First Ever CI/CD Debugger

Codefresh

DAN GARFIELD - KOSTIS KAPELONIS

Dan Garfield

Chief Technology Evangelist

codefresh

Google Developer Expert CLOUD

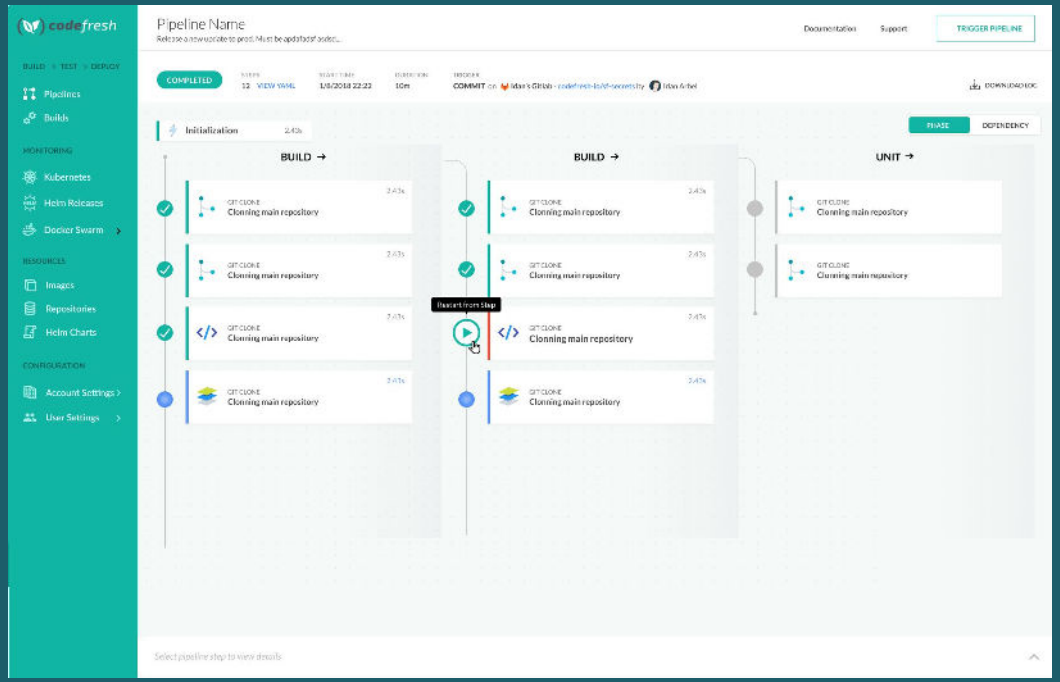
2019 OFFICIAL MEMBER Forbes Technology Council

