Docker

CSC309

Kianoosh Abbasi

UNIVERSITY OF TORONTO

Fall 2024

So far

- Develop a web app with Next.js, Prisma, React, TypeScript, and Tailwind
- But you have to install a bunch of things on every machine you clone the application!
- The app's behavior depends on the environment's OS, its config, what's installed, etc.

This session

- Concept of isolation
- Intro to Docker
    - DockerFile
    - Containers, images, registry
- Docker compose

Isolation

- Ever experienced the "Works on my machine" issue?

- Each machine has
  - Different OS
  - Different softwares installed
    - Or different versions of the same software
  - Different softwares running at the same time
    - Could impact the file system, dependencies, etc.

Traditional solutions

- Virtual machines
  - Full isolation but very heavy and slow
- Sandboxing
  - Limit a process's access to resources (e.g., RAM, CPU, file system)
- chroot jail
  - Restrict a process to a specific directory
  - Cannot access outside that directory

Traditional solutions

- Trade-off between efficiency and true isolation
- Managing multiple instances is difficult
- Still not that portable!
    - Need to redo much of the work on a new machine

Docker

- A platform for developing, shipping, and running applications
    - Revolutionized software delivery
- Allows you to package an application with all its dependencies into a standardized unit called a container
- Makes your app portable: can be stopped, restarted, copied easily!
    - No longer worry about different machines, dependencies, etc.

Key benefits

- Consistency across environments
  - Solves the “It works on my machine” problem.
- Simplified dependency management
  - Don’t have to worry about installing them manually
- Containers run in insolation
- Easier continuous integration and deployment (CI/CD)

History

- 2008: Linux Containers (LXC)
  - Used Linux kernel features like cgroups and namespaces
  - Ran multiple isolated Linux systems on a single host

- 2010: dotCloud
  - Founded by Solomon Hykes
  - Led the exploration of containerization as a core technology

History

- 2013: dotCloud open-sourced their container technology, naming it Docker.

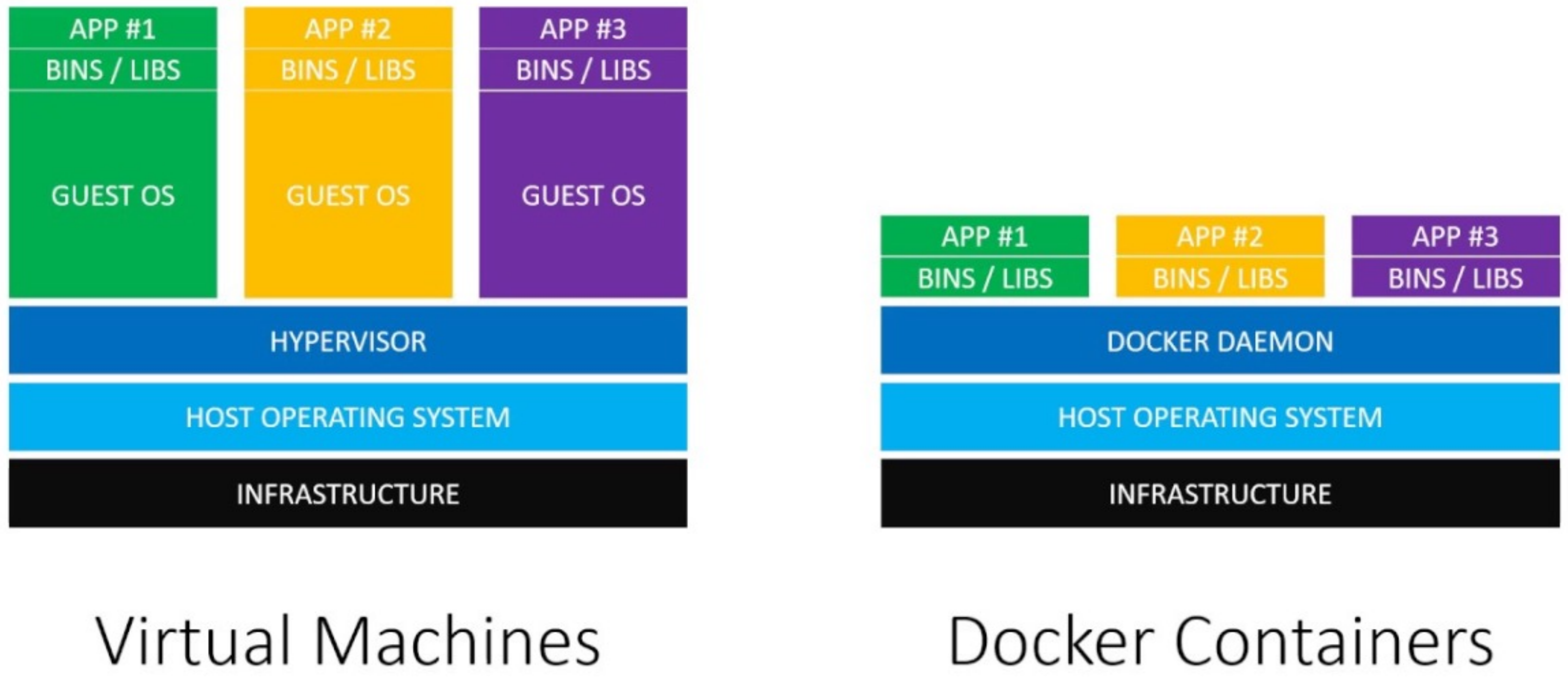
- Today, Docker is the industry standard for deployment

