CLOUD COMPUTING

Assignment 4A – Jenkins

(Creating a DevOps Pipeline, CI/CD tool)

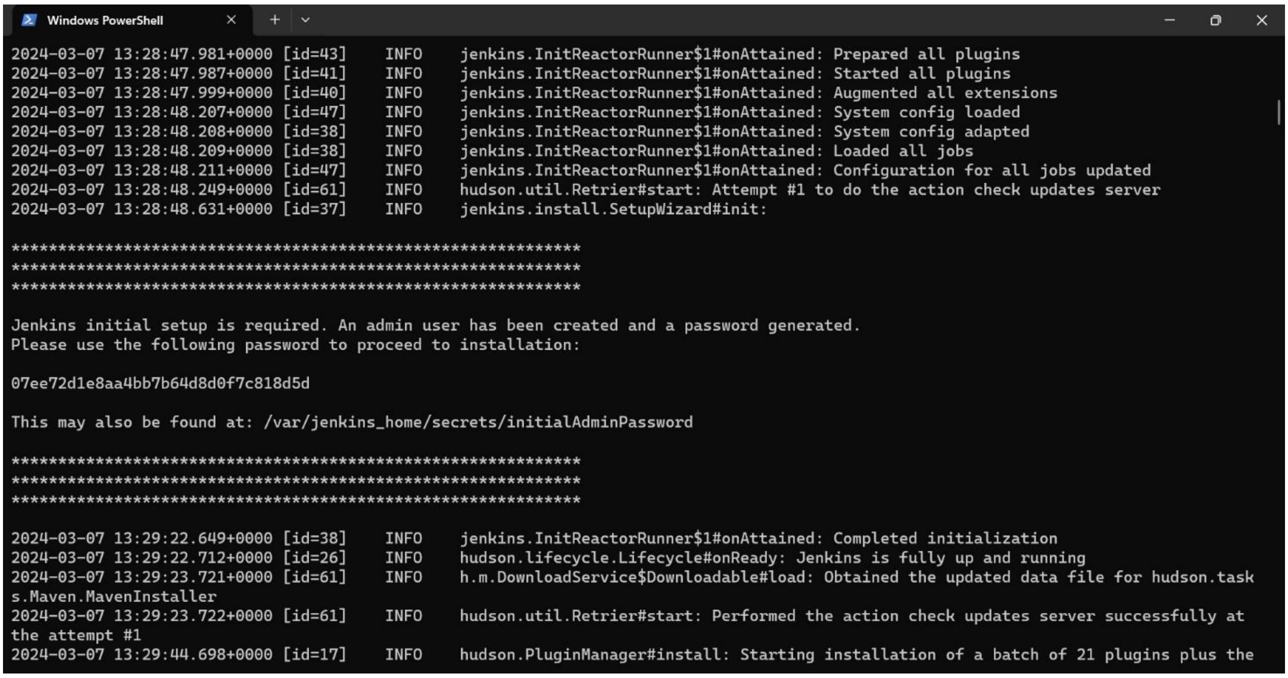
Name : Mayadevi
SRN : PES2UG21CS284
Section: E

Task-1

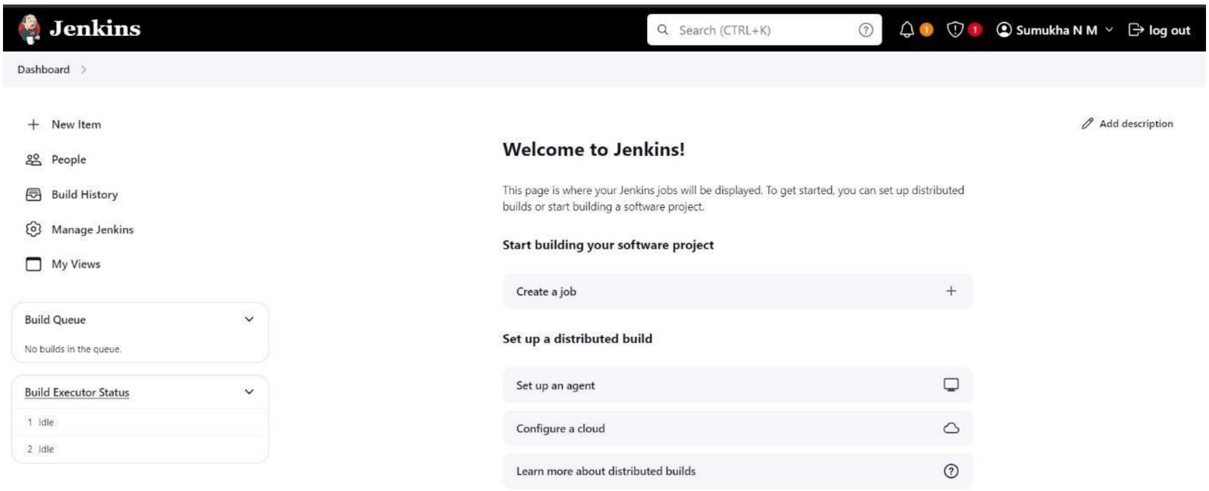
Aim: Set up Jenkins using Docker.

Deliverables:

1. Screenshot of the running Docker Container after installing Jenkins



```
2024-03-07 13:28:47.981+0000 [id=43]    INFO    jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2024-03-07 13:28:47.987+0000 [id=41]    INFO    jenkins.InitReactorRunner$1#onAttained: Started all plugins
2024-03-07 13:28:47.999+0000 [id=40]    INFO    jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2024-03-07 13:28:48.207+0000 [id=47]    INFO    jenkins.InitReactorRunner$1#onAttained: System config loaded
2024-03-07 13:28:48.208+0000 [id=38]    INFO    jenkins.InitReactorRunner$1#onAttained: System config adapted
2024-03-07 13:28:48.209+0000 [id=38]    INFO    jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2024-03-07 13:28:48.211+0000 [id=47]    INFO    jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2024-03-07 13:28:48.249+0000 [id=61]    INFO    hudson.util.Retrier#start: Attempt #1 to do the action check updates server
2024-03-07 13:28:48.631+0000 [id=37]    INFO    jenkins.install.SetupWizard#init:

*************************************************************
*************************************************************
*************************************************************

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

07ee72d1e8aa4bb7b64d8d0f7c818d5d

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

*************************************************************
*************************************************************
*************************************************************

2024-03-07 13:29:22.649+0000 [id=38]    INFO    jenkins.InitReactorRunner$1#onAttained: Completed initialization
2024-03-07 13:29:22.712+0000 [id=26]    INFO    hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
2024-03-07 13:29:23.721+0000 [id=61]    INFO    h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.task
s.Maven.MavenInstaller
2024-03-07 13:29:23.722+0000 [id=61]    INFO    hudson.util.Retrier#start: Performed the action check updates server successfully at
the attempt #1
2024-03-07 13:29:44.698+0000 [id=17]    INFO    hudson.PluginManager#install: Starting installation of a batch of 21 plugins plus the
```



No builds in the queue.

