Cloud Native Development

What is Cloud Native Development?

Cloud Native Development is the process of building and running applications that exploit the advantages of cloud computing delivery models. It emphasizes scalability, resilience, and agility using tools like containers, microservices, and CI/CD.

Core Principles

- Microservices
- Containers
- Dynamic Orchestration
- DevOps & CI/CD
- Infrastructure as Code (IaC)
- Resilient Systems
- Observability

Cloud Native Application Lifecycle

- Plan
- Develop
- Build
- Test
- Release
- Operate
- Optimize

Best Practices

Cloud Native Development

Architecture:

- Use APIs, design for failure, stateless services

Development:

- Follow 12-Factor App, small services, feature flags

CI/CD:

- Automate tests, include security scans

Security:

- Secrets management, RBAC

Observability:

- Centralized logs, dashboards, alerts

Example Stack

Cloud: AWS, GCP, Azure

Orchestration: Kubernetes

Containers: Docker

CI/CD: GitHub Actions

IaC: Terraform

Monitoring: Prometheus, Grafana

Logging: ELK

Languages: Go, Node.js, Python

Real World Example

E-commerce Platform:

Frontend: React via CloudFront

Backend: Microservices in Node.js on Kubernetes

Cloud Native Development

Database: PostgreSQL (RDS)

CI/CD: GitHub Actions with ArgoCD

Monitoring: Prometheus + Grafana

Logs: ELK Stack

IaC: Terraform

Cloud Native Development

Cloud Native Application Lifecycle Diagram