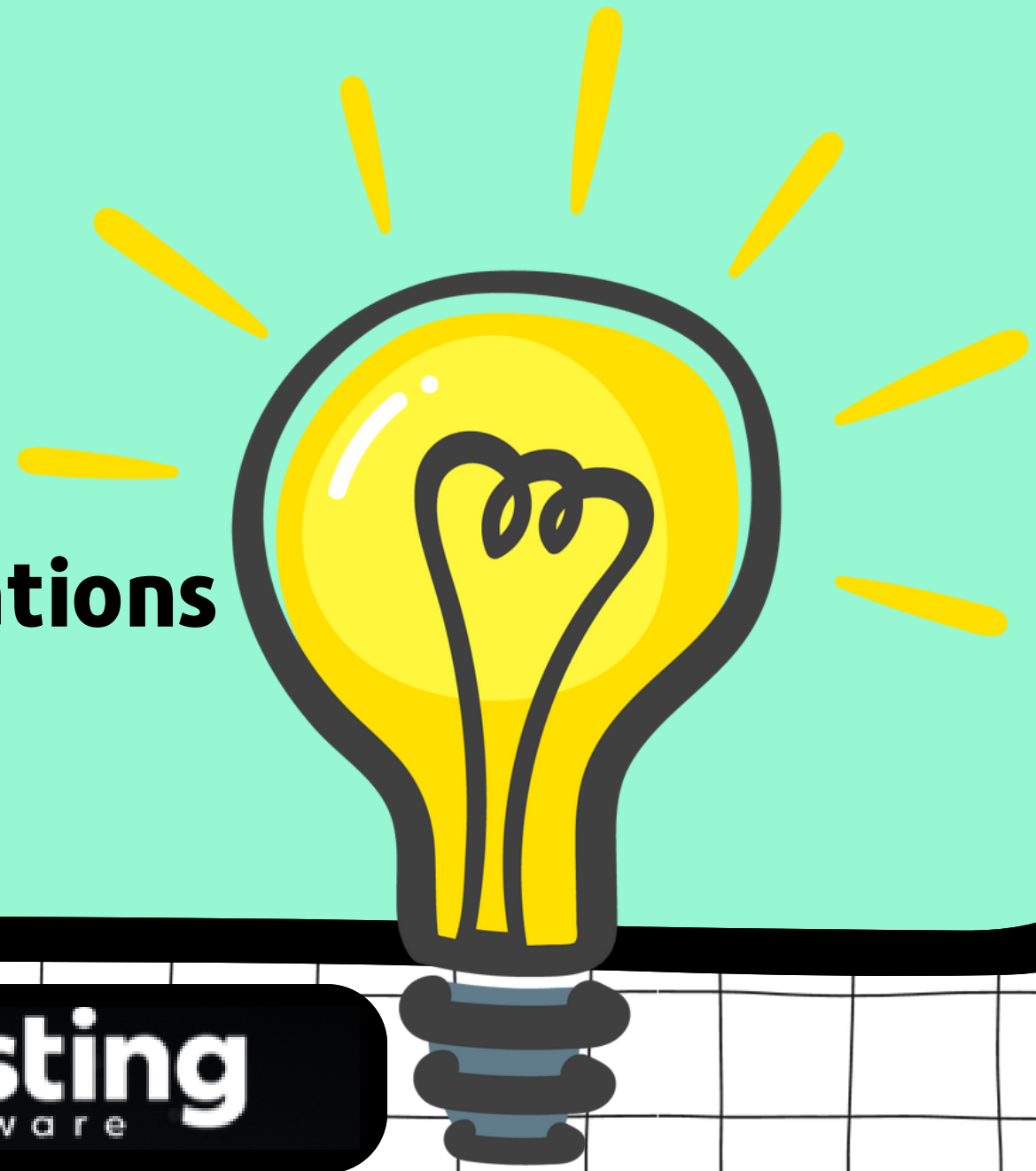


SDLC

**Integrating GitOps and GitFlow
Techniques on CNCF Native Applications**



Losting
software

The team

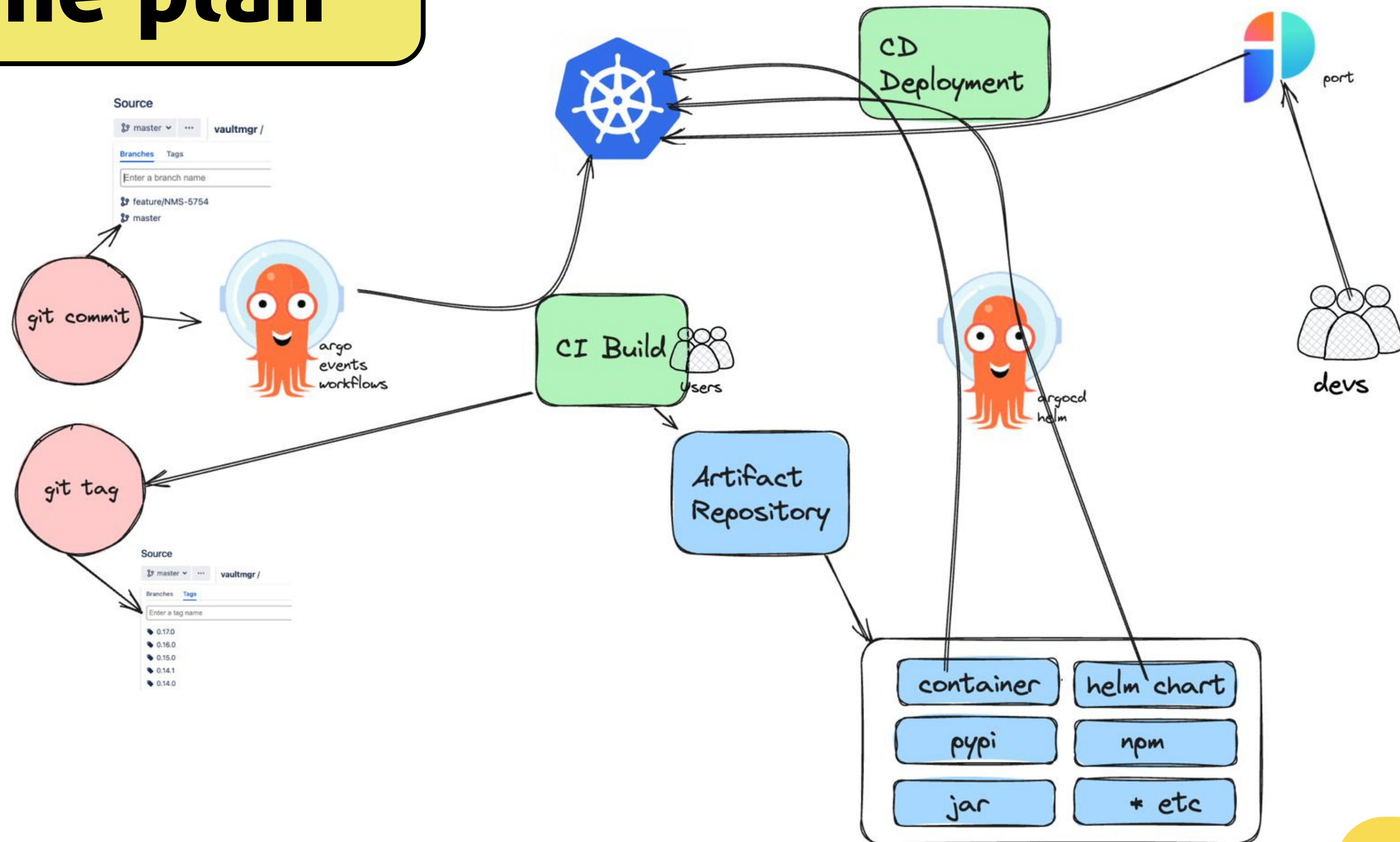


Ionel PETRUT



Marius CHISA

The plan



Content

01

Source / Versioning

02

Branching strategy

03

CI / Build

04

Artifact
Management

05

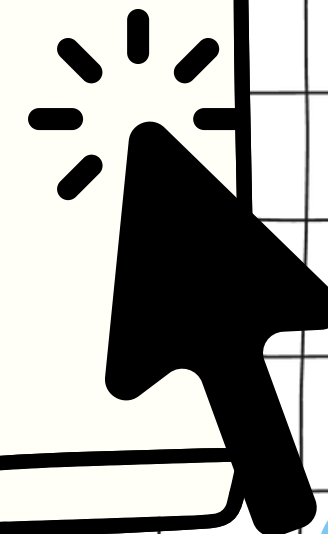
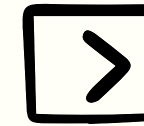
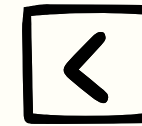
CD / Deployment

06

Security

07

Testing / Coverage
Reporting





Source / Versioning

- SemVer(semantic) became a middle ground for versioning: PEP440 and npm/helm compatible
- ex: 1.0.0 (master), 0.0.0-<digitTicket> for feature/bugfix branches
- Use GIT! Bitbucket / Github (Enterprise) / Gitlab / Gitea, Gitolite, gogs
 - Leverage Gitflow: feature/ bugfix/ release/ master branches
 - lock version(dependencies) only when releasing master versions
 - Use short-lived branches (target one sprint!)
 - avoid epic branches -> headaches to rebase/merge
 - use stable (run test on it) and unstable(not tested) artifact locations
 - develop branch should exists until you have a release peace (every quearter/epic)

Branching strategy

- include ticket number in the branch
- include feature/ bugfix/ release/ master (or main) -> GitFlow
- short lived branches
- use PR process
- trigger tests on PR
- increase PATCH on bugfix, MINOR on feature, leave MAJOR for software architect to increase
- easiest: use git tags


