PHOTO SHARING ANDROID APPLICATION

A project report submitted in partial fulfillment of the requirement for degree of

Bachelor of Technology
in
Computer Science Engineering
By
paturi Sandhya

Under the guidance of

Satyanandaram N

Lecturer in Department of Computer Science Engineering

Rajiv Gandhi University of Knowledge Technologies
RGUKT
R.K.Valley

Kadapa(Dt), Andhra Pradesh(516329), India

TABLE OF CONTENT

1.0 ABSTRACT

"Photo Sharing" is an android application which acts as platform to share photos with known persons securely.Here each one can have individual accounts and they can create sections like family,friends,relatives,etc.. under these sections,they can add their users based on their interest to whom he/she wants to share the photos with.whenever user shares photos under any particular section ,users of this particular section will be notified.
All might have this opinion that, this is quite similar to whatsapp and but this application has its own advantage over whatsapp..

Advantage over whatsapp is that user can view photo without having to download it to gallery ,in order to save it in the internal memory one could download the photo.So clearly, we are saving storage.

2.0 INTRODUCTION

2.1 Objective

The main objective of this android application is sharing photos with known persons personally in the groups that is created by you or group that you were added by others.

2.2 Problem Definition

This application helps us to save our storage. In any application if user sends photo in order to view it, that has to get downloaded locally which occupies some storage but here this application saves storage in order to view it.

2.3 Purpose

The purpose of this application is saving storage for the users without downloading other photos locally.

2.4 Scope

The Scope of this application lies on saving storage.If user want to save storage, one can go with this application.This is how the application scope is based on.

3.0 LITERATURE REVIEW

3.1 Amazon Web Services

Introduction

In 2006, Amazon Web Services (AWS) started to offer IT services to the market in the form of web services, which is nowadays known as cloud computing. With this cloud, we need not plan for servers and other IT infrastructure which takes up much of time in advance. Instead, these services can instantly spin up hundreds or thousands of servers in minutes and deliver results faster. We pay only for what we use with no up-front expenses and no long-term commitments, which makes AWS cost efficient.Today, AWS provides a highly reliable, scalable, low-cost infrastructure platform in the cloud that powers multitude of businesses in 190 countries around the world.

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers,storage,applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

History of Cloud Computing

Before emerging the cloud computing, there was Client/Server computing which is basically a centralized storage in which all the software applications, all the data and all the controls are resided on the server side.

If a single user wants to access specific data or run a program, he/she need to connect to the server and then gain appropriate access, and then he/she can do his/her business.

Then after, distributed computing came into picture, where all the computers are networked together and share their resources when needed.

But of course time has passed and the technology caught that idea and after few years we mentioned that:

In 1999, **Salesforce.com** *started delivering of applications to users using a simple website*. The applications were delivered to enterprises over the Internet, and this way the dream of computing sold as utility were true.

In 2002, **Amazon** *started Amazon Web Services*, providing services like storage, computation and even human intelligence. However, only starting with the launch of the Elastic Compute Cloud in 2006 a truly commercial service open to everybody existed.

In 2009, **Google Apps** *also started to provide cloud computing enterprise applications*Of course, all the big players are present in the cloud computing evolution, some were earlier, some were later. *In 2009*, **Microsoft** *launched Windows Azure*, and companies like Oracle and HP have all joined the game. This proves that today, cloud computing has become mainstream.

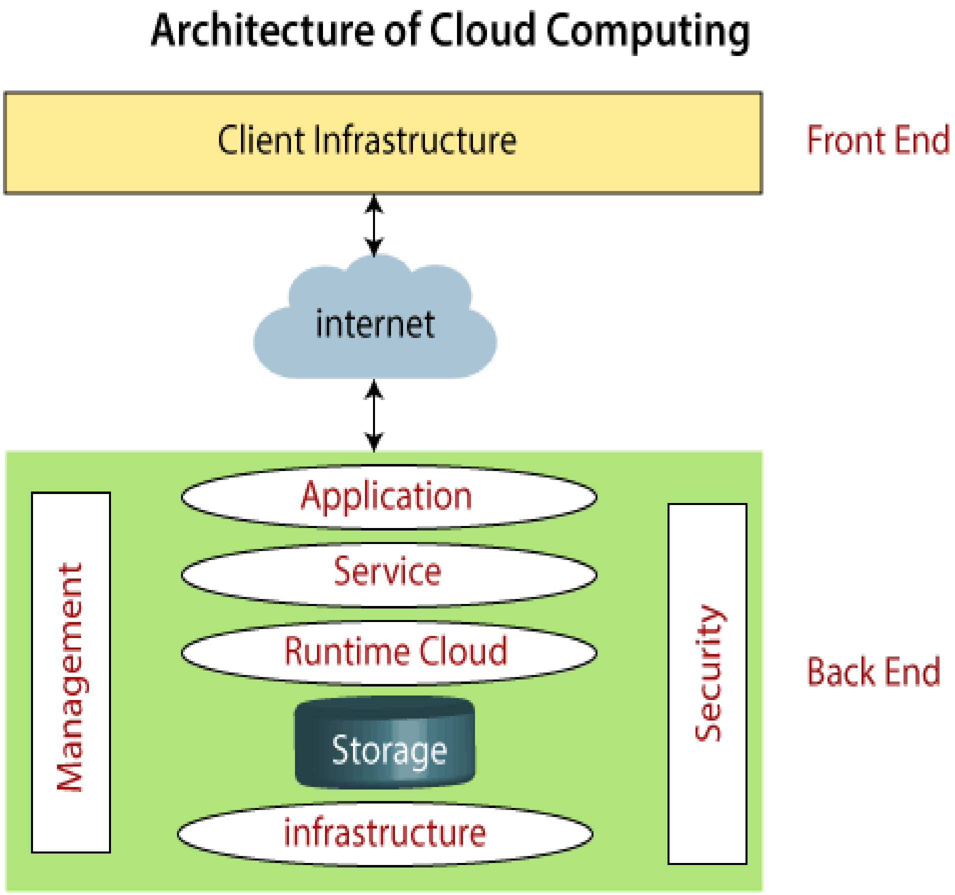
Cloud Computing Architecture

Cloud computing architecture is a combination of **service-oriented architecture** and **event-driven architecture**.

Cloud computing architecture is divided into the following two parts -

- Front End
- Back End

The below diagram shows the architecture of cloud computing.



Front End

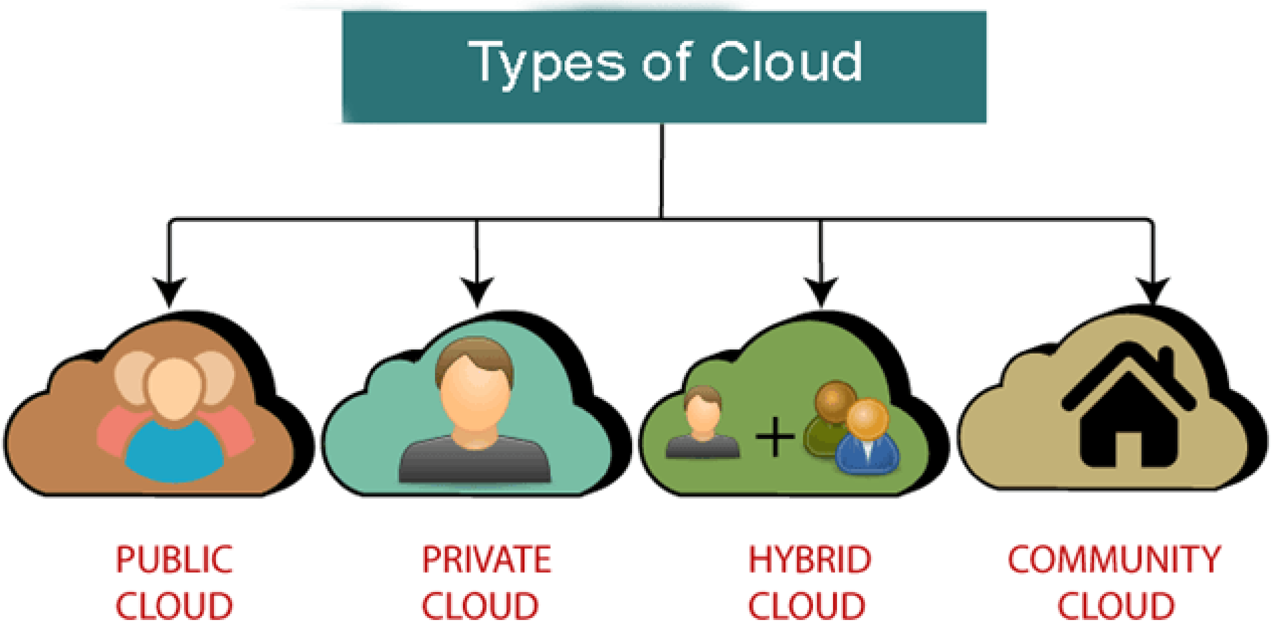
The front end is used by the client. It contains client-side interfaces and applications that are required to access the cloud computing platforms. The front end includes web servers (including Chrome, Firefox, internet explorer, etc.), thin & fat clients, tablets, and mobile device.

Back End

The back end is used by the service provider. It manages all the resources that are required to provide cloud computing services. It includes a huge amount of data storage, security mechanism, virtual machines, deploying models, servers, traffic control mechanisms, etc.

