CI/CD flow for Frontend Application

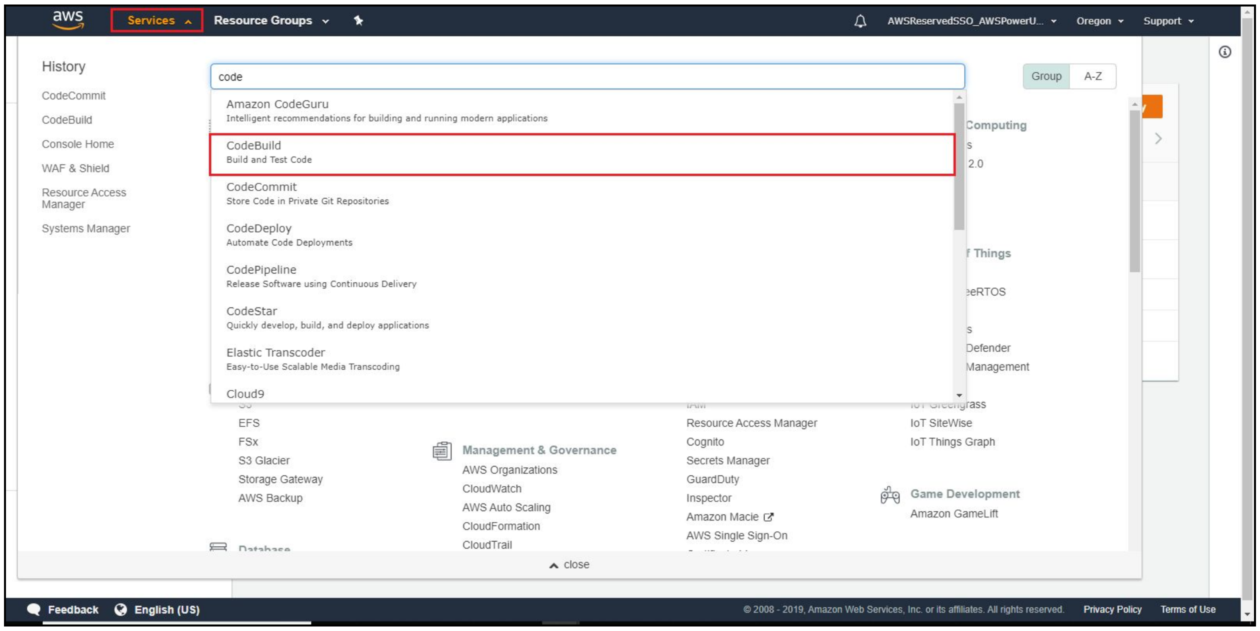
Agenda:

CodeBuild: AWS code build is a fully managed continuous integration service that compiles source code, runs tests, and produces software packages that are ready to deploy. With code build, you don't need to provision, manage, and scale your own build servers. code build scales continuously and processes multiple builds concurrently, so your builds are not left waiting in a queue. You can get started quickly by using prepackaged build environments, or you can create custom build environments that use your own build tools.

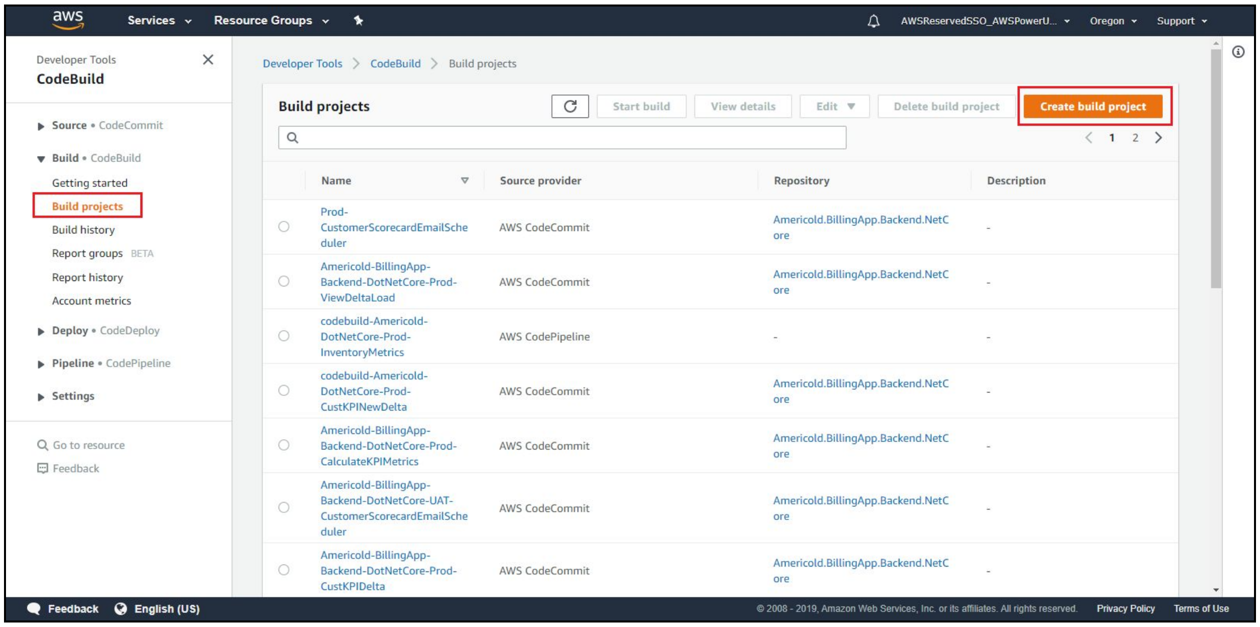
CodePipeline: AWS CodePipeline is a fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates. CodePipeline automates the build, test, and deploy phases of your release process every time there is a code change, based on the release model you define. This enables you to rapidly and reliably deliver features and updates. You can easily integrate the AWS code pipeline with third-party services such as git hub or with your own custom plugin.

By the end of the document, we learn how to create a CI-CD pipeline for angular applications.

First, we need to login to the AWS console and search for **CodeBuild** service.

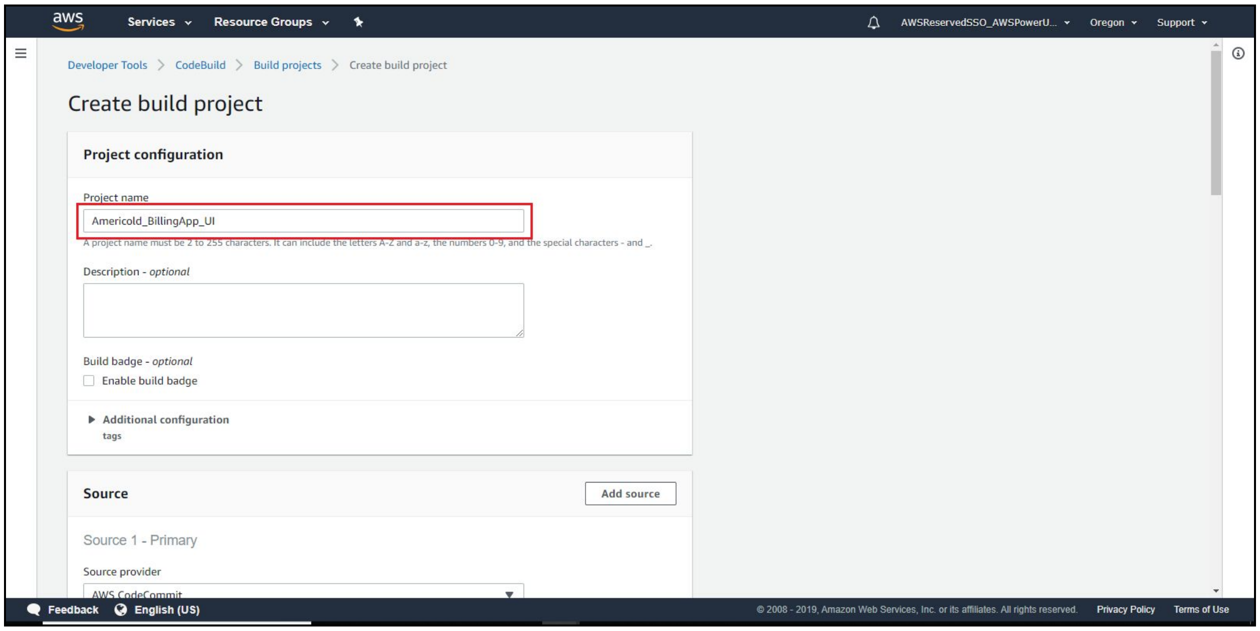

Services
Resource Groups
History
CodeCommit
CodeBuild
Console Home
WAF & Shield
Resource Access Manager
Systems Manager
code
Amazon CodeGuru
Intelligent recommendations for building and running modern applications
CodeBuild
Build and Test Code
CodeCommit
Store Code in Private Git Repositories
CodeDeploy
Automate Code Deployments
CodePipeline
Release Software using Continuous Delivery
CodeStar
Quickly develop, build, and deploy applications
Elastic Transcoder
Easy-to-Use Scalable Media Transcoding
Cloud9
Group
A-Z
close
Feedback
English (US)

Go to build projects and click on **create build project** to create a new build project for integration**.**



Name	Source provider	Repository	Description
Prod-CustomerScorecardEmailScheduler	AWS CodeCommit	Americold.BillingApp.Backend.NetCore	-
Americold-BillingApp-Backend-DotNetCore-Prod-ViewDeltaLoad	AWS CodeCommit	Americold.BillingApp.Backend.NetCore	-
codebuild-Americold-DotNetCore-Prod-InventoryMetrics	AWS CodePipeline	-	-
codebuild-Americold-DotNetCore-Prod-CustKPINewDelta	AWS CodeCommit	Americold.BillingApp.Backend.NetCore	-
Americold-BillingApp-Backend-DotNetCore-Prod-CalculateKPIMetrics	AWS CodeCommit	Americold.BillingApp.Backend.NetCore	-
Americold-BillingApp-Backend-DotNetCore-UAT-CustomerScorecardEmailScheduler	AWS CodeCommit	Americold.BillingApp.Backend.NetCore	-
Americold-BillingApp-Backend-DotNetCore-Prod-CustKPIDelta	AWS CodeCommit	Americold.BillingApp.Backend.NetCore	-

Enter the name of the project**.**


aws
Services
Resource Groups
AWSReservedSSO_AWSPowerU...
Oregon
Support
Developer Tools > CodeBuild > Build projects > Create build project
Add source
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Create build project

Project configuration

Project name: Americold_BillingApp_UI

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 _.

