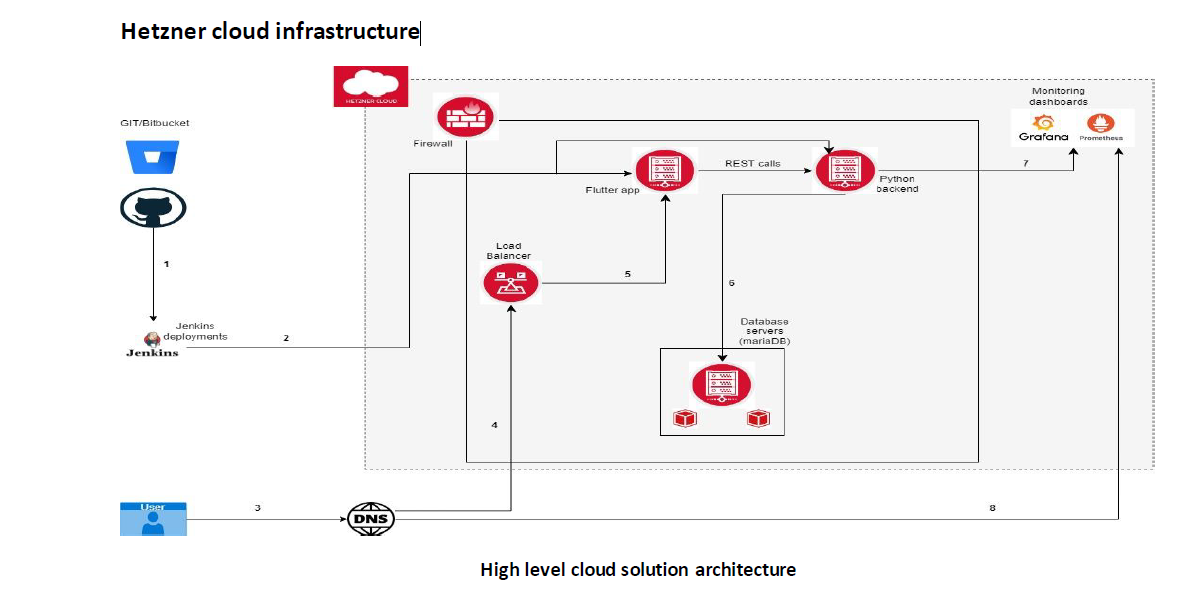
**Hetzner cloud infrastructure Step**



**Created resources:--**

1. Frontend-server --- Contain flutter app
2. Backend-server --- Contain python api
3. Database server --- Contain mariadb database
4. Monitoring Server --- Contain Prometheus, Grafana, node exporter, alert manager
5. Load Balancer ---target server “Frontend-server, port: http:80 -> 80
6. Volume ----- It is attached with Database Server – 20 Gb
7. Firewall name – **monitoring** --- allow 9093, 3000, ICMP, 9100, 22
8. Firewall name -- **frontend-server** --- allow 9093, ICMP, 9100, 80, 22,
9. Firewall name – **database** --- allow 9093, ICMP, 9100, 3306
10. Firewall name -- **backend-server** ---- allow 9093, ICMP, 9100, 22,80
11. Network resource subnet “hc\_private” --- 10.0.0.0/24

Assign static private ip address for monitor server with **Prometheus**, Grafana.

1. **Network ip assign for each server** **– for database server fixed** ip 10.0.0.2
2. **Network ip assign for each server** **– for monitoring server fixed** ip 10.0.0.3
3. **Network ip assign for each server** **– for backend server fixed** ip 10.0.0.4
4. **Network ip assign for each server** **– for frontend-server server fixed** ip 10.0.0.5
5. Create ssh key to access all server. Added public key tf\_hetzner.pub.pub --- >>> public key

All application install in serves with **cloud-config.**

**Create ssh key :---**

**Use this command :**

ssh-keygen -f ~/.ssh/ tf\_hetzner -t rsa -b 4096 -N ''

private\_key\_rsa tf\_hetzner.pub tf\_hetzner.pub.pub

tf\_hetzner.pub.pub --- >>> public key -🡪 This key added in the file **ssh.tf** file.

tf\_hetzner.pub 🡪 private key (another same file private\_key\_rsa)

And also added in all .yml file (**it will different, copy from .pub.pub file**)

    ssh\_authorized\_keys:

      - ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIBhwpoRosPVZRWo0LBJTvn5Gdu0EfaJpiblHIvsCdWdQ

Create devops users in all server with .yml file.

