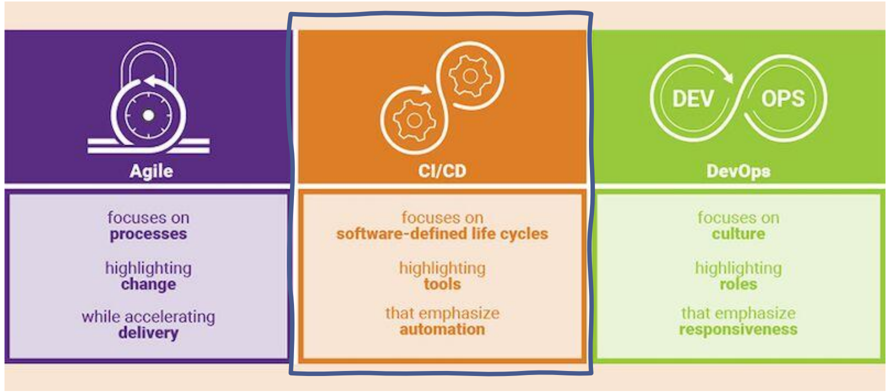
DevOps 101

Ratsameetip Wita

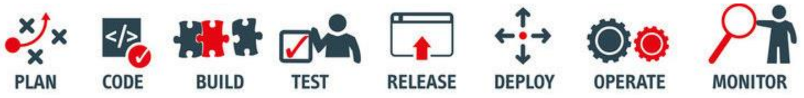
Agile, CI/CD and DevOps

↗continuous ↗continuous ↗deployment

↳integration



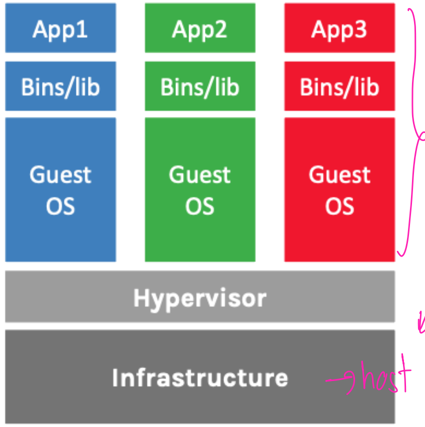
Agile	CI/CD	DevOps
focuses on **processes**	focuses on **software-defined life cycles**	focuses on **culture**
highlighting **change**	highlighting **tools**	highlighting **roles**
while accelerating **delivery**	that emphasize **automation**	that emphasize **responsiveness**



VM vs Docker

virtual machines

- virtual machines have their own virtual hardware: CPUs, memory, hard drives, etc.
- you need a hypervisor that manages different virtual machines on server
- hypervisor can run as many virtual machines as you wish
- operating system is called the "host" while those running in a virtual machine are called "guest“
- You can install a completely different operating system on this virtual machine



Machine Virtualization

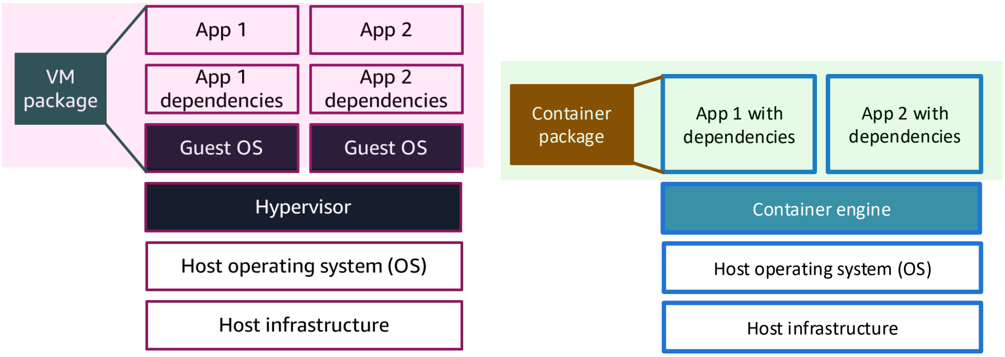
Container

จะไม่เหมือน VM หล่ะ เบากว่า

- a lightweight, efficient alternative to virtual machines.
- Consistency across multiple environments (dev, staging, production).Resource efficiency compared to traditional VMs.Faster start-up and execution times.