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

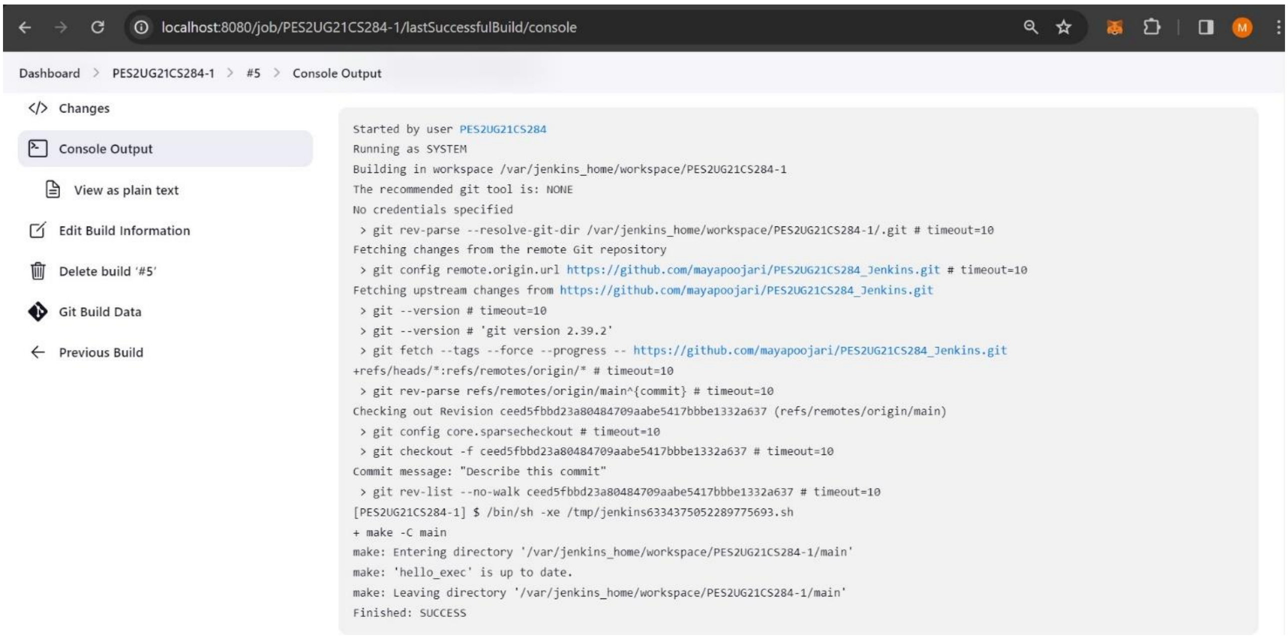
Start building your software project

Set up a distributed build

Task-2

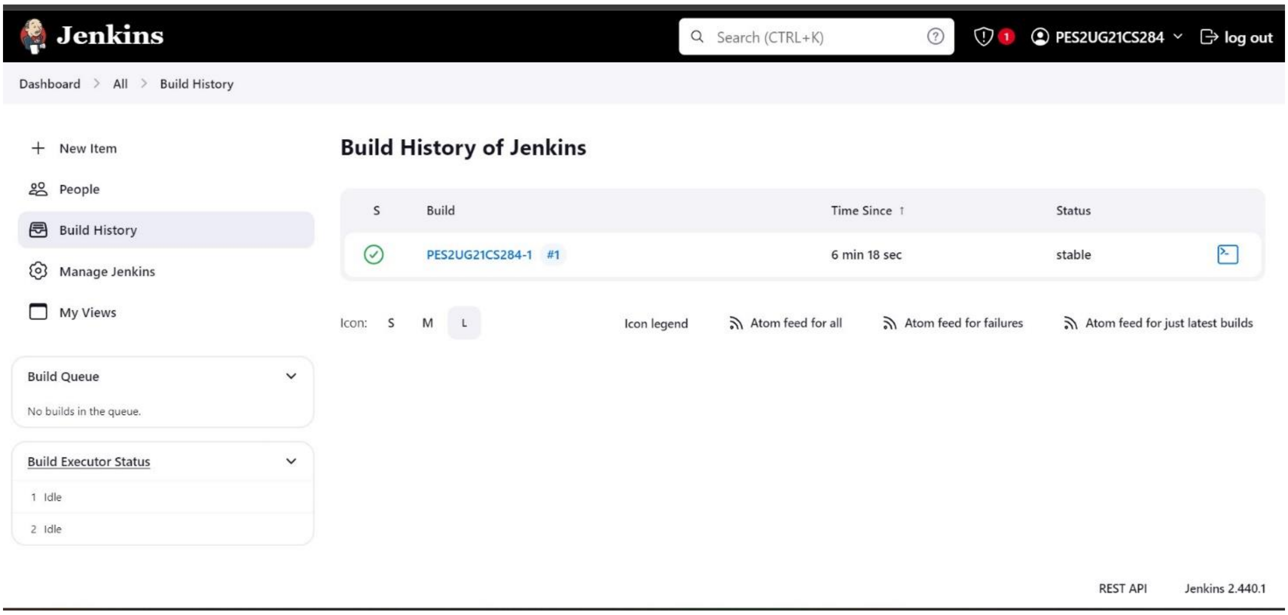
Aim: Set up a job in Jenkins to connect to your repository and build C++ hello.cpp.
Deliverables:

1. Picture showing the console output after the build is successful



```
Started by user PES2UG21CS284
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/PES2UG21CS284-1
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/jenkins_home/workspace/PES2UG21CS284-1/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/mayapoojari/PES2UG21CS284_Jenkins.git # timeout=10
Fetching upstream changes from https://github.com/mayapoojari/PES2UG21CS284_Jenkins.git
 > git --version # timeout=10
 > git --version # 'git version 2.39.2'
 > git fetch --tags --force --progress -- https://github.com/mayapoojari/PES2UG21CS284_Jenkins.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision ceed5fbbd23a80484709aabe5417bbbe1332a637 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f ceed5fbbd23a80484709aabe5417bbbe1332a637 # timeout=10
Commit message: "Describe this commit"
 > git rev-list --no-walk ceed5fbbd23a80484709aabe5417bbbe1332a637 # timeout=10
[PES2UG21CS284-1] $ /bin/sh -xe /tmp/jenkins6334375052289775693.sh
+ make -C main
make: Entering directory '/var/jenkins_home/workspace/PES2UG21CS284-1/main'
make: 'hello_exec' is up to date.
make: Leaving directory '/var/jenkins_home/workspace/PES2UG21CS284-1/main'
Finished: SUCCESS
```

