

{unscripted}

hosted by harness

How We Fired Our Butlers: Drone CI at Scale

OCTOBER 21 - 22, 2020 | VIRTUAL
9AM - 4PM PT | EACH DAY

www.unscriptedconf.io



Who are we?

- Jim Sheldon / Principal Software Engineer
- Been with Meltwater for over 6 years
- Lives in New Hampshire, USA
- Married with three children
- Enjoys boating, hiking, skiing



- Brian Burnett / Senior Software Engineer
- Been with Meltwater for over 5 years
- Statistics junkie
- Likes to automate everything
- Enjoys being outdoors with my family
- Learned CI/CD from Jim 

What is Meltwater?

**“Deliver the leading integrated media
and social intelligence solution for
PR, Comms and Marketing
departments”**

What is Meltwater?

Media Monitoring

Media Outreach

Social Listening

Social Publishing & Engagement

Social Influencer Management

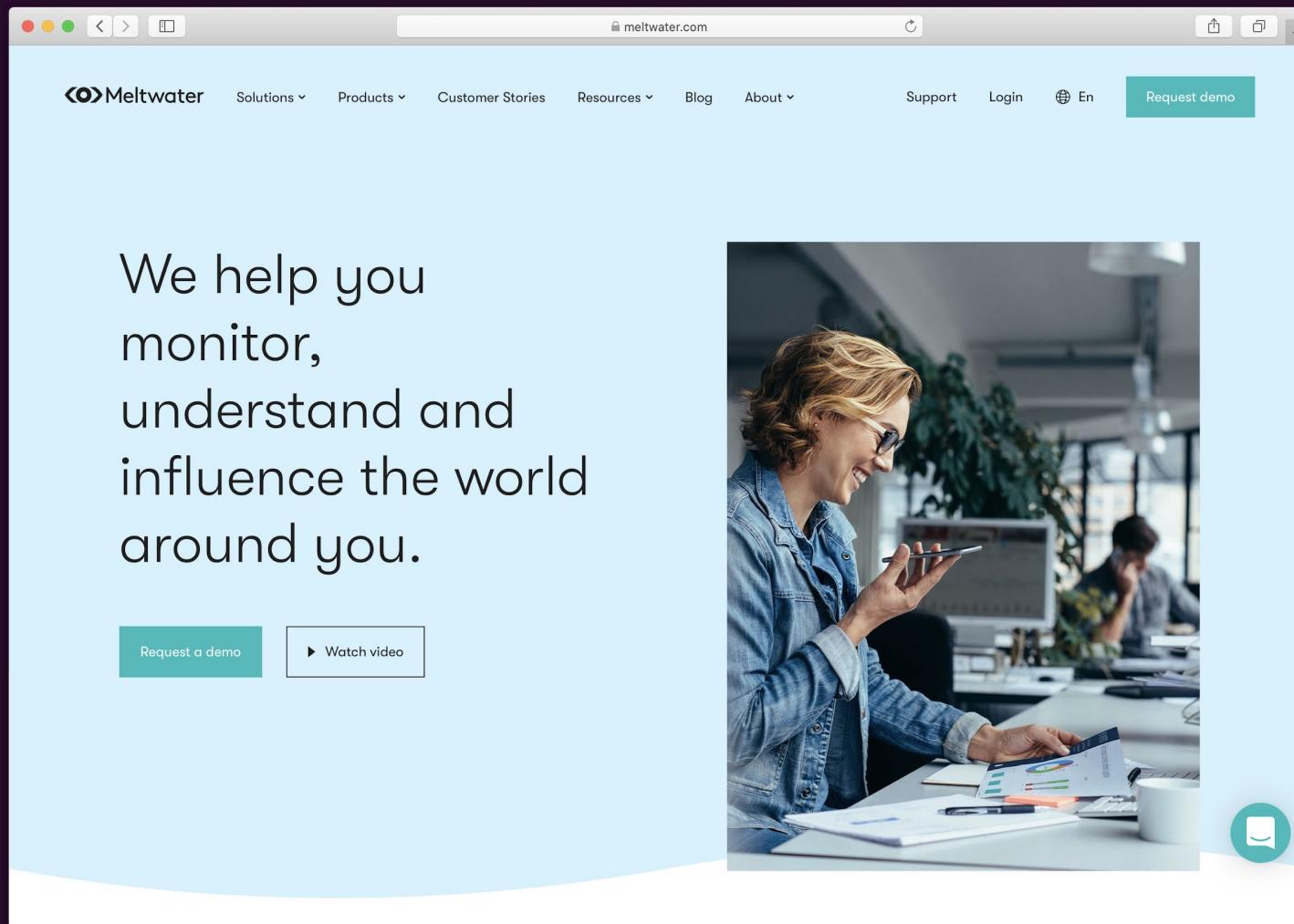
What is Meltwater?

Some of our 30K+ clients

 Spotify®	 ups	 acer	 Square	 Allianz
 Google	 Ford	 GAP	 RollingStone	 MetLife
 Pfizer	 Yale™	 Santander	 UBER	 WWF
 TESLA	 CHANEL	Qualcomm	 BARCLAYS	 Johnson & Johnson

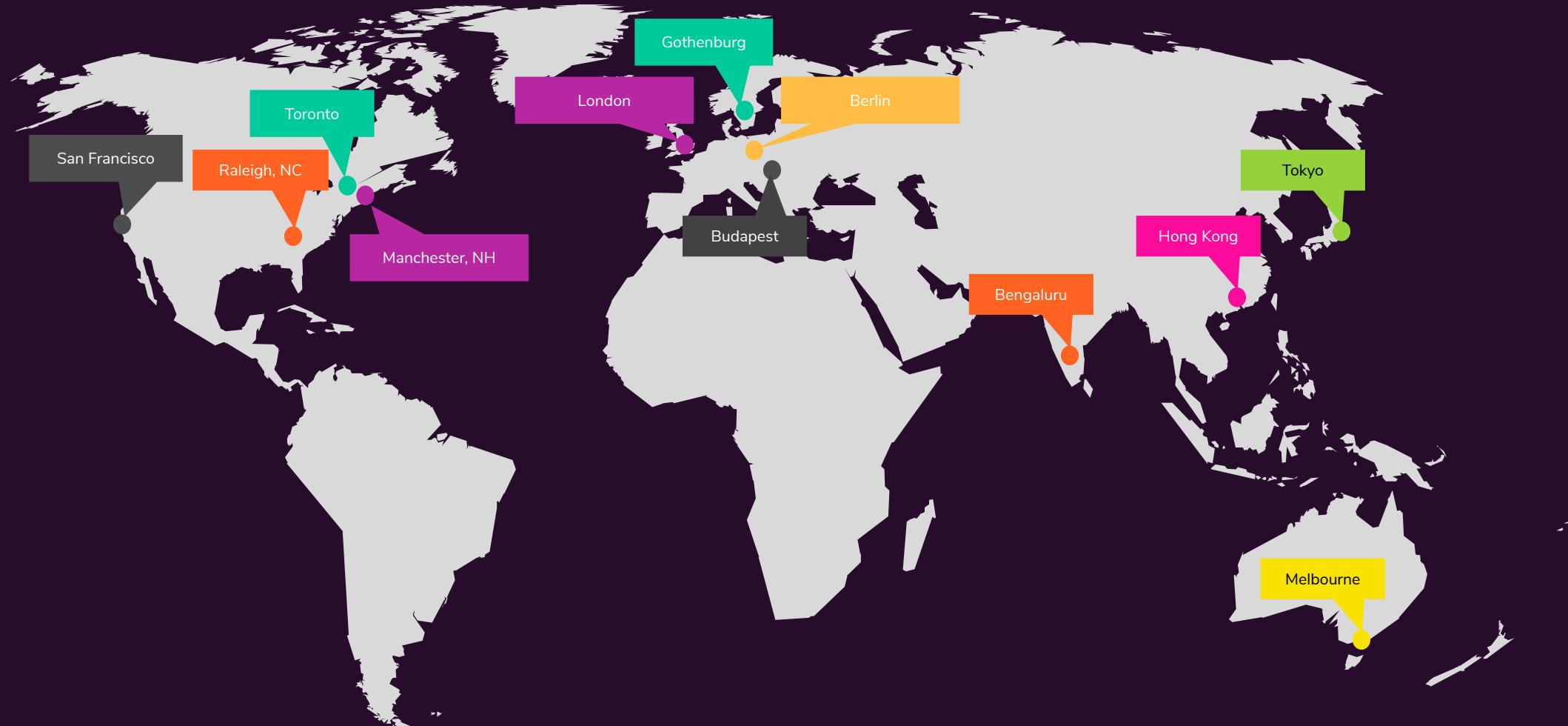
What is Meltwater?

meltwater.com



The screenshot shows the Meltwater website homepage. At the top, there's a navigation bar with links for Solutions, Products, Customer Stories, Resources, Blog, About, Support, Login, and a language selector (En). A prominent green button labeled "Request demo" is located on the right side of the nav bar. Below the nav bar, a large white text area contains the tagline: "We help you monitor, understand and influence the world around you." Two buttons are positioned below this text: a teal "Request a demo" button and a white "Watch video" button with a play icon. To the right of the text area is a photograph of a woman with blonde hair, wearing glasses and a denim jacket, sitting at a desk in an office. She is smiling and holding a smartphone to her ear. On her desk, there are papers with charts and graphs, a keyboard, and a mouse. In the background, other office workers are visible. A small teal circular icon with a white play symbol is located in the bottom right corner of the photo area.

What is Meltwater?

