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/
```

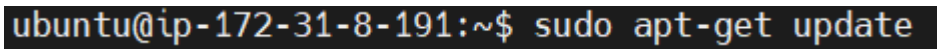

```
wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee
/etc/apt/keyrings/grafana.gpg > /dev/null
```

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
```

Update the repositories

```
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No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.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.
```

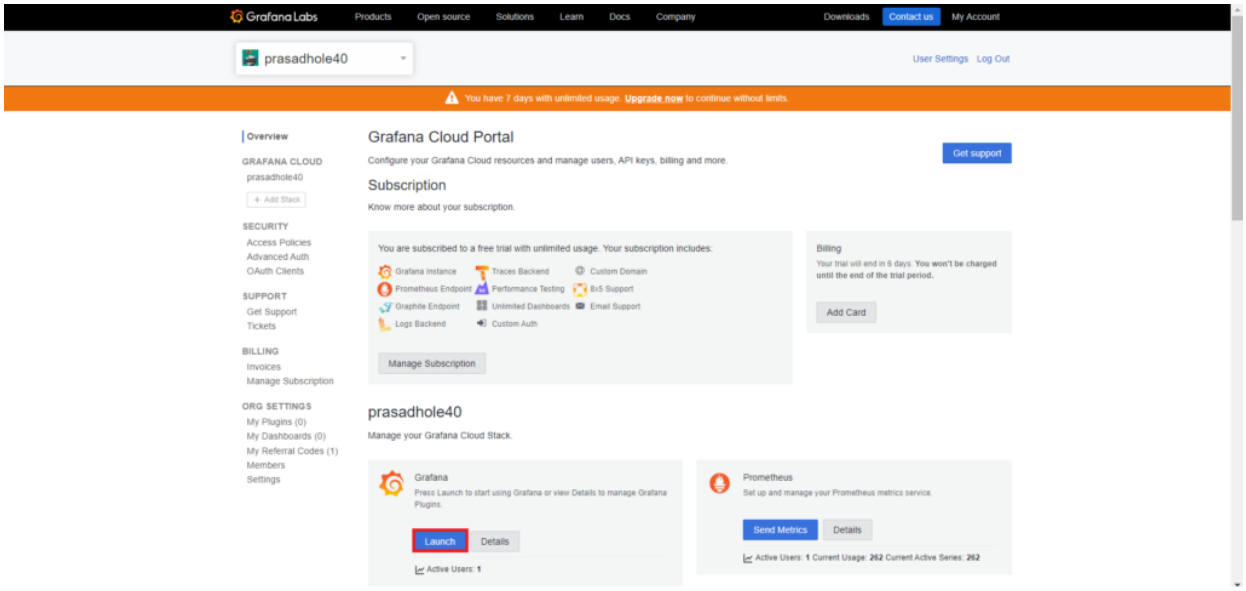
