1) Grafana:

Ubuntu and Debian(64 Bit)
SHA256: f9315c633ae47f956877427fd4b8927a5580f8d7ac4e07fbc24a06bdfc153e39

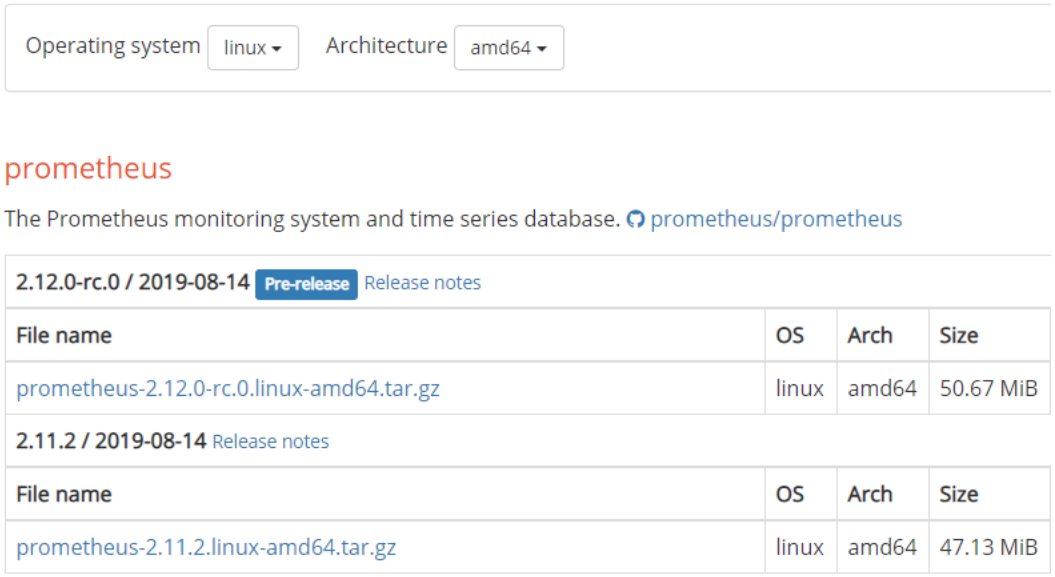
```
sudo apt-get install -y adduser libfontconfig1
wget https://dl.grafana.com/enterprise/release/grafana-enterprise_8.5.5_amd64.deb
sudo dpkg -i grafana-enterprise_8.5.5_amd64.deb
```

- Using
  By default, Grafana will be listening on http://localhost:3000. The default login is "admin" / "admin".

Site for Reference: https://grafana.com/grafana/download?pg=oss-graf&plcmt=resources

2) Prometheus:

a – Download Prometheus
First, head over to the Prometheus downloads page.
Make sure to filter for your operating system and your CPU architecture (in my case Linux and amd64)



Make sure to select the latest stable version, and not the "rc" one, as it is not considered stable enough for now.

Download the archive on your system by **clicking** on the archive, or by running a **wget** command if you are using the Terminal.

```
$ wget https://github.com/prometheus/prometheus/releases/download/v2.11.2/prometheus-2.11.2.linux-amd64.tar.gz
```

You should now have the tar.gz file on your system.

Untar it to extract the files in the archive.

```
$ tar xvzf prometheus-2.11.2.linux-amd64.tar.gz
```

```
devconnected@debian-10:~/Prometheus$ tar xvzf prometheus-2.11.2.linux-amd64.tar.gz
prometheus-2.11.2.linux-amd64/
prometheus-2.11.2.linux-amd64/NOTICE
prometheus-2.11.2.linux-amd64/prometheus.yml
prometheus-2.11.2.linux-amd64/LICENSE
prometheus-2.11.2.linux-amd64/promtool
prometheus-2.11.2.linux-amd64/prometheus
prometheus-2.11.2.linux-amd64/consoles/
prometheus-2.11.2.linux-amd64/consoles/index.html.example
prometheus-2.11.2.linux-amd64/consoles/node-cpu.html
prometheus-2.11.2.linux-amd64/consoles/node.html
prometheus-2.11.2.linux-amd64/consoles/prometheus-overview.html
prometheus-2.11.2.linux-amd64/consoles/node-overview.html
prometheus-2.11.2.linux-amd64/consoles/node-disk.html
prometheus-2.11.2.linux-amd64/consoles/prometheus.html
prometheus-2.11.2.linux-amd64/console_libraries/
prometheus-2.11.2.linux-amd64/console_libraries/menu.lib
prometheus-2.11.2.linux-amd64/console_libraries/prom.lib
```

The archive contains many important files, but here is the main ones you need to know.

