



# *CLOUD SCULPTOR*

---

Terraform + Microsoft Azure

Andrés Paredes Fernández / Javier Ramírez Moral

20 de mayo del 2024



|  |          |
|--|----------|
| <b>Despliegue masivo de máquinas con una VM image personal .....</b>     | <b>2</b> |
| Materiales Requeridos .....  | 3        |
| ¿Qué es Terraform? .....   | 3        |
| ¿Qué es Azure? .....   | 4        |
| Guía Paleta de Colores .....   | 5        |
| Diseño Del Proyecto.....   | 5        |
| Materiales Requeridos .....  | 5        |
| Operaciones .....  | 6        |
| 1. Creación de la Virtual Machine principal para crear la VM image ..... | 6        |
| 2. Personalización de la VM image.....                                   | 12       |
| 3. Creación de la VM image .....   | 14       |
| 4. Creación de una Virtual Machine con la VM image .....                 | 16       |



## Despliegue masivo de máquinas con una VM image personal.

### Introducción

En lugar de configurar manualmente cada máquina individualmente, este enfoque permite crear una imagen de máquina virtual con una configuración específica, que luego puede ser replicada y distribuida de manera eficiente en múltiples dispositivos. Esta metodología no solo ahorra tiempo, sino que también garantiza consistencia y uniformidad en toda la infraestructura.

### Objetivos del Proyecto

**Automatización del Despliegue:** Implementar un sistema automatizado que permita la creación y distribución de máquinas virtuales a gran escala utilizando una imagen personalizada.

**Eficiencia Operativa:** Reducir significativamente el tiempo y los recursos necesarios para desplegar nuevas máquinas virtuales al eliminar la necesidad de configuraciones manuales repetitivas.

**Consistencia y Uniformidad:** Garantizar que todas las máquinas desplegadas estén configuradas de acuerdo con los estándares definidos y utilizando la misma imagen base, lo que minimiza errores y asegura la coherencia en toda la infraestructura.

**Flexibilidad y Escalabilidad:** Diseñar el sistema de despliegue de manera que sea fácilmente escalable para adaptarse a las necesidades cambiantes del entorno empresarial, permitiendo la rápida implementación de nuevas máquinas virtuales según sea necesario.

**Seguridad:** Integrar medidas de seguridad robustas para proteger la integridad y la confidencialidad de los datos en todas las etapas del proceso de despliegue y operación de las máquinas virtuales.

**Monitoreo y Gestión:** Implementar herramientas de monitoreo y gestión para supervisar el rendimiento y el estado de las máquinas virtuales desplegadas, facilitando la detección y resolución proactiva de problemas.

En resumen, el proyecto tiene como objetivo transformar y optimizar el proceso de despliegue de máquinas virtuales a gran escala, mejorando la eficiencia operativa, la consistencia, la seguridad y la capacidad de respuesta de la infraestructura tecnológica de la organización.



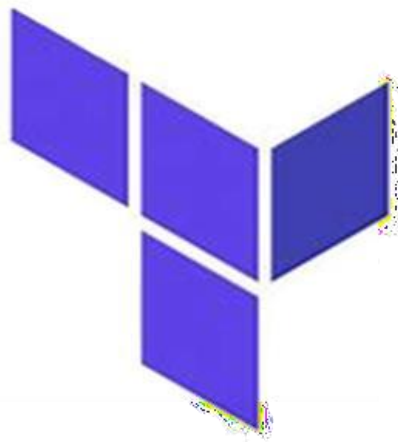
## Materiales Requeridos

### ¿Qué es Terraform?

Terraform es una herramienta de código abierto desarrollada por HashiCorp que se utiliza para la automatización de la infraestructura. Permite a los usuarios definir y provisionar de manera declarativa la infraestructura y los servicios en la nube mediante la descripción de la configuración en un formato fácil de entender y mantener.

Algunas de las características clave de Terraform incluyen:

- **Infraestructura como Código (IaC):** Terraform sigue el principio de IaC, lo que significa que la infraestructura se define mediante código en lugar de configuraciones manuales. Esto facilita la automatización y la gestión eficiente de la infraestructura.
- **Declarativo:** La configuración de Terraform describe el estado deseado de la infraestructura, y no la serie de pasos para llegar a ese estado. Terraform se encarga de determinar los cambios necesarios y aplicarlos.
- **Multiplataforma:** Terraform es compatible con múltiples proveedores de servicios en la nube, como AWS, Azure, Google Cloud Platform, y otros, así como proveedores de servicios on-premise. Esto permite gestionar infraestructuras heterogéneas y MultiCloud.
- **Reutilización de Código:** Los módulos en Terraform permiten la reutilización de configuraciones. Puedes definir módulos para encapsular y compartir fragmentos de configuración, lo que facilita la estandarización y la reutilización en proyectos más grandes.
- **Ciclo de Vida de la Infraestructura:** Terraform gestiona el ciclo de vida completo de la infraestructura, incluyendo la creación, actualización y destrucción de recursos. Esto proporciona una forma controlada y predecible de gestionar la infraestructura.



*Página de Terraform*



## ¿Qué es Azure?

Azure se refiere a Microsoft Azure, que es la plataforma de servicios en la nube de Microsoft. Es un conjunto de servicios en la nube que incluye almacenamiento, redes, bases de datos, análisis, inteligencia artificial, servicios de desarrollo y más. Microsoft Azure permite a las empresas construir, implementar y administrar aplicaciones y servicios a través de una red global de centros de datos administrados por Microsoft.