Agenda

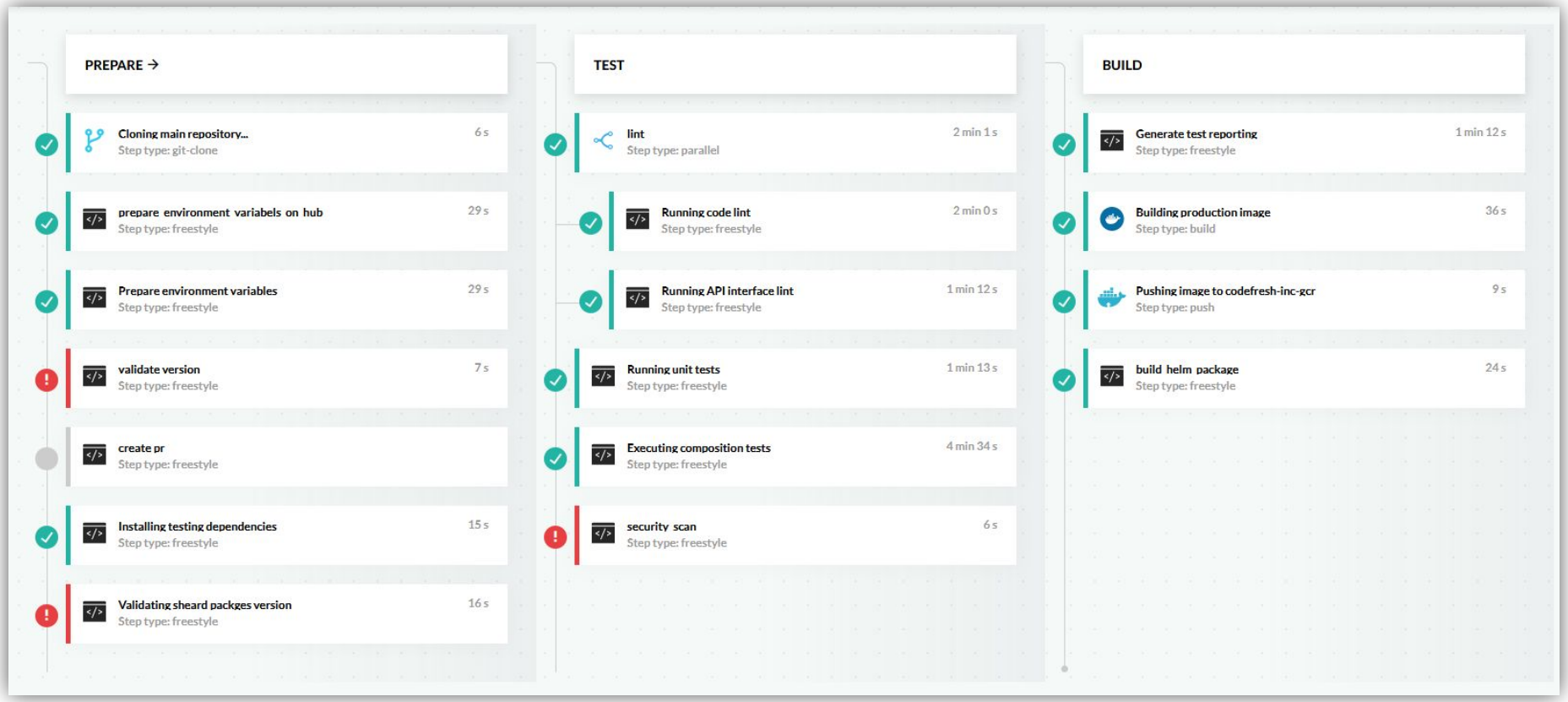
1. How debugging works in pipelines
2. Moving in/out of containers
3. Dynamic tool installation
4. Debugging and security
5. Debugging demos

About Codefresh

- Docker based CI/CD solution
- Each build step is a Docker image
- Native support for Docker, Helm, Kubernetes deployments
- Includes built-in Docker registry and Helm repository
- 20,000+ users



