

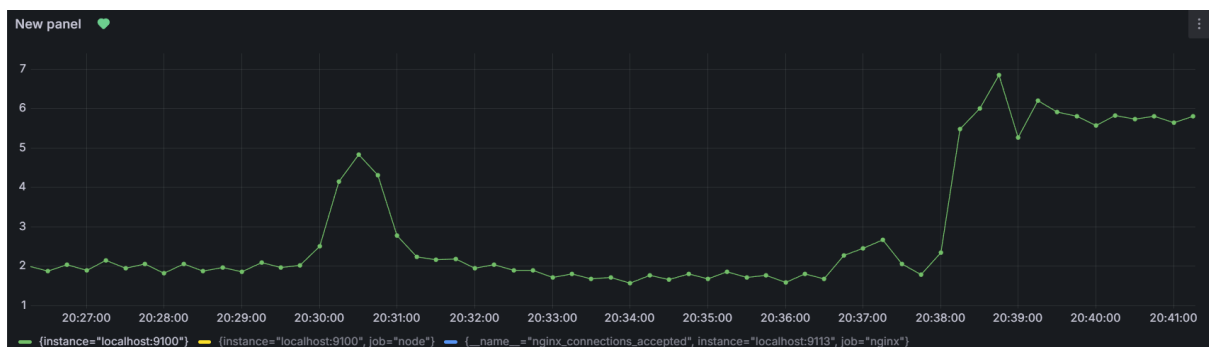
Cloud computing assignment 01

2023320302

데이터과학과

이유찬

1. Dashborad screenshots/query/explanations

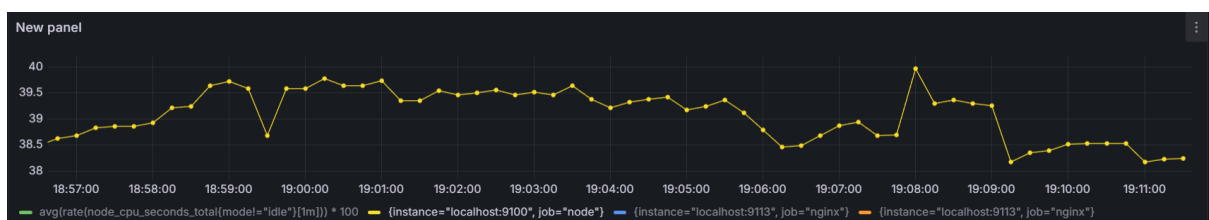


CPU usage

$100 * \text{sum by (instance) (rate(node_cpu_seconds_total\{mode\!=\"idle\"\}[1m]))} / \text{sum by (instance) (rate(node_cpu_seconds_total}[1m]))$

Gets the CPU's non-idle time, then divide it with total time passed for all cores for 1 minute window.

Then multiply by 100 to get percentage.



Memory usage

$(1 - (\text{node_memory_MemAvailable_bytes} / \text{node_memory_MemTotal_bytes})) * 100$

Subtracts the ratio of (available memory / total memory) (= unused memory) from 1, to get used memory ratio.

Then multiply by 100 to get percentage.



Nginx accepted connections (blue)

nginx_connections_accepted

Nginx HTTP Requests (Orange)

nginx_http_requests_total

Both used single metric that is scraped from nginx exporter.

(Tried using delta operation to get differences, but didn't matter that much.)

2. ApacheBench load testing

Small load

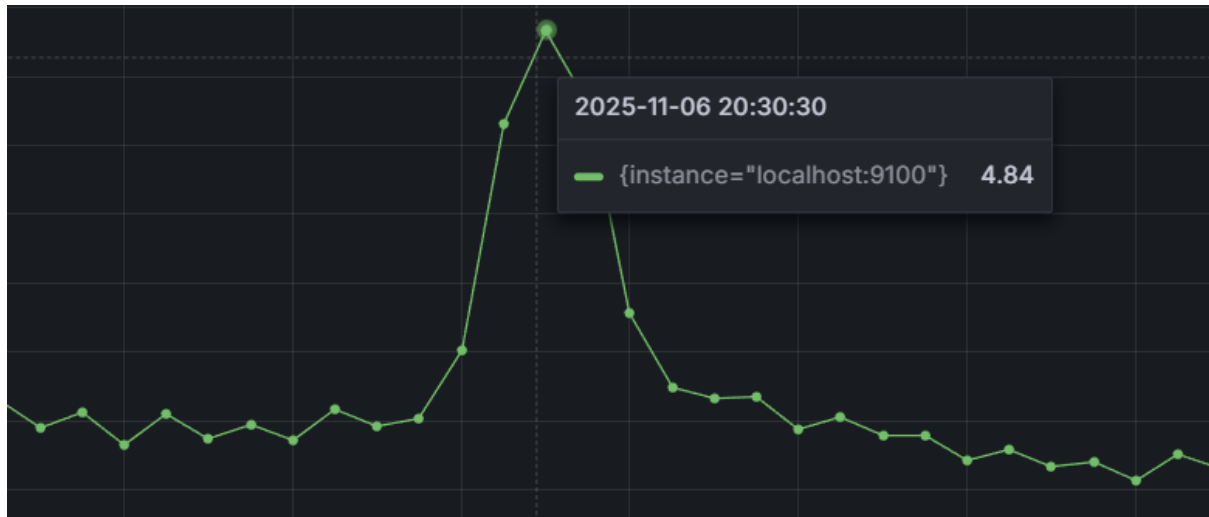


CPU usage almost had no spike, moving around 1.8%.

