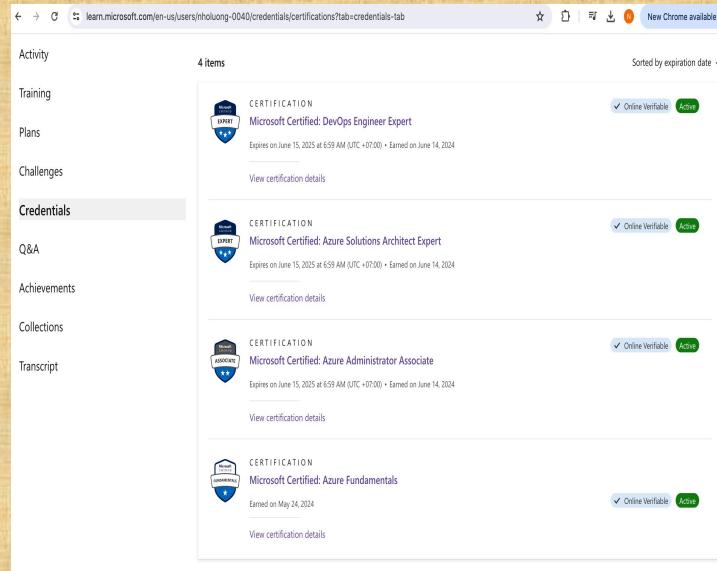
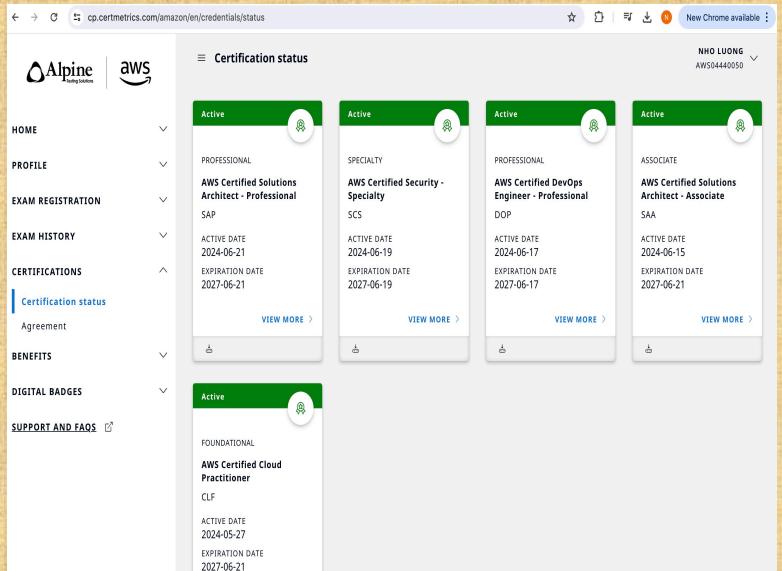


# DevOps for Beginners

Author: Nho Luong  
Skill: DevOps Engineer Lead



The image displays two screenshots of professional certification dashboards. The left screenshot is from cp.certmetrics.com/Amazon, showing a list of active certifications for Nho Luong. It includes four AWS Certifications: AWS Certified Solutions Architect - Professional (SAP), AWS Certified Security Specialty (SCS), AWS Certified DevOps Engineer - Professional (DOP), and AWS Certified Solutions Architect - Associate (SAA). The right screenshot is from learn.microsoft.com, showing a list of Microsoft Certifications. It includes three Microsoft Certifications: Microsoft Certified: DevOps Engineer Expert, Microsoft Certified: Azure Solutions Architect Expert, and Microsoft Certified: Azure Administrator Associate. Both dashboards provide details like issue date, expiration date, and a 'View certification details' link.



DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity

Amazon Web Services(AWS)

DevOps is a collaborative and multidisciplinary effort within an organization to automate continuous delivery of new software versions, while guaranteeing their correctness and reliability

## A Survey of DevOps Concepts and Challenges - L Leite

DevOps is the outcome of applying the most trusted principles from the domain of physical manufacturing and leadership to the IT value stream.

The DevOps Handbook.

DevOps relies on bodies of knowledge from Lean, Theory of Constraints, the Toyota Production System, resilience engineering, learning organizations, safety culture, human factors, and many others.

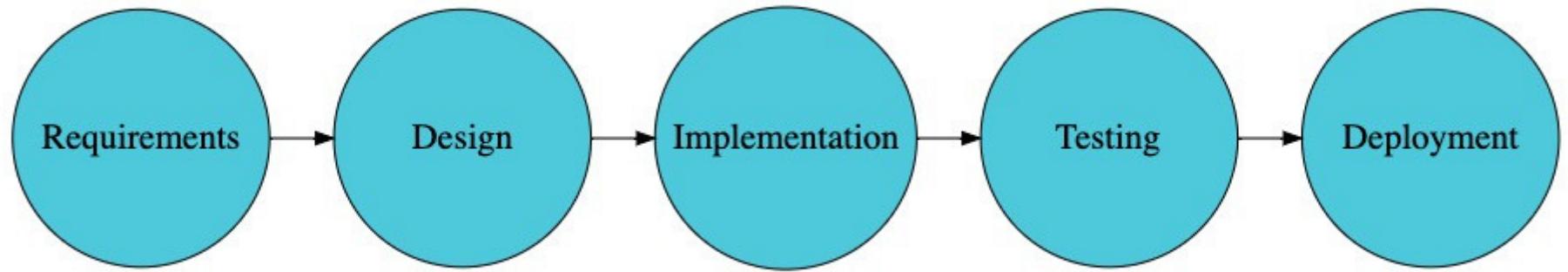
## The DevOps Handbook.

The result is world-class reliability, stability, and security at ever lower cost and effort; and accelerated flow and reliability through the technology value stream, including Product Management, Development, QA, IT Operations, and Infosec.

