

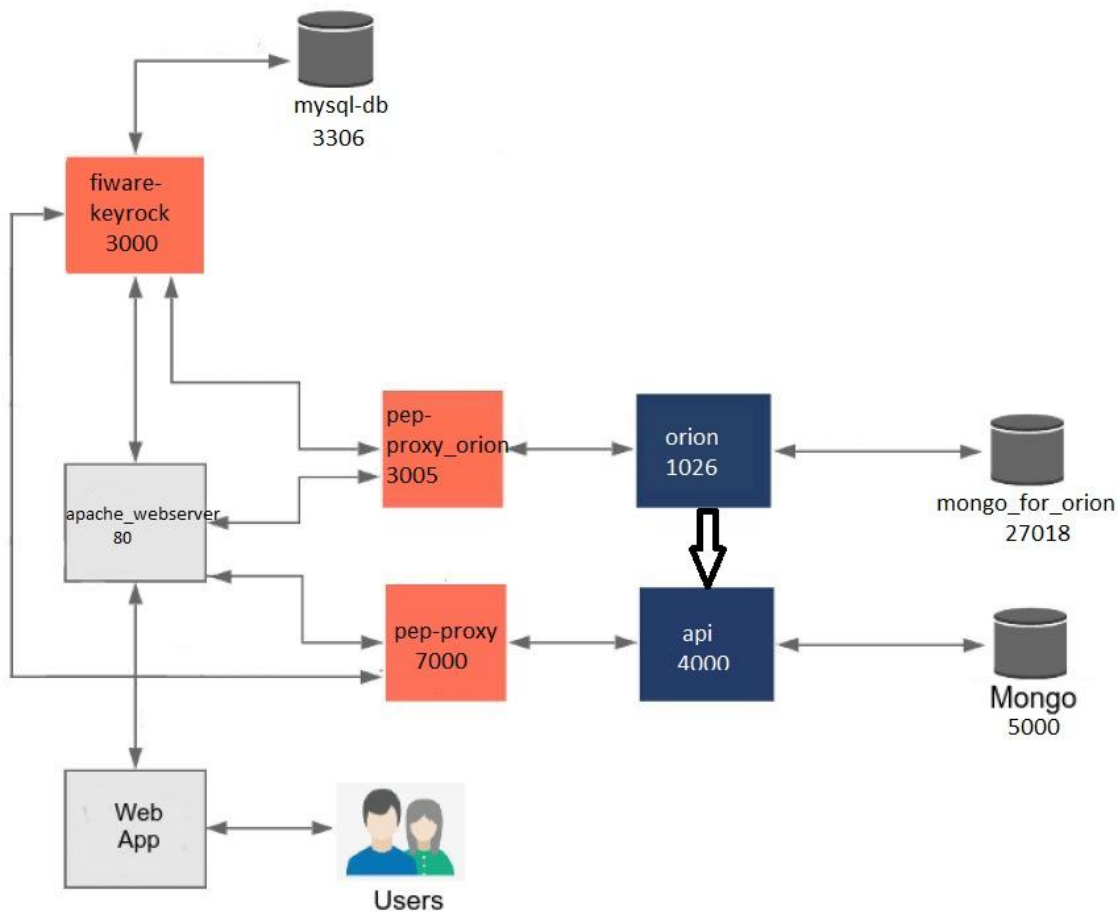


Υπηρεσίες στο Υπολογιστικό Νέφος και την Ομίχλη

Αναφορά άσκησης

Χαμαρούσιος Δημήτριος 2015030090

Οι υπηρεσίες-containers της άσκησης είναι υλοποιημένα σύμφωνα με το εξής διάγραμμα:



Κάθε “τετράγωνο” αποτελεί μια υπηρεσία και ο αριθμός αποτελεί την πόρτα στην οποία τρέχει η εκάστοτε υπηρεσία. Η αντιστοίχιση των ports με το πραγματικό μηχανήμα δεν αναφέρεται στο διάγραμμα, καθώς για την δικτύωση έχει χρησιμοποιηθεί bridge, δηλαδή παίρνει μια ip και το φυσικό μηχανήμα, οπότε έχει άμεση πρόσβαση και στις υπηρεσιών (localhost:port). Παρόλο αυτά μπορεί κανείς να δει την αντιστοίχιση στο αρχείο yml.

Frameworks που χρησιμοποιήθηκαν:

- Django (back-end)
- Vue js (μεγαλύτερο μέρος front-end)

Άλλες τεχνολογίες:

- bootstrap (front, εμφανισιακό κομμάτι)

Τα δεδομένα στον orion αποτελούνται από δύο συλλογές (csubs, entities). Ως entity θεωρείται κάθε ταινία και έχει τα εξής attributes:

- start_date
- end_date

- movie_name
- cinema

Api

Δημιουργήθηκε με χρήση django και της βιβλιοθήκης djongo

- movieAPI

GET movie/ - επιστρέφει json με όλες τις ταινίες και πληροφορίες της
 POST movie/ δημιουργία νέας ταινίας και επιστροφή σε json την ταινία που δημιουργήθηκε ώστε να μάθει κανείς το id της.
 PUT movie/{{movie_id}} -επεξεργασία την ταινίας με id=movie_id
 DELETE movie/{{movie_id}} -διαγραφή της ταινίας με id=movie_id

GET movieexists/{{movie_id}} - ελέγχει αν η ταινία με id=movie_id υπάρχει.Αν υπάρχει επιστρέφει True,αλλιώς False.

