Agenda

- High Level
- Image Inspect
- Dockerfile
- Registry API
- Manifest - Demo
- Config - Demo
- Layers - Demo
- Recap

Understanding the Design of Docker Images

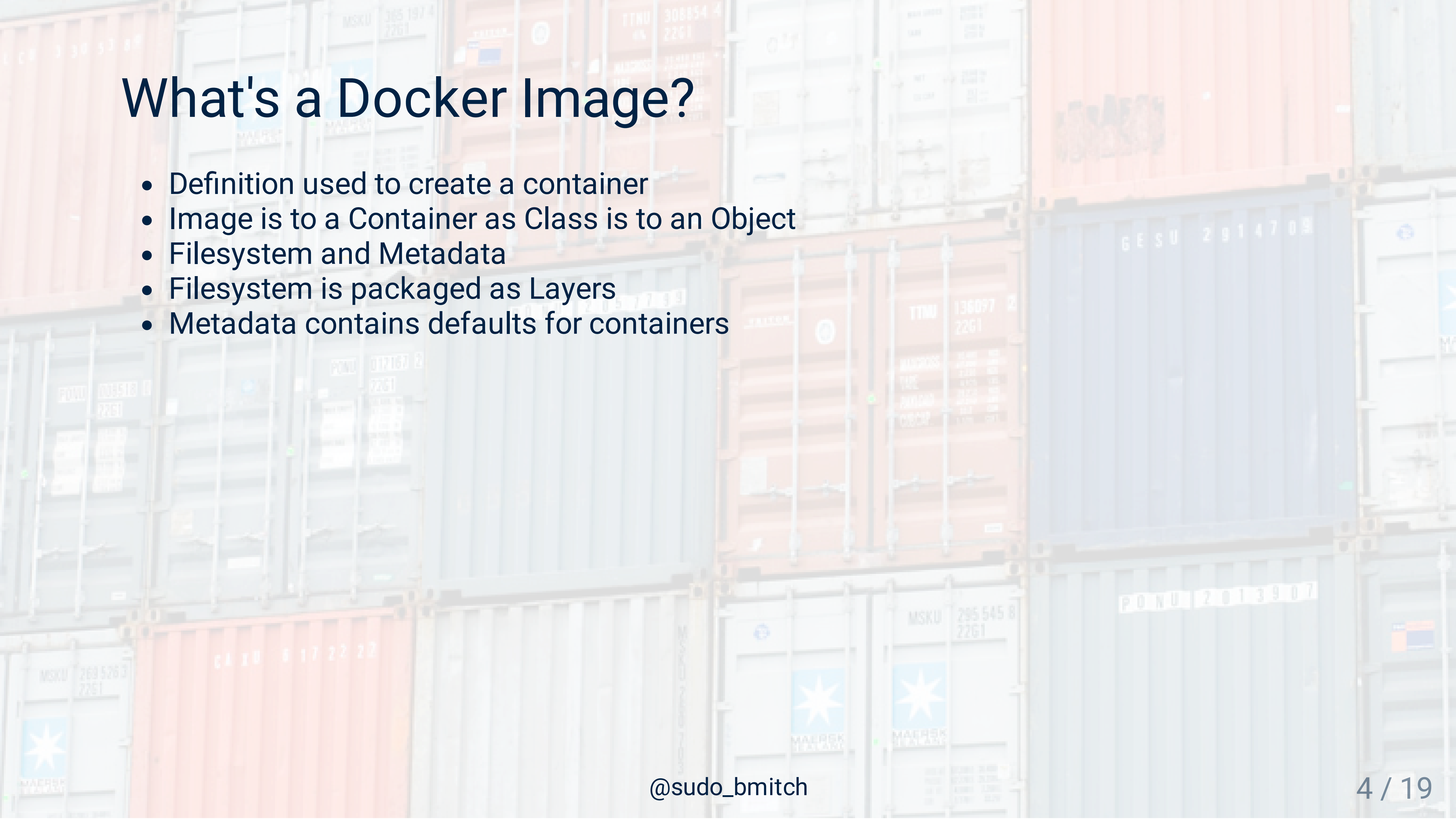
Brandon Mitchell
Twitter: @sudo_bmitch
GitHub: sudo-bmitch

```
$ whoami
- Solutions Architect @ BoxBoat
- Docker Captain
- Frequenter of StackOverflow
```

boxboat

docker CAPTAINS

stackoverflow



What's a Docker Image?

