



# Docker & Shell

Jan 23, 2025

# Whale, Hello There 🙌

Meet your Docker crew



**Ajeet Raina**  
Developer Advocate

 [ajeet\\_raina@docker.com](mailto:ajeet_raina@docker.com)



**James Hammons**  
Reg. Director, Enterprise Sales

 [james.hammons@docker.com](mailto:james.hammons@docker.com)

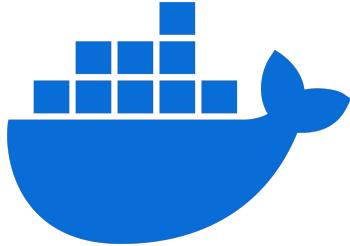


**Sabian**  
Accounts Executive, Enterprise

 [sabian.cooper@docker.com](mailto:sabian.cooper@docker.com)

# Your Instructor

- DevRel at Docker
- Former Docker Captain
- Docker Community Leader
- Distinguished Arm Ambassador
- Worked at Dell EMC, VMware, Redis



@ajeetsraina

# Agenda

## Lab 1 : Inner-Loop Dev Workflow - 60 min

- Overview
- Docker Developer Workflow
- What is a container?
- Container Terminologies
- Docker across the SDLC
- Running PostgreSQL

## Lab 2: Develop and Test a Product Catalog API Service - 60 min

- Overview / Tech Stack
- SDLC Stage - Development
- Benefits of Docker-based Development
- SDLC Stage - Test
- Benefits of Docker-based Testing

## Lab 3: Build Tools and Best Practices

- SDLC Stage - Build
- Docker-based Build Tools and Features
- Docker Init
- Compose Watch
- Best Practices
- Overview of Docker Build Cloud
- Wrapping up



Join at  
**slido.com**  
**#5162 593**



**List of Docker Integrated Tools/CLI you heard in the last 3-4 years?**

# Before we get started

Docker Desktop

# The #1 containerization software for developers and teams

Your command center for innovative container development

Get Started

Download for Mac - Apple Silicon ^

Download for Mac - Intel Chip

Download for Windows

Download for Linux

Commercial use of Docker Desktop requires a license for more than **\$10 million** in annual revenue.

Employees OR more than 100 employees (or Business).

<https://www.docker.com/products/docker-desktop/>

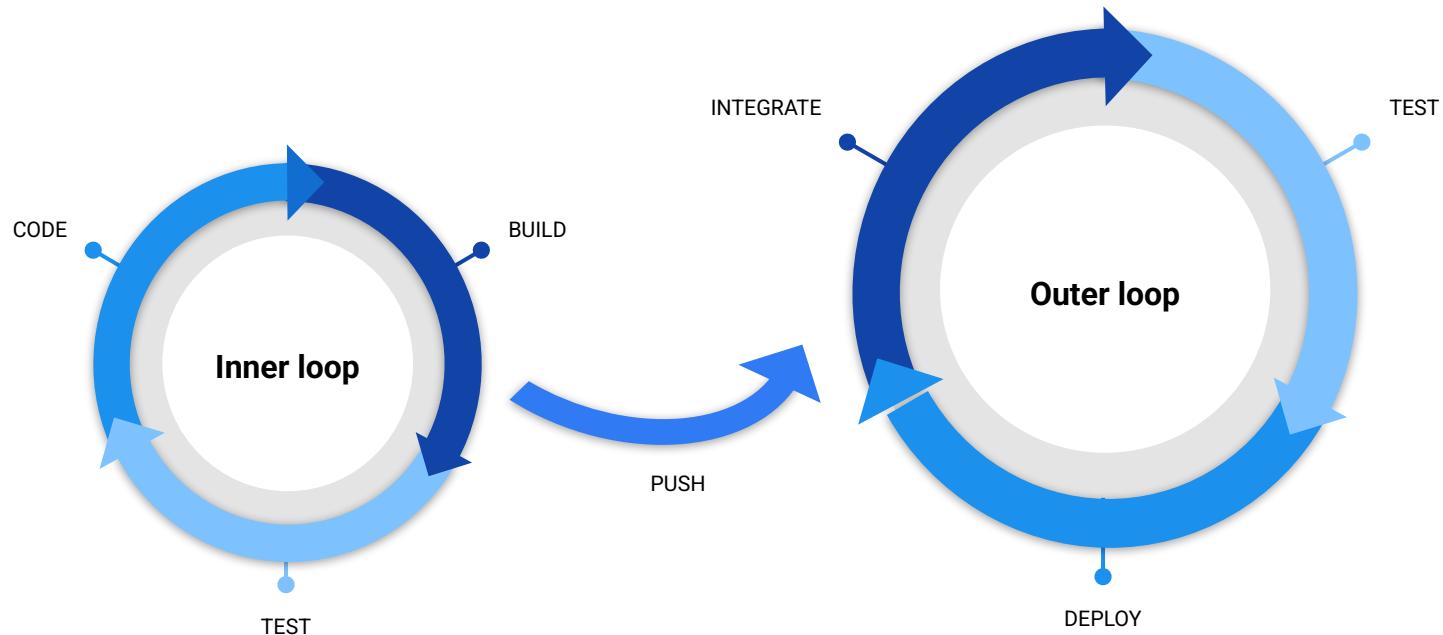
# Access the workshop

<https://dockerworkshop.vercel.app/>

# Inner-Loop Development Workflow



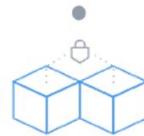
# Inner Loop Vs Outer Loop Development Workflow



