

ΠΛΗ-518 ΥΠΗΡΕΣΙΕΣ ΣΤΟ ΥΠΟΛΟΓΙΣΤΙΚΟ ΝΕΦΟΣ ΚΑΙ ΤΗΝ ΟΜΙΧΛΗ
ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2020-2021

2η ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ:

**ΕΠΕΚΤΑΣΗ WEB ΕΦΑΡΜΟΓΗΣ ΜΕ ΧΡΗΣΗ
MICROSERVICES ΣΕ ΠΕΡΙΒΑΛΛΟΝ DOCKER ΚΑΙ
ΕΞΟΙΚΕΙΩΣΗ ΜΕ ΤΟ ΠΕΡΙΒΑΛΛΟΝ GCP**

Επιμέλεια Εργασίας: Τσακός Κωνσταντίνος (email: ktsakos@isc.tuc.gr), Τζαβάρας Αιμίλιος
(email: atzavaras@isc.tuc.gr)

Καθηγητής: Πετράκης Ευριπίδης

ΠΕΡΙΓΡΑΦΗ ΑΣΚΗΣΗΣ

- Σκοπός της άσκησης είναι η εξοικείωση με ένα πραγματικό περιβάλλον υπολογιστικού νέφους (Google Cloud Platform) και η ανάπτυξη εφαρμογής με χρήση κιβωτίων (containers).
- Επιπλέον σκοπός της άσκησης είναι η εξοικείωση με ασύγχρονο τρόπο ανταλλαγής δεδομένων και η δυναμική ανανέωση του περιεχομένου των σελίδων με χρήση της τεχνολογίας AJAX.
- Παράλληλα, θα αναπτυχθεί μία υπηρεσία αποθήκευσης των δεδομένων της εφαρμογής η οποία θα δέχεται REST αιτήματα και θα τα μετατρέπει στα αντίστοιχα operations σε μία βάση MongoDB.
- Επίσης, ζητούμενο της άσκησης είναι η εξοικείωση με την επικοινωνία υπηρεσιών μέσω REST, επεκτείνοντας παράλληλα την λειτουργικότητα της εφαρμογής με έτοιμες υπηρεσίες που σχετίζονται με θέματα ασφάλειας της εφαρμογής.
- Τέλος, θα χρειαστεί να εξοικειωθείτε με τη χρήση ενός publish-subscribe μηχανισμού με σκοπό τη δημιουργία συνδρομών και ειδοποιήσεων για τους χρήστες της πλατφόρμας.

Αναλυτικά, τα ζητούμενα της άσκησης είναι τα εξής:

1) ΑΝΑΠΤΥΞΗ (DEPLOYMENT) ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΣΕ ΠΕΡΙΒΑΛΛΟΝ DOCKER

Εγκαταστήστε τις απαραίτητες υπηρεσίες σε μορφή containers. Ενδεικτικά, θα χρειαστείτε μία MySQL (για τα δεδομένα της εφαρμογής και του Keyrock), την υπηρεσία Keyrock IDM, την υπηρεσία Orion Context Broker, την υπηρεσία Pep Proxy Wilma, μια MongoDB (απαραίτητη για τη λειτουργία του Orion CB) και έναν Apache Server για την λογική της εφαρμογής (βλ. παρακάτω εικόνα).

Φροντίστε οι υπηρεσίες να συνδέονται κατάλληλα μέσω του δικτύου, να εκθέτουν εξωτερικά μόνο τις πόρτες που χρειάζεται και να διατηρούν τα δεδομένα τους ακόμα και αν πέσει κάποιο container.

Όλα τα containers της εφαρμογής θα πρέπει να περιγράφονται μέσα σε ένα αρχείο *docker-compose.yml*.

2) ΔΥΝΑΜΙΚΗ ΑΝΑΝΕΩΣΗ ΤΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ ΤΩΝ ΣΕΛΙΔΩΝ ΜΕΣΩ ΑΣΥΓΧΡΟΝΗΣ ΕΠΙΚΟΙΝΩΝΙΑΣ

Για κάθε δημιουργία, επεξεργασία, διαγραφή στη βάση δεδομένων, το ανανεωμένο περιεχόμενο θα πρέπει να εμφανίζεται *δυναμικά* στην αντίστοιχη σελίδα μετά από κάθε ενέργεια. Δηλαδή να μη χρειάζεται να ανανεωθεί η εκάστοτε σελίδα για να δει κάποιος χρήστης τις αλλαγές του.

Για παράδειγμα, ο admin διαγράφει το χρήστη πατώντας ένα κουμπί και ο πίνακας με τους χρήστες που βλέπει ανανεώνεται δυναμικά χωρίς την ανανέωση της σελίδας αυτής.

