Observing Cloud Resources

Name: Rajdeep Sharma

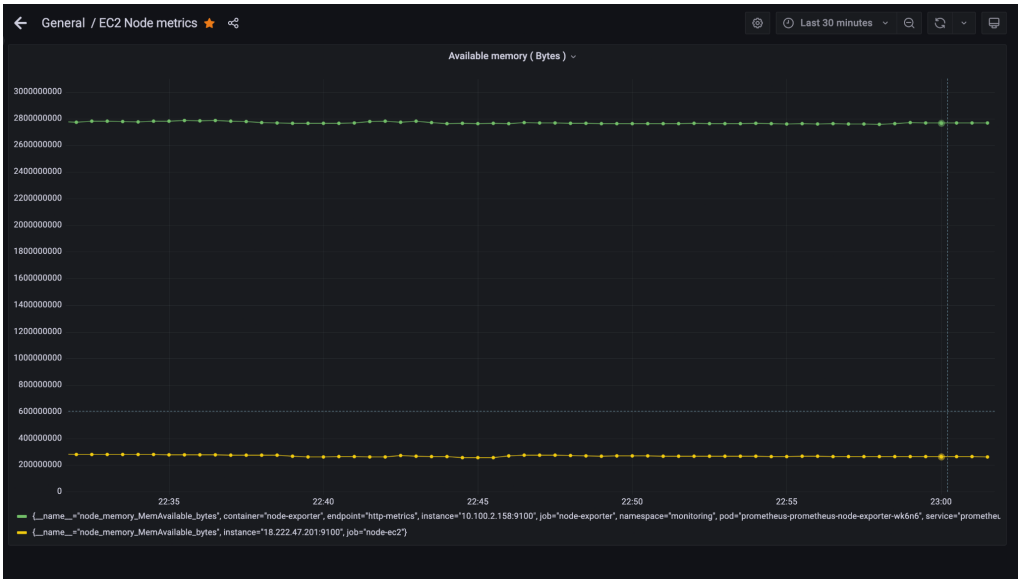
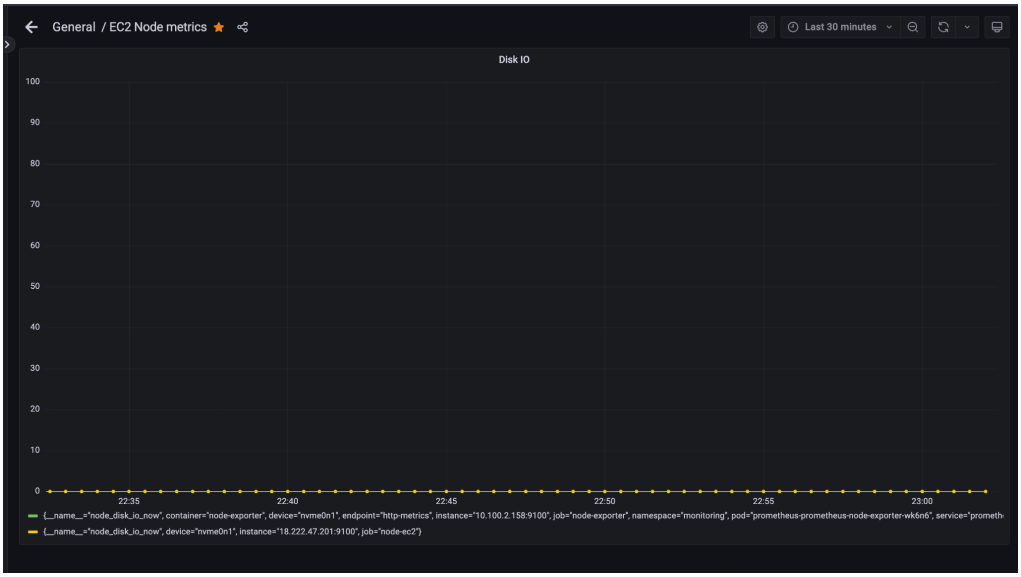
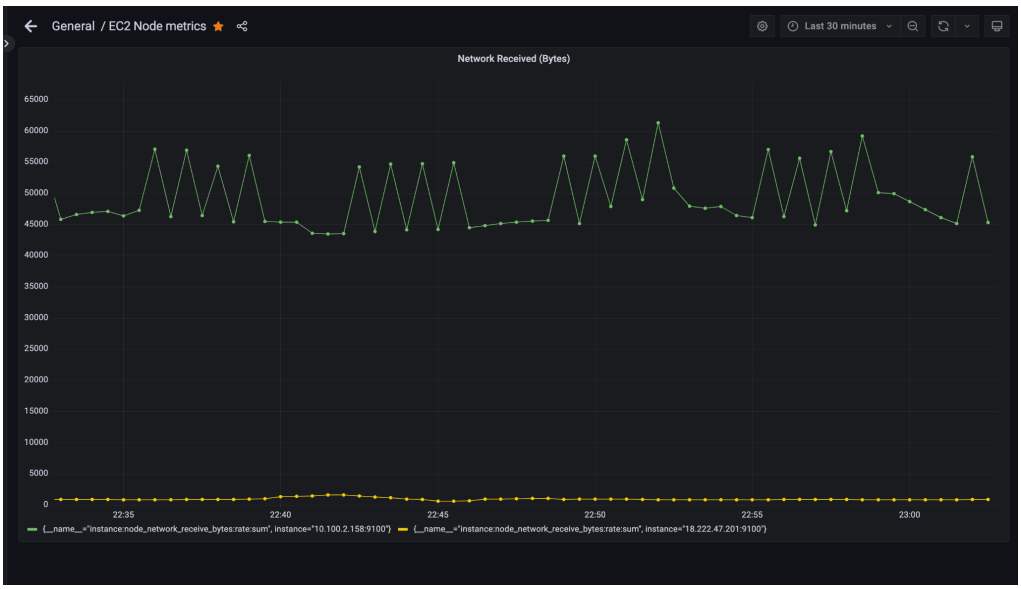
Categorize Responsibilities

