DevOps with AWS Course Documentation

Documentation Version 0.0.1

Document Prepared by

Sai Sashreek D

Document Prepared Date

11 - 12 - 2022

Course Contents

What is Devops?

The word Devops is a combination of the term's development and operations, meant to represent a collaborative or shared approach to the tasks performed by company's application development team and IT operation's Team.

Devops is a philosophy that promotes better communication and collaboration between these teams - and other teams in an organization.

It is a collaboration between Development and IT Operations to make software production and Deployment in an automated & repeated way.

It helps increases the organization's speed to deliver software applications and services.

It allows organizations to serve their customers better and compete more strongly in the market.

Why Devops used?

Devops allows Agile Development Teams to implement Continuous Integration and Continuous Delivery, which helps them launch products faster into the market. There are other important reasons to adopt to Devops:

Predictability - It offers a significantly lower failure rate of new releases.

Reproductivity - Version everything so that earlier version can be restored anytime.

Maintainability - Effortlessly recovery process in the event of a new release crashing or disabling the current system.

Time to Market - Devops reduces the time to market up to 50% through streamlined software delivery. It is particularly the case for digital and mobile applications.

Greater Quality - Devops helps team to improve application development quality by incorporating infrastructure issues.

Reduced Risk - Devops incorporates security aspects in the software delivery lifecycle, and it helps reduce defects across the lifecycle.

Resiliency - The operational state of software system is more stable, secure, and changes are auditable.

Cost Efficiency - Devops offers cost efficiency in the software development process, which is always an aspiration of IT management.

Breaks larger code base into small pieces - Devops is based on the agile programming method. Therefore, it allows breaking larger codebases into smaller and manageable chunks.

What is SDLC?

SDLC Stands for Software Development Life Cycle. SDLC is a process of planning, creating, testing, and deploying information systems across hardware and software.

Define Software Development Life Cycle

There are 8 stages for SDLC cycles.

Identify the Current Problems - This stage of SDLC means getting input from all which includes, customers, salespeople, industry experts, and programmers. Learn the strength and weaknesses of the current system with improvement as the goal.

Planning - This stage of SDLC means, the team determines the cost and resources required for implementing the requirements. It also requires the details of risk involved and provides the sub-plans for softening those risks.

Define Requirements - This stage of SDLC mean, defines, and documents the requirements and seek stakeholders' approval. This is done through SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

Design and Prototyping - SDLC requires a designing step that models how the application will work and aspects the design. Some of them are **UI, Programming, Security, Communications, Architecture, Platforms**. After the design is defined then prototyping of the application will be made and an early version will be rolled out for demonstration usage and for basic idea of how application will work.

Software Development - This phase of SDLC, where program itself is written out, either using a single developer or a large team each working on different parts of the development.

Testing - This phase of SDLC, where each code will be testing ensuring all the codes work as per the requirement. Applications must be tested continuously to ensure that they are going to run well together, as software development is often broken down into smaller projects completed by separate individuals and team.

Deployment - This phase of SDLC, applications are deployed once testing is completed which makes it available to users.

Operations and Maintenance - This phase of SDLC, once the application is deployed and is being used, the final phase of discovers bugs that slipped through the cracks during the testing and resolves them - this can start its own iterative process.

What is Method of Implementation

There are 2 ways of implementing the SDLC. Waterfall method and Agile method.

Waterfall - Waterfall SDLC is a linear model that has the development of the software start from the beginning and move through each step of the process, but next step cannot be started until previous step is not finished. It helps a company analyze continuity and feasibility of each step of the process, which can help eliminate bottlenecks.

Agile - The agile methodology focuses strongly on user input and experience, which can solve many issues that arose from older applications that were more cumbersome to use. The software, as it moves through the agile process, is very responsive to feedback and works to release software in quicker cycles to adhere to a changing and rapid market.

Cloud Computing

What is Cloud?

The cloud refers to servers that are accessed over the internet, and the software and databases that run on those servers.

Cloud servers are located in data centres all over the world.

The cloud enables users to access the same files and applications from almost any device, because the computing and storage takes place on servers in a data centres, instead of locally on the user device.

The cloud started off as a tech term industry slang. In the early days of the internet, tech diagrams, often represented the servers & networking infrastructure that make up the internet as a cloud.

Today, “The Cloud” is a widely accepted term for this style of computing.

