Site Reliability Engineer (DevOps) Test

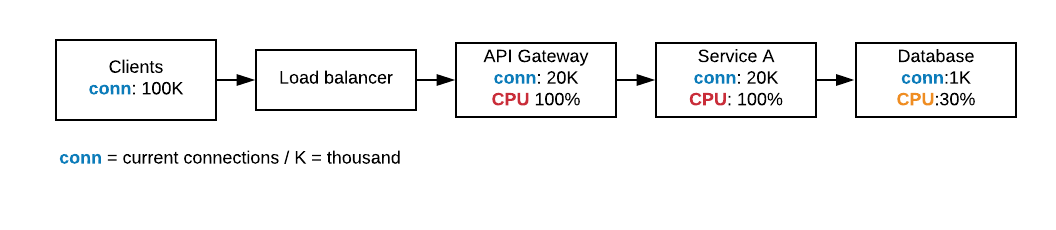
# INSTRUCTIONS

Estimated completion time: 20 hours

1. There are 21 questions in this test. Please read through all the questions before you start.
2. Please answer by explanation, diagram or example to measure your knowledge and understanding in each question. You will be judged not only on correctness but also the understanding and originality of your answer.
3. For the #21 question, please compress your answer into zip file and upload it along with this file.
4. Final score will be based on 50% from written test and 50% from the last hands-on question

# QUESTIONS

1. Describe Git branching strategies (Git-flow, single branch, feature branch etc.) which you have used and what purpose does it serves.
2. How do you revert a commit that has already been pushed and made public?
3. How do you normally solve conflicts in a feature branch before merge?
4. Assume we have application which is designed as below. Our application stopped responding due to extremely high number of clients in some circumstance.



We have tried scaling a number of API Gateway and Service A nodes but it didn’t help. What are the possible problems lies in our system in which components and how to fix it?

1. “503 service unavailable” what does it mean and how to identify the problem?
2. What are the Linux network tools do you use for troubleshooting network problems as well as usage scenario for each tool?
3. Please write short POSIX compliant shell script which accepts multiple arguments as number and print summary result of all input.

For example:

$ ./sum 1 2 3

6

$ ./sum 1 2 3 4 5 6 -7

14

$ ./sum 1 2 3 4 5.5 -10

5.5

1. Assume that you are using private cloud for your infrastructure. How do you manage logs, metrics and alerts your infrastructure and applications? Which tools do you use and why?
2. How do you secure the following?

* application
* infrastructure
* data

1. Base on your experience, how do you reduce your service downtime as much as possible during

* software upgrade
* database migration
* incident

1. Configuration management
   1. Which Among Puppet, chef, Ansible is the best Configuration management tool?
   2. Why?
   3. Do you still need to use it if you already have docker-swarm or Kubernetes?
2. How would you handle secret value in CD/CD pipeline in a secure fashion?
3. How do you keep docker image smallest as possible?
4. What is the different between overlay, bridge, host network in Docker? When to use each of them?
5. What is the difference between Alpine and Debian Linux distribution and what kind of impact the each has in containerization?
6. How do you design your Kubernetes cluster? what DNS, CNI, ingression is being use? why?
7. How does Kubernetes service talk to each other in the same cluster?
8. What’s different between L2, L4 and L7 Load balancer? When to use?
9. How do you measure service quality to give best experience to your customer? (SLO, SLA)
10. Assuming there is customer affecting incident and you are SRE responsible for the product. How would your incident handling process be until the issue resolution?
11. Hands-on work: check the provided program in *example.zip*,
    1. Please build and deploy it using docker and Kubernetes to be able to run as the diagram below. It must support multiple environments (e.g. beta, prod).

Diagram

Description automatically generated

The solution must include the following

* + - Dockerfile
    - Ingress configuration
    - Kubernetes deployment and service (including necessary datastore such as Redis)
    - Jenkinsfile (or any other similar tool’s build script) for CI/CD
    - **Extra points**  if Redis can keep state after container restart

Please update README for steps how to test, build, deploy and run the application

* 1. How do you monitor this service health? Which tool do you use as well as configuration?

-- END OF DOCUMENT --