COULD COMPUTING

Assignment 2 – Jenkins

(Creating a DevOps Pipeline, CI/CD tool)

SRN: PES1UG20CS151

NAME: Gautam R A

SEC: C

Task-1

Aim: Set up Jenkins using Docker.

Deliverables:

1. Screenshot of the running Docker Container after installing

Jenkins.

Text

Description automatically generated

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

Graphical user interface, text, application, email

Description automatically generated

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

Graphical user interface, application, Teams

Description automatically generated

Task-3

Aim: Sup 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

2. Status page of first job

3. Build History of Jenkins

4. Jenkins Dashboard

Deliverables

1. Console output of second job

2. Status page of first job

3. Build History of Jenkins

4. Jenkins Dashboard

Deliverables

1. Console output of second job

2. Status page of first job

3. Build History of Jenkins

4. Jenkins Dashboard

Deliverables

1. Console output of second job

2. Status page of first job

3. Build History of Jenkins

4. Jenkins Dashboard

Deliverables

1. Console output of second job

2. Status page of first job

3. Build History of Jenkins

4. Jenkins Dashboard

Deliverables

1. Console output of second job

2. Status page of first job

3. Build History of Jenkins

4. Jenkins Dashboard

Deliverables

1. Console output of second job

2. Status page of first job

3. Build History of Jenkins

4. Jenkins Dashboard

Deliverables:

1. Console output of second job

Graphical user interface, text, application, Teams

Description automatically generated

2. Status page of first job

Graphical user interface, text, application

Description automatically generated

3. Build History of Jenkins

Graphical user interface, application, Teams

Description automatically generated

4. Jenkins Dashboard

Graphical user interface, application

Description automatically generated

Task-4

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

In the previous tasks, we were polling changes from the repository at an interval of every 5

mins. It is an expensive approach. There is, however, a better approach. By adding a

Webhook trigger to your repository and connecting it to your Jenkins server, the instant you

commit a change to your repository, your job is automatically executed.

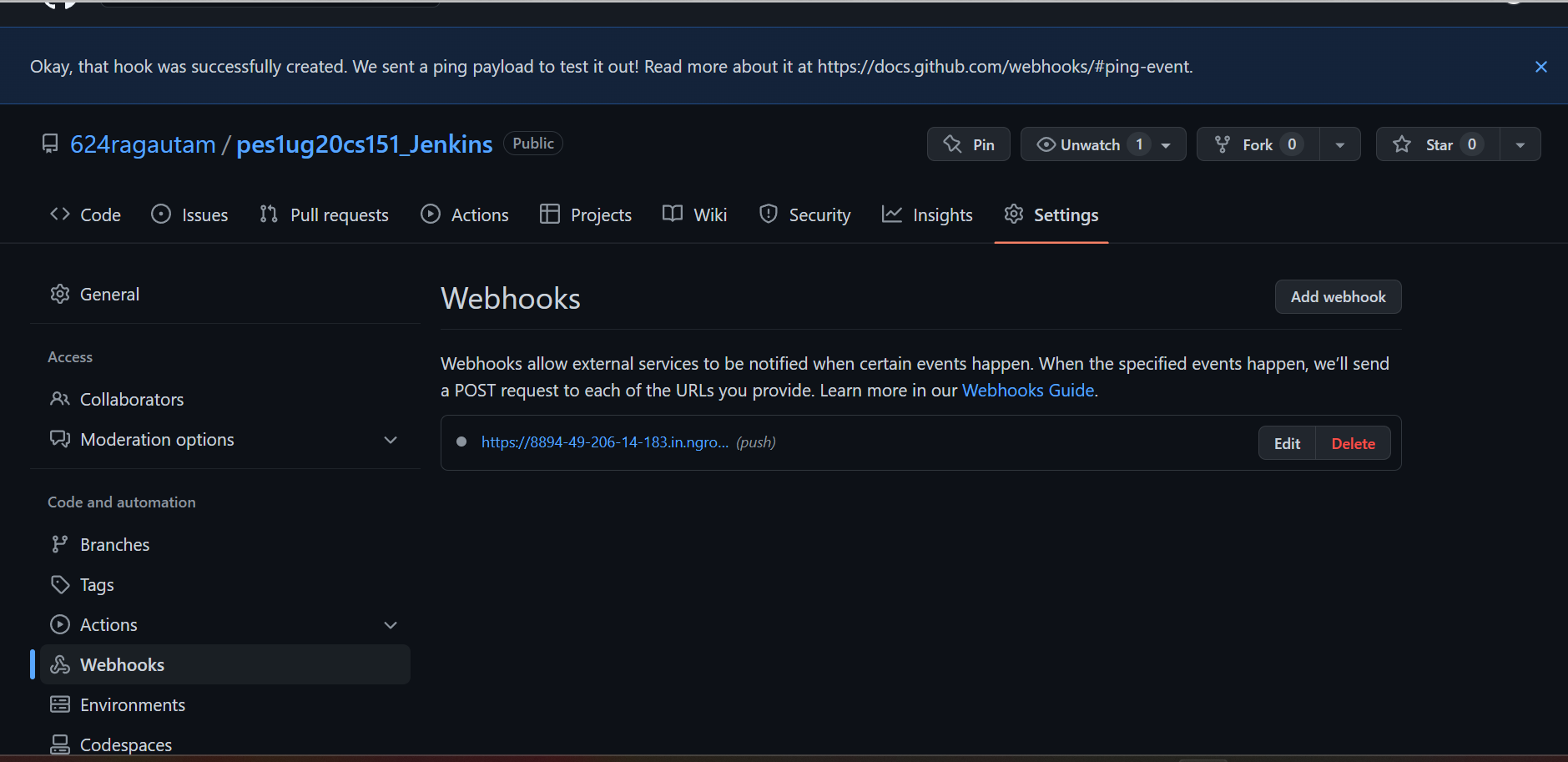
Webhooks allow external services to be notified when certain events happen. When those