2. Picture showing the Stable state of the task in Build History of Jenkins



Build History of Jenkins

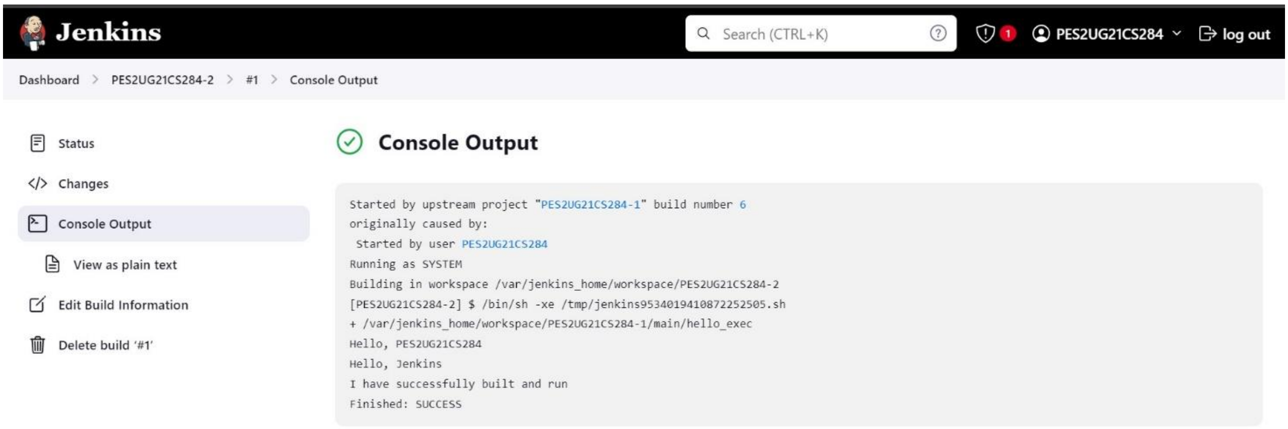
S	Build	Time Since ↑	Status
✓	PES2UG21CS284-1 #1	6 min 18 sec	stable

Task-3

Aim: Set up a second job that automatically runs after the project builds. This is different from the other job because this will not have a git repository - it doesn't even build anything.

Deliverables:

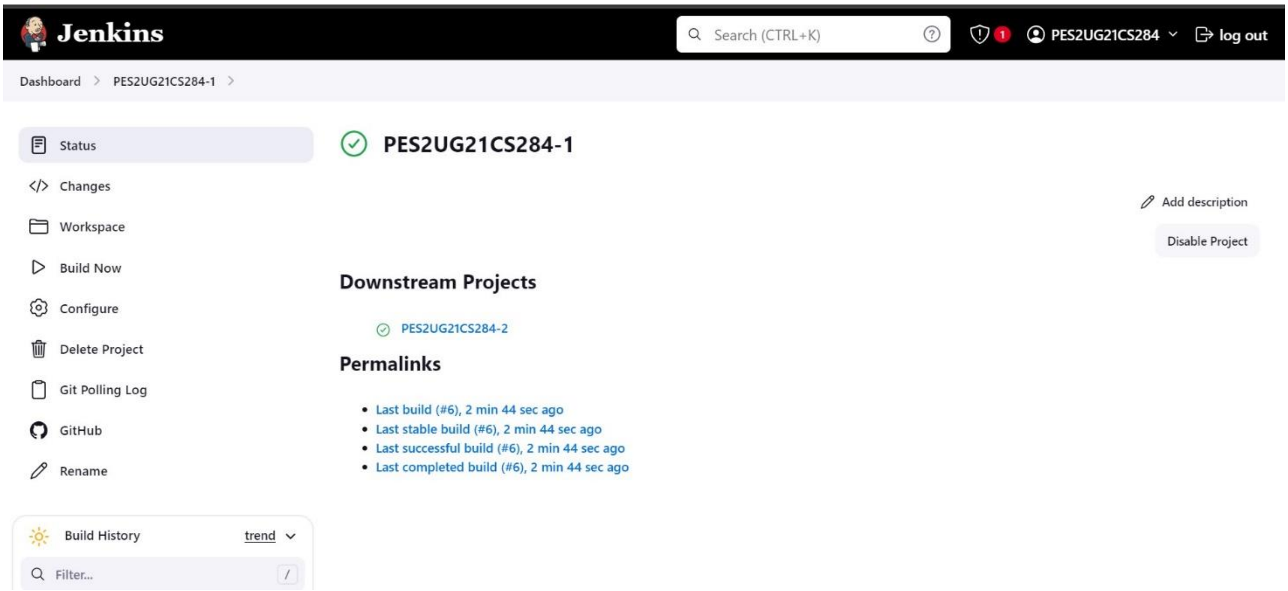
1. Console output of second job



Console Output

```
Started by upstream project "PES2UG21CS284-1" build number 6
originally caused by:
 Started by user PES2UG21CS284
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/PES2UG21CS284-2
[PES2UG21CS284-2] $ /bin/sh -xe /tmp/jenkins9534019410872252505.sh
+ /var/jenkins_home/workspace/PES2UG21CS284-1/main/hello_exec
Hello, PES2UG21CS284
Hello, Jenkins
I have successfully built and run
Finished: SUCCESS
```

2. *Status page of first job*



PES2UG21CS284-1

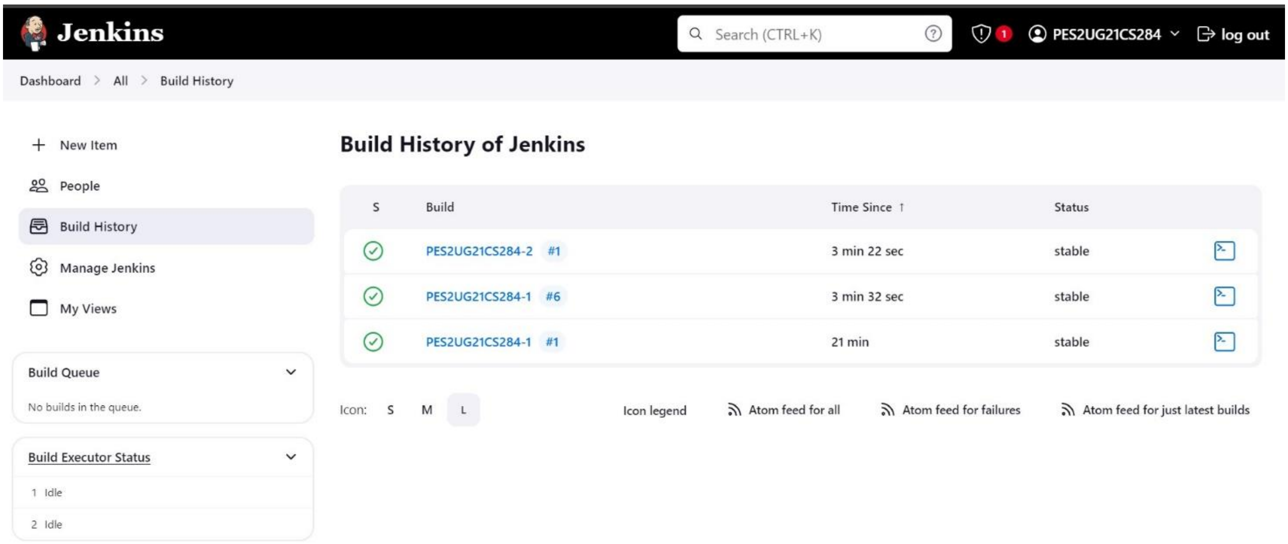
Downstream Projects

PES2UG21CS284-2

Permalinks

- Last build (#6), 2 min 44 sec ago
- Last stable build (#6), 2 min 44 sec ago
- Last successful build (#6), 2 min 44 sec ago
- Last completed build (#6), 2 min 44 sec ago

