Going Beyond CI/CD with **Prow**

@leodido

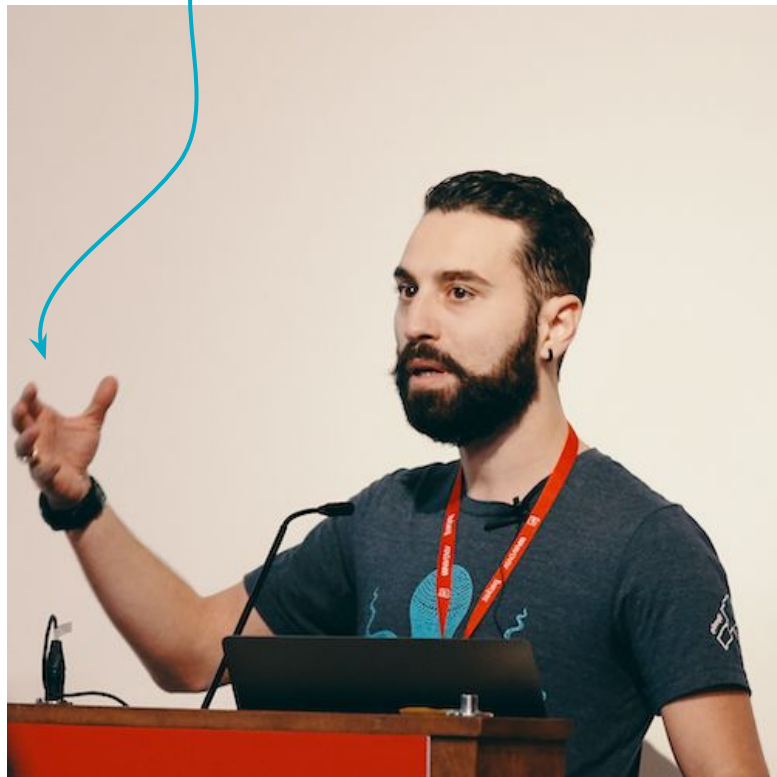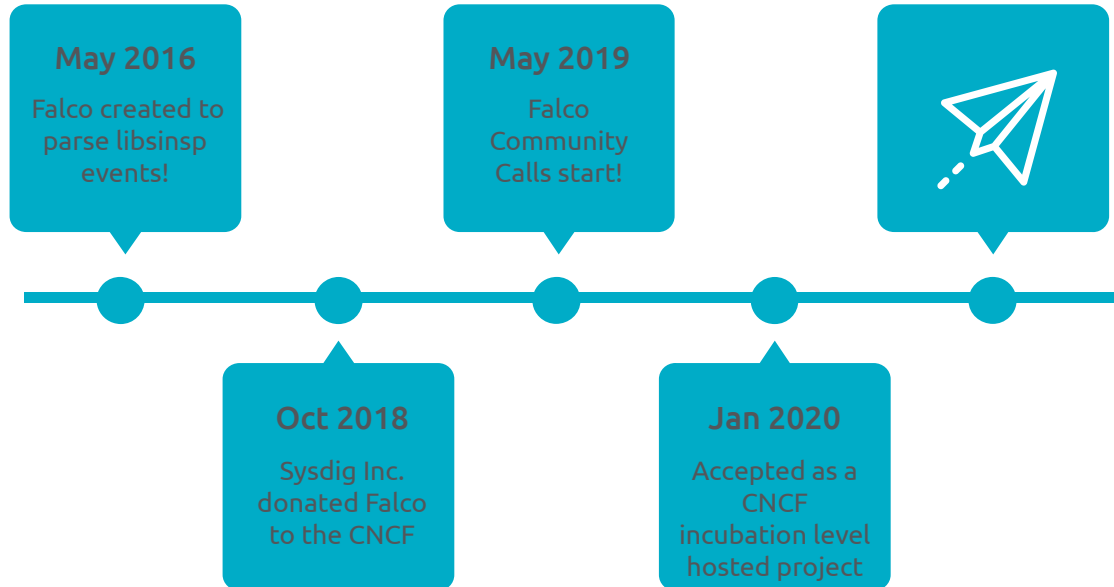**KubeCon** | **CloudNativeCon**

Europe 2020

Whoami!

