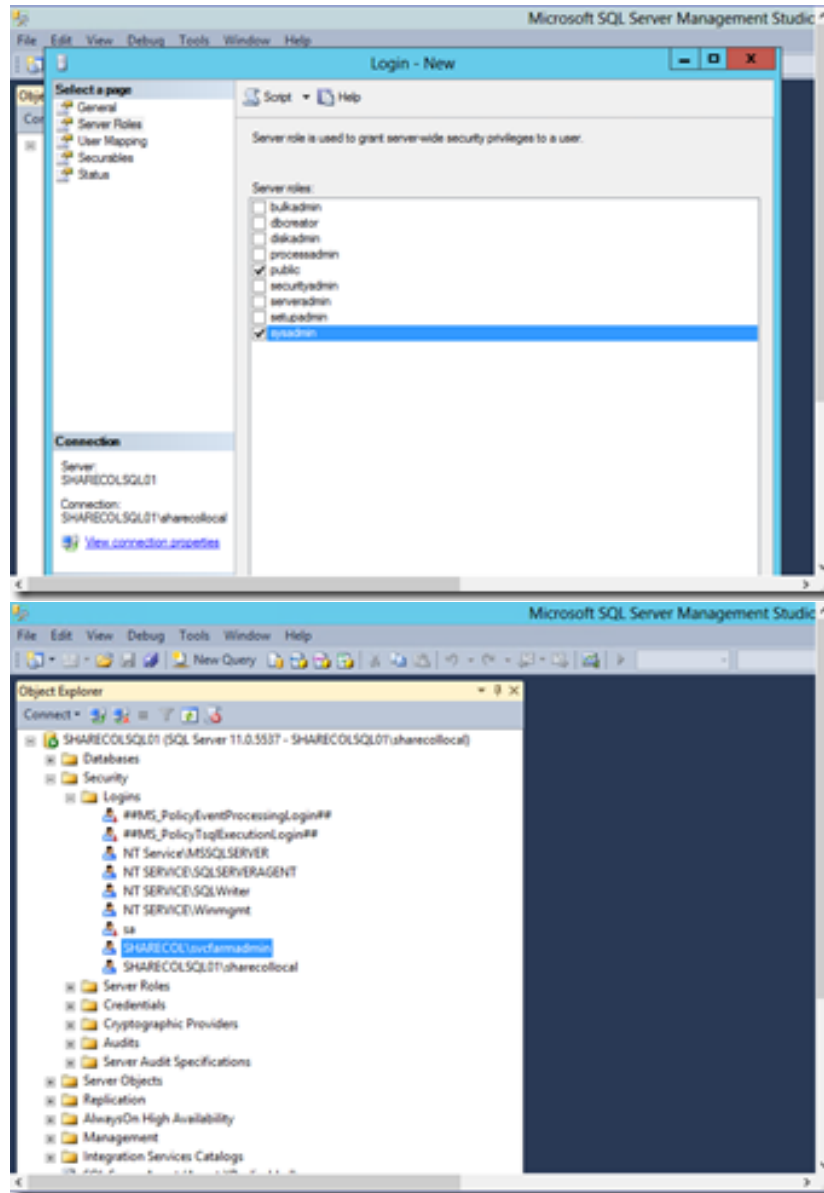
Una vez digitadas las credenciales, encontramos la información del Directorio activo que es desde donde vamos a configurar los usuarios Sysadmin. Seleccionamos el dominio y damos clic en el botón "Ok".



Luego se despliega la ventana donde ingresamos el usuario que vamos a crear en SQL Server.



Después de seleccionar el usuario del directorio activo, en el menú de la izquierda seleccionamos "Server Roles" y seleccionamos "Sysadmin". Luego damos clic en el botón "Ok".



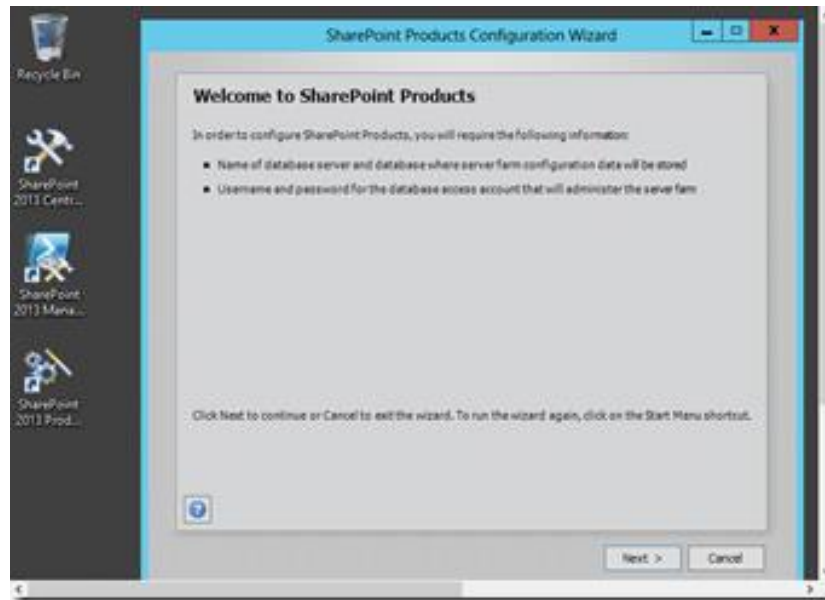
Para agregar el otro usuario, se repiten los pasos anteriores. Los dos usuarios que se han agregado a SQL Server 2012 SP2, también deben ser agregados al grupo de administradores locales del servidor de SQL Server 2012 SP2 y el de SharePoint 2013.

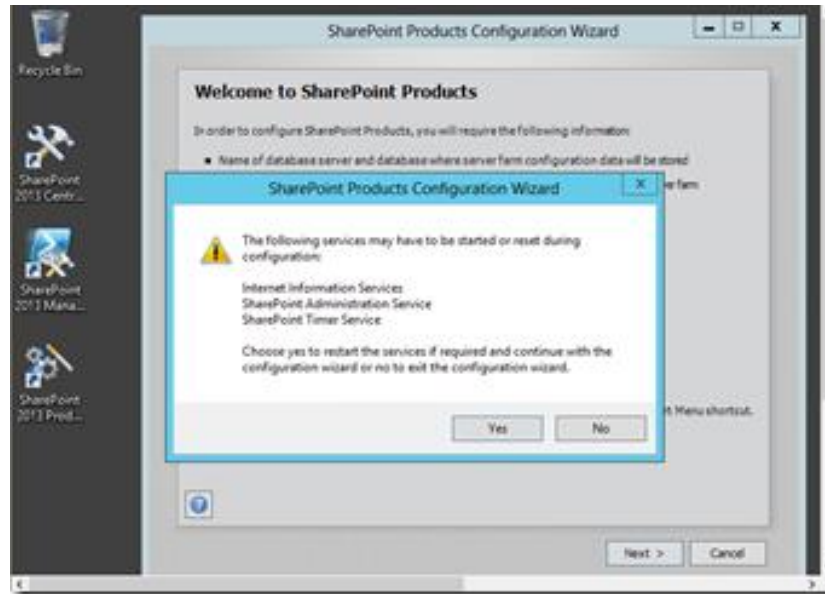
e. Creación de la Granja de SharePoint 2013 SP1

En este punto, vamos a proceder con la configuración de la granja de SharePoint 2013.

En el ítem "c" del Ejercicio, creamos una VM con SharePoint 2013 instalado y en el ítem "d" creamos una VM con SQL Server 2012 e hicimos su configuración para que funcione con usuarios de dominio, lo que nos resta es realizar la configuración de la granja de SharePoint 2013.

- Entramos a la máquina virtual por remote desktop y nos hacemos login con el usuario administrador de la granja.
- Damos clic en "SharePoint 2013 Products Configuration Wizard" y se abre la pantalla de configuración de SharePoint 2013. Damos clic en el botón Next.





- En la pantalla que aparece, seleccionamos "Create a new server farm" y damos clic en el botón Next.

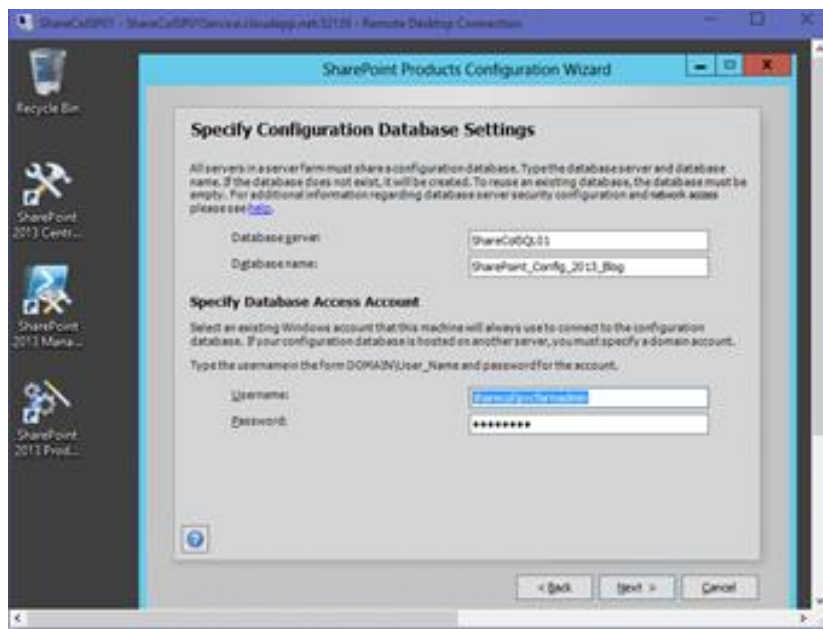


- El wizard de SharePoint nos solicita la información del servidor de base de datos, el nombre de la base de datos y el usuario con el que se va a configurar la granja. Para efectos de este post, la información queda así:

Nombre del servidor: IberianSPConfSQL01 (VM creada en el ítem "d")

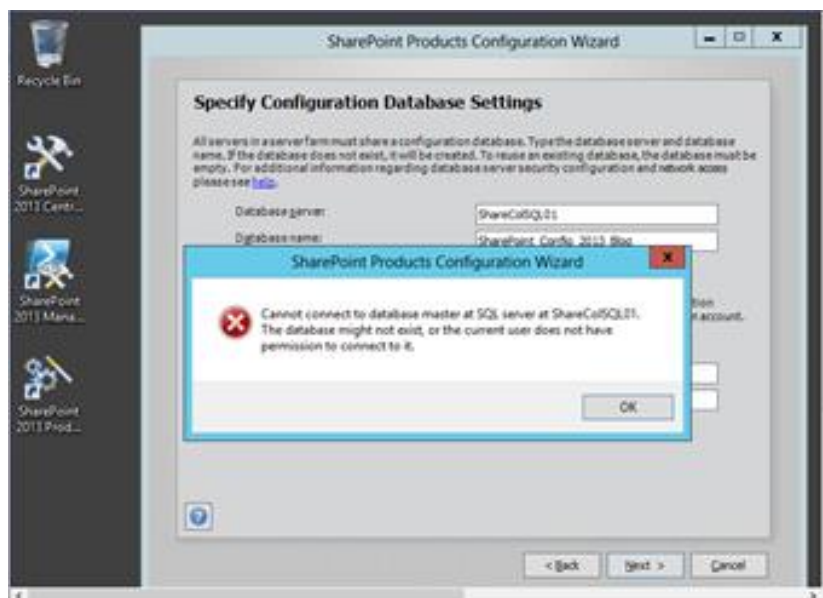
Nombre de la base de datos: SharePoint_Config_2013_ISPconf

Nombre del usuario: iberianspconf\svcfarmadmin con su respectiva contraseña.

