



سامانه مانیتورینگ پرومتئوس (Prometheus) یک سیستم مدیریت و نظارت متن باز (Open Source) است که برای جمع‌آوری و ذخیره‌سازی داده‌های نظارتی (Monitoring Data)، مانیتورینگ سیستم‌ها و خدمات مختلف استفاده می‌شود. پرومتئوس ابزاری قدرتمند برای نظارت و آنالیز سیستم‌ها و برنامه‌های کامپیوتری در محیط‌های ابری و محلی شناخته می‌شود و از طریق انتشار متن‌باز آن، به جامعه برنامه‌نویسی و اپراتورها امکان بهره‌وری از آن را می‌دهد. در این مقاله، ابتدا مفاهیم پایه مربوط به پرومتئوس را بیان می‌کنیم و سپس نحوه نصب آن را به همراه مراحل شرح می‌دهیم.

ویژگی های مهم سامانه مانیتورینگ پرومتئوس چیست؟

۱. **جمع‌آوری داده‌های نظارتی:** سامانه مانیتورینگ پرومتئوس می‌تواند از منابع مختلفی مانند وب سرورها، سیستم عامل‌ها، ابزارهای ذخیره‌سازی و برنامه‌های دیگر، داده‌های نظارتی را جمع‌آوری کند.
 ۲. **ذخیره‌سازی معتبر:** در پرومتئوس داده‌های نظارتی در یک پایگاه داده مخصوص خود ذخیره می‌شوند تا امکان تحلیل و استفاده بعدی از آن‌ها وجود داشته باشد.
 ۳. **پرس و جو و اعلام:** این ابزار از طریق زبان کوئری‌نویسی PromQL به کاربران اجازه می‌دهد تا روی داده‌های نظارتی اعلام بسازند و نتایج را تجزیه و تحلیل کنند.
 ۴. **آنالیز و هشدارها:** پرومتئوس می‌تواند براساس اصولی مانند قوانین آنومالی و تحلیل زمانی، هشدارهای نظارتی خاصی را ایجاد کند تا به کاربران اطلاع دهد که وضعیت سیستم یا خدمات تحت نظر نرم‌افزار تغییر کرده‌اند.
 ۵. **پشتیبانی از اکوسیستم ابزارهای متن‌باز:** پرومتئوس با یک اکوسیستم گسترده از ابزارهای متن‌باز (مانند Grafana) برای تصویرسازی داده‌های نظارتی و Alertmanager برای مدیریت هشدارها، تعامل دارد.
- سامانه مانیتورینگ پرومتئوس یکی از ابزارهای محبوب در حوزه مانیتورینگ، نظارت سیستمی و برنامه‌نویسی است و بسیاری از سازمان‌ها از آن برای نظارت بر سیستم‌ها و خدمات خود استفاده می‌کنند.
- در ادامه، به نحوه نصب پرومتئوس می‌پردازیم تا آشنایی دقیق با آن حاصل شود.



روش نصب سامانه مانیتورینگ پرومتئوس

پیش از آغاز فرآیند نصب سامانه مانیتورینگ پرومتئوس، لازم است از به روز بودن پکیج‌های سیستم مطمئن شوید. با اجرای دستور زیر، این عمل امکان‌پذیر خواهد شد:

```
sudo apt update
```

• ایجاد Prometheus system group

در مرحله اول لازم است که System Group برای سامانه مانیتورینگ پرومتئوس ایجاد کنید که این کار، از طریق کد زیر انجام می‌شود:

```
sudo groupadd --system prometheus
```

• ساخت Prometheus system user

سپس نیاز است یک حساب کاربری یا همان User برای Prometheus ایجاد کنیم که دستور آن در ادامه قرار داده شده است. (یک یوزر no login به نام Prometheus می‌سازیم.)

```
sudo useradd -s /sbin/nologin --system -g prometheus prometheus
```

توجه کنید که با ایجاد Group و User مشخص برای پرومتئوس، دسترسی‌های محدودی برای آن تعریف می‌شوند و به همین دلیل، از دسترسی‌های غیرمجاز جلوگیری خواهد شد.

