Sprints

بنك مصر
BANQUE MISR

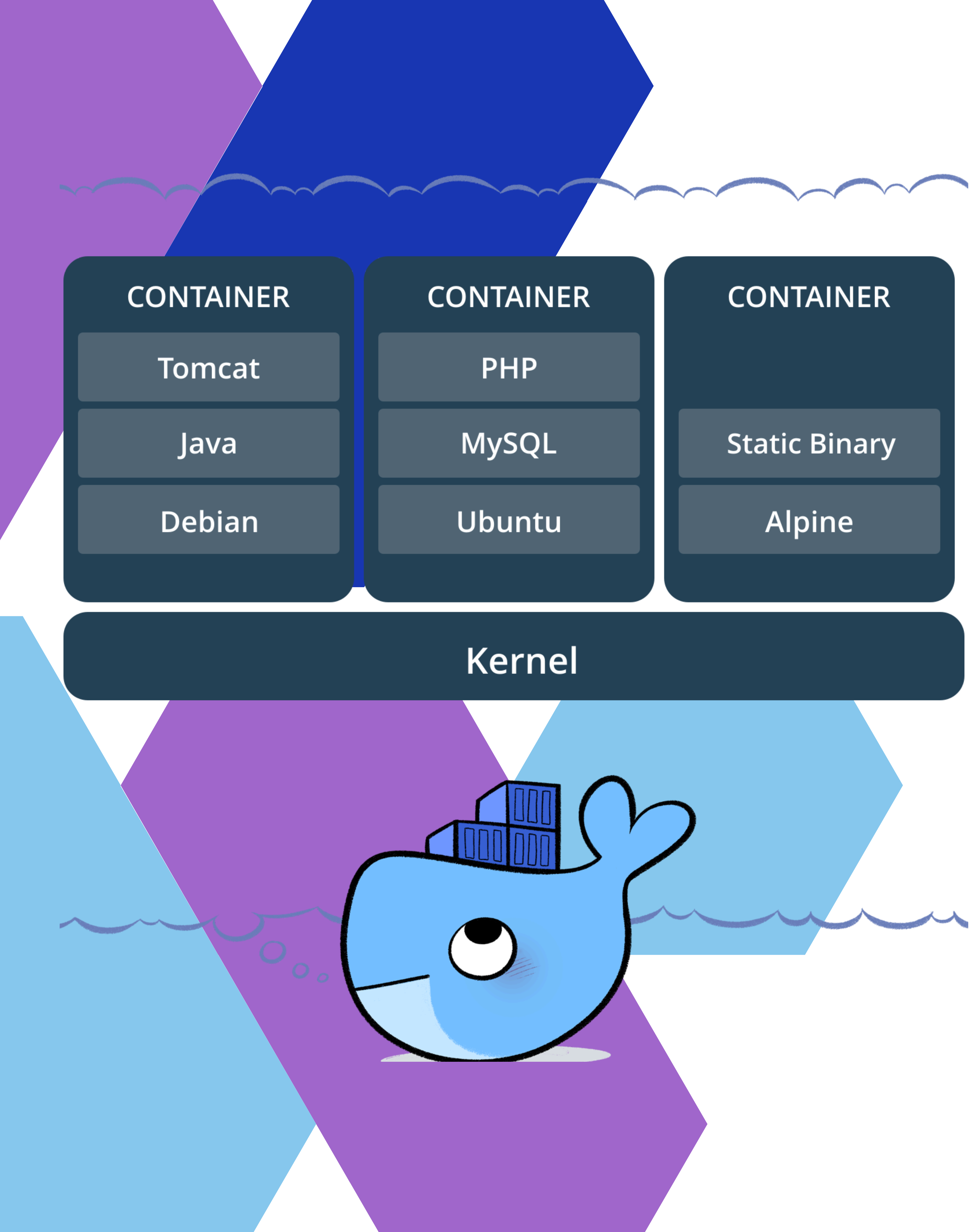
DEV OPS

WEEK 1: DOCKER RECAP

Prepared bt Team 1

Topics Covered

- Docker Containers
- Container Lifecycle
- Docker Images & Dockerfile
- Networking & Volumes in Containers
- Docker Compose Overview



Docker Containers

What are Containers?

A container is a lightweight, standalone, and executable package that includes everything needed to run an application: code, runtime, system tools, libraries, and settings.

