**Assignment 2**

Assignment:

* Deploy a static website along with domain accessible from internet

Todo:

* Deploying static assets on AWS
* Create a domain which is publicly accessible

Tech Stack:

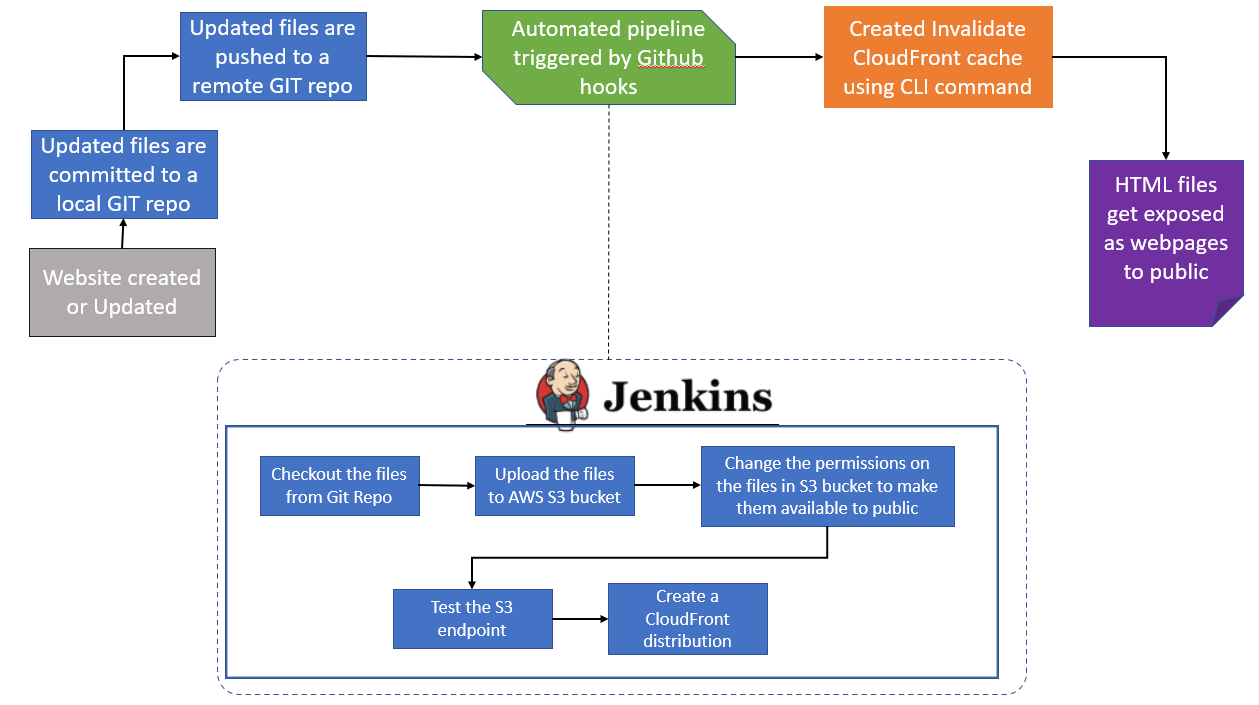
* Cloud Formation, S3
* Route 53 for creating public domain
* Static assets (could be as simple as a hello world index page)
* Jenkins groovy/ other ci/cd(gitlab or github actions) tool for deploying static assets
* If github actions/gitlab ci/cd is used, install git
* AWS CLI

🡪

## Pre-Requisites –

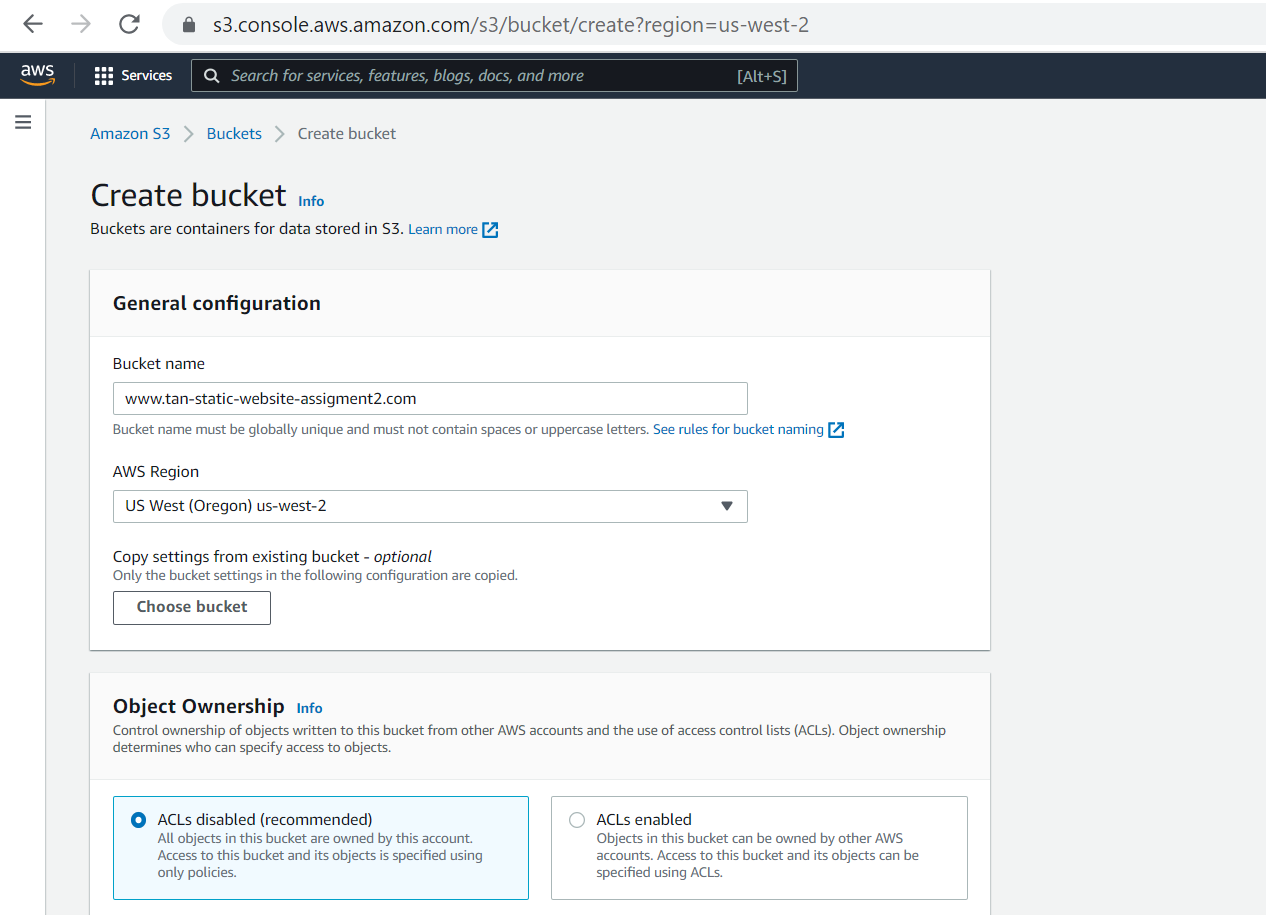
* AWS Account
* Jenkins installed on an EC2 Instance
* GitHub account