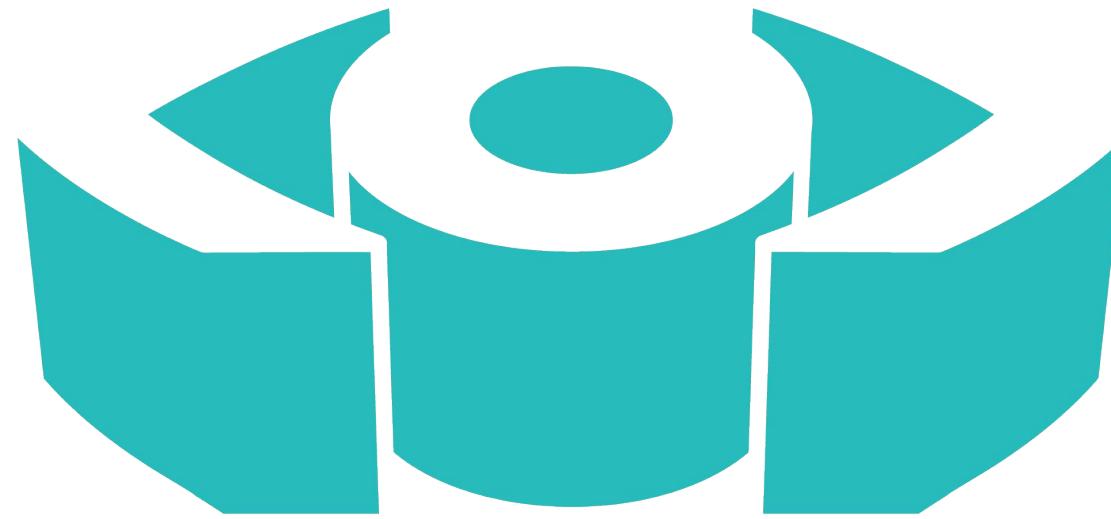


Engineering: ~300 Engineers
Company: ~2000 Employees

in 17 Offices
in 55 Offices (25 Countries)

What is Foundation?

“We enable teams to **accelerate** without thinking too much about the infrastructure, allowing a rapid path from **ideation** to **prototype** to providing **business value**”



FOUNDATION



Outline

- How we replaced multiple CI/CD services with Drone
- Features of our internal Drone service



Replacing CI/CD services with Drone

We didn't need all of these on day 1, just flexibility to create them as time progressed...

As a team, we had identified over 30 dreams for CI/CD

Feature Manifesto	CircleCI Ent.	Travis Ent.	Drone	JenkinsBlue	Concourse	CodePipeBuild	Jenkins/CloudBees	Arnake
User Interaction via GitHub (Enabling a repo, initial config, badges)	5	4	4	2	3	5	0	4
GitHub Authentication (By organization)	2	1	5	5	5	0	0	5
Github Organizations (Team Access Control)	2	3	5	1	0	0	0	5
Ability to use pre-generated kubecfg (maybe docker image?)	5	4	5	2	3	0	0	5
Webhooks as trigger to start workflow / pipeline	5	5	5	3	2	3	0	5
Secrets: Confirm secrets cannot be decrypted outside build agent	3	3	5	2	1	2	0	4
Secrets: Restrict visibility to pipeline segment	0	0	5	0	0	0	0	4
Secrets: Isolation to the step they are run within (with possible Globals)	0	0	5	0	0	0	0	4
Pipeline gives feedback / assist that client can know status at any given time	4	3	4	4	5	5	0	5
Metrics Exportability (prometheus)	4	4	5	5	5	1	0	5
Datastore Persistence (for build history)	4	4	5	4	5	5	0	5
Private / Non-cloud build agents (datacenter support)	5	5	5	5	3	0	0	4
Proven backwards compatibility across versions / config styles	5	0	5	0	0	0	0	0
Local execution environment to test pipeline changes	0	0	5	0	0	0	5	0
HA Setup	3	0	3	4	3	5	0	4
And the winner is!!!	72.41%	57.93%	95.17%	54.48%	57.24%	46.90%	3.45%	71.72%
GitHub Commits (last yr)	69	8	279	655	1430		808	299
GitHub Committers	21	95	231	62	166		5	77
GitHub Org Members	21	32	9	648	3		648	12

Our Drone Service

1,466 repos



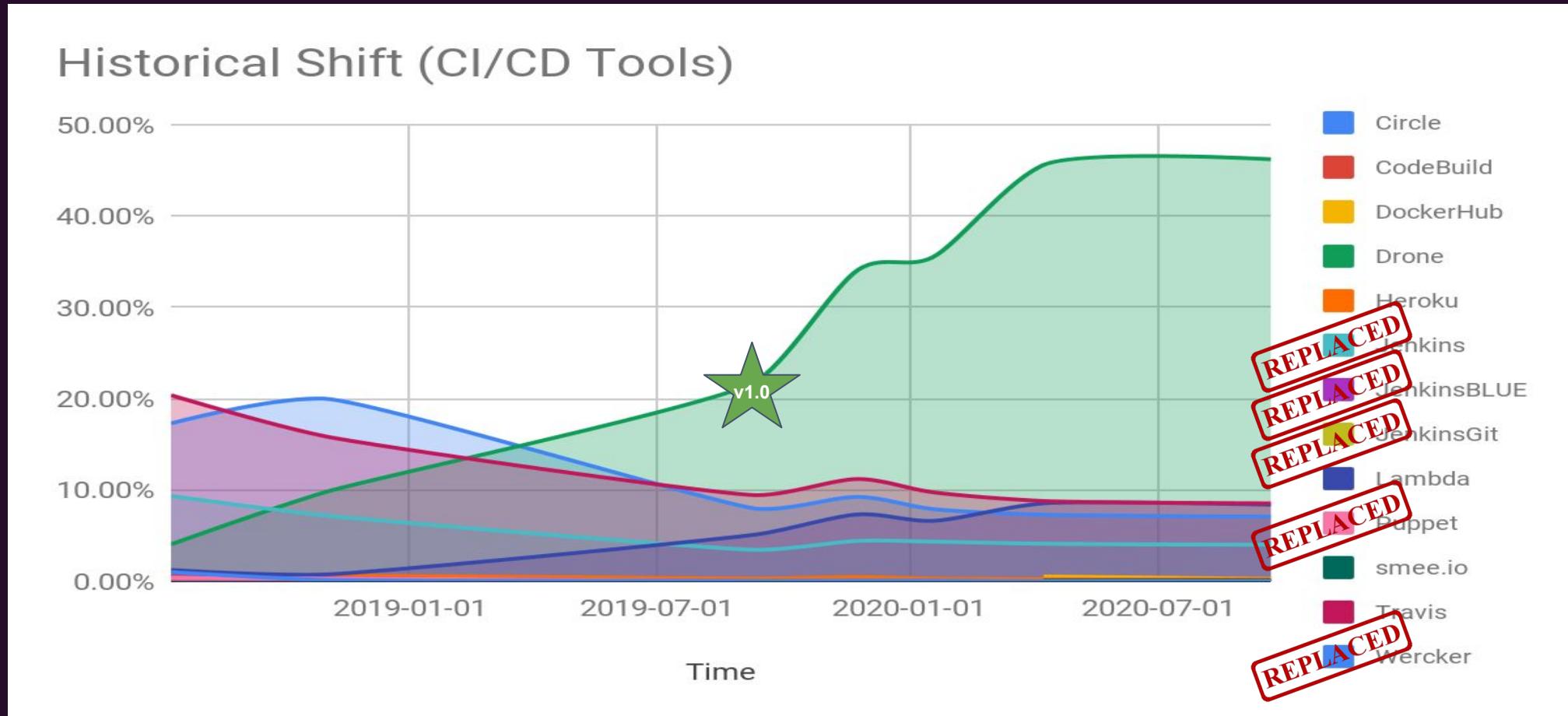
222 active users



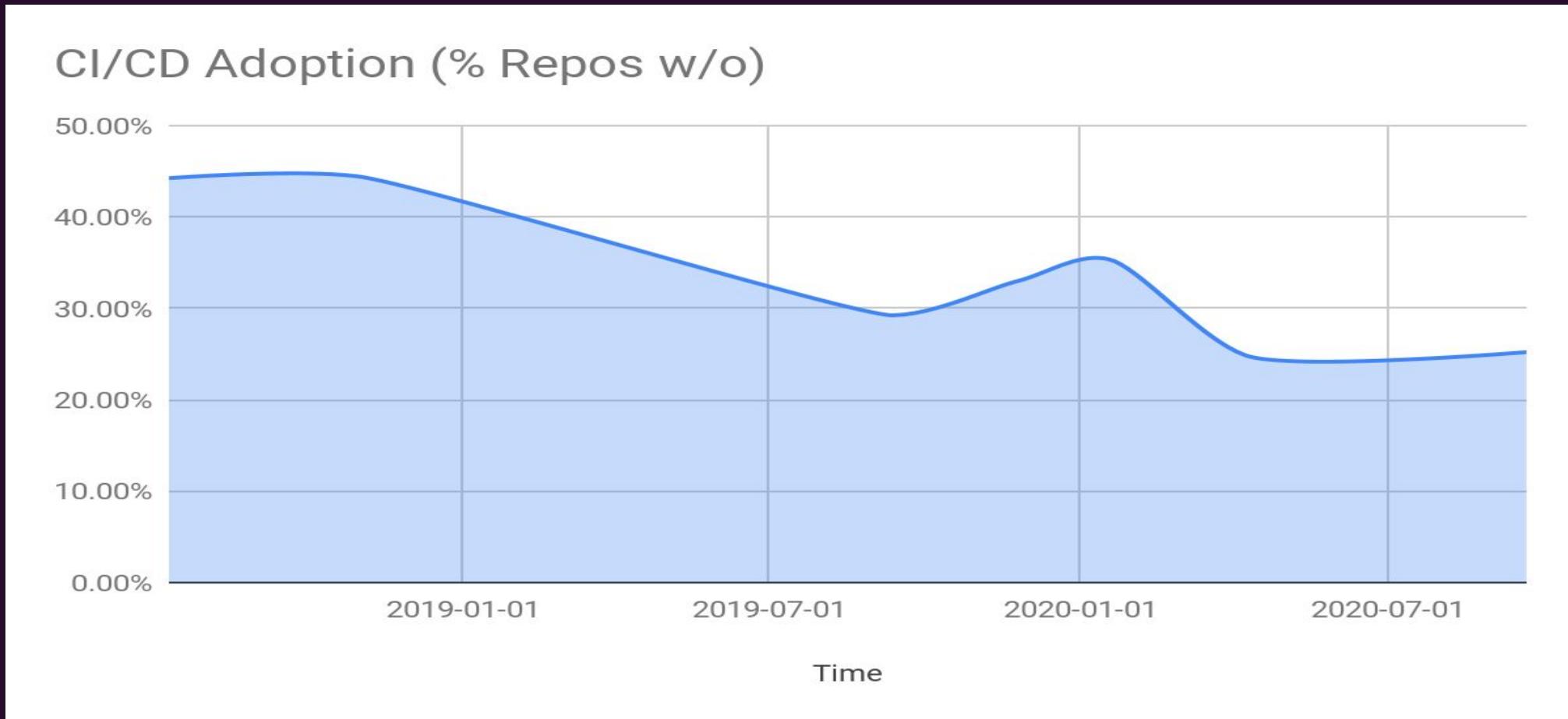
1,200+ pipelines/day



Replacing CI/CD services with Drone



Replacing CI/CD services with Drone



What is Drone?

