

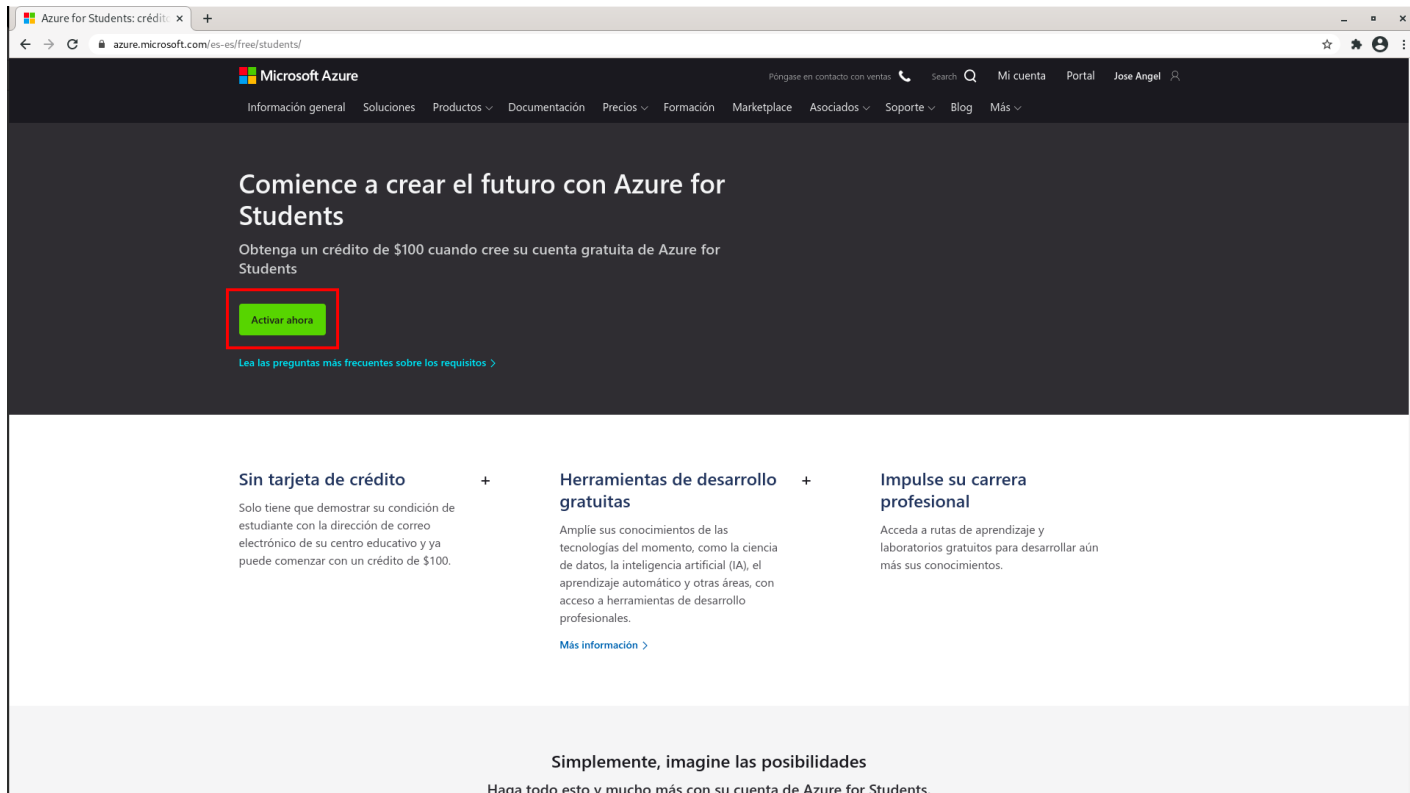
# Entornos Cloud en DevOps

José Ángel de Bustos Pérez

Azure

# Creando una cuenta Educational

- ▶ Ir a <https://azure.microsoft.com/es-es/free/students/>
- ▶ Pinchar en Activar Ahora y registrarse con la dirección de correo de UNIR y confirmar la cuenta con el correo que recibirás



# Instalando el CLI de Azure

- ▶ Aunque vamos a utilizar Terraform para crear la infraestructura necesitaremos el CLI de Azure para generar las credenciales para su uso con Terraform y también para buscar imágenes para desplegar.
- ▶ Podemos crear las credenciales también desde el portal.
- ▶ En <https://docs.microsoft.com/en-us/cli/azure/install-azure-cli> podemos encontrar las instrucciones para instalar el CLI de Azure en los diferentes Sistemas Operativos.

## Install the Azure CLI

09/25/2020 • 2 minutes to read • 

The Azure CLI is available to install in Windows, macOS and Linux environments. It can also be run in a Docker container and Azure Cloud Shell.

### Install

The current version of the Azure CLI is 2.19.0. For information about the latest release, see the [release notes](#). To find your installed version and see if you need to update, run [az version](#).