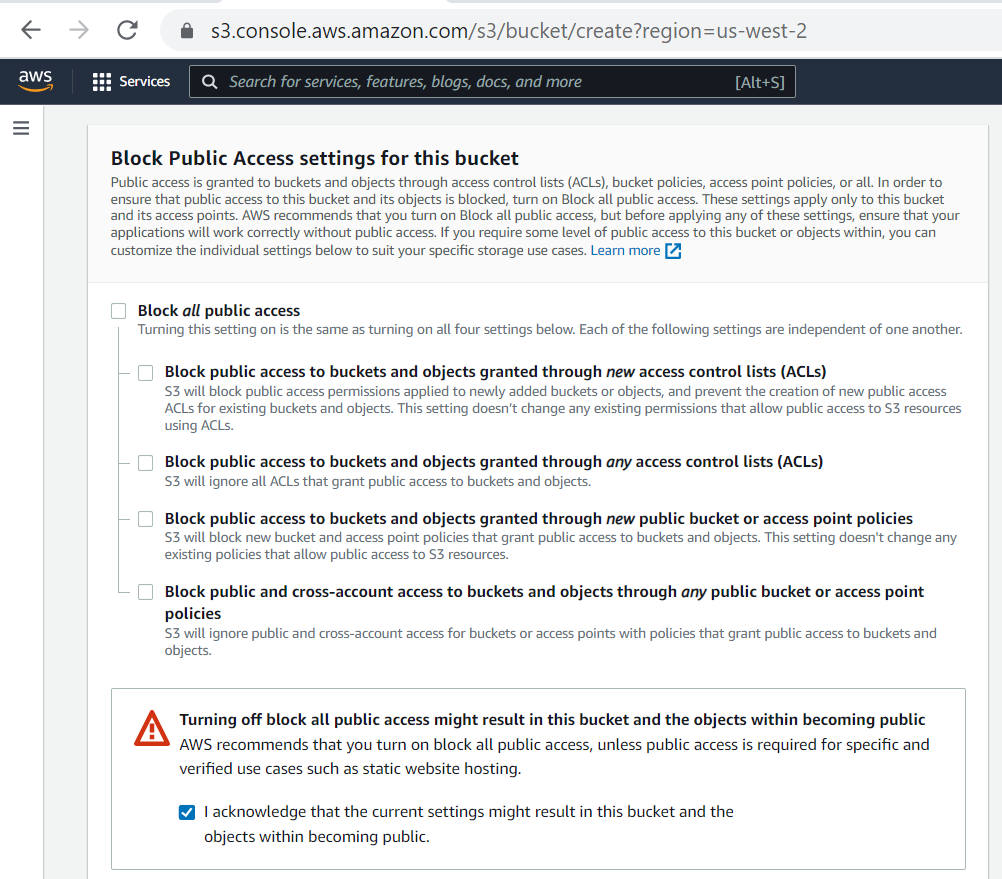
## Solution Architect –

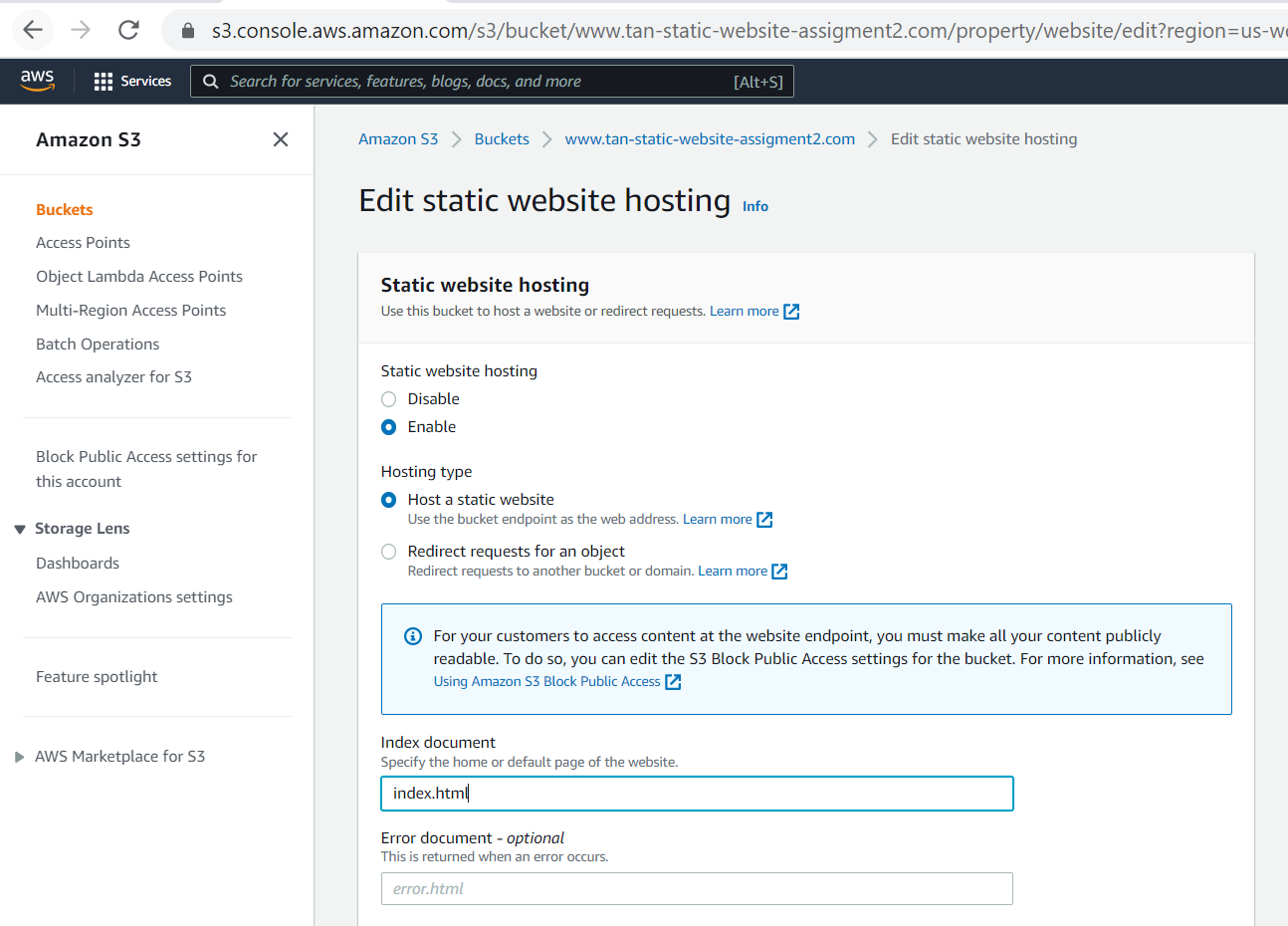


