

## 个人状态页面

看起来你安装的 `vnstat` 版本不支持 `-u` 参数。让我们按照正确的步骤配置 `vnstat` 并设置带宽监控 API。

### 步骤 1: 配置 `vnstat`

初始化 `vnstat` 以用于你的网络接口:

```
sudo vnstat -i eth0
```

### 步骤 2: 等待数据收集

`vnstat` 需要时间来收集数据。定期检查状态:

```
sudo vnstat -l
```

一段时间后, 验证数据收集:

```
sudo vnstat -d
```

### 步骤 3: 创建 API 以暴露带宽数据

安装 Flask:

```
pip install Flask
```

创建一个 Python 脚本 (`bandwidth_api.py`):

```
from flask import Flask, jsonify
import subprocess

app = Flask(__name__)

@app.route('/bandwidth', methods=['GET'])
def get_bandwidth():
    result = subprocess.run(['vnstat', '--json', 'd'], capture_output=True, text=True)
    data = result.stdout
    return jsonify(data)

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

运行脚本:

```
python bandwidth_api.py
```

#### 步骤 4: 与你的博客集成

使用以下 HTML 和 JavaScript 来获取并显示带宽数据:

```
<!DOCTYPE html>
<html lang="zh">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>状态页面</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
  <h1>状态页面</h1>
  <div id="status"></div>

  <script>
    $(document).ready(function() {
      $.ajax({
        url: 'http://your-droplet-ip:5000/bandwidth',
        method: 'GET',
        success: function(data) {
          var bandwidthData = JSON.parse(data);
          var status = bandwidthData.interfaces[1].traffic.total.rx + bandwidthData.interfaces[1].traffic.total.tx;
          $('#status').text('状态: ' + status);
        },
        error: function(error) {
          console.error('获取带宽数据时出错:', error);
        }
      });
    });
  </script>
</body>
</html>
```

将 `http://your-droplet-ip:5000/bandwidth` 替换为你的 droplet 的 IP 地址。

### 其他考虑

- **安全性：**确保你的 API 是安全的。考虑添加身份验证。
- **性能：**监控带宽可能会占用大量资源。确保你的 droplet 有足够的资源。
- **可靠性：**添加错误处理和重试逻辑以处理 API 不可用的情况。

通过按照这些步骤，你可以在博客上创建一个状态页面，根据你的 DigitalOcean droplet 的带宽使用情况显示你是否在线。