

## Project 3

# How to Integrate Linux Server for Grafana Cloud:

## Step #1: Install Grafana Agent on Ubuntu

The Grafana Agent is a lightweight, open-source agent that collects metrics, logs, and traces from your [Linux servers](#) and sends them to [Grafana Cloud](#). So first create the directory for the apt keyrings

```
sudo mkdir -p /etc/apt/keyrings/
```

```
ubuntu@ip-172-31-8-191:~$ sudo mkdir -p /etc/apt/keyrings/
```

```
wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee  
/etc/apt/keyrings/grafana.gpg > /dev/null
```

```
ubuntu@ip-172-31-8-191:~$ wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sud
```

Add Grafana package repository using following command.

```
echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable main"  
| sudo tee /etc/apt/sources.list.d/grafana.list
```

```
ubuntu@ip-172-31-8-191:~$ echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable main" | sudo tee /etc/apt/sources.list.d/grafana  
list  
deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable main
```

Update the repositories

```
sudo apt-get update
```

```
ubuntu@ip-172-31-8-191:~$ sudo apt-get update
```

After updating the repository. Install the Grafana Agent

```
sudo apt-get install grafana-agent
```

```

ubuntu@ip-172-31-8-191:~$ sudo apt-get install grafana-agent
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  grafana-agent
0 upgraded, 1 newly installed, 0 to remove and 43 not upgraded.
Need to get 118 MB of archives.
After this operation, 411 MB of additional disk space will be used.
Get:1 https://apt.grafana.com stable/main amd64 grafana-agent amd64 0.40.3-1 [118 MB]
Fetched 118 MB in 11s (10.4 MB/s)
Selecting previously unselected package grafana-agent.
(Reading database ... 65273 files and directories currently installed.)
Preparing to unpack .../grafana-agent_0.40.3-1_amd64.deb ...
Unpacking grafana-agent (0.40.3-1) ...
Setting up grafana-agent (0.40.3-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
Start the grafana agent by running the following command

```

```
sudo systemctl start grafana-agent
```

```
ubuntu@ip-172-31-8-191:~$ sudo systemctl start grafana-agent
```

After this enable the grafana agent

```
sudo systemctl enable grafana-agent
```

```

ubuntu@ip-172-31-8-191:~$ sudo systemctl enable grafana-agent.service
Created symlink /etc/systemd/system/multi-user.target.wants/grafana-agent.service → /lib/systemd/system/grafana-agent.service.

