How we benefits from KIND (kubernetes in docker)

Dario Maiocchi

What is kind ?

https://github.com/kubernetes-sigs/kind

Kubernetes in docker

Why kind?

Kind is lightweight.

Run test locally and remotely (GitHub) fast and easy.

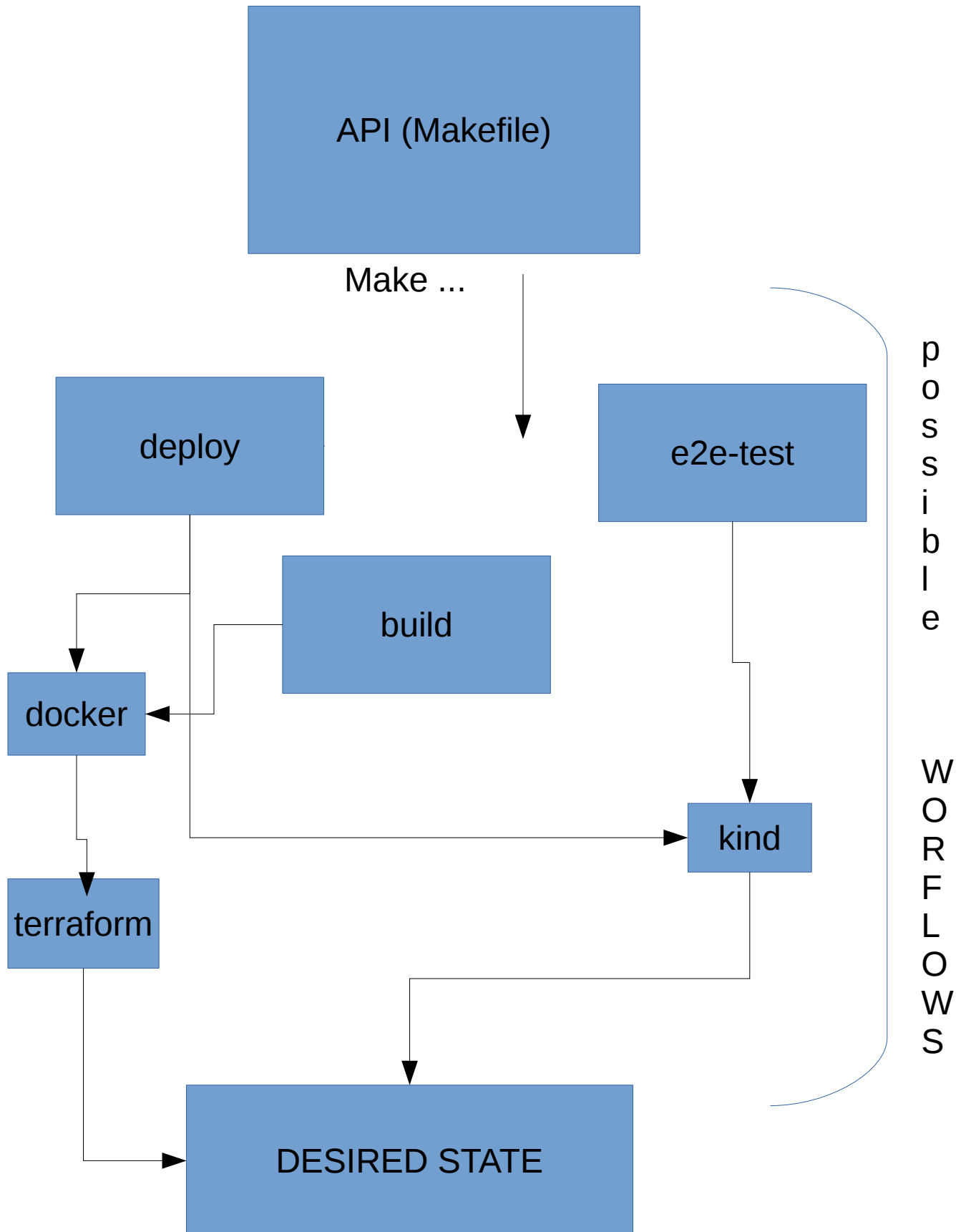It is used upstream

* Kind ecosystems and operators

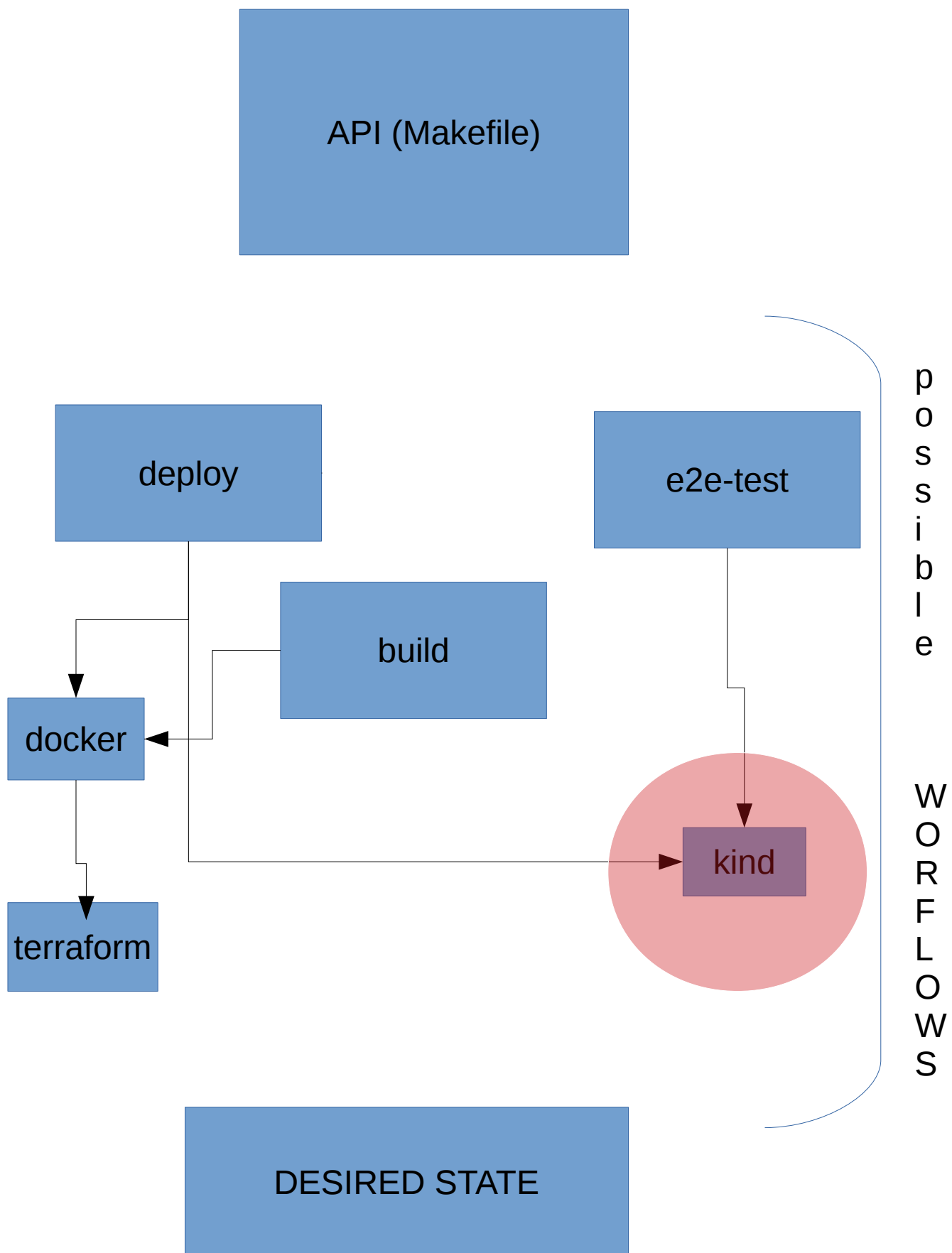1) The general idea about how kind fit in our existing context

