Lesson 04 Demo 03

Configuring Email Alerts for Critical System Thresholds

Objective: To establish an email notification system in Grafana for alerting engineers when critical system metrics exceed predefined thresholds by configuring the necessary Grafana settings

Tools required: Linux operating system

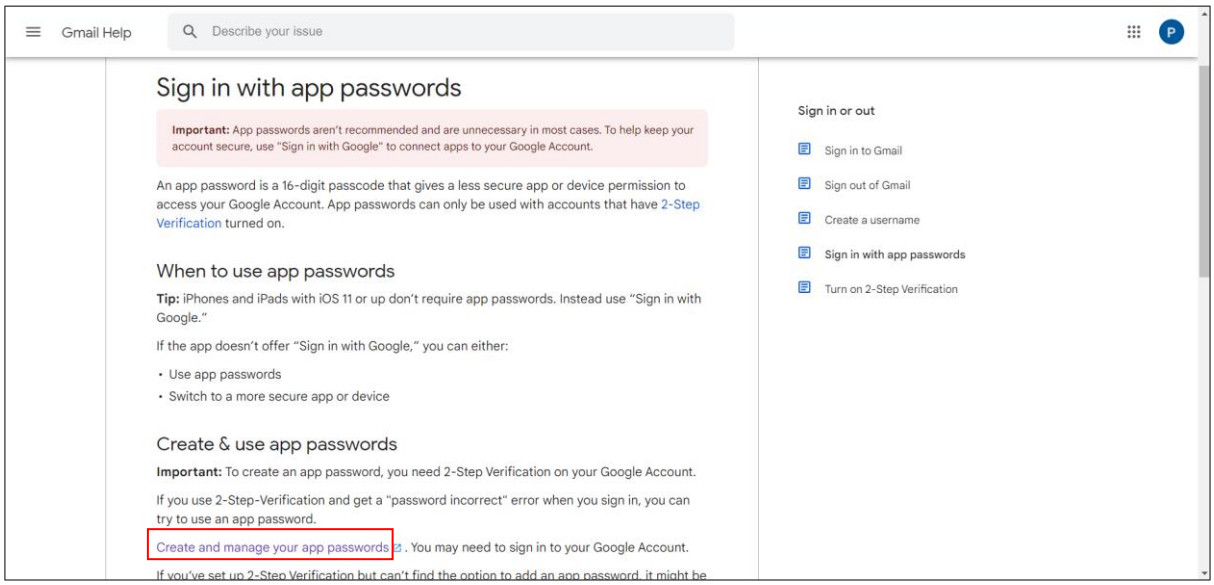
Prerequisites: Refer to Demos 01 and 02 of Lesson 04 for installing Grafana and configuring Prometheus as a data source

Steps to be followed:

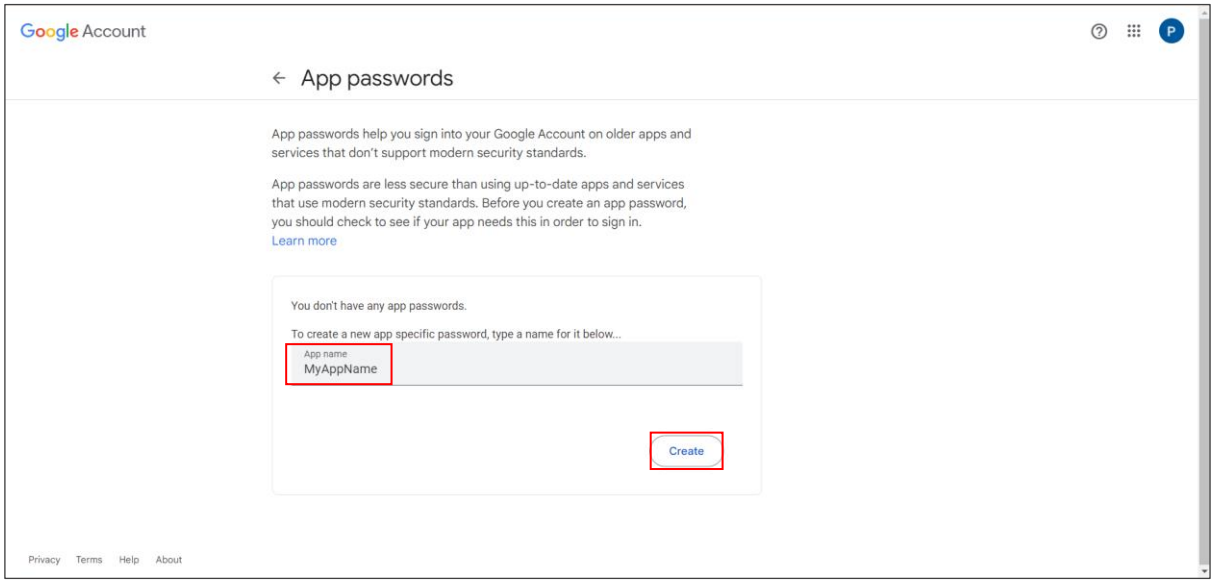
1. Set an app password through your Gmail account
2. Configure SMTP settings in the Grafana configuration file
3. Configure a contact point in the Grafana dashboard
4. Configure Notification policies
5. Configure alert rules and verify the email alert notifications

Step 1: Set an app password through your Gmail account

1.1 Navigate to Gmail using the following link to create an app password and then click on **Create and manage your app passwords**: **https://support.google.com/mail/answer/185833?hl=en**

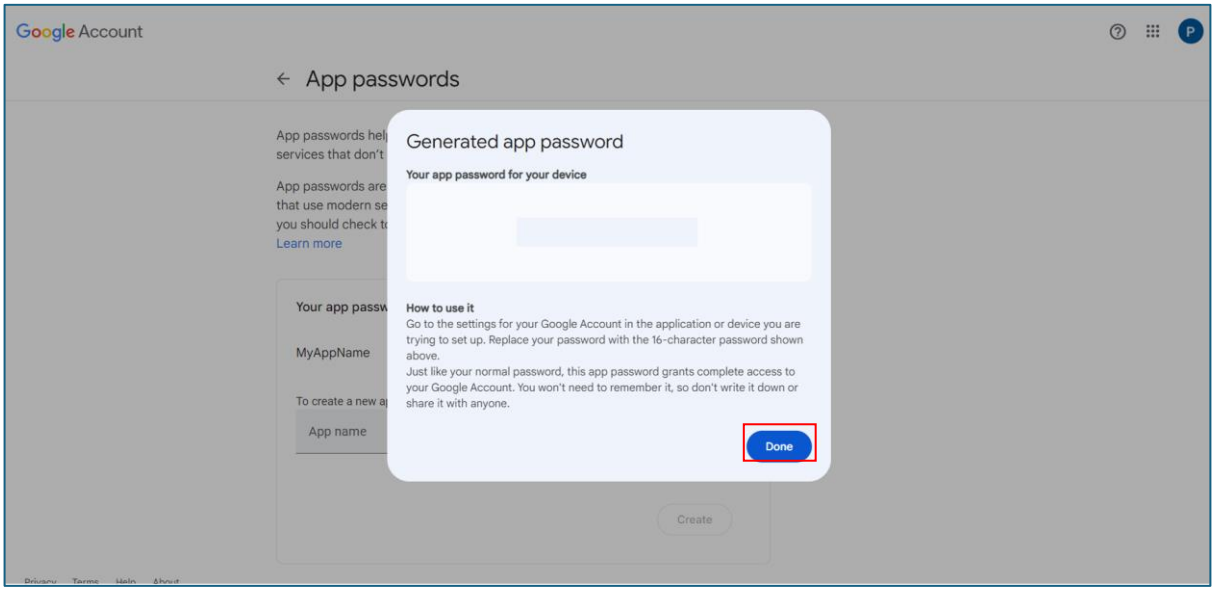


1.2 Provide the **App name** and click on **Create**



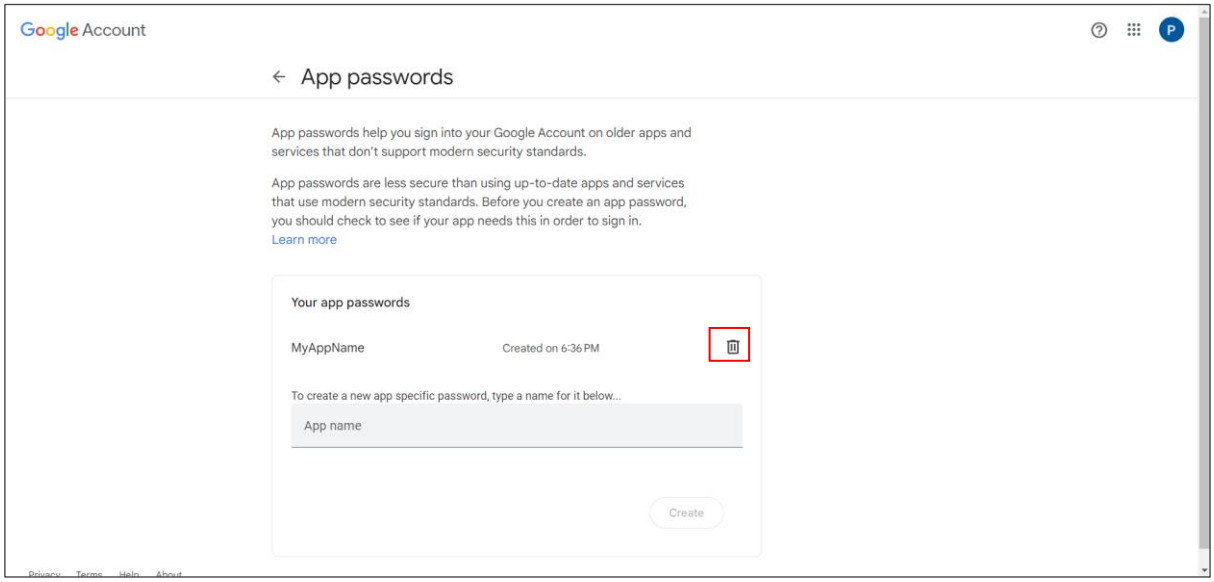
A 16-character password will be generated.