```

You can check if its running properly or not by running the following command.

```
sudo systemctl status grafana-agent
```

```

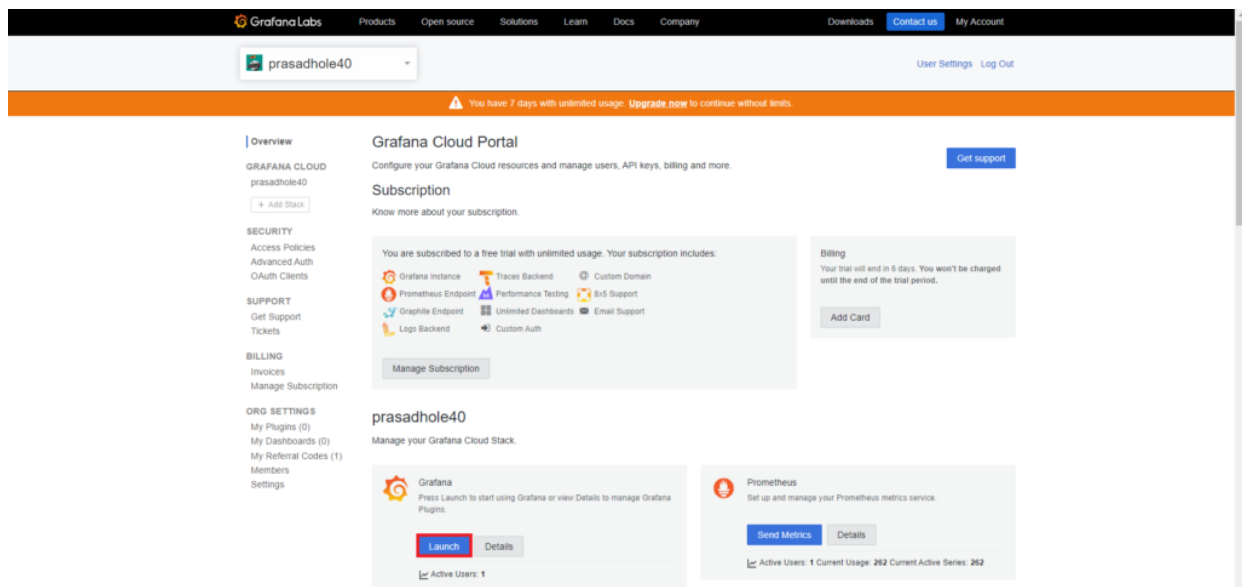
ubuntu@ip-172-31-8-191:~$ sudo systemctl status grafana-agent
● grafana-agent.service - Monitoring system and forwarder
   Loaded: loaded (/lib/systemd/system/grafana-agent.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-04-11 10:18:25 UTC; 18s ago
     Docs: https://grafana.com/docs/agent/latest/
   Main PID: 1979 (grafana-agent)
    Tasks: 6 (limit: 1121)
   Memory: 74.6M
      CPU: 229ms
   CGroup: /system.slice/grafana-agent.service
           └─1979 /usr/bin/grafana-agent --config.file /etc/grafana-agent.yaml -server.http.address=127.0.0.1:9090 -server.grpc.address=127.0.0.1:9091

Apr 11 10:18:25 ip-172-31-8-191 systemd[1]: Started Monitoring system and forwarder.

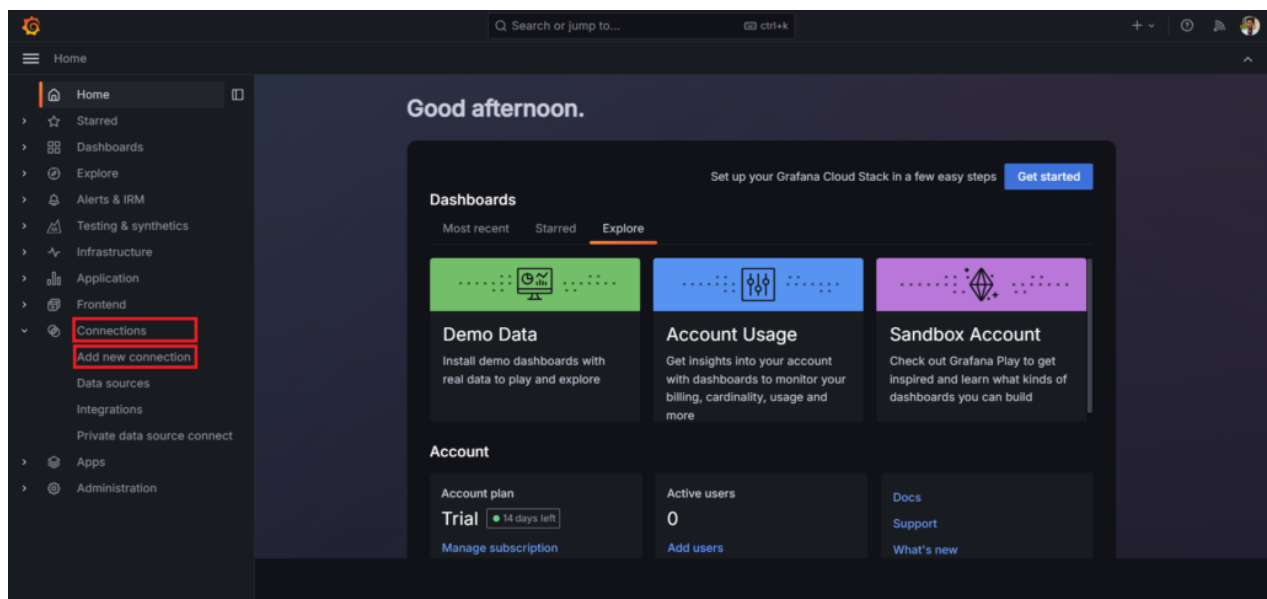
```

## Step #2: Install Linux Server Integration for Grafana Cloud

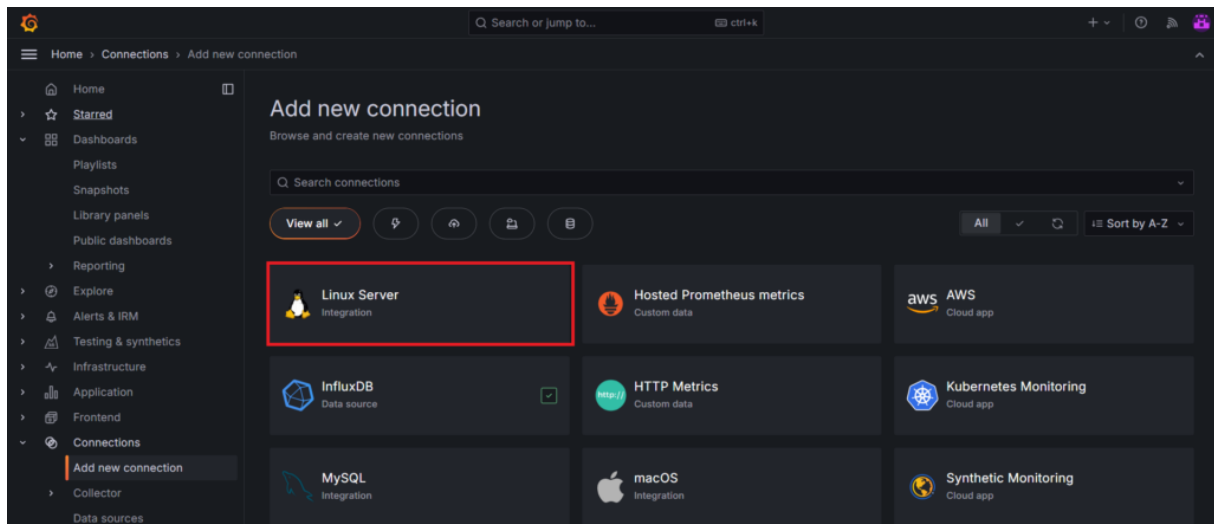
First Login to Grafana Cloud and launch the Grafana instance.



then select the Connections > Add new connections from the home bar.



