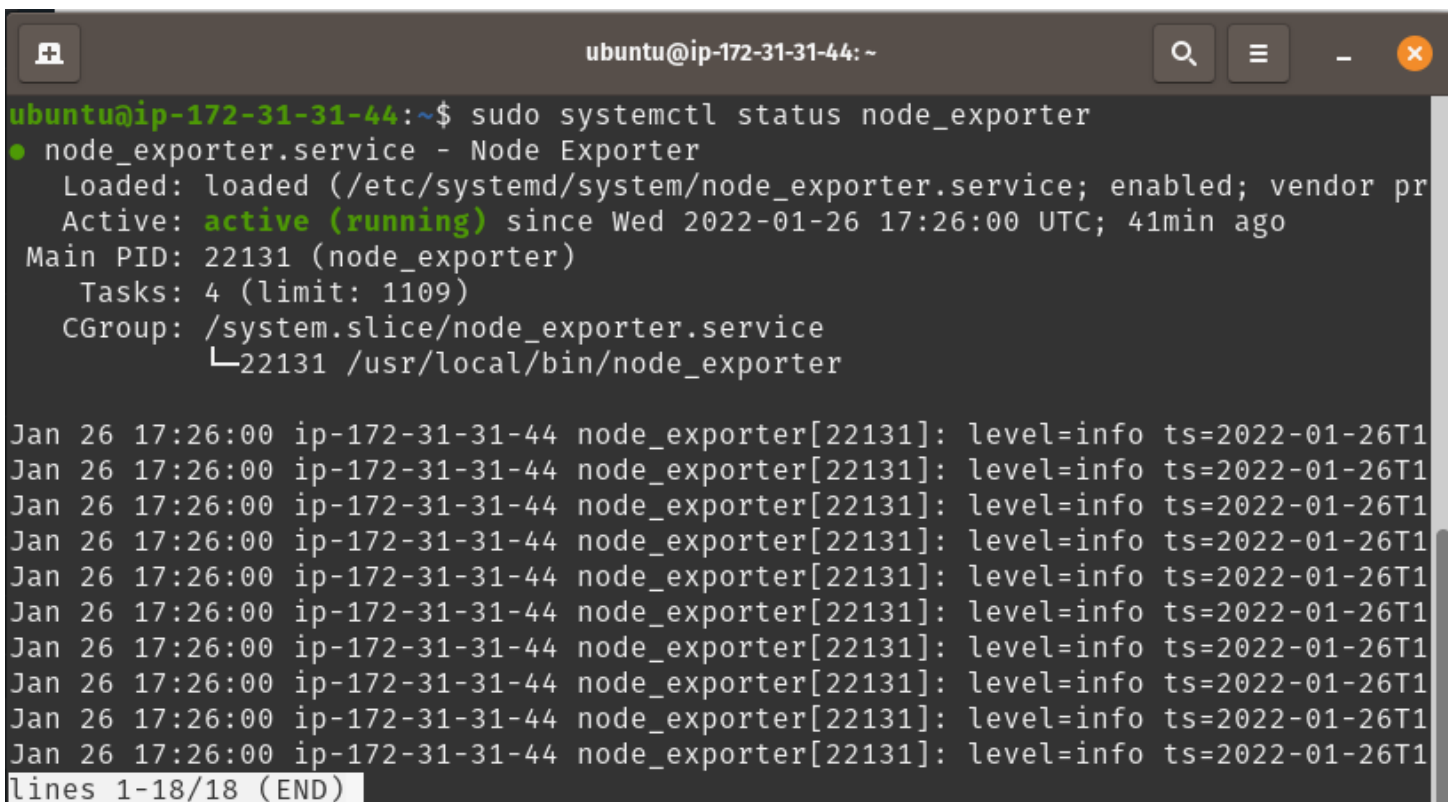# Observing Cloud Resources

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: `sudo systemctl status node_exporter`

```
ubuntu@ip-172-31-31-44:~$ sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
     Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; vendor pr
     Active: active (running) since Wed 2022-01-26 17:26:00 UTC; 41min ago
   Main PID: 22131 (node_exporter)
      Tasks: 4 (limit: 1109)
     CGroup: /system.slice/node_exporter.service
             └─22131 /usr/local/bin/node_exporter

Jan 26 17:26:00 ip-172-31-31-44 node_exporter[22131]: level=info ts=2022-01-26T1
Jan 26 17:26:00 ip-172-31-31-44 node_exporter[22131]: level=info ts=2022-01-26T1
Jan 26 17:26:00 ip-172-31-31-44 node_exporter[22131]: level=info ts=2022-01-26T1
Jan 26 17:26:00 ip-172-31-31-44 node_exporter[22131]: level=info ts=2022-01-26T1
Jan 26 17:26:00 ip-172-31-31-44 node_exporter[22131]: level=info ts=2022-01-26T1
Jan 26 17:26:00 ip-172-31-31-44 node_exporter[22131]: level=info ts=2022-01-26T1
Jan 26 17:26:00 ip-172-31-31-44 node_exporter[22131]: level=info ts=2022-01-26T1
Jan 26 17:26:00 ip-172-31-31-44 node_exporter[22131]: level=info ts=2022-01-26T1
Jan 26 17:26:00 ip-172-31-31-44 node_exporter[22131]: level=info ts=2022-01-26T1
Jan 26 17:26:00 ip-172-31-31-44 node_exporter[22131]: level=info ts=2022-01-26T1
lines 1-18/18 (END)
```

Host Metric (CPU, RAM, Disk, Network)	Dashboard
CPU	
Memory	
Disk IO	
Network Received Bytes	

UDACITY

Blacbox Exporter	

