



KUBERNETES

Mikä se on ja mihin sitä voi käyttää

Kubernetes historia

Kubernetes perustuu Googlen kehittämään konesalin hallintajärjestelmään. Lähes kaikki Googlen sovellukset perustuvat kontteihin, joita hallitaan Googlen itse kehittämällä orkestrointityökalulla nimeltä **Borg**.



Figure 1.4 - The Kubernetes logo

”Kubernetes” tulee kreikan kielestä ja tarkoittaa **perämiestä** tai **ohjaajaa**. Kuten perämies ohjaa laivaa, Kubernetes hallinnoi ja ohjaa konttien toimintaa sovellusympäristössä.

Kenelle Kubernetes sopii?



Kubernetes sopii erityisen hyvin organisaatioille, jotka toimivat pitkälti pilviympäristössä, ja joilla on monimutkainen ja/tai joustavuutta ja skaalautuvuutta edellyttävä ICT-toimintaympäristö.

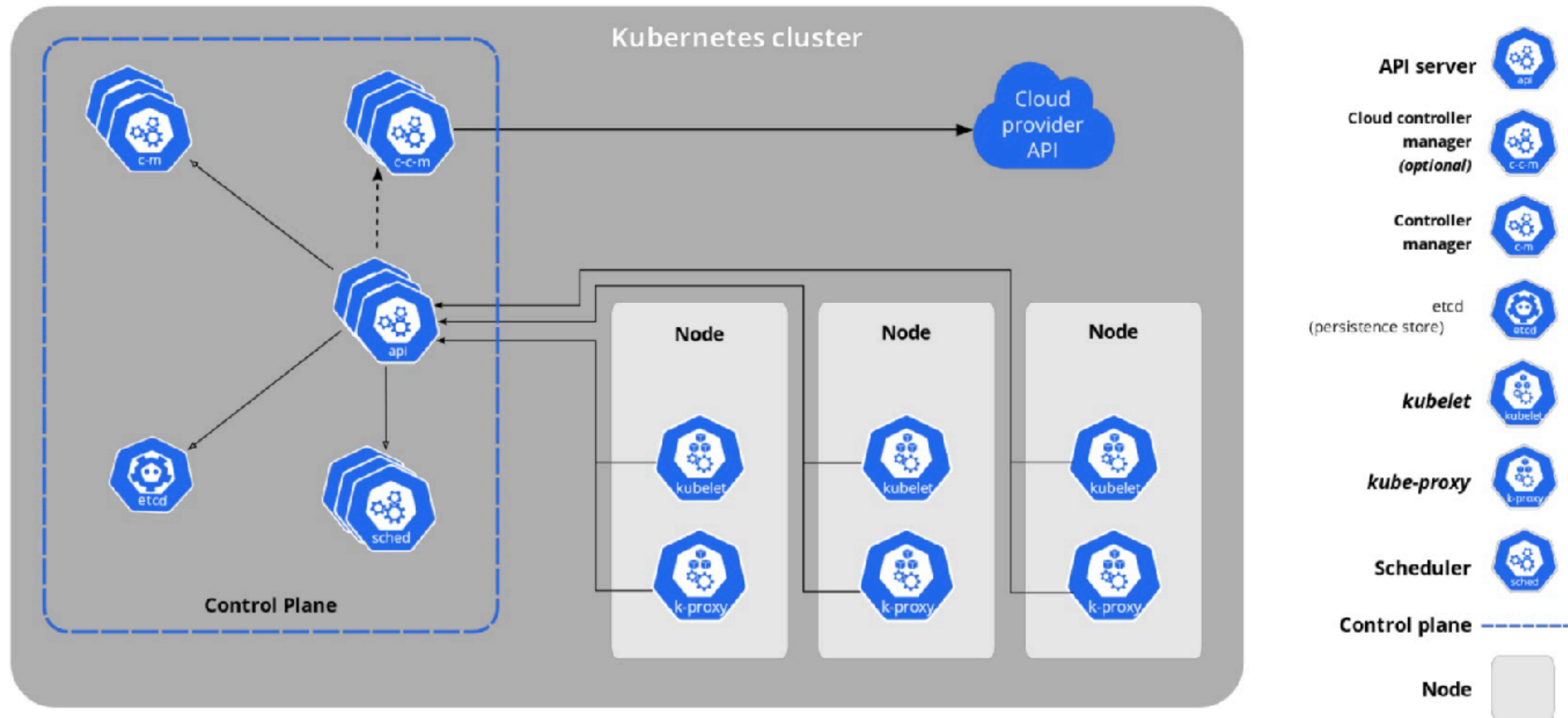
Mihin Kubernetesa tarvitaan?