# What Modern Developers Success Look Like?



Speed

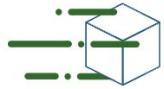


Security



Choice

# What Modern Developers Success Look Like?



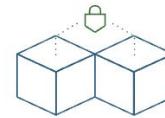
## Speed

- Faster Development
- Rapid Prototyping
- Efficient Builds
- Faster Deployment



## Security

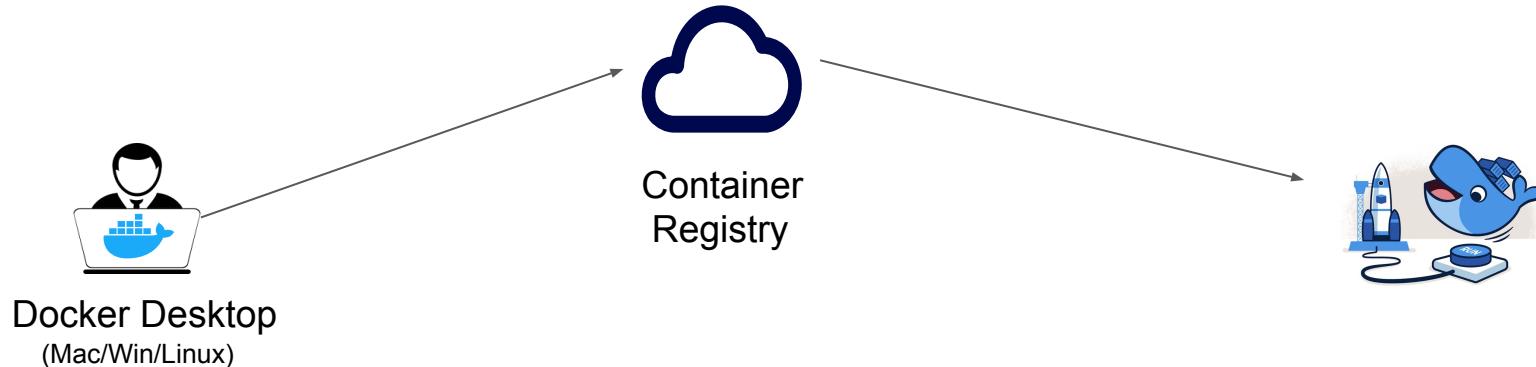
- Supply Chain Security
- Isolation
- Image Security



## Choice

- Platform Compatibility
- Ecosystem Expansion
- Technology Agnostic

# A 30,000 ft View



## BUILD

- Package applications as portable container images
- Create Multi-container apps using Docker Compose

```
$ docker build
```

## SHARE

- Collaborate and distribute via Registry
- Shareable application with clear interface for operators

```
$ docker push
```

## RUN

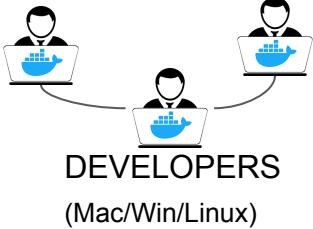
- Run multiple versions of the same application and manage pre-environment settings
- Launch your applications locally and on the cloud with AWS ECS and Azure ACI.

```
$ docker run
```

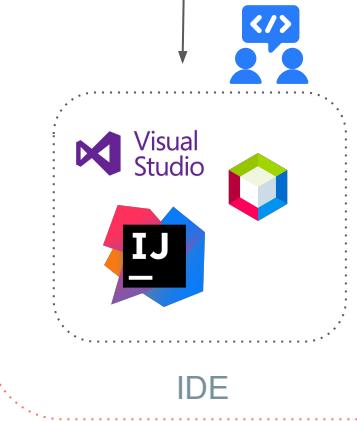


# Inner-Loop Developer Workflow

## BUILD



Code Commit

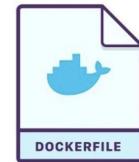


Source Control

SHARE



GitHub Actions



DOCKERFILE



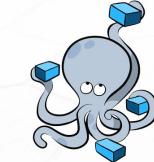
docker



Docker Compose

→

docker push



Define services

```
php:  
  build: php  
  port: "80:80"  
  ports: "441:443"  
  volumes:  
    - /path/www:/var/www/html  
  links:  
    - db
```

\$ docker-compose up



Testing app



Container Registry

RUN



Google Cloud



# Streamline your development practice

A suite of solutions supporting great developer experiences with enterprise control



How do I develop with  
and use containers?



How do I find and share  
container images?



How do I build compliant  
container images?



How do I make my  
image builds faster?



