When I say “We” I mean “We the Kubernetes Community”
Cross the Chasm

Primordial soup of container orchestration implementations
Achieve ubiquity to become a de-facto standard
Prioritize moving fast, intentionally take on technical debt
- One giant repo, development practices that aren’t friendly to newcomers
- Tie directly to implementations, eg: etcd, docker
- Avoid clean interfaces, eg: cloud provider code
Define extension points, extract functionality
  - The “core”/“in-tree” functionality should use the same interfaces as everyone else
  - We now have CNI, CRI, CSI
  - We are extracting cloud provider functionality out

Focus on stability and reliability
Focus on portability of workloads
Keep it Boring

Cluster Operators: Is my cluster a Certified Kubernetes cluster?
Application Developers: Will my workload actually be portable?
Kubernetes Developers: Is this feature ready to be considered GA?

CNCF Kubernetes Certification Program

Source: https://github.com/cncf/k8s-conformance
Keep it Boring
What is Kubernetes

Source: https://github.com/kubernetes/community/tree/master/icons
What is Kubernetes

Usage Example

Exposed Pod with 3 replicas

Source: https://github.com/kubernetes/community/tree/master/icons
What is Kubernetes

It turns out we don’t have a spec or written standard
We care about behaviors visible to the end user
That work on any given conformant Kubernetes cluster
What is Kubernetes

Source: https://github.com/kubernetes/community/blob/master/contributors/devel/architectural-roadmap.md
What is Kubernetes

Ecosystem

Interface Layer: Client Libraries and Tools

Governance Layer: Automation and Policy Enforcement

Application Layer: Deployment and Routing

Nucleus: API and Execution

Container Runtime

Network Plugin

Volume Plugin

Image Registry

Cloud Provider

Identity Provider

Source: https://github.com/kubernetes/community/blob/master/contributors/devel/architectural-roadmap.md
Improving Conformance

“OK, we use it, but it barely covers any functionality”
- What does that mean? Is that true?
Let’s add tests to exercise more functionality
- Focus on “Pod” functionality
- Ensure the definition of conformance is upstream
Let’s measure proxies for “functionality” coverage
- What API endpoints are covered when the tests are run?
- What code is covered when the tests are run?
Add Tests

Write an e2e test
Document the e2e test

Demonstrate that it meets these requirements*
- Tests only GA, non-optional features or APIs
- Works for all providers
- Is non-privileged
- Works without public internet access
- Binaries used are required for Linux kernel or kubelet to run
- Images used support all architectures for which Kubernetes releases are built
- Passes against versions of Kubernetes consistent with version skew policy
- Provides consistent results without flakes

Propose to SIG Architecture that the e2e test be promoted to Conformance
Add Tests

# of Test Cases

Source: github.com/spiffxp/adventures-in-k8s-conformance
Add Tests - 1.9 to 1.10

[sig-api-machinery] Garbage collector should delete RS created by deployment when not orphaning [Conformance]
[sig-api-machinery] Garbage collector should delete pods created by rc when not orphaning [Conformance]
[sig-api-machinery] Garbage collector should keep the rc around until all its pods are deleted if the deleteOptions says so [Conformance]
[sig-api-machinery] Garbage collector should not be blocked by dependency circle [Conformance]
[sig-api-machinery] Garbage collector should not delete dependents that have both valid owner and owner that's waiting for dependents to be deleted [Conformance]
[sig-api-machinery] Garbage collector should orphan RS created by deployment when deleteOptions.PropagationPolicy is Orphan [Conformance]
[sig-api-machinery] Garbage collector should orphan pods created by rc if delete options say so [Conformance]
[sig-apps] Daemon set [Serial] should retry creating failed daemon pods [Conformance]
[sig-apps] Daemon set [Serial] should rollback without unnecessary restarts [Conformance]
[sig-apps] Daemon set [Serial] should run and stop complex daemon [Conformance]
[sig-apps] Daemon set [Serial] should run and stop simple daemon [Conformance]
[sig-apps] Daemon set [Serial] should update pod when spec was updated and update strategy is RollingUpdate [Conformance]
[sig-apps] StatefulSet [k8s.io] Basic StatefulSet functionality [StatefulSetBasic] Burst scaling should run to completion even with unhealthy pods [Conformance]
[sig-apps] StatefulSet [k8s.io] Basic StatefulSet functionality [StatefulSetBasic] Scaling should happen in predictable order and halt if any stateful pod is unhealthy [Conformance]
[sig-apps] StatefulSet [k8s.io] Basic StatefulSet functionality [StatefulSetBasic] Should recreate evicted statefulset [Conformance]
[sig-apps] StatefulSet [k8s.io] Basic StatefulSet functionality [StatefulSetBasic] should perform canary updates and phased rolling updates of template modifications [Conformance]
[sig-apps] StatefulSet [k8s.io] Basic StatefulSet functionality [StatefulSetBasic] should perform rolling updates and roll backs of template modifications [Conformance]
Add Tests - 1.10 to 1.11

[sig-api-machinery] Watchers should be able to restart watching from the last resource version observed by the previous watch [Conformance]
[sig-api-machinery] Watchers should be able to start watching from a specific resource version [Conformance]
[sig-api-machinery] Watchers should observe add, update, and delete watch notifications on configmaps [Conformance]
[sig-api-machinery] Watchers should observe an object deletion if it stops meeting the requirements of the selector [Conformance]
Add Tests - 1.11 to 1.12

