
LEARNING PROGRESS REVIEW “WEEK 7”

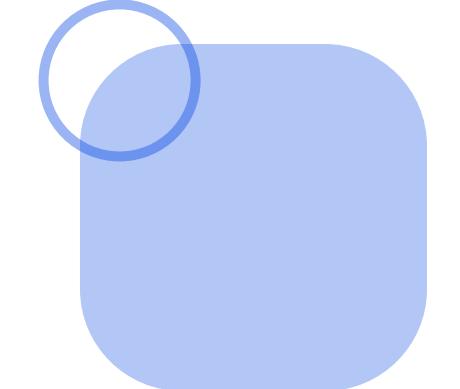
BY & DREAM (KELOMPOK 1)

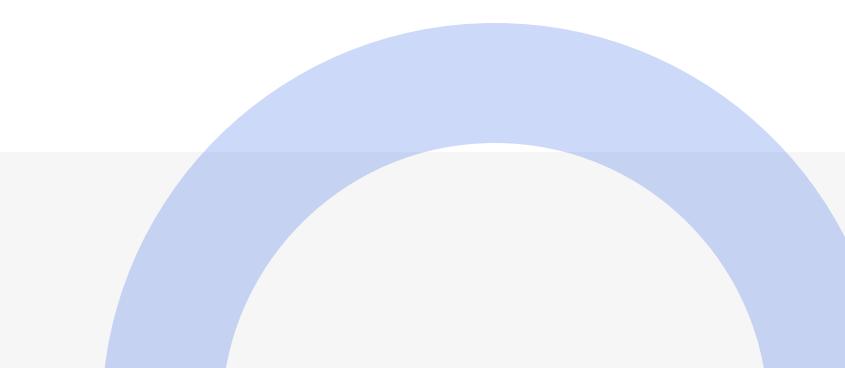
Data Realm Engineers And Maestros





ANGGOTA KELOMPOK



- 01 Afroh Fauziah**
 - 02 Althaf Nawadir Taqiyyahh**
 - 03 Andi Rosilala**
 - 04 Andrew Bintang Pratama**
 - 05 Andrew Fortino Mahardika Suadnya**
- 

**“DOCKER”
BASIC**

USER ENVIRONMENT

Beberapa perintah untuk membuat daftar dan mengatur environment variable (bisa disimpan, digunakan, dihapus, dan diyesuaikan) di Linux:

- env > Perintah untuk menjalankan program lain di lingkungan khusus tanpa mengubah yang sekarang.
- printenv > print environment variables yang ingin dilihat, mencetak nilai dari satu atau beberapa variabel lingkungan tertentu.
- set > memasukan nilai ke dalam sebuah environment variable, menampilkan juga semua fungsi & perintah, biasa digunakan di windows.
- unset > menghapus variabel environment atau fungsi tertentu.
- export > memasukan nilai ke dalam sebuah environment variable, biasa digunakan di linux.

TEST USER ENVIRONMENT

Masuk ke user lalu ketik printenv HOME (sesuai kebutuhan)

```
C:\Users\afroh>ssh digitalskola00@35.222.31.142
digitalskola00@35.222.31.142's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1053-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Tue Apr  2 05:42:48 UTC 2024

System load:  0.0          Users logged in:          0
Usage of /:   38.0% of 19.20GB   IPv4 address for br-4117523d7eee: 172.18.0.1
Memory usage: 9%
Swap usage:  0%
Processes:   118           IPv4 address for br-81e8d556b043: 172.20.0.1
                           IPv4 address for docker0:    172.17.0.1
                           IPv4 address for ens4:      10.128.0.3

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

  https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

16 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Mon Apr  1 15:48:02 2024 from 110.137.80.52
digitalskola00@instance-20240223-081149:~$
```



```
digitalskola00@instance-20240223-081149:~$ printenv HOME
/home/digitalskola00
```

— LINUX COMMANDS



- **pwd** (Print Working Directory): mencetak nama direktori kerja saat ini (direktori berada).
- **cat** (Concatenate and Print): menggabungkan file dan mencetak isi file ke terminal.
- **cp** (Copy): menyalin file atau direktori dari satu lokasi ke lokasi lain.
- **mv** (Move): memindahkan atau mengubah nama file atau direktori dari satu lokasi ke lokasi lain.
- **rm** (Remove): menghapus file atau direktori dari sistem file.
- **touch**: membuat file kosong baru. Jika file yang ditentukan tidak ada, touch akan membuatnya, jika sudah ada, itu akan mengubah tanggal dan waktu modifikasinya.
- **mkdir** (Make Directory): membuat direktori baru.

TEST LINUX COMMANDS

Membuat env terlebih dahulu dengan perintah mkdir

```
digitalskola00@instance-20240223-081149:~$ pwd  
/home/digitalskola00  
digitalskola00@instance-20240223-081149:~$ mkdir sib6-afroh  
digitalskola00@instance-20240223-081149:~$ ls -l  
total 100  
drwxrwxr-x 2 digitalskola00 digitalskola00 4096 Feb 24 03:27 fadlli  
drwxrwxr-x 2 digitalskola00 digitalskola00 4096 Apr 1 12:51 fauzanpom  
drwxrwxr-x 2 digitalskola00 digitalskola00 4096 Feb 24 03:28 gunawan  
drwxrwxr-x 3 digitalskola00 digitalskola00 4096 Feb 27 13:29 imsal  
drwxrwxr-x 2 digitalskola00 digitalskola00 4096 Feb 27 12:40 latihan  
drwxrwxr-x 2 digitalskola00 digitalskola00 4096 Feb 27 12:52 practice  
drwxrwxr-x 2 digitalskola00 digitalskola00 4096 Feb 24 03:32 rian  
drwxr-xr-x 5 root root 4096 Feb 27 13:05 shared  
drwxrwxr-x 2 digitalskola00 digitalskola00 4096 Apr 1 12:57 sib-fauzanpom  
drwxrwxr-x 2 digitalskola00 digitalskola00 4096 Apr 1 13:14 sib-zita  
drwxrwxr-x 2 digitalskola00 digitalskola00 4096 Apr 2 05:48 sib6-afroh  
drwxrwxr-x 2 digitalskola00 digitalskola00 4096 Apr 1 13:51 sib6-afroh
```



Masuk ke folder dan buat filenya lalu tampilkan

```
digitalskola00@instance-20240223-081149:~$ cd sib6-afroh/  
digitalskola00@instance-20240223-081149:~/sib6-afroh$ nano my-script.sh  
digitalskola00@instance-20240223-081149:~/sib6-afroh$ ls -l  
total 4  
-rw-rw-r-- 1 digitalskola00 digitalskola00 27 Apr 2 06:03 my-script.sh  
digitalskola00@instance-20240223-081149:~/sib6-afroh$ cat my-script.sh  
#!/bin/sh  
echo Hello World
```

```
digitalskola00@instance-20240223-081149:~/sib6-afroh  
GNU nano 4.8  
#!/bin/sh  
echo Hello World
```

[Wrote 2 lines]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^L Replace ^U Paste Text ^T To Spell ^I Go To Line M-E Redo
M-A Mark Text M-6 Copy Text

SHELL SCRIPT PROGRAMMING

- Shell > program yang menjembatani user dengan system operasi (kernel), seperti command prompt.
- Shell script programming > kegiatan menyusun beberapa perintah shell menjadi kumpulan perintah dalam sebuah file teks, yang kemudian dapat dijalankan / dieksekusi oleh shell sesuai keinginan user.

Perintah - perintah dasar shell untuk melakukan shell script programming:

- cd, pwd, times, alias, umask, exit, logout, fg, bg, ls, mkdir, rmdir, mv, cp, rm, clear, dsb.
- utilitas seperti cat, cut, paste, chmod, lpr, dsb.
- redirection (cara mengirim output ke file atau menerima input dari file), menggunakan operator redirect >, >>, <, <<

nano tesknya dan edit lalu tampilkan dengan perintah bash

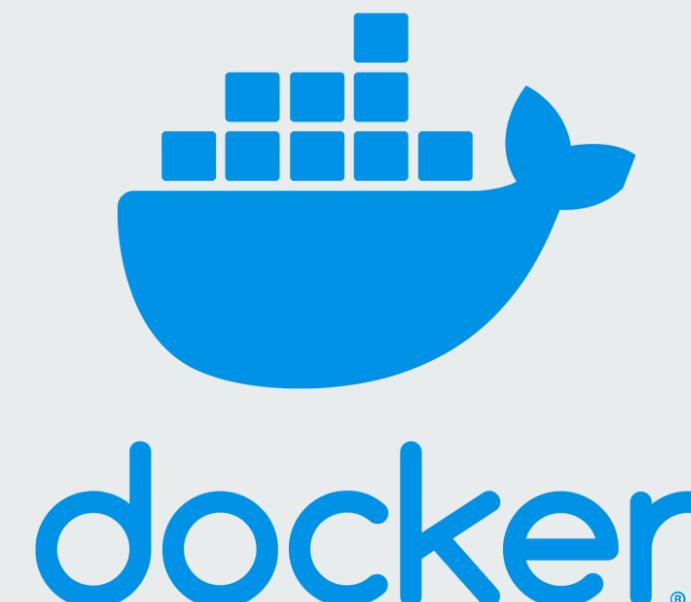


```
digita...@instance-20240223-081149:~/sib6-afroh$ bash my-script.sh
Hello World
```

PENGENALAN DOCKER

Docker adalah platform perangkat lunak yang digunakan untuk menyederhanakan pengembangan, menguji, distribusi, dan menjalankan aplikasi dalam lingkungan yang konsisten dan terisolasi (tidak terganggu oleh aplikasi lain) yang disebut "kontainer" (untuk menjalankan apk) di seluruh siklus hidup aplikasi. Fungsi docker sebagai deployment (upload aplikasi) ke sebuah server.