**To connect:**

SSH agent forwarding in ubuntu:--

eval "$(ssh-agent -s)"

ssh-add ~/.ssh/your\_private\_key

ssh -A devops@remote\_server\_ip(public ip)

#sudo nano /etc/ssh/sshd\_config

AllowAgentForwarding yes

sudo service sshd restart

sudo systemctl restart sshd

**Cloud Api Token :-- it is require to run apply terraform plan, It will different for each hetzner cloud account. Create account in hetzner could account and create this api first.**

3PlrvUxlgsgrgSBuoffds4hAJt4Cpilxz2h4ddj2aD4izwBnCO2BP73JS9y

Code uploaded in the below link:--

[git.rishabhsoft.com](https://git.rishabhsoft.com/)

[Cloud-Practice / Workspaces / docker\_for\_prometheus · GitLab (rishabhsoft.com)](https://git.rishabhsoft.com/cloud-practice/workspaces/docker_for_prometheus)

<https://git.rishabhsoft.com/cloud-practice/workspaces/docker_for_prometheus.git>

[Cloud-Practice / Workspaces / anupam · GitLab (rishabhsoft.com)](https://git.rishabhsoft.com/cloud-practice/workspaces/anupam)

Grafana password:--

GF\_SECURITY\_ADMIN\_USER=admin

GF\_SECURITY\_ADMIN\_PASSWORD=foobar

GF\_USERS\_ALLOW\_SIGN\_UP=false

Access promethous with <http://public> ip of monitering server:9090

Access Grafana with : <http://public> ip of monitering server:3000

Email alert get form alert manager and Grafana. Form alert manager get only email alert, but Grafana alert get email alert and graphical representation also.

Grafana configure and also configure alert for getting email alert: --

##### Add data source :--- Click on “Add your first data source” 🡪 select “promethous”

##### 

##### As it is Prometheus, Grafana configure in docker, so ip address or localhost will not work. Need to put as below picture. You can check below link.

##### Go solution form this link:-- <https://github.com/grafana/grafana/issues/46434>

##### <http://prometheus:9090>

##### 

##### 

add 1st dashboard:--

-----------------------------

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

import dash board and select datasource “prometheus-1”