- Container Lifecycle Hook when create a pod with lifecycle hook should execute poststart exec hook properly [NodeConformance] [Conformance]
- Container Lifecycle Hook when create a pod with lifecycle hook should execute poststart http hook properly [NodeConformance] [Conformance]
- Container Lifecycle Hook when create a pod with lifecycle hook should execute prestop exec hook properly [NodeConformance] [Conformance]
- Container Lifecycle Hook when create a pod with lifecycle hook should execute prestop http hook properly [NodeConformance] [Conformance]
- InitContainer [NodeConformance] should invoke init containers on a RestartAlways pod [Conformance]
- InitContainer [NodeConformance] should invoke init containers on a RestartNever pod [Conformance]
- InitContainer [NodeConformance] should not start app containers and fail the pod if init containers fail on a RestartNever pod [Conformance]
- InitContainer [NodeConformance] should not start app containers if init containers fail on a RestartAlways pod [Conformance]
- Namespaces [Serial] should ensure that all pods are removed when a namespace is deleted [Conformance]
- Namespaces [Serial] should ensure that all services are removed when a namespace is deleted [Conformance]
- Deployment RecreateDeployment should delete old pods and create new ones [Conformance]
- Deployment RollingUpdateDeployment should delete old pods and create new ones [Conformance]
- Deployment deployment should delete old replica sets [Conformance]
- Deployment deployment should support proportional scaling [Conformance]
- Deployment deployment should support rollover [Conformance]
- ConfigMap binary data should be reflected in volume [NodeConformance] [Conformance]
- Secrets should be able to mount in a volume regardless of a different secret existing with same name in different namespace [NodeConformance] [Conformance]
- Subpath Atomic writer volumes should support subpaths with configmap pod [Conformance]
- Subpath Atomic writer volumes should support subpaths with configmap pod with mountPath of existing file [Conformance]
- Subpath Atomic writer volumes should support subpaths with downward pod [Conformance]
- Subpath Atomic writer volumes should support subpaths with projected pod [Conformance]
- Subpath Atomic writer volumes should support subpaths with secret pod [Conformance]
Add Tests - 1.12 to 1.13

Container Runtime blackbox test when starting a container that exits should run with the expected status [NodeConformance] [Conformance]

Kubelet when scheduling a busybox Pod with hostAliases should write entries to /etc/hosts [NodeConformance] [Conformance]

Kubelet when scheduling a busybox command that always fails in a pod should have an terminated reason [NodeConformance] [Conformance]

Kubelet when scheduling a busybox container that should not write to root filesystem [NodeConformance] [Conformance]

Pods should support remote command execution over websockets [NodeConformance] [Conformance]

Kubelet when scheduling a busybox command in a pod should print the output to logs [NodeConformance] [Conformance]

Kubelet when scheduling a busybox command in a pod that exits should run with the expected status [NodeConformance] [Conformance]

Kubelet when scheduling a busybox command in a pod that always fails in a pod should be possible to delete [NodeConformance] [Conformance]

Kubelet when scheduling a read only busybox container should not write to root filesystem [NodeConformance] [Conformance]

Kubelet when scheduling a busybox command in a pod should print the output to logs [NodeConformance] [Conformance]

Kubelet when scheduling a busybox command in a pod should have an terminated reason [NodeConformance] [Conformance]

Kubelet when scheduling a busybox command in a pod should have an terminated reason [NodeConformance] [Conformance]

Kubelet when scheduling a busybox command in a pod should have an terminated reason [NodeConformance] [Conformance]

Kubelet when scheduling a busybox command in a pod should have an terminated reason [NodeConformance] [Conformance]

Pods should support retrieving logs from the container over websockets [NodeConformance] [Conformance]

ReplicaSet should adopt matching pods on creation and release no longer matching pods [NodeConformance] [Conformance]

ReplicationController should adopt matching pods on creation [Conformance]

ReplicationController should release no longer matching pods [Conformance]

Kubectl api-versions should check if v1 is in available api versions [Conformance]

Kubectl describe should check if kubectl describe prints relevant information for rc and pods [Conformance]

Kubectl expose should create services for rc [Conformance]

Kubectl label should update the label on a resource [Conformance]

Kubectl logs should be able to retrieve and filter logs [Conformance]

Kubectl patch should add annotations for pods in rc [Conformance]

Kubectl describe should check if kubectl describe prints relevant information for rc and pods [Conformance]

Kubectl replace should update a single-container pod&hashbang39's image [Conformance]

Kubectl rolling-update should support rolling-update to same image [Conformance]

Kubectl run --rm job should create a job from an image, then delete the job [Conformance]

Kubectl run deployment should create a deployment from an image [Conformance]

Kubectl run deployment should create a deployment from an image [Conformance]

Kubectl run job should create a job from an image when restart is OnFailure [Conformance]

Kubectl run pod should create a pod from an image when restart is Never [Conformance]

Kubectl run pod should create a pod from an image when restart is Never [Conformance]

Kubectl run rc should create an rc from an image [Conformance]

Kubectl version should check is all data is printed [Conformance]

Proxy server should support --unix-socket=host [Conformance]

Proxy server should support proxy with --port 0 [Conformance]

Update Demo should create and stop a replication controller [Conformance]

Update Demo should do a rolling update of a replication controller [Conformance]

Update Demo should scale a replication controller [Conformance]

EmptyDir wrapper volumes should not cause race condition when used for configmaps [Serial] [Slow] [Conformance]

EmptyDir wrapper volumes should not conflict [Conformance]
Add Tests

# 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></td>
<td></td>
</tr>
<tr>
<td>v1.10.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v1.11.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v1.12.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v1.13.0-alpha.3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: github.com/spiffxp/adventures-in-k8s-conformance
Add Tests - Next

Write more tests
Focus on Pod functionality
- Probes
- Storage
- Connectivity
- Pod lifecycle
- etc.

