End-to-End Tests in OpenShift

Bruno Barcarol Guimarães

Introduction

Test types

Releases

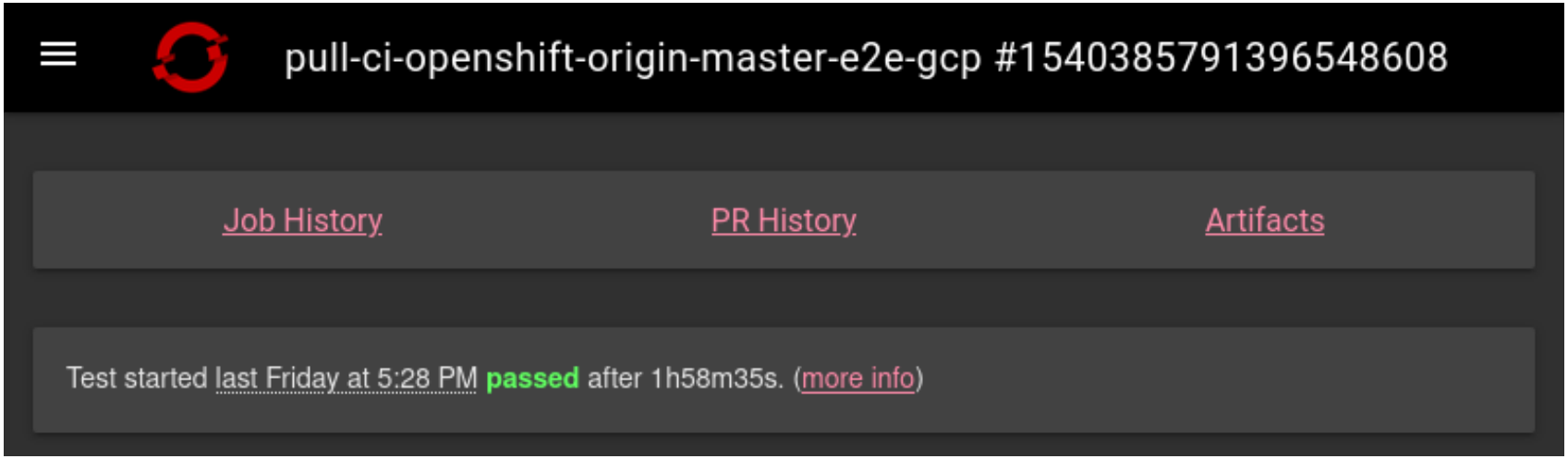
Input images

Image pipeline

Red Hat

Ravioli code

Isolated bits of code that resemble ravioli. These are easy to understand individually but—taken as a group—add to the app's call stack and complexity.



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

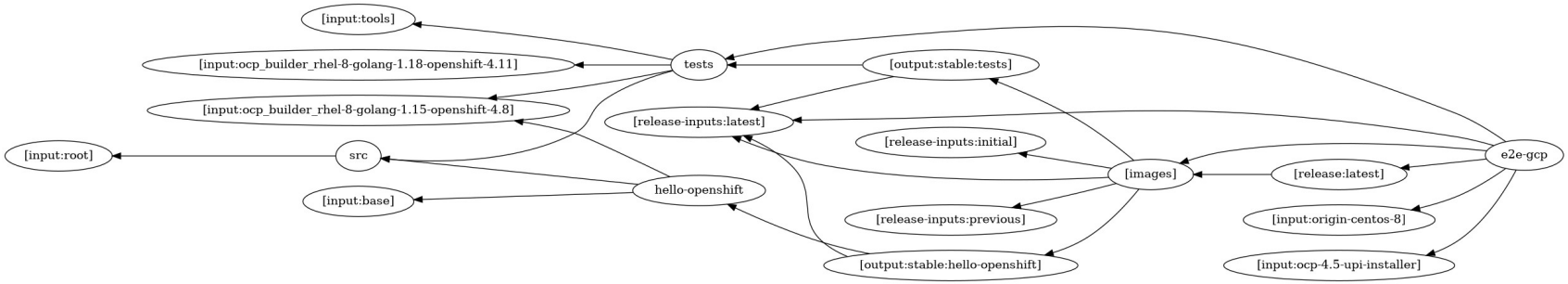
Red Hat

https://gcsweb-ci.apps.ci.l2s4.p1.openshiftapps.com/gcs/origin-ci-test/pr-logs/pull/27275/pull-ci-openshift-origin-master-e2e-gcp/1540385791396548608/prowjob.json*

```
command:
- ci-operator
args:
- --gcs-upload-secret=/secrets/gcs/service-account.json
- --image-import-pull-secret=/etc/pull-secret/.dockerconfigjson
- --lease-server-credentials-file=/etc/boskos/credentials
- --report-credentials-file=/etc/report/credentials
- --secret-dir=/secrets/ci-pull-credentials
- --secret-dir=/usr/local/e2e-gcp-cluster-profile
- --target=e2e-gcp
```

