



**Building Fast,
Securing More,
Testing Better.**

How Docker Fuels Developer Productivity

Jan 18, 2025

Hello There



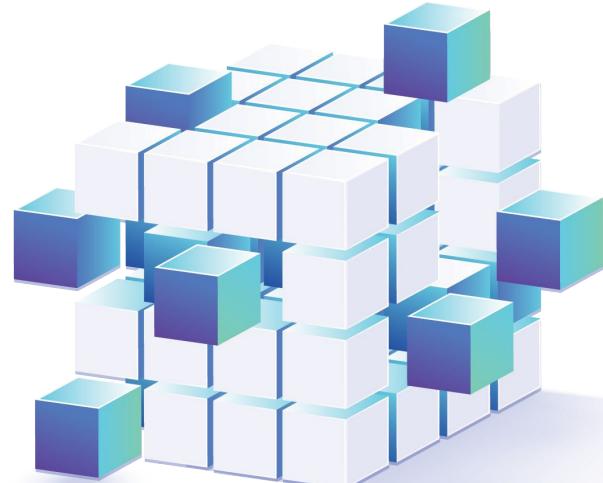
Ajeet Raina
Developer Advocate
Docker



Rebant Malhotra
Accounts Executive, Enterprise
Docker

Today's agenda

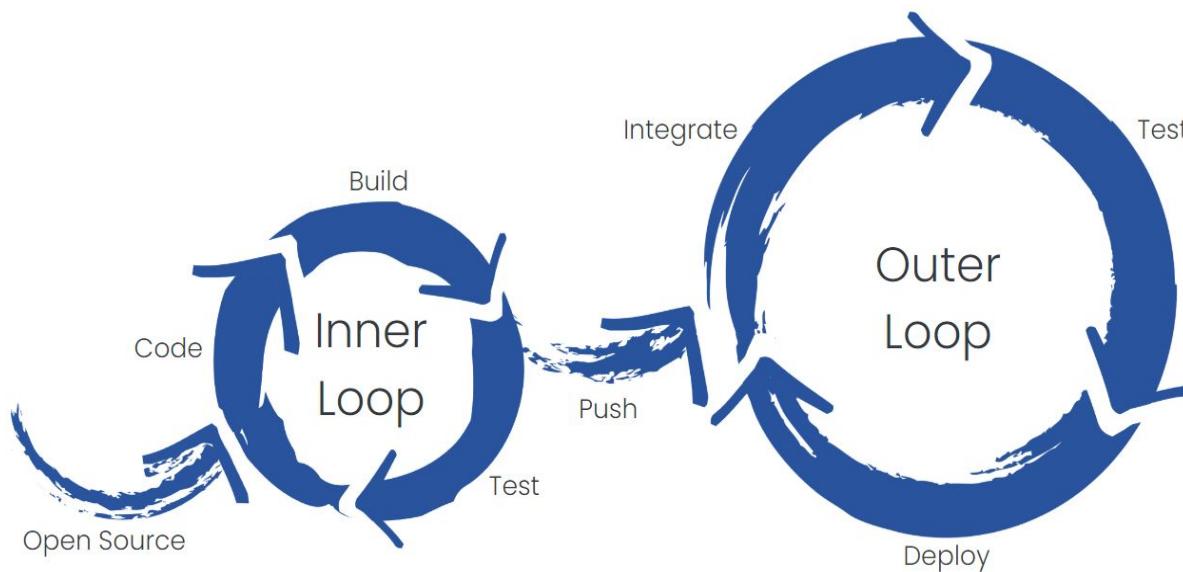
01. Inner-Loop Development Workflow
02. What is a container?
03. Docker across the SDLC
 - a. Develop
 - b. Test
 - c. Build
 - d. Secure
04. Q&A