- **Escalabilidad:** Azure proporciona la capacidad de escalar recursos de manera flexible según las necesidades, ya sea aumentando o disminuyendo la capacidad de cómputo, almacenamiento o redes.
- **Diversidad de Servicios:** Ofrece una amplia variedad de servicios, incluyendo cómputo, almacenamiento, bases de datos, inteligencia artificial, aprendizaje automático, IoT, analítica de datos, servicios de desarrollo y más.
- **Híbrido y Multinube:** Azure permite la integración de entornos en la nube y locales, facilitando la creación de soluciones híbridas. Además, es compatible con la implementación en entornos multinube, permitiendo a los usuarios integrar servicios de otras plataformas en sus soluciones.
- **Seguridad Avanzada:** Proporciona herramientas y servicios para garantizar la seguridad de los datos y las aplicaciones, incluyendo Azure Active Directory para la gestión de identidades, Azure Security Center para la supervisión de la seguridad, y características como cifrado y control de acceso.
- **Conformidad y Certificaciones:** Cumple con numerosos estándares de seguridad y privacidad, y cuenta con certificaciones de conformidad reconocidas a nivel mundial, lo que facilita el cumplimiento de regulaciones específicas de la industria.
- **Desarrollo Ágil:** Facilita prácticas ágiles de desarrollo con herramientas como Azure DevOps, que incluye capacidades de desarrollo, pruebas, entrega continua, y más.



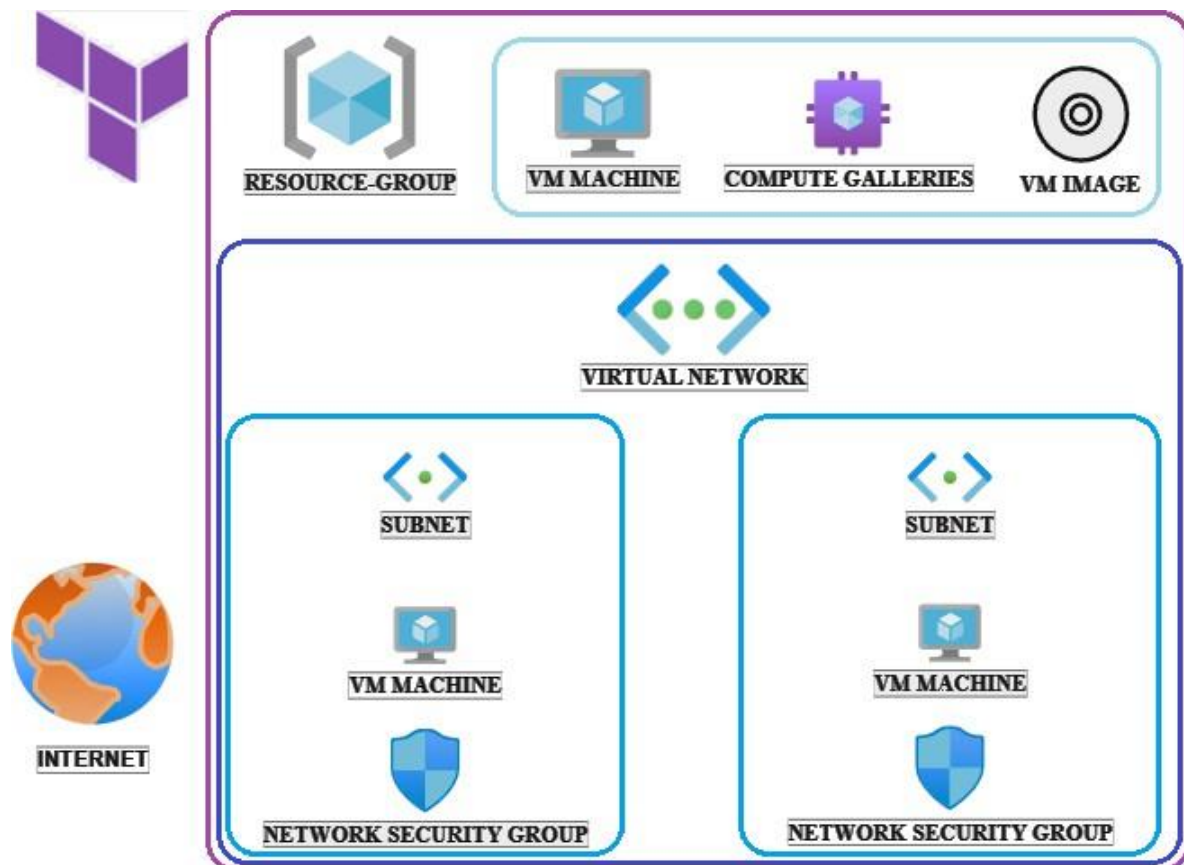
*Logo Azure*



## Guía Paleta de Colores

- Morado: Terraform
- Azul: Azure

## Diseño Del Proyecto



*Escenario del proyecto*

## Materiales Requeridos

- Cuenta de Microsoft Azure con suscripción
- Ordenador con Visual Studio Code
- Terminal con Azure CLI
- Sistema Terraform instalado