The benefit of containers

- **Virtual machines**
- **Containers**



VM ปกก. App dependencies, OS

Docker

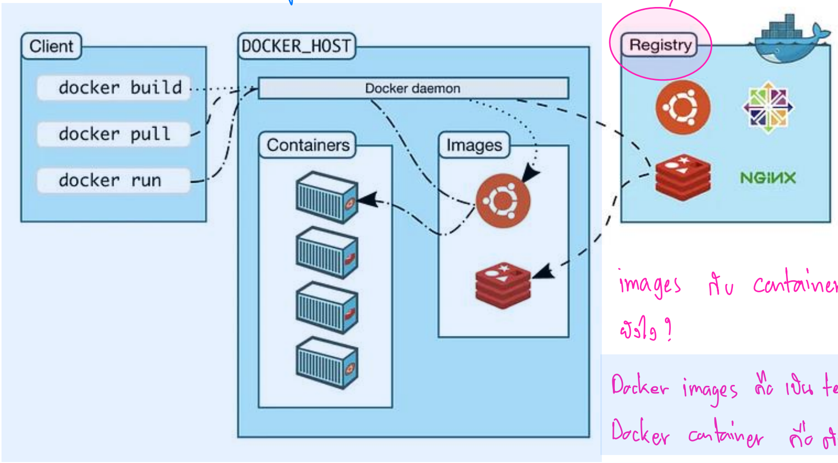
- **Docker is an open platform for developing, shipping, and running applications.**
- **Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.**
- **Docker is written in the Go programming language and takes advantage of several features of the Linux kernel to deliver its functionality.**

CI (ฝั่ง Github)

Docker Architecture

อยู่บนเครื่องเรา

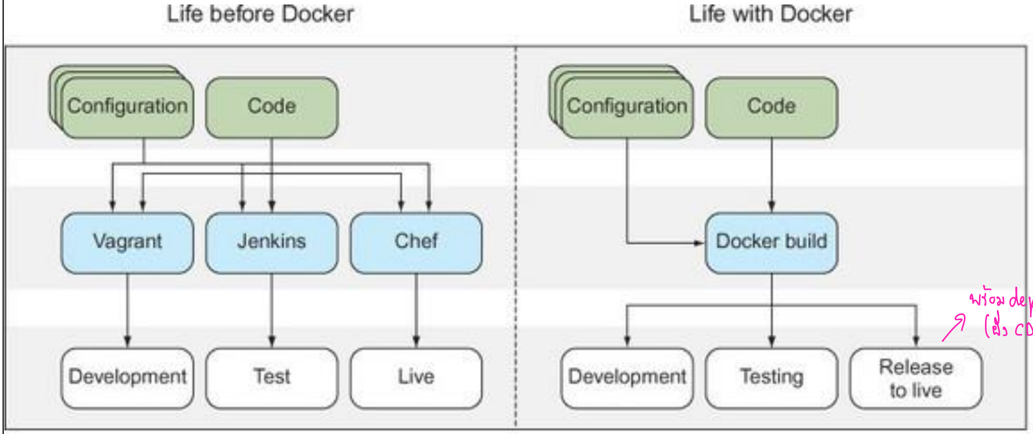
เอามาเก็บฝั่ง CD (Docker)



images กับ containers ต่างกันยังไง ?

Docker images คือ เป็น template เฉยๆ
Docker container คือ ตัวที่ run จริงๆ

https://docs.docker.com/get-started/overview/



พร้อม deploy (ฝั่ง CD → continuous delivery)

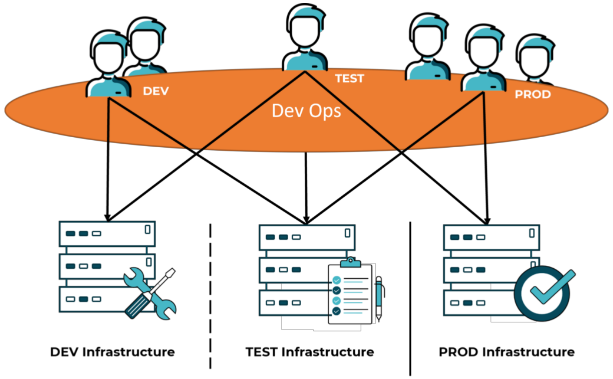
🟩 Inputs to the system requiring manual maintenance—fewer inputs here mean less of a maintenance burden.