- **prometheus.yml**: the configuration file for Prometheus. This is the file that you are going to modify in order to tweak your Prometheus server, for example to change the scraping interval or to configure custom alerts;
- **prometheus**: the binary for your Prometheus server. This is the command that you are going to execute to launch a Prometheus instance on your Linux box;
- **promtool**: this is a command that you can run to verify your Prometheus configuration.
  We are not going to execute directly the Prometheus, instead we are going to configure it as a service.

It will bring more robustness and reliability in case our Prometheus server were to stop suddenly.

b – Configuring Prometheus as a service
First of all, for security purposes, you are going to create **a Prometheus user with a Prometheus group.**

```
$ sudo useradd -rs /bin/false prometheus
```

Make sure to move the binaries to your local bin directory.

I stored my binaries in a Prometheus folder, located on my home directory.

Here's the command to move them to the bin directory.

```
$ cd Prometheus/prometheus-2.11.2.linux-amd64/
$ sudo cp prometheus promtool /usr/local/bin
```

Give **permissions** to the Prometheus user for the prometheus binary.

```
$ sudo chown prometheus:prometheus /usr/local/bin/prometheus
```

Create a folder in the */etc* folder for Prometheus and move the console files, console libraries and the prometheus configuration file to this newly created folder.

```
$ sudo mkdir /etc/prometheus
$ sudo cp -R consoles/ console_libraries/ prometheus.yml /etc/prometheus
```

Create a **data** folder at the root directory, with a prometheus folder inside.

```
$ sudo mkdir -p data/prometheus
```

Give the correct **permissions** to those folders recursively.

```
$ sudo chown -R prometheus:prometheus data/prometheus /etc/prometheus/*
```

You are now set **to create a Prometheus service**.
Head over to the */lib/systemd/system* folder and create a new file named prometheus.service

```
$ cd /lib/systemd/system
$ sudo touch prometheus.service
```

In order to see the different launch options for Prometheus, you can run the prometheus command with a h flag.

```
$ prometheus -h
usage: prometheus [<flags>]

The Prometheus monitoring server

Flags:
  -h, --help                     Show context-sensitive help (also try --help-long and
                                 --help-man).
      --version                  Show application version.
      --config.file="prometheus.yml"
                                 Prometheus configuration file path.
      --web.listen-address="0.0.0.0:9090"
```

Edit the newly created file, and paste the following content inside.

```
$ sudo nano prometheus.service

[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

[Service]
Type=simple
User=prometheus
Group=prometheus
ExecStart=/usr/local/bin/prometheus \
  --config.file=/etc/prometheus/prometheus.yml \
  --storage.tsdb.path="/data/prometheus" \
  --web.console.templates=/etc/prometheus/consoles \
  --web.console.libraries=/etc/prometheus/console_libraries \
  --web.listen-address=0.0.0.0:9090 \
  --web.enable-admin-api
```

```
Restart=always

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

Save your file, **enable** your service at startup, and **start** your service.