- Lingkungan pengembangan yang terisolasi.
- Pengujian cepat tanpa 'merusak' host.
- Development tools / alat pengembangan yang ringan.



- Membantu dalam standarisasi development.
- Repositori publik banyak image yang siap pakai : Airflow, Flask, Redash, Metabase, Laravel, Codeigniter.

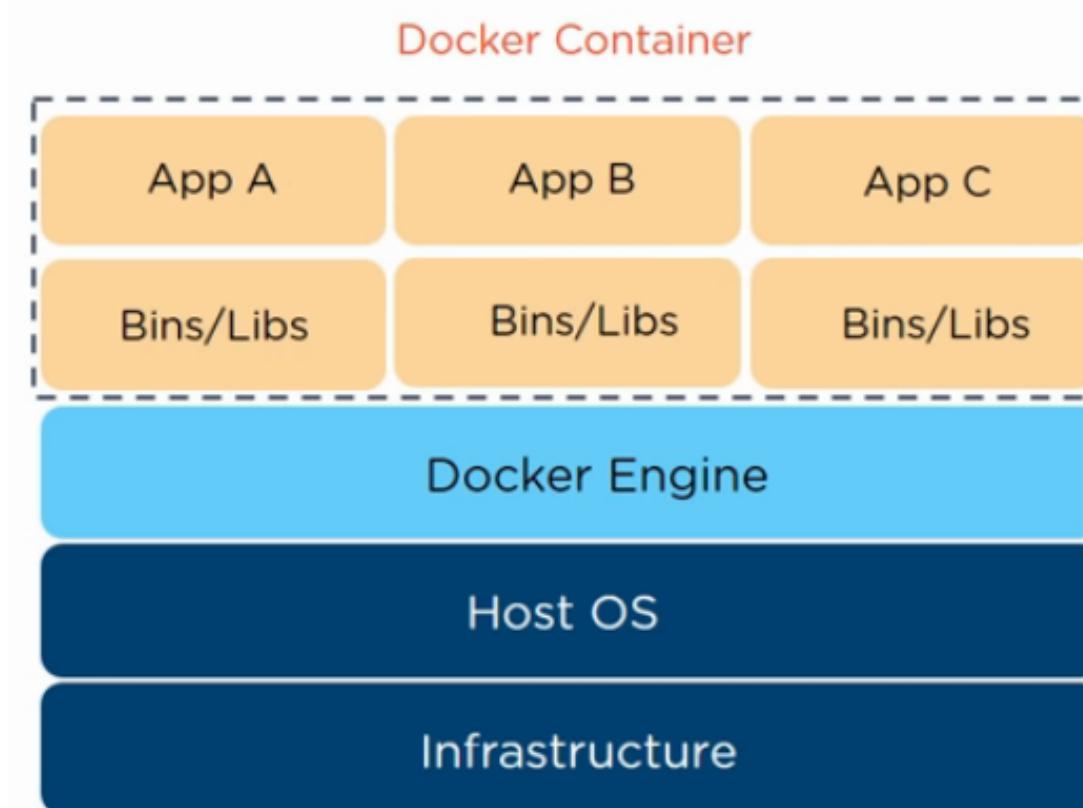
— DOCKER VS VM

Docker

~ Menggunakan isolasi sumber daya yang ringan, berbagi kernel host.

~ Dapat dikelola dengan mudah dan cepat menggunakan Dockerfile dan Docker Compose.

~ Dapat meng-konsolidasikan dan membagi sumber daya secara efisien.

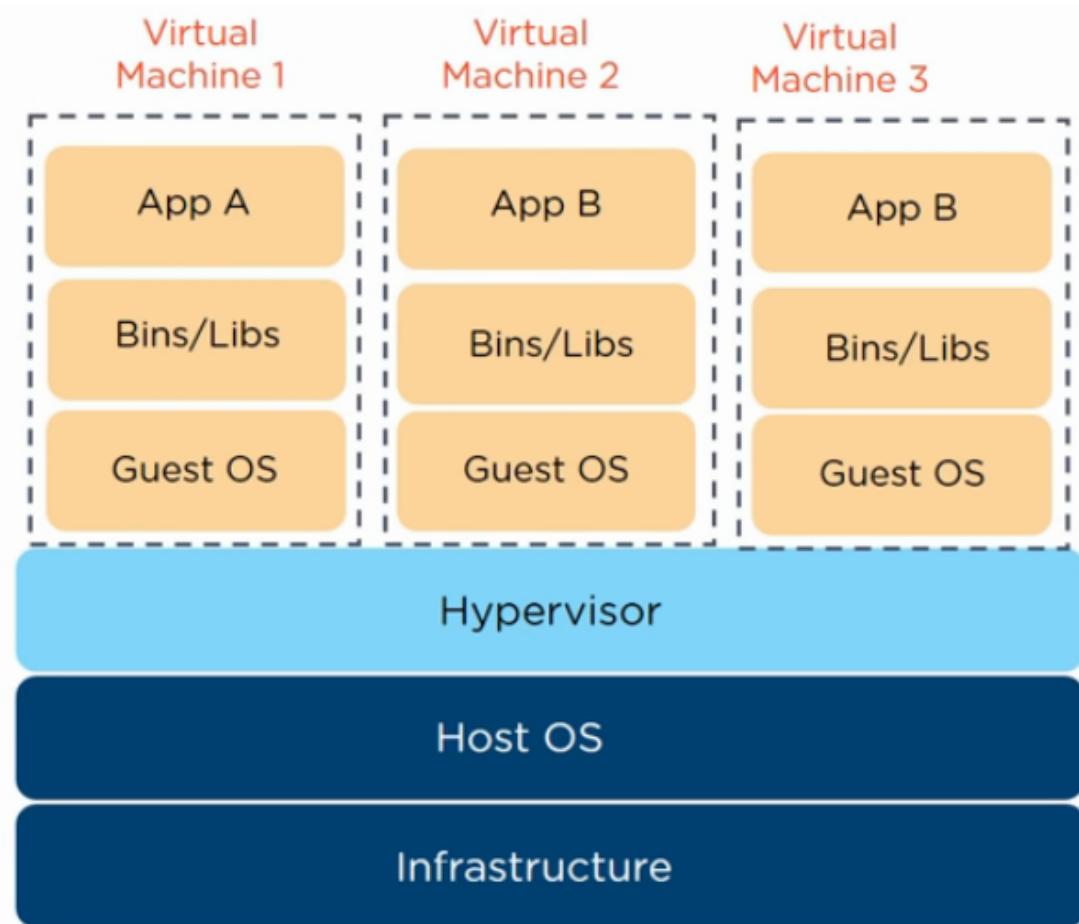


Virtual Machine

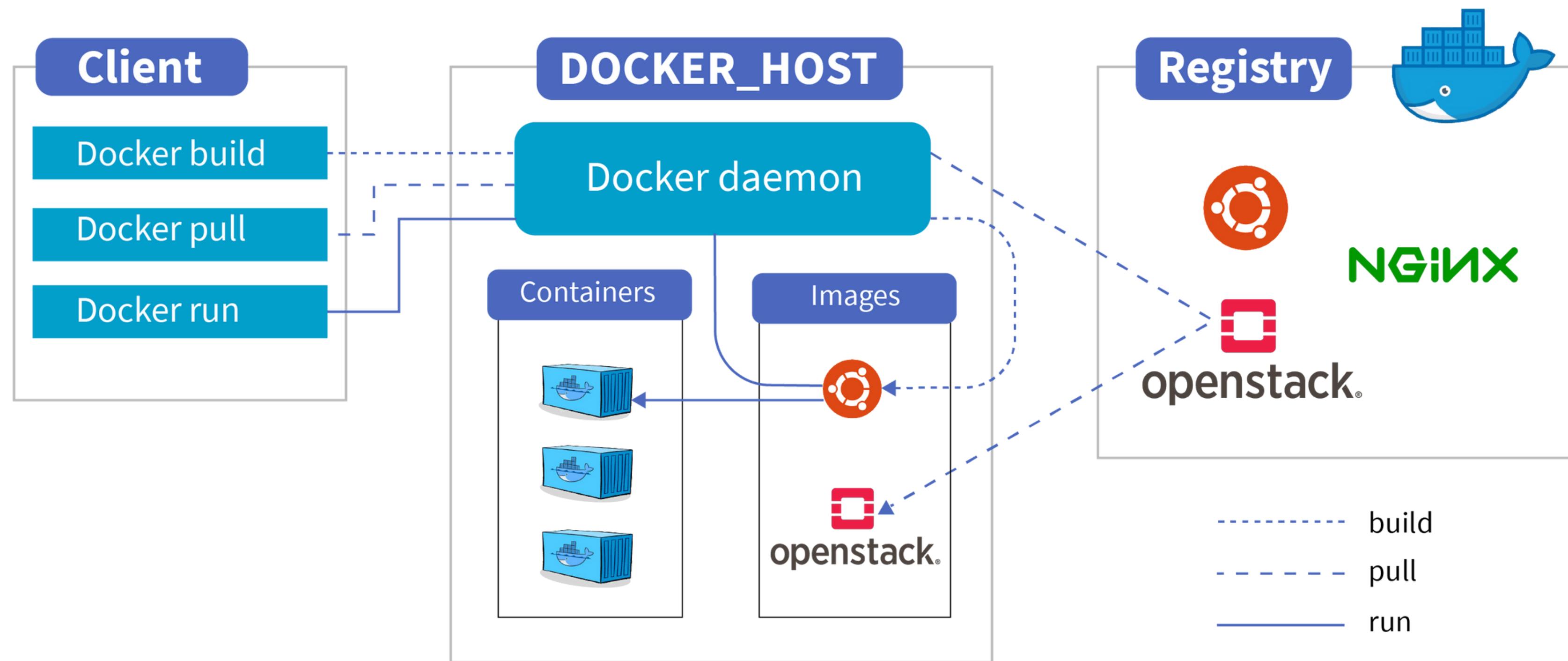
~ Menggunakan isolasi lengkap, termasuk kernel dan sistem operasi yang terpisah.

