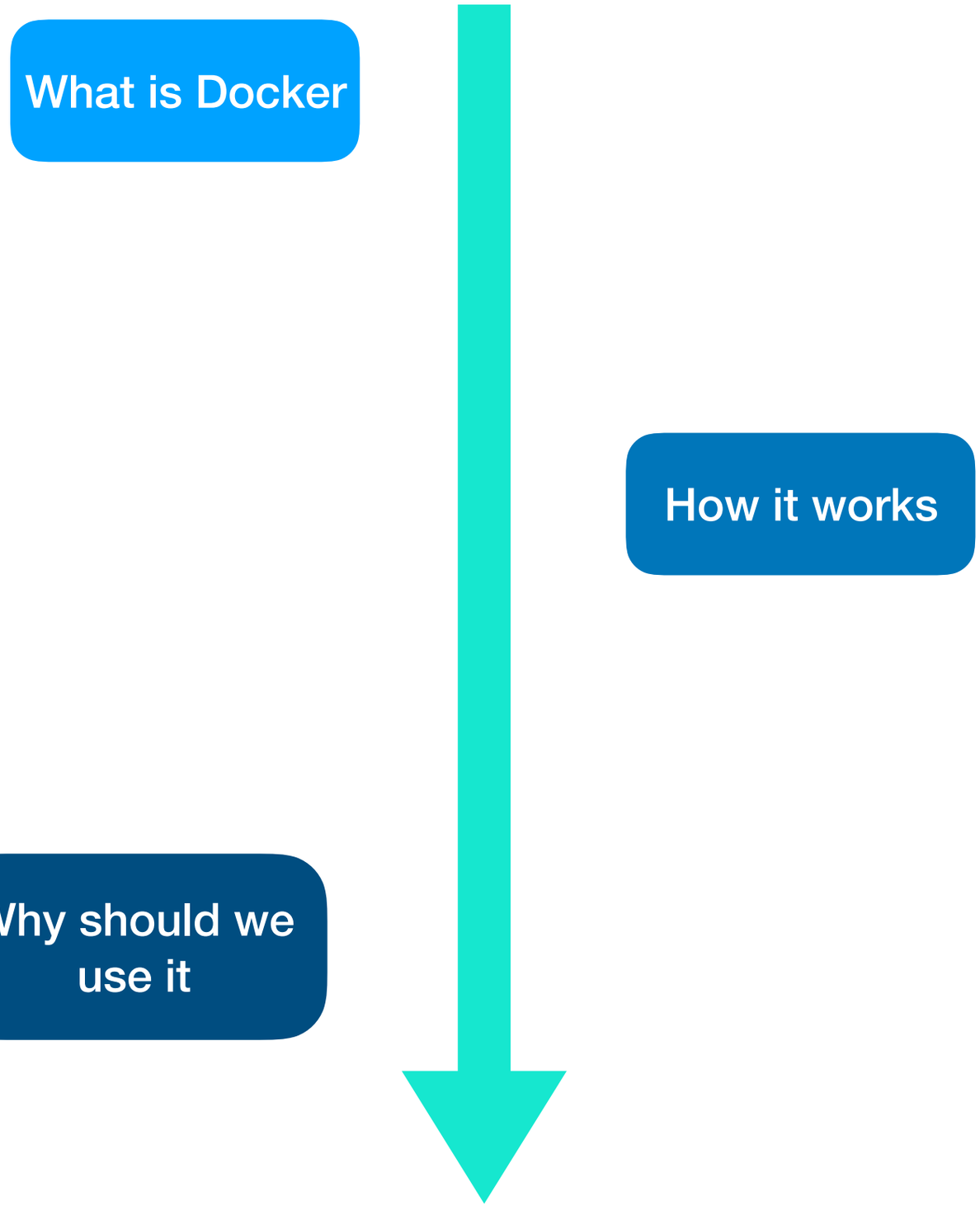
