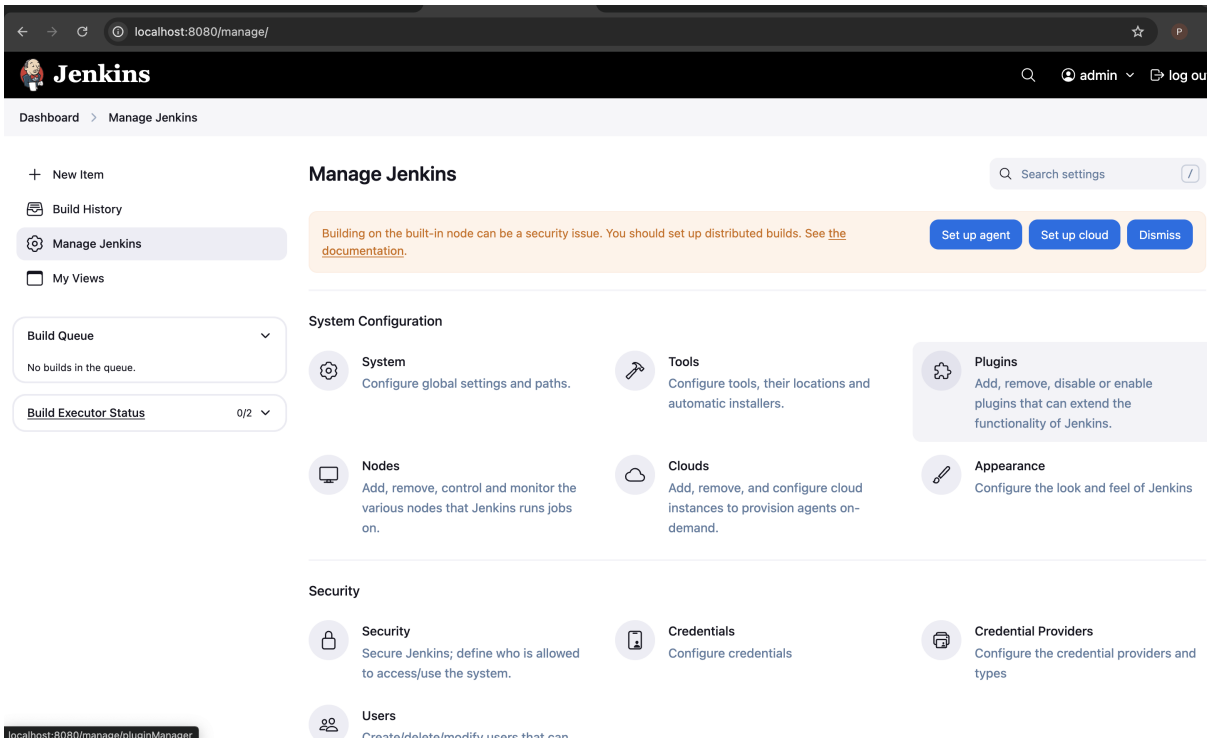
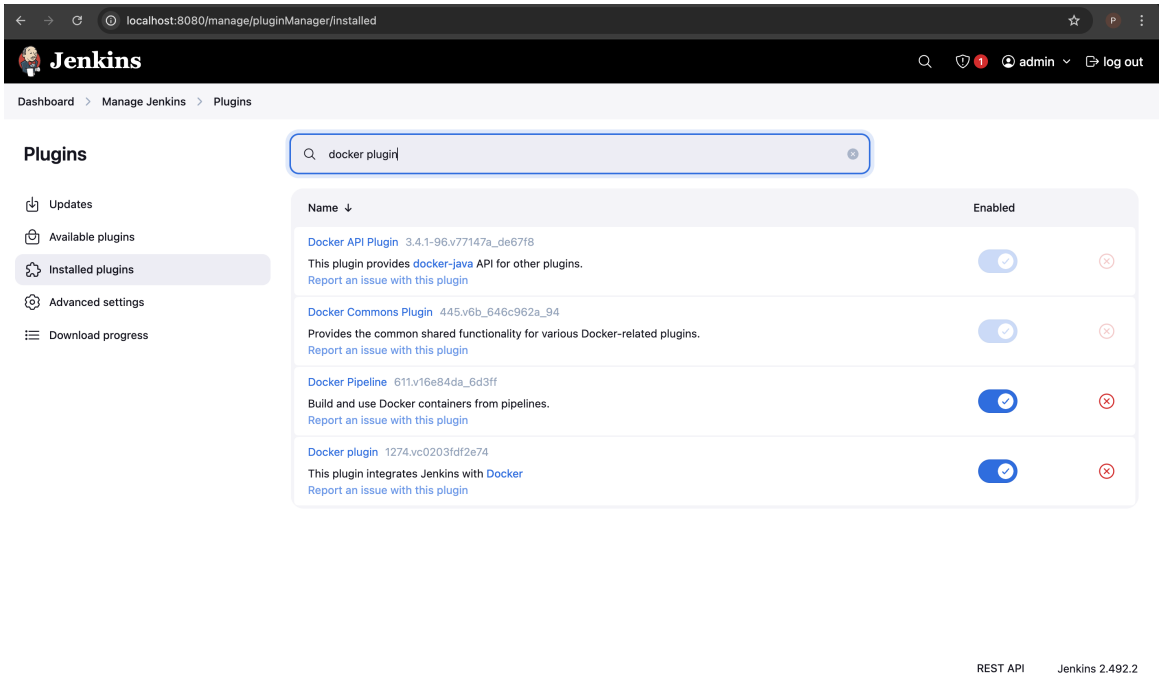
In Class: Jenkins Tour