1.3 Click on **Done**



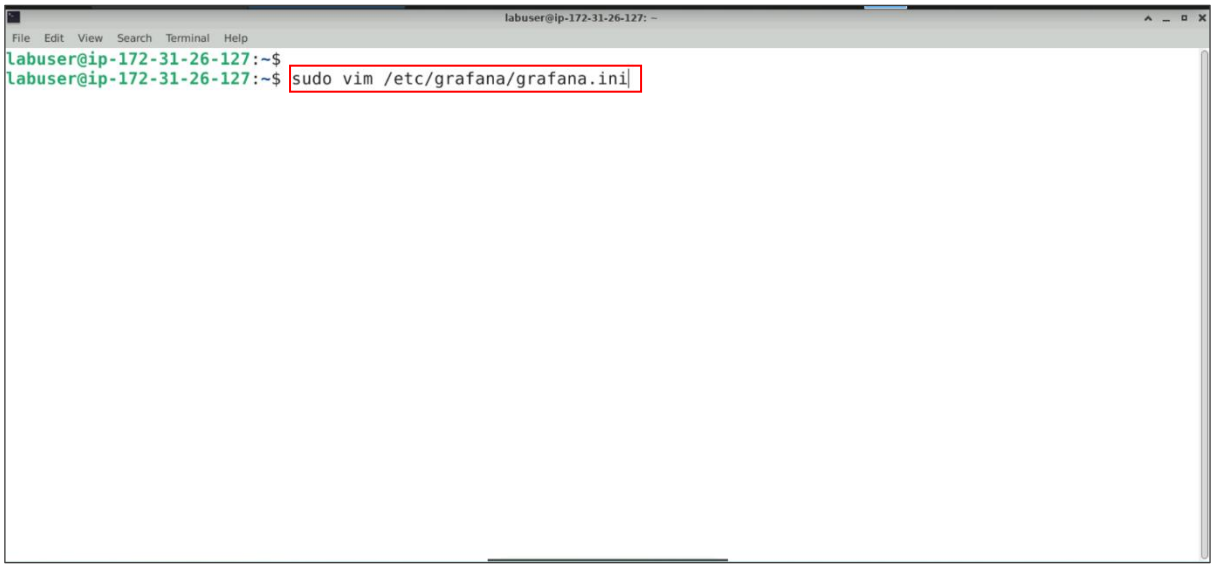
Note: The app password is in the format of **abcd efgh ijkl mnop**, but when typing it, do not use spaces; it should be **abcdefghijklmnop**.

1.4 For security reasons, remove the app password after use by clicking on the **delete** icon

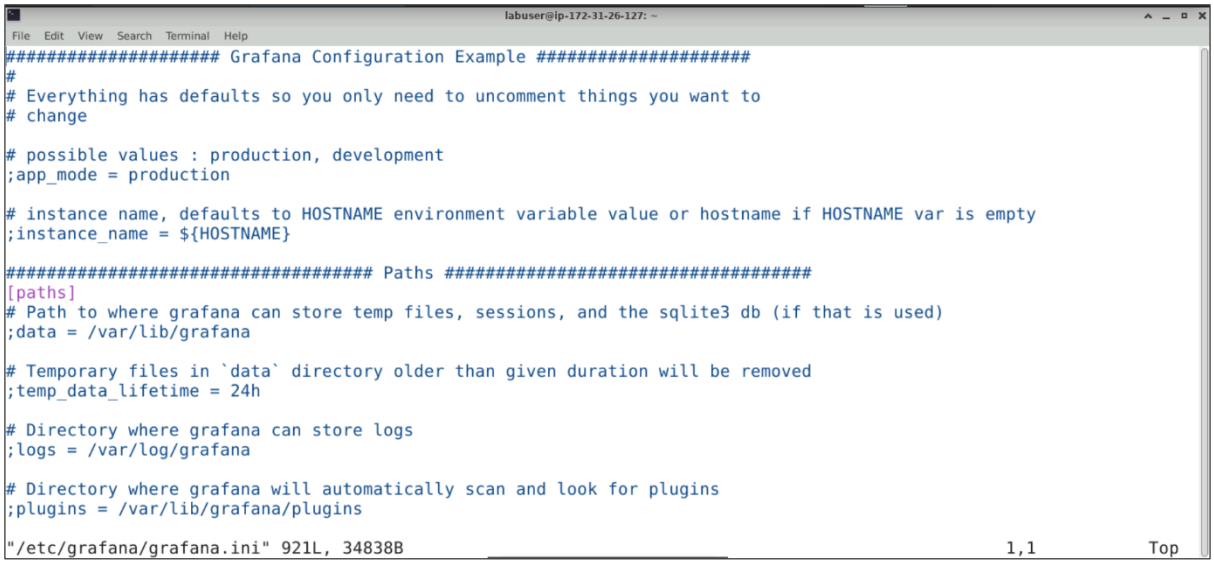


Step 2: Configure SMTP settings in the Grafana configuration file

2.1 Open the terminal and run the following command to edit the SMTP settings in the **grafana ini** file:
sudo vim /etc/grafana/grafana.ini



The ini file appears as shown below:



2.2 Find the **[smtp] section** in the file that appears as shown below:

```
labuser@ip-172-31-26-127: ~
File  Edit  View  Search  Terminal  Help
# Allow AWS users to assume a role using temporary security credentials.
# If true, assume role will be enabled for all AWS authentication providers that are specified in aws_auth_providers
; assume_role_enabled = true

################################### SMTP / Emailing #########################
[smtp]
;enabled = false
;host = localhost:25
;user =
# If the password contains # or ; you have to wrap it with triple quotes. Ex """#password;"""
;password =
;cert_file =
;key_file =
;skip_verify = false
;from_address = admin@grafana.localhost
;from_name = Grafana
# EHLO identity in SMTP dialog (defaults to instance_name)
;ehlo_identity = dashboard.example.com
# SMTP startTLS policy (defaults to 'OpportunisticStartTLS')
;startTLS_policy = NoStartTLS

[emails]
;welcome_email_on_sign_up = false
;templates_pattern = emails/*.html

                                                                    522,1         57%
```

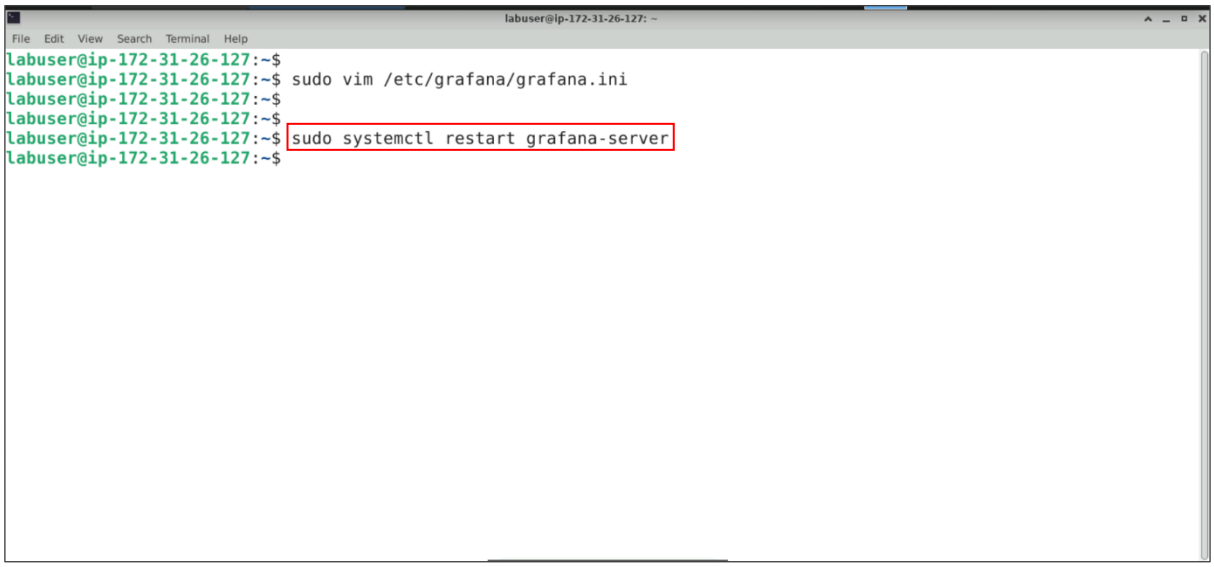
2.3 Configure the SMTP settings by updating the field values as follows:

[smtp]
enabled = true
host = smtp.gmail.com:587
user = <your_email@gmail.com>
password = <your_gmail_app_password>
skip_verify = true
from_address = <your_email@gmail.com>
from_name = Grafana

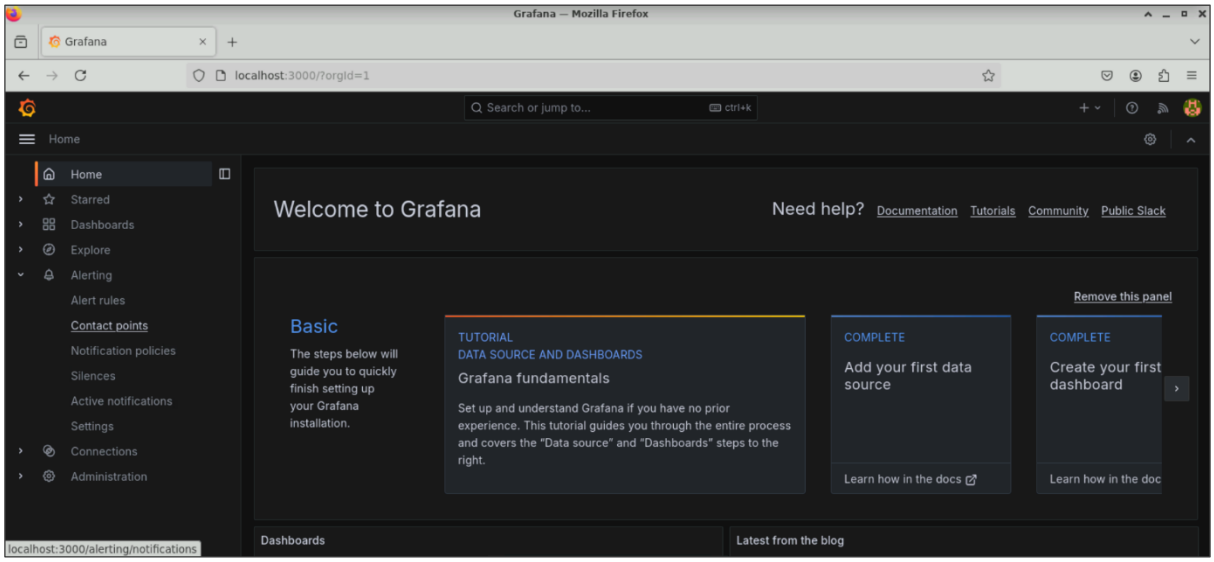
```
labuser@ip-172-31-26-127: ~
File  Edit  View  Search  Terminal  Help
# Allow AWS users to assume a role using temporary security credentials.
# If true, assume role will be enabled for all AWS authentication providers that are specified in aws_auth_providers
; assume_role_enabled = true

################################### SMTP / Emailing #########################
[smtp]
enabled = true
host = smtp.gmail.com:587
user = [redacted]
# If the password contains # or ; you have to wrap it with triple quotes. Ex """#password;"""
password = <your_gmail_app_password>
;cert_file =
;key_file =
skip_verify = true
from_address = [redacted]
from_name = Grafana
# EHLO identity in SMTP dialog (defaults to instance_name)
;ehlo_identity = dashboard.example.com
# SMTP startTLS policy (defaults to 'OpportunisticStartTLS')
;startTLS_policy = NoStartTLS

[emails]
;welcome_email_on_sign_up = false
;templates_pattern = emails/*.html

                                                                    522,6         57%
```