• ساخت دایرکتوری دیتا برای سامانه مانیتورینگ پرومتئوس

برای اینکه فایل‌های کتابخانه‌ها و کانفیگ‌ها را برای سامانه مانیتورینگ پرومتئوس ذخیره‌سازی کنید، لازم است دایرکتوری مخصوص ایجاد شود که به ترتیب زیر است:

```
sudo mkdir /var/lib/Prometheus
```

• ایجاد پیکربندی فایل برای سامانه مانیتورینگ پرومتئوس

استفاده از دایرکتوری پیکربندی برای سامانه مانیتورینگ پرومتئوس به شما یک رویکرد ساختاریافته و منظم برای کانفیگ کردن و مدیریت این ابزار ارائه می‌دهد. به‌واسطه این عمل، نگهداری، مقیاس‌پذیری و سازگاری آن بهبود می‌یابد. ساخت Configuration Directories از طریق اجرای دستور زیر ایجاد می‌شود:

```
for i in rules rules.d files_sd; do sudo mkdir -p /etc/prometheus/${i}; done
```

• دانلود و نصب سامانه مانیتورینگ پرومتئوس

در مرحله دانلود و نصب سامانه مانیتورینگ پرومتئوس، لازم است آخرین نسخه Prometheus را دانلود و به روش زیر نصب کنید:

○ نصب curl packages

پکیج Curl به‌عنوان یک ابزار کمکی، امکان انتقال داده به سرور و بالعکس را برای شما فراهم خواهد کرد و بسیاری از پروتکل‌های کاربردی را پشتیبانی می‌کند. برای نصب پکیج Curl، دستور زیر را اجرا کنید:

```
[root@Mirzabaigy-pc ~]# sudo yum -y install curl wge
```

○ دانلود آخرین ورژن

از طریق دستور زیر، به دانلود آخرین نسخه اقدام کنید:

```
[root@Mirzabaigy-pc ~]# mkdir -p /tmp/prometheus && cd /tmp/prometheus
[root@Mirzabaigy-pc ~]# curl -s
https://api.github.com/repos/prometheus/prometheus/releases/latest\
| grep browser_download_url\
| grep linux-amd64\
| cut -d '"' -f 4\
| wget -qi-
```

• انتقال فایل باینری Prometheus به مسیر /usr/local/bin/ یا هر مسیر دلخواه

اکنون لازم است فرآیند انتقال فایل باینری سامانه مانیتورینگ پرومتئوس به مسیر bin (یا سایر مسیرها) به صورت زیر انجام شود:

```
[root@Mirzabaigy-pc ~]# sudo mv prometheus promtool /usr/local/bin/
```

○ انتقال فایل پیکره بندی به مسیر /etc

برای اینکه بتوانید فایل کانفیگ یا همان پیکربندی سامانه مانیتورینگ پرومتئوس را به مسیر /etc انتقال دهید، کد زیر را اجرا کنید:

```
[root@Mirzabaigy-pc ~]# sudo mv prometheus.yml /etc/prometheus/prometheus.yml
```

○ انتقال console_libraries به دایرکتوری /etc/prometheus

حال باید کنسولها و Console Library ها را به دایرکتوری /etc/prometheus انتقال دهیم که این کار از طریق دستور زیر امکان پذیر است:

```
[root@Mirzabaigy-pc ~]# sudo mv consoles/ console_libraries/ /etc/prometheus/
cd/~
rm -rf /tmp/Prometheus
```

• ساخت و ویرایش فایل پیکربندی (Create / Edit a Prometheus configuration file)

می‌توان از طریق دستور زیر، به ویرایش / درج فایل کانفیگ سامانه مانیتورینگ پرومتئوس پرداخت:

```
[root@Mirzabaigy-pc ~]# sudo vim /etc/prometheus/prometheus.yml
```

در ادامه، یک نمونه از فایل کانفیگ قرار داده شده است تا با مشاهده آن، درک بهتری از مفاهیم به دست آورید.

```
my global config #
:global
scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default
.is every 1 minute
evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is
.every 1 minute
.scrape_timeout is set to the global default (10s) #
Alertmanager configuration #
:alerting
:alertmanagers
:static_configs -
:targets -
alertmanager:9093 # -
Load rules once and periodically evaluate them according to the global #
.'evaluation_interval
:rule_files
"first_rules.yml # - "
"second_rules.yml # - "
:A scrape configuration containing exactly one endpoint to scrape #
.Here it's Prometheus itself #
:scrape_configs
The job name is added as a label `job=<job_name>` to any timeseries scraped #
.from this config
'job_name: 'prometheus -

'metrics_path defaults to '/metrics #
.'scheme defaults to 'http #

:static_configs
targets: ['localhost:9090'] -
```