## The DevOps Handbook.

# EVOLUTION TO DEVOPS

# WATERFALL



# Waterfall

# 3 KEYS TO GREAT SOFTWARE

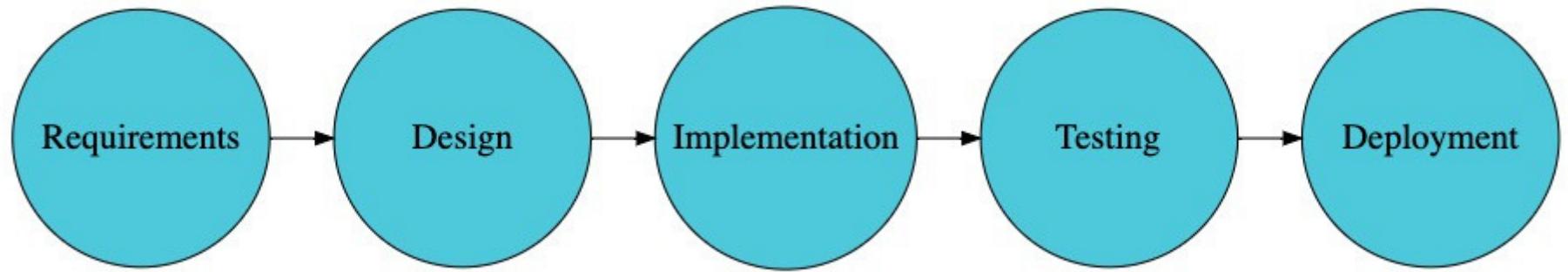
# ENHANCED COMMUNICATION

Business

Development

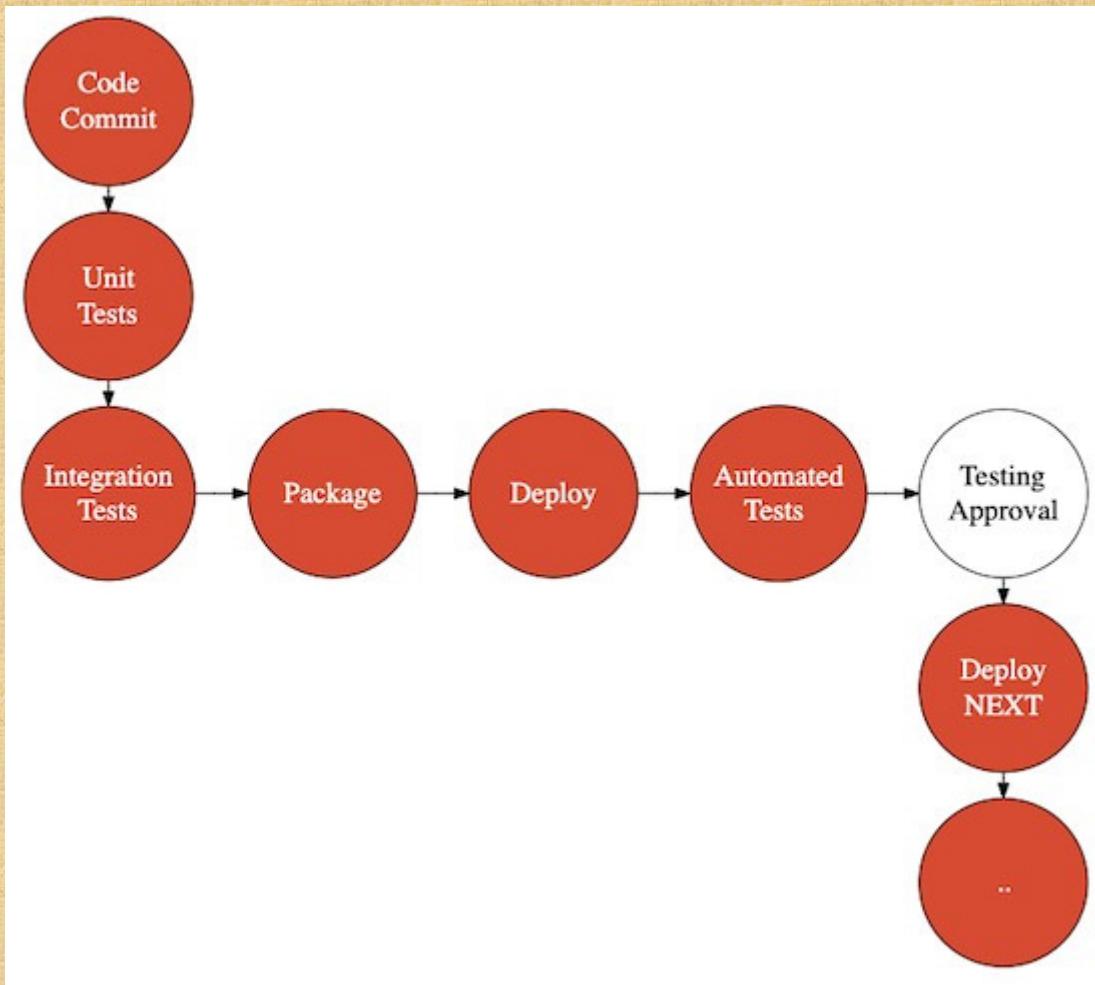
Operations

Teams

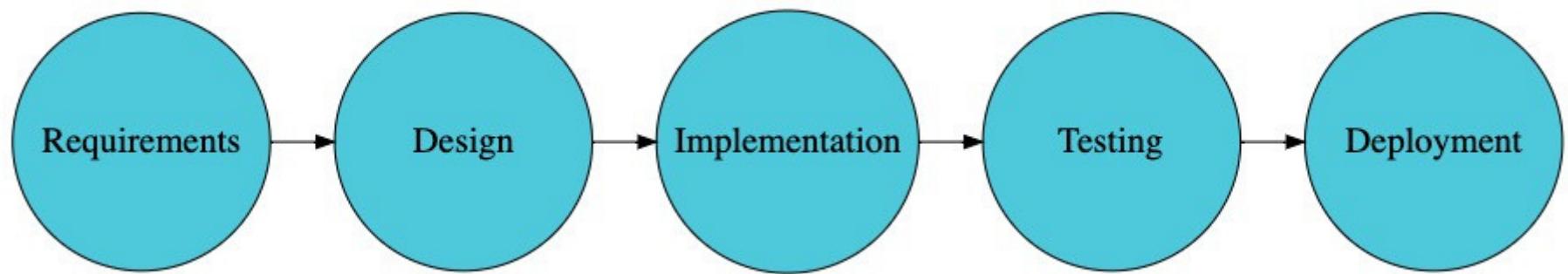


# Waterfall

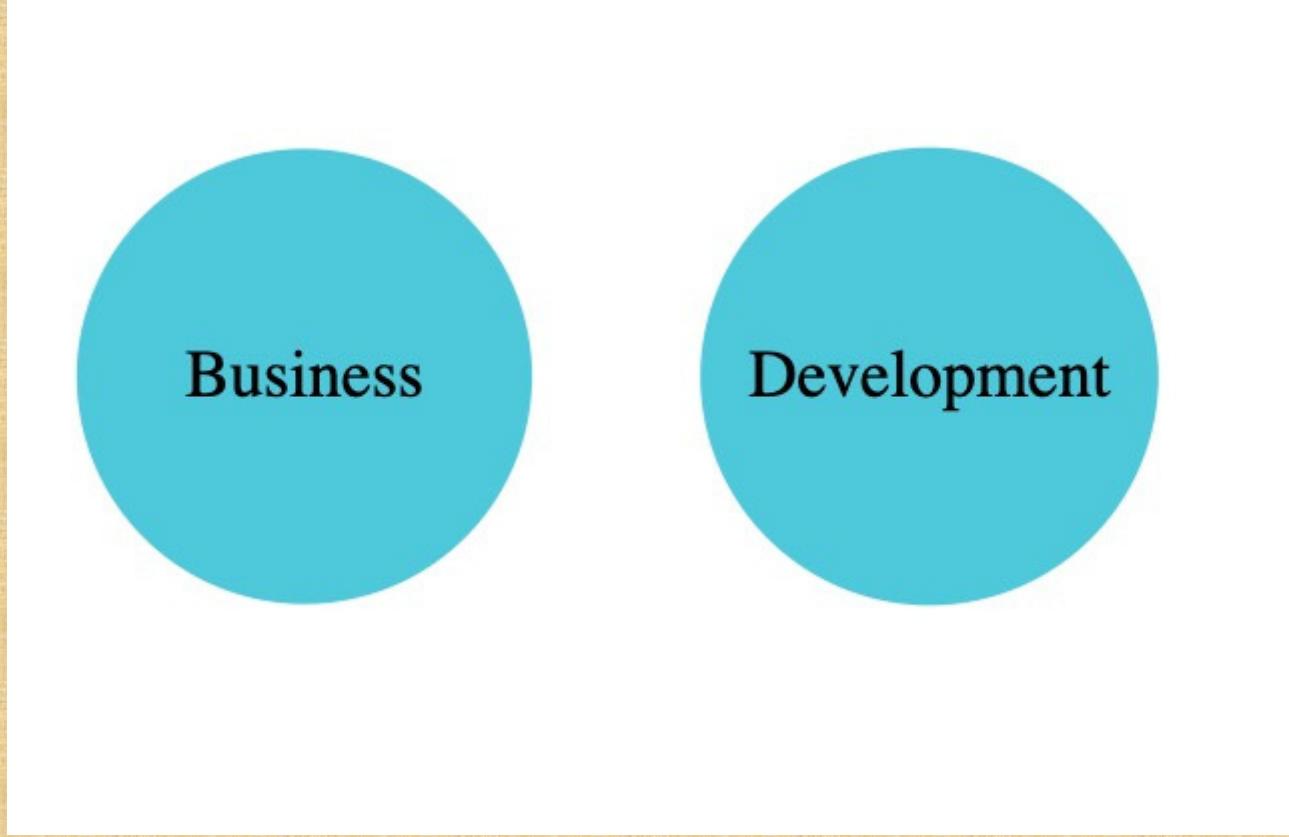
# AUTOMATION



# QUICK FEEDBACK



# AGILE

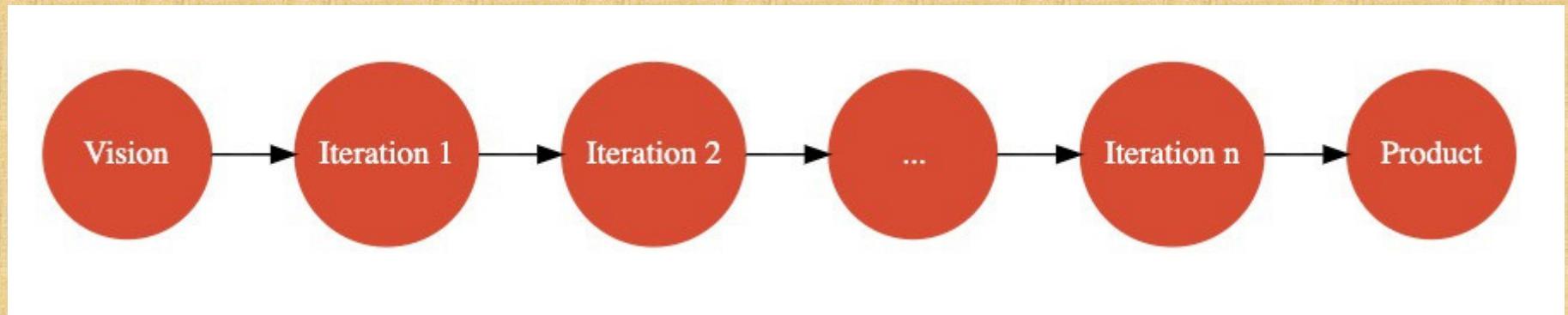


Business

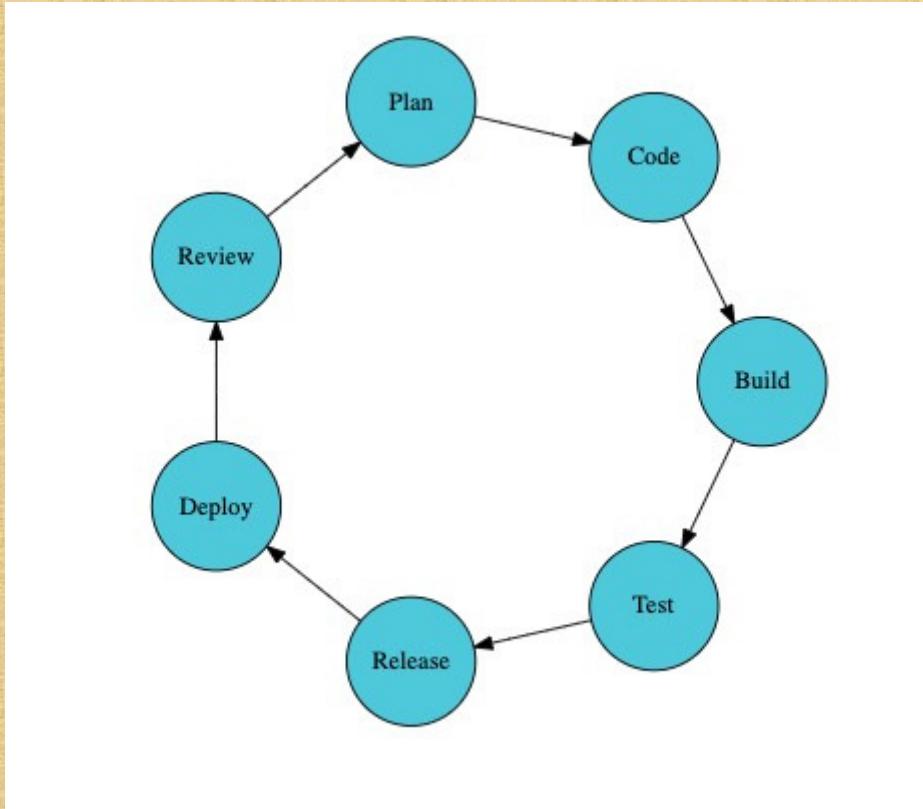
Development



Agile - One Team

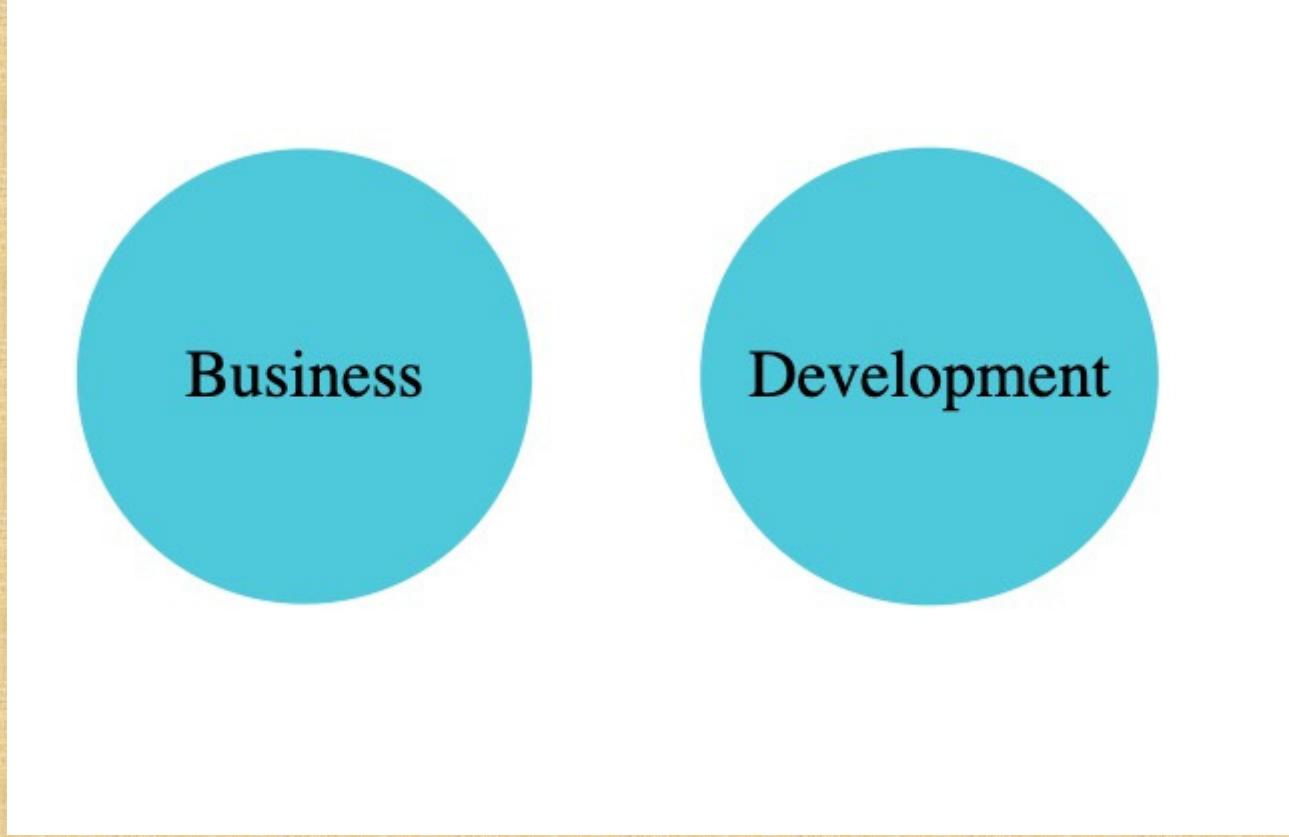


## Agile - Short Iterations



## Agile - Each Iteration

# Enhanced Communication



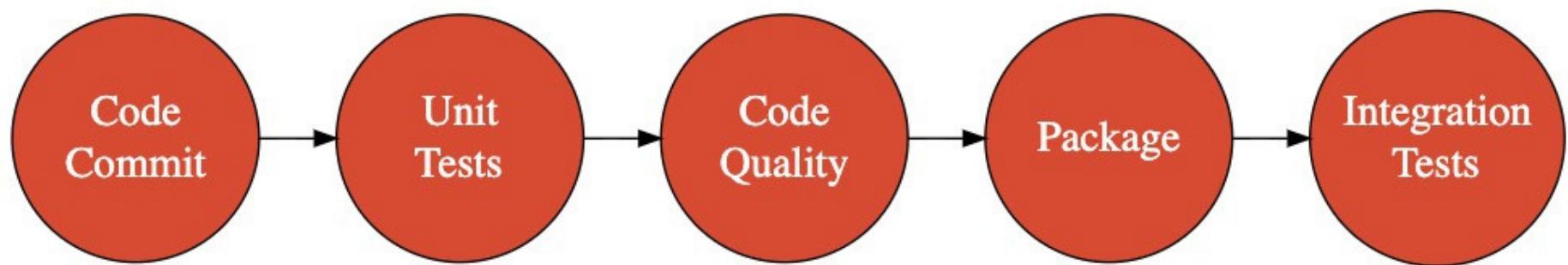
Business

Development



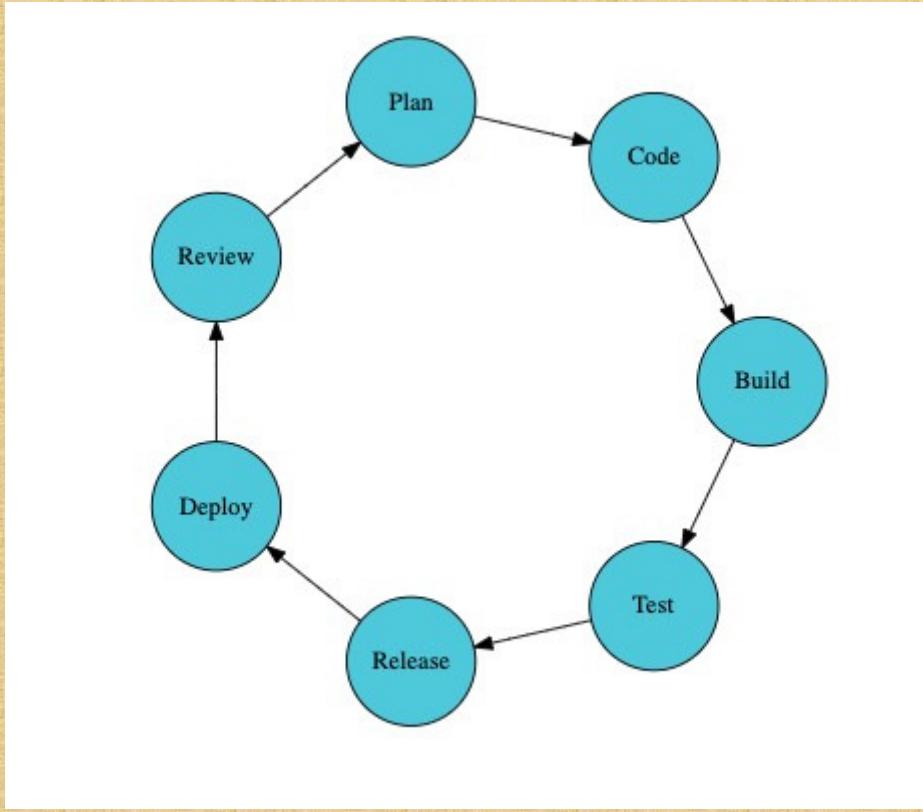
Agile - One Team

# Automation

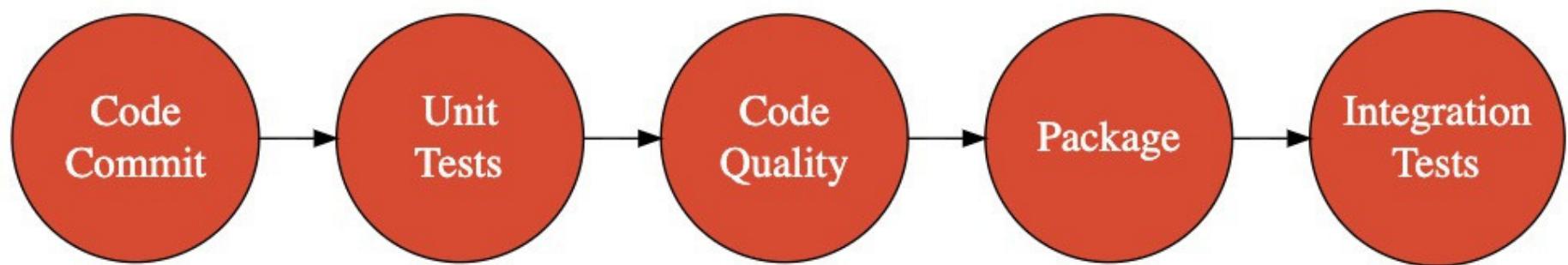


## Agile Automation

# Quick Feedback

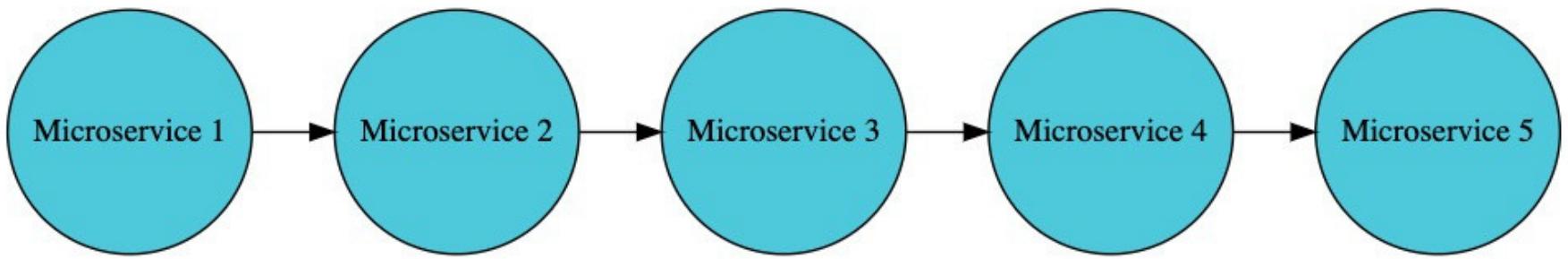


## Agile - Retrospectives



# Continuous Integration

# NEW CHALLENGES

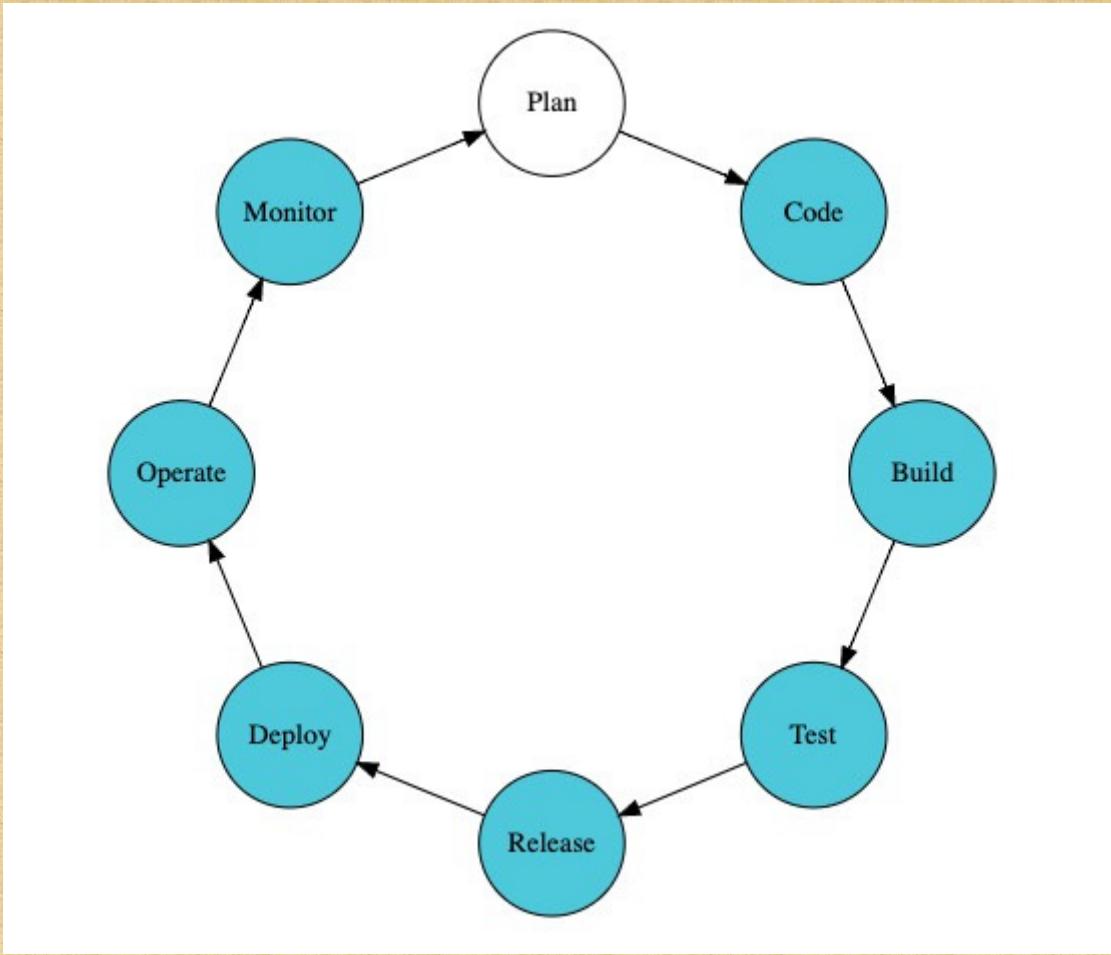


