Intern - Site Reliability Engineer Assignment

Name: Kunapoom Oprik
Email: kunapoom.opr@student.mahidol.ac.th

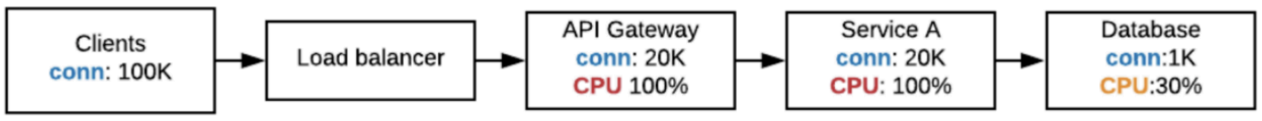
Instruction: Please answer the assignment in English by following the tasks below

---

1. Please explain your understanding about the responsibility of DevOps, System Engineer, and SRE roles?

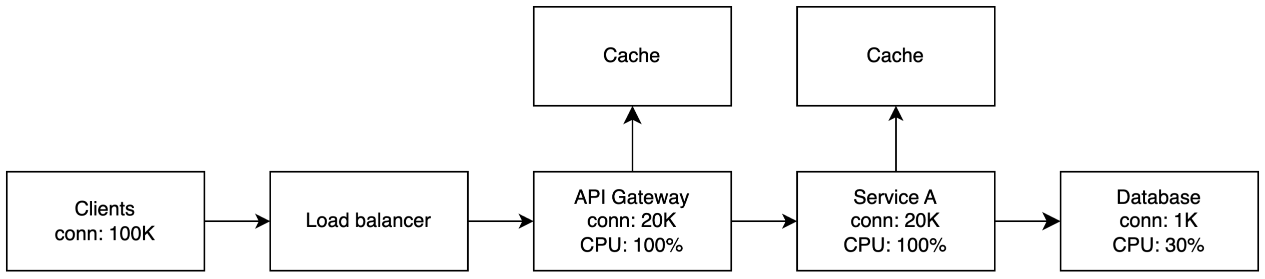
Each role has different responsibilities and focuses, but most of them use the same tools.
- **DevOps** improves software delivery and automation.
- **System Engineer** maintains infrastructure and IT systems.
- **SRE** ensures reliability, uptime of services and manages incidents.

2. Assume we have an application which is designed as below. Our application stopped responding due to the extremely high number of clients in some circumstances.



conn = current connections / K = thousand

We have tried scaling API Gateway and Service A nodes but it didn't help. What are the possible problems in our system, in which components, and how to solve them?



Since API Gateway and Service A have 100% CPU usage, Add caching at API Gateway and Service A to reduce redundant work lower load load to the system, and improve response time.

3. When you run kubectl apply on a Kubernetes manifest YAML file? What happens from there until your workload starts running? Give as much details as Possible.
    1. Kubectl reads YAML file and make API call to K8s API server.
    2. K8s API server validate the manifest YAML file and check permissions.
    3. If the resource doesn't exist, it will be created.
    4. Write / Update state into the etc data store.
    5. K8s API server call scheduler assigns the new pods to nodes based on resource availability.
    6. Kubelet on node receive pod specification from API server, then container runtime pull images from registry and start the container. And then your workload start running.

4. How do you normally solve conflicts in a feature branch before merging code to git repo?

1.Ensure that feature branch is up to date

```
git checkout main
git pull origin main
git checkout feature
git merge main
```

2.Identify conflicts using git status

3.Resolve conflict manually by edit conflict file

```
<<<<<<< HEAD
print(“Hello World!”)
=======
print("Hello, everyone!")
>>>>>>> main
```

4.Commit the merged file.

```
git add <merged_file>
git commit -m “Resolved conflict”
```

5. Please call the openweathermap API https://openweathermap.org/current and generate a report on where it's raining. Your code and report are required for this assignment. (Sub
   https://github.com/PAOPAKEM/openweather
6. Please review the following scenario and answer the following questions.

Scenario:
You are tasked with running cAdvisor on a system to collect and expose metrics in Prometheus format.The goal is to create a monitoring tool that continuously tracks container metrics throughout the day and generates a daily report based on the data. The report should summarize key metrics, identify any anomalies or issues, and focus on critical resource usage, such as CPU, memory, disk usage, and network connectivity.

