

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Lopez Baya	
	Nombre: Christian	

Informe sobre Trabajo con Packer, Azure y AWS para Herramientas DevOps

Criterio 1: Creación de una Imagen de Packer en AWS

Se ha desarrollado un proceso para la creación de una imagen utilizando Packer en AWS. El flujo de trabajo se diseñó considerando las mejores prácticas de seguridad y eficiencia. Se han empleado scripts y configuraciones específicas para asegurar la creación correcta de la imagen con todos los requisitos y dependencias necesarias.

Se comenzó creando el archivo “**node-nginx.pkr.hcl**”, se decidió usar este formato ya que es el mas reciente para poder manejar scripts Packer además de ser mas sencillo que JSON, en la parte superior se incluyo el plugin a ser instalado de Amazon para el comando “**packer init**”

```
≡ node-nginx.pkr.hcl
1 packer {
2   required_plugins{
3     amazon = {
4       source  = "github.com/hashicorp/amazon"
5       version = "~> 1"
6     }
7
8     azure = {
9       source  = "github.com/hashicorp/azure"
10      version = "~> 2"
11    }
12  }
13}
```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Lopez Baya	
	Nombre: Christian	

Posteriormente se definió la variable “**locals**” donde se definió el timestamp para poder adjuntarlo al nombre de la imagen que se generara en la segunda parte al definir el **Source**.

Posteriormente se definió el “**Source**” donde se aclaró que se usará “amazon-ebs” ya que es la mejor manera de poder crear una imagen ya que cuenta con replicación en regiones desplegadas, seguidamente se definió el nombre de la imagen como “Nginx-Node”, se configuró un filtro el cual está asociado directamente a la imagen la cual se está usando en Amazon, el tipo de instancia “**t2.micro**” al ser disponible para cuentas free y las credenciales creadas para poder acceder directamente a la cuenta, como también la región a la cual pertenecerá. En este caso al yo pertenecer a Bolivia, la más cercana era Brasil con São Paulo y por último el username para poder acceder por ssh.

```

locals {
  timestamp = regex_replace(timestamp(), "[ - TZ:]", "") 1
}

source "amazon-ebs" "Nginx-Node"{
  ami_name = "Nginx-Node-app-${local.timestamp}" 2

  source_ami_filter {
    filters = {
      name          = "al2023-ami-2023.2.20231113.0-kernel-6.1-x86_64"
      root-device-type = "ebs"
      virtualization-type = "hvm"
    }
    most_recent = true
    owners      = ["amazon"]
  }
  instance_type = "t2.micro"
  access_key= "AKIAZATYSUGMFVUFUM4L"
  secret_key= "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"
  region = "sa-east-1"
  ssh_username = "ec2-user"
}

```

Posteriormente se hizo la definición de que sources se agarrara para el build donde se definió el source previamente definido para Amazon, como también el de Azure ya que en este caso estoy creando la imagen para estas 2 nubes públicas. También se inició el provisionamiento en este caso de la aplicación “**NodeAPP.zip**” que la tengo

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Lopez Baya Nombre: Christian	

creada en Node comprimido, como también se provisiono el “node.service” para que pueda instalarse como aplicación en segundo plano al iniciar el despliegue de la imagen y por ultimo se corrió el archivo “app.sh” donde se alojan todos los scripts en formato Shell para la instalación de dependencias y configuración de la imagen.
(Todo ello se detallará en la siguiente Rubrica)

```

build {
  sources = [
    "source.amazon-ebs.Nginx-Node",
    "source.azure-arm.Nginx-Node-Azure",
  ]
}

provisioner "file" {
  source = "../NodeAPP.zip"
  destination = "/home/ec2-user/NodeAPP.zip"
}

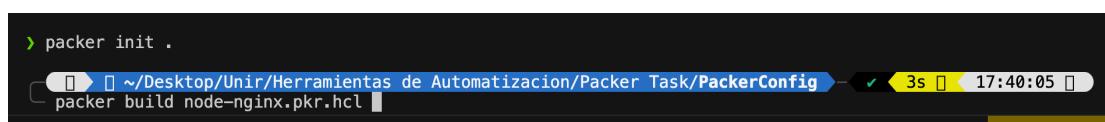
provisioner "file" {
  source = "./node.service"
  destination = "/tmp/node.service"
}

provisioner "shell" {
  script = "./app.sh"
}

```

1 2 3 4

Luego se corrió el comando “**packer init**” . para poder realizar la instalación de los plugins necesarios definidos al inicio, posteriormente se aplico el comando “**packer build node-nginx.pkr.hcl**” para poder iniciar la creación de las imágenes



```

> packer init .
[...]
└─ packer build node-nginx.pkr.hcl

```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Lopez Baya Nombre: Christian	

Teniendo como resultado los logs finalizados de creación con el respectivo nombre de creación en Amazon y Azure(Se detalla en la rubrica 3)

```
==> Wait completed after 7 minutes 40 seconds
==> Builds finished. The artifacts of successful builds are:
--> amazon-ebs.Nginx-Node: AMIs were created:
sa-east-1: ami-0da6e7aee9b9c04f1
--> azure-arm.Nginx-Node-Azure: Azure.ResourceManagement.VMImage:
```

Finalmente se puede evidenciar la creación de la imagen en la parte de AMI's en mi cuenta de AWS lista para ser desplegada

The screenshot shows the AWS EC2 dashboard with the 'Images' section selected. In the main pane, it displays the 'Amazon Machine Images (AMIs) (1/1)' page. A single AMI entry is listed, which has been highlighted with a red box. The entry shows the following details:

Name	AMI name	AMI ID	Source	Owner
Nginx-Node-app-20231126182355	ami-0da6e7aee9b9c04f1	619801911704/Nginx-Node-app-20231126182355	619801911704	619801911704

Below this, a detailed view of the AMI is shown with the ID 'ami-0da6e7aee9b9c04f1'. The details include:

- AMI ID: ami-0da6e7aee9b9c04f1
- Image type: machine
- Platform details: Linux/UNIX
- Root device type: EBS
- AMI name: Nginx-Node-app-20231126182355
- Owner account ID: 619801911704
- Architecture: x86_64
- Usage operation: RunInstances
- Root device name: /dev/xvda
- Status: Available
- Source: 619801911704/Nginx-Node-app-20231126182355
- Virtualization type: hvm

Criterio 2: Instalación correcta del stack y evidencia adjunta

Primeramente, partimos del aprovisionamiento de nuestras imágenes

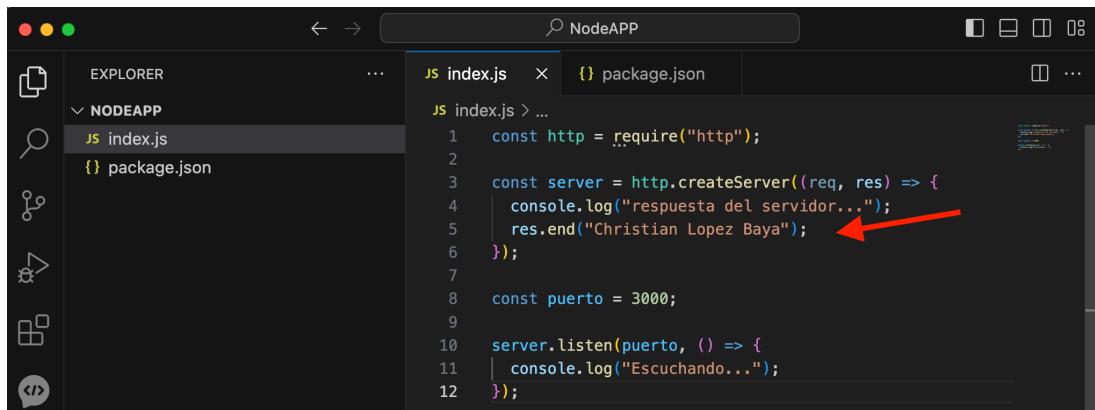
```
provisioner "file" {
  source = "../NodeAPP.zip"
  destination = "/home/ec2-user/NodeAPP.zip"
} 1

provisioner "file" {
  source = "./node.service"
  destination = "/tmp/node.service"
} 2

provisioner "shell" {
  script = "./app.sh"
} 3
```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Lopez Baya Nombre: Christian	

En el primer aprovisionamiento de tipo “file” se copio la aplicación previamente creada en Node la cual realmente es básica y únicamente muestra mi nombre al momento de levantar el servidor.