- [Install on Windows](#)
- [Install on macOS](#)
- [Install on Linux or Windows Subsystem for Linux \(WSL\)](#)
  - [Install with apt on Debian or Ubuntu](#)
  - [Install with yum on RHEL, Fedora, or CentOS](#)
  - [Install with zypper on openSUSE or SLE](#)
  - [Install from script](#)
- [Run in Docker container](#)
- [Run in Azure Cloud Shell](#)

# Creando un Service Principal

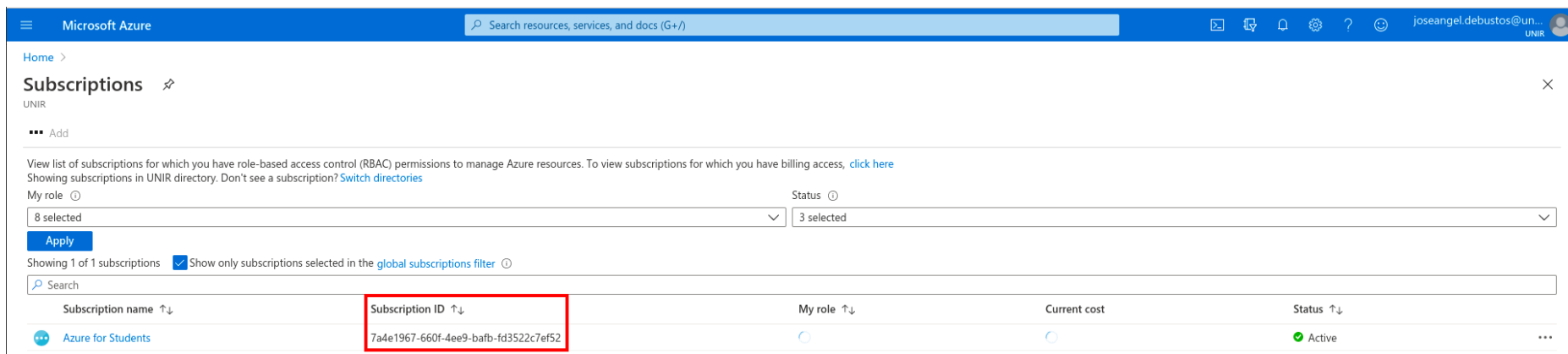
- ▶ Para poder autenticarnos con Terraform en Azure necesitaremos crear un Service Principal.
- ▶ Se puede crear en el [portal de Azure](#) directamente.
- ▶ Si queremos crear el [Service Principal desde el CLI](#) tendremos que iniciar sesión con el cli:

```
C:\> az login
```

- ▶ Se abrirá una pantalla de login en el navegador, donde introduciremos nuestras credenciales (correo de unir y contraseña).
- ▶ Si la autenticación es correcta ya podemos empezar a utilizar el CLI de Azure.

# Creando un Service Principal

- ▶ Una cosa que necesitaremos es conocer el subscription id.
- ▶ Una subscripción en Azure es un recurso que se utiliza para el proceso de billing.
- ▶ Con la cuenta educational no nos van a realizar ningún cargo, pero está limitada a \$ 100 y se utilizará para controlar el gasto hasta llegar a esos \$ 100.



Microsoft Azure

Search resources, services, and docs (G+)

Home > Subscriptions

UNIR

... Add

View list of subscriptions for which you have role-based access control (RBAC) permissions to manage Azure resources. To view subscriptions for which you have billing access, [click here](#)

Showing subscriptions in UNIR directory. Don't see a subscription? [Switch directories](#)

My role 8 selected Status 3 selected

Apply

Showing 1 of 1 subscriptions ☒ Show only subscriptions selected in the [global subscriptions filter](#)

Subscription name ↑↓	Subscription ID ↑↓	My role ↑↓	Current cost	Status ↑↓
Azure for Students	7a4e1967-660f-4ee9-bafb-fd3522c7ef52			Active

# Creando un Service Principal

- ▶ Una cosa que necesitaremos es conocer el subscription id.
- ▶ Una subscripción en Azure es un recurso que se utiliza para el proceso de billing.
- ▶ Con la cuenta educational no nos van a realizar ningún cargo, pero está limitada a \$ 100 y se utilizará para controlar el gasto hasta llegar a esos \$ 100.

# Creando un Service Principal

- ▶ Con el CLI de Azure configuramos la suscripción que vamos a utilizar:

```
C:\> az account set --subscription=7a4e1967-660f-4ee9-bafb-fd3522c7ef52
```

- ▶ A continuación ya podemos crear el Service Principal:

```
C:\> az ad sp create-for-rbac --role="Contributor"  
Creating 'Contributor' role assignment under  
scope    '/subscriptions/7a4e1967-660f-4ee9-bafb-fd3522c7ef52'  
{  
  "appId": "228ddc64-0bc9-46e4-b620-a12b8a7fc49a",  
  "displayName": "azure-cli-2021-02-07-10-10-51",  
  "name": "http://azure-cli-2021-02-07-10-10-51",  
  "password": "BW56bQG5-R2d2GQBH2k3Jas7H40g1Yb6",  
  "tenant": "476f4aea-4a06-4011-94e8-c24677f83422"  
}  
C:\
```

# Creando un Service Principal

- ▶ En **Azure Active Directory** -> **Enterprise Applications** en el portal de Azure si buscamos por el appld podemos encontrar el Service Principal.



# Autenticación en Azure con Terraform

- ▶ La autenticación se configura en el provider:

```
provider "azurerm" {  
  features {}  
  subscription_id = "7a4e1967-660f-4ee9-bafb-fd3522c7ef52"  
  client_id = "228ddc64-0bc9-46e4-b620-a12b8a7fc49a"      # appID  
  client_secret = "BW56bQG5-R2d2GQBH2k3Jas7H40g1Yb"      # password  
  tenant_id = "476f4aea-4a06-4011-94e8-c24677f83422"      # tenant  
}
```

# Buscando imágenes para desplegar

- ▶ Una cosa que tendremos que hacer es buscar imágenes de linux para desplegar.
- ▶ Es posible crear nuestra [propia imagen](#) linux para utilizarla en Azure.
- ▶ Desde el CLI podemos sacar la lista de imágenes disponibles:

```
[jadebustos@archimedes ~]$ az vm image list --output table
Command group 'vm' is experimental and under development. Reference and support levels: https://aka.ms/CLI_refstatus
You are viewing an offline list of images, use --all to retrieve an up-to-date list
```

