

Personal Status Page

It looks like the version of `vnstat` you installed does not support the `-u` parameter. Let's proceed with the correct steps to configure `vnstat` and set up the bandwidth monitoring API.

Step 1: Configure `vnstat`

Initialize `vnstat` for your network interface:

```
sudo vnstat -i eth0
```

Step 2: Wait for Data Collection

`vnstat` needs time to gather data. Check the status periodically:

```
sudo vnstat -l
```

After some time, verify data collection:

```
sudo vnstat -d
```

Step 3: Create an API to Expose Bandwidth Data

Install Flask:

```
pip install Flask
```

Create a Python script (`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)
```

Run the script:

```
python bandwidth_api.py
```

Step 4: Integrate with Your Blog

Use the following HTML and JavaScript to fetch and display the bandwidth data:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Status Page</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
  <h1>Status Page</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('Status: ' + status);
        },
        error: function(error) {
          console.error('Error fetching bandwidth data:', error);
        }
      });
    });
  </script>
</body>
</html>
```

Replace `http://your-droplet-ip:5000/bandwidth` with your droplet's IP address.

Additional Considerations

- **Security:** Ensure your API is secure. Consider adding authentication.
- **Performance:** Monitoring bandwidth can be resource-intensive. Ensure your droplet has enough resources.
- **Reliability:** Add error handling and retry logic to handle API unavailability.

By following these steps, you can create a status page on your blog that indicates whether you are online based on the bandwidth usage of your DigitalOcean droplet.