



کوبرنتیز

پارسا محمدپور

۲۰ دی ۱۴۰۳

فهرست

• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

- مقدمه
- مقایسه با ماشین مجازی

• داکر

- مقدمات
- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری
- دستورات

• تبدیل برنامه به کانتینر

• کوبرنتیز

- مقدمه
- اجزای اصلی سازنده
- مفاهیم مقدماتی
- امنیت
- معرفی چند ابزار
- فراهم‌کننده سرویس کوبر در بستر ابری

• منابع

• ماشین‌های مجازی

• مجازی‌سازی

• هایپروایزر

• نحوه کار کردن هایپروایزر

• کانتینر

• مقدمه

• مقایسه با ماشین مجازی

• داکر

• مقدمات

• اعضای سازنده

• اعضای سازنده هسته داکر

• معماری

• دستورات

• تبدیل برنامه به کانتینر

• کوبرنتیز

• مقدمه

• اجزای اصلی سازنده

• مفاهیم مقدماتی

• امنیت

• معرفی چند ابزار

• فراهم‌کننده سرویس کوبر در بستر

ابری

• منابع

ماشین‌های مجازی - مجازی‌سازی

• تعریف مجازی‌سازی

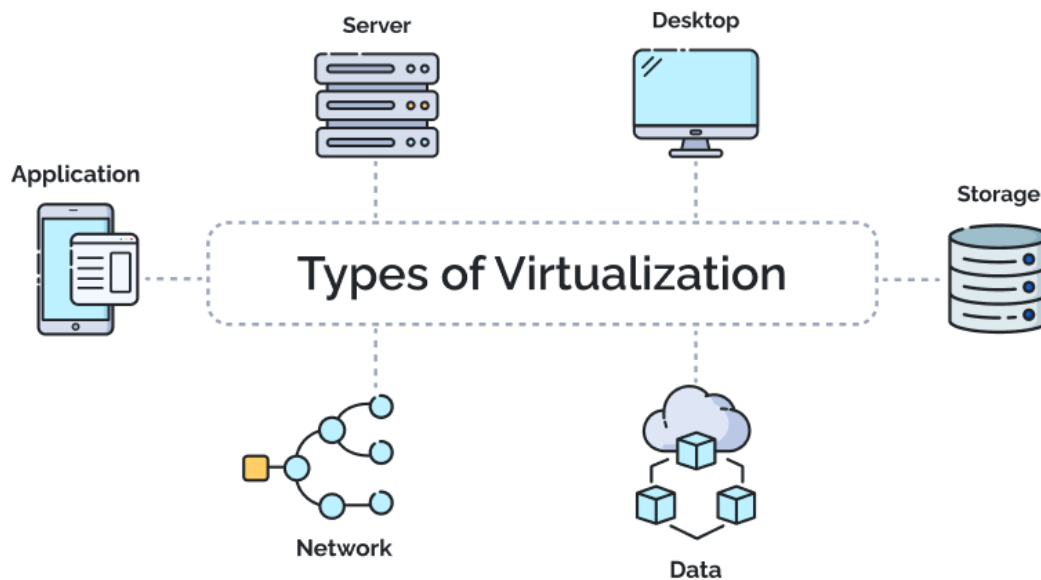
• انواع

• سرور

• شبکه

• حافظه

• دستگاه‌های فیزیکی



• ماشین‌های مجازی

• مجازی‌سازی

• هایپروایزر

• نحوه کار کردن هایپروایزر

• کانتینر

• مقدمه

• مقایسه با ماشین مجازی

• داکر

• مقدمات

• اعضای سازنده

• اعضای سازنده هسته داکر

• معماری

• دستورات

• تبدیل برنامه به کانتینر

• کوبرنتیز

• مقدمه

• اجزای اصلی سازنده

• مفاهیم مقدماتی

• امنیت

• معرفی چند ابزار

• فراهم‌کننده سرویس کوبر در بستر

ابری

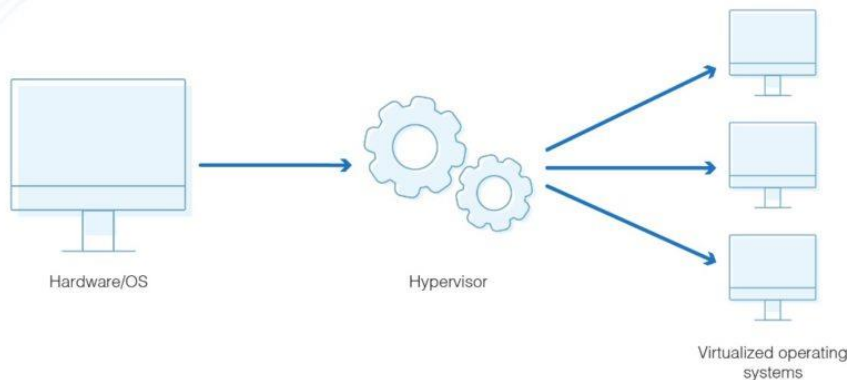
• منابع

ماشین‌های مجازی - هایپروایزر

- هایپروایزر

- نرم‌افزارپ

What Is a Hypervisor?



- مزیت‌های هایپروایزر

- بهینه‌سازی منابع

- به اشتراک گذاشتن محیط دسکتاپ (Desktop environment mirroring)

• ماشین‌های مجازی

- مجازی‌سازی

- هایپروایزر

• نحوه کارکردن هایپروایزر

- کانتینر

- مقدمه

- مقایسه با ماشین مجازی

- داکر

- مقدمات

- اعضای سازنده

- اعضای سازنده هسته داکر

- معماری

- دستورات

- تبدیل برنامه به کانتینر

- کوبرنتیز

- مقدمه

- اجزای اصلی سازنده

- مفاهیم مقدماتی

- امنیت

- معرفی چند ابزار

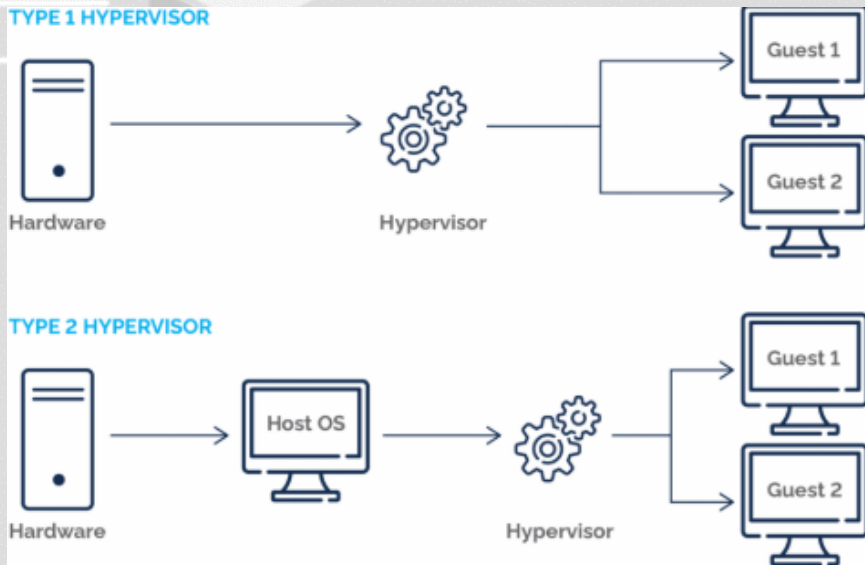
- فراهم‌کننده سرویس کوبر در بستر

- ابری

- منابع

ماشین‌های مجازی - نحوه کارکردن هایپروایزر

- نحوه کارکرد هایپروایزر



- انواع هایپروایزر

- نوع یک (مِتا هایپروایزر)

- VMware hypervisor
- Microsoft Hyper-V
- Oracle VM Server

- نوع دو

- VMware workstation
- VMware fusion
- Oracle VirtualBox

• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

• مقدمه

- مقایسه با ماشین مجازی

• داکر

- مقدمات
- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری
- دستورات

• تبدیل برنامه به کانتینر

• کوبرنتیز

• مقدمه

- اجزای اصلی سازنده

- مفاهیم مقدماتی

- امنیت

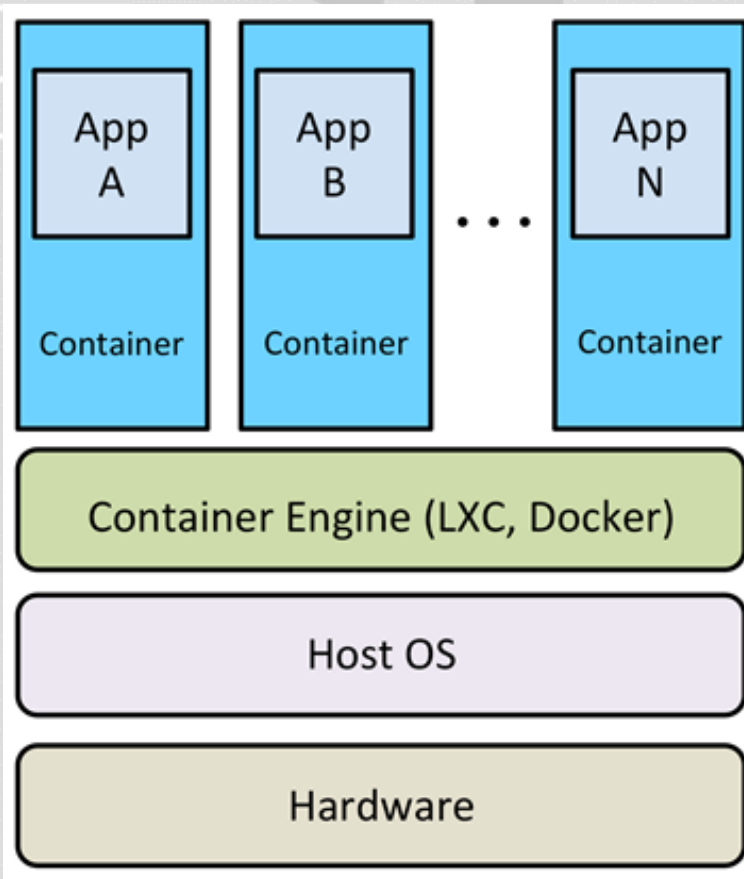
- معرفی چند ابزار

- فراهم‌کننده سرویس کوبر در بستر

ابری

• منابع

کانتینر - مقدمه



• کانتینر

- مستقل
- بسته‌های نرم‌افزاری قابل اجرا
- دارای یک هسته سیستم عامل مانند

• مزایا

- قابل حمل بودن
- سازگار و با ثبات
- کارآمد
- سبک
- چابک و سریع

• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

- مقدمه

• مقایسه با ماشین مجازی

• داکر

- مقدمات
- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری
- دستورات

• تبدیل برنامه به کانتینر

• کوبرنتیز

- مقدمه

• اجزای اصلی سازنده

- مفاهیم مقدماتی

- امنیت

• معرفی چند ابزار

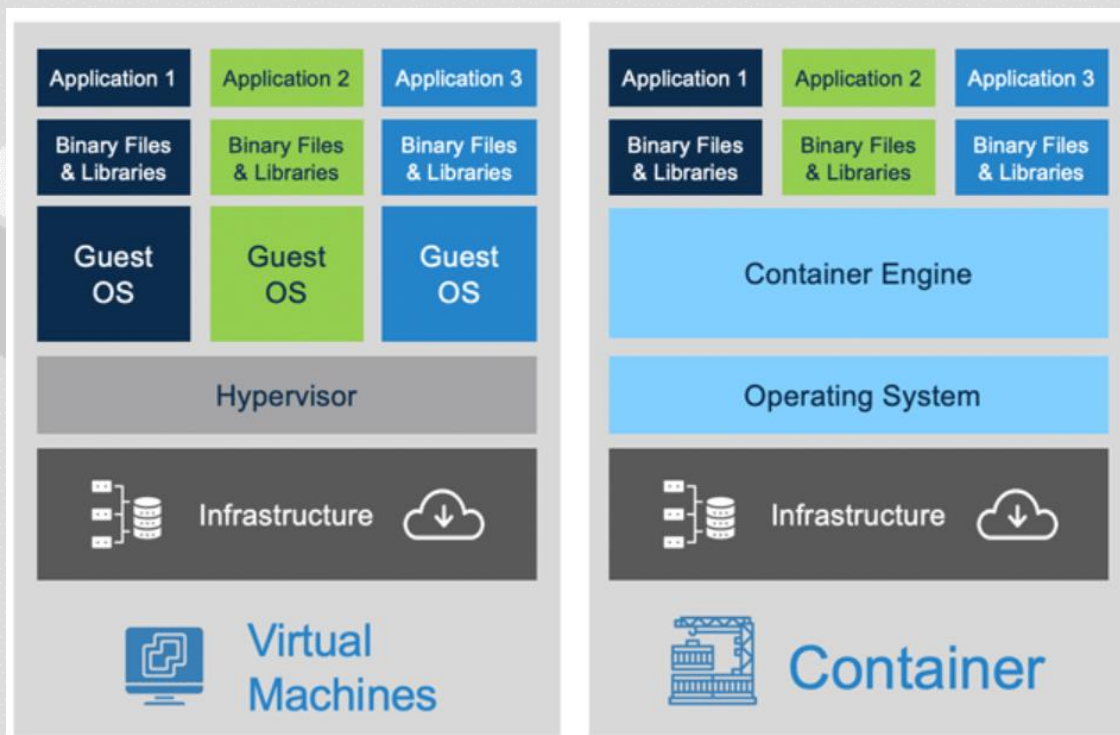
• فراهم‌کننده سرویس کوبر در بستر

ابری

• منابع

کانتینر - مقایسه با ماشین مجازی

- تفاوت بین کانتینر و ماشین مجازی
- سیستم عامل مشترک
- دردسر کمتر (less overhead)



• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

- مقدمه
- مقایسه با ماشین مجازی

• داکر

• مقدمات

- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری
- دستورات

• تبدیل برنامه به کانتینر

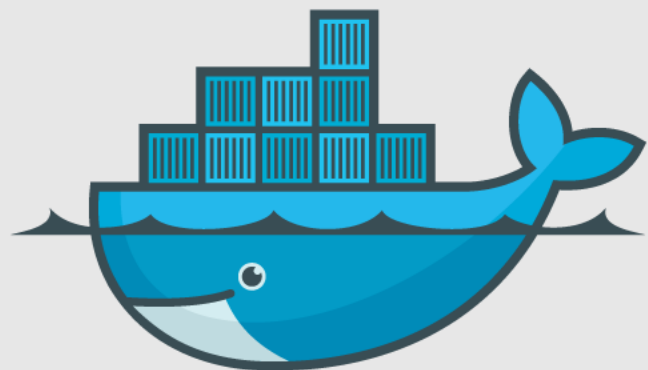
• کوبرنتیز

- مقدمه
- اجزای اصلی سازنده
- مفاهیم مقدماتی
- امنیت
- معرفی چند ابزار
- فراهم‌کننده سرویس کوبر در بستر ابری

• منابع

داگر - مقدمات

- پلتفرم به عنوان سرویس



docker

• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر
- کانتینر

• مقدمه

• مقایسه با ماشین مجازی

• داکر

• مقدمات

• اعضای سازنده

- اعضای سازنده هسته داکر
- معماری
- دستورات

• تبدیل برنامه به کانتینر

• کوبرنتیز

• مقدمه

• اجزای اصلی سازنده

• مفاهیم مقدماتی

• امنیت

• معرفی چند ابزار

• فراهم‌کننده سرویس کوبر در بستر

ابری

• منابع

داکر - اعضای سازنده

- اعضای سازنده اصلی
 - هسته داکر (docker engine)
 - داکر دیمون (docker daemon)
 - ای پی آی های رست (REST API)
 - سوکت UNIX
 - کلاینت یا همان واسط ترمینال (docker CLI) - داکر دسکتاپ
 - کانتینر
 - ایمج
 - شبکه
 - والیوم یا حجم داده ها (Data Volume)

• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

- مقدمه
- مقایسه با ماشین مجازی

• داکر

- مقدمات
- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری
- دستورات

• تبدیل برنامه به کانتینر

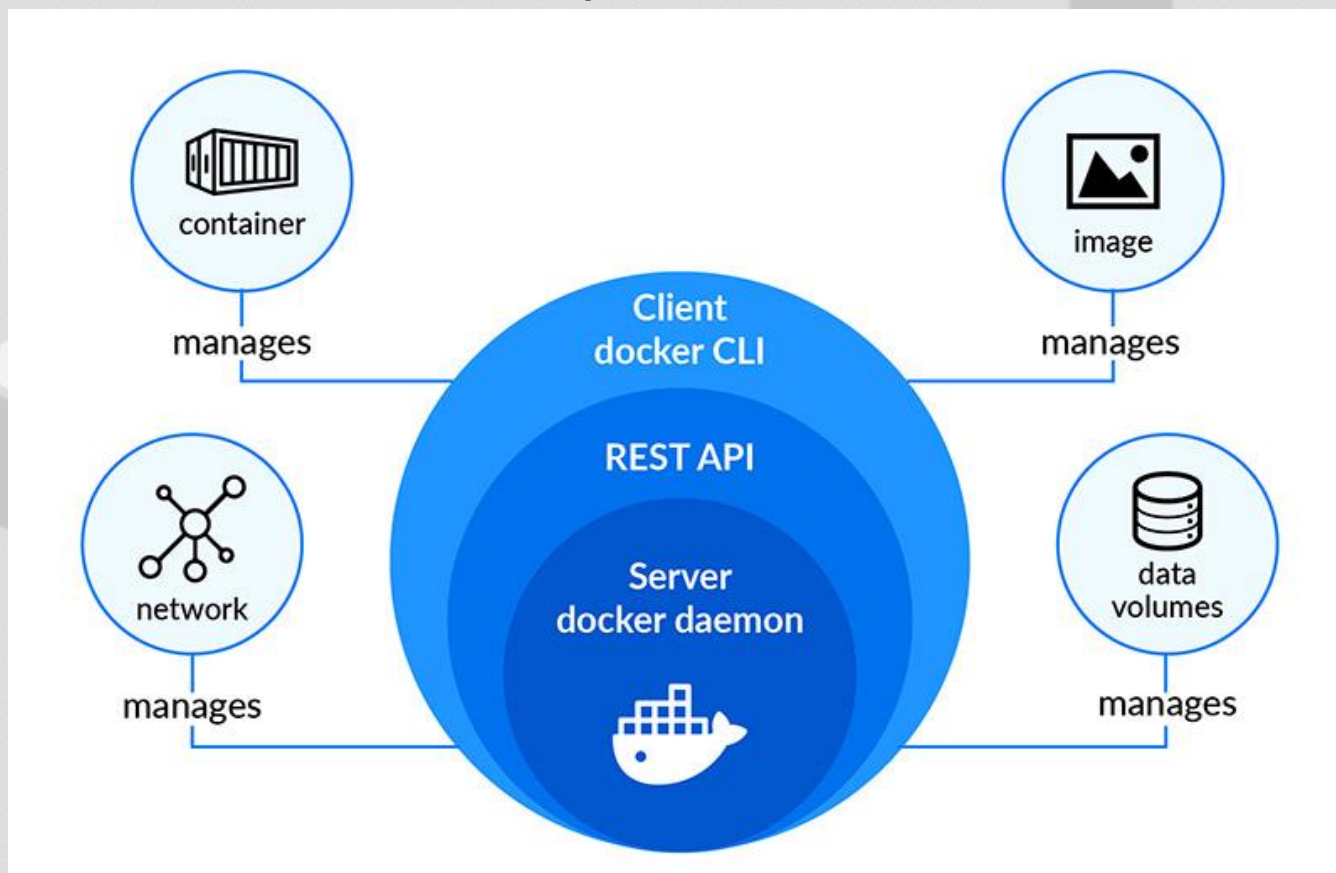
• کوبرنتیز

- مقدمه
- اجزای اصلی سازنده
- مفاهیم مقدماتی
- امنیت
- معرفی چند ابزار
- فراهم‌کننده سرویس کوبر در بستر ابری

• منابع

داکر - اعضای سازنده هسته داکر

هسته داکر



• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

- مقدمه
- مقایسه با ماشین مجازی

• داکر

- مقدمات
- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری
- دستورات

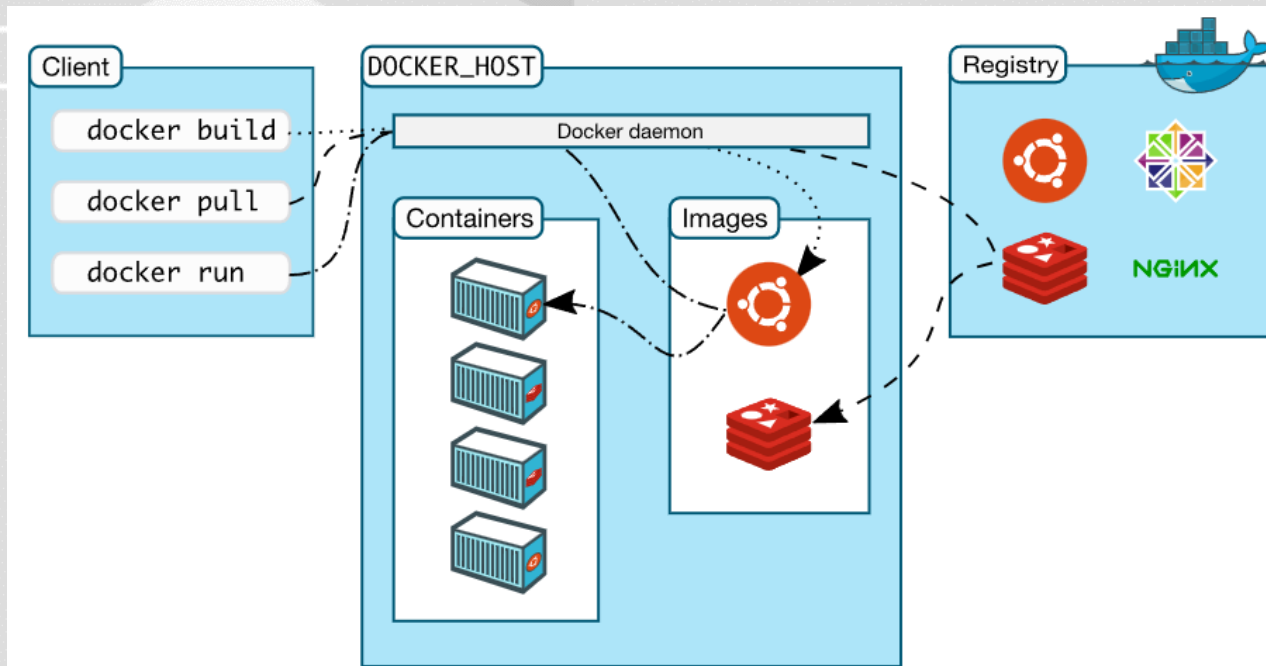
• تبدیل برنامه به کانتینر

• کوبرنتیز

- مقدمه
- اجزای اصلی سازنده
- مفاهیم مقدماتی
- امنیت
- معرفی چند ابزار
- فراهم‌کننده سرویس کوبر در بستر ابری

• منابع

داکر - معماری



- داکر دیمون
- کلاینت داکر
- رجیستری ها
- اشیاء داکر
 - ایمج ها
 - داکر فایل
 - کانٹینر ها
 - سرویس ها
 - شبکه
 - حجم داده ها یا والیوم (Data Volume)
 - پلاگین ها

• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

- مقدمه
- مقایسه با ماشین مجازی

• داکر

- مقدمات
- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری

• دستورات

• تبدیل برنامه به کانتینر

• کوبرنتیز

- مقدمه
- اجزای اصلی سازنده
- مفاهیم مقدماتی
- امنیت
- معرفی چند ابزار
- فراهم‌کننده سرویس کوبر در بستر


ابری

• منابع

داکر - دستورات

- بالا آوردن (شروع) هسته داکر (start)
- گرفتن ایمیج (pull)
- فرستادن ایمیج (push)
- اجرای ایمیج (run)
- ساختن ایمیج (build)
- متوقف کردن کانتینر (stop)
- از بین بردن کانتینر (kill)

داگر - دستورات

| | | | |
|--|--|--|--|
| www.metakoder.com | | | |
| Docker Cheat Sheet @bhanuchddha | | Containers | Networking |
| General | Container Registry | Create and run a container from an image, with a custom name \$ docker run --name <container_name> <image_name> | List all the Networks \$ docker network ls |
| Start the docker daemon \$ docker -d | Login into Docker \$ docker login -u <username> | Run a container in the background \$ docker run -d <image_name> | Create a new Network \$ docker network create --driver <driver-name> <bridge-name> |
| Get help with Docker. Can also use --help on all subcommands \$ docker --help | Publish an image to Docker Hub \$ docker push <username>/<image_name>:<tag> | Run a container and publish a container's port(s) to the host. \$ docker run -p <host_port>:<container_port> <image_name> | Connect a running container to a network \$ docker network connect <network-name> <container-name> |
| Images | Search Hub for an image \$ docker search <image_name> | Start or stop an existing container \$ docker start stop <container_name> (or <container-id>) | Disconnects a container from a network \$ docker network disconnect <network-name> <container-name> |
| Build an Image from a Dockerfile \$ docker build -t <image_name> | Pull an image from a Docker Hub \$ docker pull <image_name>:<tag> | Remove a stopped container \$ docker rm <container_name> | Remove a network \$ docker network rm <network-name> |
| Build an Image from a Dockerfile without the cache \$ docker build -t <image_name> . --no-cache | Rename an existing Docker Image \$ docker tag <imagename> <newname>:<version> | Open/ Attach a shell inside a running container \$ docker exec -it <container_name> sh | Docker Compose |
| List local images \$ docker images | Status | Kill/ Stop a running container \$ docker kill <container_id> | Create and start containers \$ docker compose up |
| Delete an Image \$ docker rmi <image_name> | Docker Stats of all the Containers \$ docker stats --all | Fetch and follow the logs of a container \$ docker logs -f <container_name> | Stop and remove containers, networks \$ docker compose down |
| Remove all unused images \$ docker image prune | Display the running processes of a container \$ docker top <container_name> or <container_id> | To inspect a running container \$ docker inspect <container_name> (or <container_id>) | View output from containers \$ docker compose logs |
|  | Show History of a Docker Image \$ docker history <imagename or imageid> | To list currently running containers \$ docker ps | Receive real time events from containers \$ docker compose events |
| | | List all docker containers (running and stopped): \$ docker ps --all | List containers launched as part of compose. \$ docker compose ps |
| | | View resource usage stats \$ docker container stats | |

• تبدیل برنامه به کانتینر

• کوبرنتیز

- مقدمه
- اجزای اصلی سازنده
- مفاهیم مقدماتی
- امنیت
- معرفی چند ابزار
- فراهم کننده سرویس کوبر در بستر ابری

• منابع

• ماشین های مجازی

- مجازی سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

- مقدمه
- مقایسه با ماشین مجازی

• داکر

- مقدمات
- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری
- دستورات

داکر - تبدیل برنامه به کانتینر

- نوشتن برنامه
- ساختن داکر فایل (docker file)
- ساختن فایل داکر کامپوز (docker compose) - اختیاری
- واکر کامپوز
 - .yml
 - .yaml
- اجرای دستور مرتبط



داکر - تبدیل برنامه به کانتینر

۱- برنامه پایتون

```
import time

import redis
from flask import Flask

app = Flask(__name__)
cache = redis.Redis(host='redis', port=6379)

def get_hit_count():
    retries = 5
    while True:
        try:
            return cache.incr('hits')
        except redis.exceptions.ConnectionError as exc:
            if retries == 0:
                raise exc
            retries -= 1
            time.sleep(0.5)

@app.route('/')
def hello():
    count = get_hit_count()
    return f'Hello World! I have been seen {count} times.\n'
```

۲- داکر فایل

```
# syntax=docker/dockerfile:1
FROM python:3.10-alpine
WORKDIR /code
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
RUN apk add --no-cache gcc musl-dev linux-headers
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt
EXPOSE 5000
COPY . .
CMD ["flask", "run", "--debug"]
```

۳- فایل داکر کامپوز

```
services:
  web:
    build: .
    ports:
      - "8000:5000"
  redis:
    image: "redis:alpine"
```

۴- اجرای دستور

```
$ docker compose up

Creating network "composetest_default" with the default driver
Creating composetest_web_1 ...
Creating composetest_redis_1 ...
Creating composetest_web_1
Creating composetest_redis_1 ... done
```

• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

- مقدمه
- مقایسه با ماشین مجازی

• داکر

- مقدمات
- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری
- دستورات

- تبدیل برنامه به کانتینر

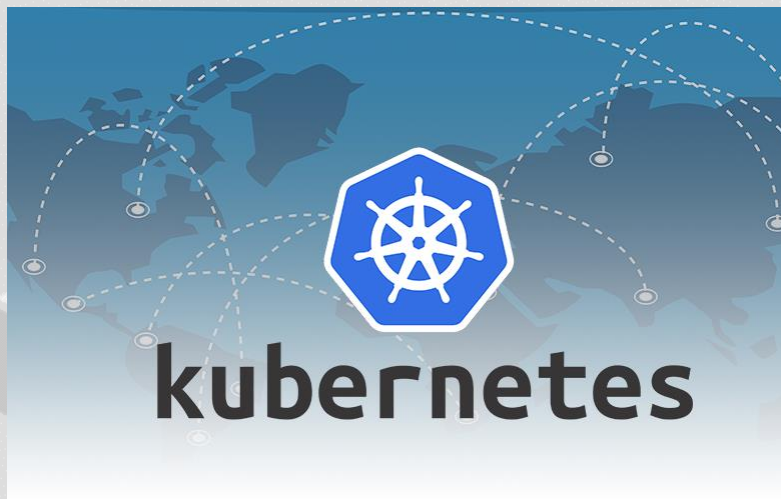
• کوبرنتیز

• مقدمه

- اجزای اصلی سازنده
- مفاهیم مقدماتی
- امنیت
- معرفی چند ابزار
- فراهم‌کننده سرویس کوبر در بستر ابری

• منابع

کوبرنتیز - مقدمه



- K8s
- مدیریت کانتینرها روی چند تا ماشین (Host)
- مکانیزم اپلیکیشن‌ها
 - دیپلوی کردن
 - نگهداری
 - مقایس کردن
- اتوماسیون (automation)
- ساخته گوگل - ۲۰۱۴
- ترکیب بورگ (Borg) و ایده‌های کامیونیتی

کوبرنتیز - مقدمه



- در سطح کانتینر (نه سخت افزار)



- پلتفرم به عنوان سرویس (PaaS)



- مونولوتیک



- فرآیند CI/CD



- لاگ - رصد کردن (مانیتور)