# Microservices

# DEVOPS

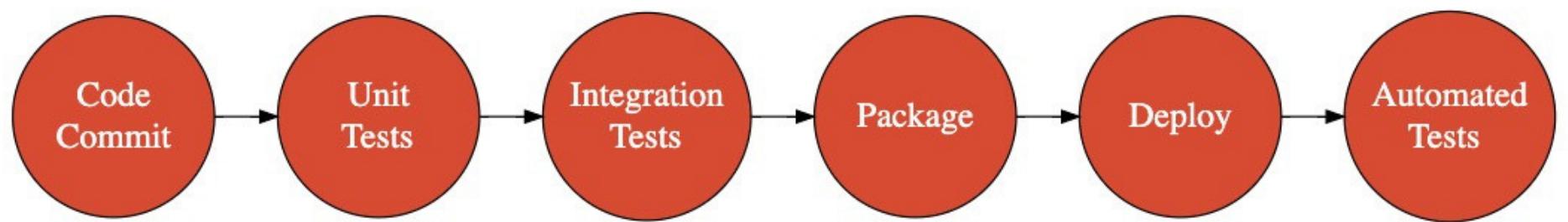


DevOps - Bring Teams Together

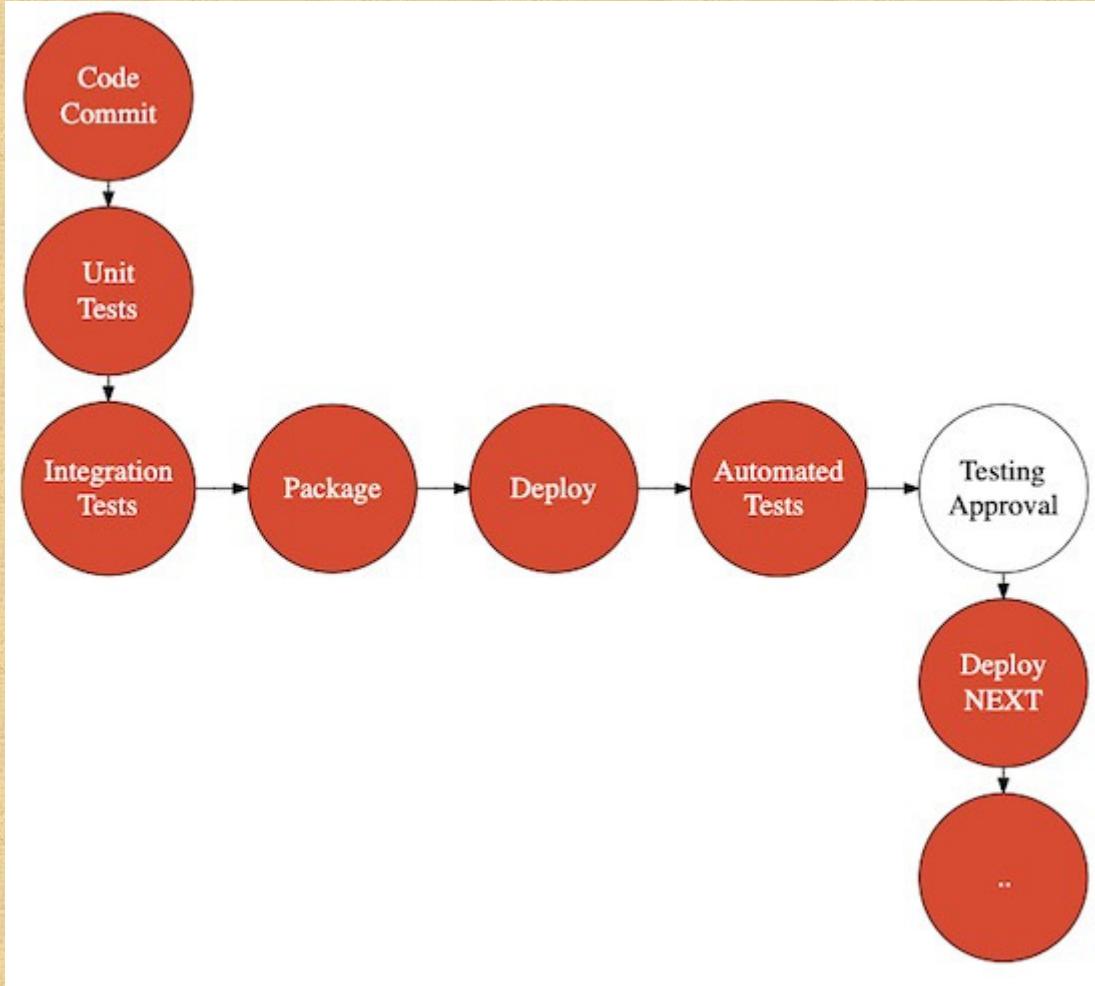


# Each Iteration

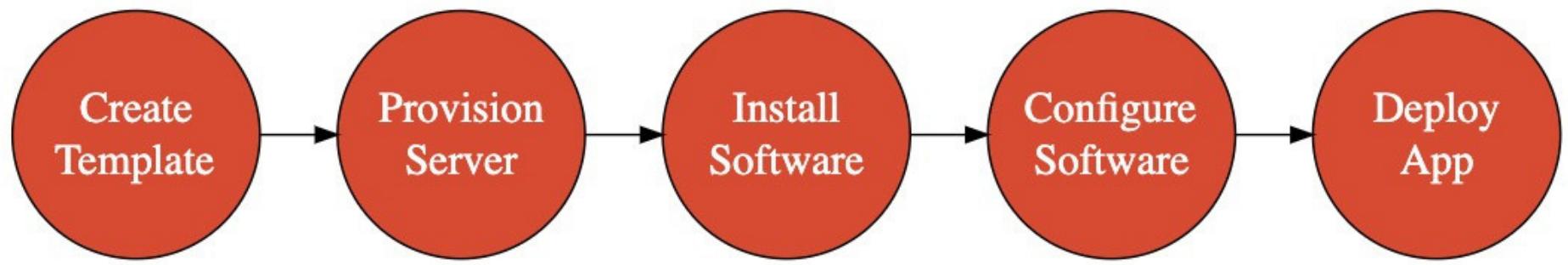
# AUTOMATION



# Continuous Deployment



# Continuous Delivery



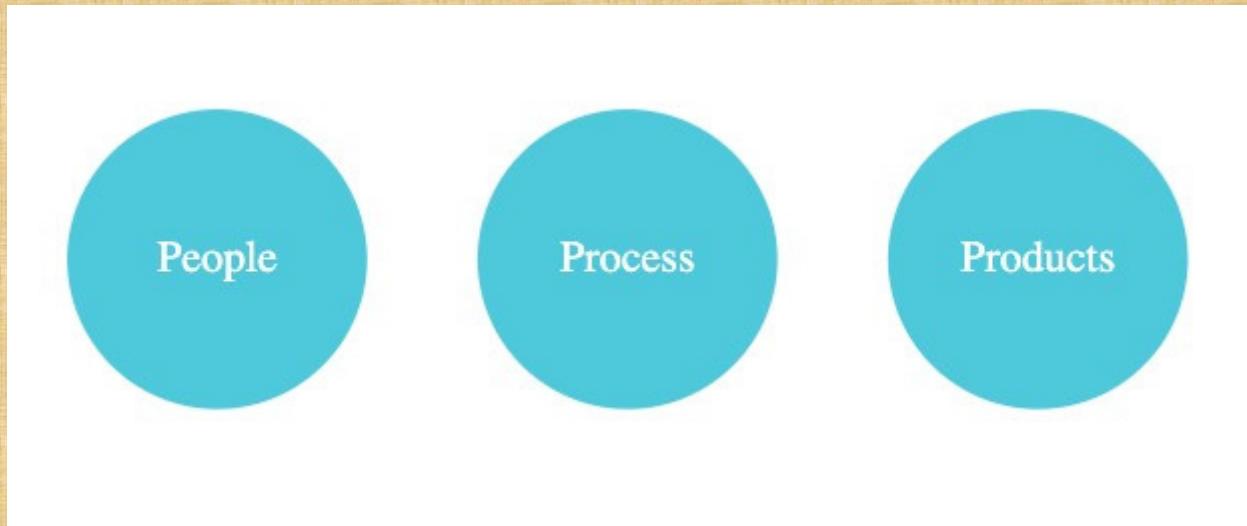
## IAC - Automate Operations

# QUICK FEEDBACK

# AGILE AND DEVOPS

# DEVOPS

Bring Down the Wall



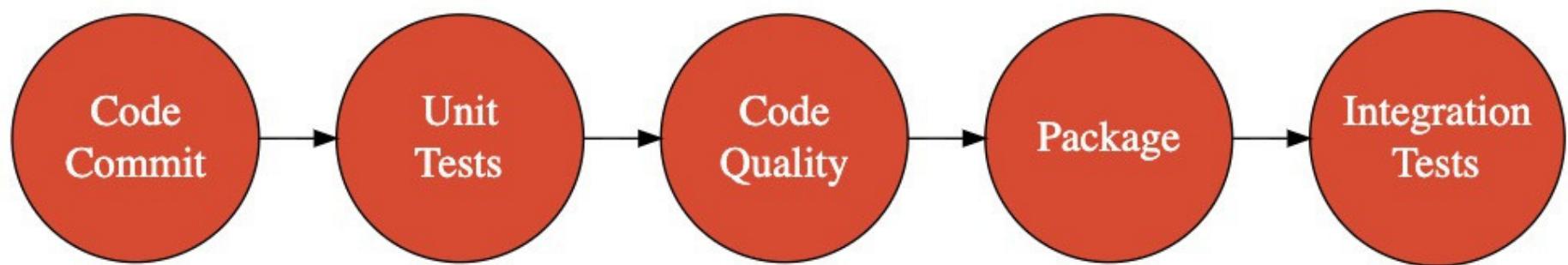


Business

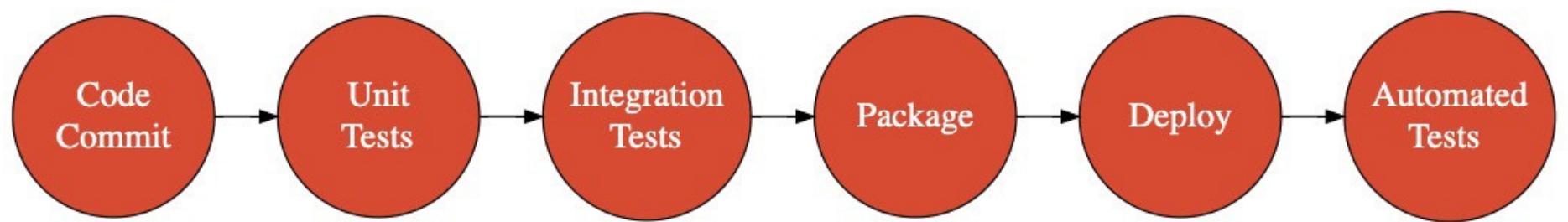
Development

Operations

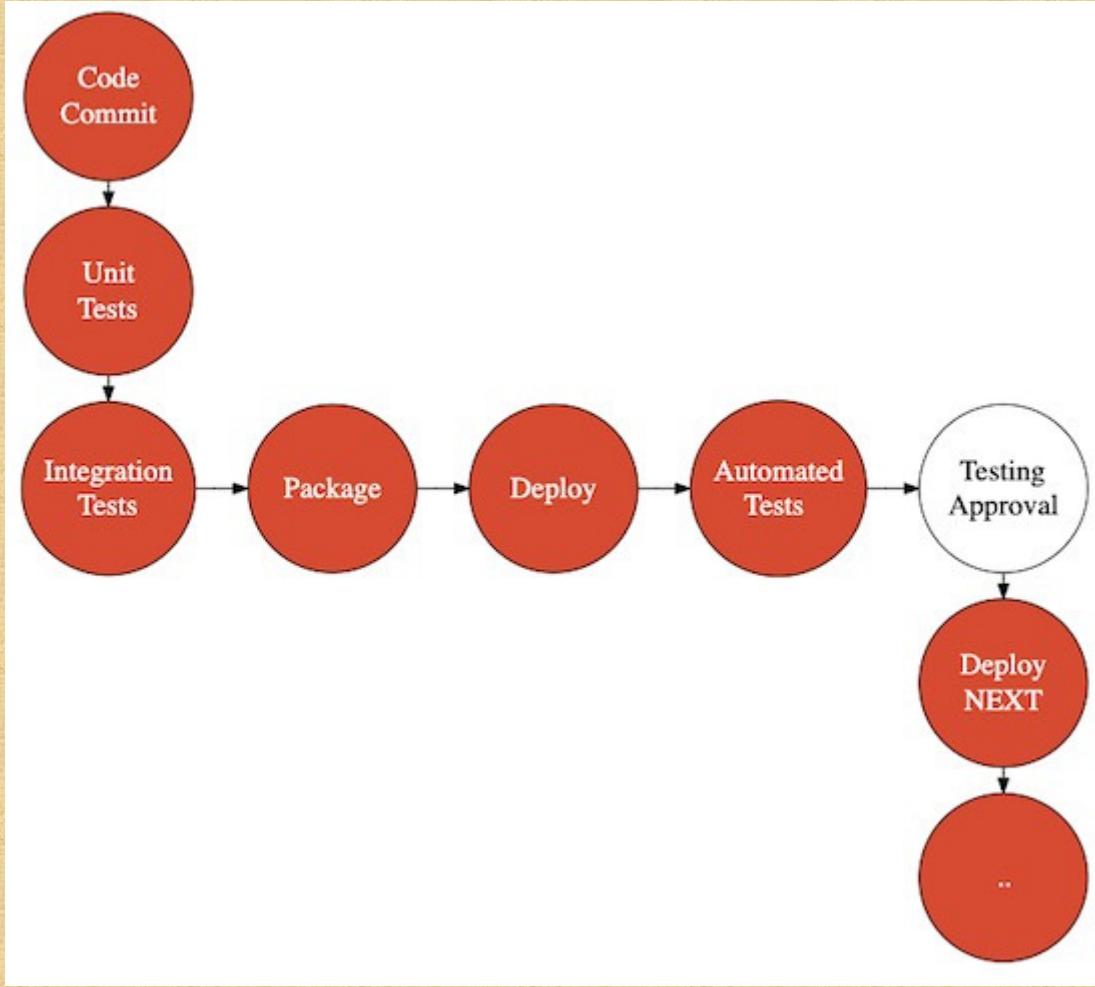
# CI CD CD



# Continuous Integration

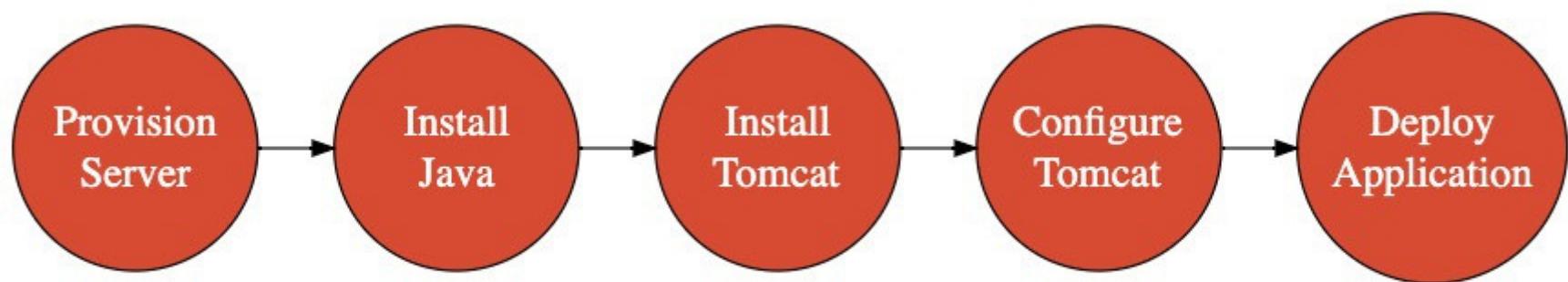


# Continuous Deployment

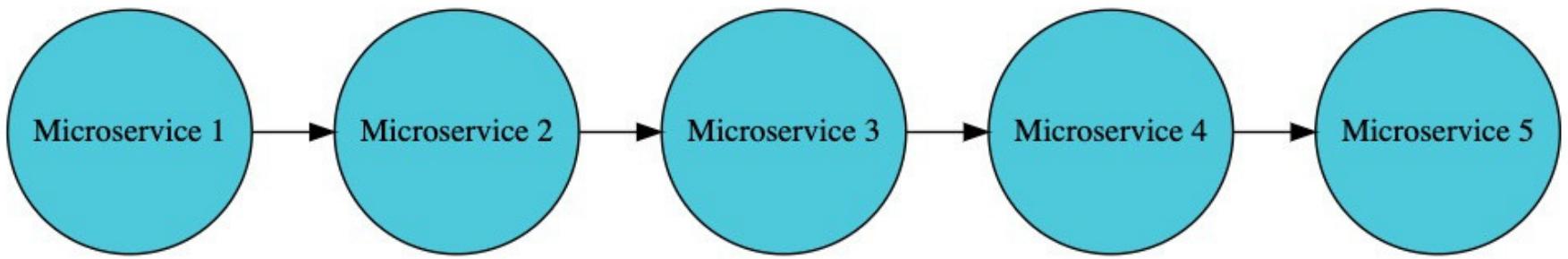


# Continuous Delivery

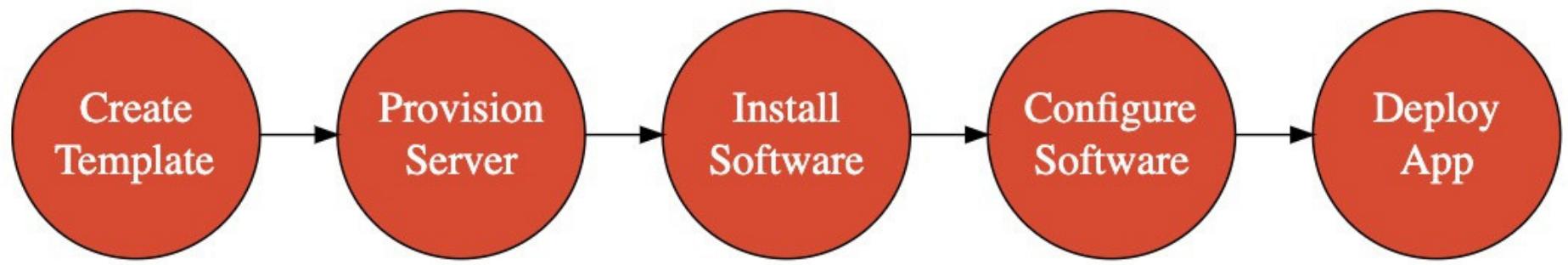
# INFRASTRUCTURE AS CODE



Manual Approach



# Microservices



## Infrastructure as Code

# ADVANTAGES

# Enable Self Provisioning

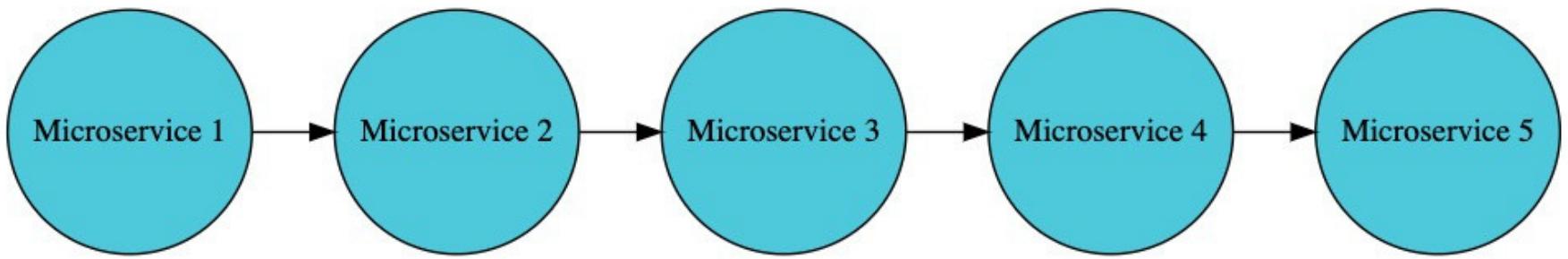
Infra Team can focus on Value  
Added Work