~ Memerlukan konfigurasi dan manajemen yang lebih rumit.

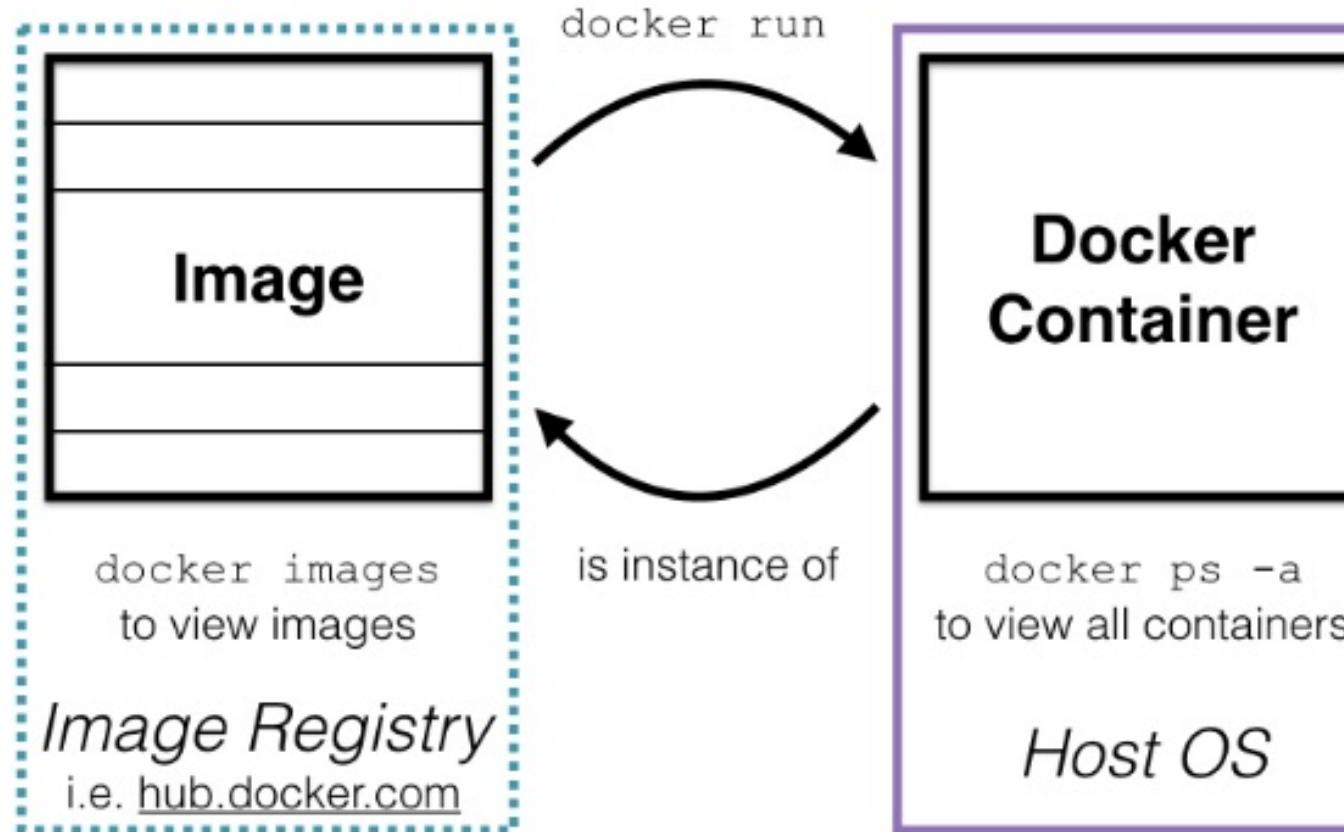
~ Memerlukan alokasi sumber daya yang lebih besar karena setiap VM memiliki kernel dan sistem operasi sendiri.



DOCKER ARCHITECTURE



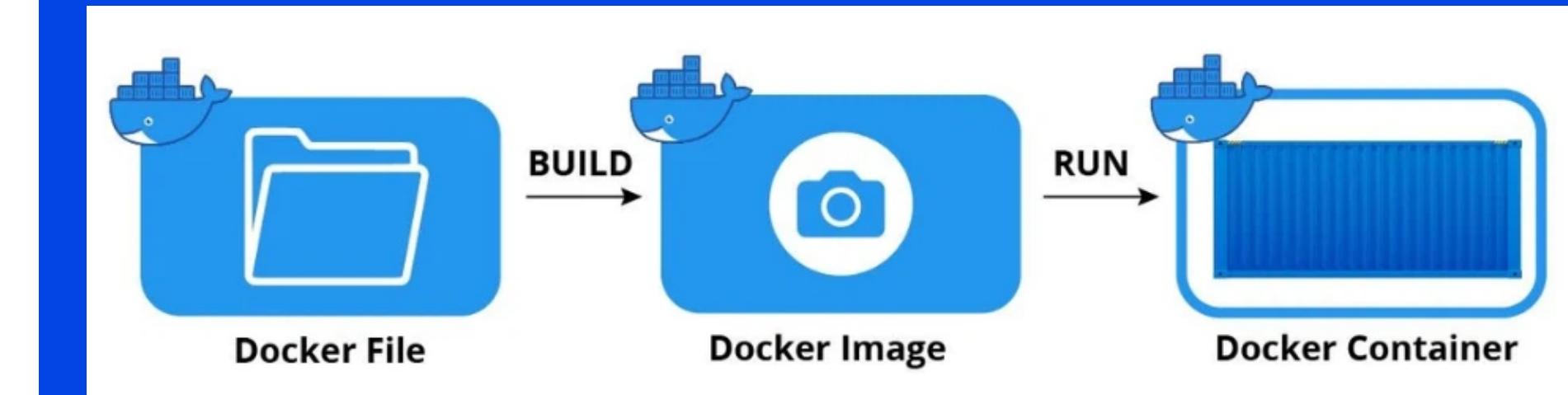
DOCKER BASIC COMMAND



contoh

```
afroh@INBookX1:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world latest d2c94e258dcb 11 months ago 13.3kB
afroh@INBookX1:~$ docker container ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
d12760d42ae8 hello-world "/hello" 2 minutes ago Exited (0) 2 minutes ago modest_chatelet
07782b0174b7 hello-world "/hello" 4 hours ago Exited (0) 4 hours ago wonderful_thompson
```

- **docker version**
- **docker pull**
- **docker run**
- **docker ps**
- **docker exec**
- **docker stop**
- **docker restart**
- **docker kill**
- **docker commit**
- **docker push**



File teks yang berisi serangkaian instruksi (FROM, RUN, COPY, dll) untuk membangun sebuah Docker Image.

Template yang berisi semua informasi yang diperlukan untuk menjalankan sebuah kontainer Docker.

Instans dari sebuah Docker Image yang sedang berjalan, berisi aplikasi dan dependensinya, menggunakan perintah docker run / docker start.

DOCKER II

Business
Process
Automation

DOCKER CLIENT

Docker Client adalah perangkat lunak yang memungkinkan pengguna untuk berinteraksi dengan Docker. Ini adalah antarmuka baris perintah yang memungkinkan Anda untuk menjalankan perintah Docker untuk mengelola container, gambar, jaringan, dan volume.

Beberapa fungsi utama dari Docker Client.

- 1. Menjalankan Perintah Docker:** Docker Client digunakan untuk menjalankan perintah Docker seperti docker run, docker build, docker pull, docker push, docker stop, docker start, dan lainnya.
- 2. Komunikasi dengan Docker Daemon:** Docker Client berkomunikasi dengan Docker Daemon (server Docker) untuk menjalankan perintah yang diberikan oleh pengguna.
- 3. Konfigurasi dan Pengaturan:** Melalui Docker Client, pengguna dapat mengkonfigurasi pengaturan Docker, seperti pengaturan jaringan, volume, dan konfigurasi container.