Types of Cloud

There are the following 4 types of cloud that you can deploy according to the organization's needs-

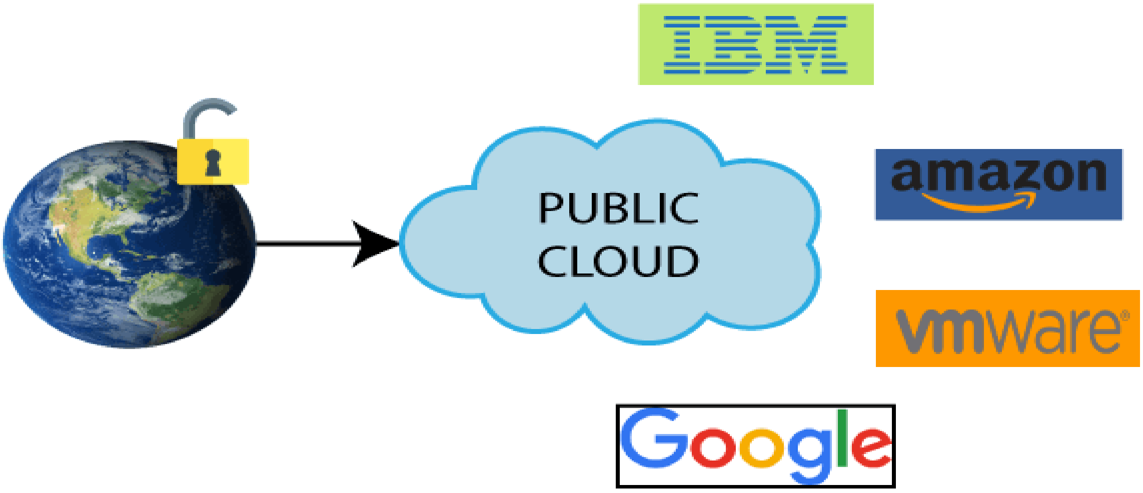


Public Cloud

Public cloud is **open to all** to store and access information via the Internet using the pay-per-usage method.

In public cloud, computing resources are managed and operated by the Cloud Service Provider (CSP).

Example: Amazon elastic compute cloud (EC2), IBM SmartCloud Enterprise, Microsoft, Google App Engine, Windows Azure Services Platform.

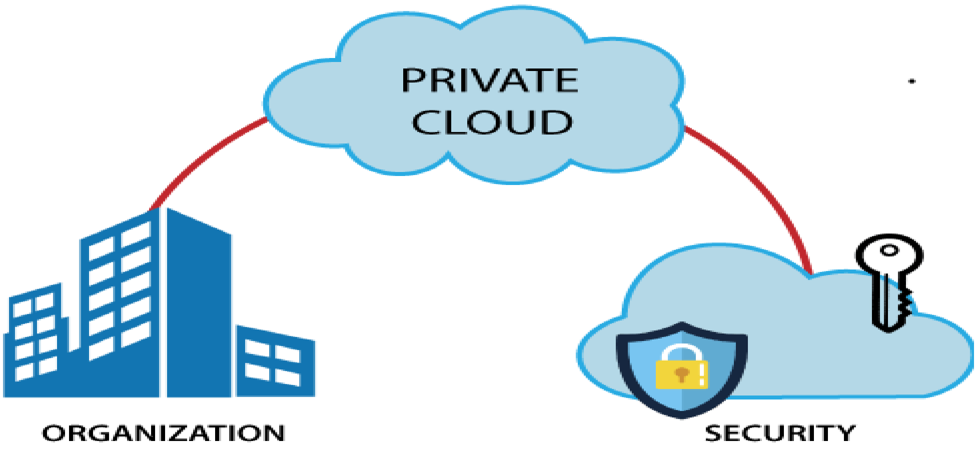


Private Cloud

Private cloud is also known as an **internal cloud** or **corporate cloud**. It is used by organizations to build and manage their own data centers internally or by the third party. It can be deployed using Opensource tools such as Openstack and Eucalyptus.

Based on the location and management, National Institute of Standards and Technology (NIST) divide private cloud into the following two parts-

- ➔ On-premise private cloud
- ➔ Outsourced private cloud



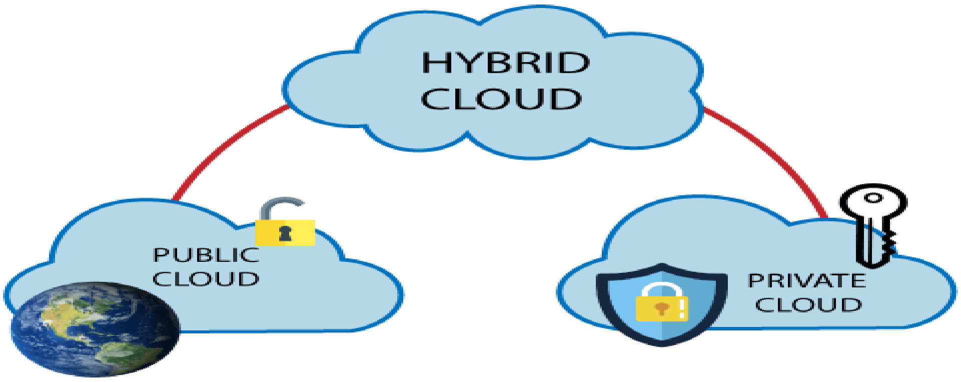
Hybrid Cloud

Hybrid Cloud is a combination of the public cloud and the private cloud. we can say:

Hybrid Cloud = Public Cloud + Private Cloud

Hybrid cloud is partially secure because the services which are running on the public cloud can be accessed by anyone, while the services which are running on a private cloud can be accessed only by the organization's users.

Example: Google Application Suite (Gmail, Google Apps, and Google Drive), Office 365 (MS Office on the Web and One Drive), Amazon Web Services.



Community Cloud

Community cloud allows systems and services to be accessible by a group of several organizations to share the information between the organization and a specific community. It is owned, managed, and operated by one or more organizations in the community, a third party, or a combination of them.

Example: Health Care community cloud

Community Cloud
Org 1
Org 2
Org 3

Difference between public cloud, private cloud, hybrid cloud, and community cloud -

The below table shows the difference between public cloud, private cloud, hybrid cloud, and community cloud.

Parameter	Public Cloud	Private Cloud	Hybrid Cloud	Community Cloud
Host	Service provider	Enterprise (Third party)	Enterprise (Third party)	Community (Third party)
Users	General public	Selected users	Selected users	Community members
Access	Internet	Internet, VPN	Internet, VPN	Internet, VPN
Owners	Service provider	Enterprise	Enterprise	Community

Cloud Service Models

There are the following three types of cloud service models -

1.Infrastructure as a Service (IaaS)

2.Platform as a Service (PaaS)

3.Software as a Service (SaaS)

IaaS:

IaaS stands for Infrastructure as a Service. It provides users with the capability to provision processing, storage, and network connectivity on demand. Using this service model, the customers can develop their own applications on these resources.

PaaS:

PaaS stands for Platform as a Service. Here, the service provider provides various services like databases, queues, workflow engines, e-mails, etc. to their customers. The customer can then use these components for building their own applications. The services, availability of resources and data backup are handled by the service provider that helps the customers to focus more on their application's functionality.

SaaS:

SaaS stands for Software as a Service. As the name suggests, here the third-party providers provide end-user applications to their customers with some administrative capability at the application level, such as the ability to create and manage their users. Also some level of

customizability is possible such as the customers can use their own corporate logos, colors, etc.

AWS SECURITY

Cloud security at AWS is the highest priority. As an AWS customer, you will benefit from a data center and network architecture built to meet the requirements of the most security-sensitive organizations. Security in the cloud is much like security in your on- premises data centers—only without the costs of maintaining facilities and hardware. In the cloud, you don’t have to manage physical servers or storage devices. Instead, you use software-based security tools to monitor and protect the flow of information into and out of your cloud resources.

Benefits of AWS Security

➔ Keep Your Data Safe: The AWS infrastructure puts strong safeguards in place to help protect your privacy. All data is stored in highly secure AWS data centers.

➔ Meet Compliance Requirements: AWS manages dozens of compliance programs in its infrastructure. This means that segments of your compliance have already been completed.

➔ Save Money: Cut costs by using AWS data centers. Maintain the highest standard of security without having to manage your own facility

➔ Scale Quickly: Security scales with your AWS Cloud usage. No matter the size of your business, the AWS infrastructure is designed to keep your data safe.

AWS COMPLIANCE

Compliance is nothing but “the action of complying with a command,” or “the state of meeting rules or standards”.AWS Cloud Compliance enables you to understand the robust controls in place at AWS to maintain security and data protection in the cloud. As systems are built on top of AWS Cloud infrastructure, compliance responsibilities will be shared. By tying together governance-focused, audit-friendly service features with applicable compliance or audit standards, AWS Compliance enablers build on traditional programs. This helps customers to establish and operate in an AWS security control environment.

AWS - Management Console::

AWS Management Console is a web application for managing Amazon Web Services. AWS Management Console consists of list of various services to choose from. It also provides all information related to our account like billing.This console provides an inbce to perform AWS tasks like working with Amazon S3 buckets, launching and connecting to Amazon EC2 instances, setting Amazon CloudWatch alarms, etc.

AWS – Console Mobile App::

The AWS Console mobile app, provided by Amazon Web Services, allows its users to view resources for select services and also supports a limited set of management functions for select resource types.

➔ Features of AWS Mobile App :

To have access to the AWS Mobile App, we must have an existing AWS account. Simply create an identity using the account credentials and select the region in the menu. This app allows us to stay signed in to multiple identities at the same time.For security reasons, it is recommended to secure the device with a passcode and to use an IAM user's credentials to log in to the app. In case the device is lost, then the IAM user can be deactivated to prevent unauthorized access.Root accounts cannot be deactivated via mobile console.

While using AWS Multi-Factor Authentication (MFA), it is recommended to use either a hardware MFA device or a virtual MFA on a separate mobile device for account security reasons.The latest version is 1.14. There is a feedback link in the App's menu to share our experiences and for any queries.

●EC2 (Elastic Compute Cloud)

- Browse, filter and search instances.
- View configuration details.
- Check status of CloudWatch metrics and alarms.
- Perform operations over instances like start, stop, reboot, termination.
- Manage security group rules.
- Manage Elastic IP Addresses.

● Elastic Load Balancing

- Browse, filter and search load balancers.
- View configuration details of attached instances.
- Add and remove instances from load balancers.

- Elastic Load Balancing
  - Browse, filter and search load balancers.
  - View configuration details of attached instances.
  - Add and remove instances from load balancers.
- S3
  - Browse buckets and view their properties.
  - View properties of objects.
- Route 53
  - Browse and view hosted zones.
  - Browse and view details of record sets.
- RDS (Relational Database Service)
  - Browse, filter, search and reboot instances.
  - View configuration details, security and network settings.
- Auto Scaling
  - View group details, policies, metrics and alarms.
  - Manage the number of instances as per the situation.
- Elastic Beanstalk
  - View applications and events.
  - View environment configuration and swap environment CNAMEs.
  - Restart app servers.
- DynamoDB
  - View tables and their details like metrics, index, alarms, etc.
- CloudFormation
  - View stack status, tags, parameters, output, events, and resources.
- OpsWorks
  - View configuration details of stack, layers, instances and applications.
  - View instances, its logs, and reboot them.

