Cloud Computing and Services

Table of contents

COURSE WORKSHOP

"Cloud Computing and Services"

EXERCISE 2024

The exercise involves creating a virtual laboratory (vlab) environment *docker* which will automatically run the functions it has.

1. Create a virtual lab

Each vlab will have:

ÿ own network

ÿ environment for viewing the results: Graphical User Interface (GUI)/Command Line Interface (CLI)

ÿ Howto storage

ÿ Automatic service execution process

ÿ one service per container

ÿ resource config for Out Of Memory Exceptions (OOME)

> ÿ Templates: 1. dockerfile-examples 2. Dockerizing app 3. Multiservices Application

2. Delivery of work

Delivery of work:

ÿ The text of the assignment will be in **asciidoc** format

ÿ It will contain **a Dockerfile** for creating the images

ÿ It will contain **a README** file with information about the virtual lab and the services/tools used

ÿ It will contain **an INSTALL** file with information about the virtual installation laboratory

ÿ

Information

ÿ The assignment should be prepared individually or in groups of two (2) students.

The texts will **all** be in asciidoc or markdown format

ÿ

Git

All files should be "uploaded" to a git service

- https://github.com/
- https://about.gitlab.com/
- https://codeberg.org

ÿ

It is not necessary to include personal information in your repository. You can use it exclusively for your tasks and nothing else, thus protecting your privacy.

3. Projects

Each group should choose to implement one of the following proposed **projects** *(they can also propose one of their own as long as the basic architecture/ design guidelines shown in the examples-exercise of the workshop are followed)*

3.1. Katex wikipedia

- Katex - https://en.wikipedia.org/wiki/KaTeX, https://en.wikipedia.org/wiki/Mathematical_notation

  ÿ https://github.com/KaTeX/KaTeX

  ÿ https://katex.org/#demo

3.2. wordpress

- wordpress

  ÿ https://github.com/gabidavila/docker-wordpress-ssl-nginx-mysql

3.3. mongo replicated

- mongo replicated - https://en.wikipedia.org/wiki/MongoDB, https://github.com/mongodb/mongo/wiki/Replication-Internals , https://en.wikipedia.org/wiki/NoSQL ÿ https://www.xosofox.de/2017/06/running-mongodb-as-a-replicaset-in-docker/

ÿ https://www.sohamkamani.com/blog/2016/06/30/docker-mongo-replica-set/#setting-up-replication

3.4. mysql replicated

- mysql replicated - https://en.wikipedia.org/wiki/Multi-master_replication

ÿ https://github.com/gritt/docker-mysql-replication

ÿ https://hub.docker.com/r/actency/docker-mysql-replication

3.5. asciidoc + online editor

- asciidoc + online editor - https://en.wikipedia.org/wiki/AsciiDoc, https://en.wikipedia.org/wiki/DocBook

ÿ https://github.com/bodiam/awesome-asciidoc

ÿ https://github.com/asciidoctor/docker-asciidoctor

3.6. collaborative LaTeX editor

- collaborative LaTeX editor - https://en.wikipedia.org/wiki/LaTeX

ÿ https://github.com/overleaf/overleaf

3.7. Realtime database backend

- Realtime database backend - https://en.wikipedia.org/wiki/Operational_transformation

ÿ https://github.com/share/sharedb

3.8. collaborative editor (like google docs)

- collaborative editor - https://en.wikipedia.org/wiki/CodeMirror

ÿ https://github.com/hectorj2f/codemirror-dockerhttps://github.com/jitsi/docker-jitsi-meet

ÿ https://codemirror.net/

ÿ https://codemirror.net/mode/dockerfile/

3.9. hadoop

- hadoop - https://en.wikipedia.org/wiki/Apache_Hadoop

ÿ https://github.com/big-data-europe/docker-hadoop

ÿ https://hub.docker.com/r/sequenceiq/hadoop-docker/

ÿ https://www.alibabacloud.com/blog/setup-a-single-node-hadoop-cluster-using-docker_595278

3.10. collaborative eclass meeting chat

- collaborative eclass meeting chat - https://en.wikipedia.org/wiki/Jitsi

  ÿ https://github.com/jitsi/docker-jitsi-meet

3.11. collaboration solution

- collaboration solution - https://en.wikipedia.org/wiki/Mattermost

  ÿ https://mattermost.com/mattermost-vs-slack/

  ÿ https://github.com/mattermost/mattermost-docker

  ÿ https://hub.docker.com/r/jasl8r/mattermost

3.12. R and Octave (statistical computing and graphics)

- R & Octave - https://en.wikipedia.org/wiki/R_(programming_language), https://en.wikipedia.org/wiki/GNU_Octave

