



KubeCon



CloudNativeCon

China 2018

Adventures in Conformance

Aaron Crickenberger, Google

@spiffxp



**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

Make it Boring



KubeCon



CloudNativeCon

China 2018

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



Keep it Boring



KubeCon



CloudNativeCon

China 2018



What is Kubernetes



KubeCon

CloudNativeCon

China 2018



Source: <https://github.com/kubernetes/community/tree/master/icons>

What is Kubernetes



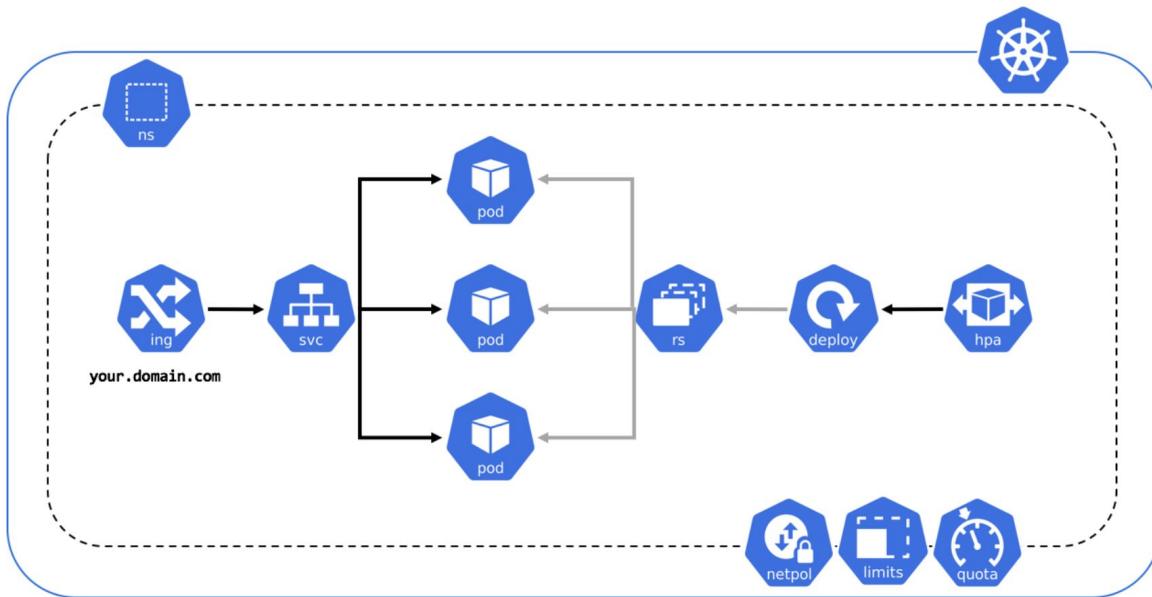
KubeCon

CloudNativeCon

China 2018

Usage Example

Exposed Pod with 3 replicas



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

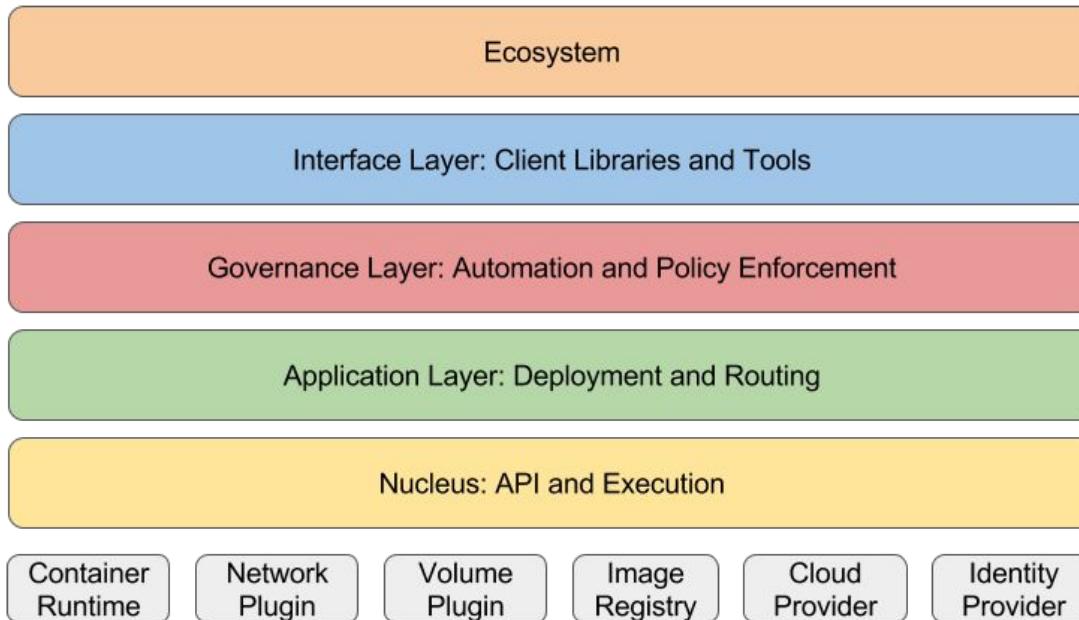
What is Kubernetes



KubeCon

CloudNativeCon

China 2018



What is Kubernetes



KubeCon

CloudNativeCon

China 2018

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

Improving Conformance



KubeCon



CloudNativeCon

China 2018

“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



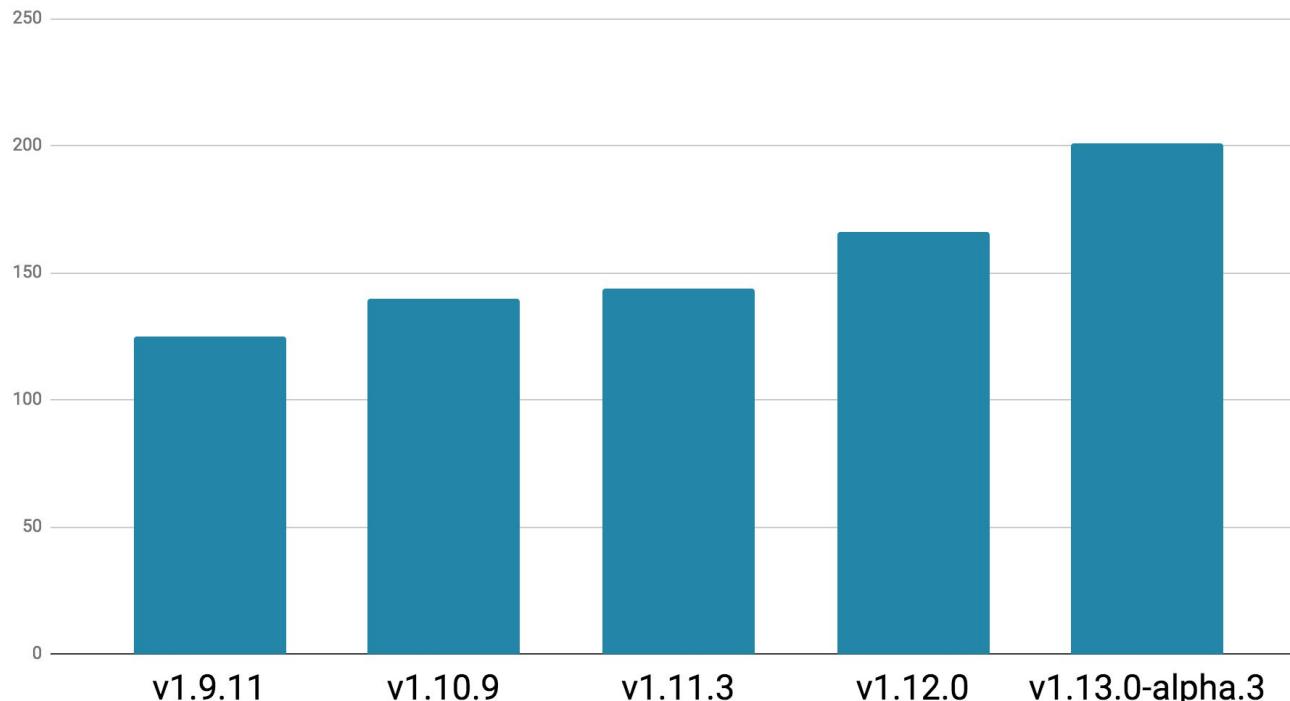
KubeCon



CloudNativeCon

China 2018

of Test Cases



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

Add Tests - 1.9 to 1.10



KubeCon



CloudNativeCon

China 2018

[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



KubeCon



CloudNativeCon

China 2018

[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



KubeCon



CloudNativeCon

China 2018

[k8s.io] Container Lifecycle Hook when create a pod with lifecycle hook should execute poststart exec hook properly [NodeConformance] [Conformance]
[k8s.io] Container Lifecycle Hook when create a pod with lifecycle hook should execute poststart http hook properly [NodeConformance] [Conformance]
[k8s.io] Container Lifecycle Hook when create a pod with lifecycle hook should execute prestop exec hook properly [NodeConformance] [Conformance]
[k8s.io] Container Lifecycle Hook when create a pod with lifecycle hook should execute prestop http hook properly [NodeConformance] [Conformance]
[k8s.io] InitContainer [NodeConformance] should invoke init containers on a RestartAlways pod [Conformance]
[k8s.io] InitContainer [NodeConformance] should invoke init containers on a RestartNever pod [Conformance]
[k8s.io] InitContainer [NodeConformance] should not start app containers and fail the pod if init containers fail on a RestartNever pod [Conformance]
[k8s.io] InitContainer [NodeConformance] should not start app containers if init containers fail on a RestartAlways pod [Conformance]
[sig-api-machinery] Namespaces [Serial] should ensure that all pods are removed when a namespace is deleted [Conformance]
[sig-api-machinery] Namespaces [Serial] should ensure that all services are removed when a namespace is deleted [Conformance]
[sig-apps] Deployment RecreateDeployment should delete old pods and create new ones [Conformance]
[sig-apps] Deployment RollingUpdateDeployment should delete old pods and create new ones [Conformance]
[sig-apps] Deployment deployment should delete old replica sets [Conformance]
[sig-apps] Deployment deployment should support proportional scaling [Conformance]
[sig-apps] Deployment deployment should support rollover [Conformance]
[sig-storage] ConfigMap binary data should be reflected in volume [NodeConformance] [Conformance]
[sig-storage] Secrets should be able to mount in a volume regardless of a different secret existing with same name in different namespace [NodeConformance] [Conformance]
[sig-storage] Subpath Atomic writer volumes should support subpaths with configmap pod [Conformance]
[sig-storage] Subpath Atomic writer volumes should support subpaths with configmap pod with mountPath of existing file [Conformance]
[sig-storage] Subpath Atomic writer volumes should support subpaths with downward pod [Conformance]
[sig-storage] Subpath Atomic writer volumes should support subpaths with projected pod [Conformance]
[sig-storage] Subpath Atomic writer volumes should support subpaths with secret pod [Conformance]

Add Tests - 1.12 to 1.13



CloudNativeCon

China 2018

[k8s.io] Container Runtime blackbox test when starting a container that exits should run with the expected status [NodeConformance] [Conformance]
[k8s.io] Kubelet when scheduling a busybox Pod with hostAliases should write entries to /etc/hosts [NodeConformance] [Conformance]
[k8s.io] Kubelet when scheduling a busybox command in a pod should print the output to logs [NodeConformance] [Conformance]
[k8s.io] Kubelet when scheduling a busybox command that always fails in a pod should be possible to delete [NodeConformance] [Conformance]
[k8s.io] Kubelet when scheduling a busybox command that always fails in a pod should have an terminated reason [NodeConformance] [Conformance]
[k8s.io] Kubelet when scheduling a read only busybox container should not write to root filesystem [NodeConformance] [Conformance]
[k8s.io] Pods should support remote command execution over websockets [NodeConformance] [Conformance]
[k8s.io] Pods should support retrieving logs from the container over websockets [NodeConformance] [Conformance]
[sig-apps] ReplicaSet should adopt matching pods on creation and release no longer matching pods [Conformance]
[sig-apps] ReplicationController should adopt matching pods on creation [Conformance]
[sig-apps] ReplicationController should release no longer matching pods [Conformance]
[sig-cli] Kubectl client [k8s.io] Guestbook application should create and stop a working application [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl api-versions should check if v1 is in available api versions [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl cluster-info should check if Kubernetes master services is included in cluster-info [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl describe should check if kubectl describe prints relevant information for rc and pods [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl expose should create services for rc [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl label should update the label on a resource [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl logs should be able to retrieve and filter logs [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl patch should add annotations for pods in rc [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl replace should update a single-container pod's image [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl rolling-update should support rolling-update to same image [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl run --rm job should create a job from an image, then delete the job [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl run default should create an rc or deployment from an image [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl run deployment should create a deployment from an image [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl run job should create a job from an image when restart is OnFailure [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl run pod should create a pod from an image when restart is Never [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl run rc should create an rc from an image [Conformance]
[sig-cli] Kubectl client [k8s.io] Kubectl version should check is all data is printed [Conformance]
[sig-cli] Kubectl client [k8s.io] Proxy server should support --unix-socket=/path [Conformance]
[sig-cli] Kubectl client [k8s.io] Proxy server should support proxy with --port 0 [Conformance]
[sig-cli] Kubectl client [k8s.io] Update Demo should create and stop a replication controller [Conformance]
[sig-cli] Kubectl client [k8s.io] Update Demo should do a rolling update of a replication controller [Conformance]
[sig-cli] Kubectl client [k8s.io] Update Demo should scale a replication controller [Conformance]
[sig-storage] EmptyDir wrapper volumes should not cause race condition when used for configmaps [Serial] [Slow] [Conformance]
[sig-storage] EmptyDir wrapper volumes should not conflict [Conformance]