Prometheus and Grafana Screenshots
Provide a screenshot of the Prometheus node_exporter service running on the EC2 instance. Use the following command to show that the system is running: `sudo systemctl status node_exporter`

```
ubuntu@ip-172-31-43-155:~$ sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-12-06 16:37:19 UTC; 50s ago
 Main PID: 4625 (node_exporter)
    Tasks: 4 (limit: 1109)
   CGroup: /system.slice/node_exporter.service
           └─4625 /usr/local/bin/node_exporter

Dec 06 16:37:19 ip-172-31-43-155 node_exporter[4625]: level=info ts=2022-12-06T16:37:19.686Z caller=node_exporter.go:115 collector=thermal_zone
Dec 06 16:37:19 ip-172-31-43-155 node_exporter[4625]: level=info ts=2022-12-06T16:37:19.686Z caller=node_exporter.go:115 collector=time
Dec 06 16:37:19 ip-172-31-43-155 node_exporter[4625]: level=info ts=2022-12-06T16:37:19.686Z caller=node_exporter.go:115 collector=timex
Dec 06 16:37:19 ip-172-31-43-155 node_exporter[4625]: level=info ts=2022-12-06T16:37:19.686Z caller=node_exporter.go:115 collector=udp_queues
Dec 06 16:37:19 ip-172-31-43-155 node_exporter[4625]: level=info ts=2022-12-06T16:37:19.686Z caller=node_exporter.go:115 collector=uname
Dec 06 16:37:19 ip-172-31-43-155 node_exporter[4625]: level=info ts=2022-12-06T16:37:19.686Z caller=node_exporter.go:115 collector=vmstat
Dec 06 16:37:19 ip-172-31-43-155 node_exporter[4625]: level=info ts=2022-12-06T16:37:19.686Z caller=node_exporter.go:115 collector=xfs
Dec 06 16:37:19 ip-172-31-43-155 node_exporter[4625]: level=info ts=2022-12-06T16:37:19.686Z caller=node_exporter.go:115 collector=zfs
Dec 06 16:37:19 ip-172-31-43-155 node_exporter[4625]: level=info ts=2022-12-06T16:37:19.687Z caller=node_exporter.go:199 msg="Listening on" address=:9100
Dec 06 16:37:19 ip-172-31-43-155 node_exporter[4625]: level=info ts=2022-12-06T16:37:19.690Z caller=tls_config.go:191 msg="TLS is disabled." http2=false
ubuntu@ip-172-31-43-155:~$
```

