# CI PoC

## Jenkins

### Pros:

* Already familiar to most
* Supports coded job model with JenkinsFile -> version control, consistency
* It’s open source, so its free
* Lots of plugin, so it’s feature rich
* Good pipeline view

### Cons:

* Self Managed -> maintain and monitor build infrastructure our self.
* JenkinsFile is written in Groovy language
* Some people just don’t like Jenkins
* Does not use Docker by default but can be configured to do so
* Extending the native functionality of Jenkins is done through plugins. Plugins are expensive to maintain, secure, and upgrade.

## GitLabCI

### Pros:

* Simple – one config files used for build, test, deploy etc which is used to create the pipeline view
* Config file is YAML
* Provides fully managed holistic devops solution all in one gui – includes repo, merge requests, ci/cd process, infrastructure to deploy onto.
* Maintained by Gitlab with various pricing plans
* Uses docker to run tests and build etc, so can run these in parallel if needed
* Seamless integration with Kubernetes for prod deployments etc and supports serverless apps including monitoring
* Includes potentially useful tools such as Jira like task board and also and in browser IDE
* Can link github repo
* Includes a docker container registry
* Can store credentials in gui securely

### Cons:

* Premium pricing plans are relatively expensive
* May not integrate as easily with AWS services compared to codepipeline

## AWS CodePipeline

### Pros:

* YAML
* Part of AWS so, it integrates well with other AWS services
* Out of the box logging and monitoring with CloudWatch
* CloudWatch Events -> trigger lambda functions to notify or run custom task to fix problem etc
* Relatively cheap compared to premium GitlabCI prices
* Lots of useful resources available such as official articles, use case tutorials, code and cloudformation templates
* Uses Docker to run parallel builds like GitlabCI

### Cons:

* Pipeline view not as good -> buildfile is not split into separate notes unless you drill down -> more difficult to debug
* Consists of separate Build, Deploy and overseeing Pipeline tools. Which makes it more complicated than GitLabCI. There’s a separate script for Build and Deploy
* In my opinion, to get the most out of the suite there’s more of a learning curve
* Doesn’t feel as intuitive to setup because it’s split into config files and also AWS console. This only applies to DevOps engineers as the developers will only need to change the Buildspec.yml file which is simple. Note: Can create a cloudformation templates to setup the ‘AWS console’ part setup.