DevOps
Code Repo and Code Build
(From S3 and Github)

Muhammad Hamza 21K-4579

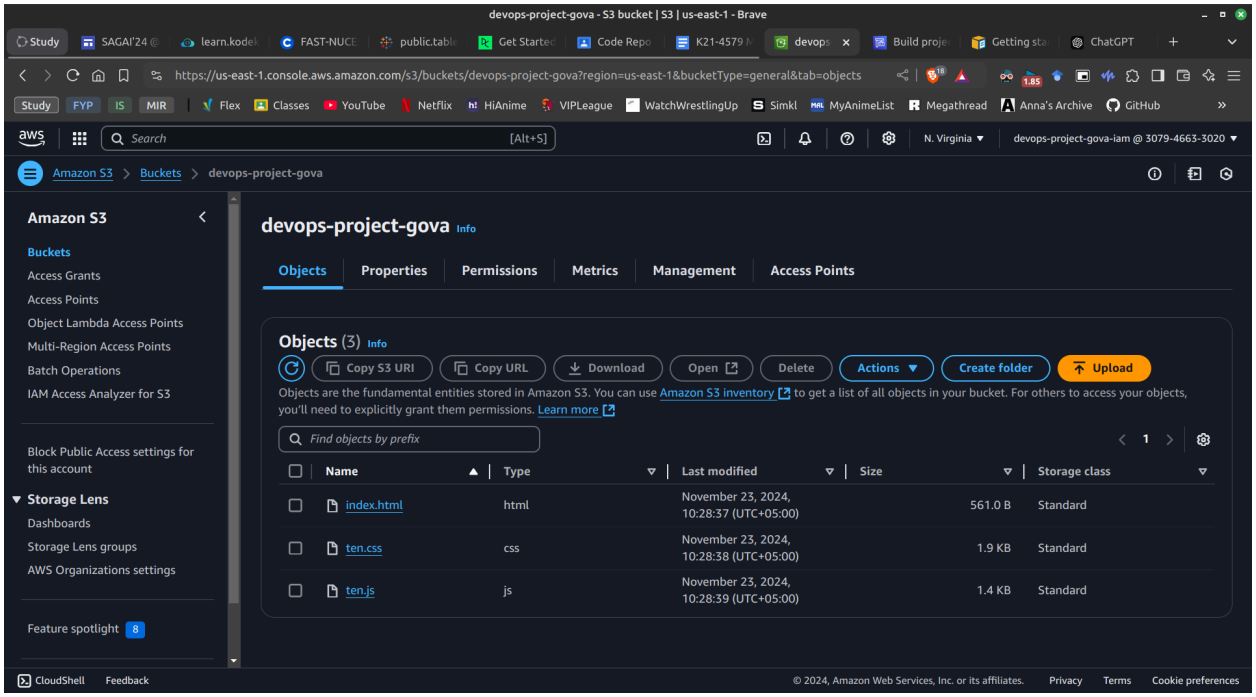
Course Lecturer:
→ Syed Sohaib Ur Rehman (BCS-7B)

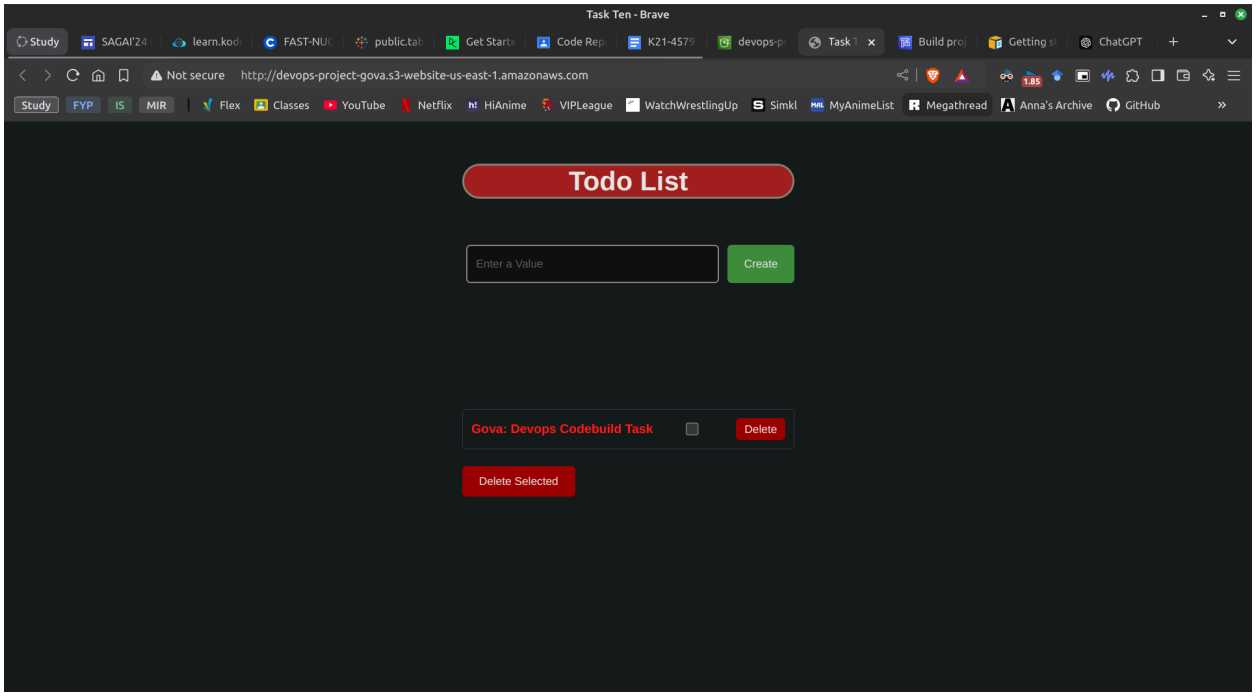
Department of Computer Science
School of Computing
FAST National University of Computer and Emerging Sciences, Karachi

