



**Department of Computer Science**

**National University of Computer & Emerging Sciences**

## **DevOps-Assignment 03**



### **Implementation of Continuous Integration Pipelines for a Full-Stack Application Using GitHub Actions**

#### **Group Members**

- Saad Rehman (21F-9640)
- Ali Shaikh (21F-9631)
- Huzaifa Saqib (21F-9644)

#### **Course Instructor**

- Jawad Khalid Cheema

#### **GitHub Repository:**

[Click here](#)

## **Table of Contents**

<b>1. Introduction.....</b>	<b>3</b>
<b>2. Assignment Breakdown.....</b>	<b>3</b>
<b>3. CI Pipeline Implementation .....</b>	<b>3</b>
<b>4. Demonstration of CI Jobs .....</b>	<b>7</b>
<b>4.1 GitHub Actions History: .....</b>	<b>7</b>
<b>4.2 Pull Requests: .....</b>	<b>9</b>
<b>4.3 Issues Tracking: .....</b>	<b>11</b>
<b>5. Monitoring with Slack Incoming Webhook.....</b>	<b>12</b>
<b>5.1 Creating the Slack Incoming Webhook .....</b>	<b>12</b>
<b>5.2 Updating GitHub Secrets .....</b>	<b>13</b>
<b>5.3 Configuring GitHub Actions for Notifications .....</b>	<b>14</b>
<b>5.4 Testing the Notification System .....</b>	<b>14</b>
<b>6. Conclusion .....</b>	<b>15</b>

## **1. Introduction**

This report outlines the process and steps taken to complete the Continuous Integration (CI) assignment for my Final Year Project (FYP). The primary objective was to set up CI pipelines using GitHub Actions for a full-stack application, ensuring the automation of builds, tests, and deployments.

## **2. Assignment Breakdown**

### **1. Project Setup**

- Created a new repository on GitHub and transferred the existing FYP codebase into this repository.
- Ensured that the repository follows the best practices for project structure and organization.

### **2. Task Breakdown and Issue Creation**

- Analyzed the project requirements and divided them into smaller tasks.
- Created GitHub issues for each task to track progress and assigned them to team members as necessary.

### **3. Branching and Git Workflow**

- Followed proper Git workflows by creating feature branches for each task.
- Merged changes to the main branch using pull requests (PRs) to maintain a clean project history.

## **3. CI Pipeline Implementation**

### **1. Setting Up CI Pipeline with GitHub Actions**

- Created a `.github/workflows/ci.yml` file in the repository for GitHub Actions.

**CI Configuration:**

```
name: CI Pipeline

on:
  push:
    branches:
      - main # Trigger on pushes to the main branch
  pull_request:
    branches:
      - main # Trigger on pull requests to the main branch

jobs:
  # 1. Build Docker Images for Components
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
```

```

uses: actions/checkout@v3 # Check out the code in the repository

- name: Set up Docker Buildx
  uses: docker/setup-buildx-action@v2 # Set up Docker Buildx for building multi-platform images

- name: Cache Docker Layers
  uses: actions/cache@v3 # Cache Docker layers to speed up builds
  with:
    path: /tmp/.buildx-cache # Specify cache path
    key: ${ runner.os }-docker-${ github.sha } # Cache key
    restore-keys: |
      ${ runner.os }-docker-

- name: Login to DockerHub
  run: echo "${ secrets.DOCKER_PASSWORD }" | docker login -u "${ secrets.DOCKER_USERNAME }" --password-stdin # Log in to DockerHub

- name: Build Frontend Container
  run: docker build -t saadrehman17100/frontend:latest -f Dockerfile . # Build the frontend Docker image

- name: Build Backend Container (if applicable)
  run: docker build -t saadrehman17100/backend:latest -f Dockerfile . # Build the backend Docker image

- name: Push Frontend Image to DockerHub
  run: docker push saadrehman17100/frontend:latest # Push the frontend image to DockerHub

- name: Push Backend Image to DockerHub
  run: docker push saadrehman17100/backend:latest # Push the backend image to DockerHub

# 2. Run Unit Tests
unit-tests:
  runs-on: ubuntu-latest
  needs: build # Ensure this job runs after the build job
  steps:
    - name: Checkout code
      uses: actions/checkout@v3 # Check out the code

    - name: Install Dependencies
      run: npm install # Install project dependencies

    - name: Run Unit Tests
      run: npm test -- --json --outputFile=test-results.json # Run tests and output results to a JSON file

    - name: Upload Test Results
      uses: actions/upload-artifact@v3 # Upload test results as artifacts