Add Tests



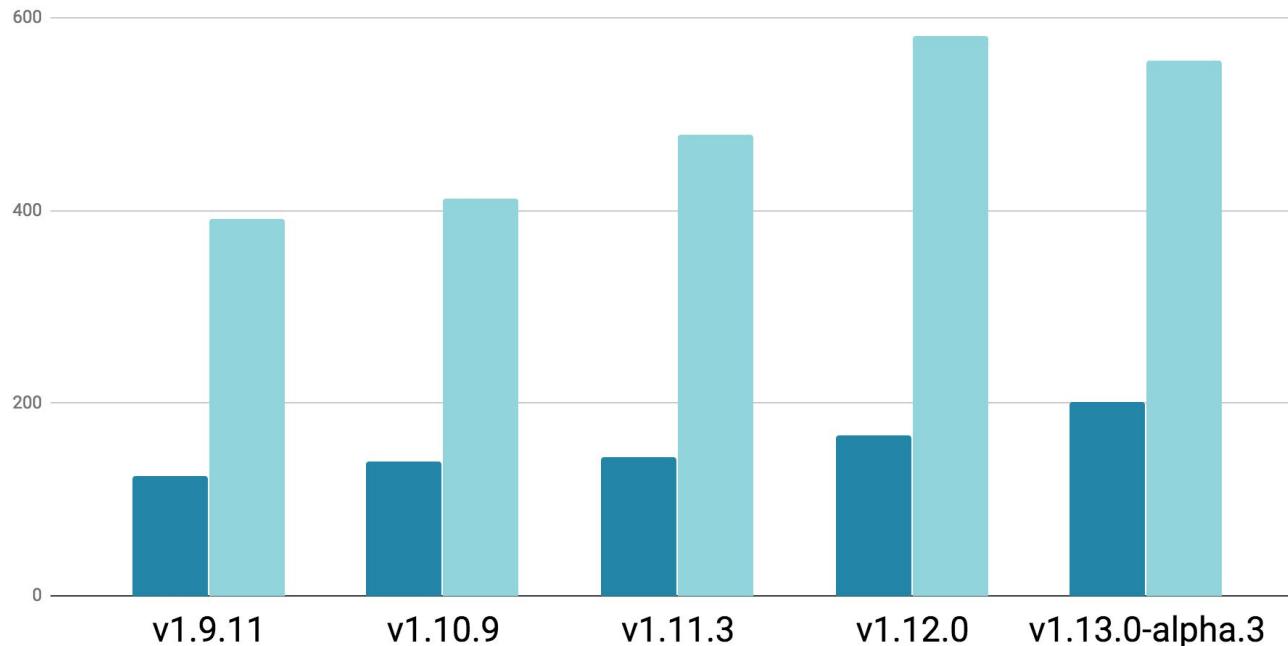
KubeCon

CloudNativeCon

China 2018

of Test Cases

■ Conformance ■ Release e2e



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



KubeCon



CloudNativeCon

China 2018

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

```
$ 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-image=gci \
--gcp-node-image=gci \
--gcp-zone=us-central1-f \
--gcp-project=your-gcp-project \
--test_args="--ginkgo.focus='[Conformance\\]' \
--timeout=200m | 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

