Build – CD Pipeline

AWS **CodeCommit** is a fully-managed source control service that hosts secure Git-based repositories. It makes it easy for teams to collaborate on code in a secure and highly scalable ecosystem. **CodeCommit** eliminates the need to operate your own source control system or worry about scaling its infrastructure.

CodeDeploy is a deployment service that automates application deployments to Amazon EC2 instances, on-premises instances, serverless Lambda functions, or Amazon ECS services. You can deploy a nearly unlimited variety of application content, including: code.

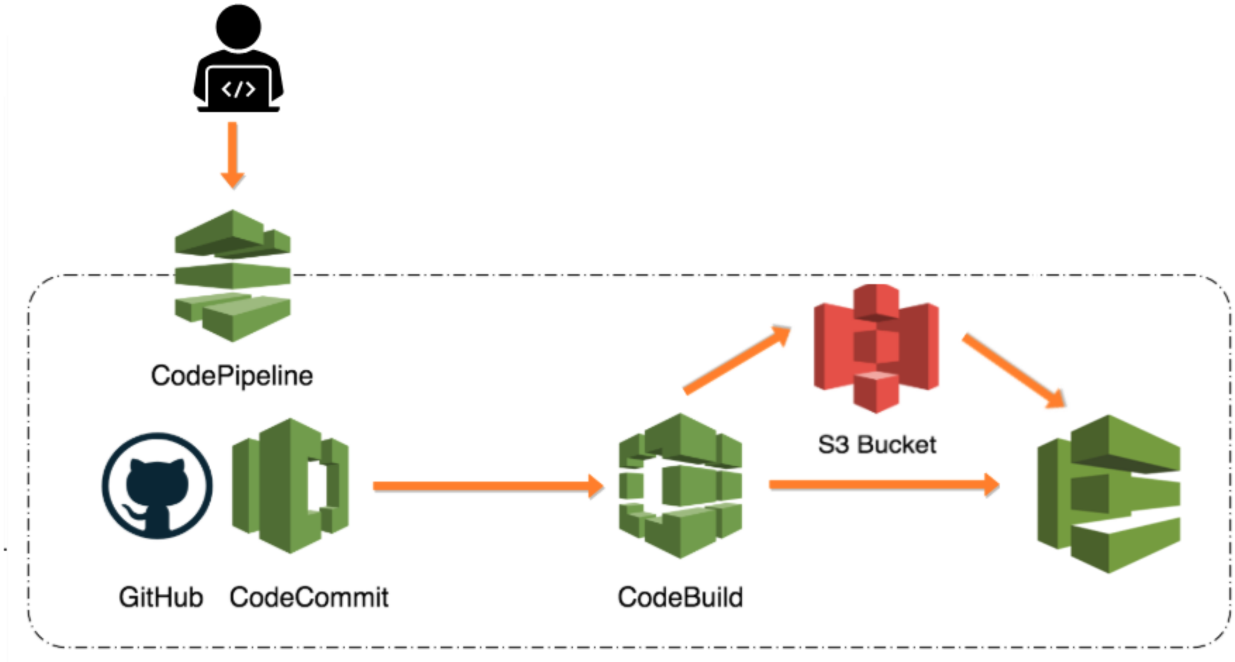
CodePipeline is a workflow management tool, which allows the user to configure a series of steps to form a **pipeline**. The service leverages many of the management tools already in the AWS environment, such as AWS CodeCommit and AWS CodeDeploy, but it does not limit itself to aggregating only internal services.

This lab shows you how to build CD pipeline with AWS code commit, code deploy and code pipeline services.

You will deploy an Amazon code pipeline with:

- Download Sample Code from git and push it to CodeCommit.
- Launch EC2 instance to run the application.
- Configure S3 to store artifact of application.
- Create Application in CodeDeploy
- Create CodePipeline

The final architecture will be:



Duration

This lab will require approximately **30 minutes** to complete.

Accessing the AWS Management Console

⚠ **Windows Users**: Please use Chrome or Firefox as your web browser for this lab. The lab instructions are **not compatible with Internet Explorer** due to a difference in the Amazon console.

Sign to the AWS Management Console using your credentials.

⚠ Please do not change the Region during this lab.

