

Assignment Number: 4

Name: Soham Satam Branch: IT/V Roll No.:109

Date:10/09/2023

Aim: To study AWS Code Pipeline and deploy web application using Code Pipeline.

LO mapped: LO1, LO2

Theory:

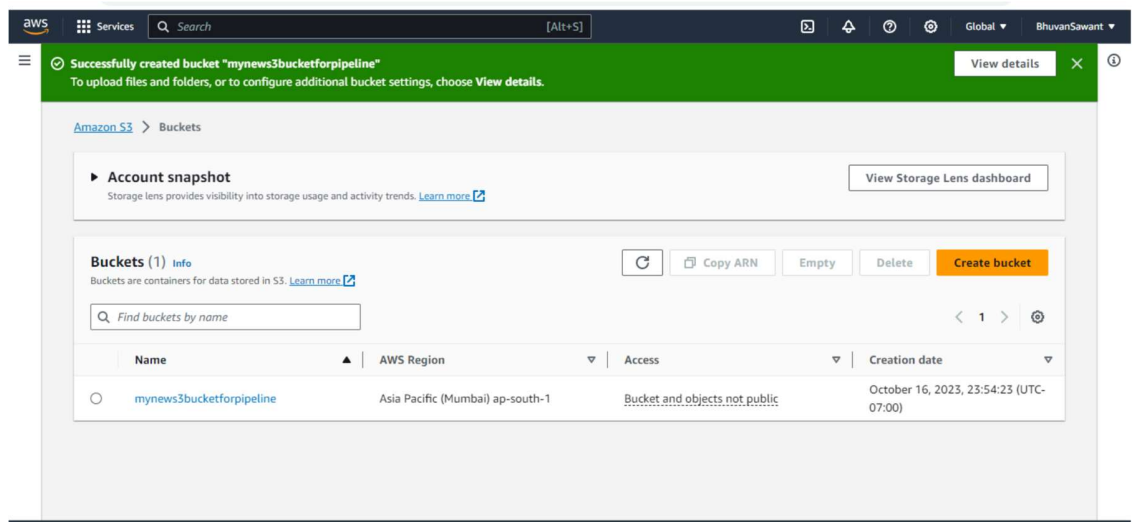
AWS CodePipeline is a continuous integration and continuous delivery (CI/CD) service provided by Amazon Web Services (AWS).

The key aspects of AWS CodePipeline:

Overview:

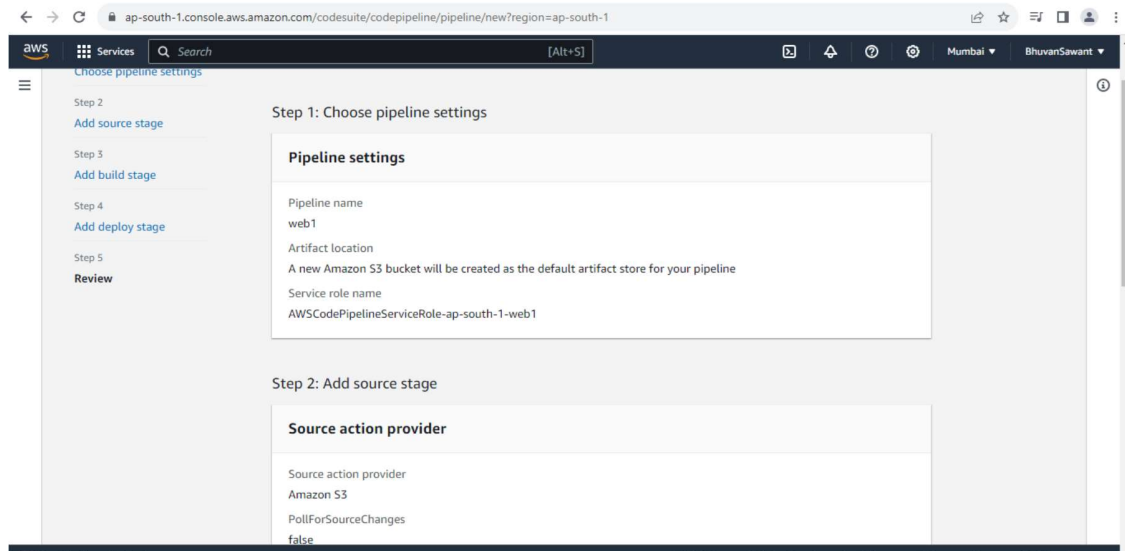
1. CI/CD Workflow:

- AWS CodePipeline facilitates the automation of the build, test, and deployment phases of the release process. It allows you to define a series of stages, each of which can represent a phase in your release pipeline.