- CloudWatch

  - View CloudWatch graphs of resources.
  - List CloudWatch alarms by status and time.
  - Action configurations for alarms.

- Services Dashboard

  - Provides information of available services and their status.
  - All information related to the billing of the user.
  - Switch the users to see the resources in multiple accounts.

AMAZON FEATURED SERVICES

1. Amazon EC2 (Elastic Compute Cloud):

It is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.These are just the virtual machines in the cloud on which you have the OS level control. You can run whatever you want in them.Amazon EC2 offers the broadest and deepest compute platform with choice of processor, storage, networking, operating system, and purchase model.It offer the
fastest processors in the cloud and it is the only cloud with 400 Gbps ethernet networking. EC2 has the most powerful GPU instances for machine learning training and graphics workloads, as well as the lowest cost-per-inference instances in the cloud. More SAP, HPC, Machine Learning, and Windows workloads run on AWS than any other cloud.

What is an Instance?

An instance is a virtual server for running applications on Amazon’s EC2. It can also be understood like a tiny part of a larger computer, a tiny part which has its own Hard drive, network connection, OS etc. But it is actually all virtual. You can have multiple “tiny”computers on a single physical machine, and all these tiny machines are called Instances.

Why AWS EC2 ?

Suppose you are a developer, and since you want to work independently you buy some servers, you estimated the correct capacity, and the computing power is enough. Now,you have to look after the updating of security patches every day, you have to troubleshoot any problem which might occur at a back end level in the servers and so on.
But if you buy an EC2 instance, you don’t have to worry about any of these things as it will .

Types of EC2 Computing Instances

Computing is a very broad term, the nature of your task decides what kind of computing you need.
Therefore,AWS EC2 offers 5 types of instances which are as follows:

1. General Instances

For applications that require a balance of performance and cost.
E.g email responding systems, where you need a prompt response as well
as the it should be cost effective, since it doesn't require much processing.

2. Compute Instances

For applications that require a lot of processing from the CPU.
E.g analysis of data from a stream of data, like Twitter stream

3. Memory Instances

For applications that are heavy in nature, therefore, require a lot of RAM.
E.g when your system needs a lot of applications running in the background
i.e multitasking.

4. Storage Instances

For applications that are huge in size or have a data set that occupies a lot ofspace.
E.g When your application is of huge size.

5. GPU Instances

For applications that require some heavy graphics rendering.
E.g 3D modeling etc.

Amazon Networking and content delivery

1.Amazon Virtual Private Cloud

It is simply a data center in the cloud in which you deploy all your resources. It allows you to better isolate your resources and secure them.You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 for most resources in your virtual private cloud, helping to ensure secure and easy access to resources and applications.

For example, you can create a public-facing subnet for your web servers that has access to the Internet, and place your backend systems, such as databases or application servers, in a private-facing subnet with no Internet access. You can leverage multiple layers of security (including security groups and network access control lists) to help control access to EC2 instances in each subnet.

2.Amazon CloudFront

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment.

It is AWS's Content Delivery Network (CDN) that consists of Edge locations that cache resources.

3.Amazon Route 53

A reliable and cost-effective way to route end users to Internet applications.
It is AWS's highly available DNS (Domain Name System) service. You can register domain names through it.Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet applications by translating names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other. Amazon Route 53 is fully compliant with IPv6 as well.

Benefits

- Highly available and reliable
- Flexible
- Designed for use with Amazon Web Services
- Simple
- Fast
- Cost-effective

- Secure
- Scalable
- Simplify the Hybrid cloud

4.Amazon Direct Connect

Using it you can connect your data center to an Availability zone using a high speed dedicated line.AWS Direct Connect permits to create a private network connection from our network to AWS location. It uses 802.1q VLANs, which can be partitioned into multiple virtual interfaces to access public resources using the same connection. This results in reduced network cost and increased bandwidth. Virtual interfaces can be reconfigured at any time as per the requirement.

5.API Gateway

Allows you to create, store and manage APIs at scale.

AWS Storage

AWS offers a complete range of services for you to store, access, govern, and analyze your data to reduce costs, increase agility, and accelerate innovation. Select from object storage, file storage, and block storage services, backup,& data migration options to build the foundation of your cloud IT environment.

1.S3(Simple Storage Service)

Storage service of AWS in which we can store objects like files, folders, images, documents, songs, etc. It cannot be used to install software, games or Operating System.

S3 (Simple Storage Service) is a scalable, high-speed, low-cost web-based service designed for online backup and archiving of data and application programs. It allows to upload, store, and download any type of files up to 5 TB in size. This service allows the subscribers to access the same systems that Amazon uses to run its own web sites. The subscriber has control over the accessibility of data, i.e. privately/publicly accessible.

We can add your own code to process data retrieved from S3 before returning it to an application

2.Amazon Elastic Block Store

Amazon Elastic Block Store (EBS) is an easy to use, high-performance, block-storage service designed for use with Amazon Elastic Compute Cloud (EC2) for both throughput and transaction intensive workloads at any scale. A broad range of workloads, such as relational and non-relational databases, enterprise applications, containerized applications, big data analytics engines, file systems, and media workflows are widely deployed onAmazon EBS

AWS DATABASE

1.Amazon RDS

Allows you to run relational databases like MySQL, MariaDB, PostgreSQL,Oracle or SQL Server. These databases are fully managed by AWS like installing antivirus and patches.It is a fully-managed SQL database cloud service that allows to create and operate relational databases. Using RDS you can access your files and database anywhere in a cost-effective and highly scalable way.

Amazon Relational Database Service (Amazon RDS) makes it easy to set up,operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups.

It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

2.Amazon DynamoDB

It is a fully managed NoSQL database service that allows to create database tables that can store and retrieve any amount of data. It automatically manages the data traffic of tables over multiple servers and maintains performance. It also relieves the customers from the burden of operating and scaling a distributed database. Hence, hardware provisioning, setup, configuration, replication, software patching, cluster scaling, etc. is
managed by Amazon. It is a highly scalable, high performance NoSQL database. It provides single-digit millisecond latency at any scale.

Here,In our android application development we are going to store user generated content in cloud storage repository called firebase as well as it provides high security to our applications

3.2 DEVOPS

DevOps

DevOps is a set of practices that combines software development and IT operations. It aims to shorten the systems development life cycle and provide continuous delivery with high software quality.

DevOps is complementary with Agile software development; several DevOps aspects came from the Agile methodology.

I. What is DevOps

DevOps is a collaboration between Development and IT Operations to make software production and deployment in an automated & repeatable way.

DevOps helps to increase the organization's speed to deliver software applications and services. The word 'DevOps' is a combination of two words, 'Development' and 'Operations.

DevOps is a practice that allows a single team to manage the entire application development life cycle, that is, development, testing, deployment, and monitoring.

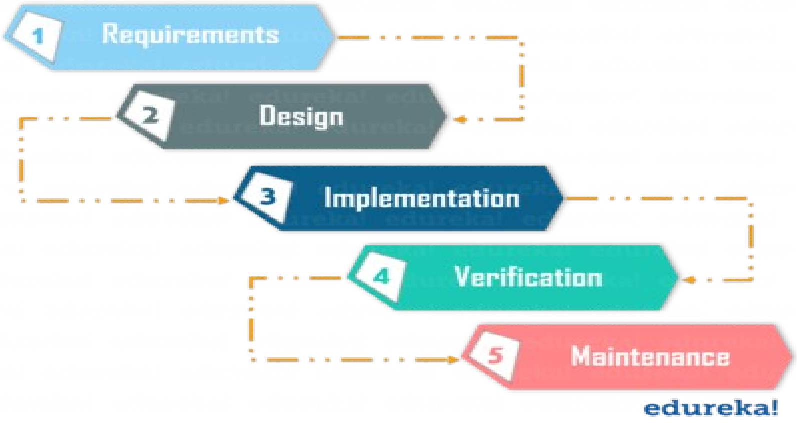
- ➔ The ultimate goal of DevOps is to decrease the duration of the system’s development life cycle while delivering features, fixes, and updates frequently in close synchronization with business objectives.
- ➔ DevOps is a software development approach with the help of which you can develop superior quality software quickly and with more reliability. It consists of various stages such as continuous development, continuous integration, continuous testing, continuous deployment, and continuous monitoring.

History Of DevOps

Waterfall model

The waterfall model is a software development model that is pretty straight forward and linear. This model follows a top-down approach.

This model has various starting with Requirements gathering and analysis. This is the phase where you get the requirements from the client for developing anapplication. After this, you try to analyze these requirement.



- The next phase is the Design phase where you prepare a blueprint of the

  software.
- Once the design is ready, you move further with the Implementation phase whereyou begin with the coding for the application. The team of developers works together on various components of the application.
- Once you complete the application development, you test it in the Verification phase. There are various tests conducted on the application such as unit testing, integration testing, performance testing, etc.
- After all the tests on the application are completed, it is deployed onto the production servers.
- At last, comes the Maintenance phase. In this phase, the application is monitored for performance. Any issues related to the performance of the application ar resolved in this phase.

