

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

Actividad individual. Despliegue de Nginx y Nodejs mediante una imagen generada por Packer

Las tareas que realizar son:

- ▶ **Creación de template packer.** Crear una template (plantilla) de Packer que te permita generar una imagen con una aplicación con Node.js ya instalada y configurada con Nginx como servidor web basada en el link de apoyo proporcionado anteriormente. Se tendrá en cuenta el uso de IP pública.
- ▶ **Despliegue sin intervención manual.** Despliegue desde la imagen de la Tarea 1 y del template implementado, modifíquelo de tal forma que se cree la imagen y realice el despliegue completo sin intervención manual. Consideración: revisar documentación del proveedor para la utilización del CLI.
- ▶ **Despliegue multinube.** Añade un builder de otro tipo para una nube pública (secundario AZURE) en caso de usar otro proveedor y justificar su uso.

El objetivo del proyecto es **crear y desplegar de forma automática una imagen de un sistema web completo en un entorno multicloud de Azure y AWS**. Este sistema web se compone de una pequeña aplicación escrita con Nodejs y un servidor web Nginx. Para conseguirlo, utilizo las tecnologías de Terraform, Ansible y Packer principalmente.

- ▶ **Tecnologías utilizadas:**

Terraform: Con Terraform centralizo todo la ejecución del proceso y despliego la infraestructura necesaria para levantar una instancia en la nube creada a partir de la imagen del sistema y accesible a través de internet.

Packer: Con Packer construyo la imagen del sistema completo. Packer utiliza como proveedor para la creación de la imagen la cloud. Levanta una instancia y toda la

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

infraestructura necesaria para la creación de la imagen y cuando termina las destruye.

Ansible: con ansible se lleva a cabo el aprovisionamiento de la instancia que levanta packer y a partir del cual se crea la imagen. En el caso de Azure este aprovisionamiento lo hago con Ansible, en el caso de AWS hago lo mismo pero directamente con scripting de Bash.

Para controlar el despliegue multicloud, se ha implementado un parámetro que se debe pasar al `terraform apply "deployment_target=""`, indicando si se quiere desplegar en las dos nubes simultaneamente o en una única nube. Si es este caso, hay que indicar en cual se desea desplegar.

► Proceso de creación y despliegue:

La secuencia de pasos del proceso sería la siguiente:

1. Inicializar ejecutando manualmente en la shell un `terraform init && terraform apply`
 2. Tras ello, terraform ejecuta el comando `packer build` que se encarga de levantar toda la infraestructura necesaria y la máquina que utilizó para la creación de la imagen. En el caso de Azure, en esta máquina se instala un Ansible y este se autoaprovisiona ejecutando un playbook y una serie de task definidos en él. En el caso de AWS, se ejecutan los mismos pasos pero en vez de con un Ansible directamente de forma manual con un scripting en Bash. El aprovisionamiento se basa entre otras cosas en la instalación y gestión de los servicios: Nodejs, Nginx, pm2 y App.js en la instancia que crea la imagen.
- A. **Nodejs:** Proporciona un entorno con todo lo necesario para que la aplicación ejecute y funcione correctamente.

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

- B. **Nginx:** servidor web que se encargará de redirijir todo el tráfico a la aplicación y de reenviar sus respuestas. Muy importante su configuración para que cuando se despliegue la iamgen el servidor esté activo y correctamente configurado para que sirva a la app. Pasa el tráfico del puerto 80 al 3000 (donde escucha la app.js)
- C. **PM2:** gestor de procesos de Nodejs que sirve para asegurar que la app.js esté activa cuando se despliegue la imagen sin tener que hacer nada más. (especialmente delicada este paso).
- D. **App.js:** aplicación central y funcional de la imagen, es importante trasferir el código fuente de la app para que sea accesible por la instancia que crea la imagen.
3. Tras esto, Packer crea la imagen y destruye toda la infraestructura que ha necesitado levantar en el proveedor correspondiente de cloud.
 4. Terraform tras esperar que la creación de la imagen finalice correctamente, levanta toda la infraestructura necesaria (par de claves, grupo de seguridad, disco...) para levantar una instancia a partir de esta imagen.
 5. Una vez terminado el despliegue, está instancia es accesible a través de internet a raiz de la ip pública.

En definitiva, solo ejecutando un: `terraform init && terraform apply` despliegas todo un entorno web funcional y accesible desde internet en la nube pública de Azure y AWS. Y además creas una imagen reutilizable para poder desplegar más instancias idénticas a estas en un futuro de una forma mucho más rápida y seguro ante posibles errores humanos.

► **Contenido del repositorio y ficheros del proyecto:**

El repositorio de GitHub se compone de dos directorios principales con dos versiones distintas: `/version-2` y `/version-3.1`

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

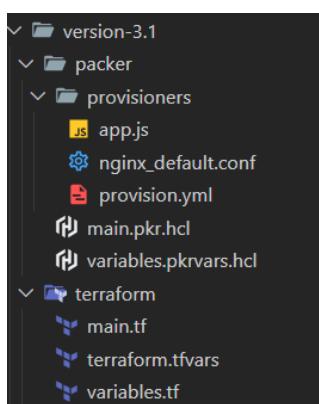
El directorio totalmente funcional y que contiene la ultima version del proyecto es el segundo (`/version-3.1`). Este es el directorio donde hay que ubicarse para desplegar el `terraform init && terraform apply` (`cd version-3.1/te*`).

Explicando brevemente el contenido del directorio:

- `/packer/`: directorio donde se encuentra todo el contenido necesario para la ejecución de Packer y para que pueda construir la imagen.
 - `/packer/main.pkr.hcl`: fichero principal de Packer donde se definen todos los recursos necesarios para contruir la imagen asi como defino todas las variables que van a utilizar.
 - `/packer/variables.pkrvars.hcl`: fichero donde les asigno valores a todos las variables definidas en el `main.pkr.hcl` menos a las credenciales de las dos nubes que por seguridad, las defino y asigno valores como variables de entorno de mi sistema operativo del host que utilizo para lanzar el terraform. Estos valores los paso como parámetros en el comando de `terraform apply` y `packer build`.
 - `/packer/providers/`: directorio donde podemos encontrar los ficheros auxiliares que sirven para crear la imagen como pueden ser, el fichero de configuración de apache (`nginx_default.conf`), el playbook que define el aprovisionamiento con ansible (`provision.yml`) y el código de la aplicación de nodejs (`app.js`).
- `/terraform/`: directorio donde se encuentra todo el contenido necesario para la ejecución de terraform y para que pueda desplegar toda la infraesturtura necesaria para el proyecto.
 - `/terraform/main.tf`: fichero principal de terraform, donde se define todo el flujo de proceso que debe seguir el despliegue y toda la infraestrutura a levantar.
 - `/terraform/variables.tf`: fichero donde se definen todas las variables utilizadas por terraform.

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

- `/terraform/terraform.tfvars`: fichero donde se les da valor a todas las variables menos a las credenciales de las dos nubes que por seguridad, las defino y asigno valores en variables de entorno de mi sistema operativo del host desde donde lanza el terraform. Estos valores los paso como parámetros en el comando de `terraform apply` y `packer build`.



► Contenido del Packer main:

El contenido de este fichero se puede diferenciar en varias partes en las cuales se definen los siguientes componentes necesarios para la creación de la imagen:

PLUGINS: Define los plugins necesarios para la plantilla.

Definición de variables: (no se les asigna valor aquí, solo alomejor el de por defecto)

BUILDER: Define cómo se construye la AMI en AWS --> `source{}` --> define el sistema base sobre el que quiero crear la imagen (ISO ubuntu) y el proveedor para el que creamos la imagen (tecnología con la que desplegará la imagen) --> AMAZON.

AZURE

PROVISIONERS: Configura el sistema operativo y la aplicación, como se va instalar y configurar el software --> `build{}`

► Vídeo de la experimentación: (en oculto)

<https://youtu.be/BhRB0716G5w>

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

Para la redacción de esta memoria se han ido capturando momentos claves durante la realización de todo el proyecto. Los apartados 1 y 2 de la memoria describen el proceso de obtención de los criterios 1 y 2, mientras que es el tercer apartado es el que describe el proceso final de **obtención de todos los objetivos del proyecto**.

1. Creación de template packer

Creación de template packer. Crear una template (plantilla) de Packer que te permita generar una imagen con una aplicación con Node.js ya instalada y configurada con Nginx como servidor web basada en el link de apoyo proporcionado anteriormente. Se tendrá en cuenta el uso de IP pública.

► **Instalo la herramienta de CLI de AWS en mi ordenador**

- Link: <https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>
- Clicko sobre el ejecutable

Historial de descargas recientes 

 AWSCLIV2.msi
40,6 MB • Hace 3 minutos

- Compruebo que se ha instalado correctamente

```
User> aws --version
aws-cli/2.22.12 Python/3.12.6 Windows/11 exe/AMD64
User> |
```

► **Inicio el laboratorio de la cuenta de estudiante de aws que nos proporciona la UNIR.**

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

► Accedo a la interfaz web de nuestra cuenta de estudiante de AWS

- Copio las credenciales de nuestro usuario. Tenemos 3 claves, `aws_access_key_id`, `aws_secret_access_key`, `aws_session_token`.