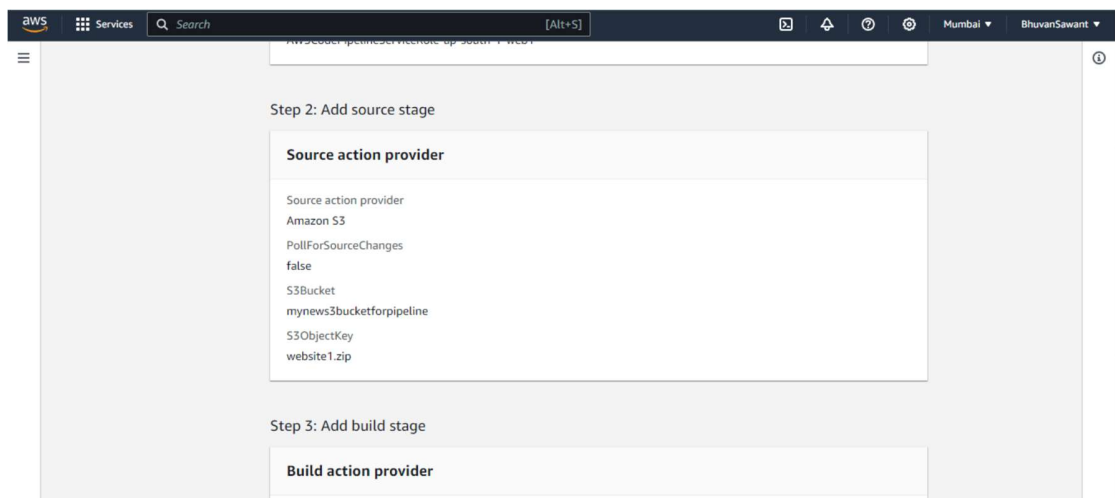
2. Integration with Other AWS Services:

- CodePipeline integrates with various AWS services, such as AWS CodeBuild for building applications, AWS CodeDeploy for automating deployments, and AWS Lambda for running custom actions.



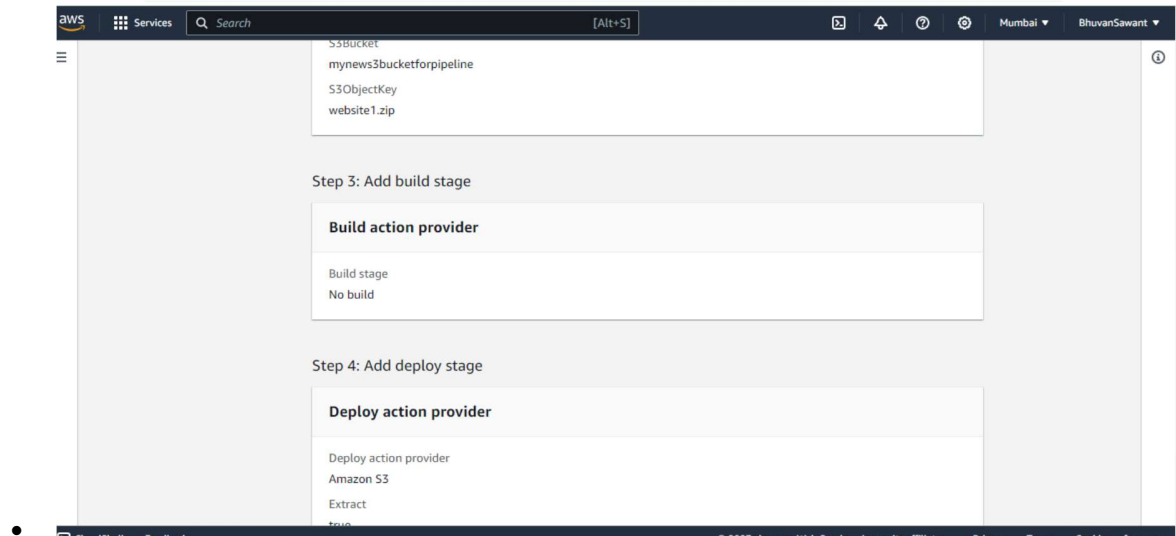
3. Pipeline Execution:

- Pipelines consist of a series of stages, and each stage can have one or more actions. Actions represent a task, such as source code retrieval or deployment to a specific environment.