• ساخت سرویس پرومتئوس (Create Prometheus systemd Service unit file)

برای آن که سامانه مانیتورینگ پرومتئوس به درستی اجرا شود و عمل مدیریت را انجام دهد، لازم است یک فایل systemd Service unit برای آن ایجاد شود. اجرای دستور زیر برای ساخت سرویس پرومتئوس به شما کمک می‌کند:

```
[root@Mirzabaigy-pc ~]# vim /etc/systemd/system/prometheus.service
[Unit]
Description=Prometheus
Documentation=https://prometheus.io/docs/introduction/overview/
Wants=network-online.target
After=network-online.target

[Service]
Type=simple
Environment="GOMAXPROCS=1"
User=prometheus
Group=prometheus
ExecReload=/bin/kill -HUP $MAINPID
ExecStart=/usr/local/bin/prometheus \
  --config.file=/etc/prometheus/prometheus.yml \
  --storage.tsdb.path=/var/lib/prometheus \
  --web.console.templates=/etc/prometheus/consoles \
  --web.console.libraries=/etc/prometheus/console_libraries \
  --web.listen-address=0.0.0.0:9090 \
  --web.external-url=

SyslogIdentifier=prometheus
Restart=always

[Install]
WantedBy=multi-user.target
```

این خط را با مقادیرهای زیر ویرایش کنید:

```
Environment="GOMAXPROCS=1"
```

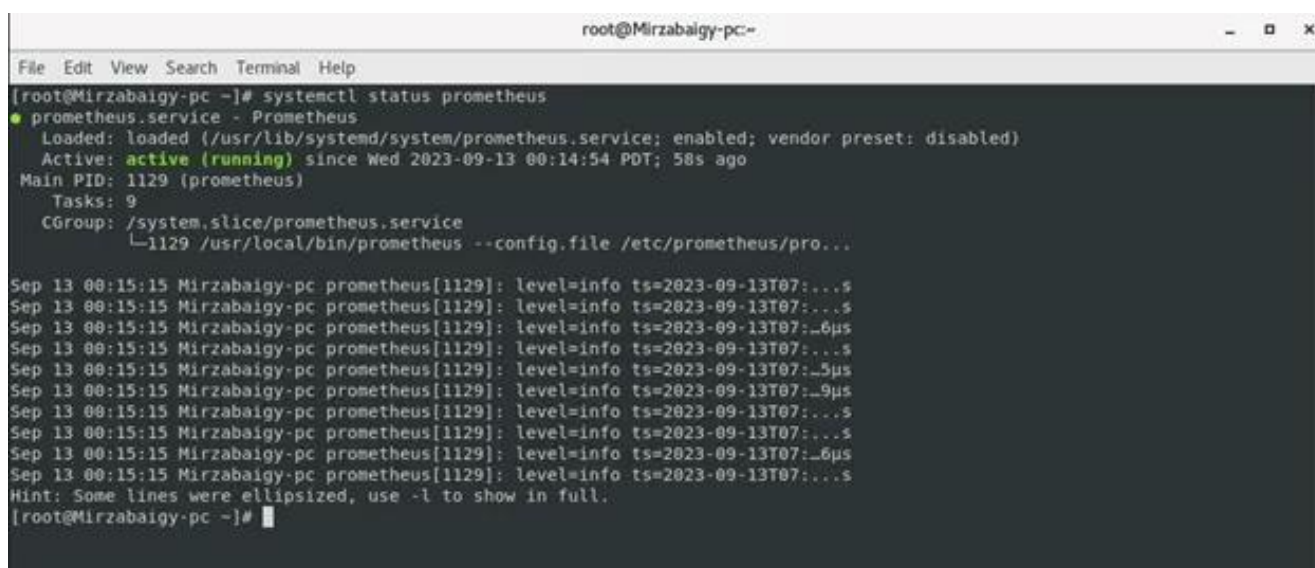
• تنظیم Firewall و راه اندازی مجدد سرویس Prometheus

```
for i in rules rules.d files_sd; do sudo chown -R prometheus:prometheus
/etc/prometheus/${i}; done
for i in rules rules.d files_sd; do sudo chmod -R 775 /etc/prometheus/${i}; done
[root@Mirzabaigy-pc ~]# sudo chown -R prometheus:prometheus /var/lib/prometheus/
[root@Mirzabaigy-pc ~]# sudo systemctl daemon-reload
[root@Mirzabaigy-pc ~]# sudo systemctl start prometheus
[root@Mirzabaigy-pc ~]# sudo systemctl enable Prometheus
```

