loodse

KubeCon | CloudNativeCon

North America 2018

Spawning Kubernetes In CI For Integration Tests

Marko Mudrinić

Software Developer @ Loodse

@xmudrii

- Integration tests ensures the controller **works in-cluster**
- Therefore, integration tests require **a real Kubernetes cluster**

But how we can get **a real Kubernetes cluster** in our **CI environment**?

loodse

Running Kubernetes in CI

- There are **many tools** for running Kubernetes
- Many tools require **systemd**
- CI environment is usually **minimal,** with **no access** to systemd

@xmudrii

kind

(Kubernetes in Docker)

@xmudrii

What is kind?

- Tool for running **local Kubernetes clusters** using **Docker**
- Supports **Kubernetes 1.11+**
- Maintained by the **Kubernetes community**

Why kind?

- Only requirement is **Docker**
- Comes with **pre-built Docker image** containing **all dependencies**
- Clusters are provisioned using **kubeadm**
- **Customizable** cluster provisioning process

kind in Travis-CI

@xmudrii

```yaml
language: go
go:
  - '1.11.x'
services:
  - docker

jobs:
  include:
    - stage: E2E Tests
      before_script:
        # Download kubectl
        - curl -Lo kubectl \
          https://storage.googleapis.com/kubernetes-release/release/v1.12.0/bin/
          linux/amd64/kubectl && chmod +x kubectl && sudo mv kubectl /usr/local/bin/
        # Download and build kind
        - go get sigs.k8s.io/kind
        # Create a new kind cluster using default properties
        - kind create cluster
        # Set KUBECONFIG environment variable
        - export KUBECONFIG="$(kind get kubeconfig-path)"
      script: make test-e2e
```

Building Docker images

@xmudrii

loodse

Building Docker Images

- We need to pass image to **Docker** running in **kind’s container on the host**
- Currently **no native feature** for passing images, but it’s **work in progress**

```
IMAGE=xmudrii/travis-kind
KIND_CLUSTER_NAME="kind-1"
KIND_CONTAINER_NAME="${KIND_CLUSTER_NAME}-control-plane"

build_image() {
   # Switch to the project’s root directory
   cd $SCRIPT_ROOT
   # Create a temporary directory to store generated Docker image
   TMP_DIR=$(mktemp -d)
   IMAGE_FILE=${TMP_DIR}/image.tar.gz

   # Build Docker image
   docker build -t "${IMAGE}":latest .
   # Export generated Docker image to an archive
   docker save "${IMAGE}" -o "${IMAGE_FILE}"
   # Copy saved archive into kind's Docker container
   docker cp "${IMAGE_FILE}" "${KIND_CONTAINER_NAME}":/image.tar.gz
   # Import image into kind's Docker daemon to make it accessible by Kubernetes
   docker exec "${KIND_CONTAINER_NAME}" docker load -i /image.tar.gz
}
```

loodse

Conclusion

- kind allows you to run local Kubernetes cluster just with Docker
- Example repository: https://github.com/xmudrii/travis-kind
- Projects using kind:
  - https://github.com/jetstack/cert-manager
  - https://github.com/xmudrii/etcdproxy-controller

@xmudrii