



# ci-operator

(not an operator)

Bruno Barcarol Guimarães

## History

- ▶ `ci-tools`

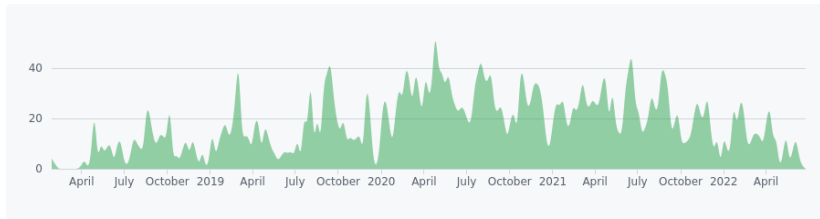
## Motivation

- ▶ Past
- ▶ Prow

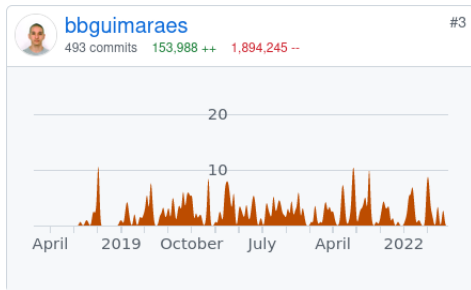
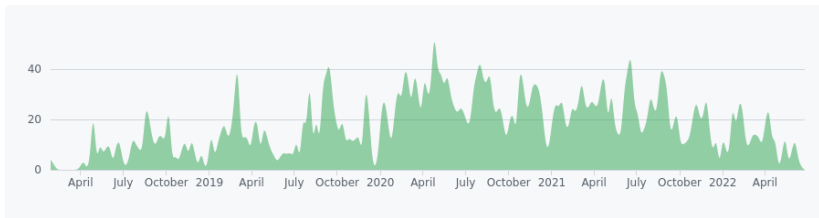
## Architecture

- ▶ Overview
- ▶ Initialization
- ▶ Example
- ▶ Etc.

<https://github.com/openshift/ci-tools/graphs/contributors>



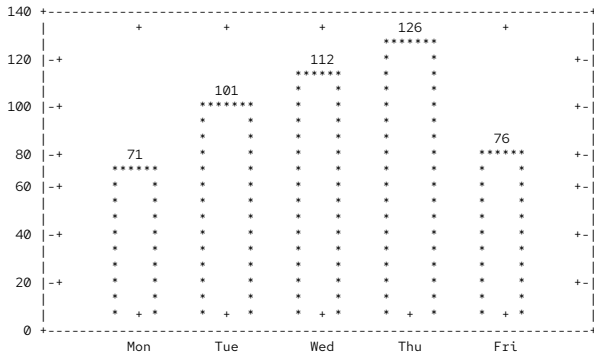
```
$ git log --format='%ad %an %s' --date=format-local:%Y-%m-%d | tail -3
2018-01-31 Steve Kuznetsov Initial implementation of the ci-operator code
2018-01-31 Steve Kuznetsov Initialize vendor directory with Glide
2018-01-31 Steve Kuznetsov Initialize .gitignore
```



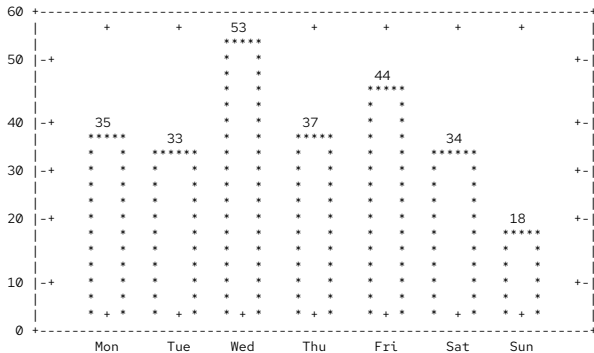
```
$ git log --format='%ad %an %s' --date=format-local:%Y-%m-%d --graph 78af6eb7c
* 2019-06-13 Bruno Barcarol Guimarães Merge Makefiles
*   2019-06-13 Bruno Barcarol Guimarães Merge ci-operator-prowgen
|\
...
| * 2018-08-23 Steve Kuznetsov Add an OWNERS file
| * 2018-08-23 Steve Kuznetsov Reorganize code, add Makefile
| * 2018-08-23 Steve Kuznetsov Copy content from openshift/release
*   2019-06-13 Bruno Barcarol Guimarães Merge ci-operator
|\
...
| * 2018-04-06 Clayton Coleman Add .gitignore for binary
| * 2018-02-09 Steve Kuznetsov Refactor StepLink to be functional
| * 2018-02-08 Steve Kuznetsov Add the release tagging step
| * 2018-02-02 Steve Kuznetsov Add a dry-run mode to the entrypoint
| * 2018-01-31 Steve Kuznetsov Initial implementation of the ci-operator code
| * 2018-01-31 Steve Kuznetsov Initialize vendor directory with Glide
| * 2018-01-31 Steve Kuznetsov Initialize .gitignore
* 2019-06-13 Bruno Barcarol Guimarães Initial commit
```

