

## Laboratorio 1 – Paradigmas emergentes en la Computación en la nube

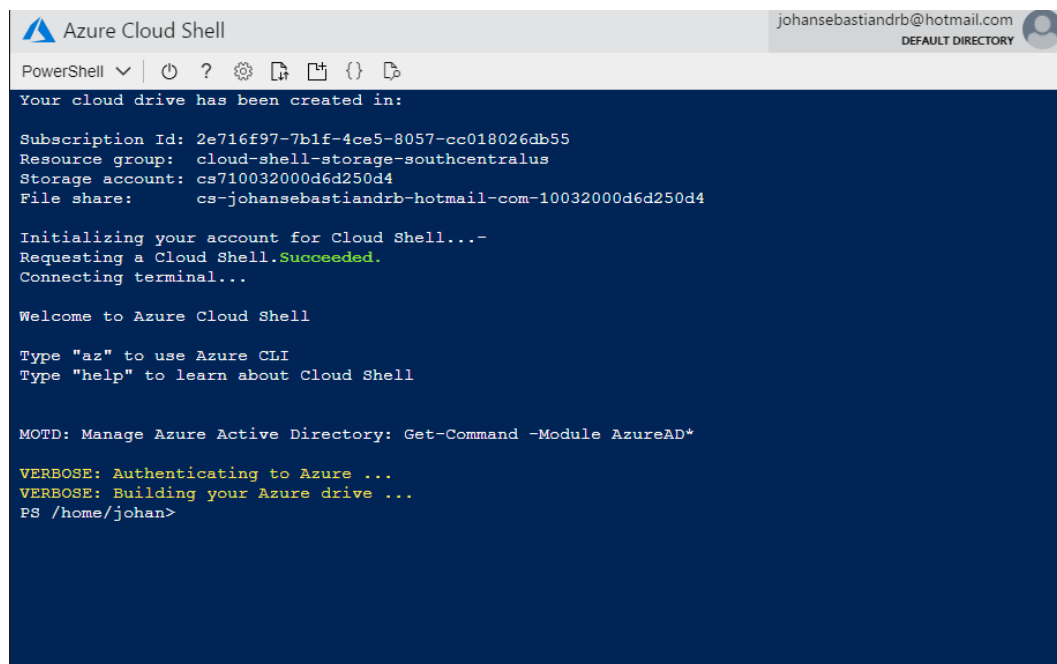
A continuación, presentaré el proceso de despliegue por diferentes métodos de máquinas virtuales en Microsoft Azure. En estos se describirá un paso a paso con su respectivo output, las implementaciones se realizaron tanto para Windows como para Linux con sus respectivos servicios web (IIS y Nginx).

### Windows:

#### a. Azure Command Line Interface Method

1. Crear el grupo de recursos donde se alojarán los componentes de nuestro despliegue eligiendo una localización optima de acuerdo con la mejor latencia obtenida:

Nota: para usar la cloud Shell de Azure es necesario tener un grupo de recursos el cual almacena los comandos que ejecutamos.



```
Azure Cloud Shell
johansebastiandr@hotmai.com
DEFAULT DIRECTORY

PowerShell | ? | Settings | File Explorer | Code | Help
Your cloud drive has been created in:

Subscription Id: 2e716f97-7b1f-4ce5-8057-cc018026db55
Resource group: cloud-shell-storage-southcentralus
Storage account: cs710032000d6d250d4
File share: cs-johansebastiandr@hotmai-com-10032000d6d250d4

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

Welcome to Azure Cloud Shell

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

MOTD: Manage Azure Active Directory: Get-Command -Module AzureAD*

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/johan>
```

Comando para crear el grupo de recursos:

`az group create --name <resourceGroupName> --location <resourceGroupLocation>`

```
PS /home/johan> az group create --name myResourceGroup --location eastus
{
  "id": "/subscriptions/2e716f97-7b1f-4ce5-8057-cc018026db55/resourceGroups/myResourceGroup",
  "location": "eastus",
  "managedBy": null,
  "name": "myResourceGroup",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
PS /home/johan>
```

2. Crear la máquina virtual indicando el nombre que tendrá esta máquina, además indicaremos el nombre de usuario con el que nos conectaremos, la imagen del sistema operativo y el grupo de recursos asociado que creamos en el paso anterior.

Comando para crear la máquina virtual:

```
az vm create \
  --resource-group myResourceGroup \
  --name myVM \
  --image win2016datacenter \
  --admin-username azureuser
```

```
PS /home/johan> az vm create --resource-group myResourceGroup --name windowsIIS --image win2019datacenter --admin-username azureuser
Admin Password:
Confirm Admin Password:
{- Finished ..
  "fqdns": "",
  "id": "/subscriptions/2e716f97-7b1f-4ce5-8057-cc018026db55/resourceGroups/myResourceGroup/providers/Microsoft.Compute/virtualMachines/windowsIIS",
  "location": "eastus",
  "macAddress": "00-22-48-27-C1-6F",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "13.68.241.206",
  "resourceGroup": "myResourceGroup",
  "zones": ""
}
```

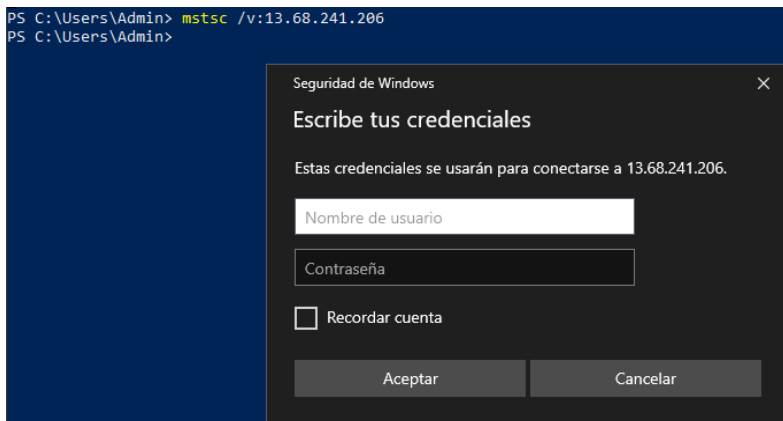
- Lo siguiente que debemos hacer es habilitar el tráfico por el puerto 80 para que se pueda consultar nuestro sitio web que crearemos más adelante. Esto es necesario ya que por defecto solo se permite tráfico proveniente del puerto 3380 (RDP).

```
az vm open-port --port 80 --resource-group myResourceGroup --name myVM
```

```
PS /home/johan> az vm open-port --port 80 --resource-group myResourceGroup --name windowsIIS
{- Finished ..
  "defaultSecurityRules": [
    {
      "access": "Allow",
      "description": "Allow inbound traffic from all VMs in VNET",
      "destinationAddressPrefix": "VirtualNetwork",
      "destinationAddressPrefixes": [],
      "destinationApplicationSecurityGroups": null,
      "destinationPortRange": "*",
      "destinationPortRanges": [],
      "direction": "Inbound",
      "etag": "W/\"cdce65d1-4d1e-4851-b760-6dc7ad6e4d71\"\"",
      "id": "/subscriptions/2e716f97-7b1f-4ce5-8057-cc018026db55/resourceGroups/myResourceGroup/providers/Microsoft.Network/networkSecurityGroups/defaultSecurityRules/AllowVnetInBound",
      "name": "AllowVnetInBound",
      "priority": 65000,
      "protocol": "*",
      "provisioningState": "Succeeded",
      "resourceGroup": "myResourceGroup",
      "sourceAddressPrefix": "VirtualNetwork",
      "sourceAddressPrefixes": [],
      "sourceApplicationSecurityGroups": null,
      "sourcePortRange": "*",
      "sourcePortRanges": [],
      "type": "Microsoft.Network/networkSecurityGroups/defaultSecurityRules"
    }
  ],
}
```