Advantages of the Waterfall Model:

- Simple to understand and use
- Allows for easy testing and analysis
- Saves a significant amount of time and money
- Good for small projects if all requirements are clearly define

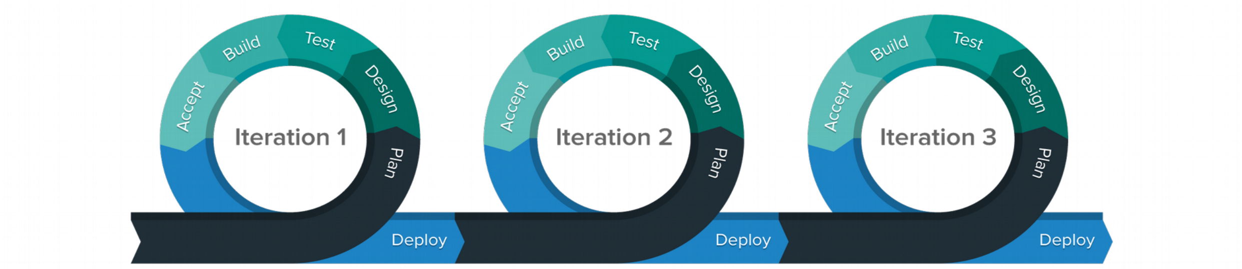
Disadvantages of Waterfall Model:

- Risky and uncertain
- Lack of visibility of the current progress•Not suitable when the requirements keep changing

- Difficult to make changes to the product when it is in the testing phase
- The end product is available only at the end of the cycle Not suitable for large and complex projects

Agile Methodology

Agile Methodology is an iterative based software development approach where the software project is broken down into various iterations or sprints. Each iteration has phases like the waterfall model such as Requirements Gathering, Design, Development, Testing, and Maintenance. The duration of each iteration is generally 2-8 weeks.



Agile Process

- n Agile, a company releases the application with some high priority features in the first iteration.
- After its release, the end-users or the customers give you feedback about the performance of the application.
- Then you make the necessary changes into the application along with some new features and the application is again released which is the second iteration.
- You repeat this entire procedure until you achieve the desired software quality.

Advantages of Agile Model

- It adaptively responds to requirement changes favorably
- Fixing errors early in the development process makes this process more cost- effective
- Improves the quality of the product and makes it highly error-free
- Allows for direct communication between people involved in software project
- Highly suitable for large & long-term projects
- Minimum resource requirements & very easy to manage

Disadvantages of Agile Model

- Highly dependent on clear customer requirements
- Quite Difficult to predict time and effort for larger projects
- Not suitable for complex projects
- Lacks documentation efficiency
- Increased maintainability risks

Why is DevOps is Needed?

- Before DevOps, the development and operation team worked in complete isolation.
- Testing and Deployment were isolated activities done after design-build. Hence they consumed more time than actual build cycles.
- Without using DevOps, team members are spending a large amount of their time in testing, deploying, and designing instead of building the project.
- Manual code deployment leads to human errors in production
- Coding & operation teams have their separate timelines and are not in synch causing further delays.

How is DevOps different from traditional IT

let's compare traditional software waterfall model with DevOps to understand the changes DevOps bring.

Old Process	Devops
After placing an order for new servers, the Development team works on testing. The Operations team works on extensive paperwork as required in enterprises to deploy the infrastructure.	After placing an order for new servers,the Development Operations team work together on the paperwork to set-up the new servers. This results in better visibility of infrastructure requirement.
Projection about failover, redundancy, data center locations, and storage requirements are skewed as no inputs are available from developers have deep knowledge of the from	Projection about failover, disatster recovery, data center locations, and storage who requirements are pretty accurate due to the inputs

the developers. application.	
Operations team has no clue on the progress of the Development team. Opefations team develop a monitoring plan as per their understanding.	In devOps,the Operations team is completely aware of the progress the dvelopers are making Operations team interact with developers and jointly develop monitoring plan that caters to the IT and business needs.They also use advance Application Performance Monitoring(APM)
Before go-live,the load testing crashes the application.The release is delayed	Before go-live,the load testing makes the application a bit slow,The development team quickly fixes the bottlenecks.

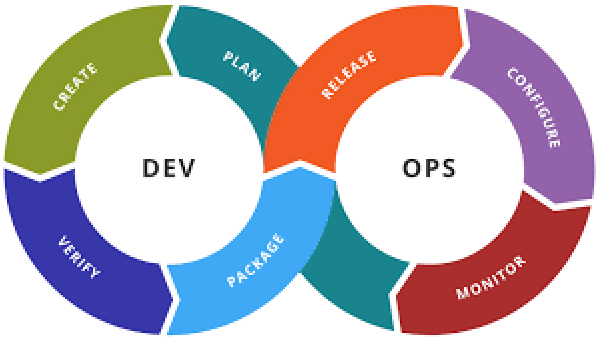
Why is DevOps used?

DevOps allows Agile Development Teams to implement Continuous Integration and Continuous Delivery. This helps them to launch products faster into the market.

When to adopt DevOps?

DevOps should be used for large distributed applications such as eCommerce sites or applications hosted on a cloud platform.

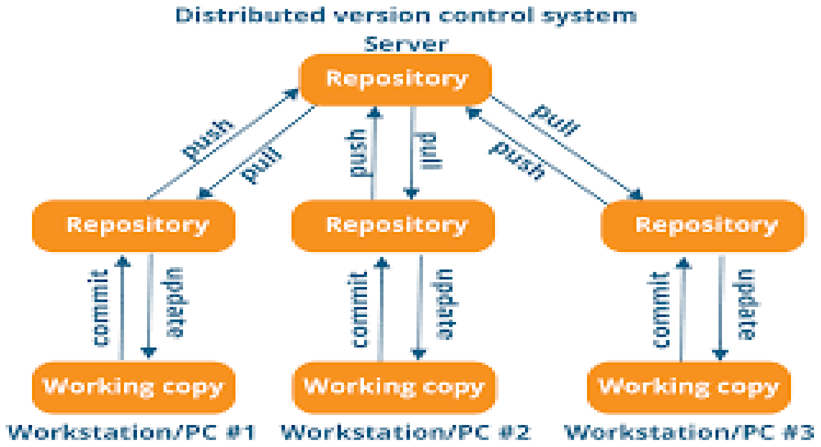
DEVOPS LIFECYCLE



Stage – 1: Continuous Development

Tools Used: Git, SVN, Mercurial, CVS

Process Flow:



- This is the phase that involves ‘planning‘ and ‘coding‘ of the software. You decide the project vision during the planning phase and the developers begin developing the code for the application.

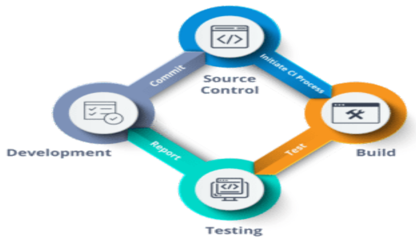
- There are no DevOps tools that are required for planning, but there are a number of tools for maintaining the code.

- The code can be in any language, but you maintain it by using Version Control tools. This process of maintaining the code is known as Source Code Management.

Stage – 2: Continuous Integration

Tools: Jenkins, TeamCity, Travis

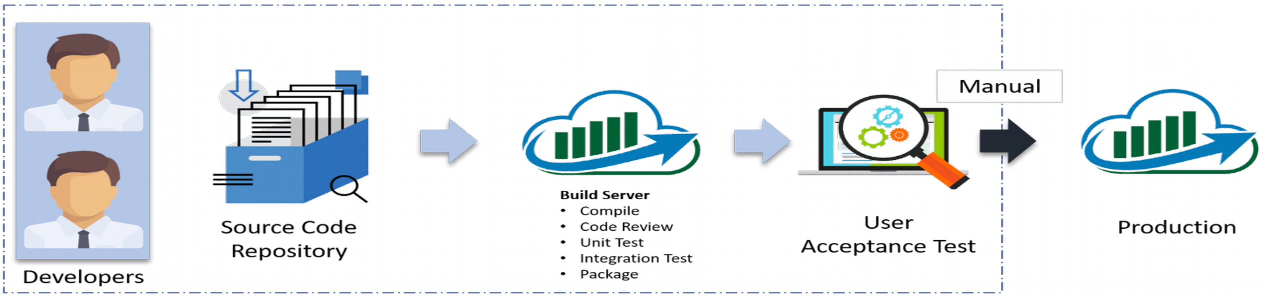
Process Flow:



Stage – 3: Continuous Testing

Tools: Jenkins,

Process Flow:



This is the stage where you test the developed software continuously for bugs using automation testing tools. These tools allow QAs to test multiple code-bases thoroughly in parallel to ensure that there are no flaws in the functionality. In this phase, you can use Docker Containers for simulating the test environment.

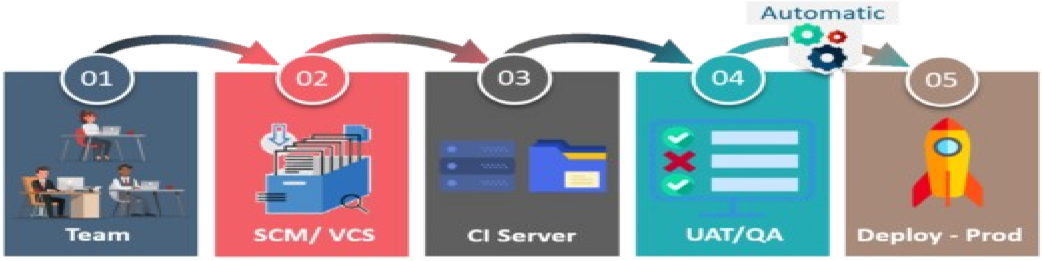
- ***Selenium*** is used for automation testing, and the reports are generated by ***TestNG***. You can automate this entire testing phase with the help of a Continuous Integration tool called Jenkins.

Stage – 4: Continuous Deployment

Tools Used: Configuration Management – Chef, Puppet, Ansible

Containerization – Docker, Vagrant

Process Flow:

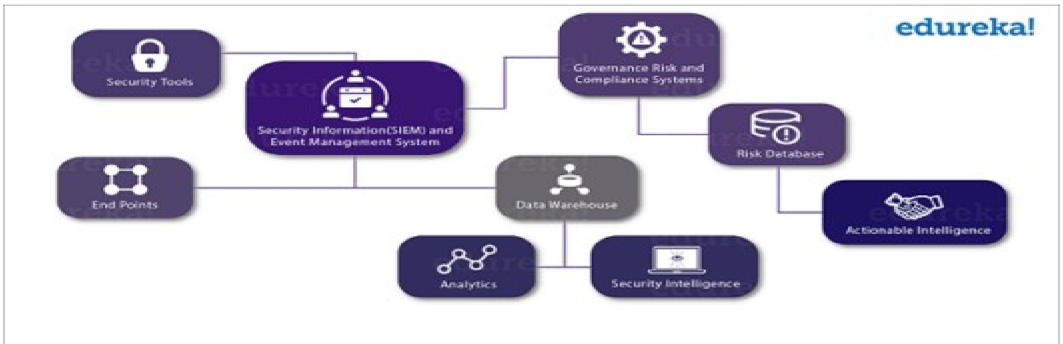


This is the stage where you deploy the code on the production servers. It is also important to ensure that you correctly deploy the code on all the servers. Beforemoving on, let us try to understand a few things about Configuration managementn and Containerization tools. These set of tools here help in achieving Continuous Deployment (CD).