Description - *optional*

Build badge - *optional*

☐ Enable build badge

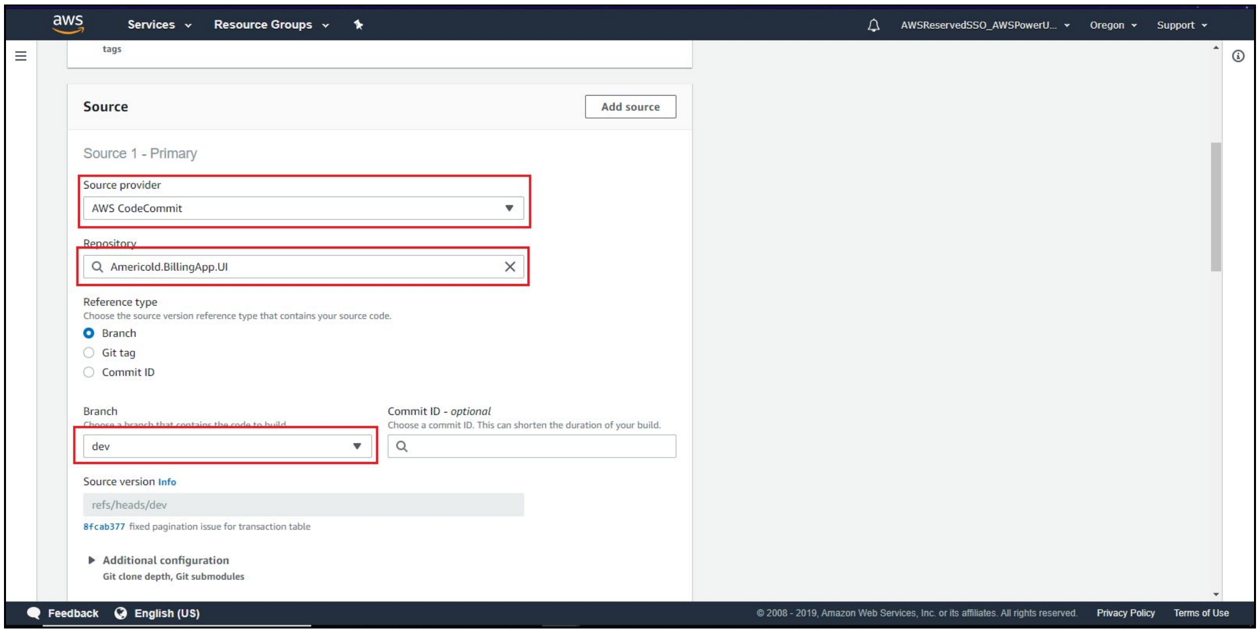
Additional configuration
tags

Source

Source 1 - Primary

Source provider: AWS CodeCommit

Under the source section, select the source provider as **AWS CodeCommit**, select the repository and branch as shown below.


aws
Services
Resource Groups
AWSReservedSSO_AWSPowerU...
Oregon
Support
tags
Add source
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Source

Source 1 - Primary

Source provider: AWS CodeCommit

Repository: Americold.BillingApp.UI

Reference type
Choose the source version reference type that contains your source code.
- [✓ Branch]
- Git tag
- Commit ID

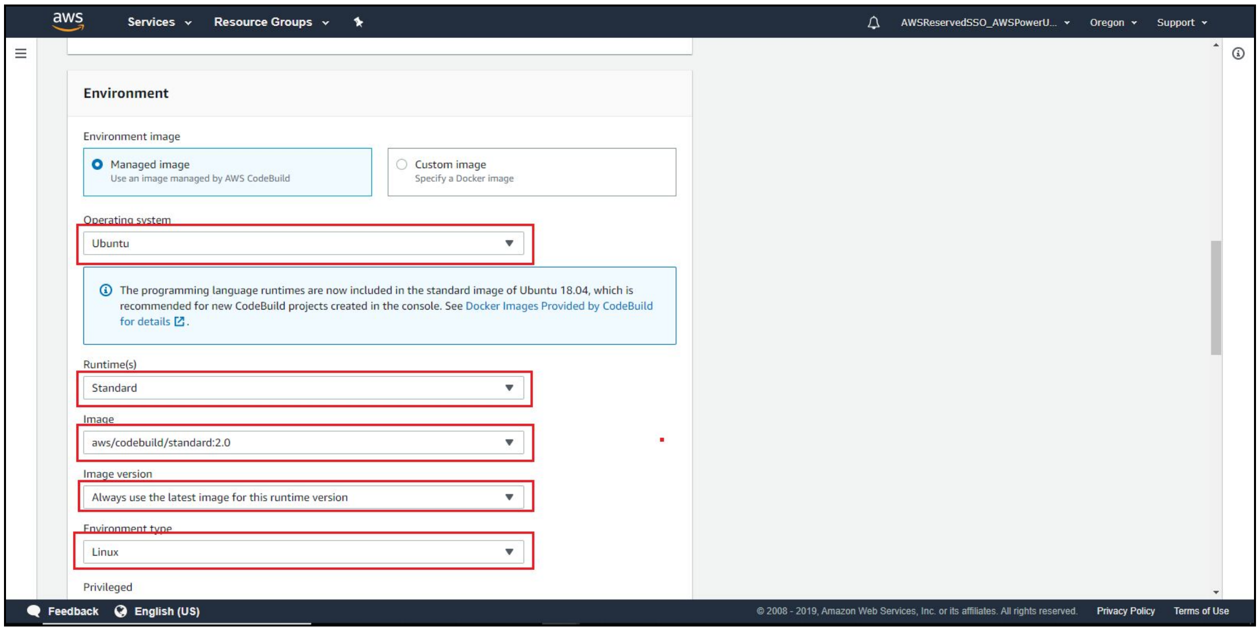
Branch
Choose a branch that contains the code to build
dev

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

Source version Info
refs/heads/dev
8fcab377 fixed pagination issue for transaction table

Additional configuration
Git clone depth, Git submodules

Under the environment section, we took the managed image that is provided by AWS, Operating system as Ubuntu, runtime as standard, Image as standard 2.0 version, image version as always latest, environment type as Linux and created new service role for the code build.


aws
Services
Resource Groups
AWSReservedSSO_AWSPowerU...
Oregon
Support
Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Environment

Environment image
- [✓ Managed image] Use an image managed by AWS CodeBuild
- Custom image Specify a Docker image

Operating system: Ubuntu

The programming language runtimes are now included in the standard image of Ubuntu 18.04, which is recommended for new CodeBuild projects created in the console. See Docker Images Provided by CodeBuild for details.

Runtime(s): Standard

Image: aws/codebuild/standard:2.0

Image version: Always use the latest image for this runtime version

Environment type: Linux

Privileged

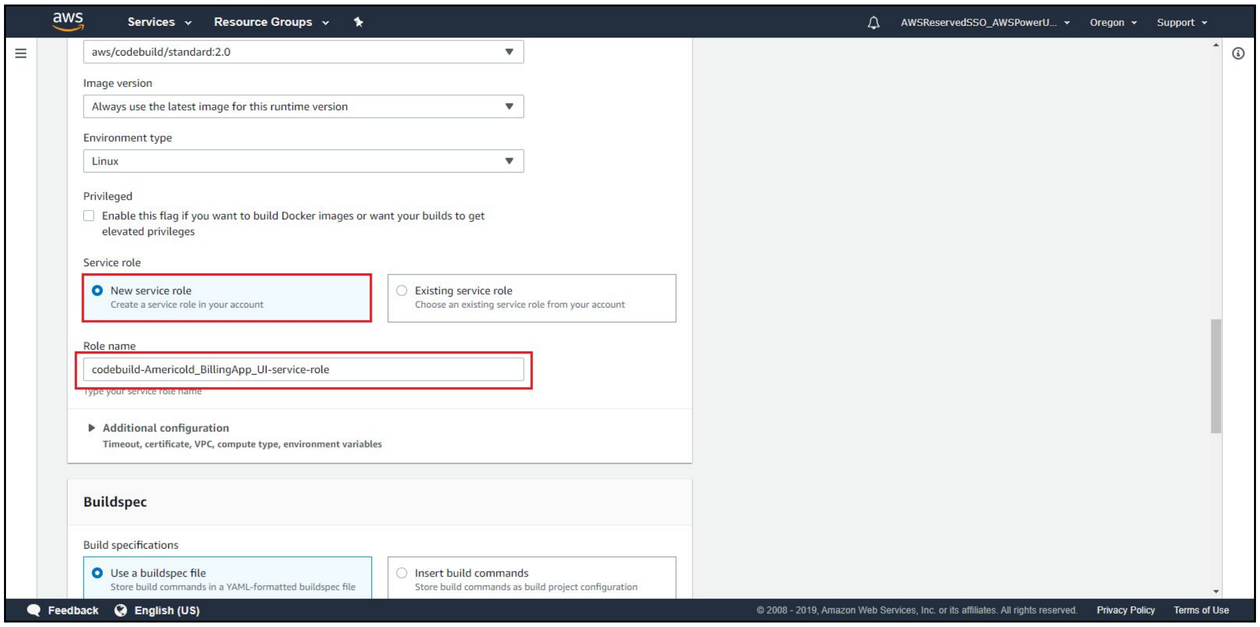


Image: aws/codebuild/standard:2.0

Image version: Always use the latest image for this runtime version

Environment type: Linux

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

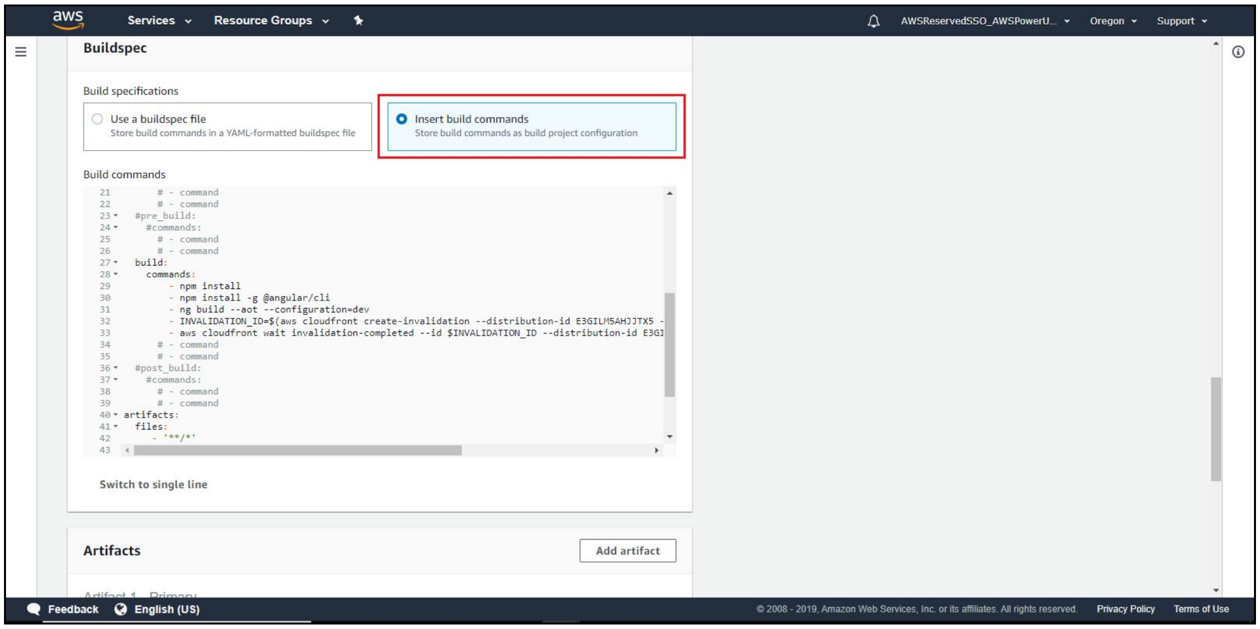
Role name: codebuild-Americold_BillingApp_UI-service-role
Type your service role name

Additional configuration
Timeout, certificate, VPC, compute type, environment variables

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

Under the build spec section, choose an option as **insert build commands** to edit the **buildspec.yml** file from console. buildspec.yml file has a set of commands for integration and deployment.



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

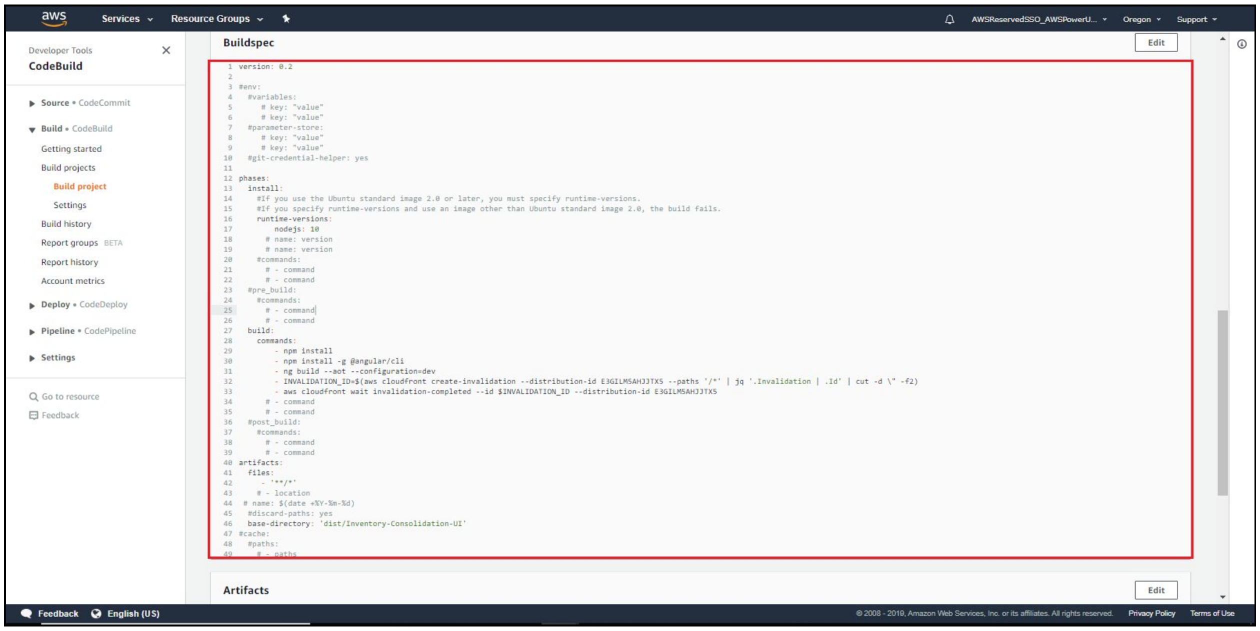
Build commands

```
      # - command
      # - command
  #pre_build:
    #commands:
      # - command
      # - command
  build:
    commands:
        - npm install
        - npm install -g @angular/cli
        - ng build --aot --configuration=dev
        - INVALIDATION_ID=$(aws cloudfront create-invalidation --distribution-id E3GILM5AHJJTX5 -
        - aws cloudfront wait invalidation-completed --id $INVALIDATION_ID --distribution-id E3GI
      # - command
      # - command
  #post_build:
    #commands:
      # - command
      # - command
artifacts:
  files:
    - '**/*'
```