Inner-Loop Development Workflow

Docker is Uniquely Focused on Developer Success



Trusted Images

Docker Desktop

Docker Ecosystem

Delivery Platforms

20M+ Active Developers

450+ Trusted Partners



Empowering the Modern Developers



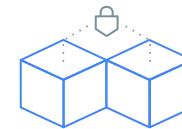
Speed

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



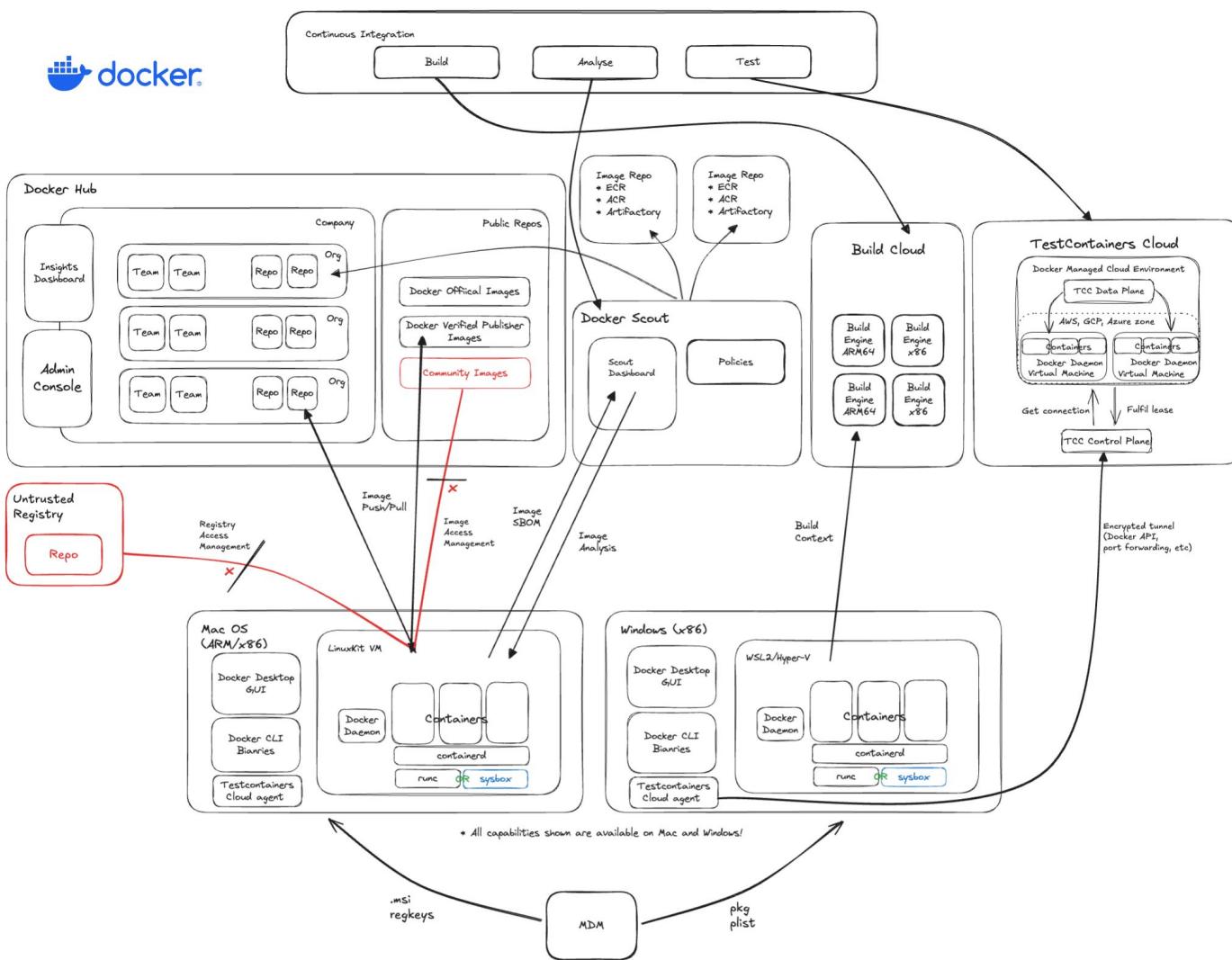
Security

- Supply Chain Security
- Isolation
- Image Security

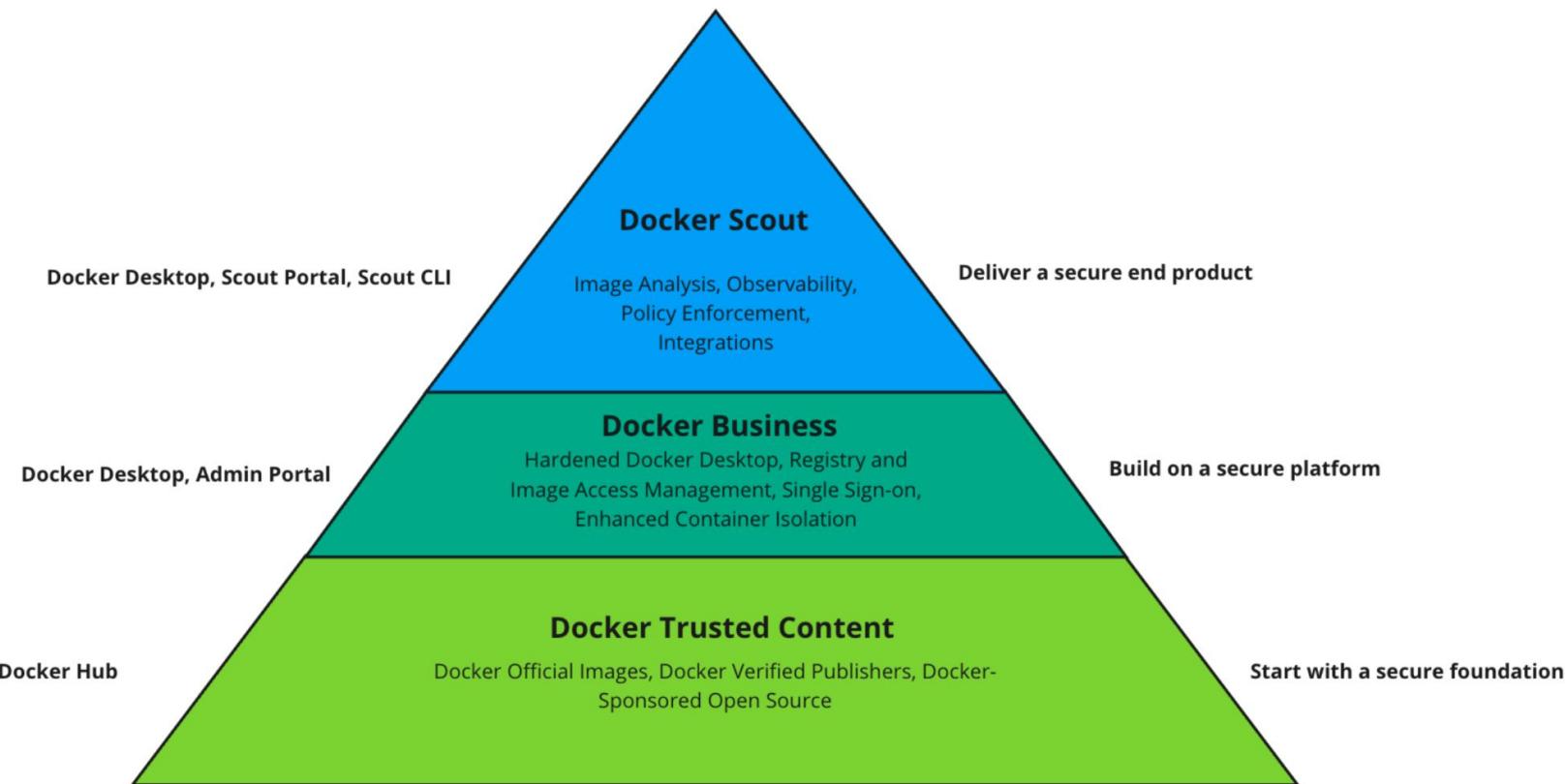


Choice

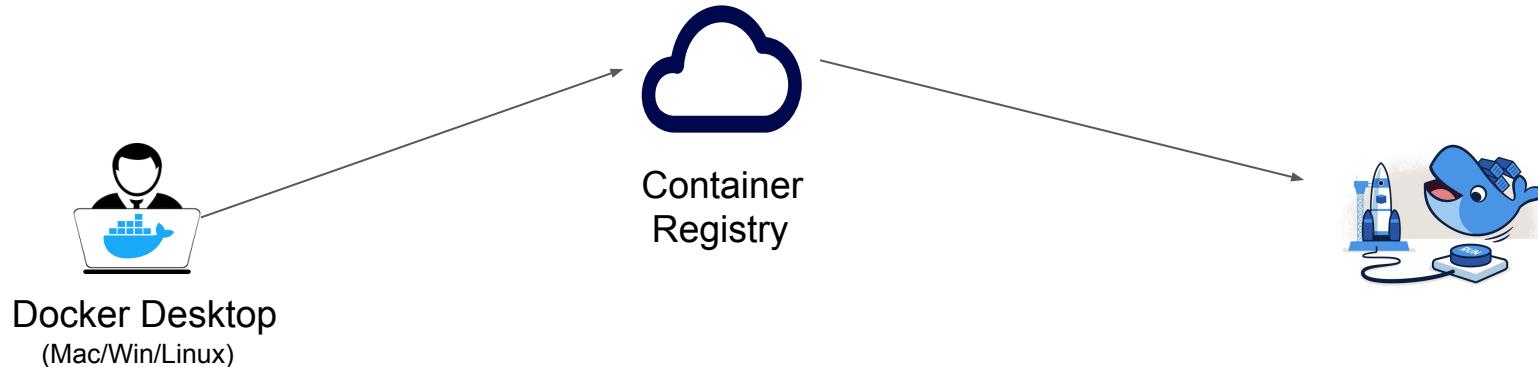
- Platform Compatibility
- Ecosystem Expansion
- Technology Agnostic



Layered Approach to Security



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