```
$ <build-log.txt sed \
-e 's/\x1b\[10-9;\]*m//g' \
-e 's/^[[WE]] [0-9]\{4\}\. [0-9]\{2\}\:[0-9]\{2\}\:[0-9]\{2\}\:[0-9]\{3\}\] //'
-e '/^SUCCESS!/{
```

Getting Data



KubeCon



CloudNativeCon

China 2018

conformance-all conformance-gce conformance-kind conformance-cloud-provider-openstack conformance-cloud-provider-vsphere conformance-vsphere conformance-alibaba-cloud-provider conformance-cloud-provider-digitalocean									
conformance-cloud-provider-baiducloud conformance-gardener conformance-hack-local-up-cluster									
Summary GCE, master (dev) GCE, v1.13 (release) GCE, v1.13 (dev) GCE, v1.12 (release) GCE, v1.12 (dev) GCE, v1.11 (release) GCE, v1.11 (dev) GCE, v1.10 (release) GCE, v1.10 (dev) kind, master (dev) kind, master (dev) [non-serial]									
kind, v1.12 (dev) kind, v1.11 (dev) local-up-cluster, master (dev) local-up-cluster-containerized, master (dev) OpenStack, master (dev) OpenStack, v1.10 (dev) OpenStack, v1.11 (dev) vSphere-cloud-provider, master (dev) vSphere-cloud-provider, v1.10 (dev) vSphere, master (dev) vSphere, v1.10 (dev) vSphere, v1.11 (dev) vSphere, v1.12 (dev) DigitalOcean Master DigitalOcean v1.10									
DigitalOcean v1.11 DigitalOcean v1.12 BaiduCloud v1.8 BaiduCloud v1.11									
GCE, master (dev): PASSING									
All passing in the past week									
GCE, v1.13 (release): FAILING									
56 of 84 tests (66.7%) and 28 of 28 runs (100.0%) failing in the past week									
Test Name	# Fails	Failure Messages	First Failed	Last Passed	Bugs	Regression			
Extract	29		11-07 15:06						
Overall	29		11-07 15:06						
GCE, v1.13 (dev): PASSING							Last green run: 875	Tests last ran on: 11-14 13:10	Last update: 11-14 16:24
All passing in the past week									
GCE, v1.12 (release): FAILING							No passing build found	Tests last ran on: 11-14 15:34	Last update: 11-14 16:12
All 56 of 84 tests (66.7%) and 28 of 28 runs (100.0%) failing in the past week									
Test Name	# Fails	Failure Messages	First Failed	Last Passed	Bugs	Regression			
Extract	29		11-07 15:06				Last green run: 28	Tests last ran on: 11-14 15:34	Last update: 11-14 16:22
Overall	29		11-07 15:06						
GCE, v1.13 (dev): PASSING							Last green run: 173	Tests last ran on: 11-14 12:26	Last update: 11-14 16:15
All passing in the past week									
GCE, v1.12 (release): PASSING							Last green run: 339	Tests last ran on: 11-14 16:18	Last update: 11-14 16:21
All passing in the past week									
GCE, v1.12 (dev): PASSING							Last green run: 339	Tests last ran on: 11-14 10:17	Last update: 11-14 16:10
All passing in the past week									
GCE, v1.11 (release): PASSING							Last green run: 677	Tests last ran on: 11-14 10:57	Last update: 11-14 16:21
All passing in the past week									
GCE, v1.11 (dev): PASSING							Last green run: 835	Tests last ran on: 11-14 14:16	Last update: 11-14 16:17
All passing in the past week									
GCE, v1.10 (release): PASSING							Last green run: 834	Tests last ran on: 11-14 11:09	Last update: 11-14 16:23
All passing in the past week									
GCE, v1.10 (dev): PASSING							Last green run: 1210	Tests last ran on: 11-14 15:38	Last update: 11-14 16:20
All passing in the past week									
kind, master (dev): PASSING							Last green run: 3579	Tests last ran on: 11-02 21:27	Last update: 11-14 16:23
6 of 19153 tests (0.0%) and 5 of 107 runs (4.7%) failing in the past week									
kind, master (dev) [non-serial]: PASSING							Last green run: 719	Tests last ran on: 11-14 15:38	Last update: 11-14 16:13
No results in the past week									
kind, v1.12 (dev): FLAKY							Last green run: 579	Tests last ran on: 11-14 15:58	Last update: 11-14 16:13
9 of 19116 tests (0.0%) and 6 of 118 runs (5.1%) failing in the past week									
kind, v1.11 (dev): PASSING							Last green run: 579	Tests last ran on: 11-14 15:58	Last update: 11-14 16:13
4 of 17220 tests (0.0%) and 4 of 123 runs (3.3%) failing in the past week									

Source: <https://testgrid.k8s.io/conformance-all>

Getting Data



KubeCon



CloudNativeCon

China 2018

conformance-all conformance-gce conformance-kind conformance-cloud-provider-openstack conformance-cloud-provider-vsphere conformance-vsphere conformance-alibaba-cloud-provider conformance-cloud-provider-digitalocean
conformance-cloud-provider-baiducloud conformance-gardener conformance-hack-local-up-cluster

Summary GCE, master (dev) GCE, v1.13 (release) **GCE, v1.13 (dev)** GCE, v1.12 (release) GCE, v1.12 (dev) GCE, v1.11 (release) GCE, v1.11 (dev) GCE, v1.10 (release) GCE, v1.10 (dev) kind, master (dev) kind, master (dev) [non-serial]
kind, v1.12 (dev) kind, v1.11 (dev) local-up-cluster, master (dev) local-up-cluster-containerized, master (dev) OpenStack, master (dev) OpenStack, v1.10 (dev) OpenStack, v1.11 (dev) vSphere-cloud-provider, master (dev)
vSphere-cloud-provider, v1.10 (dev) vSphere-cloud-provider, v1.11 (dev) vSphere-cloud-provider, v1.12 (dev) vSphere, master (dev) vSphere, v1.10 (dev) vSphere, v1.11 (dev) vSphere, v1.12 (dev) DigitalOcean Master DigitalOcean v1.10
DigitalOcean v1.11 DigitalOcean v1.12 BaiduCloud v1.8 BaiduCloud v1.11

Runs conformance tests using kubetest against kubernetes release 1.13 branch on GCE

Passed @ 27 (11-14 03:32)

	11-14			11-13				11-12			
	15:34	09:33	03:32	21:31	15:30	09:29	03:28	21:27	15:26	09:25	03:
29	28	27	26	25	24	23	22	21	20	1	
		20b5777d4			8c5acd51e				03aacded1		
			e4c33a675			9b1101302		3272d80c0			

Show 7 stale tests (no results in last 3+ runs)

- Check APIReachability
- Deferred TearDown
- DumpClusterLogs
- Extract
- IsUp
- Overall
- TearDown
- TearDown Previous
- Test
- Timeout
- Up

[k8s.io] Container Lifecycle Hook when create a pod with lifecycle hook should execute poststart exec hook properly [NodeConformance]

[k8s.io] Container Lifecycle Hook when create a pod with lifecycle hook should execute poststart http hook properly [NodeConformance]

[k8s.io] Container Lifecycle Hook when create a pod with lifecycle hook should execute prestop exec hook properly [NodeConformance]

[k8s.io] Container Lifecycle Hook when create a pod with lifecycle hook should execute prestop http hook properly [NodeConformance]

[k8s.io] Container Runtime blackbox test when starting a container that exits should run with the expected status [NodeConformance]

[k8s.io] Docker Containers should be able to override the image's default arguments (docker cmd) [NodeConformance] [Conformance]

[k8s.io] Docker Containers should be able to override the image's default command (docker entrypoint) [NodeConformance] [Conformance]

[k8s.io] Docker Containers should be able to override the image's default command and arguments [NodeConformance] [Conformance]

[k8s.io] Docker Containers should use the image defaults if command and args are blank [NodeConformance] [Conformance]

[k8s.io] InitContainer [NodeConformance] should invoke init containers on a RestartAlways pod [Conformance]

[k8s.io] InitContainer [NodeConformance] should invoke init containers on a RestartNever pod [Conformance]

[k8s.io] InitContainer [NodeConformance] should not start app containers and fail the pod if init containers fail on a RestartNever pod [Conformance]

[k8s.io] InitContainer [NodeConformance] should not start app containers if init containers fail on a RestartAlways pod [Conformance]