What is Cloud Computing?

Cloud Computing is a network of remote servers hosted on the internet for storing and retrieving data.

The Cloud provides several IT services such as servers, databases, software’s, virtual storages, and networking.

In layman terms, Cloud Computing is defined as a virtual platform that allow you to store and access data over the internet without any limitations.

Benefits of Cloud Computing

Speed

Cost

Scalability

Accessibility

Better Security

Types of Cloud Computing

It is multiplying, resulting in it being classified into several different categories.

However out of various categories there are six that stand out. These six categories are further divided into two parts.

Cloud based deployment & Cloud based Services.

Cloud based Deployment is divided into three categories:

- Public Cloud
- Private Cloud
- Hybrid Cloud

Cloud based Services is divided into three categories:

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

What are the characteristics of Cloud Computing

There are 8 key characteristics of Cloud Computing.

- On demand self service
- Resource Pooling
- Scalability & Rapid Elasticity
- Pay-per-Use Pricing
- Measured Service
- Resiliency and Availability
- Security
- Broad Network Access

What is Resource Pooling

Computing resources like networks, servers, storage applications, and a service can be pooled to serve multiple customers by security separating the resources on a logical level.

This is done using a multi-tenant model, which allows multiple customers to share the same application or physical infrastructure while retaining data security and privacy.

What are the varieties of services cloud used to provide

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

What is difference among the varieties

IaaS:

- It is used by network architects.
- It gives access to the resources like virtual machines and virtual storage.
- It is popular among developers and researchers.

PaaS:

- It is used by developers.
- It gives access to runtime environment to deployment and development tools for applications.
- It is popular among developers who focus on the development of applications and scripts.

SaaS:

- It is used by the end user.
- It is gives access to the end user.
- It is popular among consumers and companies
- Such as file sharing, emails, and networking.

Amazon Web Services (AWS)

What is AWS

AWS is a comprehensive, easy to use computing platform offered by Amazon. The platform is developed with a combination of Iaas, PaaS and packaged SaaS offerings.

AWS is a platform that offers flexible, reliable, scalable, easy to use and cost-effective cloud computing solutions.

AWS is broadly adopted cloud platform that offers several on-demand operations like compute power, database storage, content-delivery, etc., to help corporates scale and grow.

History of AWS

- 2002 - AWS was launched
- 2006 - Launched its cloud products
- 2012 - Hold first customer event
- 2015 - Reveals revenues achieved of $4.6 billion
- 2016 - Surpassed $10 billion revenue target
- 2016 - Releases Snowball and Snowmobile
- 2019 - Offers nearly 100 cloud services
- 2021 - AWS comprises over 200 products and services

Important AWS Services

Amazon Web Services offers a wide range of different business purpose global cloud-based products. The products include storage, databases, analytics, networking, mobile development tools, enterprise applications, with pay-as-you-go pricing model.

Advantages of AWS

- AWS provides user-friendly programming model, architecture, database as well as operating system that has been already known to employers.
- AWS is very cost-effective service. There is no such thing as long-term commitments for anything you would like to purchase.
- It offers billing and management for the centralized sector, hybrid computing, and fast installation or removal of your application in any location with few clicks.
- There is no need to pay any extra money on running data servers by AWS.
- AWS offers a total ownership cost at very reasonable rates in comparison to other private cloud servers.

Dis-Advantages of AWS

- AWS has supportive paid packages for intensive or immediate response. Thus, users might need to pay extra money for that.
- There might be some cloud computing problems in AWS especially when you move to a cloud server such as backup protection, downtime, and some limited control.
- From region to region, AWS sets some default limitations on resources such as volumes, images, or snapshots.
- If there is a sudden change in your hardware system, the application on the cloud might not offer great performance.

How many types of instances are there in AWS

There are 5 types of instances in AWS:

- General Purpose
- Compute Optimized
- Memory Optimized
- Accelerated Computing
- Storage Optimized
- HPC Optimized
- Instance Features
- Measure Instance Performance

What are the key components of Elastic Cloud Computing

In AWS EC2, the users must be aware about the EC2 Components, their operating systems support, security measures, pricing model, etc.,

What is EC2 Instance

AWS EC2 is a web service interface that provides resizable compute capacity in the AWS Cloud. It is designed for developers to have complete control over web-scaling and computing resources.

