CODEDEPLOY AND CODEPIPELINE PROJECT REPORT

Objectives:

- The primary objectives of using AWS Code Deploy and AWS Code Pipeline are to automate and streamline the deployment process of applications, ensuring reliability, scalability, and efficiency throughout the software development lifecycle.
- Deploying the web application aims to showcase its functionality and features to users, clients, or stakeholders. This allows them to interact with the application in a live environment and understand its capabilities.
- Deploying a web application provides valuable learning opportunities for developers and teams, allowing them to gain hands-on experience with deployment tools, technologies, and best practices.
- Set up AWS Code Pipeline to continuously integrate changes from version control systems and trigger automated builds upon new commits or pull requests.

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 artefact's and deployment files. Code Pipeline can retrieve application artefact 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:

AWS Identity and Access Management (IAM)

Objective: Manage access to AWS resources securely.

Steps:

- **Create IAM Users and Roles:**
  - Open the IAM console.
  - Click "Users" and "Add user."
  - Set user details and assign permissions (e.g., attach policies like AmazonS3FullAccess, AmazonEC2FullAccess).
- **Create IAM Roles:**
  - Click "Roles" and "Create role."
  - Select "AWS service" (e.g., EC2).
  - Attach policies and complete the role creation.

- **Apply IAM Roles to EC2 Instances:**
  - When launching or updating an EC2 instance, attach the created IAM role to the instance for permissions.

Amazon EC2 (Elastic Compute Cloud)

Objective: Set up virtual servers to host the web application.

Steps:

- **Launch an Two EC2 Instance for Production and Developer machine**

  - Create the EC2 instance
  - Select an Amazon Machine Image (AMI), such as Amazon Linux 2.
  - Choose an instance type, e.g., t2.micro for free-tier eligibility.
  - Configure instance details (leave default settings).
  - Configure security group: Exiting security group and select the Alltcp access.
  - Advance settings: Created iam role for production machine
  - Review and launch.

- **Production machine Configuration:**

- Open the putty comment prompt for production machine
- Install the code deploy for production machine

- **Developer machine Configuration:**
  - Open the putty comment prompt for Developer machine
  - Attach the IAM user to the developer machine using the access keys
  - Create the two directory and create the html file to add source code
  - Create the YAML file to deploy the source code to web server automatically
  - Create code deploy application and push code to S3 bucket from developer machine

- **Code Deploy:**
  - Create the deployment group
  - Click create deployment which push code to web browser
- **Code Pipeline:**
  - Create code pipeline to enable automatic deployment
  - Both source and deployment got succeeded

Challenges Faced:

- Challenges faced when i started the project was putty was not working in my system
- Challenges arise when dealing with case sensitivity in YAML scripts.

Lessons Learned:

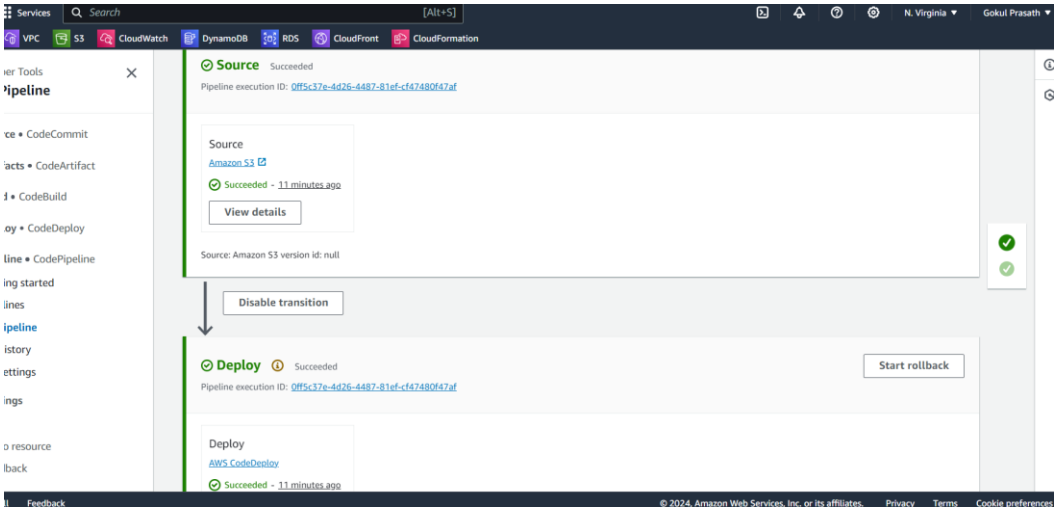
- **Understanding Deployment Tools:**
  - **AWS Code Deploy:** Learned how to configure and use AWS Code Deploy for automating code deployments.
  - **AWS Code Pipeline:** Gained proficiency in setting up pipelines that automate the build, test, and deployment phases.
  - Deployment failures are inevitable having robust role back strategy and automated error handling mechanism

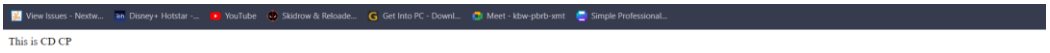
Conclusion:

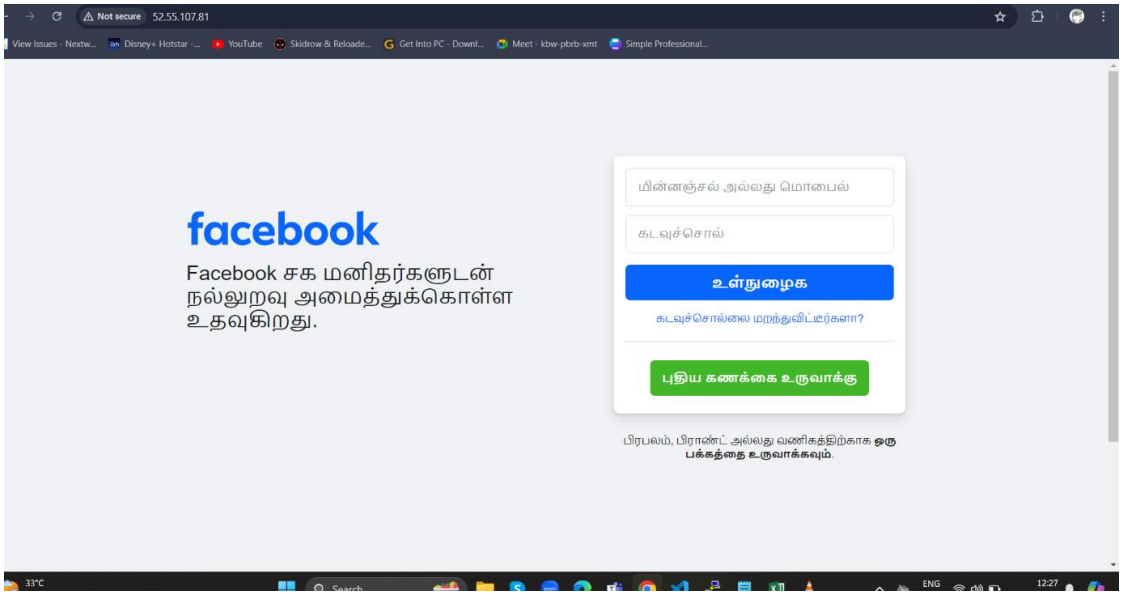
The project was a comprehensive learning experience that enhanced our understanding and skills in using AWS services. We discovered effective practices for managing cloud infrastructure, improved our automation capabilities, and recognized critical areas for future improvement.

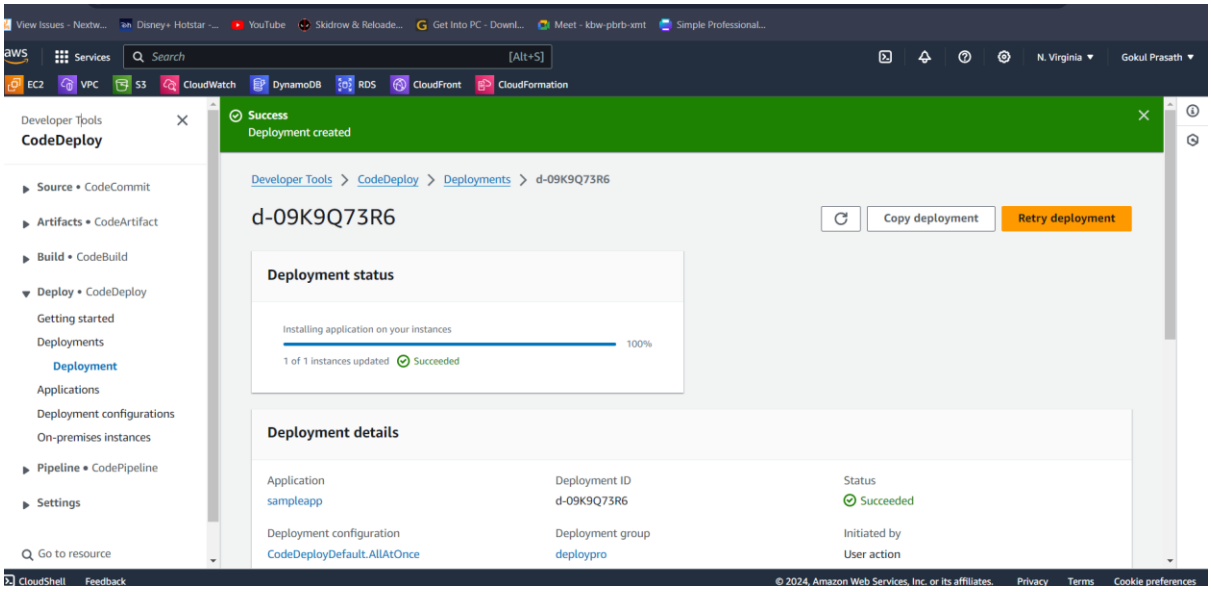
Overall, the project has better equipped us to deploy and manage scalable, secure, and efficient web applications in the cloud.

Output Screenshots:






Not secure 52.55.107.81
facebook
Facebook சக மனிதர்களுடன் நல்லுறவு அமைத்துக்கொள்ள உதவுகிறது.
மின்னஞ்சல் அல்லது மொபைல்
கடவுச்சொல்
உள்நுழைக
கடவுச்சொல்லை மறந்துவிட்டீர்களா?
புதிய கணக்கை உருவாக்கு
பிரபலம், பிரான்ட் அல்லது வணிகத்திற்காக ஒரு பக்கத்தை உருவாக்கவும்.


Developer Tools
CodeDeploy
Source • CodeCommit
Artifacts • CodeArtifact
Build • CodeBuild
Deploy • CodeDeploy
Getting started
Deployments
Deployment
Applications
Deployment configurations
On-premises instances
Pipeline • CodePipeline
Settings
Success
Deployment created
Developer Tools > CodeDeploy > Deployments > d-09K9Q73R6
d-09K9Q73R6
Copy deployment
Retry deployment
Deployment status
Installing application on your instances
100%
1 of 1 instances updated
Succeeded
Deployment details
Application
sampleapp
Deployment ID
d-09K9Q73R6
Status
Succeeded
Deployment configuration
CodeDeployDefault.AllAtOnce
Deployment group
deploypro
Initiated by
User action