

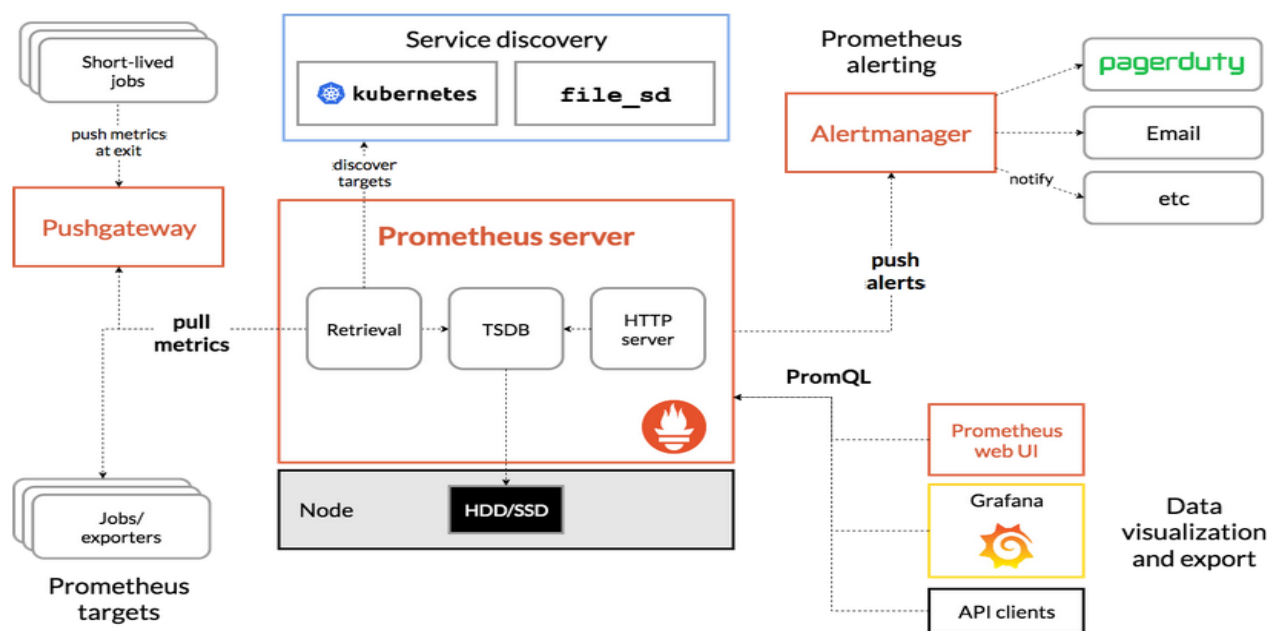


Prometheus

نرم افزار Prometheus یک سیستم قدرتمند مانیتورینگ می باشد که بیشتر قسمتهای آن با زبان برنامه نویسی Go نوشته شده است و به صورت Open Source منتشر می شود. با استفاده از نرم افزار مانیتورینگ Prometheus می توان metric های گوناگونی را از سرویس ها و سرورهای خود جمع آوری کنید و آنها را به صورت آنی و نمودارهای گرافیکی به نمایش بگذارید و با تنظیم قابلیت alert همواره سرویس های خود را کنترل و نظارت کنید.

معماری Prometheus :

روش کار Prometheus به این صورت می باشد که Prometheus Server تمامی metric ها را از exporter ها (برای سرویس های مختلف exporter های مختلفی وجود دارد) (یا push gateway واسط دریافت (pull) می کند و داده های دریافت شده را به صورت محلی ذخیره می کند و قوانینی را بر روی آنها اجرا می کند تا مجموعه ای از داده های جدید را از داده های موجود ذخیره کند یا برای آنها alert ایجاد کند. سپس می توان با استفاده از Grafana یا دیگر API ها، داده های جمع آوری شده را به صورت نمودارهای گرافیکی به نمایش گذاشت. در تصویر پایین نمایی کلی از معماری Prometheus را مشاهده می کنید:



برای نصب نرم افزار Prometheus ابتدا کفایست تا یک کاربر با نام دلخواه جهت اجرای سرویس Prometheus بر روی سیستم ایجاد کرد که در این مثال ما کاربری با نام prometheus را با استفاده از دستور زیر ایجاد می کنیم:

```
# useradd -m -s /bin/bash Prometheus
```

اکنون به کاربر prometheus وارد شوید و بسته ی prometheus را در پوشه ی خانگی کاربر دانلود کنید:

```
# su - Prometheus
```

```
$ wget -c
```

<https://github.com/prometheus/prometheus/releases/download/v2.7.1/prometheus-2.7.1.linux-amd64.tar.gz>

سپس فایل دانلود شده را از حالت فشرده خارج کنید و نام پوشه ی آن را به prometheus تغییر دهید:

```
$ tar -xzf prometheus-2.7.1.linux-amd64.tar.gz
```

```
$ mv prometheus-2.7.1.linux-amd64 prometheus
```

اکنون تمام فایل های prometheus در مسیر زیر قرار دارند:

```
/home/prometheus/prometheus
```

پیکربندی Prometheus برای Systemd :

برای کنترل سرویس prometheus باید یک فایل سرویس Systemd ایجاد کرد که برای این کار کفایست تا با کاربر root فایل prometheus.service را ایجاد کرد:

```
# vi /etc/systemd/system/prometheus.service
```

```
1 [Unit]
```

```
2 Description=Prometheus Server
```

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

```
4 After=network-online.target
```

```
5
```

```
6 [Service]
```

```
7 User=prometheus
```

```
8 Restart=on-failure
```

```
9 #Change this line if you download the
```

```

10 #Prometheus on different path user
11 ExecStart=/home/prometheus/prometheus/prometheus \
12 --config.file=/home/prometheus/prometheus/prometheus.yml \
13 --storage.tsdb.path=/home/prometheus/prometheus/data
14
15 [Install]
16 WantedBy=multi-user.target

```

اکنون یکبار `systemd` را reload کنید:

```
# systemctl daemon-reload
```

اکنون سرویس `prometheus` را start و enable کنید:

```
# systemctl start prometheus
```

```
# systemctl enable prometheus
```

برای بررسی وضعیت سرویس `prometheus` می توان از دستور زیر استفاده کرد:

```
# systemctl status prometheus
```

یا اینکه می توان از دستور زیر استفاده کرد:

```
# netstat -plntu
```

یک نمونه خروجی از دستور گفته شده را در پایین مشاهده می کنید :

```

[root@prometheus ~]# netstat -plntu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      3271/sshd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      3381/master
tcp6       0      0 :::22                  :::*                    LISTEN      3271/sshd
tcp6       0      0 :::1:25                 :::*                    LISTEN      3381/master
tcp6       0      0 :::9090                 :::*                    LISTEN      3793/prometheus
udp        0      0 0.0.0.0:68              0.0.0.0:*               *          3137/dhclient
[root@prometheus ~]#

```

همانطور که در تصویر بالا مشاهده می کنید سرویس `prometheus` بر روی پورت 9090 در حال اجرا می باشد.

