ASSIGNMENT 01

Sabaragamuwa University of Sri Lanka

Faculty of Computing

Software Engineering

SE4103

Software Configuration Management

Name	: I.K.A.D Thilakshana
Reg. No	: 19APSE4293
Academic Period	: 2nd Year 2nd Semester
Degree Program	: Software Engineering

1.GitHub

What is GitHub

GitHub is a platform for online software development. It is employed for software project collaboration, tracking, and archiving. It facilitates the sharing of code files and facilitates developer collaboration on open-source projects. In addition, coders can freely network, cooperate, and pitch their work on GitHub, which doubles as a social networking platform.

On GitHub, users may establish accounts, upload files, and develop coding projects. GitHub, however, truly comes to life when users begin collaborating. While it is possible for one person to write code, most development projects are produced by teams of people. These teams don't always work simultaneously; most of the time, they work in parallel. There are a number of issues while collaborating on projects with remote teams. GitHub greatly streamlines this procedure in several different ways.

Create New Repository file

Choose "New repository," then provide the necessary data, including the repository's name and description. You have the option to initialize the repository with a README file and to select between public and private visibility. Click "Create new file," give your file a name, then fill in the text field with content when the repository has been created.

Readme File

A README file, typically found at the repository's root, is a project manual providing essential details on the project's goals, application, installation, contribution instructions, license, and other relevant information, facilitating user and collaborator interaction.

Git Commands

1. git init

   creates a hidden folder called.git and initializes a new Git repository in the current directory. This folder holds the internal data structure needed for version control.

2. git status

   displays your working directory's current state while emphasizing modifications made to uncommitted files.

3. git commit -m "message

   Adds modifications to the repository and includes a detailed commit note. This image captures a moment in the project's past.

4. git branch

   provides a list of every branch in your repository. With an asterisk, the current branch is identified.

5. git checkout -b branch_name

   establishes and switches to a new branch (branch_name). This combines the features of git checkout and git branch.

6. git remote add origin git@...

   links your local repository to a distant repository called "origin" that is usually hosted on GitHub. The remote repository's location is specified by the URL.

7. git push -u origin main

   installs tracking and pushes your changes to the remote repository (in this example, "origin"). A connection is made between your local branch and the matching remote branch by using the -u parameter.

8. git merge command

   combines modifications from one branch with another. Git merge branch_name, for instance, would combine branch_name's modifications with the current branch.

9. git pull

   Changes are fetched and merged into the current branch from a remote repository. It is a git fetch + git merge combo.

10. git rebase --continue

    When disputes are resolved, carries out the rebase procedure. You can apply your modifications again on top of a different branch.

11. git remote add upstream

    includes the remote repository "upstream." When you fork a repository, this is frequently used, while "upstream" refers to the original repository from which you forked.

12. git fetch upstream

    Retrieves changes from the "upstream" remote repository but does not automatically merge them into your local branch.

Issues

Problems, improvements, tasks, or inquiries that are monitored in an issue tracking system are referred to as issues in software development and project management.

pull request

A user submits a pull request (PR), which is a suggested set of modifications. It proposes merging modifications into another branch (typically the main or master branch) from one branch (usually a feature branch).

Resolve Conflicts

Before the pull request is merged, conflicts arising from modifications that clash between the source and target branches must be resolved. When the identical lines of code are modified in both branches, conflicts may result.

Review Changes

Team members and collaborators may evaluate pull requests. This entails examining the code revisions, making sure the suggested adjustments accomplish the desired results, and confirming compliance with coding standards.

Accept Changes

After the changes have been reviewed and any conflicts resolved, the pull request can be "approved" or accepted. This indicates that the recommended changes have been accepted and are ready to be merged.

Merge Pull Request

The modifications from the feature branch are merged into the target branch using a pull request. The project's incorporation of the suggested changes is completed with this action.

SSH

A cryptographic network protocol called Secure Shell, or SSH, is used to communicate securely over unprotected networks. It offers a safe method of remote device or server access and management. SSH encrypts communication between the client and the server to guarantee data integrity and confidentiality. Because it enables users to safely run commands and log in to distant workstations, it is commonly used in system management. SSH is also often used for tunnelling and secure file transfers, which makes it an essential tool for guaranteeing the security of distant connections in a variety of computer contexts.

