

# System Provisioning & Configuration Management.

Module # 01

## Introduction

Copyright © 2023. All rights reserved. release 2.0.0

# Module Learning Objectives

Upon successful conclusion of this module, you will be able to:

- Understand the historical evolution of system provisioning and its impact on modern practices.
- Assess pros and cons of traditional and modern system provisioning approaches.
- Apply automation tools to enhance provisioning efficiency.
- Analyze real-world DevOps implementations in software development.
- Evaluate benefits of proficient provisioning, configuration management, and Infrastructure as Code (IaC) adoption. Develop strategic plans for implementing DevOps in diverse real-world applications.



# Module Topics

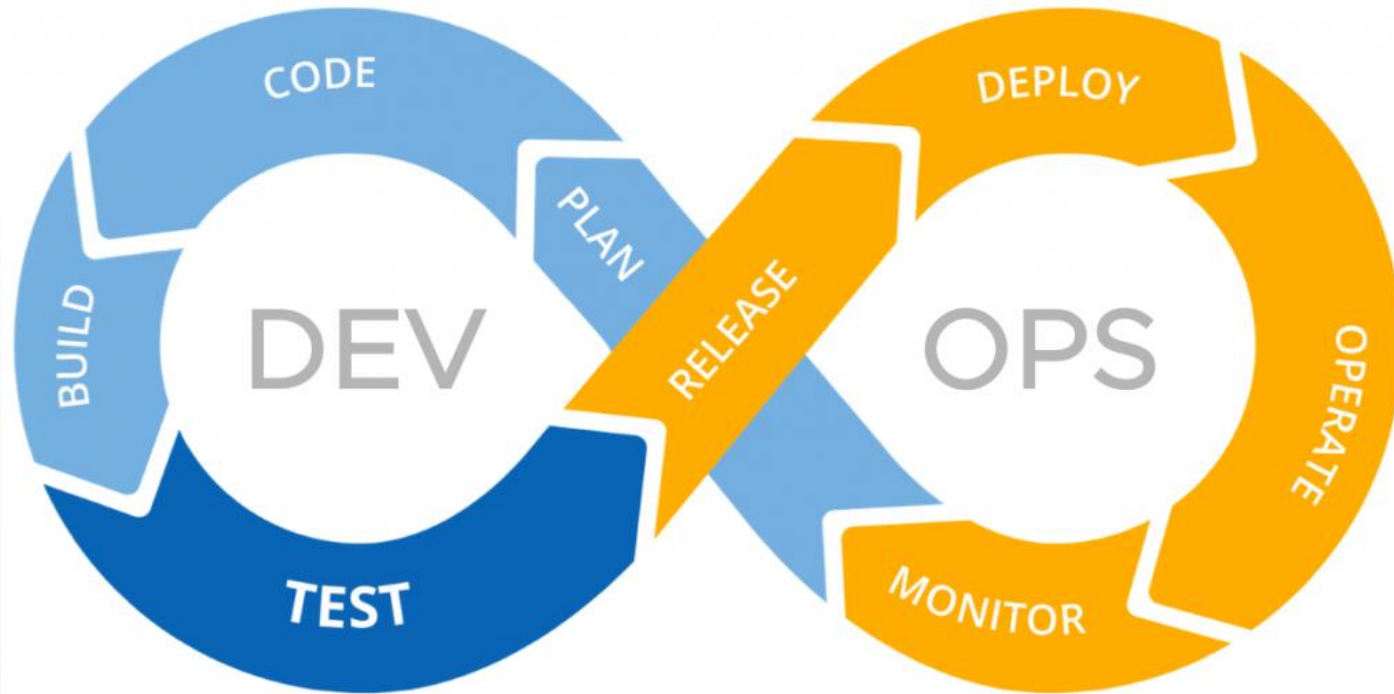
The following topics that will be covered in the module:

- Definition and importance of DevOps
- Key concepts: provisioning, configuration management, Infrastructure as Code (IaC)
- Evolution of System Provisioning
- Traditional vs. modern approaches
- Role of automation in provisioning.



# Definition of DevOps

DevOps, a merger of Development and Operations, is a set of practices to automate and enhance the relationship between software development and IT operations teams.

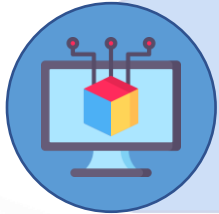


# Importance of DevOps

The significance of Devops is



To increase an organization's ability to deliver applications and services



To advocates for frequent software releases



A/B testing: Personalized messaging boosts conversions



DevOps is crucial in the tech landscape.