- ▶ `https://github.com/openshift/ci-tools.git`
  - ▶ `https://github.com/openshift/ci-operator.git`
  - ▶ `https://github.com/openshift/ci-operator-prowgen.git`
  - ▶ ...

```
$ git log --format=%ad --date=format-local: '%a %u' --author 'Bruno Barcarol Guimarães' \
  | sort | uniq -c | sort -nk 3,3 \
  | gnuplot ...
```



```
$ git log --format=%ad --date=format-local: '%a %u' --author 'Clayton Coleman' \
  | sort | uniq -c | sort -nk 3,3 \
  | gnuplot ...
```







*Dogfooding Openshift with our CI infrastructure,  
Michalis Kargakis (2018-01-27)*

<https://www.youtube.com/watch?v=rLLEjodf1Yw>

## ▶ Jenkins

- ▶ Jenkins (R.I.P.)
  - ▶ Ruby (OpenShift v2), Python, Bash, ...
  - ▶ *slow*
  - ▶ unmaintained
  - ▶ *unmaintainable*
- ▶ Kubernetes
  - ▶ <https://github.com/kubernetes/test-infra.git>
  - ▶ Prow
- ▶ OpenShift
  - ▶ ImageStreams
  - ▶ Builds
  - ▶ multi-tenancy
  - ▶ ...

[https://github.com/kubernetes/test-infra/blob/master/prow/life\\_of\\_a\\_prow\\_job.md](https://github.com/kubernetes/test-infra/blob/master/prow/life_of_a_prow_job.md)

apiVersion: prow.k8s.io/v1

kind: ProwJob

metadata:

name: 32456927-35d9-11e7-8d95-0a580a6c1504

spec:

job: pull-test-infra-bazel

decorate: true

pod\_spec:

containers:

- image: gcr.io/k8s-staging-test-infra/bazelbuild:latest-test-infra

refs:

base\_ref: master

base\_sha: 064678510782db5b382df478bb374aaa32e577ea

org: kubernetes

pulls:

- author: ixdy

number: 2716

sha: dc32ccc9ea3672ccc523b7cbaa8b00360b4183cd

repo: test-infra

type: presubmit

<https://github.com/openshift/release/blob/master/ci-operator/jobs/openshift/ci-tools/openshift-ci-tools-master-presubmits.yaml>

```
presubmits:
  openshift/ci-tools:
    - branches:
      - ^master$
      - ^master-
    cluster: build04
    labels:
      ci.openshift.io/generator: prowgen
      pj-rehearse.openshift.io/can-be-rehearsed: "true"
    name: pull-ci-openshift-ci-tools-master-unit
    spec:
      containers:
        - args:
            - --gcs-upload-secret=/secrets/gcs/service-account.json
            - --image-import-pull-secret=/etc/pull-secret/.dockerconfigjson
            - --report-credentials-file=/etc/report/credentials
            - --target=unit
          command:
            - ci-operator
          image: ci-operator:latest
```

<https://github.com/openshift/release/blob/master/ci-operator/config/openshift/ci-tools/openshift-ci-tools-master.yaml>

```
base_images:
  os:
    name: centos
    namespace: origin
    tag: stream8
binary_build_commands: >
  make production-install
build_root:
  from_repository: true
  use_build_cache: true
images:
- context_dir: >
  images/ci-operator/
  from: os
  inputs:
    bin:
      paths:
      - destination_dir: .
        source_path: >
          /go/bin/ci-operator
  to: ci-operator
promotion:
  namespace: ci
  tag: latest
test_binary_build_commands: >
  make race-install
tests:
- as: unit
  commands: make test
  container:
    from: src
- as: e2e
  steps:
    test:
      - as: e2e
        commands: make e2e
        from: test-bin
```

- ▶ <https://docs.ci.openshift.org>
  - ▶ [/docs/architecture/ci-operator](https://docs.ci.openshift.org/docs/architecture/ci-operator)
  - ▶ [/docs/architecture/ci-operator-internals](https://docs.ci.openshift.org/docs/architecture/ci-operator-internals)
  - ▶ [/docs/architecture/ci-operator-internals/steps](https://docs.ci.openshift.org/docs/architecture/ci-operator-internals/steps)
- ▶ <https://github.com/openshift/ci-docs/pulls>
  - ▶ [#233](https://github.com/openshift/ci-docs/pull/233)
  - ▶ [#235](https://github.com/openshift/ci-docs/pull/235)
  - ▶ [#266](https://github.com/openshift/ci-docs/pull/266)





