

# Prometheus

---

[Prometheus 简介](#)

[Prometheus 特点](#)

[Prometheus 架构](#)

[Prometheus 的安装](#)

[将 Prometheus 配置为系统服务](#)

[Prometheus 主配置文件](#)

[配置 Prometheus 被监控端](#)

[被监控端涉及概念：](#)

[监控指标数据模型](#)

[Prometheus 监控案例](#)

[如何监控服务](#)

[安装 grafana](#)

[监控 Linux 服务器](#)

[监控服务运行状态](#)

[监控 Docker 服务器](#)

[监控 MySQL 服务器](#)

## Prometheus 简介

Prometheus（普罗米修斯）是一个最初在SoundCloud上构建的监控系统。自2012年成为社区开源项目，拥有非常活跃的开发人员和用户社区。为强调开源及独立维护，Prometheus于2016年加入云原生云计算基金会（CNCF），成为继Kubernetes之后的第二个托管项目

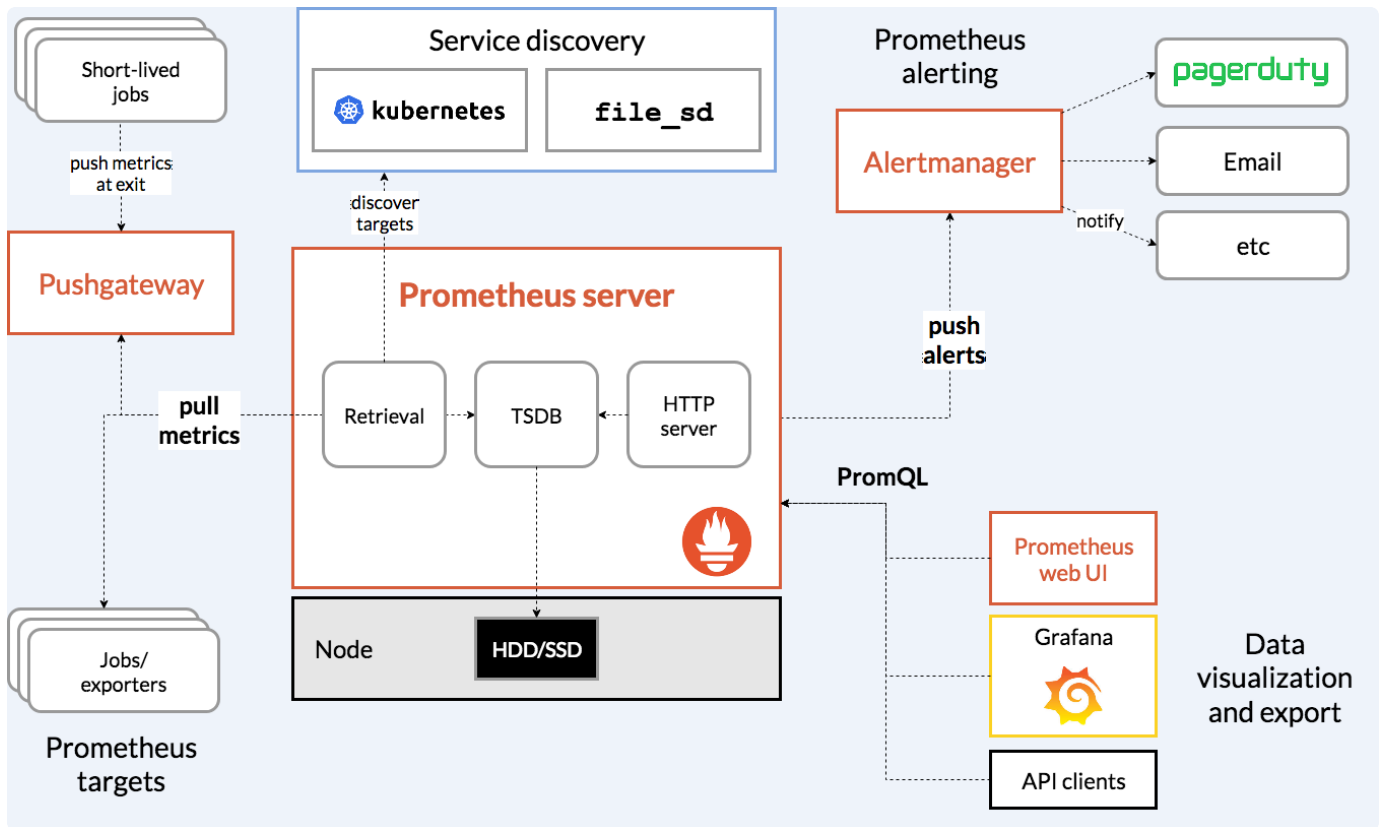
- <https://prometheus.io>
- <https://github.com/prometheus>

## Prometheus 特点

- 多维数据模型：由度量名称和键值对标识的时间序列数据
- PromQL：一种灵活的查询语言，可以利用多维数据完成复杂的查询
- 不依赖分布式存储，单个服务器节点可直接工作
- 基于HTTP的pull方式采集时间序列数据

- 推送时间序列数据通过PushGateway组件支持
- 通过服务发现或静态配置发现目标
- 多种图形模式及仪表盘支持（grafana）

## Prometheus 架构



- Pushgateway: 监控短周期任务，如cron
- jobs: 长期任务，如MySQL、Nginx 通常需要exporters
- TSDB: 将采集到的信息存放于监控节点的本地
- PromQL（查询语言）：对接可视化工具调用API Client、Grafana、自带的web UI
- Alertmanager: 对接告警
- Service Discover: 可选，自动发现被监控端

