

Cloud Native Introduction

—
Ritu Maheshwari
Developer Advocate



Contents

- Cloud Native Introduction
- Key tenets of Cloud Native Application
- Cloud Native Development
- Introduction to Kubernetes
- Why Microservices and Cloud Native?
- Introduction to CI CD



App Modernization is
inevitable



What is Cloud Native?

Build great products faster

**Exploit scale, resiliency &
agility of cloud computing**

Key tenets of a cloud native application

1. Packaged as light weight **containers**
2. Developed with best-of-breed languages and frameworks
3. Designed as loosely coupled **microservices**
4. Centered around **APIs** for interaction and collaboration
5. Architected with a clean separation of stateless and stateful services
6. Isolated from server and operating system dependencies
7. Deployed on self-service, elastic, **cloud infrastructure**
8. Managed through agile **DevOps** processes
9. Automated capabilities
10. Defined, policy-driven resource allocation

Cloud-native development



Architectures

- Microservices
- Serverless

Practices

- Automated DevOps pipeline (CI/CD)
- Small teams

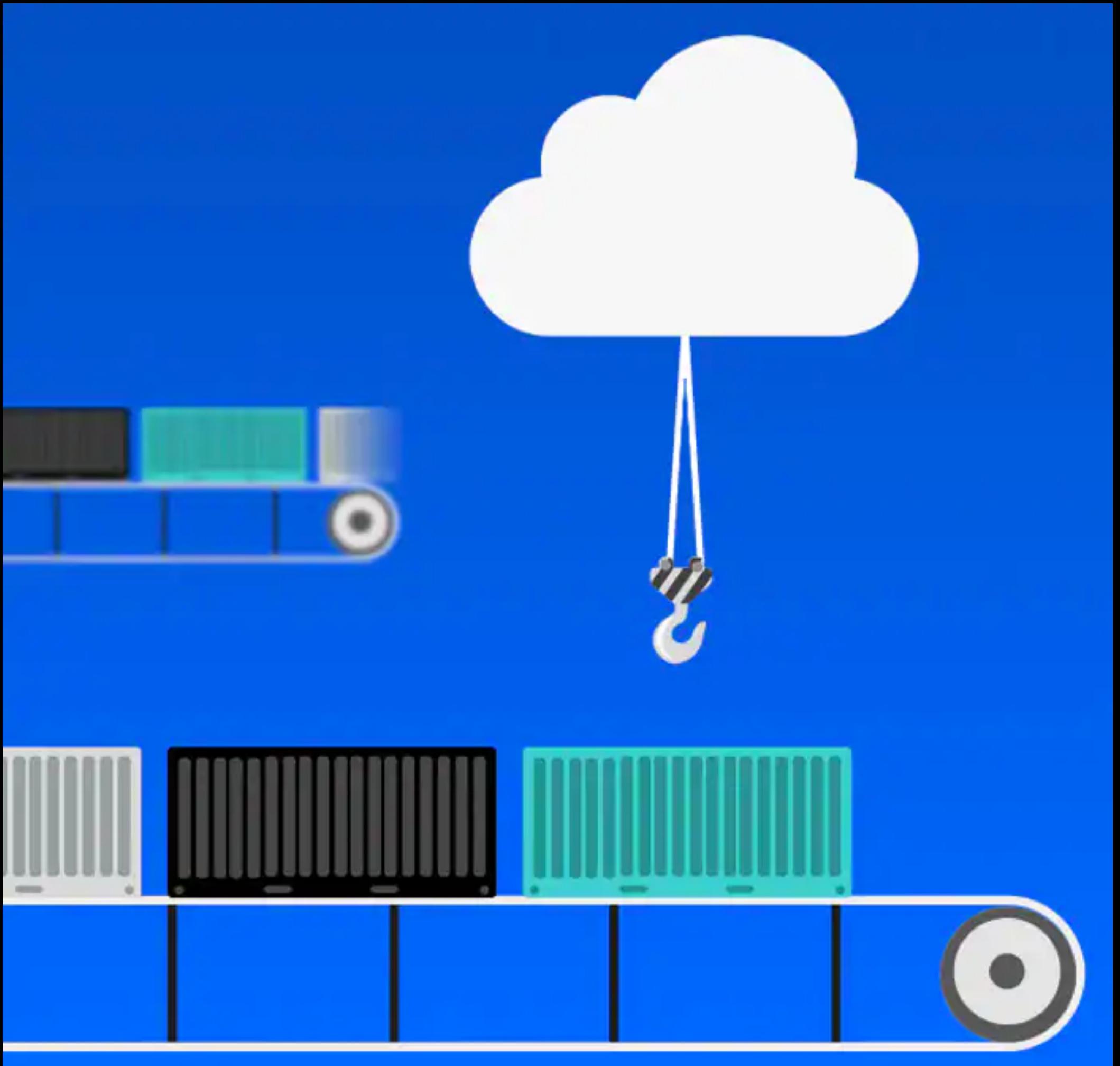
Technologies

- Kubernetes
- Data
- On-demand infrastructure

Containers change the economics of delivery

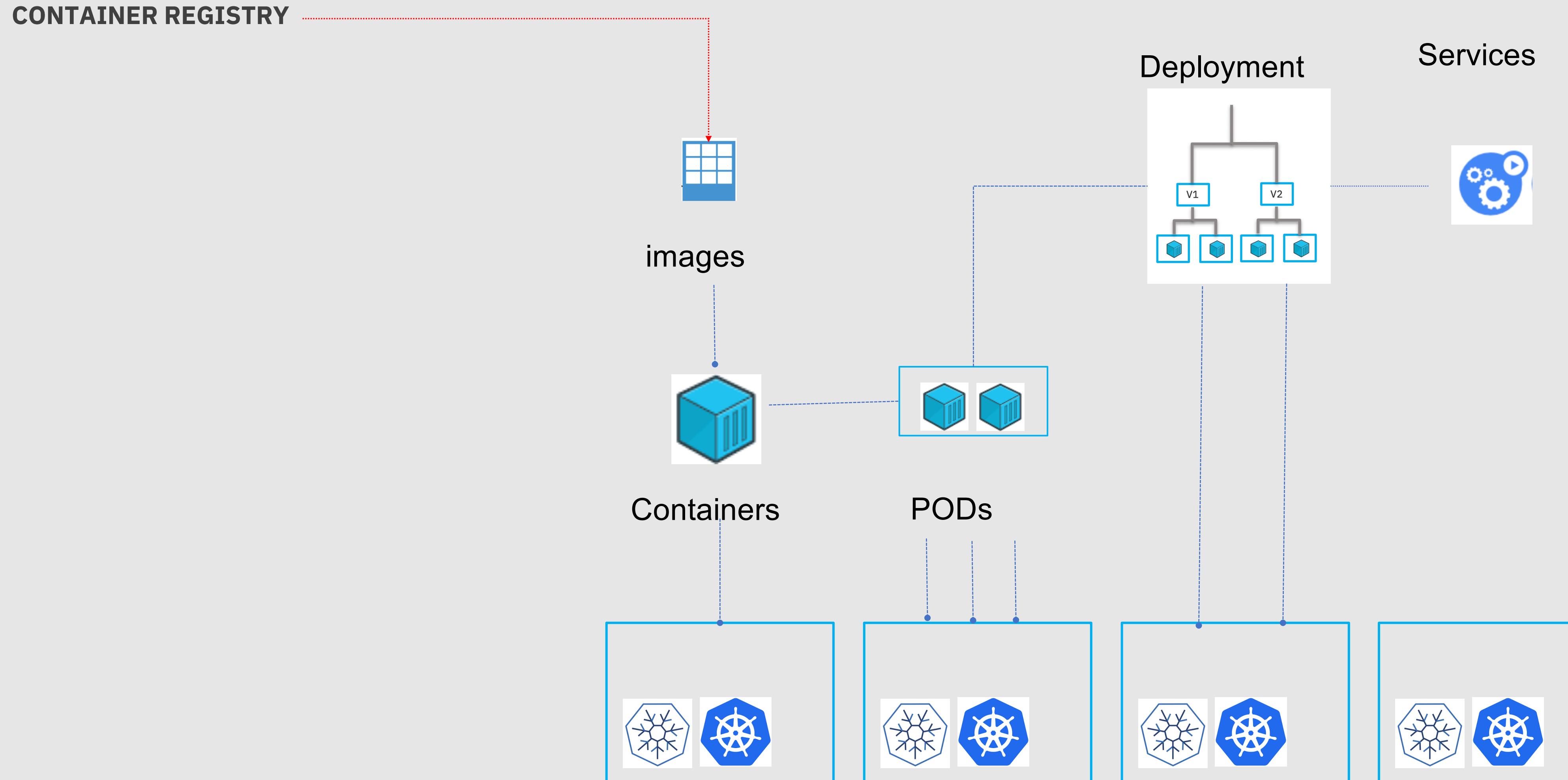
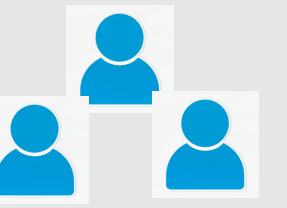
Organizations are adopting containers to improve developer productivity, efficiency in DevOps, and application portability

- Lightweight packaging that includes the software and all its dependencies
- Easily portable across on-premises and public cloud environments
- More efficient use of infrastructure than traditional VM deployments



