

SERVERLESS IS FOR PRAGMATISTS

CI/CD IN THE AGE OF IAC

CHASE DOUGLAS & ANNA SPYSZ, [STACKERY.IO](#)
PYTHON WEB CONFERENCE 2020

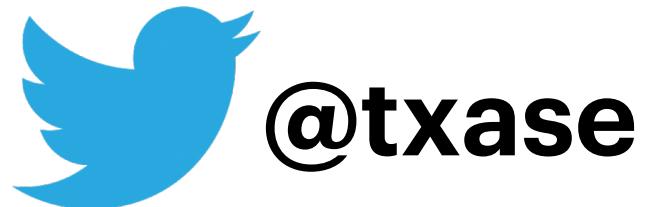
ABOUT US

CHASE DOUGLAS

CTO @ Stackery

AWS Serverless Hero 

Formerly: New Relic, Canonical, IBM



ABOUT US

ANNA SPYSZ



Software Engineer @ Stackery

AWS Certified Cloud Practitioner (mostly for re:Invent lounge access) ☕

Formerly: Writer, tech journalist



CI/CD in the age of IaC - What?

CI = Continuous integration

CD = Continuous delivery/deployment

IaC = Infrastructure as Code

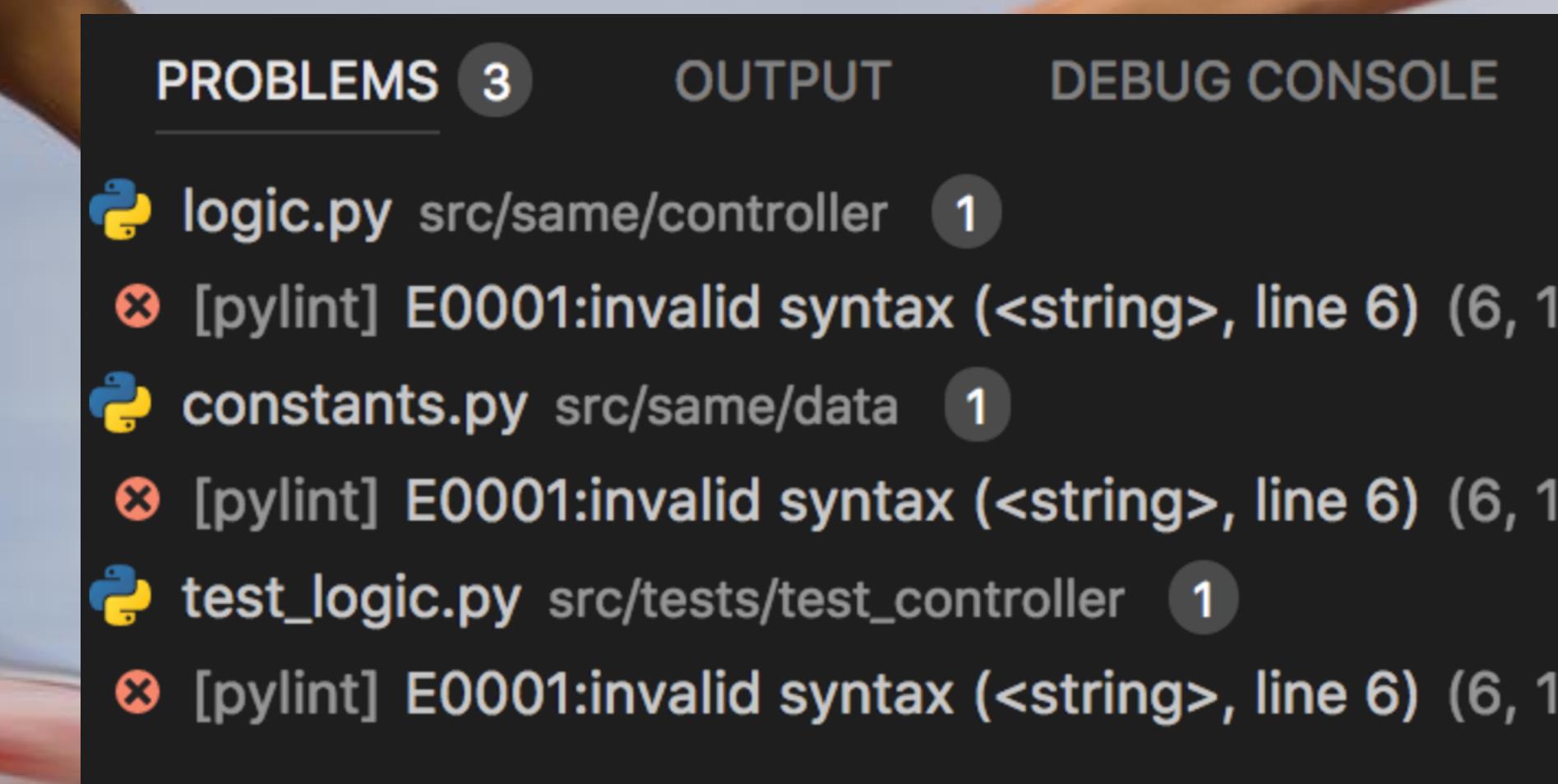
INFRASTRUCTURE



INFRASTRUCTURE AS CODE



CONTINUOUS INTEGRATION



Copyright 2009 Mike Watson Images Limited.

