

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA

CORSO DI LAUREA IN INFORMATICA



Containerizzazione di Malware Dashboard

Relazione finale di stage

Relatore

Prof. Tullio Vardanega

Laureando

Andrei Petrov

ANNO ACCADEMICO 2016-2017

Sommario

//TODO

Indice

1	L'Azienda	1
1.1	IKS	1
1.2	Profilo dell'azienda	2
1.2.1	Servizi e prodotti offerti	2
1.2.2	Struttura organizzativa	6
1.2.3	Processi aziendali	6
1.3	Rapporto con l'innovazione	8
2	L'azienda e gli stage	11
2.1	Il valore aggiunto di uno stagista	12
2.2	Alcuni temi di stage	12
2.2.1	AIOps e Machine Learning	12
2.2.2	DevOps Automazione	12
2.2.3	Sviluppo moduli evolutivi in ambito antifrode	12
2.3	Il progetto proposto	13
2.3.1	Motivazioni	13
2.3.2	Obiettivi aziendali	15
2.3.3	Obiettivi personali	16
2.4	Piano di lavoro	17
2.5	Vincoli	17
2.5.1	Vincoli temporali	17
2.5.2	Vincoli tecnologici	18
3	Lo svolgimento dello stage	21
3.1	Metodo di lavoro	21
3.2	Attività di formazione	21
3.3	Analisi dei requisiti	21
3.3.1	Classificazione dei requisiti	21
3.3.2	Requisiti	21
3.3.2.1	Funzionali	21
3.3.2.2	Non funzionali	21
3.4	Progettazione e realizzazione	21
3.4.1	Scelte progettuali	21
3.4.2	Visione architetturale a microservizi	21
3.4.3	Codifica	21
3.4.4	Test	21
3.4.4.1	Test di carico	21
3.4.4.2	Test di durata	21

4	Valutazioni retrospettive	23
4.1	Obiettivi raggiunti	23
4.2	Problematiche riscontrate	23
4.3	Bilancio formativo	23
4.3.1	Il prima	23
4.3.2	Il dopo	23
4.4	Valutazione critica del Corso di Laurea	23
	Glossario	25
	Acronimi	27
	Riferimenti	29

Elenco delle figure

1.1	Visione a processo della gestione del rischio. Immagine tratta da: http://bit.ly/2rh3V0A	2
1.2	Flusso di lavoro durante lo svolgimento di un audit. Immagine tratta da: http://bit.ly/2rdFhfv	2
1.3	Visione grafica del concetto di difesa perimetrale. Immagine tratta da: http://bit.ly/2s834O2	3
1.4	Visione del ciclo di vita del processo business continuity. Immagine tratta da: http://bit.ly/2qvCmgP	3
1.5	Vista a confronto tra un ambiente server bare metal e virtualizzato. Immagine tratta da: http://bit.ly/2qvtLLk	4
1.6	Visione della gestione di servizio in prospettiva del Framework ITIL . Immagine tratta da: http://bit.ly/2qvNryk	4
1.7	Visione architetturale a monolite e microservizi a confronto. Immagine tratta da: http://bit.ly/2rh1niY	5
1.8	Organigramma aziendale	7
1.9	Rappresentazione grafica del coinvolgimento del Cliente e i corrispettivi livelli degli interventi del gruppo commerciale, tecnico, direzionale e di supporto nella gestione di un'offerta di progetto.	8
1.10	Il legame attivo tra innovazione e il processo del cambiamento e gestione della conoscenza. Immagine tratta da: http://bit.ly/2qty3XD	9
2.1	Sia gli sviluppatori che i professionisti IT sono portatori di valore: un feedback che coinvolge ambe le parti è essenziale. Immagine tratta da: http://bit.ly/2rM9IBQ	13
2.2	Il DevOps abilita l'automazione del processo di rilascio del software e i cambi dell'infrastruttura IT. Immagine tratta da: http://bit.ly/2rsw9nm	14
2.3	Esempio di un sistema a microservizi. Immagine tratta da: http://bit.ly/2qNXKxj	15
2.4	I microservizi permettono di scalare orizzontalmente per reggere ai più esigenti carichi di lavoro. Immagine tratta da: http://bit.ly/2qI50LR	16
2.5	Le parti costituenti la piattaforma Docker sono: il demone, il client e il registry Docker. Immagine tratta da: http://bit.ly/2rmkt7g	18
2.6	Le componenti architetturali di Kubernetes sono: API server, Scheduler, Replication Controller, Kubelet, Kube proxy, Database Etcd. Immagine tratta da: http://bit.ly/2s4eKUR	19

Elenco delle tabelle

Capitolo 1

L'Azienda

Il presente capitolo è dedicato alla presentazione dell'Azienda che mi ha ospitato nella propria sede per uno stage obbligatorio per la Laurea triennale.

Ho strutturato la presentazione in tre parti: nella prima parte introduco l'azienda, nella seconda parte descrivo il suo profilo aziendale e nello specifico il settore di mercato in cui essa opera, chi sono i suoi clienti, come è strutturata e qual è il suo modello di qualità. Infine, descrivo il suo rapporto con l'innovazione.

1.1 IKS

IKS (*Information Knowledge and Supply*) è un'azienda padovana fondata dall'attuale Amministratore Delegato Paolo Pittarello nel 1999.

Nell'insieme, IKS unisce figure di alto profilo per proporre soluzioni innovative alle richieste di mercato dell'[Information and Communication Technology \(ICT\)](#) sia italiano che estero. Le soluzioni offerte interessano in particolare gli ambiti della sicurezza, dell'infrastruttura e della governance [Information Technology \(IT\)](#).

L'azienda è in continua ricerca tecnologica. Investendo sulla formazione del proprio personale, IKS si impegna di portare solo valore aggiunto al business dei propri clienti. Inoltre, l'Azienda crede fortemente nell'innovazione come strumento verso un ambiente digitale comune, [Agile \(metodologia\)](#) e completamente disponibile.

Il quartier generale aziendale è a Padova. Inoltre, IKS possiede uffici anche nelle seguenti città: Roma, Milano e Trento.

A partire dallo scorso anno, 2016, IKS SRL, Kirey SRL, Insirio SPA e System Evolution SRL hanno fondato il Gruppo Kirey. L'obiettivo comune delle quattro aziende è unire le competenze complementari e garantire un portfolio completo di soluzioni ai clienti attuali e futuri.

La creazione del Gruppo Kirey è stata guidata dalla Synergo SGR una società di [Private equity](#). E il presidente del nuovo Gruppo commerciale è Vittorio Lusvarghi.

A seguito della creazione del Gruppo, IKS SRL e le restanti tre realtà aziendali hanno conservato la propria struttura di governance e management, garantendo la continuità gestionale.

1.2 Profilo dell'azienda

1.2.1 Servizi e prodotti offerti

Nel corso degli anni IKS si è fatta notare per gli enormi contributi innovativi nell'ambito della sicurezza informatica. Tuttavia, essa non è limitata a questo ambito. Infatti, gli altri ambiti di applicazione sono: infrastruttura e governance IT.