Switch to single line

Artifacts

Configure the respective stages as per requirement in the **buildspec.yml file** as shown below.



```
version: 0.2

#env:
  #variables:
     # key: "value"
     # key: "value"
  #parameter-store:
     # key: "value"
     # key: "value"
  #git-credential-helper: yes

phases:
  install:
    #If you use the Ubuntu standard image 2.0 or later, you must specify runtime-versions.
    #If you specify runtime-versions and use an image other than Ubuntu standard image 2.0, the build fails.
    runtime-versions:
        nodejs: 10
      # name: version
      # name: version
    #commands:
      # - command
      # - command
  #pre_build:
    #commands:
      # - command
      # - command
  build:
    commands:
        - npm install
        - npm install -g @angular/cli
        - ng build --aot --configuration=dev
        - INVALIDATION_ID=$(aws cloudfront create-invalidation --distribution-id E3GILM5AHJJTX5 --paths '/*' | jq '.Invalidation | .Id' | cut -d \" -f2)
        - aws cloudfront wait invalidation-completed --id $INVALIDATION_ID --distribution-id E3GILM5AHJJTX5
      # - command
      # - command
  #post_build:
    #commands:
      # - command
      # - command
artifacts:
  files:
     - '**/*'
    # - location
 # name: $(date +%Y-%m-%d)
  #discard-paths: yes
  base-directory: 'dist/Inventory-Consolidation-UI'
#cache:
  #paths:
    # - paths
```

Below is the step by step explanation for the above script:

Install stage:

- **runtime-versions:**

  nodejs: 10: It defines the runtime environment and version for the application

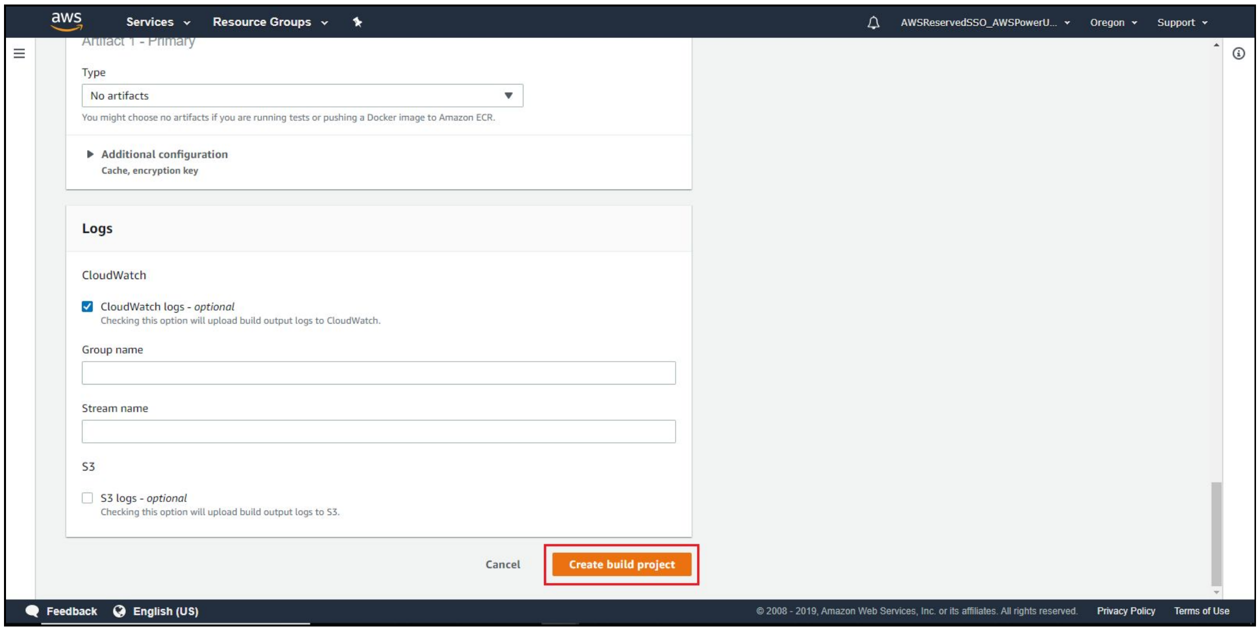
Build Stage:

- **npm install:** Install node modules using the following command
- **npm install -g @angular/cli:** Install angular cli using the following command
- **ng build --aot --configuration=dev:** Generate build artifact using the following command
- **INVALIDATION_ID=$(aws cloudfront create-invalidation --distribution-id E3GILM5AHJJTX5 --paths '/*' | jq '.Invalidation | .Id' | cut -d \" -f2) :** Create invalidation for cloudfront distribution to clear cache
- **aws cloudfront wait invalidation-completed --id $INVALIDATION_ID --distribution-id E3GILM5AHJJTX5:** Wait for invalidation to be completed

Artifacts:

- **'**/*':** To make all files into an Artifact use this regular expression
- **base-directory: 'dist/Inventory-Consolidation-UI':** Mention the artifact path as base-directory

We are enabling cloud watch logs to check the build logs. Once all the configurations were done, click on **create build project.**



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.

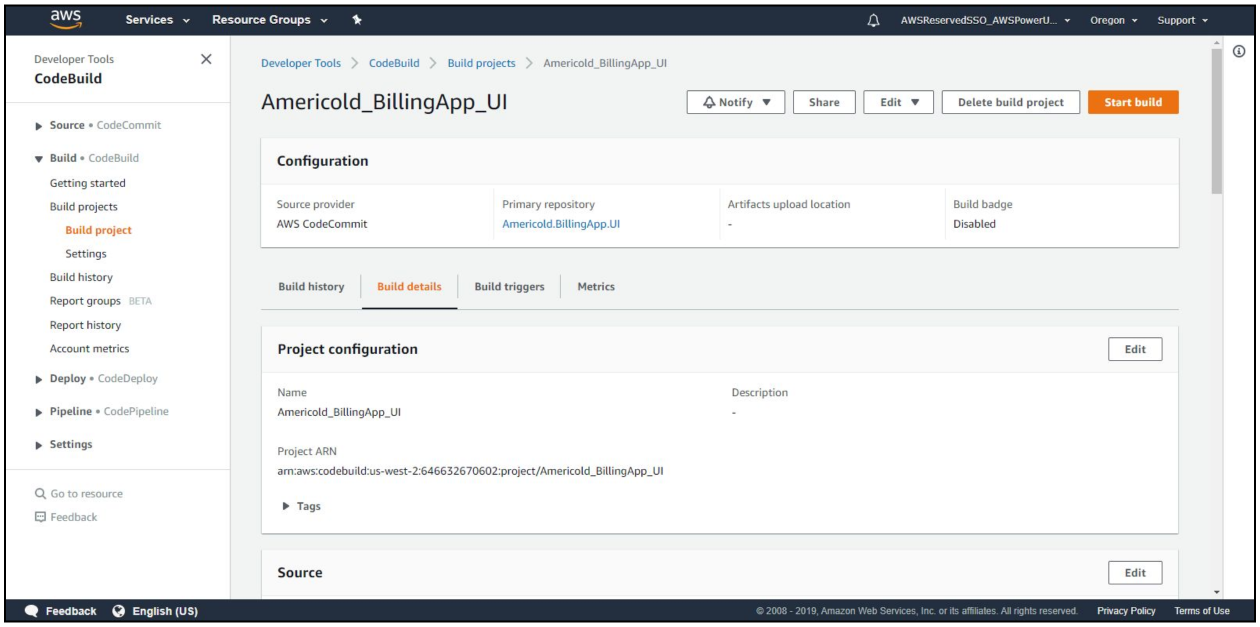
Group name

Stream name

S3

[ ] S3 logs - *optional*
Checking this option will upload build output logs to S3.

The new build project has been created and the dashboard is shown below.



Americold_BillingApp_UI

Configuration

Source provider	Primary repository	Artifacts upload location	Build badge
AWS CodeCommit	Americold.BillingApp.UI	-	Disabled

Project configuration

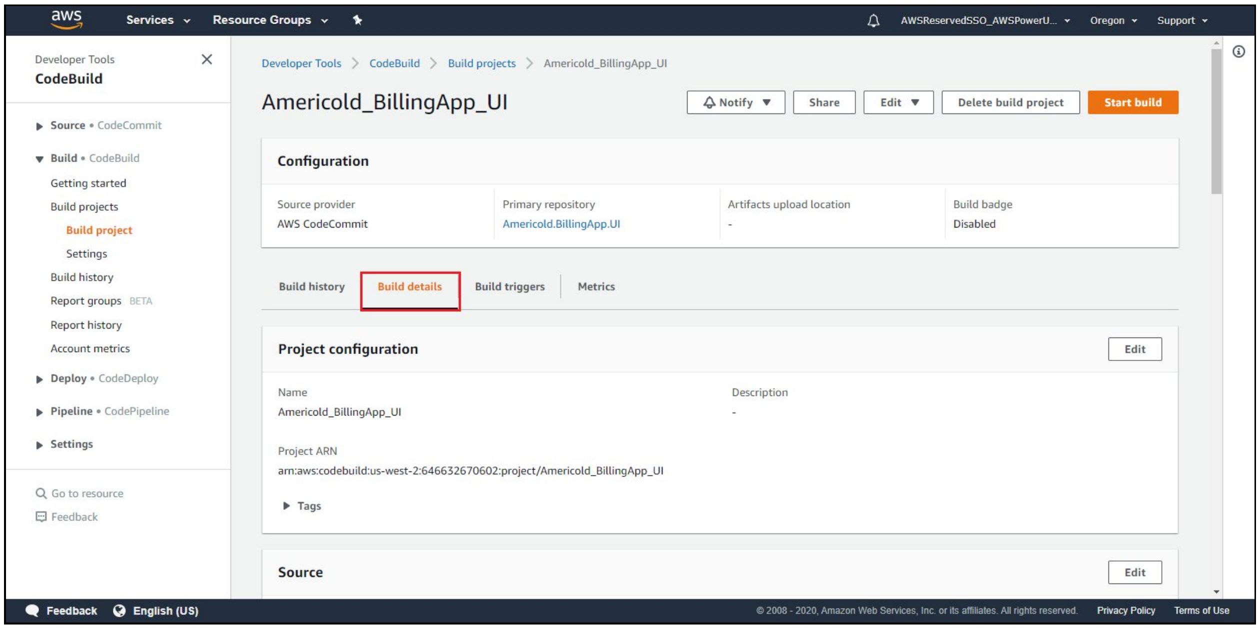
Name
Americold_BillingApp_UI

Description
-

Project ARN
arn:aws:codebuild:us-west-2:646632670602:project/Americold_BillingApp_UI

Source

Goto details and goto the **environment section** in build details.


