CS-548: Cloud-native Software Architectures

Computer Science Department • University of Crete
Instructor: Antony Chazapis

Website: https://www.csd.uoc.gr/~hy548/

Version: 202301

Assignment 3: Scaling

1. The example for autoscaling is available at GitHub (https://github.com/chazapis/hy548). Improve it so that:
    a. Instead of "Hello from Python Flask!", the flask-hello container will return the value of the `MESSAGE` environment variable when someone uses the service (use Python's `os.getenv`). Provide the new `Dockerfile` and `hello.py`. Build and upload the new container to Docker Hub.
    b. Provide two YAMLs to deploy the above container with all necessary resources (Deployment, Service, Ingress), so that "This is the first service!" is returned when someone visits the `/first` endpoint, and "This is the second service!" when someone visits the `/second`.
    c. Provide the commands needed to test the above two services with minikube (from running minikube, to `curl` or `wget` commands to use the services). Assume that the first deployment is in `first.yaml` and the second in `second.yaml`.

2. Following on from the previous exercise, extend the YAML that implements the `/first` endpoint:
    a. To limit each Pod to a maximum of 20% CPU and 256MB RAM.
    b. With a HorizontalPodAutoscaler manifest, which will increase the number of Pods in the Deployment when the average CPU usage exceeds 80%. Set a minimum of 1 Pod and a maximum of 8 for the Deployment.

    Run some http benchmark to find the maximum requests per second both services can handle, using 1 or 100 simultaneous clients. At how many containers does the scaling of the first service stop? Provide the new YAML, the test results, and a screenshot of the tool you used, or its output if it was a command line utility.

3. If you have enabled the ingress addon in minikube, remove it. Issue the commands to install the Ingress controller implemented with Nginx using Helm. You will find the chart at https://artifacthub.io/packages/helm/ingress-nginx/ingress-nginx. Try again the services of the above exercises. What changes are needed in the YAML files to make them work (if any)?

4. Create a Helm chart for the service that implements the `/first` endpoint of exercise 2. The chart should define variables for:
    a. The string to return.
    b. The endpoint to use for the service.

c. The CPU and memory limits of each Pod (optional, default is no limits).
d. The maximum number of Deployment Pods for the HorizontalPodAutoscaler (optional, default is 10).

Provide the chart files and the commands needed to install a service named "third" that will reply "This is a third service!", use the `/third` endpoint, be limited to 25% CPU (no memory limit), and scale automatically up to 20 Pods via the HorizontalPodAutoscaler.

Notes:

- The assignment is personal.
- All exercises contribute equally to the overall grade (unless individual percentages are defined).
- A day/time will be set for answering questions and giving clarifications.
- Write down your answers in a Markdown-formatted text file in either Greek or English and commit it (along with any other files) in a private GitHub repository before the exercise's deadline. Share the repository with the instructor (username "chazapis").