# Title: AWS CodeDeploy Implementation

1. **Introduction**: In this project, we explore the implementation of AWS CodeDeploy using two Amazon EC2 instances running Amazon Linux. The primary objective is to establish a streamlined deployment pipeline for a web application.
2. **Objectives**: The primary objective of this project is to implement AWS CodeDeploy using two Amazon EC2 instances running Amazon Linux 2 (AMZ2) in order to establish a streamlined deployment pipeline for a web application. By leveraging AWS CodeDeploy, we aim to achieve the following objectives:

**a**. **Automated Deployment**: Implement automated deployment processes to efficiently deploy updates and changes to the web application without manual intervention.

**b.** **Consistency and Reliability**: Ensure consistency and reliability in deployment procedures across different environments, promoting a stable and predictable deployment process.

**c**. **Separation of Concerns**: Establish a clear separation between the web server instance, configured with the AWS CodeDeploy agent for receiving deployments, and the developer workstation instance for coding and testing updates.

**e. Efficiency in Development**: Facilitate efficient development practices by providing developers with a dedicated environment for coding and testing, separate from the production environment.

By achieving these objectives, this project aims to enhance the deployment efficiency, reliability, and maintainability of the web application through effective utilization of AWS CodeDeploy with Amazon EC2 instances.

1. **AWS Services Used**: The services used in this projects are EC2,S3,CD,CP,CW,IAM,SNS.
2. **Project Implementation Details**: I have used two EC2 instance of AMZ2 Linux. First one is the web server we will be configuring, also called CodeDeploy agent. Second EC2 machine is supposed to use by developer where the codes are programmed. The names of the resources in the experiment are arbitrary and may name the resources your own.

(i) Create IAM Roles for EC2-S3-CodeDeploy access.

(ii) Create IAM user account for developer

(iii) Install and prepare the CodeDeploy agent on webserver.

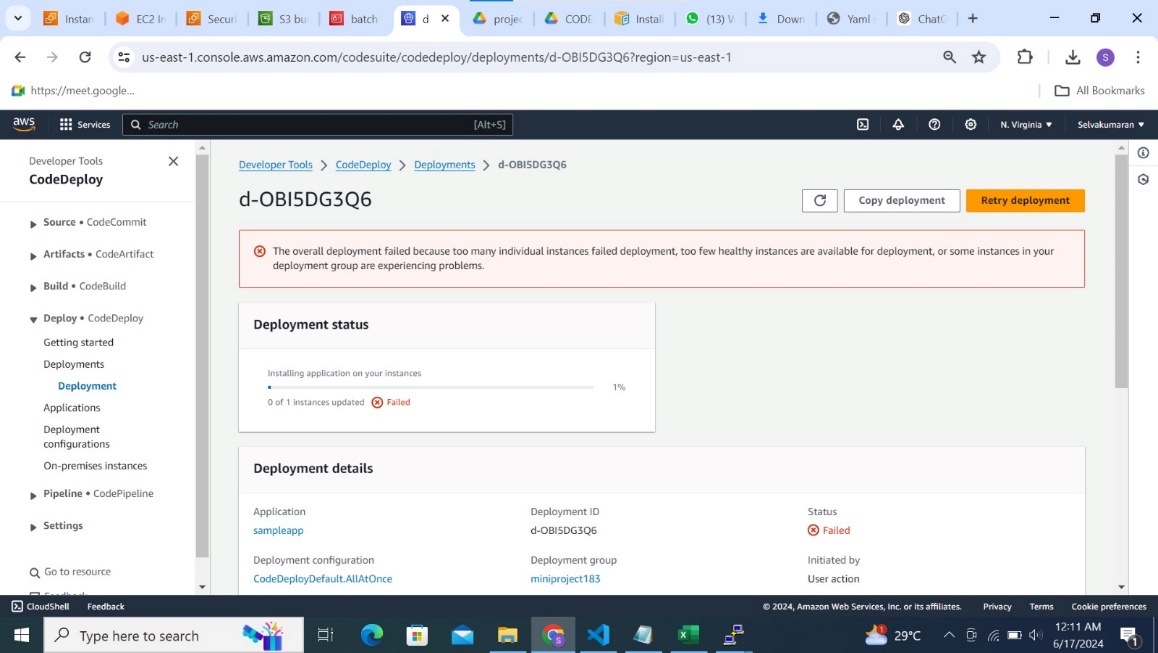
(iv) Create the code from Developer machine

(v) Create Codedeploy Application and Push the code to S3 bucket from Developer machine

