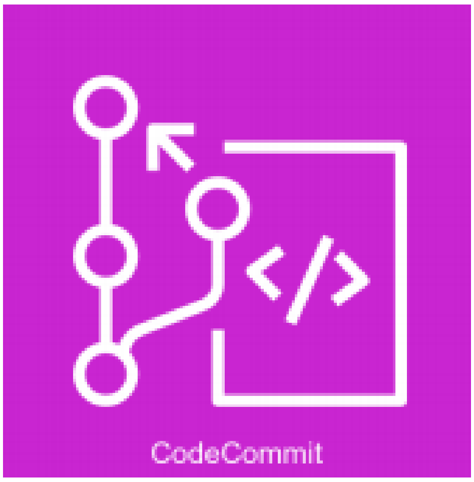
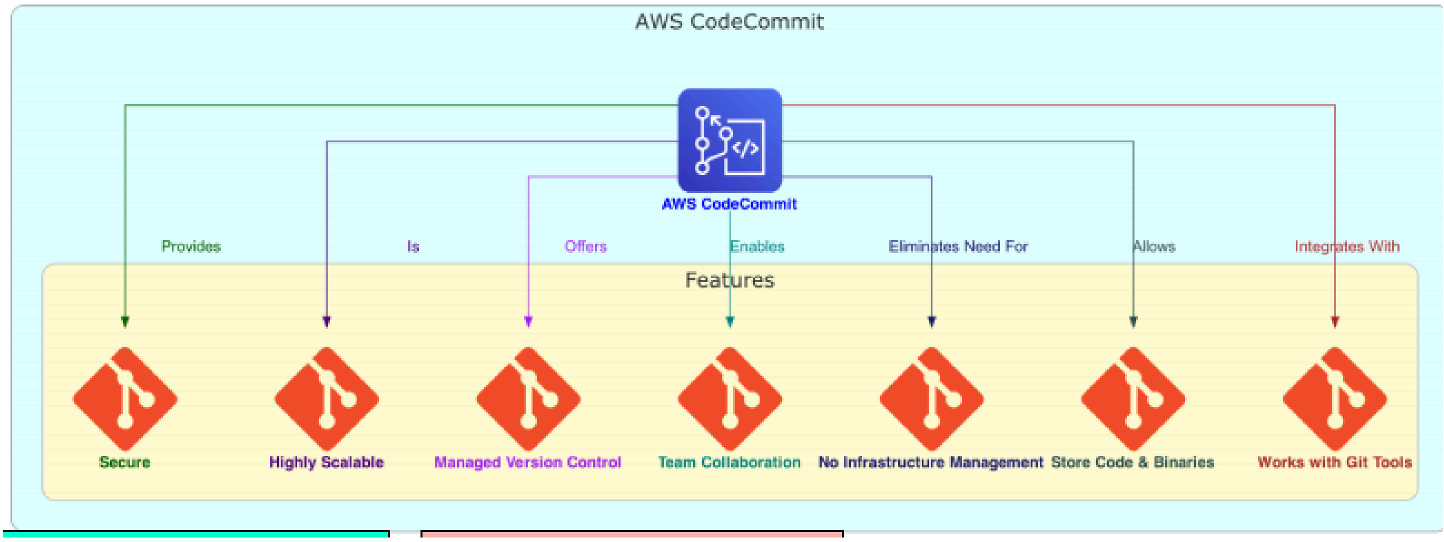
🔔 Follow me SHAIK HARI SADIA ANJUM for more AWS content! 🚀

CodeCommit



AWS CodeCommit is a fully managed Git-based source control service that allows teams to securely store and manage their code repositories in AWS. It provides high availability, scalability, and security, eliminating the need to host and manage private Git repositories manually.

- 🔹 **Similar to GitHub, GitLab, and Bitbucket but designed specifically for AWS integration.**
- 🔹 **Supports Git commands, making it easy to use for developers familiar with Git.**
- 🔹 **Provides encryption, access control, and AWS IAM integration for security.**



Why Use AWS CodeCommit?

