**Scaling workloads (45’)**

**Exercise 1 – define auto-scaling**

In this exercise, you must create a deployment using this image: stephaneey/simpleapi:dev. Create a busybox deployment with the image governmentpaas/curl-ssl and make sure to pass the following command to the container: ["/bin/sleep", "30d"]. At last, create a K8s service to expose the API on port 8080.

From the busybox, you should be able to call the API on port 8080 as follows:

curl -v http:<yourservice>.<yournamespace>.svc.cluster.local:8080/weatherforecast

Create an HorizontalPodAutoscaler with an average CPU usage of 30%, and make sure to scale simpleapi accordingly. Generate some load to see if the HPA scales out pods.

Tips:

* You will need 2 deployments and 1 service
* Make sure to define resource requests and limits on simpleapi
* You’ll need to exec in the busybox and run curl to test your API
* You can use while or for loops to generate load and look at the target HPA.

**Exercise 2 – Load testing AKS with … AKS**

In this exercise, you’ll use AKS virtual nodes to load test the simpleapi application. You can reuse the previously deployed HPA and service but you won’t need busybox anymore.

Tips:

* You’ll need an extra deployment that must be scheduled on a virtual node. Make sure to specify the right node selector and tolerations. You’ll use the azch/artillery load testing image. Look at artillery quick. More info is available here <https://www.artillery.io/docs/guides/guides/command-line>