

OpenLab

Testing Combinations of Cloud OSS

Open Source Summit Japan 2019

Kenichi Omichi, NEC



Who am I: Kenichi Omichi

- Principle Software Engineer from NEC ECT (CA, USA)

- Open Source Contributor**

- Approver of kubernetes/kubernetes repo(/test)

- Core Developer of OpenStack/Nova and OpenStack/QA

- Ex-PTL of OpenStack/QA

- Member of OpenLab

- Initial creator of Linux crash dump tool (makedumpfile)

- CKA / CKAD

Agenda of this session

1. OpenLab overview
2. Kubernetes on OpenStack Testing
3. Kubernetes E2E and Conformance Tests
4. Summary

1. OpenLab Overview

What is OpenLab



OpenLab is the playground for open source ecosystem testing, development, and collaboration

- Open testbed for combinations related to cloud OSS with community base
- To improve the usability, reliability and resiliency of tools and applications for hybrid and multi-cloud environments
- e.g. Deploy Kubernetes cluster on top of OpenStack, then test the cluster

Multiple companies provide donations of hardware and people



Test Jobs on OpenLab



Test jobs are controlled with Zuul

All test jobs in OpenLab can be seen from Zuul dashboard

- <http://status.openlabtesting.org/jobs>

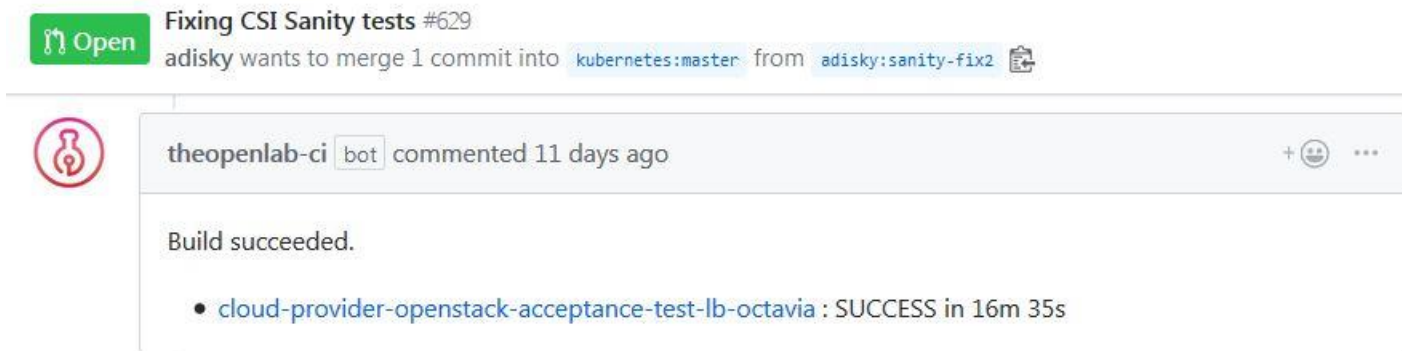
e.g. Part of all test jobs in OpenLab

- cloud-provider-openstack-test (Kubernetes on OpenStack)
- terraform-provider-test (for OpenStack, huaweicloud, teleornica)
- spark-integration-test (kubeadm-k8s, minikube-k8s)
- tensorflow-benchmarks (kubeflow-k8s)

Test Jobs for Upstream Development

cloud-provider-openstack-test is a test job for upstream development of kubernetes/cloud-provider-openstack

1. When proposing a PR to the github, the test job is kicked on OpenLab and the job result is reported back to the github

