

Υπολογιστική Νέφους και Υπηρεσίες

Πίνακας περιεχομένων

1. Δημιουργία εικονικού εργαστηρίου	1
2. Παράδοση εργασίας	2
3. Projects	2
3.1. Data collector service	3
3.2. Realtime database backend	3
3.3. Fediverse system	3
3.4. Distributed file system	3
3.5. Object Storage	4
4. Q&A	4

ΕΡΓΑΣΤΗΡΙΟ ΜΑΘΗΜΑΤΟΣ

«Υπολογιστική Νέφους και Υπηρεσίες»

ΑΣΚΗΣΗ 2021

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

1. Δημιουργία εικονικού εργαστηρίου

Κάθε vlab θα διαθέτει:

- δικό του δίκτυο
- περιβάλλον (GUI/CLI) για προβολή των αποτελεσμάτων
- χώρο αποθήκευσης!^[1]
- Αυτόματη διαδικασία εκτέλεσης των υπηρεσιών (π.χ. μέσω `make` - Παράδειγμα)
- μία υπηρεσία ανά container!^[2]
- resource config για Out Of Memory Exceptions (OOM), [Howto](#)

Πληροφορία

Για την υλοποίησή σας μπορείτε να στηριχτείτε στη βασική αρχιτεκτονική-σχεδιασμό και παραδείγματα που επιδείχθηκαν στο Εργαστήριο



Reminder

[not-supported-for-docker-stack-deploy](#)

2. Παράδοση εργασίας

Παράδοση εργασίας:

- ☑ Το κείμενο της εργασίας θα είναι σε μορφή **asciidoc** ή markdown
- ☑ Θα περιέχει **Yaml File** για Deploy της υπηρεσίας
- ☑ 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
- ☑ Θα περιέχει **DOC** file/dir με πληροφορίες σχετικά με την χρήση του εικονικού εργαστηρίου
- ☑ Θα περιέχει **Demo** για την εκτέλεση των διαθέσιμων υπηρεσιών



Πληροφορία

Το κείμενα θα είναι **όλα** σε μορφή asciidoc ή markdown ([Asccidoc](#), [Markdown](#))

Git

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



- [swarmlab](#) - Προτείνεται
- [gitlab](#)
- [github](#)

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)



Επιλογή

Η επιλογή project ανα ομάδα πρέπει να πραγματοποιηθεί μέχρι **10.04**

3.1. Data collector service

- Swarmlab
 - Swarmlab-hybrid, Menu: **hybrid-linux** → **courses/fluentd**
 - [git](#)
 - [video1](#)
 - [video2](#)

Το project "*Data collector service*" αποτελεί τη βασική επιλογή για την εξέταση του εργαστηρίου και χωρίζεται σε 3 ενότητες:



- Πρώτη
 - Δημιουργία δικτύου
 - Αυτόματη εγκατάσταση προγράμματος για την συλλογή των δεδομένων
 - onEvent - τοπική/προσωρινή αποθήκευση των δεδομένων
- Δεύτερη
 - Δημιουργία βάσης δεδομένων(Database replication)
 - onEvent αποθήκευση των δεδομένων στην βάση
- Τρίτη
 - Δημιουργία GUI ή
 - Δημιουργία Websocket σύνδεσης με την βάση ή με τις βασικές υπηρεσίες για άμεση μεταφορά των δεδομένων

3.2. Realtime database backend

- Realtime database backend - https://en.wikipedia.org/wiki/Operational_transformation
 - [git](#)

3.3. Fediverse system

- Pleroma - [Wikipedia](#)
 - [docs](#)

3.4. Distributed file system.

- IPFS Cluster - [Wikipedia](#)
 - [git](#)

3.5. Object Storage

- Minio s3 (amazon s3) - [Wikipedia](#)
 - [git](#)

4. Q&A

1. *Το project υλοποιείται σε ομάδες?*

Ναι.

2. *Εγώ δεν βρήκα ομάδα. Τι κάνω?*

Προσπάθησε να ενταχθείς σε μία ομάδα και ενημέρωσε μέσω email για την ομάδα που θα συμμετέχεις.

3. *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.

4. *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](#)

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

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

[1] Σχετικά με την υποστήριξη 'χώρου αποθήκευσης', μπορείτε να χρησιμοποιήσετε την υπηρεσία storage-mongo-replica του swarmlab.io ([git](#)) ή όποια άλλη βάση επιθυμείτε

[2] Γενικά διαχωρίζουμε τους τομείς χρησιμοποιώντας μία υπηρεσία ανά container. Αυτή η υπηρεσία μπορεί, σε διάφορες περιπτώσεις, να εκτελεστεί σε πολλές διεργασίες (για παράδειγμα, ο διακομιστής web Apache ξεκινά πολλές διεργασίες).