docker

Module Eight

Docker in CI/CD

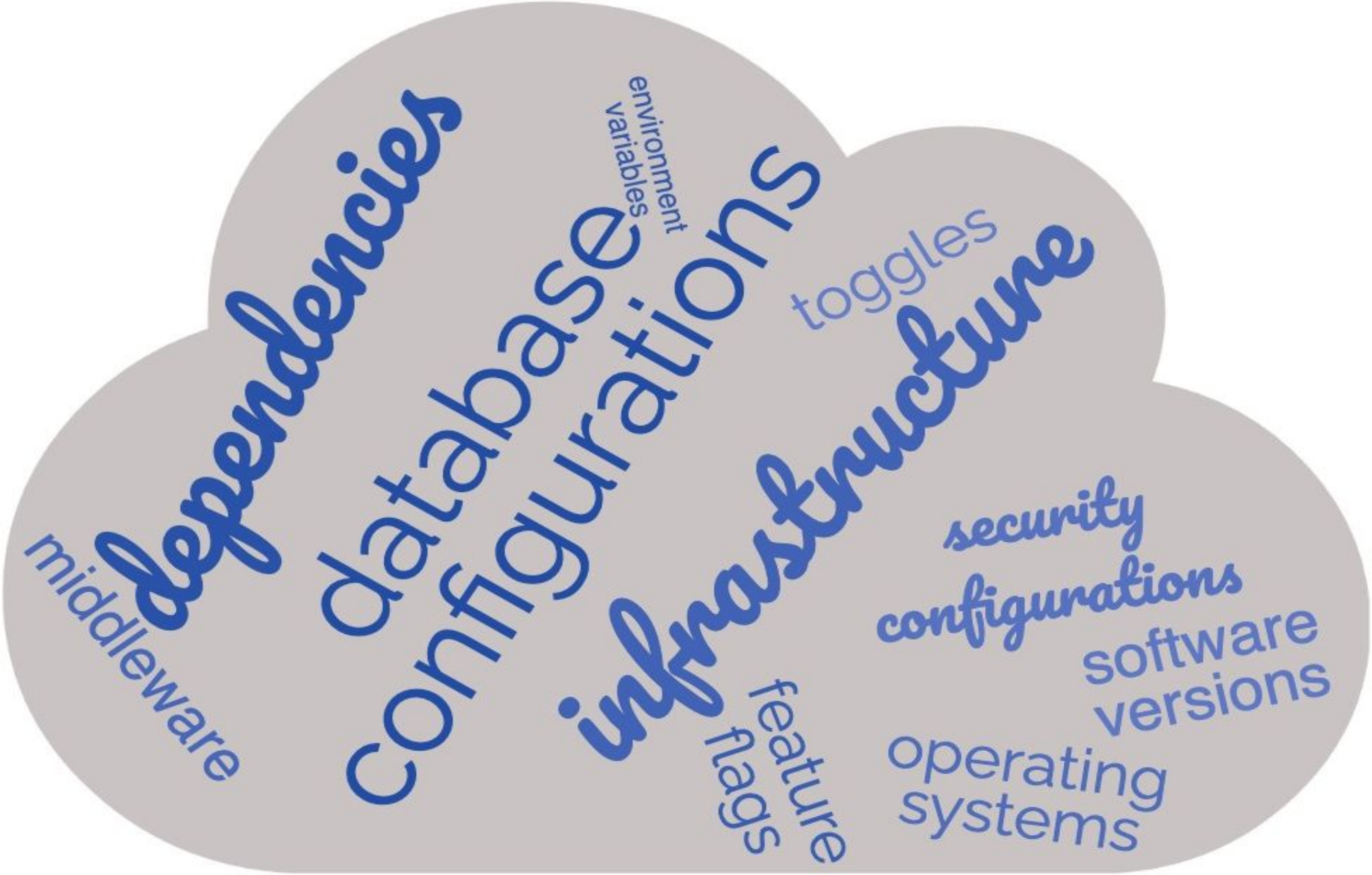
A Problem

Drift

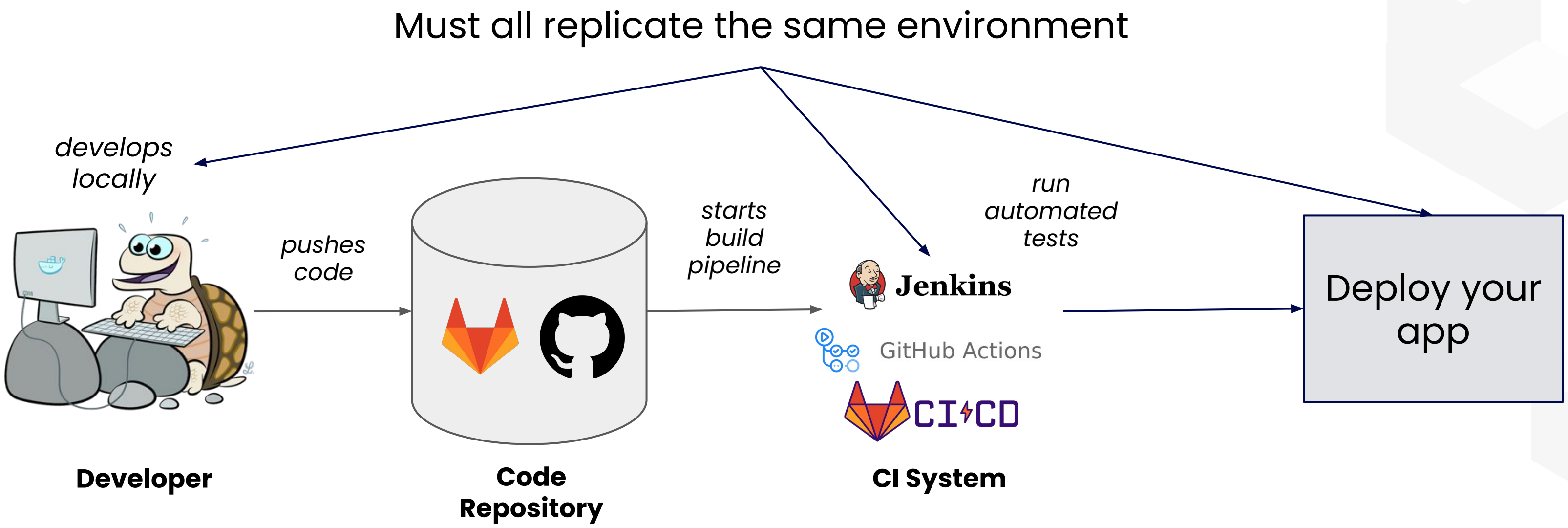
What is Drift?

"The unintentional divergence of configurations and settings across different environments"

Lots of Things Drift....



Chances for Drift



Must all replicate the same environment

develops locally

pushes code

starts build pipeline

run automated tests

Jenkins

GitHub Actions

CI/CD