- ▶ Inputs
  - ▶ repository / git revision(s)
  - ▶ command-line arguments
  - ▶ configuration
- ▶ Outputs
  - ▶ test results
  - ▶ images

▶ Inputs

- ▶ repository / git revision(s)
- ▶ command-line arguments
- ▶ configuration

▶ Outputs

- ▶ test results
- ▶ images

▶ Implementation

- ▶ build cluster/node
- ▶ temporary namespace
- ▶ image pipeline
- ▶ test types
- ▶ cloud providers
- ▶ remote storage
- ▶ image promotion

*ci-operator is at its core a task scheduling program. The input configuration is processed and used to build a task graph, which is then executed until completion, failure, or interruption. Thus, the execution flow of ci-operator can be divided in these major phases:*

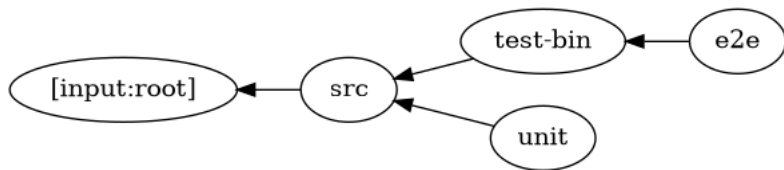
- ▶ *input processing*
- ▶ *task graph creation*
- ▶ *task graph execution*
- ▶ *cleanup*

```
$ ci-operator \  
  --unresolved-config ci-tools-master.yaml \  
  --print-graph \  
  --target unit --target e2e
```

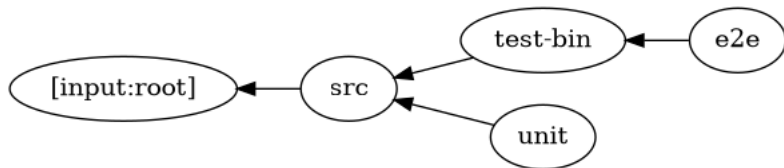
```
...  
src [input:root]  
test-bin src  
unit src  
e2e test-bin  
...
```

<https://pkg.go.dev/golang.org/x/tools/cmd/digraph>

```
$ ci-operator \  
  --unresolved-config ci-tools-master.yaml \  
  --print-graph \  
  --target unit --target e2e \  
  | hack/ci-operator/graphviz.pl -T png \  
  > out.png
```

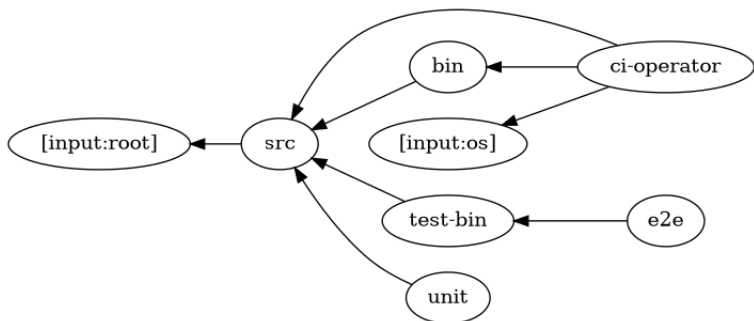


(requirement ← dependent)

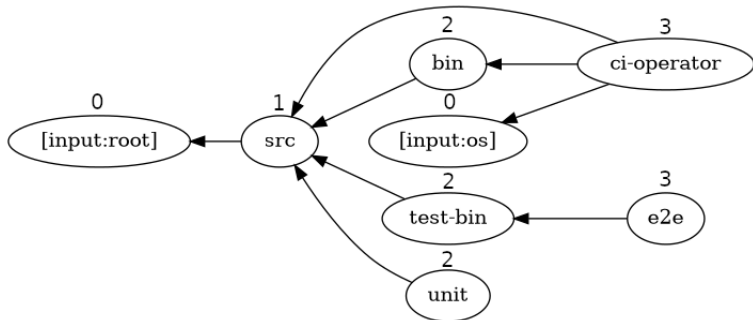


- ▶ import root image
  - ▶ build\_root
  - ▶ special name: [input:...]
- ▶ build src image
  - ▶ explicit requirement of unit, implicit requirement of test-bin
- ▶ build test-bin image
  - ▶ test\_binary\_build\_commands
  - ▶ requested by from: test-bin in e2e
- ▶ execute unit test
  - ▶ --target
- ▶ execute e2e test
  - ▶ --target

... --target unit --target e2e --target ci-operator ...



```
INFO[2022-06-27T10:46:10Z] Running [input:root], [input:os], \  
src, bin, test-bin, ci-operator, unit, e2e  
1 2 2 3 2 3
```





<https://docs.ci.openshift.org/docs/architecture/ci-operator-internals/steps/#step-types>