Aarsh Sheth || Pooja Sindham || Shivani Jariwala || Aishwariya Indi

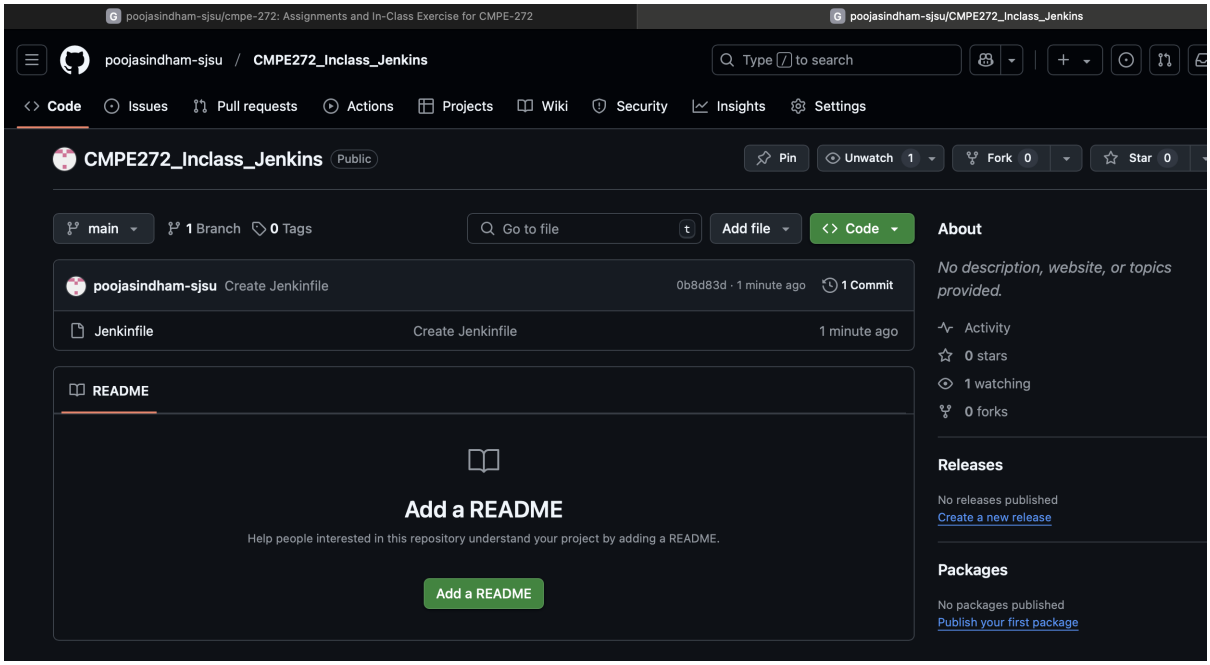
Step 1: Creating pipeline

Install *docker* plugin

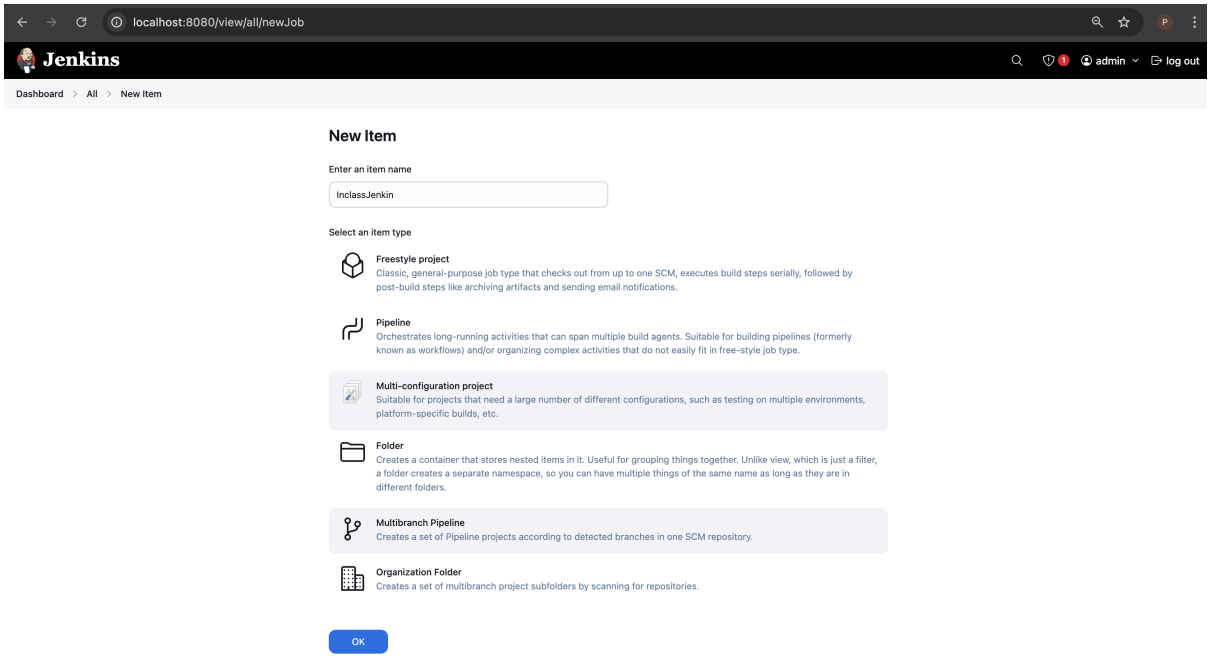




Create *JenkinFile* and upload to repo.



Create new item.



localhost:8080/job/InclassJenkin/configure

Jenkins

admin log out

Dashboard > InclassJenkins > Configuration

Configuration

- General
- Branch Sources
- Build Configuration
- Scan Multibranch Pipeline Triggers
- Orphaned Item Strategy
- Appearance
- Health metrics
- Properties

General

Enabled

Display Name

InclassJenkins

Description

Plain text Preview

Branch Sources

Single repository & branch

Allows a single, fixed branch of some repository to be configured. Not normally used, as it does not allow branches to be detected automatically. Might be used, for example, if you want to build all branches of one repository, *and* the `master` branch of a special clone.

Name

CMPE272

Source Code Management

Git

Repositories

Repository URL

https://github.com/poojasindham-sjsu/CMPE272_Inclass_Jenkins.git

Save Apply

localhost:8080/job/InclassJenkin/configure

Click to go back, hold to see history

Dashboard > InclassJenkins > Configuration

Configuration

- General
- Branch Sources
- Build Configuration
- Scan Multibranch Pipeline Triggers
- Orphaned Item Strategy
- Appearance
- Health metrics
- Properties

Git

Repositories

Repository URL

https://github.com/poojasindham-sjsu/CMPE272_Inclass_Jenkins.git

Credentials

- none -

+ Add

Advanced

Add Repository

Branches to build

Branch Specifier (blank for 'any')

*/main

Add Branch

Repository browser

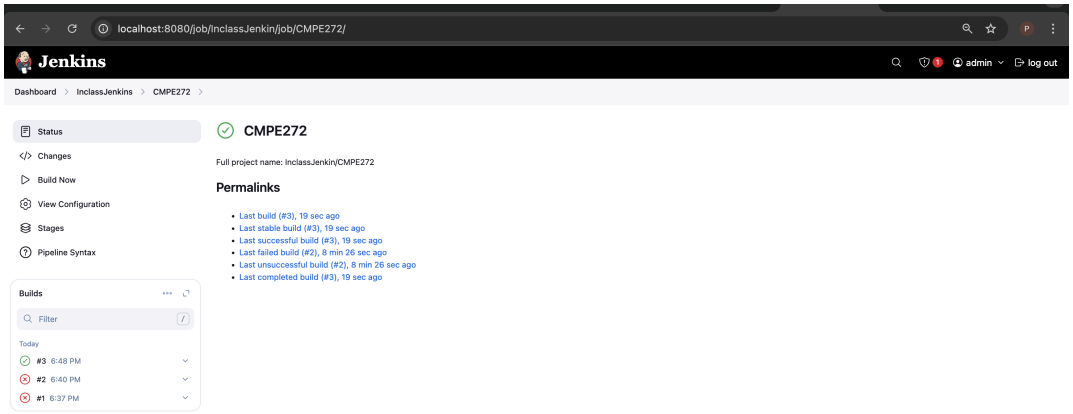
(Auto)