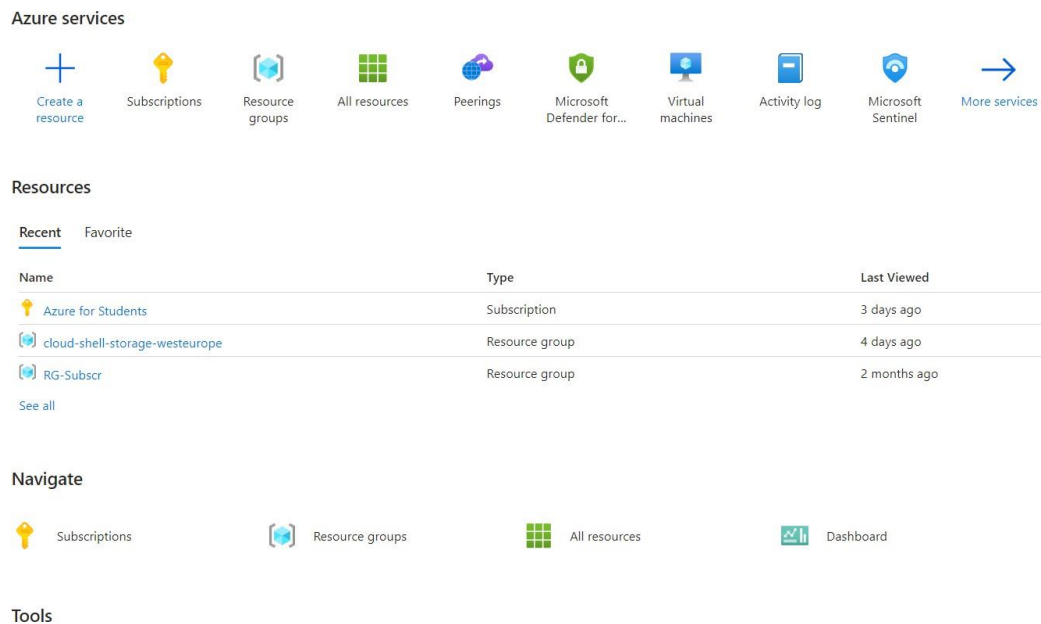


## Operaciones

### 1. Creación de la Virtual Machine principal para crear la VM image

Lo primero que tendremos que hacer es crear es una máquina virtual la cual utilizaremos para crear nuestra primera VM image.

Crearemos esta máquina virtual principal mediante el portal de Azure, tendremos que empezar logueandonos con nuestro usuario.



*Ilustración 1: Portal de Azure.*



Lo siguiente que haremos será, en la barra de búsqueda, buscar “Virtual Machines”.

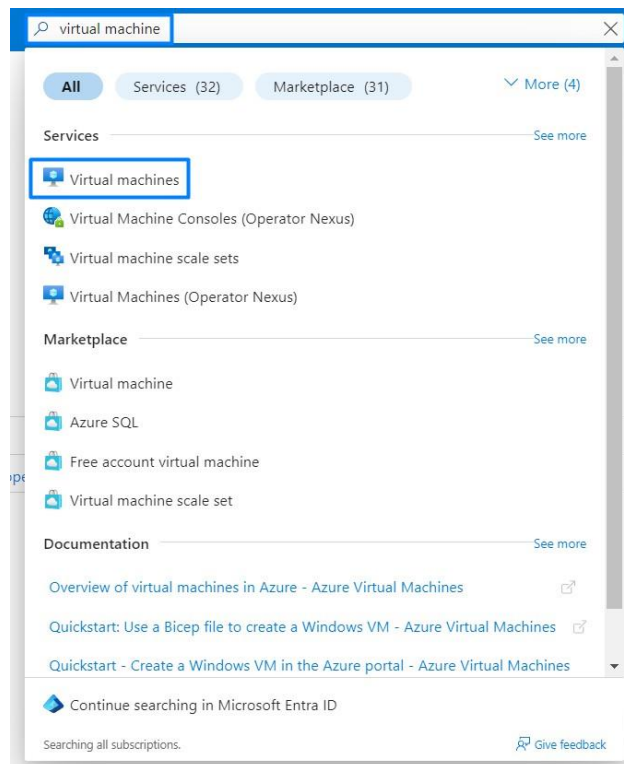


Ilustración 2: Recurso Virtual Machine.

Una vez estamos dentro del recurso, tendremos que seleccionar “Create > Azure virtual machine”.

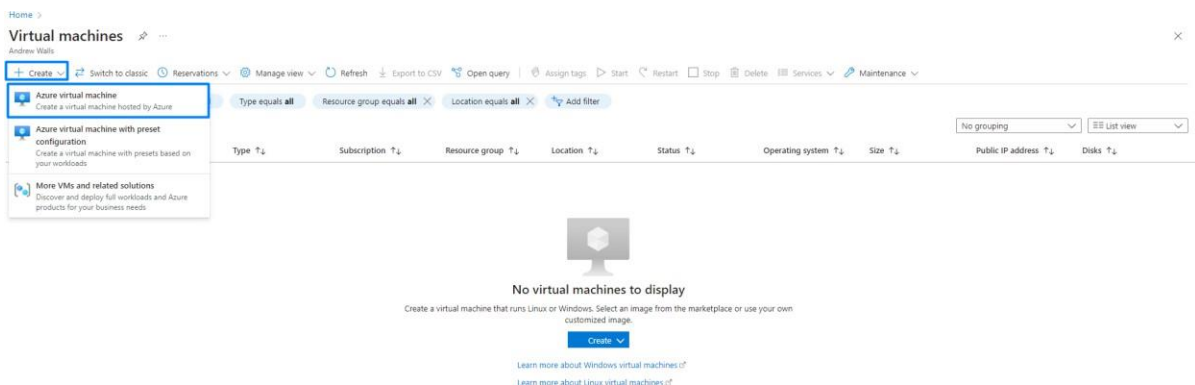


Ilustración 3: Creación de la Virtual Machine.





