

KUBERNETES

CLOUD OPERATING SYSTEM



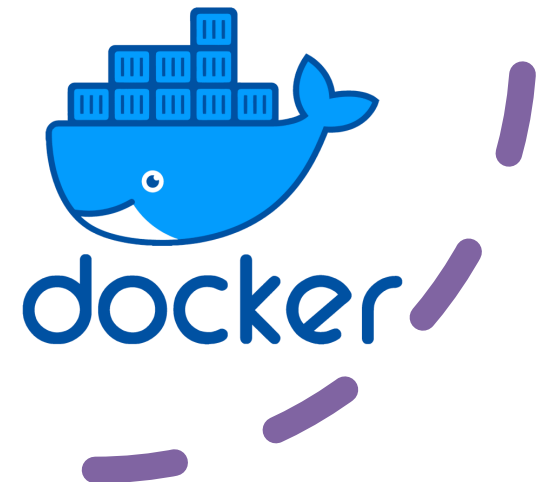
Docker Compose

Instructor: Magdy Salem



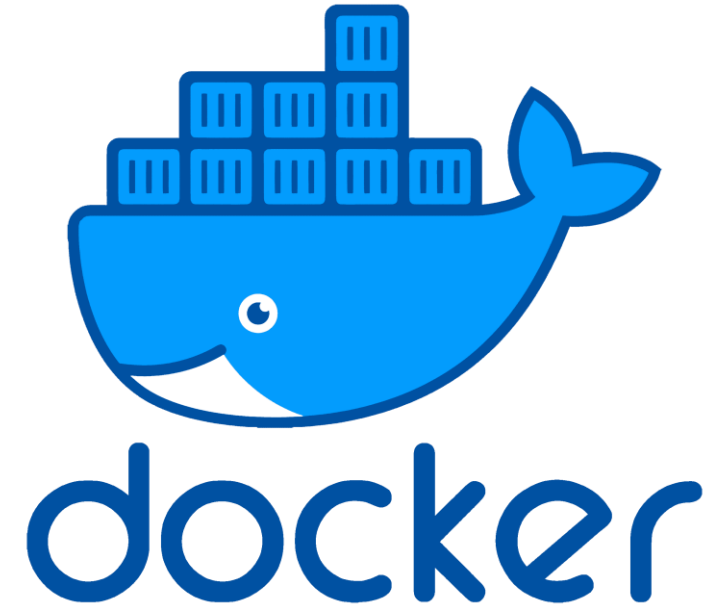
Agenda

- Introduction to Docker Compose
- File structure and syntax
- Basic and advanced usage
- Best practices
- Demo
- Lab



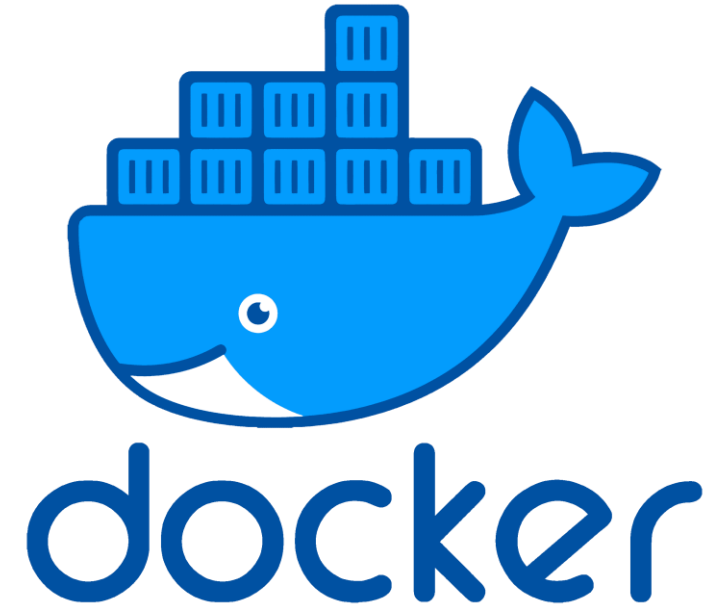
Docker Compose

- Why use Compose: orchestrate multi-container applications
- Declarative service definitions in YAML
- Simplifies development and testing



