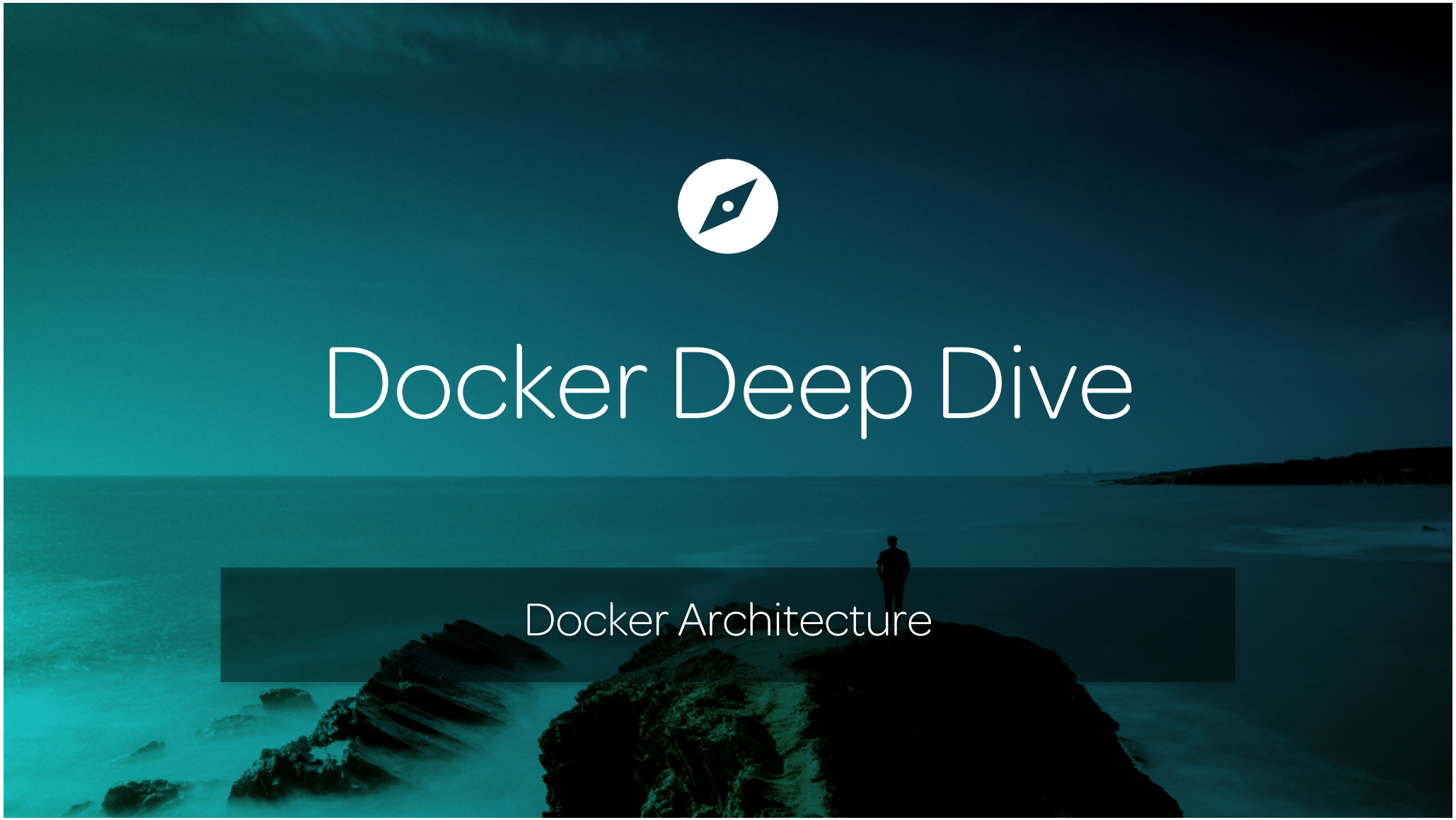

Docker Deep Dive
Docker Architecture

Architecture Overview

Docker architecture:

- Client-server architecture
- The client talks to the Docker deamon
- The Docker deamon handles:
  - Building
  - Running
  - Distributing
- Both communicate using a REST API:
  - UNIX sockets
  - Network interface

Architecture Overview (cont.)