<table>
<tr><td colspan="2" align="center">

Responsibilities

</td></tr>
<tr><td colspan="2">

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.

</td></tr>
<tr><td colspan="2">

- *Release Manager*
    - *The Release manager would need to ensure all code dependencies for the hotfix and make proper notification, then make necessary code release and rollback if necessary*
- *Infrastructure Engineer*
    - *The infrastructure engineer understands how the infrastructure is set up for the given product, and would mitigate the affected area given the hotfix changes.*

</td></tr>
<tr><td colspan="2">

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.

</td></tr>
<tr><td colspan="2">

- *System Architect*
    - *The System Architect responsibility is to create the infrastructure and can provide the needed infrastructure that needed by the development team*
- *Infrastructure Engineer*
    - *The Infrastructure Engineer can work on the deployment and help with operation activity and can suggest any possibility to automate operation tasks*

</td></tr>
<tr><td colspan="2">

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?

</td></tr>
<tr><td colspan="2">

- *Release Manager*
    - *Should there be any failure on the deployment, the Release manager can initiate the rollback procedure*

</td></tr>
</table>

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).
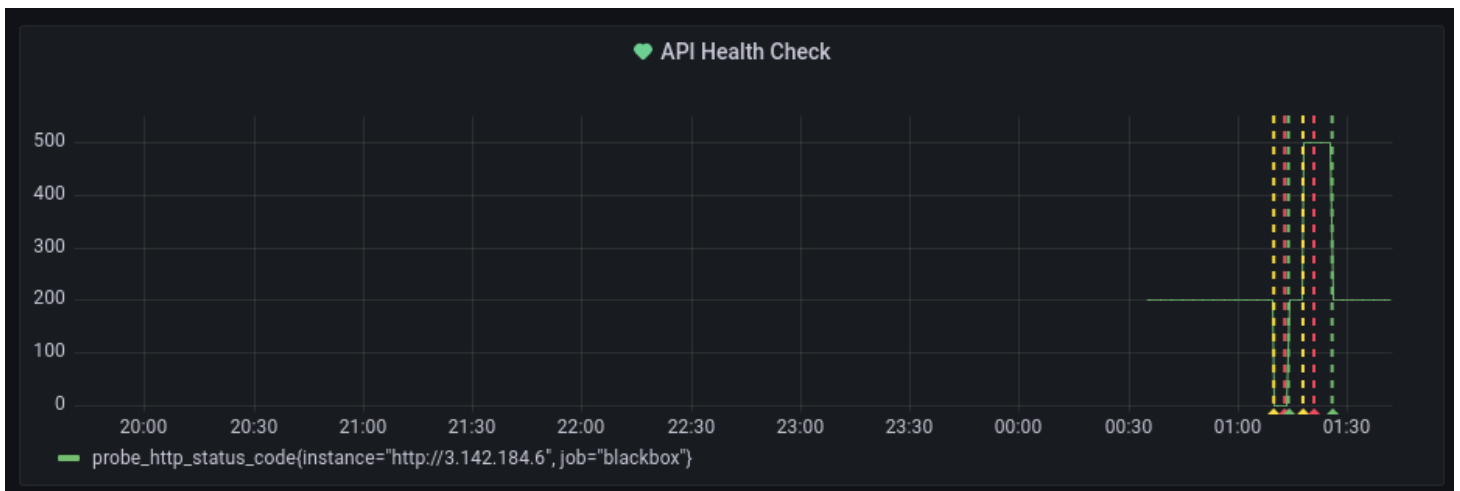
I used 2 method to trigger:
- *Kill nginx service*
- *Stop Mysql service*