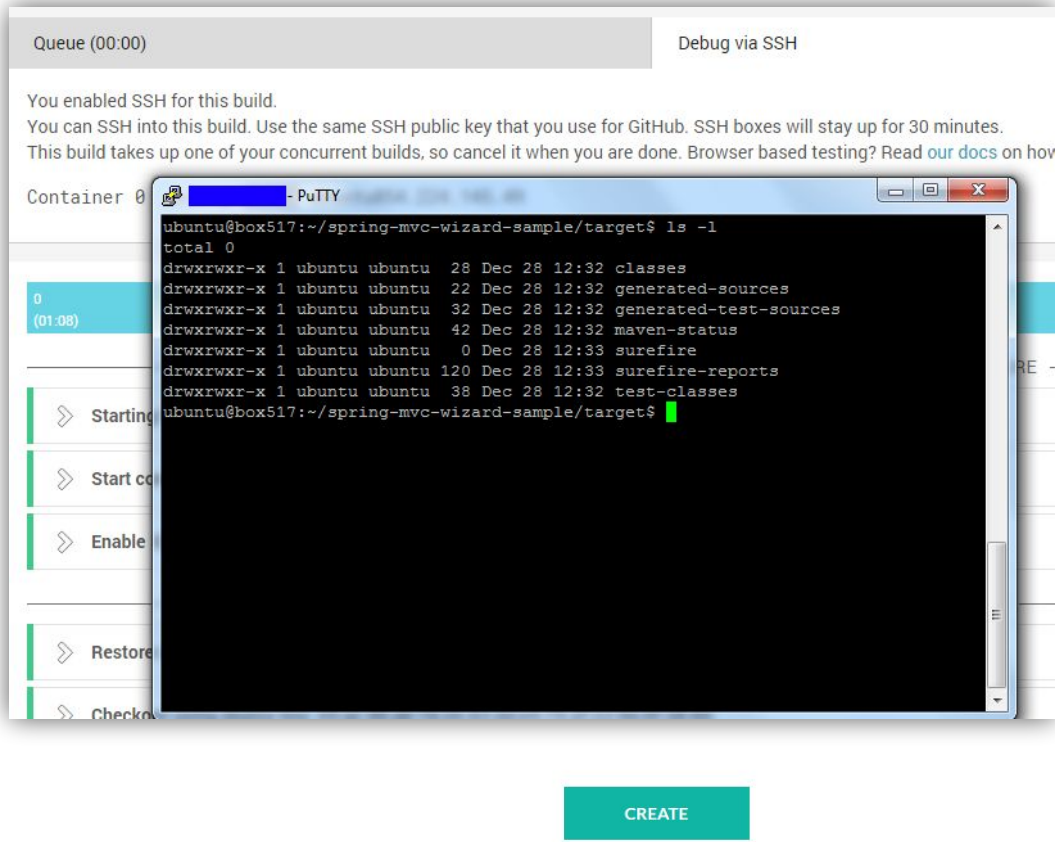
Pipelines break. It happens!



Other CI solutions

SSH is not very helpful

- Only works for legacy VM based pipelines
- Can only verify tools installed
- Lack of running context
- There is no state (e.g. environment variables)



