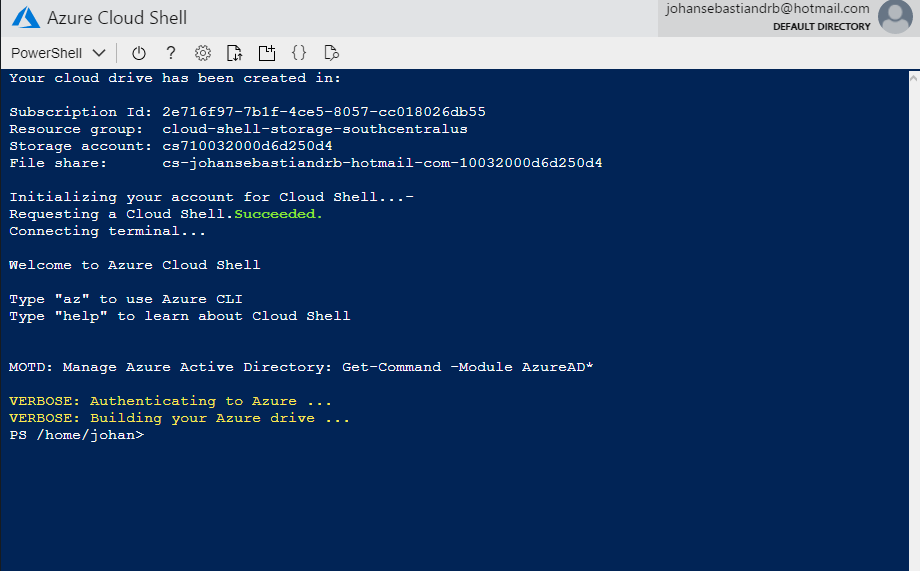
**Laboratorio 1 – Paradigmas emergentes en la Computación en la nube**

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:**

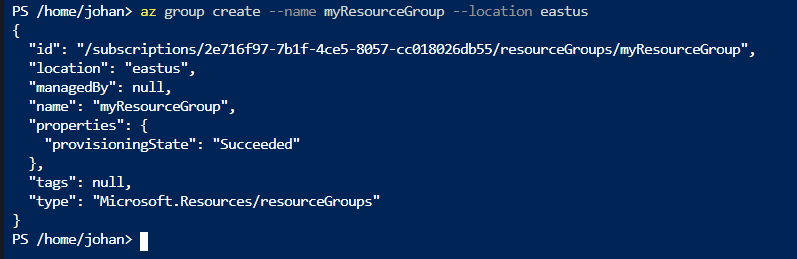
1. **Azure Command Line Interface Method**
2. Crear el grupo de recursos donde se alojarán los componentes de nuestro despliegue eligiendo una localización optima de acuerdo con la mejor latencia obtenida:

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



**Comando para crear el grupo de recursos:**

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



1. 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:**

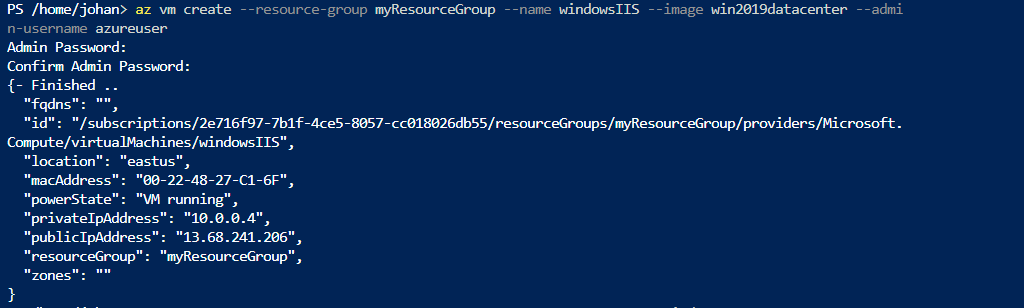
az vm create \

--resource-group myResourceGroup \

--name myVM \

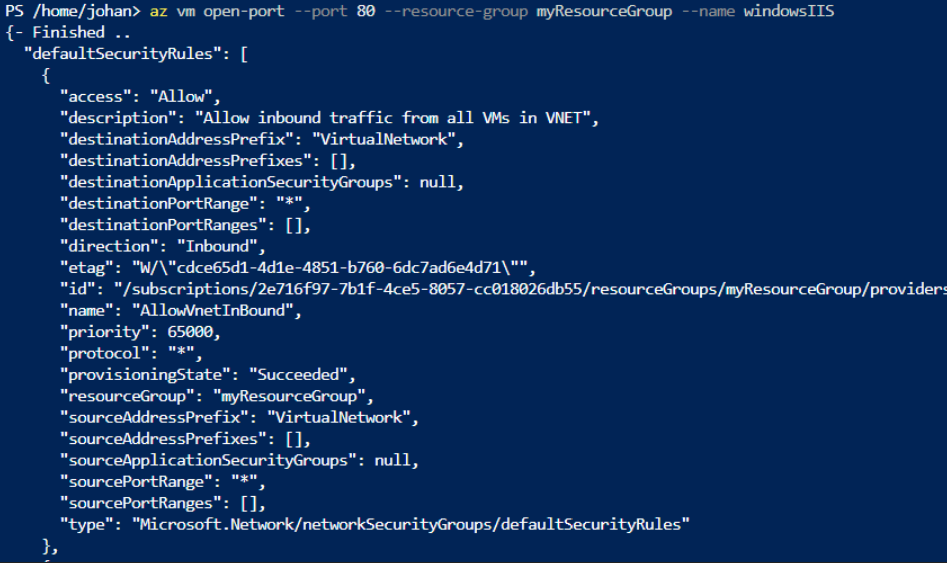
--image win2016datacenter \

--admin-username azureuser



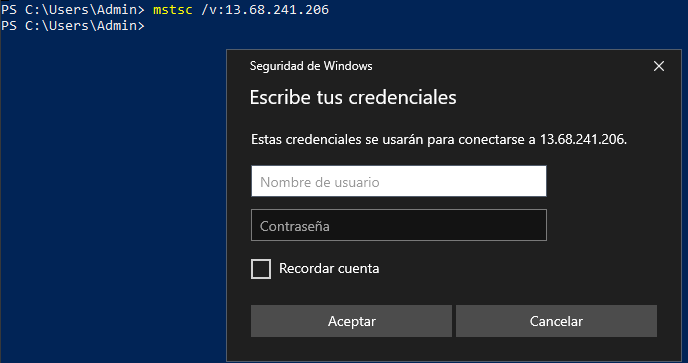
1. Lo siguiente que debemos hacer es habilitar el tráfico por el puerto 80 para que se pueda consultar nuestro sitio web que crearemos más adelante. Esto es necesario ya que por defecto solo se permite tráfico proveniente del puerto 3380 (RDP).