Offer	Publisher	Skus	Urn	UrnAlias	Version
CentOS	OpenLogic	7.5	OpenLogic:CentOS:7.5:latest	CentOS	latest
CoreOS	CoreOS	Stable	CoreOS:CoreOS:Stable:latest	CoreOS	latest
debian-10	Debian	10	Debian:debian-10:10:latest	Debian	latest
openSUSE-Leap	SUSE	42.3	SUSE:openSUSE-Leap:42.3:latest	openSUSE-Leap	latest
RHEL	RedHat	7-LVM	RedHat:RHEL:7-LVM:latest	RHEL	latest
SLES	SUSE	15	SUSE:SLES:15:latest	SLES	latest
UbuntuServer	Canonical	18.04-LTS	Canonical:UbuntuServer:18.04-LTS:latest	UbuntuLTS	latest
WindowsServer	MicrosoftWindowsServer	2019-Datacenter	MicrosoftWindowsServer:WindowsServer:2019-Datacenter:latest	Win2019Datacenter	latest
WindowsServer	MicrosoftWindowsServer	2016-Datacenter	MicrosoftWindowsServer:WindowsServer:2016-Datacenter:latest	Win2016Datacenter	latest
WindowsServer	MicrosoftWindowsServer	2012-R2-Datacenter	MicrosoftWindowsServer:WindowsServer:2012-R2-Datacenter:latest	Win2012R2Datacenter	latest
WindowsServer	MicrosoftWindowsServer	2012-Datacenter	MicrosoftWindowsServer:WindowsServer:2012-Datacenter:latest	Win2012Datacenter	latest
WindowsServer	MicrosoftWindowsServer	2008-R2-SP1	MicrosoftWindowsServer:WindowsServer:2008-R2-SP1:latest	Win2008R2SP1	latest

```
[jadebustos@archimedes ~]$
```

# Buscando imágenes para desplegar

- ▶ Una cosa que tendremos que hacer es buscar imágenes de linux para desplegar.
- ▶ Es posible crear nuestra [propia imagen](#) linux para utilizarla en Azure.
- ▶ Desde el CLI podemos sacar la lista de imágenes disponibles:

```
C:\> az vm image list --offer CentOS --all --output table
```

- ▶ Conviene redireccionar la salida a un fichero.

