Mukund parameswaran

Per Scholas

01/31/2021

Cloud Infrastructure provision & Monitor

**TABLE OF CONTENTS**

1.[INTRODUCTION](#_Toc275164849) 3

2.DEVOPS [TOOLS & CLOUD SERVICES USED](#_Toc275164850) 3

3.[DESIGN](#_Toc275164851) 4

4.[SETUP PHASE](#_Toc275164853) 5

5.[IAM ROLE CREATION](#_Toc275164853) 7

6.[BUILD 1](#_Toc275164853)1

7.LINTING [1](#_Toc275164853)7

8.[MONITORING 1](#_Toc275164860)9

9.[CHALLENGES](#_Toc275164872) 21

10.[IMPROVISATION](#_Toc275164872) 22

1.INTRODUCTION

This Case Study is my second and final declarative pipeline building a DevOps flow. I am building a CI/CD pipeline with the concepts, tools, and technologies learned during this training. The main focus of this project is to continuously make these small changes to the code, building, testing, provisioning cloud infrastructure and monitoring.

2.DevOps TOOLS & Cloud Services USED

**GIT**:

Source code Management tool.

Git manage source codes written by the developers.

**GIT HUB:**

Distributed version control & SCM. Git Hub tool is a centralized code repository and does version control.

**JENKINS:**

Continuous Integration/Continuous Deployment tool to automate process. Jenkins run jobs which build pipeline and automate the deployment process by integrating with all kinds of tool by simply by installing relevant plugins.

**CloudFormation:**

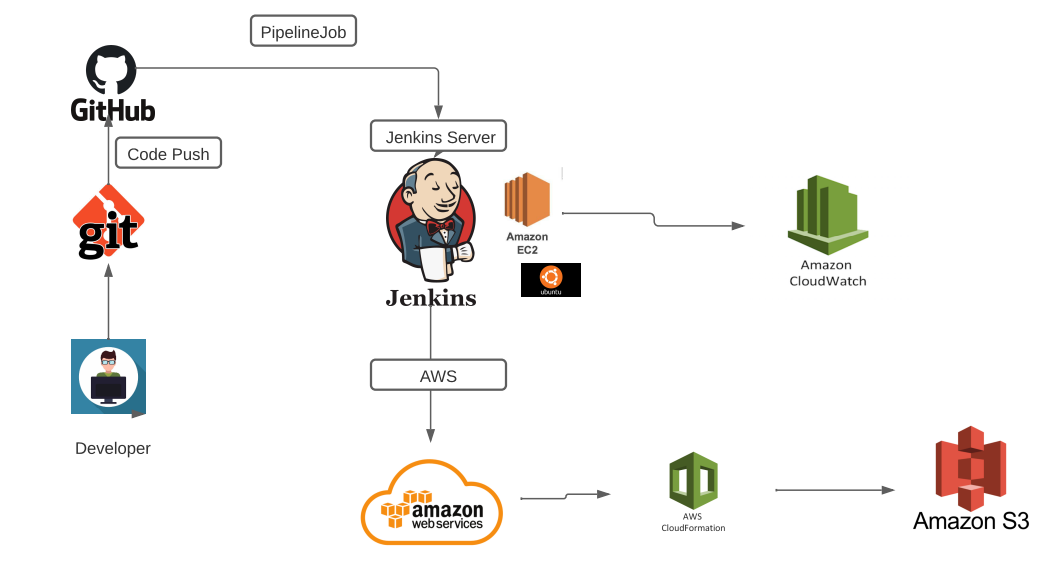
CloudFormation service is used to provision infrastructure in AWS platform. In this case I am spinning up S3 bucket with a template.

**CloudWatch:**

CloudWatch service used for monitoring AWS cloud infrastructure.

3.design

**Architecture Diagram**



* In this project developer write code and use code management tool to push it to centralized code repository [Git Hub].
* Create Jenkins declarative pipeline job which get triggered when it senses push build starts and deploys provisioning infrastructure in AWS platform.
* Monitoring configured with the running ec2 instance and its system performance watched in graphical representation.

The required set up has been done as given below:

4.set up Phase

1. In Amazon web services signup for an account and create an EC2 instance with below configurations.

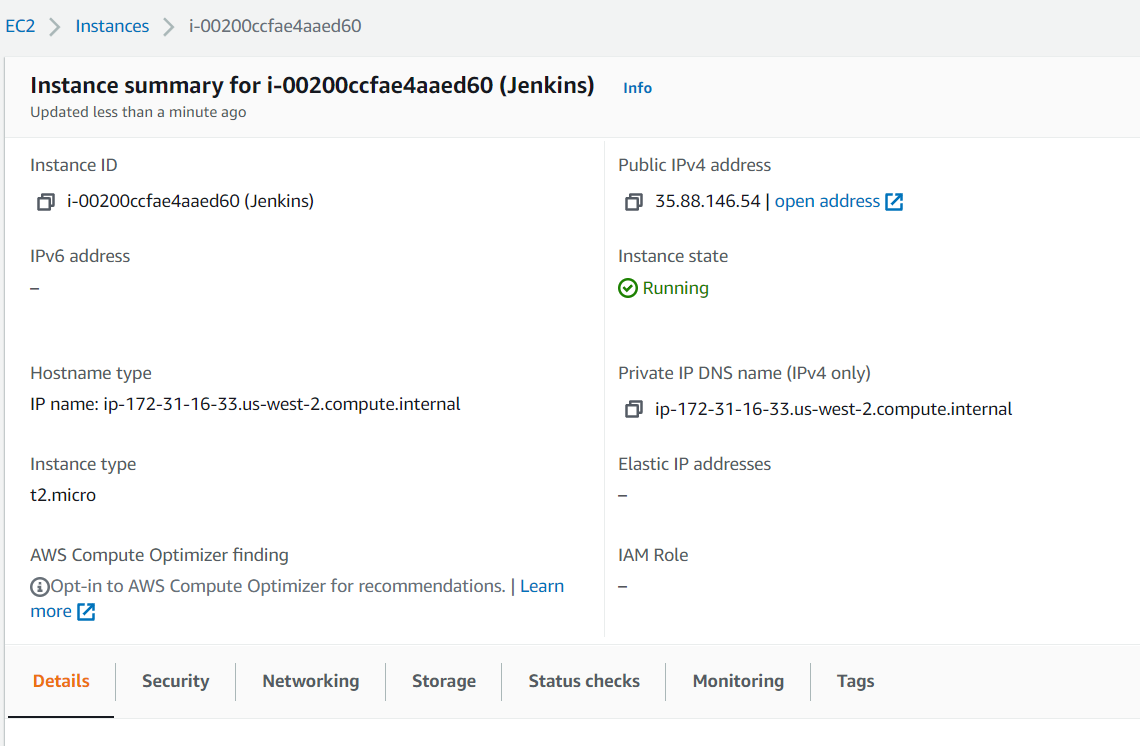
**Jenkin Server:**

Instance Type: t2.micro

Base Image: Ubuntu Server 20.04

Storage: 20 GB

Access: SSH Keypair



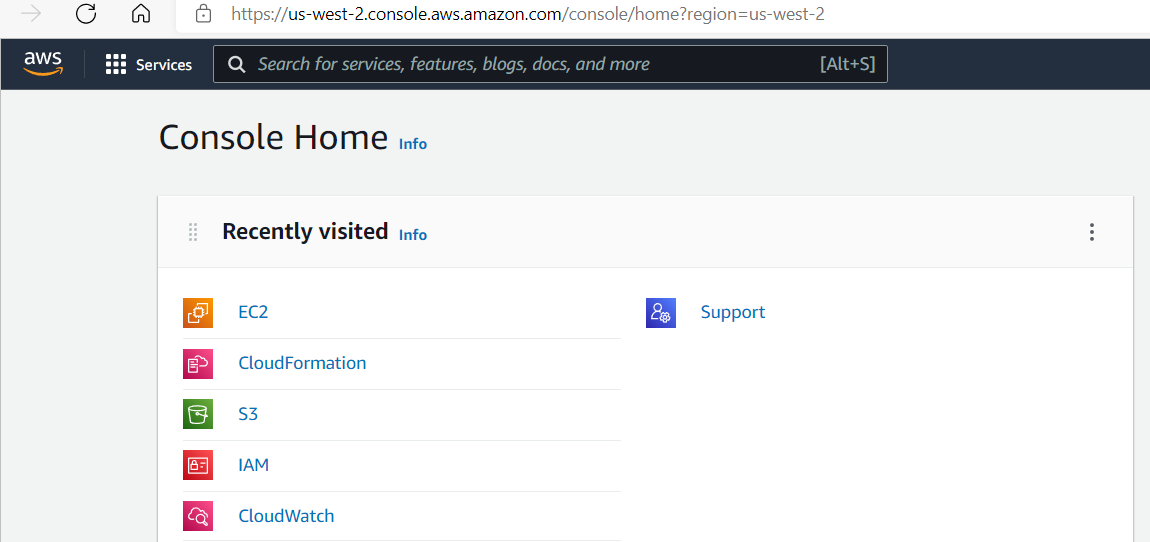
1. Enable necessary cloud services

EC2

IAM

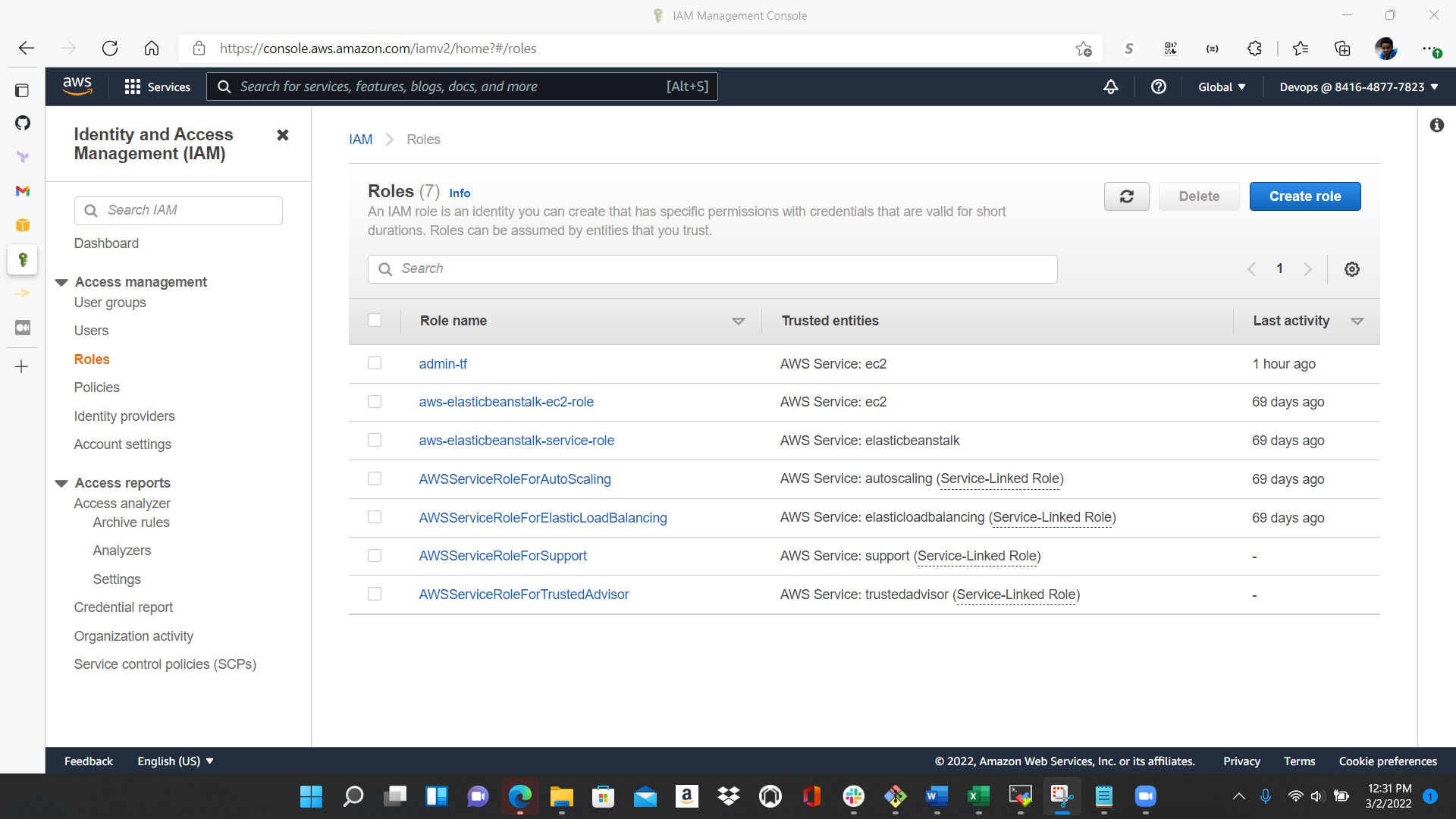
S3

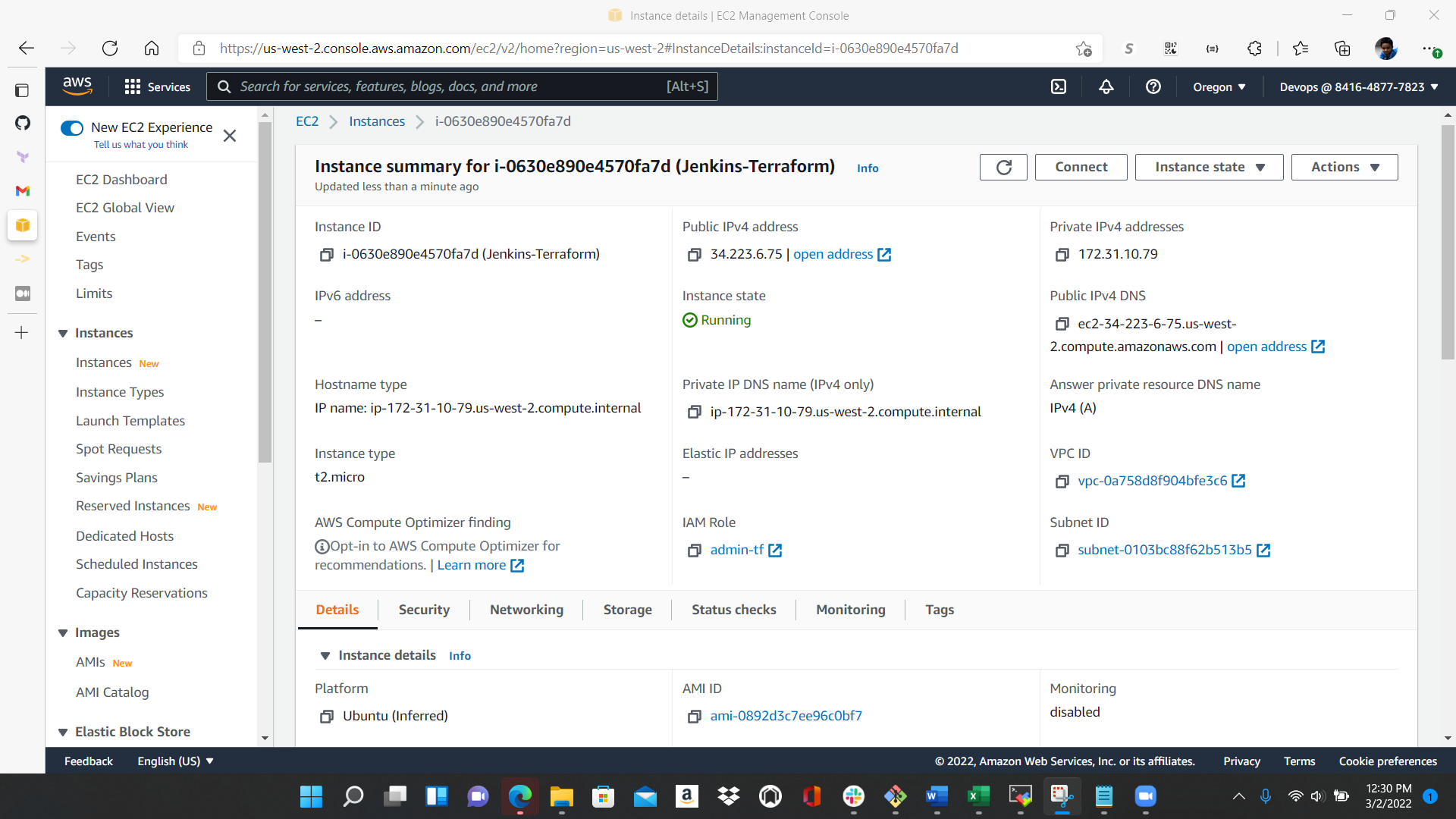
CloudFormation

CloudWatch

5.IAM ROLE CREATION

Create an IAM role with necessary permissions for EC2, S3, CloudFormation and CloudWatch access. Here I created role named “admin-tf” and attach this role with the Jenkins instance.





**Tools Installed:**

[**Jenkins 2.319.3**](https://www.jenkins.io/)