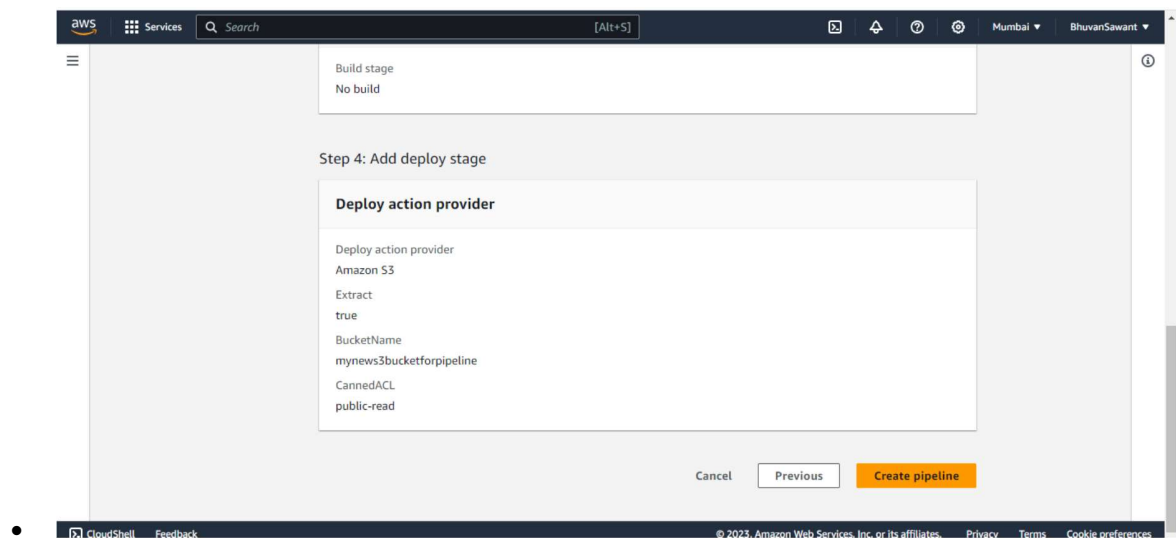
4. Source Providers:

- CodePipeline supports integration with various source code repositories, including AWS CodeCommit, GitHub, and Amazon S3.



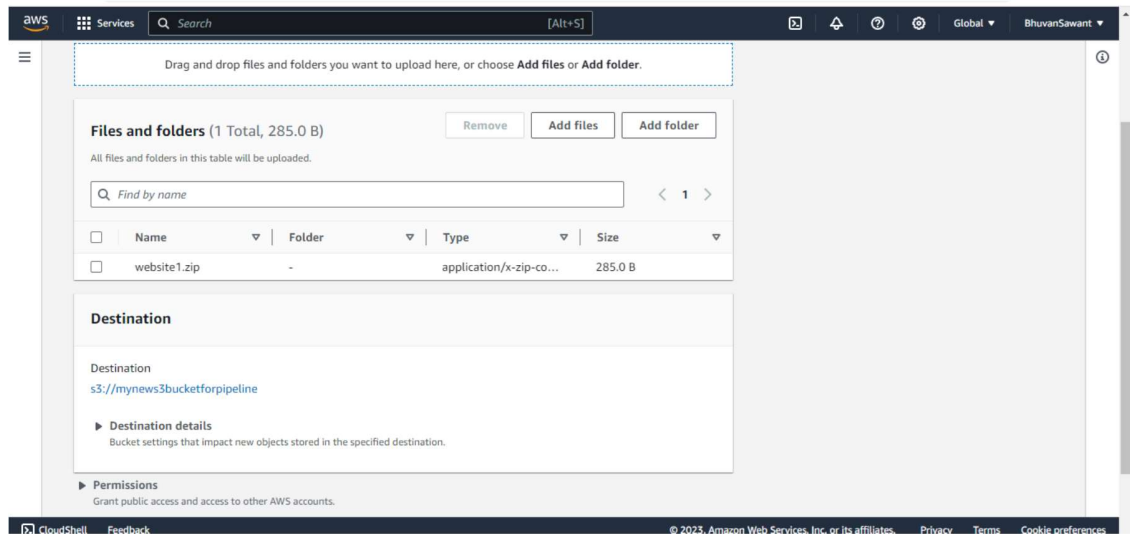
5. Artifact Management:

- CodePipeline uses artifacts to store the files and data needed for each action in a pipeline. Artifacts can be passed between stages to ensure consistency in the deployment process.



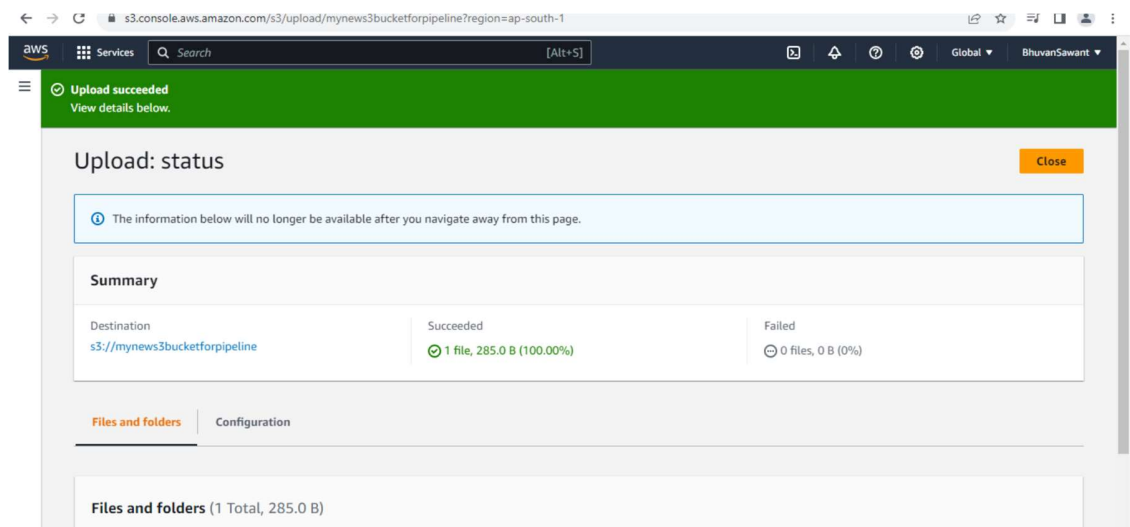
6. Integration with Third-Party Tools:

- Besides AWS services, CodePipeline supports integration with third-party tools. This is achieved through custom actions, which allow you to use external tools and scripts in your pipeline.



7. Pipeline Visualizations:

- CodePipeline provides a visual representation of your release process, making it easy to understand and monitor the status of each stage and action.



Key Concepts:

1. Pipeline:

- A pipeline is a series of stages that represents your release process. Each stage can contain one or more actions.

2. Stage:

- A stage is a logical unit in a pipeline, representing a phase in the release process. Stages are executed sequentially.

3. Action:

- An action represents a task within a stage. Actions can include tasks such as building code, deploying to a test environment, or running tests.