# Buscando imágenes para desplegar

[jdebustos@archimedes ~]\$ az vm image list --offer CentOS --all --output table				
Command group 'vm' is experimental and under development. Reference and support levels: <a href="https://aka.ms/CLI_refstatus">https://aka.ms/CLI_refstatus</a>				
Offer	Publisher	Sku	Urn	Version
centos-solver195	ansys	74-rc2-hpc	ansys:centos-solver195:74-rc2-hpc:1.20.14	1.20.14
centos-solver195	ansys	74-rc2-hpc	ansys:centos-solver195:74-rc2-hpc:1.20.15	1.20.15
centos-solvers194	ansys	74-hpc	ansys:centos-solvers194:74-hpc:1.20.14	1.20.14
centos-solvers194	ansys	76-hc	ansys:centos-solvers194:76-hc:1.20.14	1.20.14
centos-solvers195	ansys	74-hpc	ansys:centos-solvers195:74-hpc:1.20.14	1.20.14
codiad-on-centos-75	apps-4-rent	codiad-on-centos-75	apps-4-rent:codiad-on-centos-75:codiad-on-centos-75:1.0.1	1.0.1
centos-solvers195	ansys	76-hc	ansys:centos-solvers195:76-hc:1.20.14	1.20.14
elasticsearch-on-centos	apps-4-rent	elasticsearch-on-centos	apps-4-rent:elasticsearch-on-centos:elasticsearch-on-centos:1.0.0	1.0.0
issabel-ip-pbx-on-centos	apps-4-rent	issabel-ip-pbx-on-centos	apps-4-rent:issabel-ip-pbx-on-centos:issabel-ip-pbx-on-centos:1.0.2	1.0.2
centos-7-ci-rl	atmosfera	centos-7-ci-rl	atmosfera:centos-7-ci-rl:centos-7-ci-rl:1.0.0	1.0.0
centos-7-ci-rl	atmosfera	centos-7-ci-rl	atmosfera:centos-7-ci-rl:centos-7-ci-rl:1.0.2	1.0.2
centos-solvers202	ansys	74-hpc	ansys:centos-solvers202:74-hpc:1.20.15	1.20.15
jenkins-on-centos-75	apps-4-rent	jenkins-on-centos-75	apps-4-rent:jenkins-on-centos-75:jenkins-on-centos-75:1.0.0	1.0.0
centos-solvers202	ansys	76-gpu	ansys:centos-solvers202:76-gpu:1.20.14	1.20.14
centos-solvers202	ansys	76-hc	ansys:centos-solvers202:76-hc:1.20.15	1.20.15
lamp-on-centos-75	apps-4-rent	lamp-on-centos-75	apps-4-rent:lamp-on-centos-75:lamp-on-centos-75:1.0.0	1.0.0
centos-solvers211	ansys	74-hpc	ansys:centos-solvers211:74-hpc:1.20.15	1.20.15
mariadb-on-centos	apps-4-rent	mariadb-on-centos	apps-4-rent:mariadb-on-centos:mariadb-on-centos:1.0.0	1.0.0
centos-solvers211	ansys	76-gpu	ansys:centos-solvers211:76-gpu:1.20.15	1.20.15
centos-solvers211	ansys	76-hc	ansys:centos-solvers211:76-hc:1.20.15	1.20.15
mediawiki-on-centos	apps-4-rent	mediawiki-on-centos	apps-4-rent:mediawiki-on-centos:mediawiki-on-centos:1.0.0	1.0.0
mongodb-on-centos-75	apps-4-rent	mongodb-on-centos-75	apps-4-rent:mongodb-on-centos-75:mongodb-on-centos-75:2.0.0	2.0.0
mysql-on-centos-75	apps-4-rent	mysql-on-centos-75	apps-4-rent:mysql-on-centos-75:mysql-on-centos-75:1.0.0	1.0.0
rendering-centos73	batch	rendering	batch:rendering-centos73:rendering:1.1.7	1.1.7
nginx-on-centos-75	apps-4-rent	nginx-on-centos-75	apps-4-rent:nginx-on-centos-75:nginx-on-centos-75:1.0.0	1.0.0
bissan_secure_linux_centos75	bissanttechnology1583581147809	bissantlc75	bissanttechnology1583581147809:bissan_secure_linux_centos75:bissantlc75:7.5.20201224	7.5.20201224
node-js-on-centos-75	apps-4-rent	node-js-on-centos-75	apps-4-rent:node-js-on-centos-75:node-js-on-centos-75:1.0.0	1.0.0
passenger-nginx-on-centos-75	apps-4-rent	passenger-nginx-on-centos-75	apps-4-rent:passenger-nginx-on-centos-75:passenger-nginx-on-centos-75:2.0.1	2.0.1
phpbb-on-centos-75	apps-4-rent	phpbb-on-centos-75	apps-4-rent:phpbb-on-centos-75:phpbb-on-centos-75:1.0.0	1.0.0
piwigo-on-centos-75	apps-4-rent	piwigo-on-centos-75	apps-4-rent:piwigo-on-centos-75:piwigo-on-centos-75:1.0.0	1.0.0
sqlite-on-centos-75	apps-4-rent	sqlite-on-centos-75	apps-4-rent:sqlite-on-centos-75:sqlite-on-centos-75:1.0.0	1.0.0
suitecrm-on-centos75	apps-4-rent	suitecrm-on-centos75	apps-4-rent:suitecrm-on-centos75:suitecrm-on-centos75:1.0.0	1.0.0
xeams-centos	apps-4-rent	xeams-centos-v1	apps-4-rent:xeams-centos:xeams-centos-v1:1.0.0	1.0.0
cis-centos-7-v2-1-l-1-l1	center-for-internet-security-inc	cis-centos7-l1	center-for-internet-security-inc:cis-centos-7-v2-1-l-1-l1:cis-centos7-l1:3.0.4	3.0.4
cis-centos-7-v2-1-l-1-l1	center-for-internet-security-inc	cis-centos7-l1	center-for-internet-security-inc:cis-centos-7-v2-1-l-1-l1:cis-centos7-l1:3.0.5	3.0.5
cis-centos-7-v2-1-l-1-l1	center-for-internet-security-inc	cis-centos7-l1	center-for-internet-security-inc:cis-centos-7-v2-1-l-1-l1:cis-centos7-l1:3.0.6	3.0.6
cis-centos-7-v2-1-l-1-l1	center-for-internet-security-inc	cis-centos7-l1	center-for-internet-security-inc:cis-centos-7-v2-1-l-1-l1:cis-centos7-l1:3.0.7	3.0.7
cis-centos-8-l1	center-for-internet-security-inc	cis-centos8-l1	center-for-internet-security-inc:cis-centos-8-l1:cis-centos8-l1:1.0.10	1.0.10
cis-centos-8-l1	center-for-internet-security-inc	cis-centos8-l1	center-for-internet-security-inc:cis-centos-8-l1:cis-centos8-l1:1.0.11	1.0.11
cis-centos-8-l1	center-for-internet-security-inc	cis-centos8-l1	center-for-internet-security-inc:cis-centos-8-l1:cis-centos8-l1:1.0.12	1.0.12
cis-nginx-centos-7-v1-1-0-l1	center-for-internet-security-inc	cis-nginx-centos7-l1	center-for-internet-security-inc:cis-nginx-centos-7-v1-1-0-l1:cis-nginx-centos7-l1:1.1.15	1.1.15
cis-nginx-centos-7-v1-1-0-l1	center-for-internet-security-inc	cis-nginx-centos7-l1	center-for-internet-security-inc:cis-nginx-centos-7-v1-1-0-l1:cis-nginx-centos7-l1:1.1.16	1.1.16
cis-nginx-centos-7-v1-1-0-l1	center-for-internet-security-inc	cis-nginx-centos7-l1	center-for-internet-security-inc:cis-nginx-centos-7-v1-1-0-l1:cis-nginx-centos7-l1:1.1.17	1.1.17
cis-postgresql-11-centos-linux-7-level-1	center-for-internet-security-inc	cis-postgresql-11-centos-linux-7-l1	center-for-internet-security-inc:cis-postgresql-11-centos-linux-7-level-1:cis-postgresql-11-centos-linux-7-l1:1.0.14	1.0.14
cis-postgresql-11-centos-linux-7-level-1	center-for-internet-security-inc	cis-postgresql-11-centos-linux-7-l1	center-for-internet-security-inc:cis-postgresql-11-centos-linux-7-level-1:cis-postgresql-11-centos-linux-7-l1:1.0.15	1.0.15
cis-postgresql-11-centos-linux-7-level-1	center-for-internet-security-inc	cis-postgresql-11-centos-linux-7-l1	center-for-internet-security-inc:cis-postgresql-11-centos-linux-7-level-1:cis-postgresql-11-centos-linux-7-l1:1.0.16	1.0.16
docker_centos	cloud-infrastructure-services	docker_centos	cloud-infrastructure-services:docker_centos:docker_centos:0.0.3	0.0.3
squid-centos7-7	cloud-infrastructure-services	squid-centos-7-7	cloud-infrastructure-services:squid-centos7-7:squid-centos-7-7:0.0.1	0.0.1
cloudera-centos-6	cloudera	cloudera-centos-6	cloudera:cloudera-centos-6:cloudera-centos-6:1.0.0	1.0.0
cloudera-centos-6	cloudera	cloudera-centos-6	cloudera:cloudera-centos-6:cloudera-centos-6:8.0.1	8.0.1
cloudera-centos-6	cloudera	cloudera-centos-6	cloudera:cloudera-centos-6:cloudera-centos-6:8.0.2	8.0.2
cloudera-centos-6	cloudera	cloudera-centos-6	cloudera:cloudera-centos-6:cloudera-centos-6:9.0.1	9.0.1
cloudera-centos-6	cloudera	cloudera-centos-6	cloudera:cloudera-centos-6:cloudera-centos-6:9.0.2	9.0.2
cloudera-centos-os	cloudera	6_7	cloudera:cloudera-centos-os:6_7:1.0.0	1.0.0
cloudera-centos-os	cloudera	6_7	cloudera:cloudera-centos-os:6_7:1.0.1	1.0.1
cloudera-centos-os	cloudera	6_7	cloudera:cloudera-centos-os:6_7:2.0.1	2.0.1
cloudera-centos-os	cloudera	6_7	cloudera:cloudera-centos-os:6_7:2.0.4	2.0.4