پیکربندی: FirewallD

اگر بر روی سیستم خود از FirewallD استفاده می کنید باید پورت 9090 که مربوط به سرویس prometheus می باشد را باز کنید:

```
1 # firewall-cmd --zone=public --add-port=9090/tcp
```

```
2 # firewall-cmd --zone=public --add-port=9090/tcp --permanent
```

نصب پرومیتئوس با داکر :

مسئله برای اینکار نیاز است که Docker بر روی سرور نصب شده باشد

برای راه اندازی Prometheus با Docker کافیست تا دستور زیر را اجرا کنید:

```
1#docker run -d -p 9090:9090 --name prometheus --restart always -v  
/path/to/prometheus.yml:/etc/prometheus/prometheus.yml prom/prometheus
```

برای تعیین یک مسیر جداگانه برای داده های Prometheus می توان یک پوشه با نام دلخواه در یک مسیر دلخواه ایجاد کرد:

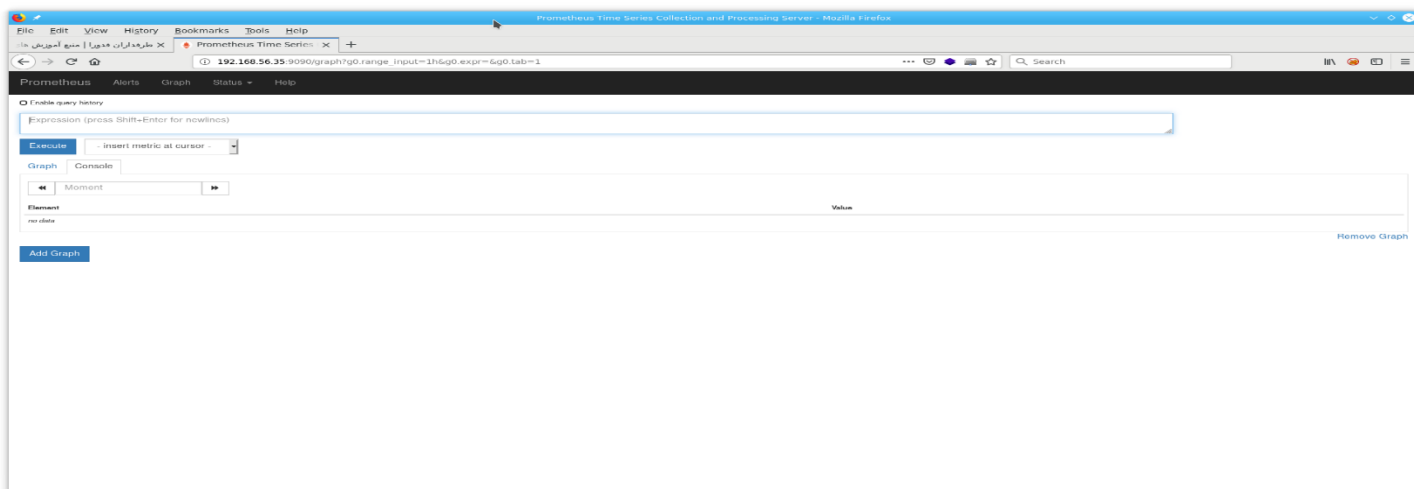
سپس برای ایجاد کانتینر و معرفی آن مسیرها به عنوان volume دستور زیر را اجرا کنید:

```
1# docker run -d -p 9090:9090 --name prometheus --restart always -v  
/home/prometheus/prometheus/prometheus.yml:/etc/prometheus/prometheus.yml -v  
/opt/prometheus:/prometheus prom/prometheus
```

با هر روشی که Prometheus را نصب کردید. اکنون یک مرورگر وب باز کنید و آدرس سرور خود به همراه پورت 9090 را در آن وارد کنید:

<http://192.168.56.35:9090/graph>

اکنون پنل وب prometheus را باید مشاهده کنید :



برای هر سرویس و هر منظوری **exporter** مخصوص وجود دارد که برخی از آنها رسمی و برخی دیگر غیر رسمی هستند. برای دیدن و دریافت **exporter** ها می توانید به لینک پایین مراجعه کنید:

<https://prometheus.io/docs/instrumenting/exporters/>

در این قسمت قصد داریم تا **node_exporter** که یکی از **exporter** های **prometheus** برای **metric** های سرور و سخت افزار می باشد و قابل نصب بر روی سیستم های یونیکسی و لینوکسی می باشد را نصب کنیم. ما قصد داریم تا **node_exporter** را بر روی همان سرور **prometheus** نصب کنیم و **metric** های آن را دریافت و سرور **prometheus** را مانیتور کنیم. کاربران سیستم عامل ویندوز باید از **WMI exporter** استفاده کنند که لینک پروژه ی آن را در پایین مشاهده می کنید:

https://github.com/martinlindhe/wmi_exporter

نصب و پیکربندی: **node_exporter**

به این منظور به کاربر **prometheus** وارد شود و **node_exporter** را دانلود کنید:

```
# su - prometheus
```

```
$ wget -c
```

https://github.com/prometheus/node_exporter/releases/download/v0.17.0/node_exporter-0.17.0.linux-amd64.tar.gz

اکنون فایل دانلود شده را از حالت فشرده خارج کنید و آن را به **node_exporter** تغییر نام دهید:

```
$ tar -xzf node_exporter-0.17.0.linux-amd64.tar.gz
```

```
$ mv node_exporter-0.17.0.linux-amd64 node_exporter
```

اکنون جهت کنترل سرویس **node_exporter** با **systemd** کفایت تا یک **service** فایل برای آن ایجاد کنیم. به این منظور با کاربر **root** فایل زیر را ایجاد کنید:

```
# vi /etc/systemd/system/node_exporter.service
```