Note: Remove the comments before updating the values. Replace **<your_gmail_app_password>** with the password created in **Step 1**, and **<your_email@gmail.com>** with the actual email address to which the alert will be sent

2.4 Execute the following command to restart Grafana:
sudo systemctl restart grafana-server



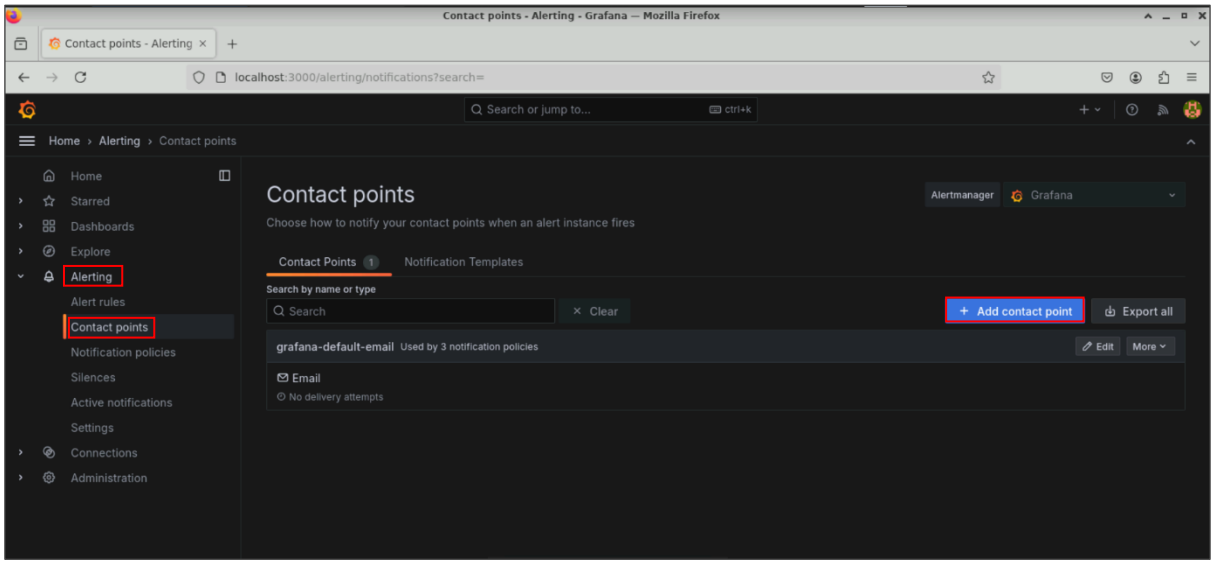
2.5 Open the preferred browser and enter the URL **http://localhost:3000** to open the Grafana UI as shown below:



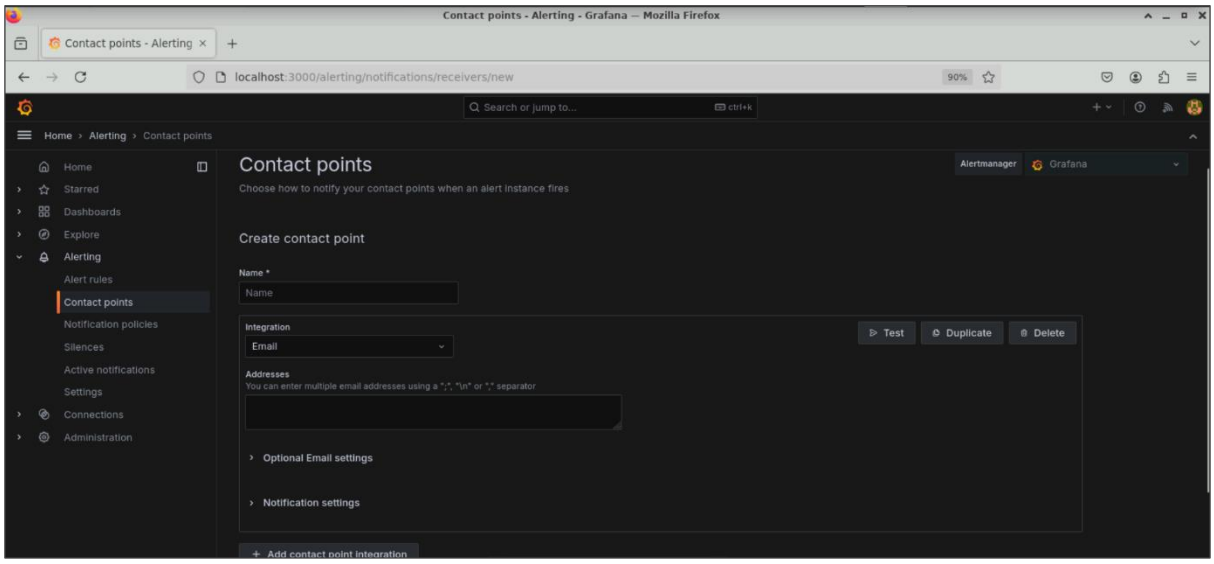
Note: Make sure that Prometheus is running before starting the Grafana server

Step 3: Configure a contact point in the Grafana dashboard

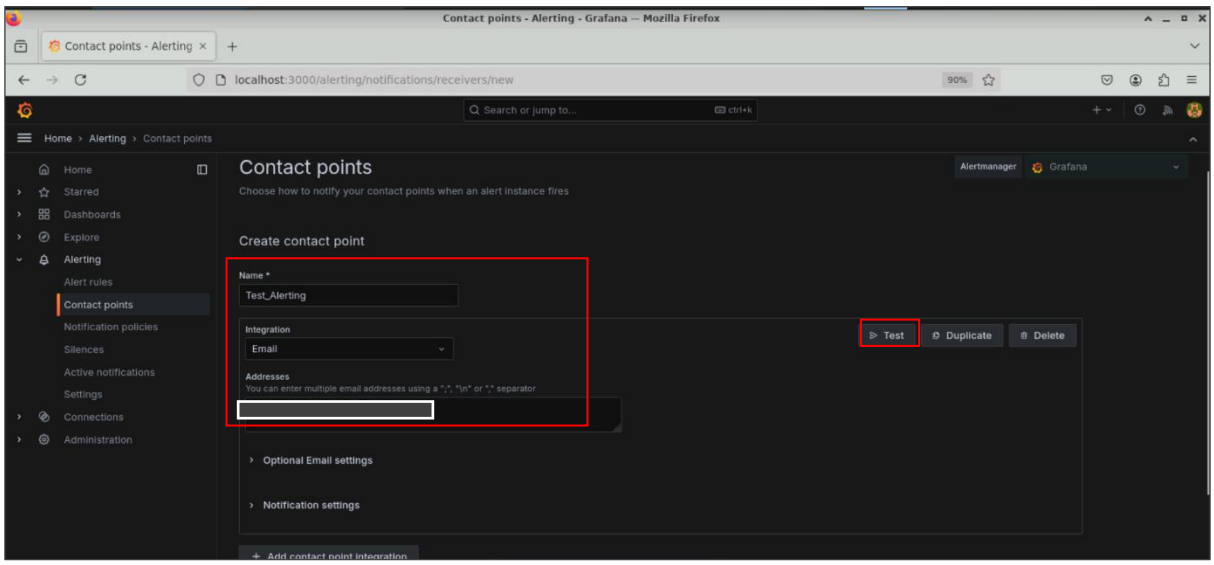
3.1 In the left-side menu, click on **Alerting**, select **Contact points**, and click on **+ Add contact point**



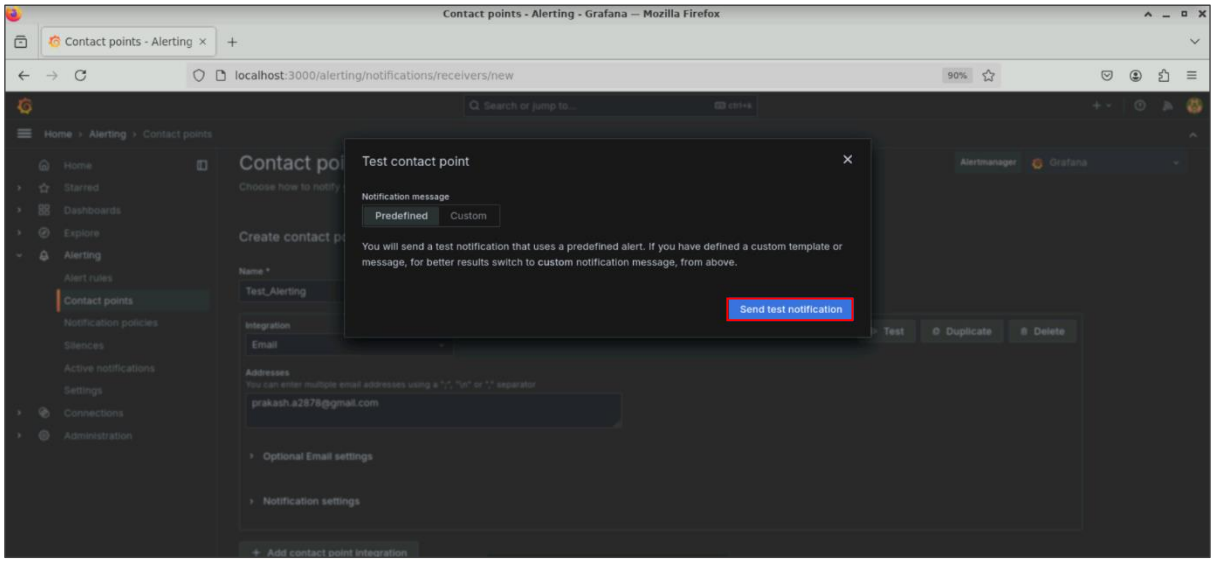
You will see the following interface:



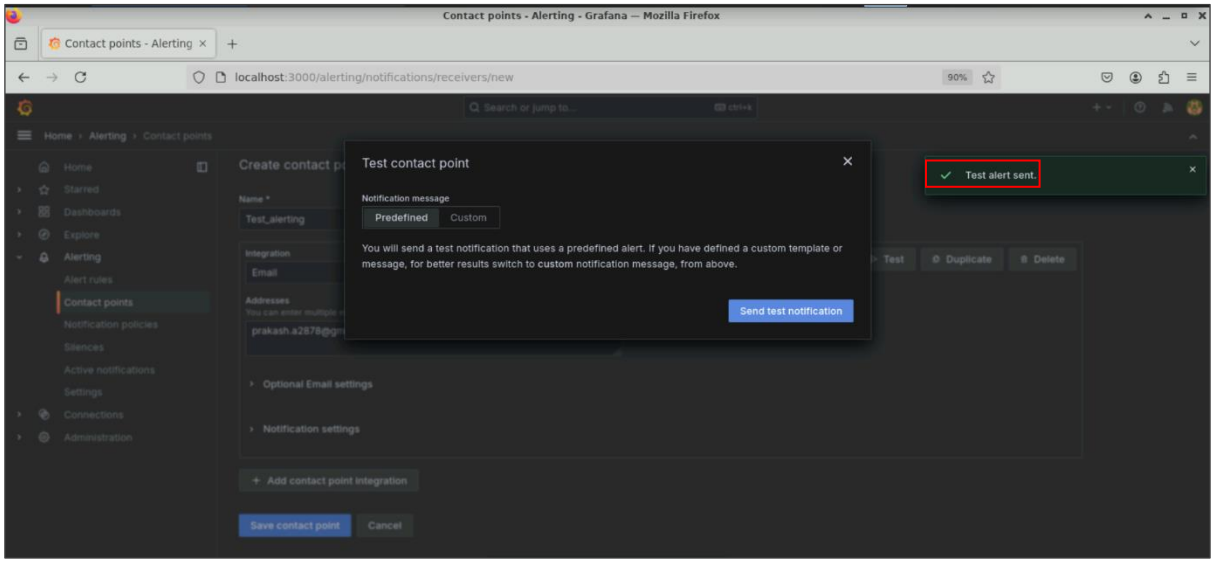
3.2 Fill in the **Name** field, select **Email** under **Integration**, enter the email addresses under the **Addresses** field to receive notifications, and click **Test** to check the configuration



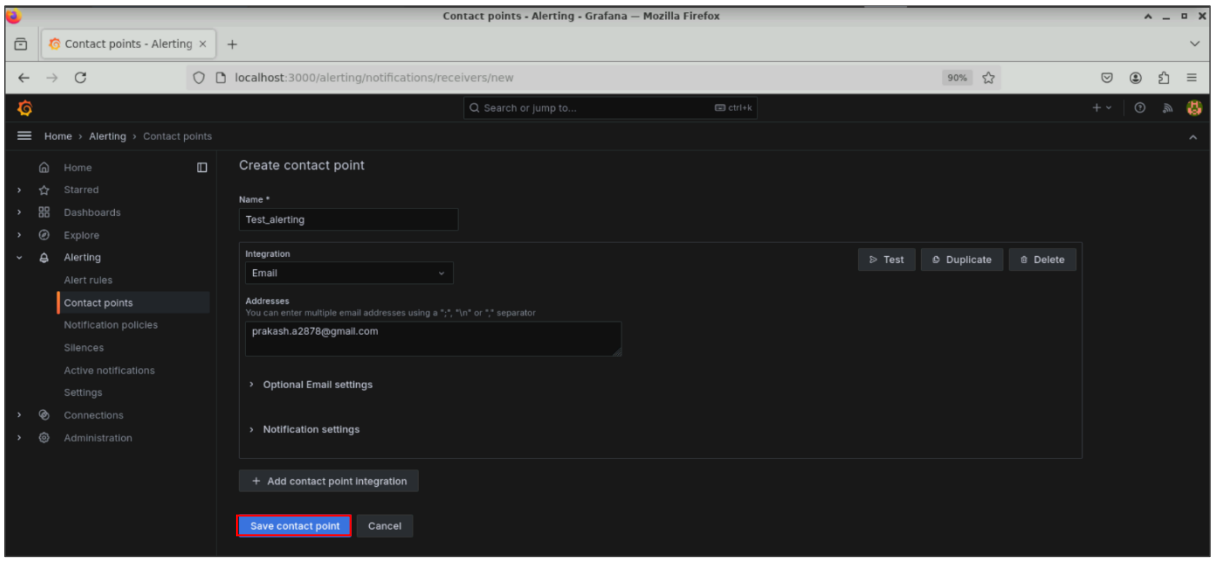
3.3 Click on **Send test notification**



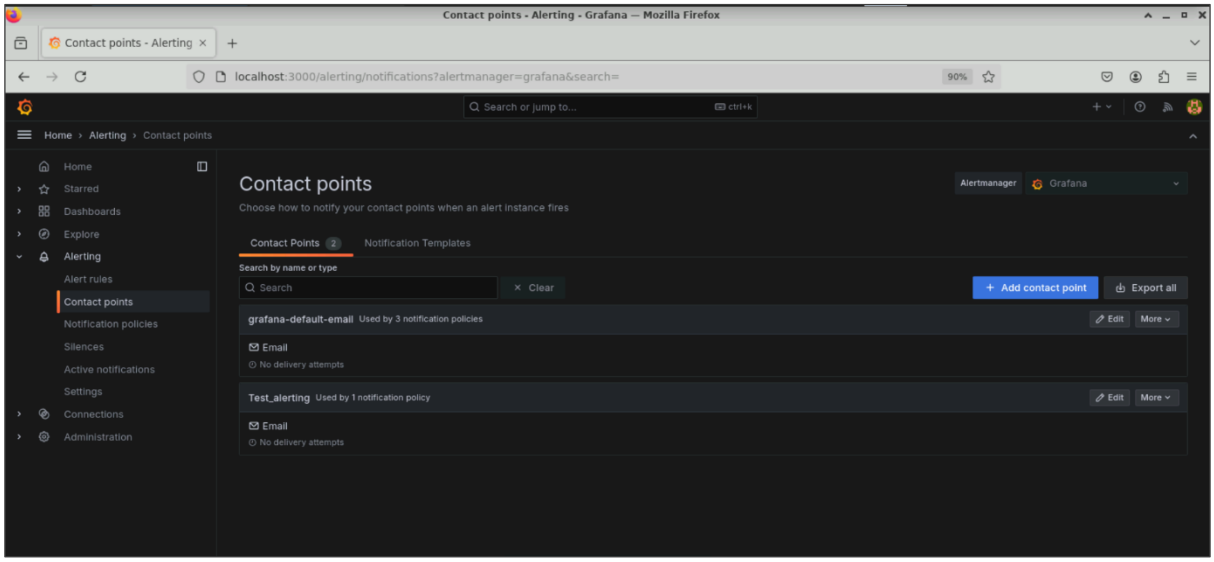
If the configuration is correct, it will show **Test alert sent** as shown below:



3.4 Save the settings by clicking **Save contact point**

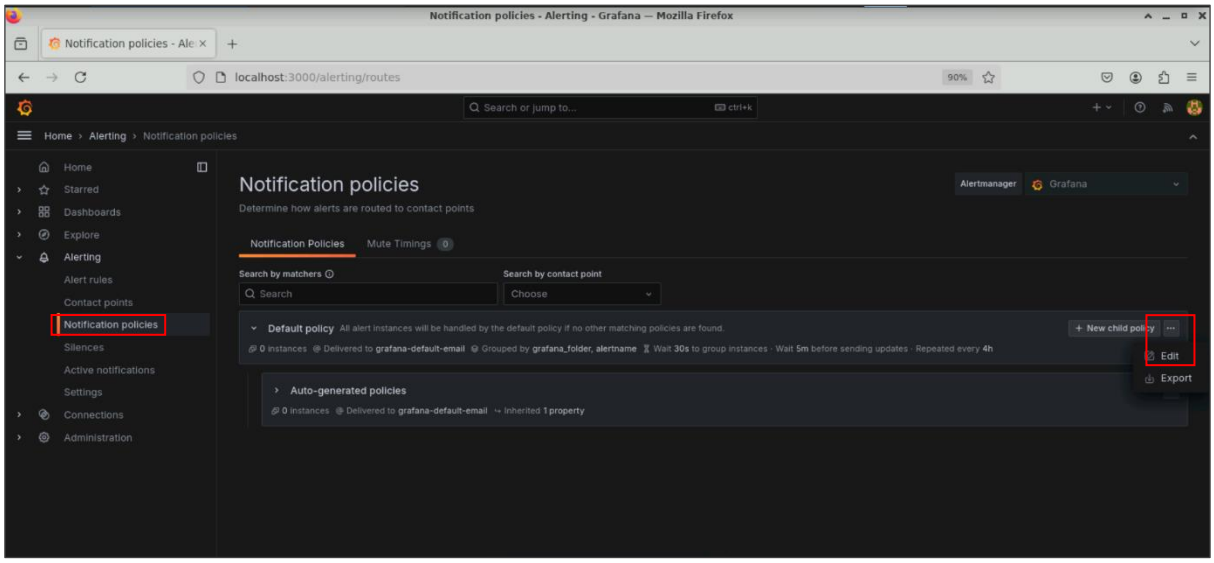


The contact point is created as shown below:

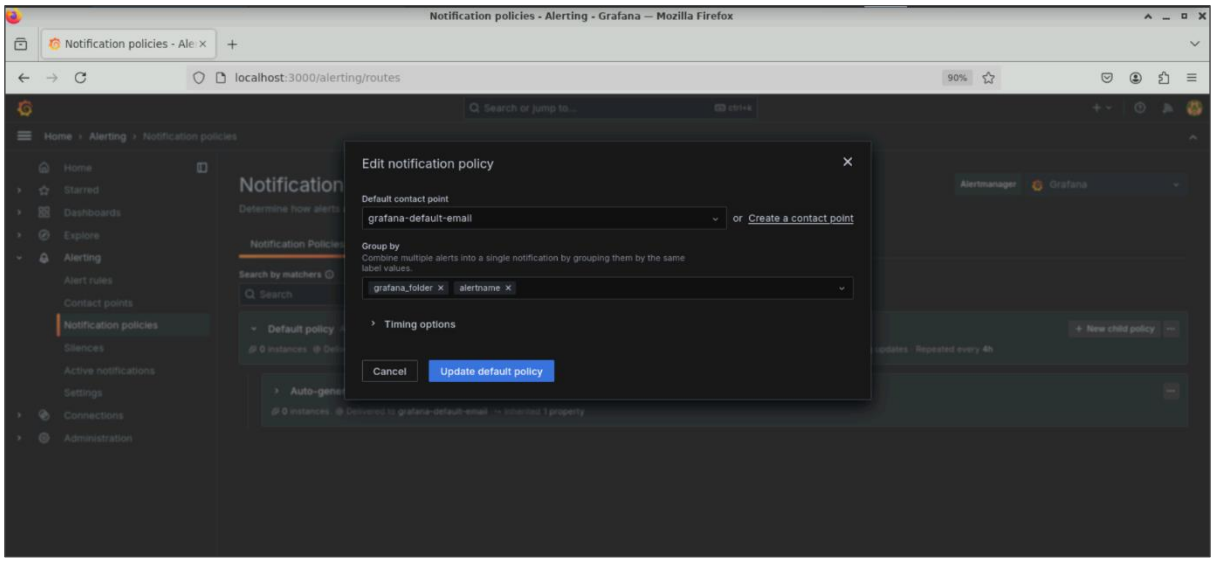


Step 4: Configure Notification policies

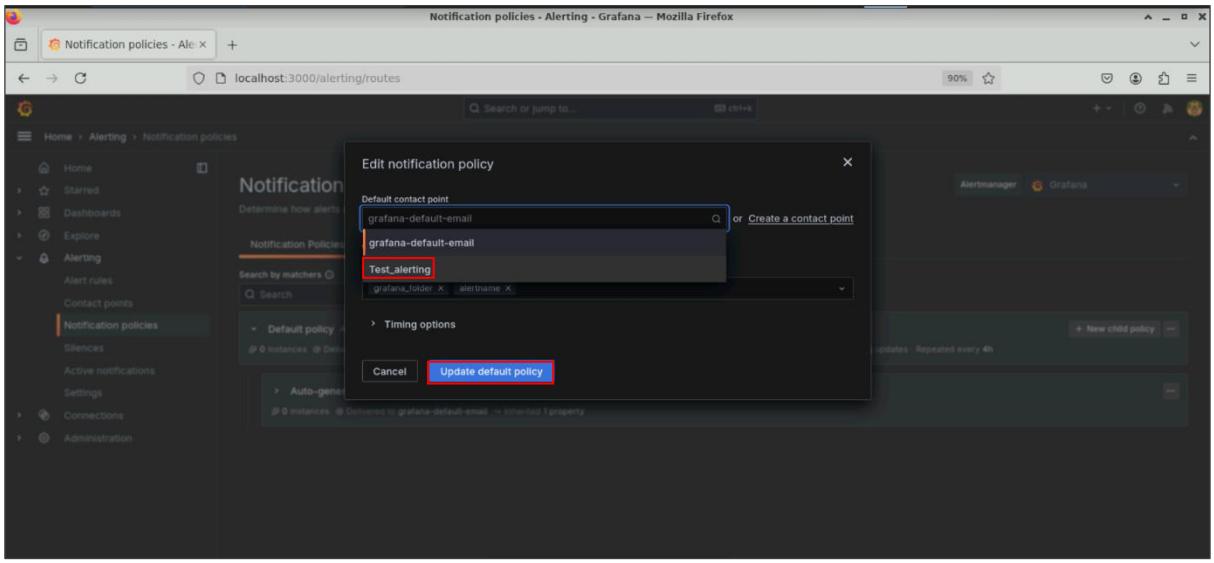
4.1 Select **Notification policies** on the left-side menu, click on the three dots **(…)**, and select **Edit**



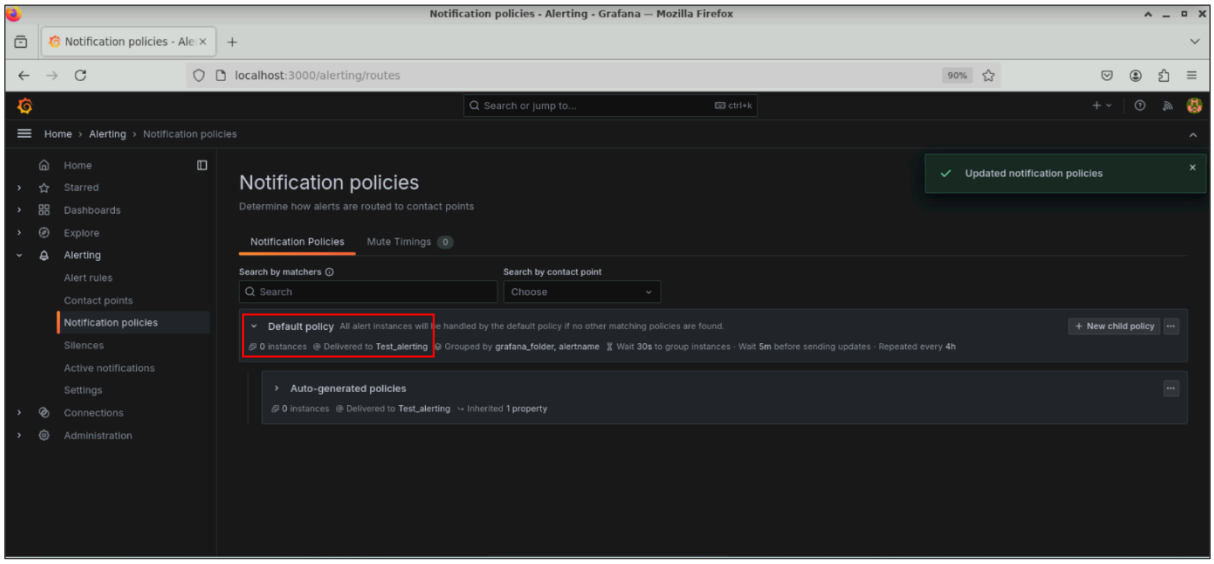
You can edit the notification in the window that pops up.



4.2 Select the contact point **Test_alerting** created earlier and click **Update default policy**

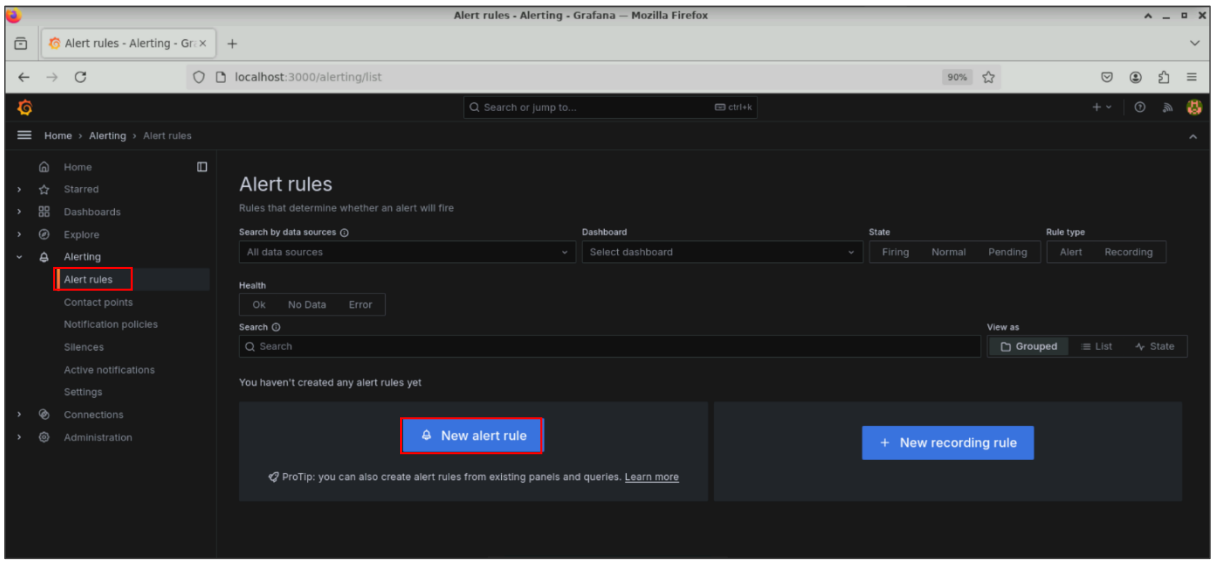


The **Notification policies** process is done. The default policy is changed to **Test_alerting**.

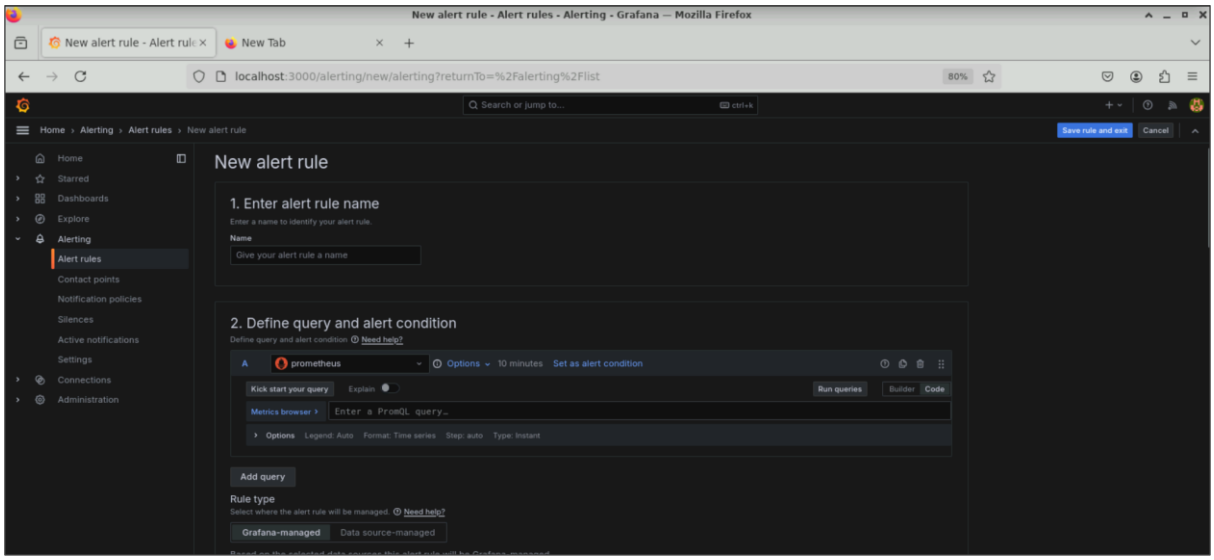


Step 5: Configure alert rules and verify the email alert notifications

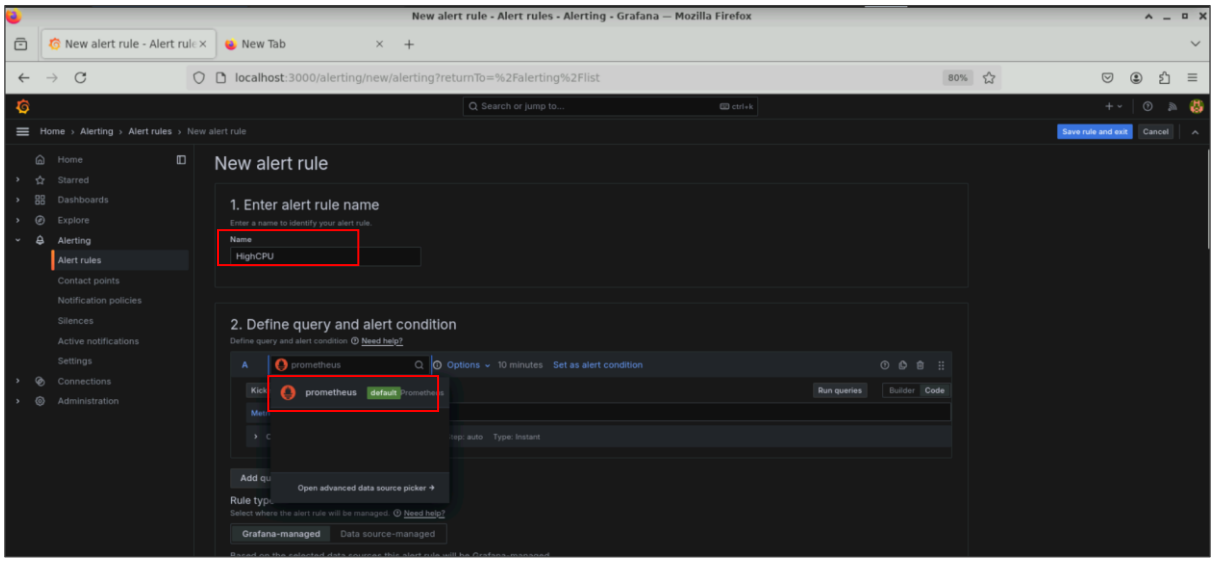
5.1 Click on **Alert rules** in the left-side menu and click **New alert rule**



