Jai Talreja 61 D15A

# Cloud Deployment with Automation

## Introduction

This case study explores the concepts of AWS CodePipeline, EC2 and S3. In this we build a simple HTML WebApp using AWS CodeBuild, and automatically upload it to an S3 bucket to then automatically deploy it to an EC2 instance using CodeDeploy.

The tools and concepts used for this case study are :-

### AWS CodeBuild

**Key Features:**

Build Automation: Fully managed service that compiles source code, runs tests, and produces software packages.

Custom Build Environments: Supports Docker images for custom build environments.

Scalability: Automatically scales to meet build demand.

Pay-as-you-go Pricing: You only pay for the compute resources you use.

**Practical Uses:**

Continuous Integration (CI) for automated testing and building of applications.

Integrating with other AWS services for a seamless development pipeline.

Building and packaging applications for deployment.

### AWS CodePipeline

**Key Features:**

Continuous Delivery: Automates the software release process using defined workflows.

Integration with Other AWS Services: Works seamlessly with CodeBuild, CodeDeploy, and third-party tools.

Customizable Workflows: Easily define stages for building, testing, and deploying applications.

**Practical Uses:**

Automating the release process from code commit to deployment.

Creating pipelines for microservices or multi-environment setups.

Enabling rapid and reliable application delivery.

**Amazon S3 (Simple Storage Service)**

**Key Features:**

Scalable Storage: Virtually unlimited storage capacity.

Durability and Availability: Designed for 99.999999999% durability and high availability.

Security Features: Supports access control, encryption, and versioning.

**Practical Uses:**

Storing build artifacts and deployment packages.

Hosting static websites and serving assets for web applications.

Backup and archival storage.

**Amazon EC2 (Elastic Compute Cloud)**

**Key Features:**

Flexible Computing: Provides resizable compute capacity in the cloud.

Variety of Instance Types: Different instance types for various workloads.

Auto Scaling: Automatically adjusts capacity based on demand.

**Practical Uses:**

Hosting applications and services in a scalable manner.

Running batch processing and data analytics workloads.

Deploying web applications or back-end services.

**AWS CodeDeploy**

**Key Features:**

Automated Deployments: Automatically deploys applications to EC2, Lambda, or on-premises servers.

Blue/Green Deployments: Reduces downtime and risks during application updates.

Monitoring and Rollback: Monitors deployment status and can roll back if issues are detected.

**Practical Uses:**

Managing updates and deployments for applications running on EC2 instances.

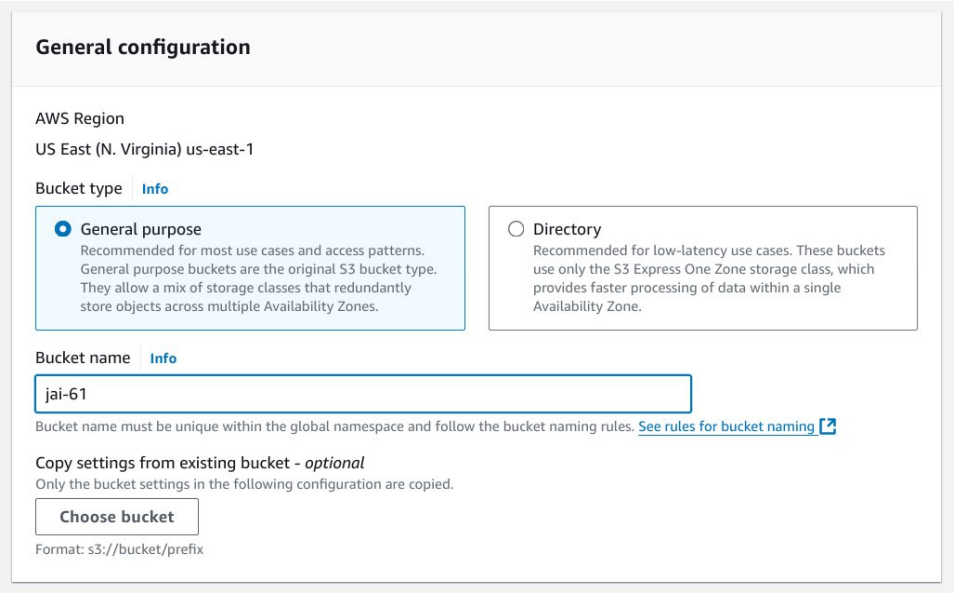
Facilitating microservices deployments with minimal downtime.

Integrating with CI/CD pipelines to automate the deployment process.