سپس خطوط زیر را درون فایل بنویسید:

```
[Unit]
```

```
Description=Node Exporter
```

```
Wants=network-online.target
```

```
After=network-online.target
```

[Service]

User=prometheus

ExecStart=/home/prometheus/node_exporter/node_exporter

[Install]

WantedBy=default.target

اکنون یکبار **systemd** را reload کنید:

```
# systemctl daemon-reload
```

اکنون برای **start** و **enable** کردن سرویس **node_exporter** کفایت تا دستورهای زیر را اجرا کنید:

```
# systemctl start node_exporter
```

```
# systemctl enable node_exporter
```

برای بررسی وضعیت سرویس **node_exporter** می توانید از دستور پایین استفاده کنید:

```
# systemctl status node_exporter
```

یا اینکه می توان از دستور زیر استفاده کرد:

```
# netstat -plntu
```

نمونه ای از دستور اجرا شده را در تصویر پایین مشاهده می کنید:

```
[root@prometheus ~]# netstat -plntu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      3271/sshd
tcp        0      0 127.0.0.1:25             0.0.0.0:*               LISTEN      3381/master
tcp6       0      0 :::9100                  :::*                    LISTEN      7193/node_exporter
tcp6       0      0 :::22                    :::*                    LISTEN      3271/sshd
tcp6       0      0 :::1:25                  :::*                    LISTEN      3381/master
tcp6       0      0 :::9090                   :::*                    LISTEN      3793/prometheus
udp        0      0 0.0.0.0:68              0.0.0.0:*               LISTEN      3137/dhclient
[root@prometheus ~]#
```

همانطور که در تصویر بالا مشاهده می کنید سرویس **node_exporter** بر روی پورت **9100** در حال اجرا می باشد.

اجرا کردن node_exporter با استفاده از Docker :

برای اجرا کردن **node-exporter** با استفاده از **Docker** می توانید از دستور پایین استفاده کنید. نکته اینکه اجرای **node-exporter** بر روی **Docker** در محیط های عملیاتی (**production**) پیشنهاد نمی شود.

```
1# docker run -d --name node_exporter --restart always -v "/proc:/host/proc" -v
"/sys:/host/sys" -v "":"/rootfs" --net="host" prom/node-exporter --path.procfs /host/proc --
path.sysfs /host/sys --collector.filesystem.ignored-mount-points
"^/(sys|proc|dev|host|etc) ($|/)"
```

اکنون در این قسمت قصد داریم تا `node_exporter` را به `Prometheus server` اضافه کنیم.

اضافه کردن `node_exporter` به `Prometheus Server`:

برای اضافه کردن `node_exporter` به `Prometheus Server` ابتدا به کاربر `prometheus` وارد شوید:

```
# su - prometheus
```

اکنون فایل پیکربندی `prometheus` که نام آن `prometheus.yml` می باشد را باز کنید:

```
$ vi prometheus/prometheus.yml
```

اکنون در زیر خط `scrape_configs` یک `job_name` به نام `node_exporter` ایجاد کنید. تنظیمات آن به صورت زیر می باشد:

```
1 - job_name: 'node_exporter'
2 static_configs:
3 - targets: ['localhost:9100']
```

نکته اینکه بجای `localhost` می توانید `IP` سرور را بنویسید و مسلماً زمانی که `node_exporter` بر روی سرور دیگری می باشد باید `IP` آن سرور نوشته شود. در تصویر پایین نمونه ای از تنظیم گفته شده نمایش داده شده است:

```
# 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']

  - job_name: 'node_exporter'
    static_configs:
      - targets: ['localhost:9100']
```

اکنون کافیسیت تا سرویس `prometheus` را یک بار `restart` کنید:

```
# systemctl restart prometheus
```

نکته اینکه اگر `Prometheus server` را با استفاده از `Docker` اجرا کرده اید، کافیسیت تا یک بار `docker container` را `restart` کنید:

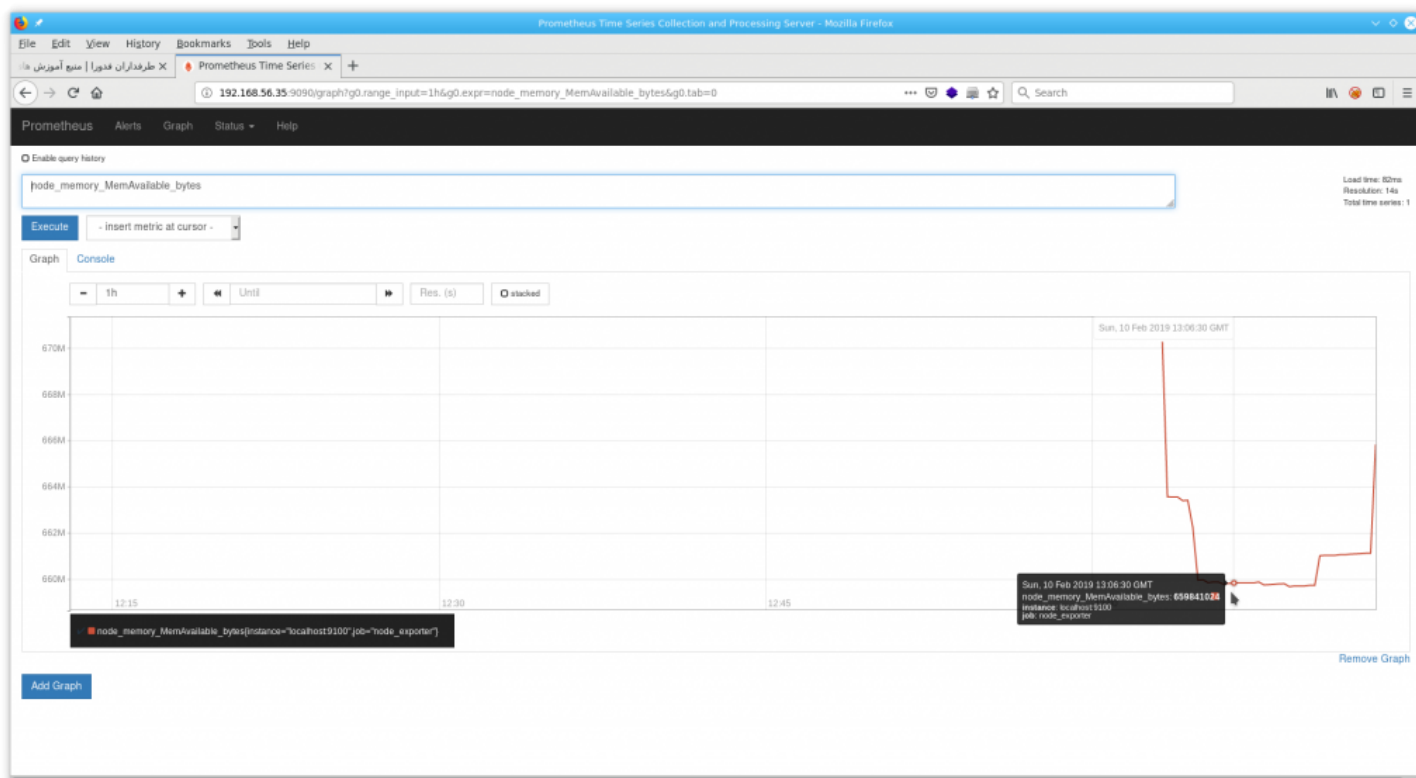
docker restart prometheus-server-container

