



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Τμήμα Πληροφορικής και Τηλεματικής
Μεταπτυχιακό πρόγραμμα
Πληροφορική & Τηλεματική

Εργασία μαθήματος «Πλατφόρμες Cloud»

Επί των φοιτητών:

Πετρίδη Αχιλλέα itp24105

Χαρίτο Δημήτρη itp24108

Διδάσκοντες:

Ανάργυρος Τσαδήμας

Γεώργιος Κουσιουρής

Καλλιθέα, 2025

Πίνακας Περιεχομένων

1. Εισαγωγή	3
1.1 Πλαίσιο του Έργου	3
1.2 Στόχοι του Έργου	3
2. Περιγραφή του Προβλήματος	4
3. Διάγραμμα Αρχιτεκτονικής.....	4
3.1 Γενική Περιγραφή.....	4
3.2 Οπτικό Διάγραμμα	4
4. Περιγραφή της Λύσης	5
4.1 Λειτουργία του Συστήματος.....	5
4.2 Τεχνολογίες που Χρησιμοποιούνται.....	5
5. Λεπτομέρειες Υλοποίησης.....	6
5.1 Ρύθμιση Kubernetes (microk8s).....	6
5.2 Διαμόρφωση Υπηρεσιών.....	6
5.3 Εργαλεία και Διεργασίες	6
6. Ροή Εργασίας	7
7. Προκλήσεις στην Υλοποίηση	7
7.1 Ρύθμιση RabbitMQ και Επικοινωνία με Node-RED	7
7.2 Σταθερότητα του Kubernetes Cluster	8
7.3 Ενοποίηση των Υπηρεσιών.....	8
8. Συμπεράσματα.....	8

1. Εισαγωγή

1.1 Πλαίσιο του Έργου

Το έργο υλοποιείται στο πλαίσιο του μεταπτυχιακού προγράμματος «Πληροφορική & Τηλεματική» και αφορά την ανάπτυξη μιας ολοκληρωμένης λύσης IoT για τη μέτρηση και παρακολούθηση θερμοκρασίας και υγρασίας από μια συσκευή ESP8266 με αισθητήρα θερμοκρασίας. Τα δεδομένα που συλλέγονται αποστέλλονται, αποθηκεύονται, επεξεργάζονται και παρουσιάζονται σε ένα σύστημα που βασίζεται σε Kubernetes (microk8s) με διάφορες υπηρεσίες όπως Minio, RabbitMQ, Node-RED, ThingsBoard, Prometheus και Grafana.

1.2 Στόχοι του Έργου

- **Συλλογή δεδομένων:** Από μια συσκευή ESP8266 με αισθητήρα θερμοκρασίας υγρασίας.
- **Αποθήκευση δεδομένων:** Χρήση του Minio ως αντικαταστάτη του Amazon S3 για την αποθήκευση των δεδομένων σε μορφή JSON.
- **Επεξεργασία δεδομένων:** Χρήση του RabbitMQ για τη μετάδοση γεγονότων και του Node-RED για την επεξεργασία των δεδομένων πριν αποσταλούν στο ThingsBoard.
- **Παρουσίαση δεδομένων:** Εμφάνιση των επεξεργασμένων δεδομένων στον χρήστη μέσω του ThingsBoard, ενώ τα δεδομένα του cluster παρακολουθούνται μέσω του Prometheus και παρουσιάζονται στο Grafana.

2. Περιγραφή του Προβλήματος

Το έργο επιδιώκει να αντιμετωπίσει τις ακόλουθες προκλήσεις:

- **Συλλογή και καταγραφή δεδομένων IoT:** Η συσκευή ESP8266 συλλέγει μετρήσεις θερμοκρασίας και υγρασίας σε πραγματικό χρόνο και τα στέλνει στο σύστημα.
- **Αξιόπιστη αποθήκευση:** Αποθήκευση των δεδομένων σε Minio, ώστε να είναι διαθέσιμα και ανθεκτικά σε επανεκκινήσεις.
- **Επεξεργασία και διανομή δεδομένων:** Ενεργοποίηση γεγονότων όταν καταχωρούνται δεδομένα, με χρήση RabbitMQ για τη μετάδοση αυτών των γεγονότων στο Node-RED, που αναλαμβάνει την ανάκτηση και επεξεργασία των δεδομένων.
- **Παρουσίαση δεδομένων σε χρήστες:** Η εμφάνιση των τελικών δεδομένων μέσω του ThingsBoard, καθώς και η παρακολούθηση της κατάστασης του Kubernetes cluster (όπως χρήση CPU, μνήμη, κατάσταση pods) μέσω του Prometheus και Grafana.
- **Διαχείριση πολλών σεναρίων:** Εξασφάλιση ότι το σύστημα μπορεί να διαχειριστεί πολλαπλές μετρήσεις (για παράδειγμα, μεγάλο πλήθος τιμών θερμοκρασίας και υγρασίας) και να παρουσιάσει τα δεδομένα με τρόπο κατανοητό από τους τελικούς χρήστες.

