CICD

Q: How is AWS CodeCommit different from other Git-based source control systems?

AWS CodeCommit offers a number of features not offered by other Git source control systems:

* Fully Managed –AWS CodeCommit eliminates the need to host, maintain, backup, and scale your own source control servers.
* Secure –AWS CodeCommit automatically encrypts your files in transit and at rest. AWS CodeCommit is integrated with AWS Identity and Access Management (IAM), allowing you to assign user-specific permissions to your repositories.
* Highly Available – AWS CodeCommit is built on highly scalable, redundant, and durable AWS services such as Amazon S3 and Amazon DynamoDB.
* Scalable - AWS CodeCommit allows you store any number of files and there are no repository size limits.
* Faster Development Lifecycle - AWS CodeCommit keeps your repositories close to your build, staging, and production environments in the AWS cloud. This allows you to increase the speed and frequency of your development lifecycle.

Q: How does AWS CodeCommit compare to a versioned S3 bucket?

AWS CodeCommit is designed for collaborative software development. It manages batches of changes across multiple files, offers parallel branching, and includes version differencing (“diffing”). In comparison, Amazon S3 versioning supports recovering past versions of individual files but doesn’t support tracking batched changes that span multiple files or other features needed for collaborative software development.

Q: How do I import my existing repository to AWS CodeCommit?

You can use Git to import any existing Git repository to AWS CodeCommit. For other repositories, such as Subversion and Perforce, you can use a Git importer to first migrate it to a Git repository. For step by step instructions on importing Git repositories, see [Migrate an Existing Repository to AWS CodeCommit.](http://docs.aws.amazon.com/codecommit/latest/userguide/how-to-migrate-repository.html) For step-by-step instructions on importing local or unversioned content, see the [Git migration documentation](http://git-scm.com/book/en/v2/Git-and-Other-Systems-Migrating-to-Git).

Q: What is the maximum size for a single file that I can store in CodeCommit?

A single file in a repository cannot be more than 2 GB in size.

Q: How do I backup my repository?

If you have a local copy of the repository from doing a full&nbsp;git clone, you can use that to restore data. If you want additional backups, there are multiple ways to do so. One way is to install Git on your backup server and run a scheduled job that uses the git clonecommand to take regular snapshots of your repository. You can use git pull instead of git clone if you want to copy only the incremental changes. Note that these operations may incur an additional user and/or request charges based on how you setup the backup server and the polling frequency.

Q: How do I restore a deleted AWS CodeCommit repository?

Deleting an AWS CodeCommit repository is a destructive one-way operation that cannot be undone. To restore a deleted repository, you will need to create the repository again and use either a backup or a local copy from a full clone to upload the data. We recommend using IAM policies along with MFA-protection to restrict users who can delete repositories. For more details, see the Can I use AWS Identity and Access Management (IAM) to manage access to AWS CodeCommit? question in the Security section of the FAQ.

Q: How do I create webhooks using AWS CodeCommit?

In the Amazon SNS console, you can create an SNS topic with an HTTP endpoint and the desired URL for the webhook. From the AWS CodeCommit console, you can then configure that SNS topic to a repository event using triggers. Additionally, customers using [AWS Chatbot](https://aws.amazon.com/chatbot/) can configure notifications to be sent to their Slack Channels or Amazon Chime chat rooms. For more details please visit [here](https://docs.aws.amazon.com/codestar-notifications/latest/userguide/notifications-chatbot.html).

Q: Can I get a history of AWS CodeCommit Git operations and API calls made in my account for security analysis and operational troubleshooting purposes?

Yes. You can review recent CodeCommit events, including Git operations and API calls, in the AWS CloudTrail console. For an ongoing record of events you can create a trail and log events in an Amazon S3 bucket. For more information, see Logging AWS CodeCommit API Calls with AWS CloudTrail.

Q: Can I use AWS Identity and Access Management (IAM) to manage access to AWS CodeCommit?