[k8s.io] Kubelet when scheduling a busybox Pod with hostAliases should write entries to /etc/hosts [NodeConformance] [Conformance]

[k8s.io] Kubelet when scheduling a busybox command in a pod should print the output to logs [NodeConformance] [Conformance]

[k8s.io] Kubelet when scheduling a busybox command that always fails in a pod should be possible to delete [NodeConformance] [Conformance]

[k8s.io] Kubelet when scheduling a read only busybox container should not write to root filesystem [NodeConformance] [Conformance]

[k8s.io] KubeletManagedEtcHosts should test kubelet managed /etc/hosts file [NodeConformance] [Conformance]

Getting Data

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

Recent runs

Result	SUCCESS
Tests	0 failed / 214 succeeded
Started	2018-11-14 19:32 CST
Elapsed	1h49m
Version	v1.13.0-beta.0.257+20b5777d454e56
Builder	gke-prow-default-pool-3c8994a8-6sn5
pod	e45515d5-e800-11e8-bc35-0a580a6c0111
infra-commit	e4c33a675
job-version	v1.13.0-beta.0.257+20b5777d454e56
master_os_image	cos-stable-65-10323-64-0
node_os_image	cos-stable-65-10323-64-0
pod	e45515d5-e800-11e8-bc35-0a580a6c0111
version	v1.13.0-beta.0.257+20b5777d454e56

[artifacts](#) [build log](#)

No Test Failures!

[Show 214 Passed Tests](#)

[Show 1708 Skipped Tests](#)

Getting Data

kubernetes-jenkins

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

Name	Size	Modified
..	-	-
bootstrap-e2e-master/	-	-
bootstrap-e2e-minion-group-8zg1/	-	-
bootstrap-e2e-minion-group-d3lt/	-	-
bootstrap-e2e-minion-group-h05k/	-	-
images.json	98	14 Nov 2018 13:22:17
junit_01.xml	41580	14 Nov 2018 13:22:17
junit_runner.xml	388	14 Nov 2018 13:22:17
metadata.json	186	14 Nov 2018 13:22:17
nodes.yaml	20633	14 Nov 2018 13:22:17

kubernetes-jenkins

/kubernetes-jenkins/logs/ci-kubernetes-gce-conformance-latest-1-13/27/artifacts/bootstrap-e2e-master/

Name	Size	Modified
..	-	-
docker.log	343907	14 Nov 2018 13:22:18
etcd-events.log	2227	14 Nov 2018 13:22:18
etcd.log	3245	14 Nov 2018 13:22:18
glbc.log	2708	14 Nov 2018 13:22:17
kern.log	10311	14 Nov 2018 13:22:18
kube-addon-manager.log	3487	14 Nov 2018 13:22:18
kube-apiserver-audit.log	1324402	14 Nov 2018 13:22:19
kube-apiserver-audit.log-20181114-1542200434.gz	9621977	14 Nov 2018 13:22:18
kube-apiserver.log	3060074	14 Nov 2018 13:22:19
kube-container-runtime-monitor.log	106	14 Nov 2018 13:22:17
kube-controller-manager.log	968178	14 Nov 2018 13:22:19
kube-master-configuration.log	1761	14 Nov 2018 13:22:17
kube-master-installation.log	1120	14 Nov 2018 13:22:18
kube-scheduler.log	43992	14 Nov 2018 13:22:18
kubelet-monitor.log	121	14 Nov 2018 13:22:18
kubelet.log	1375721	14 Nov 2018 13:22:19
serial-1.log	11858	14 Nov 2018 13:22:18

API Coverage

Use [cncf/apisnoop](#)

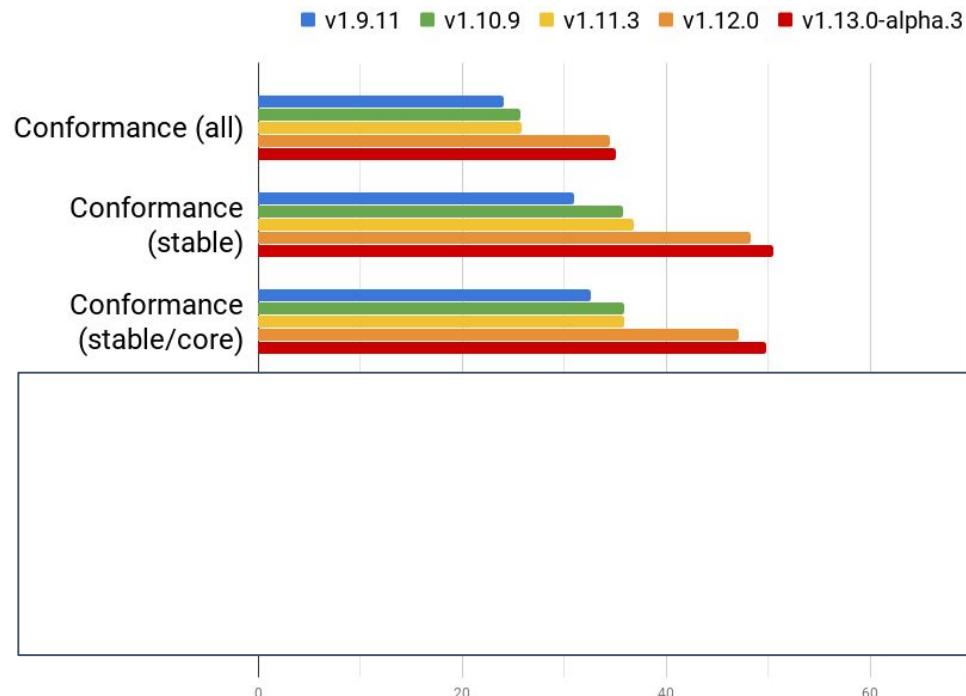
- Parse Kubernetes' OpenAPI spec to find possible endpoints
 - Endpoint = VERB path
 - e.g. POST /api/v1/namespaces/{namespace}/pods
- Parse Kubernetes audit log to find endpoint hits

Questions we can answer:

- What are the total endpoints that can be hit?
- What are the actual endpoints that were hit?

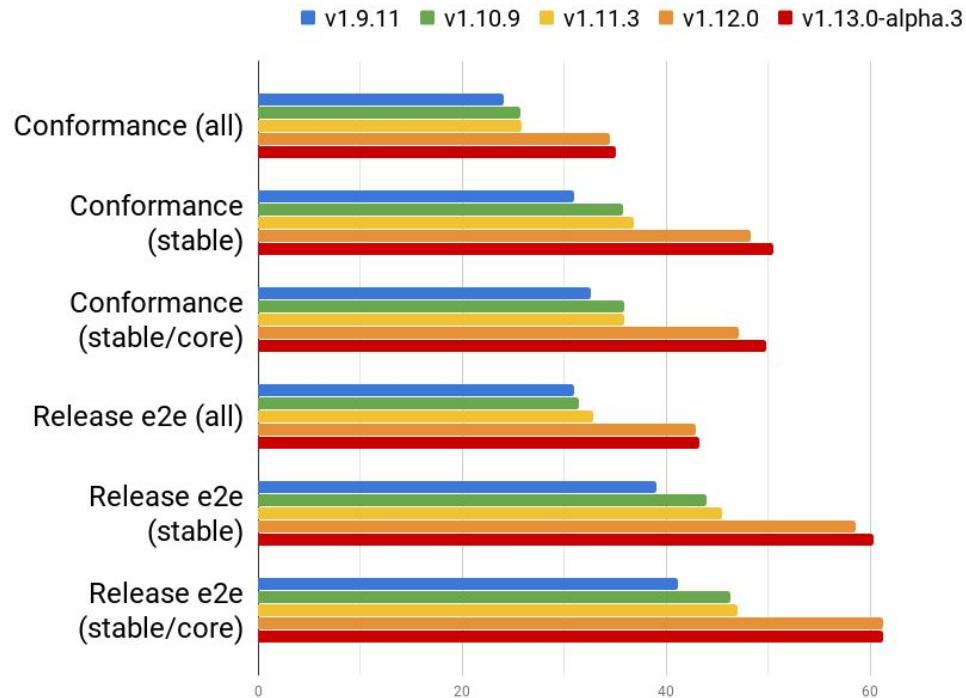
API Coverage - by %

API Coverage % by Release



API Coverage - by %

API Coverage % by Release



API Coverage - by ep

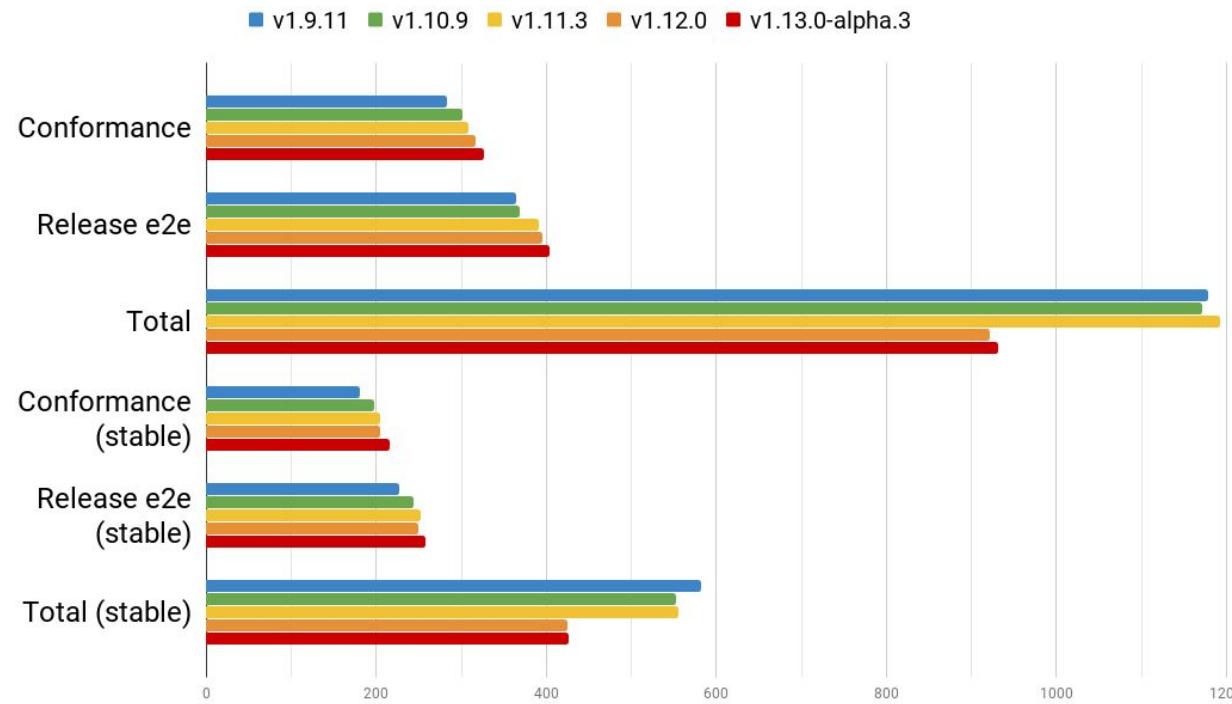


KubeCon

CloudNativeCon

China 2018

Covered Endpoints by Release



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

API Coverage - covered



KubeCon



CloudNativeCon

China 2018

get /api/	1	get /api/v1/namespaces/{namespace}/pods/{name}/portforward	0	get /api/v1/namespaces/{namespace}/serviceaccounts/{name}	1
get /api/v1/	1	post /api/v1/namespaces/{namespace}/pods/{name}/portforward	0	patch /api/v1/namespaces/{namespace}/serviceaccounts/{name}	0
watch /api/v1/componentstatuses	0	delete /api/v1/namespaces/{namespace}/pods/{name}/proxy	0	get /api/v1/namespaces/{namespace}/services	1
get /api/v1/componentstatuses/{name}	0	get /api/v1/namespaces/{namespace}/pods/{name}/proxy	1	post /api/v1/namespaces/{namespace}/services	1
get /api/v1/events	0	head /api/v1/namespaces/{namespace}/pods/{name}/proxy	0	watch /api/v1/namespaces/{namespace}/services	0
watch /api/v1/events	0	options /api/v1/namespaces/{namespace}/pods/{name}/proxy	1	delete /api/v1/namespaces/{namespace}/services/{name}	1
get /api/v1/namespaces	1	patch /api/v1/namespaces/{namespace}/pods/{name}/proxy	0	get /api/v1/namespaces/{namespace}/services/{name}	1
post /api/v1/namespaces	1	post /api/v1/namespaces/{namespace}/pods/{name}/proxy	0	patch /api/v1/namespaces/{namespace}/services/{name}	0
post /api/v1/namespaces/{namespace}/bindings	0	put /api/v1/namespaces/{namespace}/pods/{name}/proxy	0	put /api/v1/namespaces/{namespace}/services/{name}	0
get /api/v1/namespaces/{namespace}/configmaps	1	delete /api/v1/namespaces/{namespace}/pods/{name}/proxy/{path}	0	delete /api/v1/namespaces/{namespace}/services/{name}/proxy	0
post /api/v1/namespaces/{namespace}/configmaps	1	get /api/v1/namespaces/{namespace}/pods/{name}/proxy/{path}	1	get /api/v1/namespaces/{namespace}/services/{name}/proxy	1
watch /api/v1/namespaces/{namespace}/configmaps	1	head /api/v1/namespaces/{namespace}/pods/{name}/proxy/{path}	0	head /api/v1/namespaces/{namespace}/services/{name}/proxy	0
delete /api/v1/namespaces/{namespace}/configmaps/{name}	1	options /api/v1/namespaces/{namespace}/pods/{name}/proxy/{path}	0	options /api/v1/namespaces/{namespace}/services/{name}/proxy	0
get /api/v1/namespaces/{namespace}/configmaps/{name}	1	patch /api/v1/namespaces/{namespace}/pods/{name}/proxy/{path}	0	patch /api/v1/namespaces/{namespace}/services/{name}/proxy	0
patch /api/v1/namespaces/{namespace}/configmaps/{name}	0	post /api/v1/namespaces/{namespace}/pods/{name}/proxy/{path}	0	post /api/v1/namespaces/{namespace}/services/{name}/proxy	0
put /api/v1/namespaces/{namespace}/configmaps/{name}	1	put /api/v1/namespaces/{namespace}/pods/{name}/proxy/{path}	0	put /api/v1/namespaces/{namespace}/services/{name}/proxy	0
get /api/v1/namespaces/{namespace}/endpoints	1	get /api/v1/namespaces/{namespace}/pods/{name}/status	1	delete /api/v1/namespaces/{namespace}/services/{name}/proxy/{path}	0
watch /api/v1/namespaces/{namespace}/endpoints	1	put /api/v1/namespaces/{namespace}/podtemplates	1	get /api/v1/namespaces/{namespace}/services/{name}/proxy/{path}	1
get /api/v1/namespaces/{namespace}/endpoints/{name}	1	get /api/v1/namespaces/{namespace}/podtemplates	0	head /api/v1/namespaces/{namespace}/services/{name}/proxy/{path}	0
patch /api/v1/namespaces/{namespace}/endpoints/{name}	0	post /api/v1/namespaces/{namespace}/podtemplates	0	options /api/v1/namespaces/{namespace}/services/{name}/proxy/{path}	0
delete /api/v1/namespaces/{namespace}/events	0	watch /api/v1/namespaces/{namespace}/podtemplates	1	patch /api/v1/namespaces/{namespace}/services/{name}/proxy/{path}	0
get /api/v1/namespaces/{namespace}/events	0	get /api/v1/namespaces/{namespace}/replicationcontrollers	1	post /api/v1/namespaces/{namespace}/services/{name}/proxy/{path}	0
patch /api/v1/namespaces/{namespace}/events	0	post /api/v1/namespaces/{namespace}/replicationcontrollers	1	put /api/v1/namespaces/{namespace}/services/{name}/proxy/{path}	0