- Una vez hayamos instalado todo podremos conectarnos desde nuestra máquina local con el siguiente comando:

```
mstsc /v: publicIpAddress
```



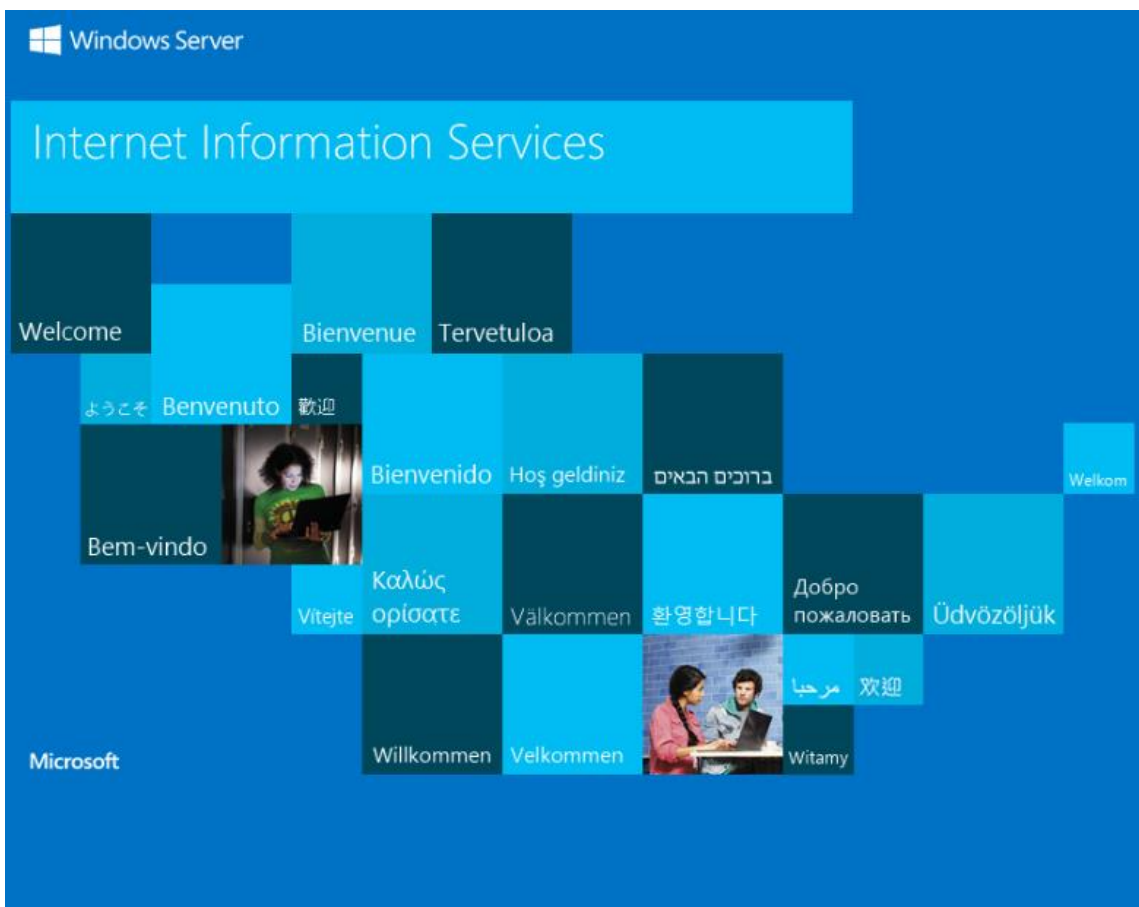
5. Por último instalaremos el servicio que va alojar nuestro servidor web, en este caso Internet Information Services (IIS) :

```
PS C:\Users\azureuser> Install-WindowsFeature -name Web-Server -IncludeManagementTools

Success Restart Needed Exit Code      Feature Result
-----
True      No          Success      {Common HTTP Features, Default Document, D...

PS C:\Users\azureuser> echo "Machine done with Azure Cli"
Machine done with Azure Cli
PS C:\Users\azureuser>
```

Verificamos que esté corriendo:



## b. Azure portal method

1. En primer lugar, debemos crear nuestra máquina virtual, podemos decidir si asociarla con un grupo de recursos ya creado o podemos crear un grupo de recursos nuevo. Es necesario diligenciar el formulario el cual contiene información como:
  - Nombre de la máquina virtual.
  - Localización de nuestra máquina.
  - Tipo de disponibilidad (en caso de que necesitemos que la máquina nunca esté fuera de servicio).
  - Imagen del sistema operativo.
  - Tamaño, el cual determina la capacidad del procesador, de memoria y de almacenamiento.

### Crear una máquina virtual

[Datos básicos](#)   [Discos](#)   [Redes](#)   [Administración](#)   [Opciones avanzadas](#)   [Etiquetas](#)   [Revisar y crear](#)

Cree una máquina virtual que ejecuta Linux o Windows. Seleccione una imagen de Azure Marketplace o use una imagen personalizada propia. Complete la pestaña Conceptos básicos y, después, use Revisar y crear para aprovisionar una máquina virtual con parámetros predeterminados o bien revise cada una de las pestañas para personalizar la configuración.  
[Más información](#)

#### Detalles del proyecto

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción \* ⓘ

Grupo de recursos \* ⓘ   
[Crear nuevo](#)

#### Detalles de instancia

Nombre de máquina virtual \* ⓘ

Región \* ⓘ

Opciones de disponibilidad ⓘ

Imagen \* ⓘ   
[Examinar todas las imágenes públicas y privadas](#)

Tamaño \* ⓘ   
[Seleccionar tamaño](#)

## 2. Definir la cuenta administradora y los puertos que tendrán acceso publico

### Crear una máquina virtual

Imagen \*

Windows Server 2019 Datacenter

[Examinar todas las imágenes públicas y privadas](#)

Tamaño \*

Standard\_D2s\_v3 - 2 vcpu, 8 GiB de memoria (USD 137.24/mes)

[Seleccionar tamaño](#)

Cuenta de administrador

Nombre de usuario \*

johanArias

Contraseña \*

\*\*\*\*\*

Confirmar contraseña \*

\*\*\*\*\*

Reglas de puerto de entrada

Seleccione los puertos de red de máquina virtual que son accesibles desde la red Internet pública. Puede especificar acceso de red más limitado o granular en la pestaña Red.

Puertos de entrada públicos \*

☐ Ninguno ☒ Permitir los puertos seleccionados

Seleccionar puertos de entrada \*

HTTP (80), RDP (3389)

⚠ Esto permitirá que todas las direcciones IP accedan a la máquina virtual. Esto solo se recomienda para las pruebas. Use los controles avanzados de la pestaña Redes a fin de crear reglas para limitar el tráfico entrante a las direcciones IP conocidas.

Licencias

Ahorre hasta un 49 % con una licencia de su propiedad con Ventaja híbrida de Azure. [Más información](#)

¿Ya tiene una licencia de Windows Server?

