

# UNIVERSITÀ DEGLI STUDI DELL'INSUBRIA

## DIPARTIMENTO DI SCIENZE TEORICHE E APPLICATE

### CORSO DI STUDIO TRIENNALE IN INFORMATICA

**Tesi di Laurea di:** Lorenzo Fusè

**Matricola:** 753168

**Relatore:** Prof.ssa Alessandra Rizzardi

**Correlatore:** Prof.ssa Sabrina Sicari

**Anno accademico:** 2024/2025

## **Riassunto: Configurazione ed implementazione di una rete basata su microservizi tramite la piattaforma KubeEdge**

Il presente lavoro di tesi ha avuto come obiettivo la sperimentazione di un'infrastruttura edge-cloud basata su **microservizi** mediante la piattaforma **KubeEdge**, estensione di Kubernetes orientata all'esecuzione e al coordinamento di componenti applicativi sui nodi periferici. Il caso di studio riguarda la raccolta di dati sulla qualità dell'aria indoor e il monitoraggio delle risorse delle macchine virtuali. L'applicazione in tale contesto è motivata dalla crescente diffusione delle architetture distribuite, che permettono di ottimizzare l'uso delle risorse e migliorare la sicurezza nei contesti Edge e Internet of Things. In questo ambito, KubeEdge consente di mantenere un piano di controllo coerente con l'ecosistema cloud-native, estendendo le funzionalità di orchestrazione verso l'edge e favorendo una gestione modulare dei servizi.

L'attività svolta ha mirato a verificare, in un ambiente controllato, la fattibilità e il comportamento dell'architettura lungo l'intera infrastruttura realizzata: dalla generazione dei dati, al trasporto crittografato mediante il modello *publish/subscribe*, fino alla loro persistenza e visualizzazione. La validazione si è concentrata su indicatori di prestazione, utilizzo delle risorse e robustezza operativa, individuando i principali limiti della soluzione proposta.

La realizzazione del sistema ha richiesto l'integrazione di diverse tecnologie open-source per garantire modularità, scalabilità e sicurezza:

- **KubeEdge:** estensione di Kubernetes che abilita l'orchestrazione dei microservizi anche nei nodi periferici.
- **Microk8s:** distribuzione leggera di Kubernetes, utilizzata per la gestione del cluster e delle risorse. Utilizzato come distribuzione K8s sottostante, poiché KubeEdge non include al suo interno un cluster Kubernetes.
- **Docker** e **Containerd:** strumenti impiegati rispettivamente per la fase di build e per l'esecuzione dei container.
- **MQTT - Mosquitto:** protocollo di comunicazione asincrona basata sul modello *publish/subscribe*, adottato per lo scambio dei dati.
- **Elliptic Curve Cryptography - AES-256:** meccanismi crittografici per garantire confidenzialità, integrità e autenticità dei messaggi utilizzati per la generazione delle chiavi e la cifratura effettiva.

- **PostgreSQL - Grafana:** utilizzati per la persistenza dei dati e la loro rappresentazione tramite dashboard personalizzate.

L'architettura implementata si articola su due livelli principali: i **nodi Edge**, che simulano il comportamento dei sensori e pubblicano i dati, e il **nodo CloudCore**, che svolge la funzione di coordinamento centrale, gestendo la pubblicazione, la validazione dei messaggi ricevuti, la persistenza e la visualizzazione dei dati.

Il flusso operativo prevede che i nodi Edge pubblichino sia l'utilizzo delle risorse di sistema sia i parametri ambientali, questi ultimi cifrati e firmati digitalmente, per poi essere trasmessi tramite il protocollo **MQTT**, mediante il broker **Mosquitto**. Il nodo CloudCore riceve i messaggi, ne verifica l'autenticità ed effettua la decifratura e infine memorizza i dati nel database **PostgreSQL**, rendendoli disponibili per la visualizzazione grafica tramite **Grafana**. La sicurezza della comunicazione è stata garantita mediante l'impiego di **Elliptic Curve Diffie-Hellman Ephemeral (ECDHE)**, un meccanismo che consente la generazione di chiavi simmetriche temporanee per ciascuna sessione di scambio. In questo modo i nodi Edge e il CloudCore sono in grado di concordare dinamicamente una chiave condivisa, senza trasmetterla direttamente sul canale, riducendo così il rischio di eventuali compromissioni. La chiave simmetrica così ottenuta viene quindi utilizzata dall'algoritmo **AES-256** utilizzato in modalità **Cipher Feedback (CFB)** per la cifratura dei messaggi, mentre **Elliptic Curve Digital Signature Algorithm (ECDSA)** assicura integrità e autenticità attraverso la firma digitale.