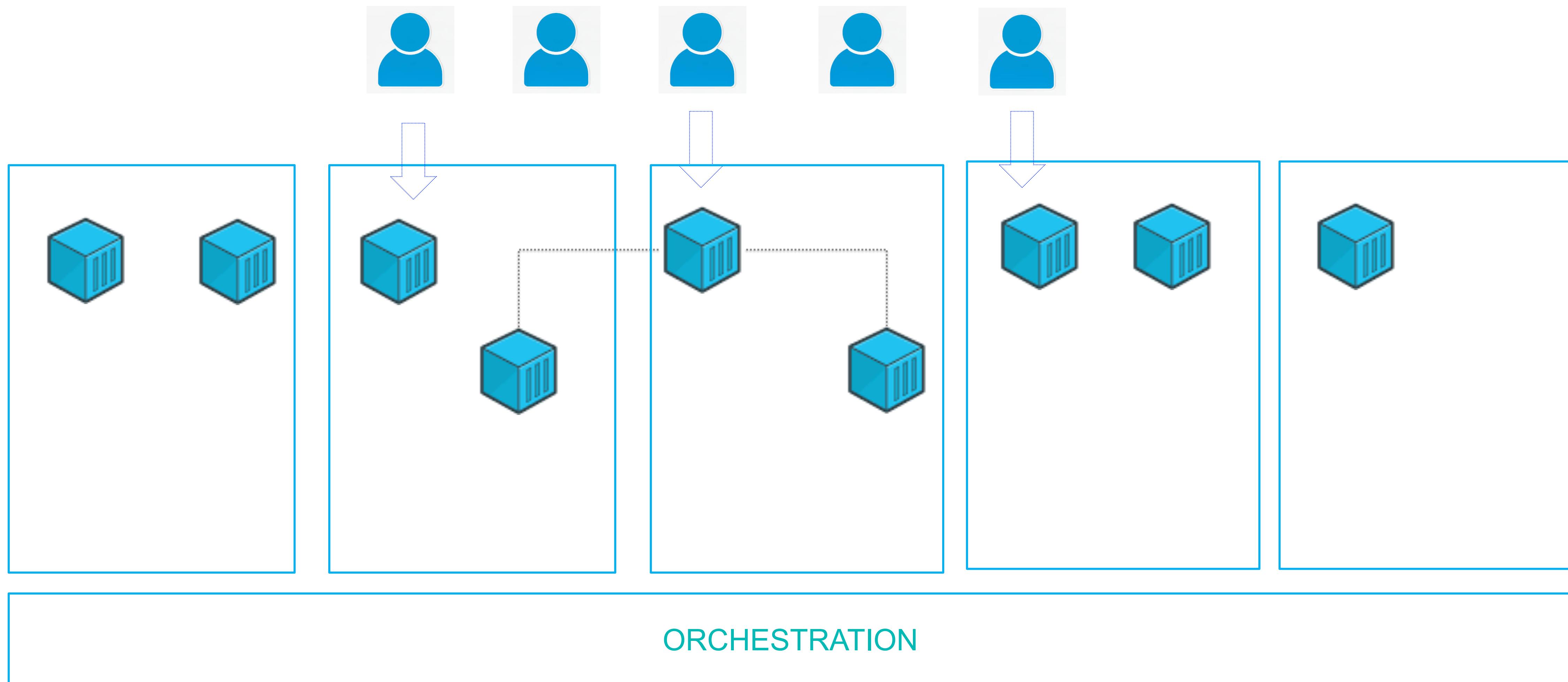
Kubernetes

Kubernetes Components



Kubernetes

Container Orchestration

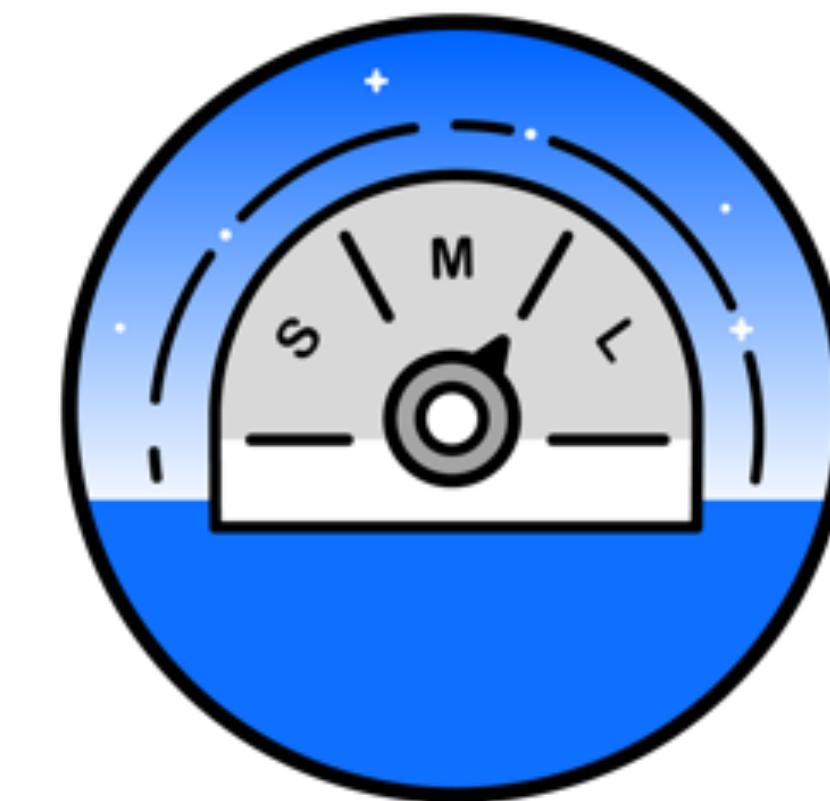




INTELLIGENT SCHEDULING



SELF-HEALING



HORIZONTAL SCALING



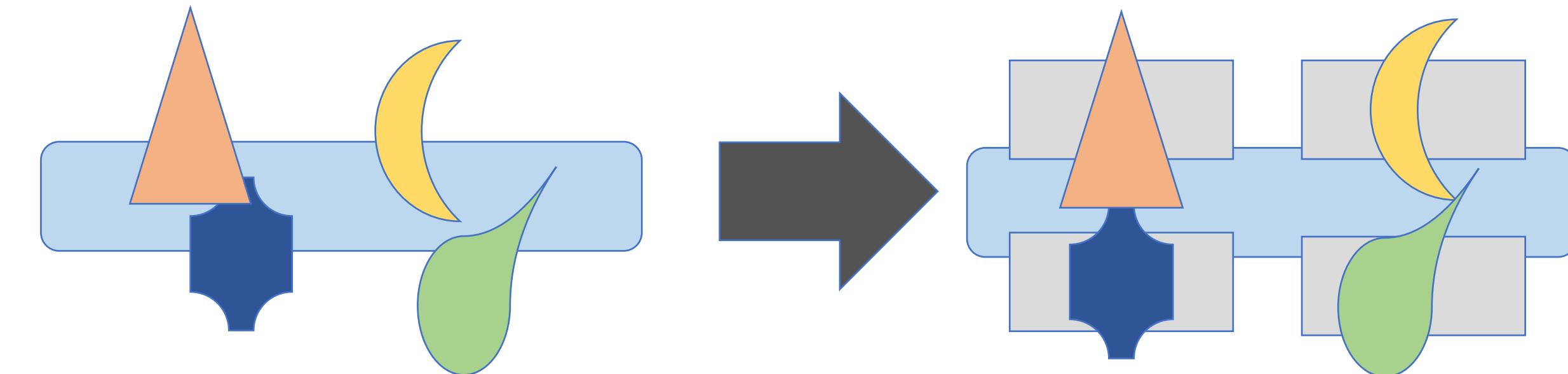
SERVICE DISCOVERY & LOAD BALANCING