##### select 🡪 Node Exporter Full -🡪 ID: 1860

##### Dashboard:-- create folder “traning”

##### 

##### Email alert configure in Grafana: ---

##### App password create in gmail ([singhakalyani68@gmail.com](mailto:singhakalyani68@gmail.com))

##### 

##### Need to enable 2-steps verification:---

##### Then go to password option 🡪 create app password -> create app password from gmail “usyd lizy eqxq xotw”

##### Grafana/grafana.ini contain :-- You can change email id in docker\_compose location.

##### <https://github.com/anupam02us/anupam02us-prometheusdocker/tree/main/prometheus-grafana/grafana>

##### [smtp]

##### enabled = true

##### host = smtp.gmail.com:587

##### user = singhakalyani68@gmail.com

##### # If the password contains # or ; you have to wrap it with triple quotes. Ex """#password;"""

##### password = usyd lizy eqxq xotw

##### ;cert\_file =

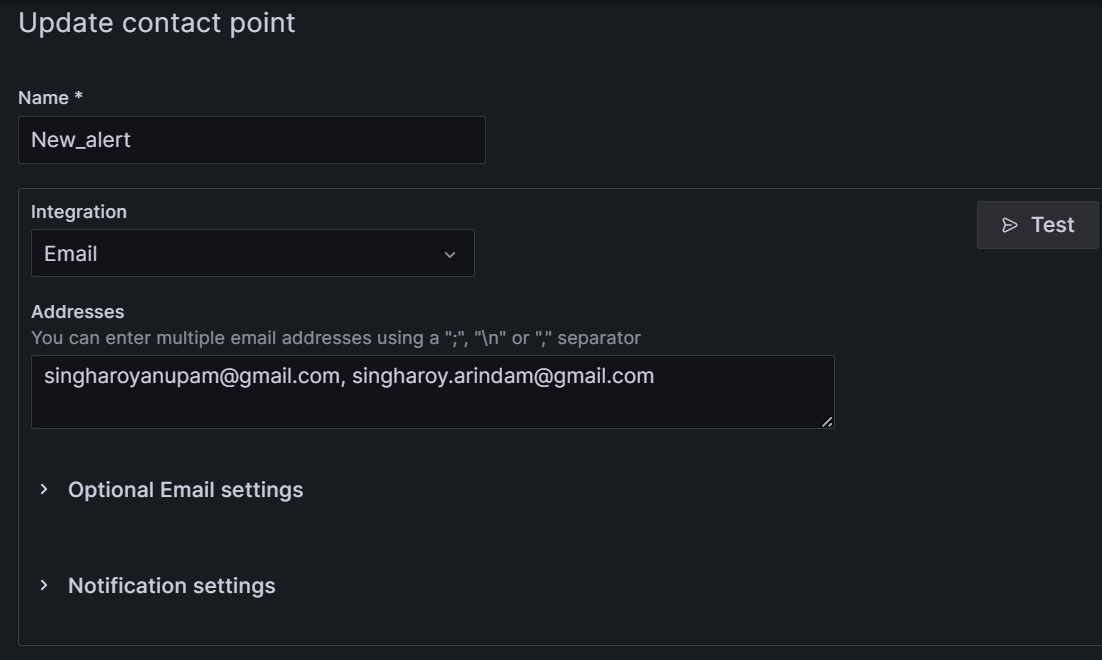
##### ;key\_file =

##### skip\_verify = true

##### from\_address = singhakalyani68@gmail.com

##### from\_name = Grafana

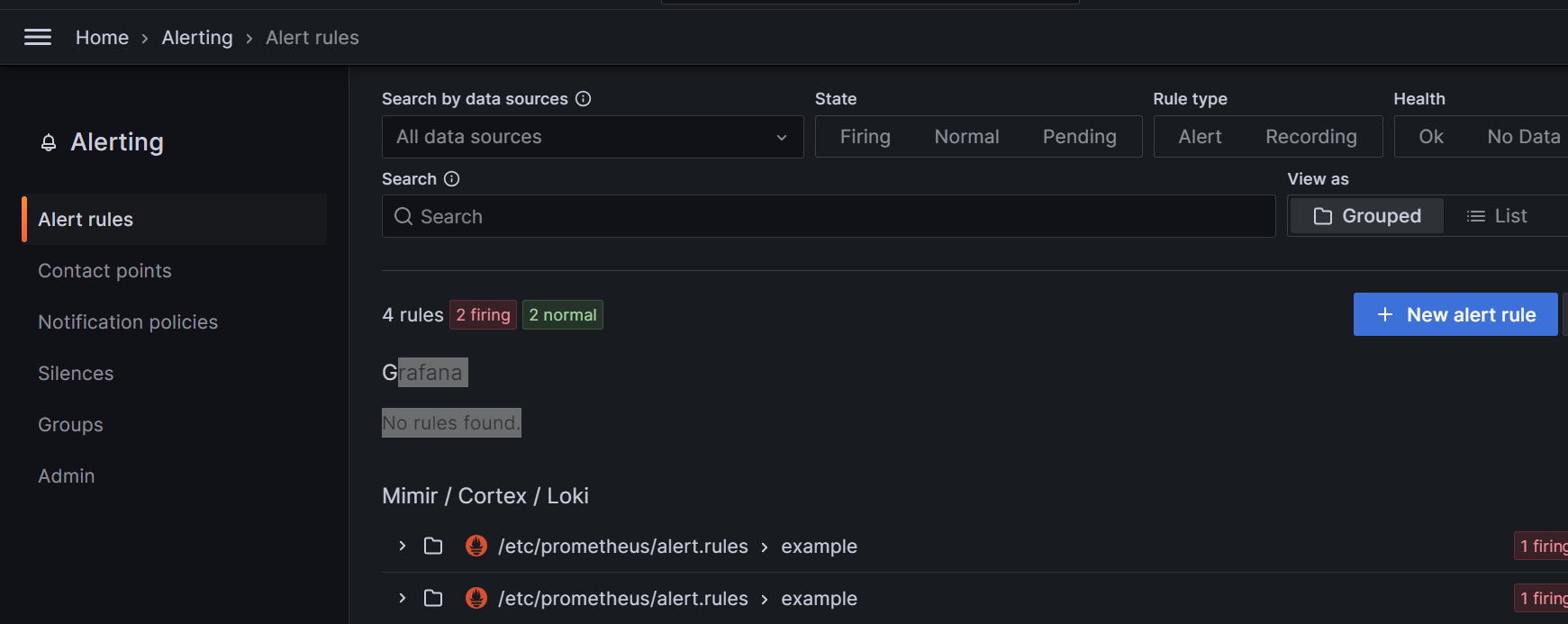
##### Go to Grafana console home 🡪 contact point 🡪



Click on test 🡪 you will get test mail in your mail id.

**Crete alert rules in Grafana:--**

Click on home -> Alerting -> Alert rules ->



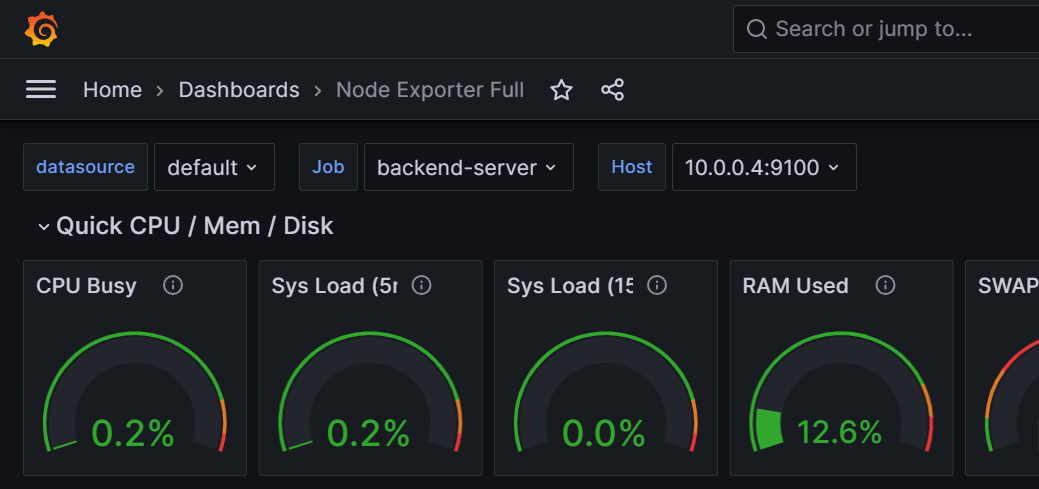
Under Grafana “No rules found”