DevOps = bringing developers and operators together

Debugging Code

The familiar experience

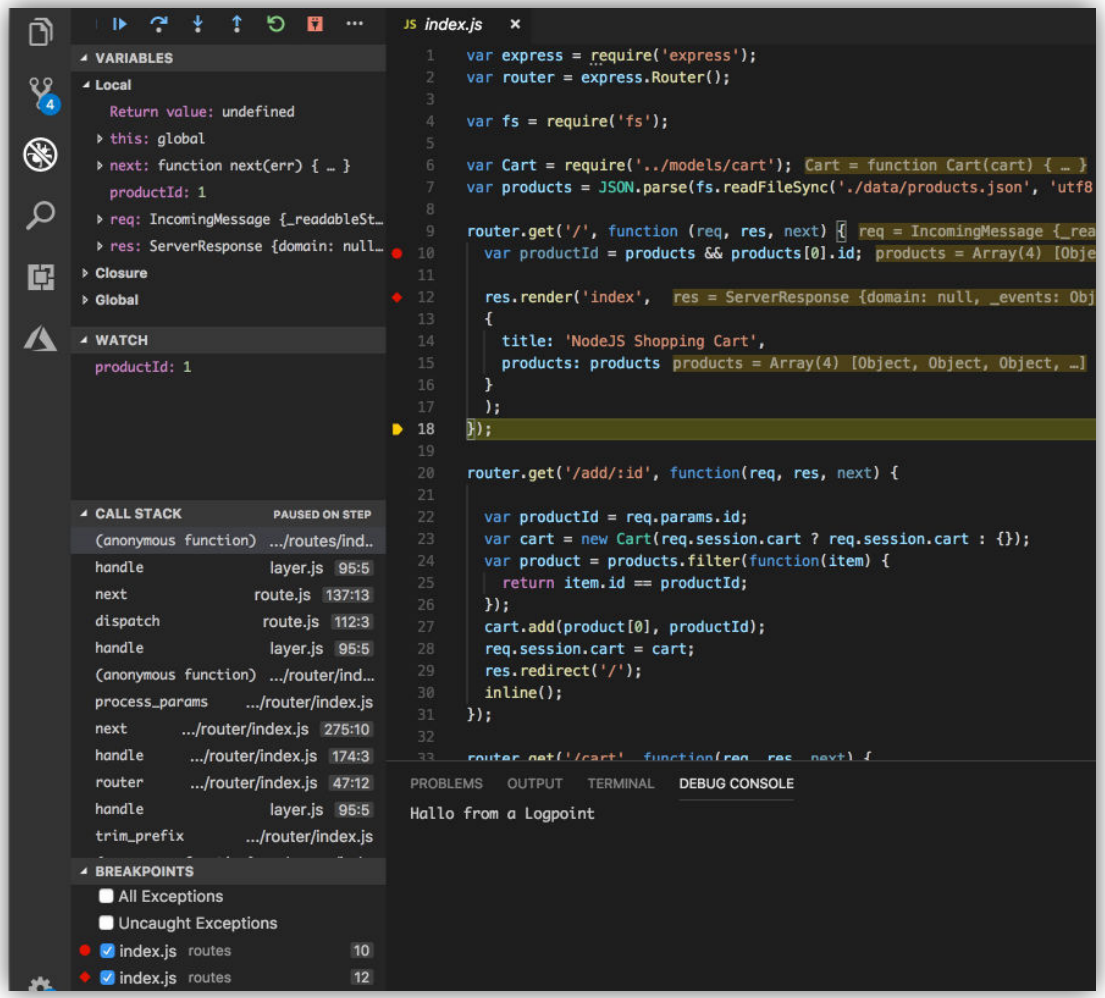
Place breaking points on code

```go
// UnmarshalYAML is a custom unmarshaler that wraps strings in arrays
func (a *StringArray) UnmarshalYAML(unmarshal func(interface{}) error) error {
    var strings []string
    if err := unmarshal(&strings); err != nil {
        var str string
        if err := unmarshal(&str); err != nil {
            return err
        }
        *a = []string{str}
    } else {
        *a = strings
    }
    return nil
}

// FlagArray is a wrapper for an array of strings
type FlagArray []string

// UnmarshalYAML is a custom unmarshaler that wraps strings in arrays
func (a *FlagArray) UnmarshalYAML(unmarshal func(interface{}) error) error {
    var flags []string
    if err := unmarshal(&flags); err != nil {
        var flagstr string
        if err := unmarshal(&flagstr); err != nil {
            return err
```