Yes. AWS CodeCommit supports [resource-level permissions](http://docs.aws.amazon.com/IAM/latest/UserGuide/PermissionsOverview.html). For each AWS CodeCommit repository, you can specify which users can perform which actions. You can also specify AWS multi-factor authentication (MFA) for a CodeCommit action. This allows you to add an extra level of protection for destructive actions such as deleting repositories. In addition to the AWS CodeCommit APIs, you can also specify git pull and git push as actions to control access from Git clients. For example, you can create a read-only user for a repository by allowing that user access to git pull but not git push on the repository. For more information on using IAM with AWS CodeCommit, see [Authentication and Access Control for AWS CodeCommit](https://docs.aws.amazon.com/codecommit/latest/userguide/auth-and-access-control.html). For more information on authenticating API access using MFA, see [Configuring MFA-Protected API Access](http://docs.aws.amazon.com/IAM/latest/UserGuide/MFAProtectedAPI.html).

Q: What communication protocols are supported by AWS CodeCommit?

You can use either the HTTPS or SSH protocols or both to communicate with AWS CodeCommit. To use HTTPS, first install the AWS CLI. The AWS CLI installs a Git credential helper that can be configured with AWS credentials. It automatically signs all HTTPS requests to AWS CodeCommit using the [Signature Version 4](http://docs.aws.amazon.com/general/latest/gr/signature-version-4.html) signing specification. To use SSH, users create their own public-private key pairs and add their public keys to their IAM users. The private key encrypts the communication with AWS CodeCommit. For step-by-step instructions on setting up HTTPS and SSH access, see the [Setting up AWS CodeCommit](http://docs.aws.amazon.com/codecommit/latest/userguide/setting-up.html) page.

Q: What ports should I open in my firewall for access to AWS CodeCommit?

You will have to open outbound access to an AWS CodeCommit service endpoint on port 22 (SSH) or port 443 (HTTPS).

Q: How do I encrypt my repository in AWS CodeCommit?

Repositories are automatically encrypted at rest. No customer action is required. AWS CodeCommit uses AWS Key Management Service (KMS) to encrypt repositories. When you create your first repository, an AWS-managed CodeCommit key is created under your AWS account. For details, see [Encryption for AWS CodeCommit Repositories](http://docs.aws.amazon.com/codecommit/latest/userguide/encryption.html).

Q: Can I enable cross-account access to my repository?

Yes. You can create an IAM role in your AWS account to delegate access to a repository to IAM users in other AWS accounts. The IAM users can then configure their AWS CLI to use AWS Security Token Service (STS) and assume the role when running commands. For details see [Assuming a Role](http://docs.aws.amazon.com/cli/latest/userguide/cli-roles.html) in the AWS CLI documentation.

**CODEBUILD**

"Bundle the dependencies in the source code during the last stage of CodeBuild"

AWS CodeBuild is a fully managed build service. There are no servers to provision and scale, or software to install, configure, and operate.

A typical application build process includes phases like preparing the environment, updating the configuration, downloading dependencies, running unit tests, and finally, packaging the built artifact.

Downloading dependencies is a critical phase in the build process. These dependent files can range in size from a few KBs to multiple MBs. Because most of the dependent files do not change frequently between builds, you can noticeably reduce your build time by caching dependencies.

This will allow the code bundle to be deployed to Elastic Beanstalk to have both the dependencies and the code, hence speeding up the deployment time to Elastic Beanstalk

**Enable CodeBuild timeouts**

A build represents a set of actions performed by AWS CodeBuild to create output artifacts (for example, a JAR file) based on a set of input artifacts (for example, a collection of Java class files).

The following rules apply when you run multiple builds:

When possible, builds run concurrently. The maximum number of concurrently running builds can vary.

Builds are queued if the number of concurrently running builds reaches its limit. The maximum number of builds in a queue is five times the concurrent build limit.

A build in a queue that does not start after the number of minutes specified in its time out value is removed from the queue. The default timeout value is eight hours. You can override the build queue timeout with a value between five minutes and eight hours when you run your build.

By setting the timeout configuration, the build process will automatically terminate post the expiry of the configured timeout.

Q: Can I use CodeBuild to automate my release process?

Yes. CodeBuild is integrated with AWS CodePipeline. You can add a build action and set up a continuous integration and continuous delivery process that runs in the cloud. You can learn how to set up and monitor your builds from the CodePipeline console [here](https://docs.aws.amazon.com/codebuild/latest/userguide/how-to-create-pipeline.html).

Q: Which source repositories does CodeBuild support?

CodeBuild can connect to AWS CodeCommit, S3, GitHub, and GitHub Enterprise and Bitbucket to pull source code for builds.