Workload APIs exercise this indirectly, we need more direct
Requirements - Next

Identify whether existing tests fit these requirements
- Tag functional requirements, e.g. [Privileged]
- Tag behavioral problem, e.g. [Flaky]

Identify further constraints or requirements
- Multiple nodes
- Mixed node clusters

Develop Profiles
- OS-specific functionality
- Optional features such as cloud provider
Getting Data

To manually reproduce

To reproduce on your own setup, you'll need to:

- ensure gcloud is configured
- ensure you have a GCP project setup (your-gcp-project below)
- ensure you have keys to a GCP service account (/etc/service-account/service-account.json below)
- ensure you have GNU sed installed
- ensure go is properly setup
- go get k8s.io/test-infra/kubetest
- clone kubernetes/kubernetes, and then

```bash
$ kubetest
   --dump=./artifacts
   --gcp-service-account=/etc/service-account/service-account.json
   --up
   --down
   --test
   --provider=gce
   --cluster-bootstrap-e2e
   --gcp-network-bootstrap-e2e
   --extract=release/stable-1.12
   --gcp-master-tempest
   --gcp-node-tempest
   --gcp-zone-us-central1-f
   --gcp-project-your-gcp-project
   --test-args=ginkgo.focus=\[\{Conformance\}\]
   --timeout=20m | tee ./build-log.txt
```

To convert build-log.txt (either manually generated, or downloaded from our CI) to the e2e.log format expected by CNCF reviewers, run the following command, which will:

- remove color codes
- strip prepended timestamps
- truncate

```bash
$ convert build-log.txt
d --c 's/\x1B\[([0-9;]*m/''
  --c 's/\[([0-9;]*\[0-9;]*z\[0-9;]*\[[0-9;]*\[[0-9;]*\[[0-9;]*\[[0-9;]*\]]\]''
  --c '/SUCCESS/g''
```

Source: https://github.com/cncf/k8s-conformance/tree/master/v1.12/kube-up-gce
# Getting Data

## Summary

<table>
<thead>
<tr>
<th>Source: <a href="https://testgrid.k8s.io/conformance-all">https://testgrid.k8s.io/conformance-all</a></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Test Name</th>
<th># Fails</th>
<th>Failure Messages</th>
<th>First Failed</th>
<th>Last Passed</th>
<th>Bugs</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract</td>
<td>29</td>
<td>11-07-15:08</td>
<td>Overall</td>
<td>29</td>
<td>11-07-15:08</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCE, v1.13 (dev)</td>
<td>PASSING</td>
<td>All passing in the past week</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCE, v1.12 (release)</td>
<td>PASSING</td>
<td>All passing in the past week</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCE, v1.12 (dev)</td>
<td>PASSING</td>
<td>All passing in the past week</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCE, v1.11 (release)</td>
<td>PASSING</td>
<td>All passing in the past week</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCE, v1.11 (dev)</td>
<td>PASSING</td>
<td>All passing in the past week</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCE, v1.10 (release)</td>
<td>PASSING</td>
<td>All passing in the past week</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCE, v1.10 (dev)</td>
<td>PASSING</td>
<td>All passing in the past week</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GCE, v1.9 (dev)</td>
<td>PASSING</td>
<td>3 of 4200 tests (0.1%) and 1 of 28 runs (3.6%) failing in the past week</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kind, master (dev)</td>
<td>PASSING</td>
<td>6 of 19153 tests (0.0%) and 5 of 107 runs (4.7%) failing in the past week</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kind, master (dev)</td>
<td>(non-serial)</td>
<td>NO results in the past week</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kind, v1.12 (dev)</td>
<td>Failing</td>
<td>9 of 19156 tests (0.0%) and 6 of 118 runs (5.1%) failing in the past week</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>kind, v1.11 (dev)</td>
<td>PASSING</td>
<td>4 of 17220 tests (0.0%) and 4 of 123 runs (3.3%) failing in the past week</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Last green run: 870 tests last run on: 11-14-18:26
Last failure: 11-14-18:28
No passing build found
Tests last run on: 11-14-15:34
Last update: 11-14-15:32

Source: https://testgrid.k8s.io/conformance-all
# Getting Data

## ci-kubernetes-gce-conformance-latest-1-13 #27

**Recent run**

<table>
<thead>
<tr>
<th>Result</th>
<th>SUCCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tests</td>
<td>0 failed / 214 succeeded</td>
</tr>
<tr>
<td>Started</td>
<td>2018-11-14 19:32 CST</td>
</tr>
<tr>
<td>Elapsed</td>
<td>1h49m</td>
</tr>
<tr>
<td>Version</td>
<td>v1.13.0-beta.0.267+20b5777d454e56</td>
</tr>
<tr>
<td>Builder</td>
<td>gke-prov-default-pool-3cb694a8-b9a5</td>
</tr>
<tr>
<td>pod</td>
<td>e45515d5-1ae0-11e8-bc35-0a5b0a6c0111</td>
</tr>
<tr>
<td>infra-commit</td>
<td>e4c3f3e675</td>
</tr>
<tr>
<td>job-version</td>
<td>v1.13.0-beta.0.267+20b5777d454e56</td>
</tr>
<tr>
<td>master_os_image</td>
<td>cos-stable-65-10323-64-0</td>
</tr>
<tr>
<td>node_os_image</td>
<td>cos-stable-65-10323-64-0</td>
</tr>
<tr>
<td>pod</td>
<td>e45515d5-1ae0-11e8-bc35-0a5b0a6c0111</td>
</tr>
<tr>
<td>version</td>
<td>v1.13.0-beta.0.267+20b5777d454e56</td>
</tr>
</tbody>
</table>

**No Test Failures!**

- Show 214 Passed Tests
- Show 1700 Skipped Tests

# Getting Data

---

## kubernetes-jenkins

`/kubernetes-jenkins/logs/ci-kubernetes-gce-conformance-latest-1-13/27/artifacts/`

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>bootstraps-e2e-master/</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>bootstraps-e2e-minion-group-0k91/</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>bootstraps-e2e-minion-group-03l/</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>bootstraps-e2e-minion-group-05b/</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>junit_01.xml</td>
<td>41580</td>
<td>14 Nov 2018 13:22:17</td>
</tr>
<tr>
<td>metadata.json</td>
<td>186</td>
<td>14 Nov 2018 13:22:17</td>
</tr>
<tr>
<td>nodes.yaml</td>
<td>20633</td>
<td>14 Nov 2018 13:22:17</td>
</tr>
</tbody>
</table>