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

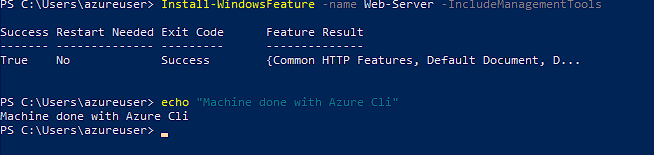


1. Una vez hayamos instalado todo podremos conectarnos desde nuestra máquina local con el siguiente comando:

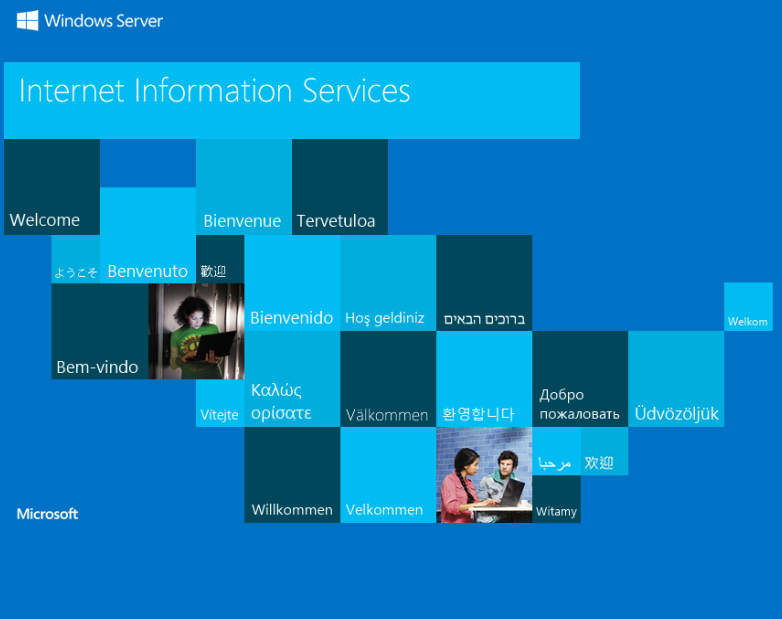
mstsc /v: publicIpAddress



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

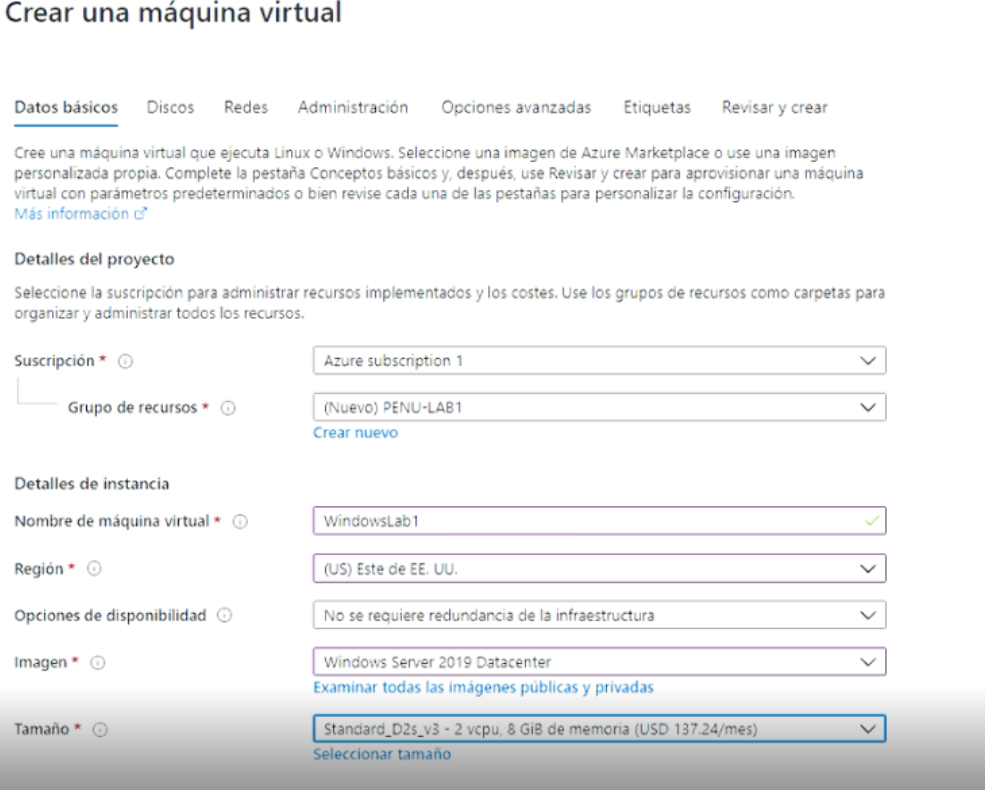


Verificamos que esté corriendo:

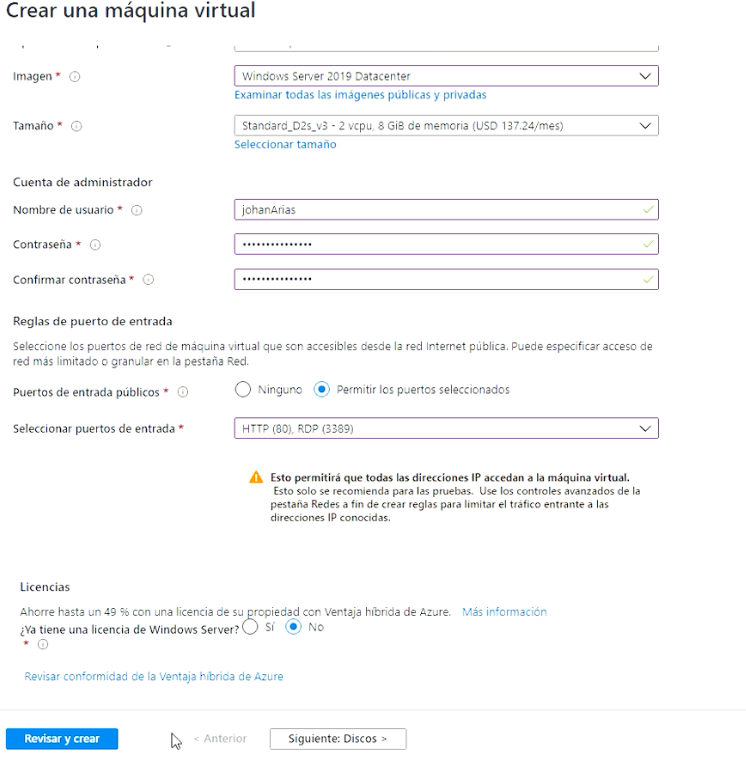


1. Azure portal method
2. En primer lugar, debemos crear nuestra máquina virtual, podemos decidir si asociarla con un grupo de recursos ya creado o podemos crear un grupo de recursos nuevo. Es necesario diligenciar el formulario el cual contiene información como:

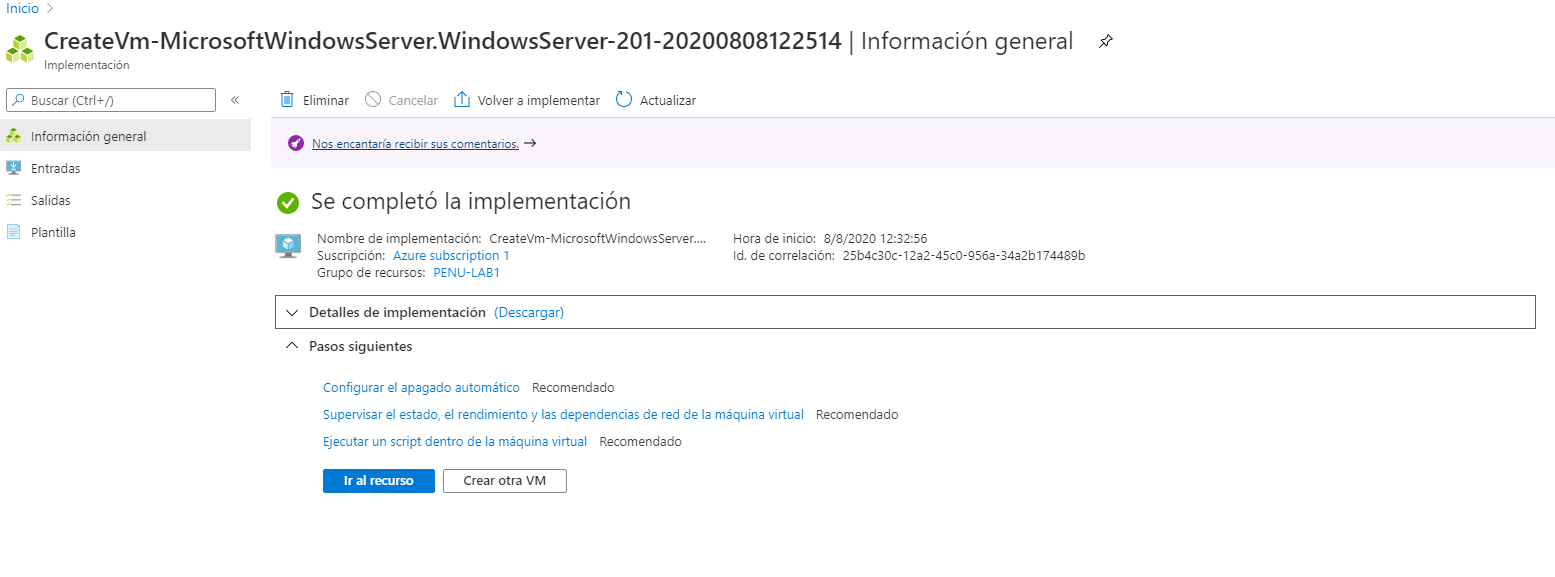
* Nombre de la máquina virtual.
* Localización de nuestra máquina.
* Tipo de disponibilidad (en caso de que necesitemos que la máquina nunca esté fuera de servicio).
* Imagen del sistema operativo.
* Tamaño, el cual determina la capacidad del procesador, de memoria y de almacenamiento.