put /api/v1/namespaces/{namespace}/events/{name}	0	watch /api/v1/namespaces/{namespace}/replicationcontrollers	1	get /api/v1/namespaces/{namespace}/services/{name}/status	0
get /api/v1/namespaces/{namespace}/events/{name}	0	get /api/v1/namespaces/{namespace}/replicationcontrollers	1	patch /api/v1/namespaces/{namespace}/services/{name}/status	0
patch /api/v1/namespaces/{namespace}/events/{name}	0	patch /api/v1/namespaces/{namespace}/replicationcontrollers	1	put /api/v1/namespaces/{namespace}/services/{name}/status	0
put /api/v1/namespaces/{namespace}/events/{name}	0	watch /api/v1/namespaces/{namespace}/replicationcontrollers	1	get /api/v1/namespaces/{namespace}/services/{name}/status	0
get /api/v1/namespaces/{namespace}/limits/ranges	1	delete /api/v1/namespaces/{namespace}/replicationcontrollers/{name}	1	delete /api/v1/nodes/{name}	1
watch /api/v1/namespaces/{namespace}/limits/ranges	0	get /api/v1/namespaces/{namespace}/replicationcontrollers/{name}	1	get /api/v1/namespaces/{name}	1
delete /api/v1/namespaces/{namespace}/limits/ranges/{name}	0	patch /api/v1/namespaces/{namespace}/replicationcontrollers/{name}	1	put /api/v1/namespaces/{name}	1
patch /api/v1/namespaces/{namespace}/limits/ranges/{name}	0	patch /api/v1/namespaces/{namespace}/replicationcontrollers/{name}/scale	0	get /api/v1/namespaces/{name}/status	0
put /api/v1/namespaces/{namespace}/limits/ranges/{name}	0	get /api/v1/namespaces/{namespace}/replicationcontrollers/{name}/status	0	patch /api/v1/namespaces/{name}/status	0
get /api/v1/namespaces/{namespace}/persistentvolumeclaims	1	patch /api/v1/namespaces/{namespace}/replicationcontrollers/{name}/status	0	delete /api/v1/nodes/{name}	0
post /api/v1/namespaces/{namespace}/persistentvolumeclaims	0	get /api/v1/namespaces/{namespace}/resourcequotas	1	get /api/v1/nodes/{name}	1
watch /api/v1/namespaces/{namespace}/persistentvolumeclaims	0	post /api/v1/namespaces/{namespace}/resourcequotas	0	patch /api/v1/nodes/{name}	1
delete /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}	0	watch /api/v1/namespaces/{namespace}/resourcequotas	0	put /api/v1/nodes/{name}	1
get /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}	0	delete /api/v1/namespaces/{namespace}/resourcequotas/{name}	0	get /api/v1/nodes/{name}	0
patch /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}	0	get /api/v1/namespaces/{namespace}/resourcequotas/{name}	0	patch /api/v1/nodes/{name}/proxy	0
put /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}	0	patch /api/v1/namespaces/{namespace}/resourcequotas/{name}	0	get /api/v1/nodes/{name}/proxy	0
get /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}/status	0	put /api/v1/namespaces/{namespace}/resourcequotas/{name}	0	patch /api/v1/nodes/{name}/proxy	0
patch /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}/status	0	put /api/v1/namespaces/{namespace}/resourcequotas/{name}/status	0	post /api/v1/nodes/{name}/proxy	0
put /api/v1/namespaces/{namespace}/persistentvolumeclaims/{name}/status	0	patch /api/v1/namespaces/{namespace}/resourcequotas/{name}/status	0	put /api/v1/nodes/{name}/proxy	0
get /api/v1/namespaces/{namespace}/pods	1	put /api/v1/namespaces/{namespace}/resourcequotas/{name}/status	0	delete /api/v1/nodes/{name}/proxy/{path}	0
post /api/v1/namespaces/{namespace}/pods	1	get /api/v1/namespaces/{namespace}/secrets	1	get /api/v1/nodes/{name}/proxy/{path}	1
watch /api/v1/namespaces/{namespace}/pods	1	post /api/v1/namespaces/{namespace}/secrets	1	head /api/v1/nodes/{name}/proxy/{path}	0
delete /api/v1/namespaces/{namespace}/pods/{name}	1	delete /api/v1/namespaces/{namespace}/secrets/{name}	1	options /api/v1/nodes/{name}/proxy/{path}	0
get /api/v1/namespaces/{namespace}/pods/{name}	1	get /api/v1/namespaces/{namespace}/secrets/{name}	1	patch /api/v1/nodes/{name}/proxy/{path}	0
patch /api/v1/namespaces/{namespace}/pods/{name}	1	patch /api/v1/namespaces/{namespace}/secrets/{name}	0	post /api/v1/nodes/{name}/proxy/{path}	0
put /api/v1/namespaces/{namespace}/pods/{name}	1	put /api/v1/namespaces/{namespace}/secrets/{name}	0	put /api/v1/nodes/{name}/proxy/{path}	0
get /api/v1/namespaces/{namespace}/pods/{name}/attach	0	get /api/v1/namespaces/{namespace}/serviceaccounts	1	get /api/v1/nodes/{name}/status	0
post /api/v1/namespaces/{namespace}/pods/{name}/eviction	0	post /api/v1/namespaces/{namespace}/serviceaccounts	1	put /api/v1/nodes/{name}/status	0
get /api/v1/namespaces/{namespace}/pods/{name}/log	1	watch /api/v1/namespaces/{namespace}/serviceaccounts	1	put /api/v1/nodes/{name}/status	0
get /api/v1/namespaces/{namespace}/pods/{name}/portforward	0	delete /api/v1/namespaces/{namespace}/serviceaccounts/{name}	0	get /api/v1/nodes/{name}/status	0

API Coverage - covered



CloudNativeCon
China 2018

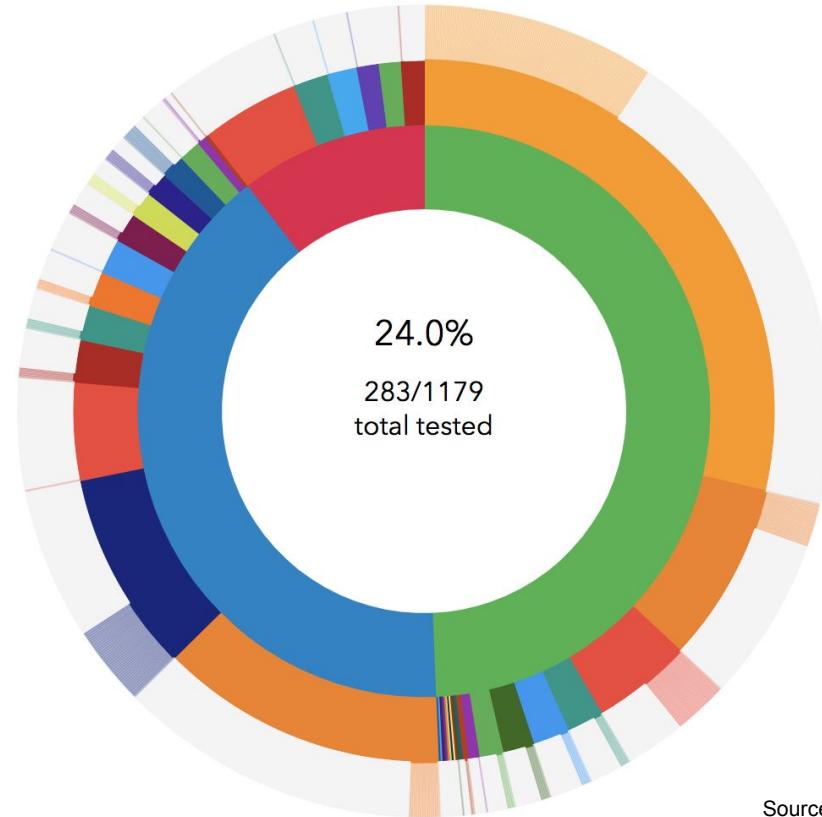
delete	/api/v1/persistentvolumes	0 put	/apis/apps/v1/namespaces/{namespace}/replicasesets/{name}/scale	0 delete	/apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}	0
get	/api/v1/persistentvolumes	1 get	/apis/apps/v1/namespaces/{namespace}/replicasesets/{name}/status	0 get	/apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}	0
post	/api/v1/persistentvolumes	0 patch	/apis/apps/v1/namespaces/{namespace}/replicasesets/{name}/status	0 patch	/apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}	0
delete	/api/v1/persistentvolumes/{name}	0 get	/apis/apps/v1/namespaces/{namespace}/statefulsets	1 put	/apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies/{name}	0
get	/api/v1/persistentvolumes/{name}	0 post	/apis/apps/v1/namespaces/{namespace}/statefulsets	1 get	/apis/policy/	0
patch	/api/v1/persistentvolumes/{name}	0 watch	/apis/apps/v1/namespaces/{namespace}/statefulsets	0 get	/apis/rbac.authorization.k8s.io/	0
put	/api/v1/persistentvolumes/{name}	0 delete	/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}	1 get	/apis/rbac.authorization.k8s.io/v1/	1
get	/api/v1/persistentvolumes/{name}/status	0 get	/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}	1 delete	/apis/rbac.authorization.k8s.io/v1/clusterrolebindings	0
patch	/api/v1/persistentvolumes/{name}/status	0 patch	/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}	0 delete	/apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}	0
put	/api/v1/persistentvolumes/{name}/status	0 put	/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}	1 put	/apis/rbac.authorization.k8s.io/v1/clusterrolebindings/{name}	0
get	/api/v1/pods	1 get	/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/scale	0 delete	/apis/rbac.authorization.k8s.io/v1/clusterroles	0
get	/apis/	1 patch	/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/scale	0 delete	/apis/rbac.authorization.k8s.io/v1/clusterroles/{name}	0
get	/apis/admissionregistration.k8s.io/	0 put	/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/scale	0 get	/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings	1
get	/apis/apextensions.k8s.io/	0 get	/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/status	0 watch	/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings	0
get	/apis/appregistration.k8s.io/	0 patch	/apis/apps/v1/namespaces/{namespace}/statefulsets/{name}/status	0 delete	/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}	0
get	/apis/appregistration.k8s.io/v1/	1 get	/apis/appregistration.k8s.io/	0 patch	/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/rolebindings/{name}	0