Stage – 5: Continuous Monitoring

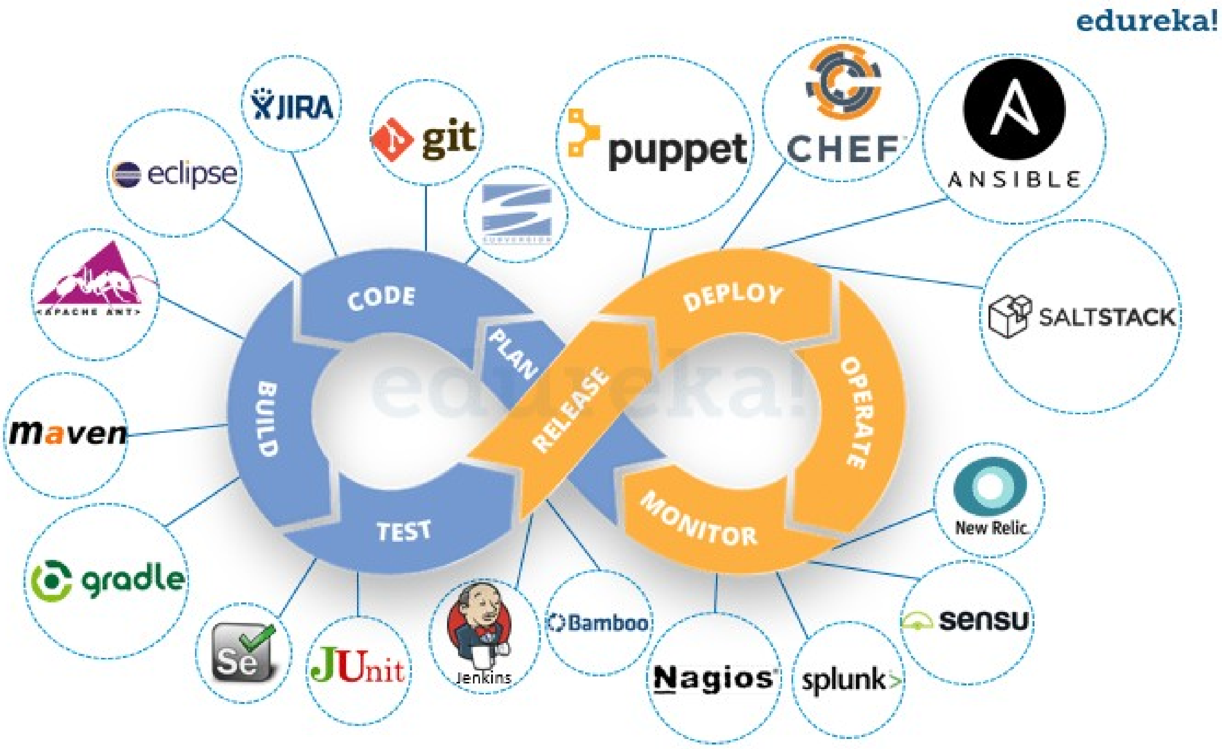
Tools Used: Splunk, ELK Stack, Nagios, New Relic

Process Flow:



•This is a very critical stage of the DevOps life cycle where you continuously monitor the performance of your application. Here you record vital information about the use of the software. You then process this information to check the proper functionality of the application.

DEVOPS TOOLS



1) Puppet

Puppet is the most widely used DevOps tool. It allows the delivery and release of the technology changes quickly and frequently. It has features of versioning, automated testing, and continuous delivery. It enables to manage entire infrastructure as code without expanding the size of the team.

2) Ansible

Ansible is a leading DevOps tool. Ansible is an open-source IT engine that automates application deployment, cloud provisioning, intra service orchestration, and other IT tools. It makes it easier for DevOps teams to scale automation and speed up productivity.

Ansible is easy to deploy because it does not use any agents or custom security infrastructure on the client-side, and by pushing modules to the clients. These modules are executed locally on the client-side, and the output is pushed back to the Ansible server.

3) Docker

Docker is a high-end DevOps tool that allows building, ship, and run distributed applications on multiple systems. It also helps to assemble the apps quickly from the components, and it is typically suitable for container management.

4) Nagios

Nagios is one of the more useful tools for DevOps. It can determine the errors and rectify them with the help of network, infrastructure, server, and log monitoring systems.

5) Chef

This technology uses Ruby encoding to develop essential building blocks such as recipes and cookbooks. The chef is used in infrastructure automation and helps in reducing manual and repetitive tasks for infrastructure management.

Chef has got its convention for different building blocks, which are required to manage andautomate infrastructure.

6) Jenkins

Jenkins is a DevOps tool for monitoring the execution of repeated tasks. Jenkins is a software that allows continuous integration. Jenkins will be installed on a server where the central build will take place. It helps to integrate project changes more efficiently by finding the issues quickly.

7) Git

Git is an open-source distributed version control system that is freely available for everyone. It is designed to handle minor to major projects with speed and efficiency. It is developed to co-ordinate the work among programmers. The version control allows you to track and work together with your team members at the same workspace. It is used as a critical distributed version-control for the DevOps tool.

8) Saltstack

Stackify is a lightweight DevOps tool. It shows real-time error queries, logs, and more directly into the workstation. SALTSTACK is an ideal solution for intelligent orchestration for the software- defined data center.

9) Splunk

Splunk is a tool to make machine data usable, accessible, and valuable to everyone. It delivers operational intelligence to DevOps teams. It helps companies to be more secure, productive, and competitive.

3.3 AWS DevOps

AWS is the best cloud service provider, and DevOps is the implementation of the software development lifecycle.

Here are some reasons which make AWS DevOps a highly popular combination, such as:

- AWS CloudFormation
- AWS EC2
- AWS CloudWatch
- AWS CodePipeline
-

AWS CloudFormation

DevOps team is required to create and release cloud instances and services more frequently in comparison to development teams. Templates of AWS resources such as EC2 instances, ECScontainers, and S3 storage buckets let you set up the entire stack without having to bring everything together.

AWS EC2

You can run containers inside EC2 instances. Hence you can leverage the AWS security and management features.

AWS CloudWatch

This monitoring tool tracks every resource that AWS has to offer. It makes it easy to use third-party tools for monitoring such as sumo logic etc.

AWS CodePipeline

Code Pipeline is an essential feature from AWS, which highly simplifies the way youmanage your CI/CD toolset. It integrates with tools such as Jenkins, GitHub, and CodeDeploy that enable you to visually control the flow of app updates from build to production

DevOps vs Agile

Parameter	DevOps	Agile
Definition	DevOps is a practice of brining devlopment and operation teams together	Agile refers to the continuous iterative approach, which focuses on collaboration, customer feedback, small, and rapid releases.
Purpose	Devops purpose is to manage end to end engineering processes	The agile purpose is to
Task	It focuses on constant testing and delivery	It focuses on constant changes.
Team Size	It has a large team size as it involves all the stack holders.	It has a small team size. As smaller is the team, the fewer people work on it so that they can move faster.
Team skillset	The DevOps divides and spreads the skill set between development and the operation team.	The Agile development emphasizes training all team members to have a wide variety of similar and equal skills.
Implementation	DevOps is focused on collaboration, so it does not have any commonly accepted framework	Agile can implement within a range of tactical frameworkssuch as safe, scrum, and sprint.
Duration	The ideal goal is to deliver the code to production daily or every few hours.	Agile development is managed in units of sprints. So this time is much less than a month for each sprint.
Target areas	End to End business solution and fast delivery.	Software development.
Feedback	Feedback comes from the internal team.	In Agile, feedback is coming from the customer.
Shift Left Principle	It supports both variations left and right.	It supports only shift left.
Focus	DevOps focuses on operational and business readiness.	Agile focuses on functional and non-functional readiness.
Importanace	In DevOps, developing, testing, and implementation all are equally important.	Developing software is inherent to Agile.

Quality	DevOps contributes to creating better quality with automation and early bug removal. Developers need to follow Coding and best Architectural practices to maintain uality standards.	The Agile produces better applications suites with the desired requirements. It can quickly adapt according to the changes made on time during the project life.
Tools	Puppet, Chef, AWS, Ansible, and team City OpenStack are popular DevOps tools.	Bugzilla, Kanboard, JIRA are some popular Agile tools.
Automation	Automation is the primary goal of DevOps. It works on the principle of maximizing efficiency when deploying software.	Agile does not emphasize on the automation.
Communication	DevOps communication involves specs and design documents. It is essential for theoperational team to fully understand the software release and its network implications for theenough running deployment process.	Scrum is the most common method of implementing Agile software development. Scrum meeting is carried out daily.

3.4 GITHUB

INTRODUCTION

Step-1 What is GitHub

GitHub is a Git repository hosting service.It is a web-based service.It is a file or code-sharing service to collaborate with different people. GitHub is a highly used software that is typically used for version control. It is helpful when more than just one person is working on a project.

Example a software developer team wants to build a website and everyone has to updatetheir codes simultaneously while working on the project. In this case, Github helps them tobuild a centralized repository where everyone can upload, edit, and manage the code files.

Why is Github so popular?

GitHub has various advantages but many people often have a doubt as to why not use dropbox or any cloud based system? Let, Say more than two software developers are working on the same file and they want to update it simultaneously. Unfortunately, theperson who save the file first will get precedence over the others.While in Github, this is not the case. Github document the changes and reflect them in an organized manner to avoid any chaos between any of the files uploaded.

