

Install Prometheus Server on Ubuntu 20.04/18.04 | Debian 10/9

By [Josphat Mutai](#) - May 5, 2020

You can support us by downloading this article as PDF from the Link below.

 [Download the guide as PDF](#)

How can I install Prometheus on Ubuntu 20.04/18.04 and Debian 10/9 Linux?.

Prometheus is a monitoring tool designed for recording real-time metrics in a time-series database. It is an open-source software project, written in Go. The Prometheus metrics are collected using HTTP pulls, allowing for higher performance and scalability. In this tutorial, we'll discuss how you can install Prometheus Server on Debian & Ubuntu Linux systems.

Other tools which make Prometheus complete monitoring tool are:

- **Exporters:** These are libraries that help with exporting metrics from third-party systems as Prometheus metrics.
- **PromQL:** Prometheus query language which allows you to filter multi-dimensional time series data.

Grafana is a tool commonly used to visualize data polled by Prometheus, for monitoring, and analysis. It is used to create dashboards with panels representing specific metrics over a set period of time.

Step 1: Create Prometheus system group

Let's start by creating the Prometheus system user and group.

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

The group with ID < 1000 is a system group. Once the system group is added, create Prometheus system user and assign primary group created.

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

Step 2: Create data & configs directories for Prometheus

Prometheus needs a directory to store its data. We will create this under */var/lib/prometheus*.

```
sudo mkdir /var/lib/prometheus
```

Prometheus primary configuration files directory is */etc/prometheus/*. It will have some sub-directories:

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

Step 3: Download Prometheus

We need to download the latest release of Prometheus archive and extract it to get binary files. You can check releases from [Prometheus releases Github](#) page.

Install wget.

```
sudo apt update
sudo apt -y install wget curl vim
```

Then download latest binary archive for Prometheus.

```
mkdir -p /tmp/prometheus && cd /tmp/prometheus
curl -s
https://api.github.com/repos/prometheus/prometheus/releases/latest |
grep browser_download_url | grep linux-amd64 | cut -d '"' -f 4 | wget -
qi -
```

Extract the file:

```
tar xvf prometheus*.tar.gz
cd prometheus*/
```

Move the binary files to `/usr/local/bin/` directory.

```
sudo mv prometheus promtool /usr/local/bin/
```

Check installed version:

```
$ prometheus --version
prometheus, version 2.15.2 (branch: HEAD, revision:
d9613e5c466c6e9de548c4dae1b9aabbf9aaf7c57)
  build user:      root@688433cf4ff7
  build date:      20200106-14:50:51
  go version:      go1.13.5
```

```
$ promtool --version
promtool, version 2.15.2 (branch: HEAD, revision:
d9613e5c466c6e9de548c4dae1b9aabbf9aaf7c57)
  build user:      root@688433cf4ff7
  build date:      20200106-14:50:51
  go version:      go1.13.5
```

Move Prometheus configuration template to /etc directory.

```
sudo mv prometheus.yml /etc/prometheus/prometheus.yml
```

Also move consoles and console_libraries to */etc/prometheus* directory:

```
sudo mv consoles/ console_libraries/ /etc/prometheus/
```

Step 4: Configure Prometheus on Debian / Ubuntu

Create or edit a configuration file for Prometheus – */etc/prometheus/prometheus.yml*.

```
sudo vim /etc/prometheus/prometheus.yml
```

The template configurations should look similar to below:

```
# 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']
```

You can edit the file to your default liking and save it.

Create a Prometheus systemd Service unit file

To be able to manage Prometheus service with systemd, you need to explicitly define this unit file.

```
sudo tee /etc/systemd/system/prometheus.service<<EOF
[Unit]
Description=Prometheus
Documentation=https://prometheus.io/docs/introduction/overview/
Wants=network-online.target
After=network-online.target

[Service]
Type=simple
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
EOF
```

Change directory permissions.

Change the ownership of these directories to Prometheus user and group.

```
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
sudo chown -R prometheus:prometheus /var/lib/prometheus/
```

Reload systemd daemon and start the service:

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

Check status using *systemctl status prometheus* command:

```
$ systemctl status prometheus
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled;
 vendor preset: enabled)
   Active: active (running) since Sun 2020-01-19 14:36:08 UTC; 14s ago
     Docs: https://prometheus.io/docs/introduction/overview/
  Main PID: 1397 (prometheus)
    Tasks: 7 (limit: 2377)
   Memory: 21.7M
    CGroup: /system.slice/prometheus.service
           └─1397 /usr/local/bin/prometheus --
config.file=/etc/prometheus/prometheus.yml --
storage.tsdb.path=/var/lib/prometheus --web.console.templates
```