.Kun useasta kontista koostuva sovellus halutaan käynnistää tuotantoympäristössä, pitää tietää, mitä kaikkia kontteja sovelluksen ajamiseen tarvitaan. Kubernetes on työkalu, joka tietää, mitä kontteja sovellus tarvitsee. Kuberneteselle kerrotaan, mitä kontteja tarvitaan, ja kun sovellus halutaan käynnistää, se osaa käynnistää kaikki tarvittavat kontit.

- Koska tuotantoympäristössä ei kontteja yleensä ajeta yhden palvelimen päällä, **pitää sovellusta käynnistettäessä tietää, mille palvelimille kontit kannattaa sijoittaa**. Kubernetes on työkalu, joka sijoittaa kontit tuotantoympäristöön. Kubernetesin avulla siis päätetään, millä palvelimilla ajetaan mitään kontteja.
- **Kun tuotantoympäristöön on käynnistetty kontteja, pitäisi tietää, millä palvelimilla on mitään kontteja ajossa**. Kubernetes pitää kirjaa siitä, mitä kontteja milläkin alustapalvelimella ajetaan.
- **Kubernetesin avulla voidaan seurata, toimivatko kontit halutusti**. Sen avulla voidaan käynnistää vikaantuneita kontteja automaattisesti uudelleen.
- **Kubernetesin avulla voidaan huolehtia sovelluksen skaalaamisesta kuormituksen lisääntyessä**. Jos sovellus kuormittuu ja tarvitsee lisää kapasiteettia, osaa Kubernetes käynnistää automaattisesti uusia kontteja ja huolehtia niiden välisestä kuorman jaosta.
- **Kubernetesin avulla voidaan päivittää sovelluksia, jotka koostuvat useista konteista**. Kun kontteihin halutaan tehdä muutoksia, osaa Kubernetes päivittää kontit. Jos muutos ei suju halutusti, Kubernetesin avulla voidaan ottaa käyttöön aiempi kontti.
- Kubernetesin avulla annetaan konteille niiden tarvitsemat IP-osoitteet ja huolehditaan sovelluksen julkaisemisesta käyttäjille

Kubernetesin klusteri

Periaatekaavio



Kubernetesin orkestrointi

Kubernetes on avoimen lähdekoodin järjestelmä, joka mahdollistaa konttien orkestroinnin, mikä tarkoittaa konttipohjaisten sovellusten automaattista hallintaa ja skaalaamista. Orkestrointi kattaa useita tärkeitä toimintoja, kuten:

Sovellusten käyttöönotto: Kubernetes mahdollistaa konttien yksinkertaisen käyttöönoton ja hallinnan.

Skaalaaminen: Voit helposti lisätä tai vähentää konttien määrää kuormituksen mukaan.

Itsenäinen palautuminen: Kubernetes valvoo kontin tilaa ja käynnistää sen uudelleen, jos se epäonnistuu.

Palveluiden hallinta: Voit hallita ja yhdistää palveluita sekä varmistaa niiden saatavuuden.

Kubernetesin keskeiset komponentit

Kubernetes koostuu useista tärkeistä komponenteista, jotka yhdessä mahdollistavat orkestroinnin:

1. Master Node

- Vastaa koko klusterin hallinnasta.
- Koostuu useista komponenteista, kuten API-serveristä, ohjauslohkosta ja aikataulutustajasta.

2. Worker Nodes

- Suorittavat varsinaiset sovellukset ja kontit.
- Sisältävät kubeletin, joka kommunikoi master-nodejen kanssa, ja kube-proxyn, joka hallitsee verkkoliikennettä.

3. Podit

- Perusyksiköt, jotka sisältävät yhden tai useamman kontin.
- Jakavat resurssit ja voivat kommunikoida keskenään.

4. Service

- Varmistaa, että kontit ovat käytettävissä, jopa kun ne skaalautuvat tai menevät alas.
- Tarjoaa pysyvän IP-osoitteen ja DNS-nimen.

5. Volume

- Tallennustilan hallinta, joka mahdollistaa datan säilyttämisen konttien välillä.
- Tukee erilaisia tallennusteknologioita.



Orkestroinnin hyödyt

Kubernetesin orkestrointi tarjoaa useita etuja:

Automaattinen Skaalaaminen: Voit määrittää säännöt, jotka säätelevät, kuinka monta konttia tarvitaan eri kuormitustilanteissa.

