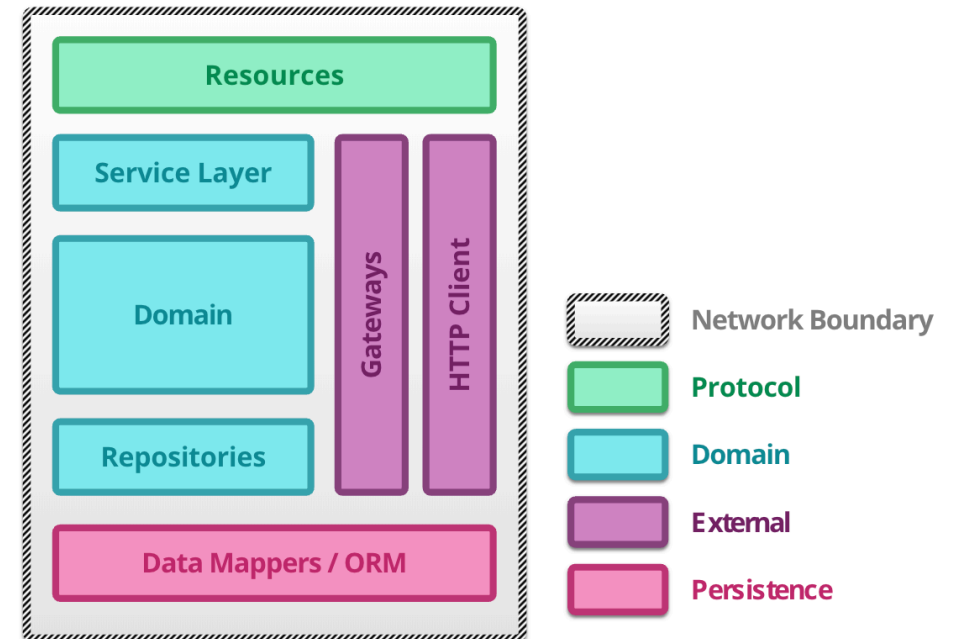
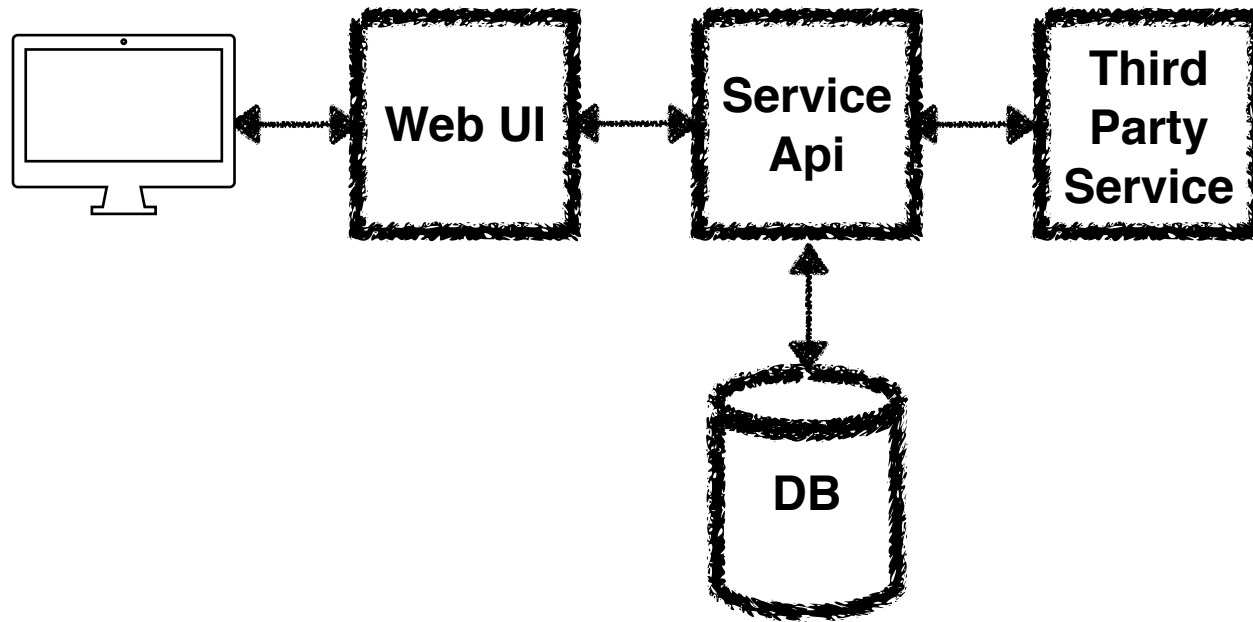