# Consistent Servers

# Less Errors

# Quick Recovery From Failures

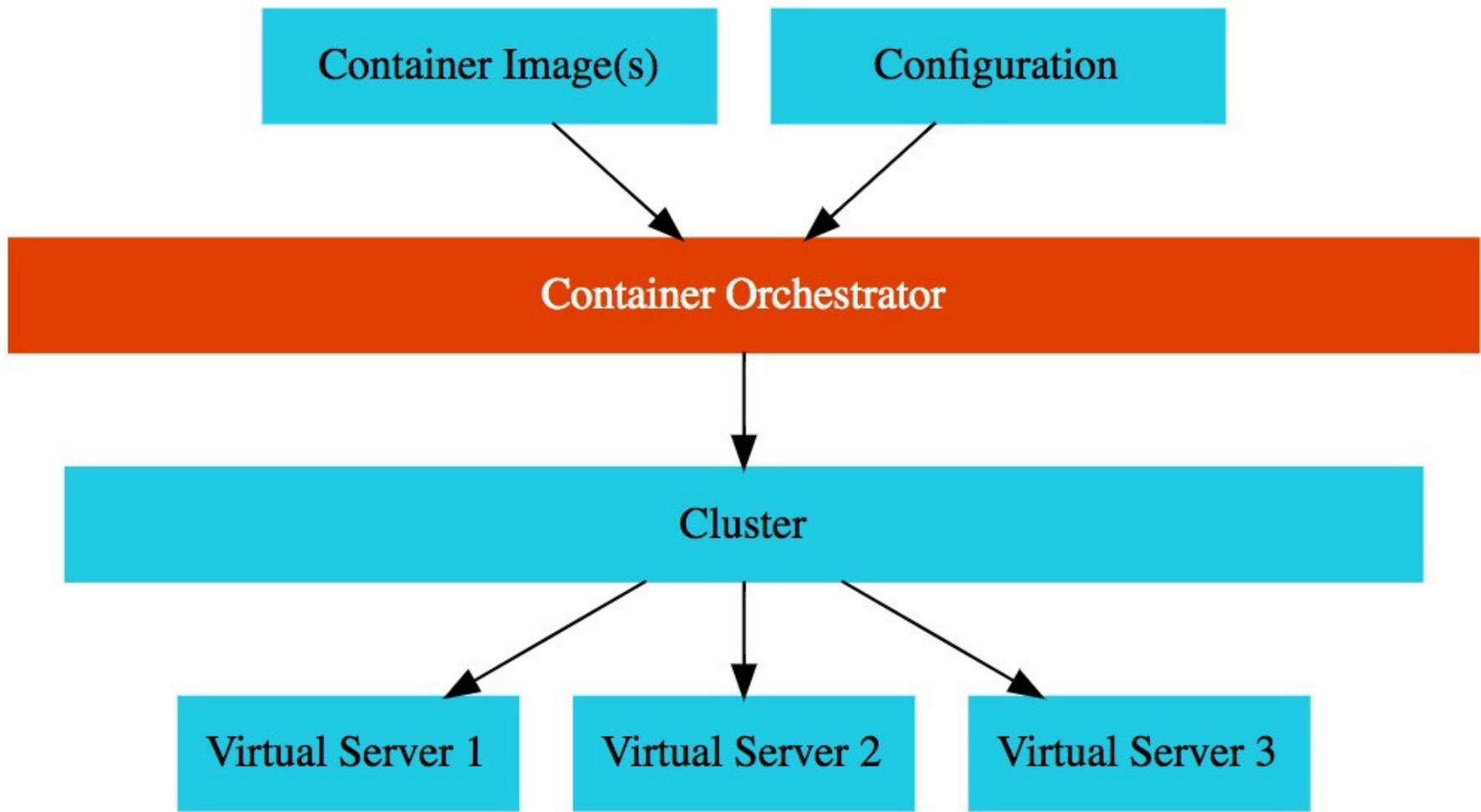
# CONTAINERIZATION

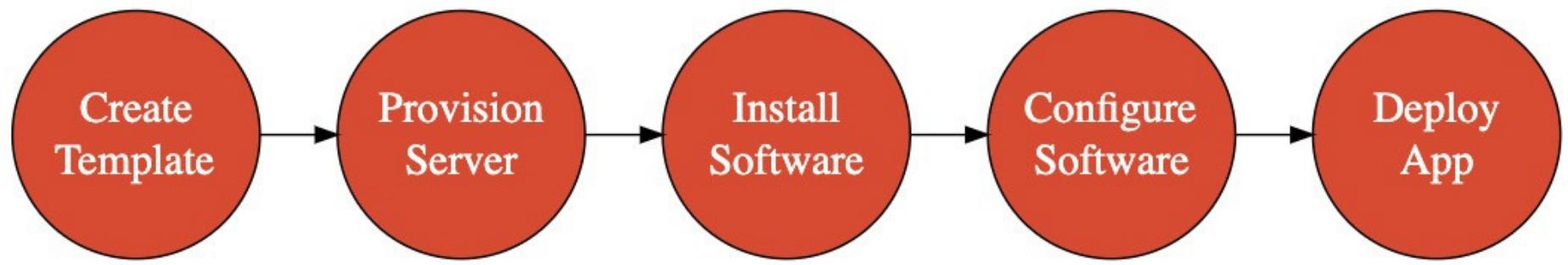


# Microservices

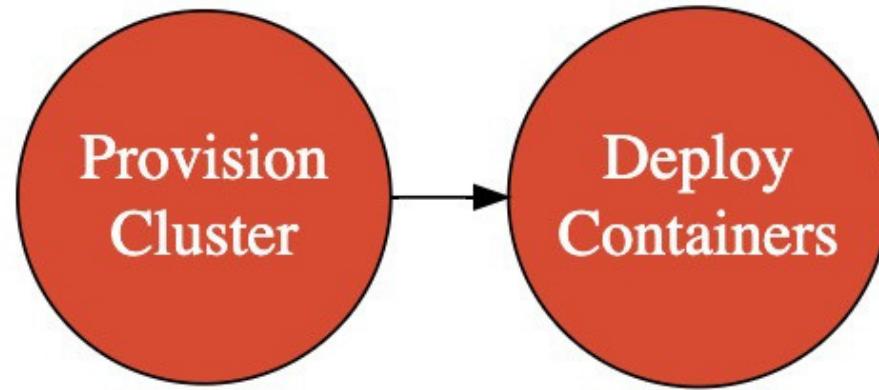


## Containers



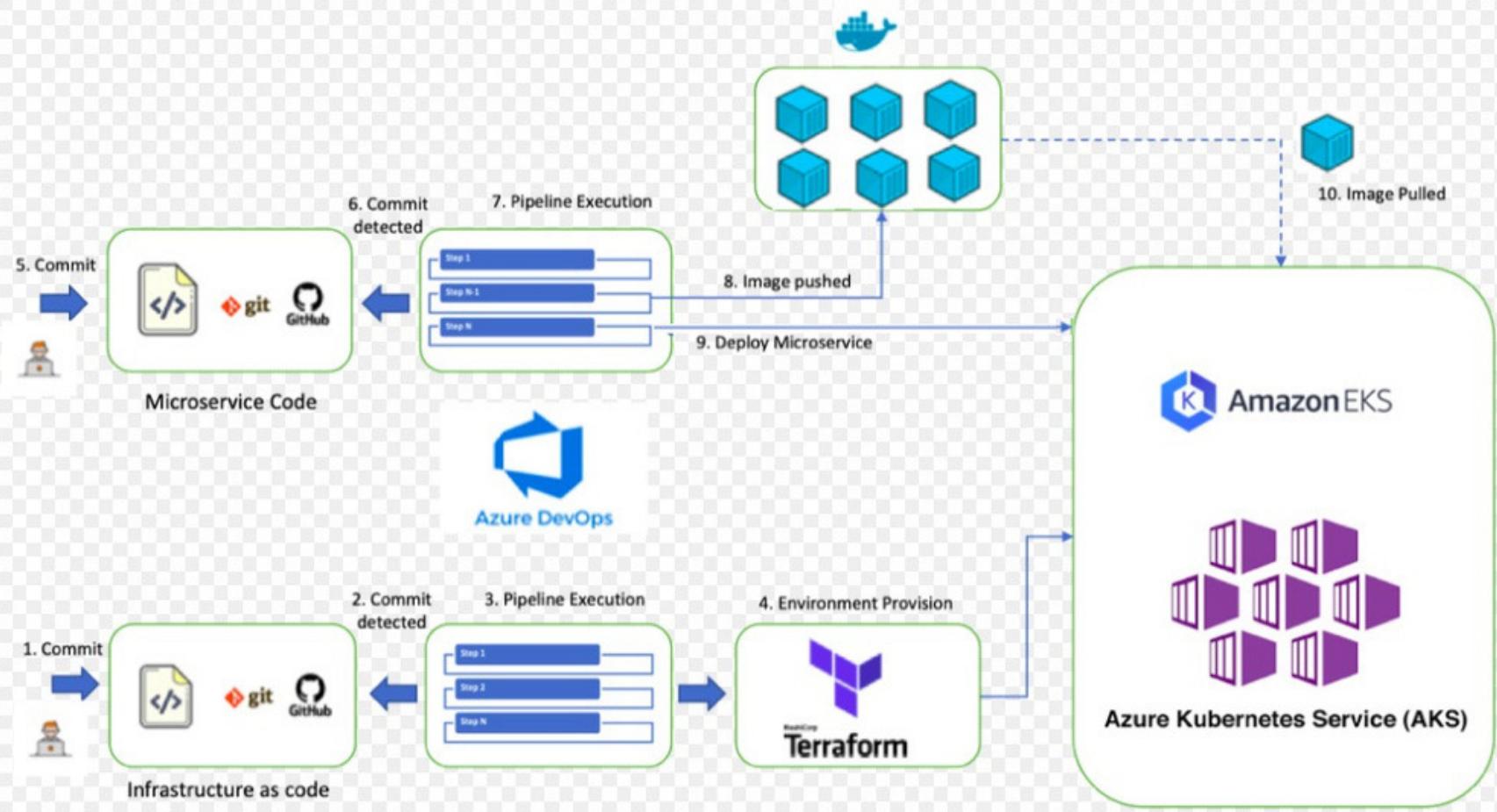


## Infrastructure as Code



## Infrastructure as Code - Containers

# 6 DEVOPS TOOLS



## 2 Example DevOps Use Cases

**1**

**Basics and Best Practices of DevOps**

**2**

**Build and Deploy Images with Docker**

**3**

**Container Orchestration with Kubernetes**

**4**

**Server Provisioning with Terraform**

**5**

**Configuration Management with Ansible**

**6**

**Azure Dev Ops - CI/CD with Pipelines**

**7**

**Jenkins - CI/CD with Pipelines**

# DEVOPS

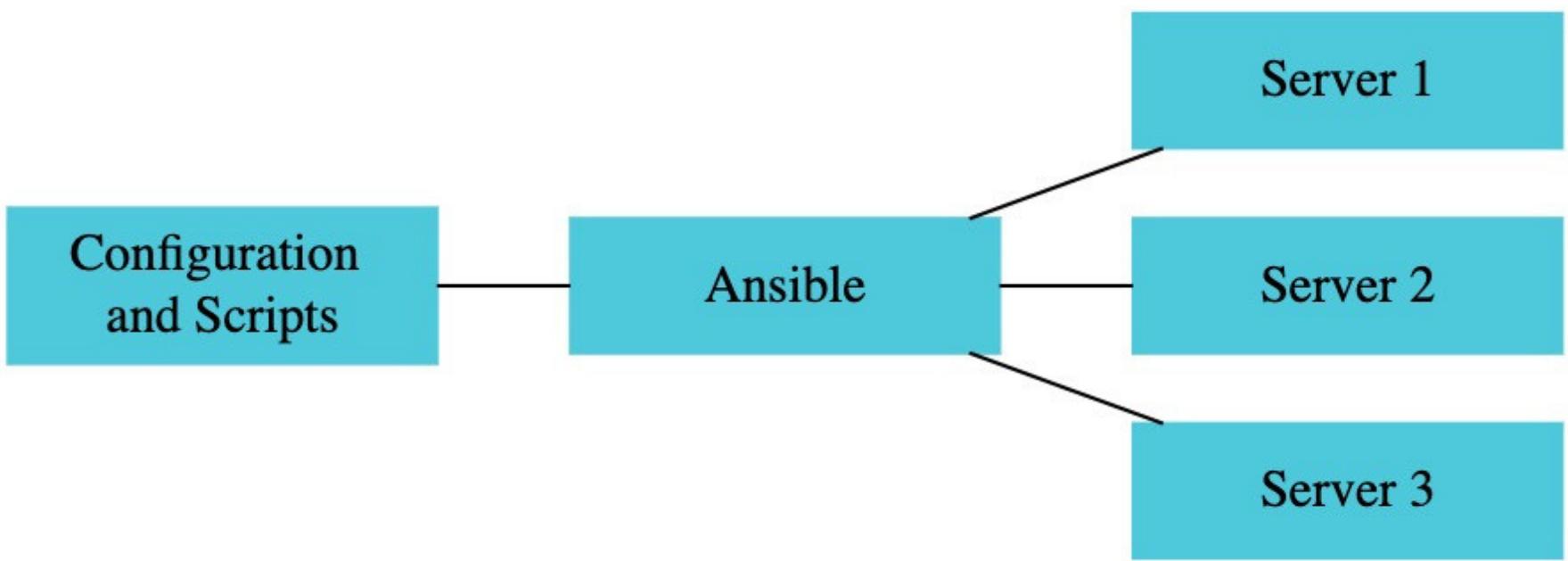
Multiple DevOps Tools  
3 Clouds Constant  
Evolution No Perfect  
Tool Set

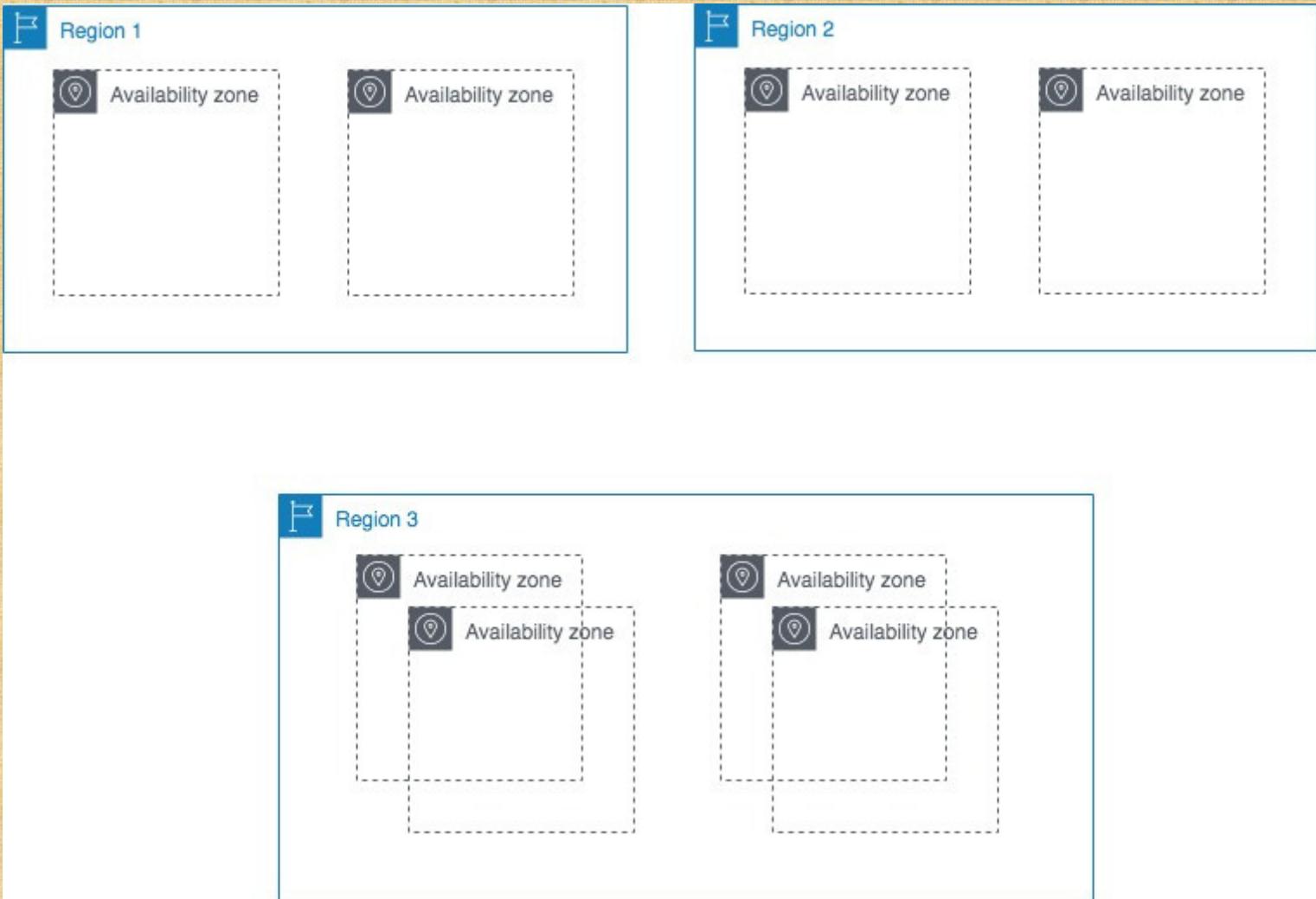
# IAC BEST PRACTICES

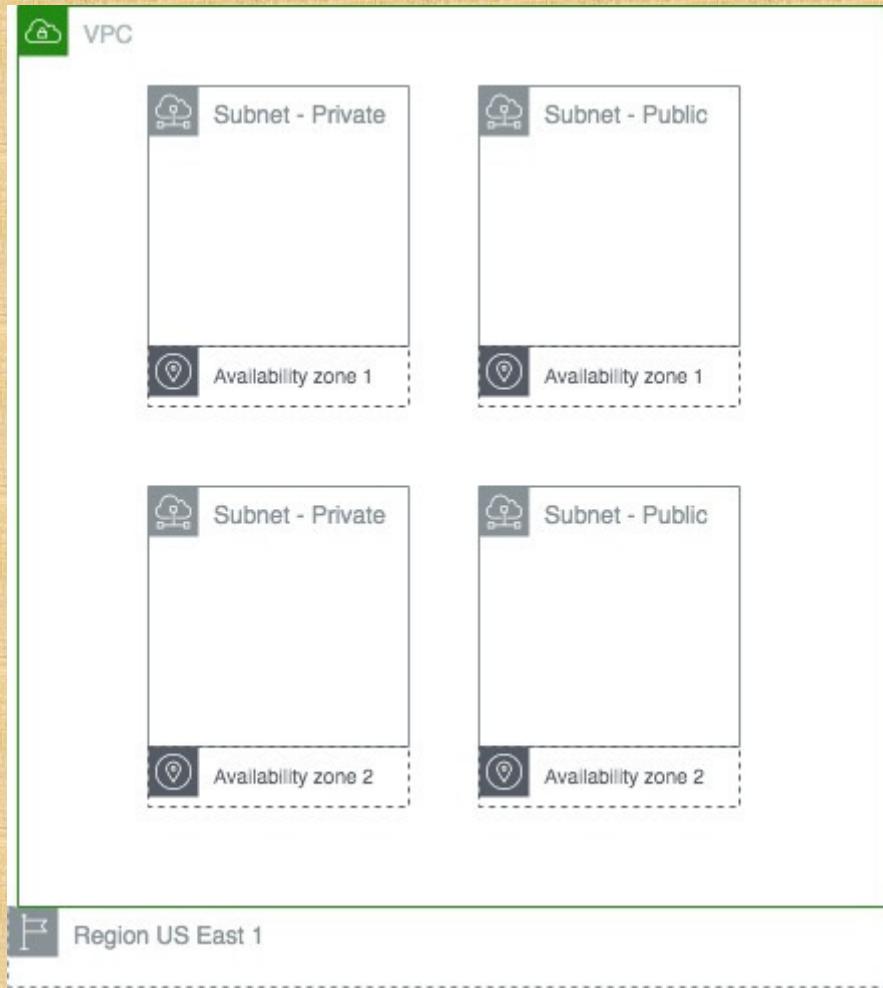
Self Provisioning  
Treat Servers  
as Disposable  
Do not do  
anything manually  
Version Your  
Infrastructure Code  
Do  
incremental changes  
Zero  
Downtime Upgrades

# DOCKER

Master DevOps









Docker  
Software

DevOps  
Topic

+ Add comparison

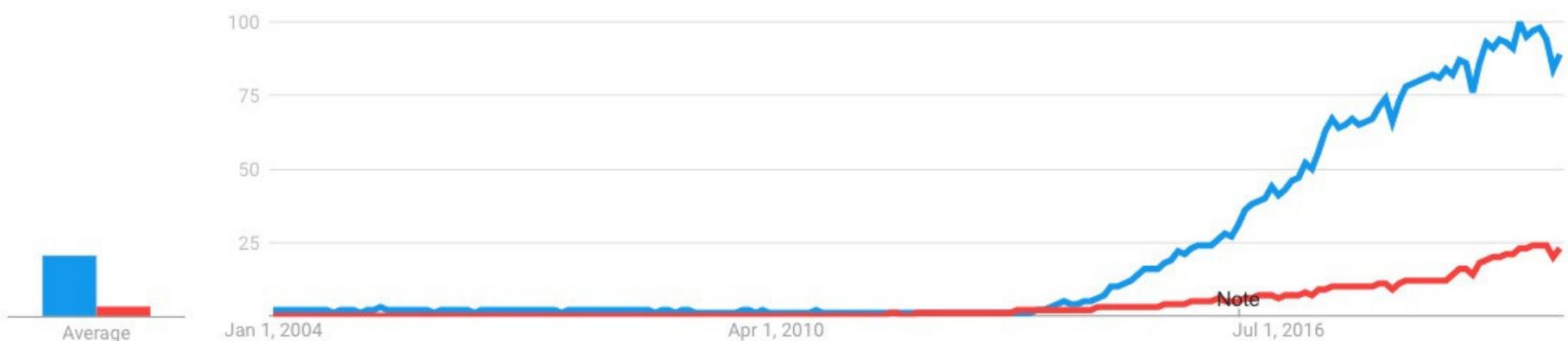
Worldwide ▾

2004 - present ▾

All categories ▾

Web Search ▾

Interest over time ?