You can check if its running properly or not by running the following command.

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

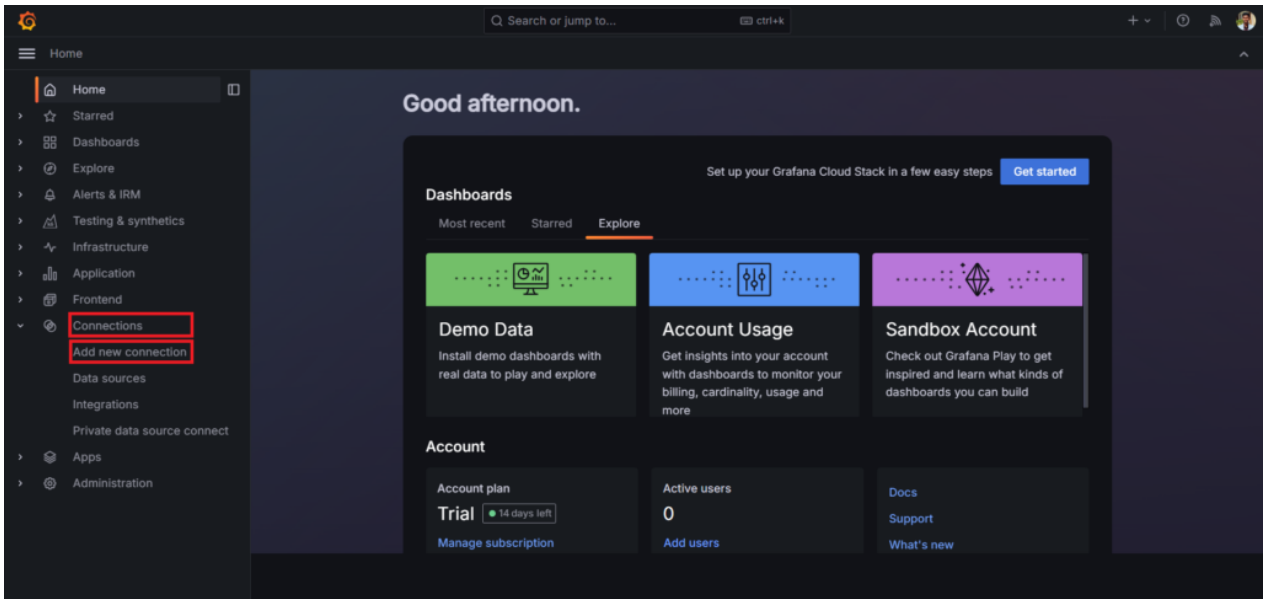
```
ubuntu@ip-172-31-8-191:~$ sudo systemctl status grafana-agent
● grafana-agent.service - Monitoring system and forwarder
     Loaded: loaded (/lib/systemd/system/grafana-agent.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2024-04-11 10:18:25 UTC; 18s ago
       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