Containers vs virtual machines

- Virtual machine
  - Runs a full OS with its own kernel and a virtualized set of hardware resources (CPU, memory, storage) on a physical machine

- Docker container
  - Uses the host OS's kernel
  - Process-level isolation
  - Shared kernel space, isolated user space

Containers vs virtual machines



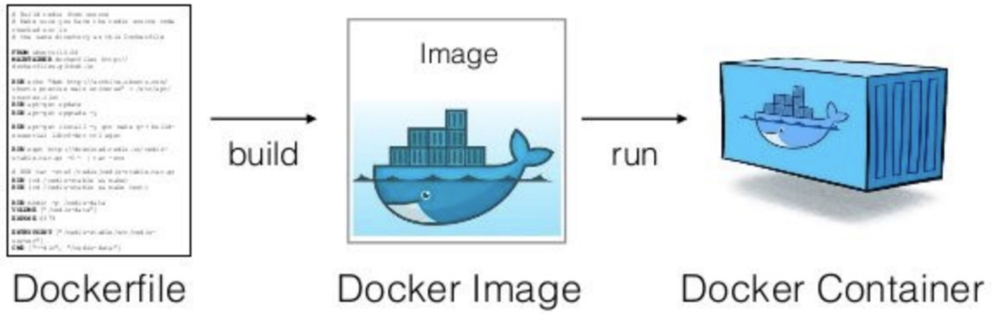
Source: https://www.linkedin.com/pulse/vms-vs-containers-baha-abu-shaqra-phd-dti-uottawa--c0slf/

Containers vs virtual machines

Containers	Virtual machines
Consistent across environments with the same OS	Consistent across all environments, regardless of OS
Lightweight	Very heavy
Fast start-up	Slow start-up
Very low performance overhead	High performance overhead

Docker concepts

- Dockerfile
- Images
- Containers
- Volumes
- Docker hub
- Compose



Source: https://itnext.io/intro-to-docker-part-1-5b1162c81735

Dockerfile

- Contains instructions on how to build a Docker image
  - Dependencies installed here

- Example Instructions
  - `FROM`: Base image to start with
  - `RUN`: Executes commands (e.g., `apt install`).
  - `COPY`: Copies files from the host to the image
  - `CMD`: Command to run when the container starts
    - Note: container exits as soon as CMD finishes.

Example Dockerfile

- Create a file named Dockerfile

- Build command:

```
docker build –t hello.
```

- Run command:

```
docker run hello
```

```
FROM python:3.11

RUN echo 'print("Hello, World!")' > /app.py

CMD python /app.py
```

Next.js Dockerfile

- Runs the Next.js application in development mode
- Copies files to the image, installs the dependencies, and expose the port
- Run command:
  `docker run -p 3000:3000 nextjs-app`

```
FROM node:20-alpine

WORKDIR /app

COPY . .

RUN npm install

EXPOSE 3000

CMD npm run dev
```

Docker images

- A lightweight, standalone, and executable package
  - Includes everything needed to run a software
  - Code, dependencies, env variables, and system tools

- A read-only template used to create Docker containers

- Build starts from a base image
  - Examples: `alpine`, `ubuntu`, `node:alpine`, `python:3.12`, etc.