## Standardized Application Packaging

Same packaging for all types of applications  
- Java, Python or JS

## Multi Platform Support

Local Machine  
Data Center  
Cloud - AWS, Azure and GCP

## Light-Weight & Isolation

Containers are Light-weight compared to VM's  
Isolated from one another

Docker

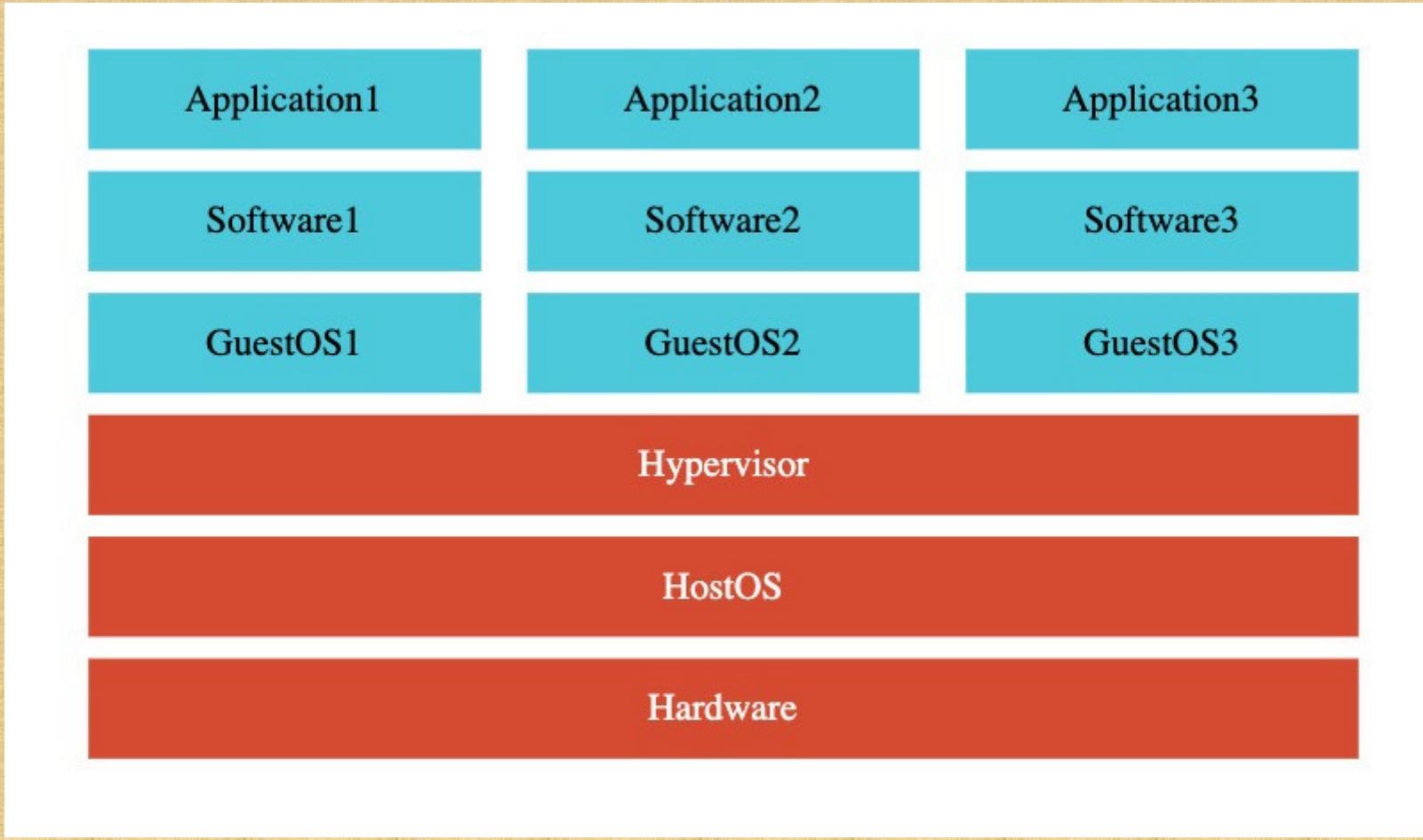
Applications

Software

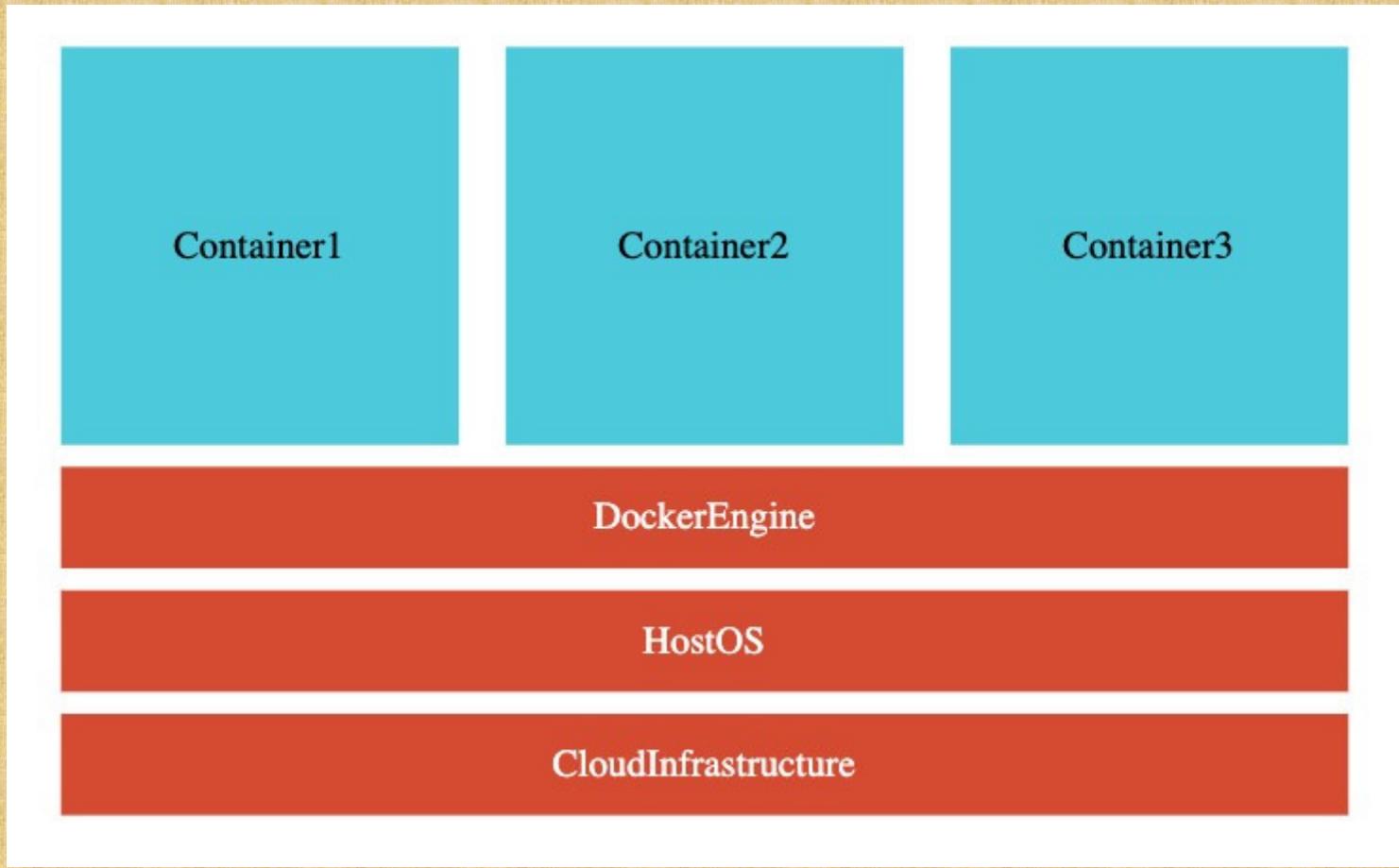
OS

Hardware

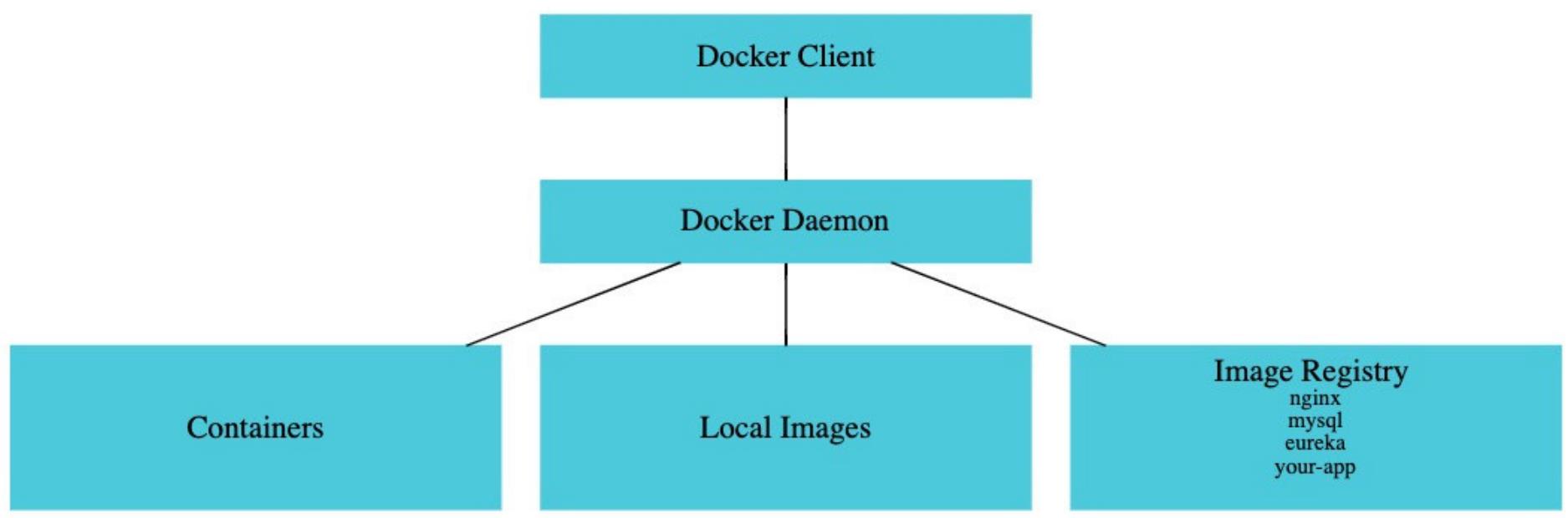
Traditional Deployment



## Deployments using Virtual Machines



## Deployments using Docker



# Docker Architecture

# DOCKER AND DEVOPS

Standardized Communication

More Successful Deployments

No More - "It works in my Local!"

Infrastructure Standardization

Container Orchestration - Kubernetes or AWS ECS

Fargate

# NEXT STEPS

Explore Docker from Operations Perspective  
Pre Built Images  
REST API Web Application talking  
with MySQL  
Docker Compose  
Separate Section for Building Docker Images

## Container Orchestration

Manage 1000's of instances  
1000's of microservices  
Declaratively

## Features

Auto Scaling  
Service Discovery  
Load Balancing  
Self Healing  
Zero Downtime Deployments

## Cloud Neutral

Standardized Platform  
on any infrastructure

# Kubernetes

# YOUTUBE, GOOGLE MAPS AND GOOGLE SEARCH

Google Kubernetes Engine (GKE) on  
Google Cloud Platform (GCP)!

Go Declarative - Go YAML  
Use Helm Stay Cloud  
Neutral

Kubernetes  
Best Practices

# Naming Server Load Balancing Distributed Tracing

Spring Cloud - A Couple Of Years Ago

spring-cloud-starter-kubernetes  
spring-cloud-gcp-starter-trace  
spring-cloud-gcp-starter-logging

Spring Cloud - In the Cloud World

Update the step referring to Docker to refer to  
Appendix Section on Docker.

# Google Console Keyboard Shortcuts

# Debugging Problems in K8s

# Billing

## NodePort Example

[https://cloud.google.com/kubernetes-engine/docs/how-to/exposing-apps#access your service](https://cloud.google.com/kubernetes-engine/docs/how-to/exposing-apps#access_your_service)

# KUBERNETES - FUN FACTS

K8S KOO - BER - NET -  
EEZ Logo - Helmsman  
Kubernetes on Cloud

AKS, Amazon EKS and GKE

Cloud Computing Services | [+](#)

[cloud.google.com](#)

Google Cloud Why Google Solutions Products Pricing Getting started Contact sales

Docs Support Language [Sign in](#)

# Build. Modernize. Scale.

Transform your business with Google Cloud.

[Get started for free](#) [Contact sales](#)



CUSTOMER STORIES

See how DSW relaunched its customer loyalty program,



CUSTOMER STORIES

How Google and Mayo Clinic will transform the future of healthcare

Cloud Computing Services | Step 1 of 2 – Free Trial – Google

console.cloud.google.com/freetrial/signup/tos?\_ga=2.175787298.-1489585026.1572932391&pli=1

Try Google Cloud Platform for free

## Step 1 of 2

Country

India

Terms of Service

I agree to the [Google Cloud Platform Terms of Service](#), and the terms of service of [any applicable services and APIs](#). I have also read and agree to the [Google Cloud Platform Free Trial Terms of Service](#).

Required to continue

CONTINUE

**Reliability and confidence, at scale**

You'll be leveraging one of the world's largest networks, empowering you to reach hundreds of millions of users with our future-proof infrastructure – 99.95% availability and no scheduled downtimes.

**\$300 credit for free**

Enjoy free, limited usage of BigQuery, Compute, Storage, and other top products for 12 months. We'll also give you a \$300 credit, on us.

**No autocharge after free trial ends**

Your payment information is only used to verify your identity – we're just making sure you are not a robot.



Account type 

Individual



### Tax information

Tax status

Unregistered individual 

PAN (optional)

Permanent account number example: ABCDE1234A

TAN (optional)

Tax deduction and collection account number example:  
DELA02603G

## How you pay



Monthly automatic payments

You pay for this service on a regular **monthly** basis, via an automatic charge when your payment is due.

## Payment method ⓘ

### # Card details

---

The personal information you provide here will be added to your payments profile. It will be stored securely and treated in accordance with the [Google Privacy Policy](#).

**START MY FREE TRIAL**

X Verify Mastercard .... 6000



Mastercard .... 6000

Enter the 3-digit security code on your card

Security code



CONTINUE

## Complete this transaction

When you click Continue, a new browser window will appear where you'll verify your info.

After you're done, you may see a small charge to your credit card. This is temporary and will be refunded.

**CONTINUE**

×

Uh oh, something went wrong

Your card does not support automatic  
recurring payments. [OR-CC3ST-02]

OK



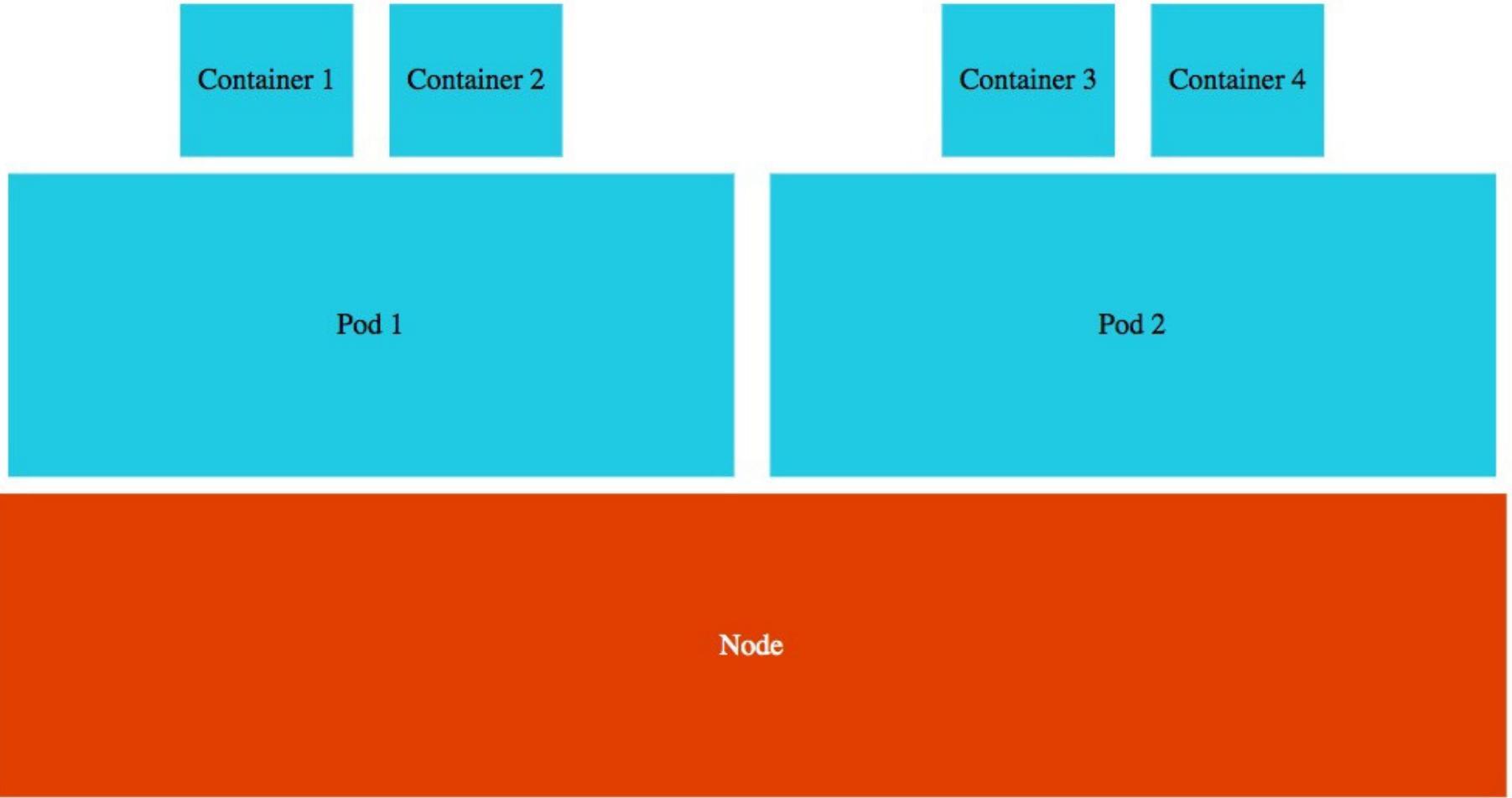
Google Cloud Platform

## Welcome Ranga!

Thanks for signing up for the 12-month free trial.

Thanks for signing up. Your free trial includes \$300 in credit to spend over the next 12 months. If you run out of credit, don't worry — you won't be billed without your permission.