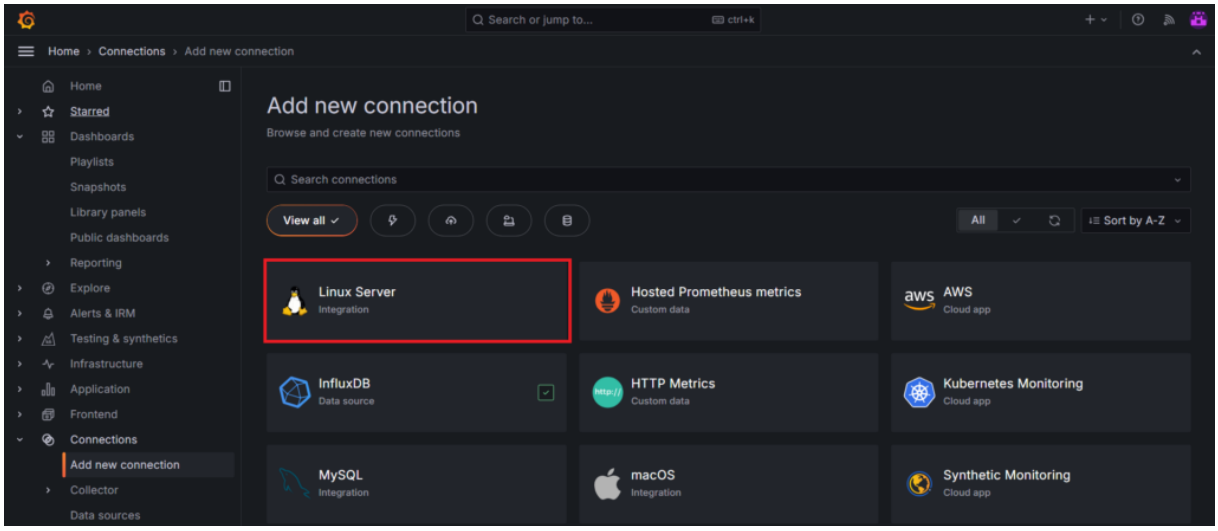
First Login to Grafana Cloud and launch the Grafana instance.



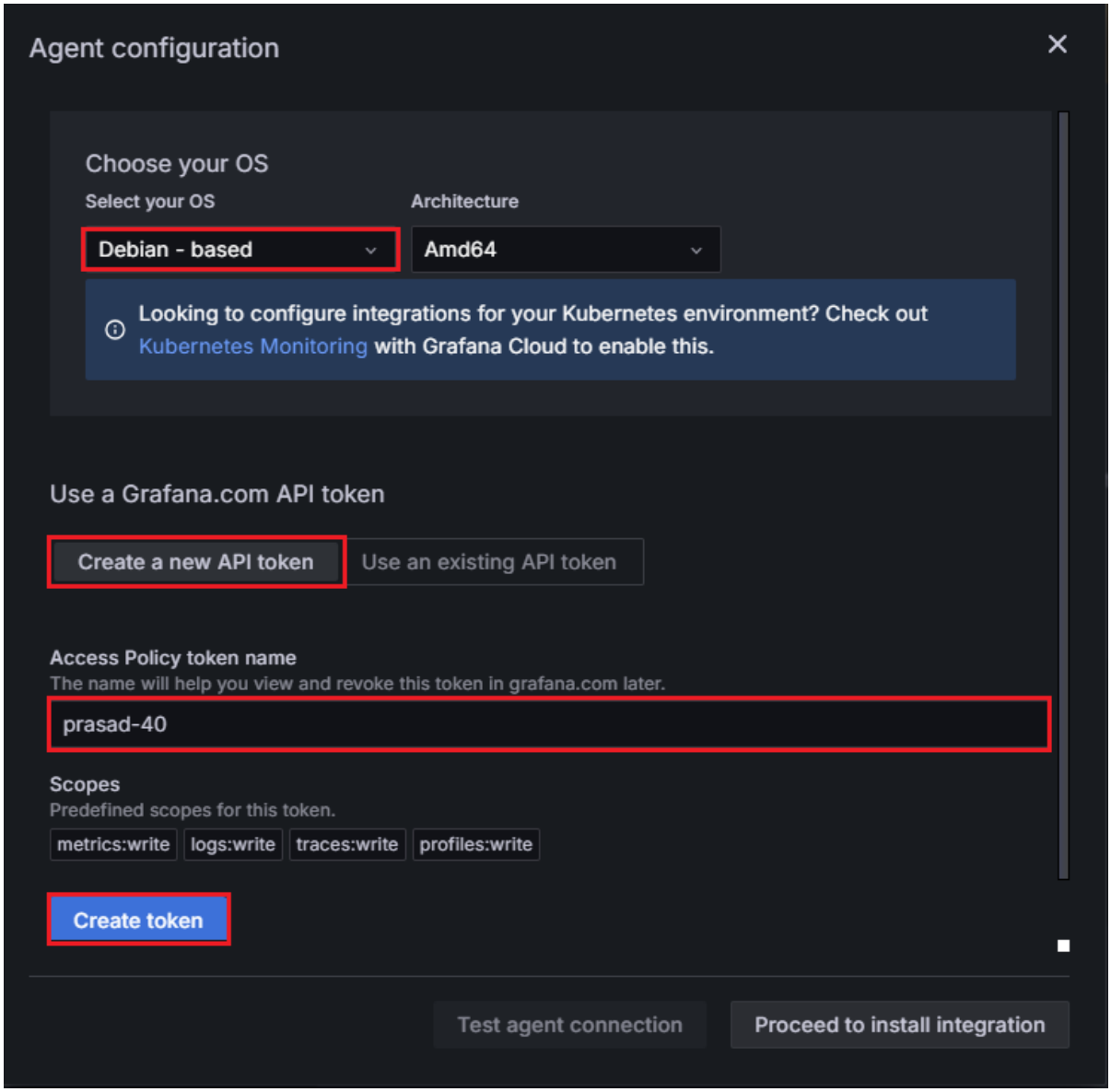
then select the Connections > Add new connections from the home bar.



Select the _Linux _Server

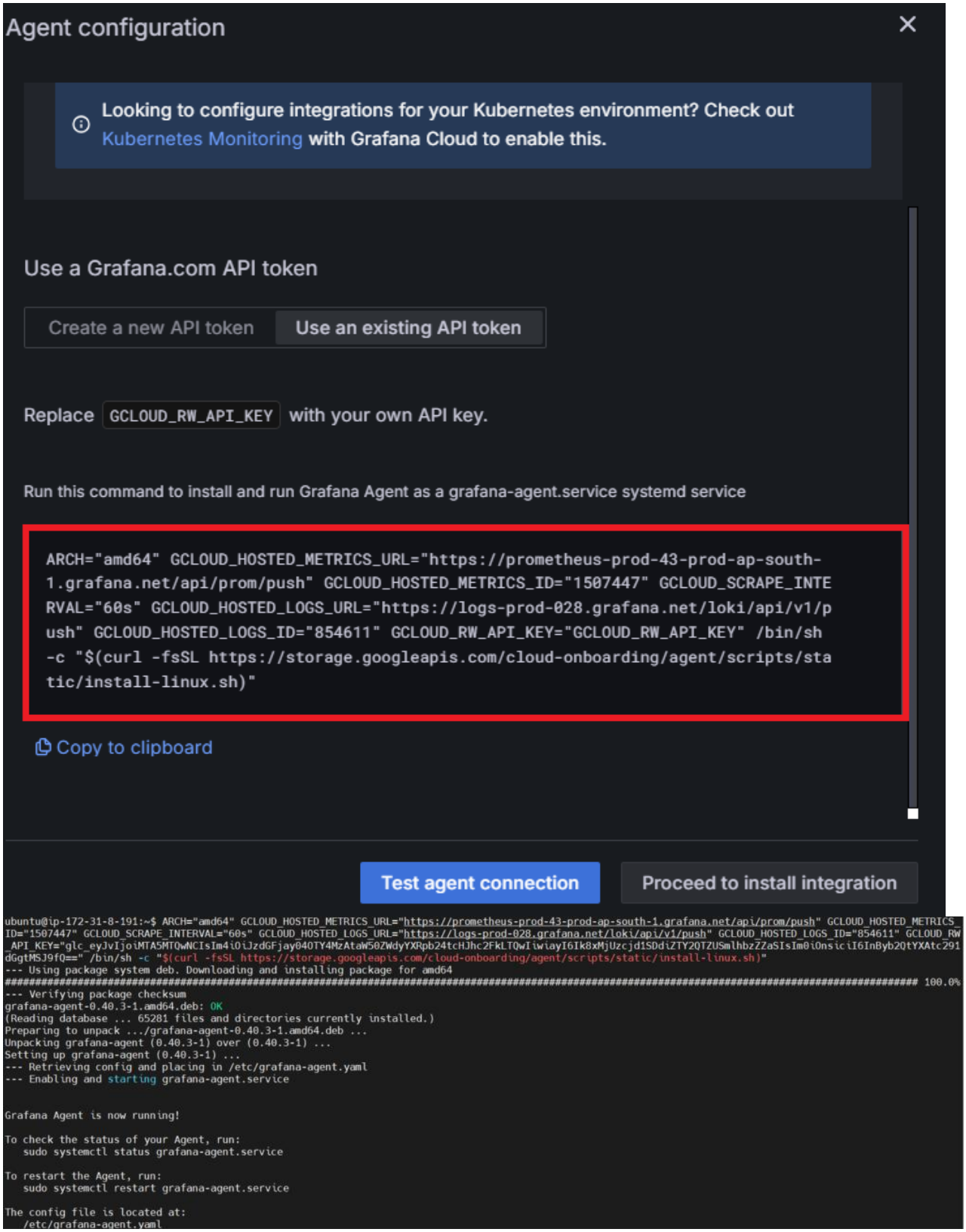


Click on _Grafana agent and choose the operating system on which your _linux server is running. Then click on Create a new API token or you can use an existing on if you have, give the Access policy token name and click on create token.
