

## Φάση 2η

Γεώργιος Ραμιώτης 2018030091

*Υπηρεσίες στο Υπολογιστικό Νέφος και την Ομίχλη,  
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών,  
Πολυτεχνείο Κρήτης*

Χειμερινό Εξάμηνο 2022-2023

# Δικτύωση εφαρμογής

Αρχικά, η "dockerized" εφαρμογή αποτελείται από τα εξής container:

- **php-apache-environment** Για τον σέρβερ Apache ο οποίος θα φιλοξενήσει την εφαρμογή
- **mysql\_db** Για την βάση δεδομένων του Keyrock
- **keyrock** Η υπηρεσία Fiware Keyrock-IDM για την αυθεντικοποίηση των χρηστών
- **phpmyadmin** Για την γραφική παρουσίαση των δεδομένων του Keyrock
- **mongo** Για την βάση δεδομένων που θα είναι αποθηκευμένα τα προϊόντα, τα καλάθια αγορών των χρηστών, ένα αντίγραφο των subscriptions καθώς και όλα τα notifications που παράγονται από τον Orion
- **data-storage** Για τον Apache σέρβερ που θα φιλοξενήσει την REST-API διεπαφή για την επικοινωνία της εφαρμογής με την βάση δεδομένων MongoDB των προϊόντων
- **dss-proxy** Η υπηρεσία Fiware PEP-Proxy για την προστασία του data-storage
- **orion** Η υπηρεσία Fiware Orion Context-Broker για την λειτουργία PUB-SUB της εφαρμογής
- **orion-proxy** Μια ακόμα PEP-Proxy για την προστασία του Orion
- **orion-mongo** Και μια ακόμη βάση MongoDB για την λειτουργία του Orion

Για την δικτύωση της εφαρμογής κατασκευάστηκε ένα προσαρμοσμένο δίκτυο στην IP διεύθυνση και subnet **172.18.1.0/24** χρησιμοποιώντας την ενσωματωμένη λειτουργία του docker το docker network.

Όλα τα container της εφαρμογής εγκαθίστανται με τα έτοιμα images των υπηρεσιών. Επίσης, έχουμε κάποιες παραμετροποιήσεις στα container των php-apache

και data storage όπου έχει οριστεί ένα dockerfile για την εγκατάσταση και ενεργοποίηση των plugins της php για την επικοινωνία με mysql και mongodb βάσεις δεδομένων.

Η επικοινωνία μεταξύ των container και ο ορισμός των ports στα οποία θα λειτουργούν καθώς και τα docker volumes για την μόνιμη αποθήκευση των δεδομένων των βάσεων έχουν οριστεί στο μοναδικό docker-compose αρχείο.

Τέλος, η αρχιτεκτονική της εφαρμογής και ο ορισμός των ports του κάθε container βασίστηκαν στο σχεδιάγραμμα που παρείχε η εκφώνηση της 2ης φάσης του πρότζεκτ.

## **Πρόσθετες πληροφορίες για την λειτουργία των container**

### **Keyrock**

Παρατηρήθηκε ότι ο Keyrock δεν επιτρέπει με την χρήση της διεπαφής του την μεταβολή της τιμής του πεδίου admin και την απενεργοποίηση ενός χρήστη με το πεδίο enabled. Έτσι, η μεταβολή αυτών των τιμών γίνεται μέσω mysql queries κατευθείαν στην mysql βάση του Keyrock.

Επιπλέον, έγινε η σύμβαση η εφαρμογή να μην επιτρέπει την εγγραφή χρηστών στον Keyrock με ίδιο email και username, οπότε αυτά τα πεδία είναι μοναδικά για κάθε χρήστη.

Τέλος, έγινε η παρακάτω αντιστοίχιση των πεδίων του keyrock με την πληροφορία που χρειαζόμαστε για τον χρήστη:

- Keyrock ID → ID του χρήστη
- username → Username
- description → Fullname
- website → Role
- email → Email
- password → Password (Automatically encoded password from Keyrock)

- enabled → Confirmed
- admin → Admin Privileges του χρήστη με τον ρόλο ADMIN

## PEP Proxy Wilma

Η εγγραφή των proxies έγιναν μέσω του keyrock και η πρόσβαση σε αυτές γίνονται με το μοναδικό για κάθε χρήστη **OAuth2.0 token** που το παρέχει ο keyrock με ένα request χρησιμοποιώντας το client id και client secret του keyrock.

## Orion και Data Storage

Η λογική του Data Storage παρέχει μια REST-API διεπαφή για την επικοινωνία της βάσης δεδομένων των προϊόντων και την λογική της εφαρμογής. Εκτός από αυτό η διεπαφή προσφέρει και επιπλέον λειτουργίες για την επικοινωνία του orion με την βάση δεδομένων των προϊόντων, την βάση δεδομένων του orion και την λογική της εφαρμογής.