- ابزار و بستر



- ارکستراسیون



- بی نیاز کردن



• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

- مقدمه
- مقایسه با ماشین مجازی

• داکر

- مقدمات
- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری
- دستورات

- تبدیل برنامه به کانتینر

• کوبرنتیز

- مقدمه

• اجزای اصلی سازنده

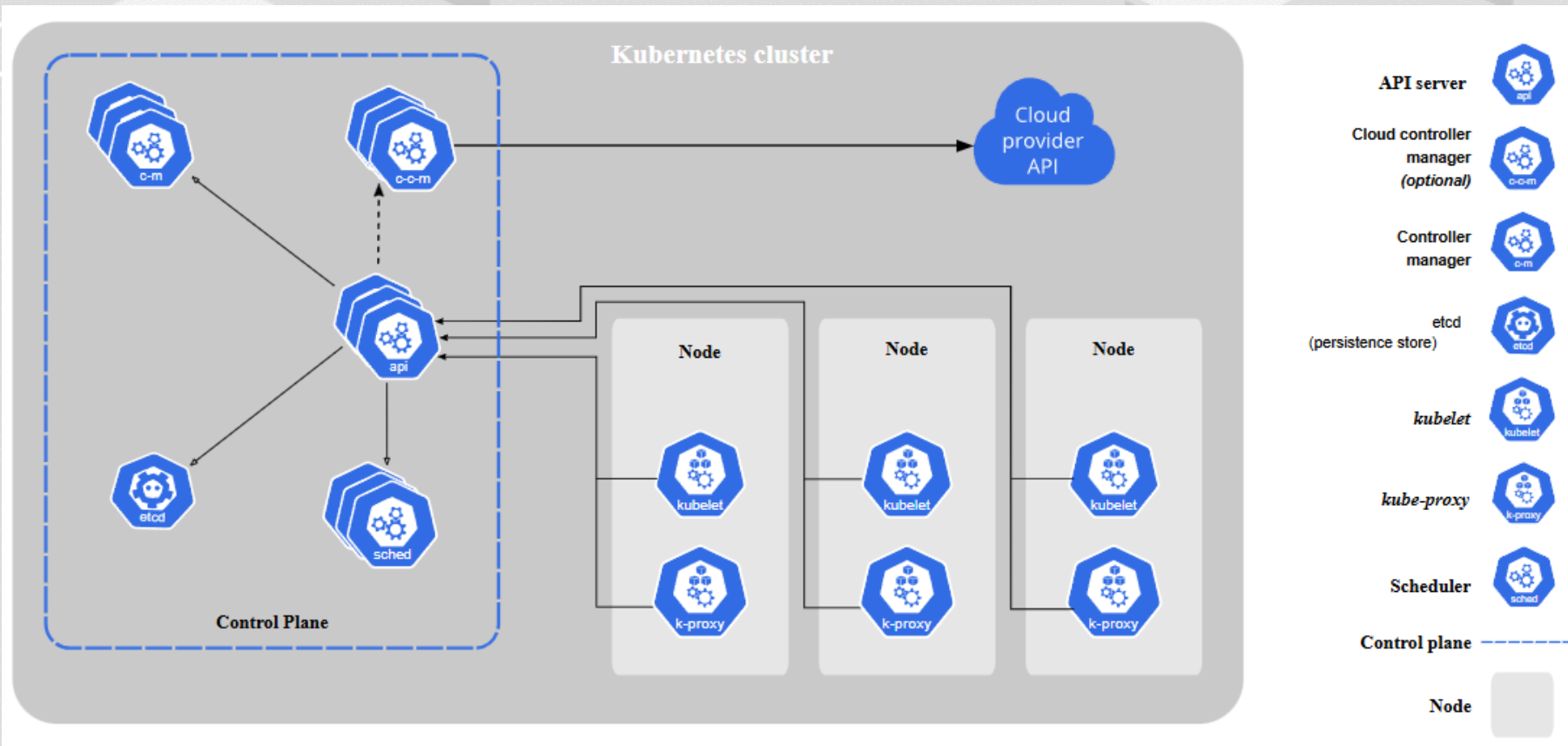
- مفاهیم مقدماتی
- امنیت
- معرفی چند ابزار
- فراهم‌کننده سرویس کوبر در بستر ابری

• منابع

کوبرنتیز – اجزای اصلی سازنده

- صفحه کنترل (control plane)
- مدیریت وضعیت کلی کلاستر
- نود (node)
- نگهداری از پادها
- فراهم کردن محیط اجرا (runtime environment)
- ادونس (addons)
- افزایش کارکرد کوبرنتیز

کوبرنتیز - اجزای اصلی سازنده



کوبرنتیز - اجزای اصلی سازنده - صفحه کنترل

• اجزا

• kube-apiserver

• ای تی سی دی (etcd)

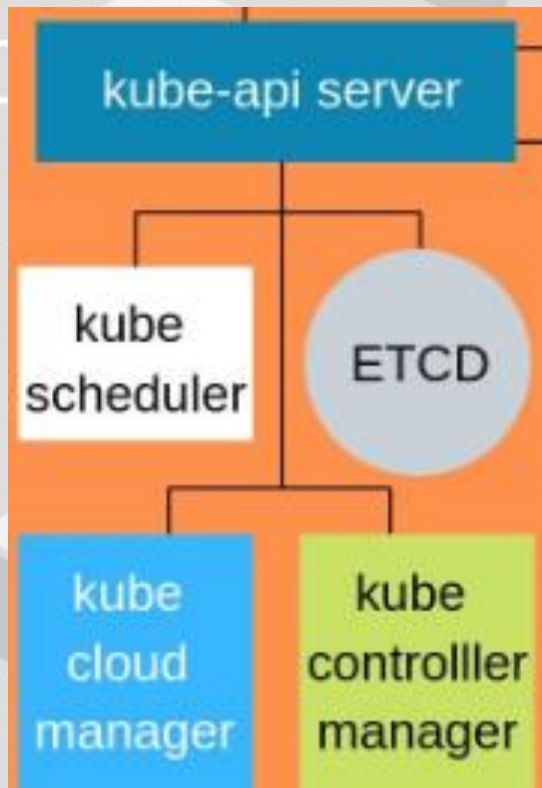
• برنامه ریز کیوب (kube scheduler)

• مدیر کنترل

(kube-controller-manager)

• مدیر کنترل ابری - اختیاری

(cloud-control-manager)



کوبرنتیز - اجزای اصلی سازنده - نود

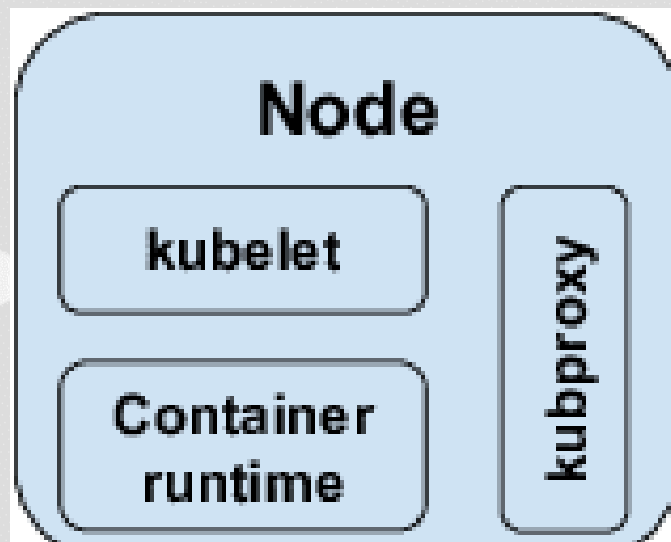
• اجزا

• کیوبلت (kubelet)

• کیوبپراکسی - اختیاری (kube-proxy)

• کانترینر زمان اجرا

(container-runtime)

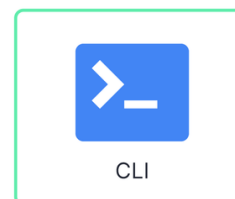
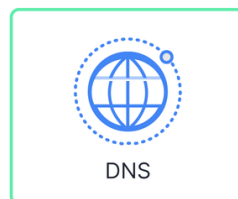
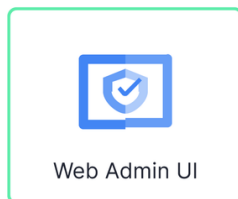


کوبرنتیز - اجزای اصلی سازنده - ادونس

• اجزا

- دیاناس (DNS)
- داشبورد یوزر در وب (Web UI/Dashboard)
- رصد کردن منابع کانتینر (container resource monitoring)
- لاگ در سطح کلاستر (cluster-level logging)

Kubernetes Extensions



• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

- مقدمه
- مقایسه با ماشین مجازی

• داکر

- مقدمات
- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری
- دستورات

- تبدیل برنامه به کانتینر

• کوبرنتیز

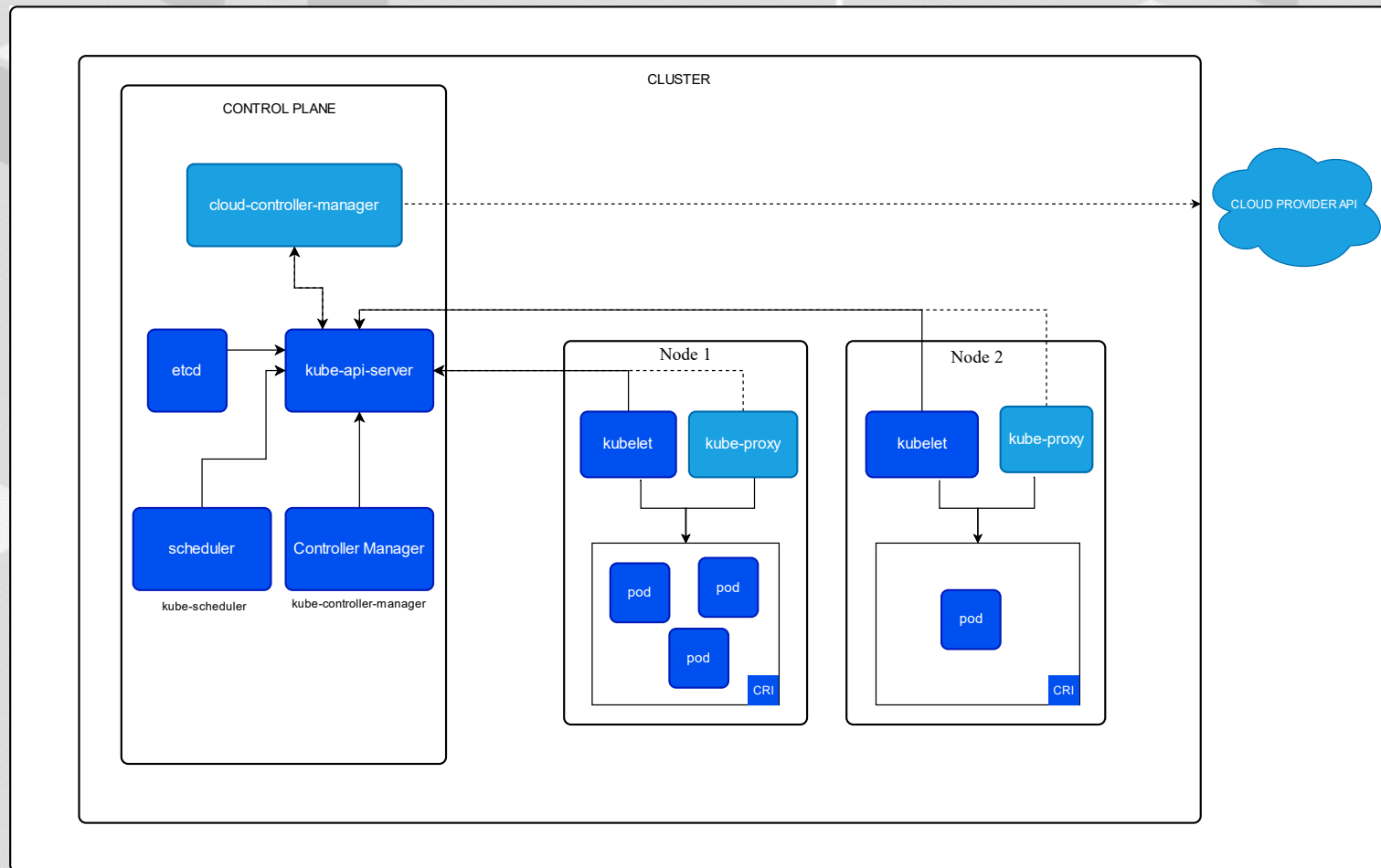
- مقدمه
- اجزای اصلی سازنده
- مفاهیم مقدماتی
- امنیت
- معرفی چند ابزار
- فراهم‌کننده سرویس کوبر در بستر ابری

• منابع

کوبرنتیز - مفاهیم مقدماتی

- کلاستر (cluster)
- نود (node)
 - مدیر (master)
 - کارگر (slave)
- پاد (pod)
- سرویس (service)
- دیپلوی (deployment)
- مجموعه وضعیت (statefulset)
- دیمون ست (DaemonSet)
- نگاشت تنظیمات (config map)
- مدیر کنترل (controller manager)
- برنامه ریز (scheduler)
- ای تی سی دی (etcd)
- نیم اسپیس (namespace)

کوبرنتیز - مفاهیم مقدماتی - کلاستر



کوبرنتیز - مفاهیم مقدماتی - کلاستر

- مجموعه‌ای از نودها
- اجازه اجرا شدن در چند ماشین
- مجازی
- فیزیکی
- ابری (cloud-based)
- اجزا
 - حداقل یک نود مدیر (master node)
 - چندین نود کارگر (slave nodes)

کوبرنتیز - مفاهیم مقدماتی - نود

- یک ماشین

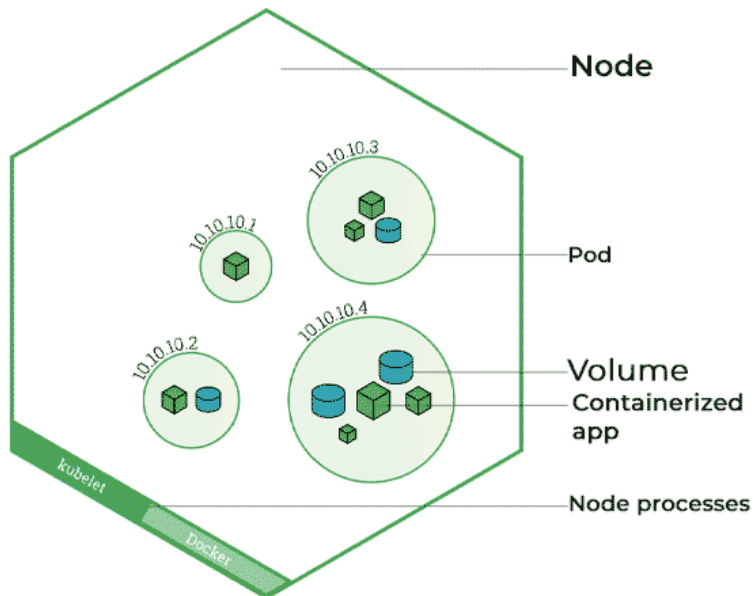
- اجزا

- کیوبپراکسی (kube-proxy)

- کیوبلت (kubelet)

- کانتینر (container)

- داکر



کوبرنتیز - مفاهیم مقدماتی - نود

• انواع

• کارگر (slave)

• همکاری مستقیم در یک شبکه (network)

• قابل جایگزین شدن

• بدون وضعیت (stateless)

• نگه نداشتن داده‌های مداوم (not saving persistent data)

• دلیل راحتی جایگزینی



کوبرنتیز - مفاهیم مقدماتی - نود

- انواع نودها
 - مدیر (master)
 - نگهداری وضعیت کلاستر (saving cluster state)
 - نگهداری در ای تی سی دی (etcd)
 - دیپلوی شدن در نودهای (ماشین‌های) مختلف
 - به شدت در دسترس (highly available)

کوبرنتیز - مفاهیم مقدماتی - نود

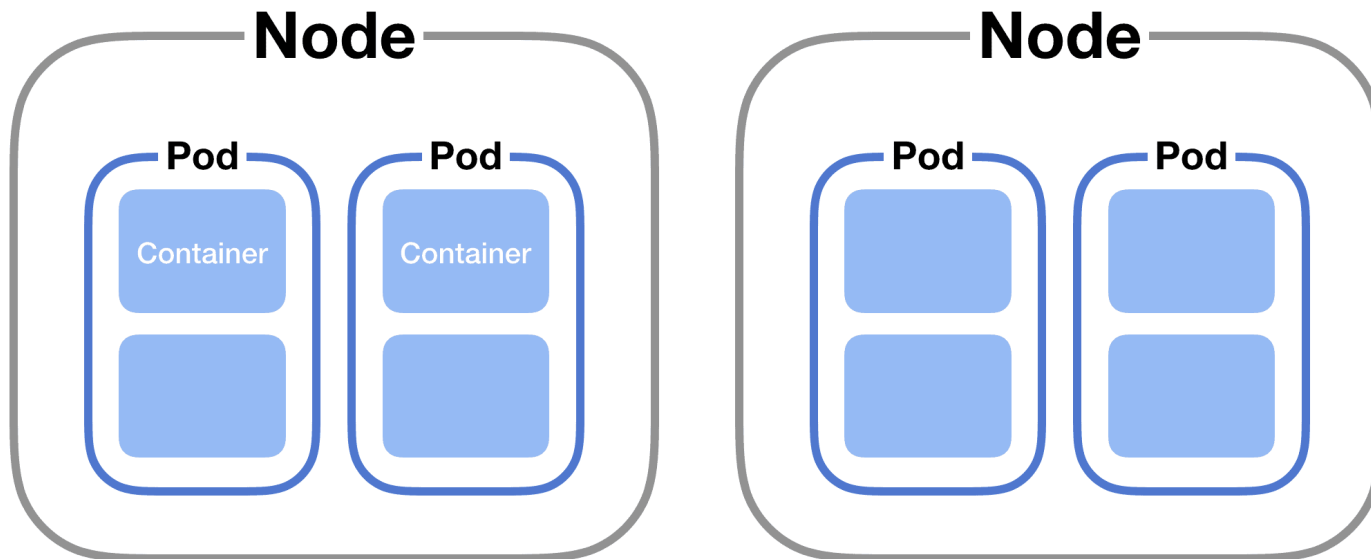
- مزایا روش مدیر و کارگر
- مقیاس پذیری راحت هنگام از بین رفتن نود کارگر
- تخصیص وظیفه به نودهای دیگر
- آسان بودن از بین بردن نود
- پاک کردن از ای تی سی دی (etcd)
- تحمل پذیری خطای بالا (high fault tolerance)
- در صورت بروز خطا برای یک نود کارگر
 - برنامه ریزی مجدد بوسیله نود مدیر
 - تخصیص وظایف به نودهای کارگر در دسترس
 - نتیجه:
- تضمین ادامه به عملکرد کلاستر