# Key Concepts of System Provisioning and Configuration Management



**Provisioning:** The process of setting up and preparing the necessary resources for software applications.



**Configuration Management:** Ensuring that the resources function seamlessly, typically involving automation and versioning.



**Infrastructure as Code (IaC):** The practice of defining and managing infrastructure in a descriptive model, using the same versioning and automation approaches as software.

# Evolution of System Provisioning

**Before :** Previously, system provisioning was a manual process, requiring IT personnel to set up each system individually. It was time-consuming and prone to human errors.



**Transition Phase :** The transition phase introduced some level of automation, such as using scripts for provisioning. This reduced the time required but still had limitations in terms of scalability and flexibility.



**Current State :** Today, system provisioning is highly automated using cloud technologies. Infrastructure as Code (IaC) allows for the instant, scalable, and error-free provisioning of systems.



Current State - Cloud Computing

# Traditional vs. Modern Approaches

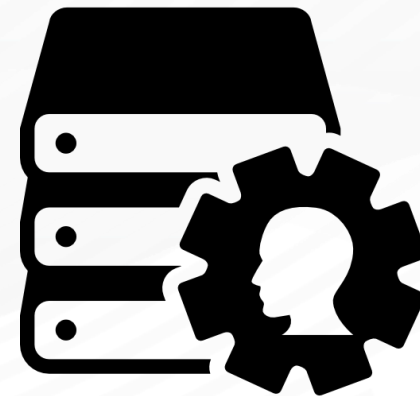
Let's compare the traditional approaches with the modern ones

## Modern Approaches

- Automation of processes
- Streamlined and optimized tasks
- Efficiency in provisioning and configuration

## Traditional Approaches

- Manual and time-consuming processes
- Manual setup and configuration
- Lack of optimization and streamlining



VS





# Role of Automation in Provisioning



**Defining Automation**



**Ensuring Consistency**



**Reducing Human Error**



**Enabling Scalability**



**Speeding Up Processes.**



**Adapting to Change**

# Role of Automation in Provisioning

01

**Defining Automation:** Tools automate tasks, streamline system setup, and accelerate infrastructure configuration.



02

**Reducing Human Error:** Automation minimizes human error, ensuring consistent and accurate provisioning processes.



03

**Speeding Up Processes:** Automation swiftly accelerates provisioning, transforming hours of manual work into minutes.



# Role of Automation in Provisioning

04

**Ensuring Consistency:** Automation maintains consistent configurations, minimizing deployment issues across development, testing, and production.



05

**Enabling Scalability:** Automation empowers seamless and efficient scalability, from small projects to increased demand



06

**Adapting to Change:** Automation enables swift adjustments to configurations, crucial for evolving requirements or rapid feature deployment.

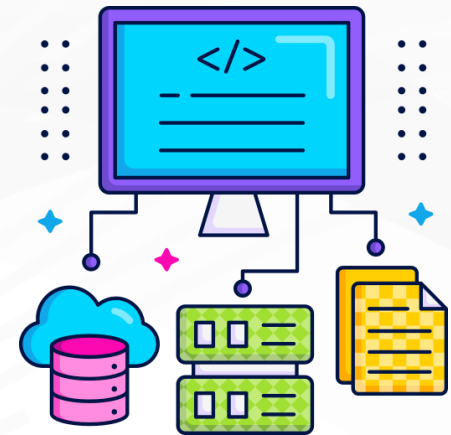
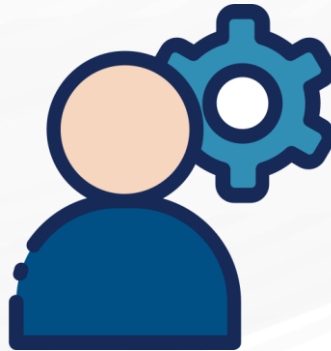


# Unlocking the Benefits: Provisioning, Configuration Management, IaC

Provisioning accelerates resource allocation and system setup, streamlining the process of getting the necessary resources for software applications.

