

Despliegue de Nginx y Nodejs mediante una imagen generada por Packer



Herramientas DevOps

JHOSTYN GABRIEL BENALCAZAR IBUJES
DICIEMBRE 2023

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Benalcázar Ibujés	18/12/2023
	Nombre: Jhostyn Gabriel	

Se indicará el procedimiento efectuado para aprovisionar a la plataforma de servicios en la nube AWS (Amazon Web Services), dando cumplimiento a la actividad a través de packer.

Como primer paso se define una plantilla que conlleva la estructura observada en clase:

- **Variables;** Se albergará la información genérica para los builders, en este caso se harán uso de las siguientes variables:
 - **aws_access_key:** Clave de acceso generado en AWS.
 - **aws_secret_key:** Clave secreta generada en AWS.
 - **aws_region:** Región en donde se montará la máquina virtual, para verificar las regiones y escoger la mejor que convenga se hizo uso del siguiente enlace.
 - https://aws.amazon.com/es/about-aws/global-infrastructure/regions_az/
 - **image_name:** El nombre identificado con el que se creará la imagen.
 - **username:** Es el nombre usuario por el medio que se podrá conectar a ssh.
- **builders:** En este caso ejecuta el proceso para la construcción de la AMI (Amazon Machine Image), en este caso es el proceso de construcción de la imagen, tomando en cuenta los atributos más importantes.
 - **Nota:** Cabe mencionar que para que ejecute correctamente el builder se debe tener instalado con anterioridad el plugin de amazon-ebs.
 - <https://developer.hashicorp.com/packer/integrations/hashicorp/amazon>
- **provisioners:** Este apartado es importante ya que será el encargado de instalar el nodejs y la puesta en marcha del proyecto con la palabra Hola Mundo.

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Benalcázar Ibujés	18/12/2023
	Nombre: Jhostyn Gabriel	

- **post-processors:** Es un atributo adicional y ejecuta después de haber creado la imagen, en este caso se ejecutarán dos, uno que muestre un resumen de la construcción y el otro que genere automáticamente la instancia de la imagen.
 - **Nota:** El segundo post procesor es el más importante debido a que es el necesario para que la instancia sea generada automáticamente, cabe mencionar que depende del resultado del manifest ya que ahí se encuentra la información del `artifact_id` que representa el identificador de la imagen.

A continuación, se muestra el código de lo explicado:

```
{
  "variables": {
    "aws_access_key": "AKIA5PALQOZEBYKRIZBI",
    "aws_secret_key": "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",
    "aws_region": "us-east-1",
    "image_naming": "tarea1-packer-node",
    "username": "ubuntu"
  },
  "builders": [
    {
      "type": "amazon-ebs",
      "access_key": "{{ user `aws_access_key` }}",
      "secret_key": "{{ user `aws_secret_key` }}",
      "source_ami_filter": {
        "filters": {
          "virtualization-type": "hvm",
          "name": "ubuntu/images/*ubuntu-bionic-18.04-amd64-server-*",
          "root-device-type": "ebs"
        },
        "owners": [
          "099720109477"
        ],
        "most_recent": true
      },
      "ami_name": "{{ user `image_naming` }}-{{ timestamp }}",
      "instance_type": "t2.small",
      "ssh_username": "{{ user `username` }}",
      "region": "{{ user `aws_region` }}"
    }
  ]
}
```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Benalcázar Ibujés	18/12/2023
	Nombre: Jhostyn Gabriel	

```

    }
  ],
  "provisioners": [
    {
      "type": "file",
      "source": "hello_world.js",
      "destination": "/tmp/hello_world.js"
    },
    {
      "type": "shell",
      "script": "node_implementation.sh"
    }
  ],
  "post-processors": [
    {
      "type": "manifest",
      "output": "output.json"
    },
    {
      "type": "shell-local",
      "inline": [
        "@echo off",
        "for /f \"%tokens=3 delims=\\\" %%a in ('type output.json",
        "^| find /i \"%artifact_id\\\"') do set ami_id=%%a",
        "set ami_id=%ami_id:~0,-2%",
        "echo AMI ID: %ami_id%",
        "echo aws ec2 run-instances --image-id %ami_id% --",
        "instance-type t2.micro --key-name my-key-pair --subnet-id subnet-",
        "020400e85546a033f --security-group-ids sg-087634422d4c635ea",
        "aws ec2 run-instances --image-id \"%ami_id%\" --",
        "instance-type t2.micro --key-name my-key-pair --subnet-id subnet-",
        "020400e85546a033f --security-group-ids sg-087634422d4c635ea"
      ]
    }
  ]
}

