

1. Jenkins trigger configuration settings have been updated to use “Poll SCM”. With this configuration Jenkins will check the public repository every 2 minutes to detect changes and trigger the pipeline:

Triggers

Set up automated actions that start your build based on specific events, like code ch

- ☐ Build after other projects are built ?
- ☐ Build periodically ?
- ☐ GitHub hook trigger for GITScm polling ?
- ☒ Poll SCM ?

Schedule ?

H/2 * * * *

Would last have run at Sunday, November 23, 2025, 4:07:00 AM Coordinated U

- ☐ Ignore post-commit hooks ?
- ☐ Trigger builds remotely (e.g., from scripts) ?

2. A Jenkinsfile was created, and the pipeline was updated to reference the Jenkinsfile included in the public repository:

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

`https://github.com/EdselCaprice/chatbot_CSC7435/`

Credentials ?

- none -

Advanced ▼

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

`*/master`

Add Branch

3. Enhancements to pipeline:

- a. Update the build stage to use Docker. Previously this stage only included skeleton code. This stage will catch all build errors prior to testing and deployment stages.

- b. Update the test stage to implement test.py to validate backend functionality prior to deployment. This will prevent app breaking changes from being deployed into production.
- c. Deploy with Docker. "Docker compose down" stops old containers to prevent conflicts. "docker compose up --build -d" rebuilds with latest code changes and ensures we are using the code that was tested in the previous stage. Lastly "docker system prune -f" cleans up the pipeline environment.

```
stages {  
  stage('Build') {  
    steps {  
      echo "Building.."   
      sh '''  
        echo "doing build stuff..."  
        docker compose build  
      '''  
    }  
  }  
  stage('Test') {  
    steps {  
      echo "Testing.."   
      sh '''  
        echo "Running backend tests..."  
        docker compose run --rm backend python test.py  
      '''  
    }  
  }  
  stage('Deliver') {  
    steps {  
      echo 'Deploying with Docker Compose..'   
      sh 'docker compose down'  
      sh 'docker compose up --build -d'  
      sh 'docker system prune -f'  
    }  
  }  
}
```