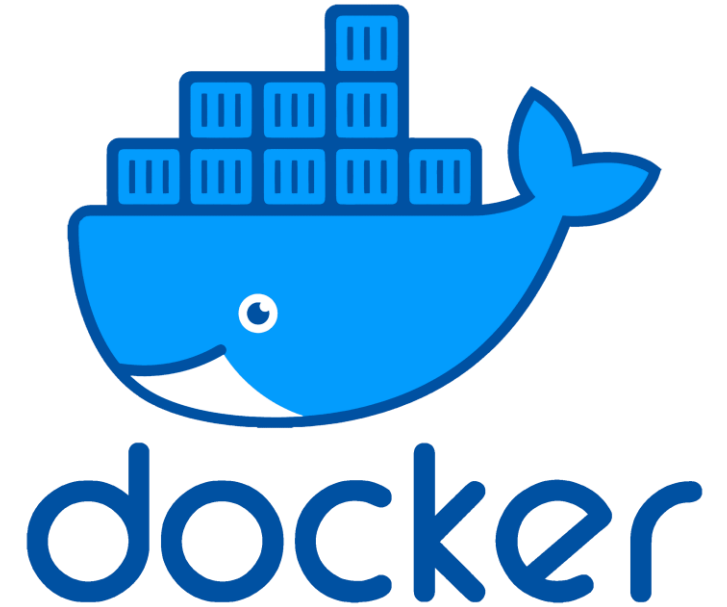
Basic Example: Web + DB Stack

- Compose YAML for Flask + Redis
- Review of directory structure and configuration
- How services talk to each other via servicenames



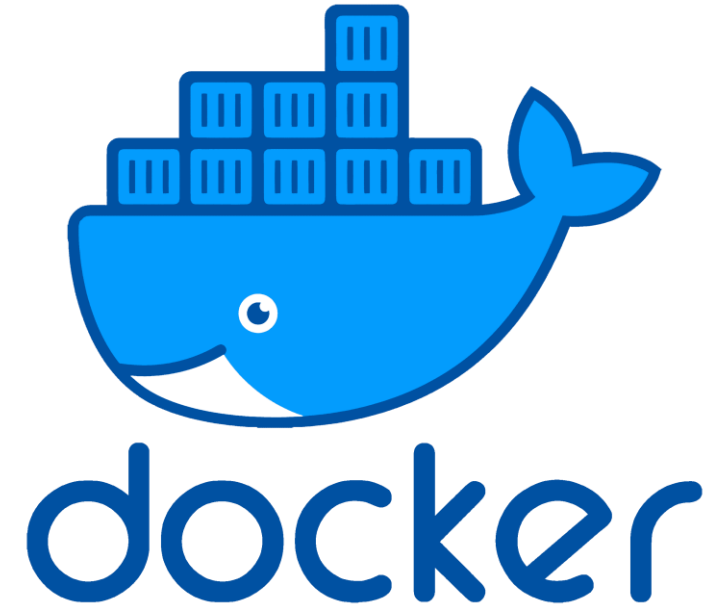
Commands and Lifecycle

- ``docker-compose up`` and ``down``
- Logs, exec, restart policies
- Using Compose in local dev vs CI