AUTOMATED ROLLOUTS AND ROLLBACKS

Key tenets of a microservices architecture

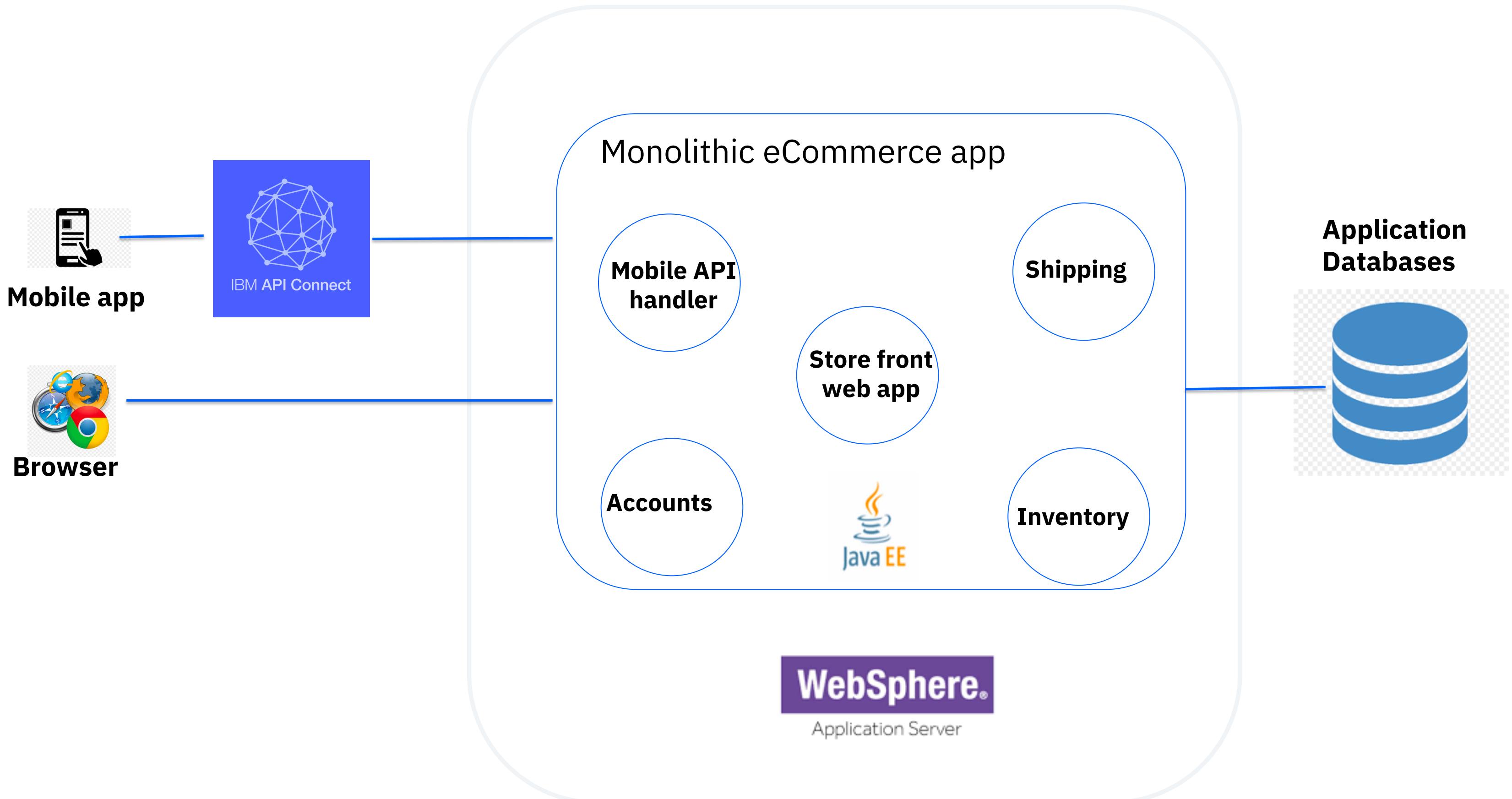
1. Large monoliths are broken down into many small services
2. Services are optimized for a single function or business capability
3. Teams that write the code should also deploy the code
4. Design for failure