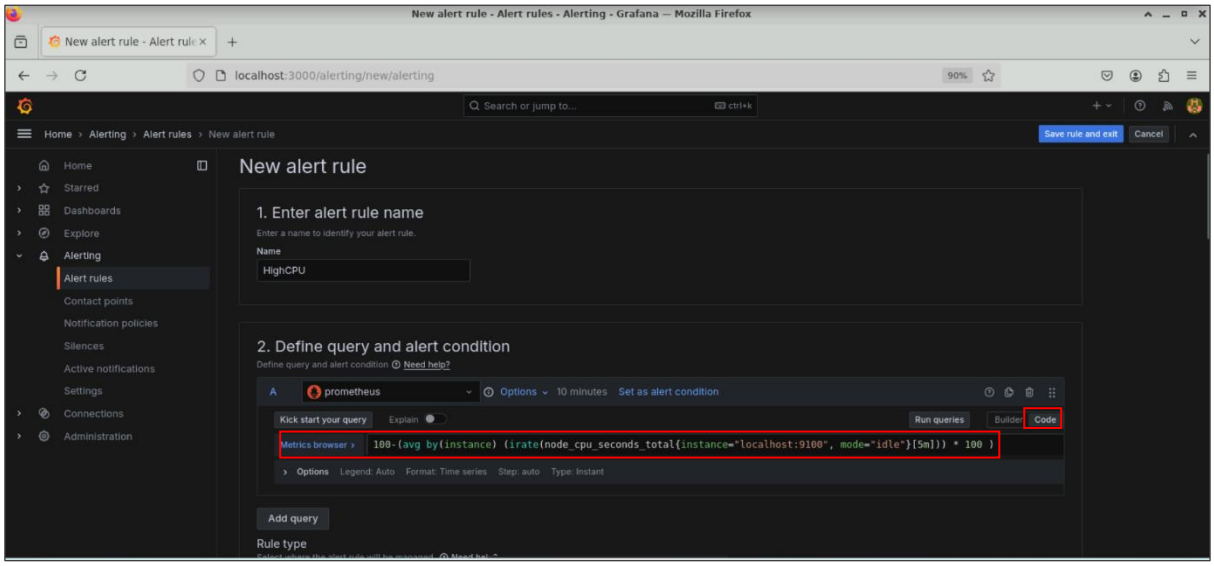
You will see the following interface:



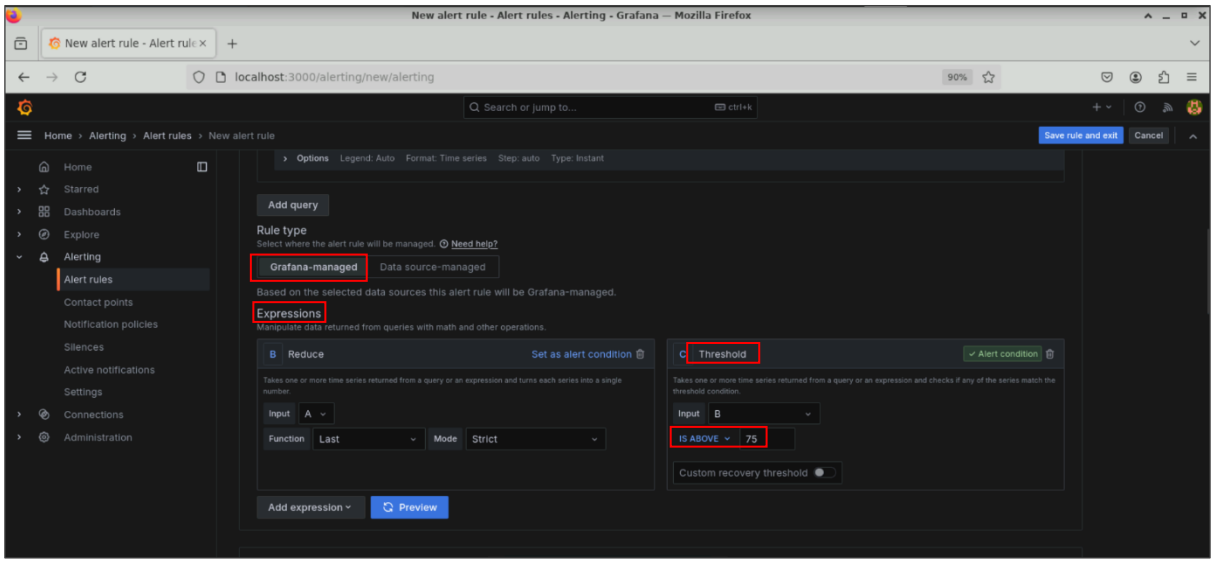
5.2 Put **HighCPU** as the **Name** under **Enter alert rule name** and select **prometheus** as the data source under **Define query and alert condition**



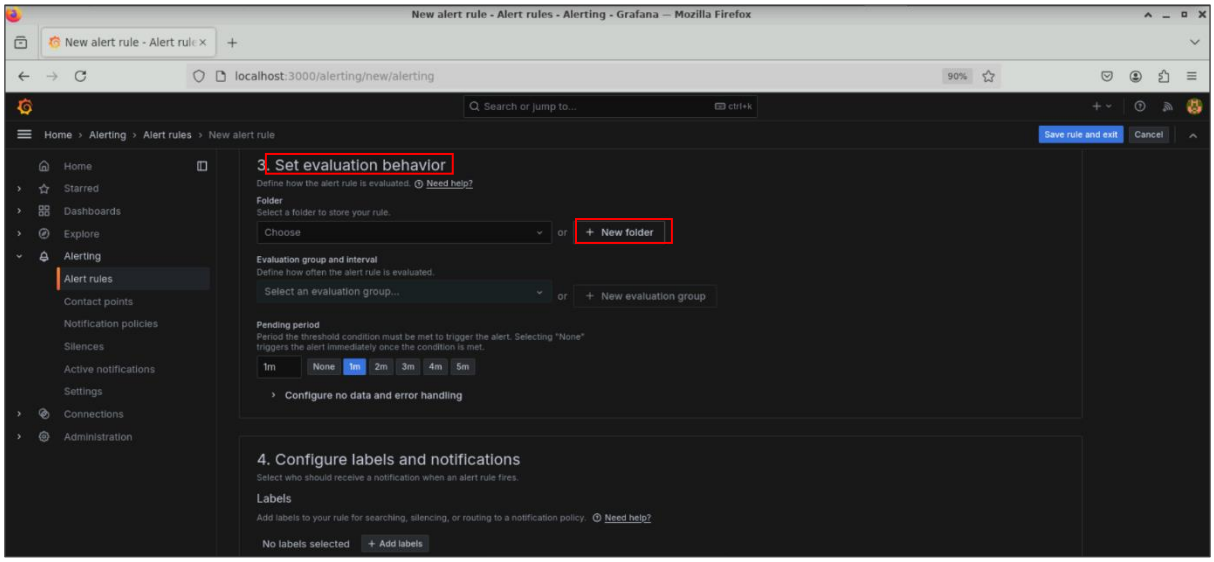
5.3 Click on the **code** editor and enter the following query in the **Metrics browser**: **100-(avg by(instance) (irate(node_cpu_seconds_total{instance="localhost:9100", mode="idle"}[5m])) * 100)**



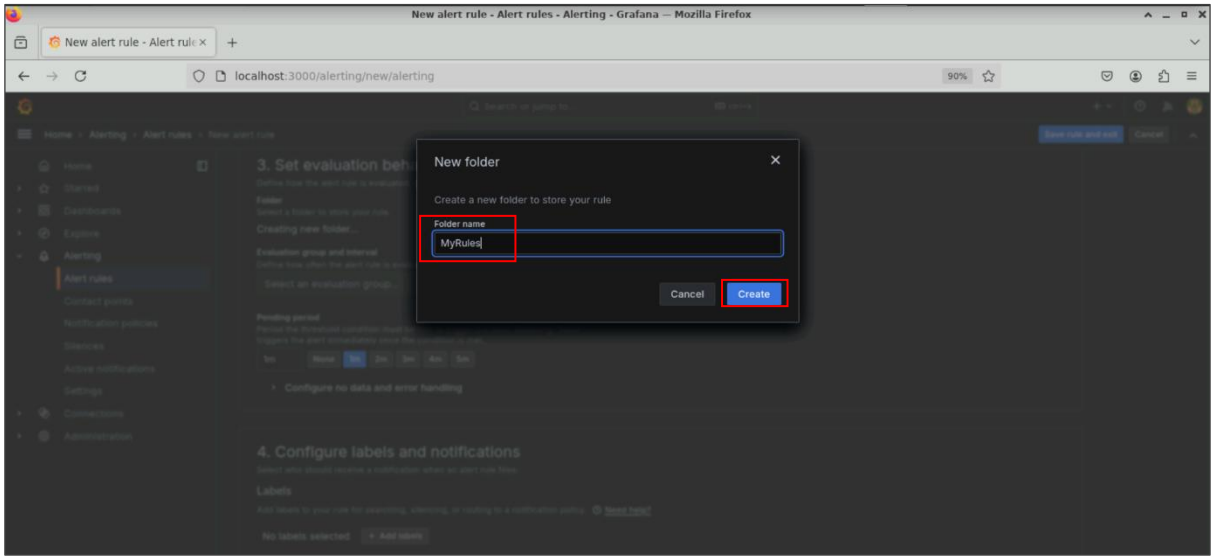
5.4 Select **Grafana-managed** as the **Rule type** and set the **Threshold** under **Expressions** to **IS ABOVE 75**