GOT IT



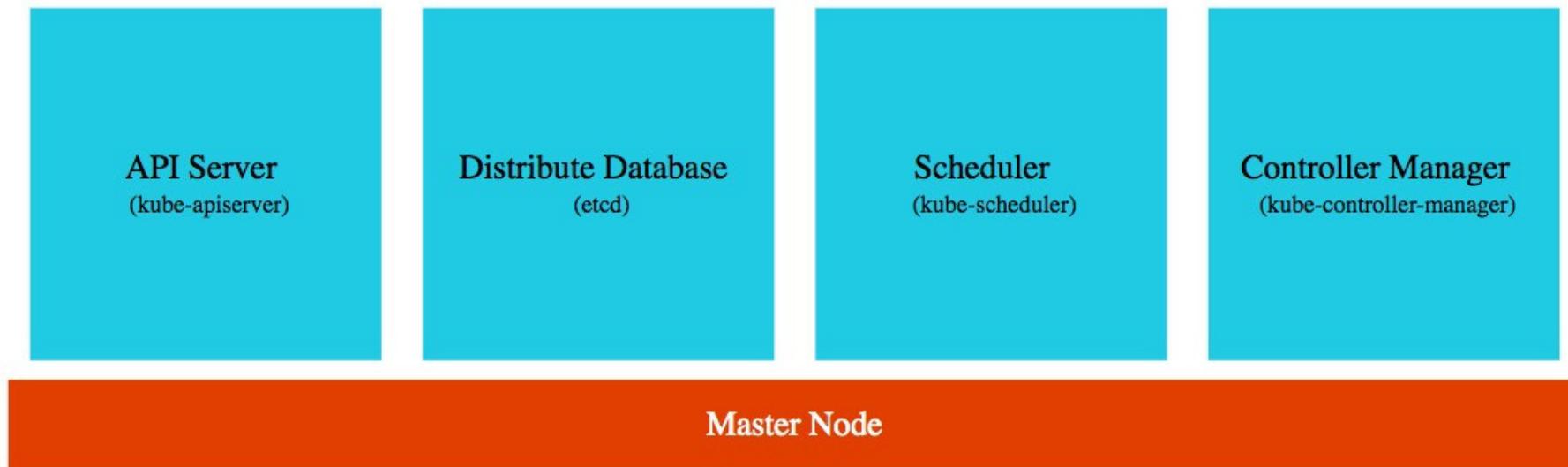
# Kubernetes Architecture

**Master Node(s)**  
Manages Cluster

**Worker Node(s)**  
Run Your Applications

Cluster

## Kubernetes Architecture



# Kubernetes Architecture

**Node Agent**  
(kubelet)

**Networking Component**  
(kube-proxy)

**Container Runtime**  
(CRI - docker, rkt etc)

**PODS**  
(Multiple pods running  
containers)

Worker Node (or) Node

# Kubernetes Architecture



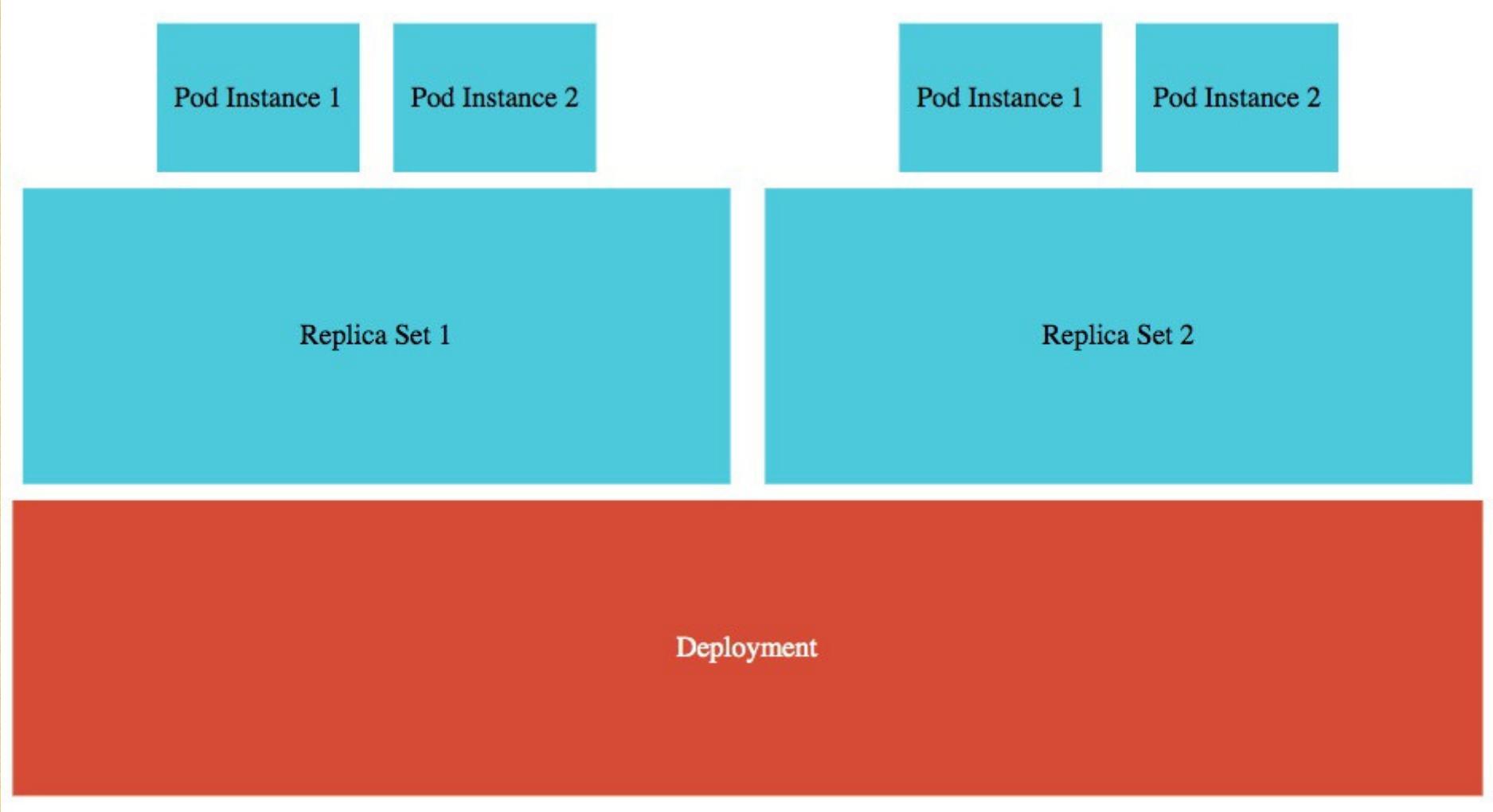
## Kubernetes Deployments

Create Cluster

Create Deployment

Docker Repository

## Kubernetes Deployments



# Kubernetes Deployments



## Kubernetes Service

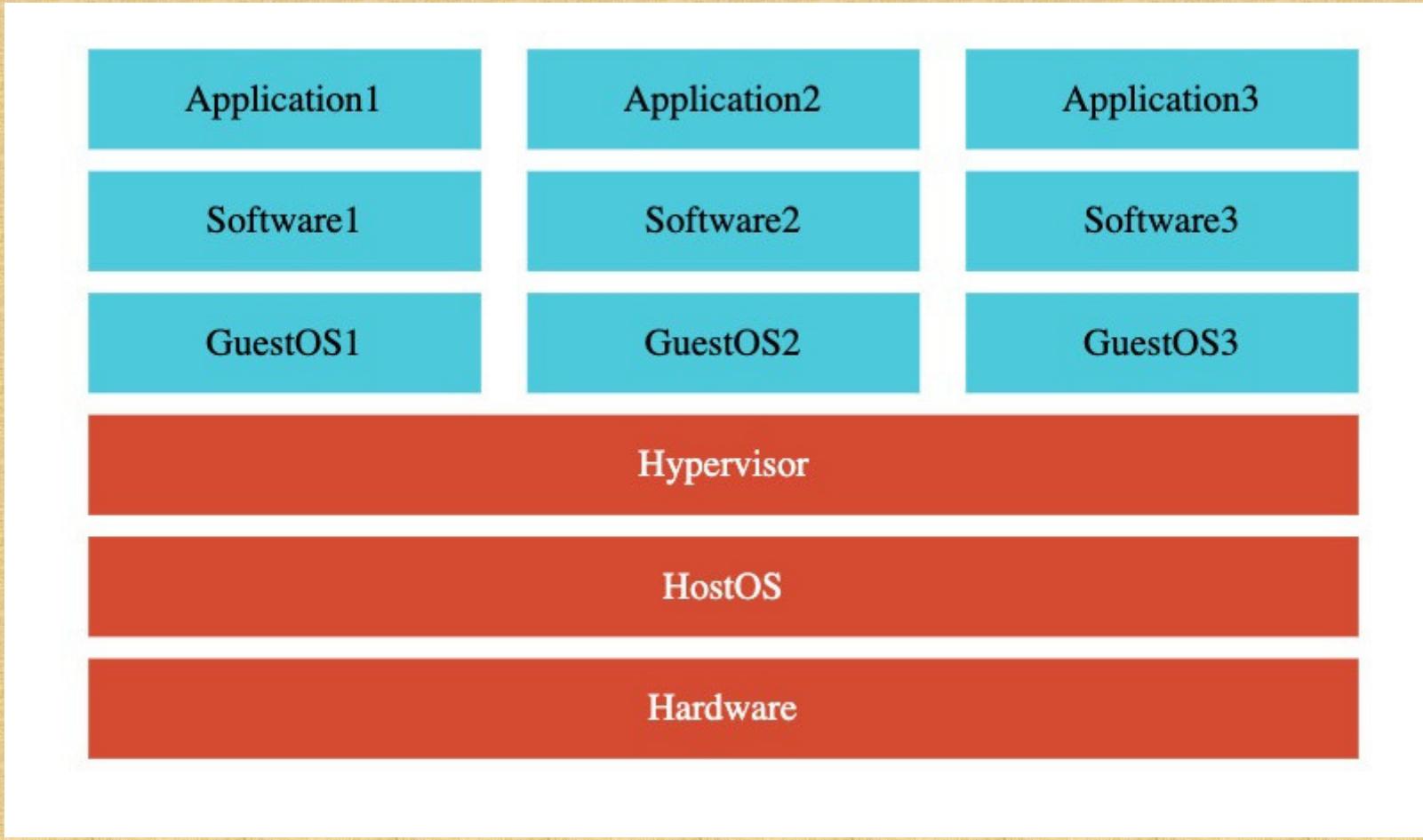
Applications

Software

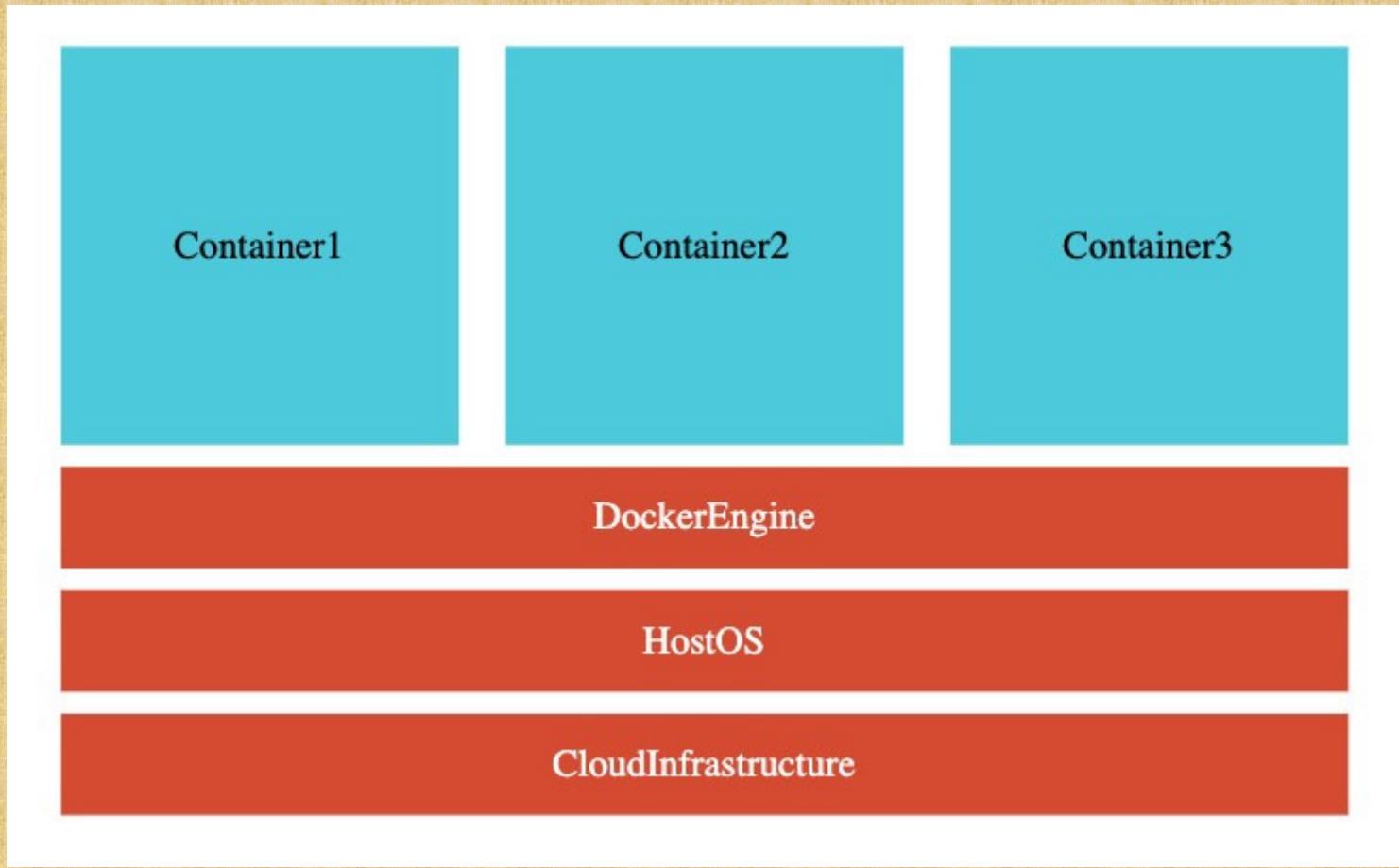
OS

Hardware

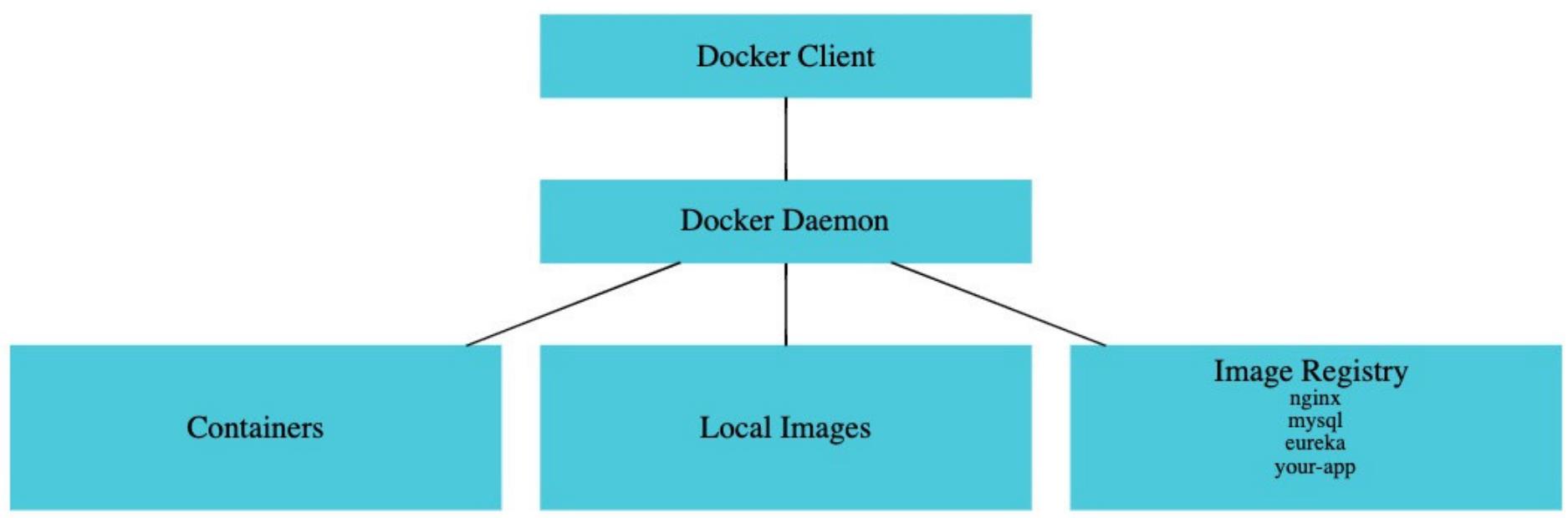
Traditional Deployment



## Deployments using Virtual Machines

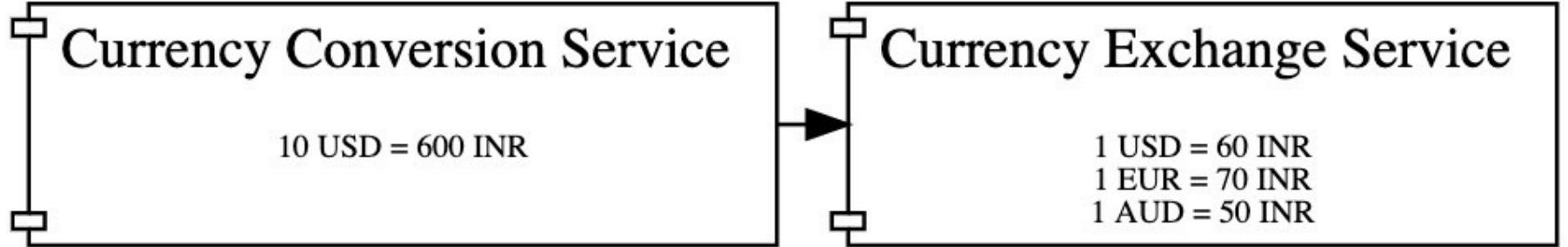


## Deployments using Docker



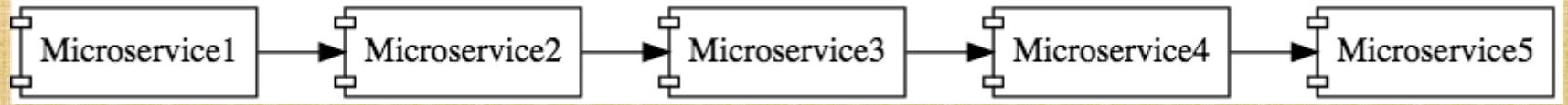
# Docker Architecture

# MICROSERVICES



## Microservices Overview

# DOCKER AND MICROSERVICES



# Microservices Chain

# EASIER DEVELOPMENT

Adopt New Technology Faster

Zero worry about deployment procedures

Fewer Environment Issues

No more - "It works in my Local"

# EASIER OPERATIONS

Consistent Deployment  
Automation Across Different  
Environments and Different  
Technologies

# KUBERNETES AND MICROSERVICES

# EASIER DEVELOPMENT

Adopt New Technology Faster

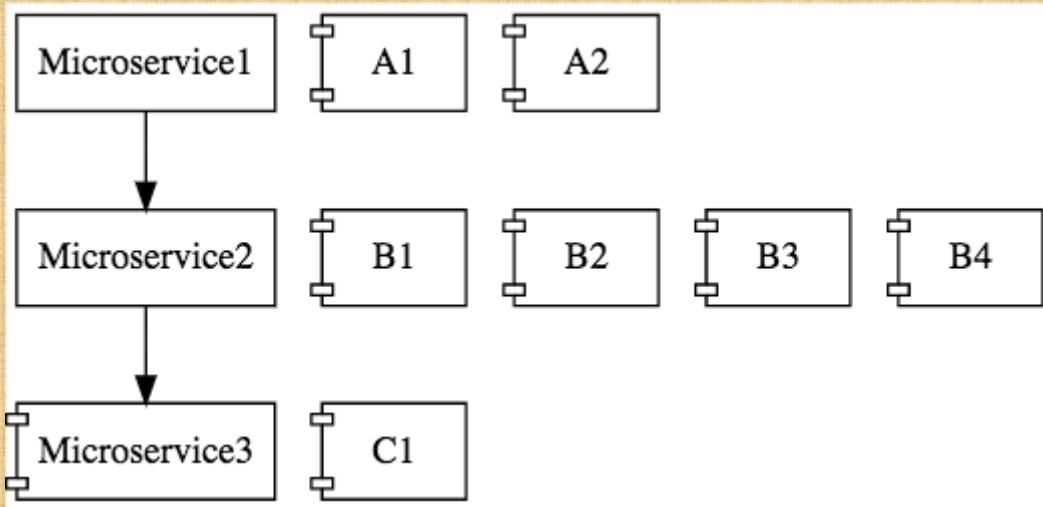
Zero worry about deployment procedures

Fewer Environment Issues

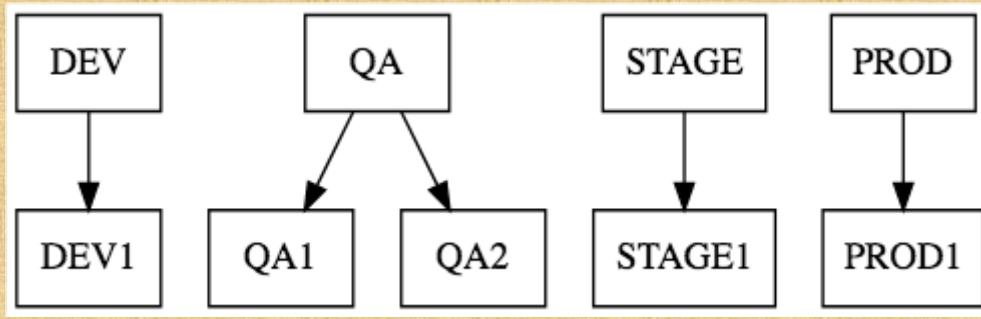
No more - "It works in my Local"

# EASIER OPERATIONS

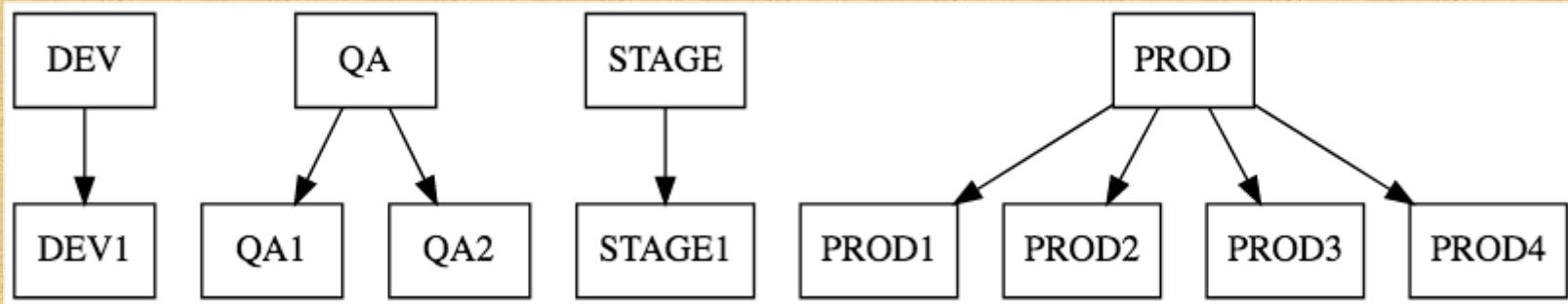
Consistent Deployment  
Automation Across Different  
Environments and Different  
Technologies



