test/e2e

This is home to e2e tests used for presubmit, periodic, and postsubmit jobs.

Some of these jobs are merge-blocking, some are release-blocking.

e2e test ownership

All e2e tests must adhere to the following policies: - the test must be owned by one and only one SIG - the test must live in/underneath a sig-owned package matching pattern: `test/e2e/[{subpath}/]{sig}/...`, e.g. - `test/e2e/auth` - all tests owned by sig-`auth` - `test/e2e/common/storage` - all tests `common` to cluster-level and node-level e2e tests, owned by sig-`node` - `test/e2e/upgrade/apps` - all tests used in `upgrade` testing, owned by sig-`apps` - each sig-owned package should have an OWNERS file defining relevant approvers and labels for the owning sig, e.g.

```
# test/e2e/node/OWNERS
# See the OWNERS docs at https://go.k8s.io/owners

approvers:
- alice
- bob
- cynthia
emeritus_approvers:
- dave
reviewers:
- sig-node-reviewers
labels:
- sig/node
```

- packages that use `{subpath}` should have an `imports.go` file importing sig-owned packages (for ginkgo's benefit), e.g.

```
// test/e2e/common/imports.go
package common

import (
    // ensure these packages are scanned by ginkgo for e2e tests
    _ "k8s.io/kubernetes/test/e2e/common/network"
    _ "k8s.io/kubernetes/test/e2e/common/node"
    _ "k8s.io/kubernetes/test/e2e/common/storage"
)
```

- test ownership must be declared via a top-level SIGDescribe call defined in the sig-owned package, e.g.

```
// test/e2e/lifecycle/framework.go
package lifecycle
```

```
import "github.com/onsi/ginkgo"

// SIGDescribe annotates the test with the SIG label.
func SIGDescribe(text string, body func()) bool {
    return ginkgo.Describe("[sig-cluster-lifecycle] "+text, body)
}

// test/e2e/lifecycle/bootstrap/bootstrap_signer.go

package bootstrap

import (
    "github.com/onsi/ginkgo"
    "k8s.io/kubernetes/test/e2e/lifecycle"
)
var _ = lifecycle.SIGDescribe("[Feature:BootstrapTokens]", func() {
  /* ... */
  ginkgo.It("should sign the new added bootstrap tokens", func() {
    /* ... */
  })
  /* etc */
})
```

These polices are enforced: - via the merge-blocking presubmit job `pull-kubernetes-verify` - which ends up running `hack/verify-e2e-test-ownership.sh` - which can also be run via `make verify WHAT=e2e-test-ownership`

more info

See kubernetes/community/.../e2e-tests.md