Χρησιμοποιήστε AJAX για ασύγχρονη επικοινωνία με τον server της εφαρμογής για να επιτύχετε όλα τα παραπάνω.

3) ΑΠΟΘΗΚΕΥΣΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΜΕΣΩ ΤΗΣ ΥΠΗΡΕΣΙΑΣ DATA STORAGE

Τα δεδομένα της εφαρμογής πρέπει πλέον - αντί να αποθηκεύονται σε μία MySQL βάση δεδομένων - να αποθηκεύονται σε μία MongoDB με χρήση της υπηρεσίας Data Storage Service.

Σχεδιάστε κατάλληλα το REST API της υπηρεσίας η οποία θα πρέπει να μετατρέπει τα REST αιτήματα σε κατάλληλα queries προς μία NoSQL MongoDB βάση δεδομένων.

Προστατέψτε και αυτήν την εφαρμογή με χρήση του PEP Proxy Wilma.

4) ΕΠΕΚΤΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΜΕ ΧΡΗΣΗ PUB/SUB ΥΠΗΡΕΣΙΑΣ (ORION CB) ΓΙΑ ΤΗΝ ΕΝΗΜΕΡΩΣΗ ΤΩΝ ΧΡΗΣΤΩΝ ΣΧΕΤΙΚΑ ΜΕ ΝΕΕΣ ΤΑΙΝΙΕΣ ΠΟΥ ΤΟΥΣ ΕΝΔΙΑΦΕΡΟΥΝ Ή ΠΟΥ ΠΑΥΟΥΝ ΝΑ ΠΑΙΖΟΝΤΑΙ

Φροντίστε κατά τη δημιουργία ενός *Σινεμά* να καταχωρείται η αντίστοιχη οντότητα στον Orion Context Broker (FIWARE). Κάθε *οντότητα τύπου "Cinema"* στον Orion θα αντιπροσωπεύει έναν κινηματογράφο και θα κρατάει πληροφορία για τις διαφορετικές ταινίες που παίζονται σε αυτό το σινεμά (και οι ταινίες με τη σειρά τους μπορούν να έχουν τις δικές τους πληροφορίες). Εναλλακτικά, μπορούν να δημιουργούνται οντότητες στον Orion για κάθε ταινία (τύπου *Movie*).

Οι απλοί χρήστες θα μπορούν να κάνουν *subscribe* σε ταινίες και να λαμβάνουν ειδοποίηση στις εξής περιπτώσεις: α) όταν μια ταινία βγαίνει σε κυκλοφορία σε διάστημα ημερών που θα θέσετε (π.χ. η ταινία “Ο Αρχοντας των Δαχτυλιδιών” είναι τώρα διαθέσιμη στο Σινεμά Αττικών) και β) όταν μια ταινία δεν παίζεται πια. Οι πληροφορίες αυτές θα πρέπει να στέλνονται αυτόματα τόσο στη βάση δεδομένων για να προβάλλονται στο χρήστη όταν εισέλθει ξανά στην εφαρμογή όσο και στο γραφικό περιβάλλον για ενημέρωση του χρήστη σε *πραγματικό χρόνο* όταν συμβαίνει κάποια ενημέρωση των στοιχείων της ταινίας από τους ιδιοκτήτες και πρέπει να ειδοποιηθούν για αυτές στο feed τους. Το feed του χρήστη θα πρέπει δηλαδή να προβάλλει πληροφορίες για αλλαγή της κατάστασης μιας ταινίας και να τις διατηρεί για συγκεκριμένο χρονικό διάστημα. Χρησιμοποιήστε για όλα τα παραπάνω τον μηχανισμό pub/sub που παρέχει ο Orion CB μέσω της REST διεπαφής. Ο Orion για να δουλέψει απαιτεί μία MongoDB βάση δεδομένων στην οποία αποθηκεύει όλα του τα δεδομένα. Για τις λειτουργίες όμως του Orion αρκεί να δουλέψετε με το REST API του.