## Microservices Multiple Instances

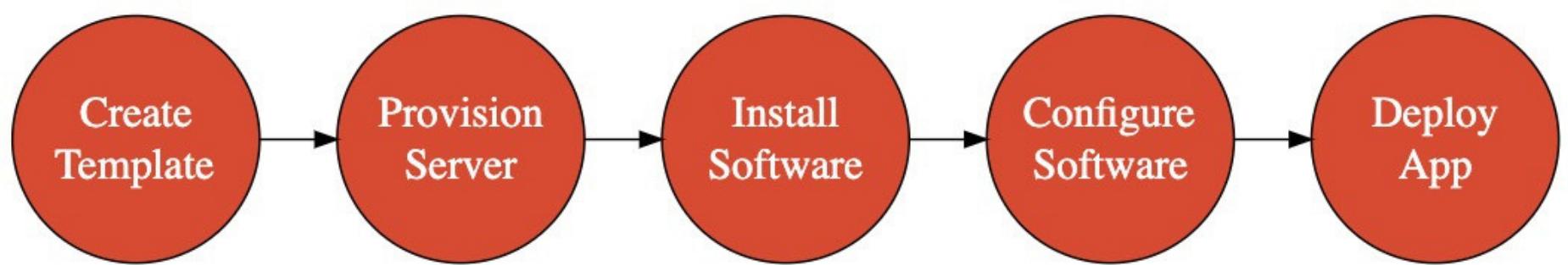


# Currency Conversion Service



# Currency Exchange Service

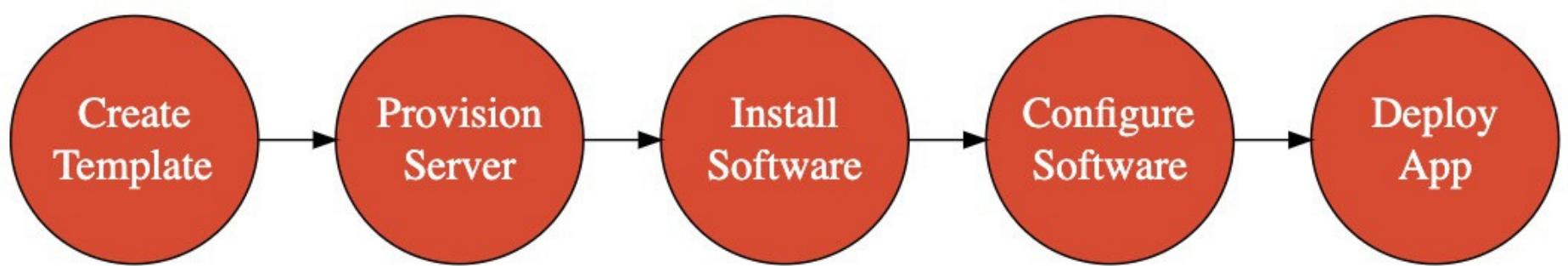
# TERRAFORM



# PREREQUISITES

AWS Account  
Visual Studio Code  
Terraform Installation

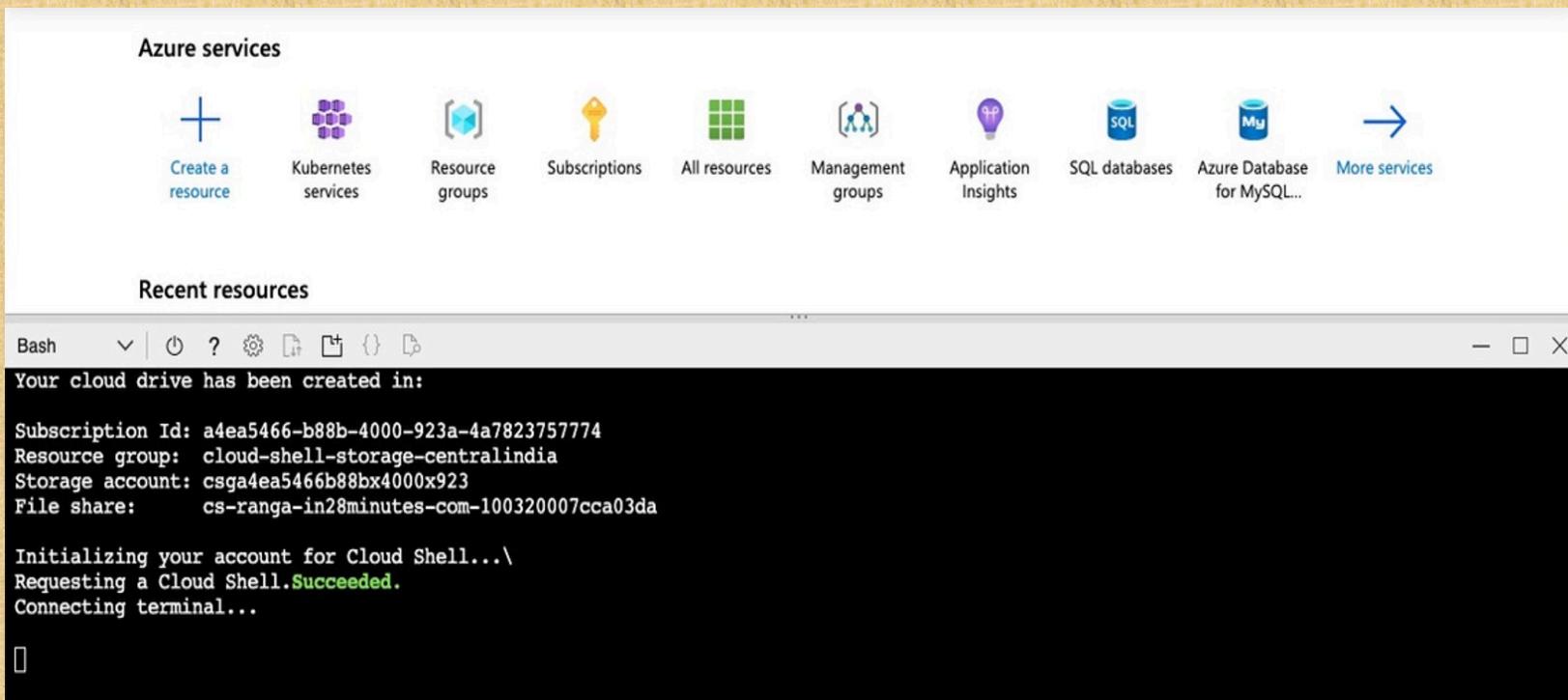
# ANSIBLE



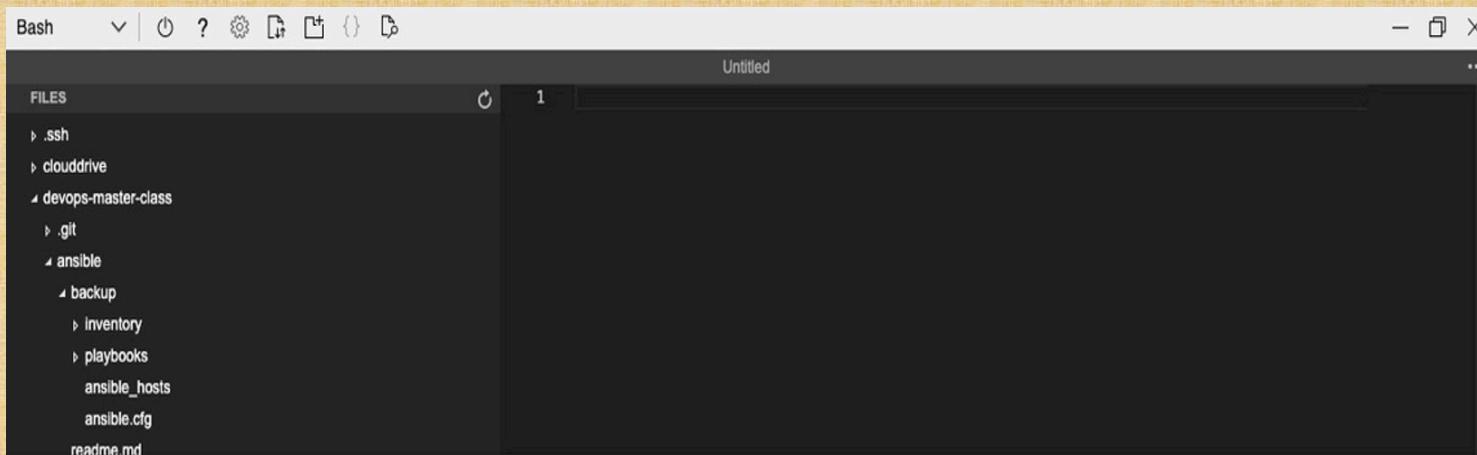
# PREREQUISITES

AWS Account Visual  
Studio Code  
Ansible Installation  
Python 2.7 or 3.4 Installed

# ANSIBLE FOR WINDOWS



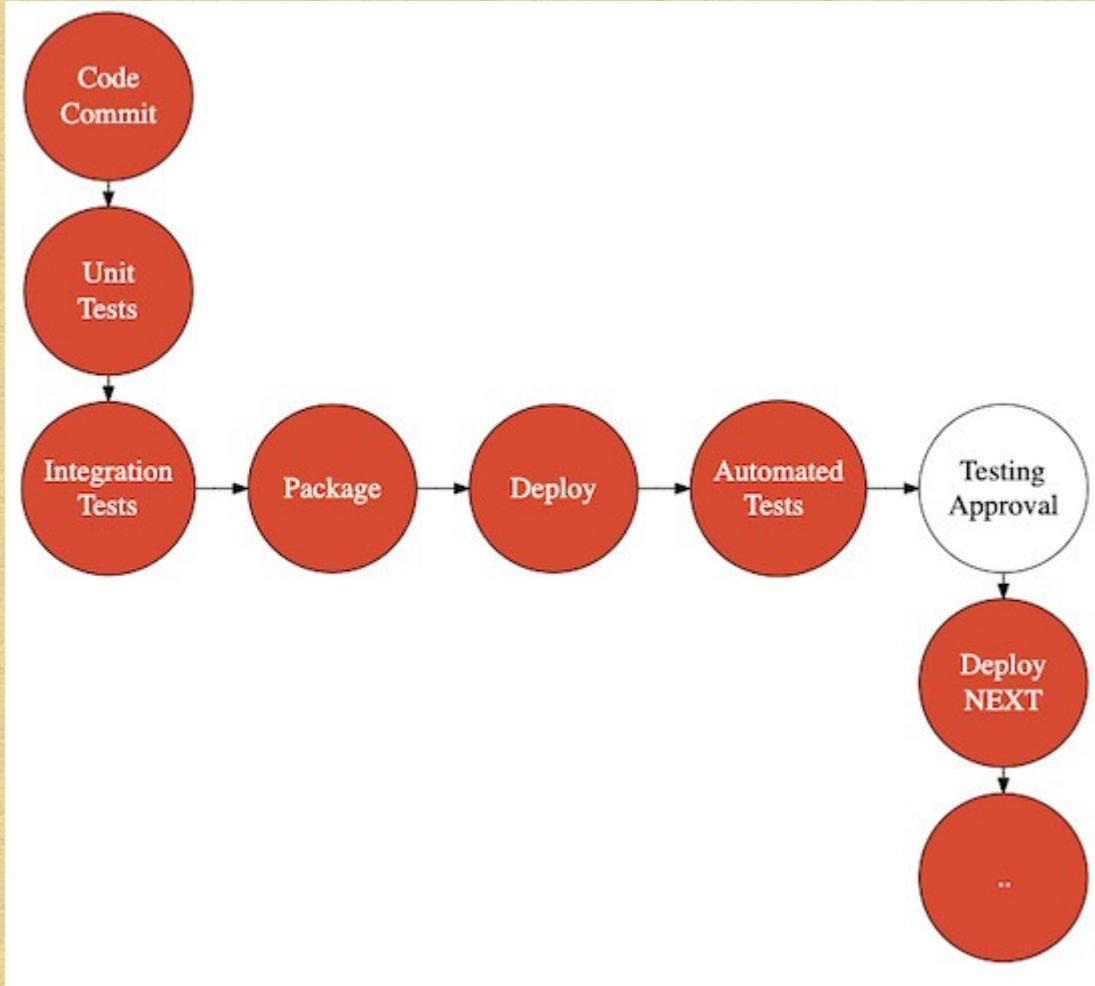
```
nholu@nholuongs-MacBook-Pro Github % git clone git@github.com:nholuongut/devops-for-beginners.git
Cloning into 'devops-for-beginners'...
cd devops-for-beginners
warning: You appear to have cloned an empty repository.
```



# ANSIBLE DYNAMIC INVENTORY

Install boto3 and botocore

# AZURE DEVOPS

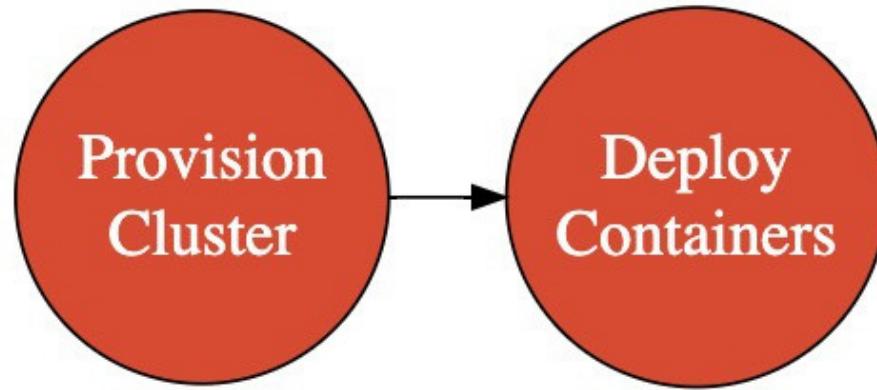


# Continuous Delivery

# PREREQUISITES

Azure Account  
Visual Studio Code  
Section : Docker

# AZURE DEVOPS AZURE AKS WITH TERRAFORM

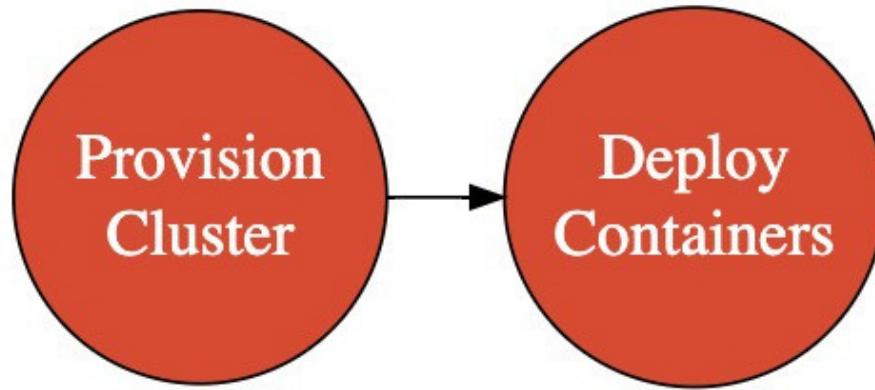


## Azure K8S with Terraform

# PREREQUISITES

Azure Account Visual  
Studio Code Section :  
Azure DevOps Section :  
Docker Section :  
Kubernetes Section :  
Terraform

# AZURE DEVOPS AWS EKS WITH TERRAFORM



## AWS EKS (Kubernetes) with Terraform

# PREREQUISITES

WARNING! NOT AWS FREE TIER!

Azure Account AWS Account

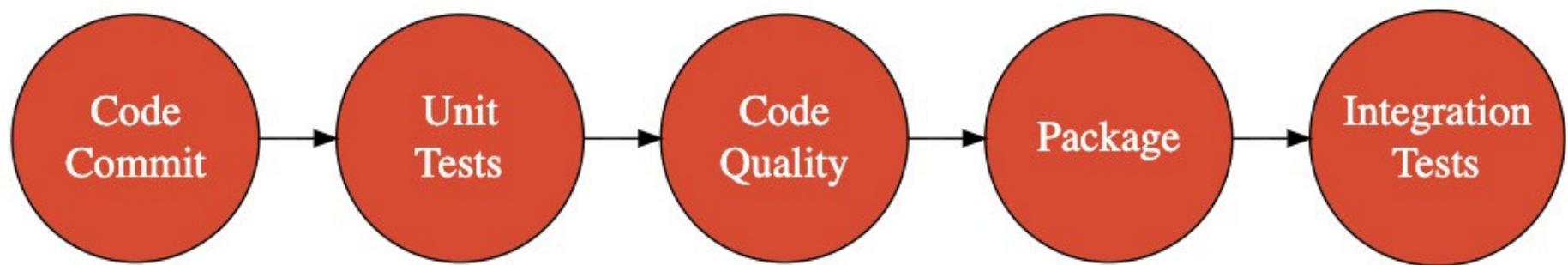
Visual Studio Code Section :

Azure DevOps Section : Docker

Section : Kubernetes Section :

Terraform

# JENKINS

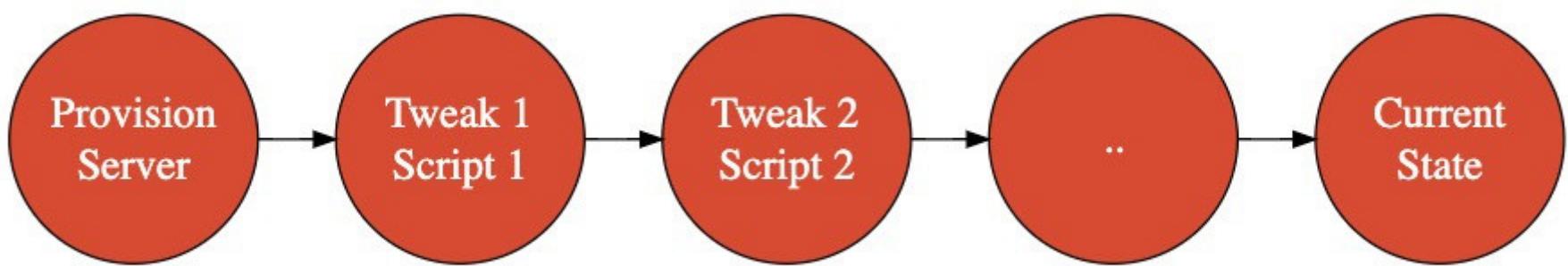


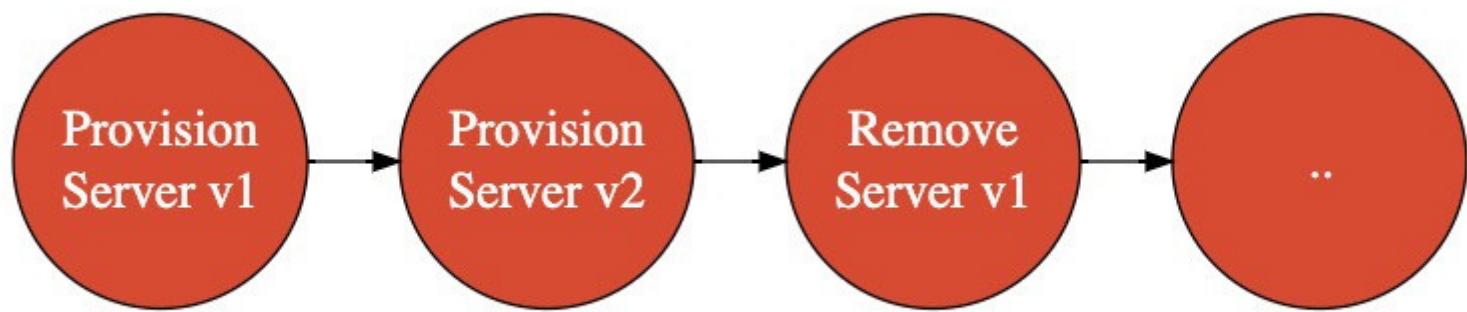
# Continuous Integration

# PREREQUISITES

Visual Studio Code  
Section : Docker  
Docker Compose

# IMMUTABLE SERVERS



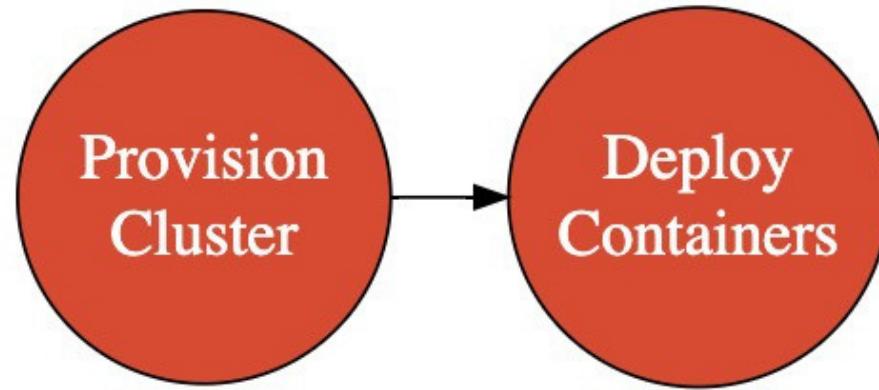


# AZURE DEVOPS PIPELINES

01-first-azure-pipeline 02-understanding-stages 03-playing-with-environment-deployment 04-build-and-push-docker-image 05-azure-kubernetes-cluster-iaac-pipeline 06-azure-kubernetes-code-ci-cd-pipeline 07-aws-kubernetes-cluster-iaac-pipeline 08-aws-kubernetes-code-ci-cd-pipeline

# AZURE DEVOPS PIPELINES

07-aws-kubernetes-cluster-iaac-pipeline  
08-aws-kubernetes-code-ci-cd-pipeline



## Kubernetes Clusters with Terraform

# DEVOPS - CAMS

Culture  
Automation  
Measurement  
Sharing

# DEVOPS METRICS

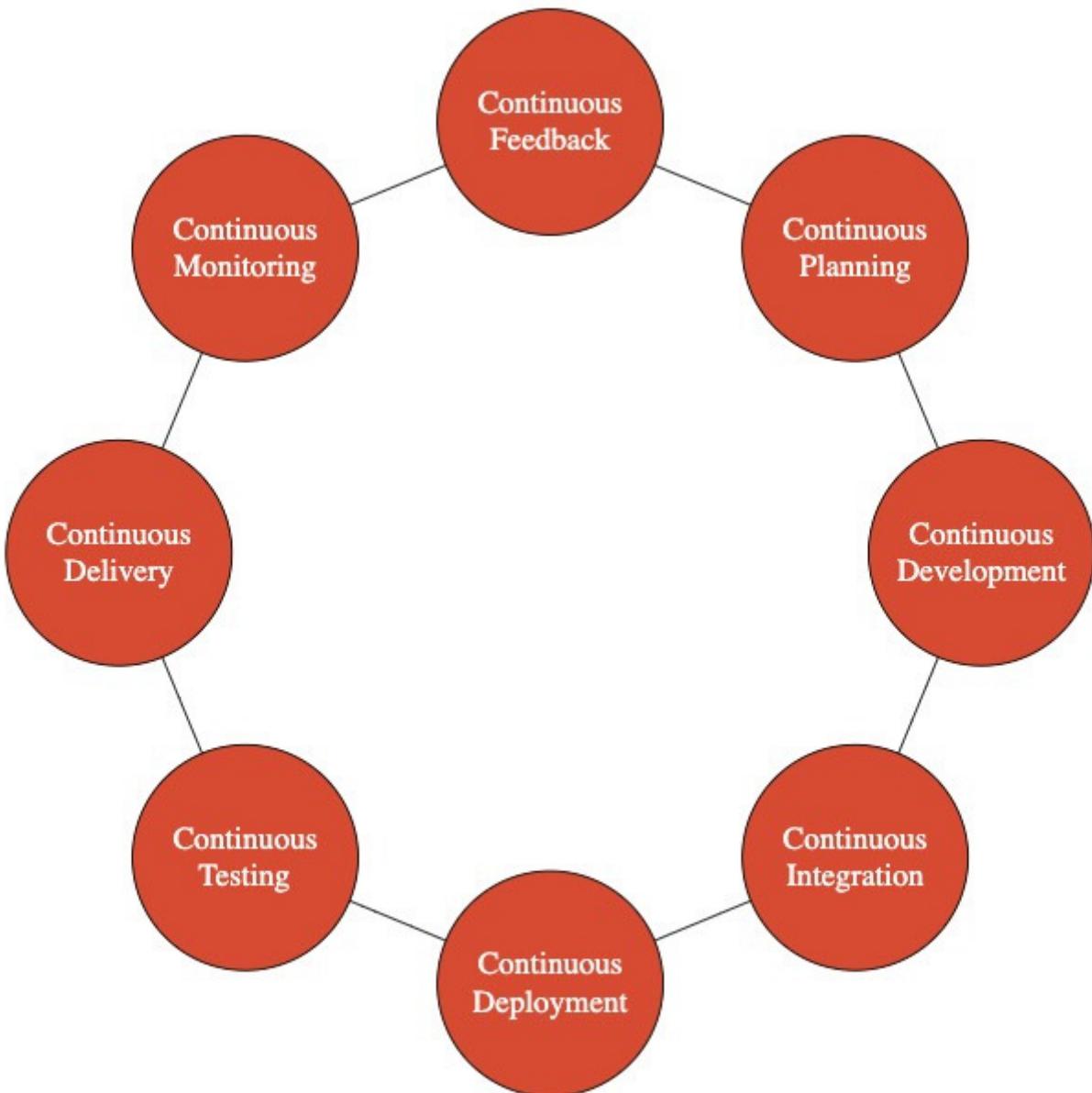
Deployment Frequency Time  
To Market Failure Rate of  
New Releases Lead Time to  
Fixes Mean Time to Recovery

# DEVOPS BEST PRACTICES

Standardization Teams with  
Cross Function Skills Focus on  
Culture Automate, Automate and  
.. Immutable Infrastructure Dev  
Prod Parity Version Control  
Everything Self Provisioning

# DEVOPS CULTURE

What would you do if something is difficult?  
Focus on the End Goal  
Continuous Improvements Culture of Learning and Sharing  
Local Discoveries > Global Improvements



# DEVOPS - 7Cs

# DEVOPS Maturity Assessments

# DEVOPS MATURITY SIGNALS

## Development

Does every commit trigger automated tests and automated code quality checks?