Developer Tools > CodeBuild > Build projects > Americold_BillingApp_UI

Americold_BillingApp_UI

Configuration

Source provider	Primary repository	Artifacts upload location	Build badge
AWS CodeCommit	Americold.BillingApp.UI	-	Disabled

Build history | Build details | Build triggers | Metrics

Project configuration

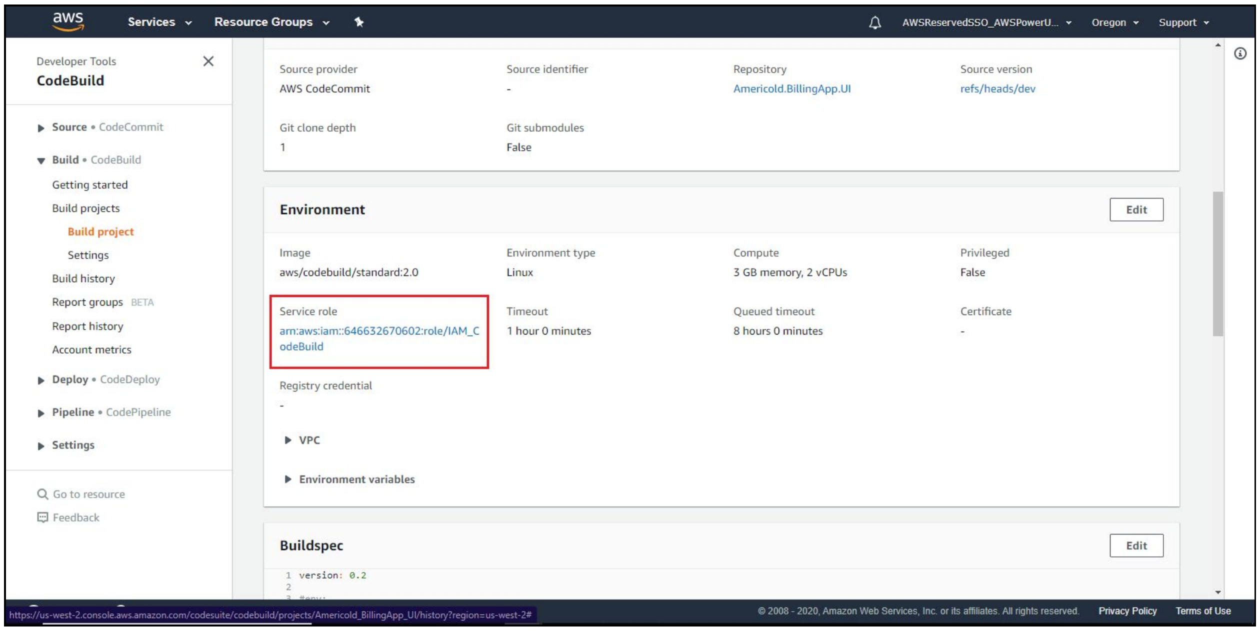
Name: Americold_BillingApp_UI

Description: -

Project ARN: arn:aws:codebuild:us-west-2:646632670602:project/Americold_BillingApp_UI

Source

Click on service role which is a new service role created while configuring code build. It will redirect us to the **IAM console**.


Source provider	Source identifier	Repository	Source version
AWS CodeCommit	-	Americold.BillingApp.UI	refs/heads/dev

Git clone depth	Git submodules
1	False

Environment

Image	Environment type	Compute	Privileged
aws/codebuild/standard:2.0	Linux	3 GB memory, 2 vCPUs	False

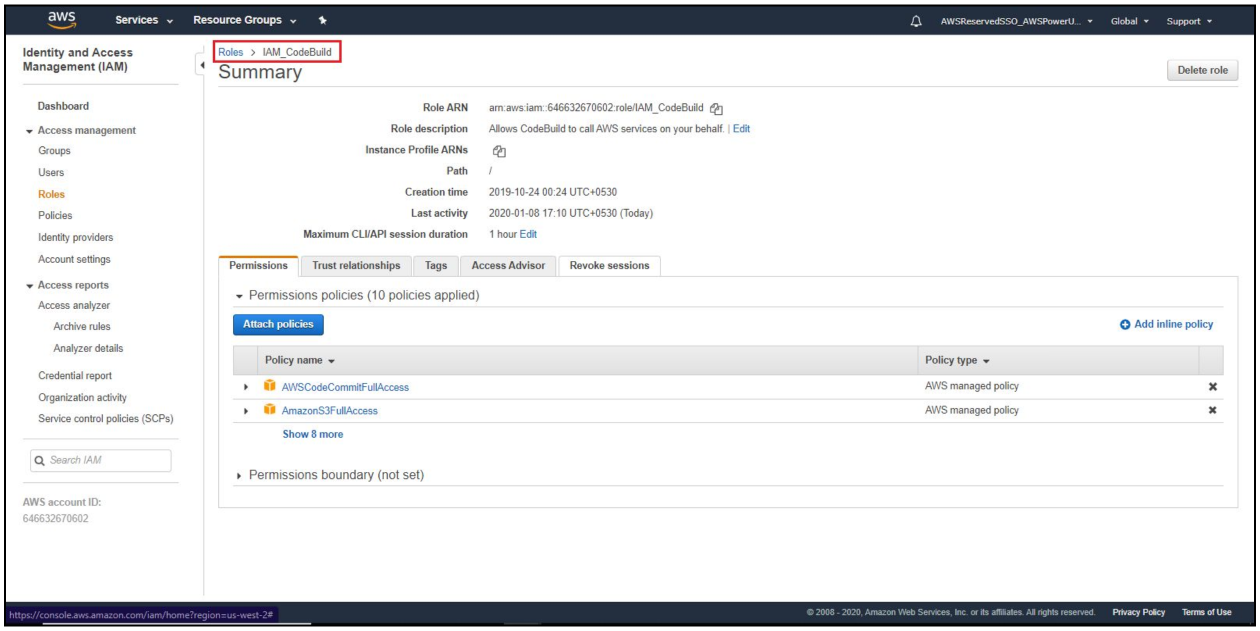
Service role	Timeout	Queued timeout	Certificate
arn:aws:iam::646632670602:role/IAM_CodeBuild	1 hour 0 minutes	8 hours 0 minutes	-

Registry credential: -

Buildspec

```
version: 0.2
```

The IAM role dashboard will be as shown below and we need to add the required policies to access the AWS services for application deployment through code build.



AWS account ID: 646632670602

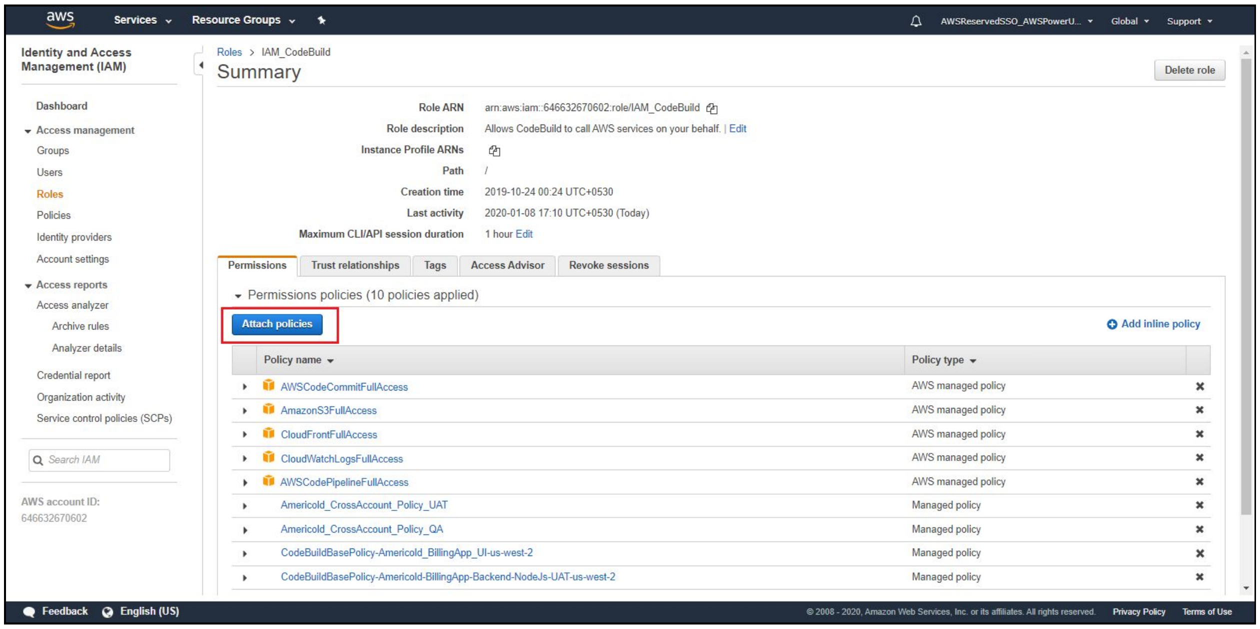
Summary

Role ARN	arn:aws:iam::646632670602:role/IAM_CodeBuild
Role description	Allows CodeBuild to call AWS services on your behalf. | Edit
Instance Profile ARNs	
Path	/
Creation time	2019-10-24 00:24 UTC+0530
Last activity	2020-01-08 17:10 UTC+0530 (Today)
Maximum CLI/API session duration	1 hour Edit

Permissions policies (10 policies applied)

Policy name	Policy type
AWSCodeCommitFullAccess	AWS managed policy
AmazonS3FullAccess	AWS managed policy

Click on **attach policies** to add the policies for the IAM role.



AWS account ID: 646632670602

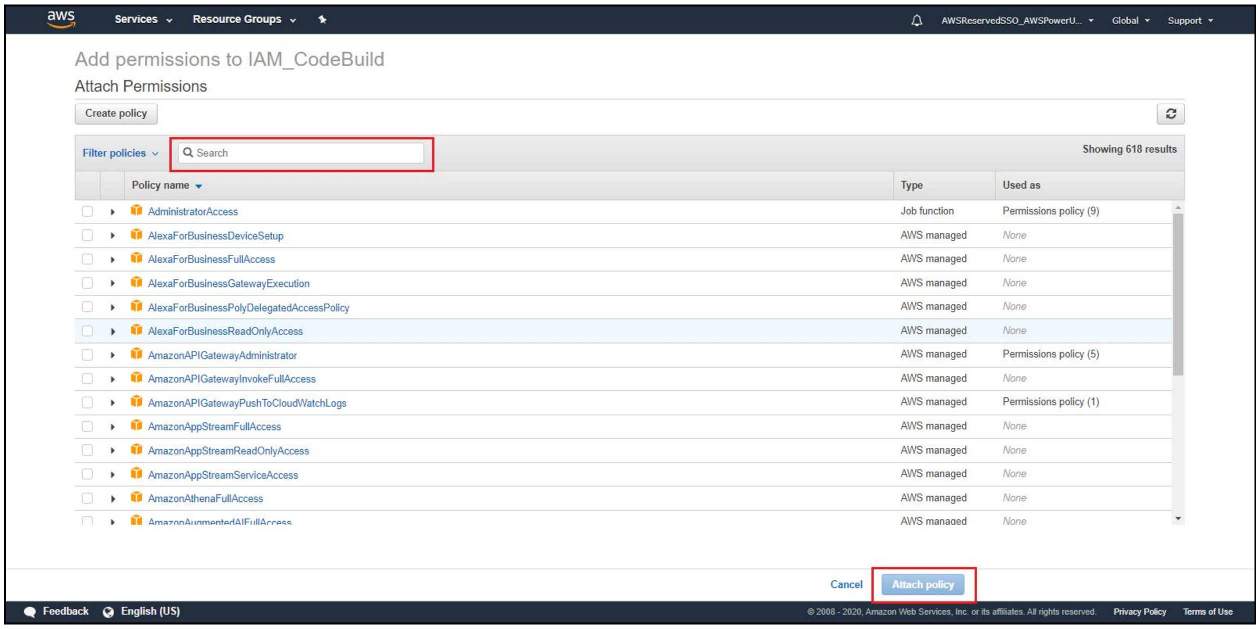
Summary

Role ARN	arn:aws:iam::646632670602:role/IAM_CodeBuild
Role description	Allows CodeBuild to call AWS services on your behalf. | Edit
Instance Profile ARNs	
Path	/
Creation time	2019-10-24 00:24 UTC+0530
Last activity	2020-01-08 17:10 UTC+0530 (Today)
Maximum CLI/API session duration	1 hour Edit

Permissions policies (10 policies applied)

Policy name	Policy type
AWSCodeCommitFullAccess	AWS managed policy
AmazonS3FullAccess	AWS managed policy
CloudFrontFullAccess	AWS managed policy
CloudWatchLogsFullAccess	AWS managed policy
AWSCodePipelineFullAccess	AWS managed policy
Americold_CrossAccount_Policy_UAT	Managed policy
Americold_CrossAccount_Policy_QA	Managed policy
CodeBuildBasePolicy-Americold_BillingApp_UI-us-west-2	Managed policy
CodeBuildBasePolicy-Americold-BillingApp-Backend-NodeJs-UAT-us-west-2	Managed policy

Search for the required policy and select the checkboxes and click on attach policies.



