DevOps Engineer (Kubernetes) Challenge

Challenge: Simulated Peering

For this challenge, we will simulate peering between http servers using a Kubernetes operator.

1. Using your preferred programming language, create a simple http server. The http server should provide a `GET /ping` route. Invoking this route should respond "`pong\n`" back to the client. The server must be able to load a configuration containing a set of given hosts and ports that will be running the same type of http server. The server should invoke the other hosts' `GET /ping` route every Nth amount of time e.g. every 1 minute.

2. Create a Kubernetes custom resource definition (CRD) that will be used by a Kubernetes operator to create N number of pods running our http server along with the desired configuration needed for step 1.

3. Implement a Kubernetes operator to observe our CRD from step 2, and create the necessary resources. The pods can be standalone or managed using either the same or different Deployment, or Replicaset. Decide how pods can resolve each other e.g. ClusterIP or headless service etc.

4. The pod should run our http server and communicate with the other pods running the same type of http server. Make it obvious that pods are communicating via logging the response "`pong\n`" in the client's stdout along with the responding server's name or any sort of identifier.

5. Any changes to the CRD should reflect the resources managed by Kubernetes operator e.g. scaling the number of pods running our http servers from 5 to 1. For this challenge, it is okay to recreate or restart the pod to load the new configuration.

6. Submit your solution (zip, tarball, git repository link) along with documentation on how to test it. Please include comments and discussions about further improvements.

BONUS:

- On CRD update, pods are not restarted, but able to load the new configuration declared from the CRD. This means that the http server can determine if hosts should be added or removed as peers without restarting or recreating the pod.
- Docker and Kubernetes best practices e.g. QoS, security policies

TIPS:

- Use a framework to create a CRD or kubernetes operator e.g. https://sdk.operatorframework.io/
- https://www.techtarget.com/searchitoperations/tip/Learn-to-use-Kubernetes-CRDs-in-this-tutorial-example
- https://github.com/dot-i/k8s-operator-node
- https://www.magalix.com/blog/creating-custom-kubernetes-operators
- https://kubernetes.io/docs/concepts/extend-kubernetes/operator/
- **HAVE FUN!**