Example monolithic application

eCommerce app

- Store front web interface
- Customer Accounts
- Inventory
- Shipping
- Back end for mobile app

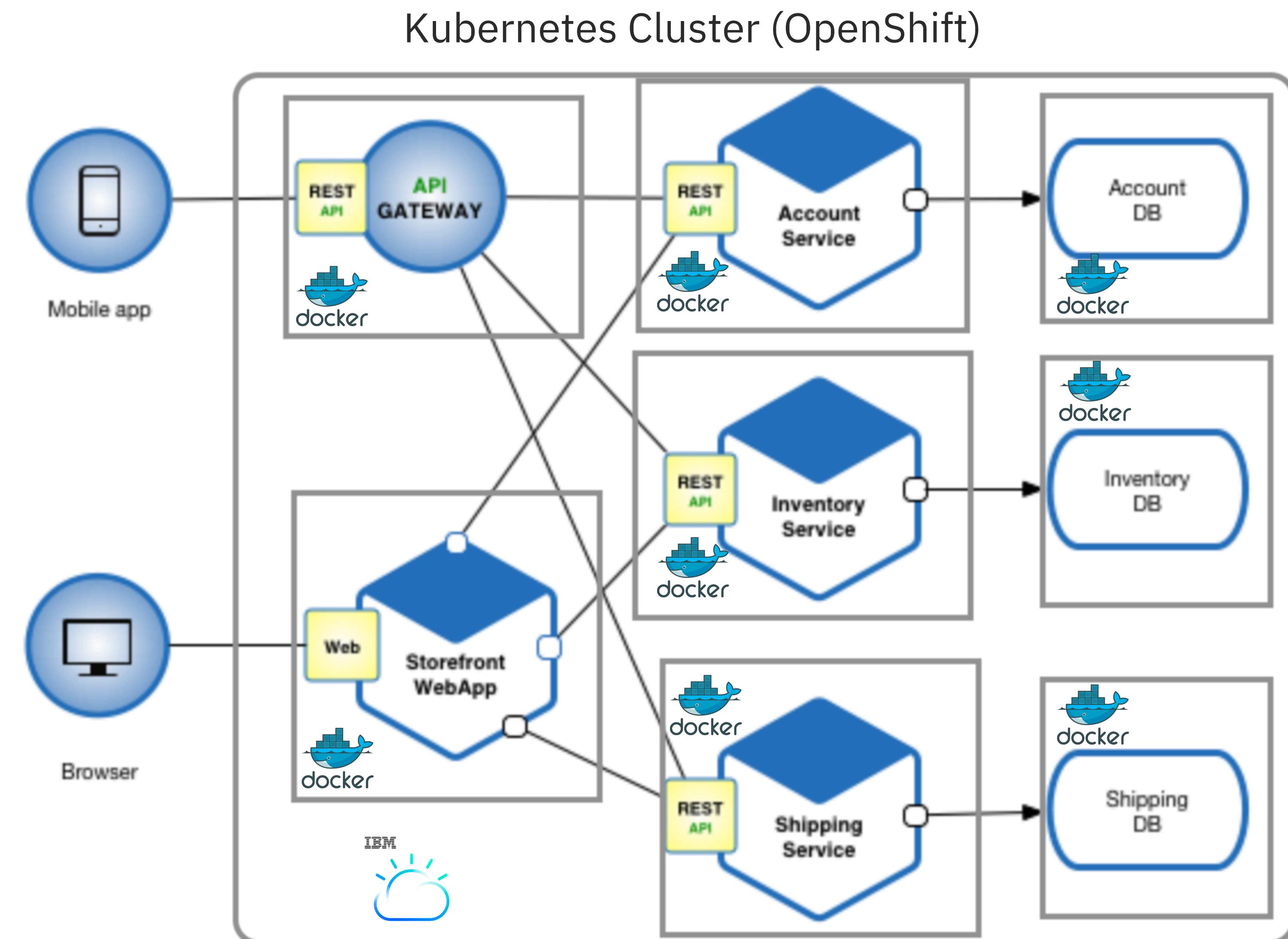


An eCommerce Java EE app on Websphere

Transformed application

Key technologies

- Containers (Docker)
- Container orchestration (Kubernetes)
- Transformation Advisor
- 12-Factor Best Practices
- CI/CD tools (e.g Jenkins)



An eCommerce microservices app on a Kubernetes cluster

Why Microservices and Cloud Native?

Efficient teams	Simplified deployment	Right tools for the job	Improved application quality	Scalability
<ul style="list-style-type: none">• End to end team ownership of relatively small codebases➤ Teams can innovate faster and fix bugs more quickly	<ul style="list-style-type: none">• Each service is individually changed, tested, and deployed without affecting other services➤ Time to market is accelerated.	