```
$ sudo systemctl enable prometheus
$ sudo systemctl start prometheus
```

Site Reference: https://devconnected.com/how-to-setup-grafana-and-prometheus-on-linux/

Creating a Prometheus data source

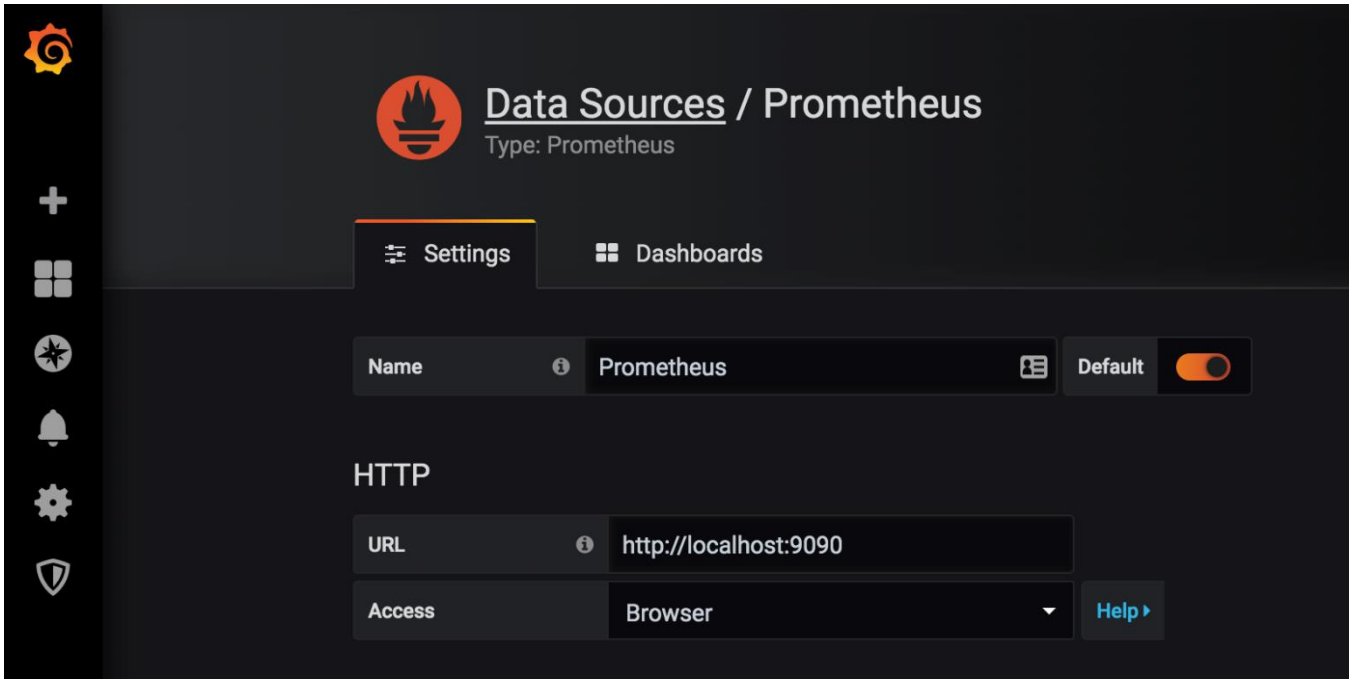
```
devconnected@debian-10:/lib/systemd/system$ sudo systemctl status prometheus
● prometheus.service - Prometheus Server
   Loaded: loaded (/lib/systemd/system/prometheus.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2019-08-15 08:55:31 EDT; 2min 12s ago
 Main PID: 4047 (prometheus)
    Tasks: 7 (limit: 4915)
   Memory: 21.7M
   CGroup: /system.slice/prometheus.service
           └─4047 /usr/local/bin/prometheus --config.file=/etc/prometheus/prometheus.yml --web.console.templates=/etc/prometheus/consoles --web.console.