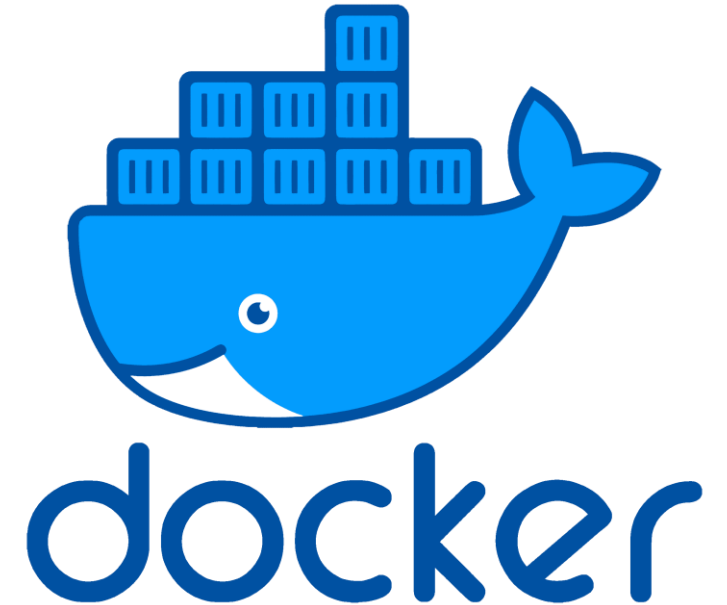
docker-compose up

- Starts and runs all the stack
- Usage Variants:
 - `docker-compose up`
 - `docker-compose up -d`
 - `docker-compose up --build`



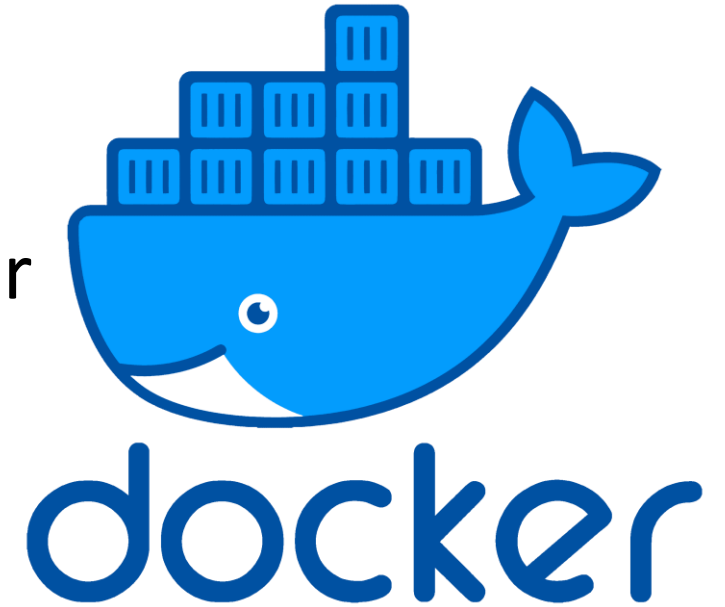
docker-compose down

- Stops and removes all stack.
- Usage Variants:
 - `docker-compose down`
 - `docker-compose down -v`
 - `docker-compose down --rmi all`



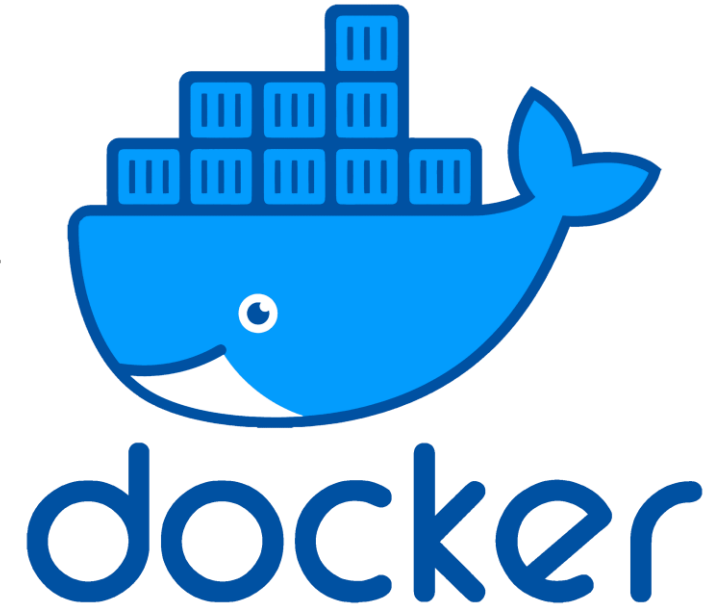
docker-compose logs

- View real-time or historical logs from one or more services.
- Usage Variants:
 - `docker-compose logs`
 - `docker-compose logs web`
 - `docker-compose logs -f`