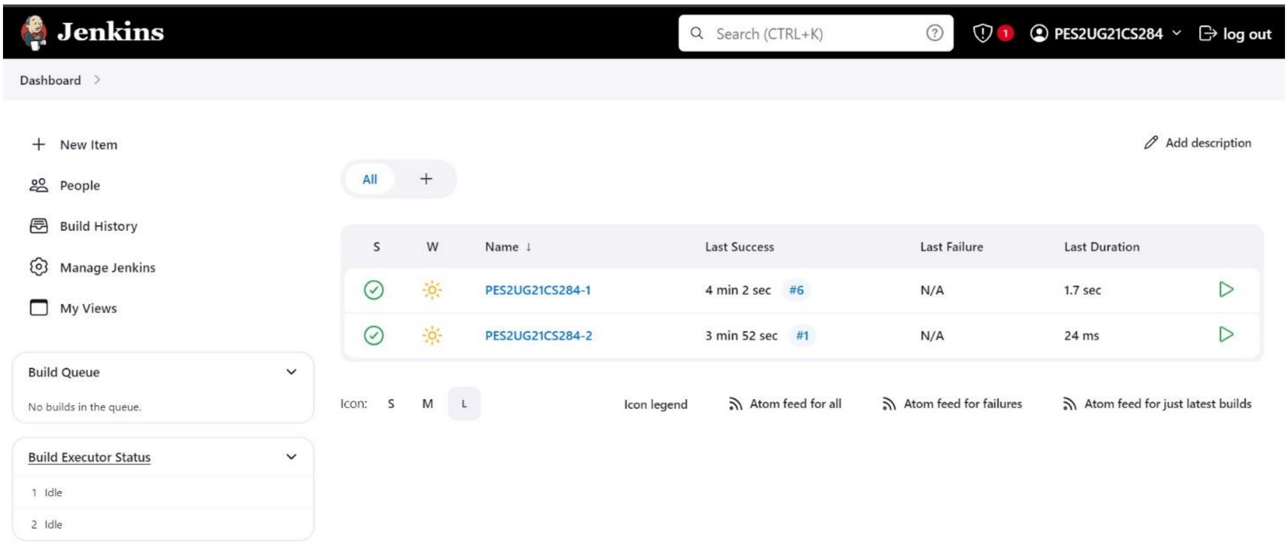
3. *Build History of Jenkins*



Build History of Jenkins

S	Build	Time Since ↑	Status
✓	PES2UG21CS284-2 #1	3 min 22 sec	stable
✓	PES2UG21CS284-1 #6	3 min 32 sec	stable
✓	PES2UG21CS284-1 #1	21 min	stable

4. Jenkins Dashboard



S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	PES2UG21CS284-1	4 min 2 sec #6	N/A	1.7 sec
✓	☀	PES2UG21CS284-2	3 min 52 sec #1	N/A	24 ms

Build Queue

No builds in the queue.

Build Executor Status

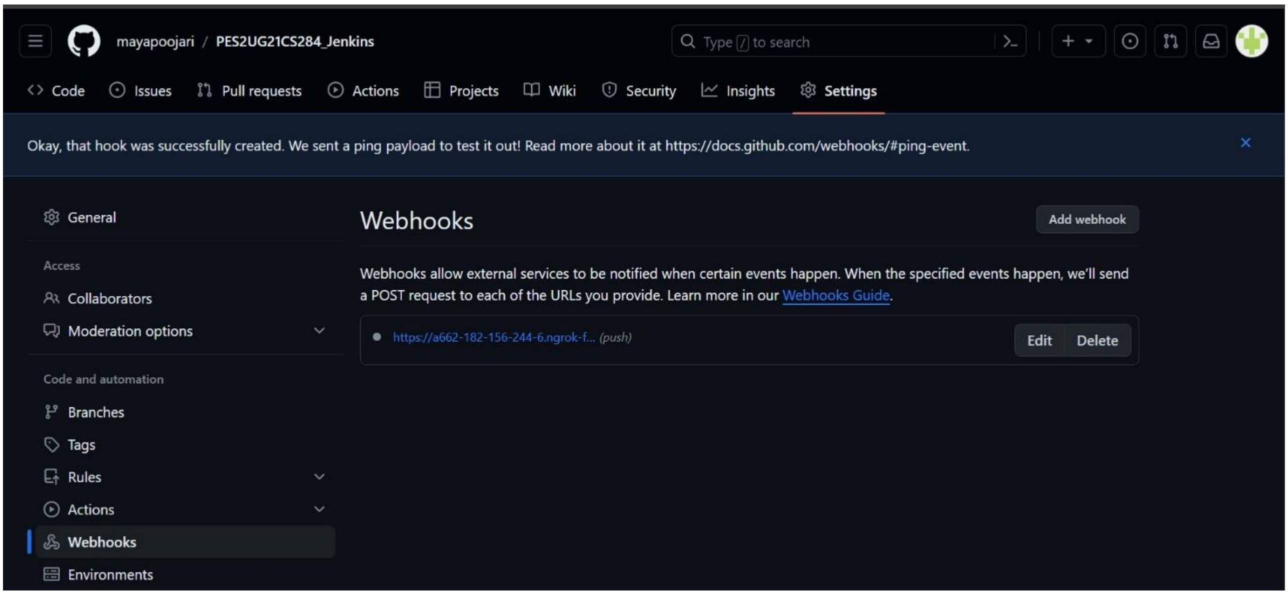
1 Idle

2 Idle

Task-4

Aim: Add a webhook trigger to your repository in order to automate builds in Jenkins

1. Webhook added to your GitHub repository



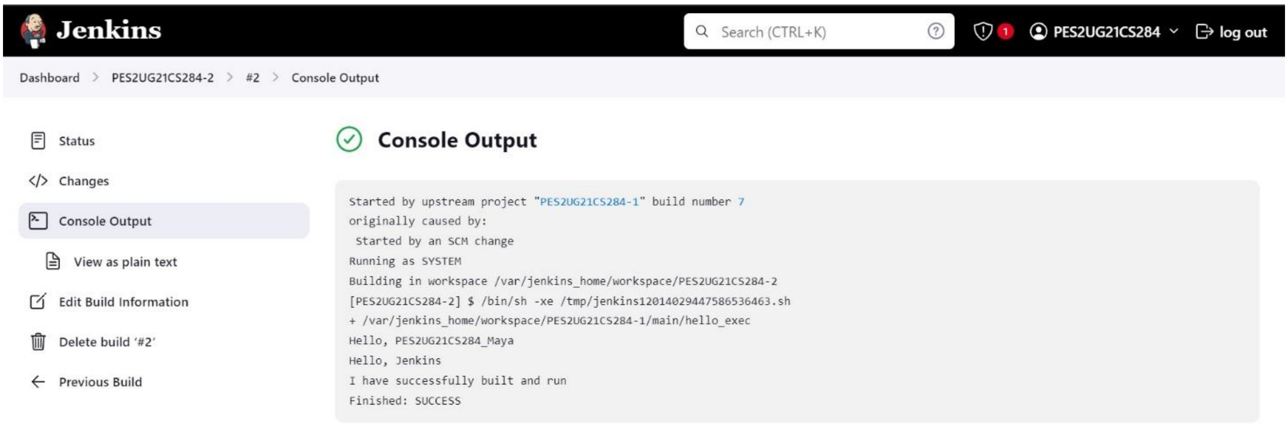
Okay, that hook was successfully created. We sent a ping payload to test it out! Read more about it at https://docs.github.com/webhooks/#ping-event.

Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

https://a662-182-156-244-6.ngrok-f... (push)

2. Console Output of second job displaying the change made in hello.cpp file.



Console Output

```
Started by upstream project "PES2UG21CS284-1" build number 7
originally caused by:
 Started by an SCM change
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/PES2UG21CS284-2
[PES2UG21CS284-2] $ /bin/sh -xe /tmp/jenkins12014029447586536463.sh
+ /var/jenkins_home/workspace/PES2UG21CS284-1/main/hello_exec
Hello, PES2UG21CS284_Maya
Hello, Jenkins
I have successfully built and run
Finished: SUCCESS
```

Task-5

Aim: To create a basic Jenkins pipeline. *Deliverables:*

1. Code/script written to create basic pipeline using GitHub repository

Sample 1 code :

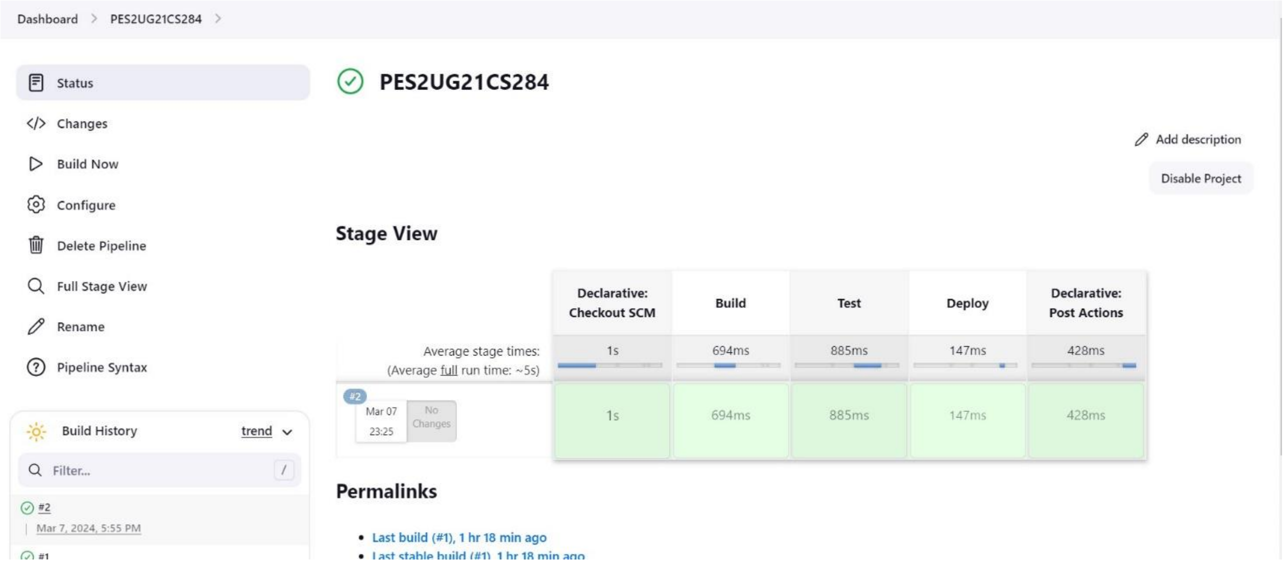
```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                script {
                    // Intentionally misspelled 'g++' as 'gcc++' to cause a
compilation error
                    try {
                        sh 'gcc++ -o my_program my_program.cpp'
                        echo 'Build Stage Successful'
                    } catch (Exception e) {
                        echo "Build Failed: ${e.message}"
                        currentBuild.result = 'FAILURE'
                        error 'Build failed'
                    }
                }
            }
        }
        stage('Test') {
            steps {
                script {
                    // Intentionally misspelled './my_program' as
'./my_program_wrong' to cause a test failure
                    try {
                        def output = sh './my_program_wrong', returnStdout: true
                        echo "Test Output: ${output}"
                        echo 'Test Stage Successful'
                    } catch (Exception e) {
                        echo "Test Failed: ${e.message}"
                        currentBuild.result = 'FAILURE'
                        error 'Test failed'
                    }
                }
            }
            post {
                always {
                    script {
                        // Intentionally added a non-existing shell command to
cause a post-test error
                        sh 'echo "Additional tests completed"'
                        sh 'this_command_does_not_exist'
                    }
                }
            }
        }

        stage('Deploy') {
            steps {
                script {
                    // Intentionally using an invalid shell command to simulate
deployment failure
                    try {
                        sh 'non_existing_deploy_command'
                        echo 'Deployment Successful'
                    } catch (Exception e) {
                        echo "Deployment Failed: ${e.message}"
                        currentBuild.result = 'FAILURE'
                        error 'Deployment failed'
                    }
                }
            }
        }
    }

    post {
        failure {
            // Intentionally removing the echo statement to cause a syntax error
            echo 'Pipeline failed'
        }
        success {
            script {
                // Intentionally misspelled 'repositoryUrl' as 'repositoryURL' to
cause an error
                def repositoryUrl = sh(script: 'git config --get
remote.origin.url', returnStdout: true)
                echo "Repository URL: ${repositoryURL}"
            }
        }
    }
}
```

Sample 2 Code for error:

```
pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                script {
                    // Compile the .cpp file
                    sh 'g++ -o my_program my_program.cpp'
                    echo 'Build Stage Successful'
                }
            }
        }

        stage('Test') {
            steps {
                script {
                    // Run the compiled program and print its output
                    sh './my_program'
                    echo 'Test Stage Successful'
                }
            }
            post {
                always {
                    script {
                        // Add any additional test steps or checks here if needed
                        echo 'Running additional test steps or checks...'
                        // Example: Run a shell command
                        sh 'echo "Additional tests completed"'
                    }
                }
            }
        }

        stage('Deploy') {
            steps {
                script {
                    try {
                        echo 'Deployment Successful'
                    } catch (Exception e) {
                        echo "Deployment Failed: ${e.message}"
                        currentBuild.result = 'FAILURE'
                        error 'Deployment failed'
                    }
                }
            }
        }
    }

    post {
        failure {
            echo 'Pipeline failed'
        }
        success {
            script {
                // Print the repository URL
                def repositoryUrl = sh(script: 'git config --get remote.origin.url', returnStdout: true).trim()
                echo "Repository URL: ${repositoryUrl}"
            }
        }
    }
}
```

2. Output of working created pipeline, the screenshot should include
   a. Stage view / Execution status of pipeline with all stages succeeded
   b. Verify Declarative: Post Actions stage for handling failures.

c. Stage view / Execution status of pipeline with all stages succeeded



PES2UG21CS284

Stage View

	Declarative: Checkout SCM	Build	Test	Deploy	Declarative: Post Actions
Average stage times: (Average full run time: ~5s)	1s	694ms	885ms	147ms	428ms
#2 Mar 07 23:25 No Changes	1s	694ms	885ms	147ms	428ms