✅ **Fully Managed** – No need to maintain servers or backups.
✅ **Highly Scalable** – Supports unlimited repositories and files of any size.
✅ **Secure** – Uses **AWS IAM** for authentication and fine-grained access control.
✅ **AWS Integration** – Works seamlessly with **AWS CodeBuild, CodePipeline, CodeDeploy, Lambda, EC2, and more**.

✅ **Automatic Backups** – AWS automatically replicates repositories across **multiple AWS Availability Zones**.
✅ **No Size Limits** – Unlike GitHub/GitLab, **CodeCommit has no storage restrictions**.
✅ **Faster Performance** – Designed for low-latency and high-throughput Git operations.
✅ **Cost-Effective** – Free tier includes **5 active users per month**, with additional users at **$1 per month**.

Key Features of AWS CodeCommit

🔹 1. Secure and Managed Git Repository

- Supports **private Git repositories** with AWS security and compliance.
- Uses **AWS IAM permissions** for **fine-grained access control**.
- **Encrypts data at rest and in transit** using AWS KMS.

🔹 2. Seamless AWS Integration

- Works with **AWS CodeBuild, CodePipeline, and CodeDeploy** for CI/CD.
- Triggers **AWS Lambda** functions on repository changes.
- Supports integration with **Amazon CloudWatch Events** for automation.

🔹 3. Collaboration & Access Control

- **IAM roles & policies** define user access at the **repository level**.
- **Fine-grained permissions** allow read/write control over specific branches.
- Supports **pull requests, code reviews, and approvals**.

🔹 4. High Performance & Scalability

- **Optimized for AWS infrastructure** → Lower latency, faster commits, and efficient large file handling.
- Supports **unlimited repositories** and **unlimited file storage**.

🔹 5. DevOps & CI/CD Automation

- **Webhooks & triggers** allow integration with CI/CD pipelines.
- Supports **automated testing** with AWS CodeBuild.
- Automates **deployment** using AWS CodeDeploy & CodePipeline.

How AWS CodeCommit Works

Step 1: Create a Repository

1. Open **AWS Console** → Navigate to **AWS CodeCommit**.
2. Click **Create repository**.
3. Enter **repository name** and optional description.
4. Click **Create** → Copy the **repository URL**.

Step 2: Clone the Repository

- Use Git CLI to clone the repository locally:

```
git clone https://git-codecommit.<region>.amazonaws.com/v1/repos/MyRepo
```

AWS IAM authentication is required via:

- **HTTPS (AWS credential helper)**
- **SSH key authentication**

Step 3: Add and Commit Code

1. Navigate to the local repository:

```
cd MyRepo
```

2.Add a file and commit:

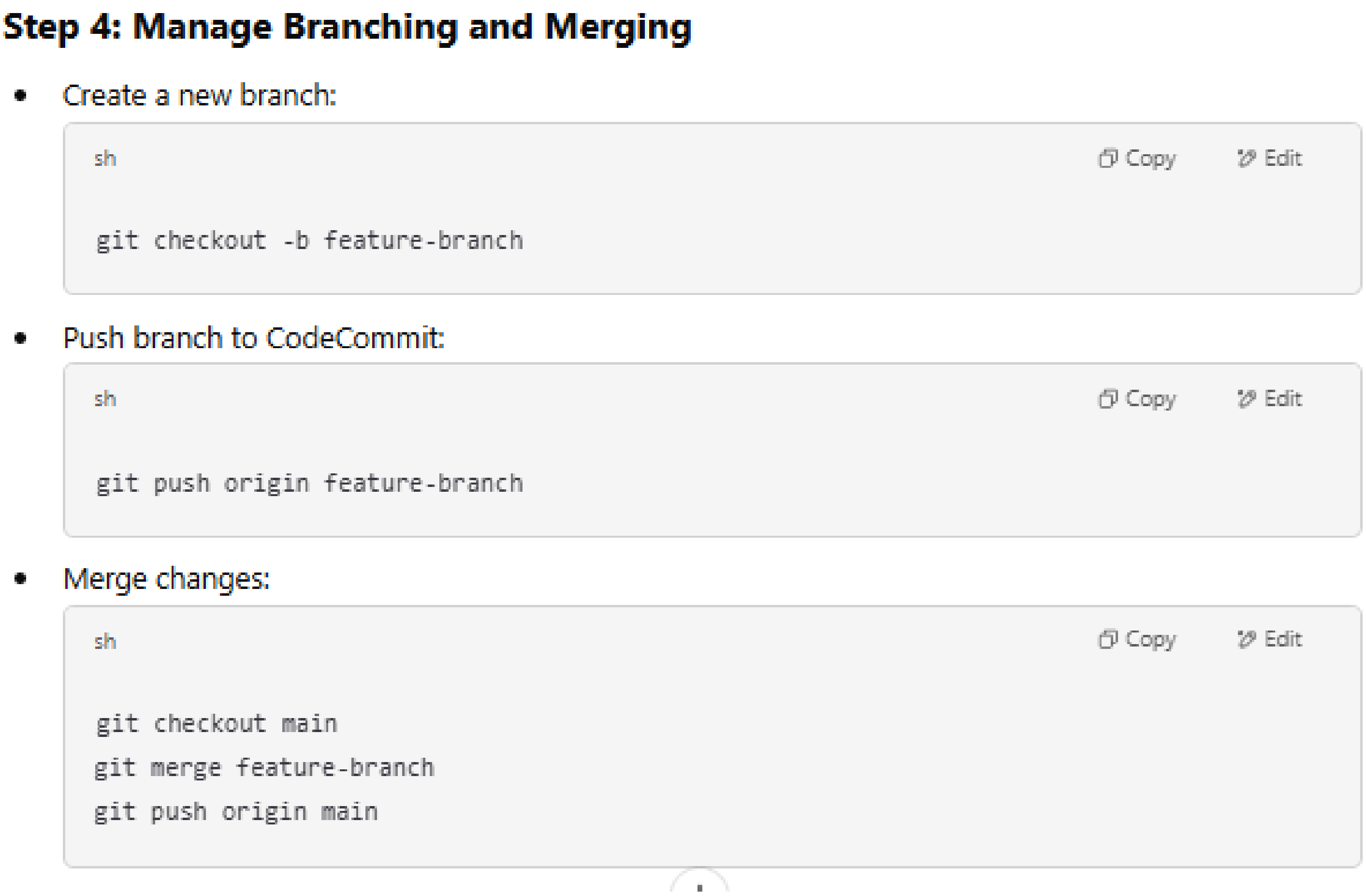
```
echo "Hello CodeCommit" > README.md

git add README.md

git commit -m "Initial commit"
```

