Success
Congratulations! The pipeline mishraabhi has been created.

Developer Tools > CodePipeline > Pipelines > mishraabhi

mishraabhi

| Edit | Stop execution | Create trigger | Clone pipeline | Release change |

Pipeline Executions Triggers Settings Tags Stage

✓ ✓ ✓ ⓘ

Source
41afb34b-7e84-4f32-9d36-4afa21485cfe
All actions succeeded.

Source
GitHub (via OAuth app) ↗
5 minutes ago

3effd2f5 ↗ Source: change bui ⋯

Build
41afb34b-7e84-4f32-9d36-4afa21485cfe
All actions succeeded.

Build
AWS CodeBuild
1 minutes ago

3effd2f5 ↗ Source: change bui ⋯

Deploy
41afb34b-7e84-4f32-9d36-4afa21485cfe
All actions succeeded.

Deploy
Amazon S3 ↗
4 minutes ago

3effd2f5 ↗ Source: change bui ⋯

---

my-static-website-mishra.s3-website.ap-south-1.amazonaws.com

🚀 **Hello from
CodePipeline + CodeBuild + CodeDeploy** ✅

Deployed at: **8/30/2025, 12:01:42 PM**

✨ Modern Static Website

A clean, light, and professional site powered by AWS CI/CD

⚡ Fast Deployment

Push code → Go live instantly with
AWS CodePipeline.

🔒 Secure Infrastructure

IAM roles, policies & CI/CD security
best practices built-in.

🌐 Global Reach

Static sites distributed via S3 +
CloudFront worldwide.

🚀 Future Ready

AI automation + DevOps practices for
scalability.

Step 2
Choose pipeline settings

Step 3
Add source stage

Step 4
Add build stage

Step 5
Add test stage

Step 6
Add deploy stage

Step 7
Review

Build – optional

Build provider
Choose the tool you want to use to run build commands and specify artifacts for your build action.

( ) Commands   (●) Other build providers

AWS CodeBuild ▼

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.

🔍 codebuild-demo ✕   or   Create project ⧉

☐ Define buildspec override - *optional*
Buildspec file or definition that overrides the latest one defined in the build project, for this build only.

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   ( ) Batch build
Triggers a single build.   Triggers multiple builds as a single execution.

Region
Asia Pacific (Mumbai) ▼

Input artifacts
Choose an input artifact for this action. Learn more ⧉

[▼]

SourceArtifact ✕
Defined by: Source

☑ Enable automatic retry on stage failure

Cancel   [Previous]   [Skip build stage]   [Next]

Step 1
Choose creation option

Step 2
Choose pipeline settings

Step 3
Add source stage

Step 4
Add build stage

Step 5
Add test stage

Step 6
Add deploy stage

Step 7
Review

Add deploy stage Info
Step 6 of 7

Deploy – optional

Deploy provider
Choose how you want to deploy your application or content. Choose the provider, and then provide the configuration details for this provider.

AWS CodeDeploy ▼

Region
Asia Pacific (Mumbai) ▼

Input artifacts
Choose an input artifact for this action. Learn more ⧉

[▼]

BuildArtifact ✕   SourceArtifact ✕
Defined by: Build   Defined by: Source

No more than 100 characters

Application name
Choose an application that you have already created in the AWS CodeDeploy console. Or create an application in the AWS CodeDeploy console and then return to this task.

🔍 codedeployabhi-demo ✕

Deployment group
Choose a deployment group that you have already created in the AWS CodeDeploy console. Or create a deployment group in the AWS CodeDeploy console and then return to this task.

🔍 codedeploymemt-group ✕

codedeploymemt-group

☐ Enable automatic retry on stage failure

Cancel   [Previous]   [Skip deploy stage]   [Next]

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

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).

buildspec.yml

▼ Batch configuration

You can run a group of builds as a single execution. Batch configuration is also available in advanced option when starting build.

☐ Define batch configuration - *optional*
You can also define or override batch configuration when starting a build batch.

▼ 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
🔍 cicd-s3-80   ✕

Name
The name of the folder or compressed file in the bucket that will contain your output artifacts. Use Artifacts packaging under Additional configuration to choose whether to use a folder or compressed file. If the name is not provided, defaults to project name.

☐ Enable semantic versioning
Use the artifact name specified in the buildspec file

Path - *optional*
The path to the build output zip file or folder.

Step 2
Choose pipeline settings

Step 3
Add source stage

Step 4
Add build stage

Step 5
Add test stage

Step 6
Add deploy stage

Step 7
Review

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.

🔍 codecommit-demo   ✕

Branch name
Choose a branch of the repository.