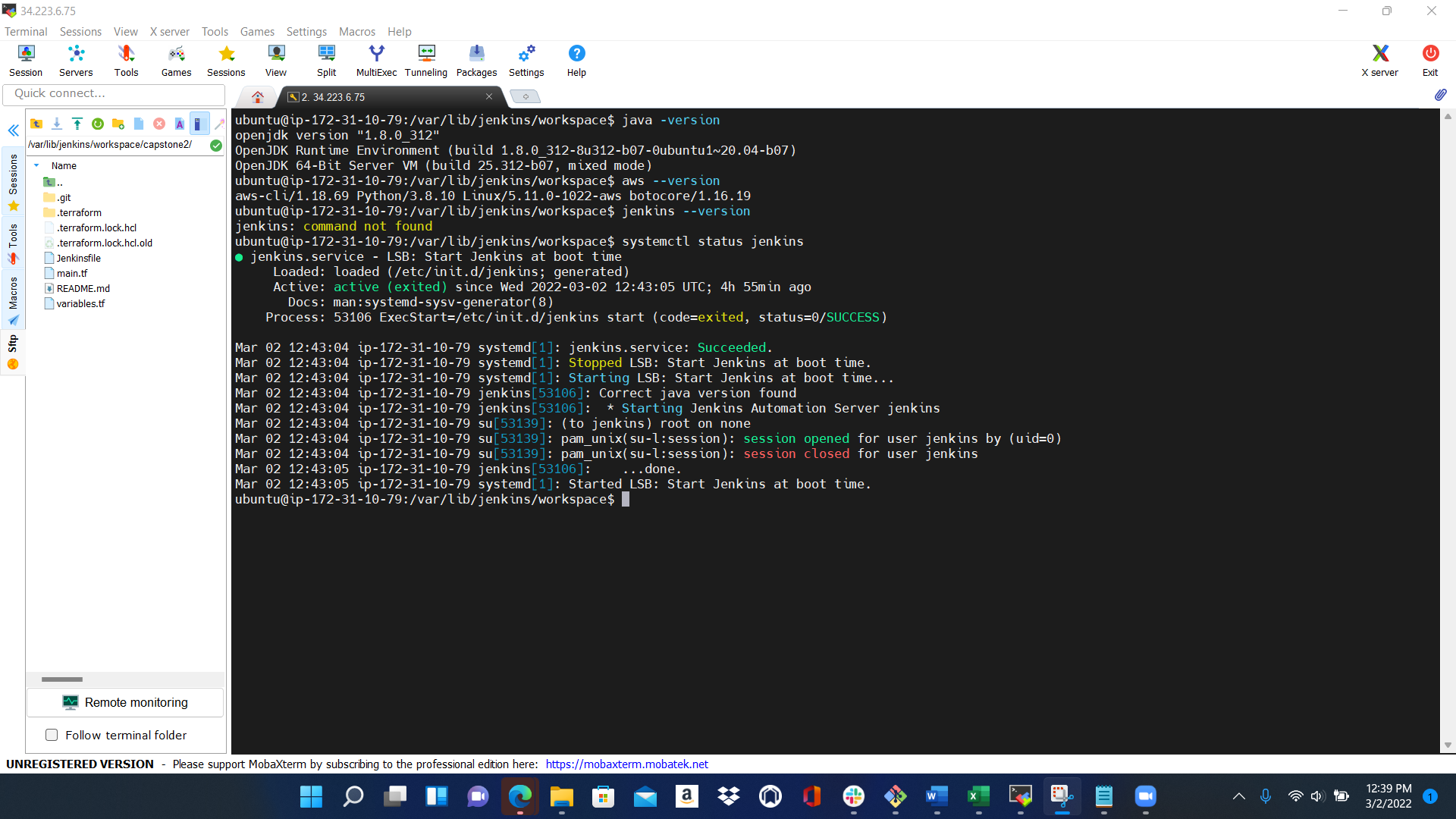
Git 2.34.0

aws-cli: 1.18.69

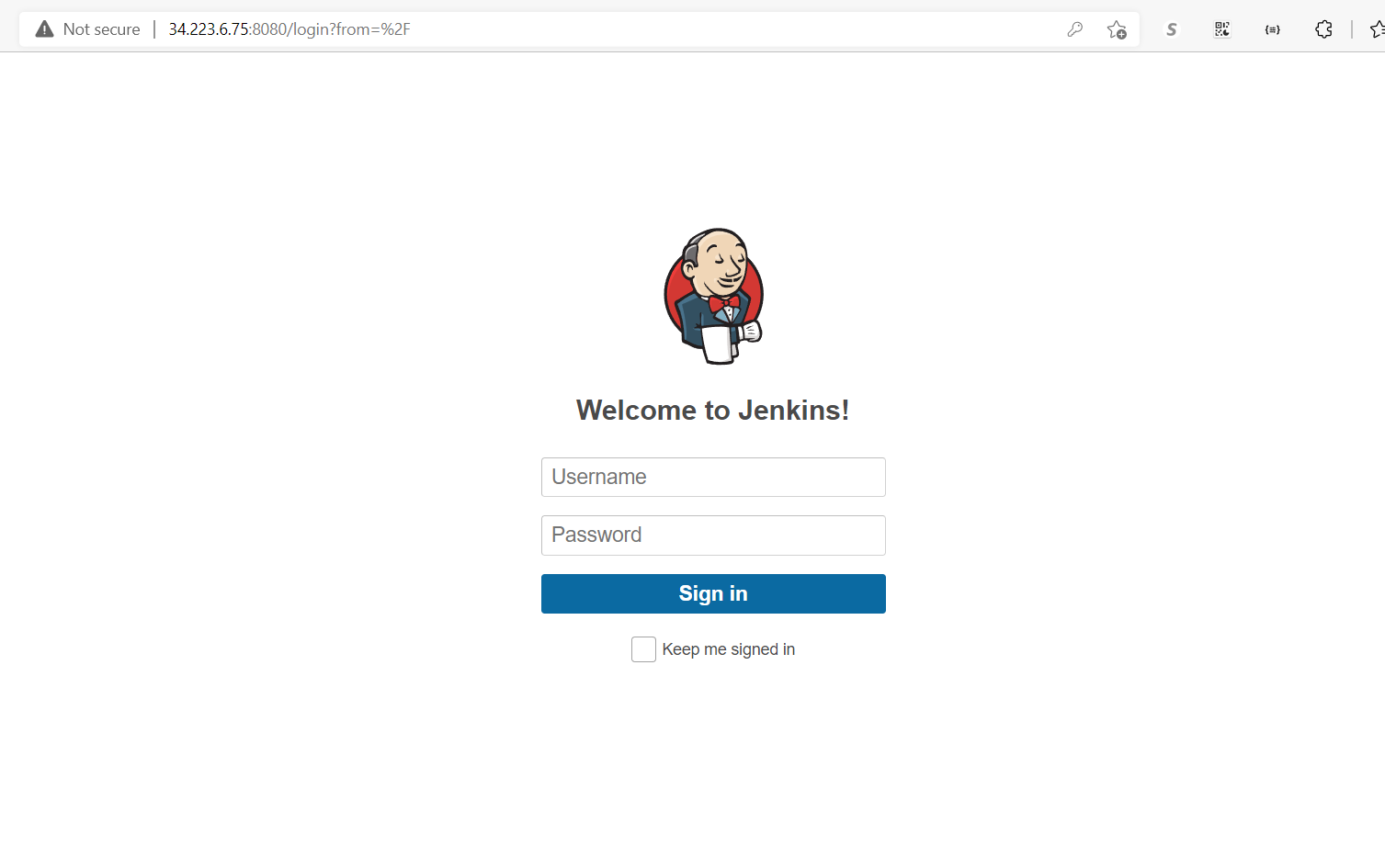
and you can check on the versions installed and service status to precheck are installations are smooth and working without any errors.

Steps involved in Jenkins Server setup:

* When the ec2 instances are started up and running install git, Jenkins with necessary plugins related to git, git hub, CloudFormation.
* After installation check the status of the services and it should be in running status.