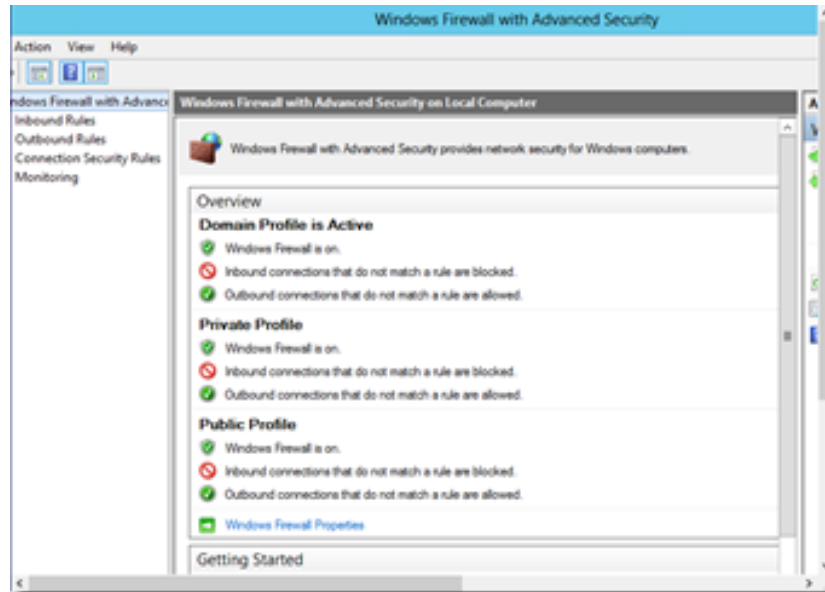


Una vez ingresada la información anterior, damos clic en el botón "Next" para continuar con la configuración. Si por alguna razón sale error, verifiquen que el firewall de Windows tanto en el servidor de SharePoint 2013 como en el de SQL Server 2012 SP2, esta desactivado.

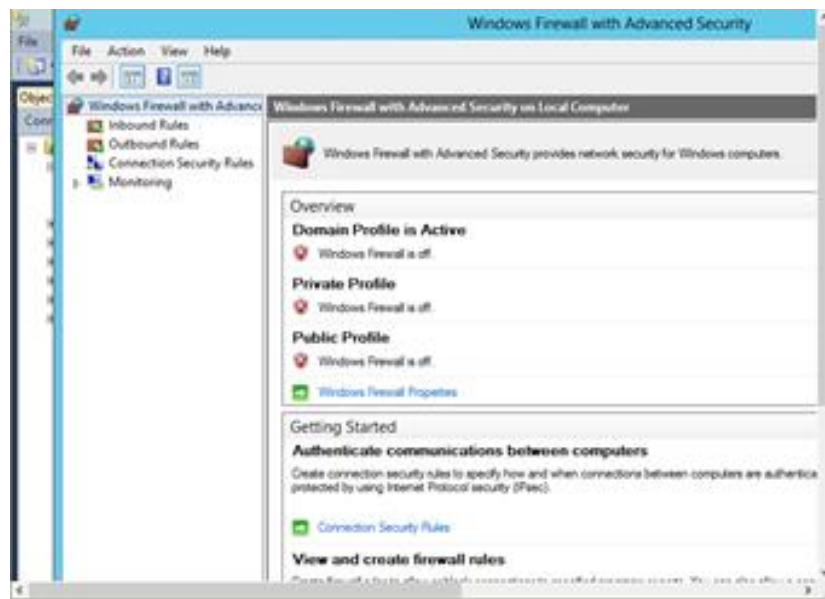
Si hay error, saldrá esta información:



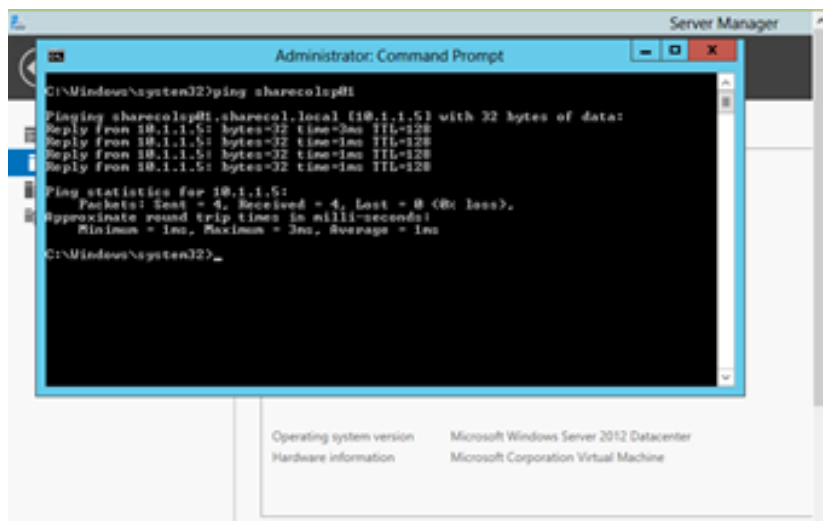
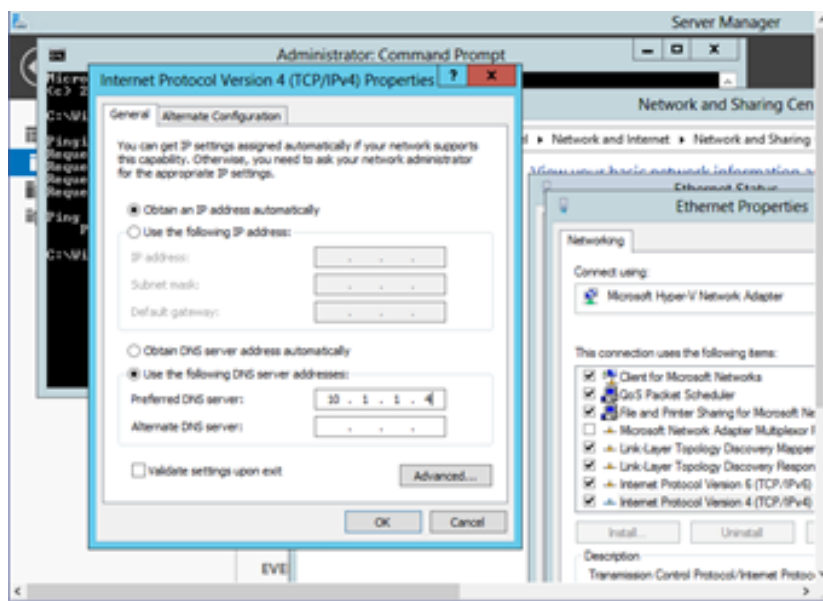
Verificamos que el Firewall este desactivado en los servidores de SQL Server 2012 SP2 y SharePoint 2013. Si está activado, es necesario desactivarlo o crear una regla para el puerto 1433 y 1434. El puerto 1433 es el estándar de conexión de SQL Server y 1434 es el Discovery de SQL Server.



Desactivamos el Firewall para el Domain, Private y Public **(NO es la mejor práctica desactivar el Firewall, pero para efectos del workshop, lo hacemos)**



Si lo anterior no es el problema, verifiquen que la tarjeta de red de las VMs de SharePoint 2013, SQL Server 2012 SP2 y Controlador de Dominio, tengan en la configuración de IPV4 el DNS 10.1.1.4 que es la IP de la máquina virtual del Controlador de Dominio que también es el DNS primario. Si en Azure no les fue asignada esa dirección IP, verifiquen que ip tiene el controlador de dominio y deberán digitarla en todos los servidores. Si tiene esa configuración, asegúrense que desde la VM de SharePoint 2013 pueden hacer Ping a la de SQL Server 2012 y viceversa.



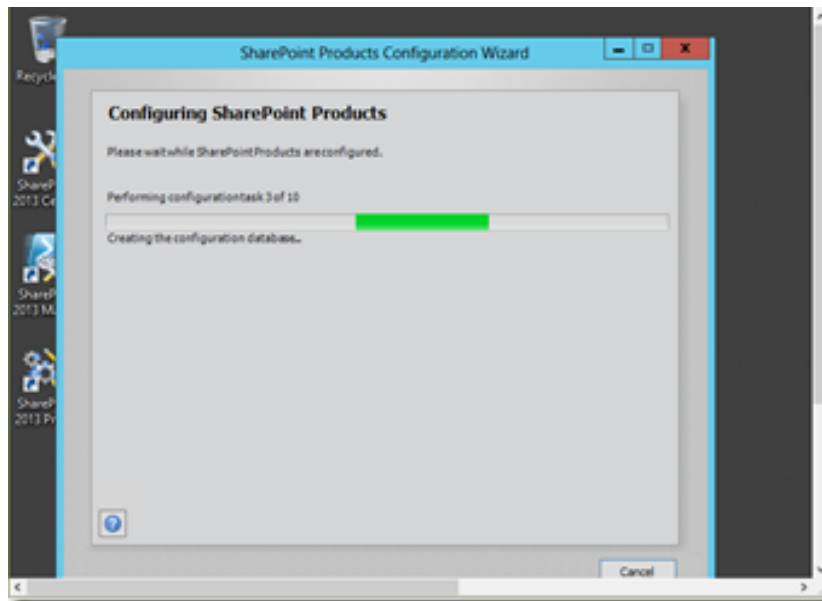
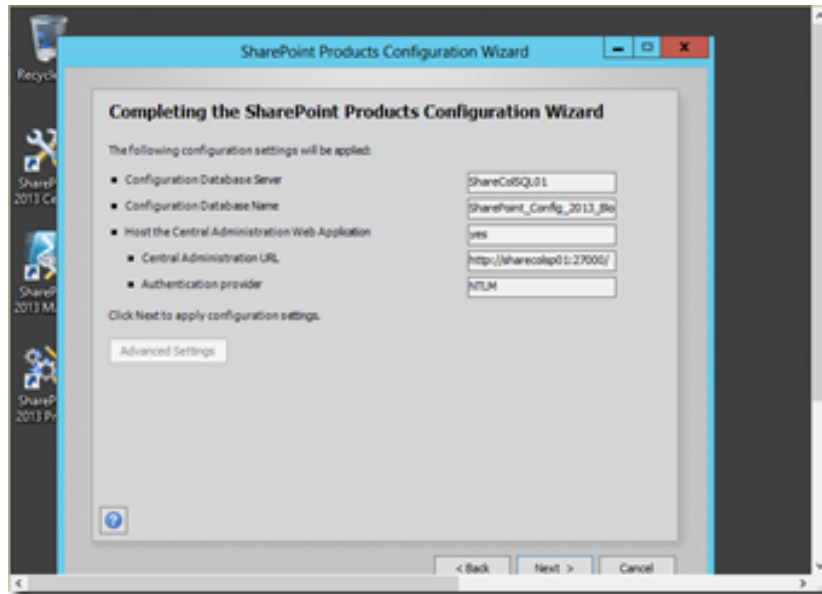
- Si todo está bien configurado, la siguiente pantalla será la creación de la "Pass phrase" para SharePoint, **Importante NO OLVIDARLA NUNCA!!!!**. Para efectos del workshop, digitamos "iberianspconf123*." sin las comillas dobles.

