

AWS CodeBuild

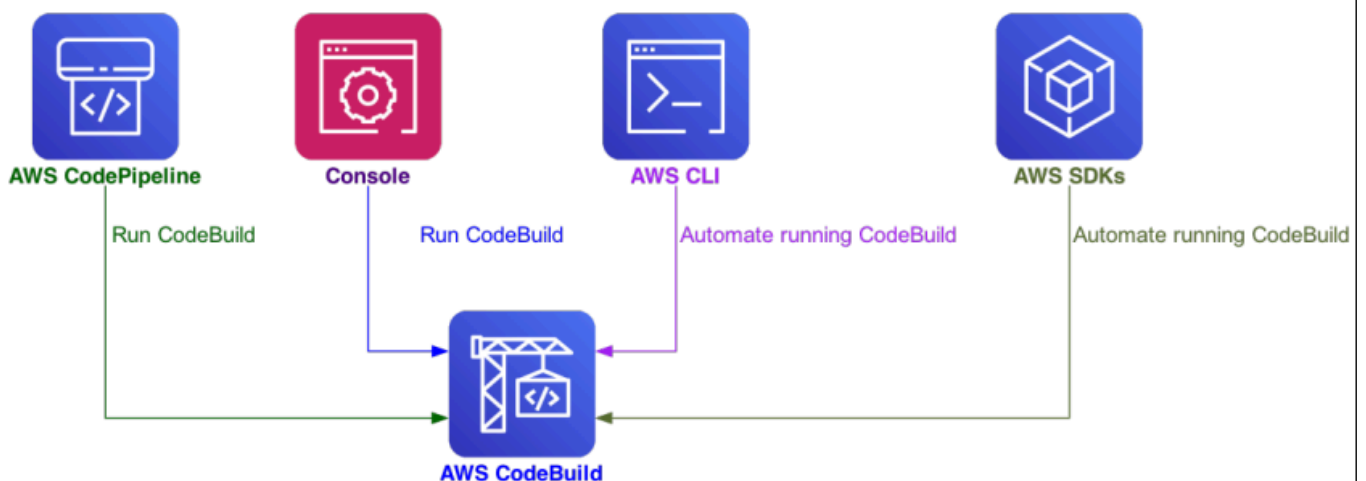


AWS CodeBuild is a fully managed continuous integration (CI) service that compiles source code, runs unit tests, and produces deployable software artifacts. It eliminates the need to set up, manage, and scale your own build servers.

Key Features of AWS CodeBuild

- ✓ **Fully Managed** – No need to provision or manage build servers.
- ✓ **Scalability** – Scales automatically to meet build demand.
- ✓ **Pay-As-You-Go Pricing** – Charges are based on build minutes used.
- ✓ **Security** – Integrated with AWS IAM for fine-grained access control.
- ✓ **Multi-Language Support** – Supports Java, Python, Node.js, Go, .NET, and more.
- ✓ **Integration** – Works with AWS CodePipeline, GitHub, GitLab, Bitbucket, Jenkins, etc.
- ✓ **Custom Build Environments** – Allows use of custom Docker containers.

How to run CodeBuild



How AWS CodeBuild Works

Step 1: Define a Build Specification (buildspec.yml)

- A buildspec.yml file is used to define build commands, environment variables, and build steps.

```
version: 0.2

phases:
  install:
    runtime-versions:
      java: corretto11
  pre_build:
    commands:
      - echo Nothing to do in the pre_build phase...
  build:
    commands:
      - echo Build started on `date`
      - mvn install
  post_build:
    commands:
      - echo Build completed on `date`
artifacts:
  files:
    - target/messageUtil-1.0.jar
```

Step 2: Create a Build Project

- Go to AWS **CodeBuild Console** → Click **Create Build Project**.
- Choose a source provider (CodeCommit, GitHub, S3, or Bitbucket).
- Define the environment (managed image or custom Docker image).
- Specify the buildspec file or define commands manually.
- Configure artifacts storage (S3 or no artifacts).
- Start the build.