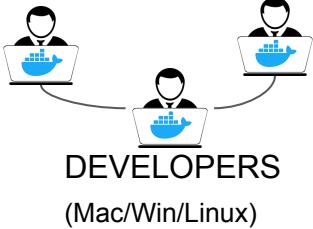
- Run multiple versions of the same application and manage pre-environment settings
- Launch your applications locally and on the cloud with AWS ECS and Azure ACI.

```
$ docker run
```

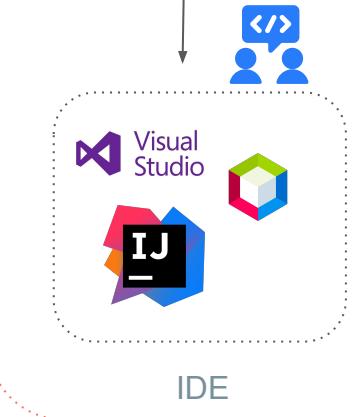


Inner-Loop Developer Workflow

BUILD



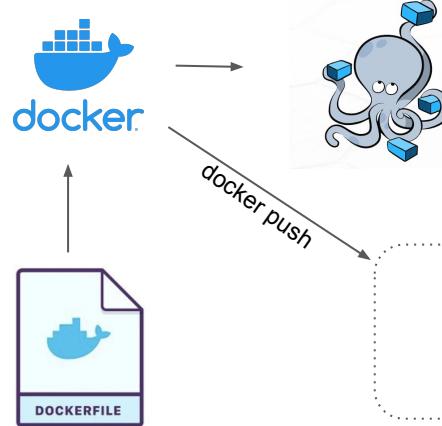
Code Commit



Source Control



GitHub Actions



SHARE

Define services

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

\$ docker-compose up

Container Registry



Testing app

RUN



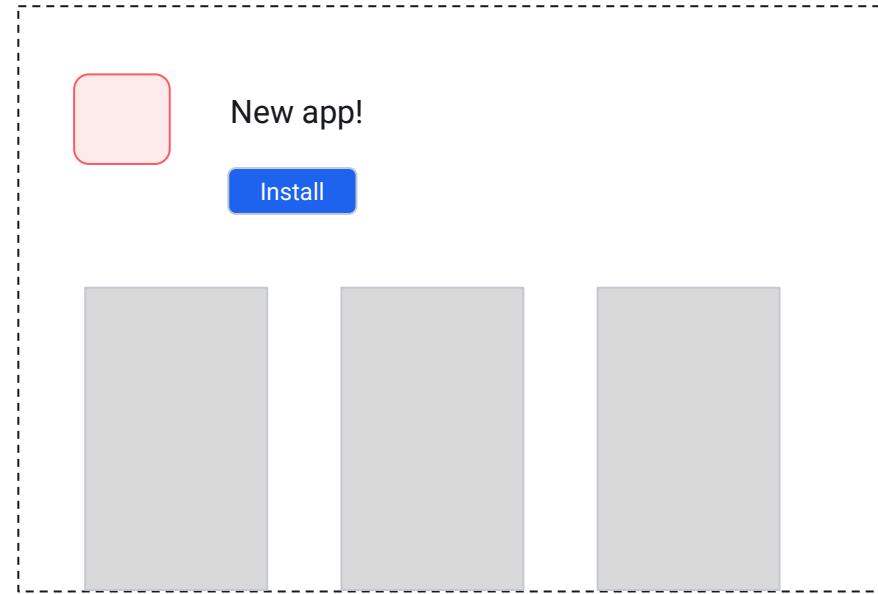
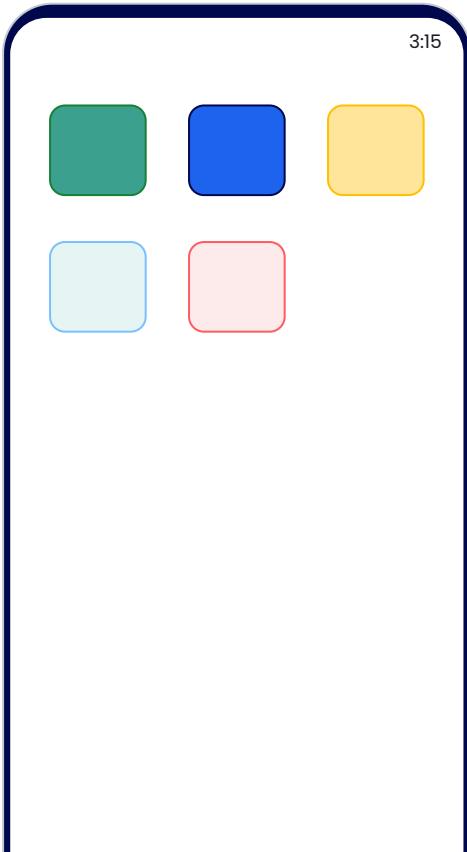
Google Cloud





What is a container?

