CNCF Cross-cloud CI
Project Intro
Project Intro

- Who, What, Why
- Demo
- Technology Overview
- Timeline & Events
- Q&A
Meet the Cross-cloud CI Team

- W. Watson @wavell
- Lucina Stricko @lixuna
- Denver Williams @denverwilliams
- Taylor Carpenter @taylor
- Joshua Smith* @nupejosh
- Robert Siekmann* @rsiekmann
**Why?** The CNCF ecosystem is large, diverse and continues to grow. CNCF would like to ensure cross-project interoperability and cross-cloud deployments of all cloud native technologies and show the daily status of builds and deployments on a status dashboard.
**What?** The Cross-cloud CI project consists of a cross-cloud testing system, status repository server and a dashboard.

The cross-cloud testing system has 3 components (build, cross-cloud, cross-project) that continually **validate the interoperability of each CNCF project for any commit on stable and head across all supported cloud providers.**

The cross-cloud testing system can reuse existing artifacts from a project’s preferred CI system or generate new build artifacts. The status repository server collects the test results and the dashboard displays them.
Goal: to target all CNCF projects

**Graduated**
- **kubernetes**
  - Orchestration
- Prometheus
  - Monitoring

**Incubating**
- **OPENTRACING**
  - Distributed Tracing API
- **fluentd**
  - Logging
- **gRPC**
  - Remote Procedure Call
- **rkt**
  - Container Runtime
- **CNI**
  - Networking API
- **envoy**
  - Service Mesh
- **JAEG**
  - Distributed Tracing

**SANDBOX**
- **spiffe**
  - Identity Spec
- **SPIRE**
  - Identity
- **Open Policy Agent**
- **cloudevents**
- **Tooling**
- **HARBOR**
  - Registry
- **OPENMETRICS**
  - Metrics Spec
- **KV**
  - Distributed K/V
- **cortex**
  - Monitoring
- **Buildpacks.io**
  - Packaging Spec
- **Falco**
  - Container Security
Goal: to target non-CNCF projects

Implemented

ONAP
OPEN NETWORK AUTOMATION PLATFORM
Goal: to target public, bare metal & private clouds
<table>
<thead>
<tr>
<th>Project</th>
<th>Build</th>
<th>Release</th>
<th>Deployments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project</strong></td>
<td><strong>Build</strong></td>
<td><strong>Release</strong></td>
<td><strong>Deployments</strong></td>
</tr>
<tr>
<td>Kubernetes Orchestratio</td>
<td>SUCCESS</td>
<td>v1.11.2</td>
<td>SUCCESS</td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>03a6d0b</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>Prometheus Monitoring</td>
<td>SUCCESS</td>
<td>v2.3.2</td>
<td>SUCCESS</td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>3241c52</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>CoreDNS Service Discovery</td>
<td>SUCCESS</td>
<td>v1.2.2</td>
<td>FAILED</td>
</tr>
<tr>
<td></td>
<td>SUCCESS</td>
<td>ad43346</td>
<td>FAILED</td>
</tr>
</tbody>
</table>

Last updated 8 hours ago
Build Command

---

= Cross-Cloud CI Trigger Client"

---

*Quick start => type default_connect*

To use defaults and get a client run: default_connect
- @tc is the trigger client.
- @c is the ciservice client (also @tc.ciservice)
  => Use @c as outlined in docs/usage_from_irb.mkd

## Manual setup
To change the data store, set the @store_file variable to use a different store file (default: db/datastore-<CRCI_ENV>.yml)
Trigger client can be created with @tc = CrossCloudCi::TriggerClient.new({store_file: @store_file})
Ci service client is available as @tc.ciservice and @c
Set debugging level with @tc.logger.level and @tc.ciservice.logger.level