Host Metric (CPU, RAM, Disk, Network)	Dashboard
CPU	

RAM	
DISK	
Network	

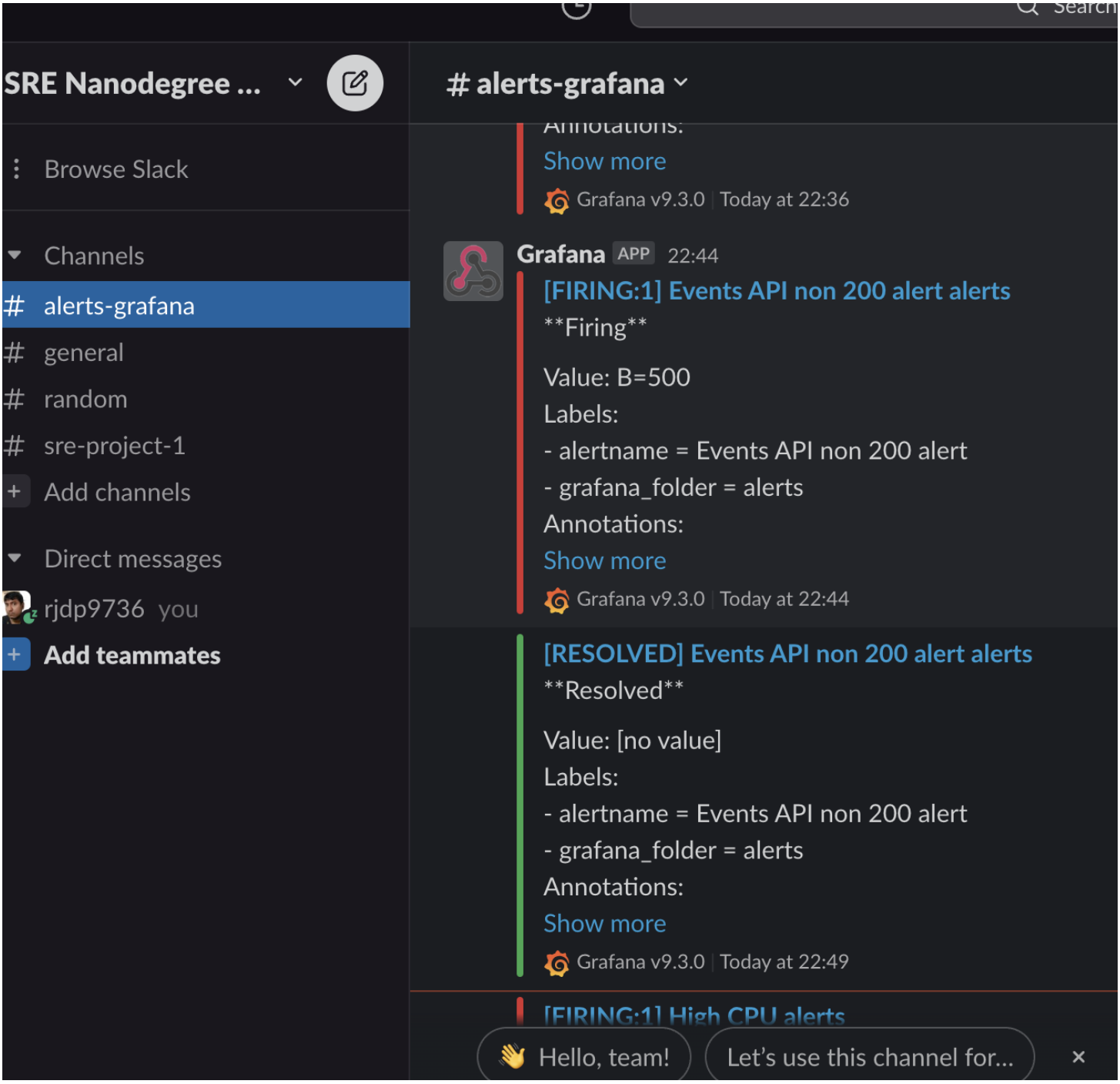
Responsibilities
1. The development team wants to release an emergency hotfix to production. Identify two roles of the SRE team who would be involved in this and why.
System Architect: to make sure the changes done for hotfix have the right trade-offs and not just some solution that would solve the current problem but create some backward incompatibility or tech debt related issues **Release Engineer:** Release the build and incase something goes wrong do a rollback
2. The development team is in the early stages of planning to build a new product. Identify two roles of the SRE team that should be invited to the meeting and why.
Team lead: Oversees the breakdown and allocation of tasks to right members and sets up the scope of task and identifies non-goals **System Architect:** Makes sure upfront cost is put into parts of design that will make the product thrive both technical and business requirements wise. Suggests tech stack for various parts of software like backend languages/libraries/framework, databases , queueing technologies so by going through pros/cons and discussing why a particular choice aligns with what they have set out to achieve
3. The emergency hotfix from question 1 was applied and is causing major issues in production. Which SRE role would primarily be involved in mitigating these issues?
Release Engineer : given hotfix has caused an issue quickest fix would be to go back to the previous stable version of the application, Release engineer will rollback to a stable build

