



ci-operator multi-stage tests

Bruno Barcarol Guimarães

Introduction

- ▶ Motivation

Test definitions

- ▶ Phases
- ▶ Images
- ▶ Credentials
- ▶ Parameters
- ▶ Dependencies
- ▶ etc.

Step registry

- ▶ Discoverable
- ▶ Referenceable
- ▶ Verifiable
- ▶ Reusable

- ▶ https://docs.google.com/document/d/1md-1BMf4_7mtKgGVoeZ3j0h4zSIBSjwl6vTTAYESwIM
 - ▶ *Multi-Stage Tests Design Document*
- ▶ <https://docs.ci.openshift.org>
 - ▶ docs/architecture/step-registry
 - ▶ docs/architecture/ci-operator

ca. Aug 2019

- ▶ Two test types.
 - ▶ container
 - ▶ template
- ▶ Desire to create tests for increasingly varied scenarios.
- ▶ Existing tests already complex and barely maintained.

Ah, the templates...

complex, esoteric and fragile

difficult to extend and use

not able to share common test logic

duplication and fragmentation

- ▶ Small number of extremely complex Pod definitions.
 - ▶ Python embedded in Bash embedded in YAML embedded in ...
 - ▶ Each responsible for the entire execution of an E2E test.
- ▶ Equally small set of people willing to / capable of “maintaining” them.
- ▶ Adding a new test scenario
 - ▶ copying an existing template (thousands of lines of YAML)
 - ▶ minor edits
 - ▶ (extreme duplication)
- ▶ Configuration exposed and required knowledge of byzantine implementation details of `ci-operator`.

- ▶ Small number of extremely complex Pod definitions.
 - ▶ Python embedded in Bash embedded in YAML embedded in ...
 - ▶ Each responsible for the entire execution of an E2E test.
- ▶ Equally small set of people willing to / capable of “maintaining” them.
- ▶ Adding a new test scenario
 - ▶ copying an existing template (thousands of lines of YAML)
 - ▶ minor edits
 - ▶ (extreme duplication)
- ▶ Configuration exposed and required knowledge of byzantine implementation details of `ci-operator`.
- ▶ *etc.*

Test definitions

- ▶ regular ci-operator test
 - ▶ images
 - ▶ release images
 - ▶ artifacts
 - ▶ cluster profiles
 - ▶ ...

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

```
tests:
```

```
- as: e2e  
  steps:  
    test:  
      - as: e2e  
        commands: ... make e2e  
        from: test-bin  
        # ...
```

```
tests:  
- as: e2e  
  steps:  
    test:  
      - as: e2e  
        commands: ... make e2e  
        from: test-bin  
        # ...
```

```
tests:  
- as: e2e  
  commands: ... make e2e  
  container:  
    from: test-bin  
    # ...
```

- ▶ pre/test/post
- ▶ serial execution
 - ▶ “short-circuit” execution for pre/test
 - ▶ post steps always executed
- ▶ each step corresponds to a Pod
 - ▶ shared data can be placed in a special directory

\$SHARED_DIR

- ▶ Small storage space for inter-step data.
- ▶ Implemented using a Kubernetes Secret.
- ▶ Hard 1MB limit, no directories.
- ▶ Completely rewritten by the contents of the directory in the pod after the step script is executed.
- ▶ State in the ephemeral cluster can be used for higher-bandwidth communication between steps.
- ▶ `kubeconfig` is treated especially.
- ▶ Data intended for debugging tests should be placed in the artifacts directory.

Images

- ▶ from
 - ▶ pipeline images
 - ▶ root, src, bin, ...
 - ▶ base_images
 - ▶ images
 - ▶ "stable" images
 - ▶ releases
 - ▶ tag_specification
- ▶ from_image
 - ▶ \approx base_images
 - ▶ from_image:
 - namespace: ocp
 - name: upi-installer
 - tag: 4.12

Credentials

- ▶ Vault → build cluster → test namespace → test pod
- ▶ ci-operator must have access to the source namespace.
- ▶ The test-credentials namespace is pre-configured for regular users.
- ▶ Supplanted old methods.
 - ▶ secret
 - ▶ secrets
 - ▶ --secret-dir
 - ▶ etc.
- ▶ credentials:
 - namespace: ns
 - name: name
 - mount_path: /path

