

Υπολογιστική Νέφους και Υπηρεσίες

Πίνακας περιεχομένων

1. Δημιουργία εικονικού εργαστηρίου	2
2. Παράδοση εργασίας	2
3. Projects	3
3.1. Katex wikipedia	3
3.2. wordpress	3
3.3. mongo replicated	3
3.4. mysql replicated	4
3.5. asciidoc + online editor	4
3.6. collaborative LaTeX editor	4
3.7. Realtime database backend	4
3.8. collaborative editor (like google docs)	4
3.9. hadoop	4
3.10. collaborative eclass meeting chat	5
3.11. collaboration solution	5
3.12. R and Octave (statistical computing and graphics)	5
3.13. Ruby, php and Node.js	5
3.14. java	5
3.15. phonegap cordova	5
3.16. visualization software	6
3.17. collaboration platform	6
3.18. perl	6
3.19. rust	6
3.20. ruby	6
3.21. r	6
3.22. opencv	6
3.23. octave	6
3.24. numpy	6
3.25. nodejs	7
3.26. vue	7
3.27. jupyter	7
3.28. etherpad	7
3.29. electron	7
3.30. xelatex	7
3.31. nodesql3	7

3.32. redis	7
3.33. mongoserver & mongoexpress	7
3.34. mariadb	7
3.35. nginx	7
3.36. haproxy	8
3.37. jsshell	8
3.38. alpine	8
3.39. ubuntu	8
4. Q&A	8

ΕΡΓΑΣΤΗΡΙΟ ΜΑΘΗΜΑΤΟΣ

«Υπολογιστική Νέφους και Υπηρεσίες»

ΑΣΚΗΣΗ 2024

Η άσκηση περιλαμβάνει τη δημιουργία ενός εικονικού εργαστηρίου(vlab) σε περιβάλλον *docker* που θα εκτελεί αυτόματα τις λειτουργίες που διαθέτει.

1. Δημιουργία εικονικού εργαστηρίου

Κάθε vlab θα διαθέτει:

- ☑ δικό του δίκτυο
- ☑ περιβάλλον για προβολή των αποτελεσμάτων: Graphical User Interface (GUI)/Command Line Interface (CLI)
- ☑ χώρο αποθήκευσης [Howto](#)
- ☑ Αυτόματη διαδικασία εκτέλεσης των υπηρεσιών
- ☑ μία υπηρεσία ανά container
- ☑ resource config για Out Of Memory Exceptions ([OOMF](#))



Templates: 1. [dockerfile-examples](#) 2. [Dockerizing app](#) 3. [Multiservices Application](#)

2. Παράδοση εργασίας

Παράδοση εργασίας:

- ☑ Το κείμενο της εργασίας θα είναι σε μορφή **asciidoc**
- ☑ Θα περιέχει **Dockerfile** για την δημιουργία των images
- ☑ Θα περιέχει **README** file με πληροφορίες σχετικά με το εικονικό εργαστήριο και τις

υπηρεσίες/εργαλεία που χρησιμοποιήθηκαν

- ☑ Θα περιέχει **INSTALL** file με πληροφορίες σχετικά με την εγκατάσταση του εικονικού εργαστηρίου



Πληροφορία



Η εργασία να εκπονηθεί κατά μονάς ή σε ομάδες των δύο (2) φοιτητών/φοιτητριών.

Το κείμενα θα είναι **όλα** σε μορφή asciidoc ή markdown



Git

Όλα τα αρχεία θα πρέπει να "ανέβουν" σε υπηρεσία git

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



Δεν είναι απαραίτητο να περιλαμβάνετε προσωπικές πληροφορίες στο repository σας. Μπορείτε να το χρησιμοποιήσετε αποκλειστικά για τις εργασίες σας και τίποτα περισσότερο, προστατεύοντας έτσι την ιδιωτικότητά σας.