- ▶ build steps
- ▶ release steps
- ▶ (optional) operator steps
- ▶ auxiliary steps
- ▶ test steps
- ▶ output steps

pkg/api/graph.go (simplified)

```
type Step interface {
    Name() string
    Description() string
    Requires() []StepLink
    Creates() []StepLink
    Inputs() (InputDefinition, error)
    Run(ctx context.Context) error
}

type StepNode struct {
    Step      Step
    Children []*StepNode
}
```

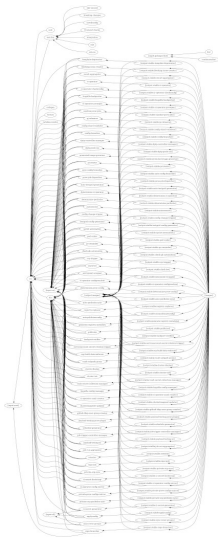
```
// StepGraph is a DAG of steps referenced by
// its roots
type StepGraph []*StepNode

// OrderedStepList is a topologically-ordered
// sequence of steps. Edges are determined
// based on the Creates/Requires methods.
type OrderedStepList []*StepNode
```

```
// TopologicalSort validates nodes form a DAG and orders them
// topologically.
func (g StepGraph) TopologicalSort() (OrderedStepList, []error) {
    ...
    var waiting []*StepNode
    ...
}
```

## pkg/steps/run.go (simplified)

```
func Run(graph api.StepGraph) {
    var seen []api.StepLink
    results := make(chan *api.StepNode)
    for _, root := range graph {
        go runStep(root, results)
    }
    for out := range results {
        seen = append(seen, out.Step.Creates()...)
        for _, child := range out.node.Children {
            if api.HasAllLinks(child.Step.Requires(), seen) {
                go runStep(child, results)
            }
        }
    }
}
```



## cmd/ci-operator/main.go

```
steps, postSteps, err := defaults.FromConfig(
    ctx, o.configSpec, &o.graphConfig, o.jobSpec,
    o.templates, o.writeParams, o.promote, o.clusterConfig,
    leaseClient, o.targets.values, o.cloneAuthConfig,
    o.pullSecret, o.pushSecret, o.censor, o.hiveKubeconfig,
    o.consoleHost, o.nodeName)
```

pkg/api/graph.go (simplified)

```
type InputDefinition []string
```

```
type Step interface {
```

```
    ...
```

```
    Inputs() (InputDefinition, error)
```

```
    ...
```

```
}
```



## cmd/ci-operator/main.go (simplified)

```
func (o *options) resolveInputs(steps []api.Step) {
    var inputs api.InputDefinition
    for _, step := range steps {
        inputs = append(inputs, step.Inputs()...)
    }
    inputs = append(inputs, string(o.configSpec))
    inputs = append(inputs, o.extraInputHash.values...)
    stat := os.Stat(exec.LookPath(os.Args[0]))
    inputs = append(inputs, fmt.Sprintf(
        "%d-%d", stat.ModTime().UTC().Unix(), stat.Size()))
    sort.Strings(inputs)
    o.inputHash = inputHash(inputs)
    if len(o.namespace) == 0 {
        o.namespace = "ci-op-{"id}"
    }
    o.namespace = strings.Replace(
        o.namespace, "{"id}", o.inputHash, -1)
}
```

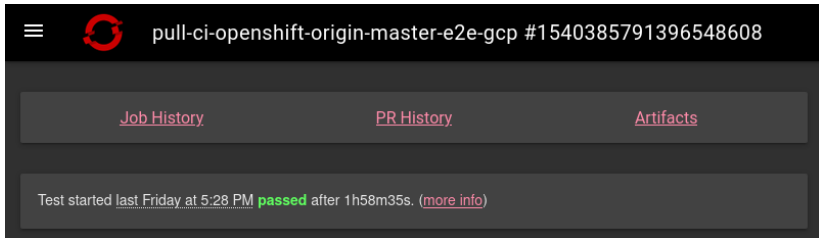
cmd/ci-operator/main.go (simplified)

```
var encoding = base32
    .NewEncoding("bcdfghijklmnpqrstvwxyz0123456789")
    .WithPadding(base32.NoPadding)

func inputHash(inputs api.InputDefinition) string {
    hash := sha256.New()
    for _, s := range inputs {
        hash.Write([]byte(s))
    }
    return encoding.EncodeToString(hash.Sum(nil)[:5])
}
```

cmd/ci-operator/main.go (very simplified)

```
steps, postSteps := defaults.FromConfig(...)
o.resolveInputs(steps)
nodes := api.BuildPartialGraph(steps, o.targets.values)
stepList := nodes.TopologicalSort()
logrus.Infof(
    "Running %s",
    strings.Join(nodeNames(stepList), ", "))
o.initializeNamespace()
steps.Run(ctx, nodes)
for _, step := range postSteps {
    runStep(ctx, step)
}
```



The screenshot shows a Jenkins job page with a dark theme. At the top left is a hamburger menu icon. Next to it is the Jenkins logo (a red circular arrow) and the job name 'pull-ci-openshift-origin-master-e2e-gcp #1540385791396548608'. Below the job name is a horizontal bar with three links: 'Job History', 'PR History', and 'Artifacts'. Underneath this bar is a summary box containing the text: 'Test started last Friday at 5:28 PM **passed** after 1h58m35s. ([more info](#))'.