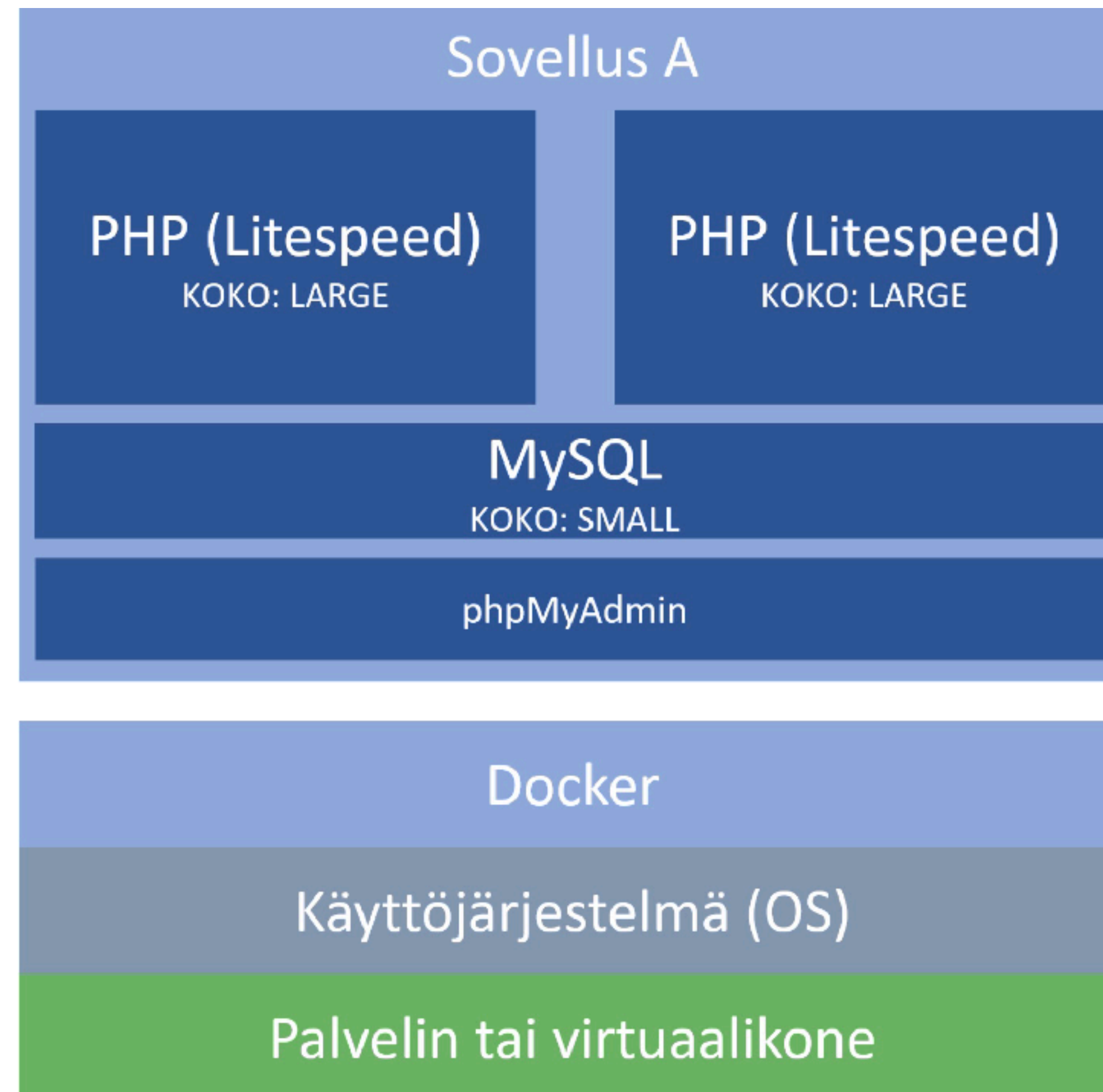
Resurssien Optimointi: Kubernetes voi optimoida resurssien käytön, mikä vähentää kustannuksia ja parantaa suorituskykyä.

Korkea Saatavuus: Sen automaattisen palautumisen ansiosta sovellukset pysyvät käytettävissä myös vikatilanteissa.

Helpompi Hallinta: Hallinta ja käyttöönotto ovat yksinkertaisempia, koska Kubernetes automatisoi monia prosesseja.

Onko Kubernetesillä kilpailijoita?