Click on “new alert rule” -> give name : test-grafana-alert ->

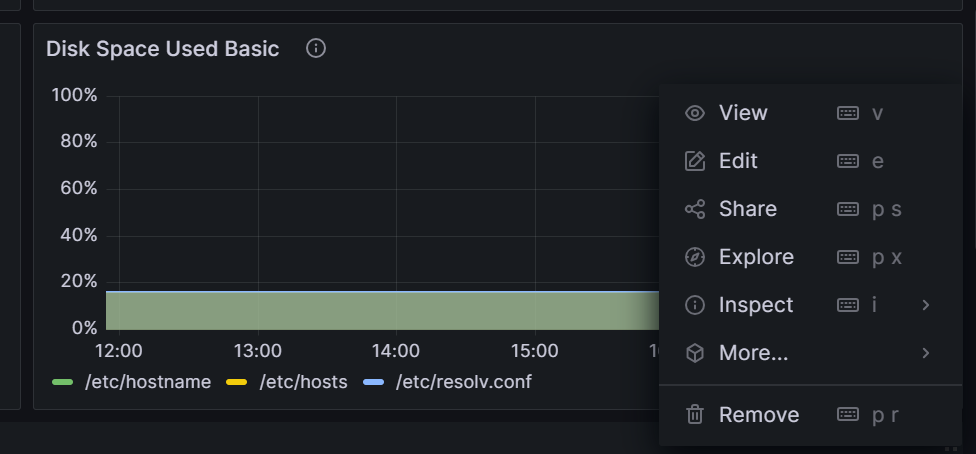
Configure Disk usage alert:---

**Configure “Disk Alert”:--**

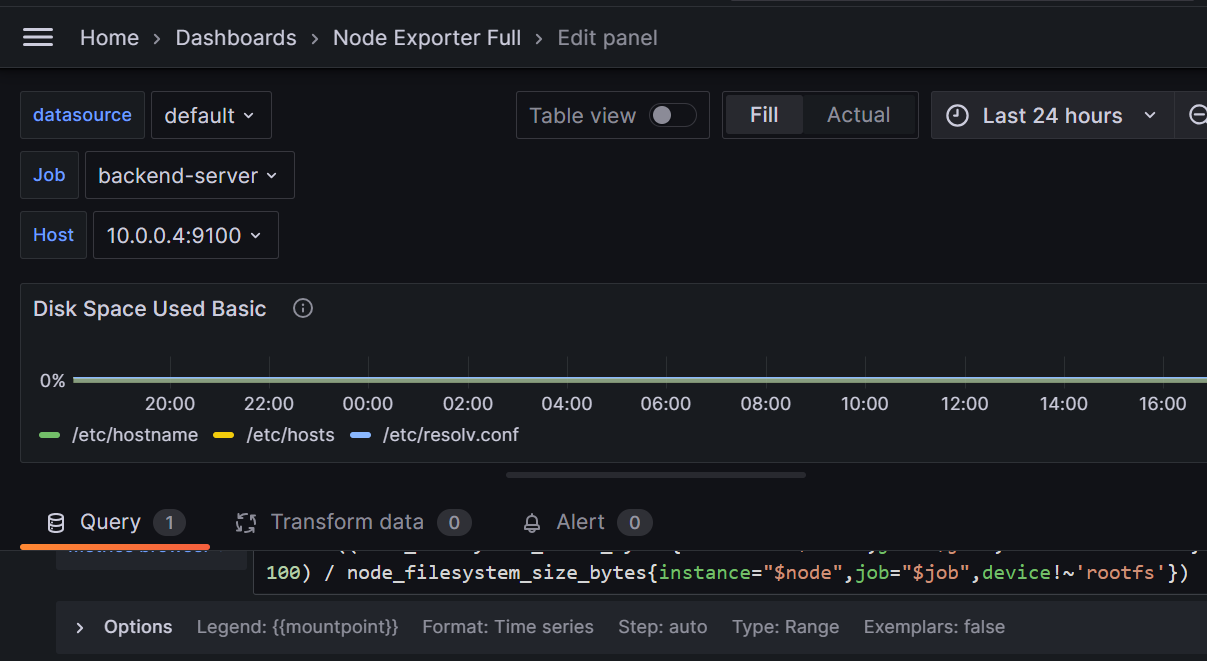
Got to “Node Exporter full” dashboard 🡪 select any server (here selected backend-server) 🡪