`https://prow.ci.openshift.org/view/gs/origin-ci-test/pr-logs/  
pull/27275/pull-ci-openshift-origin-master-e2e-gcp/  
1540385791396548608`

```
https://github.com/openshift/release/blob/master/ci-operator/  
config/openshift/origin/openshift-origin-master.yaml
```

```
tests:
```

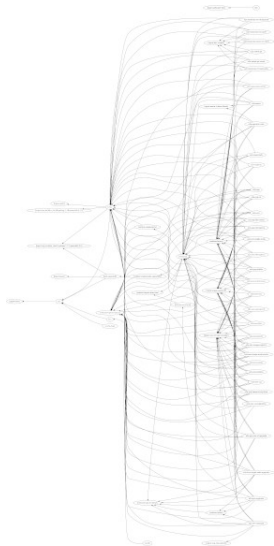
```
- as: e2e-gcp
```

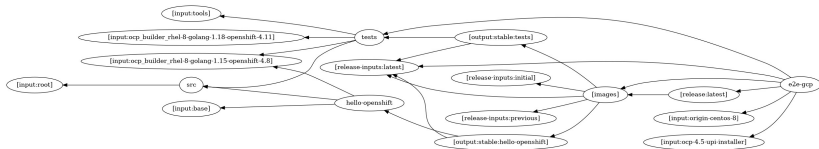
```
  steps:
```

```
    cluster_profile: gcp-openshift-gce-devel-ci-2
```

```
    workflow: openshift-e2e-gcp-loki
```

```
INFO[2022-06-24T17:28:45Z] ci-operator version v20220621-3b245f722
INFO[2022-06-24T17:28:45Z] Loading configuration from https://config.ci.openshift.org for openshift/...
INFO[2022-06-24T17:28:46Z] Resolved source https://github.com/openshift/origin to master@a946e2b9, m...
INFO[2022-06-24T17:28:46Z] Building release previous from a snapshot of ocp/4.10
INFO[2022-06-24T17:28:46Z] Building release initial from a snapshot of ocp/4.11
INFO[2022-06-24T17:28:46Z] Building release latest from a snapshot of ocp/4.11
INFO[2022-06-24T17:28:47Z] Using namespace https://console.build02.ci.openshift.org/k8s/cluster/proj...
INFO[2022-06-24T17:28:47Z] Running [input:root], [input:ocp_builder_rhel-8-golang-1.15-openshift-4.8...
INFO[2022-06-24T17:28:47Z] Tagging ocp/builder:rhel-8-golang-1.15-openshift-4.8 into pipeline:ocp_bu...
INFO[2022-06-24T17:28:47Z] Tagging ocp/builder:rhel-8-golang-1.18-openshift-4.11 into pipeline:ocp_b...
...
INFO[2022-06-24T17:28:48Z] Building src
INFO[2022-06-24T17:32:03Z] Build src succeeded after 3m57s
INFO[2022-06-24T17:32:03Z] Building hello-openshift
INFO[2022-06-24T17:32:03Z] Building tests
INFO[2022-06-24T17:35:23Z] Build hello-openshift succeeded after 3m20s
INFO[2022-06-24T17:35:23Z] Tagging hello-openshift into stable
INFO[2022-06-24T17:42:13Z] Build tests succeeded after 6m48s
INFO[2022-06-24T17:42:13Z] Tagging tests into stable
INFO[2022-06-24T17:42:14Z] Creating release image registry.build02.ci.openshift.org/ci-op-8yq06grj/r...
INFO[2022-06-24T17:43:34Z] Snapshot integration stream into release 4.11.0-0.ci.test-2022-06-24-1742...
INFO[2022-06-24T17:43:34Z] Acquiring leases for test e2e-gcp: [gcp-openshift-gce-devel-ci-2-quota-sl...
INFO[2022-06-24T17:43:34Z] Acquired 1 lease(s) for gcp-openshift-gce-devel-ci-2-quota-slice: [us-cen...
INFO[2022-06-24T17:43:34Z] Running multi-stage test e2e-gcp
INFO[2022-06-24T17:43:34Z] Running multi-stage phase pre
INFO[2022-06-24T17:43:34Z] Running step e2e-gcp-ipi-install-hosted-loki.
...
INFO[2022-06-24T19:27:16Z] Releasing leases for test e2e-gcp
INFO[2022-06-24T19:27:17Z] Ran for 1h58m30s
INFO[2022-06-24T19:27:17Z] Reporting job state 'succeeded'
```