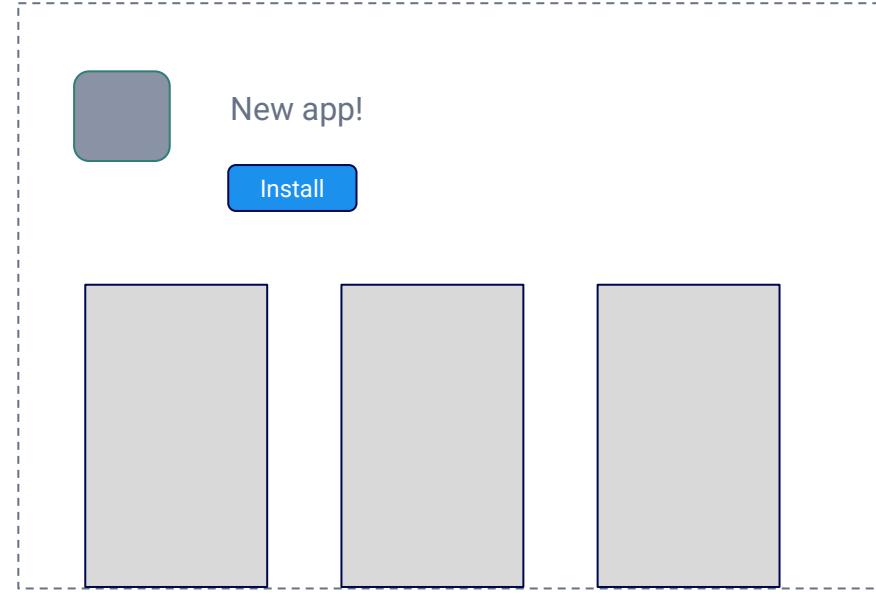
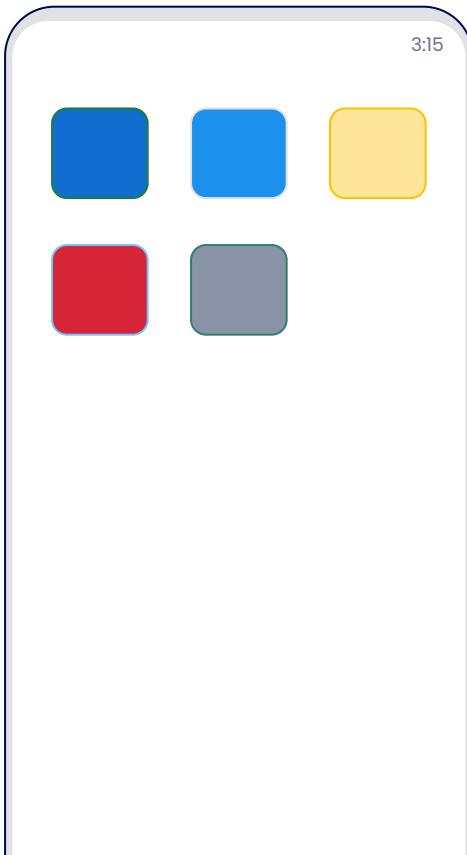
How can I run my  
resource-heavy services?





# What is a container?

# Like smartphone apps...



# Running PostgreSQL, the old way

PostgreSQL: Downloads

November 21, 2024: PostgreSQL 17.2, 16.6, 15.10, 14.15, 13.18, and 12.22 Released!

## Downloads

### PostgreSQL Downloader

PostgreSQL is available for download in various formats. If you want to build it yourself.

### Packages and Installation

Select your operating system:

- Linux (Icon: Penguin)
- Solaris (Icon: Sun)

### Source code

The source code can be found at [postgresql.org/ftp/source/v17.2/](http://postgresql.org/ftp/source/v17.2/). Instructions for building from source are available.

### Beta/RC Releases

There are source code and binary packages for evaluation of new features.

PostgreSQL: File Browser

November 21, 2024: PostgreSQL 17.2, 16.6, 15.10, 14.15, 13.18, and 12.22 Released!

## File Browser

Top → source → v17.2

### Directories

[Parent Directory]

### Files

- postgresql-17.2-docs.tar.gz
- postgresql-17.2.tar.bz2
- postgresql-17.2.tar.bz2.md5
- postgresql-17.2.tar.bz2.sha256
- postgresql-17.2.tar.gz
- postgresql-17.2.tar.gz.md5
- postgresql-17.2.tar.gz.sha256