Static Website on S3

Step 1: Host Static Website on S3

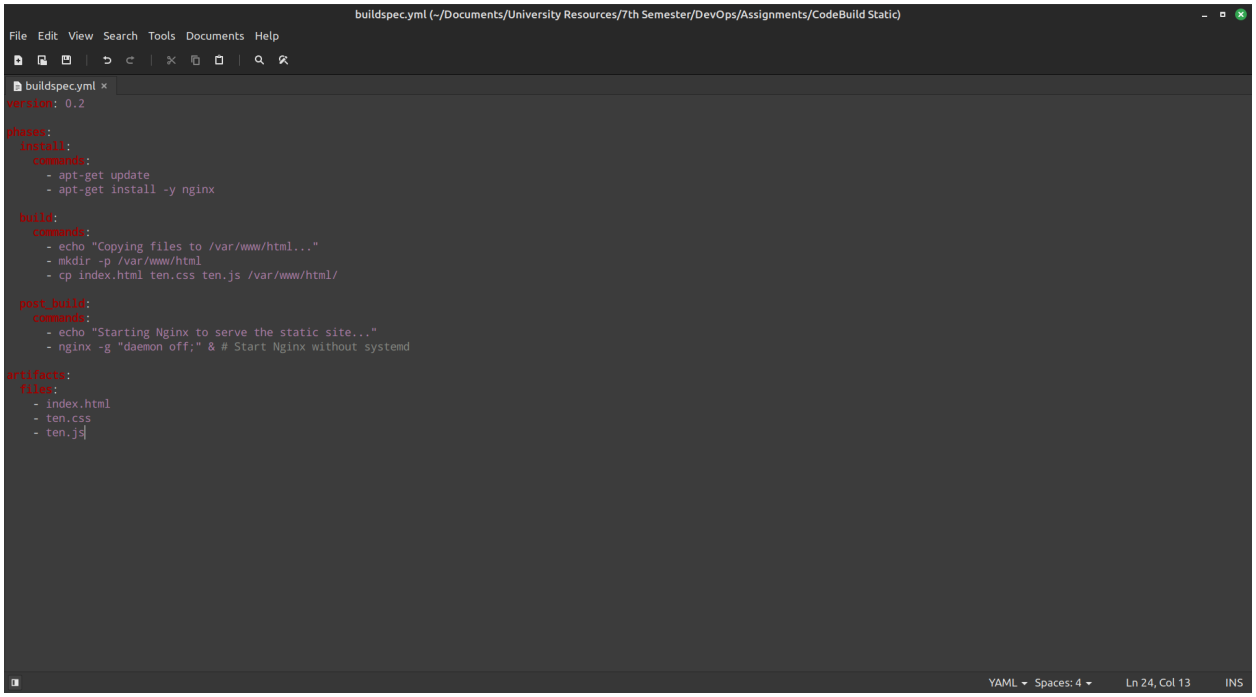
1. Create an S3 Bucket:
    - Open the AWS Management Console and navigate to the S3 service.
    - Create a new S3 bucket (devops-project-gova) with a globally unique name.
    - Enable public access by adjusting bucket permissions.
2. Upload Static Website Files:
    - Upload the provided files (index.html, ten.css, ten.js) to the bucket.
    - Set the proper permissions to make these files publicly accessible.
3. Configure the Bucket for Static Hosting:
    - Go to the bucket's Properties tab.
    - Enable Static Website Hosting.
    - Specify the index.html file as the entry point.