🔍 master   ✕

master

If disabled, follow AWS documentation to create an EventBridge rule for your source. Learn more ↗

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. Learn more ↗

☑ Enable automatic retry on stage failure

Cancel   Previous   Next

Developer Tools
CodeBuild
X

Developer Tools > CodeBuild > Build projects > Create build project

Create build project

▶ Source • CodeCommit

▶ Artifacts • CodeArtifact

▼ Build • CodeBuild
  Getting started
  Build projects
  Build history
  Report groups
  Report history
  Compute fleets **New**
  Account metrics
  ▼ Related integrations
    Jenkins ↗
    GitHub Actions ↗ **New**
    GitLab runners ↗ **New**

▶ Deploy • CodeDeploy

▶ Pipeline • CodePipeline

▶ Settings

🔍 Go to resource

💬 Feedback

Project configuration

Project name

`codebuild-demo`

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 _.

Project type

Select what type of project you would like to create. Info ↗

○ **Default project**
Create a custom CodeBuild project.

○ **Runner project**
Create a CodeBuild managed runner for workflows in GitHub Actions, GitHub Enterprise Actions, GitLab, or Buildkite.

▶ **Additional configuration**
Description, public build access, build badge, concurrent build limit, tags

▼ Source

[Add source]

Source 1 - Primary

Source provider

AWS CodeCommit ▼

Repository

🔍 codecommit-demo ✕

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.

master ▼

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

🔍 a71afb1d78e4d0bcf3a5b5a67ac045bc50a7da ✕

Developer Tools
CodeBuild
X

▶ Source • CodeCommit

▶ Artifacts • CodeArtifact

▼ Build • CodeBuild
  Getting started
  Build projects
  Build history
  Report groups
  Report history
  Compute fleets **New**
  Account metrics
  ▼ Related integrations
    Jenkins ↗
    GitHub Actions ↗ **New**
    GitLab runners ↗ **New**

▶ Deploy • CodeDeploy

▶ Pipeline • CodePipeline

▶ Settings

🔍 Go to resource

💬 Feedback

Running mode

● **Container**
Running on Docker container

○ **Instance**
Running on EC2 instance directly

Operating system

Amazon Linux ▼

Runtime(s)

Standard ▼

Image

aws/codebuild/amazonlinux-x86_64-standard:5.0 ▼

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 ARN

🔍 arn:aws:iam::599822595047:role/codebuild-role ✕

☑ Allow AWS CodeBuild to modify this service role so it can be used with this build project.

▶ **Additional configuration**
Timeout, privileged, certificate, VPC, compute type, environment variables, file systems, auto-retry, registry credential

▼ 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

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

Developer Tools > CodeDeploy > Applications > codedeployabhi-demo > Create deployment group

Create deployment group

Application

Application
codedeployabhi-demo
Compute type
EC2/On-premises

Deployment group name

Enter a deployment group name

codedeployment-group

100 character limit

Service role

Enter a service role
Enter a service role with CodeDeploy permissions that grants AWS CodeDeploy access to your target instances.

🔍 arn:aws:iam::599822595047:role/codedeploy-role   ✕

Deployment type

Choose how to deploy your application

○ In-place
Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update

○ Blue/green
Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the original environment are deregistered and can be terminated.

Choose how to deploy your application

○ In-place
Updates the instances in the deployment group with the latest application revisions. During a deployment, each instance will be briefly taken offline for its update

○ Blue/green
Replaces the instances in the deployment group with new instances and deploys the latest application revision to them. After instances in the replacement environment are registered with a load balancer, instances from the original environment are deregistered and can be terminated.

Environment configuration

Select any combination of Amazon EC2 Auto Scaling groups, Amazon EC2 instances, and on-premises instances to add to this deployment

☐ Amazon EC2 Auto Scaling groups

☑ Amazon EC2 instances
1 unique matched instance. Click here for details ↗

You can add up to three groups of tags for EC2 instances to this deployment group.
One tag group: Any instance identified by the tag group will be deployed to.
Multiple tag groups: Only instances identified by all the tag groups will be deployed to.