Docker images

- Built in layers: Each layer representing a step in Dockerfile
  - Layers are cached for efficiency and reusability

- Images are immutable and portable

- Can be versioned using tags
  - Default tag is `latest`

Questions?

Docker containers

- Instantiated from Docker images

- Run command:
```
docker run -d -p 8080:80 <image_name>:<image_tag>
```

- List running containers:
```
docker ps
```

- Stop a container
```
Docker stop <container_name>
```

- View logs
```
docker logs <container_name>
```

Docker volumes

- Persistent data storage for docker containers
- Also allows for sharing data between containers
- Use cases: database, user uploads, HTTPS ceritificates

Docker volumes

- Example: PostgreSQL

- Commands:

```
docker volume create pgdata
docker run -d \
   --name my-postgres \
   -e POSTGRES_PASSWORD=password \
   -v pgdata:/var/lib/postgresql/data \
   -p 5432:5432 \
   postgres
```

Docker hub

Visit: https://hub.docker.com

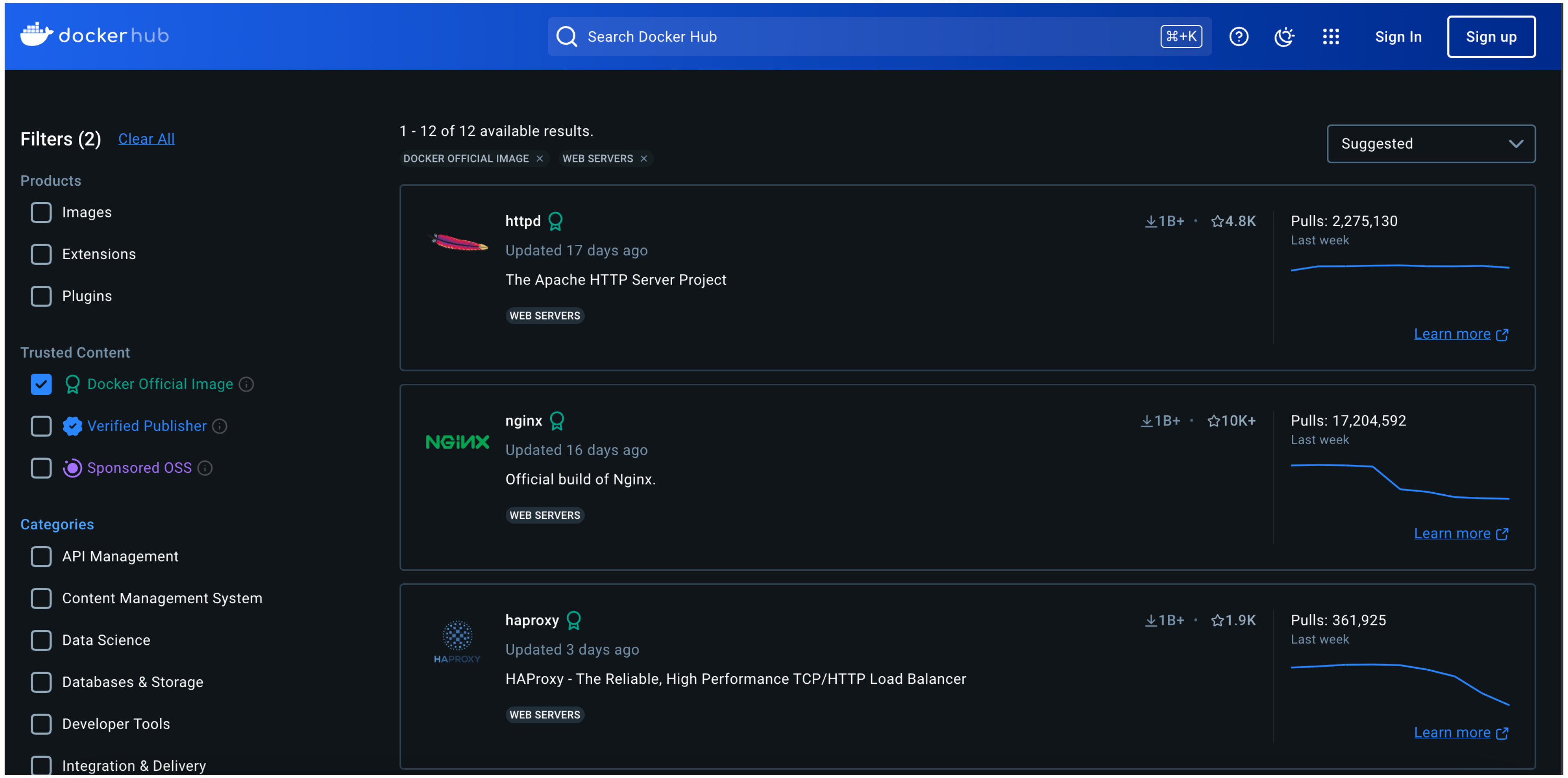
- Global repository of docker images

- You can search, explore, and use millions of images
  - Dockerfile's Base images are downloaded from Docker hub

- You can push and publish your own images as well!

```
docker login
docker push <username>/<image_name>:<image_tag>
```

Docker hub



docker hub Search Docker Hub ⌘+K Sign In Sign up

Filters (2) Clear All

Products
- Images
- Extensions
- Plugins

Trusted Content
- [✓ Docker Official Image]
- Verified Publisher
- Sponsored OSS

Categories
- API Management
- Content Management System
- Data Science
- Databases & Storage
- Developer Tools
- Integration & Delivery

1 - 12 of 12 available results.

DOCKER OFFICIAL IMAGE × WEB SERVERS ×

Suggested

httpd
Updated 17 days ago
The Apache HTTP Server Project
WEB SERVERS
1B+ · 4.8K
Pulls: 2,275,130
Last week
Learn more

nginx
Updated 16 days ago
Official build of Nginx.
WEB SERVERS
1B+ · 10K+
Pulls: 17,204,592
Last week
Learn more

haproxy
Updated 3 days ago
HAProxy - The Reliable, High Performance TCP/HTTP Load Balancer
WEB SERVERS
1B+ · 1.9K
Pulls: 361,925
Last week
Learn more

Docker compose