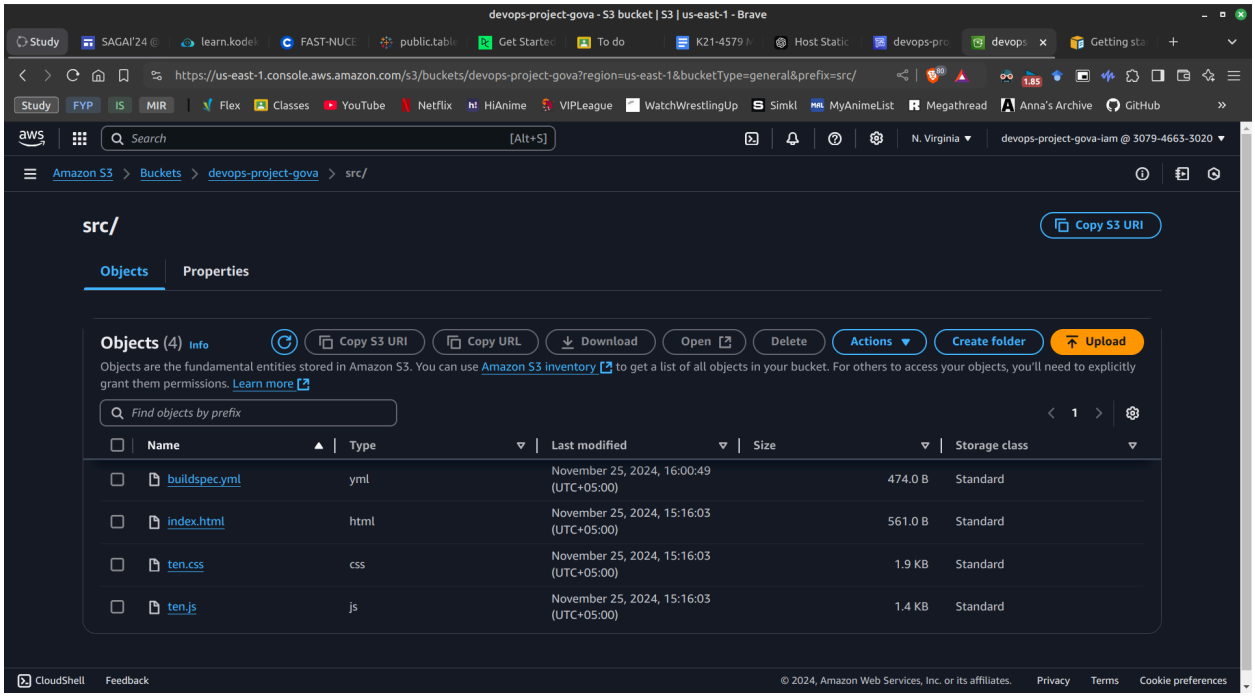


Step 2: Add Buildspec.yml File to S3

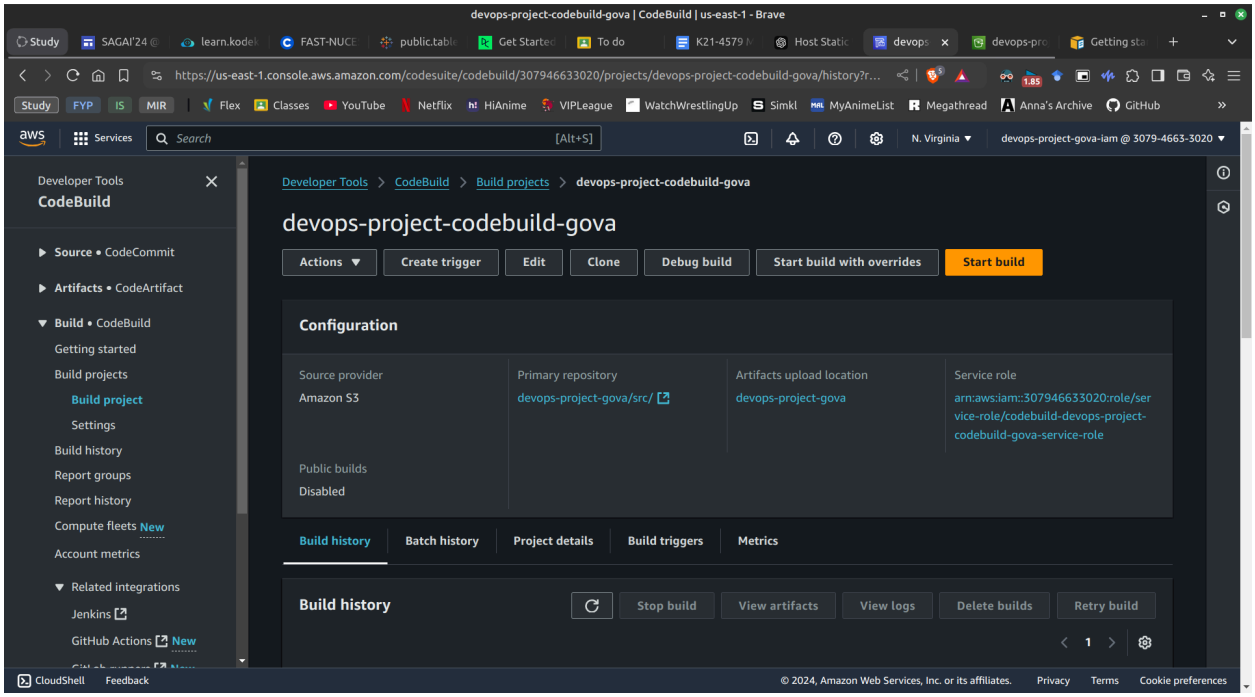
1. Prepare the buildspec.yml File:



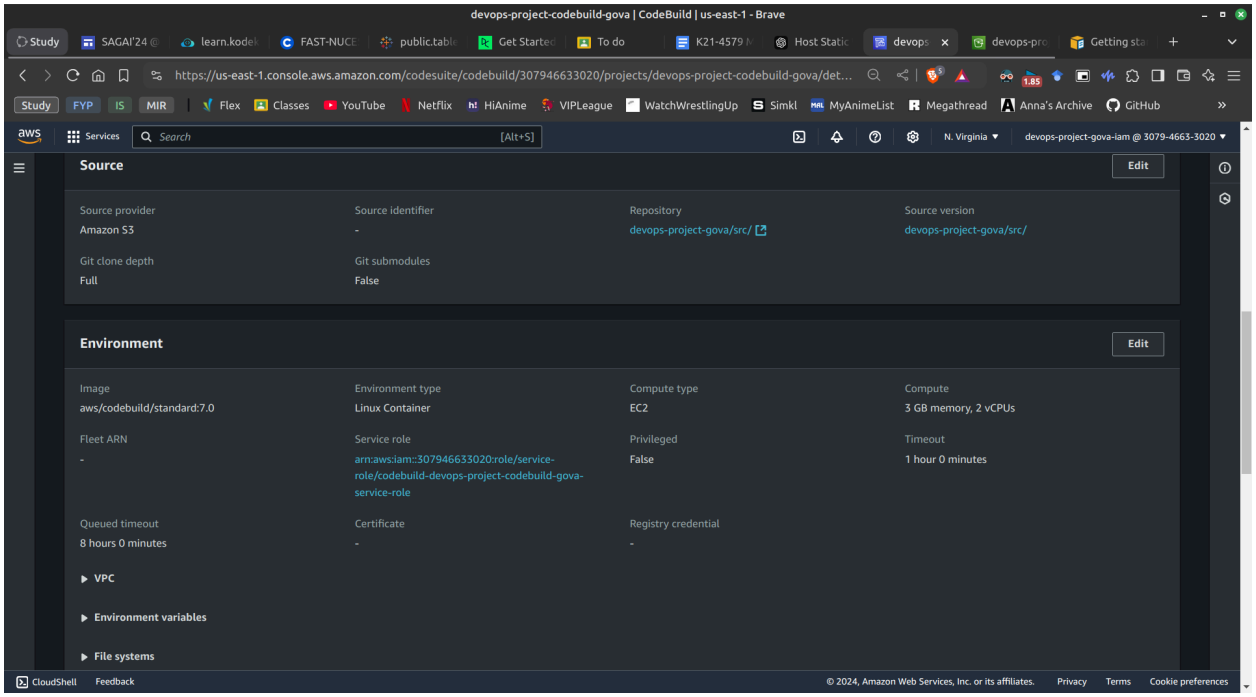
2. Upload buildspec.yml to the S3 Bucket:
    - Upload the buildspec.yml file to the same bucket (devops-project-gova).



