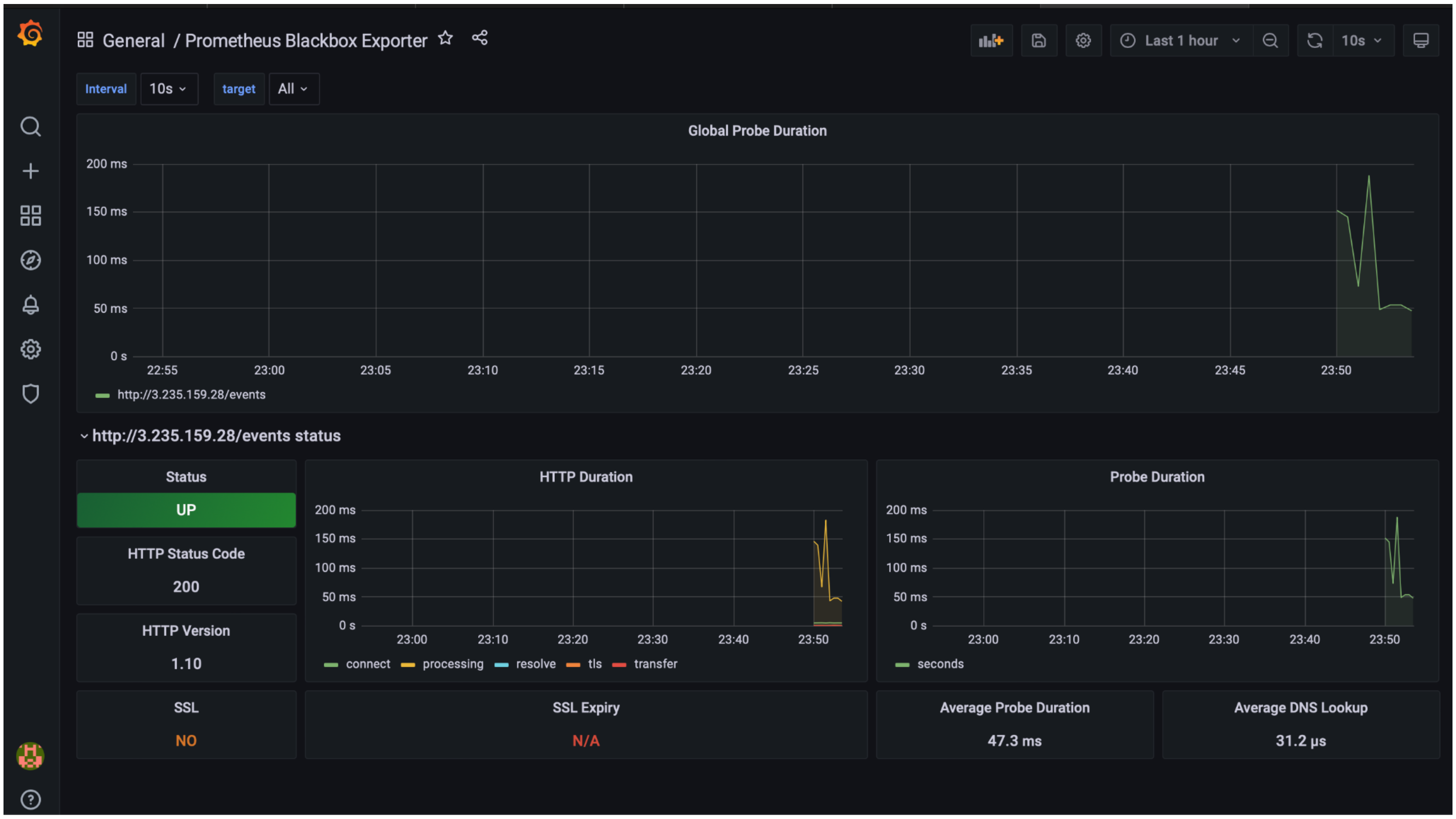
Observing Cloud Resources

SRE Project Template

Categorize Responsibilities

Prometheus and Grafana Screenshots

Screenshot Blackbox exporter imported



General / Prometheus Blackbox Exporter

Last 1 hour · 10s

Interval 10s · target All