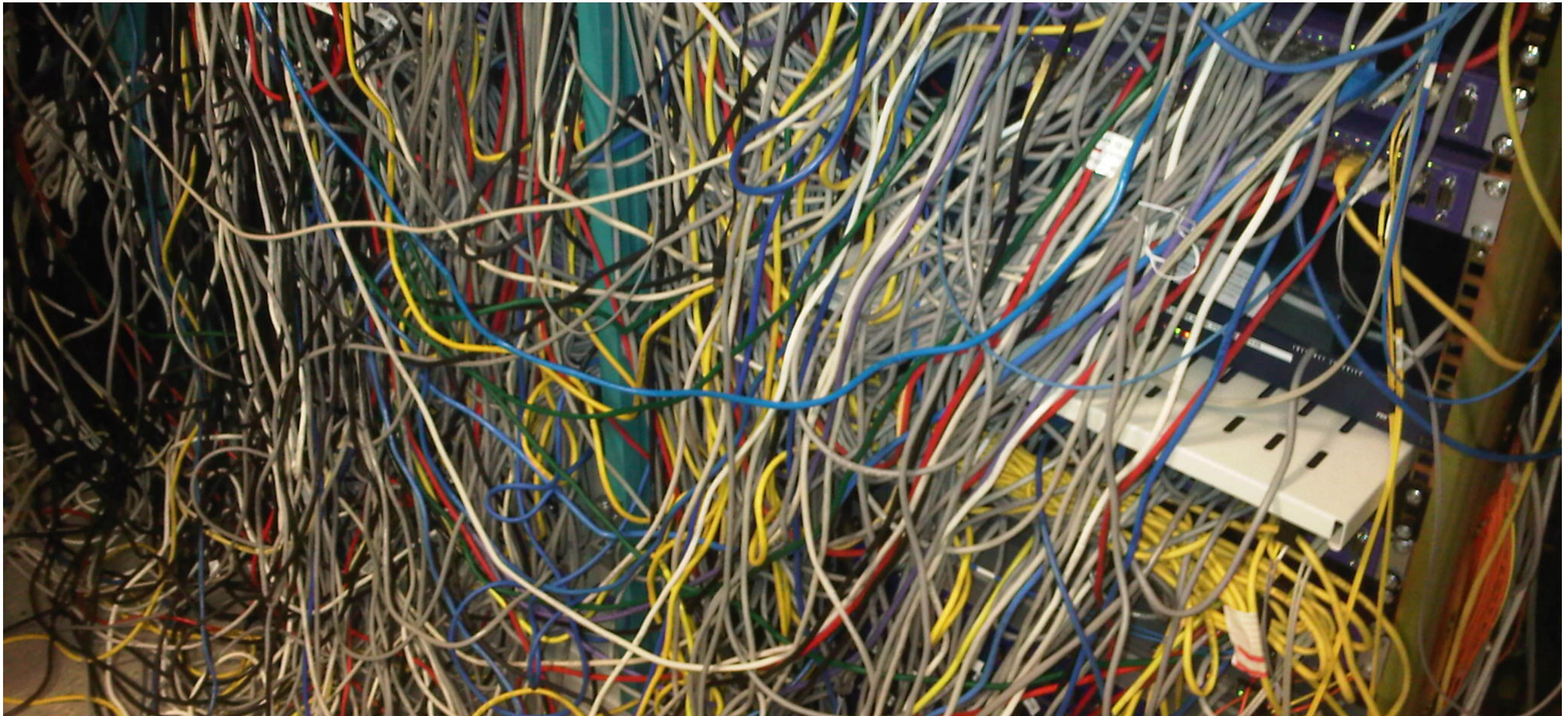
CONTINUOUS DELIVERY



SERVERLESS



Infrastructure: Challenges



THE GREAT DIVIDE

Dev

Ops

GATEKEEPING



LACK OF STANDARDIZATION



*System architecture.
Networking and security.*

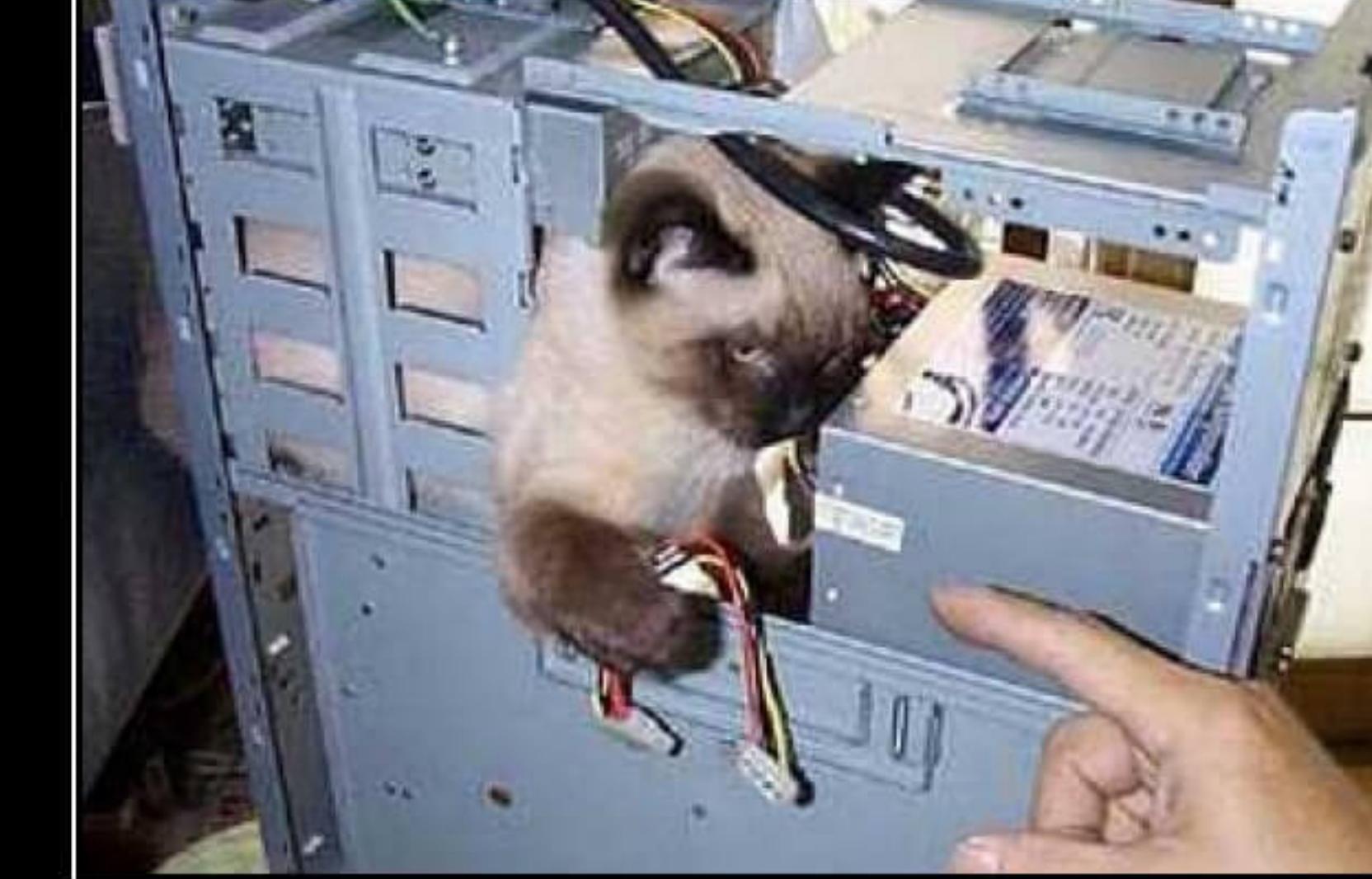
MANUAL ALL THE THINGS



INFRASTRUCTURE CONFLICTS



409
Conflict



500
Internal Server Error

SLOW TO UPDATE