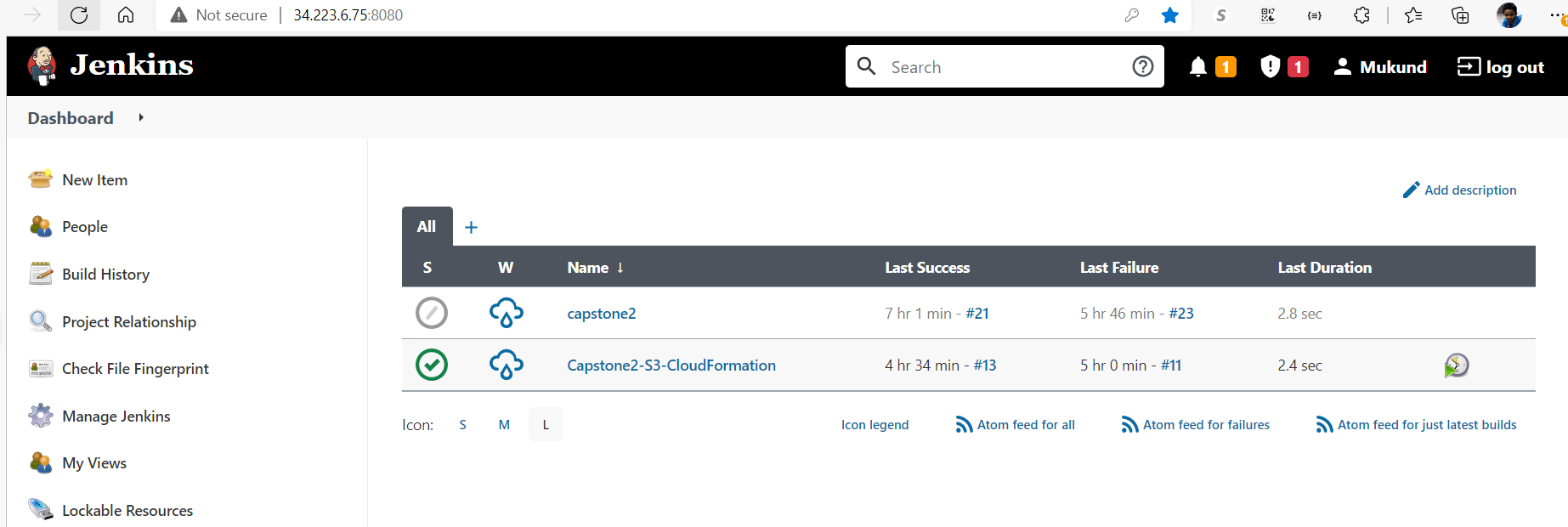


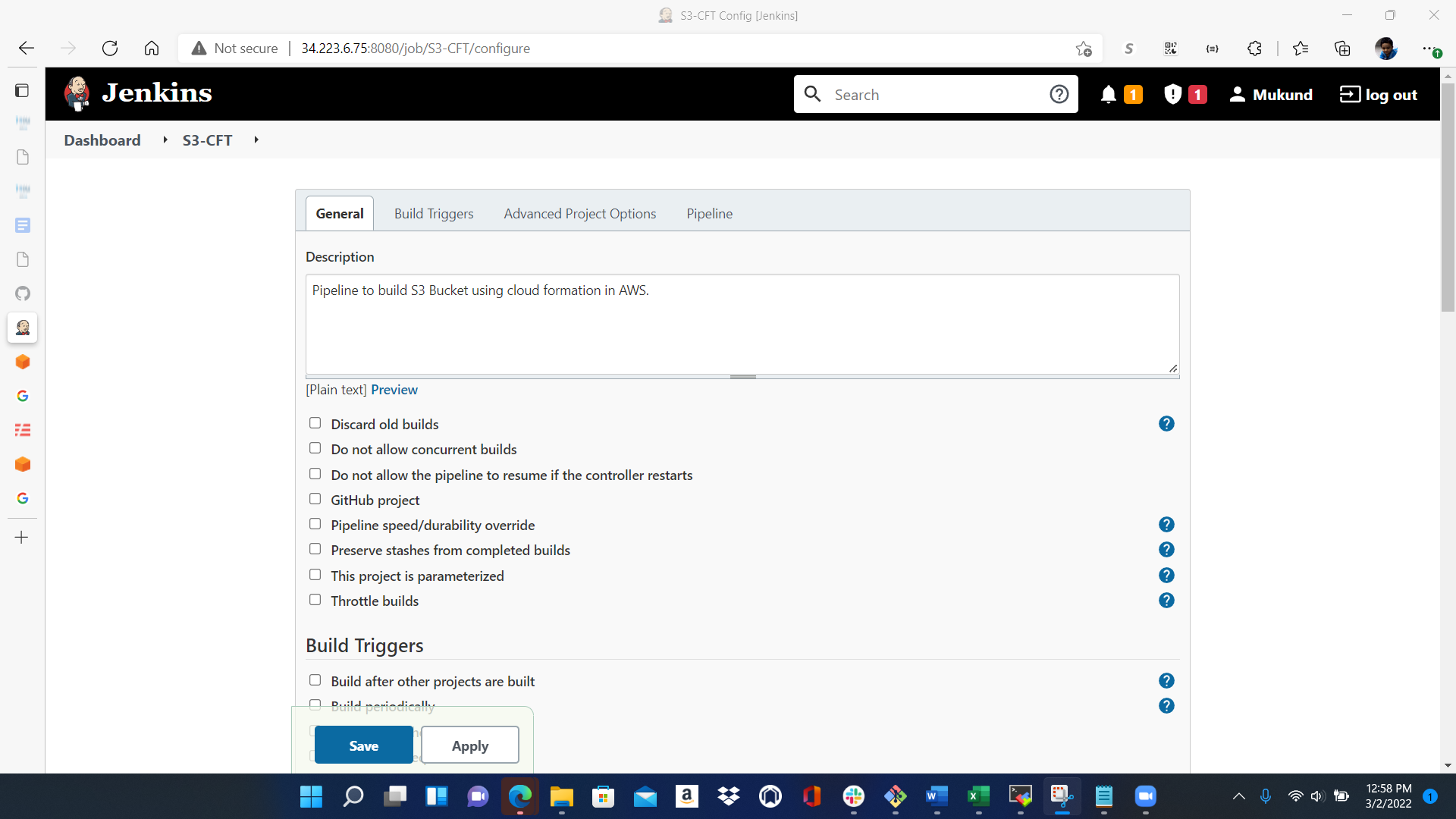
You can access your Jenkins console with your public ip of ec2 instance with default port 8080.