☐ Sí ☒ No

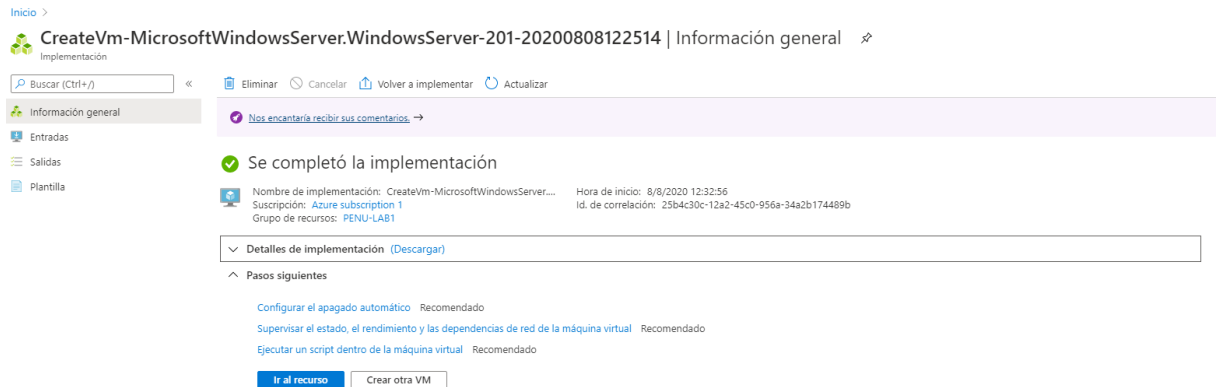
[Revisar conformidad de la Ventaja híbrida de Azure](#)

Revisar y crear

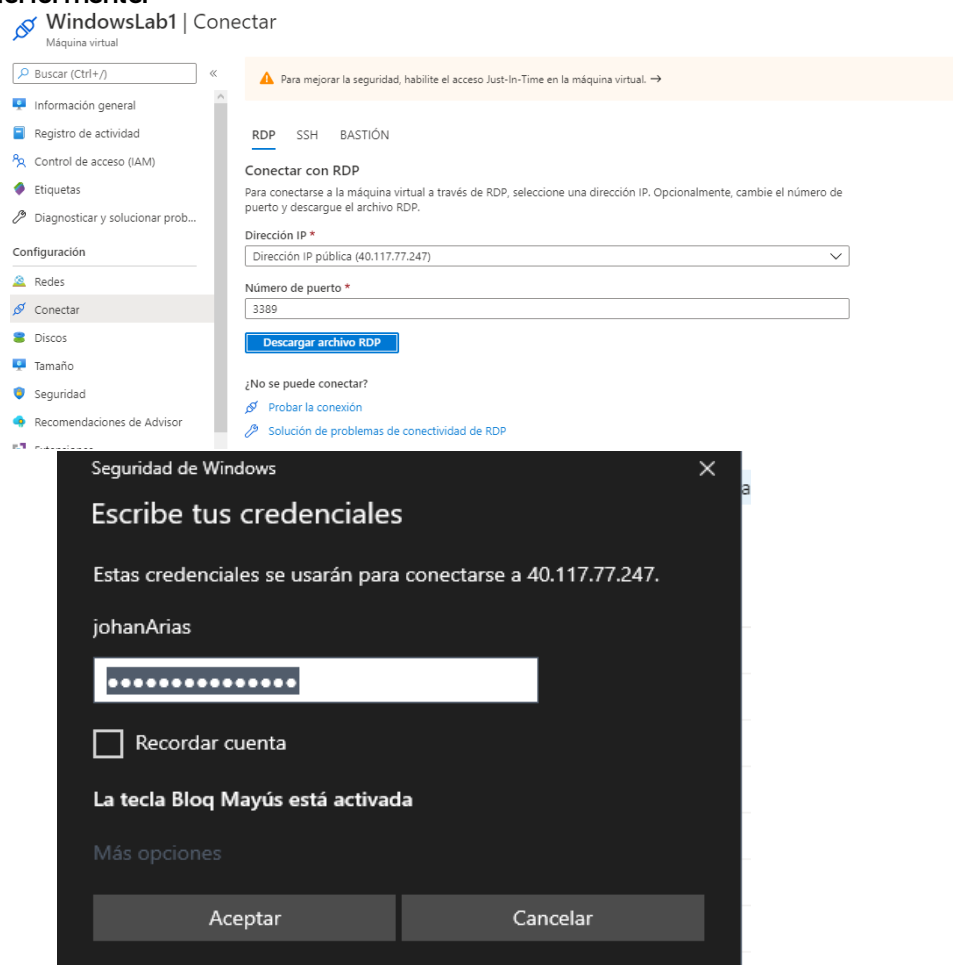
< Anterior

Siguiente: Discos >

### 3. Una vez creados el grupo de recursos y sus elementos veremos un panel de confirmación



### 4. Luego debemos conectarnos mediante el protocolo RDP (Remote Desktop Protocol) descargando un archivo .rdp e introduciendo las credenciales del usuario que creamos anteriormente.



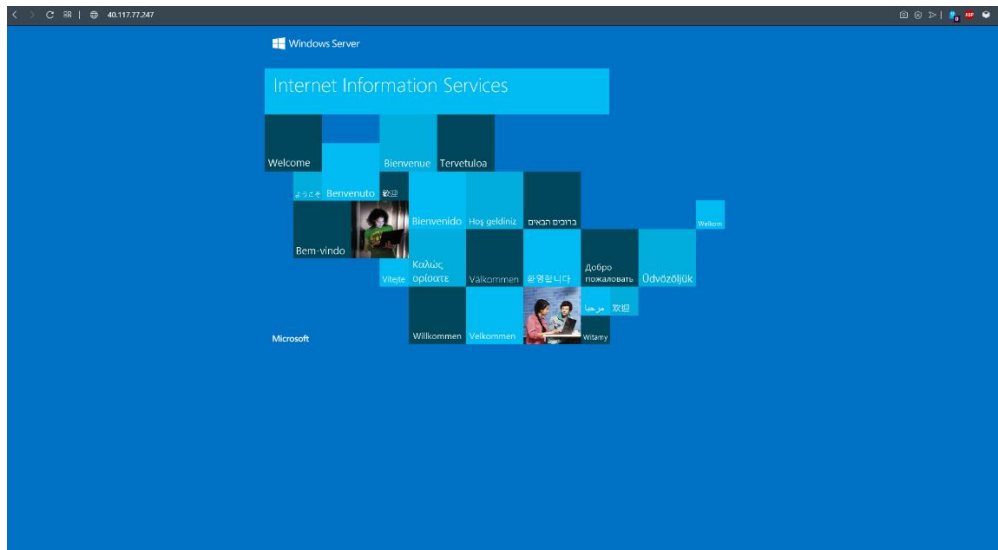
## 5. Instalamos IIS desde Powershell y verificamos que corra desde una máquina remota.

```
Select Administrator: Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\johanArias> Install-WindowsFeature -name Web-Server -IncludeManagementTools

Success Restart Needed Exit Code      Feature Result
-----
True      No              Success      {Common HTTP Features, Default Document, D...
```



## c. Powershell method

1. Para crear un grupo de recursos mediante PowerShell debemos ejecutar el siguiente comando especificando el nombre del grupo de recursos y su localización.