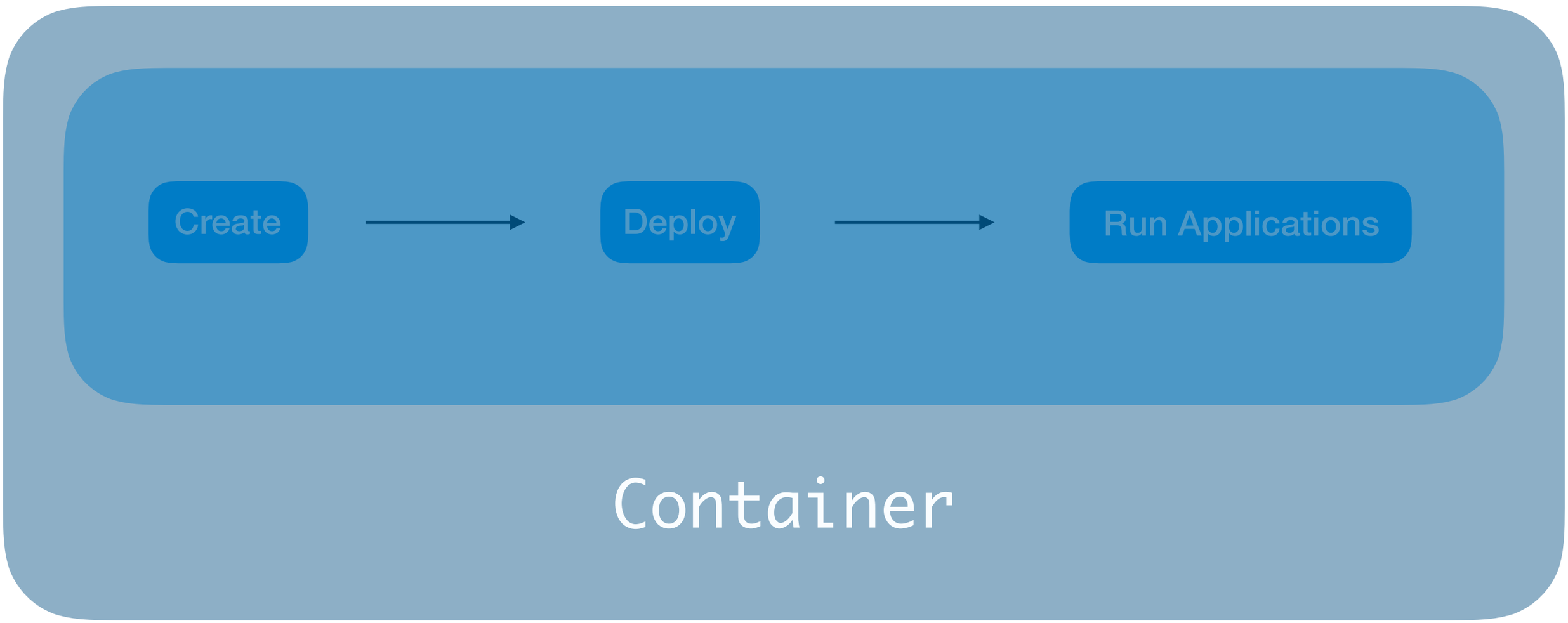
Docker