cloud.drone.io

The screenshot shows the homepage of the Drone Cloud website. At the top, there's a navigation bar with links for Cloud, Enterprise, Plugins, Support, and a prominent green LOGIN button. Below the navigation is a dark header section containing the Drone logo (a stylized white 'D' inside a circle), the text "Continuous Integration, Free for the Open Source Community", and a brief description: "Drone Cloud is a free Continuous Integration service for the Open Source community, powered by blazing fast bare-metal servers." There are two buttons at the bottom of this section: a green LOGIN button and a white READ THE DOCS button. To the right of the text is a 3D-style illustration of several server racks connected by glowing blue lines, symbolizing a network or cloud infrastructure. Below this, a large white area features the text "Accelerating Open Source Development". At the very bottom, there are three callout boxes: "Multiple Architectures" (with the subtext "Our goal is to upstream all the"), "Blazing Fast, Bare Metal Servers" (with the subtext "Drone Cloud would not be"), and "100% free for Open Source" (with the subtext "Drone Cloud would not be").

What is Drone?

The screenshot shows the Drone CI web interface. At the top, there's a search bar with placeholder text "Search repositories or jump to ...". Below it, a navigation bar includes links for "Repositories" (highlighted), "meltwater/terraform-aws-asg-dns-handler", and "#58". On the right side of the header are icons for "0 notifications" and a user profile.

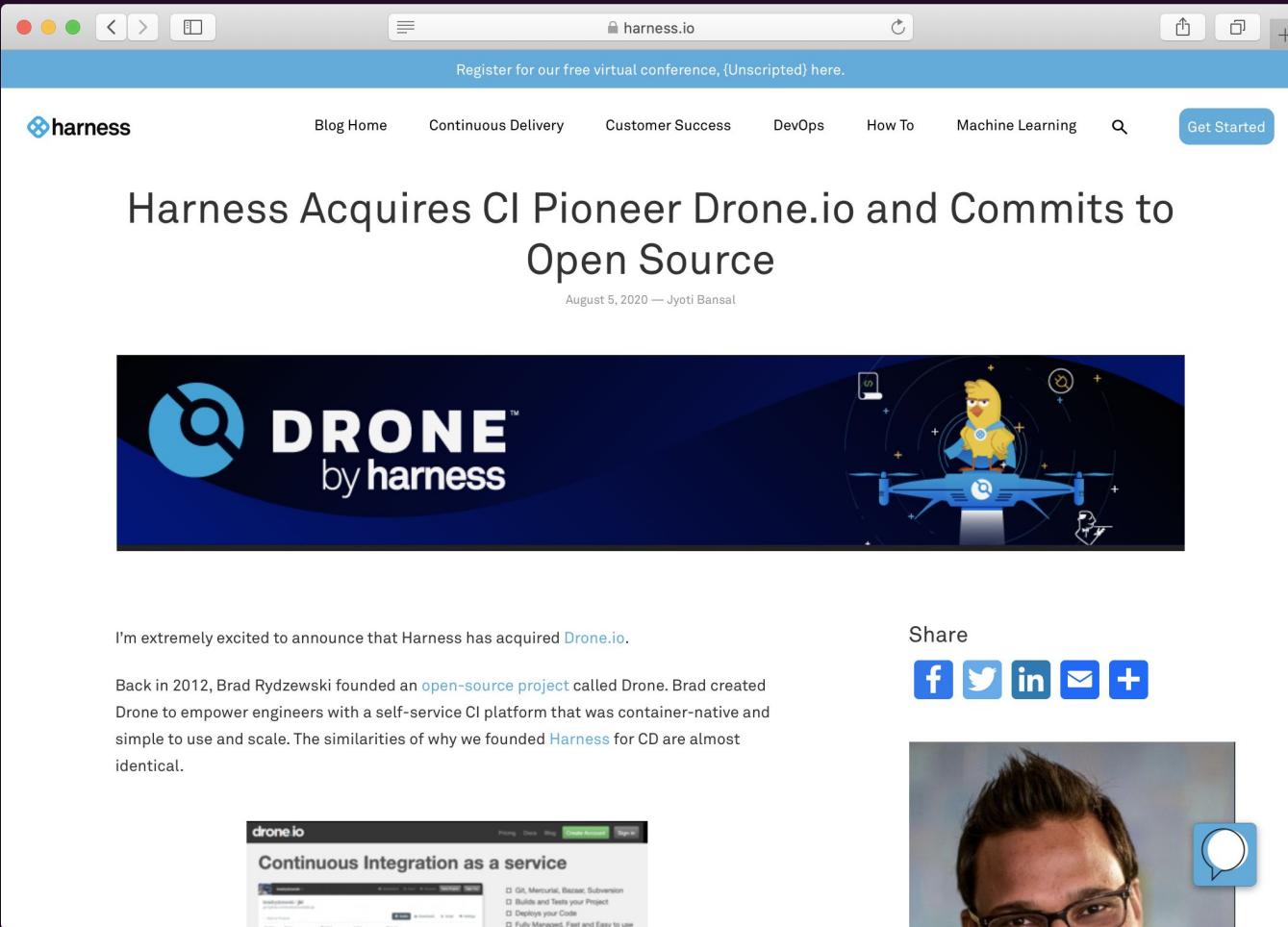
The main content area displays a pull request titled "terraformer-aws-asg-dns-handler #58. Add possibility to use public IP". It shows a green checkmark icon next to the title, indicating the pull request has been merged. Below the title, it says "feraudet opened pull request #26 to master" and "07:32 · 2 months ago".

On the left, there's an "ACTIVITY FEED" section with three items: "example" (status: green checkmark, timestamp: 07:32), "clone" (status: green checkmark, timestamp: 00:03), and "test" (status: green checkmark, timestamp: 07:29). To the right of the feed is a detailed log for the "test" step, titled "example - test 07:29". The log shows several lines of command-line output:

```
156 TestAwsDnsRecordName 2020-07-13T19:56:48Z logger.go:66:  
    module.autoscale_dns.aws sns topic subscription.autoscale_handling: Creating...  
157 TestAwsDnsRecordName 2020-07-13T19:56:49Z logger.go:66:  
    module.autoscale_dns.aws sns topic subscription.autoscale_handling: Creation  
    complete after 1s [id=arn:aws:sns:eu-west-1:161122428365:asg-handler-vpc-asg-  
    handler:1aa07c20-692c-4607-b08d-89694eb3c80c]  
158 TestAwsDnsRecordName 2020-07-13T19:56:50Z logger.go:66:  
    module.vpc.aws nat gateway.this[1]: Still creating... [10s elapsed]  
159 TestAwsDnsRecordName 2020-07-13T19:56:50Z logger.go:66:  
    module.vpc.aws nat gateway.this[0]: Still creating... [10s elapsed]  
160 TestAwsDnsRecordName 2020-07-13T19:56:50Z logger.go:66:  
    module.vpc.aws nat gateway.this[2]: Still creating... [10s elapsed]  
161 TestAwsDnsRecordName 2020-07-13T19:56:53Z logger.go:66:  
    module.vpc.aws vpn gateway.this[0]: Creation complete after 18s [id=vgw-
```

Harness Acquires Drone

harness.io/blog



The screenshot shows a web browser window for harness.io. The header includes a navigation bar with links to Blog Home, Continuous Delivery, Customer Success, DevOps, How To, Machine Learning, a search icon, and a 'Get Started' button. A blue banner at the top says 'Register for our free virtual conference, {Unscripted} here.' Below the banner, the main title of the post is 'Harness Acquires CI Pioneer Drone.io and Commits to Open Source'. Underneath the title is the author's name, 'August 5, 2020 — Jyoti Bansal'. The post features a large image of a cartoon drone character with a speech bubble, next to the 'DRONE by harness' logo. The text of the post discusses the acquisition of Drone.io and its history. At the bottom, there is a screenshot of the Drone.io website and a 'Share' button with social media icons.

I'm extremely excited to announce that Harness has acquired [Drone.io](#).

Back in 2012, Brad Rydzewski founded an [open-source project](#) called Drone. Brad created Drone to empower engineers with a self-service CI platform that was container-native and simple to use and scale. The similarities of why we founded [Harness](#) for CD are almost identical.