Select the Linux\_Server



Click on Grafana agent and choose the operating system on which your linux server is running. Then click on Create a new API token or you can use an existing one if you have, give the Access policy token name and click on create token.

This will generate API token which we will use to set up the grafana agent.

### Agent configuration

Choose your OS

Select your OS

Debian - based

Architecture

Amd64

Looking to configure integrations for your Kubernetes environment? Check out [Kubernetes Monitoring with Grafana Cloud](#) to enable this.

Use a Grafana.com API token

Create a new API token

Use an existing API token

Access Policy token name

The name will help you view and revoke this token in grafana.com later.

prasad-40

Scopes

Predefined scopes for this token.

metrics:write

logs:write

traces:write

profiles:write

Create token

Test agent connection

Proceed to install integration

Copy the code run it to install and run grafana agent as a grafana-agent.service systemd service.

Put the API token key generated before instead of “**GCLOUD\_RW\_API\_KEY**”



Looking to configure integrations for your Kubernetes environment? Check out [Kubernetes Monitoring](#) with Grafana Cloud to enable this.

## Create a new API token

### Use an existing API token

Replace `GCLOUD_RW_API_KEY` with your own API key.

```
ARCH="amd64" GCLOUD_HOSTED_METRICS_URL="https://prometheus-prod-43-prod-ap-south-1.grafana.net/api/prom/push" GCLOUD_HOSTED_METRICS_ID="1507447" GCLOUD_SCRAPE_INTERVAL="60s" GCLOUD_HOSTED_LOGS_URL="https://logs-prod-028.grafana.net/loki/api/v1/push" GCLOUD_HOSTED_LOGS_ID="854611" GCLOUD_RW_API_KEY="GCLOUD_RW_API_KEY" /bin/sh -c "$(curl -fsSL https://storage.googleapis.com/cloud-onboarding/agent/scripts/static/install-linux.sh)"
```

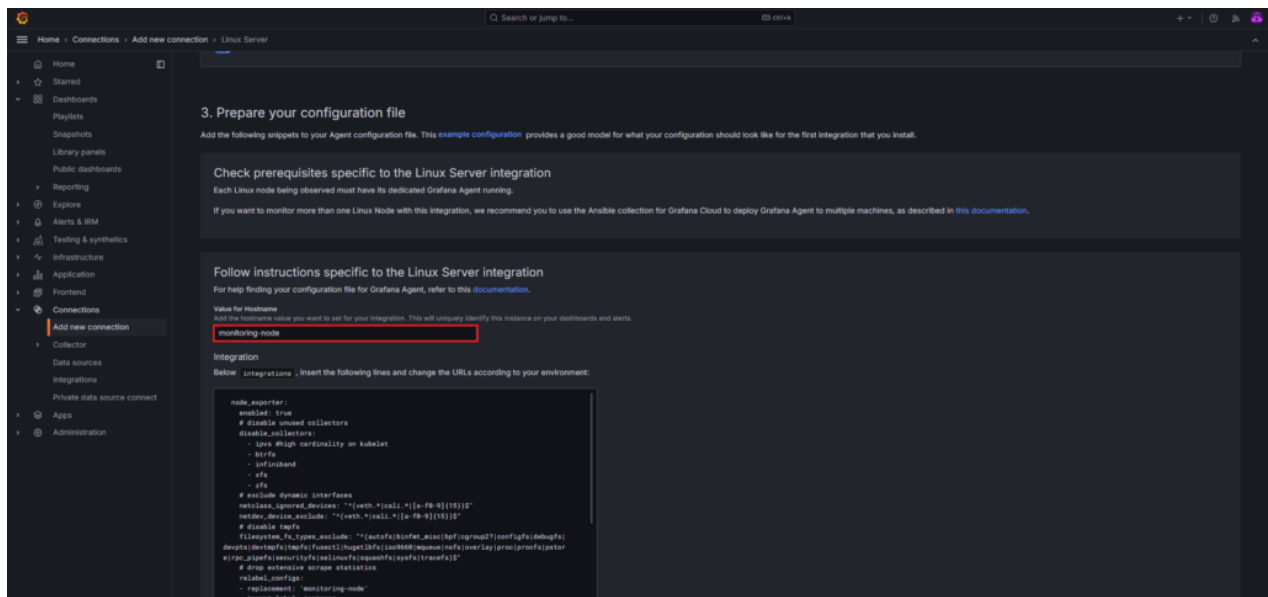
Copy to clipboard

## Test agent connection

### Proceed to install integration

## Step #3: Modify the Grafana agent yamls

Now go back to the [grafana\\_cloud](#) and scroll down.  
In prepare your configuration file give the value for hostname. Here I've given **monitoring-node**