The idea of “containerization”

Roadmap



What is docker?

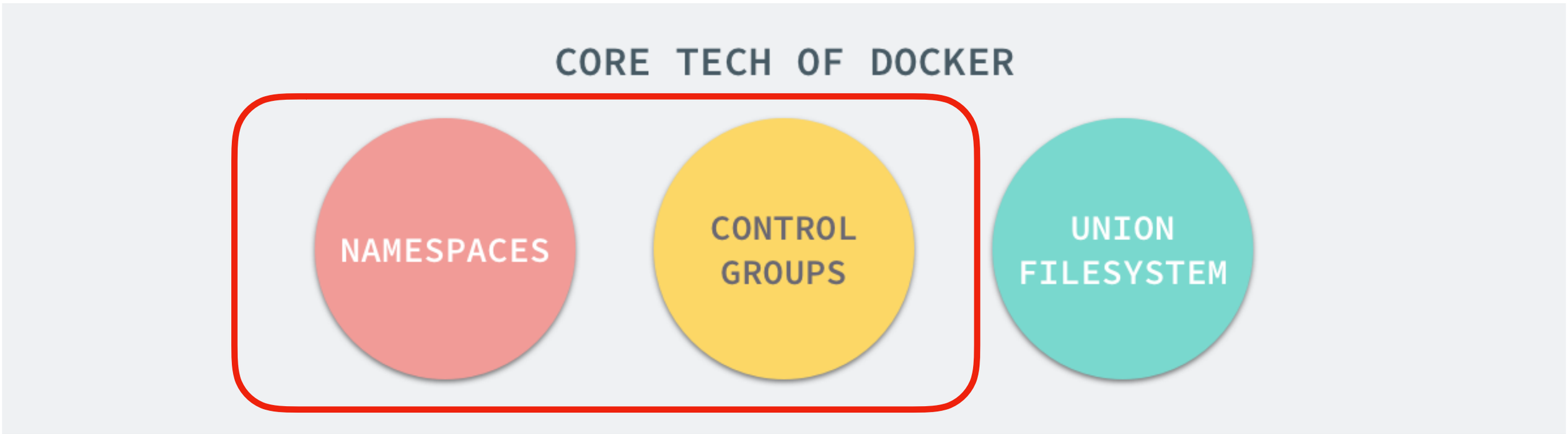


Container?

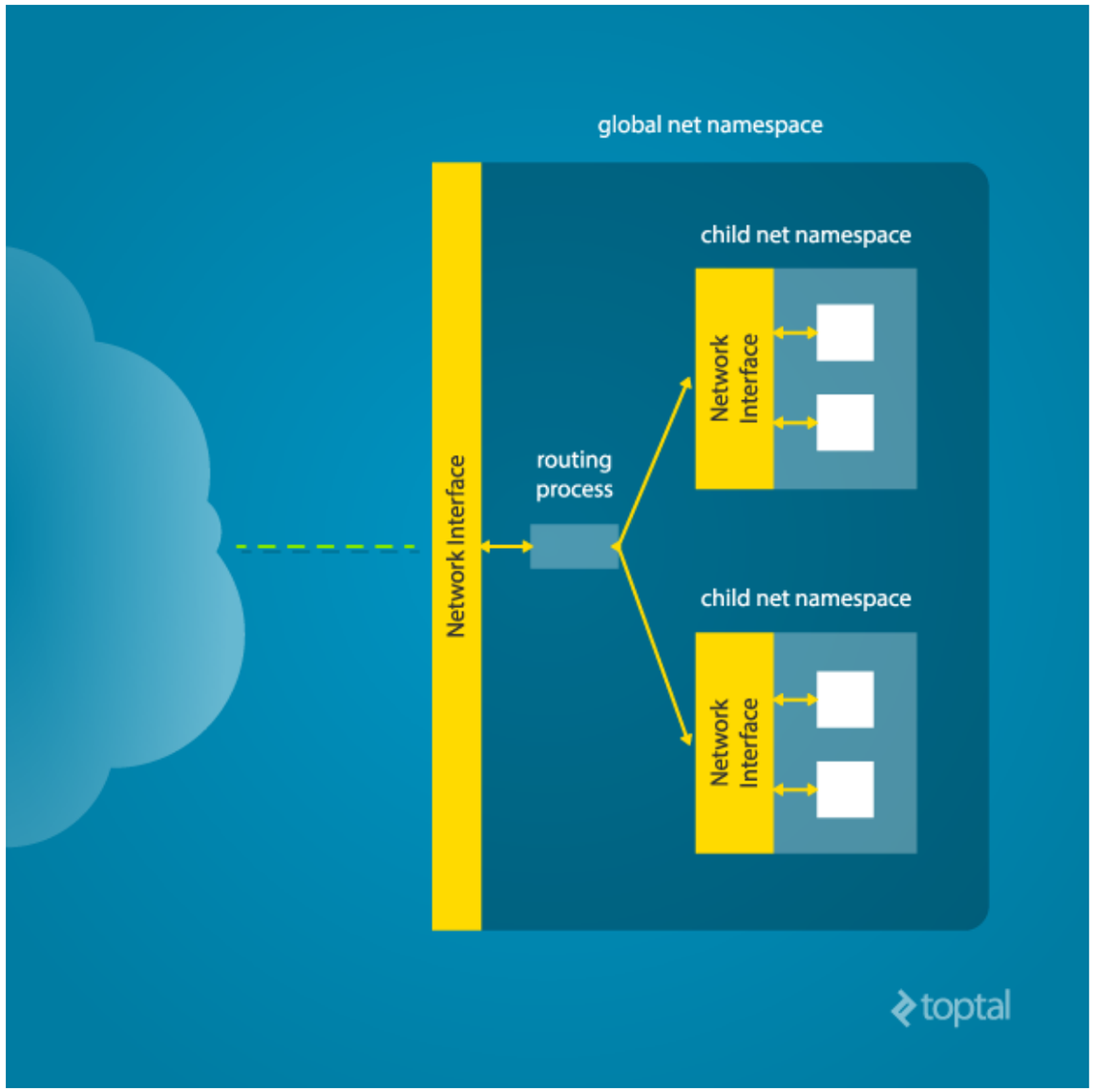
Contain applications in a way that keep them isolated from the host system that they run on.

