

# Prometheus

## Installation

Download: [Download](#) | [Prometheus](#) ?select Operating system: linux, Architecture: amd64

```
tar xzf prometheus-2.43.0.linux-amd64.tar.gz
mv prometheus-2.43.0.linux-amd64 /opt/prometheus
```

?????

```
cd /opt/prometheus/
./prometheus --config.file="prometheus.yml"
```

???? (????????):

- <http://localhost:9090/metrics>  
????????????????
- <http://localhost:9090/>  
?? Graph?? Expression ?? `promhttp_metric_handler_requests_total` ???  
Execute????????

## Configuration

???????????

```
./prometheus --config.file="prometheus.yml" \  
  --storage.tsdb.path="/data/prometheus" \  
  --storage.tsdb.retention.time=30d
```

- `--storage.tsdb.path`:  
Where Prometheus writes its database. Defaults to data/.
- `--storage.tsdb.retention.time`:  
When to remove old data. Defaults to 15d. Overrides storage.tsdb.retention if this flag is set to anything other than default.
- `--storage.tsdb.retention.size`:  
The maximum number of bytes of storage blocks to retain. The oldest data will be removed first. Defaults to 0 or disabled. Units supported: B, KB, MB, GB, TB, PB, EB.

Ex: "512MB". Based on powers-of-2, so 1KB is 1024B. Only the persistent blocks are deleted to honor this retention although WAL and m-mapped chunks are counted in the total size. So the minimum requirement for the disk is the peak space taken by the wal (the WAL and Checkpoint) and chunks\_head (m-mapped Head chunks) directory combined (peaks every 2 hours).

????(????)

## RedHat 8

???????

```
useradd -s /sbin/nologin --system prometheus
mkdir /etc/prometheus /data/prometheus
```

????

```
tar xvf prometheus-*.tar.gz
cd prometheus-*/
cp prometheus promtool /usr/local/bin/
cp -r prometheus.yml consoles/ console_libraries/ /etc/prometheus/

chown -R prometheus.prometheus /etc/prometheus
chmod -R 0755 /etc/prometheus
chown prometheus.prometheus /data/prometheus
```

?????: `/etc/systemd/system/prometheus.service`

```
[Unit]
Description=Prometheus Time Series Collection and Processing Server
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecReload=/bin/kill -HUP $MAINPID
ExecStart=/usr/local/bin/prometheus \
  --config.file /etc/prometheus/prometheus.yml \
  --storage.tsdb.path /data/prometheus/ \
```

```
--storage.tsdb.retention.time=30d \  
--web.console.templates=/etc/prometheus/consoles \  
--web.console.libraries=/etc/prometheus/console_libraries
```

[Install]

WantedBy=multi-user.target

???

```
systemctl daemon-reload  
systemctl start prometheus.service  
systemctl enable prometheus.service
```

## Monitor to Linux node

Linux Monitoring with Node Exporter

- [Node Exporter for Prometheus Dashboard EN v20201010](#)
- [Node Exporter Full](#)
- [Node Exporter Server Metrics](#)
- [Node Exporter for Prometheus Dashboard based on 11074](#)
- [Node Exporter Quickstart and Dashboard](#)
- [\[Video\] Setting up Prometheus and Grafana for monitoring your servers](#)
- [MONITORING LINUX HOST METRICS WITH THE NODE EXPORTER](#)

## On Linux target

Node Exporter Installation

Download: [Download | Prometheus](#)

```
tar xzf node_exporter-1.5.0.linux-amd64.tar.gz  
mv node_exporter-1.5.0.linux-amd64 /opt/node_exporter  
chown -R root.root /opt/node_exporter  
  
cd /opt/node_exporter  
./node_exporter  
# Ctrl + C to exit
```

## Set up node\_exporter as service

```
# Create a user
useradd -r -c "Node Exporter" -s /sbin/nologin node_exporter

# Create a service file
cat <<EOF>/etc/systemd/system/node_exporter.service
[Unit]
Description=Node Exporter

[Service]
User=node_exporter
EnvironmentFile=/etc/sysconfig/node_exporter
ExecStart=/opt/node_exporter/node_exporter $OPTIONS

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

```
# Create the file /etc/sysconfig/node_exporter
echo '#OPTIONS=""' > /etc/sysconfig/node_exporter

# Start the node exporter
systemctl daemon-reload
systemctl start node_exporter.service
```

## On Prometheus Server

prometheus.yml:

```
scrape_configs:

  # Linux Nodes
  - job_name: linux

    # Override the global default and scrape targets from this job every 5 seconds.
    scrape_interval: 15s

    static_configs:
      - targets: ['linux-node-ip:9100']
```

# Monitor to MySQL

- [MySQL Overview](#)
- [MySQL Server Exporter](#)
- [\[Video\] Setting up Prometheus and Grafana for monitoring your servers](#)
- [Prometheus ??Mysql????Grafana???](#)

# Monitor to AIX

- [nimon working with Prometheus](#)

# Monitor to RabbitMQ

prometheus.yml:

```
scrape_configs:

  # RabbitMQ Nodes
  - job_name: rabbitmq

    # Override the global default and scrape targets from this job every 5 seconds.
    scrape_interval: 15s

    static_configs:
      - targets: ['rmq01:15692', 'rmq02:15692', 'rmq03:15692']
```

# Monitor to Containers

- [MONITORING DOCKER CONTAINER METRICS USING CADVISOR](#)
- [\[Video\] Node Application Monitoring with cAdvisor Prometheus and Grafana | part 1](#)
- [?????cAdvisor](#)