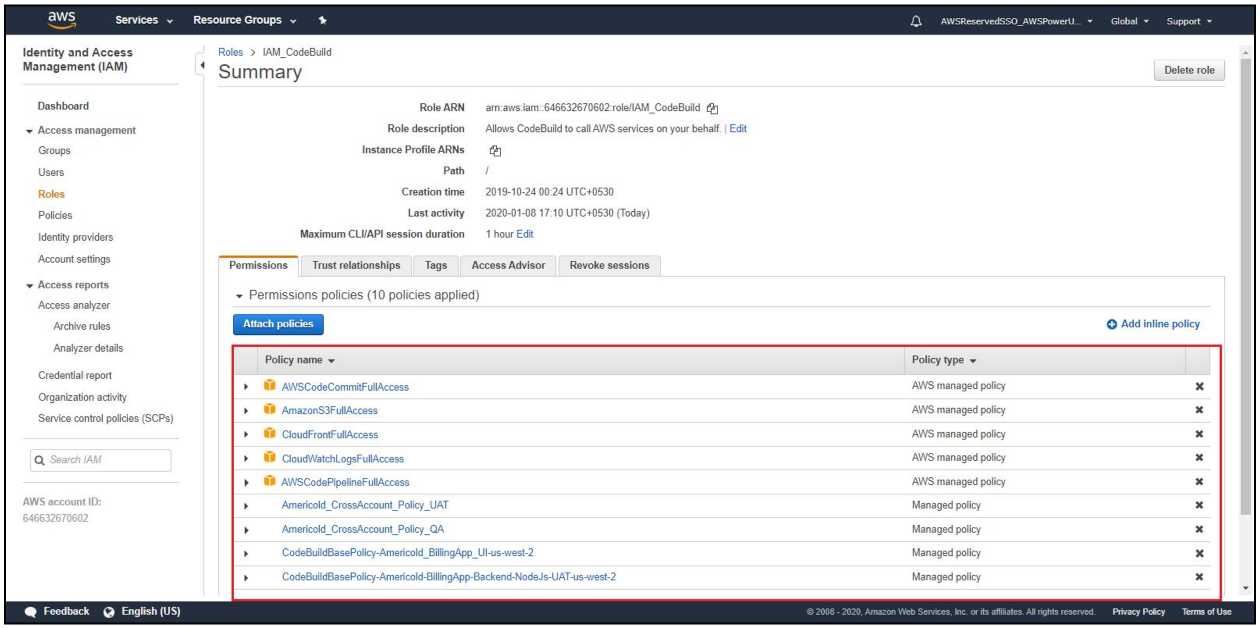
Add permissions to IAM_CodeBuild

Attach Permissions

Filter policies · Search · Showing 618 results

Policy name	Type	Used as
AdministratorAccess	Job function	Permissions policy (9)
AlexaForBusinessDeviceSetup	AWS managed	None
AlexaForBusinessFullAccess	AWS managed	None
AlexaForBusinessGatewayExecution	AWS managed	None
AlexaForBusinessPolyDelegatedAccessPolicy	AWS managed	None
AlexaForBusinessReadOnlyAccess	AWS managed	None
AmazonAPIGatewayAdministrator	AWS managed	Permissions policy (5)
AmazonAPIGatewayInvokeFullAccess	AWS managed	None
AmazonAPIGatewayPushToCloudWatchLogs	AWS managed	Permissions policy (1)
AmazonAppStreamFullAccess	AWS managed	None
AmazonAppStreamReadOnlyAccess	AWS managed	None
AmazonAppStreamServiceAccess	AWS managed	None
AmazonAthenaFullAccess	AWS managed	None
AmazonAugmentedAIFullAccess	AWS managed	None

Here we have attached the policies for **code commit, S3, cloud front, cloud watch logs, code pipeline, and cross-account policy.**



Roles > IAM_CodeBuild

Summary

Role ARN	arn:aws:iam::646632670602:role/IAM_CodeBuild
Role description	Allows CodeBuild to call AWS services on your behalf. | Edit
Instance Profile ARNs	
Path	/
Creation time	2019-10-24 00:24 UTC+0530
Last activity	2020-01-08 17:10 UTC+0530 (Today)
Maximum CLI/API session duration	1 hour Edit

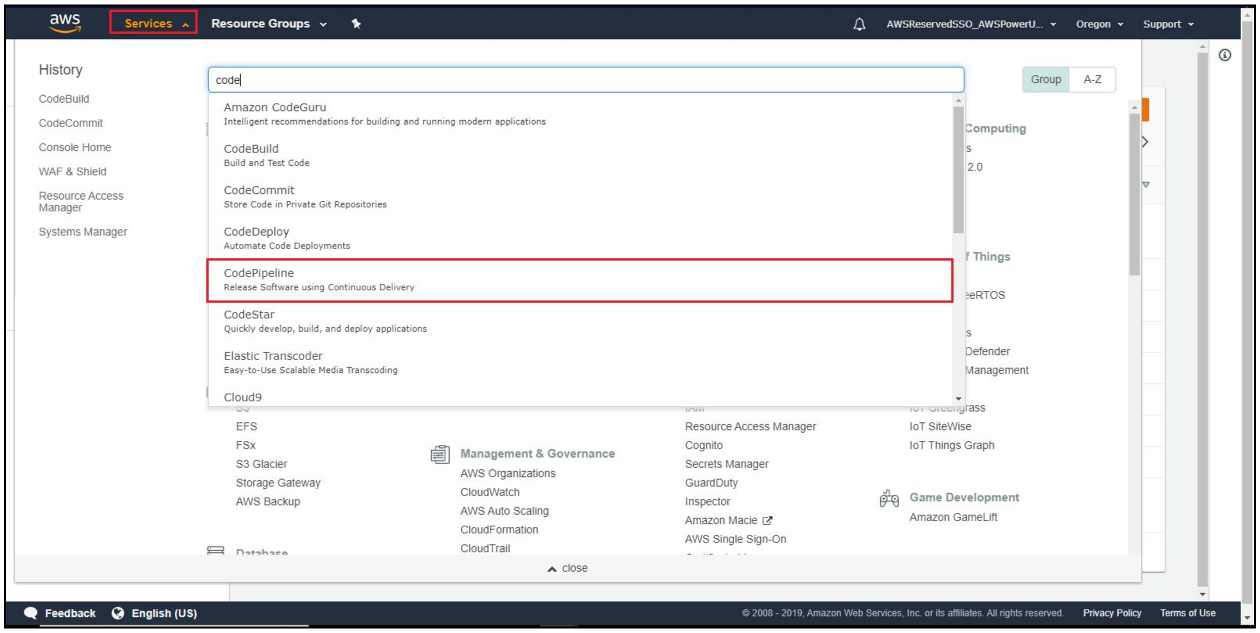
Permissions policies (10 policies applied)

Policy name	Policy type
AWSCodeCommitFullAccess	AWS managed policy
AmazonS3FullAccess	AWS managed policy
CloudFrontFullAccess	AWS managed policy
CloudWatchLogsFullAccess	AWS managed policy
AWSCodePipelineFullAccess	AWS managed policy
Americold_CrossAccount_Policy_UAT	Managed policy
Americold_CrossAccount_Policy_QA	Managed policy
CodeBuildBasePolicy-Americold_BillingApp_UI-us-west-2	Managed policy
CodeBuildBasePolicy-Americold-BillingApp-Backend-NodeJs-UAT-us-west-2	Managed policy

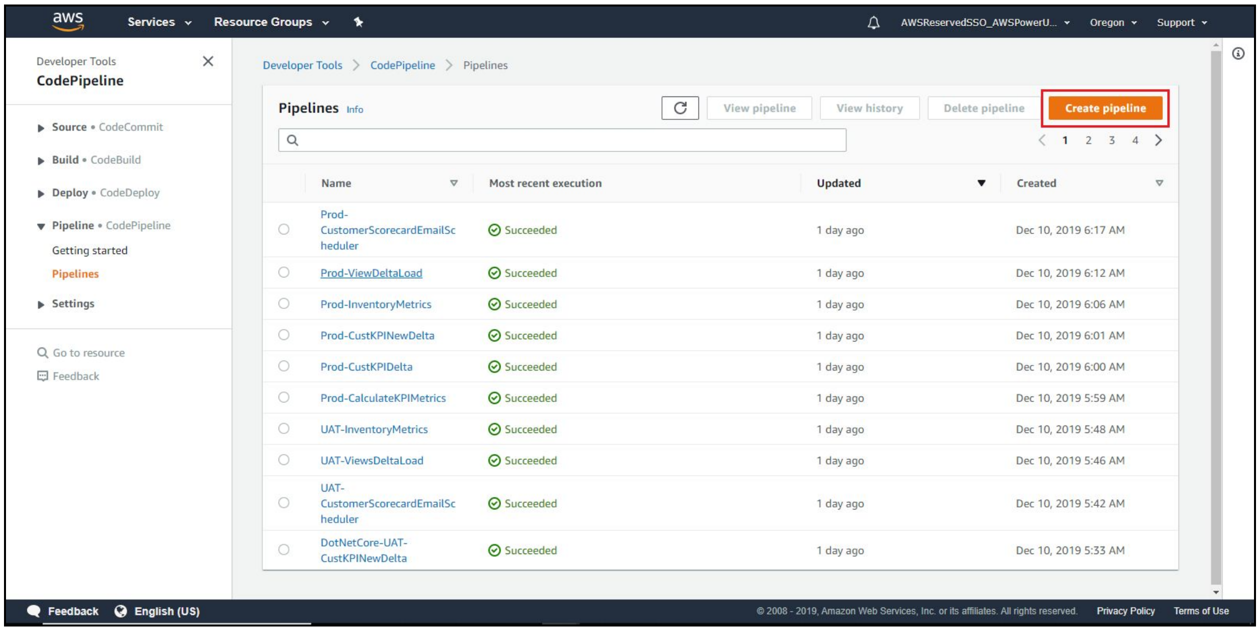
AWS account ID: 646632670602

We are going to automate the process for integration and deployment using CodePipeline.

Goto **CodePipeline service** in AWS console.

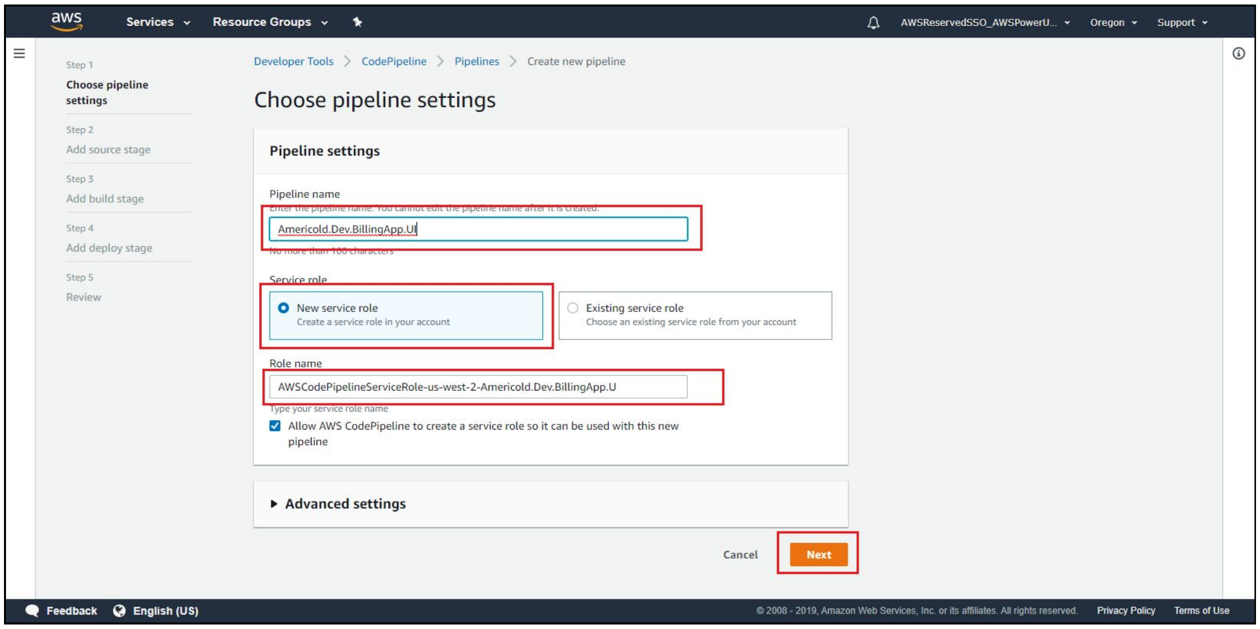

Services
Resource Groups
History
CodeBuild
CodeCommit
Console Home
WAF & Shield
Resource Access Manager
Systems Manager
code
Amazon CodeGuru
Intelligent recommendations for building and running modern applications
CodeBuild
Build and Test Code
CodeCommit
Store Code in Private Git Repositories
CodeDeploy
Automate Code Deployments
CodePipeline
Release Software using Continuous Delivery
CodeStar
Quickly develop, build, and deploy applications
Elastic Transcoder
Easy-to-Use Scalable Media Transcoding
Cloud9
Group
A-Z
close
Feedback
English (US)

Click on **create pipelines** to create a new pipeline for automated CI/CD.



