Οικονομικό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής
Τεχνολογίες Ψηφιακών Υποδομών
ΠΜΣ ΣΤΗΝ ΑΝΑΠΤΥΞΗ & ΑΣΦΑΛΕΙΑ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ (Full Time)
Ακαδημαϊκό Έτος 2022-2023
Διδάσκων: Β. Καλογεράκη
Περιγραφή Εργασίας 1

Τα τελευταία χρόνια, λόγω της τεράστιας αύξησης της ποσότητας των πληροφορίας που έχουμε διαθέσιμης, μέσω των κινητών τηλεφώνων των χρηστών, υπάρχει δυνατότητα να γίνεται πολύ πιο εύκολα η ανάπτυξη εφαρμογών, κάτι το οποίο λίγο καιρό πριν θα ήταν αδύνατο χωρίς την αγορά εξειδικευμένου εξοπλισμού. Πλέον με το πλήθος των αισθητήρων που διαθέτει η κάθε συσκευή, υπάρχει ένα τεράστιο πλήθος πληροφορίας διαθέσιμο, και το πρόβλημα έχει μετατοπιστεί στο πως θα επεξεργαστεί και θα χρησιμοποιηθεί κατάλληλα αυτή η πληροφορία.

Για την αποθήκευση και επεξεργασία αυτού του μεγάλου όγκου πληροφορίας, τα τελευταία χρόνια, έχουν αναπτυχθεί πλήθος κατανεμημένων frameworks, τα οποία χρησιμοποιούν πολλούς “απλούς” υπολογιστές προκειμένου να αποθηκεύσουν και να επεξεργαστούν την πληροφορία, αντί να χρησιμοποιούν ένα “μεγάλο” και υψηλού κόστους server.

Παράλληλα τα τελευταία χρόνια έχει εμφανιστεί μια νέα έννοια στο cloud computing που ονομάζεται Serverless Computing. Με το Serverless Computing, μπορούμε να υλοποιούμε μικροπρογράμματα (functions) σε διάφορες γλώσσες προγραμματισμού και να τα χρησιμοποιούμε στο cloud χωρίς όμως να υπάρχει η ανάγκη μόνιμης υποδομής και δέσμευσης πόρων όπως π.χ. στο Spark. Αυτή η δυνατότητα μας δίνεται μέσω της χρήσης containers, τα οποία περιέχουν ένα image του προγράμματος που θέλουμε να εκτελέσουμε. Ένας προγραμματιστής γράφει το πρόγραμμά του σε μια γλώσσα της αρεσκείας του, φτιάχνει ένα image και το κάνει deploy σε ένα περιβάλλον serverless, όπως το AWS Lambda, Google Cloud Functions, OpenFaas, OpenWhisk και αρκετά άλλα. Το πλεονέκτημα αυτού του μοντέλου είναι ότι παρέχει άμεσο scaling και χαμηλό κόστος, αφού δεν ενοικιάζεται μόνιμη υποδομή. Για παράδειγμα μπορούμε να φτιάξουμε ένα function σε Python, να δημιουργήσουμε 1000 instances αυτού του function, να τα χρησιμοποιήσουμε για 10s και να τα καταστρέψουμε πληρώνοντας ελάχιστα cents.

Στην παρούσα εργασία καλείστε να χρησιμοποιήσετε ένα τέτοιο περιβάλλον της αρεσκείας σας και να υλοποιήσετε ένα σύστημα επεξεργασίας δεδομένων ενοικιαζόμενων ποδηλάτων από την πόλη την Νέας Υόρκης. Κάθε αρχείο csv αντιστοιχεί σε δεδομένα ενός μήνα. Στο τέλος του μήνα ο διαχειριστής του συστήματος

καλείται να υλοποιήσει κάποια ερωτήματα πάνω σε αυτά τα δεδομένα με επεξεργασία Map-Reduce πάνω από Serverless περιβάλλον.
Καλείστε να φτιάξετε τις υλοποιήσεις των functions του mapper και του reducer ώστε να απαντούν τα παρακάτω ερωτήματα.

1) Έστω ότι χωρίζουμε την πόλη σε τεταρτημόρια γύρω από το σημείο (40.735923, -73.990294)
Να υπολογιστεί ο αριθμός των διαδρομών που ξεκίνησαν σε κάθε τεταρτημόριο.
Παράδειγμα output: {Q1 }->300 ... {Q2 }->500

2) Να βρεθούν οι 10 πιο δημοφιλείς σταθμοί ποδηλάτων από τους οποίους ξεκίνησαν οι περισσότερες διαδρομές ανά 6ωρο. Παράδειγμα output:
{12:00-18:00, station 1} → 500, {18:00-24:00, station 1} →750, {{00:00-6:00, station 1}} →50, {06:00-12:00, station 1} →850.
Τα 6ωρα είναι για όλο το μήνα συνολικά.
Τέλος αυτοί οι 10 σταθμοί θα πρέπει να εκτυπώνονται πάνω σε ένα χάρτη της Νέας Υόρκης (σαν εικόνα όχι interactive).

3) Bonus. Γράψτε ένα Map Reduce που κατά τη γνώμη σας παρέχει επιπλέον χρήσιμες πληροφορίες.

Τεχνική υλοποίηση:

- Πρέπει να υλοποιήσετε τα functions που απαντούν στα παραπάνω ερωτήματα σαν Map Reduce. Μπορείτε να υλοποιήσετε διαφορετικά function images για τα ερωτήματα.
- **Η επικοινωνία μεταξύ των functions θα γίνεται μέσω του Minio με τη χρήση events όπως περιγράφεται στο pdf που υπάρχει στο eclass**.
- Μπορείτε να χρησιμοποιήσετε το VM που σας έχει σταλεί το οποίο περιέχει προεγκατεστημένο το OpenFaas.
- Πρέπει να κάνετε εγκατάσταση το Minio και να φτιάξετε τα κατάλληλα events.

Λεπτομέρειες σχετικά με το deployment των functions πάνω στο OpenFaas.

Θα πρέπει να φτιάξετε 2 instances από το Mapper function και 1 instance από το Reducer function. (δείτε docker service scale https://docs.docker.com/engine/reference/commandline/service/)
Τα requests εκτελούνται αυτόματα και παράλληλα (round robin) στα deployed instances. **Συνεπώς θα πρέπει να σπάσετε με εξωτερικό script το csv αρχείο τουλάχιστον 2 parts** και να τα ανεβάσετε σε ένα αρχικό bucket στο minio. Κάθε φορά που ολοκληρώνεται το upload στο bucket θα πρέπει να στέλνει event στον Mapper Function και να ξεκινάει αυτόματα η επεξεργασία. Ο κάθε mapper πρέπει να ανεβάζει τα αποτελέσματά του σε ένα bucket με όνομα Mapper.

Η φάση του reduce θα πρέπει να ξεκινάει όταν έχει τελειώσει η φάση του mapping.

Hint! Κάθε φορά που ανεβαίνει ένα intermediate result στο Mapper Bucket, ένα event στέλνεται στον Reducer. Ο Reducer όταν λάβει το event πρέπει να **δει αν υπάρχουν όλα τα intermediate results στο Mapper bucket** (ξέρετε από πριν πόσα θα είναι) και μόνο τότε να κάνει το τελικό reduce. Το αποτέλεσμα του Reducer πρέπει να ανέβει και αυτό σε ένα bucket με όνομα Reducer.

Τέλος μετά την ολοκλήρωση του reduce θα πρέπει να καλείτε το τελικό function που θα φτιάξει την εικόνα με τα points στο χάρτη και θα την αποθηκεύει στο Minio.