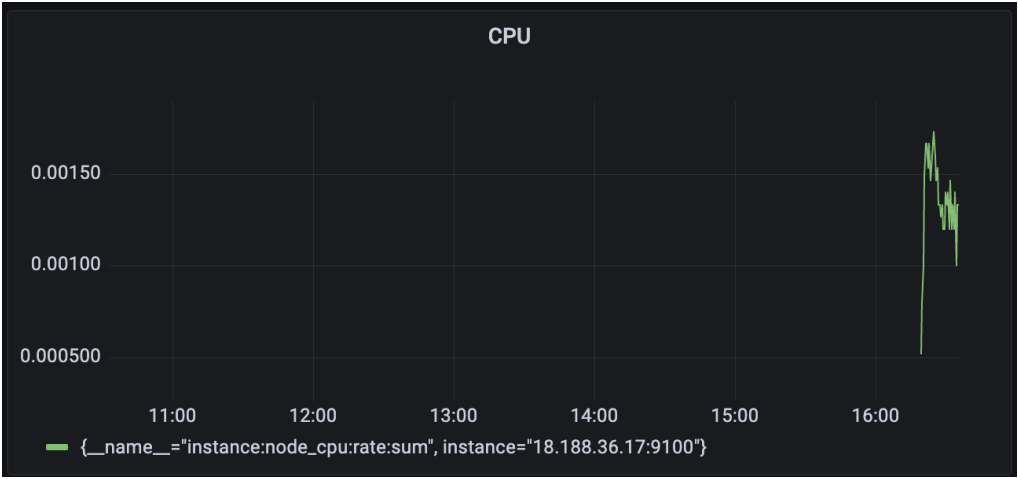


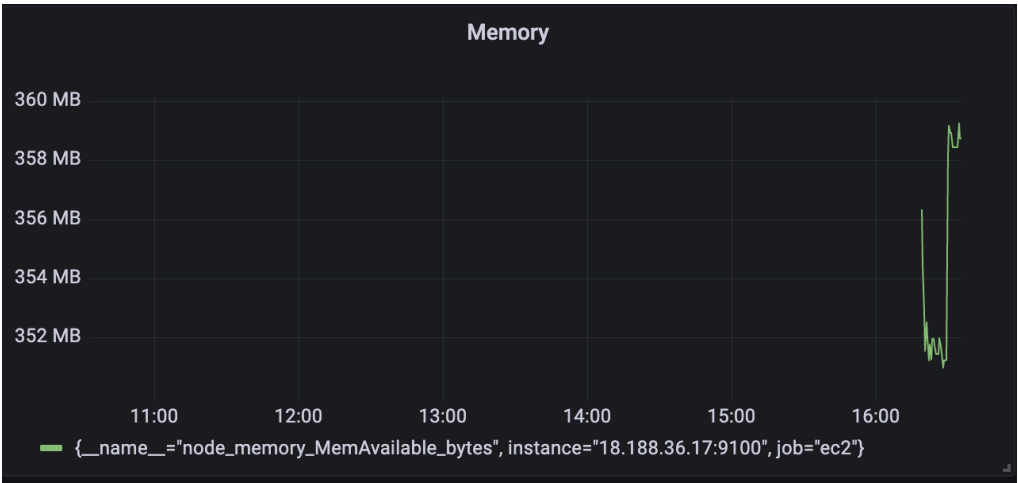
# Observing Cloud Resources

SRE Project Template

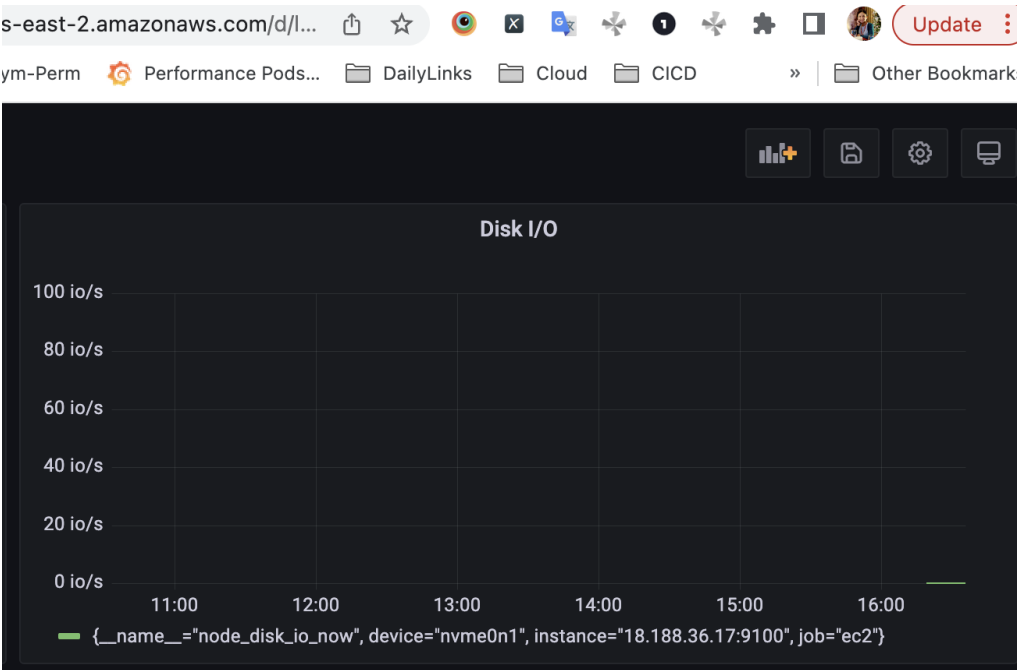
## Categorize Responsibilities

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: <code>sudo systemctl status node_exporter</code>	
<pre>ubuntu@ip-172-31-44-15:~\$ 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 2022-08-13 09:18:08 UTC; 9s ago      Main PID: 11933 (node_exporter)         Tasks: 4 (limit: 1109)       CGroup: /system.slice/node_exporter.service               └─11933 /usr/local/bin/node_exporter  Aug 13 09:18:08 ip-172-31-44-15 node_exporter[11933]: level=info ts=2022-08-13T09:18:08.813Z caller=node_exporter.go:115 collector=thermal_zone Aug 13 09:18:08 ip-172-31-44-15 node_exporter[11933]: level=info ts=2022-08-13T09:18:08.813Z caller=node_exporter.go:115 collector=time Aug 13 09:18:08 ip-172-31-44-15 node_exporter[11933]: level=info ts=2022-08-13T09:18:08.814Z caller=node_exporter.go:115 collector=timex Aug 13 09:18:08 ip-172-31-44-15 