[drone.io](#)

Continuous Integration as a service

Drone integrates with:

- Git, Mercurial, Bazaar, Subversion
- Builds and Tests your Project
- Deploys your Code
- Fully Managed, Fast and Easy to use

Share

f t in e +



Replacing CI/CD Services With Drone

- TravisCI and CircleCI
- Jenkins (We had 7 of them)
- AWS CodeBuild/CodePipeline
- ...and more



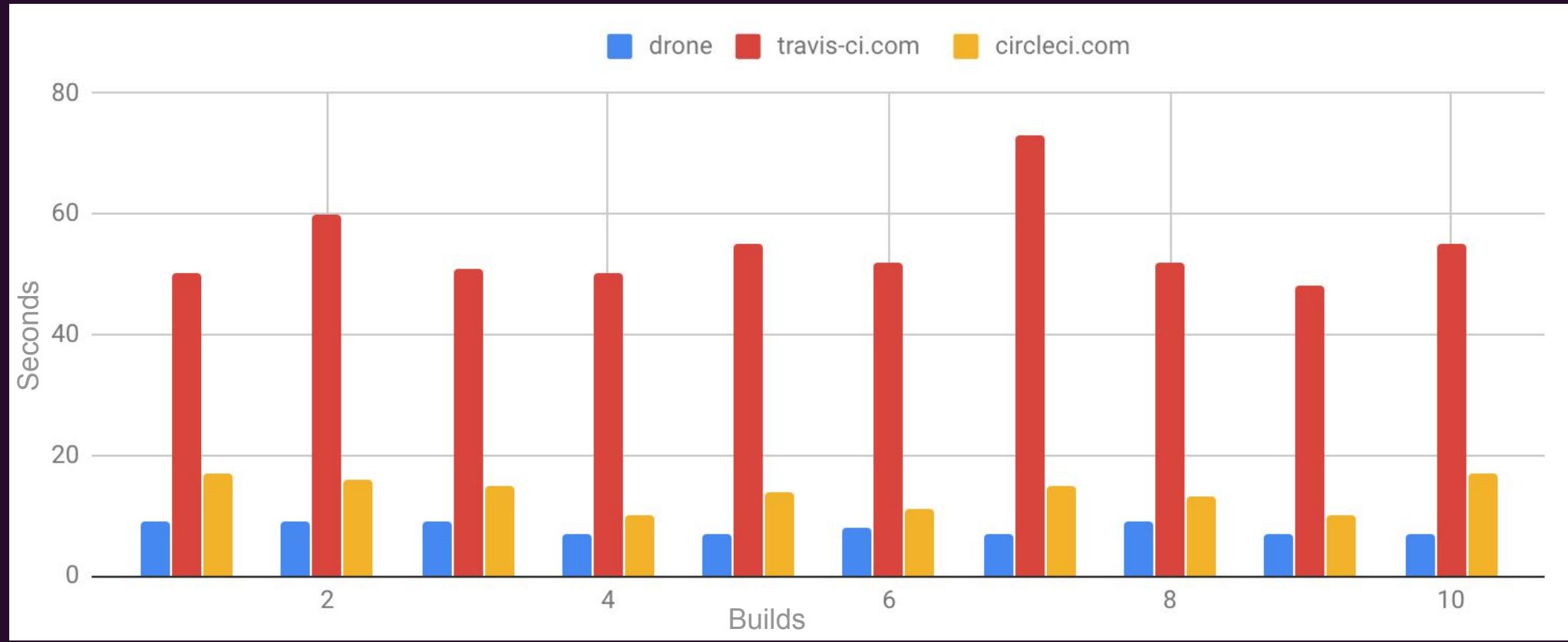
TravisCI and CircleCI

- We did not have an initiative to move teams away from TravisCI and CircleCI
- Multiple teams started moving builds from TravisCI and CircleCI to Drone almost immediately after we launched our service



