

Persönliche Statusseite

Es scheint, dass die Version von `vnstat`, die Sie installiert haben, den Parameter `-u` nicht unterstützt. Lassen Sie uns mit den richtigen Schritten fortfahren, um `vnstat` zu konfigurieren und die Bandbreitenüberwachungs-API einzurichten.

Schritt 1: Konfigurieren von `vnstat`

Initialisieren Sie `vnstat` für Ihre Netzwerkschnittstelle:

```
sudo vnstat -i eth0
```

Schritt 2: Warten auf die Datensammlung

`vnstat` benötigt Zeit, um Daten zu sammeln. Überprüfen Sie den Status regelmäßig:

```
sudo vnstat -l
```

Nach einiger Zeit die Datenerfassung überprüfen:

```
sudo vnstat -d
```

Schritt 3: Erstellen einer API zur Bereitstellung von Bandbreitendaten

Installiere Flask:

```
pip install Flask
```

Erstelle ein Python-Skript (`bandwidth_api.py`):

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

app = Flask(__name__)
CORS(app) # CORS für alle Routen aktivieren
```

```
@app.route('/bandwidth', methods=['GET']) def get_bandwidth(): # Führe den vnstat-Befehl
aus, um die Verkehrsstatistiken im 5-Minuten-Intervall für eth0 zu erhalten result = sub-
process.run(['vnstat', '-i', 'eth0', '-5', '-json'], capture_output=True, text=True) data =
result.stdout
```

```
# Gib die erfassten Daten als JSON-Antwort zurück
return jsonify(data)
```

```
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

Der Codeblock bleibt unverändert, da es sich um eine spezielle Syntax handelt, die nicht übersetzt werden muss.

Führen Sie das Skript aus:

```
```sh
python bandwidth_api.py
```

## Schritt 4: Integration in Ihren Blog

Verwenden Sie das folgende HTML und JavaScript, um die Bandbreitendaten abzurufen und anzuzeigen:

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

 // Erstelle einen Container für die Zeiten
 var timesContainer = document.createElement('div');

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

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

```

timesContainer.appendChild(pUtcTime);

var pLocalTime = document.createElement('p');
pLocalTime.textContent = `Meine lokale Zeit: ${currentLocalTime.toString()}`;
timesContainer.appendChild(pLocalTime);

// Hänge den Zeiten-Container an das Status-Div an
document.getElementById('status').appendChild(timesContainer);

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

// Tabellenkopf erstellen
var thead = document.createElement('thead');
var tr = document.createElement('tr');
var headers = ['Zeit', 'Datenverkehr (KB/s)', 'Status'];
headers.forEach(headerText => {
 var th = document.createElement('th');
 th.textContent = headerText;
 tr.appendChild(th);
});
thead.appendChild(tr);
table.appendChild(thead);

// Tabellenkörper erstellen
var tbody = document.createElement('tbody');

// Verkehrsdaten verarbeiten
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

```

```

 var timeDifference = Math.round((currentUtcTime - dateTime) / (1000 * 60)); // Zeituntersch

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

 var averageTraffic = (interval.rx + interval.tx) / 2; // Durchschnitt von RX und TX berechne
 var tdTrafficKBs = document.createElement('td');
 var trafficKBs = (averageTraffic / (5 * 60 * 1024)).toFixed(2); // In KB/s umrechnen
 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);

// Füge die Tabelle zum Status-Div hinzu
document.getElementById('status').appendChild(table);
})
.catch(error => {
 console.error('Fehler beim Abrufen der Bandbreitendaten:', error);
});

});

```

“ Der Codeblock bleibt unverändert, da es sich um eine spezielle Formatierung handelt, die nicht übersetzt werden sollte.

Ersetzen Sie `http://your-droplet-ip:5000/bandwidth` mit der IP-Adresse Ihres Droplets.

## Weitere Überlegungen

- **Sicherheit:** Stellen Sie sicher, dass Ihre API sicher ist. Erwägen Sie die Hinzufügung von Authentifizierung.

- **Leistung:** Die Überwachung der Bandbreite kann ressourcenintensiv sein. Stellen Sie sicher, dass Ihr Droplet über ausreichend Ressourcen verfügt.
- **Zuverlässigkeit:** Fügen Sie Fehlerbehandlung und Wiederholungslogik hinzu, um API-Ausfälle zu bewältigen.

Indem Sie diese Schritte befolgen, können Sie eine Statusseite auf Ihrem Blog erstellen, die anhand der Bandbreitennutzung Ihres DigitalOcean-Droplets anzeigt, ob Sie online sind.