Di seguito vengono presentati quali sono i servizi offerti da IKS per ciascun ambito:

* IT Security

– Risk analysis e vulnerability assessment

E' importante garantire la sicurezza dell'infrastruttura informatica nel suo complesso. A questo scopo IKS offre un servizio orientato alla ricerca di eventuali vulnerabilità e analisi dei rischi a esse collegate;



Figura 1.1: Visione a processo della gestione del rischio. Immagine tratta da: <http://bit.ly/2rh3V0A>.

– Audit management

Le aziende di continuo sono sottoposte a controlli di vario genere per accertare che le operazioni aziendali sono a norma con certificazioni, leggi, bilanci ed ecc. A questo scopo IKS offre un servizio di supporto per le aziende con il fine di agevolare le attività di auditing e eventualmente per migliorare i processi interni aziendali;



Figura 1.2: Flusso di lavoro durante lo svolgimento di un audit. Immagine tratta da: <http://bit.ly/2rdFhfv>.

– **Difesa perimetrale**

Sempre in ambito della sicurezza è importante prendere le giuste misure per garantire a priori uno specifico livello di sicurezza e limitare a zero le intrusioni dall'esterno di un'infrastruttura IT aziendale. A questo scopo IKS offre un'insieme di soluzioni orientate al monitoraggio degli accessi a sistemi aziendali, dei permessi sulle operazioni che un utente può fare e/o vedere, e molto altro;



Figura 1.3: Visione grafica del concetto di difesa perimetrale. Immagine tratta da: <http://bit.ly/2s834O2>.

* **IT Infrastructure**

– **Business continuity**

In ambito bancario, le infrastrutture informatiche sono molto complesse. E la loro manutenzione non è semplice. Ancora più difficile è garantire che questi sistemi siano operativi al 100%. Una simile percentuale nella pratica è impossibile. IKS con il proprio gruppo di esperti sono alla continua ricerca di soluzioni per incrementare la percentuale di continuità operativa di questi sistemi. Infatti, le soluzioni offerte dall'azienda sono orientate nel concreto all'infrastruttura del cliente richiedente supporto;



Figura 1.4: Visione del ciclo di vita del processo business continuity. Immagine tratta da: <http://bit.ly/2qvCmgP>.

– Virtualization technology

Ogni prodotto software per portare valore aggiunto a un'azienda deve essere eseguito. Mappare in esecuzione un prodotto software per server fisico richiede la disponibilità di un cospicuo numero di questi server. A questo scopo la tecnologia di virtualizzazione permette la creazione di server logici che eseguono programmi e a loro volta vanno in esecuzione su server fisici. I benefici di una simile infrastruttura è l'ottimizzazione delle risorse di calcolo, agilità di gestione e sicurezza. Alcune delle soluzioni di virtualizzazione offerte da IKS sono: VMWare, RHEV. Un'evoluzione della tecnologia di virtualizzazione è il cloud. In questo ambito IKS offre soluzioni di migrazione e supporto verso il Cloud dell'infrastruttura IT classica di un'azienda;

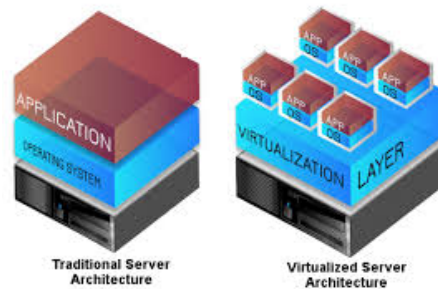


Figura 1.5: Vista a confronto tra un ambiente server bare metal e virtualizzato. Immagine tratta da: <http://bit.ly/2qvtLLk>.

* IT Governance

– Service management

Un servizio informatico richiede costante attenzione. Questo necessita di enormi investimenti economici per la manutenzione. IKS offre diversi piani di gestione per soddisfare anche i più esigenti clienti;

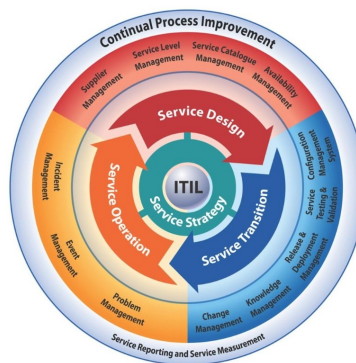


Figura 1.6: Visione della gestione di servizio in prospettiva del Framework ITIL. Immagine tratta da: <http://bit.ly/2qvNryk>.

– **Application and performance monitoring**

Un prodotto software ha il proprio specifico ciclo di vita. Concluso il ciclo di sviluppo, il prodotto è rilasciato in produzione. La seconda parte del ciclo di vita di un prodotto software è la manutenzione. Essere in grado di monitorare un applicativo è importante per avere una costante visione dello stato del prodotto e prevenire eventuali esigenze di manutenzione generica oppure di basso profilo a livello di codice sorgente. In questo dominio, grazie a partnership strategiche, IKS offre soluzioni mirate a garantire la miglior possibile esperienza di monitoraggio applicativo;

– **System and networking management**

Gestire sistemi e reti complessi è un compito complesso. Utilizzare strumenti adeguati permette di semplificare il lavoro e garantire un stato consistente del sistema nel tempo. Le soluzioni offerte di IKS sono orientate alla flessibilità e facilità d'uso dei prodotti offerti in questo contesto;

* **Innovation & Project**

– **Architetture applicative distribuite**

I sistemi informatici diventano sempre più di natura distribuita. IKS offre in questo ambito soluzioni architetturali orientate a microservizi, utilizzando le ultime tecnologie orientate alla containerizzazione e orchestrazione di container;

– **Sviluppo di applicazioni cloud native**

E' sempre più comune sentire parlare di cloud. Le classiche architetture applicative non riescono a beneficiare della flessibilità del cloud perché non sono scalabili e sono difficilmente modularizzabili vista la loro architettura a monolite. Per questo motivo le applicazioni devono essere sviluppate fin dal principio con un'architettura orientata al Cloud. Una buona guida di sviluppo di applicazioni web orientate al Cloud è la *Twelve Factor-App*. In questa direzione IKS si impegna di proporre soluzioni architetturali orientate all'affidabilità, resilienza, scalabilità orizzontale ed ecc.

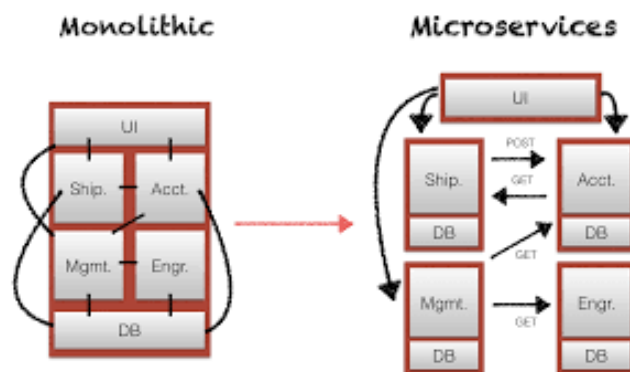


Figura 1.7: Visione architetturale a monolite e microservizi a confronto. Immagine tratta da: <http://bit.ly/2rh1niY>.

