Deploy a simple Nginx application using AWS code commit and deploy & access via browser

```
PS C:\Users\USER\Downloads\Nginxapp> git branch
* master
PS C:\Users\USER\Downloads\Nginxapp> git push -u origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 1.04 KiB | 266.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Validating objects: 100%
To https://git-codecommit.ap-south-1.amazonaws.com/v1/repos/Nginxapp
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\Users\USER\Downloads\Nginxapp>
```

Nginxapp

Reference

△ Notify ▼   master ▼   Create pull request   Clone URL ▼

Nginxapp Info   Add file ▼

	Name
🗀	scripts
🗎	appspec.yml
🗎	buildspec.yml
🗎	index.html

Create build project

Project configuration

Project name

Nginxbuild

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

Source

Add source

Source 1 - Primary

Source provider

AWS CodeCommit ▼

Repository

🔍 Nginxapp ✕

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.

🔍

Source version Info

refs/heads/master

524fbf74 Pipeline

▶ **Additional configuration**
Git clone depth, Git submodules

Environment

Provisioning model Info ↗

● On-demand	○ Reserved capacity
Automatically provision build infrastructure in response to new builds.	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	○ Custom image
Use an image managed by AWS CodeBuild	Specify a Docker image

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-Nginxbuild-service-role

Type your service role name

▶ **Additional configuration**
Timeout, privileged, certificate, VPC, compute type, environment variables, file systems

Buildspec

Build specifications

Insert build commands	Use a buildspec file
○ Store build commands as build project configuration	● 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

🔍 aws2002   ✕

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.

artifact

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

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

artifact.zip

Example: MyPath/MyArtifact.zip.

Namespace type - *optional*

None   ▼

Choose Build ID to insert the build ID into the path to the build output ZIP file or folder, e.g. MyPath/MyBuildID/MyArtifact.zip. Otherwise, choose None.

Artifacts packaging

⚪ None
   The artifact files will be uploaded to the bucket.

🔵 Zip
   AWS CodeBuild will upload artifacts into a compressed file that is put into the specified bucket.

⊘ **Project created**
You have successfully created the following project: Nginxbuild

Create a notification rule for this project   ✕

Developer Tools > CodeBuild > Build projects > Nginxbuild

Nginxbuild

Actions ▼ | Create trigger | Edit | Clone | Debug build | Start build with overrides | **Start build**

Configuration

Source provider	Primary repository	Artifacts upload location	Service role
AWS CodeCommit	Nginxapp	aws2002	arn:aws:iam::558115454328:role/service-role/codebuild-Nginxbuild-service-role

Public builds
Disabled

Nginxbuild:53f8a370-b5b0-4ca1-836d-d9b3bc1b7729

Stop build   **Retry build**

Build status

Status	Initiator	Build ARN	Resolved source version
⊘ Succeeded	root	🗇 arn:aws:codebuild:ap-south-1:558115454328:build/Nginxbuild:53f8a370-b5b0-4ca1-836d-d9b3bc1b7729	524fbf74794ecc8cbeab23de26e06b3bbb4767b5

Start time	End time	Build number
May 7, 2024 10:24 AM (UTC+5:30)	May 7, 2024 10:25 AM (UTC+5:30)	1

Build logs   **Phase details**   Reports   Environment variables   Build details   Resource utilization

Name	Status	Context	Duration	Start time	End time
SUBMITTED	⊘ Succeeded	-	<1 sec	May 7, 2024 10:24 AM (UTC+5:30)	May 7, 2024 10:24 AM (UTC+5:30)
QUEUED	⊘ Succeeded	-	<1 sec	May 7, 2024 10:24 AM (UTC+5:30)	May 7, 2024 10:24 AM (UTC+5:30)
PROVISIONING	⊘ Succeeded	-	3 secs	May 7, 2024 10:24 AM (UTC+5:30)	May 7, 2024 10:24 AM (UTC+5:30)
DOWNLOAD_SOURCE	⊘ Succeeded	-	6 secs	May 7, 2024 10:24 AM (UTC+5:30)	May 7, 2024 10:24 AM (UTC+5:30)
INSTALL	⊘ Succeeded	-	33 secs	May 7, 2024 10:24 AM (UTC+5:30)	May 7, 2024 10:25 AM (UTC+5:30)
PRE_BUILD	⊘ Succeeded	-	<1 sec	May 7, 2024 10:25 AM (UTC+5:30)	May 7, 2024 10:25 AM (UTC+5:30)

⊘ **Application created**
In order to create a new deployment, you must first create a deployment group.

Create a notification rule for this application   ✕

Developer Tools > CodeDeploy > Applications > Nginx-dg

Nginx-dg

🔔 Notify ▾   Delete application

Application details

Name	Compute platform
Nginx-dg	EC2/on-premises

Create deployment group

Application

Application
Nginx-dg
Compute type
EC2/on-premises

Deployment group name

Enter a deployment group name

Nginx-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::558115454328:role/codedeployrule ✕

