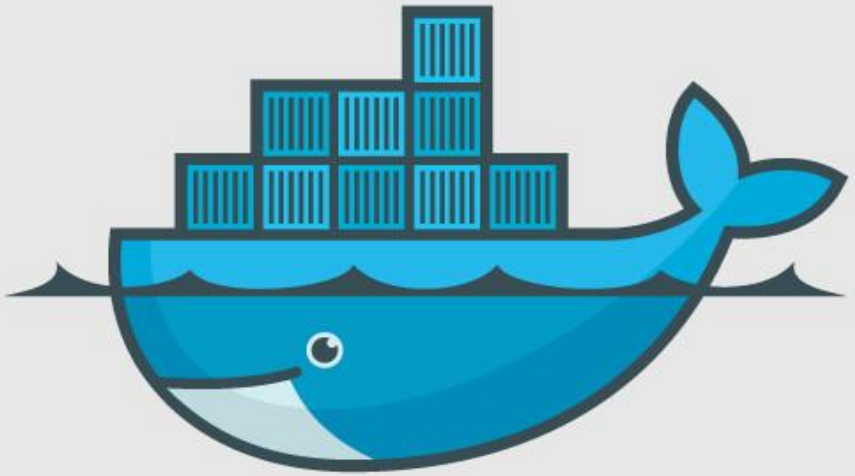


docker

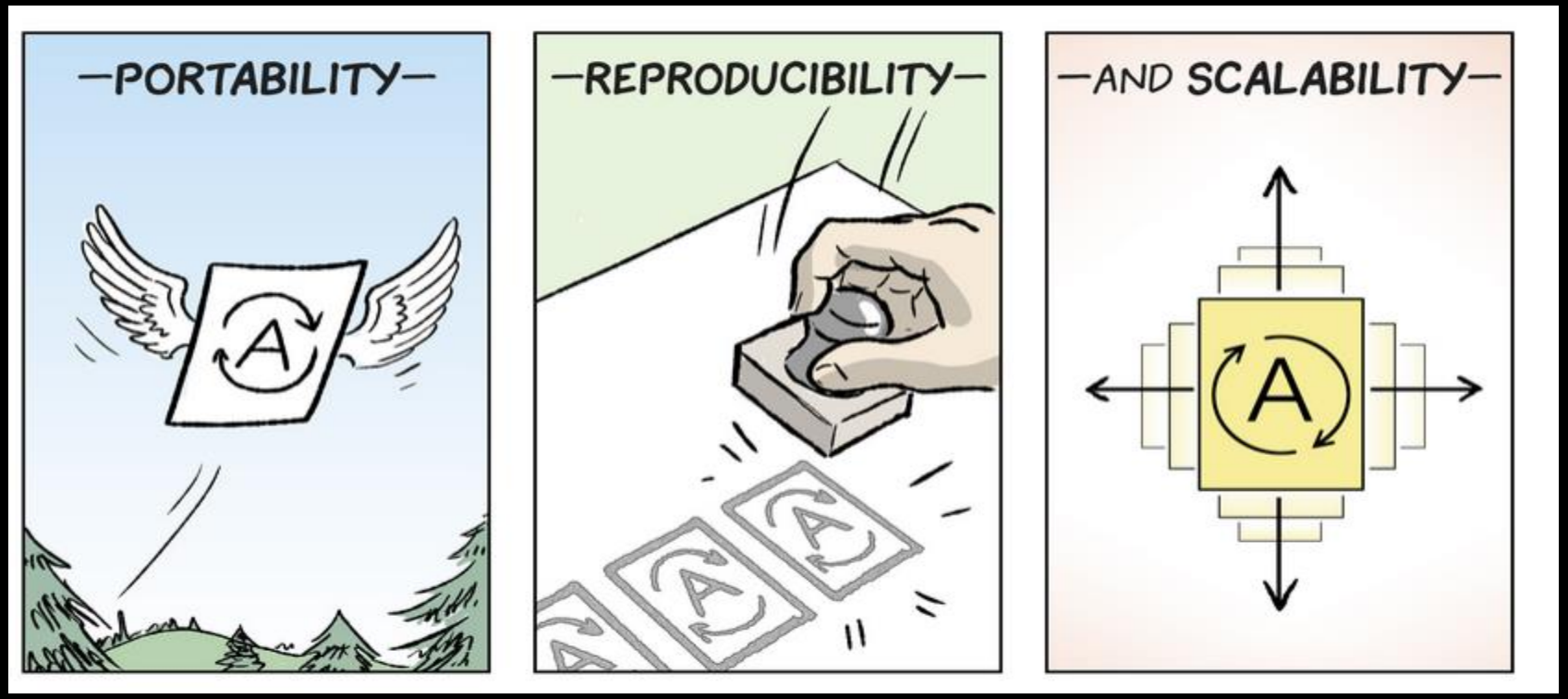
(Why, what, and how)