Deploy your app

Developer

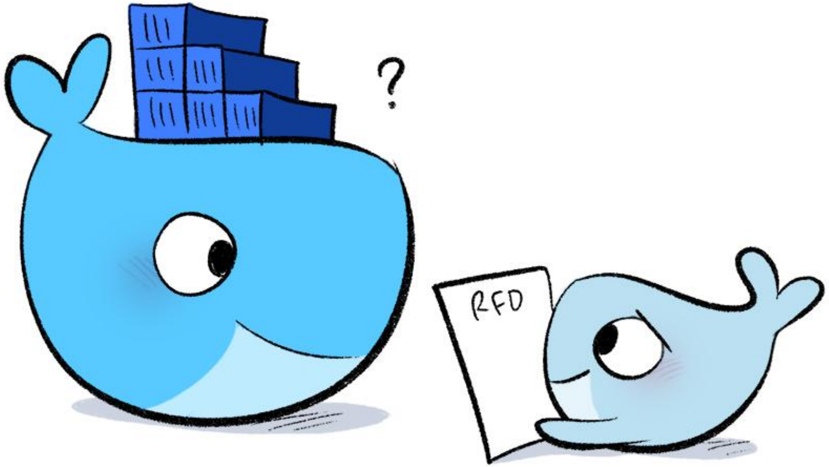
Code Repository

CI System

docker

Continuous Integration
Continuous Deployment

What is Continuous Integration (CI) ?



“An automated process of merging, building testing, scanning, and releasing code changes regularly”

Continuous Integration

✓

- Automated Testing
- Frequent Code Integration
- Immediate Feedback
- Consistent Build Process
- Collaboration and Transparency