- Definition used to create a container
- Image is to a Container as Class is to an Object
- Filesystem and Metadata
- Filesystem is packaged as Layers
- Metadata contains defaults for containers

Image Inspect

```
$ docker image inspect nginx:latest
[
    {
        "Id": "sha256:35c43ace9216212c0f0e546a65eec93fa9fc8e96b25880ee222b7ed2ca1d2151",
        "RepoTags": [
            "nginx:latest"
        ],
        "RepoDigests": [
            "nginx@sha256:f3693fe50d5b1df1ecd315d54813a77afd56b0245a404055a946574deb6b34fc"
        ],
        "Architecture": "amd64",
        "Os": "linux",
        "Created": "2021-02-17T19:20:26.056161789Z",
        "Container": "51caab8ab3beefe704d8b6fbe7653b2c7df216cd44787aca26bc5c6a2e2e9414",
        "ContainerConfig": {
            "Hostname": "51caab8ab3be",
            "Domainname": "",
            "User": "",
...
```

Image Inspect

```
        "Config": {
            "User": "",
            "ExposedPorts": {
                "80/tcp": {}
            },
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
                "NGINX_VERSION=1.19.7",
                "NJS_VERSION=0.5.1",
                "PKG_RELEASE=1~buster"
            ],
            "Cmd": [
                "nginx",
                "-g",
                "daemon off;"
            ],
            "Volumes": null,
            "WorkingDir": "",
            "Entrypoint": [
                "/docker-entrypoint.sh"
            ],
        },
```

Image Inspect

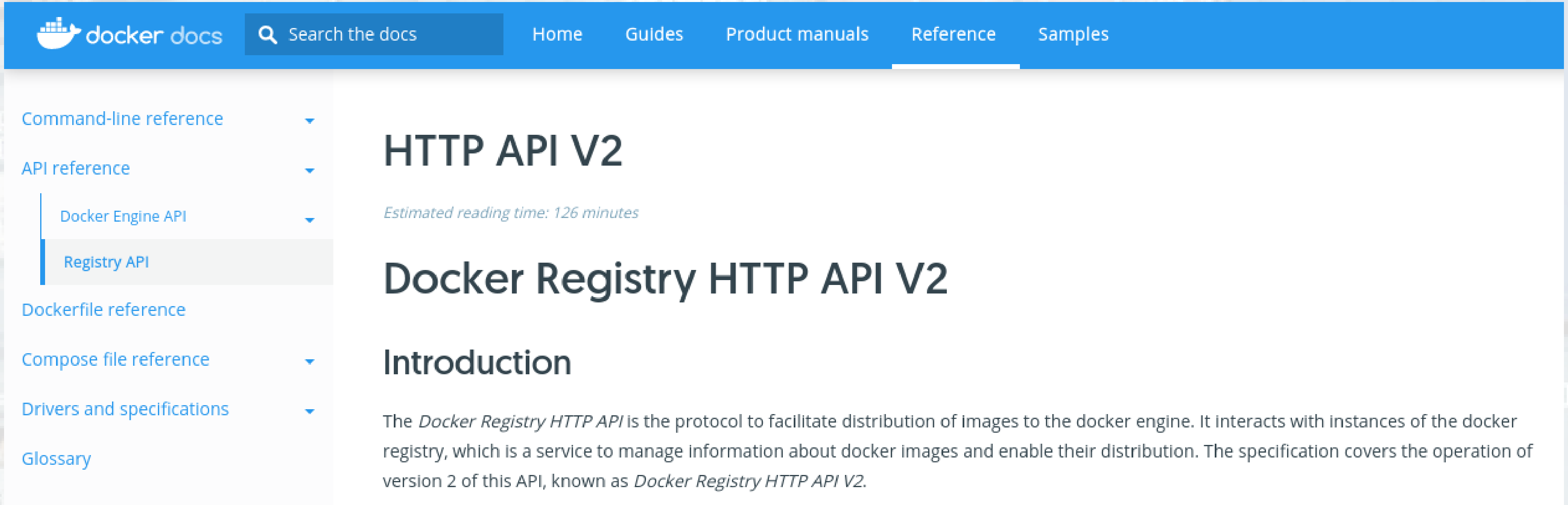
```
        "GraphDriver": {
            "Data": {
                "LowerDir": "/home/docker/overlay2/3e6b61bb8630114472d0e3a97d1685452731f194ea74bf77c
                "MergedDir": "/home/docker/overlay2/0f31e0d639be30f014498cc125e979fe9a9ed6a953bfce09
                "UpperDir": "/home/docker/overlay2/0f31e0d639be30f014498cc125e979fe9a9ed6a953bfce092
                "WorkDir": "/home/docker/overlay2/0f31e0d639be30f014498cc125e979fe9a9ed6a953bfce092c
            },
            "Name": "overlay2"
        },
        "RootFS": {
            "Type": "layers",
            "Layers": [
                "sha256:9eb82f04c782ef3f5ca25911e60d75e441ce0fe82e49f0dbf02c81a3161d1300",
                "sha256:ffd3d6313c9b2905d7cbb9011d8a4fbb0b67d6f334a194ab2f70cf0894b7f4f7",
                "sha256:9b23c8e1e6f9c47ceb8dc619d7f63d659369dedf159e0f9735478c0e41febcc6",
                "sha256:0f804d36244da89e06210c5424c217c9ad838ce91300de213fc25d10fce53c6a",
                "sha256:9f65d1d4c869b7f2910d4c30bb2eade88975529b363089b5a27821fd4c3ff1c7",
                "sha256:2acf82036f3853e985033bab258dabbeed795a98198263fc6a327ca4d1744f0d"
            ]
        },
```