2) Kind with kubernetes operators. (registry-operator and dex as example)

OUR DESIGN GOALS:

1) REMOTE = LOCAL execution

2) Frameworkless as possible

3) easy to execute, change and maintain. (API allow transparency)

4) Beeing composable of small units (UNIX principle)

API DEEP-DIVE

API (Makefile)

Make ...

deploy

e2e-test

build

docker

kind

terraform

DESIRED STATE

possible WORFLOWS

API (Makefile)

deploy

e2e-test

build

docker

kind

terraform

possible WORFLOWS

DESIRED STATE

KIND with kubernetes operators.

(Kubernetes operator pattern extend K8s cluster with custom functionality)

GITHUB = Local Env

Setup cluster with kind

Run integration- test
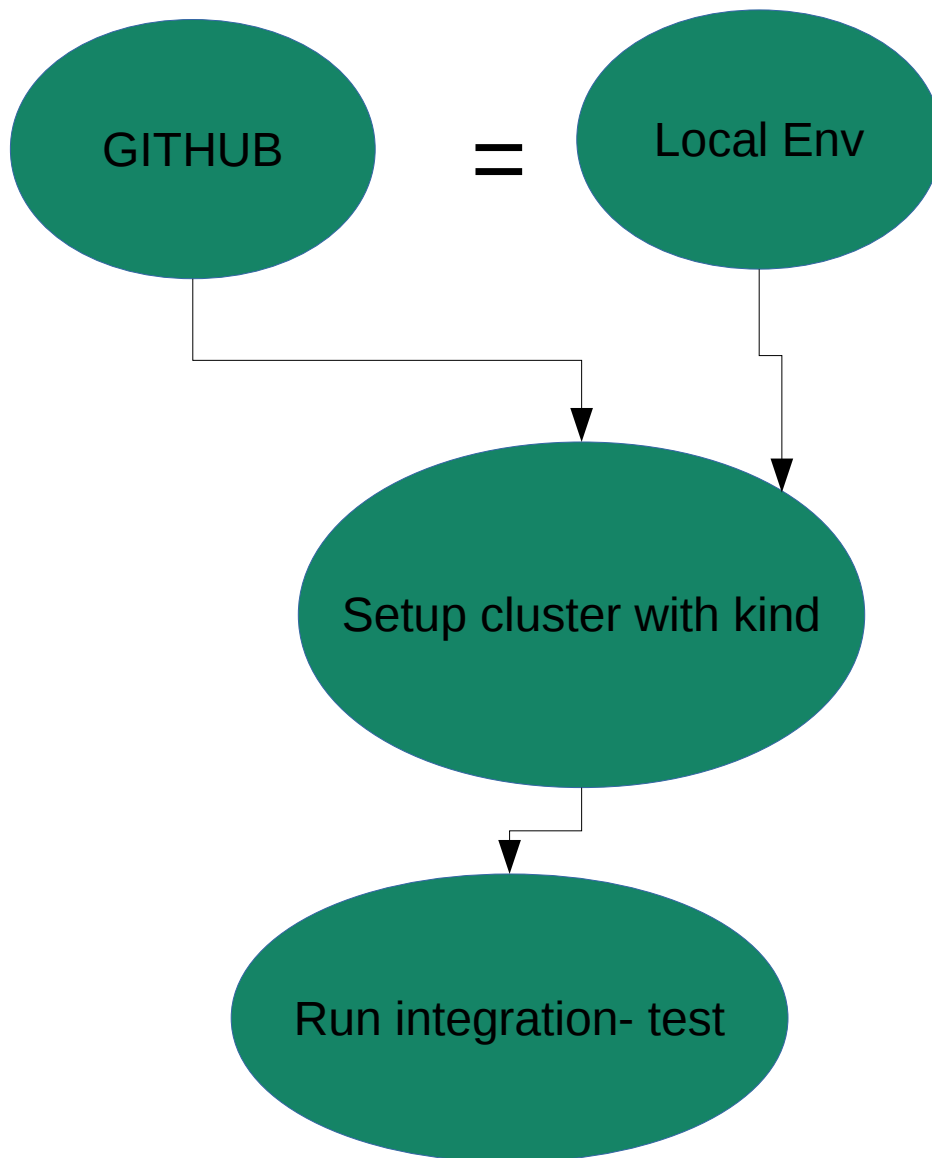
GOAL: validate operators against a cluster before merging Prs.