Una vez estamos dentro, los parámetros que configuremos en el apartado “Basics” son los siguientes:

- Resource Group: “Create new > CloudSculptor”.
- Virtual machine name: “VMPrincipal”.
- Security Type: “Standard”.
- Image: (El usuario puede elegir cualquier imagen para esta VM, esta será la base para nuestra VM image).
- Size: “Standard\_B2s”.
- Username: “Student”.
- Password: “\$Coob1...D00”.

[Home](#) > [Virtual machines](#) >

## Create a virtual machine ...

**This subscription may not be eligible to deploy VMs of certain sizes in certain regions.**

**Project details**  
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources. [Learn more](#)

Subscription \*

Resource group \*   
[Create new](#)

**Instance details**

Virtual machine name \*

Region \*

Availability options

Availability zone \*

☒ You can now select multiple zones. Selecting multiple zones will create one VM per zone. [Learn more](#)

Security type

Image \*   
[See all images](#) | [Configure VM generation](#)

VM architecture ☐ Arm64 ☒ x64

**Arm64 is not supported with the selected image.**

Ilustración 4: Parámetros de configuración de la VM.



[Home](#) > [Virtual machines](#) >

## Create a virtual machine

**i** Arm64 is not supported with the selected image.

Run with Azure Spot discount ⓘ

☐

Size \* ⓘ

Standard\_B2s - 2 vcpus, 4 GiB memory (36,21 US\$/month) ▼

[See all sizes](#)

Enable Hibernation ⓘ

☐

**i** Hibernation is not supported by the size that you have selected. Choose a size that is compatible with Hibernation to enable this feature. [Learn more](#) ⓘ

### Administrator account

Username \* ⓘ

Student ✓

Password \*

..... ✓

Confirm password \*

..... ✓

### Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

[< Previous](#)

[Next : Disks >](#)

[Review + create](#)

Ilustración 5: Parámetros de configuración de la VM.



El parámetro que configuremos en el apartado “Disks” es el siguiente:

- OS disk type: “Standard HDD”

Home > Virtual machines >

## Create a virtual machine

Basics **Disks** Networking Management Monitoring Advanced Tags Review + create

**Warning:** The desired performance might not be reached due to the maximum virtual machine disk performance cap. The current virtual machine size supports up to 23 MBps. The total for disks attached to 'VMPrincipal' is 60 MBps. [Learn more](#)

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. [Learn more](#)

**VM disk encryption**

Azure disk storage encryption automatically encrypts your data stored on Azure managed disks (OS and data disks) at rest by default when persisting it to the cloud.

Encryption at host ☐

**OS disk**

OS disk size

OS disk type 

The selected VM size supports premium disks. We recommend Premium SSD for high IOPS workloads. Virtual machines with Premium SSD disks qualify for the 99.9% connectivity SLA.

Delete with VM ☒

Key management

Enable Ultra Disk compatibility ☐

**Data disks for VMPrincipal**

You can add and configure additional data disks for your virtual machine or attach existing disks. This VM also comes with a temporary disk.

| LUN | Name | Size (GiB) | Disk type | Host caching | Delete with VM |
|-----|------|------------|-----------|--------------|----------------|
|     |      |            |           |              |                |

< Previous **Next : Networking >** Review + create

Ilustración 6: Parámetros de configuración de la VM.

Los parámetros que configuremos en el apartado “Networking” los dejaremos por defecto.



El parámetro que configuremos en el apartado “Monitoring” es el siguiente:

- Boot diagnostics: “Disable”

[Home](#) > [Virtual machines](#) >

## Create a virtual machine ...

Basics   Disks   Networking   Management   **Monitoring**   Advanced   Tags   Review + create

Configure monitoring options for your VM.

**Alerts**

Enable recommended alert rules ⓘ ☐

**Diagnostics**

Boot diagnostics ⓘ

☐ Enable with managed storage account (recommended)

☐ Enable with custom storage account

☒ **Disable**

Enable OS guest diagnostics ⓘ ☐

**Health**

Enable application health monitoring ⓘ ☐

---

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

Ilustración 7: Parámetros de configuración de la VM.

Con todo esto configurado, simplemente tendremos que ir a “Review + create” y seleccionar “Create”.



## 2. Personalización de la VM image

Lo que tendremos que hacer para que nuestra VM image tenga la configuración que nosotros queremos, tendremos que ir a “Connect > Connect”.