---

## kubernetes-jenkins


<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>docker.log</td>
<td>343907</td>
<td>14 Nov 2018 13:22:18</td>
</tr>
<tr>
<td>etcd-events.log</td>
<td>2227</td>
<td>14 Nov 2018 13:22:18</td>
</tr>
<tr>
<td>etcd.log</td>
<td>3245</td>
<td>14 Nov 2018 13:22:18</td>
</tr>
<tr>
<td>gcslog</td>
<td>2708</td>
<td>14 Nov 2018 13:22:18</td>
</tr>
<tr>
<td>kerr.log</td>
<td>10311</td>
<td>14 Nov 2018 13:22:18</td>
</tr>
<tr>
<td>kube-apiserver.log</td>
<td>300074</td>
<td>14 Nov 2018 13:22:18</td>
</tr>
<tr>
<td>kube-master-installation.log</td>
<td>1130</td>
<td>14 Nov 2018 13:22:18</td>
</tr>
<tr>
<td>kube-scheduler.log</td>
<td>43992</td>
<td>14 Nov 2018 13:22:18</td>
</tr>
<tr>
<td>kubeltl-monitor.log</td>
<td>121</td>
<td>14 Nov 2018 13:22:18</td>
</tr>
<tr>
<td>kubeltl.log</td>
<td>1375721</td>
<td>14 Nov 2018 13:22:18</td>
</tr>
<tr>
<td>serial-1.log</td>
<td>11858</td>
<td>14 Nov 2018 13:22:18</td>
</tr>
</tbody>
</table>

---

API Coverage

Use `cncf/apisnoop`

- Parse Kubernetes’ OpenAPI spec to find possible endpoints
  - Endpoint = VERB path
  - e.g. POST /api/v1/namespaces/{namespace}/pods
- Parse Kubernetes audit log to find endpoint hits

Questions we can answer:

- What are the total endpoints that can be hit?
- What are the actual endpoints that were hit?
API Coverage - by %

API Coverage % by Release

Source: github.com/spiffxp/adventures-in-k8s-conformance
API Coverage - by ep

Covered Endpoints by Release

Conformance

Release e2e

Total

Conformance (stable)

Release e2e (stable)

Total (stable)

Source: github.com/spiffxp/adventures-in-k8s-conformance
API Coverage - covered

Source: github.com/spiffxp/adventures-in-k8s-conformance
This was the first release we could actually measure API coverage because audit logging was turned on by default.
API Coverage - graph

v1.10.9 conformance

Coverage increased a little bit by adding the conformance tests that we did

25.7%
301/1172 total tested

Source: apisnoop.cncf.io
API Coverage

v1.11.3 conformance

Progress stalled a bit

Source: apisnoop.cncf.io
API Coverage

v1.12.0 conformance

Biggest impact here in terms of API coverage was removing deprecated API endpoints from our measurement.

Second big step was promoting a number of tests formerly used for NodeConformance to Conformance in general.

This is the first release where user agent was added to the audit log, so we could parse out which coverage was coming directly from the e2e.test binary.

34.4%
317/921 total tested

Source: apisnoop.cncf.io
API Coverage

v1.13.0 conformance

Thus far we have promoted a number of tests related to the use of kubectl, replication controllers and replicasets.
API Coverage

v1.13.0 conformance (e2e.test only)

If we are looking solely at API coverage caused by the e2e.test binary, it looks as though we have a very long way to go.

This is why we decided measuring api coverage client side wasn’t an effective representation of behavior exercised within kubernetes.

Source: apisnoop.cncf.io
v1.13.0 release

If every potentially qualifying e2e test we had today was promoted to conformance, this is how far we would get in terms of API coverage.

We need to be writing new test cases in order to get the rest of the way there.
API Coverage

Add user agent as field in Kubernetes’ audit log
- For each e2e test, set user agent to test name
- Parse Kubernetes audit log to find endpoint hits for each agent

Questions we can answer:
- What are the endpoints a given test hits?
- Which endpoints are hit by the most tests?
- Which endpoints are hit by only a few tests?
API Coverage

# Test cases by endpoint

Source: github.com/spiffxp/adventures-in-k8s-conformance
[sig-network] Proxy version v1 should proxy through a service and a pod [Conformance]
API Coverage

2018-11-06T14:42:06.679550Z GET /apis/v1?timeout=32s
2018-11-06T14:42:06.679550Z GET /apis/v1beta1?timeout=32s
2018-11-06T14:42:06.679550Z GET /apis/v1beta2?timeout=32s
2018-11-06T14:42:06.681464Z GET /apis/apiextensions.k8s.io/v1beta1?timeout=32s
2018-11-06T14:42:06.681464Z GET /apis/apiextensions.k8s.io/v1beta2?timeout=32s
2018-11-06T14:42:06.682795Z GET /apis/batch/v1?timeout=32s
2018-11-06T14:42:06.68305Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.684231Z GET /apis/batch/v1?timeout=32s
2018-11-06T14:42:06.684231Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.685915Z GET /apis/batch/v1beta2?timeout=32s
2018-11-06T14:42:06.688227Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.689808Z GET /apis/batch/v1?timeout=32s
2018-11-06T14:42:06.689808Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.691675Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.691675Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.692827Z GET /apis/batch/v1beta2?timeout=32s
2018-11-06T14:42:06.692827Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.699445Z GET /apis/batch/v2beta1?timeout=32s
2018-11-06T14:42:06.699445Z GET /apis/batch/v2beta2?timeout=32s
2018-11-06T14:42:06.702736Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.702736Z GET /apis/batch/v1beta2?timeout=32s
2018-11-06T14:42:06.704111Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.704111Z GET /apis/batch/v1beta2?timeout=32s
2018-11-06T14:42:06.705776Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.705776Z GET /apis/batch/v2alpha2?timeout=32s
2018-11-06T14:42:06.706999Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.706999Z GET /apis/batch/v1beta2?timeout=32s
2018-11-06T14:42:06.708305Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.708305Z GET /apis/batch/v2alpha2?timeout=32s
2018-11-06T14:42:06.710833Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.710833Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.712122Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.712122Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.71477Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.71477Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.71545Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.71545Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.716833Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.716833Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.718159Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.718159Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.720108Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.720108Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.721520Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.721520Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.722741Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.722741Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.723921Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.723921Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.725073Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.725073Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.727031Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.727031Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.730169Z CREATE /api/v1/namespaces
2018-11-06T14:42:06.736030Z WATCH /api/v1/namespaces/e2e-tests-proxy-gvqt4/serviceaccounts?fieldSelector=metadata.name%3Ddefault&watch=true
2018-11-06T14:42:06.736030Z WATCH /api/v1/namespaces/e2e-tests-proxy-gvqt4/serviceaccounts?fieldSelector=metadata.name%3Ddefault&watch=true
2018-11-06T14:42:06.763429Z WATCH /api/v1/namespaces/e2e-tests-proxy-gvqt4/serviceaccounts?fieldSelector=metadata.name%3Ddefault&watch=true
2018-11-06T14:42:06.763429Z WATCH /api/v1/namespaces/e2e-tests-proxy-gvqt4/serviceaccounts?fieldSelector=metadata.name%3Ddefault&watch=true
2018-11-06T14:42:06.765396Z CREATE /api/v1/namespaces/e2e-tests-proxy-gvqt4/services
2018-11-06T14:42:06.778053Z CREATE /api/v1/namespaces/e2e-tests-proxy-gvqt4/replicationcontrollers
2018-11-06T14:42:06.788409Z LIST /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods?labelSelector=name%3Dproxy-service-8t7gj&limit=500&resourceVersion=0
2018-11-06T14:42:06.793158Z WATCH /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods?labelSelector=name%3Dproxy-service-8t7gj&resourceVersion=11546&timeoutSeconds=310&watch=true
2018-11-06T14:42:06.793158Z WATCH /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods?labelSelector=name%3Dproxy-service-8t7gj&resourceVersion=11546&timeoutSeconds=310&watch=true
...
API Coverage