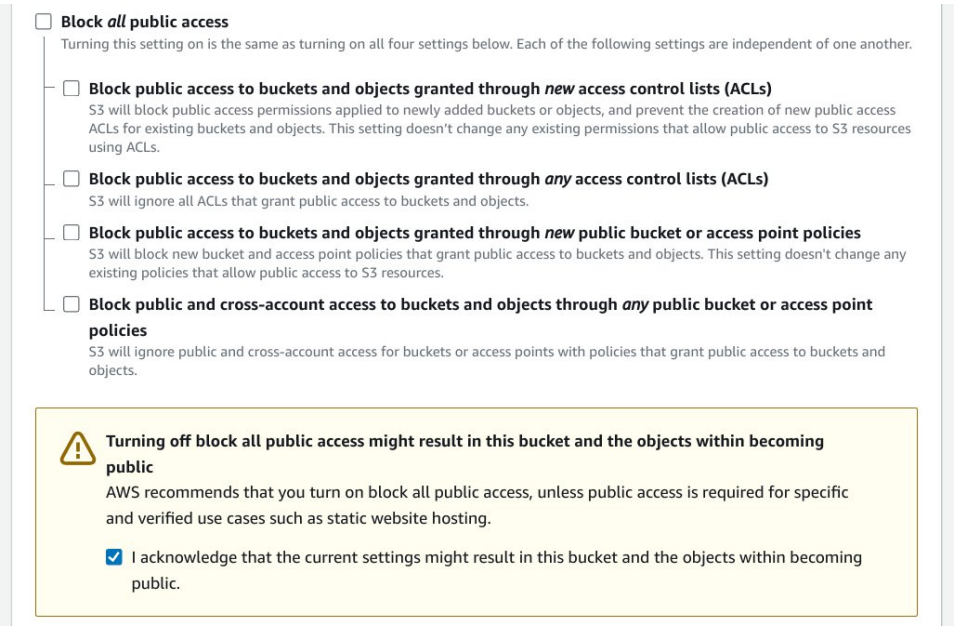
Jai Talreja 61 D15A

## Steps

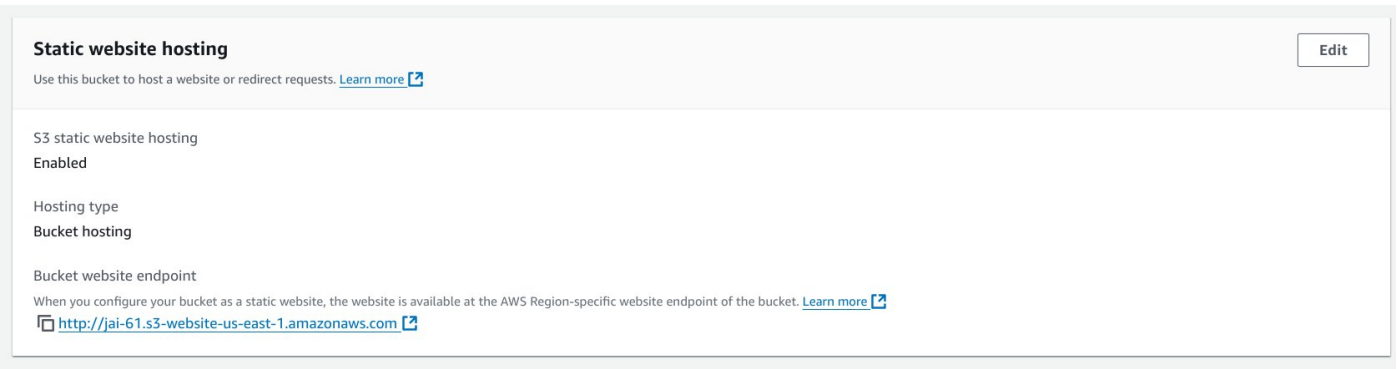
### Creating an S3 Bucket




### Allowing public access

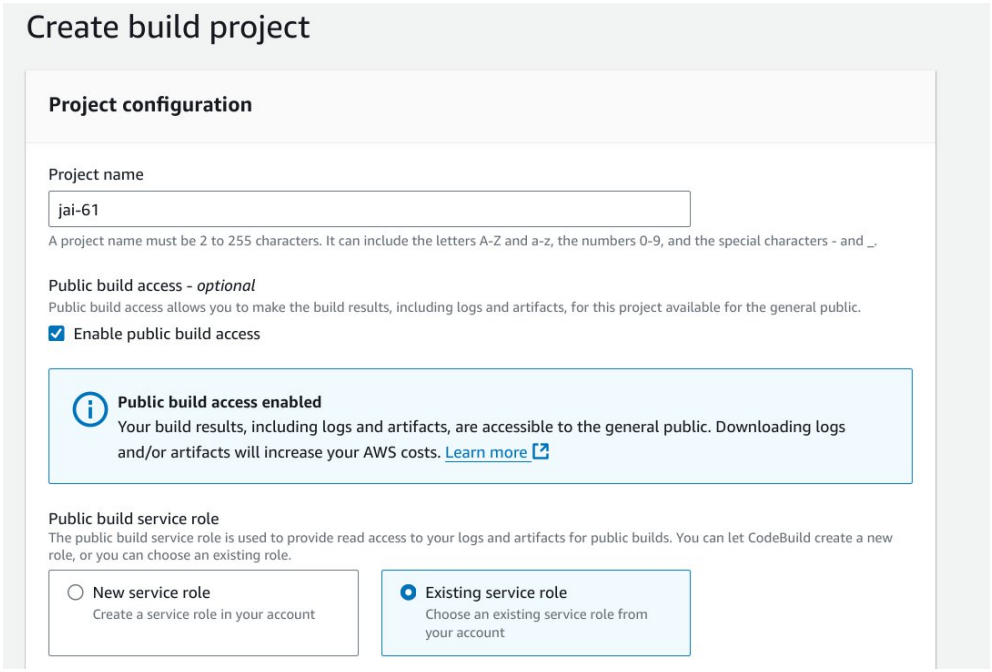



### Enabling static website hosting

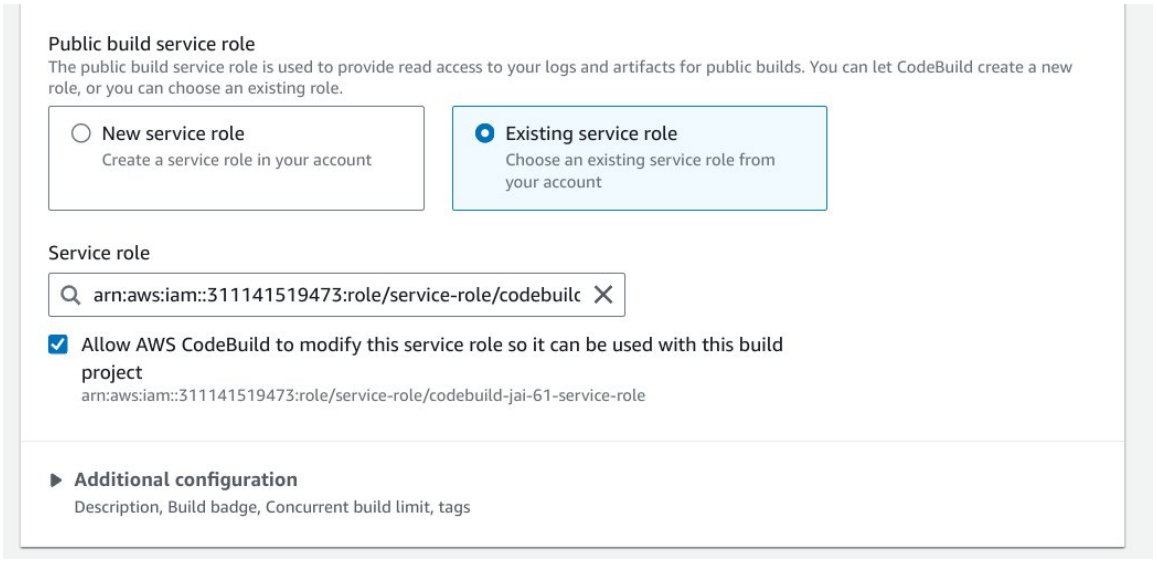