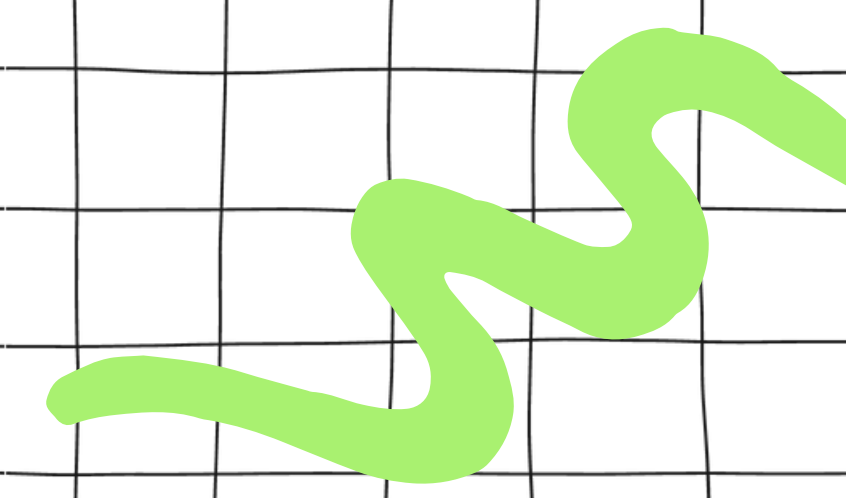


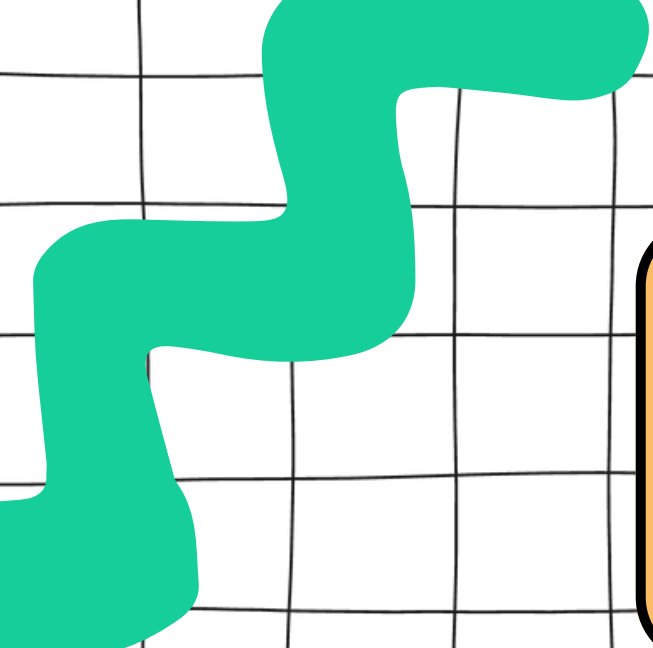
- try to use CNCF tools(build in K8s): Tekton / Argo (events/workflows) / kaniko / buildah / s2i if you app is CNCF native
- use old-school Jenkins if you only have this knowledge within team
- use DRY
- build artifacts, tag/version, push to artifact location
- build where you run: do not build with docker/compose run on containerd/K8s

CI / Build

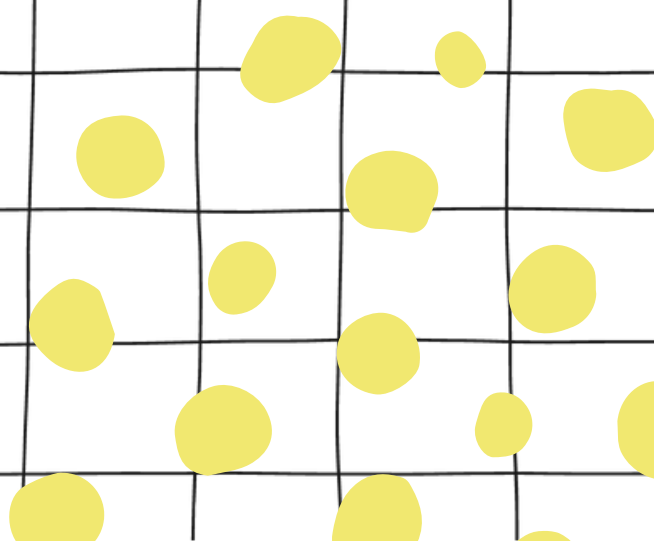


Artifact management

- 
- master version goes to unstable, run test if successful push to stable
 - stable build to reach QA team for manual/system tests
 - Artifactory / Nexus / Harbor are good choices
 - artifacts are docker images/helm charts/python plugins/npm packages etc
- 



CD / Deployment

- 
- use helm, if you are multiple environment kustomize might be a better fit
 - ArgoCD graduated as CNCF deployment tool
 - use external-secrets.io for secret management (You thought of security didn't you?)
 - use Port (getport.io) to provide IDP Service Catalog type of functionality

Security

- container: start with ubi/bitnami etc provided images, less "forking" is better
- non-root/read-only fs, drop all privileges by default > helm conftest to leverage in CI pipeline
- scan your artifacts against CVE: Anchor/Clair are good choices
- SonarQube for code quality, create gates -> software architect

Testing

- our choices(not necessary the best ones) are Pytest/Cypress
- ALM tool of choice is SpiraTeam for BA/PO engagement
- testing infrastructure consists of "simulators" and actual tests(pytest/cypress) in different microservices
- we use GraphQL for entrypoint in our testing infrastructure, kafka/postgres/elasticsearch/etc endpoint to compare
- report in SpiraTeam with #id, PO/BA would keep test coverage/feature/release tree up2date
- use a centralized debugger microservice for test advanced troubleshooting (ares?)