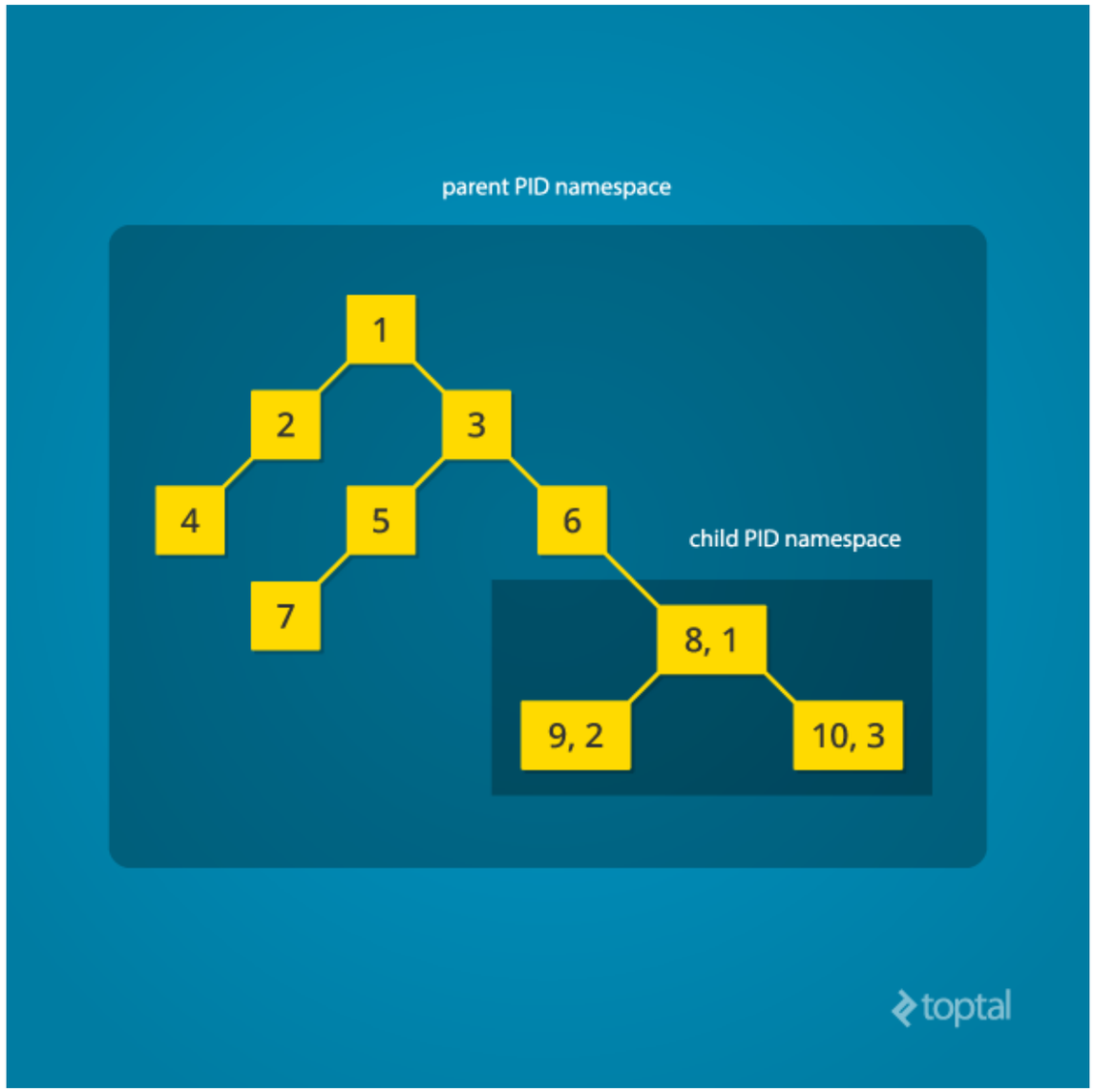
It packages up libraries, dependencies, etc.

To the outside world, the can look like their own complete system.