Step 3: Monitor and Debug Builds

- View real-time logs in the **CodeBuild Console**.
- Use **Amazon CloudWatch Logs** for detailed build logs.

AWS CodeBuild Integration with AWS CodePipeline

AWS CodePipeline automates the CI/CD pipeline. CodeBuild can be integrated into the pipeline as the build step.

How it Works

1. **Source Stage** → CodeCommit, GitHub, or S3 triggers a build.

2. **Build Stage** → AWS CodeBuild compiles and tests the code.
3. **Deploy Stage** → AWS CodeDeploy or Lambda deploys the application.

Example **CodePipeline** with **CodeBuild**:

```
stages:
  - name: Source
    actions:
      - name: SourceAction
        provider: CodeCommit
  - name: Build
    actions:
      - name: CodeBuildAction
        provider: CodeBuild
  - name: Deploy
    actions:
      - name: DeployAction
        provider: CodeDeploy
```

AWS CodeBuild vs Other Build Services

Feature	AWS CodeBuild	Jenkins	GitHub Actions	GitLab CI/CD
Fully Managed	✔ Yes	✘ No (Self-hosted)	✔ Yes	✔ Yes
Scalability	✔ Auto-scales	✘ Manual Scaling	✔ Auto-scales	✔ Auto-scales
Pricing	Pay-as-you-go	Free (Self-hosted)	Free for basic	Free for basic
Security	AWS IAM integration	Requires setup	GitHub IAM	GitLab IAM
Multi-Cloud	AWS only	Any cloud	GitHub only	GitLab only

AWS CodeBuild Pricing

AWS CodeBuild is charged based on the **compute minutes used per build**.

- **Build Duration** = Time taken to execute buildspec.yml.
- **Pricing Examples:**
 - **Small Instance (3GB RAM, 2 vCPUs)** → \$0.005/min
 - **Medium Instance (7GB RAM, 4 vCPUs)** → \$0.01/min
 - **Large Instance (15GB RAM, 8 vCPUs)** → \$0.02/min

Example:

- A build that runs for **10 minutes** on a **medium instance** costs:
10 × \$0.01 = \$0.10

Best Practices for AWS CodeBuild

- ✔ **Use IAM Roles** – Grant least privilege permissions for security.
- ✔ **Optimize Build Time** – Cache dependencies to speed up builds.
- ✔ **Use Parallel Builds** – Run multiple builds to reduce deployment time.
- ✔ **Monitor Build Logs** – Use AWS CloudWatch for detailed insights.
- ✔ **Secure Build Artifacts** – Store artifacts in **Amazon S3 with encryption**.
- ✔ **Use VPC Configuration** – Run builds in a private VPC for security.

Common Use Cases

- ✦ **Continuous Integration (CI)** → Automate code compilation and unit testing.
- ✦ **Container Image Builds** → Build Docker images and push them to Amazon ECR.
- ✦ **Serverless Application Deployment** → Deploy Lambda functions and APIs.
- ✦ **Multi-Environment Builds** → Deploy code to different AWS accounts/environments.
- ✦ **Machine Learning Model Training** → Automate ML pipeline builds.

Common Issues & Troubleshooting

Issue	Solution
Build fails due to missing dependencies	Ensure dependencies are installed in <code>buildspec.yml</code> .
IAM Access Denied Error	Grant <code>AWSCodeBuildAdminAccess</code> to the IAM role.
Slow Build Performance	Enable caching for dependencies.
Docker Build Failure	Use a CodeBuild image with Docker support.

Conclusion

AWS CodeBuild is a **powerful, fully managed build service** that automates code compilation, testing, and packaging in AWS environments. It integrates seamlessly with AWS CodePipeline for **end-to-end CI/CD workflows**, supports **Docker builds**, and provides **scalability, security, and pay-as-you-go pricing**.