**UNIVERSIDAD INTERNACIONAL DE LA RIOJA (UNIR)**  
*La Universidad en Internet*

*Herramientas DevOps*

*(MEXDEVOPS)*

**Maestría en Desarrollo y Operaciones de Software**  
**2025**

Actividad 1: Despliegue de Nginx y Nodejs mediante una imagen generada por Packer

Luis Enrique Méndez Cantero

Índice

[**Ejercicio 1: creación de un template de Packer** 3](#_Toc196424412)

[Secciones: 3](#_Toc196424413)

[¿Cómo verificar si el template de Packer es correcto? 4](#_Toc196424414)

[Preparación de pruebas para AWS a través de CLI 4](#_Toc196424415)

[Crear usuario en IAM con acceso programático. 5](#_Toc196424416)

[Configurar credenciales de acceso al CLI. 6](#_Toc196424417)

[Configurar AWS configure en mi computadora. 6](#_Toc196424418)

[Ejecución de packer build. 7](#_Toc196424419)

[**Ejercicio 2: despliegue sin intervención manual** 7](#_Toc196424420)

# **Ejercicio 1: creación de un template de Packer**

**Objetivo**: crear una imagen con Node.js y Nginx ya instalados y configurados.

**Sistema base**: ubuntu 20.04 (tal y como menciona la guía de DigitalOcean)

**AMI base**: se obtiene desde el buscador de imágenes oficiales de Ubuntu:

<https://cloud-images.ubuntu.com/locator/ec2/>

|  |  |
| --- | --- |
| Nombre | Focal Fossa |
| Versión | 20.04 LTS |
| Arquitectura | amd64 |
| Tipo de instancia | hvm:ebs-ssd |
| Región | eu-central-1 |
| AMI ID | ami-0a0bfc6462a1bba50 |

Vamos a hacer uso del formato recomendado por Hashi Corp llamado HCL2, creando el template con nombre ejercicio1.pkr.hcl. En este template se defina una infraestructura mínima para desplegar una aplicación Node.js con PM2 y Nginx sobre Ubuntu 20.04. El archivo automatiza completamente la configuración, desde la instalación del sistema hasta el arranque de la aplicación como servicio persistente gracias a PM2. Además, la aplicación será expuesta al público mediante un proxy inverso Nginx.

## Secciones:

* Plugins
  + Amazon
* Variables
  + aws\_region, source\_ami, instance\_type
* Source
  + amazon-ebs
    - region
    - source\_ami
    - instance\_type
    - ssh\_username
    - ami\_name
    - associate\_public\_ip\_address
* Build (Bloque principal donde se define qué hacer con los sources)
  + Provisioner “Shell” para ejecutar el script de setup
* Script de instalación (setup.sh)
  + Instalación de dependencias del sistema
  + Instalación de Node.js y PM2
  + Creación de la aplicación de Node.js
  + Ejecución y configuración con PM2
  + Configuración de Nginx como proxy

## ¿Cómo verificar si el template de Packer es correcto?

Dentro de la carpeta donde está el archivo ejercicio1.pkr.hcl ejecutar el siguiente comando para inicializar y descargar los plugins necesarios por el template:

PS D:\UNIR\Maestria\unir\2\_semestre\Herramientas\_DevOps\actividad1> packer init .

Installed plugin github.com/hashicorp/amazon v1.3.6 **in** "C:/Users/lemen/AppData/Roaming/packer.d/plugins/github.com/hashicorp/amazon/packer-plugin-amazon\_v1.3.6\_x5.0\_windows\_amd64.exe"

Para comprobar que es válido utilizar el siguiente comando:

PS D:\UNIR\Maestria\unir\2\_semestre\Herramientas\_DevOps\actividad1> packer validate ejercicio1.pkr.hcl

The configuration is valid.

## Preparación de pruebas para AWS a través de CLI

Para comprobar que funciona en AWS primero instalaremos el CLI dentro de mi computadora Windows 11 siguiendo el siguiente tutorial:

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto. Interfaz de usuario gráfica, Texto, Aplicación, Word

El contenido generado por IA puede ser incorrecto.

Resultado:  
PS D:\UNIR\Maestria\unir\2\_semestre\Herramientas\_DevOps\actividad1> aws --version

aws-cli/2.27.0 Python/3.13.2 Windows/11 exe/AMD64

### Crear usuario en IAM con acceso programático.

Dentro de <https://console.aws.amazon.com/iam> agregar un usuario (packer-user).

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.Adjuntar políticas directamente.

* AmazonEC2FullAccess
* AmazonVPCFullAccess

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

### Configurar credenciales de acceso al CLI.

Escala de tiempo

El contenido generado por IA puede ser incorrecto.

### Configurar AWS configure en mi computadora.

PS D:\UNIR\Maestria\unir\2\_semestre\Herramientas\_DevOps\actividad1> aws configure

AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE

AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

**Default** region name [None]: eu-central-1

**Default** output format [None]:

## Ejecución de packer build.

Dentro del directorio donde se encuentra ejercicio1.pkr.hcl y setup.sh ejecutar:

packer build ejercicio1.pkr.hcl

Resultado:

Build 'amazon-ebs.ubuntu\_node\_pm2' finished after 5 minutes 54 seconds.

==> Wait completed after 5 minutes 54 seconds

==> Builds finished. The artifacts of successful builds are:

--> amazon-ebs.ubuntu\_node\_pm2: AMIs were created:

eu-central-1: ami-08c38b97c14fbea94

Dentro de AWS se vería así:

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

# **Ejercicio 2: despliegue sin intervención manual**