PostgreSQL - PostgreSQL: Downloads

GitHub - peterc/homebrew-postgresql

peterc / homebrew-postgresql Public

Code Issues 13 Pull requests Actions Projects Wiki Security Insights

master 2 Branches 3 Tags Go to file Code

PostgreSQL minor releases 1247ce · 2 months ago 240 Commits

.cirrus.yml postgresql@16: New 2 years ago

README.md README: Add note about old versions 4 years ago

formula\_renames.json Migrate to versioned formula scheme 8 years ago

postgresql-common.rb Add license fields 5 years ago

postgresql@10.rb PostgreSQL minor releases 3 years ago

postgresql@11.rb PostgreSQL minor releases 2 years ago

postgresql@12.rb PostgreSQL minor releases 2 months ago

postgresql@13.rb PostgreSQL minor releases 2 months ago

postgresql@14.rb PostgreSQL minor releases 2 months ago

postgresql@15.rb PostgreSQL minor releases 2 months ago

postgresql@16.rb PostgreSQL minor releases 2 months ago

postgresql@17.rb PostgreSQL minor releases 2 months ago

postgresql@8.3.rb Add livecheck blocks 4 years ago

About PostgreSQL formulae for the Homebrew package manager

homebrew postgresql

Readme Activity 295 stars 9 watching 34 forks Report repository

Releases 3 bottles-201506260 (Latest) on Jun 26, 2015 + 2 releases

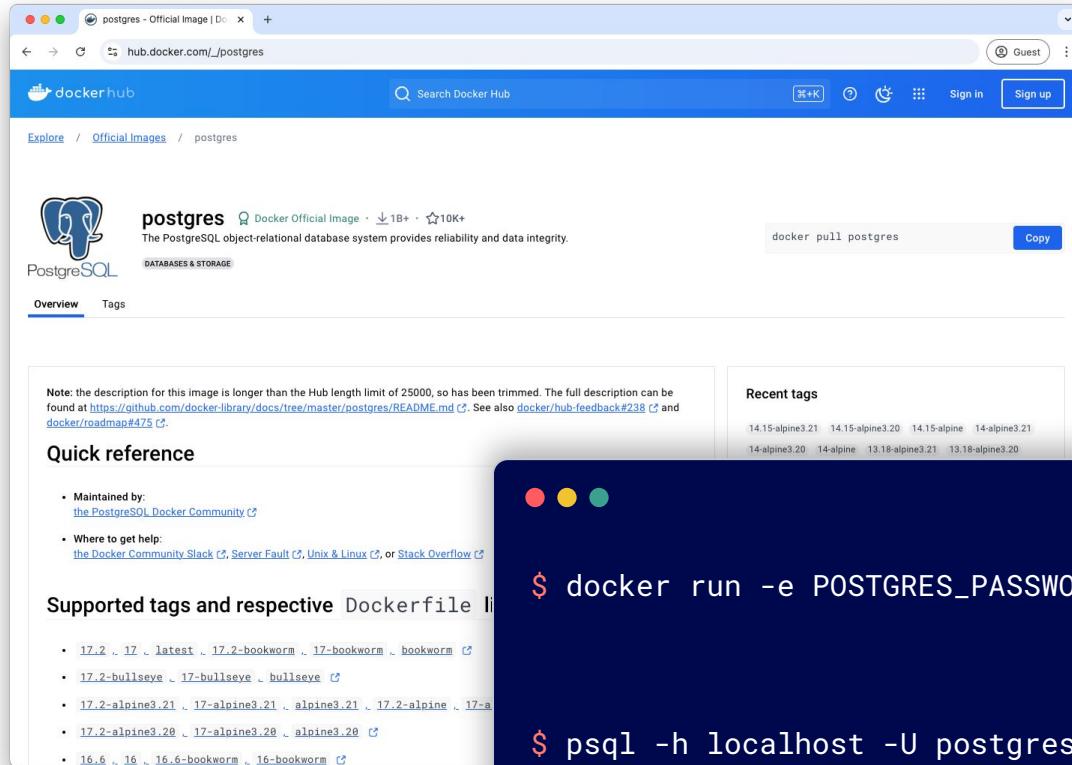
Packages No packages published

Contributors 7

Policies | Code of Conduct | About PostgreSQL | Contact

Copyright © 1996-2025 The PostgreSQL Global Development Group

# Example - Running PostgreSQL



The screenshot shows the Docker Hub website with the URL [hub.docker.com/\\_/postgres](https://hub.docker.com/_/postgres). The page displays information about the **postgres** Docker Official Image, which has over 1B+ pulls and 10K+ stars. It provides a quick reference for maintaining the image and getting help. A note states that the full description is longer than the Hub length limit of 25000. The page lists supported tags, including 17.2, 17, latest, 17.2-bookworm, 17-bookworm, bookworm, 17.2-bullseye, 17-bullseye, bullseye, 17.2-alpine3.21, 17-alpine3.21, alpine3.21, 17.2-alpine, 17-alpine3.20, 17-alpine3.20, 17-alpine3.20, alpine3.20, 16.6, 16, 16.6-bookworm, and 16-bookworm. A search bar at the top right contains the command `docker pull postgres`, with a `Copy` button next to it.

CLI

```
$ docker run -e POSTGRES_PASSWORD=dev -p 5432:5432 postgres:17.2
```

```
$ psql -h localhost -U postgres
```



# Demo time!

- ✓ Finding images on Docker Hub
- ✓ Ease of downloading and running a containerized service
- ✓ Ability to run multiple versions side-by-side with no conflicts

# Container terminology



## **Container = Isolated Process**

Not a virtual machine. Just a process.  
Runs independent of other containers  
and what's on the host machine



## **Image = Standard packaging**

Contains all binaries, files, dependencies,  
and configuration needed to run the  
containerized process



## **Registry = Image repository**

A centralized location for the hosting and  
distribution of container images. Can be  
available publicly or privately.