La clientela tipica di IKS sono aziende che operano nell'ambito della pubblica amministrazione, bancario, assicurativo e servizi.

Una lista dettagliata delle referenze può essere consultata sul sito di IKS all'URL <https://www.iks.it/referenze.html>.

1.2.2 Struttura organizzativa

Ad oggi IKS conta più di 100 dipendenti. La sua organizzazione interna è riassunta nel diagramma in **Figura 1.8**.

In seguito descrivo le unità operative che costituiscono il nucleo decisionale dell'azienda. Queste unità sono:

- * **Direzione**
Definisce gli orientamenti e le politiche aziendali. Definisce gli obiettivi per la qualità, riesamina periodicamente il sistema di qualità e gestisce il piano di formazione dei dipendenti in funzione alle esigenze e motivazioni personali;
- * **Direzione Commerciale**
Definisce le politiche commerciali, gli obiettivi e le risorse necessarie. Promuove i servizi e prodotti dell'azienda. Gestisce i clienti, i fornitori e le offerte contrattuali;
- * **Direzione tecnica o Operation**
Supporta la Direzione Commerciale nella valutazione commerciale di prodotti e/o offerte dal punti di vista tecnico. Gestisce a livello tecnico i progetti e servizi. Pianifica le risorse necessarie per i prodotti/servizi. Verifica lo stato del prodotto/servizio offerto;
- * **Amministrazione & Finanza**
Gestisce la documentazione di progetto su coordinamento della direzione commerciale e tecnica. Gestisce l'archiviazione della documentazione;
- * **Acquisti**
Su coordinamento della Direzione, gestisce i fornitori di prodotti e servizi. Gestisce il processo di acquisizione di nuovi prodotti o servizi. Il processo di acquisizione è guidato dalle necessità interne aziendali oppure da quelle dei clienti;
- * **Assicurazione Qualità**
E' in stretto contatto solo con la Direzione. Gestisce il piano di qualità, coordina le attività di ispezione, misura e stima il livello della qualità aziendale;
- * **Business Unit (BU)**
Gestisce i progetti o servizi concordati con il Cliente. Rendiconta direttamente alla Direzione Tecnica e gestisce l'emissione delle fatture verso il Cliente.

1.2.3 Processi aziendali

IKS dal 2003 è certificata UNI EN ISO 9001. Questo certifica che l'azienda cura molto la qualità del lavoro interno. Infatti, migliorare di continuo il proprio *modus operandis* permette all'azienda di rimanere competitiva sul mercato e consolidare la propria posizione di leader sul mercato del ICT italiano. Riporto di seguito alcuni obiettivi di qualità dell'azienda:

- * Mantenere e aumentare il livello di soddisfazione del Cliente;

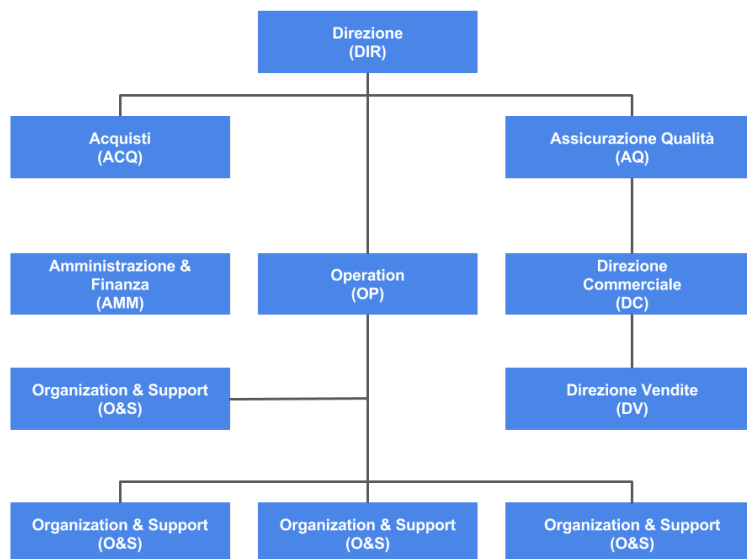


Figura 1.8: Organigramma aziendale

- * Operare in modo efficiente ed efficace per soddisfare i requisiti contrattuali, norme e regolamenti;
- * Monitorare i propri processi per garantire azioni correttive tempestivamente e permettere un comportamento pro attivo, per anticipare i bisogni e predire le risorse aziendali necessarie prima del loro effettivo bisogno;
- * Assicurare una adeguata formazione al Personale.

Il ruolo del Cliente è di prima importanza. Infatti, l'azienda cerca di coinvolgerlo il più possibile per comprendere meglio i suoi bisogni. Dopo aver formalizzato il bisogno del Cliente segue un'attività di analisi dei requisiti. L'obiettivo dell'attività è la dettagliata comprensione del contesto applicativo, quali sono le parti interagenti e quali possono essere i rischi durante l'attività di progetto per implementare i bisogni del Cliente.

A progetto concluso, il Cliente valuta criticamente la soluzione presentata. La valutazione può coinvolgere anche un reclamo, il quale è rivolto alla Direzione dell'azienda.

IKS organizza il proprio lavoro per processi: primari, direttivi e di supporto. Ciascuna categoria di processo definisce delle responsabilità e compiti. Per esempio i processi organizzativi interessano le attività per: definire la politica e strategia aziendale, pianificare e allocare le risorse, riesaminare la gestione del sistema di qualità. Invece, i processi primari ricoprono attività che garantiscono un diretto ricavo economico per l'azienda e danno un valore aggiunto al prodotto o servizio fornito. Esempi di attività sotto questa categoria sono: proporre offerte commerciali ai clienti, progettare e sviluppare prodotti software, erogare servizi IT. L'ultima categoria di processi sono quelli di supporto. Tra le consuete attività giornaliere rientrano le seguenti, per esempio: gestire le risorse umane, l'infrastruttura e gli ambienti di lavoro, monitorare e analizzare la qualità aziendale.

Una rappresentazione grafica di come sono legate le parte durante un progetto segue in **Figura 1.9**.

Periodicamente il responsabile della qualità su mandato della Direzione attua attività d'ispezione per controllare la qualità. A posteriori segue un'attività di analisi e misura i livelli di qualità per ogni reparto interno, servizio e prodotto di IKS.



Figura 1.9: Rappresentazione grafica del coinvolgimento del Cliente e i corrispettivi livelli degli interventi del gruppo commerciale, tecnico, direzionale e di supporto nella gestione di un'offerta di progetto.

