infra

Overview
Functionalities
How to use

Overview

What this piece of code does is to handle some CI/CD tasks specific to the management of the NKIA project source code base.
Using this piece of code, CI/CD tasks involving the build, stage, and test pipeline can be simplified as tweaking some values in configuration file and run the executable

Functionalities

1. Staging to downstream repositories

infractl (the binary that is the product of this piece of code) can handle staging process of part or the whole of this repository based on ci.yaml file specifically designed for relieving the hassle of managing the mono repo convergence from or divergence into multiple downstream repositories

2. Automatically runs tests in a preconfigured way

infractl can handle the test process before building the project based on test.yaml file specifically desinged for relieving the hassle of running the each and every test case by hand and check out the result.

How to use

There is no precompiled binary for infractl as in the case of nokubectl, nokubeadm, nokubelet because this is not aimed to be used in such a way.

```
git clone https://github.com/OKESTRO-AIDevOps/nkia.git
cd nkia
make release
# or
# `make build`
# if you just want to compile a binary without
# configuring essential environment
make stage
```

The above "make stage" command invokes the following command with those arguments for staging process

```
sudo ./infractl  --repo
https://github.com/OKESTRO-AIDevOps/nkia.git \
                    --id seantywork \
                    --token - \
                    --name nkia \
                    --plan ci \
```

Here, --repo option specifies the mono repo git address that holds the subset that you want to stage downstream, and --id option is for the git id, and --token option is for specifying the password of the repository. In the case of --token option, you can set it to "-" as above example so that you can feed the token from stdin --name option specifies the local directory name for the cloned repository, and --plan option specifies the exact task you want to perform, "ci" for staging, "test" for testing

The basic structure for instructing how multiple downstream repositories should be built from this mono repo is like below

```
target.v1:
- gitPackage:
    address:
'https://github.com/OKESTRO-AIDevOps/nkia-nokubeadm.git'
    name: nkia-nokubeadm
    strategy: reset
    lock: []
  root: "nkia"
  select:
  - what:
      - "*"
    from: "nokubeadm"
    not: []
    as: "*" # * means selected files will be located directly
inside the package
```