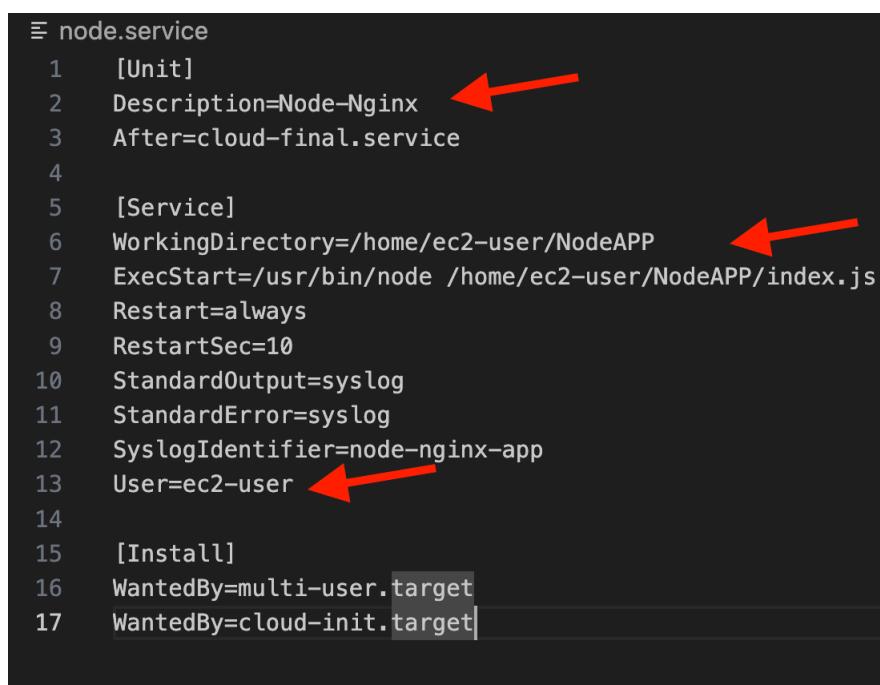


```

const http = require("http");
const server = http.createServer((req, res) => {
  console.log("respuesta del servidor...");
  res.end("Christian Lopez Baya");
});
const puerto = 3000;
server.listen(puerto, () => {
  console.log("Escuchando...");
});

```

En el segundo aprovisionamiento de tipo “file” se aprovisiono de la configuración en segundo plano de nuestra aplicación en Node lo que significa que al deployarse la imagen esta correrá en segundo plano haciendo referencia a la ruta del proyecto la cual será copiada a la ruta de “SystemD” de nuestra imagen para convertirla en segundo plano y por ultimo iniciar el servicio el cual se detallara a continuación



```

[Unit]
Description=Node-Nginx
After=cloud-final.service

[Service]
WorkingDirectory=/home/ec2-user/NodeAPP
ExecStart=/usr/bin/node /home/ec2-user/NodeAPP/index.js
Restart=always
RestartSec=10
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=node-nginx-app
User=ec2-user

[Install]
WantedBy=multi-user.target
WantedBy=cloud-init.target

```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Lopez Baya	
	Nombre: Christian	

Por ultimo se aprovisiono de un archivo llamado “**app.shell**” el cual instala todas las dependencias a ser usadas en la imagen como ser, Node, Nginx, Unzip. El cual posteriormente descomprime la aplicación previamente aprovisionada, copia el archivo de servicio aprovisionado anteriormente al path correcto en systemD e inicia los servicios en segundo plano.

```
$ app.sh
1  #!/bin/bash
2
3  sleep 30
4
5  sudo yum update -y
6
7  sudo yum install -y gcc-c++ make
8  curl -sL https://rpm.nodesource.com/setup_14.x | sudo -E bash -
9  sudo yum install -y nodejs
10
11 sudo yum install unzip -y
12 sudo yum install nginx -y
13 cd ~/ && unzip NodeAPP.zip
14 cd ~/NodeAPP && npm i
15
16 sudo mv /tmp/node.service /etc/systemd/system/node.service
17 sudo systemctl enable node.service
18 sudo systemctl start node.service
```

a tambien en los logs se pueden ver la correcta instalación de dependencias.

```
amazon-ebs.Nginx-Node: =====
amazon-ebs.Nginx-Node: Package      Arch   Version           Repository  Size
amazon-ebs.Nginx-Node: =====
amazon-ebs.Nginx-Node: Installing:
amazon-ebs.Nginx-Node: nodejs        x86_64  1:18.18.2-1.amzn2023.0.1      amazonlinux 1.8 M
amazon-ebs.Nginx-Node: Installing dependencies:
amazon-ebs.Nginx-Node: libbrotli     x86_64  1.0.9-4.amzn2023.0.2      amazonlinux 315 k
amazon-ebs.Nginx-Node: nodejs-libs   x86_64  1:18.18.2-1.amzn2023.0.1      amazonlinux 14 M
amazon-ebs.Nginx-Node: Installing weak dependencies:
amazon-ebs.Nginx-Node: nodejs-docs   noarch  1:18.18.2-1.amzn2023.0.1      amazonlinux 7.6 M
amazon-ebs.Nginx-Node: nodejs-full-i18n x86_64  1:18.18.2-1.amzn2023.0.1      amazonlinux 8.5 M
amazon-ebs.Nginx-Node: nodejs-npm    x86_64  1:9.8.1-1.18.18.2.1.amzn2023.0.1  amazonlinux 2.0 M
amazon-ebs.Nginx-Node:
amazon-ebs.Nginx-Node: Transaction Summary
amazon-ebs.Nginx-Node: =====
amazon-ebs.Nginx-Node: Install 6 Packages
amazon-ebs.Nginx-Node:
amazon-ebs.Nginx-Node: Total download size: 34 M
amazon-ebs.Nginx-Node: Installed size: 186 M
amazon-ebs.Nginx-Node: Downloading Packages:
amazon-ebs.Nginx-Node: (1/6): libbrotli-1.0.9-4.amzn2023.0.2.x86_64.rpm 2.2 MB/s | 315 KB  00:00
amazon-ebs.Nginx-Node: (2/6): nodejs-18.18.2-1.amzn2023.0.1.x86_64.rpm 11 MB/s | 1.8 MB  00:00
amazon-ebs.Nginx-Node: (3/6): nodejs-npm-9.8.1-1.18.18.2.1.amzn2023.0. 9.4 MB/s | 2.0 MB  00:00
amazon-ebs.Nginx-Node: (4/6): nodejs-full-i18n-18.18.2-1.amzn2023.0.1. 23 MB/s | 8.5 MB  00:00
amazon-ebs.Nginx-Node: (5/6): nodejs-docs-18.18.2-1.amzn2023.0.1.noarc 12 MB/s | 7.6 MB  00:00
amazon-ebs.Nginx-Node: (6/6): nodejs-libs-18.18.2-1.amzn2023.0.1.x86_6 17 MB/s | 14 MB  00:00
amazon-ebs.Nginx-Node: -----
amazon-ebs.Nginx-Node: Total                                         32 MB/s | 34 MB  00:01
```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Lopez Baya Nombre: Christian	

```
amazon-ebs.Nginx-Node: Transaction test succeeded.
amazon-ebs.Nginx-Node: Running transaction
amazon-ebs.Nginx-Node: Preparing : 1/1
amazon-ebs.Nginx-Node: Running scriptlet: nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch 1/7
amazon-ebs.Nginx-Node: Installing : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch 1/7
amazon-ebs.Nginx-Node: Installing : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch 2/7
amazon-ebs.Nginx-Node: Installing : generic-logos-httplib-18.0.0-12.amzn2023.0.3.noarch 3/7
amazon-ebs.Nginx-Node: Installing : libunwind-1.4.0-5.amzn2023.0.2.x86_64 4/7
amazon-ebs.Nginx-Node: Installing : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64 5/7
amazon-ebs.Nginx-Node: Installing : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64 6/7
amazon-ebs.Nginx-Node: Installing : nginx-1:1.24.0-1.amzn2023.0.2.x86_64 7/7
amazon-ebs.Nginx-Node: Running scriptlet: nginx-1:1.24.0-1.amzn2023.0.2.x86_64 7/7
amazon-ebs.Nginx-Node: Verifying : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64 1/7
amazon-ebs.Nginx-Node: Verifying : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64 2/7
amazon-ebs.Nginx-Node: Verifying : libunwind-1.4.0-5.amzn2023.0.2.x86_64 3/7
amazon-ebs.Nginx-Node: Verifying : nginx-1:1.24.0-1.amzn2023.0.2.x86_64 4/7
amazon-ebs.Nginx-Node: Verifying : generic-logos-httplib-18.0.0-12.amzn2023.0.3.noarch 5/7
amazon-ebs.Nginx-Node: Verifying : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch 6/7
amazon-ebs.Nginx-Node: Verifying : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch 7/7
amazon-ebs.Nginx-Node: Installed:
amazon-ebs.Nginx-Node: generic-logos-httplib-18.0.0-12.amzn2023.0.3.noarch
amazon-ebs.Nginx-Node: gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
amazon-ebs.Nginx-Node: libunwind-1.4.0-5.amzn2023.0.2.x86_64
amazon-ebs.Nginx-Node: nginx-1:1.24.0-1.amzn2023.0.2.x86_64
amazon-ebs.Nginx-Node: nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64
amazon-ebs.Nginx-Node: nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
amazon-ebs.Nginx-Node: nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch
amazon-ebs.Nginx-Node: Complete!
amazon-ebs.Nginx-Node: Archive: NodeAPP.zip
amazon-ebs.Nginx-Node: creating: NodeAPP/
amazon-ebs.Nginx-Node: inflating: NodeAPP/index.js
amazon-ebs.Nginx-Node: inflating: NodeAPP/package.json
```