✗

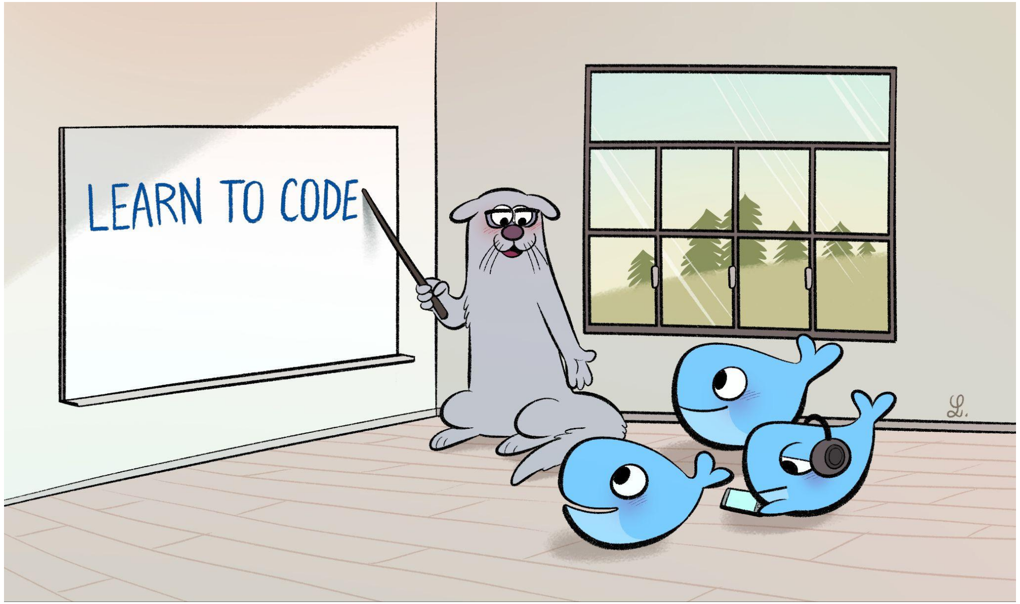
- Release Management
- Replacement for Dev Testing
- Infrastructure Management
- Readiness Guarantee
- Developer Environment

Benefits of CI

- Rapid Feedback
- Early Bug Detection
- Reduced Context Switching
- Consistent Codebase
- Enhanced Collaboration
- Reduced Integration Risks

What is Continuous Delivery (CD)?

"Continuous delivery is the practice of automating the entire software release process, ensuring that code changes are always deployable to production at any time."



Continuous Delivery

✓

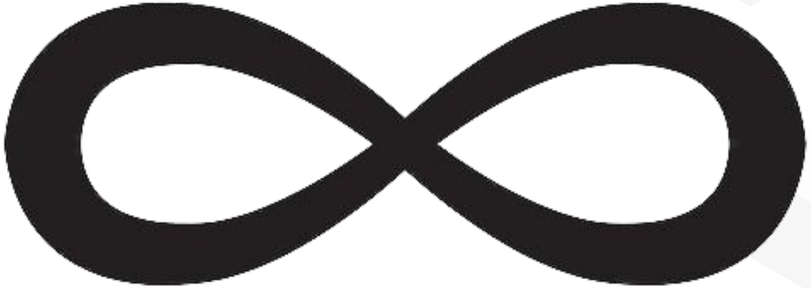
- Automated Deployment
- Frequent Releases
- Consistency
- Rollback Mechanism
- Pipeline Approach

✗

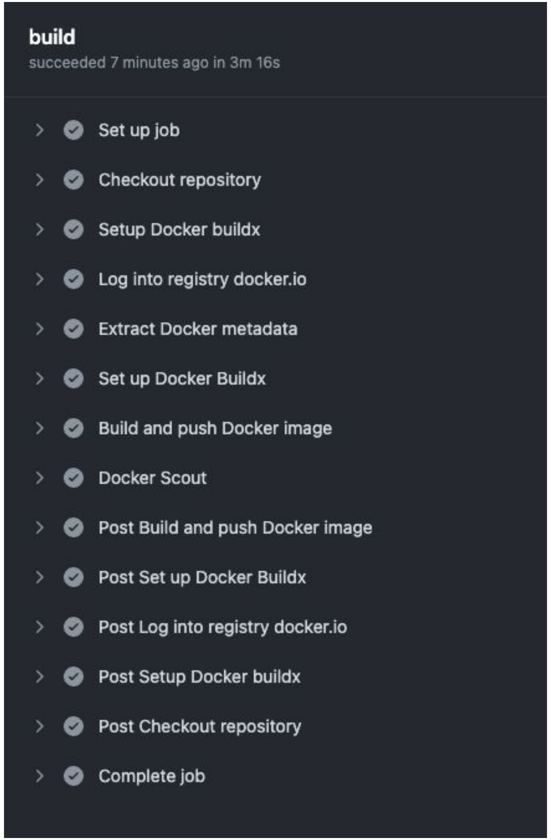
- Continuous Deployment
- Only for Large Releases
- Replacement for Testing
- Developer Responsibility
- One Size Fits All