http://3.235.159.28/events status

Status: UP

HTTP Status Code: 200

HTTP Version: 1.10

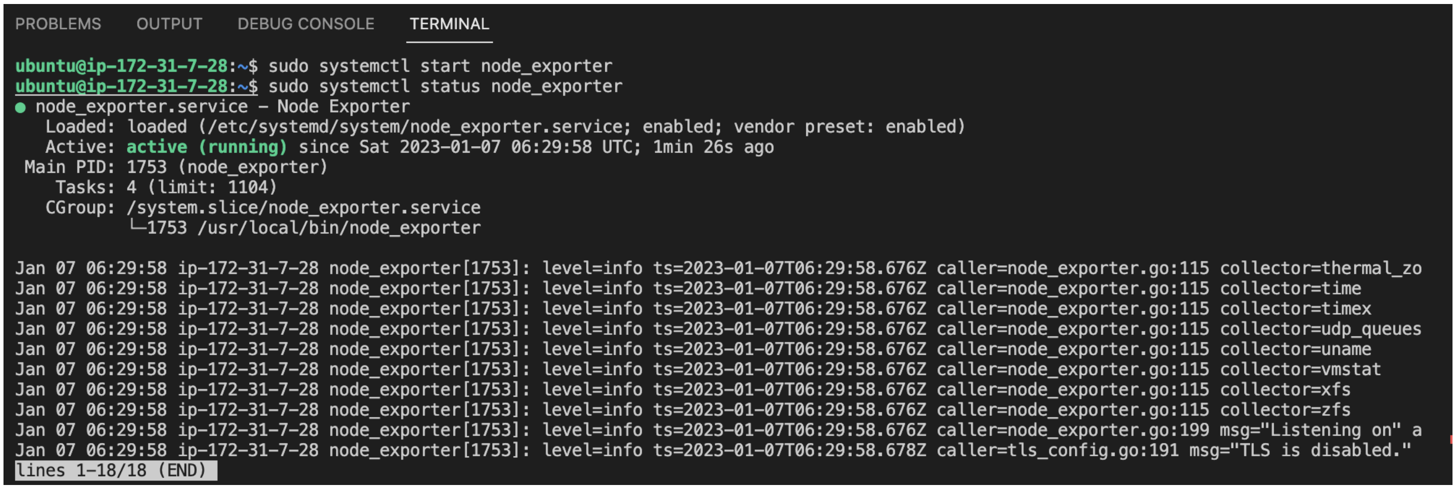
SSL: NO

SSL Expiry: N/A

Average Probe Duration: 47.3 ms

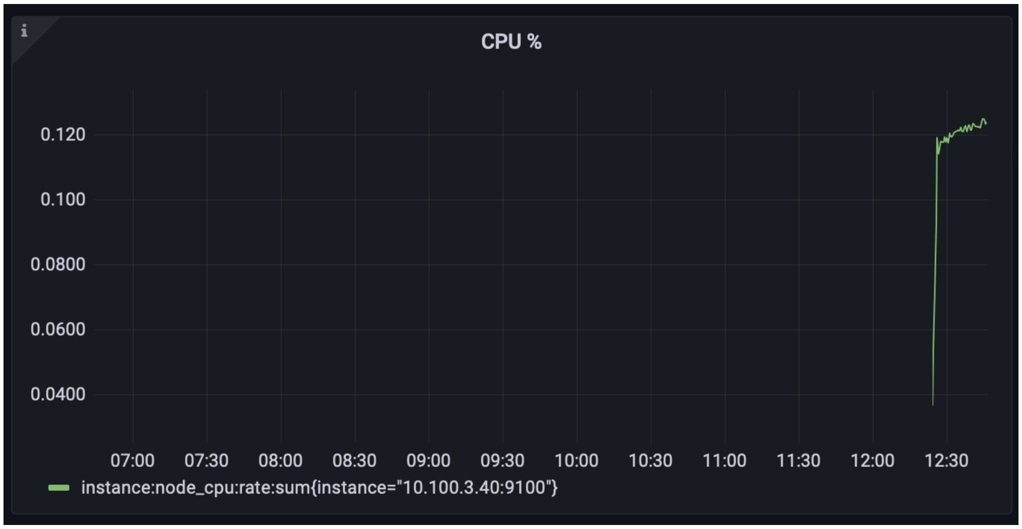
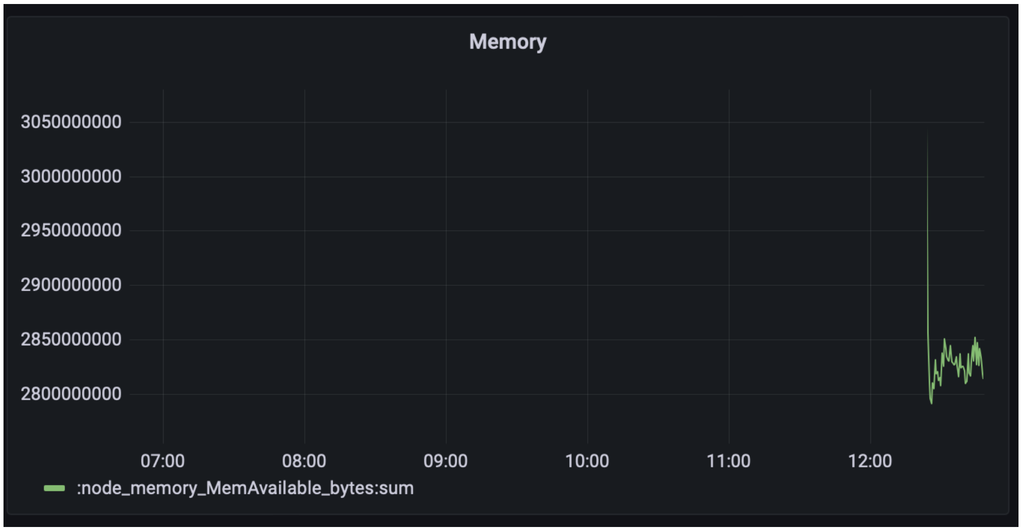
Average DNS Lookup: 31.2 µs