Dockerfile

```
FROM debian:buster-slim

LABEL maintainer="NGINX Docker Maintainers <docker-maint@nginx.com>"
ENV NGINX_VERSION   1.19.7
ENV NJS_VERSION     0.5.1
ENV PKG_RELEASE     1~buster

RUN set -x \
    ...

COPY docker-entrypoint.sh /
COPY 10-listen-on-ipv6-by-default.sh /docker-entrypoint.d
COPY 20-envsubst-on-templates.sh /docker-entrypoint.d
COPY 30-tune-worker-processes.sh /docker-entrypoint.d

ENTRYPOINT ["/docker-entrypoint.sh"]
EXPOSE 80
STOPSIGNAL SIGQUIT
CMD ["nginx", "-g", "daemon off;"]
```

Registries

- Images are designed to be portable
- Registry API defines shipping images
- Manifests, Blobs, Digests



HTTP API V2

Estimated reading time: 126 minutes

Docker Registry HTTP API V2

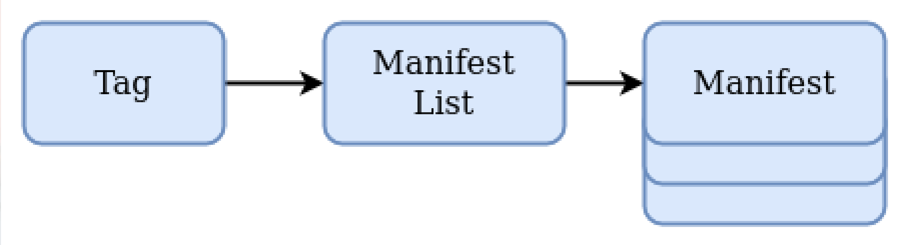
Introduction

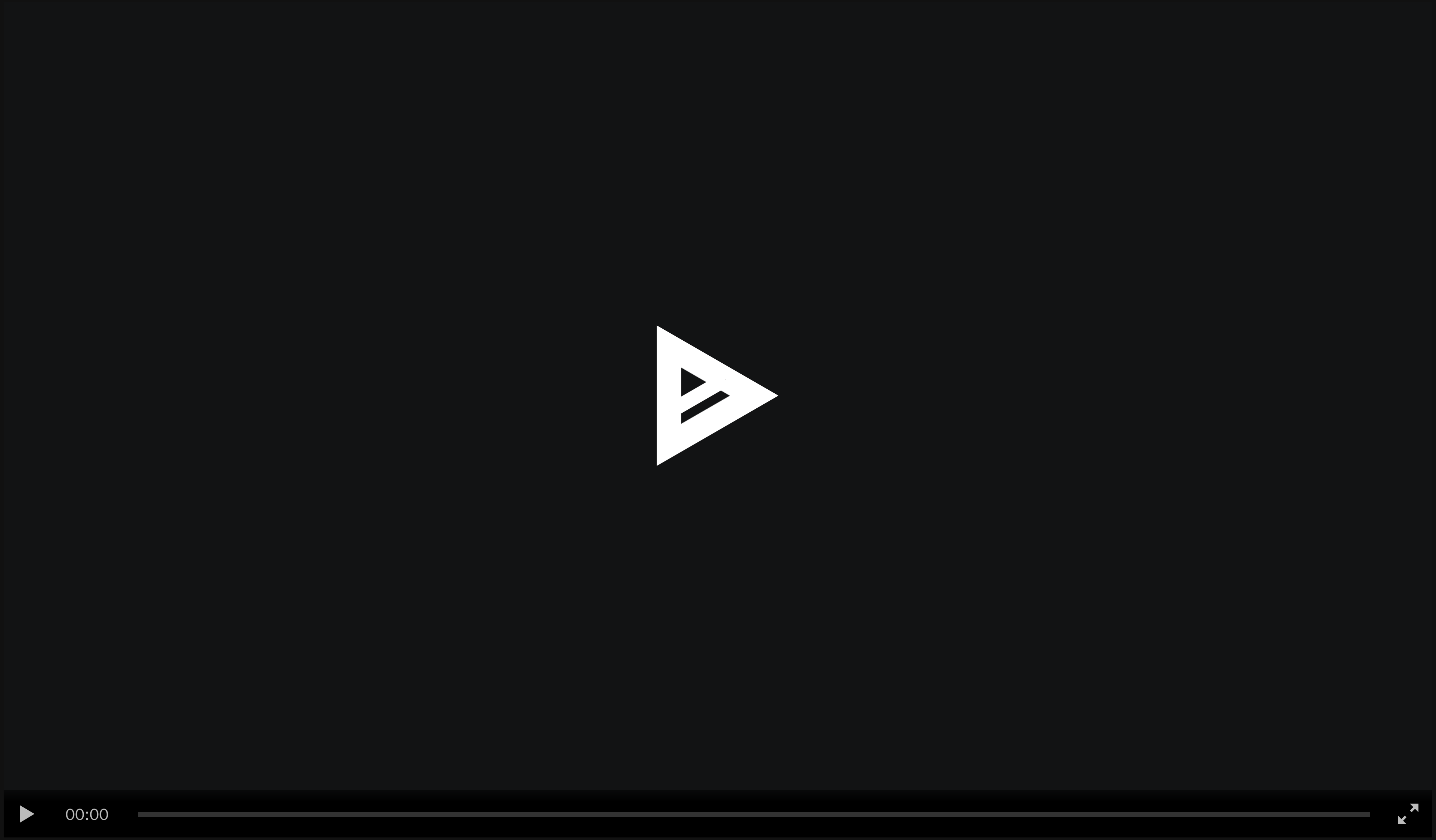
The *Docker Registry HTTP API* is the protocol to facilitate distribution of images to the docker engine. It interacts with instances of the docker registry, which is a service to manage information about docker images and enable their distribution. The specification covers the operation of version 2 of this API, known as *Docker Registry HTTP API V2*.

Definitions

Reference:	`my-registry-server.com/project-x/image-y:tag-z`
Registry:	`my-registry-server.com`
Repository:	`my-registry-server.com/project-x/image-y`
Tag:	`tag-z`