3. Διάγραμμα Αρχιτεκτονικής

3.1 Γενική Περιγραφή

Το σύστημα βασίζεται σε microk8s και αποτελείται από τις παρακάτω βασικές υπηρεσίες:

- **ESP8266 Device:** Η συσκευή IoT που συλλέγει δεδομένα (θερμοκρασία, υγρασία, χρονική σήμανση) και τα αποστέλλει στον Minio μέσω API.
- **Minio:** Λειτουργεί ως αποθηκευτικός χώρος, όπου κάθε φορά που αποθηκεύονται δεδομένα, ενεργοποιείται ένα γεγονός.
- **RabbitMQ:** Διαχειρίζεται τα γεγονότα που παράγει ο Minio, μεταφέροντάς τα μέσω ενός exchange σε συγκεκριμένες ουρές.

- **Node-RED:** Ακούει τα γεγονότα από το RabbitMQ, ανακτά τα δεδομένα από το Minio μέσω HTTP GET, τα επεξεργάζεται χρησιμοποιώντας function nodes και στη συνέχεια τα αποστέλλει στο ThingsBoard μέσω HTTP POST.
- **ThingsBoard:** Λαμβάνει τα επεξεργασμένα δεδομένα και τα παρουσιάζει στους χρήστες μέσω γραφημάτων και πινάκων.
- **Prometheus και Grafana:** Παρακολουθούν την απόδοση και την κατάσταση του Kubernetes cluster, συλλέγοντας δεδομένα όπως χρήση πόρων, κατάσταση των pods κ.λπ., και εμφανίζουν τα δεδομένα αυτά σε dashboards.

4. Περιγραφή της Λύσης

4.1 Λειτουργία του Συστήματος

- **Συλλογή Δεδομένων:** Η συσκευή ESP8266 στέλνει δεδομένα θερμοκρασίας και υγρασίας σε μορφή JSON στον Minio.
- **Αποθήκευση και Ενεργοποίηση Γεγονότων:** Ο Minio αποθηκεύει τα δεδομένα σε persistent volumes και, με την καταχώρησή τους, ενεργοποιεί ένα event.
- **Διανομή Γεγονότων:** Το event μεταδίδεται στο RabbitMQ μέσω ενός exchange.
- **Επεξεργασία Δεδομένων:** Το Node-RED ακούει την ουρά στο RabbitMQ, ανακτά τα δεδομένα από το Minio μέσω HTTP GET, τα επεξεργάζεται και τα μεταδίδει στο ThingsBoard μέσω HTTP POST.
- **Παρουσίαση Δεδομένων:** Το ThingsBoard δέχεται τα δεδομένα και τα παρουσιάζει σε ένα dashboard για τους χρήστες, ενώ ταυτόχρονα το Prometheus συλλέγει μετρήσεις από το Kubernetes cluster και το Grafana εμφανίζει αυτά τα δεδομένα σε αντίστοιχο dashboard.

4.2 Τεχνολογίες που Χρησιμοποιούνται

- **Kubernetes (microk8s):** Για τη διαχείριση και τον συντονισμό όλων των containerized υπηρεσιών.
- **Docker:** Για τη δημιουργία των container για κάθε υπηρεσία (ThingsBoard, Minio, RabbitMQ, Node-RED, Prometheus, Grafana).
- **Minio:** Για την αποθήκευση των δεδομένων.
- **RabbitMQ:** Για τη διαχείριση και μετάδοση των γεγονότων.
- **Node-RED:** Για την επεξεργασία των δεδομένων.
- **ThingsBoard:** Για την παρουσίαση και ανάλυση των δεδομένων.

- **Prometheus & Grafana:** Για την παρακολούθηση της απόδοσης του cluster και την οπτικοποίηση των μετρήσεων.

5. Λεπτομέρειες Υλοποίησης

5.1 Ρύθμιση Kubernetes (microk8s)

- **Εγκατάσταση:** Το microk8s εγκαταστάθηκε στον server και ενεργοποιήθηκαν τα απαραίτητα addons (observability που περιλαμβάνει Prometheus).
- **Διαχείριση Πόρων:** Όλες οι υπηρεσίες διαχειρίζονται μέσω YML αρχείων που βρίσκονται στο φάκελο k8s (π.χ., grafana, thingsboard, minio, node-red, rabbitmq, keycloak).
- **Storage:** Έχουν δημιουργηθεί αρχεία για PersistentVolumes (PV) και PersistentVolumeClaims (PVC) ώστε να εξασφαλιστεί η διαθεσιμότητα των δεδομένων ακόμα και μετά από επανεκκίνηση των υπηρεσιών.