node_exporter[11933]: level=info ts=2022-08-13T09:18:08.814Z caller=node_exporter.go:115 collector=udp_queues Aug 13 09:18:08 ip-172-31-44-15 node_exporter[11933]: level=info ts=2022-08-13T09:18:08.814Z caller=node_exporter.go:115 collector=uname Aug 13 09:18:08 ip-172-31-44-15 node_exporter[11933]: level=info ts=2022-08-13T09:18:08.814Z caller=node_exporter.go:115 collector=vmstat Aug 13 09:18:08 ip-172-31-44-15 node_exporter[11933]: level=info ts=2022-08-13T09:18:08.814Z caller=node_exporter.go:115 collector=xfs Aug 13 09:18:08 ip-172-31-44-15 node_exporter[11933]: level=info ts=2022-08-13T09:18:08.814Z caller=node_exporter.go:115 collector=zfs Aug 13 09:18:08 ip-172-31-44-15 node_exporter[11933]: level=info ts=2022-08-13T09:18:08.814Z caller=node_exporter.go:199 msg="Listening on" address=:9100 Aug 13 09:18:08 ip-172-31-44-15 node_exporter[11933]: level=info ts=2022-08-13T09:18:08.814Z caller=tls_config.go:191 msg="TLS is disabled." http2=false ubuntu@ip-172-31-44-15:~\$</pre>	
Host Metric (CPU, RAM, Disk, Network)	Dashboard
CPU	