Name	Most recent execution	Updated	Created
Prod-CustomerScorecardEmailScheduler	Succeeded	1 day ago	Dec 10, 2019 6:17 AM
Prod-ViewDeltaLoad	Succeeded	1 day ago	Dec 10, 2019 6:12 AM
Prod-InventoryMetrics	Succeeded	1 day ago	Dec 10, 2019 6:06 AM
Prod-CustKPINewDelta	Succeeded	1 day ago	Dec 10, 2019 6:01 AM
Prod-CustKPIDelta	Succeeded	1 day ago	Dec 10, 2019 6:00 AM
Prod-CalculateKPIMetrics	Succeeded	1 day ago	Dec 10, 2019 5:59 AM
UAT-InventoryMetrics	Succeeded	1 day ago	Dec 10, 2019 5:48 AM
UAT-ViewsDeltaLoad	Succeeded	1 day ago	Dec 10, 2019 5:46 AM
UAT-CustomerScorecardEmailScheduler	Succeeded	1 day ago	Dec 10, 2019 5:42 AM
DotNetCore-UAT-CustKPINewDelta	Succeeded	1 day ago	Dec 10, 2019 5:33 AM

Enter the name of the pipeline and choose the service role as a new service role and click on **Next** as shown below. Here a new IAM service role will be created for code pipeline.



Choose pipeline settings

Pipeline settings

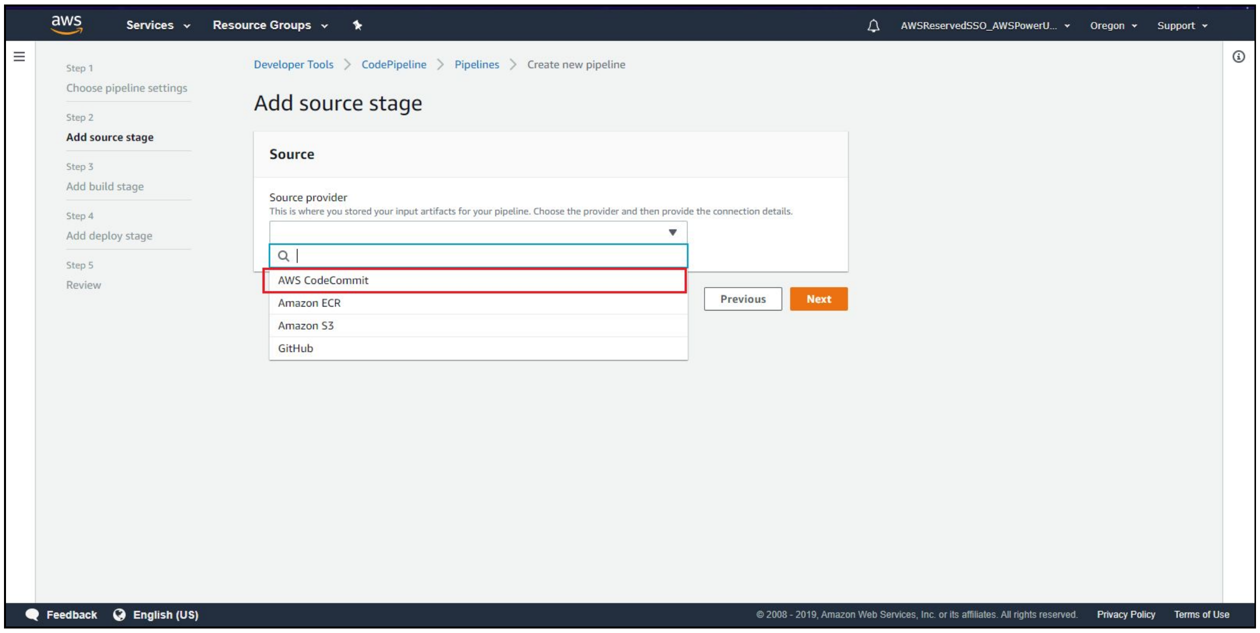
Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
Americold.Dev.BillingApp.UI
No more than 100 characters

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
AWSCodePipelineServiceRole-us-west-2-Americold.Dev.BillingApp.U
Type your service role name

☑ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

Under the add source stage, select the source provider as **AWS code commit**.

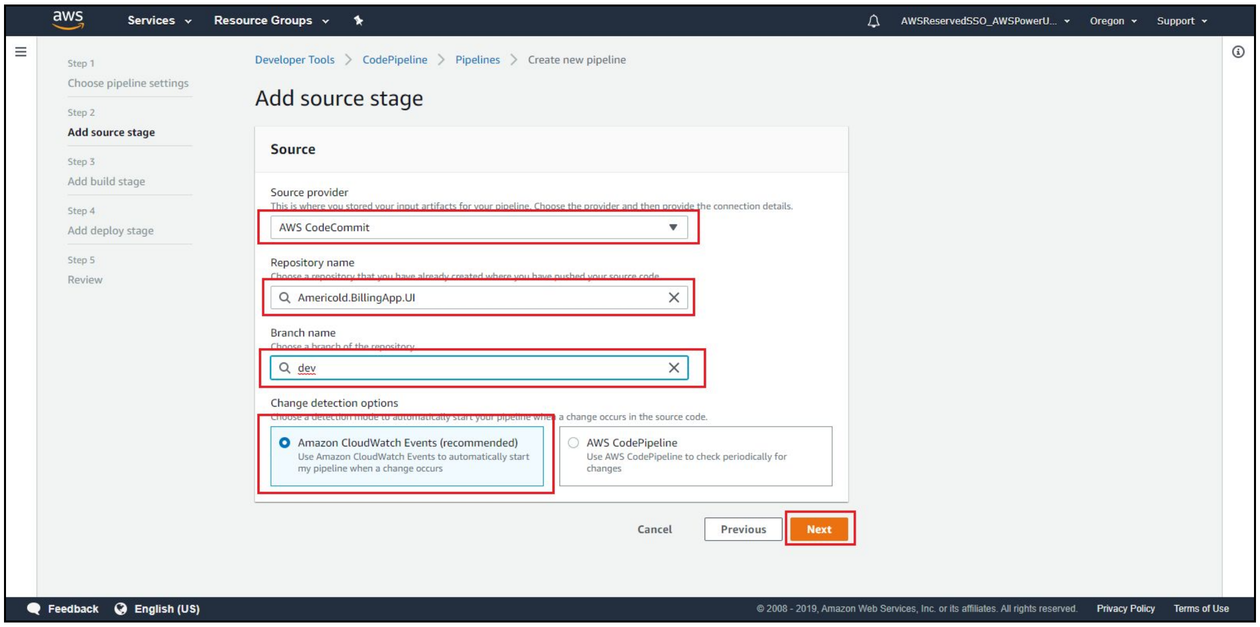


Add source stage

Source

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

Select the name of the repository and branch name. Choose the change detection options as amazon cloudwatch events to enable cloudwatch logs for the pipeline and click on **Next** to go to build configuration.



Add source stage

Source

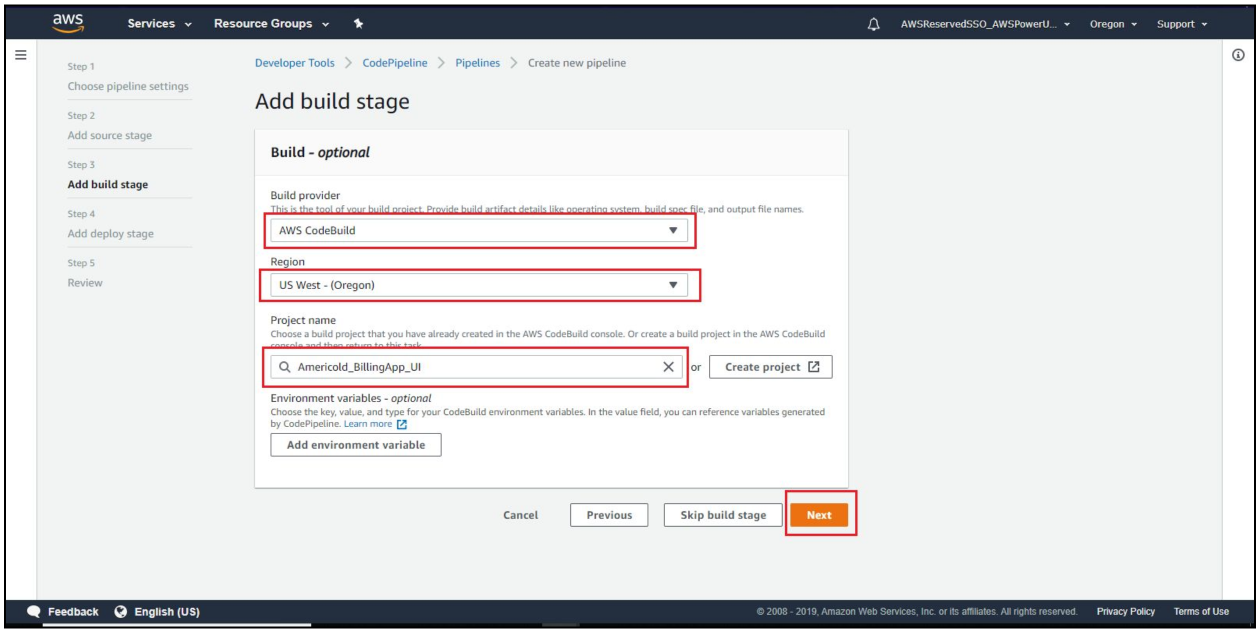
Source provider
This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.
AWS CodeCommit

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

Branch name
Choose a branch of the repository.
dev

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

Under add build stage, select the build provider as **AWS code build**, the region as **US West – (Oregon),** select the build project name that we created in the above steps and click on **Next** as shown below.