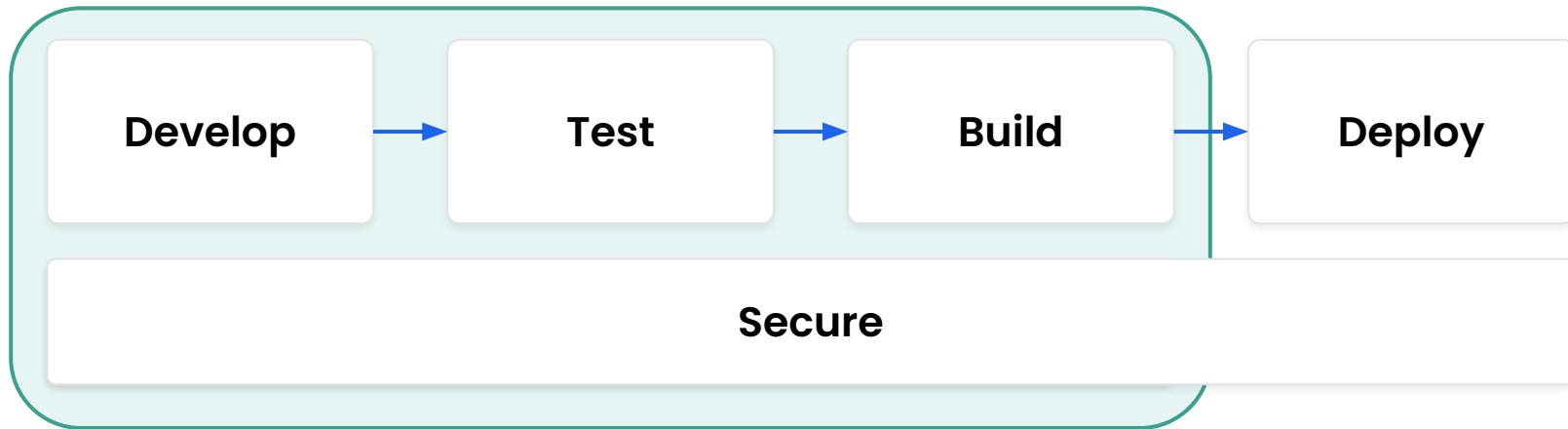


Docker provides you an entire ecosystem of building blocks



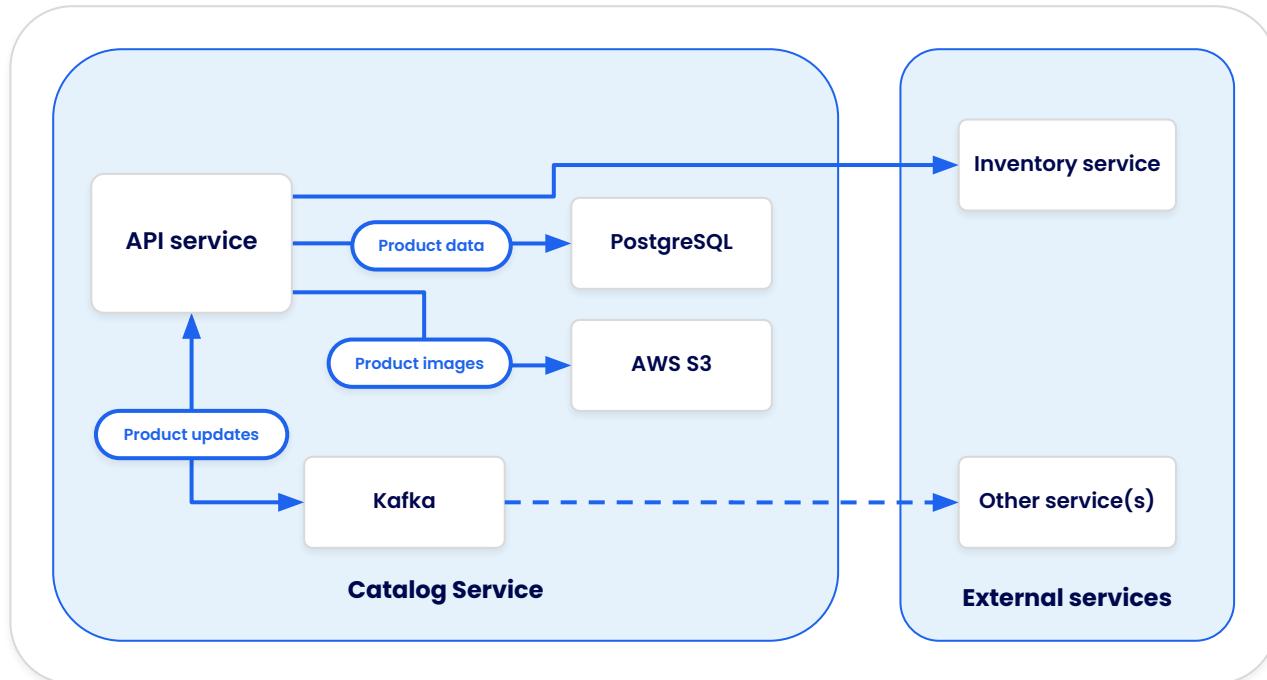
# Docker across the SDLC

# The SDLC



# Sample app = Catalog API

- Data is stored in a **PostgreSQL database**
- Product images stored in a **AWS S3 bucket**
- Inventory data comes from an **external inventory service**
- Product update events are published to a **Kafka cluster**



# The ticket

Add Objective / DEVR-981

## Add UPC code to product\_created event

+ Add Apps

### Description

As a downstream consumer of catalog events, I would like to have the UPC code included in the product\_created event so I don't need to look it up immediately after the event is received.

### Activity

Show: Comments ▾

Add a comment...

Pro tip: press M to comment

In Progress Actions Improve issue

Details	
To be done by	None
Assignee	Moby Dock
Parent	None
Sprint	S24: Sprint Awesome!
Development	<a href="#">Create branch</a> <a href="#">Create commit</a>
Reporter	Moby Dock
More fields Labels	
Automation ⚡ Rule executions	
OKRS Profit.co OKRs	
Connector for Salesforce Associations	

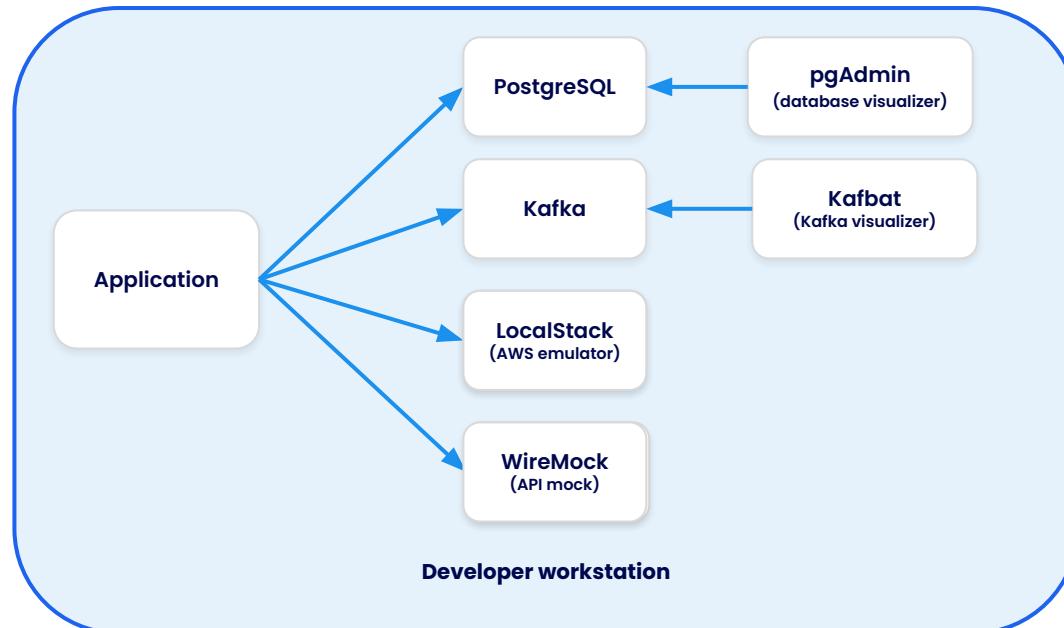
Created December 4, 2024 at 7:55 AM Updated December 9, 2024 at 9:20 AM

