CODE DEPLOY AND CODE PIPELINE PROJECT REPORT

INTRODUCTION

- In today's digital landscape, deploying web applications efficiently and reliably is crucial for businesses to maintain a competitive edge.
- AWS (Amazon Web Services) offers powerful tools like AWS Code Deploy and AWS Code Pipeline, which streamline the deployment process, enhance automation, and ensure consistent delivery of applications.
- The project aims to leverage AWS Code Deploy and AWS Code Pipeline to automate the deployment process of a web application.
- These services facilitate continuous integration, testing, and deployment, ensuring efficient and reliable updates to the application while maintaining scalability and minimizing downtime.

OBJECTIVES

- The objective of this project is to demonstrate how to deploy a web application onto a server using AWS Code Deploy and orchestrate the deployment pipeline using AWS Code Pipeline.
- By integrating source control, automating builds, and employing deployment strategies like blue/green deployments, the project enhances reliability and reduces deployment risks.
- This approach not only simplifies the deployment process but also ensures that updates and changes to the application can be managed seamlessly from development through to production.
- By achieving these objectives, we aim to equip students with valuable skills in deploying and managing web applications using AWS services, preparing them for careers in software development, DevOps engineering, and cloud computing.

AWS SERVICES USED

- **AWS Identity and Access Management (IAM)**: IAM is used to manage access to AWS services and resources securely. It provides roles and policies that allow Code Deploy and Code Pipeline to interact with other AWS services while adhering to the principle of least privilege.
- **Amazon EC2 (Elastic Compute Cloud)**: EC2 instances are virtual servers that can be used to host your web applications. Code Deploy deploys your application onto EC2 instances by connecting to them via the AWS Systems Manager or through user-defined scripts.
- **Amazon S3 (Simple Storage Service)**: S3 can be used as a source for your application artifacts and deployment files. Code Pipeline can retrieve application artifacts stored in S3 buckets as part of the build and deploy process.
- **AWS Code Deploy**: AWS Code Deploy automates the deployment of applications to EC2 instances, on-premises servers, AWS Lambda functions, or even to instances running on other cloud providers. It helps ensure rapid and reliable deployment of application updates.
- **AWS Code Pipeline**: AWS Code Pipeline is a fully managed continuous integration and continuous delivery (CI/CD) service that orchestrates the different stages of your release process. It allows you to model, visualize, and automate the steps required to release your application, including building, testing, and deploying code changes.

PROJECT IMPLEMENTATION DETAILS

IAM ROLES & USER:

- Create a role for S3 with policies allowing access to S3fullaccess.
- Create a role for Code Deploy allowing code deploy policies.
- Create IAM user with policies allowing full Administration access.

LAUNCH EC2 INSTANCE:

- Launch an EC2 instance as Production machine in Linux server and attach the S3 role in it.
- Launch another EC2 instance as Developer machine in Linux server.

PRODUCTION MACHINE CONFIGURATION:

- Open production machine CLI via Putty.
- Install and prepare code-deploy agent on web server.

DEVELOPER MACHINE CONFIGURATION:

- Open Developer machine CLI via putty.
- Attach IAM User to the developer machine using access keys.
- Create two directories and create a html file which contains source code of web application.
- Create yaml file to deploy the source code to webserver automatically.
- Create Code deploy Application and Push the code in Zip format to S3 bucket from Developer machine.

CODE DEPLOY:

- Navigate to aws code deploy console and create deployment group .
- Click create deployment which pushes code to the web server.
- The code will be successfully deployed.

CODE PIPELINE:

- Create code pipeline to enable automatic deployment.
- Both source and deployment got succeeded.

CHALLENGES FACED

- Implementing AWS Code Deploy and AWS Code Pipeline for hosting a web application faced various challenges related to deployment failures.
- Challenges arise when dealing case sensitive issues in yaml script.
- Challenges faced while pushing the code to s3 bucket (Eg.Appspec file renamed).

