Building git push workflows for Kubernetes

hasura.io

Hi!

Tanmai Gopal

Co-founder, hasura.io

@tanmaigo

What is a git-push workflow

- Write code
- `git push heroku master` to deploy

Changed the world for developers, because it used dev only tools (git). Reduced *unnecessary* abstractions.

kubectl + git. Everything can be built around these 2 systems. Build your own git-push workflows. The main goal is to simplify devops and “pipelines”.

The simplest DevOps task: Build & deploy

I have source code on my machine. I can run it locally.

I want to deploy my source code at current commit.

When git push:

- Build: Dockerfile tagged with commit
- Deploy: Apply changes to kubernetes deployment with new image tag

Before <> after

`$ docker build -t registry.com/my-image:my-tag`	
`$ docker push registry.com/my-image:my-tag`	`$ git push dev master`
`$ kubectl set image deployment/my-deployment container=registry.com/my-image:my-tag`	

Git hooks for “git push”

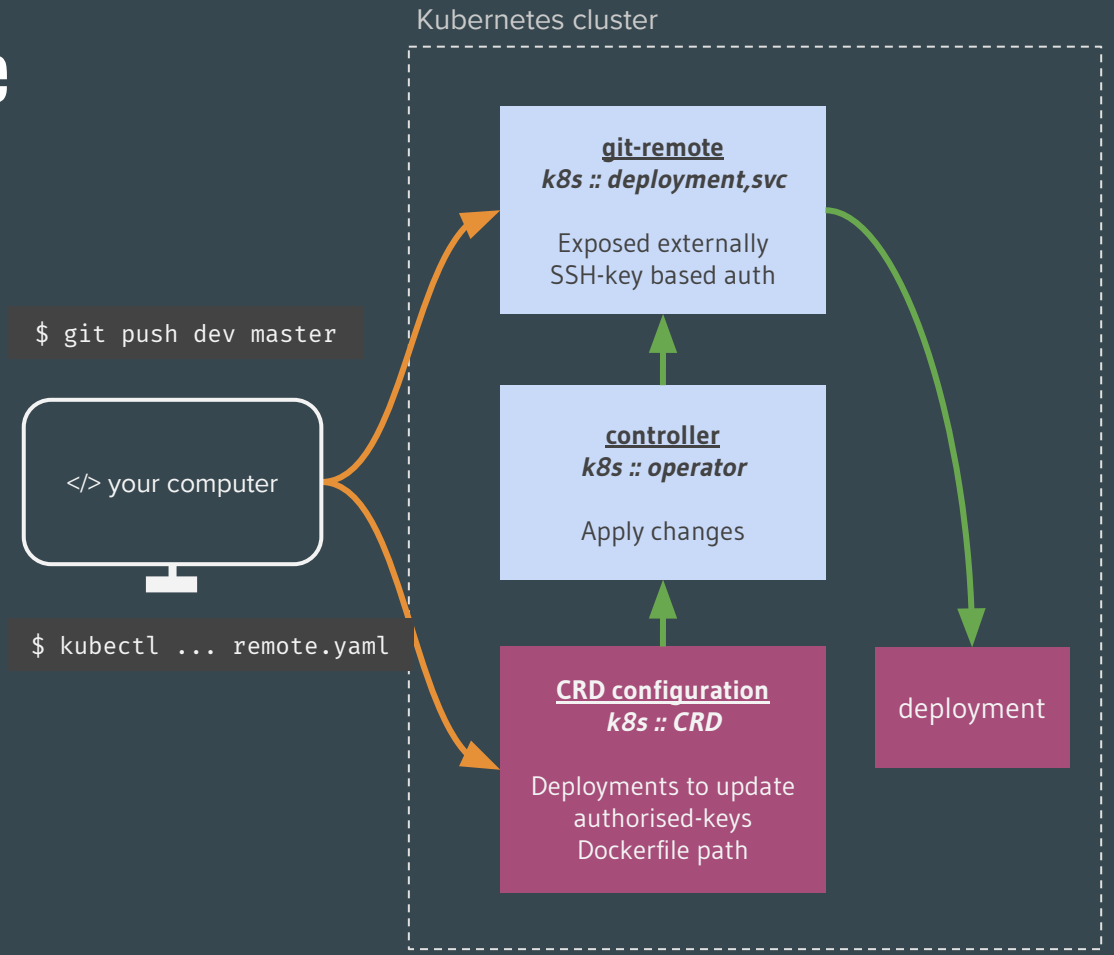
client-side	
> pre-push	Exit can abort push
server-side	
> pre-receive	Exit can abort push
> post-receive	Cannot abort push