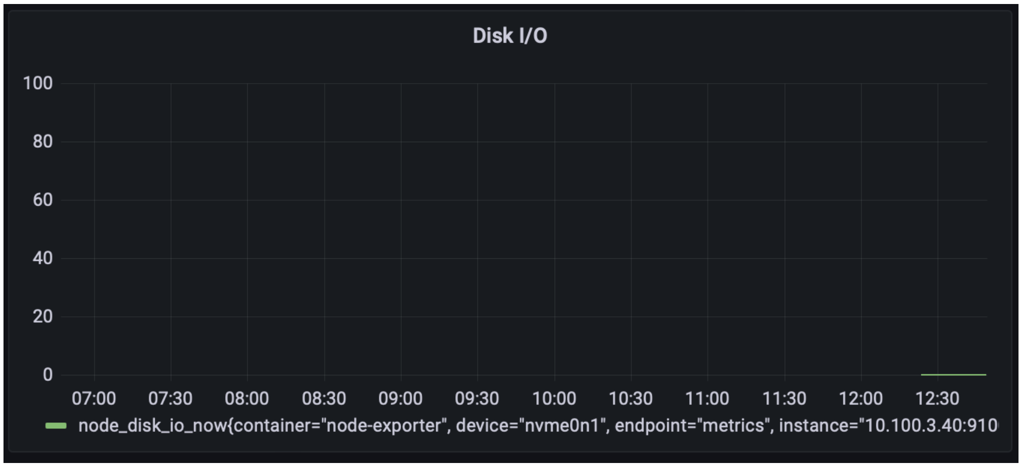
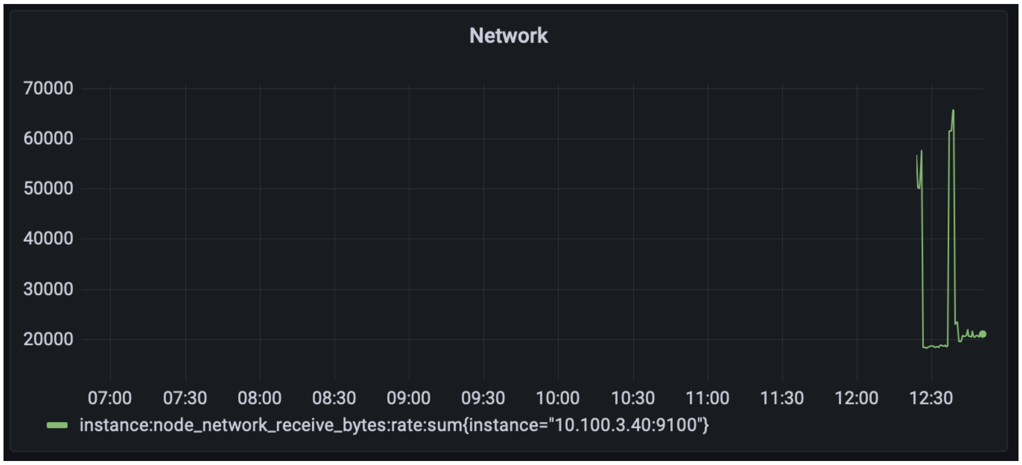
Provide a screenshot of the Prometheus node_exporter service running on the EC2 instance. Use the following command to show that the system is running: `sudo systemctl status node_exporter`



```
ubuntu@ip-172-31-7-28:~$ sudo systemctl start node_exporter
ubuntu@ip-172-31-7-28:~$ 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 Sat 2023-01-07 06:29:58 UTC; 1min 26s ago
 Main PID: 1753 (node_exporter)
    Tasks: 4 (limit: 1104)
   CGroup: /system.slice/node_exporter.service
           └─1753 /usr/local/bin/node_exporter

Jan 07 06:29:58 ip-172-31-7-28 node_exporter[1753]: level=info ts=2023-01-07T06:29:58.676Z caller=node_exporter.go:115 collector=thermal_zo
Jan 07 06:29:58 ip-172-31-7-28 node_exporter[1753]: level=info ts=2023-01-07T06:29:58.676Z caller=node_exporter.go:115 collector=time
Jan 07 06:29:58 ip-172-31-7-28 node_exporter[1753]: level=info ts=2023-01-07T06:29:58.676Z caller=node_exporter.go:115 collector=timex
Jan 07 06:29:58 ip-172-31-7-28 node_exporter[1753]: level=info ts=2023-01-07T06:29:58.676Z caller=node_exporter.go:115 collector=udp_queues
Jan 07 06:29:58 ip-172-31-7-28 node_exporter[1753]: level=info ts=2023-01-07T06:29:58.676Z caller=node_exporter.go:115 collector=uname
Jan 07 06:29:58 ip-172-31-7-28 node_exporter[1753]: level=info ts=2023-01-07T06:29:58.676Z caller=node_exporter.go:115 collector=vmstat
Jan 07 06:29:58 ip-172-31-7-28 node_exporter[1753]: level=info ts=2023-01-07T06:29:58.676Z caller=node_exporter.go:115 collector=xfs
Jan 07 06:29:58 ip-172-31-7-28 node_exporter[1753]: level=info ts=2023-01-07T06:29:58.676Z caller=node_exporter.go:115 collector=zfs
Jan 07 06:29:58 ip-172-31-7-28 node_exporter[1753]: level=info ts=2023-01-07T06:29:58.676Z caller=node_exporter.go:199 msg="Listening on" a
Jan 07 06:29:58 ip-172-31-7-28 node_exporter[1753]: level=info ts=2023-01-07T06:29:58.678Z caller=tls_config.go:191 msg="TLS is disabled."
lines 1-18/18 (END)
```

Host Metric **(CPU, RAM, Disk, Network)**	**Dashboard**
instance:node_cpu:rate:sum	
node_memory_MemAvailable_bytes	

node_disk_io_now	
instance:node_network_receive_bytes:rate:sum	

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.

The Release manager will be one person from the SRE team getting involved as he/she is responsible for code releases.
The other person which will assist in this case can be the Monitoring Engineer as he/she is the first one to know if this code release caused any errors/incidents.

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.

System Architect should definitely be one of the two roles from the SRE team to participate in such a meeting since he/she can contribute to creating scalable, reliable architecture for the product.
The other person can be the Release Engineer as he/she can already plan releases.