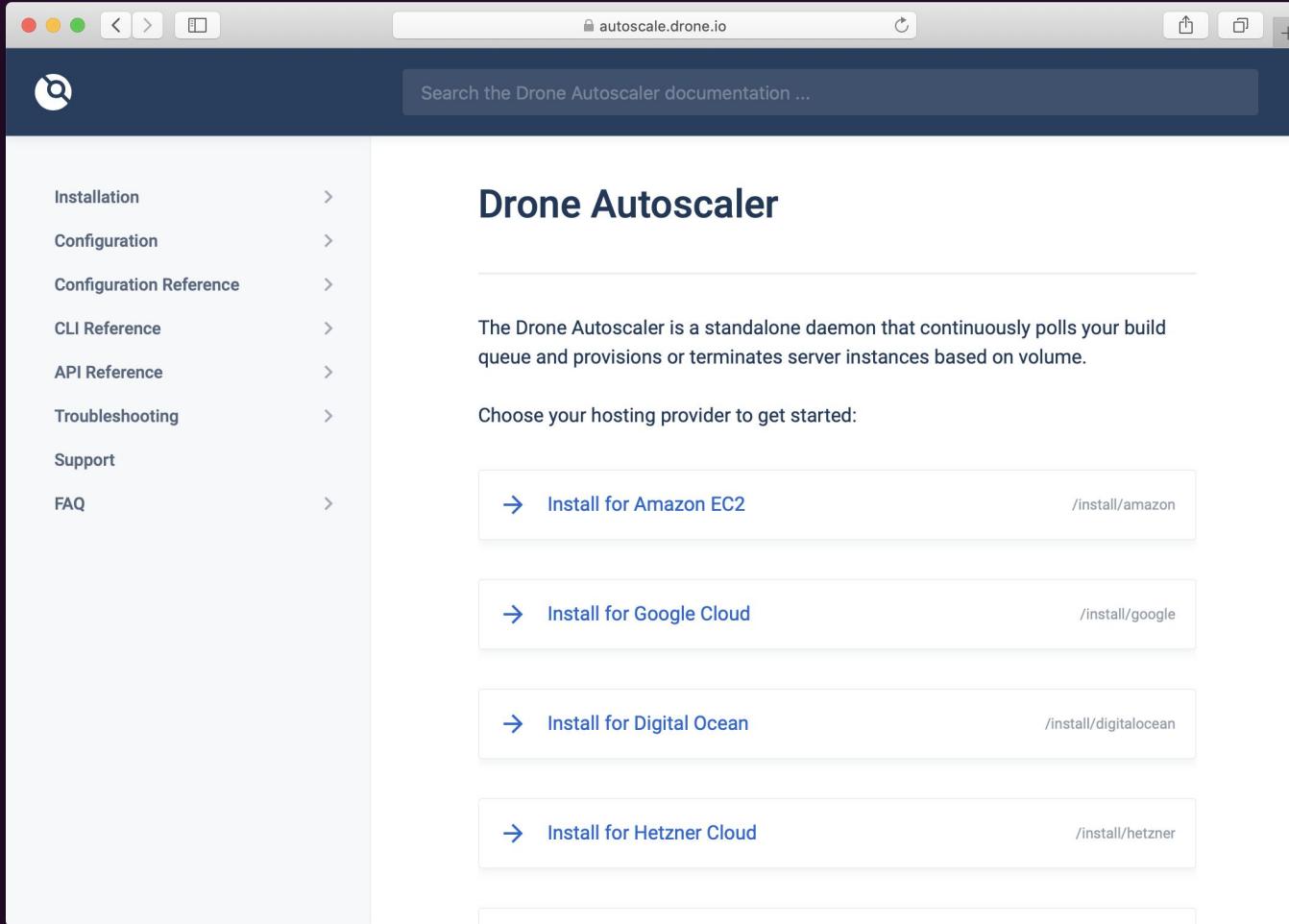
Drone vs TravisCI/CircleCI

Time from commit until first pipeline step command executes



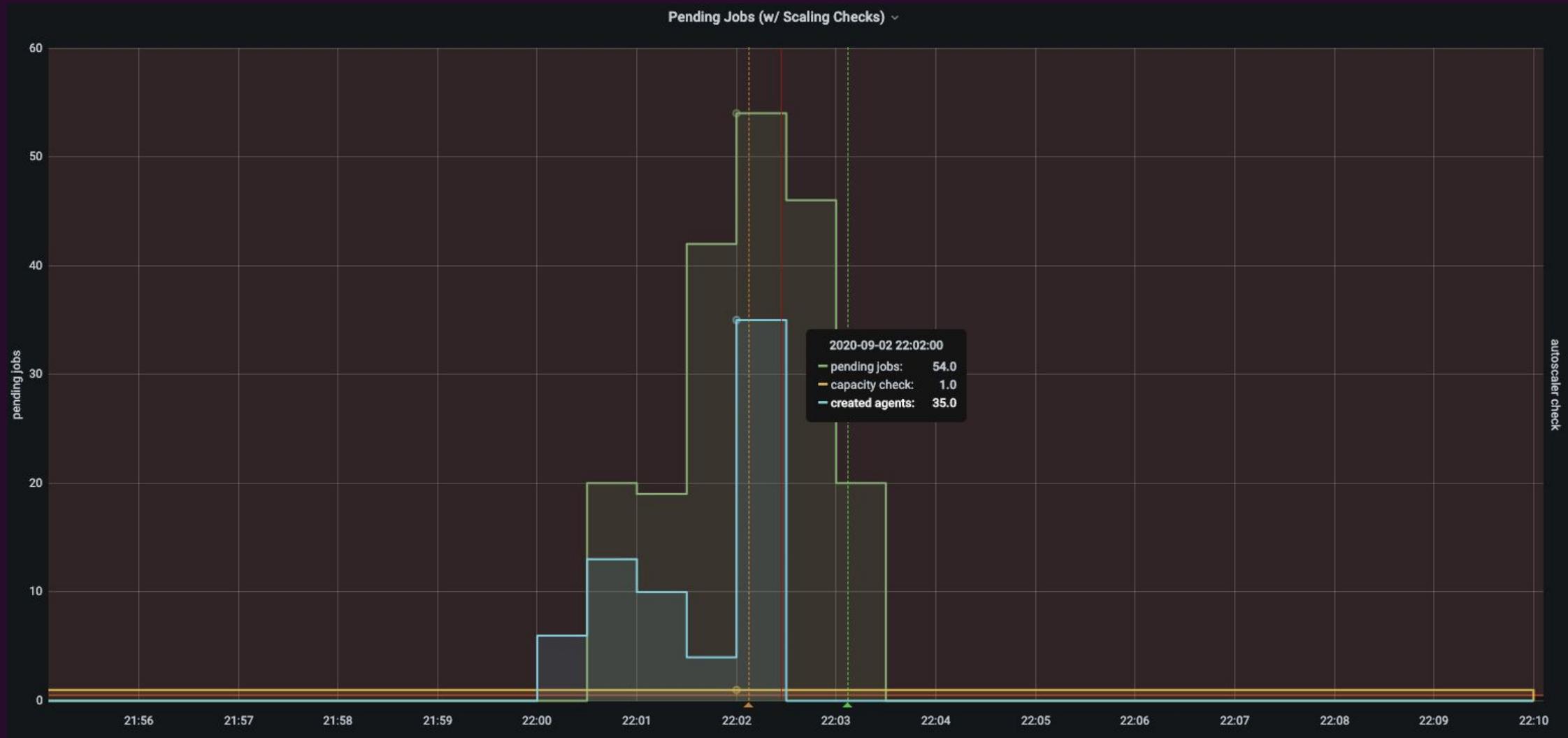
Running Drone: Autoscaler

autoscale.drone.io

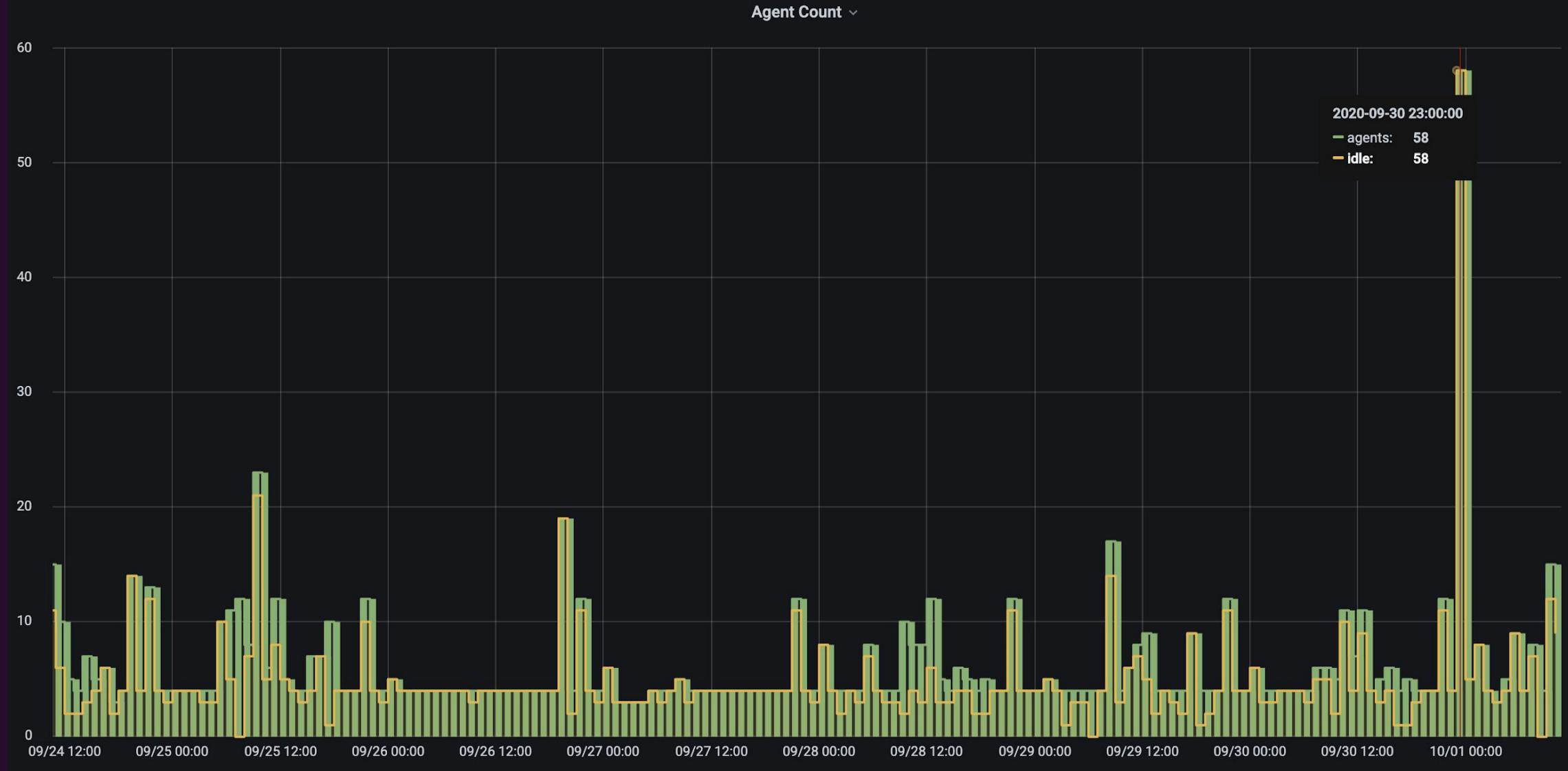


The screenshot shows a web browser window displaying the Drone Autoscaler documentation at autoscale.drone.io. The page has a dark blue header with a search bar. On the left, there's a sidebar with links to Installation, Configuration, Configuration Reference, CLI Reference, API Reference, Troubleshooting, Support, and FAQ. The main content area is titled "Drone Autoscaler" and describes it as a standalone daemon that polls a build queue and provisions or terminates server instances based on volume. It then prompts the user to choose a hosting provider to get started, listing four options: "Install for Amazon EC2" (with link </install/amazon>), "Install for Google Cloud" (with link </install/google>), "Install for Digital Ocean" (with link </install/digitalocean>), and "Install for Hetzner Cloud" (with link </install/hetzner>).

Running Drone: Autoscaler



Running Drone: Autoscaler



Running Drone: Autoscaler

```
● ● ●  
  
docker run -d \  
  -v /var/lib/autoscaler:/data \  
  -e DRONE_POOL_MIN={DRONE_POOL_MIN} \  
  -e DRONE_POOL_MAX={DRONE_POOL_MAX} \  
  -e DRONE_SERVER_PROTO={DRONE_SERVER_PROTO} \  
  -e DRONE_SERVER_HOST={DRONE_SERVER_HOST} \  
  -e DRONE_SERVER_TOKEN={DRONE_SERVER_TOKEN} \  
  -e DRONE_AGENT_TOKEN={DRONE_AGENT_TOKEN} \  
  -e DRONE_AMAZON_REGION={DRONE_AMAZON_REGION} \  
  -e DRONE_AMAZON_SUBNET_ID={DRONE_AMAZON_SUBNET_ID} \  
  -e DRONE_AMAZON_SECURITY_GROUP={DRONE_AMAZON_SECURITY_GROUP} \  
  -e DRONE_AMAZON_SSHKEY={DRONE_AMAZON_SSHKEY} \  
  -e AWS_ACCESS_KEY_ID={AWS_ACCESS_KEY_ID} \  
  -e AWS_SECRET_ACCESS_KEY={AWS_SECRET_ACCESS_KEY} \  
  -p 8080:8080 \  
  --restart=always \  
  --name=autoscaler \  
drone/autoscaler
```

Running Drone: Autoscaler



```
DRONE_AGENT_ENVIRON=DRONE_REGISTRY_ENDPOINT=https://registryplugin.drone.example.com:3000,DRONE_REGISTRY_SECRET=f3499169553b9605dabe93bbe306e51a  
DRONE_DATABASE_DRIVER=mysql  
DRONE_DATABASE_DATASOURCE=root:password@tcp(1.2.3.4:3306)/drone?  
parseTime=true  
DRONE_AGENT_IMAGE=drone/drone-runner-docker:1.5.1  
DRONE_AMAZON_INSTANCE=t2.xlarge  
DRONE_AMAZON_IMAGE=ami-9ee7a401f352b7593  
DRONE_AMAZON_VOLUME_SIZE=100  
DRONE_AMAZON_DEVICE_NAME=/dev/xvda  
DRONE_AMAZON_TAGS=Prometheus:node_exporter  
DRONE_AMAZON_USERDATA_FILE=/path/to/cloud-init.yml  
DRONE_AMAZON_IAM_PROFILE_ARN=arn:aws:iam::123456789012:root
```



Retiring Jenkins



- Used for running Terraform in multiple accounts, with a Jenkins agent running in each account
- Used for deploying to four Marathon clusters (two in datacenters, two in AWS), where the Jenkins agent ran on a machine in each datacenter



Retiring Jenkins: Why?

- Bringing up new Jenkins agents was very time consuming
- Changes to build scripts (DSL Plugin) required coordination with multiple teams
- Regular connectivity problems between the agents and server
- Server and plugins hadn't been updated in years, updating would have been a significant investment
- Distributed agents made it so that you could not upgrade / change out the server beneath without an outage

Retiring Jenkins: Terraform

- Combined instance role of our Drone agents along with “external ID” to allow teams to safely authenticate to apply their Terraform configuration

Relevant talk: youtu.be/oK0KHT6pcN8
“Automate your infrastructure with GitOps, Terraform and Drone”

Retiring Jenkins: Marathon

- “Monolithic” git repository where different steps must run depending on file(s) changed.
- Pipelines would have to run on different agents depending on the file(s) changed.