Prometheus Server: 收集指标和存储时间序列数据，并提供查询接口

ClientLibrary: 客户端库

Push Gateway: 短期存储指标数据。主要用于临时性的任务

Exporters: 采集已有的第三方服务监控指标并暴露metrics

Alertmanager: 告警

Web UI: 简单的Web控制台

## Prometheus 的安装

```
1  ./prometheus -h 命令行常用参数:
2  --config.file="prometheus.yml" # 指定配置文件
3  --web.listen-address="0.0.0.0:9090" # 监听地址和端口
4  --log.level=info # 日志级别
5  --alertmanager.timeout=10s # 与报警组件的超时时间
6  --storage.tsdb.path="data/" # 数据目录
7  --storage.tsdb.retention.time=15d # 数据保存时间, 默认15天
8
9  [root@controller01 prometheus]# ./prometheus --
   config.file="prometheus.yml" \
10  --web.listen-address="0.0.0.0:9090" \
11  --log.level=info \
12  --alertmanager.timeout=10s \
13  --storage.tsdb.path="data/" \
14  --storage.tsdb.retention.time=15d
```

## Prometheus 部署

### 解压并重命名文件夹

```
1  [root@monitor apps]# tar xzvf prometheus-2.35.0.linux-amd64.tar.gz
2  prometheus-2.35.0.linux-amd64/
3  prometheus-2.35.0.linux-amd64/conssoles/
4  prometheus-2.35.0.linux-amd64/conssoles/index.html.example
5  prometheus-2.35.0.linux-amd64/conssoles/node-cpu.html
6  prometheus-2.35.0.linux-amd64/conssoles/node-disk.html
7  prometheus-2.35.0.linux-amd64/conssoles/node-overview.html
8  prometheus-2.35.0.linux-amd64/conssoles/node.html
9  prometheus-2.35.0.linux-amd64/conssoles/prometheus-overview.html
10 prometheus-2.35.0.linux-amd64/conssoles/prometheus.html
11 prometheus-2.35.0.linux-amd64/console_libraries/
12 prometheus-2.35.0.linux-amd64/console_libraries/menu.lib
13 prometheus-2.35.0.linux-amd64/console_libraries/prom.lib
14 prometheus-2.35.0.linux-amd64/prometheus.yml
15 prometheus-2.35.0.linux-amd64/LICENSE
16 prometheus-2.35.0.linux-amd64/NOTICE
17 prometheus-2.35.0.linux-amd64/prometheus
18
19 [root@monitor apps]# mv prometheus-2.35.0.linux-amd64 prometheus
```

## 将 Prometheus 配置为系统服务

▼ prometheus.service

Bash

📄 复制代码

```
1  # vim /usr/lib/systemd/system/prometheus.service
2  [Unit]
3  Description=prometheus
4
5  [Service]
6  ExecStart=/opt/monitor/prometheus/prometheus --
   config.file=/opt/monitor/prometheus/prometheus.yml
7  ExecReload=/bin/kill -HUP $MAINPID
8  KillMode=process
9  Restart=on-failure
10
11 [Install]
12 WantedBy=multi-user.target
```

## Prometheus 主配置文件

```
1  # my global config
2  global: #全局配置
3      scrape_interval: 15s # 采集数据的时间间隔,
4      evaluation_interval: 15s # 评估告警时间规则的时间间隔
5      # scrape_timeout 采集的超时时间, 默认10s
6
7  # Alertmanager configuration
8  alerting: # alertmanager组件的位置
9      alertmanagers:
10         - static_configs:
11             - targets:
12                 # - alertmanager:9093
13
14  # Load rules once and periodically evaluate them according to the global
15  # 'evaluation_interval'.
16  rule_files: #告警规则
17      # - "first_rules.yml"
18      # - "second_rules.yml"
19
20  # A scrape configuration containing exactly one endpoint to scrape:
21  # Here it's Prometheus itself.
22  scrape_configs: #被监控端的配置
23      # The job name is added as a label `job=<job_name>` to any timeseries
24      # scraped from this config.
25      - job_name: "prometheus"
26        # metrics_path defaults to '/metrics' 指定被监控端API指标暴露位置, 默认接
27        # 口 '/metrics'
28        # scheme defaults to 'http'.默认协议 "http"
29
30      static_configs:
31          - targets: ["localhost:9090"]
```

- 更多相关配置文件信息:
  - <https://prometheus.io/docs/prometheus/latest/configuration/configuration>

## 配置 Prometheus 被监控端

### 被监控端涉及概念:

- 目标 (target) 被监控端
- 实例 (instances) 每个被监控端被称为实例
- 作业 (job) 具有相同目标的实例集合被称为作业

监控指标数据模型

- Prometheus 将所有数据存储为时间序列
- 具有相同度量名称以及标签属于同一个指标
- 每个时间序列都由度量标准名称和一组键值对（称为标签）唯一标识，通过标签查询指定指标
- 指标格式：
  - <metric name>{<label name>=<label value>,...}

Prometheus Alerts **Graph** Status Help

☐ Use local time ☐ Enable query history ☒ Enable autocomplete ☒ Enable highlighting ☒ Enable linter

prometheus\_http\_requests\_total

Execute

Table Graph

Load time: 25ms Resolution: 14s Result series: 15

Evaluation time

prometheus_http_requests_total{code="200", handler="/-/ready", instance="localhost:9090", job="prometheus"}	5
prometheus_http_requests_total{code="200", handler="/api/v1/alertmanagers", instance="localhost:9090", job="prometheus"}	1
prometheus_http_requests_total{code="200", handler="/api/v1/label/name/values", instance="localhost:9090", job="prometheus"}	5
prometheus_http_requests_total{code="200", handler="/api/v1/query", instance="localhost:9090", job="prometheus"}	5
prometheus_http_requests_total{code="200", handler="/api/v1/rules", instance="localhost:9090", job="prometheus"}	1
prometheus_http_requests_total{code="200", handler="/api/v1/status/buildinfo", instance="localhost:9090", job="prometheus"}	1
prometheus_http_requests_total{code="200", handler="/api/v1/status/runtimeinfo", instance="localhost:9090", job="prometheus"}	1
prometheus_http_requests_total{code="200", handler="/api/v1/status/tsdb", instance="localhost:9090", job="prometheus"}	1
prometheus_http_requests_total{code="200", handler="/api/v1/targets", instance="localhost:9090", job="prometheus"}	2
prometheus_http_requests_total{code="200", handler="/favicon.ico", instance="localhost:9090", job="prometheus"}	4
prometheus_http_requests_total{code="200", handler="/graph", instance="localhost:9090", job="prometheus"}	4
prometheus_http_requests_total{code="200", handler="/metrics", instance="localhost:9090", job="prometheus"}	354
prometheus_http_requests_total{code="200", handler="/static/*filepath", instance="localhost:9090", job="prometheus"}	15
prometheus_http_requests_total{code="200", handler="/targets", instance="localhost:9090", job="prometheus"}	1
prometheus_http_requests_total{code="302", handler="/", instance="localhost:9090", job="prometheus"}	2