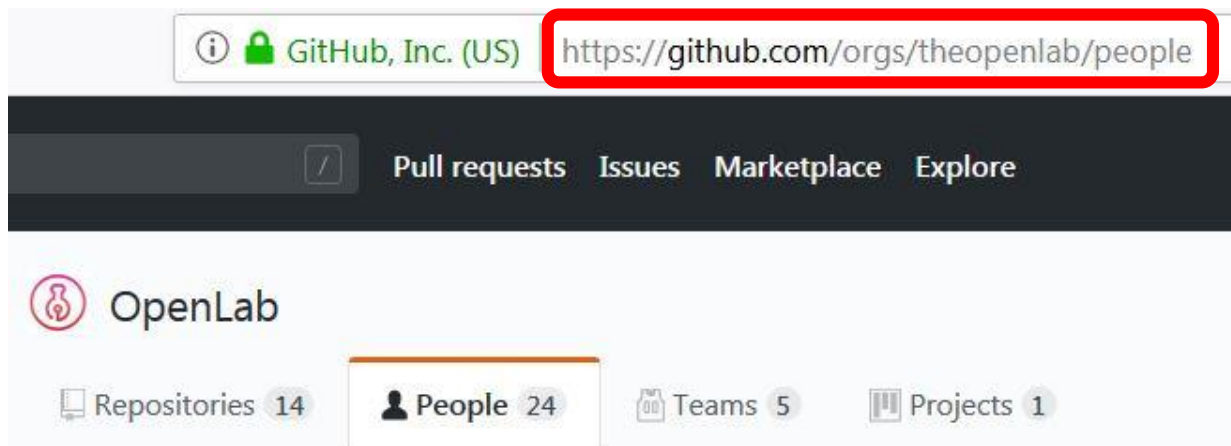


2. PR author and reviewers confirm the result, then the PR can be merged after checking the code also is good on the PR

OpenLab welcomes proposals for new test

If interested in adding test to OpenLab, be a member

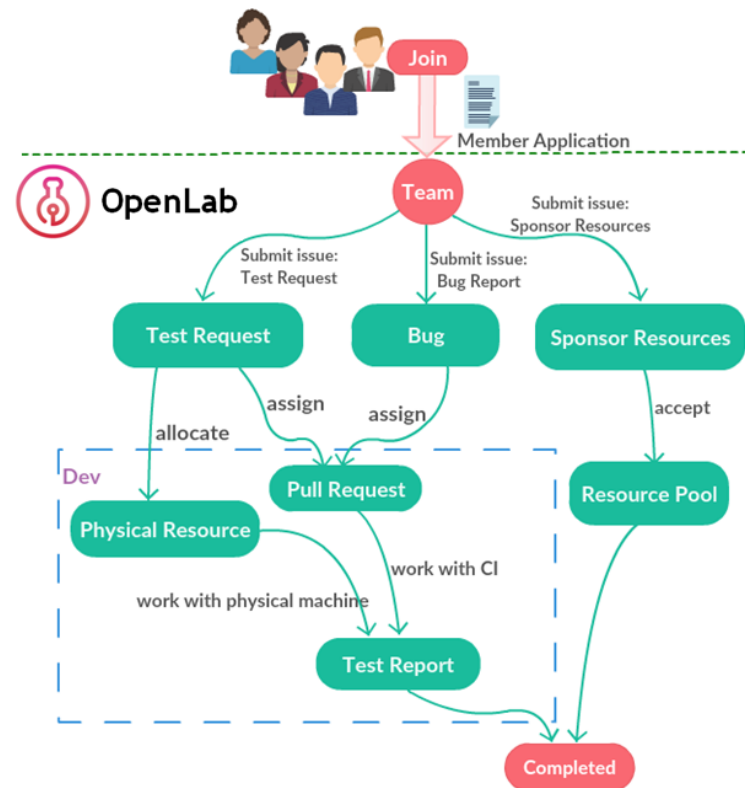
1. Open **'Join in OpenLab'** issue
2. Fulfill name, github id, email and company then submit it
3. OpenLab team sends the github invitation later
4. Confirm becoming a member by checking **people** on the github



New test proposal

After becoming a member, you can propose a new test combination of OSS

1. Open **Test Request** issue
2. Fulfill focus(Performance, Reliability, Integration or so on), OSS, machines (VM or Baremetal), OS, etc
3. OpenLab team reviews the request. If fine, test resource is allocated and test job starts running



Challenge of OSS combination tests

One purpose of OSS combination test is to detect integration failures between OSS

- An OSS depends on another OSS, the depended OSS makes a backwards incompatible change with version-up, the depending OSS can be broken

But it happens, hard to fix the failures

- Every OSS is developed by different community, they are different people
- Necessary to work together across communities, but hard and difficult

kubernetes/cloud-provider-openstack is a good practice

- Developers related to OpenStack are working in Kubernetes community
- We are contributing Kubernetes itself also
- When such issue happens, we work together with whole Kubernetes community