La fase di visualizzazione è stata realizzata con **Grafana**, attraverso dashboard dedicate all'analisi dei dati raccolti. Costituiscono un elemento centrale per l'analisi del sistema, in quanto facilitano l'individuazione di anomalie e per validare i risultati ottenuti dalle simulazioni.

Sono state predisposte due principali tipologie di viste:

- **Dashboard Indoor Air Quality:** permettono di monitorare in tempo reale i valori di **CO<sub>2</sub>** e **temperatura**, visualizzandone l'andamento temporale e consentendo confronti per stanza o zona.
- **Dashboard VM Metrics:** forniscono informazioni sullo stato delle macchine virtuali che compongono l'architettura, riportando indicatori relativi a **CPU**, **RAM**, **traffico di rete** e **disco**.

La creazione di pannelli personalizzati ha permesso una validazione immediata dei dati e delle prestazioni del sistema, garantendo una rappresentazione chiara e comprensibile della situazione dell'edificio monitorato in tempo reale. Le dashboard prevedono soglie predefinite, collegate ad un sistema di alert automatico che notifica il superamento delle condizioni stabilite.

La valutazione del sistema è stata condotta considerando parametri di prestazioni e di robustezza, con l'obiettivo di validare l'approccio implementato e individuare i principali limiti applicativi. Le misurazioni sono state effettuate in un ambiente controllato, analizzando il comportamento del sistema in termini di tempi di risposta, capacità di elaborazione dei messaggi e utilizzo delle risorse di calcolo.

- **Latenza end-to-end:** calcolata registrando la differenza di tempo tra la pubblicazione dei messaggi da parte dei nodi Edge e la loro persistenza nel database. L'analisi ha mostrato una latenza mediana di circa **37 ms**, con il 95% dei messaggi consegnati entro **76 ms**.
- **Throughput:** stimato sul numero di messaggi elaborati nell'unità di tempo durante le simulazioni dei sensori virtuali. I risultati hanno evidenziato un flusso medio di **1,8 messaggi/s per nodo Edge**, con un totale aggregato di circa **5,4 messaggi/s** per l'intero scenario monitorato.
- **Utilizzo delle risorse:** le metriche relative a CPU, RAM, disco e rete sono state raccolte tramite microservizi dedicati ed esaminate tramite Grafana. I valori registrati hanno mostrato un carico sostenibile confermando la suddivisione dei ruoli ricoperti dalle VM all'interno dell'architettura.
- **Robustezza del sistema:** la stabilità è stata verificata analizzando la presenza di messaggi duplicati e monitorando la percentuale di pacchetti con latenza superiore a una soglia prefissata (200 ms).

Il sistema presenta alcune limitazioni applicative, legate all'ambiente di test simulato quali:

- **QoS = 0:** assenza di garanzie di consegna in caso di errori di trasmissione.
- **Gestione dei Pod:** mancanza di meccanismi di *autorecovery*, con pod destinati a terminare in caso di arresto anomalo.
- **Gestione delle chiavi crittografiche:** condivisione delle chiavi pubbliche senza l'impiego di pratiche di sicurezza distribuita.
- Assenza di meccanismi di **fault tolerance**, **autoscaling** e **multi-tenant**, non previsti nello scenario sperimentale.

Sulla base delle limitazioni evidenziate, sono stati individuati alcuni possibili sviluppi futuri:

- integrazione di **dataset reali** provenienti da sensori fisici, al fine di validare l'architettura in scenari concreti;
- implementazione di pratiche avanzate per la **gestione distribuita delle chiavi crittografiche**, con protocolli di scambio sicuro;
- introduzione di meccanismi di **autoscaling** e **fault tolerance** per aumentare l'affidabilità e la resilienza del sistema in condizioni di carico variabile;
- integrazione di modelli di **Edge AI**, volti ad anticipare anomalie o prevedere l'andamento dei parametri monitorati direttamente a livello edge.