Docker For Testers



High-level Structure / Internal Structure



reference: <https://martinfowler.com/articles/microservice-testing/>



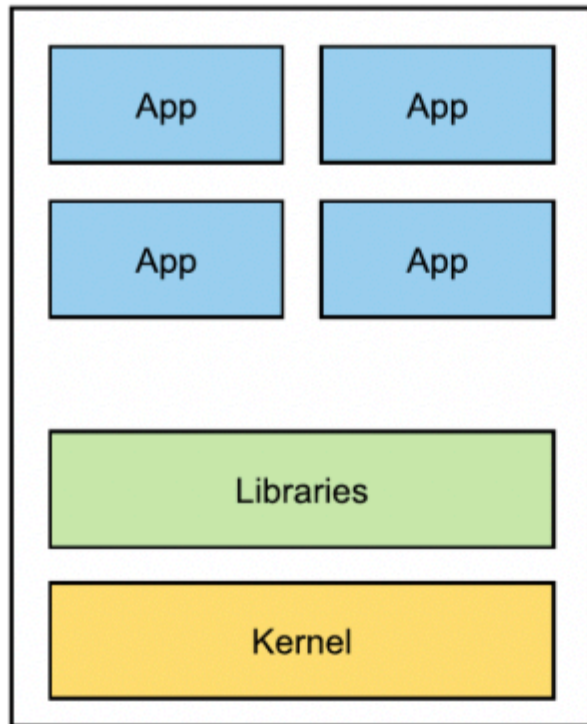
Practical Test Pyramid

- Unit
- Integration
- Contract
- UI/API*
- End-to-End*
- Acceptance*
- Exploratory
- The Confusion About Testing Terminology
- Putting Tests Into Your Deployment Pipeline
- Avoid Test Duplication
- Writing Clean Test Code

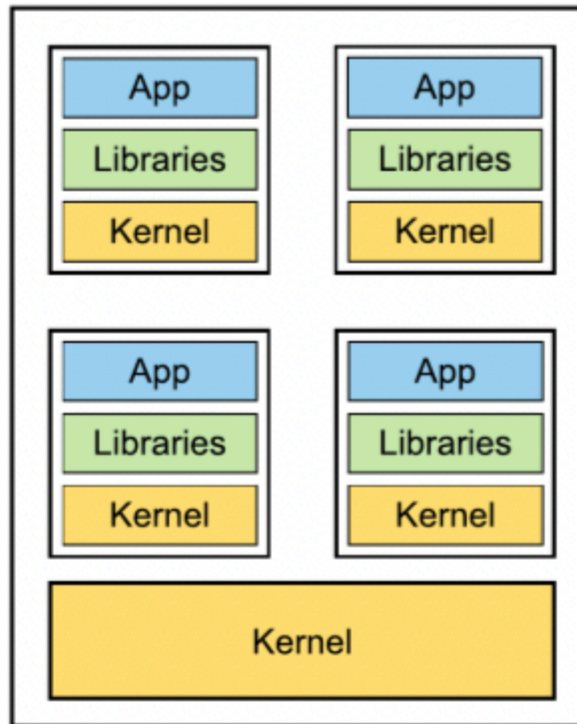
reference: <https://martinfowler.com/articles/practical-test-pyramid.html>



