

# Prometheus监控

## 1 Prometheus 简介

Prometheus（普罗米修斯）是一套开源的监控 & 报警 & 时间序列数据库的组合，随着发展，越来越多公司和组织接受采用 Prometheus，社区也十分活跃

exporter用来收集服务器性能数据

## 2 基于docker 搭建Prometheus

### 2.1 搭建环境

使用Prometheus和Grafana对本机服务器性能进行监控

监控本机，只需要一个exporter（Exporter 是一种用于将系统指标提供给 Prometheus 的工具。）

node\_exporter - 用于机器系统数据收集，包括cpu，内存，磁盘，io等基本信息

Grafana是一个开源的功能丰富的数据可视化平台，通常用于时序数据的可视化。它内置了以下数据源的支持

下载镜像包

#### 安装 node-exporter

启动后会在服务器上启动一个进程采集数据，prometheus 会每隔几秒通过接口获取服务器的 metrics（指标）数据

```
docker pull prom/node-exporter
```

```
[root@zx zx]# docker pull prom/node-exporter
Using default tag: latest
latest: Pulling from prom/node-exporter
dc6dd4561653: Pull complete
613f88646930: Pull complete
edad907fb257: Pull complete
Digest: sha256: 22fbde17ab647ddf89841e5e464464eece111402b7d599882c2a3393bc0d2810
Status: Downloaded newer image for prom/node-exporter: latest
docker.io/prom/node-exporter: latest
```

#### 安装prometheus

```
docker pull prom/prometheus
```

```
[root@zx zx]# docker pull prom/prometheus
Using default tag: latest
latest: Pulling from prom/prometheus
aa2a8d90b84c: Pull complete
b45d31ee2d7f: Pull complete
4aa62bad85d2: Pull complete
ab7c5e3650fa: Pull complete
e359808b78c5: Pull complete
a981f85f2027: Pull complete
1b686198e053: Pull complete
86d32b21d33f: Pull complete
185f69940eb1: Pull complete
075852623fbc: Pull complete
b36bbe7ca76a: Pull complete
5a0a2d9e2f98: Pull complete
Digest: sha256:5accb68b56ba452e449a5e552411acaeabbbe0f087acf19a1157ce3dd10a8bed
Status: Downloaded newer image for prom/prometheus: latest
docker.io/prom/prometheus: latest
```

## 安装grafana

Grafana（格拉法纳）是一个开源的分析和监控平台，**用于可视化大量的时间序列数据。**它支持各种数据源，如 Graphite、InfluxDB、Prometheus 等，使用户能够创建动态且具有交互性的仪表板。

```
docker pull grafana/grafana
```

## 启动node-exporter

```
docker run -d -p 9100:9100 \
-v "/proc:/host/proc:ro" \
-v "/sys:/host/sys:ro" \
-v ":/rootfs:ro" \
--net="host" \
prom/node-exporter
```

在 Docker 中，使用 docker pull 命令是从 Docker 镜像仓库下载（或拉取）一个镜像到本地机器上。

## 访问url,收集数据

```
http://服务器ip:9100/metrics
```

```
← → C 不安全 192.168.85.128:9100/metrics

# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 2.3717e-05
go_gc_duration_seconds{quantile="0.25"} 0.000119968
go_gc_duration_seconds{quantile="0.5"} 0.000194534
go_gc_duration_seconds{quantile="0.75"} 0.0003769
go_gc_duration_seconds{quantile="1"} 0.019205238
go_gc_duration_seconds_sum 0.073273816
go_gc_duration_seconds_count 141
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 10
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.15.8"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 2.878128e+08
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 2.20766964e+09
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.494867e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 2.178995e+08
# HELP go_memstats_gc_cpu_fraction The fraction of this program's available CPU time used by the GC since the program started.
# TYPE go_memstats_gc_cpu_fraction gauge
go_memstats_gc_cpu_fraction 1.397007076921317e-05
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 5.200896e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 2.878128e+08
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
# TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 2.878128e+08
```

