Project work

You are the cloud architect for a small work-for-hire company. A client wants to hire you but they are sceptical about your abilities to build an *autoscaling* service. They propose a proof of concept: build a service that runs a web application that is *deliberately slow* and load test it. Your cloud management should automatically launch new cloud servers when the load is high and remove servers when demand is low.

> **Warning**
>
> Do not continuously run your setup on Exoscale or you will run out of budget! Budget limitations are part of building cloud system so you have to learn to deal with it.

After taking a look at the capabilities of the cloud provider and discussing the constraints with your colleagues you decide that the following approach would be best:

- You are going to use Terraform to automate the setup and tear down of the cloud infrastructure. This is necessary because if you continuously use the cloud service you will not fit in the budget.
- You will use instance pools to manage the variable number of cloud servers and Network Load Balancers to balance the traffic between them.
- You will set up a dedicated monitoring and management instance which will run Prometheus to automatically monitor a varying number of servers. You will write a custom service discovery agent that creates a file with the IP addresses of the machines in the instance pool for Prometheus to consume.
- On the instance pool you will deploy the Prometheus node exporter to monitor CPU usage.
- You will install Grafana to provide a monitoring dashboard and the ability to send webhooks.
- You will configure an alert webhook in Grafana that sends a webhook to an application written by you. If the average CPU usage is above 80%, or below 20% to scale up or down respectively a webhook is sent.
- You will write an application that receives this webhook and every 60 seconds scales the instance pool up or down if a webhook has been received.

As you also have to demonstrate to the client that you can work in an agile methodology you agree in 4 week sprints with a demo at the end of each sprint as outlined in the deadlines document.

As a dummy service to generate load you will use http-load-generator.

Optionally, you can make use of the following (incurs a 5% point-penalty each):

- prometheus-sd-exoscale-instance-pools to feed instance pool data into Prometheus
- exoscale-grafana-autoscaler to drive the autoscaling behavior.