Task 1: Pre-Requisites

1. Download and install **git client** on your linux or window workstation.
2. **Configure** GIT on local workstation (shell or windows cmd)
    a. git config --global user.name <your email id>
    b. git config --global user.name "Your name"
3. Create new **AWS IAM User** (or use Admin login)
    a. Attach IAM policy "**AWSCodeCommitFullAccess**"
    b. Attach IAM Policy "**AWSCodePipelineFullAccess**"
    c. Security Credentials -> HTTPS GIT Credentials -> generate and save credentials
4. Create Service **Role** for Code Deploy
    a. Create IAM Role for CodeDeploy service
    b. Attach existing IAM Policy "**AWSCodeDeployRole**"
5. Create IAM Role for **EC2 to download artifacts from S3**
    a. Attach EC2 role with "**S3ReadonlyAccess**"

Task 2: Configure Code Commit Repository

1. Go to **CodeCommit** Service (Use Same Region for all services)

a. **Create New Code repository** with name "capg-repo"

b. Copy the clone URL -> Clone HTTPS

2. **Clone git** repository locally

   a. git clone <URL>

   b. Provide username/password. This should clone the empty repository.

3. Download Sample Application

   a. Download application locally from "**https://github.com/aws-samples/aws-codedeploy-samples/** "

   b. Unzip it and move all files and folder inside **your local repository "capg-project"** directory

   c. **Change your CMD working directory** to "capg-project"

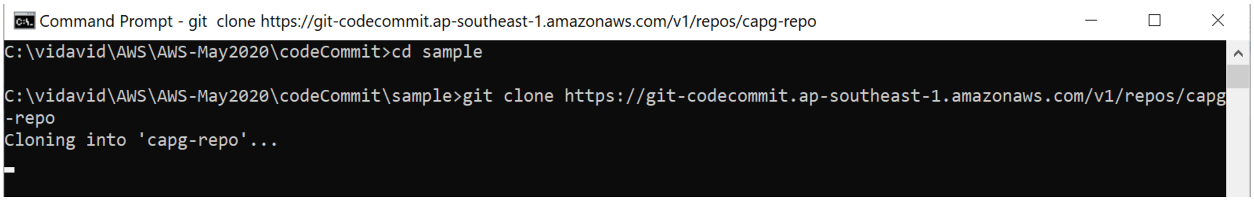
   d. git add -A

   e. git commit -m "Add Sample application files"