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

Grafana APP 1:13 AM

Today ˅

[FIRING:1] (API Health Check)
Firing

Value: [metric='probe_http_status_code{instance="http://3.142.184.6",
job="blackbox"}' labels={__name__=probe_http_status_code,
instance=http://3.142.184.6, job=blackbox} value=0]
Labels:
- alertname = API Health Check
Annotations:
Source: http://localhost:3000/alerting/V2Jj02bnk/edit
Show more

🔆 Grafana v8.3.3 | Today at 1:13 AM

[RESOLVED] (API Health Check)
Resolved

Value: [metric='probe_http_status_code{instance="http://3.142.184.6",
job="blackbox"}' labels={__name__=probe_http_status_code,
instance=http://3.142.184.6, job=blackbox} value=0]
Labels:
- alertname = API Health Check
Annotations:
Source: http://localhost:3000/alerting/V2Jj02bnk/edit
Show more

🔆 Grafana v8.3.3 | Today at 1:18 AM

[FIRING:1] (API Health Check)
Firing

Value: [metric='probe_http_status_code{instance="http://3.142.184.6",

👋 Hello, team!   Let's use this channel for...   ✕

---

rp-sre-slack ˅   ✏

 # sre-slack ˅

＋ Add a bookmark

Show more
🔆 Grafana v8.3.3 | Today at 1:18 AM

Today ˅

◇◇ Slack Connect
⋮ Browse Slack

▼ Channels
  # general
  # random
  # sre-slack
  ＋ Add channels

▼ Direct messages
  Raditya Perdhevi you
  ＋ **Add teammates**

[FIRING:1] (API Health Check)
Firing

Value: [metric='probe_http_status_code{instance="http://3.142.184.6",
job="blackbox"}' labels={__name__=probe_http_status_code,
instance=http://3.142.184.6, job=blackbox} value=500]
Labels:
- alertname = API Health Check
Annotations:
Source: http://localhost:3000/alerting/V2Jj02bnk/edit
Show more

🔆 Grafana v8.3.3 | Today at 1:21 AM

New

[RESOLVED] (API Health Check)
Resolved

Value: [metric='probe_http_status_code{instance="http://3.142.184.6",
job="blackbox"}' labels={__name__=probe_http_status_code,
instance=http://3.142.184.6, job=blackbox} value=500]
Labels:
- alertname = API Health Check
Annotations:
Source: http://localhost:3000/alerting/V2Jj02bnk/edit
Show more

🔆 Grafana v8.3.3 | Today at 1:26 AM

U D A C I T Y

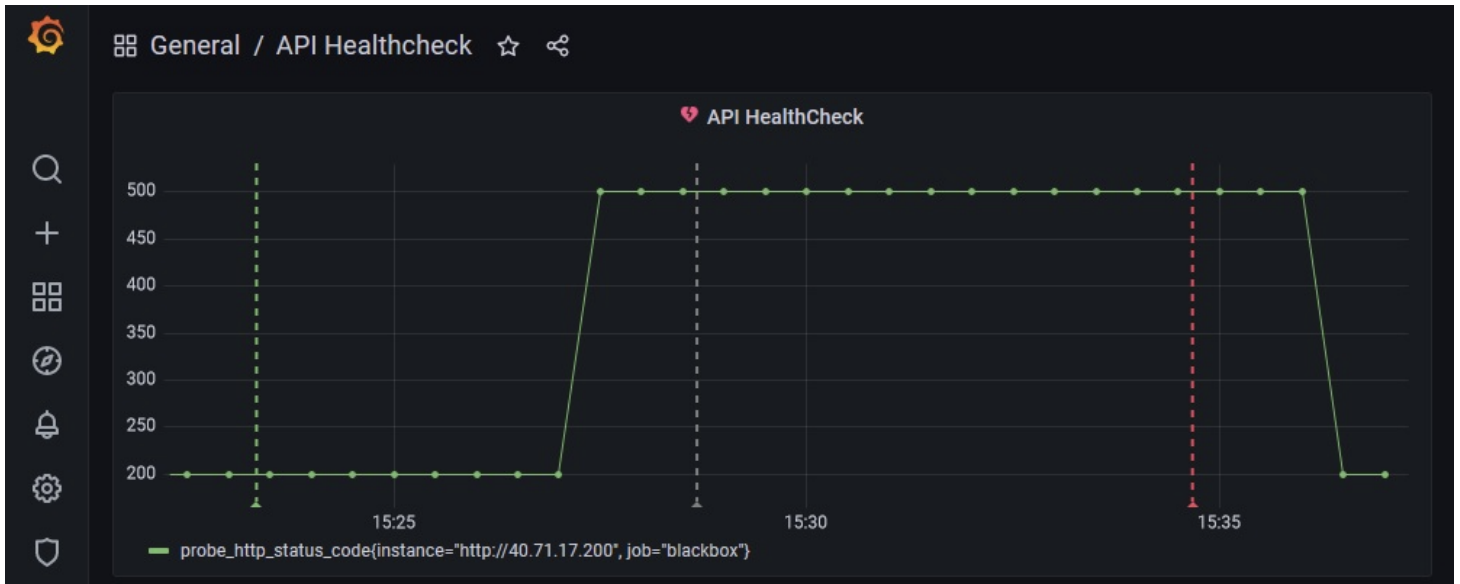Configure alert rules: Provide a screenshot of the alert rules list in Grafana.

