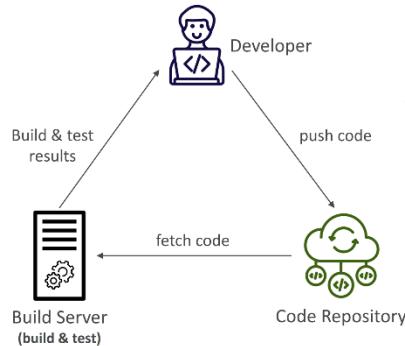


CICD – Introduction

- We have learned how to:
 - Create AWS resources, manually (fundamentals)
 - Interact with AWS programmatically (AWS CLI)
 - Deploy code to AWS using Elastic Beanstalk
- All these manual steps make it very likely for us to do mistakes!
- We would like our code “in a repository” and have it deployed onto AWS
 - Automatically
 - The right way
 - Making sure it’s tested before being deployed
 - With possibility to go into different stages (dev, test, staging, prod)
 - With manual approval where needed
- AWS CodeCommit – storing our code
- AWS CodePipeline – automating our pipeline from code to Elastic Beanstalk
- AWS CodeBuild – building and testing our code
- AWS CodeDeploy – deploying the code to EC2 instances (not Elastic Beanstalk)
- AWS CodeStar – manage software development activities in one place
- AWS CodeArtifact – store, publish, and share software packages
- AWS CodeGuru – automated code reviews using Machine Learning

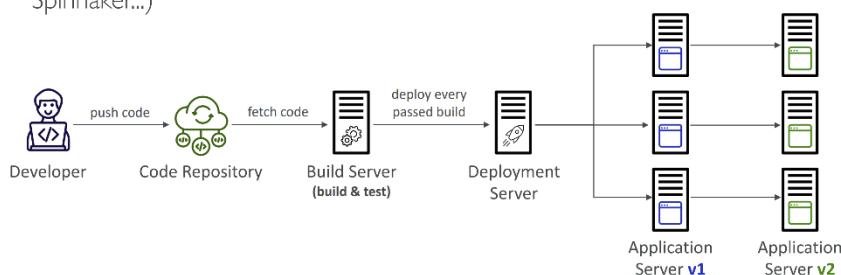
Continuous Integration (CI)

- Developers push the code to a code repository often (e.g., GitHub, CodeCommit, Bitbucket...)
- A testing / build server checks the code as soon as it’s pushed (CodeBuild, Jenkins CI...)
- The developer gets feedback about the tests and checks that have passed / failed
- Find bugs early, then fix bugs
- Deliver faster as the code is tested
- Deploy often
- Happier developers, as they’re unblocked

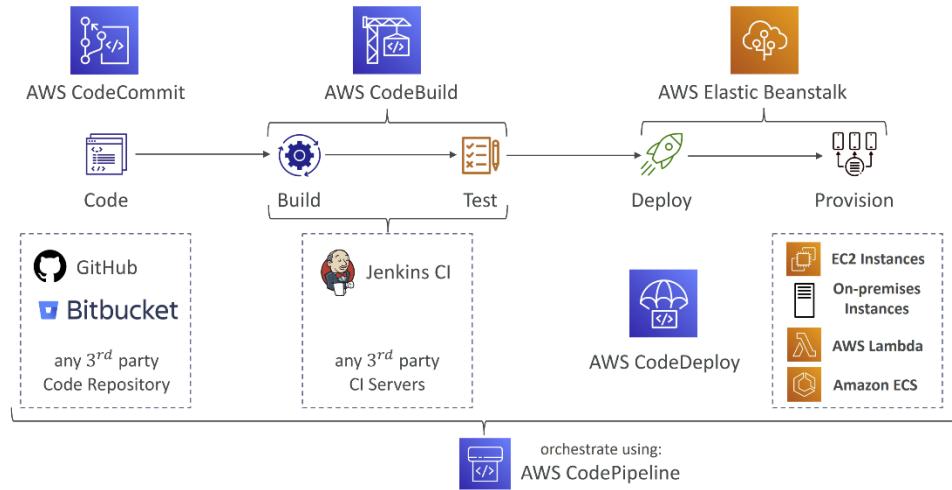


Continuous Delivery (CD)

- Ensures that the software can be released reliably whenever needed
- Ensures deployments happen often and are quick
- Shift away from “one release every 3 months” to “5 releases a day”
- That usually means automated deployment (e.g., CodeDeploy, Jenkins CD, Spinnaker...)



Technology Stack for CICD



Code Commit – brief summary:

AWS CodeCommit is a fully managed source control service provided by Amazon Web Services (AWS). It offers a secure and scalable platform for hosting private Git repositories and managing source code in the cloud.

Key Features of AWS CodeCommit:

1. **Git Repositories:** CodeCommit allows the creation and management of secure Git repositories for version control of source code, documents, and assets.
2. **Security and Access Control:** Offers fine-grained access controls using AWS Identity and Access Management (IAM) to manage repository access at various levels (repository, branch, file), ensuring security and compliance.
3. **Scalability and Performance:** CodeCommit is designed to handle large-scale repositories and high traffic, providing a scalable and reliable platform for storing and collaborating on code.
4. **Integration with DevOps Tools:** Seamlessly integrates with various DevOps tools and services, such as AWS CodeBuild, AWS CodePipeline, AWS CodeDeploy, Jenkins, and more, enabling continuous integration and deployment workflows.
5. **Encrypted Data:** Data in transit and at rest is encrypted to maintain security standards.
6. **Collaboration and Branching:** Supports collaborative development by allowing multiple developers to work on branches, merge code changes, and manage pull requests.
7. **Trigger Notifications and Events:** CodeCommit integrates with AWS CloudWatch Events to trigger notifications and automate workflows based on repository events (e.g., commits, pushes, pull requests).

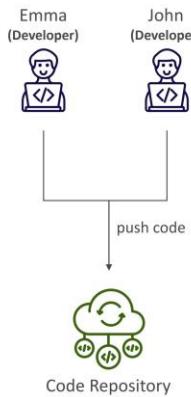
Use Cases for AWS CodeCommit:

1. **Version Control and Collaboration:** Provides a secure and scalable platform for version control and collaborative software development, especially for teams working on AWS.
2. **Continuous Integration and Deployment:** Integrates with CI/CD pipelines for automating build, test, and deployment workflows.

3. **Secure Code Management:** Suitable for organizations requiring secure and compliant source code management, particularly those handling sensitive or proprietary code.
4. **Scalable Repositories:** Offers scalability for managing large repositories and accommodating increased team sizes or codebase expansion.
5. **Integrating with AWS Services:** Enables seamless integration with other AWS services for building end-to-end development and deployment pipelines within the AWS ecosystem.

AWS CodeCommit is a viable option for teams or organizations seeking a secure, managed, and scalable Git-based source control service hosted on AWS. It facilitates collaborative software development, integrates with various DevOps tools, and ensures compliance and security in managing source code.