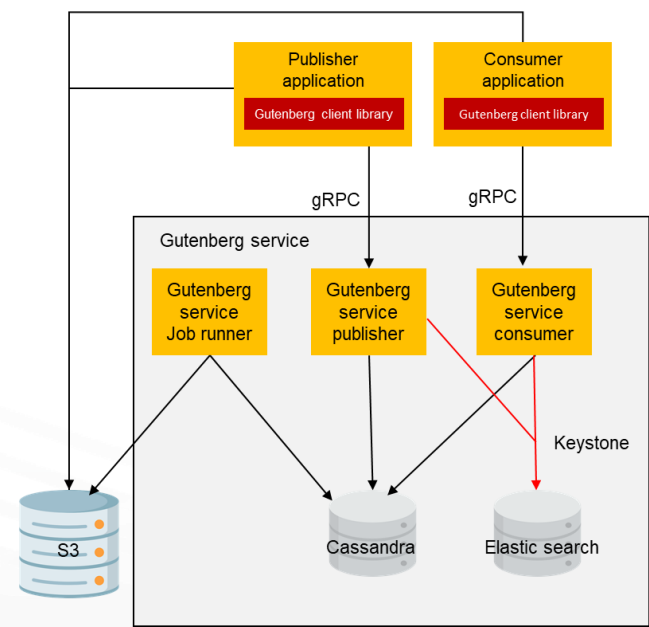
Configuration Management enhances system stability by efficiently managing changes and ensuring seamless functionality of resources.

Infrastructure as Code (IaC) provides a scalable and repeatable approach to infrastructure deployment, automating processes that traditionally required manual intervention.



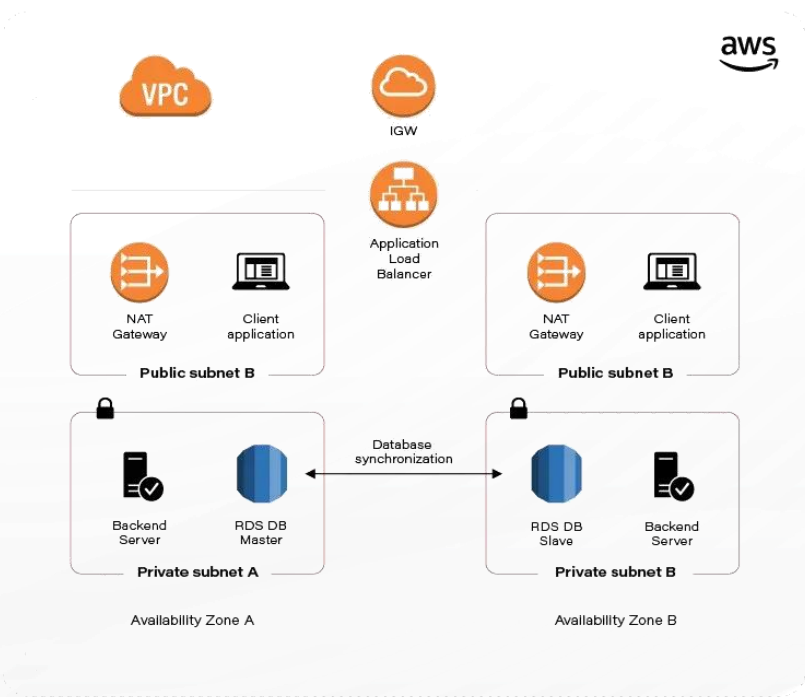
# Real-world Application Examples

## Netflix's Success



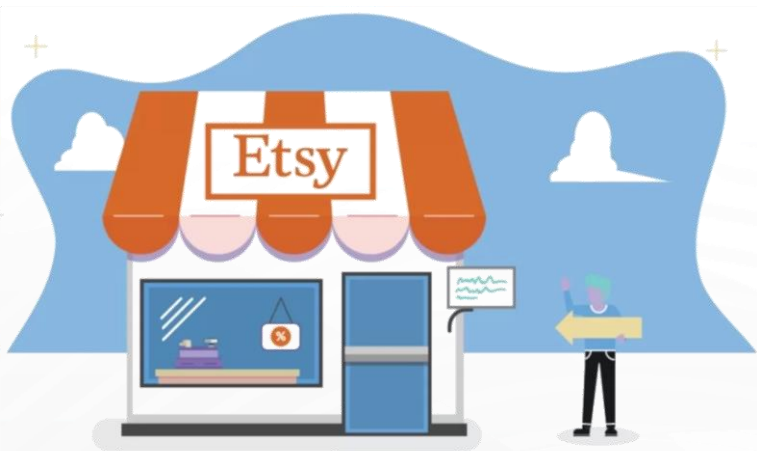
Netflix used microservices for scalability, resilience. Dynamic resource scaling ensured seamless streaming for millions.