## 启动prometheus

新建目录prometheus，编辑配置文件prometheus.yml

/opt 目录是用于存放可选（optional）软件的目录。该目录通常包含独立的、不属于操作系统核心部分的软件包。

```
mkdir /opt/prometheus
cd /opt/prometheus/
vim prometheus.yml
```

```

global:
  scrape_interval:     60s
  evaluation_interval: 60s

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

- job_name: linux
  static_configs:
    - targets: ['服务器ip:9100']
      labels:
        instance: localhost

```

注意服务器地址

```

- job_name: linux
  static_configs:
    - targets: ['服务器ip:9100']
      labels:
        instance: localhost

```

## 启动prometheus

```

docker run -d \
-p 9090:9090 \
-v /opt/prometheus/prometheus.yml:/etc/prometheus/prometheus.yml \
prom/prometheus

```

## 验证启动成功

```

[ root@zx prometheus ]# docker ps -a

```

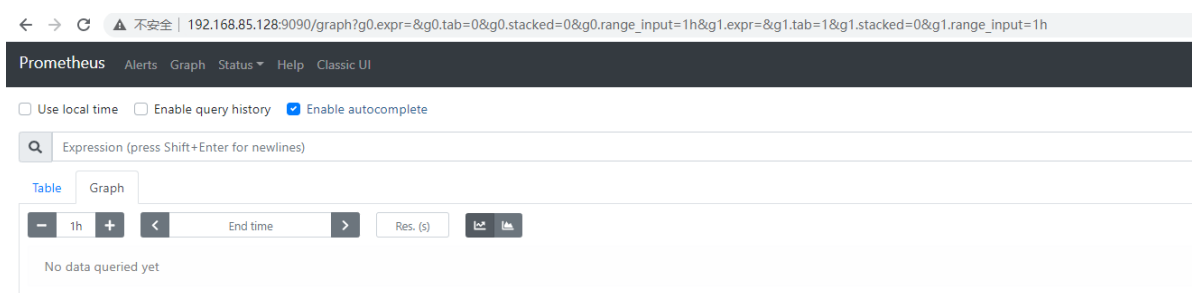
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
7e416df78199	prom/prometheus nifty_heisenberg	"/bin/prometheus --c..."	11 seconds ago	Up 10 seconds	0.0.0.0:9090->9090/tcp
9cad65c42030	prom/node-exporter	"/bin/node_exporter"	56 minutes ago	Up 56 minutes	
98bc878199f2	interesting_hypatia grafana/grafana	"/run.sh"	3 hours ago	Up 3 hours	0.0.0.0:3000->3000/tcp
006a44e1d7dc	influxdb	"/entrypoint.sh infl..."	3 hours ago	Up 3 hours	0.0.0.0:8083->8083/tcp, 0.0
1.0.0:8086->8086/tcp	JMeter-influx				

## 访问url

```

http://192.168.85.128:9090/graph

```



访问targets

```
http://192.168.85.128:9090/targets
```

## Targets

All Unhealthy Collapse All					
linux (1/1 up) show less					
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
<a href="http://192.168.85.128:9100/metrics">http://192.168.85.128:9100/metrics</a>	UP	instance="localhost" job="linux"	54.648s ago	30.695ms	
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="prometheus" job="prometheus"	36.275s ago	7.541ms	