Docker



Kubernetes vs Virtuaalipalvelin

Virtuaalipalvelin on aina yksilöidystä käyttöjärjestelmästä riippuvainen

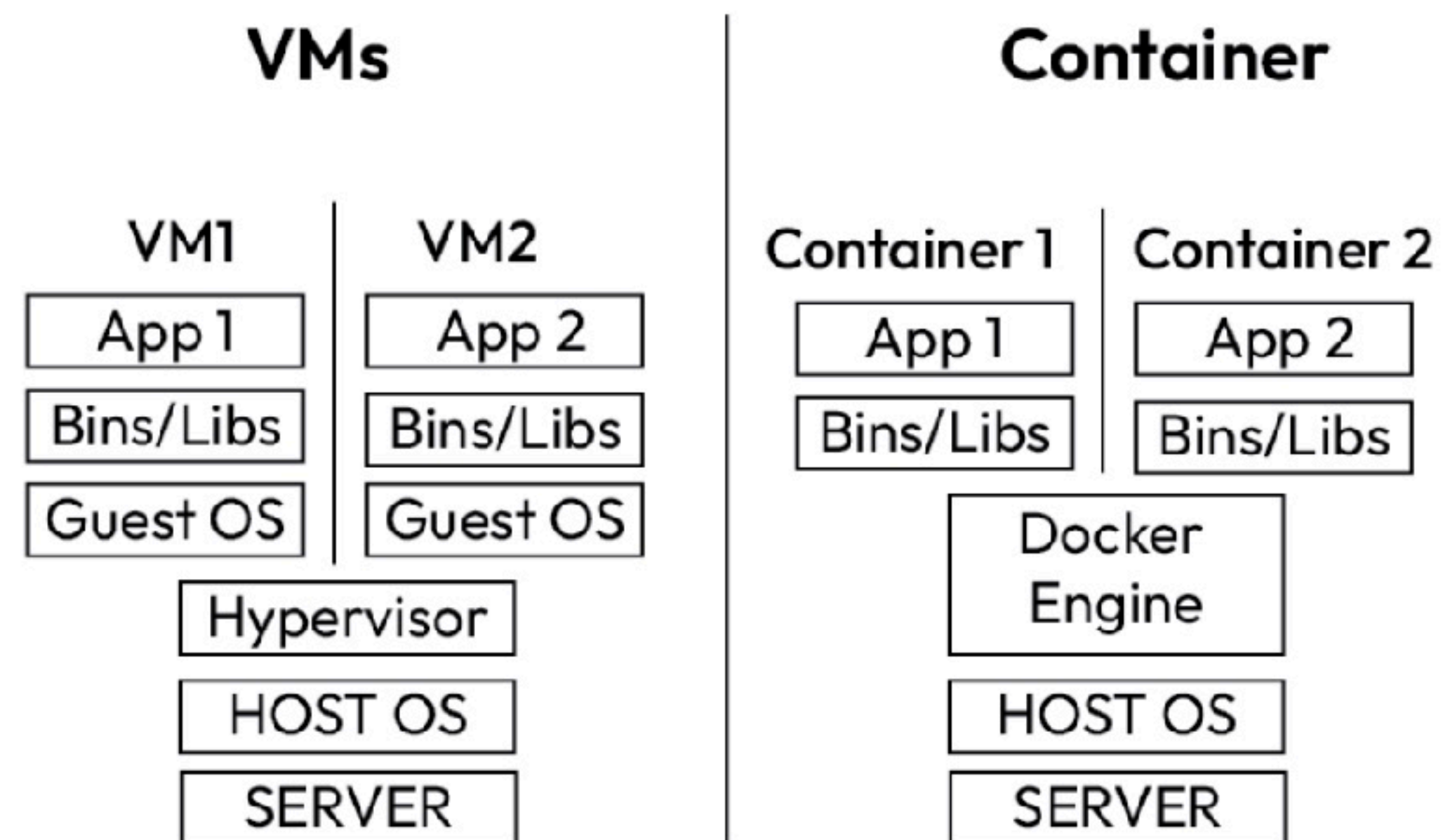


Figure 1.1 – VMs versus containers

Kubernetes klusteri

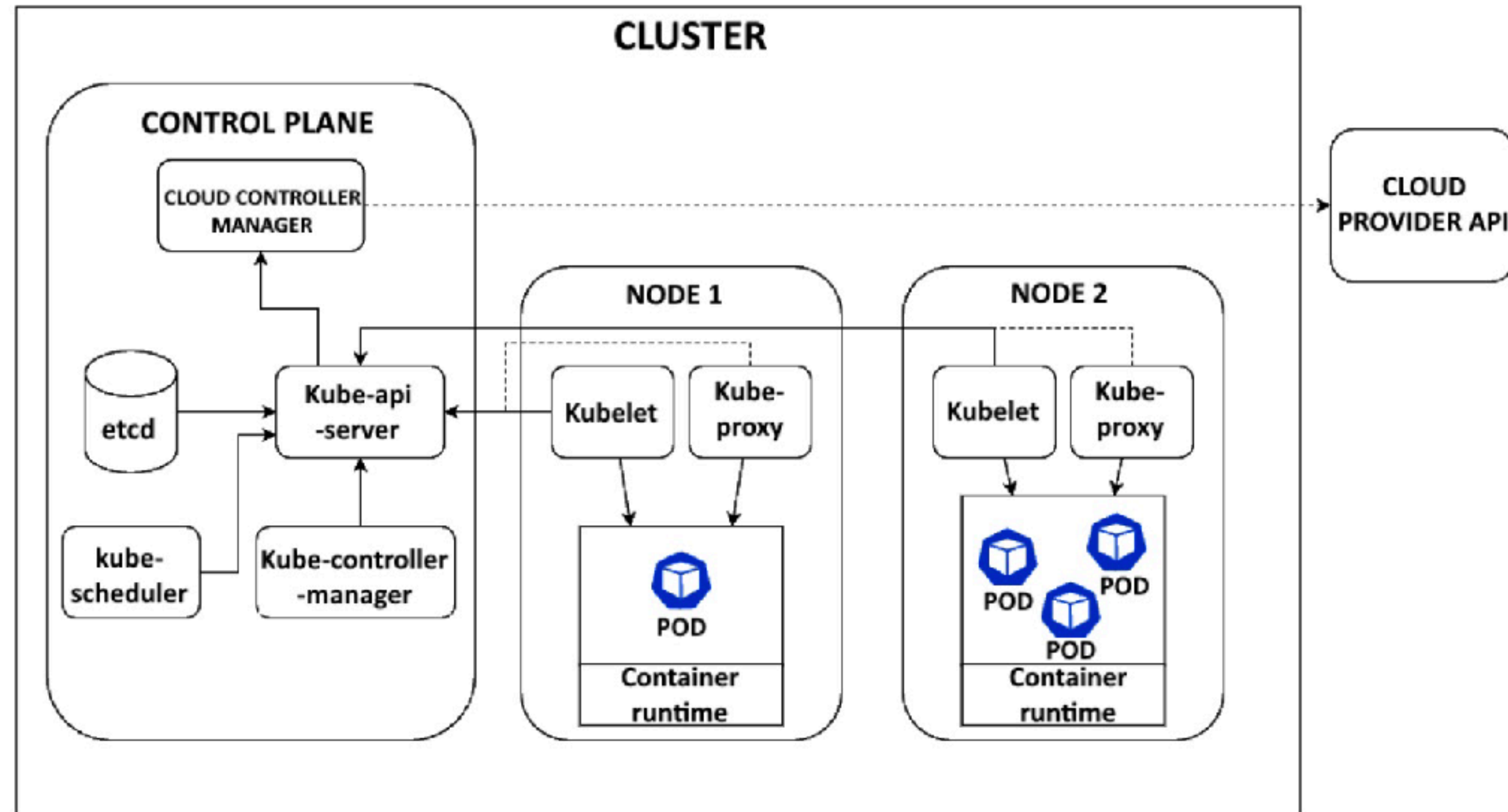


Figure 2.1 – Kubernetes architecture

Mikä ero on Kubernetesilla ja Dockerilla? 1

Docker

Mikä on Docker?

Docker on alusta, joka mahdollistaa sovellusten pakkaamisen, jakamisen ja toimittamisen **konttien** avulla. **Kontti on eristetty** ympäristö, joka sisältää sovelluksen ja kaikki sen riippuvuudet, kuten kirjastot ja asetukset.

Mitä Docker tekee?

Docker **keskittyy sovellusten luomiseen ja hallintaan eristyksissä**, mikä helpottaa kehitysprosessia ja parantaa sovellusten siirrettävyyttä eri ympäristöissä.

Mikä ero on Kubernetesilla ja Dockerilla? 1

Kubernetes

Mikä on Kubernetes?

Kubernetes on avoimen lähdekoodin **orkestrointialusta**, joka hallitsee ja **automatisoi konttipohjaisten sovellusten käyttöönottoa, skaalautumista ja hallintaa.**

Mitä Kubernetes tekee?

Kubernetes **koordinoi useita kontteja** ja varmistaa, että ne toimivat yhdessä oikein, **mahdollistaa automaattisen skaalaamisen ja itsenäisen palautumisen vikatilanteissa.**

Mikä ero on Kubernetesilla ja Dockerilla? 2

Dockerin Käyttötarkoitukset

Sovellusten kehitys: Dockerin avulla kehittäjät voivat luoda ja testata sovelluksia eristyksissä, mikä parantaa kehitysprosessia.

Käyttöönotto: Docker helpottaa sovellusten käyttöönottoa eri ympäristöissä, kuten kehitys-, testaus- ja tuotantoympäristöissä.

Mikropalveluarkkitehtuuri: Docker tukee mikropalveluarkkitehtuuria, jossa sovellukset jaetaan pienempiin, itsenäisiin palveluihin.