Μερικές από αυτές τις έξτρα λειτουργίες είναι:

- CRUD operations για τα subscriptions και notifications για την βάση του orion και την βάση του data storage που γίνονται μέσα από requests της εφαρμογής,
- και την διαχείριση των updates που στέλνει ο orion για τα αποθηκευμένα entities

Να σημειωθεί ότι όταν στέλνουμε request για την προσθήκη νέου subscription στον orion μέσω του proxy δεν επιστρέφεται στο header του response το Location που περιέχει το new subscription ID που χρησιμοποιείται για την ανάκτηση δεδομένων για το subscription. Για αυτόν τον λόγο, το request για την προσθήκη νέου συνδρομητή γίνεται απευθείας στον orion για να επιστρέφει το ID του νέου συνδρομητή. Όλα τα υπόλοιπα αιτήματα γίνονται κανονικά μέσω του proxy.

Τέλος, όλα τα request μεταξύ των υπηρεσιών γίνονται με Curl Requests.

## Αναπαράσταση Δεδομένων του Orion

Αρχικά, να σημειωθεί ότι επιλέχθηκε η β) επιλογή όπου η χρήστες μπορούν να κάνουν subscribe σε ένα προϊόν και να λαμβάνουν ειδοποιήσεις στις περιπτώσεις που το προϊόν είναι διαθέσιμο ή εξαντλημένο. Αυτό γίνεται με ένα flag(soldout) το οποίο παίρνει τις τιμές 0 ή 1.

Στον orion το κάθε προϊόν που υπάρχει στο Collection Products στην βάση του Data Storage θεωρείται σαν ένα entity. Κάθε entity που υπάρχει αποθηκεύει τα εξής δεδομένα: ένα ID το οποίο είναι το Product ID(orion\_id attribute) του κάθε προϊόντος και την κατάσταση διαθεσιμότητας του προϊόντος(τιμές 0 ή 1).

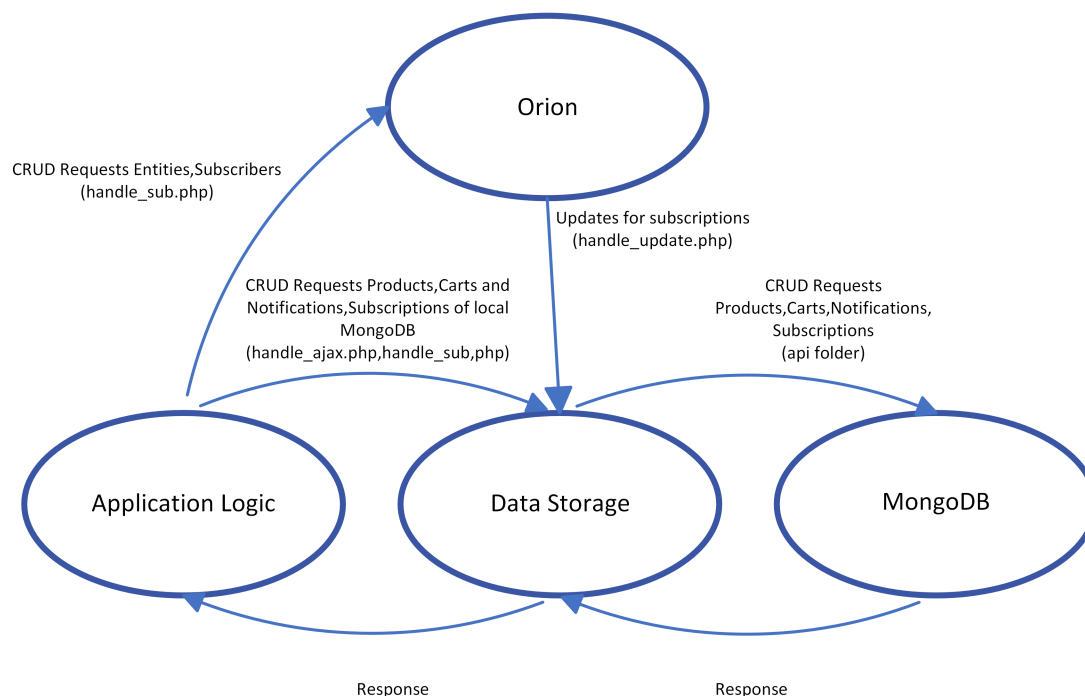
Η κάθε οντότητα κρατάει μόνο την κατάσταση διαθεσιμότητας του προϊόντος διότι είναι η μόνη κατάσταση που μας ενδιαφέρει η μεταβολή της. Οποιαδήποτε άλλη πληροφορία χρειαζόμαστε για την οντότητα χρησιμοποιούμε το αναγνωριστικό της και αναζητούμε τα ζητούμενα στοιχεία στην βάση δεδομένων των προϊόντων.