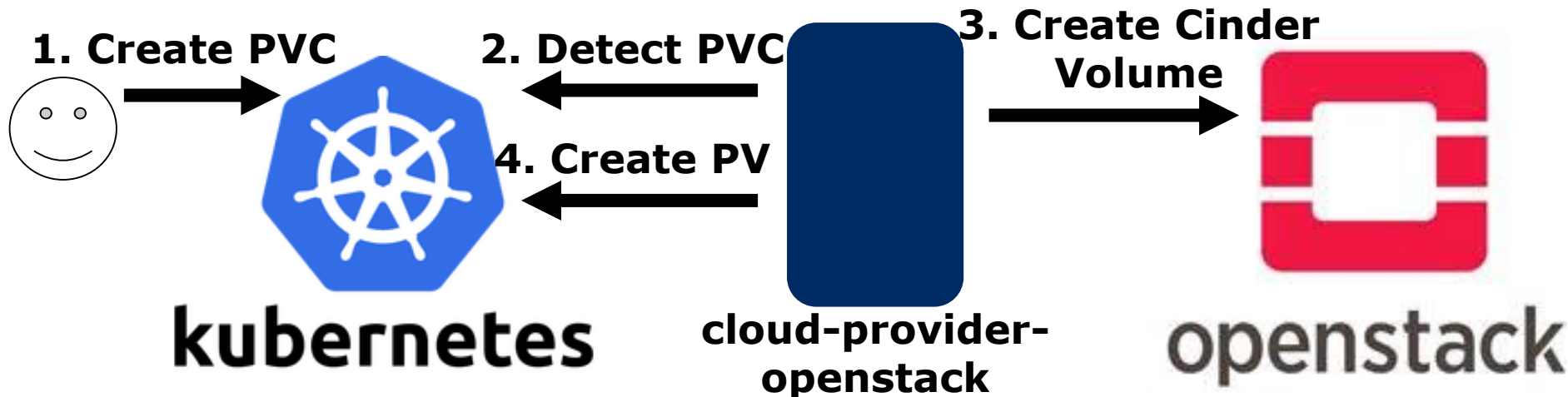
When proposing a new test of OSS combination, consider how to solve integration issues

2. Kubernetes on OpenStack Testing

A decorative graphic consisting of several thin, flowing orange lines that start from the top right and curve downwards and to the left, crossing the horizontal line that separates the title from the main content area.

Kubernetes Cloud Providers

- Some Kubernetes resources need help from IaaS
 - e.g. Actual data of PersistentVolume is stored into storage of IaaS
- Kubernetes Cloud Providers is abstraction layer for integrating with IaaS(AWS, GCE, Azure or OpenStack)
- kubernetes/cloud-provider-openstack is for integrating Kubernetes with OpenStack



cloud-provider-openstack Features

To integrate with OpenStack, cloud-provider-openstack implements

- Octavia Ingress Controller
- Cinder CSI Plugin
- Cinder Standalone Provisioner
- Manila CSI Plugin
- Manila Provisioner
- Keystone Webhook Authentication Authorization

Recommend to use Cinder CSI Plugin instead of Cinder Standalone provisioner based on community direction

Test Jobs for cloud-provider-openstack

Each feature is tested on OpenLab

Octavia Ingress Controller

- cloud-provider-openstack-acceptance-test-lb-octavia

Cinder CSI Plugin

- cloud-provider-openstack-acceptance-test-csi-cinder

Cinder Standalone Provisioner

- cloud-provider-openstack-acceptance-test-standalone-cinder

Manila CSI Plugin

- Not tested yet (Issue is submitted)

Manila Provisioner

- cloud-provider-openstack-acceptance-test-manila-provisioner

Keystone Webhook Authentication Authorization

- cloud-provider-openstack-acceptance-test-keystone-authentication-authorization

Inside of cloud-provider-openstack test job

■ e.g. Test job of Cinder CSI Plugin

1. Build cloud-provider-openstack binary with a PR
2. Create cloud-config based on OpenStack info(auth_url, project_id, username etc.)
3. Deploy local kubernetes cluster
4. Build docker image for Cinder CSI Plugin
5. Create Kubernetes resources(StatefulSet, Service, ServiceAccount, etc.) to enable Cinder CSI Plugin
6. Create Kubernetes resources for testing
 - StorageClass using Cindre CSI Plugin
 - PVC using the StorageClass
 - Pod using the PVC
7. Check the previous Pod is **Running**