#consolelog

01

argo logs ci-p7bcm -f

```
ci-p7bcm-parameter-setup-4131408244: |-----|
ci-p7bcm-parameter-setup-4131408244: |          MASTER Build          |
ci-p7bcm-parameter-setup-4131408244: |-----|
ci-p7bcm-parameter-setup-4131408244: | PARAMETERS:          VALUES |
ci-p7bcm-parameter-setup-4131408244: | Tag:                 0.13.0  |
ci-p7bcm-parameter-setup-4131408244: | Comment:             test    |
ci-p7bcm-parameter-setup-4131408244: | DeployDimensions:    false   |
ci-p7bcm-parameter-setup-4131408244: | RunFullTests        false   |
ci-p7bcm-parameter-setup-4131408244: | RunFrameworkTests:   false   |
ci-p7bcm-parameter-setup-4131408244: | RunNgpTests:         false   |
ci-p7bcm-parameter-setup-4131408244: | RunCypressTests:     false   |
ci-p7bcm-parameter-setup-4131408244: |-----|
```

02

```
ci-p7bcm-kaniko-build-3703766448: INFO[0035] Taking snapshot of full
filesystem...
ci-p7bcm-kaniko-build-3703766448: INFO[0036] Pushed
013914783300.dkr.ecr.us-east-
1.amazonaws.com/cache@sha256:aff9b49176e6e12c7a6e9107bf80f7362f4908aa0bb22
f92e070c09ec99abead
ci-p7bcm-kaniko-build-3703766448: INFO[0036] Pushing layer
013914783300.dkr.ecr.us-east-
1.amazonaws.com/cache:31314ec2e1638845e3800aa917d3d1fc030a964406827c0c3a42
7ac651a5cf53 to cache now
ci-p7bcm-kaniko-build-3703766448: INFO[0036] ENTRYPOINT ["/opt/app-
root/entrypoint.sh"]
ci-p7bcm-kaniko-build-3703766448: INFO[0036] Pushing image to
013914783300.dkr.ecr.us-east-
1.amazonaws.com/cache:31314ec2e1638845e3800aa917d3d1fc030a964406827c0c3a42
7ac651a5cf53
ci-p7bcm-kaniko-build-3703766448: INFO[0036] No files ch...
```

03

```
ci-p7bcm-helm-build-127278775: See https://github.com/instrumenta/helm-conftest for help
getting started.
ci-p7bcm-helm-build-127278775: Installed plugin: conftest
ci-p7bcm-helm-build-127278775: +-----+-----+-----+
ci-p7bcm-helm-build-127278775: | RESULT | FILE | MESSAGE |
ci-p7bcm-helm-build-127278775: +-----+-----+-----+
ci-p7bcm-helm-build-127278775: | success | -    | data.main.warn |
ci-p7bcm-helm-build-127278775: | success | -    | data.main.warn |
ci-p7bcm-helm-build-127278775: | success | -    | data.main.deny |
ci-p7bcm-helm-build-127278775: | success | -    | data.main.deny |
ci-p7bcm-helm-build-127278775: | success | -    |
```