Benefits of CD

- Reduced Deployment Risks
- Faster Time to Market
- Rapid Feedback Loop
- Consistent and Reliable Releases
- Enhanced Collaboration
- Cost Efficiency



What makes a good CI/CD pipeline?



build

succeeded 7 minutes ago in 3m 16s

- Set up job
- Checkout repository
- Setup Docker buildx
- Log into registry docker.io
- Extract Docker metadata
- Set up Docker Buildx
- Build and push Docker image
- Docker Scout
- Post Build and push Docker image
- Post Set up Docker Buildx
- Post Log into registry docker.io
- Post Setup Docker buildx
- Post Checkout repository
- Complete job

- Automated Testing
- Consistent Environments
- Fast Feedback Loops
- Rollback Mechanism
- Monitoring and Logging

CI/CD tools

- Provide automation to minimize manual intervention
- Integrate with version control systems
- Provide build and test execution
- Allow configuration and visualization of pipelines
- Provide feedback mechanisms
- Can be extended with plugins, etc

docker

Docker (in, out, context)

Docker / Docker Use Cases

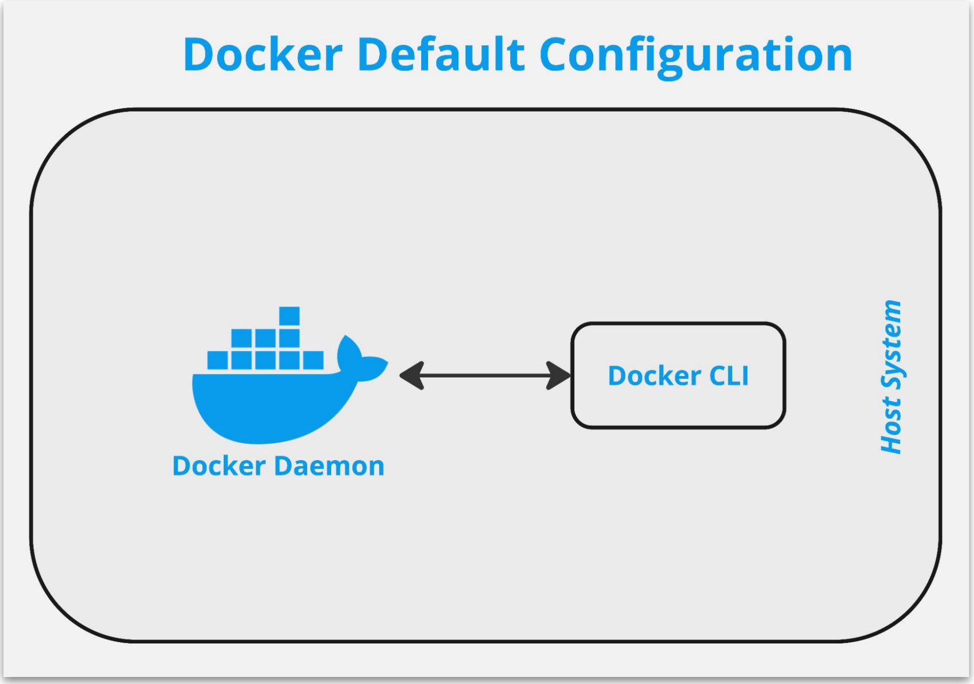
- CI/CD Pipeline Integration
- Environment Isolation
- Resource Control and Limitation
- Multi-Tenancy
- Security and Access Control
- Testing Docker at Scale
- Portability and Versioning

Standard Docker Configuration

This is the familiar, default configuration.

Client and Daemon run on the same workstation.