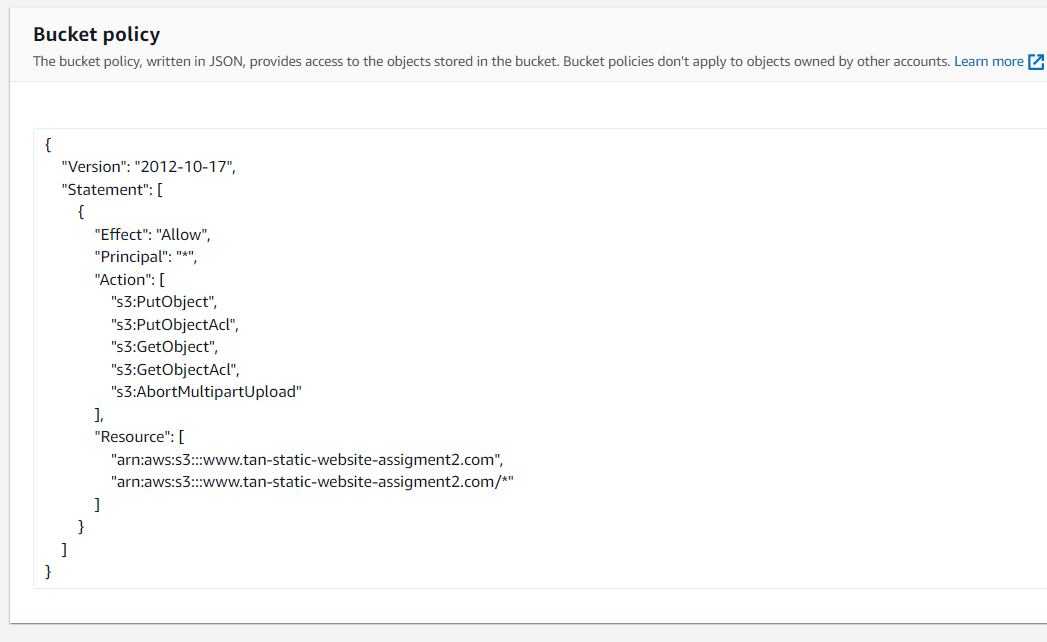
## Implementation –

1. Setting up S3 Bucket for Static Website –