Grafana

∨ 🗁 **My Alert**   5 rules: 1 firing | ✏️ 🔒

State		Name	Health	Summary
> Normal		Available Memory	ok	
> Normal		CPU %	ok	
> Normal		Disk IO	ok	
> Firing	for 3m	Endpoint Status	ok	
> Normal		Network Received	ok	

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?

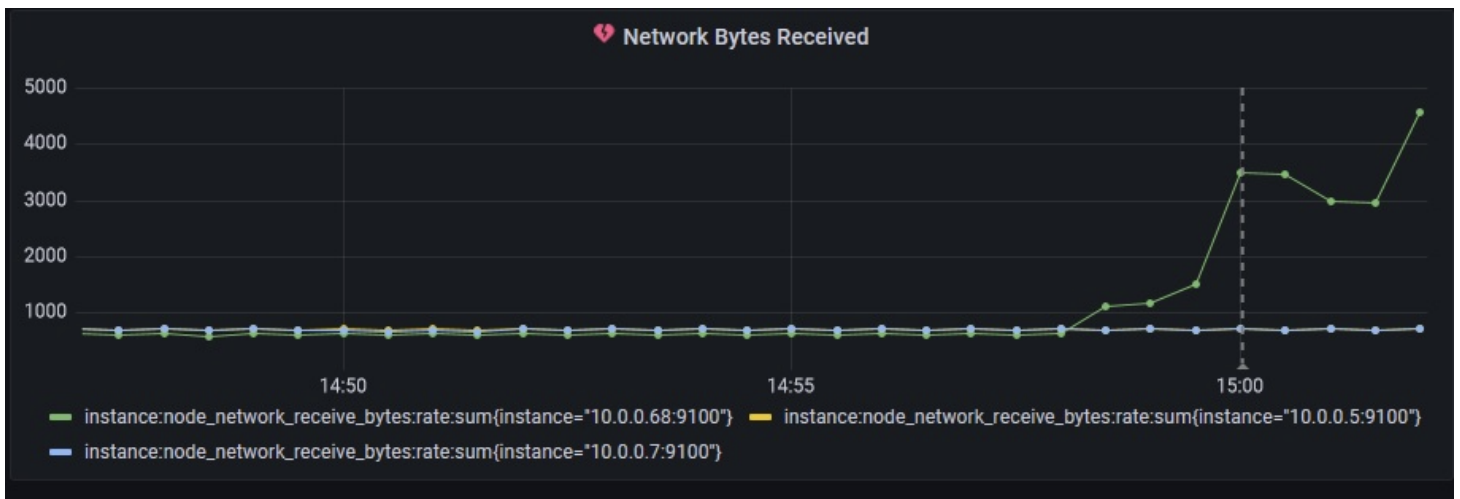*The API endpoint started to failed around 15:27 and it came back available around 15:36*

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

The customer would not be able to complete their task that involves the API call for around 9 minutes, considering the number of customers and their activity, losing the 9 minute window would be enough to give a bad impression of the service availability and when the customer discover it first before the the team would be an embarrassment

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

We can set up a synthetic monitoring that would send alert notification should there be a problem.

Graph 2

Network Bytes Received

instance:node_network_receive_bytes:rate:sum{instance="10.0.0.68:9100"} instance:node_network_receive_bytes:rate:sum{instance="10.0.0.5:9100"}
instance:node_network_receive_bytes:rate:sum{instance="10.0.0.7:9100"}

5a. Given the above graph, which instance had the increase in traffic, and approximately how many bytes did it receive (feel free to round)?
The instance receiving the increase is 10.0.0.68 with approximately 3500
5b. Which team members on the SRE team would be interested in this graph and why?
Based on the graph it would be the Monitoring Engineer as they would be the first to know and can manage the traffic load to suppress the possibility of overloaded requests.