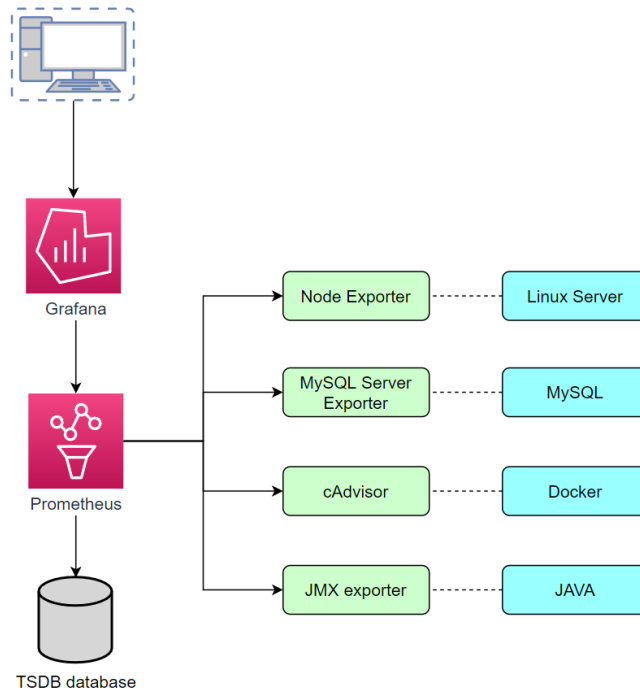
Add Panel

Remove Panel

Prometheus 监控案例

如何监控服务

- 如果要想监控，前提是能获取被监控端指标数据，并且这个数据格式必须遵循 Prometheus 数据模型，这样才能识别和采集，一般使用 exporter 提供监控指标数据



1. 自己编写 Metrics 接口
  - 1) 首先, 了解需要收集监控的指标
  - 2) 集成官方的客户端或根据数据格式编写, 并暴露
2. 使用社区维护的 exporter (采集器, 以数据模型暴露)
  - exporter 列表:  
<https://prometheus.io/docs/instrumenting/exporters>

## 安装 grafana


- Grafana 是一个开源的度量分析和可视化系统
  - 仅负责数据监控, 不负责数据采集
- 部署文档: <https://grafana.com/grafana/download>
- 访问地址: <http://IP:3000>
- 用户名/密码: admin/admin 第一次需要重置密码

使用二进制安装 Grafana

将 Grafana 加入系统服务

```
1 # vim /usr/lib/systemd/system/grafana.service
2 [Unit]
3 Description=grafana
4
5 [Service]
6 ExecStart=/opt/monitor/grafana/bin/grafana-server -
  hompath=/opt/monitor/grafana
7 ExecReload=/bin/kill -HUP $MAINPID
8 KillMode=process
9 Restart=on-failure
10
11 [Install]
12 wantedBy=multi-user.target
13
14 # systemctl daemon-reload
15 # systemctl enable --now grafana.service
```