Jai Talreja 61 D15A

Creating a Build project

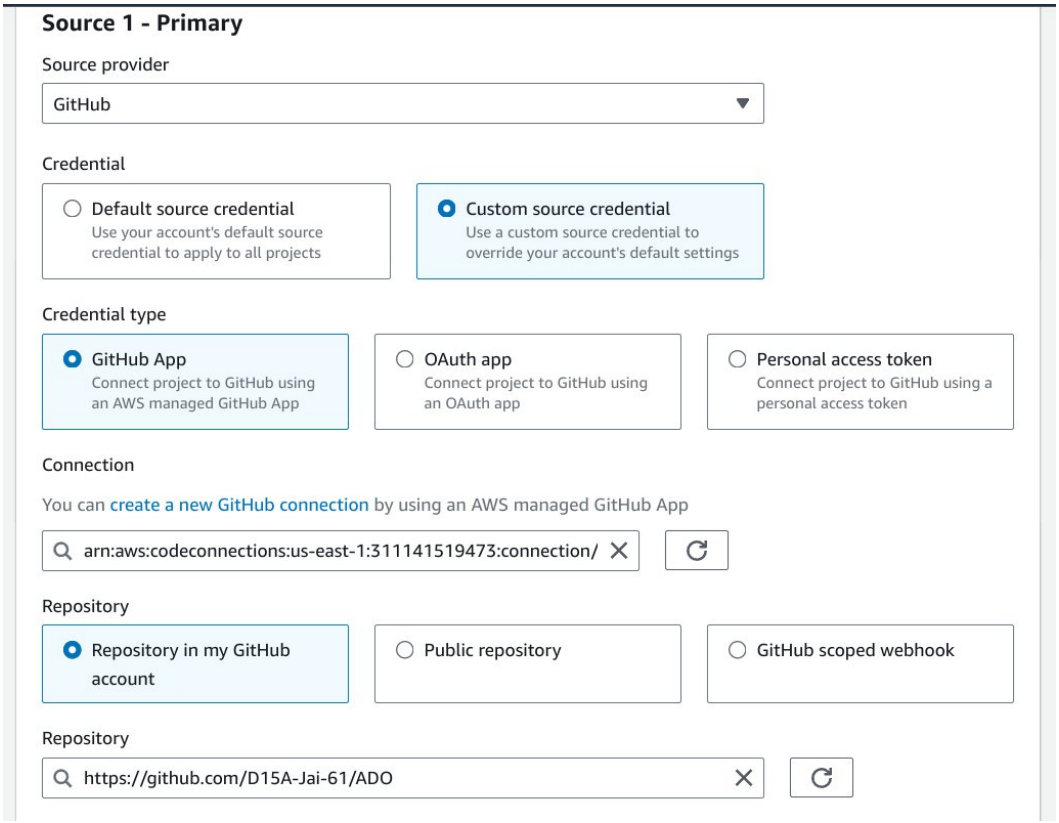



Using existing service role and allowing AWS to modify it

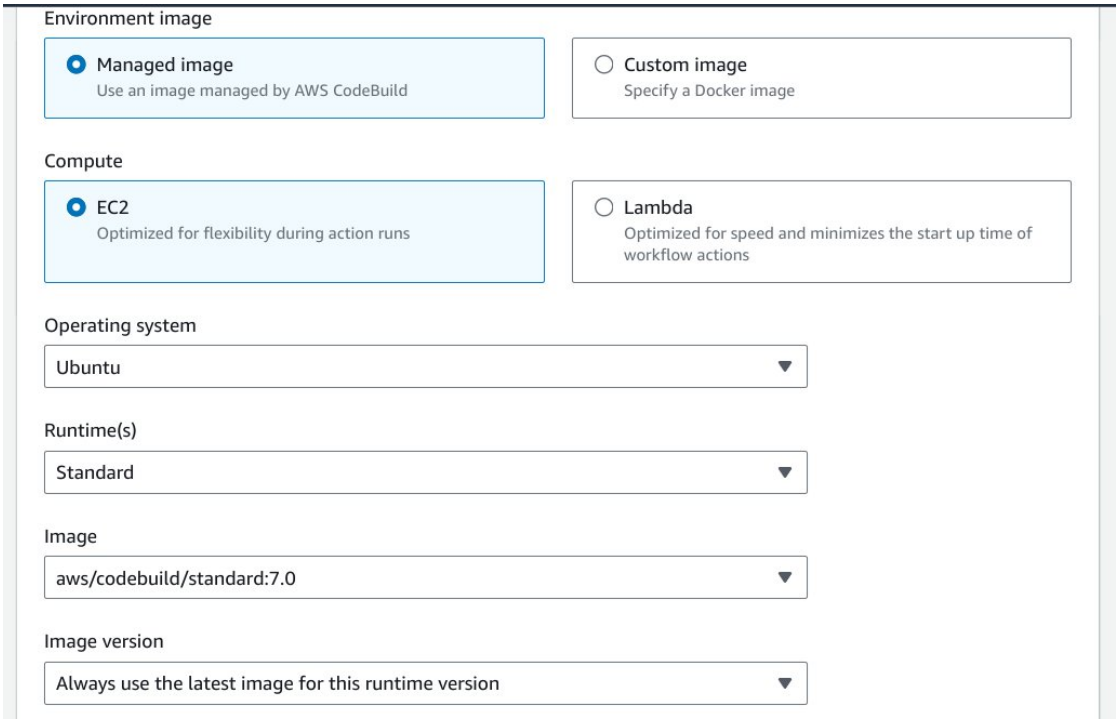


Jai Talreja 61 D15A

Configuring source for CodeBuild project as a GitHub repository