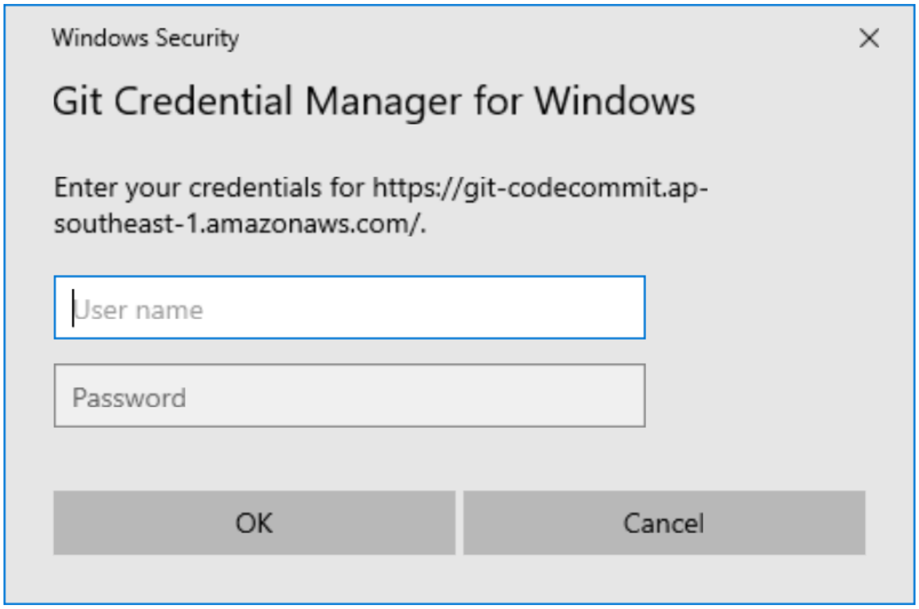
   f. git push

4. **Verify** that files are pushed to CodeCommit repo through AWS console

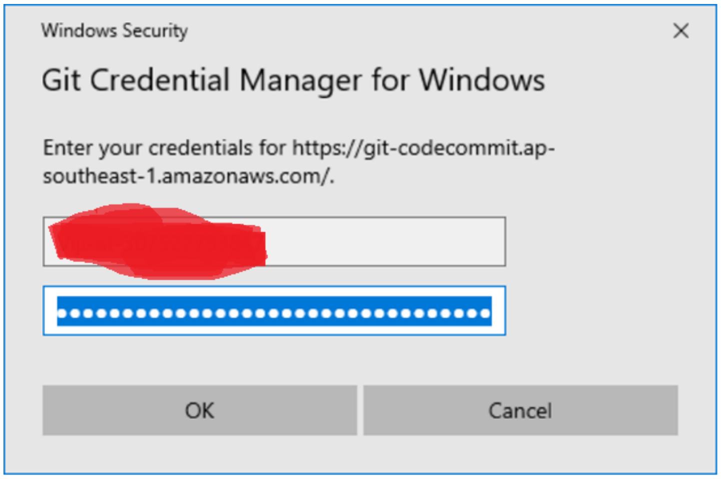
Few Sample Screenshots below:


Command Prompt - git clone https://git-codecommit.ap-southeast-1.amazonaws.com/v1/repos/capg-repo

```
C:\vidavid\AWS\AWS-May2020\codeCommit>cd sample

C:\vidavid\AWS\AWS-May2020\codeCommit\sample>git clone https://git-codecommit.ap-southeast-1.amazonaws.com/v1/repos/capg
-repo
Cloning into 'capg-repo'...
```



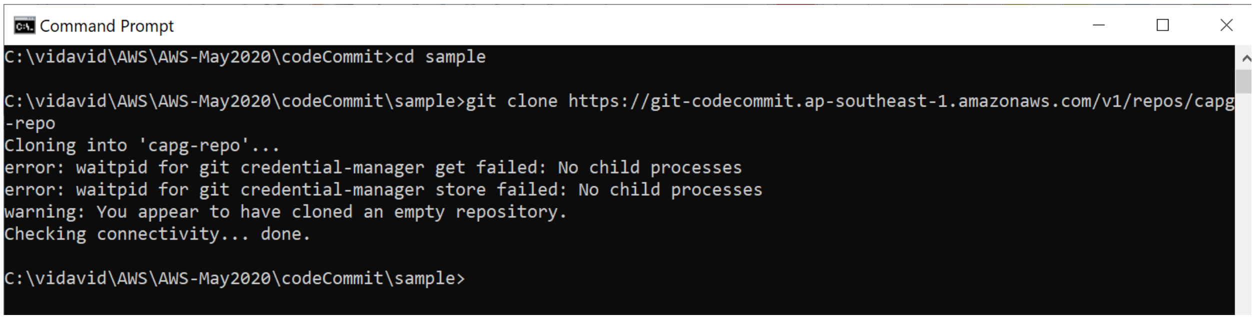
Enter your credentials for https://git-codecommit.ap-southeast-1.amazonaws.com/.



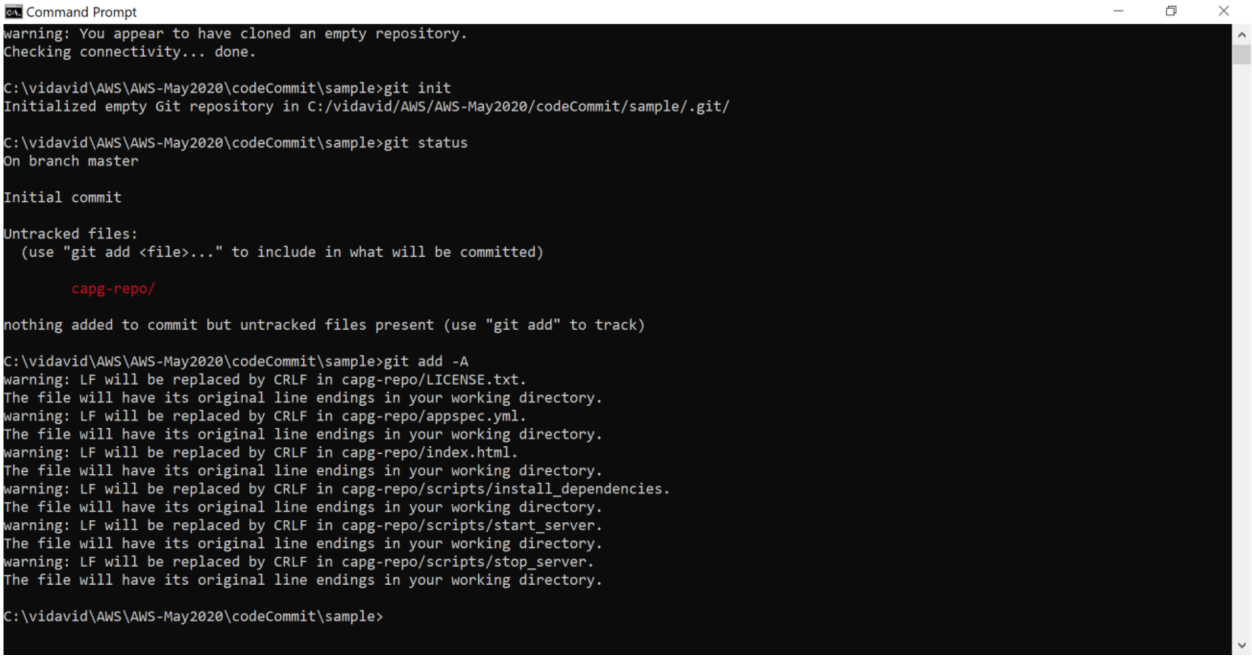
Enter your credentials for https://git-codecommit.ap-southeast-1.amazonaws.com/.