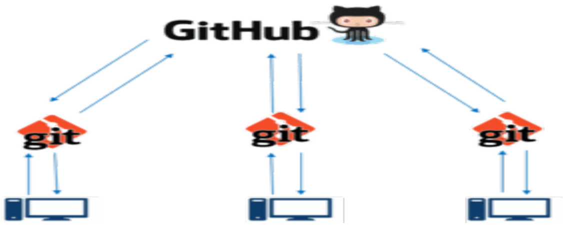
Therefore using GitHub centralized repository, it avoids all the confusion and working onthe same code becomes very easy.

What is Git.?

Git is an open-source distributed version control system. It is designed to handle minor to major projects with high speed and efficiency. It is developed to co-ordinate the work among the developers.

The version control allows us to track and work together with our team members at the same workspace. Git is foundation of many services like GitHub and GitLab, but we can use Git without using any other Git services. Git can be used privately and publicly.

Git was created byLinus Torvalds in 2005 to develop Linux Kernel. It is also used as an important distributed version-control tool for the DevOps.



GIT VS GITHUB

GitHub is a central repository and Git is a tool which allows you to create a local repository. Now people usually get confused between git and GitHub but its actually very different.

Git is a version control tool that will allow you to perform all kinds of operations to fetch data from the central server or push data to it whereas GitHub is a core hosting platform for version control collaboration. GitHub is a company that allows you to host a central repository in a remote server.

Git	GitHub
Git is a distributed version control tool that can manage a programmer's source code history.	GitHub is a cloud-based tool developed around the Git tool.
A developer installs Git tool locally.	GitHub is an online service to store code and push from the computer running the Git tool.
Git focused on version control and code sharing.	GitHub focused on centralized source code hosting.
It is a command-line tool.	It is administered through the web.
It facilitates with a desktop interface It also called Git Gui.	It also facilitates with a desktop interface called GitHub Gui.
Git does not provide any user management feature.	GitHub has a built-in user management management feature.
It has minimal tool confguration feature.	It has a market place for tool confguration

What is version control?

Version control allows you to keep track of your work and helps you to easily explore the changes you have made, be it data, coding scripts, notes, etc. You are probably already doing some type of version control, if you save multiple files,. This approach will leave you with tens or hundreds of similar files, making it rather cumbersome to directly compare different versions, and is not easy to share among collaborators. With version control software such as Git version control is much smoother and easier to implement. Using an online platform like Github to store your files means that you have an online back up of your work, which is beneficial for both you and your collaborators.

Features of GitHub

GitHub is a place where programmers and designers work together. They collaborate, contribute, and fix bugs together. It hosts plenty of open source projects and codes of various programming languages.

Some of its significant features are as follows.

- Collaboration
- Integrated issue and bug tracking
- Graphical representation of branches
- Git repositories hosting
- Project management
- Team management
- Code hosting
- Track and assign tasks
- Conversations
- Wikisc

Benefits of GitHub

GitHub can be separated as the Git and the Hub. GitHub service includes access controls as well as collaboration features like task management, repository hosting, and team management.

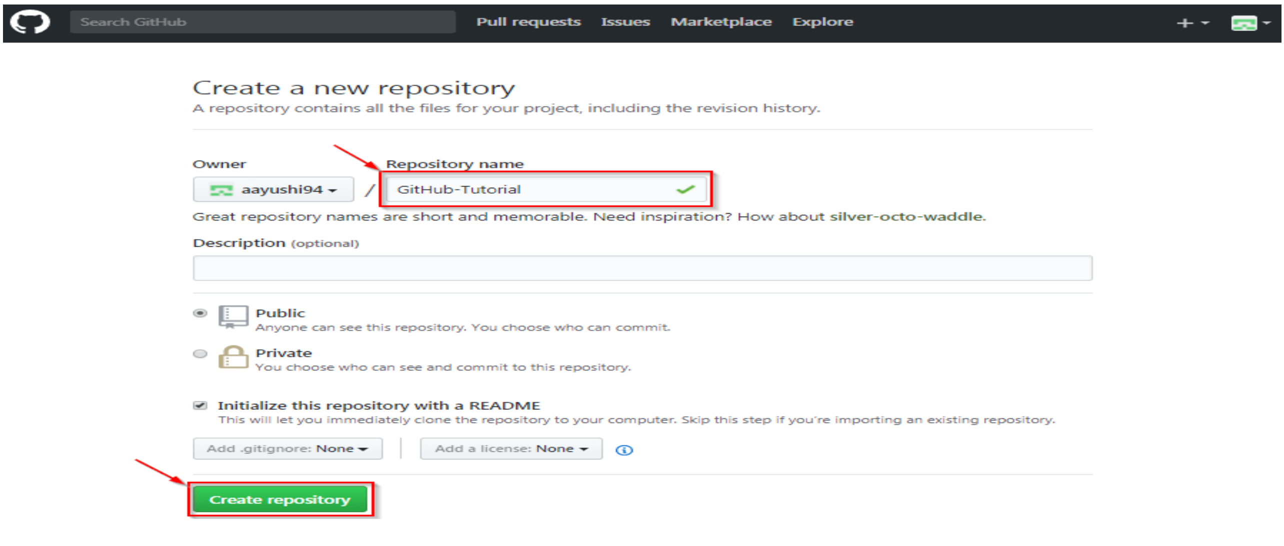
The key benefits of GitHub are as follows.

- It is easy to contribute to open source projects via GitHub.
- It helps to create an excellent document.
- You can attract recruiter by showing off your work. If you have a profile on GitHub, you will have a higher chance of being recruited.
- It allows your work to get out there in front of the public.
- You can track changes in your code across versions.
- GitHub provides you a beautiful visual interface which helps you to track or manage your version controlled projects locally.
- Once you register on GitHub, you can connect with social network and build a strong profile.

Step 2: How to create a GitHub Repository?

A repository is a storage space where your project lives. It can be local to a folder on your computer, or it can be a storage space on GitHub or another online host. You can keep code files, text files, images or any kind of a file in a repository. You need a GitHub repository when you have done some changes and are ready to be uploaded. This GitHub repository acts as your remote repository. So let me make your task easy, just follow these simple steps to create a GitHub repository:

- Go to the link: github.com Fill the sign up form and click on "Sign up for Github".
- Click on "Start a new project".



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: aayushi94 / Repository name: GitHub-Tutorial

Great repository names are short and memorable. Need inspiration? How about silver-octo-waddle.

Description (optional)

● Public
Anyone can see this repository. You choose who can commit.

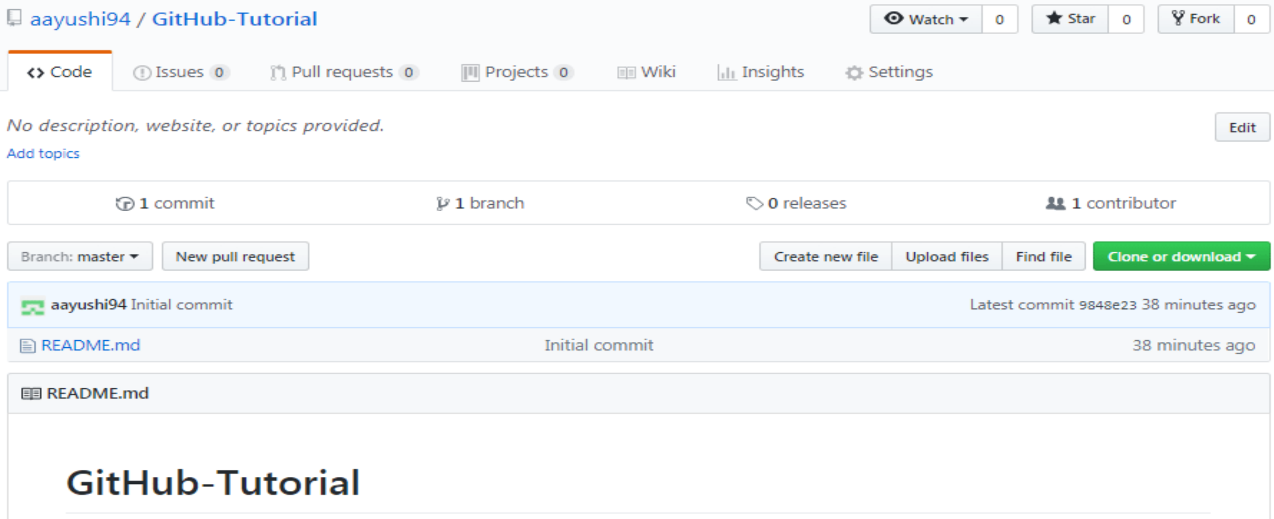
○ Private
You choose who can see and commit to this repository.

☑ Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None

Now, if you noticed by default a GitHub repository is public which means that anyone can view the contents of this repository whereas in a private repository, you can choose who can view the content. Also, private repository is a paid version. Also, if you refer the above screenshot, initialize the repository with a README file. This file contains the description of the file and once you check this box, this will be the first file inside your repository.

Congratulations, your repository is successfully created! It will look like the below screenshot:



aayushi94 / GitHub-Tutorial

No description, website, or topics provided.