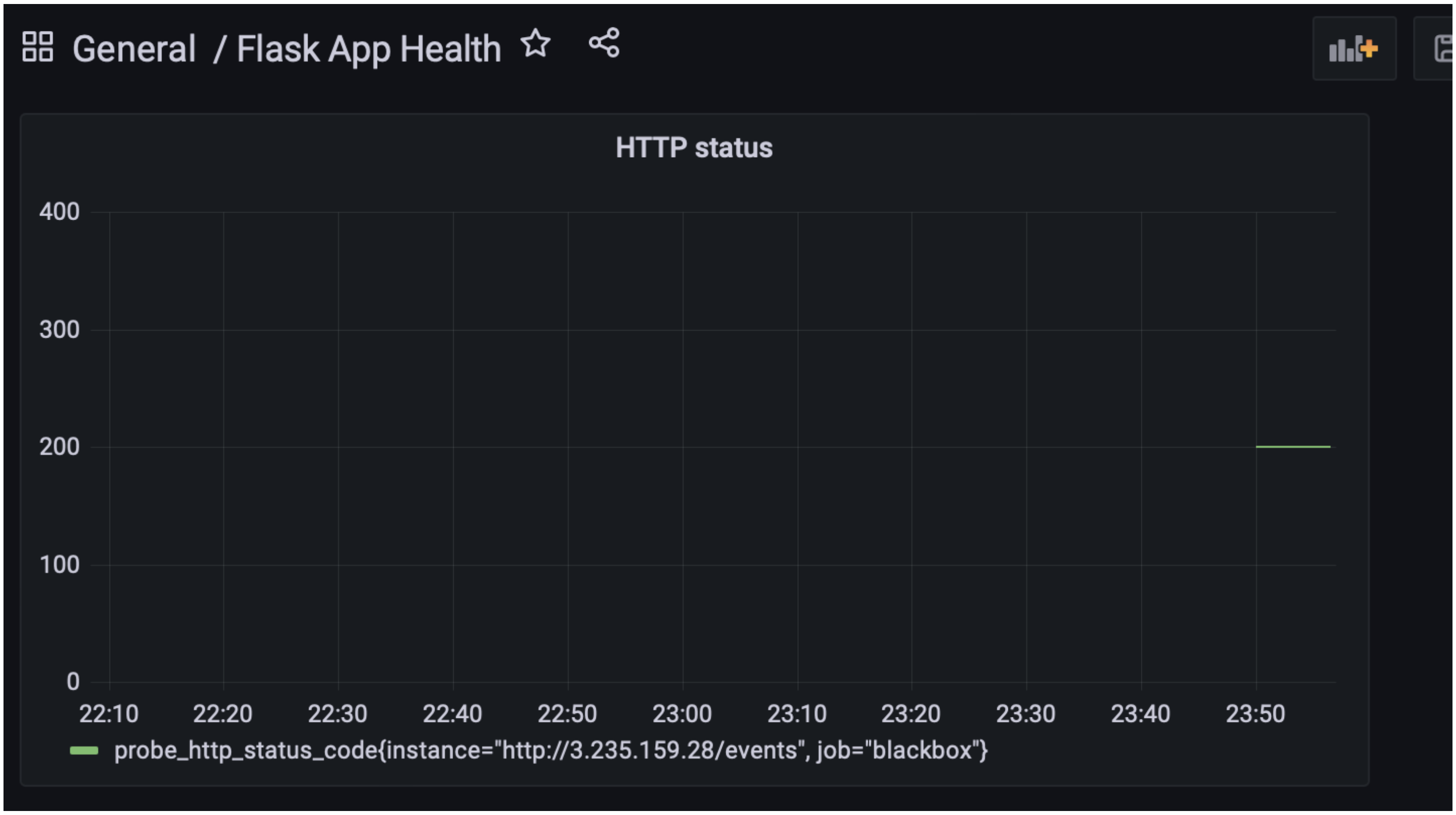
3. The emergency hotfix from question 1 was applied and is causing major issues in production. Which SRE role would primarily be involved in mitigating these issues?

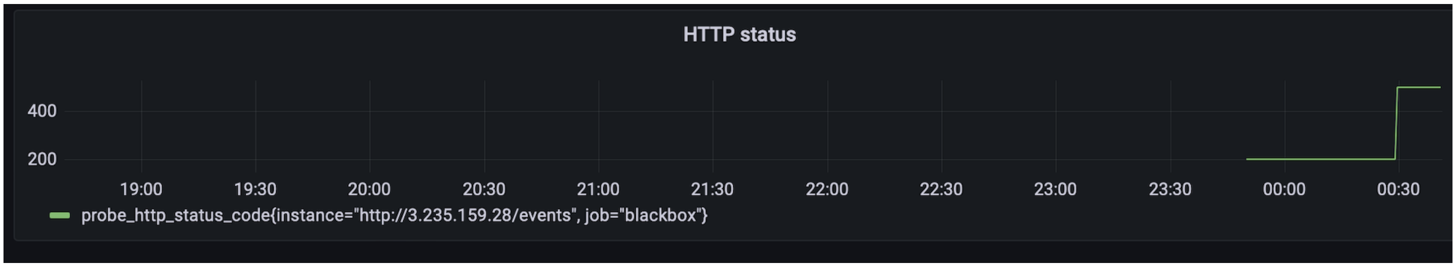
The Release Engineer will be the one involved as he/she is responsible for the releases and will be the one starting the rollback process.