🟦 Tools that use the inputs to create environments for software development

□ Stages of software development requiring an environment to run in

DevOps strategies

- Developers and admins are now talking more.
- They are a team and ideally understand which requirements should be considered beyond their own.

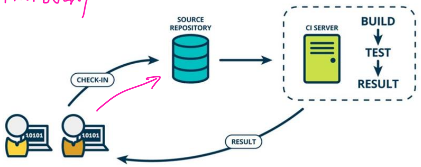


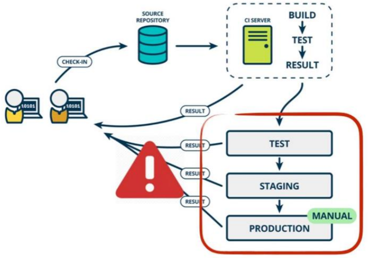
※ smoke test (?)

CI/CD

เอาโค้ดแต่ละส่วนมารวม แล้วเทสบ่อยๆ

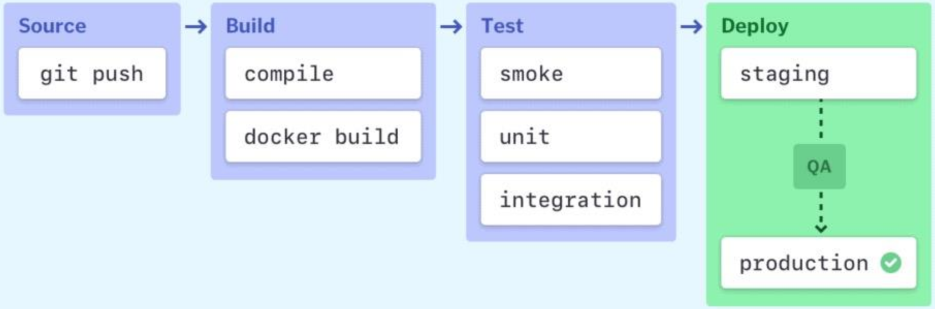
- **Continuous Integration (CI):** Practice of merging code changes frequently.
- **Continuous Deployment/Delivery (CD):** Automating the release of new code.
- Reduces integration problems.
- Allows faster, more reliable deployment.





Standalone : program ที่อยู่ในเครื่องโดยไม่ต่อใช้ internet เช่น calculator, word

CI/CD Pipeline



https://semaphoreci.com/blog/cicd-pipeline
https://softjourn.com/insights/ci-cd-pipeline-explaine

(CI)