کوبرنتیز - مفاهیم مقدماتی - پاد

- اجرای کانتینر
- کوچک‌ترین واحد کوبرنتیز
- استفاده برای اجرای اپلیکیشن
- شامل یک یا چند کانتینر
- معمولاً یک کانتینر
- تقسیم منابع مشترک بین کانتینرها (پیشرفته‌تر)
 - شبکه
 - حافظه
 - نیم‌اسپیس

کوبرنتیز - مفاهیم مقدماتی - پاد

Cluster



کوبرنتیز - مفاهیم مقدماتی - سرویس

- روشی برای باز کردن یک شبکه برای اپلیکیشن
- اپلیکیشن در یک پاد یا پادهای مختلف

• در یک کلاستر

• یک لایه انتزاعی

• نگهداری کارکردهای مهم

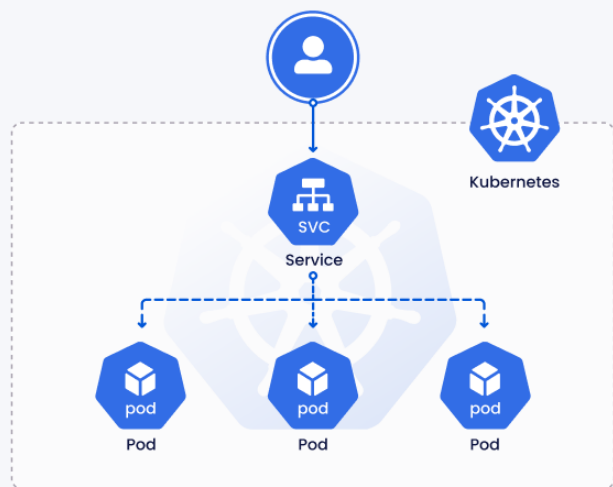
• باز کردن کارکردهای مورد نیاز

• هدف

• قراردادن یک یا چندپادروش شبکه

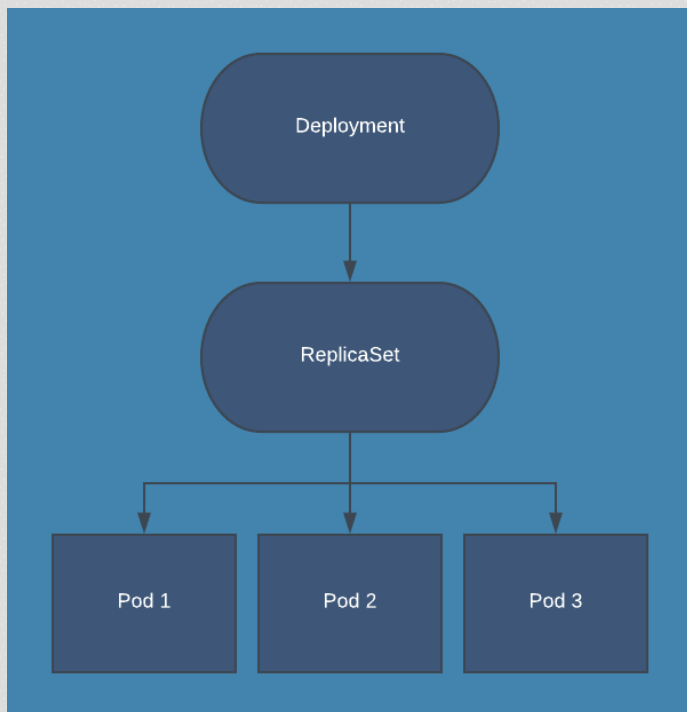
• بدون وضعیت (stateless)

• تضمینی برای نگهداری داده‌ها ندارد



کوبرنتیز - مفاهیم مقدماتی - دیپلوی

- مسئول چرخه زندگی اپلیکیشن (lifecycle)
- فراهم کننده نظم
- مدیریت تعداد نمونه های اپلیکیشن (instance)
- نمونه اپلیکیشن
- پاد
- مفهوم رپلیکا (replica) و رپلیکاست



کوبرنتیز - مفاهیم مقدماتی - دیپلوی

- تضمین وجود تعداد مشخص شده از نمونه‌ها

- ساختن پاد جدید

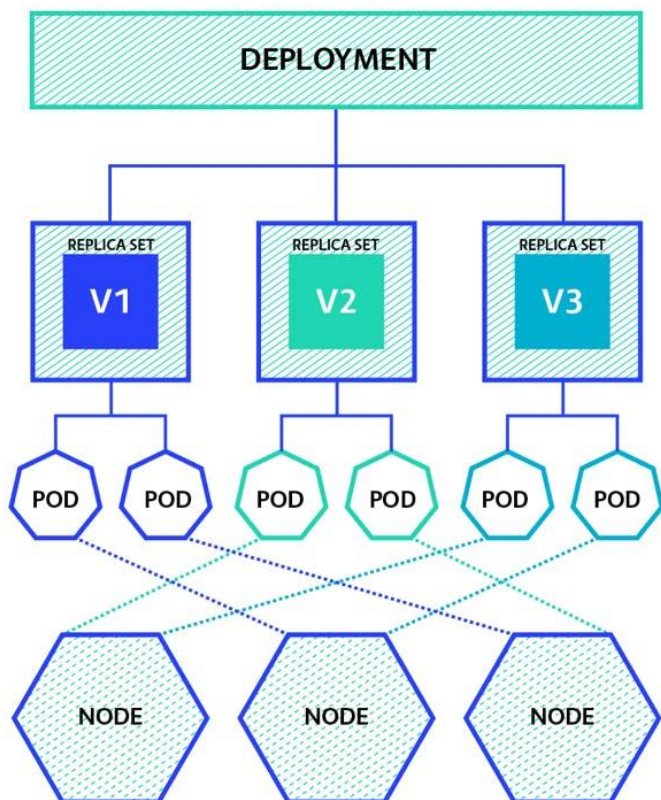
- خطا (fail)

- نابودی (termination)

- مقیاس‌بندی (scaling)

- اضافه کردن نمونه‌ها

- در صورت افزایش درخواست



کوبرنتیز - مفاهیم مقدماتی - مجموعه وضعیت

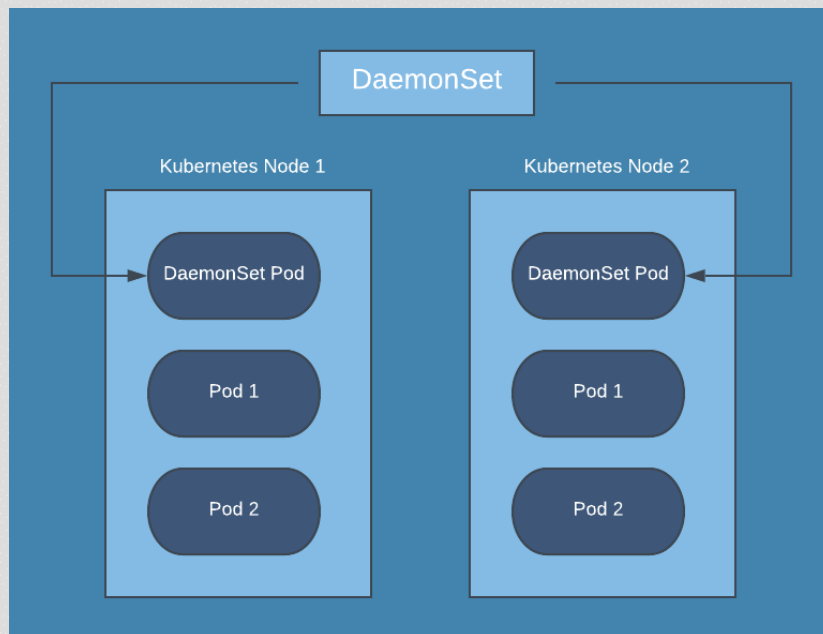
- منبع مجازی (virtual resource)
- مدیریت اپلیکیشن‌های نیازمند داده‌های ماندگار
- نگهداری وضعیت در طول زمان
- برخلاف دیپلویمنت
- مدیریت مجموعه‌ای از پادها
 - دیپلوی کردن
 - مقیاس کردن

کوبرنتیز - مفاهیم مقدماتی - مجموعه وضعیت

- تضمین
 - نظم و ترتیب پادها
 - یکتا بودن (uniqueness)
- مثال
 - یک مجموعه وضعیت سه تایی
 - پادهایی با اسامی زیر
 - وب یک
 - وب دو
 - وب سه
 - از بین رفتن پاد یک
 - ایجاد دوباره پاد با همان اسم

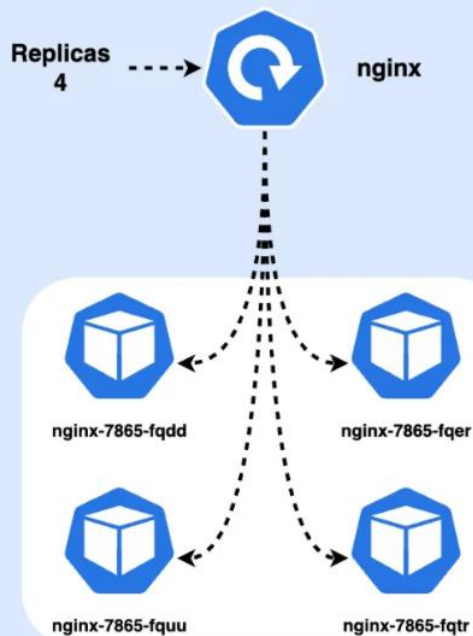
کوبرنتیز - مفاهیم مقدماتی - دیمون ست

- برای اجرا روی تمام نودها
- تضمین اجرای یک پاد در هر نود کارگر
- مقیاس پذیر نیست
- فقط یکی
- در صورت از بین رفتن
- تلاش برای ساختن دوباره
- قابل محدود کردن
- انتخاب نود برای اجرا (nodeSelector)