Configure



# SDLC Stage - Development

# The development environment



# The development tools and features



Docker Engine

Container networking



Docker CLI/GUI

Port forwarding

Docker Compose

Docker Debug

Volume mounts

Synchronized file shares

Settings management

Enhanced container isolation

Registry Access Management

Image Access Management

Air-gapped containers

# Demo time!

- ✔ Ease to setup and launch the development environment
- ✔ Ability to use additional tools to help troubleshoot and validate changes
- ✔ Ability to use containers in a hybrid setup (app is running natively)

# Benefits of Docker-based development



## Faster onboarding

Less time setting up.  
Less time switching projects.



## Consistency everywhere

No more "it worked for me".  
Less time rolling out env changes.



## Project-specific tooling

Less coupling on external services.  
Less time troubleshooting issues.



## Enterprise ready

Dev workstations kept secure.  
Satisfied organizational policies.

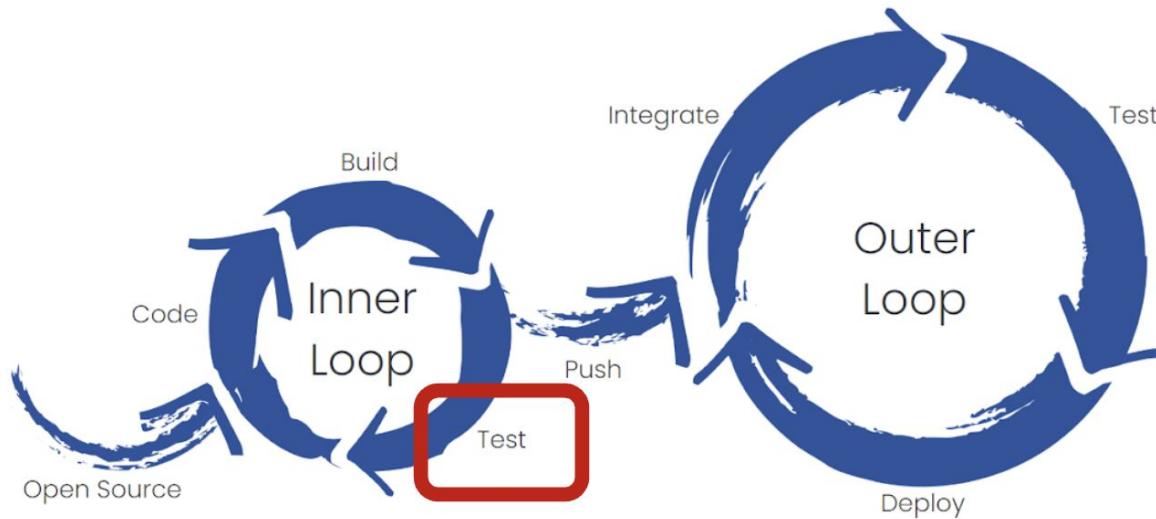




# SDLC Stage - Testing

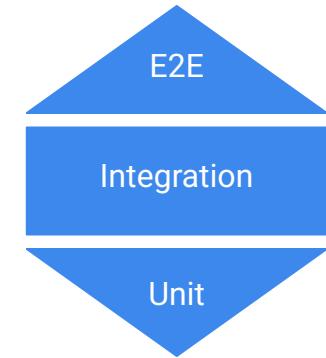
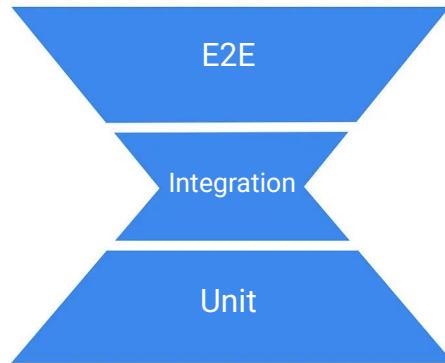
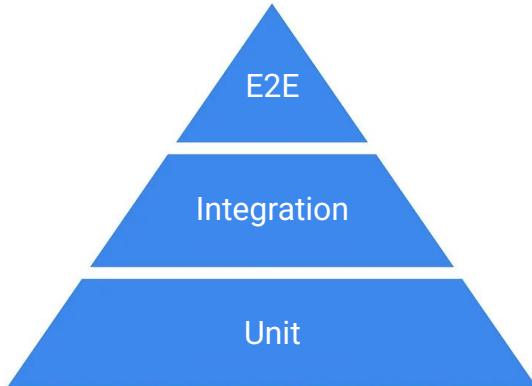
# Shifting left with Testcontainers

