QUESTIONS

Normal

1. Describe Git branching strategies (Git-flow, single branch, feature branch etc.) which you have used and what purpose does it serves.
   - Create new branch for new feature and merge to main branch for deploy
2. How do you revert a commit that has already been pushed and made public?
   - Create new branch and push new edit file to fix then merge.
3. How do you normally solve conflicts in a feature branch before merge?
   - Using merge request function in VCS
4. “200 OK” what does it mean and show use case this HTTP Status?
   - 200 OK success status response code indicates that the request has succeeded. The meaning success is server send response
5. “201 Created” what does it mean and show use case this HTTP Status?
   - 201 Created success status response code indicates that the request has succeeded and has led to the creation of a resource. Use case is result of post request
6. “301 Moved Permanently” what does it mean and show use case this HTTP Status?
   - 301 Moved Permanently. It is used for permanent redirecting, meaning that links or records returning this response should be updated. The new URL should be provided in the Location field
7. “400 Bad Request” what does it mean and how to identify the problem?
   - Server can’t process the request for example deceptive request routing
8. “401 Unauthorized” what does it mean and how to identify the problem?
   - Lack authentication please add authentication to access
9. “403 Forbidden” what does it mean and how to identify the problem?
   - Server refuses authorize please check permission
10. “404 Not Found” what does it mean and how to identify the problem?
    - Page not found please check URL
11. “500 Internal Server Error” what does it mean and how to identify the problem?
    - Internal server error such as some file is error ex. File config
12. “502 Bad Gateway” what does it mean and how to identify the problem?
    - the server, while acting as a gateway or proxy, received an invalid response from the upstream server.
13. “503 Service Unavailable” what does it mean and how to identify the problem?
    - Server is not ready to handle the request. Please refresh that page
14. “504 Gateway Timeout” what does it mean and how to identify the problem?
    - Gate way or proxy did not get response in time please check query code

15. What are Linux network tools do you use for troubleshooting network problems as well as usage scenarios for each tool?

    ANS:

    - Putty need to access to linux for see error something
      Putty has many feature for use such as SSH support, Session Management, Port Forwarding, remote server managment

Intermediate

1. Assume we have an application that is designed as below. Our application stopped responding due to an extremely high number of clients in some circumstances.
   We have tried scaling a number of API Gateway and Service A nodes but it didn't help.
   What are the possible problems that lies in our system in which components and how to fix them?

   Ans: May be possible problem is

   - memory leak or software bug
   - configuration error
   - Dependency Failures, your service may depend on other services, databases, or third-party APIs
   - Network issue such as network loss, high latency
   - Infrastructure Problems such as hardware failures, virtual machine crashes

   Or cloud provider issue

   - vulnerabilities or malicious attacks
2. How do you keep the docker image smallest as possible?

   ANS:

   - Use Minimal Base Images: Start with a minimal base image like Alpine Linux, BusyBox, or Scratch
   - Multi-Stage Builds
   - Use Distroless Images
3. How does the Kubernetes service talk to each other in the same cluster?

   ANS:

   Pods and Containers: Pods are the smallest deployable units in Kubernetes and can contain one or more containers. Each pod is assigned a unique IP address within the cluster, and containers within the same pod share the same network namespace, allowing them to communicate with each other via localhost.

4. What's different between L2, L4, and L7 Load balancers? When to use it?

   ANS:

- L4 load balancing traffic management of transactions at the network protocol layer (TCP/UDP)

  L7 load balancing works at the highest level of the OSI model

Professional

1. Assume that you are using a private cloud for your infrastructure. How do you manage logs, metrics, and alerts for your infrastructure and applications? Which tools do you use and why?
   ANS:

- Log Management
  Elasticsearch, Logstash, and Kibana (ELK Stack)
  These tools provide centralized log management, real-time log analysis, and powerful search capabilities
- Metrics Monitoring
  Prometheus with Grafana
  Prometheus is a popular open-source monitoring system that collects metrics from monitored targets by scraping HTTP endpoints.
- Alerting
  Prometheus Alert manager
  Prometheus Alert manager integrates seamlessly with Prometheus to handle alerts generated by Prometheus servers.