آزمایش Prometheus و: node_exporter

برای این منظور کافیست تا آدرس Prometheus Server را در مرورگر وب وارد کنید:

<http://192.168.56.35:9090>

اکنون عبارت `node_memory_MemAvailable_bytes` را در `query field` بنویسید و روی دکمه `Execute` کلیک کنید و چیزی مانند تصویر پایین را باید مشاهده کنید:



با اینکار شما باید میزان `memory` قابل دسترس در `localhost:9100` را دریافت کنید.

اگر بخواهید به `raw data` از `node_exporter` دسترسی داشته باشید، باید پورت `9100` را بر روی `Firewalld` باز کنید:

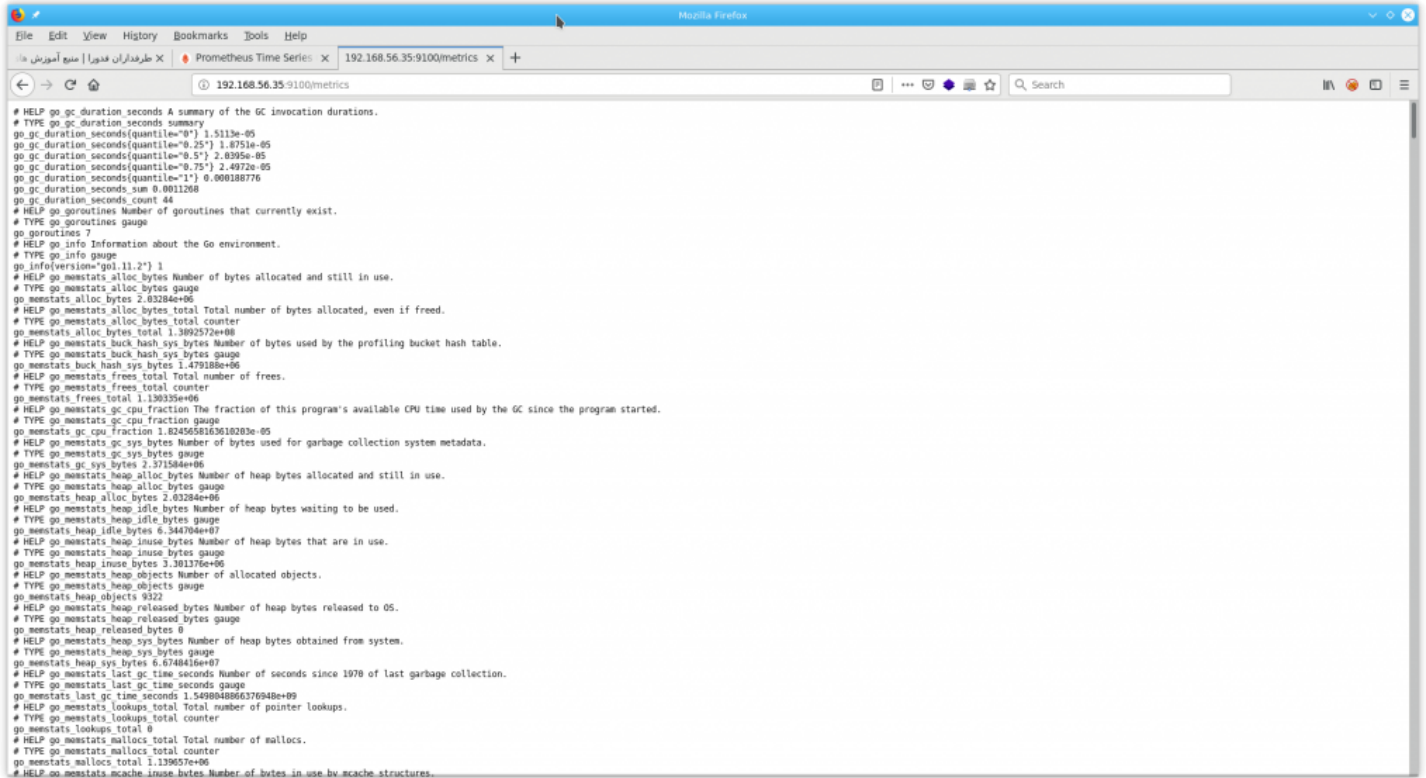
```
1 # firewall-cmd --zone=public --add-port=9100/tcp
```

```
2 # firewall-cmd --zone=public --add-port=9100/tcp --permanent
```

اکنون کافیست تا آدرس زیر را در مرورگر وب خود وارد کنید:

<http://192.168.56.35:9100/metrics>

اکنون مانند تصویر پایین باید raw data مربوط به node_exporter را مشاهده کنید:

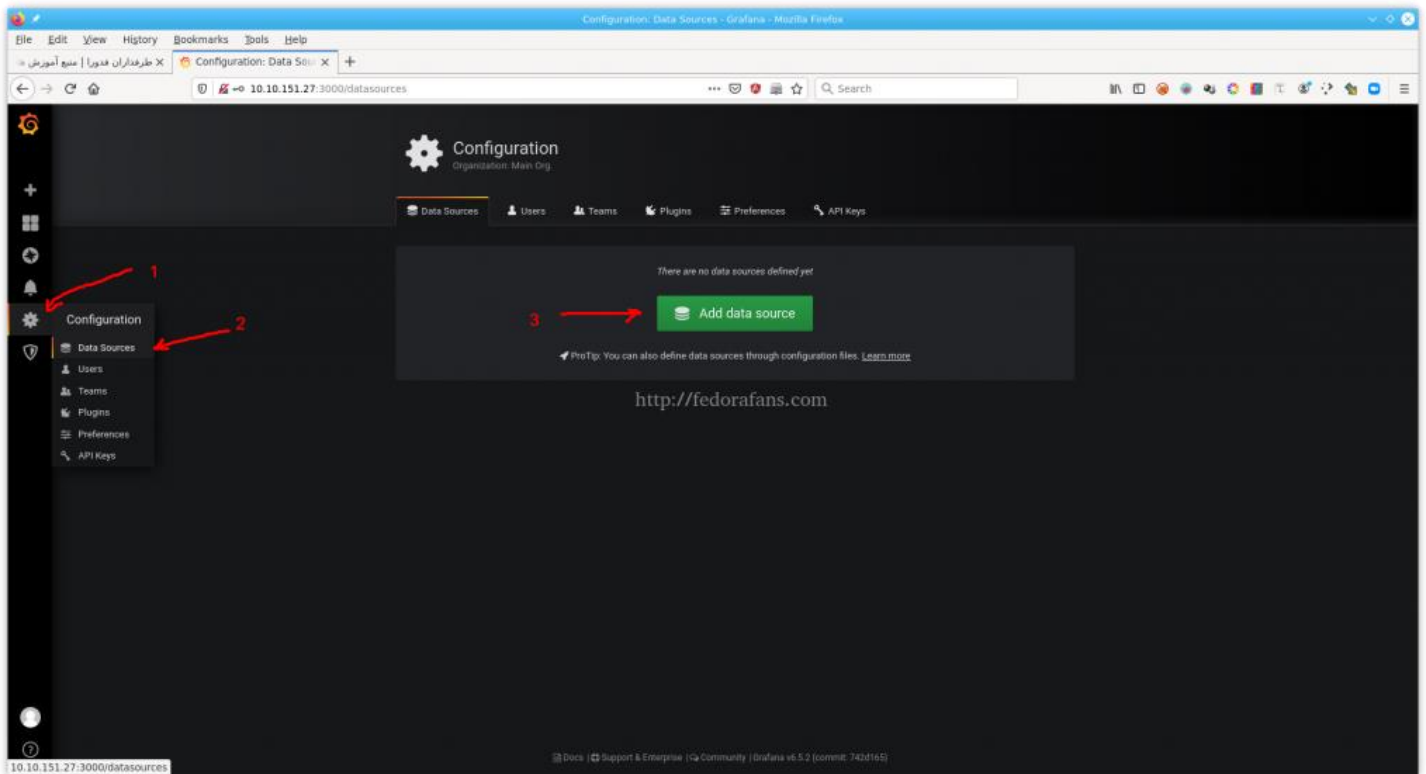


قصد داریم تا Prometheus Server را با Grafana به هم متصل کنیم.

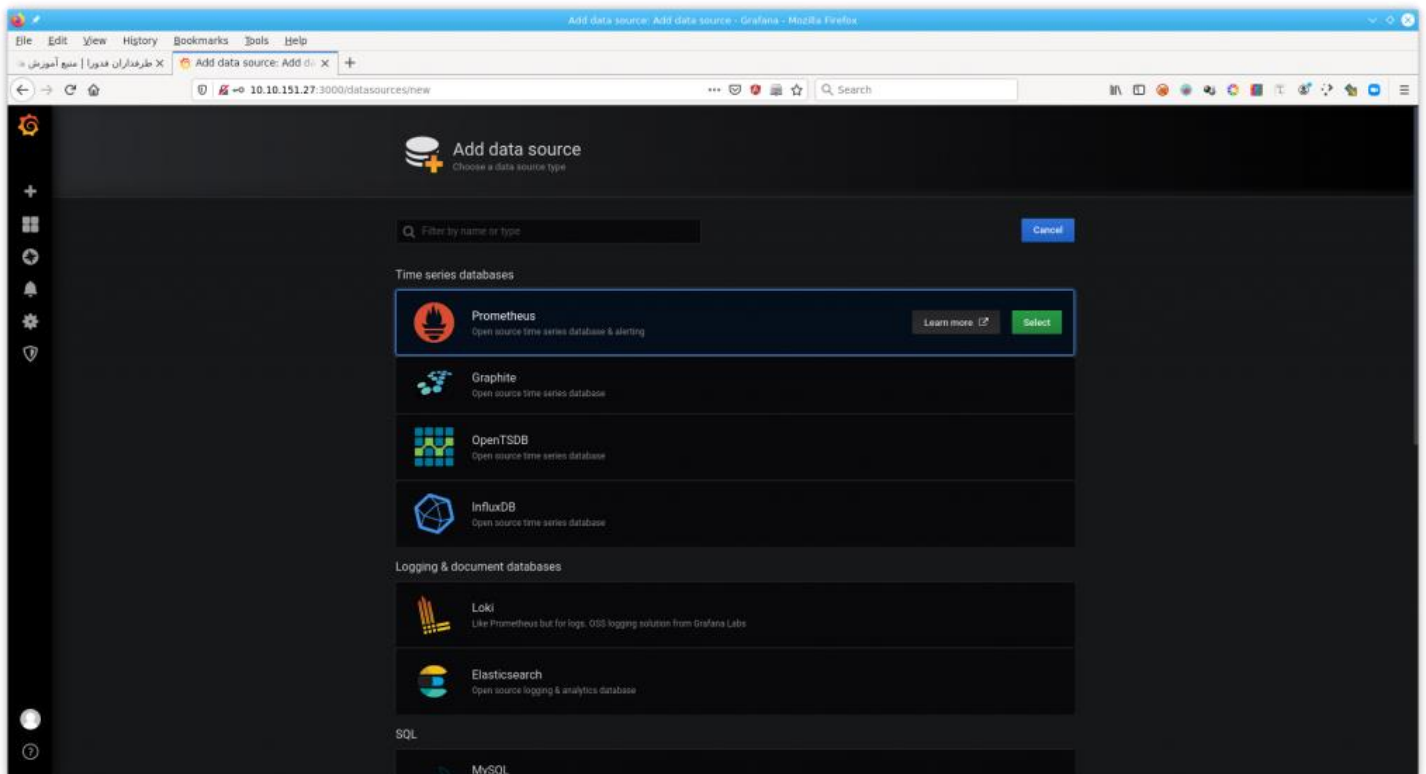
همانطور که می دانید Grafana یک ابزار برای visualize کردن داده ها و یا metric ها می باشد که اکنون با اتصال آن به Prometheus می توانیم داده های Prometheus Server را در Grafana به صورت نمودارها و گراف های زیبا نمایش دهیم.