Parameters

- ▶ Key/value data declared in a step.
- ▶ Ultimately become environmental variables.
- ▶ Can be overridden (coming soon).

```
as: openshift-e2e-test
from: tests
commands: openshift-e2e-test-commands.sh
env:
- name: TEST_SUITE
  default: openshift/conformance/parallel
  documentation: |
    The test suite to run. Use 'openshift-test
    run --help' to list available suites.
# ...
```


Dependencies

- ▶ `ci-operator` image *pull spec* → test pod
- ▶ Establishes images → test dependency.
- ▶ `as: test-step`
dependencies:
 - name: pipeline:bin
env: BIN_IMG
 - name: release:4.12
env: RELEASE_4_12
- ▶ `#!/bin/bash`
use "\$BIN_IMG"
use "\$RELEASE_4_12"

```
# openshift-e2e-tests-ref.yaml
dependencies:
- name: "release:latest"
  env: OPENSIFT_UPGRADE_RELEASE_IMAGE_OVERRIDE

# openshift-e2e-tests-commands.sh
openshift-tests run-upgrade \
  "${TEST_UPGRADE_SUITE}" \
  --to-image \
    "${OPENSIFT_UPGRADE_RELEASE_IMAGE_OVERRIDE}" \
  --options "${TEST_UPGRADE_OPTIONS-}" \
  --provider "${TEST_PROVIDER}" \
  -o "${ARTIFACT_DIR}/e2e.log" \
  --junit-dir "${ARTIFACT_DIR}/junit"
```

Leases

- ▶ `ci-operator` → Boskos → test pod
- ▶ Generalization of implicit lease added by cluster profiles
- ▶ Leased resource name is available to the test script via environmental variable.
- ▶ `leases`:
 - `env`: `OVIRT_UPGRADE_LEASED_RESOURCE`
`resource_type`: `ovirt-upgrade-quota-slice`
`count`: 42

- ▶ best-effort steps
- ▶ catalogues / optional operators
- ▶ KUBECONFIG
- ▶ cluster profiles
- ▶ oc CLI injection
- ▶ no ServiceAccount credentials
- ▶ cluster claims
- ▶ VPN connection
- ▶ ...

Step registry

Goals

- ▶ discoverable
- ▶ referenceable
- ▶ verifiable
- ▶ reusable

- ▶ <https://steps.ci.openshift.org>
- ▶ <https://steps.ci.openshift.org/workflow/ipi-aws>
- ▶ <https://steps.ci.openshift.org/chain/ipi-aws-pre>
- ▶ <https://steps.ci.openshift.org/reference/ipi-install-install>

```
https://prow.ci.openshift.org/view/gs/origin-ci-test/logs/  
periodic-ci-openshift-release-master-okd-4.10-e2e-vsphere/  
1579723667426775040
```

```
Running step e2e-vsphere-ipi-install-install.  
Logs for container test in pod e2e-vsphere-ipi-install-install:  
...  
Step e2e-vsphere-ipi-install-install failed after 23m20s.  
Step phase pre failed after 40m10s.  
...  
Link to step on registry info site: ...  
Link to job on registry info site: ...
```


`https://steps.ci.openshift.org/workflow/ipi-aws#approvers`

- ▶ wking
- ▶ vrutkovs
- ▶ abhinavdahiya
- ▶ deads2k
- ▶ staebler
- ▶ technical-release-team-approvers
- ▶ jianlinliu
- ▶ yunjiang29

- ▶ pull-ci-openshift-release-master-step-registry-shellcheck
- ▶ <https://www.shellcheck.net>
- ▶

```
find ci-operator/step-registry -name '*.sh' -print0 \  
    | xargs -0 -n1 shellcheck -S warning
```

- ▶ reference
- ▶ chain
- ▶ workflow

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

ref:

```
as: ipi-install-install
from: installer
grace_period: 10m
commands: ipi-install-install-commands.sh
cli: latest
resources:
  requests:
    cpu: 1000m
    memory: 2Gi
```

(cont.)