## Type trigger_help for more
Environment: prod
2.1.7 :001 > default_connect
I, [2018-10-19T14:35:08.332855 #21176] INFO -- : [CiService] Loading active projects
I, [2018-10-19T14:35:08.332985 #21176] INFO -- : [CiService] Loading GitLab project data
=> 0
2.1.7 :002 > build_projects
Project Configuration

```yaml
envoy:
  order: 6
  active: true
  logo_url: "https://raw.githubusercontent.com/cncf/artwork/ab42c9591f6e0fdccc62c7b88f353d3f8c825734/encov/icon/color/envoy-icon-color.png"
  display_name: Envoy
  sub_title: Service Mesh
  gitlab_name: envoy
  project_url: "https://github.com/envoyproxy/envoy"
  repository_url: "https://github.com/envoyproxy/envoy"
  timeout: 2100
  stable_ref: "v1.7.1"
  head_ref: "master"
  stable_chart: "stable"
  head_chart: "stable"
  app_layer: true
```

1 line less; before #1 21 seconds ago
Cloud Configuration

```yaml
- "container"
- "compile"

clouds:
  aws:
    active: true
    display_name: AWS
    order: 1
  azure:
    active: true
    display_name: Azure
    order: 2
  gce:
    active: true
    display_name: GCE
    order: 3
  gke:
    active: false
    display_name: GKE
    order: 4
  ibmcloud:
    active: true
    display_name: IBM Cloud
    order: 6
  packet:
    active: true
    display_name: Bare Metal (Packet)
    order: 8
```

[Integration] [cross-cloud.yml] [100%] [19:1]
Dashboard Update

- Kubernetes
  - Orchestration
  - Status: Running
  - Build: v1.11.2
  - Head: 03a6d0b

- Prometheus
  - Monitoring
  - Status: Running
  - Build: v2.3.2
  - Head: 3241c52

- CoreDNS
  - Service Discovery
  - Status: Running
  - Build: v1.2.2
  - Head: d3c2efc

- Fluentd
  - Logging
  - Status: Running
  - Build: v1.2.5
  - Head: f30865f

*Last updated a minute ago*
CI Dashboard Overview:

- Overview of cncf.ci
  - Shows status of 3 pipeline stages: Build, Provision and App Deployments
  - Refreshes at 3:00am Eastern Time every day
  - Supports dynamically adding/removing active clouds and projects
  - Clicking on Build status badge opens CI system build job URL
  - Clicking on Release Name opens project’s GitHub commit URL
  - Clicking on Deployment status badge opens “provisioning/app-deploy” job URL
Testing System Overview

- **Build Pipeline per project** (optional, can use project’s build artifacts)
- **Cloud Provisioning Pipeline** (cross-cloud)
- **App Deployment Pipeline** (cross-project)
1. **Build**: Compile binaries and e2e tests

2. **Package**: Create containers, create artifact pinning config and push to registry

3. **Update-Dashboard**: Update build status badges
1. **Build**: Prepare provisioning software from the cross-cloud project

2. **Artifacts**: Collect K8s artifact pinnings from the previous K8s builds

3. **Cross-Cloud**: Deploy K8s onto each cloud using cross-cloud provisioner

4. **Update-Dashboard**: Update deployment badges
App Deployment Pipeline Stage

1. **Artifacts:** Collect project artifact pinnings from the previous build stages
2. **Cross-Project:** Use Helm charts to deploy each project
3. **End-to-End:** Run e2e tests for each project
4. **Dashboard-Update:** Update deployment status badges
● **Unified CI/CD platform:** GitLab

● **Cross-cloud provisioning:** Terraform, Cloud-init, and per cloud K8s configuration

● **App deployments:** K8s manifest management with Helm

● **E2e tests:** Custom containers + Helm

● **Automated builds and deployments:** Git + per project yaml configuration
Dashboard Technology Overview

- **Frontend**: Vue.js
- **Status repository**: Elixir and Erlang
- **Automated builds and deployments**: Git + per project yaml configuration
CNCF CI CI Platform started (Feb 28, 2017)