2018-11-06T14:42:23.632108Z LIST /api/v1/nodes
2018-11-06T14:42:23.636469Z DELETE /api/v1/namespaces/e2e-tests-proxy-gvqt4
2018-11-06T14:42:23.640297Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4
2018-11-06T14:42:23.642977Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4
2018-11-06T14:42:25.642996Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4
2018-11-06T14:42:27.642977Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4
2018-11-06T14:42:29.643128Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4
2018-11-06T14:42:31.643193Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4
API Coverage

[sig-network] Proxy version v1 should proxy through a service and a pod  [Conformance]
- Only tests GET verb, ignores POST, PUT, PATCH, DELETE, HEAD, OPTIONS
- Doesn’t test proxy/with/subpath endpoints
- 13 endpoints we could cover
Static OpenAPI spec misses extensions registered at runtime, e.g.
- /apis/scalingpolicy.kope.io/v1alpha1
- /apis/metrics.k8s.io/v1beta1

Static audit logging policy limits environments we can inspect
- Dynamic audit policy will allow us to configure at runtime
- Can use webhooks to avoid requiring ssh access to hosts

Covering only verbs and paths limits coverage granularity
- What query parameters can be set?
- What fields are available to be set on resources
- What are valid / invalid values for fields of type F?
Line Coverage

Golang supports instrumentation for unit test line coverage
- Only gathers data for unit tests
- Only outputs coverage data when unit tests finish

What if Kubernetes was just a distributed system of unit tests?
- Generate unit tests for each kubernetes component
- Use a utility to flush coverage data to disk periodically

Deploy and e2e test Kubernetes as usual
- Gather coverage data before running e2e tests
- Gather coverage data after running e2e tests
Line Coverage

Manipulate coverage data using `gopherage`

```
gopherage merge before/kube*.cov >before.cov
gopherage merge after/kube*.cov >after.cov
gopherage diff before.cov after.cov >diff.cov
export ignore=zz_generated,third_party/,cmd/,cloudprovider/
gopherage filter --exclude-path=$ignore diff.cov >file.cov
```
Line Coverage

go tool cover -html=file.cov  # which lines were covered?
Line Coverage

gopherage html file.cov # which packages are covered?
# Line Coverage

## Coverage summary

<table>
<thead>
<tr>
<th>File</th>
<th>e2e</th>
<th>conformance</th>
</tr>
</thead>
<tbody>
<tr>
<td>k8s.io/k8s.io</td>
<td>33.5%</td>
<td>28.5%</td>
</tr>
<tr>
<td>k8s.io/kubernetes/pkg</td>
<td>37.6%</td>
<td>25.5%</td>
</tr>
<tr>
<td>k8s.io/kubernetes/plugin/pkg</td>
<td>32.6%</td>
<td>16.1%</td>
</tr>
<tr>
<td>k8s.io/vendor/k8s.io</td>
<td>31.0%</td>
<td>31.3%</td>
</tr>
</tbody>
</table>

## Coverage summary

### k8s.io/k8s.io/kubernetes/pkg

<table>
<thead>
<tr>
<th>File</th>
<th>e2e</th>
<th>conformance</th>
</tr>
</thead>
<tbody>
<tr>
<td>api</td>
<td>49.6%</td>
<td>46.5%</td>
</tr>
<tr>
<td>apiserver</td>
<td>47.4%</td>
<td>34.4%</td>
</tr>
<tr>
<td>auth</td>
<td>9.6%</td>
<td>9.8%</td>
</tr>
<tr>
<td>capabilities</td>
<td>36.4%</td>
<td>36.4%</td>
</tr>
<tr>
<td>client</td>
<td>2.2%</td>
<td>1.0%</td>
</tr>
<tr>
<td>controller</td>
<td>50.2%</td>
<td>31.5%</td>
</tr>
<tr>
<td>credentialprovider</td>
<td>28.6%</td>
<td>23.7%</td>
</tr>
<tr>
<td>features</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>fieldpath</td>
<td>51.4%</td>
<td>51.4%</td>
</tr>
<tr>
<td>kubeapiserver</td>
<td>3.8%</td>
<td>2.6%</td>
</tr>
<tr>
<td>kubectl</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>kubectl</td>
<td>43.0%</td>
<td>40.7%</td>
</tr>
<tr>
<td>master</td>
<td>16.4%</td>
<td>16.4%</td>
</tr>
<tr>
<td>printer</td>
<td>4.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>probe</td>
<td>61.2%</td>
<td>59.2%</td>
</tr>
<tr>
<td>proxy</td>
<td>18.4%</td>
<td>16.8%</td>
</tr>
<tr>
<td>quota</td>
<td>70.4%</td>
<td>7.3%</td>
</tr>
<tr>
<td>registry</td>
<td>39.0%</td>
<td>24.8%</td>
</tr>
<tr>
<td>routes</td>
<td>45.5%</td>
<td>0.0%</td>
</tr>
<tr>
<td>scheduler</td>
<td>41.1%</td>
<td>32.8%</td>
</tr>
<tr>
<td>security</td>
<td>55.3%</td>
<td>1.0%</td>
</tr>
<tr>
<td>securitycontext</td>
<td>51.2%</td>
<td>15.7%</td>
</tr>
<tr>
<td>serviceaccount</td>
<td>35.8%</td>
<td>34.1%</td>
</tr>
<tr>
<td>ssh</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>util</td>
<td>31.0%</td>
<td>20.5%</td>
</tr>
<tr>
<td>version</td>
<td>3.1%</td>
<td>3.1%</td>
</tr>
<tr>
<td>volume</td>
<td>21.6%</td>
<td>7.6%</td>
</tr>
</tbody>
</table>