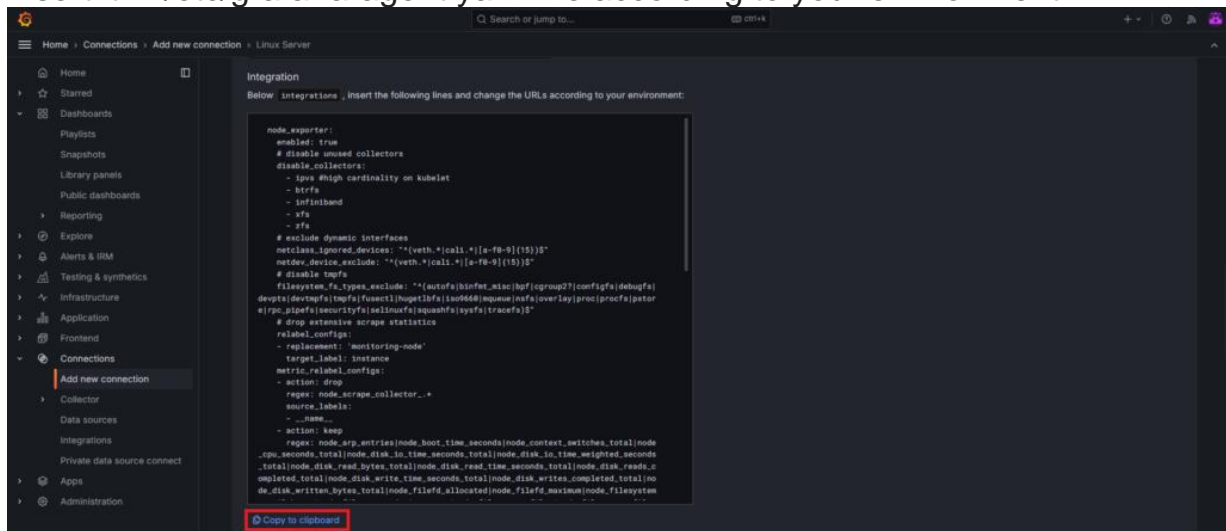


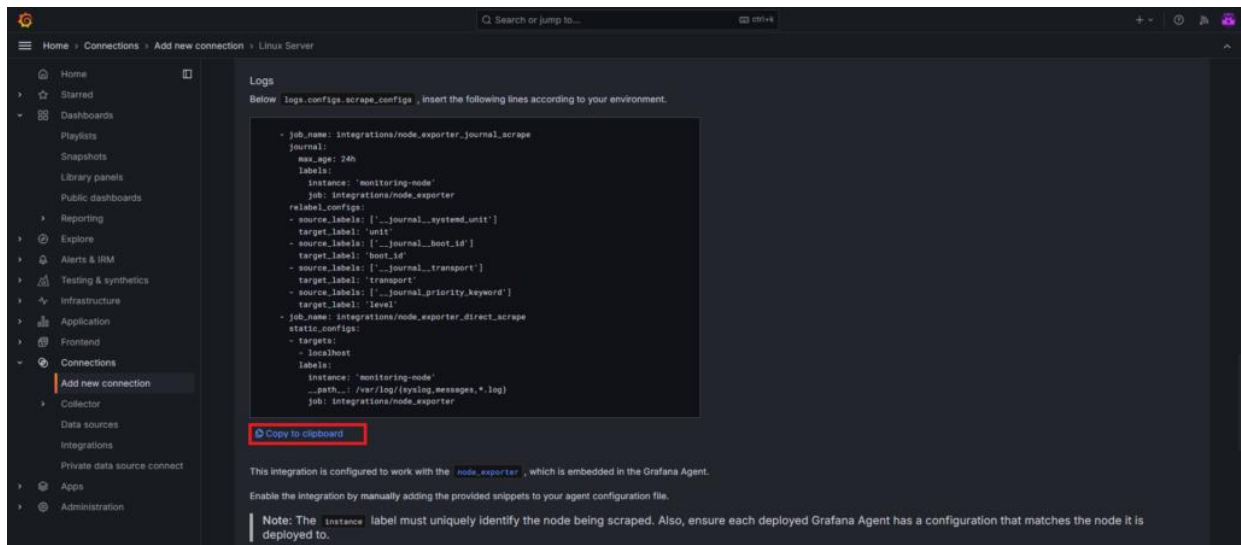
go to the `/etc/grafana-agent.yaml` by using following command.

```
sudo nano /etc/grafana-agent.yaml
```

```
ubuntu@ip-172-31-8-191:~$ sudo nano /etc/grafana-agent.yaml
```

Then copy the Integration and logs code lines from the grafana [cloud](#) and insert it in `/etc/grafana-agent.yaml` file according to your environment.