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

The screenshot shows the AWS Academy Learner Lab interface. On the left is a sidebar with navigation links: Account, Dashboard, Courses, Calendar, Inbox, History, and Help. The main area displays a session titled 'eee_i_3899903@runweb15e'. A 'Cloud Access' panel is open, showing AWS CLI configuration details. The AWS CLI section contains a command to copy and paste into a credentials file:

```
[default]
aws_access_key_id=ASIAYS2NMFU3
OYOAUFF
aws_secret_access_key=unrq21tgk
NgIgggRH6IV00AayGuibVFqf0zY75
aws_session_token=IQ0jB3jpZ2lW
2VjEGkaXvLx1d1c3QTHiJGNEQCIHv
Cfczy2y093R0DaC2gztzp4URlwMj
uvrMMUugA1Aumv1efgSVJCM9Blukj
cE1k3tJgdBaYchmxdmtXcQ/AggiF
AAaD0U59DE4NDzA2jhj105PMUTxquP
14Stjh4ekpwcSxtf03udnEtbtPo4Edf
EmmyyOT2eEUYZhjsyjSQdoT6C31/BMp6
```

► Configuro aws cli con las credenciales de nuestra cuenta.

- Introducimos las credenciales

```
User> notepad $HOME\.aws\credentials
User> |
```

- Reviso que se han configurado bien

```
Default output format [None]: json
User> aws configure list
  Name           Value      Type    Location
  ----
  profile        <not set>   None    None
  access_key     *****AUFF  shared-credentials-file
  secret_key     *****Y75J  shared-credentials-file
  region         us-east-1  config-file  ~/.aws/config
User> |
```

- Compruebo que las credenciales funcionan y sirven para autenticarnos con nuestra cuenta de aws.

```
User> aws sts get-caller-identity
{
  "UserId": "AROAYJS2NWFU3WYPS72GW4:user3693581=alejandro.ingles506@comunidadunir.net",
  "Account": "590184066359",
  "Arn": "arn:aws:sts::590184066359:assumed-role/voclabs/user3693581=alejandro.ingles506@comunidadunir.net"
}
```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

- ▶ Sin cambiar la región por defecto de nuestra cuenta de estudiante de aws (**us-east-1**) y como se recomienda en el enunciado de este ejercicio, escogería por ejemplo la imagen señalada en la captura ya que se trata de una imagen con soporte a hvm, ebs y de sistema base ubuntu 20.04LTS (como el tutorial proporcionado de guía). Sin embargo en mi plantilla, busco una que siga esas características sin indicar un modelo en específico.

Zone	Name	Version	Arch	Instance Type	Release	AMI-ID	AKI-ID
us-east-1	Xenial Xerus	16.04 LTS	amd64	hvm:ebs:ssd	20210928	ami-0b0ea68c435eb488d	hvm
us-east-1	Xenial Xerus	16.04 LTS	arm64	hvm:ebs:ssd	20210928	ami-04b3c39a8a1c62b76	hvm
us-east-1	Xenial Xerus	16.04 LTS	amd64	hvm:instance-store	20210928	ami-0059b7c09f6768050	hvm
us-east-1	Xenial Xerus	16.04 LTS	amd64	pv:ebs:ssd	20210928	ami-06ae0e97bcb59038c	hvm
us-east-1	Oracular Oriole	24.10	amd64	hvm:ebs:ssd-gp3	20241009	ami-0f67e16c79859897f	hvm
us-east-1	Oracular Oriole	24.10	arm64	hvm:ebs:ssd-gp3	20241009	ami-0ca974a65399b0985	hvm
us-east-1	Noble Numbat	24.04 LTS	amd64	hvm:ebs:ssd-gp3	20241109	ami-0e2c8caa4b6378d8c	hvm
us-east-1	Noble Numbat	24.04 LTS	arm64	hvm:ebs:ssd-gp3	20241109	ami-0932ff346ea84d48	hvm
us-east-1	Jammy Jellyfish	22.04 LTS	amd64	hvm:ebs:ssd	20241120	ami-04552bb4f4dd38925	hvm
us-east-1	Jammy Jellyfish	22.04 LTS	arm64	hvm:ebs:ssd	20241120	ami-05245affa1aea4cc6	hvm
us-east-1	Focal Fossa	20.04 LTS	amd64	hvm:ebs:ssd	20241112	ami-095ab8f574c0a0acd0	hvm
us-east-1	Focal Fossa	20.04 LTS	arm64	hvm:ebs:ssd	20241112	ami-02d0a5d34db77e33f	hvm
us-east-1	Bionic Beaver	18.04 LTS	amd64	hvm:ebs:ssd	20230531	ami-055744c75048d8296	hvm
us-east-1	Bionic Beaver	18.04 LTS	arm64	hvm:ebs:ssd	20230531	ami-0fea31578248bcd6c	hvm
us-east-1	Bionic Beaver	18.04 LTS	amd64	hvm:instance-store	20230531	ami-03025bb25a1de0fc2	hvm

Show 100 entries Search: us-east-1

OR click the ami ID, which will direct you to the AWS console

Note: Versions ending in EOL are end-of-life and are provided for reference only

Showing 1 to 15 of 15 entries (filtered from 441 total entries)

- ▶ Meto las credenciales de nuestra cuenta de aws como variables de entorno de nuestro usuario en el sistema de nuestro ordenador para no tenerlas directamente escritas en el código fuente.

```
[System.Environment]::SetEnvironmentVariable("PKR_VAR_AWS_ACCESS_KEY", "ASF", "User")
[System.Environment]::SetEnvironmentVariable("PKR_VAR_AWS_SECRET_KEY", "unrq21tg", "User")
[System.Environment]::SetEnvironmentVariable("PKR_VAR_AWS_SESSION_TOKEN", "IQoJb3JpZ2luX2ViJGME...", "User")
```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

```
User> echo ${env:PKR_VAR_aws_access_key}
ASIAYS2NWFU3YOYOAUFF
User> echo ${env:PKR_VAR_aws_access_key}
ASIAYS2NWFU3YOYOAUFF
User> echo ${env:PKR_VAR_aws_secret_key}
unrq21tgWNgiNggrH6Iv00AayGubRVffQToZY75J
User> echo ${env:PKR_VAR_aws_session_token}
IQoJb3JpZ2luX2VjEGkaCXVzLXdlc3QtMiJGMEQC1HVkCFcyZyD9B3RHxDa
hmxSdmItXCq/AggiEAAaDDU5MDE4NDA2NjM1OSIMU4Txqupl1+5tjH4eKpw
xrcIQGSfLX10wK8nIiYlrAmT9eBdM2uR6/Jy0uArqMKLHaRhnFj01pApstZ
6BV9nyRU7QyIOCh4B7/85sA9YY/gUT4rU8Aeg0tcSg1Vw01hnKDP7VzvAib
xk+TP1oQHCjmXaiRkLvfZTx7Nxq5HAiSx7YTKDCx/vVmAI39rLjyxD2nard
hT1oRZEf0KcPoQcEYOXOClkhpv1nPuoZZAyYoTgJzidYg4W77FjLrp6ar9x
78Tdt7zaYduFOmRtLRhE7wHxu/JanaQ0BCnCJlijaQ8QW/jALUeGm2Z0LKP
User> |
```

- ▶ Valido la plantilla pasandole las credenciales de la cuenta de aws como parametros.

```
PROBLEMS OUTPUT TERMINAL PORTS COMMENTS
● ACT1_Nginx-Nodejs_Packer> packer validate -var "aws_access_key=${env:PKR_VAR_aws_access_key}" -var "aws_secret_key=${env:PKR_VAR_aws_secret_key}" -var "aws_session_token=${env:PKR_VAR_aws_session_token}" -var-file="variables/variables.pkrvars.hcl" main.pkr.hcl
The configuration is valid.
○ ACT1_Nginx-Nodejs_Packer> |
```