1.3 Rapporto con l'innovazione

L'innovazione è il processo di gestione dell'intero ciclo di vita di un'idea. L'obiettivo è portare un miglioramento di processo aziendale, di prodotto e/o di servizio. Il seguente miglioramento si traduce in un valore aggiunto per l'azienda in termini di rientro economico e per il Cliente nel soddisfare un bisogno in modo più efficace ed efficiente.

L'approccio innovativo induce l'utilizzo dell'informazione, della creatività e dello spirito d'iniziativa per raccogliere maggior valore aggiunto dalle risorse a disposizione. L'azienda utilizza l'innovazione per soddisfare in modo pro attivo le richieste del Cliente. Questo principio è pienamente in linea con la strategia di qualità aziendale: *client first*.

La modalità di innovazione di IKS è un approccio incrementale. Inizialmente l'azienda cerca di soddisfare i bisogni principali e raggiungere il prima possibile gli obiettivi fissati. In seguito, l'azienda si impegna di migliorare la propria offerta mediante miglioramenti continui a ogni livello di dettaglio.

Per supportare l'innovazione IKS ha creato una cultura aziendale che permette ai propri dipendenti di scambiarsi idee, sperimentare, imparare in gruppo e mettere in atto la propria creatività. Non manca la comunicazione con i propri responsabili. Questi sono i primi a motivare di continuo le risorse umane a loro disposizione. Il dialogo dipendente-responsabile non è verticale. La cultura aziendale in questa direzione è molto drastica: favorire uno scambio di idee equo, semplice e rompere le gerarchie.

In questo contesto, per l'intera durata del mio periodo di stage e dopo un primo momento di ambientamento ho beneficiato molto del clima aziendale. Infatti, non è mancato il libero confronto con il tutor aziendale che si è mostrato molto disponibile e aperto al mio spirito d'iniziativa. Lui mi ha supportato in ogni scelta che io ho motivato e ritenuto significativo per il beneficio del mio progetto.

Le idee sono una parte del processo di gestione dell'innovazione: la realtà è molto più complessa. IKS non possiede un effettivo processo di gestione a livello aziendale. Questo viene gestito da un gruppo di persone con competenze trasversali e a livelli organizzativi differenti.

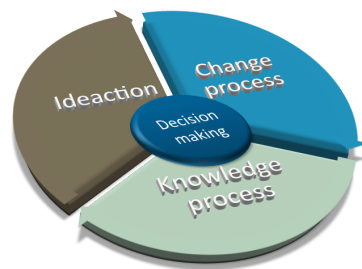


Figura 1.10: Il legame attivo tra innovazione e il processo del cambiamento e gestione della conoscenza. Immagine tratta da: <http://bit.ly/2qty3XD>.

Capitolo 2

L'azienda e gli stage

Il presente capitolo è dedicato al rapporto dell'azienda con gli stage in collaborazione con l'Università. Nello specifico presento: l'ambito, le motivazioni, obiettivi aziendali e personali, e i vincoli del mio progetto di stage.

Lo spirito dell'azienda è investire costantemente sui propri dipendenti e soddisfare i bisogni tecnologici del mercato con le giuste competenze. I dipendenti tecnici dedicano parte della propria giornata di lavoro in: approfondimenti tecnologici e attività di laboratorio.

A supporto dell'attività di laboratorio è stato installato un ambiente virtuale. Avere a disposizione un simile environment permette di sperimentare con: nuove tecnologie, integrazione di sistemi ed ecc. Inoltre, permette di mettere a disposizione degli stagisti un'infrastruttura dedicata ai loro progetti di stage. Oltre alla virtualizzazione è possibile sperimentare nei seguenti ambiti: storing, networking e security. Temi comuni di approfondimento specializzante sono: Cloud, Machine Learning e Analytics.

Di recente, l'Azienda ha concluso una partnership strategica con AWS, il provider di cloud pubblico. L'accordo di collaborazione permetterà a IKS di portare molte delle proprie soluzioni sul Cloud e beneficiare delle sue peculiarità: elasticità, flessibilità nella gestione di infrastrutture ed ecc.

Per monitorare e prevenire le frodi bancarie IKS ha sviluppato alcune soluzioni. Queste usano tecniche sofisticate di Machine Learning e analisi dei log. Inoltre, è in corso un progetto per il loro rilascio come servizio. In questo modo gli utilizzatori avranno la possibilità di integrare la soluzione nei propri sistemi e monitorare il flusso delle transazioni bancarie in tempo reale.

Ogni anno l'azienda partecipa a StageIT: un evento completamente dedicato agli studenti universitari dei Corsi di Laurea in Scienze e Ingegneria Informatica. Infatti, quest'evento è un'opportunità per lo studente di mettersi in contatto con le realtà aziendali e per queste ultime di conoscere i talent più da vicino. Durante l'evento è previsto anche un concorso che promuove il miglior progetto di stage. In generale su un'insieme di progetti effettuati nell'edizione precedente dell'evento vengono scelti un sotto insieme ristretto di finalisti da una commissione orientata alla promozione dell'innovazione.

Il vincitore, invece, è scelto dagli studenti in tempo reale durante l'evento. Il premio del vincitore è un buono d'acquisto del valore di 500 Euro.

2.1 Il valore aggiunto di uno stagista

IKS è un partecipante attivo a StageIT e annualmente propone fino a 6 progetti di stage. Questi non sono verticali su un'unica tematica ma usualmente coinvolgono temi come:

- * Sviluppo di applicazioni basate su web, [Cloud](#), mobile o migrazione su [Cloud](#)/mobile di applicazioni tradizionali;
- * Progettazione di ambienti, metodologie e strumenti di sviluppo software.

Lo stagista è una risorsa importante per l'azienda. Esso viene visto come un portatore di novità. In principio, lo stagista è impiegato su progetti di sperimentazione. I quali hanno come obiettivo: lo studio e l'analisi di fattibilità dell'integrazione delle soluzioni nell'offerta commerciale dell'azienda.

Per l'intera durata dello stage, lo stagista si emerge in un ambiente di lavoro il più possibile vero simile alla realtà aziendale. Questo permette al tutor esterno di analizzare più da vicino il candidato in stage. E alla fine dello stage allo stagista può essere proposta un'offerta d'assunzione.

L'Azienda, grazie al contributo degli stagisti, si allinea con i temi di ricerca universitari e con le tendenze tecnologiche del momento sul mercato internazionale.

2.2 Alcuni temi di stage

2.2.1 AIOps e Machine Learning

Il progetto di stage tratta l'integrazione del Machine Learning con strumenti di Application Performance Monitoring. L'obiettivo dello stage è sperimentare integrando diverse soluzioni in questo ambito e studiarne il prodotto finale. Una conseguenza critica di questo progetto è lo sviluppo di un pensiero critico per affrontare le più difficili sfide del monitoraggio di applicazioni e infrastrutture.

Il presente progetto si colloca in ambito del *application and performance monitoring* che è un servizio offerto dall'azienda al supporto della governance IT.

