Task 1:

To set up a CI/CD pipeline for a Java application using Jenkins, I followed a structured approach. First, I created three separate Jenkins jobs: one for compiling the source code (mvn compile), another for testing the code (mvn test), and a third for packaging the code (mvn package). These jobs were configured to execute their respective Maven commands. Next, I set up a CI/CD pipeline that orchestrates the execution of these jobs in sequence. This pipeline ensures that the code is compiled, tested, and packaged automatically upon changes in the source repository. Additionally, I configured a master-slave node setup within Jenkins to distribute the workload across multiple nodes, optimizing build times and resource usage. The master node orchestrates the tasks, while the slave nodes perform the actual build steps.

Commands for Job Configuration:
1. Compiling Job:

shell
    Copy code
    mvn compile
    Jenkins Job Configuration:

Name: Compile Job
Build Step: Execute shell
Command: mvn compile
2. Testing Job:

shell
    Copy code
    mvn test
Jenkins Job Configuration:

Name: Test Job
Build Step: Execute shell
Command: mvn test
3. Packaging Job:

Shell
    Copy ABC Technologies
    mvn package
Jenkins Job Configuration:

Name: Package Job
Build Step: Execute shell
Command: mvn package

CI/CD Pipeline Setup
Jenkinsfile:
Look up the groovy file in Git

Master-Slave Node Configuration:
Navigate to Manage Jenkins -> Manage Nodes and Clouds.
Add a new node (slave) and configure its properties (e.g., remote root directory, labels, etc.).
Ensure the master node delegates tasks to the slave nodes based on the labels and job configurations.

Task 2

Run the following command to build the docker image
docker build -t ABC-App .
To run the docker application, run the following command
docker run -d -p 8080:8080 ABC-App

Task 3

Step 1: Integrate Docker Host with Ansible
First, ensure Ansible is installed on your control machine. Then, configure your Docker host in the Ansible inventory file.

Step 2: Write Ansible Playbook to Create Image and Container
Create an Ansible playbook to build a Docker image and run a Docker container.

Ansible Playbook (docker-playbook.yml):

Step 3: Integrate Ansible with Jenkins
In Jenkins, install the Ansible plugin and configure it under Manage Jenkins -> Global Tool Configuration.
Jenkins Pipeline (Jenkinsfile):

Step 4:
Deploy Artifacts on Kubernetes using Ansible Folder where the configuration file for Ansible and Kubernetes deploy reside

Step 5: Integrate Kubernetes with Ansible
Ansible Playbook to Create Deployment and Service (k8s-playbook.yml) and update it with Jenkins deployment

Step 6: Update Jenkins Pipeline for Kubernetes Deployment
Updated Jenkins Pipeline (Jenkinsfile)