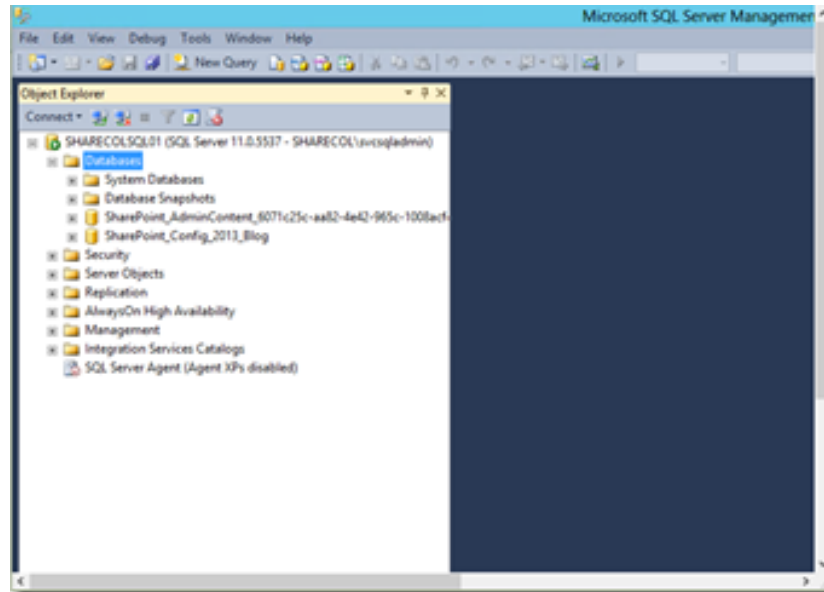


- Luego les pedirá configurar la administración central (Puerto IP y tipo de autenticación). Para efectos del workshop usaremos el puerto IP 27000 y tipo de autenticación NTLM.

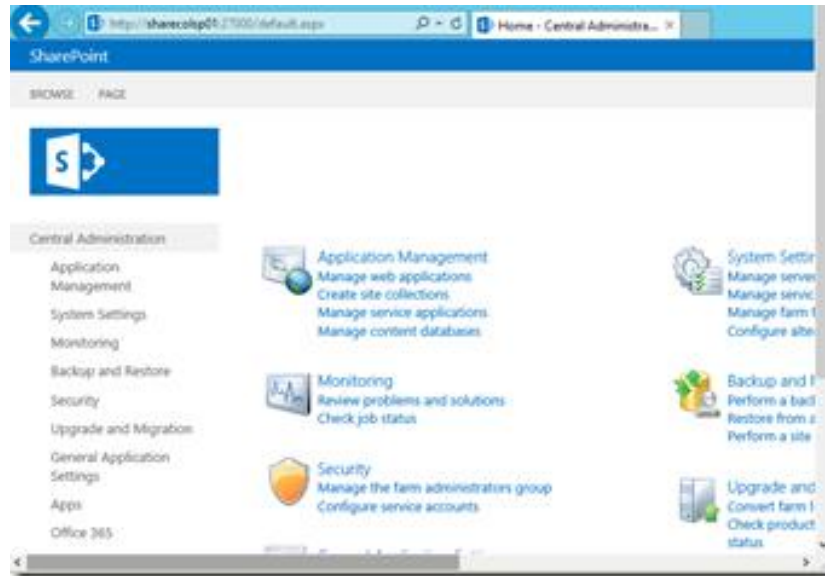


- Por último, el wizard pedirá que se verifique la información. Damos clic en el botón "Next" para que se inicie la configuración.





Posterior a la culminación del wizard, abrimos la administración central para verificar que todo este correctamente operativo.

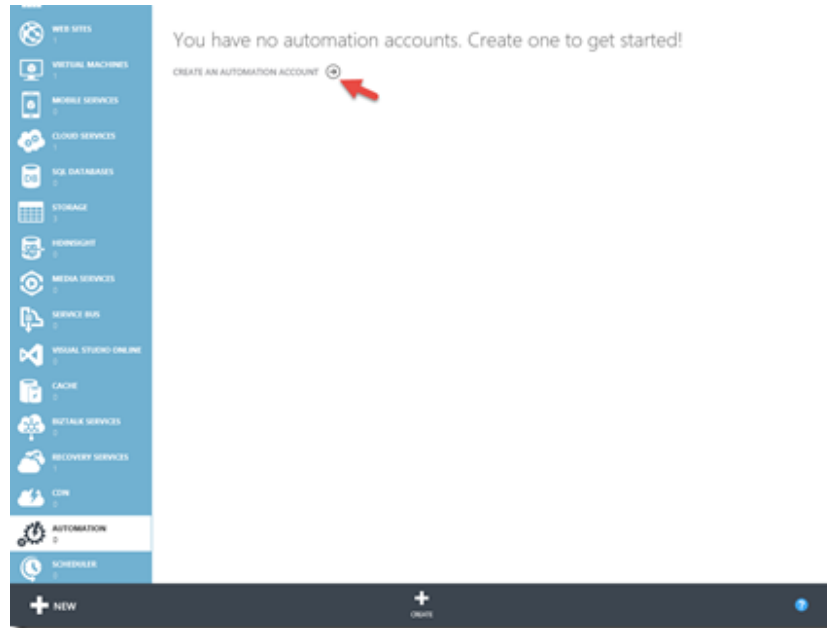


Como tarea después de esta instalación, deben instalar todas las CU (Cumulative Update) que hayan salido a la fecha. Es bueno mantener parchado SharePoint 2013 ya que algunas CU arreglan cosas de raíz.

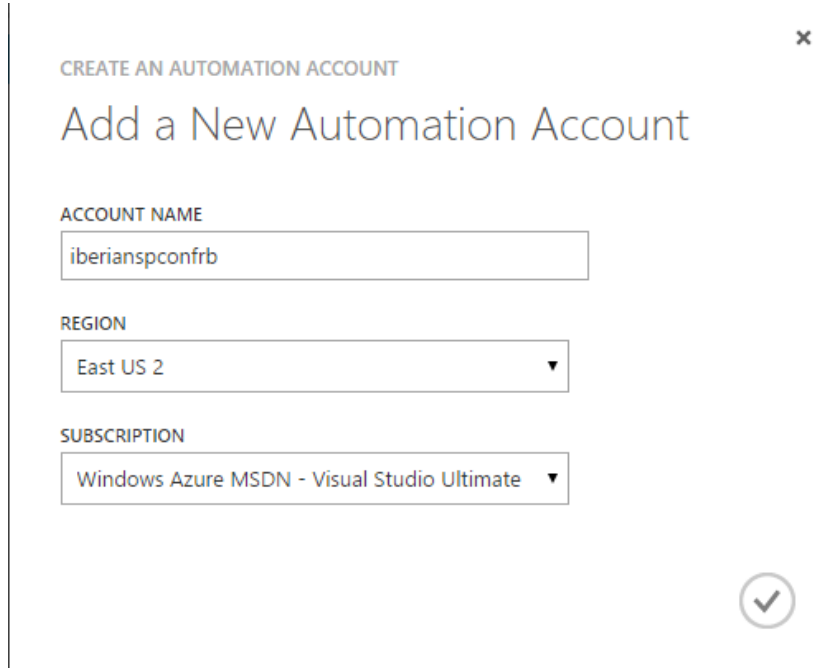
EJERCICIO 2

Configurar un RunBook en Azure para automatizar el scheduling de activación de las máquinas virtuales del ambiente de desarrollo

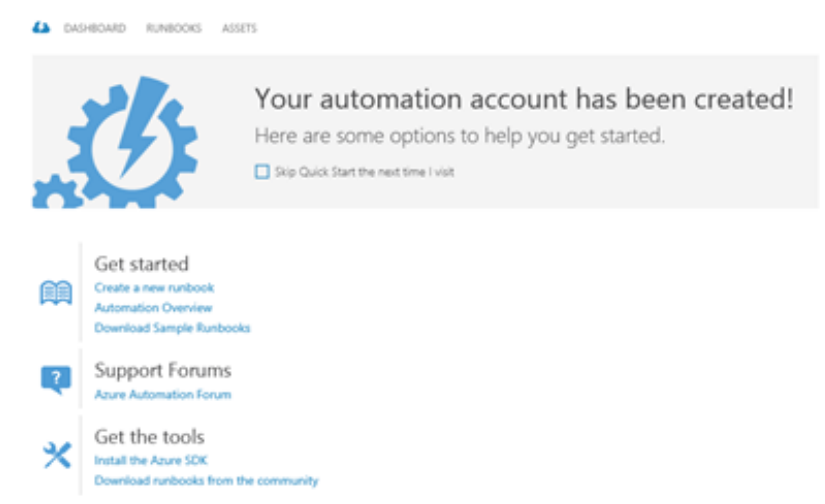
1. Entramos al trial de azure
2. Damos clic en "automation"



3. Si no tenemos una cuenta, procedemos a crear una dando clic en "Create an automation account". En Account Name ponemos "IberianSPConfrBA".