Leonardo Di Donato

Open Source Software Engineer
Falco Maintainer

@leodido 🐦 ⊙

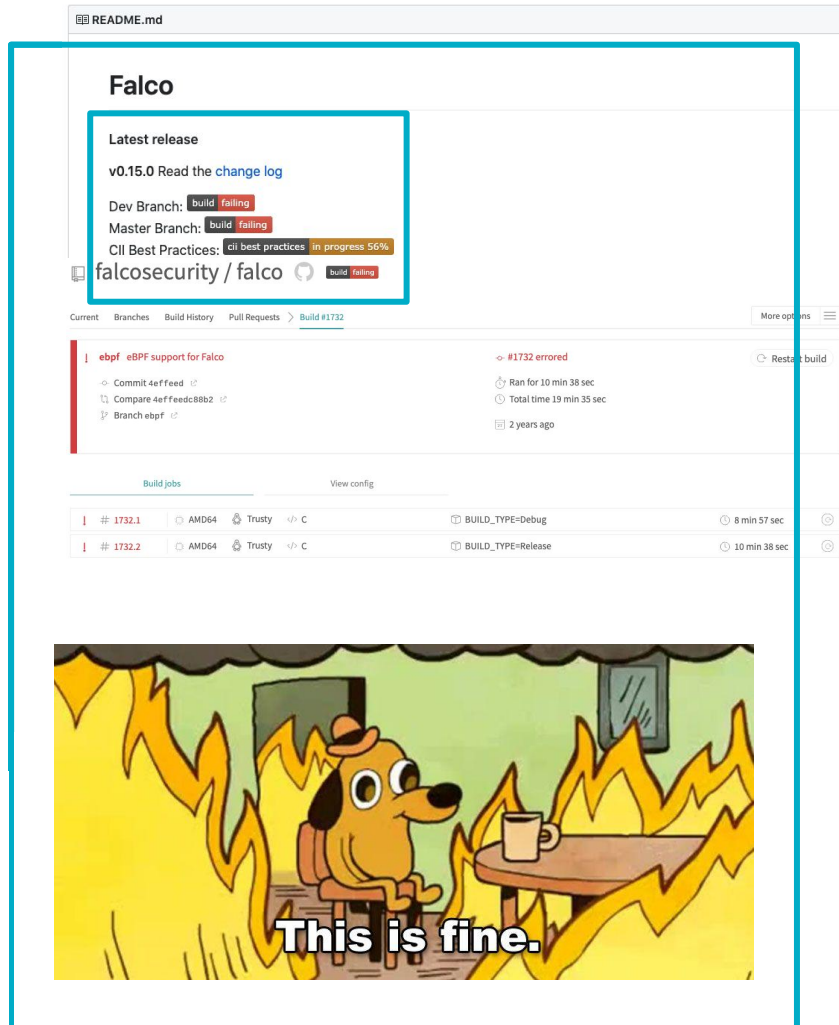extra points to who spots the meaning of this Italian hand-gesture! 😄

sysdig   ✕ Falco   CLOUD NATIVE COMPUTING FOUNDATION

2

A timeline always works fine

May 2016

Falco created to parse libsinsp events!

May 2019

Falco Community Calls start!

Oct 2018

Sysdig Inc. donated Falco to the CNCF

Jan 2020

Accepted as a CNCF incubation level hosted project

Drama!

❏   CI was built on top of Travis CI
    ❏   Barely working
    ❏   Flaky
❏   Falco drivers building & testing infrastructure not public
    ❏   Internal Jenkins

More drama!

- ❏ PRs merged even when GitHub status checks not passing
- ❏ No clear ownership
- ❏ Missing PRs reviews
- ❏ Some policies there but…
  - ❏ Not easily discoverable & auditable
  - ❏ Zero enforcement
  - ❏ Zero automation
- ❏ The list goes on…

No blame. Just solutions.™

We had some constraints…
- ❏ No time to build a custom CI/CD system from scratch
- ❏ No time to create an automatic policy enforcer
- ❏ Team of 2

While the main focus was to:
- ❏ Write eBPF
- ❏ Write C/C++ code

How Kubernetes does?

Way more than CI/CD.

- ❏   **GitHub ChatOps**
- ❏   **Define and enforce policies**
- ❏   **Auto-merge bot**
- ❏   **OSS** ❣️

Prow

Built for Kubernetes.
On Kubernetes.

CRDs

- ❏ Custom controllers
- ❏ Resource management

Pods

- ❏ Scheduling
- ❏ Lifecycle
- ❏ A bunch of containers

ConfigMaps

- ❏ Auditable
- ❏ Configurations

Secrets

- ❏ Credentials management
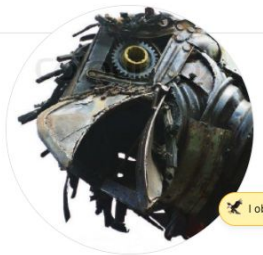
Poiana stops the drama!

💻🤓

More time spent in
Falco coding.

Healthier OSS

Auditable policies.
Clear rules.
Templated processes.
Auto enforcement.

Community

More contributors.
More projects.

Prow Architecture

- [ ] **ProwJob** custom resource
- [ ] **Horologium**
- [ ] **Hook**
- [ ] **Plank**
- [ ] **Deck**
- [ ] **Tide**
- [ ] **Sinker**
- [ ] **Crier**