CORE TECH OF DOCKER
NAMESPACES
CONTROL GROUPS
UNION FILESYSTEM

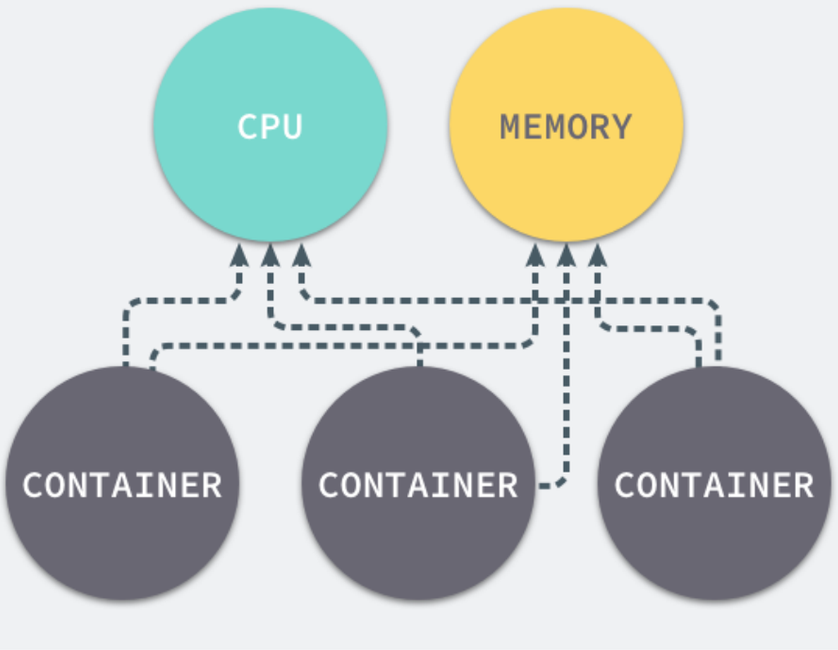
Namespaces





Only processes within a group knows the existence of each other.