RAM



DISK



NETWORK

The screenshot shows a web browser window with a Grafana dashboard. The browser's address bar displays a URL starting with 'aa5c3b6973bb04b4392e6418eae6f93-a9a5cb48ec7d8ecb.elb.us-'. The browser's tab bar shows several open tabs, including 'Advanced...', 'Shopping', 'Cooking', 'PerfTesting', 'FPT', 'Sym', 'Tools', and 'Syn'. The Grafana dashboard has a dark theme and a header bar with the text 'General / SRE-Project-1-System-Monitoring'. Below the header is a line graph titled 'Network'. The y-axis is labeled from 20 to 100 in increments of 20. The x-axis is labeled with times from 11:00 to 16:00 in hourly increments. A single data series is plotted as a green line, showing a sharp increase in network usage starting around 16:00, peaking near 100, and then fluctuating. The legend at the bottom of the graph identifies the series as '{\_\_name\_\_="instance:node\_network\_receive\_bytes:rate:sum", instance="18.188.36.17:9100"}'.

General / SRE-Project-1-System-Monitoring ☆ 🔗

Network

100  
80  
60  
40  
20

11:00 12:00 13:00 14:00 15:00 16:00

{\_\_name\_\_="instance:node\_network\_receive\_bytes:rate:sum", instance="18.188.36.17:9100"}

Responsibilities	
1. <b>Identify the problem</b>	2. <b>Define the problem</b>
3. <b>Generate hypotheses</b>	4. <b>Test the hypotheses</b>
5. <b>Evaluate the results</b>	6. <b>Communicate the findings</b>

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.

**Release manager** - Because this role is responsible for change management and code releases. When the code needs to be deployed to production, it goes to this person, who ensures the code has all dependencies satisfied and ensures proper communication is sent to the stakeholders. The release manager executes the code release and any rollback procedures if needed

**Monitoring engineer** - Because this role is responsible for creating dashboards for essential metrics such as the "four golden signals," creating and managing alert rules, and is usually the first to know if an incident occurs. Additionally, this person manages the monitoring rules and governance for monitoring newly created or existing IT infrastructure.

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** - Because this role is responsible for ensuring each team member's work is scoped and contributes to the team. In other words, this person wants to ensure everyone on the team is doing what is expected of them, no side projects for different groups or taking on more responsibilities than what is asked of them. Additionally, the team lead contributes to architecture design meetings and helps form workflow for the team.

**System architect** - Because this role is responsible for creating infrastructure that is easily scalable and replicable. Additionally, this role documents all infrastructure, reviews existing infrastructure, highlights any shortcomings, and provides ways to eliminate those. They also

make recommendations on ways to implement newer technologies and architecture and provide migration paths from existing infrastructure.