2. How do you secure the following?
   - application

   ANS:

   Not use library that high vulnerabilities

   Not use deprecated code

   Following 12 Factor App
   - infrastructure

   ANS:

   Use VPN

   Use container security concept

   Using Extended detection and response (XDR)
   - data

   ANS:

   Encrypt Sensitive Data

3. Based on your experience, how do you reduce your service downtime as much as possible during
   - software upgrade

   ANS:

- Using blue green deployment
- Automate the deployment process as much as possible to reduce the risk of human error and speed up the upgrade process.
   - database migration

   ANS:

- Back up database before data migration
   - incident

   ANS:

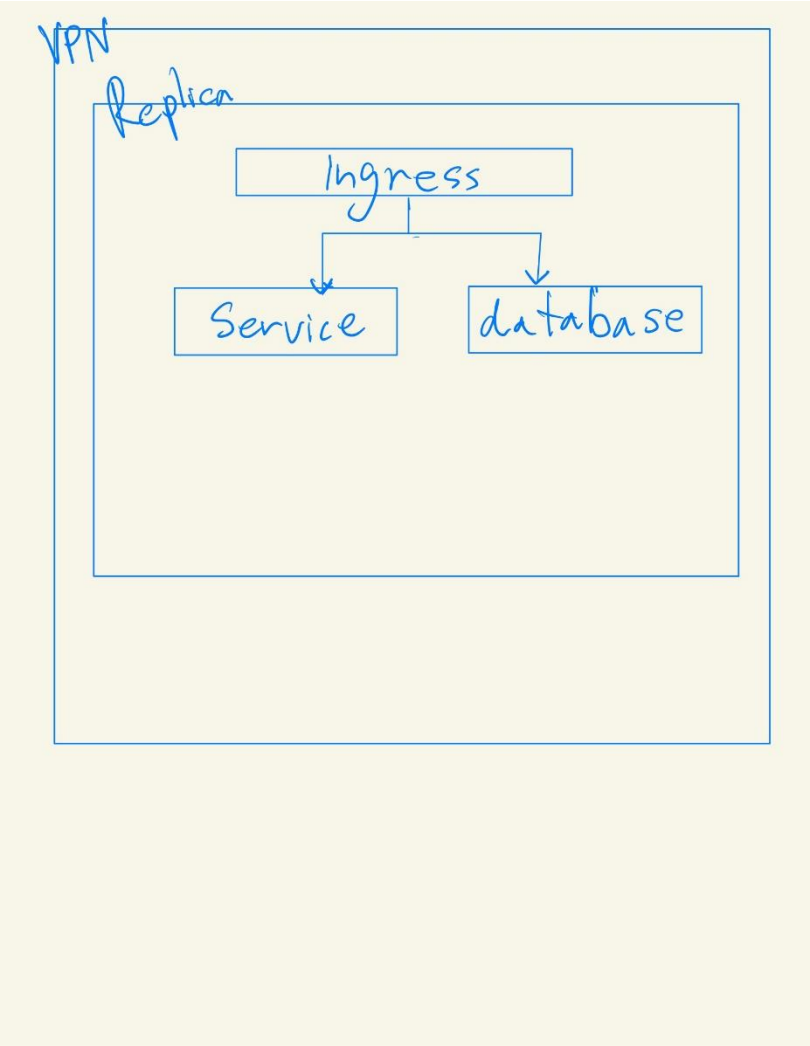
- Incident Response Plan that can help you for manage task and plan what is happed and what I will do
- Real-time Monitoring: Implement monitoring tools that continuously track the health and performance of your systems