- Applications often have multiple containers
    - Backend server, web server, database, static files, etc.

- Orchestrate all containers in one place

- Specify an order of containers to be run on startup

Docker compose

- Create a file named docker-compose.yml
- Run the setup with `docker compose up`
- Stop with `docker compose down`

Example Docker compose for a web app

It has an issue!

What about applying the migrations?

```yaml
services:
  nginx:
    image: nginx:alpine
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf:ro
    ports:
      - '80:80’
    depends_on:
      - backend

  backend:
    build:
      context: .
      dockerfile: Dockerfile
    env_file:
      - .env
    ports:
      - "3000:3000"
    depends_on:
      - db

  db:
    image: postgres:alpine
    volumes:
      - db-data:/var/lib/postgresql/data
    env_file:
      - .env

volumes:
  db-data:
```

Deploying a Docker image

- Deploying a Docker image is very easy!
  - More on deployment next week
- Every major cloud provider has services to directly deploy a Dockerfile or an image
  - e.g., AWS App Runner and GCP Cloud Run
- Great way to quickly deploy your image to the internet
  - Cloud manages the domain, permissions, load balancing, etc.

Serverless functions

- There are even easier ways!
- Just write the functions. Cloud will containerize and deploy it!
    - e.g., AWS Lambda, GCP Cloud Functions, Vercel
- Perfect for deploying a simple service as quickly as possible!

Cloud-based applications

- These days, applications are broken into micro services
- Some services are directly provided by Cloud
  - AWS RDS, GCP Cloud SQL, etc.
  - AWS S3, GCP Cloud Storage, etc.
- Services are either managed in a docker compose (for relatively smaller applications) or in an K8s orchestration

Kubernetes (K8s)

- Open-source container orchestration platform
- Designed for large-scale setups
    - Has scaling, load balancing and clustering features
- Supports automated deployment and rollbacks

Next session

- Hosting your application
- Backend and frontend deployment
- DevOps
- Course conclusion