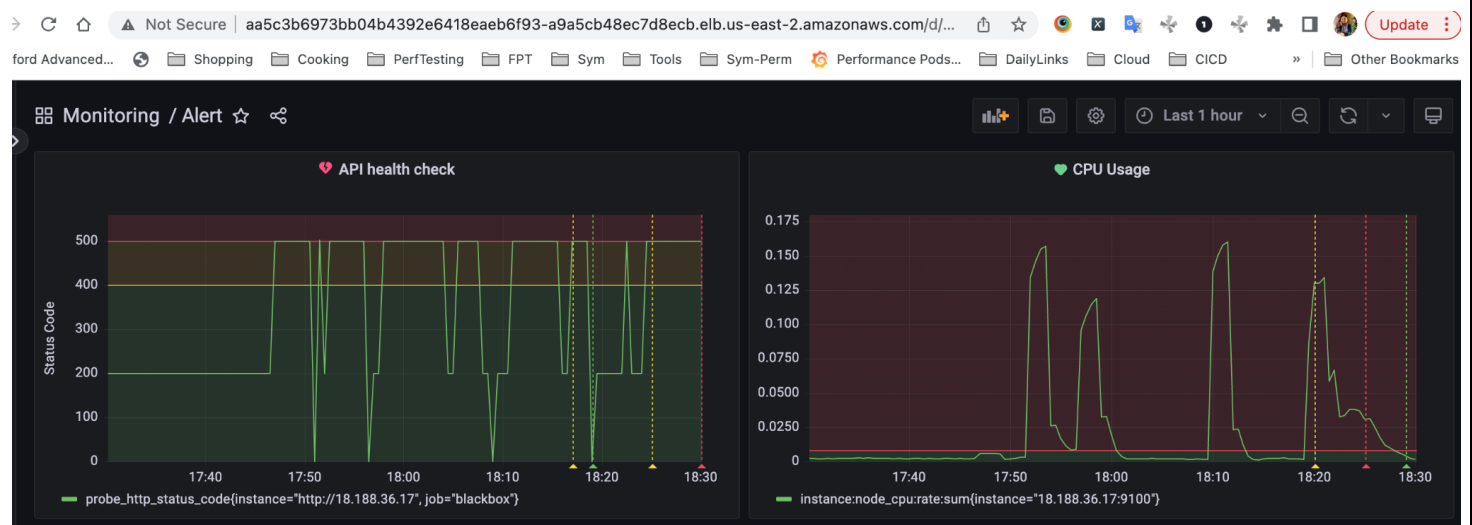
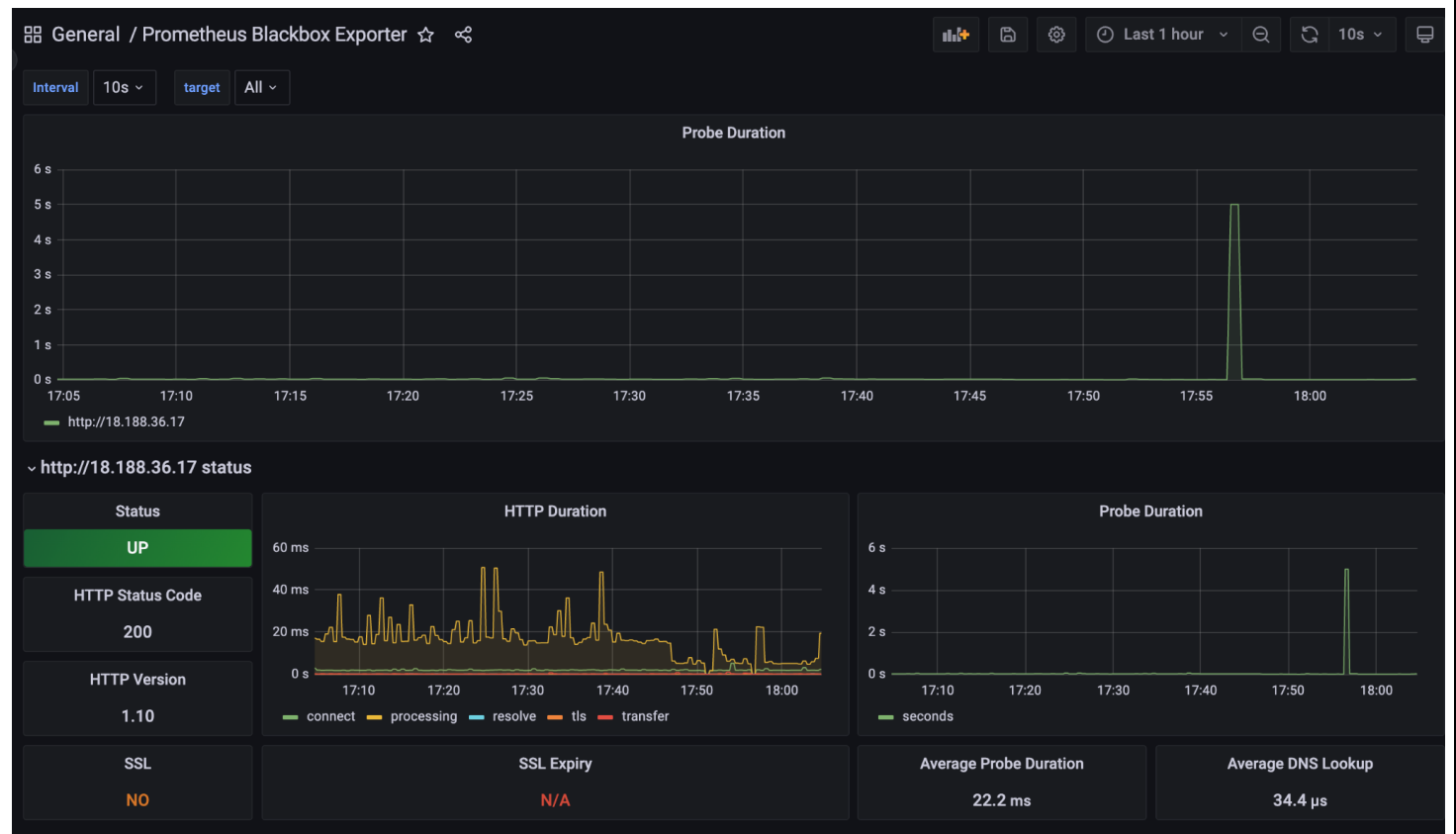
3. The emergency hotfix from question 1 was applied and is causing major issues in production. Which SRE role would primarily be involved in mitigating these issues?

