DevOps Interview Tasks

Please create a publicly available repo at Github or Bitbucket, etc. to show your work.
Upload every relevant configs, settings, yaml manifests, dockerfiles, etc. that allows us to reproduce your solution. You may use any tool to accomplish the task, just be sure to document everything in a markdown formatted Readme file.
Use infrastructure as code (IaC) approach as much as possible.

Deploy Redis service into Kubernetes

Deploy a Redis service into a Kubernetes cluster with a sort of persistent storage. Feel free to use any Kubernetes implementation. We recommend you using Minikube and the free tier of any public docker registry.

Develop and deploy a Flask application

Develop a Flask application which can communicate with the Redis database. The application should have the following endpoints:

- /healthz: return the string "ok" (without quotes)
- /alert: increment a counter in redis
- /counter: print the value of the counter
- /version: return the short git commit hash

Deploy your Flask application into the Kubernetes cluster. Use liveness probe to let Kubernetes know whether your app is healthy.

Deploy and configure Prometheus and Grafana

Deploy Prometheus and Grafana service into your Kubernetes cluster. Create a Grafana dashboard with a chart showing the number of running "non-system" pods. We call a pod "non-system" if it runs in a namespace other than kube-system.
The title of the dashboard shall be "Pod stats", and the title of the chart shall be "Running # of non-system pods".
Make sure that the dashboard is visible from outside of Kubernetes, use an appropriate Service type.

Create an alert rule for Prometheus

Deploy and configure Alertmanager service for Prometheus. Configure an alert with the following condition: when there are at least 15 "non-system" pods. In case of an alert use a webhook that calls the /alert endpoint on the previously created Flask app.
Create an nginx deployment with an appropriate number or replicas to trigger an alert. Verify the alert in Prometheus and in the Flask app checking the /counter endpoint.
Finally scale down the nginx deployment to resolve the alert.
Create a screenshot of the Grafana dashboard where we can observe the change in the number of running "non-system" pods.