启动grafana

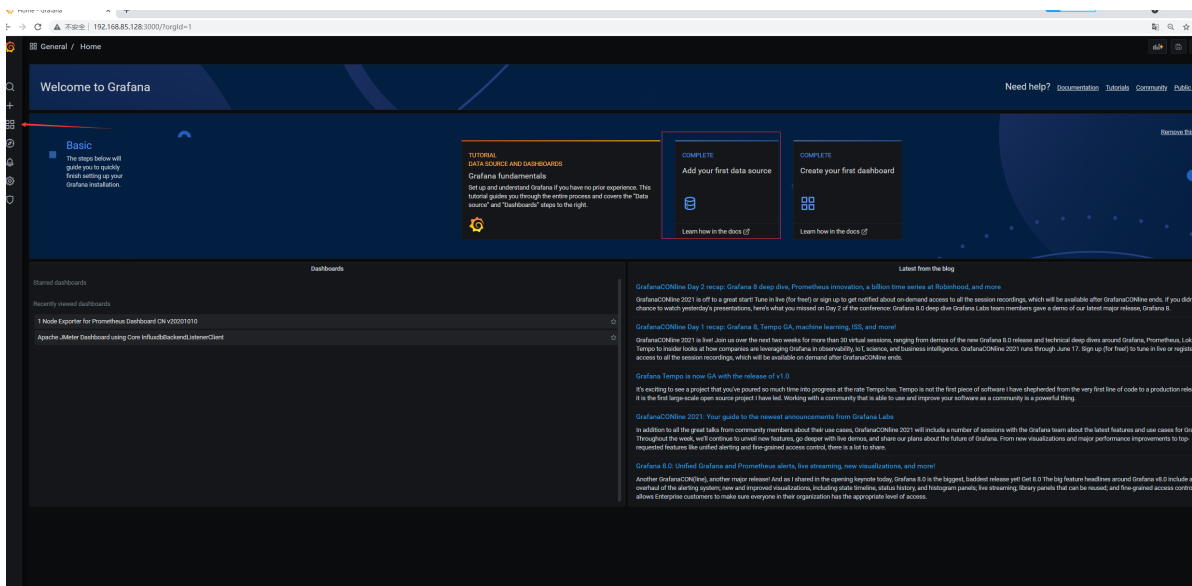
```
docker run -d --name grafana -p 3000:3000 grafana/grafana
```