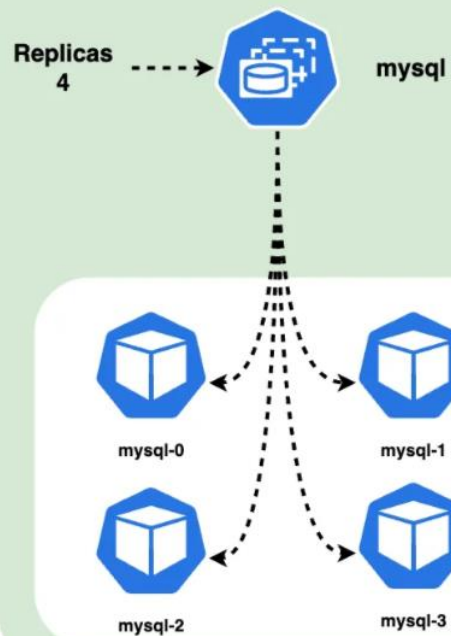


کوبرنتیز - مفاهیم مقدماتی - مقایسه

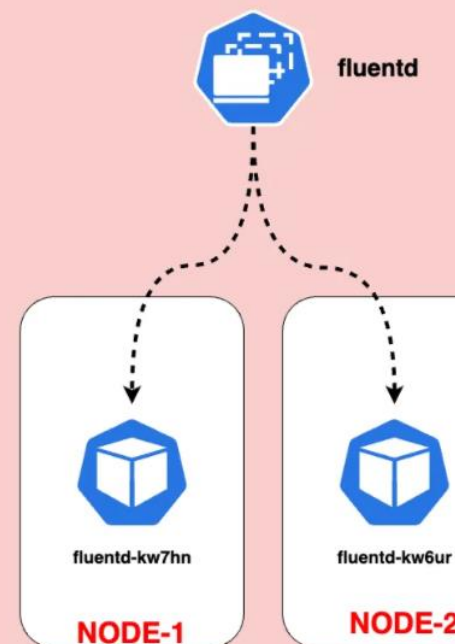
Deployment



StatefulSet



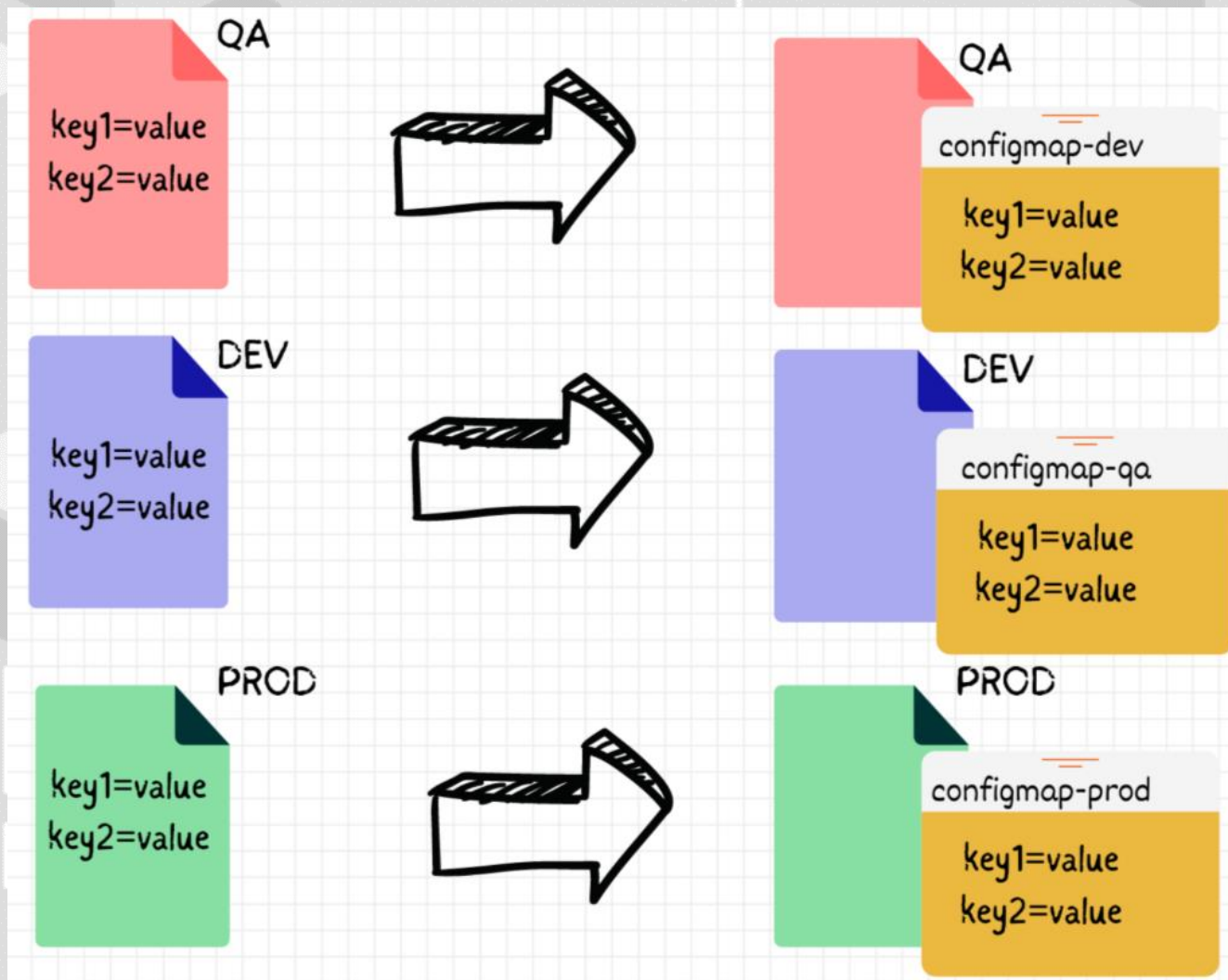
DaemonSet



کوبرنتیز - مفاهیم مقدماتی - نداشت تنظیمات

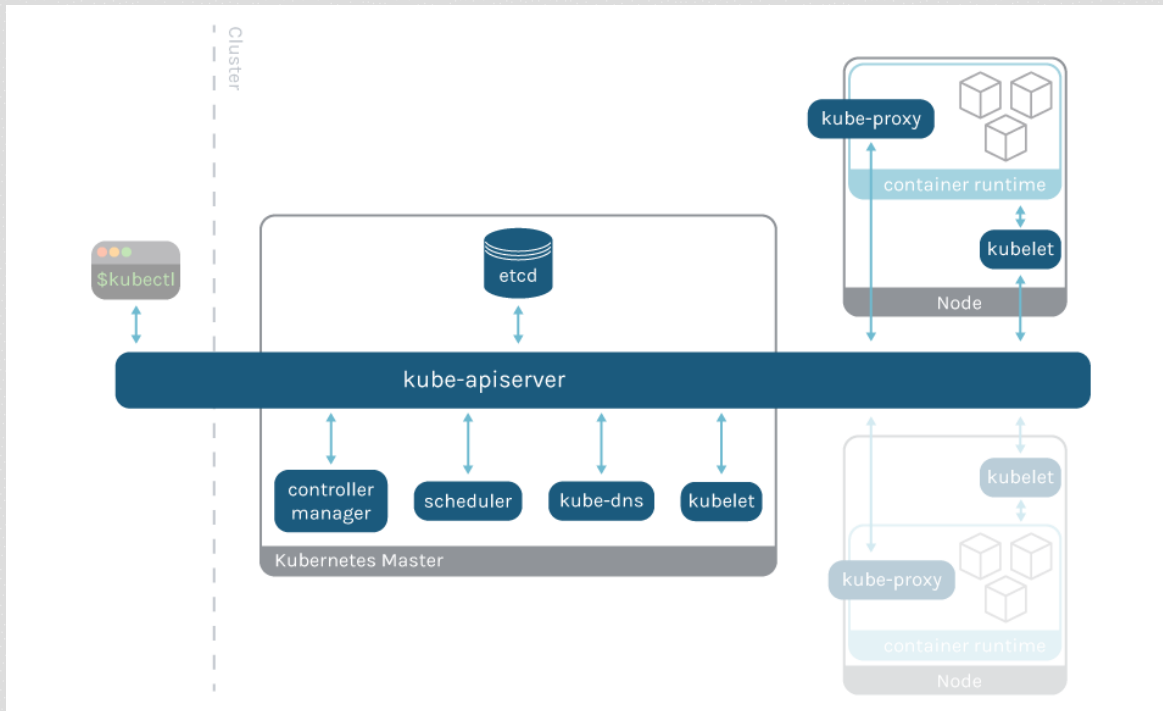
- نگهداری داده‌های غیر مهم
- نگهداری با شمایل کلید - داده (key - value)
- استفاده
- به عنوان ورودی خط اجرا پادها (command line argument)
- متغیرهای محیطی (environment variables)
- فایل تنظیمات روی دیسک
- مزیت
- جداسازی تنظیمات از ایمیج کانتینر
- متغیر بودن تنظیمات

کوبرنتیز - مفاهیم مقدماتی - نداشت تنظیمات



کوبرنتیز - مفاهیم مقدماتی - مدیر کنترل

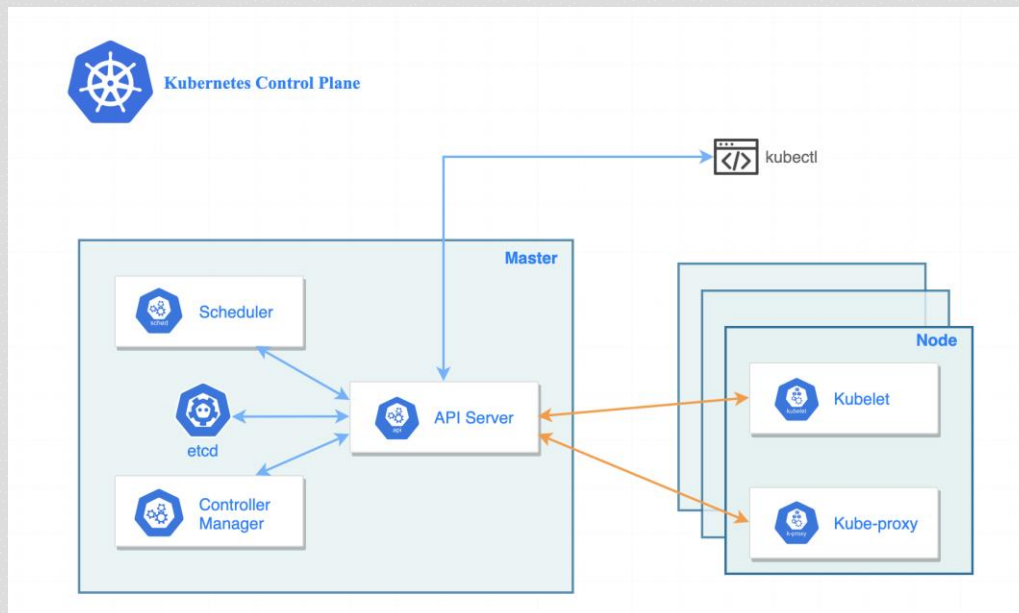
- مدیر کنترل (kube controller manager)
- یک دیمون مهم در کلاستر
- مرکز کنترل ماندگاری وضعیت کلاستر
- فرآیند همواره در حال اجرا



کوبرنتیز - مفاهیم مقدماتی - مدیر کنترل

- وظایف مهم
- رصد کردن پیوسته (continuous monitoring)
- رصد کردن مداوم کلاستر
- از طریق پای پی آی سرور کوبرنتیز (API server)

- شامل دنبال کردن
- پادها
- دیپلویمنتها
- سرویسها
- منابع دیگر



کوبرنتیز - مفاهیم مقدماتی - مدیر کنترل

- وظایف مهم (ادامه)
- تطابق وضعیت (state reconciliation)
- پیدا کردن تفاوت وضعیت کنونی با وضعیت مورد انتظار
- عملیات اصلاح (corrective actions)
- بعد از پیدا کردن تفاوت بین وضعیت کنونی و مورد انتظار
- اصلاح بوسیله کنترلر مربوطه
- شامل
 - مقایس کردن پادها
 - شروع مجدد کانتینرهای خطا خورده (restarting failed containers)
 - ساختن مجدد منابع مورد نیاز

کوبرنتیز - مفاهیم مقدماتی - مدیر کنترل

• شامل چندین کنترلر

- کنترلر رپلیکیشن (Replication Controller)
 - مسئول تضمین بالا بودن تعداد مشخصی از پادها
- کنترلر اندپوینت (Endpoint Controller)
 - مسئول ماندگاری اندپوینت هر سرویس
- کنترلر نیم‌اسپیس (Namespace Controller)
 - مسئول ساختن و نگهداری از نیم‌اسپیس‌ها
- کنترلر حساب سرویس (Service Account Controller)
 - مسئول ساختن و مدیریت حساب سرویس (service account) هر پاد
- کنترلر نود (Node Controller)
 - مسئول بررسی وضعیت سلامت و دسترس‌پذیری هر نود
- کنترلر توکن (Token Controller)
 - مسئول بررسی مشکلات توکن حساب سرویس‌ها (service account)
- کنترلر اجازه (Lease Controller)
 - مسئول مکانیزم اجازه (Leasing Mechanism) برای دسترسی به منابع مشترک (shared resource)

کوبرنتیز - مفاهیم مقدماتی - برنامه ریز

- مسئول دادن پادها به نود در یک کلاستر

- مسئولیت اصلی

- بهینه سازی مصرف منابع

- تضمین اجرا شدن کارا و راحت اپلیکیشن

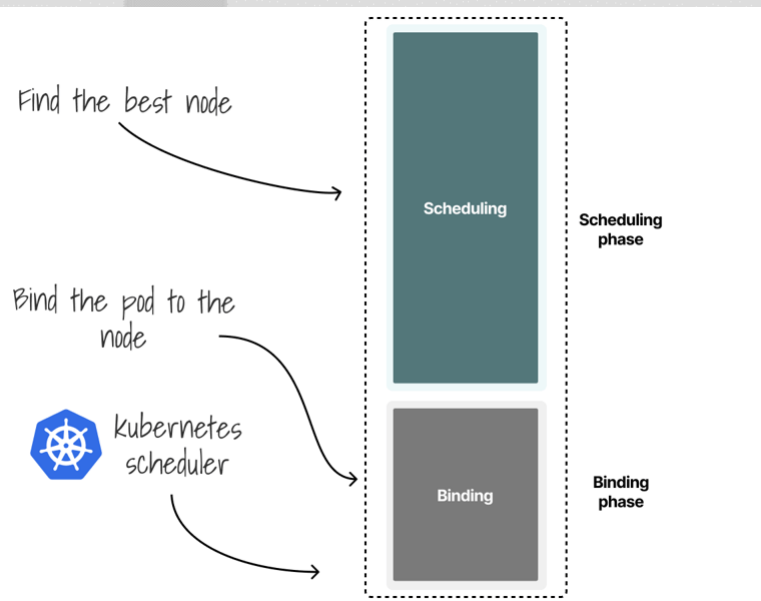
- وابسته

- توانایی سخت افزار

- منابع در دسترس

- کیفیت سرویس (Quality of Service - QoS)

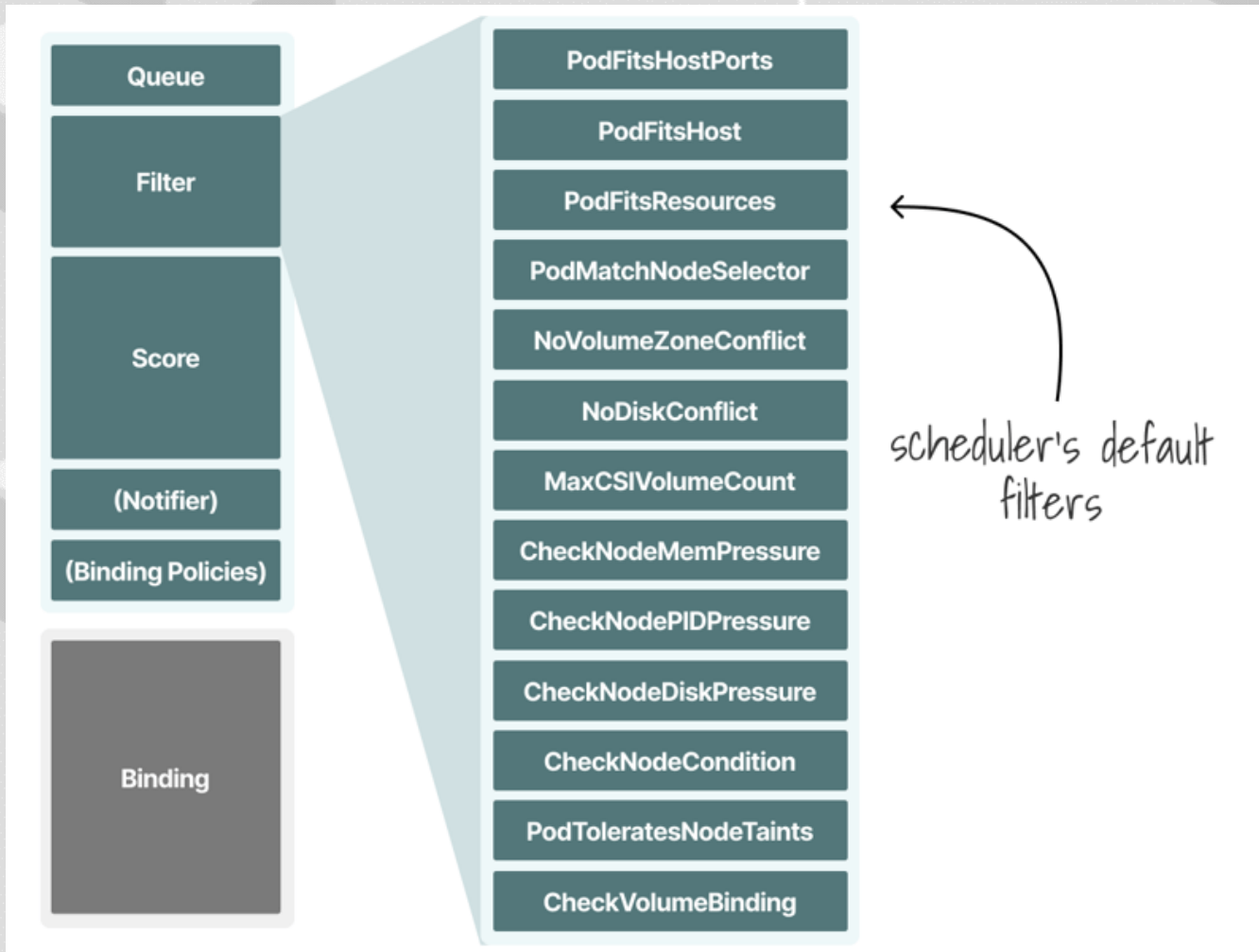
- تنظیمات



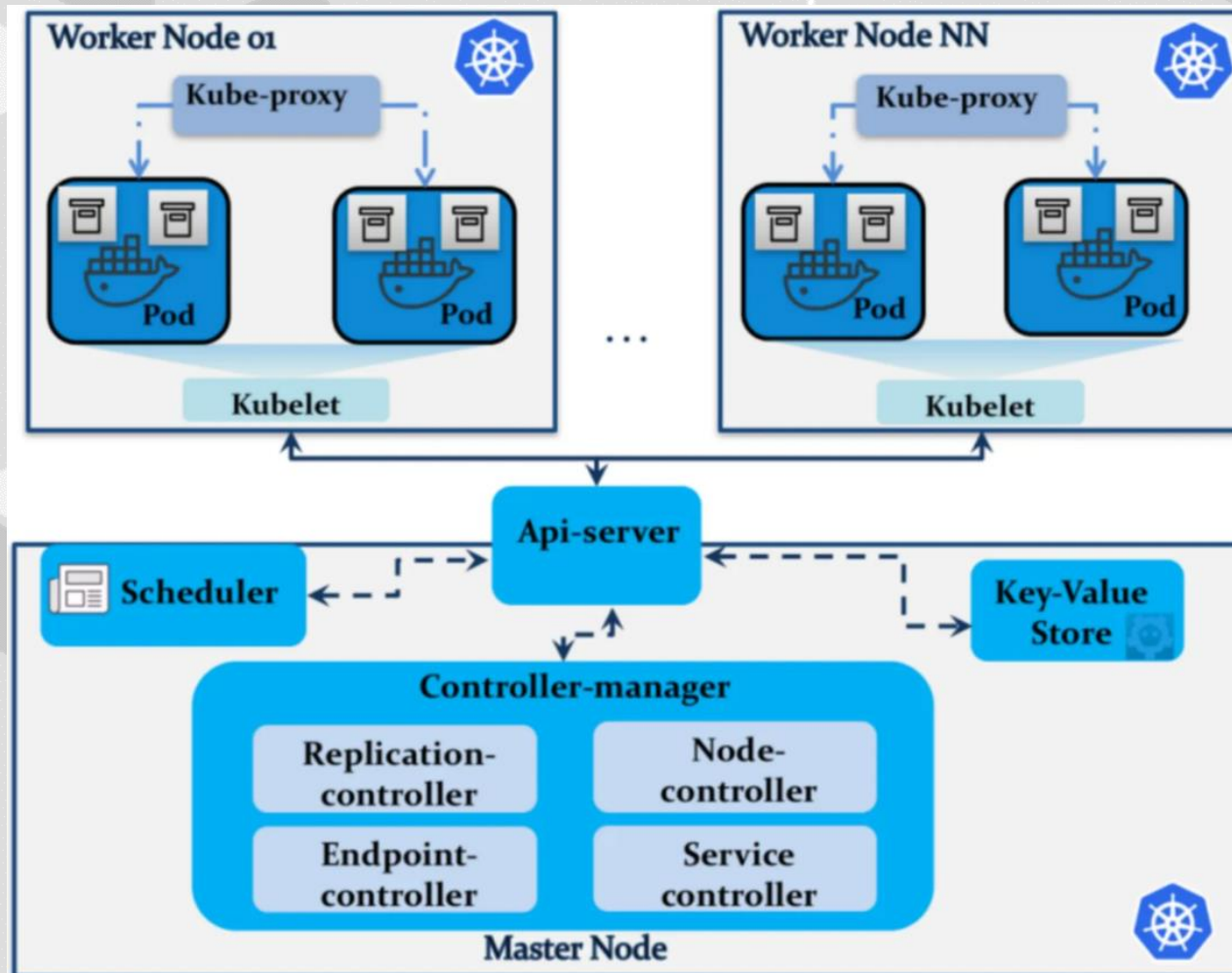
کوبرنتیز - مفاهیم مقدماتی - برنامه ریز

- تاثیر در صورت انجام بهینه برنامه ریزی
 - بهبود استفاده از منابع
 - نقش مهم توزیع منابع (resource allocation)
 - بهبود عملکرد اپلیکیشن
 - تضمین در دسترس بودن بالا (Highly Available)
 - توزیع حجم کاری در سیستم (Workload distribution)
 - حجم کاری درخواستها (Load balancing)
 - خطاپذیری (Fault tolerancing)
 - کمتر کردن تنگناها (bottlenecks)