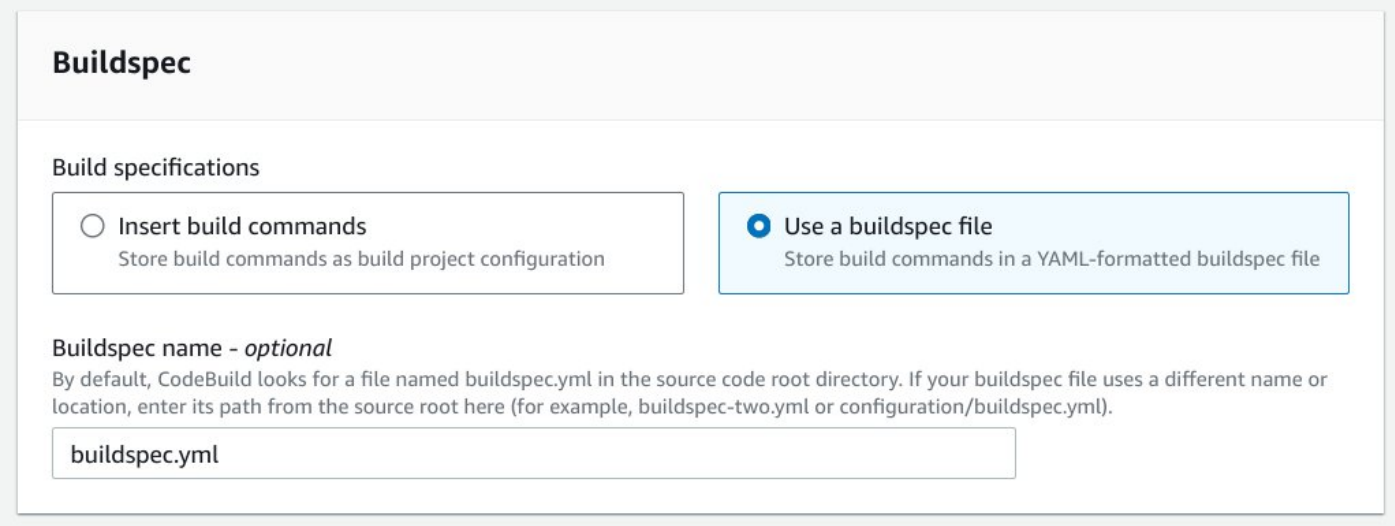



Using an Ubuntu instance to build the WebApp

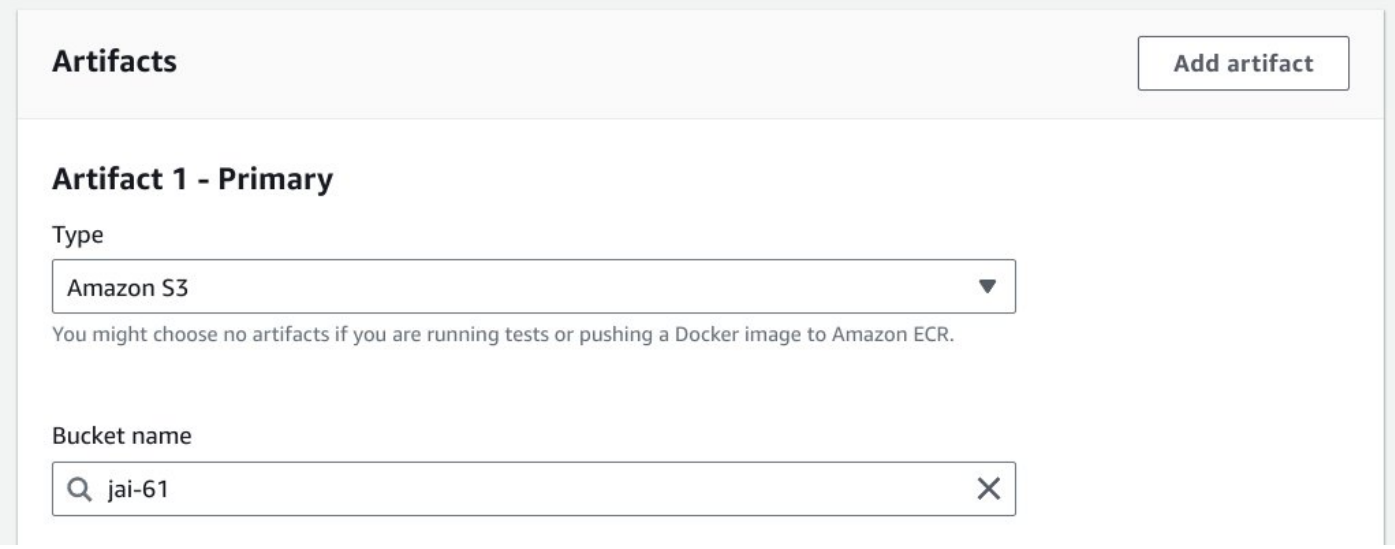


Jai Talreja 61 D15A

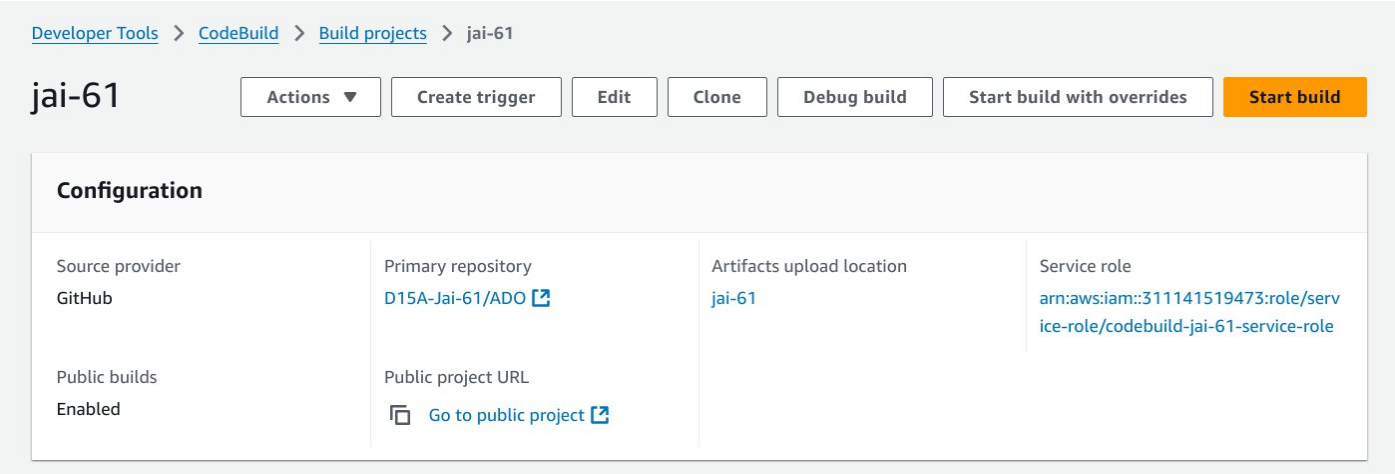
Using the BuildSpec file in the GitHub repository