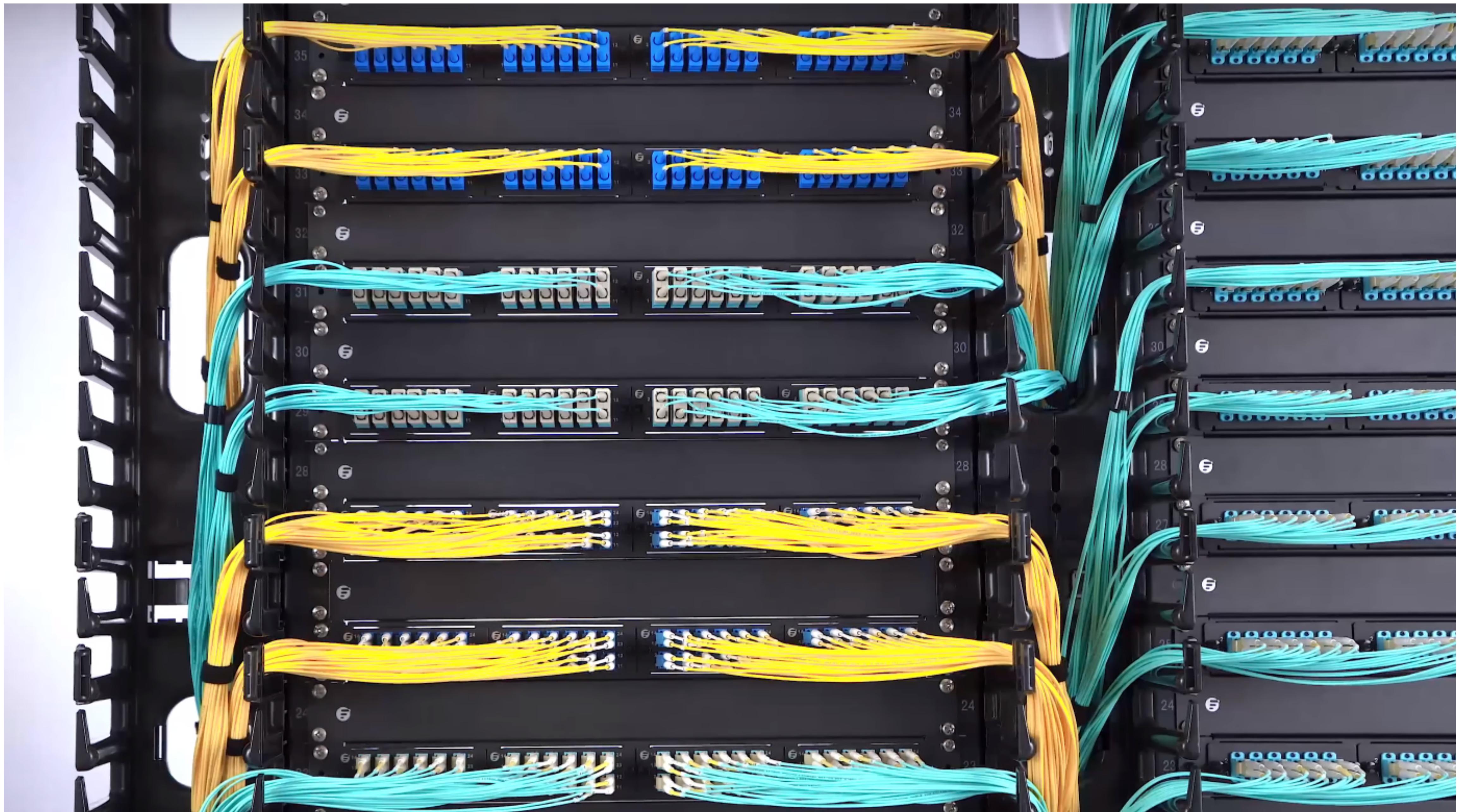


SLOW TO ROLLBACK

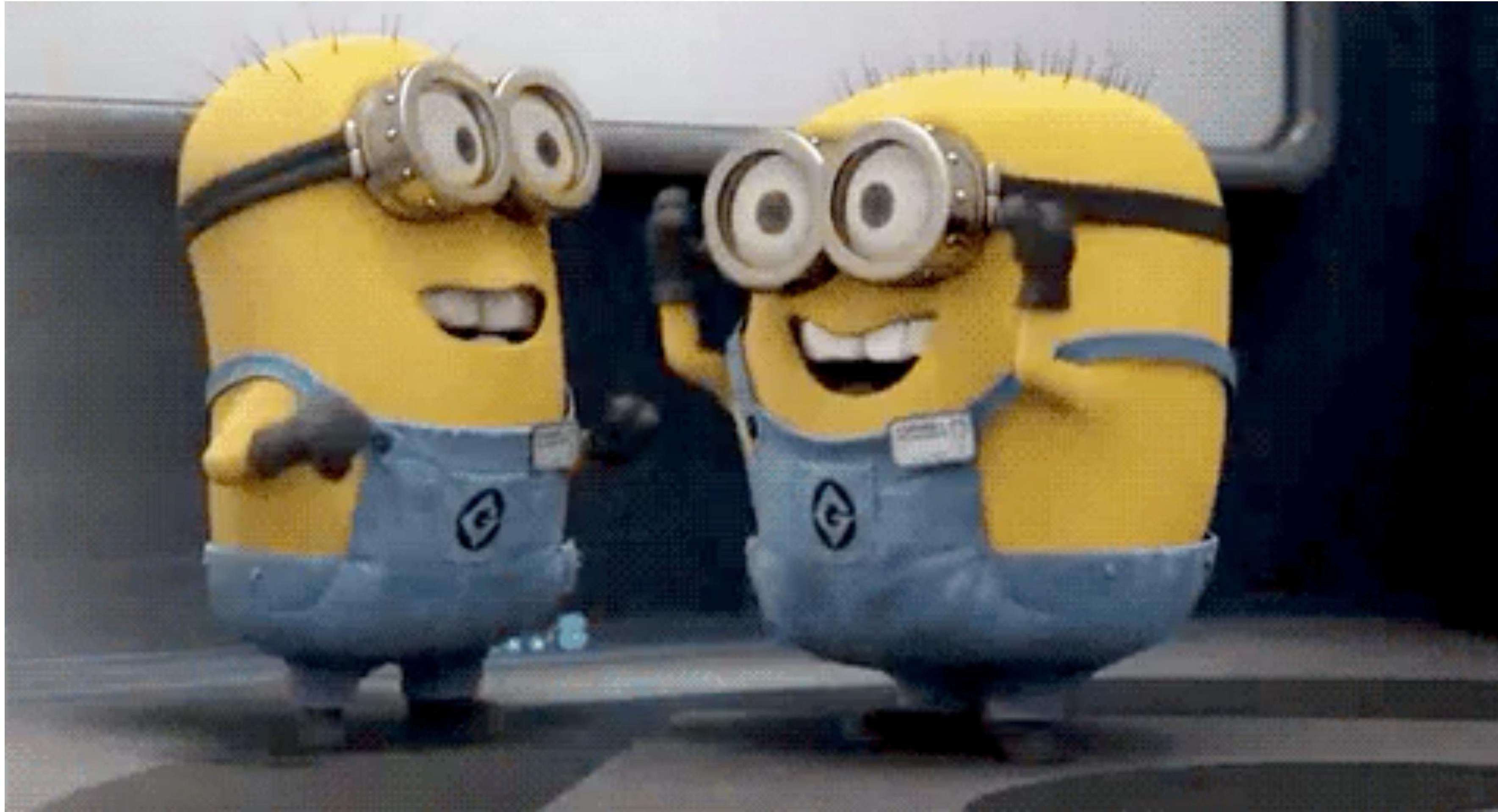
SENORGIF.COM



THEN ALONG CAME IAC



IaC + Serverless: Solutions



THE GREAT DIVIDE UNION

DevOps



KNOWLEDGE SHARING



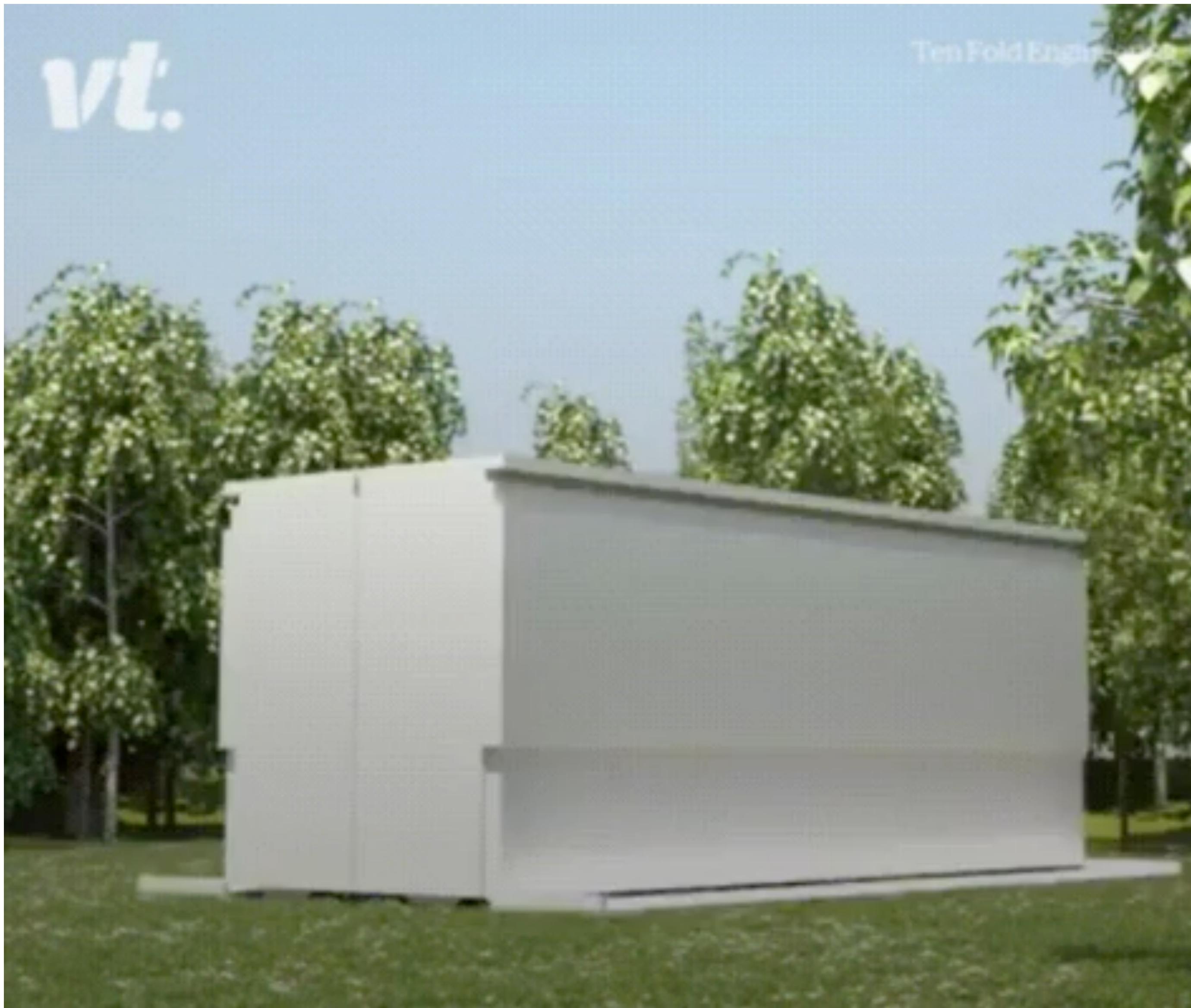
DevOps

STANDARDIZATION

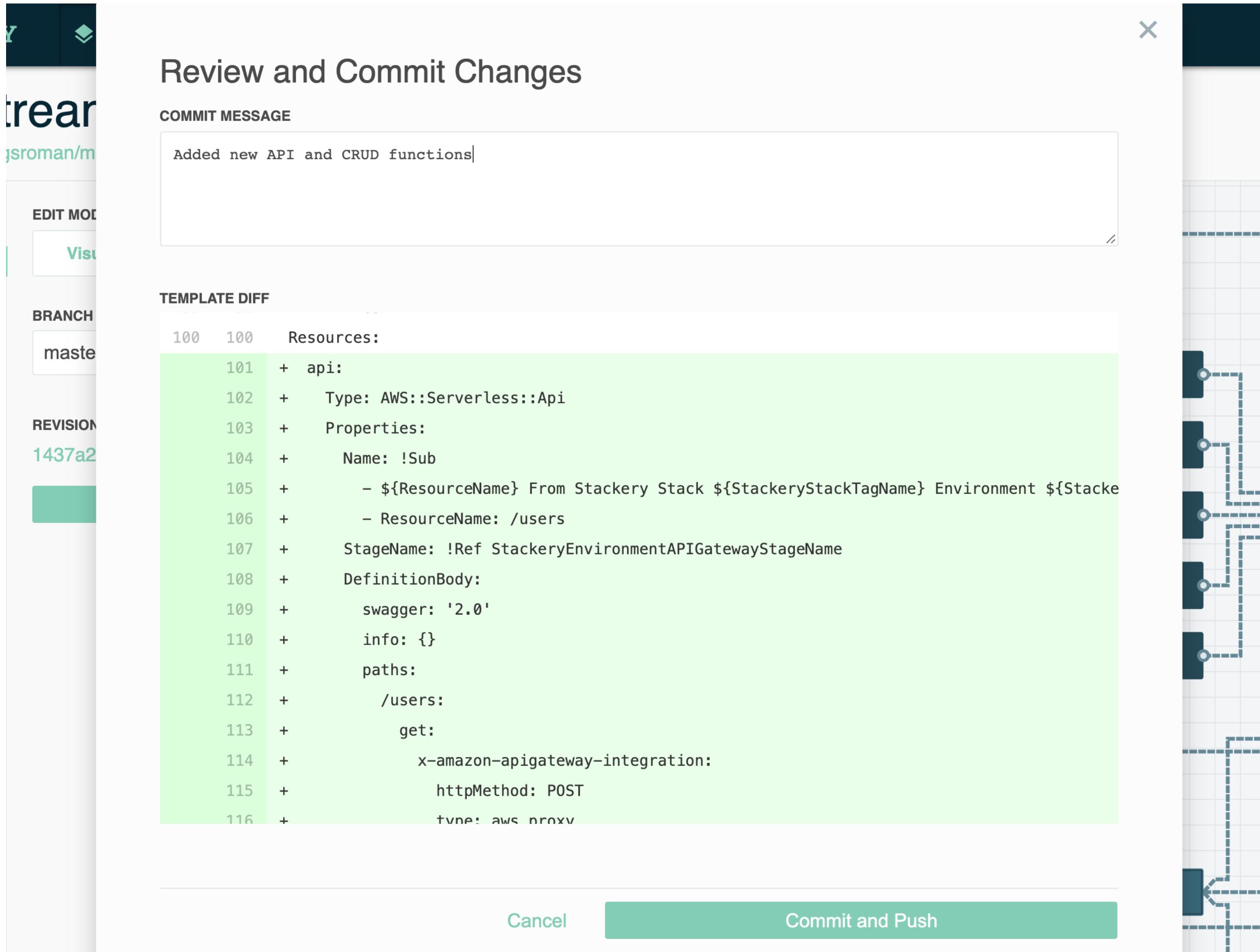


SERVERLESS IS FOR PRAGMATISTS

AUTOMATE ALL THE THINGS