Team Formation and Workflow Identification

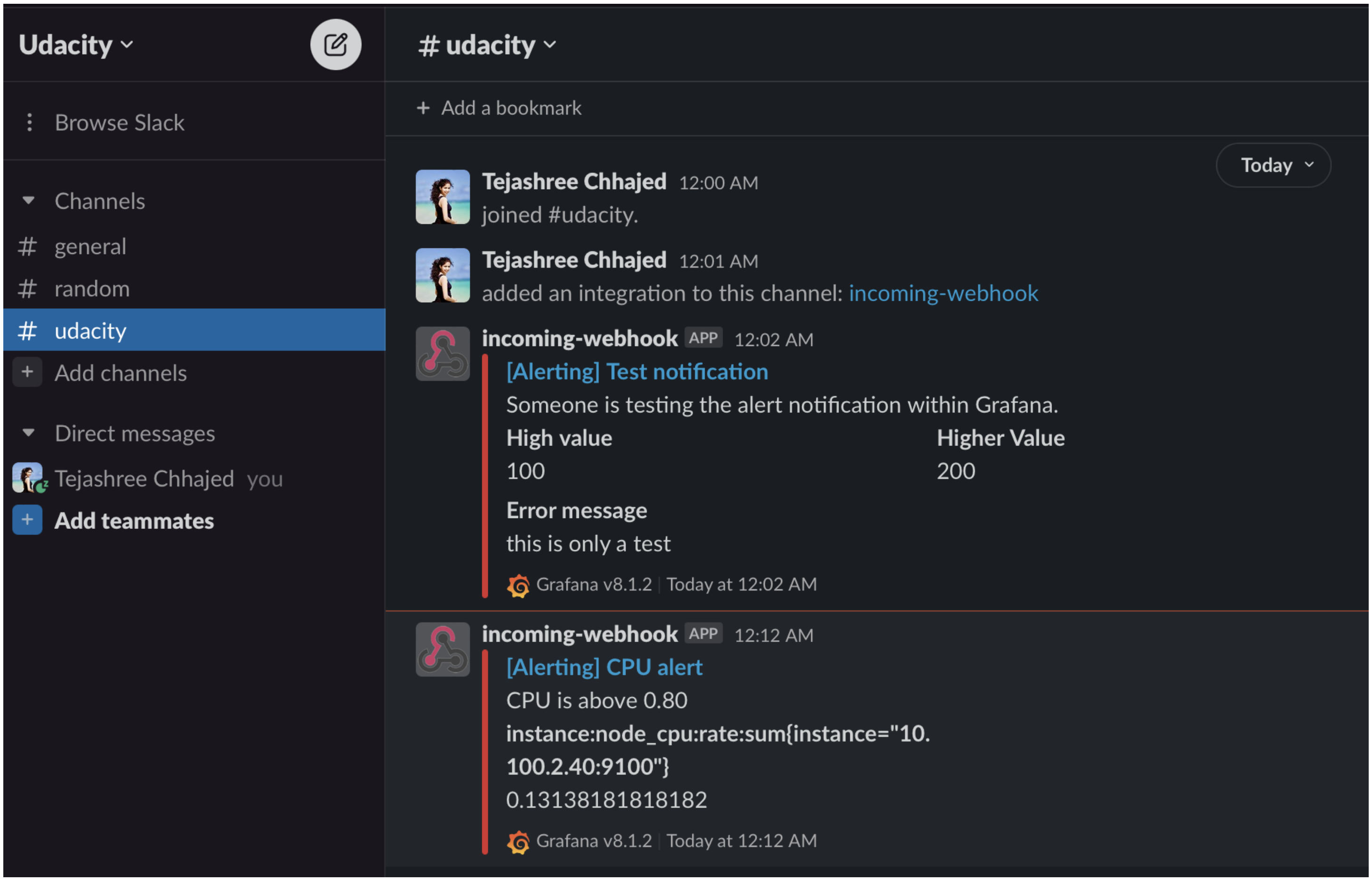
API Monitoring and Notifications

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.



Tejashree Chhajed 12:00 AM
joined #udacity.

Tejashree Chhajed 12:01 AM
added an integration to this channel: incoming-webhook

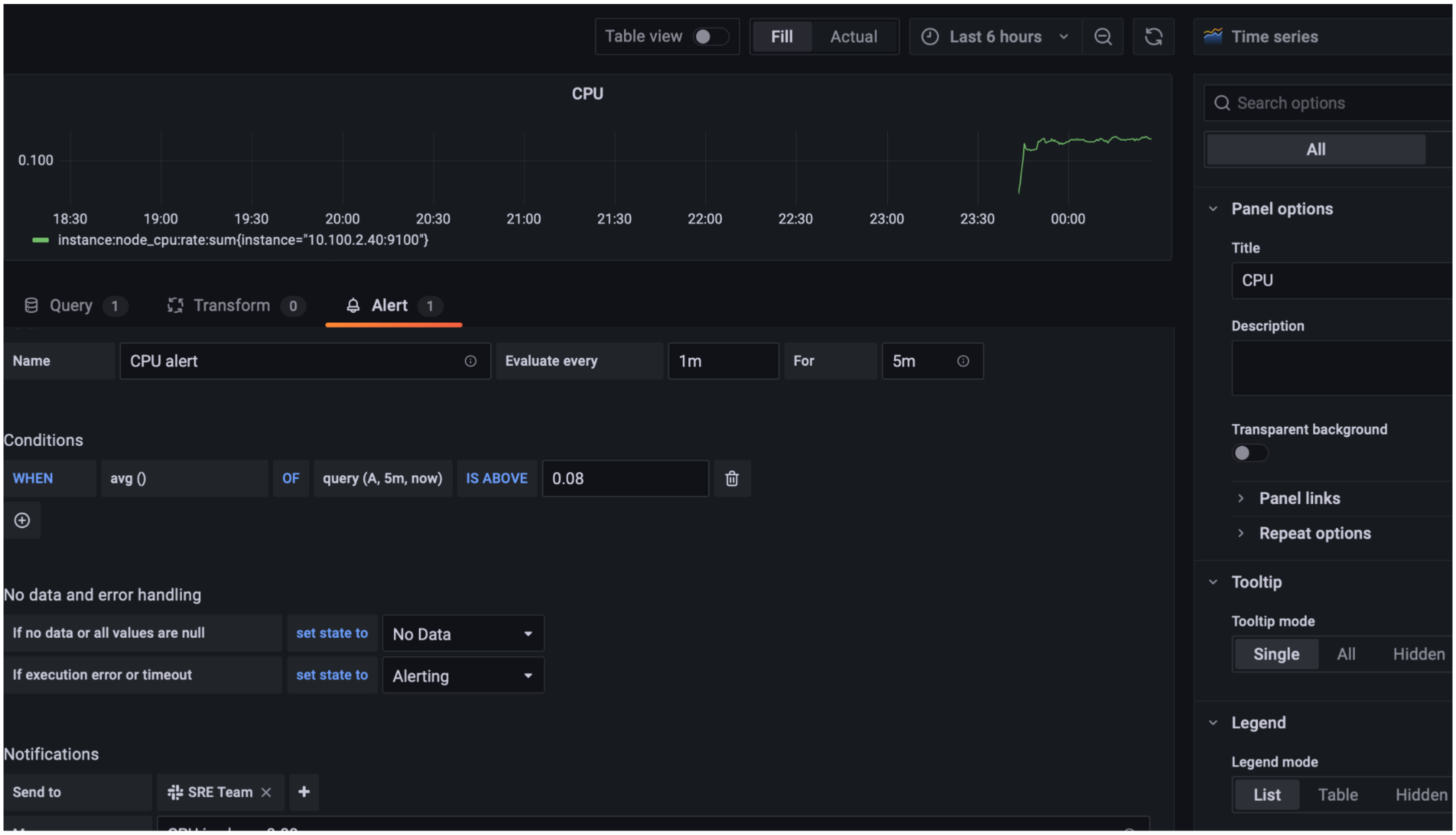
incoming-webhook APP 12:02 AM
[Alerting] Test notification
Someone is testing the alert notification within Grafana.