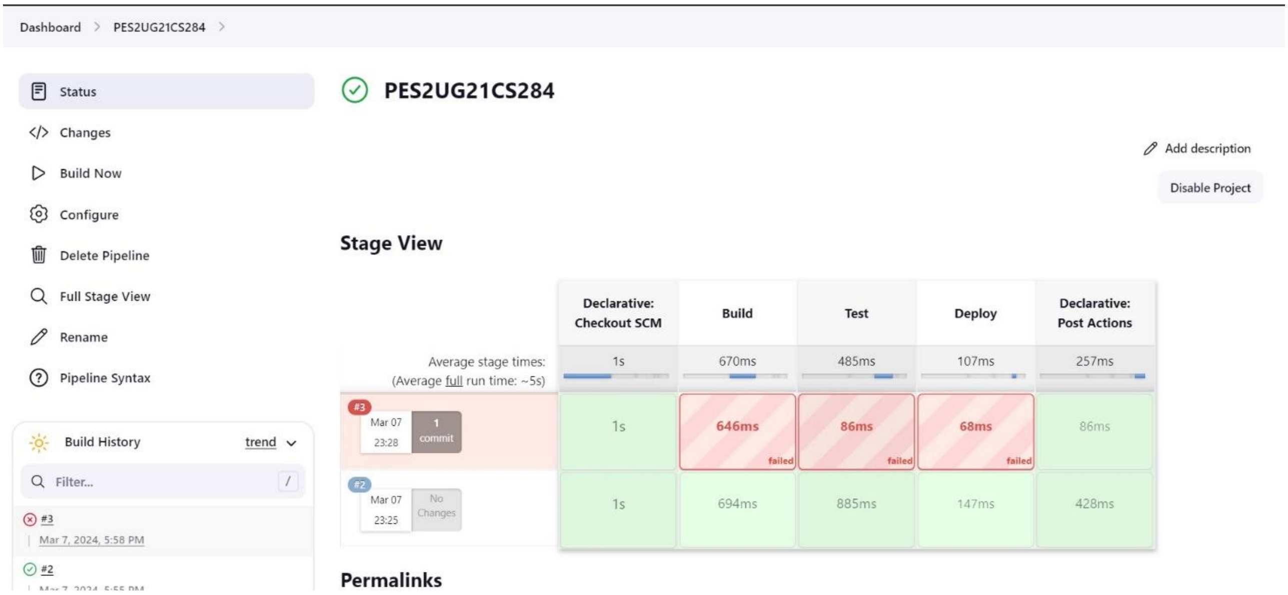
#2
Mar 7, 2024, 5:55 PM

#1

Permalinks

- Last build (#1), 1 hr 18 min ago
- Last stable build (#1), 1 hr 18 min ago

a. Verify Declarative: Post Actions stage for handling failures.



PES2UG21CS284

Stage View

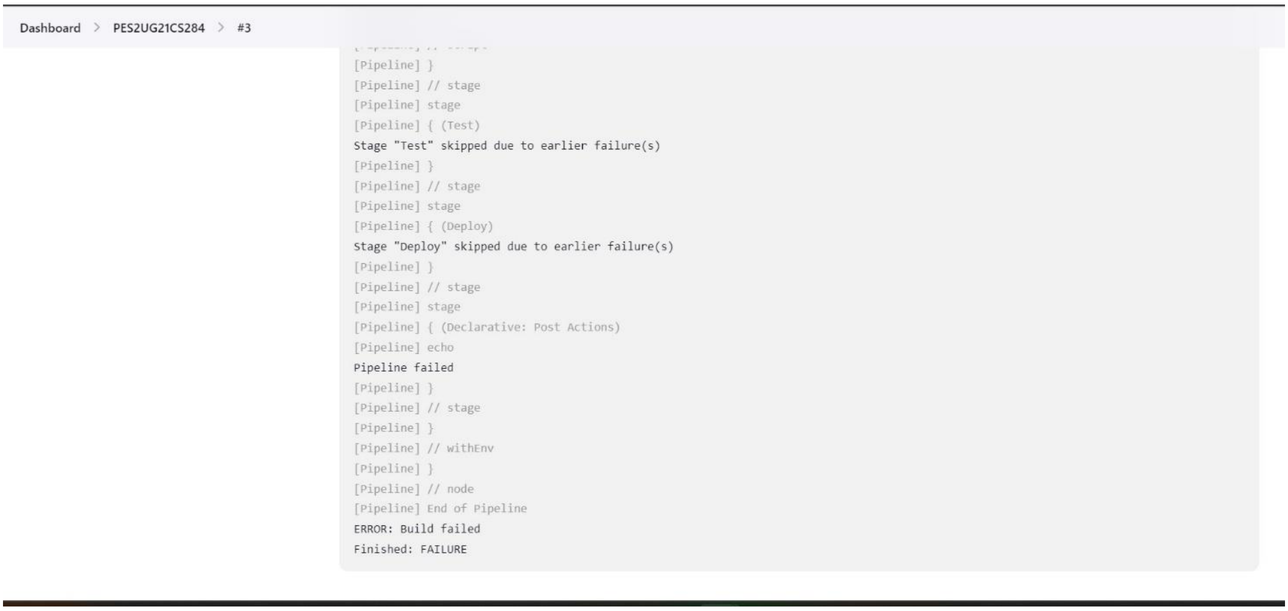
	Declarative: Checkout SCM	Build	Test	Deploy	Declarative: Post Actions
Average stage times: (Average full run time: ~5s)	1s	670ms	485ms	107ms	257ms
#3 Mar 07 23:28 1 commit	1s	646ms failed	86ms failed	68ms failed	86ms
#2 Mar 07 23:25 No Changes	1s	694ms	885ms	147ms	428ms

#3
Mar 7, 2024, 5:58 PM

#2
Mar 7, 2024, 5:55 PM

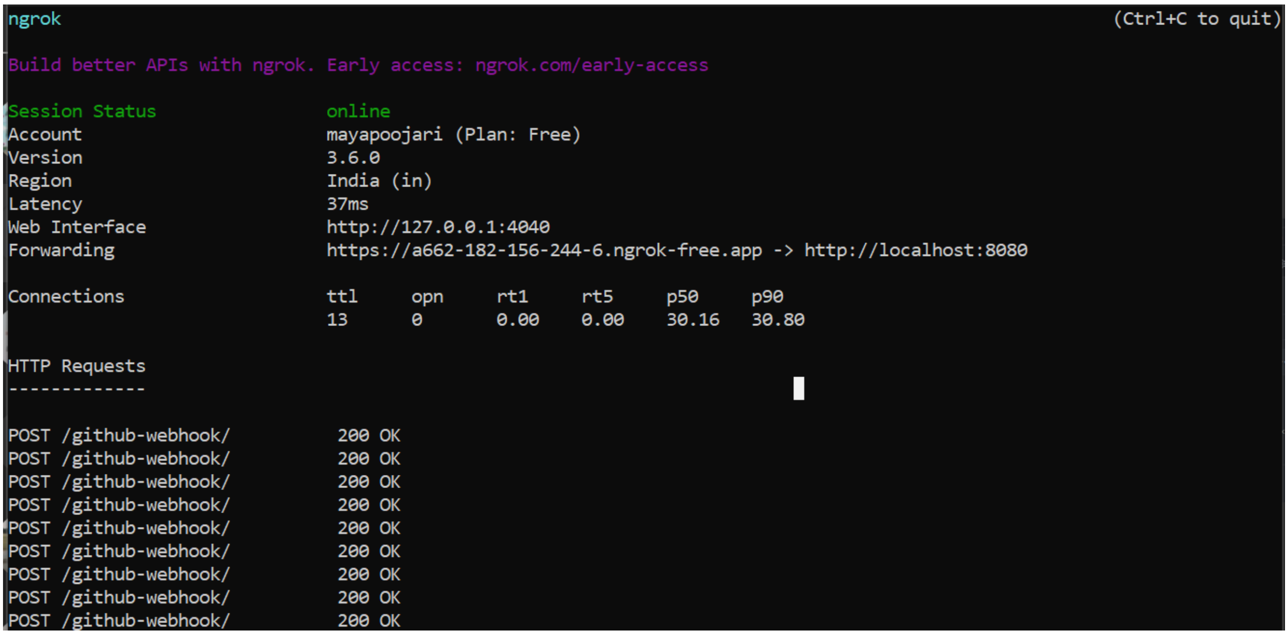
Permalinks

3. Output of the Pipeline



```
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
Stage "Test" skipped due to earlier failure(s)
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
Stage "Deploy" skipped due to earlier failure(s)
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline failed
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: Build failed
Finished: FAILURE
```

