Exercise 8

Task 1 : Describe the difference between a virtual machine and a Docker container.

Virtual Machine (VM)	**Docker Contanier**
- A VM emulates an entire hardware system, allowing you to run a full operating system (guest OS) on top of another operating system (host OS) using a hypervisor (e.g., VMware, VirtualBox, Hyper-V). **Architecture:** - Hardware - Host OS - Hypervisor - Guest OS (includes its own kernel) - Application - **Resource use:** Heavier; each VM carries its own OS, consuming more memory and storage. - **solation:** Strong; VMs are isolated at the hardware level. - **Boot time:** Slower, can take minutes to start. - **Use case:** Best when you need to run completely different OSes on the same hardware, or require strong isolation, e.g., running Linux and Windows workloads on the same server.	- A container shares the host OS kernel but isolates the application process environment using Linux features like namespaces and groups. **Architecture:** - Hardware - Host OS + Container Runtime (e.g., Docker Engine) - Containers (each has its own filesystem, libraries, dependencies, but **shares the host kernel**) - Application - **Resource use:** Much lighter; containers don't carry a full OS, leading to lower overhead. - **Isolation:** Process-level isolation, good but generally not as strong as VMs. - **Boot time:** Very fast, usually milliseconds to seconds. - **Use case:** Ideal for deploying microservices, scalable cloud apps, and quickly spinning up reproducible dev/test environments.

Task 2: What is the Main purpose of Continuous Integration ?

a) automatically deploy

b) merge all code changes into production every hour

c) frequently integrate code into a shared repository and run automated tests
d) build Docker containers for each feature

The right choice is: c) frequently integrate code into a shared repository and run automated tests
The primary goal of Continuous Integration (CI) is to enable developers to frequently incorporate their modifications into a common codebase, often several times per day. Each integration triggers automated builds and tests to catch issues early, minimize integration conflicts, and enhance overall code quality.

Task 3: in IntelliJ

Task 4: Which tool is commonly used to orchestrate CICD pipelines ?

a) Kubernetes

b) Jenkins

c) Ansible

d) Terraform

The correct answer is: b) Jenkins

Jenkins is among the most popular tools for managing CI/CD pipelines. It streamlines the processes of building, testing, and deploying applications, and offers a wide range of plugins to integrate with various other tools.

Task 5: The testing stage in your CI pipeline takes very long and causes timeouts. Name strategies to speed it up.

Strategies to Speed Up CI Testing:

1. **Parallel Testing**
   Split your test suite to run tests simultaneously across multiple machines or containers.
2. **Run Only Relevant Tests**
   Execute tests related only to the changed code (test impact analysis) instead of the entire suite.
3. **Optimize Test Code**
   Refactor slow tests and remove or fix flaky ones. Use mocks or stubs to replace slow external services.

4. **Use Faster Testing Frameworks**
   Switch to testing tools or frameworks known for quicker execution times.
5. **Cache Dependencies and Test Results**
   Cache libraries, build artifacts, or test outputs to avoid redundant work in subsequent runs.