• چک کردن وضعیت سرویس مانیتورینگ پرومتئوس (Check status using systemctl status prometheus command)

شما می‌توانید از طریق دستور زیر، وضعیت سامانه مانیتورینگ پرومتئوس را بررسی کنید:

```
[root@Mirzabaigy-pc ~]# systemctl status prometheus
```



```
root@Mirzabaigy-pc-
File Edit View Search Terminal Help
[root@Mirzabaigy-pc ~]# systemctl status prometheus
● prometheus.service - Prometheus
   Loaded: loaded (/usr/lib/systemd/system/prometheus.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2023-09-13 00:14:54 PDT; 58s ago
     Main PID: 1129 (prometheus)
       Tasks: 9
      CGroup: /system.slice/prometheus.service
              └─1129 /usr/local/bin/prometheus --config.file /etc/prometheus/pro...

Sep 13 00:15:15 Mirzabaigy-pc prometheus[1129]: level=info ts=2023-09-13T07:...s
Sep 13 00:15:15 Mirzabaigy-pc prometheus[1129]: level=info ts=2023-09-13T07:...s
Sep 13 00:15:15 Mirzabaigy-pc prometheus[1129]: level=info ts=2023-09-13T07:...6µs
Sep 13 00:15:15 Mirzabaigy-pc prometheus[1129]: level=info ts=2023-09-13T07:...s
Sep 13 00:15:15 Mirzabaigy-pc prometheus[1129]: level=info ts=2023-09-13T07:...5µs
Sep 13 00:15:15 Mirzabaigy-pc prometheus[1129]: level=info ts=2023-09-13T07:...9µs
Sep 13 00:15:15 Mirzabaigy-pc prometheus[1129]: level=info ts=2023-09-13T07:...s
Sep 13 00:15:15 Mirzabaigy-pc prometheus[1129]: level=info ts=2023-09-13T07:...s
Sep 13 00:15:15 Mirzabaigy-pc prometheus[1129]: level=info ts=2023-09-13T07:...6µs
Sep 13 00:15:15 Mirzabaigy-pc prometheus[1129]: level=info ts=2023-09-13T07:...s
Hint: Some lines were ellipsized, use -l to show in full.
[root@Mirzabaigy-pc ~]#
```

• باز کردن پورت های Firewall

در این مرحله و با دستور زیر، پورت فایروال را باز کرده و آن را Reload خواهید کرد:

```
[root@Mirzabaigy-pc ~]# sudo firewall-cmd --add-port=9090/tcp --permanent
[root@Mirzabaigy-pc ~]# sudo firewall-cmd -reload
```

• تست دسترسی به سرویس روی پورت ۹۰۹۰

در این بخش می‌خواهیم تست کنیم که آیا به سرویس Prometheus روی پورت ۹۰۹۰ دسترسی داریم یا خیر. برای این کار، اجرای دستورات زیر لازم است:

```
[root@Mirzabaigy-pc ~]# sudo yum -y install nmap-ncat
[root@Mirzabaigy-pc ~]# sudo nc -v 192.168.10.20 9090

Ncat: Version 6.40 ( http://nmap.org/ncat )
Ncat: Connected to 192.168.10.20:9090.
```

Prometheus مانیتورینگ در Node Exporter

Node Exporter یکی از مؤلفه‌های اصلی در سیستم مدیریت و نظارت Prometheus است. Node Exporter یک نرم‌افزار متن‌باز است که روی سرورها و سیستم‌های مختلف نصب می‌شود و به منظور جمع‌آوری اطلاعات و آمارهای نظارتی از اجزای سخت‌افزاری و نرم‌افزاری سیستم مورد استفاده قرار می‌گیرد.