<ul style="list-style-type: none">• Teams can use best of breed technologies, libraries, languages for the job at hand➤ Leads to faster innovation	<ul style="list-style-type: none">• Services can be tested more thoroughly in isolation➤ Leads to better code coverage	<ul style="list-style-type: none">• Services can be scaled independently at different rates as needed➤ Leads to better overall performance at lower cost

DevOps is a culture

- Developers + Operations + Quality Assurance = DevOps
- Rapid development
- Quality releases

Rapid development

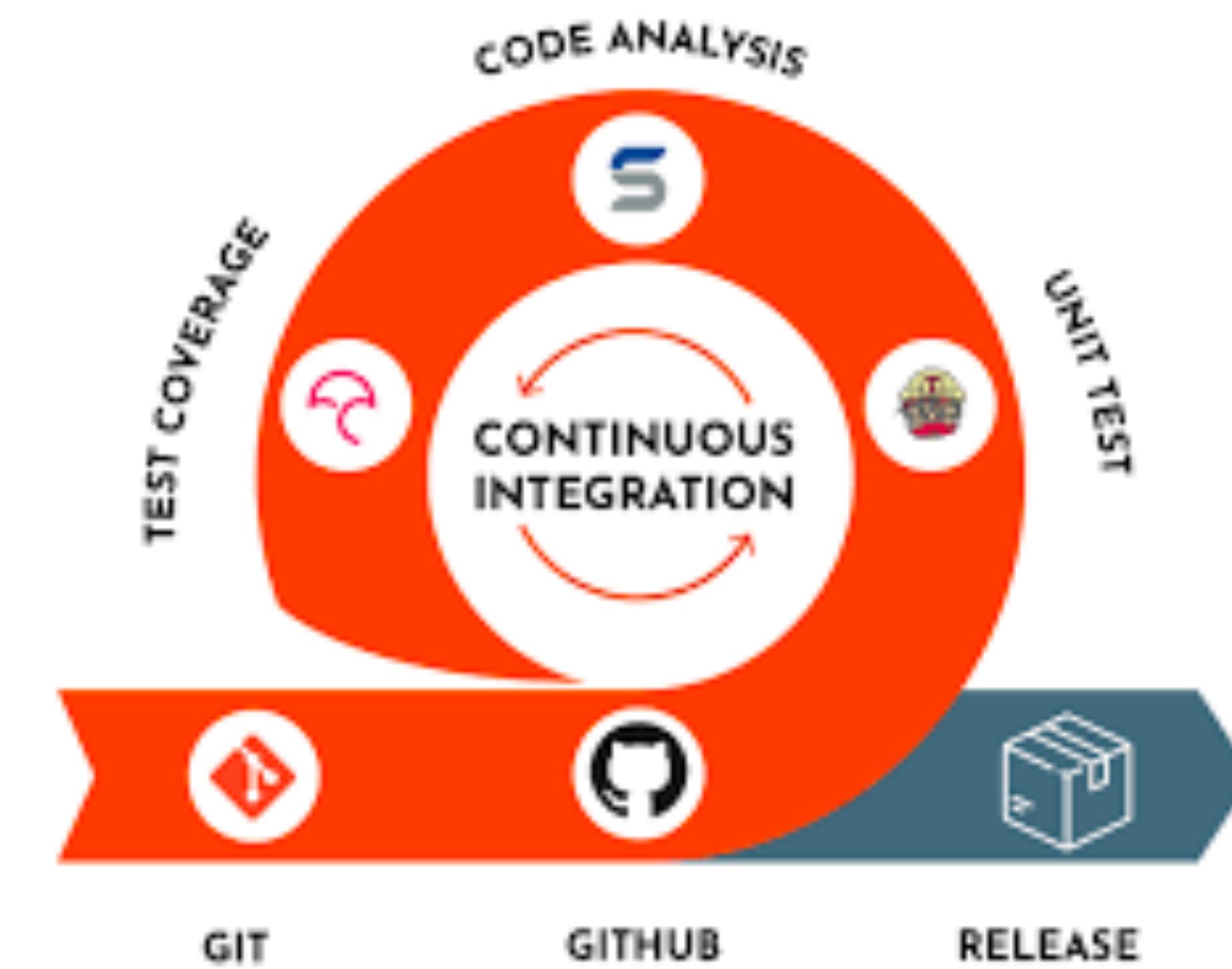
- Short development cycles
- Large jobs broken into small tasks
- Each task aligned with goals