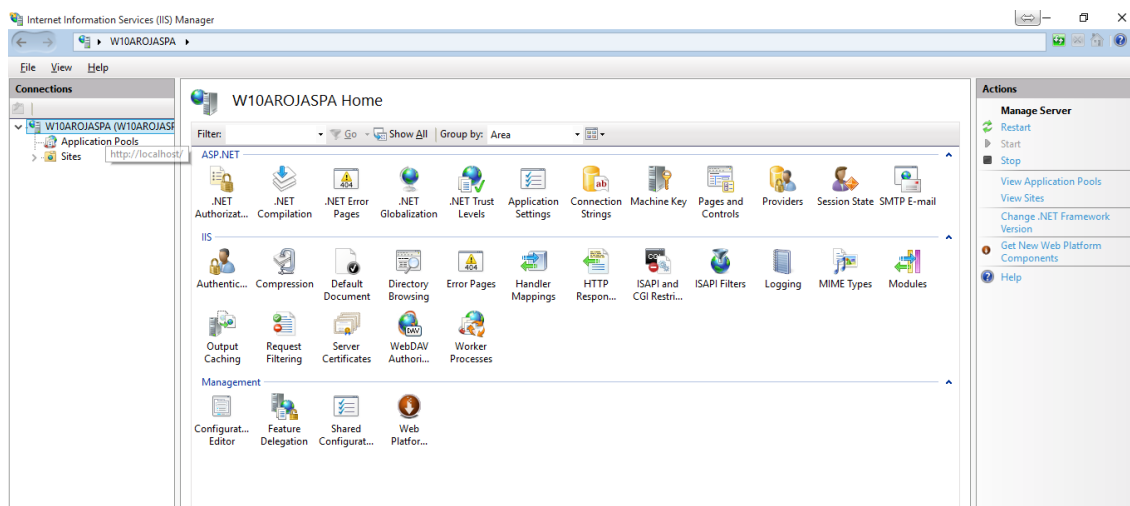


4. Una vez realizado el proceso, debe salir la siguiente pantalla:

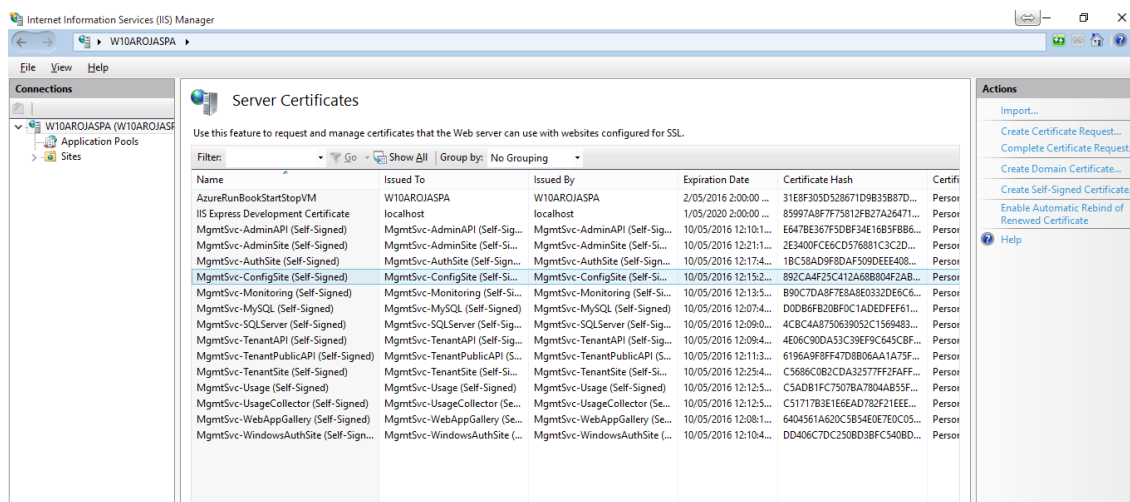


5. Antes de ejecutar un runbook, debemos asegurarnos que los siguientes componentes estén configurados:
 - a. Un certificado self-signed en IIS o generado por una entidad certificadora local o pública. Este certificado se debe cargar en azure. Para efectos del workshop, generamos uno en IIS.

■ Abrimos IIS

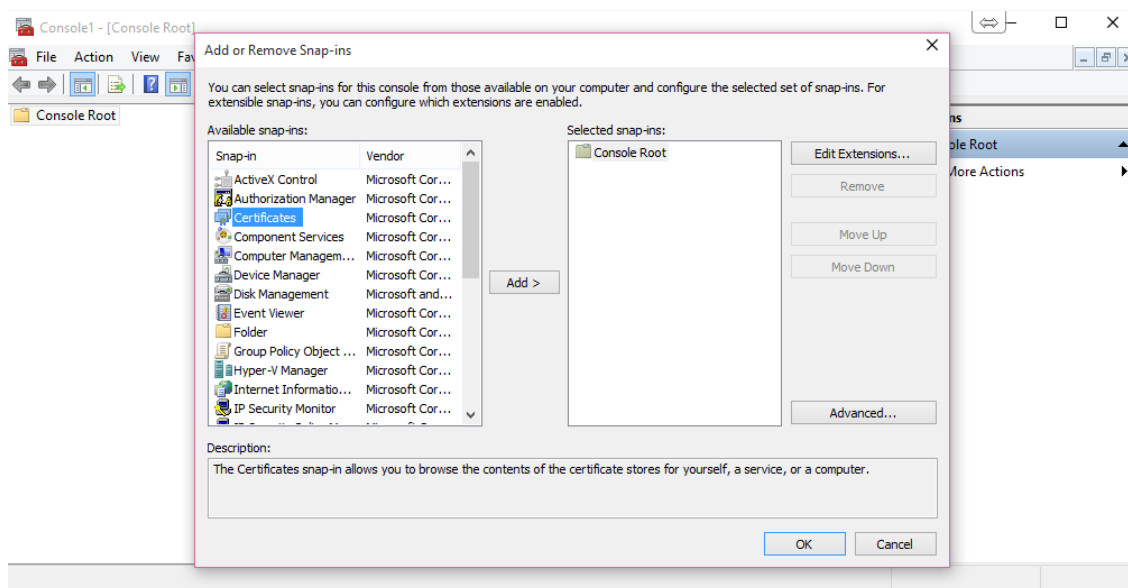


■ Damos doble clic en "Server Certificates"

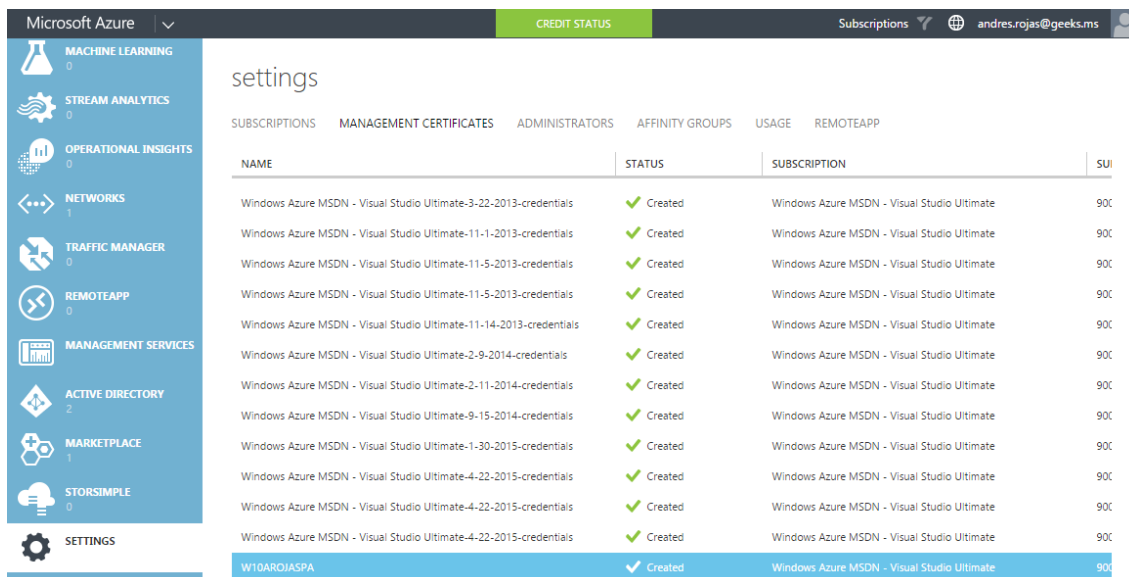


- Para exportar el certificado en formato .cer, abrimos una consola de administración: Tecla Windows + R y escribimos "MMC /a", agregamos la consola certificados "Computer Account" + "Local Computer", una vez en la consola buscamos el path personal y seleccionamos el certificado que acabamos de crear. Clic derecho sobre el certificado y le decimos exportar. Dentro de las opciones de exportación dejamos las que están por defecto, luego seleccionamos la ruta donde lo vamos a dejar y le agregamos le nombre

"IberianSPConfCert", guardamos y le damos finalizar al wizard. Para exportar el certificado en formato .pfx, abrimos una consola de administración: Tecla Windows + R y escribimos "MMC /a", agregamos la consola certificados "Computer Account" + "Local Computer", una vez en la consola buscamos el path personal y seleccionamos el certificado que acabamos de crear. Clic derecho sobre el certificado y le decimos exportar. Dentro de las opciones de exportación le decimos que exporte la llave privada, quitamos el check de la opción "include all certificates in the certification path if possible", escribimos un password para la exportación "123.abc*", luego seleccionamos la ruta donde lo vamos a dejar y le agregamos le nombre "IberianSPConfCert", guardamos y le damos finalizar al wizard.

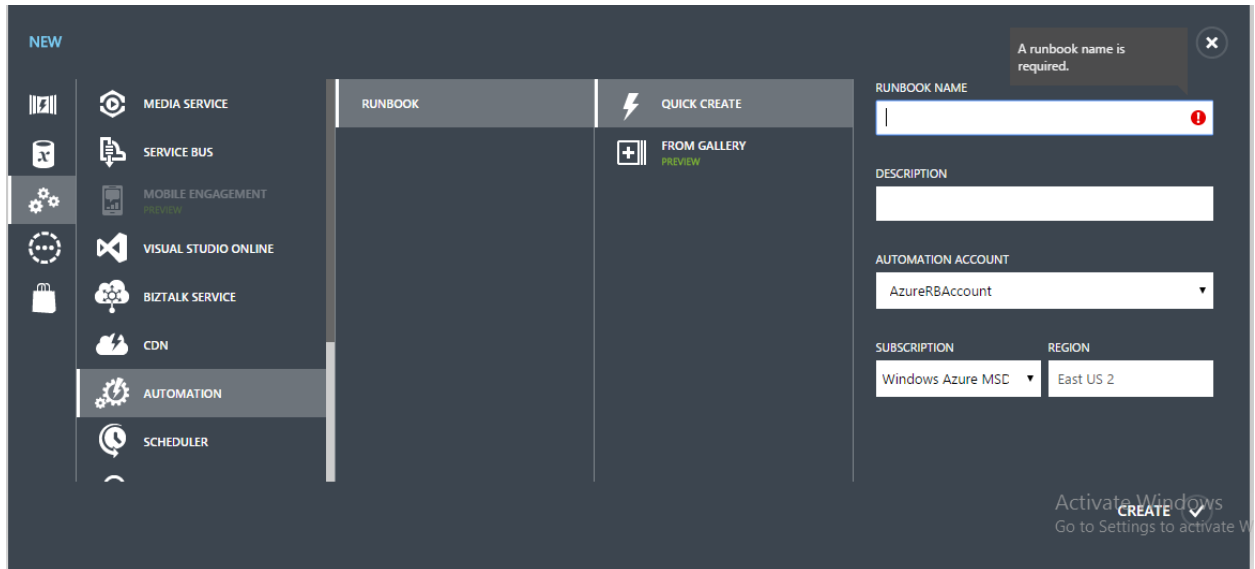


- Por último, cargamos el certificado ".cer" en azure. Para eso, volvemos a azure y damos clic en settings, luego en "Upload" y registramos la información que nos solicitan.