`New-AzResourceGroup -Name myResourceGroup -Location EastUS`

```
PS /home/johan> New-AzResourceGroup -Name PENU -Location EastUS

ResourceGroupName : PENU
Location           : eastus
ProvisioningState  : Succeeded
Tags               :
ResourceId          : /subscriptions/2e716f97-7b1f-4ce5-8057-cc018026db55/resourceGroups/PENU
```



2. Lo siguiente que haremos es crear la máquina virtual en conjunto con otros recursos como lo son:

- Red virtual.

```
PS /home/johan> New-AzVm `
>> -ResourceGroupName "PENU" `
>> -Name "chanVm" `
>> -Location "East US" `
>> -VirtualNetworkName "myVnet" `
>> -SubnetName "mySubnet" `
>> -SecurityGroupName "myNetworkSecurityGroup" `
>> -PublicIpAddressName "myPublicIpAddress" `
>> -OpenPorts 80,3389 `
>>

cmdlet New-AzVM at command pipeline position 1
Supply values for the following parameters:
Credential
User: johanArias
Password for user johanArias: *****

ResourceGroupName      : PENU
Id                     : /subscriptions/2e716f97-7b1f-4ce5-8057-cc018026db55/resourceGroups/PENU/
providers/Microsoft.Compute/virtualMachines/chanVm
VmId                   : 51612270-3081-4854-a0d9-3daa1cf6d2d2
Name                   : chanVm
Type                   : Microsoft.Compute/virtualMachines
Location               : eastus
Tags                   : {}
HardwareProfile         : {VmSize}
NetworkProfile          : {NetworkInterfaces}
OSProfile               : {ComputerName, AdminUsername, WindowsConfiguration, Secrets,
AllowExtensionOperations, RequireGuestProvisionSignal}
ProvisioningState       : Succeeded
StorageProfile          : {ImageReference, OsDisk, DataDisks}
FullyQualifiedDomainName : chanvm-ccf19a.East US.cloudapp.azure.com

PS /home/johan> 
```

- Subnet.
- Grupo de seguridad.
- Dirección IP pública.
- Puertos abiertos.

Luego de ejecutar este comando se crearán esta serie de recursos de manera automática.

3. Luego de tener todos los recursos creados podremos conectarnos a la máquina. Para consultar la ip pública de nuestra máquina podemos ejecutar el siguiente comando en Powershell:

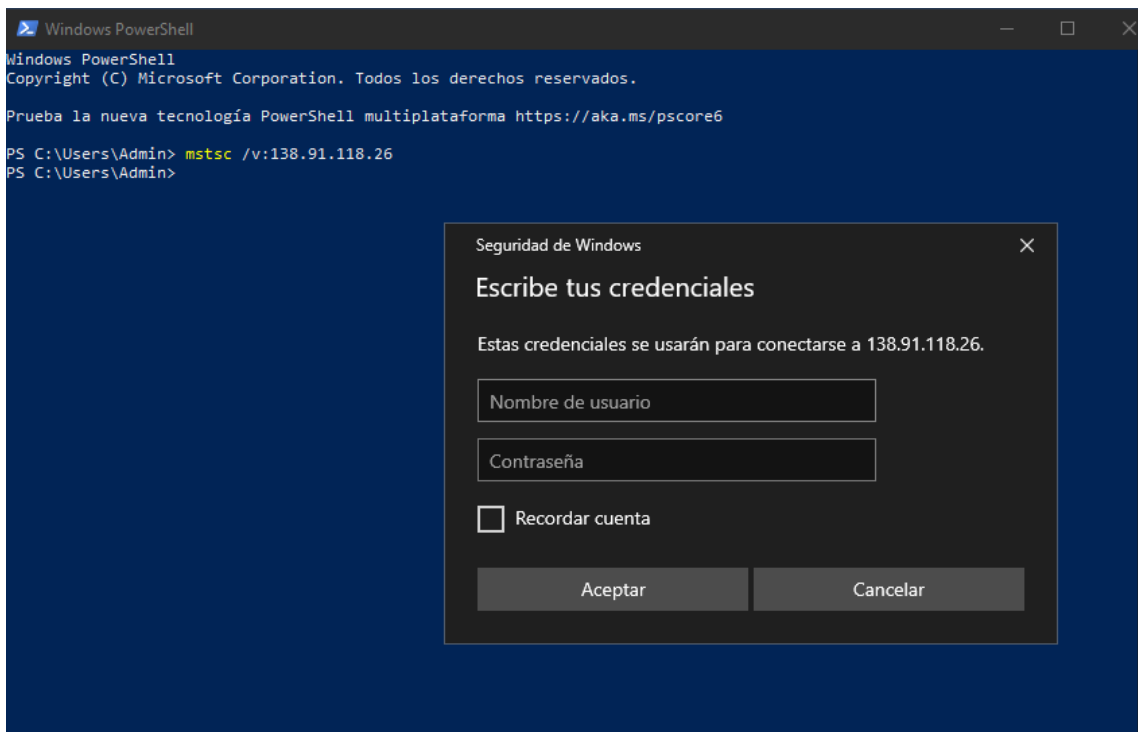
```
Get-AzPublicIpAddress -ResourceGroupName "myResourceGroup" | Select "IpAddress"
```

```
PS /home/johan> Get-AzPublicIpAddress -ResourceGroupName "PENU" | Select "IpAddress"

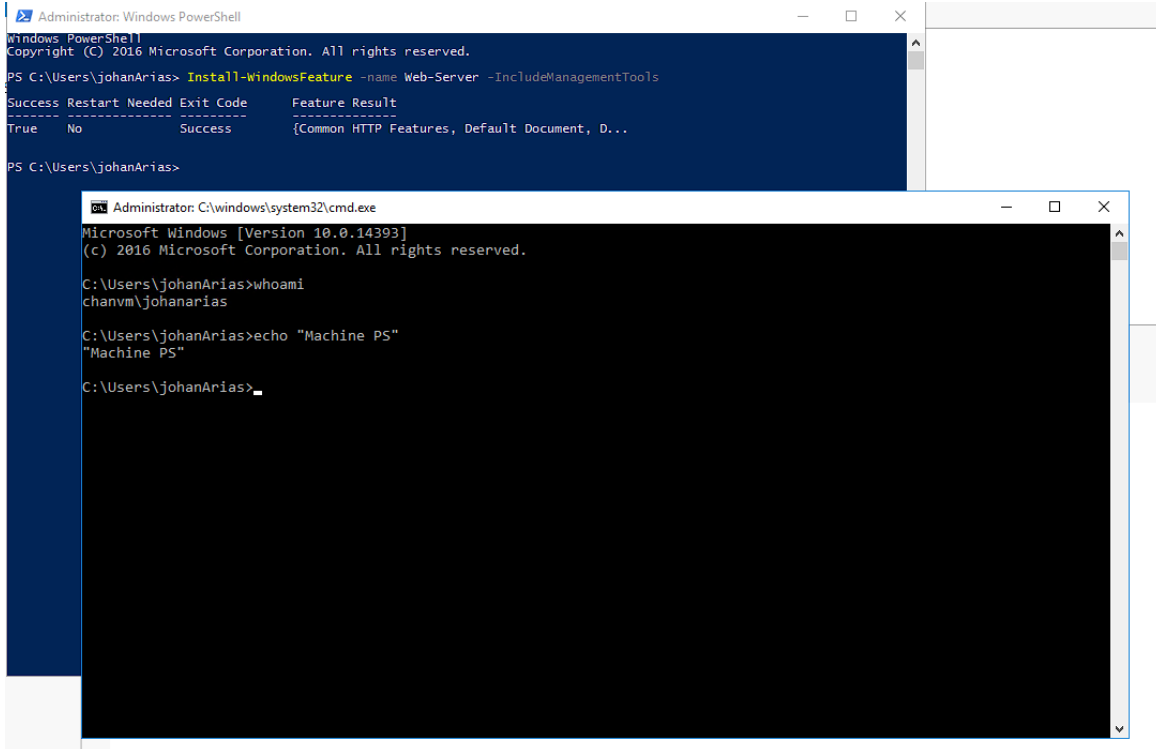
IpAddress
-----
138.91.118.26

PS /home/johan> 
```

Luego usamos mstsc para crear una conexión a una sesión remota en este caso a nuestra máquina virtual.