EXAMPLES:

```
441  $ go env                                                                                    go.env
468  Using Go 1.5 Vendoring, not checking for Godeps
469  Makefile detected                                                                           install
470  $ curl -Lo kubectl https://storage.googleapis.com/kubernetes-release/release/v1.12.0/bin/linux/amd64   before_script.1   0.35s
471  $ make kind-install KIND_VERSION="0.0.1"                                                    before_script.2   1.73s
477  $ make kind-create-cluster                                                                  before_script.3   95.65s
478  kind create cluster
479  Creating cluster 'kind-1' ...
480  ✓ Ensuring node image (kindest/node:v1.12.2) 🖼
481  ✓ [kind-1-control-plane] Creating node container 📦
482  ✓ [kind-1-control-plane] Fixing mounts 🖼
483  ✓ [kind-1-control-plane] Starting systemd 🖥
484  ✓ [kind-1-control-plane] Waiting for docker to be ready 🛳
485  ✓ [kind-1-control-plane] Starting Kubernetes (this may take a minute) ☸
486  Cluster creation complete. You can now use the cluster with:
487
488  export KUBECONFIG="$(kind get kubeconfig-path)"
489  kubectl cluster-info
490  $ make kind-e2e-tests                                                                       113.16s
491  make e2e-tests KUBECONFIG=/home/travis/.kube/kind-config-1
492  make[1]: Entering directory `/home/travis/gopath/src/github.com/kubic-project/registries-operator'
493  sh ./tests/e2e-tests.sh
494  KUBECONFIG variable point to :
495  /home/travis/.kube/kind-config-1
496
497   -- cluster info --
498  Kubernetes master is running at https://localhost:35184
499  KubeDNS is running at https://localhost:35184/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
500
501  To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
502
503  make[2]: Entering directory `/home/travis/gopath/src/github.com/kubic-project/registries-operator'
504  >>> Building cmd/registries-operator/registries-operator...
505  GO111MODULE=on GO15VENDOREXPERIMENT=1 go build -ldflags "-X=main.Version=1.0.0 -X=main.Build=`git rev-parse HEAD 2>/dev/null`" -o
     cmd/registries-operator/registries-operator cmd/registries-operator/main.go
```

```
20  go: downloading golang.org/x/text v0.3.0
21  go: downloading golang.org/x/sys v0.0.0-20181005133103-4497e2df6f9e
22  go: downloading k8s.io/utils v0.0.0-20180918230422-cd34563cd63c
23  go: downloading github.com/ugorji/go v1.1.1
24  >>> Using /home/travis/.kube/kind-config-1...
25  >>> Loading stuff with kubectl apply
26  serviceaccount/regs-jobs created
27  serviceaccount/regs-controller created
28  customresourcedefinition.apiextensions.k8s.io/registries.kubic.opensuse.org created
29  >>> Running cmd/registries-operator/registries-operator
30  cmd/registries-operator/registries-operator manager \
31               -v 5 \
32               --kubeconfig /home/travis/.kube/kind-config-1 \
33               &
34  make[2]: Leaving directory `/home/travis/gopath/src/github.com/kubic-project/registries-operator'
35  make[1]: Leaving directory `/home/travis/gopath/src/github.com/kubic-project/registries-operator'
36  The command "make kind-e2e-tests" exited with 0.
37
38
39
40  Done. Your build exited with 0.
                                                                                                 Top ▲
```

Makefile?

The Lindy effect is a concept that the future life expectancy of some non-perishable things like a technology or an idea is proportional to their current age, so that every additional period of survival implies a longer remaining life expectancy.