@leodido

```go
// ProwJobStatus provides runtime metadata, such as when it finished, whether it is
// running, etc.
type ProwJobStatus struct {
    // StartTime is equal to the creation time of the ProwJob
    StartTime metav1.Time `json:"startTime,omitempty"`
    // PendingTime is the timestamp for when the job moved from triggered to pending
    PendingTime *metav1.Time `json:"pendingTime,omitempty"`
    // CompletionTime is the timestamp for when the job goes to a final state
    CompletionTime *metav1.Time `json:"completionTime,omitempty"`
    State          ProwJobState `json:"state,omitempty"`
    Description    string       `json:"description,omitempty"`
    URL            string       `json:"url,omitempty"`

    // PodName applies only to ProwJobs fulfilled by
    // plank. This field should always be the same as
    // the ProwJob.ObjectMeta.Name field.
    PodName string `json:"pod_name,omitempty"`

    // BuildID is the build identifier vended either by tot
    // or the snowflake library for this job and used as an
    // identifier for grouping artifacts in GCS for views in
    // TestGrid and Gubernator. Idenitifiers vended by tot
    // are monotonically increasing whereas identifiers vended
    // by the snowflake library are not.
    BuildID string `json:"build_id,omitempty"`

    // JenkinsBuildID applies only to ProwJobs fulfilled
    // by the jenkins-operator. This field is the build
    // identifier that Jenkins gave to the build for this
    // ProwJob.
    JenkinsBuildID string `json:"jenkins_build_id,omitempty"`

    // PrevReportStates stores the previous reported prowjob state per reporter
    // So crier won't make duplicated report attempt
    PrevReportStates map[string]ProwJobState `json:"prev_report_states,omitempty"`
}
```

```go
// ProwJobSpec configures the details of the prow job.
//
// Details include the podspec, code to clone, the cluster it runs
// any child jobs, concurrency limitations, etc.
type ProwJobSpec struct {
    // Type is the type of job and informs how
    // the jobs is triggered
    Type ProwJobType `json:"type,omitempty"`
    // Agent determines which controller fulfills
    // this specific ProwJobSpec and runs the job
    Agent ProwJobAgent `json:"agent,omitempty"`
    // Cluster is which Kubernetes cluster is used
    // to run the job, only applicable for that
    // specific agent
    Cluster string `json:"cluster,omitempty"`
    // Namespace defines where to create pods/resources.
    Namespace string `json:"namespace,omitempty"`
    // Job is the name of the job
    Job string `json:"job,omitempty"`
    // Refs is the code under test, determined at
    // runtime by Prow itself
    Refs *Refs `json:"refs,omitempty"`
    // ExtraRefs are auxiliary repositories that
    // need to be cloned, determined from config
    ExtraRefs []Refs `json:"extra_refs,omitempty"`
    // Report determines if the result of this job should
    // be reported (e.g. status on GitHub, message in Slack, etc.)
    Report bool `json:"report,omitempty"`
    // Context is the name of the status context used to
    // report back to GitHub
    Context string `json:"context,omitempty"`
    // RerunCommand is the command a user would write to
    // trigger this job on their pull request
    RerunCommand string `json:"rerun_command,omitempty"`
    // MaxConcurrency restricts the total number of instances
    // of this job that can run in parallel at once
    MaxConcurrency int `json:"max_concurrency,omitempty"`
    // ErrorOnEviction indicates that the ProwJob should be completed and given
    // the ErrorState status if the pod that is executing the job is evicted.
    // If this field is unspecified or false, a new pod will be created to replace
    // the evicted one.
    ErrorOnEviction bool `json:"error_on_eviction,omitempty"`

    // PodSpec provides the basis for running the test under
    // a Kubernetes agent
    PodSpec *corev1.PodSpec `json:"pod_spec,omitempty"`

    // JenkinsSpec holds configuration specific to Jenkins jobs
    JenkinsSpec *JenkinsSpec `json:"jenkins_spec,omitempty"`

    // PipelineRunSpec provides the basis for running the test as
    // a pipeline-crd resource
    // https://github.com/tektoncd/pipeline
    PipelineRunSpec *pipelinev1alpha1.PipelineRunSpec
    `json:"pipeline_run_spec,omitempty"`

    // DecorationConfig holds configuration options for
    // decorating PodSpecs that users provide
    DecorationConfig *DecorationConfig `json:"decoration_config,omitempty"`

    // ReporterConfig holds reporter-specific configuration
    ReporterConfig *ReporterConfig `json:"reporter_config,omitempty"`

    // RerunAuthConfig holds information about which users can rerun the job
    RerunAuthConfig *RerunAuthConfig `json:"rerun_auth_config,omitempty"`

    // Hidden specifies if the Job is considered hidden.
    // Hidden jobs are only shown by deck instances that have the
    // `--hiddenOnly=true` or `--show-hidden=true` flag set.
    // Presubmits and Postsubmits can also be set to hidden by
    // adding their repository in Decks `hidden_repo` setting.
    Hidden bool `json:"hidden,omitempty"`
}
```
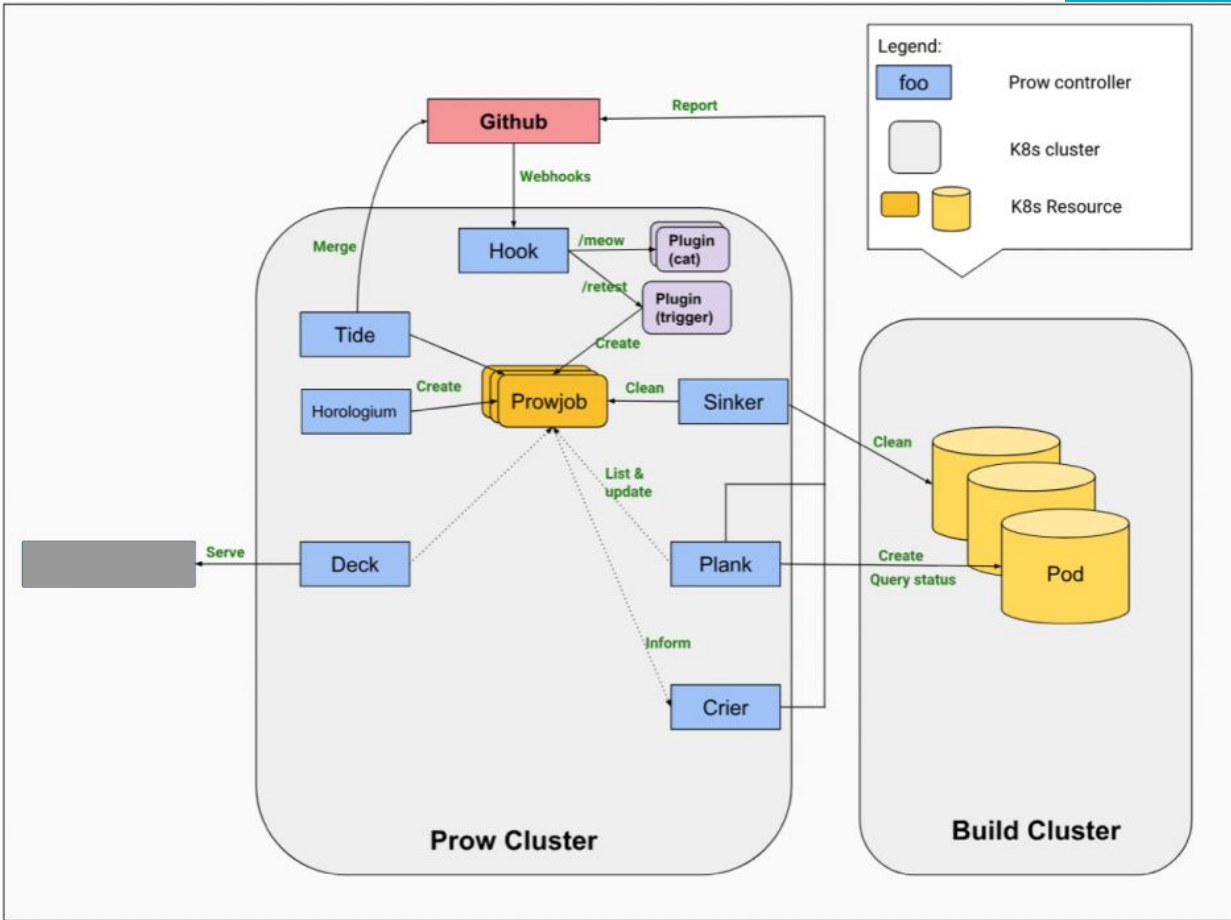
ProwJobs