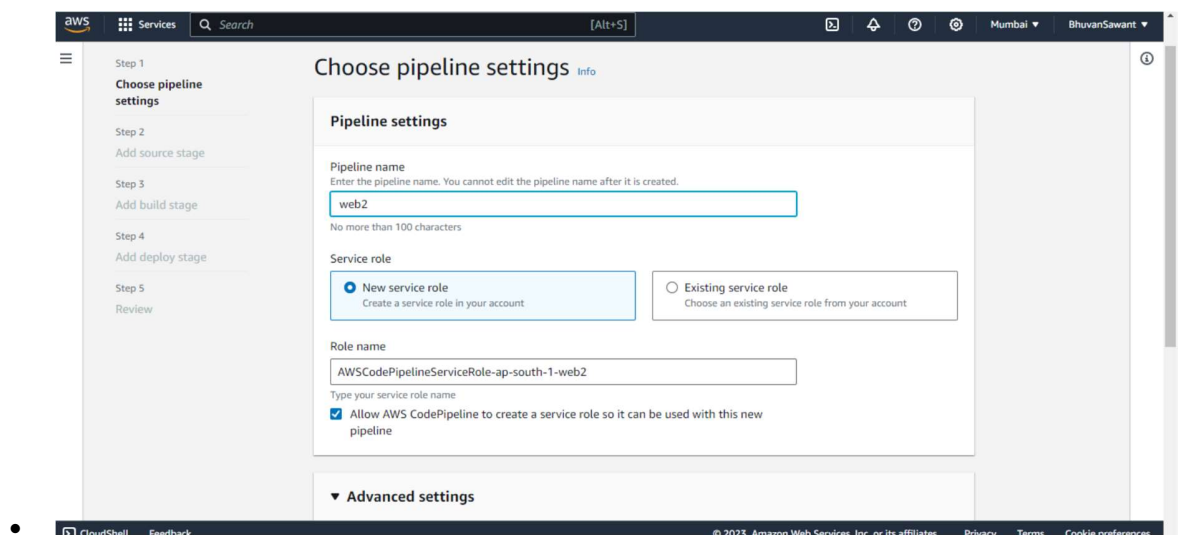
4. Artifact:

- Artifacts are the files and data that are produced as a result of an action. They are used to pass information between stages in a pipeline.

To deploy web application using CodePipeline here are the following steps to be followed:

1. Set Up Source Stage:

- Configure a source stage in AWS CodePipeline, linking to your version control system (e.g., CodeCommit, GitHub).



2. Configure Build Stage:

- Set up a build stage using AWS CodeBuild to compile, test, and package your web application.

aws Services Search [Alt+S] Mumbai BhuvanSawant

Type your service role name

☒ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

▼ Advanced settings

Artifact store

☒ Default location
Use the default artifact store (Amazon S3 codepipeline-ap-south-1-38972723336) designated in the same region and account as your pipeline

☐ Custom location
Choose an existing S3 location from your account in the same region and account as your pipeline

Encryption key

☒ Default AWS Managed Key
Use the AWS managed customer master key for CodePipeline in your account to encrypt the data in the artifact store.

☐ Customer Managed Key
To encrypt the data in the artifact store under an AWS KMS customer managed key, specify the key ID, key ARN, or alias ARN.

Cancel Next

3. Define Deployment Stage:

- Create a deployment stage using AWS CodeDeploy or another deployment provider to deploy your application to target environments.

aws Services Search [Alt+S] Mumbai BhuvanSawant

3 Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.

Deploy

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.

Amazon S3

Region
Asia Pacific (Mumbai)

Bucket
mynews3bucketforpipeline

Deployment path - optional

☒ Extract file before deploy
The deployed artifact will be unzipped before deployment.

▼ Additional configuration

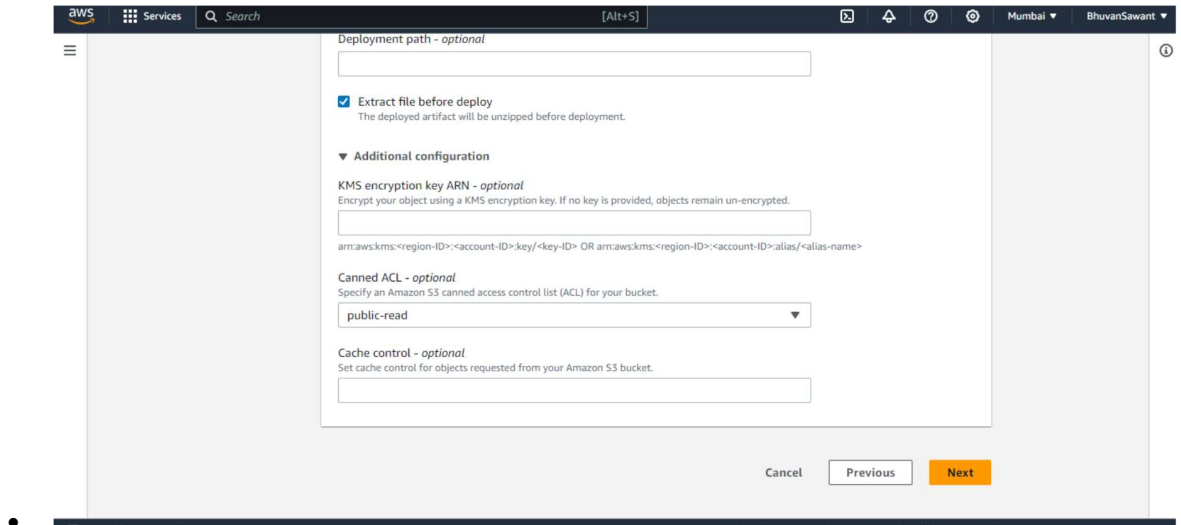
Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