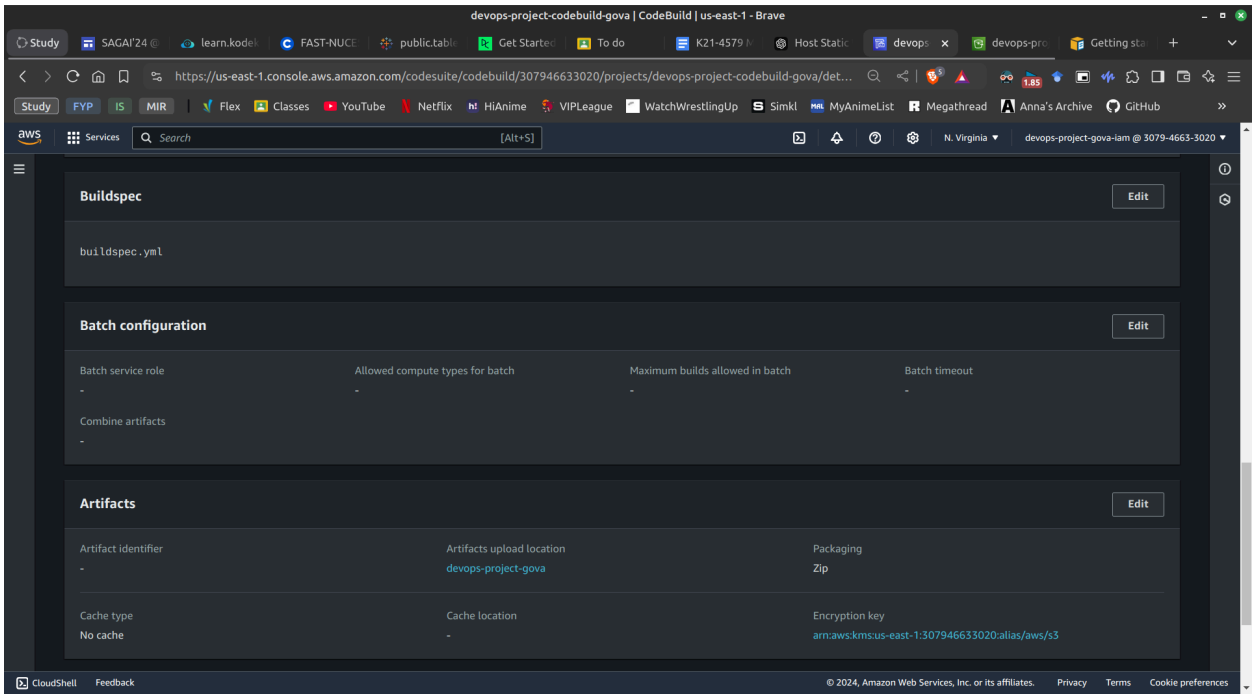
Step 3: Set Up AWS CodeBuild



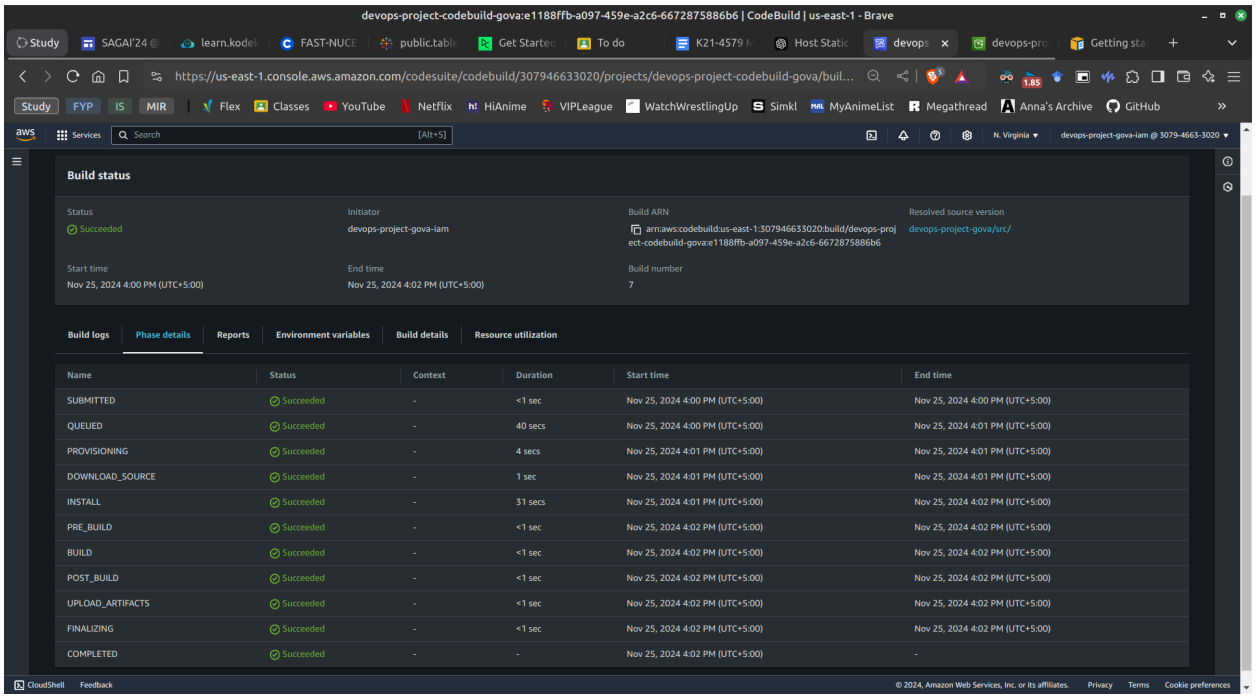
1. Create a CodeBuild Project:
    - Navigate to the AWS CodeBuild console and create a new project.
    - Specify the S3 bucket as the source and select the buildspec.yml file.
    - Configure the build environment:
        - Use a managed image (e.g., Ubuntu).
        - Select runtime (e.g., standard).



2. Define Output Artifact:
    - In the CodeBuild configuration, set the S3 bucket (devops-project-gova) as the destination for output artefacts.

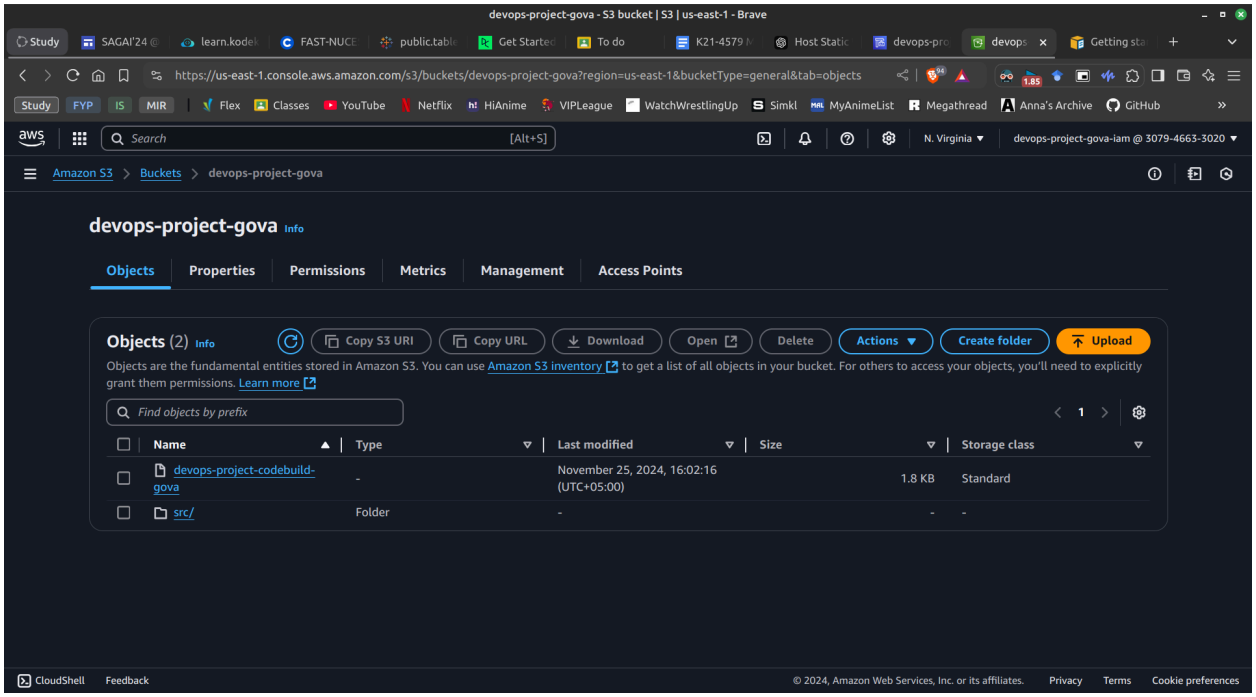


3. Run the Build:
    - Start the build process. CodeBuild will:
        - Run the buildspec.yml commands.
        - Validate and package the static website files.
        - Save the build output to the S3 bucket.