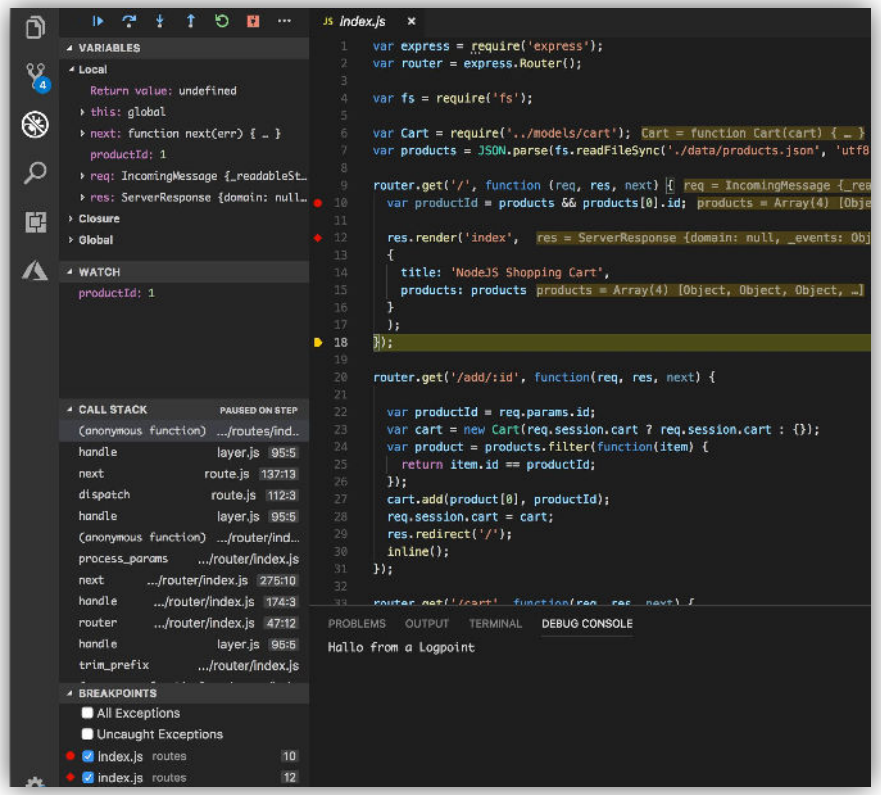
View calls/ variables, resume/step forward



Live debugging

Quick isolation of errors

- View exact running state
- Pause/resume any line
- No printf's and printouts needed anymore



