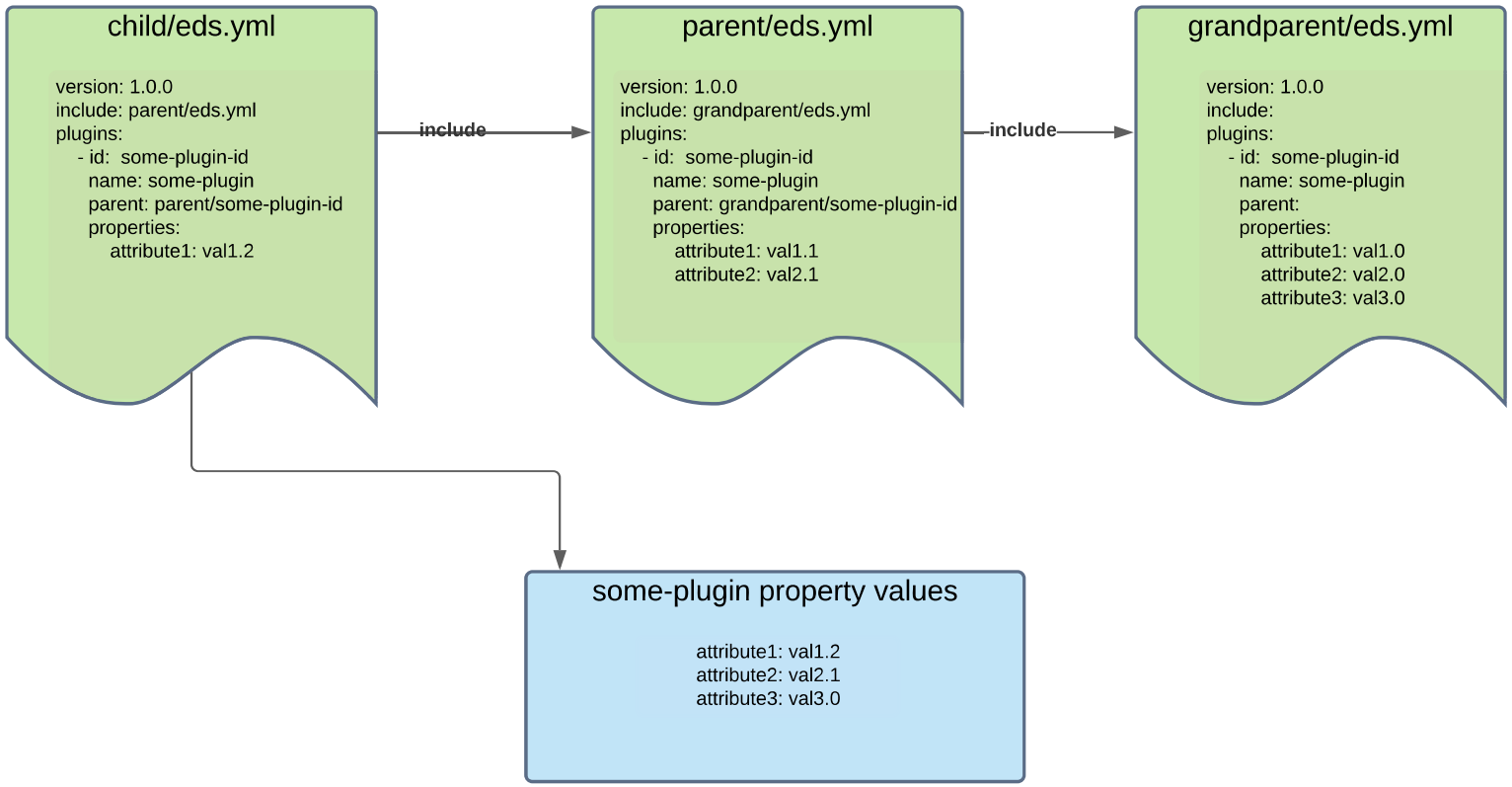
Extensible Deployment System (EDS)