# Buscando imágenes para desplegar

- ▶ Una vez que hayamos encontrado una imagen que satisfaga nuestras necesidades será necesario activarla en nuestra suscripción lo que incluye aceptar las condiciones de uso (eso que no nos solemos leer).
- ▶ De no activarla al desplegar una imagen veremos un error como este:

```
Error: creating Linux Virtual Machine "my-first-azure-vm" (Resource Group
"kubernetes_rg"): compute.VirtualMachinesClient#CreateOrUpdate: Failure sending
request: StatusCode=400 -- Original Error: Code="ResourcePurchaseValidationFailed"
Message="User failed validation to purchase resources. Error message: 'You have not
accepted the legal terms on this subscription: '7a4e1967-660f-4ee9-bafb-fd3522c7ef52'
for this plan. Before the subscription can be used, you need to accept the legal
terms of the image. To read and accept legal terms, use the Azure CLI commands
described at https://go.microsoft.com/fwlink/?linkid=2110637 or the PowerShell
commands available at https://go.microsoft.com/fwlink/?linkid=862451. Alternatively,
deploying via the Azure portal provides a UI experience for reading and accepting the
legal terms. Offer details: publisher='procomputers' offer = 'centos-8-latest', sku =
'centos-8-latest', Correlation Id: '53f72879-c992-4b15-aadb-f6297f223221'.'"

```

# Buscando imágenes para desplegar

- Para ver la información de una imagen:

```
[jadebustos@archimedes ~]$ az vm image terms show --urn cognosys:centos-8-stream-free:centos-8-stream-free:1.2019.0810
Command group 'vm' is experimental and under development. Reference and support levels: https://aka.ms/CLI_refstatus
{
  "accepted": true,
  "id": "/subscriptions/7a4e1967-660f-4ee9-bafb-fd3522c7ef52/providers/Microsoft.MarketplaceOrdering/offerTypes/VirtualMachine/
  "licenseTextLink": "https://storelegalterms.blob.core.windows.net/legalterms/3E5ED_legalterms_COGNOSYS%253a24CENTOS%253a2D8%2
4UPT5LWRG35IQ7SJVHFM LGFEXMXKVQGI.txt",
  "name": "centos-8-stream-free",
  "plan": "centos-8-stream-free",
  "privacyPolicyLink": "http://www.cogno-sys.com/cognosys-technologies-partners/privacy-policy/",
  "product": "centos-8-stream-free",
  "publisher": "cognosys",
  "retrieveDatetime": "2021-02-09T23:27:15.9765595Z",
  "signature": "3G2I53X20FZ76YGE6PV62AN4IJWL25SUULTFMAXKICWW4SDTI24SUIDXWVSNSZ06GLRKNAYHFQDVBX3JJEDN3L6COJQMSBNIATMQI",
  "type": "Microsoft.MarketplaceOrdering/offertypes"
}
```

- Podemos ir a los enlaces para ver la licencia y la política de privacidad.

# Buscando imágenes para desplegar

- ▶ Si estamos de acuerdo con las condiciones de uso, las aceptamos:

```
[jadebustos@archimedes unir-cp2-count]$ az vm image accept-terms --urn cognosys:centos-8-stream-free:centos-8-stream-free:1.2019.0810
This command has been deprecated and will be removed in version '3.0.0'. Use 'az vm image terms accept' instead.
Command group 'vm' is experimental and under development. Reference and support levels: https://aka.ms/CLI_refstatus
{
  "accepted": true,
  "id": "/subscriptions/7a4e1967-660f-4ee9-bafb-fd3522c7ef52/providers/Microsoft.MarketplaceOrdering/offerTypes/Microsoft.MarketplaceOrdering/
/current",
  "licenseTextLink": "https://storelegalterms.blob.core.windows.net/legalterms/3E5ED_legalterms_COGNOSYS%253a24CENTOS%253a2D8%253a2DSTREAM%253
53IUP4XFKJY2B4QTN6L43QJMNSF7SRMTP24UPT5LWRG35IQ7SJVFMLGFEXMXKVQGI.txt",
  "name": "centos-8-stream-free",
  "plan": "centos-8-stream-free",
  "privacyPolicyLink": "http://www.cogno-sys.com/cognosys-technologies-partners/privacy-policy/",
  "product": "centos-8-stream-free",
  "publisher": "cognosys",
  "retrieveDatetime": "2021-02-09T21:52:41.7076239Z",
  "signature": "U5W26DDNAZ2XLGK0Y6CAU3DZ5S6R60HPSVRDDKWTMEBHIDFA74BUNKSQXP5ZZFBELZF6KJYMD70ATIYI6Z5RJ467GQHVDXJJKUJBPA",
  "type": "Microsoft.MarketplaceOrdering/offertypes"
}
[jadebustos@archimedes unir-cp2-count]$
```