Verify

1. Check Build Artifacts:
    - Navigate to the S3 bucket and confirm that the build/ folder contains the output files.
2. Test the Static Website:
    - Access the static website via the S3 endpoint (or use a custom domain if configured).
3. Validate Logs:
    - Review CodeBuild logs to confirm that all phases (install, test, build, post_build) executed successfully.

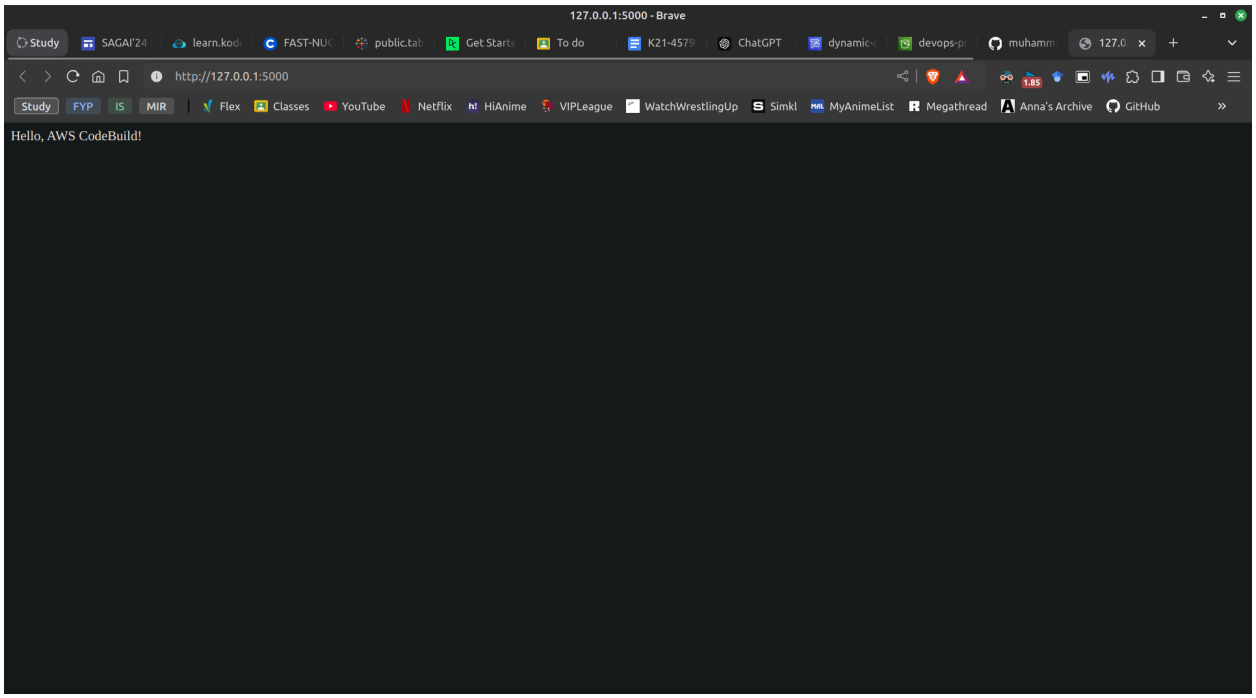


Dynamic Website on Github

Step 1: Prepare Your Dynamic Code

1. Choose Your Dynamic Code:
    - For seamless integration with AWS CodeBuild, opt for a dynamic programming language that's well-supported by AWS tools.
        - Python (with requirements.txt for dependencies)

    A Python Flask web app:



    Add a requirements.txt file for dependencies:

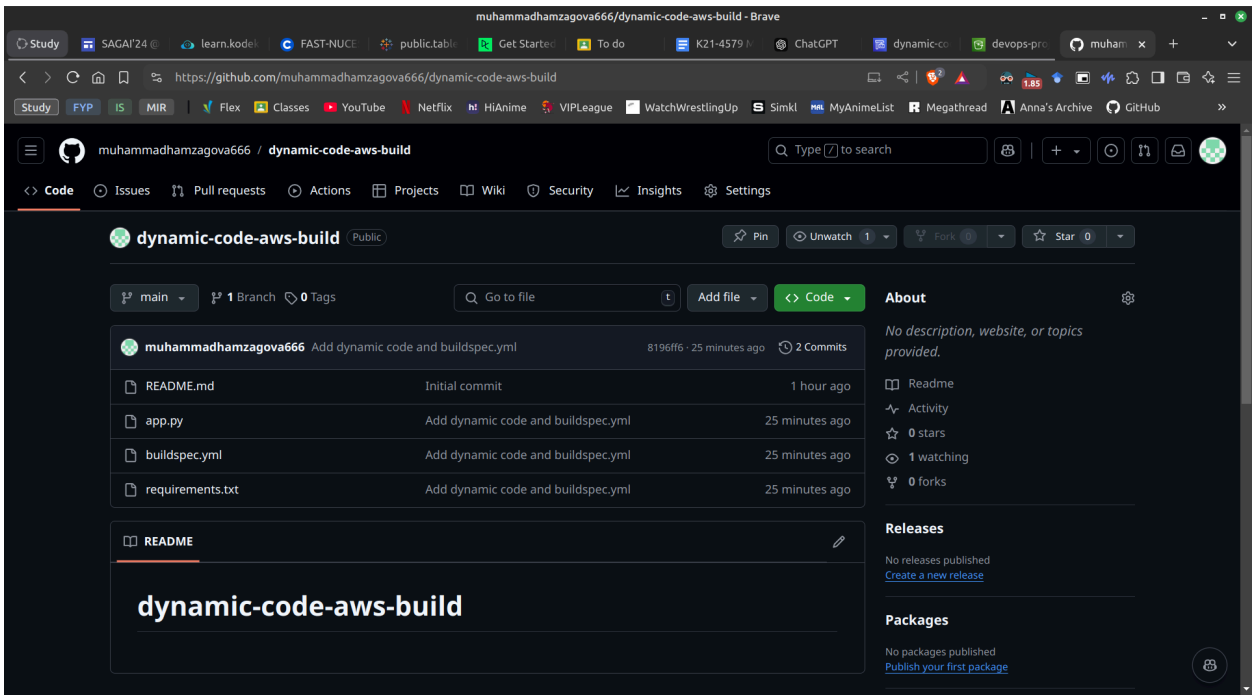
    - flask==2.3.2

Step 2: Create a GitHub Repository

1. Log in to your GitHub account.
2. Click New Repository and name your repository (e.g., dynamic-code-aws-build).
3. Initialize it with a README.md (optional).

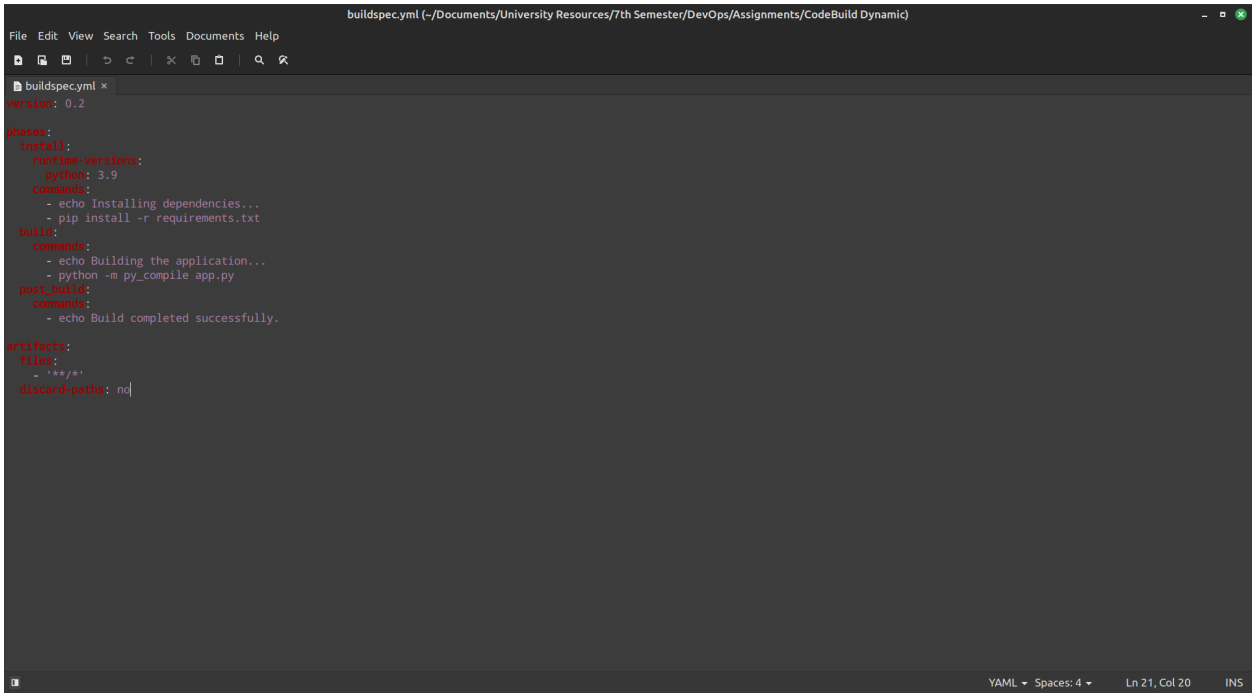
Clone the repository to your local machine
git clone
https://github.com/muhammadhamzagova666/dynamic-code-aws-build.git

4. Add your dynamic code files (app.py, requirements.txt) to the repository folder.



Step 3: Create a Buildspec File

The buildspec.yml file defines the build commands and phases for AWS CodeBuild.