- ▶ Lanzo la creación de la imagen en el builder de aws. (paso las credenciales de mi cuenta de estudiante como parámetros del comando)

```
○ ACT1_Nginx-Nodejs_Packer> packer build -var "aws_access_key=${env:PKR_VAR_aws_access_key}" -var "aws_secret_key=${env:PKR_VAR_aws_secret_key}" -var "aws_session_token=${env:PKR_VAR_aws_session_token}" -var-file="variables/variables.pkrvars.hcl" main.pkr.hcl
aws-node-nginx.amazon-ebs.aws_builder: output will be in this color.

=> aws-node-nginx.amazon-ebs.aws_builder: Prevalidating any provided VPC information
=> aws-node-nginx.amazon-ebs.aws_builder: Prevalidating AMI Name: Actividad_Node_Nginx_AMI
aws-node-nginx.amazon-ebs.aws_builder: Found Image ID: ami-095a8f574cb0ac0dd
=> aws-node-nginx.amazon-ebs.aws_builder: Creating temporary keypair: packer_675abb30-f61b-a7b0-5236-6b7ea35bb486
=> aws-node-nginx.amazon-ebs.aws_builder: Creating temporary security group for this instance: packer_675abb36-23da-868a-dcd0-ad691fe50b81
=> aws-node-nginx.amazon-ebs.aws_builder: Authorizing access to port 22 from [0.0.0.0/0] in the temporary security groups...
=> aws-node-nginx.amazon-ebs.aws_builder: Launching a source AWS instance...
aws-node-nginx.amazon-ebs.aws_builder: Instance ID: i-01e8751e6cc0172a91
=> aws-node-nginx.amazon-ebs.aws_builder: Waiting for instance (i-01e8751e6cc0172a91) to become ready...
=> aws-node-nginx.amazon-ebs.aws_builder: Using SSH communicator to connect: 54.163.190.43
=> aws-node-nginx.amazon-ebs.aws_builder: Waiting for SSH to become available...
=> aws-node-nginx.amazon-ebs.aws_builder: Connected to SSH!
=> aws-node-nginx.amazon-ebs.aws_builder: Provisioning with shell script: c:\Users\user\AppData\Local\Temp\packer-shell456845790
=> aws-node-nginx.amazon-ebs.aws_builder:
=> aws-node-nginx.amazon-ebs.aws_builder: WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
=> aws-node-nginx.amazon-ebs.aws_builder:
aws-node-nginx.amazon-ebs.aws_builder: Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
aws-node-nginx.amazon-ebs.aws_builder: Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
aws-node-nginx.amazon-ebs.aws_builder: Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
aws-node-nginx.amazon-ebs.aws_builder: Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
aws-node-nginx.amazon-ebs.aws_builder: Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
```

- ▶ Me conecto a la interfaz web de aws y después al servicio de EC2 donde se puede apreciar que está corriendo una instancia nueva.

- Una vez creada la imagen, se termina la instancia de EC2 (usada solo para la creación de la imagen).

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

```

--> aws-node-nginx.amazon-ebs.aws_builder: 0 0 0 0 0 0 0 0 --::--- --::--- --::--- 0
--> aws-node-nginx.amazon-ebs.aws_builder: Stopping the source instance...
aws-node-nginx.amazon-ebs.aws_builder: Stopping instance
--> aws-node-nginx.amazon-ebs.aws_builder: 0 0 0 0 0 0 0 0 --::--- --::--- --::--- 0
--> aws-node-nginx.amazon-ebs.aws_builder: Stopping the source instance...
aws-node-nginx.amazon-ebs.aws_builder: Stopping instance
--> aws-node-nginx.amazon-ebs.aws_builder: Stopping the source instance...
aws-node-nginx.amazon-ebs.aws_builder: Stopping instance
--> aws-node-nginx.amazon-ebs.aws_builder: Waiting for the instance to stop...
--> aws-node-nginx.amazon-ebs.aws_builder: Creating AMI Actividad_Node_Nginx_AMI_5 from instance i-09dc4e37b8b34dc72
aws-node-nginx.amazon-ebs.aws_builder: AMI: ami-0d6de2b3a371a757f
aws-node-nginx.amazon-ebs.aws_builder: AMI: ami-0d6de2b3a371a757f
--> aws-node-nginx.amazon-ebs.aws_builder: Waiting for AMI to become ready...
--> aws-node-nginx.amazon-ebs.aws_builder: Waiting for AMI to become ready...
--> aws-node-nginx.amazon-ebs.aws_builder: Skipping Enable AMI deprecation...
--> aws-node-nginx.amazon-ebs.aws_builder: Terminating the source AWS instance...
--> aws-node-nginx.amazon-ebs.aws_builder: Skipping Enable AMI deprecation...
--> aws-node-nginx.amazon-ebs.aws_builder: Terminating the source AWS instance...
--> aws-node-nginx.amazon-ebs.aws_builder: Cleaning up any extra volumes...
--> aws-node-nginx.amazon-ebs.aws_builder: Terminating the source AWS instance...
--> aws-node-nginx.amazon-ebs.aws_builder: Cleaning up any extra volumes...
--> aws-node-nginx.amazon-ebs.aws_builder: Cleaning up any extra volumes...
--> aws-node-nginx.amazon-ebs.aws_builder: No volumes to clean up, skipping
--> aws-node-nginx.amazon-ebs.aws_builder: Deleting temporary security group...
--> aws-node-nginx.amazon-ebs.aws_builder: Deleting temporary keypair...
Build 'aws-node-nginx.amazon-ebs.aws_builder' finished after 6 minutes 30 seconds.

--> Wait completed after 6 minutes 30 seconds

--> Builds finished. The artifacts of successful builds are:
--> aws-node-nginx.amazon-ebs.aws_builder: AMIs were created:
us-east-1: ami-0d6de2b3a371a757f

ACT1_Nginx-Nodejs_Packer> []

```

- ▶ Voy a la sección de AMIs del servicio de EC2 para ver que se ha creado la imagen correctamente

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

The screenshot shows the AWS EC2 console under the 'Images' section. It displays a list of available Amazon Machine Images (AMIs). One specific AMI, 'Actividad_Node_Nginx_AMI_5', is selected and highlighted with a blue border. The details for this AMI are shown in a modal window below:

AMI ID	Image type	Platform details	Root device type
ami-0d6de2b3a371a757f	machine	Linux/UNIX	EBS
AMI name	Owner account ID	Architecture	Usage operation
Actividad_Node_Nginx_AMI_5	999999999999	i386	On demand

► Lanzo la instancia usando como base la imagen creada anteriormente (Launch instances) y la configuro.

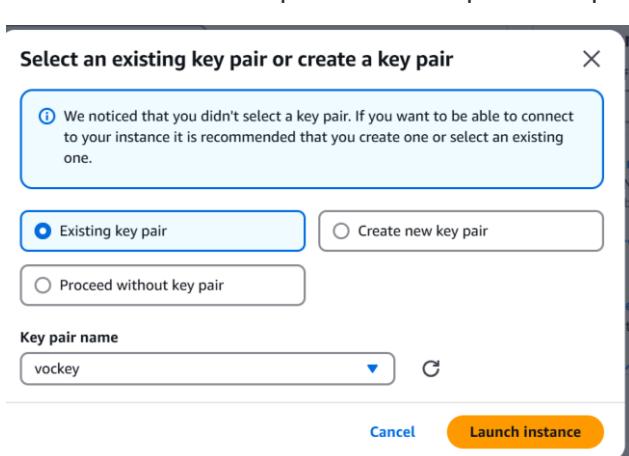
The screenshot shows the AWS EC2 console under the 'Instances' section. It displays a list of existing instances. All listed instances are in a 'Terminated' state. The table includes columns for Instance ID, Instance state, Instance type, Status check, and Alarm status:

Instance ID	Instance state	Instance type	Status check	Alarm status
i-0221337e236877393	Terminated	t2.micro	-	View alarms
i-0fb73e20462c18503	Terminated	t2.micro	-	View alarms
i-088adf5db6fc23e0	Terminated	t2.micro	-	View alarms
i-09dc4e37b8b34dc72	Terminated	t2.micro	-	View alarms

- Habilito el acceso por ssh, https, http, elijo una instancia gratuita de t2.micro...

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

- Selecciono el par de claves que viene por defecto



- Verifico que ya está corriendo la instancia

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Prueba1_Imagen	i-070285dd599693a98	Running	t2.micro	Initializing	View alarms
	i-0fh7z020167r1850z	Terminated	t2.micro	-	View alarms

- Me conecto a la consola de la instancia e intento acceder por su ip pública (44.212.9.122)

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

Connect to your instance i-070285dd599693a98 (Prueba1_Imagen_Packer) using any of these options

EC2 Instance Connect **Session Manager** **SSH client** **EC2 serial console**

Instance ID
i-070285dd599693a98 (Prueba1_Imagen_Packer)

Connection Type

- Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.
- Public IPv4 address
44.212.9.122
- IPv6 address

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, root.

root

- Acceso a la consola de la instancia

```
7 updates can be applied immediately.
6 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Inable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

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.

root@ip-172-31-88-9:~# 
```

i-070285dd599693a98 (Prueba1_Imagen_Packer)
PublicIPs: 44.212.9.122 PrivateIPs: 172.31.88.9

ERROR IDENTIFICADO en la primera vez lanzando la instancia.

No es seguro 44.212.9.122

502 Bad Gateway

nginx/1.18.0 (Ubuntu)

Al intentar conectarme por el navegador, me salta un error 502 Gateway. Lo que quiere decir que nginx no es capaz de acceder al servicio en cuestión.

Tras comprobar una serie de comandos en la consola de la instancia. Deduzco que el error no está en la configuración de nginx, está en que al desplegar la imagen el

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

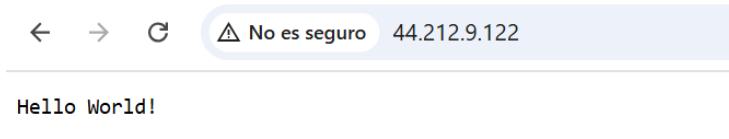
servicio de la app.js empieza sin estar arrancado (al arrancarlo manualmente todo va bien).

