



KubeCon



CloudNativeCon

Europe 2023

# Debugging Kubernetes E2E tests with Delve

Mauricio Poppe, Google



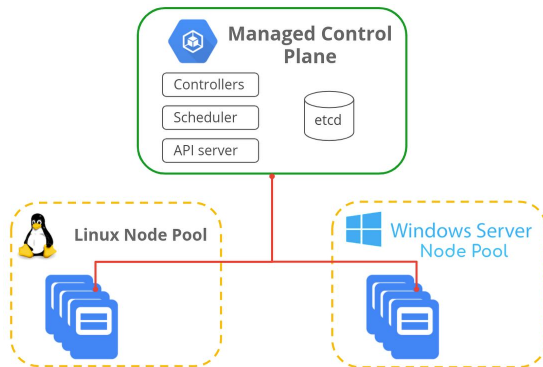
**Objective:** Bring CSI Windows  +  to GA

**Task:** Run the Kubernetes Storage E2E tests using a CSI Driver installed in a cluster with a Windows nodepool.



If you want to create a development cluster read [cluster/gce/windows/README-GCE-Windows-kube-up.md](https://github.com/kubernetes-sigs/windows-csi/blob/master/cluster/gce/windows/README-GCE-Windows-kube-up.md)

I read cluster/gce/windows/README-GCE-Windows-kube-up.md... and after **a lot** of time I created a cluster:





Nice! now all you have to do is:

- install a CSI Driver ✓
- run the e2e tests, [the instructions are here](#).

Thanks for the instructions! I can compile the e2e tests with `make WHAT=test/e2e/e2e.test` and run a few tests with `kubetest --test` and it succeeded. How do I run the storage tests?

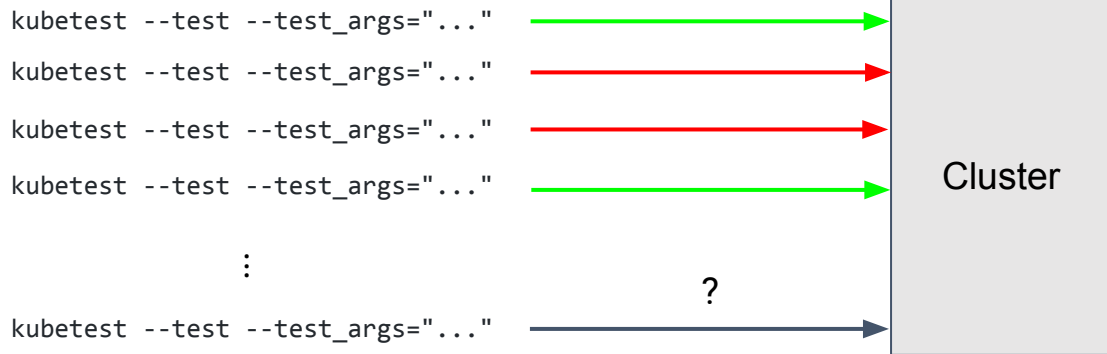


The e2e tests use ginkgo and we can use `--focus` to run e2e tests related to storage only.

I see, I can run a few storage tests with `kubetest --focus`, however, after trying to run a test a few times it **succeeds** sometimes but also it **times out** sometimes, I don't know why 🙄



I think that's happening because some tests can't run in Windows yet, try to dig deeper 😊



Read the test source code and try to correlate it with the test logs and with what I see in the cluster.

+ I see where a log is emitted and the sequence of events

- The test creates/deletes objects too fast for me to notice what's happening in the cluster 😞

Debug the test by adding debugging code (sleeps, printf, etc.)

+ I can stop in a line where I suspect things might be failing and analyze the cluster state.

- For every little change I have to recompile the test binary.

There's another option, instrument the e2e test binary to run in **debug mode** with [delve](#).