Team Formation and Workflow Identification

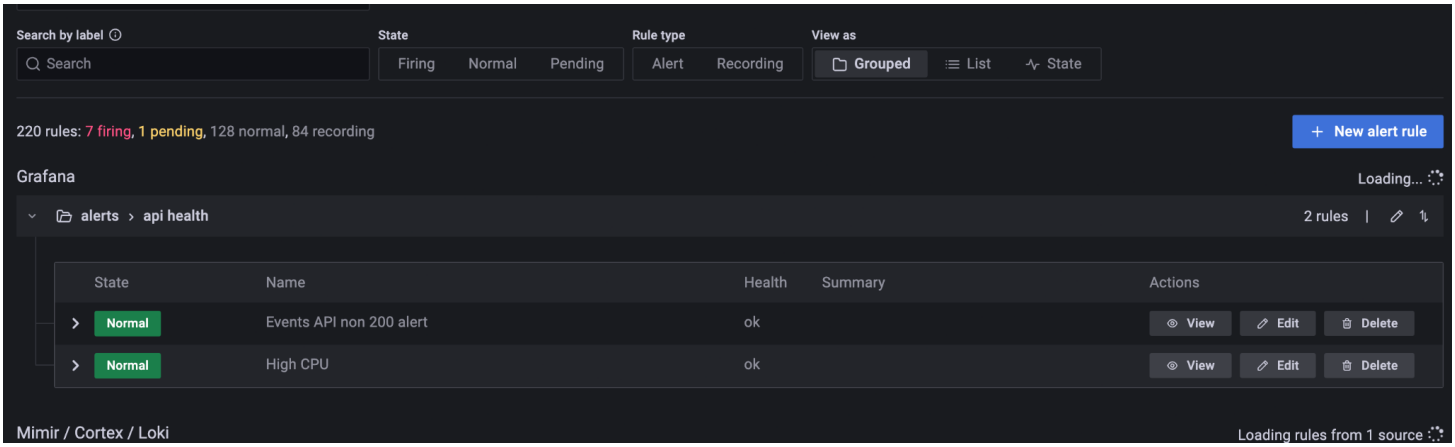
API Monitoring and Notifications

Display the status of an API endpoint: Provide a screenshot of the Grafana dashboard that will show at which point the API is unhealthy (non-200 HTTP code), and when it becomes healthy again (200 HTTP code).