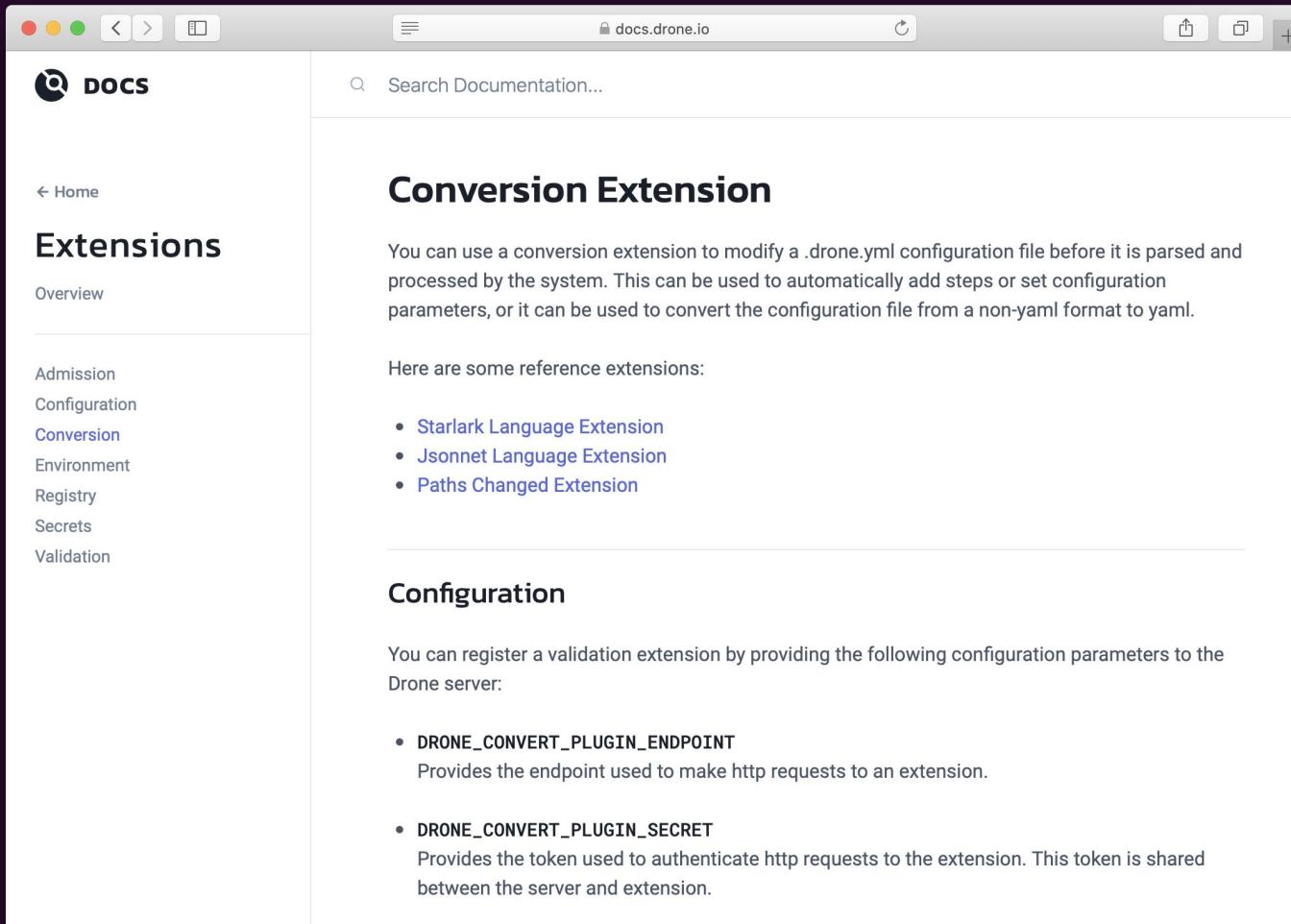
Ways The Butler Fought Back

- Build Path Sensitivity
- Build Agent Selection



Drone Conversion Extensions

docs.drone.io/extensions/conversion



The screenshot shows a web browser window displaying the Drone documentation at docs.drone.io. The page is titled "Conversion Extension". The left sidebar, under the "Extensions" heading, lists several types of extensions: Overview, Admission, Configuration, **Conversion**, Environment, Registry, Secrets, and Validation. The main content area starts with a brief introduction about conversion extensions, followed by a section on reference extensions (Starlark, Jsonnet, Paths Changed), and finally a section on configuration parameters (DRONE_CONVERT_PLUGIN_ENDPOINT and DRONE_CONVERT_PLUGIN_SECRET).

Conversion Extension

You can use a conversion extension to modify a .drone.yml configuration file before it is parsed and processed by the system. This can be used to automatically add steps or set configuration parameters, or it can be used to convert the configuration file from a non-yaml format to yaml.

Here are some reference extensions:

- [Starlark Language Extension](#)
- [Jsonnet Language Extension](#)
- [Paths Changed Extension](#)

Configuration

You can register a validation extension by providing the following configuration parameters to the Drone server:

- **DRONE_CONVERT_PLUGIN_ENDPOINT**
Provides the endpoint used to make http requests to an extension.
- **DRONE_CONVERT_PLUGIN_SECRET**
Provides the token used to authenticate http requests to the extension. This token is shared between the server and extension.

Paths Changed Extension

github.com/meltwater/drone-convert-pathschanged

The screenshot shows the GitHub repository page for `meltwater / drone-convert-pathschanged`. The repository has 10 stars, 31 forks, and 5 open issues. The code tab is selected, showing a list of commits from the master branch. The commits include:

Author	Commit Message	Date	Commits
jimsheldon	Add parse pipelines include test (#24)	Jun 24	34
	initial commit of alpha code		
	Add parse pipelines include test (#24)	3 months ago	
	initial commit of alpha code	11 months ago	
	move test to its own step (#23)	3 months ago	
	initial commit of alpha code	11 months ago	
	LICENSE is not markdown	11 months ago	
	[SKIP CI] fix token documentation (#21)	5 months ago	
	Add parse pipelines include test (#24)	3 months ago	
	Add tests (#22)	3 months ago	
	Prometheus metrics	8 months ago	

The repository also includes sections for About, Releases (0.1.0 latest), and Packages.

Paths Changed Extension



```
docker run -d \
--publish=3000:3000 \
-e DRONE_SECRET=15b290bb018de407a60b7c66e172dbed \
-e GITHUB_TOKEN=abcdefg... \
--restart=always \
--name=converter meltwater/drone-convert-pathschanged
```

```
docker run -d \
...
-e DRONE_CONVERT_PLUGIN_SECRET=15b290bb018de407a60b7c66e172dbed \
-e DRONE_CONVERT_PLUGIN_ENDPOINT=https://pathsplugin.drone.example.com \
--name=drone drone/drone:1
```



Paths Changed Extension

[REDACTED] May 26th at 3:17 PM

@foundation Is there an official (or Meltwater) plugin to detect changes in directories and run a command if there was changes since the last commit?

[REDACTED] 5 months ago

Hi [@Jim Sheldon](#) More information on the requirement:

So basically I'm trying to convert one of our repos to a monorepo and I want to build only what's necessary for a change, not everything.



Runner Labels

drone-runner-docker

```
docker run -d \
  -v /var/run/docker.sock:/var/run/docker.sock \
  -e DRONE_RPC_PROTO=https \
  -e DRONE_RPC_HOST=drone.example.com \
  -e DRONE_RPC_SECRET=c4489d1a853cb35cd49f25264dce905a \
  -e DRONE_RUNNER_CAPACITY=1 \
  -e DRONE_RUNNER_NAME=${HOSTNAME} \
  -e DRONE_RUNNER_LABELS=datacenter:A \
  -p 3000:3000 \
  --restart always \
  --name runner \
  drone/drone-runner-docker:1
```

Pipeline Routing

.drone.yml



```
---
kind: pipeline
type: docker
name: datacenter/A
```

```
node:
  datacenter: A
```





```
---
kind: pipeline
name: datacenter/A
node:
  datacenter: A
trigger:
  paths:
    include:
      - datacenter/A/**
branch:
  - development
steps:
- name: deploy
  image: alpine
  commands:
    - ./datacenter/A/deploy.sh
---
kind: pipeline
name: datacenter/B
node:
  datacenter: B
trigger:
  paths:
    include:
      - datacenter/B/**
branch:
  - development
steps:
- name: deploy
  image: alpine
  commands:
    - ./datacenter/B/deploy.sh
```