- Version control is the ability to understand the various changes that happened to the code over time (and possibly roll back)
- All these are enabled by using a version control system such as Git
- A Git repository can be synchronized on your computer; but it usually is uploaded on a central online repository
- Benefits are:
 - Collaborate with other developers
 - Make sure the code is backed-up somewhere
 - Make sure it's fully viewable and auditable
- Git repositories can be expensive
- The industry includes GitHub, GitLab, Bitbucket...
- And **AWS CodeCommit:**
 - Private Git repositories
 - No size limit on repositories (scale seamlessly)
 - Fully managed, highly available
 - Code only in AWS Cloud account => increased security and compliance
 - Security (encrypted, access control...)
 - Integrated with Jenkins, AWS CodeBuild, and other CI tools

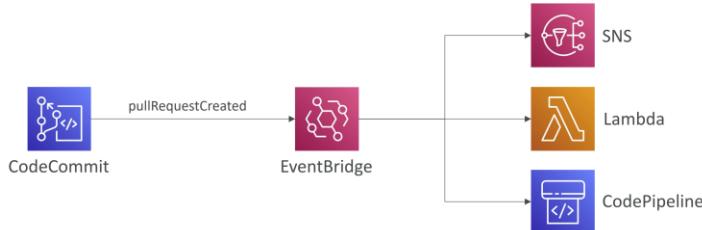


CodeCommit vs GitHub

	CodeCommit	GitHub
Support Code Review (Pull Requests)	✓	✓
Integration with AWS CodeBuild	✓	✓
Authentication (SSH & HTTPS)	✓	✓
Security	IAM Users & Roles	GitHub Users
Hosting	Managed & hosted by AWS	- Hosted by GitHub - GitHub Enterprise: self hosted on your servers
UI	Minimal	Fully Featured

CodeCommit – Monitoring with EventBridge

- You can monitor CodeCommit events in EventBridge (near real-time)
- pullRequestCreated, pullRequestStatusChanged, referenceCreated, commentOnCommitCreated...



Lab: CodeCommit

Create a repository.

The screenshot shows two windows. The top window is a search results page for 'code' in the AWS CloudTrail interface, listing services like SNS, Lambda, and CodePipeline. The bottom window is the 'Create AWS CodeCommit repo' page, which includes a 'Create repository' button.

Developer Tools > CodeCommit > Repositories > Create repository

Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

Repository settings

Repository name: my-nodejs-app

Description - optional:

Tags: Add tag

Enable Amazon CodeGuru Reviewer for Java and Python - optional:

Get recommendations to improve the quality of the Java and Python code for all pull requests in this repository. A service-linked role will be created in IAM on your behalf if it does not exist.

Create

Success: Repository successfully created

Clone URL: https://git-codecommit.eu-north-1.amazonaws.com/v1/repos/my-nodejs-app

if you are login as a root then you will see the SSH is disabled

Now upload a file
scroll down

Developer Tools > CodeCommit

Source > CodeCommit

Code:

Repositories

Code

Pull requests

Commits

Branches

Git tags

Settings

Approval rule templates

Artifacts > CodeArtifact

Build > CodeBuild

Deploy > CodeDeploy

Pipeline > CodePipeline

Settings

Go to resource

Feedback

Step 1: Prerequisites
You must use a Git client that supports Git version 1.7.9 or later to connect to an AWS CodeCommit repository. If you do not have a Git client, you can install one from Git downloads. View Git downloads page [\[?\]](#)

You must have an AWS CodeCommit managed policy attached to your IAM user, belong to a CodeStar project team, or have the equivalent permissions. Learn how to create and configure an IAM user for accessing AWS CodeCommit. [\[?\]](#) Learn how to add team members to an AWS CodeStar Project. [\[?\]](#)

Step 2: Git credentials
Create Git credentials for your IAM user, if you do not already have them. Download the credentials and save them in a secure location. Generate Git Credentials [\[?\]](#)

Step 3: Clone the repository
Clone your repository to your local computer and start working on code. Run the following command:
git clone https://git-codecommit.eu-north-1.amazonaws.com/v1/repos/my-nodejs-app

Additional details
You can find more detailed instructions in the documentation. [View documentation](#) [\[?\]](#)

my-nodejs-app Info

Name:

Add file Create file Upload file

Empty repository
Your repository is currently empty. You can add files to it directly from the console or by cloning the repository to your local computer, creating commits, and pushing content to the remote repository in AWS CodeCommit.

Now upload a file

Developer Tools > CodeCommit > Repositories > my-nodejs-app > File

Upload a file

my-nodejs-app [info]

Name	Size	Actions
index.html	4 KB	Remove file

Commit changes to main
For: my-nodejs-app/index.html

Author name: rajiv

Email address:

Commit message - optional
A default commit message will be used if you do not provide one.
first upload

Cancel [Commit changes](#)

Clone the repository

Identity and Access Management (IAM)

IAM > Users > rajiv

rajiv [info] Delete

Summary

ARN	Console access	Access key 1
arn:aws:iam::999838272208:user/rajiv	Enabled with MFA	AKIA6RSYBILIBDFZ7F - Inactive Used 151 days ago, 151 days old.
Created May 01, 2023, 21:59 (UTC+06:00)	Last console sign-in 3 hours ago	Access key 2 AKIA6RSYBILICFK5CF4K - Active Used 5 days ago, 78 days old.

Permissions Groups (1) Tags (4) Security credentials Access Advisor

Console sign-in now scroll down add https git credentials Manage console access

Identity and Access Management (IAM)

HTTPS Git credentials for AWS CodeCommit [0] Actions Generate credentials

Generate a user name and password you can use to authenticate HTTPS connections to AWS CodeCommit repositories. You can have a maximum of 2 sets of credentials (active or inactive) at a time. [Learn more](#)

User name	Created	Status
No credentials		Generate credentials

Now download and keep this user name password

Now go back to git commit page

Developer Tools > CodeCommit > Repositories > my-nodejs

my-nodejs [info]

Reference ▾ Notify ▾ main Create pull request Clone URL ▾

Clone HTTPS (highlighted)

Clone SSH

Clone HTTPS (GRC)

Connection steps

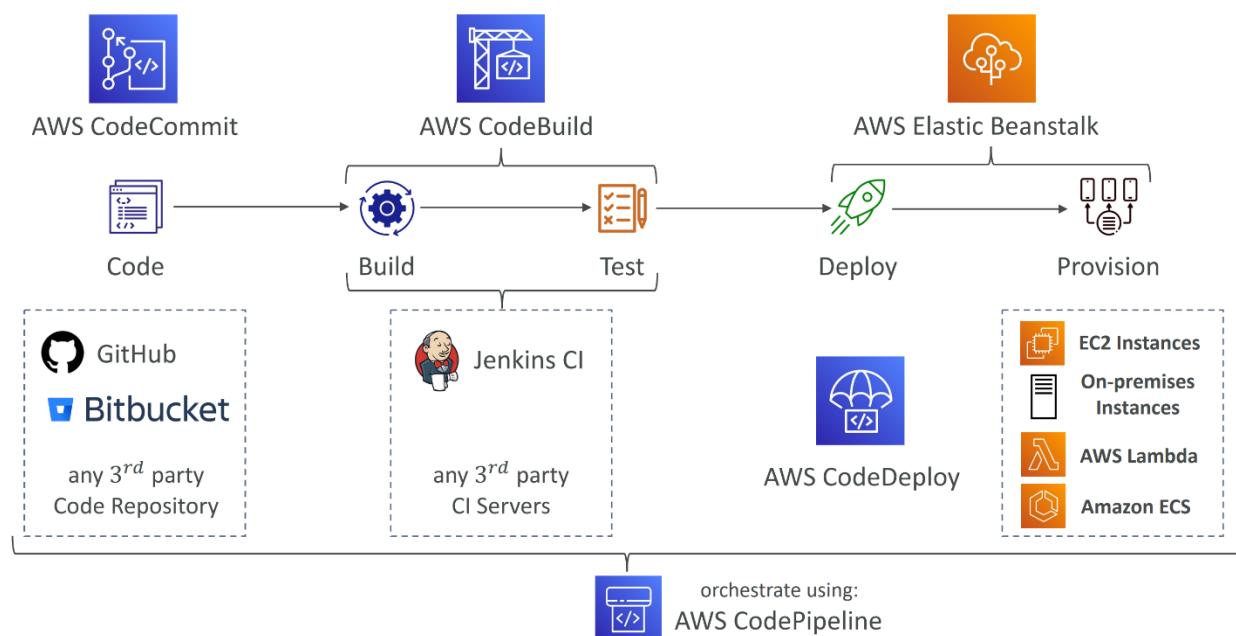
Now open git bash and clone the code

```
git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/my-node-js
```