访问 url

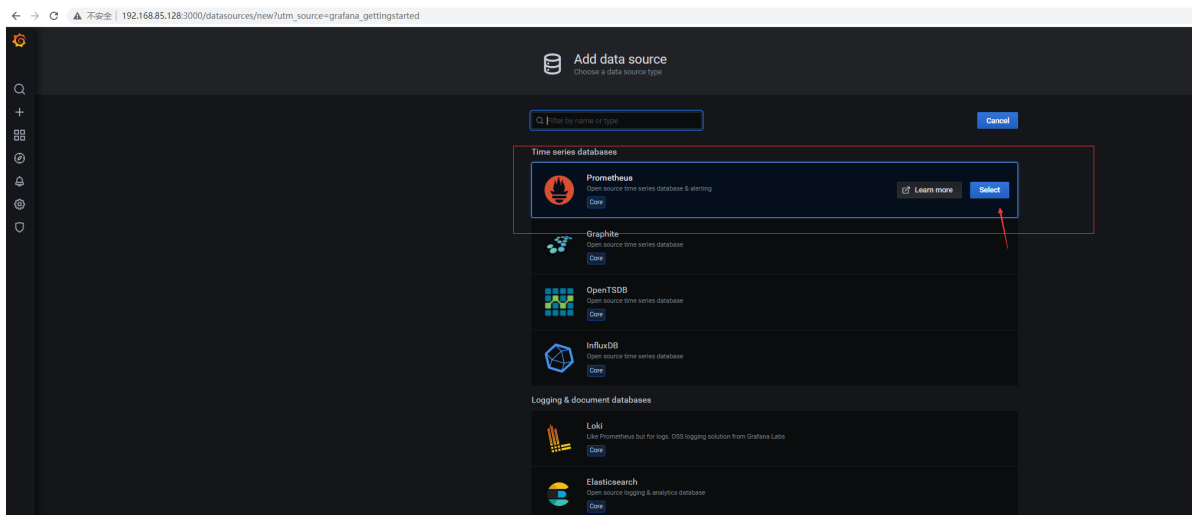
```
http://192.168.85.128:3000/
```

用户名和密码都是 admin

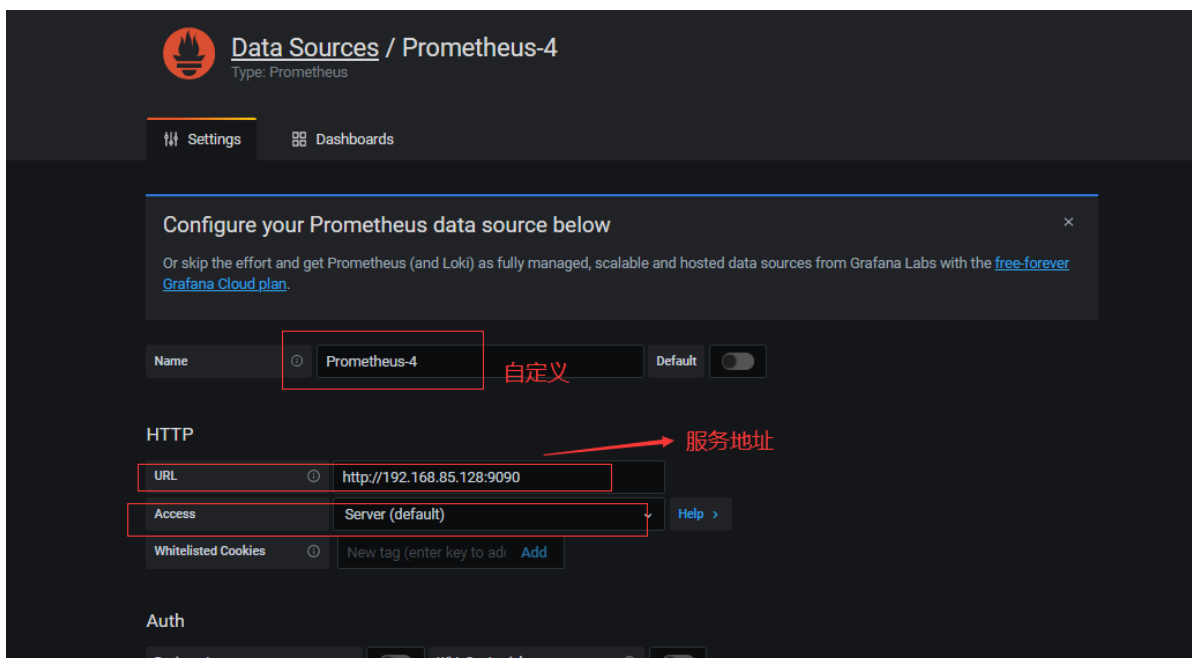
## Prometheus+Grafana生成监控信息



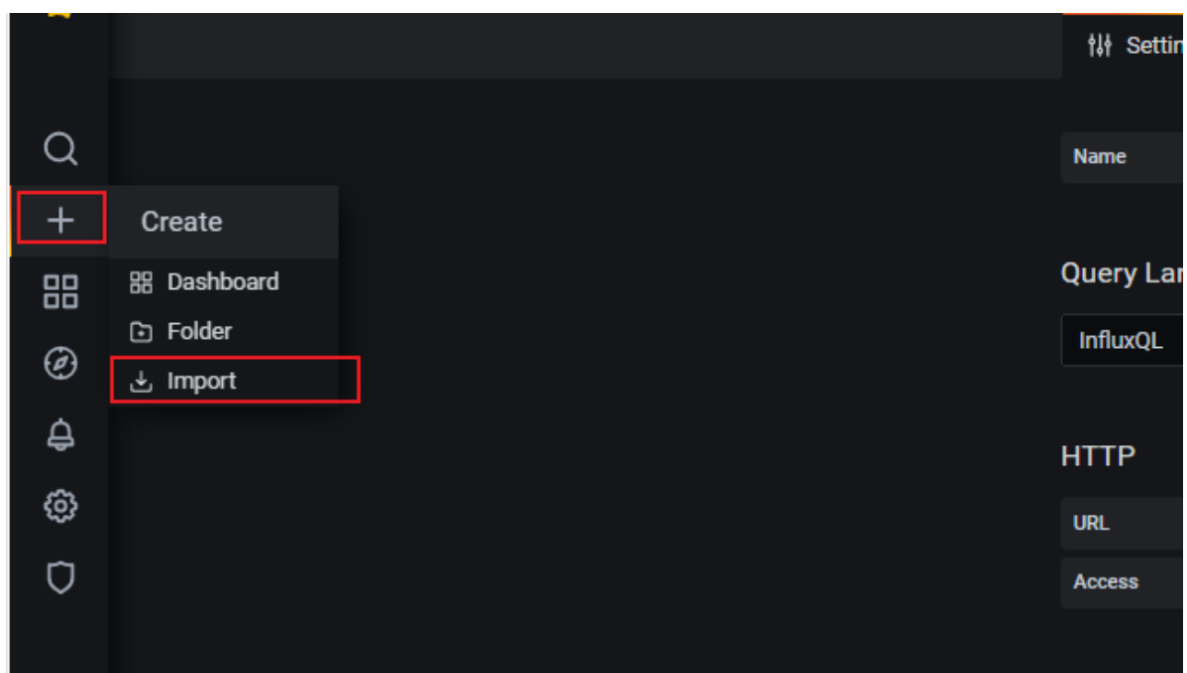
## 选择prometheus



## 配置数据源信息



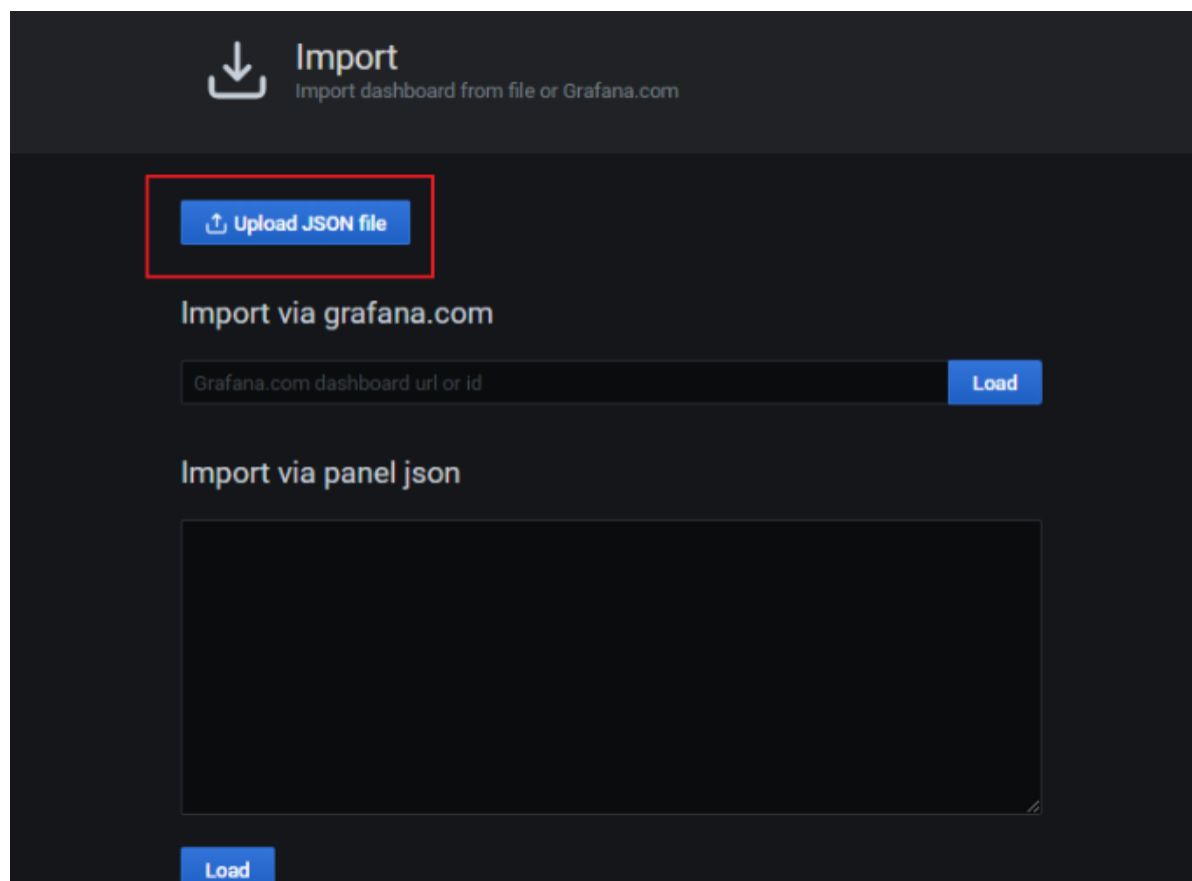
## 导入模板




## 导入模板方式

### ID 导入

8919



 **Import**  
Import dashboard from file or Grafana.com

Upload JSON file

Import via grafana.com

8919

Load

Import via panel json

Load

名称自定义

选择对应的数据源

Importing Dashboard from Grafana.com

Published by

StarsL.cn

Updated on

2021-01-30 03:15:45

Options

Name

Prometheus

Folder

General

Unique identifier (uid)