4. Finalmente instalamos IIS mediante Powershell y verificamos que sea accesible desde un lugar remoto.



The screenshot shows two overlapping windows. The top window is an Administrator Windows PowerShell terminal. It displays the command `Install-WindowsFeature -name Web-Server -IncludeManagementTools` and its output, which indicates a successful installation of the Web-Server feature. The bottom window is an Administrator Command Prompt (cmd.exe) showing the execution of `whoami` (returning `chanvm\johanarias`) and `echo "Machine PS"` (returning `"Machine PS"`).

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\johanArias> Install-WindowsFeature -name Web-Server -IncludeManagementTools
Success Restart Needed Exit Code      Feature Result
-----
True      No             Success      {Common HTTP Features, Default Document, D...

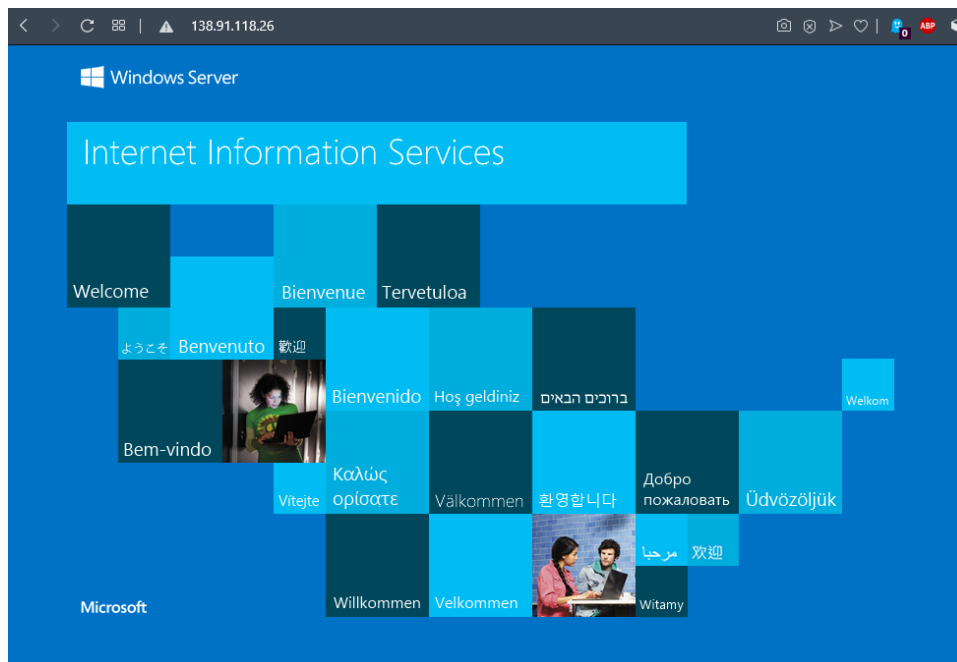
PS C:\Users\johanArias>
```

```
Administrator: C:\windows\system32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\johanArias> whoami
chanvm\johanarias

C:\Users\johanArias> echo "Machine PS"
"Machine PS"

C:\Users\johanArias>
```







## d. ARM (Azure resource manager) template method

1. Primero debemos acceder al siguiente enlace <https://portal.azure.com/#create/Microsoft.Template> en el cual va a estar alojada una plantilla en formato JSON, lo único que debemos hacer es llenar los datos necesarios y verificar la plantilla. Desde esta plantilla podemos identificar y modificar parámetros, variables y recursos.

Básico Review + create

Template

 101-vm-simple-windows   
6 resources

 Edit template  Edit parameters

Ámbito de la implementación

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción \* ⓘ Azure subscription 1 ✓

Grupo de recursos \* ⓘ (Nuevo) PenuLastW ✓  
[Crear nuevo](#)

Parámetros

Región \* ⓘ Este de EE. UU. ✓

Admin Username \* ⓘ johanArias ✓

Admin Password \* ⓘ ..... ✓

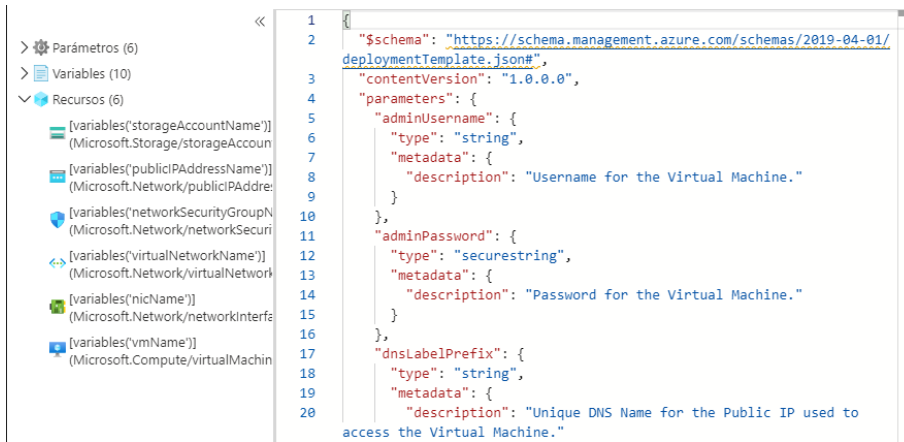
Dns Label Prefix \* ⓘ chanvmpenu ✓

Windows OS Version ⓘ 2019-Datacenter ✓

Vm Size ⓘ Standard\_D2\_v3

Location ⓘ [resourceGroup().location]

[Revisar y crear](#) < Anterior Siguiente: Revisar y crear >



```

1  {
2    "$schema": "https://schema.management.azure.com/schemas/2019-04-01/
deploymentTemplate.json#",
3    "contentVersion": "1.0.0.0",
4    "parameters": {
5      "adminUsername": {
6        "type": "string",
7        "metadata": {
8          "description": "Username for the Virtual Machine."
9        }
10     },
11     "adminPassword": {
12       "type": "securestring",
13       "metadata": {
14         "description": "Password for the Virtual Machine."
15       }
16     },
17     "dnsLabelPrefix": {
18       "type": "string",
19       "metadata": {
20         "description": "Unique DNS Name for the Public IP used to
access the Virtual Machine."

```

Verificamos los valores y creamos nuestro grupo de recursos.

#### Básico

Suscripción	Azure subscription 1
Grupo de recursos	PenuLastW
Región	Este de EE. UU.
Admin Username	johanArias
Admin Password	*****
Dns Label Prefix	chanvmpenu
Windows OS Version	2019-Datacenter
Vm Size	Standard_D2_v3
Location	[resourceGroup().location]

Crear

< Anterior

Siguiente

[Descargar una plantilla para la automatización](#)

- Por último, simplemente nos conectamos e instalamos nuestro IIS para crear nuestro servidor web.

RDP SSH BASTIÓN

**Conectar con RDP**

Para conectarse a la máquina virtual a través de RDP, seleccione una dirección IP. Opcionalmente, cambie el número de puerto y descargue el archivo RDP.

Dirección IP \*

Nombre DNS (chanvmpenu.eastus.cloudapp.azure.com)


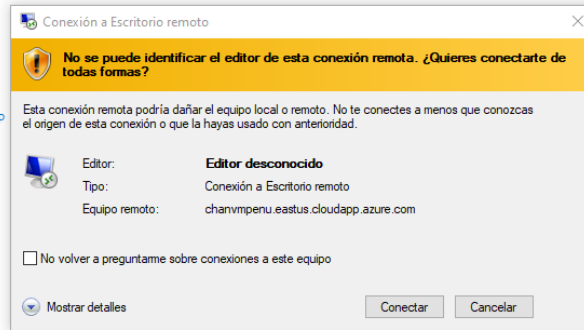
Número de puerto \*

3389

 Descargar archivo RDP
 

¿No se puede conectar?

 Probar la conexión

 Solución de problemas de conectividad de RDP