- ☐ **Life of a ProwJob**
- ☐ **Custom resource**
- ☐ **Custom controllers**
- ☐ **Types**
  - ☐ presubmit
  - ☐ postsubmit
  - ☐ batch
  - ☐ periodic

```go
// +k8s:deepcopy-gen:interfaces=k8s.io/apimachinery/pkg/runtime.Object

// ProwJob contains the spec as well as runtime metadata.
type ProwJob struct {
    metav1.TypeMeta   `json:",inline"`
    metav1.ObjectMeta `json:"metadata,omitempty"`

    Spec   ProwJobSpec   `json:"spec,omitempty"`
    Status ProwJobStatus `json:"status,omitempty"`
}
```
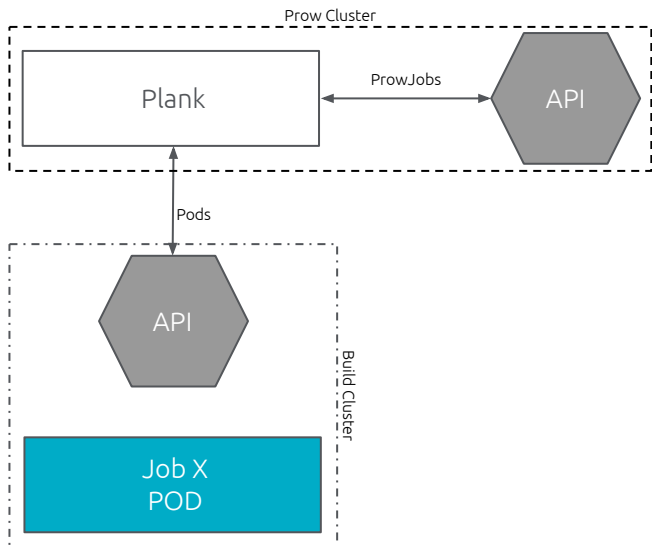
@leodido

Plank

Controller that manages the lifecycle of jobs.