GitHub Actions

GitHub offers an automation platform called GitHub Actions. With the use of YAML files, you can create workflows that automate processes like building, testing, and deploying your code right within your GitHub repository.

main.yml and static.yml

These YAML files serve as samples of how processes are defined in GitHub Actions. Your project's primary workflow may be defined by main.yml, whilst the process for managing static files may be specified by static.yml.

Deploy

The process of releasing and making your code or programme usable, usually on a server or hosting platform, is called deployment.

GitHub Pages

One of GitHub's features, GitHub Pages, lets you host static web pages straight from your repository. It makes the process of releasing information on the web easier.

CI/CD Pipeline

A collection of procedures and guidelines known as continuous deployment and delivery (CD/CI) and continuous integration (CI) are intended to automate and optimise the software development and release process. While CD automates the process of sending such changes to production, CI concentrates on continuously integrating code updates.

Jenkins

An open-source automation server called Jenkins is used to develop, test, and release code. It provides a large selection of plugins for many integration scenarios and facilitates the construction of CI/CD pipelines.

Spinnaker

A multi-cloud, open-source continuous delivery platform is called Spinnaker. With capabilities for rolling deployments, canary releases, and simple rollback options, it is intended to assist with the deployment and management of applications across various cloud environments. Because Spinnaker can integrate with several cloud providers, it may be used in a variety of difficult deployment situations.

Shell Scripting

Shell scripting is the process of automating processes and operations on a computer system by creating and running commands via a command-line interface. These scripts automate procedures, manage setups, and simplify repetitive activities on Linux and Unix operating systems by using scripting languages like Bash. Shell scripting is a useful tool for improving workflow efficiency in development and system management.

Shell File

A shell script, often referred to as a shell file, is a text file that contains a set of commands written in a shell-compatible scripting language. It enables the sequential execution of commands and the automation of tasks.

sh run.sh(run a shell script file)

This programme uses the sh command to launch a shell script file called "run.sh". When the script is run, a series of instructions are carried out one after the other.

Commands

In shell scripting, commands refer to directives or tasks that the script is supposed to carry out. Common commands are used to work with files, output data, calculate, and communicate with the system.

Clear

The clear command creates a tidy and uncluttered interface by clearing the terminal screen.

Echo

To print text or variables to the terminal, use the echo command. Scripts frequently utilise it to show messages.

Date

The current date and time are retrieved and printed using the date command.

Expr

In shell scripts, the expr command is used to evaluate expressions and carry out mathematical operations.

Sleep

The sleep command allows script delays by pausing the script's execution for a certain amount of time.

read

When a script is running, the read command is used to capture user input.

Variables

Shell programming uses variables to handle and save data. They can be given values and be utilized by the script to store data for later use.

Functions

In shell programming, functions are code blocks that carry out certain operations. They aid in the modularization of scripts and improve the readability and maintainability of code.

If then else condition

In shell scripts, conditional branching is achieved by using the if, then, and else constructs. They let programmes decide what to do depending on predetermined parameters.

while do done

In shell scripting, a loop is created using the while, do, and done components. The loop's commands will be carried out for as long as the given condition holds true.

Command line color change

Using escape sequences or terminal commands to change the way text appears on the command line, emphasize important information, or provide visual clues are all part of changing the colour of command line text. This is frequently used to improve scripts' or terminal apps' user interfaces.

Docker

What is docker

A platform called Docker is used to create, ship, and operate programmes inside of containers. Applications may operate well in a variety of computer environments because to containers' lightweight, uniform environment.

Vmware

a software framework created by VMware Inc. that enables the use of many operating systems on a single physical machine.

virtual box

An open-source hypervisor for x86 computers currently being developed by Oracle Corporation.

Containerization

Code, libraries, environment variables, configuration files, and other components required to operate an application are all included in these small, independent, executable software packages.

Docker availability

a cloud-based registry that Docker offers, allowing users to cooperate, exchange, and save container images.

How to install docker

Installing Docker involves downloading and installing the Docker Engine for the computer. Installation instructions are available in the official Docker manual and vary depending on the operating system.

Docker hub

a cloud-based registry that Docker offers, allowing users to cooperate, exchange, and save container images.