Q: What happens when a build is run?

CodeBuild will create a temporary compute container of the class defined in the build project, load it with the specified runtime environment, download the source code, execute the commands configured in the project, upload the generated artifact to an S3 bucket, and then destroy the compute container. During the build, CodeBuild will stream the build output to the service console and Amazon CloudWatch.

Q: Can I use CodeBuild with Jenkins?

Yes. The [CodeBuild Plugin for Jenkins](https://github.com/awslabs/aws-codebuild-jenkins-plugin) can be used to integrate CodeBuild into Jenkins jobs. The build jobs are sent to CodeBuild, eliminating the need for provisioning and managing the Jenkins worker nodes.

Q: How can I view past build results?

You can access your past build results through the console, CloudWatch, or the API. The results include outcome (success or failure), build duration, output artifact location, and log location. With the CodeBuild dashboard, you can view metrics to understand build behavior over time. The dashboard displays number of builds attempted, succeeded, and failed, as well as build duration. You can also visit the CloudWatch console to view more detailed build metrics. To learn more about monitoring CodeBuild with CloudWatch, visit our documentation.

Q: How can I debug a past build failure?

You can debug a build by inspecting the detailed logs generated during the build run or you can use [CodeBuild Local](https://aws.amazon.com/blogs/devops/announcing-local-build-support-for-aws-codebuild/" \t "_blank) to locally test and debug your builds.

Q: How do I receive notifications or alerts for any events in AWS CodeBuild?

You can create notifications for events impacting your build projects. Notifications will come in the form of [Amazon SNS](https://aws.amazon.com/sns/) notifications. Each notification will include a status message as well as a link to the resources whose event generated that notification. Notifications has no additional cost; but, you may be charged for other AWS services utilized by notifications, such as Amazon SNS. To learn how to get started with notifications, see the [notifications user guide](https://docs.aws.amazon.com/codestar-notifications/latest/userguide/welcome.html). Additionally, customers using [AWS Chatbot](https://aws.amazon.com/chatbot/) can configure notifications to be sent to their Slack Channels or Amazon Chime chat rooms. For more details please check [here](https://docs.aws.amazon.com/codestar-notifications/latest/userguide/notifications-chatbot.html).

**CODE DEPLOY**

Q: How is AWS CodeDeploy different from other AWS deployment and management services such as AWS Elastic Beanstalk and AWS OpsWorks?  
AWS CodeDeploy is a building block service focused on helping developers deploy and update software on any instance, including Amazon EC2 instances and instances running on-premises. AWS Elastic Beanstalk and AWS OpsWorks are end-to-end application management solutions

Q: What is an application?  
An application is a collection of software and configuration to be deployed to a group of instances. Typically, the instances in the group run the same software. For example, if you have a large distributed system, the web tier will likely constitute one application and the data tier another application.

Q: : What is a revision?  
A revision is a specific version of deployable content, such as source code, post-build artifacts, web pages, executable files, and deployment scripts, along with an [AppSpec](http://docs.aws.amazon.com/codedeploy/latest/userguide/writing-app-spec.html" \t "_blank) file. The AWS CodeDeploy Agent can access a revision from GitHub or an Amazon S3 bucket.

Q: What is a deployment group?  
A deployment group is the AWS CodeDeploy entity for grouping EC2 instances or AWS Lambda functions in a CodeDeploy deployment. For EC2 deployments, it is a set of instances associated with an application that you target for a deployment. You can add instances to a deployment group by specifying a tag, an Auto Scaling group name, or both. In an AWS Lambda deployment, a deployment group defines a set of AWS CodeDeploy configurations for future serverless Lambda deployment to the group, like alarms and rollbacks.

You can define multiple deployment groups for an application such as staging and production. For information on tags, see [Working with Amazon EC2 Tags in the Console](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html#Using_Tags_Console). For more information on deploying to Auto Scaling groups, see [Auto Scaling Integration](http://docs.aws.amazon.com/codedeploy/latest/userguide/auto-scaling-integ.html).

Q: What is a deployment configuration?  
A deployment configuration specifies how the behavior for how deployment should proceed, including how to handle deployment failure, through for a deployment group. You can use a deployment configuration to perform zero-downtime deployments to multi-instance deployment groups. For example, if your application needs at least 50% of the instances in a deployment group to be up and serving traffic, you can specify that in your deployment configuration so that a deployment does not cause downtime. If no deployment configuration is associated with either the deployment or the deployment group, then by default AWS CodeDeploy will deploy to one instance at a time.  For more information on deployment configuration, see [Instance Health](http://docs.aws.amazon.com/codedeploy/latest/userguide/host-health.html).