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

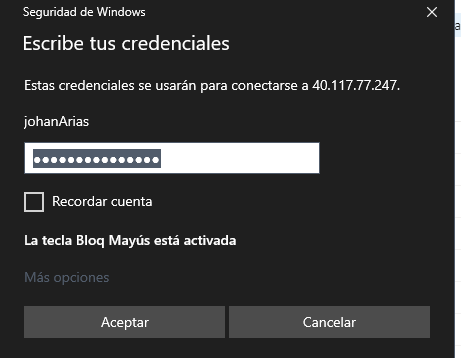


1. Una vez creados el grupo de recursos y sus elementos veremos un panel de confirmación

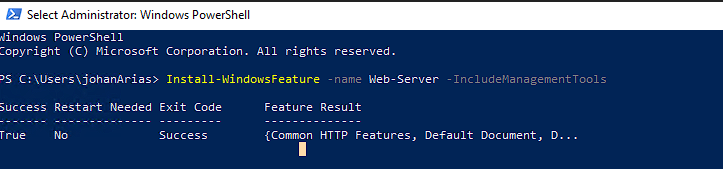


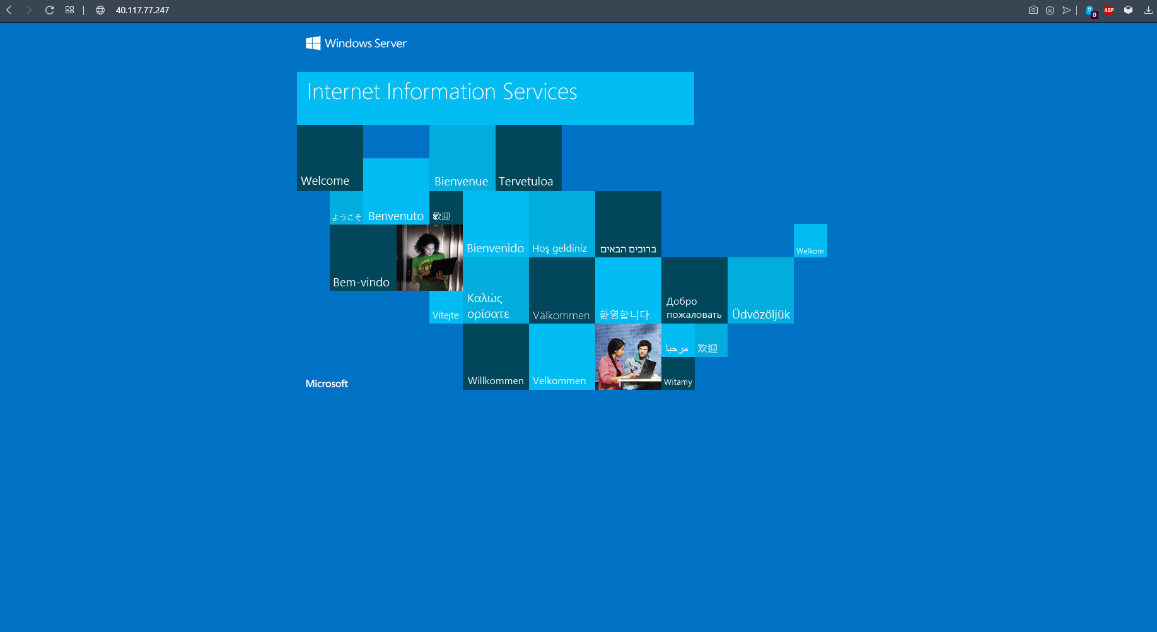
1. Luego debemos conectarnos mediante el protocolo RDP (Remote Desktop Protocol) descargando un archivo .rdp e introduciendo las credenciales del usuario que creamos anteriormente.





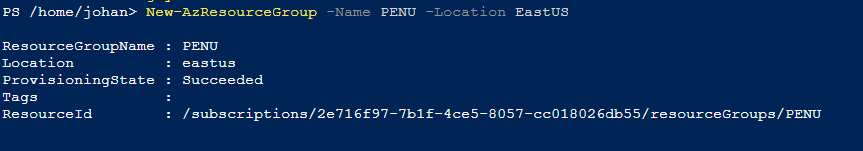
1. Instalamos IIS desde Powershell y verificamos que corra desde una máquina remota.



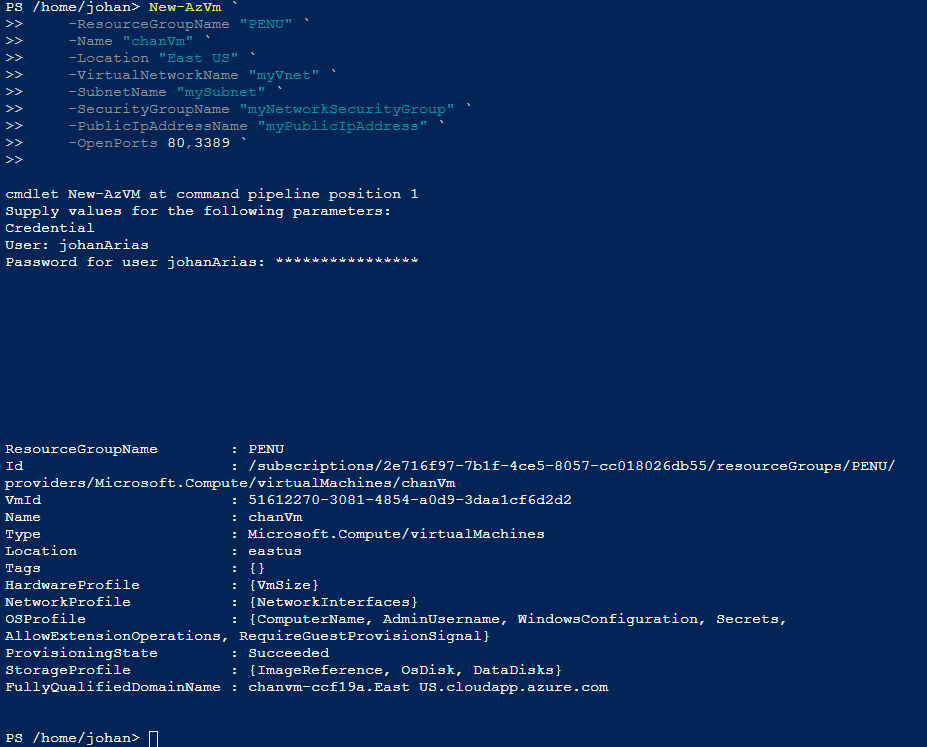


1. Powershell method
2. Para crear un grupo de recursos mediante PowerShell debemos ejecutar el siguiente comando especificando el nombre del grupo de recursos y su localización.

New-AzResourceGroup -Name myResourceGroup -Location EastUS



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

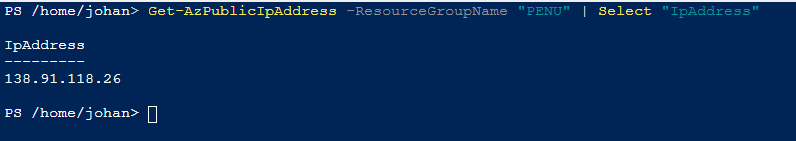
* Red virtual.
* Subnet.
* Grupo de seguridad.
* Dirección IP pública.
* Puertos abiertos.

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

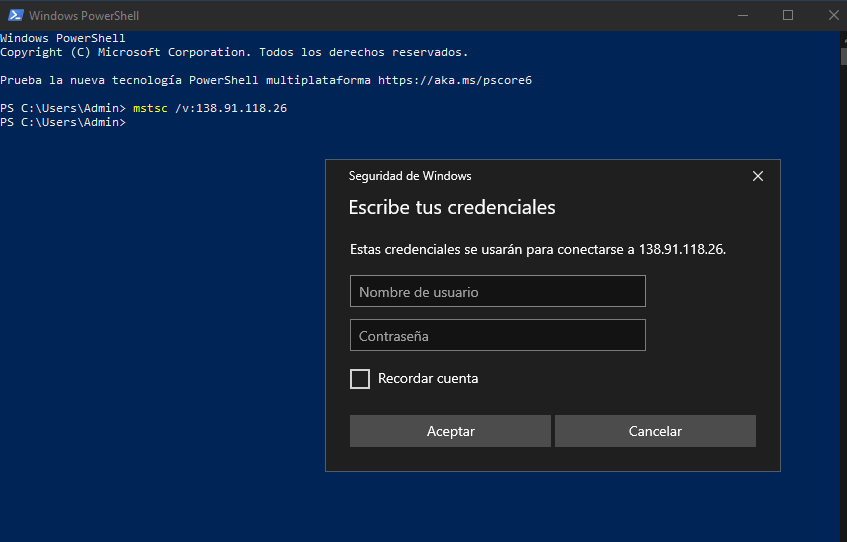
1. Luego de tener todos los recursos creados podremos conectarnos a la máquina.