وظایف Node Exporter

Node Exporter در سامانه مانیتورینگ پرومتئوس وظایف زیر را انجام می‌دهد:

۱. **جمع‌آوری اطلاعات سیستم:** Node Exporter اطلاعاتی مانند مصرف CPU، حافظه RAM و دیسک، ترافیک شبکه، تعداد فرآیندها و خدمات در حال اجرا و... را از سیستم جمع‌آوری می‌کند.
۲. **اتصال به Prometheus:** اطلاعات جمع‌آوری شده توسط Node Exporter به سرور Prometheus ارسال می‌شوند تا برای تحلیل و نظارت در دسترس باشند.

۳. **پیکربندی آسان:** نصب و پیکربندی Node Exporter نسبت به سایر مؤلفه‌های سامانه مانیتورینگ پرومتئوس ساده‌تر است و می‌توان آن را به‌سرعت بر روی انواع سرورها و سیستم‌ها نصب کرد.

Node Exporter به‌عنوان یک کلاینت Prometheus عمل می‌کند و اطلاعات جمع‌آوری شده از سرورها را به‌صورت متن‌باز و قابل تفسیر توسط Prometheus ارائه می‌دهد. سپس، این اطلاعات توسط Prometheus جمع‌آوری، ذخیره و برای تحلیل و تصمیم‌گیری درمورد نظارت و عملکرد سیستم‌ها استفاده می‌شوند.

نصب Node Exporter

در این مرحله، به نصب Node Exporter در سامانه مانیتورینگ پرومتئوس پرداخته خواهد شد.

- به این ترتیب، آخرین نسخه را از آرشیو دانلود کنید:

```
[root@Mirzabaigy-pc ~]# curl -s
https://api.github.com/repos/prometheus/node_exporter/releases/latest|
grep browser_download_url|grep linux-amd64|cut -d '"' -f 4|wget -qi -
```

- بعد از دانلود فایل، آن را Extract کرده و به مسیر زیر انتقال دهید:

```
/usr/local/bin
```

- این عمل از طریق دستور زیر قابل انجام است:

```
tar -xvf node_exporter*.tar.gz
cd node_exporter*/
sudo cp node_exporter /usr/local/bin
cd
```

- اکنون از طریق دستور `-version` نسخه node exporter سیستم خود را چک کنید:

```
[root@Mirzabaigy-pc ~]# node_exporter -version
```

اگر نصب به درستی انجام شده باشد، پس از اجرای دستور فوق، یک خروجی مشابه زیر در صفحه شما نمایش داده خواهد شد:

```
node_exporter, version 1.6.1 (branch: HEAD, revision:
1b48970ffcf5630534fb00bb0687d73c66d1c959)
build user: root@Mirzabaigy
build date: 20230923-09:59:09
go version: go1.19.3
platform: linux/amd64
```

• در این مرحله، یک سرویس node exporter ایجاد کنید:

```
sudo tee /etc/systemd/system/node_exporter.service <<EOF
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=default.target
EOF
```

• بارگذاری مجدد Systemd و شروع سرویس

با اجرای دستورات زیر، فرآیند بارگذاری مجدد Systemd و آغاز سرویس node exporter انجام می‌شود:

```
sudo systemctl daemon-reload
sudo systemctl start node_exporter
sudo systemctl enable node_exporter
```

○ بررسی وضعیت سرویس

به منظور بررسی وضعیت فعلی node exporter در سامانه مانیتورینگ پرومتئوس، دستور زیر را اجرا کنید:

```
[root@Mirzabaigy-pc ~]# systemctl status node_exporter.service
```