登录 Grafana



Welcome to Grafana

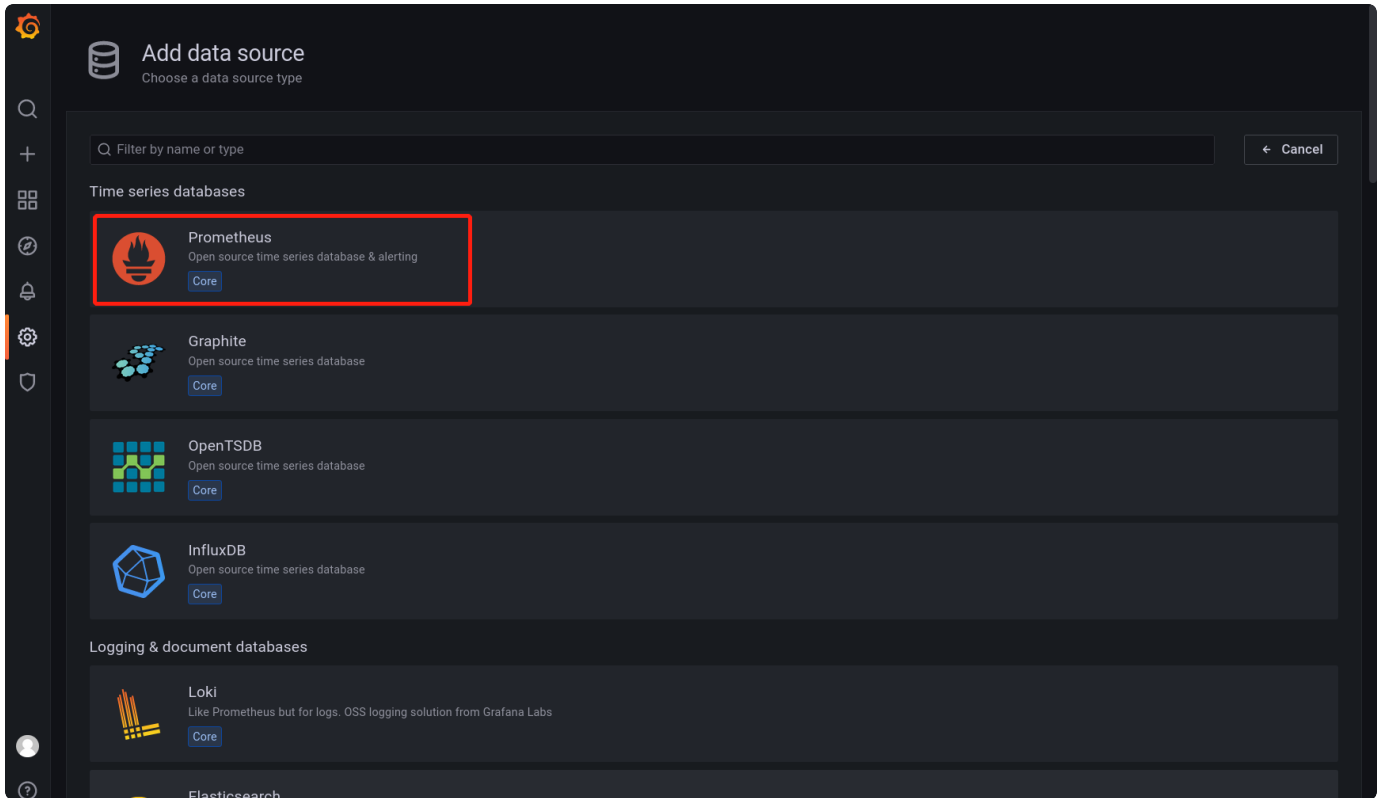
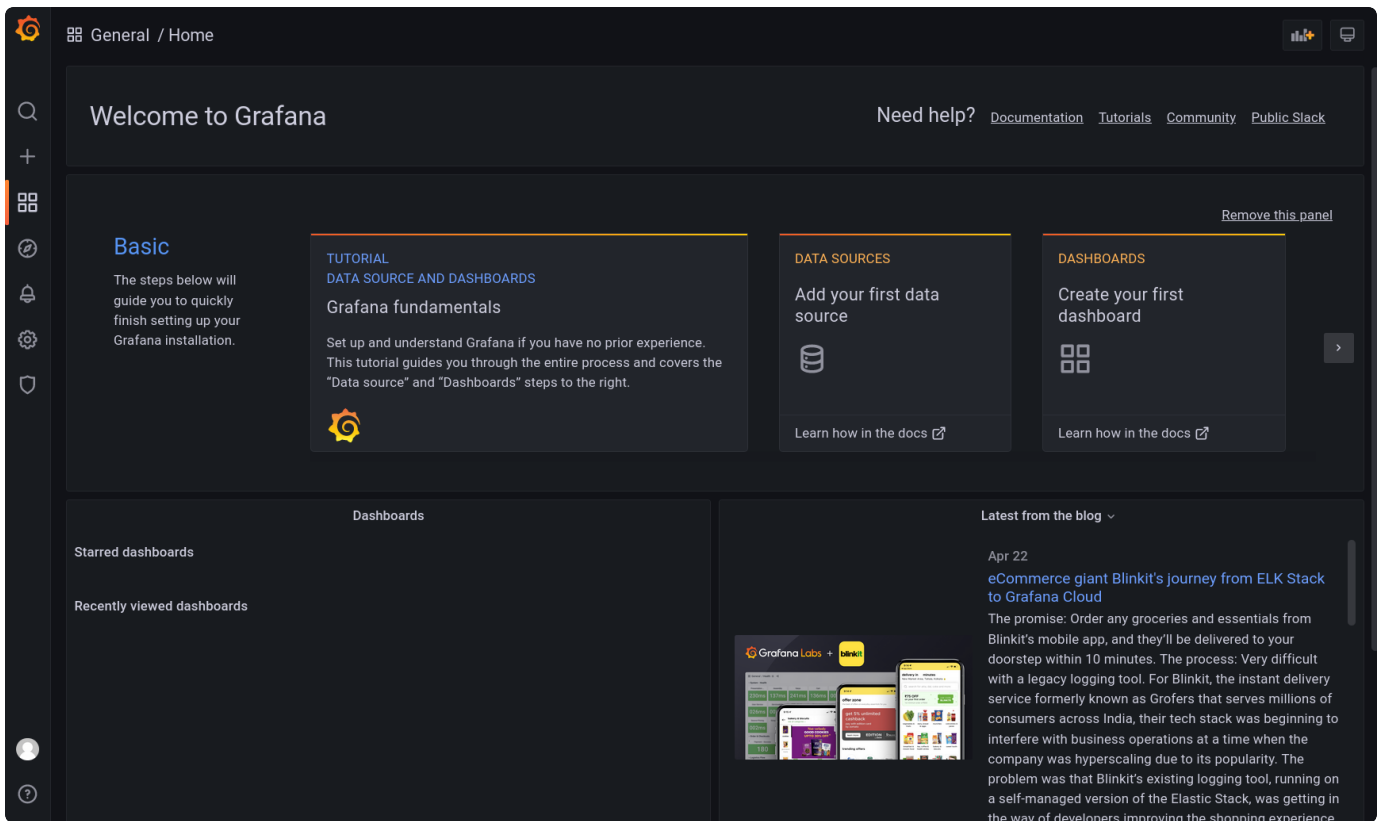
Email or username

