



# Building code with AWS Code Build

**CodeBuild** is a fully managed continuous integration (CI) service provided by AWS that compiles source code, runs tests, and produces deployable software packages. It is designed to automate the build process in software development pipelines, integrating seamlessly with other AWS services and CI/CD tools.

## Key Features of AWS CodeBuild:

1. **Build Automation:** Automates compiling, testing, and packaging of code.
2. **Fully Managed:** No need to provision or manage servers; AWS handles the infrastructure.
3. **Scalability:** Scales automatically to handle multiple builds concurrently.
4. **Integration:** Works with other AWS services like CodePipeline, CodeCommit, S3, and third-party repositories such as GitHub, GitLab, and Bitbucket.
5. **Custom Build Environments:** Supports pre-configured environments for popular programming languages and allows custom build environments using Docker.
6. **Pay-as-you-go Pricing:** Charges are based on the number of build minutes used.

## How AWS CodeBuild Works

1. **Input:** The source code is stored in a repository (e.g., CodeCommit, GitHub, Bitbucket, or S3).
2. **Build Specification:** A `buildspec.yml` file defines the build steps, including phases like `install`, `pre-build`, `build`, and `post-build`.
3. **Build Environment:** A build container is provisioned based on the programming language and runtime environment.
4. **Output:** CodeBuild generates artifacts (e.g., binaries, packaged applications) and can store them in Amazon S3 or deploy them directly using other AWS services.

## Example Workflow

1. **Source:** Code is pushed to a repository (e.g., GitHub).
2. **Trigger:** CodeBuild is triggered (manually or via a CI/CD tool like CodePipeline).
3. **Build Phases:**
  - o **Install:** Install dependencies.
  - o **Pre-Build:** Prepare the environment (e.g., database migrations).
  - o **Build:** Compile source code or package the application.
  - o **Post-Build:** Run tests or deploy artifacts.

4. **Artifacts:** The resulting application or output is stored in S3 or deployed.

## Benefits of CodeBuild

1. **Ease of Use:** Simplifies build processes with managed infrastructure.
2. **Cost-Effective:** Pay only for the resources used during builds.
3. **Flexibility:** Custom build environments and support for multiple programming languages.
4. **Security:** Integrated with AWS Identity and Access Management (IAM) for secure access control.

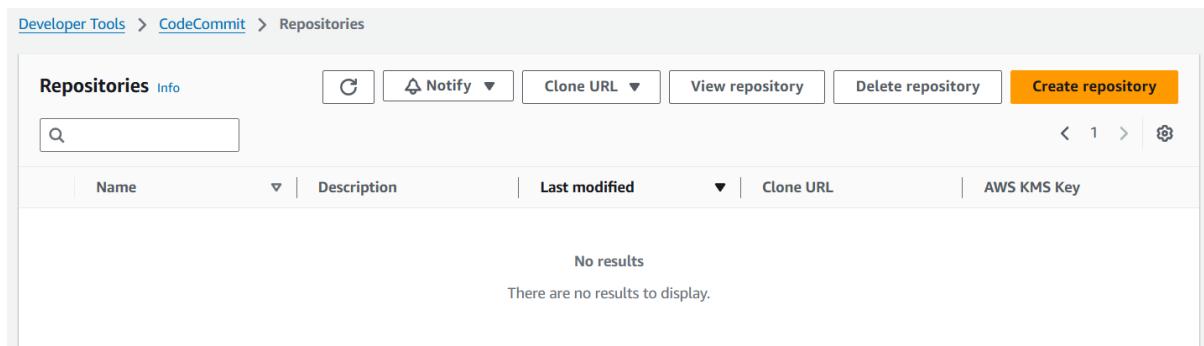
## Common Use Cases

- **Continuous Integration:** Automating code builds and testing in a CI/CD pipeline.
- **Build Automation:** Simplifying repetitive tasks like dependency management and compilation.
- **Code Testing:** Running unit, integration, or performance tests during the build phase.
- **Container Builds:** Building and pushing Docker containers to Amazon Elastic Container Registry (ECR).

CodeBuild is a versatile tool for modern DevOps workflows, streamlining software delivery in cloud-native environments.

## 😊 To begin with the Lab:

1. So, before building the code build, we need a code commit repository.
2. First, we need to open the code commit and click on Create repository. Then give a name to your repository and click on create.



The screenshot shows the AWS CodeCommit console under the 'Developer Tools > CodeCommit > Repositories' path. At the top, there's a navigation bar with 'Repositories' and 'Info' tabs, and several buttons: 'C' (Create), 'Notify', 'Clone URL', 'View repository', 'Delete repository', and 'Create repository' (which is highlighted in orange). Below the navigation bar is a search bar. The main area displays a table with columns: Name, Description, Last modified, Clone URL, and AWS KMS Key. A message at the bottom center says 'No results' and 'There are no results to display.'

## Repository settings

Repository name

100 characters maximum. Other limits apply.

