Kubernetes

Fork di Google nasce come proposta Orchestration, cosa è un Orchestrator ?

Largamente conosciuto e supportato resta sicuro è **continuamente mantenuto aggiornato**, chi lo mantiene ?

Pensato in modo da garantire **affidabilità ai progetti più complessi**, permette di averne il controllo, che processi seguono ?

La funzioni di **kubernetes** sono proprio quelle di cui abbiamo bisogno, che **capacità** abbiamo allo stato attuale dei lavori ?

Fork di Google nasce come proposta Orchestration, cosa è un Orchestrator ?

Definizione generica:

Definiamo orchestrazione la configurazione, la gestione e il coordinamento automatici di sistemi informatici, applicazioni e servizi.
È una metodologia che aiuta i team IT a gestire più facilmente attività e flussi di lavoro molto complessi.

Fonte : https://www.redhat.com/it/topics/automation/what-is-orchestration .

Definizione specifica per Kubernetes:

Kubernetes è un software formato da più componenti software disposte secondo il pattern orchestrator.
Tale pattern distingue i partecipanti in master e nodi.
Essi si coordinano per l'esecuzione del carico di lavoro sui server che vanno a formare un cluster controllato, da Kubernetes.

Fonte : https://it.wikipedia.org/wiki/Kubernetes .

Largamente conosciuto e supportato resta sicuro è **continuamente mantenuto aggiornato**, chi lo mantiene ?

Google

CLOUD NATIVE COMPUTING FOUNDATION

Chi è il team responsabile di Kubernetes:

Kubernetes (abbreviato K8s) è un sistema open-source di orchestrazione e gestione di container.
Inizialmente sviluppato da Google, adesso è mantenuto da Cloud Native Computing Foundation.

La Cloud Native Computing Foundation è un progetto della Linux Foundation fondato nel 2015 per aiutare a far progredire la tecnologia dei container e allineare il settore tecnologico alla sua evoluzione.

Fonte primaria : https://it.wikipedia.org/wiki/Kubernetes
Fonte secondaria : https://en.wikipedia.org/wiki/Cloud_Native_Computing_Foundation

Pensato in modo da garantire **affidabilità ai progetti più complessi**, permette di averne il controllo, che processi seguono ?

Gestione di un Sistema che è composto da più sotto sistemi (risorse) :

In Kubernetes lo stato del sistema è descritto mediante l'utilizzo del **concetto di risorsa**.
Le risorse descrivono il **desired state**, cioè lo stato desiderato del sistema.
Una volta creata o modificata la descrizione di una risorsa sul master,
le varie componenti di Kubernetes apportano le necessarie modifiche per variare dallo stato attuale del sistema,
verso lo stato desiderato.

Alcuni esempi di modifiche al sistema possono essere l'avvio di nuovi container e la configurazione della rete su Internet.

L'utilizzo di un modello a risorse permette di descrivere con un **linguaggio dichiarativo** lo stato che il sistema deve assumere,
senza la necessità di conoscere la tecnologia sottostante.
Questo aspetto è particolarmente importante nei **contesti cloud in cui Kubernetes viene offerto come servizio**,
avendo la flessibilità di scelta sulla tecnologia sottostante.

Questo permette di creare **riferimenti tra le varie risorse** per implementare molteplici funzionalità (riferimenti ai *labels*) .

Fonte : https://it.wikipedia.org/wiki/Kubernetes .

Gestione di un Sistema che è composto da più sotto sistemi (risorse)

kubeadm

Risorse del cluster

pod

Il pod è la risorsa che descrive l'unità elementare,
eseguibile su un nodo del cluster.
Un pod raggruppa dei container che condividono le risorse,
e che vengono eseguiti sullo stesso nodo.
Il pod si occupa di astrarre rete e storage al fine di poter essere spostato
e replicato facilmente sui nodi del cluster,
permettendo una forte scalabilità orizzontale,
in particolare alle applicazioni orientate ai microservizi.

service

Il *service* definisce come esporre dei pod su una rete interna o esterna .
Il service definisce un nome che viene risolto dal DNS interno al cluster
con uno dei pod a esso associati .
I pod associati al service sono quelli che hanno in comune la label ,
definita dal service .
Di default un service è esposto all'interno di un cluster .

Fonte : https://it.wikipedia.org/wiki/Kubernetes

La funzioni di **kubernetes** sono proprio quelle di cui abbiamo bisogno, che **capacità** abbiamo allo stato attuale dei lavori ?

Lavoriamo per padroneggiare **soluzioni** che ci diano :

Vantaggi tecnici *: Capacità di mantenere , preservare e ristrutturare i dati .*
*Vogliamo e possiamo fare questo con **semplicità** e **senza limitazioni** di tempo ,*
e/o di capacità d'azione .
Vogliamo che agendo sulla fonte primaria i dati così costituiti
*vengano gestiti e mantenuti da **algoritmi** affidabili e totalmente **autonomi** .*

Vantaggi che derivano dai vantaggi tecnici **in** ambito di **produzione** :

Soluzione Affidabile *:*

Consistente *: I dati creati dagli algoritmi (autonomi e non), sono **integri e inalterati** .*
Essi possono essere usati senza conseguenze inaspettate ,
per essere poi così indirizzati ai servizi per cui sono destinati .

Scalabile *: Nel Sistema Kubernetes la scalabilità dona al sistema la capacità di **aumentare o diminuire** **l'architettura** effettivamente usata dai processi .*
*Questo in funzione delle necessità e delle disponibilità aiuta a **incrementare le prestazioni** .*