Los logs completos están dentro del archivo comprimido que se subirá como tarea con el nombre “**logsAWS-AZURE.txt**”

Criterio 3: Builder para otra Nube Pública (Azure)

Se ha configurado un builder en Packer para generar una imagen compatible con Azure. Se proporciona una descripción detallada de los pasos necesarios para adaptar el proceso de creación de imágenes en Packer para que sea compatible con la infraestructura de Azure. Se han tomado en cuenta las diferencias clave entre los servicios de AWS y Azure para asegurar la portabilidad de la imagen entre nubes públicas. Se detalla a continuación del proceso de Azure que es similar a AWS únicamente varía el Source.

Primeramente, se comienza definiendo el source específico para AZURE en el archivo “**node-nginx.pkr.hcl**” donde se detallan el uso de “**Azure-arm**” seguido del nombre de la imagen la cual es “**Nginx-Node-Azure**”, posteriormente se detallan mas requerimientos como ser el “**client_id**” el cual se saca de las credenciales de dominio de Azure, el client_Secret y posteriormente se detallan detalles de la imagen, en este caso se uso una versión de CentOS con su respectiva versión, también se detalla la

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Lopez Baya	
	Nombre: Christian	

locación en este caso no estaba disponible la de Brazil Sao Paulo por tal razón se uso de UUEE, se definió el nombre de la imagen como también algunas otras credenciales correspondientes al dominio y por ultimo el tamaño de la imagen en la cual se selecciono “Standard_B1s” ya que es la compatible con versiones free

```
source "azure-arm" "Nginx-Node-Azure" {
    client_id                  = "eb1baf29-ecdb-464e-9510-bbf8ddc16148"
    client_secret               = "4X38Q~Dr5v2eErsaLa4NT7WHsEsas0CBkzXLcbQM"
    image_offer                 = "CentOS"
    image_publisher             = "OpenLogic"
    image_sku                   = "8_5-gen2"
    location                    = "East US"
    managed_image_name          = "Nginx-Node-Azure-${local.timestamp}"
    managed_image_resource_group_name = "packer-rgroup"
    os_type                     = "Linux"
    subscription_id             = "e9a5c656-3fa3-4fde-ac20-d2085991d7d1"
    tenant_id                   = "c24682df-35ba-452f-bc02-8f19f6ca28f5"
    vm_size                     = "Standard_B1s"
    ssh_username                = "ec2-user"
}
```

Posteriormente este nuevo source es incluido en el build junto a Amazon

```
build {
    sources = [
        "source.amazon-ebs.Nginx-Node",
        "source.azure-arm.Nginx-Node-Azure",
    ]
}
```

Como después los demás pasos son los mismos de la instalación y demás los cuales serán adjuntados en capturas a continuación sobre los logs de la instalación de el stack y el log al finalizar la creación de la imagen

