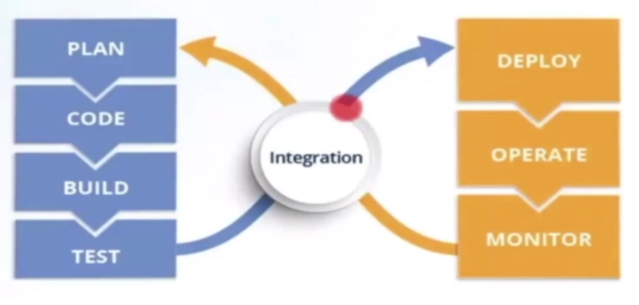
**DevOps Understanding**

1. **What we will learn:**

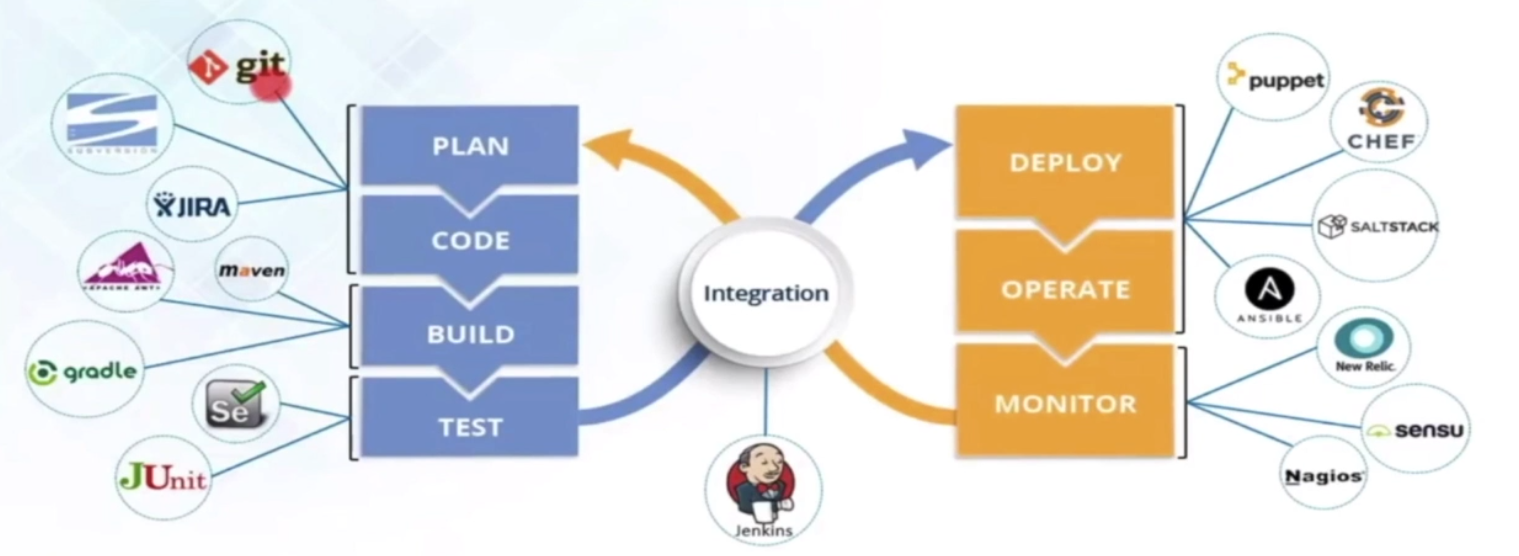
* VCS-GIT and Github
* Continuous Integration-Jenkins CI-CD Delivery Pipeline- Jenkins,
* Docker- Docker containers, Docker Image and Networking, Docker Compose, Docker Swarm Orchestration System
* Kubernetes- Kubernetes Cluster Administration on Kubernetes, HELM Package Chart Manger

1. **What is DevOps**

DevOps is a software development approach (like waterfall model) which involves Continuous Development, Continuous Testing, Continuous Integration, Continuous Monitoring, of the software throughout its development life Cycle