   1. www S3 bucket :

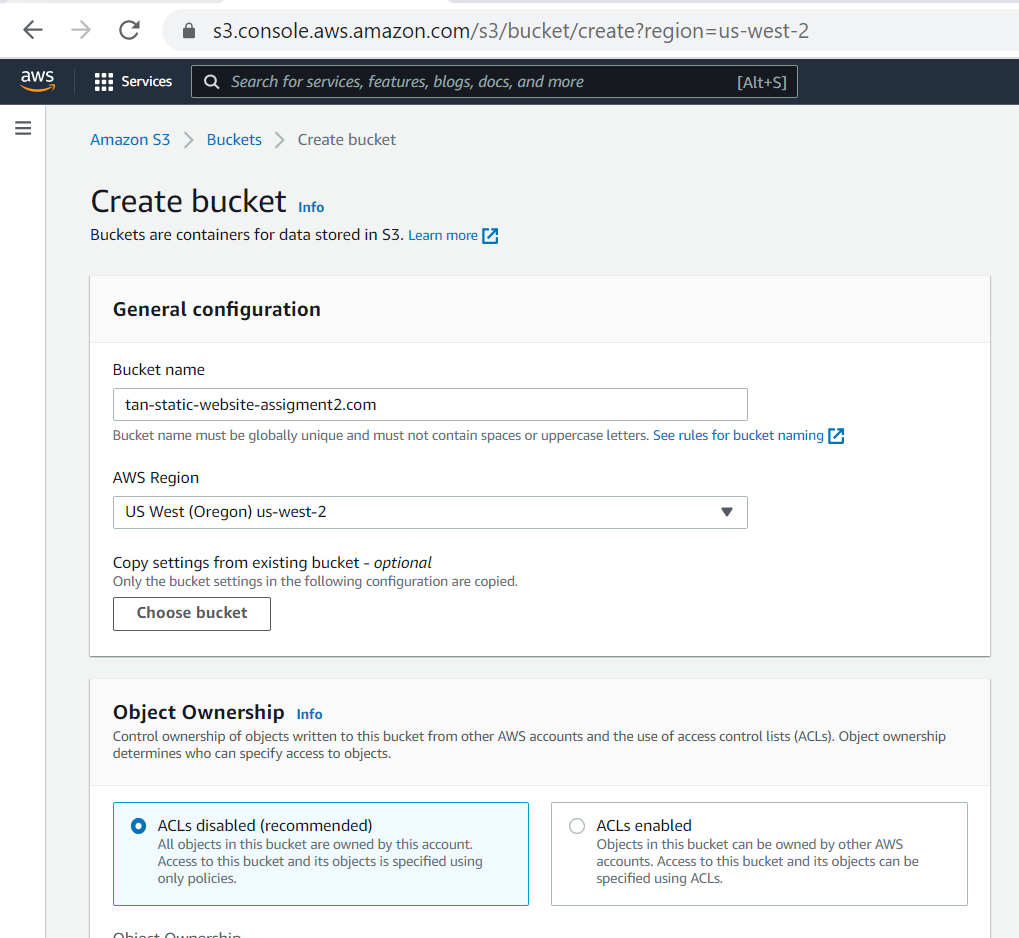


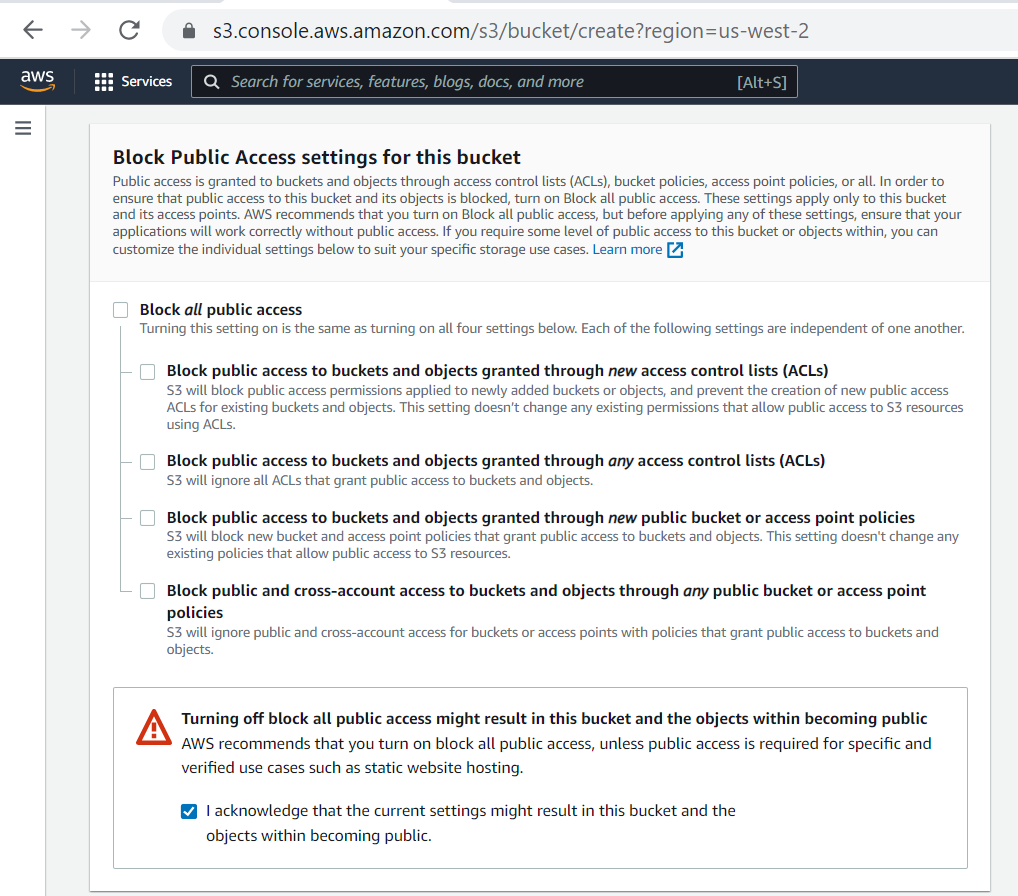


