

Use a Bucket as Git repository and AWS CodePipeline source with git-remote-s3

Date:	2024-11-17
modified:	2024-11-17
tags:	AWS, S3, CodePipeline, Coding, Git, OpenSource
description:	How-to use a S3 Bucket as serverless git repository and source action for AWS CodePipeline with git-remote-s3
category:	Code
slug:	use-a-bucket-as-git-repository-and-aws-codepipeline-source-with-git-remote-s3
Author:	Dominik Wombacher
lang:	en
transid:	use-a-bucket-as-git-repository-and-aws-codepipeline-source-with-git-remote-s3
Status:	published

Let's talk about how to use a ordinary S3 Bucket as serverless Git repository with the remote helper script [git-remote-s3](#). Release [v0.2.0](#), contains a feature that let you use the repo as source for AWS CodePipeline as well.

I was looking for a way to replace AWS CodeCommit in a Workshop I build. The reasons: [New customer access was closed](#) (Archive: [\[1\]](#), [\[2\]](#)) on July 25, 2024. [git-remote-s3](#) looked promising for my use-case, I can use a Bucket as remote, push and pull code, leverage AWS IAM to grant access. The only challenge was AWS CodePipeline, more on that later.

First, how does [git-remote-s3](#) work under the hood? It is a so called [remote helper](#) script. And uses the [git bundle](#) feature. The bundle file is then then stored in the S3 bucket as `<prefix>/<ref>/<sha>.bundle`. [git-remote-s3](#) interacts with the bucket through the S3 API. It performs a get or put that either updates the bundle in the bucket or retrieves the file and adds the containing data to the local git clone. When a push was successful, the previous bundle is removed. There is only one bundle file per git `ref` at any point in time stored on the bucket.

To learn more, I recommend the [Under the hood](#) and [use S3 remotes](#) section in the [git-remote-s3 README](#).

[AWS CodePipeline](#) offers an [Amazon S3 source action](#) as location for your code and application files. But this requires to upload the source files as a single ZIP file. [git-remote-s3](#) can create and upload zip archives. Use `s3+zip` as URI Scheme when you add the remote and [git-remote-s3](#) will automatically place an archive on the S3 bucket that can be used by AWS CodePipeline.

You might wonder, where the archive file is located? Let's assume your bucket name is `my-git-bucket` and the repo is called `my-repo`. Run `git remote add origin s3+zip://my-git-bucket/my-repo` to use it as remote. When you now commit your changes and push to the remote, an additional `repo.zip` file will be uploaded to the bucket. For example, if you push to the main branch (`git push origin main`), the file is available under `s3://my-git-bucket/my-repo/refs/heads/main/repo.zip`. When you push to a branch called `fix_a_bug`, it's available under `s3://my-git-bucket/my-repo/refs/heads/fix_a_bug/repo.zip`. Or if you create and push a tag called `v1.0`, it will be `s3://my-git-bucket/my-repo/refs/tags/v1.0/repo.zip`.

Your AWS CodePipeline Action configuration, to trigger on updates of your `main` branch, would then look like this:

- Action Provider: `Amazon S3`
- Bucket: `my-git-bucket`
- S3 object key: `my-repo/refs/heads/main/repo.zip`
- Change detection options: `AWS CodePipeline`

Check out the Tutorial [Create a simple pipeline \(S3 bucket\)](#) to learn more about a S3 bucket as source action.

I'm proud that [my first contribution](#), to **git-remote-s3**, introduced the feature that allows using it together with AWS CodePipeline. And while I was working on it, I had the opportunity to make a couple more code improvements.

If you have any other use-case that requires a ZIP archive, then this feature will work for you too, it's not limited to AWS CodePipeline.