- Vemos que inicialmente el servicio está inactivo y lo tengo que conectar manualmente

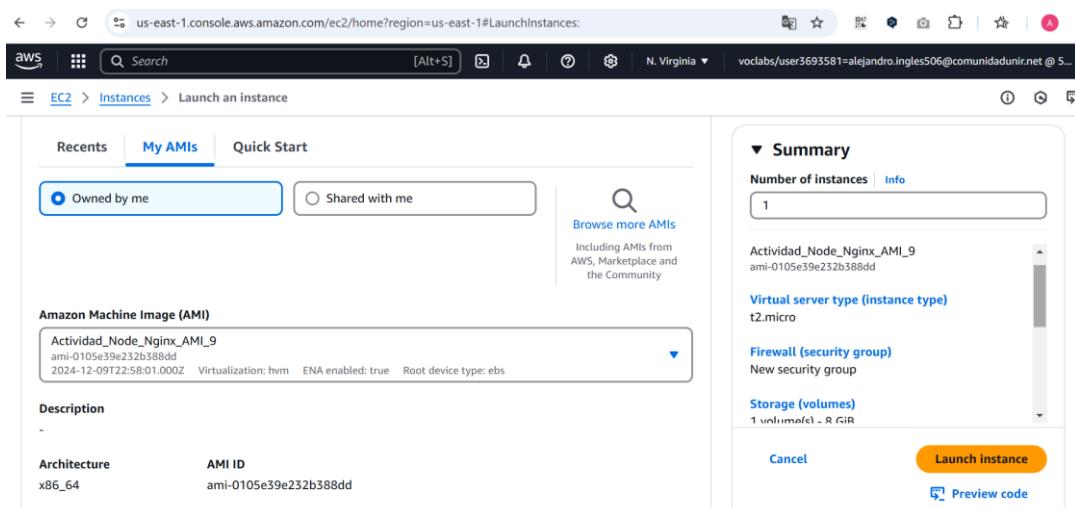
```
root@ip-172-31-88-9:~# pm2 list
[PM2] Spawning PM2 daemon with pm2_home=/root/.pm2
[PM2] PM2 Successfully daemonized
[PM2] [WARN] Current process list is not synchronized with saved list. App app differs. Type 'pm2 save' to synchronize.
root@ip-172-31-88-9:~# pm2 start /home/ubuntu/app.js
[PM2] Starting /home/ubuntu/app.js in fork_mode (1 instance)
[PM2] Done.

[ id | name | namespace | version | mode | pid | uptime | ⚡ | status | cpu | mem | user | watching ]
+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 | app | default | N/A | fork | 1315 | 0s | 0 | online | 0% | 25.3mb | root | disabled |
+---+---+---+---+---+---+---+---+---+---+---+---+
root@ip-172-31-88-9:~# curl http://localhost:3000
Hello World!
```

- Al iniciarla manualmente puedo acceder desde su ip pública.



► **SOLUCIÓN:** Modificando el provisioner que gestionaba el proceso de pm2 he conseguido que al iniciar la instancia con esta imagen el servicio app.js esté activo directamente como se puede ver en la captura de la consola que directamente con un pm2 list sale el proceso activo.



Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

The screenshot shows the AWS Cloud9 IDE interface. At the top, there's a navigation bar with tabs like 'Launch', 'Launch', 'Instance', and 'EC2 Inst'. Below the navigation bar, there's a message: 'Thanks for your response! Please leave us feedback if you want to say more.' The main area is a terminal window displaying the following text:

```

ee https://ubuntu.com/esm or run: sudo pro status
ew release '22.04.5 LTS' available.
un 'do-release-upgrade' to upgrade to it.

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

buntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
pplicable law.

oot@ip-172-31-89-10:~# pm2 list

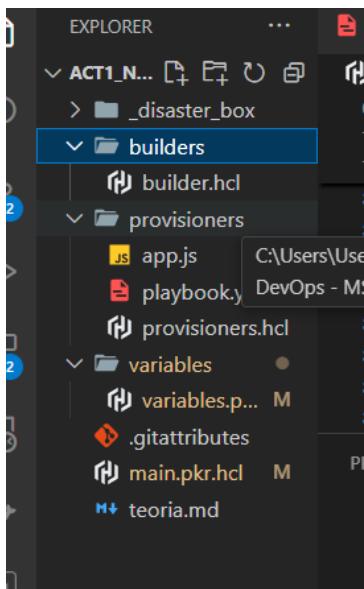
```

id	name	namespace	version	mode	pid	uptime	o	status	cpu	mem	user	watching
0	app	default	N/A	Fork	799	23s	0	online	0%	38.9mb	root	disabled

oot@ip-172-31-89-10:~#

- ▶ Estructura de archivos en el momento de la experimentación tras conseguir este logro (falta el providers/nginx_default.conf).

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024



2. Despliegue sin intervención manual

Despliegue sin intervención manual. Despliegue desde la imagen de la Tarea 1 y del template implementado, modifíquelo de tal forma que se cree la imagen y realice el despliegue completo sin intervención manual. Consideración: revisar documentación del proveedor para la utilización del CLI.

La idea es levantar todo la infraestructura y centralizar todo el proceso con terraform. Será en este terraform donde se ejecutará el packer build y tras esperar a que la imagen se cree correctamente el terraform levantará la instancia correspondiente a partir de esta.

► **Instalo terraform**

- Link: <https://developer.hashicorp.com/terraform/install>

```
User> terraform --version
Terraform v1.10.1
on windows_amd64
User> |
```

- Ejecuto el comando terraform init para inicializar el entorno de terraform.

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

```

PROBLEMS OUTPUT TERMINAL PORTS COMMENTS

● terraform> terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Finding latest version of hashicorp/null...
- Installing hashicorp/aws v5.80.0...
- Installed hashicorp/aws v5.80.0 (signed by HashiCorp)
- Installing hashicorp/null v3.2.3...
- Installed hashicorp/null v3.2.3 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.

○ terraform>

```

- Compruebo la plantilla terraform (le paso los valores de las credenciales como parámetros del comando para evitar definirlas en el código fuente).

```

● terraform> terraform plan -var "aws_access_key=$env:PKR_VAR_aws_access_key" -var "aws_secret_key=$env:PKR_VAR_aws_secret_key" -var "aws_session_token=$env:PKR_VAR_aws_session_token"
data.aws_vpc.default: Reading...
data.aws_ami.latest_ami: Reading...
data.aws_ami.latest_ami: Read complete after 1s [id=ami-0105e39e232b388dd]
data.aws_vpc.default: Read complete after 3s [id=vpc-053265629dc400380]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.web_server will be created
+ resource "aws_instance" "web_server" {
    + ami                               = "ami-0105e39e232b388dd"
    + arn                             = (known after apply)
    + associate_public_ip_address      = (known after apply)
    + availability_zone                = (known after apply)
    + cpu.core_count                   = (known after apply)
    + cpu.threads_per_core            = (known after apply)
    + disable_api_stop                = (known after apply)
    + disable_api_termination         = (known after apply)
    + ebs.optimized                   = (known after apply)
    + get_password_data              = false
    + host_id                         = (known after apply)
    + host_resource_group.arn          = (known after apply)
}

```

- Despliego el sistema descrito por terraform que levantará en primer lugar con packer una instancia en aws para crear la imagen y tras ello levantará otra instancia de aws que usará como base esta imagen.

```

● terraform> terraform apply -var "aws_access_key=$env:PKR_VAR_aws_access_key" -var "aws_secret_key=$env:PKR_VAR_aws_secret_key" -var "aws_session_token=$env:PKR_VAR_aws_session_token"
data.aws_vpc.default: Reading...
data.aws_ami.latest_ami: Reading...
data.aws_ami.latest_ami: Read complete after 1s [id=ami-0105e39e232b388dd]
data.aws_vpc.default: Read complete after 3s [id=vpc-053265629dc400380]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

- Imagen creada a partir de la primera instancia levantada. Terraform espera a que se termine de crear la AMI para levantar la segunda instancia que usa esta.

```
[5] 0:~/Desktop/Actividad_Node_Nginx_AMI_9$ terraform apply
INFO: A temporary file was created for this run. This file will be deleted when the session exits.
null_resource.packer_ami: Still creating... [4m0s elapsed]
null_resource.packer_ami: Still creating... [4m10s elapsed]
null_resource.packer_ami: Still creating... [4m20s elapsed]
null_resource.packer_ami: Still creating... [4m30s elapsed]
null_resource.packer_ami: Still creating... [4m40s elapsed]
null_resource.packer_ami: Still creating... [4m50s elapsed]
null_resource.packer_ami: Still creating... [5m0s elapsed]
null_resource.packer_ami: Still creating... [5m10s elapsed]
null_resource.packer_ami: Still creating... [5m20s elapsed]
null_resource.packer_ami: Still creating... [5m30s elapsed]
null_resource.packer_ami: Still creating... [5m40s elapsed]
```

Name	AMI ID	Source	Owner
Actividad_Node_Nginx_AMI_9	ami-0c576c1aeb5984984	590184066359/Actividad_Node_Nginx...	590184...