Para consultar la ip pública de nuestra máquina podemos ejecutar el siguiente comando en Powershell:

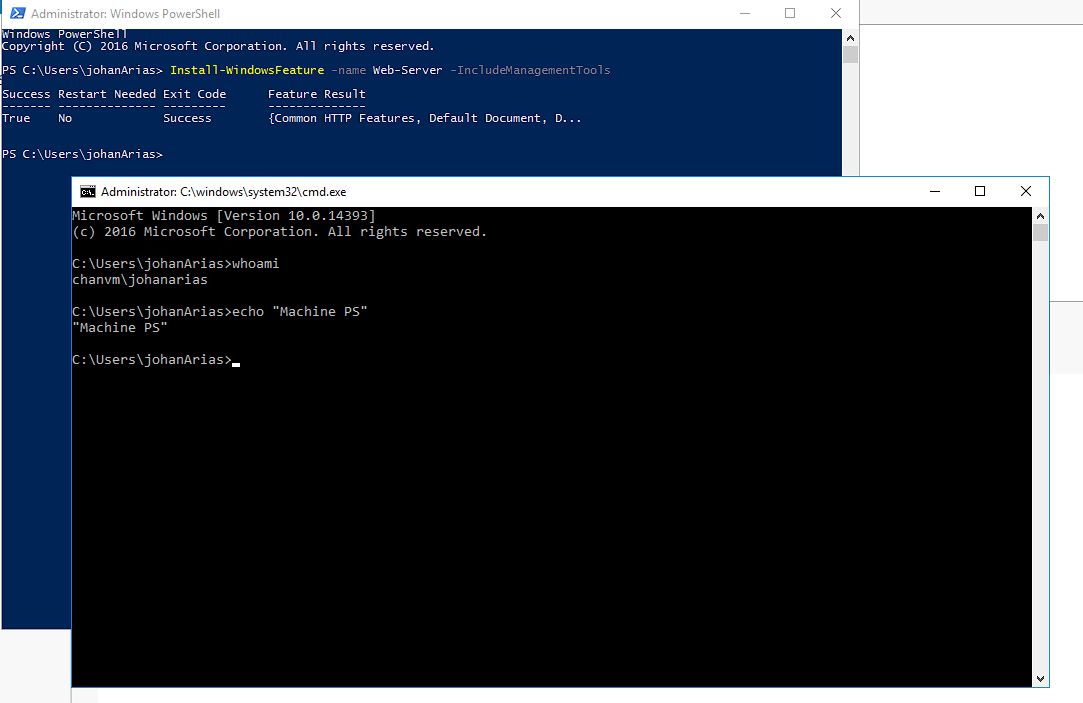
Get-AzPublicIpAddress -ResourceGroupName "myResourceGroup" | Select "IpAddress"

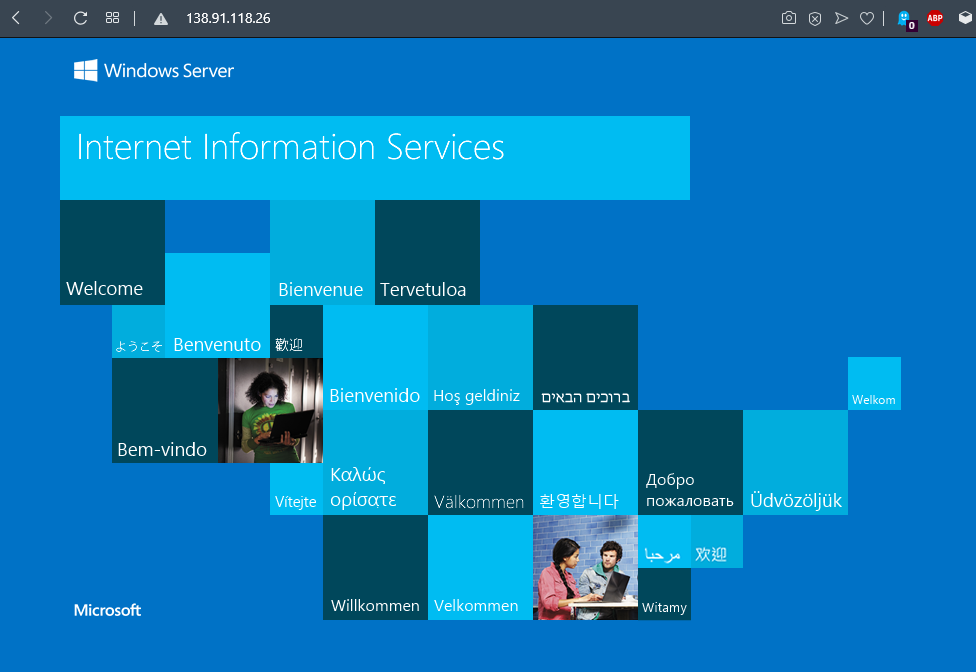


Luego usamos mstsc para crear una conexión a una sesión remota en este caso a nuestra máquina virtual.

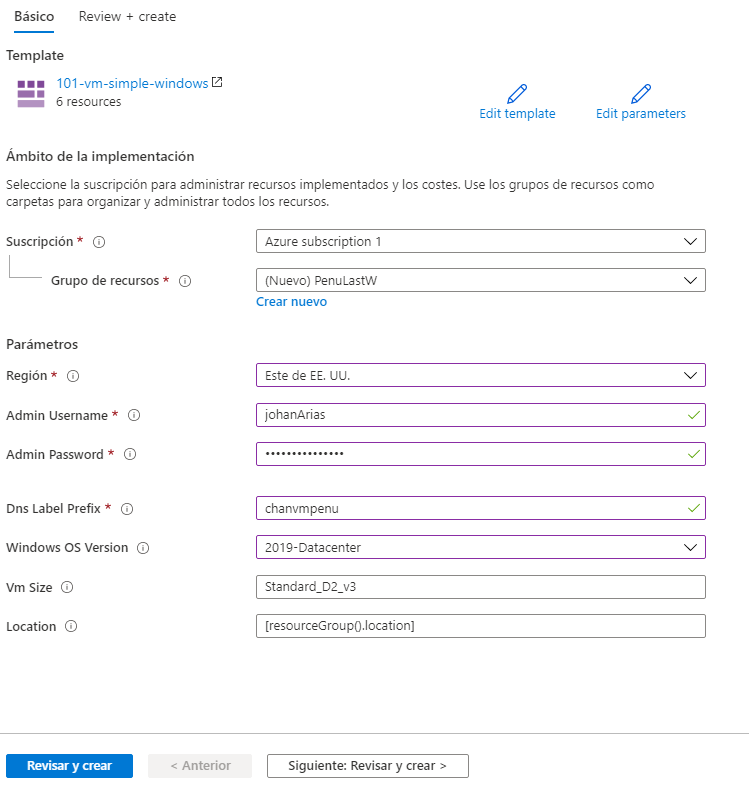


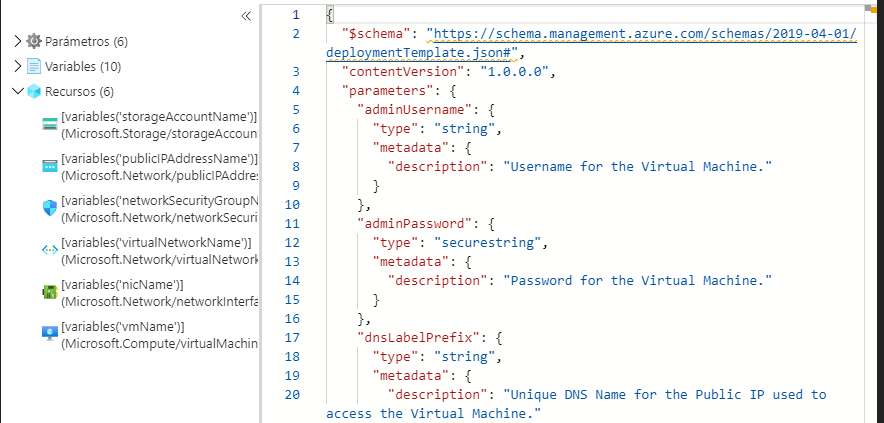
1. Finalmente instalamos IIS mediante Powershell y verificamos que sea accesible desde un lugar remoto.