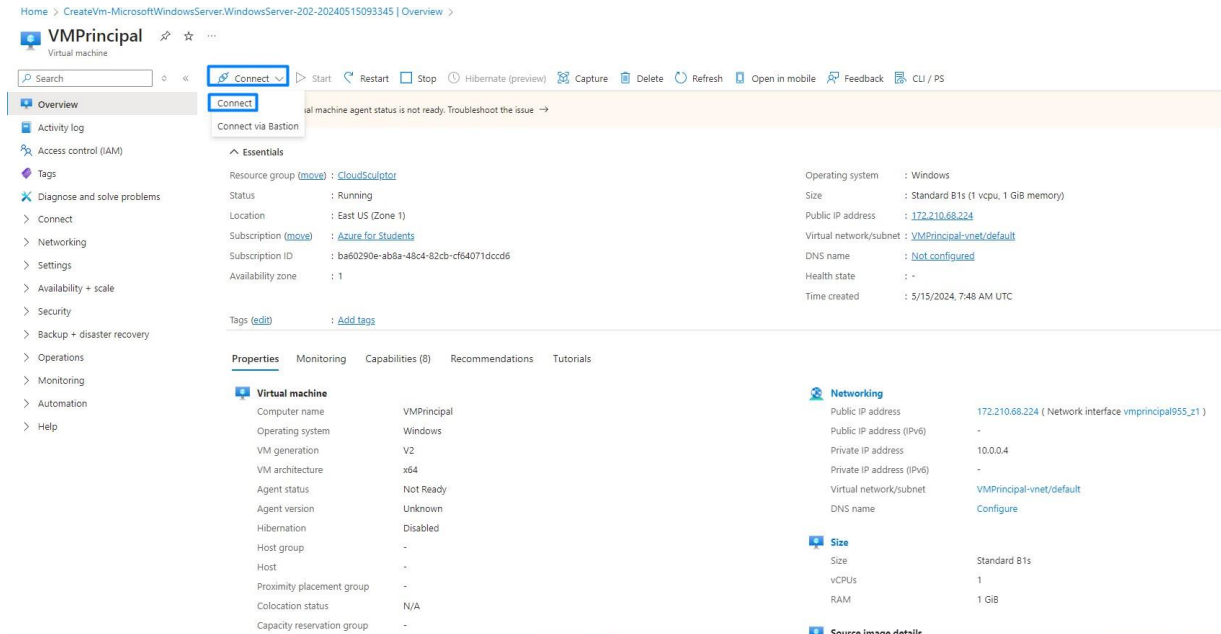


Ilustración 8: Conexión con nuestra VM.

Una vez estemos en “Connect”, tendremos que ir a “Native RDP > Download RDP File”

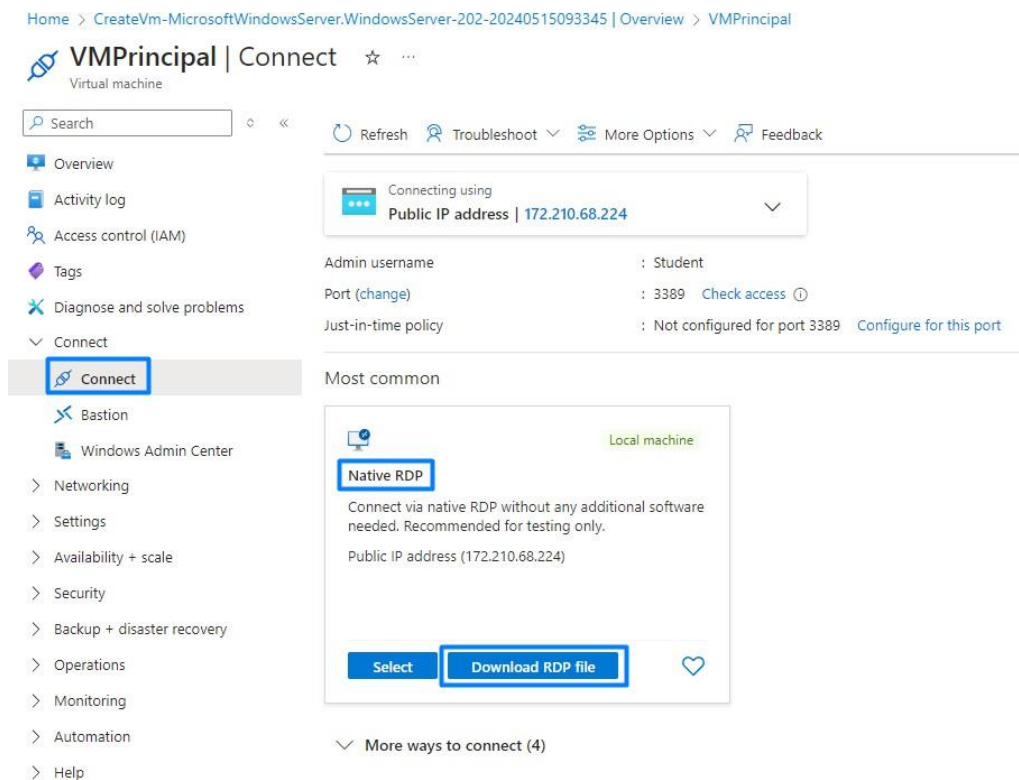
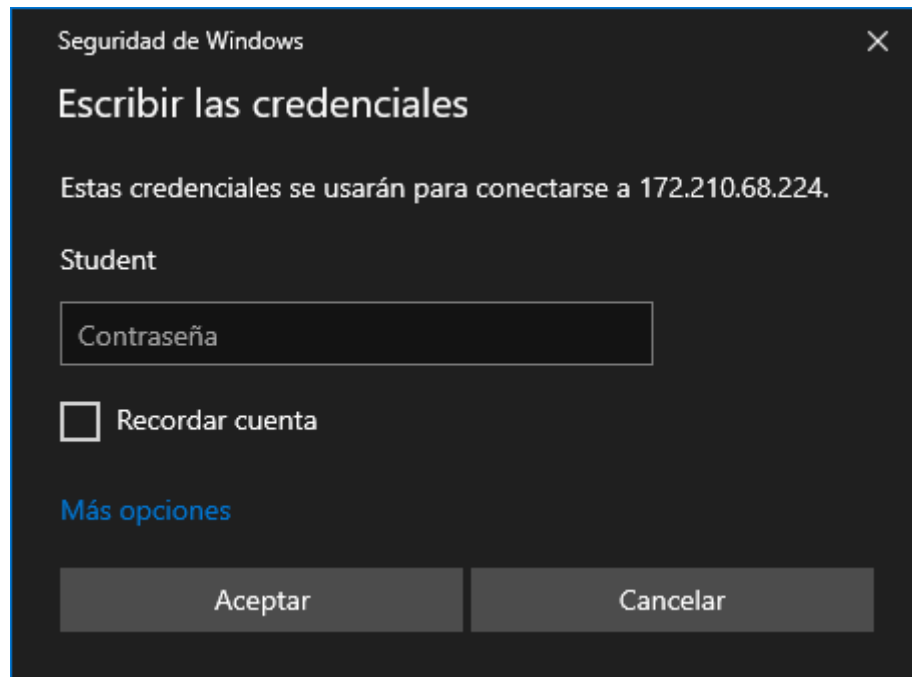