Configuring the created S3 bucket to store the built project files



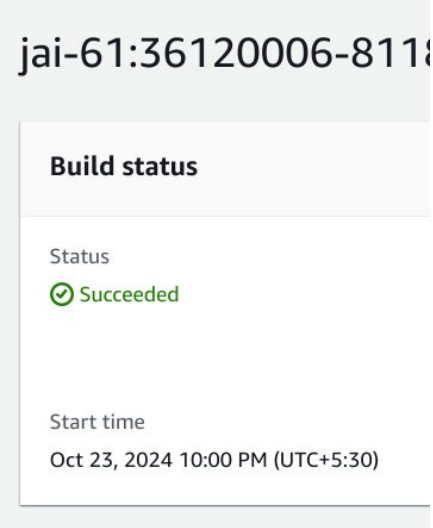

Build project created



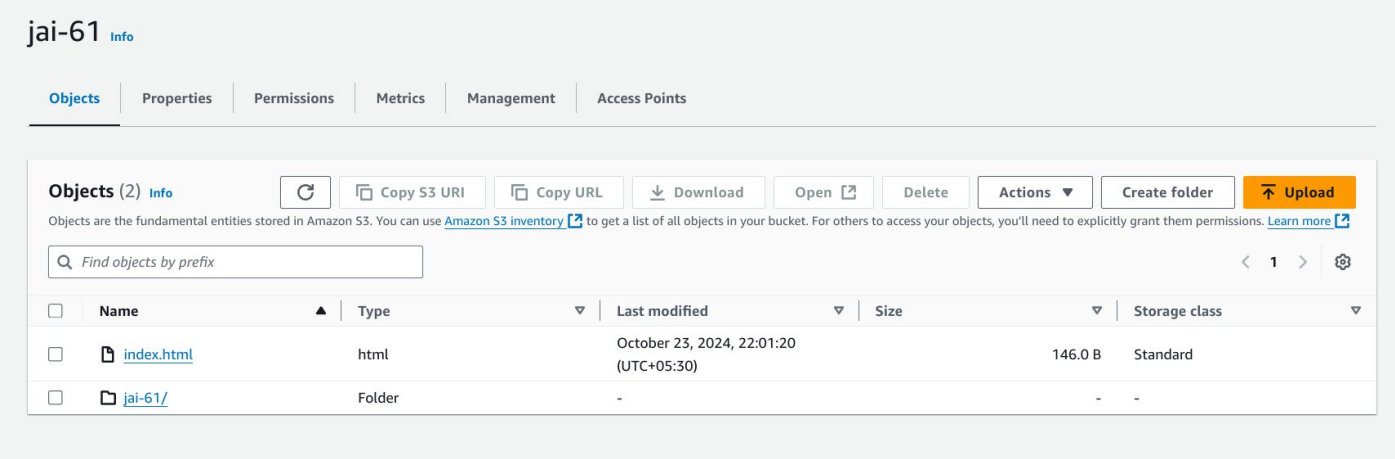
Jai Talreja 61 D15A

Build process succeeded and finished


Success

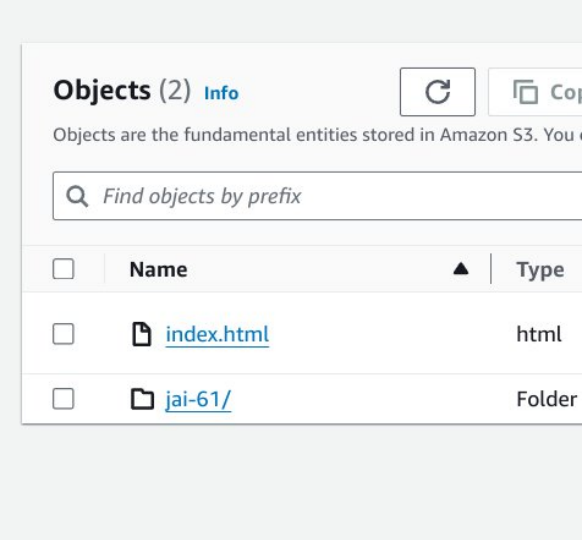



The built project files have been successfully uploaded to the created S3 bucket