```
null_resource.packer_ami: Still creating... [5m30s elapsed]
null_resource.packer_ami: Still creating... [5m40s elapsed]
null_resource.packer_ami: Still creating... [5m50s elapsed]
null_resource.packer_ami: Still creating... [6m0s elapsed]
null_resource.packer_ami: Still creating... [6m10s elapsed]
null_resource.packer_ami: Still creating... [6m20s elapsed]
null_resource.packer_ami: Still creating... [6m30s elapsed]
null_resource.packer_ami: Still creating... [6m40s elapsed]
null_resource.packer_ami: Still creating... [6m50s elapsed]
null_resource.packer_ami (local-exec): ==> aws-node-nginx.amazon-ebs.aws_builder: Skipping Enable AMI depreciation...
null_resource.packer_ami (local-exec): ==> aws-node-nginx.amazon-ebs.aws_builder: Terminating the source AWS instance...
null_resource.packer_ami: Still creating... [7m0s elapsed]
null_resource.packer_ami: Still creating... [7m10s elapsed]
null_resource.packer_ami (local-exec): ==> aws-node-nginx.amazon-ebs.aws_builder: Cleaning up any extra volumes...
null_resource.packer_ami (local-exec): ==> aws-node-nginx.amazon-ebs.aws_builder: No volumes to clean up, skipping
null_resource.packer_ami (local-exec): ==> aws-node-nginx.amazon-ebs.aws_builder: Deleting temporary security group...
null_resource.packer_ami: Still creating... [7m20s elapsed]
null_resource.packer_ami (local-exec): ==> aws-node-nginx.amazon-ebs.aws_builder: Deleting temporary keypair...
null_resource.packer_ami (local-exec): Build 'aws-node-nginx.amazon-ebs.aws_builder' finished after 7 minutes 380 milliseconds.

null_resource.packer_ami (local-exec): ==> Wait completed after 7 minutes 381 milliseconds

null_resource.packer_ami (local-exec): ==> Builds finished. The artifacts of successful builds are:
null_resource.packer_ami (local-exec): --> aws-node-nginx.amazon-ebs.aws_builder: AMIs were created:
null_resource.packer_ami (local-exec): us-east-1: ami-0c576c1aeb5984984

null_resource.packer_ami: Creation complete after 7m3s [id=2972218884354216918]
data.aws_amilatest_ami: Reading...
data.aws_amilatest_ami: Read complete after 2s [id=ami-0c576c1aeb5984984]
aws_instance.web_server: Creating...
```

- Segunda instancia levantada. Como salida exitosa nos imprime en la terminal el id de la instancia y la ip pública a partir de la cual podemos acceder al servicio.

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

```
File Edit Selection View ... < > ACT1_Nginx-Nodejs_Packer

EXPLORER ... PROBLEMS OUTPUT TERMINAL PORTS COMMENTS
ACT1_NGINX-NODEJS.PACKER null_resource.packer_ami (local-exec): -> aws-node-nginx.amazon-ebs.aws_builder: AMIs were created;
null_resource.packer_ami (local-exec): -> ami-east-1: ami-0c576caeb5984984

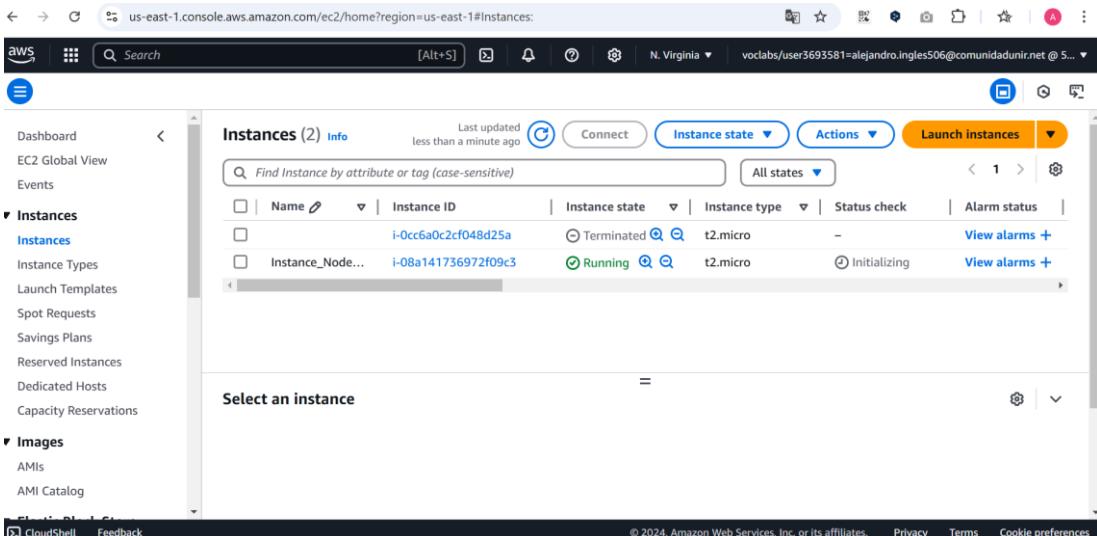
null_resource.packer_ami: Creation complete after 7ms [id=297221884354216918]
data.aws_amazon_latest_am: Reading...
data.aws_amazon_latest_am: Read complete after 2s [id=ami-0c576caeb5984984]

aws_instance.web_server: Still creating... [10s elapsed]
aws_instance.web_server: Provisioning with 'remote-exec'...
aws_instance.web_server (remote-exec): Connecting to remote host via SSH...
aws_instance.web_server (remote-exec): Host: 98.81.81.253
aws_instance.web_server (remote-exec): User: ubuntu
aws_instance.web_server (remote-exec): Password: false
aws_instance.web_server (remote-exec): Private key: true
aws_instance.web_server (remote-exec): Certificate: false
aws_instance.web_server (remote-exec): SSH Agent: false
aws_instance.web_server (remote-exec): Checking Host Key: false
aws_instance.web_server (remote-exec): Target Platform: unix
aws_instance.web_server (remote-exec): Still creating... [20s elapsed]
aws_instance.web_server: Still creating... [20s elapsed]
aws_instance.web_server (remote-exec): Connecting to remote host via SSH...
aws_instance.web_server (remote-exec): Host: 98.81.81.253
aws_instance.web_server (remote-exec): User: ubuntu
aws_instance.web_server (remote-exec): Password: false
aws_instance.web_server (remote-exec): Private key: true
aws_instance.web_server (remote-exec): Certificate: false
aws_instance.web_server (remote-exec): SSH Agent: false
aws_instance.web_server (remote-exec): Checking Host Key: false
aws_instance.web_server (remote-exec): Target Platform: unix
aws_instance.web_server (remote-exec): connected!
aws_instance.web_server (remote-exec): (la instancia esta configurada correctamente.)
aws_instance.web_server: Creation complete after 38s [id=08a141736972f09c3]

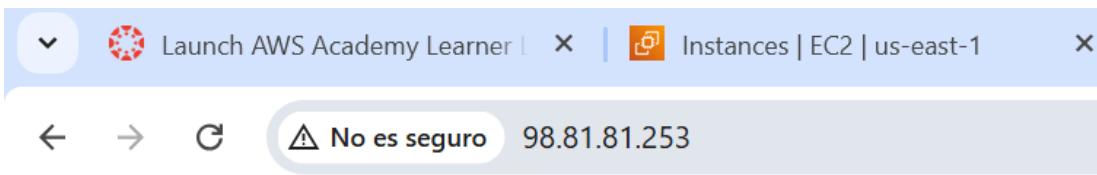
Apply complete! Resources: 2 added, 0 changed, 1 destroyed.

Outputs:
instance_id = "i-08a141736972f09c3"
public_ip = "98.81.81.253"
terraform = []

DEBUG CONSOLE
Filter (e.g. text, ...)
```



► Acceso exitoso al servicio



Hello World!

▶ Acceso a la instancia a través de una conexión ssh

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

```

instance_id = "i-08a141736972f09c3"
public_ip = "98.81.81.253"
○ terraform> ssh -i C:\Users\User\.aws\unir.pem ubuntu@98.81.81.253
The authenticity of host '98.81.81.253 (98.81.81.253)' can't be established.
ED25519 key fingerprint is SHA256:VrqImyVb0aE4S4ht9coFstUGI9XAxUwxE0McC5RvnFE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '98.81.81.253' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1072-aws x86_64)

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

System information as of Tue Dec 10 23:34:46 UTC 2024

System load: 0.08 Processes: 109
Usage of /: 28.9% of 7.57GB Users logged in: 0
Memory usage: 24% IPv4 address for eth0: 172.31.20.130
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

18 updates can be applied immediately.
17 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Tue Dec 10 23:31:34 2024 from 116.255.16.57
ubuntu@ip-172-31-20-130:~$ pm2 list

```

- Se puede ver como efectivamente el proceso de app.js está activo y accesible

```

ubuntu@ip-172-31-20-130:~$ sudo pm2 list

```

id	name	namespace	version	mode	pid	uptime	ø	status	cpu	mem	user	watching
0	app	default	N/A	fork	910	5m	0	online	0%	40.0mb	root	disabled

```

ubuntu@ip-172-31-20-130:~$ exit
logout
Connection to 98.81.81.253 closed.

```

► Destrucción de infraestructura desplegada tras la experimentación a través del terraform destroy

