Prometheus: Prometheus Blackbox Exporter configuration

How to use Prometheus Blackbox Exporter to probe HTTP and ICMP (ping) endpoints

👋 Welcome to the Stackhero documentation!

Stackhero offers a ready-to-use Prometheus cloud solution:

- Alert Manager included to send alerts to Slack , Mattermost , Pagerduty , etc.

- Dedicated email server to send **unlimited email alerts**

- Blackbox to probe HTTP , ICMP , TCP , and more

- **Easy configuration** with online configuration file editor

- **One-click updates** for easy maintenance

- Run in a **private VM** with a **dedicated IPv4** for maximum **performances** and **security**
  Save time and **simplify your life**: it only **takes 5 minutes** to test Stackhero's Prometheus cloud hosting solution!

Stackhero

Customize your service domain

Use the CLI

Forward logs to Graylog

Monitor your instances

Benefits   Pricing   Faq   Documentations

core configuration, and the prometheus.yml file, for the exporter configuration.

Configuration of Prometheus Blackbox Exporter core

The Blackbox Exporter core configuration (the blackbox-exporter.yml file) is where you will configure the different protocols you want to support and how you want to interact with them.

We have pre-configured it on **Stackhero for Prometheus** so you will not have to modify this file. Your Blackbox supports, out-of-the-box, ICMP monitoring and HTTP/HTTPS monitoring via GET and POST requests.

If you are an advanced user and really need to edit this file, you will find some useful configuration examples on the Blackbox Exporter repository and more on the official documentation.

Configuration of Prometheus Blackbox Exporter target

The main part of the configuration will take place in the Prometheus configuration (the prometheus.yml file). This is where you will configure Prometheus to scrape the Blackbox Exporter target as any other exporter.

Here is an example where we check ICMP (ping) replies from "prometheus.io" and "grafana.com" every 15 seconds (defined by scrape_interval).
To use this example, simply copy/paste the whole block

want to monitor.

```
  - job_name: "blackbox-icmp"
    metrics_path: "/blackbox-
exporter/probe"
    params:
      module: [ "icmp" ]
    relabel_configs:
      - source_labels: [ "__address__" ]
        target_label: "__param_target"
      - source_labels: [
"__param_target" ]
        target_label: "instance"
      - target_label: "__address__"
        replacement: "prometheus-
blackbox-exporter:9115"
    static_configs:
      - targets: [ "prometheus.io",
"grafana.com" ]
```

Here is an other example where we check HTTP servers.
If the response status is not 2xx the target will be in
error.
You can simply copy/paste this block to your
 prometheus.yml file in the scrape_configs part and fill
the targets array with he URLs you want to monitor.

```
  - job_name: "blackbox-http"
    metrics_path: "/blackbox-
exporter/probe"
```

```
relabel_configs:
  - source_labels: [ "__address__" ]
    target_label: "__param_target"
  - source_labels: [
"__param_target" ]
    target_label: "instance"
  - target_label: "__address__"
    replacement: "prometheus-
blackbox-exporter:9115"
  static_configs:
    - targets: [
"https://prometheus.io/",
"https://grafana.com/" ]
```
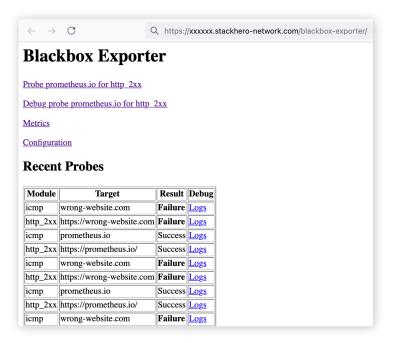
Once your scrape jobs are configured, you will be able to see the debug logs in your "Blackbox UI" (you will find the link in your Stackhero dashboard).

Blackbox UI showing the results of the last scrapes

Blackbox Exporter to probe HTTP and ICMP (ping) endpoints

- [Introduction to Prometheus Blackbox Exporter](#)

- [Prometheus Blackbox Exporter configuration](#)
  - [Configuration of Prometheus Blackbox Exporter core](#)
  - [Configuration of Prometheus Blackbox Exporter target](#)

- [Configure Grafana to show Prometheus data retrieved from Blackbox Exporter](#)

Other articles about Prometheus that might interest you

- [Introduction](#)
  How Prometheus works

- [Using Node Exporter](#)
  How to retrieve data metrics from Linux servers in Prometheus using Node Exporter

- [Alerts](#)
  How Prometheus alerts work and how to configure them

Our Managed Services

Code-Hero   Directus   Docker

Stackhero for Prometheus   Benefits   Pricing   Faq   Documentations

Privacy Policy

Graylog   InfluxDB   Kibana   Documentations

MariaDB   Matomo   Mattermost   Support

Status

Memcached   Mercure-Hub   MinIO

English

Mosquitto   MySQL   Nextcloud

Global

Node-RED   Node.js   PHP

Postfix   PostgreSQL   Prometheus

Python   RabbitMQ   Redis®*

RethinkDB   Ruby