Mikä ero on Kubernetesilla ja Dockerilla? 2

Kubernetesin Käyttötarkoitukset

Orkestrointi: Kubernetes **hallitsee useita kontteja**, niiden riippuvuuksia ja viestintää, jolloin kehittäjien ei tarvitse huolehtia manuaalisesta hallinnasta.

Skaalaaminen: Kubernetes voi **automaattisesti lisätä tai vähentää konttien määrää kuormituksen mukaan**.

Palautuminen: Kubernetes **valvoo kontin tilaa ja käynnistää sen uudelleen, jos se epäonnistuu**, mikä lisää sovellusten saatavuutta ja luotettavuutta.

Mikä ero on Kubernetesilla ja Dockerilla? 3

Dockerin Ominaisuudet

Konttien luominen: Docker mahdollistaa konttien helpon luomisen ja hallinnan Dockerfile-tiedostojen avulla.

Docker Hub: Pilvipohjainen säilytyspaikka, josta kehittäjät voivat ladata ja jakaa Docker-kuvia.

Docker Compose: Työkalu, joka mahdollistaa useiden konttien määrittelyn ja hallinnan yhdellä komennolla, mikä on erityisen hyödyllistä monimutkaisissa sovelluksissa.

Mikä ero on Kubernetesilla ja Dockerilla? 3

Kubernetesin Ominaisuudet

Orkestrointi: Kubernetes hallitsee useita kontteja ja varmistaa, että ne toimivat yhdessä oikein.

Palvelun hallinta: Kubernetes mahdollistaa palveluiden automatisoidun hallinnan, mukaan lukien kuormituksen tasaamisen ja palveluiden yhdistämisen.

Itsenäinen palautuminen: Kubernetes valvoo kontin tilaa ja käynnistää sen uudelleen, jos se epäonnistuu.

Mikä ero on Kubernetesilla ja Dockerilla? 4

Toimintatapa

Docker keskittyy yksittäisten konttien luomiseen ja hallintaan. Se on työkalu, joka mahdollistaa sovellusten pakkaamisen ja jakamisen.

Kubernetes toimii orkestrointialustana, joka hallitsee useita kontteja ja niiden vuorovaikutusta. Se ei luo kontteja, vaan hallitsee niitä, kun ne on jo luotu Dockerin avulla.

Käyttöliittymä

Docker tarjoaa *komentorivipohjaisen käyttöliittymän*, jossa kehittäjät voivat suorittaa komentoja konttien hallitsemiseksi.

Kubernetes tarjoaa *laajemman ja monimutkaisemman käyttöliittymän*, joka voi sisältää myös *graafisia käyttöliittymiä*, kuten Kubernetes Dashboardin, joka helpottaa klusterin hallintaa.

Skaalautuvuus

Docker itsessään *ei tarjoa laajaa skaalausmahdollisuutta*, vaan se keskittyy yksittäisten konttien hallintaan.

Kubernetes mahdollistaa *automaattisen skaalaamisen*, jolloin se voi lisätä tai vähentää konttien määrää perustuen kuormitustietoihin.

Docker ja Kubernetesin haasteet

Dockerin Haasteet

Turvallisuus: Konttien eristyksestä huolimatta turvallisuus voi olla huolenaihe, erityisesti kun käytetään epäluotettavia kuvia.

Monimutkaisuus: Dockerin käyttö voi aluksi tuntua monimutkaiselta, erityisesti uusille käyttäjille (merkkipohjaisuus).

Kubernetesin Haasteet

Oppimiskäyrä: Kubernetesin oppimiskäyrä voi olla jyrkkä, ja sen hallintaan vaaditaan erityistä osaamista.

Konfigurointi: Oikean konfiguroinnin löytäminen voi olla aikaa vievää, ja väärin määritetyt resurssit voivat johtaa ongelmiin.

Docker ja Kubernetes yhteenveto

Yhteenveto

Docker ja Kubernetes ovat molemmat olennaisia työkaluja nykyaikaisessa ohjelmistokehityksessä, mutta niiden roolit ovat erilaisia. Docker keskittyy konttien luomiseen ja hallintaan, kun taas Kubernetes hallitsee ja orkestroii useita kontteja, varmistaen niiden sujuvan toiminnan yhdessä.

Yhteenvetona voidaan todeta, että Docker on erinomainen valinta sovellusten pakkaamiseen ja kehittämiseen, kun taas Kubernetes on paras valinta, kun tarvitaan monimutkaisempaa konttien hallintaa, automaattista skaalaamista ja korkean saatavuuden varmistamista. Molempia teknologioita voidaan käyttää yhdessä, jolloin kehittäjät voivat hyödyntää kummankin parhaita puolia.