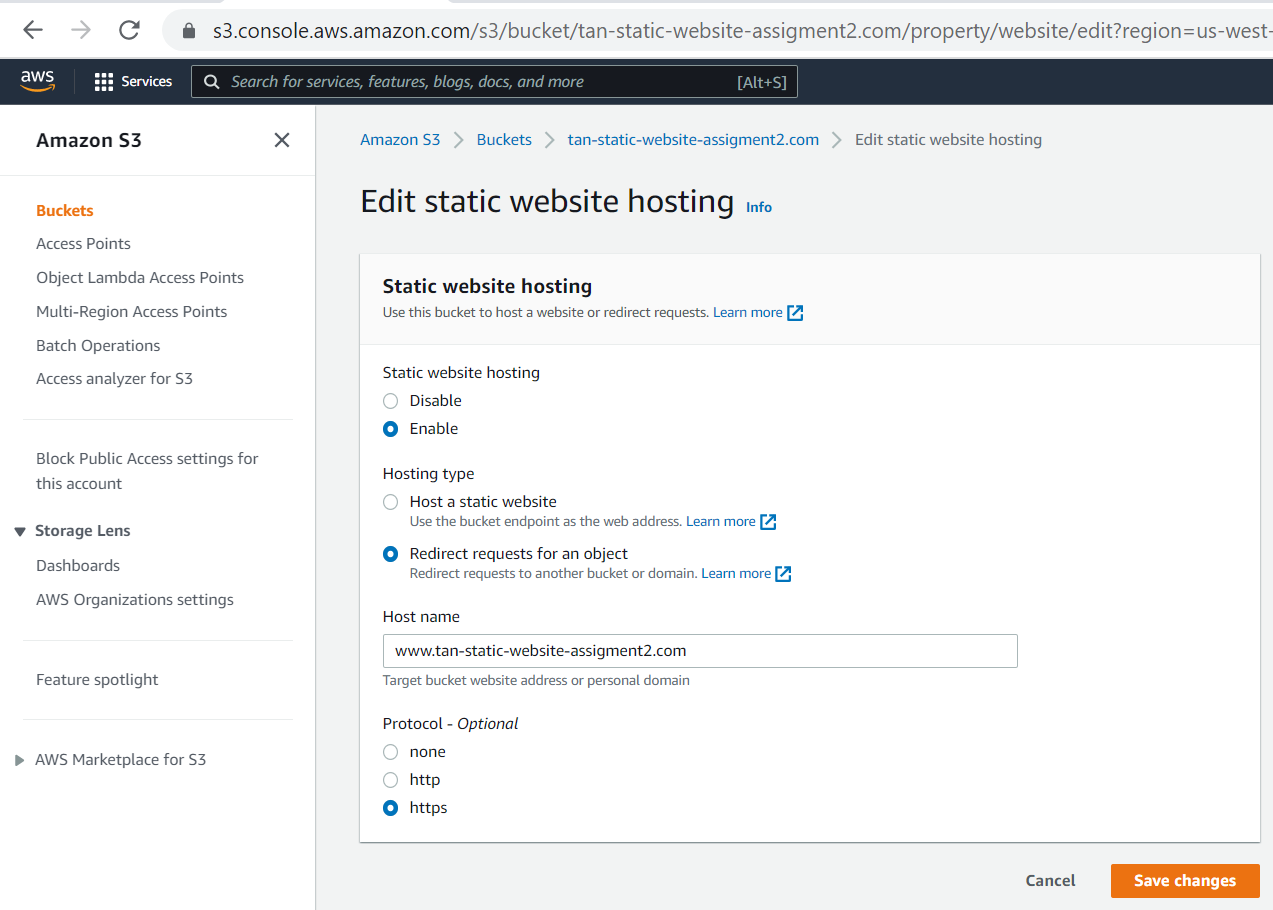




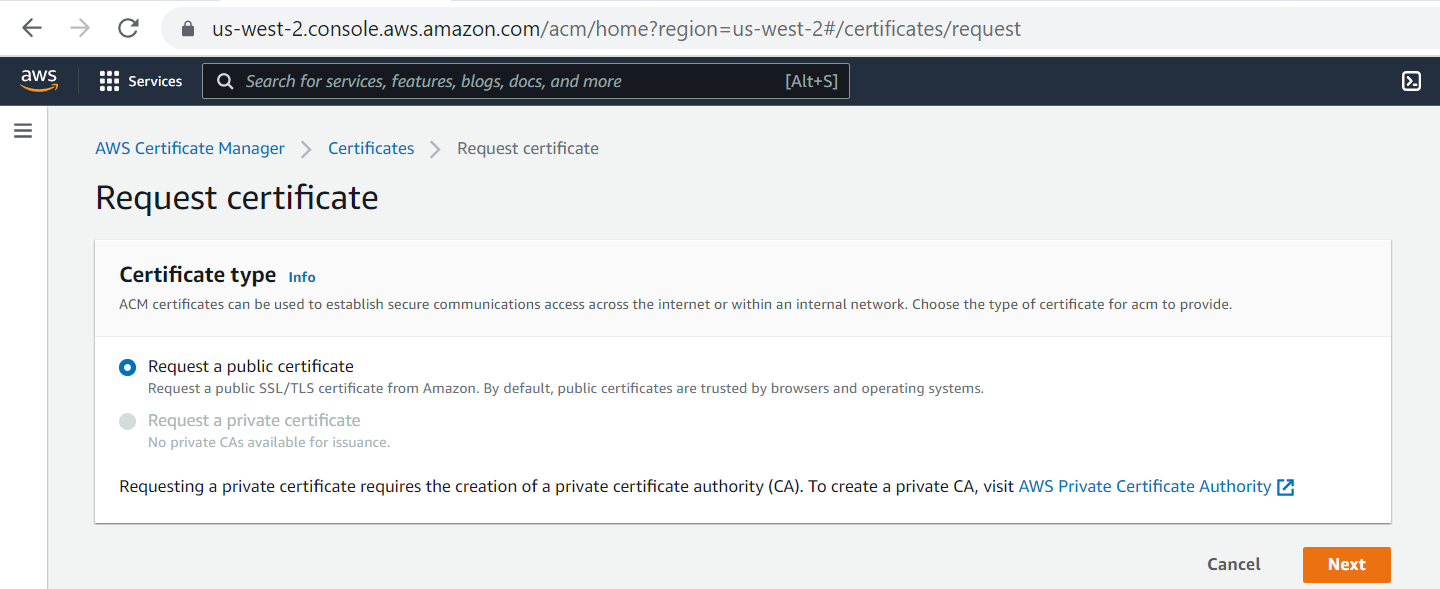
* 1. non-www S3 bucket :