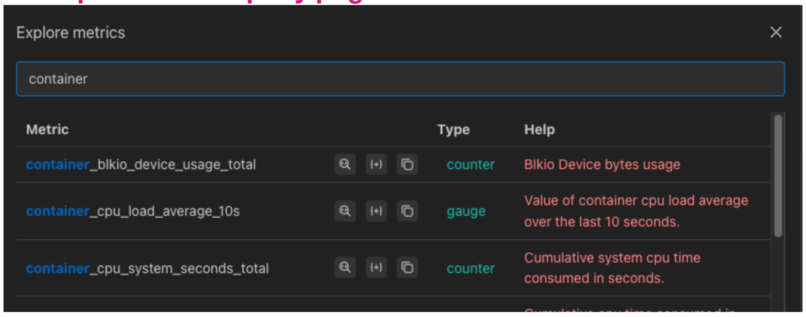
Question:
Imagine you’re working in a team that needs to monitor the performance of containers using cAdvisor, and you’re asked to:

1. Set up cAdvisor on a system to collect and expose container metrics in Prometheus format.
   a. How would you go about setting up cAdvisor on a host and ensuring it exposes metrics properly?

   Running cAdvisor in a Docker Container and visiting https://localhost:8080/metrics **to verify that metrics are available**

```
VERSION=v0.49.1 # use the latest release version from https://github.com/google/
cadvisor/releases
sudo docker run \
  --volume=/:/rootfs:ro \
  --volume=/var/run:/var/run:ro \
  --volume=/sys:/sys:ro \
  --volume=/var/lib/docker/:/var/lib/docker:ro \
  --volume=/dev/disk/:/dev/disk:ro \
  --publish=8080:8080 \
  --detach=true \
  --name=cadvisor \
  --privileged \
  --device=/dev/kmsg \
  gcr.io/cadvisor/cadvisor:$VERSION
```

b. What are the key container metrics (such as CPU, memory, disk, and network usage) that you would monitor?
From permetheus query page. I Choose



Explore metrics

container

Metric	Type	Help
container_blkio_device_usage_total	counter	Blkio Device bytes usage
container_cpu_load_average_10s	gauge	Value of container cpu load average over the last 10 seconds.
container_cpu_system_seconds_total	counter	Cumulative system cpu time consumed in seconds.

1. **container_cpu_usage_seconds_total:** Cumulative cpu time consumed in seconds.
2. **container_cpu_load_average_10s:** Value of container cpu load average over the last 10 seconds.
3. **container_memory_usage_bytes:** Current memory usage in bytes, including all memory regardless of when it was accessed.
4. **container_memory_failcnt:** Number of memory usage hits limits
5. **container_fs_usage_bytes:** Number of bytes that are consumed by the container on this filesystem.
6. **container_fs_io_current:** Number of I/Os currently in progress.
7. **container_network_receive_bytes_total:** Cumulative count of bytes received.
8. **container_network_receive_errors_total:** Cumulative count of errors encountered while receiving

2. Create a monitoring tool to collect and analyze these metrics throughout the day.
   a. What tools or frameworks would you use to monitor and store metrics over time?
   Prometheus: Scrapes metrics from cAdvisor and stores them in a time-series database.
   b. How would you ensure that the data is collected at regular intervals and stored for further analysis?
   - check that we define right targets and interval in prometheus.yml file.
   - **-- storage.tsdb.retention.time=30d** use this flag to retain data in storage.
3. Generate a daily report based on the collected metrics, including the identification of any issues or anomalies.
   a. What specific metrics would you focus on in your daily report?
   CPU, Memory, Disk, Network, Anomalies Alerts
   b. How would you summarize and present these metrics to highlight any potential issues (e.g., high CPU usage, memory overuse, disk space exhaustion, or network errors)?
   use Gafrana Graph to highlight anomalies
   c. What format would you use for the report? Would you prefer a graphical representation (e.g., graphs in Grafana) or a written summary in a document like PDF/Excel?
   graphs in Grafana
4. Identify and highlight potential problems in the system's performance