[redacted]

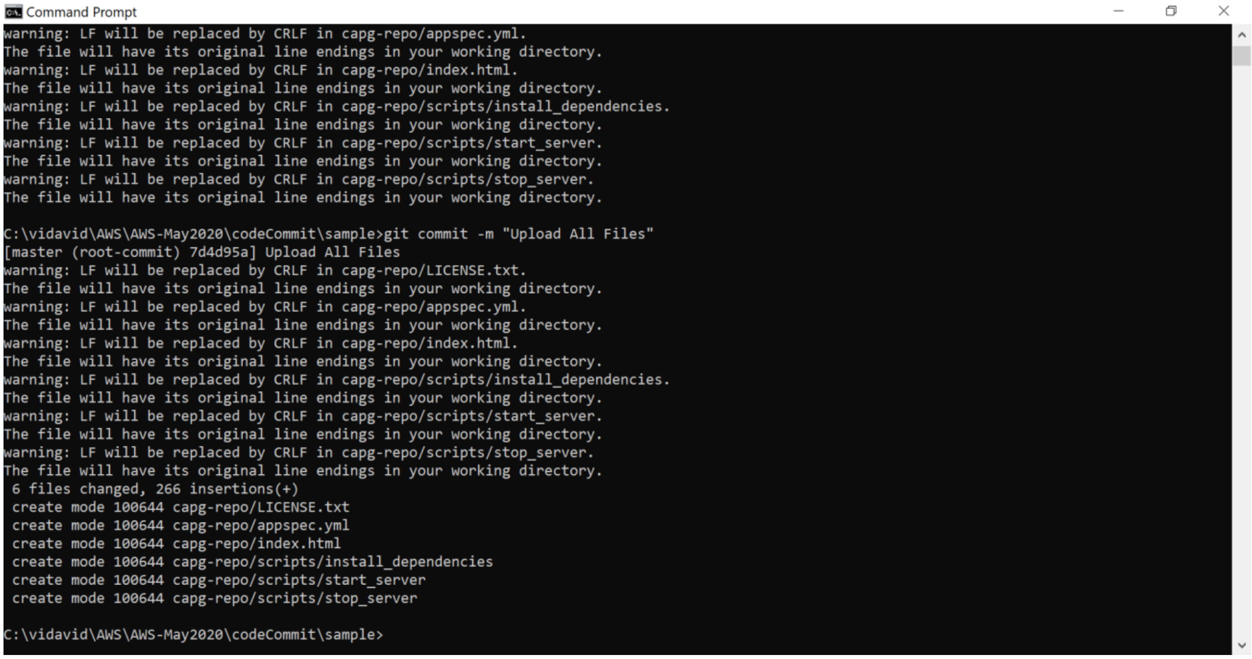
•••••••••••••••••••••••••••••••••



```
C:\vidavid\AWS\AWS-May2020\codeCommit>cd sample

C:\vidavid\AWS\AWS-May2020\codeCommit\sample>git clone https://git-codecommit.ap-southeast-1.amazonaws.com/v1/repos/capg
-repo
Cloning into 'capg-repo'...
error: waitpid for git credential-manager get failed: No child processes
error: waitpid for git credential-manager store failed: No child processes
warning: You appear to have cloned an empty repository.
Checking connectivity... done.

C:\vidavid\AWS\AWS-May2020\codeCommit\sample>
```



```
warning: You appear to have cloned an empty repository.
Checking connectivity... done.

C:\vidavid\AWS\AWS-May2020\codeCommit\sample>git init
Initialized empty Git repository in C:/vidavid/AWS/AWS-May2020/codeCommit/sample/.git/

C:\vidavid\AWS\AWS-May2020\codeCommit\sample>git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        capg-repo/

nothing added to commit but untracked files present (use "git add" to track)

C:\vidavid\AWS\AWS-May2020\codeCommit\sample>git add -A
warning: LF will be replaced by CRLF in capg-repo/LICENSE.txt.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in capg-repo/appspec.yml.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in capg-repo/index.html.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in capg-repo/scripts/install_dependencies.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in capg-repo/scripts/start_server.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in capg-repo/scripts/stop_server.
The file will have its original line endings in your working directory.

C:\vidavid\AWS\AWS-May2020\codeCommit\sample>
```



```
warning: LF will be replaced by CRLF in capg-repo/appspec.yml.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in capg-repo/index.html.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in capg-repo/scripts/install_dependencies.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in capg-repo/scripts/start_server.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in capg-repo/scripts/stop_server.
The file will have its original line endings in your working directory.

C:\vidavid\AWS\AWS-May2020\codeCommit\sample>git commit -m "Upload All Files"
[master (root-commit) 7d4d95a] Upload All Files
warning: LF will be replaced by CRLF in capg-repo/LICENSE.txt.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in capg-repo/appspec.yml.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in capg-repo/index.html.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in capg-repo/scripts/install_dependencies.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in capg-repo/scripts/start_server.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in capg-repo/scripts/stop_server.
The file will have its original line endings in your working directory.
 6 files changed, 266 insertions(+)
 create mode 100644 capg-repo/LICENSE.txt
 create mode 100644 capg-repo/appspec.yml
 create mode 100644 capg-repo/index.html
 create mode 100644 capg-repo/scripts/install_dependencies
 create mode 100644 capg-repo/scripts/start_server
 create mode 100644 capg-repo/scripts/stop_server

C:\vidavid\AWS\AWS-May2020\codeCommit\sample>
```