* *how-tos: document artifacts directory* #266 – openshift/ci-docs

Red Hat


[input:tools]
[input:ocp_builder_rhel-8-golang-1.18-openshift-4.11]
tests
[output:stable:tests]
[input:ocp_builder_rhel-8-golang-1.15-openshift-4.8]
[release-inputs:latest]
[release-inputs:initial]
[input:root]
src
[images]
[release:latest]
e2e-gcp
hello-openshift
[input:base]
[release-inputs:previous]
[input:origin-centos-8]
[output:stable:hello-openshift]
[input:ocp-4.5-upi-installer]

Red Hat

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
```


Red Hat

Test types

Red Hat

ibid

```
- as: verify-deps
  commands: make verify-deps …
  container:
    from: src
```

…/openshift-origin-release-3.11.yaml

```
- as: e2e-gcp
  commands: … run-tests
  openshift_ansible:
    cluster_profile: gcp
```


Red Hat

https://github.com/openshift/ci-tools/blob/master/pkg/api/types.go

```
type TestStepConfiguration struct {
    As string `json:"as"`
    Commands string `json:"commands,omitempty"`
    // …
    // Only one of the following can be not-null.
    ContainerTestConfiguration                          …
    MultiStageTestConfiguration                         …
    MultiStageTestConfigurationLiteral                  …
    OpenshiftAnsibleClusterTestConfiguration            …
    OpenshiftAnsibleSrcClusterTestConfiguration         …
    OpenshiftAnsibleCustomClusterTestConfiguration      …
    OpenshiftInstallerClusterTestConfiguration          …
    OpenshiftInstallerUPIClusterTestConfiguration       …
    OpenshiftInstallerUPISrcClusterTestConfiguration    …
    OpenshiftInstallerCustomTestImageClusterTestConfiguration …
}
```


Red Hat

```
// Only one of the following can be not-null.
ContainerTestConfiguration \
    *ContainerTestConfiguration \
    `json:"container,omitempty"`
// …
```


Red Hat

```
type ContainerTestConfiguration struct {
    From PipelineImageStreamTagReference
    MemoryBackedVolume *MemoryBackedVolume
    Clone *bool
}
```


Red Hat

```
type TestStepConfiguration struct {
    As string
    Commands string
    Cluster Cluster
    Secret *Secret
    Secrets []*Secret
    Cron *string
    Interval *string
    ReleaseController bool
    Postsubmit bool
    ClusterClaim *ClusterClaim
    RunIfChanged string
    Optional bool
    SkipIfOnlyChanged string
    Timeout *prowv1.Duration
    // …
```


Red Hat

```
- as: e2e-gcp
  commands: … run-tests
  openshift_ansible:
    cluster_profile: gcp
```

```
args:
- --image-import-pull-secret=/etc/pull-secret/.dockerconfigjson
- --report-credentials-file=/etc/report/credentials
- --secret-dir=/usr/local/e2e-gcp-periodic-cluster-profile
- --target=e2e-gcp-periodic
- --template=/usr/local/e2e-gcp-periodic
- --gcs-upload-secret=/secrets/gcs/service-account.json
command:
- ci-operator
```


Red Hat

```
    volumeMounts:
    - mountPath: /usr/local/e2e-gcp-periodic
      name: job-definition
      subPath: cluster-launch-e2e.yaml
  volumes:
  - configMap:
      name: prow-job-cluster-launch-e2e
    name: job-definition
```


Red Hat

https://github.com/openshift/release/tree/master/ci-operator/templates

```
ci-operator/templates/
    master-sidecar-3.yaml
    master-sidecar-4.4.yaml
    openshift/
        installer/
            cluster-launch-installer-custom-test-image.yaml
            cluster-launch-installer-e2e.yaml
            cluster-launch-installer-libvirt-e2e.yaml
            cluster-launch-installer-metal-e2e.yaml
            cluster-launch-installer-openstack-e2e.yaml
            cluster-launch-installer-openstack-upi-e2e.yaml
            cluster-launch-installer-src.yaml
            cluster-launch-installer-upi-e2e.yaml
        openshift-ansible/
            cluster-launch-e2e-openshift-ansible.yaml
            cluster-launch-e2e.yaml
            cluster-scaleup-e2e-40.yaml
```


Red Hat

.../openshift/installer/cluster-launch-installer-e2e.yaml