Codepipeline

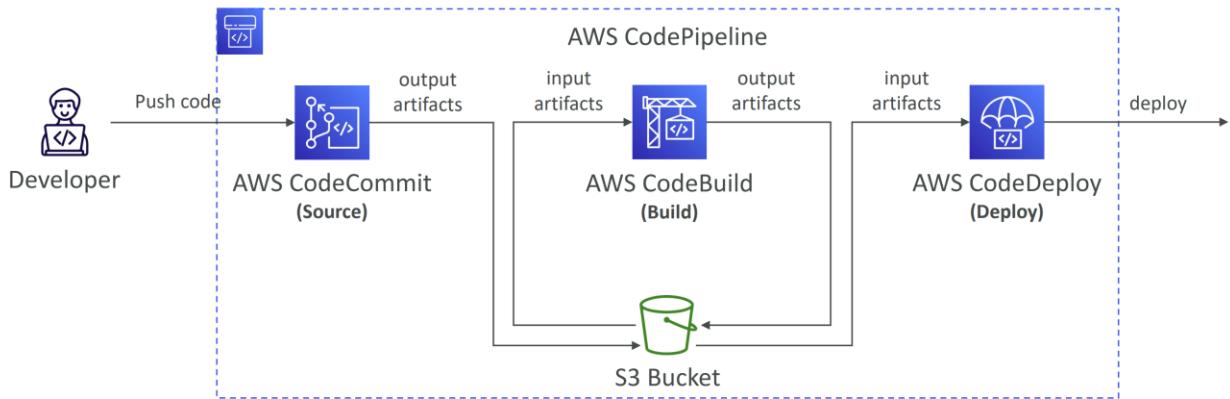
- Visual Workflow to orchestrate your CICD
- Source – CodeCommit, ECR, S3, Bitbucket, GitHub
- Build – CodeBuild, Jenkins, CloudBees, TeamCity
- Test – CodeBuild, AWS Device Farm, 3rd party tools, ...
- Deploy – CodeDeploy, Elastic Beanstalk, CloudFormation, ECS, S3, ...
- Invoke – Lambda, Step Functions
- Consists of stages:
 - Each stage can have sequential actions and/or parallel actions
 - Example: Build → Test → Deploy → Load Testing → ...
 - Manual approval can be defined at any stage

Technology Stack for CICD



CodePipeline – Artifacts

- Each pipeline stage can create artifacts
- Artifacts stored in an S3 bucket and passed on to the next stage

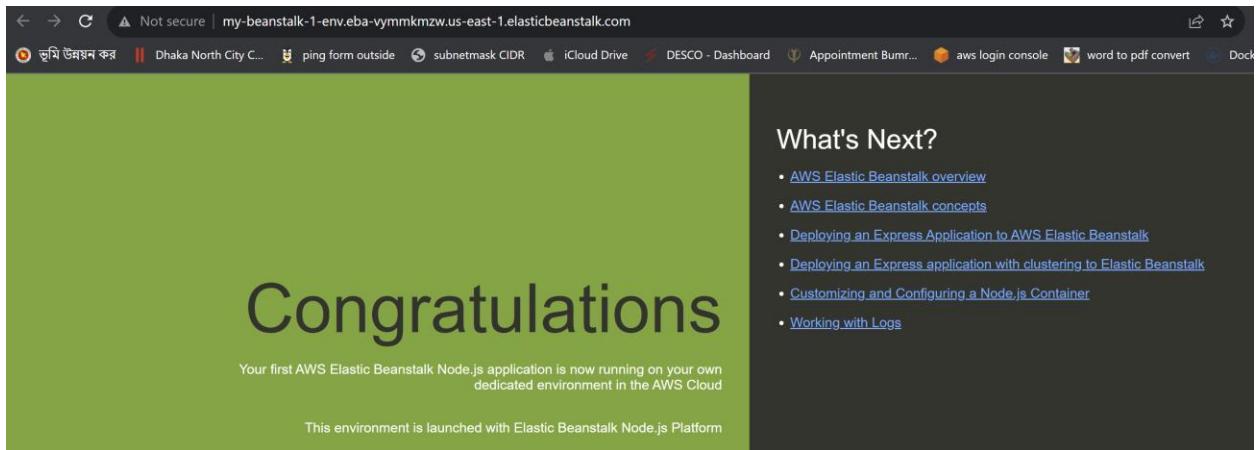


CodePipeline – Troubleshooting

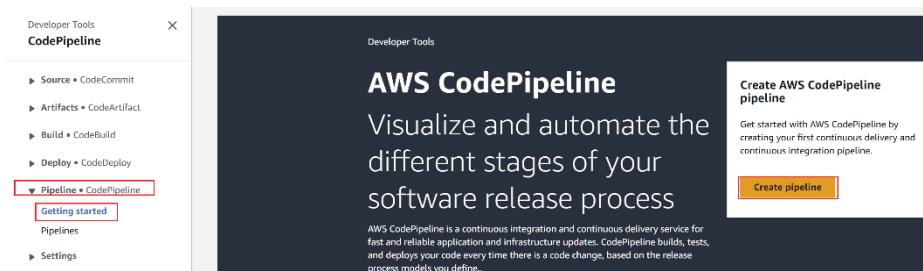
- For CodePipeline Pipeline/Action/Stage Execution State Changes
- Use **CloudWatch Events (Amazon EventBridge)**. Example:
 - You can create events for failed pipelines
 - You can create events for cancelled stages
- If CodePipeline fails a stage, your pipeline stops, and you can get information in the console
- If pipeline can't perform an action, make sure the "IAM Service Role" attached does have enough IAM permissions (IAM Policy)
- AWS CloudTrail can be used to audit AWS API calls

Lab: code pipeline

Create 2 beanstalk one is for environment, and another is for production.
and the DNS of beanstalk



Now create a codepipeline



The screenshot shows the "Choose pipeline settings" step of the AWS CodePipeline creation wizard. The left sidebar shows steps: Step 1 (Choose pipeline settings), Step 2 (Add source stage), Step 3 (Add build stage), Step 4 (Add deploy stage), Step 5 (Review). The main area is titled "Pipeline settings" and "Step 1 of 5". It includes fields for "Pipeline name" (set to "pipeline-1"), "Pipeline type" (radio buttons for V1 and V2, V1 selected), "Service role" (radio buttons for New service role and Existing service role, New service role selected), "Role name" (text input set to "AWSCodePipelineServiceRole-us-east-1-pipeline-1"), and a checkbox for "Allow AWS CodePipeline to create a service role so it can be used with this new pipeline".

Variables

You can add variables at the pipeline level. You can choose to assign the value when you start the pipeline. Choosing this option requires pipeline type V2. [Learn more](#)

No variables defined at the pipeline level in this pipeline.

[Add variable](#)

You can add up to 50 variables.

i The first pipeline execution will fail if variables have no default values.

▼ Advanced settings

Artifact store

Default location
Create a default S3 bucket in your account.

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

Click next

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1 Choose pipeline settings

Step 2 Add source stage Info

Step 3 Add build stage

Step 4 Add deploy stage

Step 5 Review

Add source stage

Source

Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

AWS CodeCommit

Repository name
Choose a repository that you have already created where you have pushed your source code.

my-node-js X

Branch name
Choose a branch of the repository

main X

Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

Amazon CloudWatch Events (recommended)
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs

AWS CodePipeline
Use AWS CodePipeline to check periodically for changes

Output artifact format
Choose the output artifact format.