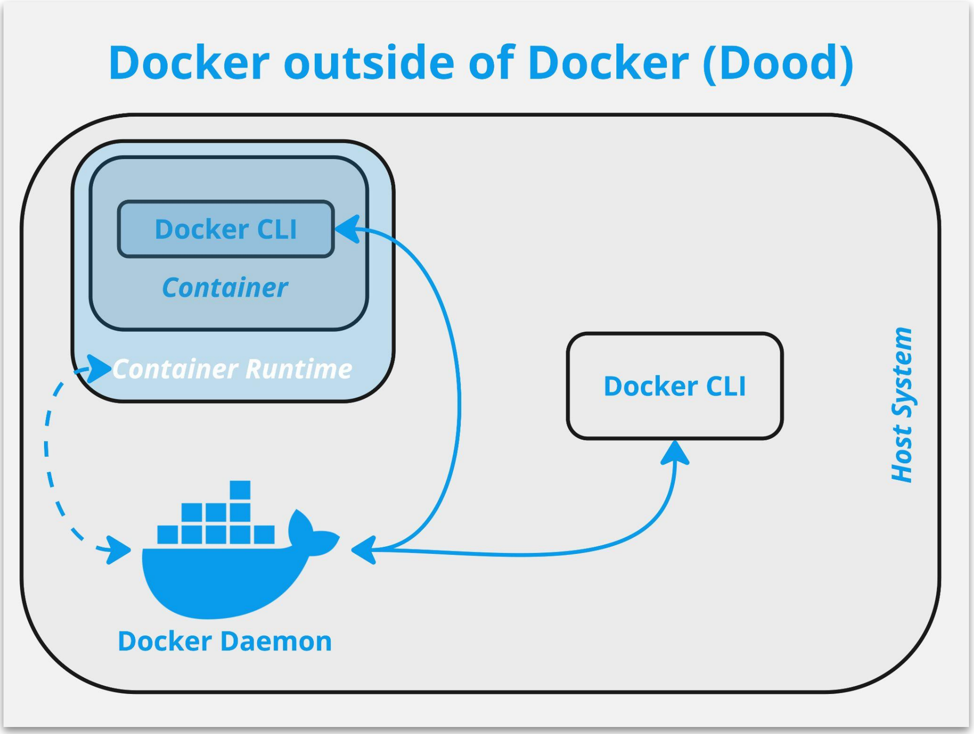
Note there is an additional abstraction layer with Docker Desktop and it’s internal VM.



Docker outside of Docker (DooD)

DooD uses the host system Docker installation from the container.

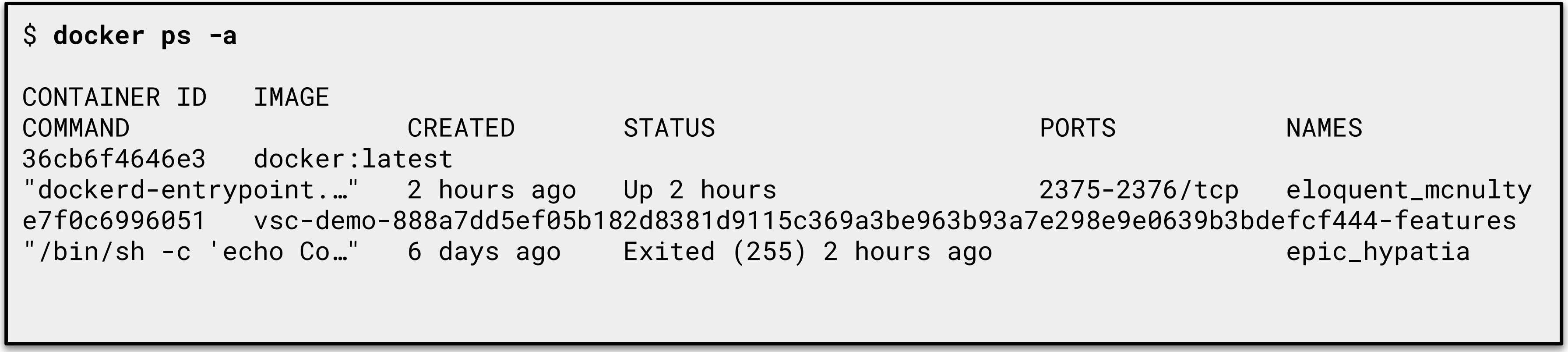
- Pros
  - Simpler to set up
  - Uses host resources directly
  - No nested virtualization
  - Does not require privileged containers
  - Could use sysbox
- Cons
  - Potential security implications
  - Exposes Daemon to container
  - Shared Daemon may cause confusion



Running DooD

```
$ docker run -v /var/run/docker.sock:/var/run/docker.sock -ti --rm docker:latest /bin/sh

# docker ps -a
CONTAINER ID   IMAGE
COMMAND                  CREATED          STATUS                     PORTS           NAMES
117d332ffa18   docker:latest
"dockerd-entrypoint.…"   5 seconds ago    Up 4 seconds               2375-2376/tcp
ecstatic_mestorf
36cb6f4646e3   docker:latest
"dockerd-entrypoint.…"   2 hours ago      Up 2 hours                 2375-2376/tcp
eloquent_mcnulty
e7f0c6996051   vsc-demo-888a7dd5ef05b182d8381d9115c369a3be963b93a7e298e9e0639b3bdefcf444-features
"/bin/sh -c 'echo Co…"   6 days ago       Exited (255) 2 hours ago                   epic_hypatia

# exit
```