delete	/apis/appregistration.k8s.io/v1/apiservices	0 get	/apis/authentication.k8s.io/v1/	1 get	/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles	1
get	/apis/appregistration.k8s.io/v1/apiservices/{name}	0 get	/apis/authentication.k8s.io/v1/	0 watch	/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles	0
patch	/apis/appregistration.k8s.io/v1/apiservices/{name}	0 get	/apis/authentication.k8s.io/v1/	0 delete	/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}	0
put	/apis/appregistration.k8s.io/v1/apiservices/{name}	0 get	/apis/authentication.k8s.io/v1/	1 delete	/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}	0
get	/apis/appregistration.k8s.io/v1/apiservices/{name}/status	0 post	/apis/authentication.k8s.io/v1/namespaces/{namespace}/localsubjectaccesreviews	0 patch	/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}	0
patch	/apis/appregistration.k8s.io/v1/apiservices/{name}/status	0 post	/apis/authentication.k8s.io/v1/selfsubjectaccesreviews	0 put	/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}	0
get	/apis/apps/	0 post	/apis/authentication.k8s.io/v1/selfsubjectaccesreviews	0 get	/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}	0
get	/apis/apps/v1/	1 post	/apis/authentication.k8s.io/v1/selfsubjectaccesreviews	0 get	/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}	0
get	/apis/apps/v1/namespaces/{namespace}/controllerrevisions	1 get	/apis/authentication.k8s.io/v1/subjectaccesreviews	0 get	/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}	0
watch	/apis/apps/v1/namespaces/{namespace}/controllerrevisions	0 get	/apis/autoscaling/	0 get	/apis/rbac.authorization.k8s.io/v1/namespaces/{namespace}/roles/{name}	0
patch	/apis/apps/v1/namespaces/{namespace}/controllerrevisions/{name}	0 get	/apis/autoscaling/v1/	1 get	/apis/storage.k8s.io/v1/	1
get	/apis/apps/v1/namespaces/{namespace}/daemonsets	1 get	/apis/autoscaling/v1/namespaces/{namespace}/horizontalpodautoscalers	1 delete	/apis/storage.k8s.io/v1/storageclasses	0
post	/apis/apps/v1/namespaces/{namespace}/daemonsets	1 post	/apis/autoscaling/v1/namespaces/{namespace}/horizontalpodautoscalers	0 delete	/apis/storage.k8s.io/v1/storageclasses/{name}	0
watch	/apis/apps/v1/namespaces/{namespace}/daemonsets	1 watch	/apis/autoscaling/v1/namespaces/{namespace}/horizontalpodautoscalers	0 get	/apis/storage.k8s.io/v1/storageclasses	0
delete	/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}	0 delete	/apis/autoscaling/v1/namespaces/{namespace}/horizontalpodautoscalers/{name}	0 patch	/apis/storage.k8s.io/v1/storageclasses/{name}	0
get	/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}	0 get	/apis/autoscaling/v1/namespaces/{namespace}/horizontalpodautoscalers/{name}	0 put	/apis/storage.k8s.io/v1/storageclasses/{name}	0
patch	/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}	1 patch	/apis/autoscaling/v1/namespaces/{namespace}/horizontalpodautoscalers/{name}	0 get	/logs/	0
put	/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}	1 put	/apis/autoscaling/v1/namespaces/{namespace}/horizontalpodautoscalers/{name}	0 get	/logs/lologpath/	0
get	/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}/status	0 get	/apis/autoscaling/v1/namespaces/{namespace}/horizontalpodautoscalers/{name}/status	0 get	/version/	0
patch	/apis/apps/v1/namespaces/{namespace}/daemonsets/{name}/status	0 patch	/apis/autoscaling/v1/namespaces/{namespace}/horizontalpodautoscalers/{name}/status	0		
get	/apis/apps/v1/namespaces/{namespace}/deployments	0 put	/apis/autoscaling/v1/namespaces/{namespace}/horizontalpodautoscalers/{name}/status	0		
post	/apis/apps/v1/namespaces/{namespace}/deployments	1 put	/apis/autoscaling/v1/namespaces/{namespace}/horizontalpodautoscalers/{name}/status	0		
watch	/apis/apps/v1/namespaces/{namespace}/deployments	1 get	/apis/batch/v1/	1		
delete	/apis/apps/v1/namespaces/{namespace}/deployments/{name}	0 get	/apis/batch/v1/namespaces/{namespace}/jobs	1		
get	/apis/apps/v1/namespaces/{namespace}/deployments/{name}	1 watch	/apis/batch/v1/namespaces/{namespace}/jobs	0		
patch	/apis/apps/v1/namespaces/{namespace}/deployments/{name}	0 get	/apis/batch/v1/namespaces/{namespace}/jobs/{name}	1		
put	/apis/apps/v1/namespaces/{namespace}/deployments/{name}	1 patch	/apis/batch/v1/namespaces/{namespace}/jobs/{name}	0		
get	/apis/apps/v1/namespaces/{namespace}/deployments/{name}/scale	0 put	/apis/batch/v1/namespaces/{namespace}/jobs/{name}	0		
patch	/apis/apps/v1/namespaces/{namespace}/deployments/{name}/scale	0 get	/apis/batch/v1/namespaces/{namespace}/jobs/{name}/status	0		
put	/apis/apps/v1/namespaces/{namespace}/deployments/{name}/scale	0 patch	/apis/batch/v1/namespaces/{namespace}/jobs/{name}/status	0		
get	/apis/apps/v1/namespaces/{namespace}/deployments/{name}/status	0 get	/apis/certificates.k8s.io/	0		
patch	/apis/apps/v1/namespaces/{namespace}/deployments/{name}/status	0 get	/apis/coordination.k8s.io/	0		
get	/apis/events.k8s.io/	1 get	/apis/events.k8s.io/	0		
post	/apis/events.k8s.io/	1 get	/apis/events.k8s.io/	0		
patch	/apis/events.k8s.io/	0 get	/apis/extensions/	0		
watch	/apis/events.k8s.io/	1 get	/apis/networking.k8s.io/	0		
get	/apis/events.k8s.io/	1 get	/apis/networking.k8s.io/v1/	1		
patch	/apis/events.k8s.io/	0 get	/apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies	1		
get	/apis/events.k8s.io/	0 post	/apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies	0		
patch	/apis/events.k8s.io/	0 watch	/apis/networking.k8s.io/v1/namespaces/{namespace}/networkpolicies	0		

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

API Coverage

v1.9.11 conformance

This was the first release we could actually measure API coverage because audit logging was turned on by default



API Coverage - graph



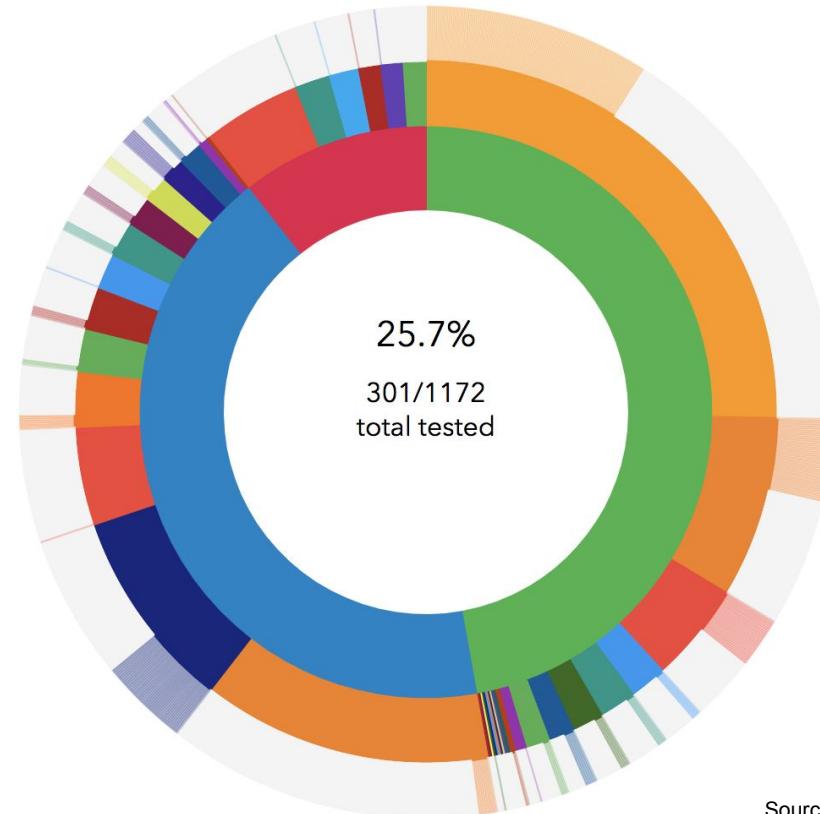
KubeCon

CloudNativeCon

China 2018

v1.10.9 conformance

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

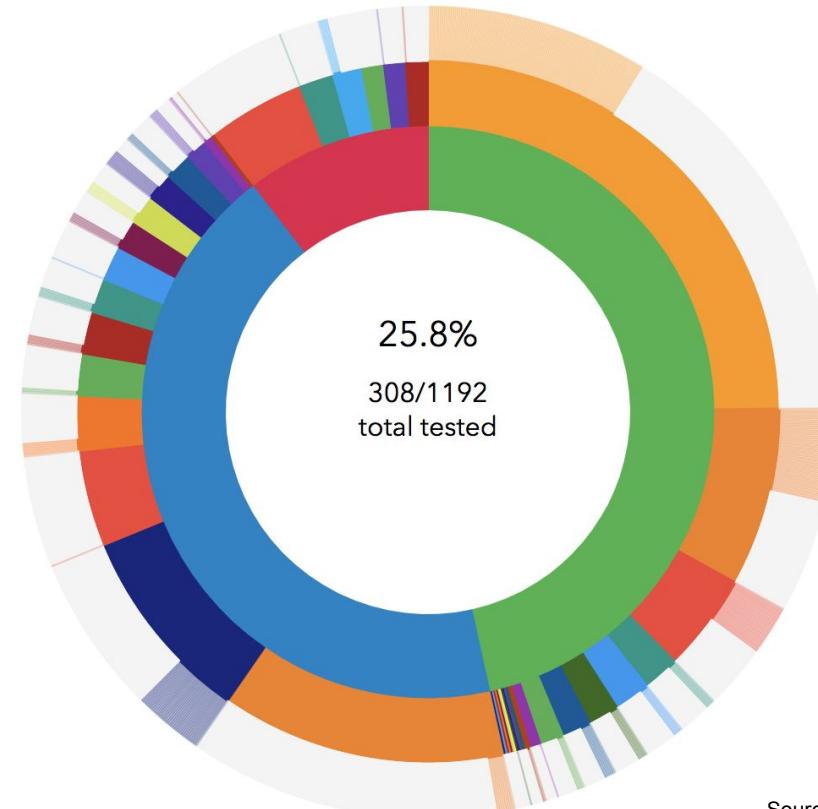


Source: apisnoop.cncf.io

API Coverage

v1.11.3 conformance

Progress stalled a bit



API Coverage



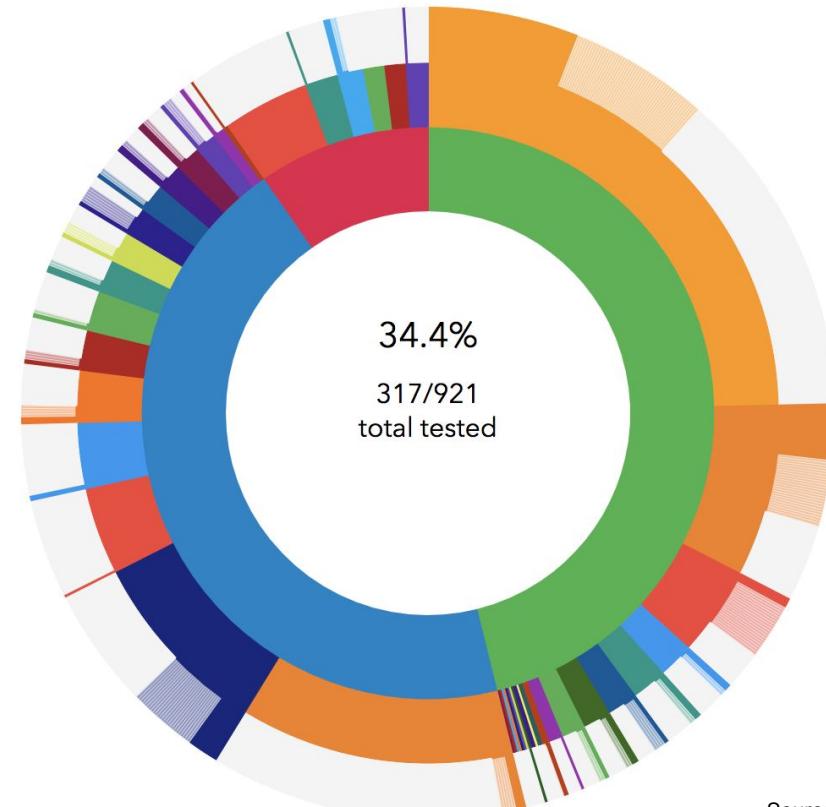
CloudNativeCon
China 2018

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

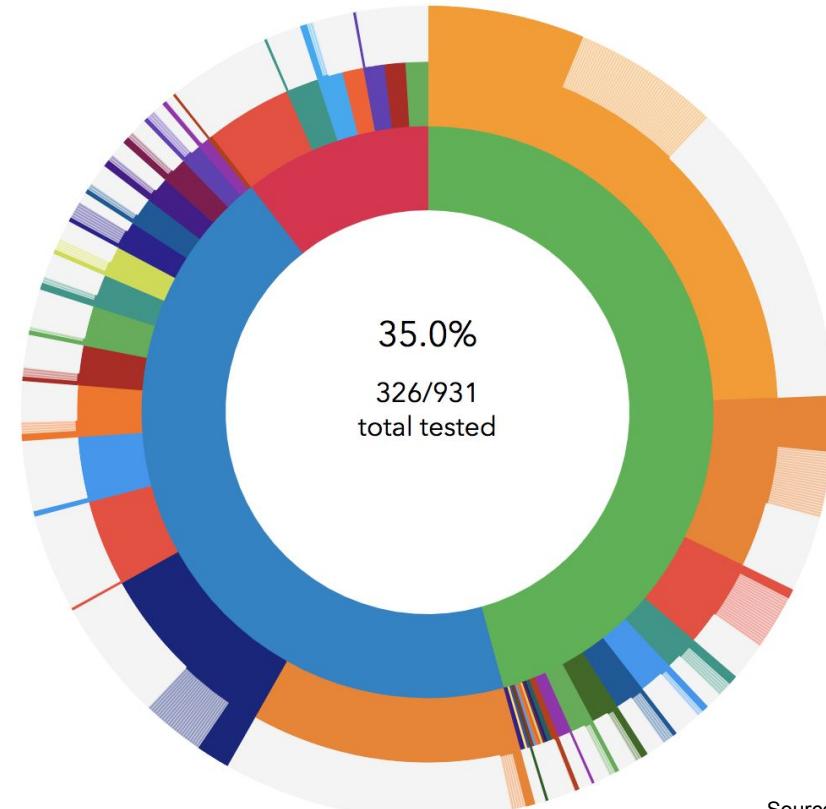


Source: apisnoop.cncf.io

API Coverage

v1.13.0 conformance

Thus far we have promoted a number of tests related to the use of kubectl, replication controllers and replicaset

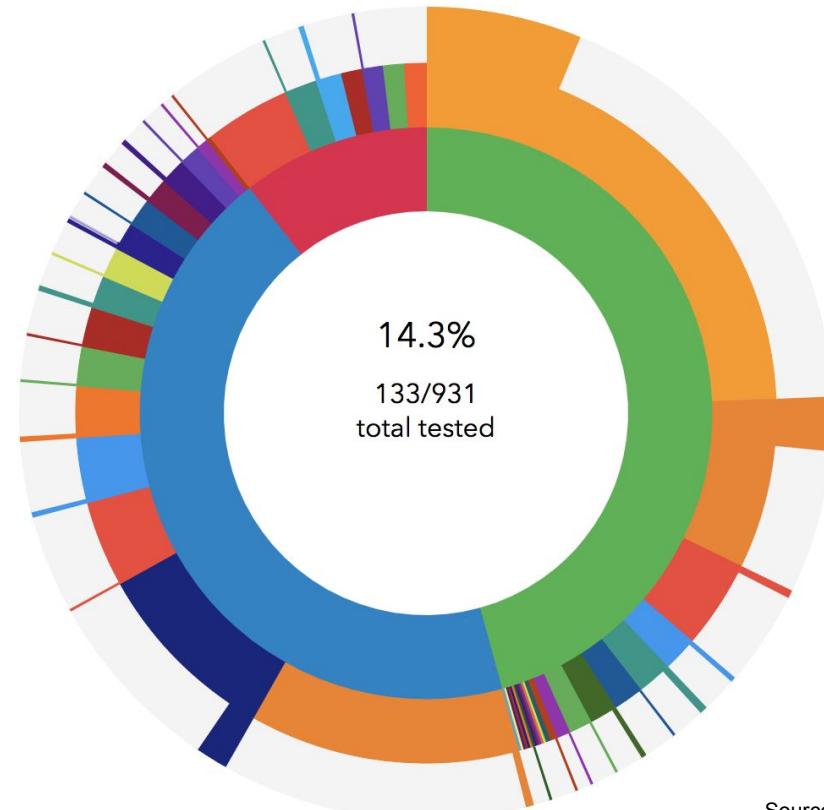


API Coverage

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



API Coverage



KubeCon

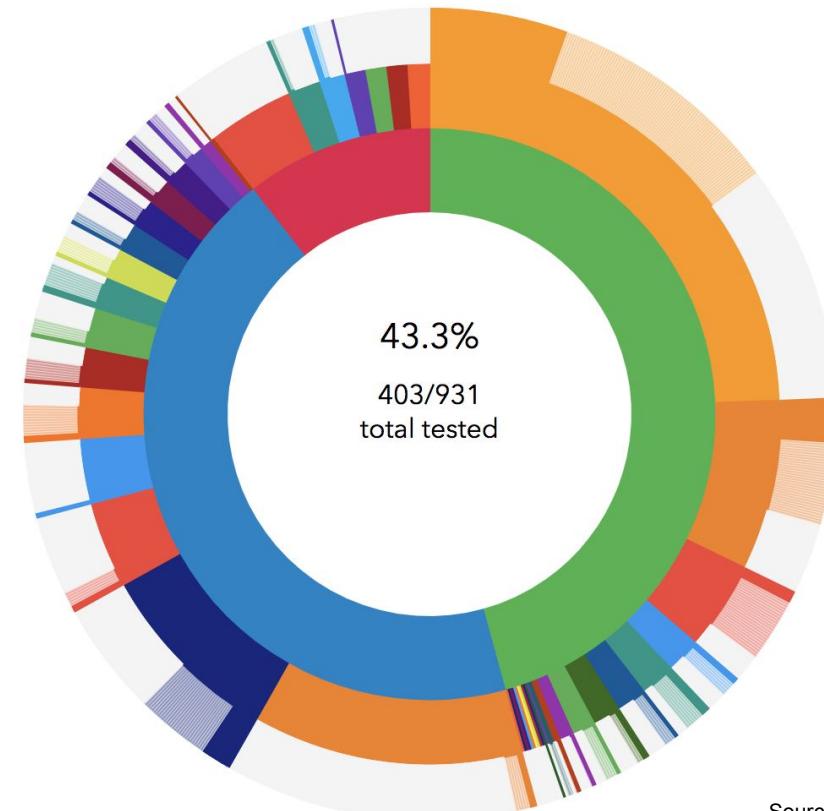
CloudNativeCon

China 2018

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



Source: apisnoop.cncf.io

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



KubeCon

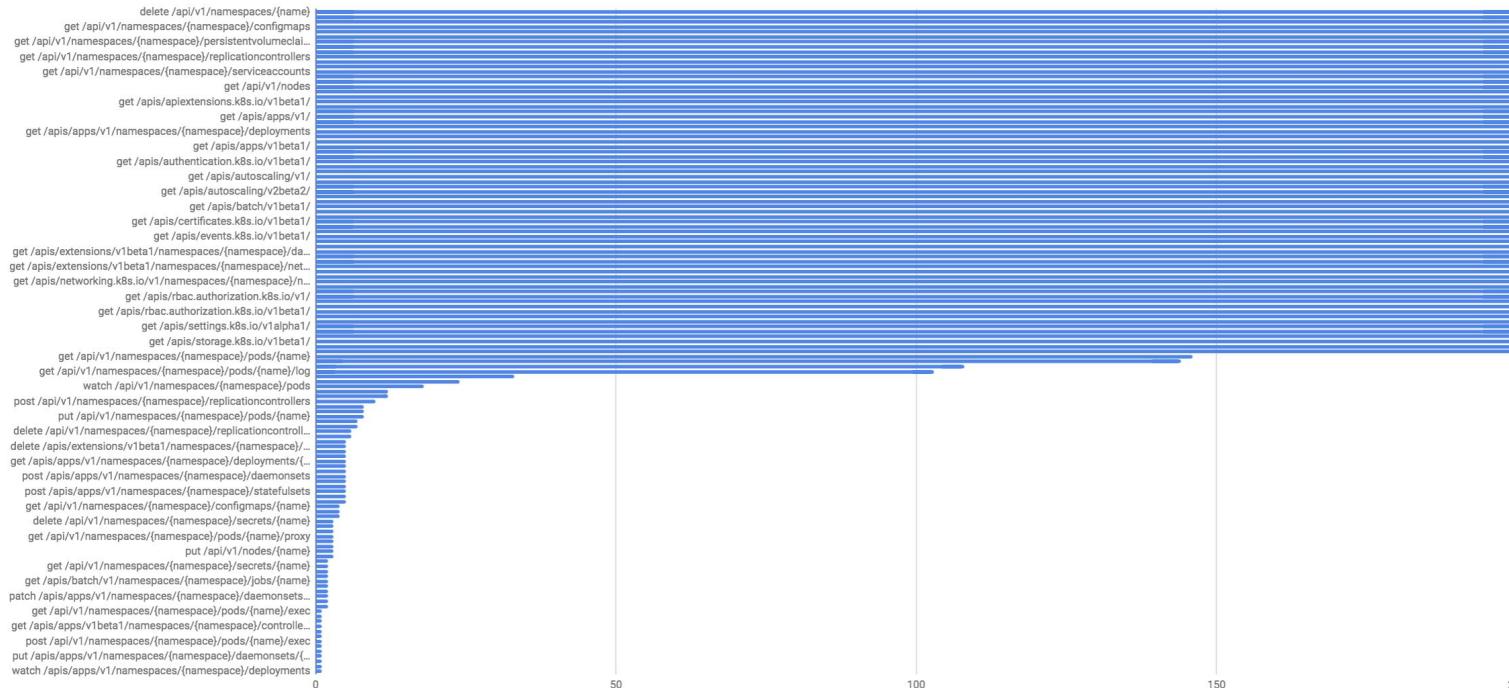


CloudNativeCon

China 2018

...

Test cases by endpoint



API Coverage



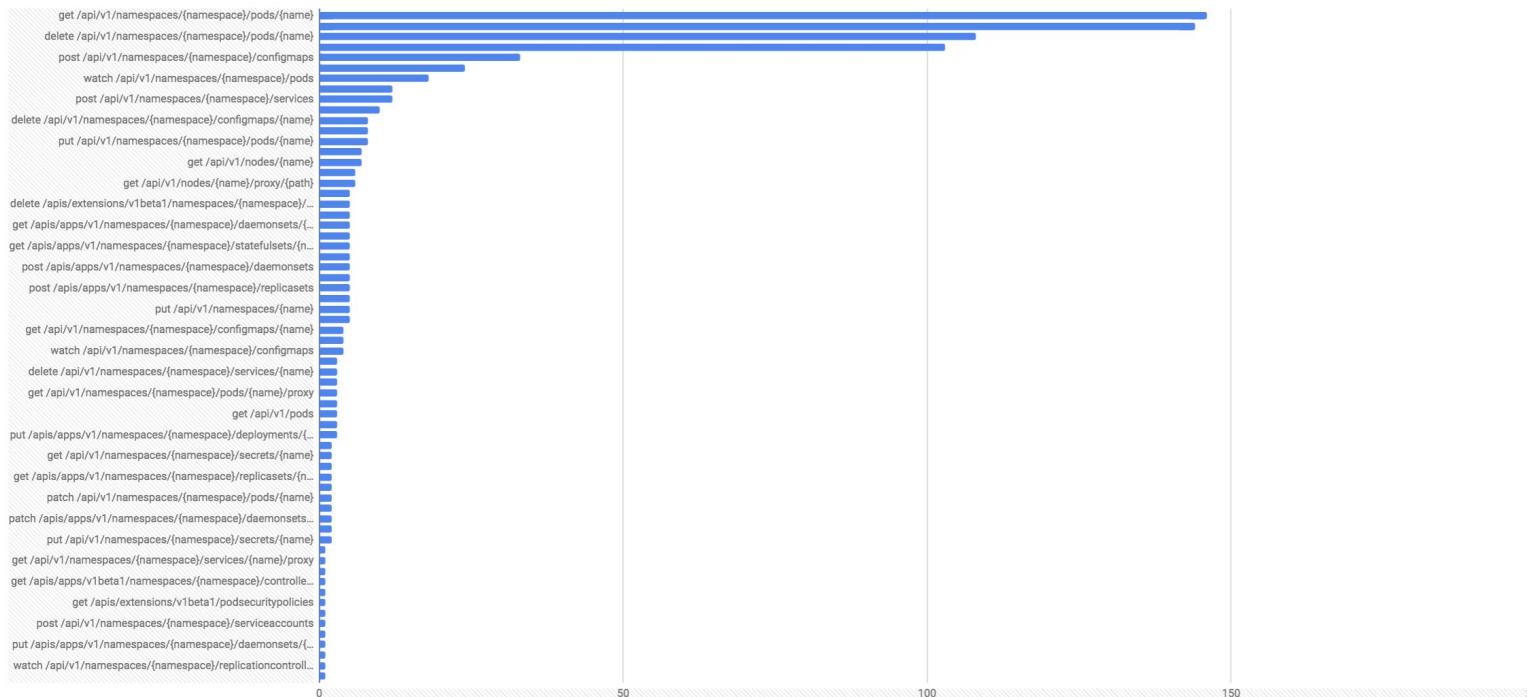
KubeCon



CloudNativeCon

China 2018

Test cases by endpoint



API Coverage

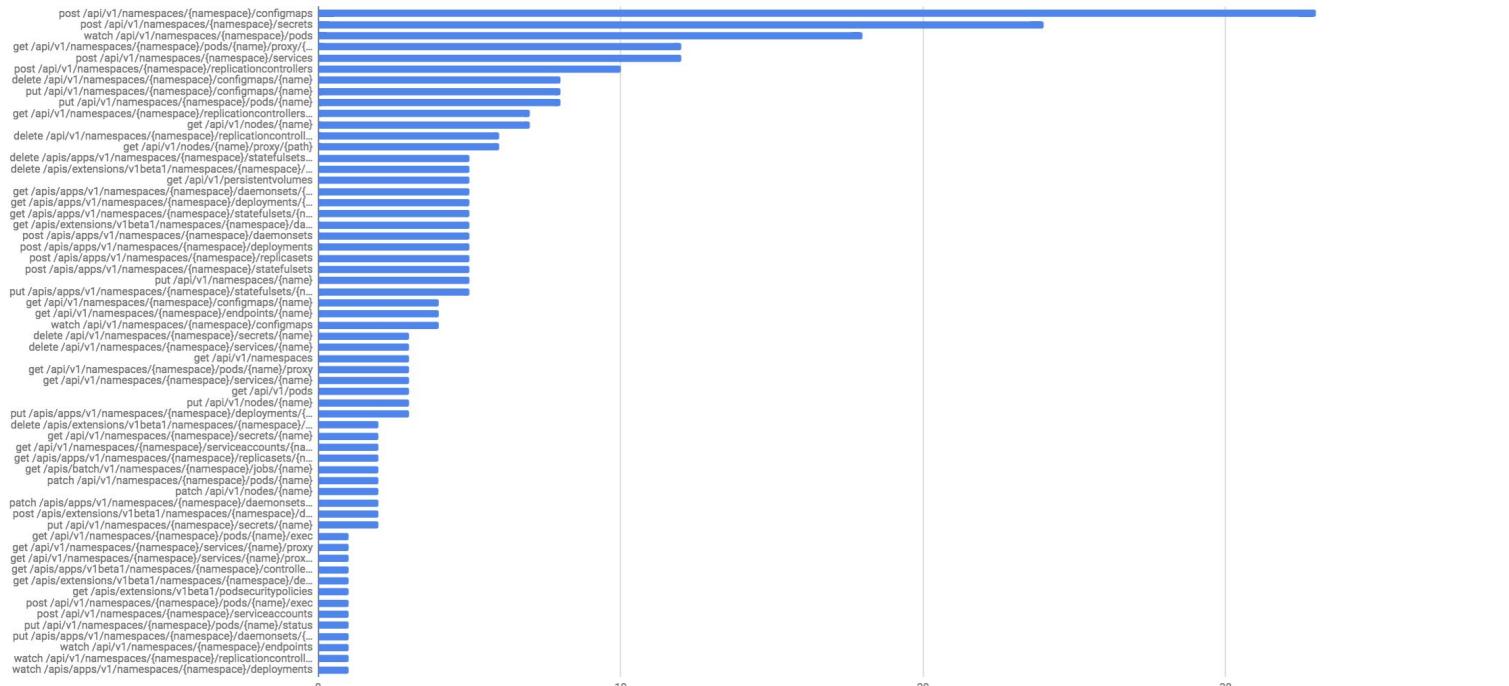


KubeCon

CloudNativeCon

China 2018

Test cases by endpoint



API Coverage - logs



KubeCon

CloudNativeCon

China 2018

[sig-network] Proxy version v1 should proxy through a service and a pod [Conformance]

