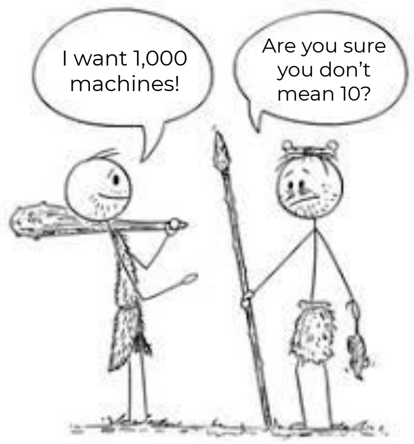
# Validating Configs from Code to Deployment

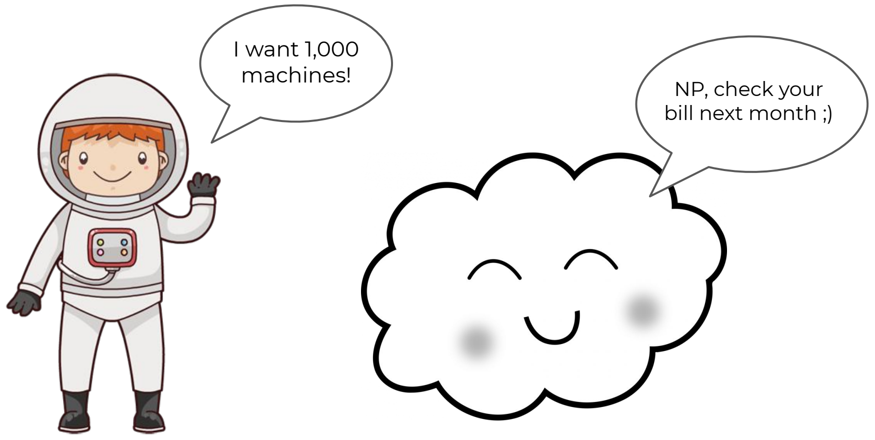
in the GitOps World

## What's the problem?

- **Continuous delivery is the goal and GitOps is the mean**
- **But** being a GitOps organization can be very dangerous
- **Because** every config change by a developer can cause production issues



B.C



A.D

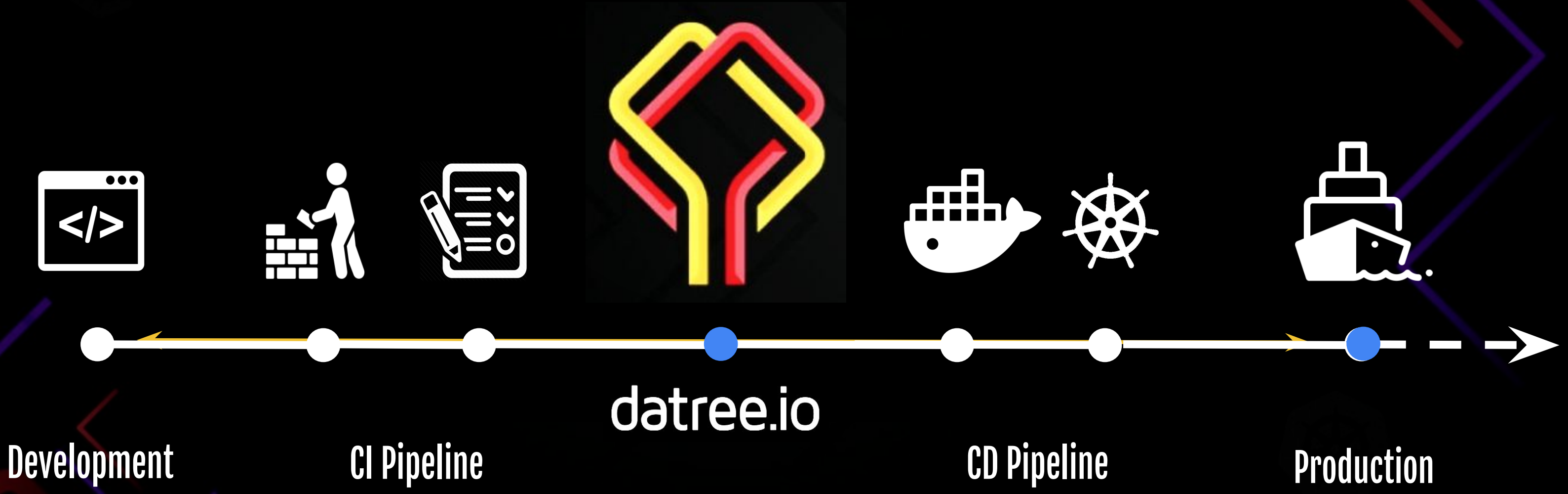
## So...

