

Universidad Tecnológica de la Riviera Maya

Instalación de servicios web

Desarrollo Web Integral

Maestro:

José Rafael Pino Rusconi Chio

Alumna:

Rut Chulin Delgado

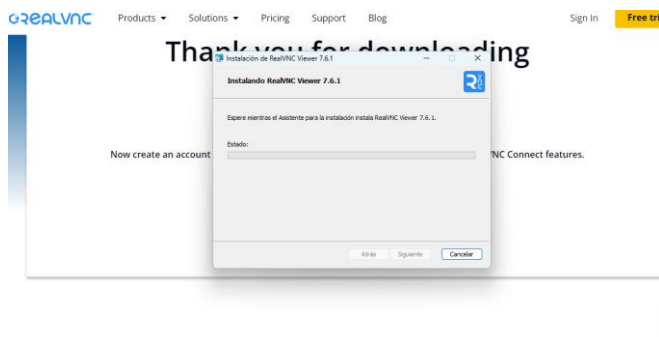
TI-91

Fecha:

25/09/2023

Introducción

Las máquinas virtuales son sistemas operativos que se almacenan dentro de un servidor, lo que hace es comportarse como si fuera una computadora real, la ventaja de esto es que podemos probar diferentes sistemas desde una sola computadora ya que estas MV podemos personalizarlas como queramos a un costo mucho más bajo que comprar una, por tal motivo se realizamos la instalación de una máquina virtual utilizando los servidores de Oracle utilizando estas herramientas se realizó la instalación de node.js la cual nos permite ejecutar nuestras aplicaciones.



Crear una instancia



Conexión a la máquina virtual

Public IP: 159.54.136.213

```

utch@DESKTOP-KSU42L3 MINGW64 ~
ssh ubuntu@140.84.167.246
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1040-oracle aarch64)

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

System information as of Mon Sep 25 19:54:16 UTC 2023

System load: 0.29          Processes:             149
Usage of /:   3.4% of 44.96GB Users logged in:              0
Memory usage: 0%          IPv4 address for enp0s6: 10.0.0.17
Swap usage:   0%

```

Instalar Node js

- Actualizaciones generales de Ubuntu

```

ubuntu@instance-desarrolloweb:~$ sudo apt update
Hit:1 http://mx-queretaro-1-ad-1.clouds.ports.ubuntu.com/ubuntu-ports focal InRelease
Get:2 http://ports.ubuntu.com/ubuntu-ports focal-security InRelease [114 kB]
Get:3 http://mx-queretaro-1-ad-1.clouds.ports.ubuntu.com/ubuntu-ports focal-updates InRelease [114 kB]
Get:4 http://mx-queretaro-1-ad-1.clouds.ports.ubuntu.com/ubuntu-ports focal-backports InRelease [108 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports focal-security/main arm64 Packages [126 kB]
Get:6 http://mx-queretaro-1-ad-1.clouds.ports.ubuntu.com/ubuntu-ports focal/universe arm64 Packages [8458 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports focal-security/main Translation-en [35 kB]

```

- Instalar

```

12 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@instance-desarrolloweb:~$ sudo apt install nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libc-ares2 libnode64 nodejs-doc
Suggested packages:
  npm
The following NEW packages will be installed:

```

Crear el servidor web

Crear una carpeta e ir a la ruta

```

1.0.19.0
ubuntu@instance-desarrolloweb:~$ mkdir webserver
ubuntu@instance-desarrolloweb:~$ cd webserver/

```

npm init

```

ubuntu@instance-desarrolloweb:~/webserver$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help json' for definitive documentation on these fields
and exactly what they do.

Use 'npm install <pkg>' afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (webserver) |
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/ubuntu/webserver/package.json:

{
  "name": "webserver",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes)

```

Creación del archivo index.js

```
ubuntu@instance-desarrolloweb: ~/webserver
GNU nano 4.8 index.js
const http = require('http');
hostname = '0.0.0.0';
const port = 3000;
const server = http.createServer((req, res) => {
  var currentDate = new Date();
  var utcDate = currentDate.toUTCString();
  res.statusCode = 200;
  res.setHeader('Content-type', 'text/plain');
  if(req.method === 'GET' && req.url === '/greeting'){
    //This is the Hello World inovation
    res.end('Hello World\n');
    console.log('Hello World invoked ' + utcDate);
  } else {
    //This is the health check from the load balancer
    res.end('Healthy\n');
    //console.log('Health check ' + utcDate);
  }
});server.listen(port, hostname, () => {
  console.log('Server running at http://${hostname}:');
});
```

Correr el código

```
ubuntu@instance-desarrolloweb:~/webserver$ node index.js
Server running at http://0.0.0.0:3000/
```

Test el servidor web

Crear una nueva conexión

```
Last login: Mon Sep 25 19:54:17 2023 from 189.176.92.69
ubuntu@instance-desarrolloweb:~$ curl http://0.0.0.0:3000
Healthy
ubuntu@instance-desarrolloweb:~$
```

Configuración del Firewall

Instalar firewall

```
ubuntu@instance-desarrolloweb:~$ sudo apt install firewall
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  gir1.2-nm-1.0 ipset libipset13 libjansson4 libnftables1 libnm0 python3-decorator
  python3-firewall python3-nftables python3-selinux python3-slip python3-slip-dbus
The following NEW packages will be installed:
  firewall gir1.2-nm-1.0 ipset libipset13 libjansson4 libnftables1 libnm0 python3-de
  python3-firewall python3-nftables python3-selinux python3-slip python3-slip-dbus
```

```
ubuntu@instance-desarrolloweb:~$ sudo systemctl start firewalld
ubuntu@instance-desarrolloweb:~$ sudo firewall-cmd --state
running
ubuntu@instance-desarrolloweb:~$ |
```