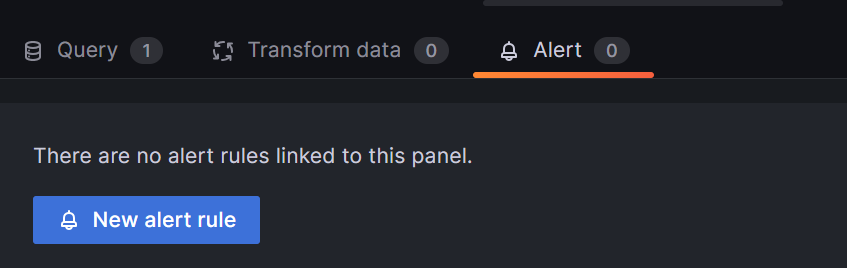


Click on Edit for configure Disk alert

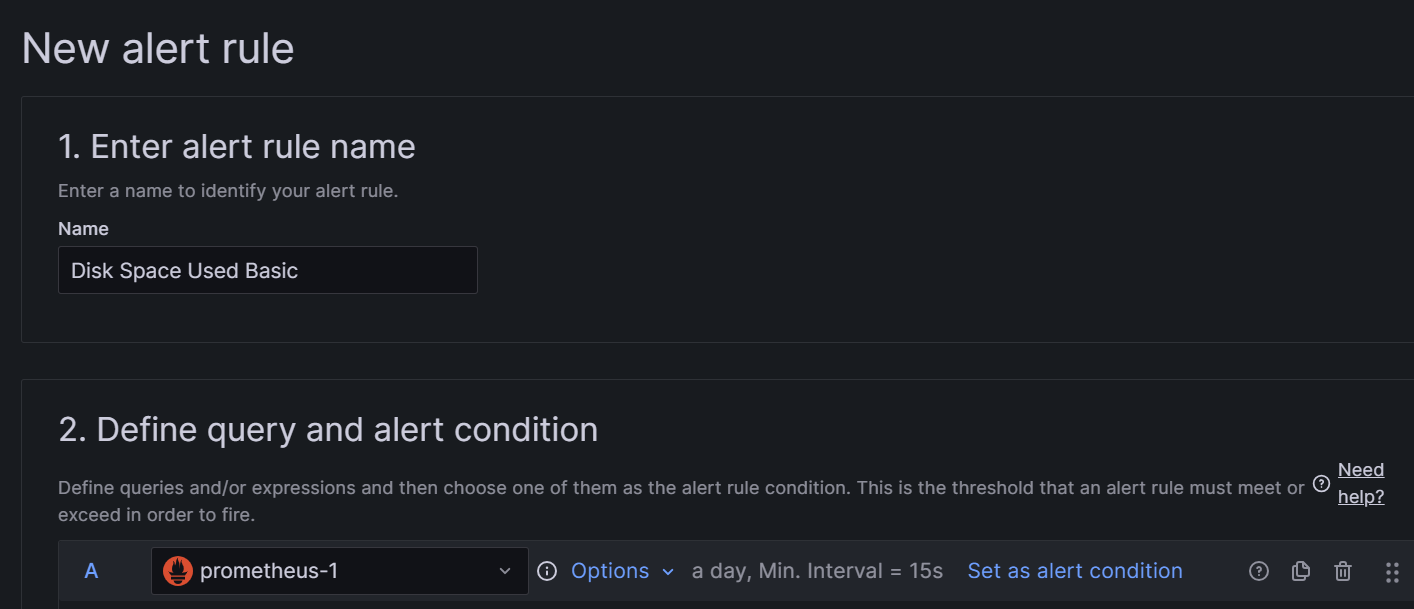


Click on alert

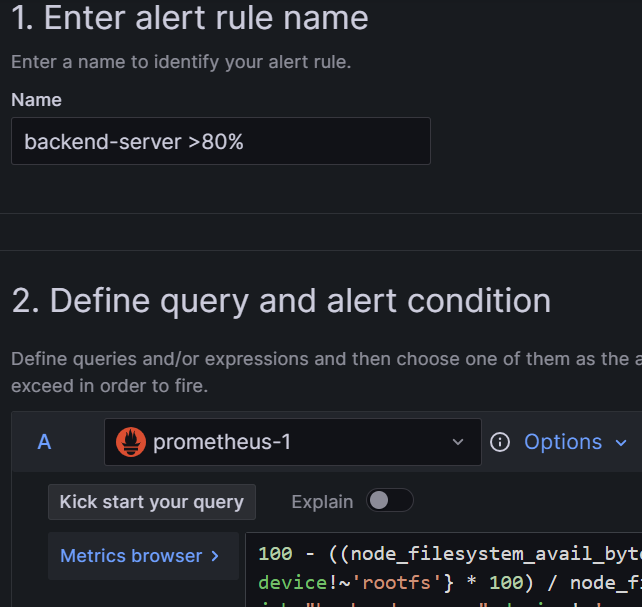