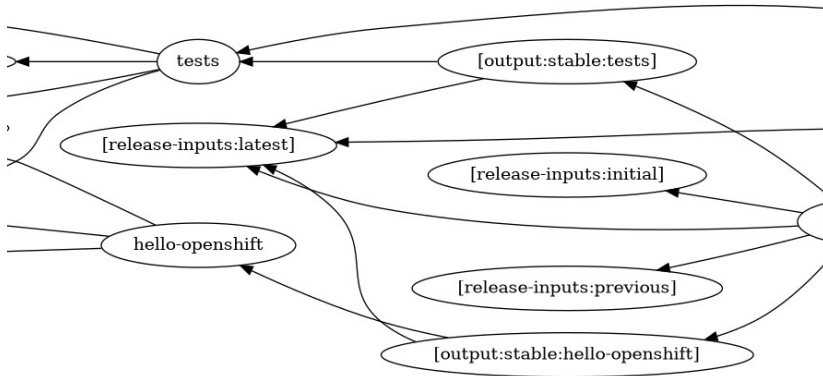




```
ci-operator version v20220621-3b245f722
Loading configuration \
  from https://config.ci.openshift.org \
  for openshift/origin@master
Resolved source https://github.com/openshift/origin \
  to master@a946e2b9, merging: \
  #27275 457391d6 @DennisPeriquet
```

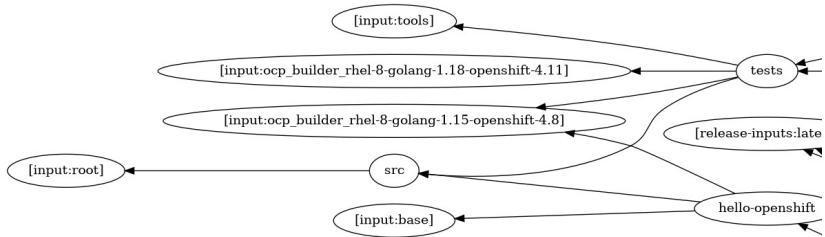
Building release previous from a snapshot of ocp/4.10  
Building release initial from a snapshot of ocp/4.11  
Building release latest from a snapshot of ocp/4.11

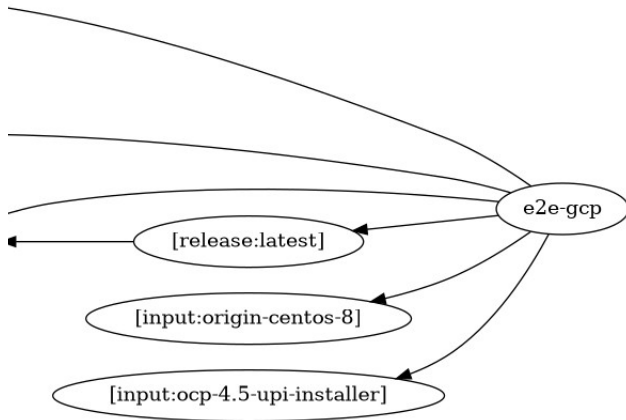
```
releases:
  initial:
    integration:
      name: "4.11"
      namespace: ocp
  latest:
    integration:
      include_built_images: true
      name: "4.11"
      namespace: ocp
  previous:
    integration:
      name: "4.10"
      namespace: ocp
```



```
Using namespace https://console.build02.ci.openshift.org/\
  k8s/cluster/projects/ci-op-8yq06grj
Running [input:root], \
  [input:ocp_builder_rhel-8-golang-1.15-openshift-4.8], \
  [input:ocp_builder_rhel-8-golang-1.18-openshift-4.11], \
  [input:tools], [input:base], [input:origin-centos-8], \
  [input:ocp-4.5-upi-installer], [release-inputs:previous], \
  [release-inputs:initial], [release-inputs:latest], \
  src, tests, hello-openshift, \
  [output:stable:tests], [output:stable:hello-openshift], \
  [images], [release:latest], e2e-gcp
```

```
Tagging ocp/builder:rhel-8-golang-1.15-openshift-4.8 into \  
  pipeline:ocp_builder_rhel-8-golang-1.15-openshift-4.8.  
Tagging ocp/builder:rhel-8-golang-1.18-openshift-4.11 into \  
  pipeline:ocp_builder_rhel-8-golang-1.18-openshift-4.11.  
Tagging openshift/release:rhel-8-release-golang-1.18-openshift-4.11 \  
  into pipeline:root.  
Tagging origin/centos:8 into pipeline:origin-centos-8.  
Tagging ocp/4.11:base into pipeline:base.  
Tagging ocp/4.5:upi-installer into pipeline:ocp-4.5-upi-installer.  
Tagging ocp/4.11:tools into pipeline:tools.
```





```
base_images:  
  base: {...}  
  ocp_builder_rhel-8-golang-1.15-openshift-4.8: {...}  
  ocp_builder_rhel-8-golang-1.18-openshift-4.11: {...}  
  tools: {...}
```



```
build_root:  
  from_repository: true
```

```
https://github.com/openshift/origin/blob/master/  
.ci-operator.yaml
```

```
build_root_image:  
  name: release  
  namespace: openshift  
  tag: rhel-8-release-golang-1.18-openshift-4.11
```

