Realizzazione di una infrastruttura in cloud per ambiti DevOps

CANDIDATO: MATTIA CARLINO

RELATORE: PROF. ENRICO BINI

TUTOR AZIENDALE: FRANCESCO LORUSSO

Università degli Studi di Torino

Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

Obiettivo dello stage

Lo stage ha come scopo quello di creare una soluzione che permetta il rilascio automatizzato di un'applicazione basata su microservizi.

INFRASTRUTTURA IN CLOUD

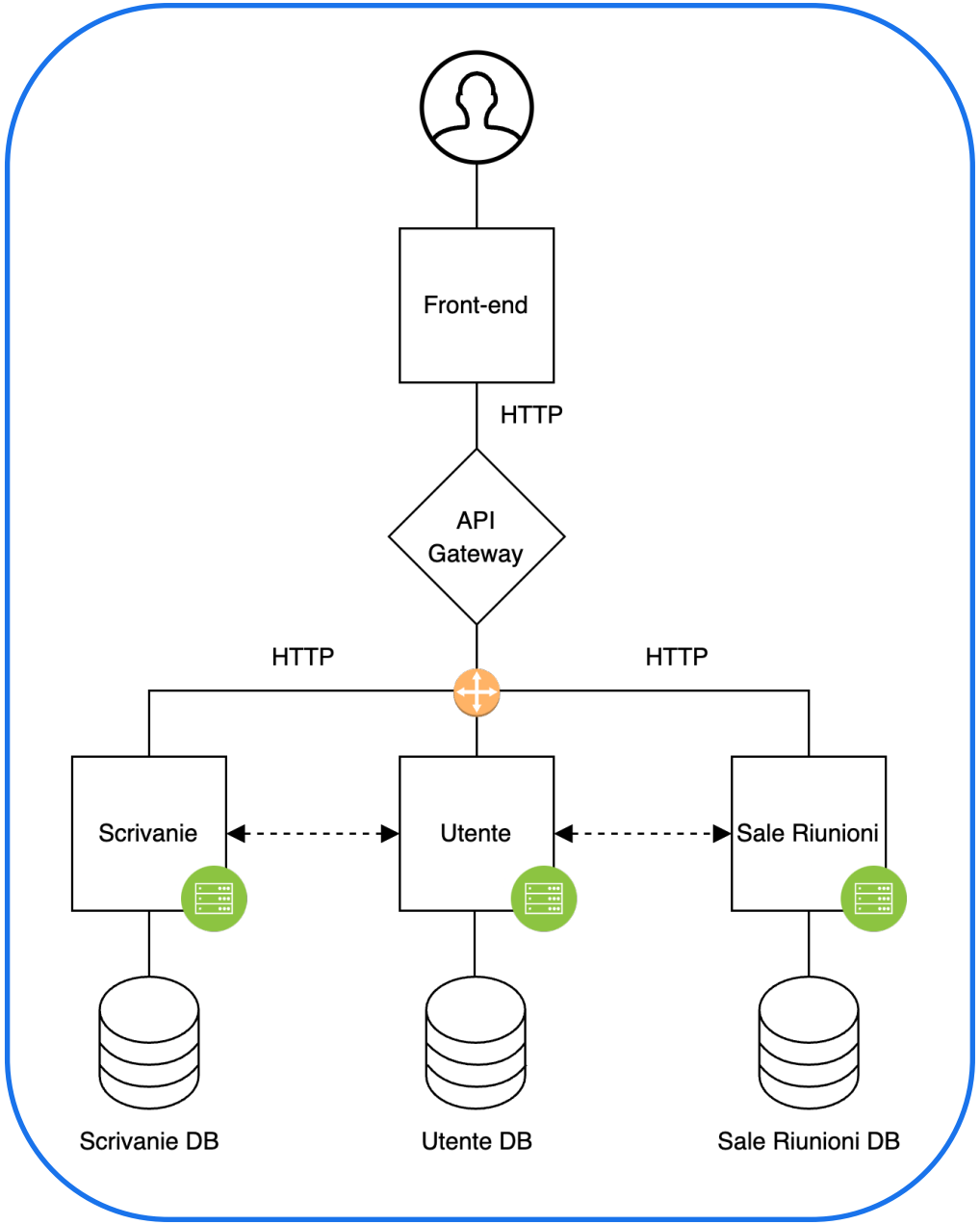
RILASCIO AUTOMATIZZATO DI UN’APPLICAZIONE