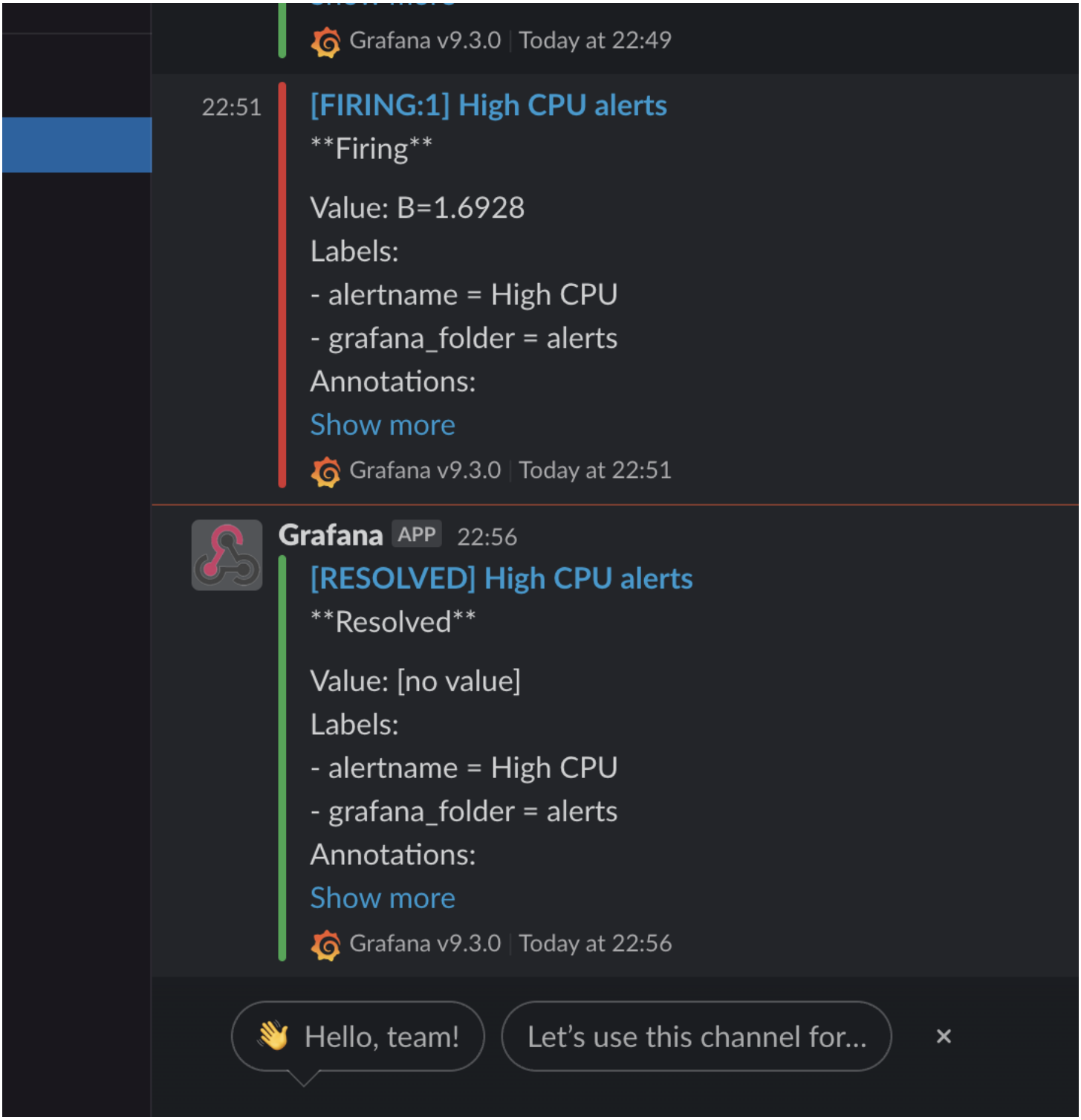
Create a notification channel: Provide a screenshot of the Grafana notification which shows the summary of the issue and when it occurred.