**Infrastructure engineer** - This role is divided between 50% development tasks and 50% operations tasks. This person controls their own destiny as far as managing operations tasks because they can potentially use 50% of their time to automate some of those tasks. This person is also responsible for planning and executing system patches and updates

# 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.

- API



sre-api-system-monitoring-alerts

APP 6:03 PM

[FIRING:1] (API Health Check alert)

[FIRING:1] (API Health Check alert)

\*\*Firing\*\*

Value: [ var='BO' metric='probe\_http\_status\_code{instance="http://18.188.36.17", job="blackbox"}' labels={\_\_name\_\_=probe\_http\_status\_code, instance=http://18.188.36.17, job=blackbox} value=500 ]

Labels:

- alertname = API Health Check

- monitoring = alert

Annotations:


- description = There are issues from server side, APIs are returning 5xx status code

- summary = Unknown issue from Server side, Clients are getting 5xx status codes

Source: <http://localhost:3000/alerting/grafana/Sv32Z2m4k/view>

Silence: <http://localhost:3000/alerting/silence/new?alertmanager=grafana&matcher=alertname%3DAPI+Health+Check&matcher=monitoring%3Dalert>

[Show less](#)

 Grafana v9.0.5 | Today at 6:03 PM

6:08 [RESOLVED] (API Health Check alert)

[RESOLVED] (API Health Check alert)

\*\*Resolved\*\*

Value: [no value]

Labels:

- alertname = API Health Check

- monitoring = alert

Annotations:


- description = There are issues from server side, APIs are returning 5xx status code

- summary = Unknown issue from Server side, Clients are getting 5xx status codes

Source: <http://localhost:3000/alerting/grafana/Sv32Z2m4k/view>

Silence: <http://localhost:3000/alerting/silence/new?alertmanager=grafana&matcher=alertname%3DAPI+Health+Check&matcher=monitoring%3Dalert>

[Show less](#)

 Grafana v9.0.5 | Today at 6:08 PM

- CPU



[FIRING:1] (High\_CPU\_Usage\_Alert alert)

Today ▾

[FIRING:1] (High\_CPU\_Usage\_Alert alert)

\*\*Firing\*\*

Value: [ var='B0' metric='instance:node\_cpu:rate:sum{instance="18.188.36.17:9100"}'

labels={\_\_name\_\_=instance:node\_cpu:rate:sum, instance=18.188.36.17:9100}

value=0.030933333333333334 ]

Labels:

- alertname = High\_CPU\_Usage\_Alert

- monitoring = alert

Annotations:

- description = Spike in CPU usage

- summary = High CPU usage

Source: <http://localhost:3000/alerting/grafana/XtFQWhmVz/view>

Silence: <http://localhost:3000/alerting/silence/new?>

alertmanager=grafana&matcher=alertname%3DHigh\_CPU\_Usage\_Alert&matcher=monitoring%3Dalert

Dashboard: <http://localhost:3000/d/6P8ki2m4z>

Panel: <http://localhost:3000/d/6P8ki2m4z?viewPanel=4>

[Show less](#)

Grafana v9.0.5 | Today at 6:25 PM

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

- Alert Rules List

Normal CPU Usage ok Spike in CPU usage

Go to dashboard Go to panel Silence Show state history View Edit De

Labels monitoring=alert Data source Prometheus

Dashboard UID 6P8ki2m4z

Panel ID 4

Summary Spike in CPU usage

Matching instances Search by label Search State Normal Alerting Pending NoData Error

State	Labels	Created
> Normal	alertname=CPU Usage monitoring=alert	-

- API

Edit alert rule Cancel Delete Save Save

1 Set a query and alert condition

A Prometheus now-10m to now

Query patterns Run queries Explain Builder Beta

Metrics browser > probe\_http\_status\_code

> Options Legend: Verbose Format: Time series Step: auto Type: Range

B (Expression)

Operation Classic condition

Conditions WHEN last() OF A IS ABOVE 499

+ Add query + Add expression Run queries

Set alert condition

Select one of your queries or expressions set above that contains your alert condition.

☐ A - query

☒ B - expression

2 Alert evaluation behavior

- CPU



Edit alert rule

CancelDeleteSaveSave

1 Set a query and alert condition

A

Prometheus

now-10m to now

Query patterns

Run queries

Explain

Builder

Beta

Metrics browser >

instance:node\_cpu:rate:sum{instance="18.188.36.17:9100"}

> Options

Legend: Verbose

Format: Time series

Step: auto

Type: Range

B

(Expression)

Operation

Classic condition

Conditions

WHEN

last()

OF

A

IS ABOVE

0,006

+ Add query

+ Add expression

Run queries

Set alert condition

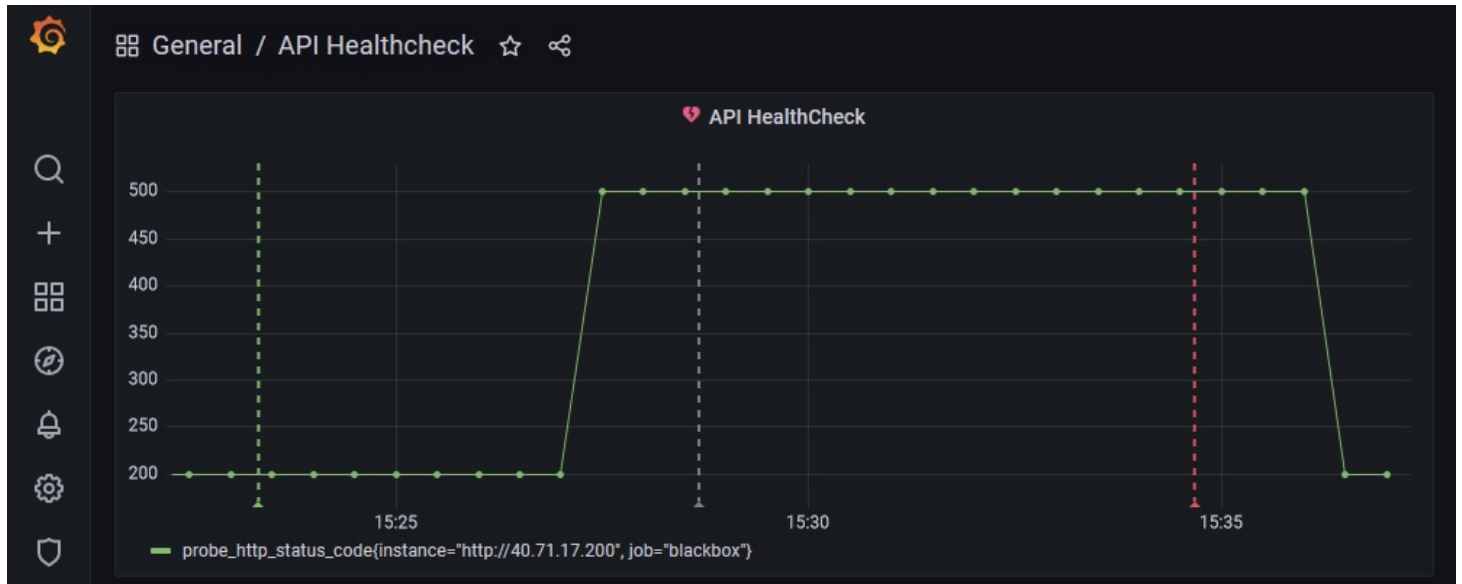
Select one of your queries or expressions set above that contains your alert condition.