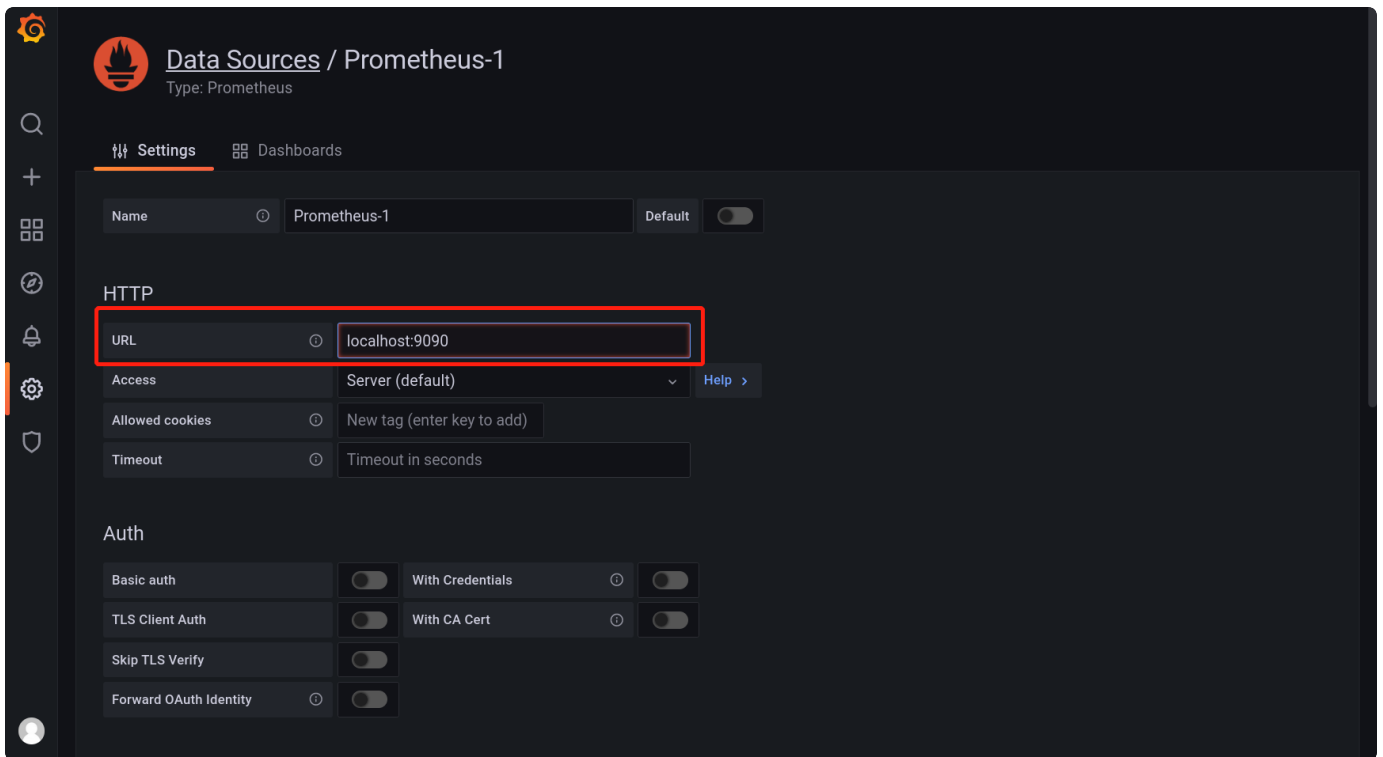
Password

[Log in](#)

[Forgot your password?](#)







## 监控 Linux 服务器

1. 部署 exporters 组件
  - a. 下载 node\_exporter ---> GitHub
  - b. 将 node\_exporter 部署到所有要监控的主机
2. 解压，并配置为系统服务
  - a. 访问端口为 9100

Bash | 复制代码

```
1 # vim /usr/lib/systemd/system/node_exporter.service
2 [Unit]
3 Description=node_exporter
4
5 [Service]
6 ExecStart=/exporters/node_exporter/node_exporter
7 ExecReload=/bin/kill -HUP $MAINPID
8 KillMode=process
9 Restart=on-failure
10
11 [Install]
12 WantedBy=multi-user.target
13 # systemctl enable --now node_exporter.service
```

### 3. 修改Prometheus服务端配置文件

YAML | 复制代码

```
1 # vim prometheus.yml
2 ... output omitted ...
3 - job_name: "Cluster-Base-Data"
4   metrics_path: "/metrics"
5   scheme: "http"
6   static_configs:
7     - targets: ["172.25.250.25:9100"]
```

### 4. 检查配置文件是否编写正确

Bash | 复制代码

```
1 # ./promtool check config ./prometheus.yml
2 Checking ./prometheus.yml
3 SUCCESS: ./prometheus.yml is valid prometheus config file syntax
```

### 5. 重启Prometheus并验证

Prometheus Alerts Graph Status ▾ Help

Configuration Copy to clipboard

```
global:
  scrape_interval: 15s
  scrape_timeout: 10s
  evaluation_interval: 15s
alerting:
  alertmanagers:
    - follow_redirects: true
      enable_http2: true
      scheme: http
      timeout: 10s
      api_version: v2
      static_configs:
        - targets: []
scrape_configs:
  - job_name: prometheus
    honor_timestamps: true
    scrape_interval: 15s
    scrape_timeout: 10s
    metrics_path: /metrics
    scheme: http
    follow_redirects: true
    enable_http2: true
    static_configs:
      - targets:
        - localhost:9090
  - job_name: Cluster-Base-Data
    honor_timestamps: true
    scrape_interval: 15s
    scrape_timeout: 10s
    metrics_path: /metrics
    scheme: http
    follow_redirects: true
    enable_http2: true
    static_configs:
      - targets:
        - 172.25.250.25:9100
```

## 6. 配置 Grafana 对该监控的可视化

Prometheus Alerts Graph Status ▾ Help

Targets

All Unhealthy Collapse All

Cluster-Base-Data (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://172.25.250.25:9100/metrics">http://172.25.250.25:9100/metrics</a>	UP	<a href="#">instance="172.25.250.25:9100"</a> <a href="#">job="Cluster-Base-Data"</a>	18.1s ago	20.540ms	

prometheus (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://localhost:9090/metrics">http://localhost:9090/metrics</a>	UP	<a href="#">instance="localhost:9090"</a> <a href="#">job="prometheus"</a>	19.2s ago	2.790ms	



为 node\_exporter 配置认证

1. 在被监控端配置 node\_exporter 目录下创建一个配置文件 config.yml
  - a. 为保证密码的安全性，需要对密码进行加密，使用 httpd-tools

```
Bash | 复制代码

1 # yum install -y httpd-tools
2 ## htpasswd -nBC 12 '' | tr -d ':\n'
3 New password:
4 Re-type new password:
5 $2y$12$o8pGmif/6VzgJKjPHSd74uxKl0Zcj667zDuvKmgRq1CCinhkYedz6
```

- b. 编写 config.yml 格式为 "USR: PASSWD"

```
Bash | 复制代码


1 # vim config.yml
2 basic_auth_users:
3   prometheus:
4     $2y$12$o8pGmif/6VzgJKjPHSd74uxKl0Zcj667zDuvKmgRq1CCinhkYedz6
```

