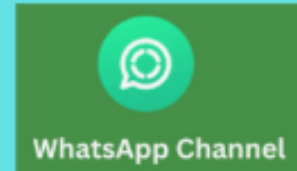# AWS CodeCommit: Interview Questions

What is AWS CodeCommit and how is it different from GitHub or Bitbucket?

https://aws.amazon.com/blogs/devops/how-to-migrate-your-aws-codecommit-repository-to-another-git-provider/

AWS CodeCommit is a **fully-managed source control service** from AWS that hosts **secure, scalable, private** Git repositories. It helps teams collaborate on code without managing their own Git servers.

🔍 **Key Features**

- **Fully Managed**: No need to manage servers, patching, or scaling.
- **Secure**:
  - Encryption at rest and in transit
  - IAM-based access control
- **Scalable**: Handles large repos, branches, histories—no artificial size/file limits.
- **Git-Compatible**: Works with standard Git commands and tools.
- **AWS Integration**: Natively integrates with CodePipeline, CodeBuild, CodeDeploy, Lambda, and more.

🔄 CodeCommit vs GitHub vs Bitbucket

Feature	AWS CodeCommit	GitHub	Bitbucket
Hosting Model	Fully managed on AWS	SaaS, GitHub Enterprise for on-prem	SaaS, Data Center for on-prem
Integration	Deep with AWS services	GitHub Actions, Packages, 3rd-party tools	Jira, Confluence, Trello, Bamboo
Audience	AWS-focused teams	Open-source & enterprise	Atlassian users & small teams
Private Repos	Private by default	Free/private with limits	Free private repos for up to 5 users
CI/CD	CodePipeline, CodeBuild, CodeDeploy	GitHub Actions, many integrations	Bitbucket Pipelines, Jenkins, Bamboo
Code Review	Pull requests, approvals	Robust discussions & reviews	Full code review flow
Community	Small, AWS-centric	Largest dev community, open-source hub	Strong in enterprise and Atlassian ecosystem
UI/UX	Functional, AWS-style	Polished, intuitive	Friendly for teams
Cost	Pay-as-you-go, free for 5 active users	Tiered pricing, some free usage	Free for small teams, then per-user pricing

How do you configure access to a CodeCommit repository using IAM roles or credentials?

1. IAM User with Git Credentials (HTTPS)
- Create IAM user with AWSCodeCommitPowerUser or custom policy.
- Generate Git credentials in IAM console.

- Use the credentials for git clone or configure Git credential helper.

2. IAM User with SSH Keys
- Generate SSH keys.
- Upload public key to IAM under "SSH keys for AWS CodeCommit".
- Use SSH URL for cloning.

3. AWS CLI Credential Helper (Recommended)
- Install and configure AWS CLI with IAM credentials.
- Run:

```
git config --global credential.helper '!aws codecommit credential-helper $@'
```
- Clone using HTTPS URL; CLI handles credentials.

4. IAM Roles for AWS Services
- Create role (e.g., CodeBuildServiceRole) with necessary CodeCommit permissions.
- Attach the role to:
    - EC2: via instance profile
    - CodeBuild/CodePipeline: in service settings

🛡️ *Apply the principle of least privilege when granting permissions.*

How do you migrate from Git/GitHub/GitLab to CodeCommit?

Migration Steps:
1. **Create empty CodeCommit repo**
2. **Mirror existing repo locally:**

```bash
------
git clone --mirror https://github.com/user/repo.git
cd repo.git
```

3. **Add CodeCommit as remote:**

```bash
------
git remote add codecommit https://git-codecommit.region.amazonaws.com/v1/repos/MyRepo
```

4. **Push all data:**

```bash
------
git push codecommit --all
git push codecommit --tags
```

🔄 **Post-Migration Checklist:**
- Verify branches, commits
- Clean up local repo (rm -rf repo.git)
- Update webhooks/integrations

- Handle Git LFS if used

A. CodeCommit will be permanently shut down

B. New customer access will be closed

C. CodeCommit will become a paid-only service

D. All repositories will be auto-migrated to GitHub

✅ **Correct Answer: B. New customer access will be closed**