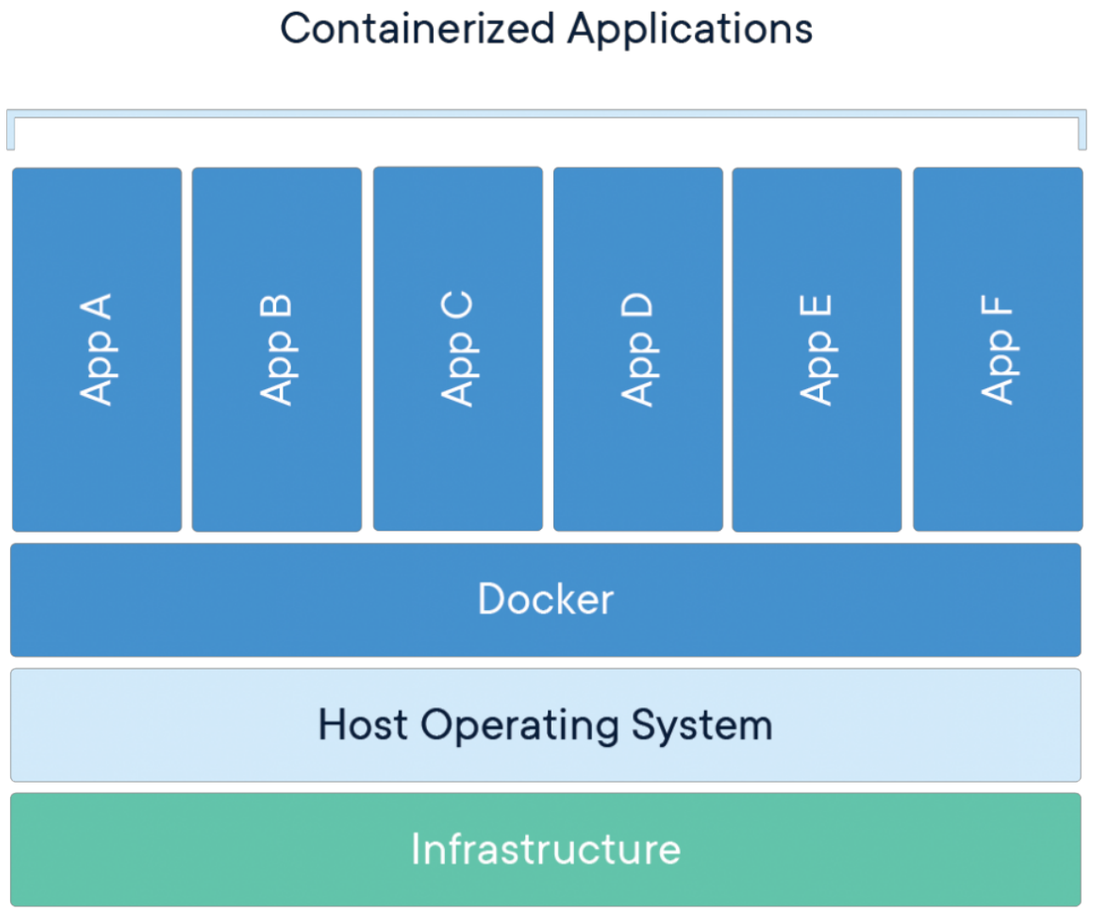
Control Groups

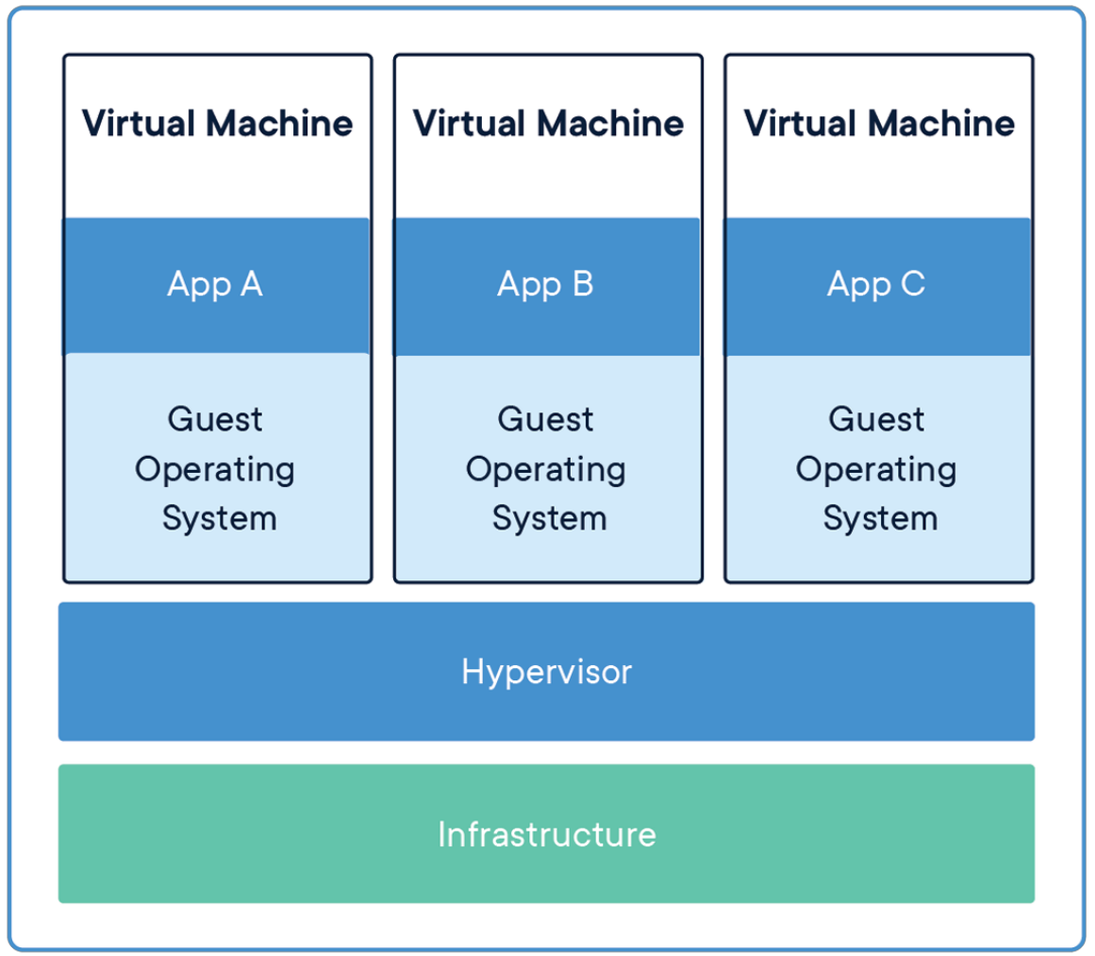


Containers know nothing about each other, but they are sharing the same physical resources.