5) ΧΡΗΣΗ ΤΗΣ ΥΠΗΡΕΣΙΑΣ KEYROCK IDM (FIWARE) ΓΙΑ ΑΥΘΕΝΤΙΚΟΠΟΙΗΣΗ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΤΩΝ ΧΡΗΣΤΩΝ

Αλλάξτε την έως τώρα λογική της εφαρμογής για ταυτοποίηση χρηστών και χρησιμοποιήστε την REST διεπαφή του Keyrock για την αυθεντικοποίησή τους (user authentication) μέσω του OAuth2 πρωτοκόλλου. Θα χρειαστεί επιπλέον να κάνετε register την εφαρμογή σας μέσα από το γραφικό περιβάλλον του Keyrock και να κωδικοποιήσετε το `client_id` και το `client_secret` που θα σας δοθούν από τον Keyrock. Η κωδικοποίηση αυτή θα πρέπει να γίνει σε *base64* (`client_id:client_secret`). Το αποτέλεσμα της κωδικοποίησης θα αποτελέσει το Authorization Basic header που απαιτείται στο REST call του Keyrock για το authentication και την παροχή ενός προσωρινού access token στην εφαρμογή.

Ουσιαστικά καταργούμε την admin page της εφαρμογής και όλη η διαχείριση γίνεται από το dashboard του Keyrock.

Για να έχουν πρόσβαση στην εφαρμογή, οι χρήστες και οι ιδιοκτήτες σινεμά θα πρέπει να έχουν κάνει sign up στον Keyrock και να έχουν *επιβεβαιωθεί* από κάποιον διαχειριστή: α) τόσο για πρόσβαση στον Keyrock, όσο και β) για δικαίωμα πρόσβασης στην εφαρμογή.

6) ΧΡΗΣΗ ΤΗΣ ΥΠΗΡΕΣΙΑΣ PER PROXY - WILMA (FIWARE) ΓΙΑ ΠΡΟΣΤΑΣΙΑ ΤΩΝ BACKEND ΕΦΑΡΜΟΓΩΝ ΑΠΟ ΜΗ ΕΞΟΥΣΙΟΔΟΤΗΜΕΝΟΥΣ ΧΡΗΣΤΕΣ

Με την υπηρεσία Per Proxy, κάθε αίτημα προς την υπηρεσία Orion Context Broker ελέγχεται για τον αν θα προωθηθεί ή όχι σε αυτή. Έτσι, τα αιτήματα προς τον Orion CB γίνονται μέσω του Per Proxy, ο οποίος επιβεβαιώνει την ορθότητα του *masterkey* που δίνεται στο header των αιτημάτων και, μόνο αν αυτό είναι σωστό, προωθεί το αίτημα στον Orion. Με αυτόν τον τρόπο, η υπηρεσία προστατεύεται από κακόβουλους χρήστες και εφαρμογές.

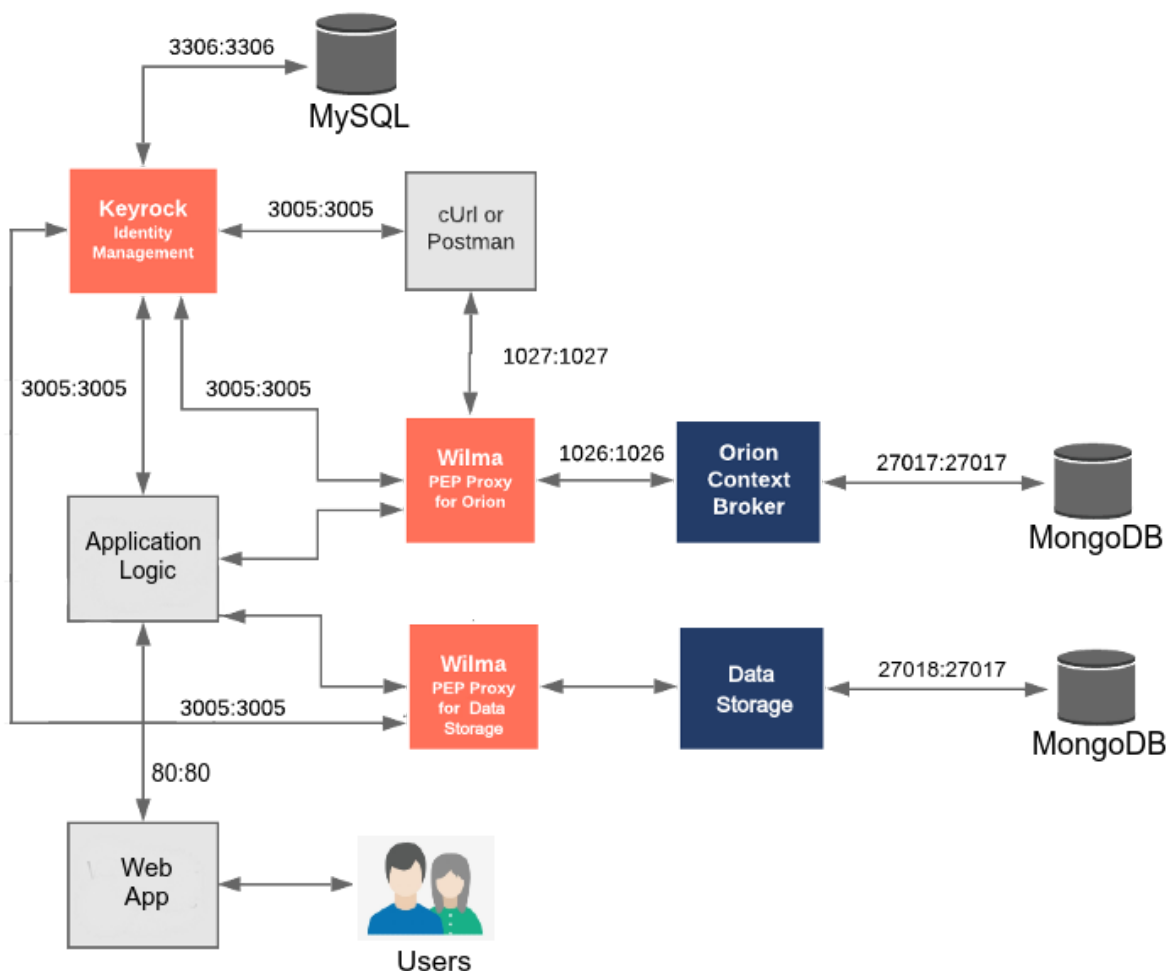
7) MIGRATION ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΣΕ INSTANCE ΤΟΥ GOOGLE CLOUD PLATFORM