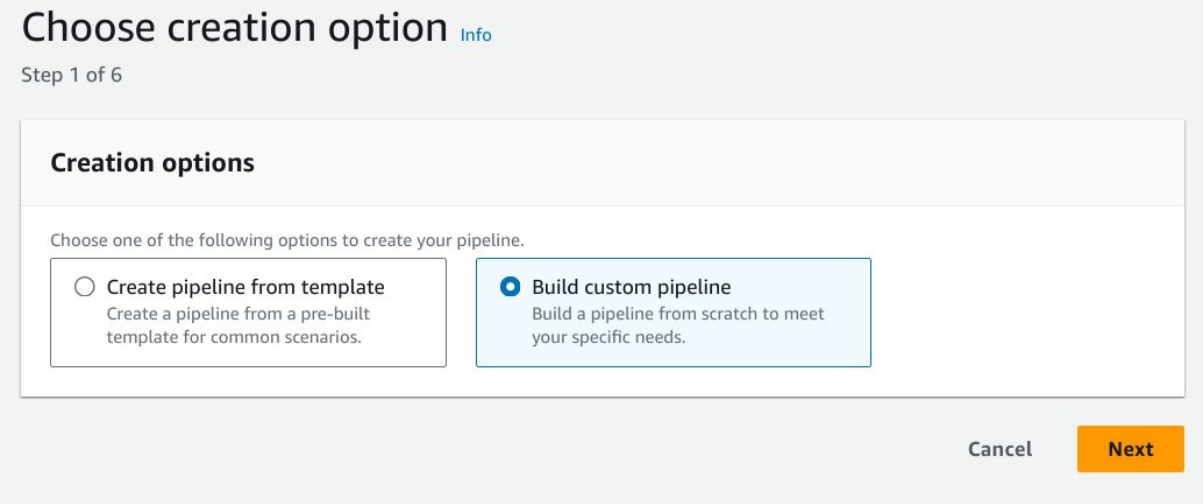


Jai Talreja 61 D15A

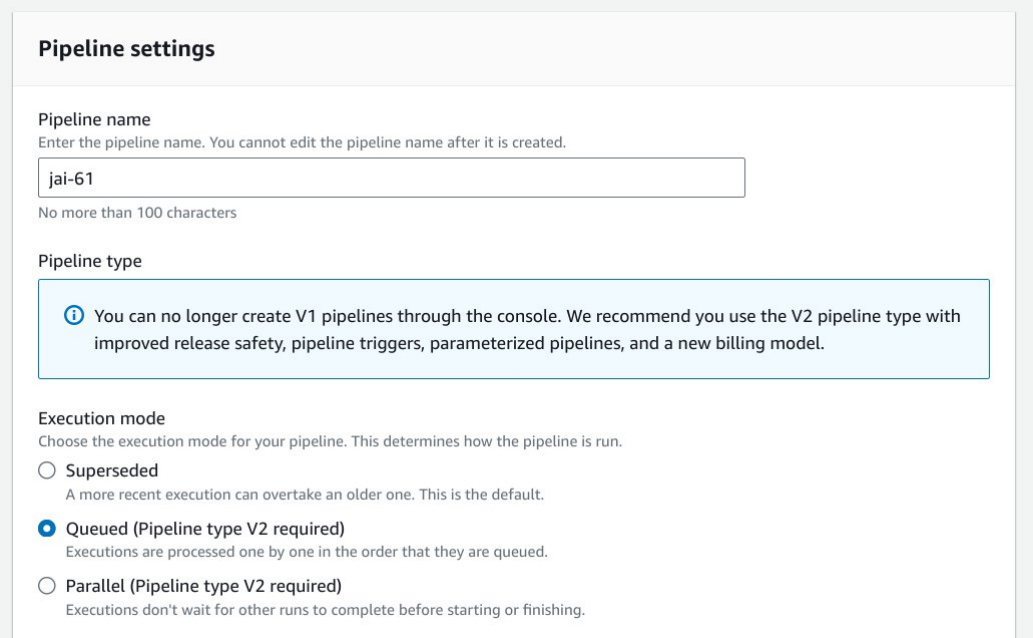
All built project files




Creating a custom Pipeline to deploy the built project

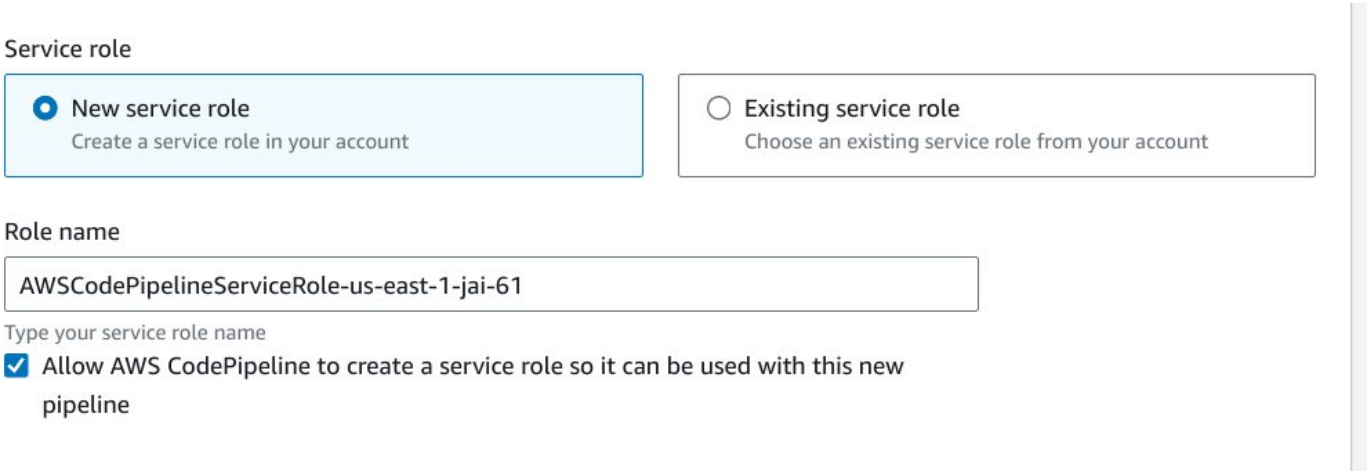



Pipeline settings

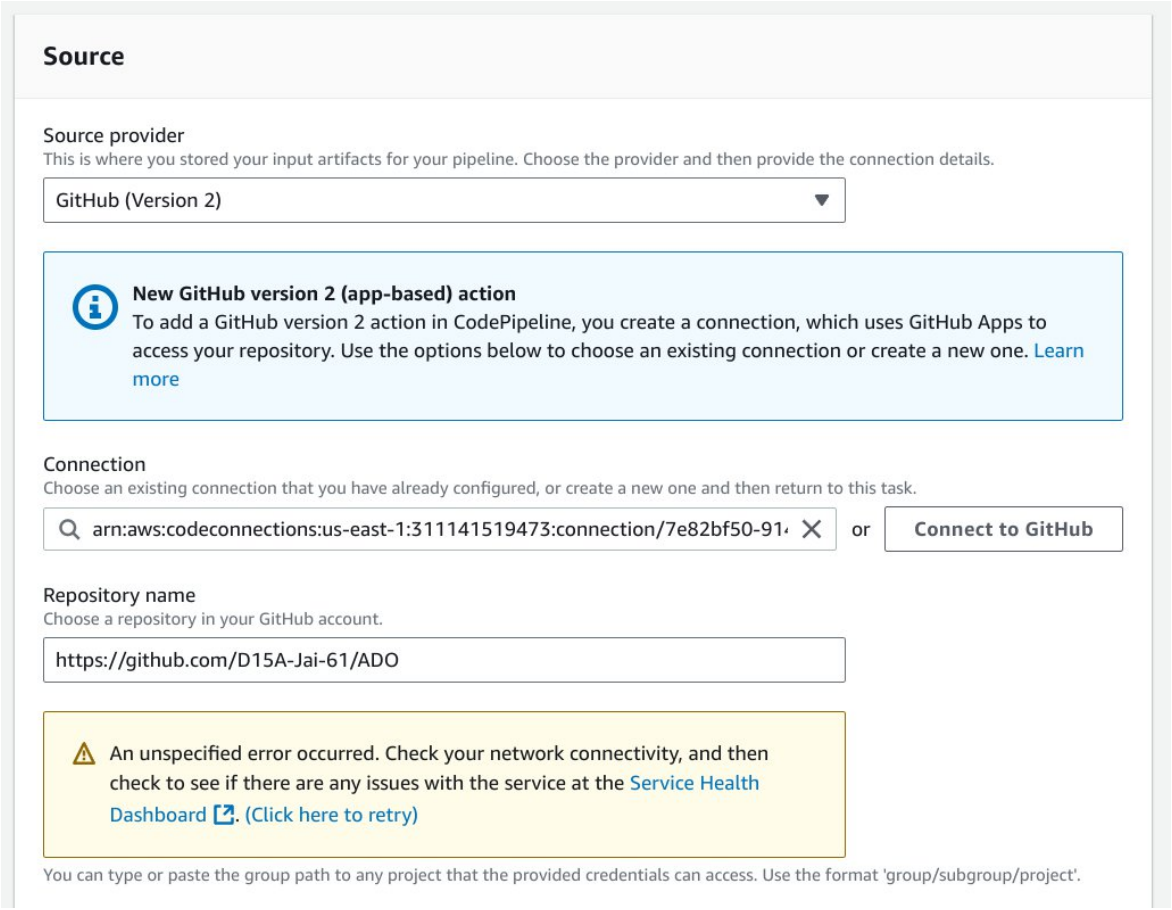


Jai Talreja 61 D15A