Control groups isolates physical resources, such as CPU, memory, disk I/O

Virtual Machine?



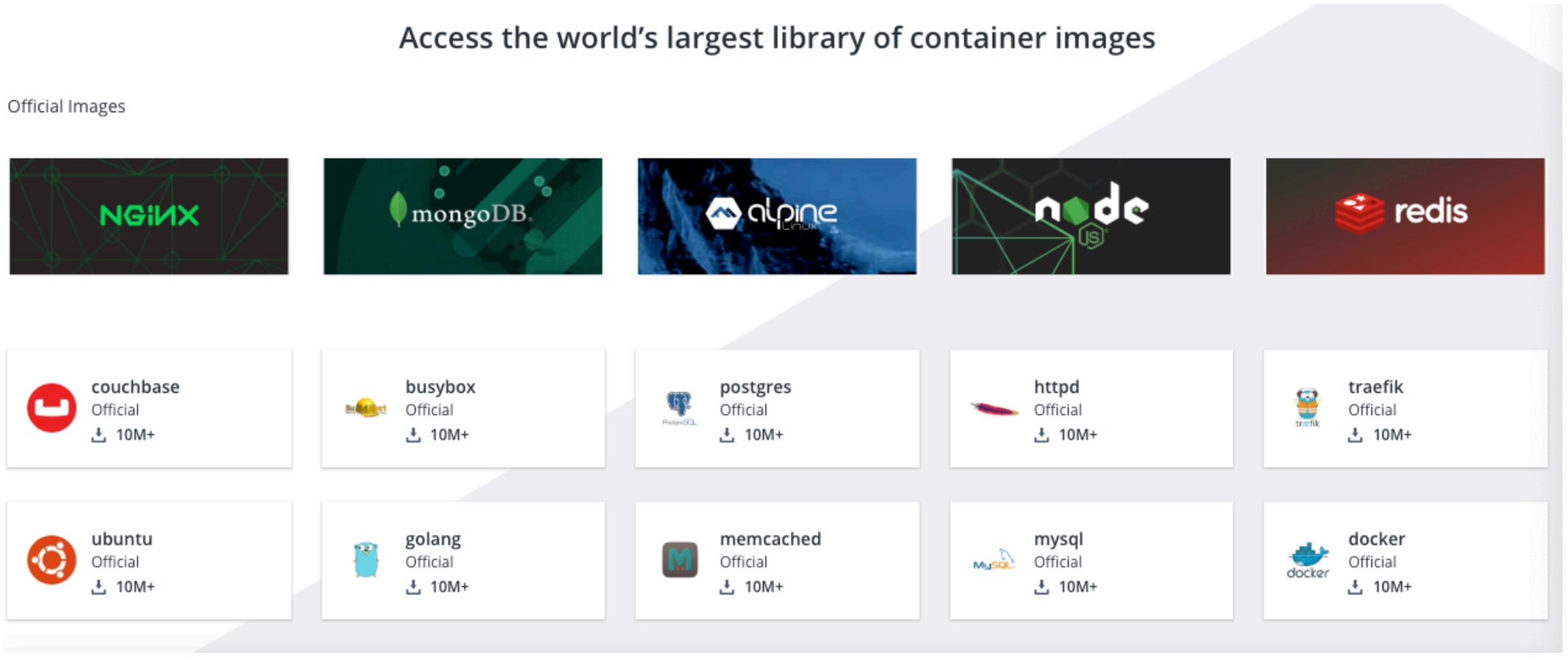


By contrast, container gives performance boost and reduces the size of the application.

Why docker?

Shipping code to the server is too hard

Instead, we can use containers by images, which are templates for containers.



Dockerfile

then, just **docker-compose up**

Compose file

```
services:
  db:
    container_name:test-db
    image: postgres:9.6.3
    volumes:
      - ./data/postgresql:/var/lib/postgresql/data
    env_file:
      - .env
    restart: on-failure

  redis:
    container_name: test-redis
    image: redis:5.0
    volumes:
    - ./data/postgresql:/var/lib/postgresql/data
    env_file:
    - .env
    restart: on-failure

    ……
```

“Developers worried about inside the box,
Infrastructurers worried about outside the box”

Start using Docker today!

1. Docker introduces the idea of containers by utilising OS-level support

2. More efficient and better performance compared to virtual machines

3. Separate the concern of DevOps and Development.