- ▶ Una vez aceptadas ya podemos desplegar máquinas virtuales a partir de dicha imagen.
- ▶ <https://docs.microsoft.com/en-us/cli/azure/vm/image?view=azure-cli-latest>

# Desplegando una VM en Azure

- ▶ Necesitamos una imagen para desplegar la VM.
- ▶ Necesitamos crear una red y una subred.
- ▶ Necesitamos crear una NIC y asignarla a la subred para que se le asigne una dirección IP.
- ▶ Necesitamos crear una máquina virtual.
- ▶ Necesitamos asignar la NIC a la máquina virtual.
- ▶ Si queremos que la máquina virtual sea accesible desde el exterior de Azure será necesario crear un security group y asignarlo a la NIC.



# Terraform plan

- Crearemos un plan de Terraform:

```
[jadebustos@archimedes unir-cp2-terraform]$ tree single-vm/  
single-vm/  
├── main.tf  
├── network.tf  
├── security.tf  
├── vars.tf  
└── vm.tf  
  
0 directories, 5 files  
[jadebustos@archimedes unir-cp2-terraform]$
```

# Terraform plan (vars.tf)

- ▶ Declaramos las variables.
- ▶ Tenemos varias [localizaciones disponibles](#) y tendremos que elegir en que región desplegar.
- ▶ Especificamos el tamaño de la VM:

```
az vm list-sizes --location westeurope
```

```
variable "location" {  
  type = string  
  description = "Región de Azure donde crearemos la infraestructura"  
  default = "West Europe"  
}  
  
variable "vm_size" {  
  type = string  
  description = "Tamaño de la máquina virtual"  
  default = "Standard_D1_v2" # 3.5 GB, 1 CPU  
}
```

# Terraform plan (vars.tf)

```
[jadebustos@archimedes unir-cp2-count]$ az account list-locations -o table
DisplayName      Name      RegionalDisplayName
-----
East US          eastus    (US) East US
East US 2        eastus2   (US) East US 2
South Central US southcentralus (US) South Central US
West US 2        westus2   (US) West US 2
Australia East   australiaeast (Asia Pacific) Australia East
Southeast Asia   southeastasia (Asia Pacific) Southeast Asia
North Europe     northeurope (Europe) North Europe
UK South         uksouth   (Europe) UK South
West Europe      westeurope (Europe) West Europe
Central US        centralus  (US) Central US
North Central US northcentralus (US) North Central US
West US          westus    (US) West US
South Africa North southafricanorth (Africa) South Africa North
Central India     centralindia (Asia Pacific) Central India
East Asia         eastasia   (Asia Pacific) East Asia
Japan East        japaneast  (Asia Pacific) Japan East
Korea Central     koreacentral (Asia Pacific) Korea Central
Canada Central    canadacentral (Canada) Canada Central
France Central     francecentral (Europe) France Central
Germany West Central germanywestcentral (Europe) Germany West Central
Norway East        norwayeast (Europe) Norway East
Switzerland North switzerlandnorth (Europe) Switzerland North
UAE North          uaenorth   (Middle East) UAE North
Brazil South       brazilsouth (South America) Brazil South
Central US (Stage) centralusstage (US) Central US (Stage)
East US (Stage)    eastusstage (US) East US (Stage)
East US 2 (Stage)  eastus2stage (US) East US 2 (Stage)
North Central US (Stage) northcentralusstage (US) North Central US (Stage)
South Central US (Stage) southcentralusstage (US) South Central US (Stage)
West US (Stage)    westusstage (US) West US (Stage)
West US 2 (Stage)  westus2stage (US) West US 2 (Stage)
Asia              asia       Asia
Asia Pacific       asiapacific Asia Pacific
Australia          australia  Australia
Brazil            brazil     Brazil
Canada            canada     Canada
Europe            europe     Europe
Global            global     Global
India             india      India
Japan            japan      Japan
United Kingdom     uk         United Kingdom
United States       unitedstates United States
East Asia (Stage)  eastasiastage (Asia Pacific) East Asia (Stage)
Southeast Asia (Stage) southeastasiastage (Asia Pacific) Southeast Asia (Stage)
Central US EUAP    centraluseuap (US) Central US EUAP
East US 2 EUAP     eastus2euap (US) East US 2 EUAP
West Central US    westcentralus (US) West Central US
```

# Terraform plan (main.tf)

```
# https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs
terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "~>2.46.1"
    }
  }
}

# crea un service principal y rellena los siguientes datos para autenticar
provider "azurerm" {
  features {}
  subscription_id = "<SUBSCRIPTION ID>"
  client_id       = "<APP_ID>"
  client_secret   = "<PASSWORD>"
  tenant_id      = "<TENANT>"
}

# https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/resource_group
resource "azurerm_resource_group" "rg" {
  name     = "kubernetes_rg"
  location = var.location

  tags = {
    environment = "CP2"
  }
}

# Storage account
# https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/storage_account
resource "azurerm_storage_account" "stAccount" {
  name                = "staccountcp2"
  resource_group_name = azurerm_resource_group.rg.name
  location            = azurerm_resource_group.rg.location
  account_tier        = "Standard"
  account_replication_type = "LRS"

  tags = {
    environment = "CP2"
  }
}
```

- Configuramos el provider.
- Incluimos las credenciales (service principal).
- Creamos un grupo de recursos.
- Creamos una storage account.