Abrir el puerto 3000 permanentemente

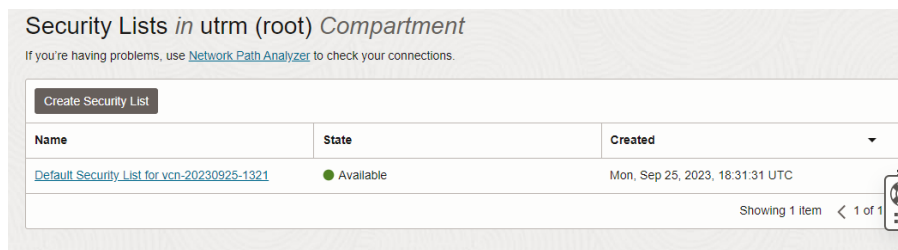
```
ubuntu@instance-desarrolloweb:~$ sudo firewall-cmd --permanent --add-port=3000/tcp
success
ubuntu@instance-desarrolloweb:~$ sudo firewall-cmd --reload
success
```

Creación de regla sin estado

Clic en virtual cloud network



Clic en security list



Ingresar regla

Add Ingress Rules

To enable bidirectional traffic flow, make sure a complementary rule in the opposite direction exists.

Source Type: CIDR

Source CIDR: 0.0.0.0/0

IP Protocol: TCP

Source Port Range: Optional (All)

Destination Port Range: Optional (3000)

Description: Optional

Maximum 255 characters

+ Another Ingress Rule

Add Ingress Rules Cancel

Probar la conexión a la IP publica

```
ubuntu@instance-desarrolloweb:~$ curl http://140.84.167.246:3000
Healthy
```

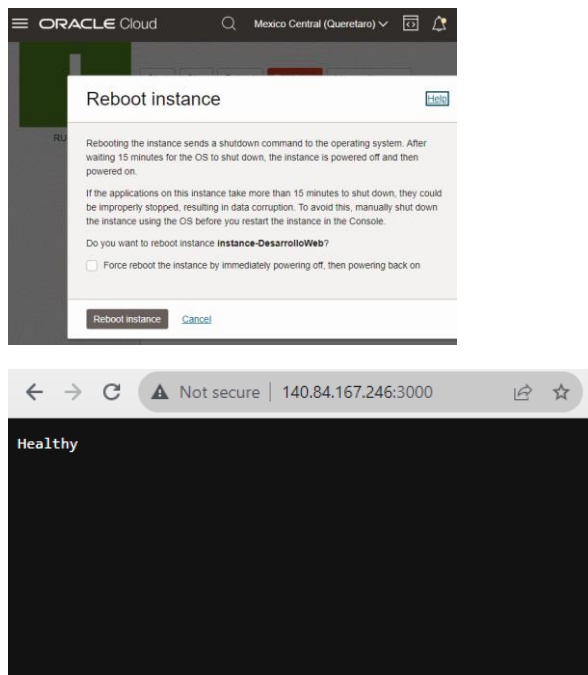
Configuración del servidor web para auto iniciar

Script startwebservice.service

```
GNU nano 4.8 startwebservice.service
[Unit]
Description=Start the Node webserver web server on startup
[Service]
Type=oneshot
ExecStart=/usr/bin/node /home/ubuntu/webserver/index.js
StandardOutput=syslog
StandardError=syslog
RemainAfterExit=yes
[Install]
WantedBy=multi-user.target

ubuntu@instance-desarrolloweb:/etc/systemd/system$ sudo nano startwebservice.service
ubuntu@instance-desarrolloweb:/etc/systemd/system$ sudo systemctl enable startwebservice.service
Created symlink /etc/systemd/system/multi-user.target.wants/startwebservice.service + /etc/systemd/system/startwebservice.service.
```

Revision que el script este funcionando desde OCI Tenancy Console



Arquitectura realizada

En esta practica se utiliza la arquitectura cliente-servidor en donde el servidor en este caso seria la maquina virtual que se creó, en la cual se instaló node.js siendo el que brinda el entorno para crear aplicaciones web, la pagina web se almacena dentro de la instancia en la cual al conectarnos al navegador nos convierte en el cliente para realizar una solicitud al servidor.

Podemos ver que esta arquitectura el servidor es quien permite a los clientes conectarse, permitiendo una comunicación continua.

Conclusión

Durante esta práctica se lograron desarrollar conocimientos como la creación de una instancia para instalar un servicio web, utilizando node.js como entorno de comunicación con el cliente, de igual forma se implantó el firewall para mantener la seguridad de la red privada ante cualquier ataque. Con esto se pudo entender de una manera practica como es realizar la configuración de una aplicación para la red.

Bibliografía

Kaira. (2023, March 27). *Arquitectura de software: Definición, elementos y tipos*. Canvia.

<https://canvia.com/arquitectura-software/>

Kinsta. (2023, June 15). *Qué es Node.js y por qué deberías usarlo*. Kinsta®.

<https://kinsta.com/es/base-de-conocimiento/que-es-node-js/>

Schiaffarino, A., & Schiaffarino, A. (2019). Modelo cliente servidor. *Infranetworking*.

<https://blog.infranetworking.com/modelo-cliente-servidor/>