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Lopez Baya Nombre: Christian	

```

azure-arm.Nginx-Node-Azure: * https://rpm.nodesource.com/setup_16.x - Node.js v16 "Gallium"
azure-arm.Nginx-Node-Azure: * https://rpm.nodesource.com/setup_18.x - Node.js v18 LTS "Hydrogen" (reco
mmended)
azure-arm.Nginx-Node-Azure: * https://rpm.nodesource.com/setup_19.x - Node.js v19 "Nineteen"
azure-arm.Nginx-Node-Azure: * https://rpm.nodesource.com/setup_20.x - Node.js v20 "Iron" (current)

Please see https://github.com/nodejs/Release for details about which
version may be appropriate for you.

The NodeSource Node.js distributions repository contains
information both about supported versions of Node.js and supported Linux
distributions. To learn more about usage, see the repository:
https://github.com/nodesource/distributions
=====

--- azure-arm.Nginx-Node-Azure: =====
--- =====

=====
azure-arm.Nginx-Node-Azure: Install 54 Packages
azure-arm.Nginx-Node-Azure:
azure-arm.Nginx-Node-Azure: Total download size: 14 M
azure-arm.Nginx-Node-Azure: Installed size: 42 M
azure-arm.Nginx-Node-Azure: Downloading Packages:
azure-arm.Nginx-Node-Azure: (1/54): jbigkit-libs-2.1-14.el8.x86_64.rpm      1.6 MB/s | 55 kB  00:00
azure-arm.Nginx-Node-Azure: (2/54): gd-2.2.5-7.el8.x86_64.rpm      2.9 MB/s | 144 kB 00:00
azure-arm.Nginx-Node-Azure: (3/54): abattis-cantarell-fonts-0.0.25-6.el8.no 2.4 MB/s | 156 kB 00:00
azure-arm.Nginx-Node-Azure: (4/54): libX11-common-1.6.8-5.el8.noarch.rpm 5.0 MB/s | 158 kB 00:00
azure-arm.Nginx-Node-Azure: (5/54): libXau-1.0.9-3.el8.x86_64.rpm    1.3 MB/s | 37 kB 00:00
azure-arm.Nginx-Node-Azure: (6/54): libXpm-3.5.12-8.el8.x86_64.rpm  2.4 MB/s | 58 kB 00:00
azure-arm.Nginx-Node-Azure: (7/54): libjpeg-turbo-1.5.3-12.el8.x86_64.rpm 5.4 MB/s | 157 kB 00:00
azure-arm.Nginx-Node-Azure: (8/54): libtiff-4.0.9-20.el8.x86_64.rpm   4.4 MB/s | 188 kB 00:00
azure-arm.Nginx-Node-Azure: (9/54): libX11-1.6.8-5.el8.x86_64.rpm   4.0 MB/s | 611 kB 00:00
azure-arm.Nginx-Node-Azure: (10/54): libwebp-1.0-0.5.el8.x86_64.rpm  3.1 MB/s | 272 kB 00:00
azure-arm.Nginx-Node-Azure: (11/54): nginx-all-modules-1.14.1-9.module_el8. 1.7 MB/s | 23 kB 00:00
azure-arm.Nginx-Node-Azure: (12/54): libxcb-1.13.1-1.el8.x86_64.rpm  2.4 MB/s | 229 kB 00:00
azure-arm.Nginx-Node-Azure: (13/54): nginx-filesystem-1.14.1-9.module_el8.0 792 kB/s | 24 kB 00:00
azure-arm.Nginx-Node-Azure: (14/54): nginx-mod-http-image-filter-1.14.1-9.m 1.4 MB/s | 35 kB 00:00
azure-arm.Nginx-Node-Azure: (15/54): nginx-1.14.1-9.module_el8.0_0+184+e34f 5.3 MB/s | 570 kB 00:00
azure-arm.Nginx-Node-Azure: (16/54): nginx-mod-http-perl-1.14.1-9.module_el 888 kB/s | 45 kB 00:00
azure-arm.Nginx-Node-Azure: (17/54): nginx-mod-http-xslt-filter-1.14.1-9.mo 667 kB/s | 33 kB 00:00
azure-arm.Nginx-Node-Azure: (18/54): nginx-mod-stream-1.14.1-9.module_el8.0 2.8 MB/s | 85 kB 00:00
azure-arm.Nginx-Node-Azure: (19/54): nginx-mod-mail-1.14.1-9.module_el8.0 1.2 MB/s | 64 kB 00:00
azure-arm.Nginx-Node-Azure: (20/54): fontpackages-filesystem-1.44-22.el8.no 681 kB/s | 16 kB 00:00
azure-arm.Nginx-Node-Azure: (21/54): fontconfig-2.13.1-4.el8.x86_64.rpm  4.4 MB/s | 274 kB 00:00
azure-arm.Nginx-Node-Azure: (22/54): libxslt-1.1.32-6.el8.x86_64.rpm  4.8 MB/s | 250 kB 00:00
azure-arm.Nginx-Node-Azure: (23/54): perl-Carp-1.42-396.el8.noarch.rpm 559 kB/s | 30 kB 00:00
azure-arm.Nginx-Node-Azure: (24/54): perl-Exporter-5.72-396.el8.noarch.rpm 1.5 MB/s | 34 kB 00:00
azure-arm.Nginx-Node-Azure: (25/54): perl-Errno-1.28-420.el8.x86_64.rpm 1.5 MB/s | 76 kB 00:00
azure-arm.Nginx-Node-Azure: (26/54): perl-File-Path-2.15-2.el8.noarch.rpm 1.4 MB/s | 38 kB 00:00
azure-arm.Nginx-Node-Azure: (27/54): perl-File-Temp-0.230.600-1.el8.noarch. 2.2 MB/s | 63 kB 00:00
azure-arm.Nginx-Node-Azure: (28/54): perl-Getopt-Long-2.50-4.el8.noarch.rpm 2.1 MB/s | 63 kB 00:00

--> azure-arm.Nginx-Node-Azure: Azure.ResourceManagement.VMImage:

OSType: Linux
ManagedImageResourceGroupName: packer-rgroup
ManagedImageName: Nginx-Node-Azure-20231126182355
ManagedImageId: /subscriptions/e9a5c656-3fa3-4fde-ac20-d2085991d7d1/resourceGroups/packer-rgroup/providers/Microsoft.Compute/images/Nginx-Node-Azure-20231126182355
ManagedImageLocation: East US

```

Criterio 4: Despliegue sin intervención Humana

Para tal acción en el archivo “node-nginx.pkr.hcl” al final de la configuración se estableció “post-procesadores” los cuales al final de la creación de la AMI en AWS realizan las siguientes acciones:

- “Post-Processor Manifests” el cual se encargar de generar un archivo en formato JSON de la salida de la terminal donde se encuentra la ID_AMI y demás información de la imagen creada

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Lopez Baya Nombre: Christian	

2. "Post-Processor Shell-Local" Lo que hace es agarrar el archivo "Manifests.json" y saca la ID_AMI para posteriormente mediante la CLI de AWS levantar la instancia automaticamente incluyendo la ID_AMI, tipo de instancia, security groups (Donde se Estan Aperturados los Puertos TCP -> 3000)

```

post-processor "manifest" {
  output = "manifest.json"
}

post-processor "shell-local" {
  inline = [
    "# Obtener la ID de la imagen creada
    "AMI_ID=$(jq -r '.builds[-1].artifact_id' manifest.json | cut -d ':' -f2)",
    "echo $AMI_ID",
    "# Lanzar una instancia usando AWS CLI
    "echo 'Lanzando la instancia en AWS...'",
    "aws ec2 run-instances --image-id $AMI_ID --instance-type t2.micro --key-name newKey --security-groups launch-wizard-1",
  ]
}

```

Posteriormente se pudo evidenciar en la consola la salida la salida de la CLI con todos los datos de la instancia desplegada

```

=> amazon-eks.Nginx-Node: Deleting temporary keypair...
=> amazon-eks.Nginx-Node: Running post-processor: (type manifest)
=> amazon-eks.Nginx-Node: Running post-processor: (type shell-local)
=> amazon-eks.Nginx-Node (shell-local): Running local shell script: /var/folders/r2/k4wlh67j20s0kx3z4tbsf1v4000gn/T/packer-shell3652619295
amazon-eks.Nginx-Node (shell-local): ami-071b574ca4a5719eb
amazon-eks.Nginx-Node (shell-local): Lanzando la instancia en AwS... ←
amazon-eks.Nginx-Node (shell-local): {
  "Groups": [],
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-071b574ca4a5719eb",
      "InstanceId": "i-021eb08798015a380",
      "InstanceType": "t2.micro",
      "KeyName": "newKey",
      "LaunchTime": "2023-12-14T19:29:53+00:00",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "sa-east-1c",
        "GroupName": "",
        "Tenancy": "default"
      },
      "PrivateDnsName": "ip-172-31-43-13.sa-east-1.compute.internal",
      "PrivateIpAddress": "172.31.43.13",
      "ProductCodes": [],
      "PublicDnsName": "",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
      "StateTransitionReason": "",
      "SubnetId": "subnet-0148c6a9a5cec9b8f",
      "VpcId": "vpc-0e2a2b539c928b55c",
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "57b8e459-47f3-4d16-9d81-deae16f1d742",
      "EbsOptimized": false,
      "EnaSupport": true,
      "Hypervisor": "xen",
      "NetworkInterfaces": [
        {
          "Attachment": {
            "AttachTime": "2023-12-14T19:29:53+00:00",
            "AttachmentId": "eni-attach-0434f8584ac14731f",
            "DeleteOnTermination": true,
            "DeviceIndex": 0,

```

Posteriormente al ingresar a nuestra consola de AWS ya se puede evidenciar la instancia que esta lanzada como también se puede evidenciar la dirección IP PUBLICA mediante la cual podemos acceder al despliegue

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Lopez Baya Nombre: Christian	

The screenshot shows the AWS EC2 Instances page. At the top, there's a search bar and a 'Launch instances' button. The main area displays a table of instances. One instance, 'i-021eb5879a815a380', is highlighted with a red box. Below the table, a detailed view for this specific instance is shown, also with a red box around its public IPv4 address: '15.228.253.219 [open address]'. Other details shown include the instance ID, state (Running), and DNS names.