- Feb 1st, 2017: CNCF CI Goals posted to mailing list
- Feb 28, 2017: CI Platform started
CI Dashboard Greenlight (Sept 22nd, 2017)

- Greenlight to start implementation of CI Dashboard
- Granted during Open Source Summit in Los Angeles

Timeline:
- Feb 28: CI Platform started
- June 27: 1st demo of CI Platform
- Aug 13: CI Platform v.1.0.0 Release
- Sept 22: Greenlight for Dashboard
Cncf.ci v1.0.0 Release (Jan 26th, 2018)

- Cncf.ci v1.0.0 released to production
- Included the Dashboard, Status API server, 4 active projects and 5 active clouds
- Projects: K8s, Prometheus, CoreDNS, Linkerd
- Clouds: AWS, Azure, GCE, GKE, Packet

Timeline:
- Feb 28: CI Platform started
- June 27: 1st demo of CI Platform
- Aug 13: CI Platform v.1.0.0 Release
- Sept 22: Greenlight for Dashboard
- Jan 26: Dashboard v.1.0.0 Release
Cncf.ci v1.1+ Releases (March, 2018)

• Cncf.ci v1.1.0 released
  ○ Added Fluentd and IBM Cloud
• Cncf.ci v1.2.0 released
  ○ Added ONAP SO project
• Cncf.ci v1.3.0 released
  ○ Added OpenStack
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb 28</td>
<td>CI Platform started</td>
<td></td>
</tr>
<tr>
<td>June 27</td>
<td>1st demo of CI Platform</td>
<td></td>
</tr>
<tr>
<td>Aug 13</td>
<td>CI Platform v.1.0.0 Release</td>
<td></td>
</tr>
<tr>
<td>Sept 22</td>
<td>Greenlight for Dashboard</td>
<td></td>
</tr>
<tr>
<td>Jan 26</td>
<td>Dashboard v1.0.0 Release</td>
<td></td>
</tr>
<tr>
<td>Mar 20</td>
<td>Dashboard v1.1.+ Releases</td>
<td></td>
</tr>
<tr>
<td>July 7</td>
<td>Dashboard v1.4.0 Release</td>
<td></td>
</tr>
</tbody>
</table>

- Cncf.ci v1.4.0 released to production
- Added VMware vSphere to providers
Cncf.ci v1.5.0 Release (Sept 7, 2018)

- Cncf.ci v1.5.0 released to production
- Added Envoy to projects
What’s next for Cross-cloud CI?

Adding New CNCF Projects:

- Jaeger
- Notary
- Vitess
- etc
What’s next for Cross-cloud CI?

Adding New Features:

- Automate project release updates
- API for history of builds, deployments and end-to-end tests
- Rollback to previous working release
- New screens TBD: Per Project, Per Deployment, etc
Cross-Group Collaboration:

- OpenCI Community
- OPNfv collaboration
- NSM collaboration
Upcoming Events

- **CI WG Public Meetings**
  - **Day:** 4th Tuesday of month
  - **Time:** 11:00am Pacific Time (US and Canada)
  - **Location:** https://zoom.us

- **Intro** and **Deep Dive** at KubeCon North America 2018
  - **Days:** Tuesday, December 11 and Wednesday, December 12
  - **Location:** Seattle, WA USA
How to Collaborate

- **Attend CI WG meetings:**
  - [https://github.com/cncf/wg-ci](https://github.com/cncf/wg-ci)

- **Subscribe to the CNCF CI public mailing list:**
  - [https://lists.cncf.io/g/cncf-ci-public](https://lists.cncf.io/g/cncf-ci-public)

- **Create issues on GitHub:**
  - [https://github.com/crosscloudci/cross-cloud/issues](https://github.com/crosscloudci/cross-cloud/issues)
Connect with the team

@crosscloudci
@crosscloudci
@vulkcoop

crosscloudci@vulk.coop
Thank you

Today’s Demo Prepared by:

denver@debian.nz
watson@vulk.coop
lucina@vulk.coop
taylor@vulk.coop