### k8s.io/k8s.io/kubernetes/plugin/pkg

<table>
<thead>
<tr>
<th>File</th>
<th>e2e</th>
<th>conformance</th>
</tr>
</thead>
<tbody>
<tr>
<td>admission</td>
<td>32.2%</td>
<td>13.4%</td>
</tr>
<tr>
<td>auth</td>
<td>34.1%</td>
<td>28.1%</td>
</tr>
</tbody>
</table>

### k8s.io/k8s.io/vendor/k8s.io

<table>
<thead>
<tr>
<th>File</th>
<th>e2e</th>
<th>conformance</th>
</tr>
</thead>
<tbody>
<tr>
<td>api</td>
<td>31.2%</td>
<td>29.6%</td>
</tr>
<tr>
<td>apiregistration</td>
<td>36.0%</td>
<td>38.2%</td>
</tr>
<tr>
<td>apimachinery</td>
<td>34.8%</td>
<td>42.0%</td>
</tr>
<tr>
<td>apiserver</td>
<td>24.1%</td>
<td>27.3%</td>
</tr>
<tr>
<td>cli-runtime</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>client-go</td>
<td>25.3%</td>
<td>27.1%</td>
</tr>
<tr>
<td>cloud-provider</td>
<td>17.5%</td>
<td>8.8%</td>
</tr>
<tr>
<td>cluster-bootstrap</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>csi-api</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>kube-aggregator</td>
<td>49.2%</td>
<td>50.0%</td>
</tr>
<tr>
<td>kube-controller-manager</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>kube-openapi</td>
<td>33.6%</td>
<td>33.6%</td>
</tr>
<tr>
<td>kube-proxy</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>kube-scheduler</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>kubelet</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>metrics</td>
<td>6.2%</td>
<td>2.1%</td>
</tr>
<tr>
<td>util</td>
<td>55.6%</td>
<td>52.4%</td>
</tr>
</tbody>
</table>
# Line Coverage