```
kind: Template
apiVersion: template.openshift.io/v1

parameters:
- name: JOB_NAME
  required: true
- name: JOB_NAME_SAFE
  required: true
# …
```


Red Hat

```
objects:
# We want the cluster to be able to access
# these images
- kind: RoleBinding
  apiVersion: authorization.openshift.io/v1
  metadata:
    name: ${JOB_NAME_SAFE}-image-puller
    namespace: ${NAMESPACE}
  # …
```


Red Hat

```
# The e2e pod spins up a cluster, runs e2e tests,
# and then cleans up the cluster.
- kind: Pod
  apiVersion: v1
  metadata:
    name: ${JOB_NAME_SAFE}
    namespace: ${NAMESPACE}
  # …
```


Red Hat

```
containers:
# Once the cluster is up, executes shared tests
- name: test
# …
# Runs an install
- name: setup
# …
# Performs cleanup of all created resources
- name: teardown
# …
```


Red Hat

```
parameters:
# …
- name: IMAGE_FORMAT
- name: IMAGE_INSTALLER
  required: true
- name: IMAGE_TESTS
  required: true
# …
- name: RELEASE_IMAGE_LATEST
  required: true
# …
```


Red Hat

- test definition
- test phases
  - pre
  - test
  - post
- step registry
  - references
  - chains
  - workflows
  - observers

- container image
- parameters
- dependencies
- credentials
- leases
- overriding
- ...

Red Hat

```
type MultiStageTestConfiguration struct {
    ClusterProfile ClusterProfile
    Pre []TestStep
    Test []TestStep
    Post []TestStep
    Workflow *string
    Environment TestEnvironment
    Dependencies TestDependencies
    DNSConfig *StepDNSConfig
    Leases []StepLease
    AllowSkipOnSuccess *bool
    AllowBestEffortPostSteps *bool
    Observers *Observers
    DependencyOverrides DependencyOverrides
}
```


Red Hat

```
type MultiStageTestConfigurationLiteral struct {
    ClusterProfile ClusterProfile
    Pre []LiteralTestStep
    Test []LiteralTestStep
    Post []LiteralTestStep
    Environment TestEnvironment
    Dependencies TestDependencies
    DNSConfig *StepDNSConfig
    Leases []StepLease
    AllowSkipOnSuccess *bool
    AllowBestEffortPostSteps *bool
    Observers []Observer
    DependencyOverrides DependencyOverrides
    Timeout *prowv1.Duration
}
```


Red Hat

```
type LiteralTestStep struct {
    As string
    From string
    FromImage *ImageStreamTagReference
    Commands string
    Resources ResourceRequirements
    Timeout *prowv1.Duration
    GracePeriod *prowv1.Duration
    Credentials []CredentialReference
    Environment []StepParameter
    Dependencies []StepDependency
```


Red Hat

```
    DNSConfig *StepDNSConfig
    Leases []StepLease
    OptionalOnSuccess *bool
    BestEffort *bool
    Cli string
    Observers []string
    RunAsScript *bool
}
```


Red Hat

```
type TestStep struct {
    *LiteralTestStep
    Reference *string
    Chain *string
}
```


Red Hat

```
tests:
- as: multi-stage
  steps: # …
- as: multi-stage-literal
  literal_steps: # …

$ JOB_SPEC=… ci-operator
$ ci-operator --config …
$ ci-operator --unresolved-config …
$ CONFIG_SPEC=… ci-operator …
$ UNRESOLVED_CONFIG=… ci-operator …
```


Red Hat

Releases

Red Hat

https://github.com/openshift/ci-tools/blob/master/pkg/api/types.go

```
type ReleaseBuildConfiguration struct {
    Metadata Metadata
    InputConfiguration
    // …
}

type InputConfiguration struct {
    // …
    Releases map[string]UnresolvedRelease
}
```


Red Hat

```
type UnresolvedRelease struct {
    // Integration describes an integration stream
    // which we can create a payload out of
    Integration *Integration
    // Candidate describes a candidate release
    // payload
    Candidate *Candidate
    // Prerelease describes a yet-to-be released
    // payload
    Prerelease *Prerelease
    // Release describes a released payload
    Release *Release
}
```


Red Hat

