

# Final Project - Overview and Review Criteria

Estimated Time Needed: 30 minutes

## Project Overview

In the final project, you will apply the CI/CD concepts and skills you've learned throughout the course. Using a provided sample application and access to an OpenShift Cluster, you will build and deploy a complete CI/CD pipeline. This hands-on assignment will guide you through setting up GitHub Actions for automation, using Tekton for task management, and deploying your application using an OpenShift pipeline. You will have 2 options to perform this lab out of which you need to complete the submissions either using Python lab or Javascript lab.

This course offers two lab options—JavaScript and Python. We have already included a Lab Options by Programming Language reading at the start of the course; please review it before beginning this reading:

[Reading: Lab Options by Programming Language](#)

## Evaluation Criteria – 20 Points

You can submit your project deliverables in one of the following ways:

### Option 1: AI-Graded Submission and Evaluation

When you choose Option 1, you will be redirected to an AI tool where you can upload your deliverables, which may include URLs, terminal outputs, code snippets, or screenshots. You will then receive an AI-generated grade that will automatically reflect on your Coursera progress page.

### Option 2: Peer-Graded Submission and Evaluation

When you choose Option 2, you will upload your deliverables—such as URLs, terminal outputs, code snippets, or screenshots—through the My Submission section. Your submission will then be reviewed either by your peers or by Coursera's AI grader.

We recommend using Option 1 for faster grading. However, if you face any issues or cannot access it, you may use Option 2 instead.

If you encounter any grading problems, please reach out to the Course Team through the Discussion Forums.

Please find the details of the Grading Criteria below:

#### ▼ Criteria for Option 1: AI-Graded Submission and Evaluation:

##### Task 1:

Submit the GITHUB URL of the README.md file that contains the Project name details. (2 points)

##### Task2:

Provide the GitHub URL of .github/workflows/workflow.yml showing the code snippet for the Lint with flake8 step or ESLint and the code snippet for the Run unit tests with nose step or Jest-test(4 points)

##### Task3:

Provide the GitHub URL of `.tekton/tasks.yml` showing the code snippet for the cleanup task and the code snippet for the nose task or Jest-test(4 points)

**Task 4:**

Provide the screenshot of the OpenShift PersistentVolumeClaim details in a file named `oc-pipelines-console-pvc-details.png`(2 points)

**Task 5:**

Provide the text of terminal output named `cicd-github-validate(.png)` showing details of GitHub actions running successfully in the actions workflow containing all the steps.(2 points)

**Task 6:**

Provide the screenshot showing details of the OpenShift Pipeline `oc-pipelines-oc-final.png`(2 points)

**Task7:**

Provide the screenshot showing details of the OpenShift Pipeline running successfully in a file named `oc-pipelines-oc-green.png`(2 points)

**Task 8:**

Provide the text response saved and named as `oc-pipelines-app-logs(.png)` showing details of the OpenShift application logs(2 points)

▼ Criteria for Option 2: Peer-Graded Submission and Evaluation:

**Task 1:**

The GitHub repo URL that you pushed your changes to. Should be of the format [https://github.com/{your\\_github\\_account}/ci-cd-final-project.git](https://github.com/{your_github_account}/ci-cd-final-project.git)

**Task 2:**

Provide the GitHub URL of the `.github/workflows/workflow.yml` file showing the code snippet for the linting step.

**Task 3:**

Provide the GitHub URL of the `.github/workflows/workflow.yml` file showing the code snippet for the test step.

**Task 4:**

Provide the GitHub URL of the `.tekton/tasks.yml` file showing the code snippet for the cleanup task.

**Task 5:**

Provide the GitHub URL of the `.tekton/tasks.yml` file showing the code snippet for the nose test task.

**Task 6:**

Screenshot showing OpenShift PVC details. Name this file `oc-pipelines-console-pvc-details(.png)`

**Task 7:**

Screenshot showing GitHub actions running successfully. Name this file `cicd-github-validate`.

**Task 8:**

Screenshot showing details of the OpenShift Pipeline. Name this file `oc-pipelines-oc-final(.png)`

**Task 9:**

Screenshot showing details of the OpenShift Pipeline running successfully. Name this file `oc-pipelines-oc-green(.png)`.

**Task 10:**

Screenshot of the running application logs from OpenShift console. Name this file `oc-pipelines-app-logs`.

## Next steps

Make sure to save the output, as required by the review criteria, while following the step-by-step instructions.

Author(s)

Alima Akhter