Como también podemos ver la aplicación funcional con nuestro nombre desplegado
Con Node y Nginx sin intervención humana alguna



El despliegue desde la imagen generada en Azure se ha realizado desde la misma plataforma

Primeramente, se crearon los security Groups en AZURE para poder permitir cualquier tipo de tráfico TCP proveniente de afuera hacia el puerto 3000 que es en el cual trabaja Node

The screenshot shows the Microsoft Azure NSG settings page for 'Node-Nginx-Azure-nsq'. It displays the resource group ('Node-Nginx-Azure_group'), location ('East US'), and subscription information. The 'Inbound security rules' section is highlighted with a red box. It lists two rules: one for SSH (port 22) and another for port 3000 allowing traffic from anywhere to anywhere.

Priority ↑	Name ↑	Port ↑	Protocol ↑	Source ↑	Destination ↑	Action ↑
300	SSH	22	TCP	Any	Any	Allow
310	AllowAnyCustom3000In...	3000	TCP	Any	Any	Allow

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Lopez Baya Nombre: Christian	

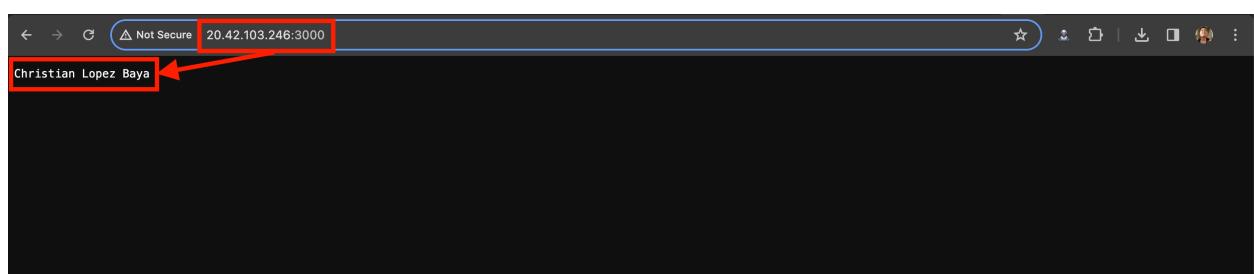
Posteriormente se hizo el despliegue de la imagen creada en AZURE(Virtual Machine) donde se puede evidenciar que está “Running” con su respectiva IP Publica

Se puede Evidenciar la ip publica de AZURE como 20.42.103.246

The screenshot shows the Microsoft Azure portal interface. On the left, the 'Virtual machines' blade is open, displaying a list of VMs. One VM, 'Node-Nginx-Azure', is selected and highlighted with a red box. On the right, the detailed view for 'Node-Nginx-Azure' is shown. The 'Essentials' section provides key information: Resource group (NODE-NGINX-AZURE_GROUP_11271922), Status (Running), Location (East US (Zone 1)), and Public IP address (20.42.103.246). A red arrow points from the 'Status' field to the 'Public IP address' field.

Por ultimo al ingresar al navegador con dicha ip seguida del puerto 3000 se puede evidenciar la aplicación en node funcionando sin ningún problema donde se muestra mi nombre.

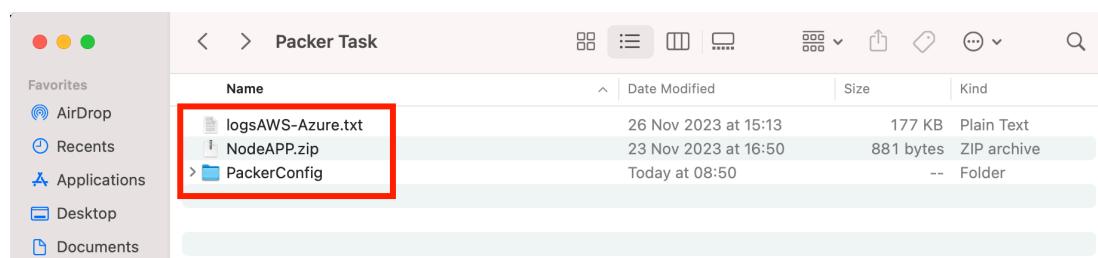
AZURE 20.42.103.246:3000



Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Lopez Baya	
	Nombre: Christian	

Criterio 5: Detalles de los entregables y limitaciones

La entrega se realiza en una carpeta comprimida llamada “**Packer Task**”, dentro de ella se encuentran: El archivo con todos los logs de build del archivo packer llamado “**logsAWS-Azure.txt**” como también la aplicación en node comprimida llamada “**NodeAPP.zip**” y por ultimo la carpeta con todo el código fuente llamado “**PackerConfig**”