Additional Behaviours

Add

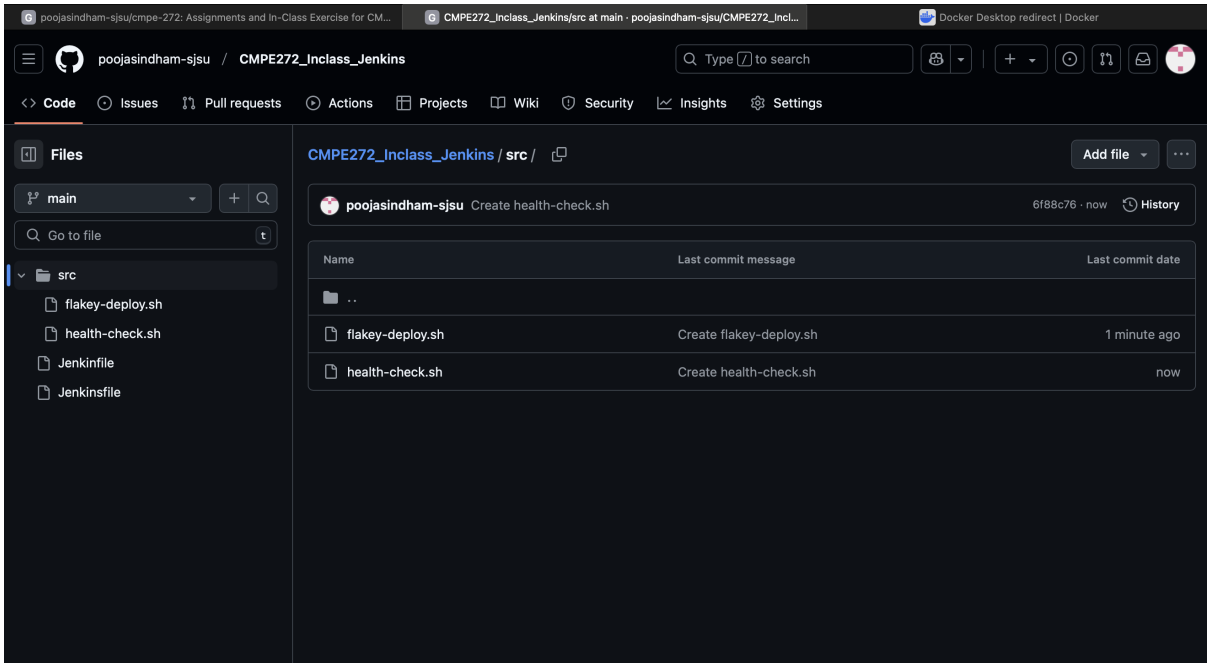
Save Apply

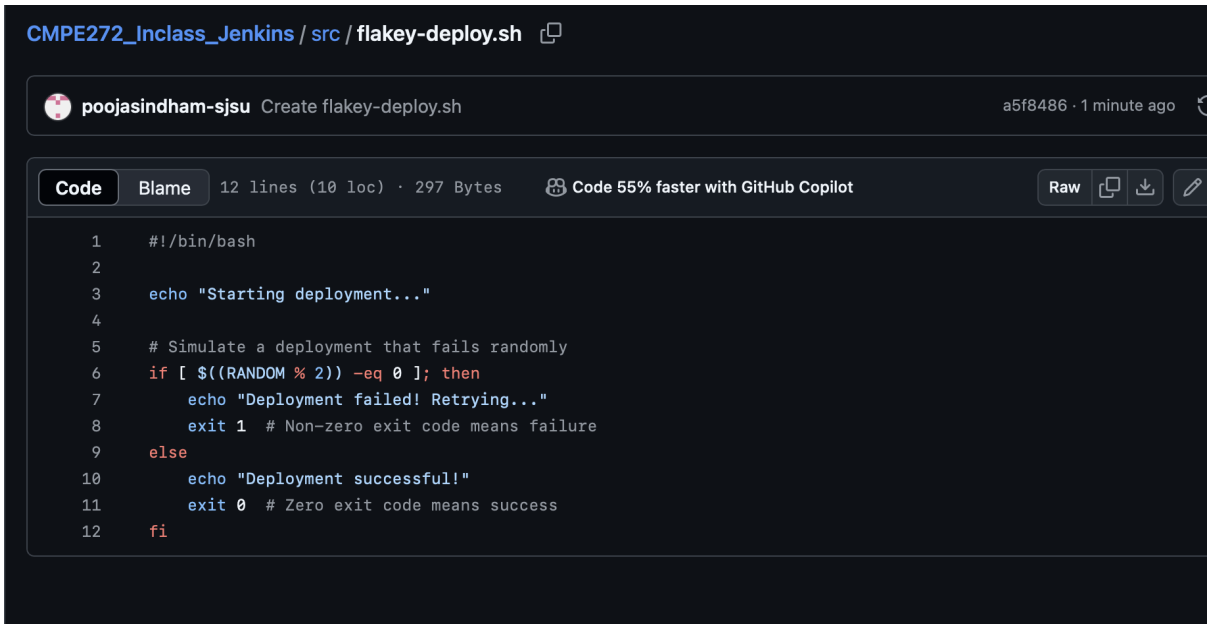
First pipeline ran successfully.



Step 2: Running multiple pipelines

Created two files *flakey-deploy.sh* and *health-check.sh* just for simulation.





CMPE272_Inclass_Jenkins / src / health-check.sh

poojasindham-sjsu Create health-check.sh 6f88c76 · 1 minute ago History

Code Blame 22 lines (17 loc) · 566 Bytes Code 55% faster with GitHub Copilot Raw

```bash
#!/bin/bash

echo "Running health check..."

# URL to check (Change this based on your actual service)
URL="http://localhost:8080/health"

# Try to get a 200 OK response within 10 seconds
for i in {1..10}; do
    STATUS_CODE=$(curl -s -o /dev/null -w "%{http_code}" $URL)

    if [ "$STATUS_CODE" -eq 200 ]; then
        echo "Health check passed!"
        exit 0
    else
        echo "Health check failed (Attempt $i)... retrying in 5s"
        sleep 5
    fi
done

echo "Health check failed after multiple attempts."
exit 1  # Fail if we never get a 200 response
```

Updated *JenkinsFile*

Code Blame 41 lines (37 loc) · 1.01 KB Code 55% faster with GitHub Copilot Raw

```bash
pipeline {
    agent any

    stages {
        stage('Deploy') {
            steps {
                retry(3) {
                    sh './src/flakey-deploy.sh'
                }

                timeout(time: 3, unit: 'MINUTES') {
                    sh './src/health-check.sh'
                }
            }
        }

        stage('Test') {
            steps {
                sh 'echo "Fail!"; exit 1' // This will fail intentionally
            }
        }
    }

    post {
        always {
            echo 'This will always run, no matter what'
        }
        success {
            echo 'Success! This runs only if the pipeline succeeds'
        }
        failure {
            echo 'Failure! This runs only if the pipeline fails'
        }
        unstable {
            echo 'Unstable! This runs only if the build is unstable (e.g., some tests fail but not the entire pipeline)'
        }
        changed {
            echo 'Changed! This runs if the pipeline's status changed from the last run'
        }
    }
}
```

Got the following error.

```
Commit message: "Update Jenkinsfile"
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] retry
[Pipeline] {
[Pipeline] sh
+ ./src/flakey-deploy.sh
/Users/poojasindham/.jenkins/workspace/InclassJenkin_CMPE272@tmp/durable-58c68839/script.sh.copy: line 1: ./src/flakey-deploy.sh: Permission denied
[Pipeline] }
ERROR: script returned exit code 1
Retrying
[Pipeline] {
[Pipeline] sh
+ ./src/flakey-deploy.sh
/Users/poojasindham/.jenkins/workspace/InclassJenkin_CMPE272@tmp/durable-5e87fe3b/script.sh.copy: line 1: ./src/flakey-deploy.sh: Permission denied
[Pipeline] }
ERROR: script returned exit code 1
Retrying
[Pipeline] {
[Pipeline] sh
+ ./src/flakey-deploy.sh
/Users/poojasindham/.jenkins/workspace/InclassJenkin_CMPE272@tmp/durable-13ddebf9/script.sh.copy: line 1: ./src/flakey-deploy.sh: Permission denied
[Pipeline] }
[Pipeline] // retry
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
Stage "Test" skipped due to earlier failure(s)
[Pipeline] getContext
```

Adding one more step of adding permissions.

CMPE272_Inclass_Jenkins / Jenkinsfile

poojasindham-sjsu Update Jenkinsfile

Code Blame 46 lines (42 loc) · 1.15 KB Code 55% faster with GitHub Copilot

```
pipeline {
    agent any

    stages {
        stage('Set Permissions') {
            steps {
                sh 'chmod +x src/flakey-deploy.sh src/health-check.sh'
            }
        }
        stage('Deploy') {
            steps {
                retry(3) {
                    sh './src/flakey-deploy.sh'
```

Multi pipeline ran Successfully.

Dashboard > InclassJenkins > CMPE272 > #8

```
Health check failed after multiple attempts.
[Pipeline] }
[Pipeline] // timeout
[Pipeline] }
ERROR: script returned exit code 1
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] catchError
[Pipeline] {
[Pipeline] sh
+ echo 'Fail!'
Fail!
+ exit 1
[Pipeline] }
ERROR: script returned exit code 1
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
This will always run, no matter what
[Pipeline] echo
Changed! Runs if the pipeline's status changed from the last run
[Pipeline] echo
SUCCESS: The pipeline completed successfully
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Dashboard > InclassJenkins > CMPE272 >

- Status
- Changes
- Build Now
- View Configuration
- Full Stage View
- Favorite
- Open Blue Ocean
- Stages
- Pipeline Syntax

Builds

Filter /

Today

- #8 7:22 PM
- #7 7:20 PM
- #6 7:15 PM
- #5 7:12 PM
- #4 7:07 PM
- #3 6:48 PM
- #2 6:40 PM
- #1 6:37 PM

CMPE272

Full project name: InclassJenkin/CMPE272

Stage View

	Declarative: Checkout SCM	Set Permissions	Deploy	Test	Declarative: Post Actions
Average stage times: (full run time: ~55s)	695ms	361ms	39s	206ms	156ms
#8 19:22 2 commits	752ms	367ms	52s	517ms	195ms
#7 19:20 1 commit	730ms	352ms	52s failed	112ms failed	171ms
#6 19:15 1 commit	579ms	368ms	52s failed	82ms failed	151ms
#5 19:12 1 commit	722ms	358ms	1s failed	115ms failed	109ms

Permalinks

- Last build (#8), 13 min ago
- Last stable build (#8), 13 min ago
- Last successful build (#8), 13 min ago
- Last failed build (#7), 15 min ago
- Last unsuccessful build (#7), 15 min ago

Step 3: Defining Execution Environments

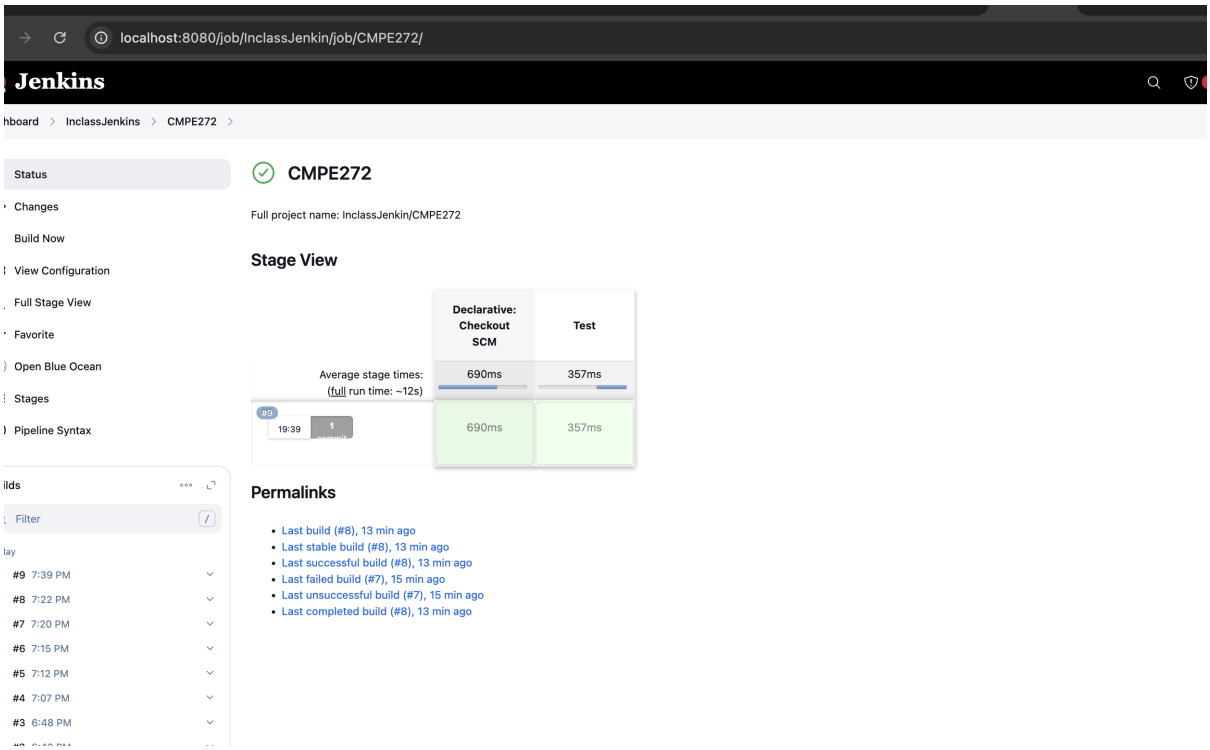
Updating Jenkinsfile for Execution environment.

CMPE272_Inclass_Jenkins / Jenkinsfile

poojasindham-sjsu step3 e2cb1d2 · now History

Code Blame 12 lines (12 loc) · 247 Bytes Code 55% faster with GitHub Copilot Raw