Horologium

Periodic jobs... 👉gh:falcosecurity/test-infra/pull/78

Payload

```
{
  "id": 10303926682,
  "sha": "571f8a28e7d338ecac5b12a492df97812e208045",
  "name": "falcosecurity/falco",
  "target_url": "https://circleci.com/gh/falcosecurity/falco/3469?utm_campaign=vcs-integration-link&utm_medium=ref
  "avatar_url": "https://avatars2.githubusercontent.com/oa/4808?v=4",
  "context": "ci/circleci: centos7",
  "description": "Your tests passed on CircleCI!",
  "state": "success",
  "commit": {
    "sha": "571f8a28e7d338ecac5b12a492df97812e208045",
    "node_id": "MDY6Q29tbWl0NDk5ODDYwNDY6NTcxZjhhMjhlN2QzMzhlY2FjNWIxMmE0OTJkZjk3ODEyZTIwODA0NQ==",
    "commit": {
      "author": {
        "name": "kaizhe",
        "email": "derek0405@gmail.com",
        "date": "2020-07-23T23:22:48Z"
      },
      "committer": {
        "name": "poiana",
        "email": "51138685+poiana@users.noreply.github.com",
        "date": "2020-07-25T06:53:06Z"
      },
      "message": "add macro user_read_sensitive_file_containers\n\nSigned-off-by: kaizhe <derek0405@gmail.com>",
      "tree": {
        "sha": "b19b618b1bfd5d0248ae3323ba3c8e938c2a2ad1",
        "url": "https://api.github.com/repos/falcosecurity/falco/git/trees/b19b618b1bfd5d0248ae3323ba3c8e938c2a2ad
      },
      "url": "https://api.github.com/repos/falcosecurity/falco/git/commits/571f8a28e7d338ecac5b12a492df97812e20804
```
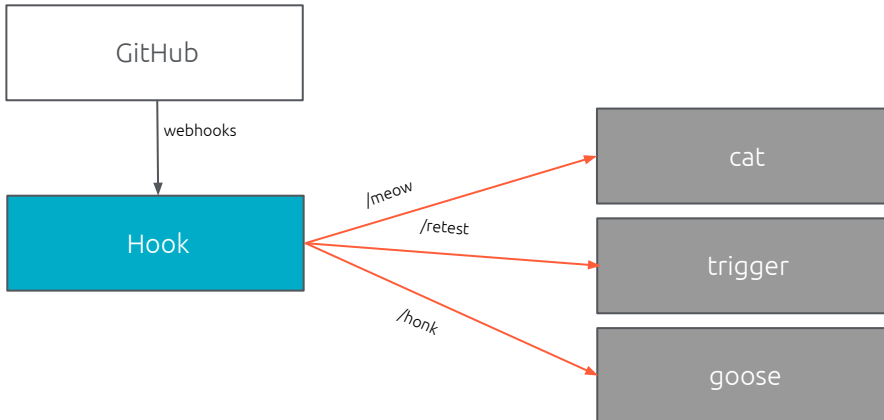
Hook

👉 kubernetes/test-infra/prow/hook
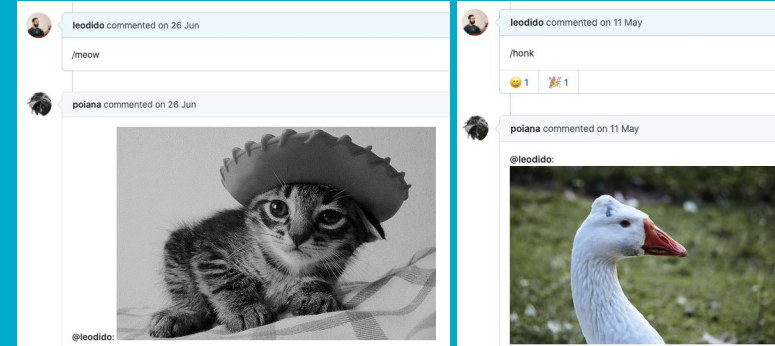
@leodido

Plugins for everybody

👉kubernetes/test-infra/prow/plugins

- ❏ config_updater
- ❏ trigger
- ❏ approve, lgtm
- ❏ hold, wip
- ❏ ...

leodido commented on 24 Jun   Author

/hold

to be merged in after #139 (which needs required checks to be changed manually in order to be merged i

🏷 poiana added size/XS   do-not-merge/hold labels on 24 Jun

poiana commented 16 days ago   Member ☺ •••

@leogr: Updated the `configs` configmap in namespace `default` at cluster `default` using the following files:

- key `configs.yaml` using file `prow/config/configs.yaml`

▶ Details

👁 kris-nova reviewed 15 days ago

kris-nova left a comment

LGTM - Can we also add a small section to falco.org
community calls.

👍 1

poiana assigned kris-nova 15 days ago

poiana added the lgtm label 15 days ago

poiana commented 15 days ago

LGTM label has been added.