CREATE

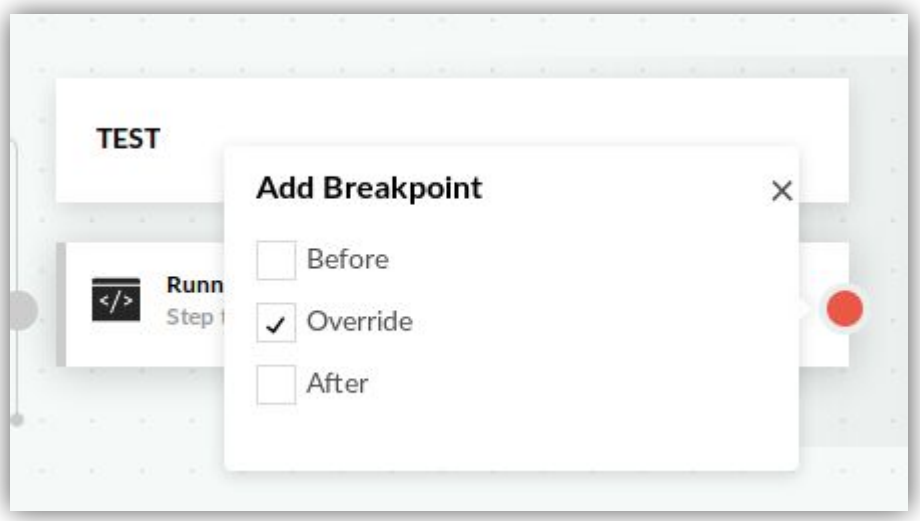
Developers

We love IDEs and live debugging

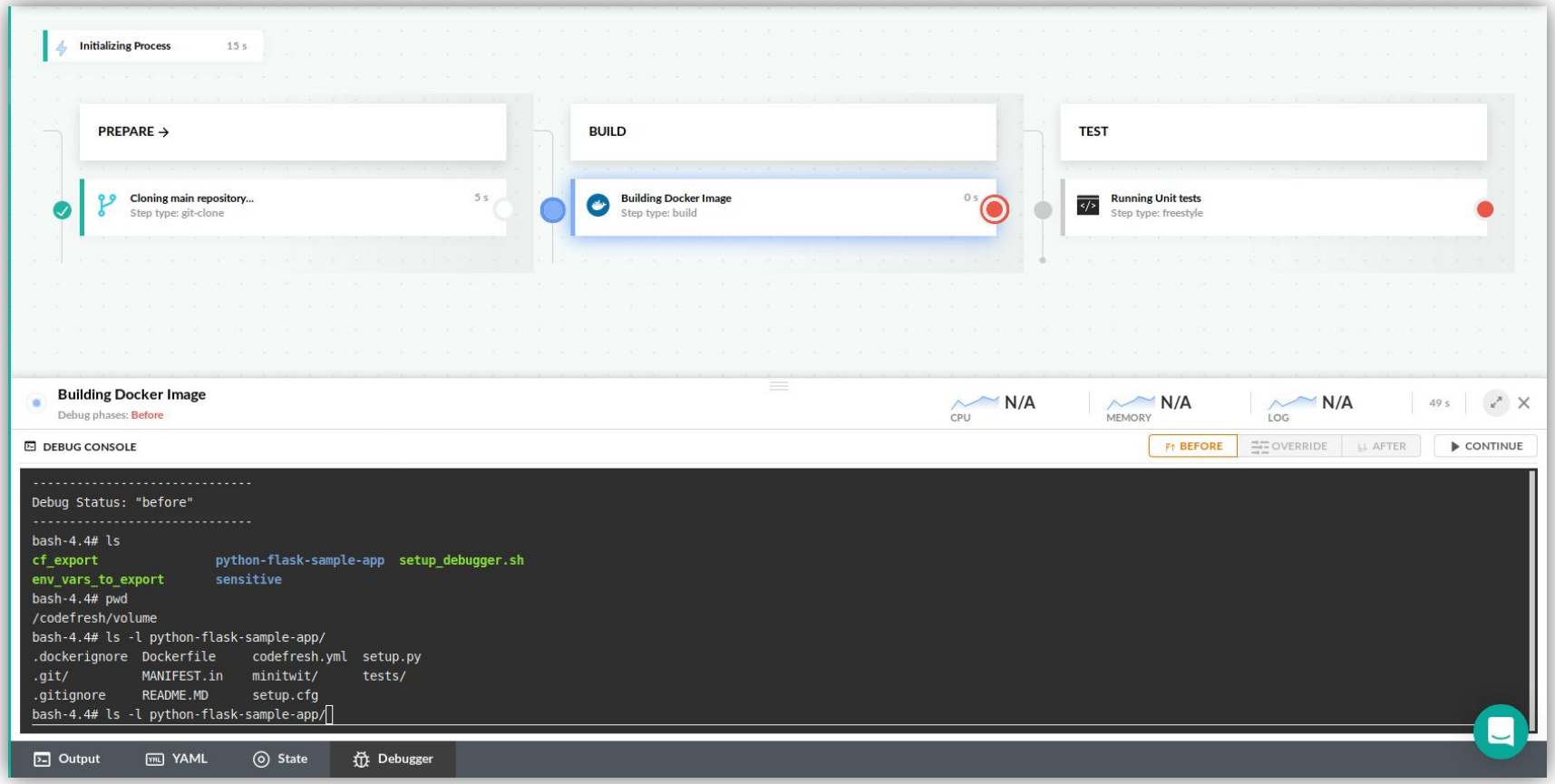
Operators

We want live debugging as well!

CI/CD Live Debugging



CI/CD live debugging



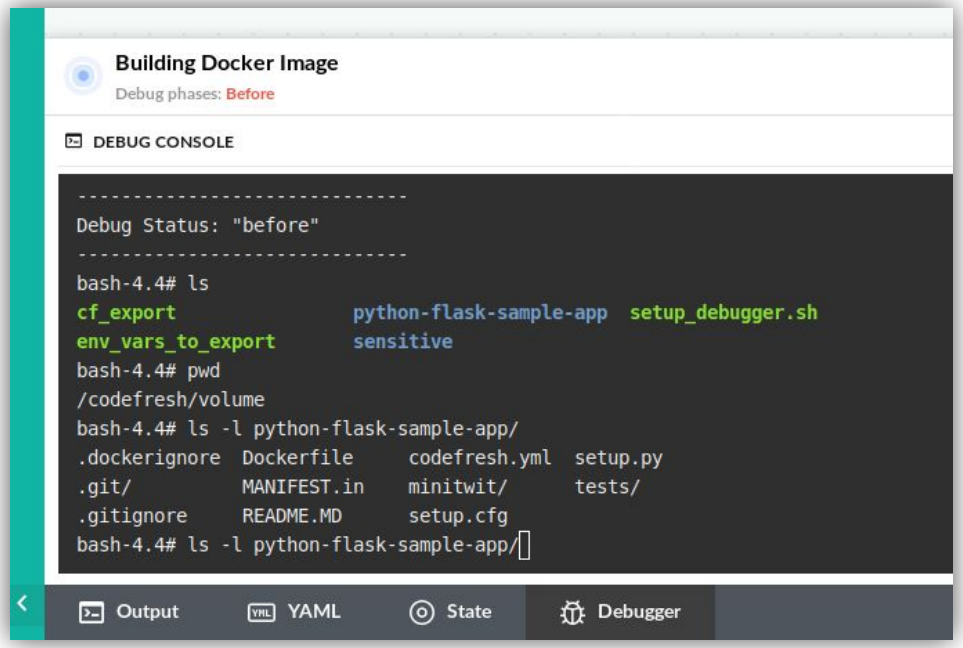
Demo: Simple Debugging

Demo: Container Tooling

Move in/out of containers

Install any tool

- Use Curl, nano, go, node
- Example: apk add curl
- All existing tools in your image