```
type Candidate struct {
    Product ReleaseProduct
    Architecture ReleaseArchitecture
    Stream ReleaseStream
    Version string
    Relative int
}

type Prerelease struct {
    Product ReleaseProduct
    Architecture ReleaseArchitecture
    VersionBounds VersionBounds
}
```


Red Hat

```
pkg/release/
    candidate/
        client.go
        types.go
    client.go
    config/
        client.go
        config.go
    official/
        client.go
        types.go
    prerelease/
        client.go
```


Red Hat

- `candidate` / `prerelease`
  - `https://amd64.ocp.releases.ci.openshift.org`
  - release controller
- `release`
  - `https://api.openshift.com/api/upgrades_info/v1/graph`
  - cincinnati

Red Hat

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

- `ReleaseImagesTagStep`
  - source → destination `ImageStream`
  - `$namespace/$name` → `ci-op-*/stable*`
- `AssembleReleaseStep`
  - `ImageStream` → release payload
  - `stable*` → `release:*`
  - will wait for built images if `include_built_images`

Red Hat

```
tag_specification:
  namespace: ocp
  name: "4.10"
```

- always `initial` and `latest`
- `include_built_images` implicitly for `latest`
- `ReleaseImagesTagStep` (≈ `ReleaseSnapshotStep`)
- `RELEASE_IMAGE_*`

Red Hat

```
$ oc adm release extract \
    --file image-references \
    quay.io/openshift/okd:4.10.0-0.okd-2022-07-09-073606 \
    | yaml
kind: ImageStream
apiVersion: image.openshift.io/v1
metadata:
  name: 4.10.0-0.okd-2022-07-09-073606
  creationTimestamp: 2022-07-10T09:12:53Z
  annotations:
    release.openshift.io/from-image-stream: >
      origin/4.10-2022-07-09-073606
    release.openshift.io/from-release: >
      registry.ci.openshift.org/origin/release:4.10.0-0.…
…
```


Red Hat

```
spec:
  lookupPolicy:
    local: false
  tags:
  - name: alibaba-cloud-controller-manager
    annotations:
      io.openshift.build.commit.id: 0
      io.openshift.build.commit.ref: release-4.10
      io.openshift.build.source-location: >
        https://github.com/openshift/…
    from:
      kind: DockerImage
      name: quay.io/openshift/okd-content@sha256:…
    generation: null
    importPolicy:
    referencePolicy:
      type: 0
…
```


Red Hat

