



# Deadlines

Each sprint has a hard deadline. Sprints can be handed in earlier. If a deadline is missed the points are not awarded and the solution is uploaded to GitHub so the next sprint can be accomplished.

Furthermore, if you run out of budget on your Exoscale account a **-15% penalty** will be incurred. Make sure you monitor your Exoscale account. (Please contact us if you require more budget.)

## Sprint 1: Instance Pools

**Deadline:** 15<sup>th</sup> of October 2020

In this sprint you must demonstrate your ability to set up an instance pool and a network load balancer.

### Pass criteria:

#### MANUAL SOLUTION (5%)

You must demonstrate on a call how you set up your Exoscale environment in detail. Otherwise, the requirements are identical as with the automated solution.

#### AUTOMATED SOLUTION (15%)

You must upload your Terraform to GitHub or an alternative Git service and submit your link in Moodle. Your code must accept two input variables, `exoscale_key` and `exoscale_secret`. The automated checking system will run your code against a completely empty Exoscale account from a standard Ubuntu 20.04 environment. The automated system will check the following:

- The Terraform code must execute successfully,
- The presence of a single NLB with a single service listening on port 80,
- The presence of a single instance pool,
- That all backends on the NLB are green within 5 minutes of running Terraform,
- When deleting all instances from the instance pool, the instance pool must provision the new instances and the health check must become green within 5 minutes,
- When accessing the IP address of the NLB with the `/health` URL, that URL should respond with "OK" and a HTTP status code 200. (The easiest way to achieve this is to run the [http load generator](#).)

## Sprint 2: Monitoring

**Deadline:** 15<sup>th</sup> of November 2020

In this sprint you must demonstrate your ability to monitor a varying number of instances set up on an instance pool in the previous sprint.

**Pass criteria:** Your monitoring instance must be set up with Prometheus and Grafana running. Prometheus must track the instances in the instance pool using custom service discovery and collect their CPU usage in a graph in Grafana.

## Sprint 3: Autoscaling

**Deadline:** 15<sup>th</sup> of December 2020

In this sprint you must demonstrate your ability to receive a webhook from Grafana and adjust the number of instances in the instance pool from the previous sprint under load.

**Pass criteria:** Your service will be hit with a large number of requests and it must scale up as outlined in the [project work document](#).