Running DooD: Caveat



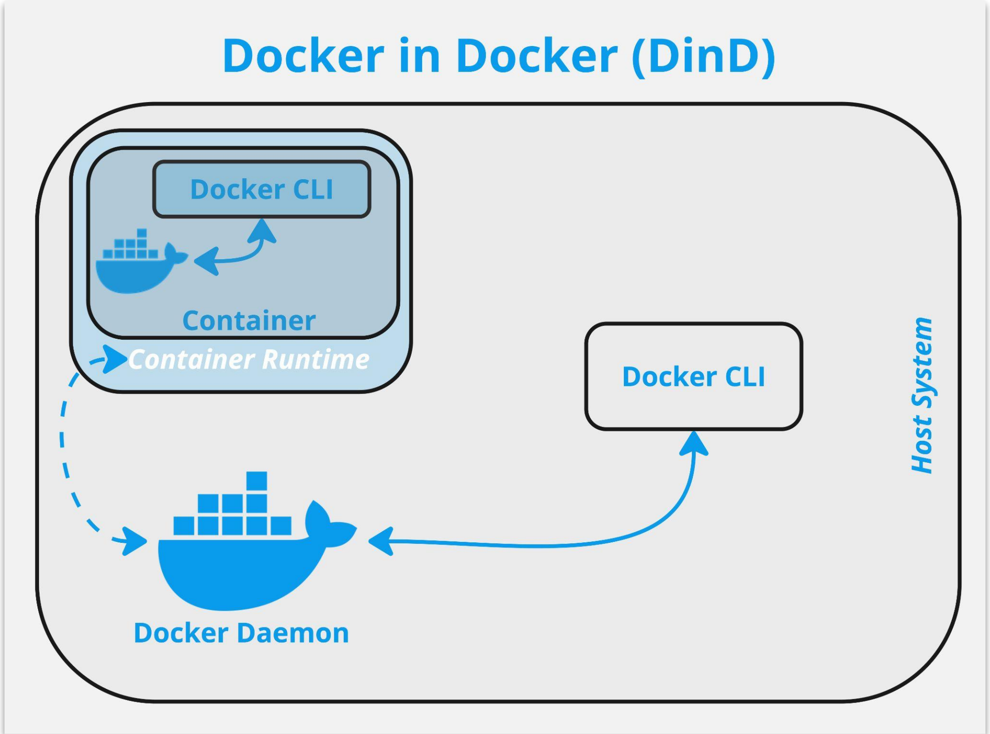
```
$ docker ps -a

CONTAINER ID   IMAGE
COMMAND                  CREATED       STATUS                           PORTS           NAMES
36cb6f4646e3   docker:latest
"dockerd-entrypoint.…"   2 hours ago   Up 2 hours                       2375-2376/tcp   eloquent_mcnulty
e7f0c6996051   vsc-demo-888a7dd5ef05b182d8381d9115c369a3be963b93a7e298e9e0639b3bdefcf444-features
"/bin/sh -c 'echo Co…"   6 days ago    Exited (255) 2 hours ago                         epic_hypatia
```

As inside…so outside…

Docker in Docker (DinD)

DinD runs its own Docker daemon in the container.

- Pros
  - Isolated docker environment
  - Simplified CI/CD
  - Multi-tenancy workloads
- Cons
  - Requires nested virtualization
  - Requires privileged containers
  - Greater complexity
  - Requires root access
  - Security model implications
  - Cannot use compose by default