```
pipeline {
    agent {
        docker { image 'node:22.14.0-alpine3.21' }
    }
    stages {
        stage('Test') {
            steps {
                sh 'node --eval "console.log(process.arch,process.platform)"'
            }
        }
    }
}
```



Step 4: Using environment variables

CMPE272_Inclass_Jenkins / Jenkinsfile

poojasindham-sjsu step4 77cb332 · 1 minute ago History

Code Blame 20 lines (18 loc) · 372 Bytes Code 55% faster with GitHub Copilot Raw

```
pipeline {
    agent {
        label '!windows'
    }

    environment {
        DISABLE_AUTH = 'true'
        DB_ENGINE    = 'sqlite'
    }

    stages {
        stage('Build') {
            steps {
                echo "Database engine is ${DB_ENGINE}"
                echo "DISABLE_AUTH is ${DISABLE_AUTH}"
                sh 'printenv'
            }
        }
    }
}
```

CMPE272

Full project name: InclassJenkin/CMPE272

Stage View

	Declarative: Checkout SCM	Build
Average stage times: (full run time: ~2s)	597ms	395ms
#10 19:45 1 commit	597ms	395ms

Permalinks

- Last build (#9), 6 min 6 sec ago
- Last stable build (#9), 6 min 6 sec ago
- Last successful build (#9), 6 min 6 sec ago
- Last failed build (#7), 25 min ago
- Last unsuccessful build (#7), 25 min ago
- Last completed build (#9), 6 min 6 sec ago

localhost:8080/job/InclassJenkin/job/CMPE272/10/console

Dashboard > InclassJenkins > CMPE272 > #10

```
XPC_FLAGS=0x0
BUILD_DISPLAY_NAME=#10
JENKINS_URL=http://localhost:8080/
BUILD_ID=10
JAVA_MAIN_CLASS_64670=executable.Main
GIT_PREVIOUS_SUCCESSFUL_COMMIT=e2cb1d25d967b816750e8ed7099d0c94d8e8298f
XPC_SERVICE_NAME=0
JOB_BASE_NAME=CMPE272
RUN_TESTS_DISPLAY_URL=http://localhost:8080/job/InclassJenkin/job/CMPE272/10/display/redirect?page=tests
SHLVL=3
HOME=/Users/poojasindham
GIT_BRANCH=origin/main
HOMEBREW_PREFIX=/opt/homebrew
CI=true
WORKSPACE_TMP=/Users/poojasindham/.jenkins/workspace/InclassJenkin_CMPE272@tmp
EXECUTOR_NUMBER=0
JENKINS_SERVER_COOKIE=durable-e2d01dced598865866e60e8b62048716490619fc665f142453018c9f9a201564
NODE_LABELS=built-in
GIT_URL=https://github.com/poojasindham-sjsu/CMPE272_Inclass_Jenkins.git
LOGNAME=poojasindham
HUDSON_HOME=/Users/poojasindham/.jenkins
NODE_NAME=built-in
JOB_DISPLAY_URL=http://localhost:8080/job/InclassJenkin/job/CMPE272/display/redirect
BUILD_NUMBER=10
INFOPATH=/opt/homebrew/share/info:
HOMEBREW_CELLAR=/opt/homebrew/Cellar
HUDSON_COOKIE=7898cffb-578b-48ed-9622-2ef62c7fd93c
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

REST API Jenkins 2.492.2

Step 5: Recording tests and artifacts

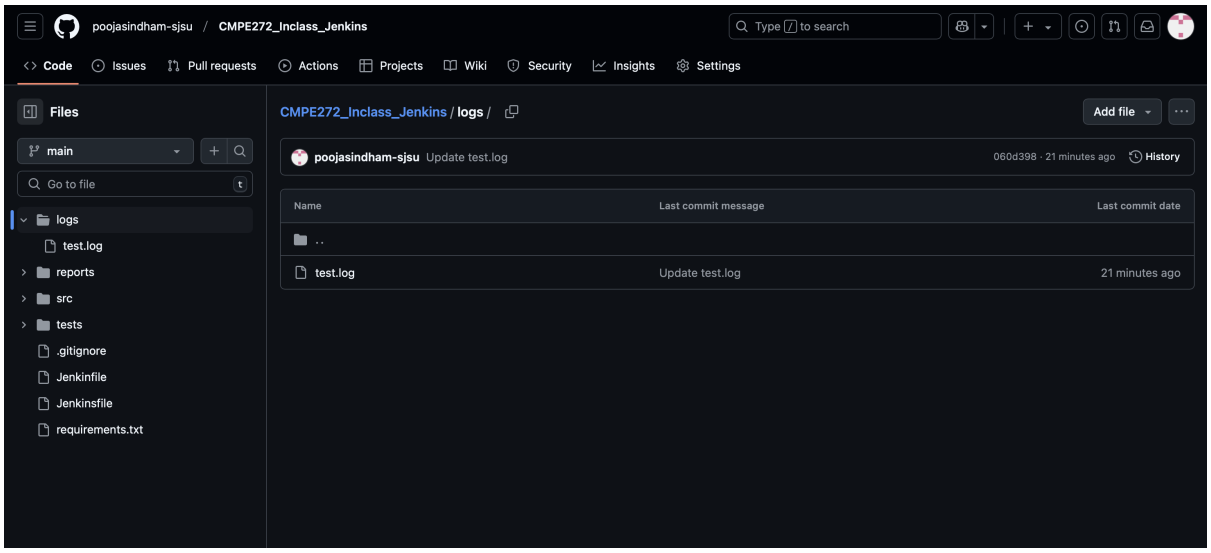
Update Jenkinsfile.