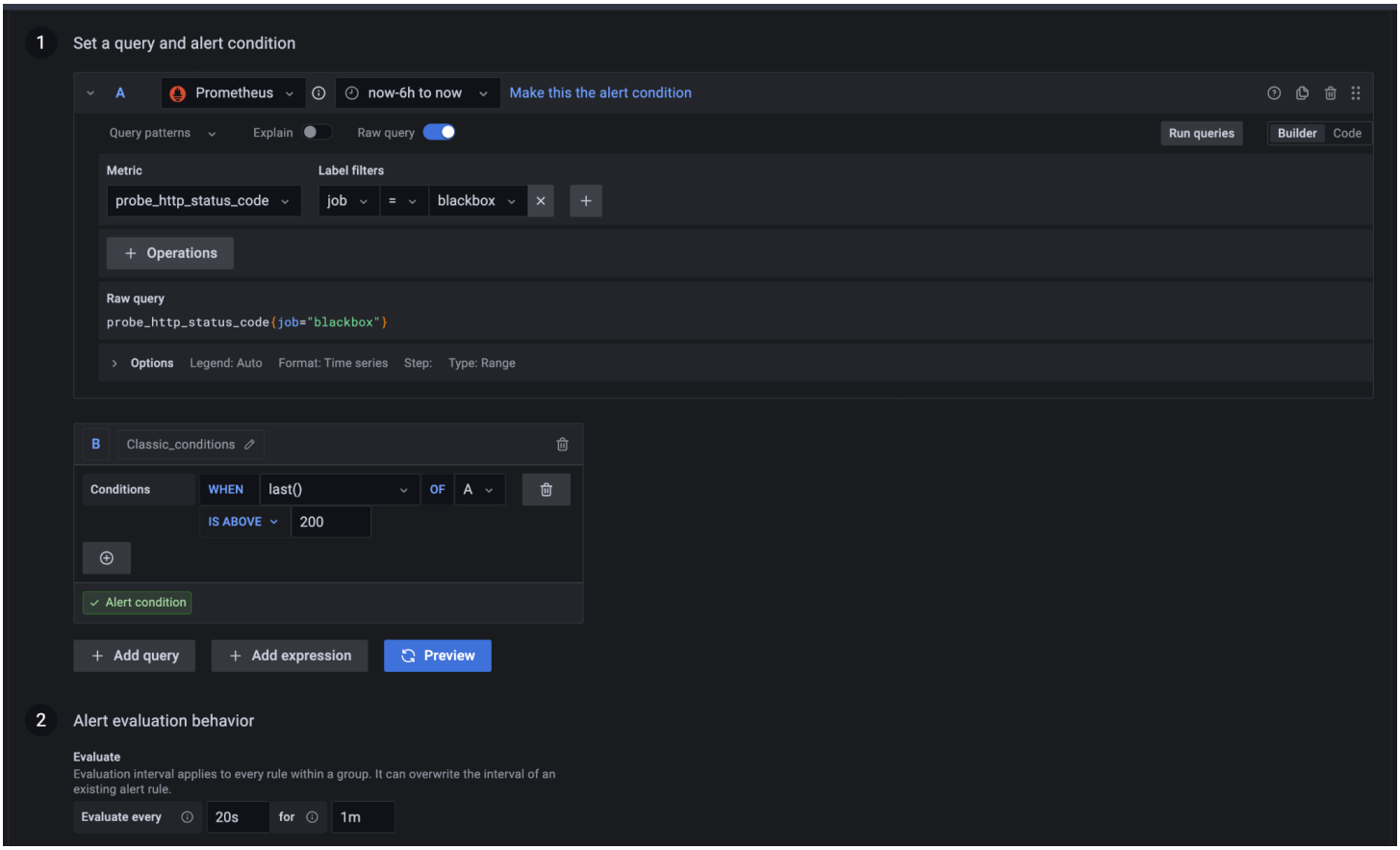
Configure alert rules: Provide a screenshot of the alert rules list in Grafana.