Aug 15 08:55:31 debian-10 prometheus[4047]: level=info ts=2019-08-15T12:55:31.765Z caller=main.go:331 host_details="(Linux 4.19.0-5-amd64 #1 SMP Debian
Aug 15 08:55:31 debian-10 prometheus[4047]: level=info ts=2019-08-15T12:55:31.766Z caller=main.go:332 fd_limits="(soft=1024, hard=524288)"
Aug 15 08:55:31 debian-10 prometheus[4047]: level=info ts=2019-08-15T12:55:31.766Z caller=main.go:333 vm_limits="(soft=unlimited, hard=unlimited)"
Aug 15 08:55:31 debian-10 prometheus[4047]: level=info ts=2019-08-15T12:55:31.771Z caller=main.go:652 msg="Starting TSDB ..."
Aug 15 08:55:31 debian-10 prometheus[4047]: level=info ts=2019-08-15T12:55:31.786Z caller=web.go:448 component=web msg="Start listening for connections"
Aug 15 08:55:31 debian-10 prometheus[4047]: level=info ts=2019-08-15T12:55:31.787Z caller=main.go:667 fs_type=EXT4_SUPER_MAGIC
Aug 15 08:55:31 debian-10 prometheus[4047]: level=info ts=2019-08-15T12:55:31.789Z caller=main.go:668 msg="TSDB started"
Aug 15 08:55:31 debian-10 prometheus[4047]: level=info ts=2019-08-15T12:55:31.790Z caller=main.go:738 msg="Loading configuration file" filename=/etc/pro
Aug 15 08:55:31 debian-10 prometheus[4047]: level=info ts=2019-08-15T12:55:31.790Z caller=main.go:766 msg="Completed loading of configuration file" file
Aug 15 08:55:31 debian-10 prometheus[4047]: level=info ts=2019-08-15T12:55:31.791Z caller=main.go:621 msg="Server is ready to receive web requests."
lines 1-19/19 (END)
```

To create a Prometheus data source in Grafana:

1. Click on the "cogwheel" in the sidebar to open the Configuration menu.
2. Click on "Data Sources".
3. Click on "Add data source".
4. Select "Prometheus" as the type.
5. Set the appropriate Prometheus server URL (for example, `http://localhost:9090/`)
6. Adjust other data source settings as desired (for example, choosing the right Access method).
7. Click "Save & Test" to save the new data source.

The following shows an example data source configuration:



3) Tomcat Monitoring on Grafana:

a) Download Blackbox exporter and node exporter for monitor website
   Site: **https://prometheus.io/download/**
b) Copy blackbox exporter and node exporter tar file in /root/
c) Untar the blackbox and node exporter
   1. tar xvzf blackbox_exporter-0.21.0.linux-amd64.tar.gz
   2. tar xvzf node_exporter-1.3.1.darwin-amd64.tar.gz
d) cd node_exporter-*.*-amd64
e) ./node_exporter
f) curl http://localhost:9100/metrics
g) curl http://localhost:9100/metrics | grep "node_"
h) add contents in prometheus.yml file

```
global:
  scrape_interval: 15s


scrape_configs:
- job_name: node
  static_configs:
  - targets: ['localhost:9100']
```

Site reference: https://prometheus.io/docs/guides/node-exporter/

For blackbox exporter:

1. cd blackbox_exporter-0.21.0.linux-amd64.tar.gz
2. Add content in /etc/systemd/system/blackbox.service

[Unit]

Description=Prometheus Server

Documentation=https://prometheus.io/docs/introduction/overview/

After=network-online.target

[Service]

User=root

Restart=on-failure

ExecStart=/root/blackbox_exporter-0.21.0.linux-amd64/blackbox_exporter --config.file=/root/blackbox_exporter-0.21.0.linux-amd64/blackbox.yml

[Install]

WantedBy=multi-user.target

3. systemctl daemon-reload
4. systemctl start blackbox.service
5. systemctl status blackbox.service
6. add content in Prometheus.yml

```
- job_name: 'blackbox'
  metrics_path: /probe
  params:
   module: [http_2xx]  # Look for a HTTP 200 response.
  static_configs:
   - targets:
    - http://prometheus.io    # Target to probe with http.
    - https://prometheus.io   # Target to probe with https.
    - http://example.com:8080 # Target to probe with http on port 8080.
  relabel_configs:
   - source_labels: [__address__]
    target_label: __param_target
   - source_labels: [__param_target]
    target_label: instance
   - target_label: __address__
    replacement: 127.0.0.1:9115  # The blackbox exporter's real hostname:port.
```

7. systemctl restart prometheus.service
8. systemctl status prometheus.service
9. systemctl restart blackbox.service
10. systemctl restart tomcat.service
11. systemctl restart grafana-server.service
12. systemctl restart named

Site Reference: https://github.com/prometheus/blackbox_exporter