Like smartphone apps...



Running PostgreSQL, the old way

The image shows a Mac desktop with two open browser windows. The left window is a PostgreSQL download page from [postgresql.org/download/](https://www.postgresql.org/download/). It features a 'Downloads' section with links for Linux and Solaris, a 'Source code' section, and a 'Beta/RC Releases' section. The right window is a GitHub repository for Homebrew PostgreSQL, specifically the [peter/re/homebrew-postgresql](https://github.com/pete/re/homebrew-postgresql) branch. The GitHub interface shows a list of commits by user 'peter' for PostgreSQL minor releases, with details like commit hash, date, and file changes.

PostgreSQL: Downloads

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

Downloads

PostgreSQL Download

PostgreSQL is available for download or you can want to build it yourself.

Packages and Installation

Select your operating system:

- Linux
- Solaris

Source code

The source code can be found here. Instructions for building from source are available.

Beta/RC Releases

There are source code and documentation for evaluation of new features.

GitHub - peter/re/homebrew-postgresql

peter / 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 1247c1e · 2 months ago 240 Commits

File	Description	Age
.cirrus.yml	postgres@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

Readme Activity 295 stars 9 watching 34 forks Report repository

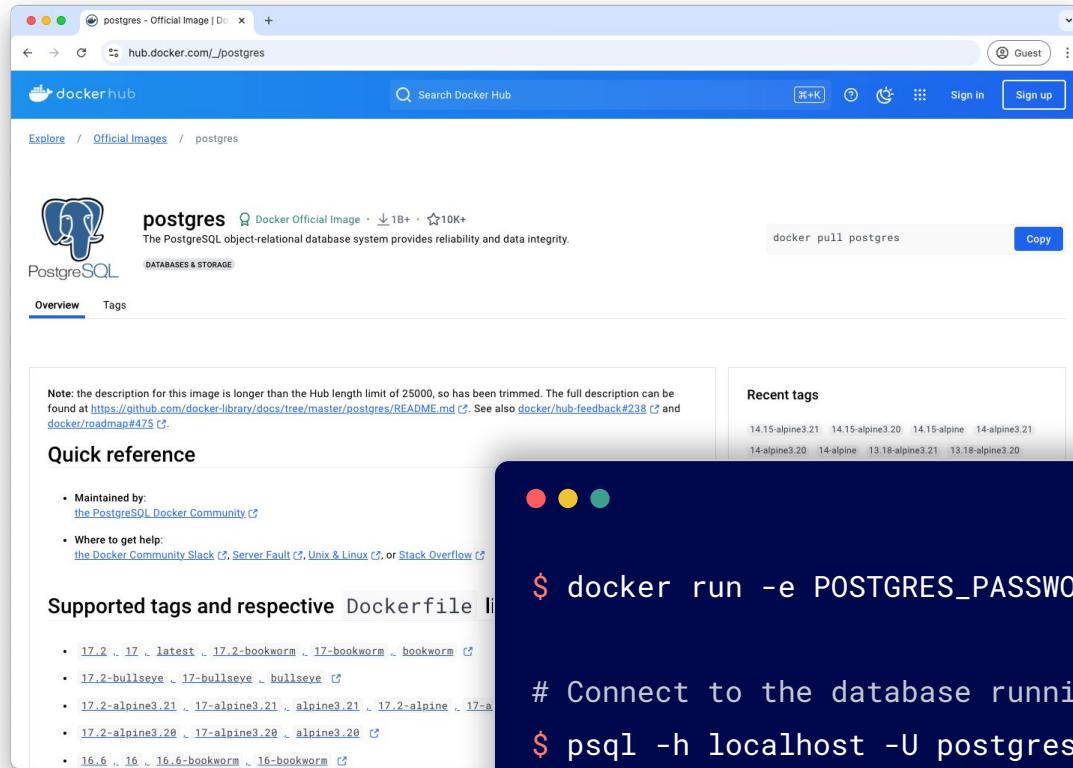
Releases 3 bottles-201506260 (Latest) on Jun 26, 2015 + 2 releases

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. The page displays the official Docker image for PostgreSQL. It includes the PostgreSQL logo, a brief description, and a "Quick reference" section with links to the Docker Community and Stack Overflow. Below this is a "Supported tags and respective Dockerfile links" section listing various tag versions. A search bar at the top right contains the command `docker pull postgres`.

Recent tags

- 14.15-alpine3.21
- 14.15-alpine3.20
- 14.15-alpine
- 14-alpine3.21
- 14-alpine3.20
- 14-alpine
- 13.18-alpine3.21
- 13.18-alpine3.20