EC2 Instances can be resized, and the number of instances scaled up or down as per our requirement. These instances can be launched in one or more geographical locations or regions, and Availability Zones (AZs). Each region comprises of several AZs at distinct locations, connected by low latency networks in the same region.

What is OnDemand Service

On-demand service is one of the major characteristics of cloud computing. It is a business computing model in which computing resources are made available to the user on an “as needed” basis and the user needs to pay only for the amount of resources that team requires.

On the AWS, On-Demand Instances happen to be the primary EC2 deployment model. This is the pricing option that allows you to instantly purchase uninterrupted Amazon cloud computing capacities using pay-as-you-go.

What is Spot Instance

EC2 Spot Instances, according to Amazon, can potentially save you up to 90% of what you’d otherwise spend on On-Demand Instances.

While it’s been proven that even Reserved EC2 instances are cheaper than their On-Demand counterparts, it turns out Spot Instances are additionally capable of pushing their discounts beyond the reach of the Reserved Instances.

A Spot Instance is a specialized Amazon Web Services (AWS) instance that allows you to access and utilize unused EC2 capacity at a steeply discounted rate.

Amazon offers them as a means of helping EC2 users optimize their cloud computing costs - by capitalizing on idle EC2 capacity, instead, instead of relying entirely on the rather costlier On-Demand Instances.

What is AWS Reserved Instance

An AWS reserved instance is officially described as a "billing discount" applied to the use of an on-demand instance in your account. In other words, a reserved instance is not actually a physical instance, rather **it is the discounted billing you get when you commit to using a specific on-demand instance for a long-term period of one or three years.**

Reserved instances are ideal for steady and predictable usage. They can help you save significantly on your Amazon EC2 costs compared to on-demand instance pricing because in exchange for your commitment to pay for all the hours in a one-year or three-year term, the hourly rate is lowered significantly.

Introduction to LINUX

What is Operating System?

Operating System is an interface between user and the computer hardware. The hardware of the computer cannot understand the human readable language as it works on binaries i.e., 0’s and 1’s.

Also, it is very tough for humans to understand the binary language., in such case we need an interface which can translate human language to hardware and vice-versa for effective communication.

Types of Operating System:

- ✓ Single User - Single Task Operating System
- ✓ Single User - Multitasking Operating System
- ✓ Multi User - Multitasking Operating System

Single User - Single Task Operating System:

In this type of OS only one user can log into system and can perform only one task at a time.

Single User - Multitasking Operating System:

This type of OS supports only one user to log into the system, but a user can perform multiple tasks at a time, browsing internet, playing games, playing songs/video’s etc.

Eg: Windows - 98, XP, Vista, 7, 8, 10

Multi User - Multi Tasking Operating System:

This type of O/S provides multiple users to log into the system and each user can perform various tasks at a time.

In a broader term multiple users can logged in to system and share the resources of the system at the same time.

Eg: UNIX / LINUX

What is LINUX

Linux is an open-source operating system like other operating systems such as Microsoft Windows, Apple Mac OS, iOS, Google Android etc.

An operating system is a software that enables the communication between computer hardware and software. It conveys input to get processed by the processor and brings output to the hardware to display it. This is the basic function of every operating system.

