

CI Live Class Assignment

Q1. Build an artifact and store the output of a Linux command in it > download the artifact locally.

Ans. Build an artifact

Objective: Set up a basic Artifact CI pipeline that builds a project automatically whenever new code is pushed.

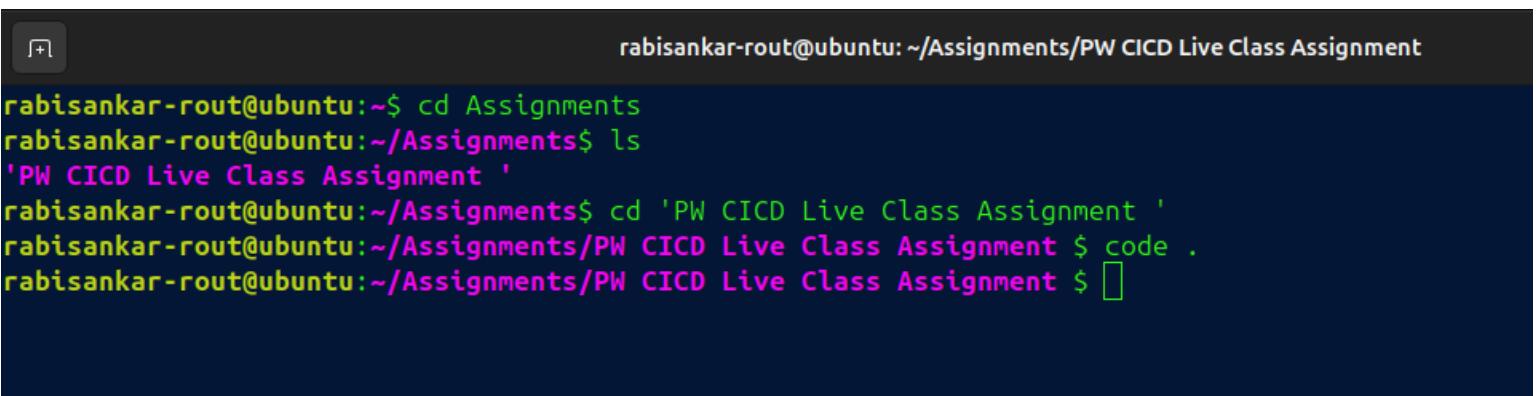
Tools: GitHub Actions

Goal:

- Run a Linux command
- Store the output into a file
- Upload the file as an artifact
- Optionally download it locally from the GitHub Actions UI

GitHub Actions Workflow Example:

Create a file: .github/workflows/build-Arti.yml



```
rabisankar-rout@ubuntu:~/Assignments/PW CICD Live Class Assignment
rabisankar-rout@ubuntu:~/Assignments$ ls
'PW CICD Live Class Assignment '
rabisankar-rout@ubuntu:~/Assignments$ cd 'PW CICD Live Class Assignment '
rabisankar-rout@ubuntu:~/Assignments/PW CICD Live Class Assignment $ code .
rabisankar-rout@ubuntu:~/Assignments/PW CICD Live Class Assignment $ 
```

Code:

```
name: Artifacts CI
on: [push]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout repository
        uses: actions/checkout@v3
      - name: Build project
        run: |
          mkdir -p output
          echo "Hello world!" >> output/artifact.txt
          ls -lh output/
          pwd >> output/artifact.txt
          cat output/artifact.txt
      - name: Upload artifact
```

```

uses: actions/upload-artifact@v4
with:
  name: My-artifact
  path: output/
download:
  needs: build
  runs-on: ubuntu-latest
steps:
  - name: Download Artifact
    uses: actions/download-artifact@v4
    with:
      name: My-artifact
      path: downloaded-artifact
  - name: Display content
    run: |
      ls -l downloaded-artifact/
      cat downloaded-artifact/artifact.txt

```

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer View:** Shows the file structure with `.github/workflows/Arti.yml` selected.
- Code Editor:** Displays the `Arti.yml` file content:

```

name: Artifacts CI
on: [push]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout repository
        uses: actions/checkout@v3
      - name: Build project
        run: |
          mkdir -p output
          echo "Hello world!" >> output/artifact.txt
          ls -lh output/
          pwd >> output/artifact.txt
          cat output/artifact.txt
      - name: Upload artifact
        uses: actions/upload-artifact@v4
        with:
          name: My-artifact
          path: output/

```
- Terminal:** Shows the command `git push origin main` being run, resulting in the output:

```

[main 3d5904b] Question-2
 1 file changed, 1 insertion(+), 1 deletion(-)
rabisankar-rout@ubuntu:~/Assignments/PW CICD Live Class Assignment $ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 365 bytes | 182.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Rabisankar-Rout/PWskills-CiCd-Live-Class-assignment
 47cc3a3..3d5904b main -> main
rabisankar-rout@ubuntu:~/Assignments/PW CICD Live Class Assignment $

```
- Ask Copilot Panel:** On the right, it says "Copilot is powered by AI, so mistakes are possible. Review output carefully before use." with instructions for using @, #, and /.

After pushing the code in Github Action, Result in the bellow.

2. Run jobs based on file changes in `src/**` to avoid unnecessary executions.

Goal:

- Only run a job if files in the `src/` directory have changed (to optimize workflows)

GitHub Actions Workflow Example:

Create a file: `.github/workflows/CI.yml`

Code:

```
name: CI Pipeline
```

```
on:
```

```
push:
```

```
  paths:
```

```
    - 'src/**'
```

```
pull_request:
```

```
  paths:
```

```
    - 'src/**'
```

```
jobs:
```

```
build:
```

```
  runs-on: ubuntu-latest
```

```
  steps:
```

```
    - name: Checkout repository
```

```
      uses: actions/checkout@v4
```

```
    - name: Run custom shell commands
```

```
      run: |
```

```

mkdir -p output
echo "Build started at $(date)" > output/build.log
echo "Listing src directory contents:" >> output/build.log
ls -la src >> output/build.log
echo "Build completed at $(date)" >> output/build.log

```

- name: Upload build log as artifact

uses: actions/upload-artifact@v4

with:

name: build-log

path: output/build.log

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure with files like Arti.yml, Cl.yml, Question-1.txt, and README.md.
- Terminal View:** Displays the GitHub Actions workflow code and its execution logs. The workflow code includes steps for building, committing, and pushing changes, as well as uploading artifacts.
- Status Bar:** Shows the current file is Arti.yml, the commit hash f3cfdbd..47cc3a3, and the date You, 7 minutes ago.
- Bottom Right Panel:** An "Ask Copilot" feature is visible, indicating AI assistance is available.

```

Wed Jul 23 8:28:17 PM

github > workflows > Arti.yml
You, 8 minutes ago | 2 authors (Karuna Issa and one other)
1 name: Artifacts CI
2 on: [push]
3 jobs:
4   build:
5     runs-on: ubuntu-latest
6     steps:
7       - name: Checkout repository
8         uses: actions/checkout@v3
9       - name: Build project
10      run:
11        mkdir -p output
12        echo "Hello world!" >> output/artifact.txt
13        ls -lh output/
14        pwd >> output/artifact.txt
15        cat output/artifact.txt
16   - name: Upload artifact
17     uses: actions/upload-artifact@v4
18     with:
19       name: My-artifact
20       path: output/

f3cfdbd..47cc3a3 main -> main
● rabisankar-rout@ubuntu:~/Assignments/PW CICD Live Class Assignment $ git add .
● rabisankar-rout@ubuntu:~/Assignments/PW CICD Live Class Assignment $ git commit -m "Question-2"
[main 3d5904b] Question-2
 1 file changed, 1 insertion(+), 1 deletion(-)
● rabisankar-rout@ubuntu:~/Assignments/PW CICD Live Class Assignment $ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 365 bytes | 182.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

```

After Pushing the code to Github Action, Result in the bellow.

The screenshot shows the GitHub Actions summary for a workflow run. The run was triggered via push 9 minutes ago by Rabisankar-Rout pushed to branch main. The status is Success, total duration is 9s, and there is 1 artifact. The CI.yml file has a single build step that completed successfully in 6s. The artifact build-log is a 295 Bytes file with digest sha256:a0e5eb32a8bd90323b4f55f7088ffdebf... A download icon is available for the artifact.

This is all about all Workflow files:

The screenshot shows the GitHub Actions page. The left sidebar is titled 'Actions' and includes sections for All workflows, Artifacts CI, CI Pipeline, Management, Caches, Attestations, Runners, Usage metrics, and Performance metrics. The main area displays 'All workflows' showing runs from all workflows. There are 2 workflow runs listed: 'Question-2' and 'Question-1'. Both runs were triggered by pushes to the main branch. The first run for 'Question-2' was triggered 8 minutes ago and completed 9s ago. The second run for 'Question-1' was triggered 11 minutes ago and completed 15s ago. A filter for workflow runs is present at the top right.

Link of this workflow file on My Github Account: <https://github.com/Rabisankar-Rout/PWskills-CiCd-Live-Class-assignment>

Assignment submitted by Rabisankar Rout

Devops & Cloud Computing Course