NO CONFLICTS HERE



QUICK TO UPDATE



QUICK ROLLBACKS



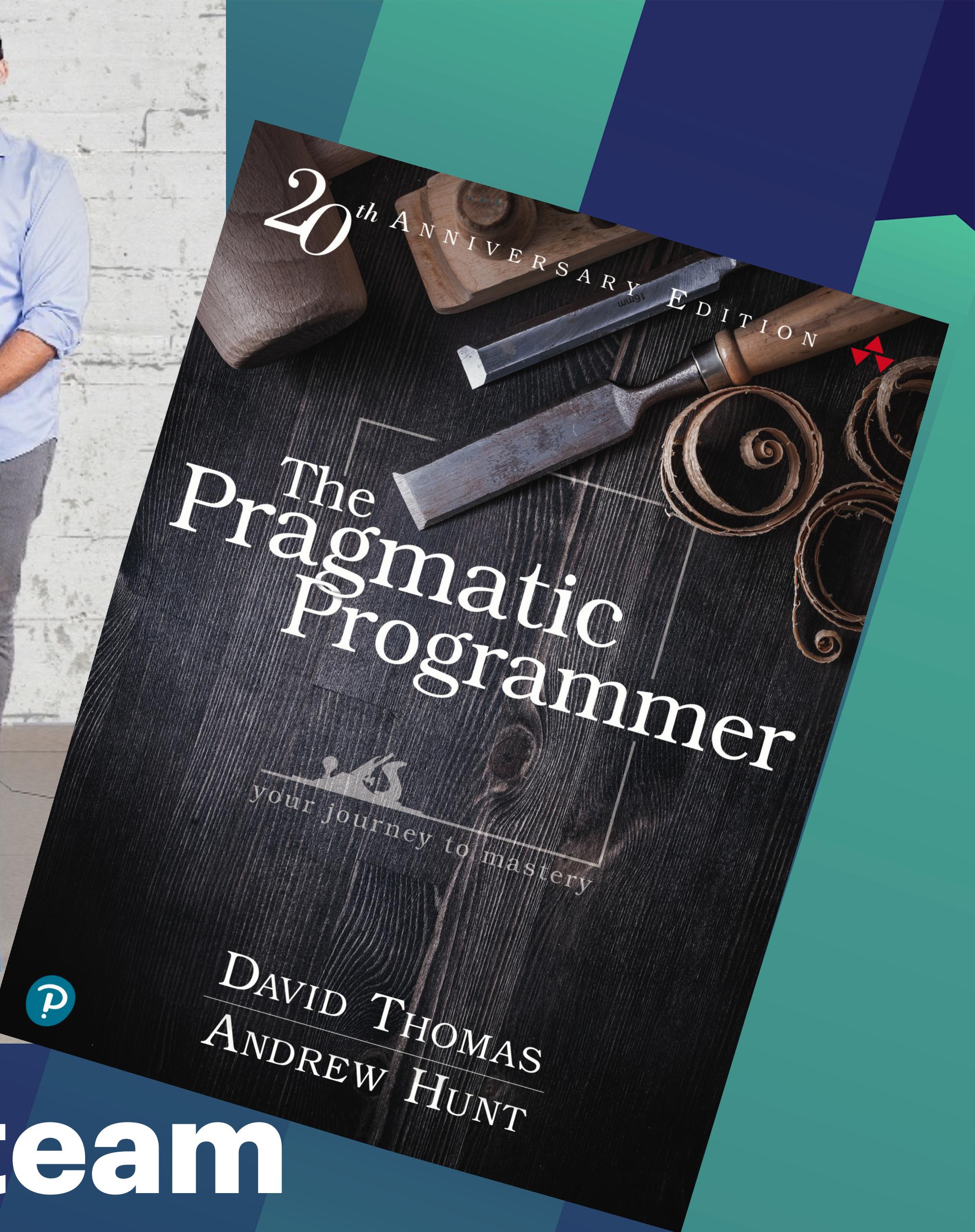
How to IaC in a serverless team





Meet the NoisyHerd team

SERVERLESS IS FOR PRAGMATISTS



WHY SERVERLESS?

- Events are bursty workloads by definition
- Managed services held together by IaC in version control
- Ability to choose managed services specialized for video streaming