Demo: Dynamic Tools

Install tools on demand

Demo: Services

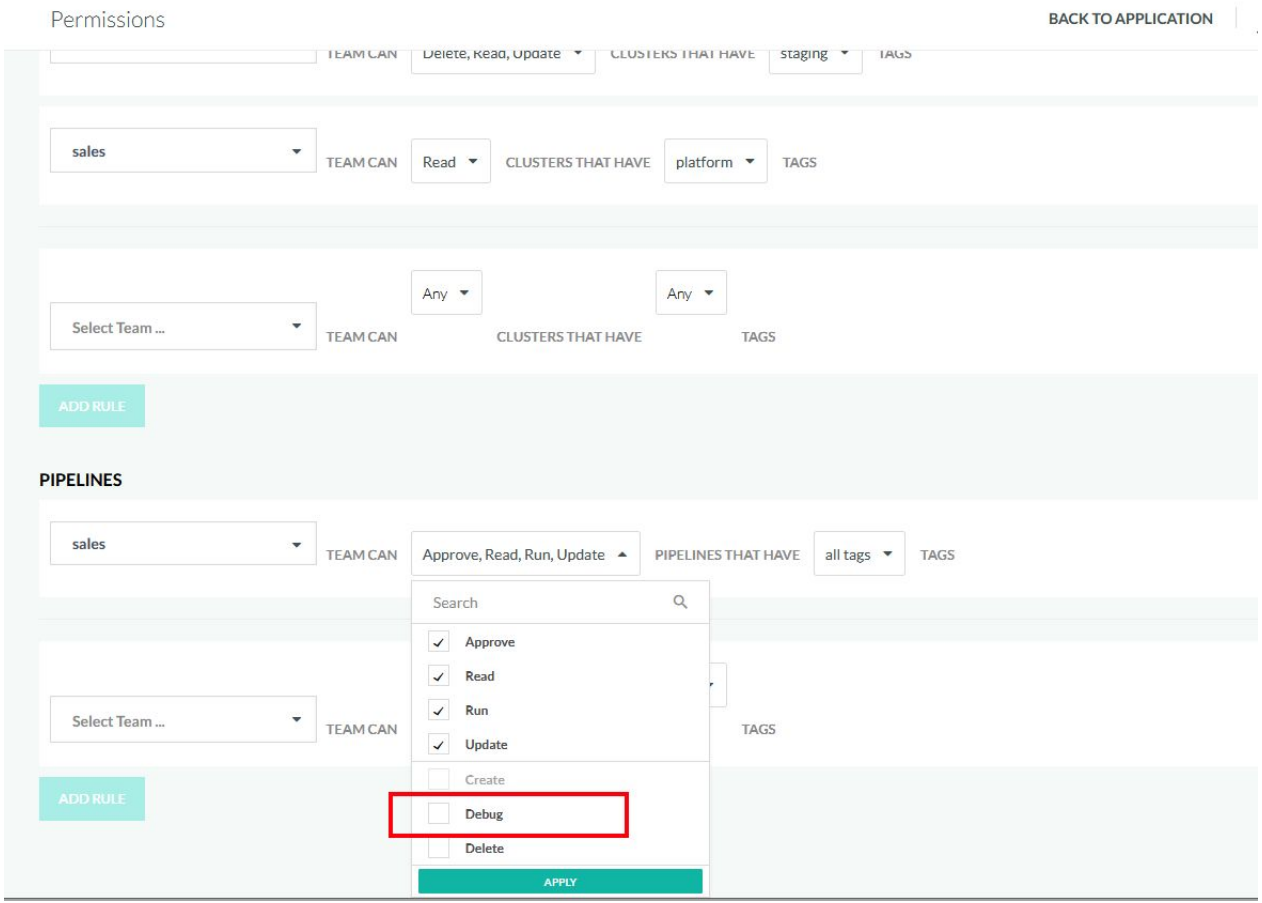
Access running services

What about Security?

Codefresh access controls

Control

- Pipelines
- Clusters
- From other repo



Pro debugging plan

- Breakpoints on any step
- Debug window
- Pause/Resume
- Standard commands (cd, ls, printenv etc)

Enterprise debugging plan

- Includes Pro plan
- Add extra tools dynamically (e.g. curl, wget, redis-cli)
- Shell auto-completion
- Restrict debug access

Summary

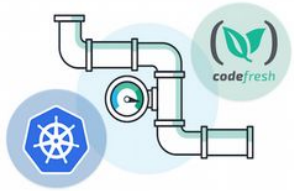
- World's first CI/CD debugger
- Place breakpoints on pipeline steps
- Stop/Resume pipeline
- Inspect files/variables/services
- Install tools on demand
- Lock-down debugging controls

About Kostis Kapelonis

Helm Tutorial | February 7, 2019

Using a Kanban board to manage and promote Helm Releases

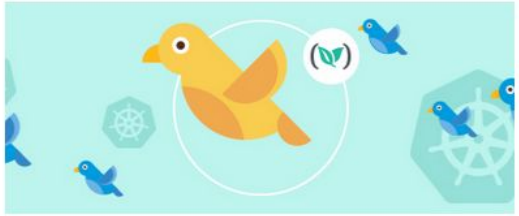
Kostis Kapelonis



Continuous Integration | September 11, 2018

Programmatic Creation of Codefresh Pipelines – Part1

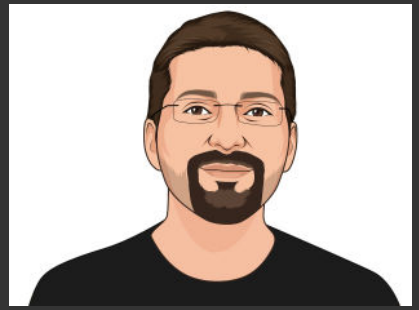
Kostis Kapelonis



Continuous Deployment/Delivery | August 30, 2018

Fully automated canary deployments in Kubernetes

Kostis Kapelonis



Questions?

Build Fast, Deploy Faster

Signup for a FREE account with UNLIMITED builds

& schedule a 1:1 with our experts at https://codefresh.io