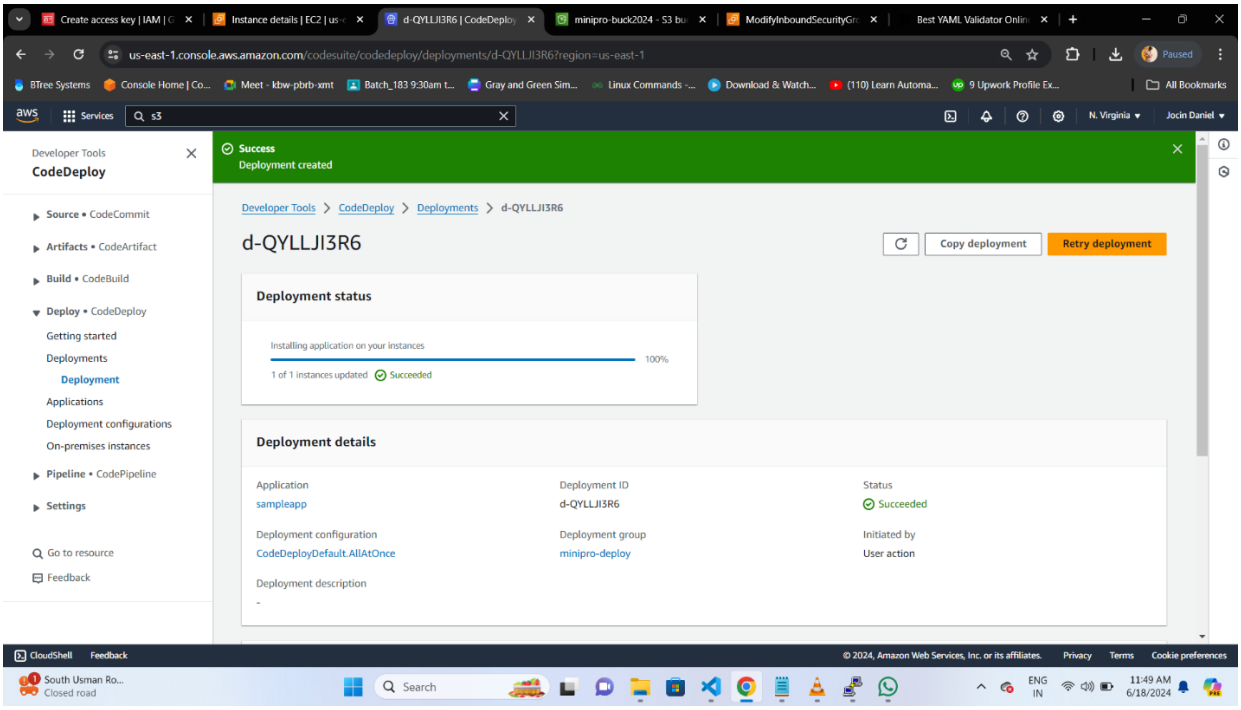
LESSON LEARNED

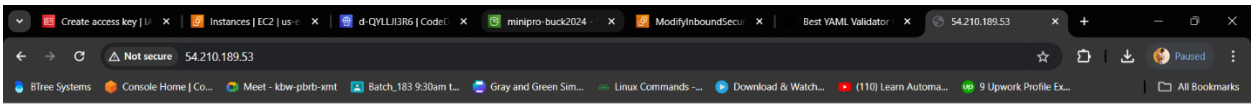
- Working on a project involving AWS Code Deploy and Code Pipeline to host a web application provides a hands-on learning experience in cloud deployment automation.
- Setting up AWS Code Pipeline involves configuring multiple stages (source, build, deploy) and integrating with various AWS services.
- Deployment failures can occur due to configuration errors, environment discrepancies, or application issues.

CONCLUSION

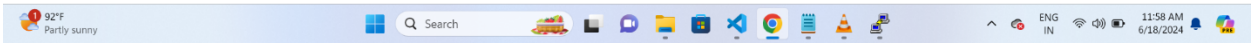
- In conclusion, AWS Code Deploy and Code Pipeline have proven instrumental in streamlining our deployment workflows, enhancing application reliability, and enabling rapid iteration and delivery of features.
- The project has not only strengthened our technical capabilities but also reinforced the importance of collaboration, adherence to best practices, and continuous innovation in achieving successful cloud deployments.

