Nombre: Victor Manuel Gomez

Parcial final servicios telemáticos

parte 1:

para esta parte utilizaré el siguiente vagrantfile

VagrantFile:

```
Vagrant.configure("2") do |config|
 if Vagrant.has_plugin? "vagrant-vbguest"
 config.vbguest.no_install = true
 config.vbguest.auto_update = false
 config.vbguest.no_remote = true
 end
 config.vm.define :cliente do |cliente|
 cliente.vm.box = "generic/centos9s"
 cliente.vm.network :private_network, ip: "192.168.50.2"
 cliente.vm.hostname = "cliente"
 end
 config.vm.define :servidor do |servidor|
 servidor.vm.box = "generic/centos9s"
 servidor.vm.network :private_network, ip: "192.168.50.3"
 servidor.vm.hostname = "servidor"
 end
end
```

Instalación Prometheus en maquina centos9

paso a paso:

1. sudo apt update

2. sudo groupadd --system prometheus

3. sudo useradd -s /sbin/nologin --system -g prometheus prometheus

4. sudo mkdir /etc/prometheus

5. sudo mkdir /var/lib/prometheus

6. wget https://github.com/prometheus/prometheus/releases/download/v2.43.0/prometheus-2.43.0.linux-amd64.tar.gz

7. tar vxf prometheus*.tar.gz

Nombre: Victor Manuel Gomez

8. cd prometheus*/

9. sudo mv prometheus /usr/local/bin

10. sudo mv promtool /usr/local/bin

11. sudo chown prometheus:prometheus /usr/local/bin/prometheus

12. sudo chown prometheus:prometheus /usr/local/bin/promtool

13. sudo mv consoles /etc/prometheus

14. sudo mv console_libraries /etc/prometheus

15. sudo mv prometheus.yml /etc/prometheus

16. sudo chown prometheus:prometheus /etc/prometheus

17. sudo chown -R prometheus:prometheus /etc/prometheus/consoles

18. sudo chown -R prometheus:prometheus /etc/prometheus/console_libraries

19. sudo chown -R prometheus:prometheus /var/lib/prometheus

20. sudo vi /etc/systemd/system/prometheus.service

dentro de este archivo debe ir el siguiente contenido:

```
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/prometheus \
    --config.file /etc/prometheus/prometheus.yml \
    --storage.tsdb.path /var/lib/prometheus/ \
    --web.console.templates=/etc/prometheus/consoles \
    --web.console.libraries=/etc/prometheus/console_libraries

[Install]
WantedBy=multi-user.target
```

Nombre: Victor Manuel Gomez

21. sudo systemctl daemon-reload

22. sudo systemctl enable prometheus

23. sudo systemctl start prometheus

24. sudo systemctl status prometheus

25. sudo firewall-cmd --zone=public --add-port=9090/tcp --permanent

26. sudo firewall-cmd --get-zones

27. sudo firewall-cmd --reload

28. abrimos nuestro navegador y nos dirigimos a
<ip_address de nuestra maquina servidor>:9090

en mi caso es la http://192.168.50.3:9090

Instalacion y configuracion Node Exporter

1. wget https://github.com/prometheus/node_exporter/releases/download/v1.7.0/node_exporter-1.7.0.linux-amd64.tar.gz

2. tar xvfz node_exporter-*.*-amd64.tar.gz

3. cd node_exporter-*.*-amd64

4. ./node_exporter

5. curl http://localhost:9100/metrics

6. curl http://localhost:9100/metrics | grep "node_"

7. añadimos las siguientes líneas a nuestro archivo prometheus.yml

```
scrape_configs:
- job_name: node
  static_configs:
  - targets: ['localhost:9100']
```

8. sudo systemctl restart prometheus

9. sudo systemctl status prometheus

Nombre: Victor Manuel Gomez

parte 2:

1. Maquina anfitrion:

git clone https://github.com/omondragon/APIRestFlaskMySQLUbuntu

Instalación de Node.js en Ubuntu

2. sudo apt update

3. sudo apt install nodejs npm

4. Verificar instalación:
   node -v
   npm -v

5. npm install express mysql body-parser

Accede a MySQL

1. mysql -u root -p

2. ALTER USER 'victorgomez'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Autonoma123*';

3. GRANT ALL PRIVILEGES ON *.* TO 'victorgomez'@'localhost' WITH GRANT OPTION;

4. exit;

código nodejs

contenido archivo app.js

```
const express = require('express');
const mysql = require('mysql');
const bodyParser = require('body-parser');

const app = express();
const port = 3000;

// MySQL Connection
const db = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: 'root',
  database: 'myflaskapp'
});

db.connect((err) => {
  if (err) {
    throw err;
  }
  console.log('MySQL connected');
});

// Middleware
app.use(bodyParser.json());

// Get all books
app.get('/books', (req, res) => {
  const query = 'SELECT * FROM books';
  db.query(query, (err, result) => {
    if (err) throw err;
    res.json({ books: result });
  });
});

// Get one book by id
app.get('/books/:book_id', (req, res) => {
  const bookId = req.params.book_id;
  const query = `SELECT * FROM books WHERE id = ${bookId}`;
  db.query(query, (err, result) => {
    if (err) throw err;
    res.json({ book: result[0] });
  });
});

// Add new book
app.post('/books', (req, res) => {
  const { title, description, author } = req.body;
  const query = 'INSERT INTO books (title, description, author) VALUES (?, ?, ?)';
  db.query(query, [title, description, author], (err, result) => {
    if (err) throw err;
    res.status(201).json({ book: req.body });
  });
});

// Edit a Book
app.put('/books/:book_id', (req, res) => {
  const bookId = req.params.book_id;
  const queryGet = `SELECT * FROM books WHERE id = ${bookId}`;
  db.query(queryGet, (err, result) => {
    if (err) throw err;
    const book = result[0];
    const { title, description, author } = req.body;
    const queryUpdate = 'UPDATE books SET title=?, description=?, author=? WHERE id=?';
    db.query(queryUpdate, [title || book.title, description || book.description, author || book.author, bookId], (err, result) => {
      if (err) throw err;
      res.json({ book: { id: bookId, title, description, author } });
    });
  });
});

// Delete a Book
app.delete('/books/:book_id', (req, res) => {
  const bookId = req.params.book_id;
  const query = `DELETE FROM books WHERE id = ${bookId}`;
  db.query(query, (err, result) => {
    if (err) throw err;
    res.json({ result: true });
  });
});

// Start the server
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

ahora por ultimo ejecutamos

```
node app.js
```