(Originally presented during my internship at Wandercraft, with annotations added after)

3 points

1. 4 benefits of Docker
2. What is Docker?
   - Using an analogy
   - Technically
3. Docker Examples

1. 4 Benefits of Docker



Compatible with any computer with Docker installed

An image always makes the same container

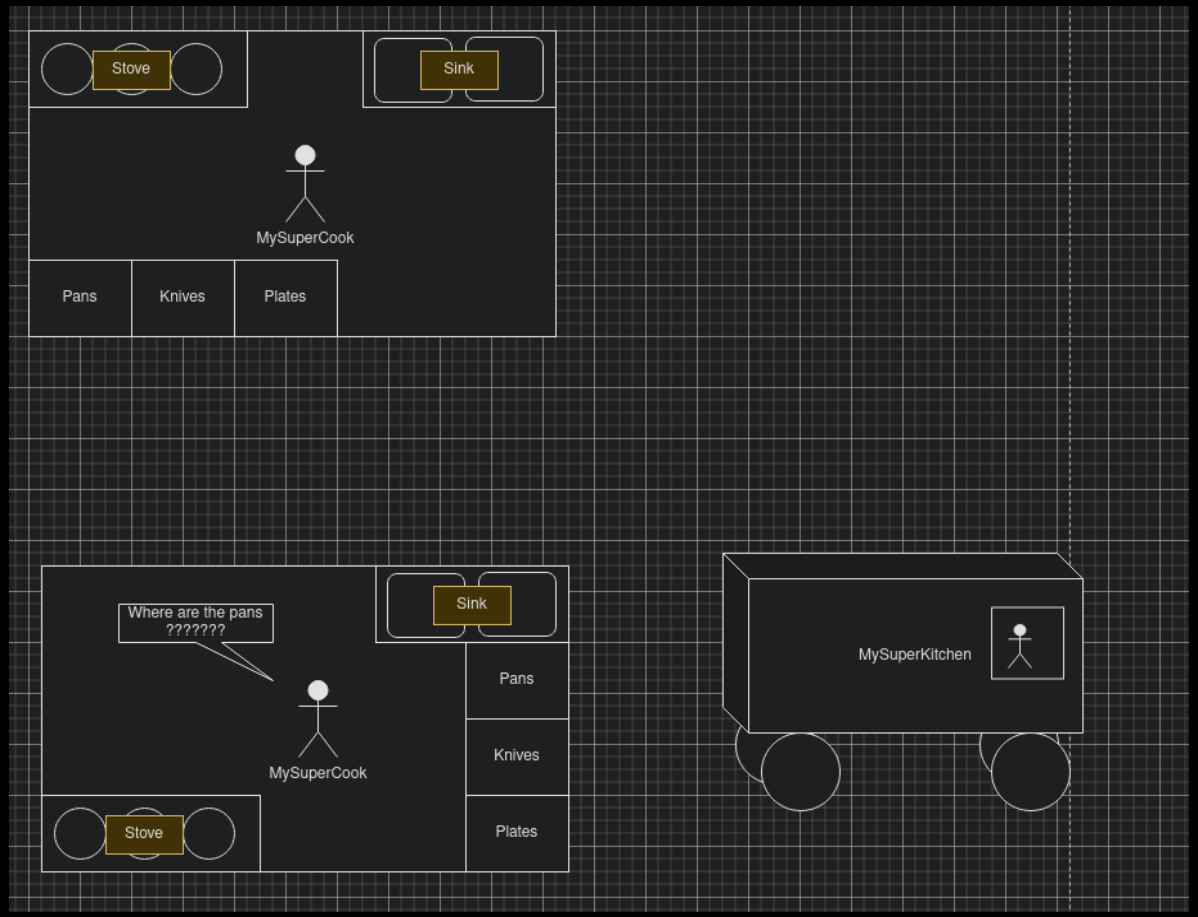
Multiple containers can be run across multiple computers