Ilustración 9: Conexión con nuestra VM.



Tendremos descargado un archivo .rdp, lo iniciaremos, diremos que nos queremos conectar, escribiremos nuestras credenciales y nos conectaremos a nuestra VM.



*Ilustración 10: Conexión con nuestra VM.*

Y listo, una vez dentro, instalaremos y configuraremos nuestra VM para luego tener su VM image (Es importante hacer un sysprep del sistema para evitar conflicto entre las máquinas).



### 3. Creación de la VM image

Ya tenemos nuestra VM con todo listo para la VM image, lo que tendremos que hacer es cerrar la ventana de la conexión remota de nuestra VM y dentro del portal de Azure, pararemos la maquina en cuestión (Al crear la VM image, esta máquina quedará inutilizable).

Home > CreateVm-MicrosoftWindowsServer.WindowsServer-202-20240515093345 | Overview >

VMPrincipal Virtual machine

Search

Connect Start Restart **Stop** Hibernate (preview) Capture Delete Refresh Open in mobile Feedback CLI / PS

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Connect

Connect

Bastion

Windows Admin Center

Networking

Settings

Availability + scale

Security

Backup + disaster recovery

Operations

Monitoring

Automation

Help

Essentials

Resource group (move) : CloudSculptor

Status : Running

Location : East US (Zone 1)

Subscription (move) : Azure for Students

Subscription ID : ba60290e-ab8a-48c4-82cb-cf64071dcd6

Availability zone : 1

Tags (edit) : Add tags

Operating system : Windows (Windows Server 2022 Datacenter Azure Edition)

Size : Standard B1s (1 vcpu, 1 GiB memory)

Public IP address : 172.210.68.224

Virtual network/subnet : VMPrincipal-vnet/default

DNS name : Not configured

Health state : -

Time created : 5/15/2024, 7:48 AM UTC

Properties Monitoring Capabilities (8) Recommendations Tutorials

Virtual machine

Computer name : VMPrincipal

Operating system : Windows (Windows Server 2022 Datacenter Azure Edition)

VM generation : V2

VM architecture : x64

Agent status : Ready

Agent version : 2.7.41491.1121

Hibernation : Disabled

Host group : -

Host : -

Proximity placement group : -

Colocation status : N/A

Capacity reservation group : -

Disk controller type : SCSI

Networking

Public IP address : 172.210.68.224 ( Network interface vmprincipal955\_21 )

Public IP address (IPv6) : -

Private IP address : 10.0.0.4

Private IP address (IPv6) : -

Virtual network/subnet : VMPrincipal-vnet/default

DNS name : Configure

Size

Size : Standard B1s

vCPUs : 1

RAM : 1 GiB

Source image details

Source image publisher : MicrosoftWindowsServer

Source image offer : WindowsServer

Ilustración 11: Preparación de la VM image.

Una vez la tenemos parada, tendremos que ir al apartado “Capture”.