- favoriteAPI

GET favorite/ - επιστρέφει json με όλα τα αγαπημένα και πληροφορίες τους
 POST favorite/ δημιουργία νέου αγαπημένου και επιστροφή σε json του αγαπημένου που δημιουργήθηκε ώστε να μάθει κανείς το id του.
 PUT favorite/{{favorite_id}} -επεξεργασία του αγαπημένου με id=favorite_id
 DELETE favorite/{{favorite_id}} -διαγραφή του αγαπημένου με id=favorite_id

- cinemasAPI

GET cinema/{{owner_id}} - επιστρέφει json με όλα τα cinema και πληροφορίες τους,που ανήκουν στον owner με id=owner_id
 POST cinema/ δημιουργία νέου σινεμά και επιστροφή σε json του σινεμά που δημιουργήθηκε ώστε να μάθει κανείς το id του.
 PUT cinema/{{cinema_id}} -επεξεργασία του σινεμα με id=cinema_id
 DELETE favorite/{{cinema_id}} -διαγραφή του αγαπημένου με id=cinema_id

- checksubAPI

Χρησιμοποιείται,ώστε να γνωρίζουμε ποιος χρήστης είναι γραμμένος σε ποια ταινία. Είναι ανεξάρτητο από το csubs του orion,καθώς βρίσκεται στην άλλη βάση δεδομένων.

GET checksub/{{user_id}} - επιστρέφει json με όλα τα checksubs και πληροφορίες τους,που ανήκουν στον user με id=user_id
 POST checksub/ δημιουργία νέου subscription και επιστροφή σε json του subscription που δημιουργήθηκε ώστε να μάθει κανείς το id του.

DELETE checksub/{{sub_id}} -διαγραφή του subscription με id=sub_id

- subAPI

Αφορά το end point που χρησιμοποιεί ο orion,προκειμένου να κάνει post κάποια αλλαγή που έγινε σε κάποια ταινία.Χρησιμοποιείται στο feed του χρηστη, ώστε να ενημερώνεται για τυχόν αλλαγές σε ταινίες.

GET sub/{{user_id}} - επιστρέφει json με όλα τα subs και πληροφορίες τους,που ανήκουν στον owner με id=user_id

POST sub/ δημιουργία νέου sub και επιστροφή σε json του sub που δημιουργήθηκε ώστε να μάθει κανείς το id του.

DELETE sub/{{sub_id}} -διαγραφή του sub με id=sub_id

Εγκατάσταση(GCP)

Η εφαρμογή βρίσκεται ήδη εγκατεστημένη στο google cloud και μπορεί κανείς να την βρει εδώ: <http://34.107.87.234/>

Το gui του keyrock εδώ: <http://34.107.87.234:3000> .Για τα στοιχεία εισόδου βλ. στο *Credentials*, πιο κάτω.

Προκειμένου να εγκατασταθεί στο GCP,έπρεπε να φτιαχτεί ένα vm instance με ubuntu,να εγκατασταθεί το docker εκεί,να φτιαχτούν οι κατάλληλοι rules στο firewall,ώστε να επιτρέπεται η κίνηση από http κι τέλος στα αρχεία του project,έπρεπε να αλλαχτούν αρκετές ips στο ip του vm instance που φτιαχτηκε.

Εγκατάσταση(τοπική σε Ubuntu)

Για να τρέξει κάποιος το όλο project,αρκεί να τρέξει μέσα στον root folder της εργασίας την εντολή:

sudo docker-compose up

Αυτόματα όταν ξεκινούν τα services, γίνεται η εισαγωγή κάποιων δεδομένων,όπως λογαριασμοί χρηστών,ταινίες,αγαπημένα κτλ.

Χρήσιμες πληροφορίες:

Keyrock: <http://172.18.1.5:3000/>

Web App: <http://172.18.1.7/>

Credentials:

ιδιοκτήτης σινεμά 1: simple_owner@fake.com 1234

ιδιοκτήτης σινεμά 2: simple_owner2@fake.com 1234

θεατής 1: simple_viewer@fake.com 1234

θεατής 2: test9@gmail.com 1234

admin account: admin@test.com 1234

Παρακαλώ εισέλθετε σαν admin μόνο στον Keyrock και όχι στο webApp.

Κάθε λειτουργία που ζητείται στην εκφώνηση, έχει υλοποιηθεί και λειτουργεί πλήρως.