IaC has challenges too

PEACE IS PAVED
WITH KINGDOM SAVED
HURRAH TO MARIO
OUR ONLY HERO
THIS ENDS YOUR TRIP
OF A LONG FRIENDSHIP



INFRASTRUCTURE AS CODE

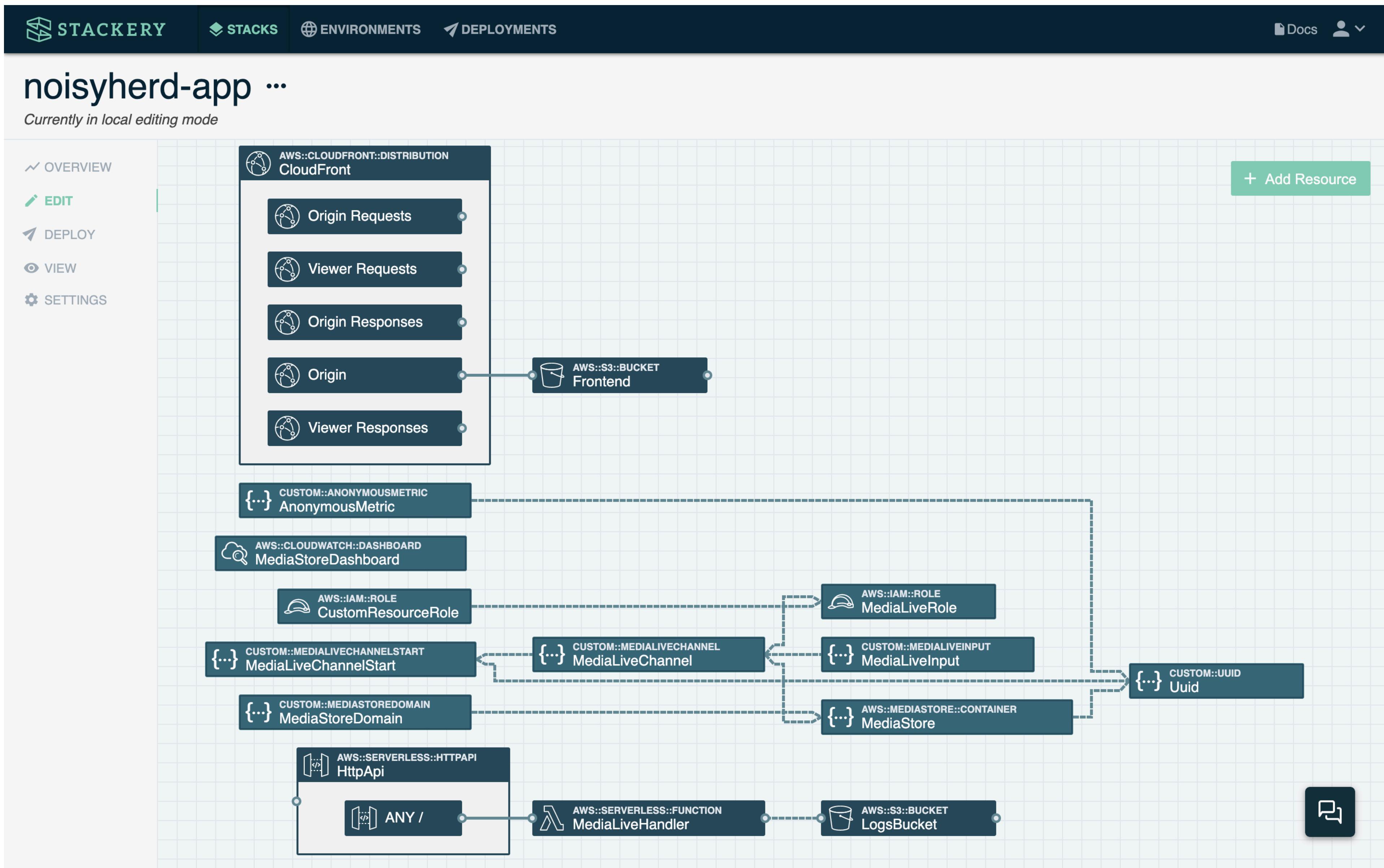


The screenshot shows a code editor interface with a vertical toolbar on the left containing icons for file operations, search, and navigation. The main area displays an AWS CloudFormation template named `template.yaml`. The template defines a stack named `noisyherd-app` with the following parameters:

- AWSTemplateFormatVersion:** 2010-09-09
- Transform:** AWS::Serverless-2016-10-31
- Description:** NoisyHerd Example
- Parameters:**
 - InputType:** Type: String, Default: URL_PULL, AllowedValues: RTP_PUSH, RTMP_PUSH, URL_PULL
 - InputCIDR:** Description: 'Specify the MediaLive input Security Group CIDR Block, this is used to restrict access to the MediaLive Input' Type: String, Default: ''
 - PullUrl:** Description: Specify the input URL for MediaLive to pull content from. Type: String, Default: https://d15an60oaeed9r.cloudfront.net/live_stream_v2/sports_reel_with_markers.m3u8
 - PullUser:** Description: If your PULL URL requitres authentication enter the username Type: String, Default: ''
 - PullPass:** Description: If your PULL URL requitres authentication enter the password Type: String, Default: '', NoEcho: true
 - EncodingProfile:** Description: 'Select an encoding profile. HD 1080p [1920x1080, 1280x720, 960x540, 768x432, 640x360, 512x288] HD 720p [1280x720, 960x540, 768x432, 640x360, 512x288] SD 540p [960x540, 640x360, 512x288]' Type: String, Default: HD-720p, AllowedValues: HD-1080p, HD-720p, SD-540p
 - ChannelStart:** Description: Start the Live Channel as part of the deployment Type: String, Default: 'Yes', AllowedValues: Yes, No
 - StackTagName:** Type: String, Description: Stack Name (injected by Stackery at deployment time)
 - EnvironmentTagName:** Type: String, Description: Environment Name (injected by Stackery at deployment time)

SERVERLESS IS FOR PRAGMATISTS

INFRASTRUCTURE AS CODE



ISOLATION OF INFRASTRUCTURE

Did I just drop the prod table?

development

staging

production

WHAT ABOUT PARAMETERS & SECRETS?

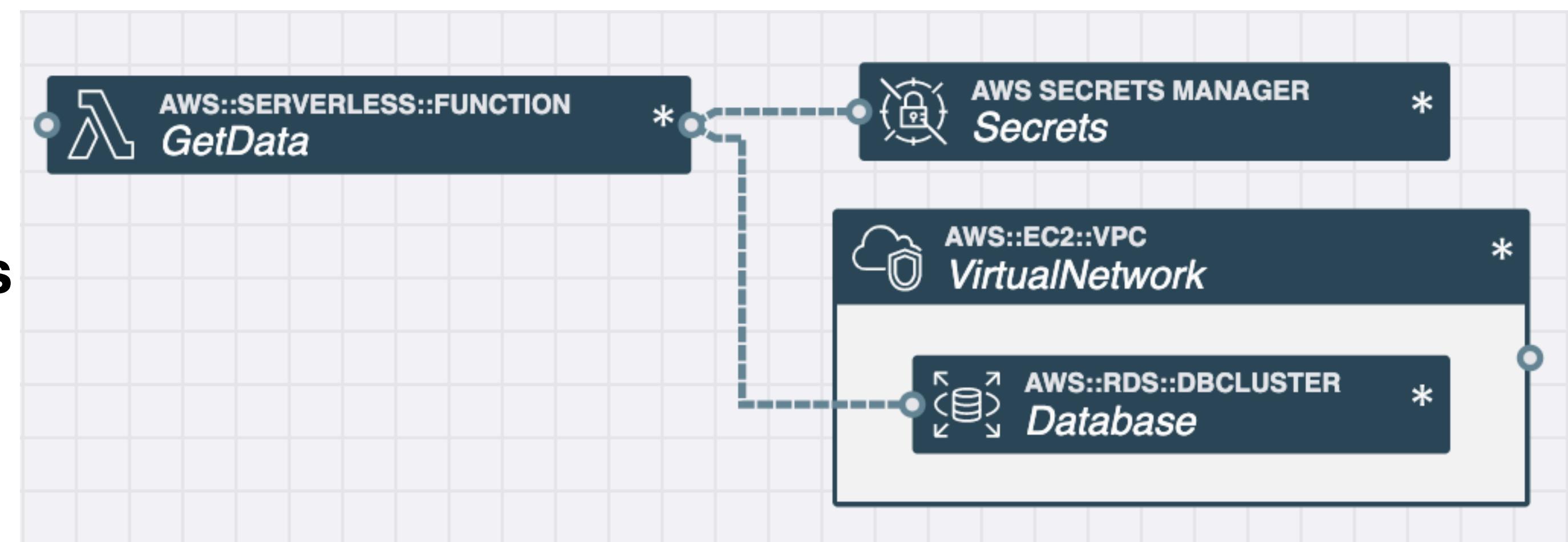
Please don't commit your database password to GitHub!

- Secrets manager
- Namespaced environment parameters

/dev/databasePassword

/staging/databasePassword

/production/databasePassword



ISOLATION OF INFRASTRUCTURE

Did I just drop the prod table?