How to delegate IAC responsibilities to developers and still sleep well at night?

datree.io

## $ whoami

```yaml
apiVersion: 32
kind: CPO
metadata:
     name: “Eyar Zilberman”
     labels:
          company: datree
          role: co-founder
more:
  - Organizer of the biggest github community
  - Hate SQL
  - Love RegEx
```


datree.io
Development
CI Pipeline
CD Pipeline
Production

## @ Datree

# Policies are how we roll

🍣

## A practical guide

How to find the yummiest carrots

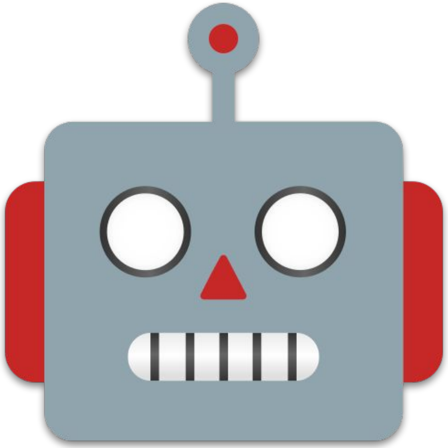
datree.io

## A practical guide

How to help your developers troubleshoot and avoid IAC misconfigurations

datree.io

## Framework - V.V.F.F



1. Validation steps:
   A. Valid file
   B. Valid schema
   C. Following best practices
   D. Following team/org policies
2. Automate & shift-left

## 1. A) valid file

- The file syntax is correct (YAML, JSON, XML, etc.)

**[X]** K8s.yaml

```yaml
--
apiVersion: apps/v1
kind: Deployment
metaData:
  name: rss-site
  namespace: test
  labels:
    app: web
```

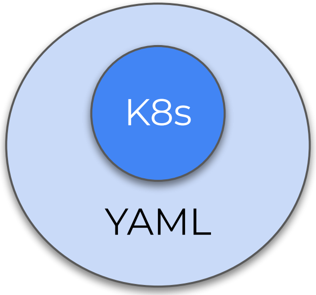
**[V]** K8s.yaml

```yaml
---
apiVersion: apps/v1
kind: Deployment
metaData:
  name: rss-site
  namespace: test
  labels:
    app: web
```

- yq - a portable command-line YAML processor
- Remember: YAML can be convert to JSON and vice versa

## 1. B) valid schema

- The technology syntax is correct (K8s, Ansible, Terraform, etc.)



[X] K8s.yaml

```
ApiVersion: apps/v1
Kind: Deployment
```

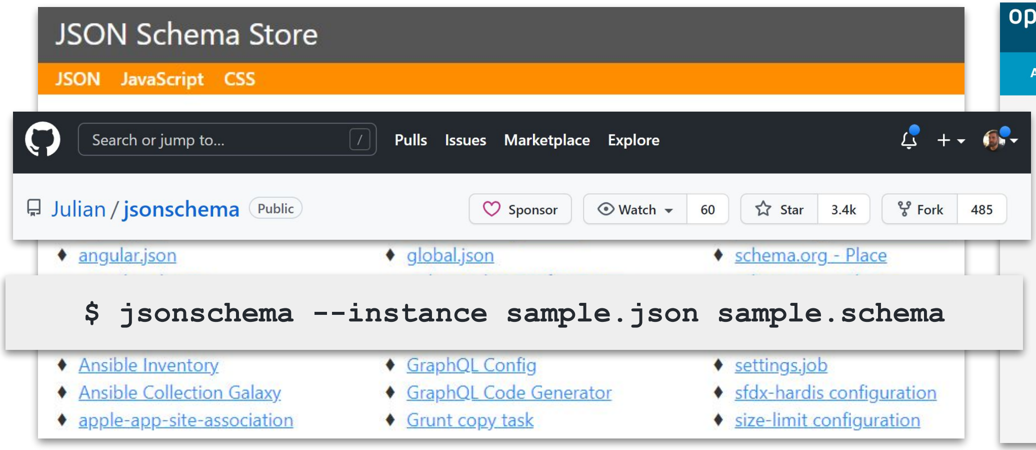
[V] K8s.yaml

```
apiVersion: apps/v1
kind: Deployment
```