5.2 Διαμόρφωση Υπηρεσιών

- **Minio:** Ορίζεται με τα αρχεία minio-deployment.yml, minio-service.yml, και τα σχετικά αρχεία για PVC/PV, ώστε να αποθηκεύει τα JSON δεδομένα.
- **RabbitMQ:** Ρυθμίζεται μέσω statefulset και service ώστε να διατηρείται η κατάσταση των ουρών και exchanges.
- **Node-RED:** Υλοποιείται με YML αρχεία για deployment, όπου δημιουργούνται flows που ακούνε τα events από το RabbitMQ και πραγματοποιούν HTTP GET και POST αιτήσεις.
- **ThingsBoard:** Ρυθμίζεται ώστε να δέχεται δεδομένα μέσω HTTP POST, να τα αποθηκεύει και να τα εμφανίζει μέσω dashboards.
- **Prometheus & Grafana:** Το Prometheus συλλέγει metrics από το cluster, ενώ το Grafana, συνδεδεμένο ως data source στο Prometheus, παρουσιάζει τα δεδομένα με προσαρμοσμένα dashboards.

5.3 Εργαλεία και Διεργασίες

- **Διαχείριση Κώδικα:** Όλος ο κώδικας, τα YML αρχεία, τα Dockerfiles και τα scripts διαχειρίζονται μέσω Git (σε ένα Git repository).

- **Αυτοματοποιημένες Εγκαταστάσεις:** Έχουν δημιουργηθεί scripts (π.χ. το start-yaml.sh) για την αυτόματη εκτέλεση των YAML αρχείων, καθιστώντας τη διαδικασία εγκατάστασης εύκολη και επαναλαμβανόμενη.
- **Εγχειρίδια Χρήσης (README):** Έχουν δημιουργηθεί εγχειρίδια για το πώς να εγκαταστήσετε και να δοκιμάσετε τις υπηρεσίες, μαζί με πληροφορίες για test accounts.

6. Ροή Εργασίας

1. **Αποστολή Δεδομένων από το ESP8266:**
 - a. Η συσκευή στέλνει δεδομένα θερμοκρασίας, υγρασίας και χρονικής σήμανσης στον Minio μέσω API.
2. **Αποθήκευση και Ενεργοποίηση Γεγονότων στο Minio:**
 - a. Κάθε φορά που τα δεδομένα αποθηκεύονται, ο Minio ενεργοποιεί ένα event.
3. **Διανομή Γεγονότων μέσω RabbitMQ:**
 - a. Το event αποστέλλεται στο RabbitMQ μέσω ενός exchange.
4. **Επεξεργασία στο Node-RED:**
 - a. Το Node-RED ακούει το RabbitMQ, ανακτά τα δεδομένα από το Minio μέσω HTTP GET, τα επεξεργάζεται και τα στέλνει στο ThingsBoard μέσω HTTP POST.
5. **Παρουσίαση Δεδομένων στο ThingsBoard:**
 - a. Το ThingsBoard παρουσιάζει τα δεδομένα στους χρήστες μέσω dashboards με και widgets.
6. **Παρακολούθηση του Kubernetes Cluster με Prometheus και Grafana:**
 - a. Το Prometheus συλλέγει metrics για τους κόμβους του Kubernetes cluster και το Grafana εμφανίζει αυτές τις πληροφορίες σε dashboards.

7. Προκλήσεις στην Υλοποίηση

7.1 Ρύθμιση RabbitMQ και Επικοινωνία με Node-RED

- Η σωστή διαμόρφωση του RabbitMQ για να διαχειρίζεται τα γεγονότα από το Minio και να τα μεταδίδει στο Node-RED ήταν μια από τις μεγαλύτερες προκλήσεις.
- Ο συγχρονισμός μεταξύ των υπηρεσιών (Minio, RabbitMQ, Node-RED) απαιτούσε προσεκτική ρύθμιση των exchanges και queues.

7.2 Σταθερότητα του Kubernetes Cluster

- Η υλοποίηση του συστήματος στο microk8s απαιτούσε αποτελεσματική διαχείριση πόρων (memory, CPU) και σωστή διαμόρφωση persistent storage ώστε να διατηρούνται τα δεδομένα.

7.3 Ενοποίηση των Υπηρεσιών

- Η ομαλή επικοινωνία μεταξύ των διαφόρων υπηρεσιών (Minio, RabbitMQ, Node-RED, ThingsBoard) απαιτούσε προσεκτικό συγχρονισμό των APIs και των ρυθμίσεων ασφάλειας.
- Επίσης, η διαχείριση του CI/CD για τη διανομή των Docker images και των YML αρχείων απαιτούσε αυτοματοποίηση και συνεχή ενσωμάτωση (Continuous Integration/Continuous Deployment).

8. Συμπεράσματα

Η λύση που υλοποιήθηκε παρέχει ένα ολοκληρωμένο σύστημα IoT που συλλέγει, αποθηκεύει, επεξεργάζεται και παρουσιάζει δεδομένα από αισθητήρα θερμοκρασίας. Χρησιμοποιώντας ένα ευέλικτο περιβάλλον Kubernetes, τα δεδομένα είναι ανθεκτικά και παρακολουθούνται σε πραγματικό χρόνο μέσω των εργαλείων Prometheus και Grafana. Η ομαλή ενοποίηση των υπηρεσιών Minio, RabbitMQ, Node-RED και ThingsBoard εξασφαλίζει ότι οι τελικοί χρήστες μπορούν να έχουν πρόσβαση σε ακριβείς και ενημερωμένες πληροφορίες για τις μετρήσεις του IoT συστήματος.