+ Isolation

Containers can only communicate through established channels

2a. What is Docker? Chef in a kitchen

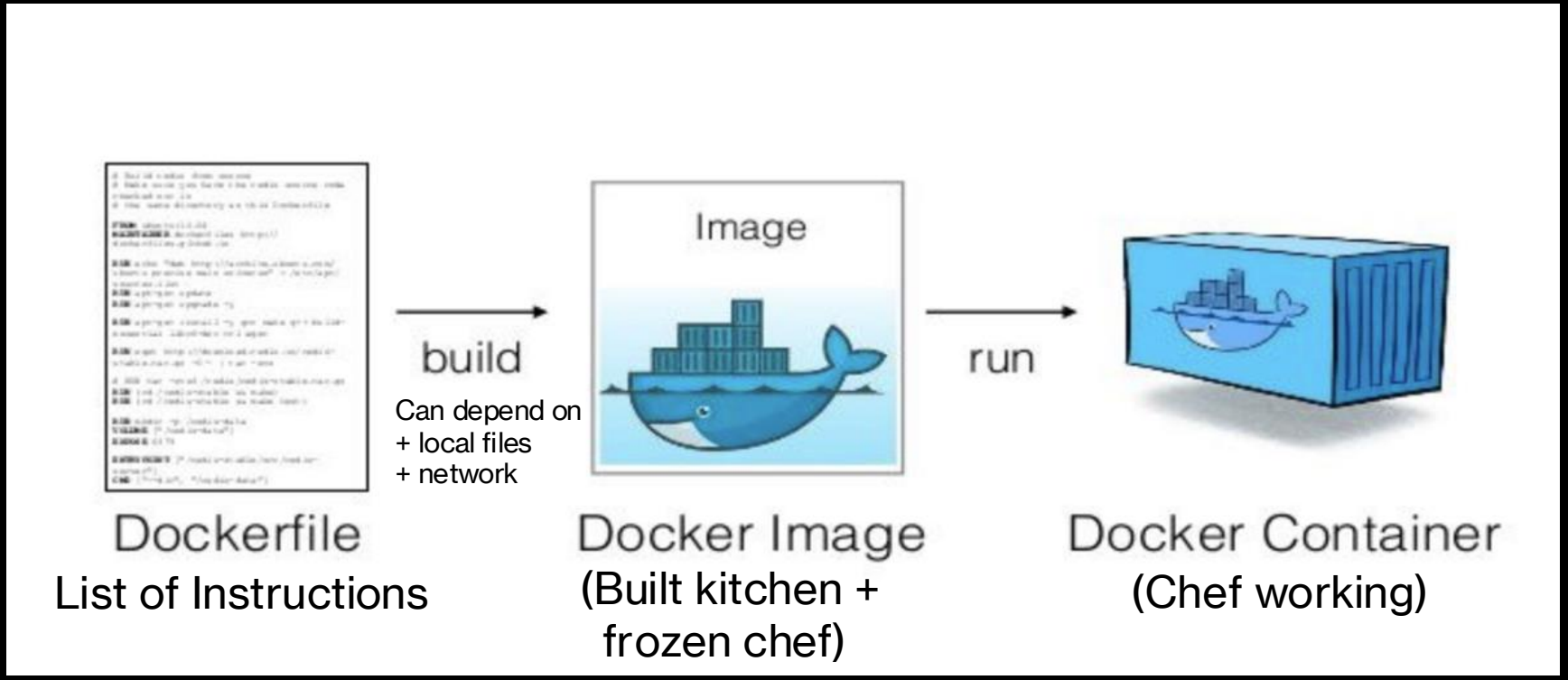
A chef can work well in their own kitchen