CodePipeline default
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

Full clone
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

[Cancel](#)
Next

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Add build stage Info

Step 1 Choose pipeline settings
 Step 2 Add source stage
 Step 3 **Add build stage**
 Step 4 Add deploy stage
 Step 5 Review

This time we skip build stage

Build - optional

Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.

Cancel Previous Skip build stage Next

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Add deploy stage Info

Step 1 Choose pipeline settings
 Step 2 Add source stage
 Step 3 **Add build stage**
 Step 4 **Add deploy stage**
 Step 5 Review

You cannot skip this stage
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.
AWS Elastic Beanstalk

Region
US East (N. Virginia)

Application name
Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.
Q my-beanstalk-1 X

Environment name
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.
Q My-beanstalk-1-env X

Cancel Previous Next

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Review Info

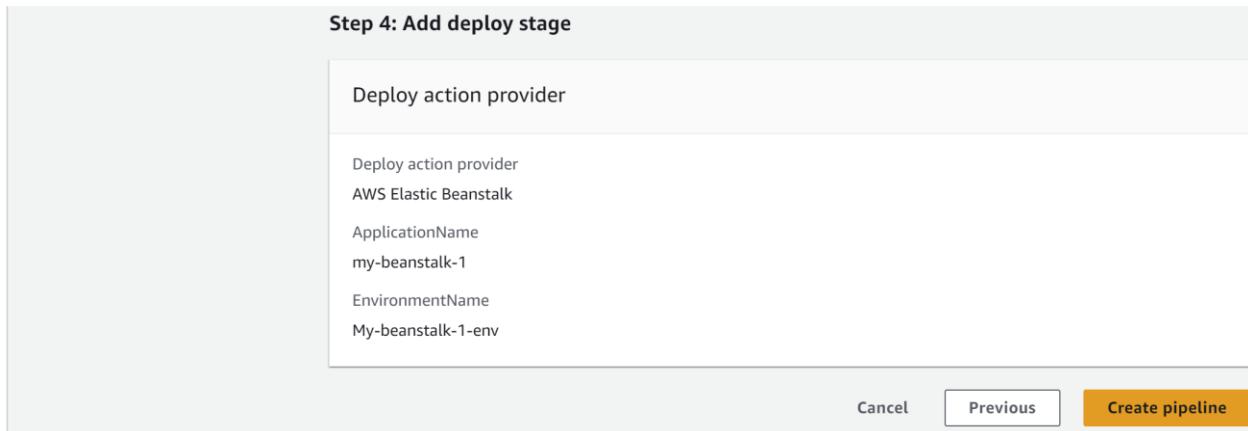
Step 1 Choose pipeline settings
 Step 2 Add source stage
 Step 3 Add build stage
 Step 4 Add deploy stage
 Step 5 **Review**

Step 1: Choose pipeline settings

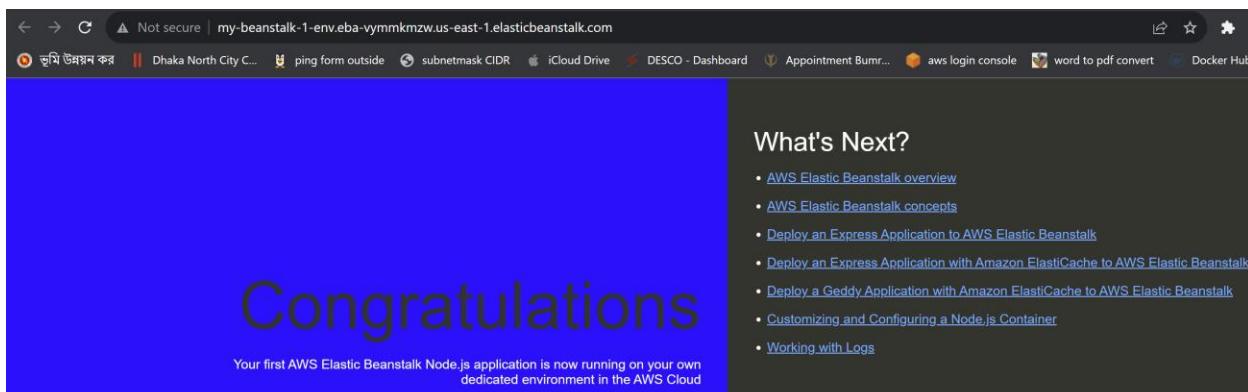
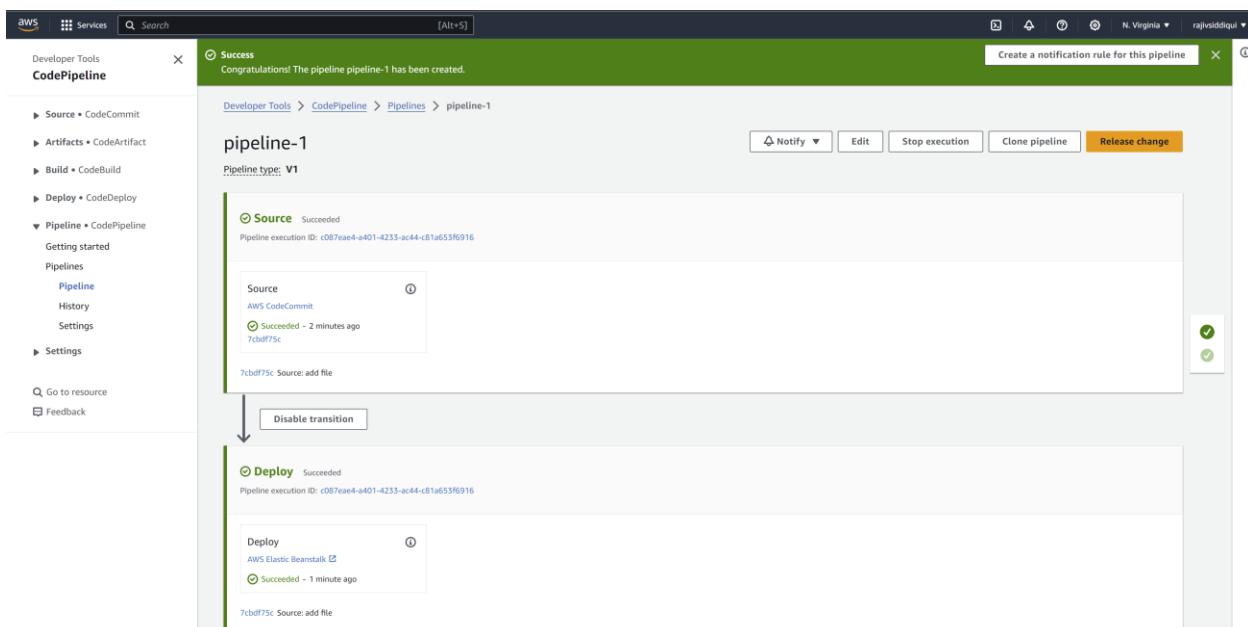
Pipeline settings

Pipeline name
pipeline-1
Pipeline type
V1
Artifact location
A new Amazon S3 bucket will be created as the default artifact store for your pipeline
Service role name
AWSCodePipelineServiceRole-us-east-1-pipeline-1

just review the and click Create pipeline



Now we can see the website deployed and website page color changed so means its deployed the new one.



Now edit pipeline add one more stage

The screenshot shows the AWS CodePipeline console with two main windows. The top window displays the pipeline named 'pipeline-1' with a single stage named 'Source'. The 'Edit' button in the top right corner is highlighted with a red box. The bottom window shows the 'Edit' interface for the pipeline. It includes sections for 'Edit: Variables', 'Edit: Source', and 'Edit: Deploy'. In the 'Edit: Deploy' section, there is a '+ Add stage' button, which is also highlighted with a red box.