Add build stage

Build - *optional*

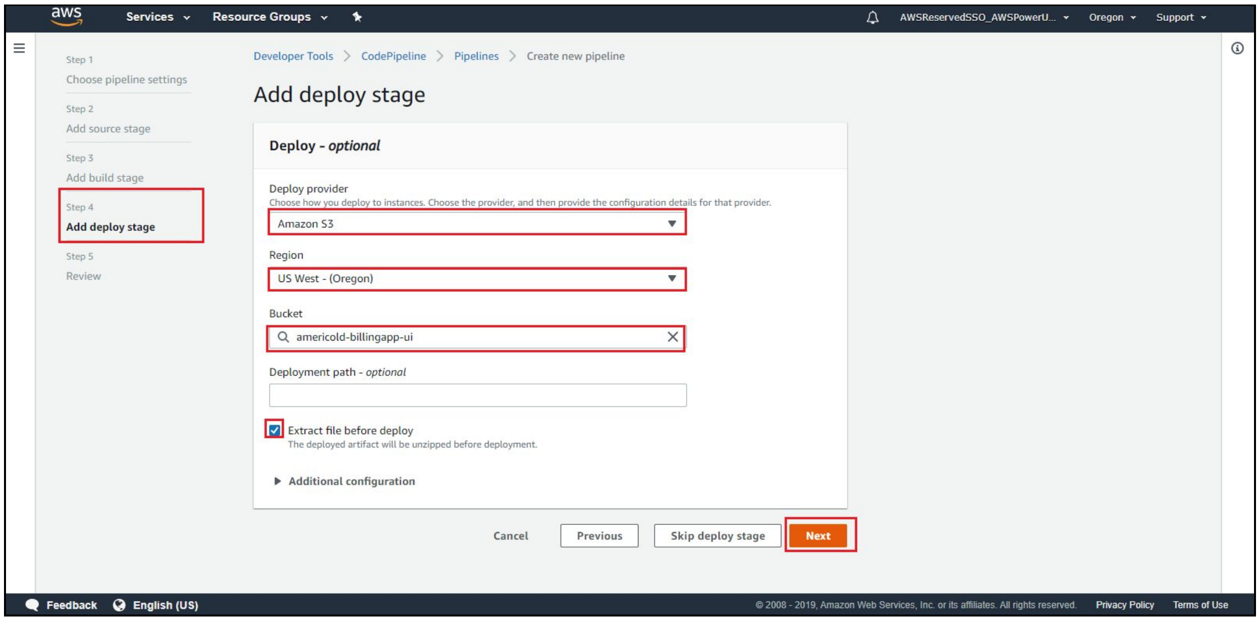
Build provider
This is the tool of your build project. Provide build artifact details like operating system, build spec file, and output file names.
AWS CodeBuild

Region
US West - (Oregon)

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.
Americold_BillingApp_UI

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

In deploy stage select S3 as deploy provider, a region of the bucket, name of the bucket and select extract before deploying and click on next.



Add deploy stage

Deploy - *optional*

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
Amazon S3

Region
US West - (Oregon)

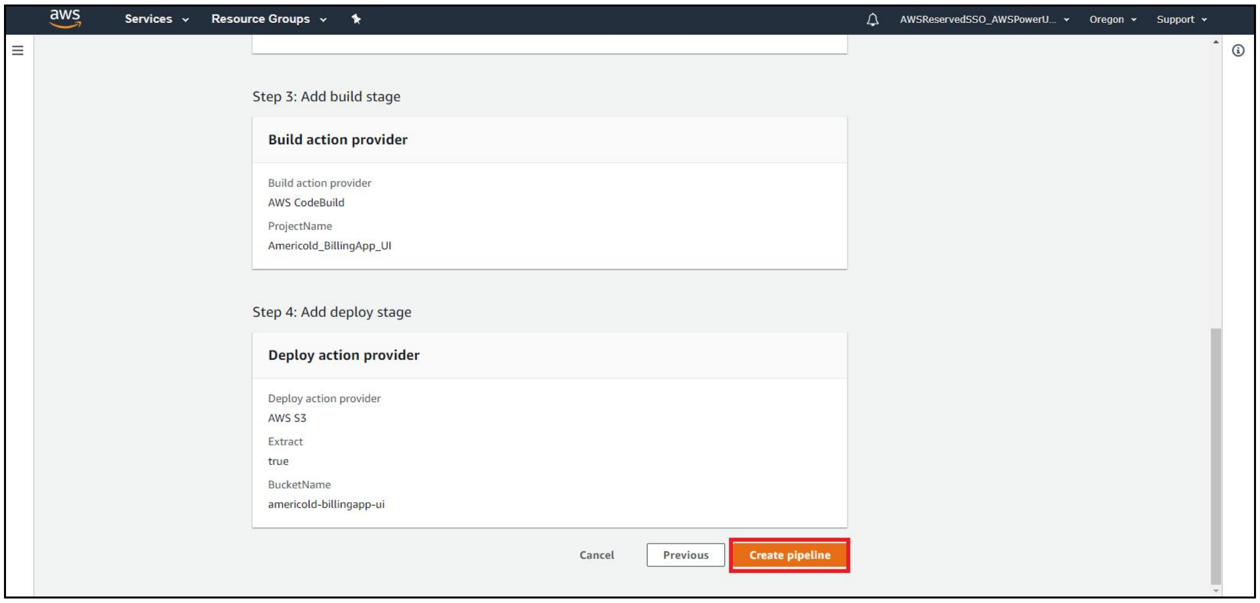
Bucket
americold-billingapp-ui

Deployment path - *optional*

[✓] Extract file before deploy
The deployed artifact will be unzipped before deployment.

Additional configuration

Review the pipeline and click on create a pipeline.



Step 3: Add build stage

Build action provider

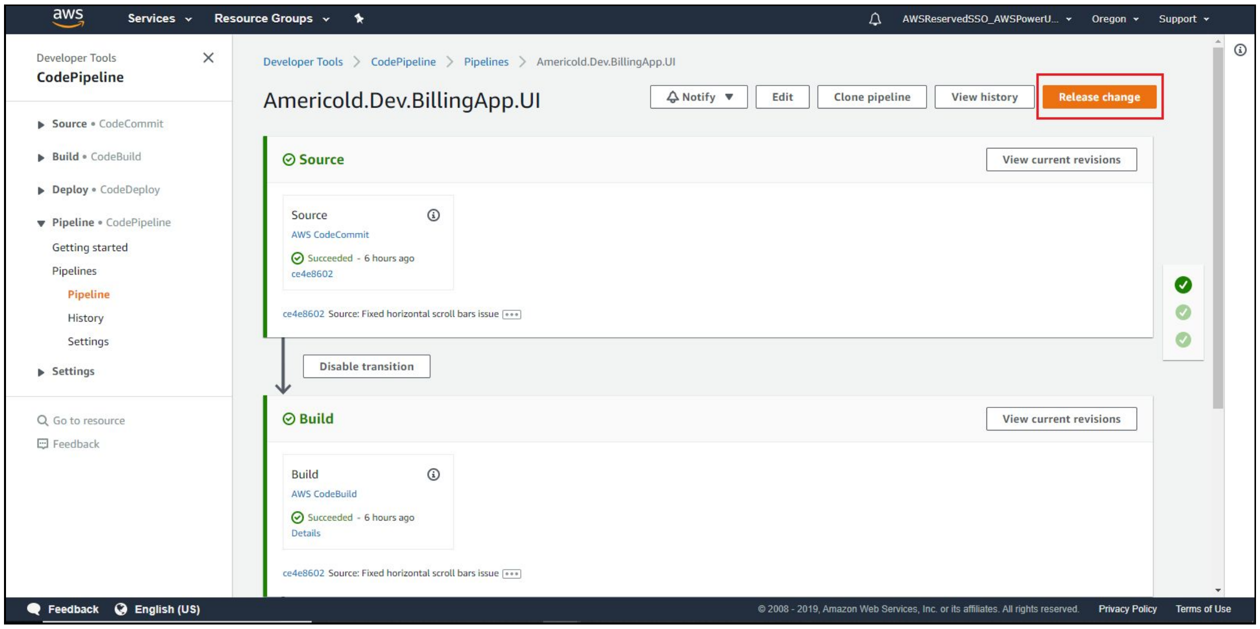
Build action provider
AWS CodeBuild
ProjectName
Americold_BillingApp_UI

Step 4: Add deploy stage

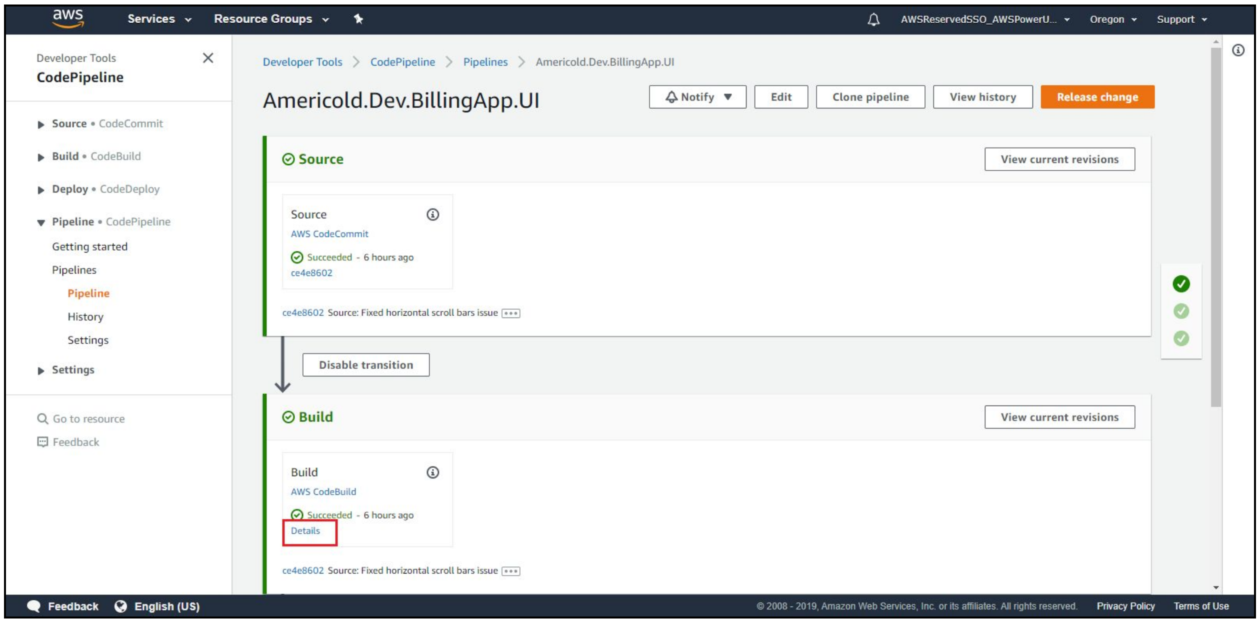
Deploy action provider

Deploy action provider
AWS S3
Extract
true
BucketName
americold-billingapp-ui

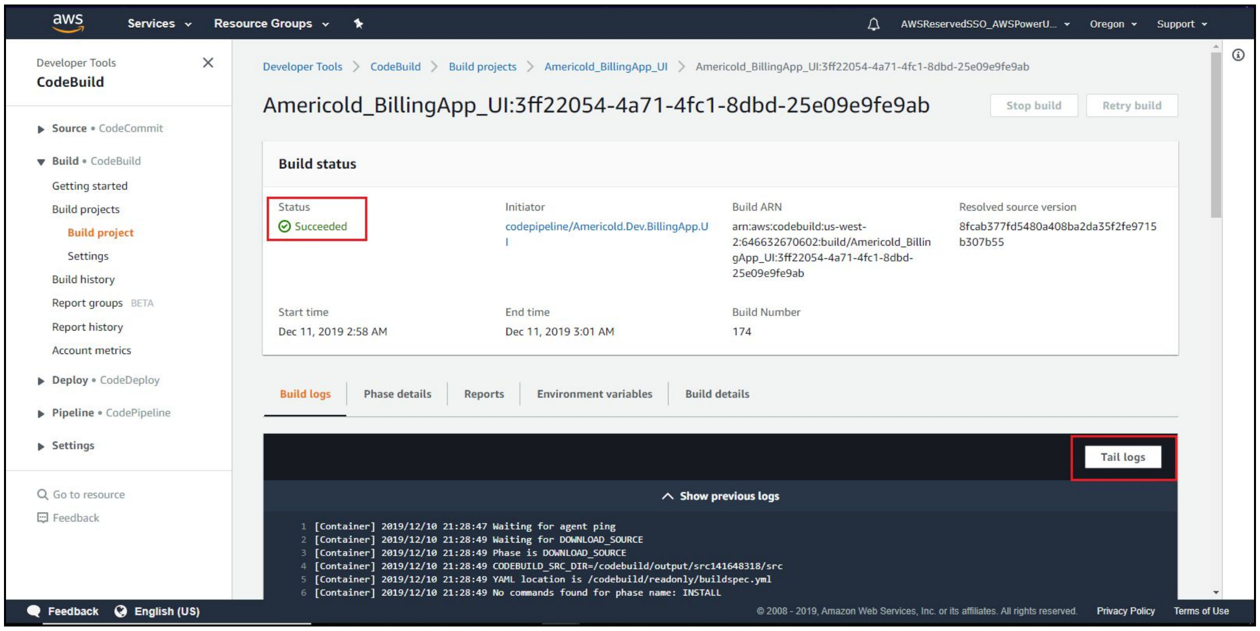
After Click on release change to start the CI/CD pipeline.



We can see the process is automated for CI/CD. We can check the stage logs by clicking on the details. The pipeline will trigger automatically for the first time to release changes and later it will be triggered when the new changes are committed to the repository.