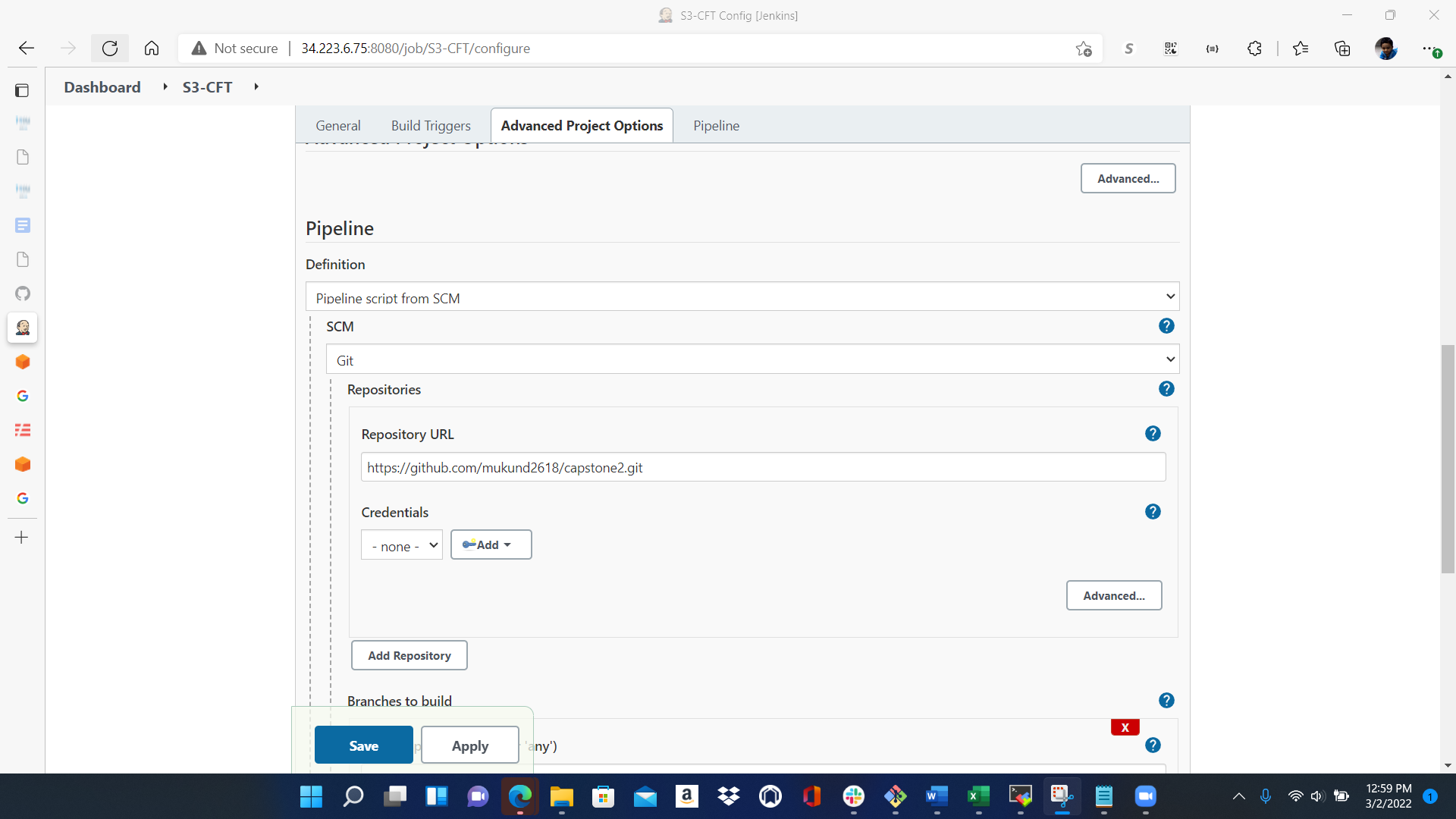


1. Create a Pipeline job in Jenkin console.

6.BUILD

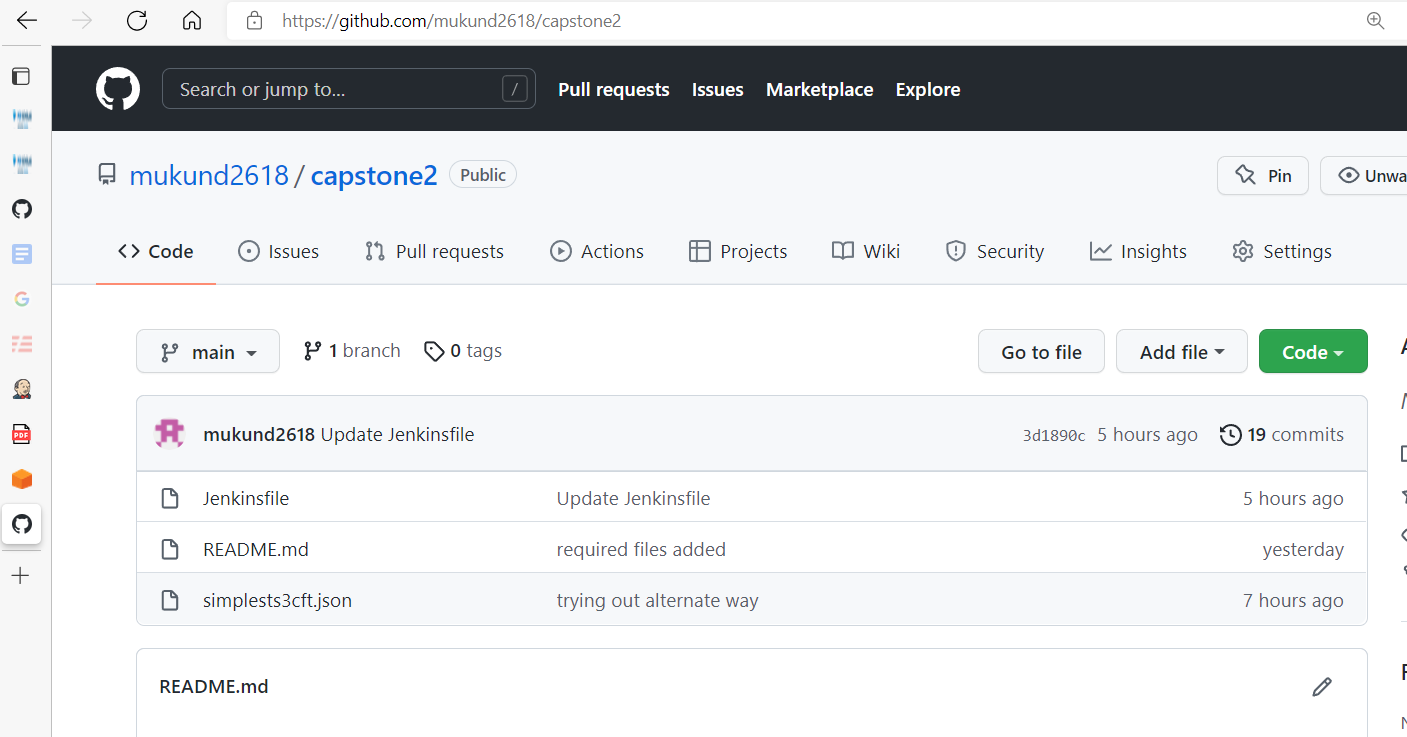




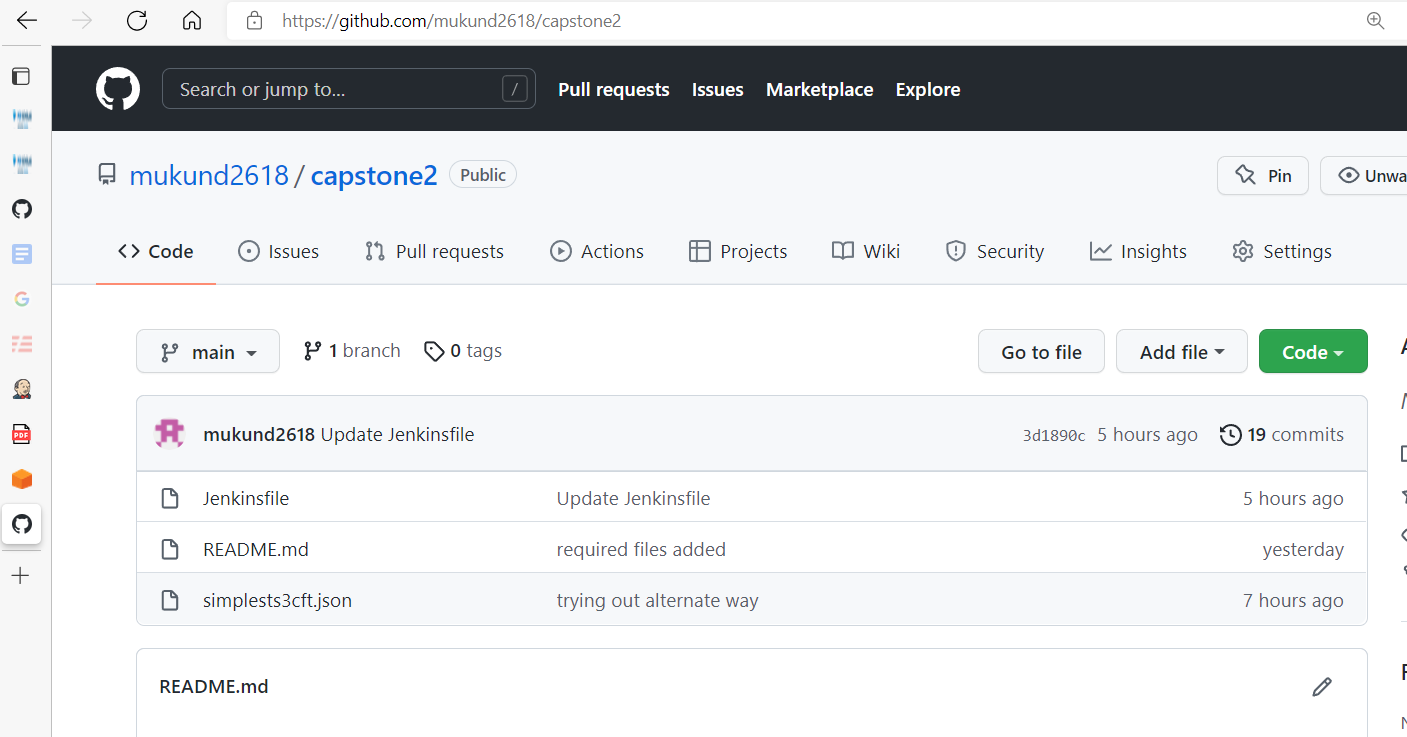


Through ssh login in to Jenkins server and do the below:

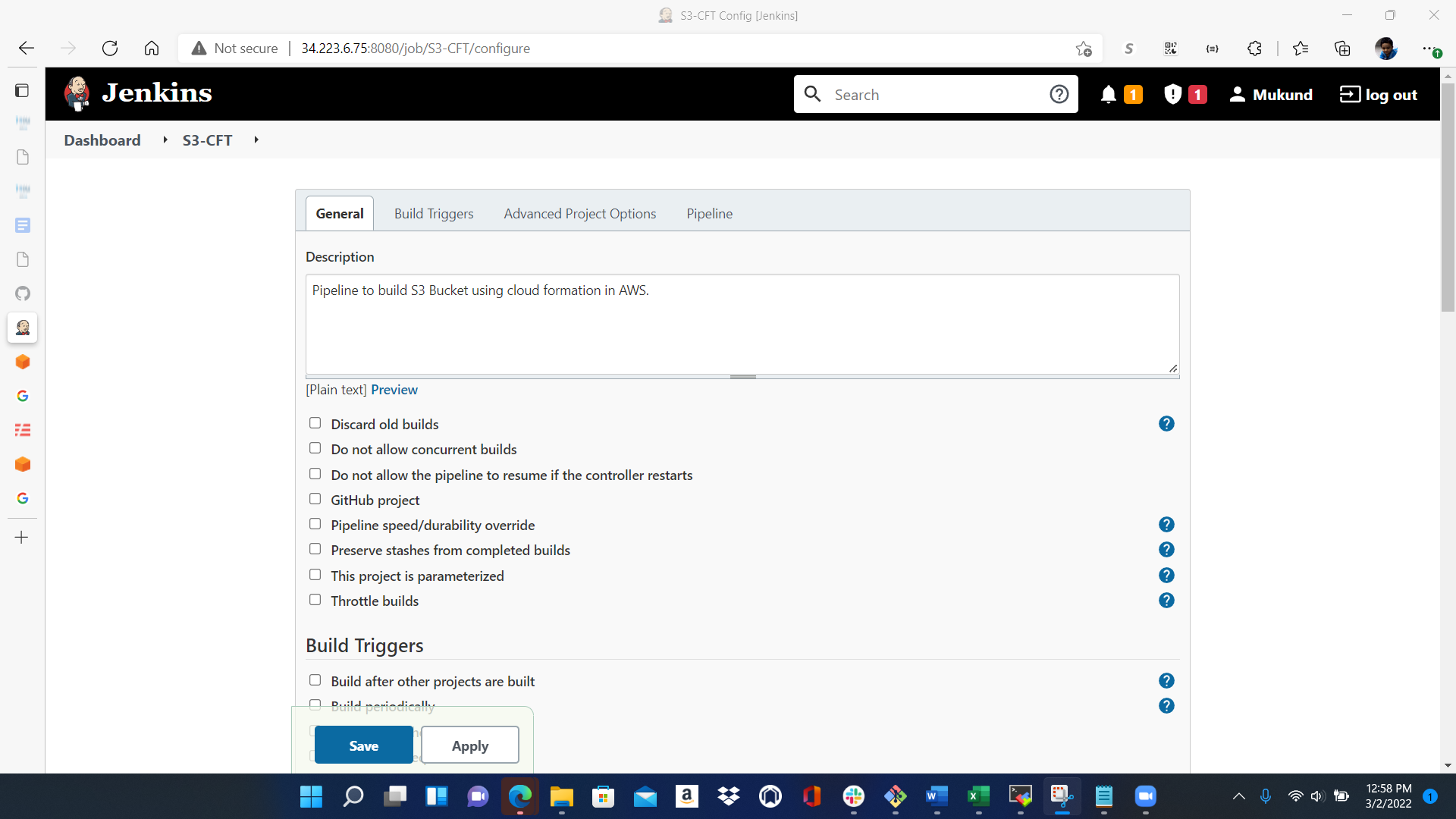
1. Develop necessary codes and dependencies and push it to github repository.
2. Login to your Github account and create a repository for this project.