```

63 framework.Conformance("should proxy through a service and a pod", func() {
64     start := time.Now()
65     label := map[string]string{"proxy-service-target": "true"}
66     service := f.ClientSet.CoreV1().Services(f.Namespace.Name).Create(&v1.Service{
67         ObjectMeta: metav1.ObjectMeta{
68             GenerationName: "proxy-service",
69         },
70     })
71     spec := v1.ServiceSpec{
72         Selector: labels,
73         Ports: []v1.ServicePort{
74             {
75                 Name: "portname1",
76                 Port: 80,
77             },
78             TargetPort: intstr.FromString("dest1"),
79         },
80         {
81             Name: "portname2",
82             Port: 81,
83         },
84         TargetPort: intstr.FromInt(162),
85     }
86     {
87         Name: "tlsportname1",
88         Port: 443,
89         TargetPort: intstr.FromString("tlsdest1"),
90     },
91     {
92         Name: "tlsportname2",
93         Port: 444,
94     },
95     TargetPort: intstr.FromInt(462),
96 },)
97     Expect(err).NotTo(HaveOccurred())
98
99 // Make an RC with a single pod. The 'porter' image is
100 // a simple server which serves the values of the
101 // environment variables below.
102 // By("starting an end server on multiple ports")
103 // pods := []v1.Pod{
104 //     cfg := testutils.RCConfig{
105 //         Labels: f.NamespaceLabels(),
106 //         Client: f.ClientSet,
107 //         InternalClient: f.InternalClientset,
108 //         Image: imageutils.GetE2EImage(imageutils.Porter),
109 //         Name: service.Name,
110 //         Namespace: f.Namespace.Name,
111 //         replicas: 1,
112 //         PollInterval: time.Second,
113 //         Env: map[string]string{
114 //             "SERVE_PORT_80": "<a href=\"/rewriteme>test</a>",
115 //             "SERVE_PORT_160": "<a href=\"/rewriteme>test</a>",
116 //             "SERVE_PORT_162": "bar",
117 //             "SERVE_PORT_443": "<a href=\"https://rewriteme>test</a>",
118 //             "SERVE_TLS_PORT_460": "tls baz",
119 //             "SERVE_TLS_PORT_462": "tls qux",
120 //         },
121 //         Ports: map[int]string{
122 //             "dest1": 160,
123 //             "dest2": 162,
124 //             "tlddest1": 460,
125 //             "tlddest2": 462,
126 //         },
127 //     }
128     ReadinessProbe: &v1.Probe{
129         Handler: v1.Handler{
130             HTTPGet: &v1.HTTPGetAction{
131                 Port: intstr.FromInt(80),
132             },
133         },
134     },
135     InitialDelaySeconds: 1,
136     TimeoutSeconds: 5,
137     PeriodSeconds: 10,
138 },
139 Labels: labels,
140 })
141 var proxyMethods = []string{"GET", "POST", "PUT", "PATCH", "DELETE"}
142
143 func TestProxyThroughService(t *testing.T) {
144     ctx := context.Background()
145     proxy := proxy.NewProxy(&http.Transport{TLSHandshakeTimeout: 10 * time.Second}, &v1.ClientSet, &v1.NamespaceList{}, 10, true, true, false)
146
147     log := logrus.New()
148     log.SetLevel(logrus.DebugLevel)
149     defer log.Close()
150
151     t.RunParallel(func(t *testing.T) {
152         f := &framework.Fixture{
153             Label: label,
154             InitialDelaySeconds: 1,
155             TimeoutSeconds: 5,
156             PeriodSeconds: 10,
157         },
158
159         createdPods := &pods{
160             1,
161             Expect(framework.RunRc(cfg)).NotTo(HaveOccurred()),
162             defer framework.DeleteRCandWaitForGCf(clientSet, f.Namespace.Name, cfg.Name),
163             Expect(framework.WaitForEndpoint(f.ClientSet, f.Namespace.Name, service.Name)).NotTo(HaveError),
164             expectFrameworkExpectations("v1", "services", "ports", "targets"),
165             // table constructors
166             // Try proxying through the service and directly to the pod.
167             subresourceServiceProxyURL := func(scheme, port string) string {
168                 return "http://" + namespaces[0].Name + "/services/" + net.JoinSchemeNamePort(scheme, port) + "/proxy"
169             },
170             subresourcePodProxyURL := func(scheme, port string) string {
171                 return prefix + "/namespaces/" + f.Namespace.Name + "/pods/" + net.JoinSchemeNamePort(scheme, port) + "/proxy"
172         }
173     }
174
175     // construct the table
176     expectations := map[string]string{
177         subresourceServiceProxyURL("http", "portname1") + "/": "foo",
178         subresourceServiceProxyURL("http", "portname1") + "/": "foo",
179         subresourceServiceProxyURL("http", "portname2") + "/": "bar",
180         subresourceServiceProxyURL("http", "portname2") + "/": "bar",
181         subresourceServiceProxyURL("https", "tlsportname1") + "/": "tls baz",
182         subresourceServiceProxyURL("https", "tlsportname1") + "/": "tls qux",
183         subresourcePodProxyURL("http", "") + "/": <a href="" + subresourcePodProxyURL("http", ""),
184             test"/>",
185         subresourcePodProxyURL("http", "1088") + "/": <a href="" + subresourcePodProxyURL("http", ""),
186             eme"/>test",
187         subresourcePodProxyURL("http", "1089") + "/": <a href="" + subresourcePodProxyURL("http", ""),
188         writtenme"/>test",
189         subresourcePodProxyURL("http", "1609") + "/": "foo",
190         subresourcePodProxyURL("http", "1609") + "/": "foo",
191         subresourcePodProxyURL("http", "1621") + "/": "bar",
192         subresourcePodProxyURL("http", "1621") + "/": "bar",
193         subresourcePodProxyURL("https", "443") + "/": <a href="" + subresourcePodProxyURL("https",
194             rewrite"/>test"/>",
195         subresourcePodProxyURL("https", "4669") + "/": "tls baz",
196         subresourcePodProxyURL("https", "4669") + "/": "tls baz",
197         // TODO: sniff�px) This is a lot of copy/paste of the above, can we refactor
198         subresourceServiceProxyURL("http", "portname1") + "/im/ar/meeseks/look/at/me": "foo",
199         subresourceServiceProxyURL("http", "portname1") + "/im/ar/meeseks/look/at/me": "bar",
200         subresourceServiceProxyURL("http", "portname2") + "/im/ar/meeseks/look/at/me": "foo",
201         subresourceServiceProxyURL("http", "portname2") + "/im/ar/meeseks/look/at/me": "bar",
202         subresourcePodProxyURL("https", "4622") + "/": im/ar/meeseks/look/at/me",
203         subresourcePodProxyURL("https", "4622") + "/": im/ar/meeseks/look/at/me",
204         // TODO: below entries don't work, but I believe we should make them work.
205         podPrefix + "/dest1": "foo",
206         podPrefix + "/dest2": "bar",
207         podPrefix + "/dest3": "bar",
208
209     }
210
211     wr := sync.WaitGroup{}
212     errs := []string{
213         errlock := sync.Mutex{}})
214     recorder := klog.NewRecorder(framework.CreateString(), &wr)
215     recordErrlock := sync.Mutex{}
216     defer errlock.Unlock()
217     errs = appenderrs, s
218
219     // TODO(spiffyxx): OPTIONS and HEAD should work but fail, or don't hit the audit log, why?
220     var proxyMethods = []string{"GET"}
221
222     d := time.Since(start)
223     framework.Log("setup took %v, starting test cases", d)
224     numberTestCases := len(expectations) * len(proxyMethods)
225     totalAttempts := numberTestCases * proxyAttempts
226     By(fmt.Sprintf("running %v cases, %v attempts per case, %v total attempts", numberTestCases,
227     totalAttempts))
228
229     // var proxyMethods = []string("GET")
230     d := time.Since(start)
231     framework.Log("setup took %v, starting test cases", d)
232     numberTestCases := len(expectations) * len(proxyMethods)
233     totalAttempts := numberTestCases * proxyAttempts
234     By(fmt.Sprintf("running %v cases, %v attempts per case, %v total attempts", numberTestCases,
235     totalAttempts))
236
237     for i := 0; i < proxyAttempts; i++ {
238         wg.Add(len(testCases))
239         for path, val := range expectations {
240             for _, verb := range proxyMethods {
241                 g := func(int, verb, path, val string) {
242                     doProxy(f, verb, path, i)
243                     // this runs the test case
244                     body, status, d, err := doProxy(f, verb, path, i)
245                     if err != nil {
246                         if serr, ok := err.(errors.StatusError); ok {
247                             recordError(fmt.Sprintf("%v (%v; %v) verb %v path %v gave status error: %v", i, verb, path, status, d, err))
248                         } else {
249                             recordError(fmt.Sprintf("%v: verb %v path %v gave error: %v", i, verb, path, err))
250                         }
251                     }
252                     return
253                 }
254                 if status != http.StatusOK {
255                     recordError(fmt.Sprintf("%v: verb %v path %v gave status: %v", i, verb, path, status))
256                 } else if a, b := val, string(body); a == b {
257                     recordError(fmt.Sprintf("%v: verb %v path %v wanted %v, got %v", i, verb, path, a, b))
258                 }
259             }
260             if d > proxyHTTPCallTimeout {
261                 recordError(fmt.Sprintf("%v: verb %v path %v took %v > %v", i, verb, path, d, proxyHTTPCallTimeout))
262             }
263         }
264         wg.Wait()
265
266         if lenerrs) != 0 {
267             body, err := f.ClientSet.CoreV1().Pods(f.Namespace.Name).GetLogs(pods[0].Name, &v1.PodLogOptions{Container: "porter", Follow: true})
268             if err != nil {
269                 framework.Log("Error getting logs for pod %s: %v", pods[0].Name, err)
270             } else {
271                 framework.Log("Pod %s has the following error logs: %s", pods[0].Name, body)
272             }
273         }
274     }
275
276     func doProxy(f *framework.Framework, verb, path string, i int) (body []byte, statusCode int, d time.Duration) {
277         // About all of the proxy access in this file
278         // ends with /dowbar because it preserves the trailing '/'.
279         // +Do(.Rawl) is used (instead of Dowbar()) because it will turn an
280         // error from apiserver proxy into an actual error, and there is no
281         // chance of the things we are talking to being confused for an error
282         // that they would have emitted.
283         start := time.Now()
284         body, err := f.ClientSet.CoreV1().RESTClient().Verb(verb).AbsPath(path).Do().StatusCode(&statusCode)
285         d = time.Since(start)
286         if lenbody) > 0 {
287             framework.Log("%v: %s (%v; %v) path, %v, body=%v", i, verb, path, statusCode, body)
288         } else {
289             framework.Log("%v: %s (%v; %v) path, no body, statusCode, %v", i, verb, path, statusCode, d)
290         }
291     }
292     return
293 }
```

API Coverage



```
2018-11-06T14:42:06.676909Z GET /api?timeout=32s
2018-11-06T14:42:06.678291Z GET /apis?timeout=32s
2018-11-06T14:42:06.679550Z GET /api/v1?timeout=32s
2018-11-06T14:42:06.681464Z GET /apis/apiregistration.k8s.io/v1?timeout=32s
2018-11-06T14:42:06.682795Z GET /apis/apiregistration.k8s.io/v1beta1?timeout=32s
2018-11-06T14:42:06.684231Z GET /apis/extensions/v1beta1?timeout=32s
2018-11-06T14:42:06.685915Z GET /apis/apps/v1?timeout=32s
2018-11-06T14:42:06.687592Z GET /apis/apps/v1beta2?timeout=32s
2018-11-06T14:42:06.689808Z GET /apis/apps/v1beta1?timeout=32s
2018-11-06T14:42:06.691675Z GET /apis/events.k8s.io/v1beta1?timeout=32s
2018-11-06T14:42:06.692827Z GET /apis/authentication.k8s.io/v1?timeout=32s
2018-11-06T14:42:06.693966Z GET /apis/authentication.k8s.io/v1beta1?timeout=32s
2018-11-06T14:42:06.695125Z GET /apis/authorization.k8s.io/v1?timeout=32s
2018-11-06T14:42:06.696394Z GET /apis/authorization.k8s.io/v1beta1?timeout=32s
2018-11-06T14:42:06.697963Z GET /apis/autoscaling/v1?timeout=32s
2018-11-06T14:42:06.699445Z GET /apis/autoscaling/v2beta1?timeout=32s
2018-11-06T14:42:06.701260Z GET /apis/autoscaling/v2beta2?timeout=32s
2018-11-06T14:42:06.702736Z GET /apis/batch/v1?timeout=32s
2018-11-06T14:42:06.704111Z GET /apis/batch/v1beta1?timeout=32s
2018-11-06T14:42:06.705776Z GET /apis/batch/v2alpha1?timeout=32s
2018-11-06T14:42:06.706999Z GET /apis/certificates.k8s.io/v1beta1?timeout=32s
2018-11-06T14:42:06.708305Z GET /apis/networking.k8s.io/v1?timeout=32s
2018-11-06T14:42:06.709665Z GET /apis/policy/v1beta1?timeout=32s
2018-11-06T14:42:06.710833Z GET /apis/rbac.authorization.k8s.io/v1?timeout=32s
2018-11-06T14:42:06.712122Z GET /apis/rbac.authorization.k8s.io/v1beta1?timeout=32s
2018-11-06T14:42:06.713477Z GET /apis/settings.k8s.io/v1alpha1?timeout=32s
2018-11-06T14:42:06.715456Z GET /apis/storage.k8s.io/v1?timeout=32s
2018-11-06T14:42:06.716833Z GET /apis/storage.k8s.io/v1beta1?timeout=32s
2018-11-06T14:42:06.718159Z GET /apis/admissionregistration.k8s.io/v1beta1?timeout=32s
2018-11-06T14:42:06.720108Z GET /apis/apiextensions.k8s.io/v1beta1?timeout=32s
2018-11-06T14:42:06.721520Z GET /apis/scheduling.k8s.io/v1beta1?timeout=32s
2018-11-06T14:42:06.722741Z GET /apis/scheduling.k8s.io/v1alpha1?timeout=32s
2018-11-06T14:42:06.723921Z GET /apis/coordination.k8s.io/v1beta1?timeout=32s
2018-11-06T14:42:06.725073Z GET /apis/scalingpolicy.kope.io/v1alpha1?timeout=32s
2018-11-06T14:42:06.727031Z GET /apis/metrics.k8s.io/v1beta1?timeout=32s
```

API Coverage



```
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
...
2018-11-06T14:42:15.838371Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/endpoints/proxy-service-8t7gj
2018-11-06T14:42:15.874337Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods/http:proxy-service-8t7gj-wgqpz:1080/proxy/
2018-11-06T14:42:15.874337Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods/http:proxy-service-8t7gj-wgqpz:1080/proxy/
2018-11-06T14:42:15.848967Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods/proxy-service-8t7gj-wgqpz:162/proxy/
2018-11-06T14:42:15.848967Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods/proxy-service-8t7gj-wgqpz:162/proxy/
2018-11-06T14:42:15.866811Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/services/http:proxy-service-8t7gj:portname1/proxy/
2018-11-06T14:42:15.866811Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/services/http:proxy-service-8t7gj:portname1/proxy/
2018-11-06T14:42:15.873896Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods/proxy-service-8t7gj-wgqpz:160/proxy/
2018-11-06T14:42:15.873896Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods/proxy-service-8t7gj-wgqpz:160/proxy/
2018-11-06T14:42:15.870038Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods/proxy-service-8t7gj-wgqpz/proxy/
2018-11-06T14:42:15.870038Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods/proxy-service-8t7gj-wgqpz/proxy/
2018-11-06T14:42:15.861357Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods/http:proxy-service-8t7gj-wgqpz:160/proxy/
2018-11-06T14:42:15.856677Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/services/proxy-service-8t7gj:portname2/proxy/
2018-11-06T14:42:15.861357Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods/http:proxy-service-8t7gj-wgqpz:160/proxy/
2018-11-06T14:42:15.856677Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/services/proxy-service-8t7gj:portname2/proxy/
2018-11-06T14:42:15.857196Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods/proxy-service-8t7gj-wgqpz:1080/proxy/
2018-11-06T14:42:15.857196Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods/proxy-service-8t7gj-wgqpz:1080/proxy/
2018-11-06T14:42:15.869212Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/services/http:proxy-service-8t7gj:portname2/proxy/
2018-11-06T14:42:15.869212Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/services/http:proxy-service-8t7gj:portname2/proxy/
2018-11-06T14:42:15.869212Z GET /api/v1/namespaces/e2e-tests-proxy-gvqt4/services/http:proxy-service-8t7gj:portname2/proxy/
...

```

API Coverage



```
2018-11-06T14:42:16.467844Z LIST /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods?labelSelector=name%3Dproxy-service-8t7gj&limit=500&resourceVersion=0
2018-11-06T14:42:16.469716Z WATCH /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods?labelSelector=name%3Dproxy-service-8t7gj&resourceVersion=11586&timeoutSeconds=478&watch=true
2018-11-06T14:42:16.518353Z DELETE /api/v1/namespaces/e2e-tests-proxy-gvqt4/replicationcontrollers/proxy-service-8t7gj
2018-11-06T14:42:16.469716Z WATCH /api/v1/namespaces/e2e-tests-proxy-gvqt4/pods?labelSelector=name%3Dproxy-service-8t7gj&resourceVersion=11586&timeoutSeconds=478&watch=true
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
2018-11-06T14:42:25.642996Z 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