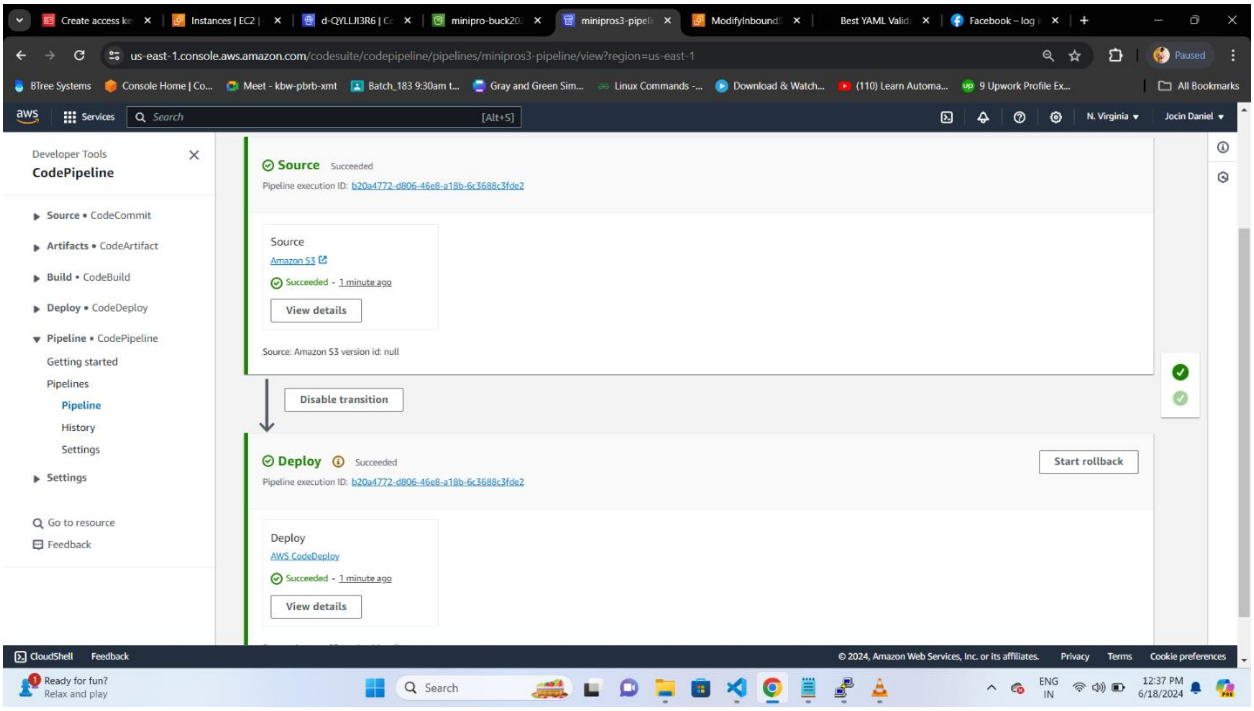
OUTPUT SCREENSHOTS

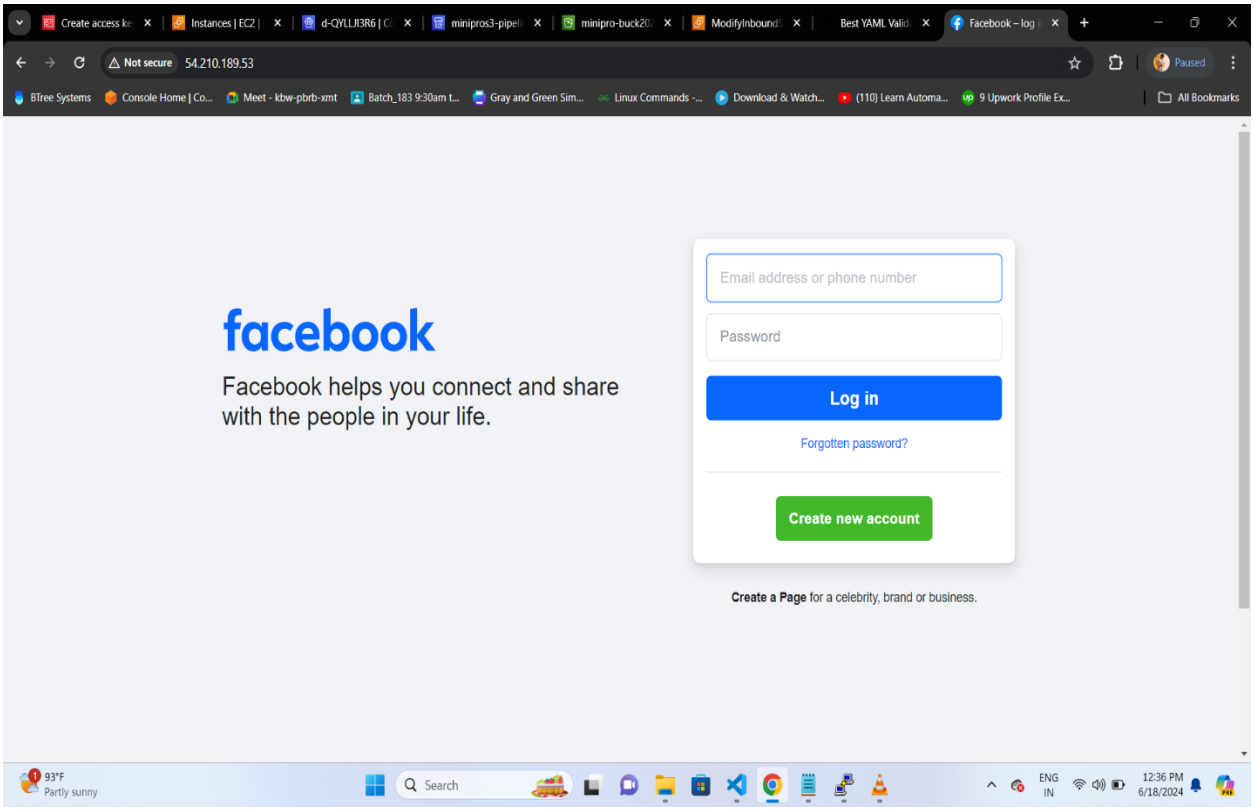




this is new project






Not secure 54.210.189.53
facebook
Facebook helps you connect and share with the people in your life.
Email address or phone number
Password
Log in
Forgotten password?
Create new account
Create a Page for a celebrity, brand or business.
93°F
Partly sunny
Search
ENG
IN
12:36 PM
6/18/2024