Tag group 1

Key   Value - optional
🔍 Name   ✕   🔍 server2   ✕   Remove tag

Add tag

+ Add tag group

☐ On-premises instances

Matching instances
1 unique matched instance. Click here for details ↗

Agent configuration with AWS Systems Manager Info

⚠ We recommend configuring your CodeDeploy Agent install and updates with AWS Systems Manager.

```
● codedeploy-agent.service - AWS CodeDeploy Host Agent
     Loaded: loaded (/usr/lib/systemd/system/codedeploy-agent.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-08-29 02:31:54 UTC; 2min 10s ago
   Main PID: 27971 (ruby)
      Tasks: 3 (limit: 1111)
     Memory: 66.0M
        CPU: 1.104s
     CGroup: /system.slice/codedeploy-agent.service
             ├─27971 "codedeploy-agent: master 27971"
             └─27973 "codedeploy-agent: InstanceAgent::Plugins::CodeDeployPlugin::CommandPoller of master 27971"

Aug 29 02:31:53 ip-172-31-47-47.ap-south-1.compute.internal systemd[1]: Starting codedeploy-agent.service - AWS CodeDeploy Host Agent...
Aug 29 02:31:54 ip-172-31-47-47.ap-south-1.compute.internal systemd[1]: Started codedeploy-agent.service - AWS CodeDeploy Host Agent.
```

```
lines 1-13/13 (END)
[ec2-user@ip-172-31-47-47 tmp]$
[ec2-user@ip-172-31-47-47 tmp]$ sudo yum update -y
Last metadata expiration check: 0:43:07 ago on Fri Aug 29 01:52:29 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-47-47 tmp]$ sudo amazon-linux-extras enable nginx1
sudo: amazon-linux-extras: command not found
[ec2-user@ip-172-31-47-47 tmp]$
[ec2-user@ip-172-31-47-47 tmp]$ sudo yum install -y nginx
Last metadata expiration check: 0:43:24 ago on Fri Aug 29 01:52:29 2025.
Package nginx-1:1.28.0-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-47-47 tmp]$ sudo systemctl start nginx
[ec2-user@ip-172-31-47-47 tmp]$ sudo systemctl enable nginx
[ec2-user@ip-172-31-47-47 tmp]$ sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-08-29 01:52:34 UTC; 43min ago
   Main PID: 4055 (nginx)
      Tasks: 2 (limit: 1111)
     Memory: 2.5M
        CPU: 59ms
     CGroup: /system.slice/nginx.service
             ├─4055 "nginx: master process /usr/sbin/nginx"
             └─4062 "nginx: worker process"

Aug 29 01:52:34 ip-172-31-47-47.ap-south-1.compute.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Aug 29 01:52:34 ip-172-31-47-47.ap-south-1.compute.internal nginx[3953]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Aug 29 01:52:34 ip-172-31-47-47.ap-south-1.compute.internal nginx[3953]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Aug 29 01:52:34 ip-172-31-47-47.ap-south-1.compute.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[ec2-user@ip-172-31-47-47 tmp]$
```

Developer Tools > CodeCommit > Repositories > Create repository

ⓘ AWS CodeCommit is no longer available to new customers. Existing customers of AWS CodeCommit can   ✕
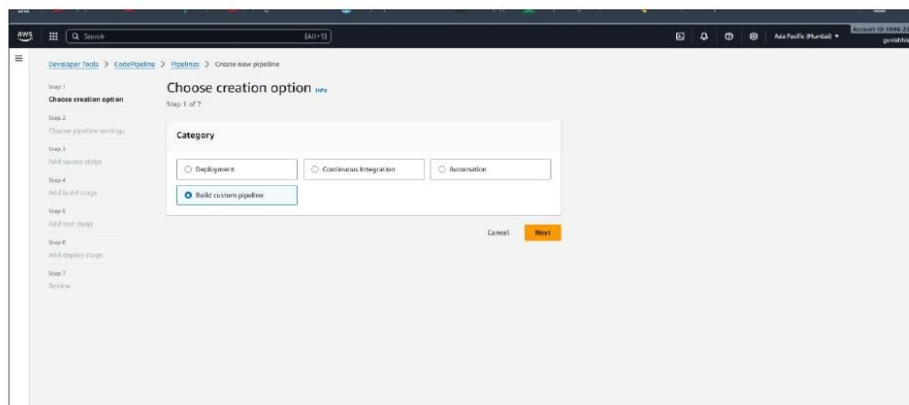continue to use the service as normal. Learn more 🗗

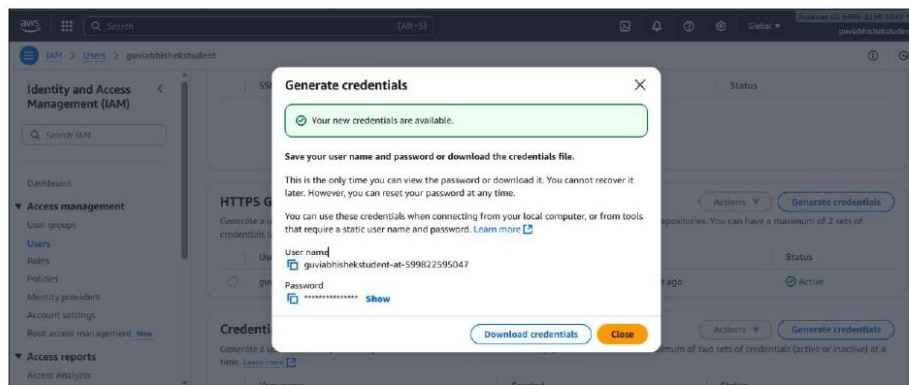## Create repository

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

```
AWS-CI-CD
```

100 characters maximum. Other limits apply.

Description - *optional*

1,000 characters maximum

Tags

[Add tag]

▶ **Additional configuration**
AWS KMS key

                   Cancel   [Create]

HTTPS   SSH   HTTPS (GRC)

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.ap-south-1.amazonaws.com/v1/repos/AWS-CI-CD
```
   [Copy 🗗]

Additional details

You can find more detailed instructions in the documentation. View documentation 🗗

AWS-CI-CD Info   [Add file ▼]

Name