API Coverage - Next

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?

```
k8s.io/kubernetes/pkg/apis/core/v1/helper/helpers.go (49.8%)
not tracked    low coverage * * * * * * * * * * high coverage
}

// TODO: make method on LoadBalancerStatus?
func LoadBalancerStatusDeepCopy(lb v1.LoadBalancerStatus) *v1.LoadBalancerStatus {
    c := &v1.LoadBalancerStatus{}
    c.Ingress = make([]v1.LoadBalancerIngress, len(lb.Ingress))
    for i := range lb.Ingress {
        c.Ingress[i] = lb.Ingress[i]
    }
    return c
}

// GetAccessModesAsString returns a string representation of an array of access modes.
// modes, when present, are always in the same order: RWO,ROX,RWX.
func GetAccessModesAsString(modes []v1.PersistentVolumeAccessMode) string {
    modes = removeDuplicateAccessModes(modes)
    modesStr := ""
    if containsAccessMode(modes, v1.ReadWriteOnce) {
        modesStr = append(modesStr, "RWO")
    }
    if containsAccessMode(modes, v1.ReadOnlyMany) {
        modesStr = append(modesStr, "ROX")
    }
    if containsAccessMode(modes, v1.ReadWriteMany) {
        modesStr = append(modesStr, "RWX")
    }
    return strings.Join(modesStr, ",")
}

// GetAccessModesAsString returns an array of AccessModes from a string created by GetAccessModesAsString
func GetAccessModesFromString(modesString) []v1.PersistentVolumeAccessMode {
    modes := strings.Split(modesString, ",")
    accessModes := []v1.PersistentVolumeAccessMode{}
    for _, s := range modes {
        strings.TrimSpace(s)
        switch {
        case s == "RWO":
            accessModes = append(accessModes, v1.ReadWriteOnce)
        case s == "ROX":
            accessModes = append(accessModes, v1.ReadOnlyMany)
        case s == "RWX":
            accessModes = append(accessModes, v1.ReadWriteMany)
        }
    }
    return accessModes
}

// removeDuplicateAccessModes returns an array of access modes without any duplicates
func removeDuplicateAccessModes(modes []v1.PersistentVolumeAccessMode) []v1.PersistentVolumeAccessMode {
    accessModes := []v1.PersistentVolumeAccessMode{}
    for _, m := range modes {
        if !containsAccessMode(accessModes, m) {
            accessModes = append(accessModes, m)
        }
    }
    return accessModes
}

func containsAccessMode(modes []v1.PersistentVolumeAccessMode, mode v1.PersistentVolumeAccessMode) bool {
    for _, m := range modes {
        if m == mode {
            return true
        }
    }
    return false
}

k8s.io/kubernetes/pkg/apis/core/v1/helper/helpers.go (33.0%)
not tracked    low coverage * * * * * * * * * * high coverage
}

// TODO: make method on LoadBalancerStatus?
func LoadBalancerStatusDeepCopy(lb v1.LoadBalancerStatus) *v1.LoadBalancerStatus {
    c := &v1.LoadBalancerStatus{}
    c.Ingress = make([]v1.LoadBalancerIngress, len(lb.Ingress))
    for i := range lb.Ingress {
        c.Ingress[i] = lb.Ingress[i]
    }
    return c
}

// GetAccessModesAsString returns a string representation of an array of access modes.
// modes, when present, are always in the same order: RWO,ROX,RWX.
func GetAccessModesAsString(modes []v1.PersistentVolumeAccessMode) string {
    modesStr := []string{}
    if containsAccessMode(modes, v1.ReadWriteOnce) {
        modesStr = append(modesStr, "RWO")
    }
    if containsAccessMode(modes, v1.ReadOnlyMany) {
        modesStr = append(modesStr, "ROX")
    }
    if containsAccessMode(modes, v1.ReadWriteMany) {
        modesStr = append(modesStr, "RWX")
    }
    return strings.Join(modesStr, ",")
}

// GetAccessModesAsString returns an array of AccessModes from a string created by GetAccessModesAsString
func GetAccessModesFromString(modesString) []v1.PersistentVolumeAccessMode {
    modes := strings.Split(modesString, ",")
    accessModes := []v1.PersistentVolumeAccessMode{}
    for _, s := range modes {
        strings.TrimSpace(s)
        switch {
        case s == "RWO":
            accessModes = append(accessModes, v1.ReadWriteOnce)
        case s == "ROX":
            accessModes = append(accessModes, v1.ReadOnlyMany)
        case s == "RWX":
            accessModes = append(accessModes, v1.ReadWriteMany)
        }
    }
    return accessModes
}

// removeDuplicateAccessModes returns an array of access modes without any duplicates
func removeDuplicateAccessModes(modes []v1.PersistentVolumeAccessMode) []v1.PersistentVolumeAccessMode {
    accessModes := []v1.PersistentVolumeAccessMode{}
    for _, m := range modes {
        if !containsAccessMode(accessModes, m) {
            accessModes = append(accessModes, m)
        }
    }
    return accessModes
}

func containsAccessMode(modes []v1.PersistentVolumeAccessMode, mode v1.PersistentVolumeAccessMode) bool {
    for _, m := range modes {
        if m == mode {
            return true
        }
    }
    return false
}
```

Line Coverage



```
gopherage html file.cov # which packages are covered?
```

Line Coverage



KubeCon



CloudNativeCon

China 2018

Coverage summary

[k8s.io/](#)

File	e2e	conformance
kubernetes/	▲ 33.5%	28.5%

Coverage summary

[k8s.io/kubernetes/](#)

File	e2e	conformance
pkg/	▲ 37.6%	25.5%
plugin/	▲ 32.6%	16.1%
vendor/	▼ 31.0%	31.3%

Coverage summary

[k8s.io/kubernetes/pkg/](#)

File	e2e	conformance
api/	▲ 49.9%	46.6%
apis/	▲ 47.4%	34.2%
auth/	9.8%	9.8%
capabilities/	36.4%	36.4%
client/	▲ 2.2%	1.0%
controller/	▲ 50.2%	31.5%
credentialprovider/	▲ 28.6%	23.7%
features/	0.0%	0.0%
fieldpath/	51.4%	51.4%
kubeapiserver/	▲ 3.8%	2.6%
kubectl/	0.0%	0.0%
kubelet/	▲ 43.0%	40.7%
master/	16.4%	16.4%
printers/	▲ 4.0%	0.0%
probe/	▲ 61.2%	59.2%
proxy/	▲ 18.4%	16.8%
quota/	▲ 70.4%	7.3%
registry/	▲ 39.0%	24.6%
routes/	▲ 45.5%	0.0%
scheduler/	▲ 41.1%	32.8%
security/	▲ 55.3%	1.0%
securitycontext/	▲ 51.2%	15.7%
serviceaccount/	▲ 35.8%	34.1%
ssh/	0.0%	0.0%
util/	▲ 31.0%	20.5%
version/	3.1%	3.1%
volume/	▲ 21.6%	7.6%

Coverage summary

[k8s.io/kubernetes/plugin/pkg/](#)

File	e2e	conformance
admission/	▲ 32.2%	13.4%
auth/	▲ 34.1%	26.1%

Coverage summary

[k8s.io/kubernetes/vendor/k8s.io/](#)

File	e2e	conformance
api/	▲ 31.2%	29.8%
apiextensions-apiserver/	▼ 36.0%	38.2%
apimachinery/	▲ 43.8%	42.0%
apiserver/	▼ 24.1%	27.3%
cli-runtime/	0.0%	0.0%
client-go/	▼ 26.3%	27.1%
cloud-provider/	▲ 17.5%	8.8%
cluster-bootstrap/	0.0%	0.0%
csi-api/	0.0%	0.0%
kube-aggregator/	▼ 49.2%	50.0%
kube-controller-manager/	0.0%	
kube-openapi/	33.6%	33.6%
kube-proxy/	0.0%	
kube-scheduler/	0.0%	
kubelet/	0.0%	
metrics/	▲ 6.2%	2.1%
utils/	▲ 55.6%	52.4%

Line Coverage

Coverage summary

k8s.io/kubernetes/pkg/

File	325	315	305	295	285	275	265	255
api/	48.5%	▼ 48.5%	▲ 48.7%	▼ 46.7%	▲ 48.7%	▼ 48.2%	▲ 48.8%	48.7%
apis/	33.8%	▲ 33.8%	▼ 33.7%	33.7%	▲ 33.7%	▼ 33.7%	▲ 33.7%	33.7%
auth/	9.8%	9.8%	9.8%	9.8%	9.8%	9.8%	9.8%	9.8%
capabilities/	36.4%	36.4%	36.4%	36.4%	36.4%	36.4%	36.4%	36.4%
client/	1.0%	1.0%	▲ 1.0%	0.6%	0.6%	▲ 0.6%	0.6%	0.6%
controller/	▲ 31.3%	▲ 31.3%	▼ 31.1%	▼ 31.2%	▲ 31.3%	▲ 31.2%	▼ 31.2%	31.2%
credentialprovider/	23.7%	23.7%	23.7%	23.7%	23.7%	23.7%	23.7%	23.7%
features/	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
fieldpath/	51.4%	51.4%	51.4%	51.4%	51.4%	51.4%	51.4%	51.4%
kubeapiserver/	2.6%	▼ 2.6%	2.6%	2.6%	2.6%	▼ 2.6%	2.8%	2.8%
kubectl/	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
kubelet/	▼ 40.6%	▲ 40.7%	▼ 40.4%	▼ 40.5%	▲ 40.7%	▲ 40.6%	▼ 40.5%	40.7%
master/	16.4%	16.4%	▲ 16.4%	16.3%	16.3%	16.3%	16.3%	16.3%
printers/	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
probe/	59.2%	59.2%	59.2%	59.2%	59.2%	59.2%	59.2%	59.2%
proxy/	16.8%	16.8%	▼ 16.8%	16.9%	16.9%	16.9%	16.9%	16.9%
quota/	7.1%	7.1%	7.1%	7.1%	7.1%	7.1%	7.1%	7.1%
registry/	24.6%	24.6%	▼ 24.6%	▲ 24.7%	▲ 24.7%	▼ 24.6%	▲ 24.7%	24.6%
routes/	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
scheduler/	▲ 32.6%	▼ 31.9%	▼ 32.6%	▲ 32.8%	▼ 32.3%	▲ 32.5%	▼ 32.1%	32.3%
security/	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%	1.0%
securitycontext/	15.7%	15.7%	15.7%	15.7%	15.7%	15.7%	15.7%	15.7%
serviceaccount/	34.1%	34.1%	34.1%	34.1%	34.1%	34.1%	34.1%	34.1%
ssh/	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
util/	20.4%	▼ 20.4%	▼ 20.4%	20.6%	▲ 20.6%	20.4%	20.4%	20.4%
version/	3.1%	3.1%	3.1%	3.1%	3.1%	3.1%	3.1%	3.1%
volume/	▲ 7.7%	▼ 7.6%	▼ 7.6%	▼ 7.6%	▲ 7.7%	▲ 7.7%	▲ 7.6%	7.6%

Line Coverage

Coverage summary

k8s.io/kubernetes/pkg/kubelet/dockershim/

File	e2e	conformance
cm/	5 / 40 (12.5%)	5 / 40 (12.5%)
convert.go	40 / 53 (75.5%)	40 / 53 (75.5%)
docker_checkpoint.go	6 / 6 (100.0%)	6 / 6 (100.0%)
docker_container.go	▲ 136 / 173 (78.6%)	132 / 173 (76.3%)
docker_image.go	44 / 71 (62.0%)	44 / 71 (62.0%)
docker_image_linux.go	0 / 1 (0.0%)	0 / 1 (0.0%)
docker_legacy_service.go	5 / 28 (17.9%)	5 / 28 (17.9%)
docker_logs.go	0 / 1 (0.0%)	0 / 1 (0.0%)
docker_sandbox.go	▲ 238 / 296 (80.4%)	221 / 296 (74.7%)
docker_service.go	▲ 53 / 155 (34.2%)	50 / 155 (32.3%)
docker_stats_linux.go	0 / 2 (0.0%)	0 / 2 (0.0%)
docker_streaming.go	▲ 65 / 79 (82.3%)	21 / 79 (26.6%)
exec.go	▲ 27 / 38 (71.1%)	25 / 38 (65.8%)
helpers.go	▲ 93 / 140 (66.4%)	81 / 140 (57.9%)
helpers_linux.go	▲ 27 / 44 (61.4%)	25 / 44 (56.8%)
libdocker/	▲ 274 / 906 (30.2%)	248 / 906 (27.4%)
metrics/	1 / 6 (16.7%)	1 / 6 (16.7%)
naming.go	19 / 27 (70.4%)	19 / 27 (70.4%)
network/	▼ 316 / 1012 (31.2%)	318 / 1012 (31.4%)
remote/	0 / 15 (0.0%)	0 / 15 (0.0%)
security_context.go	▲ 64 / 83 (77.1%)	59 / 83 (71.1%)
selinux_util.go	12 / 12 (100.0%)	12 / 12 (100.0%)

Line Coverage

Coverage summary

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

File	e2e	conformance
prober.go	▲ 65 / 95 (68.4%)	62 / 95 (65.3%)
prober_manager.go	61 / 73 (83.6%)	61 / 73 (83.6%)
results/	20 / 26 (76.9%)	20 / 26 (76.9%)
worker.go	▲ 64 / 68 (94.1%)	62 / 68 (91.2%)

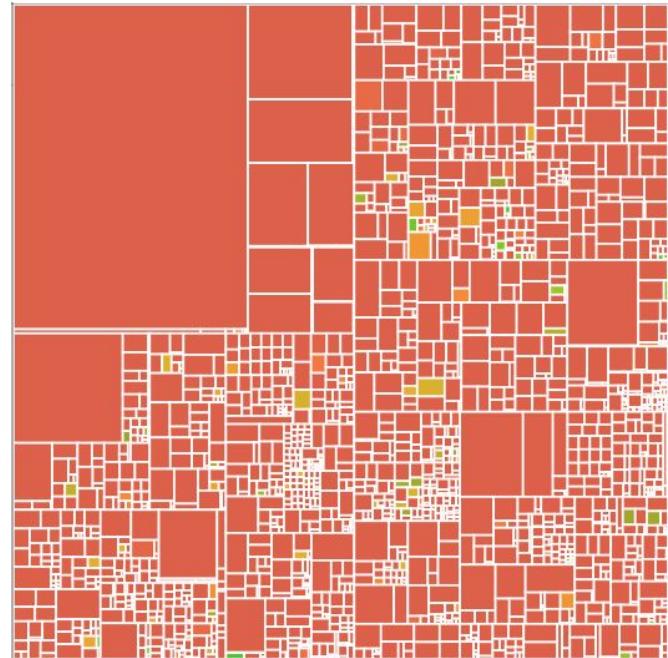
Coverage summary

k8s.io/kubernetes/pkg/probe/

File	e2e	conformance
exec/	▲ 9 / 11 (81.8%)	8 / 11 (72.7%)
http/	21 / 29 (72.4%)	21 / 29 (72.4%)
tcp/	0 / 9 (0.0%)	0 / 9 (0.0%)

Line Coverage

codecov.io, eg: <https://codecov.io/gh/spiffxp/kubernetes>



Line Coverage - next



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



KubeCon



CloudNativeCon

China 2018



@BenTheElder



@hh



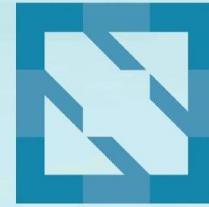
@Katharine



@zachmandeville



KubeCon



CloudNativeCon

China 2018



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

Requirements*

Works without public internet access

- We will assume test images have been pre-pulled onto nodes
- No interaction with services outside of the cluster

Requirements*

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

Requirements*

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

Requirements*



CloudNativeCon
China 2018