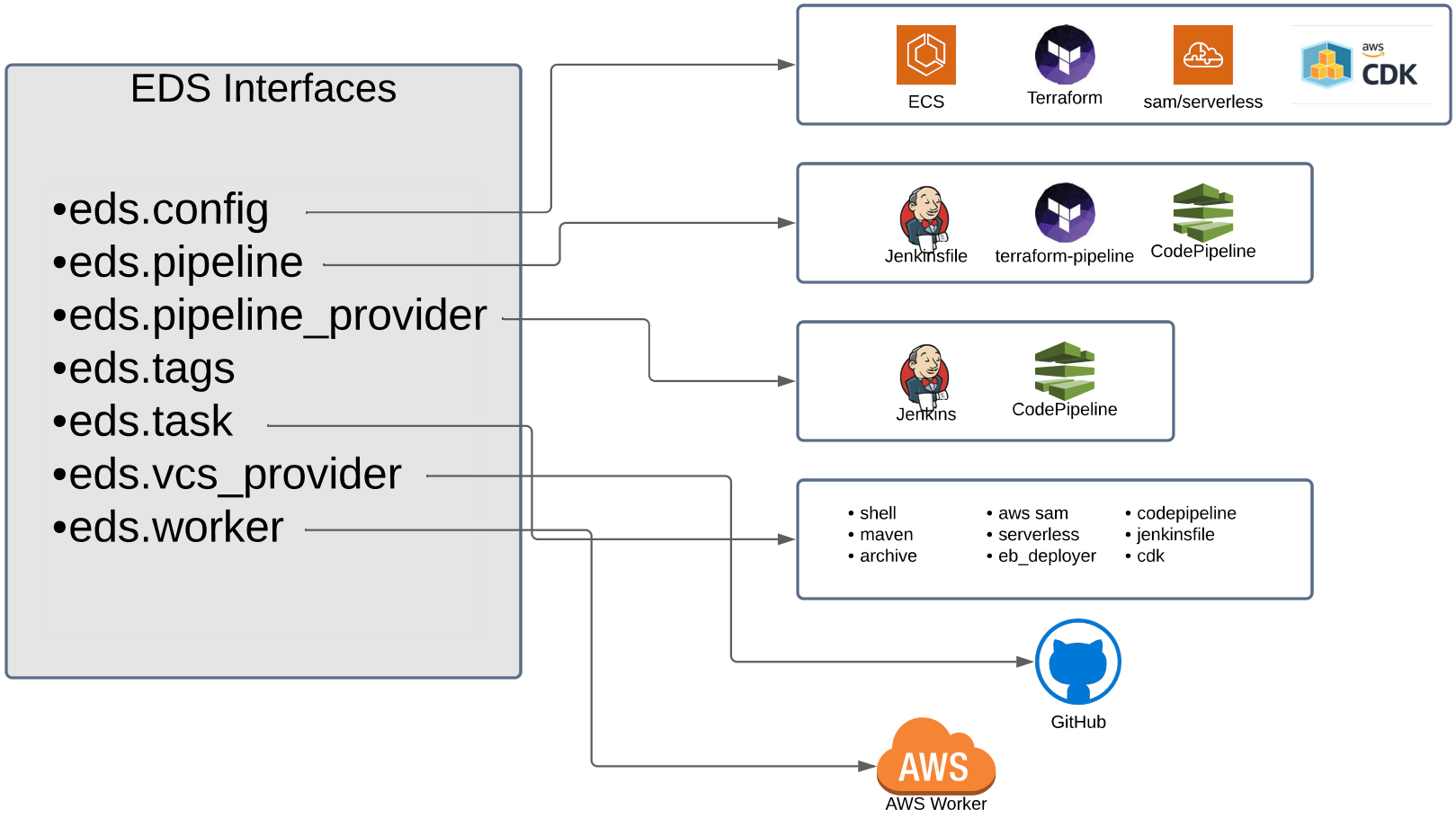
https://github.com/manheim/eds

- Configure projects using a single "eds.yml" file.
- Inherit from parent "eds.yml" files to stay DRY.
- Support multiple technologies using plugins.
- The EDS Worker handles the details for you.
- Open source.