Home > CreateVm-MicrosoftWindowsServer.WindowsServer-202-20240515093345 | Overview >

VMPrincipal Virtual machine

Search

Connect Start Restart Stop **Capture** Delete Refresh Open in mobile Feedback CLI / PS

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Connect

Connect

Bastion

Windows Admin Center

Networking

Settings

Availability + scale

Security

Backup + disaster recovery

Operations

Monitoring

Automation

Help

Essentials

Resource group (move) : CloudSculptor

Status : Stopped (deallocated)

Location : East US (Zone 1)

Subscription (move) : Azure for Students

Subscription ID : ba60290e-ab8a-48c4-82cb-cf64071dcd6

Availability zone : 1

Tags (edit) : Add tags

Operating system : Windows

Size : Standard B1s (1 vcpu, 1 GiB memory)

Public IP address : 172.210.68.224

Virtual network/subnet : VMPrincipal-vnet/default

DNS name : Not configured

Health state : -

Time created : 5/15/2024, 7:48 AM UTC

Properties Monitoring Capabilities (8) Recommendations Tutorials

Virtual machine

Computer name : VMPrincipal

Operating system : Windows

VM generation : V2

VM architecture : x64

Hibernation : Disabled

Host group : -

Host : -

Proximity placement group : -

Colocation status : N/A

Capacity reservation group : -

Disk controller type : SCSI

Networking

Public IP address : 172.210.68.224 ( Network interface vmprincipal955\_21 )

Public IP address (IPv6) : -

Private IP address : 10.0.0.4

Private IP address (IPv6) : -

Virtual network/subnet : VMPrincipal-vnet/default

DNS name : Configure

Size

Size : Standard B1s

vCPUs : 1

RAM : 1 GiB

Source image details

Source image publisher : MicrosoftWindowsServer

Source image offer : WindowsServer

Ilustración 12: Preparación de la VM image.



Los parámetros que configuraremos en el apartado “Basics” son:

- Automatically delete this virtual machine after creating the image: Marca la casilla en el caso de que quieras eliminar la maquina una vez tengas la imagen, es lo más recomendable.
- Target Azure compute gallery: “Create new > VM\_Principal”
- Target VM image definition: “Create new >
  - VM image definition name: “VM\_ISO”
  - Publisher: “MicrosoftWindowsServer”
  - Offer: “WindowsServer”
  - SKU: “2022-principal”
- Versión number: “0.0.1”
- End of life time: (El usuario puede elegir cualquier fecha en la que se eliminara la VM image).

Ilustración 13: Preparación de la VM image.

Una vez tenemos todo esto, simplemente iremos a “Review + Create” y tendremos nuestra VM image.





## 4. Creación de una Virtual Machine con la VM image

Para crear una máquina virtual utilizando la VM image utilizaremos los archivos de Terraform del repositorio de GitHub

[https://github.com/CloudSculptorProject/Terraform\\_X\\_Azure](https://github.com/CloudSculptorProject/Terraform_X_Azure)

(ESTE SCRIPT DE A CONTINUACIÓN ES SOLO PARA SISTEMAS WINDOWS, EN EL REPOSITORIO DE GITHUB ESTAN LOS CODIGOS PARA SISTEMAS WINDOWS Y LINUX)

En el tenemos los siguientes códigos:

### PROVIDERS.TF

Este archivo .tf lo usamos para decir al terraform que lo usaremos para la nube de Azure.

```
provider "azurerm" {  
  features {}  
}
```

### VARIABLES.TF

Este archivo .tf lo usamos para que el usuario escriba cuantas maquinas quiere en el momento en el que ejecute el script.

```
variable "vm_count" {  
  description = "Number of Virtual Machine you want to create"  
  type        = number  
}
```



## MAIN.TF

Este script de Terraform aprovisiona recursos de Azure para implementar máquinas virtuales (VM) de Windows dentro de una red virtual. Analicemos el guion:

1. **Resource Group creation:** Crea un grupo de recursos de Azure denominado "CloudSculptor" en la región "eastus".
2. **Locate the existing custom image from the Shared Image Gallery:** Localiza la VM image que hemos creado para utilizarla en las VM que crearemos.
3. **Virtual Network creation:** Crea una red virtual denominada "winvnetexamen" dentro del grupo de recursos creado anteriormente. Tiene un espacio de direcciones de "10.0.0.0/16".
4. **Subnet creation within the Virtual Network:** Crea una subred denominada "interna" dentro de la red virtual. La subred utiliza el espacio de direcciones "10.0.2.0/24".
5. **Windows VM creation:** Crea máquinas virtuales de Windows dentro del grupo de recursos y la red virtual. Crea una VM por el recuento especificado (controlado por la variable "var.vm\_count"). Cada VM está asociada con una interfaz de red, utiliza un tamaño de VM específico ("Standard\_B2s") y está configurada con un perfil de sistema operativo y configuraciones específicas de Windows.
5. **Network Interface creation for each VM:** Crea interfaces de red para cada VM. Cada interfaz está asociada con la subred creada anteriormente y asigna dinámicamente direcciones IP privadas.
6. **Public IP creation for each VM:** Crea direcciones IP públicas dinámicas para cada VM.
7. **Network Security Group creation for each VM:** Crea grupos de seguridad de red (NSG) para cada VM. Cada NSG contiene una regla entrante que permite el tráfico RDP (Protocolo de escritorio remoto) en el puerto 3389.
8. **Associate NSG with each VM's NIC:** Asocia cada NSG con la interfaz de red de la VM correspondiente.
9. **Output the public IP addresses of the Windows VMs:** Genera las direcciones IP públicas de las máquinas virtuales de Windows.