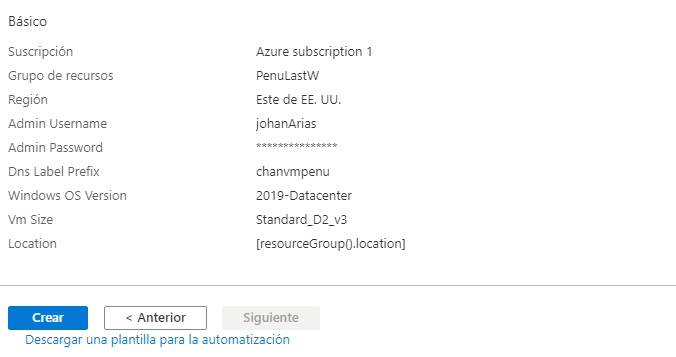




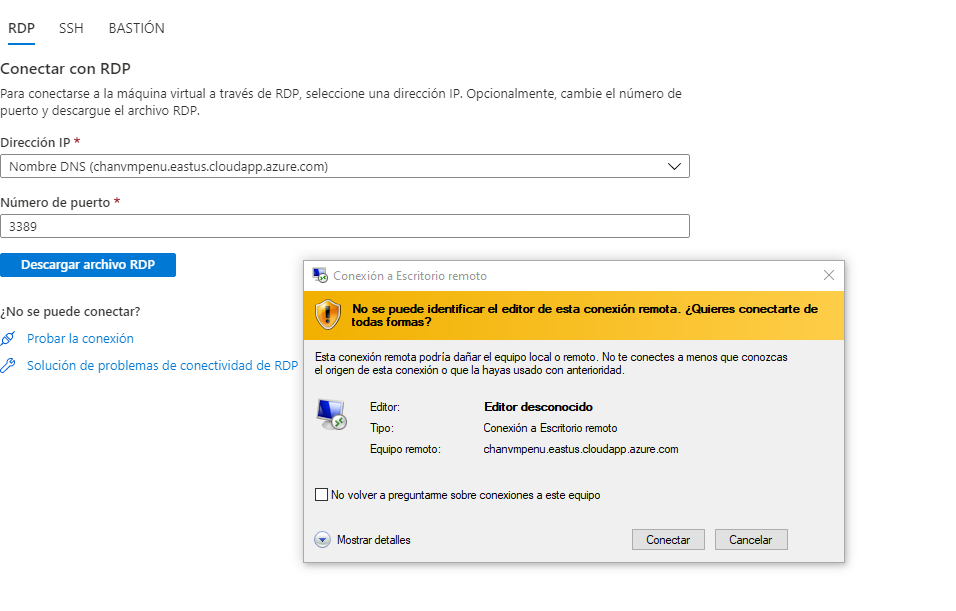
1. **ARM (Azure resource manager) template method**
2. Primero debemos acceder al siguiente enlace <https://portal.azure.com/#create/Microsoft.Template> en el cual va a estar alojada una plantilla en formato JSON, lo único que debemos hacer es llenar los datos necesarios y verificar la plantilla. Desde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

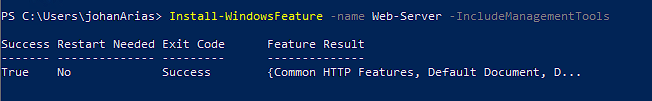




Verificamos los valores y creamos nuestro grupo de recursos.

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

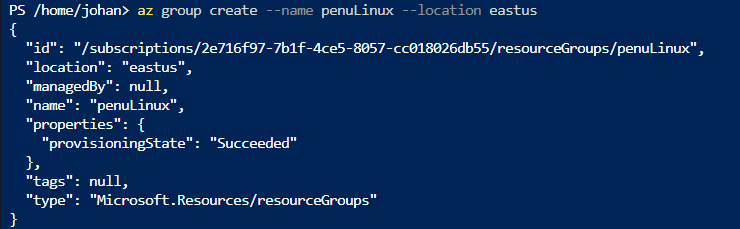




**LINUX**

1. **Azure command line interface method**
2. **En primer lugar debemos crear un grupo de recursos en el cual indicamos su localización**

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



1. **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 \**

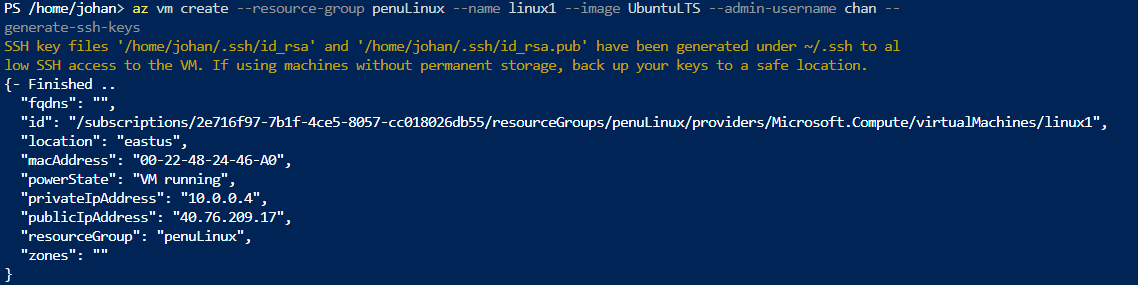
**--resource-group myResourceGroup \**

**--name myVM \**

**--image UbuntuLTS \**

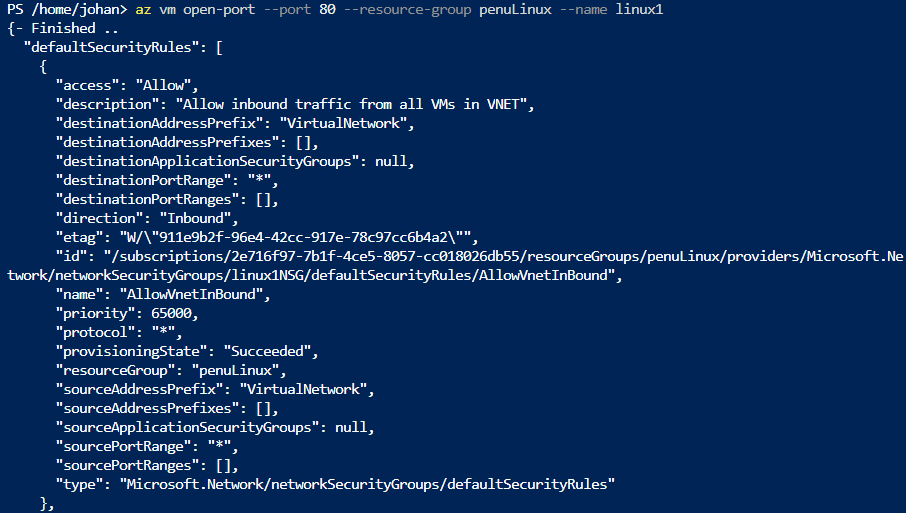
**--admin-username azureuser \**

**--generate-ssh-keys**

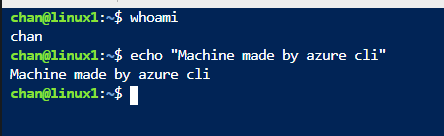


1. Ahora debemos habilitar el trafico por el puerto 80 para poder recibir peticiones a nuestro servidor web.

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



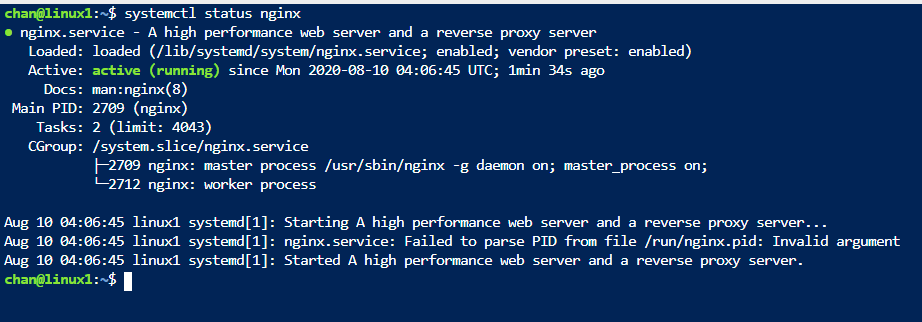
1. Lo siguiente que nos queda hacer es conectarnos a la máquina mediante ssh indicando el usuario administrador que creamos anteriormente

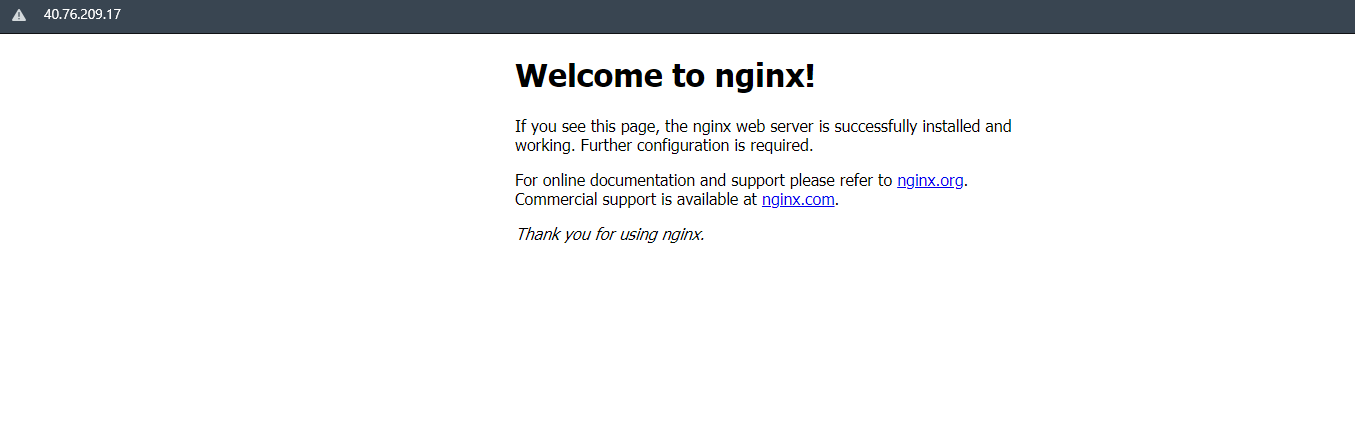


1. Por último instalamos Nginx mediante el administrador de paquetes de Linux “apt” y verificamos que nuestro servidor web se pueda consultar de forma remota.

