

JOIN THE DevOps/SRE BATCH 8.0 - <https://bit.ly/devopsbatch8>

Check the DevOps Placements - <https://www.youtube.com/watch?v=zm5jAUbDXL0>

AWS DEVOPS CODEPIPELINE

Stage1

Step 1 - Use the CodeCommit console to create the CodeCommit repository



Developer Tools > CodeCommit > Repositories > Create repository

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

 100 characters maximum. Other limits apply.

Description - *optional*

 1,000 characters maximum

Tags

Step 2- Create the IAM user role for codecommit and add the policy as permission: [singambatch]

AWSCodeCommitPowerUser - Add as permission to user

IAM > Users > singambatch > Add permissions

Step 1
Add permissions

Step 2
 Review

Add permissions

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Permissions options

☐ Add user to group
 Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

☐ Copy permissions
 Copy all group memberships, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.

☒ Attach policies directly
 Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1114)

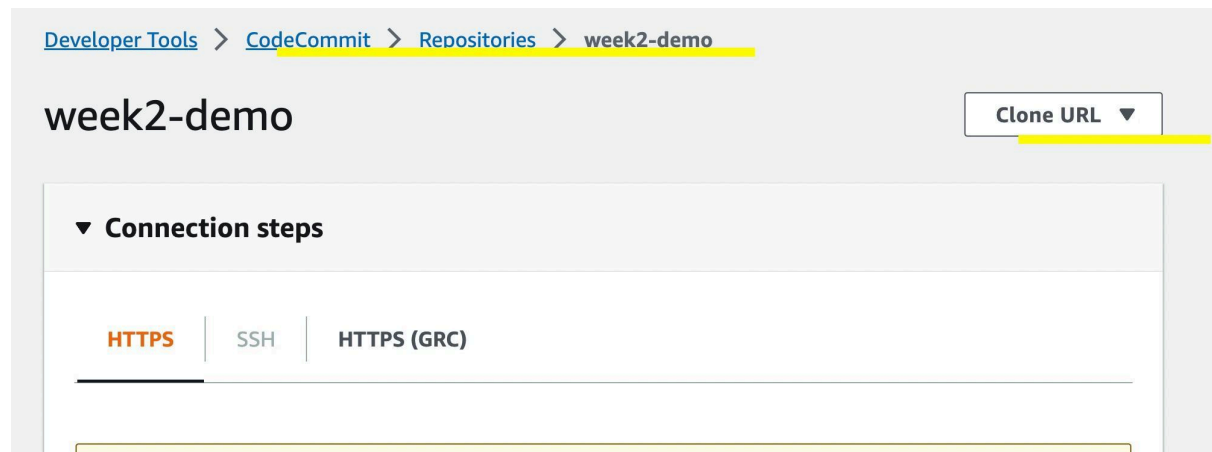
Filter by Type
 All types 1 match

<input checked="" type="checkbox"/>	Policy name	Type	Attached entities
<input checked="" type="checkbox"/>	AWSCodeCommitPowerUser	AWS managed	0

Cancel Next

Step3: Clone your aws codecommit repo in your local, Make sure you have the aws configure command already set up to access aws cli.

Clone repo -



Step 3.1:

Step 3: Create Git credentials for HTTPS connections to CodeCommit

After you have installed Git, create Git credentials for your IAM user in IAM.

To set up HTTPS Git credentials for CodeCommit

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>. Make sure to sign in as the IAM user who will create and use the Git credentials for connections to CodeCommit.
2. In the IAM console, in the navigation pane, choose **Users**, and from the list of users, choose your IAM user.

Note
You can directly view and manage your CodeCommit credentials in **My Security Credentials**. For more information, see [View and manage your credentials](#).

3. On the user details page, choose the **Security Credentials** tab, and in **HTTPS Git credentials for AWS CodeCommit**, choose **Generate**.



IAM > Users > singambatch

singambatch Info

Summary

ARN
arn:aws:iam::164297528770:user/singambatch

Created
May 27, 2023, 19:49 (UTC+05:30)

Console access
Enabled without MFA

Last console sign-in
6 days ago

Access key 1
AKIASMQHFHKB3KUF3LX - Active
Used 42 days ago. 77 days old.

Access key 2
Create access key

Permissions

Groups

Tags (1)

Security credentials

Access Advisor

Console sign-in

Manage console access

Console sign-in link
https://164297528770.signin.aws.amazon.com/console

Console password
Updated 77 days ago (2023-05-27 19:49 GMT+5:30)
Last console sign-in
6 days ago (2023-08-06 17:05 GMT+5:30)

Multi-factor authentication (MFA) (0)

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. [Learn more](#)

Remove

Resync

Assign MFA device

Device type	Identifier	Certifications	Created on
No MFA devices. Assign an MFA device to improve the security of your AWS environment.			