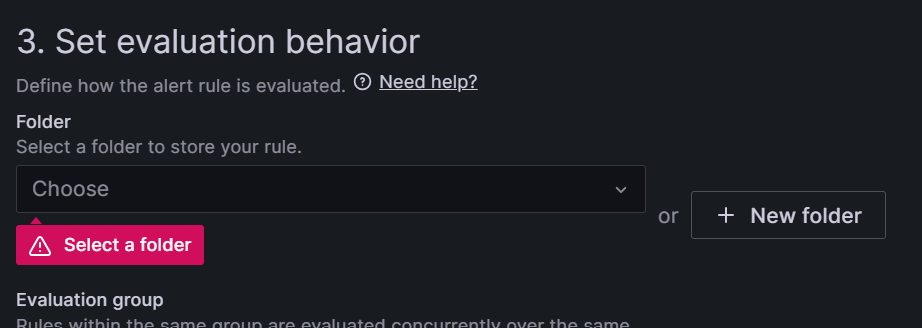
Click on “new alert rule” 🡪

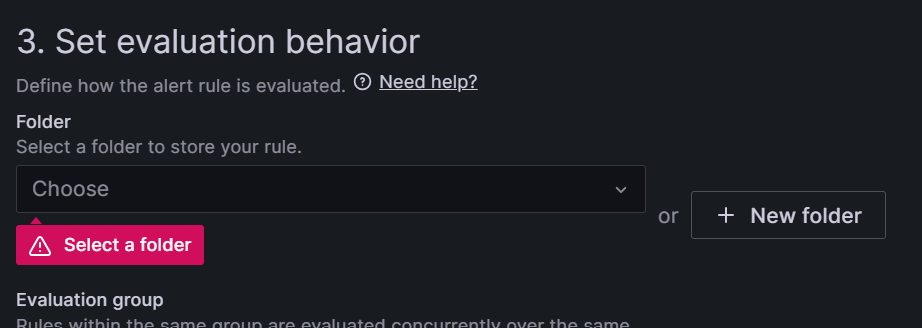


Give name

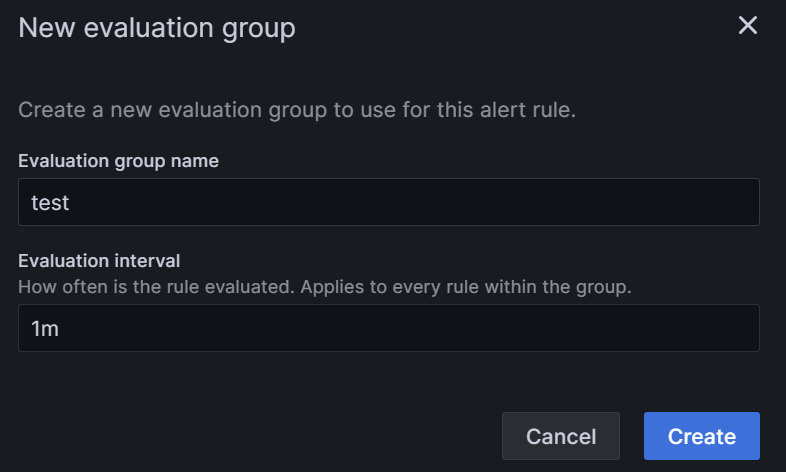


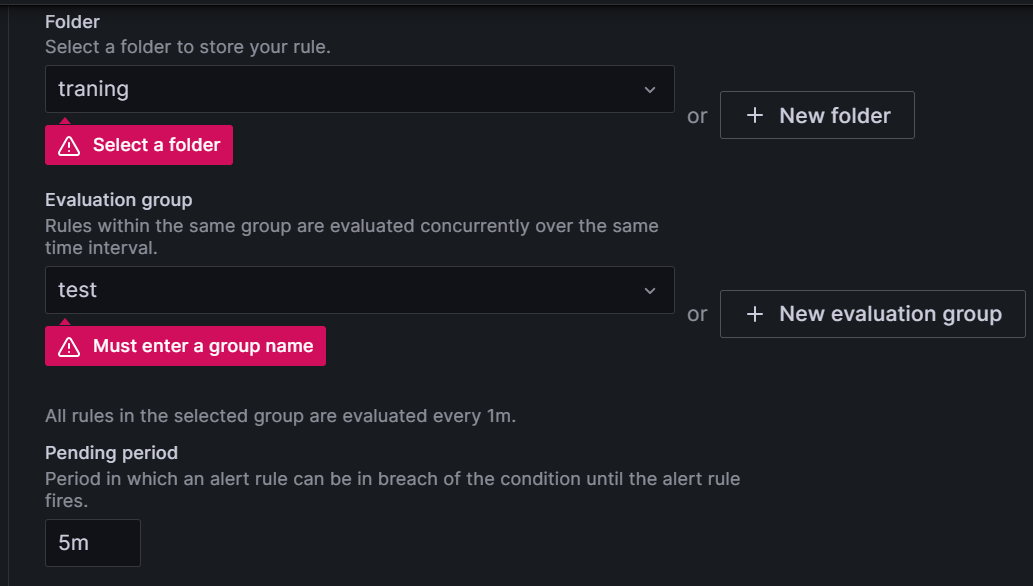