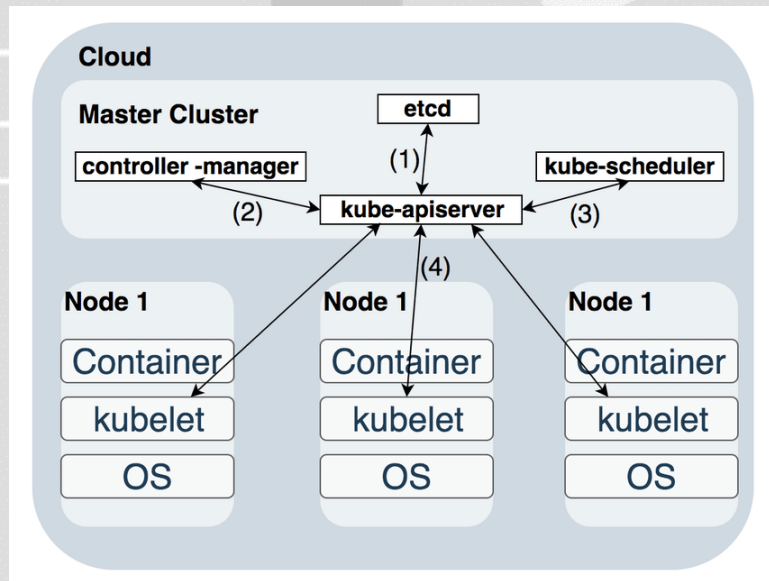
کوبرنتیز - مفاهیم مقدماتی - برنامه ریز



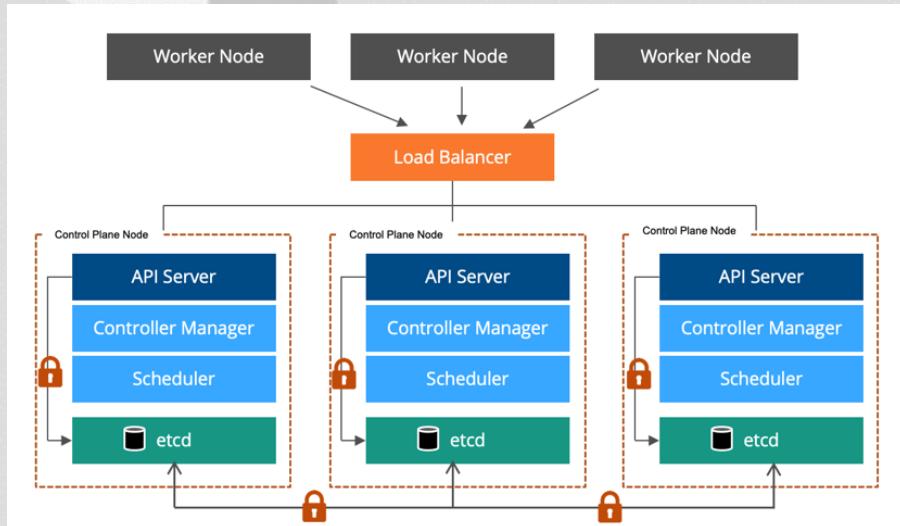
کوبرنتیز - مفاهیم مقدماتی - برنامه ریز



کوبرنتیز - مفاهیم مقدماتی - ای تی سی دی

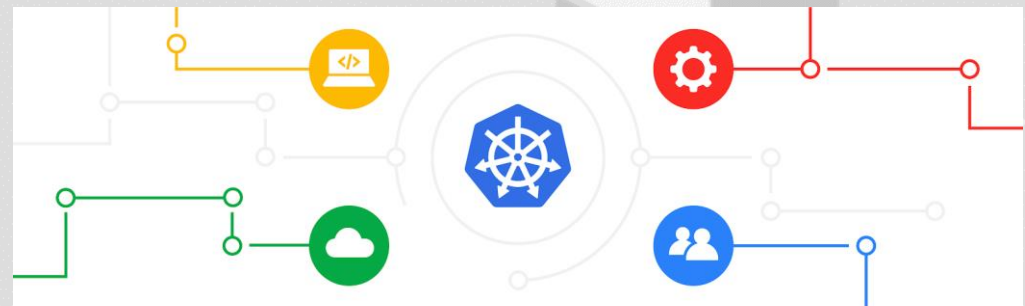
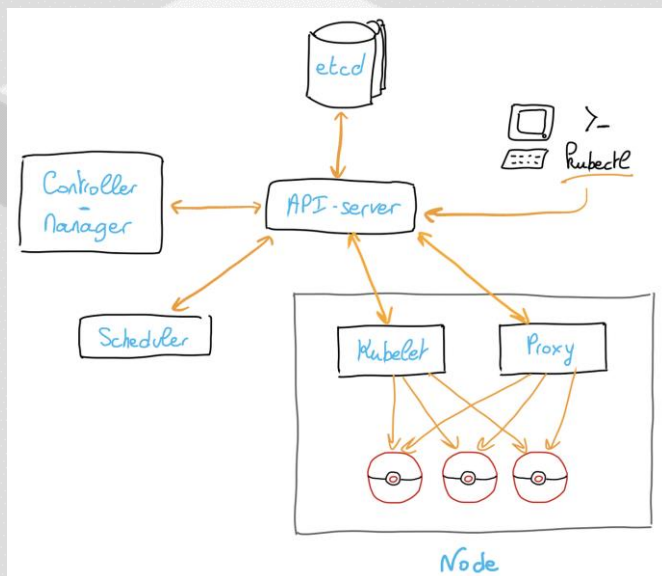


- نگهداشتن تمام اطلاعات کلاستر
- وضعیت فعلی (current state)
- وضعیت مورد نظر (desired state)
- تنظیمات منابع (resource configuration)
- داده‌های زمان اجرا (runtime data)
- نگهداری به صورت کلید-داده (key-value)
- پیدا کردن سرویس‌ها (service discovery)



کوبرنتیز - مفاهیم مقدماتی - ای تی سی دی

- کارایی
- رصد کردن نودها
- پیدا کردن منابع خالی (در دسترس)
- رصد کردن سلامتی نودها
- پیاده‌سازی چند مکانیزم برای جلوگیری از استارویشن (starvation) منابع
- تضمین در دسترس بودن و قابل اطمینان بودن



کوبرنتیز - مفاهیم مقدماتی - ای تی سی دی

• کارایی (ادامه)

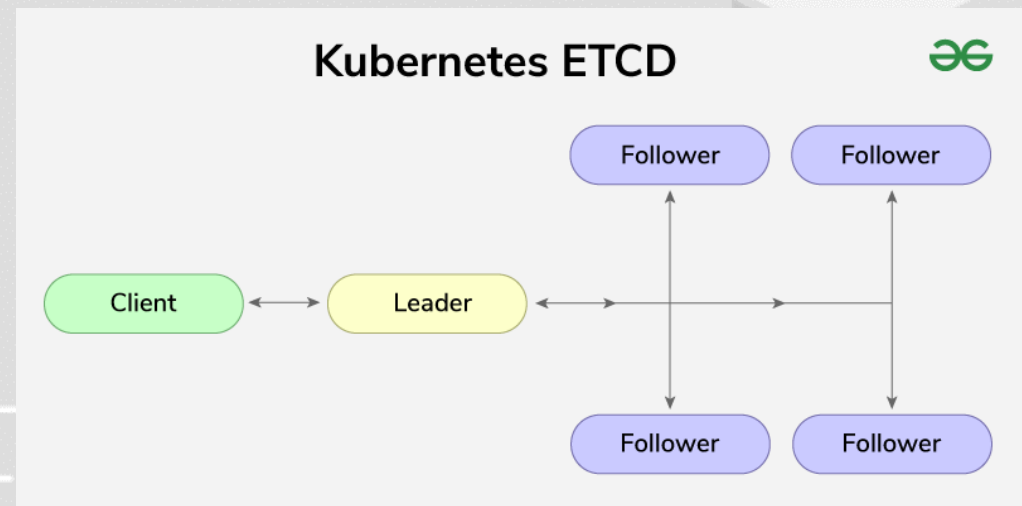
• ویژگی‌ها سیستم‌های توزیع شده (پیاده‌سازی الگوریتم‌ها)

• پیدا کردن سرویس‌ها (service discovery)

• انتخاب فرمانده (Leader Election)

• قفل توزیع شده (Distributed Locks)

• سایر موارد

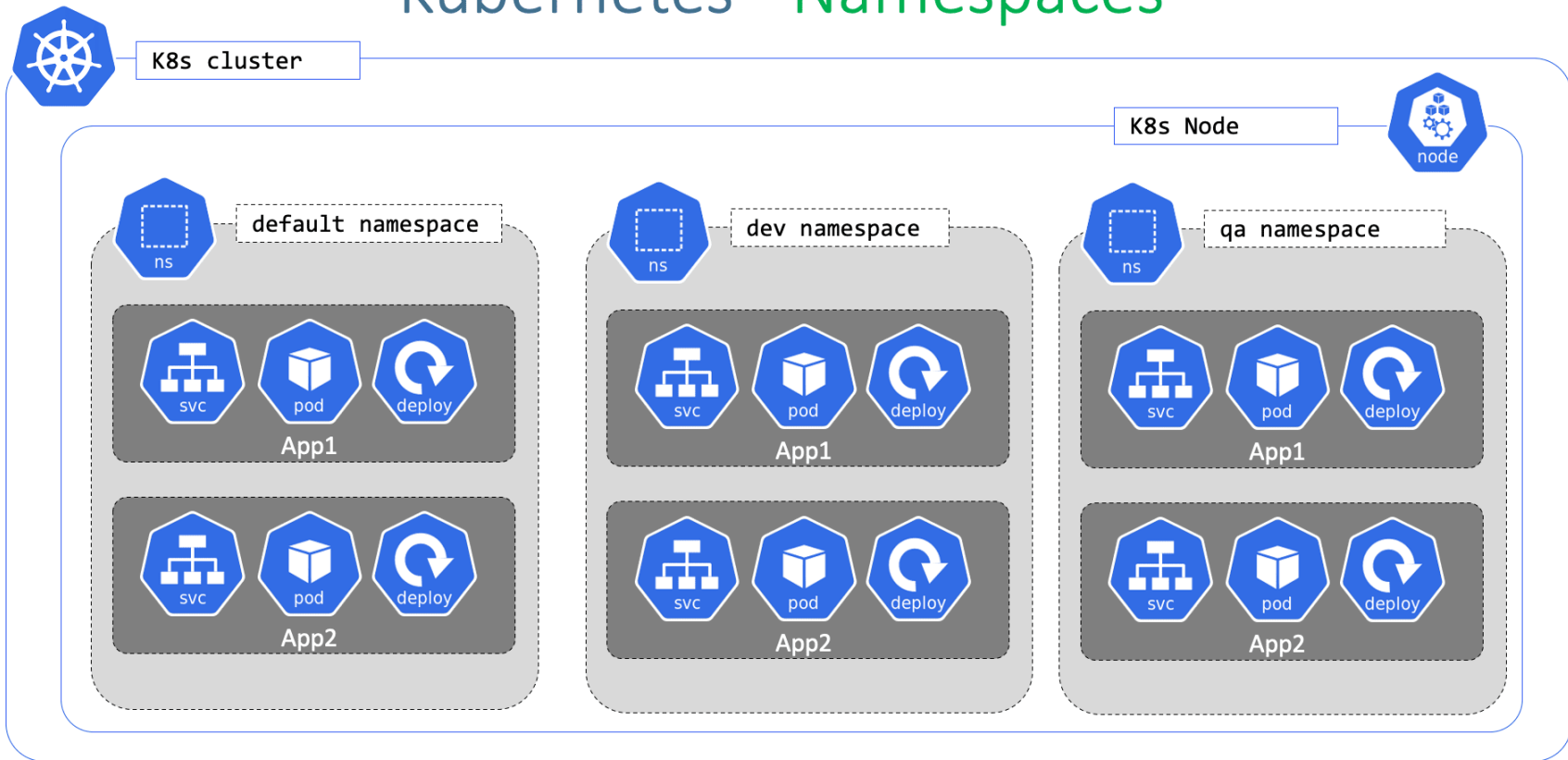


کوبرنتیز - مفاهیم مقدماتی - نیم‌اسپیس

- راهی برای تبدیل کلاستر به زیرکلاستر مجازی (virtual sub-cluster)
- تفاوت زیرکلاسترها از نظر منطقی
- امکان ارتباط بین دو زیرکلاستر
- استفاده در زمان استفاده همزمان
 - تیم‌های متفاوت
 - پروژه‌های متفاوت
- بدون محدودیت در تعداد نیم‌اسپیس‌ها در یک کلاستر

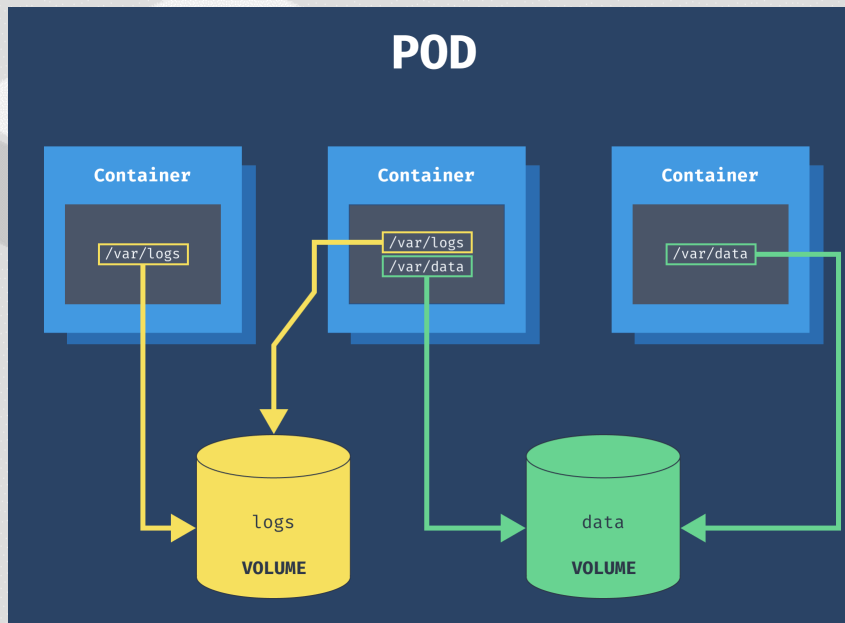
کوبرنتیز - مفاهیم مقدماتی - نیم اسپیس

Kubernetes - Namespaces



کوبرنتیز - مفاهیم مقدماتی - والیوم

- دایرکتوری (directory) با داده‌های مشترک
- بین چند کانتینر در یک پاد
- تفاوت والیوم و کانتینر
- ماندگاری داده‌ها در والیوم بعد از شروع مجدد ناشی از خطا (crash)
- برداشتن داده از وضعیت قبلی (data at the state before crash)