DRY Config

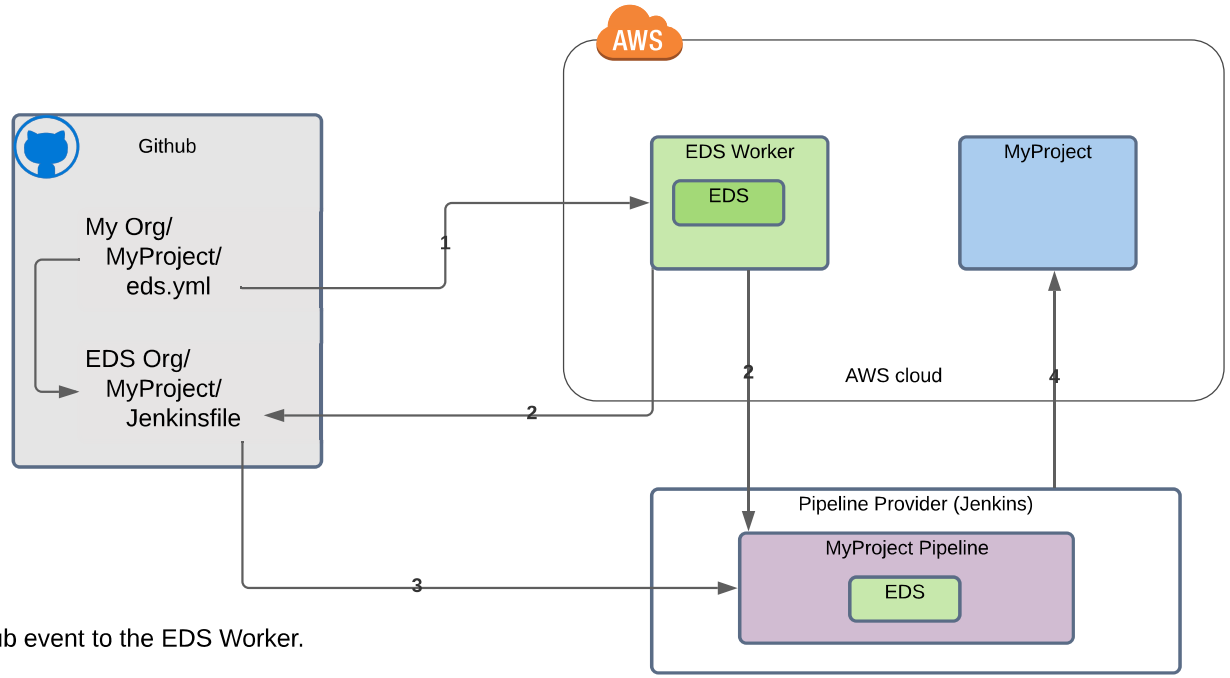

child/eds.yml
version: 1.0.0
include: parent/eds.yml
plugins:
- id: some-plugin-id
name: some-plugin
parent: parent/some-plugin-id
properties:
attribute1: val1.2
include
parent/eds.yml
version: 1.0.0
include: grandparent/eds.yml
plugins:
- id: some-plugin-id
name: some-plugin
parent: grandparent/some-plugin-id
properties:
attribute1: val1.1
attribute2: val2.1
include
grandparent/eds.yml
version: 1.0.0
include:
plugins:
- id: some-plugin-id
name: some-plugin
parent:
properties:
attribute1: val1.0
attribute2: val2.0
attribute3: val3.0
some-plugin property values
attribute1: val1.2
attribute2: val2.1
attribute3: val3.0

Plugins


EDS Interfaces
•eds.config
•eds.pipeline
•eds.pipeline_provider
•eds.tags
•eds.task
•eds.vcs_provider
•eds.worker
ECS
Terraform
sam/serverless
aws
CDK
Jenkinsfile
terraform-pipeline
CodePipeline
Jenkins
CodePipeline
• shell
• maven
• archive
• aws sam
• serverless
• eb_deployer
• codepipeline
• jenkinsfile
• cdk
GitHub
AWS
AWS Worker