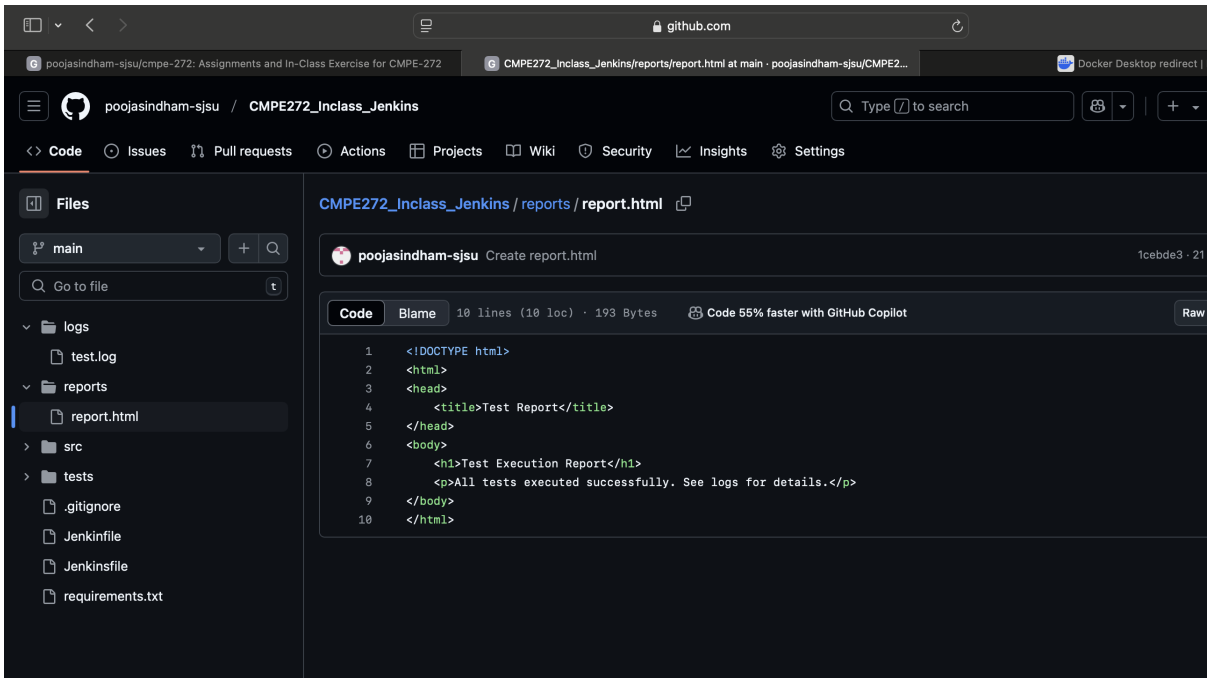
```
pipeline {
agent {
      docker { image 'python:3.8' }  // Use Python Docker image
   }
   stages {
      stage('Setup') {
         steps {
            echo "Setting up virtual environment..."
            sh '''
            python -m venv venv
            . venv/bin/activate  # Use dot instead of source
            pip install --upgrade pip
            pip install -r requirements.txt
            '''
         }
      }

      stage('Run Tests') {
         steps {
            echo "Running tests..."
            sh '''
            . venv/bin/activate
            venv/bin/pytest tests/test_sample.py --junitxml=test-results.xml  # Use full path
            '''
         }
      }

      stage('Publish Test Results') {
         steps {
            echo "Publishing test results..."
            catchError(buildResult: 'SUCCESS', stageResult: 'UNSTABLE') {
               junit 'test-results.xml'
            }
         }
      }

      stage('Archive Artifacts') {
         steps {
            echo "Archiving logs and test reports..."
            archiveArtifacts artifacts: 'logs/test.log, reports/report.html', fingerprint: true
         }
      }
   }

   post {
      always {
         echo "Pipeline execution complete!"
      }
      success {
         echo "SUCCESS: Tests executed, results published, and artifacts stored!"
      }
      unstable {
```

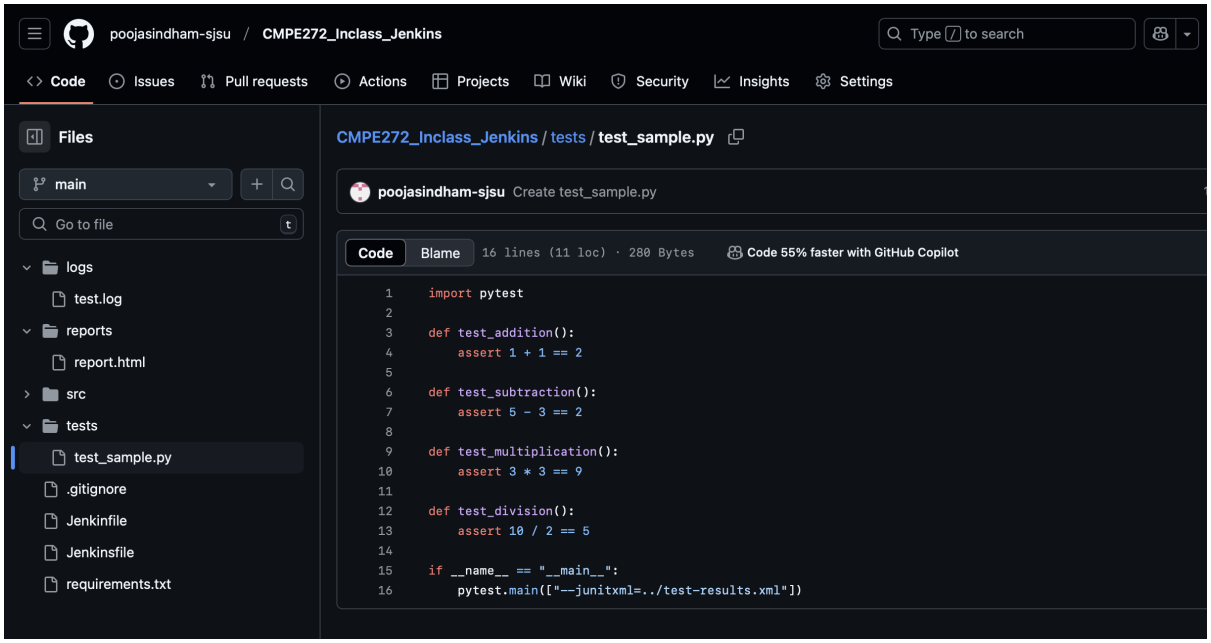
```
            echo "Some tests failed, but results and artifacts are available."
        }
    }
}
```

Created file to test artifacts and archive.

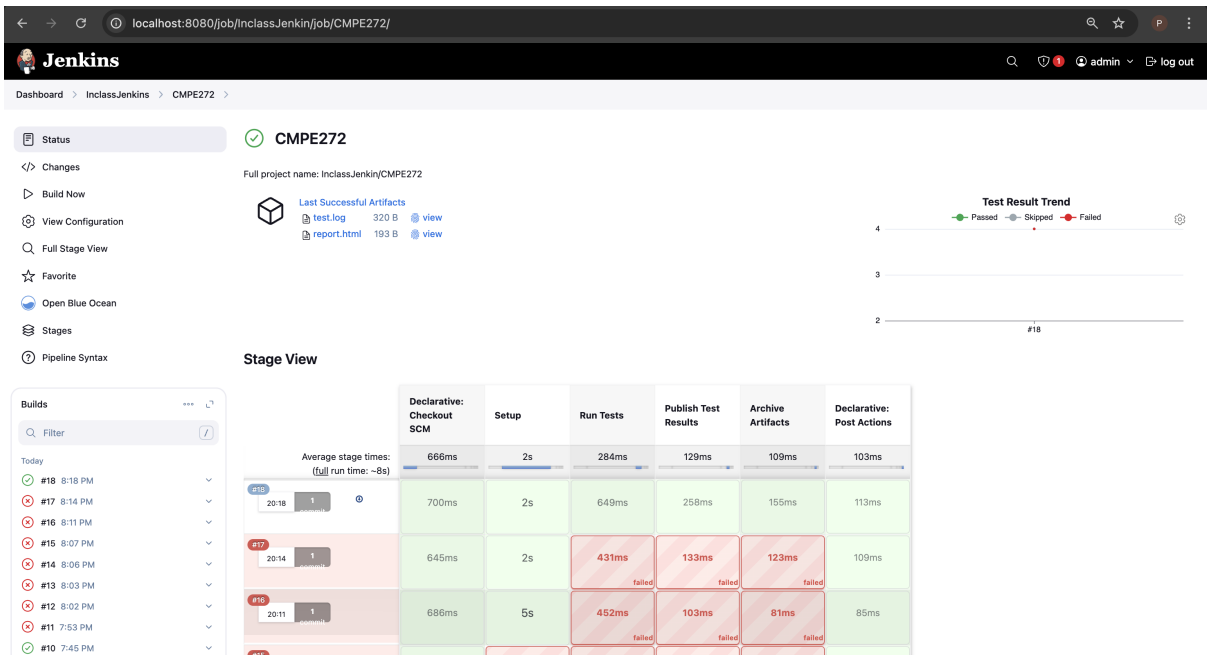
- A sample Python test file using *pytest*.
- A log file to archive.
- A dummy report file to store as an artifact.
- A *.gitignore* to avoid unnecessary files in the repo.







Pipeline ran Successfully.