## 2. 更改 node\_exporters.service 配置文件内容

```
▼ Bash | 复制代码

1  # vim /usr/lib/systemd/system/node_exporter.service
2  [Unit]
3  Description=node_exporter
4
5  [Service]
6  ExecStart=/exporters/node_exporter/node_exporter --
   web.config=/exporters/node_exporter/config.yml
7  ExecReload=/bin/kill -HUP $MAINPID
8  KillMode=process
9  Restart=on-failure
10
11 [Install]
12 WantedBy=multi-user.target
13
14 # systemctl daemon-reload
15 # systemctl restart node_exporter.service
```

此时，查看被监控端

 Prometheus Alerts Graph Status ▾ Help

All Unhealthy Collapse All

Filter by endpoint or labels

Cluster-Base-Data (0/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://172.25.250.25:9100/metrics">http://172.25.250.25:9100/metrics</a>	DOWN	instance="172.25.250.25:9100" job="Cluster-Base-Data"	13.983s ago	0.738ms	server returned HTTP status 401 Unauthorized

注：401 为启用认证

配置认证，在 Prometheus 的配置文件中

```
1 # vim prometheus.yml
2 ... output omitted ...
3   - job_name: "Cluster-Base-Data"
4     metrics_path: "/metrics"
5     scheme: "http"
6     basic_auth:
7       username: "prometheus"
8       password: "redhat"
9     static_configs:
10      - targets: ["172.25.250.25:9100"]
```

Prometheus Alerts Graph Status Help

## Targets

All Unhealthy Expand All

Filter by endpoint or labels

Cluster-Base-Data (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://172.25.250.25:9100/metrics">http://172.25.250.25:9100/metrics</a>	UP	<a href="#">instance="172.25.250.25:9100"</a> <a href="#">job="Cluster-Base-Data"</a>	9.89s ago	6.702ms	

## 监控服务运行状态

- 使用 systemd 管理的服务都可以被 Prometheus 监控
- 在 node\_exporter.service 配置文件中定义
- 使用 ./node\_exporter --help 查看帮助

### 1. 编辑配置文件

```

1  # vim /usr/lib/systemd/system/node_exporter.service
2  [Unit]
3  Description=node_exporter
4
5  [Service]
6  ExecStart=/exporters/node_exporter/node_exporter --collector.systemd --
  collector.systemd.unit-include=(sshd|docker|httpd).service
7  ExecReload=/bin/kill -HUP $MAINPID
8  KillMode=process
9  Restart=on-failure
10
11 [Install]
12 WantedBy=multi-user.target
13
14 # systemctl daemon-reload
15 # systemctl restart node_exporter.service

```

## 2. 查看采集数据是否成功

172.25.250.25:9100/metrics

Most Visited Fedora Docs Fedora Magazine Fedora Project User Communities Red Hat Free Content

```

# HELP node_sockstat_udp_mem_bytes Number of UDP sockets in state mem_bytes.
# TYPE node_sockstat_udp_mem_bytes gauge
node_sockstat_udp_mem_bytes 0
# HELP node_sockstat_sockets_used Number of IPv4 sockets in use.
# TYPE node_sockstat_sockets_used gauge
node_sockstat_sockets_used 481
# HELP node_softnet_dropped_total Number of dropped packets
# TYPE node_softnet_dropped_total counter
node_softnet_dropped_total{cpu="0"} 0
node_softnet_dropped_total{cpu="1"} 0
# HELP node_softnet_processed_total Number of processed packets
# TYPE node_softnet_processed_total counter
node_softnet_processed_total{cpu="0"} 15221
node_softnet_processed_total{cpu="1"} 48
# HELP node_softnet_times_squeezed_total Number of times processing packets ran out of quota
# TYPE node_softnet_times_squeezed_total counter
node_softnet_times_squeezed_total{cpu="0"} 0
node_softnet_times_squeezed_total{cpu="1"} 0
# HELP node_systemd_system_running Whether the system is operational (see 'systemctl is-system-running')
# TYPE node_systemd_system_running gauge
node_systemd_system_running 1
# HELP node_systemd_unit_state Systemd unit
# TYPE node_systemd_unit_state gauge
node_systemd_unit_state{name="sshd.service",state="activating",type="notify"} 0
node_systemd_unit_state{name="sshd.service",state="active",type="notify"} 1
node_systemd_unit_state{name="sshd.service",state="deactivating",type="notify"} 0
node_systemd_unit_state{name="sshd.service",state="failed",type="notify"} 0

```



Prometheus Alerts Graph Status Help

☐ Use local time ☐ Enable query history ☒ Enable autocomplete ☒ Enable highlighting ☒ Enable linter

node\_systemd\_unit\_state Execute

Table Graph Load time: 26ms Resolution: 14s Result series: 5

Evaluation time

node_systemd_unit_state{instance="172.25.250.25:9100", job="Cluster-Base-Data", name="sshd.service", state="activating", type="notify"}	0
node_systemd_unit_state{instance="172.25.250.25:9100", job="Cluster-Base-Data", name="sshd.service", state="active", type="notify"}	1
node_systemd_unit_state{instance="172.25.250.25:9100", job="Cluster-Base-Data", name="sshd.service", state="deactivating", type="notify"}	0
node_systemd_unit_state{instance="172.25.250.25:9100", job="Cluster-Base-Data", name="sshd.service", state="failed", type="notify"}	0
node_systemd_unit_state{instance="172.25.250.25:9100", job="Cluster-Base-Data", name="sshd.service", state="inactive", type="notify"}	0