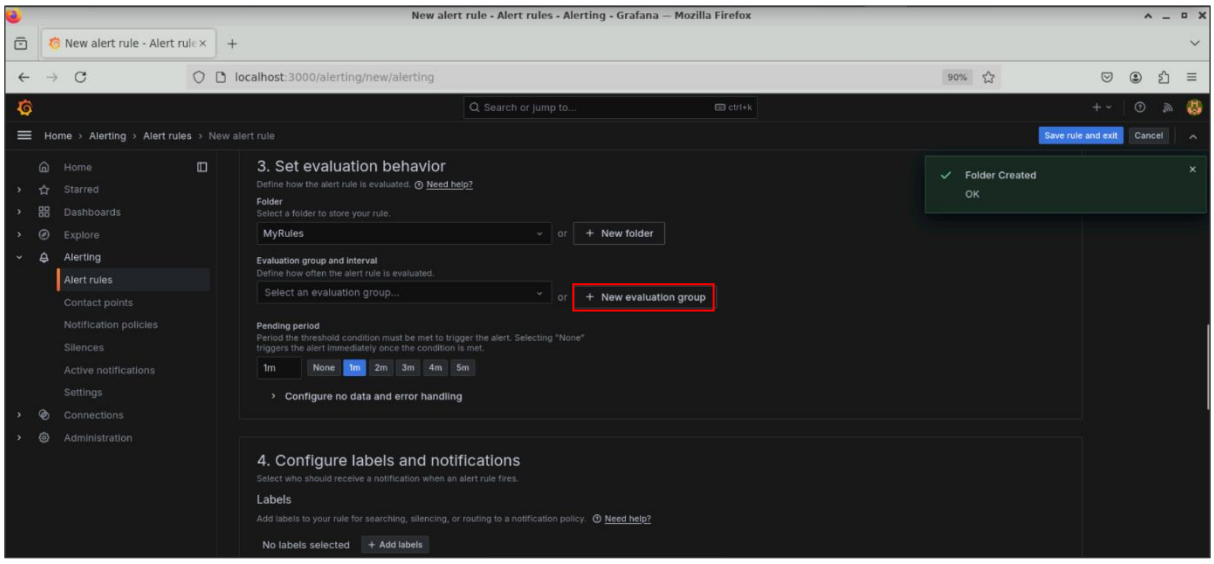
5.5 Scroll down; under **Set evaluation behavior**, click on **+ New folder**



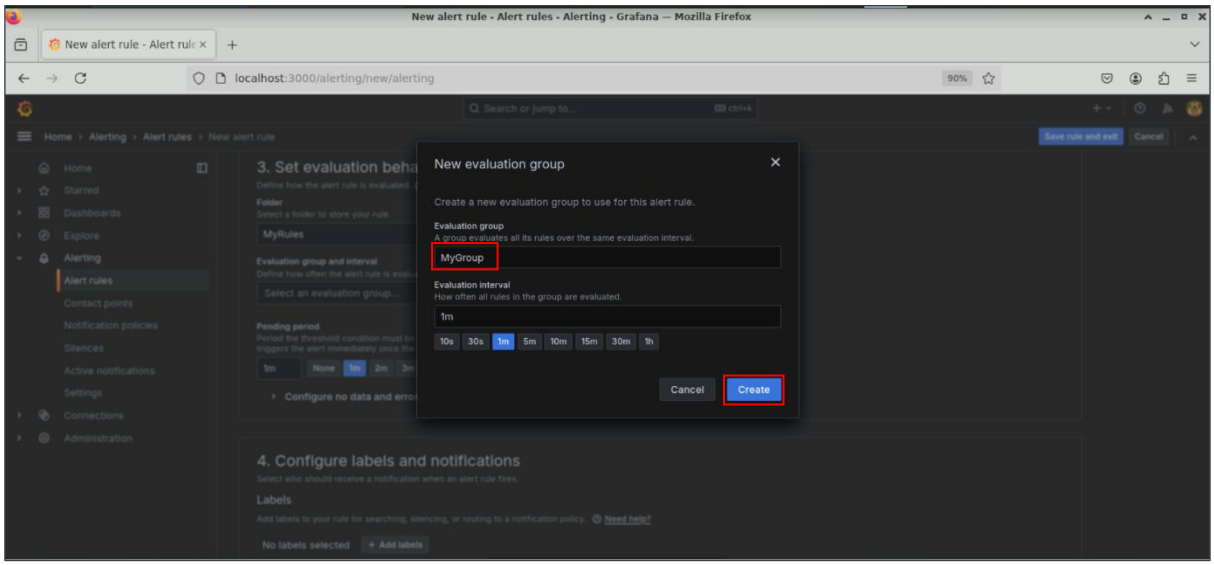
5.6 Name the folder **MyRules** and click **Create**



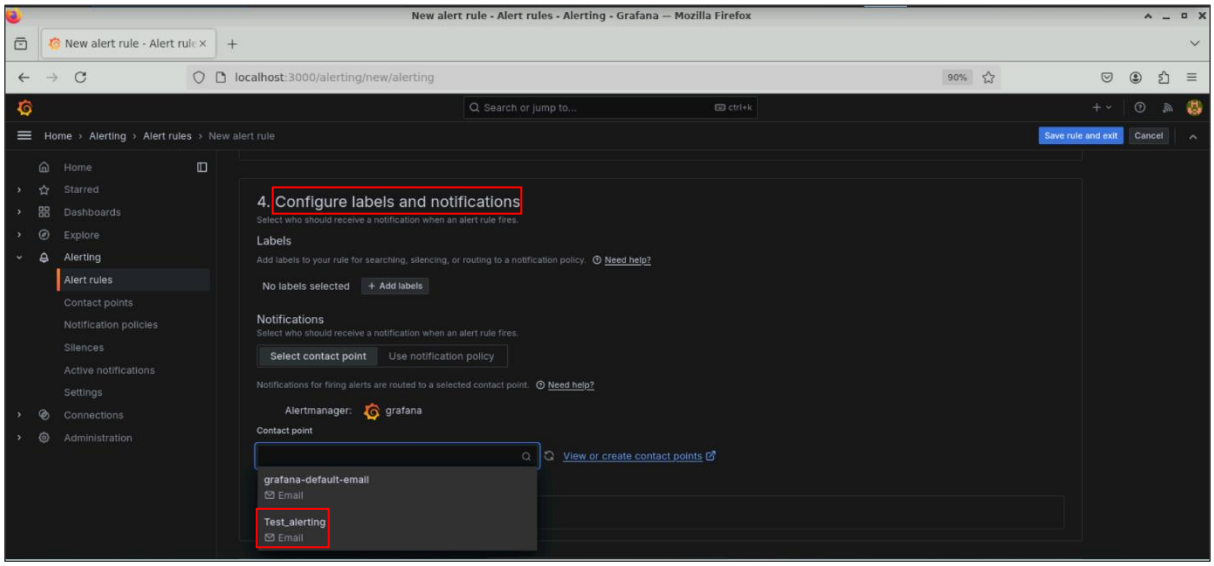
5.7 Under **Set evaluation behavior**, click on **+ New evaluation group**



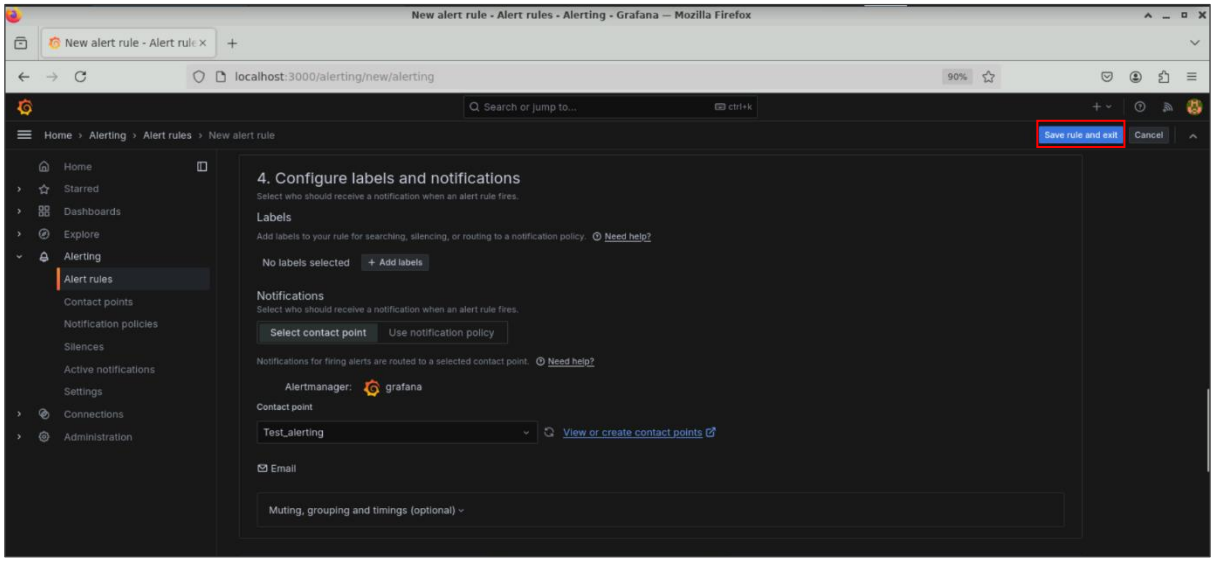
5.8 Name the **Evaluation group** as **MyGroup** and set the **Evaluation interval** to **1m**; then, click on **Create**



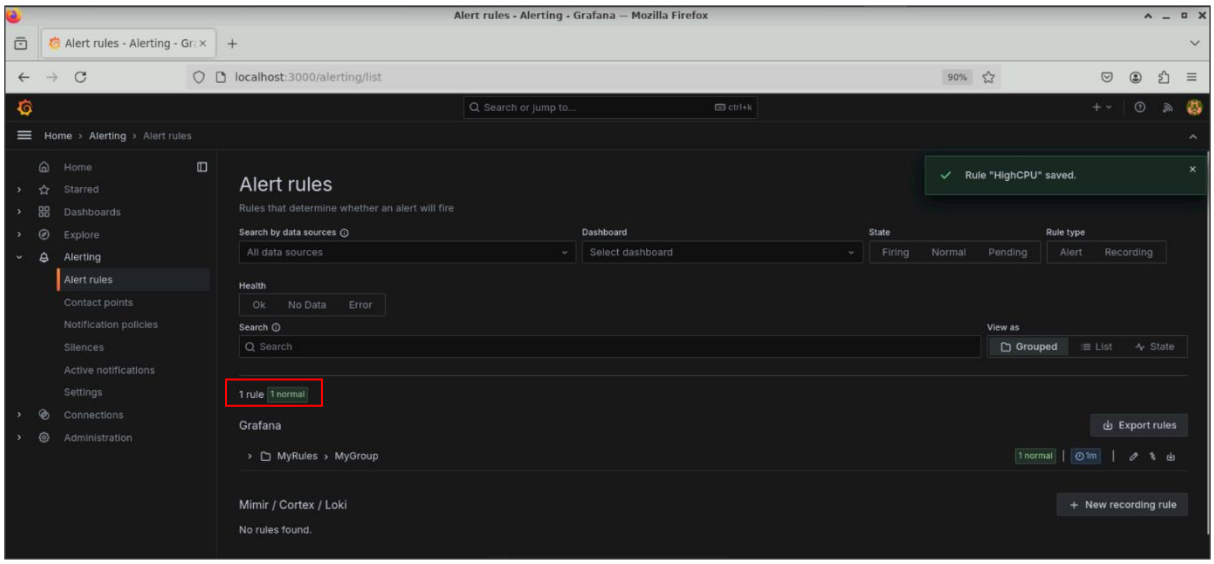
5.9 Scroll down to **Configure labels and notifications** and select **Test_alerting** as the **Contact point** created earlier