A chef might struggle to work in a different kitchen

Solution:
Ship the chef with the kitchen

2b. What is Docker? Technically



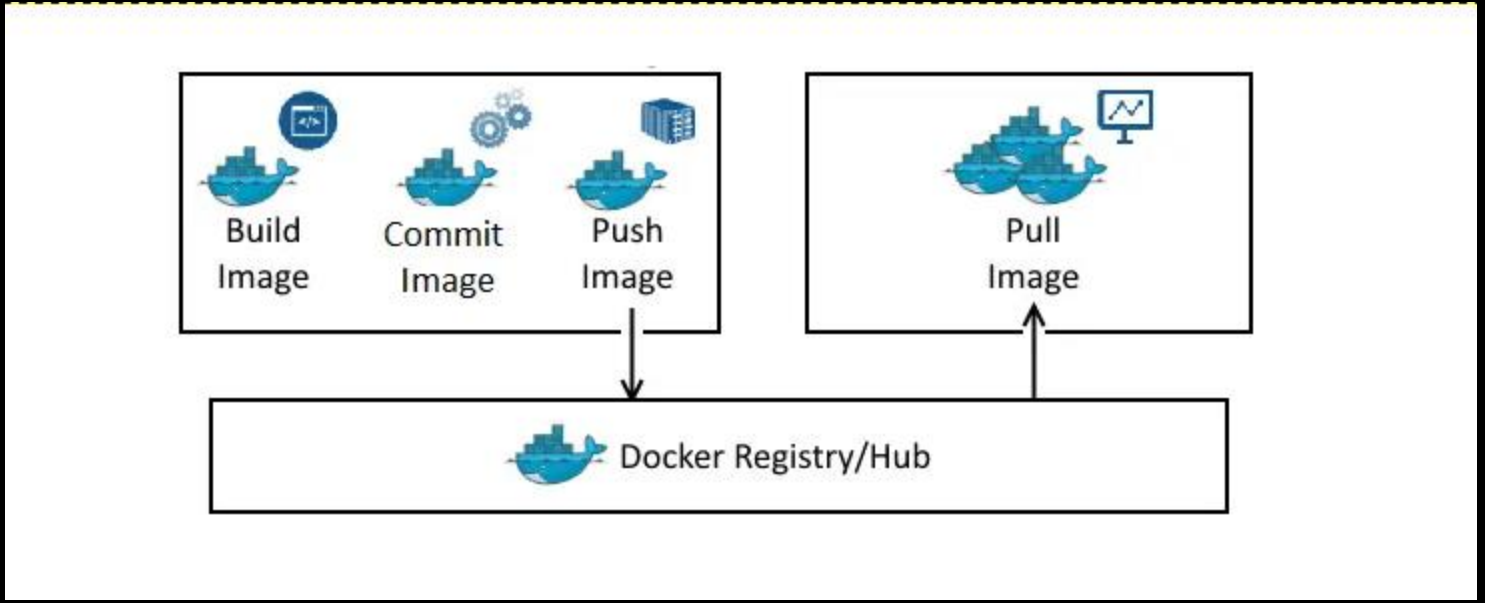
3. Docker Examples

3a. Registries

3b. Container Orchestration

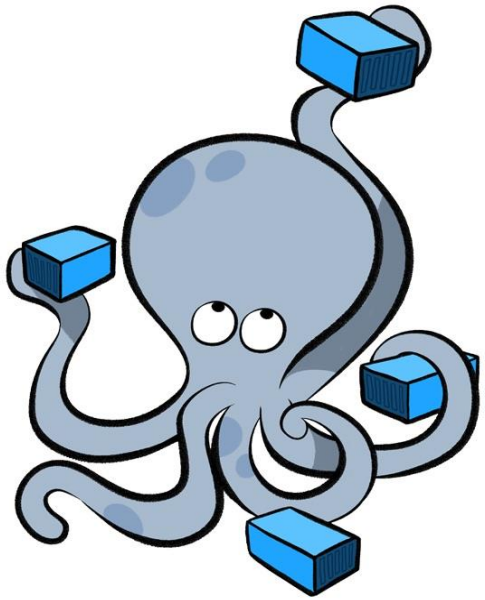
3c. (removed for confidentiality -> discussion of internal tools)