DOCKER RUN

docker run -it ubuntu bash

docker run hello-world

```
Ubuntu 18.04.6 LTS
n denied
docker: Cannot connect to the Docker daemon at tcp://localhost:2375. Is the docker daemon running?.
See 'docker run --help'.
singgih@SINGGIH:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:53641cd209a4fecfc68e21a99871ce8c6920b2e7502df0a20671c6fccc73a7c6
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

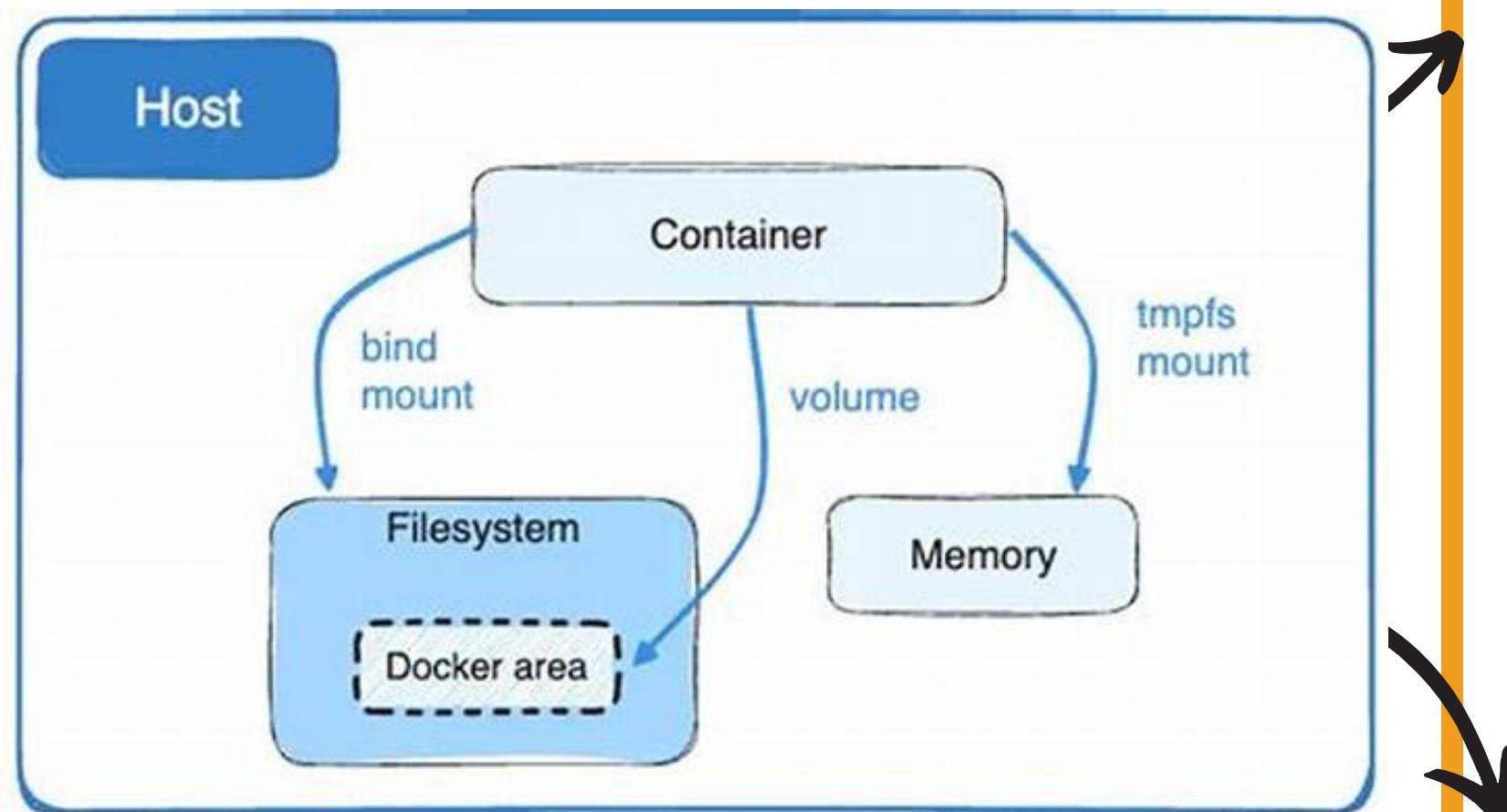
For more examples and ideas, visit:
```

```
root@singgih:~$ docker run -it ubuntu bash
WARNING: Error loading config file: /home/singgih/.docker/config.json: open /home/singgih/.docker/config.json: permission denied
docker: Cannot connect to the Docker daemon at tcp://localhost:2375. Is the docker daemon running?.
See 'docker run --help'.
singgih@SINGGIH:~$ sudo docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
bccd10f490ab: Pull complete
Digest: sha256:77906da86b60585ce12215807090eb327e7386c8fafb5402369e421f44eff17e
Status: Downloaded newer image for ubuntu:latest
root@98115db5a0cc:/#
```

DOCKER VOLUME

- **Data Persisten:** Volume Docker digunakan untuk menyimpan data persisten yang terkait dengan container. Misalnya, jika sebuah container menjalankan database, Anda dapat menggunakan volume Docker untuk menyimpan data database, sehingga data tersebut tidak hilang saat container dihapus atau diperbarui.
- **Independen dari Container:** Volume Docker dapat di-attach dan di-detach dari container tanpa mengganggu data yang disimpan di dalamnya. Ini berarti Anda dapat dengan mudah menghubungkan volume yang sama ke beberapa container atau menghapus container tanpa kehilangan data yang berada di volume.
- **Jenis Volume:** Ada beberapa jenis volume yang dapat digunakan dalam Docker, seperti:
 - **Volume Dikaitkan (Host-mounted Volume):** Memiliki lokasi di host yang spesifik dan di-mount ke dalam container.
 - **Volume Dalam Container (Container-internal Volume):** Volume yang dihasilkan dan diatur oleh Docker secara internal.

DOCKER VOLUME



- Volume Docker direpresentasikan sebagai bagian dalam area Docker, yang terhubung baik ke sistem host maupun sistem file kontainer. Hal ini menunjukkan bahwa volume Docker memberikan cara untuk berbagi data antara host dan kontainer, dan bahwa volume tersebut dikelola secara terpisah dari kontainer itu sendiri.

DOCKER FILE

Deployment Dockerfile adalah sebuah file teks yang digunakan untuk mendefinisikan langkah-langkah atau perintah yang diperlukan untuk membangun sebuah Docker image secara otomatis. Dockerfile berisi serangkaian perintah yang akan dieksekusi secara berurutan oleh Docker Engine saat membangun image.

Shell Script Programming: Dockerfile mirip dengan shell script karena berisi serangkaian perintah yang dieksekusi secara berurutan. Anda dapat menggunakan perintah seperti RUN untuk menjalankan perintah di dalam image Docker, mirip dengan menjalankan perintah shell di lingkungan Linux.

User Environment: Dockerfile digunakan untuk mendefinisikan lingkungan (environment) di dalam image Docker. Anda dapat menentukan variabel lingkungan (ENV) atau pengaturan spesifik lainnya untuk menciptakan kondisi kerja yang sesuai dengan kebutuhan aplikasi atau layanan yang akan dijalankan dalam kontainer Docker.



DOCKER FILE

```
FROM example/rails

MAINTAINER example@example@gmail.com

# Add here your preinstall lib(e.g. imagemagick, mysql lib, ssh config)
## Install imagemagick
RUN apt-get update
RUN apt-get -qq -y install libmagickwand-dev imagemagick

## Install for mysql gem
RUN apt-get install -qq -y mysql-server mysql-client libmysqlclient-dev

## Install for Webshots
RUN apt-get install libssl1.0.2 -y
RUN apt-get install ttf-unfonts-core -y

#{(required) Install Rails App
ADD Gemfile /app/Gemfile
ADD Gemfile.lock /app/Gemfile.lock
RUN bundle install --without development test
ADD . /app

# Overwrite unicorn
ADD config/unicorn.rb /app/config/unicorn.rb

FROM example/rails

MAINTAINER example@example@gmail.com

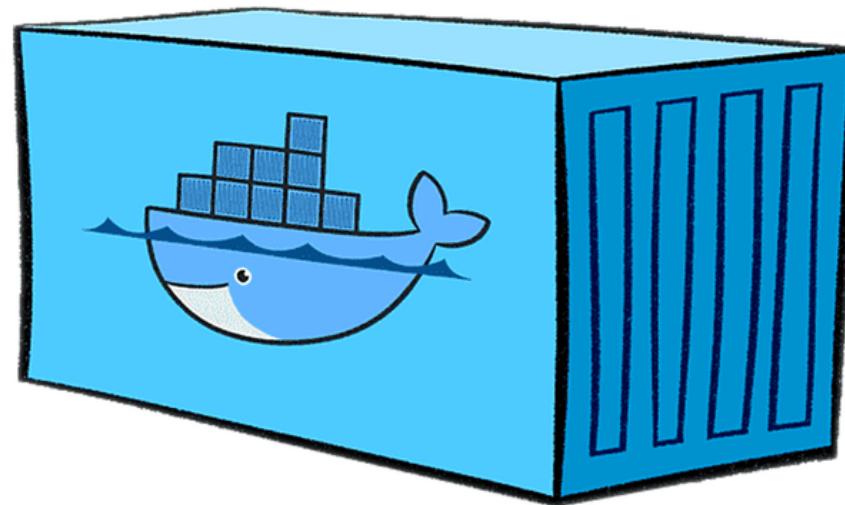
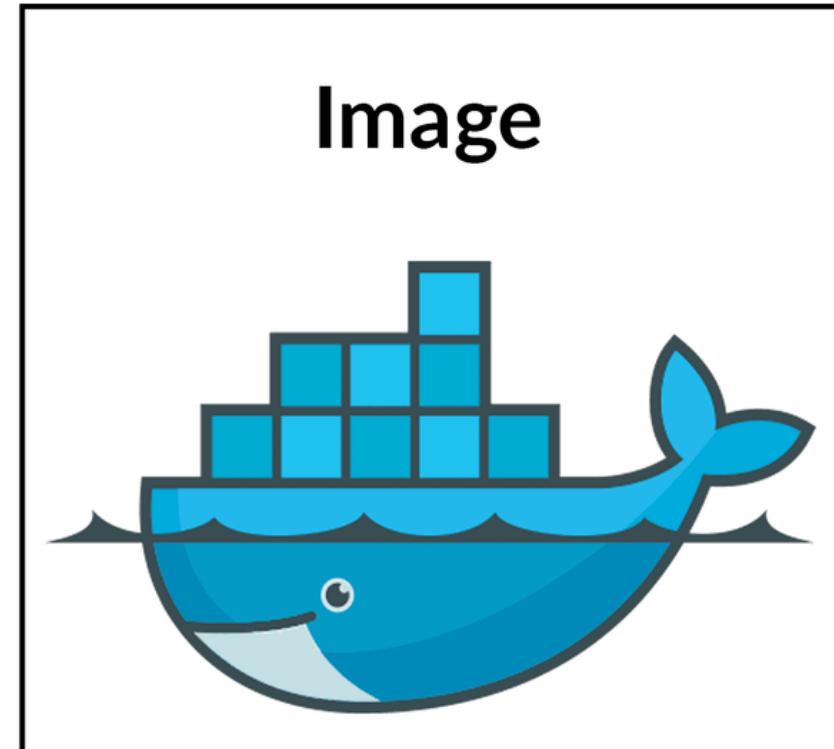
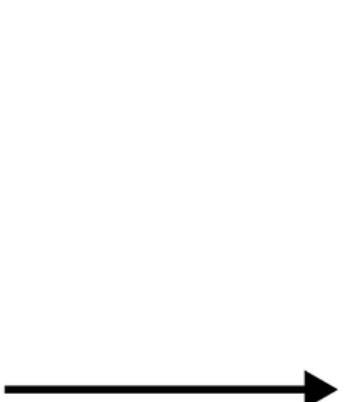
# Add here your preinstall lib(e.g. imagemagick, mysql lib, ssh config)
## Install imagemagick
RUN apt-get update
RUN apt-get -qq -y install libmagickwand-dev imagemagick

## Install for mysql gem
RUN apt-get install -qq -y mysql-server mysql-client libmysqlclient-dev

## Install for Webshots
RUN apt-get install libssl1.0.2 -y
RUN apt-get install ttf-unfonts-core -y

#{(required) Install Rails App
ADD Gemfile /app/Gemfile
ADD Gemfile.lock /app/Gemfile.lock
RUN bundle install --without development test
ADD . /app

# Overwrite unicorn
ADD config/unicorn.rb /app/config/unicorn.rb
```