NAME	STATUS	SUBSCRIPTION	SUI
Windows Azure MSDN - Visual Studio Ultimate-3-22-2013-credentials	✓ Created	Windows Azure MSDN - Visual Studio Ultimate	90C
Windows Azure MSDN - Visual Studio Ultimate-11-1-2013-credentials	✓ Created	Windows Azure MSDN - Visual Studio Ultimate	90C
Windows Azure MSDN - Visual Studio Ultimate-11-5-2013-credentials	✓ Created	Windows Azure MSDN - Visual Studio Ultimate	90C
Windows Azure MSDN - Visual Studio Ultimate-11-5-2013-credentials	✓ Created	Windows Azure MSDN - Visual Studio Ultimate	90C
Windows Azure MSDN - Visual Studio Ultimate-11-14-2013-credentials	✓ Created	Windows Azure MSDN - Visual Studio Ultimate	90C
Windows Azure MSDN - Visual Studio Ultimate-2-9-2014-credentials	✓ Created	Windows Azure MSDN - Visual Studio Ultimate	90C
Windows Azure MSDN - Visual Studio Ultimate-2-11-2014-credentials	✓ Created	Windows Azure MSDN - Visual Studio Ultimate	90C
Windows Azure MSDN - Visual Studio Ultimate-9-15-2014-credentials	✓ Created	Windows Azure MSDN - Visual Studio Ultimate	90C
Windows Azure MSDN - Visual Studio Ultimate-1-30-2015-credentials	✓ Created	Windows Azure MSDN - Visual Studio Ultimate	90C
Windows Azure MSDN - Visual Studio Ultimate-4-22-2015-credentials	✓ Created	Windows Azure MSDN - Visual Studio Ultimate	90C
Windows Azure MSDN - Visual Studio Ultimate-4-22-2015-credentials	✓ Created	Windows Azure MSDN - Visual Studio Ultimate	90C
Windows Azure MSDN - Visual Studio Ultimate-4-22-2015-credentials	✓ Created	Windows Azure MSDN - Visual Studio Ultimate	90C
W10AR0JASPA	✓ Created	Windows Azure MSDN - Visual Studio Ultimate	90C

- b. Volvemos a Automation, clic en la cuenta de "IberianSPConfrBA", clic en Assets y luego clic en add settings para crear una conexión.
 - Agregamos una conexión con nombre "IberianSPConfCon". Para eso, necesitamos el nombre del certificado cargado en el punto anterior y el id de la suscripción.
 - c. Creamos un certificado. Aquí vamos a necesitar el ".pfx". Clic en add settings y damos clic en "create a credential". En credential type, seleccionamos "certificate". Ponemos como nombre "IberianSPConfCre". Seleccionamos el certificado ".pfx" y luego digitamos la clave que usamos en la exportación "123.abc*". Le decimos que no sea exportable.
6. Una vez terminada la configuración, procedemos a crear un runbook el que nos va permitir prender la máquina de directorio activo antes que las demás. Para eso, damos clic en "+ New" en la parte inferior izquierda del portal de azure. Seleccionamos "app Service\automation\runbook" + quick create. En runbook name ponemos "iberianSPConfStarDCVM", seleccionamos la cuenta "IberianSPConfrBA" y damos clic en "Create"



7. Asegúrense de ejecutar el siguiente código con la información de los nombres de sus máquinas virtuales:

```
$MyConnection = "Nombre de la conexión"
$MyCert = "nombre del certificado"
$MySubscriptionName = "nombre de su suscripción de azure"

# Write-Output "Subscription Name: $MySubscriptionName"

# Get the Azure Automation Connection

$Con = Get-AutomationConnection -Name $MyConnection
if ($Con -eq $null)
{
    Write-Output "Connection entered: $MyConnection does not exist in the automation
service. Please create one "
}
else
{
    $SubscriptionID = $Con.SubscriptionID
    $ManagementCertificate = $Con.AutomationCertificateName

    Write-Output "-----"
    Write-Output "Connection Properties: "
    Write-Output "SubscriptionID: $SubscriptionID"
# Write-Output "Certificate setting name: $ManagementCertificate"
}
```

```
# Get Certificate & print out its properties
$Cert = Get-AutomationCertificate -Name $MyCert
if ($Cert -eq $null)
{
    Write-Output "Certificate entered: $MyCert does not exist in the automation service. Please
create one "
}
else
{
    $Thumbprint = $Cert.Thumbprint

# Write-Output "Certificate Properties: "
# Write-Output "Thumbprint: $Thumbprint"
}
```

```
#Set and Select the Azure Subscription
Set-AzureSubscription -SubscriptionName $MySubscriptionName -Certificate $Cert -
SubscriptionId $SubscriptionID
Select-AzureSubscription -SubscriptionName $MySubscriptionName
Write-Output "-----"
#Virtual Machines
```

```
#Action: Start VM
$vmService = 'nombre del servicio que maneja el controlador de dominio'
$vmName = 'nombre de la maquina controlador de dominio'
Get-AzureVM -ServiceName $vmService -Name $vmName | Start-AzureVM
Write-Output "Service Started $vmService"
Write-Output "VM Started $vmName"
Write-Output "-----"
```

8. Luego creamos otros runbook para prender el resto de máquinas (SharePoint, SQL Server y demas que tengamos). Nombre del runbook "iberianSPConfStarOthersVM"

9. Asegúrense de ejecutar el siguiente código con la información de los nombres de sus máquinas virtuales:

```
$MyConnection = "Nombre de la conexión"
$MyCert = "nombre del certificado"
$MySubscriptionName = "nombre de su subscription de azure"
#Write-Output "Subscription Name: $MySubscriptionName"

# Get the Azure Automation Connection

$Con = Get-AutomationConnection -Name $MyConnection
if ($Con -eq $null)
{
    Write-Output "Connection entered: $MyConnection does not exist in the automation service.
Please create one "
}
else
{
    $SubscriptionID = $Con.SubscriptionID
    $ManagementCertificate = $Con.AutomationCertificateName

    Write-Output "-----"
    Write-Output "Connection Properties: "
    Write-Output "SubscriptionID: $SubscriptionID"
    #Write-Output "Certificate setting name: $ManagementCertificate"
}

# Get Certificate & print out its properties
$Cert = Get-AutomationCertificate -Name $MyCert
if ($Cert -eq $null)
{
    Write-Output "Certificate entered: $MyCert does not exist in the automation service. Please
create one "
}
else
{
    $Thumbprint = $Cert.Thumbprint

    #Write-Output "Certificate Properties: "
    #Write-Output "Thumbprint: $Thumbprint"
}
```

```
#Set and Select the Azure Subscription
Set-AzureSubscription -SubscriptionName $MySubscriptionName -Certificate $Cert -
SubscriptionId $SubscriptionID
Select-AzureSubscription -SubscriptionName $MySubscriptionName
Write-Output "-----"
#Virtual Machines
$vm = @()
$vm += 'Nombre de VM de SP'
$vm += 'Nombre de VM de SQL'
$vmService = @()
$vmService += 'Nombre del servicio que controla la VM de SP'
$vmService += 'Nombre del servicio que controla la VM de SQL'

#Action: Start VM
for($i=0; $i -lt $vm.Length; $i++)
{
    Get-AzureVM -ServiceName $vmService[$i] -Name $vm[$i] | Start-AzureVM
    Write-Output "Service Started " $vmService[$i]
    Write-Output "VM Started " $vm[$i]
    Write-Output "-----"
}
}
```

10. Luego creamos otros runbook para prender el resto de máquinas (SharePoint, SQL Server y demás que tengamos). Nombre del runbook "iberianSPConfStarOthersVM"

11. Asegúrense de ejecutar el siguiente código con la información de los nombres de sus máquinas virtuales:

```
$MyConnection = "Nombre de la conexión"
$MyCert = "nombre del certificado"
$MySubscriptionName = "nombre de su suscripción de azure"

#Write-Output "Subscription Name: $MySubscriptionName"

# Get the Azure Automation Connection

$Con = Get-AutomationConnection -Name $MyConnection
if ($Con -eq $null)
{
    Write-Output "Connection entered: $MyConnection does not exist in the automation service.
Please create one "
}
else
```

```
{
  $SubscriptionID = $Con.SubscriptionID
  $ManagementCertificate = $Con.AutomationCertificateName

  Write-Output "-----"
  Write-Output "Connection Properties: "
  Write-Output "SubscriptionID: $SubscriptionID"
  #Write-Output "Certificate setting name: $ManagementCertificate"
}

# Get Certificate & print out its properties
$Cert = Get-AutomationCertificate -Name $MyCert
if ($Cert -eq $null)
{
  Write-Output "Certificate entered: $MyCert does not exist in the automation service. Please
create one "
}
else
{
  $Thumbprint = $Cert.Thumbprint

  #Write-Output "Certificate Properties: "
  #Write-Output "Thumbprint: $Thumbprint"
}

#Set and Select the Azure Subscription
Set-AzureSubscription -SubscriptionName $MySubscriptionName -Certificate $Cert -
SubscriptionId $SubscriptionID
Select-AzureSubscription -SubscriptionName $MySubscriptionName
Write-Output "-----"
#Virtual Machines
$vms = @()
$vms += 'ShareColDC01'
$vms += 'ShareColSP01'
$vms += 'ShareColSQL01'
$vmsservice = @()
$vmsservice += 'ShareColDC01service'
$vmsservice += 'ShareColSP01service'
$vmsservice += 'ShareColSQL01service'

#Action: Start VM
```

```
for($i=0; $i -lt $vms.Length; $i++)  
{  
    Stop-AzureVM -Name $vms[$i] -ServiceName $vmsservice[$i] -Force  
    Write-Output "Service Stopped " $vmsservice[$i]  
    Write-Output "VM Stopped " $vms[$i]  
    Write-Output "-----"  
}
```

Ejercicio 3

Instalar y configurar una granja de SharePoint 2013 usando PowerShell

1. Crear una VM en azure solo con sistema operativo Windows Server 2012 y sin incluirla en el dominio "iberianspconf"

Buscamos la imagen:

```
( Get-AzureVMImage | where-object { $_.Label -like "Windows Server 2012*" } )
```