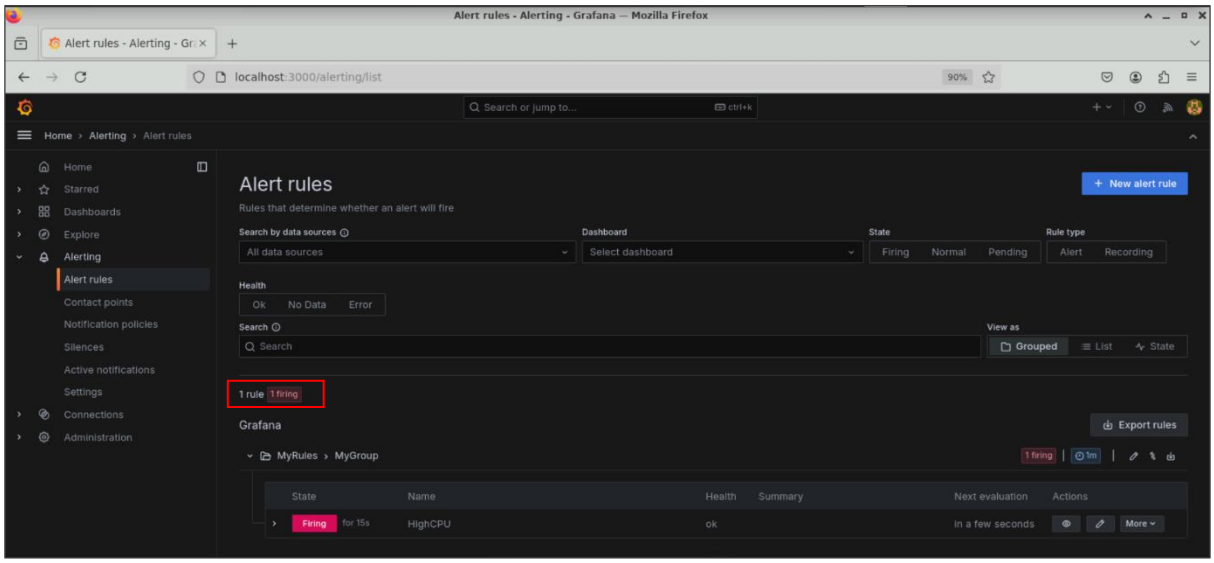
5.10 Click on **Save rule and exit** to save all configurations



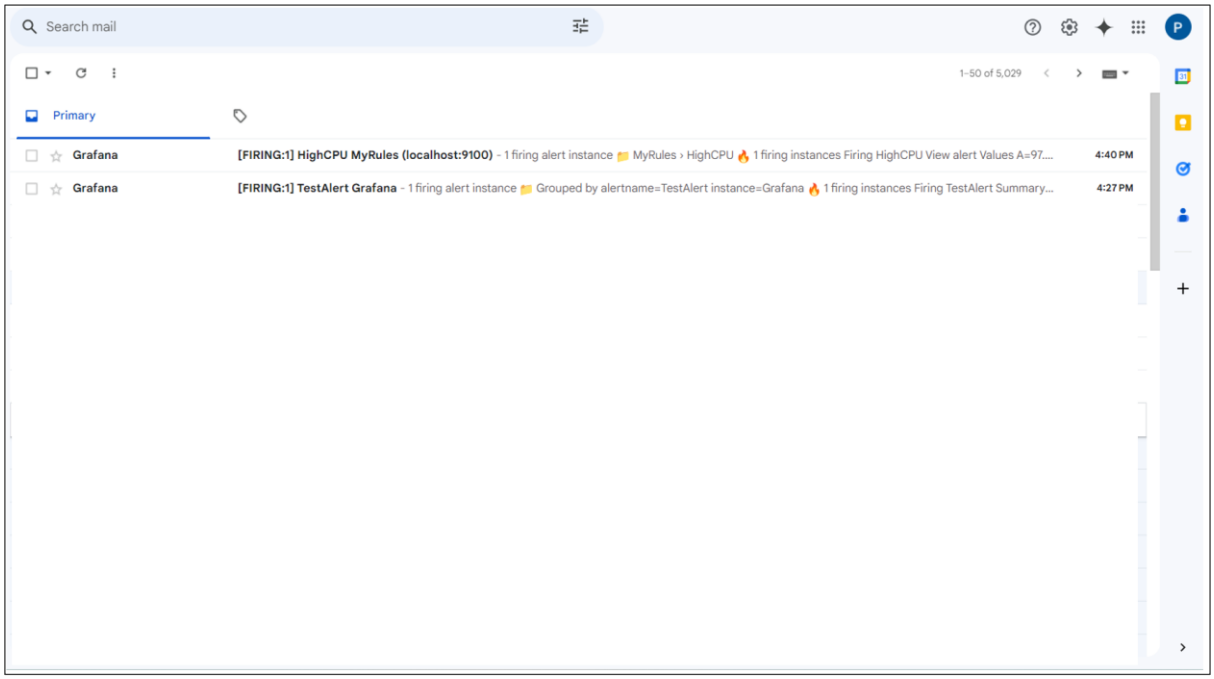
The alert rule is created. Initially, the CPU is normal as shown below:



When the CPU utilization goes above the configured threshold, the alert status changes to **Firing** as shown below:

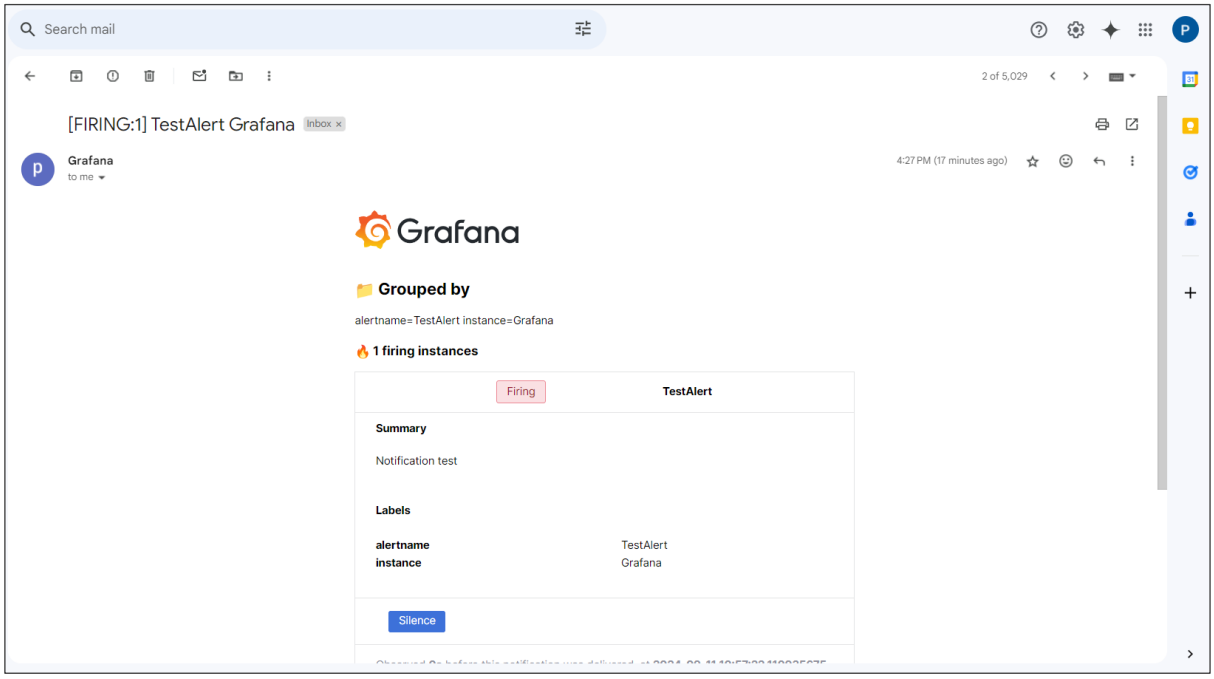


Check the email for alert notifications sent to the contact point addresses

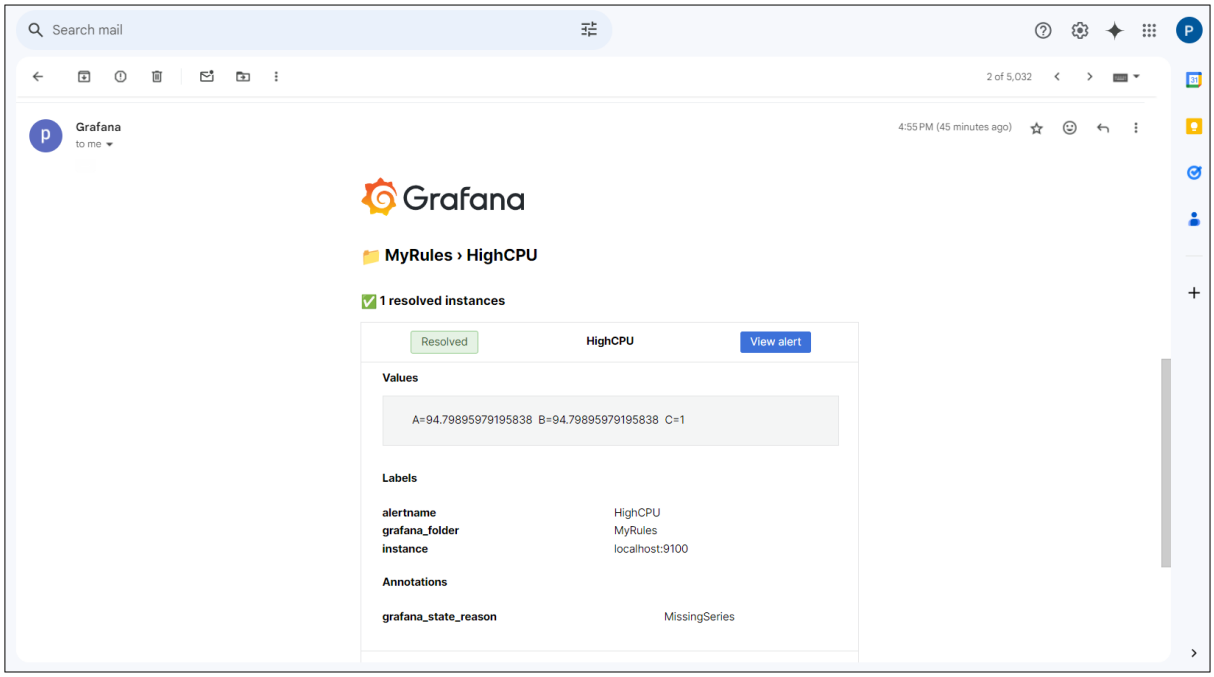


Note: The email IDs mentioned in **Addresses** under **Contact point** are the ones receiving alert notifications.

The following email shows the alert for the firing instance:



The email below shows that the CPU usage is normal and below the 75% threshold:



By following these steps, you have successfully configured SMTP settings, set up contact points, and created alert rules for email notifications to monitor system metrics, ensuring timely alerts for critical threshold breaches.