Using a new service role for the Pipeline

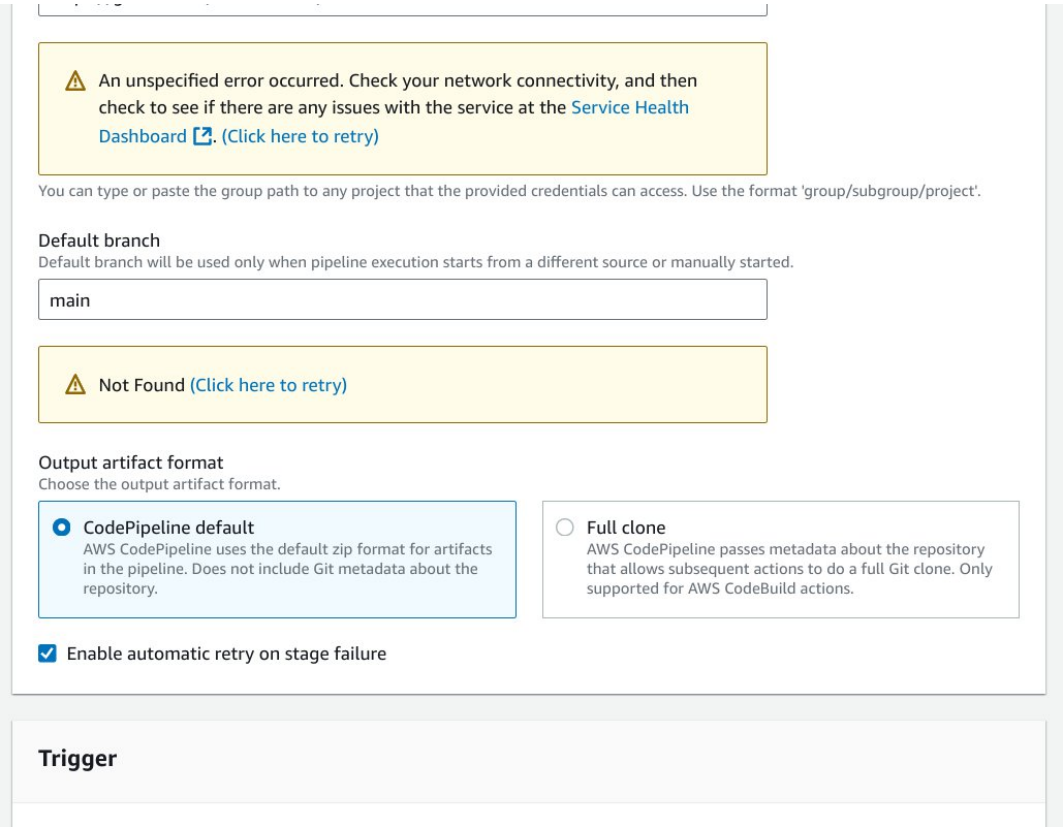



Configuring the source for the Pipeline as the same repository as the CodeBuild project (the issue here is, even though the connection ID and repository link along with the branch name have been specified correct, the service isn't able to find it)