```
PS C:\Users\johanArias> Install-WindowsFeature -name Web-Server -IncludeManagementTools
```

Success	Restart Needed	Exit Code	Feature Result
True	No	Success	{Common HTTP Features, Default Document, D...

LINUX

## a. Azure command line interface method

1. En primer lugar debemos crear un grupo de recursos en el cual indicamos su localización

```
az group create --name myResourceGroup --location eastus
```

```
PS /home/johan> az group create --name penuLinux --location eastus
{
  "id": "/subscriptions/2e716f97-7b1f-4ce5-8057-cc018026db55/resourceGroups/penuLinux",
  "location": "eastus",
  "managedBy": null,
  "name": "penuLinux",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
```

## 2. Luego de esto creamos nuestra máquina virtual indicando:

- El nombre del grupo de recursos asociado.
- El nombre de la máquina virtual.
- La imagen del sistema operativo de nuestra máquina.
- El nombre del usuario administrador de la máquina.

Además, es necesario indicar que queremos generar llaves ssh para poder conectarnos luego

Comando : `az vm create \`  
`--resource-group myResourceGroup \`  
`--name myVM \`  
`--image UbuntuLTS \`  
`--admin-username azureuser \`  
`--generate-ssh-keys`

```
PS /home/johan> az vm create --resource-group penulinux --name linux1 --image UbuntuLTS --admin-username chan --generate-ssh-keys
SSH key files '/home/johan/.ssh/id_rsa' and '/home/johan/.ssh/id_rsa.pub' have been generated under ~/.ssh to allow SSH access to the VM. If using machines without permanent storage, back up your keys to a safe location.
{
  "fqdns": "",
  "id": "/subscriptions/2e716f97-7b1f-4ce5-8057-cc018026db55/resourceGroups/penulinux/providers/Microsoft.Compute/virtualMachines/linux1",
  "location": "eastus",
  "macAddress": "00-22-48-24-46-A0",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "40.76.209.17",
  "resourceGroup": "penulinux",
  "zones": ""
}
```

## 3. Ahora debemos habilitar el trafico por el puerto 80 para poder recibir peticiones a nuestro servidor web.

`az vm open-port --port 80 --resource-group myResourceGroup --name myVM`

```
PS /home/johan> az vm open-port --port 80 --resource-group penulinux --name linux1
{- Finished ..
"defaultSecurityRules": [
  {
    "access": "Allow",
    "description": "Allow inbound traffic from all VMs in VNET",
    "destinationAddressPrefix": "VirtualNetwork",
    "destinationAddressPrefixes": [],
    "destinationApplicationSecurityGroups": null,
    "destinationPortRange": "*",
    "destinationPortRanges": [],
    "direction": "Inbound",
    "etag": "W/\"911e9b2f-96e4-42cc-917e-78c97cc6b4a2\"",
    "id": "/subscriptions/2e716f97-7b1f-4ce5-8057-cc018026db55/resourceGroups/penulinux/providers/Microsoft.Network/networkSecurityGroups/linux1NSG/defaultSecurityRules/AllowVnetInBound",
    "name": "AllowVnetInBound",
    "priority": 65000,
    "protocol": "*",
    "provisioningState": "Succeeded",
    "resourceGroup": "penulinux",
    "sourceAddressPrefix": "VirtualNetwork",
    "sourceAddressPrefixes": [],
    "sourceApplicationSecurityGroups": null,
    "sourcePortRange": "*",
    "sourcePortRanges": [],
    "type": "Microsoft.Network/networkSecurityGroups/defaultSecurityRules"
  },

```

4. Lo siguiente que nos queda hacer es conectarnos a la máquina mediante ssh indicando el usuario administrador que creamos anteriormente

```
chan@linux1:~$ whoami
chan
chan@linux1:~$ echo "Machine made by azure cli"
Machine made by azure cli
chan@linux1:~$
```

5. Por último instalamos Nginx mediante el administrador de paquetes de Linux "apt" y verificamos que nuestro servidor web se pueda consultar de forma remota.

Comandos: `sudo apt-get -y update`  
`sudo apt-get -y install nginx`

```
chan@linux1:~$ systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-08-10 04:06:45 UTC; 1min 34s ago
     Docs: man:nginx(8)
   Main PID: 2709 (nginx)
    Tasks: 2 (limit: 4043)
   CGroup: /system.slice/nginx.service
           └─2709 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
              └─2712 nginx: worker process

Aug 10 04:06:45 linux1 systemd[1]: Starting A high performance web server and a reverse proxy server...
Aug 10 04:06:45 linux1 systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid: Invalid argument
Aug 10 04:06:45 linux1 systemd[1]: Started A high performance web server and a reverse proxy server.
chan@linux1:~$
```



40.76.209.17

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.


For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

### b. Azure portal method

#### 1. Crear un grupo de recursos, indicar información como:

- Nombre de la máquina virtual.
- Región de localización de nuestro grupo de recursos.
- Opciones de disponibilidad en caso de que sea un sistema que vaya a soportar mucho tráfico.
- Imagen del sistema operativo.
- Tamaño que representa la capacidad de procesamiento, memoria y almacenamiento.

Suscripción *	Azure subscription 1
Grupo de recursos *	(Nuevo) PenuLinux1
<a href="#">Crear nuevo</a>	
Detalles de instancia	
Nombre de máquina virtual *	LinuxManual
Región *	(US) Este de EE. UU.
Opciones de disponibilidad	Zona de disponibilidad
 Se recomienda una alta disponibilidad para las cargas de trabajo de producción.	
Zona de disponibilidad *	
Imagen *	Ubuntu Server 18.04 LTS
<a href="#">Examinar todas las imágenes públicas y privadas</a>	
Tamaño *	Standard_DS1_v2 - 1 vcpu, 3.5 GiB de memoria (USD 53.29/mes)
<a href="#">Seleccionar tamaño</a>	

 Se recomienda la serie D para las cargas de trabajo de uso general.

2. Debemos seleccionar un método de autenticación, para este caso lo haremos mediante ssh por lo que debemos indicar un nombre de usuario administrador, además tendremos que generar un par de llaves para realizar la conexión mediante este protocolo. De igual forma, es necesario habilitar el puerto 22 y el 80 para que se habilite el tráfico desde cualquier ip.

Cuenta de administrador

Tipo de autenticación ⓘ ☒ Clave pública SSH ☐ Contraseña

ⓘ Ahora, Azure genera automáticamente un par de claves SSH y le permite almacenarlo para usarlo en el futuro. Es una forma rápida, sencilla y segura de conectarse a la máquina virtual.

Nombre de usuario \* ⓘ  ✓

Origen de clave pública SSH  ▼

Nombre de par de claves \*  ✓

Reglas de puerto de entrada

Seleccione los puertos de red de máquina virtual que son accesibles desde la red Internet pública. Puede especificar acceso de red más limitado o granular en la pestaña Red.

Puertos de entrada públicos \* ⓘ ☐ Ninguno ☒ Permitir los puertos seleccionados

Seleccionar puertos de entrada \*  ▼

⚠ Esto permitirá que todas las direcciones IP accedan a la máquina virtual. Esto solo se recomienda para las pruebas. Use los controles avanzados de la pestaña Redes a fin de crear reglas para limitar el tráfico entrante a las direcciones IP conocidas.

3. Descargamos nuestra llave privada generada de formato .pem le damos permisos únicamente de lectura y de escritura (chmod 400 o 600 en Linux) y luego nos autenticamos mediante ssh.