# Terraform plan (network.tf)

- ▶ Definimos una red: **10.0.0.0/16**
- ▶ Definimos una subred: **10.0.1.0/24**
- ▶ La subred tiene que estar contenida dentro de la red.
- ▶ Podemos crear varias subredes, todas ellas incluidas dentro de la red creada.
- ▶ Creamos una NIC y la asignamos a la subnet.
- ▶ Le asignamos una dirección estática con ip: **10.0.1.10**.
- ▶ Reemplazando **Static** por **Dynamic** la ip se asignará por DHCP (private\_ip\_address no será necesario).

```
# Creación de red
resource "azurerm_virtual_network" "myNet" {
  name                = "kubernetesnet"
  address_space       = ["10.0.0.0/16"]
  location             = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  tags = {
    environment = "CP2"
  }
}

# Creación de subnet
resource "azurerm_subnet" "mySubnet" {
  name                 = "terraformsubnet"
  resource_group_name = azurerm_resource_group.rg.name
  virtual_network_name = azurerm_virtual_network.myNet.name
  address_prefixes     = ["10.0.1.0/24"]
}

# Create NIC
resource "azurerm_network_interface" "myNic1" {
  name                = "vmnic1"
  location             = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  ip_configuration {
    name                       = "myipconfiguration1"
    subnet_id                 = azurerm_subnet.mySubnet.id
    private_ip_address_allocation = "Static"
    private_ip_address        = "10.0.1.10"
    public_ip_address_id      = azurerm_public_ip.myPublicIp1.id
  }

  tags = {
    environment = "CP2"
  }
}

# IP pública
resource "azurerm_public_ip" "myPublicIp1" {
  name                = "vmip1"
  location             = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  allocation_method   = "Dynamic"
  sku                 = "Basic"

  tags = {
    environment = "CP2"
  }
}
```

# Terraform plan (network.tf)

```
# Creación de red
resource "azurerm_virtual_network" "myNet" {
  name            = "kubernetesnet"
  address_space   = ["10.0.0.0/16"]
  location        = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  tags = {
    environment = "CP2"
  }
}

# Creación de subnet
resource "azurerm_subnet" "mySubnet" {
  name                 = "terraformsubnet"
  resource_group_name = azurerm_resource_group.rg.name
  virtual_network_name = azurerm_virtual_network.myNet.name
  address_prefixes     = ["10.0.1.0/24"]
}

# Create NIC
resource "azurerm_network_interface" "myNic1" {
  name            = "vmnic1"
  location        = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  ip_configuration {
    name                       = "myipconfiguration1"
    subnet_id                 = azurerm_subnet.mySubnet.id
    private_ip_address_allocation = "Static"
    private_ip_address        = "10.0.1.10"
    public_ip_address_id      = azurerm_public_ip.myPublicIp1.id
  }

  tags = {
    environment = "CP2"
  }
}

# IP pública
resource "azurerm_public_ip" "myPublicIp1" {
  name            = "vmip1"
  location        = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name
  allocation_method = "Dynamic"
  sku              = "Basic"

  tags = {
    environment = "CP2"
  }
}
```

- Creamos una [IP pública](#) para poder acceder desde fuera de Azure.

# Terraform plan (security.tf)

- ▶ Creamos un [Security Group](#)
- ▶ En el Security Group definimos el tráfico que vamos a autorizar incluyendo [Security Rules](#).
- ▶ [Asociamos](#) el Security Group con una NIC para definir el tráfico que se permitirá en la NIC.

```
# Security group
# https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/network_security_group

resource "azurerm_network_security_group" "mySecGroup" {
  name                = "sshtraffic"
  location            = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  security_rule {
    name                = "SSH"
    priority            = 1001
    direction          = "Inbound"
    access              = "Allow"
    protocol            = "Tcp"
    source_port_range   = "*"
    destination_port_range = "22"
    source_address_prefix = "*"
    destination_address_prefix = "*"
  }

  tags = {
    environment = "CP2"
  }
}

# Vinculamos el security group al interface de red
# https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/network_interface_security_group_association

resource "azurerm_network_interface_security_group_association" "mySecGroupAssociation1" {
  network_interface_id = azurerm_network_interface.myNic1.id
  network_security_group_id = azurerm_network_security_group.mySecGroup.id
}
```

# Terraform plan (vm.tf)

```
Creamos una máquina virtual
# https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/linux_virtual_machine

resource "azurerm_linux_virtual_machine" "myVM1" {
  name                        = "my-first-azure-vm"
  resource_group_name        = azurerm_resource_group.rg.name
  location                   = azurerm_resource_group.rg.location
  size                       = var.vm_size
  admin_username              = "adminUsername"
  network_interface_ids      = [ azurerm_network_interface.myNic1.id ]
  disable_password_authentication = true

  admin_ssh_key {
    username   = "adminUsername"
    public_key = file("~/ssh/id_rsa.pub")
  }

  os_disk {
    caching              = "ReadWrite"
    storage_account_type = "Standard_LRS"
  }

  plan {
    name      = "centos-8-stream-free"
    product   = "centos-8-stream-free"
    publisher = "cognosys"
  }

  source_image_reference {
    publisher = "cognosys"
    offer     = "centos-8-stream-free"
    sku       = "centos-8-stream-free"
    version   = "1.2019.0810"
  }

  boot_diagnostics {
    storage_account_uri = azurerm_storage_account.stAccount.primary_blob_endpoint
  }

  tags = {
    environment = "CP2"
  }
}
```

- ▶ Creamos una [VM](#).
- ▶ Definimos el tamaño.
- ▶ Le asignamos la NIC que habíamos creado.
- ▶ Indicamos el usuario administrador.
- ▶ Especificamos la clave pública para el usuario administrador.
- ▶ Utilizaremos el usuario especificado y la clave privada asociada a la pública para acceder a la VM.
- ▶ Definimos el tipo de disco y la [replicación](#) (Standard\_LRS).