UTILIZZO DI METODOLOGIE DEVOPS

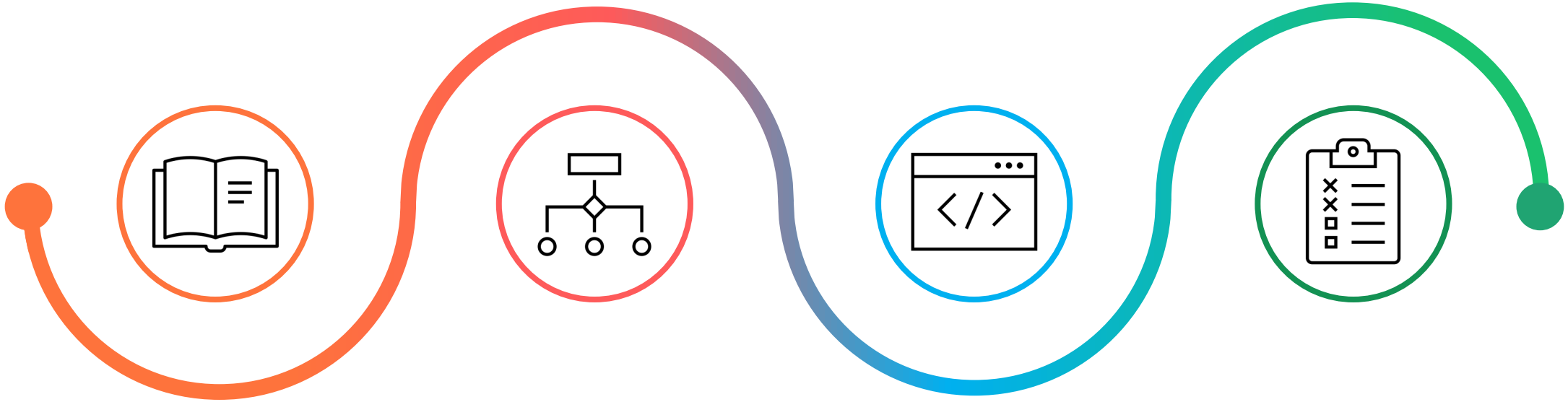
Rilascio in Produzione

Applicazione “Prenotazioni Aziendali” rilasciata e composta da cinque **micro-servizi**:

- Tre di back-end, ciascuno collegato ad un proprio database.
- Uno di front-end.
- Un API Gateway, che si occupa di smistare le richieste.



Fasi del progetto



Studio

Studio delle tecnologie

Progettazione

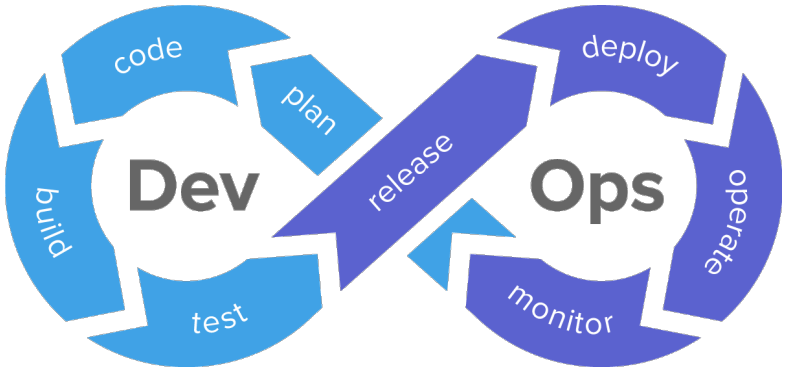
Analisi requisiti e progettazione della soluzione

Implementazione

Implementazione della soluzione

Test Finali

Verifica del corretto funzionamento



Approccio DevOps

Cos'è il DevOps?