En la carpeta “**PackerConfig**” es donde se encuentra el archivo packer “**node-nginx.pkr.hcl**”, el archivo de configuración de servicio “**node.service**” y el archivo con los scripts Shell llamado “**app.sh**”.

```

node-nginx.pkr.hcl
1 packer {
2   required_plugins{
3     amazon = {
4       source  = "github.com/hashicorp/amazon"
5       version = "~> 1"
6     }
7
8     azure = {
9       source  = "github.com/hashicorp/azure"
10      version = "~> 2"
11    }
12  }
13
14  locals {
15    timestamp = regex_replace(timestamp(), "[ -TZ:]", "")
16  }
17
18  source "amazon-ebs" "Nginx-Node"{
19    ami_name = "Nginx-Node-app-${local.timestamp}"
20
21    source_ami_filter {
22      filters = [
23        { name = "al2023-ami-2023.2.20231113.0-kernel-6.1-x86_64" }
24      ]
25    }
26  }
27
28  build {
29    provider = "amazon"
30
31    image {
32      name = "Nginx-Node"
33      ami_id = "ami-0a1a1a1a1a1a1a1a1"
34    }
35
36    variables {
37      app_name = "Nginx-Node"
38      port = 80
39      host_ip = "0.0.0.0"
40      host_port = 80
41      instance_type = "t2.micro"
42      region = "us-east-1"
43    }
44
45    provisioners {
46      shell {
47        inline = [
48          "apt-get update && apt-get install -y nginx"
49        ]
50      }
51    }
52  }
53
54  artifacts {
55    type = "archive"
56    path = "Nginx-Node"
57    name = "Nginx-Node"
58  }
59
60  post {
61    provisioners {
62      shell {
63        inline = [
64          "aws s3 cp ./Nginx-Node s3://$ami_name"
65        ]
66      }
67    }
68  }
69
70  network_interface {
71    interface_type = "eni"
72    primary_ip = "172.31.0.10"
73    primary_ip_nic_index = 0
74  }
75
76  tags {
77    Name = "Nginx-Node"
78  }
79
80  timeout {
81    build_timeout = "10m"
82  }
83
84  provisioners {
85    shell {
86      inline = [
87        "apt-get update && apt-get install -y nginx"
88      ]
89    }
90  }
91
92  artifacts {
93    type = "archive"
94    path = "Nginx-Node"
95    name = "Nginx-Node"
96  }
97
98  post {
99    provisioners {
100      shell {
101        inline = [
102          "aws s3 cp ./Nginx-Node s3://$ami_name"
103        ]
104      }
105    }
106  }
107
108  network_interface {
109    interface_type = "eni"
110    primary_ip = "172.31.0.10"
111    primary_ip_nic_index = 0
112  }
113
114  tags {
115    Name = "Nginx-Node"
116  }
117
118  timeout {
119    build_timeout = "10m"
120  }
121
122  provisioners {
123    shell {
124      inline = [
125        "apt-get update && apt-get install -y nginx"
126      ]
127    }
128  }
129
130  artifacts {
131    type = "archive"
132    path = "Nginx-Node"
133    name = "Nginx-Node"
134  }
135
136  post {
137    provisioners {
138      shell {
139        inline = [
140          "aws s3 cp ./Nginx-Node s3://$ami_name"
141        ]
142      }
143    }
144  }
145
146  network_interface {
147    interface_type = "eni"
148    primary_ip = "172.31.0.10"
149    primary_ip_nic_index = 0
150  }
151
152  tags {
153    Name = "Nginx-Node"
154  }
155
156  timeout {
157    build_timeout = "10m"
158  }
159
160  provisioners {
161    shell {
162      inline = [
163        "apt-get update && apt-get install -y nginx"
164      ]
165    }
166  }
167
168  artifacts {
169    type = "archive"
170    path = "Nginx-Node"
171    name = "Nginx-Node"
172  }
173
174  post {
175    provisioners {
176      shell {
177        inline = [
178          "aws s3 cp ./Nginx-Node s3://$ami_name"
179        ]
180      }
181    }
182  }
183
184  network_interface {
185    interface_type = "eni"
186    primary_ip = "172.31.0.10"
187    primary_ip_nic_index = 0
188  }
189
190  tags {
191    Name = "Nginx-Node"
192  }
193
194  timeout {
195    build_timeout = "10m"
196  }
197
198  provisioners {
199    shell {
200      inline = [
201        "apt-get update && apt-get install -y nginx"
202      ]
203    }
204  }
205
206  artifacts {
207    type = "archive"
208    path = "Nginx-Node"
209    name = "Nginx-Node"
210  }
211
212  post {
213    provisioners {
214      shell {
215        inline = [
216          "aws s3 cp ./Nginx-Node s3://$ami_name"
217        ]
218      }
219    }
220  }
221
222  network_interface {
223    interface_type = "eni"
224    primary_ip = "172.31.0.10"
225    primary_ip_nic_index = 0
226  }
227
228  tags {
229    Name = "Nginx-Node"
230  }
231
232  timeout {
233    build_timeout = "10m"
234  }
235
236  provisioners {
237    shell {
238      inline = [
239        "apt-get update && apt-get install -y nginx"
240      ]
241    }
242  }
243
244  artifacts {
245    type = "archive"
246    path = "Nginx-Node"
247    name = "Nginx-Node"
248  }
249
250  post {
251    provisioners {
252      shell {
253        inline = [
254          "aws s3 cp ./Nginx-Node s3://$ami_name"
255        ]
256      }
257    }
258  }
259
260  network_interface {
261    interface_type = "eni"
262    primary_ip = "172.31.0.10"
263    primary_ip_nic_index = 0
264  }
265
266  tags {
267    Name = "Nginx-Node"
268  }
269
270  timeout {
271    build_timeout = "10m"
272  }
273
274  provisioners {
275    shell {
276      inline = [
277        "apt-get update && apt-get install -y nginx"
278      ]
279    }
280  }
281
282  artifacts {
283    type = "archive"
284    path = "Nginx-Node"
285    name = "Nginx-Node"
286  }
287
288  post {
289    provisioners {
290      shell {
291        inline = [
292          "aws s3 cp ./Nginx-Node s3://$ami_name"
293        ]
294      }
295    }
296  }
297
298  network_interface {
299    interface_type = "eni"
300    primary_ip = "172.31.0.10"
301    primary_ip_nic_index = 0
302  }
303
304  tags {
305    Name = "Nginx-Node"
306  }
307
308  timeout {
309    build_timeout = "10m"
310  }
311
312  provisioners {
313    shell {
314      inline = [
315        "apt-get update && apt-get install -y nginx"
316      ]
317    }
318  }
319
320  artifacts {
321    type = "archive"
322    path = "Nginx-Node"
323    name = "Nginx-Node"
324  }
325
326  post {
327    provisioners {
328      shell {
329        inline = [
330          "aws s3 cp ./Nginx-Node s3://$ami_name"
331        ]
332      }
333    }
334  }
335
336  network_interface {
337    interface_type = "eni"
338    primary_ip = "172.31.0.10"
339    primary_ip_nic_index = 0
340  }
341
342  tags {
343    Name = "Nginx-Node"
344  }
345
346  timeout {
347    build_timeout = "10m"
348  }
349
350  provisioners {
351    shell {
352      inline = [
353        "apt-get update && apt-get install -y nginx"
354      ]
355    }
356  }
357
358  artifacts {
359    type = "archive"
360    path = "Nginx-Node"
361    name = "Nginx-Node"
362  }
363
364  post {
365    provisioners {
366      shell {
367        inline = [
368          "aws s3 cp ./Nginx-Node s3://$ami_name"
369        ]
370      }
371    }
372  }
373
374  network_interface {
375    interface_type = "eni"
376    primary_ip = "172.31.0.10"
377    primary_ip_nic_index = 0
378  }
379
380  tags {
381    Name = "Nginx-Node"
382  }
383
384  timeout {
385    build_timeout = "10m"
386  }
387
388  provisioners {
389    shell {
390      inline = [
391        "apt-get update && apt-get install -y nginx"
392      ]
393    }
394  }
395
396  artifacts {
397    type = "archive"
398    path = "Nginx-Node"
399    name = "Nginx-Node"
400  }
401
402  post {
403    provisioners {
404      shell {
405        inline = [
406          "aws s3 cp ./Nginx-Node s3://$ami_name"
407        ]
408      }
409    }
410  }
411
412  network_interface {
413    interface_type = "eni"
414    primary_ip = "172.31.0.10"
415    primary_ip_nic_index = 0
416  }
417
418  tags {
419    Name = "Nginx-Node"
420  }
421
422  timeout {