Remove Panel

Add Panel

根据表达式进行查询，1为启动，0为停止

Prometheus Alerts Graph Status Help

☐ Use local time ☐ Enable query history ☒ Enable autocomplete ☒ Enable highlighting ☒ Enable linter

node\_systemd\_unit\_state{name="httpd.service", state="active"} Execute

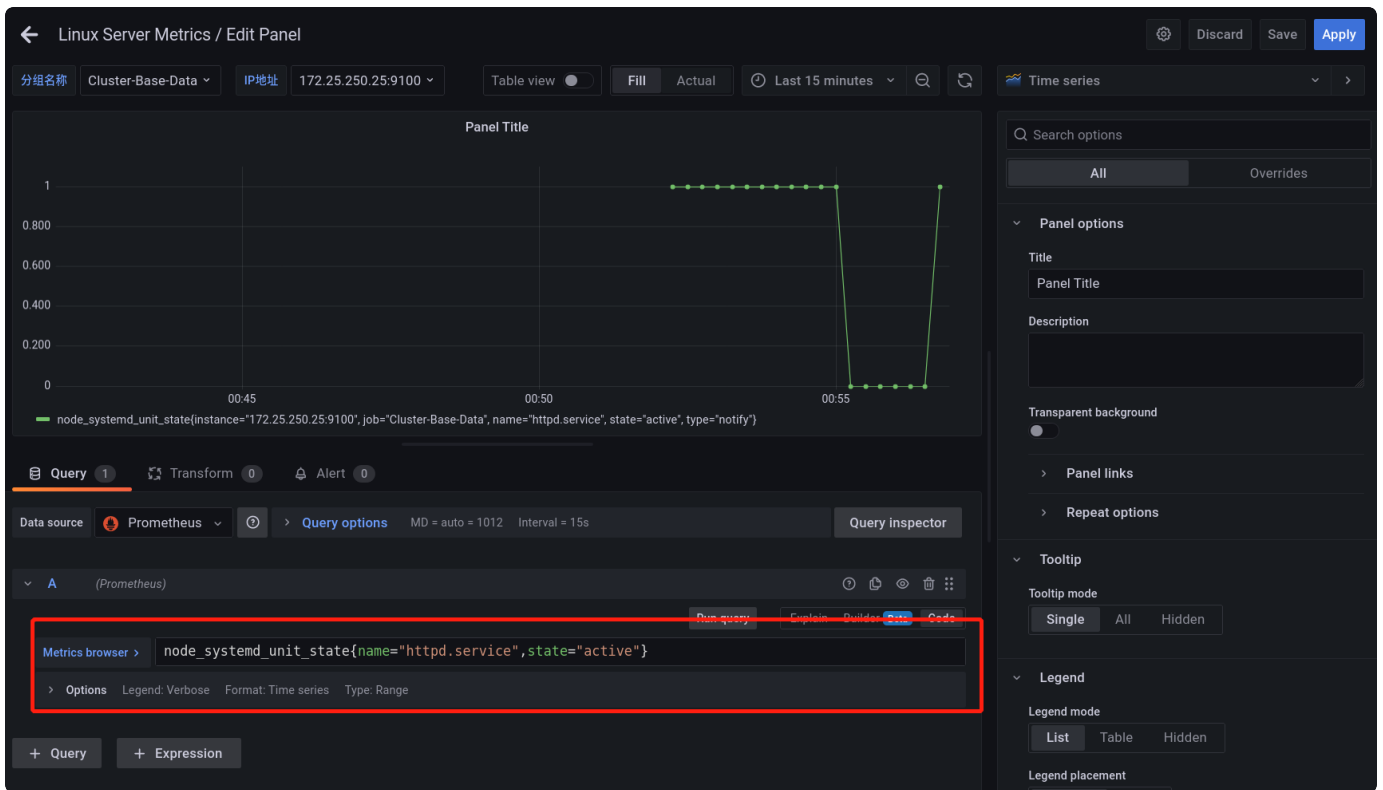
Table Graph Load time: 22ms Resolution: 14s Result series: 1

Evaluation time

node_systemd_unit_state{instance="172.25.250.25:9100", job="Cluster-Base-Data", name="httpd.service", state="active", type="notify"}	1
--	---

Remove Panel

Add Panel



## 监控 Docker 服务器

- cAdvisor (Container Advisor) : 用于收集正在运行的容器资源使用 and 性能信息
- 项目地址: <https://github.com/google/cadvisor>


使用 Docker 部署 cAdvisor

```
Bash | 复制代码

1 # docker run -d \
2 --volume=/:/rootfs:ro \
3 --volume=/var/run:/var/run:ro \
4 --volume=/sys:/sys:ro \
5 --volume=/var/lib/docker/:/var/lib/docker:ro \
6 --volume=/dev/disk/:/dev/disk:ro \
7 --publish=8080:8080 \
8 --detach=true \
9 --name=cadvisor \
10 google/cadvisor:latest
```

← → ↺ 172.25.250.25:8080/containers/ ☆ ⌵ ⌵ ⌵

☆ Most Visited 🌐 Fedora Docs 📖 Fedora Magazine 📁 Fedora Project 👤 User Communities 📁 Red Hat 📁 Free Content



# cAdvisor

/

root

[Docker Containers](#)

Subcontainers

[/docker](#)

[/system.slice](#)

[/user.slice](#)

Isolation

CPU

Shares 1024 *shares*

Allowed Cores 0 1

Memory

Reservation unlimited

Limit 1.93 GB

▼ YAML | 复制代码

```
1 # vim prometheus.yml
2 - job_name: "Docker-Server"
3   metrics_path: "/metrics"
4   scheme: "http"
5   static_configs:
6     - targets: ["172.25.250.25:8080"]
7
```



在一个监控界面监控多个 Docker 主机中的容器

← Docker Server Monitoring / Settings

General

Annotations

**Variables**

Links

Versions

Permissions

<> JSON Model

Save dashboard

Save As...

### Variables > Edit

General

Name	Node	Type	Query
Label	Docker-Server	Hide	
Description	descriptive text		

Query Options

Data source: Prometheus Refresh On dashboard load

Query: label\_values(up,instance)