DevOps è la combinazione di sviluppo (Dev) e operazioni (Ops) che ha portato a nuove modalità di comunicazione, collaborazione e integrazione continua tra gli ambienti di sviluppo.

A livello pratico

Descrive gli approcci da adottare per accelerare i processi che consentono di agevolare il processo di sviluppo e rilascio del software.

Si concentra su **tre aree**:

- Automazione di infrastrutture
- Rilascio continuo
- Affidabilità applicativo

Cloud Computing

È l'esecuzione di un carico di lavoro in ambiente cloud.

PRIMA - METODO TRADIZIONALE

- Gestione delle risorse sul posto:
  - Costi fissi
  - Problemi di configurazione (difficile replicazione, soggetta ad errori)
- Difficoltà nell'avere applicazioni scalabili:
  - Sotto/sovra utilizzo delle risorse allocate
  - Applicazione e sistema poco performanti

DOPO - CLOUD

- Risorse allocabili online su richiesta
  - Costi variabili all'utilizzo (Pay As You Go)
  - Configurazione semplificata (Infrastruttura replicabile)
- Applicazione scalabile in base alle richieste di traffico
- Monitoraggio delle risorse

Infrastructure as Code (IaC)

Si intende il processo con il quale è possibile creare e gestire un'infrastruttura IT utilizzando file di configurazione.

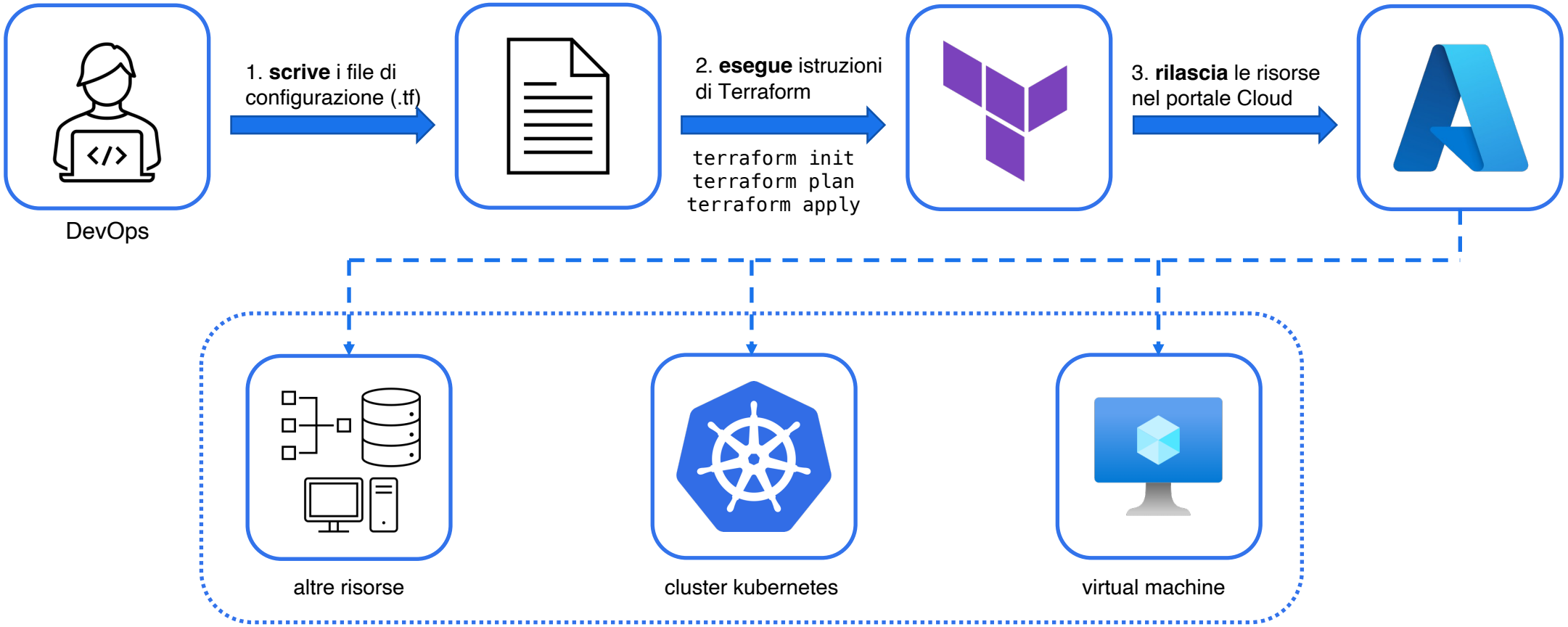
Perché la IaC?

Per prevenire gli svantaggi del metodo "tradizionale" (prima del cloud).

- **Efficienza**: vengono semplificati i processi di configurazione, manutenzione e gestione dell'infrastruttura.
- **Replicabilità**: i file possono essere riutilizzati per creare infrastrutture identiche
- **Gestione delle versioni**: è facile ripristinare versioni passate e determinare chi ha apportato modifiche ai file.

In questa soluzione, è stato utilizzato lo strumento **Terraform** per realizzare l'infrastruttura IaC all'interno del cloud provider **Azure.**

Workflow con Terraform



Ora che l'infrastruttura è pronta, si può definire il processo di rilascio automatizzato.

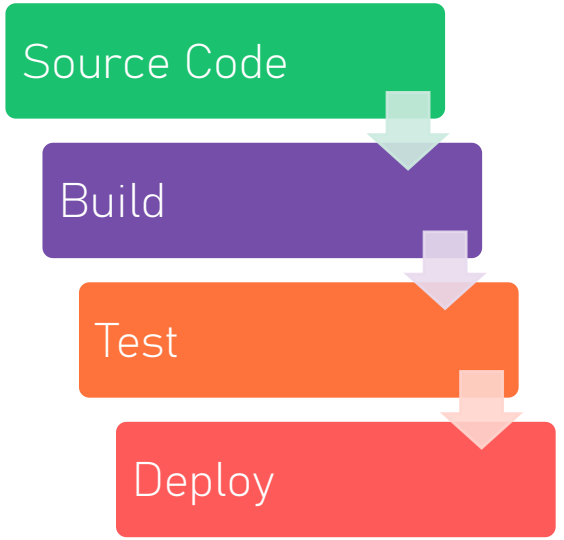
Continuous Integration - Continuous Delivery

Con CI/CD si intende l'insieme dei passaggi automatizzati che vengono eseguiti per rilasciare una nuova versione software, intregrazione delle modifiche al codice, build di tale codice e rilascio dell'applicativo.

Come avviene questo?

Tramite la realizzazione di Pipeline, un insieme di attività eseguite sequenzialmente ed automaticamente a seguito di un **trigger**.

Una pipeline è costituita da più fasi, suddivisibili in **job** (cosa fare) e **stage** (quando fare).



All'interno di questa soluzione è stato utilizzato lo strumento **Jenkins**.

Containerizzazione

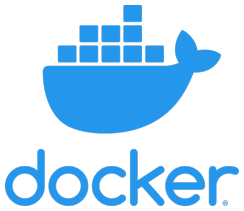
Processo di implementazione del software che suddivide l’applicazione in container, componenti eseguibili standardizzati che combinano il codice sorgente dell'applicazione con le librerie e le dipendenze necessarie al loro interno.

Perché i container?

Garantiscono portabilità, modularità e un’esecuzione efficiente.

A questo scopo, tramite lo strumento Docker e la definizione di Dockerfile, sono state create le immagini docker di ogni micro-servizio dell’applicativo.