```
releases:
  latest:
    release:
       architecture: amd64
       channel: stable
       version: "4.10"
```

- candidate / prerelease
- ImportReleaseStep
  - release payload → ImageStream
  - $src → release:*
  - tags → ImageStream
  - release:* → oc … extract → stable*

Red Hat

Input images

Red Hat

"CI cycle"

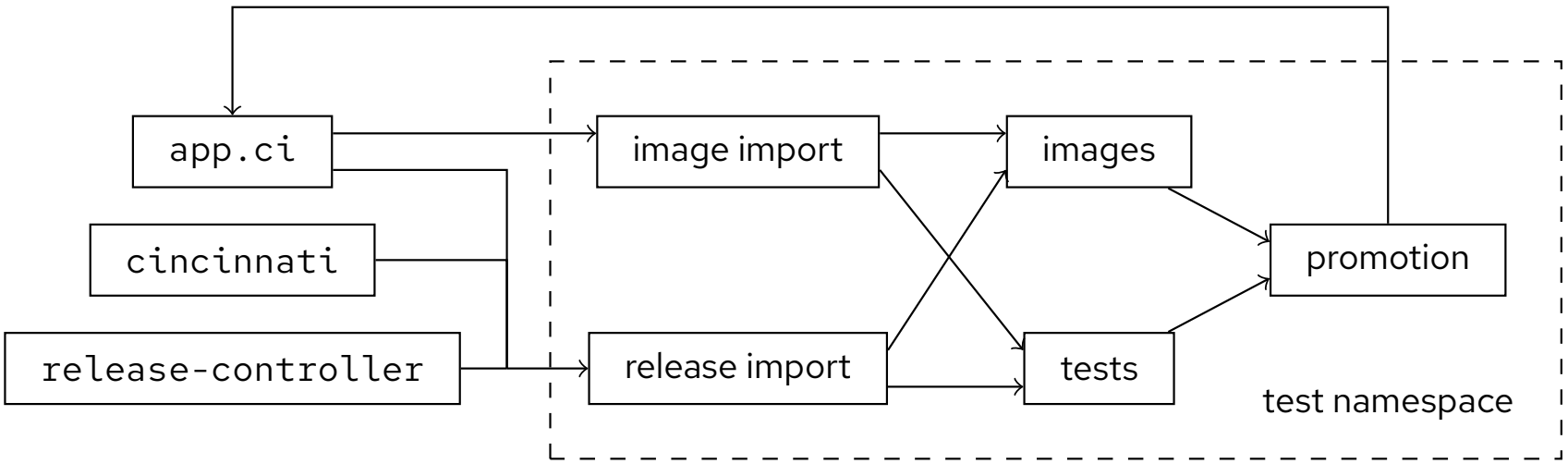
0. import images / releases
1. build images
2. execute tests
3. promote images
4. `goto 0`

Red Hat

"CI cycle"

-1. ?
 0. import images / releases
 1. build images
 2. execute tests
 3. promote images
 4. `goto 0`

Red Hat


app.ci
cincinnati
release-controller
image import
release import
images
tests
promotion
test namespace

Red Hat

- supplemental images
  - `https://github.com/openshift/release/tree/master/clusters/app.ci/supplemental-ci-images`
  - `BuildConfigs`
- image mirroring
  - `https://github.com/openshift/release/tree/master/core-services/image-mirroring`
  - Quay/etc. ↔ `app.ci`
- ART / OCP builder images
  - `https://docs.ci.openshift.org/docs/architecture/images/`
  - `https://github.com/openshift/ocp-build-data.git`

Red Hat

- supplemental images
  - `https://github.com/openshift/release/tree/master/clusters/app.ci/supplemental-ci-images`
  - `BuildConfigs`
  - `registry.ci.openshift.org/ci/ci-tools-build-root`
- image mirroring
  - `https://github.com/openshift/release/tree/master/core-services/image-mirroring`
  - Quay/etc. ↔ `app.ci`
  - `registry.ci.openshift.org/coreos/stream9:9`
- ART / OCP builder images
  - `https://docs.ci.openshift.org/docs/architecture/images/`
  - `https://github.com/openshift/ocp-build-data.git`
  - `registry.ci.openshift.org/ocp/builder:…`

Red Hat

dptp-controller-manager

- cmd/dptp-controller-manager/
- pkg/controller/test-images-distributor/

Red Hat

```
args:
…
- --enable-controller=test_images_distributor
- --enable-controller=promotionreconciler
- --enable-controller=serviceaccount_secret_refresher
- --enable-controller=testimagestreamimportcleaner
…
```


Red Hat

```
…
- --release-repo-git-sync-path=/var/repo/release
- --kubeconfig-dir=/var/kubeconfigs
- --registry-cluster-name=app.ci
- --testImagesDistributorOptions \
    .additional-image-stream-tag=ocp/builder:golang-1.10
…
- --testImagesDistributorOptions \
    .additional-image-stream-tag= \
    ocp/builder:rhel-7-golang-1.11
…
- --testImagesDistributorOptions \
    .additional-image-stream-namespace=ci
- --testImagesDistributorOptions \
    .additional-image-stream=rhcos/machine-os-content
…
```


Red Hat

pkg/api/helper/imageextraction.go

- `TestInputImageStreamsFromResolvedConfig`
- `TestInputImageStreamTagsFromResolvedConfig`

Red Hat

```
https://prow.ci.openshift.org/view/gs/origin-ci-test/
logs/branch-ci-openshift-ci-tools-master-images/
1561993456950185984

