

AWS DevOps

—

Topics Covered

- **Introduction to DevOps**
- **VM Hypervisor**
- **AWS CLI**
- **Shell Script**
- **Version Control (Git)**
- **SCM Tools (Github or Bitbucket)**
- **Configuration Management (Ansible)**
- **Infrastructure as Code (IaC) (Terraform)**
- **CI/CD (Jenkins or AWS pipeline)**
- **Containers and Orchestration (Docker and Kubernetes)**
- **Monitoring and Log Management**
- **Labs and Project**

Introduction to DevOps

Overview of DevOps:

- DevOps is a set of practices that combines software development (Dev) and IT operations (Ops) to enhance the efficiency and effectiveness of software delivery.
- Aims to shorten the development lifecycle and provide continuous delivery with high software quality.

Key Principles and Practices:

- **Collaboration:** Encourages close collaboration between development and operations teams.
- **Automation:** Automates repetitive tasks such as testing, deployment, and monitoring.
- **Continuous Integration (CI):** Integrates code changes frequently to detect issues early.
- **Continuous Delivery (CD):** Automates the release process to ensure code is always in a deployable state.
- **Monitoring:** Continuously monitors applications and infrastructure for performance and issues.

Benefits and Challenges:

- **Benefits:** Faster time-to-market, improved collaboration, higher quality, and better scalability.
- **Challenges:** Requires cultural change, tool integration, and managing complexity.

DevOps Lifecycle:

- **Plan:** Define requirements and plan features.
- **Develop:** Write and test code.
- **Build:** Compile code and prepare artifacts.
- **Release:** Deploy artifacts to production.
- **Deploy:** Roll out code to production environments.
- **Operate:** Monitor and manage applications in production.
- **Monitor:** Gather feedback and performance metrics.

What is SDLC

- **Software Development Life Cycle (SDLC):** A process used by software industry to design, develop, and test high-quality software.
- Provides a structured approach to software development.

Phases of SDLC:

1. **Planning:** Define the scope and purpose of the project.
2. **Requirements Analysis:** Gather and analyze user requirements.
3. **Design:** Create architectural and detailed design plans.
4. **Implementation (Coding):** Write and compile the source code.
5. **Testing:** Validate the software to ensure it meets requirements.
6. **Deployment:** Install and configure the software in the production environment.
7. **Maintenance:** Provide ongoing support and enhancements.

How DevOps Improves SDLC:

- **Enhanced Collaboration:** Breaks down silos between development and operations teams.
- **Continuous Integration and Continuous Delivery (CI/CD):** Enables frequent code integrations and automated deployments, reducing time-to-market.
- **Automation:** Streamlines repetitive tasks like testing and deployment, improving efficiency.
- **Continuous Monitoring:** Provides real-time feedback and insights, ensuring better quality and performance.
- **Agility:** Increases the flexibility to adapt to changes and address issues quickly.

VM Hypervisor

What is Virtualization:

- **Virtualization:** The process of creating a virtual version of something, such as hardware platforms, storage devices, and network resources.
- Allows multiple operating systems and applications to run on a single physical machine, improving resource utilization and flexibility.

Definition and Types of Hypervisors:

- **Hypervisor:** A virtual machine manager that allows multiple virtual machines (VMs) to run on a single physical host.
- **Type 1 Hypervisor (Bare-Metal):** Runs directly on the hardware (e.g., VMware ESXi, Microsoft Hyper-V).
- **Type 2 Hypervisor (Hosted):** Runs on top of an operating system (e.g., VMware Workstation, Oracle VirtualBox).

Role of Hypervisors in Virtualization

- Abstracts and allocates physical resources to VMs.
- Manages VM lifecycle and resource allocation.

Key Hypervisor Technologies:

- **VMware:** Provides enterprise-grade virtualization solutions.
- **Hyper-V:** Microsoft's virtualization platform.
- **KVM (Kernel-based Virtual Machine):** Open-source virtualization for Linux.