```

```

with:
  name: test-results # Artifact name
  path: test-results.json # Path to the test results file

# 3. Linter Job (ESLint)
lint:
  runs-on: ubuntu-latest
  steps:
    - name: Checkout code
      uses: actions/checkout@v3 # Check out the code

    - name: Install ESLint
      run: npm install eslint --save-dev # Install ESLint as a development dependency

    - name: Run ESLint
      run: npx eslint ./ --ext .ts,.tsx --fix # Run ESLint to check code quality and automatically fix issues

# 4. SAST Job (Semgrep)
sast:
  runs-on: ubuntu-latest
  steps:
    - name: Checkout code
      uses: actions/checkout@v3

    - name: Run Semgrep for Security Analysis
      run: |
        docker run --rm --volume "$PWD:/src" returntocorp/semgrep semgrep --config=p/ci .

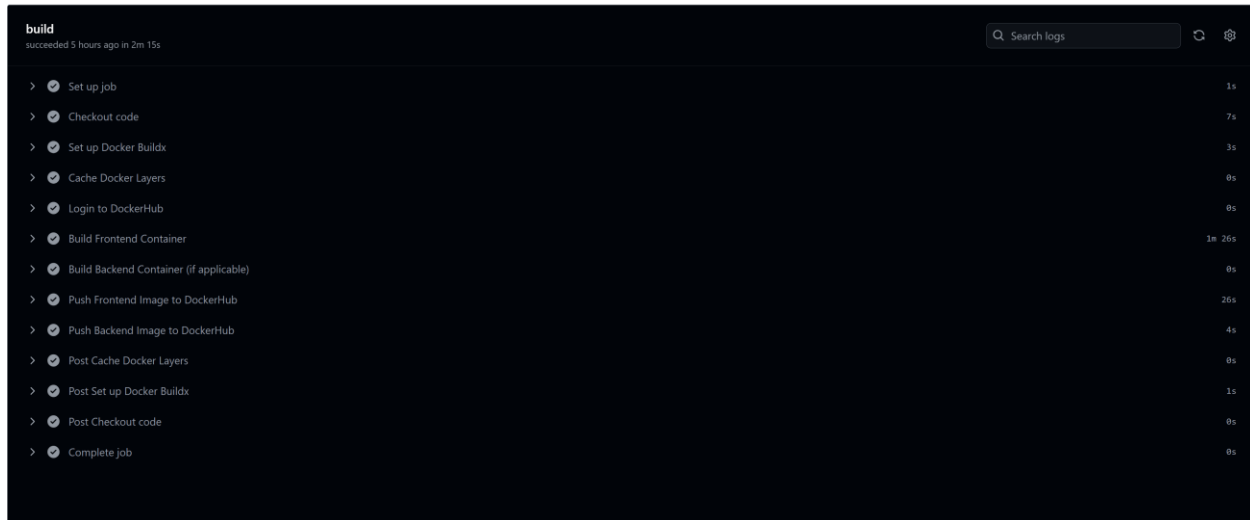
# 5. Slack Notification Job
notify:
  runs-on: ubuntu-latest
  needs: [build, lint, unit-tests, sast]
  steps:
    - name: Notify Slack
      run: |
        curl -X POST -H 'Content-type: application/json' --data '{
          "text": "CI Pipeline Notification: All jobs have completed successfully!"
        }' ${ secrets.SLACK_WEBHOOK_URL } # Use the secret

```

## 2. Commands for Testing Each Task

- **Task 1: Build Docker Containers**
  - Command to trigger the pipeline:

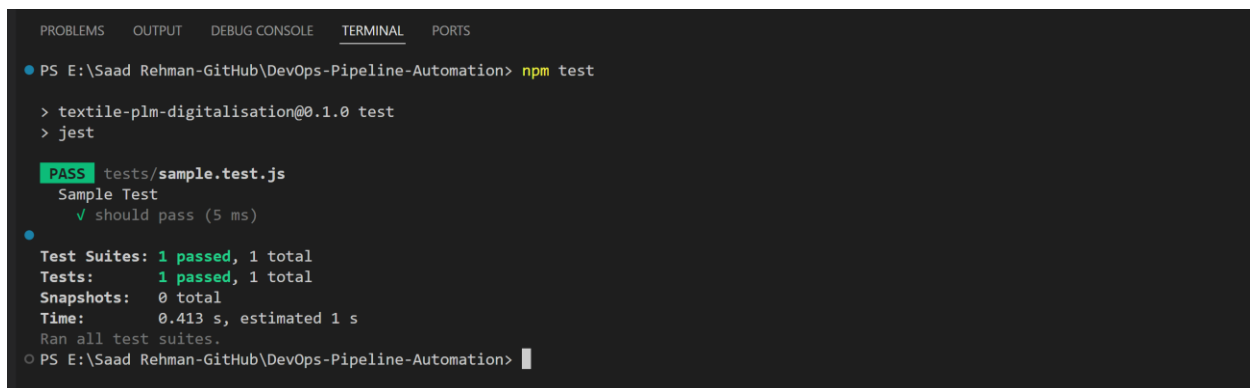
- `git push origin main`



### • Task 2: Run Unit Tests

- Command to trigger the pipeline:

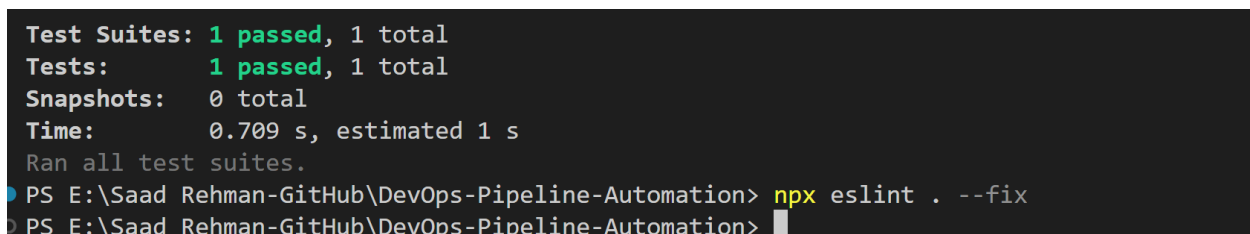
- `npm test`



### • Task 3: Run Linter and SAST Jobs

- Command to run linter:

- `npx eslint . --fix`



- Command to run SAST using Semgrep

## DevOps-Assignment03

- `docker run --rm --volume "$PWD:/src" returntocorp/semgrep semgrep --config=p/ci .`

