**PERFORM SERVER CONFIGURATION**

A **server** is a powerful computer or a software application that provides resources, data, or services to other devices, known as clients, over a network. Servers play a crucial role in handling tasks like hosting websites, storing data, running applications, or processing complex computations. Here’s a breakdown of the essentials around servers:

**1. Types of Servers**

* **Web Server**: Hosts websites and delivers web content to users. Common software includes Apache.
* **Database Server**: Stores, retrieves, and manages data for applications. Examples include MySQL, and MongoDB.
* **Application Server**: Hosts and manages applications for client computers, often serving business logic to web applications.
* **File Server**: Manages and provides centralized file storage that can be accessed by other computers on a network.
* **Mail Server**: Manages the sending, receiving, and storage of emails.
* **Game Server**: Hosts online games, allowing players to interact in real-time.

**Examples of Linux**

1. Ubuntu

2. CentOS

4. Fedora

6. Kali Linux

**Development Operations (DevOps)**

**Development Operations (DevOps)** is a set of practices and tools designed to streamline and improve collaboration between software development (Dev) and IT operations (Ops) teams.

***Core Principles of DevOps/ features***

**Collaboration**: Promotes close communication between development and operations teams to break down silos.

**Automation**: Automates repetitive tasks, such as testing, integration, and deployment, to speed up processes and reduce errors.

**Continuous Integration and Continuous Deployment (CI/CD)**: Focuses on automating code integration, testing, and deployment, allowing for rapid and reliable releases.

**Configuration Management**: Manages and maintains configuration files and settings for different environments (development, testing, production).

**Benefits of DevOps**

* **Faster Delivery**: Automated testing and deployment reduce the time from code development to production.
* **Improved Quality**: Continuous testing and feedback allow teams to detect and fix issues early.
* **Reliability**: Continuous monitoring ensures higher uptime and quicker response to incidents.
* **Efficiency**: Automating workflows and managing resources effectively improves productivity.
* Reduce transportation cost ,Enhance collaboration

**DevOps Lifecycle**

DevOps is a continuous process, often visualized as an infinite loop encompassing the following stages:

* **Plan**: Teams plan features, fixes, and updates collaboratively.
* **Code**: Developers write code and manage it with version control tools like **Git**: Tracks code changes and manages versions, commonly used with platforms like GitHub.
* **Build**: Code is compiled, created using tools like Jenkins.
* **Test**: Automated tests are run to validate the code.
* **Release**: Validated code is deployed to staging or production environments.
* **Deploy**: New code is released to the live production environment.
* **Operate**: The application is monitored to ensure stability and performance.
* **Monitor**: Steps to identify the risk of failure. Splunk tool can be used in monitoring, tracking the system.

**Node**: A single machine in a cluster that can be either a physical server or a VM.

**Infrastructure as Code (IaC)** is a way to manage and set up computer infrastructure (like servers, networks, and databases) by writing code instead of doing it manually. Imagine building your whole computer setup just by running a script—IaC lets you do that for all the infrastructure a company might need.

In short, IaC helps developers and IT teams manage infrastructure efficiently, reliably, and quickly by treating it like software.

**Infrastructure as a Service (IaaS)** is a cloud computing model where a provider gives you virtual access to basic IT infrastructure over the internet. With IaaS, you get resources like virtual servers, storage, and networks, which you can rent and use as needed.

**Example**: Instead of buying servers to host a website, you can use IaaS to rent servers on platforms like Amazon Web Services (AWS), Microsoft Azure, or Google Cloud.

**CI/CD**

Stands for **Continuous Integration and Continuous Deployment** (or Continuous Delivery), which are practices in software development aimed at improving the development process through automation and efficiency.

**1. Continuous Integration (CI)**

CI is the practice of frequently integrating code changes into a shared repository, usually multiple times a day.

**How It Works**:

When developers add new code, automated tests run to check that the new code works correctly with the existing codebase. This helps catch bugs early.

**Benefits**:

* Reduces integration problems.
* Encourages regular code commits, which keeps the codebase up-to-date.
* Helps maintain a stable application.

**2. Continuous Delivery (CD)**

Continuous Delivery is the practice of ensuring that code changes are automatically prepared for release to production. It extends CI by automating the release process.

**How It Works**: After code passes automated tests, it’s packaged and made ready for deployment. The deployment to production can still be done manually, but the process is streamlined and predictable.

**Benefits**:

* Faster and more reliable releases.
* Reduces the risk of deployment failures.
* Improves collaboration between development and operations teams.