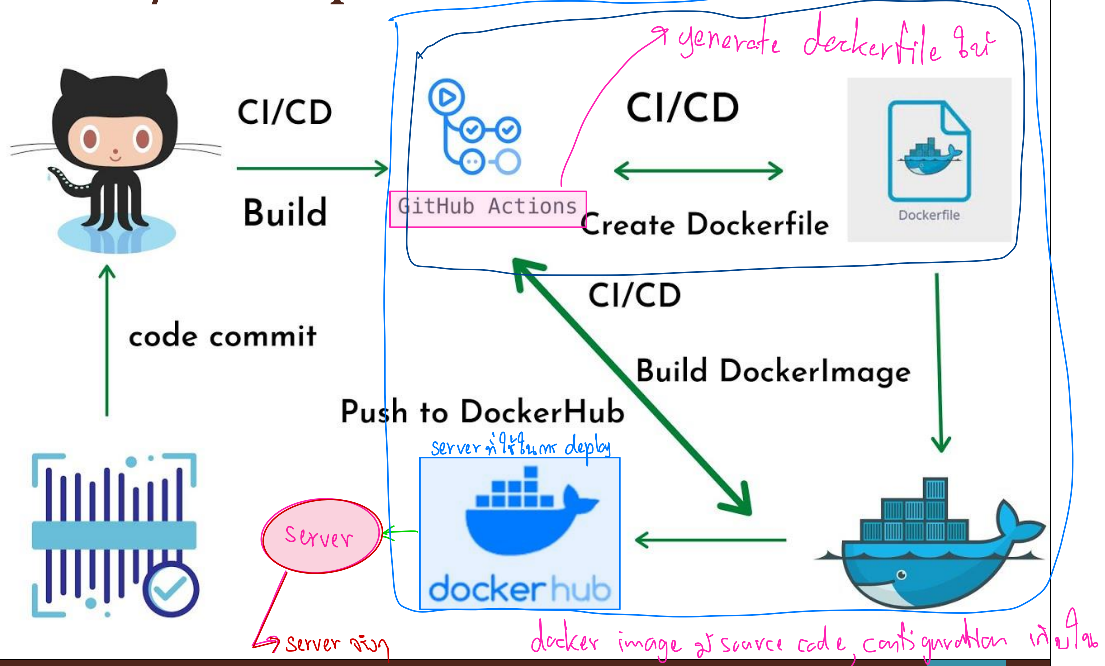
Github Actions

- Basic Concept:
  - Integrated CI/CD service within GitHub.
  - Automates workflows using YAML files.
  - Triggered by events like commits, pull requests, and releases.
- Key Features:
  - Event-driven workflows.
  - Customizable with reusable actions.
  - Deep integration with GitHub.

Github Actions

- **Benefits:**
  - **Easy to use with simple setup.**
  - **Flexible for various languages and environments.**
  - **Scalable to meet project needs.**
  - **Cost-effective with free tier options.**
  - **Enhances collaboration with GitHub's ecosystem.**