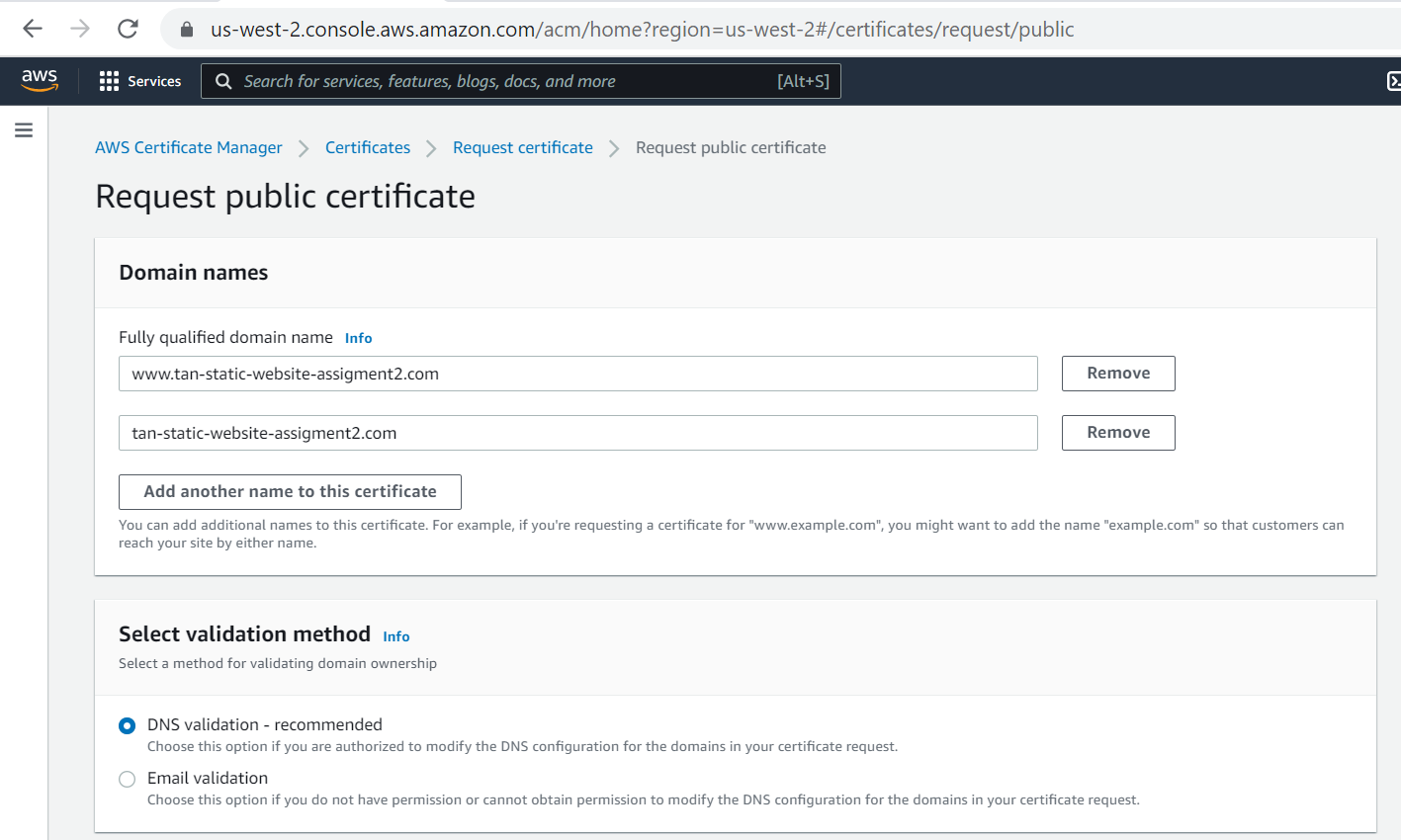


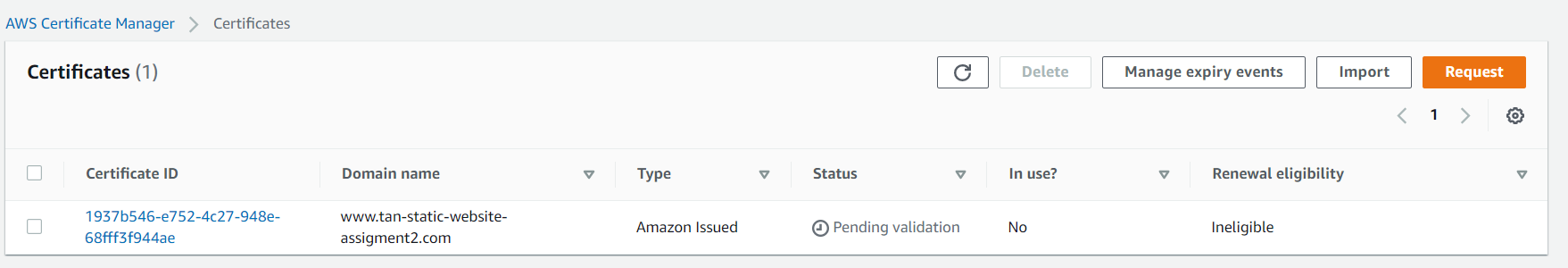




1. SSL certificate in AWS Certificate Manager –

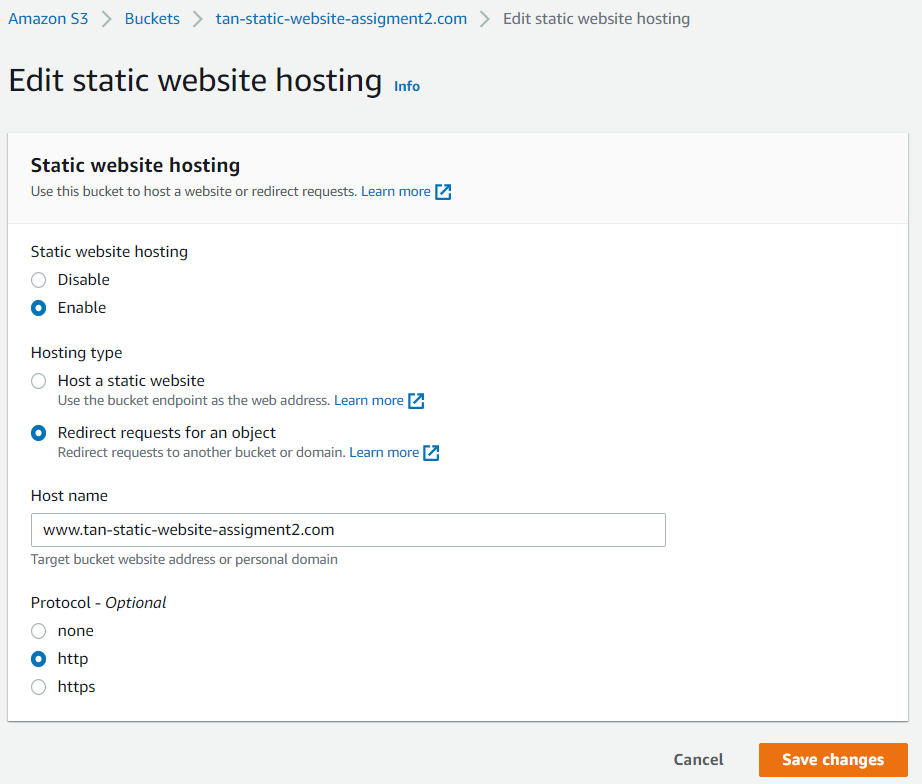






1. Route 53 as our DNS provider which have cost related to it so for now skipping it.

Reverting https to http –

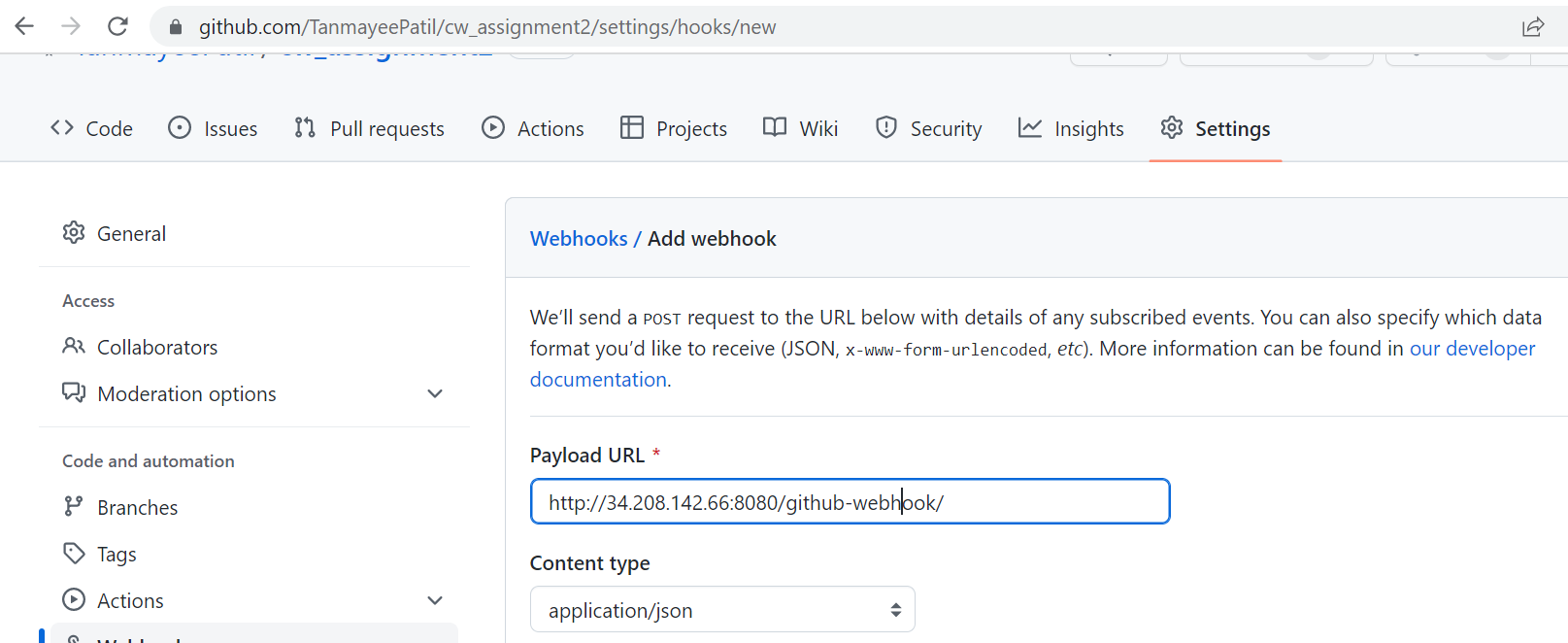


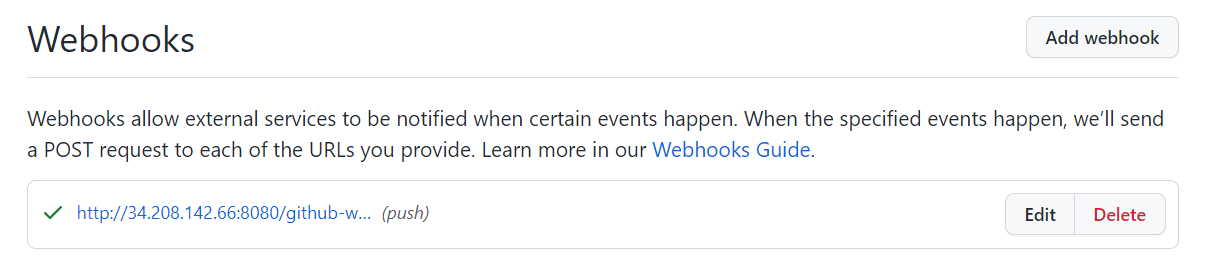
1. Created a CloudFront distribution using Jenkins (Please refer Jenkins Section below)
   1. My IAM account has AdministratorAccess so no need to add anything. Otherwise need to add CloudFrontFullAccess to create a CloudFront distribution

