**AWS CODE-COMMIT**

Version control is the ability to understand the various changes that happened to the code over time and possibly roll back

All these can be enabled by using version control system such as git

Central repo you can use github or codecommit

**Benefits**:

* + Collaborate with other developers
  + Store your code securely
  + Easily scale your version control projects
  + Store anything, anytime
  + Integrate with other AWS and third-party services.
  + Easily migrate files from other remote repositories
  + Use the Git tools you already know
  + Make sure the code is backed up somewhere
  + Make sure its fully viewable and auditable

Central repo can be expensive

Github: free public but paid private repo

Bitbucket

**AWS CodeCommit**

* Private git repository
* No size limit on repo [scale seamlessly]
* Fully managed high availability
* Code only in AWS Cloud Account ==🡺 increased security and compliance
* Secure [encryption, access control]
* Can be integrated with codebuild, Jenkins and other CI tools

It’s a way to create git repository in AWS and its advatnages over other service like github is to be able to have your code privately held your repo within your vpc.

You are signed in using a root account. You cannot configure SSH connections for a root account, and HTTPS connections for a root account are not recommended. Consider signing in as an IAM user and then setting up your connection.

**How AWS CodeCommit Works?**


        Typical CodeCommit workflow
      

**AWS CodeCommit First Repo & HTTPS**

1] Create IAM user and attach the policy and generate HTTPS git credentials for codecommit or generate for ssh.

2] Install git in local machine

3] Create repo in codecommit and clone the same through git --🡪 pass https git credentials

4] Clone add commit push…..

5] create branch, commit locally and push the branch to remote, then once your are ready for merge, create pull request to merge the branch into master

6] **To protect merge directly, you can attach policy to developers group to deny the developers to merge their brances directly to master.**

**IAM …..GROUPS[DEVELOPER]…PERMISSION…INLINE POLICY…..CUSTOM….CREATE POLICY**

**Deny git push pull merge delete every repo in every region for developers when condition eaqual to master branch and production branch.**



7] **Notifications**[ create….Event….Target] Target is a location for receiving notifications

CodeCommit…… settings…… create notification…..give notification name….. select Events {on commit, on pull request, on status changes, merge, branch} and finally invoke the targets [**SNS, CHATBOT**] submit

This notification rules get added in cloud watch automatically.

8] **Triggers**

CodeCommit……. Settings………. create triggers……….. select event and select targets [**SNS && LAMDA**] submit

Codecommit can be integrated with lamda and cloud watch