Buongiorno,

Sono Matteo Sirri e il mio elaborato riguarda la creazione di una piattaforma riutilizzabile in contesto aziendale.

Di seguito l’agenda degli argomenti che andrò a trattare e in particolare dopo una breve introduzione vi descriverò i componenti principali della piattaforma, come avviene la gestione e il rilascio dei componenti finali per poi chiudere con gli obiettivi raggiunti.

Il prodotto software realizzato è una piattaforma *cloud based* che è stata progettata con lo scopo di supportare le esigenze e le operazioni di business e di management delle aziende e dei liberi professionisti.

Essendo le esigenze di una azienda uniche nel suo genere, la piattaforma non è stata progettata come un sistema autonomo pronto ad essere utilizzato in un contesto specifico ma è stata ideata come una base/delle fondamenta di un sistema più complesso che verrà costruito in base alle esigenze del cliente. Pertanto, i servizi erogati dalla piattaforma sono quelli più comuni come, ad esempio, la gestione degli utenti oppure l’invio di notifiche di vario tipo.

Inoltre, la realizzazione di una piattaforma di questo tipo permetterà anche di ridurre i costi e i tempi per costruire nuove soluzioni per futuri clienti visto che è già stata predisposta tutta la piattaforma di base.

La piattaforma è organizzata con una architettura a microservizi in quanto la scalabilità del sistema è un fattore fondamentale.

Gli elementi principali della piattaforma sono l’Api Web che è una applicazione web server che offre una interfaccia per accedere ai vari servizi; il microservizio doc che offre l’accesso ad una interfaccia interattiva basata su un file di specifica OpenAPI e il microservizio mailer che permette di inviare e-mail personalizzate agli utenti attraverso la renderizzazione di template grafici sfruttando il template Engine Nunjucks. Questo microservizio poi si interfaccia con il servizio esterno Simple E-mail Service, facente parte degli Amazon Web services, con il quale vengono effettivamente inviate le e-mail agli utenti.

Poi, abbiamo il message broker che permette l’interazione tra l’Api Web e il microservizio Mailer. Nella piattaforma viene usato il protocollo AMQP e un broker RabbitMQ. Vista la possibilità di dover gestire numerose e-mail contemporaneamente, è stato deciso di utilizzare il modello a consumers Concorrenti potendo così sfruttare potenzialmente tante istanze del microservizio mailer collegate ad uno stesso canale del broker.

Ed infine un un database server basato su mongoDB che permette di gestire lo storage di dati.

Sarà poi presente anche una applicazione client che si interfaccerà con la piattaforma e l’utente ma il suo sviluppo non è stato trattato nel mio lavoro.

Sulla base di questa piattaforma verranno poi costruiti e aggiunti i componenti necessari per supportare in maniera efficace le esigenze del cliente.

Di seguito alcuni dettagli sul design della Api Web.

In questo componente le funzionalità sono incapsulate in moduli relazionati tra loro.

Ci sono quattro moduli: lo user module che si occupa della gestione delle informazioni degli utenti; il demo module che permette agli utenti di iscriversi a un generico servizio di prova; il Mail module che controlla la logica di gestione delle mail e pubblicazione di messaggi sul broker.

Ed infine l’auth module. Questo si occupa dei servizi di autenticazione e autorizzazione degli utenti.

In particolare, l’autenticazione si basa su una coppia di credenziali: e-mail e password e per questo motivo sono state implementate procedure di archiviazione sicura delle password attraverso l’uso congiunto di funzioni di hash e salt.

Una volta che l’utente accede alla piattaforma ottiene una coppia di token in formato json web token detti access token e refresh token. L’access token identifica l’autorizzazione di un utente, ha una durata breve e permette di accedere alle risorse protette; mentre il refresh token permette di ottenere nuovi access token.

Sono stati poi implementati un modello di controllo sugli accessi attraverso delle access control lists basate sui ruoli dell’utente e il protocollo OAuth2.0 per gestire le funzionalità di login e registrazione con provider di terze parti, in particolare google e facebook.

I componenti della piattaforma vengono gestiti utilizzando i container Docker per via della loro leggerezza e portabilità.

La creazione dei container avviene attraverso il meccanismo del multi-stage build. Questo semplifica la generazione di container perché grazie ad un solo dockerFile è stato possibile creare in maniera rapida delle immagini ottimizzate da utilizzare nelle fasi di test e di deploy sul production server.

I vari componenti della piattaforma verranno poi eseguiti in container separati orchestrati dal servizio Docker Compose che permette di definire quali servizi devono essere eseguiti e le loro proprietà a runtime attraverso un file di configurazione.

Per gestire le fasi di rilascio dei componenti è stata implementata una pipeline di Continuous integration/ continuous deployment.

Nel dettaglio si ha che la pipeline viene avviata ogni volta che uno sviluppatore effettua la pubblicazione di un commit sul repository remoto condiviso. Una volta fatto ciò abbiamo l’esecuzione del processo di integrazione continua in cui viene effettuata la build di tutto il codice aggiornato e vengono eseguiti gli unit test e gli integration test per verificare che l’aggiornamento non introduca bug o errori nel codice preesistente. Se non si sono verificati errori viene eseguita la pipeline di continuos deployment in cui vengono pubblicate le immagini sul registro docker remoto ed infine avviene il deploy sul production server in cui vengono scaricate le varie immagini ed eseguite con docker compose.

L’uso di questa pipeline ha permesso di impostare i lavori secondo un meccanismo ispirato alla metodologia Agile e alle pratiche DevOps riuscendo così ad di ottimizzare il processo di sviluppo, integrazione e rilascio di software, la collaborazione fra i vari gruppi di lavoro e la stabilità del prodotto realizzato.

In conclusione, il risultato ottenuto è una piattaforma riutilizzabile e personalizzabile sulla quale costruire nuove soluzioni per i clienti riducendo i tempi e costi di sviluppo.

La piattaforma è stata sviluppata durante il mio periodo di tirocinio presso Fama Labs, un’azienda erogatrice di soluzioni cloud su commissione e un esempio di utilizzo della piattaforma risiede nella Medical Adaptive Platform, un nuovo servizio erogato da Fama Labs. Questo è una

suite di soluzioni digitali per le Società Scientifiche, aziende biofarmaceutiche e professionisti

sanitari che offre servizi per gestire l’amministrazione della società, organizzare eventi di formazione a distanza, corsi e congressi.

La piattaforma realizzata verrà inoltre sfruttata dalla stessa azienda in una applicazione interna per il management dei servizi erogati e delle risorse.

Grazie per l’attenzione. Se ci sono domande sono a vostra disposizione.