CI/CD Pipeline tools



CI/CD

Build

code commit

GitHub Actions

จ generate dockerfile ให้

CI/CD

Create Dockerfile

Dockerfile

CI/CD

Build DockerImage

Push to DockerHub

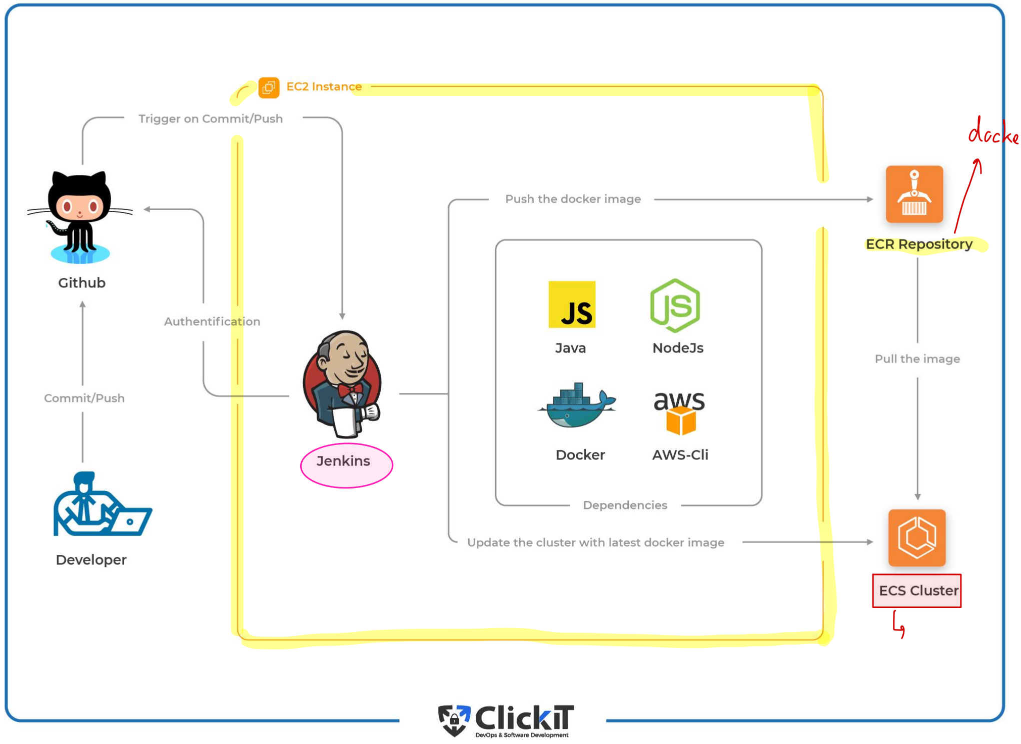
server ที่ใช้ในการ deploy

docker hub

Server

server จริง

docker image มี source code, configuration เก็บไว้ใน dockerhub



EC2 Instance

Trigger on Commit/Push

Github

Authentification

Commit/Push

Developer

Jenkins

Push the docker image

JS

Java

NodeJs

Docker

aws

AWS-Cli

Dependencies

Update the cluster with latest docker image

dockerhub

ECR Repository

Pull the image

ECS Cluster

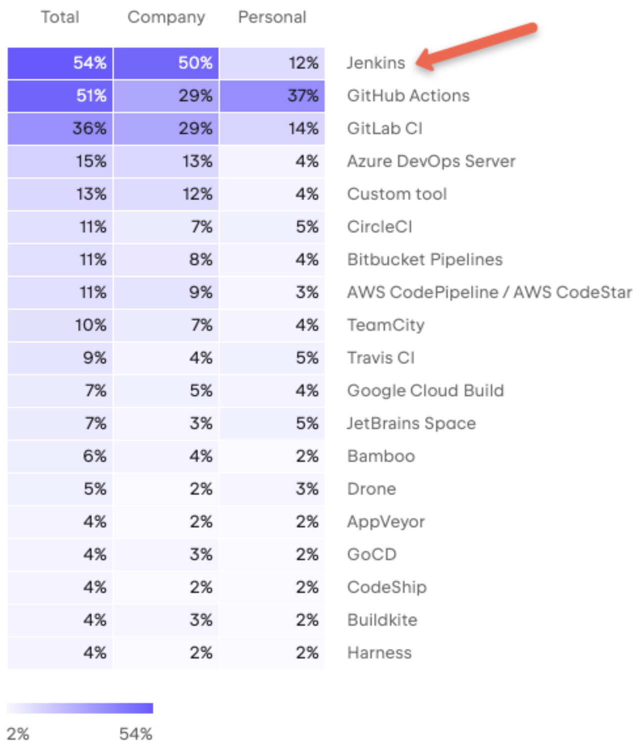
ClickIT

DevOps & Software Development

https://www.clickittech.com/devops/ci-cd-docker/

Github Actions VS Jenkins

- Hosting:
  - Jenkins is self-hosted, requiring server setup and maintenance;
  - GitHub Actions is cloud-hosted by GitHub.
- Integration:
  - GitHub Actions is natively integrated with the GitHub ecosystem;
  - Jenkins integrates with various version control systems and services.
- Plugins and Community:
  - Jenkins has a larger plugin ecosystem
  - GitHub Actions offers reusable actions within GitHub.
- Ease of Use:
  - GitHub Actions is simpler for GitHub-centric workflows
  - Jenkins offers more flexibility and control for diverse environments.



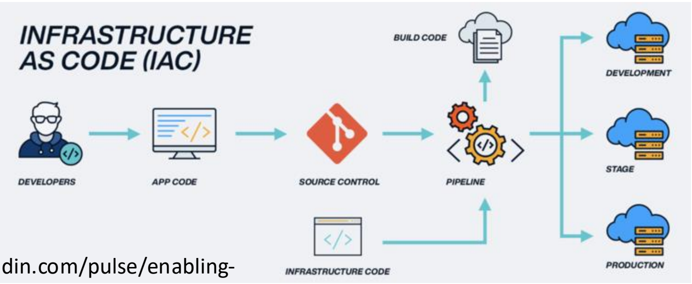
Total	Company	Personal	
54%	50%	12%	Jenkins
51%	29%	37%	GitHub Actions
36%	29%	14%	GitLab CI
15%	13%	4%	Azure DevOps Server
13%	12%	4%	Custom tool
11%	7%	5%	CircleCI
11%	8%	4%	Bitbucket Pipelines
11%	9%	3%	AWS CodePipeline / AWS CodeStar
10%	7%	4%	TeamCity
9%	4%	5%	Travis CI
7%	5%	4%	Google Cloud Build
7%	3%	5%	JetBrains Space
6%	4%	2%	Bamboo
5%	2%	3%	Drone
4%	2%	2%	AppVeyor
4%	3%	2%	GoCD
4%	2%	2%	CodeShip
4%	3%	2%	Buildkite
4%	2%	2%	Harness

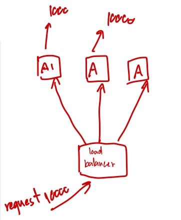
https://www.jetbrains.com/lp/devecosystem-2023/team-tools/

Infrastructure as Code (IaC)

- IaC is a key DevOps practice involving **managing and provisioning infrastructure** using **machine-readable definition** files instead of manual configuration.