Alpine

Alpine Linux is a security-focused, lightweight Linux distribution that is frequently used as the Docker container base image. Because to its compact size, it is well-liked for producing simple yet effective container pictures.

Postgres

Open-source PostgreSQL, sometimes known as Postgres, is a potent relational database management system. Docker is frequently used to containerise it, making database instance deployment and maintenance simple.

Docker commands

Docker images

The locally accessible Docker images may be listed using the Docker images command. You may use docker run to launch a container with the PostgreSQL image name, such as docker run postgres, in order to utilise the image.

Docker run

A new container is created and started using the docker run command, which uses an image as its basis.

docker ps

A list of all active containers is displayed by docker ps, together with information about each one, including its ID, name, and status.

docker run –td

With this command, a container is launched in detached mode (-d), freeing up the terminal for other commands (-t for terminal) to execute in the background.

Clear

To create a tidy interface, use the clear command to empty the terminal screen.

Docker stop

A running container may be stopped with the docker stop command, enabling a smooth shutdown.

Docker exec –it

To run commands in an active container, use docker exec. The -it option switches on interactive mode, which lets you communicate directly with the container's terminal.

#ls

The container's current directory's files and folders are listed using the ls command.

git –version

The Docker container's installed Git version may be seen by running git --version.

docker rm

One or more halted containers are eliminated using docker rm. Containers that are no longer needed are cleaned.

docker rmi

One or more Docker images can be removed with docker rmi to free up storage on the host computer.

docker in docker

Docker in Docker (DinD) is the term used to describe running Docker containers within another Docker container. This approach is used in some circumstances where multilayer containerisation is required, usually for testing or continuous integration/delivery requirements.

Python Image

"Python image" probably means a Docker image that has the Python runtime environment attached to it. Installing the Python interpreter and necessary tools in a Docker image requires utilizing package managers like apt (for Debian-based images) or apk (for Alpine-based images).

Docker file commands

- FROM - Specifies the base image for the new image
- COPY - Copies files or directories from the host machine to the image.
- CMD - Specifies the default command to run when a container is started
- VOLUME - Creates a mount point for externally mounted volumes
- RUN - Executes commands during the image-building process.

Readme.md

A markdown file called readme.md is usually placed in a project's repository to offer information and documentation about the project. It frequently contains instructions on how to utilise, set up, and participate in the project.

Docker Build -t

The command to create a Docker image from a Dockerfile is docker build -t. You can annotate a picture with a name and, if desired, a version with the -t parameter.

.dockerignore

A file called.dockerignore lists the directories and files that should not be included in the Docker build context. It enhances build performance by assisting in reducing the amount of context given to the Docker daemon.

apk add update python3 py3-pip

Using the apk package manager, this programme installs or updates Python 3 and pip (Python package installer) for Alpine Linux.

Docker Compose

A tool for creating and executing multi-container Docker applications is called Docker Compose. It enables you to start or stop the complete application stack with a single command and configure the services, networks, and volumes of your application in a single docker-compose.yaml file.

Docker-compose.yaml

a YAML configuration file that defines and controls multi-container Docker applications using Docker Compose. It defines two services in this instance, "postgres" and "pgadmin4".

Docker compose up

Start all containers specified in the docker-compose.yml file using this command. Containers are created and started for every service that is listed in the file.

Docker compose down

the command to erase every storage, network, and container that Docker Compose started. When you have resources that you no longer require, this is helpful for clearing things up.

Docker compose stop

The docker-compose.yml file's stated commands will all running containers to be gracefully stopped. This guarantees the preservation of any data written during shutdown until the procedure concludes.

Docker network ls

the command to display a list of all local networks that are accessible. Networks are used to link containers across servers or inside a swarm.

pgAdmin

a graphical solution for PostgreSQL database management that lets administrators control their databases without utilizing SQL commands by using a user interface.

Kubernetes

Overview: Kubernetes is an open-source technology for container orchestration, also referred to as K8s.

Container Orchestration: It offers a stable environment for executing distributed systems by automating the deployment, scaling, and maintenance of containerised apps.

Kubernetes Provides

1. **Service discovery and load balancing**

   A container can be made accessible by Kubernetes using either a distinct IP address or a DNS name. Kubernetes may distribute and balance traffic to a container that receives a lot of it, keeping the application responsive and reliable.