4. Link to the created GitHub repository

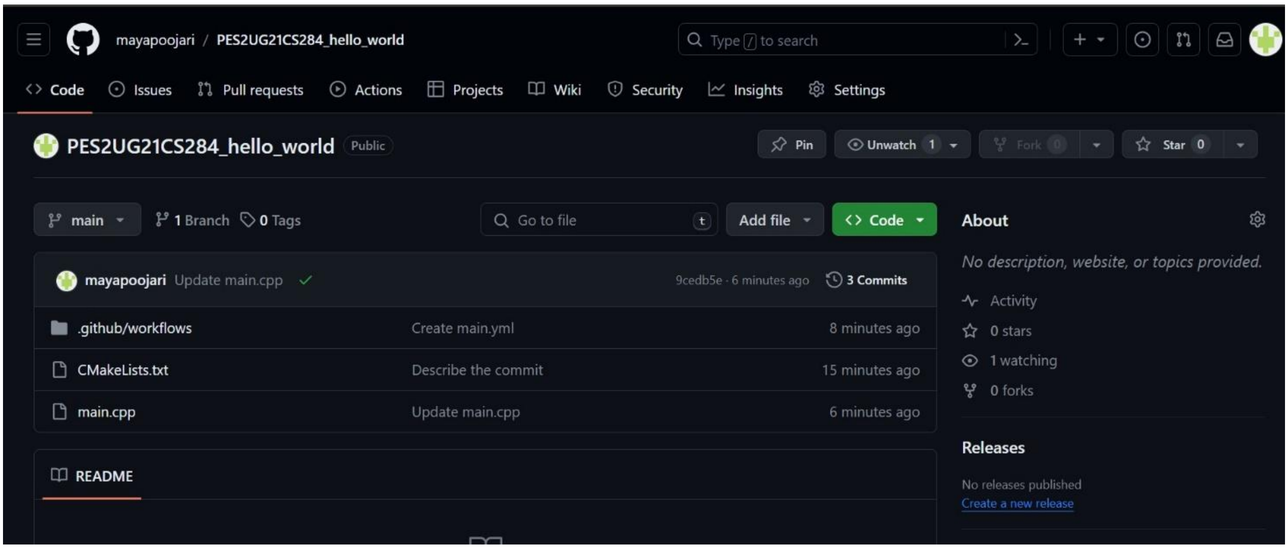


```
ngrok                                                                                        (Ctrl+C to quit)

Build better APIs with ngrok. Early access: ngrok.com/early-access

Session Status                online
Account                       mayapoojari (Plan: Free)
Version                       3.6.0
Region                        India (in)
Latency                       37ms
Web Interface                 http://127.0.0.1:4040
Forwarding                    https://a662-182-156-244-6.ngrok-free.app -> http://localhost:8080

Connections                   ttl     opn     rt1     rt5     p50     p90
                              13      0       0.00    0.00    30.16   30.80

HTTP Requests
-------------

POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
POST /github-webhook/          200 OK
```

Assignment – 4B – GitHub Actions

Building A CI Pipeline With GitHub Actions

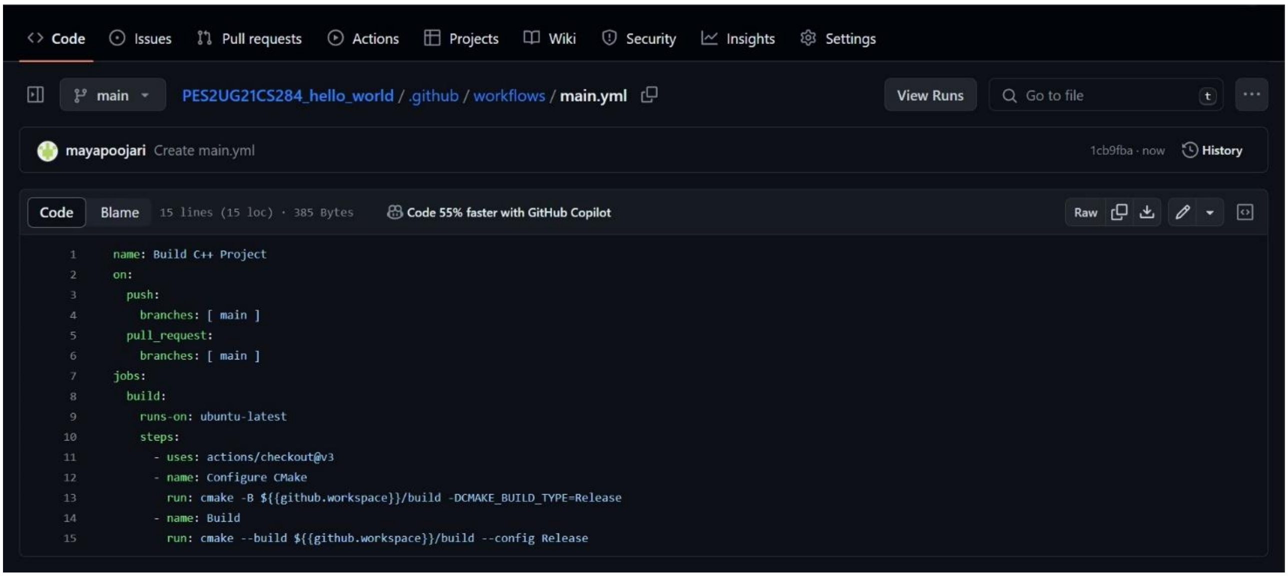
(a) Take a screenshot of the repository in the manner given below after the two files have been uploaded onto the repository and name it (a)



mayapoojari Update main.cpp ✓ 9cedb5e · 6 minutes ago 3 Commits

.github/workflows	Create main.yml	8 minutes ago
CMakeLists.txt	Describe the commit	15 minutes ago
main.cpp	Update main.cpp	6 minutes ago

(b) Take a screenshot of the code after it has been pasted onto the workflow and name it (b)