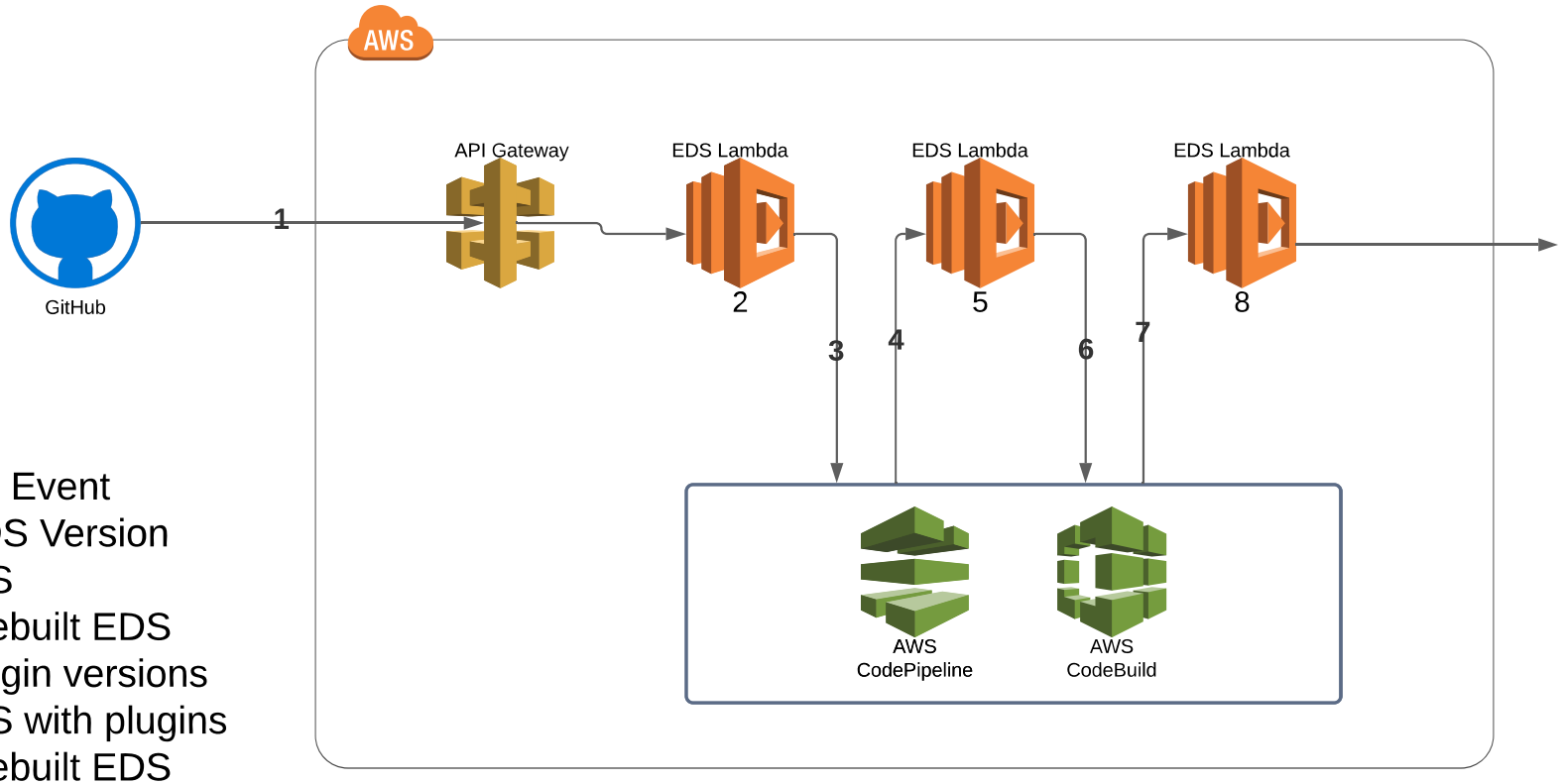
Workflow

(Using the Github, Jenkins, and AWS Worker plugins)