2. **Storage orchestration**

   You may use Kubernetes to automatically mount any kind of storage system, including public cloud services, local drives, and more.

3. **Automated rollouts and rollbacks**

   You may specify the intended state of your containers with Kubernetes, and the system will automatically modify the current state to conform to your specification. For example, it can effortlessly migrate resources from old to new containers, create new ones, and delete old ones—all under your control.

4. **Automatic bin packing**

   To enable Kubernetes to carry out containerised tasks, you provide it a cluster of nodes. You provide to Kubernetes how much RAM and CPU power each container requires.

5. **Secret and configuration management**

   Passwords and authentication credentials are only two examples of the sensitive data that Kubernetes helps you handle and store securely. This data is safe and independent from your application settings, and you may update it without affecting your container setup.

6. **IPv4/IPv6 dual-stack**

   Allocation of IPv4 and IPv6 addresses to Pods and Services

Advantages

- Easily scale applications up or down
- keeps the application available while having self-healing properties.
- functions in a variety of on-premises and cloud settings.

Declarative Configuration - specifies the system's intended state, and Kubernetes makes sure the system operates in that manner.

Community and Ecosystem - backed by a thriving community and boasting an extensive tool and extension ecosystem.

Use Cases - extensively utilized for effective containerised application deployment, management, and scalability.

Cluster

A collection of real or virtual computers running containerised apps under Kubernetes management is called a Kubernetes cluster. It is made up of two primary parts,

1. Master Node - oversees and directs the cluster. In addition to scheduling and coordinating application deployment, it keeps an eye on the cluster's general health and reacts to various events, such as scaling up or starting up new apps.
2. Worker Nodes (Minions) - These are the real computers that are used to deploy and operate containers. Every worker node interacts with the master to carry out its designated responsibilities.

Pods

In Kubernetes, a pod is the smallest deployable unit. It wraps one or more containers and represents a single instance of a running process inside a cluster. Because they are contained in the same network namespace, containers in a pod can connect with one another via localhost.

BackUp

Assets - Any useful data, equipment, or other part of an organization's systems is considered an asset

What is backup

Making a copy of your system's data to use as a backup in case the original is lost or corrupted is known as data backup. You can also use backup to recover copies of older files if you have deleted them from your system. One of the main elements of a company's disaster recovery plan and business continuity strategy is backup, which is used by many businesses and organizations to safeguard their important data.

Importance of backup

- Data Loss Prevention
- For Business Continuity
- Protection Against Cyber Threats
- To recover past and historical data
- To recover from different hardware failures

Data backup solutions

1. **Hardware appliances -** Often included with these appliances is storage, which is a 19" rack-mounted appliance that you install and link to your network. Installing and configuring the machines is simple. Most of the time, installing software or setting up a separate server or operating system is not necessary.

2. **Software solutions -** Software solutions manage the backup procedure and are installed on your personal computers. While many solutions let you utilize your current systems, others need dedicated servers that are only used as

backups. You must install and set up the operating system and backup software for them.

3. **Cloud services -** Backup-as-a-service (BaaS) is a cloud-based solution that several companies offer. By installing small agents on your computers, you may use the vendor's or service provider's cloud infrastructure to schedule and manage your backups. Because there are no operating systems to configure or systems to deploy, the BaaS is even more straightforward than software.

4. **Hybrid data backup solutions -** An all-in-one hybrid backup solution that lets you install the software or use it as a cloud service is the newest development in the backup industry. For many organizations, these solutions are the ideal option since they integrate the finest aspects of both worlds.

Testing

There are 3 type of testing

1. Unit Testing
2. Integration testing
3. End to end testing

Unit Testing

- Unit testing is a technique that involves testing separate software application units or components on their own. It seeks to confirm that every software component operates as intended.
- isolates and mocks external dependencies while concentrating on a single function, method, or module.
- finds issues early in the development cycle and makes sure the software works properly throughout.