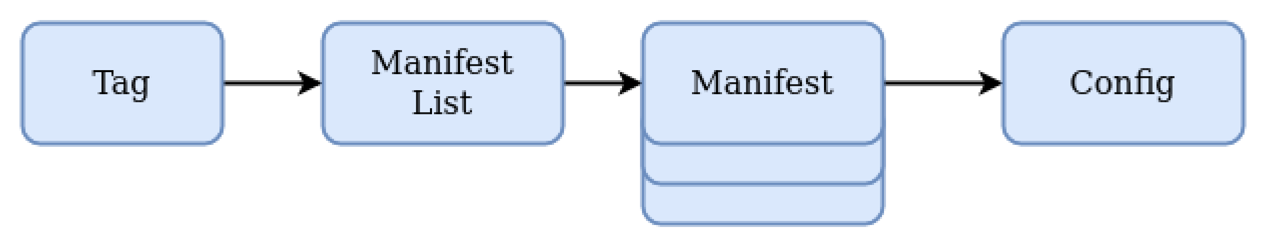
Manifest

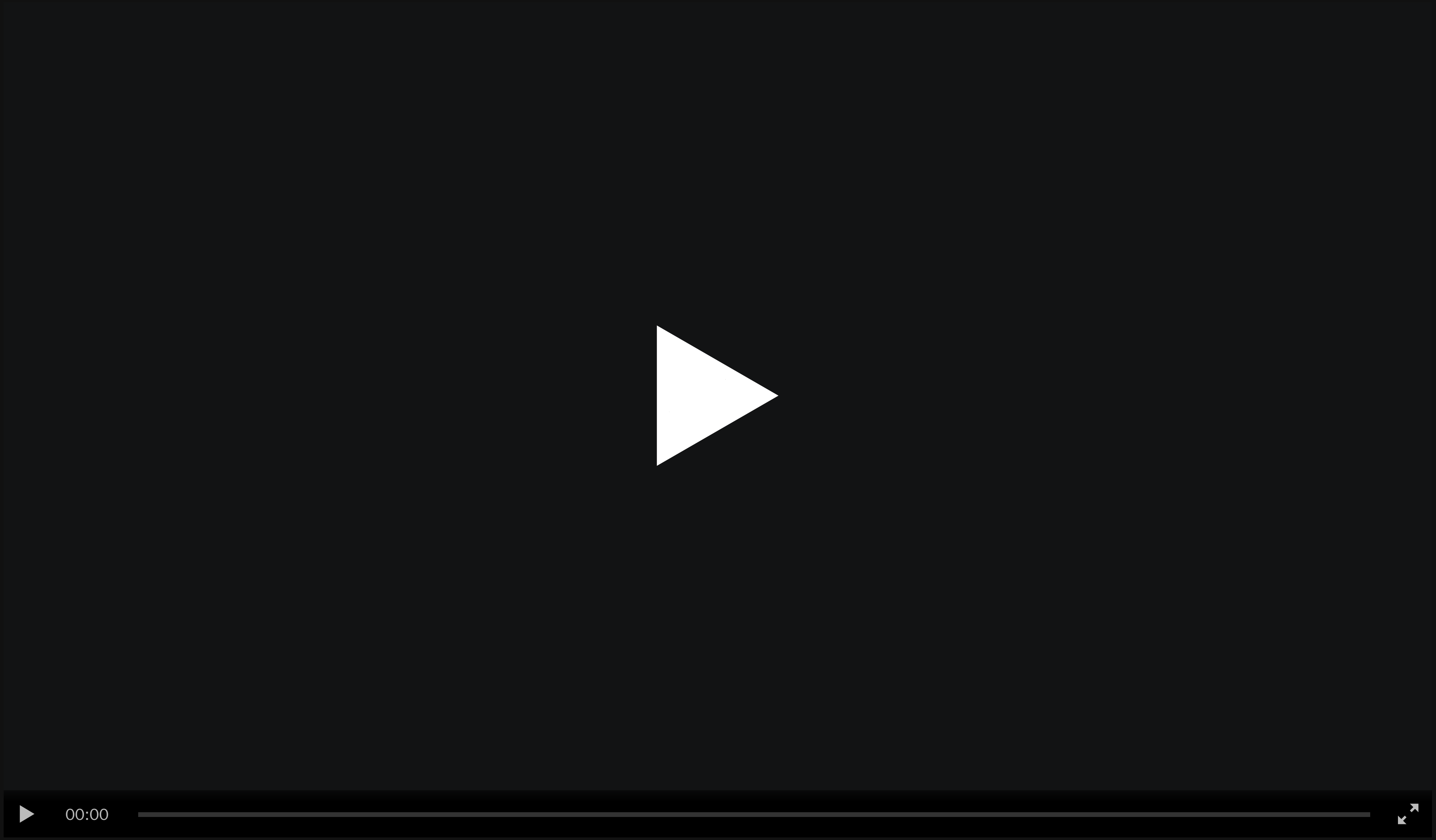




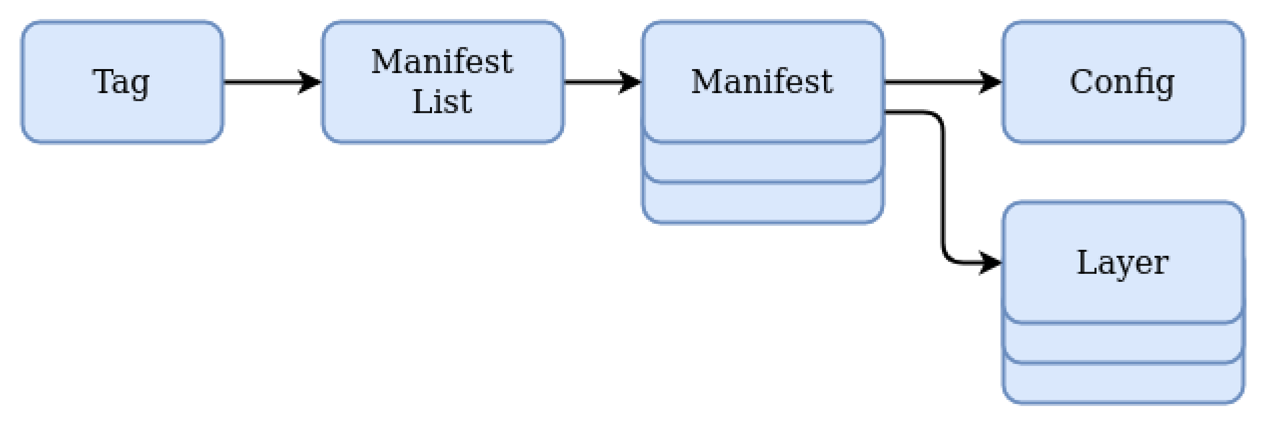
00:00