development

API: https://1234.execute-api.us-west-2.amazonaws.com/development/*

DynamoDB Table: noisyherdapp-development-userTable

staging

API: https://1234.execute-api.us-west-2.amazonaws.com/staging/*

DynamoDB Table: noisyherdapp-staging-userTable

production

API: https://noisyherd.com/*

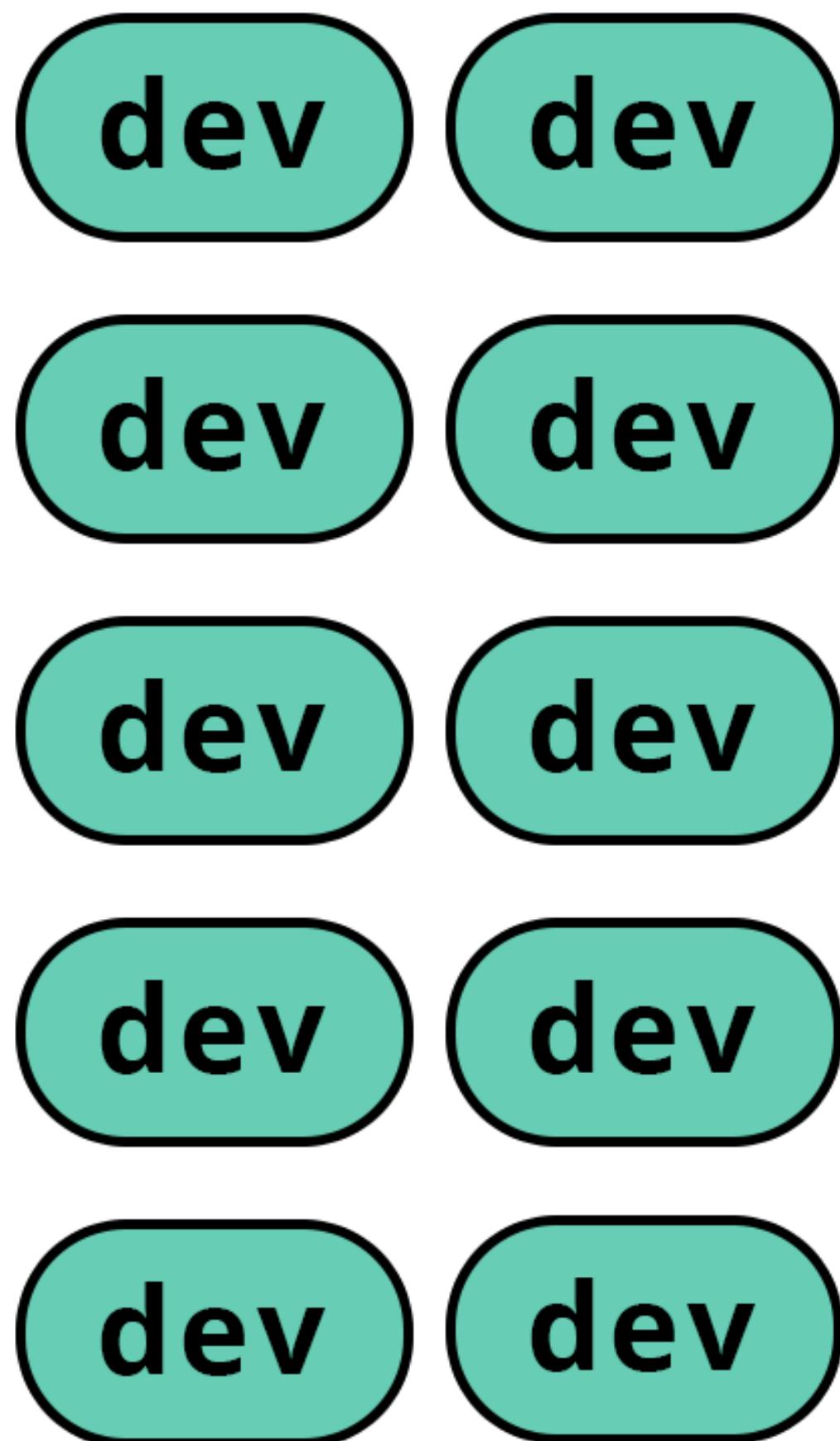
DynamoDB Table: noisyherdapp-production-userTable

ISOLATION OF INFRASTRUCTURE

```
import os
env = os.environ.get("PYTHON_ENV")
if(env == "production"):
    url="noisyherd.com"
else :
    url=f"1234.execute-api.us-
west-2.amazonaws.com/
{PYTHON_ENV}/"
```

LOCAL DEVELOPMENT

How do you run AWS on localhost? (hint: don't)