```
[chan@parrot]~$ ssh -i LinuxManual key.pem johanArias@20.55.0.192
The authenticity of host '20.55.0.192 (20.55.0.192)' can't be established.
ECDSA key fingerprint is SHA256:kgU0xyF7ErTelle48upTiWam1/ocgB8zGYECvxULQYk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '20.55.0.192' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.3.0-1034-azure x86_64)

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

System information as of Mon Aug 10 06:01:29 UTC 2020

System load:  0.0          Processes:    108
Usage of /:   4.4% of 28.90GB Users logged in:  0
Memory usage: 8%          IP address for eth0: 10.0.0.4
Swap usage:  0%

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

johanArias@LinuxManual:~$
```

4. Por último instalamos Nginx mediante apt y verificamos que sea accesible desde una red remota.

```
johanArias@LinuxManual:~$ service nginx status
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-08-10 06:03:30 UTC; 53s ago
     Docs: man:nginx(8)
   Main PID: 2924 (nginx)
    Tasks: 2 (limit: 4043)
   CGroup: /system.slice/nginx.service
           └─2924 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─2927 nginx: worker process

Aug 10 06:03:29 LinuxManual systemd[1]: Starting A high performance web server and a reverse proxy server...
Aug 10 06:03:30 LinuxManual systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid: Invalid argument
Aug 10 06:03:30 LinuxManual systemd[1]: Started A high performance web server and a reverse proxy server.
```

20.55.0.192

## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

*Thank you for using nginx.*

### c. Powershell method

1. Generar un par de llaves ssh y crear un grupo de recursos indicando su localización:

```
ssh-keygen -m PEM -t rsa -b 4096
```

```
New-AzResourceGroup -Name "myResourceGroup" -Location "EastUS"
```

```
PS /home/johan> ssh-keygen -m PEM -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/johan/.ssh/id_rsa):
/home/johan/.ssh/id_rsa already exists.
Overwrite (y/n)? n
PS /home/johan> New-AzResourceGroup -Name "Linux3" -Location "EastUS"

ResourceGroupName : Linux3
Location           : eastus
ProvisioningState  : Succeeded
Tags               :
ResourceId         : /subscriptions/2e716f97-7b1f-4ce5-8057-cc018026db55/resourceGroups/Linux3
```

2. Crear los recursos de nuestra red virtual:

- Red virtual.
- Configuración de subred.
- Dirección ip pública.

```
PS /home/johan> $vnet = New-AzVirtualNetwork `
>> -ResourceGroupName "Linux3" `
>> -Location "EastUS" `
>> -Name "myVNET" `
>> -AddressPrefix 192.168.0.0/16 `
>> -Subnet $subnetConfig
PS /home/johan> $pip = New-AzPublicIpAddress `
>> -ResourceGroupName "Linux3" `
>> -Location "EastUS" `
>> -AllocationMethod Static `
>> -IdleTimeoutInMinutes 4 `
>> -Name "mypublicdns$(Get-Random)"
WARNING: Breaking changes in the cmdlet 'New-AzPublicIpAddress' :
WARNING: - Default behaviour of Zone will be changed

WARNING: Cmdlet invocation changes :
      Old Way : Zone = [] means the Standard Public IP address is zone-redundant
      New Way : Zone = [] means the Standard Public IP has no zones. If you want to create a zone-redundant Pu
blic IP address, please specify all the zones in the region. For example, Zone = ["1", "2", "3"].To learn more v
isit aka.ms/standardpublicip
WARNING: NOTE : Go to https://aka.ms/azps-changewarnings for steps to suppress this breaking change warning, and
other information on breaking changes in Azure PowerShell.
```

3. Añadir dos reglas de entrada para permitir el tráfico por el puerto 22 y el 80, indicando una wildcard en este caso "\*" la cual representa que se puede recibir tráfico desde cualquier dirección ip. Luego crear un grupo de seguridad de la red al cual asignaremos las reglas que creamos.

```
PS /home/johan> $nsgRuleWeb = New-AzNetworkSecurityRuleConfig `
>> -Name "myNetworkSecurityGroupRuleWWW" `
>> -Protocol "Tcp" `
>> -Direction "Inbound" `
>> -Priority 1001 `
>> -SourceAddressPrefix * `
>> -SourcePortRange * `
>> -DestinationAddressPrefix * `
>> -DestinationPortRange 80 `
>> -Access "Allow"
PS /home/johan> $nsgRuleSSH = New-AzNetworkSecurityRuleConfig `
>> -Name "myNetworkSecurityGroupRuleSSH" `
>> -Protocol "Tcp" `
>> -Direction "Inbound" `
>> -Priority 1000 `
>> -SourceAddressPrefix * `
>> -SourcePortRange * `
>> -DestinationAddressPrefix * `
>> -DestinationPortRange 22 `
>> -Access "Allow"
PS /home/johan> 
```

```
PS /home/johan> $nsg = New-AzNetworkSecurityGroup `
>> -ResourceGroupName "Linux3" `
>> -Location "EastUS" `
>> -Name "myNetworkSecurityGroup" `
>> -SecurityRules $nsgRuleSSH,$nsgRuleWeb
PS /home/johan> 
```

4. Crear una tarjeta de red virtual

```
PS /home/johan> $nic = New-AzNetworkInterface `
>> -Name "myNic" `
>> -ResourceGroupName "Linux3" `
>> -Location "EastUS" `
>> -SubnetId $vnet.Subnets[0].Id `
>> -PublicIpAddressId $pip.Id `
>> -NetworkSecurityGroupId $nsg.Id
PS /home/johan> 
```

5. Luego creamos nuestra máquina virtual.

- 5.1 Primero debemos definir un objeto en Powershell que va a contener nuestro usuario administrador y la contraseña.
- 5.2 Configurar el nombre de la Vm y su tamaño.
- 5.3 Definir la imagen del SO.
- 5.4 Asignar la tarjeta de red virtual (NIC)
- 5.5 Agregamos nuestra llave pública a los "authorized keys" para poder autenticarnos a la máquina que estamos creando.

```

PS /home/johan> $securePassword = ConvertTo-SecureString ' ' -AsPlainText -Force
PS /home/johan> $cred = New-Object System.Management.Automation.PSCredential ("azureuser", $securePassword)
PS /home/johan> $vmConfig = New-AzVMConfig `
>> -VMName "Linux3Vm" `
>> -VMSize "Standard_D1" | `
>> Set-AzVMOperatingSystem `
>> -Linux `
>> -ComputerName "Linux3Vm" `
>> -Credential $cred `
>> -DisablePasswordAuthentication | `
>> Set-AzVMSourceImage `
>> -PublisherName "Canonical" `
>> -Offer "UbuntuServer" `
>> -Skus "18.04-LTS" `
>> -Version "latest" | `
>> Add-AzVMNetworkInterface `
>> -Id $nic.Id
PS /home/johan> $sshPublicKey = cat ~/.ssh/id_rsa.pub
PS /home/johan> Add-AzVMSshPublicKey `
>> -VM $vmconfig `
>> -KeyData $sshPublicKey `
>> -Path "/home/johan/.ssh/authorized_keys"

Name           : Linux3Vm
HardwareProfile : {VmSize}
NetworkProfile  : {NetworkInterfaces}
OSProfile       : {ComputerName, AdminUsername, AdminPassword, LinuxConfiguration}
StorageProfile  : {ImageReference}

PS /home/johan>

```

## 6. Nos conectamos por ssh con la llave privada nuestra e instalamos Nginx.