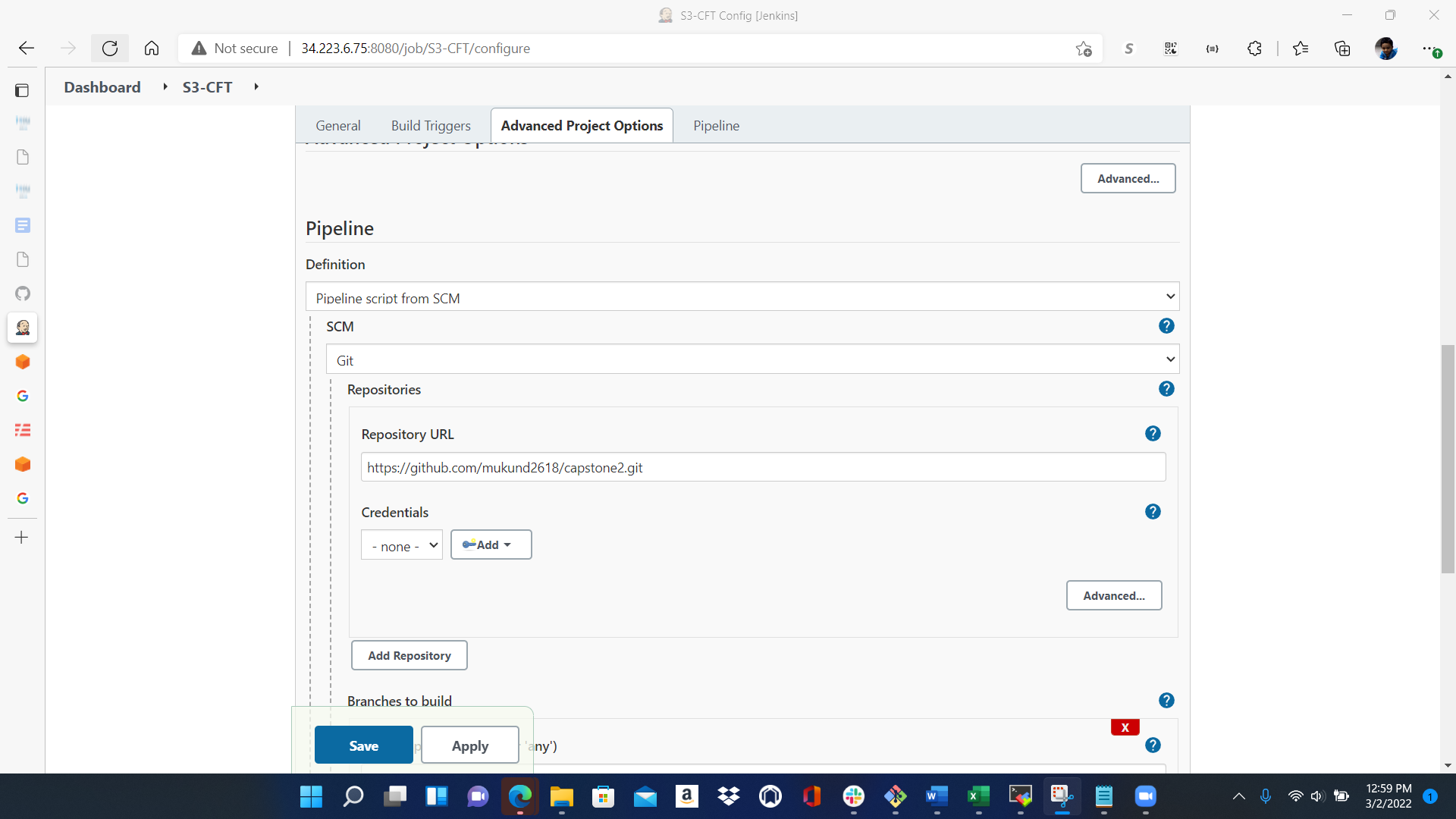


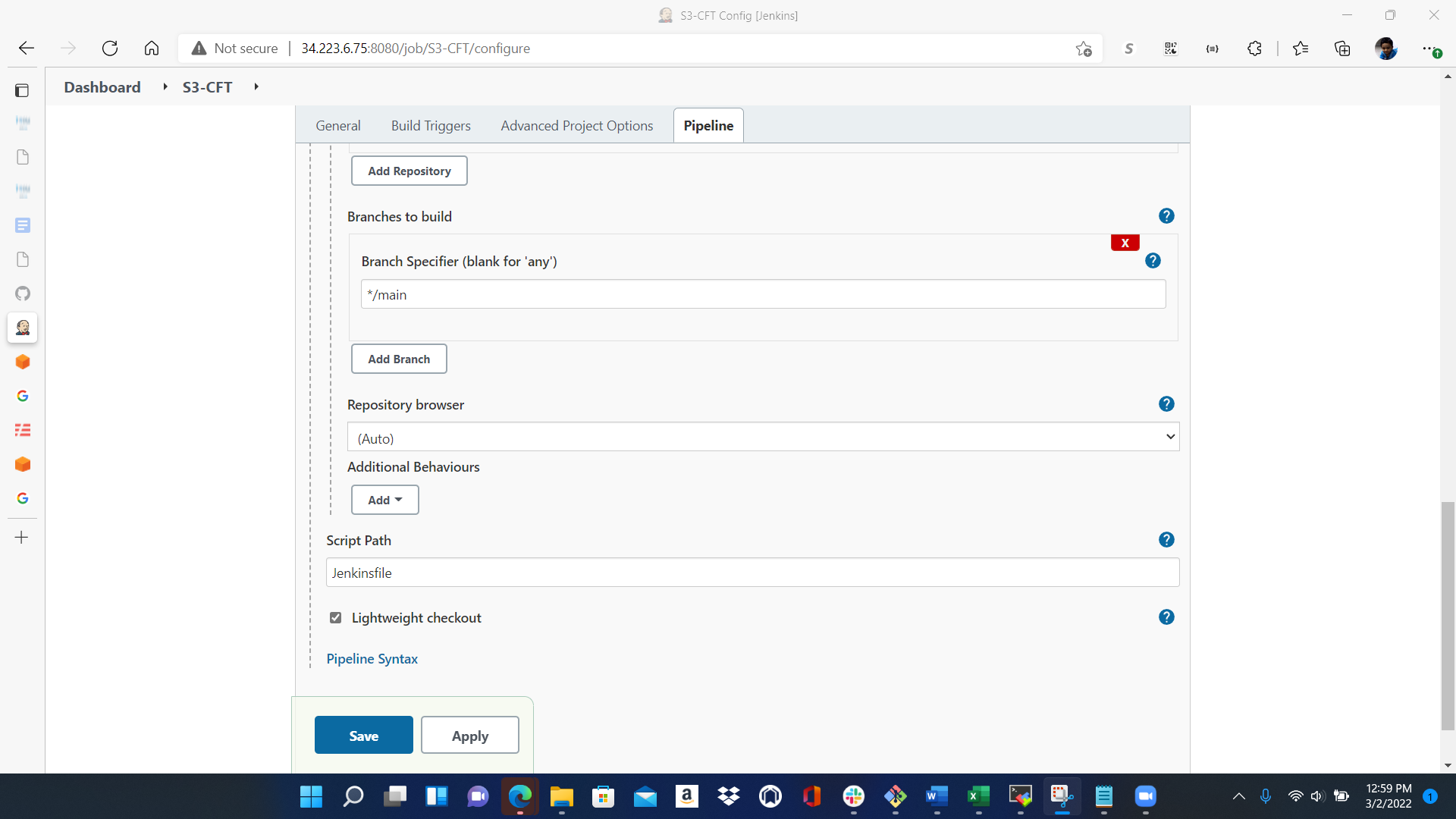
1. Push the related files to your git hub repository.



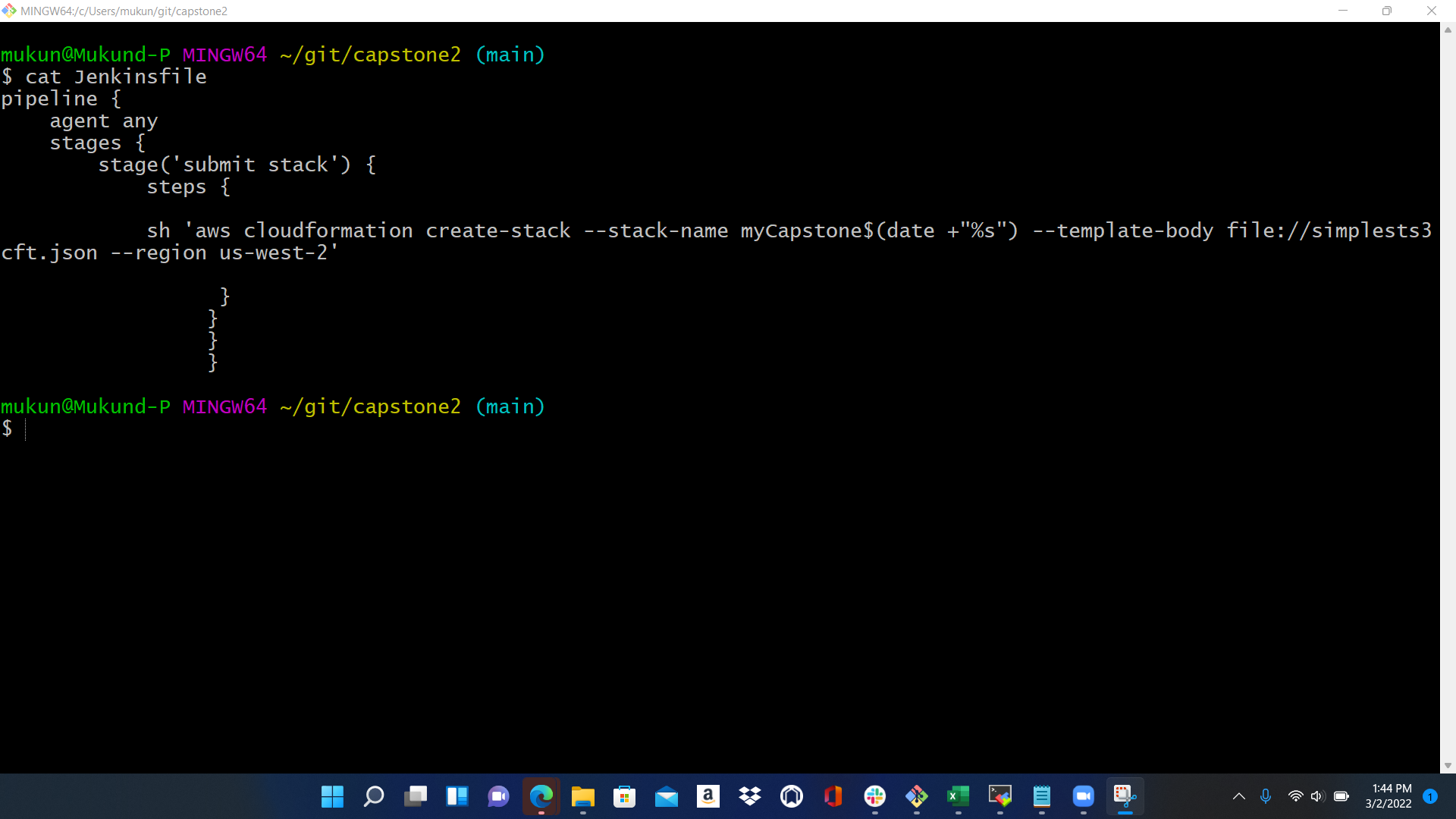
1. Integrate Jenkins to git hub repository through SCM in Jenkins pipeline.