## Automated Pipeline Approaches

About Pipeline –

1. Added Webhook settings so Github can trigger the pipeline –





1. Set global variable or take it as input (here I have set them as env) –

To be able to upload to S3, you need to save your credentials in environment variables on your Jenkins:

Jenkins – Manage Jenkins – Configure System – Global properties – Environment variables.

AWS\_DEFAULT\_REGION=<region of bucket>

AWS\_ACCESS\_KEY\_ID=<aws id>

AWS\_SECRET\_ACCESS\_KEY=<your secret access key>

1. Executed Jenkins Pipeline –

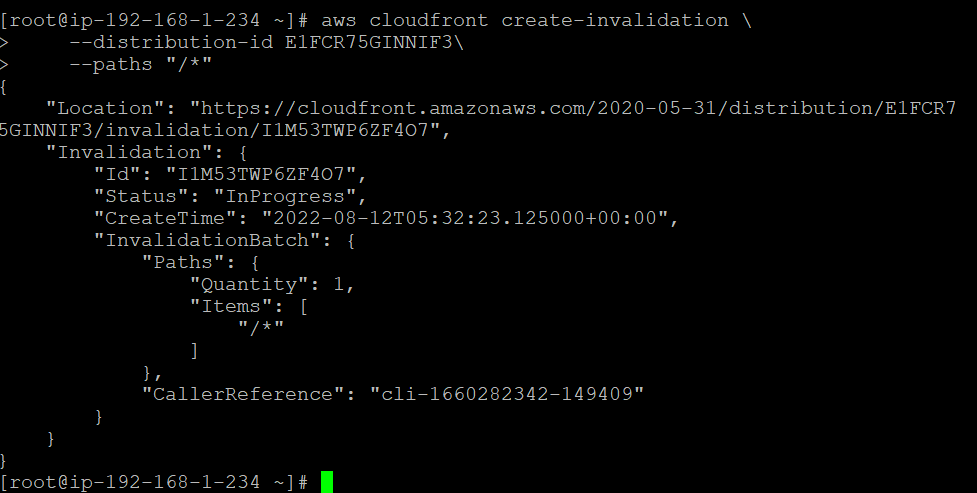


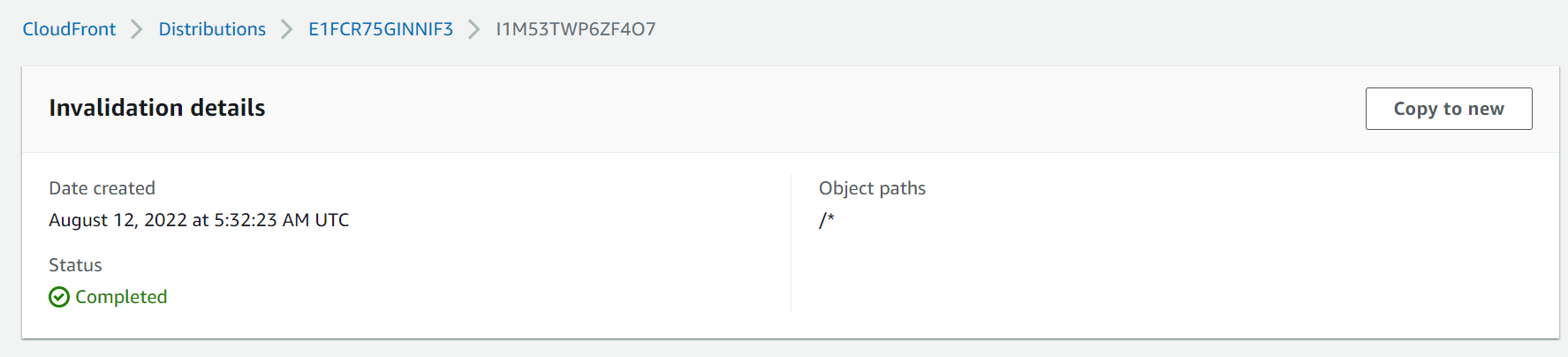
1. Once the CloudFront distribution created successfully
2. Created Invalidate CloudFront cache using CLI command –

aws cloudfront create-invalidation \

--distribution-id E1FCR75GINNIF3\

--paths "/\*"





**GREAT SUCCESS 😊**

**Distribution domain name :**

<https://d2z5q4gq6owmnc.cloudfront.net>



**S3 URL:**

<https://s3.us-west-2.amazonaws.com/www.tan-static-website-assigment2.com/index.html>