## Coverage summary

```markdown
<table>
<thead>
<tr>
<th>File</th>
<th>325</th>
<th>315</th>
<th>305</th>
<th>295</th>
<th>285</th>
<th>275</th>
<th>265</th>
<th>255</th>
</tr>
</thead>
<tbody>
<tr>
<td>api/</td>
<td>48.5%</td>
<td>48.5%</td>
<td>48.7%</td>
<td>48.7%</td>
<td>48.7%</td>
<td>48.2%</td>
<td>48.6%</td>
<td>48.7%</td>
</tr>
<tr>
<td>apis/</td>
<td>33.8%</td>
<td>33.6%</td>
<td>33.7%</td>
<td>33.7%</td>
<td>33.7%</td>
<td>33.7%</td>
<td>33.7%</td>
<td>33.7%</td>
</tr>
<tr>
<td>auth/</td>
<td>9.8%</td>
<td>9.6%</td>
<td>9.6%</td>
<td>9.6%</td>
<td>9.6%</td>
<td>9.6%</td>
<td>9.6%</td>
<td>9.6%</td>
</tr>
<tr>
<td>capabilities/</td>
<td>30.4%</td>
<td>36.4%</td>
<td>36.4%</td>
<td>36.4%</td>
<td>36.4%</td>
<td>36.4%</td>
<td>36.4%</td>
<td>36.4%</td>
</tr>
<tr>
<td>client/</td>
<td>1.0%</td>
<td>1.0%</td>
<td>1.0%</td>
<td>0.6%</td>
<td>0.6%</td>
<td>0.6%</td>
<td>0.6%</td>
<td>0.6%</td>
</tr>
<tr>
<td>controller/</td>
<td>31.3%</td>
<td>31.3%</td>
<td>31.1%</td>
<td>31.2%</td>
<td>31.3%</td>
<td>31.2%</td>
<td>31.2%</td>
<td>31.2%</td>
</tr>
<tr>
<td>credentialprovider/</td>
<td>23.7%</td>
<td>23.7%</td>
<td>23.7%</td>
<td>23.7%</td>
<td>23.7%</td>
<td>23.7%</td>
<td>23.7%</td>
<td>23.7%</td>
</tr>
<tr>
<td>features/</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>fieldpath/</td>
<td>51.4%</td>
<td>51.4%</td>
<td>51.4%</td>
<td>51.4%</td>
<td>51.4%</td>
<td>51.4%</td>
<td>51.4%</td>
<td>51.4%</td>
</tr>
<tr>
<td>kubeprovider/</td>
<td>2.6%</td>
<td>2.6%</td>
<td>2.6%</td>
<td>2.6%</td>
<td>2.6%</td>
<td>▼ 2.6%</td>
<td>▼ 2.6%</td>
<td>▼ 2.6%</td>
</tr>
<tr>
<td>kubectl/</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>kubelet/</td>
<td>▼ 40.6%</td>
<td>▼ 40.7%</td>
<td>▼ 40.4%</td>
<td>▼ 40.5%</td>
<td>▼ 40.7%</td>
<td>▼ 40.6%</td>
<td>▼ 40.5%</td>
<td>▼ 40.7%</td>
</tr>
<tr>
<td>master/</td>
<td>16.4%</td>
<td>16.4%</td>
<td>▲ 16.4%</td>
<td>16.3%</td>
<td>16.3%</td>
<td>16.3%</td>
<td>16.3%</td>
<td>16.3%</td>
</tr>
<tr>
<td>printers/</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>probe/</td>
<td>59.2%</td>
<td>59.2%</td>
<td>59.2%</td>
<td>59.2%</td>
<td>59.2%</td>
<td>59.2%</td>
<td>59.2%</td>
<td>59.2%</td>
</tr>
<tr>
<td>proxy/</td>
<td>16.8%</td>
<td>16.8%</td>
<td>▲ 16.8%</td>
<td>16.9%</td>
<td>16.9%</td>
<td>16.9%</td>
<td>16.9%</td>
<td>16.9%</td>
</tr>
<tr>
<td>quota/</td>
<td>7.1%</td>
<td>7.1%</td>
<td>7.1%</td>
<td>7.1%</td>
<td>7.1%</td>
<td>7.1%</td>
<td>7.1%</td>
<td>7.1%</td>
</tr>
<tr>
<td>registry/</td>
<td>24.6%</td>
<td>24.6%</td>
<td>▲ 24.6%</td>
<td>▲ 24.7%</td>
<td>▲ 24.7%</td>
<td>▲ 24.6%</td>
<td>▲ 24.7%</td>
<td>▲ 24.6%</td>
</tr>
<tr>
<td>routes/</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>scheduler/</td>
<td>▼ 32.6%</td>
<td>▼ 31.9%</td>
<td>▼ 32.6%</td>
<td>▼ 32.8%</td>
<td>▼ 32.3%</td>
<td>▼ 32.5%</td>
<td>▼ 32.1%</td>
<td>▼ 32.3%</td>
</tr>
<tr>
<td>security/</td>
<td>1.0%</td>
<td>1.0%</td>
<td>1.0%</td>
<td>1.0%</td>
<td>1.0%</td>
<td>1.0%</td>
<td>1.0%</td>
<td>1.0%</td>
</tr>
<tr>
<td>securitycontext/</td>
<td>15.7%</td>
<td>15.7%</td>
<td>15.7%</td>
<td>15.7%</td>
<td>15.7%</td>
<td>15.7%</td>
<td>15.7%</td>
<td>15.7%</td>
</tr>
<tr>
<td>serviceaccount/</td>
<td>34.1%</td>
<td>34.1%</td>
<td>34.1%</td>
<td>34.1%</td>
<td>34.1%</td>
<td>34.1%</td>
<td>34.1%</td>
<td>34.1%</td>
</tr>
<tr>
<td>ssh/</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>util/</td>
<td>20.4%</td>
<td>▼ 20.4%</td>
<td>▼ 20.4%</td>
<td>20.6%</td>
<td>▼ 20.6%</td>
<td>20.4%</td>
<td>▼ 20.4%</td>
<td>20.4%</td>
</tr>
<tr>
<td>version/</td>
<td>3.1%</td>
<td>3.1%</td>
<td>3.1%</td>
<td>3.1%</td>
<td>3.1%</td>
<td>3.1%</td>
<td>3.1%</td>
<td>3.1%</td>
</tr>
<tr>
<td>volume/</td>
<td>▲ 7.7%</td>
<td>▼ 7.6%</td>
<td>▼ 7.6%</td>
<td>▼ 7.6%</td>
<td>▼ 7.7%</td>
<td>▼ 7.7%</td>
<td>▼ 7.6%</td>
<td>▼ 7.6%</td>
</tr>
</tbody>
</table>
```

---

**k8s.io/kubernetes/pkg/**
# Coverage summary

[k8s.io/kubernetes/pkg/kubelet/dockershim/](k8s.io/kubernetes/pkg/kubelet/dockershim/)

<table>
<thead>
<tr>
<th>File</th>
<th>e2e</th>
<th>conformance</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm/</td>
<td>5 / 40 (12.5%)</td>
<td>5 / 40 (12.5%)</td>
</tr>
<tr>
<td>convert.go</td>
<td>40 / 53 (75.5%)</td>
<td>40 / 53 (75.5%)</td>
</tr>
<tr>
<td>docker_checkpoint.go</td>
<td>6 / 6 (100.0%)</td>
<td>6 / 6 (100.0%)</td>
</tr>
<tr>
<td>docker_container.go</td>
<td>▲ 136 / 173 (76.6%)</td>
<td>132 / 173 (76.3%)</td>
</tr>
<tr>
<td>docker_image.go</td>
<td>44 / 71 (62.0%)</td>
<td>44 / 71 (62.0%)</td>
</tr>
<tr>
<td>docker_image_linux.go</td>
<td>0 / 1 (0.0%)</td>
<td>0 / 1 (0.0%)</td>
</tr>
<tr>
<td>docker_legacy_service.go</td>
<td>5 / 28 (17.9%)</td>
<td>5 / 28 (17.9%)</td>
</tr>
<tr>
<td>docker_logs.go</td>
<td>0 / 1 (0.0%)</td>
<td>0 / 1 (0.0%)</td>
</tr>
<tr>
<td>docker_sandbox.go</td>
<td>▲ 238 / 296 (80.4%)</td>
<td>221 / 296 (74.7%)</td>
</tr>
<tr>
<td>docker_service.go</td>
<td>▲ 53 / 155 (34.2%)</td>
<td>50 / 155 (32.3%)</td>
</tr>
<tr>
<td>docker_stats_linux.go</td>
<td>0 / 2 (0.0%)</td>
<td>0 / 2 (0.0%)</td>
</tr>
<tr>
<td>docker_streaming.go</td>
<td>▲ 65 / 79 (82.3%)</td>
<td>21 / 79 (26.8%)</td>
</tr>
<tr>
<td>exec.go</td>
<td>▲ 27 / 38 (71.1%)</td>
<td>25 / 38 (65.8%)</td>
</tr>
<tr>
<td>helpers.go</td>
<td>▲ 93 / 140 (66.4%)</td>
<td>81 / 140 (57.9%)</td>
</tr>
<tr>
<td>helpers_linux.go</td>
<td>▲ 27 / 44 (61.4%)</td>
<td>25 / 44 (56.8%)</td>
</tr>
<tr>
<td>libblocker/</td>
<td>▲ 274 / 906 (30.2%)</td>
<td>248 / 906 (27.4%)</td>
</tr>
<tr>
<td>metrics/</td>
<td>1 / 6 (16.7%)</td>
<td>1 / 6 (16.7%)</td>
</tr>
<tr>
<td>naming.go</td>
<td>19 / 27 (70.4%)</td>
<td>19 / 27 (70.4%)</td>
</tr>
<tr>
<td>network/</td>
<td>▼ 316 / 1012 (31.2%)</td>
<td>318 / 1012 (31.4%)</td>
</tr>
<tr>
<td>remote/</td>
<td>0 / 15 (0.0%)</td>
<td>0 / 15 (0.0%)</td>
</tr>
<tr>
<td>security_context.go</td>
<td>▲ 64 / 83 (77.1%)</td>
<td>59 / 83 (71.1%)</td>
</tr>
<tr>
<td>selinux_util.go</td>
<td>12 / 12 (100.0%)</td>
<td>12 / 12 (100.0%)</td>
</tr>
</tbody>
</table>
## Coverage Summary