Dockerfile

Docker Image

Docker Container

DOCKER FILE

- Pada Dockerfile khusus ini, base image diatur menjadi example/rails. Penjaga (maintainer) diatur menjadi example@example@gmail.com. Kemudian, Dockerfile melanjutkan dengan menginstal berbagai pustaka yang diperlukan untuk aplikasi Rails, seperti imagemagick, mysql, dan libssl0.9.8.
- Setelah menginstal pustaka-pustaka yang diperlukan, Dockerfile menyalin Gemfile dan Gemfile.lock dari direktori saat ini ke direktori /app dalam image, dan menjalankan perintah bundle install --without development test untuk menginstal pustaka-pustaka yang dibutuhkan.
- Selanjutnya, Dockerfile menyalin sisa file-file dari direktori saat ini ke direktori /app dalam image. Kemudian, file unicorn.rb dalam image ditimpa dengan file unicorn.rb dari direktori saat ini.

DOCKER COMPOSE

Build
Deploy
Run

DOCKER COMPOSE

Docker Compose adalah tool untuk menjalankan multi-container applications di Docker. Application's services dapat dikonfigurasikan dengan sebuah YAML file.

Terdapat beberapa kasus di mana docker harus menjalankan beberapa kontainer. Docker compose memudahkan pengembang untuk membuat, menghubungkan, dan meluncurkan semua container dengan hanya menjalankan satu perintah.

MANFAAT

- Mengelola container dengan mudah dan cepat
- Memudahkan konfigurasi
- Menghemat banyak waktu



DOCKER COMPOSE VS DOCKER RUN

Contoh aplikasi WordPress yang terdiri dari layanan WordPress yang berkomunikasi dengan database MySQL.

```
docker run -d \
--name db \
--restart always \
-v db_data:/var/lib/mysql \
-e MYSQL_ROOT_PASSWORD=supersecret \
-e MYSQL_DATABASE=exampledb \
-e MYSQL_USER=exampleuser \
-e MYSQL_PASSWORD=examplepass \
mysql:5.7
```

Menjalankan Aplikasi docker dengan banyak container tanpa docker compose.

Menjalankan aplikasi docker dengan banyak container menggunakan docker compose (file yml).

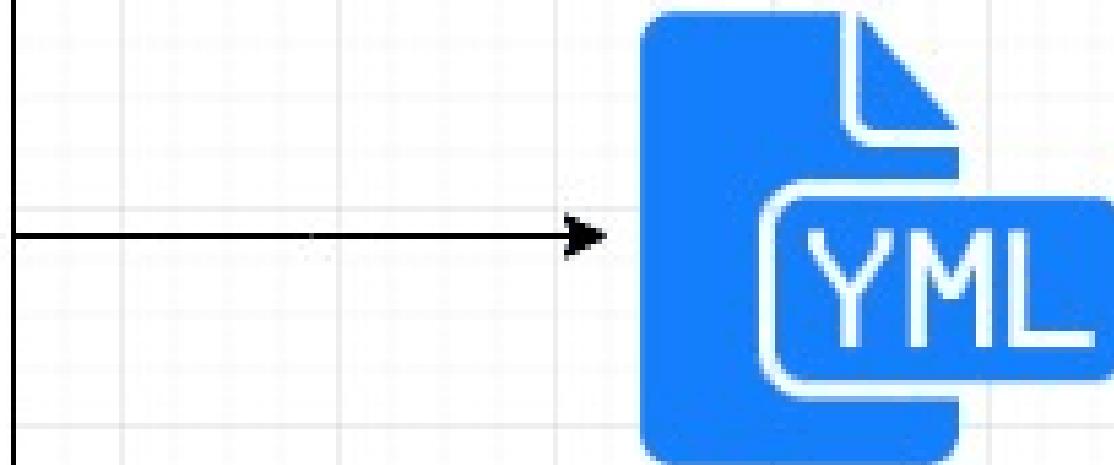
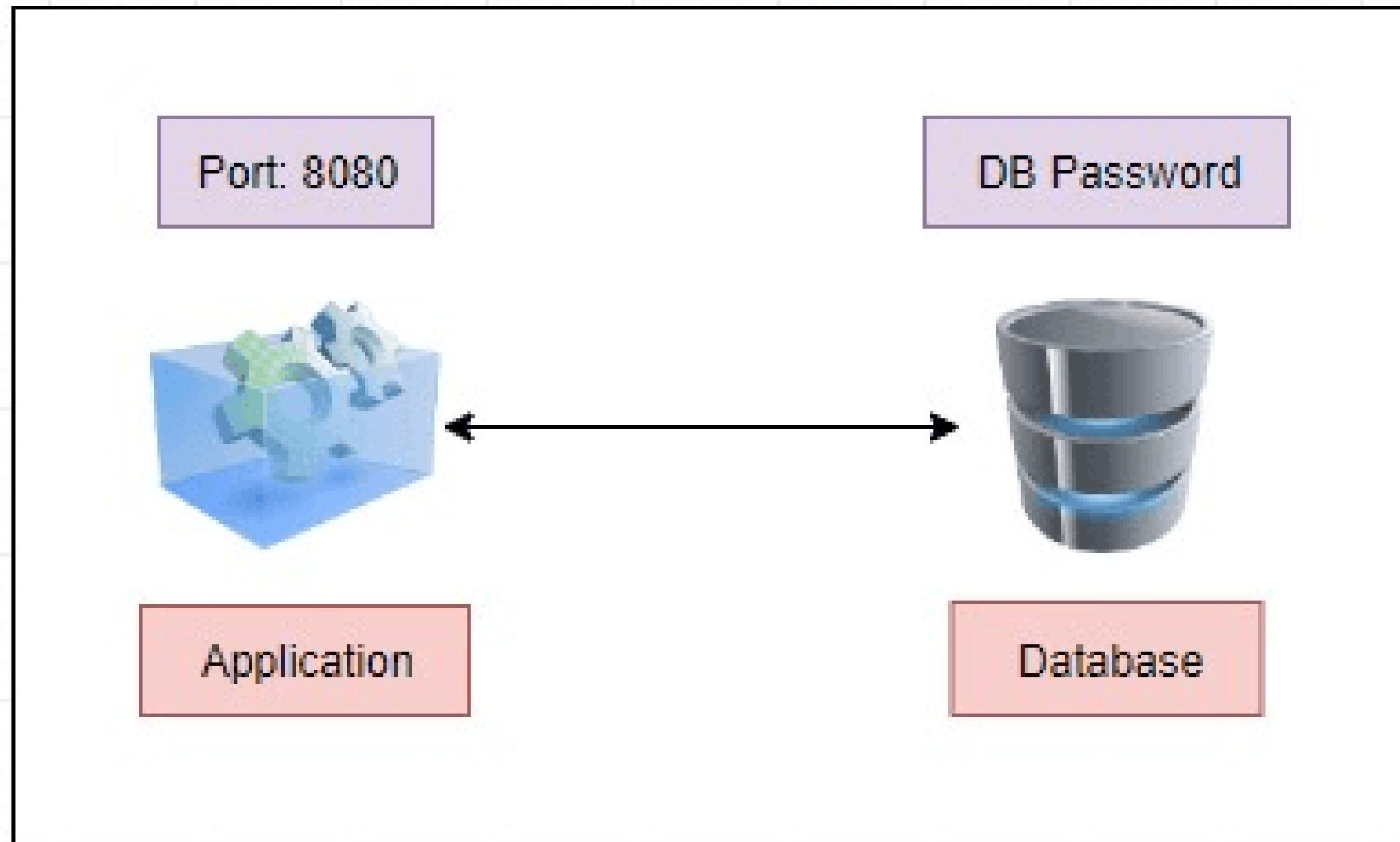
```
version: '3.9'

services:
  db:
    image: mysql:5.7
    restart: always
    environment:
      MYSQL_DATABASE: exampledb
      MYSQL_USER: exampleuser
      MYSQL_PASSWORD: examplepass
      MYSQL_ROOT_PASSWORD: supersecret
    volumes:
      - db_data:/var/lib/mysql
  wordpress:
    image: wordpress
    restart: always
    ports:
      - 8080:80
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: exampleuser
      WORDPRESS_DB_PASSWORD: examplepass
      WORDPRESS_DB_NAME: exampledb
    volumes:
      - wordpress_data:/var/www/html
  volumes:
    wordpress_data:
    db_data:
```

HOW DOCKER COMPOSE WORKS

in nutshell

docker-compose



yML FILE STRUCTURE

Menentukan versi sintaks Docker Compose yang digunakan.