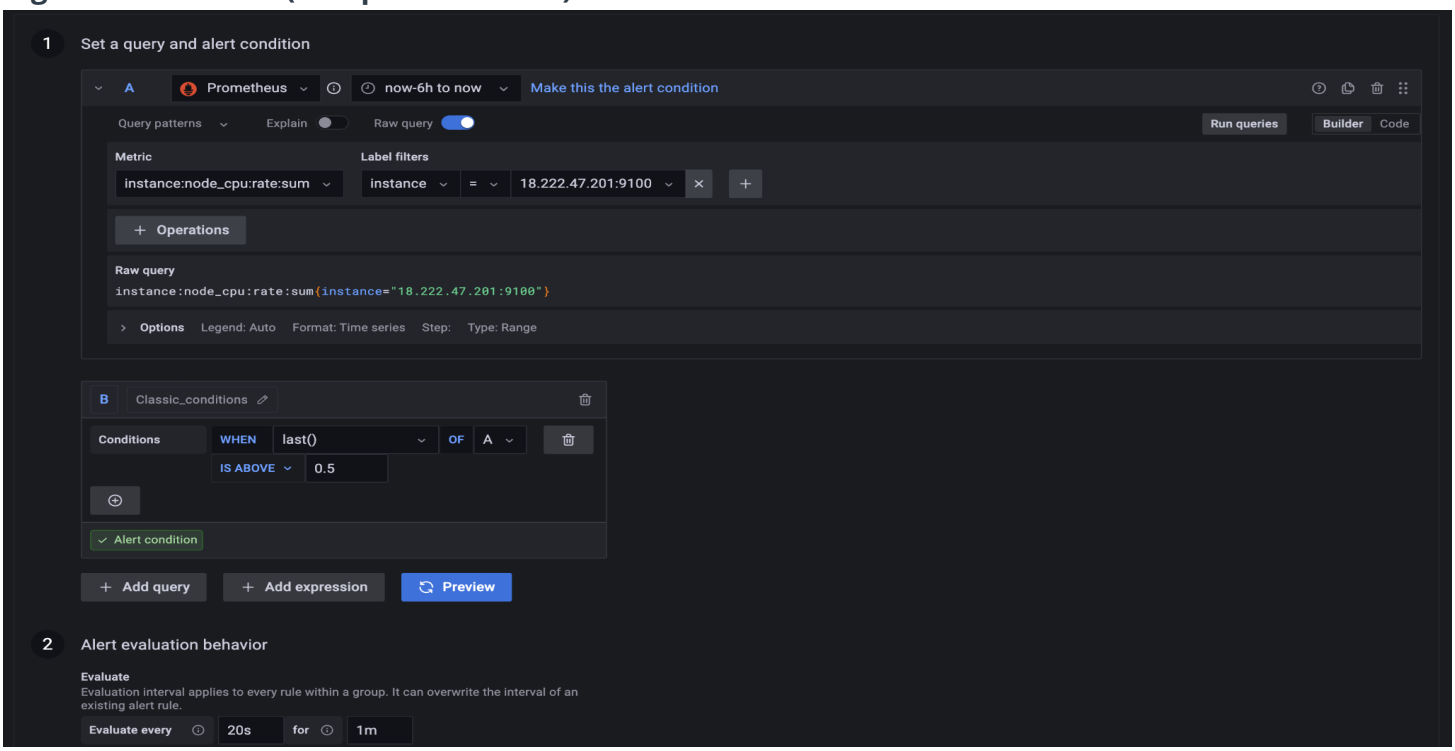
ALERT notification for High CPU



API health alert rule (for non 200 status codes)

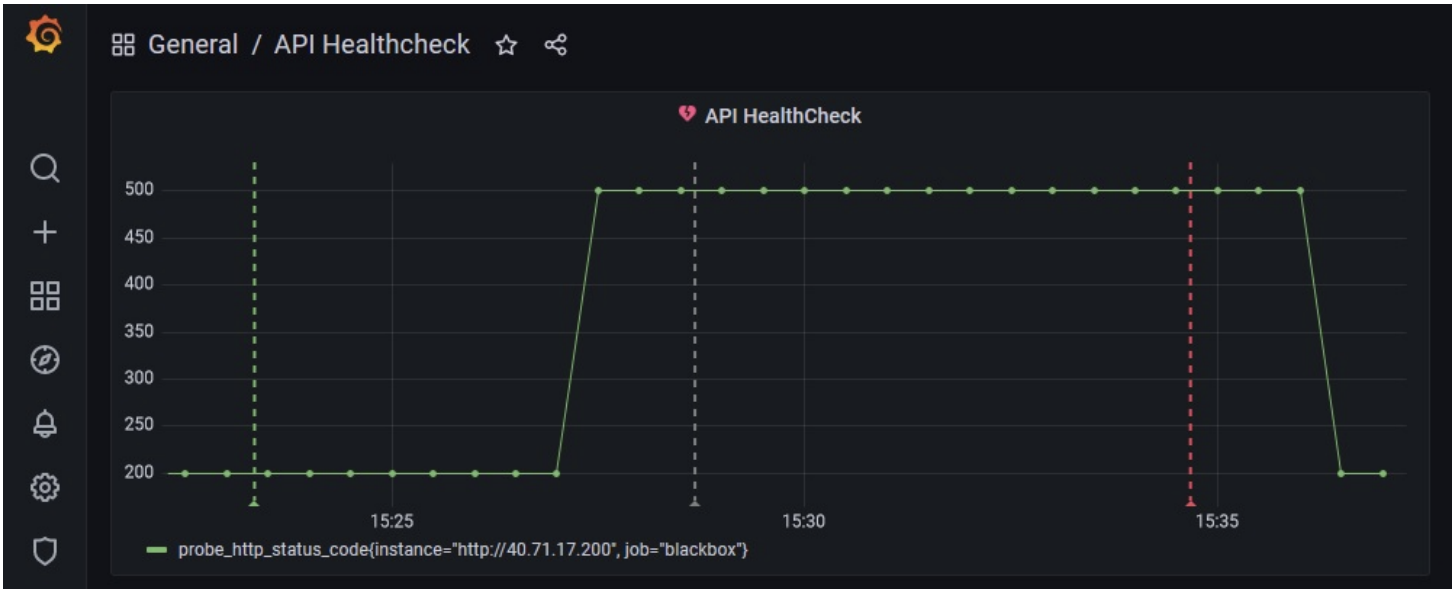


High CPU alert rule (for cpu above 50%)



Applying the Concepts

Graph 1



4a. Given the above graph, where does it show that the API endpoint is down? Where on the graph does this show that the API is healthy again?