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\Saad Rehman-GitHub\DevOps-Pipeline-Automation> docker run --rm --volume "$(Get-Location):/src" returntocorp/semgrep semgrep --config=p/ci .
METRICS: Using configs from the Registry (like --config=p/ci) reports pseudonymous rule metrics to semgrep.dev.
To disable Registry rule metrics, use "--metrics=off".
Using configs only from local files (like --config=xyz.yml) does not enable metrics.
More information: https://semgrep.dev/docs/metrics
```

### • Task 4: Push Containers to DockerHub

- Command to log in to DockerHub and push images:
  - `docker push saadrehman17100/frontend:latest`

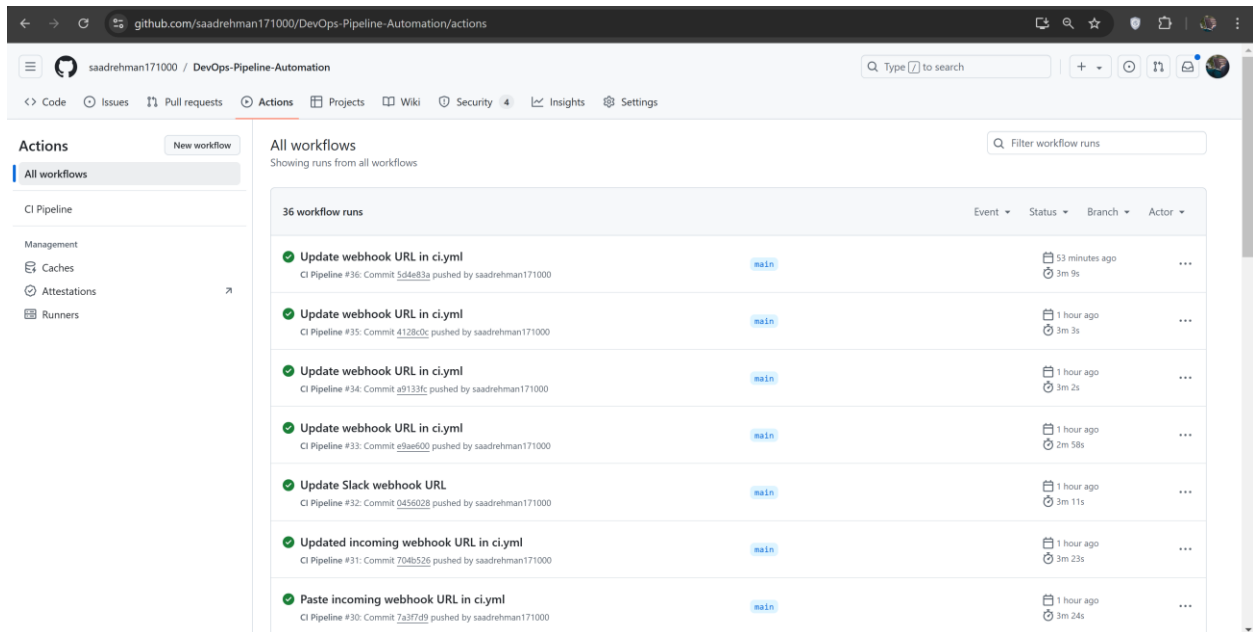
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\Saad Rehman-GitHub\DevOps-Pipeline-Automation> docker push saadrehman17100/frontend:latest
The push refers to repository [docker.io/saadrehman17100/frontend]
a36a89153f48: Pushed
d0d97deec1f9: Pushed
f3af7a5da10a: Pushed
d25fdc0186e7: Mounted from saadrehman17100/myapp-image
e2be10e97665: Layer already exists
06fd85419b65: Layer already exists
f58c462fa079: Layer already exists
63ca1fbb43ae: Layer already exists
latest: digest: sha256:85ac11123a6c9bd4d9bce6a1a74bc385fc291c779d60116f2abdcdf50392a8834 size: 1996
PS E:\Saad Rehman-GitHub\DevOps-Pipeline-Automation>
```

