

Guidelines for Cloud and Fog Computing

Course Lab Project - 2021

1 Intro

1.1 Rationale

- Give you an opportunity to play with OpenStack and Kubernetes using something that you already know or want to play with.
- Give us an opportunity to evaluate your understanding of IaaS and PaaS Lab.

2 Architecture

2.1 Project architecture draft

- The project architecture/description should have maximum length of three A4 pages, including tables, figures/diagrams, and references. You do not need to fill the entire pages.
- This draft will serve as a plan for you and not as evaluation means for us.

2.2 Project draft deadline

- The deadline is 4 May at 12:00 noon CEST. So we can give you some feedback before/during/after the class of 5 May.
- Upload the file on the course site assignment.

2.3 Project guidelines

We expect you to mention usage of:

- IaaS
 - Users
 - Projects (in the OpenStack sense)
 - Images
 - Flavors
 - Instances
 - Ssh Keys
 - Networks, Floating IPs
 - Volumes

- PaaS
 - Docker containers
 - Docker container images
 - Pods
 - ReplicaSets
 - Deployments
 - Services
 - Volumes
 - Namespaces
 - Labels & Selectors
- ... any other concept/service of cloud and fog computing

We expect that you will detail a bit the idea of your presentation. Creating from scratch your project live will probably take more than the allowed presentation time. Think about simulating a failure or showing some automation for update, rollback, recover of your services.

2.4 Project draft mandatory fields/format

- The document must contain:
 - Course: Fog and Cloud Computing 2021
 - Architecture: `${PROJECT_TITLE:=Multi deployment of the Cheshire Cat}`
 - Group Number: `${X:=01}`
 - Authors:
 - * `${STUD1:=P. Liddell, Alice W. <a@example.com>}`
 - * `${STUD2:=White, Rabbit <rabbit@example.net>}`
- The document must be named with group name and slug title without spaces: `${X}-arch-${PROJECT_SLUG:=multi-deployment-cheshire-cat}.pdf`
- The PDF must be in A4 format and with a maximum length of 3 pages.

2.5 Project draft acknowledgement

- We will accept your project or make few suggestions by 5 May.
- You should take note of the suggestions for the final presentation.
- Usually, you do not need to resubmit the architecture document.

3 Implementation

3.1 Project mandatory steps

- For clarity we will call the machines created into your OpenStack Nested VM (NVMs) or instances.
- We do not judge the application you will use, but at least your deployment must be composed by two services (eg: database and web-

server) or two processes (distributed machine learning) running on different NVMs/Containers/Clusters.

- Your project must use at least one of the following technologies: OpenStack or Kubernetes. Of course you may choose to use both.
- Create on your VM (both IaaS and PaaS) an `eval` user named `eval`.
- Add an `eval` user with `eval` password to your custom OpenStack project for us that has read only access to your project.
- Create an private and public key for the `eval` user. Make sure that the `eval` user can ssh into all your NVM.
- Create a Kubernetes cluster (if any) in the PaaS VM using the `eval` user, we expect to find the KUBECONFIG file to access the cluster on this standard location: `/home/eval/.kube/kind-config-eval`.
- This course gave you the building blocks for deploying applications, is up to you to use them in a meaningful way. We suggest to use this freedom to make a project using/adapting previous software you developed for other courses, or for example to setup an environment that will help you in your future courses/projects.
- If you do not have an application ready to deploy you are free to take inspiration from the <https://github.com/awesome-selfhosted/awesome-selfhosted> repository.
- If you start from already existing recipes, keep track of the original source and your changes using your preferred source code management system.
- We do not want to list examples in order to not limit your fantasy.

3.2 Tips

- Everything can be automated. Ideally each step of your project should be automated, but we do not enforce that. Strive for a balance.
- For the automation you can use, in order of flexibility, cloud-init, scripting with the command line, programming with the APIs, using a deployment management tool like ansible, terraform... your own.
- You are encouraged to keep your project under revision control, but you are not enforced to do so.

3.3 Project limitation

- Do not use proprietary or confidential applications or applications with licences that prohibits you and us to run on our servers.
- High load application are fine, but unsuitable for our existing infrastructure, so limit their running time.
- Is up to you to choose a meaningful application that respect the hardware limits.

3.4 Project presentation

- On 8 or 9 June (we will make a pool for the allocations) you will present the running application in 5 minutes, and then we (and maybe the audience) will question you for 5 minutes about your project.

- We expect both member to speak during the presentation.
- We suggest you to use at max two optional slides: one for a diagram of the deployment, one for explaining the demo.
- Name the file `${X}-slides-${PROJECT_SLUG}.pdf`
- The deadline for sending the slides is 7 June at 12 noon CEST.

3.5 Project presentation report

- You may attach a short report (max 2 pages, in the same format of the draft) detailing the difference between the planned architecture and the real implementation.
- Change the title in report as:
 - Report: `${PROJECT_TITLE}`
 - Name the file as `${X}-report-${PROJECT_SLUG}.pdf`
- The deadline for sending the report is 7 June at 12 noon CEST.

4 Project evaluation

- We will evaluate, at the end of the project, your ability to use the OpenStack and Kubernetes platform to develop or test or deploy your sample application.
- For example: correct usage of networking, flavors, volumes, OpenStack API, Kubernetes commands...
- We may ask to change something in the project live, and ask you to explain the consequences.

4.1 Evaluation parameters

- Effective utilisation of the technologies (OpenStack and Kubernetes)
- Project presentation
- Answers to questions
- Completion and complexity of the project
- Scripting and automation

□