4. How do you design your Kubernetes cluster? what DNS, CNI, ingression is being used? Why?

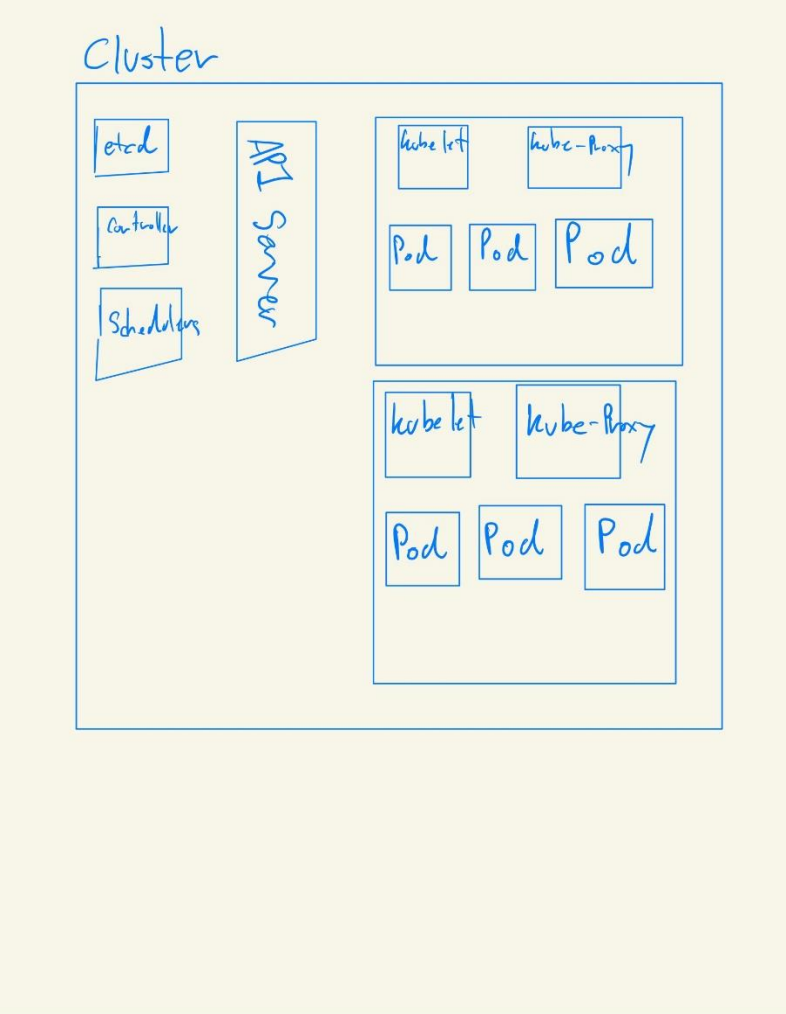
- The Domain Name System (DNS) turns domain names into IP addresses, which browsers use to load internet pages. Every device connected to the internet has its own IP address, which is used by other devices to locate the device.
- CNI Container Network Interface is allow different networking to be used with container runtimes

- Ingression is load balancer

5. How do you design your ? (Answer as draw.io or your tool of choices)
    - Networking
    - Security
    - Data Durability
    - High Availability
    - Autoscaling



6. How do you measure service quality to give the best experience to your customer? (SLO, SLA)
   ANS:
   - Customer Surveys: to get provide direct feedback from customer
   - Service Level Agreements: defined standards establish expectation for response that help ensure consistent service quality
   - Customer feedback: analyzing customer complaints and improve that your product to better
7. How do you design a highly scalable Kubernetes cluster and which Kubernetes objects are being used? (answer as drawio or your tool of choices)



Hands-On

1. Please build and deploy your application of choice using Kubernetes. It should support multiple environments (e.g. beta, prod).

The solution must include the following

- Architecture Diagram
- IAC Scripts (CDK.TF or Pulumi) to provision an infrastructure
- Dockerfile or Buildpacks supports the following languages:
  - .Net Core
  - Go
  - Java
  - Node
  - Rust

CI/CD pipeline file:

- You can choose your CI/CD tool:
  - Gitlab CI
  - Github Actions
  - CircleCI
  - Jenkins
  - Travis CI
  - GoCD
  - BitBucket
  - Azure DevOps
- Kubernetes Manifests (Deployments, Services, Config Maps, Secret Management (Secrets or Cloud Managed Vault)
- Kubernetes Ingress Configuration

Please update README for steps on how to test, build, deploy and run the application using Kubernetes (answer as draw.io or your tool of choice with source code and demo)

Algorithms

Choose your favorite language (Rust would be an advantage) to solve the following challenges:

1. Given an input string `s` and a pattern `p`, implement regular expression matching with support for `'.'` and `'*'` where:

- `'.'` Matches any single character.
- `'*'` Matches zero or more of the preceding element.

The matching should cover the entire input string (not partial).

Example 1:

Input: s = "aa", p = "a"

Output: false

Explanation: "a" does not match the entire string "aa".