```

El aprovisionamiento del archivo .js consta de un código simple javascript en dónde crea un server local y la impresión del Hola Mundo.

```

var http = require("http");
var port = 8080;
var now = new Date();
http

```

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Benalcázar Ibujés	18/12/2023
	Nombre: Jhostyn Gabriel	

```
.createServer(function (req, res) {
    res.writeHead(200, { "Content-Type": "text/plain" });
    res.write("Hola Mundo DevOps - Saluda Jhostyn Benalcázar\n");
    res.write("Fecha del servidor: " + now.toGMTString());
    res.end("\nTarea 1: Despliegue de Nginx y Nodejs mediante una
imagen generada por Packer.");
})
.listen(port, "");
console.log("Server running at port: " + port);
```

Por otra parte, en el archivo .sh se ejecuta la instalación de las herramientas necesarias en este caso node y también pm2 cual es un administrador de aplicaciones node.

```
#!/bin/sh

# Install node js
cd ~
curl -sL https://deb.nodesource.com/setup_14.x -o nodesource_setup.sh

sudo bash nodesource_setup.sh

sudo apt install nodejs

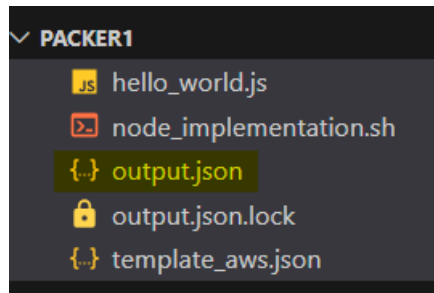
# Install pm2
sudo npm install -g pm2

# Configure pm2 to run Hello World on startup
mkdir -p ~/code/app-dist
mv /tmp/hello_world.js ~/code/app-dist/hello_world.js
cd ~/code/app-dist/
sudo pm2 start hello_world.js
sudo pm2 startup systemd
sudo pm2 save
sudo pm2 list
```

Antes de hacer la ejecución del packer, se procederá a hacer un validate, en donde demostrará si la configuración es la correcta:

```
● PS C:\Users\Jhostyn\Documents\Maestría\Administración de sistemas Cloud\Prácticas\Packer1> packer validate template_aws.json
The configuration is valid.
○ PS C:\Users\Jhostyn\Documents\Maestría\Administración de sistemas Cloud\Prácticas\Packer1> 
```

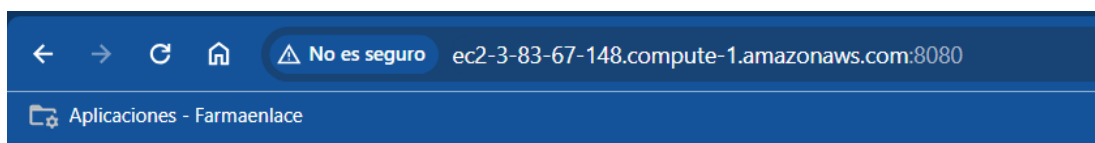

Asignatura	Datos del alumno	Fecha
Herramientas DevOps	Apellidos: Benalcázar Ibujés	18/12/2023
	Nombre: Jhostyn Gabriel	



```
{
  "builds": [
    {
      "name": "amazon-ebs",
      "builder_type": "amazon-ebs",
      "build_time": 1702878772,
      "files": null,
      "artifact_id": "us-east-1:ami-086a9b13547b26433",
      "packer_run_uuid": "40003b41-883e-0eb4-b46e-0f39b5dad78c",
      "custom_data": null
    }
  ],
  "last_run_uuid": "40003b41-883e-0eb4-b46e-0f39b5dad78c"
}
```

De ser todo el proceso correcto se procederá a revisar la instancia generada y se ingresa al puerto designado para validar el archivo generado:

- <http://ec2-3-83-67-148.compute-1.amazonaws.com:8080/>



Hola Mundo DevOps - Saluda Jhostyn Benalcázar
 Fecha del servidor: Mon, 18 Dec 2023 21:41:47 GMT
 Tarea 1: Despliegue de Nginx y Nodejs mediante una imagen generada por Packer.