Is your code continuously delivered to production?

Do you use pair programming?

Do you use TDD and BDD?

Do you have a lot of re-usable modules?

Can development teams self provision environments?

How long does it take to deliver a quick fix to production?

# DEVOPS MATURITY SIGNALS

## Test

Are your tests full automated with high quality production like test data?

Does your builds fail when your automated tests fail?

Are your testing cycles small?

Do you have automated NFR tests?

# DEVOPS MATURITY SIGNALS

## Deployment

Do you have Dev Prod Parity?

Do you use A/B Testing?

Do you use canary deployments?

Can you deploy at the click of a button?

Can you rollback at the click of a button?

Can you provision and release infrastructure at the click of a button?

Do you use IAC and version control your infrastructure?

# DEVOPS MATURITY SIGNALS

## Monitoring

Does the team use a centralized monitoring system?

Can development team get access to logs at the click of a button?

Does the team get an automated alert if something goes wrong in production?

# DEVOPS MATURITY SIGNALS

## Teams and Processes

Is the team looking to continuously improve?

Does the team have all the skills it needs from Business,  
Development and Operations?

Does the team track the key DevOps metrics and improve on  
them?

Do you have the culture of take Local Discoveries and using  
them to make Global Improvements?

# DEVOPS TRANSFORMATIONS

Leadership Buy-in is Critical  
Involves Upfront Costs  
Setup COEs to help teams Choose the right  
application and team

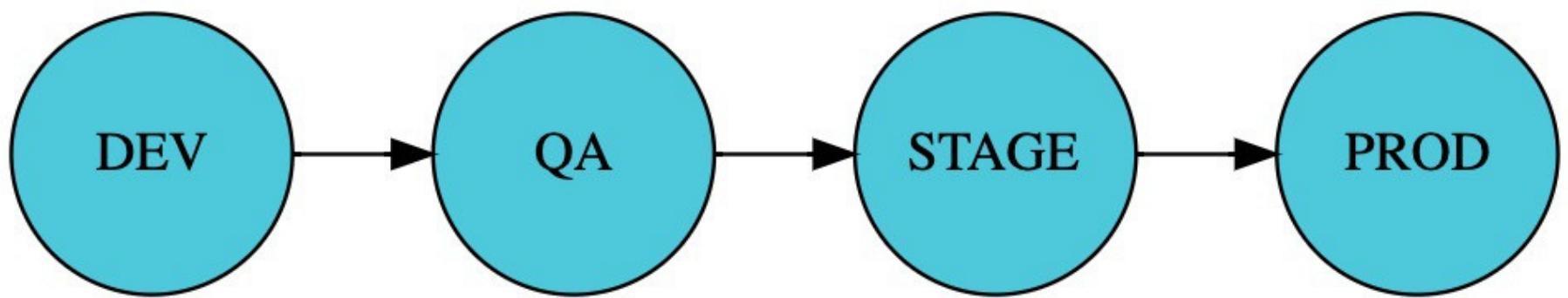
# DEVOPS TRANSFORMATIONS

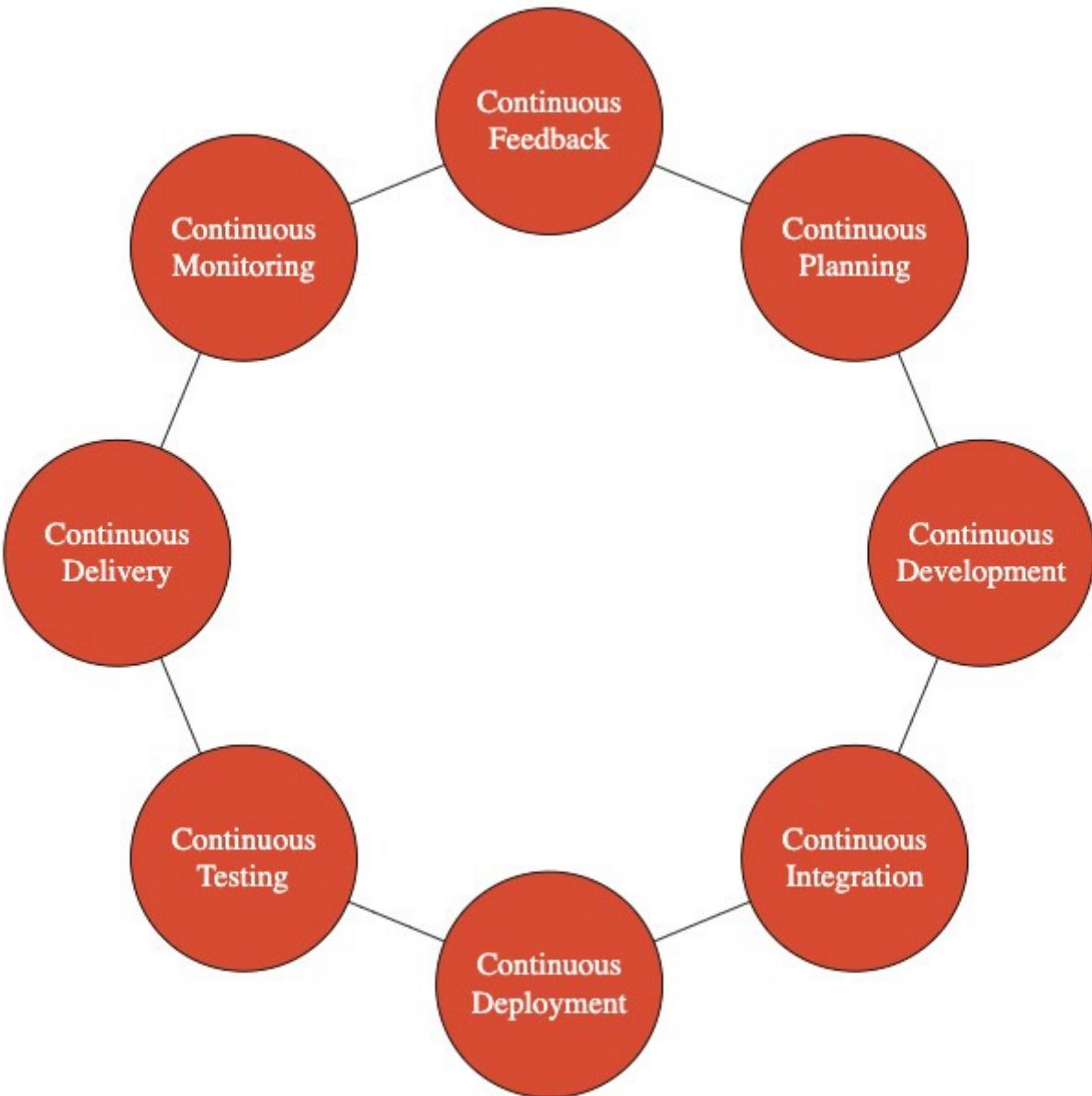
Start Small Sharing Learnings (Newsletters, Communication, COEs)

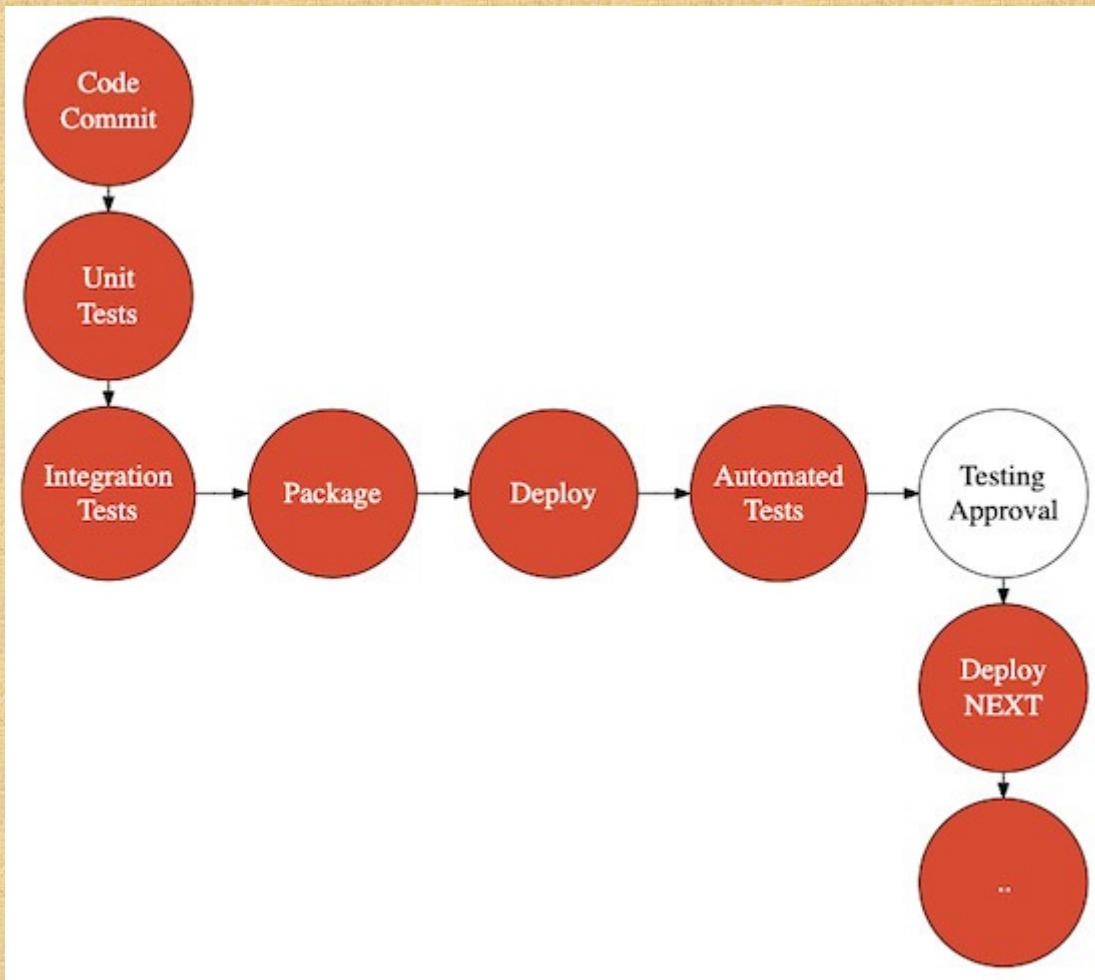
Encourage People with Exploration and Automation Mindset

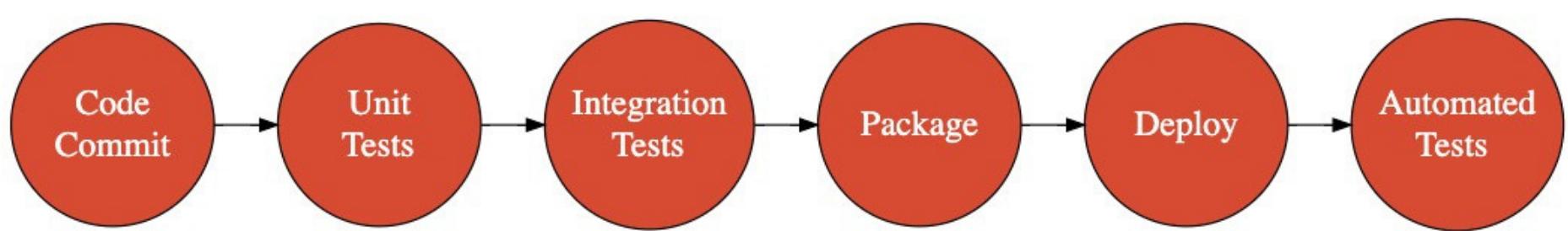
Recognize DevOps Teams

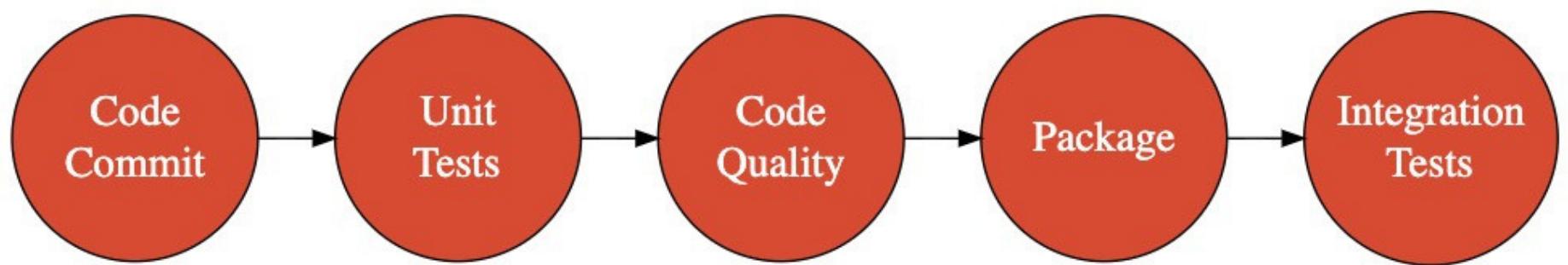
- 1 Basics and Best Practices of DevOps**
- 2 Build and Deploy Images with Docker**
- 3 Container Orchestration with Kubernetes**
- 4 Server Provisioning with Terraform**
- 5 Configuration Management with Ansible**
- 6 Azure Dev Ops - CI/CD with Pipelines**
- 7 Jenkins - CI/CD with Pipelines**

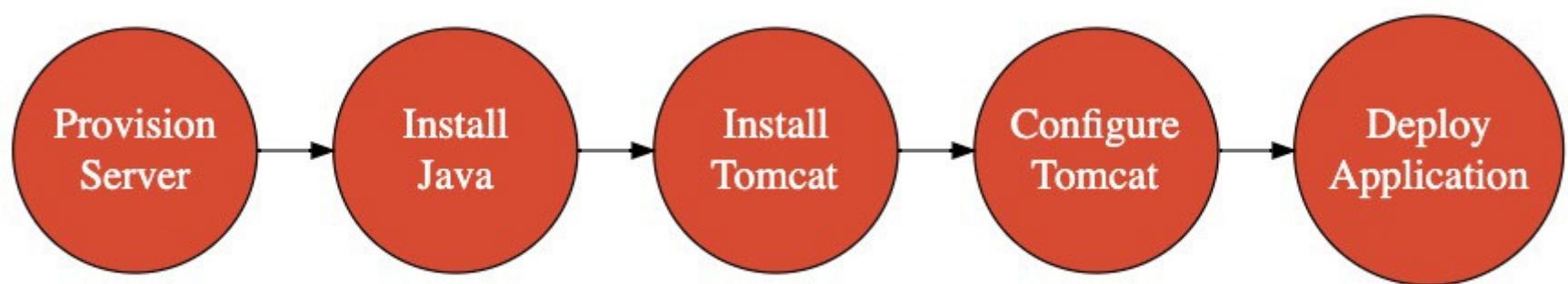


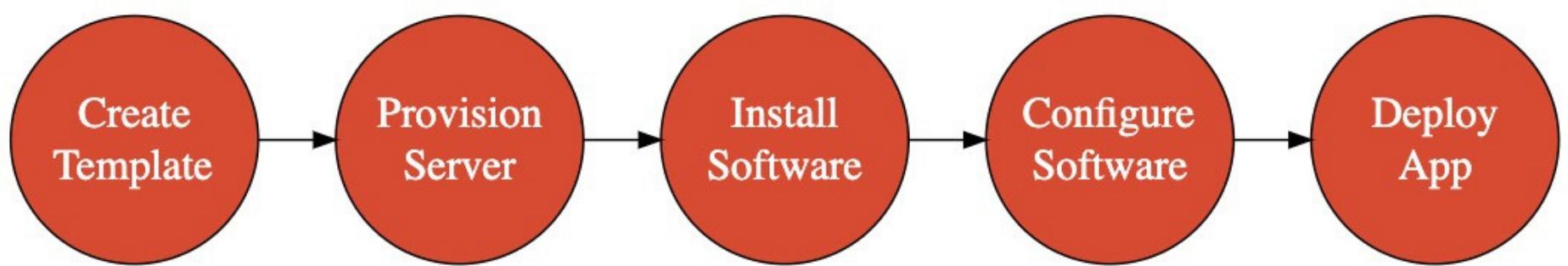


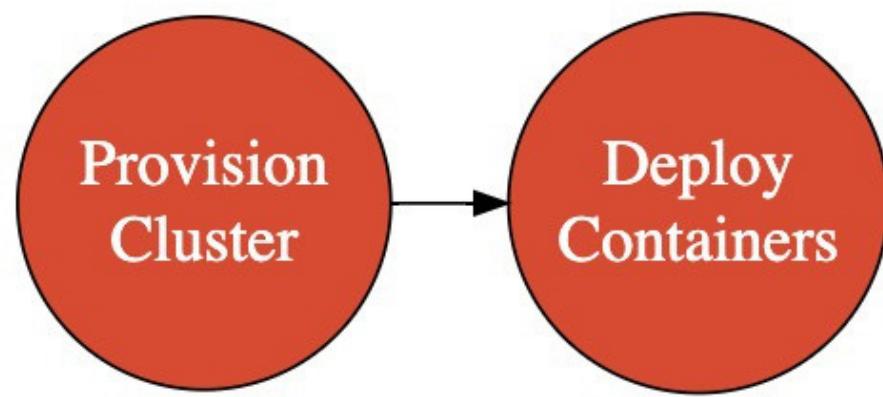














# Thank You