▶ Details

```go
// We need to empty import all enabled plugins so that they will be linked into any
hook binary.
import (
	_ "k8s.io/test-infra/prow/plugins/approve" // Import all enabled plugins.
	_ "k8s.io/test-infra/prow/plugins/assign"
	_ "k8s.io/test-infra/prow/plugins/blockade"
	_ "k8s.io/test-infra/prow/plugins/blunderbuss"
	_ "k8s.io/test-infra/prow/plugins/branchcleaner"
	_ "k8s.io/test-infra/prow/plugins/bugzilla"
	_ "k8s.io/test-infra/prow/plugins/buildifier"
	_ "k8s.io/test-infra/prow/plugins/cat"
	_ "k8s.io/test-infra/prow/plugins/cherrypickunapproved"
	_ "k8s.io/test-infra/prow/plugins/cla"
	_ "k8s.io/test-infra/prow/plugins/dco"
	_ "k8s.io/test-infra/prow/plugins/dog"
	_ "k8s.io/test-infra/prow/plugins/golint"
	_ "k8s.io/test-infra/prow/plugins/goose"
	_ "k8s.io/test-infra/prow/plugins/heart"
	_ "k8s.io/test-infra/prow/plugins/help"
	_ "k8s.io/test-infra/prow/plugins/hold"
	_ "k8s.io/test-infra/prow/plugins/invalidcommitmsg"
	_ "k8s.io/test-infra/prow/plugins/label"
	_ "k8s.io/test-infra/prow/plugins/lgtm"
	_ "k8s.io/test-infra/prow/plugins/lifecycle"
	_ "k8s.io/test-infra/prow/plugins/merge-method-comment"
	_ "k8s.io/test-infra/prow/plugins/mergecommitblocker"
	_ "k8s.io/test-infra/prow/plugins/milestone"
	_ "k8s.io/test-infra/prow/plugins/milestoneapplier"
	_ "k8s.io/test-infra/prow/plugins/milestonestatus"
	_ "k8s.io/test-infra/prow/plugins/override"
	_ "k8s.io/test-infra/prow/plugins/owners-label"
	_ "k8s.io/test-infra/prow/plugins/pony"
	_ "k8s.io/test-infra/prow/plugins/project"
	_ "k8s.io/test-infra/prow/plugins/projectmanager"
	_ "k8s.io/test-infra/prow/plugins/releasenote"
	_ "k8s.io/test-infra/prow/plugins/require-matching-label"
	_ "k8s.io/test-infra/prow/plugins/requiresig"
	_ "k8s.io/test-infra/prow/plugins/retitle"
	_ "k8s.io/test-infra/prow/plugins/shrug"
	_ "k8s.io/test-infra/prow/plugins/sigmention"
	_ "k8s.io/test-infra/prow/plugins/size"
	_ "k8s.io/test-infra/prow/plugins/skip"
	_ "k8s.io/test-infra/prow/plugins/slackevents"
	_ "k8s.io/test-infra/prow/plugins/stage"
	_ "k8s.io/test-infra/prow/plugins/trigger"
	_ "k8s.io/test-infra/prow/plugins/updateconfig"
	_ "k8s.io/test-infra/prow/plugins/verify-owners"
	_ "k8s.io/test-infra/prow/plugins/welcome"
	_ "k8s.io/test-infra/prow/plugins/wip"
	_ "k8s.io/test-infra/prow/plugins/yuks"
)
```
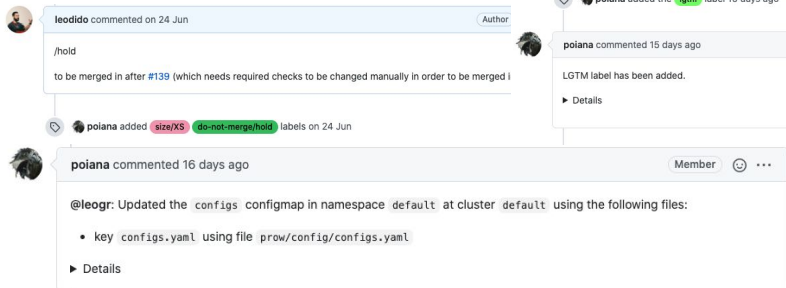
Deck

@leodido

```yaml
tide:
  context_options:
    skip-unknown-contexts: true
    from-branch-protection: true
  merge_method:
    falcosecurity/falco: rebase
  queries:
    - repos:
        - falcosecurity/falco
      labels:
        - approved
        - lgtm
        - "dco-signoff: yes"
      missingLabels:
        - do-not-merge
        - do-not-merge/hold
        - do-not-merge/invalid-owners-file
        - do-not-merge/work-in-progress
        - do-not-merge/release-note-label-needed
        - needs-rebase

branch-protection:
  enforce_admins: true # rules apply to admins too!
  restrictions: # restrict who can push
    teams: ["maintainers", "machine_users"]
  required_pull_request_reviews:
    dismiss_stale_reviews: true # automat dismiss old reviews
    dismissal_restrictions: # allow review dismissals
      teams:
        - maintainers
        - machine_users
    require_code_owner_reviews: true # require a code owner approval
    required_approving_review_count: 1 # number of approvals
  required_status_checks:
    strict: false # we don't want to block any PR if it's not up to date, we have
the rebase merge strategy and needs-rebase plugin
  orgs:
    falcosecurity:
      # dco check is required for all PRs of any falcosecurity project
      required_status_checks:
        contexts:
          - dco
      falco:
        required_pull_request_reviews:
          required_approving_review_count: 2
        required_status_checks:
          contexts:
            - ... # other required GitHub status check, even not ProwJob ones!
```

Poiana requires these labels to put a Falco PR in merge queue!

These prevent a PR on Falco to be merged in!

Overrides!