Δημιουργήστε ένα instance με λειτουργικό Ubuntu, εγκαταστήστε το Docker σε αυτό και σηκώστε εκεί την εφαρμογή.

Θα χρειαστεί να μεταφέρετε στην εικονική μηχανή τα αρχεία του κώδικά σας και τα αντίστοιχα Dockerfiles, docker-compose, κλπ.

Θα χρειαστεί επίσης να “ανοίξετε” έξω από το VM (expose) τις πόρτες που αφορούν τον server της εφαρμογής και τον Keyrock για πρόσβαση στα Web UIs τους.

Ακολουθεί ένα βοηθητικό/ενδεικτικό διάγραμμα της αρχιτεκτονικής που θα συνθέτουν οι παραπάνω υπηρεσίες.



Ενδεικτικό διάγραμμα αρχιτεκτονικής υπηρεσιών

ΧΡΗΣΙΜΑ LINKS:

<https://fiware-orion.readthedocs.io/en/master/>

<https://fiware-pep-proxy.readthedocs.io/en/latest/>

<https://fiware-idm.readthedocs.io/en/latest/index.html>

<https://documenter.getpostman.com/view/513743/RWMLLRui>

<https://documenter.getpostman.com/view/513743/RWaHxUgP>

<https://documenter.getpostman.com/view/513743/fiware-getting-started/RVu5kp1c>

https://www.tutorialspoint.com/mongodb/mongodb_php.htm

<https://www.php.net/manual/en/mongodb.tutorial.library.php>

ΣΗΜΕΙΩΣΗ:

- Φροντίστε η εφαρμογή να είναι ευχάριστη και εύκολη στη χρήση (εικόνες, εύχρηστο μενού, γραμματοσειρές, χρώματα, έλεγχος εγκυρότητας δεδομένων εισόδου στις φόρμες, κτλ.).
- Είστε ελεύθεροι να χρησιμοποιήσετε οποιοδήποτε Web Framework σας εξυπηρετεί τόσο στη δημιουργία του γραφικού περιβάλλοντος (frontend) όσο και στην υλοποίηση της λογικής της εφαρμογής (backend).
- Φροντίστε κατά το migration της εφαρμογής σε άλλο host μηχανήμα να *διατηρούνται* τα δεδομένα των βάσεων δεδομένων.

ΠΑΡΑΔΟΤΕΑ:

- Scripts κώδικα με επεξηγηματικά σχόλια.
- Αναφορά μισής σελίδας με παρατηρήσεις για τη δουλειά σας: πώς κάνατε το migration στο GCP, τι δικτύωση υπάρχει μεταξύ των υπηρεσιών και πώς αυτές συνεργάζονται, τεχνολογίες και frameworks που χρησιμοποιήθηκαν και πιθανές ελλείψεις της εφαρμογής που δεν υλοποιήθηκαν ή υλοποιήθηκαν μερικώς, τον τρόπο αναπαράστασης των δεδομένων στον Orion, τι θεωρείτε ως entity και τι attributes έχει, το rest API reference του Data Storage το οποίο σχεδιάσατε και υλοποιήσατε
- Αρχείο .sql (backup αρχείο της mysql βάσης).

