

Página de Estado Personal

Parece que la versión de `vnstat` que instalaste no soporta el parámetro `-u`. Continuemos con los pasos correctos para configurar `vnstat` y configurar la API de monitoreo de ancho de banda.

Paso 1: Configurar `vnstat`

Inicializa `vnstat` para tu interfaz de red:

```
sudo vnstat -i eth0
```

Nota: El comando no necesita traducción, ya que es una instrucción en la terminal. Sin embargo, si necesitas una explicación en español, aquí está:

Este comando se utiliza para mostrar estadísticas de uso de red en la interfaz `eth0` utilizando la herramienta `vnstat`. El prefijo `sudo` indica que el comando se ejecuta con privilegios de superusuario.

Paso 2: Esperar la recopilación de datos

`vnstat` necesita tiempo para recopilar datos. Verifica el estado periódicamente:

```
sudo vnstat -l
```

Después de algún tiempo, verifica la recopilación de datos:

```
sudo vnstat -d
```

Paso 3: Crear una API para exponer los datos de ancho de banda

Instalar Flask:

```
pip install Flask
```

Crea un script en Python (`bandwidth_api.py`):

```
from flask import Flask, jsonify
from flask_cors import CORS
import subprocess
```

```

app = Flask(__name__)
CORS(app) # Habilitar CORS para todas las rutas

@app.route('/bandwidth', methods=['GET']) def get_bandwidth(): # Ejecuta el comando vnstat para obtener las estadísticas de tráfico en intervalos de 5 minutos para eth0 result = subprocess.run(['vnstat', '-i', 'eth0', '-5', '-json'], capture_output=True, text=True) data = result.stdout

# Devolver los datos capturados como una respuesta JSON
return jsonify(data)

if name == 'main': app.run(host='0.0.0.0', port=5000)

```

(No hay texto para traducir dentro del bloque de código. El bloque de código se mantiene igual en español)

Ejecuta el script:

```

```sh
python bandwidth_api.py

```

*Nota: Los comandos en la terminal no se traducen, ya que son instrucciones específicas que deben ejecutarse tal cual en el sistema.*

#### Paso 4: Integrar con tu Blog

Utiliza el siguiente código HTML y JavaScript para obtener y mostrar los datos de ancho de banda:

```

document.addEventListener('DOMContentLoaded', function () {
 fetch('https://www.lzwjava.xyz/bandwidth')
 .then(response => response.json())
 .then(data => {
 var bandwidthData = JSON.parse(data);

 // Crear un contenedor para los tiempos
 var timesContainer = document.createElement('div');

```

```

var currentUtcTime = new Date();
var currentLocalTime = new Date(currentUtcTime.getTime());

var pUtcTime = document.createElement('p');
pUtcTime.textContent = `Hora UTC: ${currentUtcTime.toUTCString()}`;
timesContainer.appendChild(pUtcTime);

var pLocalTime = document.createElement('p');
pLocalTime.textContent = `Mi Hora Local: ${currentLocalTime.toString()}`;
timesContainer.appendChild(pLocalTime);

// Agregar el contenedor de tiempos al div de estado
document.getElementById('status').appendChild(timesContainer);

// Crear una tabla

var table = document.createElement('table');
table.border = '1';
table.style.borderCollapse = 'collapse';
table.style.width = '100%';

// Crear la cabecera de la tabla
var thead = document.createElement('thead');
var tr = document.createElement('tr');
var headers = ['Tiempo', 'Tráfico (KB/s)', 'Estado'];
headers.forEach(headerText => {
 var th = document.createElement('th');
 th.textContent = headerText;
 tr.appendChild(th);
});
thead.appendChild(tr);
table.appendChild(thead);

// Crear el cuerpo de la tabla
var tbody = document.createElement('tbody');

// Procesar datos de tráfico
var fiveMinuteData = bandwidthData.interfaces[0].traffic.fiveminute.reverse();

```

```

 fiveMinuteData.forEach(interval => {
 var tr = document.createElement('tr');

 var dataTime = new Date(Date.UTC(interval.date.year, interval.date.month - 1, interval.date.day, interval.date.hours, interval.date.minutes));
 var timeDifference = Math.round((currentUtcTime - dataTime) / (1000 * 60)); // Diferencia de tiempo en minutos

 var tdTimeDifference = document.createElement('td');
 tdTimeDifference.textContent = timeDifference + ' mins ago';
 tr.appendChild(tdTimeDifference);

 var averageTraffic = (interval.rx + interval.tx) / 2; // Calcular el promedio de RX y TX
 var tdTrafficKBs = document.createElement('td');
 var trafficKBs = (averageTraffic / (5 * 60 * 1024)).toFixed(2); // Convertir a KB/s
 tdTrafficKBs.textContent = trafficKBs;
 tr.appendChild(tdTrafficKBs);

 var tdStatus = document.createElement('td');
 tdStatus.textContent = trafficKBs > 5 ? 'Online' : 'Offline';
 tdStatus.className = trafficKBs > 5 ? 'status-online' : 'status-offline';
 tr.appendChild(tdStatus);

 tbody.appendChild(tr);
 });
 table.appendChild(tbody);

 // Añadir la tabla al div de estado
 document.getElementById('status').appendChild(table);
})

.catch(error => {
 console.error('Error al obtener los datos de ancho de banda:', error);
});

"""

Reemplaza http://your-droplet-ip:5000/bandwidth con la dirección IP de tu droplet.

```

## Consideraciones Adicionales

- **Seguridad:** Asegúrate de que tu API sea segura. Considera agregar autenticación.

- **Rendimiento:** Monitorear el ancho de banda puede ser intensivo en recursos. Asegúrate de que tu droplet tenga suficientes recursos.
- **Fiabilidad:** Agrega manejo de errores y lógica de reintento para gestionar la indisponibilidad de la API.

Siguiendo estos pasos, puedes crear una página de estado en tu blog que indique si estás en línea basándose en el uso de ancho de banda de tu droplet en DigitalOcean.