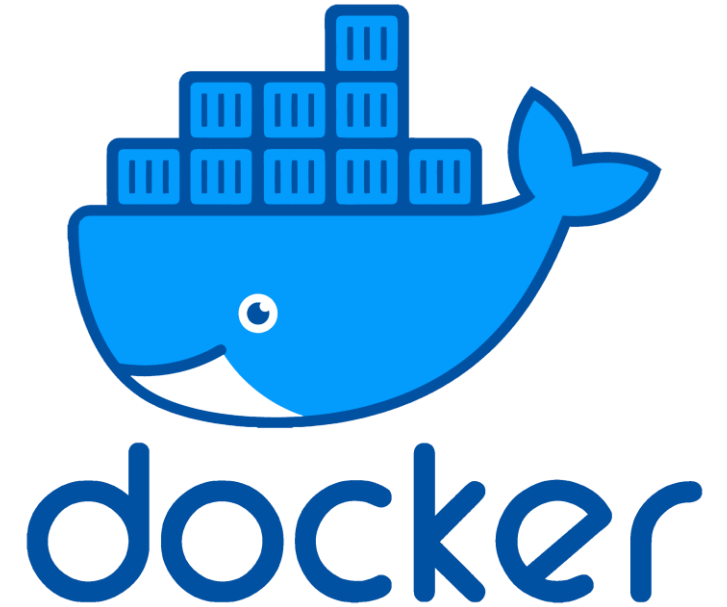
docker-compose exec

- Run interactive commands inside a running service container.
- Usage Variants:
 - `docker-compose exec web bash`
 - `docker-compose exec redis redis-cli`
 - `docker-compose logs -f`



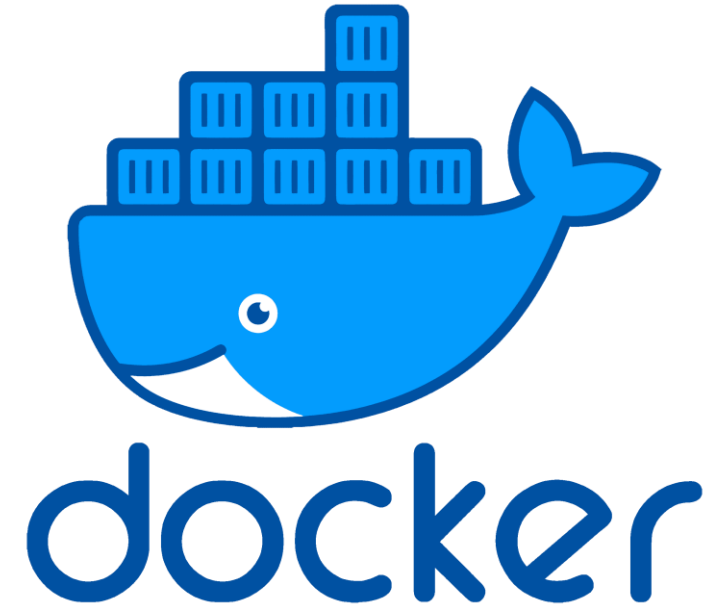
Advanced Features

- **Depends_on** and startup ordering
- Environment variables and **.env** files
- Profiles and override files



Best Practices

- Keep services loosely coupled
- Use named volumes and networks
- Separate dev and QA Compose files
 - `docker-compose.yml`
 - `docker-compose.dev.yml`
 - `docker-compose.qa.yml`
 - `docker compose -f docker-compose.yml -f docker-compose.dev.yml up -d`



Demo



LAB

Lab Github link here

