

The Computational Cloud Conundrum

➤ ΓΕΡΟΝΤΑΣ ΑΛΕΞΑΝΔΡΟΣ



Ανάλυση προβλήματος

- Οι πληροφορίες για το περιβάλλον και τα stacks θα πρέπει να αποθηκεύονται σε ένα αρχείο με συγκεκριμένη δομή.
 - Γίνεται να αποθηκευτούν σε αρχεία json ή yaml.
 - Μπορούν να αποθηκευτούν είτε ξεχωριστά ή στο ίδιο αρχείο.
- Η γλώσσα προγραμματισμού θα πρέπει:
 - Να περιέχει κλάσεις για την κατάλληλη αναπαράσταση των πληροφοριών του περιβάλλοντος και των stack.
 - Μπορούν να χρησιμοποιηθούν οι γλώσσες C ή C++.
 - Να περιέχει δομές για την καταχώρηση των πληροφοριών του περιβάλλοντος και των stack.
 - Να είναι συμβατή με κάποια βιβλιοθήκη που να μπορεί να κάνει parse αρχεία json / yaml.

Επίλυση προβλήματος

- Όλα τα stacks και τα dependencies τους μπορούν να αποθηκευτούν σε ένα αρχείο json.
- Το κάθε περιβάλλον θα αποθηκεύεται σε αρχείο json με τα stack που το αποτελούν σε ένα αντικείμενο json_array.
- Το πρόγραμμα θα υλοποιηθεί σε γλώσσα προγραμματισμού C++
- Μπορούν να χρησιμοποιηθούν οι κλάσεις για την αναπαράσταση του περιβάλλοντος και των stack.
- Οι πληροφορίες για το περιβάλλον και το κάθε stack μπορούν να καταχωρούνται στις δομές List και Map της γλώσσας.
- Η γλώσσα είναι συμβατή με τη cJSON βιβλιοθήκη που θα χρησιμοποιηθεί για την ανάγνωση και εγγραφή json αρχείων.

Το πρόγραμμα αποτελείται από τρεις κλάσεις:

- Κλάση Stack:
 - Στην κλάση Stack καταχωρείται το όνομα του stack και τα dependency του σε μία λίστα.
- Κλάση Environment:
 - Στην κλάση Environment καταχωρείται το όνομα του περιβάλλοντος και μια λίστα με τα Stack που το απαρτίζουν.
- Κλάση Stacks:
 - Στην κλάση stacks φορτώνουμε όλα τα stack που βρίσκονται στο json αρχείο και καταχωρούνται σε ένα map.
 - Η κλάση διαθέτει κατάλληλες μεθόδους για την εξαγωγή των stack με βάση το όνομα ή το index του αντικειμένου το οποίο δίνεται σαν όρισμα.

Μέθοδοι κλάσης Stacks

- ▶ Δημιουργία περιβάλλοντος:
 - ▶ Η μέθοδος δέχεται σαν όρισμα ένα αντικείμενο json σαν string.
 - ▶ Διαβάζει το όνομα και τα Stack από το json, δημιουργεί και επιστρέφει ένα αντικείμενο Environment
- ▶ Έλεγχος της δομής του περιβάλλοντος:
 - ▶ Η μέθοδος δέχεται σαν όρισμα ένα αντικείμενο Environment.
 - ▶ Για κάθε Stack που υπάρχει στο Environment ελέγχεται αν περιέχονται όλα τα dependency στο περιβάλλον.
 - ▶ Αν δεν βρεθούν όλα τα dependency στο περιβάλλον τυπώνεται αντίστοιχο μήνυμα και η συνάρτηση επιστρέφει false.
- ▶ Ταξινόμηση των Stack του περιβάλλοντος:
 - ▶ Η μέθοδος δέχεται σαν όρισμα ένα αντικείμενο Environment.
 - ▶ Δημιουργεί μια νέα λίστα στην οποία προσθέτει μόνο Stack χωρίς dependency ή Stack με dependency που υπάρχουν στην λίστα.
 - ▶ Επιστρέφει ένα αντικείμενο Environment με τη νέα λίστα Stack.