Comandos : sudo apt-get -y update

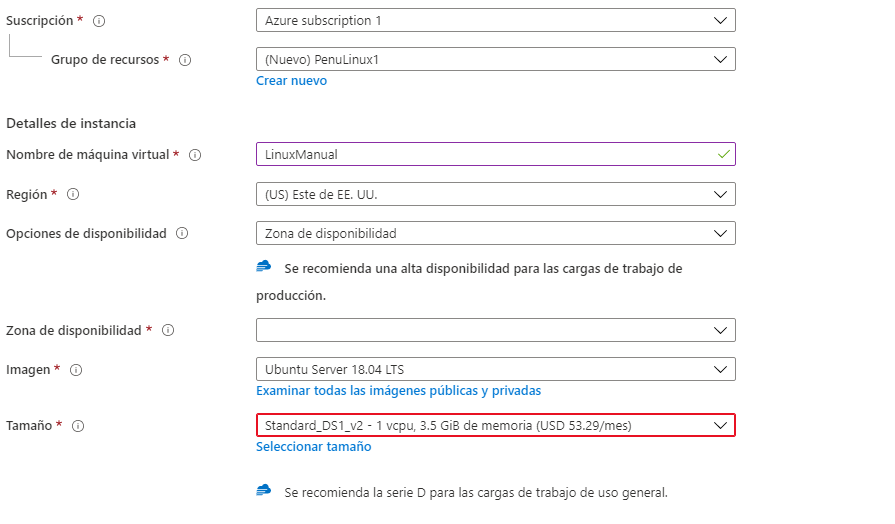
sudo apt-get -y install nginx



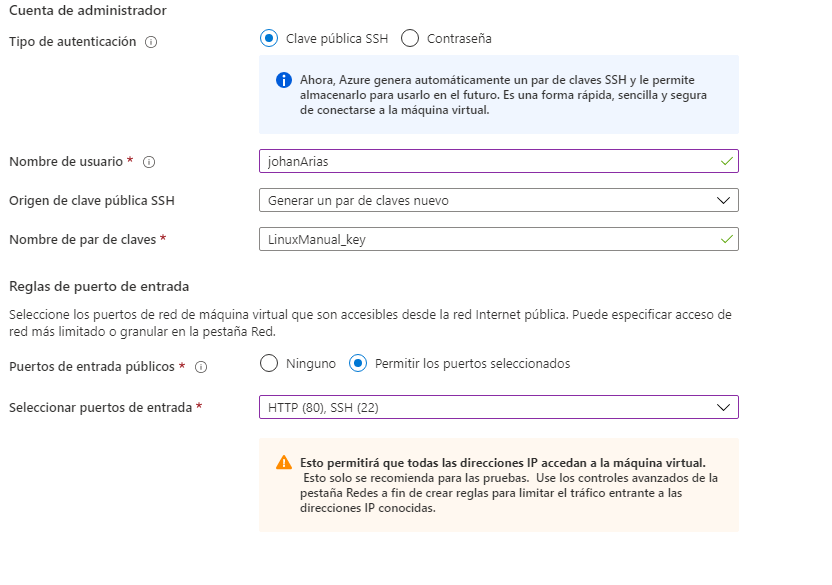


1. Azure portal method
2. Crear un grupo de recursos, indicar información como:

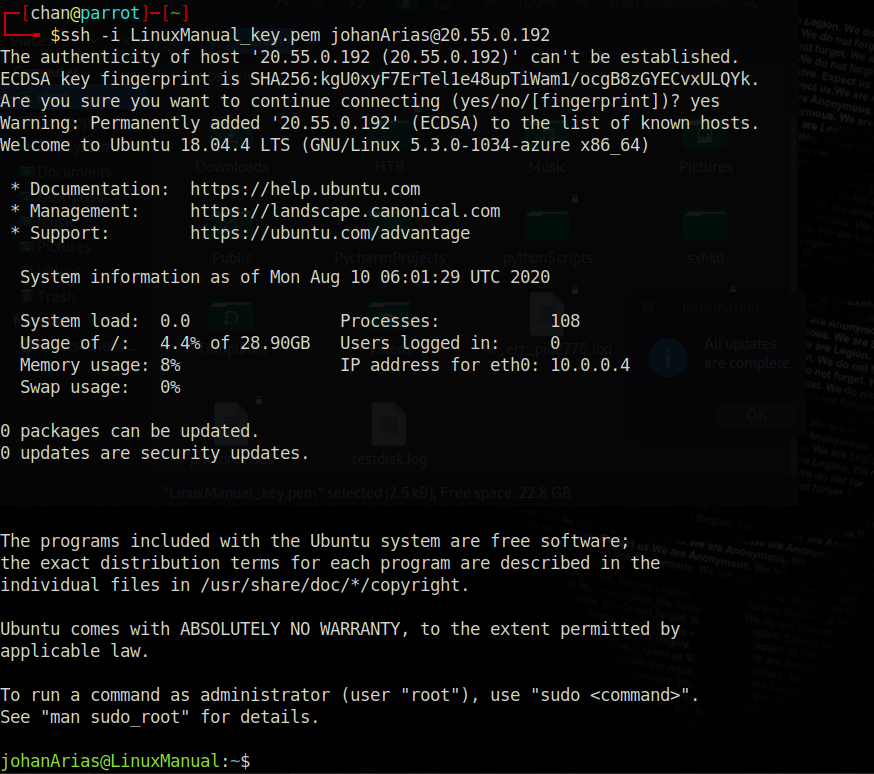
* Nombre de la máquina virtual.
* Región de localización de nuestro grupo de recursos.
* Opciones de disponibilidad en caso de que sea un sistema que vaya a soportar mucho tráfico.
* Imagen del sistema operativo.
* Tamaño que representa la capacidad de procesamiento, memoria y almacenamiento.



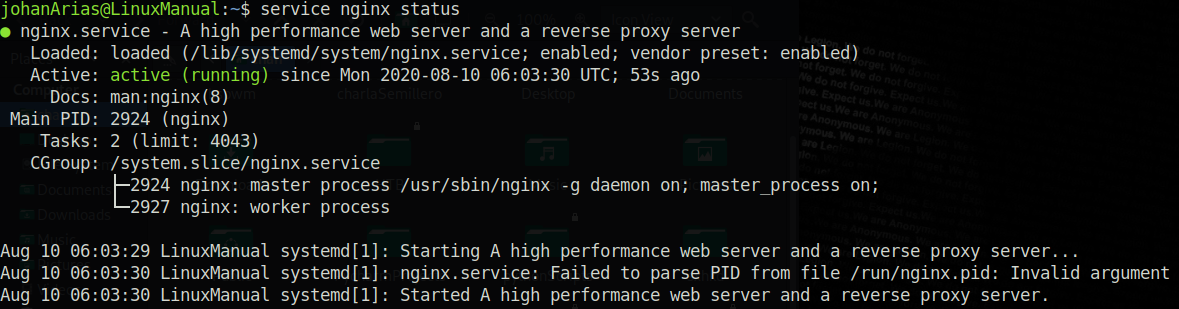
1. Debemos seleccionar un método de autenticación, para este caso lo haremos mediante ssh por lo que debemos indicar un nombre de usuario administrador, además tendremos que generar un par de llaves para realizar la conexión mediante este protocolo. De igual forma, es necesario habilitar el puerto 22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

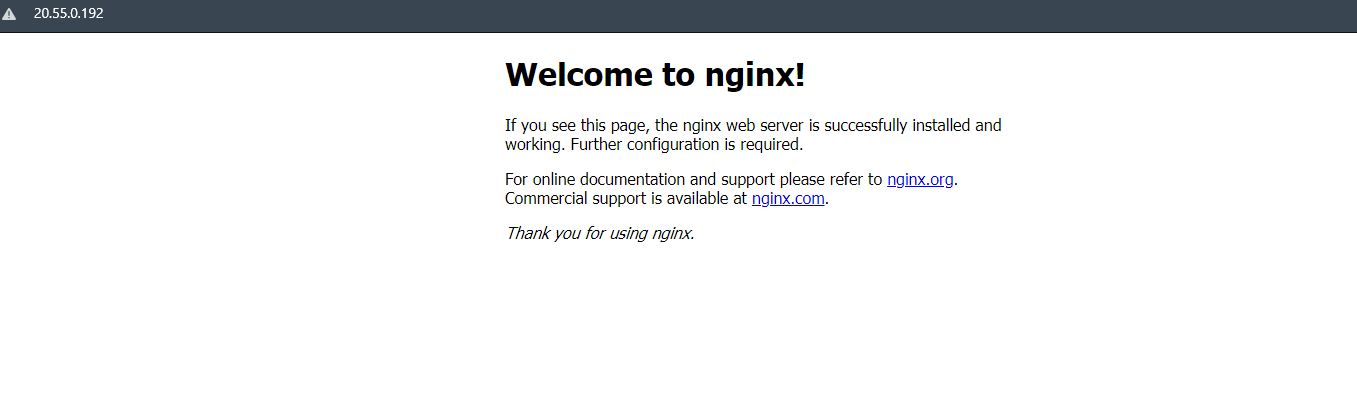


1. Descargamos nuestra llave privada generada de formato .pem le damos permisos únicamente de lectura y de escritura (chmod 400 o 600 en Linux) y luego nos autenticamos mediante ssh.