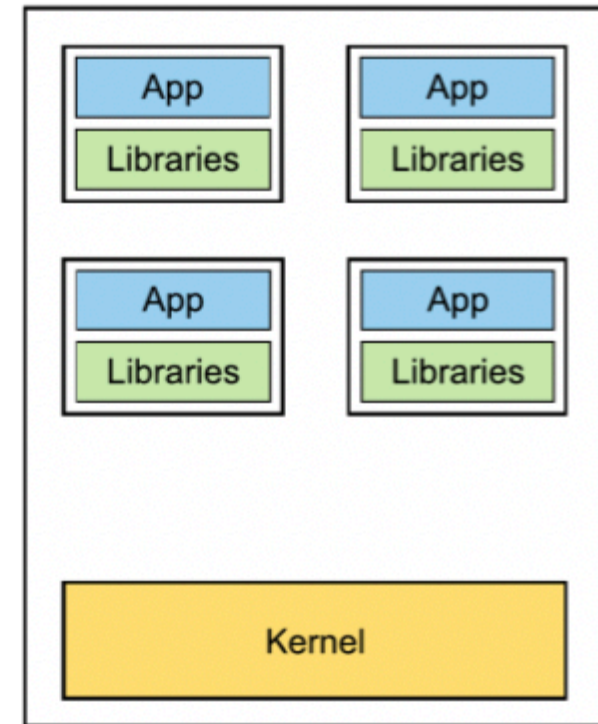
Evolution of Containers



Shared Machine



VMs



Containers

reference: Getting Started with Kubernetes



Several Specific Benefits of Containers

- Language Flexibility
- Isolation Without Overhead: light weight
- Developer Efficiency: Isolating Dependencies(libs, configuration)
- **Reproducibility**: Containers make it easier to reproduce your application environment.



Install Docker and Editor

1. [get-docker](#)
2. [install windows home](#)
[WSL2](#) must be installed before you can install and use Docker.
3. Install VSCode & Docker Plugin



Docker Desktop for Mac

A native application using the macOS sandbox security model which delivers all Docker tools to your Mac.



Docker Desktop for

Windows

A native Windows application which delivers all Docker tools to your Windows computer.



Docker for Linux

Install Docker on a computer which already has a Linux distribution installed.



Hello, World Docker

```
docker run hello-world
```

Docker run: behind the scenes

- Docker looks for the image on this computer
- Is it installed
- Docker searches Docker Hub for the image.
- Is it on Docker Hub
- Docker downloads the image
- The image layers are installed on the computer
- Docker create a new container and starts the program.
- The container is running.





Lab01 Create Docker Image

from Scratch

- Modify, Commit, and Tag
- Sign Up <https://hub.docker.com>
- Push Image to hub.docker.com
- Pull and Run

Run Docker

```
$ docker run hello-world
```

```
$ docker container run hello-world
```

```
$ docker image ls
```

```
$ docker container ls -a
```

Run with -d

```
$ docker container run boyone/hello-world
```

```
$ docker container run -d boyone/hello-world
```

```
$ docker container ls
```

```
$ docker container logs <container id | container name>
```

```
$ docker container logs -f <container id | container name>
```

Run with --name

```
$ docker container run boyone/hello-world
```

```
$ docker container run -d --name hello boyone/hello-world
```

```
$ docker container ls
```

```
$ docker container logs hello
```

```
$ docker container logs -f hello
```

Run

start, stop, and remove

```
$ docker container stop <container id| container name>
```

```
$ docker container stop hello
```

```
$ docker container start hello
```

```
$ docker container logs -f hello
```

```
$ docker container start hello
```

```
$ docker container rm <container id| container name>
```

```
$ docker container rm hello
```

Run commit

```
$ docker container run -d --name hello boyone/hello-world
```

```
# copy new binary to container
```

```
$ docker container cp ./hello <container id| container name>:/root/
```

```
$ docker container cp ./hello hello:/root/
```

```
# commit container
```

```
$ docker container commit <container id| container name>  
image_name:tag
```

```
$ docker container commit hello boyone/hello-world:0.0.2
```

Run push

Sign Up <https://hub.docker.com>

```
# login to hub.docker.com
```

```
$ docker login
```

```
# push image to hub.docker.com
```

```
$ docker image push username/image_name:tag
```

```
$ docker image push boyone/hello-world:0.0.2
```

```
# tag latest
```

```
$ docker image tag boyone/hello-world:0.0.2  
boyone/hello-world:latest
```

Run

remove image

```
# login to hub.docker.com
$ docker image rm image_name
$ docker image rm boyone/hello-world
$ docker image rm boyone/hello-world:0.0.2

$ docker image ls

$ docker container run boyone/hello-world
$ docker container run boyone/hello-world:0.0.2
```