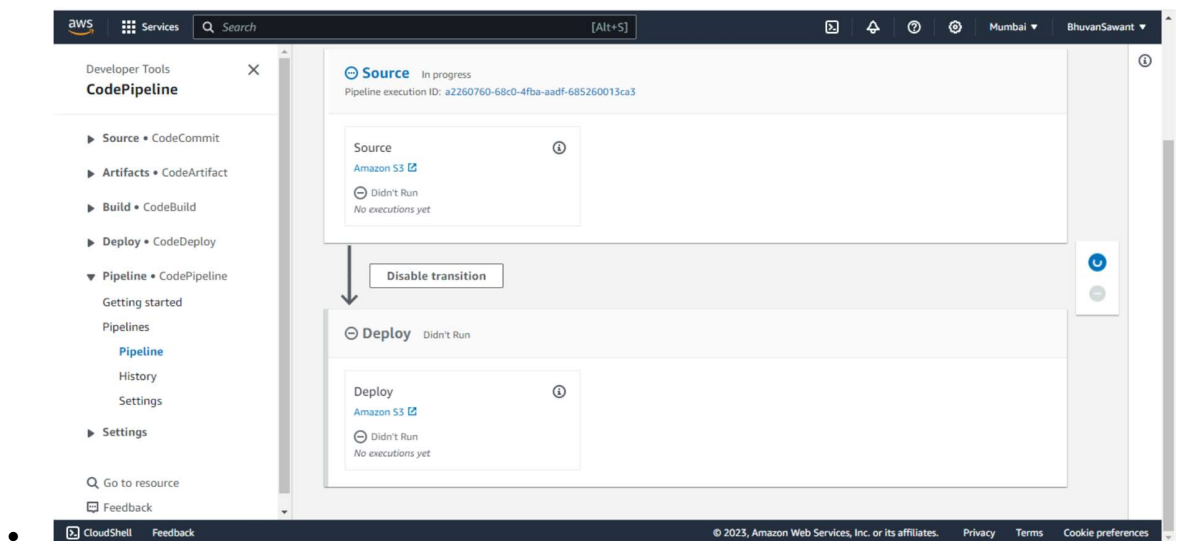
4. Configure Approval (Optional):

- Optionally, add a manual approval stage to review and approve deployments before proceeding to the next stage.



5. Artifact Passing:

- Ensure proper passing of artifacts between stages to maintain consistency in the deployment process.



6. Add Monitoring (Optional):

- Integrate monitoring tools (e.g., AWS CloudWatch) to track the performance and health of your application during and after deployment.

7. Configure Notifications (Optional):

- Set up notifications using AWS SNS or other services to receive alerts about pipeline events and status changes.

8. Test and Validate:

- Test the pipeline by triggering a build, ensuring that each stage executes successfully, and the application deploys as expected.

9. Modify Pipeline as Needed:

- Make adjustments to the pipeline configuration based on the specific requirements of your web application and deployment process.

10. Continuous Improvement:

- Implement continuous improvement practices, such as monitoring feedback, optimizing build and deployment scripts, and iterating on the pipeline structure.

Conclusion: By this assignment we learned how to host static web page through codepipeline.