95% of developers write tests within their inner development loop



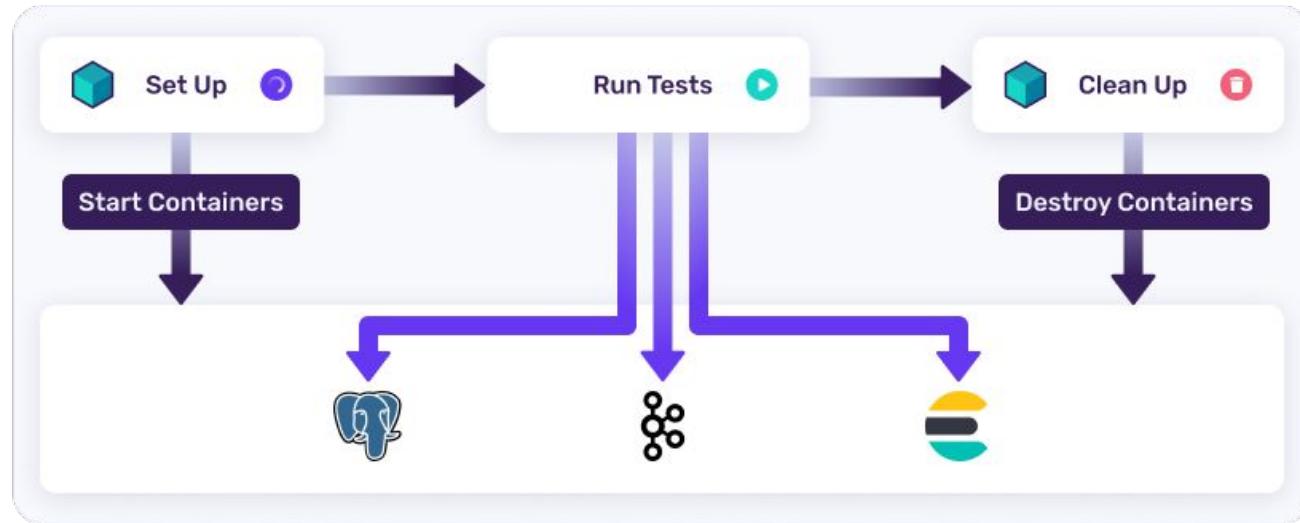
[State of Local Development and Testing](#)

# The shapes of testing



# Introducing Testcontainers

Open source collection of libraries to programmatically control the lifecycle of containers during testing



```
var postgres = new PostgreSQLContainer<>(DockerImageName.parse("postgres:17.2"));
```

# Run the containers anywhere!

Run containers locally



**Testcontainers  
Desktop** by docker.

Great for local development  
and validation

Run with cloud resources



**Testcontainers  
Cloud** by docker.

Great for CI pipelines or when more  
resources are needed locally

# Demo time!

- ✓ Complete control of services during testing
- ✓ Ability to run containers for testing locally or remotely
- ✓ Consistency of test results whether running locally or in CI pipelines

# Benefits of Docker-based testing



## Use real services

Increased deployment confidence.  
Reduced code complexity.



## Consistent testing envs

Reduced test failures.  
Faster test failure resolution.



## Dynamic scaling

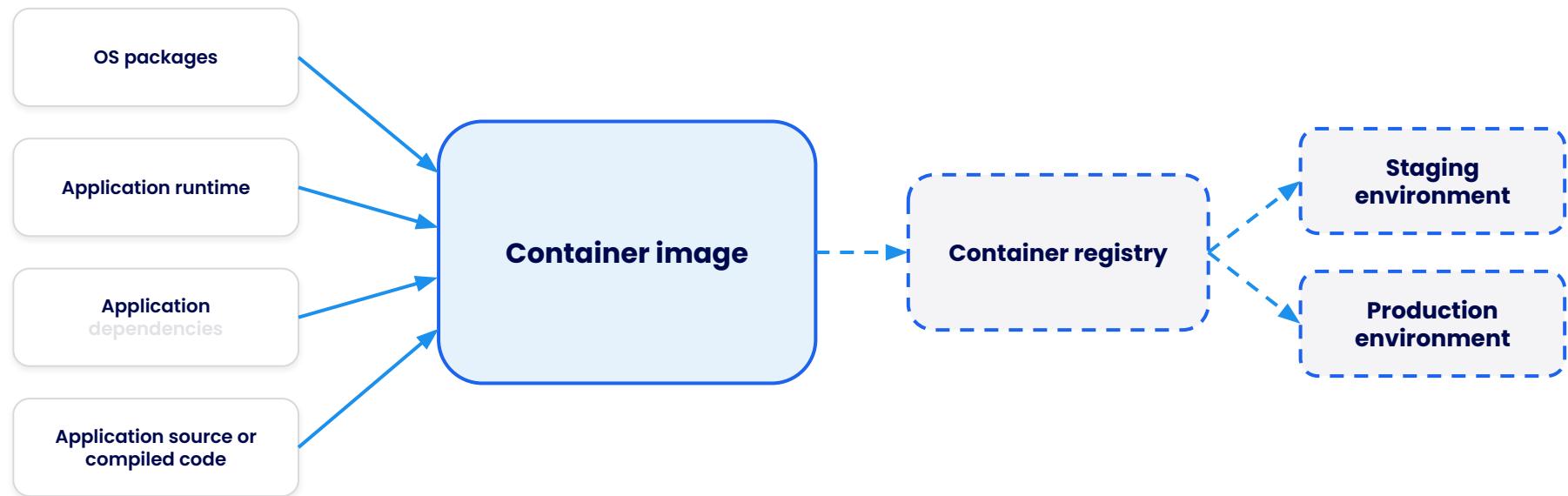
Reduced testing infra costs.  
Reduced execution times.