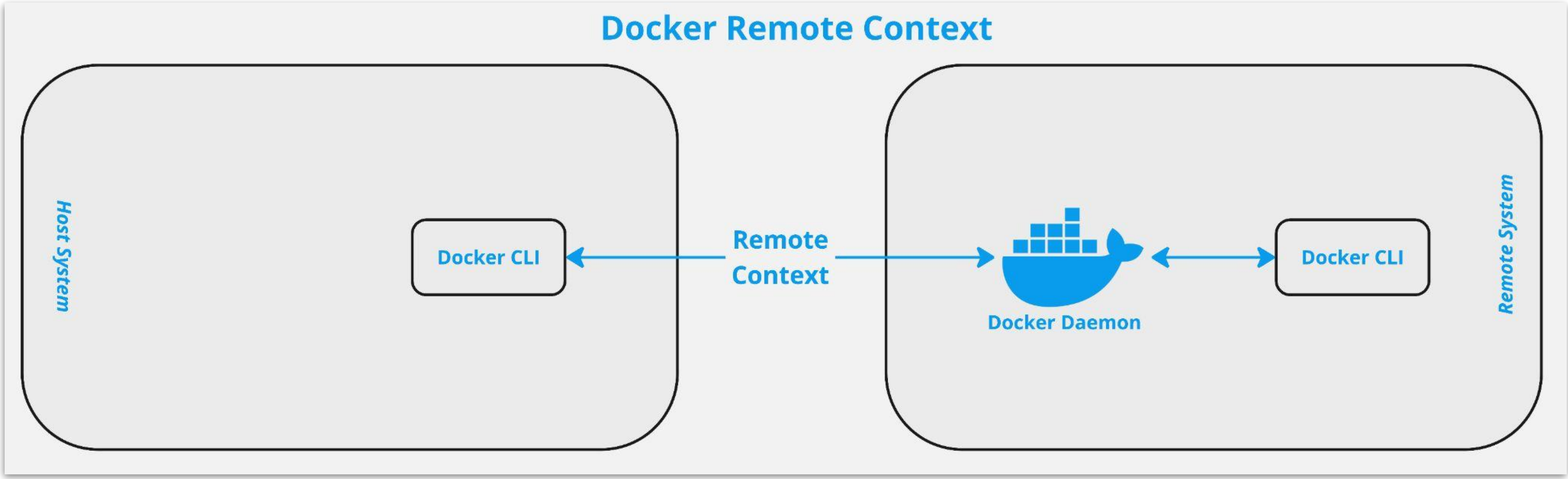
Running DinD

```
$ docker run --privileged -d --name demo-container docker:dind
731e2ac792f4f9c17cfece34395d2b9fa2dd4eb3ef13833cb1df3755aa4a8d5d
$ docker exec -it 731e2ac792f4f9c17cfece34395d2b9fa2dd4eb3ef13833cb1df3755aa4a8d5d /bin/sh

# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

# docker run -P -d nginx:latest
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
← SNIP →
Digest: sha256:6db391d1c0cfb30588ba0bf72ea999404f2764febf0f1f196acd5867ac7efa7e
Status: Downloaded newer image for nginx:latest
1a16c1907ffc05d0c305d8444528ba59149eec032d6364fed1b4d7b9c041480f

# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORTS
NAMES
1a16c1907ffc   nginx:latest   "/docker-entrypoint.…"   6 seconds ago   Up 5 seconds
0.0.0.0:32768->80/tcp   vibrant_mccarthy

# exit
```

Running DinD: Isolated

```
$ docker ps -a
CONTAINER ID   IMAGE
COMMAND                  CREATED              STATUS                     PORTS          NAMES
731e2ac792f4   docker:dind
"dockerd-entrypoint.…"   About a minute ago   Up About a minute          2375-2376/tcp
demo-container
94d0c9d09aa8   docker:dind
"dockerd-entrypoint.…"   About a minute ago   Up About a minute          2375-2376/tcp
brave_hoover
e7f0c6996051   vsc-demo-888a7dd5ef05b182d8381d9115c369a3be963b93a7e298e9e0639b3bdefcf444-features
"/bin/sh -c 'echo Co…"   6 days ago           Exited (255) 2 hours ago
epic_hypatia
```