(cont.)

```
credentials:
- namespace: test-credentials
  name: loki-stage-collector-test-secret
  mount_path: /var/run/loki-secret
# ...
env:
- name: OPENSIFT_INSTALL_EXPERIMENTAL_DUAL_STACK
  default: "false"
  documentation: Using experimental Azure dual-stack support
# ...
dependencies:
- name: "release:latest"
  env: OPENSIFT_INSTALL_RELEASE_IMAGE_OVERRIDE
# ...
documentation: |-
  The IPI install step runs the OpenShift Installer ...
```

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

chain:

as: ipi-aws-pre

steps:

- chain: ipi-conf-aws

- chain: ipi-install

documentation: |-

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

<https://steps.ci.openshift.org/workflow/ipi-aws>

workflow:

```
as: ipi-aws
```

```
steps:
```

```
  pre:
```

```
    - chain: ipi-aws-pre
```

```
  post:
```

```
    - chain: ipi-aws-post
```

```
documentation: |-
```

```
The IPI workflow provides pre- and post- steps that provision and deprovision an OpenShift cluster with a default configuration on AWS, allowing job authors to inject their own end-to-end test logic.
```

All modifications to this workflow should be done by modifying the `ipi-aws-{pre,post}`chains` to allow other workflows to mimic and extend this base workflow without a need to backport changes.

```
as: e2e-aws
```

```
steps:
```

```
  pre:
```

- as: conf-this
 commands: # ...
- as: conf-that
 commands: # ...
- as: install
 commands: # ...
- as: rbacs
 commands: # ...

```
test:
```

- as: test
 commands: # ...

```
post:
```

- as: gather-this
 commands: # ...
- as: gather-that
 commands: # ...
- as: uninstall
 commands: # ...


```
as: e2e-aws
```

```
steps:
```

```
  pre:
```

- ref: conf-this
- ref: conf-that
- ref: install
- ref: rbacs

```
# ...
```

```
ref:
```

```
  as: conf-this
```

```
  commands: # ...
```

```
ref:
```

```
  as: conf-that
```

```
  commands: # ...
```

```
...
```

```
as: e2e-aws
steps:
  pre:
    - chain: aws-pre
  test: # ...
  post:
    - chain: aws-post
```

```
chain:
as: aws-pre
  steps:
    - ref: conf-this
    - ref: conf-that
    - ref: install
    - ref: rbacs
```

```
chain:
as: aws-post
steps:
# ...
```

```
as: e2e-aws
steps:
  workflow: aws-ipi
  test: # ...
```

```
workflow:
  as: aws-ipi
  pre:
    - chain: aws-pre
  post:
    - chain: aws-post
```

```
as: e2e-aws
steps:
  workflow: aws-ipi
```

```
workflow:
  as: openshift-e2e-aws
  pre:
    - chain: aws-pre
  test:
    - ref: openshift-e2e-test
  post:
    - chain: aws-post
```

```
$ find ci-operator/step-registry/ -name 'ipi-conf-*-ref.yaml' \  
  | wc -l  
75  
$ find ci-operator/step-registry/ -name 'ipi-conf-*-ref.yaml' \  
  | sed 's,.*/,,'; s/-ref\.yaml//' | shuf | head -15 | sort  
ipi-conf-additional-enabled-capabilities  
ipi-conf-alibabacloud  
ipi-conf-azure-provisioned-des  
ipi-conf-azurestack-creds  
ipi-conf-azure-vmgenv1  
ipi-conf-azure-workers-marketimage  
ipi-conf-etcd-on-ramfs  
ipi-conf-libvirt  
ipi-conf-openstack-enable-octavia  
ipi-conf-ovirt-generate-csi-test-manifest  
ipi-conf-ovirt-generate-csi-test-manifest-release-4.6-4.8  
ipi-conf-ovirt-generate-install-config  
ipi-conf-ovirt-generate-install-config-params  
ipi-conf-ovirt-generate-ovirt-config  
ipi-conf-vsphere-zones
```

[https://docs.ci.openshift.org/docs/architecture/step-registry/
#hierarchical-propagation](https://docs.ci.openshift.org/docs/architecture/step-registry/#hierarchical-propagation)

```
as: openshift-e2e-test
env:
- name: TEST_SUITE
# ...
```

```
tests:  
- as: e2e  
  steps:  
    test:  
    - ref: openshift-e2e-test  
  env:  
    TEST_SUITE: openshift/conformance/parallel
```

```
workflow:  
  as: openshift-e2e-serial  
  steps:  
    test:  
      - ref: openshift-e2e-test  
    env:  
      TEST_SUITE: openshift/conformance/serial  
  
tests:  
- as: e2e  
  steps:  
    workflow: openshift-e2e-serial
```


Thank you