

# Page de Statut Personnelle

Il semble que la version de `vnstat` que vous avez installée ne supporte pas le paramètre `-u`.  
Procédons avec les étapes correctes pour configurer `vnstat` et mettre en place l'API de surveillance de la bande passante.

## Étape 1 : Configurer `vnstat`

Initialisez `vnstat` pour votre interface réseau :

```
sudo vnstat -i eth0
```

## Étape 2 : Attendre la collecte des données

`vnstat` a besoin de temps pour collecter des données. Vérifiez l'état périodiquement :

```
sudo vnstat -l
```

Après un certain temps, vérifiez la collecte des données :

```
sudo vnstat -d
```

## Étape 3 : Créer une API pour exposer les données de bande passante

Installer Flask :

```
pip install Flask
```

Créez un script Python (`bandwidth_api.py`) :

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

app = Flask(__name__)
CORS(app)  # Activer CORS pour toutes les routes
```

```

@app.route('/bandwidth', methods=['GET'])
def get_bandwidth():
    # Exécute la commande vnstat pour obtenir les statistiques de trafic par intervalle de 5 minutes po
    result = subprocess.run(['vnstat', '-i', 'eth0', '-5', '--json'], capture_output=True, text=True)
    data = result.stdout

# Retourner les données capturées sous forme de réponse JSON
return jsonify(data)

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

```

Exécutez le script :

```

```sh
python bandwidth_api.py

```

## Étape 4 : Intégrer à votre blog

Utilisez le code HTML et JavaScript suivant pour récupérer et afficher les données de bande passante :

```

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

            // Créer un conteneur pour les heures
            var timesContainer = document.createElement('div');

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

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

```

```

    var pLocalTime = document.createElement('p');
    pLocalTime.textContent = `Mon Heure Locale : ${currentLocalTime.toString()}`;
    timesContainer.appendChild(pLocalTime);

// Ajouter le conteneur des temps au div de statut
document.getElementById('status').appendChild(timesContainer);

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

    // Créer l'en-tête du tableau
    var thead = document.createElement('thead');
    var tr = document.createElement('tr');
    var headers = ['Time', 'Traffic (KB/s)', 'Status'];
    headers.forEach(headerText => {
        var th = document.createElement('th');
        th.textContent = headerText;
        tr.appendChild(th);
    });
    thead.appendChild(tr);
    table.appendChild(thead);

// Créer le corps du tableau
var tbody = document.createElement('tbody');

    // Traiter les données de trafic
    var fiveMinuteData = bandwidthData.interfaces[0].traffic.fiveminute.reverse();
    fiveMinuteData.forEach(interval => {
        var tr = document.createElement('tr');

var dateTime = new Date(Date.UTC(interval.date.year, interval.date.month - 1, interval.date.day, interval.date.hour, interval.date.minute, interval.date.second));
var timeDifference = Math.round((currentUtcTime - dateTime) / (1000 * 60)); // Différence de temps en minutes

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

```

```

        var averageTraffic = (interval.rx + interval.tx) / 2; // Calculer la moyenne de RX et TX
        var tdTrafficKBs = document.createElement('td');
        var trafficKBs = (averageTraffic / (5 * 60 * 1024)).toFixed(2); // Convertir en Ko/s
        tdTrafficKBs.textContent = trafficKBs;
        tr.appendChild(tdTrafficKBs);

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

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

// Ajouter le tableau au div de statut
document.getElementById('status').appendChild(table);
})
.catch(error => {
    console.error('Erreur lors de la récupération des données de bande passante :', error);
});

});
""

```

Remplacez `http://your-droplet-ip:5000/bandwidth` par l'adresse IP de votre droplet.

## Considérations supplémentaires

- **Sécurité** : Assurez-vous que votre API est sécurisée. Envisagez d'ajouter une authentification.
- **Performance** : La surveillance de la bande passante peut être gourmande en ressources. Assurez-vous que votre droplet dispose de ressources suffisantes.
- **Fiabilité** : Ajoutez une gestion des erreurs et une logique de réessai pour gérer l'indisponibilité de l'API.

En suivant ces étapes, vous pouvez créer une page de statut sur votre blog qui indique si vous êtes en ligne en fonction de l'utilisation de la bande passante de votre droplet DigitalOcean.