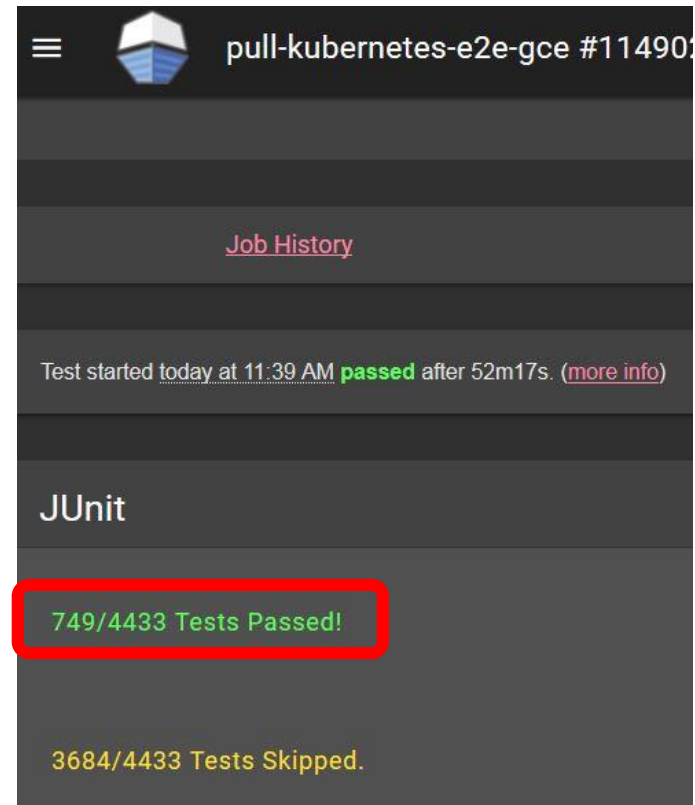
3. Kubernetes E2E and Conformance Tests

Kubernetes E2E Test

E2E(endpoint-to-endpoint) test is to verify Kubernetes API behaviors and the test is used for Kubernetes development

K8s v1.15.0 contains 4,400+ as E2E test, and 700+ tests are used for the development usually

- E2E contains for cloud-provider specific, Alpha APIs, disruptive and flake also which are skipped



Kubernetes Conformance Test

- Conformance Test(221) is subset of E2E test(4,400+) to verify **basic** Kubernetes features
- Conformance Test needs to follow the requirements:
 - It tests only GA, non-optional features or APIs
 - It works for all providers
 - It is not-privileged
 - etc.
- That means conformance test is expected to be passed on any environments

Conformance test is useful to verify your Kubernetes cluster before operating all E2E test

Certified Kubernetes Conformance Program

Kubernetes interoperability across vendors becomes important so that users can deploy clusters anywhere based on price, performance etc.

- Multiple cloud vendors provides managed Kubernetes services and a lot of Kubernetes clusters around the world

Certified Kubernetes Conformance Program is to verify the interoperability for each vendor

- 80+ vendors are certified and can use the log

The requirement is to pass Conformance Test on their own services/products



Conformance Test Improvement

PRs which make E2E promoted to Conformance are inflight

- By covering more features with conformance, users will be able to use more features across cloud vendors
- Difficult to check all conformance requirements for each PR by reviewers

In v1.15.0, we've added **conformance-requirements** check

- It checks `It works for all providers` only
- It detected violations in the existing conformance test and we fixed them
- We will extend requirement checks more

35/35 Tests Passed!

api-groups

bazel

boilerplate

cli-conventions

codegen

conformance-requirements

How to Run Conformance Test

Sonobuoy is easy way

1. Run Sonobuoy

```
$ go get -u -v github.com/heptio/sonobuoy
```

```
$ sonobuoy run
```

```
$ sonobuoy status
```

PLUGIN	STATUS	COUNT
e2e	running	1
systemd_logs	complete	3

Sonobuoy is still running. Runs can take up to 60 minutes.

2. Check the result

```
$ tail plugins/e2e/results/e2e.log
```

```
Ran 200 of 1946 Specs in 7965.569 seconds
```

```
SUCCESS! -- 200 Passed | 0 Failed | 0 Pending | 1746 Skipped PASS
```

Summary

1. OpenLab Overview

- If interested in adding OSS integration test, be OpenLab member
- When proposing a new test, consider how to solve integration issues

2. Kubernetes on OpenStack Testing

- cloud-provider-openstack is for integrating Kubernetes with OpenStack
- Recommend to use Cinder CSI Plugin instead of Cinder standalone provisioner based on community direction

3. Kubernetes E2E and Conformance Tests

- Conformance test is expected to be passed on any environments
- Conformance test is useful to verify your Kubernetes cluster before operating all E2E test

References

OpenLab: <https://openlabtesting.org/>

OpenLab Zuul Dashboard: <http://status.openlabtesting.org/jobs>

Certified Kubernetes Conformance Program:

<https://www.cncf.io/certification/software-conformance/>

Cinder CSI Plugin test: <https://github.com/theopenlab/openlab-zuul-jobs/tree/master/playbooks/cloud-provider-openstack-acceptance-test-csi-cinder>

Kubernetes Conformance Requirements:

<https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/conformance-tests.md#conformance-test-requirements>

Ref: How to Run Conformance Test (Hard way)

Build E2E test binary

1. Download Kubernetes source code

```
$ go get k8s.io/kubernetes
```

2. Checkout the same version as the target cluster

```
$ git checkout -b tag-v1.15.0
```

3. Build E2E test binary

```
$ sudo /usr/local/go/bin/go run hack/e2e.go -- --build
```

Run Conformance test

1. Set environment variables

```
$ export KUBECONFIG=$HOME/admin.conf  
$ export KUBERNETES_CONFORMANCE_TEST=true
```

2. Run conformance test

```
$ go run hack/e2e.go -- --provider=skeleton --test  
  --test_args= "--ginkgo.focus=¥[Conformance¥]"
```


 **Orchestrating** a brighter world

NEC