# Terraform plan (vm.tf)

- ▶ Standard\_LRS o Locally Redundant Storage.
- ▶ Cuando definamos images del marketplace tendremos que definir **plan** y **source\_image\_reference** con los datos de la imagen que utilizemos.
- ▶ Definimos la storage account a utilizar para almacenar información de troubleshooting.

```
Creemos una máquina virtual
# https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/linux_virtual_machine

resource "azurerm_linux_virtual_machine" "myVM1" {
  name                        = "my-first-azure-vm"
  resource_group_name        = azurerm_resource_group.rg.name
  location                   = azurerm_resource_group.rg.location
  size                       = var.vm_size
  admin_username             = "adminUsername"
  network_interface_ids      = [ azurerm_network_interface.myNic1.id ]
  disable_password_authentication = true

  admin_ssh_key {
    username   = "adminUsername"
    public_key = file("~/ssh/id_rsa.pub")
  }

  os_disk {
    caching              = "ReadWrite"
    storage_account_type = "Standard_LRS"
  }

  plan {
    name           = "centos-8-stream-free"
    product        = "centos-8-stream-free"
    publisher      = "cognosys"
  }

  source_image_reference {
    publisher = "cognosys"
    offer     = "centos-8-stream-free"
    sku       = "centos-8-stream-free"
    version   = "1.2019.0810"
  }

  boot_diagnostics {
    storage_account_uri = azurerm_storage_account.stAccount.primary_blob_endpoint
  }

  tags = {
    environment = "CP2"
  }
}
```

# Desplegando

```
C:\Terraform> terraform init
...
C:\Terraform> terraform plan
...
C:\Terraform> terraform apply
...
```

# Desplegando varias VMs

- ▶ Para desplegar varias VMs tendremos que incluir los recursos asociados a una VM para cada VM.
- ▶ Una NIC por cada VM.
- ▶ Una IP pública por cada VM que queramos que sea accesible desde fuera de Azure.
- ▶ Un security group por cada VM.
- ▶ Una definición de VM por cada VM.

# Desplegando varias VMs (la forma fácil)

- ▶ La forma fácil es duplicar el código tal cual está cambiando los nombres de los recursos.
- ▶ Si queremos añadir una máquina nueva, duplicamos el código.

# Desplegando varias VMs (la forma eficiente)

- ▶ Utilizando **count**.
- ▶ Definimos una nueva variable en el fichero **vars.tf**:

```
variable "location" {  
  type = string  
  description = "Región de Azure donde crearemos la infraestructura"  
  default = "West Europe"  
}  
  
variable "vm_size" {  
  type = string  
  description = "Tamaño de la máquina virtual"  
  default = "Standard_D1_v2" # 3.5 GB, 1 CPU  
}  
  
variable "vms" {  
  description = "Máquinas virtuales a crear"  
  type = list(string)  
  default = ["master", "worker01", "worker02", "nfs"]  
}
```

- ▶ En este ejemplo creamos una lista con los nombres que queremos utilizar para las VMs.

# Desplegando varias VMs (la forma eficiente)

- ▶ Como las máquinas son iguales en prácticamente todo, menos el nombre, el uso de **count** es la forma más sencilla para crearlas.
- ▶ Como cada máquina tendrá asignada una NIC en el fichero network.tf tendremos que reescribir el recurso que crea las NICs usando **count**:

```
# Create NIC
# https://registry.terraform.io/providers/hashicorp/azurerm/latest/docs/resources/network_interface

resource "azurerm_network_interface" "myNic" {
  name      = "nic-${var.vms[count.index]}"
  count     = length(var.vms)
  location  = azurerm_resource_group.rg.location
  resource_group_name = azurerm_resource_group.rg.name

  ip_configuration {
    name                  = "ipconf-${var.vms[count.index]}"
    subnet_id            = azurerm_subnet.mySubnet.id
    private_ip_address_allocation = "Static"
    private_ip_address    = "10.0.1.${count.index + 10}"
    public_ip_address_id  = azurerm_public_ip.myPublicIp[count.index].id
  }

  tags = {
    environment = "CP2"
  }
}
```

# Desplegando varias VMs (la forma eficiente)

- ▶ El resto de recursos necesarios para desplegar una VM se deberán reescribir de igual manera.
- ▶ IP pública.
- ▶ Security Group.
- ▶ VM.
- ▶ Una vez reescrito el código si quisiéramos desplegar n máquinas bastaría con incluir n nombres en los valores de la variable vms que hemos definido.
- ▶ No habría que modificar nada más. Eso sí todas las VMs serían iguales.

# Recursos adicionales

- ▶ Documentación [Open Source de Azure](#).
- ▶ Creación de [VMs Linux en Azure](#).
- ▶ Creación de [imágenes personalizadas en Azure](#).
- ▶ [Regiones de Azure](#).
- ▶ [Bandwidth pricing details](#).
- ▶ [Azure products available by region](#).
- ▶ Tipo de [VMs disponibles en Azure](#).
- ▶ [Tamaño de VMs en Azure](#).
- ▶ [Network Security Groups en Azure](#).





[www.unir.net](http://www.unir.net)