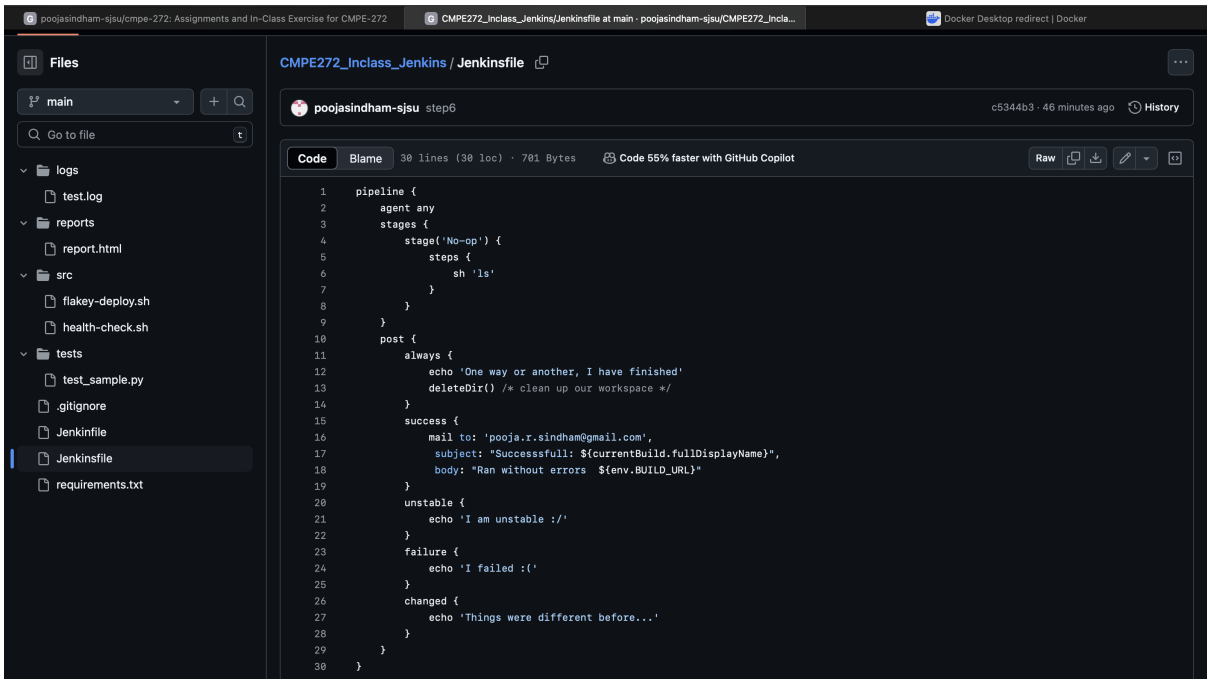


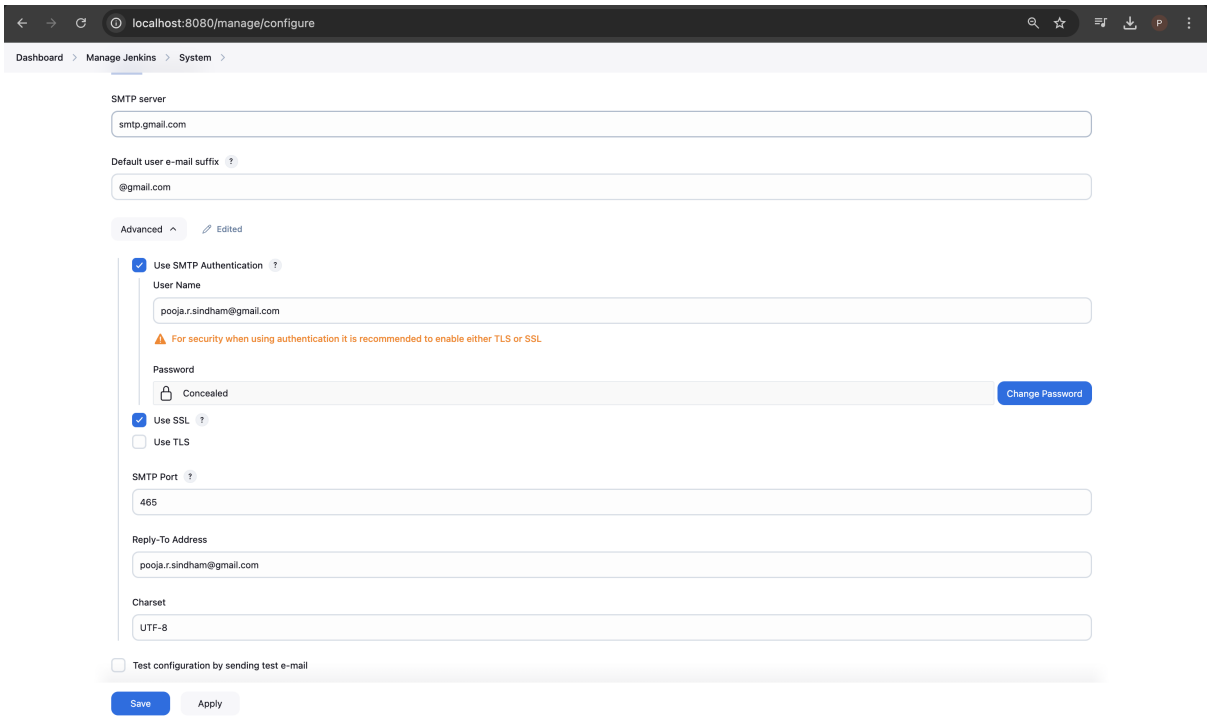
localhost:8080/job/InclassJenkin/job/CMPE272/18/console

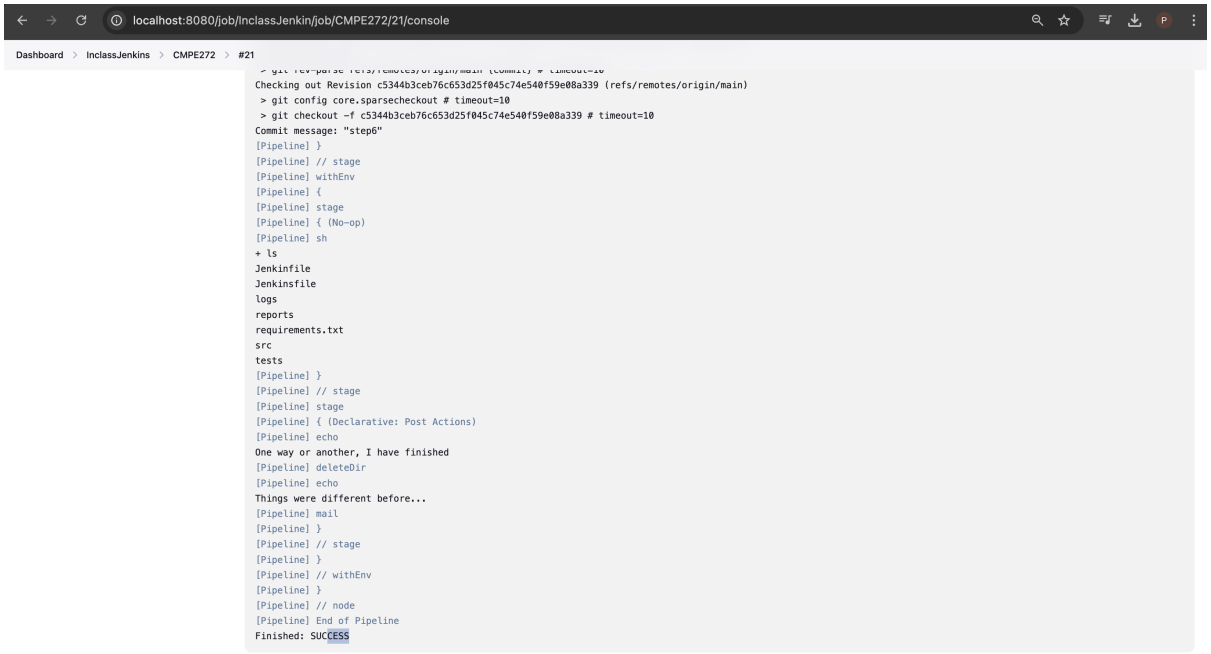
Dashboard > InclassJenkins > CMPE272 > #18