1 commit | 1 branch | 0 releases | 1 contributor

aayushi94 Initial commit — Latest commit 9848e23 38 minutes ago

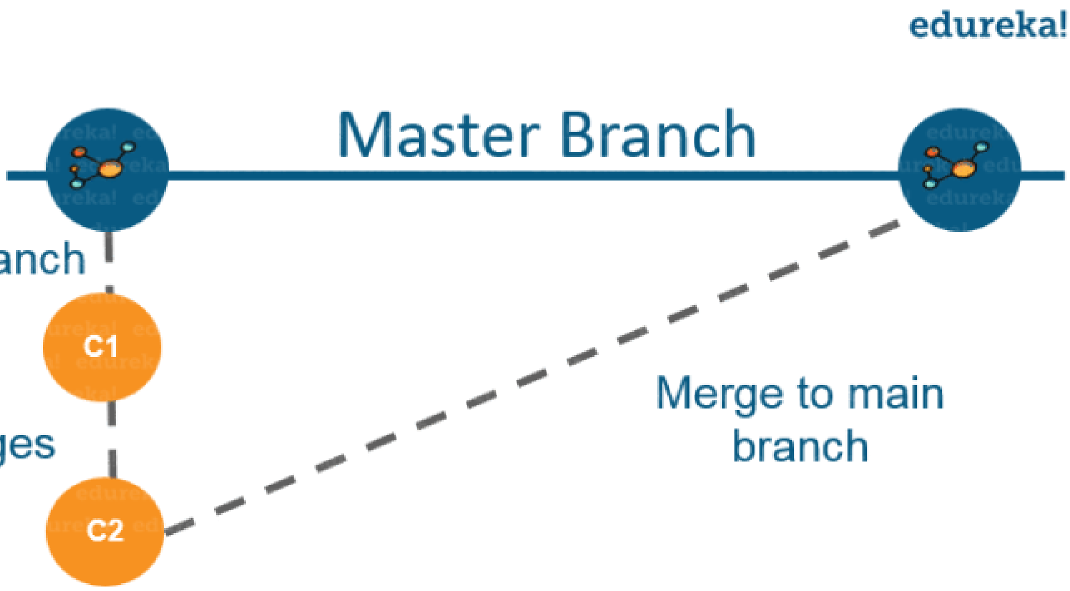
README.md — Initial commit — 38 minutes ago

README.md

GitHub-Tutorial

So now my central repository has been sucessfully created! Once this is done, you are ready to commit, pull, push and perform all the other operations. Now let’s move forward and understand branching in GitHub.

Step 3: Create Branches and Perform Operations

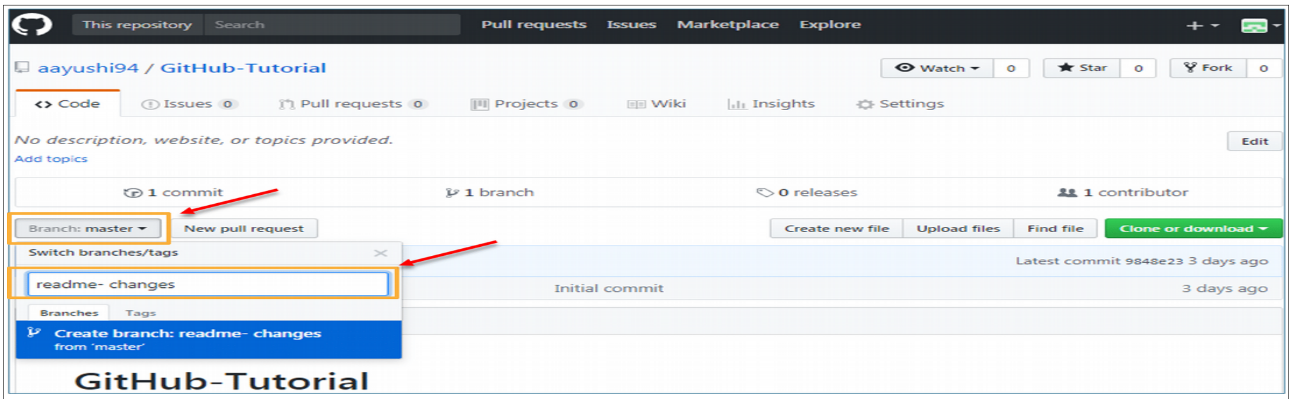


As depicted in the above image, there is a master/ production branch which has a new branch for testing. Under this branch, two set of changes are done and once it completed, it is merged back to the master branch. So this is how branching works!

Let’s move ahead in ‘how to use GitHub’ blog, and learn how you can create a branch.

To create a branch in GitHub, follow the below steps:

- Click on the dropdown “Branch: master”
- As soon as you click on the branch, you can find an existing branch or you can create a new one. In my case, I am creating a new branch with a name “readme- changes”. Refer to the below screenshot for better understanding.



readme- changes

Once you have created a new branch, you have two branches in your repository now i.e. read-me (master branch) and readme- changes. The new branch is just the copy of master branch. So let's perform some changes in our new branch and make it look different from the master branch.

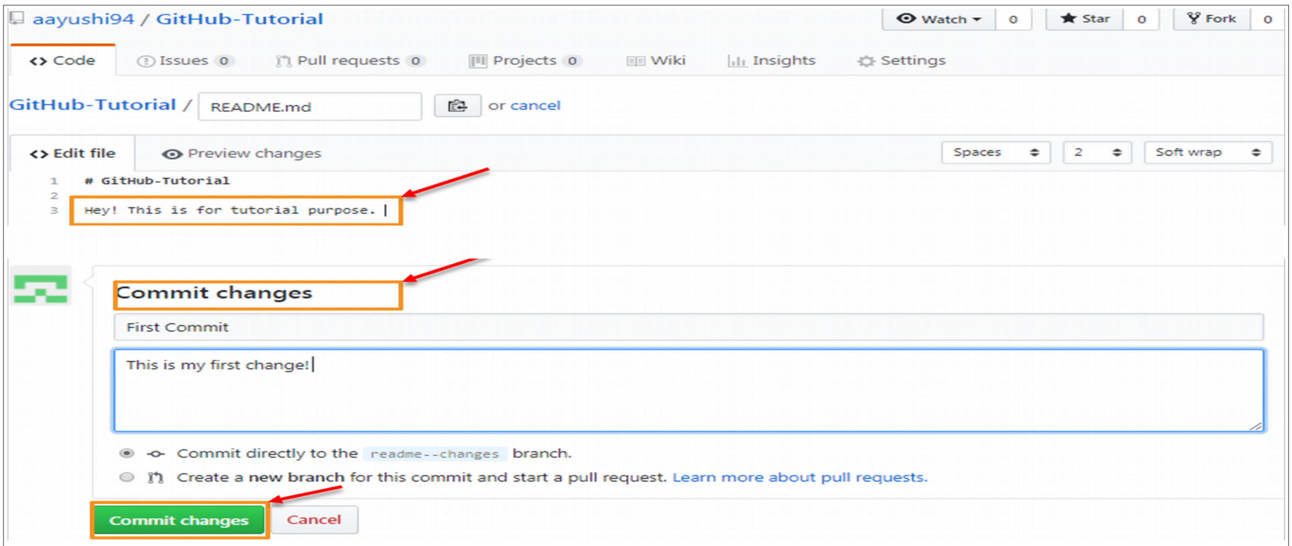
How to use GitHub: Operations

Commit Command:

This operation helps you to save the changes in your file. When you commit a file, you should always provide the message, just to keep in the mind the changes done by you. Though this message is not compulsory but it is always recommended so that it can differentiate the various versions or commits you have done so far to your repository. These commit messages maintain the history of changes which in turn help other contributors to understand the file better. Now let's make our first commit, follow the below steps:

Click on "readme- changes" file which we have just created.

- Click on the "edit" or a pencil icon in the righmost corner of the file.
- Once you click on that, an editor will open where you can type in the changes or anything.
- Write a commit message which identifies your changes.
- Click commit changes in the end.



```
# GitHub-Tutorial

Hey! This is for tutorial purpose.
```

First Commit

This is my first change!

● Commit directly to the readme--changes branch.

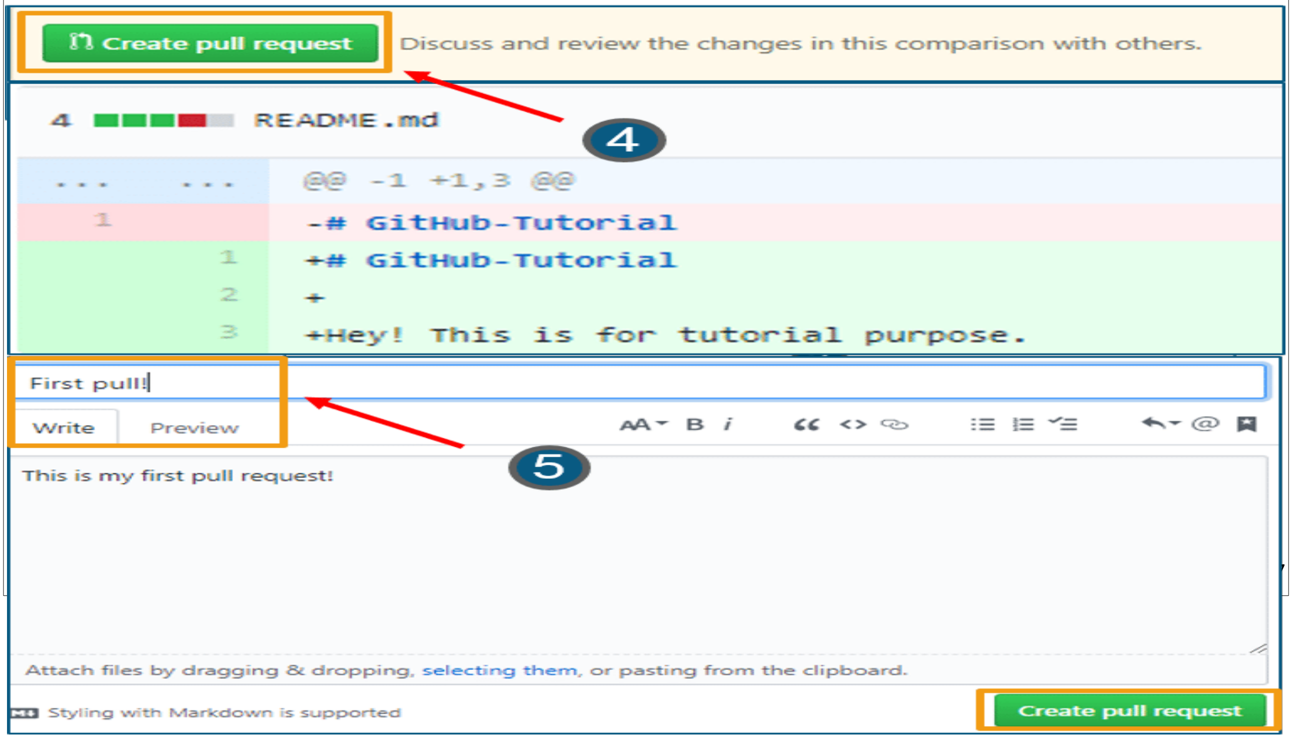
○ Create a new branch for this commit and start a pull request. Learn more about pull requests.