2.2.2 DevOps Automazione

L'automazione è fondamento di ogni realtà aziendale contemporanea. Infatti, il numero di macchine da gestire spesso non è piccolo. Per semplificare i compiti di gestione si devono utilizzare strumenti di configurazione e automazione. Queste tecnologie permettono di automatizzare tutte le operazioni manuali che un sistemista spesso compie durante le attività di manutenzione giornaliera. L'obiettivo di questo progetto è l'integrazione di alcuni strumenti che semplificano il [Patching](#) dei server e sperimentare con nuove tecnologie del settore. Il presente progetto si colloca nell'ambito del *system management*.

2.2.3 Sviluppo moduli evolutivi in ambito antifrode

IKS ha grande esperienza in ambito della sicurezza informatica bancaria. Come prodotto risultante di questa esperienza è SMASH. L'obiettivo dello stage è estendere il prodotto con qualche funzionalità di monitoraggio di azioni sospette. Oltre allo sviluppo di moduli evolutivi lo stagista ha la possibilità di apprendere delle competenze forti

nell'ambito della sicurezza informatica. La presente proposta di stage è un progetto inter business unit dell'azienda. Esso si colloca nell'ambito dello sviluppo di prodotti software e della sicurezza informatica nel settore bancario.

2.3 Il progetto proposto

2.3.1 Motivazioni

E' sempre più comune nelle realtà aziendali l'impiego dell'approccio agile nelle attività di lavoro giornaliero. Infatti, questo approccio promuove la comunicazione in generale e focalizza l'attenzione di tutti i stakeholder sul valore finale, per esempio di un prodotto software oppure di una strategia di mercato, che deve essere garantito.

Se gli sviluppatori hanno come obiettivo primario lo sviluppo di un prodotto software, i professionisti dell'IT hanno come priorità la sua garanzia di servizio e manutenzione periodica.

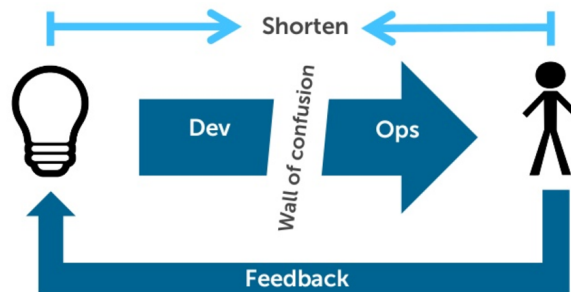


Figura 2.1: Sia gli sviluppatori che i professionisti IT sono portatori di valore: un feedback che coinvolge ambe le parti è essenziale. Immagine tratta da: <http://bit.ly/2rM9lBQ>.

Tra i due gruppi esiste un muro di incomprensione. Questo fenomeno è dovuto alla mancanza di comunicazione ed interazione. In caso di eventuali problemi che possono incorrere dopo il rilascio del prodotto software è sola responsabilità dei professionisti IT rimuoverli e riportare il prodotto software in uno stato di operatività non a rischio.

Un simile scenario per un'azienda informatica che deve affrontare un numero elevato di rilasci giornalieri non è accettabile. A questo scopo si è creato un movimento culturale, chiamato DevOps, orientato all'unione degli sviluppatori e sistemisti. L'unione promuove un cambio di mentalità, creazione di nuove competenze e sviluppo di nuovi strumenti che diminuiscano la distanza tra le due realtà.

Il DevOps ha conseguenze più profonde del semplice cambio culturale. Un'azienda che approccia il DevOps affronta un cambiamento interno che ripensa il proprio modello di qualità. I benefici del cambio sono i seguenti: maggiore innovazione, agilità nel cogliere i bisogni di mercato del momento, flessibilità nel gestire il cambiamento e maggior qualità di prodotto e processo.

Una tipica rappresentazione del ciclo di vita DevOps è come segue in figura.

L'abilità di poter cambiare quando necessario in modo agile è un beneficio notevole per le aziende; in questo modo i dipendenti sono impiegati nel creare solo valore aggiunto per l'azienda e il mercato. Invece, la gestione dell'infrastruttura è disciplinata, sistematica e standardizzata. Con un approccio standardizzato e ben strumentalizzato

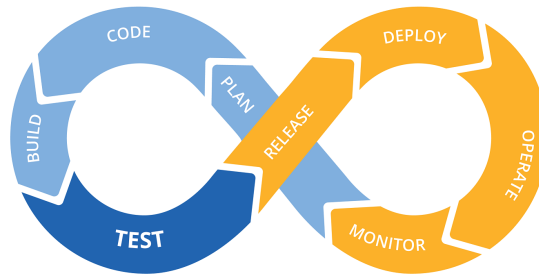


Figura 2.2: Il DevOps abilita l'automazione del processo di rilascio del software e i cambi dell'infrastruttura IT. Immagine tratta da: <http://bit.ly/2rsw9nm>.

è sempre più difficile individuare server nomadi. Può succedere che in caso di operazioni sofisticate di manutenzione un server scompaia dall'orizzonte di visibilità. Se siamo in ambiente Cloud questo risulta in costi in eccesso. Se siamo in ambiente virtualizzato questo risulta in riduzione della capacità complessiva di calcolo.

Sebbene il DevOps possiede uno scopo più ampio, il *continuous delivery* è un approccio che promuove l'automazione di tutti i processi che vengono coinvolti durante il rilascio di un prodotto software. Il *continuous delivery* permette di abbreviare i tempi di rilascio e aumentarne il loro numero, e migliorare la gestione del cambiamento.

Essere veloci nel *delivery* di un prodotto software non è sufficiente. E' necessario prevedere una *pipeline* di *deploy* per il software che si vuole portare nell'ambiente di produzione. L'ambiente direttamente esposto all'uso dei clienti. Automatizzare questo passaggio implica minor intervento manuale e minor numero di errori e conseguentemente maggior rigore nelle attività complessive coinvolte.

L'utilizzo di *application container* permette di confezionare le applicazioni in un'unità singola che contiene anche le sue dipendenze. Confezionare in questo modo le applicazioni rende facile lo scambio di artefatti tra i diversi gruppi. Di conseguenza lo stesso artefatto creato da uno sviluppatore verrà consegnato al verificatore che attuerà il *testing* di quel particolare prodotto con una specifica versione e sistemista che ne prevederà il suo successivo rilascio.

Un esempio di strumento di containerizzazione è LXC (Linux Kernel Container), Docker ed ecc. Il primo è un progetto che permette il supporto dei container a livello del kernel Linux. Docker è un'altra soluzione di containerizzazione. Le prime versioni Docker offrivano un insieme di API per interagire con LXC, ora Docker offre una propria soluzione di containerizzazione completamente indipendente da quella Linux.

Con la containerizzazione segue un'estrema facilità nel gestire le applicazioni durante il loro intero ciclo di vita.