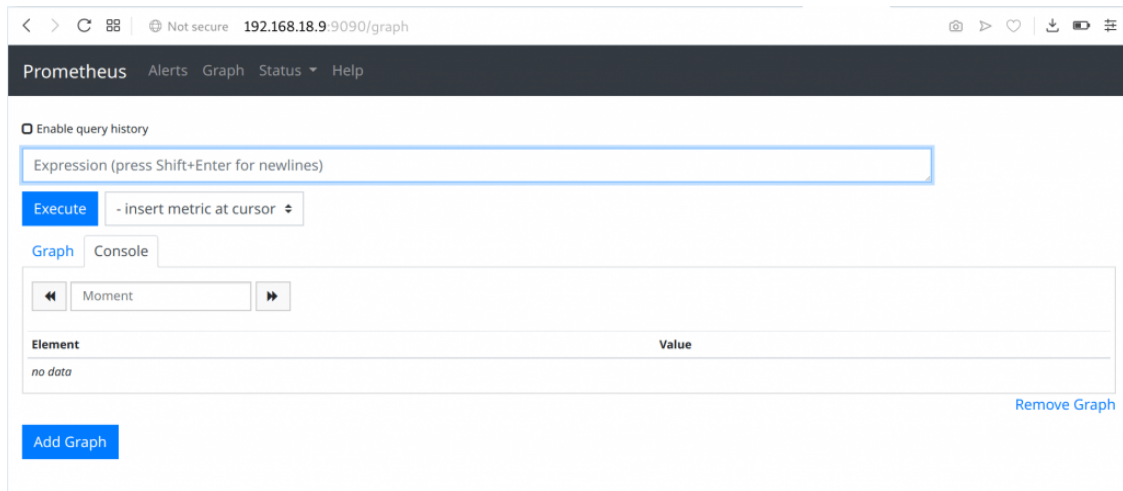
```
Jan 19 14:36:08 deb10 prometheus[1397]: level=info ts=2020-01-
19T14:36:08.959Z caller=main.go:334 vm_limits="(soft=unlimited,
hard=unlimited)"
Jan 19 14:36:08 deb10 prometheus[1397]: level=info ts=2020-01-
19T14:36:08.960Z caller=main.go:648 msg="Starting TSDB ..."
Jan 19 14:36:08 deb10 prometheus[1397]: level=info ts=2020-01-
19T14:36:08.964Z caller=head.go:584 component=tsdb msg="replaying WAL,
this may take awhile"
Jan 19 14:36:08 deb10 prometheus[1397]: level=info ts=2020-01-
19T14:36:08.964Z caller=web.go:506 component=web msg="Start listening
for connections" add
```

```
Jan 19 14:36:08 deb10 prometheus[1397]: level=info ts=2020-01-19T14:36:08.965Z caller=head.go:632 component=tsdb msg="WAL segment loaded" segment=0 maxS
Jan 19 14:36:08 deb10 prometheus[1397]: level=info ts=2020-01-19T14:36:08.966Z caller=main.go:663 fs_type=EXT4_SUPER_MAGIC
Jan 19 14:36:08 deb10 prometheus[1397]: level=info ts=2020-01-19T14:36:08.966Z caller=main.go:664 msg="TSDB started"
Jan 19 14:36:08 deb10 prometheus[1397]: level=info ts=2020-01-19T14:36:08.966Z caller=main.go:734 msg="Loading configuration file" filename=/etc/prometh
Jan 19 14:36:08 deb10 prometheus[1397]: level=info ts=2020-01-19T14:36:08.967Z caller=main.go:762 msg="Completed loading of configuration file" filename
Jan 19 14:36:08 deb10 prometheus[1397]: level=info ts=2020-01-19T14:36:08.967Z caller=main.go:617 msg="Server is ready to receive web requests."
```

If your server has a running firewall service, you'll need to open port *9090*.

```
sudo ufw allow 9090/tcp
```

Confirm that you can connect to port *9090* by access the Prometheus server IP address / DNS name in your web browser.



Next, we will cover installing exporters on nodes to be monitored and configuring targets on our Prometheus server so that we can scrap metrics and visualize with Grafana.

Prometheus Monitoring guides

[Monitoring Ceph Cluster with Prometheus and Grafana](#)

[Monitoring Apache Web server with Prometheus and Grafana](#)

[How to Monitor Linux Server Performance with Prometheus and Grafana in 5 minutes](#)

[How to Monitor BIND DNS server with Prometheus and Grafana](#)

[How to Monitor Redis Server with Prometheus and Grafana in 5 minutes](#)

[Monitoring MySQL / MariaDB with Prometheus in five minutes](#)

Monitor Etcd cluster with Prometheus

You can support us by downloading this article as PDF from the Link below.

 [Download the guide as PDF](#)

Josphat Mutai

<https://computingforgeeks.com/>

Founder of Computingforgeeks. Expertise in Virtualization, Cloud, Linux/UNIX Administration, Automation, Storage Systems, Containers, Server Clustering e.t.c.

in