Description - optional

1,000 characters maximum

Tags

[Add tag](#)

▶ Additional configuration

AWS KMS key

[Cancel](#)
Create

- Once our repository is created then we need to create a file for that scroll down to the bottom and click on create the file.

[demo-repo](#) [Info](#)

Name

Add 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.

[Create file](#)

- Here we need to use this code to create the file. Then give your file a name, the author's name and email address, and click on commit changes.

```
#include <iostream>
```

```
int main()
{
    std::cout << "Hello\n";
}
```

**demo-repo** [Info](#)

The code editor uses the Tab key to control indentation. To navigate away from the code editor, use Escape plus Tab keys.

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << "Hello\n";
6 }
```

**Commit changes to main**

File name  
For example, file.txt  
  
demo-repo/demo.cpp

Author name

Email address

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

[Cancel](#) [Commit changes](#)

5. Now navigate to code build to build your first project. Click on Create project.

Developer Tools > [CodeBuild](#) > Build projects

**Build projects** [Info](#)

[!\[\]\(ccb87cee2d02fdfdb93bab74b01d2585\_img.jpg\) Actions](#) [Create trigger](#) [View details](#) [Start build](#) [Create project](#)

Your projects ▾ < 1 > ⚙

Name	Source provider	Repository	Latest build status	Description	Last Modified
No results There are no results to display.					

6. We need to give it a project name and scroll down.

## Project configuration

Project name

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 \_.

### Public build access - *optional*

Public build access allows you to make the build results, including logs and artifacts, for this project available for the general public.

Enable public build access

### ► Additional configuration

Description, Build badge, Concurrent build limit, tags

7. In the source choose AWS Code commit as your source provider and repository is the demo repo and scroll down.

## Source

[Add source](#)

### Source 1 - Primary

Source provider



Repository



#### 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.



#### Commit ID - *optional*

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



### ► Additional configuration

Git clone depth, Git submodules

8. For the environment choose the same settings as shown below.

## Environment

### Provisioning model [Info](#)

#### On-demand

Automatically provision build infrastructure in response to new builds.

#### Reserved capacity

Use a dedicated fleet of instances for builds. A fleet's compute and environment type will be used for the project.

### 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



### 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

codebuild-demo-build-project-service-role

Type your service role name

### ► Additional configuration

Timeout, privileged, certificate, VPC, compute type, environment variables, file systems, auto-retry

9. Then for the build spec file choose to insert the build command and use the code given below.

**version: 0.2**

**phases:**

**build:**

**commands:**

- echo "Building the application"
- g++ demo.cpp -o my-app # Compile the demo.cpp file into an executable named my-app

**artifacts:****files:**

- my-app # Include the compiled executable as an artifact

**Buildspec**

Build specifications

Insert build commands  
Store build commands as build project configuration

Use a buildspec file  
Store build commands in a YAML-formatted buildspec file

**Build commands** [Info](#) [Edit](#)

```
1 version: 0.2
2
3 phases:
4   build:
5     commands:
6       - echo "Building the application"
7       - g++ demo.cpp -o my-app # Compile the demo.cpp file into an executable named my-app
8
9 artifacts:
10  files:
11    - my-app # Include the compiled executable as an artifact
12
```

10. Then move to the artifacts here you need to choose amazon S3 and choose a bucket in the same region where you are working.

**Artifacts** [Add artifact](#)

**Artifact 1 - Primary**

Type

Amazon S3

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

Bucket name

coderepbuckets3

11. Once it is done scroll down and turn off the cloud watch logs click on Create.
12. Now you need to click on Start build so that you can build your project.

Developer Tools > CodeBuild > Build projects > demo-build-project

## demo-build-project

Actions ▾ Create trigger Edit Clone Debug build Start build with overrides **Start build**

**Configuration**

Source provider AWS CodeCommit	Primary repository demo-repo	Artifacts upload location coderepobuckets3	Service role arn:aws:iam::463646775279:role/service-role/codebuild-demo-build-project-service-role
Public builds Disabled			

13. Below you can see that our build has been succeeded.

### demo-build-project:5d4289e1-a010-43ab-8f0c-51bb9c8e05d9

Stop build Retry build

**Build status**

Status <b>Succeeded</b>	Initiator cfpulkit	Build ARN <a href="#">arn:aws:codebuild:ap-south-1:463646775279:build/demo-build-project:5d4289e1-a010-43ab-8f0c-51bb9c8e05d9</a>	Resolved source version <a href="#">96a7518feeabc9e23596d0324444d20a63373eb9</a>
Start time Nov 25, 2024 5:42 PM (UTC+5:30)	End time Nov 25, 2024 5:42 PM (UTC+5:30)	Build number 2	

14. Once this is done navigate to S3 and in your bucket which you defined as the artifact you will the binary here.

### demo-build-project/

[Copy S3 URI](#)

Objects **1** [Info](#) [C](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">my-app</a>	-	November 25, 2024, 17:42:36 (UTC+05:30)	16.0 KB	Standard