" load balancer " เป็นตัว manage ระบบ



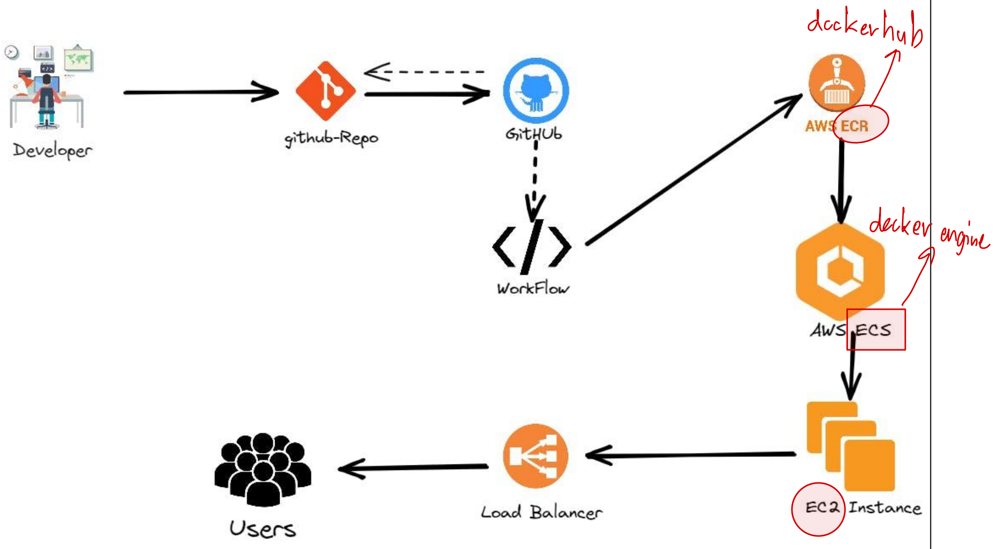


https://www.linkedin.com/pulse/enabling-infrastructure-code-iac-cicd-key-benefits-customers-bilal/

Infrastructure as Code (IaC)

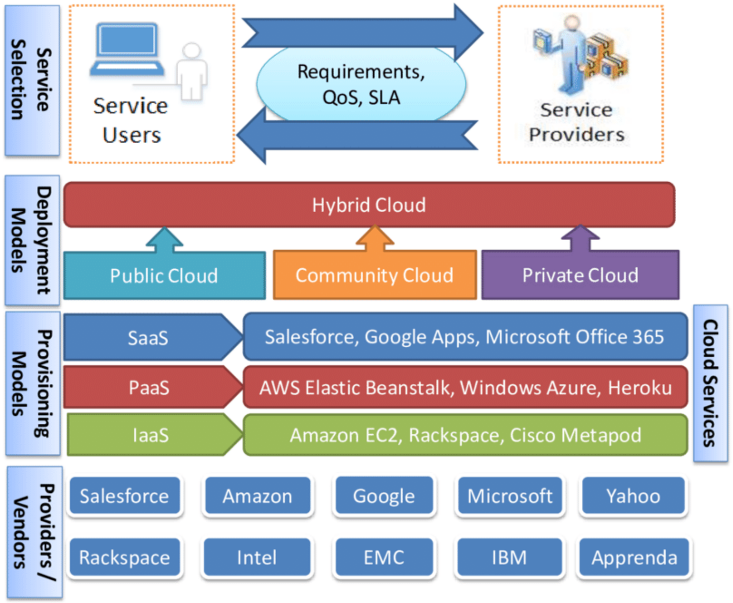
- benefits of IaC: Speed, consistency, minimization of human error, scalability.
- IaC is aiding in seamless development, testing, and deployment workflows in CI/CD pipelines.
- IaC can be implemented using tools
  - Terraform: Multi-cloud, highly flexible.
  - CloudFormation: Deep AWS integration.
  - Ansible/Puppet/Chef: Good for configuration management.