Developer Tools

CodePipeline

Source • CodeCommit

Artifacts • CodeArtifact

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

History

Settings

Settings

Go to resource

Developer Tools

CodePipeline

Source • CodeCommit

Artifacts • CodeArtifact

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

History

Settings

Settings

Go to resource

Feedback

Developer Tools > CodePipeline > Pipelines > pipeline-1

pipeline-1

Pipeline type: V1

Source Succeeded

Pipeline execution ID: c0870ae4-a401-4233-ac44-81a653fb916

Source AWS CodeCommit

Succeeded - 17 minutes ago

7cdff75c

7cdff75c: Source:add file

Disable transition

Notify Edit Stop execution Clone pipeline Release change

Pipeline type

V1

Edit: Variables

Pipeline type V2 required

Name	Default value	Description
No variables		
No variables defined at the pipeline level in this pipeline.		

Edit: Source

Source AWS CodeCommit

Edit stage

+ Add stage

Edit: Deploy

Deploy AWS Elastic Beanstalk

Edit stage

+ Add stage

Developer Tools

CodePipeline

- ▶ Source • CodeCommit
- ▶ Artifacts • CodeArtifact
- ▶ Build • CodeBuild
- ▶ Deploy • CodeDeploy
- ▼ Pipeline • CodePipeline
 - Getting started
 - Pipelines
 - Pipeline
 - History
 - Settings
- ▶ Settings

[Go to resource](#)

[Feedback](#)

Edit: Source

Source

AWS CodeCommit

[Edit stage](#)

[+ Add stage](#)

Edit: Deploy

Deploy

AWS Elastic Beanstalk

[Edit stage](#)

[+ Add stage](#)

Edit: DeployToProd

[Cancel](#)

[Delete](#)

[Done](#)

[+ Add action group](#)

ManualApproval

Manual approval

[+ Add action](#)

[+ Add action group](#)

[+ Add stage](#)

Name Default value Description

No variables

Developer Tools

CodePipeline

Edit action

Action name

Choose a name for your action

ManualApproval

Action provider

Manual approval

Configure the approval request.

SNS topic ARN - optional

[Q](#)

URL for review - optional

Type the URL you want to provide to the reviewer as part of the approval request. The URL must begin with "http://" or "https://".

[Q](#)

Comments - optional

Comments you type here display for the reviewer in email notifications or the console.

Variable namespace - optional

Choose a namespace for the output variables from this action. You must choose a namespace if you want to use the variables this action produces in your configuration. [Learn more](#)

[Cancel](#)

[Done](#)

[+ Add stage](#)

Developer Tools

CodePipeline

- ▶ Source • CodeCommit
- ▶ Artifacts • CodeArtifact
- ▶ Build • CodeBuild
- ▶ Deploy • CodeDeploy
- ▼ Pipeline • CodePipeline
 - Getting started
 - Pipelines
 - Pipeline
 - History
 - Settings
- ▶ Settings

[Go to resource](#)

[Feedback](#)

Edit: Source

Source AWS CodeCommit

+ Add stage

Edit: Deploy

Deploy AWS Elastic Beanstalk

+ Add stage

Edit: DeployToProd

+ Add action group

ManualApproval Manual approval + Add action

+ Add action group

+ Add stage

Cancel Delete Done

Edit action

Action name
Choose a name for your action
DeployToProdBeanstalk
No more than 100 characters

Action provider
AWS Elastic Beanstalk

Region
US East (N. Virginia)

Input artifacts
Choose an input artifact for this action. [Learn more](#)

SourceArtifact
No more than 100 characters

Application name
Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.
Q my-beanstalk-1

Environment name
Choose an environment that you have already created in the AWS Elastic Beanstalk console. Or create an environment in the AWS Elastic Beanstalk console and then return to this task.
Q My-beanstalk-1-prod

Variable namespace - optional
Choose a namespace for the output variables from this action. You must choose a namespace if you want to use the variables this action produces in your configuration. [Learn more](#)

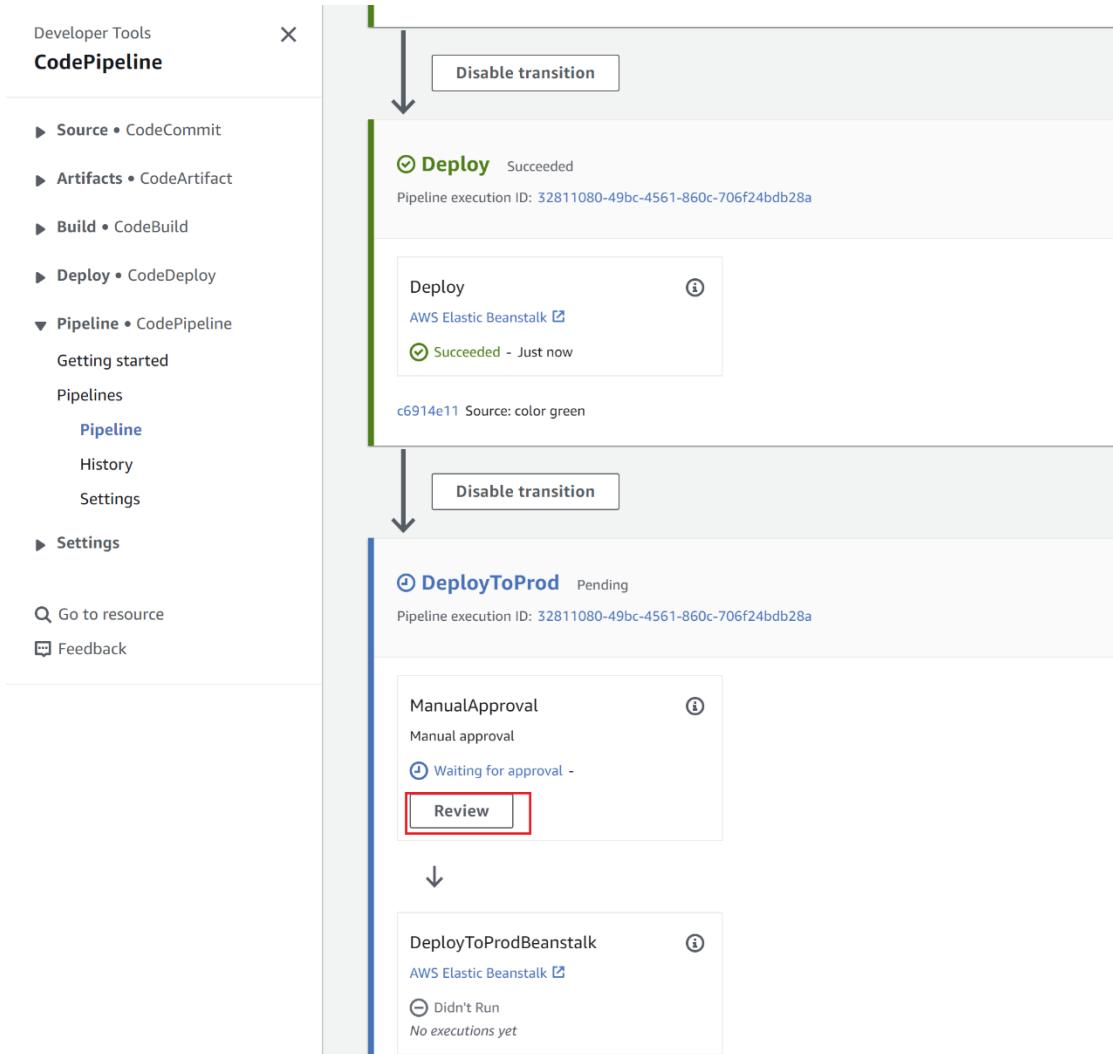
Cancel Done

The screenshot shows the AWS CodePipeline console with the following details:

- Left Sidebar:** Developer Tools > CodePipeline > Pipelines > pipeline-1
- Header:** Editing: pipeline-1
- Buttons:** Delete, Cancel, Save (Save is highlighted)
- Edit: Pipeline type:** Pipeline type V1
- Edit: Variables:** Pipeline type V2 required. A table shows "No variables" defined at the pipeline level.
- Edit: Source:** Stage named "Source" with "AWS CodeCommit" selected. Buttons: Edit stage, + Add stage.
- Edit: Deploy:** Stage named "Deploy" with a placeholder icon. Buttons: Edit stage.

