



## create a md file for this , with compresive step

Here is a comprehensive `monitoring-setup.md` file for setting up Prometheus + Blackbox Exporter + Grafana for home WiFi and network monitoring **without SNMP or Speedtest**:

# Home WiFi Monitoring with Prometheus, Blackbox Exporter, and Grafana

This guide walks you through setting up a robust, open-source network monitoring stack for your home WiFi using Docker Compose. It includes Prometheus for metrics, Blackbox Exporter for network/website probing, and Grafana for dashboards.

**NO need for SNMP or speedtest exporters!**

## Prerequisites

- Docker and Docker Compose installed
- Basic Linux/command line experience

## 1. Directory Structure

```
mkdir -p ~/wifi-monitoring/prometheus-data ~/wifi-monitoring/grafana-data
cd ~/wifi-monitoring
```

## 2. docker-compose.yml

```
version: '3'

networks:
  monitoring:
    name: monitoring

services:
  prometheus:
    image: prom/prometheus:latest
    container_name: prometheus
    volumes:
      - ./prometheus.yml:/etc/prometheus/prometheus.yml
      - ./prometheus-data:/prometheus
    ports:
```

```

    - "9090:9090"
  restart: unless-stopped
  networks:
    - monitoring

blackbox-exporter:
  image: prom/blackbox-exporter:latest
  container_name: blackbox-exporter
  ports:
    - "9115:9115"
  restart: unless-stopped
  networks:
    - monitoring

grafana:
  image: grafana/grafana:latest
  container_name: grafana
  volumes:
    - ./grafana-data:/var/lib/grafana
  ports:
    - "3000:3000"
  restart: unless-stopped
  environment:
    - GF_SECURITY_ADMIN_PASSWORD=admin
  networks:
    - monitoring

```

### 3. prometheus.yml

```

global:
  scrape_interval: 15s
  evaluation_interval: 15s
  external_labels:
    monitor: 'home-wifi-monitor'

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'blackbox-ping'
    metrics_path: /probe
    params:
      module: [icmp]
    static_configs:
      - targets:
          - 1.1.1.1          # Cloudflare DNS
          - 8.8.8.8          # Google DNS
          - 192.168.0.1      # Your router
          - google.com
          - youtube.com
          - github.com
    relabel_configs:
      - source_labels: [__address__]

```

```

    target_label: __param_target
  - source_labels: [__param_target]
    target_label: instance
  - target_label: __address__
    replacement: blackbox-exporter:9115

- job_name: 'blackbox-http'
  metrics_path: /probe
  params:
    module: [http_2xx]
  static_configs:
    - targets:
      - https://google.com
      - https://youtube.com
      - https://github.com
      - https://stackoverflow.com
  relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target
    - source_labels: [__param_target]
      target_label: instance
    - target_label: __address__
      replacement: blackbox-exporter:9115

```

## 4. Deploy Monitoring Stack

```

# Stop any existing monitoring containers
docker-compose down

# Start the new stack
docker-compose up -d

# Check that all services are running
docker-compose ps

```

## 5. Access Prometheus & Grafana

- Open Prometheus: <http://localhost:9090>
  - Check **Status** → **Targets** to verify all targets are "UP"
- Open Grafana: <http://localhost:3000>
  - Login: admin / admin (change password at first login)

## 6. Add Prometheus Data Source in Grafana

1. In Grafana, go to **Connections** → **Data sources** → **Add data source**
2. Select **Prometheus**
3. Set URL to `http://prometheus:9090`
4. Click **Save & Test**

## 7. Import a Blackbox Exporter Dashboard (ID: 7587)

1. Go to **Dashboards** → **New** → **Import**.
2. Enter dashboard ID `7587` and click **Load**.
3. Select your Prometheus data source.
4. Click **Import**.

## 8. Key Metrics You Now Monitor

- **probe\_success**: 1 (up) / 0 (down) for each target
- **probe\_duration\_seconds**: Ping/HTTP latency
- **probe\_icmp\_duration\_seconds**: ICMP ping times
- **probe\_http\_status\_code**: HTTP status codes from sites
- **probe\_ssl\_earliest\_cert\_expiry**: SSL cert expiry (days left)

## 9. Common Custom Panels/Queries

- **Internet status (stat panel):**

```
probe_success{job="blackbox-ping", instance="google.com"}
```

- **Router status:**

```
probe_success{instance="192.168.0.1"}
```

- **Average ping latency (gauge):**

```
avg(probe_icmp_duration_seconds{job="blackbox-ping"}) * 1000
```

- **HTTP response time:**

```
probe_http_duration_seconds
```

- **SSL certificate expiry days:**

```
(probe_ssl_earliest_cert_expiry - time()) / 86400
```

## 10. Tips

- Add/remove targets in `prometheus.yml` to monitor more/less devices or sites.
- Create alert rules in Grafana for downtime, high latency, or SSL cert expiry.
- Use auto-refresh (30s or 1m) for near-real-time dashboards.

### **Enjoy reliable and open-source WiFi/home network monitoring!**

You can save this as `monitoring-setup.md` and later attach/share as documentation or project resource.