staging

production

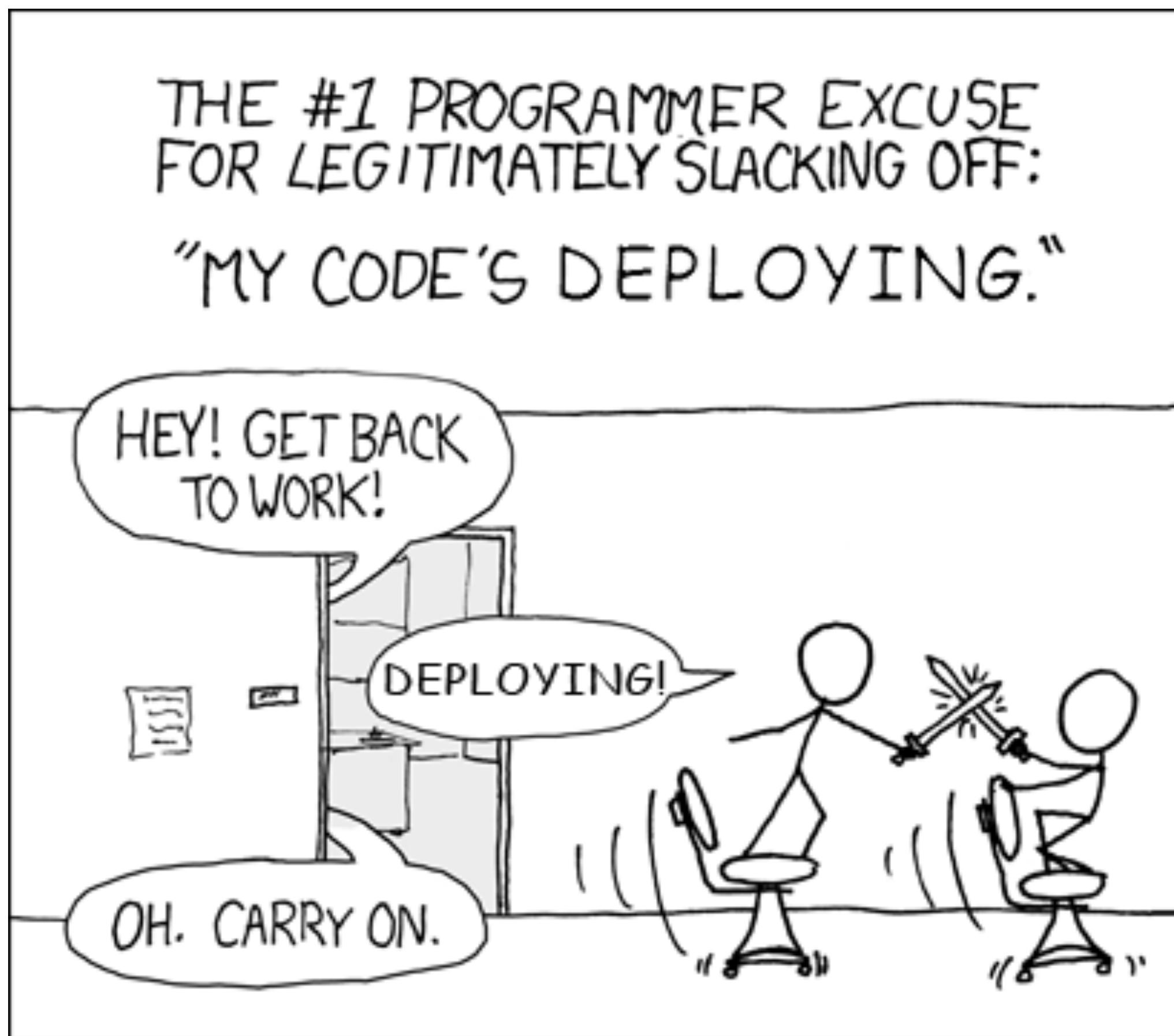
LOCAL DEVELOPMENT

Mocking resources



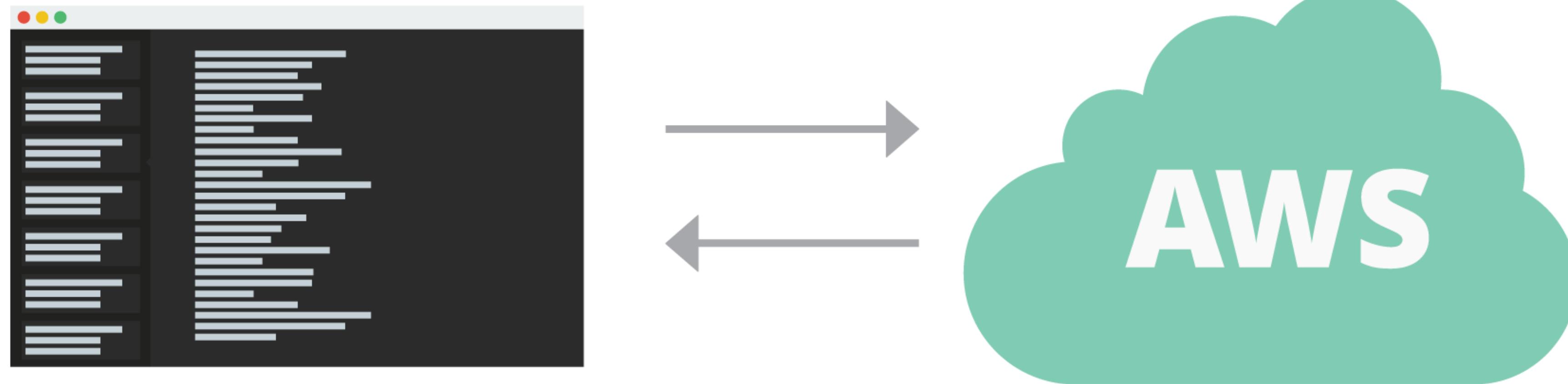
LOCAL DEVELOPMENT

Deploying every change



CLOUD + LOCAL = CLOUDLOCAL

The best of both worlds

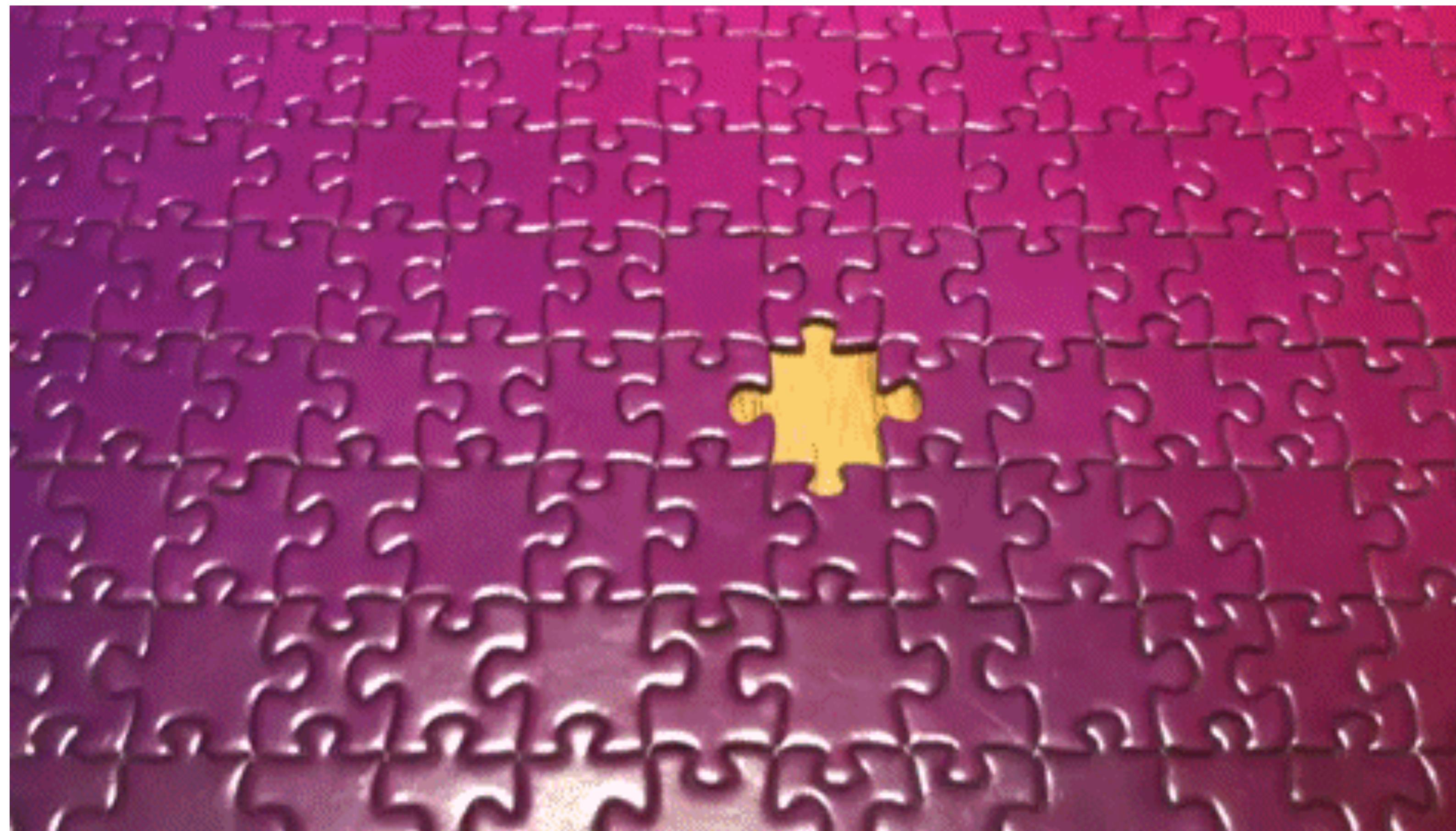


LOCAL DEVELOPMENT IRL

- Checkout code from version control
- Add new infrastructure to the IaC template
- Deploy to isolated dev environment
- Make code changes locally, against live resources
- PR both code and infrastructure

TESTING

Unit tests are fine for your code, but what about testing the whole system?



HOW DOES THIS RELATE TO CI/CD?

Continuous integration + continuous delivery = continuous deployments



HOW DOES THIS RELATE TO CI/CD?

Continuous integration + continuous delivery = continuous deployments



WHAT ABOUT SECURITY?

Everyone is now DevSecOps

- Built into the process through:
 - Environment isolation
 - Change management - code review process
 - Tooling to ensure each function is scanned for vulnerabilities
 - Resources have permissions too: least-privileged permissions defined in IaC
 - Permissions: who can deploy to prod