node_exporter.service – Node Exporter •

```
Loaded: loaded (/usr/lib/systemd/system/node_exporter.service; enabled;
vendor preset: disabled)
Active: active (running) since Wed 2023-09-13 00:14:54 PDT; 33min ago
Main PID: 1126 (node_exporter)
Tasks: 5
CGroup: /system.slice/node_exporter.service
└─1126 /var/lib/prometheus/node_exporter/node_exporter
```

```
Sep 13 00:14:54 Mirzabaigy-pc node_exporter[1126]: time="2023-09-13T00:14:54-07:00" level=info msg=" - sockstat" sou...:104"
Sep 13 00:14:54 Mirzabaigy-pc node_exporter[1126]: time="2023-09-13T00:14:54-07:00" level=info msg=" - stat" source=...:104"
Sep 13 00:14:54 Mirzabaigy-pc node_exporter[1126]: time="2023-09-13T00:14:54-07:00" level=info msg=" - textfile" sou...:104"
Sep 13 00:14:54 Mirzabaigy-pc node_exporter[1126]: time="2023-09-13T00:14:54-07:00" level=info msg=" - time" source=...:104"
Sep 13 00:14:54 Mirzabaigy-pc node_exporter[1126]: time="2023-09-13T00:14:54-07:00" level=info msg=" - timex" source...:104"
Sep 13 00:14:54 Mirzabaigy-pc node_exporter[1126]: time="2023-09-13T00:14:54-07:00" level=info msg=" - uname" source...:104"
Sep 13 00:14:54 Mirzabaigy-pc node_exporter[1126]: time="2023-09-13T00:14:54-07:00" level=info msg=" - vmstat" sourc...:104"
Sep 13 00:14:54 Mirzabaigy-pc node_exporter[1126]: time="2023-09-13T00:14:54-07:00" level=info msg=" - xfs" source="...:104"
Sep 13 00:14:54 Mirzabaigy-pc node_exporter[1126]: time="2023-09-13T00:14:54-07:00" level=info msg=" - zfs" source="...:104"
Sep 13 00:14:54 Mirzabaigy-pc node_exporter[1126]: time="2023-09-13T00:14:54-07:00" level=info msg="Listening on :91...:170"
Hint: Some lines were ellipsized, use -l to show in full.
```

پیکربندی سرویس Prometheus

پس از اطمینان از اینکه سرویس در حال اجرا است، باید سرویس Node_Exporter را به سامانه مانیتورینگ پرومتئوس اضافه کرد.

```
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries
  scraped from this config.
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]
  - job_name: 'node_exporter'
    static_configs:
      - targets: ['localhost:9100']
```

نکته: لازم است سرورهایی را به فایل کانفیگ Prometheus اضافه کنید که قصد مانیتور کردن آن‌ها را دارید.

- **ری‌استارت سرویس Prometheus**

برای Restart کردن سامانه مانیتورینگ پرومتئوس، دستور زیر را اجرا کنید:

```
[root@Mirzabaigy-pc ~]# sudo systemctl restart Prometheus
```

- **بررسی Node_exporter**

در مرورگر باید یک صفحه مشابه زیر را شما مشاهده کنید:

```
# HELP go_gc_duration_seconds A summary of the GC invocation durations.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 3.354e-06
go_gc_duration_seconds{quantile="0.25"} 5.872e-06
go_gc_duration_seconds{quantile="0.5"} 6.981e-06
go_gc_duration_seconds{quantile="0.75"} 1.1079e-05
go_gc_duration_seconds{quantile="1"} 0.00011718
go_gc_duration_seconds_sum 0.00105385
go_gc_duration_seconds_count 99
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 8
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.12.5"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 2.8844e+06
# 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 3.28787992e+08
# 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 1.503953e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 2.809032e+06
# 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 2.6359522433321314e-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 2.177728e+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 2.8844e+06
# 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 6.3053824e+07
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
# TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 3.694592e+06
# HELP go_memstats_heap_objects Number of allocated objects.
# TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 21930
# HELP go_memstats_heap_released_bytes Number of heap bytes released to OS.
# TYPE go_memstats_heap_released_bytes gauge
go_memstats_heap_released_bytes 0
# HELP go_memstats_heap_sys_bytes Number of heap bytes obtained from system.
# TYPE go_memstats_heap_sys_bytes gauge
go_memstats_heap_sys_bytes 6.6748416e+07
# HELP go_memstats_last_gc_time_seconds Number of seconds since 1970 of last garbage collection.
# TYPE go_memstats_last_gc_time_seconds gauge
go_memstats_last_gc_time_seconds 1.6945924635095661e+09
# HELP go_memstats_lookups_total Total number of pointer lookups.
# TYPE go_memstats_lookups_total counter
go_memstats_lookups_total 0
# HELP go_memstats_mallocs_total Total number of mallocs.
# TYPE go_memstats_mallocs_total counter
go_memstats_mallocs_total 2.830662e+06
```

مراحل نصب Blackbox Exporter

Blackbox Exporter یک کامپوننت نرم‌افزاری است که در کنار سامانه مانیتورینگ پرومتئوس مورد استفاده قرار می‌گیرد و اصلی‌ترین مورد استفاده آن، نظارت سرویس‌های خارجی، انواع وب اپلیکیشن‌ها و موارد دیگر است. شما می‌توانید پرومتئوس را طوری کانفیگ کنید که داده‌ها را از Blackbox Exporter اسکرپ کند. این عمل به شما اجازه می‌دهد تا نتایج را ذخیره‌سازی و تصویرسازی کنید و داشبوردهای شخصی‌سازی شده‌ای را ارائه دهید.