Now get effect on this change we need to edit the index file from codecommit

And we can see its first edit the evn beanstalk then it ask for our manual approval if we approve it then it will update the production env beanstalk.



Codepipeline pricing

<https://aws.amazon.com/codepipeline/pricing/>

CodeBuild

AWS CodeBuild is a fully managed continuous integration service provided by Amazon Web Services (AWS). It automates the build, test, and packaging processes for applications and software projects, allowing developers to focus on writing code without the need to manage build servers or infrastructure.

Key Features of AWS CodeBuild:

- Build Automation:** CodeBuild automates the build and compilation process for applications, compiling source code, running tests, and generating build artifacts.
- Managed Build Environments:** Provides managed build environments that support various programming languages, build tools, and frameworks, allowing developers to build applications written in Java, Python, Node.js, Ruby, Go, Docker, and more.
- Custom Build Environments:** Offers flexibility by allowing users to create custom build environments with specific configurations, dependencies, and runtime versions.

4. **Scalable and Pay-as-You-Go:** Scales automatically accommodate build demands and charges based on the actual compute resources consumed during builds, optimizing costs.
5. **Integration with Source Control and CI/CD:** Integrates seamlessly with source code repositories (e.g., GitHub, CodeCommit) and CI/CD pipelines, triggering builds on code changes or specific events.
6. **Extensive Logging and Metrics:** Captures detailed build logs, metrics, and reports, allowing developers to analyze build performance and troubleshoot issues.
7. **Support for Build Plugins and Pre-configured Build Environments:** Supports integration with third-party build plugins and pre-configured build environments for popular build tools and frameworks.

Use Cases for AWS CodeBuild:

1. **Continuous Integration and Continuous Deployment (CI/CD):** Integrating CodeBuild into CI/CD pipelines for automated building, testing, and packaging of applications.
2. **Automated Testing:** Running unit tests, integration tests, or other automated tests as part of the build process to ensure code quality.
3. **Software Packaging:** Compiling and packaging software applications, libraries, or artifacts for deployment.
4. **Build Validation and Quality Assurance:** Validating code changes, performing static code analysis, and ensuring code quality standards are met before deployment.
5. **Multi-Language and Multi-Framework Support:** Supporting a wide range of programming languages, build tools, and frameworks, providing flexibility for diverse development environments.

AWS CodeBuild is suitable for development teams and organizations looking for a scalable and fully-managed solution to automate build processes, improve development workflows, and integrate seamlessly into their CI/CD pipelines for continuous integration and deployment of software applications.

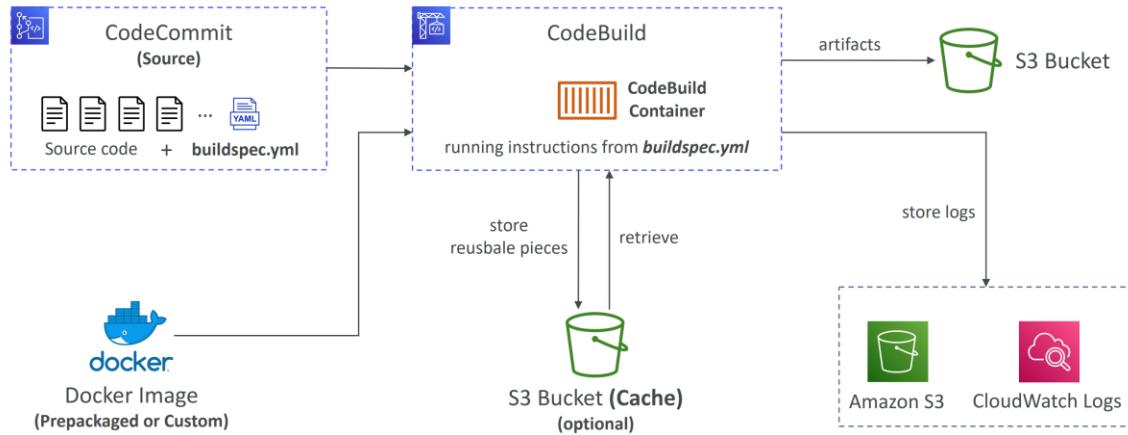
CodeBuild

- A fully managed continuous integration (CI) service
- Continuous scaling (no servers to manage or provision – no build queue)
- Compile source code, run tests, produce software packages, ...
- Alternative to other build tools (e.g., Jenkins)
- Charged per minute for compute resources (time it takes to complete the builds)
- Leverages Docker under the hood for reproducible builds
- Use prepackaged Docker images or create your own custom Docker image
- Security:
 - Integration with KMS for encryption of build artifacts
 - IAM for CodeBuild permissions, and VPC for network security
 - AWS CloudTrail for API calls logging
- Source – CodeCommit, S3, Bitbucket, GitHub
- Build instructions: Code file `buildspec.yml` or insert manually in Console
- Output logs can be stored in Amazon S3 & CloudWatch Logs
- Use CloudWatch Metrics to monitor build statistics
- Use EventBridge to detect failed builds and trigger notifications
- Use CloudWatch Alarms to notify if you need “thresholds” for failures
- Build Projects can be defined within CodePipeline or CodeBuild

CodeBuild – Supported Environments

- Java
- Ruby
- Python
- Go
- Node.js
- Android
- .NET Core
- PHP
- Docker – extend any environment you like

CodeBuild – How it Works



CodeBuild – `buildspec.yml`

- `buildspec.yml` file must be at the **root** of your code
- **env** – define environment variables
 - **variables** – plaintext variables
 - **parameter-store** – variables stored in SSM Parameter Store
 - **secrets-manager** – variables stored in AWS Secrets Manager
- **phases** – specify commands to run:
 - **install** – install dependencies you may need for your build
 - **pre_build** – final commands to execute before build
 - **Build** – actual build commands
 - **post_build** – finishing touches (e.g., zip output)
- **artifacts** – what to upload to S3 (encrypted with KMS)
- **cache** – files to cache (usually dependencies) to S3 for future build speedup

```

version: 0.2

env:
  variables:
    JAVA_HOME: "/usr/lib/jvm/java-8-openjdk-amd64"
    parameter-store:
      LOGIN_PASSWORD: /CodeBuild/dockerLoginPassword

phases:
  install:
    commands:
      - echo "Entered the install phase..."
      - apt-get update -y
      - apt-get install -y maven
  pre_build:
    commands:
      - echo "Entered the pre_build phase..."
      - docker login -u User -p $LOGIN_PASSWORD
  build:
    commands:
      - echo "Entered the build phase..."
      - echo "Build started on `date`"
      - mvn install
  post_build:
    commands:
      - echo "Entered the post_build phase..."
      - echo "Build completed on `date`"

artifacts:
  files:
    - target/messageUtil-1.0.jar

cache:
  paths:
    - "/root/.m2/**/*"
  
```

Lab: Code build

The screenshot shows the AWS CodeBuild console interface. On the left, there is a sidebar titled 'Developer Tools' with a 'CodeBuild' section expanded. Under 'CodeBuild', the following items are listed: Source (CodeCommit), Artifacts (CodeArtifact), Build (CodeBuild), Getting started, Build projects, Build history, Report groups, Report history, and Account metrics. The main content area has a dark header 'Developer Tools' and a title 'AWS CodeBuild'. Below the title, it says 'Build and test code with elastic scaling. Pay only for the build time you use.' A paragraph explains that AWS CodeBuild is a fully managed continuous integration service. To the right, there is a call-to-action box with a 'Create project' button.