- + I can set breakpoints in any line.
- + I can analyze golang objects created in the test
- + I can analyze the cluster objects
- + One time setup!
- No docs on how to do that in kubernetes, however there are many resources online on how to setup a go debugger (delve) with a go program
- It might take a lot of time to do it

Even if I compile the e2e tests in debug mode, how do I run it with delve?

What happens when I run kubetest?



```
kubetest --test --test_args="<storage specific ginkgo flags>"
```

```
./hack/ginkgo-e2e.sh <storage specific ginkgo flags>
```

This is the e2e test binary, launched through **ginkgo**, can we launch it with delve instead?

```
_output/dockerized/bin/linux/amd64/ginkgo _output/bin/e2e.test --  
--kubeconfig=/usr/local/google/home/mauriciopoppe/.kube/config <lots of flags!> <storage  
specific ginkgo flags>
```

```
make WHAT=test/e2e/e2e.test DBG=1
```

Compile the binary with debug symbols, delve requirement

```
dlv exec _output/bin/e2e.test --  
--kubeconfig=/usr/local/google/home/mauriciopoppe/.kube/config <lots of flags!> <storage  
specific ginkgo flags>
```

Same e2e test binary, launched through **delve** this time



KubeCon



CloudNativeCon

Europe 2023  
North America 2021

ide-dark

```
~/go/src/k8s.io/kubernetes #v1.26.0 ?19 mauriciopoppe@mauriciopoppe 21:43:55
> dlv exec /usr/local/google/home/mauriciopoppe/go/src/k8s.io/kubernetes/_output/bin/e2e.test -- --kubeconfig=/usr/local/google/home/mauriciopoppe/.kube/config --host=https://35.232.217.5 --provider=gce --gce-project=mauriciopoppe-gke-dev --gce-zone=us-central1-b --gce-region=us-central1 --gce-multizone=false --gke-cluster=e2e-test-mauriciopoppe --kube-master=e2e-test-mauriciopoppe-master --cluster-tag= --cloud-config-file= --repo-root=/usr/local/google/home/mauriciopoppe/go/src/k8s.io/kubernetes --node-instance-group=e2e-test-mauriciopoppe-minion-group --prefix=e2e --network=e2e --node-tag=e2e-test-mauriciopoppe-minion --master-tag=e2e-test-mauriciopoppe-master --docker-config-file= --dns-domain=cluster.local --prepull-images=false --master-os-distro=gci --node-os-distro=gci --num-nodes=2 --ginkgo.slow-spec-threshold=300s --ginkgo.poll-progress-after=300s --ginkgo.poll-progress-interval=20s --ginkgo.source-root=/usr/local/google/home/mauriciopoppe/go/src/k8s.io/kubernetes --ginkgo.timeout=24h --ginkgo.flake-attempts=1 '--ginkgo.focus=sig-storage.*csi-hostpath.*fs.*should.resize.volume.when.PVC.is.edited.while.pod.is.using.it'
Type 'help' for list of commands.
(dlv)
```

kubernetes 1 dlv 2 k9s

&lt; Wed Mar 22 &lt; 21:44

Oh I see why it succeeds and fails sometimes, if the Pod gets scheduled to run in a Linux VM then it works but if it runs in a Windows VM then it fails to come up.



Good job! Let's add a nodeSelector conditionally in the Pods so they always get scheduled in Windows VMs. That way we can always hit the issue and debug further.

I think the debugger would be helpful for others too. Let's update the e2e test entrypoint to run a debugger if needed.



[E2E\\_TEST\\_DEBUG\\_TOOL=delve ./hack/ginkgo-e2e.sh ...](#)



# Can we debug the kube-controller-manager?

- Compile the kube-controller-manager in debug mode

```
make WHAT=cmd/kube-controller-manager DBG=1
```

- Remove the kube-controller-manager manifest from a CP host to remove the static Pod
- Start the kube-controller-manager with delve in a headless mode

```
dlv --listen :38697 --accept-multiclient --api-version=2 --headless \  
  exec ./_output/bin/kube-controller-manager -- \  
  --kubeconfig=${HOME}/.kube/config --leader-elect=false --v=4 \  
  --controllers="persistentvolume-binder,pvc-protection,pv-protection" \  
  --pvclaimbinder-sync-period=10000h
```