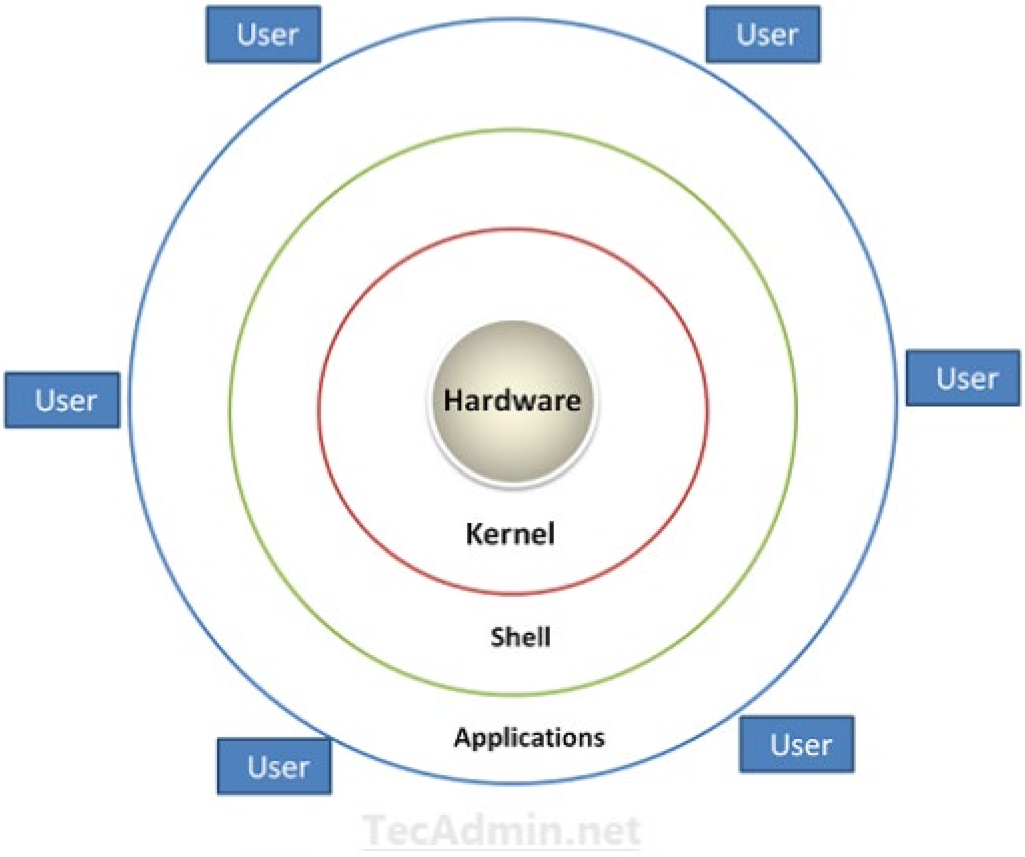
Linux is around as since mid-90s. It can be used from wristwatches to supercomputers. It is everywhere in our phones, laptops, PCs, cars, and even in refrigerators. It is very much famous among developers and normal computers.

Evolution of LINUX OS'

The LINUX OS was developed by Linux Torvalds in 1991, which sprouted as an idea to improve UNIX OS. He suggested improvements but was rejected by UNIX designers. Therefore, he thought of launching an OS, designed in a way that could be modified by its users.

Nowadays, Linux is the fastest growing OS. It is used from phones to supercomputers by almost all major hardware devices.

Architecture of LINUX



Linux System Architecture

An operating system is a collection of software, designed for a specific function.

Linux OS has following components:

Kernel - The Kernel is one of the core sections of an operating system. It is responsible for each of the major actions of the Linux OS. This operating system contains distinct types of modules and cooperates with underlying hardware directly. The Kernel facilitates required abstraction for hiding details of low-level hardware or application programs to the system. There are some of the important kernel types which are mentioned below:

- Monolithic Kernel
- Micro Kernels
- Exo Kernels
- Hybrid Kernels

System Libraries - These libraries can be specified as some special functions. These are applied for implementing the operating system's functionality and don't need code access rights of the modules of kernels.

System Utility Programs - It is responsible for doing specialized level and individual activities.

Hardware Layer - Linux operating system contains a hardware layer that consists of several peripheral devices like CPU, HDD, and RAM.

Shell - It is among the Kernel and User. It can afford the services of Kernel. It can take commands through the users and runs the functions of the Kernel. The Shell is available in distinct types of OSes. These operating systems are categorized into two different types, which are **graphical shells** and **command-line shells**.

Explain File Hierarchy of LINUX

The Linux File Hierarchy Structure or the Filesystem Hierarchy Standards (FHS) defines the directory structure and directory contents in Unix-like operating systems. It is maintained by the Linux Foundation.

In FHS, all files and directories appear under the root directory /, even if they are stored on different physical or virtual devices.

Some of these directories only exists on a particular system if the certain subsystems, such as Windows System, are installed.

Most of these directories exist in all UNIX Operating Systems and are generally used in much the same way; however, the descriptions here are those used specifically for the FHS and are not considered authoritative for platforms other than Linux.