کوبرنتیز - مفاهیم مقدماتی - والیوم

- مدل والیوم‌ها
 - زودگذر (Ephemeral)
 - زمان ماندگاری (Lifetime) برابر با پاد
 - استفاده برای داده‌های گذرا (temporary)
 - داده‌ها و اپلیکیشن‌های بدون نیاز به ماندگاری داده‌ها (data persistency)
 - سریع
 - انواع از این مدل
 - دایرکتوری خالی (emptyDir)
 - اولین موجودیت ساخته شده هنگام واگذار شدن (assign) پاد به نود
 - نگاشت تنظیمات (configMap)
 - محرمانه (Secret)
 - سایر موارد

کوبرنتیز - مفاهیم مقدماتی - والیوم

• مدل والیوم‌ها - ادامه

• با دوام (Durable)

• زمان ماندگاری (Lifetime) جدا از پاد

• ماندگاری داده‌ها در زمان متوقف شدن (crash) یا پاک شدن کانتنر

• انواع این مدل

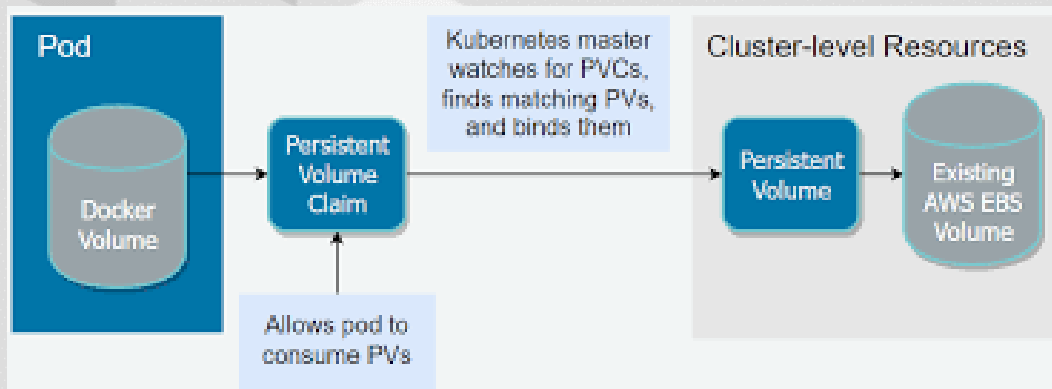
• والیوم کیلیم‌های ماندگار (persistentVolumeClaim)

• بوک‌استور الاستیک (awsElasticBlockStore)

• دیسک آزور (azureDisk)

• دیسک ماندگار جی‌سی‌ای (gcePersistentDisk)

• سایر موارد



کوبرنتیز - مفاهیم مقدماتی - والیوم ماندگار

- بیانگر حافظه ماندگار در کوبرنتیز

- منظور از حافظه

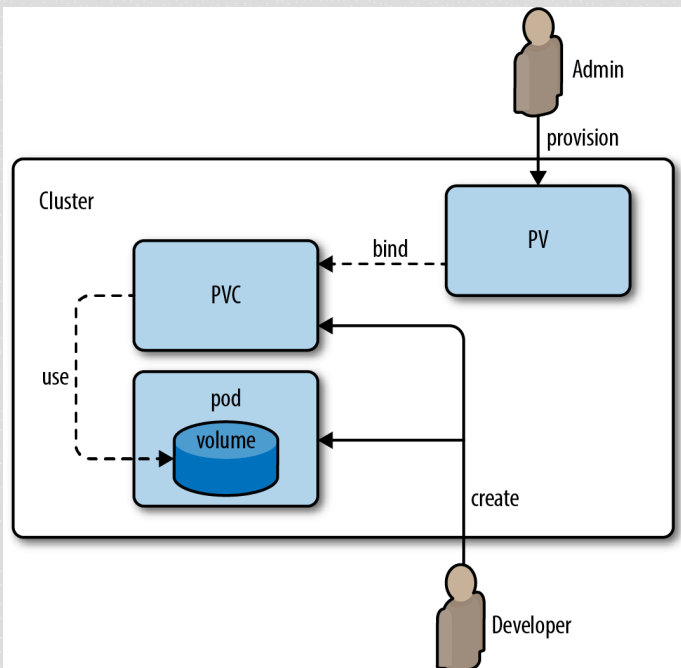
- حافظه در کلاستر (storage resource in cluster)

- ارتباط تنگاتنگ با ادعای والیوم ماندگار (Persistent Volume)
(Claims - PVC)

- ادعای والیوم ماندگار (PVC)

- اجازه دادن به پاد برای درخواست والیوم ماندگار کردن

- بهینه‌سازی حافظه ماندگار در کلاستر



کوبرنتیز - مفاهیم مقدماتی - والیوم ماندگار

- انواع
 - محلی (local)
 - ذخیره‌سازی داده‌ها به صورت محلی در نودهای کلاستر
 - مسیر میزبان (hostPath)
 - ذخیره‌سازی داده‌ها در دایرکتوری ناگذاری شده در نود
 - طراحی شده برای اهداف تستی (تست کردن)
 - فایل سیستم شبکه (nfs - network file system)
 - آی‌اس‌سی‌آی (iscsi)
 - سی‌اس‌آی (Container Storage Interface - CSI)
 - سف (CephFS)
 - اف‌سی (Fibre channel - fc)
 - آر‌بی‌دی (Rados Black Device - RBD)

• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

- مقدمه
- مقایسه با ماشین مجازی

• داکر

- مقدمات
- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری
- دستورات

- تبدیل برنامه به کانتینر

• کوبرنتیز

- مقدمه
- اجزای اصلی سازنده
- مفاهیم مقدماتی

• امنیت

- معرفی چند ابزار
- فراهم‌کننده سرویس کوبر در بستر ابری

• منابع

کوبرنتیز-امنیت

- محرمانگی با سِکِرِت
- حاوی داده‌های حساس و مهم (sensitive) کم حجم
- مثال
 - پسورد
 - یوزرنیم
 - توکن
 - کلید
 - سرتیفیکیت‌ها (certificates)
- اگر نبود
 - داخل پاد
 - داخل ایمج

کوبرنتیز-امنیت

- نحوه استفاده

- متغیرهای محیطی در کانتینر (Environment Variables)

- استفاده راحت

- دسترسی راحت اپلیکیشن

- فایل در کانتینر

- قرارگرفتن (mounted) به عنوان فایل در کانتینر

- به عنوان ورودی خط اجرا پادها (command line argument)

کوبرنتیز-امنیت



• انواع سِکِرِت

• کِدِر یا مات (Opaque)

• پایه‌ای

• نگهداری داده‌ها در فرمت دلخواه

• جِیسون (json)

• باینری (binary files)

• کلید-داده (key-value)

• برای نگهداری داده‌های حساس

• پسورد

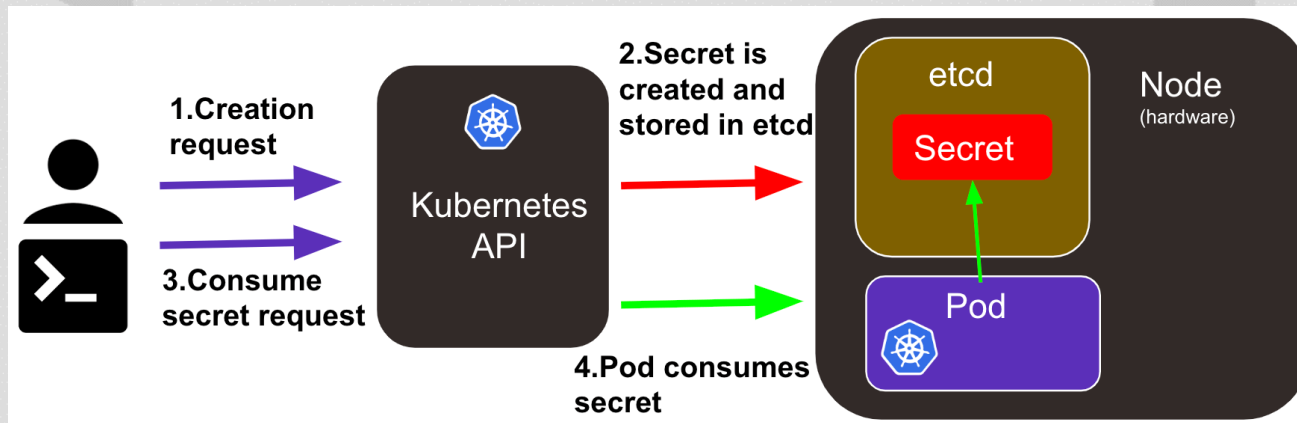
• سرتیفیکیت‌ها (certificates)

• توکن (token)

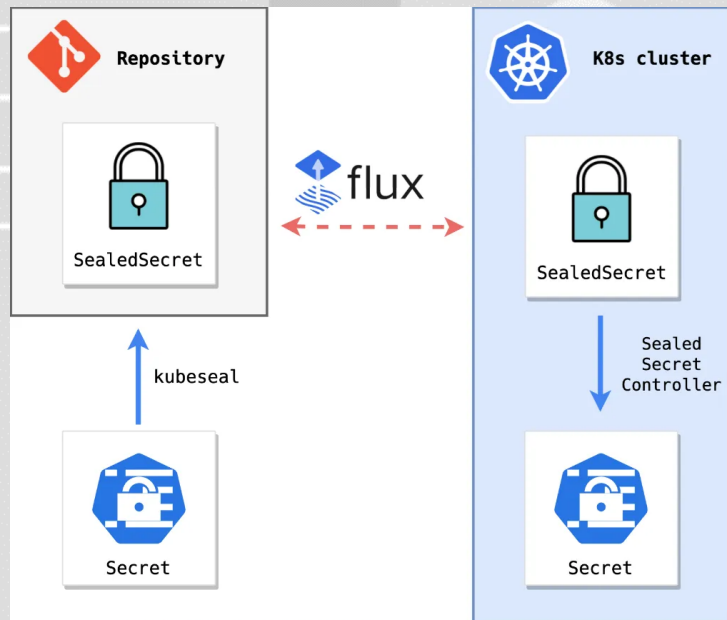
کوبرنتیز-امنیت

- انواع سِکِرِت - ادامه
- تی‌اِل‌اِس (TLS)
- استفاده

- برای نگهداری سرتیفیکیت‌ها و کلیدها
- امن کردن ارتباط بین اعضای مختلف کوبرنتیز
- در یک کلاستر
- بین سرویس‌ها و کلاینت‌های خارجی (external clients)



کوبرنتیز-امنیت



• انواع سِکِرِت - ادامه

• داکر سی‌اف‌جی (Dockerconfig)

• یکی از اجزای داکر

• داکر

• ذخیره کردن داده‌های حساس داکر

• مثال

• اطلاعات رجیستری (registry credentials)

• توکن احراز هویت (authentication token)

• کوبرنتیز

• در زمان اجرا (container runtime)

• برای گرفتن ایمج از رجیستری

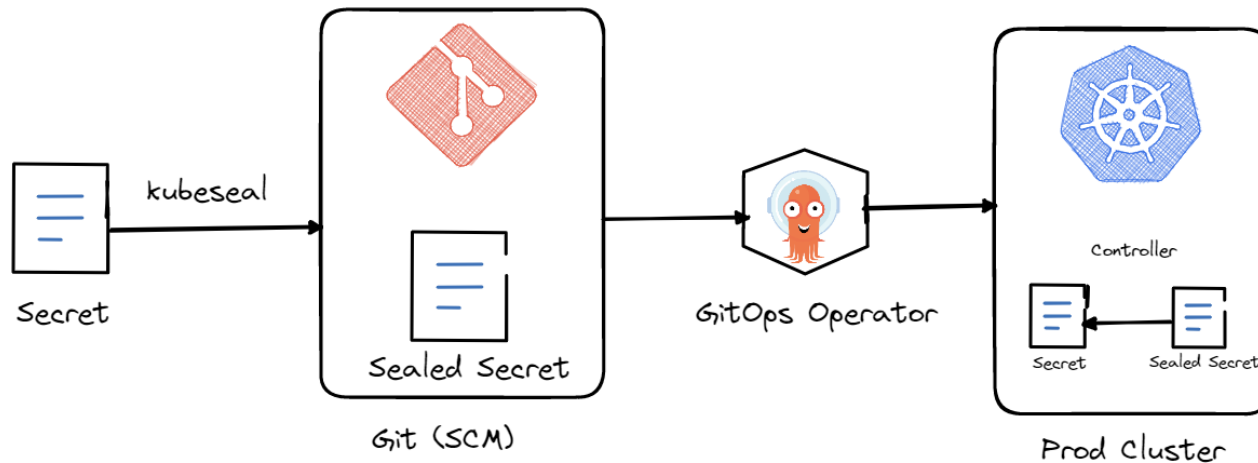
کوبرنتیز-امنیت

- انواع سِکِرِت - ادامه

- اِس_اِس_اِچ (SSH)

- نگهداری کلیدهای اِس_اِس_اِچ (SSH)

- قابل احراز شدن بوسیله سرویس‌های دیگر



Sealed Secrets Flow

• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

- مقدمه
- مقایسه با ماشین مجازی

• داکر

- مقدمات
- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری
- دستورات

• تبدیل برنامه به کانتینر

• کوبرنتیز

- مقدمه
- اجزای اصلی سازنده
- مفاهیم مقدماتی
- امنیت

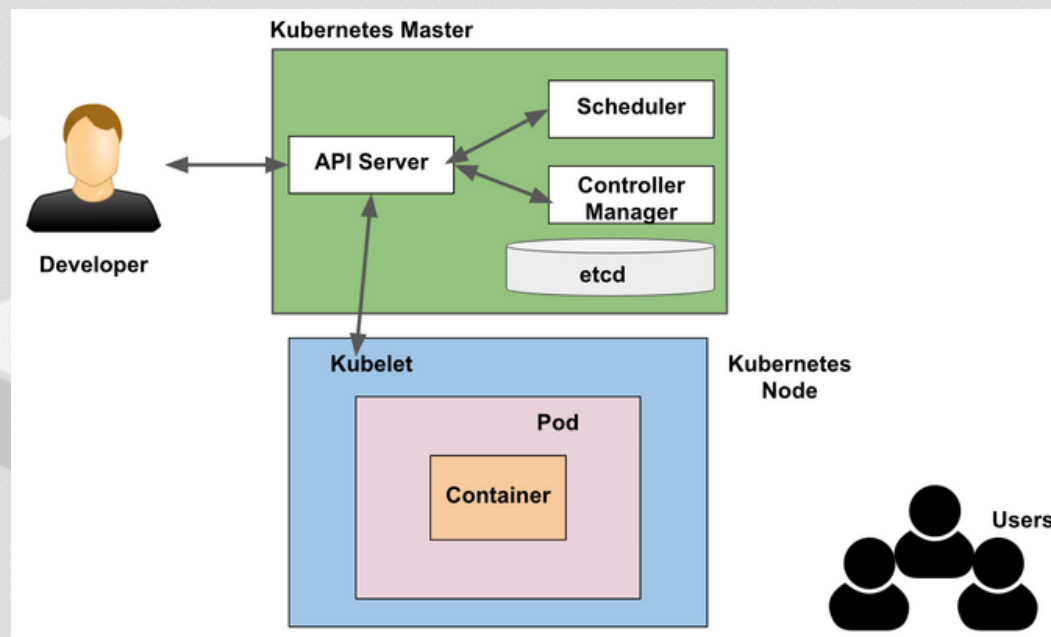
• معرفی چند ابزار

- فراهم‌کننده سرویس کوبر در بستر ابری

• منابع

کوبرنتیز - معرفی چند ابزار - کیوبلت

- کیوبلت
- مامور نود اولیه (primary node agent)
- توانایی اضافه کردن نود با ای پی آی سرور (API server)