High value	Higher Value
100	200

Error message
this is only a test
Grafana v8.1.2 | Today at 12:02 AM

incoming-webhook APP 12:12 AM
[Alerting] CPU alert
CPU is above 0.80
instance:node_cpu:rate:sum{instance="10.100.2.40:9100"}
0.13138181818182
Grafana v8.1.2 | Today at 12:12 AM

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



Name: CPU alert | Evaluate every: 1m | For: 5m

Conditions

WHEN avg () OF query (A, 5m, now) IS ABOVE 0.08

No data and error handling

If no data or all values are null set state to No Data

If execution error or timeout set state to Alerting

Notifications

Send to: SRE Team

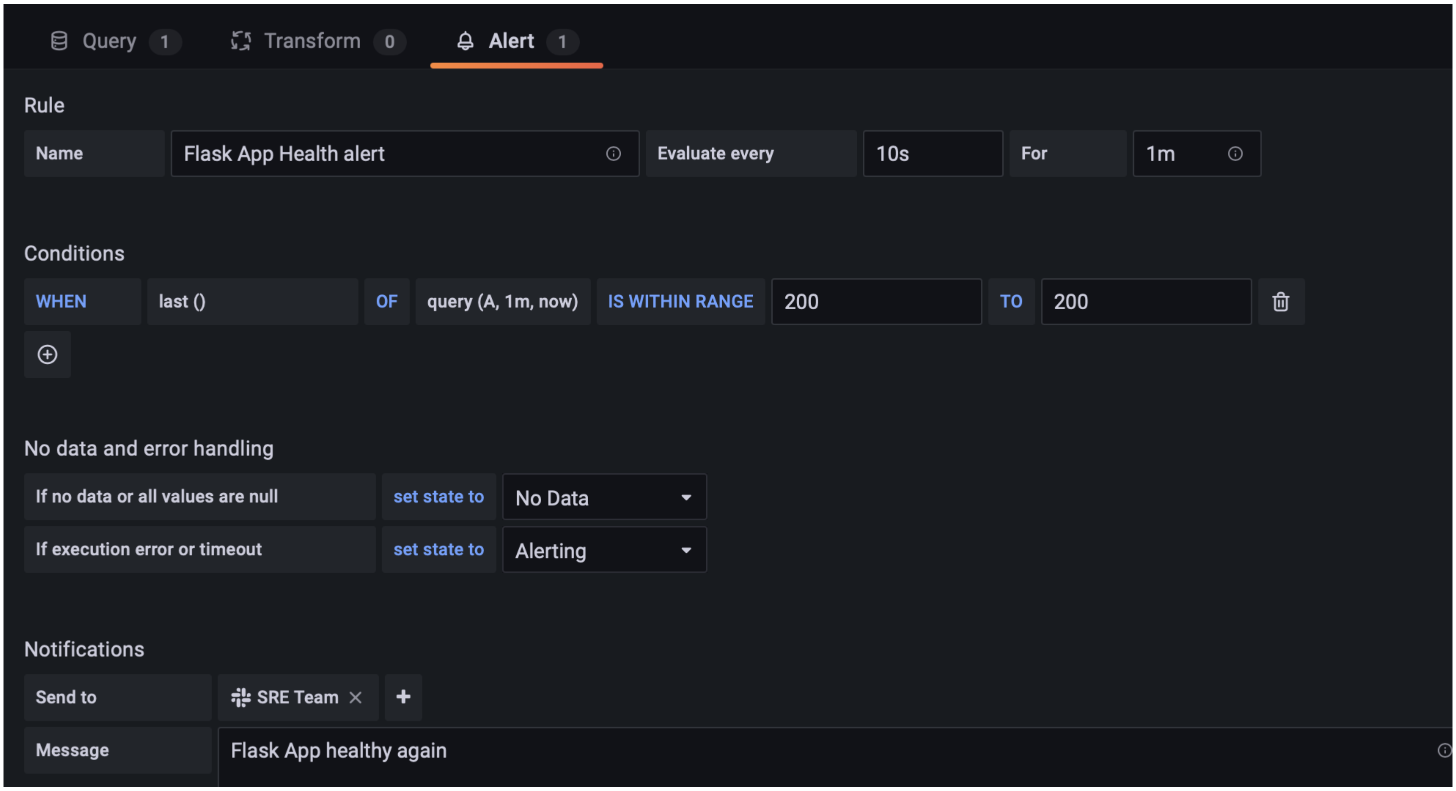
Title: CPU

Description

Transparent background

Tooltip mode

Legend mode



Rule

Name: Flask App Health alert | Evaluate every: 10s | For: 1m

Conditions

WHEN last () OF query (A, 1m, now) IS WITHIN RANGE 200 TO 200

No data and error handling

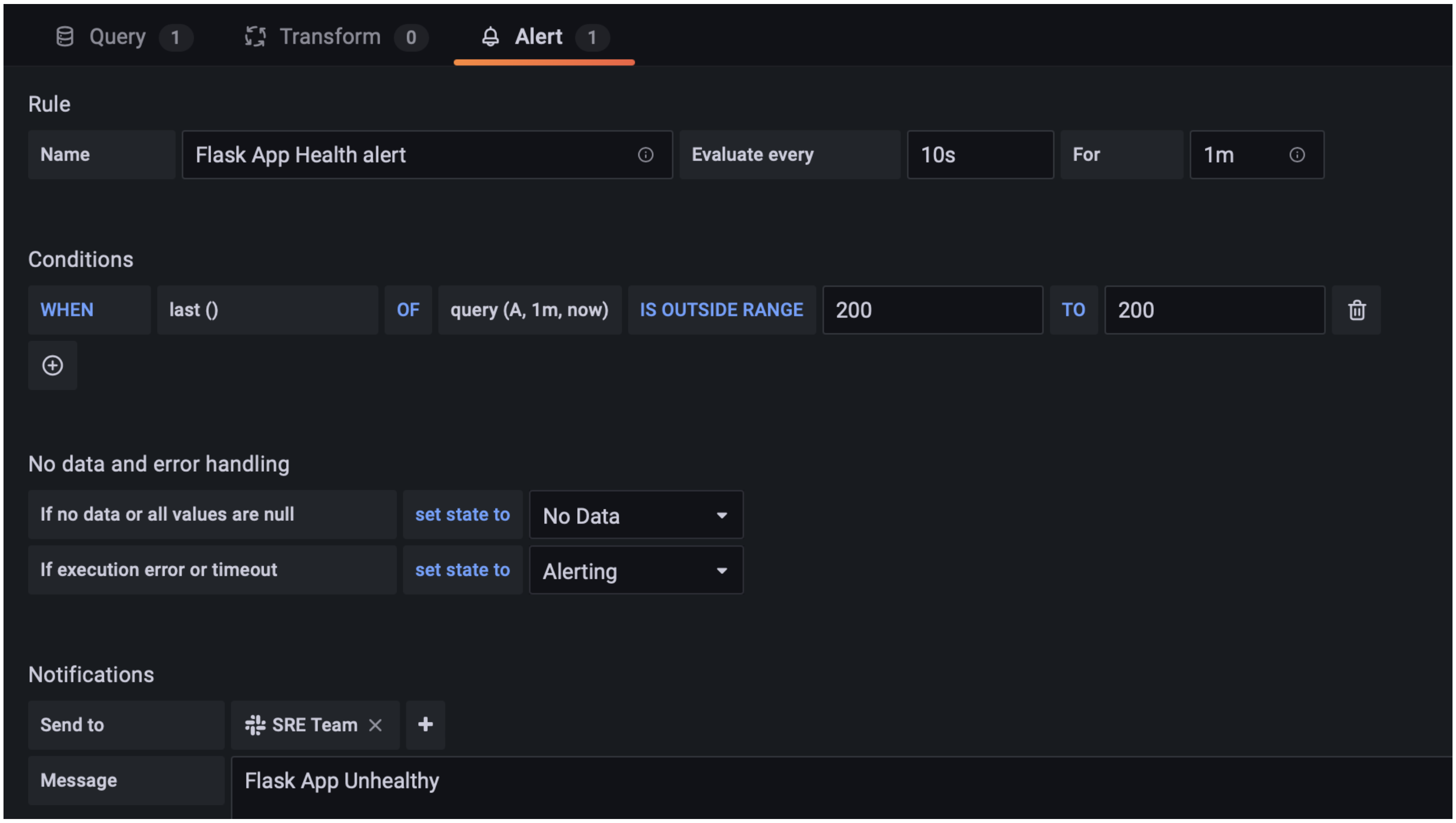
If no data or all values are null set state to No Data