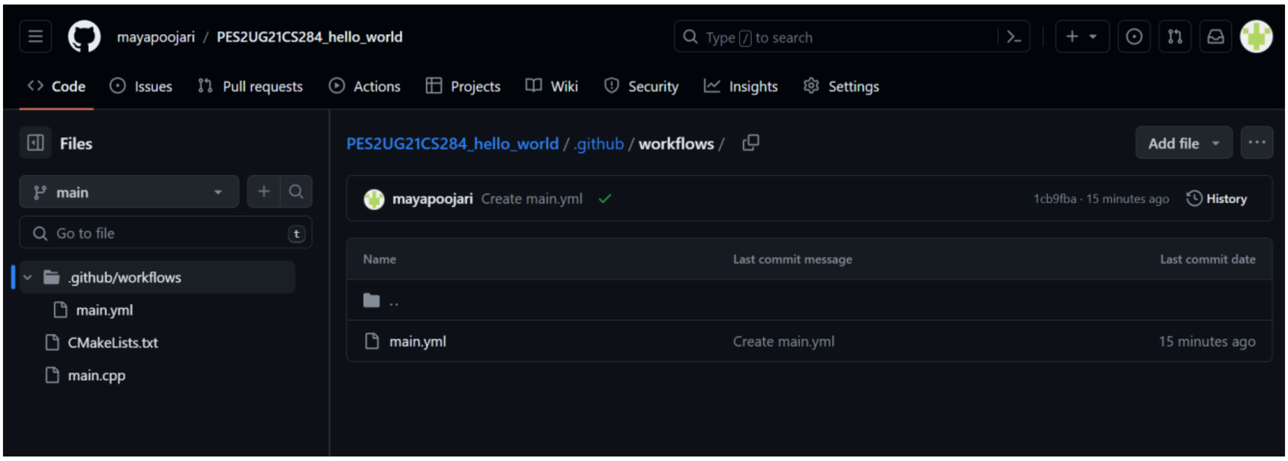
PES2UG21CS284_hello_world / .github / workflows / main.yml

mayapoojari Create main.yml 1cb9fba · now

15 lines (15 loc) · 385 Bytes Code 55% faster with GitHub Copilot

```yaml
name: Build C++ Project
on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - name: Configure CMake
        run: cmake -B ${{github.workspace}}/build -DCMAKE_BUILD_TYPE=Release
      - name: Build
        run: cmake --build ${{github.workspace}}/build --config Release
```

(c). Take a screenshot as given below indicating that the build is successful and name it



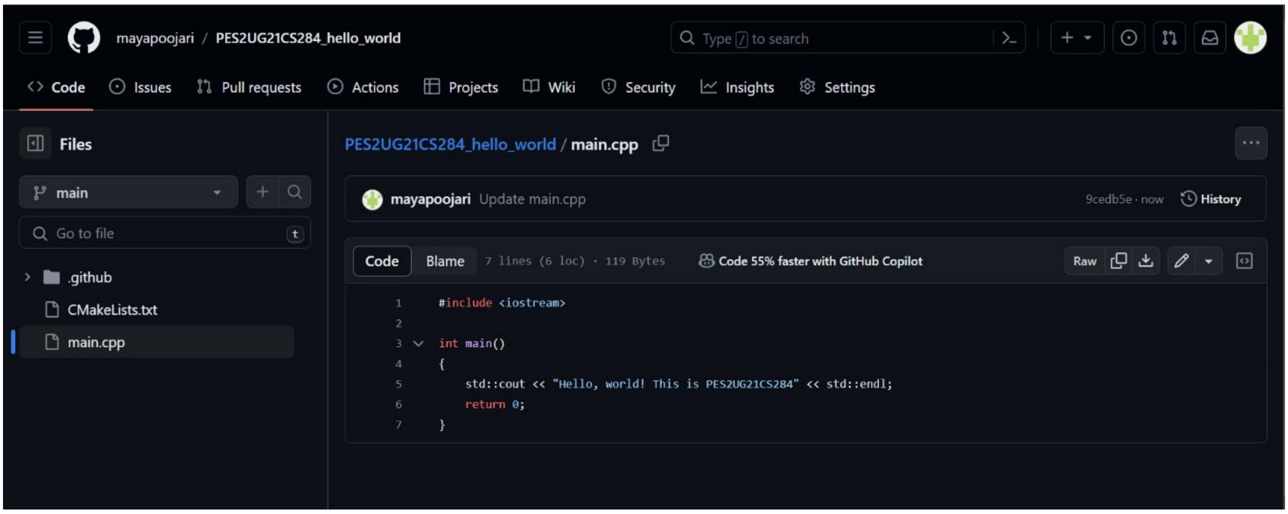
PES2UG21CS284_hello_world / .github / workflows /

mayapoojari Create main.yml ✓ 1cb9fba · 15 minutes ago

.github/workflows
main.yml
CMakeLists.txt
main.cpp

Name	Last commit message	Last commit date
..		
main.yml	Create main.yml	15 minutes ago

(d) Take a screenshot as given below after making the change in the main.cpp file and name it



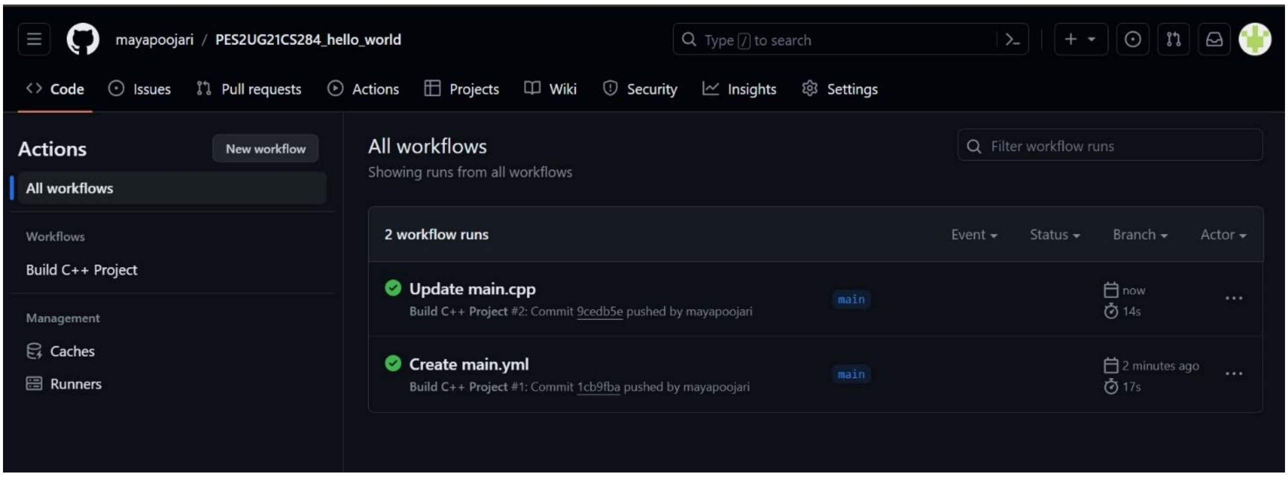
PES2UG21CS284_hello_world / main.cpp

mayapoojari Update main.cpp 9cedb5e · now

7 lines (6 loc) · 119 Bytes

```cpp
#include <iostream>

int main()
{
    std::cout << "Hello, world! This is PES2UG21CS284" << std::endl;
    return 0;
}
```

(e) Take a screenshot of the status of the build after it has passed and name it



All workflows

Showing runs from all workflows

2 workflow runs

✓ Update main.cpp
Build C++ Project #2: Commit 9cedb5e pushed by mayapoojari — main — now — 14s

✓ Create main.yml
Build C++ Project #1: Commit 1cb9fba pushed by mayapoojari — main — 2 minutes ago — 17s