SRE TRAINING (DAY 13) - PYTHON JENKINS CI/CD PIPELINE PROJECT

Project Objective

Bhoomikaushik's CI/CD pipeline successfully:

* Builds a Python wheel package
* Runs automated unit tests
* Creates and runs a Docker container
* Deploys the application automatically upon successful pipeline execution

Project Implementation Stages

STAGE 1 - Project Initialization

Bhoomikaushik:

* Created a dedicated project directory for the Python application
* Initialized a Git repository to track project changes and enable version control

STAGE 2 - Python Project Configuration

Bhoomikaushik:

* Created a pyproject.toml file defining build system requirements using hatchling as the build backend
* Specified comprehensive project metadata including name, version, and dependencies

STAGE 3 - Adding Source Code & Test Cases

Bhoomikaushik:

* Implemented the main Python function that outputs a message
* Wrote thorough unit tests using Python's unittest framework to verify the main function's output

STAGE 4 - Dockerization of the Project

Bhoomikaushik:

* Created a Dockerfile to containerize the Python application
* Configured Docker to install the wheel file and run the application using the defined entry point

STAGE 5 - Jenkins Pipeline Configuration

Bhoomikaushik:

* Created a Jenkinsfile defining the complete pipeline stages:
  + Checkout: Pulled the latest code from the Git repository
  + Build Wheel: Built the Python wheel file
  + Test: Installed pytest and executed automated tests
  + Build Docker Image: Created a Docker image from the Dockerfile
  + Deploy: Stopped any existing Docker container and deployed the new one

Key Challenge Overcome:

* Problem: Encountered externally-managed-environment error while installing Python packages inside the Jenkins virtual environment
* Resolution: Bhoomikaushik implemented a workaround using the --break-system-packages flag

STAGE 6 - Docker Image Build & Deployment

Bhoomikaushik:

* Built the Docker image from the application's wheel file
* Implemented container management by stopping any previously running Docker containers before deploying the new version

STAGE 7 - Setting up Pipeline and Building in Jenkins

Bhoomikaushik:

* Installed all required Jenkins plugins - Pipeline, Git, and Docker pipeline
* Created and configured a new Pipeline Project
* Connected to the Git repository
* Added the Jenkinsfile to the project
* Successfully ran the Pipeline (Build)