```

azureuser@myVM:~$ systemctl status nginx && whoami
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2020-08-11 04:52:54 UTC; 1min 33s ago
     Docs: man:nginx(8)
  Main PID: 2443 (nginx)
    Tasks: 2 (limit: 4043)
   CGroup: /system.slice/nginx.service
           └─2443 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─2446 nginx: worker process

Aug 11 04:52:53 myVM systemd[1]: Starting A high performance web server and a reverse proxy server...
Aug 11 04:52:54 myVM systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid: Invalid argument
Aug 11 04:52:54 myVM systemd[1]: Started A high performance web server and a reverse proxy server.
azureuser
azureuser@myVM:~$

```



# Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](https://nginx.org).  
Commercial support is available at [nginx.com](https://nginx.com).

*Thank you for using nginx.*

## d. ARM (Azure resource manager) template method

1. Ingresar al siguiente url <https://portal.azure.com/#create/Microsoft.Template> y diligenciar los datos necesarios. Una vez modificado y verificado el template creamos la máquina de manera simple. Es importante agregar nuestra llave publica en el método de autenticación para luego entrar con nuestra llave privada

### Deploy a simple Ubuntu Linux VM 18.04-LTS.

Azure quickstart template

#### Deployment scope

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	<input type="text" value="Azure subscription 1"/>
Resource group *	<input type="text" value="(New) LinuxLast"/> <a href="#">Create new</a>


#### Parameters

Region *	<input type="text" value="East US"/>
Vm Name	<input type="text" value="simpleLinuxVM"/>
Admin Username *	<input type="text" value="johanArias"/>
Authentication Type	<input type="text" value="sshPublicKey"/>
Admin Password Or Key *	<input type="password" value="*****"/>
Dns Label Prefix	<input type="text" value="[toLower(concat('simplelinuxvm-', uniqueString(resourceGroup().id)))]"/>
Ubuntu OS Version	<input type="text" value="18.04-LTS"/>
Location	<input type="text" value="[resourceGroup().location]"/>
Vm Size	<input type="text" value="Standard_B51_v2"/>
Virtual Network Name	<input type="text" value="vNet"/>
Subnet Name	<input type="text" value="Subnet"/>
Network Security Group Name	<input type="text" value="SecGroupNet"/>





[Review + create](#)[< Previous](#)[Next : Review + create >](#)

```
1 {
2   "$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",
3   "contentVersion": "1.0.0.0",
4   "parameters": {
5     "vmName": {
6       "value": "simpleLinuxVM"
7     },
8     "adminUsername": {
9       "value": "johanArias"
10    },
11    "authenticationType": {
12      "value": "sshPublicKey"
13    },
14    "adminPasswordOrKey": {
15      "value": "REDACTED"
16    },
17    "dnsLabelPrefix": {
18      "value": "[toLowerCase(concat('simplelinuxvm-', uniqueString(resourceGroup().id)))]"
19    },
20    "ubuntuOSVersion": {
21      "value": "18.04-LTS"
22    },
23    "location": {
24      "value": "[resourceGroup().location]"
25    },
26    "vmSize": {
27      "value": "Standard_B1_v2"
28    },
29    "virtualNetworkName": {
```

2. Verificamos la creación de los recursos y asignamos una regla de entrada al grupo de seguridad de la red para permitir el tráfico del puerto 80

 **Port\_HTTP** ×

SecGroupNet

 Save  Discard  Basic  Delete

Source \* ⓘ  

Any

Source port ranges \* ⓘ  

\*

Destination \* ⓘ  

Any

Destination port ranges \* ⓘ  

80

Protocol \*  

Any TCP UDP ICMP

Action \*  

Allow Deny

Priority \* ⓘ  

1010

Name  

Port\_HTTP

Description



+ Add   Edit columns   Delete resource group   Refresh   → Move   Export to CSV   Open query   Assign tags   Delete   Export template   Feedback

Subscription (change) : Azure subscription 1   Deployments : 1 Succeeded  
 Subscription ID : 2e716f97-7b1f-4ce5-8057-cc018026db55  
 Tags (change) : Click here to add tags

Filter by name...   Type == all   Location == all   Add filter

Showing 1 to 6 of 6 records.   Show hidden types   No grouping

Name ↑↓	Type ↑↓	Location ↑↓
SecGroupNet	Network security group	East US
simpleLinuxVM	Virtual machine	East US
simpleLinuxVM_OsDisk_1_02f8090bead343f485e573a44cf05b06	Disk	East US
simpleLinuxVMNetint	Network interface	East US
simpleLinuxVMPublicIP	Public IP address	East US
vNet	Virtual network	East US

3. Habilitamos el puerto 80 desde la máquina mediante las iptables(firewall) e instalamos Nginx para correr un servidor web.

▲ 137.117.32.96

**Welcome to nginx!**

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

```

chan@simpleLinuxVM: ~
Aug 11 05:59:35 simpleLinuxVM systemd[1]: Started A high performance web server and a reverse proxy server.
chan@simpleLinuxVM:~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
chan@simpleLinuxVM:~$
  
```

## Definiciones

- i. **Resource Group** : es un conjunto de recursos que comparten el mismo ciclo de vida, permisos y políticas
- ii. **Network Security Group:**  
Un grupo de seguridad de red es una capa de seguridad que actúa como un firewall virtual para controlar el tráfico de entrada y salida de máquinas virtuales (a través de interfaces de red) y subredes. Contiene un conjunto de reglas de seguridad que permiten o niegan el tráfico entrante y saliente utilizando las siguientes 5 tuplas: protocolo, rango de dirección IP de origen, rango de puerto de origen, rango de dirección IP de destino y rango de puerto de destino.
- iii. **Virtual Net:** Azure Virtual Network (VNet) es el bloque de creación fundamental de una red privada en Azure. VNet permite muchos tipos de recursos de Azure,

como Azure Virtual Machines (máquinas virtuales), para comunicarse de forma segura entre usuarios, con Internet y con las redes locales.

- iv. **VM Sizes:** Es el tamaño para soportar la carga de trabajo de la VM. El tamaño que se elija determinará factores como la potencia de procesamiento, la memoria y la capacidad de almacenamiento. Azure ofrece una amplia variedad de tamaños para soportar muchos tipos de usos. Azure cobra un precio por hora basado en el tamaño y el sistema operativo de la máquina virtual.
- v. **VM Power State:** Indica el estado en el que se encuentra la máquina virtual. Existen 7 estados en los que puede encontrarse una máquina:
  - **Starting:** Indica que la VM se está iniciando.
  - **Running:** Indica que la VM se está corriendo.
  - **Stopping:** Indica que la VM se está deteniendo.
  - **Stopped:** Indica que la VM se está detenida. Sin embargo, todavía sigue generando cargas de computación.
  - **Deallocating:** Indica que la VM se está repartiendo.
  - **Deallocated:** Indica que la VM se ha retirado completamente del hipervisor pero aún está disponible en el avión de control. Las VM en el estado de "deallocated" no incurren en cargos de computación.
  - **.Unknown (-):** Indica que el estado de la VM es desconocido.
- vi. **ARM Template:** Las plantillas de Azure Resource Manager son archivos de JavaScript Object Notation (JSON) que definen la infraestructura y la configuración de su proyecto. Estas plantillas sirven para agilizar el proceso de creación de un proyecto.