Services mendefinisikan layanan yang akan berjalan pada kontainer.

Volumes digunakan untuk mendefinisikan volume yang akan digunakan oleh layanan.

Networks memungkinkan untuk mendefinisikan dan menghubungkan layanan ke jaringan khusus.

- Version
- Services
 - Build
 - Image
 - Environment
 - Ports
 - Volumes
- Volumes
- Networks



— COMMANDS

syntax

```
docker-compose [-f <arg>...] [options] [COMMAND] [ARGS...]
```

Untuk memulai Docker Compose

```
docker-compose up
```

```
docker-compose start
```

Jika file kita memiliki nama yang berbeda dari default (`docker-compose.yml`), kita bisa menggunakan command ini:

```
docker-compose -f custom-compose-file.yml start
```



— COMMANDS

Compose dapat berjalan di latar belakang sebagai daemon saat diluncurkan dengan opsi -d:

```
docker-compose up -d
```

Untuk menghentikan layanan dengan aman sambil mempertahankan semua modifikasi, gunakan perintah stop.

```
docker-compose stop
```

Untuk mereset proyek, jalankan perintah down yang akan menghapus semua kecuali volume eksternal.

```
docker-compose down
```

KUBERNETES I

Introducción



kubernetes

Kubernetes atau K8s adalah platform open-source yang digunakan untuk otomatisasi deployment, skalabilitas, dan manajemen aplikasi kontainer. Dirancang Oleh Google, Kubernetes menyediakan lingkungan yang kuat dan fleksibel untuk mengelola aplikasi yang berjalan dalam wadah (container) di berbagai lingkungan, mulai dari pusat data lokal hingga cloud publik.

KUBERNETES
APALIKA
TUNGAN
JSETE
?SERTA
APADA



kubernetes

- Kubernetes berasal dari bahasa Yunani yang berarti "juru mudi", pemandu Anda melewati perairan yang tidak dikenal.
- Kubernetes adalah kernel linux untuk sistem terdistribusi.
- Kubernetes adalah linux-nya awan!
- Kubernetes adalah sebuah platform dan alat orkestrasi kontainer untuk mengotomatisasi penyebaran, penskalaan, dan pengoperasian kontainer aplikasi.
- Kubernetes mendukung, Containerd, CRI-O, kontainer Kata (sebelumnya clear dan hyper) dan Virtlet.

MENGAPA KUBERNETES PENTING?

Skalabilitas

Kubernetes memungkinkan pengguna untuk mengelola aplikasi yang terdiri dari ratusan atau bahkan ribuan kontainer dengan mudah.

Resilience

Kubernetes secara otomatis mengelola pemulihan dan replikasi jika salah satu komponen aplikasi gagal.

Portabilitas

Dengan kubernetes pengguna dapat menjalankan aplikasinya di berbagai lingkungan, baik itu di lingkungan lokal, pusat data di lokasi, atau di cloud publik.

Otomatisasi

Dengan konsep seperti auto-scaling, Kubernetes memungkinkan otomatisasi yang luas dalam manajemen aplikasi.



KONSEP DASAR KUBERNETES



kubernetes

01

Node

Node adalah mesin fisik atau virtual yang menjalankan aplikasi dan layanan dalam lingkungan Kubernetes. Setiap node memiliki agen Kubernetes (kubelet) yang bertanggung jawab untuk berkomunikasi dengan kontrol plane Kubernetes dan menjalankan tugas-tugas yang ditetapkan padanya.

02

Pod

Pod adalah unit dasar dalam Kubernetes yang terdiri dari satu atau beberapa kontainer yang saling berbagi sumber daya dan jaringan. Kontainer dalam satu pod selalu berjalan di node yang sama dan memiliki lingkungan yang sama.

03

Kontrol Plane

Kontrol plane adalah komponen inti dari Kubernetes yang bertanggung jawab untuk mengatur dan mengontrol kluster Kubernetes. Ini terdiri dari beberapa komponen.

04

Service

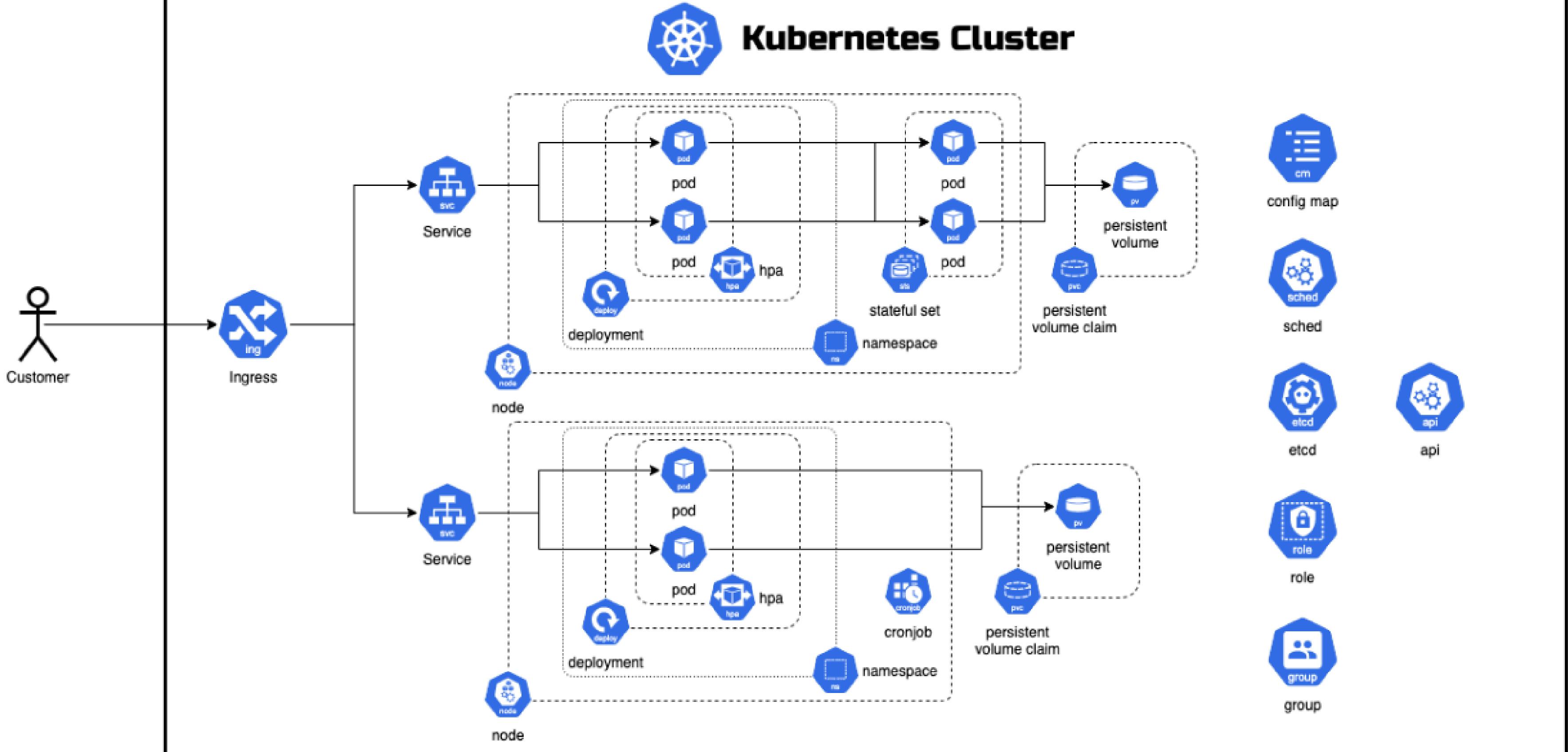
Service adalah cara untuk mengekspos aplikasi yang berjalan di Kubernetes ke luar dunia. Layanan memberikan alamat IP yang stabil dan nama DNS yang terhubung ke satu atau beberapa pod, memungkinkan aplikasi lain atau pengguna untuk berkomunikasi dengan aplikasi yang berjalan di kluster.