3. Projects

Κάθε ομάδα θα πρέπει να επιλέξει προς υλοποίηση ένα από τα κάτωθι προτεινόμενα **projects** (μπορεί επίσης να προτείνει και κάποιο δικό της αρκεί να τηρούνται οι βασικές κατευθύνσεις αρχιτεκτονικής/σχεδιασμού που εδείχθησαν στα παραδείγματα-άσκηση του εργαστηρίου)

3.1. Katex [wikipedia](https://en.wikipedia.org/wiki/KaTeX)

- Katex - <https://en.wikipedia.org/wiki/KaTeX>, https://en.wikipedia.org/wiki/Mathematical_notation
 - <https://github.com/KaTeX/KaTeX>
 - <https://katex.org/#demo>

3.2. wordpress

- wordpress
 - <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>

- <https://www.xosofox.de/2017/06/running-mongodb-as-a-replicaset-in-docker/>
- <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
 - <https://github.com/gritt/docker-mysql-replication>
 - <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>
 - <https://github.com/bodiam/awesome-asciidoc>
 - <https://github.com/asciidoctor/docker-asciidoctor>

3.6. collaborative LaTeX editor

- collaborative LaTeX editor - <https://en.wikipedia.org/wiki/LaTeX>
 - <https://github.com/overleaf/overleaf>

3.7. Realtime database backend

- Realtime database backend - https://en.wikipedia.org/wiki/Operational_transformation
 - <https://github.com/share/sharedb>

3.8. collaborative editor (like google docs)

- collaborative editor - <https://en.wikipedia.org/wiki/CodeMirror>
 - <https://github.com/hectorj2f/codemirror-docker><https://github.com/jitsi/docker-jitsi-meet>
 - <https://codemirror.net/>
 - <https://codemirror.net/mode/dockerfile/>

3.9. hadoop

- hadoop - https://en.wikipedia.org/wiki/Apache_Hadoop
 - <https://github.com/big-data-europe/docker-hadoop>
 - <https://hub.docker.com/r/sequenceiq/hadoop-docker/>
 - 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>
 - <https://github.com/jitsi/docker-jitsi-meet>

3.11. collaboration solution

- collaboration solution - <https://en.wikipedia.org/wiki/Mattermost>
 - <https://mattermost.com/mattermost-vs-slack/>
 - <https://github.com/mattermost/mattermost-docker>
 - <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/R_(programming_language)), https://en.wikipedia.org/wiki/GNU_Octave
 - <https://github.com/INWTLab/r-docker>
 - <https://www.r-project.org/>
 - <https://www.gnu.org/software/octave/>

3.13. Ruby, php and Node.js

- Ruby, php and Node.js
 - <https://hub.docker.com/r/ethanw93/docker-ruby-php-node/>
 - <https://github.com/dsferruzza/docker-php-node-ruby-build/blob/master/Dockerfile>

3.14. java

- java and scala
 - <https://github.com/docker-java/docker-java>
 - <https://www.scala-lang.org/documentation/your-first-lines-of-scala.html>
 - <https://github.com/hseeberger/scala-sbt>

3.15. phonegap cordova

- phonegap cordova - https://en.wikipedia.org/wiki/Apache_Cordova
 - <https://hub.docker.com/r/webratio/phonegap/>
 - <https://github.com/idotta/docker-phonegap>
 - <https://github.com/oren/docker-cordova>
 - <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>
 - <https://github.com/plantuml/plantuml-server>
 - <https://github.com/emfloyd2/docker-graphviz>
 - <https://github.com/omerio/graphviz-server>

3.17. collaboration platform

- Nextcloud (similar to Dropbox, Office 365 or Google Drive) - <https://en.wikipedia.org/wiki/Nextcloud>,
 - <https://github.com/nextcloud/docker/tree/63438ef792fdedd4ceb80664d22391aca21f6bd1>
 - <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 & mongoexpress

- <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. Το project υλοποιείται σε ομάδες?

Προσπαθήστε να ενταχθείτε σε μια ομάδα και ενημερώστε μέσω email σε ποια ομάδα θα ενταχθείτε. Εάν αυτό δεν είναι δυνατόν, μπορείτε επίσης να το κάνετε μόνοι σας.

2. Μπορώ να υλοποιήσω κάποιο άλλο project εκτός λίστας?

Ναι. Αρκεί να μην είναι **"stand alone application"**. Το project πρέπει να υλοποιηθεί σύμφωνα με τις προδιαγραφές που περιγράφονται και για τα υπόλοιπα.

3. Δεν γνωρίζω asciidoc θα χρειαστεί χρόνο για να το μάθω?

Οχι είναι απλό. Δεν θα χρειαστείς πάνω από 5 λεπτά και σε περίπτωση που θέλεις κάτι ακόμη πιο απλό γράψε σε markdown - online Editor: [markdown](#), [asciidoc](#)

4. Δεν γνωρίζω git θα χρειαστεί χρόνο για να το μάθω?

Οχι είναι απλό. Δεν θα χρειαστείς πάνω από 10 λεπτά - Βοήθεια: [Git In Five Minutes](#)