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

🔍 Name   ✕

Value - *optional*

🔍 aws_nginx   ✕   Remove tag

Add tag

＋ Add tag group

☐ On-premises instances

Matching instances

1 unique matched instance. Click here for details ⬈

```
ubuntu@ip-172-31-39-125:~$ systemctl status codedeploy-agent
● codedeploy-agent.service - LSB: AWS CodeDeploy Host Agent
     Loaded: loaded (/etc/init.d/codedeploy-agent; generated)
     Active: active (running) since Tue 2024-05-07 05:25:35 UTC; 6min ago
       Docs: man:systemd-sysv-generator(8)
      Tasks: 2 (limit: 1121)
     Memory: 56.8M
        CPU: 1.179s
     CGroup: /system.slice/codedeploy-agent.service
             ├─2590 "codedeploy-agent: master 2590" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
             └─2592 "codedeploy-agent: InstanceAgent::Plugins::CodeDeployPlugin::CommandPoller of master 2590"

May 07 05:25:35 ip-172-31-39-125 systemd[1]: Starting LSB: AWS CodeDeploy Host Agent...
May 07 05:25:35 ip-172-31-39-125 codedeploy-agent[2584]: Starting codedeploy-agent:
May 07 05:25:35 ip-172-31-39-125 systemd[1]: Started LSB: AWS CodeDeploy Host Agent.
ubuntu@ip-172-31-39-125:~$ sudo systemctl restart codedeploy-agent
ubuntu@ip-172-31-39-125:~$ systemctl status codedeploy-agent
● codedeploy-agent.service - LSB: AWS CodeDeploy Host Agent
     Loaded: loaded (/etc/init.d/codedeploy-agent; generated)
     Active: active (running) since Tue 2024-05-07 05:32:33 UTC; 7s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 5063 ExecStart=/etc/init.d/codedeploy-agent start (code=exited, status=0/SUCCESS)
      Tasks: 2 (limit: 1121)
     Memory: 63.0M
        CPU: 1.155s
     CGroup: /system.slice/codedeploy-agent.service
             ├─5070 "codedeploy-agent: master 5070" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ""
             └─5072 "codedeploy-agent: InstanceAgent::Plugins::CodeDeployPlugin::CommandPoller of master 5070"

May 07 05:32:33 ip-172-31-39-125 systemd[1]: Stopped LSB: AWS CodeDeploy Host Agent.
May 07 05:32:33 ip-172-31-39-125 systemd[1]: codedeploy-agent.service: Consumed 1.698s CPU time.
May 07 05:32:33 ip-172-31-39-125 systemd[1]: Starting LSB: AWS CodeDeploy Host Agent...
May 07 05:32:33 ip-172-31-39-125 codedeploy-agent[5063]: Starting codedeploy-agent:
May 07 05:32:33 ip-172-31-39-125 systemd[1]: Started LSB: AWS CodeDeploy Host Agent.
ubuntu@ip-172-31-39-125:~$
```

Nginx-group

Edit   Delete   **Create deployment**

Deployment group details

Deployment group name	Application name	Compute platform
Nginx-group	Nginx-dg	EC2/on-premises
Deployment type	Service role ARN	Deployment configuration
In place	arn:aws:iam::558115454328:role/codedeployrule	CodeDeployDefault.AllAtOnce
Rollback enabled	Agent update scheduler	
False	Learn to schedule update in AWS Systems Manager ↗	

Environment configuration: Amazon EC2 instances

Key	Value
Name	aws_nginx

Create deployment

Deployment settings

Application

Nginx-dg

Deployment group

🔍 Nginx-group ✕

Compute platform

EC2/on-premises

Deployment type

In place

createDeploymentGroup.formSection.managedHookRole.label
createDeployment.formSection.managedHookRole.description createDeployment.formSection.managedHookRole.updateDeploymentGroup

-

Revision type

● My application is stored in Amazon S3	○ My application is stored in GitHub

Revision location

Copy and paste the Amazon S3 bucket where your revision is stored

🔍 s3://aws2002/artifact.zip/artifact ✕

---

d-RP3BJ70U4

↻   Copy deployment   **Retry deployment**

Deployment status

Installing application on your instances

━━━━━━━━━━━━━━━━━━━━━━━━━ 100%

1 of 1 instances updated ⊘ Succeeded

Deployment details

Application	Deployment ID	Status
Nginx-dg	d-RP3BJ70U4	⊘ Succeeded
Deployment configuration	Deployment group	Initiated by
CodeDeployDefault.AllAtOnce	Nginx-group	User action
Deployment description		
-		

Event	Duration	Status	Error code	Start time	End time
ApplicationStop	less than one second	⊘ Succeeded	-	May 7, 2024 11:09 AM (UTC+5:30)	May 7, 2024 11:09 AM (UTC+5:30)
DownloadBundle	less than one second	⊘ Succeeded	-	May 7, 2024 11:09 AM (UTC+5:30)	May 7, 2024 11:09 AM (UTC+5:30)
BeforeInstall	less than one second	⊘ Succeeded	-	May 7, 2024 11:09 AM (UTC+5:30)	May 7, 2024 11:09 AM (UTC+5:30)
Install	less than one second	⊘ Succeeded	-	May 7, 2024 11:09 AM (UTC+5:30)	May 7, 2024 11:09 AM (UTC+5:30)
AfterInstall	10 seconds	⊘ Succeeded	-	May 7, 2024 11:09 AM (UTC+5:30)	May 7, 2024 11:09 AM (UTC+5:30)
ApplicationStart	1 second	⊘ Succeeded	-	May 7, 2024 11:09 AM (UTC+5:30)	May 7, 2024 11:09 AM (UTC+5:30)
ValidateService	less than one second	⊘ Succeeded	-	May 7, 2024 11:09 AM (UTC+5:30)	May 7, 2024 11:09 AM (UTC+5:30)

← → C ⚠ Not secure 13.127.29.35   ☆ ⧉ 👤 ⋮

Hello Devops