- “Under the hood” schema validation is done with JSON Schema

## 1. B) valid schema - Tools & tips

- Use the built-in option (validate / check / dry-run)
- K8s has a bug so it's not working offline - use kubeconform instead
- JSON Schema Store + jsonschema (py)



```
$ jsonschema --instance sample.json sample.schema
```

Why you need to use Kubernetes schema validation tools

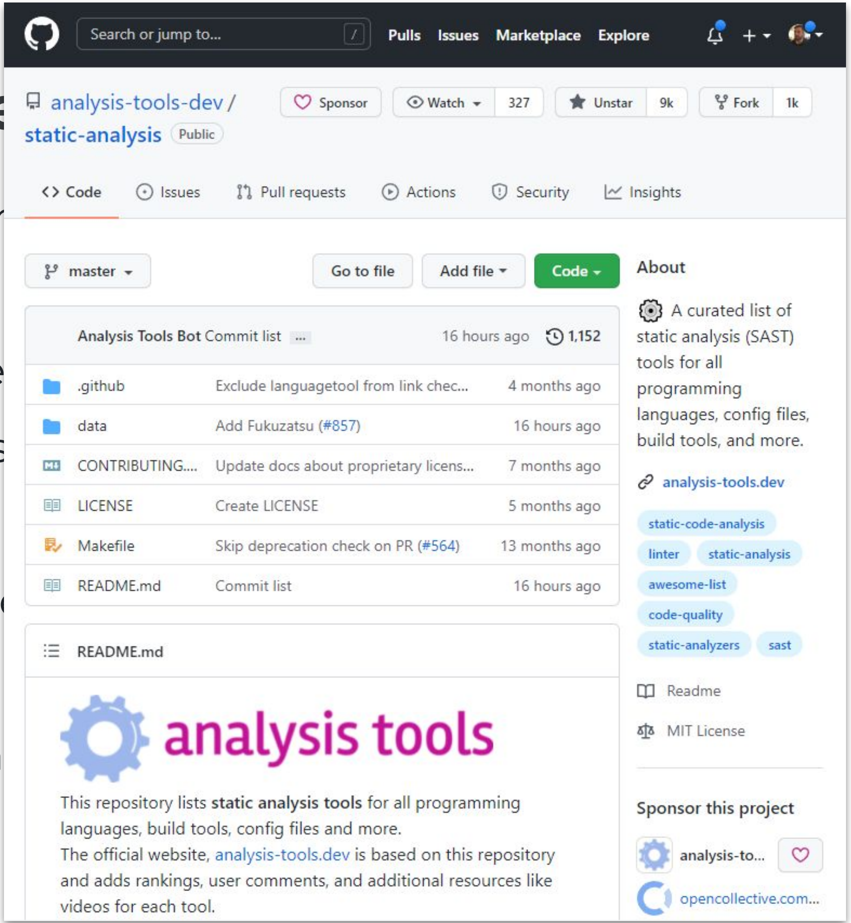
Compare schema validation tools that help you avoid misconfigurations in your Kubernetes clusters.

06 Jul 2021 | Eyar Zilberman | 43

datree.io

## 1. C) Following best practice

- "A **best practice** is a method or tech
  accepted as superior"
- Aka - lessons learned from other pe
- Helping gain better security, more s
- <u>Examples:</u>
  - Security - Each container imag
  - Stability - Each container has a
  - Cost - Each workload has a con



analysis-tools-dev / static-analysis Public

| | | |
|---|---|---|
| Analysis Tools Bot Commit list ... | 16 hours ago | 1,152 |
| .github | Exclude languagetool from link chec... | 4 months ago |
| data | Add Fukuzatsu (#857) | 16 hours ago |
| CONTRIBUTING.... | Update docs about proprietary licens... | 7 months ago |
| LICENSE | Create LICENSE | 5 months ago |
| Makefile | Skip deprecation check on PR (#564) | 13 months ago |
| README.md | Commit list | 16 hours ago |

About

A curated list of static analysis (SAST) tools for all programming languages, config files, build tools, and more.

analysis-tools.dev

static-code-analysis linter static-analysis awesome-list code-quality static-analyzers sast

Readme

MIT License

Sponsor this project

analysis-to...

opencollective.com...

