Cloud Native Development and Deployment

Cloud native development is an approach to building and running applications that exploits the advantages of the cloud computing delivery model. Cloud native is about how applications are created and deployed, not where.

Cloud Native Lifecycle:

1. Plan:
   - Activities: Define business needs, choose technologies
   - Examples: JIRA for planning, Confluence for documentation

2. Develop:
   - Activities: Writing microservices-based code using cloud native principles
   - Examples: Using Spring Boot, Node.js, or Go

3. Build:
   - Activities: Compile code, create container images
   - Examples: Maven, Gradle, Docker

4. Test:
   - Activities: Unit testing, integration testing, security scanning
   - Examples: JUnit, Selenium, SonarQube

5. Release:
   - Activities: Versioning and storing artifacts in repositories
   - Examples: GitHub Releases, Nexus, Artifactory

6. Deploy:

- Activities: Deploying to cloud platforms using CI/CD pipelines
- Examples: Jenkins, GitLab CI/CD, ArgoCD

7. Operate:

- Activities: Ensure the system is reliable and scalable
- Examples: Kubernetes for orchestration, Istio for service mesh

8. Monitor:

- Activities: Monitoring performance, errors, usage
- Examples: Prometheus, Grafana, ELK Stack

This approach enables faster development, scalability, resilience, and agility by leveraging microservices, containers, DevOps practices, and continuous delivery/deployment in a dynamic environment.