- Connect to it using your editor instead of the delve repl

Detailed instructions:

<https://github.com/mauriciopoppe/kubernetes-playground/blob/master/docs/kube-controller-manager.md>

# Can we debug the kube-controller-manager?



KubeCon



CloudNativeCon

```
ide-dark

* pv_controller.go pv_controller_base.go
243
244 : // filteredDialOptions configures any dialing done by the controller.
245 : filteredDialOptions *proxyutil.FilteredDialOptions
246 }
247
248 // syncClaim is the main controller method to decide what to do with a claim.
249 // It's invoked by appropriate cache.Controller callbacks when a claim is
250 // created, updated or periodically synced. We do not differentiate between
251 // these events.
252 // For easier readability, it was split into syncUnboundClaim and syncBoundClaim
253 // methods.
254 func (ctrl *PersistentVolumeController) syncClaim(ctx context.Context, claim *v1.
PersistentVolumeClaim) error {
255 : klog.V(4).Infof("synchronizing PersistentVolumeClaim[%s]: %s", claimToClaimKey(claim),
getClaimStatusForLogging(claim))
256
257 : // Set correct "migrated-to" annotations on PVC and update in API server if
258 : // necessary
259 : newClaim, err := ctrl.updateClaimMigrationAnnotations(ctx, claim)
260 : if err != nil {
261 : // Nothing was saved; we will fall back into the same
262 : // condition in the next call to this method
263 : return err
264 : }
265 : claim = newClaim
266
267 : if !metav1.HasAnnotation(claim.ObjectMeta, storagehelpers.AnnBindCompleted) {
268 : return ctrl.syncUnboundClaim(ctx, claim)
269 : } else {
270 : return ctrl.syncBoundClaim(claim)
271 : }
272 }
273
274 // checkVolumeSatisfyClaim checks if the volume requested by the claim satisfies the
requirements of the claim
275 func checkVolumeSatisfyClaim(volume *v1.PersistentVolume, claim *v1.PersistentVolumeClaim)
error {
276 : requestedQty := claim.Spec.Resources.Requests[v1.ResourceName(v1.ResourceStorage)]
277 : requestedSize := requestedQty.Value()
278
279 : // check if PV's DeletionTimeStamp is set, if so, return error.
NORMAL b46a3f pkg/controller/volume/persistentvolume/pv_controller.go 12% 255:21

~/go/src/k8s.io/kubernetes 1 editor
Wed Mar 22 22:34
```

# Can we debug the kubelet?

- Assumption a kind cluster with at least 1 worker node
- Add debugging tools to a kind worker (install delve and run dlv exec through systemd)

```
cdebug exec --image kubelet-debug:latest -it docker://kind-worker  
'$CDEBUG_WORKSPACE/app/kubelet-debug-entrypoint.sh'
```

Uses kubelet-debug, an image defined defined in my repo

- Compile the kubelet in debug mode

```
KUBE_VERBOSE=0 KUBE_FASTBUILD=true KUBE_RELEASE_RUN_TESTS=n \  
./build/make-in-container.sh make all WHAT=cmd/kubelet DBG=1
```

- Restart the systemd service for the kubelet-debug service (created through cdebug).

```
docker cp _output/dockerized/bin/linux/arm64/kubelet kind-worker:/usr/bin/kubelet-debug  
docker exec -i kind-worker bash -c "systemctl daemon-reload; systemctl restart  
kubelet-debug"
```

- Connect to it using your editor instead of the delve repl

Detailed instructions:

<https://github.com/mauriciopoppe/kubernetes-playground/blob/master/docs/kubelet.md>