Executable script in `.git/hooks/` named <hook>

`.git/hooks/pre-receive`

The obvious architecture

- Git remote agent on your cluster. Pre-receive hook:
  - Build docker image
  - Apply to k8s deployment
- Configuration:
  - SSH keys
  - Dockerfile path
- Sync configuration changes with the git-remote agent

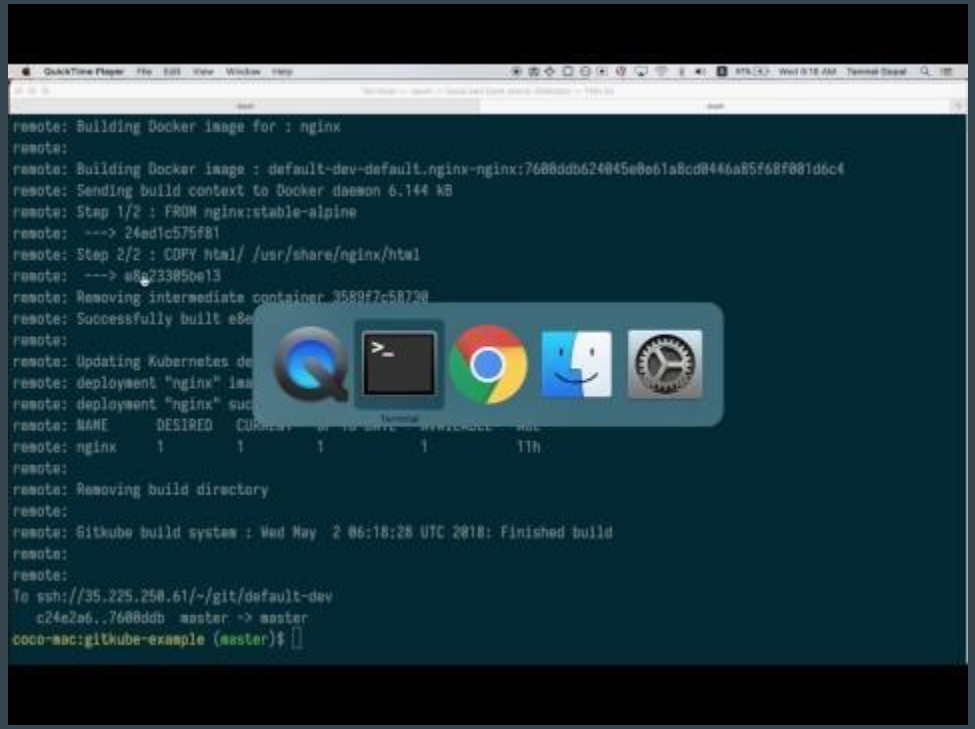


Setup

```
kubectl create -f
```

```yaml
apiVersion: gitkube.sh/v1alpha1
kind: Remote
metadata:
  name: dev
  namespace: default
spec:
  deployments:
  - name: nginx
    containers:
    - name: nginx
    path: .
    dockerfile: Dockerfile
  authorizedKeys:
  - "ssh-rsa <key>"
```