Prometheus Blackbox

- ساخت User

```
[root@Mirzabaigy-pc ~]# useradd --no-create-home --shell /bin/false  
blackbox_exporter
```

- دانلود آخرین ورژن و extract کردن آن

```
[root@Mirzabaigy-pc ~]# wget  
https://github.com/prometheus/blackbox_exporter/releases/download/v0.24.0/  
blackbox_exporter-0.24.0.linux-amd64.tar.gz
```

```
[root@Mirzabaigy-pc ~]# tar -xvf blackbox_exporter-0.14.0.linux-  
amd64.tar.gz
```

- انتقال و تغییر دسترسی

```
[root@Mirzabaigy-pc ~]# cp blackbox_exporter-0.24.0.linux-  
amd64/blackbox_exporter /usr/local/bin/blackbox_exporter  
[root@Mirzabaigy-pc ~]# chown blackbox_exporter:blackbox_exporter  
/usr/local/bin/blackbox_exporter
```

- حذف فایل آرشیو

```
[root@Mirzabaigy-pc ~]# rm -rf blackbox_exporter-0.14.0.linux-amd64*
```

- ایجاد دایرکتوری `blackbox_exporter`

```
[root@Mirzabaigy-pc ~]# mkdir /etc/blackbox_exporter  
[root@Mirzabaigy-pc ~]# vim /etc/blackbox_exporter/blackbox.yml
```

- پیکربندی فایل کانفیگ

```
modules:  
  http_2xx:  
    prober: http  
    timeout: 5s  
  http:  
    valid_status_codes: []  
    method: GET
```

- ویرایش کانفیگ مربوط به دسترسی‌ها اعطای دسترسی `ownership` به `User`

```
[root@Mirzabaigy-pc ~]# chown blackbox_exporter:blackbox_exporter  
/etc/blackbox_exporter/blackbox.yml
```

- ایجاد سرویس (the systemd unit file)

```
[root@Mirzabaigy-pc ~]# vim /etc/systemd/system/blackbox_exporter.service
```

- پیکربندی فایل کانفیگ سرویس

```
[Unit]
Description=Blackbox Exporter
Wants=network-online.target
After=network-online.target

[Service]
User=blackbox_exporter
Group=blackbox_exporter
Type=simple
ExecStart=/usr/local/bin/blackbox_exporter --config.file
/etc/blackbox_exporter/blackbox.yml

[Install]
WantedBy=multi-user.target
```

- بارگذاری مجدد systemd daemon و راهاندازی مجدد سرویس

```
[root@Mirzabaigy-pc ~]# systemctl daemon-reload
[root@Mirzabaigy-pc ~]# systemctl start blackbox_exporter
```

- بررسی و اعتبارسنجی وضعیت سرویس

```
[root@Mirzabaigy-pc ~]# systemctl status blackbox_exporter
```