Ejecutamos el script de creación de la maquina:

```
#Clean screen
#-----
cls

#import modules to connect azure instance
#-----
import-module "C:\Program Files (x86)\Microsoft
SDKs\Azure\PowerShell\ServiceManagement\Azure\Azure.psd1"
Import-AzurePublishSettingsFile
"D:\MVPEvents\IberianSharePointConference2015\Workshop\Desarrollo-pruebas de MSDN -
Pago por uso-Windows Azure MSDN - Visual Studio Ultimate-6-7-2015-
credentials.publishsettings"
Set-AzureSubscription -SubscriptionName "windows Azure MSDN - Visual Studio Ultimate"
-CurrentStorageAccountName "storageiberianspconf"
Select-AzureSubscription -SubscriptionName "windows Azure MSDN - Visual Studio
Ultimate"

#specify the Domain Controller DNS IP to 10.1.1.4 and VM Name
$myVMDNS = new-azuredns -Name "IberianSPConfDNS01" -IPAddress "10.1.1.4"
$vmName = "IberianSPConfSP01"

#OS Image to use
$image = "a699494373c04fc0bc8f2bb1389d6106__windows-Server-2012-Datacenter-201505.01-
en.us-127GB.vhd"
$service = "IberianSPConfWS01Svc"
$ag = "IberianSPConfAF"
$vnet = "iberianspconfnet"
$adminlocal = "iberianadmin"
$domainuser = "iberianadmin"
$password = "iberianspconf123*."
$size = "A5"
$netbiosname = "iberianspconf"
$domainname = "iberianspconf.local"
$subnetback = "BACKEND"

$myVM = (New-AzureVMConfig -Name $vmName -InstanceSize $size -ImageName $image |
Add-AzureProvisioningConfig -windowsDomain -AdminUsername $adminlocal -Password
$password -Domain $netbiosname -DomainUsername $domainuser -DomainPassword $password -
JoinDomain $domainname |
Set-AzureSubnet -SubnetNames $subnetback);

New-AzureVM -ServiceName $service -VMS $myVM -AffinityGroup $ag -DnsSettings $myVMDNS
-VNetName $vnet;
```

2. Verificar las condiciones del servidor ejecutando el script **"1-VerificacionServidorSP.ps1"**

```
$ComputerSystem = @{{Expression="{0:N2}" -f
($_.Totalphysicalmemory/1GB)};Label="RAM (GB)"},
@{{Expression="{$_}.SystemType";Label="Processor Type"},
@{{Expression="{$_}.NumberOfLogicalProcessors";Label="# of Processors"}}
cls
echo "**** Test Server Configuration before Install of SharePoint 2013 ****"
echo ""
echo "Processor Type = x64"
echo "Minimum Processors = 4"
echo "Minimum RAM = 8GB"
echo "Minimum System Drive Size = 80GB"
echo ""
GWMi Win32_ComputerSystem | FT $ComputerSystem -AutoSize
Get-PSDrive C
```

3. Verificamos las características instaladas en el servidor que son prerequisites para la instalación de SharePoint 2013. Ejecutamos el script **"2-VerificarServerFeaturesSP_PowerShell.ps1"**

```
Get-WindowsFeature -Name NET-WCF-HTTP-Activation45,NET-WCF-TCP-Activation45,NET-
WCF-Pipe-Activation45,Net-Framework-Features,Web-Server,Web-WebServer,Web-Common-
Http,Web-Static-Content,Web-Default-Doc,Web-Dir-Browsing,Web-Http-Errors,Web-App-
Dev,Web-Asp-Net,Web-Net-Ext,Web-ISAPI-Ext,Web-ISAPI-Filter,Web-Health,Web-Http-
Logging,Web-Log-Libraries,Web-Request-Monitor,Web-Http-Tracing,Web-Security,Web-
Basic-Auth,Web-Windows-Auth,Web-Filtering,Web-Digest-Auth,Web-Performance,Web-
Stat-Compression,Web-Dyn-Compression,Web-Mgmt-Tools,Web-Mgmt-Console,Web-Mgmt-
Compat,Web-Metabase,Application-Server,AS-Web-Support,AS-TCP-Port-Sharing,AS-WAS-
Support, AS-HTTP-Activation,AS-TCP-Activation,AS-Named-Pipes,AS-Net-
Framework,WAS,WAS-Process-Model,WAS-NET-Environment,WAS-Config-APIs,Web-Lgcy-
Scripting,Windows-Identity-Foundation,Server-Media-Foundation,Xps-Viewer
```

4. Procedemos con la configuración de las características que son prerequisites para la instalación de SharePoint 2013. Ejecutamos el script **"3-ConfigurarFeaturesSP_PowerShell.ps1"**

```
#Install-WindowsFeature NET-WCF-HTTP-Activation45,NET-WCF-TCP-Activation45,NET-
WCF-Pipe-Activation45 -Source D:\Sources\sxs
```

```
Install-WindowsFeature NET-WCF-HTTP-Activation45,NET-WCF-TCP-Activation45,NET-WCF-
Pipe-Activation45
```

```
#Install-WindowsFeature Net-Framework-Features,Web-Server,Web-WebServer,Web-
Common-Http,Web-Static-Content,Web-Default-Doc,Web-Dir-Browsing,Web-Http-
Errors,Web-App-Dev,Web-Asp-Net,Web-Net-Ext,Web-ISAPI-Ext,Web-ISAPI-Filter,Web-
Health,Web-Http-Logging,Web-Log-Libraries,Web-Request-Monitor,Web-Http-
Tracing,Web-Security,Web-Basic-Auth,Web-Windows-Auth,Web-Filtering,Web-Digest-
Auth,Web-Performance,Web-Stat-Compression,Web-Dyn-Compression,Web-Mgmt-Tools,Web-
Mgmt-Console,Web-Mgmt-Compat,Web-Metabase,Application-Server,AS-Web-Support,AS-
TCP-Port-Sharing,AS-WAS-Support, AS-HTTP-Activation,AS-TCP-Activation,AS-Named-
Pipes,AS-Net-Framework,WAS,WAS-Process-Model,WAS-NET-Environment,WAS-Config-
APIs,Web-Lgcy-Scripting,Windows-Identity-Foundation,Server-Media-Foundation,Xps-
Viewer -Source D:\Sources\sxs
```

```
Install-WindowsFeature Net-Framework-Features,Web-Server,Web-WebServer,Web-Common-
Http,Web-Static-Content,Web-Default-Doc,Web-Dir-Browsing,Web-Http-Errors,Web-App-
Dev,Web-Asp-Net,Web-Net-Ext,Web-ISAPI-Ext,Web-ISAPI-Filter,Web-Health,Web-Http-
Logging,Web-Log-Libraries,Web-Request-Monitor,Web-Http-Tracing,Web-Security,Web-
Basic-Auth,Web-Windows-Auth,Web-Filtering,Web-Digest-Auth,Web-Performance,Web-
Stat-Compression,Web-Dyn-Compression,Web-Mgmt-Tools,Web-Mgmt-Console,Web-Mgmt-
Compat,Web-Metabase,Application-Server,AS-Web-Support,AS-TCP-Port-Sharing,AS-WAS-
Support, AS-HTTP-Activation,AS-TCP-Activation,AS-Named-Pipes,AS-Net-
```


Framework, WAS, WAS-Process-Model, WAS-NET-Environment, WAS-Config-APIs, Web-Lgcy-Scripting, Windows-Identity-Foundation, Server-Media-Foundation, Xps-Viewer

5. Se reinicia el servidor usando el comando **"Restart-Computer -force"**
6. Verificamos nuevamente que las características se hayan instalado correctamente en el servidor. Ejecutamos el script **"2-VerificarServerFeaturesSP_PowerShell.ps1"**

```
Get-WindowsFeature -Name NET-WCF-HTTP-Activation45, NET-WCF-TCP-Activation45, NET-
WCF-Pipe-Activation45, Net-Framework-Features, Web-Server, Web-WebServer, Web-Common-
Http, Web-Static-Content, Web-Default-Doc, Web-Dir-Browsing, Web-Http-Errors, Web-App-
Dev, Web-Asp-Net, Web-Net-Ext, Web-ISAPI-Ext, Web-ISAPI-Filter, Web-Health, Web-Http-
Logging, Web-Log-Libraries, Web-Request-Monitor, Web-Http-Tracing, Web-Security, Web-
Basic-Auth, Web-Windows-Auth, Web-Filtering, Web-Digest-Auth, Web-Performance, Web-
Stat-Compression, Web-Dyn-Compression, Web-Mgmt-Tools, Web-Mgmt-Console, Web-Mgmt-
Compat, Web-Metabase, Application-Server, AS-Web-Support, AS-TCP-Port-Sharing, AS-WAS-
Support, AS-HTTP-Activation, AS-TCP-Activation, AS-Named-Pipes, AS-Net-
Framework, WAS, WAS-Process-Model, WAS-NET-Environment, WAS-Config-APIs, Web-Lgcy-
Scripting, Windows-Identity-Foundation, Server-Media-Foundation, Xps-Viewer
```

7. Ahora usaremos un script desarrollado por Craig Lussier que nos permite descargar los prerequisites de software que requiere SharePoint 2013. Antes de ejecutar el script, creamos la carpeta "c:\PrerrequisitosSP2013". Ejecutamos el script Download-SP2013PreReqFiles.ps1, nos solicitará el nombre de la carpeta donde descargaremos todos los prerequisites.

```
param([string] $SharePoint2013Path = $(Read-Host -Prompt "Please enter the
directory path to where you wish to save the SharePoint 2013 Prerequisite
files."))

# Import Required Modules
Import-Module BitsTransfer

# Specify download url's for SharePoint 2013 prerequisites
$DownloadUrls = (
    "http://download.microsoft.com/download/9/1/3/9138773A-505D-43E2-
AC08-9A77E1E0490B/1033/x64/sqlncli.msi", # Microsoft SQL Server 2008 R2 SP1 Native
Client
    "http://download.microsoft.com/download/E/0/0/E0060D8F-2354-4871-
9596-DC78538799CC/Synchronization.msi", # Microsoft Sync Framework Runtime v1.0
SP1 (x64)
    "http://download.microsoft.com/download/A/6/7/A678AB47-496B-4907-
B3D4-0A2D280A13C0/windowsServerAppFabricSetup_x64.exe", # windows Server App
Fabric
    "http://download.microsoft.com/download/7/B/5/7B51D8D1-20FD-4BF0-87C7-
4714F5A1C313/AppFabric1.1-RTM-KB2671763-x64-ENU.exe", # Cumulative Update Package
1 for Microsoft AppFabric 1.1 for windows Server (KB2671763)
    "http://download.microsoft.com/download/D/7/2/D72FD747-69B6-40B7-875B-
C2B40A6B2BDD/windows6.1-KB974405-x64.msu", #Windows Identity Foundation (KB974405)
    "http://download.microsoft.com/download/0/1/D/01D06854-CA0C-46F1-
ADBA-EBF86010DCC6/rtm/MicrosoftIdentityExtensions-64.msi", # Microsoft Identity
Extensions
    "http://download.microsoft.com/download/9/1/D/91DA8796-BE1D-46AF-
8489-663AB7811517/setup_msipc_x64.msi", # Microsoft Information Protection and
Control Client
    "http://download.microsoft.com/download/8/F/9/8F93DBBD-896B-4760-
AC81-646F61363A6D/wcfDataServices.exe" # Microsoft WCF Data Services 5.0
)

function DownloadPrerequisites()
{
    Write-Host ""
```

```

Write-Host
"=====
Write-Host "      Downloading SharePoint 2013 Prerequisites Please wait..."
Write-Host
"=====

$ReturnCode = 0

foreach ($DownloadUrl in $DownloadUrls)
{
    ## Get the file name based on the portion of the URL after the last slash
    $FileName = $DownloadUrl.Split('/')[ -1]
    Try
    {
        ## Check if destination file already exists
        If (!(Test-Path "$SharePoint2013Path\$FileName"))
        {
            ## Begin download
            Start-BitsTransfer -Source $DownloadUrl -Destination
            $SharePoint2013Path\$FileName -DisplayName "Downloading `'$FileName`' to
            $SharePoint2013Path" -Priority High -Description "From $DownloadUrl..." -
            ErrorVariable err
            If ($err) {Throw ""}
        }
        Else
        {
            Write-Host " - File $FileName already exists, skipping..."
        }
    }
    Catch
    {
        $ReturnCode = -1
        Write-Warning " - An error occurred downloading `'$FileName`'"
        Write-Error $_
        break
    }
}
Write-Host " - Done downloading Prerequisites required for SharePoint 2013"

return $ReturnCode
}