```

$ terraform> terraform destroy -var "aws_access_key=$env:PKR_VAR_aws_access_key" -var "aws_secret_key=$env:PKR_VAR_aws_secret_key" -var "aws_session_token"
null_resource.packer_ami: Refreshing state... [id=2972218884354216918]
data.aws_vpc.default: Reading...
data.aws_ami.latest_ami: Reading...
data.aws_ami.latest_ami: Read complete after 1s [id=ami-0c576c1aeb5984984]
data.aws_vpc.default: Read complete after 2s [id=vpc-053265629dc400380]
data.aws_security_group.existing_sg: Reading...
data.aws_security_group.existing_sg: Read complete after 1s [id=sg-0e913990fd252976b]
aws_instance.web_server: Refreshing state... [id=i-08a141736972f09c3]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
- destroy

Terraform will perform the following actions:

# aws_instance.web_server will be destroyed
- resource "aws_instance" "web_server" {
  - ami
  - arm
  - associate_public_ip_address
  - availability_zone
  - cpu_core_count
  - cpu_threads_per_core
  - disable_api_stop
  - disable_api_termination
  - ebs_optimized
  - enable_performance_data
}

```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

```

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

aws_instance.web_server: Destroying... [id=i-08a141736972f09c3]
aws_instance.web_server: Still destroying... [id=i-08a141736972f09c3, 10s elapsed]
aws_instance.web_server: Still destroying... [id=i-08a141736972f09c3, 20s elapsed]
aws_instance.web_server: Still destroying... [id=i-08a141736972f09c3, 30s elapsed]
aws_instance.web_server: Destruction complete after 33s
null_resource.packer_ami: Destroying... [id=2972218884354216918]
null_resource.packer_ami: Destruction complete after 0s

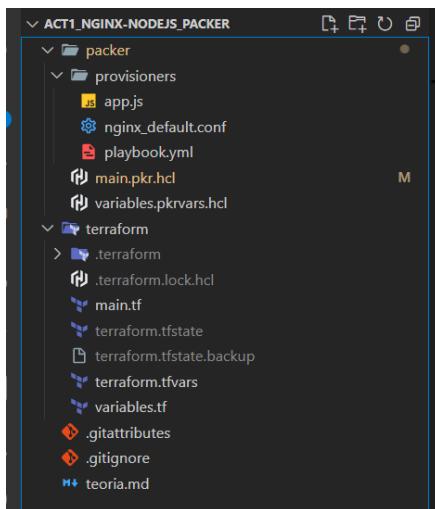
Destroy complete! Resources: 2 destroyed.
○ terraform>

```

The screenshot shows the AWS CloudWatch Metrics interface. On the left, there's a sidebar with navigation links: Dashboard, EC2 Global View, Events, Instances (selected), Instances, Instance Types, Launch Templates, Spot Requests, and Savings Plans. The main area is titled 'Instances (2) Info' and shows a table of two terminated EC2 instances. The table columns include Name, Instance ID, Instance state, Instance type, Status check, and Alarm status. Both instances are listed as terminated.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
i-0cc6a0c2cf048d25a	i-0cc6a0c2cf048d25a	Terminated	t2.micro	-	View alarms +
Instance_Node...	i-08a141736972f09c3	Terminated	t2.micro	-	View alarms +

- Estructura de archivos en el momento de la experimentación tras conseguir este logro.



Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

3. Despliegue multicloud

Despliegue multinube. Añade un builder de otro tipo para una nube pública (secundario AZURE) en caso de usar otro proveedor y justificar su uso.

Esta tercera parte describe el proceso de obtención final del proyecto. En particular, el objetivo de esta parte es realizar el despliegue automatizado como antes pero esta vez hacerlo multicloud, a la vez en dos nubes diferentes. Para poder realizar las capturas y redactar el proceso bien, se ha habilitado un comando con el que puedes controlar la nube donde quieras lanzar la ejecución. Esto permite que no se mezclen los logs y que sea unas capturas más ejemplificativas. En primer lugar se muestra el proceso de despliegue de aws y después el de azure. En el vídeo se puede ver como se despliegan simultáneamente.

► **Descarga de herramienta Azure CLI**

- Link de descarga para instalar la herramienta de Azure Cli

<https://learn.microsoft.com/en-us/cli/azure/install-azure-cli-windows?tabs=azurecli>

```
Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://
User> az --version
azure-cli          2.67.0
core              2.67.0
telemetry         1.1.0

Dependencies:
msal              1.31.0
azure-mgmt-resource 23.1.1

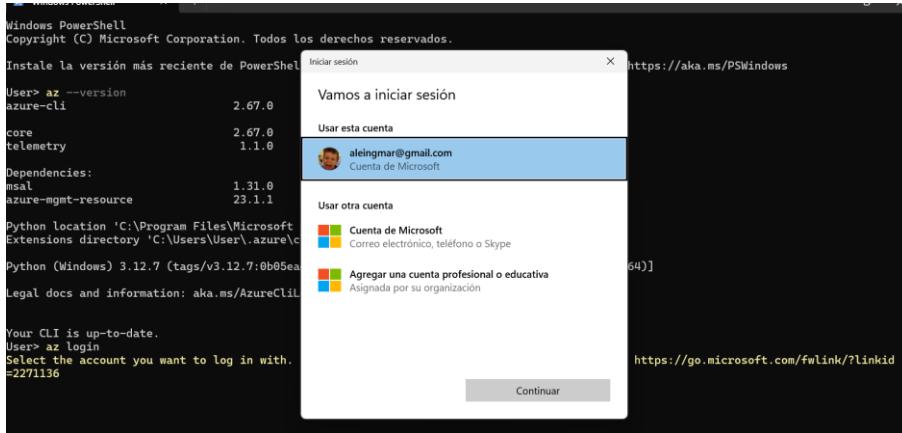
Python location 'C:\Program Files\Microsoft SDKs\Azure\CLI2\python.exe'
Extensions directory 'C:\Users\User\.azure\cliextensions'

Python (Windows) 3.12.7 (tags/v3.12.7:0b05ead, Oct  1 2024, 03:06:41) [MSC v.1941 64 bit (AMD64)]
Legal docs and information: aka.ms/AzureCliLegal

Your CLI is up-to-date.
User> |
```

- Inicio sesión con mi cuenta

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024



► Reviso el valor de mis credenciales de sesión en Azure

```
User> az login
Select the account you want to log in with. For more information on login with Azure CLI, see https://go.microsoft.com/fwlink/?linkid=2271136

Retrieving tenants and subscriptions for the selection...

[Tenant and subscription selection]
No Subscription name Subscription ID Tenant
--- -----
[1] * Azure for Students 28d01465-██████████ Directorio predeterminado

The default is marked with an *; the default tenant is 'Directorio predeterminado' and subscription is 'Azure for Students' (28d01465-1eab-4260-8ebb-56ce3edb9f98).

Select a subscription and tenant (Type a number or Enter for no changes): 1

Tenant: Directorio predeterminado
Subscription: Azure for Students (28d01465-1eab-4260-8ebb-56ce3edb9f98)

[Announcements]
With the new Azure CLI login experience, you can select the subscription you want to use more easily. Learn more about it and its configuration at https://go.microsoft.com/fwlink/?linkid=2271236

If you encounter any problem, please open an issue at https://aka.ms/azclibug

[Warning] The login output has been updated. Please be aware that it no longer displays the full list of available subscriptions by default.

User> az account list --output table
Name CloudName SubscriptionId TenantId State IsDefault
----- ----- -----
Azure for Students AzureCloud 28d01465-1eab-4260-8ebb-56ce3edb9f98 df2ee206-9be8-45█████████████████████ Enabled True

User> az ad sp create-for-rbac --name "act1-herrdevops-packer" --role Contributor --scopes /subscriptions/28d01465-1eab-4260-8ebb-56ce3edb9f98
Creating 'Contributor' role assignment under scope '/subscriptions/28d01465-1eab-4260-8ebb-56ce3edb9f98'
The output includes credentials that you must protect. Be sure that you do not include these credentials in your code or check the credentials into your source control. For more information, see https://aka.ms/azadsp-cli
{
  "appId": "c26a269b-█████████████████████",
  "displayName": "act1-herrdevops-packer",
  "password": "idz8Q~QzNn",
  "tenant": "df2ee206-9be8-45█████████████████████"
}
```

- Configuramos las credenciales como variables de entorno

```
User> $env:ARM_CLIENT_ID="c26a269b-e1cb-4b8e-833a-████████████████"
User> $env:ARM_CLIENT_SECRET="idz8Q~QzNn████████████████"
User> $env:ARM_TENANT_ID="df2ee206-9be8-45████████████████"
User> $env:ARM_SUBSCRIPTION_ID="act1-herrdevops-packer"
```

```
User> [System.Environment]::SetEnvironmentVariable("ARM_SUBSCRIPTION_ID", "act1-herrdevops-packer")
User> [System.Environment]::SetEnvironmentVariable("ARM_TENANT_ID", "df2ee206-9be8-45████████████████")
User> [System.Environment]::SetEnvironmentVariable("ARM_CLIENT_SECRET", "idz8Q~QzNn████████████████")
User> [System.Environment]::SetEnvironmentVariable("ARM_CLIENT_ID", "c26a269b-e1cb-4b8e-833a-████████████████")
User>
```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