CLI

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

# Connect to the database running in the container
$ 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.

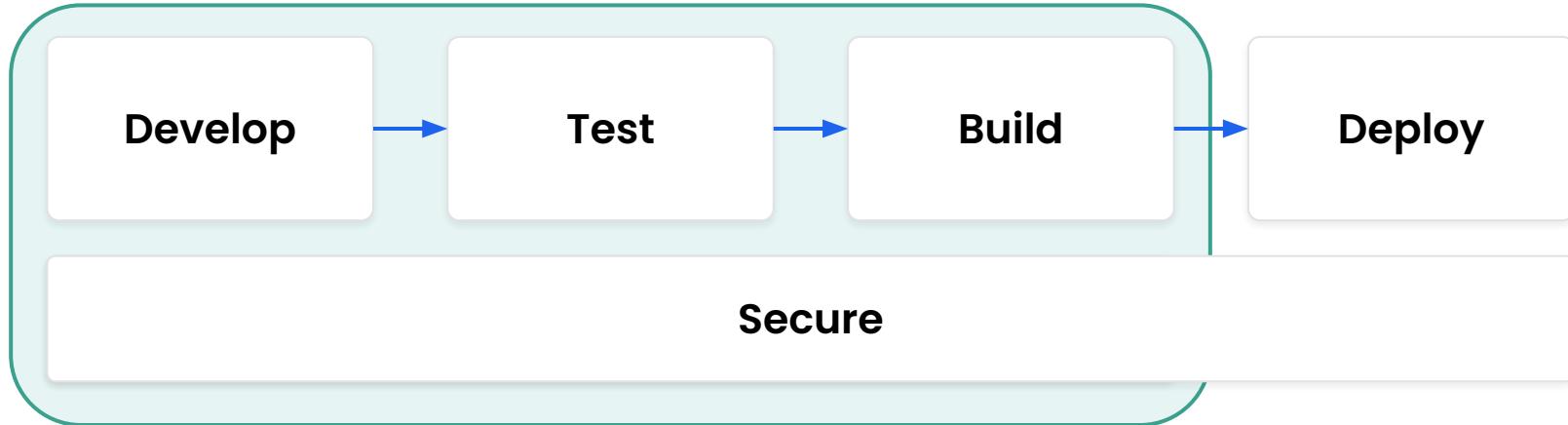


Docker provides you an entire ecosystem of building blocks



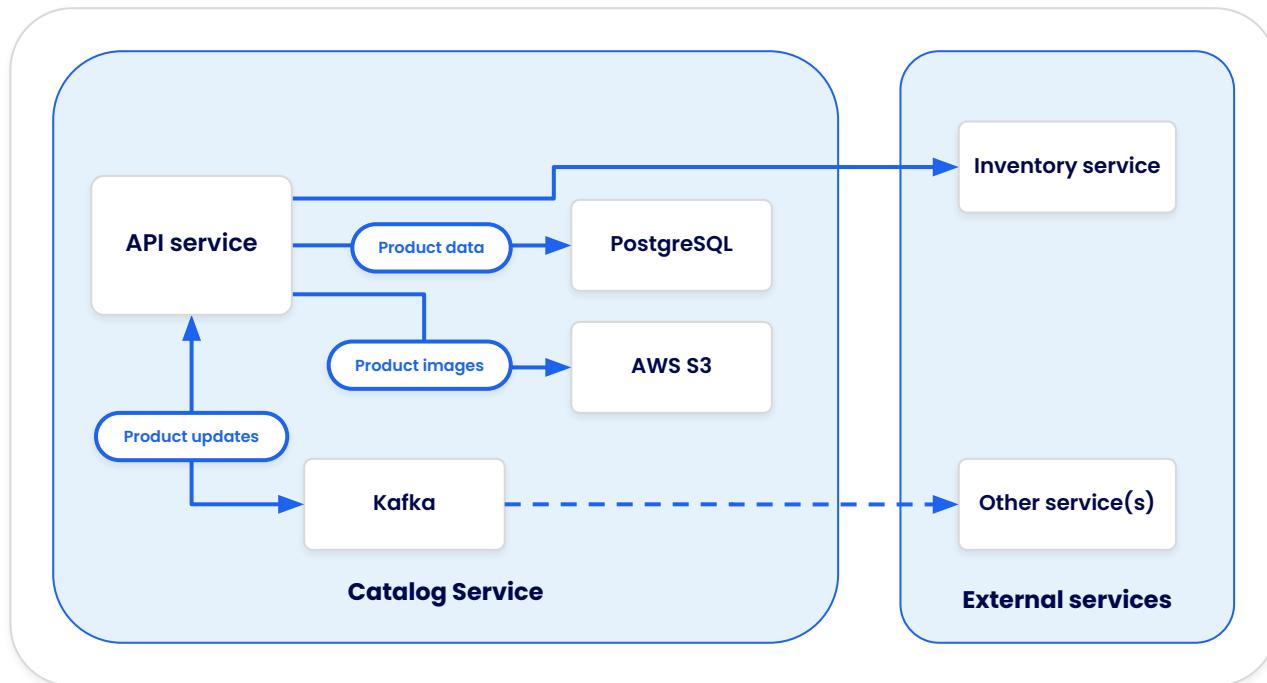
Docker across the SDLC

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**



Welcome!
Glad to have
you on the
team!

We already
assigned you an
easy JIRA Ticket.
- see if you can
knock it out by
lunch!



The ticket

The screenshot shows a Jira ticket page for issue DEVR-981. The ticket title is "Add UPC code to product_created event". The description states: "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." The ticket status is "In Progress".

Activity: Show: Comments ▾ Add a comment... Pro tip: press M to comment

Details:

- To be done by: None
- Assignee: Moby Dock
- Parent: None
- Sprint: S24: Sprint Awesome!
- Development: [Create branch](#) [Create commit](#)
- 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



Welcome!
Glad to have
you on the
team!