# SDLC Stage - Build

# Building our application



# The build tools and features



Docker Engine

Multi-stage builds

Registry Access Management



Docker CLI/GUI

Build caching

Image Access Management



BuildKit/Buildx

Multi-architecture builds

Docker GitHub Actions

Docker init

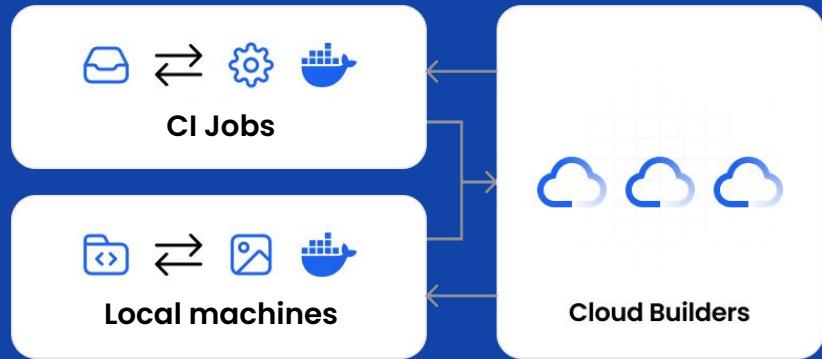
Dockerfiles

Build secrets

Build insights and logs

# Benefits of Build Cloud

- Shared build cache
- Native multi-architecture builds
- Seamless integration with your CI tools
- Minimal changes in local development



# Demo time!

- ✓ Docker Hub provides trusted base images to provide extension points
- ✓ Docker Build Cloud dramatically improves build speeds
- ✓ Build insights provide logs and input for troubleshooting and debugging

# Benefits of Docker-based building



## Time savings

Decreased build times.  
Decreased machine resources.



## Consistency of tooling

Reduced tool sprawl.  
Increased build confidence.



## Useful insights

Reduced time to recover.  
Increased best practice adoption.





# SDLC Stage – Secure

# Example policies

Is our base image current?

Is our base image from a trusted source?

Is the default user a non-root user?

Are there any high-profile vulnerabilities?

**Container image**

Do we have any open-source license issues?

Are there any fixable critical vulnerabilities?

Does our image have build provenance?

Does our image have an attached SBOM (software bill of materials)?

# The secure tools and features



CLI tooling

Build provenance  
generation

Environment  
recording



GUI integrations

Change notifications

Image comparison

Policy analysis

Remediation  
guidance

Official GitHub  
Actions

SBOM generation

Vulnerability  
assessment

External registry  
integrations

# Demo time!

- ✓ Docker Scout helps identify policy violations without waiting for CI
- ✓ Docker Scout provides guidance to issues, whether from base images or extensions
- ✓ CI integrations help provide feedback early and often



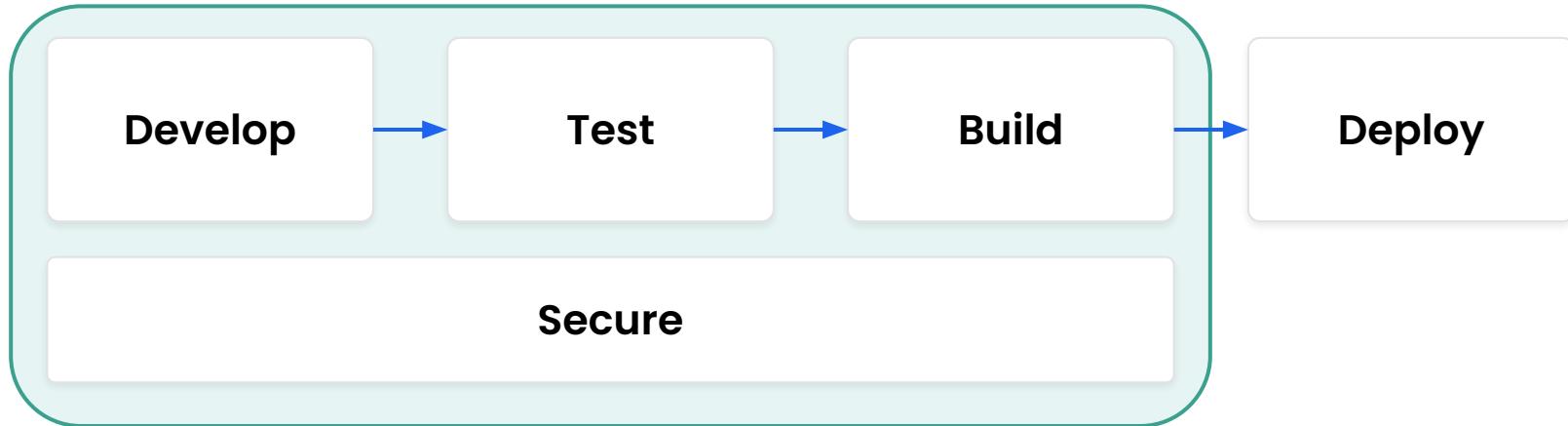
# Recap

# Docker's vision

**Increase the time every  
product development team  
spends on innovation**



# The SDLC

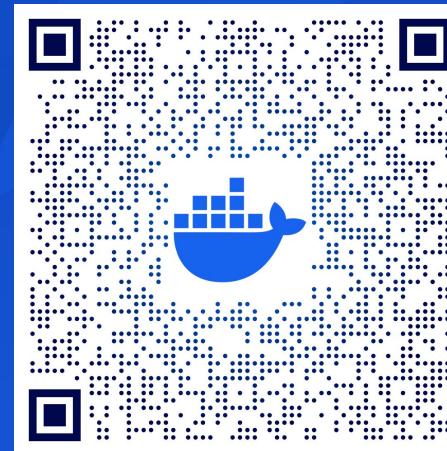




# Thank you!



**Sample project repo**  
[github.com/dockersamples/catalog-service-node](https://github.com/dockersamples/catalog-service-node)



**This slide deck**