```
User> Get-ChildItem Env: | Where-Object { $_.Name -like "ARM_*" }

Name          Value
----          -----
ARM_CLIENT_ID c26a269b-e1cb-4bcd-
ARM_CLIENT_SECRET idz8Q~QzNnerq
ARM_SUBSCRIPTION_ID act1-herrdevops-packer
ARM_TENANT_ID df2ee206-9be8-4502-
```

```
User> az account show
{
  "environmentName": "AzureCloud",
  "homeTenantId": "df2ee206-[REDACTED]",
  "id": "28d01465-1eab-4260-8ebb-56ce3edb9f98",
  "isDefault": true,
  "managedByTenants": [],
  "name": "Azure for Students",
  "state": "Enabled",
  "tenantDefaultDomain": "aleingmargmail.onmicrosoft.com",
  "tenantDisplayName": "Directorio predeterminado",
  "tenantId": "df2ee206-9be8-[REDACTED]",
  "user": {
    "name": "aleingmar@gmail.com",
    "type": "user"
  }
}
User> |
```

► Reniciar el secreto de este servicio principal

```
User> az ad sp credential reset --id c26a269b-e1cb-4bcd-b7c1-3ff2ba03ba3e
The output includes credentials that you must protect. Be sure that you do not include t
edentials into your source control. For more information, see https://aka.ms/azadsp-cl
{
  "appId": "c26a269b-e1cb-4bcd-b7c1-3ff2ba03ba3e",
  "password": "[REDACTED]ECsKOQWr~4nwkab1",
  "tenant": "df2ee206-9be8-4502-84cd-4e1762dbd20a"
}
```

► Instalación de plugin de packer para desplegar y crear la imagen en Azure

```
• packer> packer init main.pkr.hcl
  installed plugin github.com/hashicorp/azure v2.2.0 in "C:/Users/User/AppData/Roaming/packer.d/plugins/github.com/hashicorp/azure/packer-plugin-azure_v2.2.0_x5.0_windows_amd64.exe"
◆ packer> |
```

► **Proceso de despliegue en AWS.** Proceso de despliegue de servicios web únicamente en aws y proceso de creacion de la imagen

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

```

PROBLEMS OUTPUT TERMINAL PORTS COMMENTS
terraform terraform apply -var "deployment_target=aws" ^ -var "aws_access_key=$env:PKR_VAR_aws_access_key" ^ -var "aws_secret_key=$env:PKR_VAR_aws_secret_key" ^ -var "aws_session_token=$env:PKR_VAR_session_token" ^ -var "azure_subscription_id=$env:ARM_SUBSCRIPTION_ID" ^ -var "azure_client_id=$env:ARM_CLIENT_ID" ^ -var "azure_client_secret=$env:ARM_CLIENT_SECRET" ^ -var "azure_tenant_id=$env:ARM_TENANT_ID"
null_resource.packer_ami[0]: Refreshing state... [id=1157172209659885200]
data.aws_vpc.default[0]: Reading...
data.aws_vpc.default[0]: Read complete after 3s [id=vpcl-053265629dc400388]
aws_security_group.web_server_sg[0]: Refreshing state... [id=sg-e4ba4be148a0e82d62]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create
-/ destroy and then create replacement
<+ read (data resources)

Terraform will perform the following actions:

# data.aws_ami.latest_ami[0] will be read during apply
# (depends on a resource or a module with changes pending)
<+ data.aws_ami "latest_ami" {
    + architecture      = (known after apply)
    + arn               = (known after apply)
}
```

► Instancia levantada por packer para la creacion de la imagen

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
i-02517d9998f9c23c9	Running	t2.micro	Initializing		

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
i-02517d9998f9c23c9	Stopped	t2.micro	-		

► Logs de la ejecución del aprovisionamiento de la instancia para crear la imagen.

```

null_resource.packer_ami[0] (local-exec): cloud-node-nginx.amazon-eks.aws_builder: [PM2] Spawning PM2 daemon with pm2_home=/root/.pm2
null_resource.packer_ami[0] (local-exec): cloud-node-nginx.amazon-eks.aws_builder: [PM2] PM2 successfully daemonized
null_resource.packer_ami[0] (local-exec): cloud-node-nginx.amazon-eks.aws_builder: [PM2] Starting /home/ubuntu/app.js in fork_mode (1 instance)
null_resource.packer_ami[0] (local-exec): cloud-node-nginx.amazon-eks.aws_builder: [PM2] Done.
null_resource.packer_ami[0] (local-exec): cloud-node-nginx.amazon-eks.aws_builder: [PM2] Watching
null_resource.packer_ami[0] (local-exec): cloud-node-nginx.amazon-eks.aws_builder: [PM2] Stopped
null_resource.packer_ami[0] (local-exec): cloud-node-nginx.amazon-eks.aws_builder: [PM2] Done.
null_resource.packer_ami[0] (local-exec): cloud-node-nginx.amazon-eks.aws_builder: [PM2] Waiting for the instance to stop...
null_resource.packer_ami[0] (local-exec): still creating... [3m20s elapsed]
null_resource.packer_ami[0] (local-exec): still creating... [3m55s elapsed]
null_resource.packer_ami[0] (local-exec): still creating... [4m50s elapsed]
null_resource.packer_ami[0] (local-exec): still creating... [4m55s elapsed]
```

► Imagen creada

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

Amazon Machine Images (AMIs) (1) [Info](#)

(C) Recycle Bin EC2 Image Builder Actions ▾ Launch instance from AMI

Owned by me Find AMI by attribute or tag < 1 > ⚙

Name	AMI name	AMI ID	Source	Owner
IMAGEN_Node_Nginx	ami-06682737c35dbec37	590184066359/IMAGEN_Node_Nginx	590184066359	

► Máquina levantada con el servicio en cuestión

Instances (3) [Info](#) Last updated less than a minute ago (C) Connect Instance state Actions ▾ Launch instances

Find Instance by attribute or tag (case-sensitive) All states < 1 > ⚙

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Instance_Node...	i-025ab0c524feae0c4	Running	t2.micro	2/2 checks passed	View alarms +

```
aws_instance.web_server[0] (remote-exec): Checking host key: raise
aws_instance.web_server[0] (remote-exec): Target Platform: unix
aws_instance.web_server[0] (remote-exec): Connected!
aws_instance.web_server[0]: Still creating... [30s elapsed]
aws_instance.web_server[0] (remote-exec): La instancia está configurada correctamente.
aws_instance.web_server[0]: Creation complete after 33s [id=i-025ab0c524feae0c4]

Apply complete! Resources: 2 added, 0 changed, 1 destroyed.

Outputs:

aws_instance_id = "i-025ab0c524feae0c4"
aws_public_ip = "54.196.175.149"
└ terraform>
```

► Servicio accesible a través de internet con su red

← → C No es seguro 54.196.175.149

Hello World!

► Servicio activo

ubuntu@ip-172-31-18-146:~\$ sudo pm2 list

id	name	namespace	version	mode	pid	uptime	idle	status	cpu	mem	user	watching
0	app	default	N/A	fork	891	4m	0	online	0%	40.1mb	root	disabled

► Proceso de despliegue en Azure. Proceso de despliegue de servicios web únicamente en aws y proceso de creacion de la imagen

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

The screenshot shows the Microsoft Azure portal's 'Suscripciones' (Subscriptions) page. At the top, there are buttons for 'Agregar' (Add) and 'Opciones avanzadas' (Advanced options). Below this, a note states: 'Los administradores globales pueden administrar todas las suscripciones de esta lista actualizando su configuración de directiva [aqui](#)'. It also mentions: 'Consulte la lista de suscripciones para las que tiene permisos de control de acceso basado en roles (RBAC) para administrar los recursos de Azure. Si quiere ver aquellas para las que tiene acceso de facturación, [haga clic aqui](#)'. A note at the bottom says: 'Se muestran las suscripciones en el directorio Directorio predeterminado. ¿No ve ninguna suscripción? [Cambiar directorios](#)'.

Nombre de la susc...	Id. de la suscripción	Mi rol	Costo actual	Puntuación de seg...	Grupo de administración pri...	Estado
Azure for Students	28d01465-1eab-4260-8ebb-56ce...	Propietario	0,00	100%	Tenant Root Group	Activo

► Busqueda de imágenes de SO base posibles para las instancias de Azure