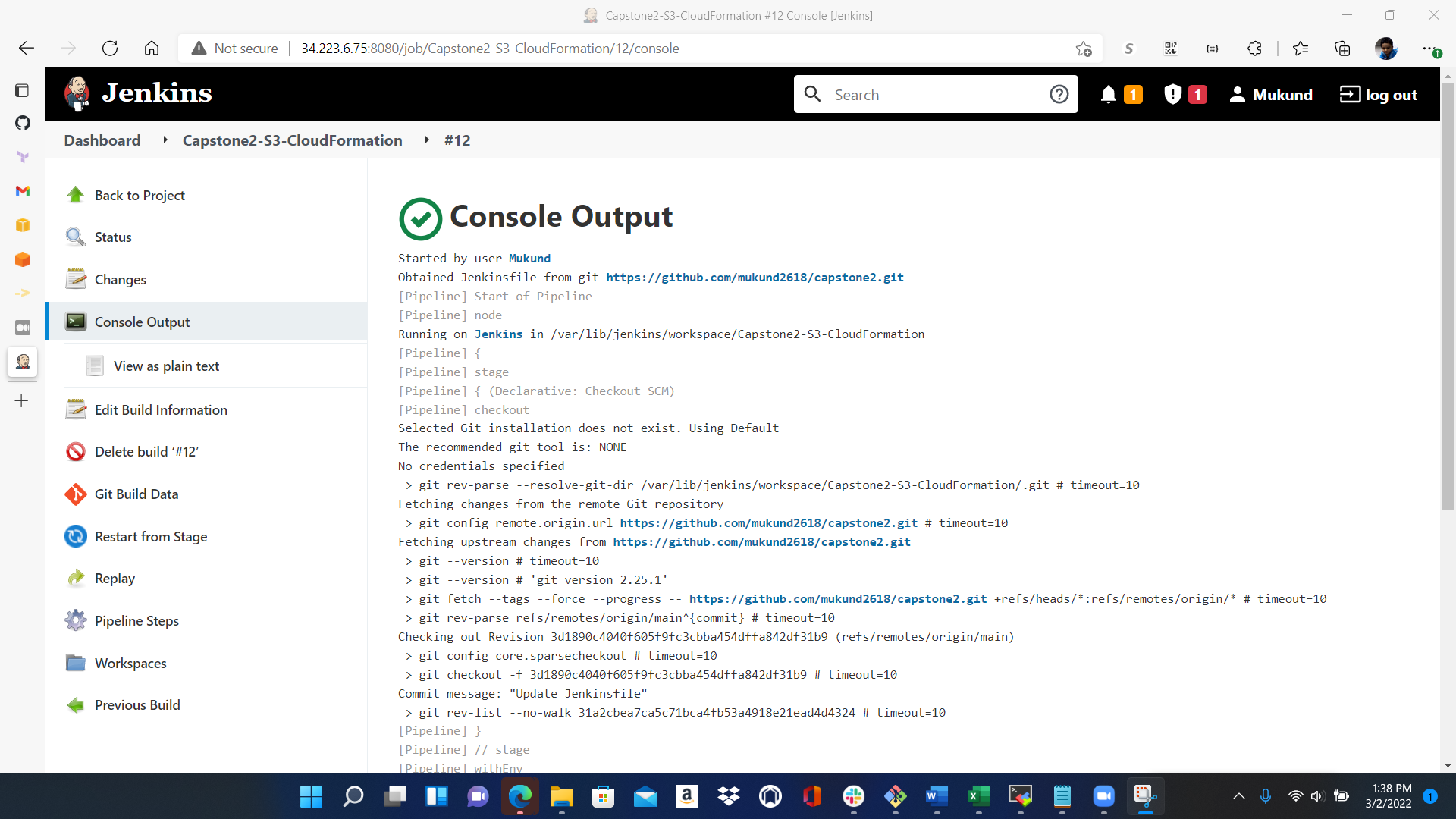


My Jenkinsfile and [simplests3cft.json](https://github.com/mukund2618/capstone2/blob/main/simplests3cft.json) file screenshot for your reference.

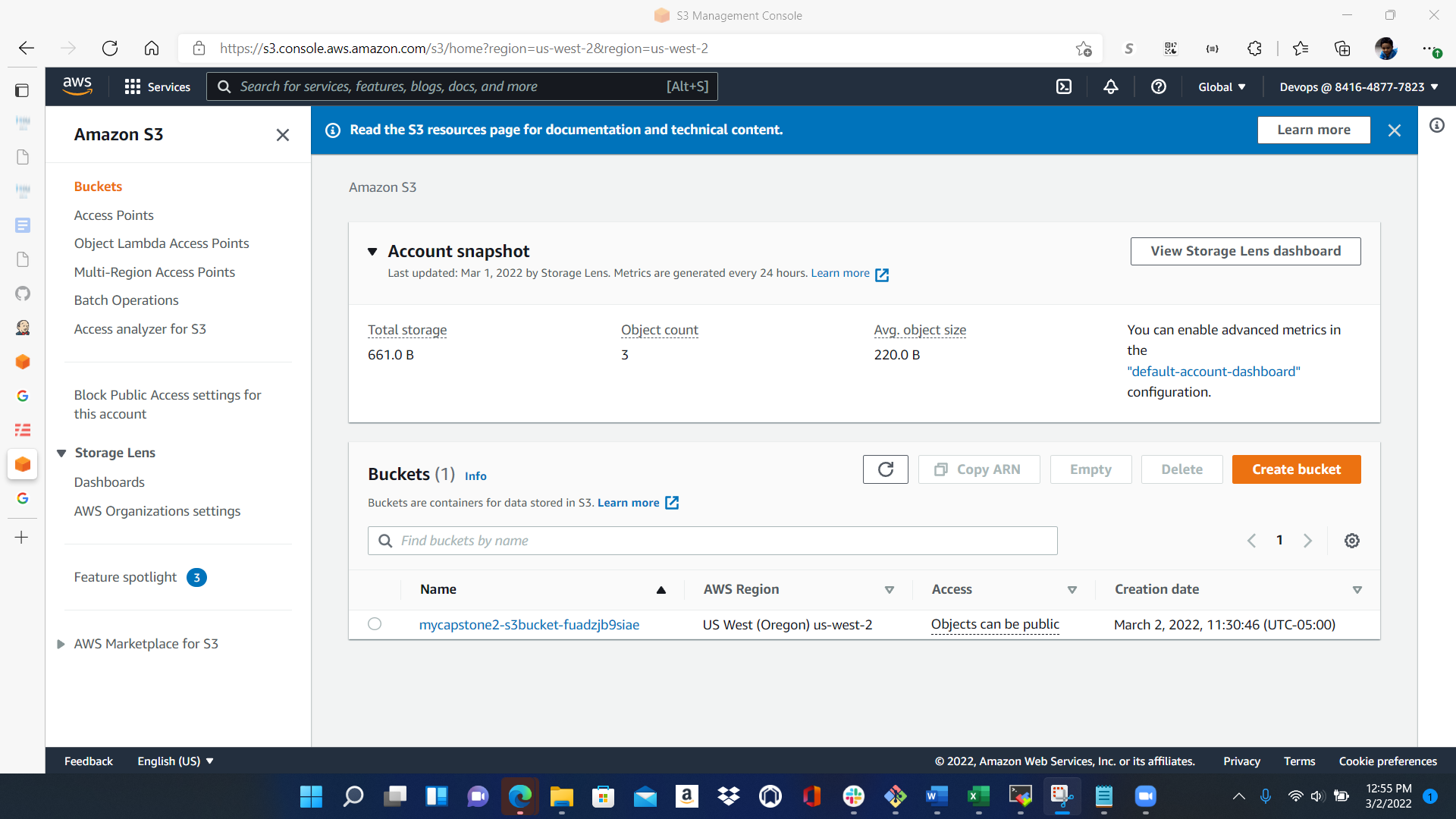




After the pipeline setup if you build the pipeline if everything goes well we can see the console output like the below screenshot.