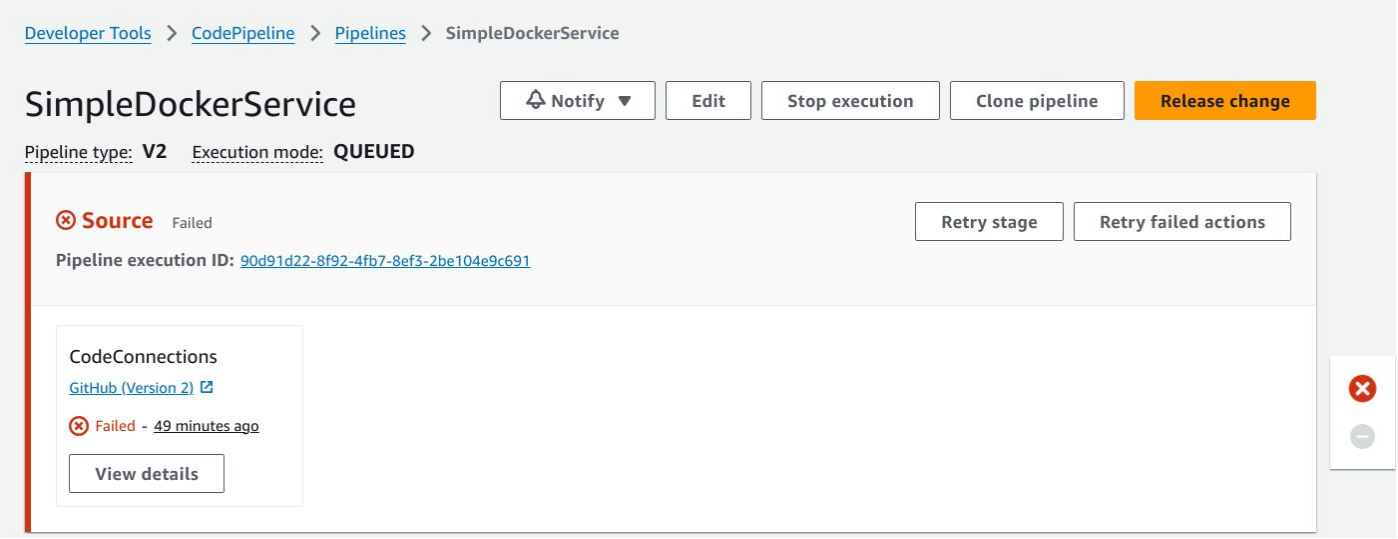


Jai Talreja 61 D15A

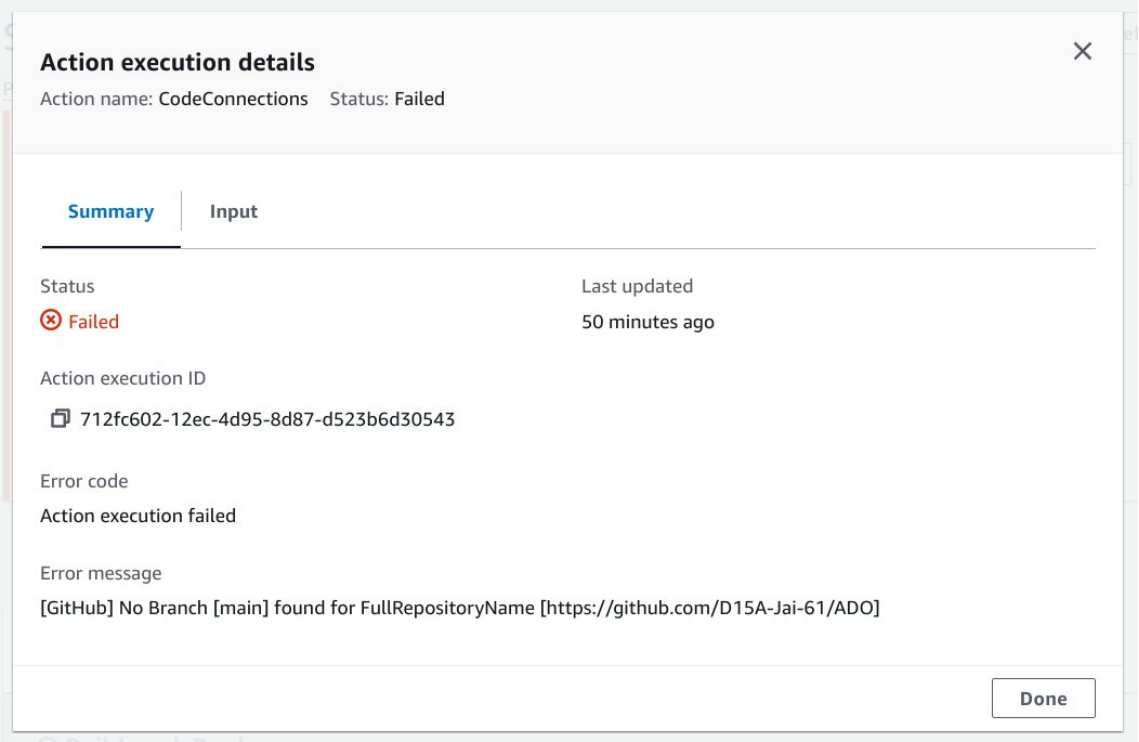
Unable to find the source even though correctly typed




Pipeline created but failed execution due to above mentioned issue



Execution details include the error message that specifies the above mentioned issue




## Conclusion

The case study performance was unsuccessful due to presumably a server side issue from AWS, not from performed steps being incorrect.

The project was built successfully and uploaded to S3 bucket automatically successfully, but not deployed to EC2 instance successfully.