Step 4 -

git clone

<https://git-codecommit.us-west-1.amazonaws.com/v1/repos/we>
[e k2-demo](#)

git clone --mirror

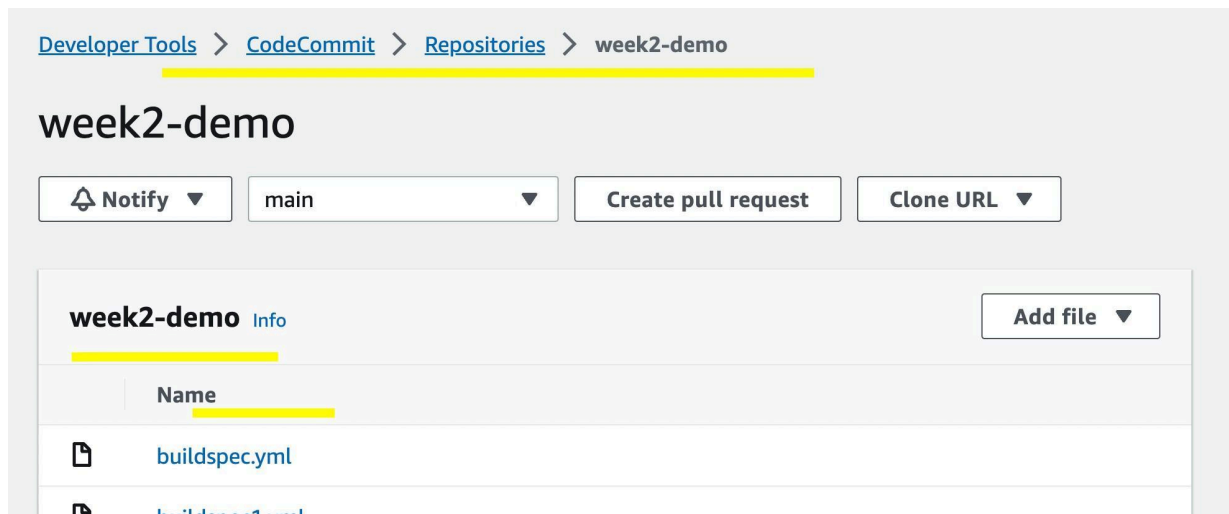
https://github.com/praveen1994dec/aws_pipeline.git github-repo

cd github-repo

git push

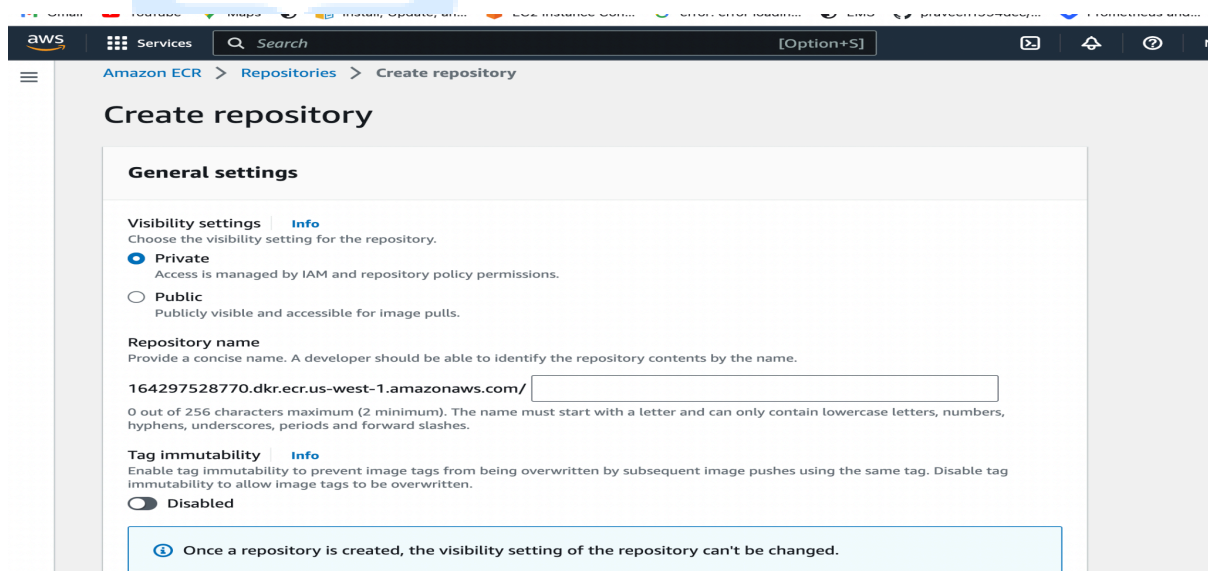
<https://git-codecommit.us-west-1.amazonaws.com/v1/repos/we>
[e k2-demo](#) --all

Step 5- Go to code commit and check the repo details



Stage2: Configure CodeBuild

Create Repository in Elastic Container Registry




164297528770.dkr.ecr.us-west-1.amazonaws.com/singambatch

2.1 -> #Setup a CodeBuild project

Click on create code build project -> select repository -> select branch -> and add the below data

Select the PRIVILEGED checkbox

Project name
 

A project name must be 1-255 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


Enable concurrent build limit - *optional*
Limit the number of allowed concurrent builds for this project.
☐ Restrict number of concurrent builds this project can start

► Additional configuration
tags


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.