This will generate API token which we will use to set up the _grafana agent.



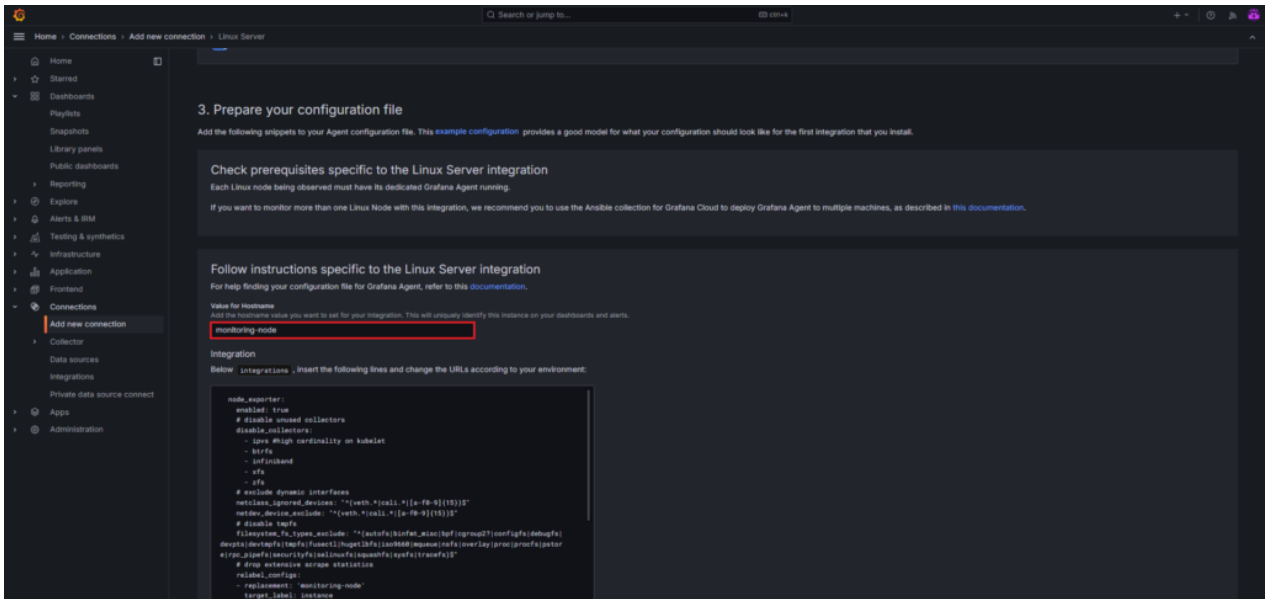
Copy the code run it to install and run grafana agent as a grafana-agent.service systemd service.