Q: What are the parameters that I need to specify for a deployment?  
There are three parameters you specify for a deployment:

1. Revision - Specifies what to deploy.
2. Deployment group - Specifies where to deploy.
3. Deployment configuration - An optional parameter that specifies how to deploy.

Q: What changes do I need to make to my code to deploy using AWS CodeDeploy?  
You don’t need to make any changes to your code. You simply add a configuration file (called an [AppSpec](http://docs.aws.amazon.com/codedeploy/latest/userguide/writing-app-spec.html" \t "_blank) file) in the root directory of your revision bundle that specifies the files to be copied and scripts to be executed.

Q: How can I deploy an application from my source control system using AWS CodeDeploy?  
If you are using GitHub, you can deploy a revision in a .zip, .tar, or .tar.gz format from your repository directly to instances. For other source control systems, you can bundle and upload the revision to an Amazon S3 bucket in a .zip, .tar, or .tar.gz format and specify the Amazon S3 location when doing a deployment. If your application needs a build step, make sure that the GitHub repository or the Amazon S3 bucket contains the post-build artifacts. For more information on using GitHub with AWS CodeDeploy, see our [product integrations](https://aws.amazon.com/codedeploy/product-integrations/) page. For more information on using Amazon S3 for storing revisions, see [Push a Revision](http://docs.aws.amazon.com/codedeploy/latest/userguide/how-to-push-revision.html).

Q: How does AWS CodeDeploy work with Auto Scaling?  
You can associate an Auto Scaling group with a deployment group to make sure that newly launched instances always get the latest version of your application. Every time a new Amazon EC2 instance is launched for that Auto Scaling group, it will be first put in a Pending state and a deployment of the last successful revision for that deployment group triggered on that Amazon EC2 instance. If the deployment completes successfully, the state of the Amazon EC2 instance is changed to InService. If that deployment fails, the Amazon EC2 instance is terminated, a new Amazon EC2 instance is launched in Pending state, and a deployment triggered for the newly launched EC2 instance. For more information on Auto Scaling group instance lifecycle events, see [Auto Scaling Group Lifecycle](http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/AutoScalingGroupLifecycle.html).

Q: Can I stop an in-flight deployment?  
Yes. When you stop an in-flight deployment, the AWS CodeDeploy service will instruct the agent on each instance to stop executing additional scripts. To get your application back to a consistent state, you can either redeploy the revision, or deploy another revision.

Q: How do I receive notifications or alerts for any events in AWS CodeDeploy?  
You can create notifications for events impacting your deployments. Notifications will come in the form of [Amazon SNS](https://aws.amazon.com/sns/) notifications. Each notification will include a status message as well as a link to the resources whose event generated that notification. Notifications has no additional cost; but, you may be charged for other AWS services utilized by notifications, such as Amazon SNS. To learn how to get started with notifications, see the [notifications user guide](https://docs.aws.amazon.com/codestar-notifications/latest/userguide/welcome.html). Additionally, customers using [AWS Chatbot](https://aws.amazon.com/chatbot/) can configure notifications to be sent to their Slack Channels or Amazon Chime chat rooms. For more details please check [here](https://docs.aws.amazon.com/codestar-notifications/latest/userguide/notifications-chatbot.html).

Q: Can I use AWS CodeDeploy to deploy an application to Amazon EC2 instances running within an Amazon Virtual Private Cloud (VPC)?  
Yes, but the AWS CodeDeploy agent installed on the Amazon EC2 instances must be able to access the public AWS CodeDeploy and Amazon S3 service endpoints. For more information, see [AWS CodeDeploy Endpoints](http://docs.aws.amazon.com/general/latest/gr/rande.html#codedeploy_region) and [Amazon S3 Endpoints](http://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region).

You can specify one or more deployment groups for a CodeDeploy application. The deployment group contains settings and configurations used during the deployment. Most deployment group settings depend on the compute platform used by your application. Some settings, such as rollbacks, triggers, and alarms can be configured for deployment groups for any compute platform.