Linux uses single rooted, inverted tree like file system hierarchy

/	• This is top level directory. • It is parent directory for all other directories. • It is called as ROOT directory. • It is represented by forward slash (/)
/root	• It is home directory for root users (Super User). • It provides working environment for root user.
/home	• It is home directory for other users. • It provides working environment for other users (other than root)
/boot	• It contains bootable files of Linux. • Like vmlinux (Kernel) …… ntoskrnl Initrd (Initial Ram Disk). • Like GRUB (Grand Unified Boot Loader) … boot.ini, ntldr.
/etc	It contains all configuration files
/usr	By default, software's are installed in /usr directory. (UNIX Sharable Resources)
/opt	• It is optional directory for /usr. • It contains third-party software's.
/bin	It contains commands used by all users. (Binary Files)

/sbin	It contains commands used by only Super User (root). (Super User’s binary files)
/dev	It contains device files. Like for hard disk /dev/hda, for cd rom /dev/cd. Like device manager of windows.
/proc	• It contains process files. Its contents are not permanent, they keep changing. • It is also called as Virtual Directory. • Its file contains useful information used by OS. • Like information of RAM/SWAP - /proc/meminfo. • Like information of CPU - /proc/cpuinfo
/var	It contains variable data like mails, log files.
/tmp	It contains the temporary files for small period time.
/mnt	• It is default mount point for any partition. • It is empty by default.
/media	It contains all removable media like CD-ROM, pen-drive.
/lib	• It contains library files which are used by OS. • It is like dll files in Windows. • Library files in Linux are SO (Shared Object) files.

Basic Commands

Creating, Removing, Copying, Moving files & Directories

Creating a file in Linux

Using cat command:

```
cat > myfile

Hello world
```

To display the content using cat command:

```
cat myfile
```

To append the data in the already existing file

```
cat >> myfile
```

Welcome back to Linux

Creating multiple files at same time using touch command

Syntax: `touch <file_name-1> <file_name-2> <file_name-1>`

Method1: `touch file1 file2 file3`

Method2: `touch file{1-15}`

Creating a single directory

```
mkdir mydir
```

Creating multiple directories

```
Mkdir -p World/{India/{Hyd, Chennai}, AUS/{Melbourne, Perth},
USA/{California, Miami}}
```

```
tree world
```

Copying files into directory

Syntax: cp <source filename> <destination directory in which to paste the file>

```
cp file1 mydir
```

Copying directories from one location to other

Syntax: cp -rvfp <dir_name> <destination name>

```
cp -rvfp mydir2 mydir
```

Moving files from one location to other

Syntax: mv <dir_name> <destination directory>

```
mv file3 mydir
```

Moving directories from one location to other

Syntax: mv <dir_name> <destination directory name>

```
mv mydir2 mydir8
```

Renaming a file / Directory

Syntax: mv <old_name> <new_name>

```
mv file2 file8
```

```
mv mydir2 mydir9
```

Removing a file / Directory

Syntax1: rm <file_name>

```
rm file2
```

Syntax2 (Without prompting): rm -f <file_name>

```
Rm -f file10
```

Syntax3: rm <dir_name>

```
rm mydir3
```

Syntax4: rmdir -rf <dir_name> == To remove dir_name with files/directories

rmdir -rf mydir3 == where **-r** stands for **recursive** and **f** stands for **forcefully**.

GIT

Source Code Management

Source Code Management SCM is used to track modifications to a source code repository. SCM tracks a running history of changes to a code base and helps resolve conflicts when merging updates from multiple contributors.

It is a software tool that programmers use to manage source code. It tracks modifications to a source code repository and helps deal with merge conflicts.

Importance of Source Code Management

- Track Changes
- Synchronization
- Backup & Restore
- Undoing
- Branching & Merging
- Identify Conflicts & Preventing Overwrites

What is Version Control System

Version Control is also known as Source Control, is the practice of tracking and managing changes to software code. Version Control systems and software tools that help software team manage changes to source code over time.

Version Control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.

What are the types of Version Control System

There are 3 types of Version Control System.

- Local Version Control System
- Centralized Version Control System
- Distributed Version Control System

Local Version Control System:

A local version control system is a local database located on your local computer, in which every file change is stored as a patch. The main problem with this is that everything is stored locally. If anything were to happen to the local database all the patches would be lost.

Also, collaborating with other developers or a team is very hard or nearly impossible.

Centralized Version Control System:

A centralized version control system has a single server that contains all the file versions. This enables multiple clients to simultaneously access files on the server, pull them to their local computer or push them onto the server from their local computer.

This allows for easy collaboration with other teammates.