3.Push the changes:

```
git push origin main
```



Step 4: Manage Branching and Merging

- Create a new branch:

```sh
git checkout -b feature-branch
```

- Push branch to CodeCommit:

```sh
git push origin feature-branch
```

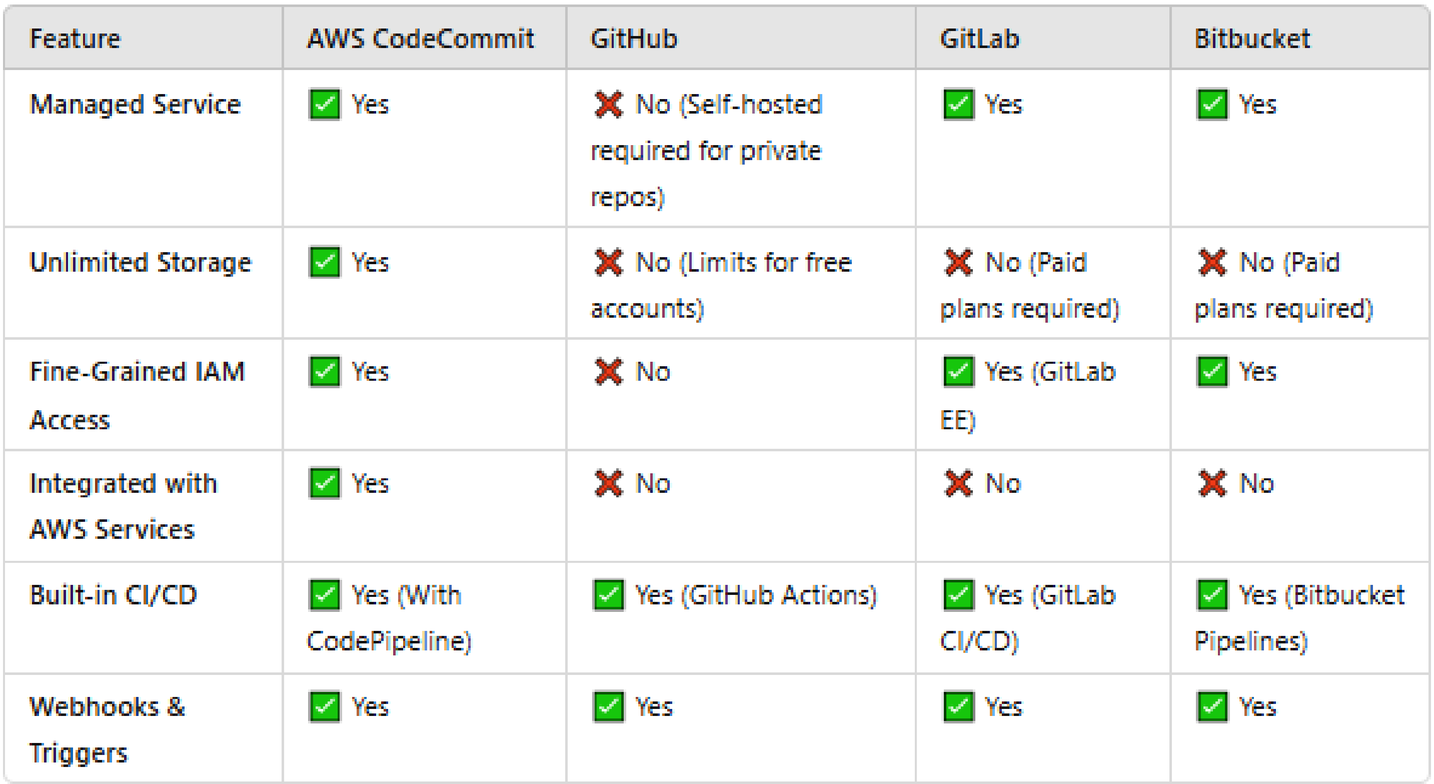
- Merge changes:

```sh
git checkout main
git merge feature-branch
git push origin main
```

Step 5: Enable CI/CD with AWS CodePipeline

- Create an AWS **CodePipeline** to automatically pull code from CodeCommit and deploy it.
- Set up **AWS CodeBuild** to build the application.
- Use **AWS CodeDeploy** to deploy changes to EC2, ECS, or Lambda.

AWS CodeCommit vs Other Git Repositories



Feature	AWS CodeCommit	GitHub	GitLab	Bitbucket
Managed Service	✅ Yes	❌ No (Self-hosted required for private repos)	✅ Yes	✅ Yes
Unlimited Storage	✅ Yes	❌ No (Limits for free accounts)	❌ No (Paid plans required)	❌ No (Paid plans required)
Fine-Grained IAM Access	✅ Yes	❌ No	✅ Yes (GitLab EE)	✅ Yes
Integrated with AWS Services	✅ Yes	❌ No	❌ No	❌ No
Built-in CI/CD	✅ Yes (With CodePipeline)	✅ Yes (GitHub Actions)	✅ Yes (GitLab CI/CD)	✅ Yes (Bitbucket Pipelines)
Webhooks & Triggers	✅ Yes	✅ Yes	✅ Yes	✅ Yes

AWS CodeCommit Pricing

AWS CodeCommit **offers a free tier**:

- **5 active users per month FREE**
- **$1 per additional user per month**

No additional costs for: ✓ Unlimited repositories
✓ Unlimited data storage
✓ High availability

Example Cost Calculation:

- If **3 users** use CodeCommit, it's **FREE**.