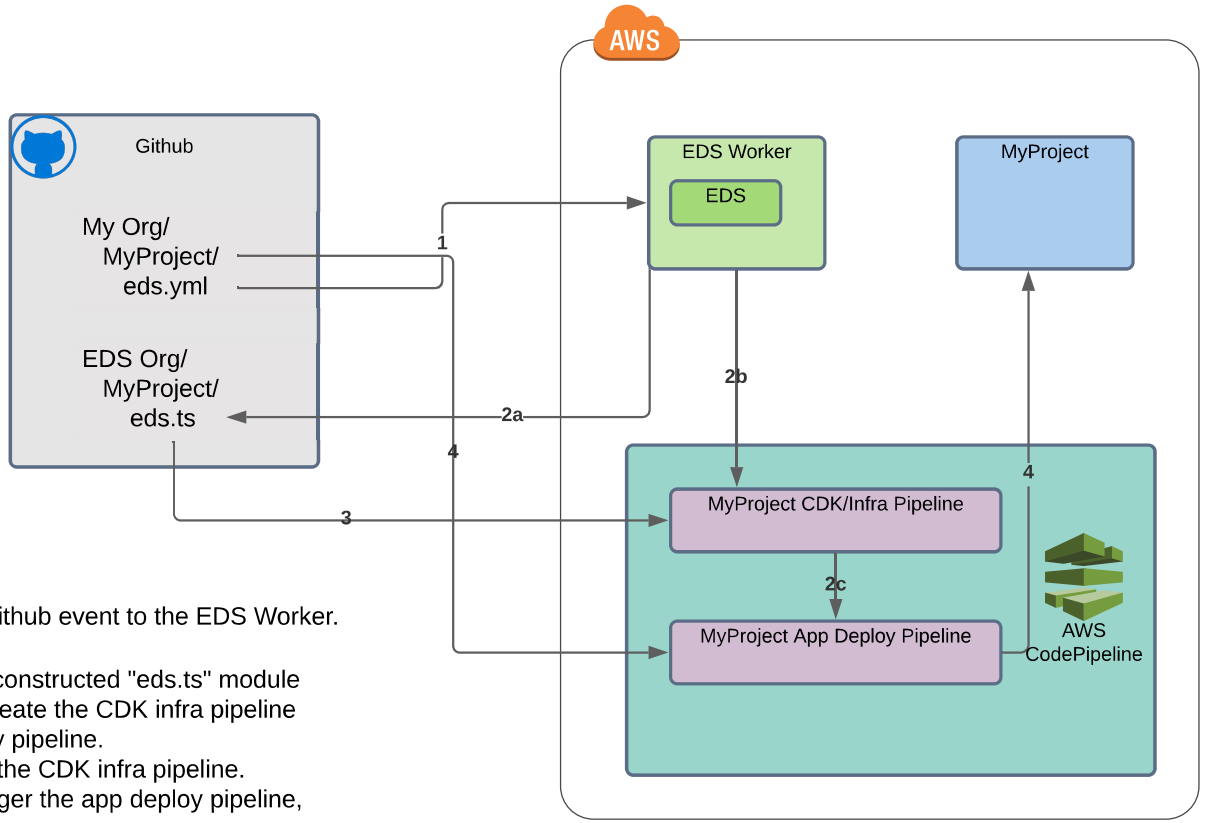
1. Commits to "MyProject" sends a Github event to the EDS Worker.
2. EDS Worker uses the EDS Library to:
   a. Create "MyProject" in the EDS org to hold the pipeline configuration.
   b. Create a corresponding "MyProject" pipeline.
   c. Write a Jenkinsfile that will deploy "MyProject" as specified by `eds.yml`. The Jenkinsfile is configured to deploy the version of "MyProject" that triggered the Event.
3. The commit to "MyProject" triggers the corresponding pipeline.
4. The pipeline uses the EDS Library to build, test, and deploy "MyProject".

AWS Worker



1. Webhook Event
2. Parse EDS Version
3. Build EDS
4. Launch rebuilt EDS
5. Parse plugin versions
6. Build EDS with plugins
7. Launch rebuilt EDS
8. Do the work....

EDS + CDK



1. Commits to "MyOrg/MyProject" sends a Github event to the EDS Worker.
2. EDS Worker uses the EDS Library to:
   a. Create "EDSOrg/MyProject" to hold the constructed "eds.ts" module
   b. Do the initial "cdk deploy" bootstrap to create the CDK infra pipeline
   c. The CDK pipeline creates the app deploy pipeline.
3. Future updates to "eds.ts" by EDS trigger the CDK infra pipeline.
4. Future updates to ""MyOrg/MyProject" trigger the app deploy pipeline, which deploys the app.