  ÿ https://github.com/INWTlab/r-docker

  ÿ https://www.r-project.org/

  ÿ https://www.gnu.org/software/octave/

3.13. Ruby, php and Node.js

- Ruby, php and Node.js

  ÿ https://hub.docker.com/r/ethanw93/docker-ruby-php-node/

  ÿ https://github.com/dsferruzza/docker-php-node-ruby-build/blob/master/Dockerfile

3.14. java

- java and scala

  ÿ https://github.com/docker-java/docker-java

  ÿ https://www.scala-lang.org/documentation/your-first-lines-of-scala.html

  ÿ https://github.com/hseeberger/scala-sbt

3.15. phonegap cordova

- phonegap cordova - https://en.wikipedia.org/wiki/Apache_Cordova

  ÿ https://hub.docker.com/r/webratio/phonegap/

  ÿ https://github.com/idotta/docker-phonegap

  ÿ https://github.com/oren/docker-cordova

  ÿ https://github.com/walterwhites/docker-cordova

3.16. visualization software

- Graphviz + PlantUML - https://en.wikipedia.org/wiki/PlantUML, https://en.wikipedia.org/wiki/Unified_Modeling_Language, https://en.wikipedia.org/wiki/Graphviz

  ÿ https://github.com/plantuml/plantuml-server

  ÿ https://github.com/emfloyd2/docker-graphviz

  ÿ https://github.com/omerio/graphviz-server

3.17. collaboration platform

- Nextdloud (similar to Dropbox, Office 365 or Google Drive) - https://en.wikipedia.org/wiki/nextcloud,

  ÿ https://github.com/nextcloud/docker/tree/63438ef792fdedd4ceb80664d22391aca21f6bd1

  ÿ https://github.com/nextcloud/docker

3.18. perl

- https://git.swarmlab.io:3000/swarmlab/microservice-perlfull

3.19. rust

- https://git.swarmlab.io:3000/swarmlab/microservice-rust

3.20. ruby

- https://git.swarmlab.io:3000/swarmlab/microservice-ruby

3.21. r

- https://git.swarmlab.io:3000/swarmlab/microservice-r

3.22. opencv

- https://git.swarmlab.io:3000/swarmlab/microservice-opencv

3.23. octave

- https://git.swarmlab.io:3000/swarmlab/microservice-octave

3.24. numpy

- https://git.swarmlab.io:3000/swarmlab/microservice-numpy

3.25. nodejs

- https://git.swarmlab.io:3000/swarmlab/microservice-nodejs

3.26. vue

- https://git.swarmlab.io:3000/swarmlab/microservice-vue

3.27. jupyter

- https://git.swarmlab.io:3000/swarmlab/microservice-jupyter26

3.28. etherpad

- https://git.swarmlab.io:3000/swarmlab/microservice-etherpad

3.29. electron

- https://git.swarmlab.io:3000/swarmlab/microservice-electron

3.30. xelatex

- https://git.swarmlab.io:3000/swarmlab/microservice-xelatexthesis

3.31. nodesql3

- https://git.swarmlab.io:3000/swarmlab/microservice-nodesqlite3

3.32. redis

- https://git.swarmlab.io:3000/swarmlab/microservice-redisserver

3.33. mongoserver & mongooexpress

- https://git.swarmlab.io:3000/swarmlab/microservice-mongoserver, https://git.swarmlab.io:3000/swarmlab/microservice-mongoexpress

3.34. mariadb

- https://git.swarmlab.io:3000/swarmlab/microservice-mariadb

3.35. nginx

- https://git.swarmlab.io:3000/swarmlab/microservice-nginx

3.36. haproxy

- https://git.swarmlab.io:3000/swarmlab/microservice-haproxy

3.37. jsshell

- https://git.swarmlab.io:3000/swarmlab/microservice-jsshell

3.38. alpine

- https://git.swarmlab.io:3000/swarmlab/microservice-alpine315

3.39. ubuntu

- https://git.swarmlab.io:3000/swarmlab/microservice-ubuntu

4. Q&A

1. *Is the project implemented in*

   groups? Try to join a group and let us know via email which group you will be joining. If this is not possible, you can also do it yourself.

2. *Can I implement another project outside the list?*

   Yes. As long as **it is not a "stand alone application".** The project must be implemented according to the specifications described for the rest.

3. *I don't know asciidoc will it take time to learn it?*

   No, it's simple. You won't need more than 5 minutes and in case you want something even simpler, write in markdown - online Editor: markdown, asciidoc

4. *I don't know git will it take time to learn it?*

   No, it's simple. It won't take you more than 10 minutes - Help: Git In Five Minutes