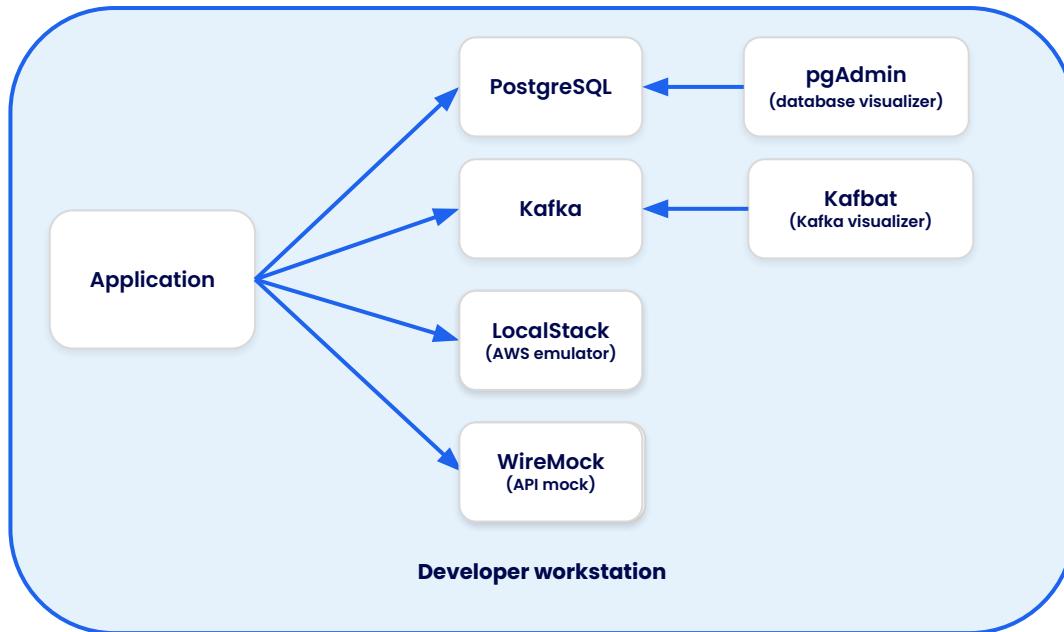
Clone the repo, use
README to install
env. Spend the first
week updating the
docs and setting up
your machine. Then
work on the task





SDLC Stage - Development

The development environment



The development tools and features



Docker Engine

Container networking

Settings management



Docker CLI/GUI

Port forwarding

Registry Access Management

Docker Compose

Docker Debug

Image Access Management

Volume mounts

Synchronized file shares

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.

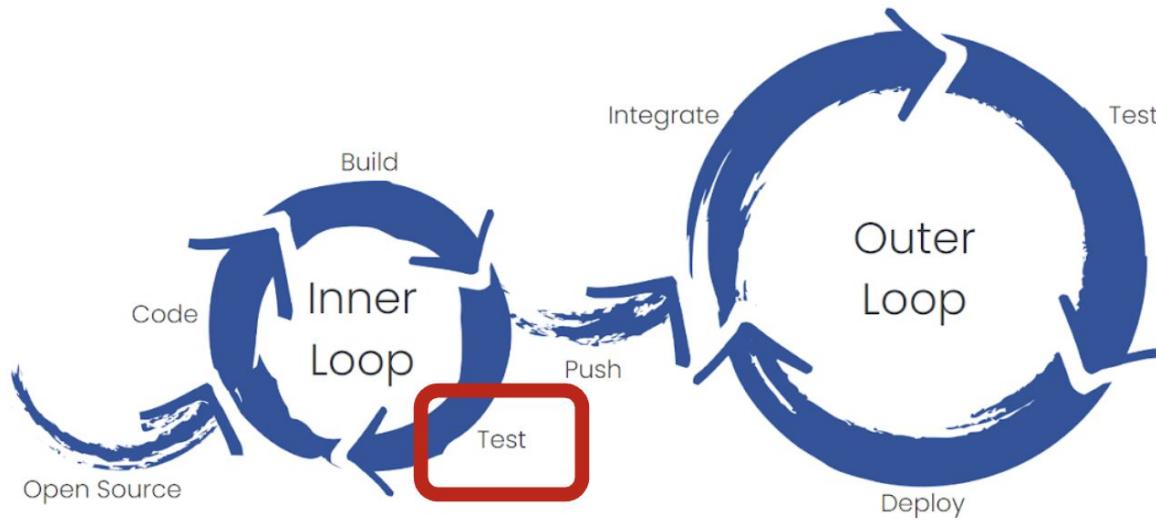




SDLC Stage - Testing

Shifting left with Testcontainers

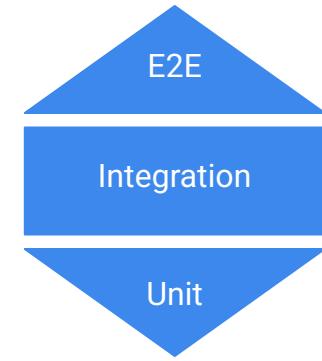
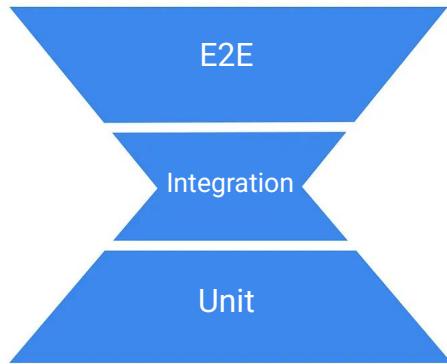
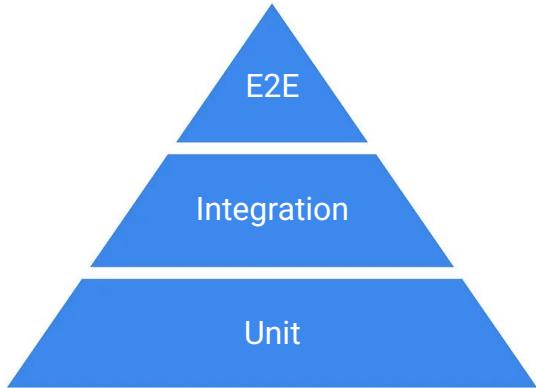
95% of developers write tests within their inner development loop



[State of Local Development and Testing](#)