Source version [Info](#)
refs/heads/main
c0f204ba first commit

▼ Additional configuration
[Git clone depth](#) [Git submodules](#)

New environment image



Managed image

Use an image managed by AWS CodeBuild



Custom image

Specify a Docker image

Operating system

Amazon Linux 2



Runtime(s)

Standard



Image

aws/codebuild/amazonlinux2-x86_64-standard:4.0



Image version

aws/codebuild/amazonlinux2-x86_64-standard:4.0-23.02.16



Privileged



Enable this flag if you want to build Docker images or want your builds to get elevated privileges.

Service role

Choose an existing service role from your account



arn:aws:iam::164297528770:role/service-role/codebuild-service-rolec



FILL THE ENV VARIABLES:

- ☐ 7 GB memory, 4 vCPUs
- ☐ 15 GB memory, 8 vCPUs
- ☐ 145 GB memory, 72 vCPUs

Environment variables

Name	Value	Type	
AWS_DEFAULT_REGION	us-west-1	Plaintext ▼	Remove
AWS_ACCOUNT_ID	164297528770	Plaintext ▼	Remove
IMAGE_TAG	latest	Plaintext ▼	Remove
IMAGE_REPO_NAME	demo3	Plaintext ▼	Remove

Add environment variable

Create parameter



Logs

CloudWatch

☒ CloudWatch logs - *optional*

Checking this option will upload build output logs to CloudWatch.

Group name

demo3

Stream name

demo3

S3

☐ S3 logs - *optional*

Checking this option will upload build output logs to S3.

Cancel

Create build project

2.2 -> Go to codebuild IAM role and attach this policy so that the codebuild can access ECR - [AmazonEC2ContainerRegistryPowerUser](#)

IAM > Roles > codebuild-B-service-role > Add permissions

Attach policy to codebuild-B-service-role

▶ Current permissions policies (2)

Other permissions policies (Selected 1/869) ↻ Create policy ↗

Filter policies by property or policy name and press enter. 3 matches

amazonec2containerre ✕ Clear filters

<input type="checkbox"/>	Policy name ↗	Type	Description
<input type="checkbox"/>	AmazonEC2ContainerRegistryReadOnly	AWS managed	Provides read-only access to
<input checked="" type="checkbox"/>	AmazonEC2ContainerRegistryPowerUser	AWS managed	Provides full access to Amaz
<input type="checkbox"/>	AmazonEC2ContainerRegistryFullAccess	AWS managed	Provides administrative acce

Cancel Add permissions

2.3 - Go to codebuild -> Run the build

Services Search [Option+S] N. California singambatch @ 1642-9752

Developer Tools > CodeBuild > Build projects > Build-demo3 > Build-demo3:b9b2b69c-a4b8-4979-a27d-05fc5efd90ec

Build-demo3:b9b2b69c-a4b8-4979-a27d-05fc5efd90ec Stop build Retry build

Build status

Status	Initiator	Build ARN	Resolved source version
In progress	singambatch	arn:aws:codebuild:us-west-1:164297528770:build/Build-demo3:b9b2b69c-a4b8-4979-a27d-05fc5efd90ec	-

Start time	End time	Build number
Aug 6, 2023 7:49 PM (UTC+5:30)	-	1

Stage3 - Create a CodePipeline

Step1 - Create the code pipeline

Developer Tools > CodePipeline > Pipelines > Create new pipeline

Step 1
Choose pipeline settings

Step 2
Add source stage

Step 3
Add build stage

Step 4
Add deploy stage

Step 5
Review

Choose pipeline settings [Info](#)

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.

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

Type your service role name

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

▼ Advanced settings

Artifact store

☒ **Default location**
Create a default S3 bucket in your account.

☐ **Custom location**
Choose an existing S3 location from your account in the same region and account as your pipeline

Encryption key

☒ **Default AWS Managed Key**
Use the AWS managed customer master key for CodePipeline in your account to encrypt the data in the artifact store.

☐ **Customer Managed Key**
To encrypt the data in the artifact store under an AWS KMS customer managed key, specify the key ID, key ARN, or alias ARN.

Step2-

Developer Tools

>

CodePipeline

>

Pipelines

>

Create new pipeline

Step 1

Choose pipeline settings

Step 2

Add source stage

Step 3

Add build stage

Step 4

Add deploy stage

Step 5

Review

Add source stage Info

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.

week2-demo

Branch name
Choose a branch of the repository

main

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

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.

Cancel

Previous

Next

Step3 -

Developer Tools

>

CodePipeline

>

Pipelines

>

Create new pipeline

Step 1

Choose pipeline settings

Step 2

Add source stage

Step 3

Add build stage

Step 4

Add deploy stage

Step 5

Review

Add build stage Info

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 (N. California)

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.

build-job-pipeline

 or

Create project

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**
Triggers a single build.

☐ **Batch build**
Triggers multiple builds as a single execution.

Cancel

Previous

Skip build stage

Next

Step4 -

Skip the deploy stage and save [The pipeline will trigger]

Happy

Learning,

Singam