Example 2:

Input: s = "aa", p = "a*"

Output: true

Explanation: '*' means zero or more of the preceding element, 'a'. Therefore, by repeating 'a' once, it becomes "aa".

Example 3:

Input: s = "ab", p = ".*"

Output: true

Explanation: ".*" means "zero or more (*) of any character (.)".

Constraints:

- 1 <= s.length <= 20

- 1 <= p.length <= 30
- s contains only lowercase English letters.
- p contains only lowercase English letters, '.', and '*'.
- It is guaranteed for each appearance of the character '*', there will be a previous valid character to match.

2. Bob has a server farm crunching numbers. He has nodes servers in his farm. His company has a lot of work to do. The work comes as a number workload which indicates how many jobs there are. Bob wants his servers to get an equal number of jobs each. If that is impossible, he wants the first servers to receive more jobs. He also wants the jobs sorted, so that the first server receives the first jobs. The way this works, Bob wants an array indicating which jobs are going to which servers.
Can you help him distribute all this work as evenly as possible onto his servers?

Example

Bob has 2 servers and 4 jobs. The first server should receive job 0 and 1 while the second should receive 2 and 3.

distribute(2, 4) # => [[0, 1], [2, 3]]

On a different occasion Bob has 3 servers and 3 jobs. Each should get just one.

distribute(3, 3) # => [[0], [1], [2]]

A couple of days go by and Bob sees a spike in jobs. Now there are 10, but he hasn't got more than 4 servers available. He boots all of them. This time the first and second should get a job more than the third and fourth.

distribute(4, 10) # => [[0, 1, 2], [3, 4, 5], [6, 7], [8, 9]]

3. It's tricky keeping track of who is owed what when spending money in a group. Write a function to balance the books.

- The function should take one parameter: a dict with two or more name-value pairs which represent the members of the group and the amount spent by each.
- The function should return a dict with the same names, showing how much money the members should pay or receive.

Further points:

- The values should be positive numbers if the person should receive money from the group, negative numbers if they owe money to the group.
- If value is a decimal, round to two decimal places.

Example:

3 friends go out together: A spends $20, B spends $15, and C spends $10. The function should return an object/dict showing that A should receive $5, B should receive $0, and C should pay $5.

```
var group = {
   A: 20,
   B: 15,
   C: 10
}

splitTheBill(group) // returns {A: 5, B: 0, C: -5}
```

4. Fibonacci

The Fibonacci numbers are the numbers in the following integer sequence.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ……..

In mathematical terms, the sequence Fn of Fibonacci numbers is defined by the recurrence relation

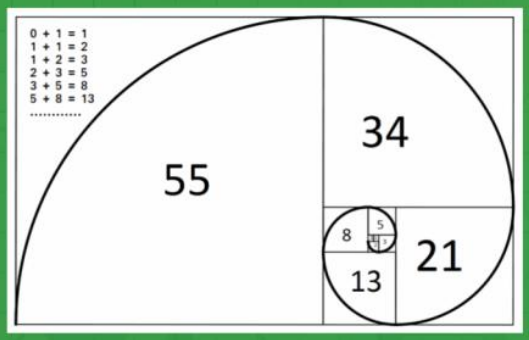
$F_n = F_{n-1} + F_{n-2}$

with seed values

$F_0 = 0$ and $F_1 = 1$.

Given a number n, print n-th Fibonacci Number.

Program for Fibonacci numbers



Examples:

```
Input  : n = 2
Output : 1

Input  : n = 9
Output : 34
```

5. Palindrome
Given a string, write a function to check if it is palindrome or not.

- **In case: even** you can reverse the string
- **In case: odd** you should use the middle to split the word and then reverse the string

A string is said to be palindrome if the reverse of the string is the same as the string. For example, “abba” is Palindrome, but “abbc” is not Palindrome.

Anna, civic, kayak, level, madam, mom, noon, racecar, radar, redder, refer, repaper, rotator, 12321, 15651

How to Submit this Technical Home Test (in your Github Repository)

Output format on Github

jumpbox-part-time or jumpbox-contract or jumpbox-intern

|-> README.md

|-> question

|-> hand-on

    |-> h01

    |-> h02

    |-> h03

|-> algorithms

    |-> a01

    |-> a02