Pull Command

Pull command is the most important command in GitHub. It tell the changes done in the file and request other contributors to view it as well as merge it with the master branch. Once the commit is done, anyone can pull the file and can start a discussion over it. Once its all done, you can merge the file. Pull command compares the changes which are done in the file and if there are any conflicts, you can manually resolve it. Now let us see different steps involved to pull request in GitHub.

Click the ‘Pull requests’ tab.

- Click ‘New pull request’.
- Once you click on pull request, select the branch and click ‘readme- changes’ file to view changes between the two files present in our repository.
- Click “Create pull request”.
- Enter any title, description to your changes and click on “Create pull request”. Refer to the below screenshots.


Create pull request
Discuss and review the changes in this comparison with others.
4
5

```
4 README.md
...   ...   @@ -1 +1,3 @@
1           -# GitHub-Tutorial
      1     +# GitHub-Tutorial
      2     +
      3     +Hey! This is for tutorial purpose.
```

First pull!

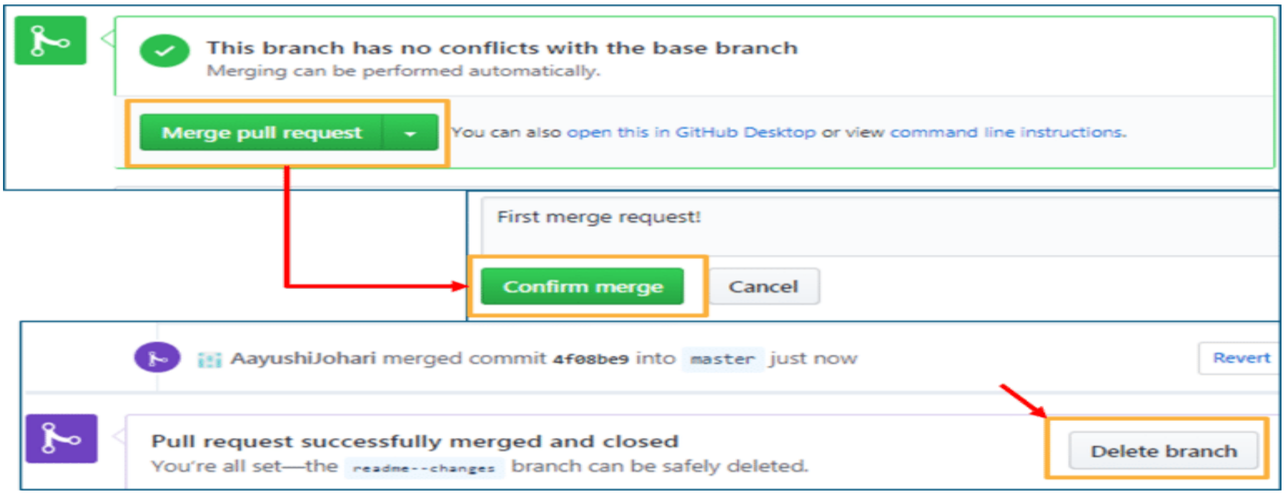
This is my first pull request!

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.
Styling with Markdown is supported
Create pull request

Merge Command

Here comes the last command which merge the changes into the main master branch. We saw the changes in pink and green color, now let's merge the "readme- changes" file with the master branch/ read-me. Go through the below steps to merge pull request.

- **Click on "Merge pull request" to merge th**e changes into master branch.
- Click "Confirm merge".
- You can delete the branch once all the changes have been incorporated and if there are no conflicts. Refer to the below screenshots.



This branch has no conflicts with the base branch
Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

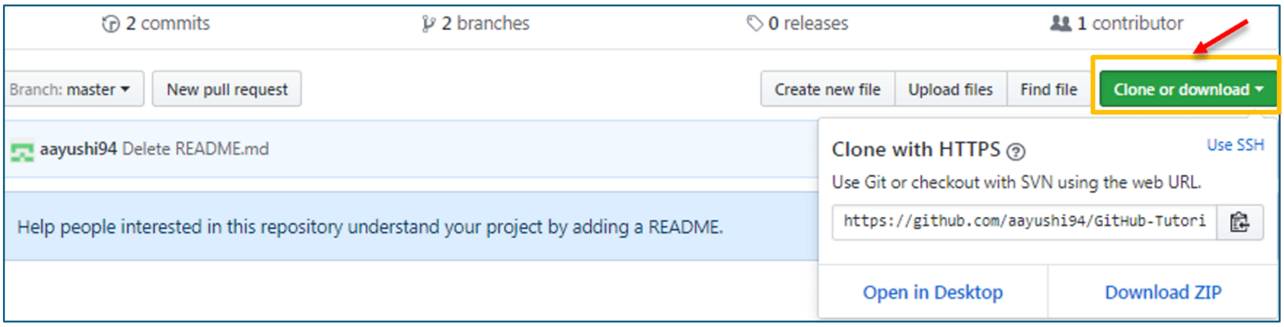
First merge request!

AayushiJohari merged commit `4f08be9` into `master` just now

Pull request successfully merged and closed
You're all set—the `readme--changes` branch can be safely deleted.

Step 4: Cloning and Forking GitHub Repository

Cloning: Before I actually talk about cloning a GitHub repository, first let us understand why do we need to clone a repository. The answer is simple! Suppose you want to use some code which is present in a public repository, you can directly copy the contents by cloning or downloading. Refer to the below screenshot for a better understanding.



2 commits | 2 branches | 0 releases | 1 contributor

aayushi94 Delete README.md

Help people interested in this repository understand your project by adding a README.

Clone with HTTPS
Use Git or checkout with SVN using the web URL.
https://github.com/aayushi94/GitHub-Tutori

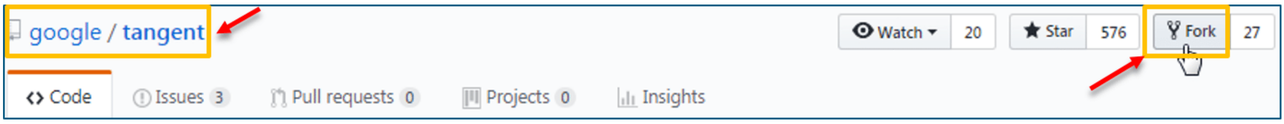
Forking: First, let us talk about why do we need forking. Suppose, you need some code which is present in a public repository, under your repository and GitHub account. For this, we need to fork a repository.

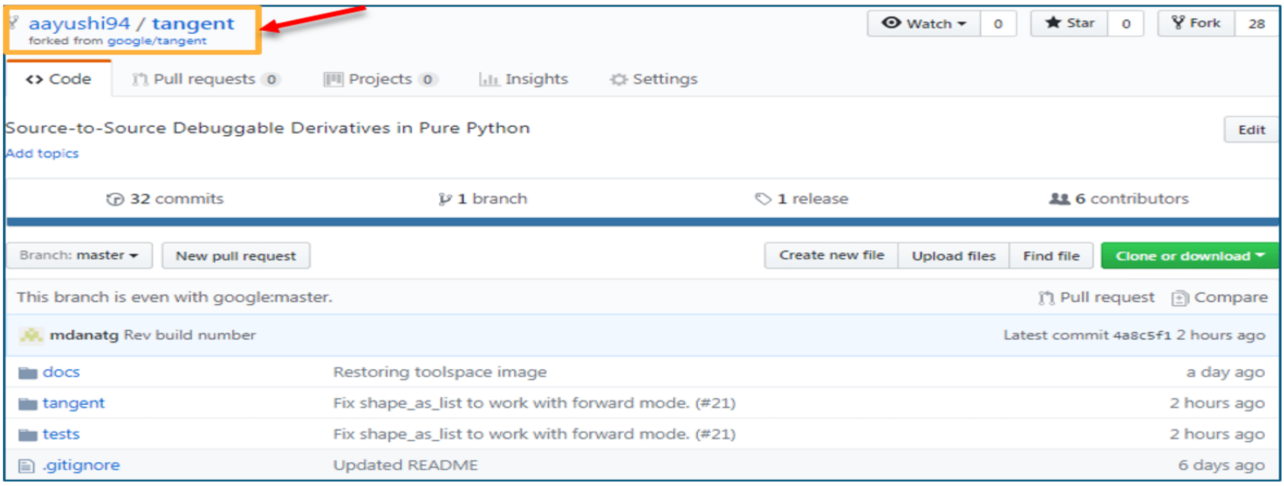
Before we get started with forking, there are some important points which you should always keep in mind.

- Changes done to the original repository will be reflected back to the forked repository.
- If you make a change in forked repository, it will not b reflected to the original repository until and unless you have made a pull request.

Now let’s see how can you want to fork a repository. For that, follow the below steps:

- Go to Explore and search for public repositories.
- Click “fork”. Note that this “tangent” repository is already forked 27 times and it is under “google” account. Refer the below image for better understanding.





Source-to-Source Debuggable Derivatives in Pure Python

This branch is even with google:master.

mdanatg Rev build number — Latest commit 4a8c5f1 2 hours ago

docs	Restoring toolspace image	a day ago
tangent	Fix shape_as_list to work with forward mode. (#21)	2 hours ago
tests	Fix shape_as_list to work with forward mode. (#21)	2 hours ago
.gitignore	Updated README	6 days ago

Congratulations! You have successfully forked an existing repository under your own account.

4.0 HARDWARE & SOFTWARE REQUIREMENTS

Hardware Requirements:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. The hardware requirements required for this project are:

- Intel i5
- 8GB RAM

Software Requirements:

Software Requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application.These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed. The software requirements that are required for this project are:

- Android Studi0
- Cloud Storage for firebase

5.0 CONCLUSION

This project helps us to save storage and we are able to share photos with known persons securely with higher security.

REFERENCES

1.https://www.edureka.co/blog/what-is-devops

2.https://www.javatpoint.com/devops-vs-agile

3.https://www.guru99.com/devops-tutorial.html#4

4.https://en.wikipedia.org/wiki/DevOps

5.https://github.com

6.https://guides.github.com/activities/hello-world7.

7.http://www.telusko.com

8. https://aws.amazon.com

9.https://www.tutorialspoint.com/git/git_environment.htm

10.https://www.javatpoint.com/history-of-cloud-computing