After that enter “**git push**”

Task 3: Launch EC2 instance to host application

1. Launch EC2 instance

   a. **AMI**: Amazon Linux2

   b. **Instance type**: t2.micro

   c. VPC default

   d. **IAM Role:** Role that you created in **Task1**

   e. **Advanced** section, paste below userdata to install CodeDeploy Agent

```
#!/bin/bash

yum -y update

yum install -y ruby

yum install -y aws-cli

cd /home/ec2-user

wget https://aws-codedeploy-ap-southeast-1.s3.ap-southeast-

1.amazonaws.com/latest/install

chmod +x ./install

./install auto
```

   f. **Tag**: **Key** = name, **value** = capg

   g. **Security Group**: OPen port 22 and 80

   h. **SSH Key pair**: Use existing KeyPair/Create new One

   i. Launch

Task 4: Create Application in CodeDeploy

2. Go to **CodeDeploy** Service
3. Create **New Application**
    a. **Name**: Capg
    b. **Compute Platform**: EC2/On-Premises
    c. Create Application
4. **Deployment Groups**
    a. **Name**: Capg-Group
    b. **Deployment Type**: in-place deployment
    c. **Environment Configuration**:
        i. Amazon **EC2 instances**
        ii. Enter the Key=**name** and Value=**capg** (that you created while launching ec2-instance)
5. **Deployment Configuration**
    a. Select **CodeDeploydefault**: **OneAtaTime**
6. **Disable ELB configuration.**

Task 5: Create CodePipeline

1. Go to AWS **CodePipeline** Service
2. **Create Pipeline**
    a. **Name**: Capg-pipeline
    b. **Service Role**: New Service Role
    c. **Artifact Store**: Default
3. **Add Source Stage**

   a. **Source Provider**: AWS CodeCommit
   b. **Repository**: Select your repository "capg-repo"
   c. **Branch**: Select master branch
   d. **Detection option**: Select ColudWatch Events

4. Add **build Stage -> Skip**
5. Add **Deploy Stage**
   a. **Deploy provider**: AWS CodeDeploy
   b. **Application Name**: Capg (This you had created previously)
   c. **Deployment Group**: capg-group (this you has created previously)

Lab Complete

Congratulations! You have completed the lab. Click End Lab at the top of this page to clean up your lab environment.