integrations:

prometheus\_remote\_write:

- basic\_auth:

password:

glc\_eyJvIjoiMTA5MTQwNCIsIm4iOiJzdGFjay04OTY4MzAtaW50ZWdyYXRpb24tcHJhc2FkLTQwIiwiaWYlI6Ik8xMjUzcjd1SDdiZTY2QTZUSmlhbzZZaSIslm0iOmsiciI6InByb2QtYXAtdGg0tMSJ9fQ==

username: 1507447

url: https://prometheus-prod-43-prod-ap-south-1.grafana.net/api/prom/push

agent:

enabled: true

relabel\_configs:

- action: replace

source\_labels:

- agent\_hostname

target\_label: instance

- action: replace



target\_label: job

replacement: "integrations/agent-check"

metric\_relabel\_configs:

- action: keep

regex:

(prometheus\_target\_sync\_length\_seconds\_sum|prometheus\_target\_scrapes\_.\*|prometheus\_target\_interval.\*|prometheus\_sd\_discovered\_targets|agent\_build.\*|agent\_wal\_samples\_appended\_total|process\_start\_time\_seconds)

source\_labels:

- \_\_name\_\_

# Add here any snippet that belongs to the `integrations` section.

# For a correct indentation, paste snippets copied from Grafana Cloud at the beginning of the **line**.

**node\_exporter:**

**enabled: true**

**# disable unused collectors**

**disable\_collectors:**

- ipvs **#high cardinality on kubelet**

- btrfs

- infiniband

- xfs

- zfs

**# exclude dynamic interfaces**

**netclass\_ignored\_devices: "^(veth.\*|cali.\*|[a-f0-9]{15})\$"**

**netdev\_device\_exclude: "^(veth.\*|cali.\*|[a-f0-9]{15})\$"**

**# disable tmpfs**

**filesystem\_fs\_types\_exclude:**  
**"^(autofs|binfmt\_misc|bpf|cgroup2?|configfs|debugfs|devpts|devtmpfs|tmpfs|fusectl|**  
**hugetlbfs|iso9660|mqueue|nsfs|overlay|proc|procfs|pstore|rpc\_pipefs|securityfs|seli**  
**nuxfs|squashfs|sysfs|tracefs)\$"**

**# drop extensive scrape statistics**

**relabel\_configs:**

**- replacement: 'monitoring-node'**

**target\_label: instance**

**metric\_relabel\_configs:**

**- action: drop**

**regex: node\_scrape\_collector\_.\***

**source\_labels:**

**- \_\_name\_\_**

**- action: keep**

**regex:**  
**node\_arp\_entries|node\_boot\_time\_seconds|node\_context\_switches\_total|node\_cpu\_se**  
**conds\_total|node\_disk\_io\_time\_seconds\_total|node\_disk\_io\_time\_weighted\_seconds\_to**  
**tal|node\_disk\_read\_bytes\_total|node\_disk\_read\_time\_seconds\_total|node\_disk\_reads\_**  
**completed\_total|node\_disk\_write\_time\_seconds\_total|node\_disk\_writes\_completed\_to**  
**tal|node\_disk\_written\_bytes\_total|node\_filefd\_allocated|node\_filefd\_maximum|node\_f**  
**ilesystem\_avail\_bytes|node\_filesystem\_device\_error|node\_filesystem\_files|node\_filesy**  
**stem\_files\_free|node\_files>**

**source\_labels:**

**- \_\_name\_\_**

**logs:**

**configs:**

- clients:

- basic\_auth:

password:

glc\_eyJvIjoiMTA5MTQwNCIsIm4iOiJzdGFjay04OTY4MzAtaW50ZWdyYXRpb24tcHJhc2FkLTQwliwiayI6Ik8xMjUzcjd1SDdiZTY2QTZUSmlhbmZSaSI0iOmsicil6InByb2QtYXAAtc291dGgtMSJ9fQ==

username: 854611

url: https://logs-prod-028.grafana.net/loki/api/v1/push

name: integrations

positions:

filename: /tmp/positions.yaml

scrape\_configs:

# Add here any snippet that belongs to the `logs.configs.scrape\_configs` section.

# For a correct indentation, paste snippets copied from Grafana Cloud at the beginning of the line.

- **job\_name: integrations/node\_exporter\_journal\_scrape**

**journal:**

**max\_age: 24h**

**labels:**

**instance: 'monitoring-node'**

**job: integrations/node\_exporter**

**relabel\_configs:**

- **source\_labels: ['\_journal\_systemd\_unit']**

**target\_label: 'unit'**

- **source\_labels: ['\_journal\_boot\_id']**

**target\_label: 'boot\_id'**

**- source\_labels: ['\_\_journal\_transport']**

**target\_label: 'transport'**

**- source\_labels: ['\_\_journal\_priority\_keyword']**

**target\_label: 'level'**

**- job\_name: integrations/node\_exporter\_direct\_scrape**

**static\_configs:**

**- targets:**

**- localhost**

**labels:**

**instance: 'monitoring-node'**

**\_\_path\_\_: /var/log/{syslog,messages,\*.log}**

**job: integrations/node\_exporter**

metrics:

configs:

**- name: integrations**

remote\_write:

**- basic\_auth:**

password:

glc\_eyJvIjoiMTA5MTQwNCIsIm4iOiJzdGFjay04OTY4MzAtaW50ZWdyYXRpb24tcHJhc2FkLTQwliwiayI6lk8xMjUzcjd1SDdiZTY2QTZUSmlhbzZZaSIslm0iOnsicil6InByb2QtYXAAtc291dGgtMSJ9fQ==

username: 1507447

```
url: https://prometheus-prod-43-prod-ap-south-1.grafana.net/api/prom/push
```

```
scrape_configs:
```

```
# Add here any snippet that belongs to the `metrics.configs.scrape_configs` section.
```

```
# For a correct indentation, paste snippets copied from Grafana Cloud at the beginning of the line.
```

```
global:
```

```
scrape_interval: 60s
```

```
wal_directory: /tmp/grafana-agent-wal
```

save the file and restart the [\\_grafana](#) agent service

```
systemctl restart grafana-agent.service
```

Give the password to authenticate.

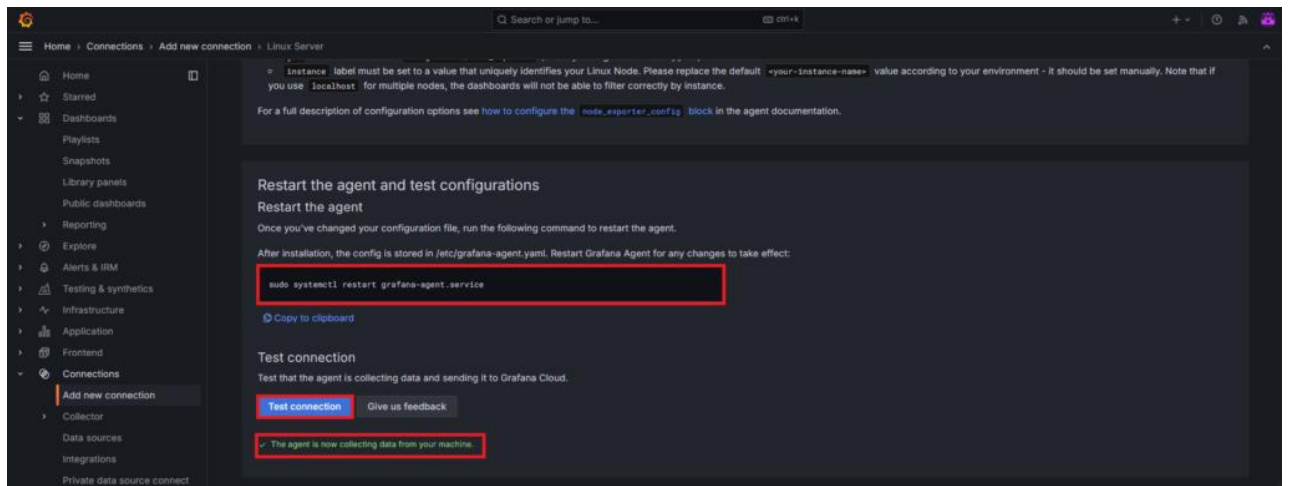
```
ubuntu@ip-172-31-8-191:~$ systemctl restart grafana-agent.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to restart 'grafana-agent.service'.
Authenticating as: Ubuntu (ubuntu)
Password:
==== AUTHENTICATION COMPLETE ====
```

If you don't know the password you can change it using following command.

```
sudo passwd ubuntu
```

