

E2E Testing

Part 2

Let's do a simple change to the code...

```
$ cd $GOPATH_K8S
```

```
→ kubernetes git:(api_tweak) x git diff
diff --git a/cmd/kube-apiserver/app/server.go b/cmd/kube-apiserver/app/server.go
index 2660130f1d0..c9d4eb3d46c 100644
--- a/cmd/kube-apiserver/app/server.go
+++ b/cmd/kube-apiserver/app/server.go
@@ -147,6 +147,7 @@ @@ cluster's shared state through which all other components interact.`
func Run(completeOptions completedServerRunOptions, stopCh <-chan struct{}) error {
    // To help debugging, immediately log version
    klog.Infof("Version: %+v", version.Get())
+   klog.Infof("Friday Mode Activated: %+v", version.Get())

    server, err := CreateServerChain(completeOptions, stopCh)
    if err != nil {
(END)
```

```
# Build kind node image and create cluster
```

```
$ kind build node-image --image=kube_api_tweak:1234
```

```
$ kind create cluster --name=api-tweak --image=kube_api_tweak:1234
```

```
$ kubectl -n kube-system logs kube-apiserver-api-tweak-control-plane | grep Friday
```

```
10724 08:53:55.244753      1 server.go:150] Friday Mode Activated: v1.19.0-beta.0.276+081f97aa2b129c-dirty
```

<https://gist.github.com/rflorenc/b4aed6587adf84cb0f6b40b861a430b9> (for all the steps)



Bazel

{Fast, Correct} - Choose two

Bazel

- <https://bazel.build/>

Install Bazel via Homebrew

```
$ brew install bazel
```

Upgrade to a newer version of Bazel using the following command:

```
$ brew upgrade bazel
```

Kind and Bazel workflow

```
$ cd $GOPATH_K8S
```

```
# Create your kind node image
```

```
$ kind build node-image --image=kube_api_tweak:1234
```

Optional: create e2e cluster config kind-config.yaml.

For example: 1 node master/node vs 2 nodes (1 master/1 worker) cluster.

Use the previous config to create your cluster:

```
$ kind create cluster --config kind-config.yaml --image kube_api_tweak:1234 -v4
```

Build the **e2e.test** Kubernetes binary from your Kubernetes src code / branch

```
$ bazel build //test/e2e:e2e.test
```

(output) Target //test/e2e:e2e.test up-to-date: bazel-bin/test/e2e/e2e.test

Dry Run: `./bazel-bin/test/e2e/e2e.test -ginkgo.focus="\[sig-network\].*Conformance" -ginkgo.dryRun -num-nodes 2`

Execute your tests: `./bazel-bin/test/e2e/e2e.test -context kind-kind -ginkgo.focus="\[sig-network\].*Conformance" -num-nodes 2`

```
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
networking:
  ipFamily: ipv4
nodes:
- role: control-plane
- role: worker
- role: worker
```

Examples

```
$ ./bazel-bin/test/e2e/e2e.test -ginkgo.focus="Logging soak \[Performance\] \[Slow\] \[Disruptive\]" -ginkgo.dryRun -num-nodes 1
```

```
$ ./bazel-bin/test/e2e/e2e.test -context kind-kind -ginkgo.focus="\[sig-network\].*Conformance" -num-nodes 2
```

```
$ ./bazel-bin/test/e2e/e2e.test -ginkgo.focus="\[Feature:PerformanceDNS\]\[Serial\]" -ginkgo.dryRun -num-nodes 2
```

Framework

<https://github.com/kubernetes/kubernetes/blob/master/test/e2e/framework/framework.go>

Specifies a wrapper around Ginkgo.

Implements own versions of `afterEach`, `beforeEach`, and others.

Source

<https://github.com/kubernetes/community/blob/master/contributors/devel/sig-testing/writing-good-e2e-tests.md#writing-good-e2e-tests-for-kubernetes>

<https://github.com/kubernetes/community/blob/master/contributors/devel/sig-testing/e2e-tests.md#overview>

<https://github.com/kubernetes/community/blob/master/contributors/devel/sig-testing/writing-good-e2e-tests.md#some-concrete-tools-at-your-disposal>

```
framework.KubeDescribe("Secret", func() {  
    ginkgo.It("should create a pod that reads a secret", func() {  
        test := "test/fixtures/doc-yaml/user-guide/secrets"  
        secretYaml := readFile(test, "secret.yaml")  
        podYaml := readFile(test, "secret-pod.yaml.in")  
  
        nsFlag := fmt.Sprintf("--namespace=%v", ns)  
        podName := "secret-test-pod"  
  
        ginkgo.By("creating secret and pod")  
        framework.RunKubectlOrDieInput(ns, secretYaml, "create", "-f", "-", nsFlag)  
        framework.RunKubectlOrDieInput(ns, podYaml, "create", "-f", "-", nsFlag)  
        err := e2epod.WaitForPodNoLongerRunningInNamespace(c, podName, ns)  
        framework.ExpectNoError(err)  
  
        ginkgo.By("checking if secret was read correctly")  
        _, err = framework.LookForStringInLog(ns, "secret-test-pod", "test-container", "value-1", serverStartTimeout)  
        framework.ExpectNoError(err)  
    })  
})
```