```

```

function CheckProvidedDownloadPath()
{

    $ReturnCode = 0

    Try
    {
        # Check if destination path exists
        If (Test-Path $SharePoint2013Path)
        {
            # Remove trailing slash if it is present
            $script:SharePoint2013Path = $SharePoint2013Path.TrimEnd('\')
            $ReturnCode = 0
        }
        Else {

            $ReturnCode = -1
            Write-Host ""
            Write-Warning "Your specified download path does not exist. Please verify
            your download path then run this script again."
            Write-Host ""
        }
    }
}

```

```
}  
Catch  
{  
    $ReturnCode = -1  
    Write-Warning "An error has occurred when checking your specified  
download path"  
    Write-Error $_  
    break  
}  
  
return $ReturnCode  
}  
  
function DownloadPreReqs()  
{  
    $rc = 0  
    $rc = CheckProvidedDownloadPath  
    # Download Pre-Reqs  
    if($rc -ne -1)  
    {  
        $rc = DownloadPreRequisites  
    }  
  
    if($rc -ne -1)  
    {  
        Write-Host ""  
        Write-Host "Script execution is now complete!"  
        Write-Host ""  
    }  
}  
  
DownloadPreReqs
```

8. Descargamos el instalador de SharePoint 2013 Trial con su respectiva licencia
<http://www.microsoft.com/en-us/evalcenter/evaluate-sharepoint-server-2013>
Licencia: **NQTMW-K63MQ-39G6H-B2CH9-FRDWJ**
9. Una vez descargada la imagen, damos doble clic para que haga el mount de la imagen y podamos copiar todos los archivos a una carpeta que llamaremos "c:\SP2013", usamos el siguiente comando "xcopy F:* c:\SP2013 /s /e" donde F es la letra donde quedó la imagen montada. Vamos a la carpeta "c:\sp2013\prerequisiteinstallerfiles" y ahí copiamos los prerequisites que descargamos en el punto 7.
10. Ahora con otro script desarrollado por Craig Lussier, procederemos a la instalación de prerequisites de software que requiere SharePoint 2013. Ejecutamos el script "Install-SP2013PreReqFiles.ps1" y nos pedirá la ruta donde está el instalador de SharePoint 2013 que copiamos en el punto 9.

```
param([string] $SharePoint2013Path = $(Read-Host -Prompt "Please enter the
directory path to where your SharePoint 2013 installation files exist."))

function InstallPreReqFiles()
{
    $ReturnCode = 0

    Write-Host ""
    Write-Host
    "=====
    Write-Host "Installing Prerequisites required for SharePoint 2013"
    Write-Host ""
    Write-Host "This uses the supported installing offline method"
    Write-Host ""
    Write-Host "If you have not installed the necessary Roles/Features"
    Write-Host "this will occur at this time."
    Write-Host
    "=====

    Try
    {
        Start-Process "$SharePoint2013Path\PrerequisiteInstaller.exe" -
ArgumentList "`
/SQLNcli:`"$SharePoint2013Path\PrerequisiteInstallerFiles\sqlncli.msi`" `
/IDFX:`"$SharePoint2013Path\PrerequisiteInstallerFiles\windows6.1-KB974405-
x64.msu`" `
/IDFX11:`"$SharePoint2013Path\PrerequisiteInstallerFiles\MicrosoftIdentityExtensio
ns-64.msi`" `
/Sync:`"$SharePoint2013Path\PrerequisiteInstallerFiles\Synchronization.msi`" `
/AppFabric:`"$SharePoint2013Path\PrerequisiteInstallerFiles\windowsServerAppFabric
Setup_x64.exe`" `
/KB2671763:`"$SharePoint2013Path\PrerequisiteInstallerFiles\AppFabric1.1-RTM-
KB2671763-x64-ENU.exe`"
```

```

/MSPCCClient: `"$SharePoint2013Path\PrerequisiteInstallerFiles\setup_msipc_x64.msi`

/WCFDataServices: `"$SharePoint2013Path\PrerequisiteInstallerFiles\wcfDataServices.
exe`""
    }
    Catch
    {
        $ReturnCode = -1
        Write-Error $_
        break
    }
    return $ReturnCode
}

function CheckProvidedSharePoint2013Path()
{
    $ReturnCode = 0
    Try
    {
        # Check if destination path exists
        If (Test-Path $SharePoint2013Path)
        {
            # Remove trailing slash if it is present
            $script:SharePoint2013Path = $SharePoint2013Path.TrimEnd('\')
            $ReturnCode = 0
        }
        Else {
            $ReturnCode = -1
            Write-Host ""
            Write-Warning "Your specified download path does not exist. Please verify
your download path then run this script again."
            Write-Host ""
        }
    }
    Catch
    {
        $ReturnCode = -1
        Write-Warning "An error has occurred when checking your specified
download path"
        Write-Error $_
        break
    }
    return $ReturnCode
}

function InstallPreReqs()
{
    $rc = 0
    $rc = CheckProvidedSharePoint2013Path

    # Install the Pre-Reqs
    if($rc -ne -1)

```

```
{  
    $rc = InstallPreReqFiles  
}  
  
if($rc -ne -1)  
{  
    Write-Host ""  
    Write-Host "Script execution is now complete!"  
    Write-Host ""  
}  
  
}
```

InstallPreReqs

11. El servidor se reinicia
12. Luego del reinicio, se hace unos últimos ajustes a la instalación. verificamos que los prerrequisitos estén instalados ejecutando el script **"4-VerificacionPreReqSP_PowerShell.ps1"**
13. Procederemos a instalar SharePoint. Verificamos la configuración con la que vamos a instalar SharePoint ejecutando el siguiente comando en una ventana de PowerShell que se esté ejecutando como administrador: **"get-content c:\LabFiles\5-ConfiguracionSWSP.xml"**
14. Luego procedemos a instalar SharePoint 2013:

start-process -wait -nonewwindow -passthru -filepath c:\sp2013\setup.exe -argumentlist "/config c:\labfiles\5-ConfiguracionSWSP.xml"

15. Una vez instalado, verificamos la instalación ejecutando el script:

"6-VerificarInstalacionSP_PowerShell.ps1"

16. Procedemos a realizar la encriptación de la contraseña del usuario que vamos a utilizar en la configuración de la granja. Ejecutamos el script:

"EncriptarPassword7Script.ps1"

17. Se procede con la configuración de la granja de SharePoint. Ejecutamos el archivo

"7-ConfigurarGranjaSP_PowerShell.ps1"

18. Una vez instalado, procedemos a abrir la administración central para realizar las pruebas pertinentes. Otra forma de verificar es ejecutar el script

"8-VerificacionGranjaSP_Powershell.ps1"

Ejercicio 4

Administrar SharePoint 2013 usando PowerShell

- Administrando usuarios y grupos
- Administrando la granja
- Administrando aplicaciones de servicio y características

Administrando usuarios y grupos

1. Crear un usuario de directorio activo usando PowerShell.
 - a. Abrir una ventana de PowerShell en el servidor controlador de dominio
 - b. Ejecutar la instrucción **Import-Module ActiveDirectory** para que la ventana de PowerShell cargue la información de los cmdlets de directorio activo
 - c. Ejecutar la instrucción **get-help New-ADUser -example** para traer la ayuda de PowerShell respecto al cmdlet New-ADUser y que adicionalmente traiga los ejemplos asociados al cmdlet. Si les aparece un mensaje que si quieren actualizar la ayuda, le dicen que si "y".
 - d. Una vez actualizados los cmdlets, procedemos a crear una llave de encriptación para las contraseñas del usuario. Digitamos **\$Key = (1..24) y enter**
 - e. Sin salir de la ventana de PowerShell. Procedemos a cargar la clave encriptada que creamos en el ejercicio anterior y que esta almacenada en el archivo c:\Labfiles\Password.txt. para esto, ejecutamos la siguiente instrucción: **\$encrypted = Cat c:\Labfiles\Password.txt | ConvertTo-SecureString -Key \$Key**
 - f. Sin salir de la ventana de PowerShell, vamos a crear el usuario SP_TestAdmin ejecutando la siguiente instrucción: **New-ADUser -Name SP_TestAdmin -UserPrincipalName sp_testadmin@iberianspconf.local -path "ou=USERS,ou=SHAREPOINT2013,dc=iberianspconf,dc=local" -AccountPassword \$encrypted -PassThru | Enable-ADAccount**
 - g. Sin salir de la ventana de PowerShell, agregamos el usuario al grupo de administradores locales con la siguiente instrucción: **Add-ADGroupMember Administrators SP_TestAdmin**
 - h. Para salir del ventana de PowerShell ejecutamos **Exit**

Administrando la Granja

2. Creando un usuario el SharePoint 2013 como administrador de la granja
 - a. Abrir una ventana de PowerShell en el servidor de SharePoint
 - b. Ejecutar la instrucción **Add-PSSnapin Microsoft.SharePoint.PowerShell** para que la ventana de PowerShell cargue la información de los cmdlets de SharePoint 2013
 - c. Crear una variable para guardar la ruta donde está el grupo de administradores de la granja de SharePoint 2013. Ejecutamos la siguiente instrucción:


```
$farmAdminsGroup=(Get-SPWebApplication -IncludeCentralAdministration | Where {$_.DisplayName -like "*SharePoint Central Administration*"}).Sites[0].RootWeb.SiteGroups["Farm Administrators"]
```
 - d. Adicionamos el usuario creado en el ejercicio anterior para que sea administrador de la granja. Ejecutamos la siguiente instrucción:


```
$farmAdminsGroup.AddUser("iberianspconf\SP_TestAdmin" , "" , " iberianspconf\SP_TestAdmin" , "")
```
 - e. Ejecutamos solo la variable para identificar que el usuario es parte del grupo administradores de la granja: **\$farmAdminsGroup**
 - f. Sin salir de la ventana, procedemos a darle acceso SharePoint_Shell_Access al usuario para completar los permisos de administrador de la granja. Ejecutamos **Add-SPShellAdmin iberianspconf\SP_TestAdmin**. Para verificar que el usuario tiene los permisos necesarios, solo resta ejecutar **Get-Shelladmin**
 - g. Sin salir de la ventana, procedemos a darle acceso SharePoint_Shell_Access a la aplicación web que creamos en el proceso de instalación de SharePoint 2013 al usuario **SP_TestAdmin**. Ejecutamos **Get-SPContentDatabase -WebApplication http://nombredelservidor:28000 | Add-SPShellAdmin iberianspconf\SP_TestAdmin**.
 - h. Ahora procederemos a crear el usuario como sysadmin de SQL server desde PowerShell. Entramos al servidor de SQL Server y abrimos una ventana de Windows PowerShell. Lo primero que debemos cargar son los cmdlets de SQL Server con la siguiente instrucción: **import-module SQLPS**. Si les aparece el siguiente mensaje **"cannot be loaded because running scripts is disabled on this system."**, solo deben ejecutar la instrucción **Set-ExecutionPolicy remotesigned**
 - i. Sin salir de la ventana, ejecutamos la instrucción: **Invoke-SQLCMD "sp_addsrvrolemember 'iberianspconf\SP_TestAdmin','sysadmin' "**
 - j. Para salir de la ventana de PowerShell ejecutamos **Exit**