Distributed Version Control System:

Distributed Version Control System, clients don’t just check out the latest snapshot of the files from the server, they fully mirror the repository, including its full history. Thus, everyone collaborating on a project owns a local copy of the whole project, i.e., owns their own local database with their own complete history.

What are the demerits of Central Version Control System

The biggest dis-advantage is the single-point of failure embedded within the centralized server.

If the remote server goes down, then no one can work on the code or push changes.

The lack of offline access means that any disruption can significantly impact code development and even result in code loss.

What are the advantages & dis-advantages of Distributed Version Control System.

Advantages	Dis-Advantages
Because of local commits, the full history is always available.	It may not always be obvious who did the most recent changes
No need to access a remote server (faster access)	File locking doesn't allow different developers to work on the same piece of code simultaneously. It helps to avoid merge conflicts but slow down the development.
Ability to push your changes continuously	DVCS enables you to clone the clone the repository - this could mean a security issue.
Saves time, especially with SSH Keys	Managing non-mergeable files is contrary to the DVCS concept
Good for projects with offshore developers	Working with lot of binary files require huge amount of space, and developers can't do diffs.

What is Git

Git is version control system used for tracking changes in computer files. It is generally used for source code management in software development.

- Git is used to track changes in source codes
- The distributed version control tool is used for source code management
- It allows multiple developers to work together
- It supports non-linear development through it thousands of parallel branches

Features of GIT:

- Tracks History
- Free and open source
- Supports non-linear development
- Creates Backups
- Scalable
- Supports collaboration
- Branching is easier
- Distributed development

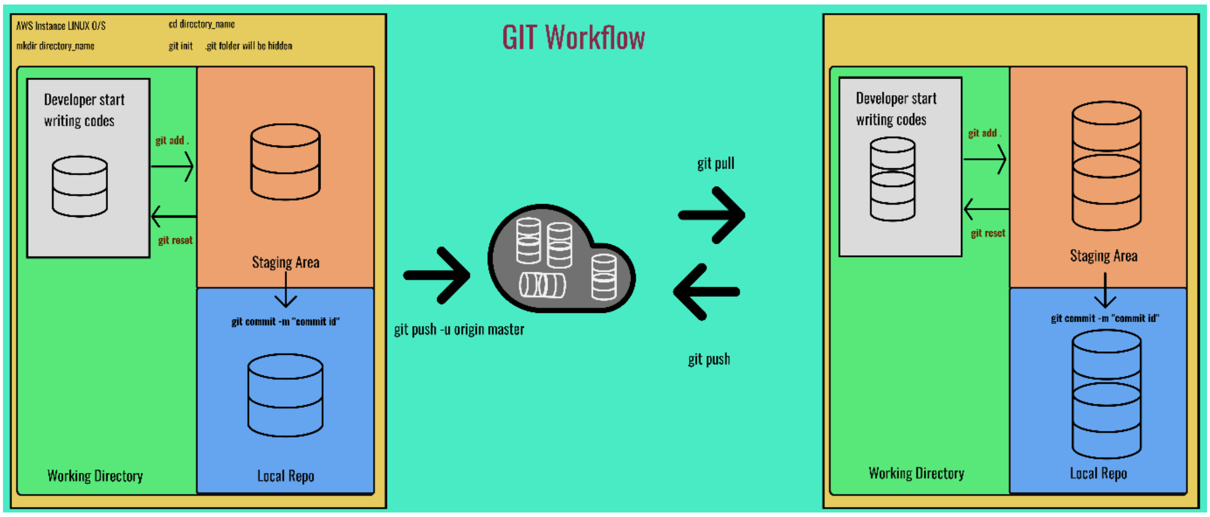
Git Workflow

The Git workflow is divided into three stages:

Working Directory: Modify files in working directory

Staging Area: Stage the files and add snapshots of them to your staging area

Git Directory / Local Repo: Perform a commit that stores the snapshots permanently to your Git directory. Checkout any existing version, make changes, stage them, and commit.


AWS Instance LINUX O/S
cd directory_name
mkdir directory_name
git init
.git folder will be hidden
GIT Workflow
Developer start writing codes
git add .
git reset
Staging Area
git commit -m "commit id"
Working Directory
Local Repo
git pull
git push -u origin master
git push
Developer start writing codes
git add .
git reset
Staging Area
git commit -m "commit id"
Working Directory
Local Repo