Όσον αφορά τους συνδρομητές ενημερώνονται με ειδοποίηση για την κατάσταση διαθεσιμότητας των προϊόντων που είναι εγγεγραμμένοι, όπως αναφέραμε και παραπάνω. Οπότε, μοναδικό attribute για παρακολούθηση από τον orion είναι το soldout state.

## Rest API

Η Rest API διεπαφή που σχεδιάστηκε παρέχει λειτουργίες για την μεταβολή στοιχείων της βάσης δεδομένων του Data Storage. Η επικοινωνία γίνεται μέσω curl requests από την κύρια εφαρμογή στον σέρβερ που φιλοξενεί την διεπαφή μέσω ενός pep proxy για την προστασία του. Πιο συγκεκριμένα, η διεπαφή παρέχει λειτουργίες για CRUD operations για τα Collections Products, Carts, Subscriptions και Notifications που είναι αποθηκευμένα στην βάση. Επίσης, η διεπαφή διαχειρίζεται τα updates που στέλνει ο orion για τις οντότητες και δημιουργεί notifications με το κατάλληλο μήνυμα για τον χρήστη που αφορούν.

Παρακάτω, παρατίθεται μια γραφική απεικόνιση για την επικοινωνία του orion, του application logic, του API και της βάσης δεδομένων:



*Επικοινωνία API, Application Logic, Orion και MongoDB*

## Migration της εφαρμογής στο περιβάλλον GCP

Το migration της εφαρμογής στην πλατφόρμα GCP ήταν πολύ εύκολη. Αρχικά, δημιουργήθηκε ένα VM με το λειτουργικό Ubuntu και ρυθμίστηκε ακριβώς όπως έγινε στο φροντιστήριο του μαθήματος. Στην συνέχεια, για την μεταφορά της εφαρμογής, απλά δημιουργήθηκε ένα συμπιεσμένο αρχείο zip από το directory της εφαρμογής και μέσω ssh σύνδεσης που παρέχει η GCP πλατφόρμα και με την λειτουργία upload file ανέβηκε το αρχείο στο μηχάνημα. Μετά, αποσυμπιέστηκε το αρχείο, εγκαταστάθηκαν τα docker-ce και docker-compose πακέτα και με την εντολή docker-compose up σηκώθηκε η εφαρμογή και έτρεχε στην εξωτερική διεύθυνση IP του μηχανήματος.

Επιπλέον, για να εμφανίζεται η εφαρμογή στην διεύθυνση που θέλουμε προστέθηκε ένας κανόνας στο firewall του μηχανήματος για την κίνηση στην θύρα 80 που φιλοξενεί τον σέρβερ της εφαρμογής. Έτσι, μόνο η θύρα 80 είναι προσβάσιμη στον "έξω κόσμο".

## Επιπλέον πληροφορίες για την εργασία

Όπως διατυπώθηκε και σε προηγούμενη ενότητα οι ενημερώσεις που θα λαμβάνει ο χρήστης θα γίνονται με βάση την επιλογή β) με ένα soldout flag. Οπότε, δεν θα υπάρχουν ενημερώσεις με βάση την διαθεσιμότητα σε βάθος ημερών (επιλογή α) ή με βάση την ημερονία απόσυρσης του προϊόντος (πρώτο σκέλος της επιλογής β) και μπορεί να θεωρηθεί ότι αυτή η λειτουργία υλοποιήθηκε μερικώς.

Επίσης, υπάρχουν ελλείψεις στο validation των πεδίων των προϊόντων και των χρηστών στις σελίδες **seller.php** και **administartion.php**, όπου υποστηρίζεται η ζωντανή επεξεργασία των στοιχείων. Παράδειγμα προβλήματος στην επεξεργασία των στοιχείων είναι το username του χρήστη να αλλάζει και να είναι ίδιο με ένα υπάρχον username.

Τέλος, το ID που υπήρχε για τα προϊόντα στην πρώτη φάση της εργασίας μεταφέρθηκε και στην βάση MongoDB αλλά δεν χρησιμοποιείται πουθενά και θέτεται NULL στα καινούργια προϊόντα. Πλέον το νέο αναγνωριστικό των προϊόντων είναι το \_id που δημιουργεί η MongoDB αυτόματα.

**ΣΗΜΕΙΩΣΗ:** Υπάρχουν ήδη εγγεγραμμένοι confirmed λογαριασμοί στην εφαρμογή για κάθε ρόλο, που μπορείται να δοκιμάσετε:

*Email: g@i.com(ADMIN), e@i.com(USER), k@i.com(PRODUCTSELLER)*  
*Password: 123*