Need to create folder



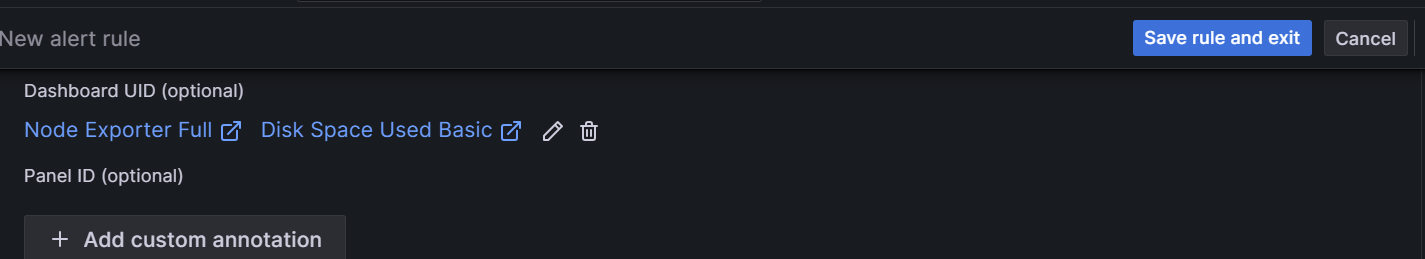


Create group name also:--

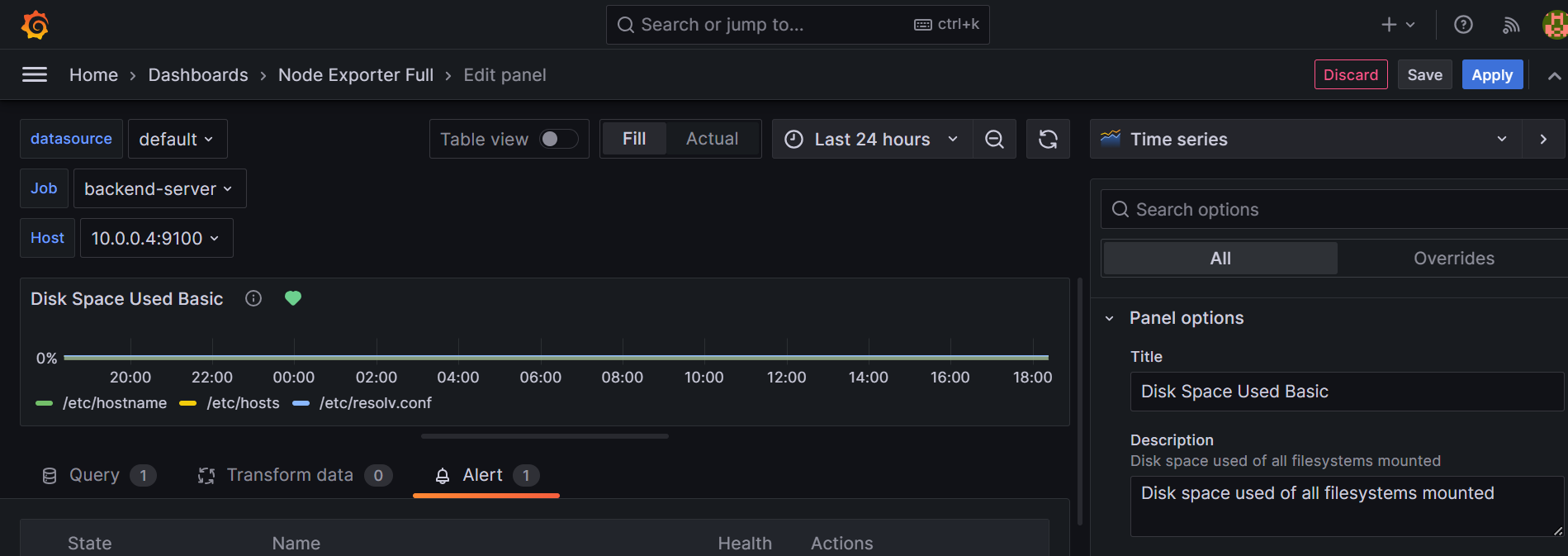




Save and exit



Then “Apply”