<https://steps.ci.openshift.org/workflow/openshift-e2e-gcp-loki>

workflow:

```
as: openshift-e2e-gcp-loki
steps:
  allow_best_effort_post_steps: true
  pre:
    - ref: ipi-install-hosted-loki
    - chain: ipi-gcp-pre
  test:
    - ref: openshift-e2e-test
  post:
    - chain: ipi-gcp-post
```

documentation: |-

The Openshift E2E GCP workflow executes the common end-to-end test suite on GCP with a default cluster configuration with loki as log collector.

<https://steps.ci.openshift.org/chain/ipi-gcp-pre>

chain:

```
as: ipi-gcp-pre
```

```
steps:
```

```
- chain: ipi-conf-gcp
```

```
- chain: ipi-install
```

```
documentation: |-
```

```
The IPI setup step contains all steps that  
provision an OpenShift cluster with a default  
configuration on GCP.
```

<https://steps.ci.openshift.org/chain/ipi-conf-gcp>

chain:

```
as: ipi-conf-gcp
```

```
steps:
```

- ref: ipi-conf
- ref: ipi-conf-gcp
- ref: ipi-install-monitoringpvc

```
documentation: >-
```

This chain generates an `install-config.yaml` file configured to run clusters in the GCP CI project.

The GCP specific configs are added to the file generated by the `ipi-conf` steps. This resulting file is stored in the shared directory for future consumption.

<https://steps.ci.openshift.org/reference/ipi-conf>

<https://steps.ci.openshift.org/reference/ipi-conf-gcp>

ref:

as: ipi-conf

from\_image:

namespace: origin

name: centos

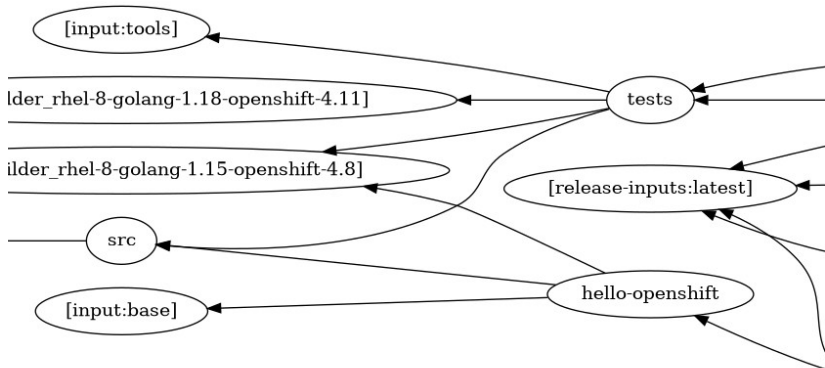
tag: '8'

(from\_image  $\approx$  base\_images + from)

<https://steps.ci.openshift.org/reference/gather-gcp-console>

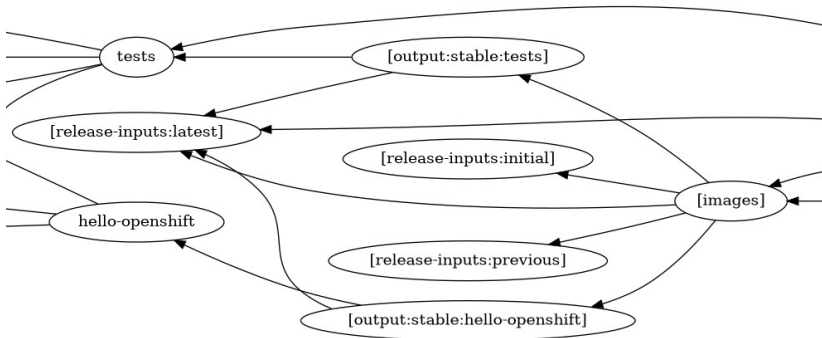
```
ref:  
  as: gather-gcp-console  
  optional_on_success: true  
  from_image:  
    namespace: ocp  
    name: "4.5"  
    tag: upi-installer  
  ...
```