### k8s.io/kubernetes/pkg/kubelet/prober/

<table>
<thead>
<tr>
<th>File</th>
<th>e2e</th>
<th>conformance</th>
</tr>
</thead>
<tbody>
<tr>
<td>prober.go</td>
<td>65 / 95 (68.4%)</td>
<td>62 / 95 (65.3%)</td>
</tr>
<tr>
<td>prober_manager.go</td>
<td>61 / 73 (83.6%)</td>
<td>61 / 73 (83.6%)</td>
</tr>
<tr>
<td>results/</td>
<td>20 / 26 (76.9%)</td>
<td>20 / 26 (76.9%)</td>
</tr>
<tr>
<td>worker.go</td>
<td>64 / 68 (94.1%)</td>
<td>62 / 68 (91.2%)</td>
</tr>
</tbody>
</table>

### k8s.io/kubernetes/pkg/probe/

<table>
<thead>
<tr>
<th>File</th>
<th>e2e</th>
<th>conformance</th>
</tr>
</thead>
<tbody>
<tr>
<td>exec/</td>
<td>9 / 11 (81.8%)</td>
<td>8 / 11 (72.7%)</td>
</tr>
<tr>
<td>http/</td>
<td>21 / 29 (72.4%)</td>
<td>21 / 29 (72.4%)</td>
</tr>
<tr>
<td>tcp/</td>
<td>0 / 9 (0.0%)</td>
<td>0 / 9 (0.0%)</td>
</tr>
</tbody>
</table>
Line Coverage

codecov.io, eg: https://codecov.io/gh/spiffxp/kubernetes
increased filtering
testgrid integration
treemap that links to source
How You Can Help

Contribute tests that exercise Pod functionality
Categorize existing tests
Run Conformance tests against your Kubernetes cluster
Participate in discussion at CNCF Conformance WG
Participate in discussion at SIG Architecture
References

- What is Kubernetes
  - https://github.com/kubernetes/community/tree/master/icons
  - https://github.com/kubernetes/community/blob/master/contributors/devel/arch-roadmap-1.png
- Using Conformance
  - https://github.com/cncf/k8s-conformance
  - https://testgrid.k8s.io/conformance-all
- Conformance Requirements
  - https://github.com/kubernetes/community/blob/master/contributors/devel/conformance-tests.md
  - https://github.com/kubernetes/community/tree/master/sig-architecture#conformance-definition
- Improving Conformance
  - https://github.com/cncf/apisnoop
  - https://blog.golang.org/cover
  - https://github.com/kubernetes/test-infra/tree/master/gopherage
- SIGs and WGs
  - https://github.com/kubernetes/community/tree/master/sig-architecture
  - https://github.com/kubernetes/community/tree/master/sig-testing
  - https://github.com/cncf/k8s-conformance/blob/master/README-WG.md
Contact Me

Aaron Crickenberger
@spiffxp
spiffxp@google.com
Thanks

@BenTheElder

@hh

@Katharine

@zachmandeville
Requirements*

Tests only GA, non-optional features or APIs
- No alpha/beta endpoints
- No feature flags required
- No deprecated features or endpoints
- Only default admission plugins
Requirements*

Works for all providers

- No SkipIfProviderIs / SkipUnlessProviderIs calls in test
- No cloud-provider-specific features, e.g. GCE monitoring, S3 Bucketing, etc.
Requirements*

Is non-privileged

- No root on nodes required
- No access to raw network interfaces required
- No cluster admin permissions required
Works without public internet access

- We will assume test images have been pre-pulled onto nodes
- No interaction with services outside of the cluster
Binaries used are required for Linux kernel or kubelet to run

- Binaries on node
- Can use binaries such as ip, mount
- No binaries such as git
Requirements*

Images used support all architectures for which Kubernetes releases are built

- amd64 (x86_64)
- arm64 (aarch64)
- arm
- ppc64le
- s390x
Passes against versions of Kubernetes consistent with version skew policy

- Nodes can lag control plane by 2 minor versions
- Clients can skew +/- 1 minor version from control plane
- The v1.3 control plane can work with v1.1, v1.2, v1.3 nodes
- The v1.3 control plane can work with v1.2, v1.3, v1.4 clients
- Tests are clients, conformance tests can’t be backward compatible
- Therefore conformance tests can lag control plane by 1 minor version
- The v1.3 conformance tests can work with v1.3, v1.4 clusters
Requirements*

Provides consistent results without flakes

- Ideally should be capable of running in parallel with other test cases
- Must not rely on test cases being run in a specific order
- Must not rely on best-effort delivery such as Events
- Must not validate content of fields which may change over time, e.g.
  - Reason
  - Message
  - Event