Il cambio di filosofia è percepibile anche al livello architetturale dei prodotti software. In un ambiente dinamico caratterizzato dall'automazione, verifiche e deploy automatici le classiche architetture software non riescono a beneficiare di questa flessibilità. A questo scopo i microservizi rappresentano uno stile architetturale in sintonia con la filosofia delle *pipeline* Unix: ogni microservizio implementa una sola funzionalità - basso accoppiamento.

In figura è possibile notare diversi microservizi. Ciascuno ha una responsabilità ben definita. Questi possono comunicare tra di loro. Per garantire un alto disaccoppiamento tra i servizi è necessario introdurre un microservizio di servizio, chiamato *Service Disco-*

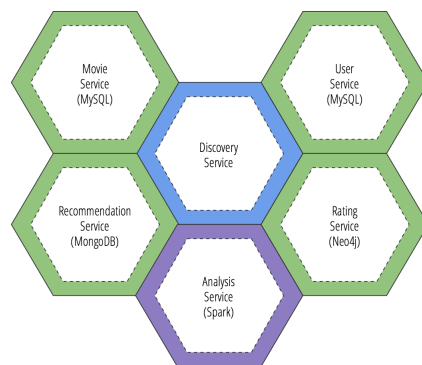


Figura 2.3: Esempio di un sistema a microservizi. Immagine tratta da: <http://bit.ly/2qNXXkj>.

very, utilizzato come un DNS (Domain Name System). In questo modo i microservizi possono coesistere nello stesso ambiente e comunicare solo quando necessario senza conoscersi direttamente. Inoltre, i microservizi ottengono un'indipendenza dal luogo di esecuzione. Se un microservizio X in esecuzione su una macchina A migra per eseguire su una macchina B allora un microservizio Y che vuole comunicare con X deve contattare il *Service Discovery* per ottenere l'indirizzo di X. L'effetto che si ottiene è un alto tasso di mobilità dei servizi.

E' usuale incapsulare un microservizio in un container software. In questo modo ogni microservizio diventa un'unità funzionale indipendente e nello stesso ambiente possono coesistere due o più copie dello stesso servizio eliminando l'interferenza dell'uno sull'altro. Per aumentare la capacità di robustezza e affidabilità di un microservizio sarà sufficiente scalare in orizzontale creando una copia aggiuntiva del microservizio che soffre delle proprietà precedentemente citate. Il traffico in ingresso sarà bilanciato con qualche politica di distribuzione del carico ai due microservizi attivi tramite un *Load Balancer*.

I microservizi semplificano di molto l'applicazione complessiva scomponendo il prodotto in sotto applicazioni indipendenti; complicano l'applicazione nel complesso perché vengono aggiunte componenti nuove e il traffico inter microservizio diventa molto più difficile da gestire.

Si aprono nuove sfide sia per gli sviluppatori che per i sistemisti. E queste sfide caratterizzano il contratto di collaborazione tra i due gruppi.

2.3.2 Obiettivi aziendali

Un team interno di IKS ha sviluppato una soluzione di Executive e Malware Dashboard basata sullo stack applicativo: Elasticsearch, Logstash e Kibana.

L'obiettivo principale del mio progetto di stage è la containerizzazione della soluzione precedentemente implementata e garantire l'alta affidabilità delle dashboard.

Inoltre, tramite l'utilizzo della tecnologia dei container applicativi l'azienda ha l'obiettivo di individuare una soluzione architettuale dell'applicativo affinché sia portabile su ambienti come Cloud, macchine virtuali e/o fisiche, e sul portatile dello sviluppatore.

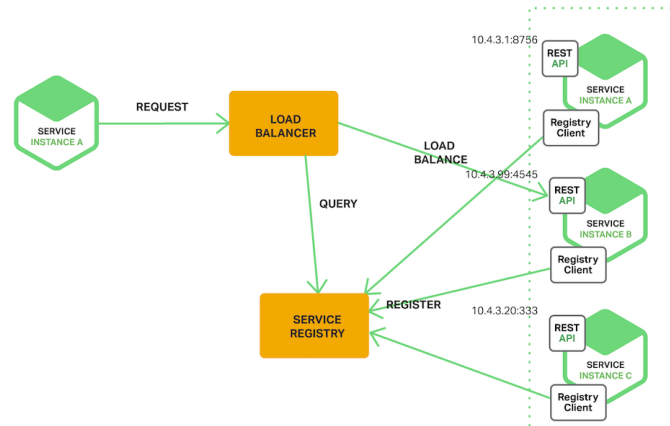


Figura 2.4: I microservizi permettono di scalare orizzontalmente per reggere ai più esigenti carichi di lavoro. Immagine tratta da: <http://bit.ly/2qI50LR>.

Oltre alla portabilità dell'applicativo è necessario garantire anche la portabilità dell'infrastruttura che ospita l'applicativo per l'esecuzione.

Con la garanzia di alta portabilità il team di gestione dell'infrastruttura potrà gestire lo stesso ambiente in contesti differenti e affini a scopi diversi in modo univoco.

2.3.3 Obiettivi personali

Come attività preliminare alla ricerca di un progetto di stage per la Laurea ho attuato uno studio individuale di mercato. Lo scopo era capire: tendenze tecnologiche, architetture e metodologiche. Se da un lato le mie ricerche hanno cercato di cogliere le novità del momento, dall'altro a livello personale queste erano mirate alla ricerca di un contesto in cui potermi applicare e maturare.

Con il presente progetto gli obiettivi personali erano:

- * Apprendere conoscenze e competenze in ambito di:
 - Virtualizzazione basata sulla tecnologia a container;
 - Sistemi distribuiti;
 - Amministrazione di sistema Linux;
- * Acquisire esperienza pratica nella gestione delle reti di calcolatori in ambito dei sistemi, nello specifico le reti definite in modo programmatico per le tecnologie orientate alla containerizzazione;
- * Acquisire esperienza nell'analisi, progettazione e implementazione di sistemi orientati ai microservizi;
- * Famigliarizzare con la piattaforma Kubernetes e i principi del [Cloud](#).

2.4 Piano di lavoro

Il piano di lavoro è stato pianificato per un totale di 300 ore complessive. Il contenuto del piano è stato presentato in un documento di cui una copia è stata consegnata all'Ufficio degli Stage presso l'Ateneo dell'Università di Padova, una seconda copia è stata consegnata al tutor interno e l'ultima copia controfirmata dall'ufficio stage dell'Università è stata consegnata all'azienda. Il piano è stato strutturato in tre fasi il cui contenuto presento di seguito:

- * Fase 1 - Formazione (56 ore)

- Docker: la tecnologia per la containerizzazione;
- Kubernetes: la tecnologia per l'orchestrazione;
- ELK: lo stack applicativo;
- Verifiche delle competenze acquisite;

- * Fase 2 - Analisi e progettazione (56 ore)

- Analisi delle funzionalità della soluzione non containerizzata di dashboard;
- Analisi delle modalità di containerizzazione delle componenti;
- Analisi delle modalità di [Deployment](#);
- Progettazione delle modalità di verifica della non regressione;
- Progettazione architetturale della soluzione;
- Progettazione della modalità di [Deployment](#);
- Documentazione;

- * Fase 3 - Implementazione (188 ore)

- Installazione e configurazione dell'orchestratore;
- Implementazione della soluzione in un contesto con e senza orchestratore;
- Verifica di non regressione;
- Documentazione.