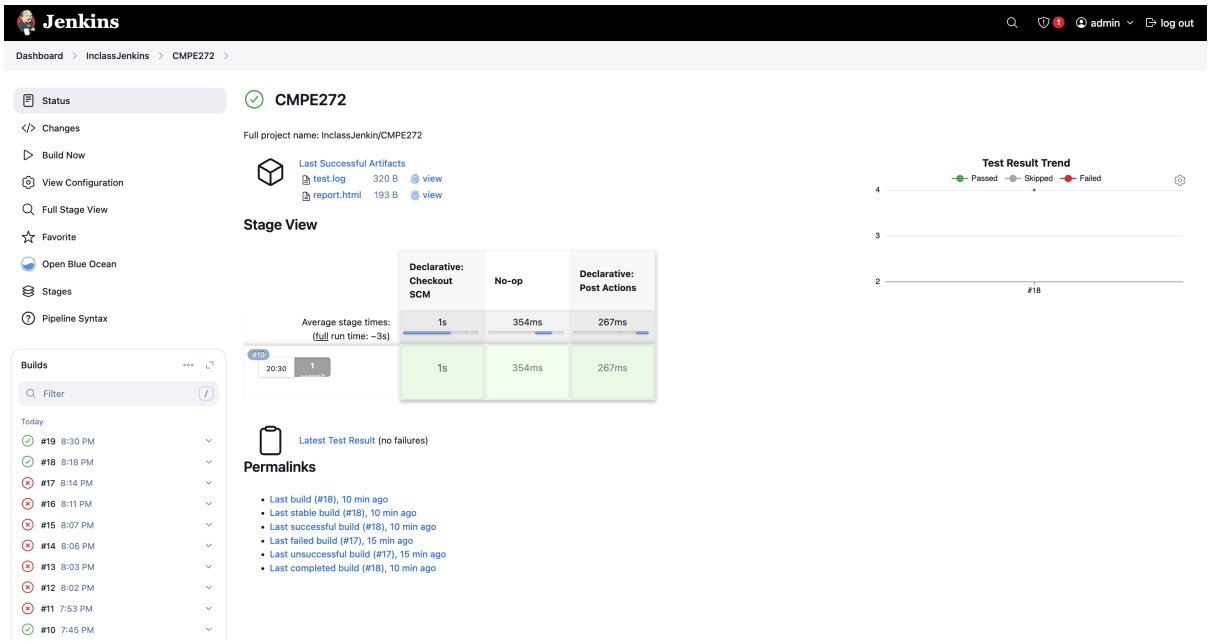
```
Publishing test results...
[Pipeline] catchError
[Pipeline] {
[Pipeline] junit
Recording test results
[Checks API] No suitable checks publisher found.
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Archive Artifacts)
[Pipeline] echo
Archiving logs and test reports...
[Pipeline] archiveArtifacts
Archiving artifacts
Recording fingerprints
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline execution complete!
[Pipeline] echo
SUCCESS: Tests executed, results published, and artifacts stored!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
$ docker stop --time=1 75ea7e0e75f9c8de217ad9aeb006a6569445493a8ae4ab01235eb73a2cb0a366
$ docker rm -f --volumes 75ea7e0e75f9c8de217ad9aeb006a6569445493a8ae4ab01235eb73a2cb0a366
[Pipeline] // withDockerContainer
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

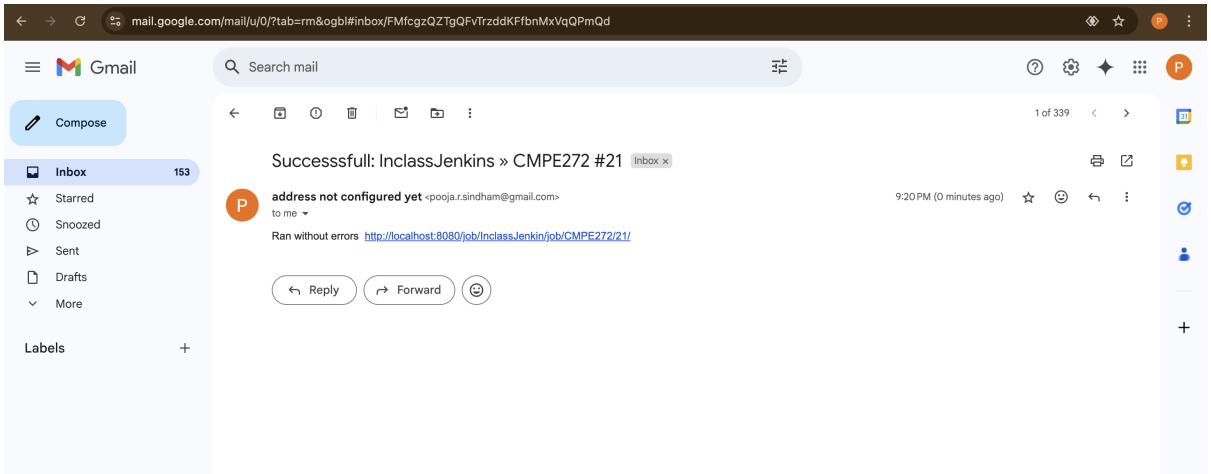
Step 6: Cleaning Up and Notifications






localhost:8080/job/InclassJenkin/job/CMPE272/21/console
Dashboard > InclassJenkins > CMPE272 > #21
Checking out Revision c5344b3ceb76c653d25f045c74e540f59e08a339 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f c5344b3ceb76c653d25f045c74e540f59e08a339 # timeout=10
Commit message: "step6"
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (No-op)
[Pipeline] sh
+ ls
Jenkinfile
Jenkinsfile
logs
reports
requirements.txt
src
tests
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
One way or another, I have finished
[Pipeline] deleteDir
[Pipeline] echo
Things were different before...
[Pipeline] mail
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS


Jenkins
admin
log out
Dashboard > InclassJenkins > CMPE272 >
Status
Changes
Build Now
View Configuration
Full Stage View
Favorite
Open Blue Ocean
Stages
Pipeline Syntax
Builds
Filter
Today
#19 8:30 PM
#18 8:18 PM
#17 8:14 PM
#16 8:11 PM
#15 8:07 PM
#14 8:06 PM
#13 8:03 PM
#12 8:02 PM
#11 7:53 PM
#10 7:45 PM
CMPE272
Full project name: InclassJenkin/CMPE272
Last Successful Artifacts
test.log 320 B view
report.html 193 B view
Stage View
Declarative: Checkout SCM
No-op
Declarative: Post Actions
Average stage times:
(full run time: ~3s)
1s
354ms
267ms
#19
20:30
Latest Test Result (no failures)
Permalinks
Last build (#18), 10 min ago
Last stable build (#18), 10 min ago
Last successful build (#18), 10 min ago
Last failed build (#17), 15 min ago
Last unsuccessful build (#17), 15 min ago
Last completed build (#18), 10 min ago
Test Result Trend
Passed
Skipped
Failed
#18


mail.google.com/mail/u/0/?tab=rm&ogbl#inbox/FMfcgzQZTgQFvTrzddKFfbnMxVqQPmQd
Gmail
Search mail
Compose
Inbox 153
Starred
Snoozed
Sent
Drafts
More
Labels
1 of 339
Successsfull: InclassJenkins » CMPE272 #21
Inbox
address not configured yet <pooja.r.sindham@gmail.com>
to me
9:20 PM (0 minutes ago)
Ran without errors http://localhost:8080/job/InclassJenkin/job/CMPE272/21/
Reply
Forward

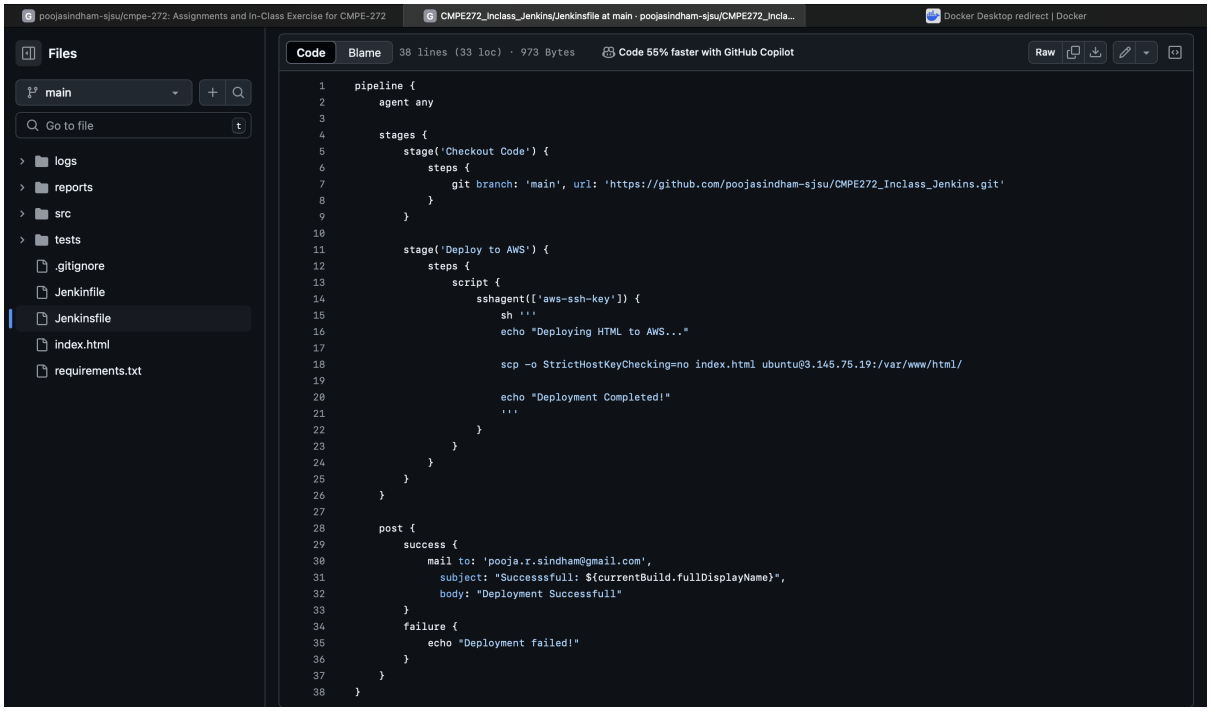
Step 7: Deployment.

Created one sample index.html and deploying it on AWS EC2 instances through Jenkins.

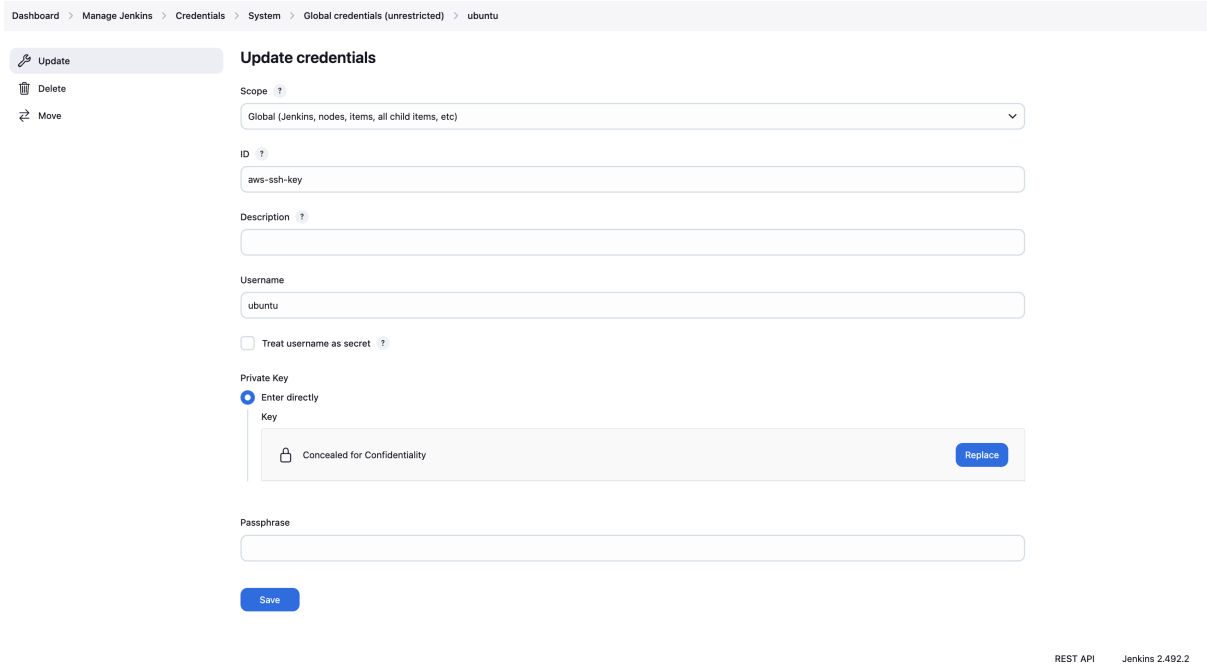
index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Jenkins Deployment on AWS</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      text-align: center;
      background: linear-gradient(135deg, #ff7e5f, #feb47b);
      color: white;
    }
    .container {
      width: 80%;
      margin: 100px auto;
      padding: 20px;
      background: rgba(255, 255, 255, 0.1);
      border-radius: 10px;
      box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
    }
    h1 {
      font-size: 2.5em;
    }
    p {
      font-size: 1.2em;
    }
    .btn {
      display: inline-block;
      padding: 10px 20px;
      margin-top: 20px;
      background: #fff;
      color: #ff7e5f;
      text-decoration: none;
      font-weight: bold;
      border-radius: 5px;
      transition: 0.3s;
    }
    .btn:hover {
      background: #ff7e5f;
      color: #fff;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>🚀 Jenkins Deployment on AWS EC2</h1>
    <p>This page has been successfully deployed using Jenkins.</p>
    <p>Enjoy automated CI/CD pipelines with GitHub, Jenkins, and AWS!</p>
    <a href="https://github.com/poojasindham-sjsu/CMPE272_Inclass_Jenkins" class="btn" target="_blank">View Repository</a>
  </div>
</body>
</html>
```