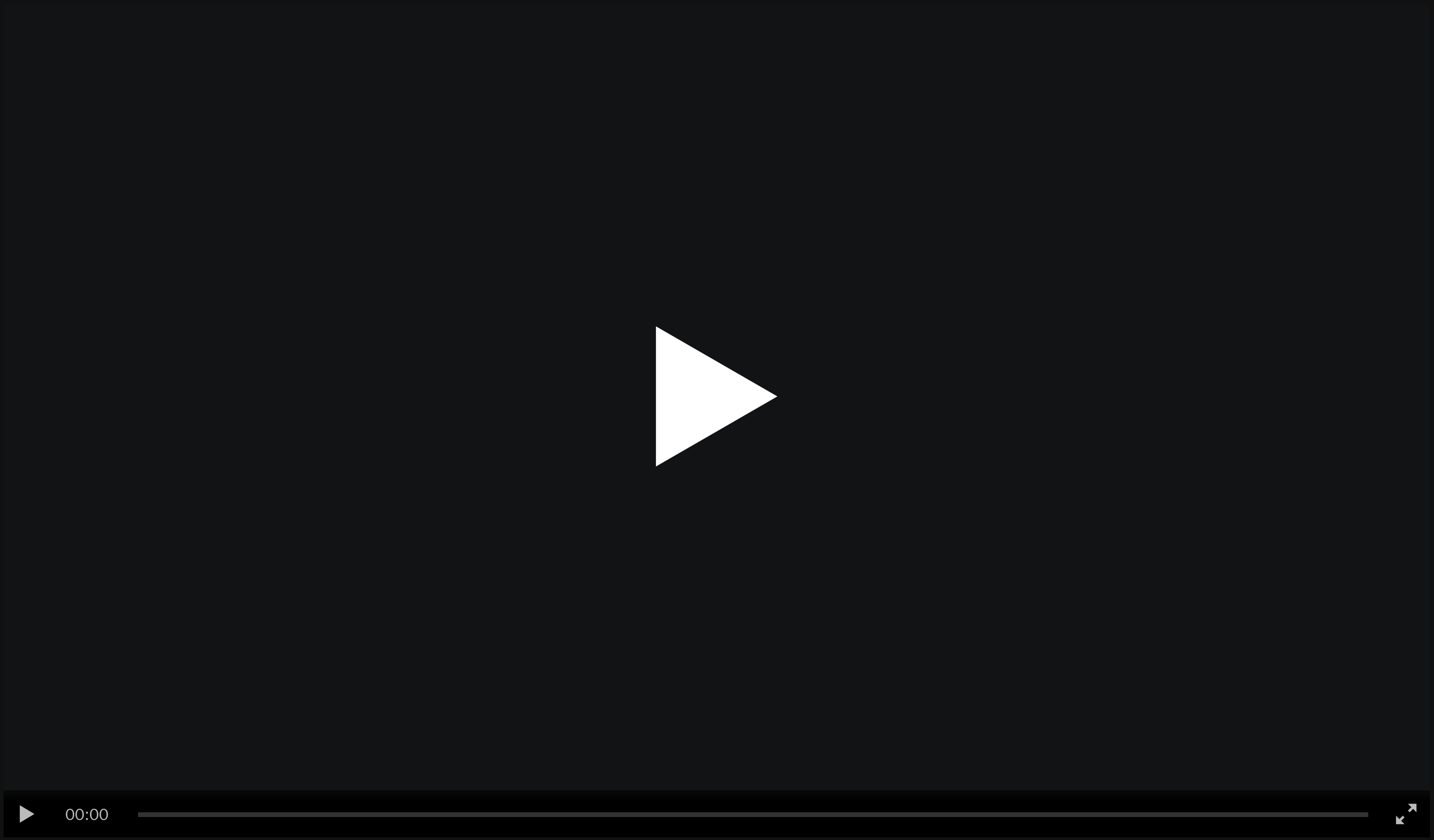
Config



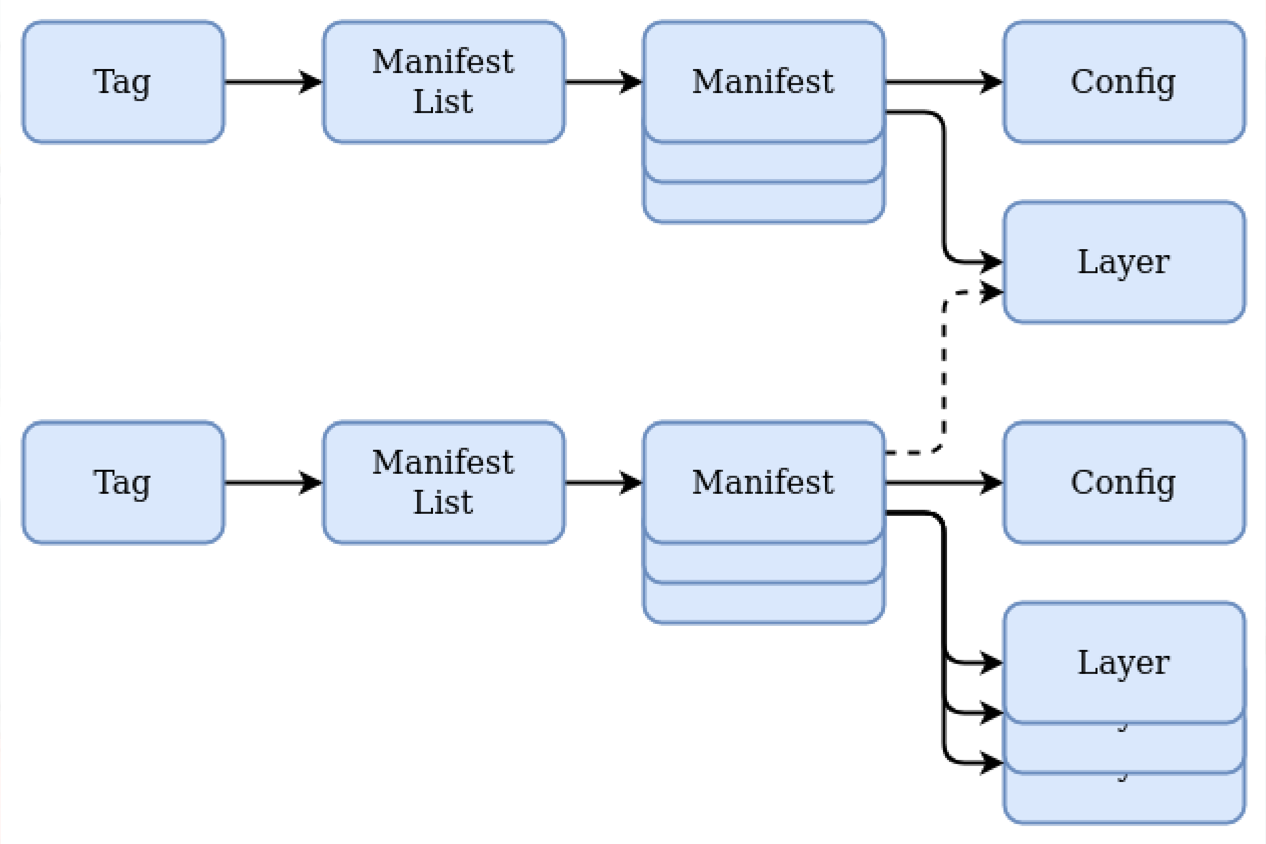

00:00

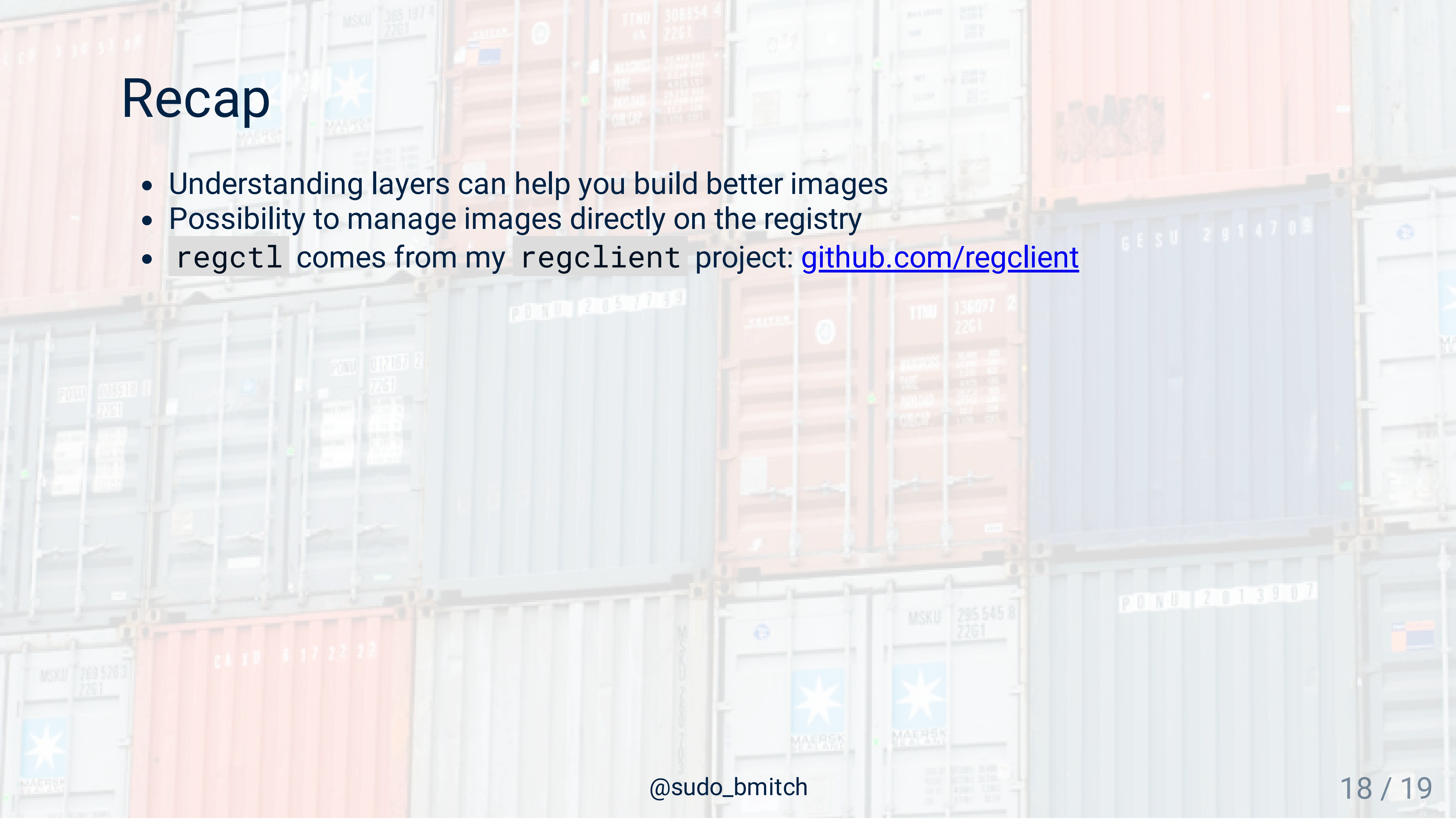
Layers




00:00

Shared Layers





Recap

- Understanding layers can help you build better images
- Possibility to manage images directly on the registry
- `regctl` comes from my `regclient` project: github.com/regclient

Thank You

github.com/sudo-bmitch/presentations

Brandon Mitchell
Twitter: @sudo_bmitch
GitHub: sudo-bmitch