



Corso di Laurea Magistrale in Informatica

Anno Accademico 2018/2019

Software Architecture

Kubernetes

Matteo Marchesini

0000856336

matteo.marchesini12@studio.unibo.it

Indice

1	Descrizione del sistema	2
2	Contesto	4
2.1	Scopo del sistema	4
2.2	Entità esterne	5
2.2.1	Stakeholders	6
2.2.2	Development	6
2.2.3	Piattaforme	6
2.2.4	Concorrenti	6
3	Drivers architetturali	7
4	Struttura	8
5	Funzioni	9
6	Comportamento	10
7	Razionale	11
8	Aspetti analitici	12
9	Stili architetturali simili o derivati	13
10	Referenze	14

Capitolo 1

Descrizione del sistema



kubernetes

Kubernetes è un sistema open source per la gestione di applicazioni containerizzate tra più host; fornisce un meccanismo per il deployment, la manutenzione e lo scaling di applicazioni. È stato inizialmente sviluppato dal team di Google, per poi passare nel 2015 sotto il controllo del *Cloud Native Computing Foundation (CNCF)*, che attualmente lo supporta.

Al giorno d'oggi è uno dei sistemi di orchestrazione per applicazioni containerizzate più utilizzato in assoluto, con una vastità di utenti, partners e una comunità di development attiva. Non a caso tre dei quattro maggior providers di servizi Cloud - Microsoft, IBM e Google - offrono piattaforme di Container as a Service (CaaS) basate su Kubernetes. I servizi che Kubernetes mette a disposizione sono molteplici: fornisce un ambiente la gestione di container, microservizi e piattaforme cloud. Inoltre organizza l'infrastruttura di rete e di archiviazione per conto dell'utente.

In generale un sistema distribuito necessita di più componenti per il corretto funzionamento, alcune open source e altre commerciali; invece Kubernetes da solo fornisce uno scenario in cui le componenti lavorano insieme, andando così a for-

mare un unico componente combinando la semplicità del Platform as a Service (PaaS) con la flessibilità dell'Infrastructure as a Service (IaaS).

L'orchestrazione di container ha avuto un profondo impatto in ogni aspetto del software development e deployment moderno; in particolare ha influenzato l'architettura del Platform as a Service, fornendo un aperto ed efficiente modello per il packaging, deployment, isolamento, scaling e rolling upgrade. Kubernetes svolgerà un ruolo cruciale nell'utilizzo di container da parte di imprese e start-up emergenti.

L'oggetto di questo report è di studiare tutto ciò che riguarda e circonda l'architettura di Kubernetes.

Capitolo 2

Contesto

2.1 Scopo del sistema

In questo capitolo verrà discusso lo scopo di Kubernetes e la sua interazione con le entità esterne, nonché i principali casi d'uso.

Kubernetes, come introdotto nel Capitolo 1, è un sistema di orchestrazione di container e per questo motivo si occupa principalmente di **deployment**, **scaling** e **management** di applicazioni containerizzate. Di seguito viene definito ognuno di questi tasks:

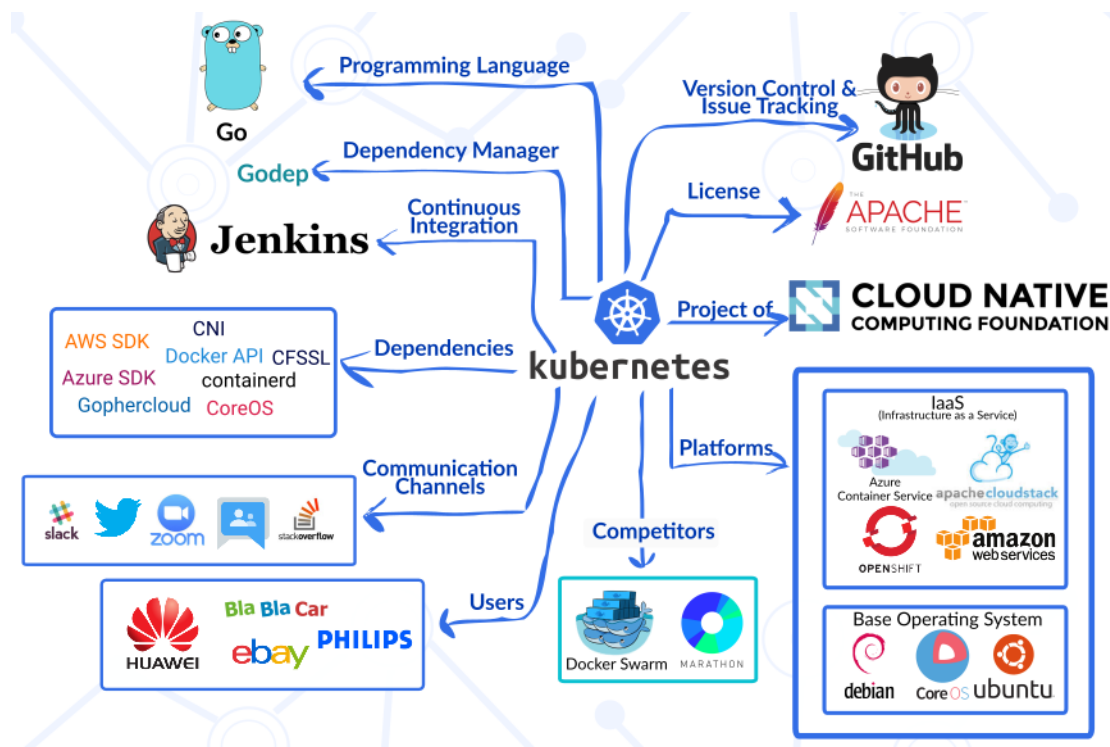
- **Deployment:** gestisce la distribuzione di applicazioni assegnando ai nodi del cluster ciascuna istanza dell'applicazione. Il deployment in Kubernetes può essere eseguito in una varietà di ambienti con pattern differenti, ed esistono appunto diversi modelli, quali:
 - Container as a Service (CaaS);
 - Public Cloud - Infrastructure as a Service (IaaS);
 - Utilizzo on-premises all'interno di data center;
 - Deployment ibrido
- **Scaling:** permette di ridimensionare l'applicazione a seconda delle esigenze dell'utente, andando a modificare le dimensioni del cluster e il numero

di repliche dei pod. Un **pod** è il più piccolo oggetto deployabile nel modello a oggetti di Kubernetes e può incapsulare un singolo container o più container che necessitano di lavorare insieme.

- **Management:** fornisce un'interfaccia per la gestione dei cluster e delle applicazioni containerizzate. Un cluster di Kubernetes viene ospitato e gestito da un venditore commerciale, quali ad esempio Google Container Engine (GKE), Amazon EC2 Container Service e Azure Container Service di Microsoft, che offrono servizi di CaaS nel cloud pubblico. Molti utenti hanno iniziato ad usare Kubernetes attraverso Google Container Engine, essendo uno dei primi servizi di gestione di Kubernetes nel mercato.

2.2 Entità esterne

Nella figura seguente è espresso il *system context diagram* di Kubernetes, che definisce le relazioni tra esso e le entità esterne.



Le entità di maggior rilievo all'interno del diagramma e che verranno analizzate in seguito sono gli stakeholders, il development, le piattaforme e i concorrenti.

2.2.1 Stakeholders

Per poter comprendere al meglio il concetto di stakeholder all'interno di un progetto così grande come quello di Kubernetes, è necessario introdurre il **Cloud Native Computing Foundation (CNCF)**.

CNCF è nato nel 2015 da un accordo tra Google e la Linux Foundation con l'annuncio di Kubernetes 1.0, considerato il progetto principale. Da lì in poi molte industrie del cloud computing si sono unite a CNCF per incubare, sviluppare e mantenere un ecosistema di progetti cloud sotto una visione comune e condivisa. I membri di CNCF sono divisi per categorie, quali Platinum, Gold, Silver, End-User, accademici e no-profit. Tra i membri Platinum abbiamo Google, Docker, IBM, Cisco e Oracle.

Oltre all'organizzazione CNCF, Kubernetes attrae migliaia di contributori che coordinano i loro sforzi attraverso piattaforme online, quali GitHub, Slack e StackOverflow (fornitori). Gli Special Interest Groups (SIG) si occupano dello sviluppo di Kubernetes, in particolare dell'architettura, del product management e del testing, nonché di implementazione specifiche per fornitori di servizi come AWS e Azure.

2.2.2 Development

2.2.3 Piattaforme

2.2.4 Concorrenti

Capitolo 3

Drivers architeturali

Capitolo 4

Struttura

Capitolo 5

Funzioni

Capitolo 6

Comportamento

Capitolo 7

Razionale

Capitolo 8

Aspetti analitici

Capitolo 9

Stili architettureali simili o derivati

Capitolo 10

Referenze