Tide

× dco — Commits in PR missing Signed-off-by

● tide *Pending — Not mergeable. Needs approved, dco-signoff: yes, lgtm labels.*

leodido commented 13 days ago

/hold cancel

poiana removed the do-not-merge/hold label 13 days ago

poiana merged commit **c4b7f17** into master 13 days ago
8 checks passed

poiana deleted the docs/0.24.0-changelog branch 13 days ago

Crier

1. Detects ProwJob status changes
2. Reports them
   - ❑ Gerrit
   - ❑ Pubsub
   - ❑ GitHub
   - ❑ Slack

📣 Disable report feature of Plank! (--skip-report=true)
More at 👉 test-infra/prow/cmd/crier

Sinker

🗑️🧹 garbage collector

- ❑ > 48 hours completed jobs (customisable)
- ❑ > 30 min completed pods (customisable)

Workflow
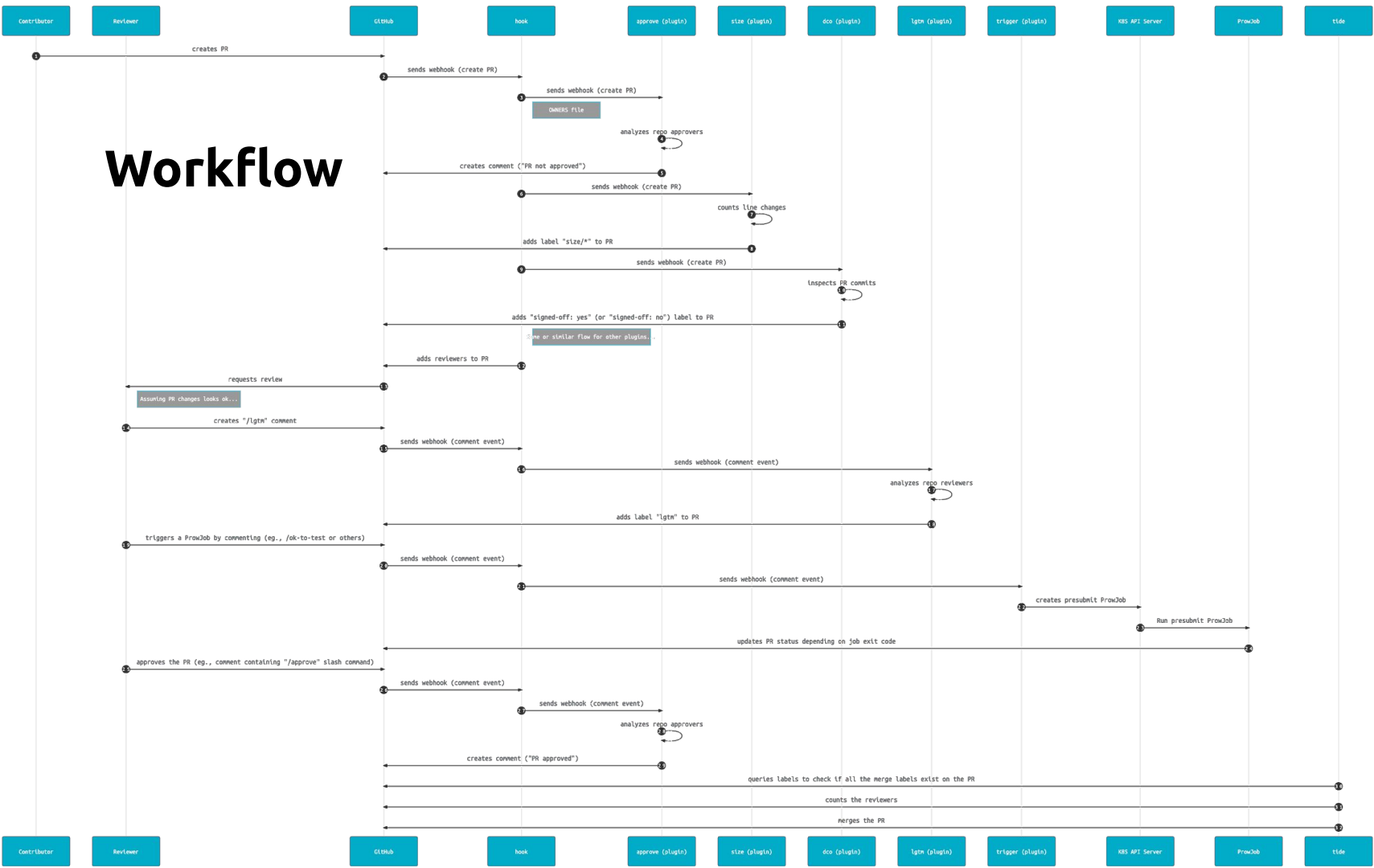
👉 getting started guide
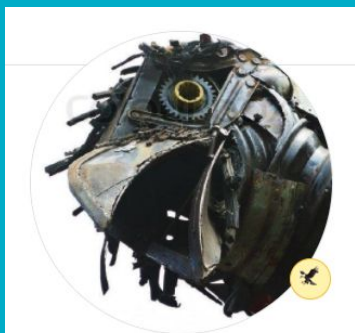
poiana
poiana

I fly around **@falcosecurity** projects.
For issues talk with **@kris-nova** ,
@leodido and **@fntlnz**

I want it too!