# Can we debug the kubelet?

## volumeManager Reconciler

```

volumemanager/cache/actual_state_of_world... volumemanager/cache/desiredStateOfWorld
Locals:
  rc *k8s.io/kubernetes/pkg/kubelet/vol... 169
  ▶ k8s.io/kubernetes/pkg/kubelet/volum... 170
  ▶ kubeClient k8s.io/client-go/kuberne... 171
  ▶ controllerAttachDetachEnabled bool 172
  ▶ loopSleepDuration time.Duration = k... 173
  ▶ waitForAttachTimeout time.Duration 174
  ▶ nodeName k8s.io/apimachinery/pkg/types.NodeName = "kind-worker" 175
  ▶ desiredStateOfWorld k8s.io/kubernet... 176
  ▶ actualStateOfWorld k8s.io/kubernet... 177
  ▶ populatorHasAddedPods func() bool = 178
  ▶ operationExecutor k8s.io/kubernetes... 179
  ▶ mounter k8s.io/mount-utils.Interface 180
  ▶ hostutil k8s.io/kubernetes/pkg/volu... 181
NORMAL ▶ a866cb <pes 44% 8:2
reconciler.go:
180 rc.unmountVolumes()
282 for volumeName, glblVolumeInfo := r...

volumemanager/cache/actual_state_of_world... volumemanager/cache/desiredStateOfWorld
Reconciler
169 func (rc *reconciler) reconcile() {
170 // referenced by a pod that was deleted and is now referenced by another
171 // pod is unmounted from the first pod before being mounted to the new
172 // pod.
180 rc.unmountVolumes()
181
182 // Next we mount required volumes. This function could also trigger
183 // attach if kubelet is responsible for attaching volumes.
184 // If underlying PVC was resized while in-use then this function also handles volume
185 // resizing.
186 rc.mountOrAttachVolumes()
187
188 // Ensure devices that should be detached/unmounted are detached/unmounted.
189 rc.unmountDetachDevices()
190
191 // After running the above operations if skippedDuringReconstruction is not empty
192 // then ensure that all volumes which were discovered and skipped during reconstruction
193 // are added to actualStateOfWorld in uncertain state.
194 if len(rc.skippedDuringReconstruction) > 0 {
195 rc.processReconstructedVolumes()
196 }
197 }
198
199 func (rc *reconciler) unmountVolumes() {
200 // Ensure volumes that should be unmounted are unmounted.
201 for _, mountedVolume := range rc.actualStateOfWorld.GetAllMountedVolumes() {
202 if !rc.desiredStateOfWorld.PodExistsInVolume(mountedVolume.PodName, mountedVolume.
203 -VolumeName, mountedVolume.SELinuxMountContext) {
204 // Volume is mounted, unmount it
205 klog.V(5).Infof(mountedVolume.GenerateMsgDetailed("Starting operationExecutor.
206 -UnmountVolume", ""))
207 err := rc.operationExecutor.UnmountVolume(
208 mountedVolume.MountedVolume, rc.actualStateOfWorld, rc.kubeletPodsDir)
209 if err != nil && !isExpectedError(err) {
210 klog.Errorf(err, mountedVolume.GenerateErrorDetailed(fmt.
211 -Printf("operationExecutor.UnmountVolume failed (controllerAttachDetachEnabled %v)", rc.
212 -controllerAttachDetachEnabled), err).Error())
213 }
214 }
215 }
216 }

```



KubeCon



CloudNativeCon

Europe 2023

North America 2021

**LET'S ADD A DEBUGGER  
TO ALL THE THINGS**





**KubeCon**



**CloudNativeCon**

**Europe 2023**

**Thank you**



# Debugging Kubernetes E2E tests with Delve



**KubeCon**



**CloudNativeCon**

Europe 2023

**Tuesday April 18**  
**17:30 CEST**



**Mauricio Poppe**  
Google





**Tim Hockin (thockin.yaml)**  
@thockin

It's true! Read [cluster/](#) and [hack/](#) in the kubernetes codebase

...

Ain't nothing wrong with shell scripts. That's all Kubernetes is, anyway...I kid, I kid. There's some Salt mixed in there, too.

Source: <https://twitter.com/thockin/status/860386482107801600>