…
Building src
Build src succeeded after 4m48s
Building bin
Build bin succeeded after 25m54s
Building determinize-peribolos
Building ci-secret-generator
Building ci-operator-config-mirror
…
```

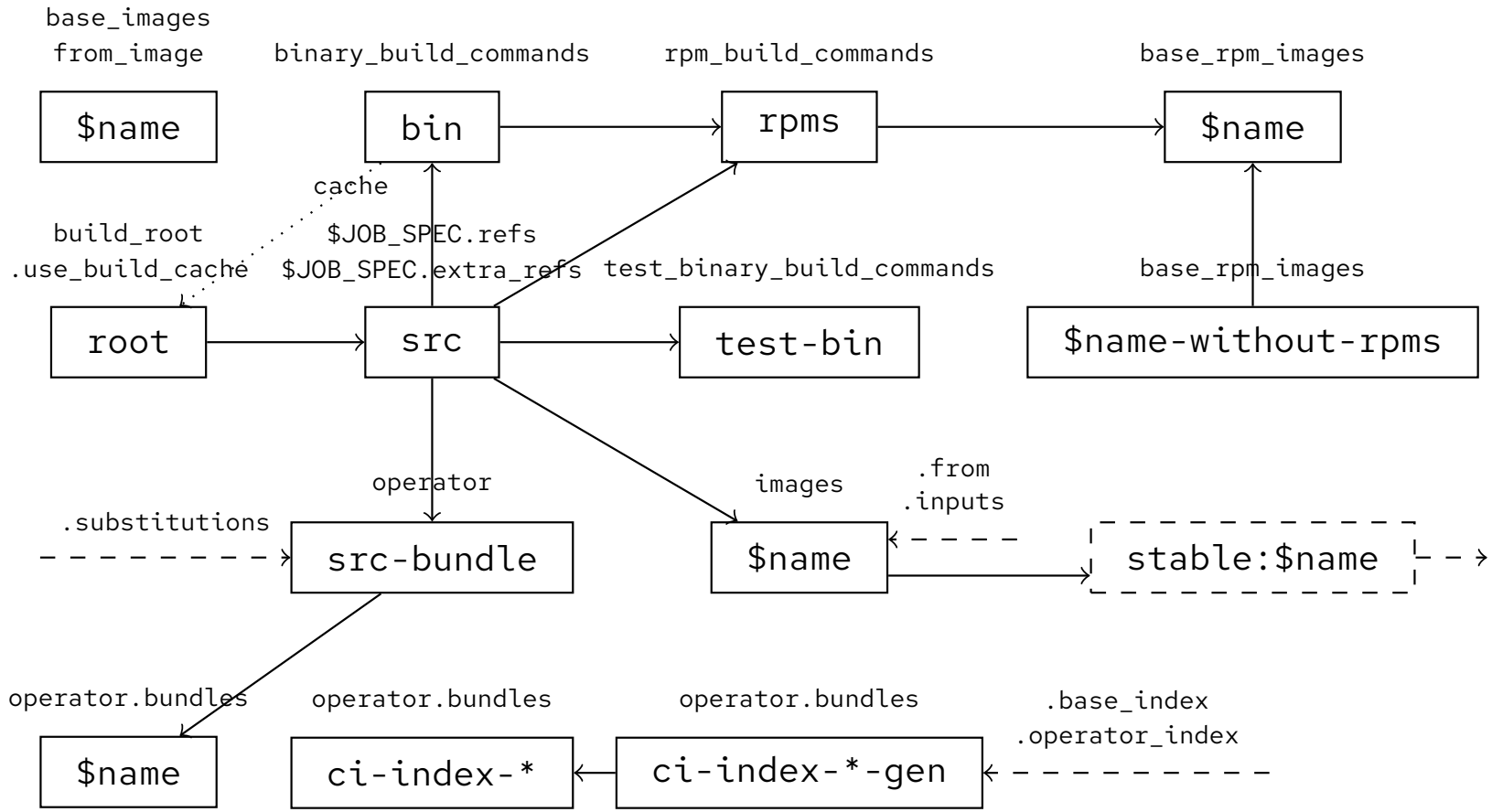

Red Hat

```
…
Build prcreator succeeded after 14m26s
Tagging prcreator into stable
Build private-prow-configs-mirror \
    succeeded after 15m51s
Tagging private-prow-configs-mirror into stable
Promoting tags to ci/${component}:latest: \
    applyconfig, auto-aggregator-job-names, \
    auto-config-brancher, auto-peribolos-sync, \
    auto-sippy-config-generator, …
Ran for 1h7m10s
```


Red Hat

Image pipeline

Red Hat


base_images
from_image
$name
binary_build_commands
bin
rpm_build_commands
rpms
base_rpm_images
$name
cache
build_root
.use_build_cache
$JOB_SPEC.refs
$JOB_SPEC.extra_refs
test_binary_build_commands
base_rpm_images
root
src
test-bin
$name-without-rpms
operator
images
.from
.inputs
.substitutions
src-bundle
$name
stable:$name
operator.bundles
operator.bundles
operator.bundles
.base_index
.operator_index
$name
ci-index-*
ci-index-*-gen

Red Hat

Thank you

Red Hat