☐ A - query

☒ B - expression

# 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?

*The request is down at 15:27*

*The API endpoint is healthy again at 15:36*

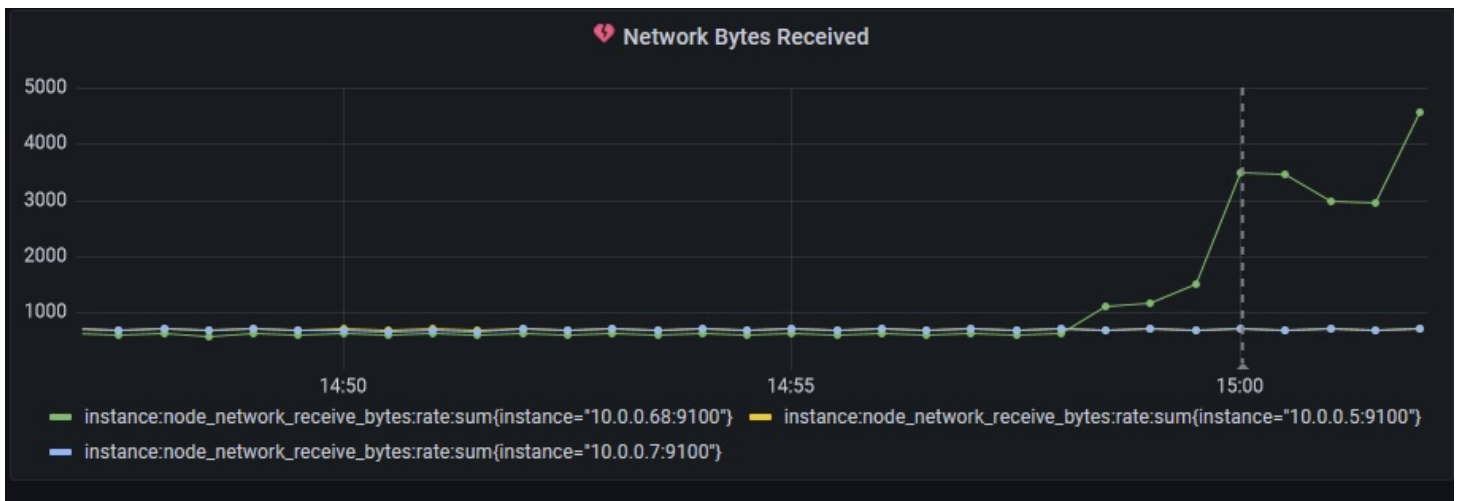
4b. If there was no SRE team, how would this outage affect customers?

*Customers are unable to run requests against sever with public ip '40.71.17.200'*

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

*Out Stage occurs in shorter period of time because SRE got alert right after the few requests got error and spike in CPU usage*

## 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 10.0.0.68 with port 9100 and it received around ~3500 bytes at 15:00*

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

*Infrastructure Engineer – because they spend 50% effort on operation tasks, so they have to know what happen with server 10.0.0.68 and why the traffic are increased quickly to avoid any outage for this instance in further*