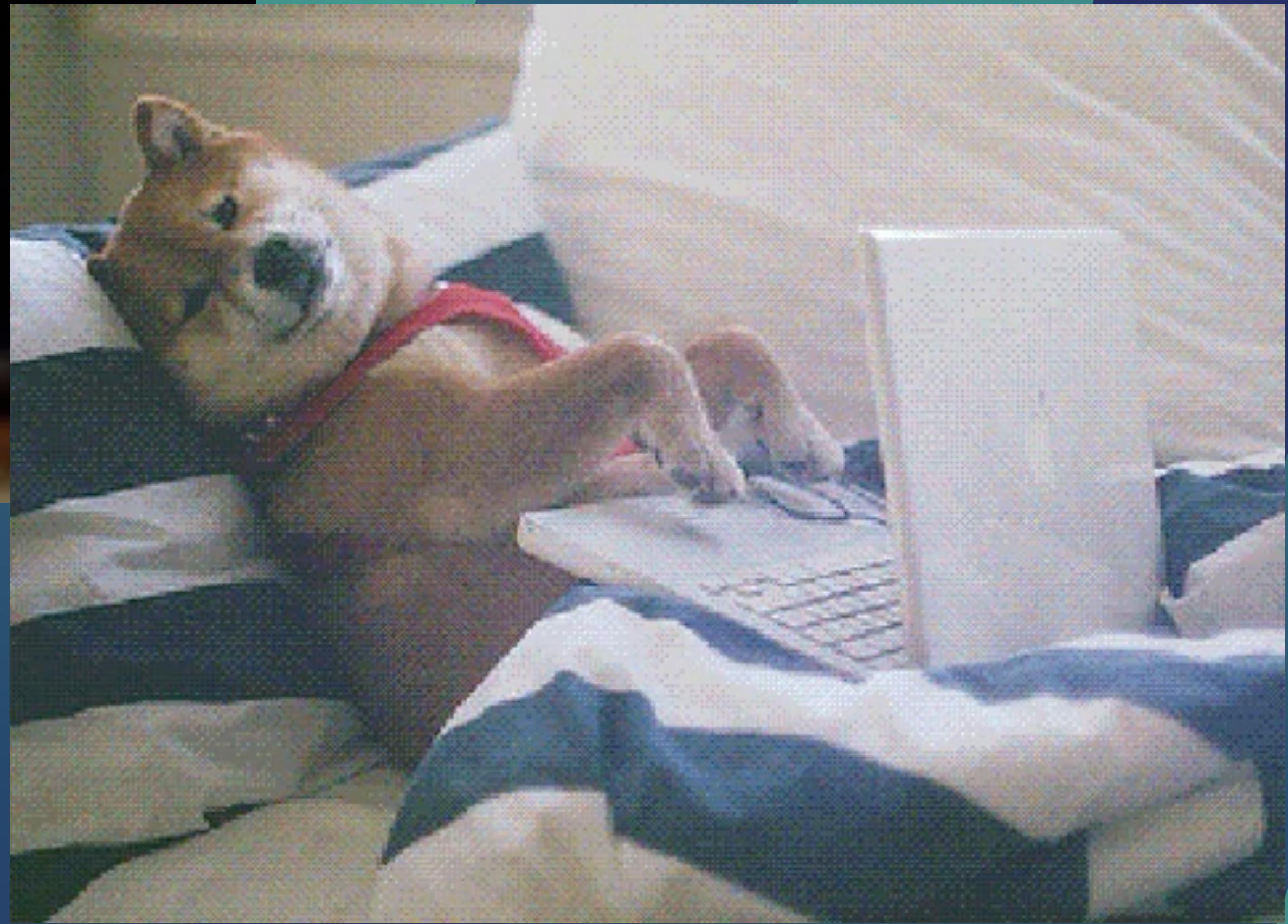
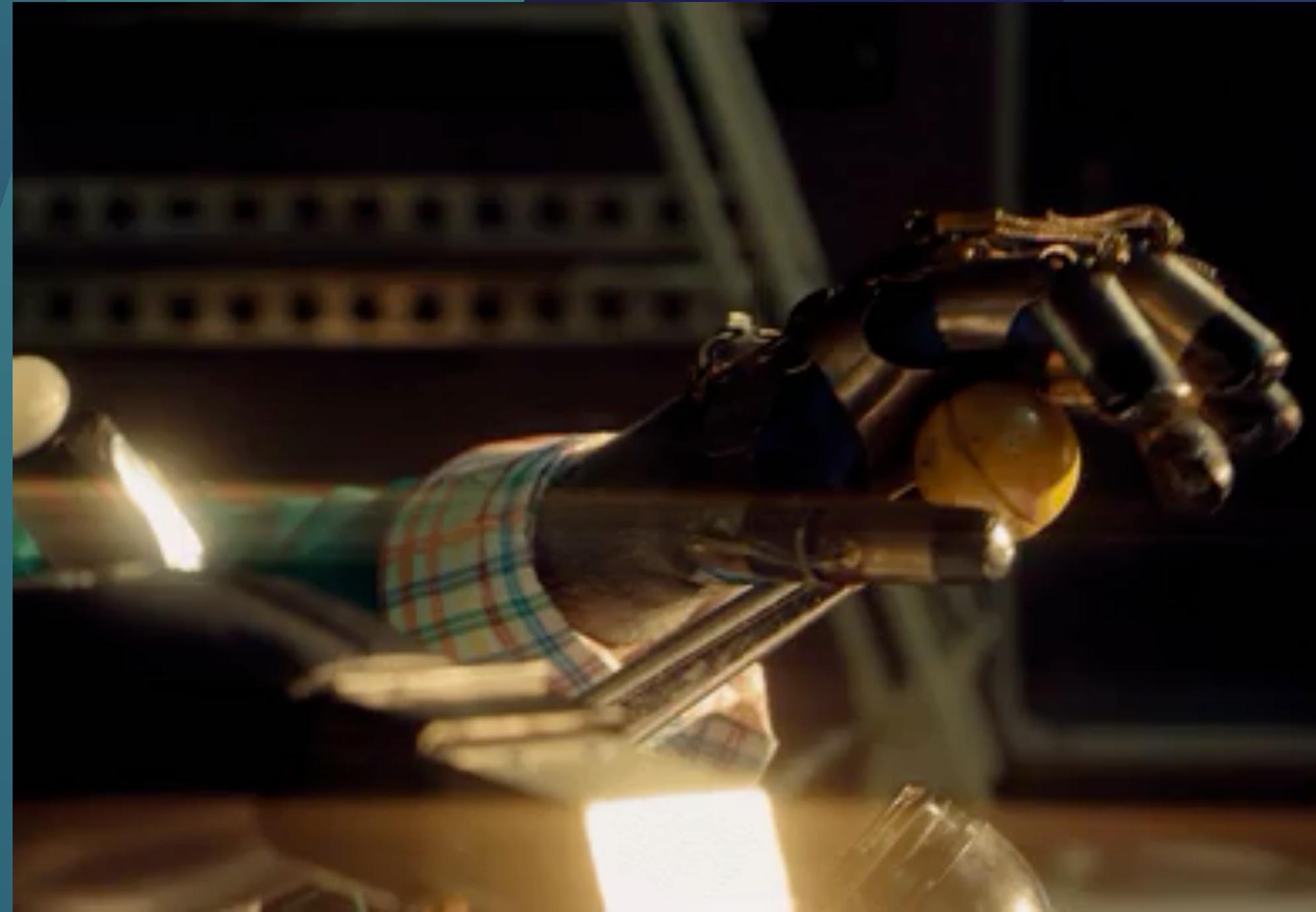
CI/CD for IaC: Benefits for Engineering Teams



Robust local development against live resources



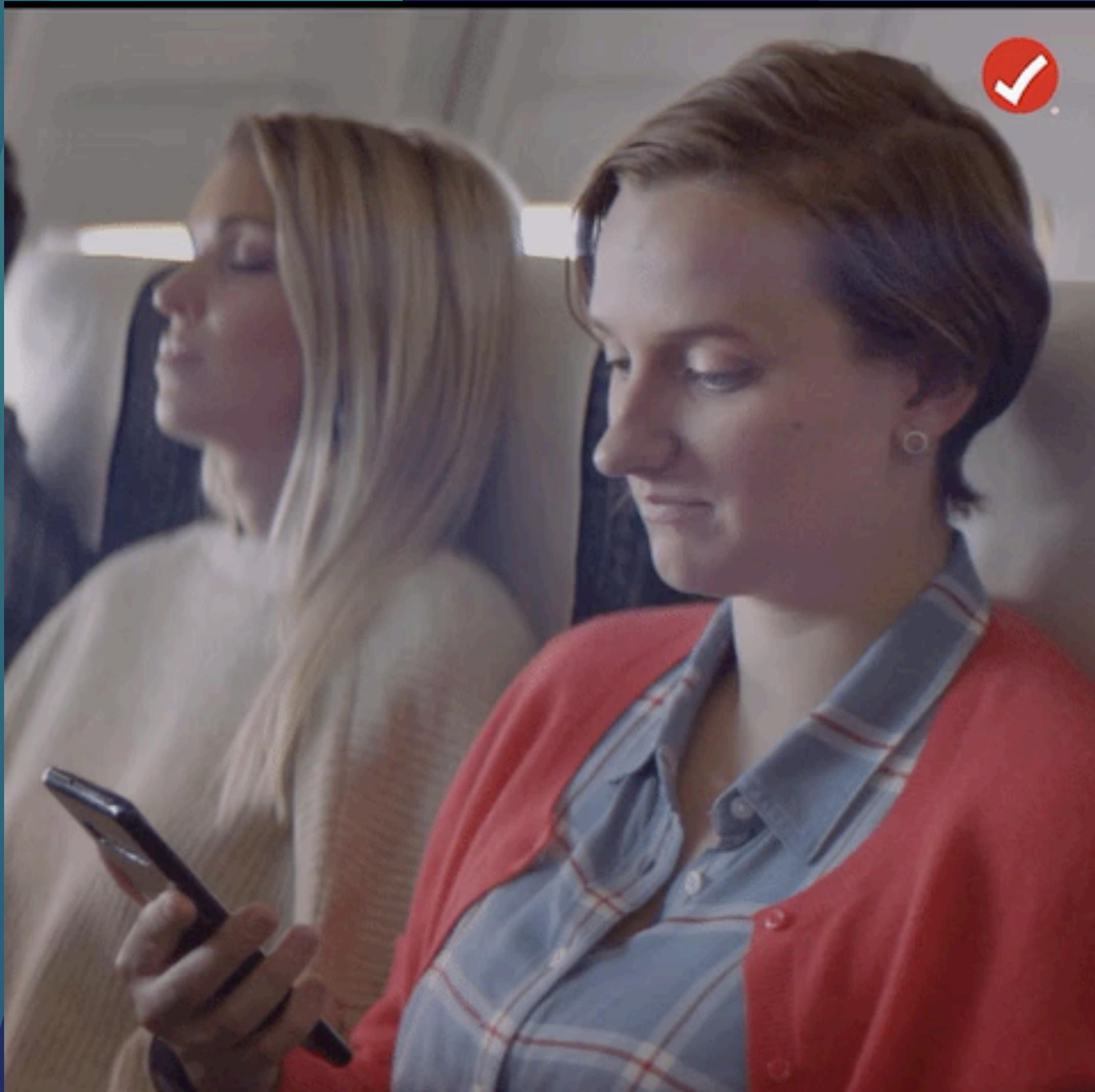
Devs can experiment safely



Automated testing & deployment



A fast-moving team



A safe and secure app for users

THANK YOU!



STACKERY
www.stackery.io



@stackeryio
@txase, @annaspies

CHASE DOUGLAS & ANNA SPYSZ, STACKERY.IO
PYTHON WEB CONFERENCE 2020