Quality releases

- Testing at every step
- Monitoring
- Disaster testing

Continuous Integration

- Source control is your friend
- Master is the source of truth



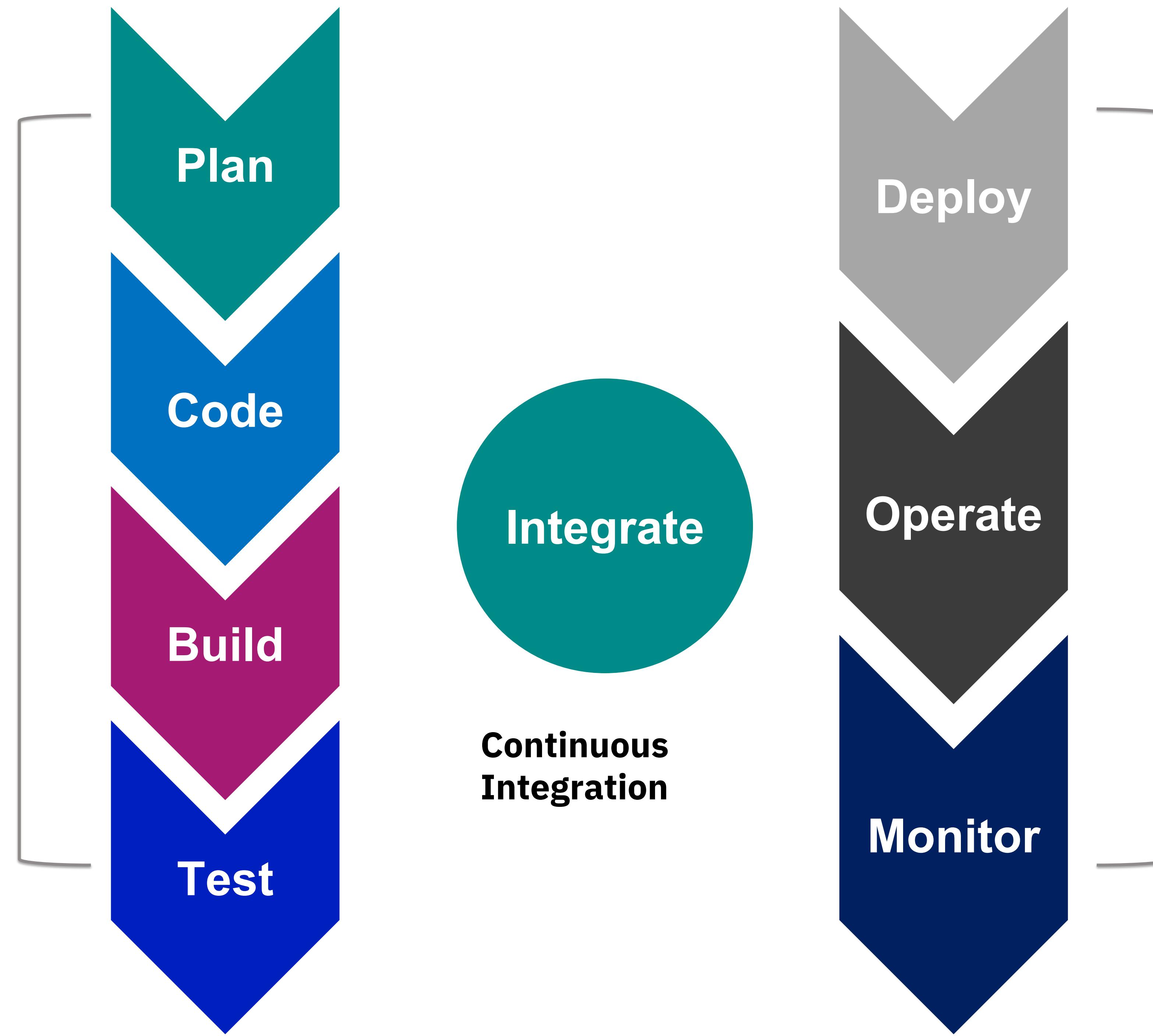
Continuous Delivery

- Delivery vs deployment
- Automation is key

CI/CD is a method

- Continuous testing + continuous integration + continuous delivery
= CI/CD
- Test everywhere
- Merge into master branch
- Product is always available