Demo 1: git push dev master to deploy an HTML webpage

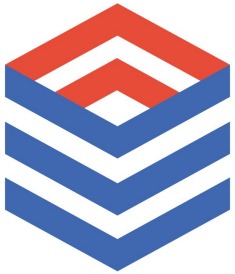


The pre-receive hook

https://github.com/hasura/gitkube/blob/master/build/gitkubed/pre_receive.sh

Run on any kubernetes cluster

Customise and extend for your own use-case



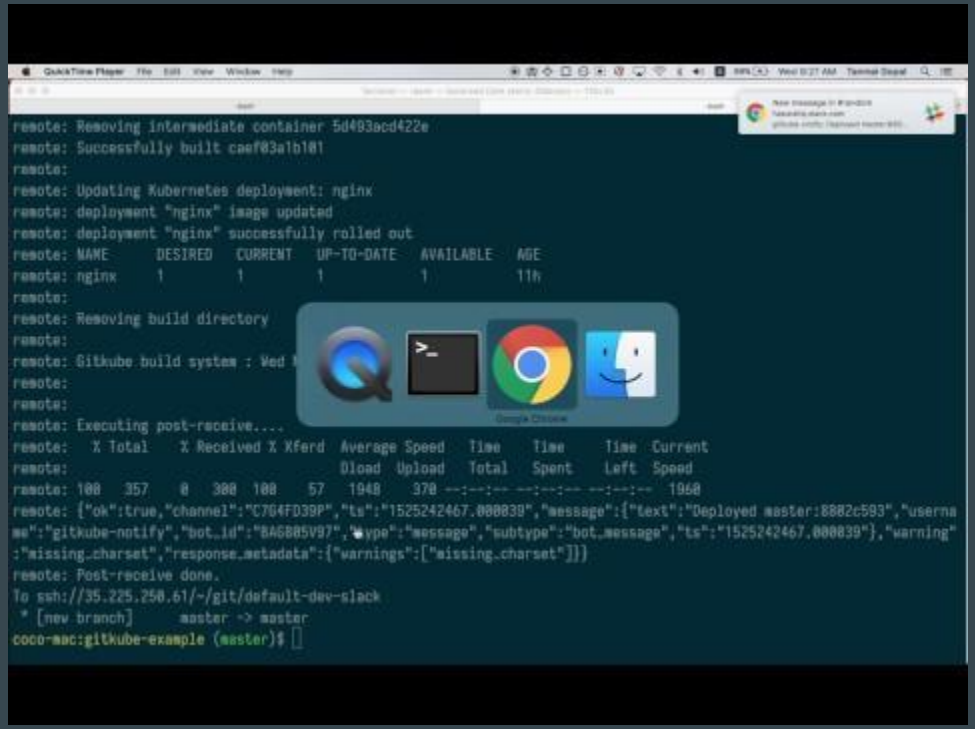
GITKUBE

https://github.com/hasura/gitkube

Benefits

- A developer only needs to have git.
- Customise the hook and just do you.
  - Your “hooks” have direct access to the cluster environment for your deployment tasks
  - Eg: Use secretKeyRefs to provide custom docker build args

Demo 2: Using k8s secrets in hooks to send a slack notif



We’re onto something here...

Extend this idea to more devops tasks?

git push to:

1. Build and run unit-tests
2. Deploy code
3. Deploy configuration
4. Apply stateful migrations
5. Run integration tests

But all based on just one idea

git push to apply changes to kubernetes objects.

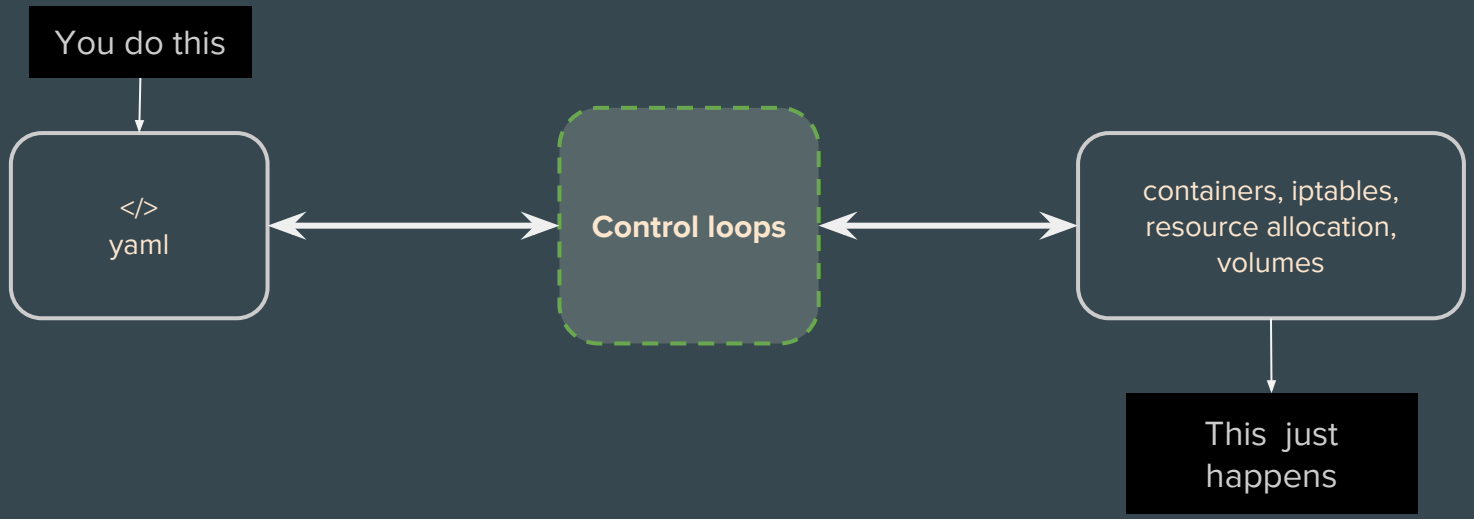
This way, git just works for AllTheThings™

```
git checkout <commit>

git push dev my-branch:master
```

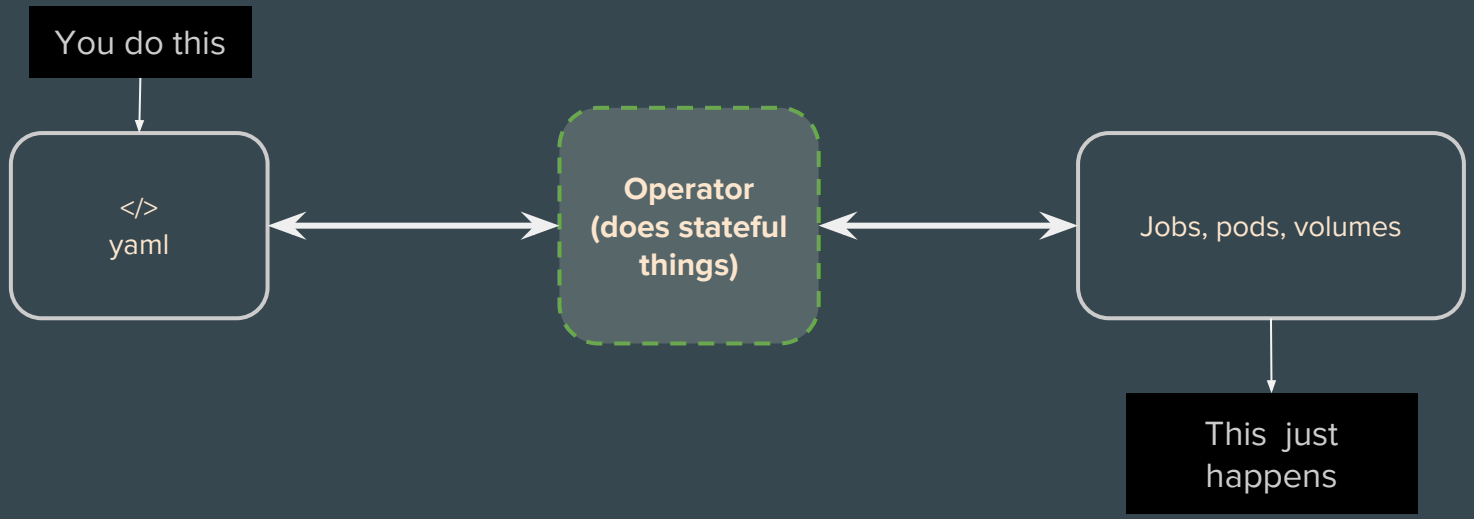
Kubernetes controller

The most awesome thing about how kubernetes works:

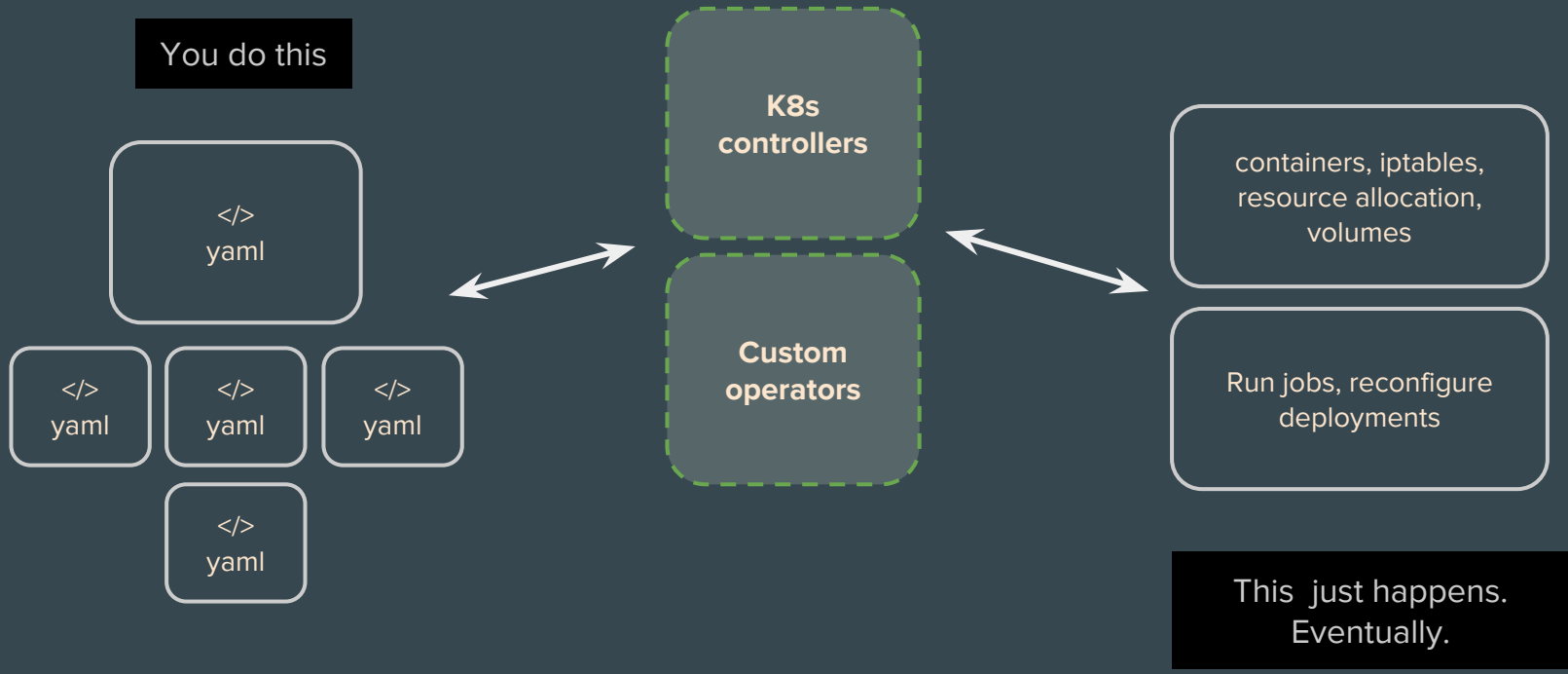


And everything is moving in this direction. Eg: The CRD + operator pattern

CRD + operator pattern



There is no notion of “sequence”



A typical DevOps pipeline needs to:

Build & test: Source code + unit tests

Deploy: Update configuration, run stateful tasks

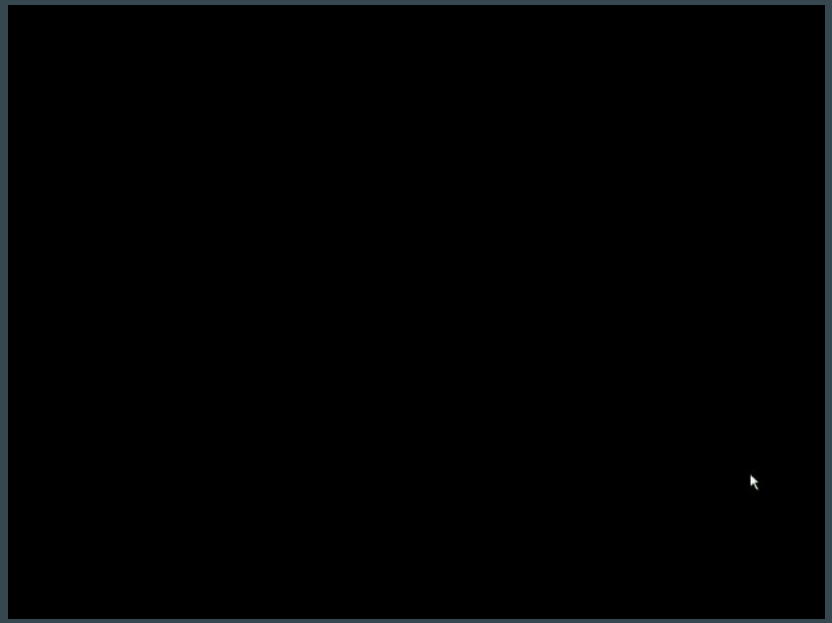
Integration tests: Test microservice1's dependency on microservice2

A typical DevOps pipeline needs to:

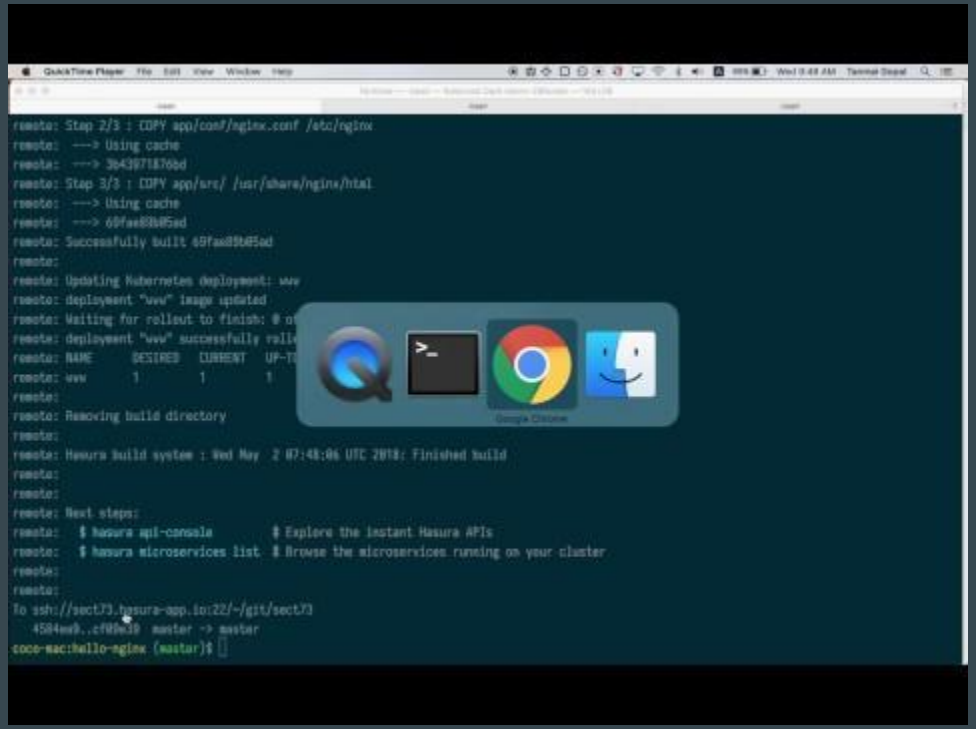
Build + run unit-tests	**Dockerfile**
Production build (artifacts)	**Multi-stage dockerfile**
Deploy configuration	**Update kubernetes manifests**
Run stateful tasks (database migrations)	**Update CRs**
Run integration tests	**Run jobs with init-containers to check if microservices are ready**

Goodbye “pipelines”?

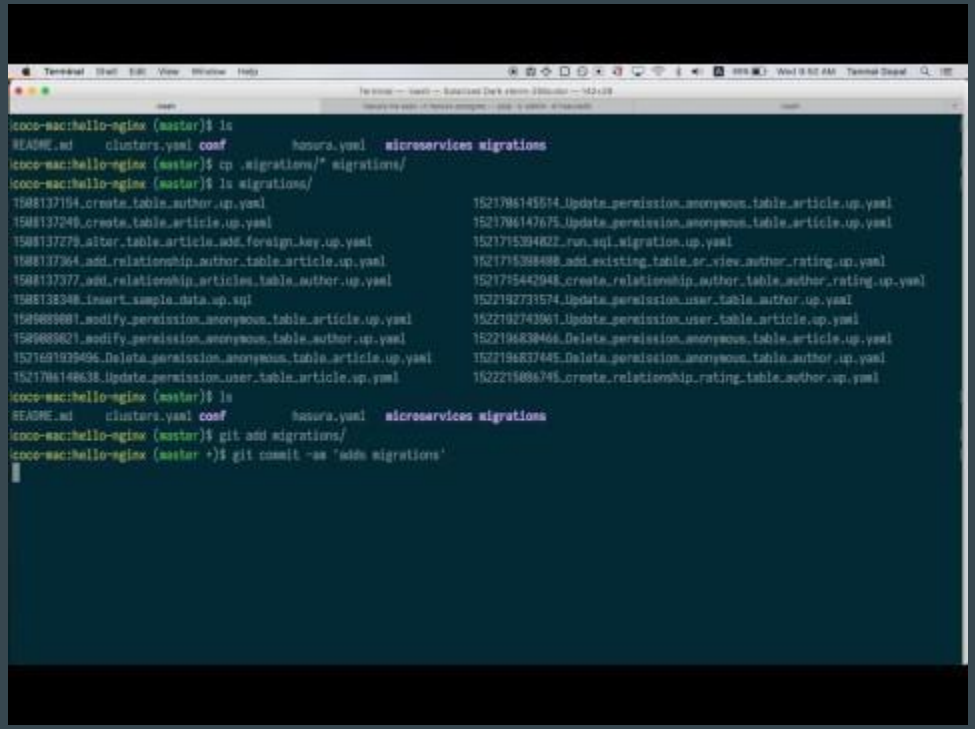
If everything is a kubernetes manifest backed by operators, then everything is declarative. Note: this is not talking about human-in-loop governance type pipelines, just the idea of sequenced tasks. Governance pipelines are a separate concern.



Demo 3: Change a subdomain by changing a YAML value



Demo 4: Applying postgres migrations on git push



Things that are hard with GitOps

- Secrets
  - Must be applied without committing to git
    - *pre-push hook?*
- Templating
  - Helm
  - Kubernetes native templating
- Releases, canary deployments?
  - GitOps with istio :)

Gitkube roadmap

- Easy to write custom hooks
  - Write in any language
  - Boilerplate/plugins
- UI to see past “pushes”
- `git push` ≣ `/vendor/webhook`

gitkube.sh

hasura.io

@tanmaigo

@HasuraHQ

SU-C30