The Docker daemon (`dockerd`):

- Listens for Docker API requests and manages Docker objects:
  - Images
  - Containers
  - Networks
  - Volumes

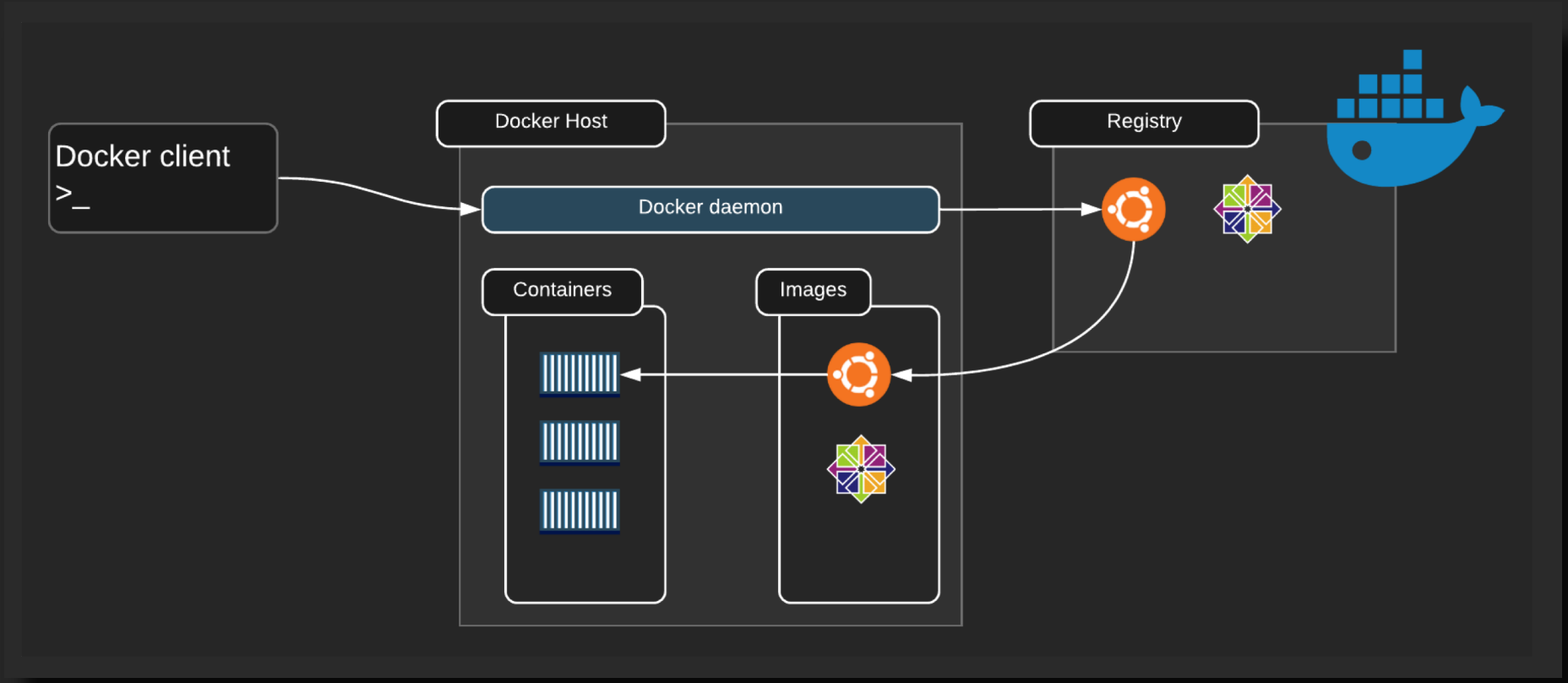
The Docker client (docker):

- Is how users interact with Docker
- The client sends these commands to `dockerd`

Docker registries:

- Stores Docker images
- Public registry such as DockerHub
- Run your own private registry

Architecture Overview (cont.)



Architecture Overview (cont.)

Docker objects:

- Images
  - Read-only template with instructions for creating a Docker container
  - Image is based on another image
  - Create your own images
  - Use a Dockerfile to build images
- Containers
  - Runnable instance of an image
  - Connect a container to networks
  - Attach storage
  - Create a new image based on its current state
  - Isolated from other containers and the host machine

Architecture Overview (cont.)

Docker objects:

- Services
  - Scale containers across multiple Docker daemons
  - Docker Swarm
  - Define the desired state
  - Service is load-balanced

Docker Swarm:

- Multiple Docker daemon (Master and Workers)
- The daemons all communicate using the Docker API
- Supported in Docker 1.12 and higher