## 4. Demonstration of CI Jobs

### 4.1 GitHub Actions History:

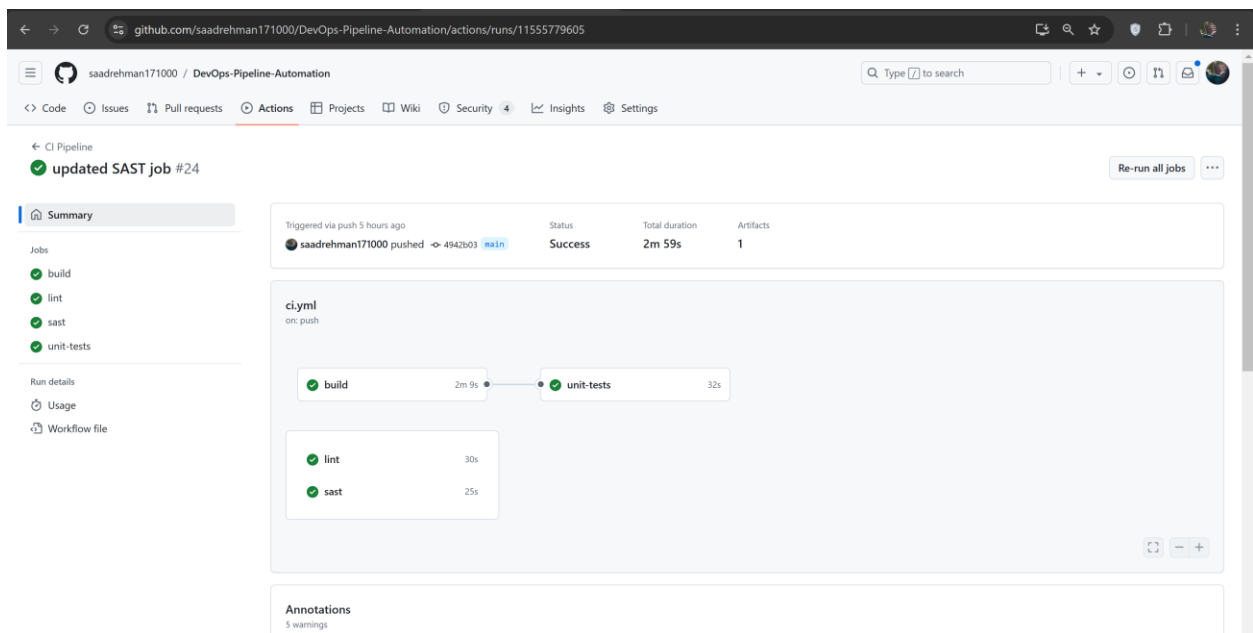
github.com/saadrehman171000/DevOps-Pipeline-Automation/actions			
<div>Actions</div> <div>New workflow</div> <div>All workflows</div> <div>CI Pipeline</div> <div>Management</div> <div>Caches</div> <div>Attestations</div> <div>Runners</div>	CI Pipeline #21: Pull request #5 opened by zailfish701		
	✓ Merge pull request #4 from saadrehman171000/issues	main	5 hours ago
	CI Pipeline #20: Commit 83efe51 pushed by saadrehman171000		
	✓ Update tailwind.config.ts	Issues	6 hours ago
	CI Pipeline #19: Pull request #4 opened by shailikh		
	✓ Updated ci.yml	main	6 hours ago
	CI Pipeline #18: Commit 5a8bc99 pushed by saadrehman171000		
	✗ Updated ci.yml	main	6 hours ago
	CI Pipeline #17: Commit e1b88a7 pushed by saadrehman171000		
	✗ Updated ci.yml	main	6 hours ago
	CI Pipeline #16: Commit 7691222 pushed by saadrehman171000		
	✗ Updated ci.yml	main	6 hours ago
	CI Pipeline #15: Commit cbe3aac pushed by saadrehman171000		
	✗ Updated ci.yml	main	6 hours ago
	CI Pipeline #14: Commit 79bc31e pushed by saadrehman171000		
	✗ Updated tailwind file	main	6 hours ago
	CI Pipeline #13: Commit fa7a91d pushed by saadrehman171000		
	✗ Update CI pipeline and tailwind file	main	6 hours ago
	CI Pipeline #12: Commit 91e1898 pushed by saadrehman171000		

## DevOps-Assignment03



This screenshot shows the 'All workflows' page in a GitHub repository. The left sidebar contains navigation links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. Under the 'Actions' tab, there are links for 'All workflows', 'CI Pipeline', 'Management', 'Caches', 'Attestations', and 'Runners'. The main area displays a list of 36 workflow runs. Each run entry includes a green checkmark, the workflow name 'Update webhook URL in ci.yml', the CI Pipeline number, the commit hash, the actor's name, the branch 'main', and the time since the run completed. The runs are sorted by time, with the most recent at the top.

Event	Status	Branch	Actor
Update webhook URL in ci.yml	Success	main	saadrehman171000
Update webhook URL in ci.yml	Success	main	saadrehman171000
Update webhook URL in ci.yml	Success	main	saadrehman171000
Update webhook URL in ci.yml	Success	main	saadrehman171000
Update Slack webhook URL	Success	main	saadrehman171000
Updated incoming webhook URL in ci.yml	Success	main	saadrehman171000
Paste incoming webhook URL in ci.yml	Success	main	saadrehman171000



This screenshot shows the 'Run details' page for a specific workflow run. The left sidebar has links for 'Summary', 'Jobs', 'Run details', 'Usage', and 'Workflow file'. The 'Summary' tab is selected, showing a green checkmark and the text 'updated SAST job #24'. The main area displays the workflow's execution details, including the trigger 'Triggered via push 5 hours ago', the actor 'saadrehman171000', the status 'Success', the total duration '2m 59s', and the number of artifacts '1'. Below this, a diagram shows the workflow steps: 'build' (2m 9s) and 'unit-tests' (32s). A separate box shows 'lint' (30s) and 'sast' (25s). At the bottom, there is a section for 'Annotations' with 5 warnings.

Triggered via push 5 hours ago

saadrehman171000 pushed -> 4942b03 main

Status: Success

Total duration: 2m 59s

Artifacts: 1

ci.yml on: push

build 2m 9s

unit-tests 32s

lint 30s

sast 25s

Annotations: 5 warnings



## DevOps-Assignment03

The screenshot shows a GitHub Actions workflow run for the repository 'saadrehman171000 / DevOps-Pipeline-Automation'. The workflow is named 'Update Slack webhook URL #32'. The run was triggered by a push to the 'main' branch by user 'saadrehman171000' at 04:56:28. The status is 'Success', the total duration is '3m 11s', and there is 1 artifact.

The workflow file 'ci.yml' is shown, triggered on 'push'. The jobs are:

- build**: 2m 17s
- unit-tests**: 27s
- notify**: 2s
- lint**: 34s
- sast**: 28s

The jobs are connected in a sequence: build → unit-tests → notify. The lint and sast jobs are parallel to build and also feed into unit-tests.