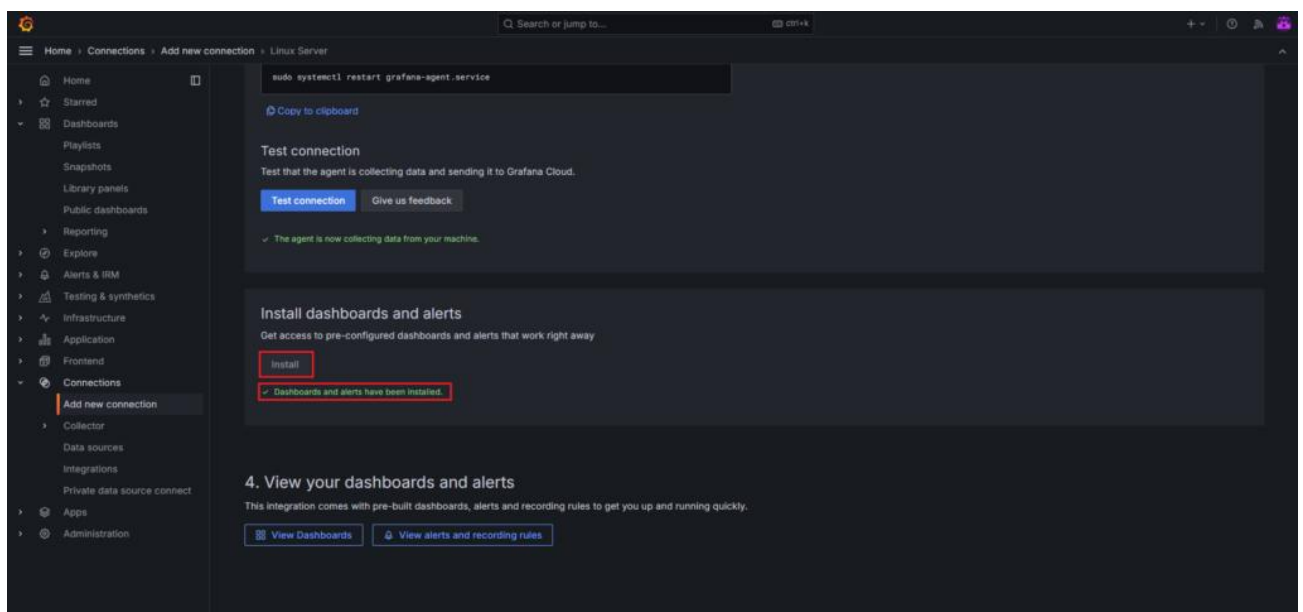
scroll down and click on **Test connections** to test that the agent is collecting data and sending it to [\\_Grafana Cloud](#).

You will get the message **"The agent is now collecting data from your machine."**

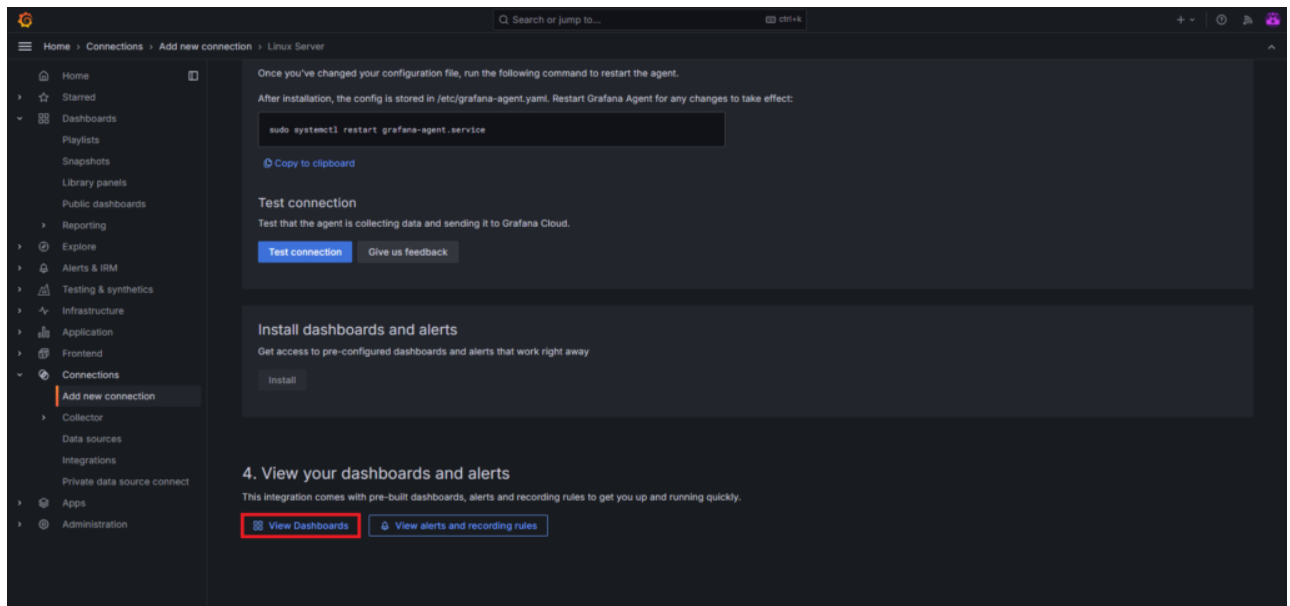


## Step #4: Install Dashboards and alerts on Grafana Cloud

Next click on **Install** to install the pre-configured dashboards and alerts.