(vi) Create Deployment Group to include web server

(vii) Create Deployment to push the code to the Webserver

(viii) Test the website configuration

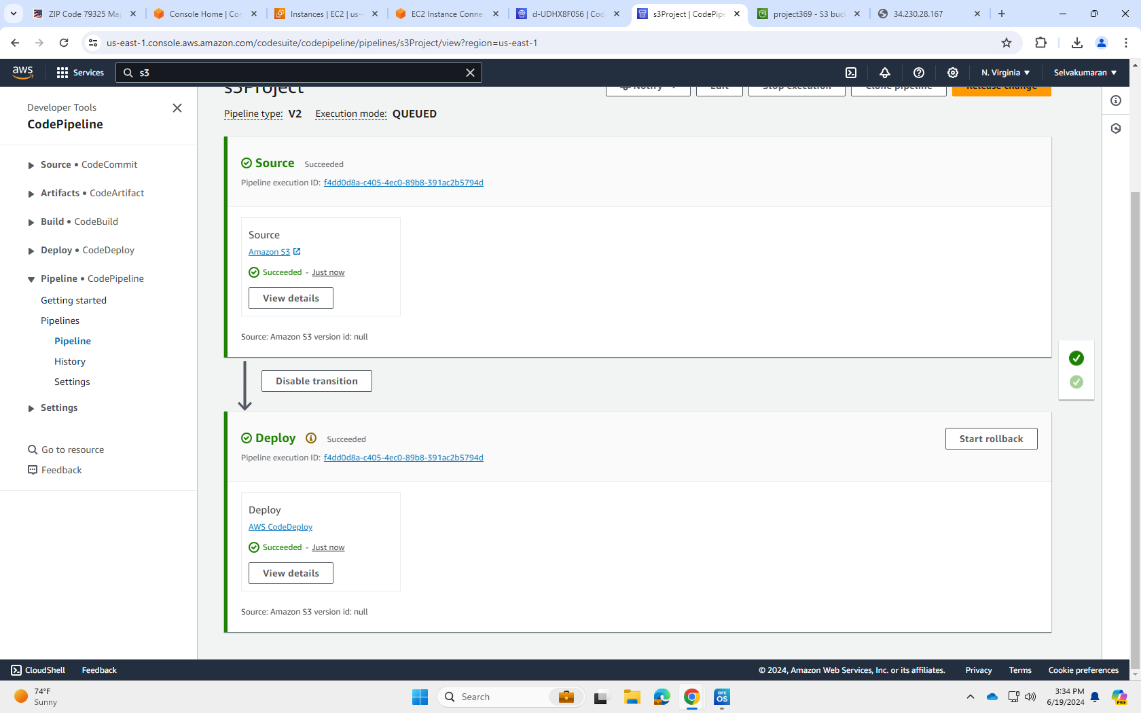
1. **Challenges Faced**: I have faced error while deploying in deployment

And I addressed the cause of error in the YAML code and I rectified the error by correcting the code A screenshot of a computer

Description automatically generated.

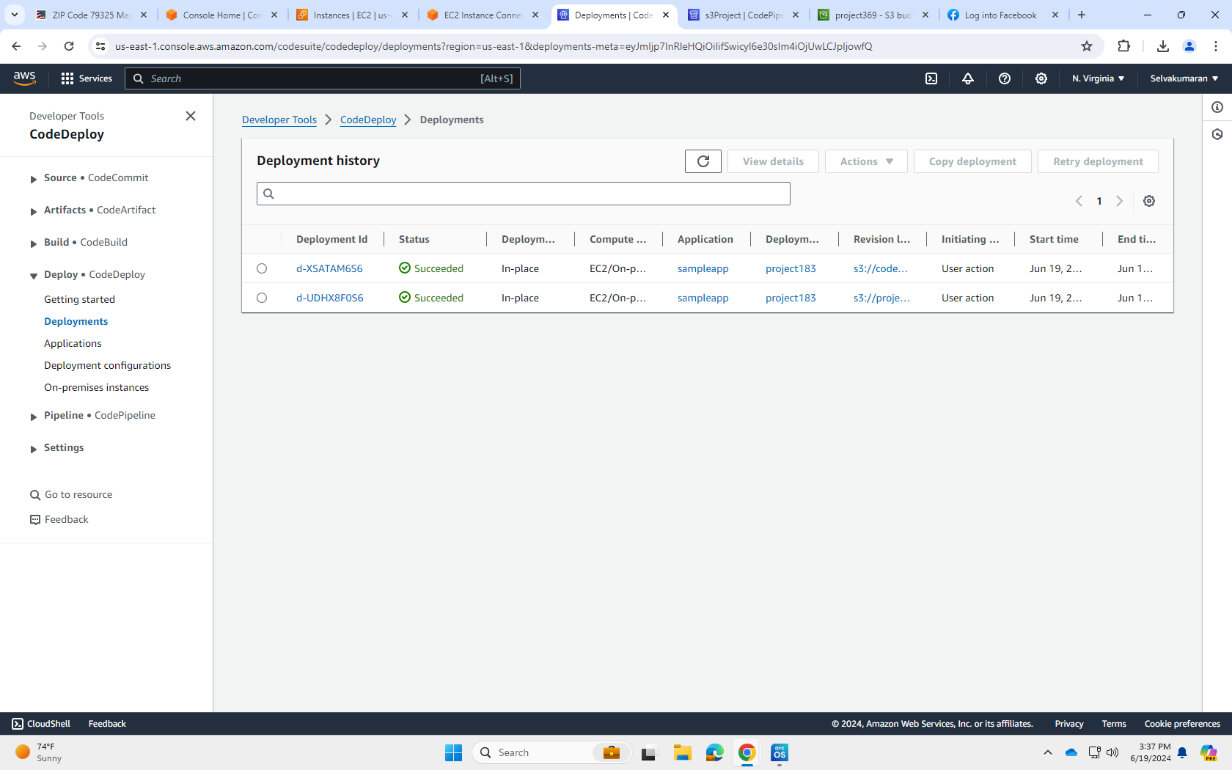
1. **Lessons Learned**: The Lesson i learned from this project is to always check and clarify the code is proper and without error before deployment/execution.
2. **Conclusion**: In conclusion, this project has successfully demonstrated the implementation of a deployment pipeline using AWS services, specifically utilizing two Amazon EC2 instances running Amazon Linux. The primary EC2 instance served as the web server, equipped with the AWS CodeDeploy agent for streamlined deployments. The second EC2 instance acted as a dedicated developer workstation where code changes were developed and tested.

**Output Screenshots** : These are the Screenshot of my project after Execution



A screenshot of a computer

Description automatically generated



.