**# Resource Group creation**

```
resource "azurerm_resource_group" "windows_rg" {  
  name    = "Maquinas-Virtuales-Windows"  
  location = "eastus"  
}
```

**# Locate the existing custom image from the Shared Image Gallery**

```
data "azurerm_shared_image" "main" {  
  name          = "VM_ISO"  
  gallery_name   = "VM_Principal"  
  resource_group_name = "CloudSculptor"  
}
```

**# Virtual Network creation**

```
resource "azurerm_virtual_network" "windows_vnet" {  
  name          = "winvnetexamen"  
  resource_group_name = azurerm_resource_group.windows_rg.name  
  location      = azurerm_resource_group.windows_rg.location  
  address_space = ["10.0.0.0/16"]  
}
```

**# Subnet creation within the Virtual Network**

```
resource "azurerm_subnet" "windows_subnet" {  
  name          = "internal"  
  resource_group_name = azurerm_resource_group.windows_rg.name  
  virtual_network_name = azurerm_virtual_network.windows_vnet.name  
  address_prefixes = ["10.0.2.0/24"]  
}
```



## # Windows VM creation

```
resource "azurerm_virtual_machine" "windows_vm" {  
  count          = var.vm_count  
  
  name          = "win-vm${count.index}"  
  resource_group_name = azurerm_resource_group.windows_rg.name  
  location      = azurerm_resource_group.windows_rg.location  
  network_interface_ids = [azurerm_network_interface.windows_nic[count.index].id]  
  vm_size       = "Standard_B2s"  
  
  storage_image_reference {  
    id = data.azurem_shared_image.main.id  
  }  
  storage_os_disk {  
    name          = "win-myosdisk${count.index}"  
    caching       = "ReadWrite"  
    create_option = "FromImage"  
    managed_disk_type = "Standard_LRS"  
  }  
  os_profile {  
    computer_name = "win-vm${count.index}"  
    admin_username = "Student"  
    admin_password = "$Coob1...D00"  
  }  
  os_profile_windows_config {  
    provision_vm_agent      = true  
    enable_automatic_upgrades = true  
    timezone                = "UTC"  
  }  
}
```



# Network Interface creation for each VM

```
resource "azurerm_network_interface" "windows_nic" {  
  count = var.vm_count  
  
  name          = "win-example-nic${count.index}"  
  location      = azurerm_resource_group.windows_rg.location  
  resource_group_name = azurerm_resource_group.windows_rg.name  
  
  ip_configuration {  
    name          = "internal"  
    subnet_id      = azurerm_subnet.windows_subnet.id  
    private_ip_address_allocation = "Dynamic"  
    public_ip_address_id      = azurerm_public_ip.windows_ip[count.index].id  
  }  
}
```

# Public IP creation for each VM

```
resource "azurerm_public_ip" "windows_ip" {  
  count      = var.vm_count  
  name       = "win-example-ip${count.index}"  
  resource_group_name = azurerm_resource_group.windows_rg.name  
  location   = azurerm_resource_group.windows_rg.location  
  allocation_method = "Dynamic"  
}
```



# Network Security Group creation for each VM

```
resource "azurerm_network_security_group" "windows_nsg" {  
  count          = var.vm_count  
  name           = "win-nsg-${count.index}"  
  location       = azurerm_resource_group.windows_rg.location  
  resource_group_name = azurerm_resource_group.windows_rg.name  
  
  security_rule {  
    name              = "Allow-RDP-${count.index}"  
    priority          = 100  
    direction         = "Inbound"  
    access            = "Allow"  
    protocol          = "Tcp"  
    source_port_range = "*"   
    destination_port_range = "3389"  
    source_address_prefix = "*"   
    destination_address_prefix = "*"   
  }  
}
```

# Associate NSG with each VM's NIC

```
resource "azurerm_network_interface_security_group_association" "nsg_association" {  
  count          = var.vm_count  
  network_interface_id = azurerm_network_interface.windows_nic[count.index].id  
  network_security_group_id =  
  azurerm_network_security_group.windows_nsg[count.index].id  
}
```

# Output the public IP addresses of the Windows VMs

```
output "windows_public_ips" {  
  value = [azurerm_public_ip.windows_ip[*].ip_address]  
}
```



Deberemos tener estos archivos dentro del dispositivo el cual tiene el Azure CLI, una vez lo tengamos, iniciaremos la terminal del sistema.



Iremos hasta la ruta donde tenemos el archivo de Terraform con el comando

```
cd .\Ruta\Al\Fichero\
```

Tendremos que comenzar escribiendo el comando:

`az login`

Una vez que estemos conectados a nuestro portal de Azure mediante CLI, ejecutaremos los siguientes comandos:

`terraform init`

Sirve para instalar las últimas versiones de los providers de Terraform.

`terraform plan`

Sirve para ejecutar una prueba del sistema, así comprobaremos si hay algún error en el sistema.

`terraform apply`

Sirve para ejecutar nuestro archivo de Terraform en el portal.

Cuando ejecutemos este script de Terraform, aparecerá un error, pero **NO** se alarmen, el sistema funciona correctamente. Terraform lo interpreta como problema ya que el archivo tarda mucho tiempo en completarse.

Y listo, con todo esto tendremos nuestras máquinas virtuales con una ISO personalizada.

Si queremos saber cuáles son las IP públicas de las VM para conectarnos posteriormente de forma remota, lo único que tendremos que hacer es iniciar el comando:

```
az network public-ip list --resource-group "nombre del grupo de recursos" --query "[].{Name:name, IPAddress: ipAddress}" --output table
```