Le immagini sono state salvate all’interno del Docker Repository **"JFrog Artifactory".**



Orchestrazione

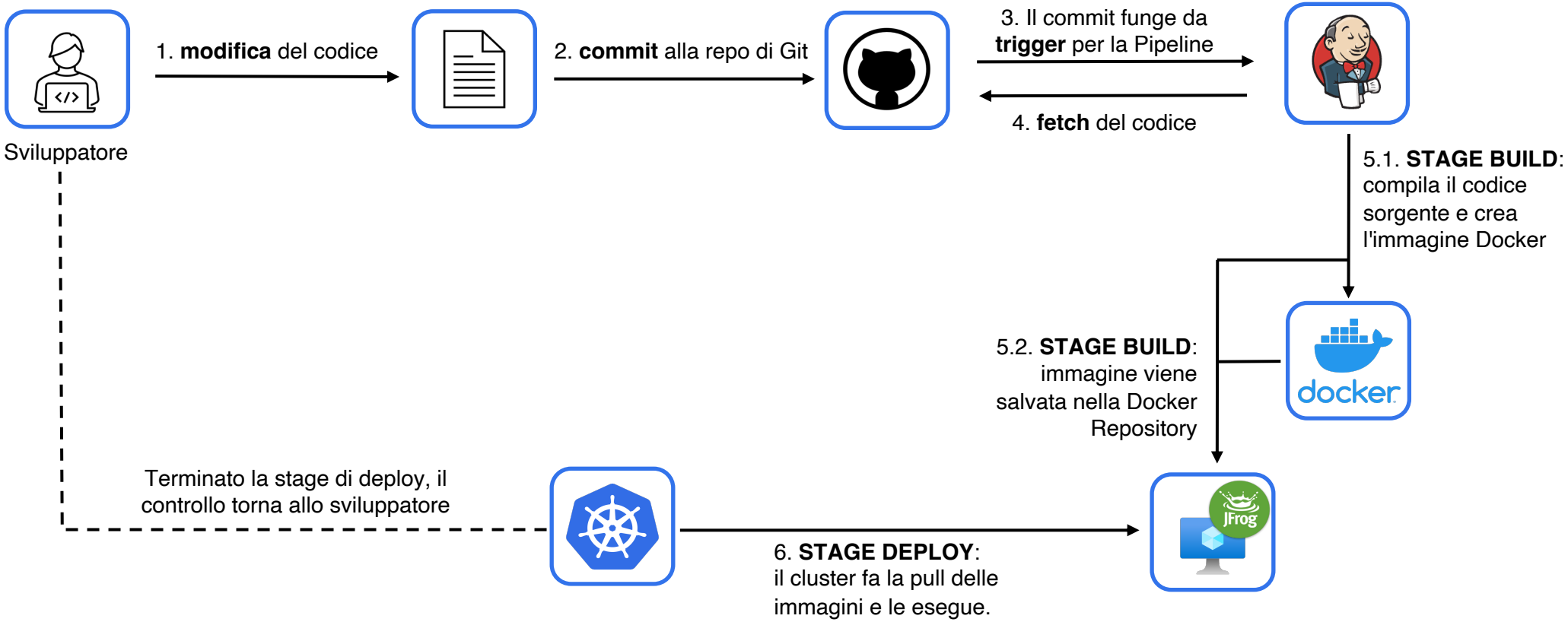
È l'**automatizzazione** dei processi di deployment, gestione, scalabilità e networking dei container presenti in un cluster.

Vantaggi:

- Bilanciamento del carico
- Allocazione dinamica delle risorse
- Networking semplificata

Per la creazione, gestione e comunicazione dei diversi container della soluzione è stato usato **Kubernetes**.

Workflow della Pipeline



Virtual Machine ed il cluster si trovano nella IaC definita in precedenza dal DevOps.

Possibili sviluppi futuri

MONITORAGGIO

SICUREZZA

DATABASE ESTERNO AL CLUSTER

Q&A

Grazie per l’attenzione