2.5 Vincoli

2.5.1 Vincoli temporali

Lo stage è durato 8 settimane per un complessivo di 310 ore di lavoro. Ho lavorato a tempo pieno con il seguente orario: 9.00-18.00. Con la pausa pranzo di 1 ora dalle 12.30 alle 13.30. Come stabilito nel PdL (Piano di Lavoro) le attività sono state strutturate in tre fasi. Ogni fase ha coinvolto attività mirate al raggiungimento di specifici obiettivi. Per maggior dettaglio sul contenuto del PdL riferire la [sezione Piano di Lavoro](#).

2.5.2 Vincoli tecnologici

Fin dal primo giorno di lavoro l'azienda mi ha fornito un portatile dedicato per l'intero periodo di stage. Inoltre, mi è stato vietato di collegare alla rete aziendale qualsiasi dispositivo personale. Inoltre, il portatile di lavoro non poteva essere portato a casa. Per comunicare internamente sono stati utilizzati strumenti di messaggistica istantanea, come Skype, per comunicazioni informali e la posta elettronica.

Oltre a questo vincolo, a livello tecnologico sono state fissate le seguenti tecnologie:

- * CentOS7: il sistema operativo installato sulle macchine di laboratorio. CentOS7 è la versione open source di RHEL7 (Red Hat Enterprise Linux versione 7);
- * Docker: è uno strumento che permette in modo estremamente facile la creazione, il deploy e l'esecuzione di applicazioni utilizzando la tecnologia a container. In questo modo l'attenzione dell'utente è focalizzata su questioni diverse dall'installazione e configurazione dell'applicazione. L'architettura di Docker segue in figura.

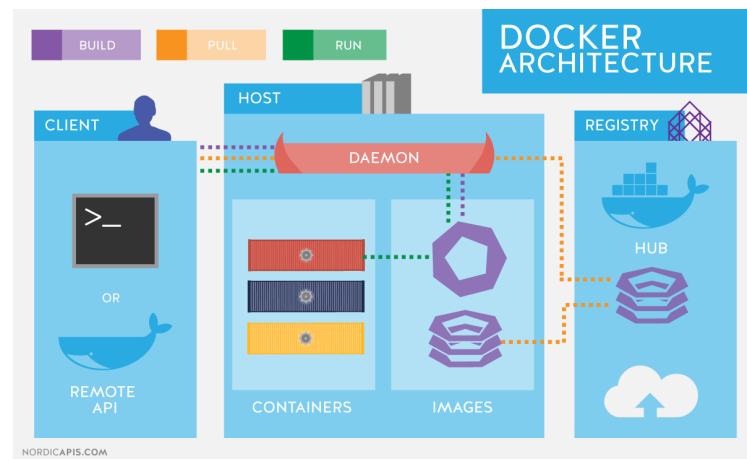


Figura 2.5: Le parti costituenti la piattaforma Docker sono: il demone, il client e il registry Docker. Immagine tratta da: <http://bit.ly/2rmkt7g>.

Le componenti architetturali costituenti la piattaforma Docker sono il: demone, client e registry. Si può notare che l'architettura di alto livello è un'architettura client/server. Il server di Docker è il demone che ha la responsabilità di gestione dei container sulla macchina locale. Mentre il client si interfaccia tramite un'interfaccia REST al demone e permette di interagire in modo agile con i container, creare reti virtuali, gestire i dati che devono essere condivisi tra i container e il file system locale della macchina. Infine, il registry di Docker è una repository che può essere pubblica o privata e ha la responsabilità di abilitare la condivisione di immagini utili alla creazione dei container. Come modello mentale, in relazione con il paradigma ad oggetti, è possibile paragonare le immagini Docker a classi che devono essere istanziate per la creazione di oggetti, ovvero container.

Per favorire il libero scambio di immagini Docker, l'azienda Docker Inc ha messo a disposizione degli utenti un hub completamente gratuito. Nella versione privata di una repository è disponibile il servizio di *scanning* delle immagini per individuare le vulnerabilità di sicurezza.

Quando si esegue il comando `run` tramite la CLI di Docker il demone controlla che l'immagine da usare per la creazione del container sia presente in locale. In caso affermativo il container viene creato e messo in esecuzione, altrimenti il demone scarica l'immagine dal registry e al termine del *download* istanzia il container;

- * Kubernetes: è un sistema open source per automatizzare il deploy, la scalabilità e gestione di applicazioni containerizzate. La tecnologia è un prodotto risultante da 15 anni di esperienza in Google con i container. Essa garantisce la portabilità delle applicazioni e l'indipendenza dall'ambiente fisico di esecuzione.

Kubernetes è un sistema distribuito e il modello architetturale è master/slave.

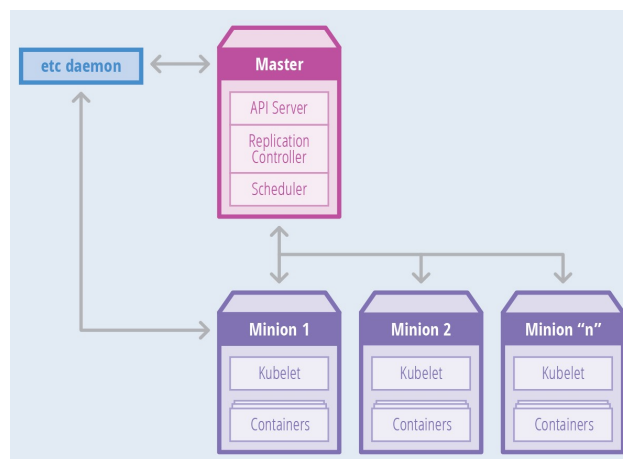


Figura 2.6: Le componenti architetturali di Kubernetes sono: API server, Scheduler, Replication Controller, Kubelet, Kube proxy, Database Etcd. Immagine tratta da: <http://bit.ly/2s4eKUR>.

Il master è un pannello di controllo del sistema K8s (Kubernetes). La sua responsabilità è gestire il *workload* a container. Inoltre, esegue le componenti critiche del sistema: il database chiave valore ad alta consistenza etcd, il gestore delle repliche per la scalabilità orizzontale - replication controller e lo scheduler.

La componente slave esegue il carico di lavoro. Essa comunica solo con il master e salva le informazioni di servizio tramite il API server nel database etcd.

Ogni componente master e slave eseguono un agente locale chiamato Kubelet. Il compito dell'agente è quello di collegare le varie componenti e comunicare con il demone Docker.