README.md

analysis tools

This repository lists **static analysis tools** for all programming languages, build tools, config files and more.
The official website, analysis-tools.dev is based on this repository and adds rankings, user comments, and additional resources like videos for each tool.

datree.io

## 1. D) Following team/org policies

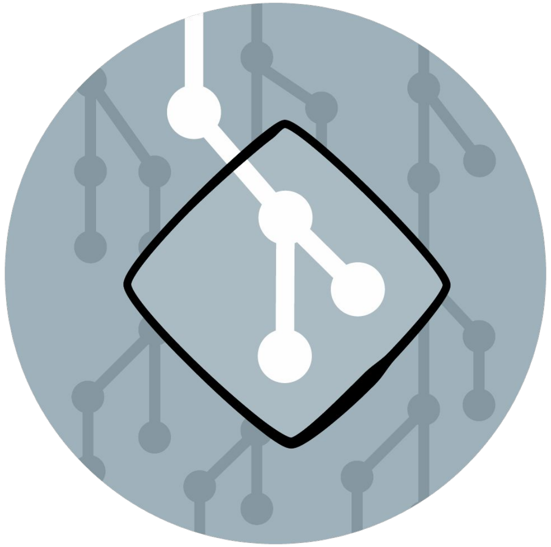
- "principle of action adopted or proposed by a business, or party"
- Aka - lessons learned from our own post-mortems
- Examples:
  - Pull all images from private registry (`artifactory.io/nginx:1.16.8`)
  - AI applications should have at least 4GB CPU
  - Only use pre-approved ports / namespace / labels / etc.

datree.io

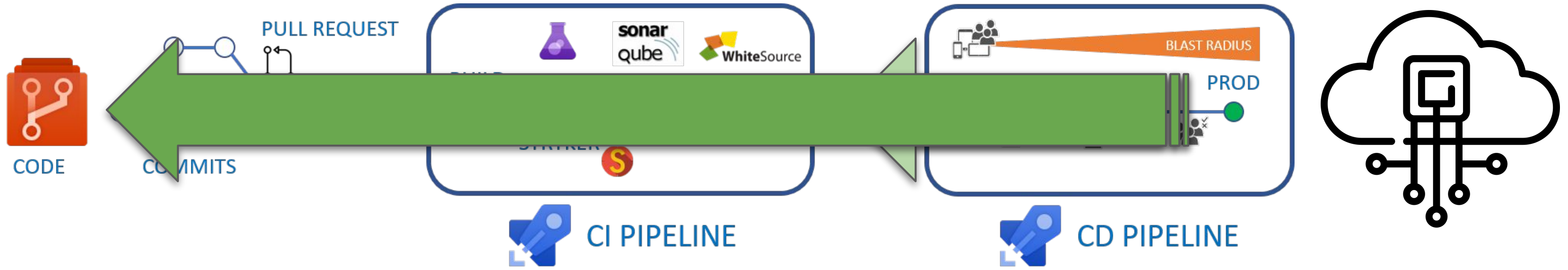
## 1. D) Following team/org policies - tools & tips

- Easy way to write and maintain policies and rules (declarative vs language)
- Threat your policies like your infrastructure and code - policy as code
  - Version control
  - Automation
  - Collaboration
- How to manage multiple GitOps projects?