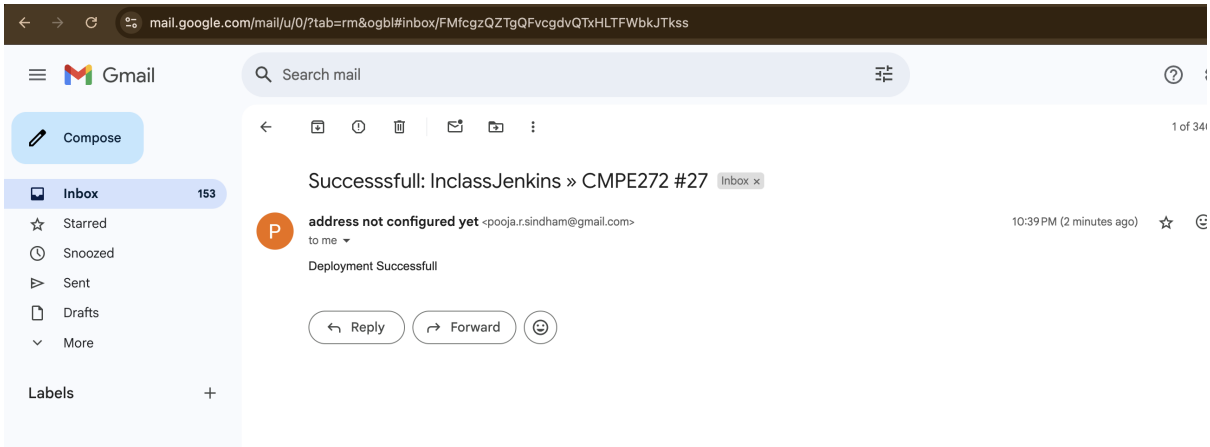
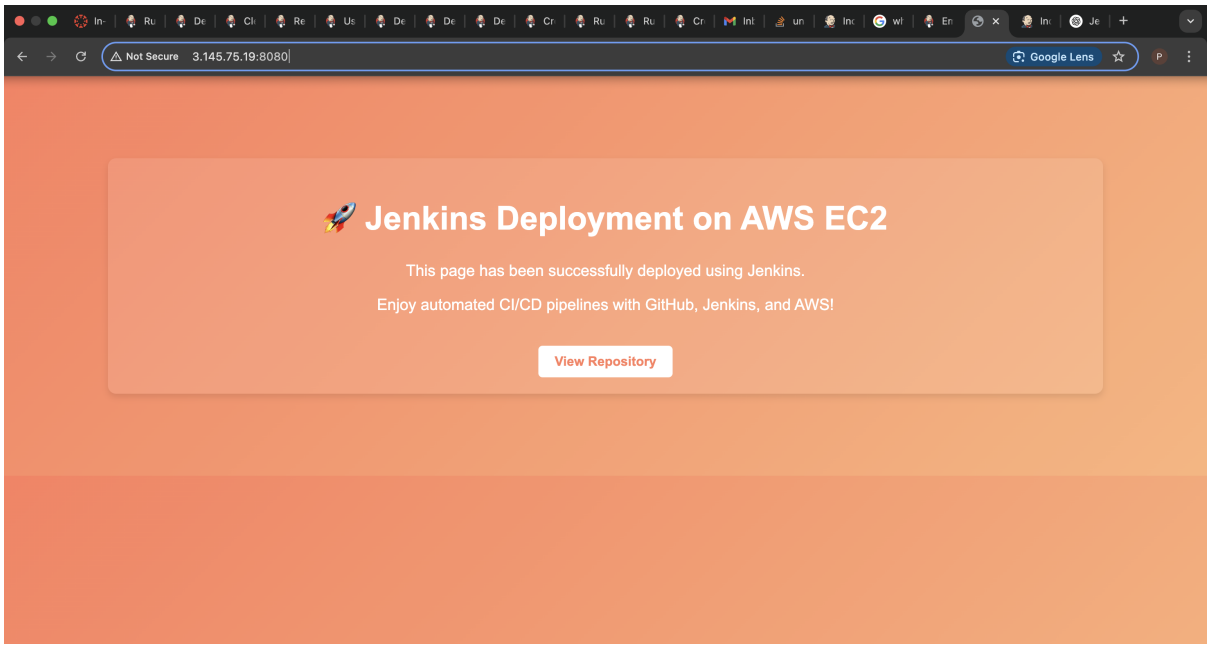
JenkinsFile



Adding credentials in Jenkins



Email received for successful deployment



Deployment successful