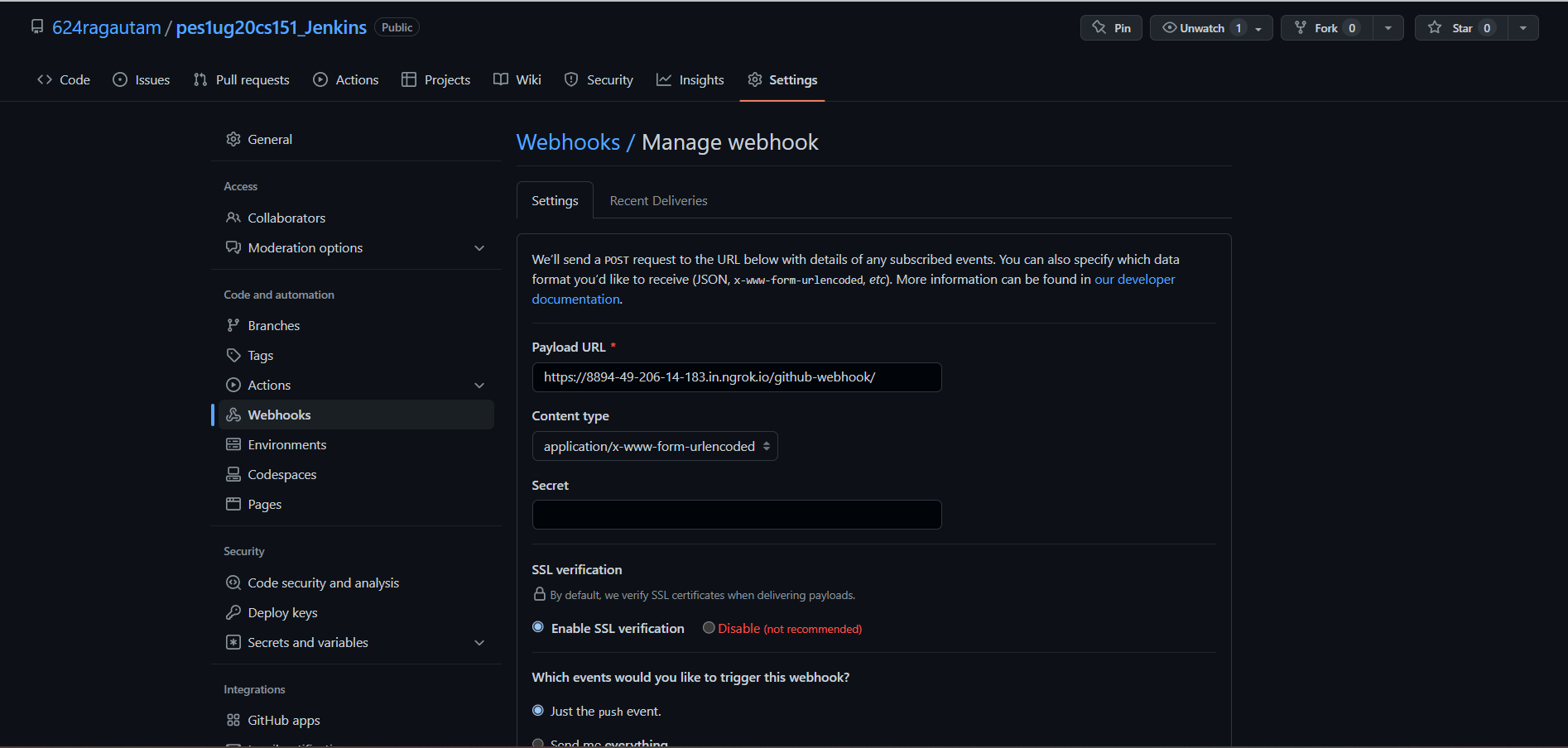
events happen, a POST request is sent to the designated URL.