  **D.R.Y with a centralized solution!**
  - Consul (HashiCorp)
  - etcd (CNCF)
  - Modularize pipeline

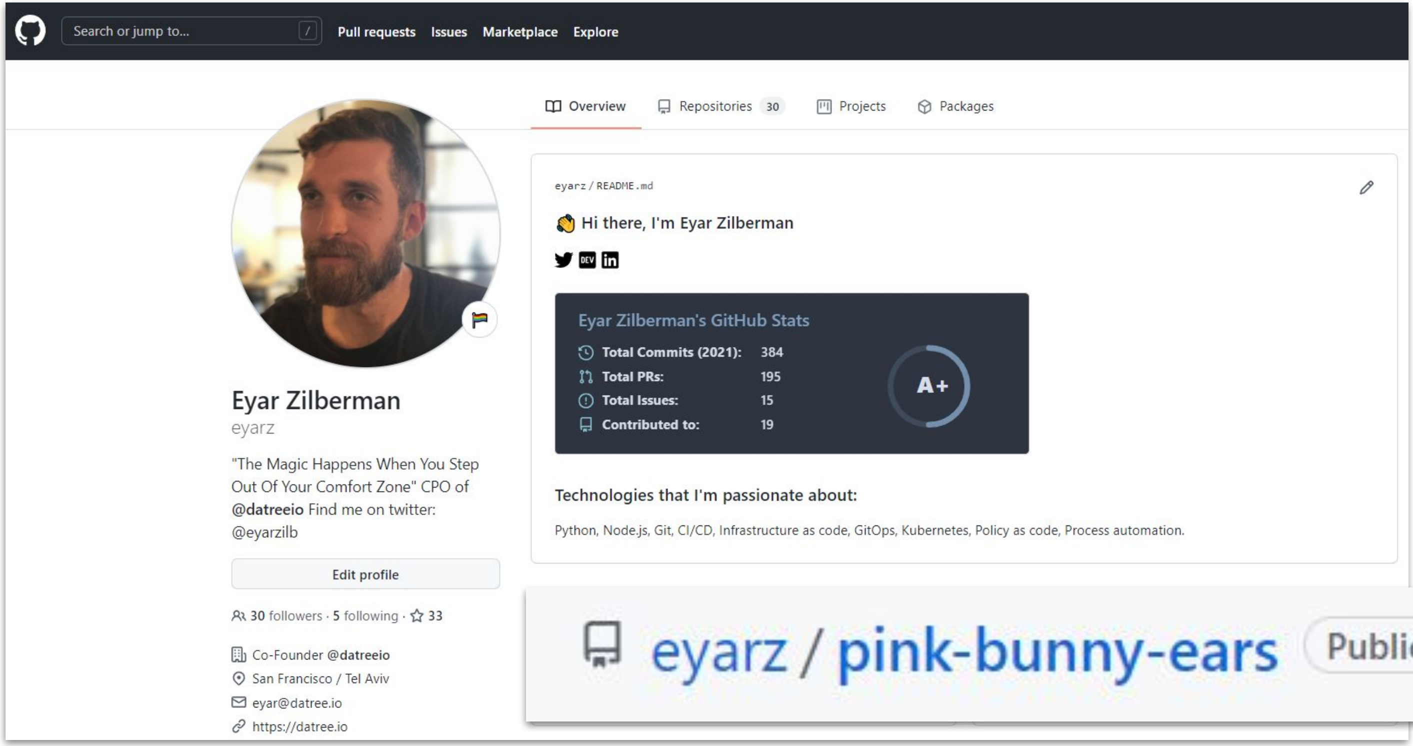


## 2. Automate and Shift-left



- “Shift-left” as much as possible (CD -> CI -> Git)
- Build a pipeline for PaC (policy as code)

## Thank you



eyarz / README.md

👋 Hi there, I'm Eyar Zilberman

Eyar Zilberman's GitHub Stats

| | |
|---|---|
| Total Commits (2021): | 384 |
| Total PRs: | 195 |
| Total Issues: | 15 |
| Contributed to: | 19 |

A+

Technologies that I'm passionate about:

Python, Node.js, Git, CI/CD, Infrastructure as code, GitOps, Kubernetes, Policy as code, Process automation.

Eyar Zilberman
eyarz

"The Magic Happens When You Step Out Of Your Comfort Zone" CPO of **@datreeio** Find me on twitter: @eyarzilb

30 followers · 5 following · 33

Co-Founder @datreeio
San Francisco / Tel Aviv
eyar@datree.io
https://datree.io

eyarz / pink-bunny-ears Public

datree.io