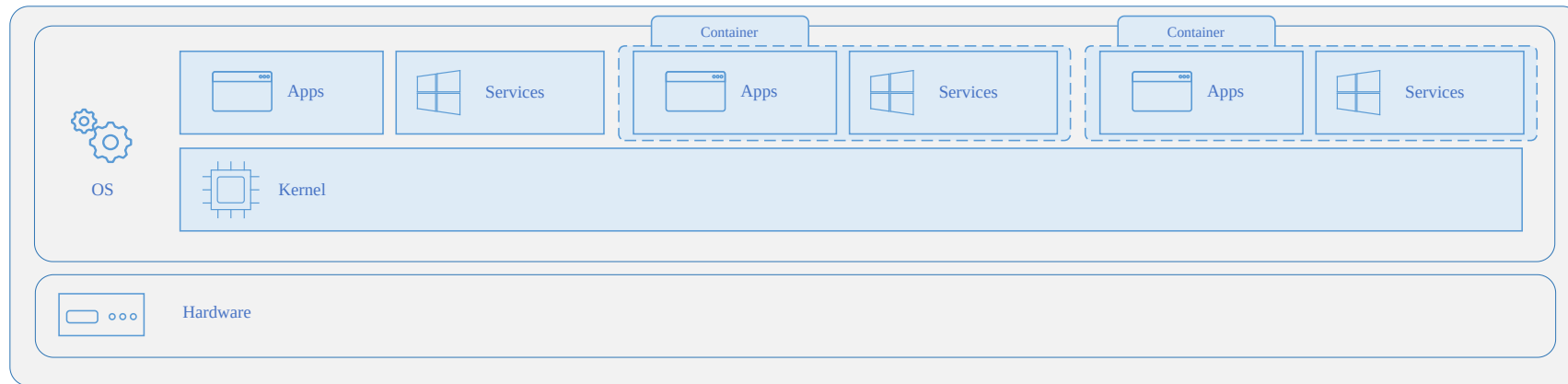
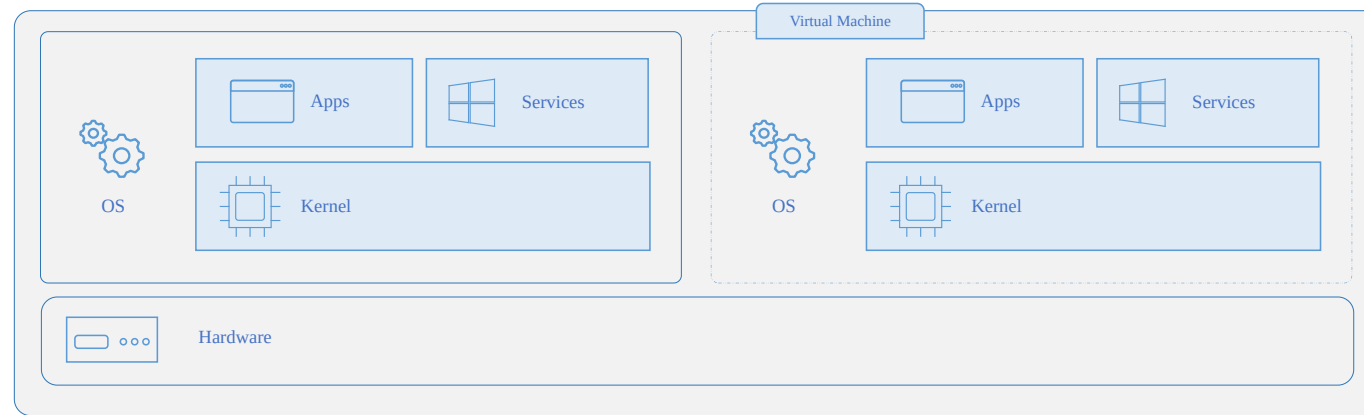


# Containers e docker

# Container vs máquina virtual



# Containers, docker e OCI

- Inspirado a partir de sistemas enjaulados (BSD)
- Depende do control groups (cgroups) do kernel Linux
- Docker foi base para o Open Container Initiative
- Existem outras formas de executar containers do padrão OCI
- Docker-compose: permite a execução de vários containers através de apenas um arquivo YAML

# Entre sistemas operacionais...



0 que containers resolvem?

~\(\ツ)/~

**IT WORKS**  
*on my machine*

# 0 que precisamos para usar um container de Docker?

## Dockerfile

```
FROM alpine:3.23.2

WORKDIR /arquivos

RUN apk add python3

CMD ["sh"]
```

## docker-compose.yaml

```
services:
  teste:
    build:
      context: .
      dockerfile: Dockerfile
    image: cc7261:teste
    volumes:
      - ./arquivos
```

docker-compose.yaml é opcional

# Persistência dos dados

- Um Dockerfile gera uma imagem
- Imagens são imutáveis depois de criadas
- Container é uma instância de uma imagem
- Podemos montar volumes no container para armazenar arquivos/alterações

# Para desenvolvedores

- Devcontainers
  - VSCode
  - JetBrains IDEs
  - devcontainers cli
- Red Hat Toolbox e Distrobox
- Distribuições Linux
  - Fedora: CoreOS -> Atomic Desktop
  - openSUSE: MicroOS -> Aeon/Kalpa
  - Ubuntu: Core + Snapd



**IT WORKS**  
*on my machine*



# Para a aula

- Disciplina se chama Sistemas Distribuídos
- Precisamos de mais de uma máquina para ser distribuído
- Vamos usar containers para executar processos como se fossem em máquinas diferentes
- Implementamos os processos ("containers") e o docker cuida da conexão entre eles