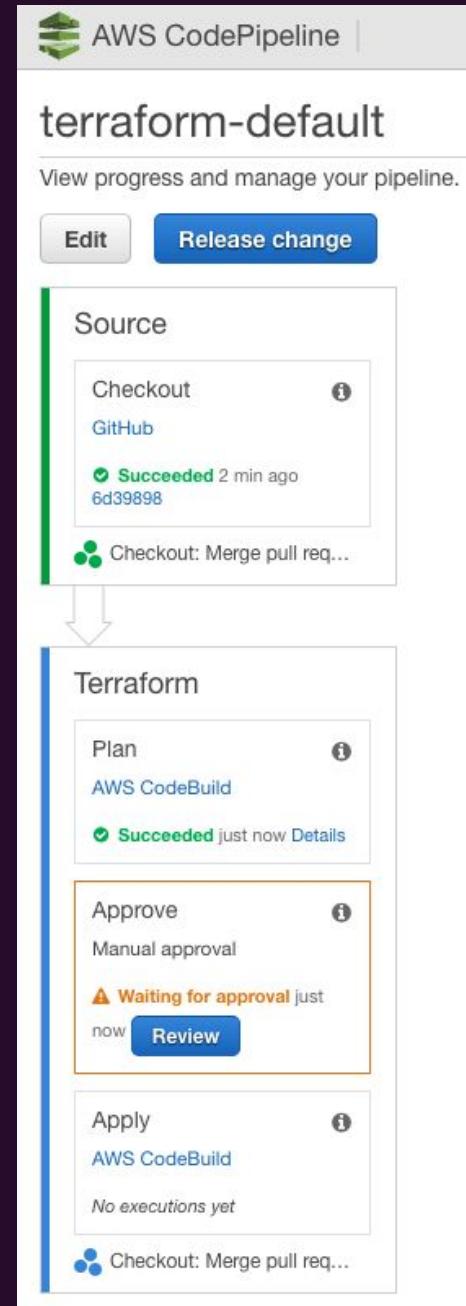
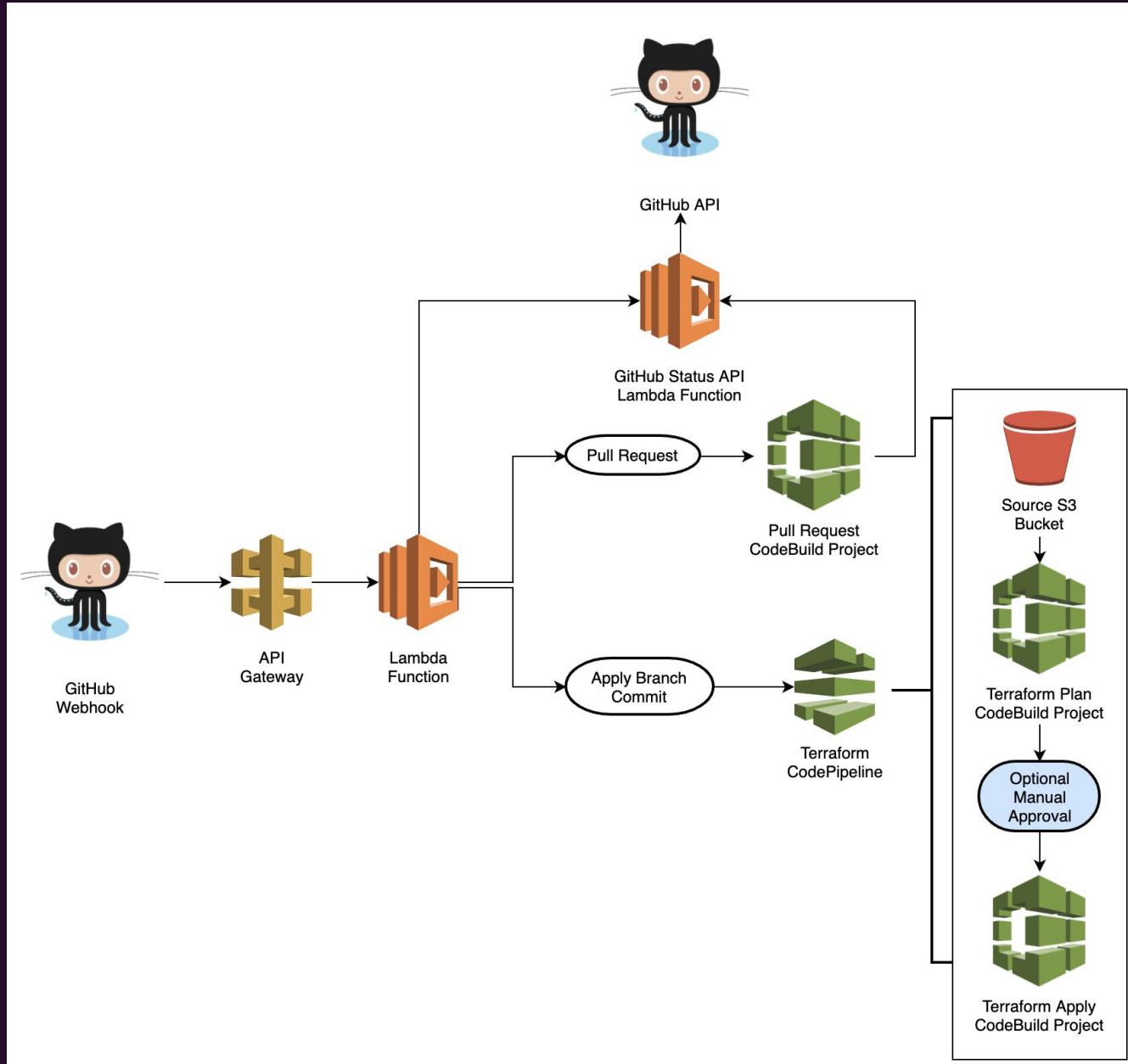
Nadine Shaabana on [Unsplash](#)



Retiring CodePipeline

- Between Jenkins and Drone, we developed a CloudFormation template that would bring up infrastructure to apply Terraform (Meltwaters IaC platform) under CodeBuild in CodePipeline





Retiring CodePipeline: Why?

- Inflexible, new versions of Terraform would break it
- Required a lot of configuration
- Required use of 3 separate billable products (CodeBuild, CodePipeline, and CloudWatch)
- Had to be maintained in every account (we have >200)
- Approvals required access to the AWS account and caused code to become out-of-sync
- Viewing logs required access to the AWS account

Retiring Continues?



As part of Core Platform's Move out of the Data Centre initiative, we successfully released "Goodbye GoCD, hello Drone!". We are now saving an additional £4k of server costs a month by removing 2 cabinets in a Data Centre used for OpenStack which hosted an internal version of GoCD (the teams CI/CD system of choice about 5 years ago). We've therefore close our GoCD instance (as the servers no longer exist) and moved our builds to Drone so not only will be benefit financially, but a chunk of the teams time on managing GoCD and Openstack are removed too. FYI I never had grey hairs before we started using GoCD... .

Basically, thank you for giving us an awesome [Drone.io](#) build environment to use that reduces the effort needed from the Core Platform team! Plus our builds are faster due to the parallelism we get from multi-step builds.

