

サービス	CodeCommit	CodeBuild	CodeDeploy	CodePipeline	CodeStar
機能	コードのリポジトリ	ビルド	デプロイ	CI/CD	プロジェクト作成
概要	Gitリポジトリをホスティング。IAMユーザーに設定したCodeCommit用の認証情報にてアクセス（SSHまたはHTTPS）。ブランチ、タグ、プルリクエスト機能（ブランチのマージ）。サービス内部ではDynamoDBとS3が利用されている。	CodeCommit等からコードを取り出してビルド。アーティファクト（ビルド成果物）はS3に保管。ビルドの「フェーズ」ごとにコマンドを実行可能（ビルド前のパッケージダウンロード、ビルド後のログ出力など）。サービス内部ではDockerコンテナを使用。	「リビジョン」（アーティファクト）をデプロイグループ（EC2インスタンス等）にデプロイ。デプロイ前後にコマンドを実行可能（サービスの停止・再起動など）。	リポジトリからのコードの取り出し、ビルド、デプロイを連続的に実行。内部的に「アーティファクトストア（S3）」を使用。	テンプレートからツールチェーンをすばやく作成。チーム作業を支援（ダッシュボード、チームメンバー管理）。イシュー管理（JIRA統合）。IDEとの連携が可能。
主なAWSリソース	・リポジトリ	・ビルドプロジェクト	・アプリケーション ・デプロイグループ ・デプロイ	・パイプライン ・ステージ ・アクション	・プロジェクト
設定ファイル	なし	buildspec.yml	appspec.yml	なし	なし
記述例	（CodeCommitではなく）Gitの設定ファイルとしては.gitignoreやGitフック等。	version: 0.2 phases: build: commands: date >> index.html artifacts: files: - index.html - appspec.yml	version: 0.0 os: linux files: - source: /index.html destination: /var/www/html		
IAMロール	なし	ビルドプロジェクトとともに自動作成される。 codebuild-ppp-service-role （ppp: プロジェクト名。）	事前に作成する必要がある。 ①サービスロール：CodeDeployのサービスロールを作成。AWSCodeDeployRoleポリシーがアタッチされる。 ②デプロイ先のEC2インスタンス用ロール：AmazonEC2RoleforAWSCodeDeployポリシーをアタッチする。	パイプラインとともに自動作成される。 AWSCodePipelineServiceRole-rrr-ppp （rrr: リージョン,ppp: パイプライン名。）	CodeStarを最初に使用するとき自動作成される。 aws-codestar-service-role
備考	fatal: unable to access 'https://git-codecommit.ap-northeast-1.amazonaws.com/v1/repos/myrepo/': The requested URL returned error: 403 →Keychainに記録された認証情報を削除	ビルドプロジェクト作成前に、アーティファクト格納用のS3バケットを作成しておく必要がある。アーティファクトの名前は「Name」で指定（「app.zip」などのように指定）。  CodeBuildのビルドを手動で開始した場合は、アーティファクトは上記の（ビルドプロジェクトに設定した）バケットに格納（更新）される。CodePipelineのステージからビルドが開始された場合は、アーティファクトはCodePipelineの「アーティファクトストア」内に格納（更新）され、ビルドプロジェクトに設定したバケットのアーティファクトは更新されない。	デプロイグループ作成時、アーティファクトのパス（「s3://bucket/app.zip」など）を指定。  デプロイ先がEC2インスタンスの場合、CodeDeployエージェントをインストール。  #1/bin/bash yum update -y yum install -y ruby wget wget https://aws-codedeploy-us-west-2.s3.amazonaws.com/latest/install chmod +x ./install ./install auto  yum install -y httpd systemctl enable --now	アーティファクトストアはパイプライン作成時に自動で作成される。	
役割別ポリシー	AWSCodeCommitFullAccess	AWSCodeBuildAdminAccess	AWSCodeDeployFullAccess	AWSCodePipelineFullAccess	
	AWSCodeCommitPowerUser	AWSCodeBuildDeveloperAccess	AWSCodeDeployDeployerAccess	AWSCodePipelineApproverAccess	
	AWSCodeCommitReadOnly	AWSCodeBuildReadOnlyAccess	AWSCodeDeployReadOnlyAccess	AWSCodePipelineReadOnlyAccess	
演習（デモ）手順例	ローカルPCにGit for Windowsをインストール。IAMユーザーのCodeCommit用認証情報（HTTPS）を設定。CodeCommitでリポジトリを作成。ローカルPCでリポジトリをcloneし、index.html、buildspec.yml、appspec.ymlを配置してコミット・プッシュ。CodeCommitでコミット結果を確認。	アーティファクト格納用のS3バケットを作成。ビルドプロジェクトを作成してビルドを開始。ビルドが完了したらS3バケットからアーティファクトをダウンロードして結果を確認。	CodeDeploy用サービスロール、EC2用ロールを作成。EC2インスタンスを起動（Name=web）、ロールをアタッチ、ユーザーデータにてCodeDeployエージェントとhttpdをインストール。セキュリティグループはHTTPを許可。CodeDeployアプリケーションを作成。デプロイグループを作成（タグでEC2インスタンスを指定）。デプロイを実行。EC2インスタンスにWebアクセスして結果を確認。	パイプラインを作成し、各ステージで、CodeCommitリポジトリ、CodeCommitビルドプロジェクト、CodeDeployアプリケーションを指定。ローカルPCでコードをプッシュするとパイプラインが実行され、ビルドとデプロイが自動的に行われる。EC2インスタンスにWebアクセスして結果を確認。	