## 4.2 Pull Requests:

The screenshot shows the GitHub Pull Requests page for the repository 'saadrehman171000 / DevOps-Pipeline-Automation'. The page displays a list of pull requests with filters set to 'is:pr is:closed'.

Filters:  Labels:  Milestones:  [New pull request](#)

Clear current search query, filters, and sorts

<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	0 Open	3 Closed	Author	Label	Projects	Milestones	Reviews	Assignee	Sort
<input type="checkbox"/>	<input checked="" type="checkbox"/>				<a href="#">Update ci.yml</a> ✓	#5 by zaifishiekh01					merged 5 hours ago
<input type="checkbox"/>	<input checked="" type="checkbox"/>				<a href="#">Update tailwind.config.ts</a> ✓	#4 by shailikh					merged 5 hours ago
<input type="checkbox"/>	<input checked="" type="checkbox"/>				<a href="#">updated ci.yml</a> ✗	#1 by saadrehman171000					merged 9 hours ago

ProTip! Exclude your own issues with `-author:saadrehman171000`

## DevOps-Assignment03

This screenshot shows a GitHub pull request titled "Update ci.yml #5" in the repository "saadrehman171000 / DevOps-Pipeline-Automation". The pull request is merged and closed. It shows a commit from the "features" branch to the "main" branch, merged by "saadrehman171000" 5 hours ago. The commit message is "Update ci.yml". The pull request is verified and has 4 checks passed. A message indicates the pull request was successfully merged and closed, and the "features" branch can be safely deleted. The right sidebar shows no reviews, no assignees, and no labels. The bottom section has a "Delete branch" button and a "Add a comment" section.

Update ci.yml #5

Merged saadrehman171000 merged 1 commit into main from features 5 hours ago

Conversation 0 Commits 1 Checks 4 Files changed 1

zaiffishiekh01 commented 5 hours ago

No description provided.

Update ci.yml

Verified ✓ 2da2bf3

saadrehman171000 merged commit ec5b5de into main 5 hours ago

4 checks passed

Pull request successfully merged and closed

You're all set—the features branch can be safely deleted.

Delete branch

Add a comment

Write Preview

Add your comment here...

This screenshot shows a GitHub pull request titled "Update tailwind.config.ts #4" in the repository "saadrehman171000 / DevOps-Pipeline-Automation". The pull request is merged and closed. It shows a commit from the "Issues" branch to the "main" branch, merged by "saadrehman171000" 5 hours ago. The commit message is "Update tailwind.config.ts". The pull request is verified and has 4 checks passed. A message indicates the pull request was successfully merged and closed, and the "Issues" branch can be safely deleted. The right sidebar shows no reviews, no assignees, and no labels. The bottom section has a "Delete branch" button and a "Add a comment" section.

Update tailwind.config.ts #4

Merged saadrehman171000 merged 1 commit into main from Issues 5 hours ago

Conversation 0 Commits 1 Checks 4 Files changed 1

shaiikh commented 6 hours ago

No description provided.

Update tailwind.config.ts

Verified ✓ f690611

saadrehman171000 merged commit a3efe51 into main 5 hours ago

4 checks passed

Pull request successfully merged and closed

You're all set—the Issues branch can be safely deleted.

Delete branch

Add a comment

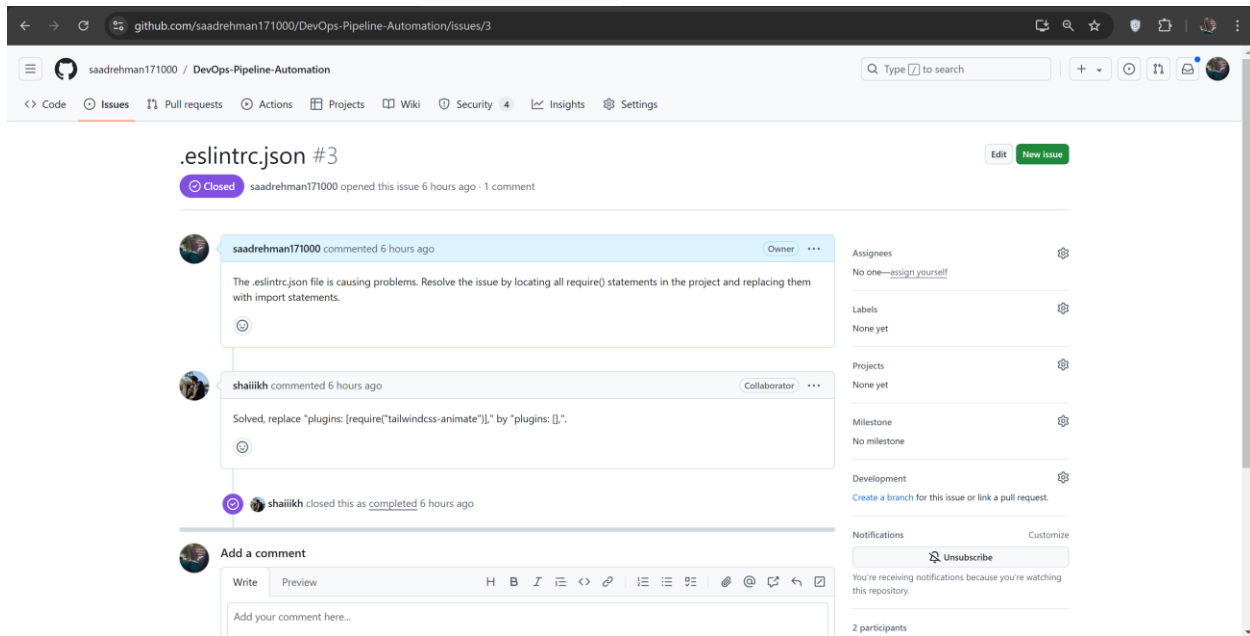
Write Preview

Add your comment here...