Create build project

Project configuration

Project name

myBuild-1

A project name must be 2 to 255 characters. It can include the letters A-Z and a-z, the numbers 0-9, and the special characters - and _.

▼ Additional configuration

Description, Build badge, Concurrent build limit, tags

Description - *optional*

testing congratulation test

Build badge - *optional*

Enable build badge

Enable concurrent build limit - *optional*

Limit the number of allowed concurrent builds for this project.

Restrict number of concurrent builds this project can start

Tags

Key

Value

[Remove tag](#)

[Add tag](#)

Source

Add source

Source 1 - Primary

Source provider

AWS CodeCommit

Repository

my-node-js



Reference type

Choose the source version reference type that contains your source code.

Branch

Git tag

Commit ID

Branch

Choose a branch that contains the code to build.

main

Commit ID - *optional*

Choose a commit ID. This can shorten the duration of your build.



Source version [Info](#)

refs/heads/main

c6914e11 color green

► Additional configuration

Git clone depth, Git submodules

Environment

Environment image

Managed image

Use an image managed by AWS CodeBuild

Custom image

Specify a Docker image

Compute

EC2

Optimized for flexibility during action runs

Lambda

Optimized for speed and minimizes the start up time of workflow actions

Operating system

Ubuntu

Runtime(s)

Standard

Image

aws/codebuild/standard:7.0

Image version

Always use the latest image for this runtime version

Use GPU-enhanced compute

Privileged

Enable this flag if you want to build Docker images or want your builds to get elevated privileges

Service role

New service role

Create a service role in your account

Existing service role

Choose an existing service role from your account

Role name

Type your service role name

▼ Additional configuration

Timeout, certificate, VPC, compute type, environment variables, file systems

Timeout

Default timeout is 1 hour

keep as it is don't change anything

Hours

Minutes

Timeout must be between 5 minutes and 8 hours

Queued timeout

Default time in build queue is 8 hours

Hours

Minutes

Timeout must be between 5 minutes and 8 hours

Certificate

If you have a self-signed certificate or a certificate signed by a certification authority, choose the option to install it from your S3 bucket.

 Do not install any certificate Install certificate from your S3 bucket

VPC

Select a VPC that your AWS CodeBuild project will access.

Compute

- 3 GB memory, 2 vCPUs
- 7 GB memory, 4 vCPUs
- 15 GB memory, 8 vCPUs
- 145 GB memory, 72 vCPUs

Environment variables

Name	Value	Type	Remove
<input type="text"/>	<input type="text"/>	<input type="text" value="Plaintext"/> ▼	<input type="button" value="Remove"/>

[Add environment variable](#)

[Create parameter](#)

don't change anything

File systems

Identifier

ID

[Remove](#)

Directory path - *optional*

Mount point

Mount options - *optional*

[Add file system](#)

Buildspec

Build specifications

Use a buildspec file

Store build commands in a YAML-formatted buildspec file

Insert build commands

Store build commands as build project configuration

Buildspec name - *optional*

By default, CodeBuild looks for a file named buildspec.yml in the source code root directory. If your buildspec file uses a different name or location, enter its path from the source root here (for example, buildspec-two.yml or configuration/buildspec.yml).

Artifacts

[Add artifact](#)

Artifact 1 - Primary

Type

No artifacts



You might choose no artifacts if you are running tests or pushing a Docker image to Amazon ECR.

► Additional configuration

Cache, encryption key

Logs

CloudWatch

CloudWatch logs - *optional*

Checking this option will upload build output logs to CloudWatch.

Group name

Stream name

S3

S3 logs - *optional*

Checking this option will upload build output logs to S3.

Cancel

Create build project

Logs

CloudWatch

CloudWatch logs - *optional*

Checking this option will upload build output logs to CloudWatch.

Group name

Stream name

S3

S3 logs - *optional*

Checking this option will upload build output logs to S3.

Cancel

Create build project

Now click start build but it will fail because we did not add the yaml file

The screenshot shows the AWS CodeBuild configuration page for a project named 'myBuild-1'. The left sidebar contains navigation links for Source, Artifacts, Build, Deploy, Pipeline, and Settings. The main area is titled 'Configuration' and shows details like 'Source provider: AWS CodeCommit', 'Primary repository: my-node-js', and 'Artifacts upload location: -'. Below this is a 'Build history' section with tabs for Build history, Batch history, Build details, Build triggers, and Metrics. The 'Build history' tab is selected, showing a table with columns for Build run, Status, Build number, Source version, Submitter, Duration, and Completed. A message at the bottom states 'No results' and 'There are no results to display.'

The screenshot shows the AWS CodeBuild build status page for a specific build. The top bar indicates 'Build started' with the build ID 'myBuild-1:af6f44d2-8639-4509-9e69-1fe003003cdb'. The left sidebar is identical to the previous screenshot. The main area is titled 'Build status' and shows detailed information for each phase: Status (Failed), Initiator (root), Build ARN (arn:aws:codebuild:us-east-1:999838272208:build/myBuild-1:af6f44d2-8639-4509-9e69-1fe003003cdb), Start time (Nov 19, 2023 11:26 AM (UTC+6:00)), End time (Nov 19, 2023 11:26 AM (UTC+6:00)), and Build number (1). Below this is a table of phase details with columns for Name, Status, Context, Duration, Start time, and End time. The 'Phase details' tab is highlighted in red. One row in the table, 'DOWNLOAD_SOURCE', has a status of 'Failed' and a context of 'YAML_FILE_ERROR: YAML file does not exist', which is also highlighted with a red box.

Now add the buildspec.yml file and push it or we can run it from codecommit console

Now we click the Retry build and now we can see the build is run successfully

Developer Tools

CodeBuild

- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
 - Getting started
 - Build projects
 - Build project
 - Settings
 - Build history
 - Report groups
 - Report history
 - Account metrics
- Deploy • CodeDeploy
- Pipeline • CodePipeline
- Settings

Q Go to resource

Feedback

Build started
You have successfully started the following build: myBuild-1:cf87297a-f930-452e-97b4-4a5d74450303

Developer Tools > CodeBuild > Build projects > myBuild-1 > myBuild-1:cf87297a-f930-452e-97b4-4a5d74450303

myBuild-1:cf87297a-f930-452e-97b4-4a5d74450303

Stop build Retry build

Build status			
Status Succeeded	Initiator root	Build ARN arn:aws:codebuild:us-east-1:999836272208:build/myBuild-1:cf87297a-f930-452e-97b4-4a5d74450303	Resolved source version 6df590490c25d7e99f04d72a4854b4111d96a0
Start time Nov 19, 2023 11:38 AM (UTC+6:00)	End time Nov 19, 2023 11:38 AM (UTC+6:00)	Build number 2	
Build logs Phase details Reports Environment variables Build details Resource utilization			
Showing the last 50 lines of the build log. View entire log Tail logs			
Show previous logs			
<pre>1 [Container] 2023/11/19 05:38:29.357616 Waiting for agent ping 2 [Container] 2023/11/19 05:38:38.358556 Waiting for DOWNLOAD_SOURCE 3 [Container] 2023/11/19 05:38:37.327219 Phase 1 is DOWNLOAD_SOURCE 4 [Container] 2023/11/19 05:38:37.338990 CODEBUILD_SRC_DIR=/codebuild/output/src3135720951/src/git-codecommit.us-east-1.amazonaws.com/v1/repos/my-node-js 5 [Container] 2023/11/19 05:38:37.339524 YAML location is /codebuild/output/src3135720951/src/git-codecommit.us-east-1.amazonaws.com/v1/repos/my-node-js/buildspec.yml 6 [Container] 2023/11/19 05:38:37.341545 Not setting HTTP client timeout for source type codecommit</pre>			