Put the API token key generated before instead of “**GCLOUD_RW_API_KEY**“



Step #3:Modify the Grafana agent yaml file

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**

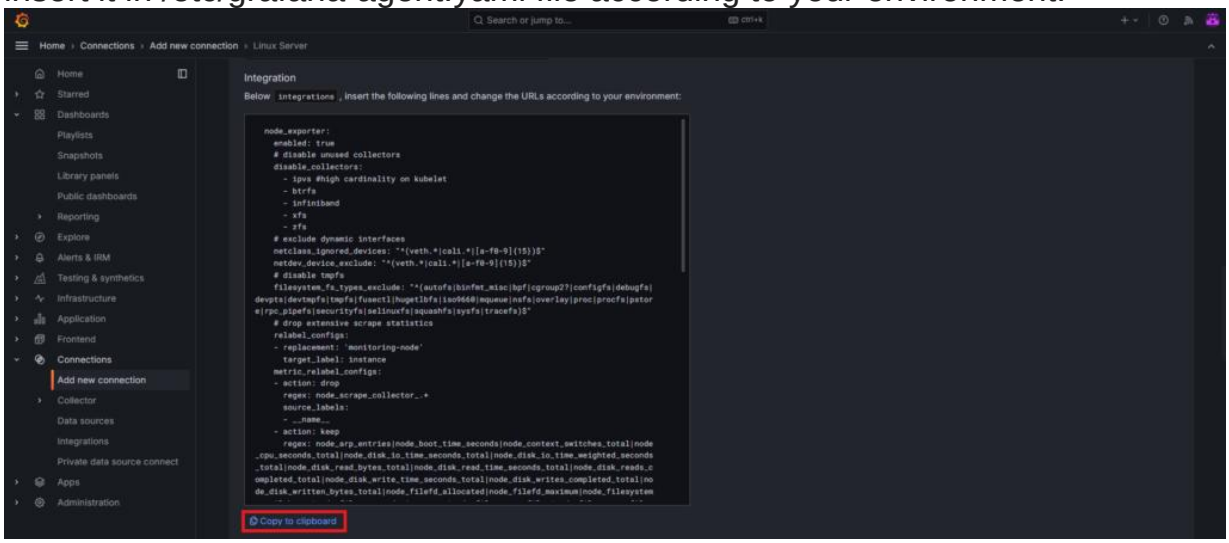


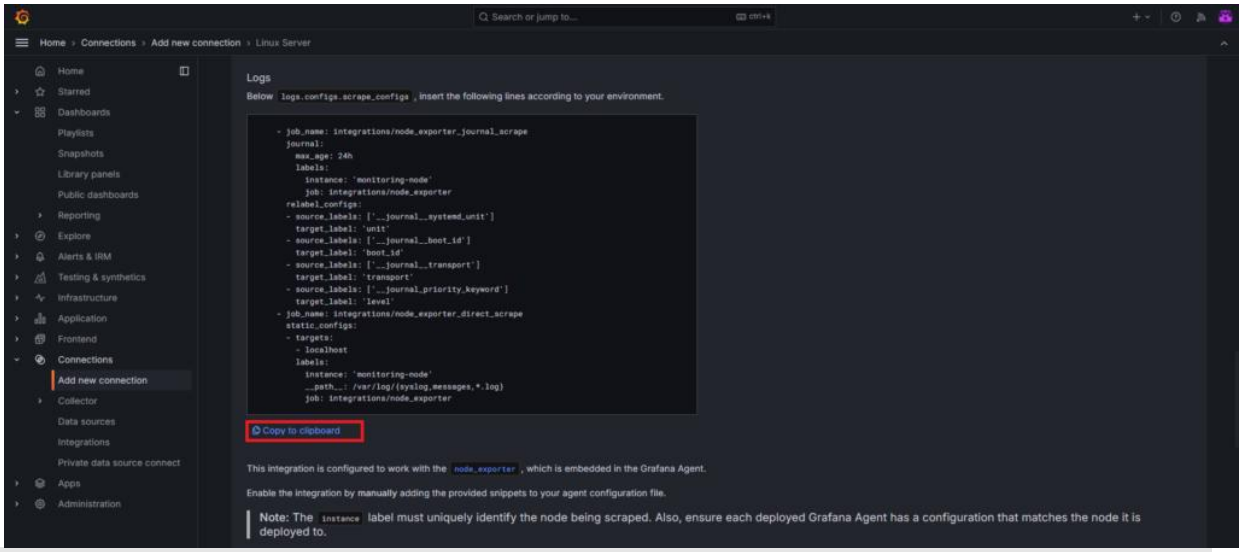
go to the /etc/grafana-agent.yaml by using following command.