423    build_timeout = "10m"
424  }
425
426  provisioners {
427    shell {
428      inline = [
429        "apt-get update && apt-get install -y nginx"
430      ]
431    }
432  }
433
434  artifacts {
435    type = "archive"
436    path = "Nginx-Node"
437    name = "Nginx-Node"
438  }
439
440  post {
441    provisioners {
442      shell {
443        inline = [
444          "aws s3 cp ./Nginx-Node s3://$ami_name"
445        ]
446      }
447    }
448  }
449
450  network_interface {
451    interface_type = "eni"
452    primary_ip = "172.31.0.10"
453    primary_ip_nic_index = 0
454  }
455
456  tags {
457    Name = "Nginx-Node"
458  }
459
460  timeout {
461    build_timeout = "10m"
462  }
463
464  provisioners {
465    shell {
466      inline = [
467        "apt-get update && apt-get install -y nginx"
468      ]
469    }
470  }
471
472  artifacts {
473    type = "archive"
474    path = "Nginx-Node"
475    name = "Nginx-Node"
476  }
477
478  post {
479    provisioners {
480      shell {
481        inline = [
482          "aws s3 cp ./Nginx-Node s3://$ami_name"
483        ]
484      }
485    }
486  }
487
488  network_interface {
489    interface_type = "eni"
490    primary_ip = "172.31.0.10"
491    primary_ip_nic_index = 0
492  }
493
494  tags {
495    Name = "Nginx-Node"
496  }
497
498  timeout {
499    build_timeout = "10m"
500  }
501
502  provisioners {
503    shell {
504      inline = [
505        "apt-get update && apt-get install -y nginx"
506      ]
507    }
508  }
509
510  artifacts {
511    type = "archive"
512    path = "Nginx-Node"
513    name = "Nginx-Node"
514  }
515
516  post {
517    provisioners {
518      shell {
519        inline = [
520          "aws s3 cp ./Nginx-Node s3://$ami_name"
521        ]
522      }
523    }
524  }
525
526  network_interface {
527    interface_type = "eni"
528    primary_ip = "172.31.0.10"
529    primary_ip_nic_index = 0
530  }
531
532  tags {
533    Name = "Nginx-Node"
534  }
535
536  timeout {
537    build_timeout = "10m"
538  }
539
540  provisioners {
541    shell {
542      inline = [
543        "apt-get update && apt-get install -y nginx"
544      ]
545    }
546  }
547
548  artifacts {
549    type = "archive"
550    path = "Nginx-Node"
551    name = "Nginx-Node"
552  }
553
554  post {
555    provisioners {
556      shell {
557        inline = [
558          "aws s3 cp ./Nginx-Node s3://$ami_name"
559        ]
560      }
561    }
562  }
563
564  network_interface {
565    interface_type = "eni"
566    primary_ip = "172.31.0.10"
567    primary_ip_nic_index = 0
568  }
569
570  tags {
571    Name = "Nginx-Node"
572  }
573
574  timeout {
575    build_timeout = "10m"
576  }
577
578  provisioners {
579    shell {
580      inline = [
581        "apt-get update && apt-get install -y nginx"
582      ]
583    }
584  }
585
586  artifacts {
587    type = "archive"
588    path = "Nginx-Node"
589    name = "Nginx-Node"
590  }
591
592  post {
593    provisioners {
594      shell {
595        inline = [
596          "aws s3 cp ./Nginx-Node s3://$ami_name"
597        ]
598      }
599    }
600  }
601
602  network_interface {
603    interface_type = "eni"
604    primary_ip = "172.31.0.10"
605    primary_ip_nic_index = 0
606  }
607
608  tags {
609    Name = "Nginx-Node"
610  }
611
612  timeout {
613    build_timeout = "10m"
614  }
615
616  provisioners {
617    shell {
618      inline = [
619        "apt-get update && apt-get install -y nginx"
620      ]
621    }
622  }
623
624  artifacts {
625    type = "archive"
626    path = "Nginx-Node"
627    name = "Nginx-Node"
628  }
629
630  post {
631    provisioners {
632      shell {
633        inline = [
634          "aws s3 cp ./Nginx-Node s3://$ami_name"
635        ]
636      }
637    }
638  }
639
640  network_interface {
641    interface_type = "eni"
642    primary_ip = "172.31.0.10"
643    primary_ip_nic_index = 0
644  }
645
646  tags {
647    Name = "Nginx-Node"
648  }
649
650  timeout {
651    build_timeout = "10m"
652  }
653
654  provisioners {
655    shell {
656      inline = [
657        "apt-get update && apt-get install -y nginx"
658      ]
659    }
660  }
661
662  artifacts {
663    type = "archive"
664    path = "Nginx-Node"
665    name = "Nginx-Node"
666  }
667
668  post {
669    provisioners {
670      shell {
671        inline = [
672          "aws s3 cp ./Nginx-Node s3://$ami_name"
673        ]
674      }
675    }
676  }
677
678  network_interface {
679    interface_type = "eni"
680    primary_ip = "172.31.0.10"
681    primary_ip_nic_index = 0
682  }
683
684  tags {
685    Name = "Nginx-Node"
686  }
687
688  timeout {
689    build_timeout = "10m"
690  }
691
692  provisioners {
693    shell {
694      inline = [
695        "apt-get update && apt-get install -y nginx"
696      ]
697    }
698  }
699
700  artifacts {
701    type = "archive"
702    path = "Nginx-Node"
703    name = "Nginx-Node"
704  }
705
706  post {
707    provisioners {
708      shell {
709        inline = [
710          "aws s3 cp ./Nginx-Node s3://$ami_name"
711        ]
712      }
713    }
714  }
715
716  network_interface {
717    interface_type = "eni"
718    primary_ip = "172.31.0.10"
719    primary_ip_nic_index = 0
720  }
721
722  tags {
723    Name = "Nginx-Node"
724  }
725
726  timeout {
727    build_timeout = "10m"
728  }
729
730  provisioners {
731    shell {
732      inline = [
733        "apt-get update && apt-get install -y nginx"
734      ]
735    }
736  }
737
738  artifacts {
739    type = "archive"
740    path = "Nginx-Node"
741    name = "Nginx-Node"
742  }
743
744  post {
745    provisioners {
746      shell {
747        inline = [
748          "aws s3 cp ./Nginx-Node s3://$ami_name"
749        ]
750      }
751    }
752  }
753
754  network_interface {
755    interface_type = "eni"
756    primary_ip = "172.31.0.10"
757    primary_ip_nic_index = 0
758  }
759
760  tags {
761    Name = "Nginx-Node"
762  }
763
764  timeout {
765    build_timeout = "10m"
766  }
767
768  provisioners {
769    shell {
770      inline = [
771        "apt-get update && apt-get install -y nginx"
772      ]
773    }
774  }
775
776  artifacts {
777    type = "archive"
778    path = "Nginx-Node"
779    name = "Nginx-Node"
780  }
781
782  post {
783    provisioners {
784      shell {
785        inline = [
786          "aws s3 cp ./Nginx-Node s3://$ami_name"
787        ]
788      }
789    }
790  }
791
792  network_interface {
793    interface_type = "eni"
794    primary_ip = "172.31.0.10"
795    primary_ip_nic_index = 0
796  }
797
798  tags {
799    Name = "Nginx-Node"
800  }
801
802  timeout {
803    build_timeout = "10m"
804  }
805
806  provisioners {
807    shell {
808      inline = [
809        "apt-get update && apt-get install -y nginx"
810      ]
811    }
812  }
813
814  artifacts {
815    type = "archive"
816    path = "Nginx-Node"
817    name = "Nginx-Node"
818  }
819
820  post {
821    provisioners {
822      shell {
823        inline = [
824          "aws s3 cp ./Nginx-Node s3://$ami_name"
825        ]
826      }
827    }
828  }
829
830  network_interface {
831    interface_type = "eni"
832    primary_ip = "172.31.0.10"
833    primary_ip_nic_index = 0
834  }
835
836  tags {
837    Name = "Nginx-Node"
838  }
839
840  timeout {
841    build_timeout = "10m"
842  }
843
844  provisioners {
845    shell {
846      inline = [
847        "apt-get update && apt-get install -y nginx"
848      ]
849    }
850  }
851
852  artifacts {
853    type = "archive"
854    path = "Nginx-Node"
855    name = "Nginx-Node"
856  }
857
858  post {
859    provisioners {
860      shell {
861        inline = [
862          "aws s3 cp ./Nginx-Node s3://$ami_name"