1

GitHub Bot Account

❏ Write access to the repos
❏ Owner access for the org(s)
❏ Create access token with repo:status + public_repo scopes
  ❏ admin:org_hook if handling GitHub org

2

Become a YAML developer

1. Create the YAML manifests, cluster role bindings…
2. Create the secrets from tokens (hmac + oauth2)
3. Apply them on your Kubernetes cluster

3

Install the webhook

```
go get -u k8s.io/test-infra/experiment/update-hook

update-hook --hmac-path=path/to/hmac/secret \
   --github-token-path=path/to/oauth/secret \
   --hook-url http://1.2.3.4/hook --repo my_organization --confirm=true
```

falcosecurity/test-infra/prow a.k.a @poiana

```
master ▾    test-infra / prow / cluster /
```

2 authors and **poiana** update(prow): compatibility with kubernetes 1.16 ...

..

📄	0-test.yaml	fix: loving yaml indentation
📄	branchprotector.yaml	update(prow): compatibility with kubernetes 1.16
📄	configs.yaml	new: specify status checks for branch protection
📄	deck.yaml	update(prow): compatibility with kubernetes 1.16
📄	hook.yaml	update(prow): compatibility with kubernetes 1.16
📄	horologium.yaml	update(prow): compatibility with kubernetes 1.16
📄	job.yaml	new: prow custom resource definition
📄	needs-rebase.yaml	update(prow): compatibility with kubernetes 1.16
📄	plank.yaml	update(prow): compatibility with kubernetes 1.16
📄	plugins.yaml	update(prow): compatibility with kubernetes 1.16
📄	sinker.yaml	update(prow): compatibility with kubernetes 1.16
📄	tide.yaml	update(prow): compatibility with kubernetes 1.16

```makefile
SHELL := /bin/bash

cluster/oauth-token.yaml: oauth
    kubectl create secret generic --dry-run -o yaml oauth-token --from-file=$< >> $@

cluster/hmac-token.yaml: hmac
    kubectl create secret generic --dry-run -o yaml hmac-token --from-file=$< >> $@

cluster/unsplash-token.yaml: unsplash_secret
    kubectl create secret generic --dry-run -o yaml unsplash-token --from-file=$< >> $@

prow: cluster/oauth-token.yaml cluster/hmac-token.yaml cluster/unsplash-token.yaml cluster/*.yaml
    kubectl apply -f cluster

plugins: config/plugins.yaml
    kubectl create configmap plugins --from-file=plugins.yaml=config/plugins.yaml --dry-run -o yaml | kubectl replace configmap plugins -f -

configs: config/configs.yaml
    kubectl create configmap configs --from-file=configs.yaml=config/configs.yaml --dry-run -o yaml | kubectl replace configmap configs -f -

clean:
    rm -f cluster/oauth-token.yaml
    rm -f cluster/hmac-token.yaml
    rm -f cluster/unsplash-token.yaml
```

Bring Your Own Prow Plugin

STEPS

1. Write a Go server!
2. Wrap it into a docker image
3. Kubernetes service and deployment
4. kubectl apply -f them!
5. Add it to your Prow cluster

```
external_plugins:
  <org>/<repo>:
  - name: <name>
    endpoint:
http://<name>.default.svc.cluster.local
    events:
    - issue_comment
```

RESOURCES

- ❏ custom external plugin intro
- ❏ phony test utility
- ❏ help helpers
  - ❏ k8s.io/test-infra/prow/pluginhelp
  - ❏ k8s.io/test-infra/prow/pluginhelp/externalplugins
- ❏ GitHub client and utilities
  - ❏ k8s.io/test-infra/prow/github
- ❏ config agent and secrets
  - ❏ k8s.io/test-infra/prow/config/secret
  - ❏ k8s.io/test-infra/prow/config
- ❏ flag helpers
  - ❏ k8s.io/test-infra/pkg/flagutil
  - ❏ k8s.io/test-infra/prow/flagutil

👉 leodido/saymyname-prow-plugin 👈

Read The Source Code

leodido commented now   Member   Author   ☺   ···

/poiana

---

poiana commented now   Member   ☺   ···

@leodido: Poiana stops the drama!

▶ Details

---

leodido commented 30 seconds ago   Member   ☺   ···

/poiana

---

poiana commented 28 seconds ago   Member   ☺   ···

@leodido: I'm Poiana, I stop the drama!

▶ Details

---

leodido commented 17 seconds ago   Member   ☺   ···

/poiana

---

poiana commented 15 seconds ago   Member   ☺   ···

@leodido: I'm Poiana, I'm here to look at Kris's commit messages...

▶ Details

Speculations about the future!

- ❑ Enable the [Prow GitHub API proxy](#)
- ❑ Falco Drivers Test Grid through ProwJobs
- ❑ Code Coverage
- ❑ Code Linting
- ❑ Custom plugin for externals CI
- ❑ Spyglass
- ❑ Plugin that automatically creates a release PR when a milestone is done
- ❑ Periodic job to cleanup old unused Falco version and prebuilt Falco drivers
- ❑ ...

Thanks!

Questions and feedback welcome

- ❏ twitter.com/leodido
- ❏ github.com/leodido
- ❏ github.com/falcosecurity/falco
- ❏ slack.k8s.io, #falco channel
- ❏ github.com/falcosecurity/test-infra
- ❏ github.com/kubernetes/test-infra
- ❏ thanks to Apulia 💙 for inspiration

@leodido