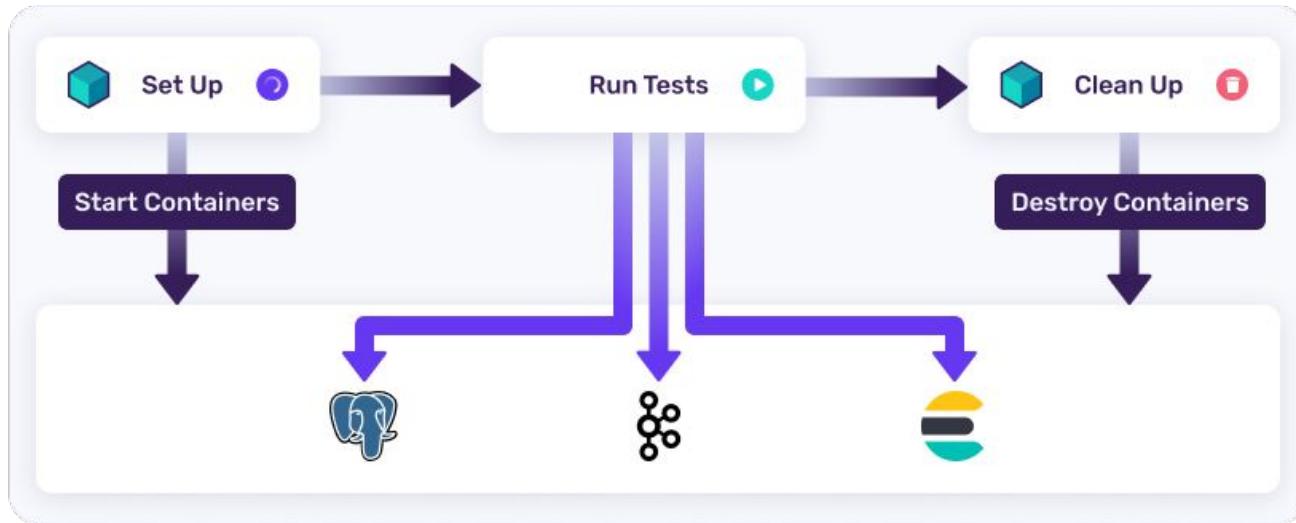


The shapes of testing



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

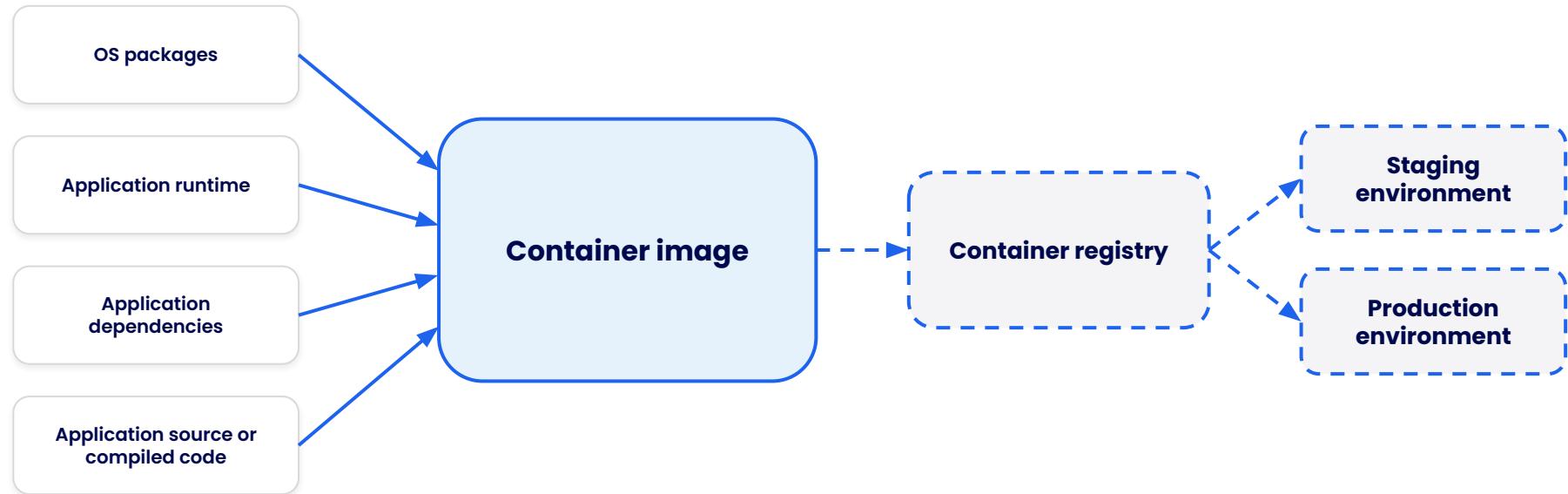
Reduced testing infra costs.
Reduced execution times.