1. **DevOps Tools required**

**A diagram of a process

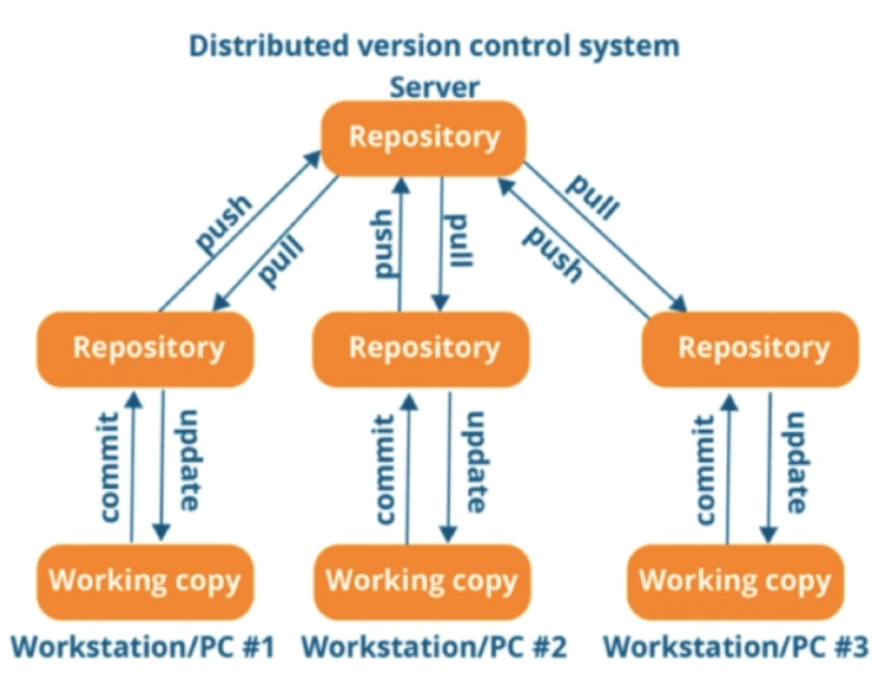
Description automatically generated**

1. **Continuous Development**

Continuous Development involves planning and coding of the software application’s functionality. The version of the project is decided during the planning phase.Code can be done in any language but maintain by version control system.

**Why code versioning is Important:**

Versions are maintained to hold a single source of application. Using centralized single source code , Operations can access the same code what they plan to release. Easy to rollout the faulty snippet of code.

1. **Continuous Testing**

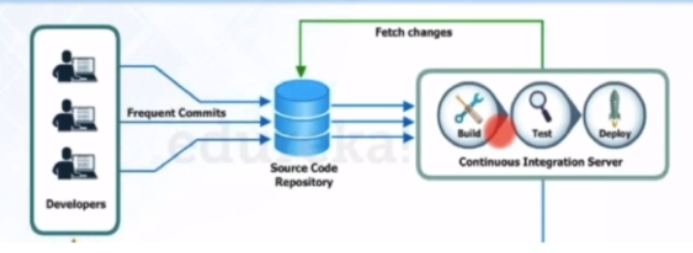
Staging Area

Continuous testing is the process of executing automated tests as part of the software delivery pipeline to provide immediate feedback on the quality of the code being developed. The goal is to detect and address issues as early as possible, reducing the risk of defects making it into production.

1. **Continuous Integration**

Continuous Integration is the process of Automated Build and Automated Tests. Continuous Integration helps to detect error quickly and locate them more easily. Continuous integration is not about resolving the bug but to find them more quickly.

**How it works:**

After every commit to the source code an auto build is triggered and then it is automatically deployed on the test server. If the test results shows that there is a bug in the code then the developers only have to check the last commit made to the source code. This increases the frequency of the new software releases.

Benefits:

No long and tense integrations. Increase visibility enabling greater communication. Catch issues early. Spend less time debugging and more time adding features. Reduce integration problems allowing you to deliver software more rapidly.

1. **Continuous Deployment**

It is related to continuous integration and refers to keeping your application deployable at any point. With the help of continuous deployment you can deploy the application latest version on test or production environment. It is the combination of two parts:

**Configuration management**

Configuration management in continuous deployment is about making sure that all parts of your software and its environment (like servers, networks, and databases) are set up correctly and consistently every time you deploy new code. Consists of 4 phases:

1. Establish and maintain Applications functional requirements and performance.
2. Releasing deployments to servers. For e.g. 1.8 version should be released to all the servers
3. Scheduling updates on all servers
4. Maintain configuration consistency on all server .All servers should have same specs, same os, same ram.

**Containerization.**

Containerization tool is a set of tools which will maintain the consistency across the environments. i.e. installing dependencies of an applications in a containers and send the container across all the servers.

It helps scheduling updates on all servers.

It maintains configuration consistency on all servers.

1. **Continuous Delivery**

It is the practice of keeping your codebase deployable at any point. Beyond making sure your application passes automated tests it must have all the configuration necessary to push it into production.

1. Continuous Monitoring

It monitors the entire system performance. Also monitors the product overall performance .Suppose your application is deployed on a machine which has configuration like 32 bit gb ram 512 hdd . It checks how much memory is being used , how the performance is low etc. We monitor the productions at different different stages