CI/CD with IaC



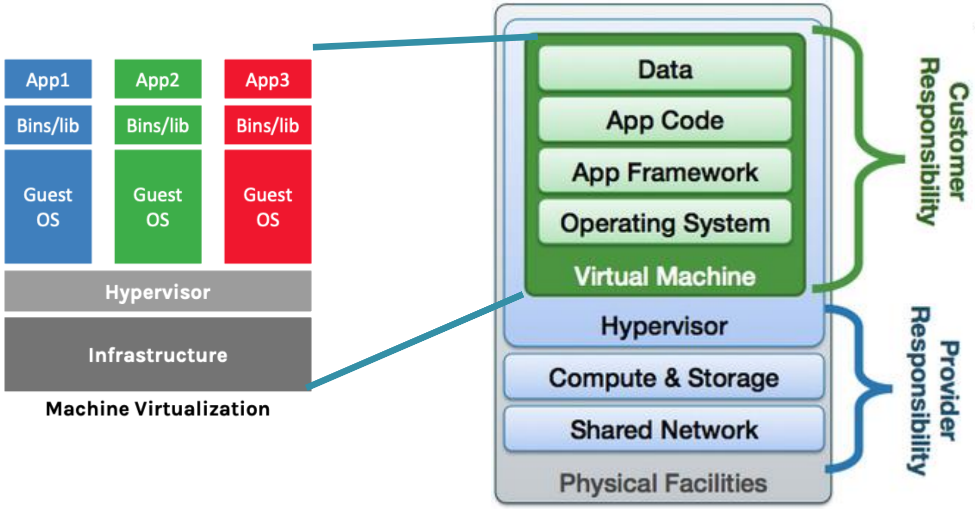
https://blog.kubesimplify.com/cicd-pipeline-github-actions-with-aws-ecs

Cloud Infrastructure

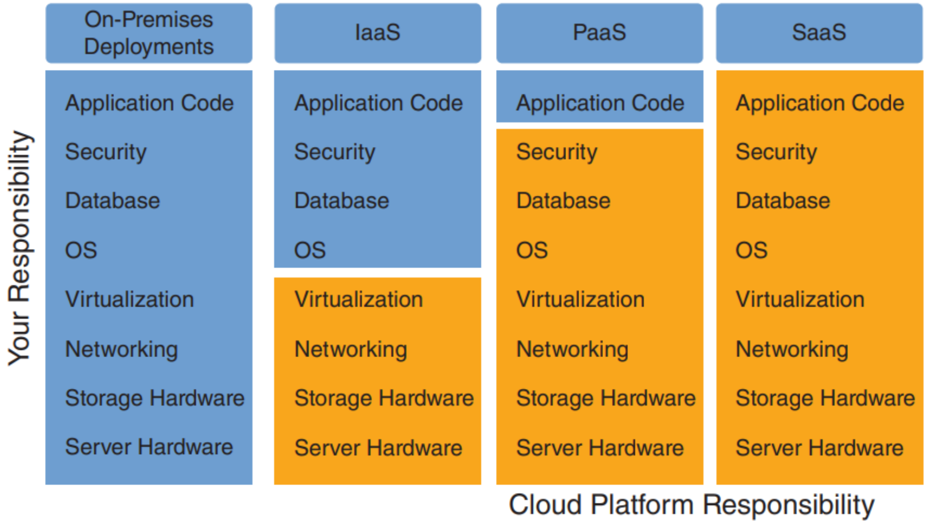


https://www.researchgate.net/publication/303393274_Trends_and_Directions_in_Cloud_Service_Selection

Cloud Architecture



Shared Responsibility Model



Your Responsibility

On-Premises Deployments	IaaS	PaaS	SaaS
Application Code	Application Code	Application Code	Application Code
Security	Security	Security	Security
Database	Database	Database	Database
OS	OS	OS	OS
Virtualization	Virtualization	Virtualization	Virtualization
Networking	Networking	Networking	Networking
Storage Hardware	Storage Hardware	Storage Hardware	Storage Hardware
Server Hardware	Server Hardware	Server Hardware	Server Hardware

Cloud Platform Responsibility

https://www.testpreptraining.com/tutorial/aws-cloud-practitioner/define-the-aws-shared-responsibility-model/

Amazon ECS

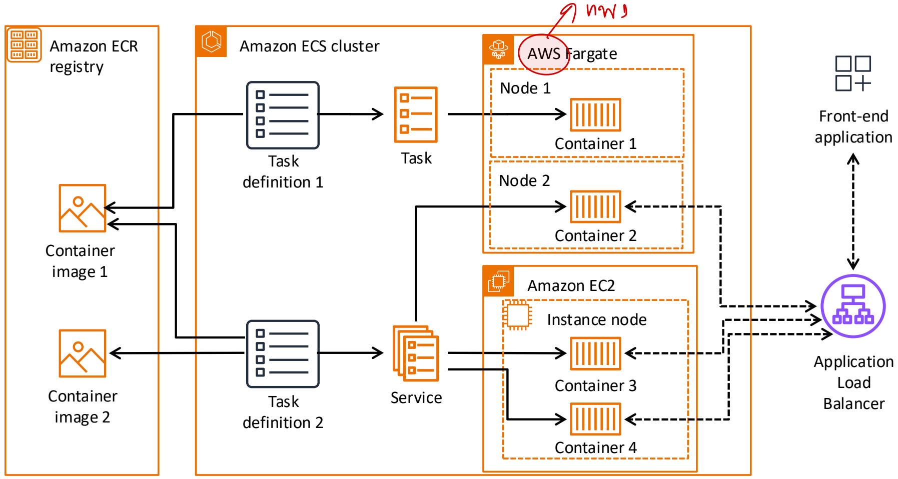
docker engine ที่อยู่บน cloud

- **/AWS ECS - Elastic Container Service**
- **Amazon’s container orchestration service.**
- **Supports Docker containers.**
- **Two launch types:**
  - **EC2 and Fargate.**
- **Benefits:**
  - **Managed service, scales automatically.**
  - **Integrates with other AWS services (e.g., IAM, VPC).**

AWS container services

Registry	Orchestration	Compute options
Amazon Elastic Container Registry (Amazon ECR)	Amazon Elastic Container Service (Amazon ECS)	Amazon Elastic Compute Cloud (Amazon EC2)
	Amazon Elastic Kubernetes Service (Amazon EKS)	AWS Fargate
		AWS Lambda

Deploying and invoking containers on Amazon ECS



แพง

Amazon ECR registry

Container image 1

Container image 2

Amazon ECS cluster

Task definition 1

Task

Task definition 2

Service

AWS Fargate

Node 1

Container 1

Node 2

Container 2

Amazon EC2

Instance node

Container 3

Container 4

Front-end application

Application Load Balancer

ECS Setup and Infrastructure Understanding Workshop

- Day 1: 14th Sept - (Full Day) --- 361 Mandatory Workshop
- Morning Session (9:00 AM - 12:30 PM)
- Workshop 1: Setting Up Infrastructure with AWS CDK
  - Hands-On: Use AWS CDK to create a simple ECS Cluster.
  - Create a VPC, Subnets, and Security Groups using CDK.
  - Deploy an EC2 instance and connect it to the ECS Cluster.
- Afternoon Session (1:30 PM - 5:00 PM)
  - Deep Dive into ECS Components
  - Load Balancer and Target Groups: Setting up using AWS CDK.
  - Auto Scaling Groups and Launch Templates: Configuring with AWS CDK.
- Workshop 2: Deploying an Application on ECS Using AWS CDK
  - Hands-On: Deploy a sample application using AWS CDK.
  - Create ECS Task Definitions and Services.
  - Set up Auto Scaling policies and integrate with Load Balancers.

CI – Github Action/CD - Jenkins

- Day 2: 15th Sept - Continuous Integration with GitHub Actions 9:00 AM - 12:30 PM
- Workshop 3: Integrating CI with AWS CDK
  - Hands-On: Create a GitHub Actions workflow that deploys the AWS CDK stack.
  - Automate the deployment process for ECS using GitHub Actions.
  - Run tests and linting before deployment.
- Lab Session
  - Extend the CI pipeline to include additional features like notifications or automated rollback.
- Continuous Deployment with Jenkins (1:30 PM - 3:30 PM)
- Workshop 4: Continuous Deployment with Jenkins and AWS CDK
  - Hands-On: Set up a Jenkins pipeline to deploy AWS CDK stacks.
  - Integrate Jenkins with GitHub Actions and ECS.
  - Implement a rollback mechanism in the Jenkins pipeline.
- Lab Session
  - Customize the Jenkins pipeline to manage different environments (staging, production).
- Q&A and Wrap-up. Afternoon Session (3:30 PM - 4:30 PM)
  - Open Q&A: Address questions related to AWS CDK, ECS, CI/CD, and other course content.
  - Recap and Discussion: Review key concepts and ensure practical understanding.
  - Feedback Session: Collect feedback on the workshops and overall course structure.