Now add this build in the pipeline

Developer Tools

CodePipeline

- Source • CodeCommit
- Artifacts • CodeArtifact
- Build • CodeBuild
- Deploy • CodeDeploy
- Pipeline • CodePipeline
 - Getting started
 - Pipelines Pipelines

Pipelines Info

Developer Tools > CodePipeline > Pipelines

Name	Type	Most recent execution	Latest source revisions	Last executed
pipeline-1	V1	Succeeded	Source - 6df59049: add buildspec.yml file	10 minutes ago

pipeline-1

Pipeline type: V1

Edit Stop execution Clone pipeline Release change

Source Succeeded
Pipeline execution ID: 2416b548-9e0a-4554-b7d8-bfc84a2b8ab4

Source AWS CodeCommit
Succeeded - 15 minutes ago 6df59049

6df59049 Source: add buildspec.yml file

Editing: pipeline-1

Edit Pipeline type

Pipeline type V1

Delete Cancel Save Edit

Developer Tools X

CodePipeline

- ▶ Source • CodeCommit
- ▶ Artifacts • CodeArtifact
- ▶ Build • CodeBuild
- ▶ Deploy • CodeDeploy
- ▼ Pipeline • CodePipeline
 - Getting started
 - Pipelines
 - Pipeline**
 - History
 - Settings
 - ▶ Settings
- Q Go to resource
- ✉ Feedback

Developer Tools > CodePipeline > Pipelines > pipeline-1 > Edit pipeline-1

Editing: pipeline-1

Edit: Pipeline type

Pipeline type
The pipeline type determines the pipeline structure and availability of parameters such as triggers. Pipeline type selection will impact features and pricing. [Which pipeline is right for me?](#)

V1 V2

Edit: Variables

Pipeline type V2 required

Name	Default value	Description
No variables		
No variables defined at the pipeline level in this pipeline.		

Edit: Source

Source Edit stage

AWS CodeCommit

Make sure add this stage just after the source

+ Add stage

Delete Cancel Done

Add stage

Stage name

No more than 100 characters

Cancel Add stage

Now add the stage

Developer Tools X

CodePipeline

- ▶ Source • CodeCommit
- ▶ Artifacts • CodeArtifact
- ▶ Build • CodeBuild
- ▶ Deploy • CodeDeploy
- ▼ Pipeline • CodePipeline
 - Getting started
 - Pipelines
 - Pipeline**
 - History
 - Settings
 - ▶ Settings
- Q Go to resource
- ✉ Feedback

Developer Tools > CodePipeline > Pipelines > pipeline-1 > Edit pipeline-1

Editing: pipeline-1

Edit: Variables

Pipeline type V2 required

Name	Default value	Description
No variables		
No variables defined at the pipeline level in this pipeline.		

Edit: Source

Source Edit stage

AWS CodeCommit

+ Add stage

Edit: BuildAndTest

+ Add action group

Cancel Delete Done

Edit action

Action name
Choose a name for your action
TestForCongratulation
No more than 100 characters

Action provider
AWS CodeBuild

Region
US East (N. Virginia)

Input artifacts
Choose an input artifact for this action. Learn more [\[?\]](#)

SourceArtifact
[Add](#)
No more than 100 characters

Project name
Choose a build project that you have already created in the AWS CodeBuild console. Or create a build project in the AWS CodeBuild console and then return to this task.
Q. myBuild-1
or [Create project \[?\]](#)

Environment variables - optional
Choose the key, value, and type for your CodeBuild environment variables. In the value field, you can reference variables generated by CodePipeline. Learn more [\[?\]](#)

[Add environment variable](#)

Build type
 Single build
Triggers a single build.
 Batch build
Triggers multiple builds as a single execution.

Variable namespace - optional
Choose a namespace for the output variables from this action. You must choose a namespace if you want to use the variables this action produces in your configuration. Learn more [\[?\]](#)

Output artifacts
Choose a name for the output of this action.

[Add](#)
No more than 100 characters

[Cancel](#) [Done](#)

Developer Tools > CodePipeline

Editing: pipeline-1

[Delete](#) [Cancel](#) [Save](#)

Edit: Pipeline type

Pipeline type
The pipeline type determines the pipeline structure and availability of parameters such as triggers. Pipeline type selection will impact features and pricing. [Which pipeline is right for me?](#)

V1 V2

Edit: Variables

Pipeline type V2 required

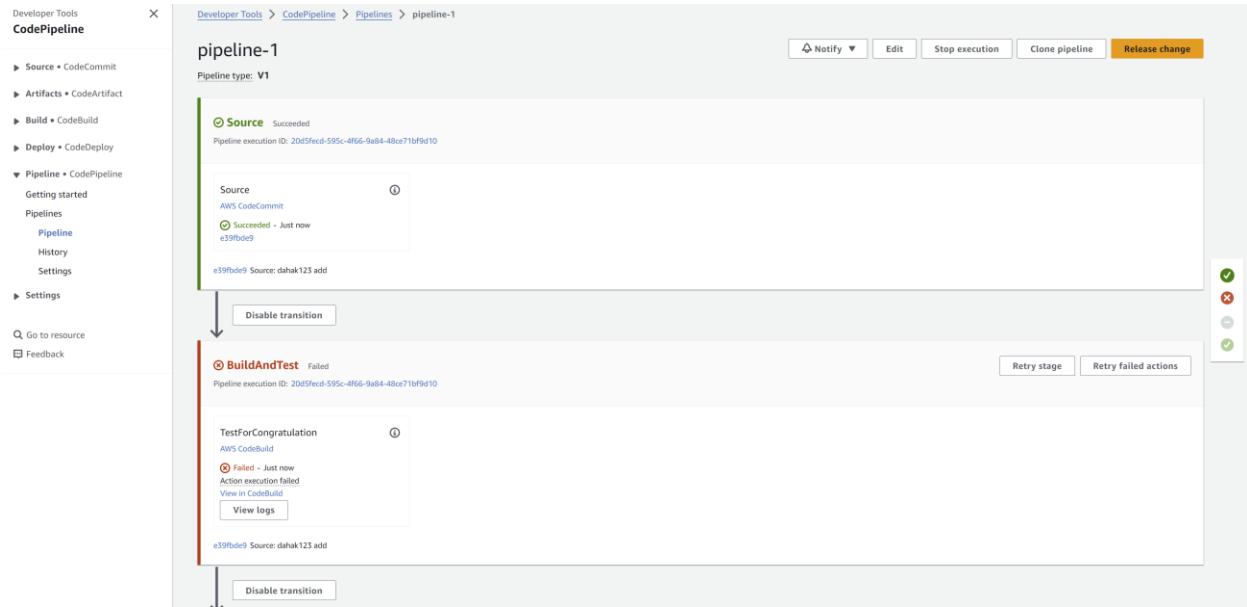
Name	Default value	Description
No variables		
No variables defined at the pipeline level in this pipeline.		

Edit: Source

Source
AWS CodeCommit

[Edit stage](#)

Now edit the index file and push it and now this time we want make this build fail so we will delete the congratulation text from index file.



Now we will edit the index file again and add the congratulation text again and push the code and we can see now its successfully test and build and update the production environment.

