1.) Your team has just deployed a new build. The application starts crashing because it's running out of memory after several hours. The team suspects a memory leak has been introduced in one of the apps dependencies.
    a.) How would you help the web team diagnose and fix the problem?
    b.) How would you stop issues like this making its way into production?

2.) Describe some of the advantages and disadvantages of SQL and NoSQL data stores, and the general use cases you think they each fit.

3.) Describe the workflows, ecosystems, and technology you would use/create to ensure the dual goals of GDPR/privacy compliance and data accessibility for business/application needs.

4.) What's your pre and post-deployment checklist for a new application, inclusive of the entire stack (assume the infra/cloud provider of your choice)?

Engineering Exercise

Part 1 - The Web Service

Write a web service in any language that takes in a JSON payload, does some basic validation against an expected message format and content, puts that payload into a queue of your choice, and stores the relevant info in a relational DB.

Example valid payload:

```
{
	"ts": "1530228282",
	"sender": "testy-test-service",
	"message": {
		"foo": "bar",
		"baz": "bang"
	},
	"sent-from-ip": "1.2.3.4",
	"priority": 2

}
```

Validation rules:

- “ts” must be present and a valid Unix timestamp
- “sender” must be present and a string
- “message” must be present, a JSON object, and have at least one field set
- If present, “sent-from-ip” must be a valid IPv4 address
- All fields not listed in the example above are invalid, and should result in the message being rejected.

Part 2 - Describe how to run your application

Part 3 - Dockerize It (Please note this is a bonus section)

Write a Dockerfile to produce a container that will run your service, and one to spin up an instance of the queue you used in Part 1

Part 4 - Kubeify it (Please note this is a bonus section)

Write Kubernetes manifests to create a Service and Deployment object for both your service and the queue

Part 5 - Productionize It (Please note this is a bonus section)

Now that you have your service, describe/implement your approach to making it ready for production at scale (100k RPS).