devops@backend-server:~$ df -Th

Filesystem Type Size Used Avail Use% Mounted on

tmpfs tmpfs 382M 1.3M 381M 1% /run

/dev/sda1 ext4 38G 4.5G 32G 13% /

tmpfs tmpfs 1.9G 0 1.9G 0% /dev/shm

tmpfs tmpfs 5.0M 0 5.0M 0% /run/lock

/dev/sda15 vfat 253M 150K 252M 1% /boot/efi

tmpfs tmpfs 382M 0 382M 0% /run/user/1000

Create few 10gb file:--

devops@backend-server:~$ fallocate -l 10G filename2

devops@backend-server:~$ fallocate -l 10G filename3

devops@backend-server:~$ ls -ltrh

total 31G

-rw-rw-r-- 1 devops devops 10G Feb 15 12:57 filename

-rw-rw-r-- 1 devops devops 10G Feb 15 12:57 filename2

-rw-rw-r-- 1 devops devops 10G Feb 15 12:57 filename3

devops@backend-server:~$ df -Th

Filesystem Type Size Used Avail Use% Mounted on

tmpfs tmpfs 382M 1.3M 381M 1% /run

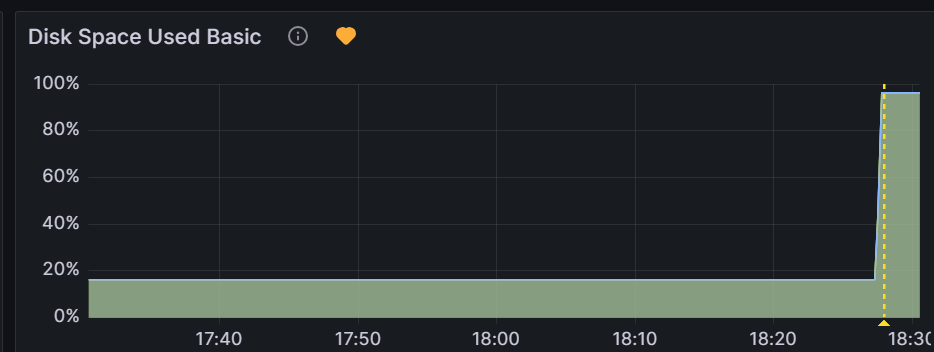
/dev/sda1 ext4 38G 35G 1.3G 97% /

tmpfs tmpfs 1.9G 0 1.9G 0% /dev/shm

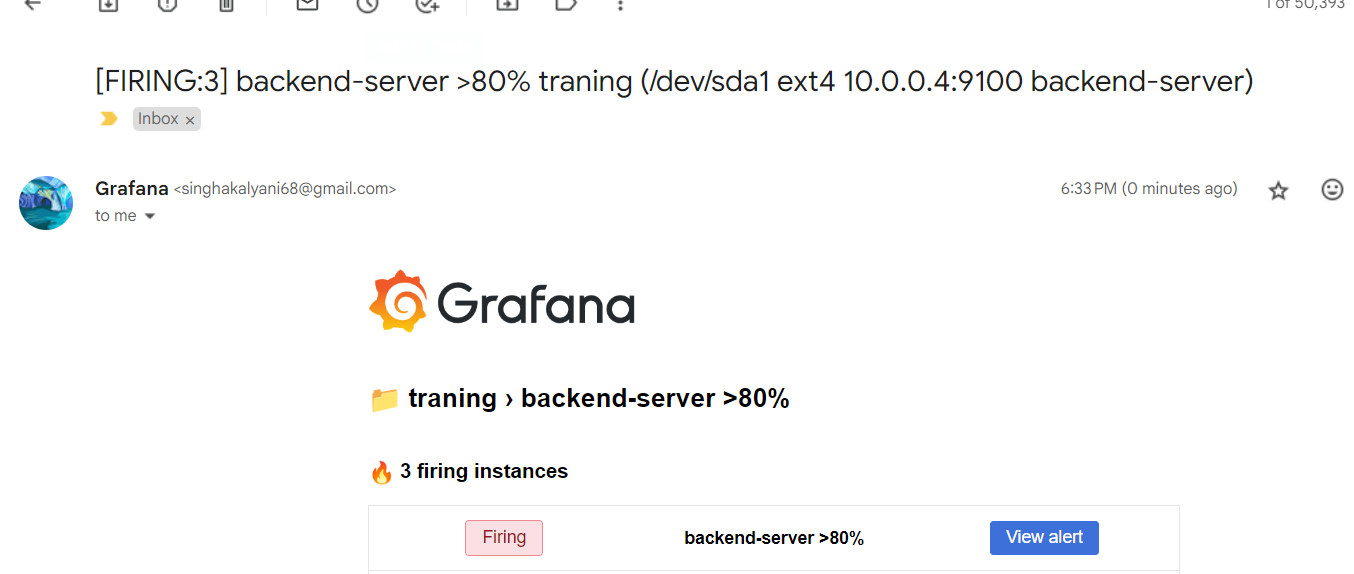
tmpfs tmpfs 5.0M 0 5.0M 0% /run/lock

/dev/sda15 vfat 253M 150K 252M 1% /boot/efi

tmpfs tmpfs 382M 0 382M 0% /run/user/1000



Got alert in mail



Now remove all 10gb files form server.

Docker compose file pull form this public repo:--

<https://github.com/anupam02us/anupam02us-prometheusdocker.git>

### All flutter app, angular build, script file([mariadb.sh](https://github.com/anupam02us/flutterbuild22/blob/main/mariadb.sh), [mariadbapi.py](https://github.com/anupam02us/flutterbuild22/blob/main/mariadbapi.py), [test.service](https://github.com/anupam02us/flutterbuild22/blob/main/test.service), keep in the below public repo:--

<https://github.com/anupam02us/flutterbuild22.git>

[mariadb.sh](https://github.com/anupam02us/flutterbuild22/blob/main/mariadb.sh) === this file help to create database, blank table and change bind address form localhost to any ip address, so any ip can access mariadb database.

[mariadbapi.py](https://github.com/anupam02us/flutterbuild22/blob/main/mariadbapi.py) == This file contain python Api code for backend server.

[test.service](https://github.com/anupam02us/flutterbuild22/blob/main/test.service) === This file contain code so that python api work as test.servce. I observe if I run mariadbapi.py file in background, it automatically exited after some time, so I create service(test.service) for running long time.

Latest flutter build file with able to connect with public ip of backend server, that on keep in this path:--

<https://github.com/anupam02us/flutterbuild22/tree/main/new_build_for_connection>

=======

frontend-server

Frontend-server running with nginx server.

#vi /var/www/html/ main.dart.js

Change ip address of backend server:

<http://95.217.169.228/get_data>

Then stop and start nginx:--

sudo systemctl stop nginx

sudo systemctl start nginx