Infine, il Kube proxy è la componente che gestisce il traffico di rete dell'intera infrastruttura.

Kubernetes, essendo un sistema fin dall'inizio pensato per essere componibile si può integrare bene con soluzioni di terzi parti, come per esempio: diverse soluzioni per lo storage, diversi plugin per la rete ed ecc;

- * Elasticsearch, Logstash e Kibana: Le tre componenti sono comunemente conosciute con l'acronimo ELK. Esse vengono utilizzate insieme come una soluzione open source in progetti che hanno forti esigenze di ricerca e analisi di dati.

Elasticsearch è il cuore dello stack applicativo. Esso è un database NoSQL e distribuito implementato in Java. Orientato all'immagazzinamento di dati non strutturati permette di effettuare ricerche complesse impiegando millisecondi contro i secondi necessari utilizzando un classico DB SQL. Kibana, invece, è la componente dello stack che offre la funzionalità di visualizzazione dei dati presenti in Elasticsearch. Una peculiare caratteristica di Kibana è l'interfaccia di creazione di cruscotti. Essendo la soluzione nativa di visualizzazione per Elasticsearch, Kibana permette di sfruttare questa integrazione per esprimere ricerche molto complesse e visualizzarle a video tramite effetti grafici accattivanti. Infine, Logstash è la componente di estrazione, trasformazione e caricamento dei dati dalla sorgente in Elasticsearch. Con Logstash risulta semplice filtrare l'informazione utile per l'analisi dei dati ed eliminare il rumore di fondo. Essendo uno strumento implementato in Java offre un insieme ricco di strumenti di terzi parti che arricchiscano ulteriormente il suo insieme di funzionalità. Per esempio, tramite uno plugin esterno è possibile programmare Logstash a interagire con le API del social network Twitter per cercare l'informazione che soddisfa dei particolari criteri di ricerca. Dal punto di vista architetturale la soluzione ELK è flessibile e permette di scalare orizzontalmente in proporzione al carico di lavoro.

Inizialmente sono state fissate anche le rispettive versioni delle componenti sopra citate. Tuttavia, nel corso dello stage ho realizzato che bloccare l'evoluzione di un'infrastruttura può comportare qualche problema nel futuro. A questo scopo ho predisposto un ambiente tollerante agli aggiornamenti e che si auto aggiorna. Durante la personalizzazione dell'ambiente mi sono ispirato al principio *self driven infrastructure* di CoreOS. In questo modo gli aggiornamenti delle componenti possono essere effettuati in modo completamente trasparente.

Le immagini per la creazione di container dovevano essere provenienti solo dal repository ufficiale di Docker ed essere le immagini ufficiali.

Capitolo 3

Lo svolgimento dello stage

TODO: Aggiungere sintesi al capitolo

3.1 Metodo di lavoro

3.2 Attività di formazione

3.3 Analisi dei requisiti

3.3.1 Classificazione dei requisiti

3.3.2 Requisiti

3.3.2.1 Funzionali

3.3.2.2 Non funzionali

3.4 Progettazione e realizzazione

3.4.1 Scelte progettuali

3.4.2 Visione architetturale a microservizi

3.4.3 Codifica

3.4.4 Test

3.4.4.1 Test di carico

3.4.4.2 Test di durata

Capitolo 4

Valutazioni retrospettive

TODO: Aggiungere sintesi al capitolo

4.1 Obiettivi raggiunti

4.2 Problematiche riscontrate

4.3 Bilancio formativo

4.3.1 Il prima

4.3.2 Il dopo

4.4 Valutazione critica del Corso di Laurea

Glossario

Agile Metodo per lo sviluppo del software che coinvolge quanto più possibile il committente, ottenendo in tal modo una elevata reattività alle sue richieste. . [1](#), [25](#)

Cloud Paradigma di erogazione di risorse informatiche, come l'archiviazione, l'elaborazione o la trasmissione di dati, caratterizzato dalla disponibilità on demand attraverso Internet a partire da un insieme di risorse preesistenti e configurabili. . [12](#), [16](#), [25](#)

Deployment Consegna o rilascio al cliente, con relativa installazione e messa in funzione o esercizio, di una applicazione o di un sistema software tipicamente all'interno di un sistema informatico aziendale. . [17](#), [25](#)

Framework Architettura logica di supporto su cui un software può essere progettato e realizzato, spesso facilitandone lo sviluppo da parte del programmatore. La sua funzione è quella di creare una infrastruttura generale, lasciando al programmatore il contenuto vero e proprio dell'applicazione. . [vii](#), [4](#), [25](#)

ICT insieme di metodi e tecnologie che implementano i sistemi di trasmissione, ricezione e elaborazione di informazioni.. [27](#)

IT utilizzo di qualsiasi tecnologia di calcolo per offrire servizio di memorizzazione, reti per creare, processare, memorizzare e mettere in sicurezza ogni forma immaginabile di dato elettronico. . [27](#)

Patching Applicare una patch, porzione di software progettata per aggiornare o migliorare un programma. Una patch permette di risolvere vulnerabilità di sicurezza e altri BugFix di un applicativo sviluppato. . [12](#), [25](#)

Private equity Da definire . [1](#), [25](#)

Acronimi

ICT [Information and Communication Technology](#). 1

IT [Information Technology](#). 1

Riferimenti

Bibliografia

- Baier, Jonathan. *Getting Started With Kubernetes*. PACKT, 2015.
- Clinton Gormley, Zachary Tong. *Elasticsearch: The Definitive Guide*. O'Reilly Media, 2015.
- Rafal Kuć, Marek Rogoziński. *Elasticsearch Server*. PACKT, 2016.
- Turnbull, James. *The Docker Book: Containerization is the new virtualization*. James Turnbull, 2016.
- Vohra, Deepak. *Kubernetes Microservices With Docker*. Apress, 2016.

Sitografia

- Documentazione Docker*. URL: <https://docs.docker.com/>.
- Documentazione Elasticsearch*. URL: <https://www.elastic.co/guide/en/elasticsearch/reference/2.4/index.html>.
- Documentazione Kibana*. URL: <https://www.elastic.co/guide/en/kibana/4.6/index.html>.
- Documentazione Kubernetes*. URL: <https://kubernetes.io/docs/home/>.
- Documentazione Linux*. URL: <http://www.tldp.org/>.
- Documentazione Logstash*. URL: <https://www.elastic.co/guide/en/logstash/2.3/index.html>.
- Documentazione Nginx*. URL: <https://nginx.org/en/docs/>.
- Martin Fowler: Micorservices*. URL: <https://martinfowler.com/articles/microservices.html>.
- Pattern architetturale a microservizi*. URL: <http://microservices.io/index.html>.
- Wikipedia: Cloud computing*. URL: https://en.wikipedia.org/wiki/Cloud_computing.
- Wikipedia: Microservices*. URL: <https://en.wikipedia.org/wiki/Microservices>.