```
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_eyJvIjoiMTA5MTQwNCIsIm4iOiJzdGFjay04OTY4MzAtaW50ZWdyYXRpb24tcHJhc2FkLTQwIiwiayI6Ik8xMjUzcjd1SDdiZTY2QTZUSmlhbzZZaSIsIm0iOnsiciI6InByb2QtYXAtc291dGgtMSJ9fQ==

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_eyJvIjoiMTA5MTQwNCIsIm4iOiJzdGFjay04OTY4MzAtaW50ZWdyYXRpb24tcHJhc2FkLTQwIiwiayI6Ik8xMjUzcjd1SDdiZTY2QTZUSmlhbzZZaSIsIm0iOnsiciI6InByb2QtYXAtc291dGgtMSJ9fQ==

        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']
```

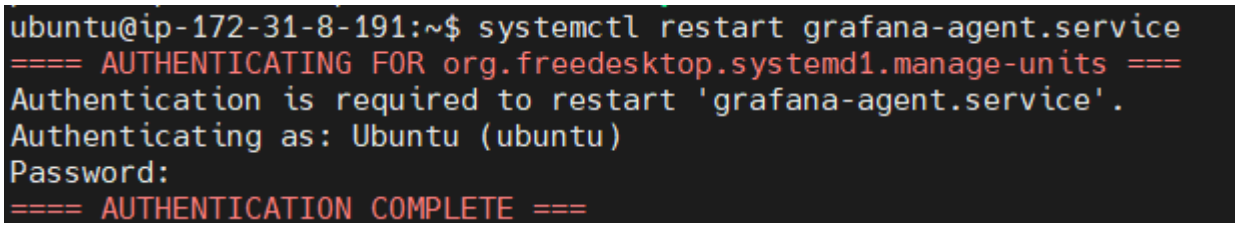
```
      target_label: 'boot_id'
    - source_labels: ['__journal__transport']
      target_label: 'transport'
    - source_labels: ['__journal_priority_keyword']
      target_label: 'level'
  - job_name: integrations/node_exporter_direct_scrape
    static_configs:
    - targets:
      - localhost
      labels:
        instance: 'monitoring-node'
        __path__: /var/log/{syslog,messages,*.log}
        job: integrations/node_exporter


metrics:
  configs:
  - name: integrations
    remote_write:
    - basic_auth:
        password:
glc_eyJvIjoiMTA5MTQwNCIsIm4iOiJzdGFjay04OTY4MzAtaW50ZWdyYXRpb24tcHJhc2FkLTQwIiwiayI6Ik8xMjUzcjd1SDdiZTY2QTZUSmlhbzZZaSIsIm0iOnsiciI6InByb2QtYXAtc291dGgtMSJ9fQ==
        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
```

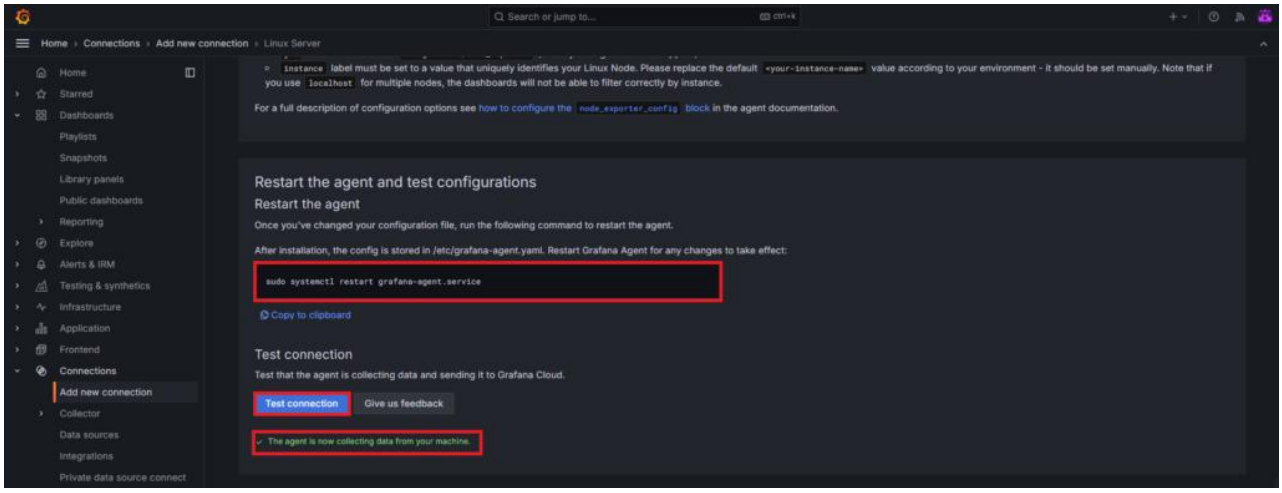
Give the password to authenticate.