If execution error or timeout set state to Alerting

Notifications

Send to: SRE Team

Message: Flask App healthy again



Rule

Name: Flask App Health alert | Evaluate every: 10s | For: 1m

Conditions

WHEN last () OF query (A, 1m, now) IS OUTSIDE RANGE 200 TO 200

No data and error handling

If no data or all values are null set state to No Data

If execution error or timeout set state to Alerting

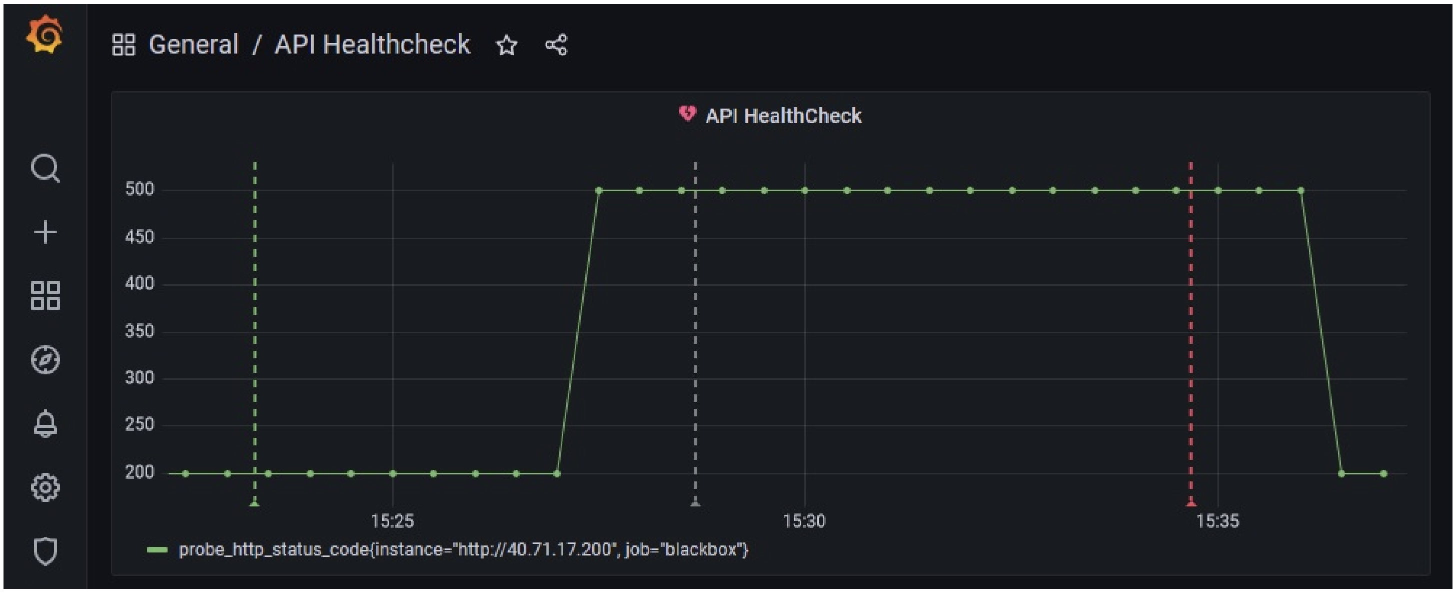
Notifications

Send to: SRE Team

Message: Flask App Unhealthy

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 15.29 the endpoint is down and on 15:35 the endpoint is healthy again

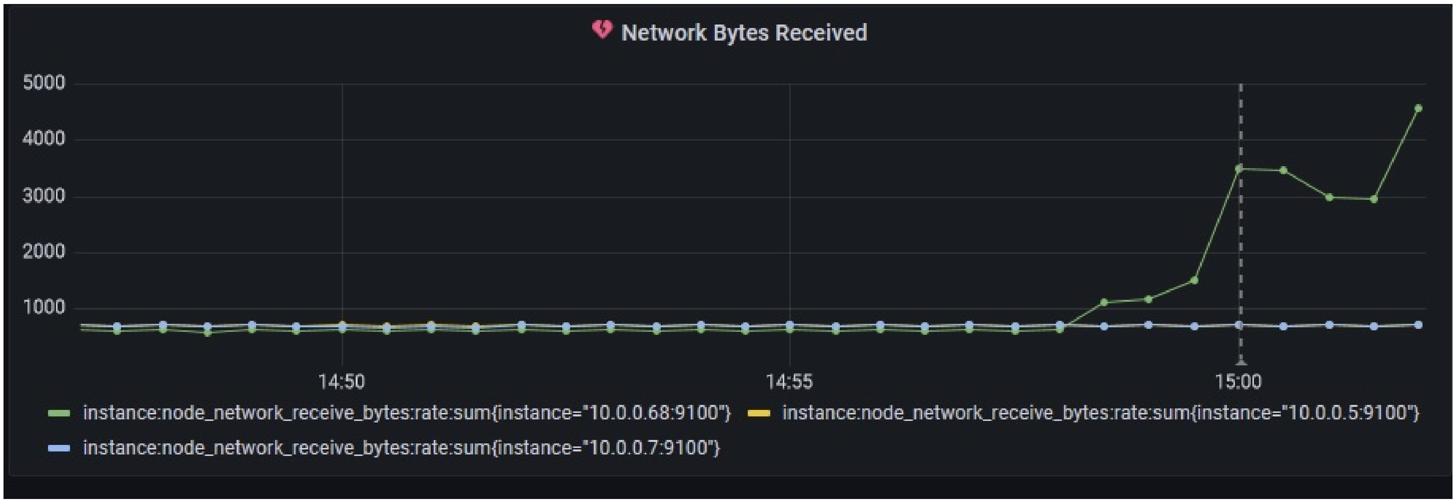
4b. If there was no SRE team, how would this outage affect customers?

If there was no SRE team the information about this outage would be known only when customers complain. The customers would be frustrated about the fact that the outage was unknown to the company and that no one is taking care of fixing it hence the time it might require to come online again is unknown.

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

Monitoring needs to be set up with a synthetic monitoring solution so that the SRE team knows about the outage before the customer does.

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 is 10.0.0.68:9100 is the one which had increased traffic. It received more than 4000 bytes.

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

Monitoring Engineer to observe any imminent failures due to unexpected traffic.
Release Engineer to observe if any recent release caused this increased traffic.