کوبرنتیز - معرفی چند ابزار - هلم



• هلم

• کمک کردن در مدیریت اپلیکیشن‌های کوبرنتیز

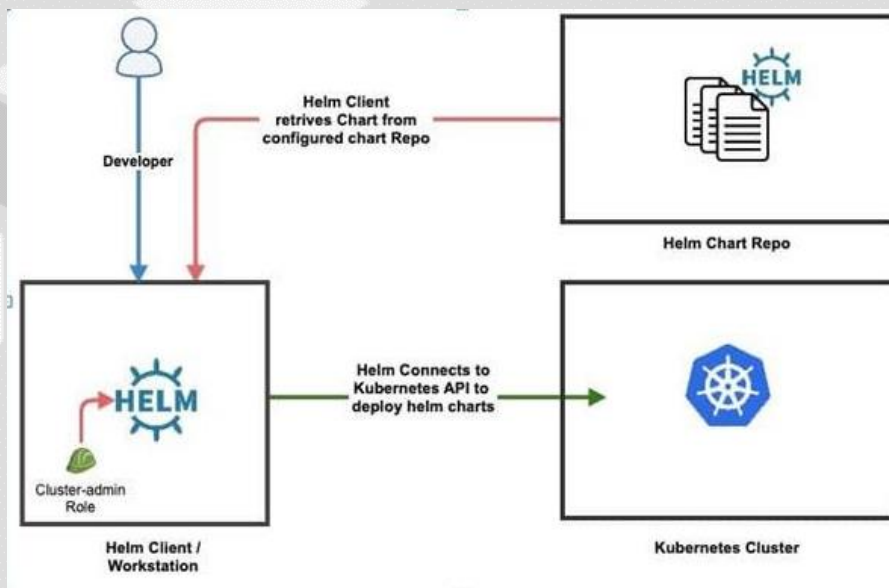
• کمک کردن در

• نصب

• بروزرسانی

• تا حدی شبیه یک پکیج منیجر برای یک زبان برنامه نویسی

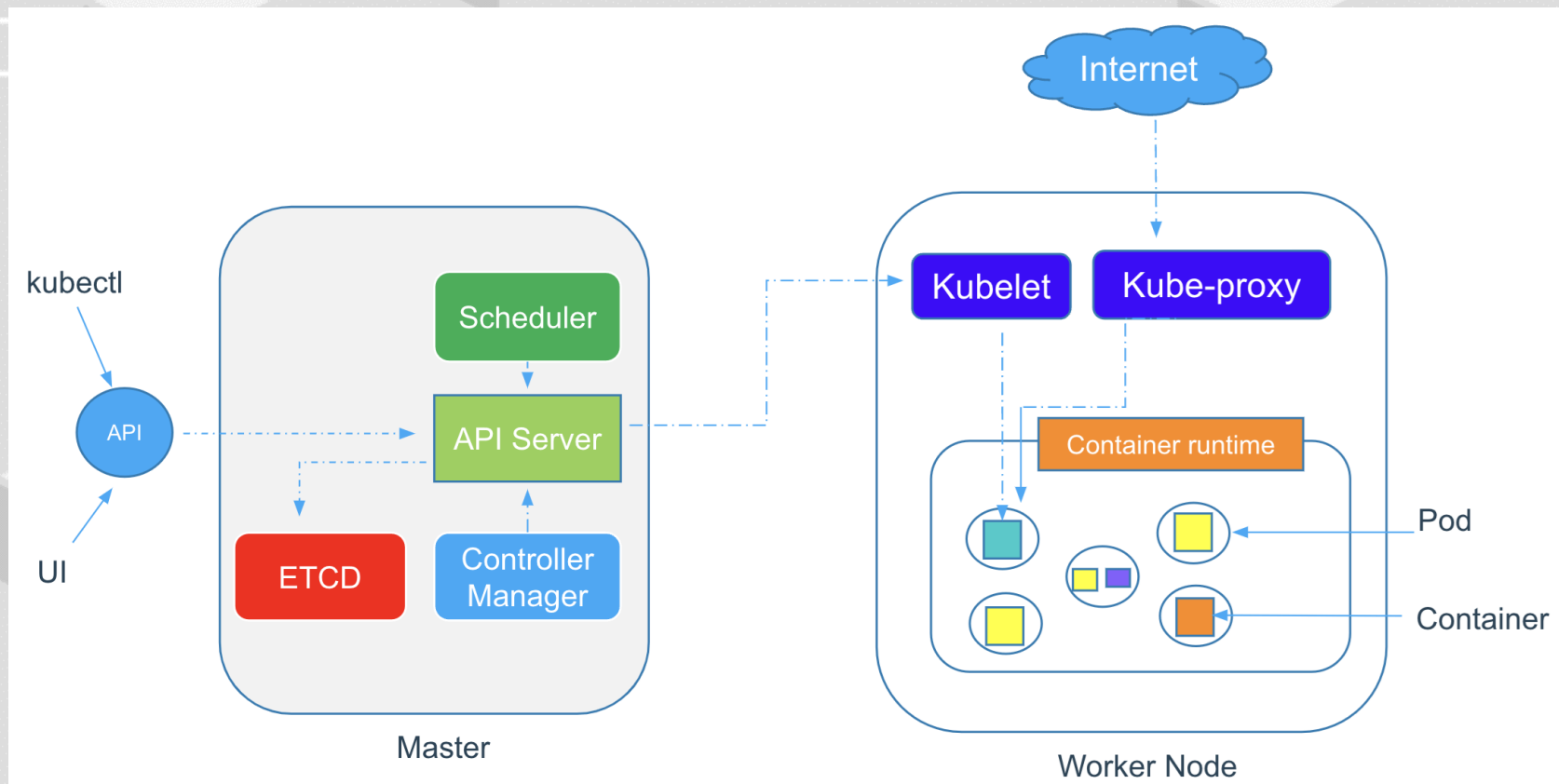
• مثل پپ (pip) برای پایتون



کوبرنتیز - معرفی چند ابزار - کیوب پراکسی

- کیوب پراکسی
 - میرا بودن (Ephemeral) پادها
 - آی پی آدرس غیر قابل اطمینان
 - فراهم کردن یک آی پی آدرس مطمئن برای برقراری ارتباط دو پاد
 - نصب شده در هر نود
 - نحوه کارکرد
 - رصد کردن تمامی تغییرات در سرویس ها و اندپوینت هایشان (endpoints)
 - تصویر کردن (translate) تغییرات در شبکه واقعی داخل نود
 - اجرا در کلاستر عموماً به صورت دیمون ست
 - قابلیت نصب جداگانه در لینوکس

کوبرنتیز - معرفی چند ابزار - کیوب پراکسی

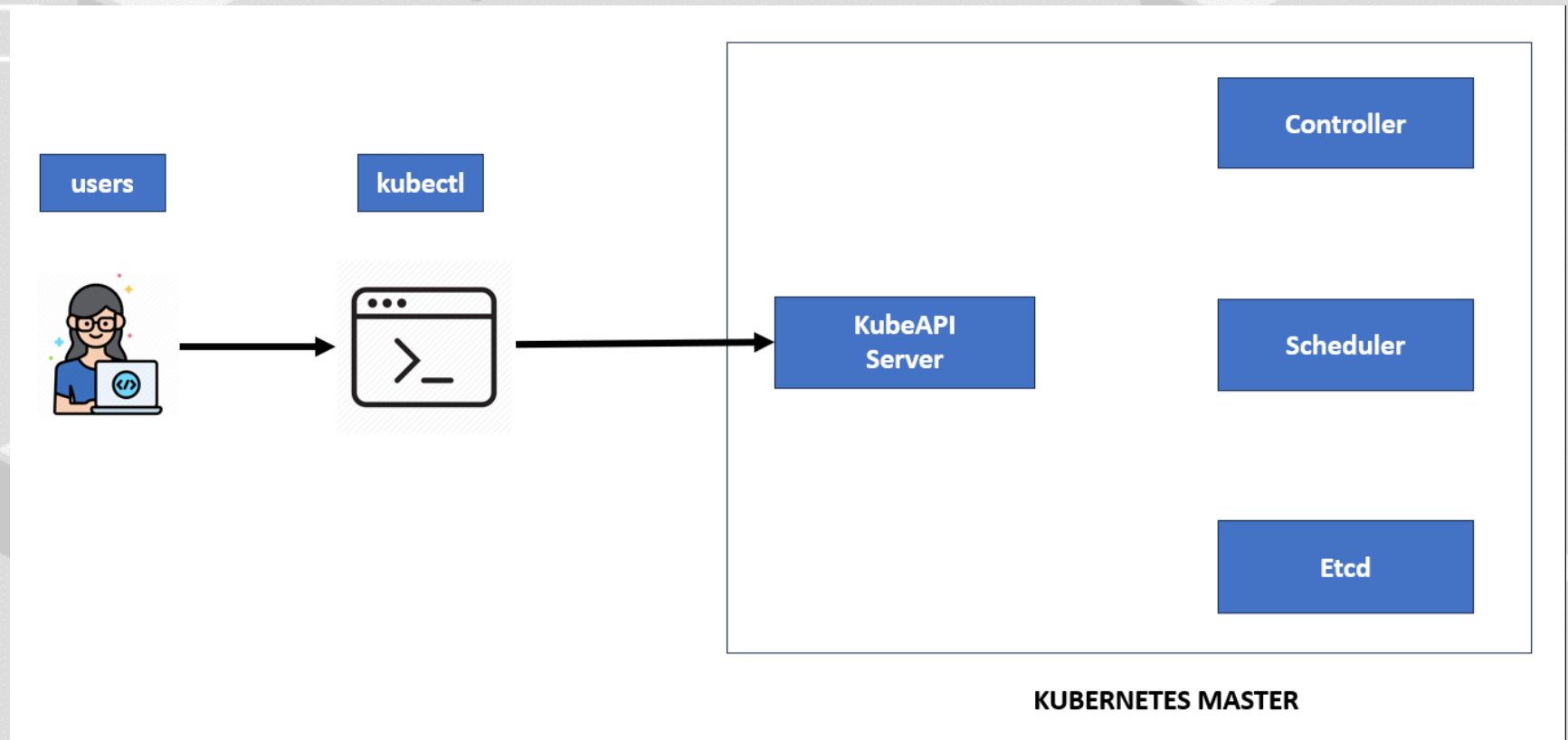




کوبرنتیز - معرفی چند ابزار - کیوبسی تی ال

- کیوبسی تی ال (kubectl)
- ابزار خط اجرا (command line tool)
- اجرا کردن دستورات در کوبرنتیز
- استفاده
- دیپلوی کردن اپلیکیشن
- نظارت و مدیریت منابع کلاستر
- دیدن لاگها
- سایر موارد

کوبرنتیز - معرفی چند ابزار - کیوبسی تی ال



کوبرنتیز - معرفی چند ابزار - کیوب سی تی ال

دستورات کیوب سی تی ال

Kubectl Commands Cheat Sheet



| | | |
|--|--|---|
| <h3 style="color: #007bff; text-align: center;">Pod & Container Introspection</h3> <pre># List the current pods kubectl get pods # Describe pod <name> kubectl describe pod <name> # List the replication controllers kubectl get rc # List the replication controllers in <namespace> kubectl get rc --namespace=<namespace> # Describe replication controller <name> kubectl describe rc <name> # List the services kubectl get svc # Describe service <name> kubectl describe svc <name> # Delete pod <name> kubectl delete pod <name> # Watch nodes continuously kubectl get nodes --w</pre> <h3 style="color: #007bff; text-align: center;">Cluster Introspection</h3> <pre># Get version information kubectl version # Get cluster information kubectl cluster-info # Get the configuration kubectl config view # Output information about a node kubectl describe node <node></pre> | <h3 style="color: #007bff; text-align: center;">Debugging</h3> <pre># Execute <command> on <service> optionally # selecting container <\$container> kubectl exec <service> <command> [-c <\$container>] # Get logs from <service <name> optionally # selecting container <\$container> kubectl logs -f <name> [-c <\$container>] # Watch the Kubelet logs watch -n 2 cat /var/log/kublet.log # Show metrics for nodes kubectl top node # Show metrics for pods kubectl top pod</pre> <h3 style="color: #007bff; text-align: center;">Quick Commands</h3> <pre># Launch a pod called <name> # using image <image-name> kubectl run <name> --image=<image-name> # Create a service described # in <manifest.yaml> kubectl create -f <manifest.yaml> # Scale replication controller # <name> to <count> instances kubectl scale --replicas=<count> rc <name> # Map port <external> to # port <internal> on replication # controller <name> kubectl expose rc <name> --port=<external> --target- port=<internal> # Stop all pods on <n> kubectl drain <n> --delete-local-data --force --ignore- daemonsets # Create namespace <name> kubectl create namespace <namespace> # Allow Kubernetes master nodes to run pods kubectl taint nodes --all node-role.kubernetes.io/master-</pre> | <h3 style="color: #007bff; text-align: center;">Objects</h3> <pre>all clusterrolebindings clusterroles cm = configmaps controllerrevisions crd = customresourcedefinition cronjobs cs = componentstatuses csr = certificatesigningrequests deploy = deployments ds = daemonsets ep = endpoints ev = events hpa = horizontalpodautoscalers ing = ingresses jobs limits = limitranges netpol = networkpolicies no = nodes ns = namespaces pdb = poddisruptionbudgets po = pods podpreset podtemplates psp = podsecuritypolicies pv = persistentvolumes pvc = persistentvolumedaims quota = resourcequotas rc = replicationcontrollers rolebindings roles rs = replicasets sa = serviceaccounts sc = storageclasses secrets sts = statefulsets</pre> |
|--|--|---|

کوبرنتیز - معرفی چند ابزار - مینی کیوب

- مینی کیوب (minikube)

- ابزاری برای نصب و تنظیم محیط کوبرنتیز

- بر روی کامپیوتر شخصی (local PC)

- بر روی لپ‌تاپ

- مثل کوبر در همه چیز به جز

- فقط یک ماشین

- کوبر در چند ماشین

- پیچیدگی بیشتر برای برخی از کارها

- باز کردن پورت

- امکان پذیر

- اضافه کردن `--vm-driver=none` به دستور `minikube start` برای اجرا

- به همراه `sudo` در لینوکس



minikube

• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

- مقدمه
- مقایسه با ماشین مجازی

• داکر

- مقدمات
- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری
- دستورات

• تبدیل برنامه به کانتینر

• کوبرنتیز

- مقدمه
- اجزای اصلی سازنده
- مفاهیم مقدماتی
- امنیت
- معرفی چند ابزار
- فراهم‌کننده سرویس کوبر در بستر ابری

• منابع

کوبرنتیز - فراهم کننده سرویس کوبر در بستر ابری

- برترین ها (پر استفاده ترین ها)

- آمازون وب سرویس (Amazon Web Service - AWS)



- آژور کوبرنتیز (Azure Kubernetes)



- گوگل کلود پلتفرم (Google Kubernetes Platform - GKE)



کوبرنتیز - فراهم کننده سرویس کوبر در بستر ابری

- چندتای دیگر
 - آمازون الاستیک کوبرنتیز (Amazon Elastic Kubernetes)
 - جی کی ای (GKE)
 - دیجیتا اوشن (DigitalOcean)
 - رد هت اُپنشیفت (Red Hat OpenShift)
 - آی بی ام کلود کوبرنتیز سرویس (IBM Cloud Kubernetes Service)
 - علیبابا کلود کانتینر سرویس فور کوبرنتیز (Alibaba Cloud Container Service for Kubernetes)

• ماشین‌های مجازی

- مجازی‌سازی
- هایپروایزر
- نحوه کار کردن هایپروایزر

• کانتینر

- مقدمه
- مقایسه با ماشین مجازی

• داکر

- مقدمات
- اعضای سازنده
- اعضای سازنده هسته داکر
- معماری
- دستورات

• تبدیل برنامه به کانتینر

• کوبرنتیز

- مقدمه
- اجزای اصلی سازنده
- مفاهیم مقدماتی
- امنیت
- معرفی چند ابزار
- فراهم‌کننده سرویس کوبر در بستر ابری

• منابع

منابع

1. سایت رسمی [کوبرنتیز](#)
2. سایت رسمی [داکر](#)
3. سایت [مدیوم](#)
4. سایت [گیگز-فور-گیگز](#)
5. سایت [آی بی ام](#)
6. سایت [پارک پلیس تکنولوژی](#)
7. کانال نانا در [یوتیوب](#)
8. بقیه منابع که تعدادشان زیاد است و در دسترس نیستند.