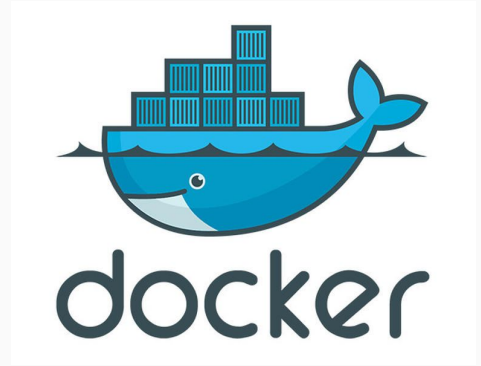
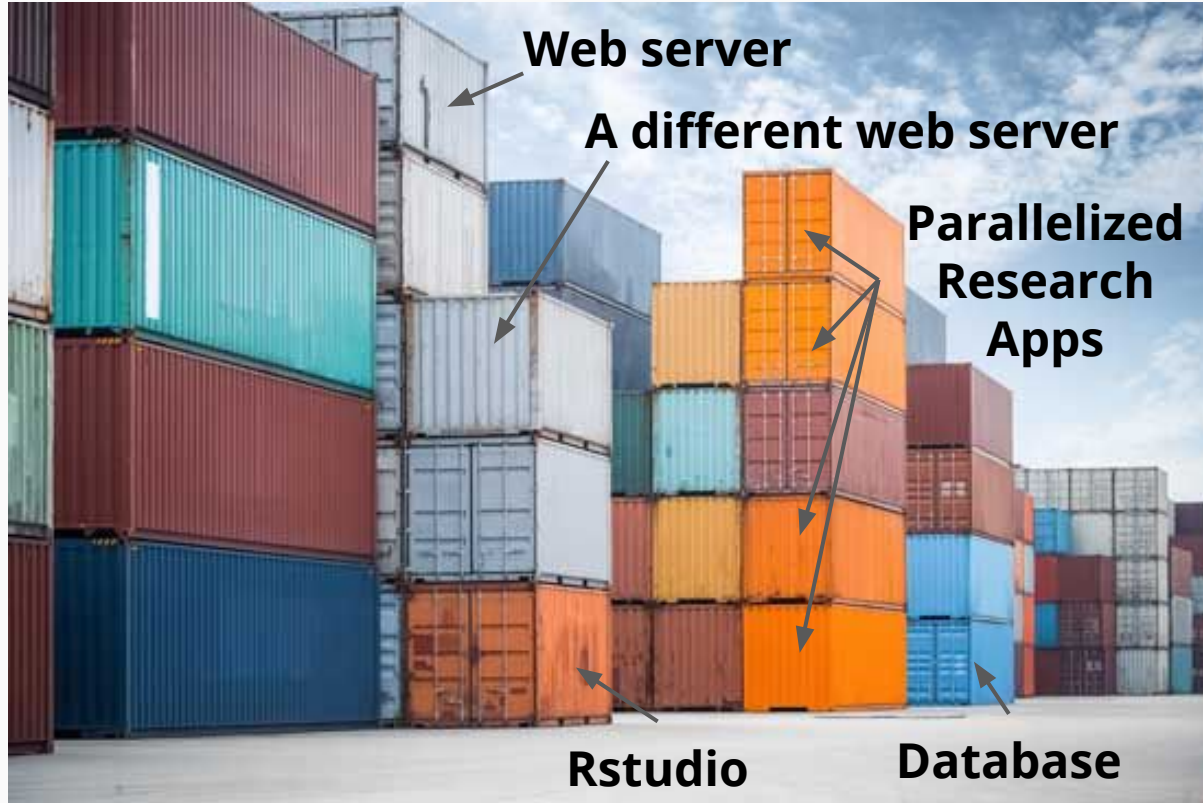


Docker Demo

Containerization



Containerization

- Complete environment description
- Essentially mini operating systems
- Contain all specific software needed for an analysis
- Custom code can also be loaded into container
- Current technologies: docker and singularity
- Strengths: ideal strategy for encapsulating and sharing environments
- Disadvantages: requires sysadmin knowhow, docker has prohibitive technical requirements for SCC

Docker basics

- Docker software is an environment that runs containers
- Containers execute in an isolated environment independent of host
- Containers may be allowed access to specific host resources
 - e.g. Local hard drive, networking, other containers

Docker images

- An *image*: description of a set of software
- Each image starts with a *base image* that provides a basic OS and environment
 - E.g. Ubuntu, alpine linux, Windows, etc
- More specific software is built upon the base image
- Any docker image can be a base image

Dockerfiles

- Images are specified in *Dockerfiles*
- Dockerfiles are written in a language that has specific syntax
- Many commands are available, but most often needed are:
 - FROM, RUN, COPY, CMD
- Docker uses the Dockerfile to *build* the container

Example Dockerfile

Specify base image name and version

Base image

Copy files into the image from the local directory

Run a command while building the container

Execute this command when running the container

```
FROM ubuntu:18.04
COPY . /app
RUN make /app
CMD python /app/app.py
```

Docker Hub

- Cloud application that hosts docker images
- Many different images available
- Images include a name and a version
- Specialized images you might consider:
 - r-base
 - continuumio/miniconda
 - python

Docker demo