MEMAHAMI KONSEP DASAR KUBERNETES

Di Kubernetes, ada sebuah node master dan beberapa node pekerja, setiap node pekerja dapat menangani beberapa pod. Pod hanyalah sekumpulan kontainer yang dikelompokkan bersama sebagai unit kerja. Anda dapat mulai mendesain aplikasi Anda menggunakan pod. Setelah pods Anda siap, Anda dapat menentukan definisi pod ke node master, dan berapa banyak yang ingin Anda terapkan. Dari titik ini, Kubernetes memegang kendali. Ia mengambil pod dan menyebarkannya ke nodus pekerja.

Kubernetes Cluster



PENGELOLAAN APLIKASI DENGAN KUBERNETES

Deployment

Deployment adalah objek Kubernetes yang digunakan untuk mendefinisikan dan mengelola aplikasi dalam kluster. Ini menyediakan cara untuk mendefinisikan jumlah replika dari pod, strategi rolling update, dan konfigurasi lainnya.

StatefulSet

StatefulSet adalah jenis objek Kubernetes yang dirancang untuk mengelola aplikasi yang membutuhkan penyimpanan persisten dan identitas yang stabil. Contoh aplikasi yang cocok untuk StatefulSet termasuk database dan sistem manajemen pesan.

DaemonSet

DaemonSet memastikan bahwa setiap node dalam kluster memiliki sebuah pod yang menjalankan instance dari aplikasi tertentu. Ini berguna untuk menjalankan layanan yang perlu berjalan pada setiap node, seperti pengumpul log atau agen pemantauan.

MENGENAL KIND: ALAT UNTUK MEMBUAT KLUSTER KUBERNETES LOKAL



Apa itu Kind?

Kind (Kubernetes IN Docker) adalah alat open-source yang memungkinkan pengguna untuk membuat kluster Kubernetes lokal menggunakan Docker sebagai backend. Dengan Kind, pengguna dapat dengan mudah membuat kluster Kubernetes untuk pengembangan, pengujian, dan eksperimen, tanpa perlu mengkonfigurasi infrastruktur yang rumit.

Berikut adalah langkah-langkah dasar untuk menggunakan Kind:

- 1. Instalasi Kind:** Instal Kind dengan mengunduh binary atau menggunakan manajer paket yang tersedia.
- 2. Membuat Kluster:** Gunakan perintah `kind create cluster` untuk membuat kluster Kubernetes baru (dengan 1 node), perintah `kind create cluster --config kind-config.yaml` untuk membuat kluster Kubernetes baru (dengan konfigurasi kustom).
- 3. Mengelola Kluster:** Setelah kluster dibuat, dapat menggunakan perintah `kubectl` untuk berinteraksi dengan kluster seperti biasa.
- 4. Menghapus Kluster:** Gunakan perintah `kind delete cluster` untuk menghapus kluster setelah selesai digunakan.

MENGENAL KUBECTL: ALAT COMMAND-LINE UNTUK MENGELOLA KUBERNETES



kubectl

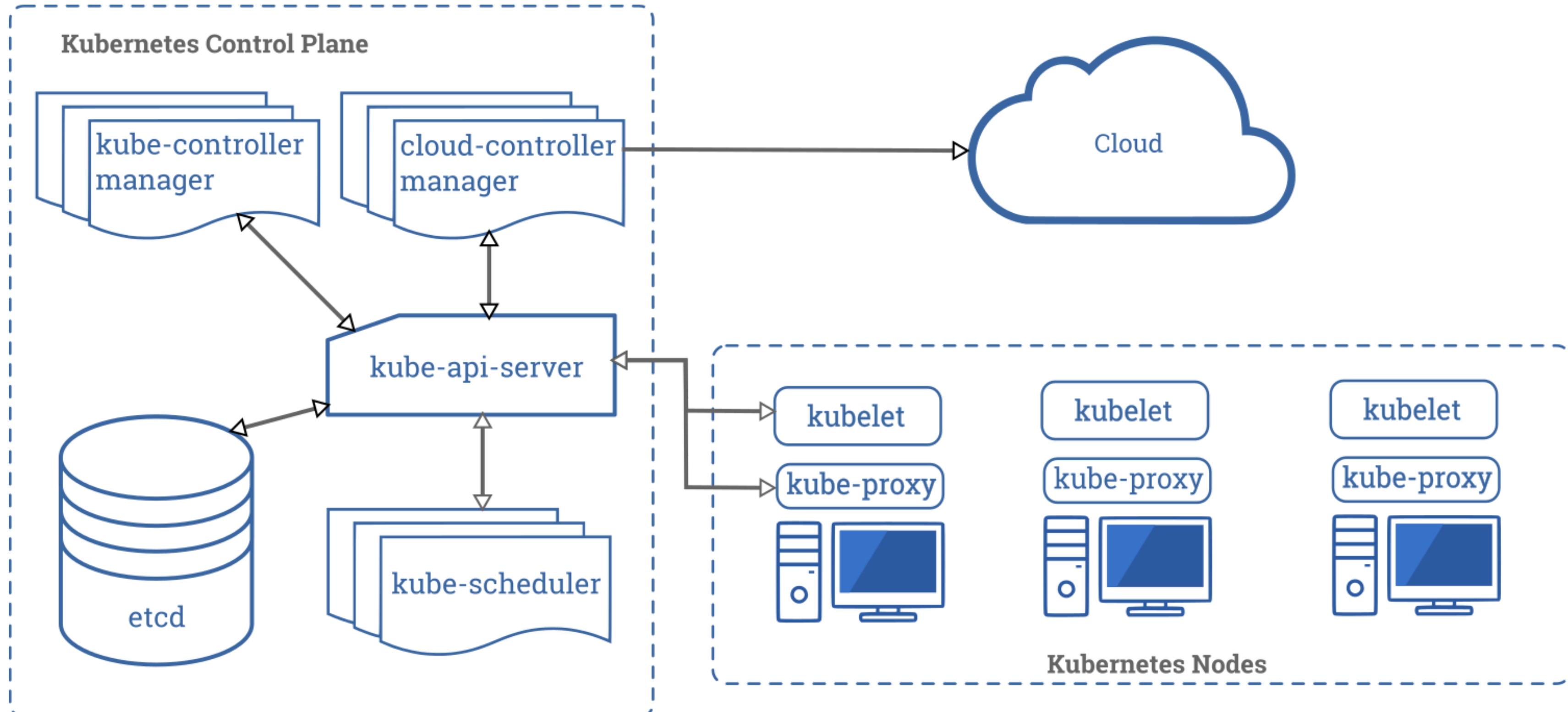
Apa itu Kubectl?

Kubectl adalah alat command-line utama yang digunakan untuk berinteraksi dengan kluster Kubernetes. Dengan kubectl, pengguna dapat membuat, mengedit, dan mengelola sumber daya Kubernetes seperti pod, layanan, deployment, dan banyak lagi, langsung dari terminal.

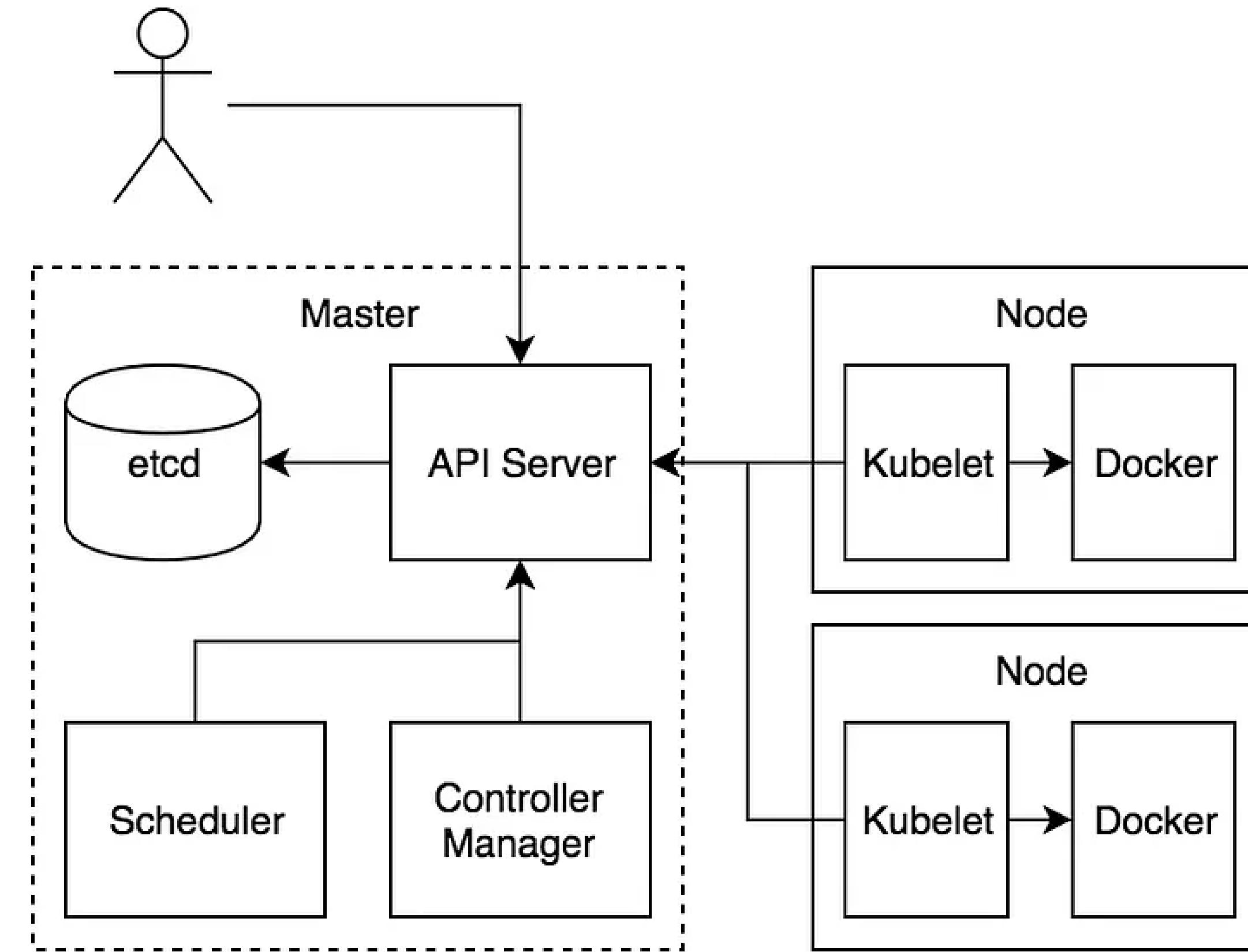
Beberapa perintah utama kubectl beserta fungsinya:

1. ***kubectl create***: Membuat sumber daya Kubernetes dari file konfigurasi atau langsung dari baris perintah.
2. ***kubectl get***: Mengambil informasi tentang sumber daya Kubernetes yang ada di dalam kluster, seperti pod, layanan, dan deployment.
3. ***kubectl describe***: Menampilkan detail lengkap tentang sumber daya Kubernetes tertentu, termasuk konfigurasi dan status saat ini.
4. ***kubectl apply***: Menggunakan konfigurasi yang diberikan untuk membuat atau memperbarui sumber daya Kubernetes, dengan mendeteksi perubahan yang perlu dilakukan.
5. ***kubectl delete***: Menghapus sumber daya Kubernetes yang telah dibuat sebelumnya dari kluster.

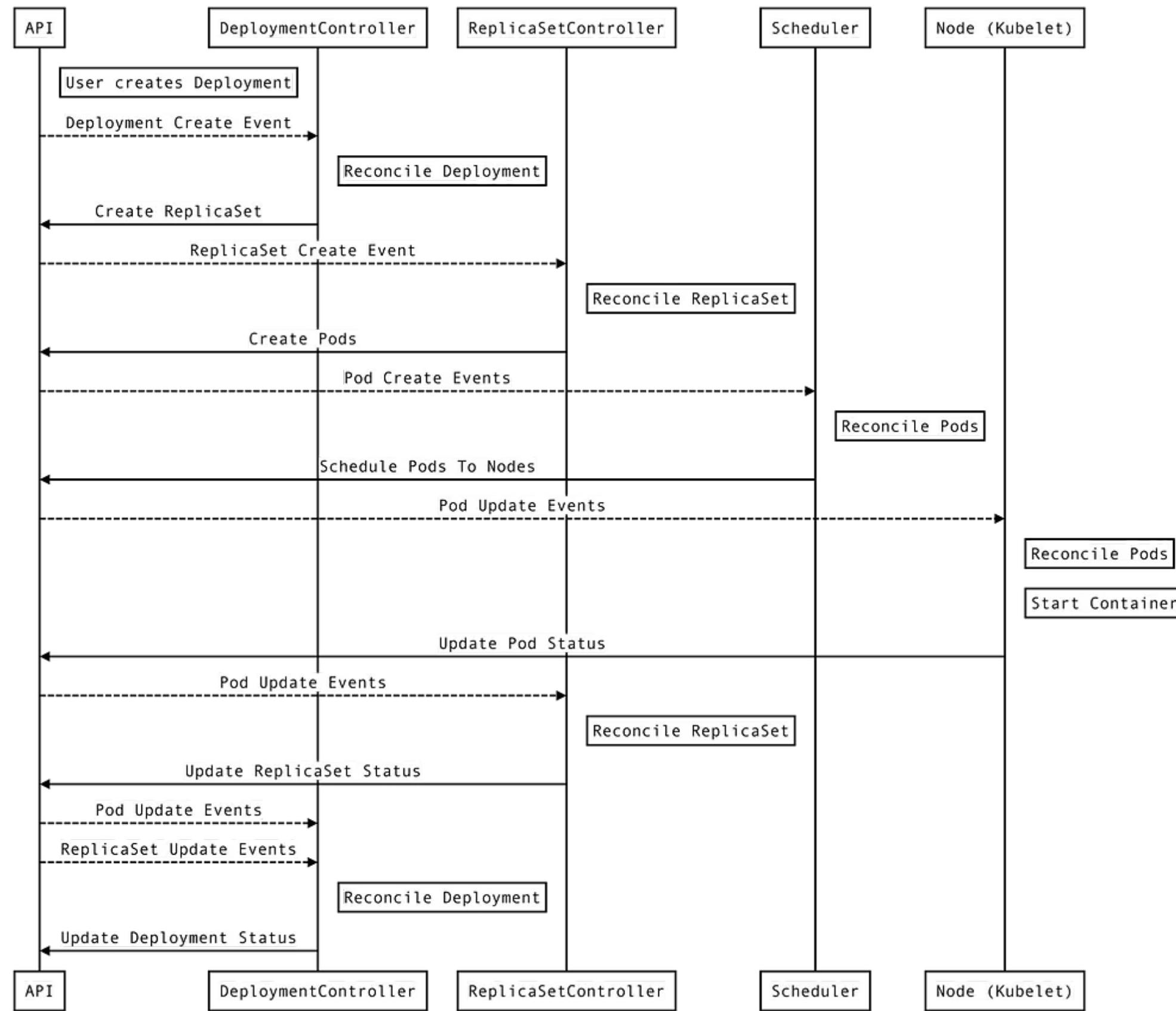
KUBERNETES COMPONENTS



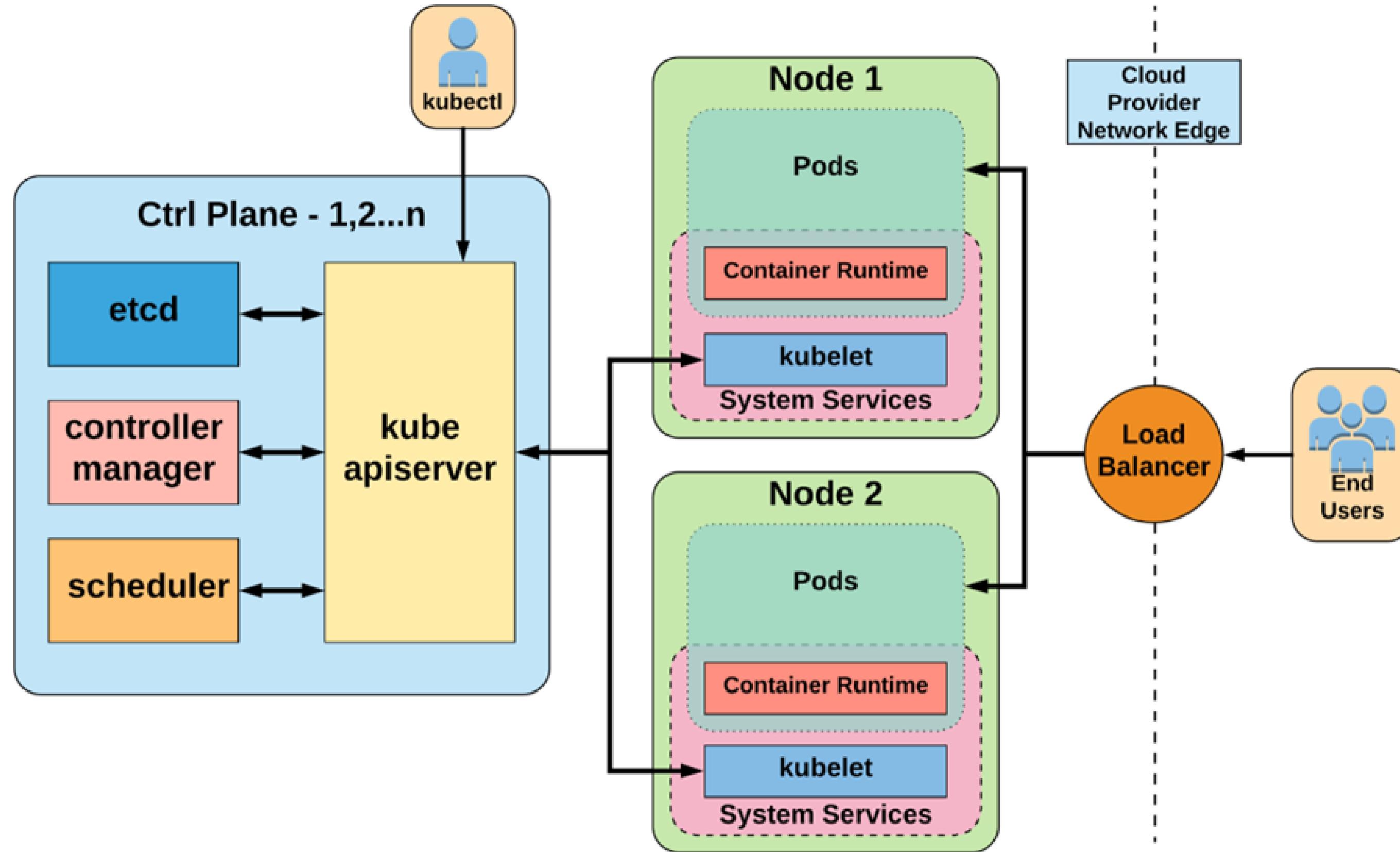
A TYPICAL FLOW: HOW K8S API WORKS



KUBERNETES COMPONENTS FLOW



KUBERNETES ARCHITECTURE OVERVIEW



THANK YOU

by DREAM