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

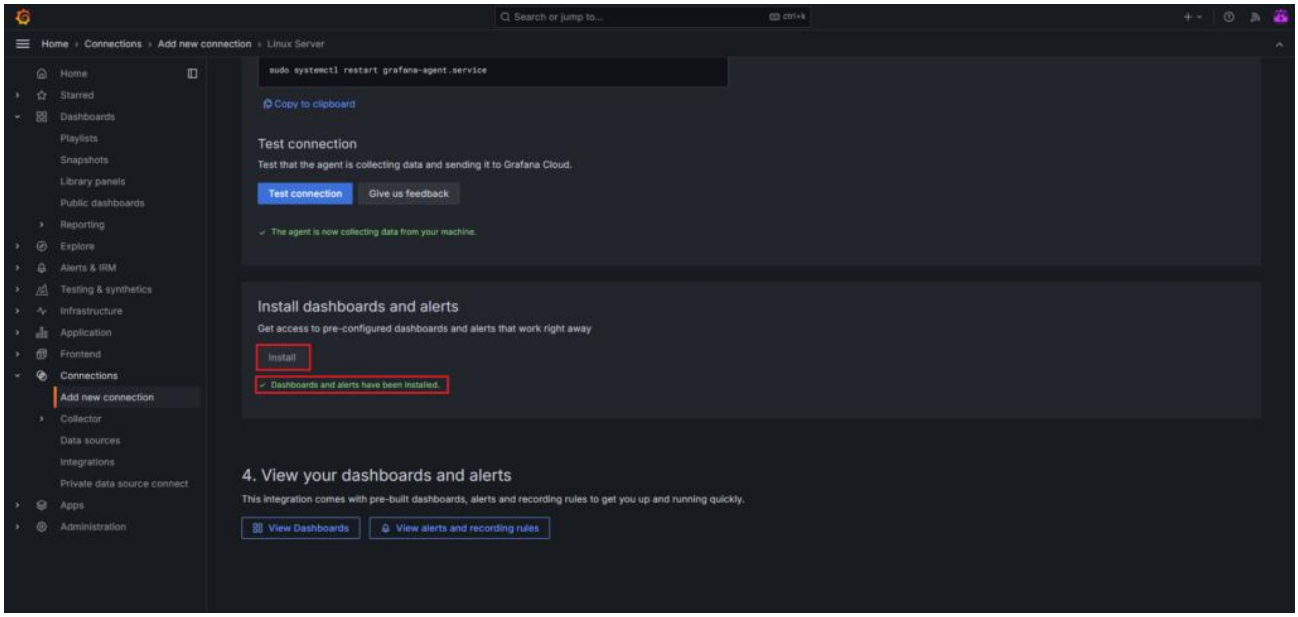
```
sudo passwd ubuntu
```

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.**"

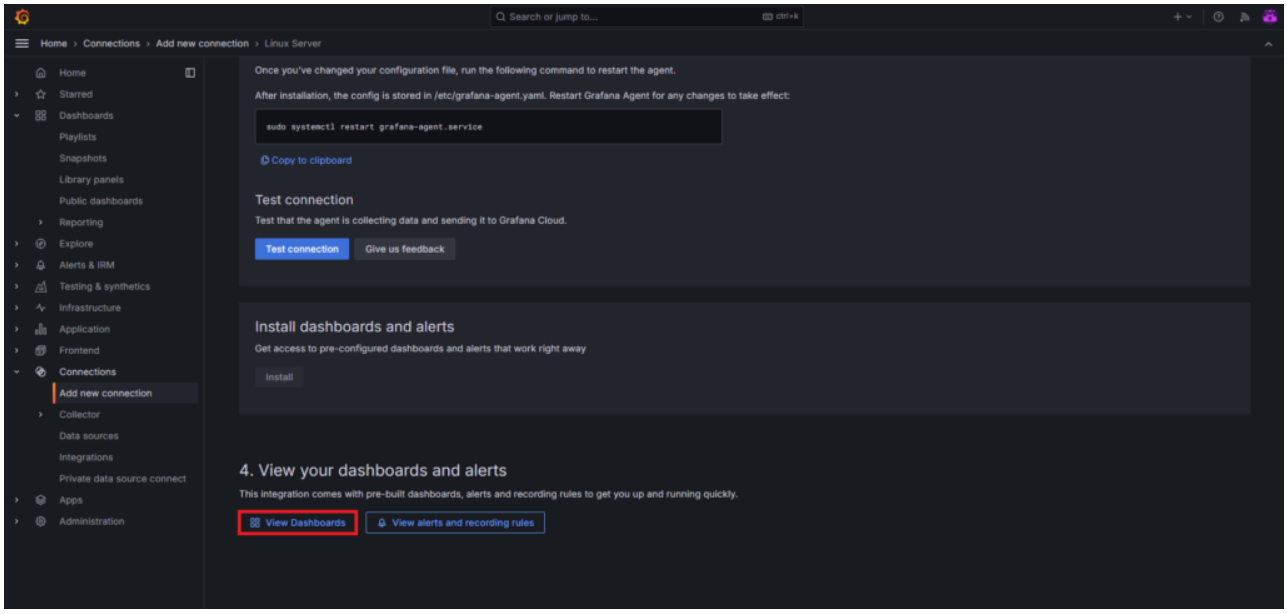


Step #4:Install Dashboards and alerts on Grafana Cloud

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

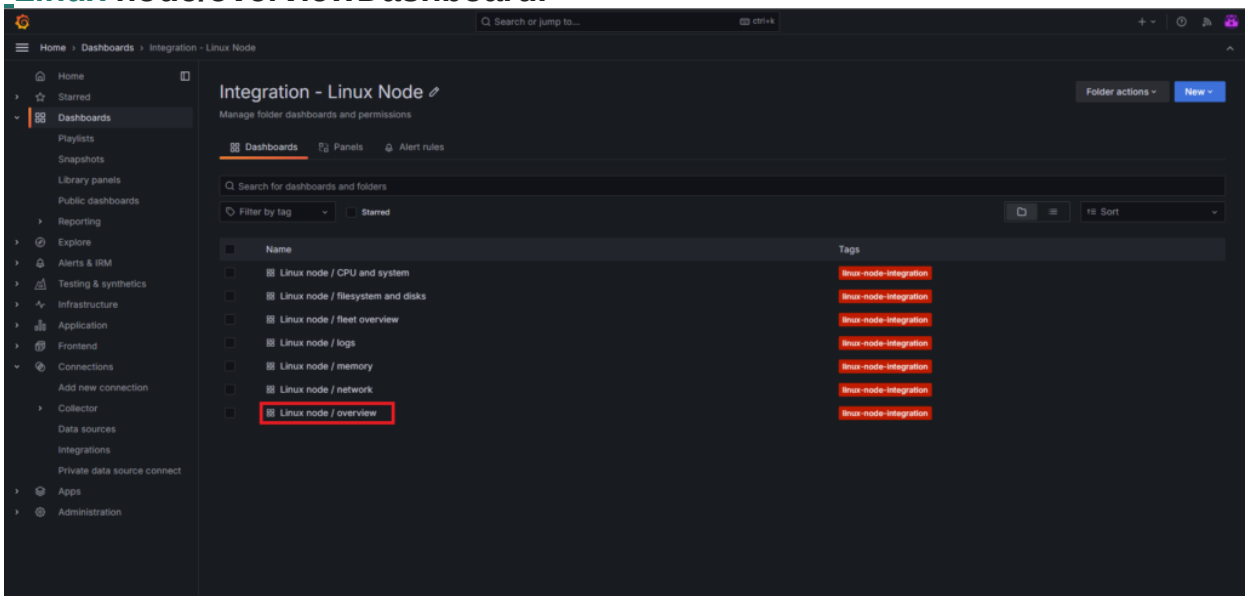