Integration Testing

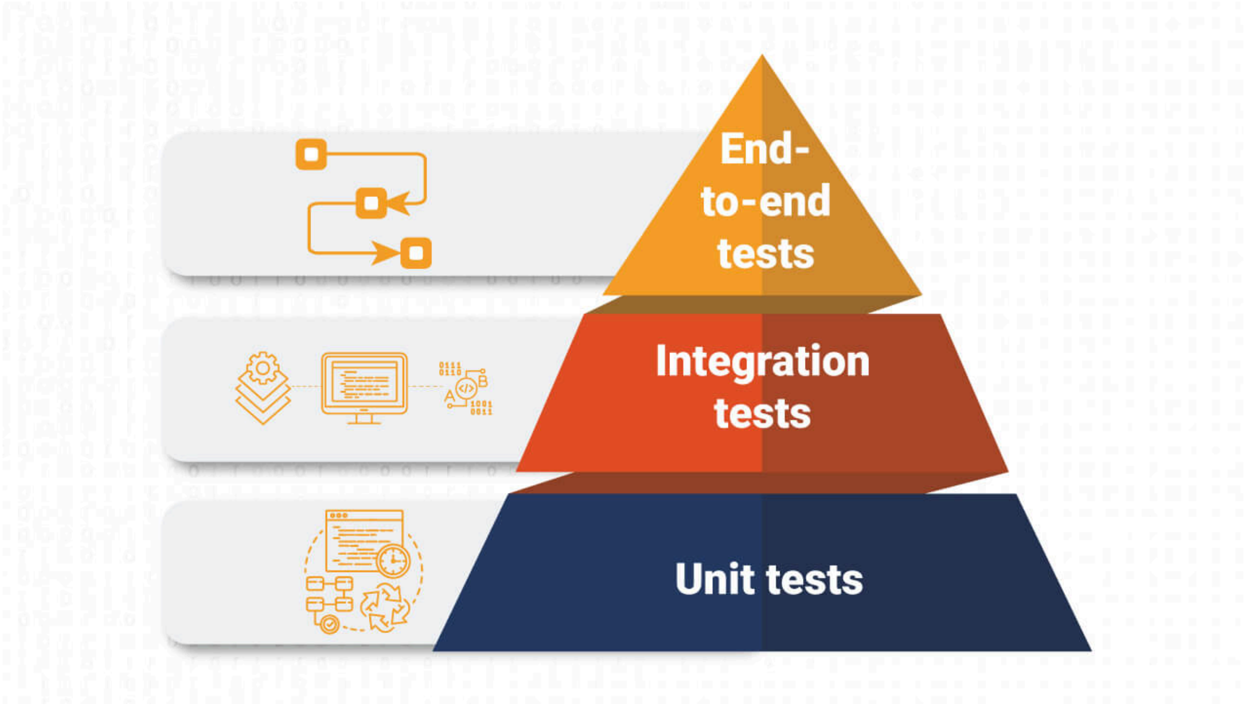
- Testing interfaces and interactions between various software components or systems to make sure they function as a unit is known as integration testing.
- looks into the connections between modules or units to ensure that data is flowing and communicating properly.

- checks sure integrated components are operating correctly and identifies any problems that could occur when they interact.

End to End Testing

- End-to-end testing assesses a software application's whole functioning from beginning to conclusion. It mimics actual user interactions and circumstances.
- involves evaluating all system components as well as how they interact with one another.
- identifies problems that could occur throughout an entire workflow by validating the system's behavior and performance in an actual setting.

Test pyramid



A visual paradigm that supports a well-rounded testing approach is the test pyramid. It promotes having more unit tests at the base, a reasonable amount of integration tests in the middle, and less end-to-end tests at the top of a pyramid-shaped structure. The time and resources required for testing are decreased by this structure, which encourages efficiency in locating and fixing problems at the lowest level of granularity. The test pyramid guarantees broad test coverage across several application levels by prioritizing unit tests for individual components, integration tests for component interactions, and comprehensive end-to-end tests for full system validation.

Test Automation: Test automation is the practice of running tests more quickly and reliably by utilizing automated tools and scripts.

Test Frameworks: Software frameworks known as "test frameworks" facilitate the creation and execution of tests by offering an organized method for doing so.

Unit Test with “jest”

Facebook created the JavaScript testing framework Jest. JavaScript code, particularly those created using well-known libraries and frameworks like React, Vue, and Angular, is frequently tested with it. Jest offers an effective and developer-friendly testing environment that is easy to set up and utilize.

To make sure that individual units or components perform as intended, you follow a disciplined process while developing unit tests using Jest.