3a. Registries

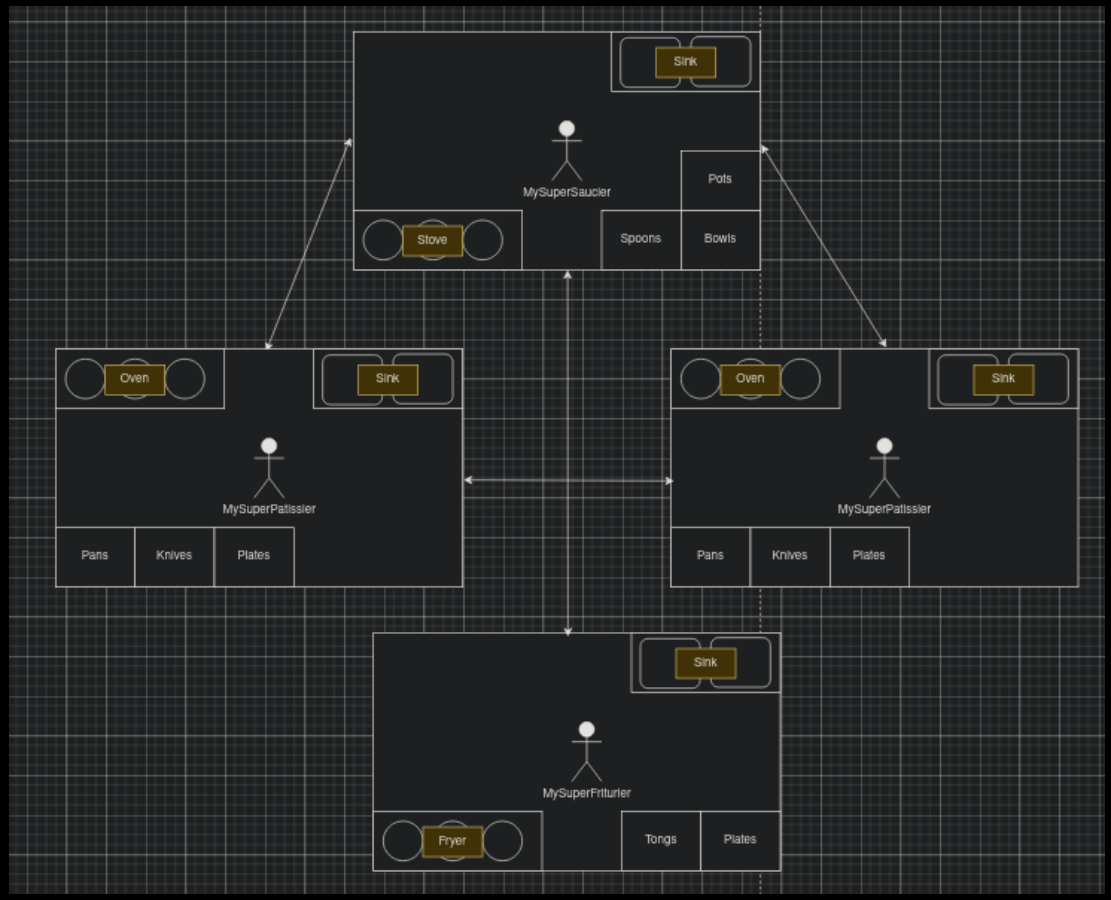


- Docker hub (https://hub.docker.com) or Self-hosted
- Like a remote repository for code (Gitlab, Github, etc.)

3b. Container Orchestration



Docker Compose



Chefs get sick:
auto-healing replacement

Scaling kitchen:
increase/decrease chefs

Service specialization:
specific chef + kitchen

Startup and teardown:
one step setup/stop

Security:
isolated internal network

3c. (removed for confidentiality -> discussion of internal tools)

- Development Workspace with Tools and Configuration set up

  (images removed)

Summary

1. Portable, Reproducible, Scalable, Isolated
2. Dockerfiles, Docker Images, Docker Containers
3. Registries, Orchestration, Development Environments

Resources + References

References and extra reading

- **https://devopswithdocker.com/**
- https://roadmap.sh/docker (key concepts)
- https://courses.devopsdirective.com/docker-beginner-to-pro/lessons/11-development-workflow/00-devx-wishlist
- https://aws.amazon.com/compare/the-difference-between-docker-vm/