We can check the **build logs** here.



Americold_BillingApp_UI:3ff22054-4a71-4fc1-8dbd-25e09e9fe9ab

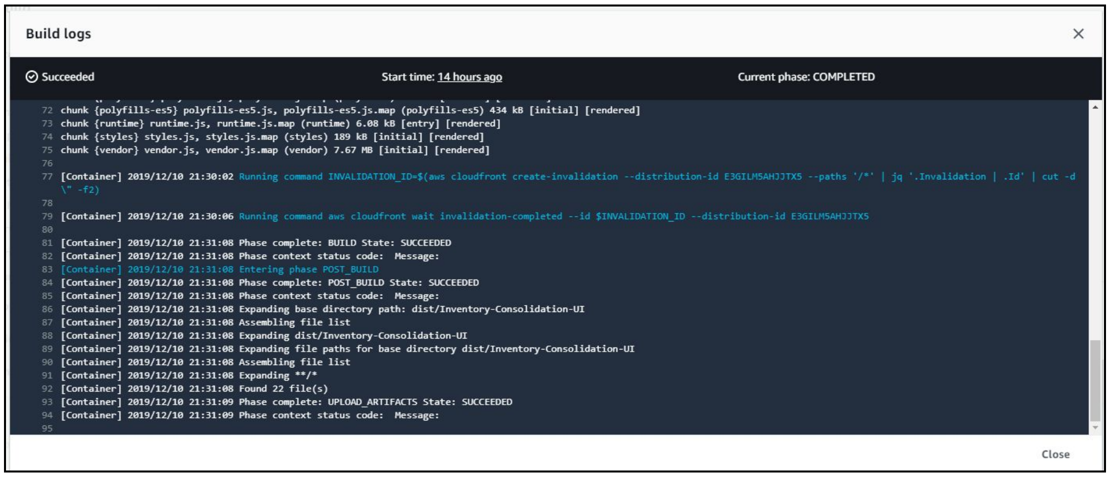
Build status

Status	Initiator	Build ARN	Resolved source version
Succeeded	codepipeline/Americold.Dev.BillingApp.UI	arn:aws:codebuild:us-west-2:646632670602:build/Americold_BillingApp_UI:3ff22054-4a71-4fc1-8dbd-25e09e9fe9ab	8fcab377fd5480a408ba2da35f2fe9715b307b55

Start time	End time	Build Number
Dec 11, 2019 2:58 AM	Dec 11, 2019 3:01 AM	174

```
[Container] 2019/12/10 21:28:47 Waiting for agent ping
[Container] 2019/12/10 21:28:49 Waiting for DOWNLOAD_SOURCE
[Container] 2019/12/10 21:28:49 Phase is DOWNLOAD_SOURCE
[Container] 2019/12/10 21:28:49 CODEBUILD_SRC_DIR=/codebuild/output/src141648318/src
[Container] 2019/12/10 21:28:49 YAML location is /codebuild/readonly/buildspec.yml
[Container] 2019/12/10 21:28:49 No commands found for phase name: INSTALL
```

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Succeeded   Start time: 14 hours ago   Current phase: COMPLETED

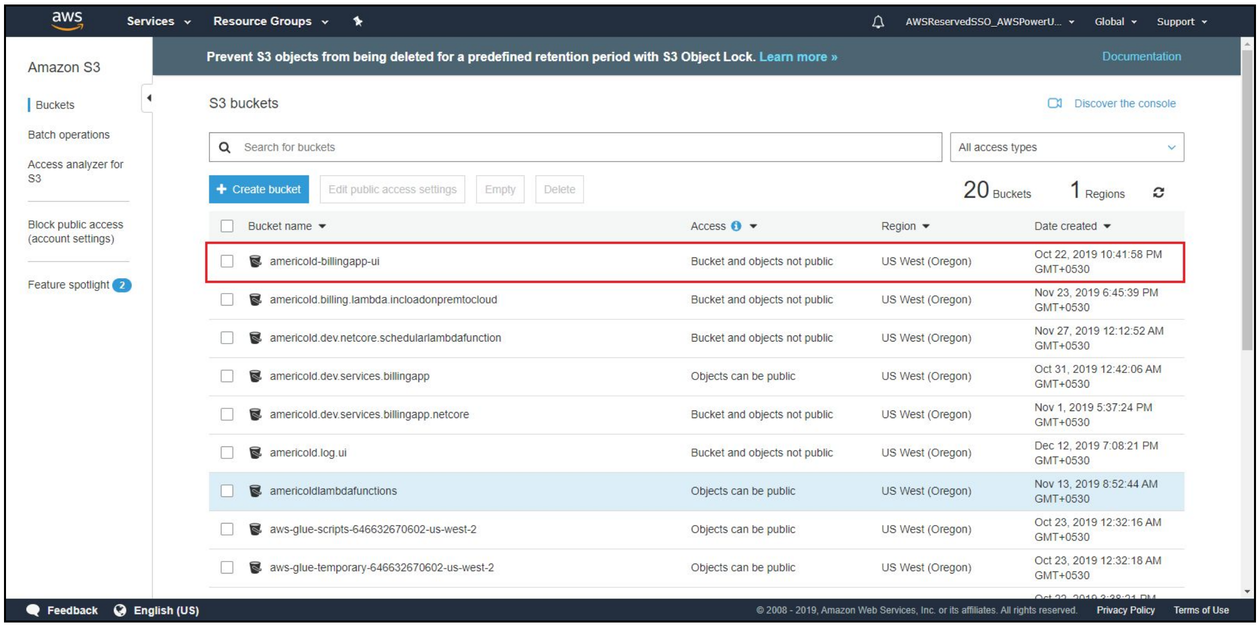
```
chunk {polyfills-es5} polyfills-es5.js, polyfills-es5.js.map (polyfills-es5) 434 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.08 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 189 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 7.67 MB [initial] [rendered]

[Container] 2019/12/10 21:30:02 Running command INVALIDATION_ID=$(aws cloudfront create-invalidation --distribution-id E3GILM5AHJJTX5 --paths '/*' | jq '.Invalidation | .Id' | cut -d
\" -f2)

[Container] 2019/12/10 21:30:06 Running command aws cloudfront wait invalidation-completed --id $INVALIDATION_ID --distribution-id E3GILM5AHJJTX5

[Container] 2019/12/10 21:31:08 Phase complete: BUILD State: SUCCEEDED
[Container] 2019/12/10 21:31:08 Phase context status code:  Message:
[Container] 2019/12/10 21:31:08 Entering phase POST_BUILD
[Container] 2019/12/10 21:31:08 Phase complete: POST_BUILD State: SUCCEEDED
[Container] 2019/12/10 21:31:08 Phase context status code:  Message:
[Container] 2019/12/10 21:31:08 Expanding base directory path: dist/Inventory-Consolidation-UI
[Container] 2019/12/10 21:31:08 Assembling file list
[Container] 2019/12/10 21:31:08 Expanding dist/Inventory-Consolidation-UI
[Container] 2019/12/10 21:31:08 Expanding file paths for base directory dist/Inventory-Consolidation-UI
[Container] 2019/12/10 21:31:08 Assembling file list
[Container] 2019/12/10 21:31:08 Expanding **/*
[Container] 2019/12/10 21:31:08 Found 22 file(s)
[Container] 2019/12/10 21:31:09 Phase complete: UPLOAD_ARTIFACTS State: SUCCEEDED
[Container] 2019/12/10 21:31:09 Phase context status code:  Message:
```

After completing the CI/CD we can check the updated data in the S3 bucket.



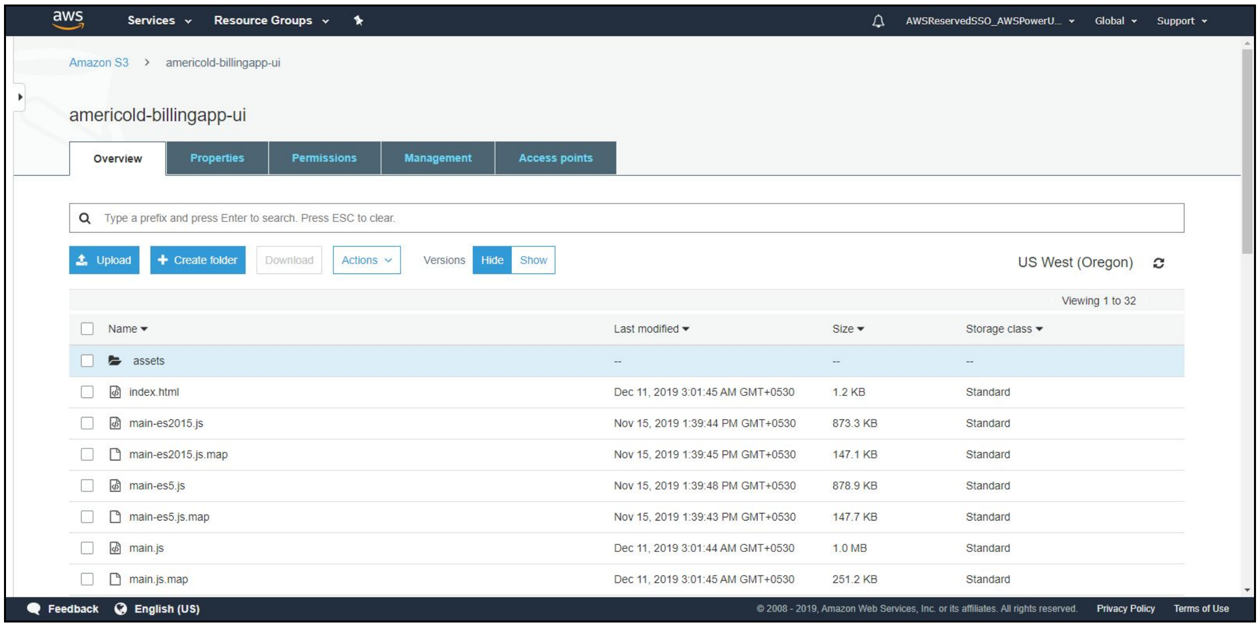
Prevent S3 objects from being deleted for a predefined retention period with S3 Object Lock. Learn more »

S3 buckets

20 Buckets 1 Regions

Bucket name	Access	Region	Date created
americold-billingapp-ui	Bucket and objects not public	US West (Oregon)	Oct 22, 2019 10:41:58 PM GMT+0530
americold.billing.lambda.incloadonpremtocloud	Bucket and objects not public	US West (Oregon)	Nov 23, 2019 6:45:39 PM GMT+0530
americold.dev.netcore.schedularlambdafunction	Bucket and objects not public	US West (Oregon)	Nov 27, 2019 12:12:52 AM GMT+0530
americold.dev.services.billingapp	Objects can be public	US West (Oregon)	Oct 31, 2019 12:42:06 AM GMT+0530
americold.dev.services.billingapp.netcore	Bucket and objects not public	US West (Oregon)	Nov 1, 2019 5:37:24 PM GMT+0530
americold.log.ui	Bucket and objects not public	US West (Oregon)	Dec 12, 2019 7:08:21 PM GMT+0530
americoldlambdafunctions	Objects can be public	US West (Oregon)	Nov 13, 2019 8:52:44 AM GMT+0530
aws-glue-scripts-646632670602-us-west-2	Objects can be public	US West (Oregon)	Oct 23, 2019 12:32:16 AM GMT+0530
aws-glue-temporary-646632670602-us-west-2	Objects can be public	US West (Oregon)	Oct 23, 2019 12:32:18 AM GMT+0530

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.



americold-billingapp-ui

Viewing 1 to 32

Name	Last modified	Size	Storage class
assets	--	--	--
index.html	Dec 11, 2019 3:01:45 AM GMT+0530	1.2 KB	Standard
main-es2015.js	Nov 15, 2019 1:39:44 PM GMT+0530	873.3 KB	Standard
main-es2015.js.map	Nov 15, 2019 1:39:45 PM GMT+0530	147.1 KB	Standard
main-es5.js	Nov 15, 2019 1:39:48 PM GMT+0530	878.9 KB	Standard
main-es5.js.map	Nov 15, 2019 1:39:43 PM GMT+0530	147.7 KB	Standard
main.js	Dec 11, 2019 3:01:44 AM GMT+0530	1.0 MB	Standard
main.js.map	Dec 11, 2019 3:01:45 AM GMT+0530	251.2 KB	Standard

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.