Μέθοδοι κλάσης Environment

- ▶ Προσθήκη Stack:
 - ▶ Η μέθοδος δέχεται σαν όρισμα ένα αντικείμενο Stack και το προσθέτει στη λίστα με τα dependency.
- ▶ Έλεγχος αν το Stack υπάρχει στο περιβάλλον:
 - ▶ Η μέθοδος δέχεται σαν όρισμα ένα αντικείμενο Stack και ελέγχει αν υπάρχει στο περιβάλλον.
- ▶ Αποθήκευση ως json:
 - ▶ Η μέθοδος δέχεται σαν όρισμα το όνομα του αρχείου και αποθηκεύει το περιβάλλον σε μορφή json.

Συνάρτηση Main

- ▶ Το πρόγραμμα δίνει στο χρήστη πέντε επιλογές:
 - ▶ Δημιουργία περιβάλλοντος από αρχείο.
 - ▶ Ο χρήστης εισάγει το όνομα του αρχείου που θέλει να ανοίξει.
 - ▶ Προσθήκη νέου Stack στο περιβάλλον.
 - ▶ Ο χρήστης επιλέγει το Stack που θέλει να προσθέσει από μία λίστα.
 - ▶ Επαλήθευση των Stack σε ένα περιβάλλον.
 - ▶ Τυπώνει στον χρήστη αν λείπουν dependency και από ποιο Stack.
 - ▶ Ταξινόμηση των Stack στο περιβάλλον.
 - ▶ Ταξινομεί τα Stack στο περιβάλλον και τυπώνει τη νέα σειρά.
 - ▶ Αποθήκευση περιβάλλοντος σε αρχείο.
 - ▶ Ο χρήστης εισάγει το όνομα του αρχείου που θέλει να αποθηκεύσει το περιβάλλον.
- ▶ Έξοδος.

Εκτέλεση προγράμματος

- ▶ Αρχείο json:

```
{"environment": [{"name": "environment_1", "stacks": ["IAM", "WAF", "Audit", "Firewall"]}]}
```

- ▶ Φόρτωση ενός περιβάλλοντος από το αρχείο

```
1. load Environment from file
2. add dependency to Environment
3. verify Environment dependencies
4. serialize Environment dependencies
5. save Environment
>> 1
enter input filename
>> env1.txt

environment name: environment_1
dependencies:
0. IAM
1. WAF
2. Audit
3. Firewall
```


Εκτέλεση προγράμματος

- ▶ Έλεγχος της δομής του περιβάλλοντος:

```
1. load Environment from file
2. add dependency to Environment
3. verify Environment dependencies
4. serialize Environment dependencies
5. save Environment
>> 3
WAF dependency Network missing
Firewall dependency Network missing
```

- ▶ Ταξινόμηση των Stack του περιβάλλοντος:

```
1. load Environment from file
2. add dependency to Environment
3. verify Environment dependencies
4. serialize Environment dependencies
5. save Environment
>> 4

dependencies:
0. IAM
1. Audit
2. Network
3. WAF
4. Firewall
```

Μελλοντικά ζητήματα

- ▶ Στο πρόγραμμα θα μπορούσαν να προστεθούν οι εξής δυνατότητες:
 - ▶ Φόρτωση και επεξεργασία περισσότερου από ένα περιβάλλον σε κάθε εκτέλεση του προγράμματος.
 - ▶ Υποστήριξη διαφορετικών τύπων αρχείων εκτός από json όπως yaml.
- ▶ Το πρόγραμμα θα μπορούσε να δίνει τις παρακάτω επιλογές στο χρήστη:
 - ▶ Αυτόματη εισαγωγή Stack που λείπουν από το περιβάλλον.
 - ▶ Προσθήκη / Αφαίρεση πολλαπλών Stack σε ένα περιβάλλον.
 - ▶ Επιλογή και αφαίρεση Stack από το περιβάλλον καθώς και των dependency τους αν δεν απαιτούνται από άλλο Stack.