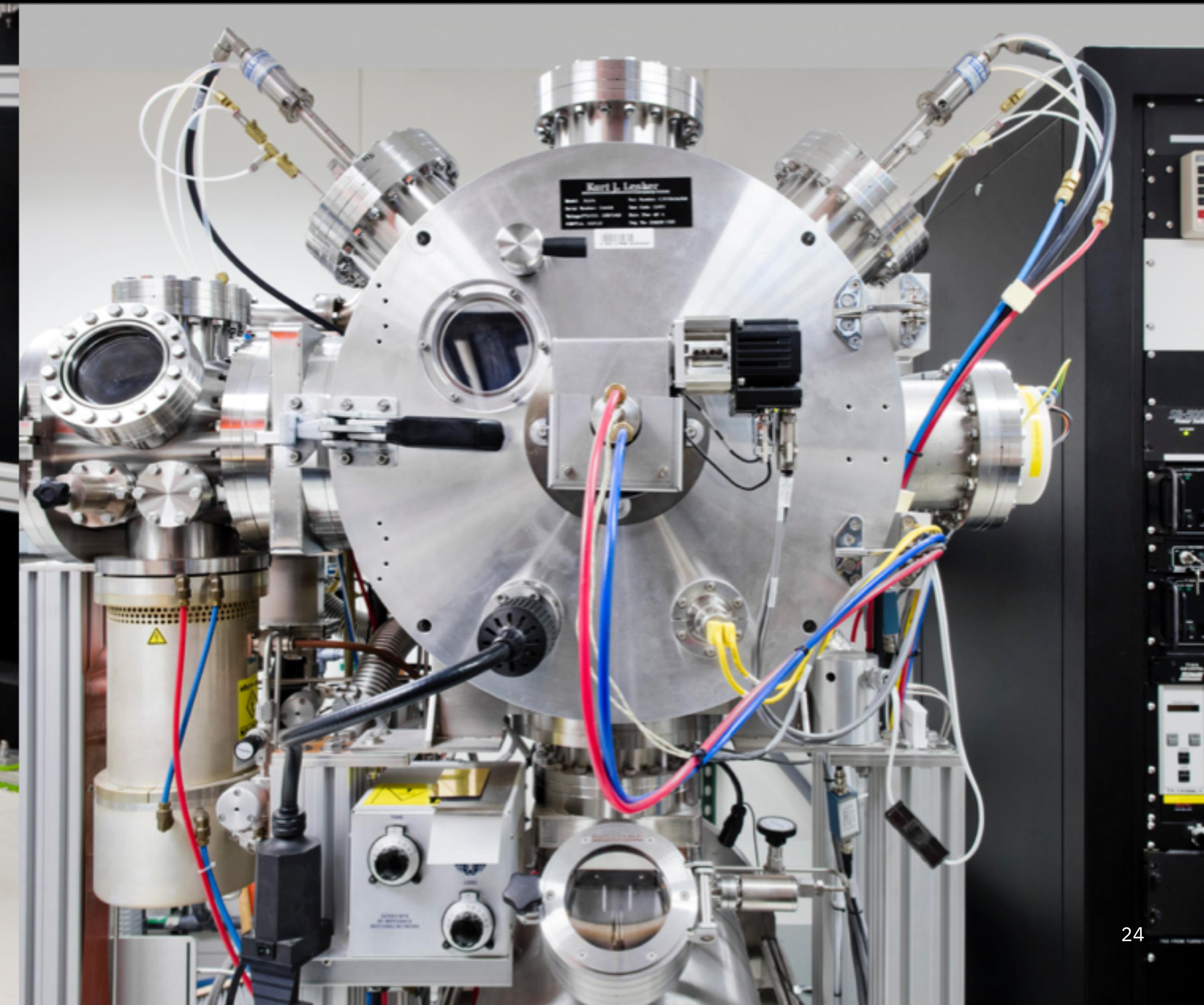
Continuous Delivery
- Development
- Test



Continuous Deployment

Demo Time!

<https://github.com/IBMDevConnect/CloudNative/tree/master/Lab%20-%20Kubernetes%20Deployment>



Thank you

Ritu Maheshwari

—

Email id: rimahe12@in.ibm.com

© Copyright IBM Corporation 2020. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and ibm.com are trademarks of IBM Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at [Copyright and trademark information](#).