Now click on **View Dashboards**.

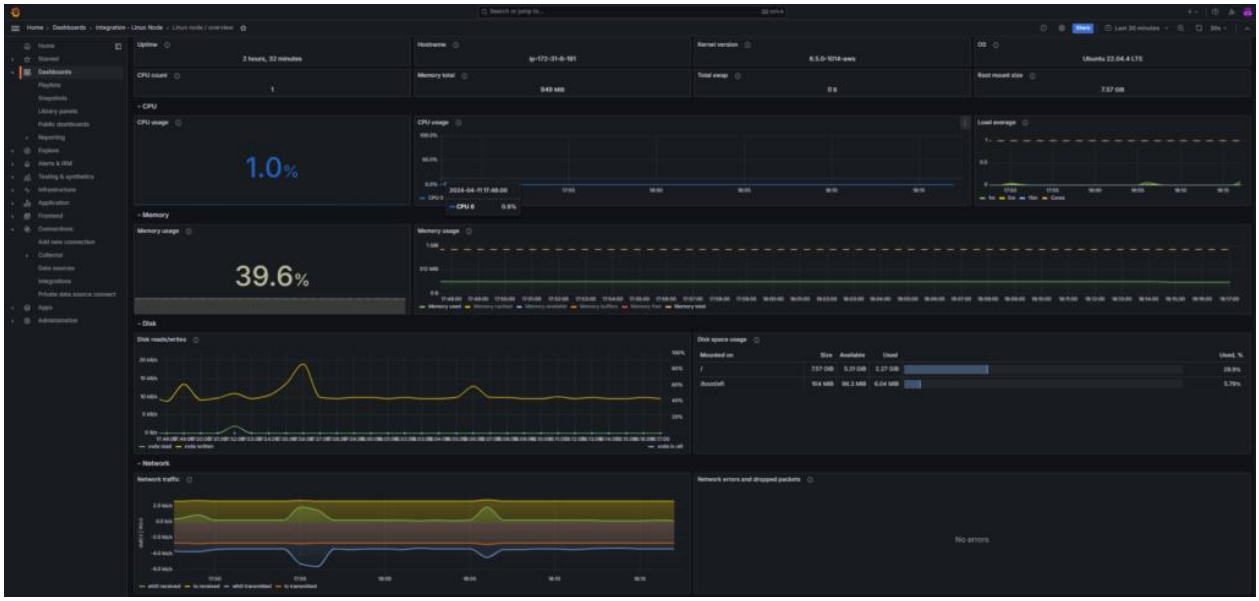


You will see the many pre-build dashboards installed. For now we will select

Linux node/overviewDashboard.



The dashboard displays various graphs and metrics which are related to _server performance.
Like CPU usage, memory usage, network traffice 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.