Memory usage had small spikes, moving around 38%.

Both accepted connections and requests were growing linearly, as more and more requests were made.

Medium load



CPU usage had a spike, up to 4.84%.

Memory usage also had small spikes, moving around 39%.

Both accepted connections and requests were growing linearly, as more and more requests were made.

Large load



CPU usage had a big spike, up to 6.85%.

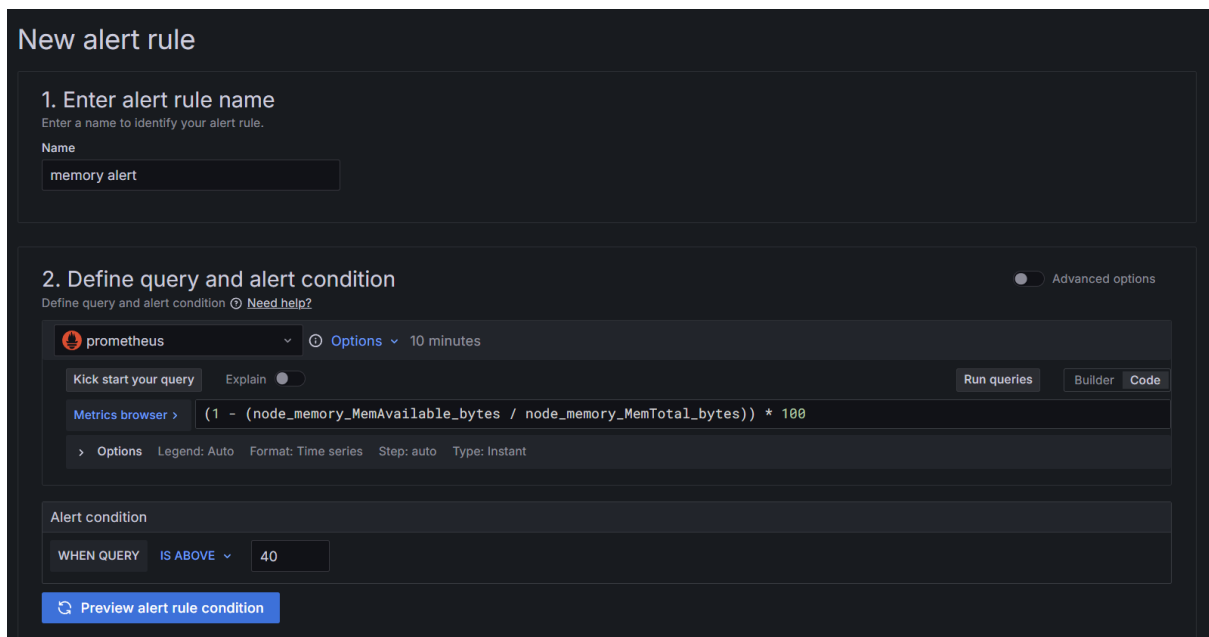
Memory usage also had spikes, moving around 41%.

Both accepted connections and requests were growing linearly, as more and more requests were made.

For all of them, 'Nginx HTTP requests' were slightly bigger than 'Nginx accepted connections'.

3. Alerts

Query: same with above's memory query



The screenshot shows the 'New alert rule' configuration page in Prometheus. It is divided into two main sections:

- 1. Enter alert rule name:** A text input field labeled 'Name' contains the text 'memory alert'.
- 2. Define query and alert condition:** This section includes:
 - A dropdown menu set to 'prometheus'.
 - A toggle for 'Options' and a time range set to '10 minutes'.
 - A 'Kick start your query' button and an 'Explain' toggle.
 - Buttons for 'Run queries', 'Builder', and 'Code'.
 - A 'Metrics browser' link and a query input field containing: $(1 - (\text{node_memory_MemAvailable_bytes} / \text{node_memory_MemTotal_bytes})) * 100$.
 - A row of settings: 'Options', 'Legend: Auto', 'Format: Time series', 'Step: auto', and 'Type: Instant'.
 - An 'Alert condition' section with a 'WHEN QUERY' dropdown, an 'IS ABOVE' dropdown, and a value input field set to '40'.
 - A 'Preview alert rule condition' button at the bottom.

If memory usage is over 40%, fire alert.

I used apache bench (ab -n 100000 -c 1000) to make it fire.

The alert:

☆ [받은메일함] [FIRING:1] memory alert alert (localhost:9100 node) [🔗](#)

2025. 11. 6. (목) 20:08

보낸사람 Grafana<udyann@korea.ac.kr>
받는사람 <udyann@korea.ac.kr>



alert › memory alert

1 firing instances

Firing

memory alert

View alert

Values

A=40.09889885200308 C=1

Labels

alertname	memory alert
grafana_folder	alert
instance	localhost:9100
job	node

Silence

The dashboard:

