Assignment 3 - A not so simple service

The task is to design a cloud service from your field of expertise. This is your course project. For example, how can we make control-as-a-Service or network-simulation-on-demand? You should get together in the groups and decide on what should be an interesting service to have.
One possibility is to reuse an existing single server application and just place it in a cloud context and provide it as a service. For example, assume that you are really into quantum computing and use QISkit for your simulations. How would such a service look like in a cloud setting? How is multitenancy and isolation handled? How and when do we scale it? If you are into control simulation, you might want to think of running Julia as a service? Or maybe build your own Platform-as-a-Service for robotics workloads.

The important thing is for you to consider what are the design goals and the performance indicators. How will your application sustain loss of nodes and how is recovery made? Elasticity is an integral part of cloud computing and your application should address that to. How will your feedback look for autoscaling look like? Will there be any possibilities to cache results for faster response times?

The size of the assignment is typically one or two pages long and provides both an architectural overview, i.e. what are the building blocks, as well as a description of interaction patterns, i.e. who calls whom and when.

Please, also include some kind of work-load generator to facilitate testing, as well as a chaos animal of choice to allow for testing of the resilience to faults.

Once the design is presented, you will go off and implement it. So, keep it simple☺

Besides getting started with your course project you should also setup your own Kubernetes environment. Don't worry, we have prepared a Docker container that does most of the work for you. All you have to do is follow the instructions. All of them.

https://gitlab.datahub.erdc.ericsson.net/Eker/kubernetes-sandbox

Next week we will start to play with Kubernetes, so make sure to have your cluster up by then.