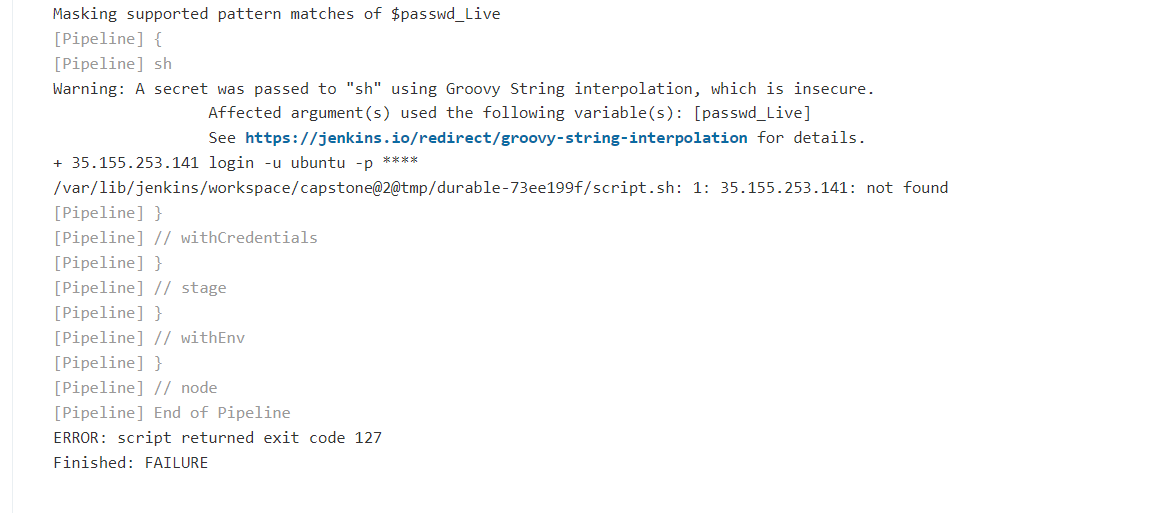


We can login to AWS console and check weather new S3 bucket with unique name created.

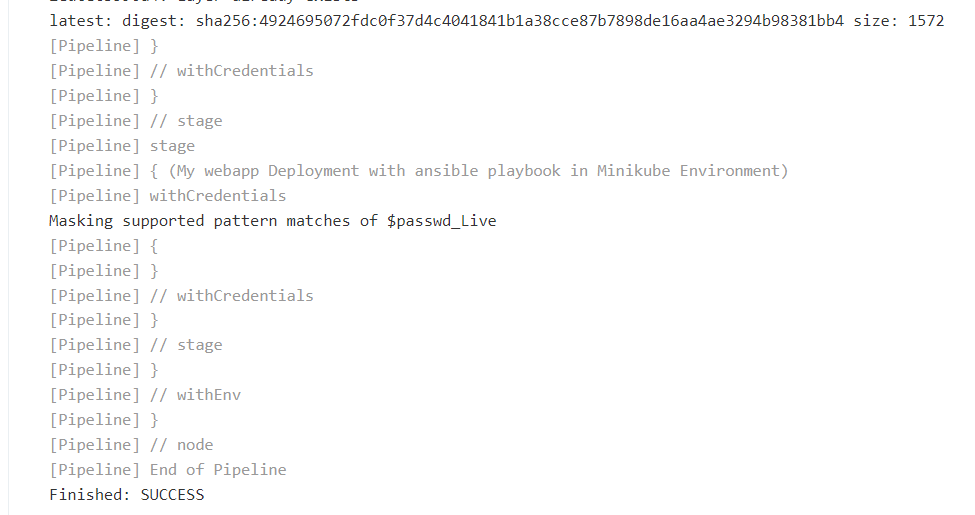


7.LINTING

Linting Screenshot:



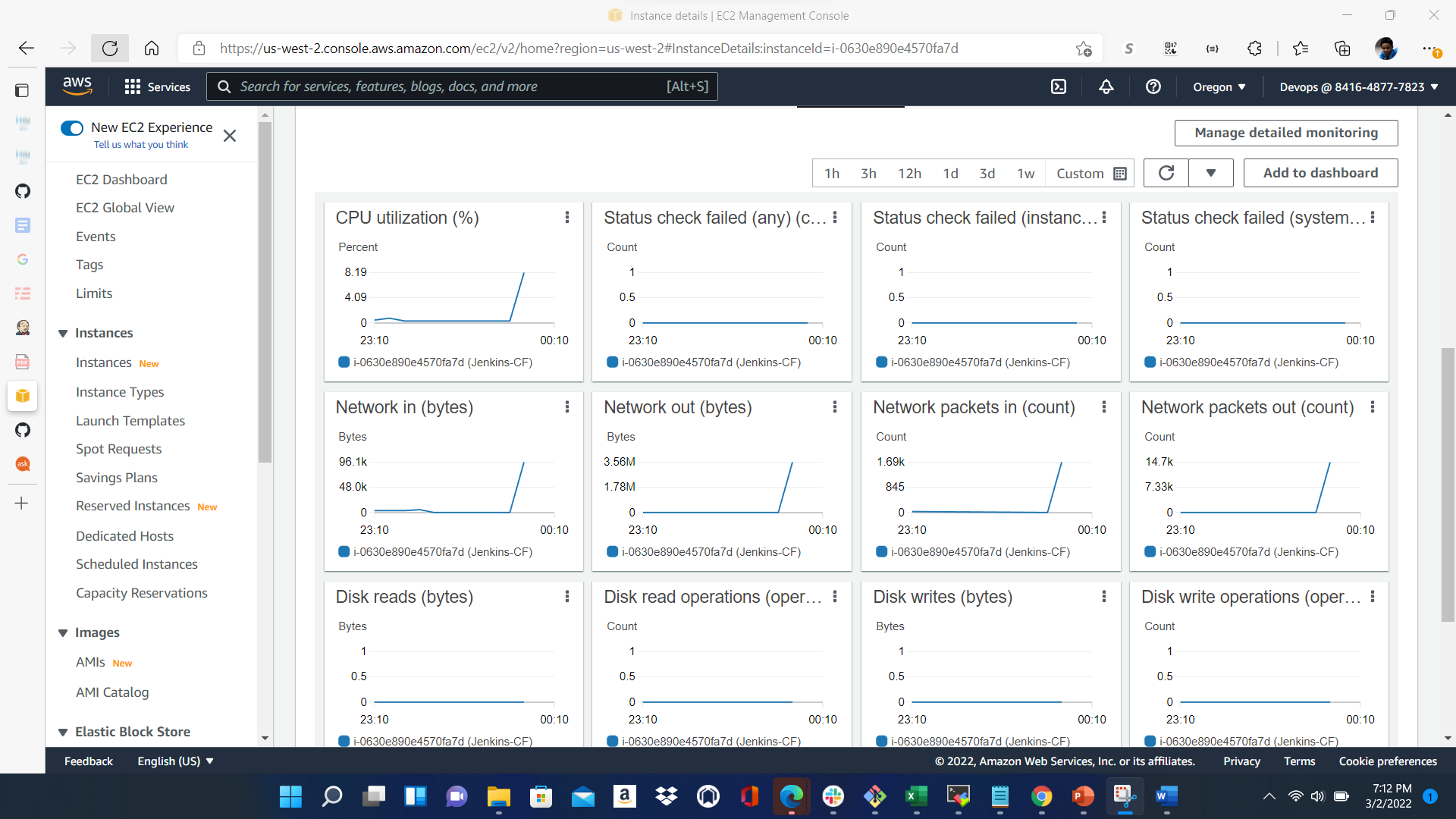
Jenkins is trying to search for file in wrong server and the same corrected.



You can see from the above image Jenkins configured to go and search in the right server where the file resides.

8.Monitoring

* Tested the server performance and stress test using yes command.



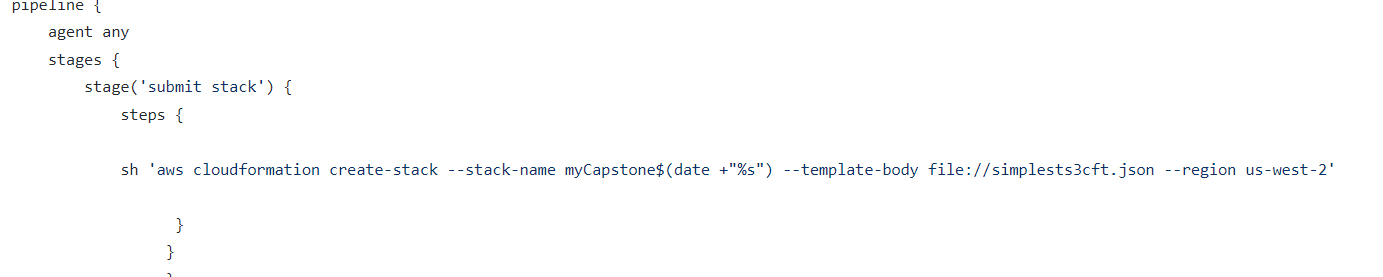
* Misconfigured jenkinsfile and tested in pipeline built.
* Tested end to end CI/CD pipeline build and the infrastructure provisioning successfully.

9.CHALLENGES

During repeat pipeline build process I see the s3 bucket provisioned successfully but output throws error which describes the stack already exists.



The same error corrected by adding “date” format of CloudFormation script line in Jenkinsfile.



10.improvisation

In EC2 instance whenever I restart production server the public ip changes and the CI/CD pipeline job fails. To overcome this issue, I improvised assigning elastic Ip in AWS environment.

The end