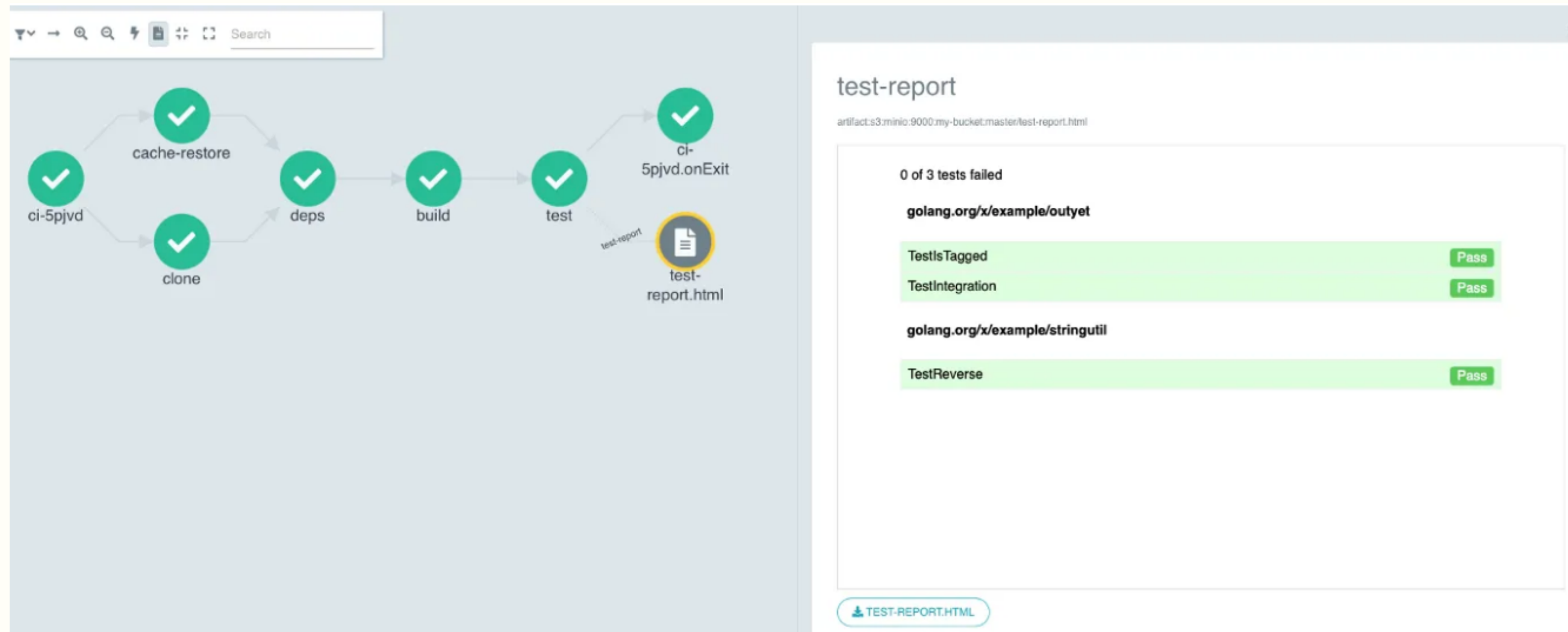
04

```
ci-p7bcm-tagging-and-locking-3927130954: + git push
https://jenkins:idiDirect@vault.idirect.net/scm/~mchisa/alarmmgrtest.git --tags
ci-p7bcm-tagging-and-locking-3927130954: To https://vault.idirect.net/scm/~mchisa/alarmmgrtest.git
ci-p7bcm-tagging-and-locking-3927130954: * [new tag]          0.13.0 -> 0.13.0
ci-p7bcm-tagging-and-locking-3927130954: time="2023-11-07T10:05:17.348Z" level=info msg="sub-process
exited" argo=true error="<nil>"
```

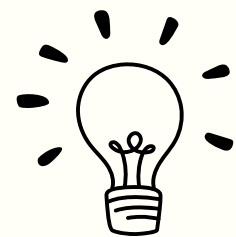
In action...

#gui

In action...



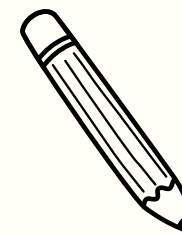
References



code: <https://github.com/djmario-ro/argo-ci/tree/main>

ArgoCD: <https://argo-cd.readthedocs.io/en/stable/>

SemVer: <https://semver.org/>



kaniko:
<https://github.com/GoogleContainerTools/kaniko>

argo events/workflows:
<https://argoproj.github.io/>

Crossplane: <https://www.crossplane.io/>



helm conftest:
<https://github.com/instrumenta/helm-conftest>

Clair: <https://github.com/quay/clair>

Port: <https://www.getport.io/>



Thank you!