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

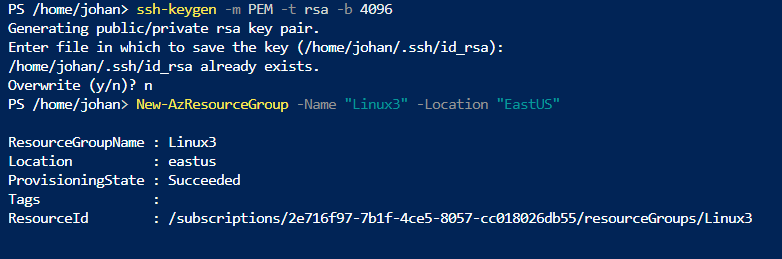




1. Powershell method
2. Generar un par de llaves ssh y crear un grupo de recursos indicando su localización:

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

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

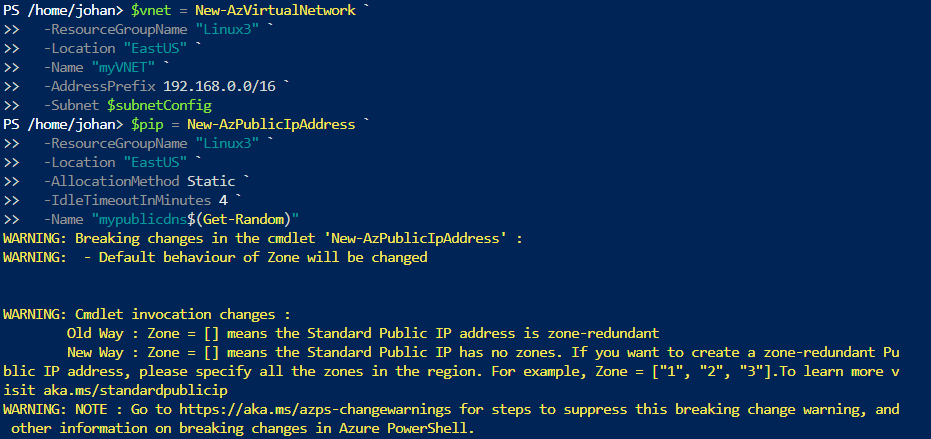


1. Crear los recursos de nuestra red virtual:

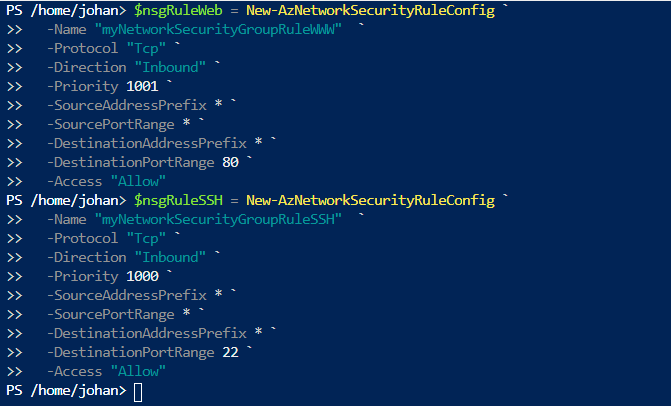
- Red virtual.

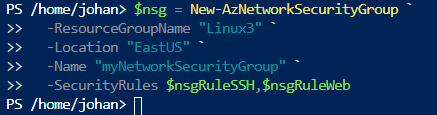
- Configuración de subred.

- Dirección ip pública.

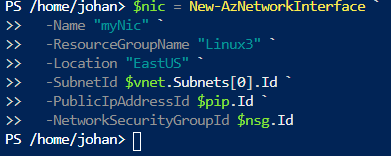


1. Añadir dos reglas de entrada para permitir el tráfico por el puerto 22 y el 80, indicando una wildcard en este caso “ \* ” la cual representa que se puede recibir tráfico desde cualquier dirección ip. Luego crear un grupo de seguridad de la red al cual asignaremos las reglas que creamos.

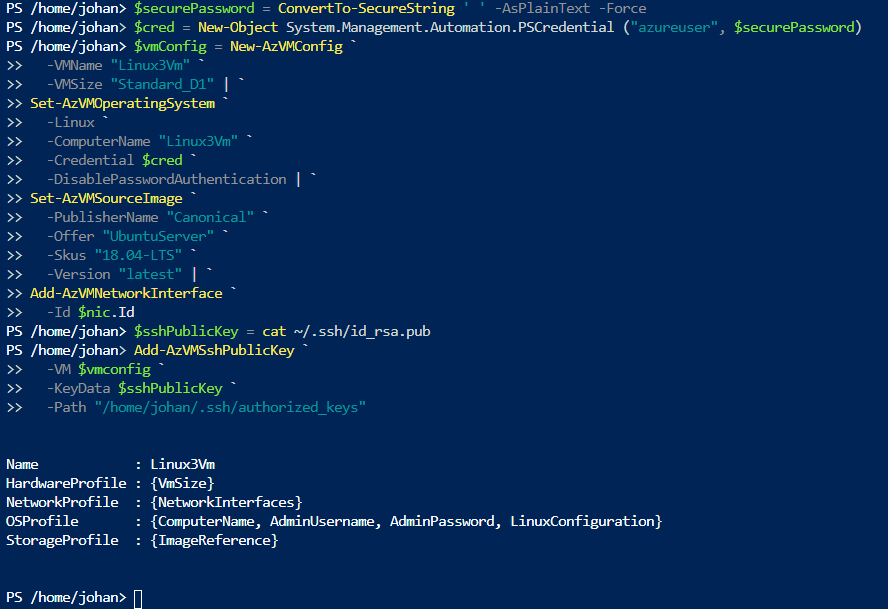




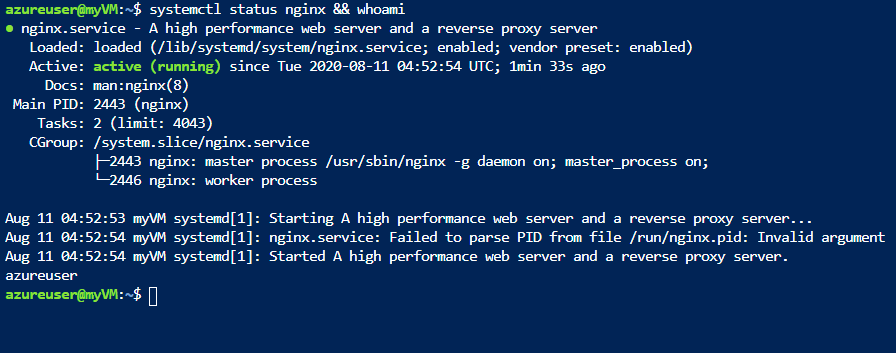
1. Crear una tarjeta de red virtual

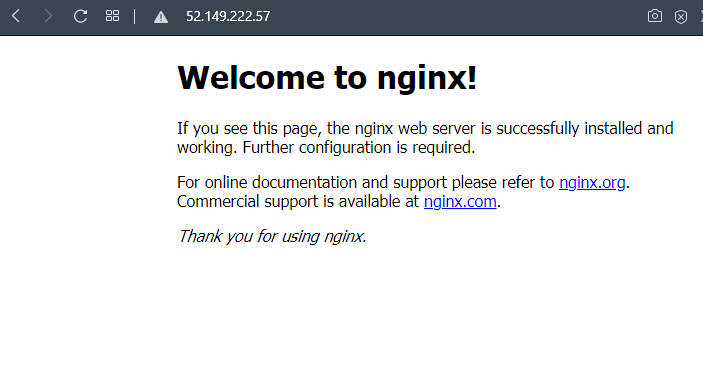


1. Luego creamos nuestra máquina virtual.
   1. Primero debemos definir un objeto en Powershell que va a contener nuestro usuario administrador y la contraseña.
   2. Configurar el nombre de la Vm y su tamaño.
   3. Definir la imagen del SO.
   4. Asignar la tarjeta de red virtual (NIC)
   5. Agregamos nuestra llave pública a los “authorized keys” para poder autenticarnos a la máquina que estamos creando.

