Kubernetes Conformance - Shanghai (Introduction)

Hosted by the CNCF.

Aaron Crickenberger (Google), Kevin Wang (Huawei)
Goals of Conformance

Consistency, Portability:
End users are able to describe workloads in a common format, and run them everywhere.

Predictability, Behave as expected:
End users can rely on a high level of common functionality.
Certification Program:
We all win when user expectations are met. Establish and maintain a Conformance Certification program to drive this.

Improve Conformance Coverage:
Ensure that Conformance covers enough functionality to be meaningful to users.
Status on Conformance Work

**v1.11.0**
4 new test cases

**v1.12.0**
Conformance test requirements
API Coverage via audit logging
22 new test cases

**v1.13**
Code coverage of conformance tests
Upstream kube-conformance image
35 new test cases
Status on Conformance Work

### # of Conformance Test Cases

<table>
<thead>
<tr>
<th>Version</th>
<th>Test Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.9.11</td>
<td>120</td>
</tr>
<tr>
<td>v1.10.9</td>
<td>130</td>
</tr>
<tr>
<td>v1.11.3</td>
<td>140</td>
</tr>
<tr>
<td>v1.12.0</td>
<td>150</td>
</tr>
<tr>
<td>v1.13.0-alpha.3</td>
<td>190</td>
</tr>
</tbody>
</table>
Status on Conformance Work

# of Test Cases

<table>
<thead>
<tr>
<th>Version</th>
<th>Conformance</th>
<th>Release e2e</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.9.11</td>
<td>150</td>
<td>350</td>
</tr>
<tr>
<td>v1.10.9</td>
<td>120</td>
<td>320</td>
</tr>
<tr>
<td>v1.11.3</td>
<td>140</td>
<td>360</td>
</tr>
<tr>
<td>v1.12.0</td>
<td>180</td>
<td>500</td>
</tr>
<tr>
<td>v1.13.0-alpha.3</td>
<td>190</td>
<td>550</td>
</tr>
</tbody>
</table>
Status on Conformance Work

Covered Endpoints

- Conformance (all)
- Release e2e (all)
- Conformance (stable)
- Release e2e (stable)
- Conformance (stable/core)
- Release e2e (stable/core)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.9.11</td>
<td>300</td>
<td>250</td>
<td>200</td>
<td>150</td>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>v1.10.9</td>
<td>350</td>
<td>300</td>
<td>250</td>
<td>200</td>
<td>150</td>
<td>100</td>
</tr>
<tr>
<td>v1.11.3</td>
<td>400</td>
<td>350</td>
<td>300</td>
<td>250</td>
<td>200</td>
<td>150</td>
</tr>
<tr>
<td>v1.12.0</td>
<td>450</td>
<td>400</td>
<td>350</td>
<td>300</td>
<td>250</td>
<td>200</td>
</tr>
<tr>
<td>v1.13.0-alpha.3</td>
<td>500</td>
<td>450</td>
<td>400</td>
<td>350</td>
<td>300</td>
<td>250</td>
</tr>
</tbody>
</table>
Certification Program

Certified Kubernetes is a software conformance program.

Vendors can certify with CNCF that their Kubernetes products conform to the community test suite.

Launched in Nov 2017, by now more than 70 platforms have been certified.
Participant Requirements

1. Determine goods and services
2. Self-testing for Qualifying Offerings
3. Submit signed Participation Form to LF
4. Submit self-testing results and get accepted
   a. Send pull request to github.com/cncf/k8s-conformance
5. Pay fees (no charge for CNCF members and non-profit organizations)
6. Abide by the terms and conditions
Key Conditions

- **Eligible Versions**
  - Latest two minor releases
  - E.g. k8s v1.12.2, v1.11.4

- **Conformance time Period**
  - Remains valid up to 12 months after release version that certified
  - Platforms must complete recertification each year with latest two minor versions
    E.g. A company “Acme” initially certified its product with v1.7.0 (released on June 30, 2017).
    On June 30, 2018, latest minor releases are v1.11.0 and v1.10.5
    On or before June 30, 2018, company “Acme” needs to recertify with v1.11.0 or v1.10.5 (or v1.9.9 before v1.11.0 released)

- **End user reproducibility**
  - End users can run self-tests on a Qualifying Offering themselves

- **More details**
  - Refer to: Terms and Conditions
Conformance FAQ

● What’s the cost of certification?
  ○ CNCF members: free of charge
  ○ Commercial organizations (not a CNCF member): same as joining the CNCF
  ○ Non-profit organizations: free of charge

● How to run self-tests?
  ○ With kubetest
  ○ With sonobuoy
# ensure kubetest is installed
```
go get -u k8s.io/test-infra/kubetest
```

# build test binaries, ginkgo, and kubectl first:
```
make WHAT="test/e2e/e2e.test vendor/github.com/onsi/ginkgo/ginkgo cmd/kubectl"
```

# setup for conformance tests
```
export KUBECONFIG=/path/to/kubeconfig
export KUBERNETES_CONFORMANCE_TEST=y
```

# Option A: run all conformance tests serially
```
kubetest --provider=skeleton --test --test_args="--ginkgo.focus=[Conformance]"
```

# Option B: run parallel conformance tests first, then serial conformance tests serially
```
GINKGO_PARALLEL=y kubetest --provider=skeleton --test --test_args="--ginkgo.focus=[Conformance] --ginkgo.skip=[Serial]"
kubetest --provider=skeleton --test --test_args="--ginkgo.focus=[Serial].*[Conformance]"
```

More details: [Conformance Testing in Kubernetes](https://kubernetes.io/docs/conformance/testing/)

Running Conformance Tests with Sonobuoy

● Prerequisites
  ○ Have your k8s cluster ready
  ○ Have kubectl installed
  ○ An admin kubeconfig file, and KUBECONFIG environment variable set.

● Run sonobuoy CLI
  ○ go get -u -v github.com/heptio/sonobuoy
  ○ sonobuoy run

● Checkout results
  ○ Wait until `sonobuoy status` show the run as completed
  ○ Copy output to a local directory `sonobuoy retrieve {local-dir}`
Kubernetes Conformance Program

Guidelines to Conformance Test Development

Aaron Crickenberger
Github: spiffxp
Twitter: spiffxp

Kevin Wang
Github: kevin-wangzefeng
Twitter: kevinwzf0126

CNCF Conformance WG
- Mailing List
- Slack: slack.k8s.io #k8s-conformance
- 4th Wednesday of every month at 11 AM PT
- Agenda Doc