- If **10 users**, cost = **(10 - 5) × $1 = $5/month**.

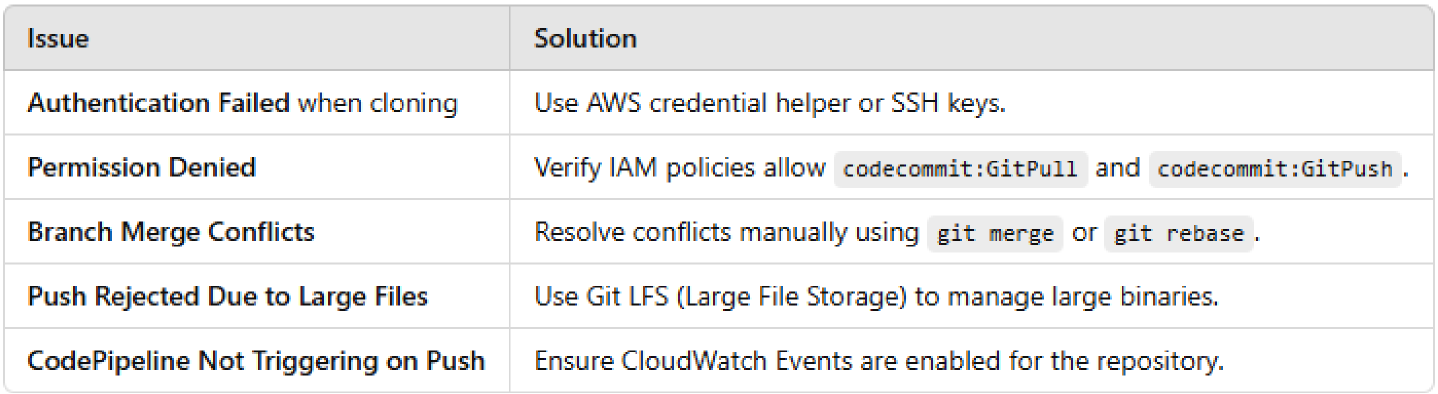
Best Practices for AWS CodeCommit

✅ **Use IAM for Access Control** – Assign least-privileged permissions to users.
✅ **Enable Encryption** – Use **AWS KMS** to encrypt repository data.
✅ **Use Multi-Factor Authentication (MFA)** – For securing repository access.
✅ **Automate CI/CD Pipelines** – Use **CodePipeline + CodeBuild** for automation.
✅ **Set Up Webhooks & Triggers** – Automate builds, notifications, and deployments.
✅ **Use Branching Strategies** – Follow Git best practices (**feature branches, main/master, develop**).
✅ **Monitor Repository Activity** – Use **AWS CloudTrail** to track who accesses and modifies repositories.

Common Use Cases

📌 **Enterprise-Grade Git Repository** – Secure alternative to GitHub for AWS users.
📌 **CI/CD Pipelines** – Automate testing and deployment using AWS CodePipeline.
📌 **Serverless Application Deployment** – Use CodeCommit with Lambda & API Gateway.
📌 **Infrastructure as Code (IaC)** – Store Terraform/CloudFormation templates.
📌 **Collaboration on Private Projects** – Securely manage repositories for internal development teams.

Common Issues & Troubleshooting



Issue	Solution
Authentication Failed when cloning	Use AWS credential helper or SSH keys.
Permission Denied	Verify IAM policies allow `codecommit:GitPull` and `codecommit:GitPush`.
Branch Merge Conflicts	Resolve conflicts manually using `git merge` or `git rebase`.
Push Rejected Due to Large Files	Use Git LFS (Large File Storage) to manage large binaries.
CodePipeline Not Triggering on Push	Ensure CloudWatch Events are enabled for the repository.

Conclusion

AWS CodeCommit is a **secure, scalable, and fully managed Git-based source control system** that integrates seamlessly with AWS services for CI/CD automation. It is ideal for organizations already using AWS, offering **IAM-based access control, automated triggers, and seamless integration with AWS DevOps tools**.