Administrando Aplicaciones de Servicio y Características

- a. Abrir una ventana de PowerShell en el servidor de SharePoint
- b. Ejecutar la instrucción **Add-PSSnapin Microsoft.SharePoint.PowerShell** para que la ventana de PowerShell cargue la información de los cmdlets de SharePoint 2013
- c. Creamos un nuevo pool de aplicaciones con al siguiente instrucción:
New-SPServiceApplicationPool -Name "Excel Web Services" -Account iberianspconf\usuarioadministrador
- d. Creamos la aplicación de servicio de Excel ejecutando la siguiente instrucción:
New-SPEExcelServiceApplication -Name "Excel Web Application" -ApplicationPool "Excel Web Services"
- e. Iniciamos la instancia del servicio de Excel Calculation con la siguiente instrucción
Get-SPServiceInstance | Where {\$_.TypeName -like "Excel Calculation Services"} | Start-SPServiceInstance
- f. Verificamos que la instancia se haya iniciado:
Get-SPServiceInstance | FT TypeName, Status -Auto
- g. Asignamos a una variable la Información de Excel Web Application to a variable:
\$EWA = Get-SPServiceApplication -Name "Excel Web Application"
- h. Buscamos la información de seguridad de los administradores de la aplicación de servicio Excel Web Application con la siguiente instrucción:
\$AdminEWA = \$EWA | Get-SPServiceApplicationSecurity -Admin
- i. Usamos otra variable para almacenar las credenciales del usuario que será el administrador de la aplicación de servicio de Excel Web application:
\$Principal = New-SPClaimsPrincipal "iberianspconf\SP_TestAdmin" -IdentityType WindowsSamAccountName
- j. Adicionamos el usuario al objeto donde creamos la seguridad de la aplicación de servicio (punto h de este ejercicio) de Excel Web Application:
Grant-SPObjSecurity \$AdminEWA \$Principal "Full Control"
- k. Actualizamos el objeto de seguridad para la aplicación de servicio de Excel Web Application con las nuevas credenciales:
Set-SPServiceApplicationSecurity \$EWA \$AdminEWA -Admin
- l. Hacemos la respectiva verificación que el usuario fue asignado con los permisos correspondientes en la aplicación de servicio de Excel Web Application:
Get-SPServiceApplication -Name "Excel Web Application" | Get-SPServiceApplicationSecurity -Admin

Ejercicio 5

Realizar tareas de mantenimiento de SharePoint 2013 usando PowerShell

- Backups / Restores
- Monitoreo y Auditorias

Backups/Restores

- Se debe instalar los siguientes 3 módulos para poder usar el PowerShell de SQL en el servidor de SharePoint
 - Microsoft® System CLR Types for Microsoft® SQL Server® 2012 (SQLSysClrTypes.msi)
 - Microsoft® SQL Server® 2012 Shared Management Objects (SharedManagementObjects.msi)
 - Microsoft® Windows PowerShell Extensions for Microsoft® SQL Server® 2012 (PowerShellTools.msi)
- Abrimos una ventana de Windows PowerShell y ejecutamos las siguientes instrucciones:

Add-PSSnapin Microsoft.SharePoint.PowerShell; Import-Module SQLPS

- Para hacer un backup de la granja, solo debemos ejecutar la siguiente instrucción:
Backup-SPFarm -Directory c:\LabFiles\Backup -ConfigurationOnly -Backupmethod Full -Verbose
- Para hacer un backup de una colección de sitios <http://nombredelservidor:28000>:
Backup-SPSite http://nombredelservidor:28000 -Path c:\LabFiles\Backup\28000.bak -UseSQLSnapshot -Verbose -Force
- Verificamos que los backups se hayan realizado en el directorio especificado
Get-ChildItem c:\LabFiles\Backup
- Para eliminar una colección de sitios <http://nombredelservidor:28000>
Remove-SPSite http://nombredelservidor:28000
- Intentar abrir el sitio web para verificar si existe
Start-Process <http://nombredelservidor:28000>
- Cerrar el sitio web
- Ahora usamos el backup que realizamos del sitio web para hacer la restauración del mismo:
Restore-SPSite http://nombredelservidor:28000 -Path c:\LabFiles\Backup\28000.bak -DatabaseName WSS_Content -Force
- Volvemos a abrir el sitio web y verificamos que la operación de restauración se haya ejecutado con éxito.
Start-Process <http://nombredelservidor:28000>
- Cerrar el sitio web

Monitoreo y auditoria

- a. Abrimos una ventana de Windows PowerShell y ejecutamos las siguientes instrucciones:

Add-PSSnapin Microsoft.SharePoint.PowerShell

- b. Ejecutamos la siguiente instrucción para listar el nombre de las base de datos de configuración y contenido de SharePoint

Get-SPDatabase | FT Name,Type,WebApplication,Server -Auto

- c. Para listar todas las propiedades globales de la granja de SharePointthe

Get-SPFarmConfig

- d. Exportar la línea base de cómo está configurada la granja de SharePoint a un archive xml

Get-SPFarmConfig | Export-Clixml c:\Labfiles\SPFarmConfig_Baseline.xml -Depth 4

- e. Para guardar las propiedades de la granja a una variable:

\$SPFarmConfig = Get-SPFarmConfig

- f. Para modificar una de las propiedades de la granja usando la variable anterior:

\$SPFarmconfig.WorkflowBatchSize=120

- g. Aplicar los cambios

\$SPFarmConfig | Set-SPFarmConfig

- h. Exportar los cambios a un archivo xml

Get-SPFarmConfig | Export-Clixml c:\Labfiles\SPFarmConfig_Current.xml -Depth 4

- i. Guardar la línea base de la configuración a una variable

\$Baseline = Get-Content c:\Labfiles\SPFarmConfig_Baseline.xml

- j. Guardar la configuracion actual a una variable

\$Current = Get-Content c:\Labfiles\SPFarmConfig_Current.xml

- k. Vamos a comparar ambas variables para ver la diferencia en los valores que han cambiado

Compare-Object \$Baseline \$Current

- l. La información de salida muestra que la propiedad "WorkflowBatchSize" ha cambiado de 100 a 120.

Otros cmdlets de interés:

- Get-SPHealthAnalysisRule
- Get-SPTimerJob
- Get-SPDiagnosticsPerformanceCounter

Ejercicio 6

Office 365 PowerShell

- Instalar Microsoft Online Services Sign-In Assistant for IT Professionals BETA en el equipo desde donde quieran administrar Office 365. Lo encuentran en `c:\LabFiles\PowerShell for O365\msoidcli_64.msi` o lo pueden descargar de la siguiente url.
<http://www.microsoft.com/en-us/download/details.aspx?id=39267>
- Instalar el Azure AD Modulo en el equipo desde donde quieran administrar Office 365. Lo encuentran en `c:\LabFiles\PowerShell for O365\AdministrationConfig-en.msi` o lo pueden descargar de la siguiente url.
<http://go.microsoft.com/fwlink/p/?linkid=236297>
- Para conectarse al trial de O365 debemos seguir los siguientes pasos:
 - Abrir Windows Azure Active Directory Module for Windows PowerShell
 - Ejecutar la instrucción **Connect-MsolService**. Se abrirá una ventana emergente donde se deben digitar las credenciales de acceso al trial de O365.
- Digitamos la instrucción **Get-MsolAccountSku** para ver el plan que tenemos asignado en nuestro tenant de O365.
- Para ver los usuarios activos actuales, ejecutar la siguiente instrucción: **get-msoluser**
- Para crear un usuario, ejecutar la siguiente instrucción:

```
New-MsolUser -DisplayName "Pruebas Iberian" -FirstName Pruebas -LastName  
Sills -UserPrincipalName piberian@tudominio.onmicrosoft.com -Department  
Operations -UsageLocation CO -PassWord TempP@@sW0rd -LicenseAssignment  
tudominio:ENTERPRISEPACK
```

- Para listar el total de usuarios licenciados:
Get-MsolUser | Sort islicensed | group islicensed
- Más Información de usuarios, aquí:
<https://support.office.com/en-ca/article/Add-users-to-Office-365-with-Windows-PowerShell-ca91a94f-8aeb-4f9a-b6db-24db4e14ec21>

Otros cmdlets para Office 365:

Get-MsolAccountSku
Get-MsolAdministrativeUnit
Get-MsolAdministrativeUnitMember
Get-MsolCompanyInformation
Get-MsolContact
Get-MsolDirSyncConfiguration
Get-MsolDirSyncProvisioningError
Get-MsolDomain
Get-MsolDomainFederationSettings
Get-MsolDomainVerificationDns
Get-MsolFederationProperty
Get-MsolGroup
Get-MsolGroupMember
Get-MsolPartnerContract
Get-MsolPartnerInformation
Get-MsolPasswordPolicy
Get-MsolRole
Get-MsolRoleMember
Get-MsolScopedRoleMember
Get-MsolServicePrincipal
Get-MsolServicePrincipalCredential
Get-MsolSubscription
Get-MsolUser
Get-MsolUserByStrongAuthentication
Get-MsolUserRole

PowerShell para SharePoint Online:

- a. Instalar el SharePoint Online Management Shell en el equipo desde donde quieran administrar SharePoint Online. Lo encuentran en c:\LabFiles\PowerShell for O365\sharepointonlinemanagementshell_4017-1200_x64_en-us.msi o lo pueden descargar de la siguiente url.
<https://www.microsoft.com/en-us/download/details.aspx?id=35588>
- b. Para conectarse al tenant de SharePoint Online, se ejecuta la siguiente instrucción:
Connect-SPOService -Url "administration site" -credential
admin@tudominio.onmicrosoft.com

Mas Información de PowerShell para SharePoint Online:

<https://technet.microsoft.com/en-us/library/fp161397.aspx>