863        ]
864      }
865    }
866  }
867
868  network_interface {
869    interface_type = "eni"
870    primary_ip = "172.31.0.10"
871    primary_ip_nic_index = 0
872  }
873
874  tags {
875    Name = "Nginx-Node"
876  }
877
878  timeout {
879    build_timeout = "10m"
880  }
881
882  provisioners {
883    shell {
884      inline = [
885        "apt-get update && apt-get install -y nginx"
886      ]
887    }
888  }
889
890  artifacts {
891    type = "archive"
892    path = "Nginx-Node"
893    name = "Nginx-Node"
894  }
895
896  post {
897    provisioners {
898      shell {
899        inline = [
900          "aws s3 cp ./Nginx-Node s3://$ami_name"
901        ]
902      }
903    }
904  }
905
906  network_interface {
907    interface_type = "eni"
908    primary_ip = "172.31.0.10"
909    primary_ip_nic_index = 0
910  }
911
912  tags {
913    Name = "Nginx-Node"
914  }
915
916  timeout {
917    build_timeout = "10m"
918  }
919
920  provisioners {
921    shell {
922      inline = [
923        "apt-get update && apt-get install -y nginx"
924      ]
925    }
926  }
927
928  artifacts {
929    type = "archive"
930    path = "Nginx-Node"
931    name = "Nginx-Node"
932  }
933
934  post {
935    provisioners {
936      shell {
937        inline = [
938          "aws s3 cp ./Nginx-Node s3://$ami_name"
939        ]
940      }
941    }
942  }
943
944  network_interface {
945    interface_type = "eni"
946    primary_ip = "172.31.0.10"
947    primary_ip_nic_index = 0
948  }
949
950  tags {
951    Name = "Nginx-Node"
952  }
953
954  timeout {
955    build_timeout = "10m"
956  }
957
958  provisioners {
959    shell {
960      inline = [
961        "apt-get update && apt-get install -y nginx"
962      ]
963    }
964  }
965
966  artifacts {
967    type = "archive"
968    path = "Nginx-Node"
969    name = "Nginx-Node"
970  }
971
972  post {
973    provisioners {
974      shell {
975        inline = [
976          "aws s3 cp ./Nginx-Node s3://$ami_name"
977        ]
978      }
979    }
980  }
981
982  network_interface {
983    interface_type = "eni"
984    primary_ip = "172.31.0.10"
985    primary_ip_nic_index = 0
986  }
987
988  tags {
989    Name = "Nginx-Node"
990  }
991
992  timeout {
993    build_timeout = "10m"
994  }
995
996  provisioners {
997    shell {
998      inline = [
999        "apt-get update && apt-get install -y nginx"
1000       ]
1001    }
1002  }
1003
1004  artifacts {
1005    type = "archive"
1006    path = "Nginx-Node"
1007    name = "Nginx-Node"
1008  }
1009
1010  post {
1011    provisioners {
1012      shell {
1013        inline = [
1014          "aws s3 cp ./Nginx-Node s3://$ami_name"
1015        ]
1016      }
1017    }
1018  }
1019
1020  network_interface {
1021    interface_type = "eni"
1022    primary_ip = "172.31.0.10"
1023    primary_ip_nic_index = 0
1024  }
1025
1026  tags {
1027    Name = "Nginx-Node"
1028  }
1029
1030  timeout {
1031    build_timeout = "10m"
1032  }
1033
1034  provisioners {
1035    shell {
1036      inline = [
1037        "apt-get update && apt-get install -y nginx"
1038      ]
1039    }
1040  }
1041
1042  artifacts {
1043    type = "archive"
1044    path = "Nginx-Node"
1045    name = "Nginx-Node"
1046  }
1047
1048  post {
1049    provisioners {
1050      shell {
1051        inline = [
1052          "aws s3 cp ./Nginx-Node s3://$ami_name"
1053        ]
1054      }
1055    }
1056  }
1057
1058  network_interface {
1059    interface_type = "eni"
1060    primary_ip = "172.31.0.10"
1061    primary_ip_nic_index = 0
1062  }
1063
1064  tags {
1065    Name = "Nginx-Node"
1066  }
1067
1068  timeout {
1069    build_timeout = "10m"
1070  }
1071
1072  provisioners {
1073    shell {
1074      inline = [
1075        "apt-get update && apt-get install -y nginx"
1076      ]
1077    }
1078  }
1079
1080  artifacts {
1081    type = "archive"
1082    path = "Nginx-Node"
1083    name = "Nginx-Node"
1084  }
1085
1086  post {
1087    provisioners {
1088      shell {
1089        inline = [
1090          "aws s3 cp ./Nginx-Node s3://$ami_name"
1091        ]
1092      }
1093    }
1094  }
1095
1096  network_interface {
1097    interface_type = "eni"
1098    primary_ip = "172.31.0.10"
1099    primary_ip_nic_index = 0
1100  }
1101
1102  tags {
1103    Name = "Nginx-Node"
1104  }
1105
1106  timeout {
1107    build_timeout = "10m"
1108  }
1109
1110  provisioners {
1111    shell {
1112      inline = [
1113        "apt-get update && apt-get install -y nginx"
1114      ]
1115    }
1116  }
1117
1118  artifacts {
1119    type = "archive"
1120    path = "Nginx-Node"
1121    name = "Nginx-Node"
1122  }
1123
1124  post {
1125    provisioners {
1126      shell {
1127        inline = [
1128          "aws s3 cp ./Nginx-Node s3://$ami_name"
1129        ]
1130      }
1131    }
1132  }
1133
1134  network_interface {
1135    interface_type = "eni"
1136    primary_ip = "172.31.0.10"
1137    primary_ip_nic_index = 0
1138  }
1139
1140  tags {
1141    Name = "Nginx-Node"
1142  }
1143
1144  timeout {
1145    build_timeout = "10m"
1146  }
1147
1148  provisioners {
1149    shell {
1150      inline = [
1151        "apt-get update && apt-get install -y nginx"
1152      ]
1153    }
1154  }
1155
1156  artifacts {
1157    type = "archive"
1158    path = "Nginx-Node"
1159    name = "Nginx-Node"
1160  }
1161
1162  post {
1163    provisioners {
1164      shell {
1165        inline = [
1166          "aws s3 cp ./Nginx-Node s3://$ami_name"
1167        ]
1168      }
1169    }
1170  }
1171
1172  network_interface {
1173    interface_type = "eni"
1174    primary_ip = "172.31.0.10"
1175    primary_ip_nic_index = 0
1176  }
1177
1178  tags {
1179    Name = "Nginx-Node"
1180  }
1181
1182  timeout {
1183    build_timeout = "10m"
1184  }
1185
1186  provisioners {
1187    shell {
1188      inline = [
1189        "apt-get update && apt-get install -y nginx"
1190      ]
1191    }
1192  }
1193
1194  artifacts {
1195    type = "archive"
1196    path = "Nginx-Node"
1197    name = "Nginx-Node"
1198  }
1199
1200  post {
1201    provisioners {
1202      shell {
1203        inline = [
1204          "aws s3 cp ./Nginx-Node s3://$ami_name"
1205        ]
1206      }
1207    }
1208  }
1209
1210  network_interface {
1211    interface_type = "eni"
1212    primary_ip = "172.31.0.10"
1213    primary_ip_nic_index = 0
1214  }
1215
1216  tags {
1217    Name = "Nginx-Node"
1218  }
1219
1220  timeout {
1221    build_timeout = "10m"
1222  }
1223
1224  provisioners {
1225    shell {
1226      inline = [
1227        "apt-get update && apt-get install -y nginx"
1228      ]
1229    }
1230  }
1231
1232  artifacts {
1233    type = "archive"
1234    path = "Nginx-Node"
1235    name = "Nginx-Node"
1236  }
1237
1238  post {
1239    provisioners {
1240      shell {
1241        inline = [
1242          "aws s3 cp ./Nginx-Node s3://$ami_name"
1243        ]
1244      }
1245    }
1246  }
1247
1248  network_interface {
1249    interface_type = "eni"
1250    primary_ip = "172.31.0.10"
1251    primary_ip_nic_index = 0
1252  }
1253
1254  tags {
1255    Name = "Nginx-Node"
1256  }
1257
1258  timeout {
1259    build_timeout = "10m"
1260  }
1261
1262  provisioners {
1263    shell {
1264      inline = [
1265        "apt-get update && apt-get install -y nginx"
1266      ]
1267    }
1268  }
1269
1270  artifacts {
1271    type = "archive"
1272    path = "Nginx-Node"
1273    name = "Nginx-Node"
1274  }
1275
1276  post {
1277    provisioners {
1278      shell {
1279        inline = [
1280          "aws s3 cp ./Nginx-Node s3://$ami_name"
1281        ]
1282      }
1283    }
1284  }
1285
1286  network_interface {
1287    interface_type = "eni
```