## AWS and IaC



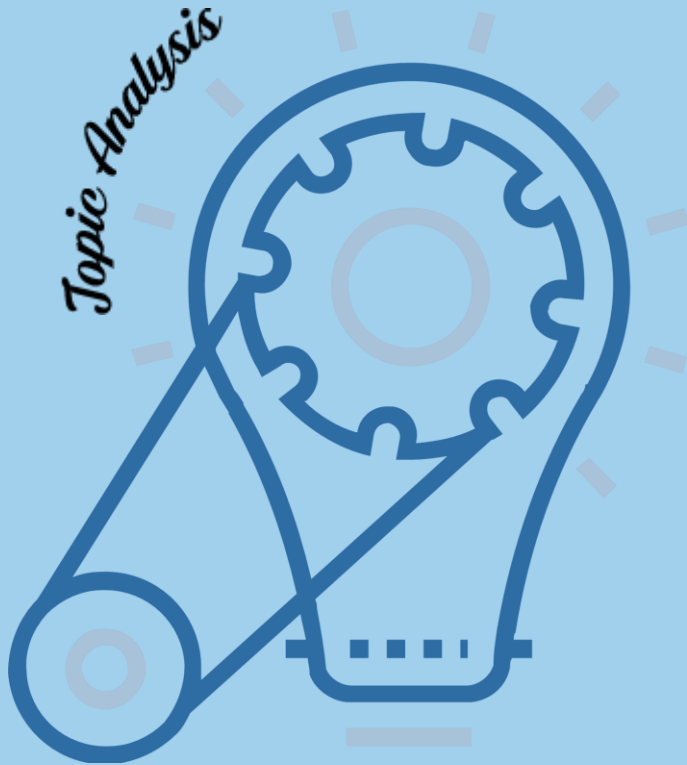
AWS demonstrates the power of Infrastructure as Code (IaC) for managing infrastructure via code, allowing rapid deployment.

## Etsy Marketplace.



Etsy, an e-commerce platform, masters continuous delivery, a DevOps core. Strong automation and configuration management ensure frequent, reliable releases.

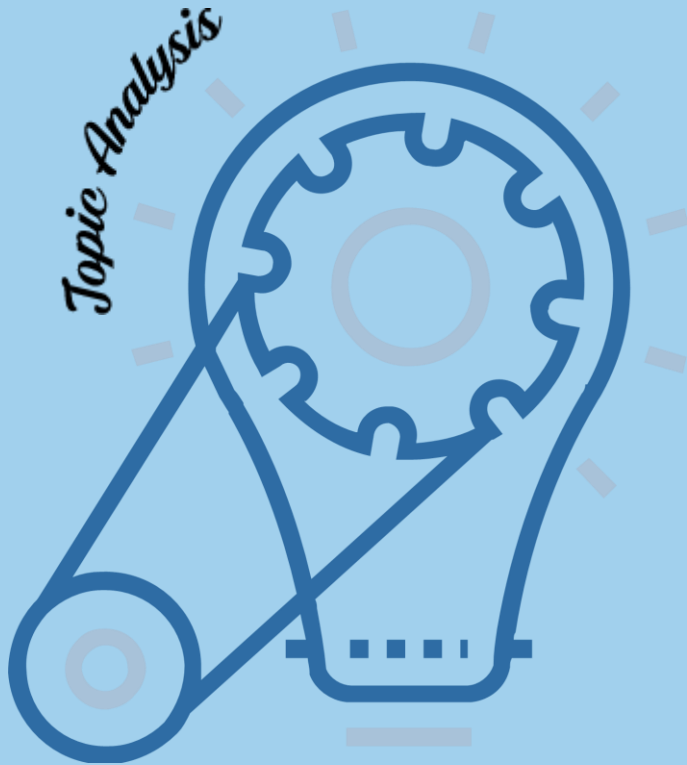
# What did you grasp?



1. What is the primary goal of Infrastructure as Code (IaC) in system provisioning and configuration management?
  - A. To manually set up hardware resources
  - B. To accelerate resource allocation through code
  - C. To increase human error in provisioning
  - D. To depend solely on physical hardware



# What did you grasp?



2. Which historical development marked a significant turning point in the evolution of system provisioning?
- A. Introduction of cloud computing**
  - B. Adoption of microservices architecture**
  - C. Emergence of virtualization**
  - D. Implementation of Infrastructure as Code (IaC)**

# In a nutshell, we learnt:



- We've defined DevOps and its importance in the tech landscape.
- Compared traditional manual processes with today's streamlined and optimized tasks.
- Explored key concepts including Provisioning, Configuration Management, and Infrastructure as Code (IaC).
- Highlighted the role of automation in DevOps, reducing human error and speeding up processes.
- Discussed the evolution of System Provisioning, from historical approaches to modern practices.
- Examined the practical implications of DevOps, such as faster development cycles and improved collaboration.

# End of Module

Next Module 2:

