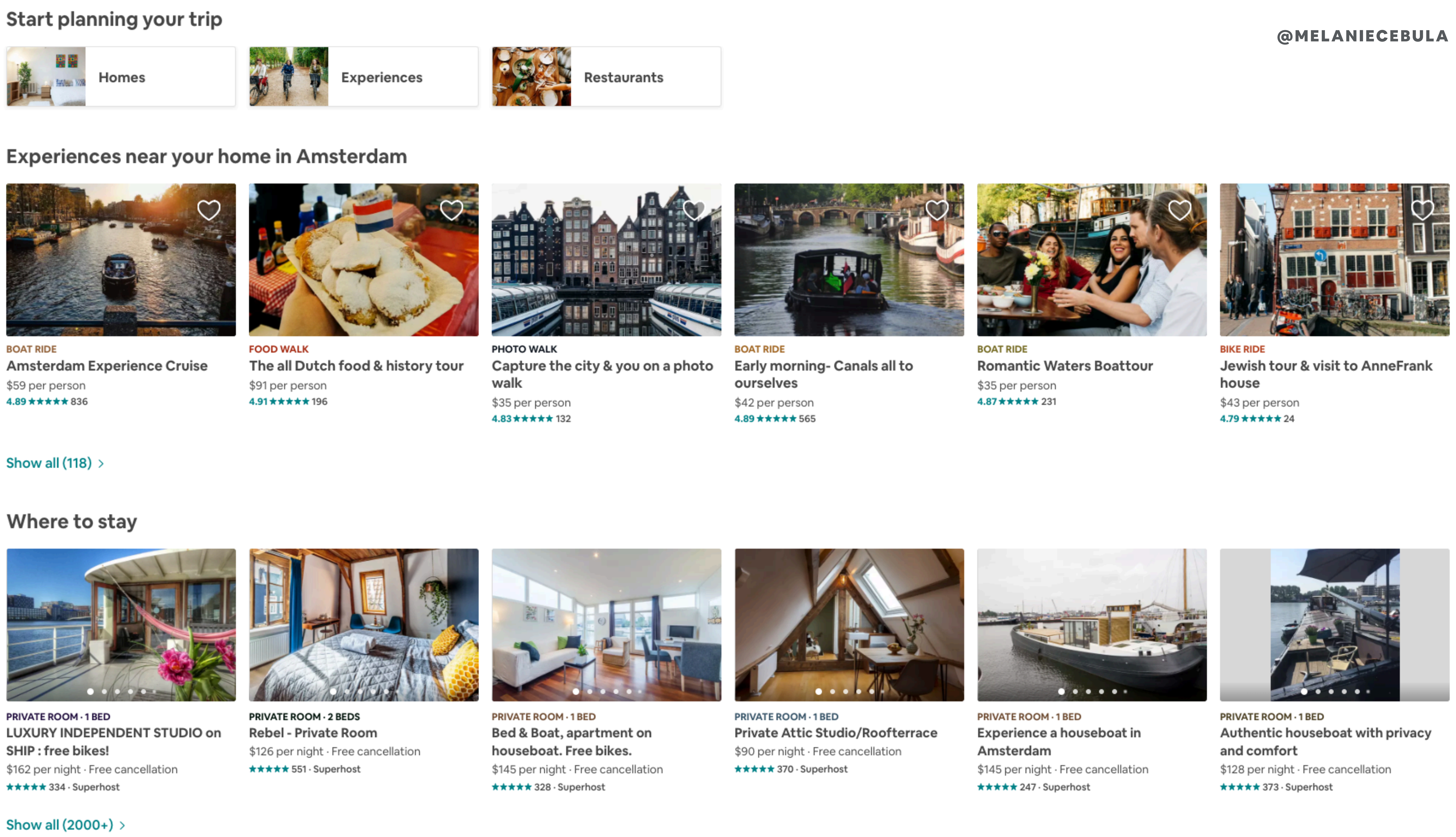
@MELANIECEBULA / DEC 2018 / KUBECON NORTH AMERICA

Developing Kubernetes Services

at Airbnb Scale



Start planning your trip

Homes

Experiences

Restaurants

Experiences near your home in Amsterdam

BOAT RIDE
Amsterdam Experience Cruise
$59 per person
4.89 ★★★★★ 836

FOOD WALK
The all Dutch food & history tour
$91 per person
4.91 ★★★★★ 196

PHOTO WALK
Capture the city & you on a photo walk
$35 per person
4.83 ★★★★★ 132

BOAT RIDE
Early morning- Canals all to ourselves
$42 per person
4.89 ★★★★★ 565

BOAT RIDE
Romantic Waters Boattour
$35 per person
4.87 ★★★★★ 231

BIKE RIDE
Jewish tour & visit to AnneFrank house
$43 per person
4.79 ★★★★★ 24

Show all (118) ›

Where to stay

PRIVATE ROOM · 1 BED
LUXURY INDEPENDENT STUDIO on SHIP : free bikes!
$162 per night · Free cancellation
★★★★★ 334 · Superhost

PRIVATE ROOM · 2 BEDS
Rebel - Private Room
$126 per night · Free cancellation
★★★★★ 551 · Superhost

PRIVATE ROOM · 1 BED
Bed & Boat, apartment on houseboat. Free bikes.
$145 per night · Free cancellation
★★★★★ 328 · Superhost

PRIVATE ROOM · 1 BED
Private Attic Studio/Roofterrace
$90 per night · Free cancellation
★★★★★ 370 · Superhost

PRIVATE ROOM · 1 BED
Experience a houseboat in Amsterdam
$145 per night · Free cancellation
★★★★★ 247 · Superhost

PRIVATE ROOM · 1 BED
Authentic houseboat with privacy and comfort
$128 per night · Free cancellation
★★★★★ 373 · Superhost

Show all (2000+) ›

What is airbnb?

AN ONLINE MARKETPLACE FOR SHARING HOMES AND EXPERIENCES

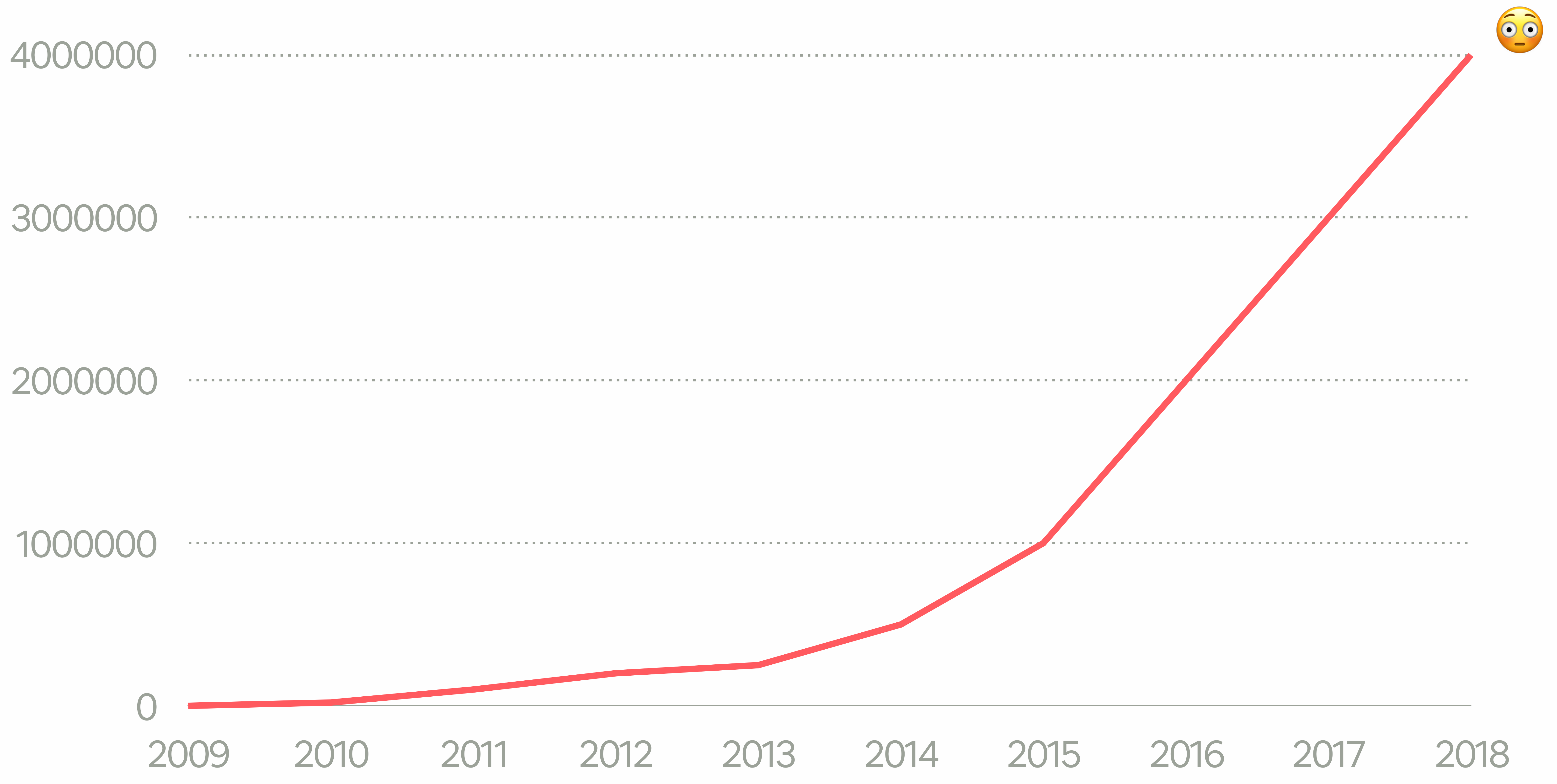
81k	191+	5 mil
cities	countries	homes

Who am I?

A BRIEF HISTORY

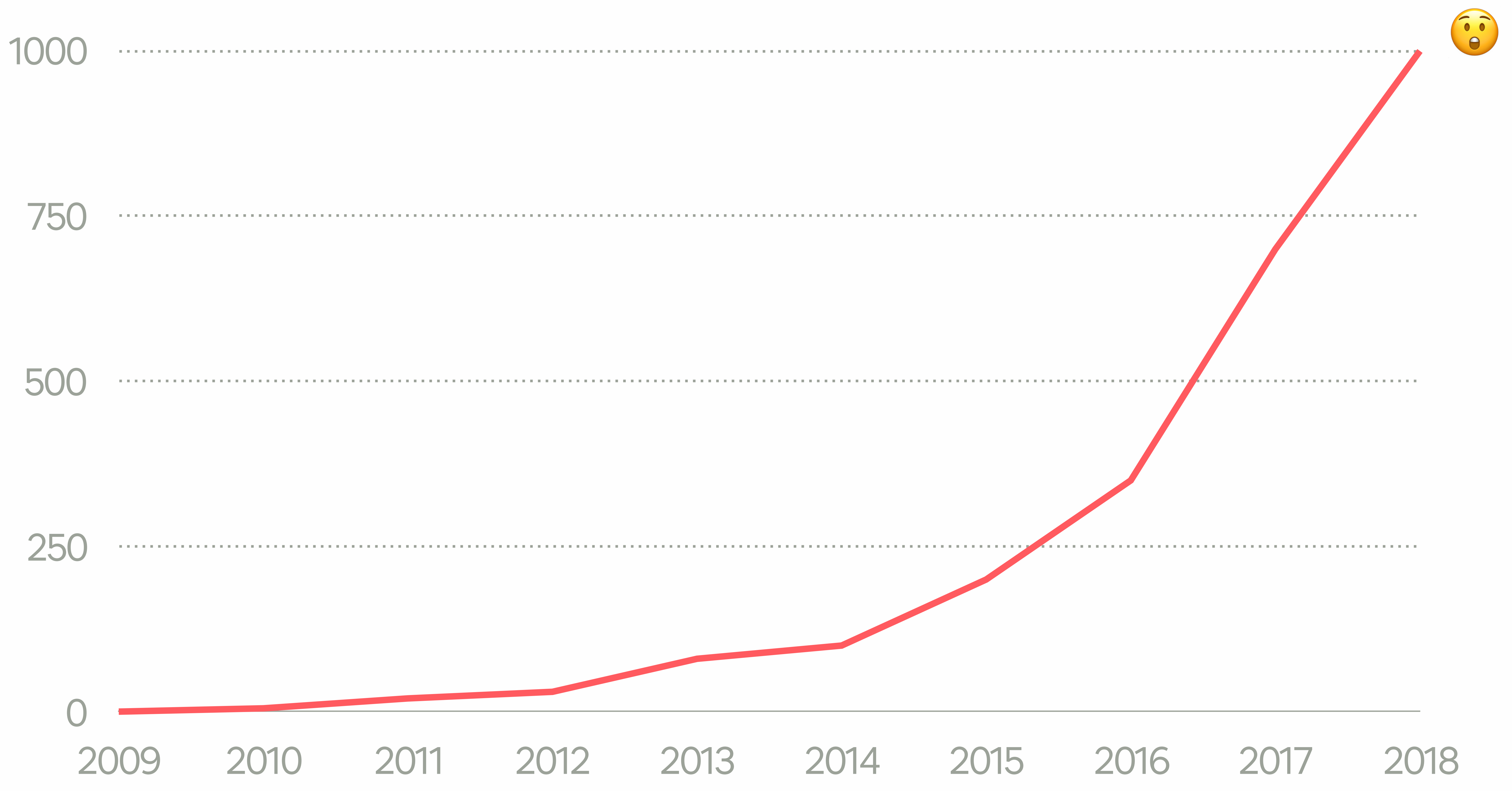
Why Microservices?

MONOLITH LOC



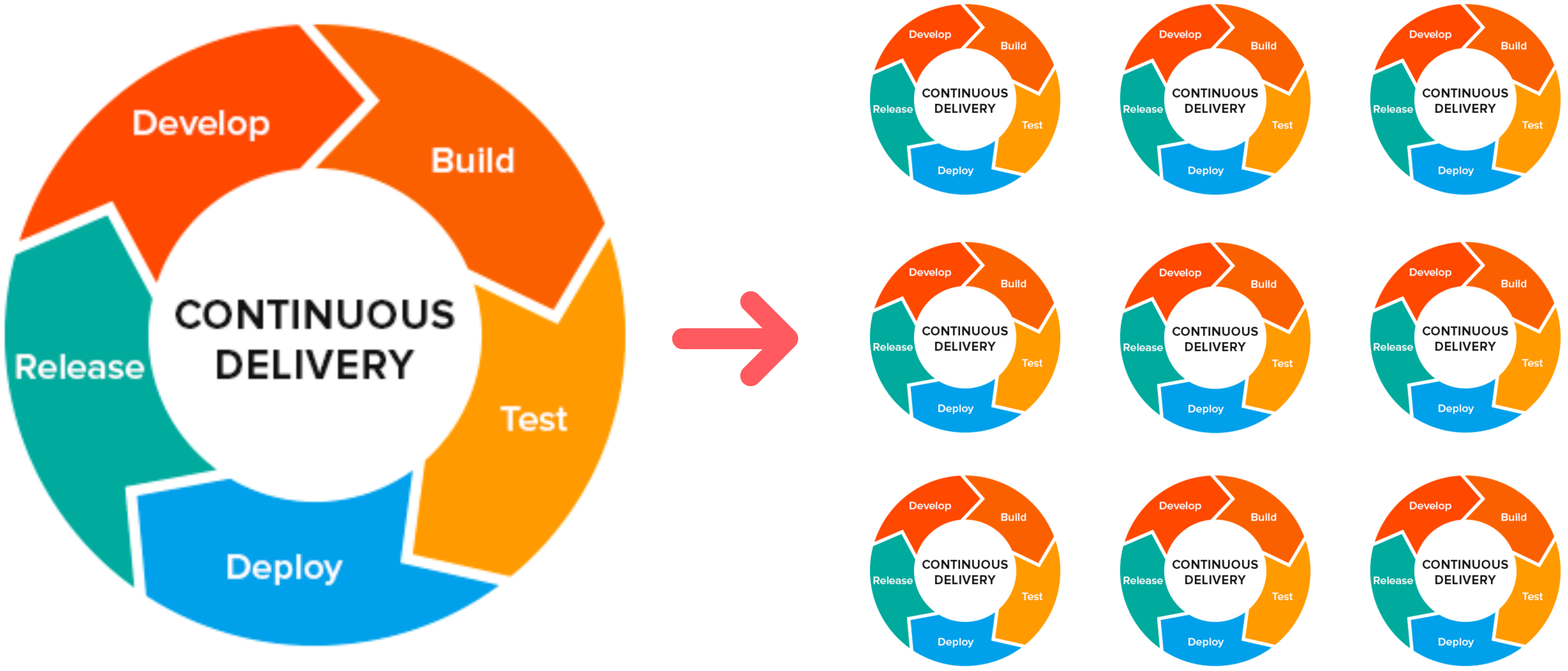
Why Microservices?

ENGINEERING TEAM



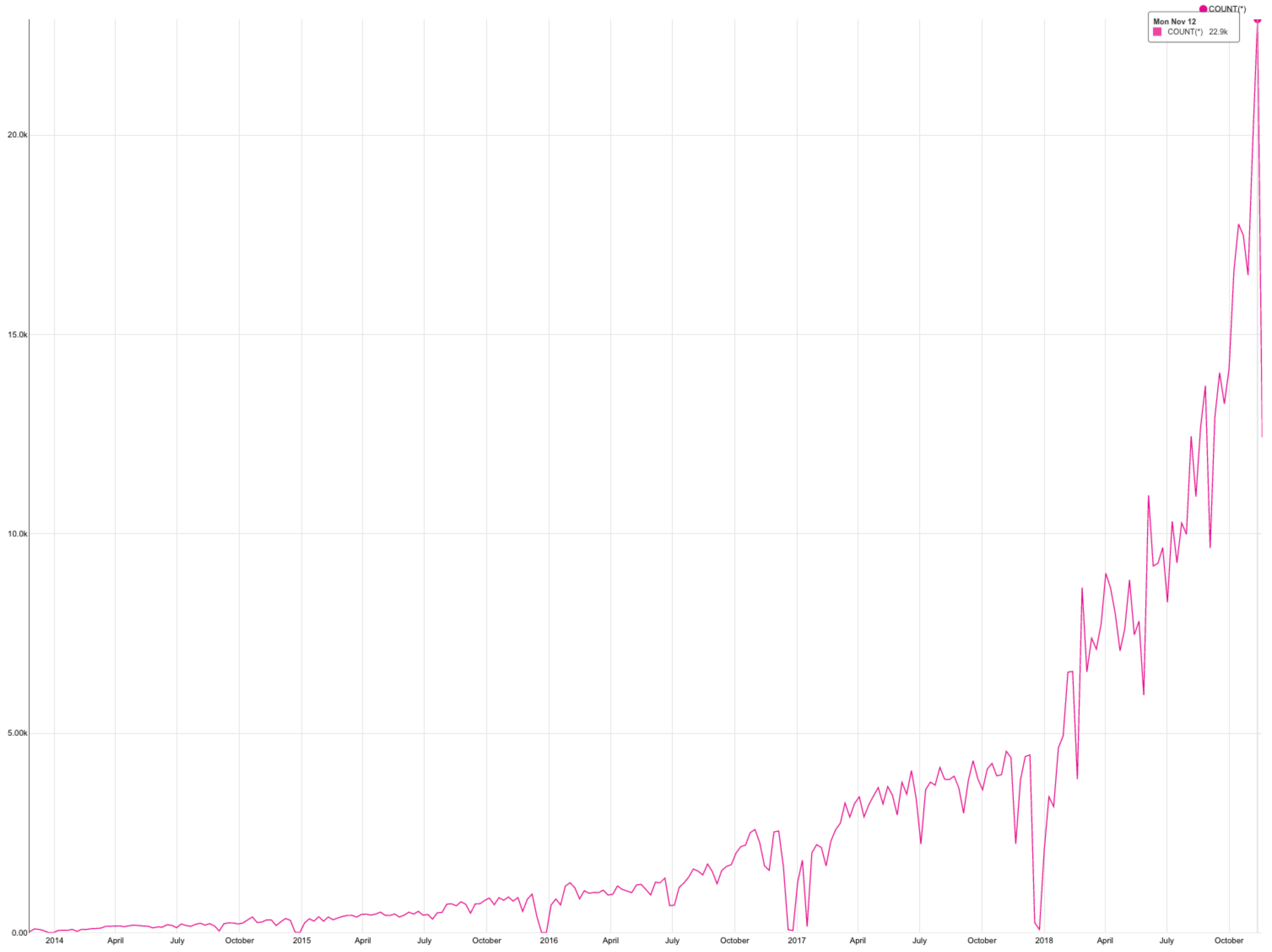
Why Microservices?

SCALING CONTINUOUS DELIVERY



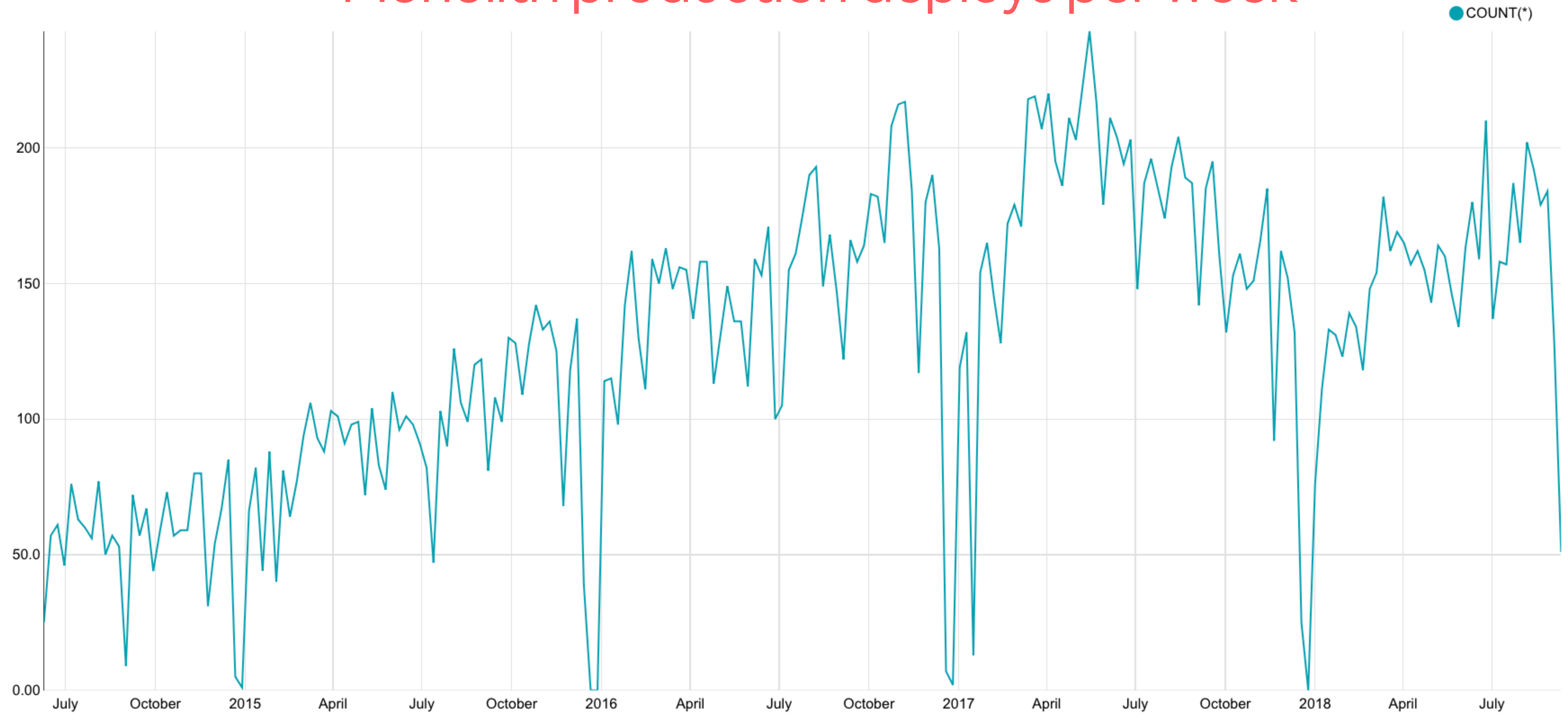
Why Microservices?

Deploys per week (all apps, all environments)



Why Microservices?

Monolith production deploys per week



Why Microservices?

125,000

production deploys

per year

Why kubernetes?

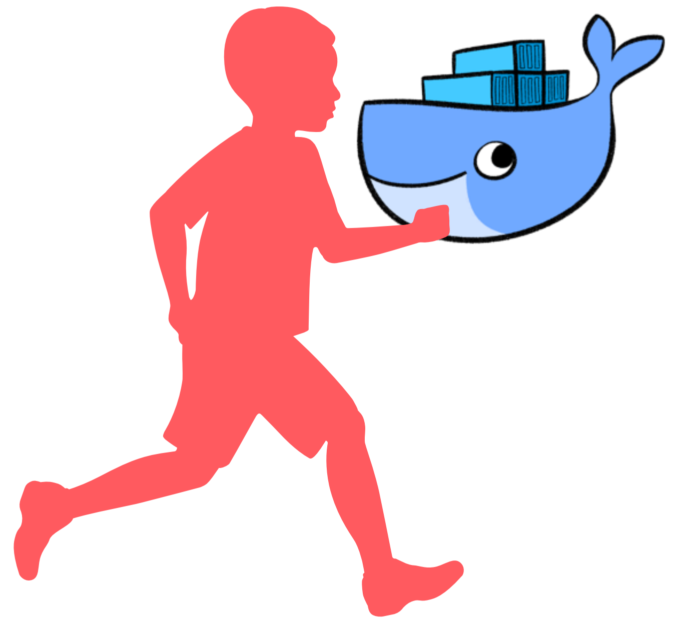
EVOLUTION OF CONFIGURATION MANAGEMENT

Manually configuring boxes

Automate configuration of applications with Chef

Automate configuration and orchestration of containerized applications with Kubernetes

Why kubernetes?



- declarative
- efficient scheduling
- extensible API

- portable
- immutable
- reproducible

- human-friendly data
- standard format

Challenges with kubernetes?

- complex configuration
- complex tooling
- open issues
- integrating with your current infrastructure

Challenges with kubernetes?

- complex configuration $\rightarrow$ generating kubernetes files
- complex tooling
- open issues
- integrating with your current infrastructure

Challenges with kubernetes?

- complex configuration $\rightarrow$ generating kubernetes files
- complex tooling $\rightarrow$ kubectl wrapper
- open issues
- integrating with your current infrastructure

Challenges with kubernetes?

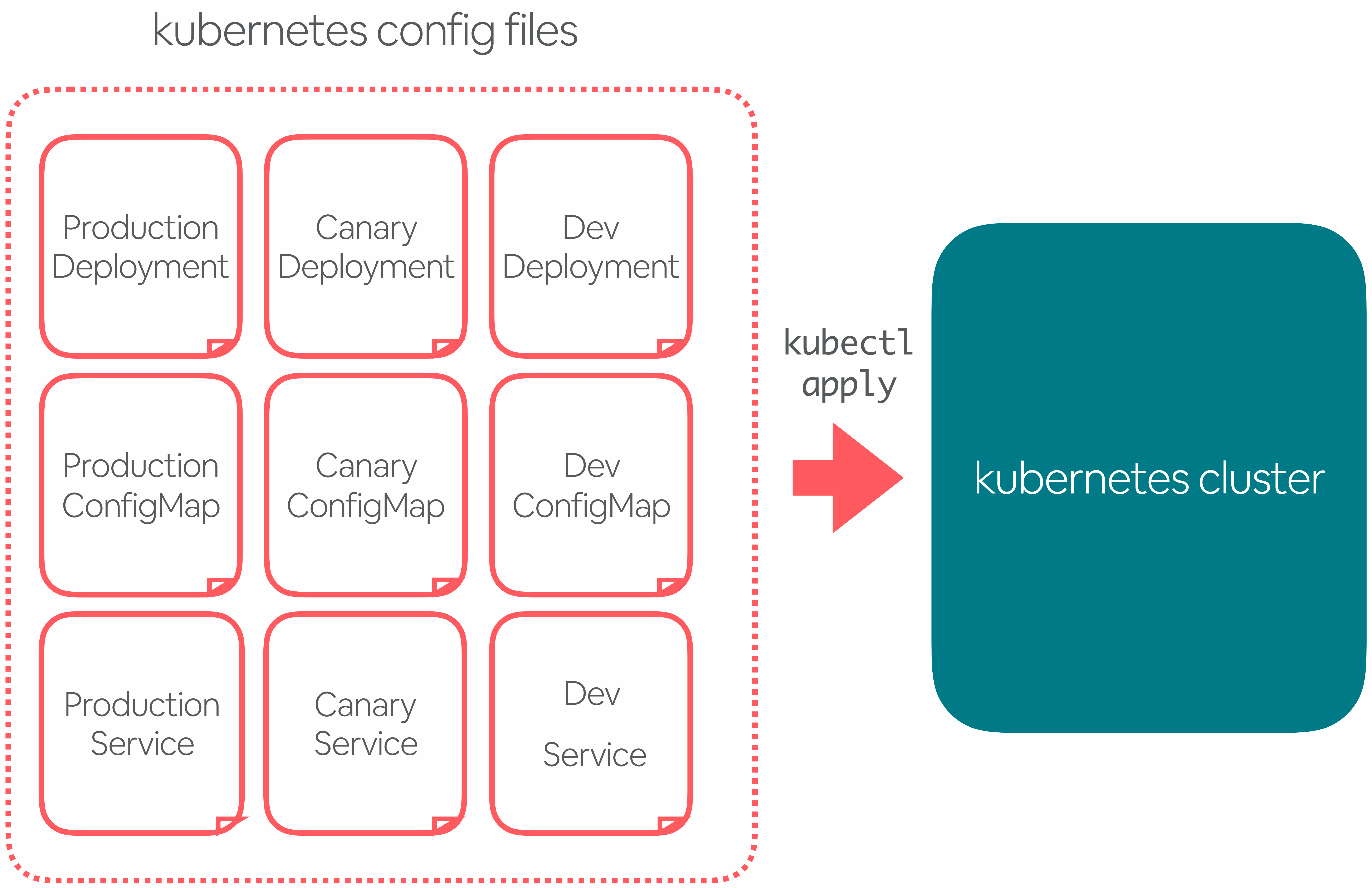
- complex configuration $\rightarrow$ generating kubernetes files
- complex tooling $\rightarrow$ kubectl wrapper
- open issues $\rightarrow$ contribute to open source and custom controllers
- integrating with your current infrastructure

Challenges with kubernetes?

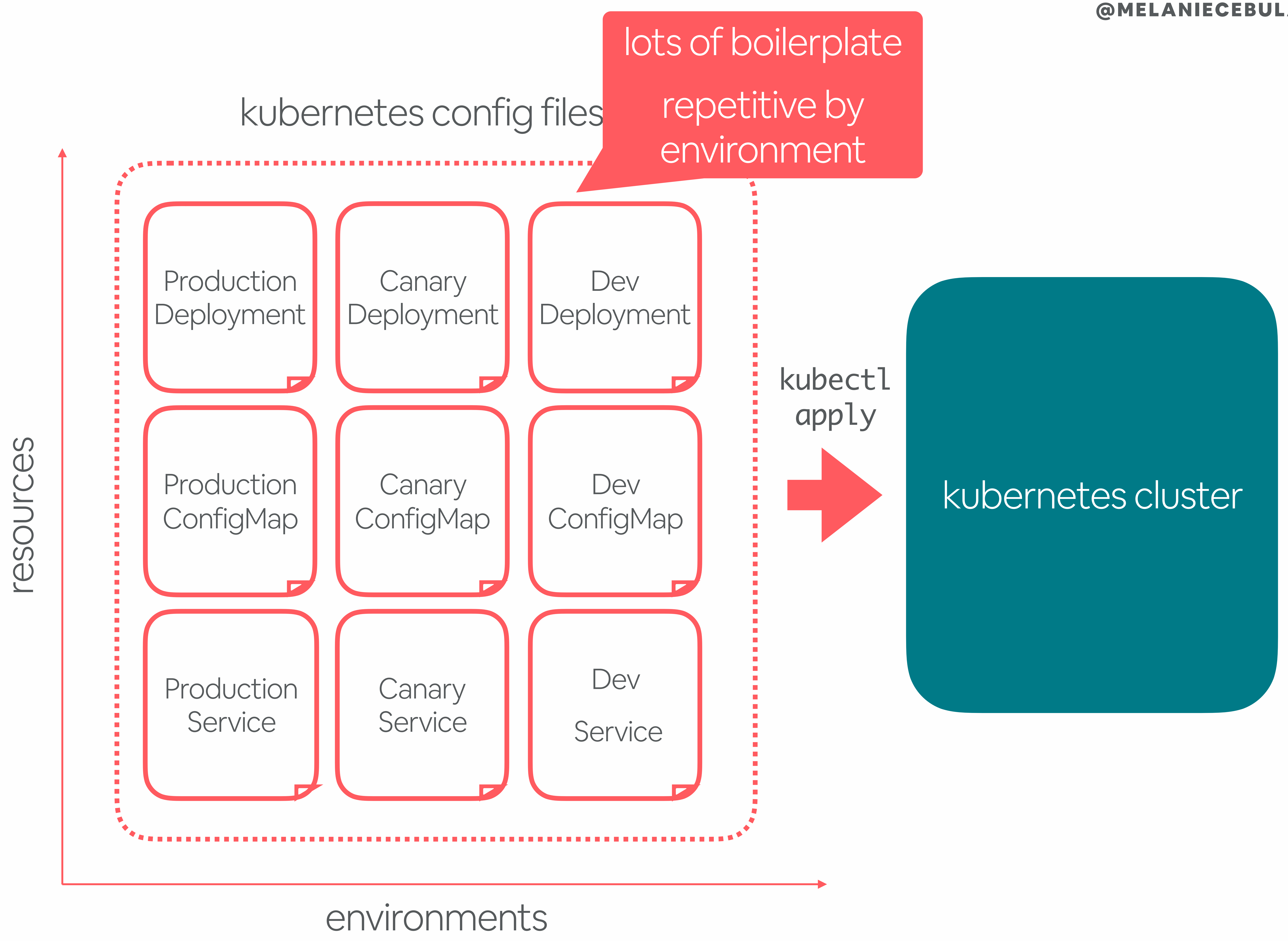
- complex configuration $\rightarrow$ generating kubernetes files
- complex tooling $\rightarrow$ kubectl wrapper
- open issues $\rightarrow$ contribute to open source and custom controllers
- integrating with your current infrastructure $\rightarrow$ custom controllers and custom resources

GENERATING KUBERNETES FILES

kubernetes


kubernetes config files
Production Deployment
Canary Deployment
Dev Deployment
Production ConfigMap
Canary ConfigMap
Dev ConfigMap
Production Service
Canary Service
Dev Service
kubectl apply
kubernetes cluster

kubernetes


kubernetes config files
lots of boilerplate
repetitive by environment
Production Deployment
Canary Deployment
Dev Deployment
Production ConfigMap
Canary ConfigMap
Dev ConfigMap
Production Service
Canary Service
Dev Service
resources
environments
kubectl apply
kubernetes cluster

Reducing boilerplate

OUR REQUIREMENTS

- **Prefer templating** over file inheritance
- Input should be **optionally templated YAML** files
- Make it easier to **migrate legacy services**
- Make it easier to **retrain >1000 engineers**

Reducing boilerplate

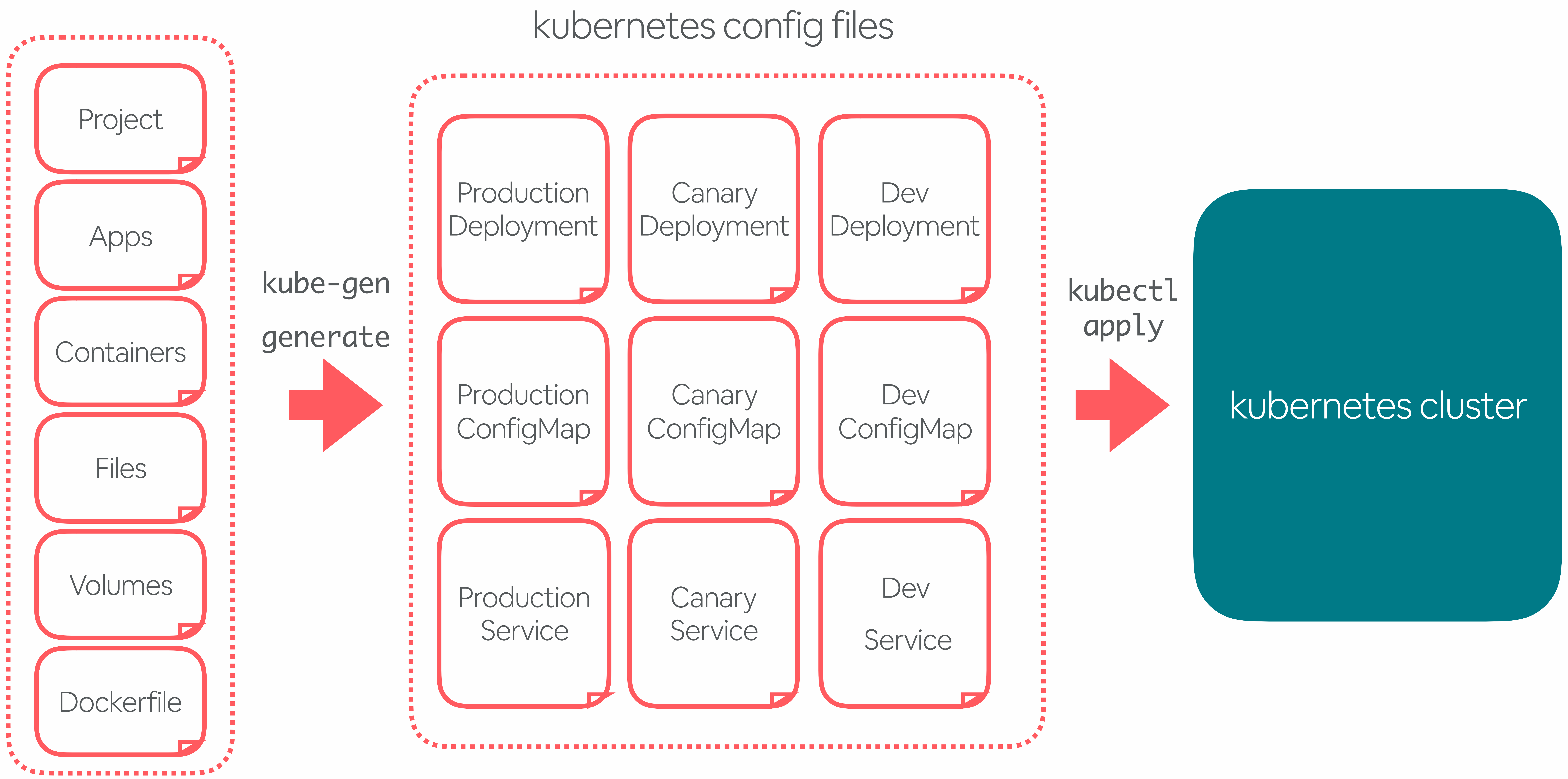
WHAT WE WENT WITH

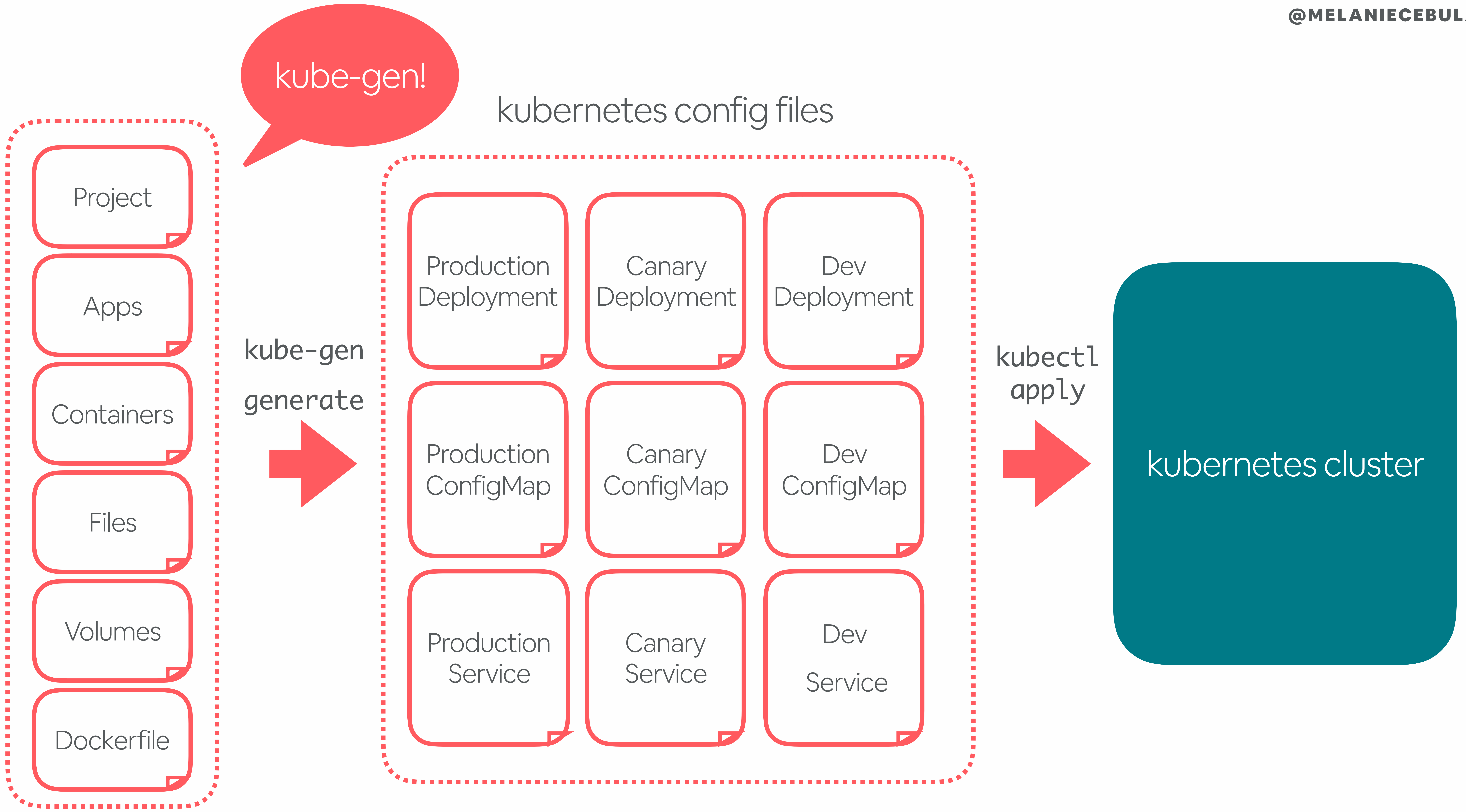
- Use custom files that map legacy concepts to k8s
- Support go templating
- Tools handle both k8s files and custom files

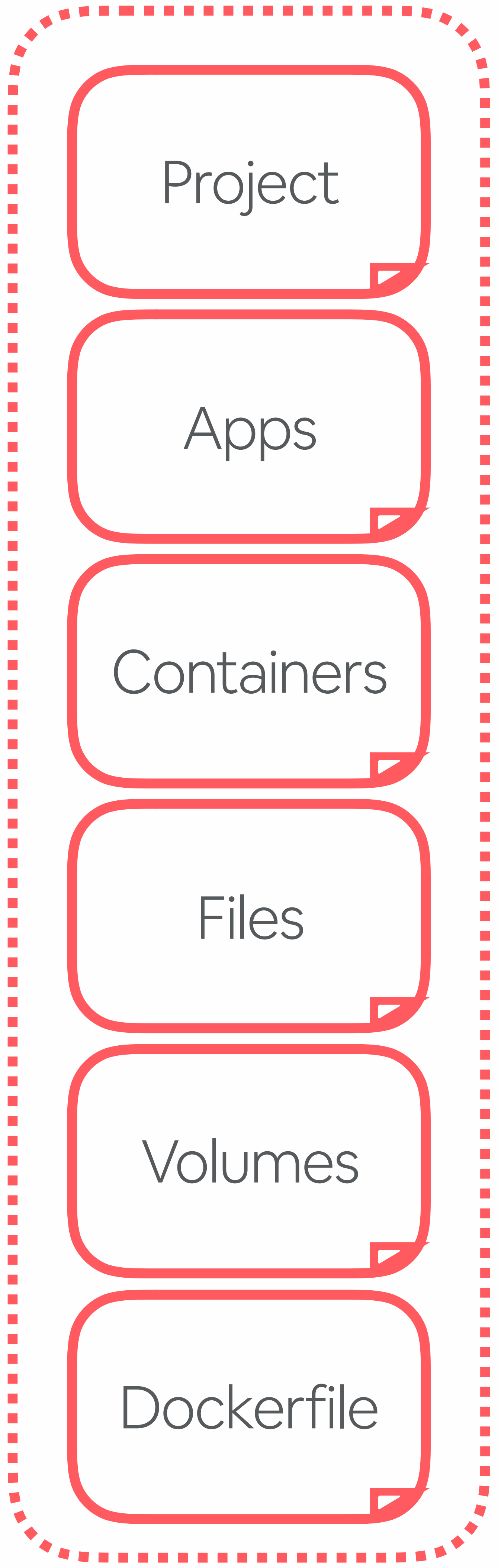
Reducing boilerplate

OPEN SOURCE OPTIONS

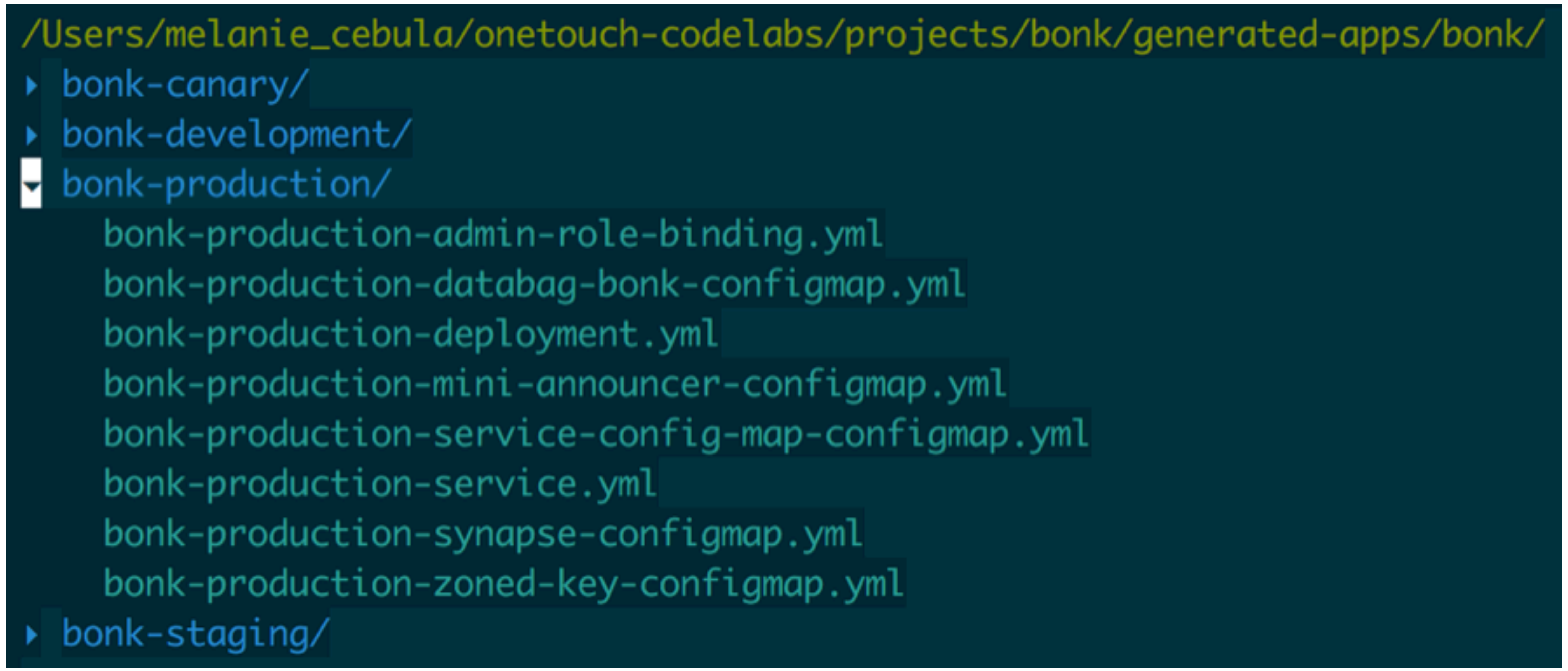
1. Combine with package management (ex: helm)
2. Override configuration via file inheritance (ex: kustomize)
3. Override configuration via templating (ex: kapitan)


kubernetes config files
Project
Apps
Containers
Files
Volumes
Dockerfile
kube-gen generate
Production Deployment
Canary Deployment
Dev Deployment
Production ConfigMap
Canary ConfigMap
Dev ConfigMap
Production Service
Canary Service
Dev Service
kubectl apply
kubernetes cluster

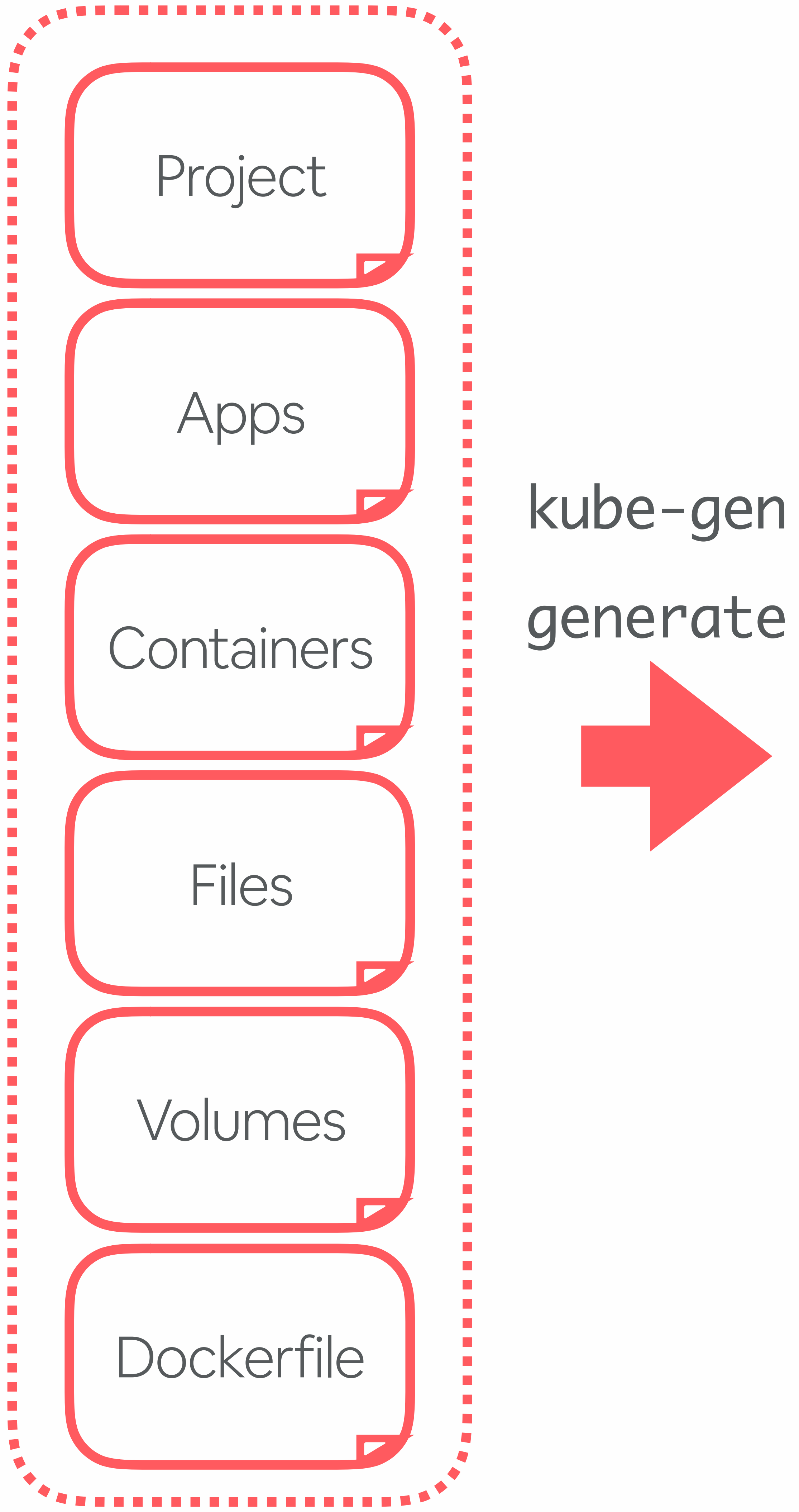

kube-gen!
kubernetes config files
Project
Apps
Containers
Files
Volumes
Dockerfile
kube-gen generate
Production Deployment
Canary Deployment
Dev Deployment
Production ConfigMap
Canary ConfigMap
Dev ConfigMap
Production Service
Canary Service
Dev Service
kubectl apply
kubernetes cluster

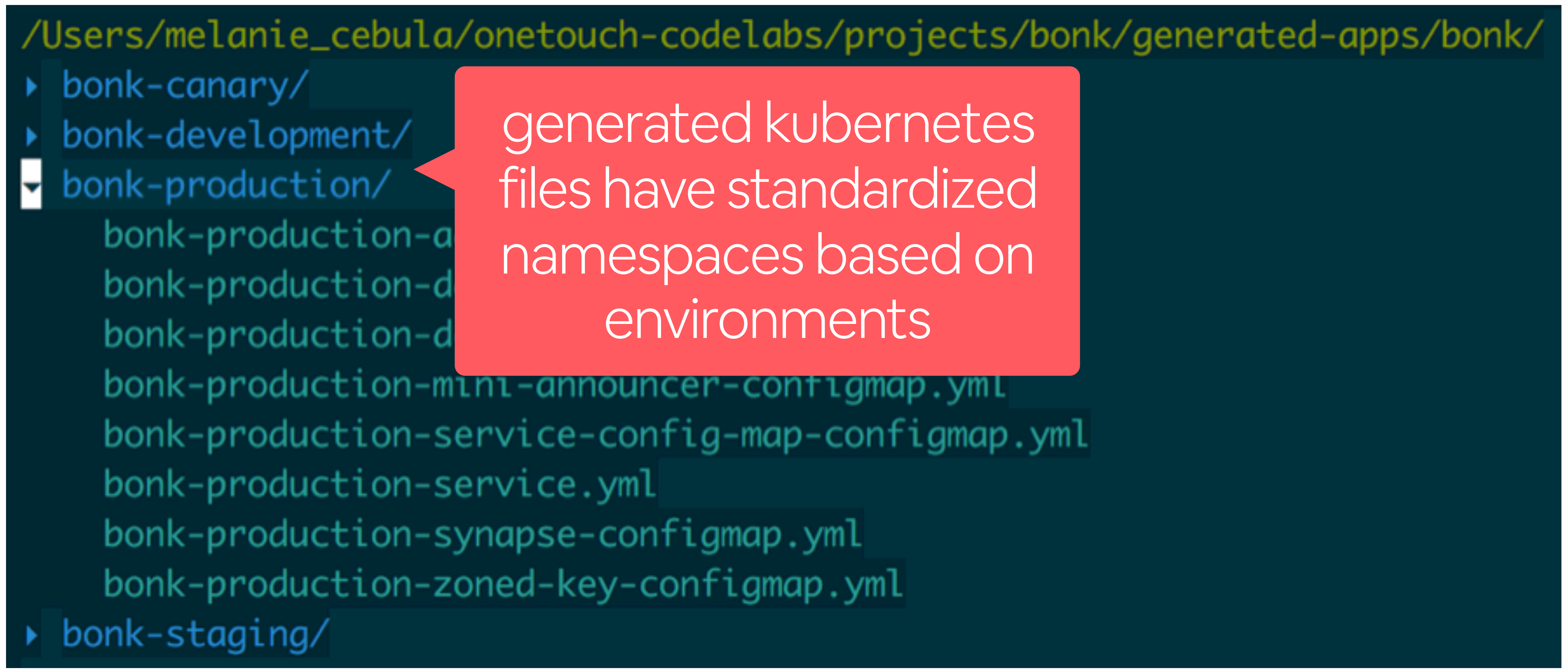


kube-gen generate



```
/Users/melanie_cebula/onetouch-codelabs/projects/bonk/generated-apps/bonk/
▸ bonk-canary/
▸ bonk-development/
▾ bonk-production/
    bonk-production-admin-role-binding.yml
    bonk-production-databag-bonk-configmap.yml
    bonk-production-deployment.yml
    bonk-production-mini-announcer-configmap.yml
    bonk-production-service-config-map-configmap.yml
    bonk-production-service.yml
    bonk-production-synapse-configmap.yml
    bonk-production-zoned-key-configmap.yml
▸ bonk-staging/
```



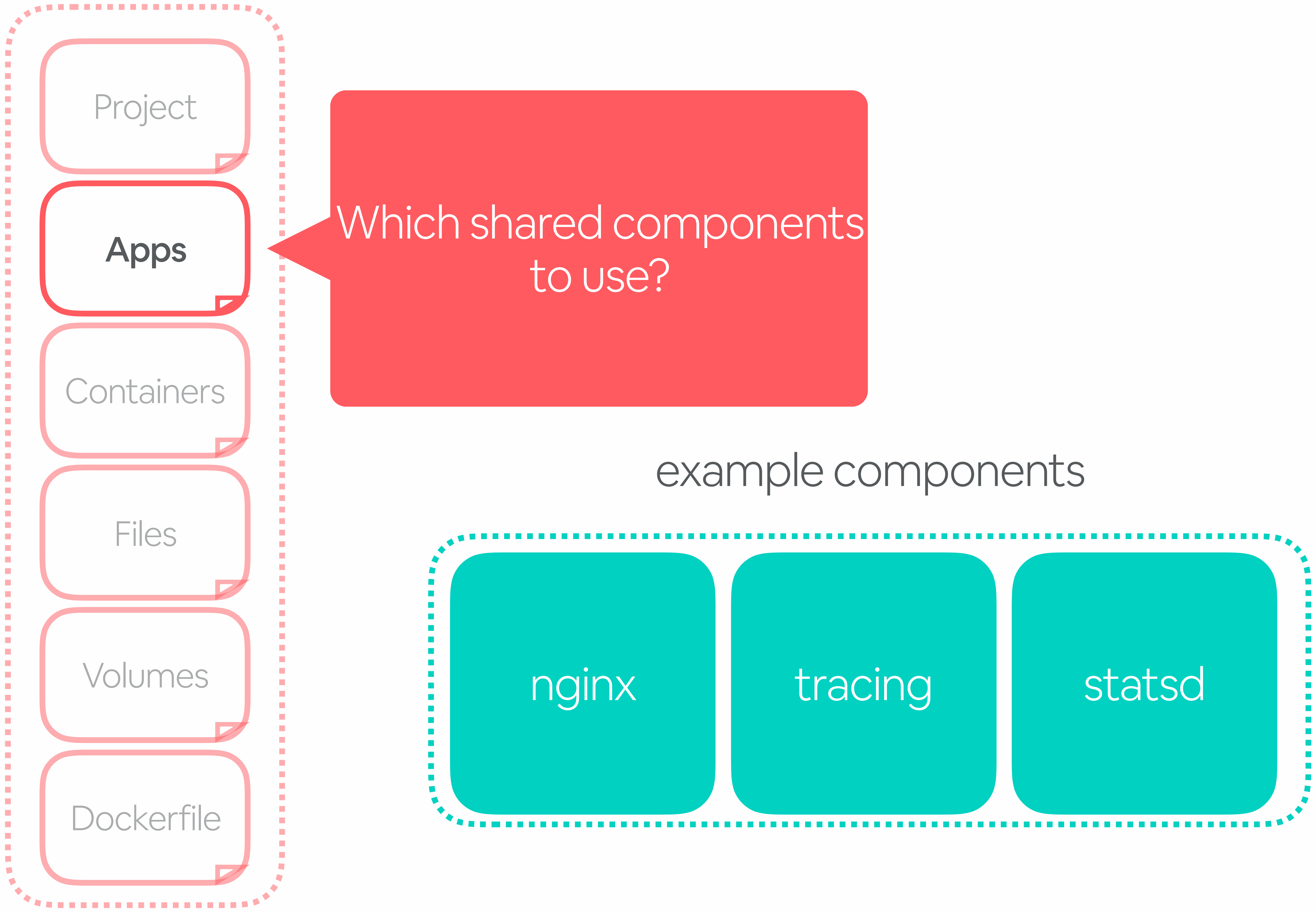


```
/Users/melanie_cebula/onetouch-codelabs/projects/bonk/generated-apps/bonk/
▸ bonk-canary/
▸ bonk-development/
▾ bonk-production/
    bonk-production-a
    bonk-production-d
    bonk-production-d
    bonk-production-mini-announcer-configmap.yml
    bonk-production-service-config-map-configmap.yml
    bonk-production-service.yml
    bonk-production-synapse-configmap.yml
    bonk-production-zoned-key-configmap.yml
▸ bonk-staging/
```

generated kubernetes files have standardized namespaces based on environments

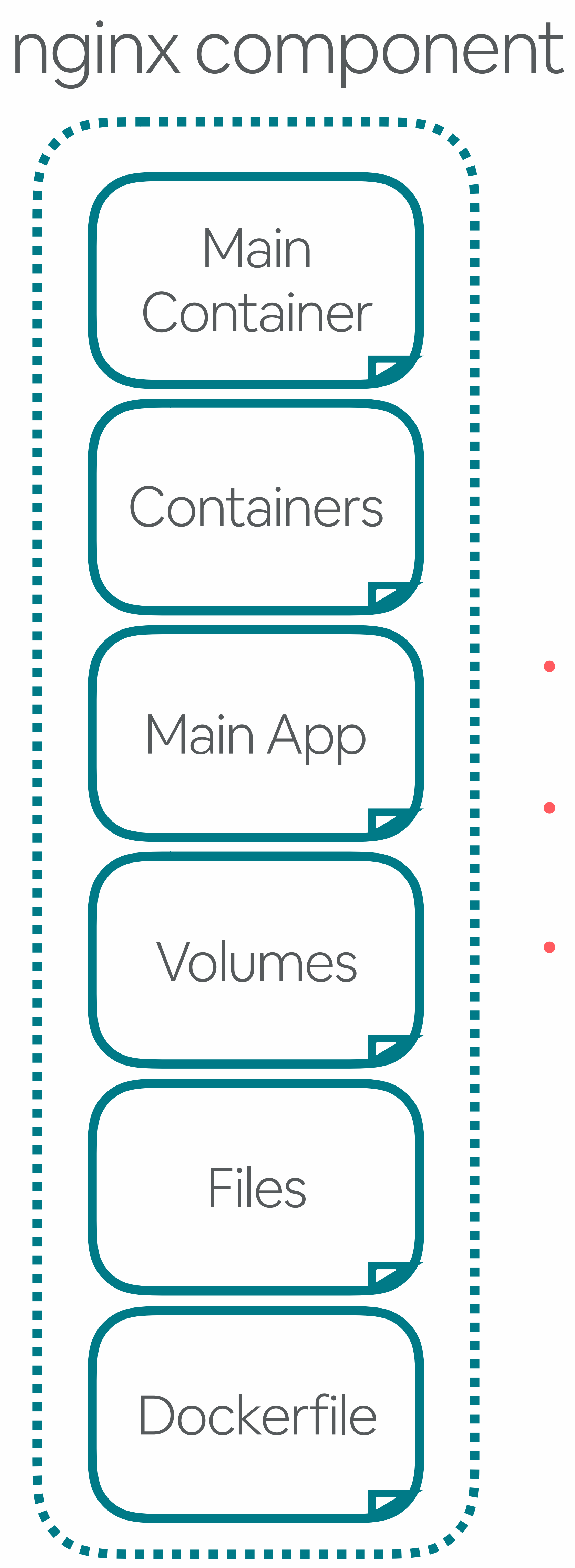
kube-gen

COMPONENTS



kube-gen

COMPONENTS



- common patterns are abstracted into a component
- component yaml merged into project
- can require or set default params

Takeaways

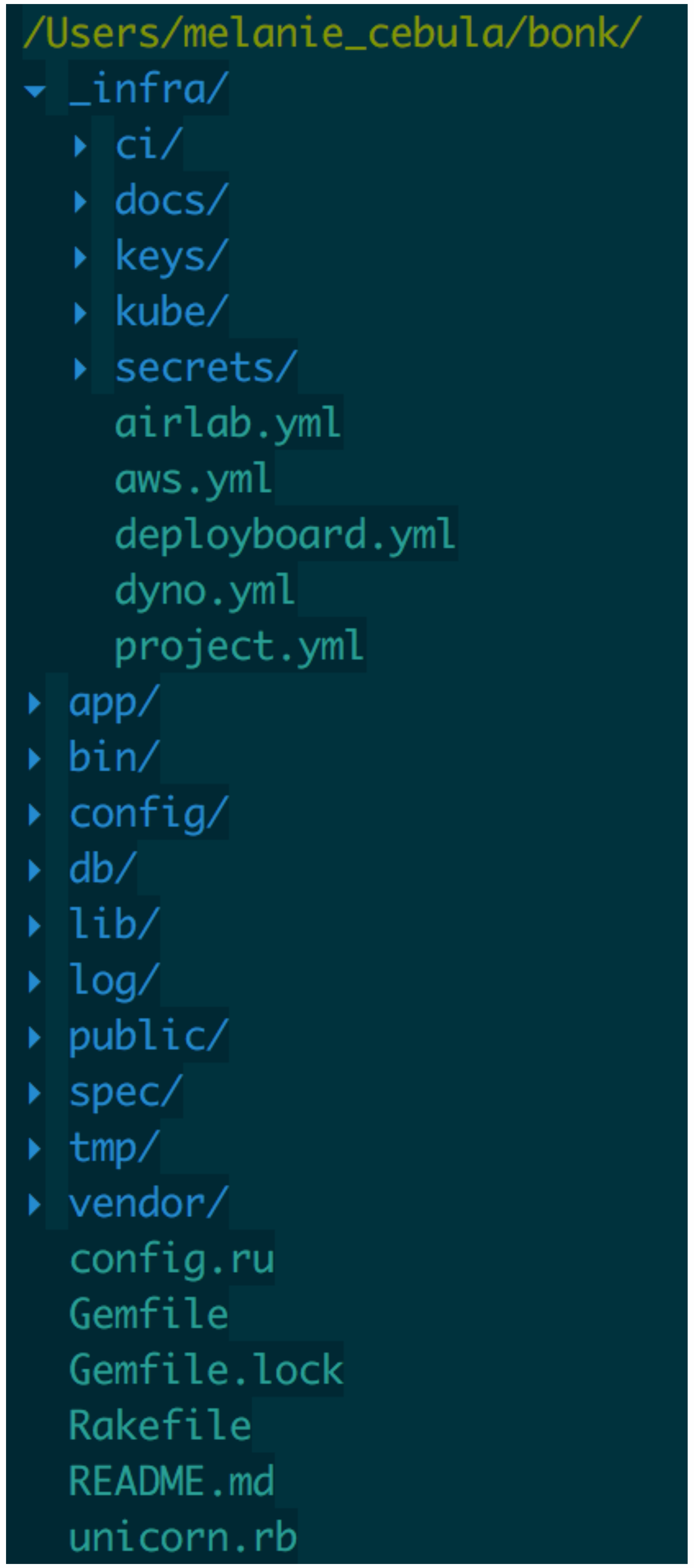
- Reduce kubernetes boilerplate
- Standardize on environments and namespaces

GENERATING SERVICE BOILERPLATE

Everything about a service is in one place in git, and managed with one process.

Configuration

LIVES IN ONE PLACE



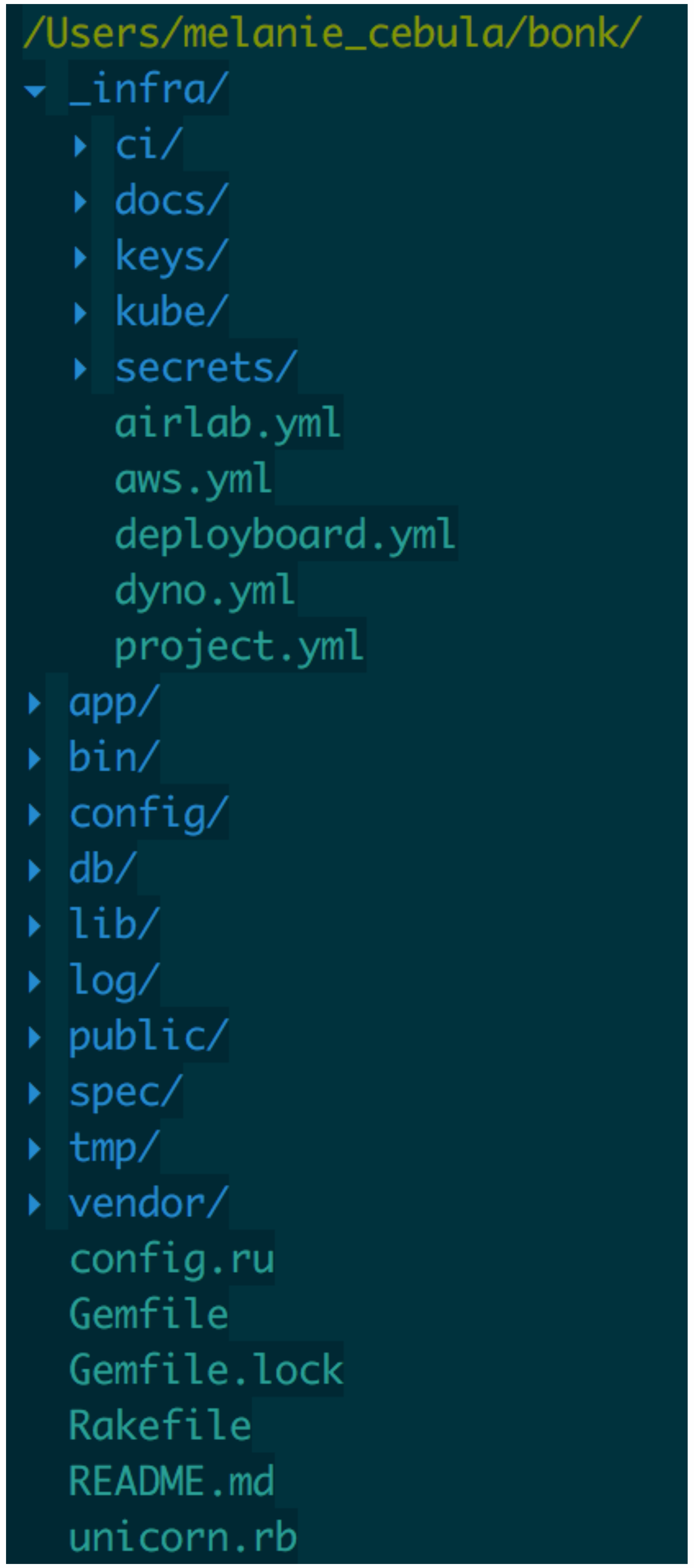
```
/Users/melanie_cebula/bonk/
▾ _infra/
  ▸ ci/
  ▸ docs/
  ▸ keys/
  ▸ kube/
  ▸ secrets/
    airlab.yml
    aws.yml
    deployboard.yml
    dyno.yml
    project.yml
▸ app/
▸ bin/
▸ config/
▸ db/
▸ lib/
▸ log/
▸ public/
▸ spec/
▸ tmp/
▸ vendor/
  config.ru
  Gemfile
  Gemfile.lock
  Rakefile
  README.md
  unicorn.rb
```

Everything about a service is in one place in git

- All configuration lives in `_infra` alongside project code
- Edit code and configuration with one pull request
- Easy to add new configuration
- Statically validated in CI/CD

Configuration

LIVES IN ONE PLACE



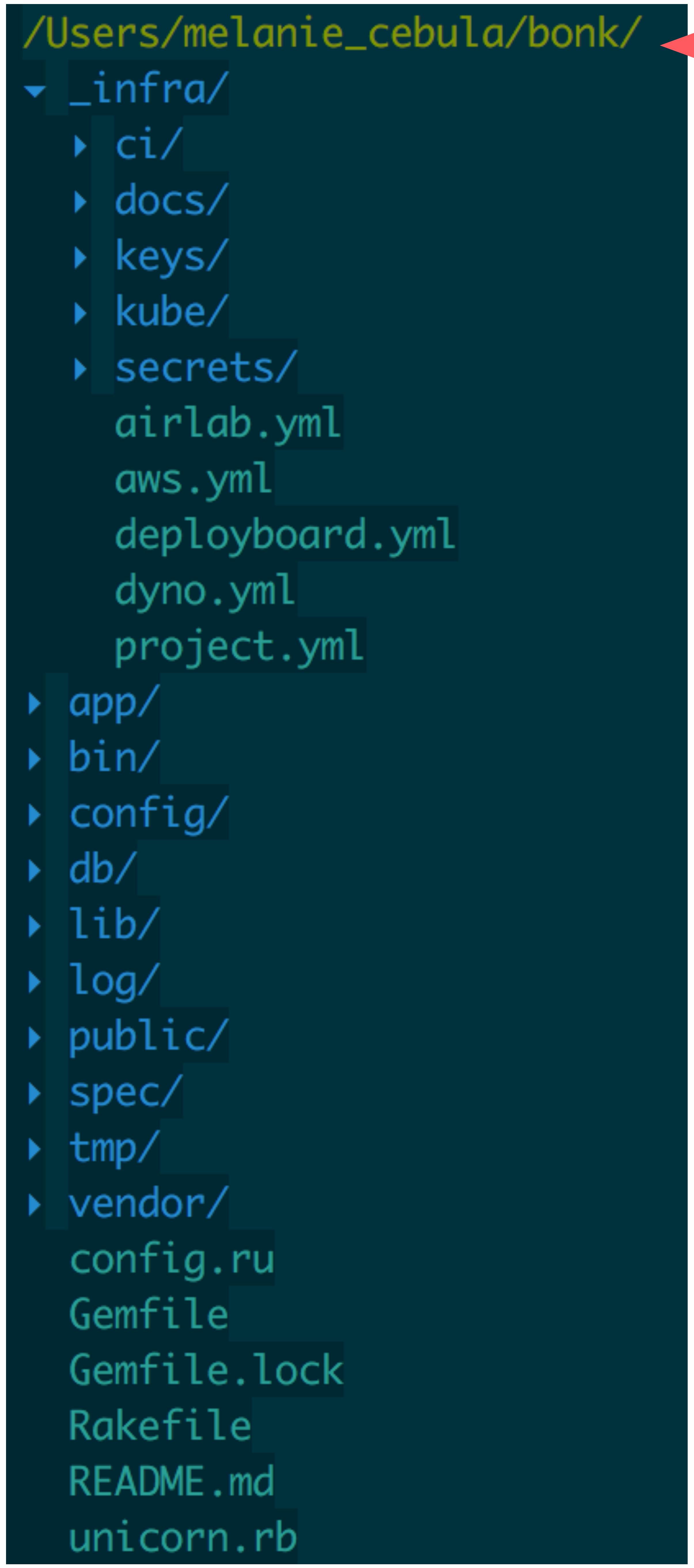
```
/Users/melanie_cebula/bonk/
▾ _infra/
  ▸ ci/
  ▸ docs/
  ▸ keys/
  ▸ kube/
  ▸ secrets/
    airlab.yml
    aws.yml
    deployboard.yml
    dyno.yml
    project.yml
▸ app/
▸ bin/
▸ config/
▸ db/
▸ lib/
▸ log/
▸ public/
▸ spec/
▸ tmp/
▸ vendor/
  config.ru
  Gemfile
  Gemfile.lock
  Rakefile
  README.md
  unicorn.rb
```

What we support:

- kube-gen files
- framework boilerplate
- API boilerplate
- CI/CD
- docs
- AWS IAM roles
- project ownership
- storage
- .. and more!

Configuration

LIVES IN ONE PLACE

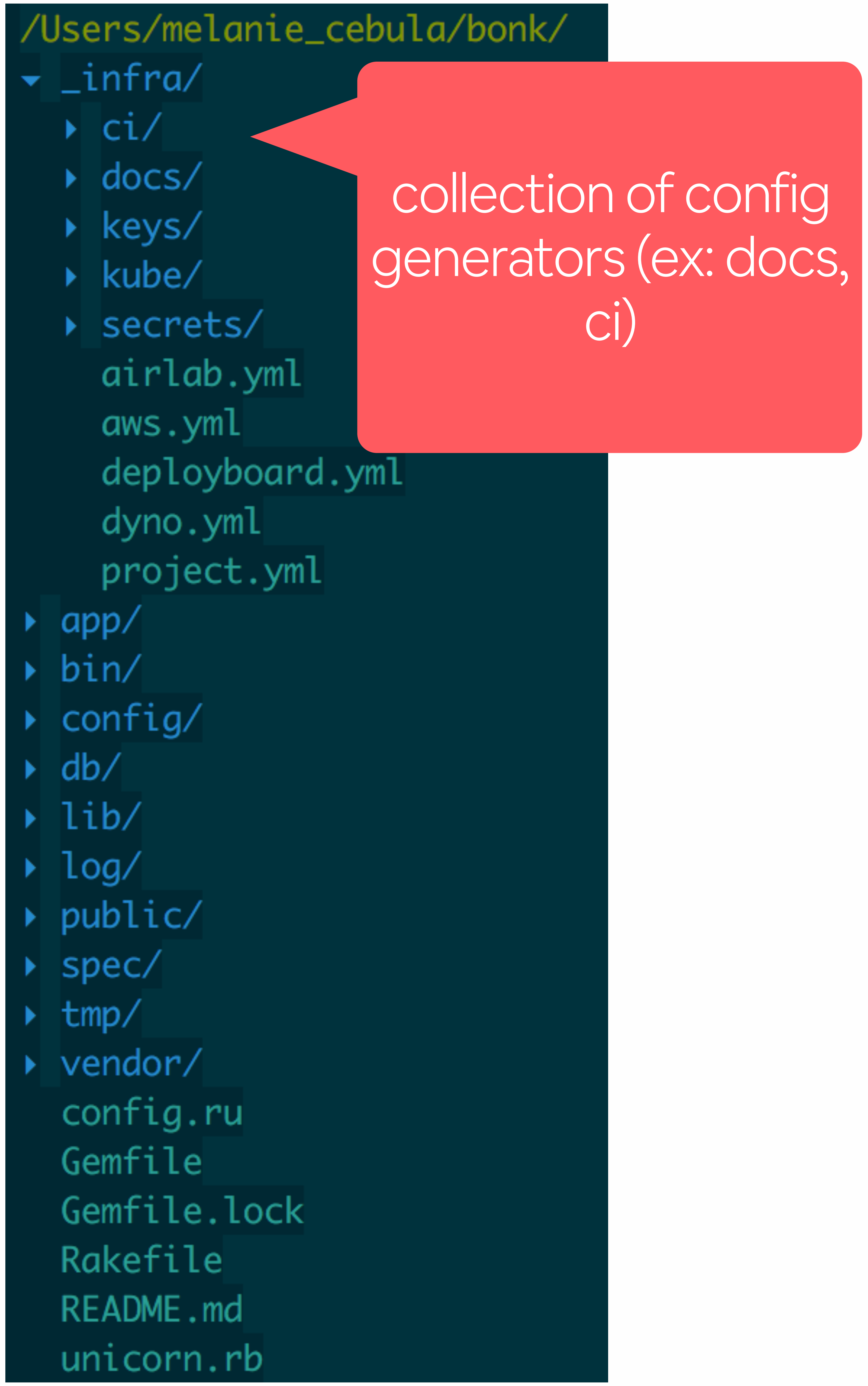


```
/Users/melanie_cebula/bonk/
▾ _infra/
  ▸ ci/
  ▸ docs/
  ▸ keys/
  ▸ kube/
  ▸ secrets/
    airlab.yml
    aws.yml
    deployboard.yml
    dyno.yml
    project.yml
▸ app/
▸ bin/
▸ config/
▸ db/
▸ lib/
▸ log/
▸ public/
▸ spec/
▸ tmp/
▸ vendor/
  config.ru
  Gemfile
  Gemfile.lock
  Rakefile
  README.md
  unicorn.rb
```

this “hello world” service was created in one command

Configuration

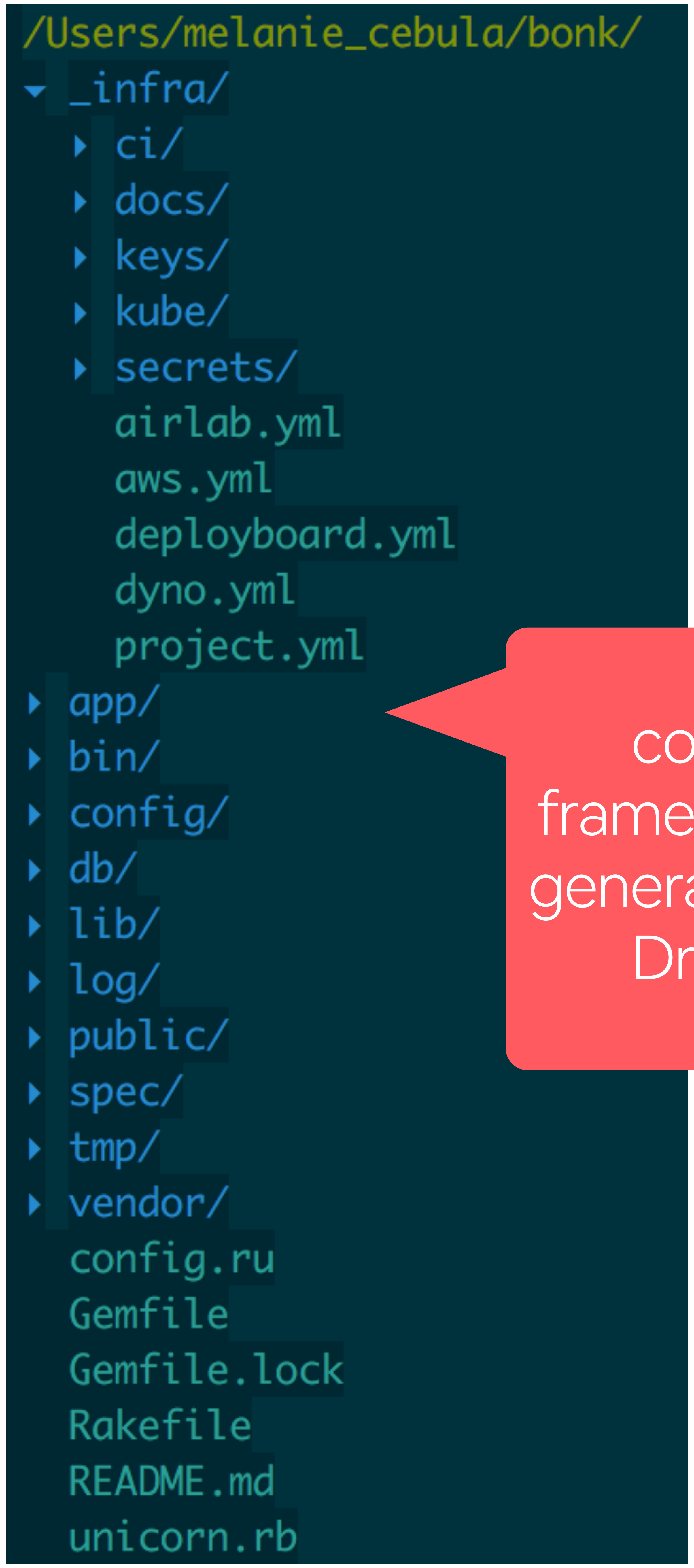
LIVES IN ONE PLACE



```
/Users/melanie_cebula/bonk/
▾ _infra/
  ▸ ci/
  ▸ docs/
  ▸ keys/
  ▸ kube/
  ▸ secrets/
    airlab.yml
    aws.yml
    deployboard.yml
    dyno.yml
    project.yml
▸ app/
▸ bin/
▸ config/
▸ db/
▸ lib/
▸ log/
▸ public/
▸ spec/
▸ tmp/
▸ vendor/
  config.ru
  Gemfile
  Gemfile.lock
  Rakefile
  README.md
  unicorn.rb
```

Configuration

LIVES IN ONE PLACE

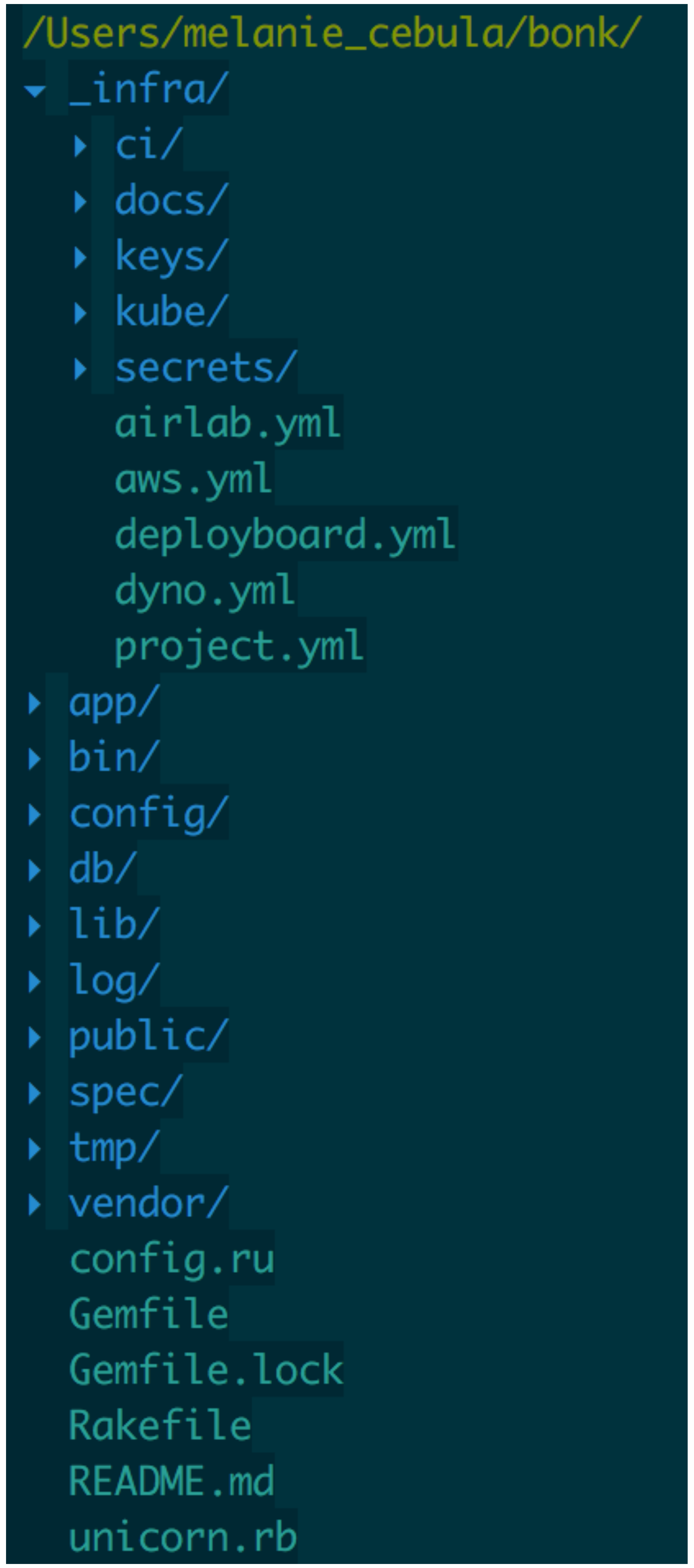


```
/Users/melanie_cebula/bonk/
▾ _infra/
  ▸ ci/
  ▸ docs/
  ▸ keys/
  ▸ kube/
  ▸ secrets/
    airlab.yml
    aws.yml
    deployboard.yml
    dyno.yml
    project.yml
▸ app/
▸ bin/
▸ config/
▸ db/
▸ lib/
▸ log/
▸ public/
▸ spec/
▸ tmp/
▸ vendor/
  config.ru
  Gemfile
  Gemfile.lock
  Rakefile
  README.md
  unicorn.rb
```

collection of framework-specific generators (ex: Rails, Dropwizard)

Configuration

CAN BE GENERATED



```
/Users/melanie_cebula/bonk/
▾ _infra/
  ▸ ci/
  ▸ docs/
  ▸ keys/
  ▸ kube/
  ▸ secrets/
    airlab.yml
    aws.yml
    deployboard.yml
    dyno.yml
    project.yml
▸ app/
▸ bin/
▸ config/
▸ db/
▸ lib/
▸ log/
▸ public/
▸ spec/
▸ tmp/
▸ vendor/
  config.ru
  Gemfile
  Gemfile.lock
  Rakefile
  README.md
  unicorn.rb
```

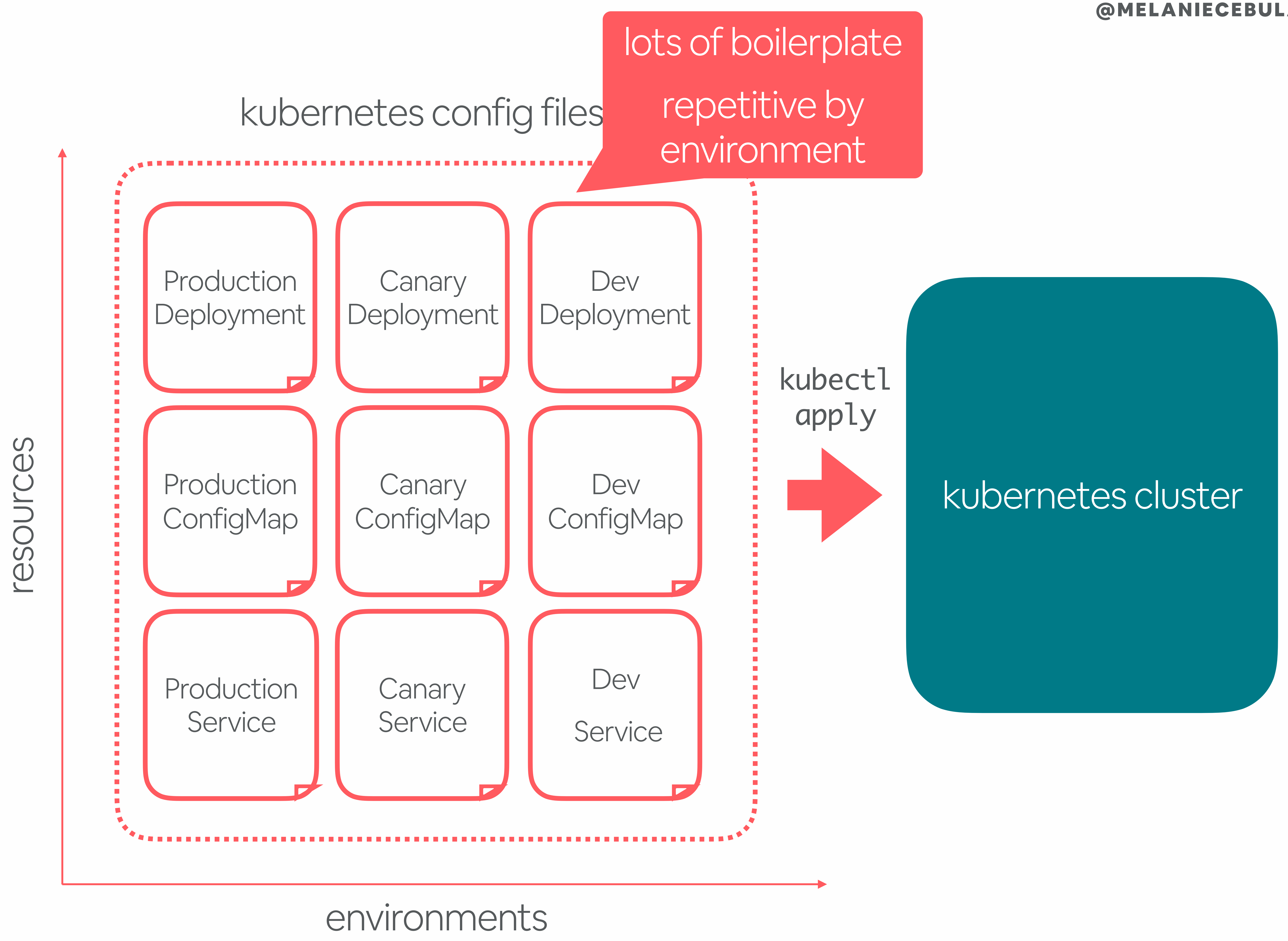
- make best practices the *default* (ex: deploy pipeline, autoscaling, docs)
- run generators individually or as a group
- support for review, update, commit

Takeaways

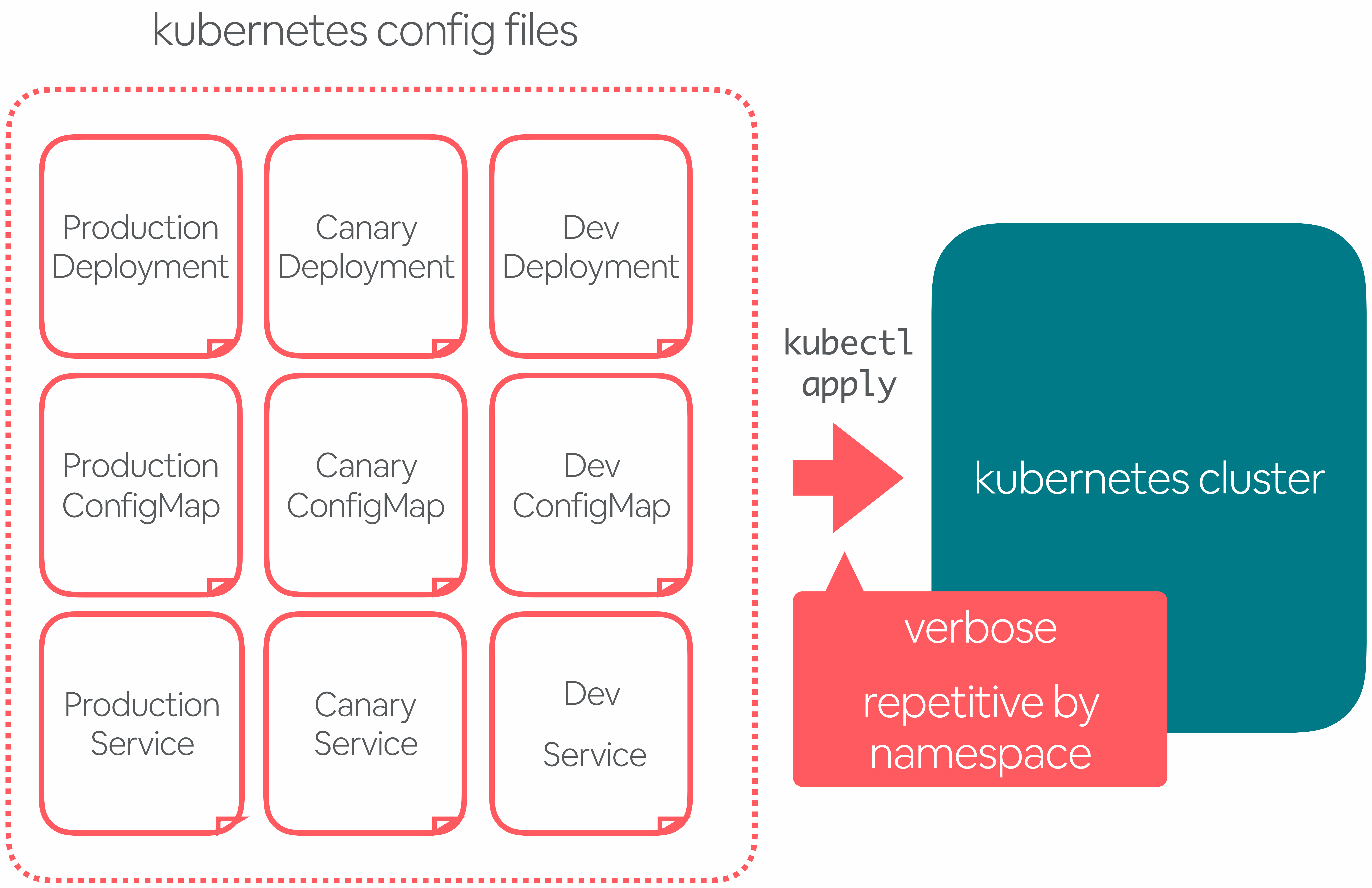
- Everything about a service should be in one place in git
- Make best practices the *default* by generating configuration

KUBECTL WRAPPER

kubernetes

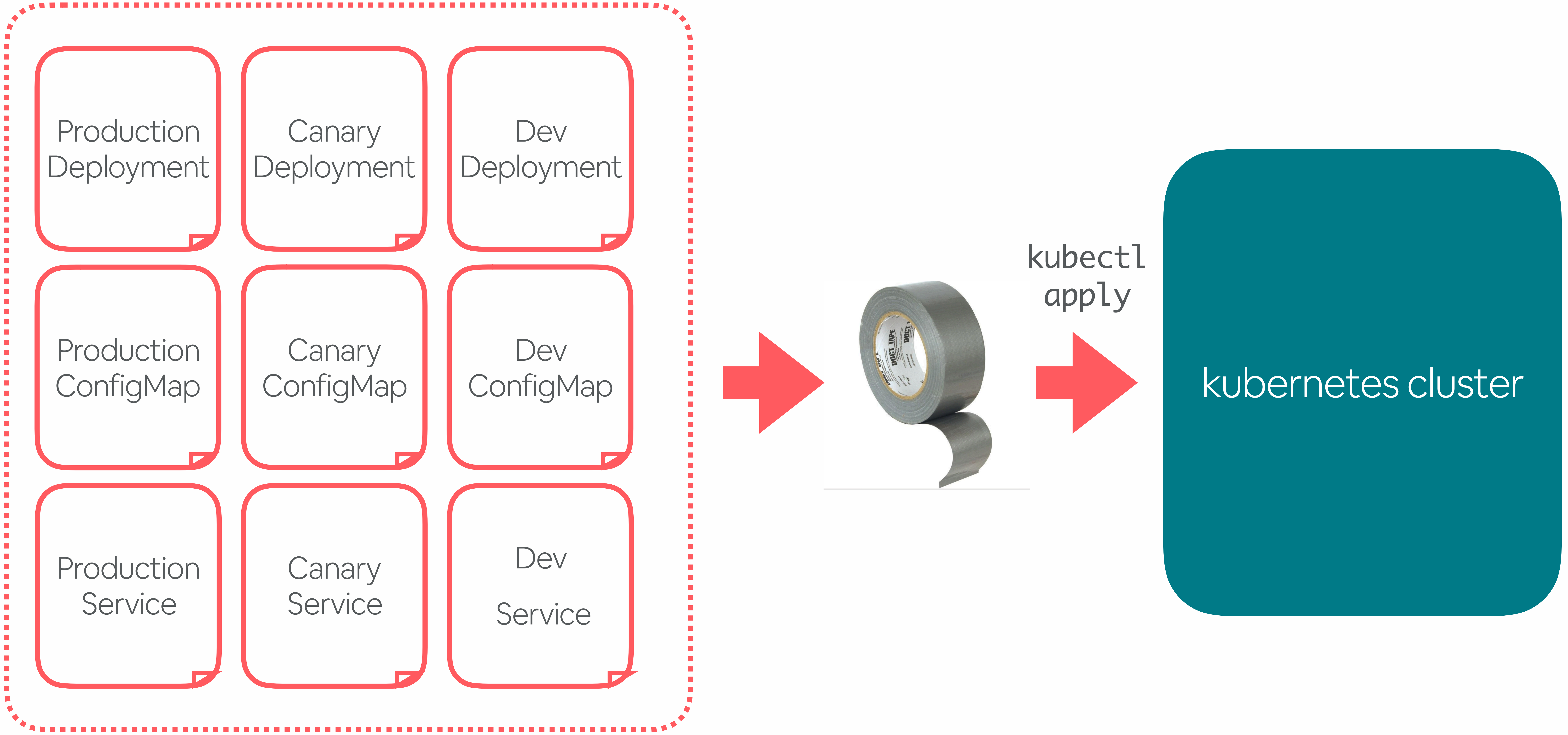

kubernetes config files
lots of boilerplate
repetitive by environment
Production Deployment
Canary Deployment
Dev Deployment
Production ConfigMap
Canary ConfigMap
Dev ConfigMap
Production Service
Canary Service
Dev Service
resources
environments
kubectl apply
kubernetes cluster

kubernetes


kubernetes config files
Production Deployment
Canary Deployment
Dev Deployment
Production ConfigMap
Canary ConfigMap
Dev ConfigMap
Production Service
Canary Service
Dev Service
kubectl apply
kubernetes cluster
verbose
repetitive by namespace

k tool

KUBECTL WRAPPER



k tool

THE ALL PURPOSE CLI WRAPPER

k tool

USES ENV VARS

- Runs in the project home directory:

```
$ cd /path/to/bonk
$ k status
```

- Environment variables for arguments:

```
$ k status ENV=staging
```

- Prints the command that it will execute:

```
$ k status ENV=staging
kubectl get pods --namespace=bonk-staging
```

standardized namespaces!

k tool

SIMPLIFIES BUILDS AND DEPLOYS

- `k generate` transforms kube-gen files to kubernetes files
- `k build` performs project build, docker build and docker push with tags
- `k deploy` creates namespace, applies/replaces kubernetes files, sleeps and checks deployment status
- can chain commands; ex: `k all`

k tool

A DEBUGGING TOOL

- defaults to random pod, main container:

```
$ k ssh ENV=staging
```

- specify particular pod, specific container:

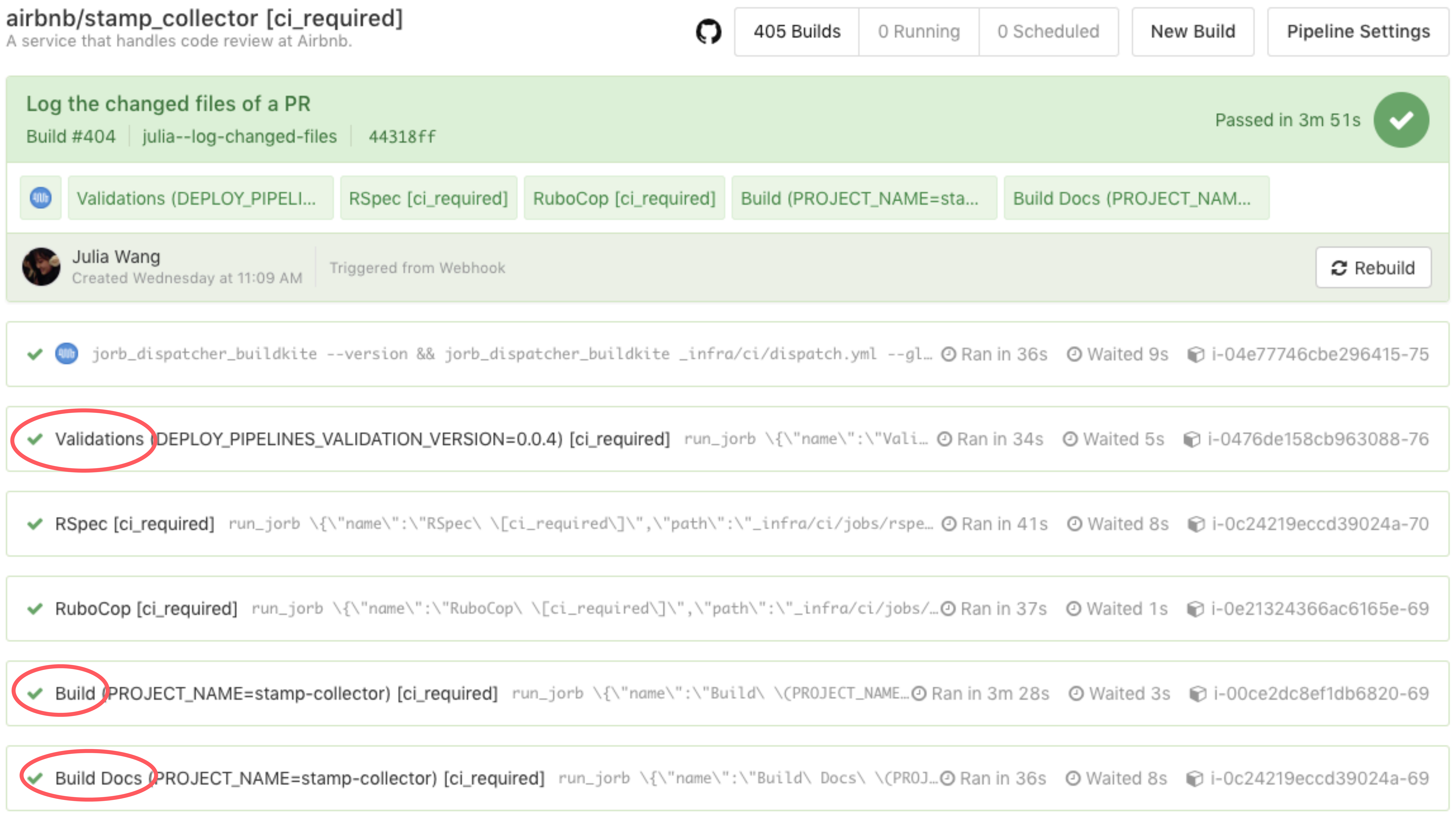
```
$ k logs ENV=staging POD=… CONTAINER=statsd-proxy
```

- automates debugging with `k diagnose`:
  1. `k status ENV=… POD=…`
  2. for each unready container, output message and last state, and get logs with `k logs CONTAINER=…`
  3. `kubectl get events` with field-selector for `POD` and filter out “Normal” events

Takeaways

- Create a wrapper for kubectl commands
- Automate common k8s workflows

CI/CD



airbnb/stamp_collector [ci_required]
A service that handles code review at Airbnb.

405 Builds | 0 Running | 0 Scheduled | New Build | Pipeline Settings

Log the changed files of a PR
Build #404 | julia--log-changed-files | 44318ff

Passed in 3m 51s

Validations (DEPLOY_PIPELI... | RSpec [ci_required] | RuboCop [ci_required] | Build (PROJECT_NAME=sta... | Build Docs (PROJECT_NAM...

Julia Wang
Created Wednesday at 11:09 AM
Triggered from Webhook
Rebuild

```
✓ jorb_dispatcher_buildkite --version && jorb_dispatcher_buildkite _infra/ci/dispatch.yml --gl... Ran in 36s  Waited 9s  i-04e77746cbe296415-75
✓ Validations (DEPLOY_PIPELINES_VALIDATION_VERSION=0.0.4) [ci_required] run_jorb \{\"name\":\"Vali... Ran in 34s  Waited 5s  i-0476de158cb963088-76
✓ RSpec [ci_required] run_jorb \{\"name\":\"RSpec\ \[ci_required\]\",\"path\":\"_infra/ci/jobs/rspe... Ran in 41s  Waited 8s  i-0c24219eccd39024a-70
✓ RuboCop [ci_required] run_jorb \{\"name\":\"RuboCop\ \[ci_required\]\",\"path\":\"_infra/ci/jobs/... Ran in 37s  Waited 1s  i-0e21324366ac6165e-69
✓ Build (PROJECT_NAME=stamp-collector) [ci_required] run_jorb \{\"name\":\"Build\ \(PROJECT_NAME... Ran in 3m 28s  Waited 3s  i-00ce2dc8ef1db6820-69
✓ Build Docs (PROJECT_NAME=stamp-collector) [ci_required] run_jorb \{\"name\":\"Build\ Docs\ \(PROJ... Ran in 36s  Waited 8s  i-0c24219eccd39024a-69
```

Build (PROJECT_NAME=stamp-collector) [ci_required]
run_jorb \{\"name\":\"Build\ \(PROJECT_NAME…
Ran in 3m 28s
Log
Artifacts
Agent
Environment
Expand groups
Collapse groups

```
Running global environment hook
Running global pre-checkout hook
Preparing working dire
Running global checkou
Running global post-ch
Running commands
Running 'install kube-
Running 'k generate'
Running 'k build'
Running 'create titanic build info in ./config'
Running 'set build tags'
Running 'tar artifact (project only)'
Running 'upload titanic artifact'
Running 'k clean'
Running global pre-exit hook
```

Each step in our CI /CD jobs are RUN steps in a build Dockerfile

✓ Build (PROJECT_NAME=stamp-collector) [ci_required] run_jorb \{\"name\":\"Build\ \(PROJECT_NAME... Ran in 3m 28s

Log
Artifacts
Agent
Environment
Expand groups
Collapse groups

```
   1 ▸ Running global environment hook
   5 ▸ Running global pre-checkout hook
  11 ▸ Preparing working directory
  14 ▸ Running global checkout hook
  18 ▸ Running global post-checkout hook
  24 ▸ Running commands
  42 ▸ Running 'install kube-gen (
 106 ▸ Running 'k generate'
 160 ▸ Running 'k build'
1003 ▸ Running 'create titanic bui
1004 ▸ Running 'set build tags'
1008 ▸ Running 'tar artifact (project only)'
1023 ▸ Running 'upload titanic artifact'
1036 ▸ Running 'k clean'
1053 ▸ Running global pre-exit hook
```

Takeaways

- CI/CD should run the same commands that engineers run locally
- CI/CD should run in a container
- Validate configuration as part of CI/CD

DEPLOY PROCESS

A single deploy process for every change

Develop

Write code and config under your project

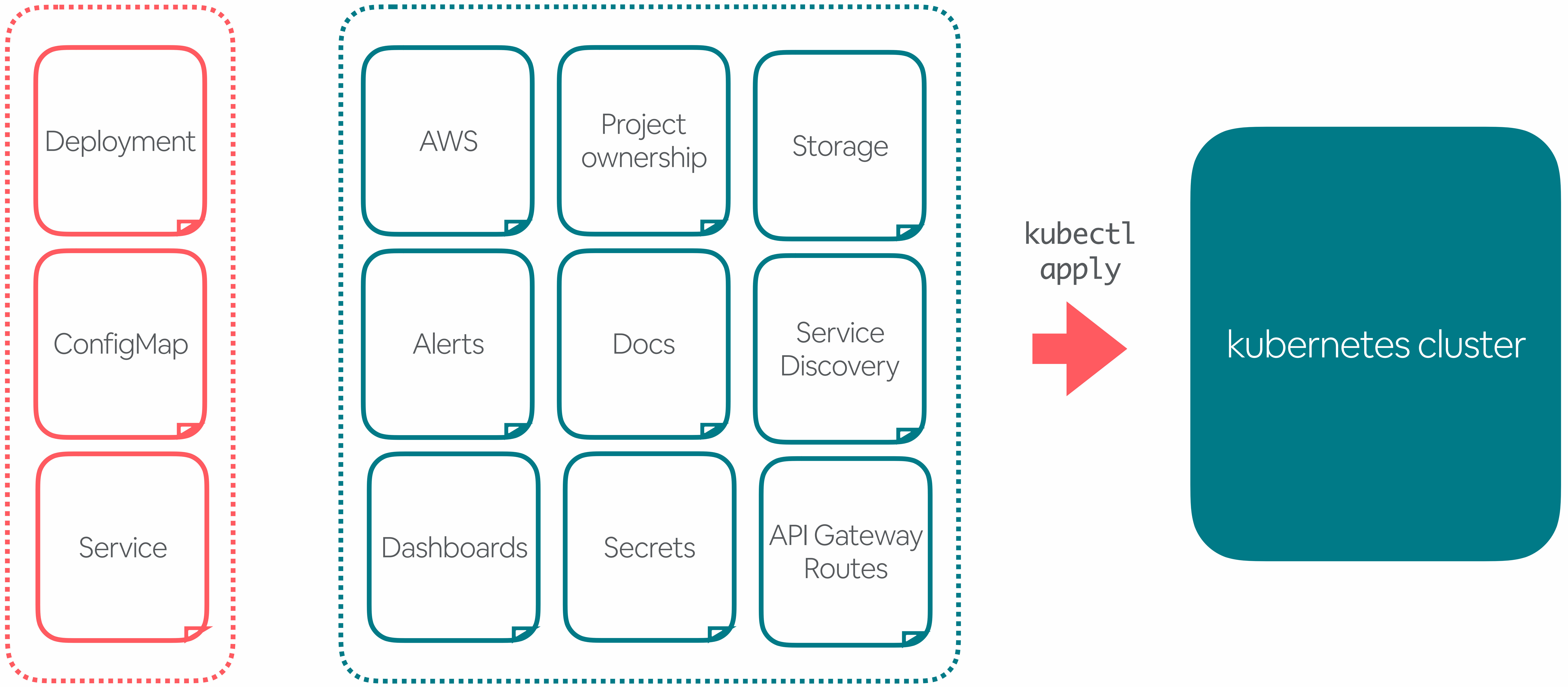
Merge

Open a PR and merge your code to master

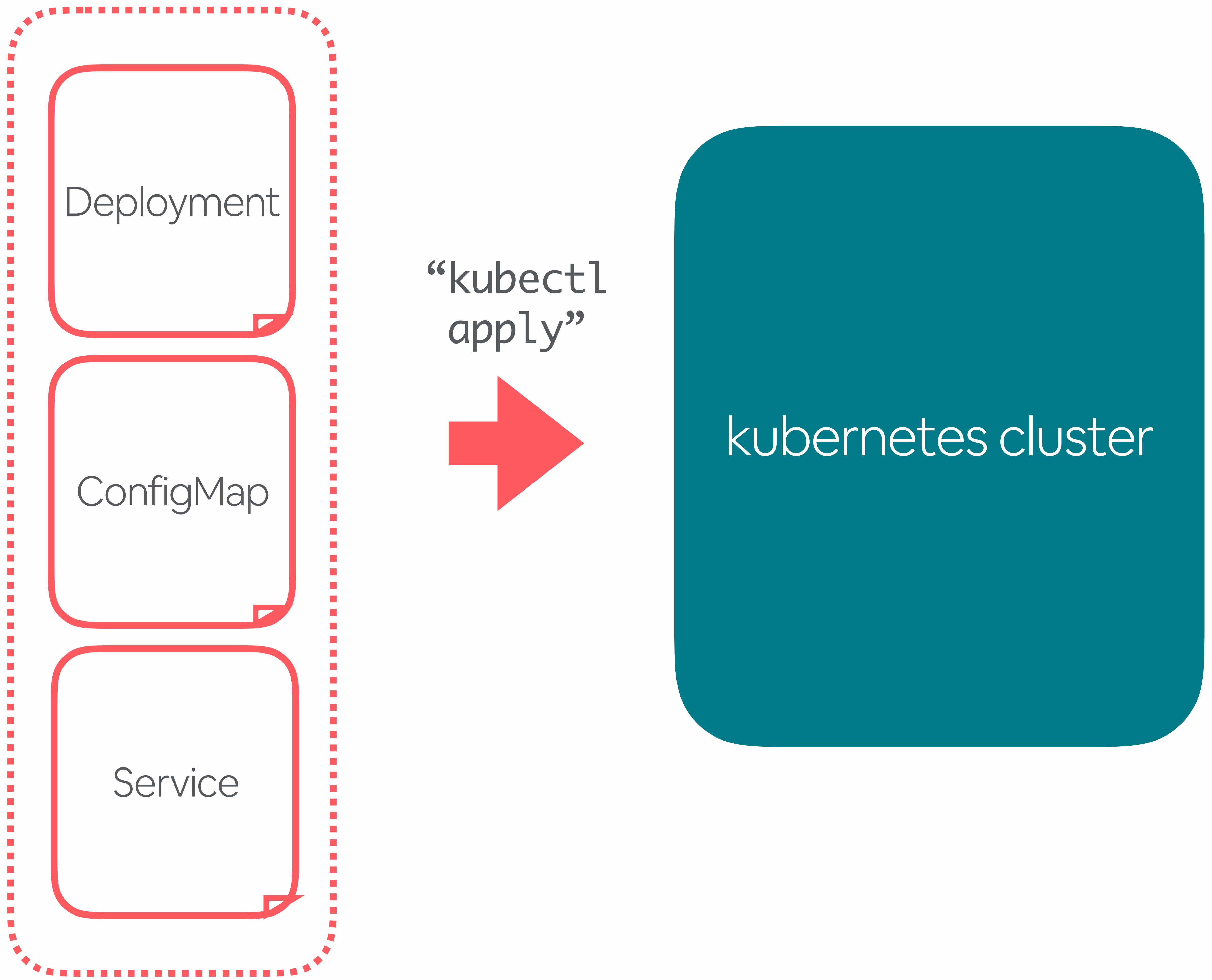
Deploy

Deploy all code and config changes

A single deploy process for every change

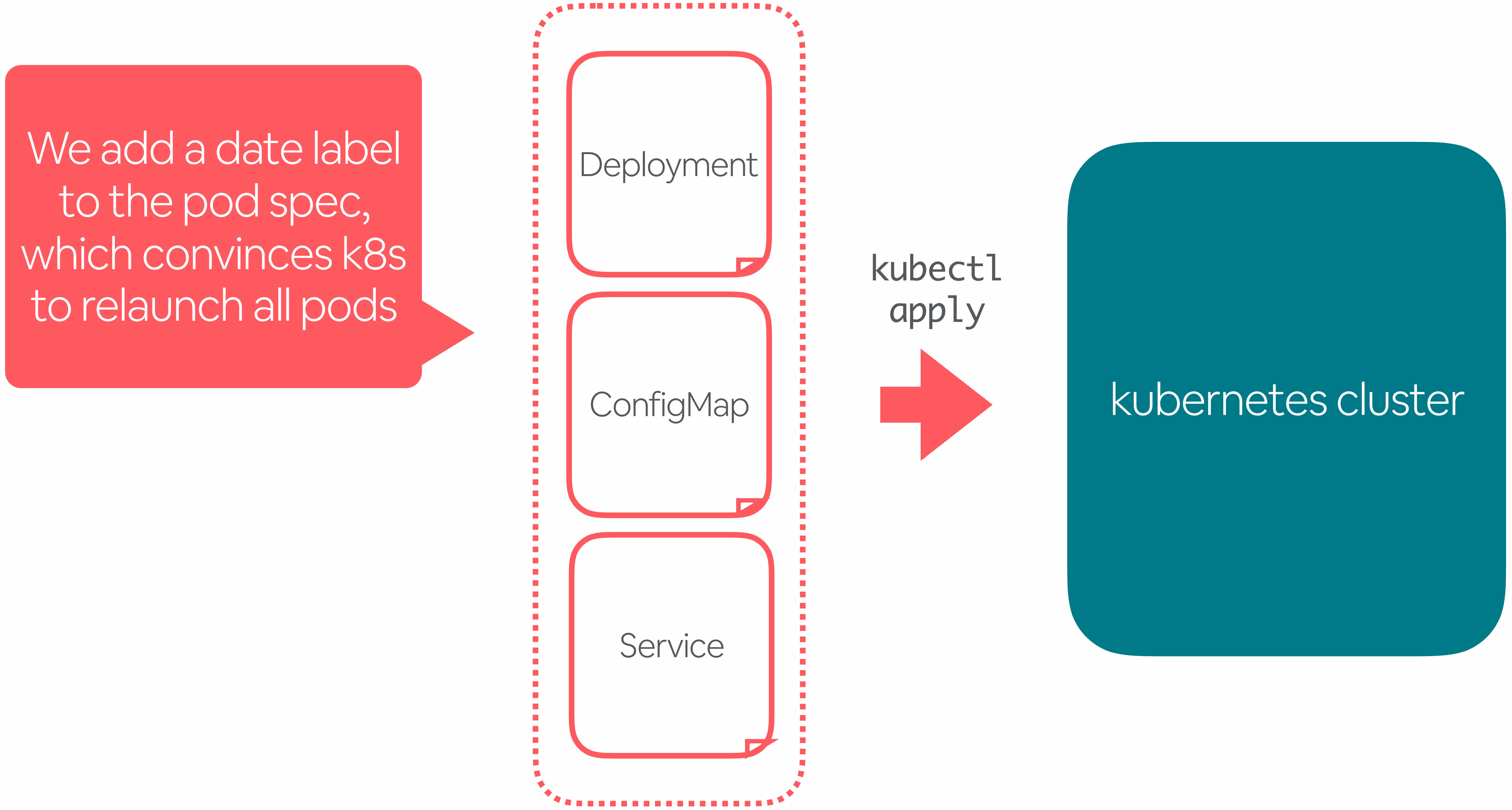


How do we apply k8s configuration?

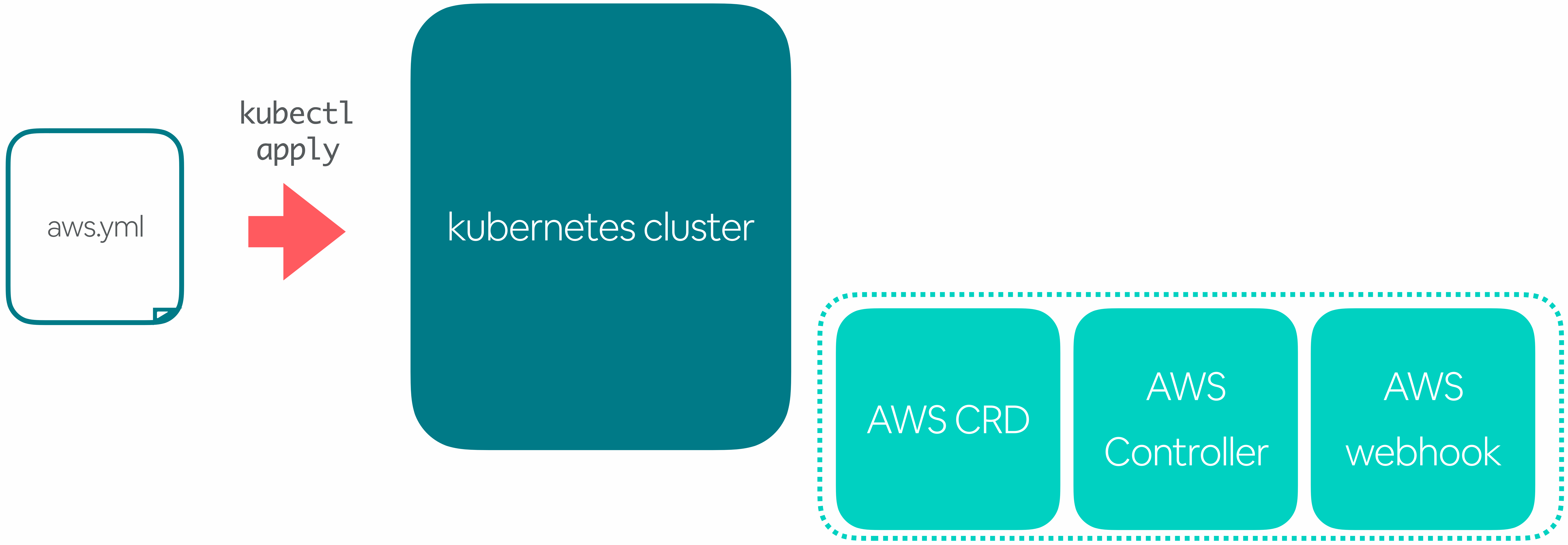


- kubectl apply all files
- in some cases where apply fails, replace files without force
- always restart pods on deploy to pick up changes
- return atomic success or failure state by sleeping and checking status

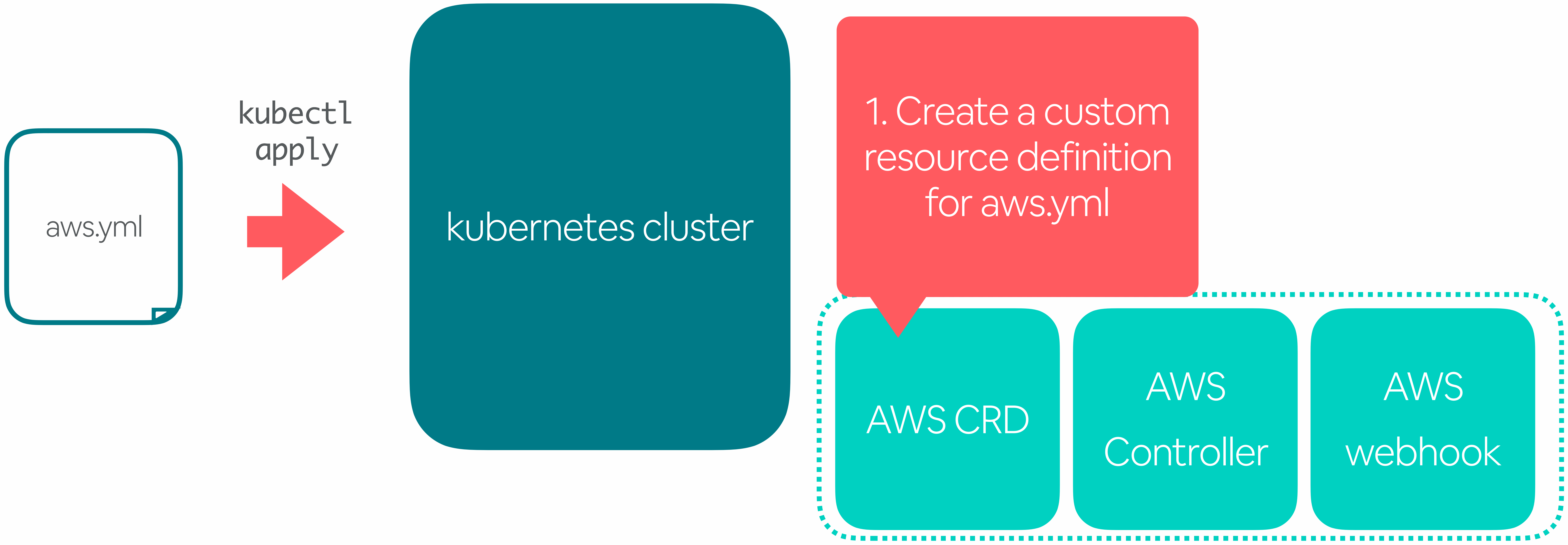
How do you always restart pods on deploy?


We add a date label to the pod spec, which convinces k8s to relaunch all pods
Deployment
ConfigMap
Service
kubectl apply
kubernetes cluster

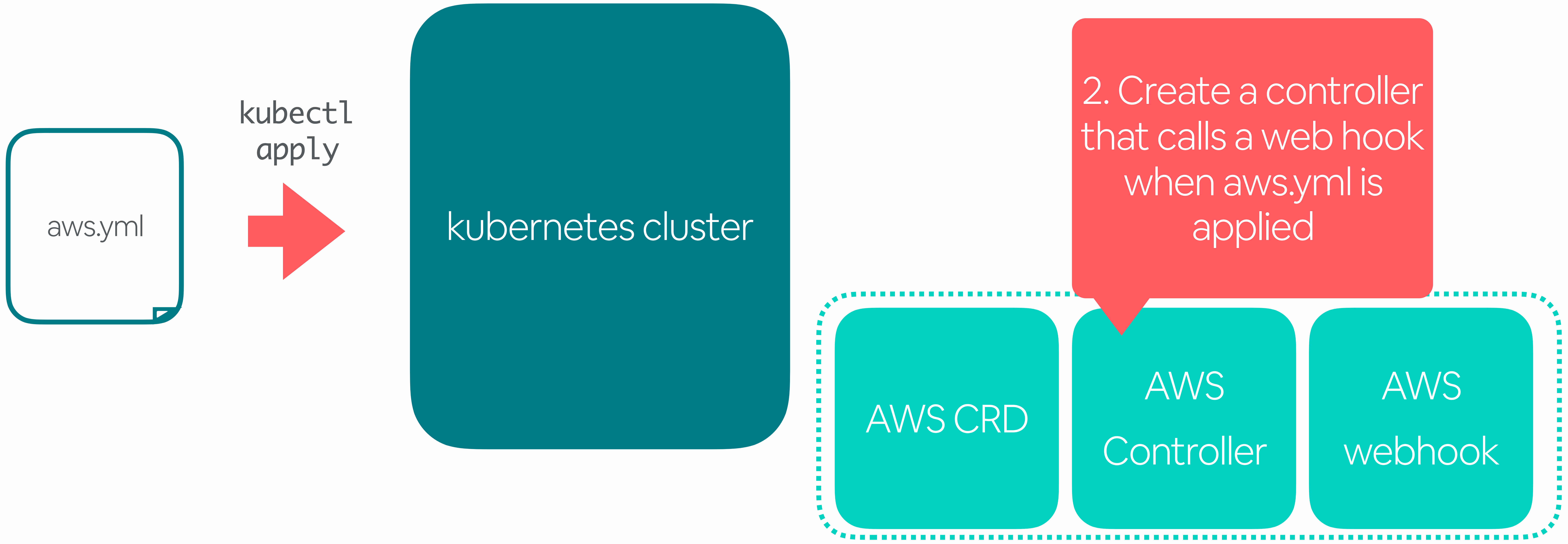
How do we apply custom configuration?



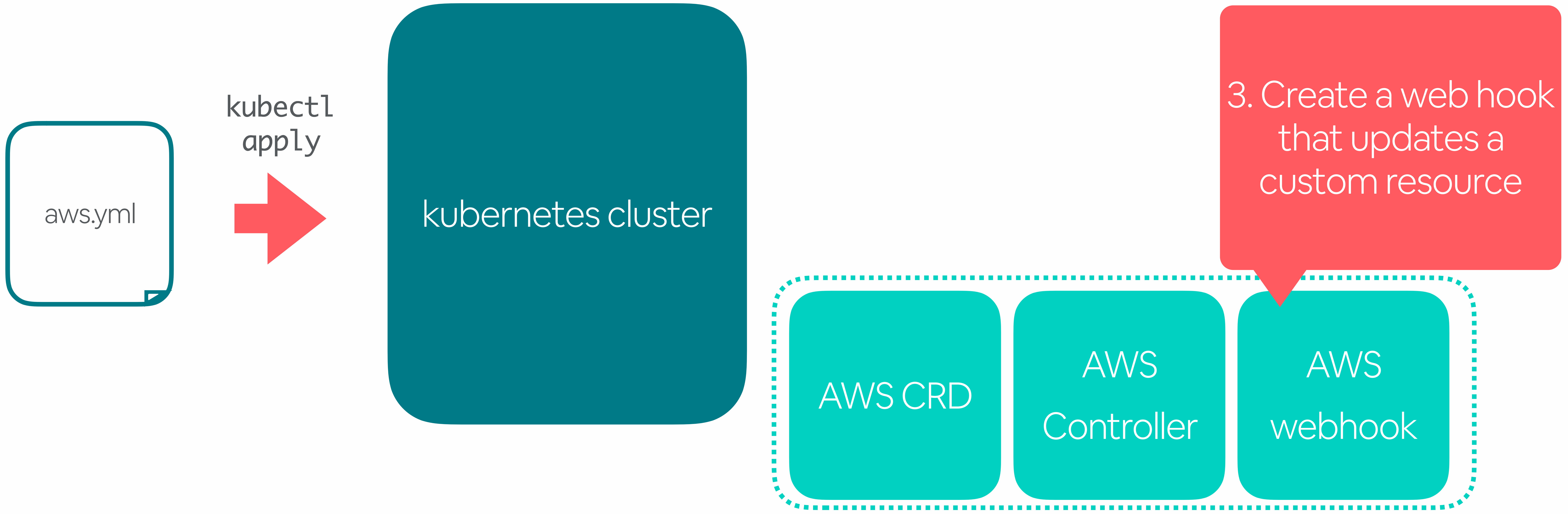
How do we apply custom configuration?



How do we apply custom configuration?



How do we apply custom configuration?



Takeaways

- Code and configuration should be deployed with the same process
- Use custom resources and custom controllers to integrate k8s with your infra

10 Takeaways

1. **Reduce kubernetes boilerplate**
2. **Standardize** on environments and namespaces
3. Everything about a service should be in **one place in git**
4. **Make best practices the default** by generating configuration
5. **Create a wrapper** for kubectl commands
6. **Automate** common k8s workflows
7. **CI/CD should run the same commands** that engineers run locally, **in a container**
8. **Validate configuration** as part of CI/CD
9. Code and configuration should be **deployed with the same process**
10. **Use custom resources and custom controllers** to integrate with your infrastructure

Thanks!

- learn more @ medium.com/airbnb-engineering
- jobs @ airbnb.com/careers
- reach me @melaniecebula