Deliverables:

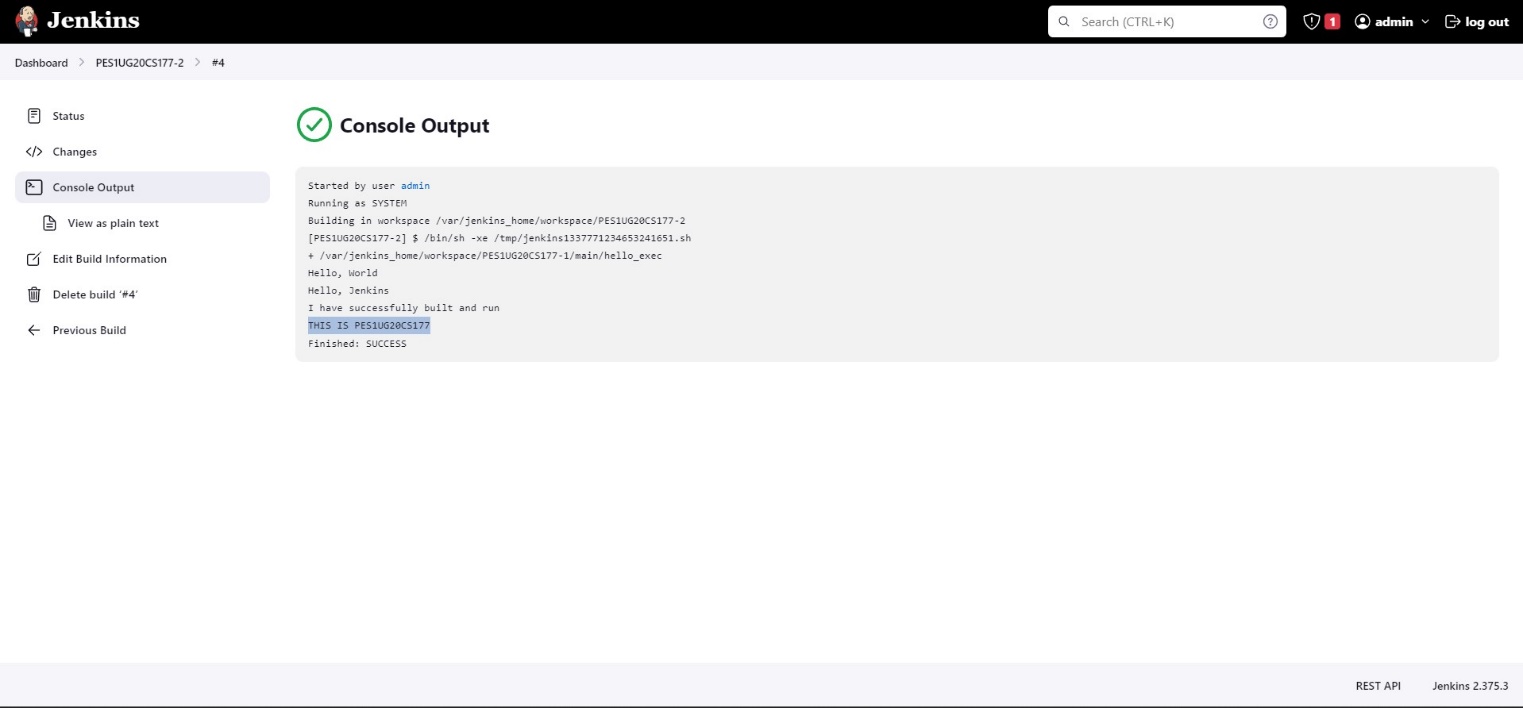
Deliverables

1. Webhook added to your GitHub repository





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



Deliverables:

Task-5

Aim: To create a basic Jenkins pipeline.

Deliverables:

Deliverables:

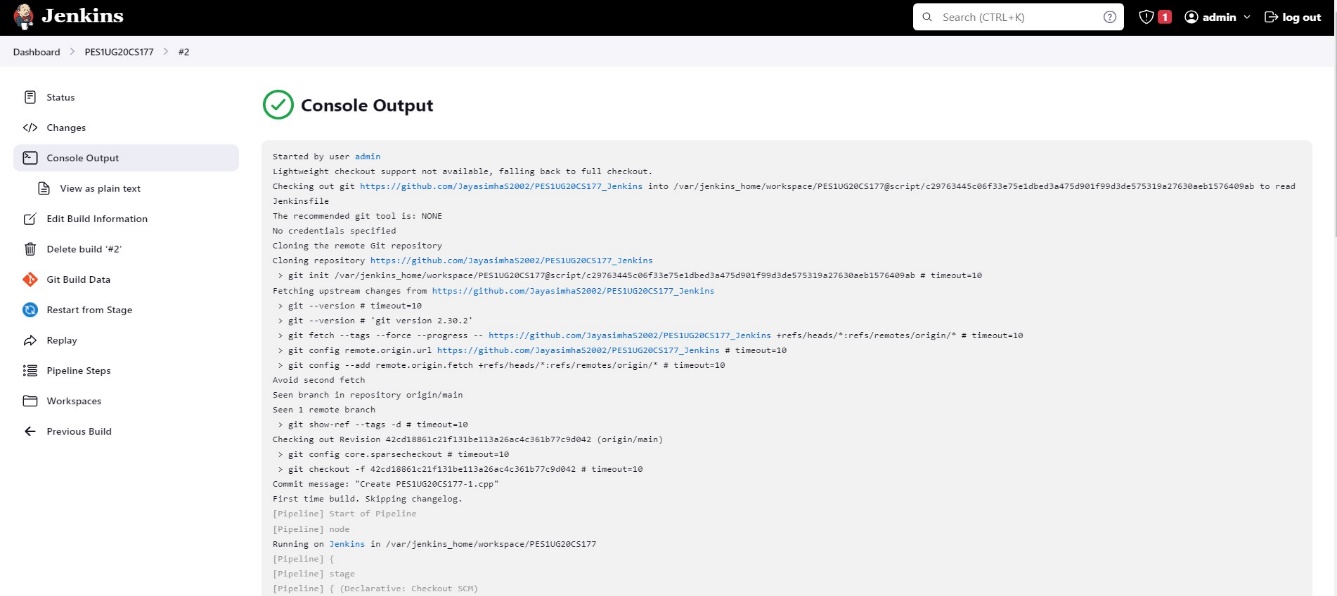
1. Code/script written to create basic pipeline using GitHub repository



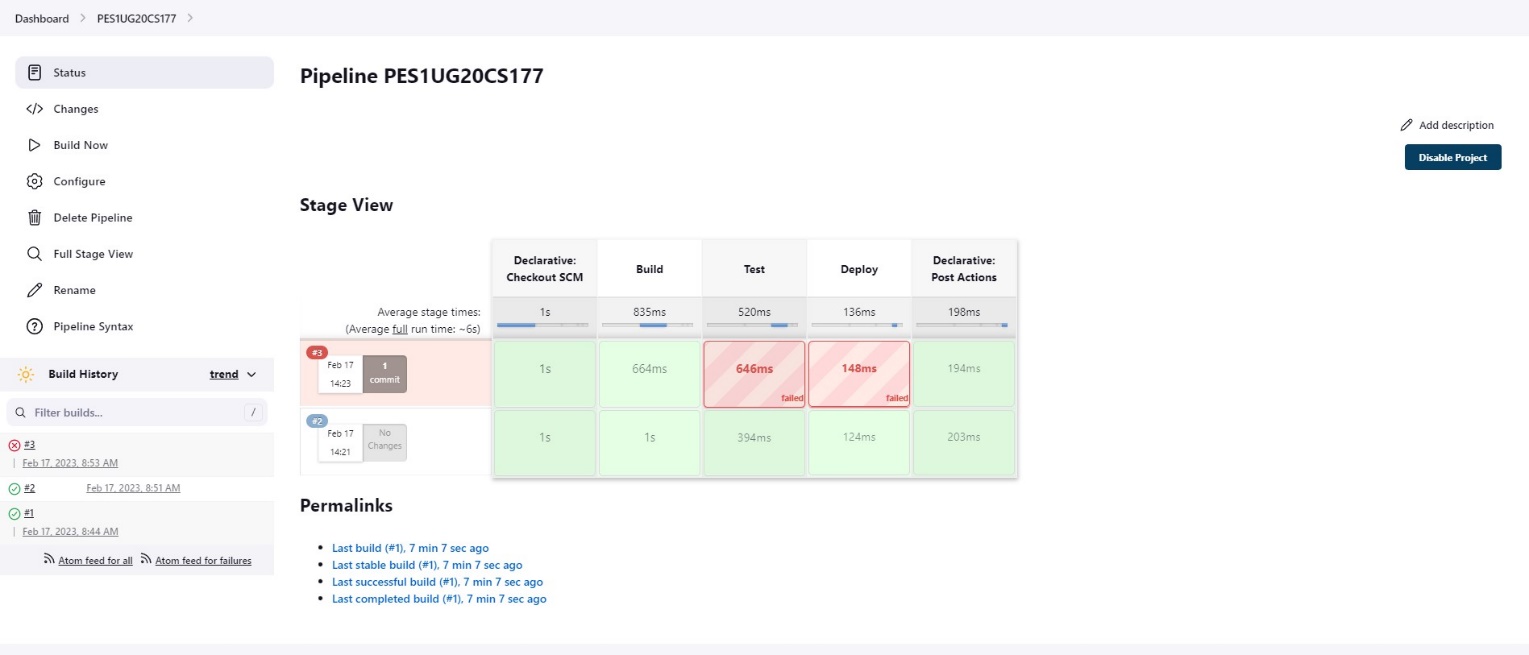
2. Output of working created pipeline, the screenshot should include

Stage view / Execution status of pipeline with all stages succeeded

Verify Declarative: Post Actions stage succeed for handling failures.



3. Console Output of the Pipeline



4. Link to the created GitHub repository