Docker Remote Context



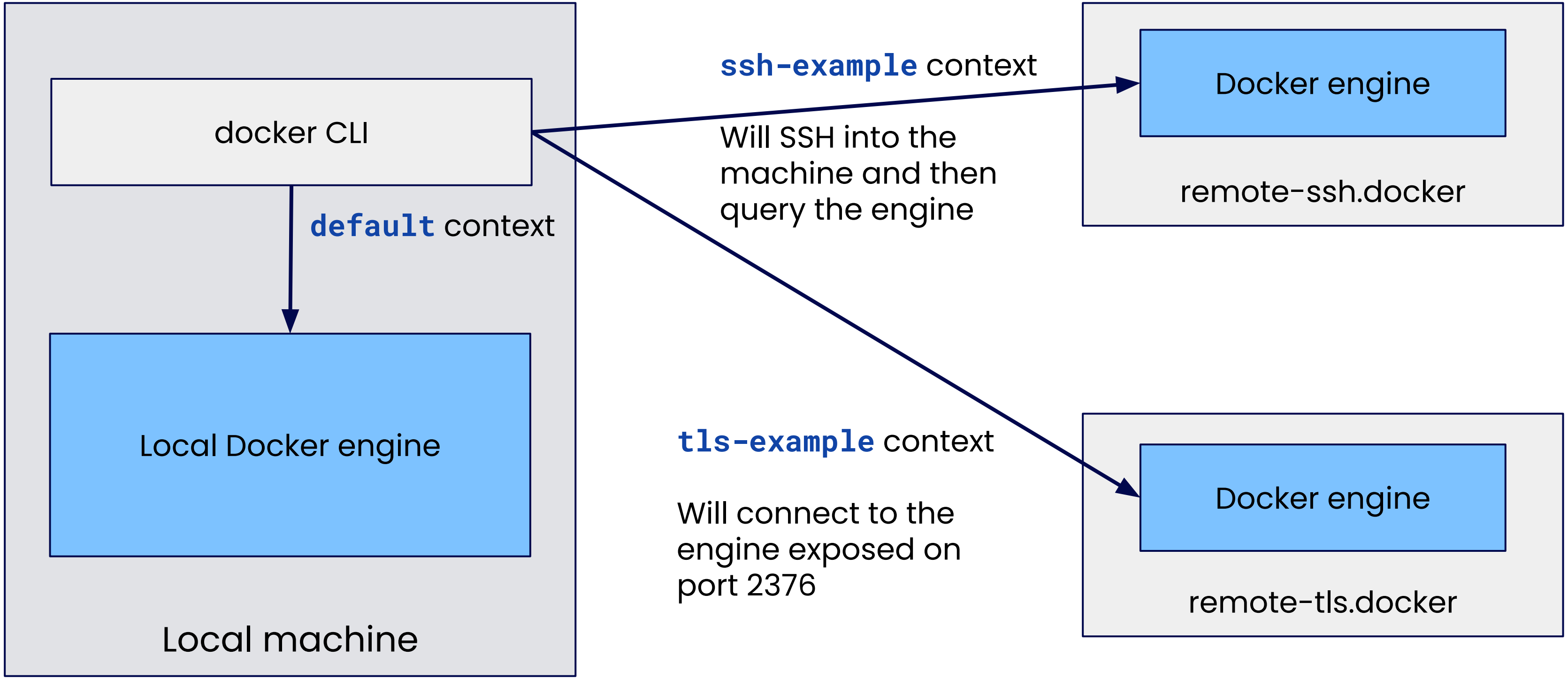
Connects across a tcp port or ssh connection to allow access to a remote docker daemon

Docker contexts 101

- Contexts allow your local CLI to work with other Docker engines
- The default context works with the engine on your machine
- Common commands:
    - `docker context create` - create a new context
    - `docker context ls` - list current contexts
    - `docker context use <name>` - make the specified context the default

```
docker context create ssh-example --docker \
"host=ssh://user@remote-ssh.docker"

docker context create tls-example --docker \
"host=tcp://remote-tls.docker:2376,ca=ca.pem,cert=cert.pem,key=key.pem"
```

Docker contexts visually



Using Docker Contexts

```
$ docker context create docker01 --docker host=ssh://root@docker01.virington.com
docker01
Successfully created context "docker01"

$ docker context use docker01
docker01
Current context is now "docker01"

$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE

$ docker run -d -P nginx:latest
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
← Snip →
Digest: sha256:6db391d1c0cfb30588ba0bf72ea999404f2764febf0f1f196acd5867ac7efa7e
Status: Downloaded newer image for nginx:latest
10bdfd89871ee1c47318e5ad2902cb70e72353871daf77307ee402eabb6f2793

$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORTS
NAMES
10bdfd89871e   nginx:latest   "/docker-entrypoint.…"   3 seconds ago   Up 2 seconds
0.0.0.0:32768->80/tcp, :::32768->80/tcp   gracious_hawking
```

Using Docker Contexts, Continued

```
$ docker context use desktop-linux
desktop-linux
Current context is now "desktop-linux"

$ docker ps -a
CONTAINER ID   IMAGE
COMMAND                  CREATED          STATUS                     PORTS            NAMES
731e2ac792f4   docker:dind
"dockerd-entrypoint.…"   26 minutes ago   Up 26 minutes              2375-2376/tcp
demo-container
94d0c9d09aa8   docker:dind
"dockerd-entrypoint.…"   26 minutes ago   Up 26 minutes              2375-2376/tcp    brave_hoover
e7f0c6996051   vsc-demo-888a7dd5ef05b182d8381d9115c369a3be963b93a7e298e9e0639b3bdefcf444-features
"/bin/sh -c 'echo Co…"   6 days ago       Exited (255) 3 hours ago                    epic_hypatia
```

```
$ ssh docker01.virington.com -l root

# docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS
NAMES
10bdfd89871e   nginx:latest   "/docker-entrypoint.…"   2 minutes ago    Up 2 minutes
0.0.0.0:32768->80/tcp, :::32768->80/tcp   gracious_hawking
```

docker

Questions and Answers