Regex: /\*(?(text>.)?(?<value>.)\*)\*/

Sort: Disabled

Selection options

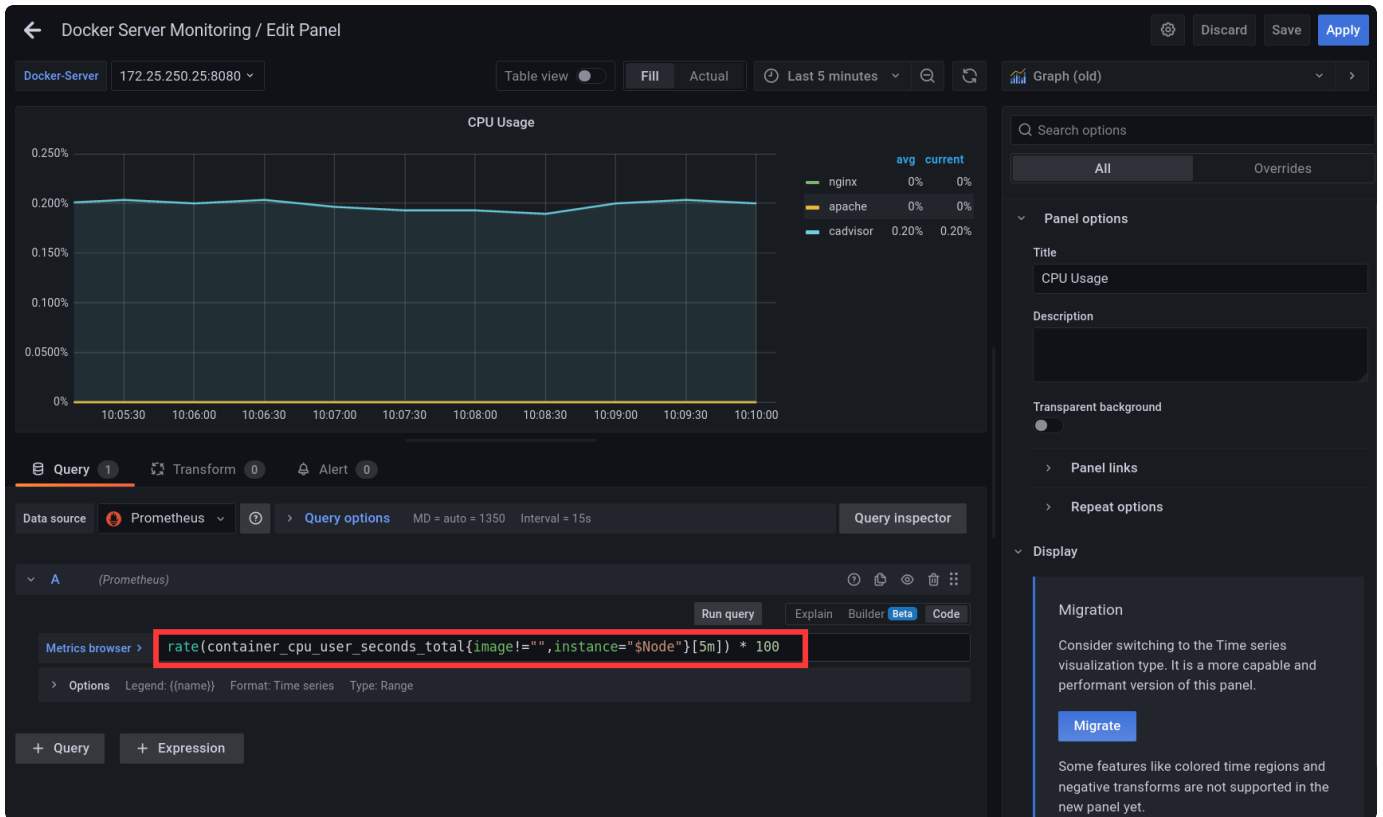
Multi-value: ☐

Include All option: ☐

Preview of values

172.25.250.24:8080 172.25.250.25:8080 172.25.250.25:9100 localhost:9090

Update

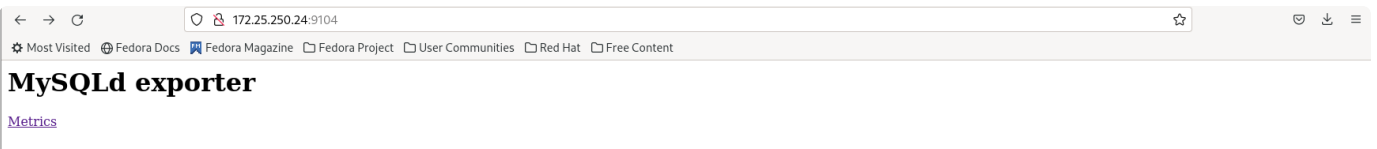


## 监控 MySQL 服务器

YAML | 复制代码

```
1 # vim .my.cnf
2 [client]
3 user=exporter
4 password=redhat
```

```
1  [Unit]
2  Description=mysql_exporter
3
4  [Service]
5  ExecStart=/exporters/mysql_exporter/mysql_exporter --config.my-
   cnf=/exporters/mysql_exporter/.my.cnf
6  ExecReload=/bin/kill -HUP $MAINPID
7  KillMode=process
8  Restart=on-failure
9
10 [Install]
11 WantedBy=multi-user.target
12
13 # systemctl daemon-reload
14 # systemctl restart mysql_exporter.service
```



```
1  - job_name: "MySQL-Server"
2    metrics_path: "/metrics"
3    scheme: "http"
4    static_configs:
5      - targets: ["172.25.250.24:9104"]
6
7  # systemctl restart prometheus.service
```

Prometheus
Alerts
Graph
Status
▼
Help

## Targets

All
Unhealthy
Collapse All

### Cluster-Base-Data (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://172.25.250.25:9100/metrics">http://172.25.250.25:9100/metrics</a>	UP	instance="172.25.250.25:9100" job="Cluster-Base-Data"	13.463s ago	13.696ms	

### Docker-Server (2/2 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://172.25.250.25:8080/metrics">http://172.25.250.25:8080/metrics</a>	UP	instance="172.25.250.25:8080" job="Docker-Server"	6.945s ago	56.247ms	
<a href="http://172.25.250.24:8080/metrics">http://172.25.250.24:8080/metrics</a>	UP	instance="172.25.250.24:8080" job="Docker-Server"	9.342s ago	37.391ms	

### MySQL-Server (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://172.25.250.24:9104/metrics">http://172.25.250.24:9104/metrics</a>	UP	instance="172.25.250.24:9104" job="MySQL-Server"	7.685s ago	12.835ms	

### prometheus (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://localhost:9090/metrics">http://localhost:9090/metrics</a>	UP	instance="localhost:9090" job="prometheus"	14.467s ago	3.127ms	