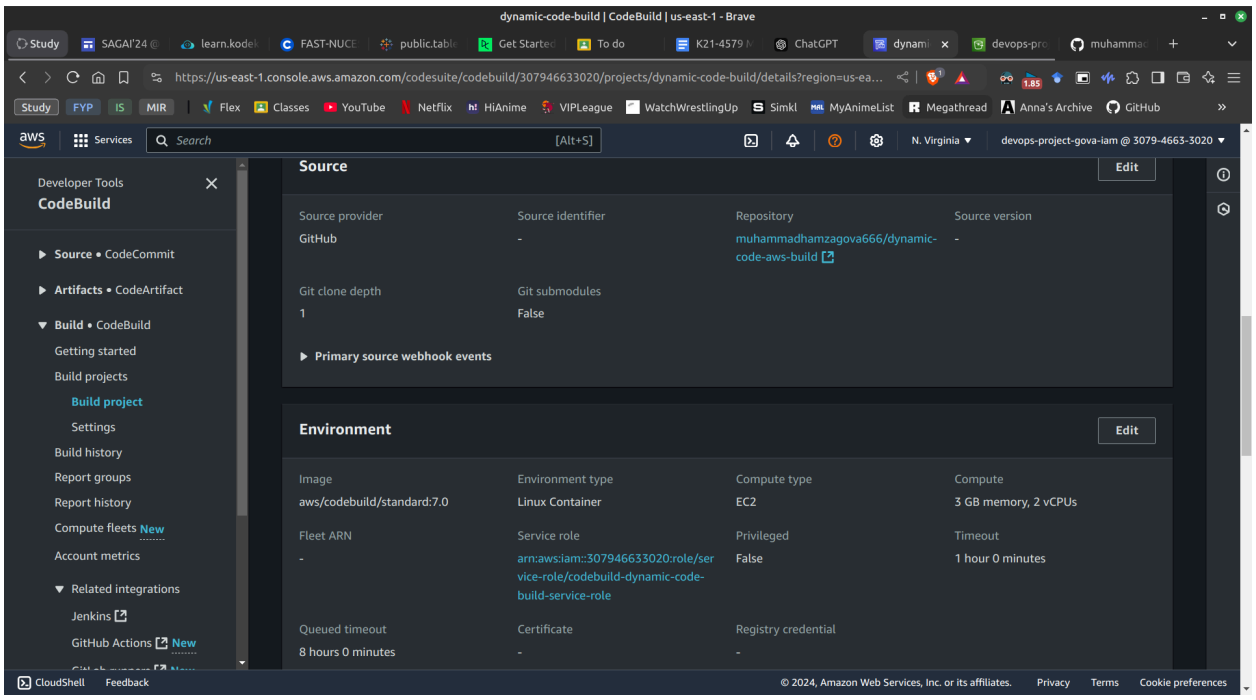
Step 4: Push Your Code and Buildspec to GitHub

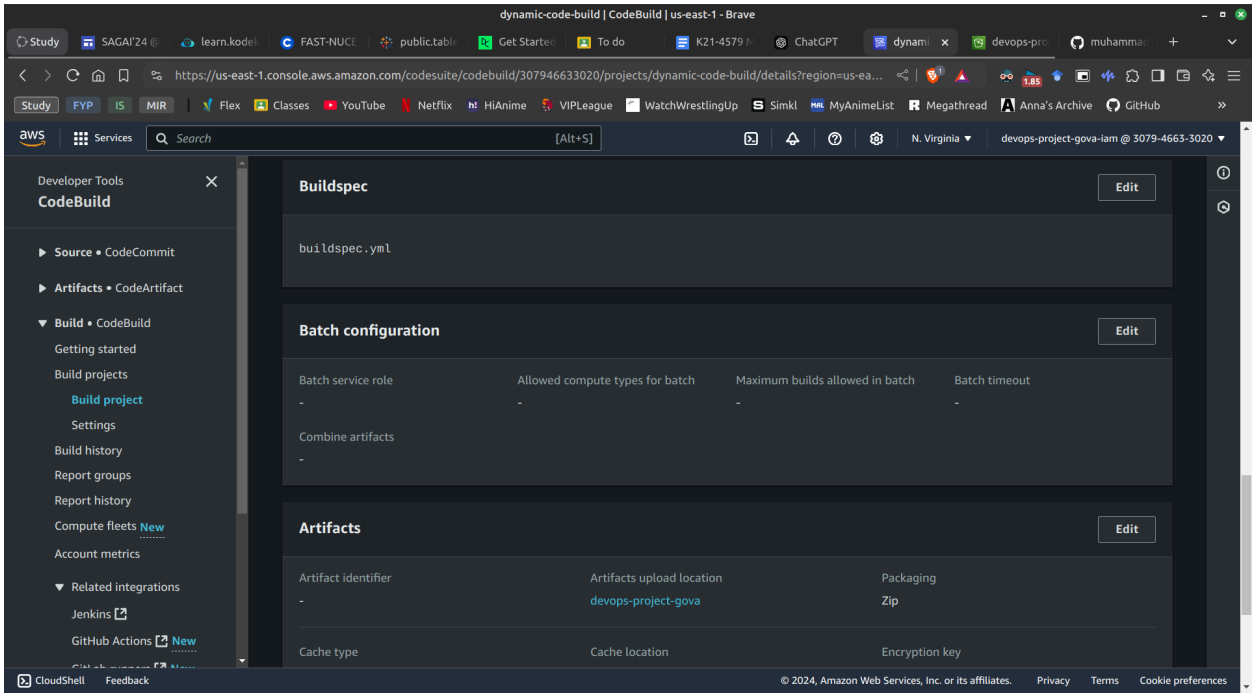
1. Add the files to the repository
    - git add .
2. Commit the changes:
    - git commit -m "Add dynamic code and buildspec.yml"
3. Push to GitHub
    - git push origin main

Step 5: Set Up AWS CodeBuild

1. Create a CodeBuild Project:
    - Go to the AWS CodeBuild Console.
    - Click Create Build Project.
    - Fill in the project details:
        - Project Name: dynamic-code-build
        - Source: Choose GitHub and connect your repository.
        - Environment: Use a managed image (Ubuntu).
        - Buildspec: Use the buildspec.yml file from your repository.
    - Set Artifacts to "Amazon S3":

- Specify the S3 bucket and path where the build artefact should be saved.





Step 6: Run the Build

1. In the CodeBuild console, select your project and click Start Build.
2. Monitor the logs to ensure the build completes successfully.
3. Check your S3 bucket for the output artefacts.