```
● terraform> az vm image list-skus --location "East US" --publisher Canonical --offer UbuntuServer --output table
Location      Name
-----      -----
eastus        12.04.5-LTS
eastus        14.04.0-LTS
eastus        14.04.2-LTS
eastus        14.04.3-LTS
eastus        14.04.4-LTS
eastus        14.04.5-LTS
eastus        16.04-DAILY-LTS
eastus        16.04-LTS
eastus        16.04.0-LTS
eastus        16_04-daily-lts-gen2
eastus        16_04-lts-gen2
eastus        16_04_0-lts-gen2
eastus        18.04-DAILY-LTS
eastus        18.04-LTS
eastus        18.10
eastus        18_04-daily-lts-arm64
eastus        18_04-daily-lts-gen2
eastus        18_04-lts-arm64
eastus        18_04-lts-gen2
eastus        19.04
eastus        19.10-DAILY
eastus        19_04-daily-gen2
eastus        19_04-gen2
eastus        19_10-daily-gen2
```

► Tipos de instancias disponibles en la region

64	688128	Standard_E112iads_v5	112	1047552	3891200
4	4096	Standard_B2als_v2	2	1047552	0
4	8192	Standard_B2as_v2	2	1047552	0
4	1024	Standard_B2ats_v2	2	1047552	0
8	8192	Standard_B4als_v2	4	1047552	0
8	16384	Standard_B4as_v2	4	1047552	0
16	16384	Standard_B8als_v2	8	1047552	0
16	32768	Standard_B8as_v2	8	1047552	0
32	32768	Standard_B16als_v2	16	1047552	0
32	65536	Standard_B16as_v2	16	1047552	0
32	65536	Standard_B32als_v2	32	1047552	0
32	131072	Standard_B32as_v2	32	1047552	0
2	512	Standard_B1ls	1	1047552	4096
2	2048	Standard_B1ms	1	1047552	4096
2	1024	Standard_B1s	1	1047552	4096
4	8192	Standard_B2ms	2	1047552	16384
4	4096	Standard_B2s	2	1047552	8192
8	16384	Standard_B4ms	4	1047552	32768
16	32768	Standard_B8ms	8	1047552	65536
16	49152	Standard_B12ms	12	1047552	98304
32	65536	Standard_B16ms	16	1047552	131072
32	81920	Standard_B20ms	20	1047552	163840
4	3584	Standard_D1_v2	1	1047552	51200
2	7168	Standard_D2_v2	2	1047552	102400

► Recursos levantados en el proceso de creación de la imagen de aws

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

Filtrar por cualquier ca... Suscripción es igual a todo Grupo de recursos es igual a todo Tipo es igual a todo Ubicación es igual a todo Agregar filtro

Recursos no seguros	Recomendaciones	Recursos cambiados	Agrupar por grupo de rec...	Vista de lista
Nombre ↑	Tipo ↑	Grupo de recursos ↑	Ubicación ↑	Suscripción ↑
networkwatcherrg	Network Watcher	NetworkWatcherRG	East US	Azure for Students
NetworkWatcher_eastus				
packer-images	Interfaz de red	packer-images	East US	Azure for Students
Instance-Node-Nginx-nic	Grupo de seguridad de red	packer-images	East US	Azure for Students
Instance-Node-Nginx-nsg	Dirección IP pública	packer-images	East US	Azure for Students
Instance-Node-Nginx-public-ip	Red virtual	packer-images	East US	Azure for Students
Instance-Node-Nginx-vnet				
pkr-resource-group-smcky9al8				
pkripsmcky9al8	Dirección IP pública	pkr-Resource-Group-smcky9al8	East US	Azure for Students
pkrnismcky9al8	Interfaz de red	pkr-Resource-Group-smcky9al8	East US	Azure for Students
pkrossmcky9al8	Disco	PKR-RESOURCE-GROUP-SMCK...	East US	Azure for Students
pkrgsmcky9al8	Grupo de seguridad de red	pkr-Resource-Group-smcky9al8	East US	Azure for Students
pkrvmsmcky9al8	Máquina virtual	PKR-RESOURCE-GROUP-SMCK...	East US	Azure for Students
pkrvnsmcky9al8	Red virtual	pkr-Resource-Group-smcky9al8	East US	Azure for Students

< Anterior Página 1 de 1 Siguiente > Mostrando de 1 a 11 de 11 registros.

► Máquina virtual levantada por packer para crear la imagen

pkrvmsmcky9al8

Información general

Información esencial

Vista JSON

Conectar Iniciar Reiniciar Detener Hibernar Captura

Estado: En ejecución

Ubicación: East US

Suscripción: Azure for Students

Id. de suscripción: 28d01465-1eab-4260-8ebb-56ce3edb9f98

Sistema operativo: Linux (ubuntu 18.04)

Tamaño: Standard B1ls (1 vcpu, 0.5 GiB de memoria)

Dirección IP pública:

Red virtual/subred:

Nombre DNS:

Estado de mantenimiento:

Hora de creación: 15/12/2024, 5:39 UTC

► Imagen creada

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

Microsoft Azure Buscar recursos, servicios y documentos (G+)

Copilot DIRECTORIO PREDETERMINADO aleingmar@gmail.com

Inicio > Todos los recursos

Filtrar por cualquier ca... Suscripción es igual a todo Grupo de recursos es igual a todo Tipo es igual a todo Ubicación es igual a todo Agregar filtro

Recursos no seguros Recomendaciones Recursos cambiados

Agrupar por grupo de rec... Vista de lista

Nombre ↑	Tipo ↑	Grupo de recursos ↑	Ubicación ↑	Suscripción ↑	...
NetworkWatcher_eastus	Network Watcher	NetworkWatcherRG	East US	Azure for Students	...
packer-images					...
custom-azure-image	Imagen	packer-images	East US	Azure for Students	...
Instance-Node-Nginx-nic	Interfaz de red	packer-images	East US	Azure for Students	...
Instance-Node-Nginx-vnet	Red virtual	packer-images	East US	Azure for Students	...
pkr-resource-group-3d6fhv1gb6					...
pkros3d6fhv1gb6	Disco	PKR-RESOURCE-GROUP-3D6F...	East US	Azure for Students	...

Microsoft Azure Buscar recursos, servicios y documentos (G+)

Copilot DIRECTORIO PREDETERMINADO aleingmar@gmail.com

Inicio > Todos los recursos >

Todos los recursos

Filtrar por cualquier campo...

Nombre ↑

networkwatcherr NetworkWatcher_eastus packer-images custom-azure-image Instance-Node-Nginx-nic Instance-Node-Nginx-osdisk Instance-Node-Nginx-vm Instance-Node-Nginx-vnet

Página 1 de 1

custom-azure-image

Imagen

Buscar + Crear máquina virtual Clonar en una imagen de máquina virtual Eliminar ...

Información general

- Registro de actividad
- Control de acceso (IAM)
- Etiquetas
- Configuración
- Automation
- Ayuda

Essentials

- Grupo de recursos (mover) packer-images Sistema operativo Linux
- Ubicación (mover) East US Máquina virtual de origen pkrvm3d6fhv1gb6
- Suscripción (mover) Azure for Students Generación de VM V1
- Id. de suscripción 28d01465-1eab-4260-8ebb-56ce3edb9f98 Resistencia de zona Deshabilitado
- Estado de aprovisionamiento Succeeded
- Etiquetas (editar) environment: dev

Disco del SO

► Instancia para levantar la imagen y desplegar el servicio

Microsoft Azure Buscar recursos, servicios y documentos (G+)

Copilot DIRECTORIO PREDETERMINADO aleingmar@gmail.com

Inicio > Todos los recursos >

Todos los recursos

Filtrar por cualquier campo...

Nombre ↑

NetworkWatcher_eastus packer-images custom-azure-image Instance-Node-Nginx-nic Instance-Node-Nginx-nsg Instance-Node-Nginx-osdisk Instance-Node-Nginx-public-ip Instance-Node-Nginx-vm Instance-Node-Nginx-vnet

Página 1 de 1

Instance-Node-Nginx-vm

Máquina virtual

Buscar Ayuda para copiar esta máquina virtual en cualquier región

Conectar ▶ Iniciar Reiniciar Detener Hibernar Captura ...

Información general

- Registro de actividad
- Control de acceso (IAM)
- Etiquetas
- Diagnosticar y solucionar problemas
- Conectar
- Redes
- Configuración
- Disponibilidad y escala
- Seguridad
- Copia de seguridad y recuperación
- ante desastres

Información esencial

- Grupo de recursos (mover) packer-images Sistema operativo Linux (ubuntu 18.04)
- Estado En ejecución Tamaño Standard B1s (1 vcpu, 0.5 GiB de memoria)
- Ubicación East US Dirección IP pública 20.169.211.161
- Suscripción (mover) Azure for Students Red virtual/subred Instance-Node-Nginx-vnet/Instance-Node...
- Id. de suscripción 28d01465-1eab-4260-8ebb-56ce3edb9f98 Nombre DNS Sin configurar Estado de mantenimiento
- Hora de creación 15/12/2024, 7:36 UTC

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Inglés Martínez Nombre: Alejandro	04/12/2024

```
te/virtualMachines/Instance-Node-Nginx-vm]
Apply complete! Resources: 9 added, 0 changed, 0 destroyed.

Outputs:

azure_public_ip = "20.169.211.161"
azure_vm_id = "/subscriptions/28d01465-1eab-4260-8ebb-56ce3edb9f98/resourceGroups/packer-image
azure_vm_ip = "10.0.1.4"
```

► Servicio activo y accesible desde internet

```
Last login: Sun Dec 15 07:37:15 2024 from 116.255.16.57
ubuntu@Instance-Node-Nginx:~$ pm2 list
  id  name  namespace  version  mode  pid  uptime  ⚡  status  cpu  mem  user  watching
  0  app   default    N/A      fork  1676  2m     0  online  0%  38.9mb  ubuntu  disabled
ubuntu@Instance-Node-Nginx:~$
```

← → ⌛ ⚠ No es seguro 20.169.211.161

Hello World!

► Estructura de ficheros final