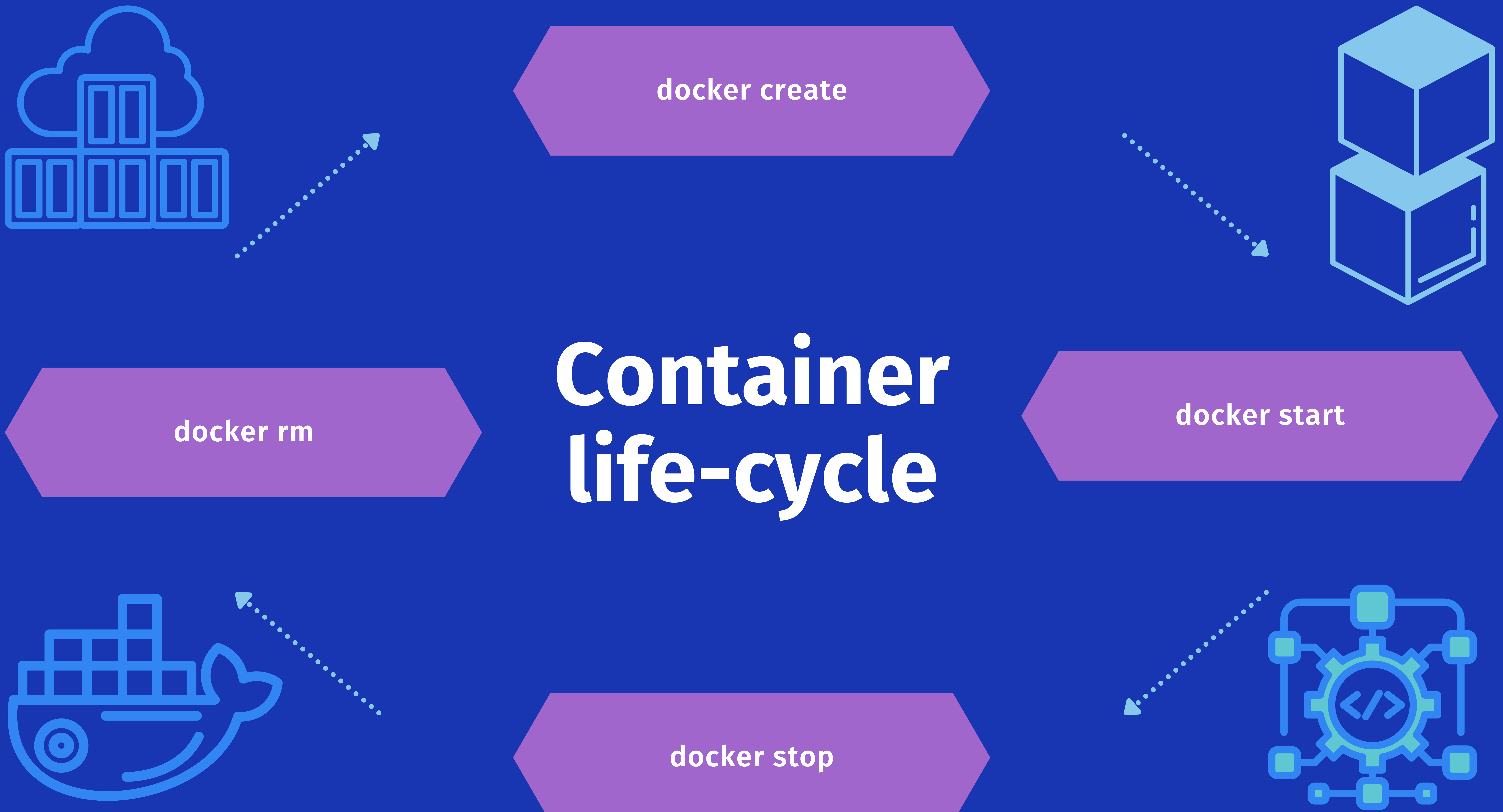
Why It Matters?

Containers allow developers to build once and run anywhere — from a laptop to a production server — without worrying about dependency conflicts.

Container vs. VM

Containerization isolates applications at the process level using the host OS kernel, while virtualization runs entire operating systems on virtual hardware through a hypervisor.

Container life-cycle



Docker Images & Dockerfile

Key Concepts:

- 🧱 **Built in Layers**: Each instruction in the Dockerfile creates a new cached layer.
- 📦 **Immutable**: Once built, an image doesn't change — containers run from images.
- 🔄 **Reusable**: The same image can spin up multiple containers.
- 🐙 **Based on Base Images**: You can start from lightweight images like alpine or full OS images like ubuntu.

Common Instructions in a Dockerfile:

- **FROM** – Sets the base image
- **COPY** – Adds files to the image
- **RUN** – Installs packages or builds code
- **CMD** – Sets the default command for the container

Networking & Volumes in Containers

Networking

```
docker network connect [network] [container]
```

Containers use virtual networks to communicate with each other and the outside world. Docker provides built-in drivers like bridge and host to manage container networking and isolation.

Volumes

```
docker run [image] -v my_volume:[container path]
```

Docker volumes are used to persist data generated and used by containers, even after the container is removed. They are managed by Docker and stored outside the container's writable layer, making them ideal for databases and configuration files.

Docker Compose Overview

Definition	Docker Compose is a tool that allows you to define and manage multi-container applications using a single YAML file (docker-compose.yml).
Key Features	• Define multiple services (e.g., web, database) in one file. • Set up networks, volumes, and environment variables declaratively. • Simplifies orchestration: docker-compose up builds, creates, and starts everything.
Benefits	• Easier local development and testing • Reproducible environments • One command to start, stop, or destroy full app stacks

Thank you!

Ready for the quiz?