Voiko Kubernetesia käyttää muihin tietoteknisiin tarkoituksiin?

1. Mikropalveluarkkitehtuuri

Kubernetesia voidaan käyttää hallitsemaan monimutkaisempia mikropalveluarkkitehtuureja, joissa sovellukset koostuvat useista pienistä, itsenäisistä palveluista.

Se mahdollistaa palveluiden välisten yhteyksien hallinnan ja skaalaamisen tarpeen mukaan.

2. Datapalveluiden hallinta

Kubernetes voi hallita tietokantoja ja muita datapalveluita, kuten:

PostgreSQL; MySQL; MongoDB

Se mahdollistaa tietokantojen klusteroinnin, varmuuskopioinnin ja palautuksen sekä skaalaamisen.

3. Kontekstin ja resurssien hallinta

Voit käyttää Kubernetesia erilaisten resurssien, kuten muistien, CPU:n ja tallennustilan, hallintaan.

Se voi auttaa optimoimaan resurssien käyttöä ja parantamaan tehokkuutta.

Voiko Kubernetesistä käyttää vielä muihin tietoteknisiin tarkoituksiin?

4. Kubernetesin laajennettavuus

Kubernetesin ekosysteemissä on monia laajennuksia ja työkaluja, jotka mahdollistavat sen käytön myös muihin tarkoituksiin, kuten:

CI/CD-prosessit: Voit automatisoida ohjelmiston *jatkuvan integroinnin* ja *toimituksen*.

Kubernetes Operators: Nämä ovat ohjelmistoja, jotka automatisoivat sovellusten hallintatehtäviä Kubernetesin sisällä.

5. Edge-laskenta

Kubernetesia voidaan käyttää myös edge-laskennassa, jossa sovelluksia suoritetaan lähellä datan lähdettä, kuten IoT-laitteet.

Tämä auttaa vähentämään viiveitä ja parantamaan suorituskykyä.

6. Koulutus- ja testausympäristöt

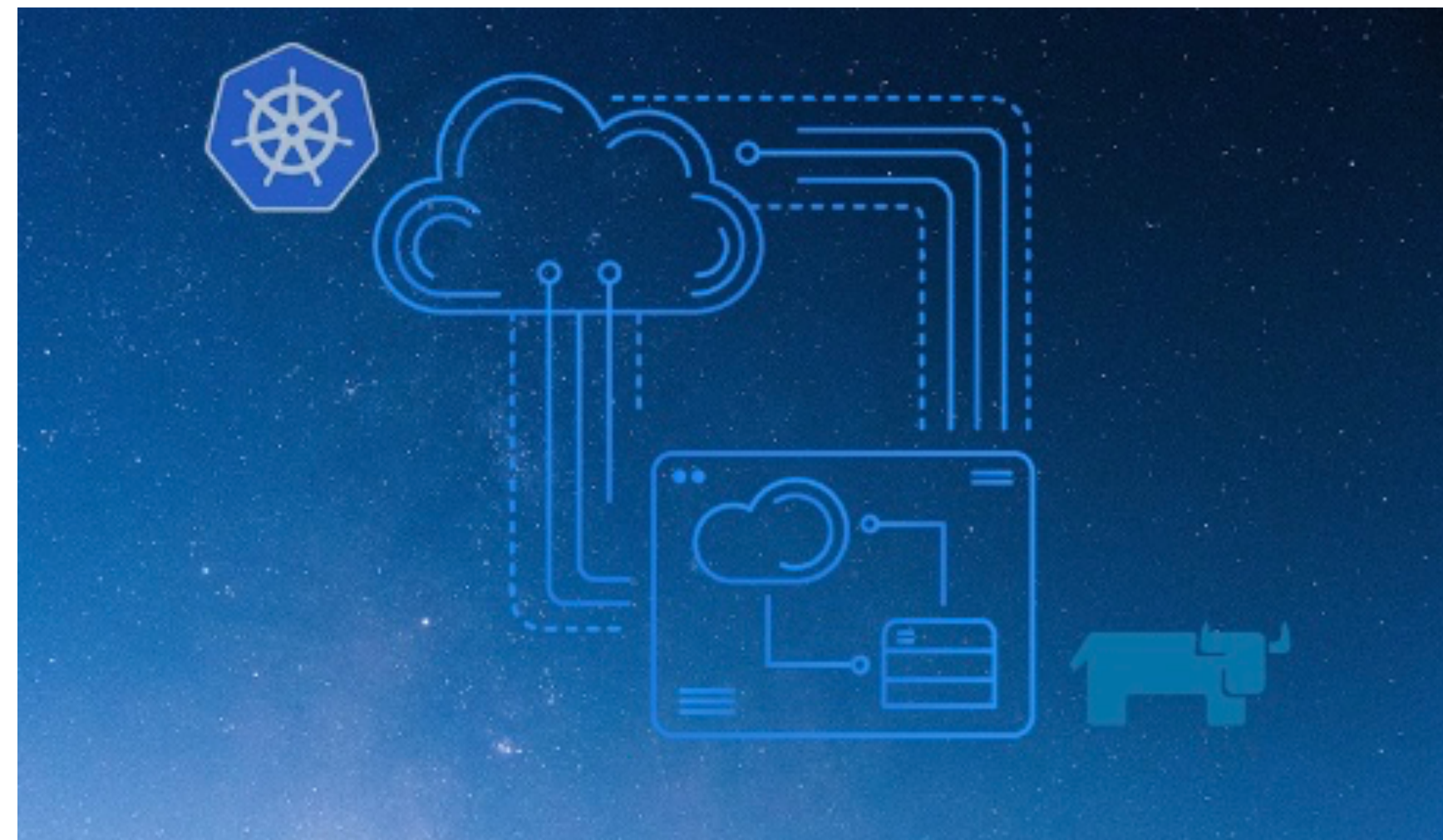
Kubernetes voi olla hyödyllinen koulutuksessa ja testauksessa, sillä voit helposti luoda ja hallita eristyksissä olevia ympäristöjä eri projekteille.