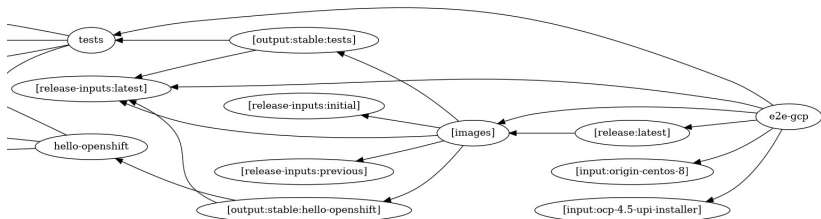
```
Building src
Build src succeeded after 3m57s
Building hello-openshift
Building tests
Build hello-openshift succeeded after 3m20s
Tagging hello-openshift into stable
Build tests succeeded after 6m48s
Tagging tests into stable
```





```
Build hello-openshift succeeded after 3m20s  
Tagging hello-openshift into stable  
Build tests succeeded after 6m48s  
Tagging tests into stable
```



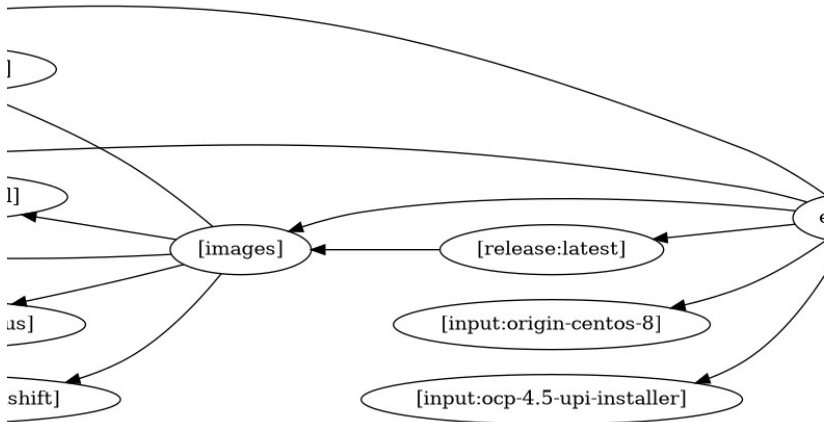


```
https://github.com/openshift/release/blob/master/ci-operator/  
jobs/openshift/origin/openshift-origin-master-postsubmits.yaml  
(simplified)
```

```
name: branch-ci-openshift-origin-master-images  
spec:  
  containers:  
    - args:  
      - --target=[images]  
      - --promote  
    command:  
      - ci-operator
```

```
Creating release image registry.build02.ci.openshift.org/\
  ci-op-8yq06grj/release:latest.
Snapshot integration stream into release \
  4.11.0-0.ci.test-2022-06-24-174214-ci-op-8yq06grj-latest \
  to tag release:latest
```

```
releases:  
  latest:  
    integration:  
      include_built_images: true  
      name: "4.11"  
      namespace: ocp
```



```
Acquiring leases for test e2e-gcp: \  
  [gcp-openshift-gce-devel-ci-2-quota-slice]  
Acquired 1 lease(s) for \  
  gcp-openshift-gce-devel-ci-2-quota-slice: \  
  [us-central1--gcp-openshift-gce-devel-ci-2-quota-slice-08]
```



<https://github.com/openshift/ci-tools/blob/master/pkg/api/types.go> (heavily abbreviated)

```
type ClusterProfile string

const ClusterProfileGCP2 ClusterProfile =
    "gcp-openshift-gce-devel-ci-2"

func (p ClusterProfile) LeaseType() string {
    switch p {
    case ClusterProfileGCP2:
        return "gcp-openshift-gce-devel-ci-2-quota-slice"
    }
}
```

```
Running multi-stage test e2e-gcp
Running multi-stage phase pre
Running step e2e-gcp-ipi-install-hosted-loki.
Step e2e-gcp-ipi-install-hosted-loki succeeded after 20s.
Running step e2e-gcp-ipi-conf.
Step e2e-gcp-ipi-conf succeeded after 20s.
...
Running multi-stage phase test
Running step e2e-gcp-openshift-e2e-test.
Step e2e-gcp-openshift-e2e-test succeeded after 55m0s.
Step phase test succeeded after 55m0s.
Running multi-stage phase post
Running step e2e-gcp-gather-gcp-console.
Step e2e-gcp-gather-gcp-console succeeded after 50s.
...
Step phase post succeeded after 16m40s.
```

```
Releasing leases for test e2e-gcp  
Ran for 1h58m30s  
Reporting job state 'succeeded'
```

- ▶ build clusters
- ▶ `ci-ns-ttl-controller`
- ▶ image distribution
- ▶ types of releases
- ▶ (*init*) containers
- ▶ test types
  - ▶ container
  - ▶ template
  - ▶ multi-stage
- ▶ artifacts
- ▶ image promotion

# Thank you