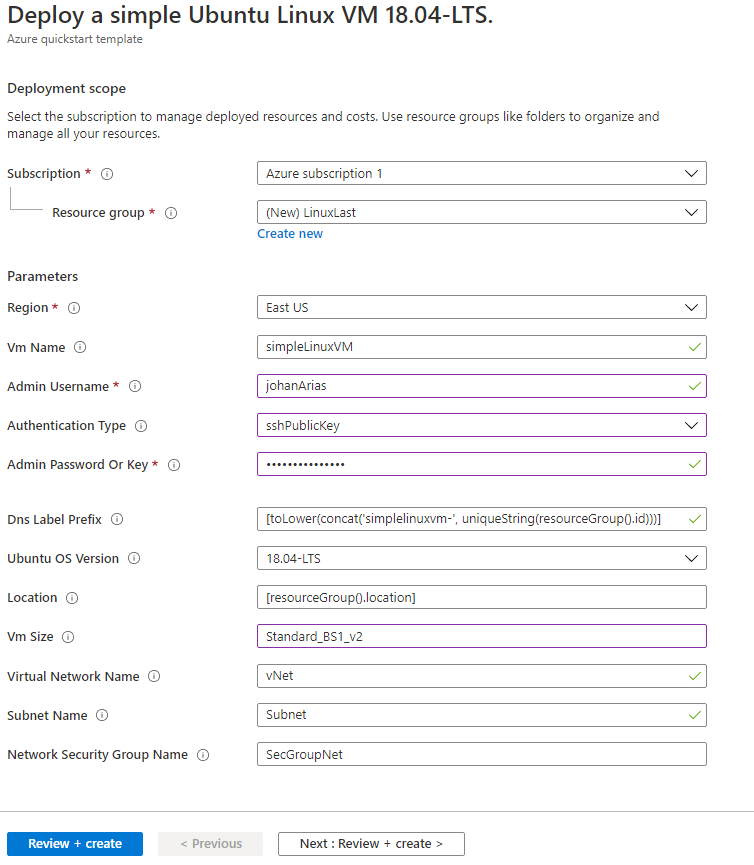


1. Nos conectamos por ssh con la llave privada nuestra e instalamos **N**ginx.



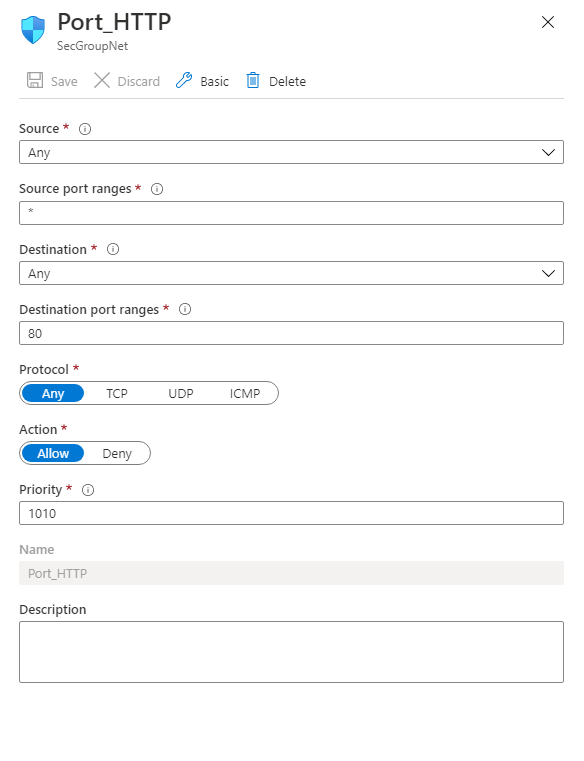


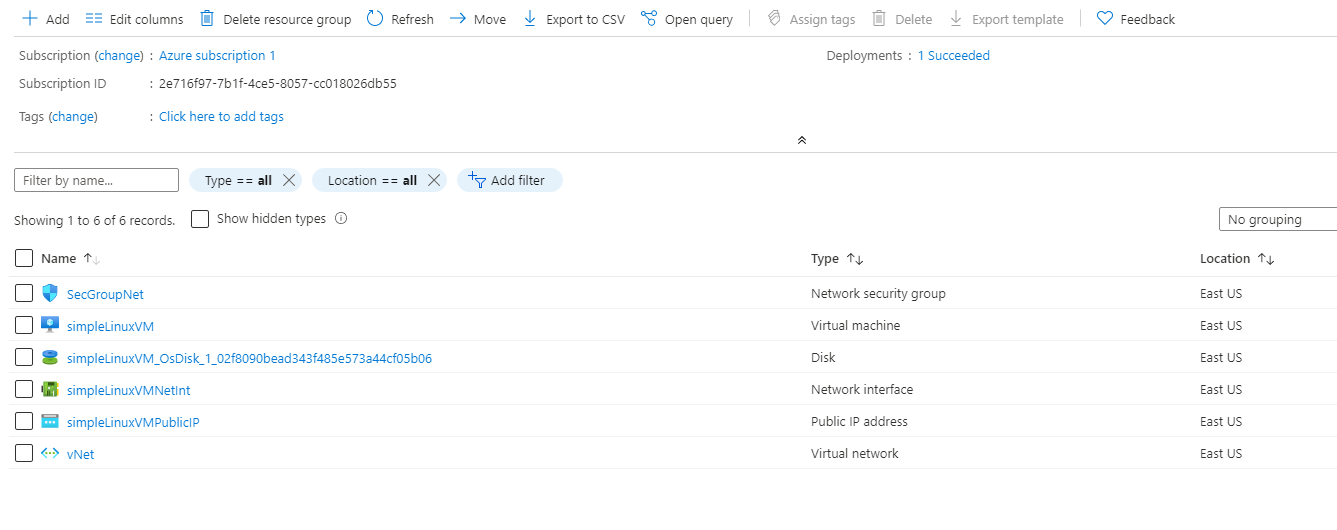
1. **ARM (Azure resource manager) template method**
2. Ingresar al siguiente url <https://portal.azure.com/#create/Microsoft.Template> y diligenciar los datos necesarios. Una vez modificado y verificado el template creamos la máquina de manera simple. Es importante agregar nuestra llave publica en el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



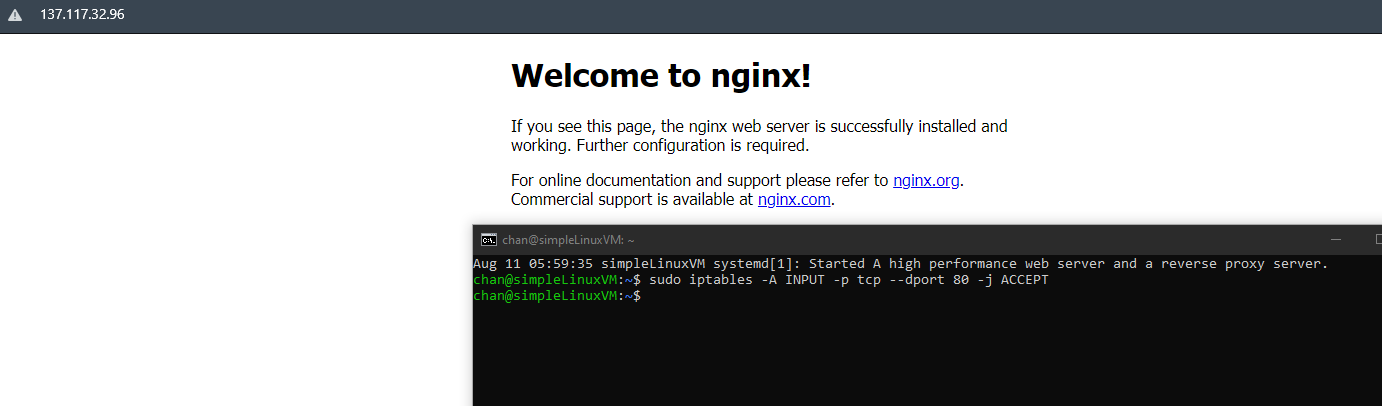


1. Verificamos la creación de los recursos y asignamos una regla de entrada al grupo de seguridad de la red para permitir el tráfico del puerto 80





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



Definiciones

1. **Resource Group :** es un conjunto de recursos que comparten el mismo ciclo de vida, permisos y políticas
2. **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.

1. **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.
2. **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.
3. **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.
4. **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.