## 4.3 Issues Tracking:

The screenshot shows the GitHub Issues page for the repository 'saadrehman171000 / DevOps-Pipeline-Automation'. The URL in the browser is 'github.com/saadrehman171000/DevOps-Pipeline-Automation/issues?q=is%3Aissue+is%3Aclosed'. The page has a navigation bar with links to Code, Issues (selected), Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A search bar is present with the text 'Type / to search'. Below the navigation bar, there is a message: 'Label issues and pull requests for new contributors. Now, GitHub will help potential first-time contributors discover issues labeled with good first issue.' Below this, there are filters for 'is:issue is:closed', a 'Labels' dropdown, a 'Milestones' dropdown, and a 'New issue' button. A table of issues is shown with columns for '0 Open' and '2 Closed'. The closed issues are: '.eslintrc.json' (commented by saadrehman171000, closed 6 hours ago) and 'Semgrep-ci.yml' (commented by saadrehman171000, closed 6 hours ago). A 'ProTip!' message at the bottom says: 'Type / on any issue or pull request to go back to the pull request listing page.'

The screenshot shows the details of GitHub Issue #2 for the repository 'saadrehman171000 / DevOps-Pipeline-Automation'. The issue title is 'Semgrep-ci.yml #2'. The issue is marked as 'Closed'. The issue was opened by 'saadrehman171000' 6 hours ago. The issue description is 'Configure semgrep issue, and resolve'. The issue has two comments: one by 'saadrehman171000' (Owner) and one by 'zaiffishiekh01' (Collaborator). The comment by 'zaiffishiekh01' says 'Solved, commit and push this updated ci.yml'. The issue was closed by 'zaiffishiekh01' as 'completed' 6 hours ago. The right sidebar shows the 'Assignees' section with 'No one—assign yourself', the 'Labels' section with 'None yet', the 'Projects' section with 'None yet', the 'Milestone' section with 'No milestone', the 'Development' section with 'Create a branch for this issue or link a pull request', and the 'Notifications' section with 'Unsubscribe' and 'Customize'. The bottom of the page shows a '2 participants' section.



## 5. Monitoring with Slack Incoming Webhook

To implement monitoring for the CI pipeline, an incoming webhook was created in Slack, allowing notifications to be sent to a designated channel. This setup helps track the status of CI jobs and provides immediate feedback on job execution.

### 5.1 Creating the Slack Incoming Webhook

1. **Accessing Slack API:** The process started by navigating to the Slack API website and selecting the appropriate workspace.
2. **Creating a New App:** A new app was created within the Slack workspace, followed by configuring the app settings.
3. **Adding Incoming Webhook:** The incoming webhook feature was enabled, which provided a unique webhook URL. This URL is crucial for sending messages to the designated Slack channel.

## DevOps-Assignment03

The top screenshot shows the Slack API 'Incoming Webhooks' settings page. The 'Activate Incoming Webhooks' toggle is turned on. The page explains that incoming webhooks are a simple way to post messages from external sources into Slack. It also mentions that adding incoming webhooks requires a bot user and that a new Webhook URL will be generated each time the app is installed.

The bottom screenshot shows the 'Webhook URLs for Your Workspace' section. It provides instructions on how to dispatch messages with a webhook URL. Below the instructions, there is a sample curl request to post to a channel:

```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello, World!"}' https://hooks.slack.com/services/T07U6R3ZJ6M/B07TE6R0AQP/ucf1WAW5CgMA3S8SCU9arti
```

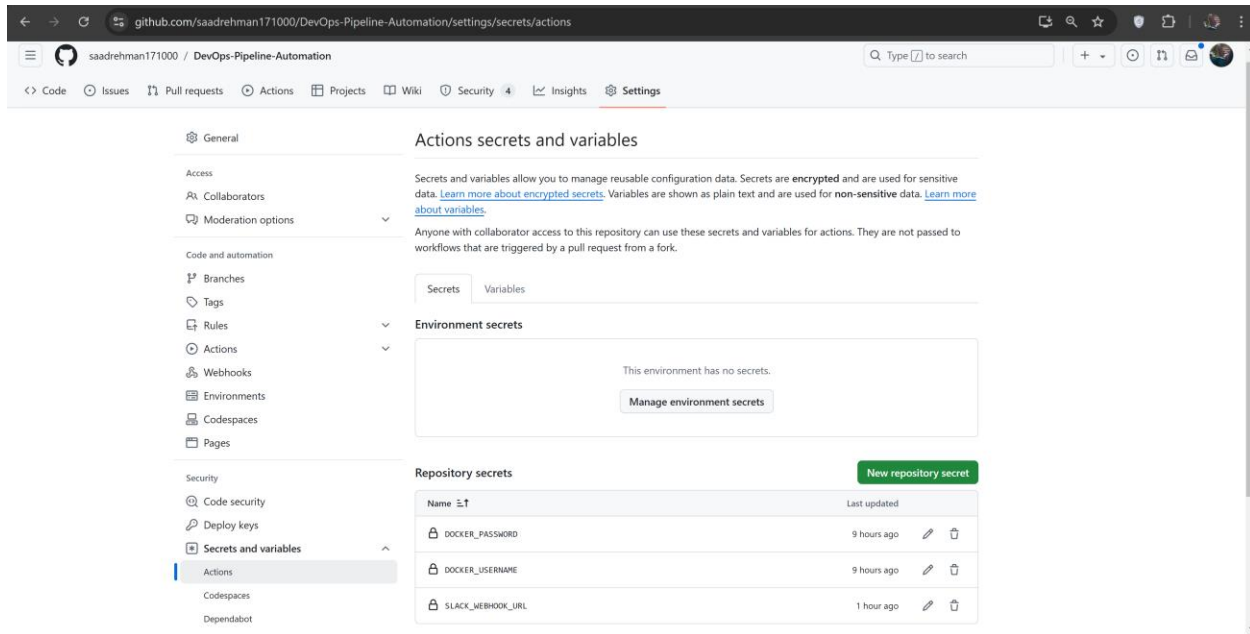
Below the curl request, there is a table of active webhooks:

Webhook URL	Channel	Added By
<a href="https://hooks.slack.com/services/T07U6R3ZJ6M/B07TE6R0AQP/ucf1WAW5CgMA3S8SCU9arti">https://hooks.slack.com/services/T07U6R3ZJ6M/B07TE6R0AQP/ucf1WAW5CgMA3S8SCU9arti</a>	#automation	saadrehman17100 Oct 28, 2024
<a href="https://hooks.slack.com/services/T07U6R3ZJ6M/B07TE6R0AQP/ucf1WAW5CgMA3S8SCU9arti">https://hooks.slack.com/services/T07U6R3ZJ6M/B07TE6R0AQP/ucf1WAW5CgMA3S8SCU9arti</a>	#automation	saadrehman17100 Oct 28, 2024

At the bottom of the table, there is a button labeled 'Add New Webhook to Workspace'.

## 5.2 Updating GitHub Secrets

1. **Navigating to Repository Settings:** The GitHub repository settings were accessed to add the new webhook URL as a secret.
2. **Adding Secret:** Under the "Secrets and variables" section, the webhook URL was pasted into the secret field labeled `SLACK_WEBHOOK_URL`. This ensures that the webhook URL is secure and not exposed in the code.



### 5.3 Configuring GitHub Actions for Notifications

1. **Editing the CI Configuration File:** The CI configuration file (ci.yml) was updated to include a job that sends notifications to Slack upon completion of other jobs in the pipeline.
2. **Adding the Notify Step:** A notify step was defined in the CI configuration to trigger a curl command, which sends a notification to the Slack channel using the incoming webhook URL.

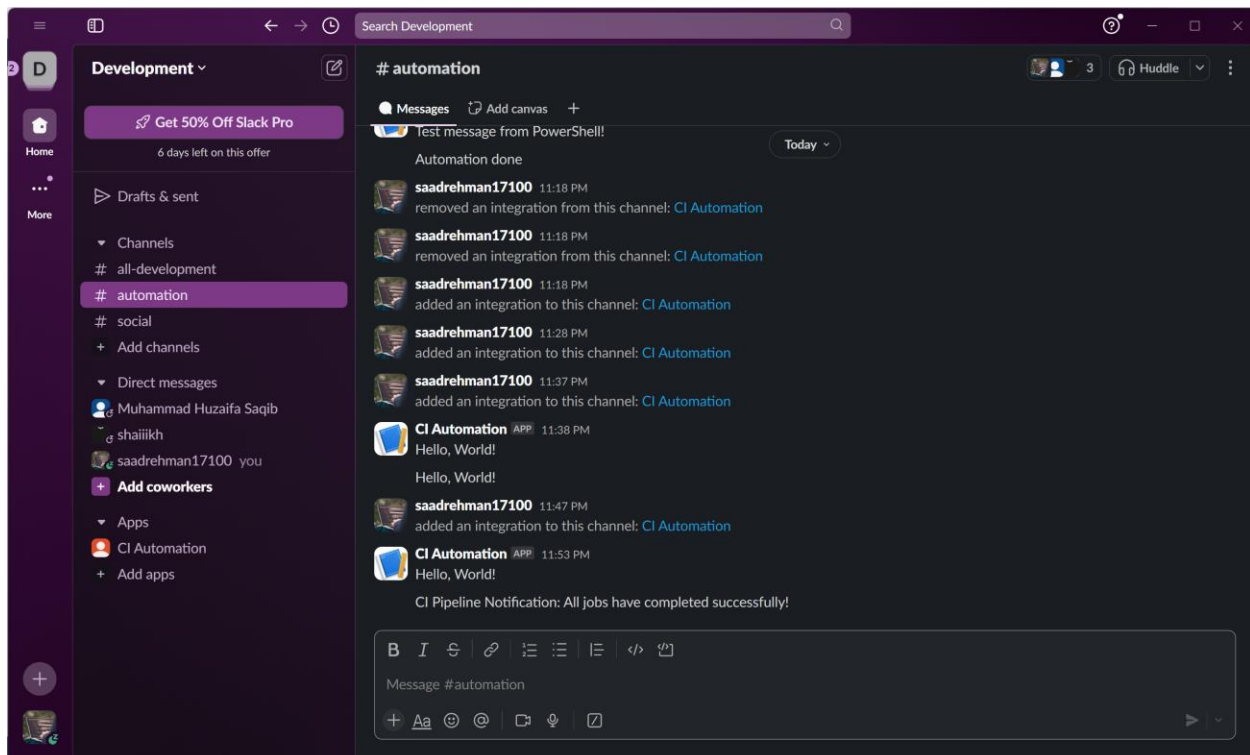
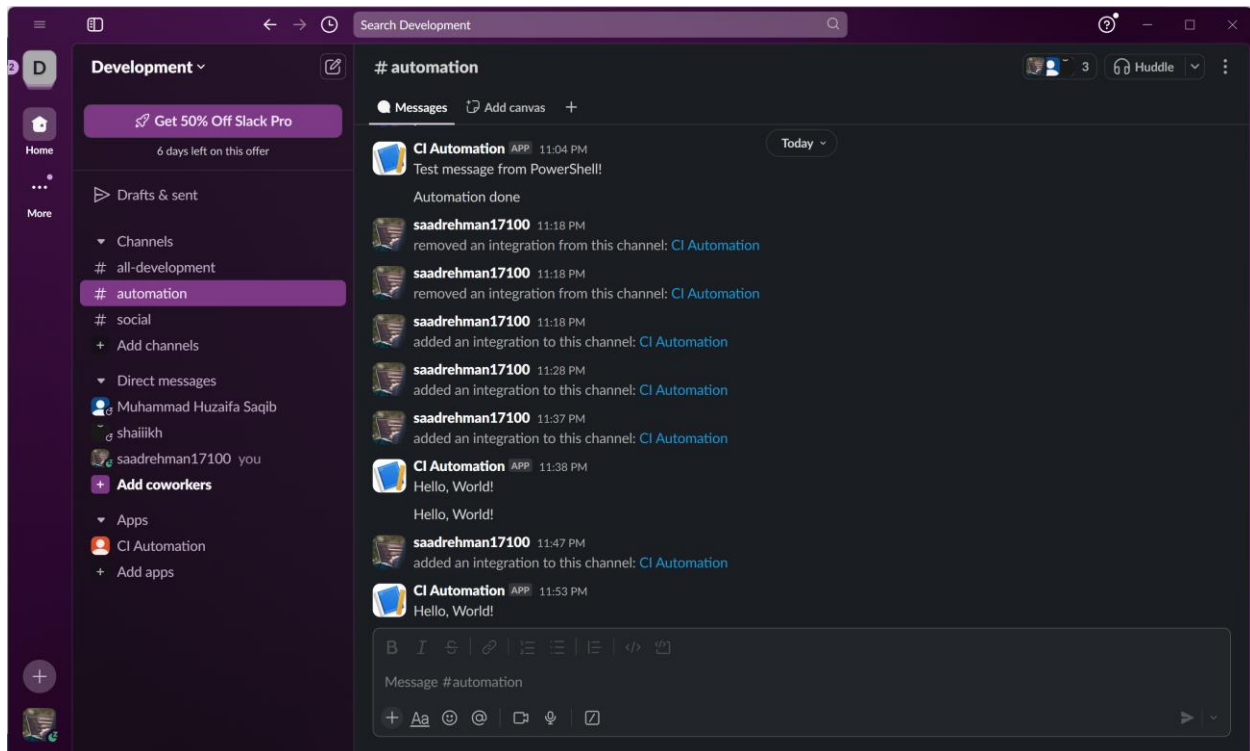
#### Command:

```
.github > workflows > ! ci.yml
11 jobs:
89 # 5. Slack Notification Job
90 notify:
91   runs-on: ubuntu-latest
92   needs: [build, lint, unit-tests, sast]
93   steps:
94     - name: Notify Slack
95       run: |
96         curl -X POST -H 'Content-type: application/json' --data '{
97           "text": "CI Pipeline Notification: All jobs have completed successfully!"
98         }' ${ secrets.SLACK_WEBHOOK_URL } # Use the secret
```

### 5.4 Testing the Notification System

1. **Triggering CI Pipeline:** After completing the setup, the CI pipeline was triggered by making a commit to the main branch or creating a pull request.
2. **Verifying Notifications in Slack:** Upon successful execution of the CI jobs, a notification was sent to the specified Slack channel, confirming the completion of the jobs.

## DevOps-Assignment03



## 6. Conclusion

The completion of the CI pipeline assignment demonstrated a comprehensive understanding of continuous integration practices using modern tools and technologies. By setting up a robust pipeline

utilizing GitHub Actions, Docker, and additional tools like Semgrep for security analysis, the project successfully achieved its objectives.

The following key aspects were effectively addressed throughout the assignment:

1. **Structured Breakdown:** The project was carefully divided into manageable tasks, enabling clear tracking and assignment through GitHub issues. This approach facilitated collaboration among team members and ensured that each aspect of the project was thoroughly covered.
2. **CI Pipeline Implementation:** The CI pipeline was implemented to automate critical tasks, including building Docker containers for frontend and backend components, running unit tests, conducting linting and SAST checks, and pushing images to Docker Hub. Each job was optimized for performance, utilizing caching features and adhering to best practices.
3. **Demonstration of CI Jobs:** Clear documentation of CI job executions, along with detailed histories of GitHub Actions, pull requests, and issues tracking, provided transparency and a comprehensive overview of the development process. This information is vital for assessing the pipeline's efficiency and effectiveness.
4. **Monitoring through Slack Integration:** The integration of Slack incoming webhooks enabled real-time notifications regarding the status of CI jobs. This feature enhances team communication and ensures prompt awareness of job outcomes, thereby fostering a proactive development environment.

Overall, the assignment not only fulfilled the outlined requirements but also provided valuable insights into CI/CD practices, team collaboration, and project management. The skills developed during this project are essential for future endeavors in software development and DevOps, paving the way for continuous improvement in project delivery and team dynamics. The successful implementation of this assignment showcases the ability to integrate modern tools to streamline workflows and enhance productivity within a development team.