THANK
YOU! 2

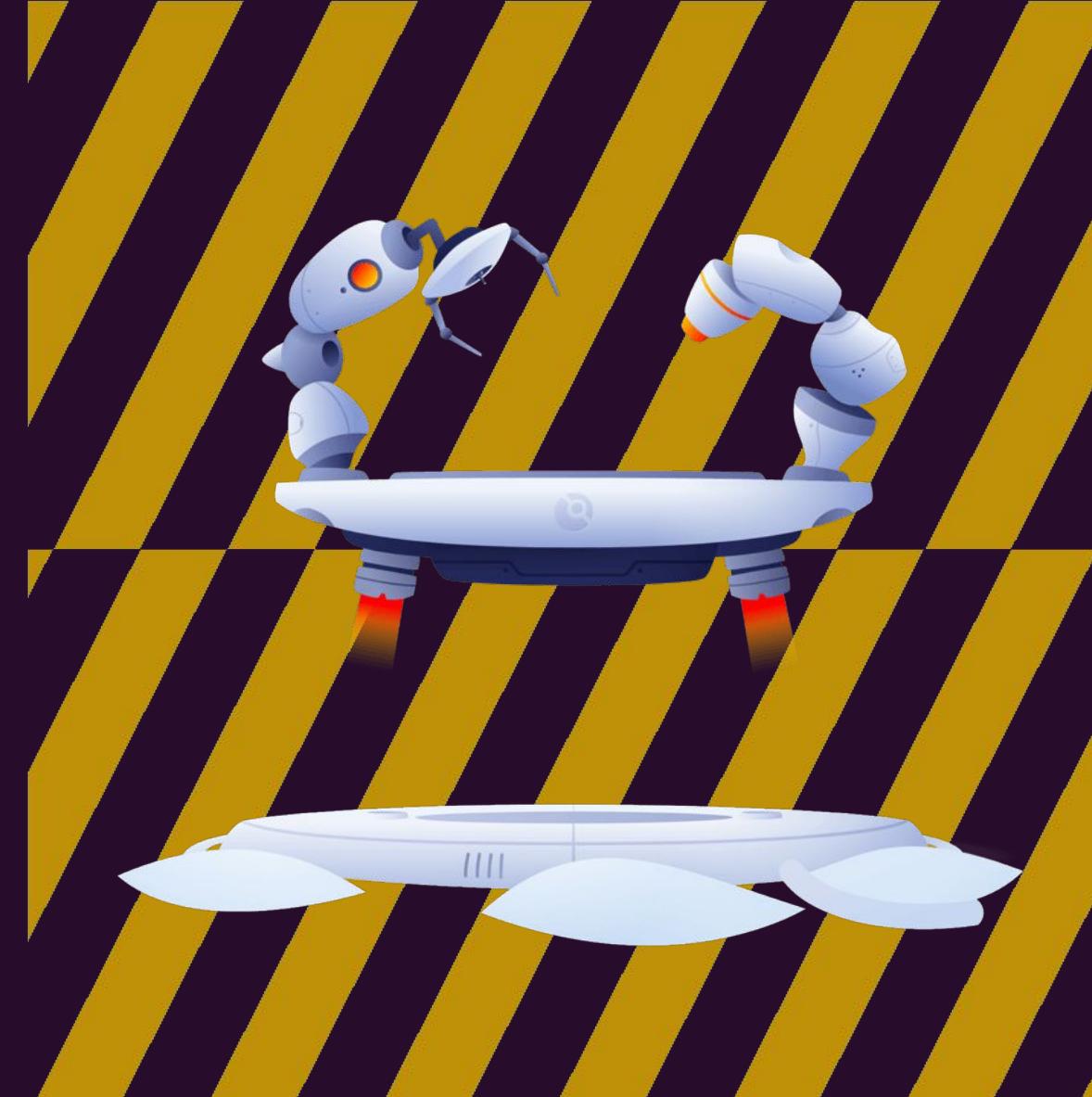


+ 1



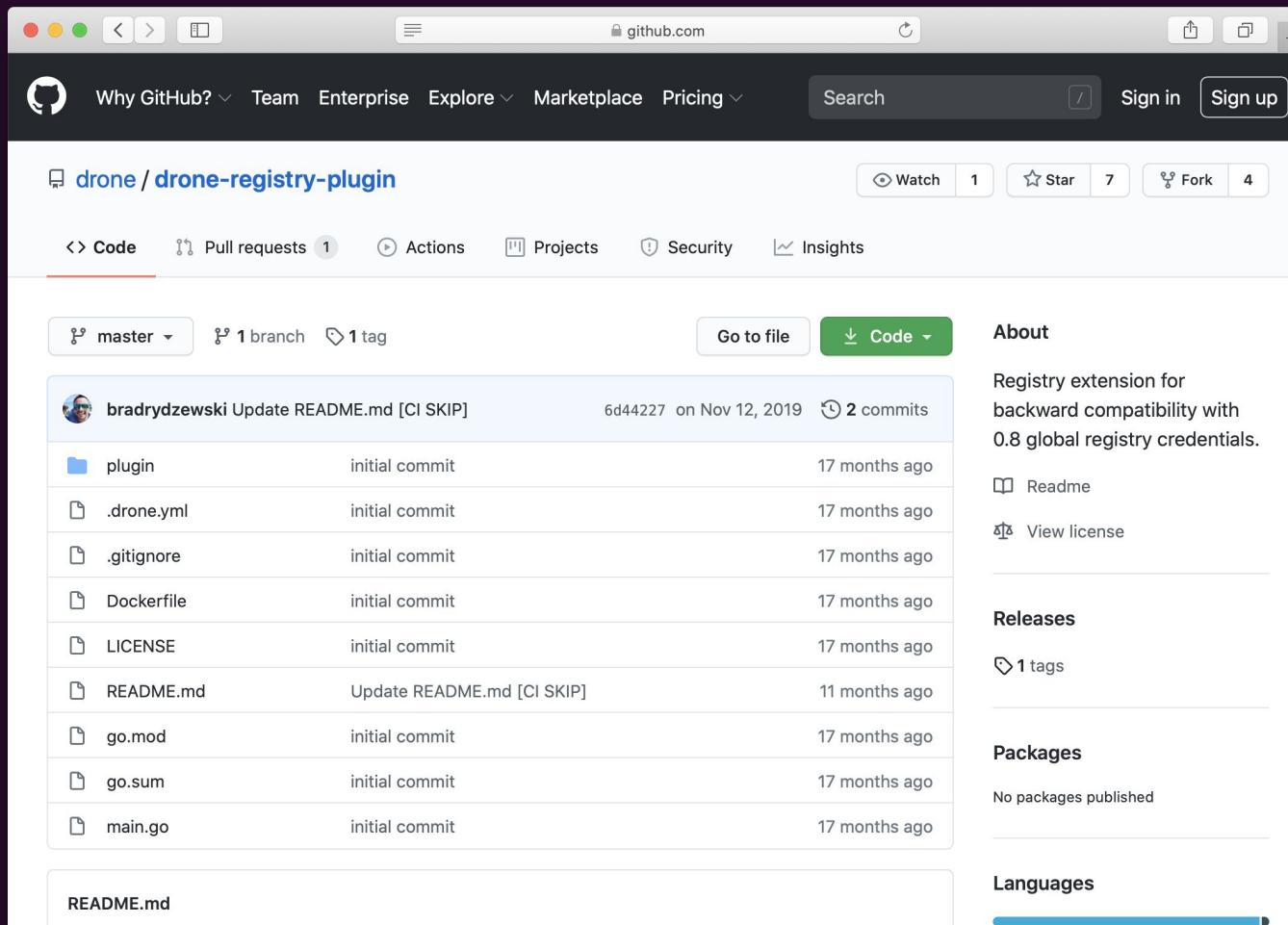
How we got developers attention

- Docker Registry Plugin
- Shared Docker Images Repository
- Drone Cache Plugin
- Drone CLI
- Automation



Running Drone: Registry Plugin

<https://github.com/drone/drone-registry-plugin>



Running Drone: Registry Plugin



```
docker run \
-e DRONE_SECRET=bea26a2221fd8090ea38720fc445eca6 \
-e DRONE_CONFIG_FILE=/path/to/config.yml \
-v /path/to/config.yml:/path/to/config.yml \
--detach \
--restart=always \
-p 3000:3000 \
drone/registry-plugin
```



Running Drone: Registry Plugin Config



config.yml

```
- address: docker.io
  username: octocat
  password: correct-horse-batter-staple
- address: 012345678910.dkr.ecr.us-east-1.amazonaws.com
  aws_access_key_id: a50d28f4dd477bc184fb10b376de753
  aws_secret_access_key: bc5785d3ece6a9cdefa42eb99b58986f9095ff1c
```



Running Drone: Registry Plugin Step



/.drone.yml

```
steps:
  - name: build
    pull: default
    image: 123456789123.dkr.ecr.eu-west-1.amazonaws.com/wrty-oracle-8-jdk-
serverless:1.8.0_192-b12
    commands:
      - mvn -DfailIfNoTests -U clean package -s .drone/settings.xml
environment:
  ARTIFACTORY_PASSWORD:
    from_secret: artifactory_password
  ARTIFACTORY_USERNAME:
    from_secret: artifactory_username
  AWS_REGION: eu-west-1
```



Running Drone: CLI

docs.drone.io/cli/install

The screenshot shows a web browser window displaying the Drone CLI documentation at docs.drone.io/cli/install. The page has a dark theme with a light-colored header bar. The header includes a logo with a stylized 'D' inside a circle, the word 'DOCS', and a search bar with placeholder text 'Search Documentation...'. Below the header is a navigation bar with a 'Home' link. The main content area is titled 'Installation' and contains a brief description: 'The Drone command line tools are used to interact with the Drone from the command line, and provide important utilities for managing users and repository settings.' A horizontal line separates this from the 'Binary Downloads' section. This section is titled 'Binary Downloads' and instructs users to 'Download and install the raw binaries by platform:'. It features a table with five rows, each representing a different platform and its corresponding download links. The table has three columns: 'PLATFORM', 'DOWNLOAD', and 'VERIFY'. The platforms listed are Linux x64, Linux arm64, Linux arm, Windows x64, and Darwin x64. Each row contains a 'tarball' link under 'DOWNLOAD' and a 'checksum' link under 'VERIFY'.

PLATFORM	DOWNLOAD	VERIFY
Linux x64	tarball	checksum
Linux arm64	tarball	checksum
Linux arm	tarball	checksum
Windows x64	tarball	checksum
Darwin x64	tarball	checksum

Running Drone: CLI

```
$ env | grep ^DRONE
DRONE_TOKEN=abcdefg...
DRONE_AUTOSCALER=https://autoscaler.drone.example.com:8080
DRONE_SERVER=https://drone.example.com

$ drone server ls
agent-tE2FRatB
agent-Ujig5NFy

$ drone server info agent-Ujig5NFy
Name: agent-Ujig5NFy
Address: 10.121.1.78
Region: eu-west-1a
Size: t2.xlarge
State: staging

$ drone server info agent-Ujig5NFy
Name: agent-Ujig5NFy
Address: 10.121.1.78
Region: eu-west-1a
Size: t2.xlarge
State: running
```

Running Drone: CLI (Local Exec)



~/gitHub/foundation-test-nonsense/

```
[✓]~/gitHub/foundation-test-nonsense(master)$ drone exec \
--event pull_request \
--branch master \
--repo meltwater/foundation-test-nonsense
```

```
[slack-ateam:0] + echo hello world
[slack-ateam:1] hello world
```



Third-party service outage notifications

7:29 AM Opsgenie APP

#9333: Heartbeat [dockerReg-Quay] is expired

Foundation service checks are observing failures on ability to pull images from quay.io. In this case we have not been able to pull a test layer for the last 15 minutes (meaning the check has failed more than twice). Images and plugins hosted here may not be available to services like [drone](#), kubernetes, or mesos during this time. Please check <https://status.quay.io/> and alert others of this state as you see fit. This room will show an update once successfully testing this service again.

Priority

P3

6:04 AM Opsgenie APP

#35: Heartbeat [githubWebhooks] is expired

Foundation service checks are observing failures on webhook reception from GitHub commits. In this case, we have failed to receive two webhooks on a test repository and are potentially 15minutes behind on build triggering. Till service is restored, notifications via webhook to lambda, circle-ci, travis, and [drone](#) will all be affected. Please check <https://www.githubstatus.com/> and alert others of this state as you see fit. This room will show an update once successfully testing this service again.

Priority

P3

Lessons Learned

- Don't try and force a toolset, if it's the right one developers will move on their own
- Combination of custom docker images and flexible yaml configuration was key
- Flexibility to customize and add to any solution should be a baseline requirement (an UNopinionated solution)

Meltwater Engineering Blog

underthehood.meltwater.com

The screenshot shows a desktop browser window displaying the blog's homepage. The header features the site's name and a subtitle: "under the hood < The official meltwater engineering blog". Below the header is a navigation bar with links for BLOG, ABOUT, JOBS, OPEN SOURCE, and ARCHIVES. The main content area displays a post from July 10, 2020, titled "Improving Record Linking for our Knowledge Graph (Part 2)". The post includes a brief summary, a detailed description of the record linking process, and a "Read on >" button. Below this is another post from June 29, 2020, titled "The Record Linking Pipeline for our Knowledge Graph (Part 1)". The sidebar contains sections for "About Us", "Recent Posts", and "GitHub Repos".

About Us:
We are the engineers of [Meltwater](#). Find our open source projects at [GitHub](#). Here we write about the things we do.

Recent Posts:

- [Improving Record Linking for our Knowledge Graph \(Part 2\)](#)
- [The Record Linking Pipeline for our Knowledge Graph \(Part 1\)](#)
- [Let's Talk about Feelings!](#)
- [Tech Talk: Scalability Testing of a Production Kubernetes Cluster](#)
- [The Journey To Front-End Performance — Assessing Current Performance](#)

GitHub Repos:

- addict**
User management lib for Phoenix Framework
- AGDISTIS**
AGDISTIS - Agnostic Named Entity Disambiguation

Don't Forget to Promote

@UnscriptedConf

#UnscriptedConf

#UnscriptedConf2020

... Tell your Friends!

Thanks!

Get Started with Drone!

/

Contact:
Brian Burnett
Jim Sheldon



@hikerspath
@jimsheldon



@hikerspath
@bitberk



@UnscriptedConf