The unique identifier (uid) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The uid allows having consistent URL's for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

9CWBz0bi1

VictoriaMetrics

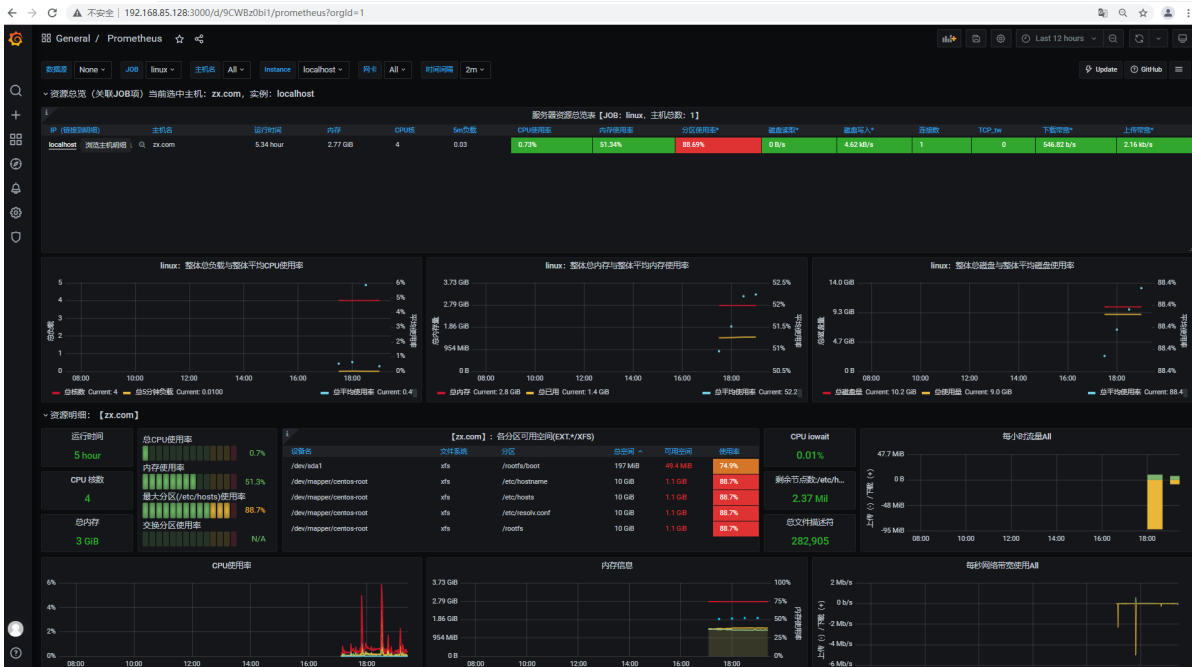
Prometheus01

Import

Cancel

点击导入

生成图表信息



## 案例

## JMeter项目配置

- 聚合报告
- 后端监听器
- 察看结果树

基本高级

Web服务器

协议: 服务器名称或IP: 192.168.85.128

HTTP请求

方法: GET 路径:

☐ 自动重定向 ☒ 跟随重定向 ☒ 使用 KeepAlive ☐ 对POST使用multipart / form-data ☐ 与浏览器兼容的头

参数消息体数据文件上传

同请求一起发送参数:

名称:	值	编码?
-----	---	-----

## 线程组设置



- 聚合报告
- 后端监听器
- 察看结果树

## 在取样器错误后要执行的动作

☒ 继续 ☐ 启动下一进程循环 ☐ 停止线程 ☐ 停止测试 ☐ 立即停止测试

## 线程属性

线程数:

1

Ramp-Up时间 (秒):

1

循环次数

☒ 永远

☒ Same user on each iteration

☐ 延迟创建线程直到需要

☐ 调度器

持续时间 (秒)

启动延迟 (秒)

## 观察grafana数据指标变化