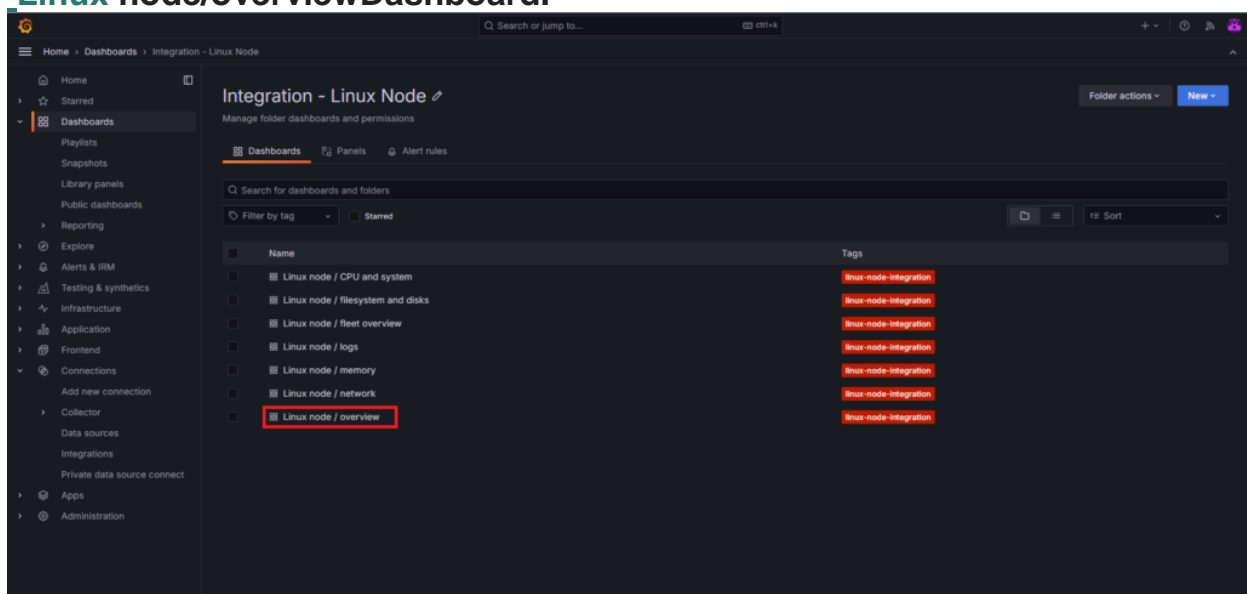


Now click on **View Dashboards**.



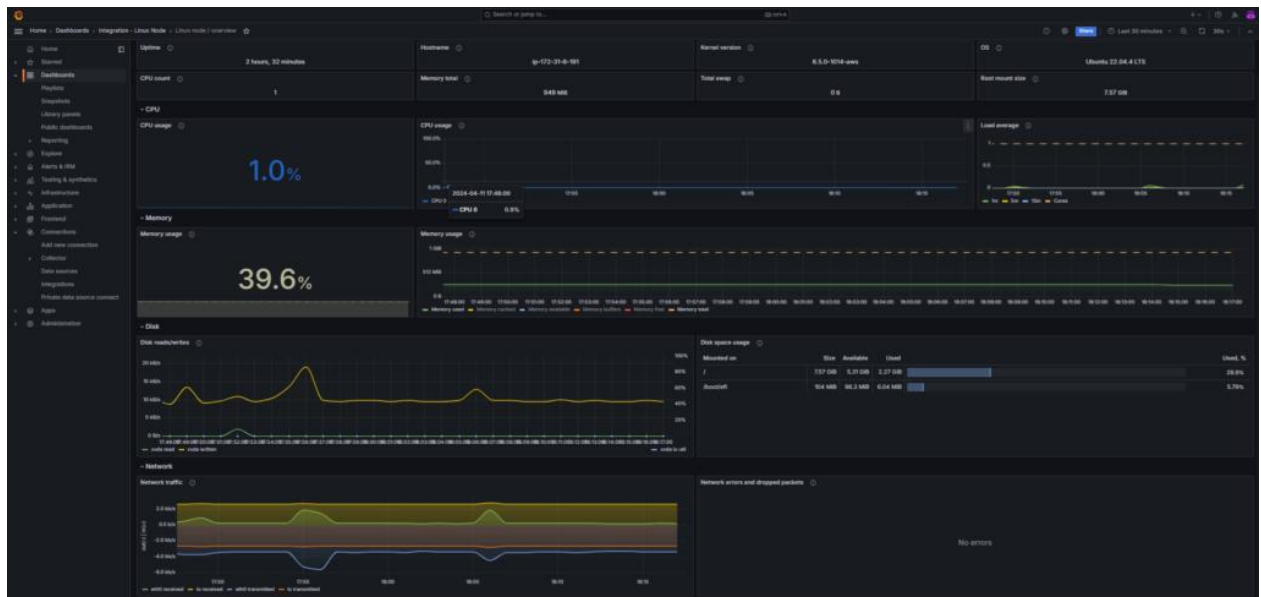
You will see the many pre-build dashboards installed. For now we will select

## Linux node/overviewDashboard.



The dashboard displays various graphs and metrics which are related to server performance.

Like CPU usage, memory usage, network traffic and many more.



## Conclusion:

Integrating [Linux](#) servers with Grafana Cloud provides valuable insights into the performance and health of your infrastructure. You can seamlessly integrate Linux [server](#) with [Grafana](#) Cloud. By following the steps outlined in this guide, you can easily set up monitoring and visualization for your Linux servers and gain visibility into their operation. Whether you're managing a single server or a large-scale deployment, Grafana Cloud offers the tools you need to monitor and optimize your infrastructure effectively.

Project by :KHUSHI SHARMA  
2<sup>ND</sup> SEMESTER