At mid point of 15:25 and 15:30 the endpoint started responding with 500 status code and at around 15:36 it recovered as we start observing 200 status code at this point of time

4b. If there was no SRE team, how would this outage affect customers?

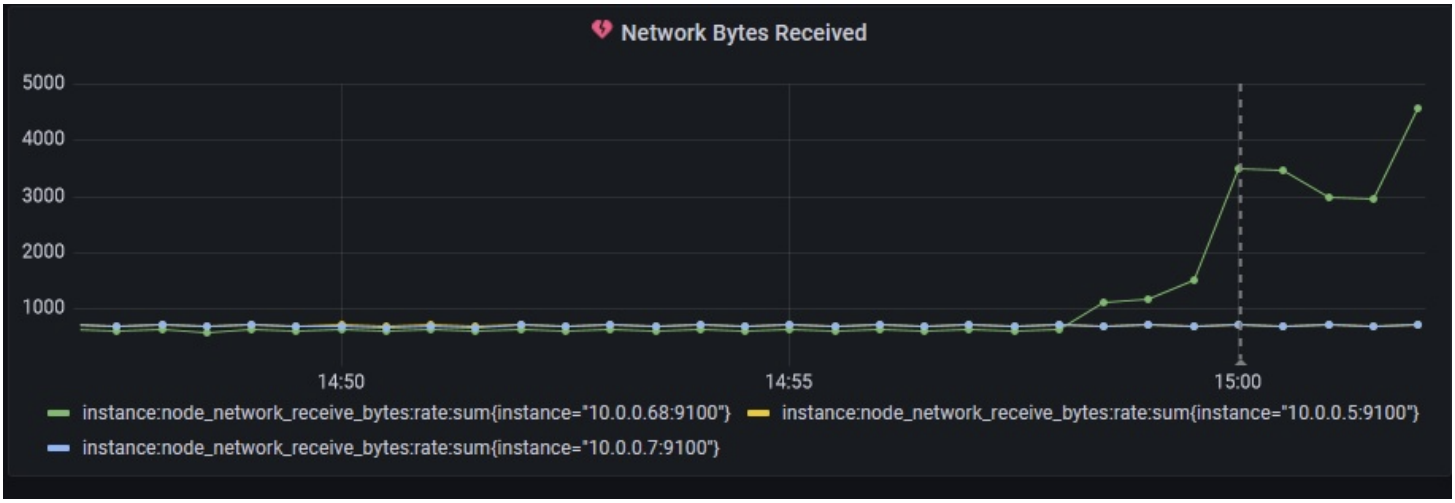
If there was no SRE team

- this might have gone unnoticed and the duration of outage must have increased multifold until some customer takes initiative to report this outage
- even when it gets resolved later there would be no incident management thus it would be more likely for this issue to reoccur

4c. What could be put in place so that the SRE team could know of the outage before the customer does?

- Synthetic monitoring to monitor the endpoints using blackbox and setup alert on non 200 status codes
- for example an alert that is evaluated every 30 seconds for 2 minutes and if this is satisfied alert is set to right group slack or a email group

Graph 2



5a. Given the above graph, which instance had the increase in traffic, and approximately how many bytes did it receive (feel free to round)?

Instance with IP "**10.0.0.68**" saw increase in traffic, given the metric for this graph is "`instance:node_network_receive_bytes:rate:sum`" the approx bytes received since it saw a increase would be sum of all values of points after the increase point i.e 1100 + 1100 + 1500 + 3500 + 3500 + 3000 + 3000 + 4500 = **21200** bytes

5b. Which team members on the SRE team would be interested in this graph and why?

Monitoring engineer : can see this graph and find if this has correlation with latency issue in the services deployed and thus setup appropriate alert rules and send them to right stakeholders
System Architect: the graph would influence the design decisions of architect when she is architecting an application and it would help her make tradeoffs on his design decisions
Infrastructure Engineer: As he has 50% development tasks he can probably pinpoint to actual code which might be causing a spike and according take optimisation decision such as to restrict files greater than certain threshold size from user or even rate limit clients requests