7. Kubernetesin yhdistäminen muihin teknologioihin

Voit integroida Kubernetesin muiden teknologioiden, kuten **Apache Kafka** (tiedonsiirtoon) tai **TensorFlow** (koneoppimiseen), kanssa.

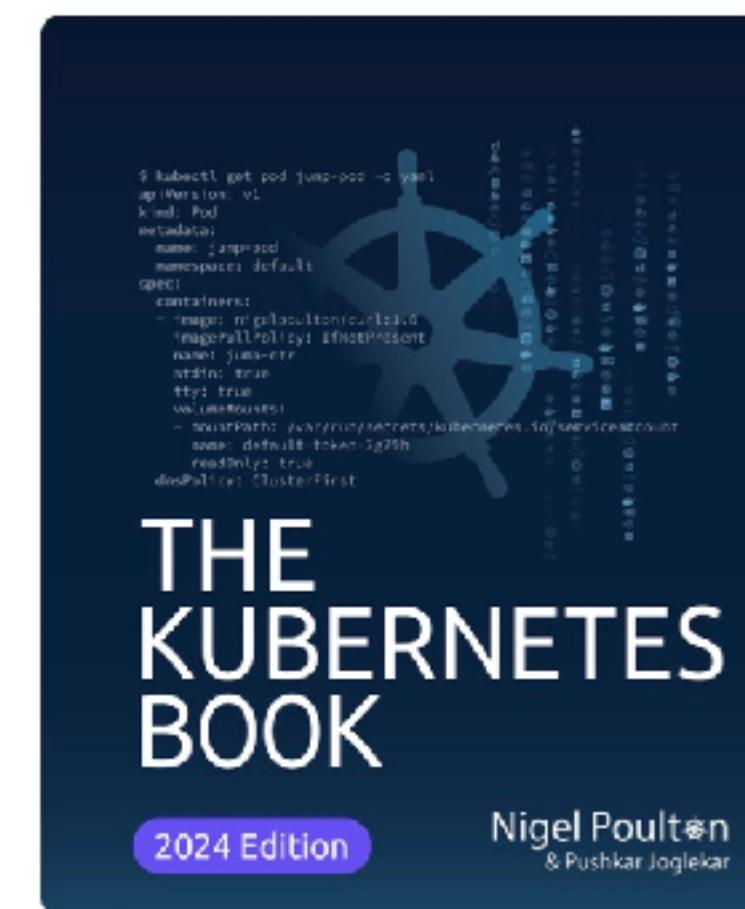
Yhteenveto

Kubernetes on nykyään erittäin monipuolinen työkalu, joka ei rajoitu vain sovellusten hallintaan. Sen kyky hallita resursseja, laajentaa ympäristöjä ja integroida muihin teknologioihin tekee siitä erinomaisen valinnan monenlaisiin käyttötapoihin.



Lähteet

Kuvat ja periaatekaaviot ovat kopioitu internetin ilmaisjakeluista ja oikealla mainituista kirjoista.



<https://packt.link/free-ebook/978-1-83546-214-0>

Myös Opera-selaimen Aria-tekoälyosiota on hyödynnetty.



Nigel Poulton

(c) 2024 Nigel Poulton Ltd.



@nigelpoulton

nigelpoulton.com/books

qskbook@nigelpoulton.com