blackbox_exporter.service – Blackbox Exporter •

```
Loaded: loaded (/etc/systemd/system/blackbox_exporter.service; enabled;
vendor preset: disabled)
Active: active (running) since Sat 2023-09-16 22:24:59 PDT; 7min ago
Main PID: 1125 (blackbox_export)
Tasks: 5
CGroup: /system.slice/blackbox_exporter.service
└─1125 /usr/local/bin/blackbox_exporter --config.file
/etc/blackbox_exporter/blackbox.yml
```

```
Sep 16 22:24:59 Mirzabaigy-pc systemd[1]: Started Blackbox Exporter.
Sep 16 22:25:00 Mirzabaigy-pc blackbox_exporter[1125]: level=info ts=2023-
09-17T05:25:00.707297275Z caller=main.go:213 msg="Starting
blackbox_exporter" version="(v...4f7d968a)"
Sep 16 22:25:00 Mirzabaigy-pc blackbox_exporter[1125]: level=info ts=2023-
09-17T05:25:00.752357678Z caller=main.go:226 msg="Loaded config file"
Sep 16 22:25:00 Mirzabaigy-pc blackbox_exporter[1125]: level=info ts=2023-
09-17T05:25:00.763928674Z caller=main.go:330 msg="Listening on address"
address=:9115
Hint: Some lines were ellipsized, use -l to show in full.
```

```
root@Mirzabaigy-pc ~# systemctl status blackbox_exporter
● blackbox_exporter.service – Blackbox Exporter
   Loaded: loaded (/etc/systemd/system/blackbox_exporter.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2023-09-16 22:24:59 PDT; 7min ago
     Main PID: 1125 (blackbox_export)
        Tasks: 5
      CGroup: /system.slice/blackbox_exporter.service
              └─1125 /usr/local/bin/blackbox_exporter --config.file /etc/blackbox_exporter/blackbox.yml

Sep 16 22:24:59 Mirzabaigy-pc systemd[1]: Started Blackbox Exporter.
Sep 16 22:25:00 Mirzabaigy-pc blackbox_exporter[1125]: level=info ts=2023-09-17T05:25:00.707297275Z caller=main.go:213 msg="Starting blackbox_exporter" version="(v...4f7d968a)"
Sep 16 22:25:00 Mirzabaigy-pc blackbox_exporter[1125]: level=info ts=2023-09-17T05:25:00.752357678Z caller=main.go:226 msg="Loaded config file"
Sep 16 22:25:00 Mirzabaigy-pc blackbox_exporter[1125]: level=info ts=2023-09-17T05:25:00.763928674Z caller=main.go:330 msg="Listening on address" address=:9115
Hint: Some lines were ellipsized, use -l to show in full.
root@Mirzabaigy-pc ~#
```

فعال سازی سرویس •

```
[root@Mirzabaigy-pc ~]# systemctl enable blackbox_exporter
```


• **مانتورینگ API**

حالا اگر بخواهید یک API را مانیتور کنید، باید این مراحل را دنبال کنید:

```
[root@Mirzabaigy-pc ~]# vim /etc/prometheus/conf.d/blackbox-exporter.yml
```

```
- targets:
  - https://test.ir/health
labels:
  project: 'test'
  service: 'app'
  env: 'production'
  healthcheck: 'app'
```

• **درنهایت، باید به فایل پیکربندی Prometheus اضافه کنیم.**

```
[root@Mirzabaigy-pc ~]# vim /etc/prometheus/prometheus.yml
```

```
- job_name: 'blackbox'
  metrics_path: /probe
  params:
    module: [http_2xx]
  static_configs:
    - targets:
      - http://localhost:8080
  relabel_configs:
    - source_labels: [__address__]
      target_label: __param_target
    - source_labels: [__param_target]
      target_label: instance
    - target_label: __address__
      replacement: localhost:9115
```

• **باز کردن یک ترمینال جدید و ایجاد یک index.html**

```
[root@Mirzabaigy-pc ~]# echo "ok" > index.html
```

• **استارت یک SimpleHTTPServer روی port :8080**

```
[root@Mirzabaigy-pc ~]# python -m SimpleHTTPServer 8080
```

Blackbox Exporter

[Probe prometheus.io for http_2xx](#)

[Debug probe prometheus.io for http_2xx](#)

[Metrics](#)

[Configuration](#)

Recent Probes

Module	Target	Result	Debug
http_2xx	prometheus.io	Failure	Logs
http_2xx	prometheus.io	Failure	Logs

- با استفاده از دستور زیر، سرویس را Restart کنید:

```
[root@Mirzabaigy-pc ~]# systemctl restart Prometheus
```

- درنهایت، آن را به گرافانا اضافه کنید.

پس از انجام این کار، باید یک خروجی مشابه زیر بر روی صفحه شما نمایش داده شود.





جمع بندی

در این مقاله، به بررسی نحوه نصب و استفاده از سامانه مانیتورینگ پرومتئوس به صورت مرحله به مرحله پرداخته شد. علاوه بر این، مراحل نصب node exporter و Blackbox exporter آموزش داده شدند. پرومتئوس یک ابزار متن باز کاربردی است که سازمان‌ها آن را به منظور نظارت بر اپلیکیشن‌ها و سیستم‌های خود در ابعاد وسیع استفاده می‌کنند. به همین دلیل، این مقاله نیک آموز با هدف افزایش آگاهی شما از روند بهره‌وری از این ابزار ارائه شده است.