اتصال Prometheus server و: Grafana

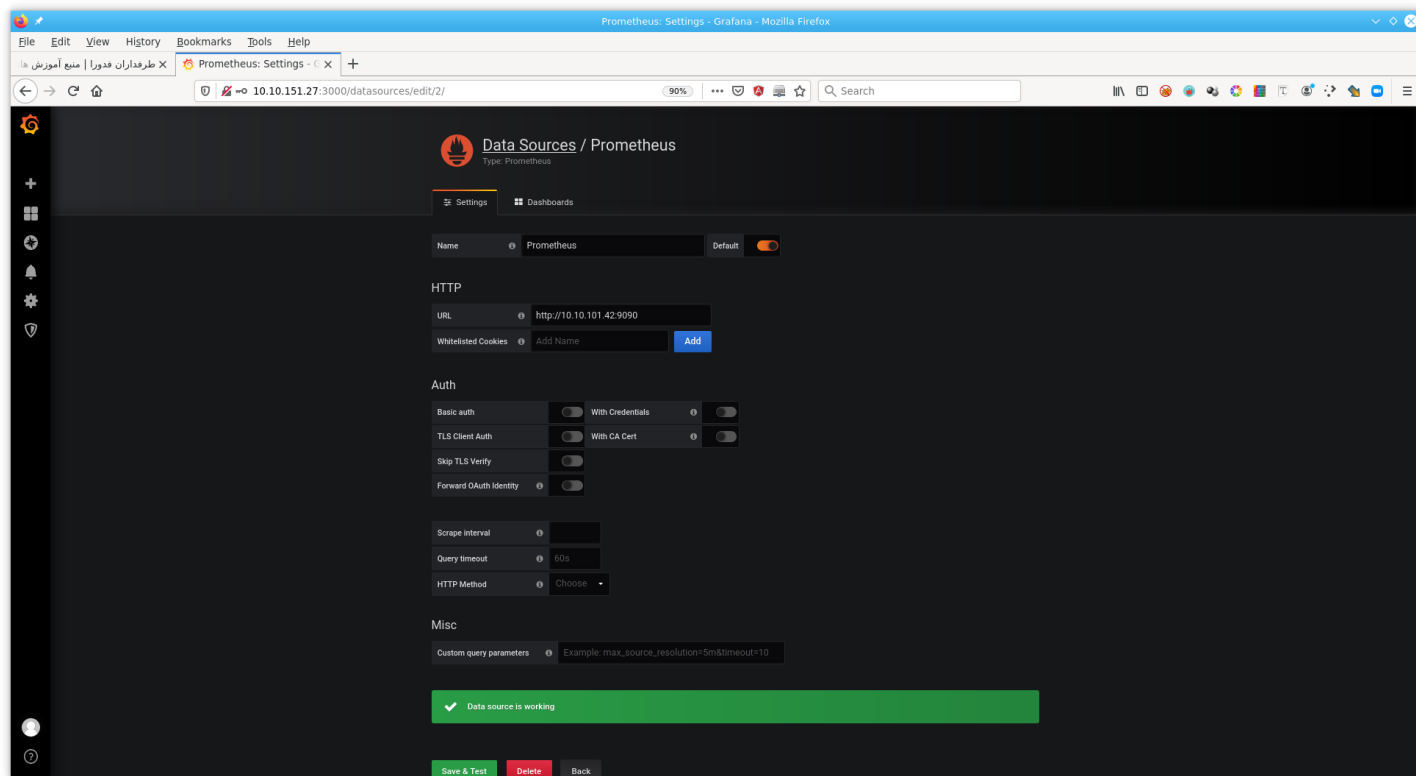
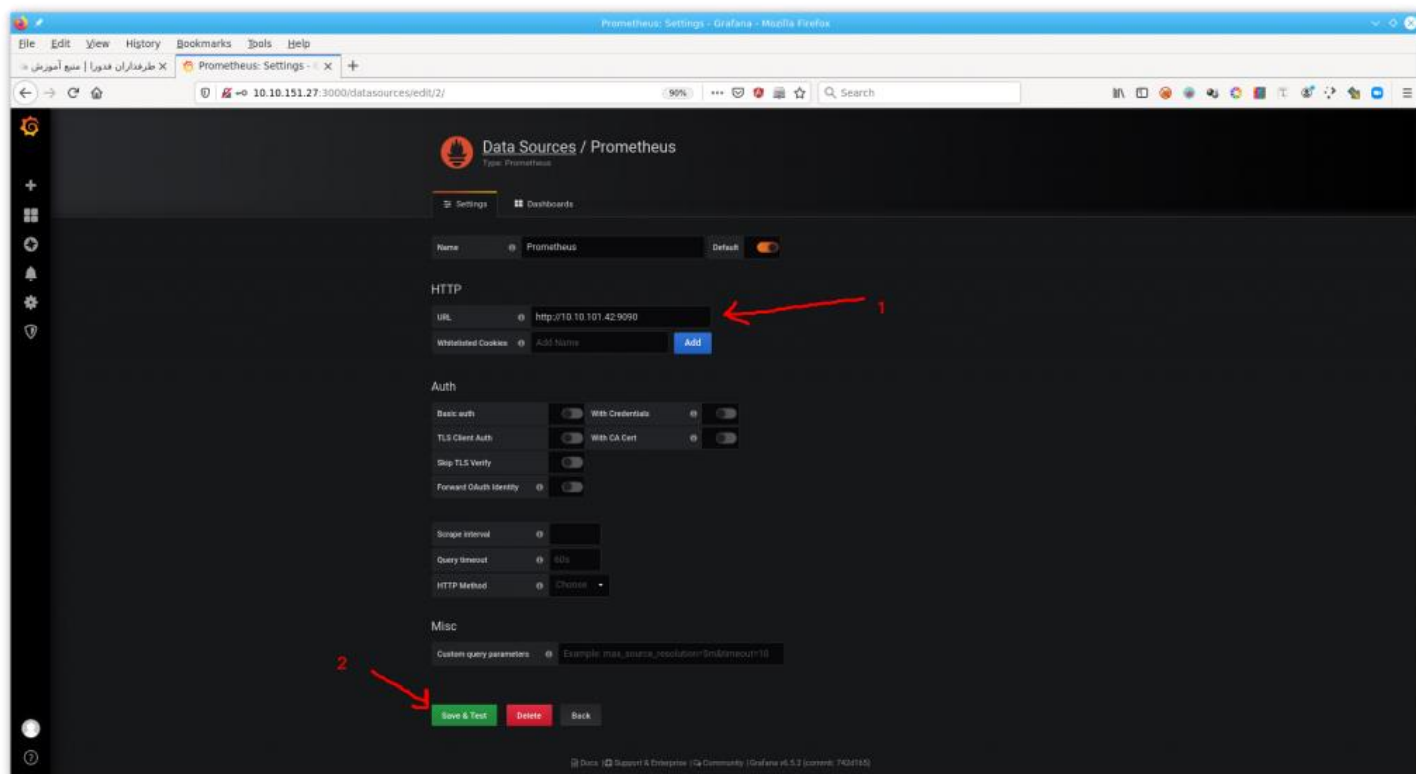
برای اینکار به پنل مدیریتی تحت وب Grafana وارد شوید و مانند تصویر پایین به قسمت Configuration و سپس Data Sources بروید، سپس جهت اضافه کردن Data Source روی دکمه ی Add data source کلیک کنید:



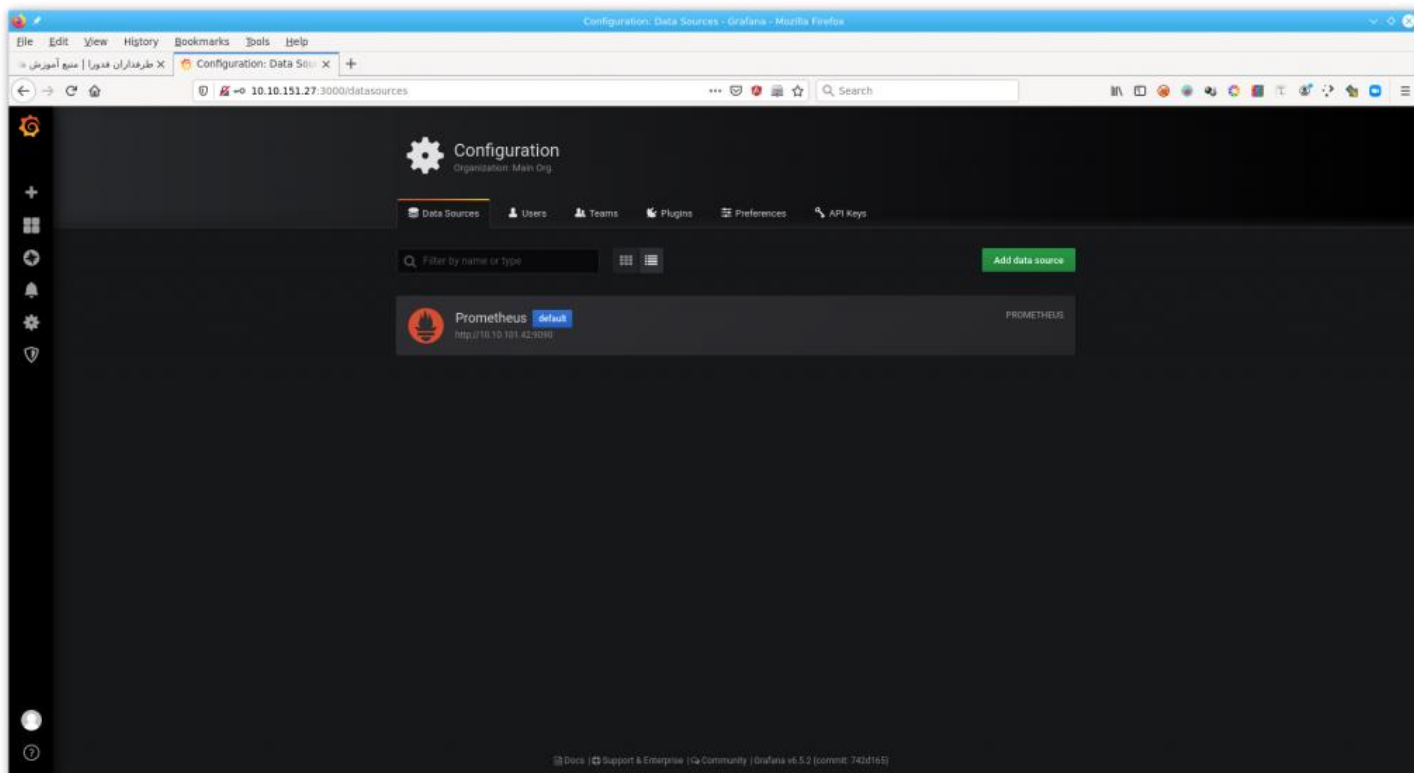
سپس وارد صفحه ی پایین می شوید که در این قسمت مانند تصویر پایین Prometheus را انتخاب کنید:



اکنون پس از انتخاب Prometheus به صفحه ای مانند تصویر پایین وارد خواهید شد که باید آدرس و اطلاعات مربوط به Prometheus server خود را وارد کنید و سپس بر روی دکمه ی Save & Test کلیک کنید:

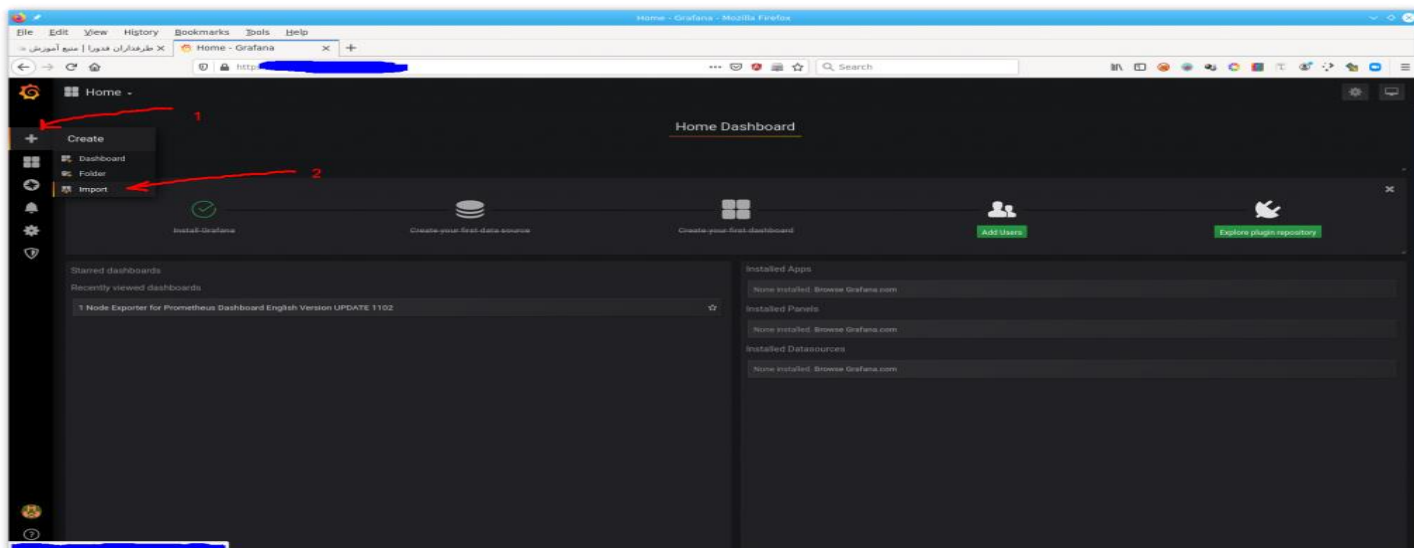


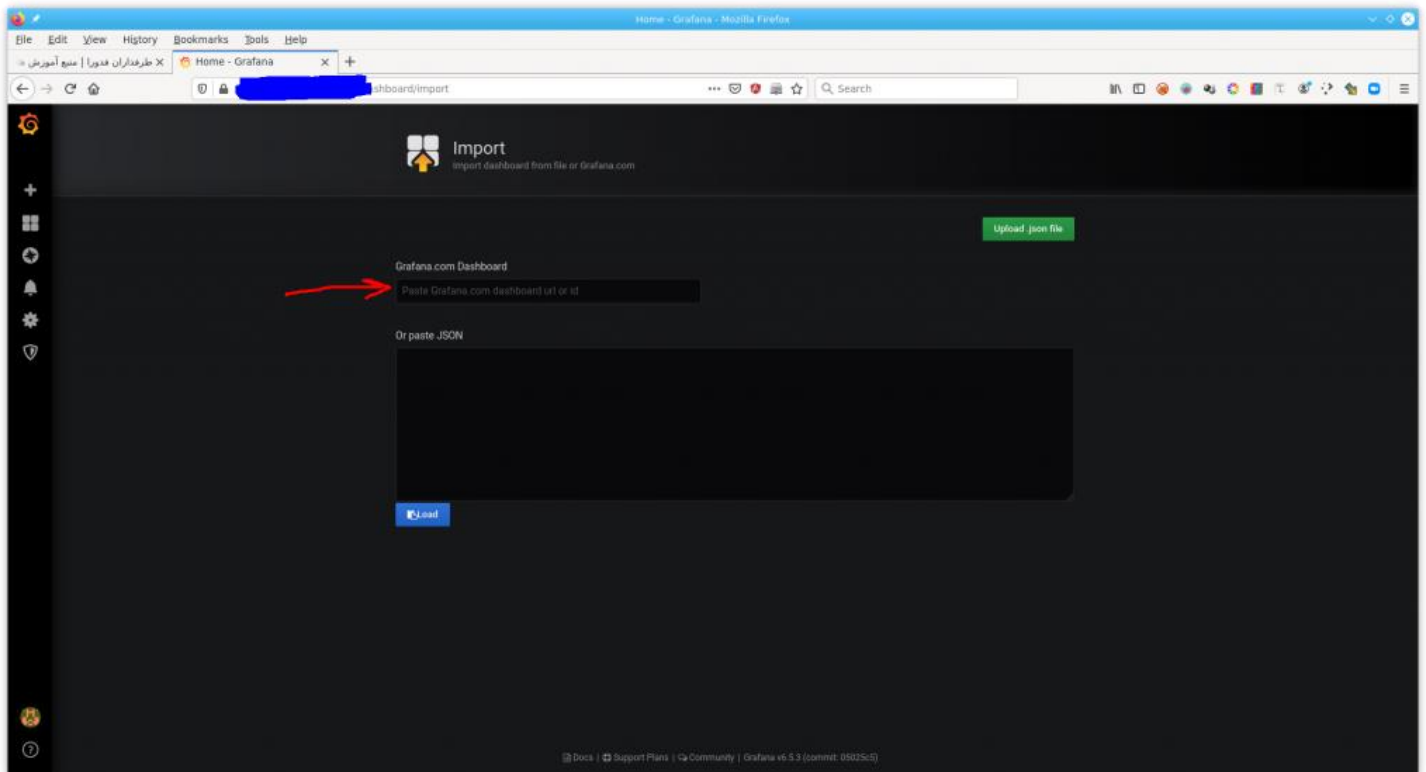
پس از انجام اینکارها همانطور که در تصویر پایین مشاهده می کنید Prometheus server به عنوان یک Data Source به Grafana اضافه شد و اکنون می توانید پنل های مانیتورینگ خود را ایجاد و یا Import کنید.



اضافه کردن پنل مانیتورینگ برای Node Exporter ها:

برای ایجاد پنل ها شما می توانید آنها را ایجاد کنید یا اینکه از پنل های آماده استفاده کنید که هر کدام برای اهداف و Exporter های خاصی کاربرد دارند. در این قسمت قصد داریم تا از یک پنل آماده برای Node Exporter استفاده کنیم. برای اضافه کردن پنل ابتدا مانند تصویر پایین بر روی دکمه ی Import کلیک کنید :





اکنون برای Import کردن پنل مورد نظر خود می توانید ID آن و یا Link آن را در قسمت مشخص شده وارد کنید و یا اینکه محتویات فایل JSON آن را وارد کنید. همواره شما می توانید پنل های خود را از مخزن وب سایت Grafana پیدا کنید که در این مرحله ما قصد داریم تا از پنل زیر استفاده کنیم:

<https://grafana.com/grafana/dashboards/11074>

همچنین شما می توانید فایل پنل را دانلود و سپس آن را Import کنید. به هر حال پس از اضافه کردن پنل می توانید از آن استفاده کنید:

