

Setting Up Cloud-based CI/CD Pipeline

This experiment sets up a Continuous Integration/Continuous Deployment (CI/CD) pipeline using GitHub Actions, Docker, and LocalStack to simulate AWS services. It provides hands-on experience in automating deployments with AWS CLI and S3, demonstrating cloud-based automation workflows.

This document provides a comprehensive breakdown of all commands, inputs, outputs, and their explanations, ensuring a clear understanding of each step in the workflow.

1. Creating an S3 Bucket

Command:

```
aws --endpoint-url=http://localhost:4566 s3 mb s3://my-ci-cd-artifacts
```

Explanation:

- `aws s3 mb` → Creates a new S3 bucket.
- `s3://my-ci-cd-artifacts` → The name of the bucket being created.
- `--endpoint-url=http://localhost:4566` → Uses **LocalStack** to simulate AWS services.

Output:

```
make_bucket: my-ci-cd-artifacts
```

2. Attempting to Create a CodeCommit Repository

Command:

```
aws --endpoint-url=http://localhost:4566 codecommit create-repository --repository-name my-repo
```

Explanation:

- `aws codecommit create-repository` → Creates a new AWS CodeCommit repository.
- `--repository-name my-repo` → Assigns the repository name as `my-repo`.

- `--endpoint-url=http://localhost:4566` → Uses **LocalStack**.

Error Output:

An error occurred (InternalFailure) when calling the CreateRepository operation: API **for** service '**codecommit**' not yet implemented or pro feature - please check <https://docs.localstack.cloud/references/coverage/> **for** further information

3. Initializing a Git Repository

Command:

```
git init
```

Explanation:

- `git init` → Initializes a new **Git repository** in the current directory.

Output:

```
Initialized empty Git repository in C:/Users/rawat/Documents/8  
SEMESTER/Cloud Computing/Lab/Experiment 10/Codes/.git/
```

4. Staging and Committing Files

Commands:

```
git add .  
git commit -m "Initial commit"
```

Explanation:

- `git add .` → Stages all files for commit.
- `git commit -m "Initial commit"` → Commits the staged files with a message.

Output:

```
[master (root-commit) 2dfb5b6] Initial commit  
3 files changed, 1153 insertions(+)
```

```
create mode 100644 Command Prompt Input and Output Explanation.md
create mode 100644 Command Prompt Input and Output Explanation.pdf
create mode 100644 Command Prompt Input and Output.txt
```

5. Uploading a ZIP File to S3

Command:

```
aws --endpoint-url=http://localhost:4566 s3 cp my-code.zip
s3://my-ci-cd-artifacts/
```

Explanation:

- `aws s3 cp` → Copies a file to S3.
- `my-code.zip` → The file being uploaded.
- `s3://my-ci-cd-artifacts/` → Destination bucket in S3.
- `--endpoint-url=http://localhost:4566` → Uses **LocalStack**.

Error Output:

```
The user-provided path my-code.zip does not exist.
```

6. Creating a ZIP Archive

Command:

```
powershell Compress-Archive -Path * -DestinationPath my-code.zip
```

Explanation:

- `Compress-Archive -Path * -DestinationPath my-code.zip` → Creates a ZIP archive of all files in the directory.

7. Uploading the ZIP File Again

Command:

```
aws --endpoint-url=http://localhost:4566 s3 cp my-code.zip  
s3://my-ci-cd-artifacts/
```

Output:

```
upload: .\my-code.zip to s3://my-ci-cd-artifacts/my-code.zip
```

8. Listing the Uploaded Files in S3

Command:

```
aws --endpoint-url=http://localhost:4566 s3 ls s3://my-ci-cd-artifacts/
```

Output:

```
2025-03-08 10:32:42      289415 my-code.zip
```

9. Setting Up a Remote Git Repository

Commands:

```
git remote add origin https://github.com/madhurimarawat/Cloud-  
Computing.git  
git branch -M main  
git push -u origin main
```

Explanation:

- `git remote add origin <repo-url>` → Links the local repository to GitHub.
- `git branch -M main` → Renames the current branch to `main`.
- `git push -u origin main` → Pushes the code to GitHub.

Error Output:

```
To https://github.com/madhurimarawat/Cloud-Computing.git  
! [rejected]        main -> main (fetch first)  
error: failed to push some refs to  
'https://github.com/madhurimarawat/Cloud-Computing.git'  
hint: Updates were rejected because the remote contains
```

work that you **do** not
hint: have locally. This is usually caused by another
repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: '**git pull**' before pushing again.

Fix:

To resolve this issue, run:

```
git pull origin main --rebase
git push -u origin main
```

10. Pulling the Latest Changes from GitHub

Command:

```
git pull origin main --rebase
```

Explanation:

- Fetches changes from the **remote repository** and applies them using **rebase** instead of a merge.
- Ensures a linear commit history by reapplying local changes on top of the latest remote changes.

Output:

```
remote: Enumerating objects: 240, done.
remote: Counting objects: 100% (240/240), done.
remote: Compressing objects: 100% (212/212), done.
remote: Total 240 (delta 100), reused 43 (delta 21), pack-reused 0
Receiving objects: 100% (240/240), 9.22 MiB | 1.11 MiB/s, done.
Resolving deltas: 100% (100/100), done.
From https://github.com/madhurimarawat/Cloud-Computing
* branch          main          -> FETCH_HEAD
* [new branch]     main          -> origin/main
Successfully rebased and updated refs/heads/main.
```

11. Staging All Changes

Command:

```
git add .
```

Explanation:

- Stages all modified and newly created files in the **current directory** for the next commit.

12. Checking for an Ongoing Rebase

Command:

```
git rebase --continue
```

Explanation:

- Used to **continue** an ongoing rebase operation if there are conflicts.
- In this case, the **error** means there was **no ongoing rebase**, so this step was unnecessary.

Output:

```
fatal: no rebase in progress
```

13. Pushing Changes to GitHub

Command:

```
git push -u origin main
```

Explanation:

- Pushes local changes to the **remote repository** (`origin`), setting `main` as the **upstream branch**.
- This makes future `git push` commands **simpler** by automatically pushing to `origin main`.

Output:

```
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 282.43 KiB | 31.38 MiB/s, done.
Total 5 (delta 1), reused 1 (delta 0), pack-reused 0
```

```
remote: Resolving deltas: 100% (1/1), completed with 1 local object.  
To https://github.com/madhurimarawat/Cloud-Computing.git  
    b201b02..eb4faf7  main -> main  
branch 'main' set up to track 'origin/main'.
```

Output Breakdown:

- **Delta compression** → Reduces the size of transmitted data.
- **Objects written successfully** → Confirms the **push** was successful.
- **Tracking branch set up** → Future `git push` commands will default to `origin main`.