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

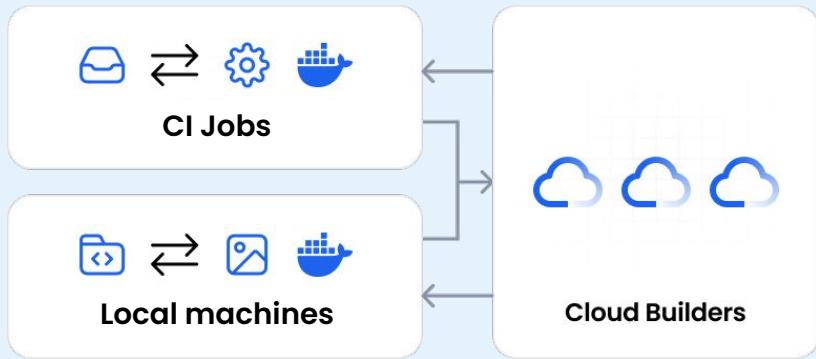
Build secrets

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

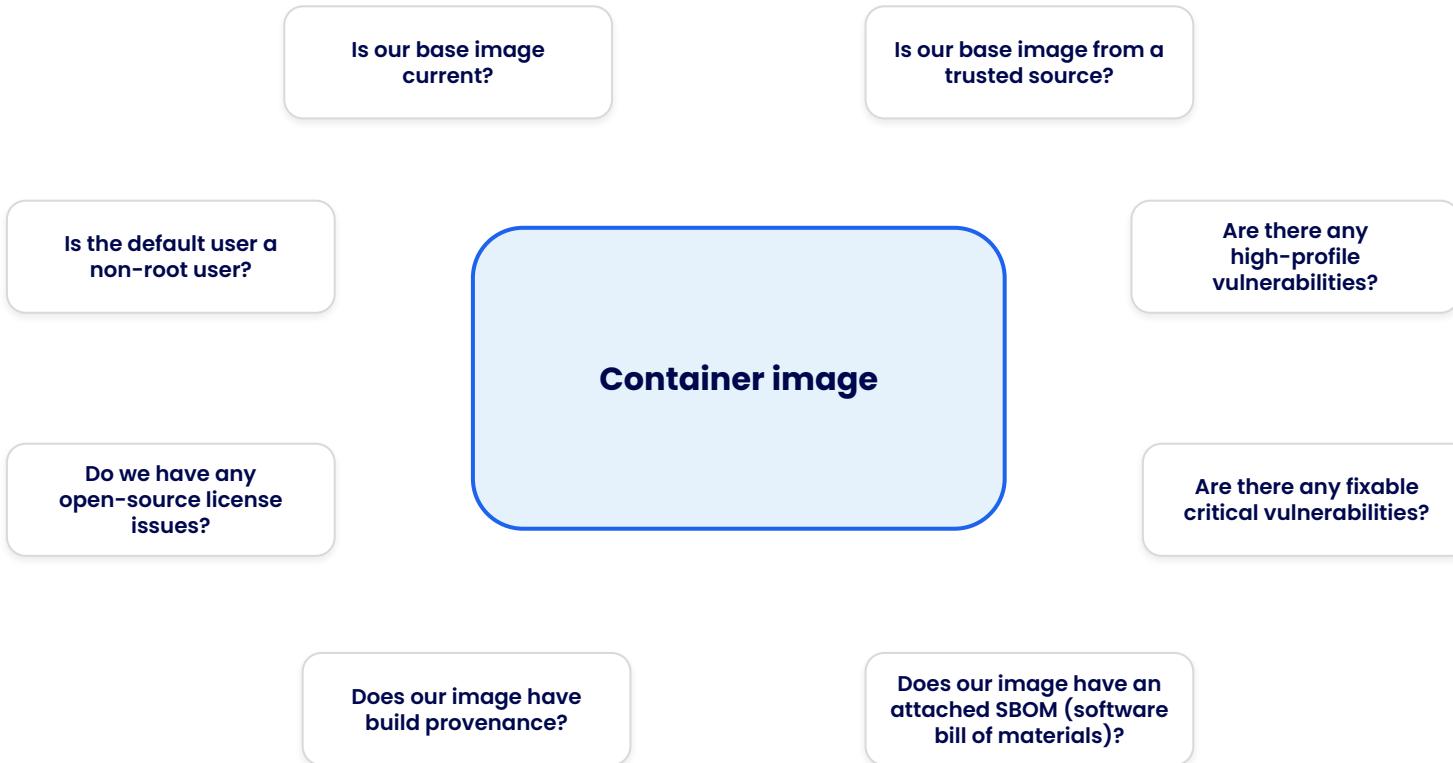
Reduced time to recover.
Increased best practice adoption.





SDLC Stage – Secure

Example policies



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



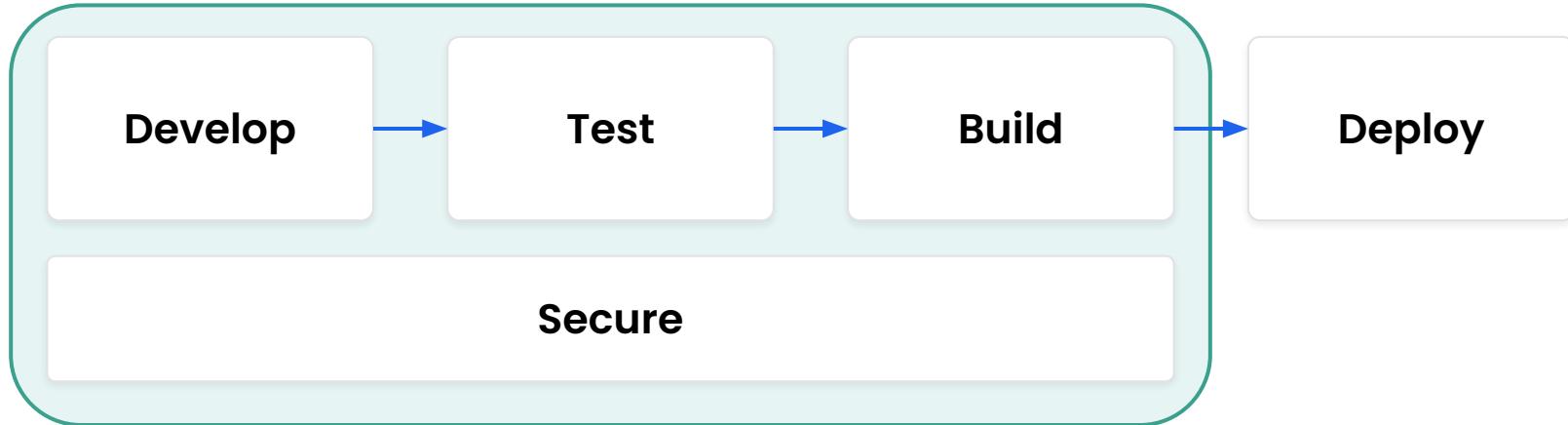
Recap

Docker's vision

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



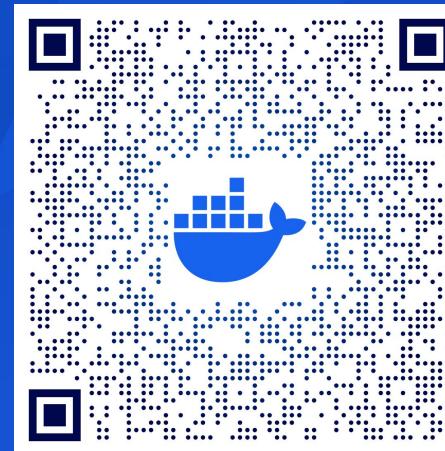
The SDLC



Thank you!



Sample project repo
github.com/dockersamples/catalog-service-node



This slide deck



Thank you
See you again soon.

