

Extending Drone:

A Continuous Integration Journey

with Jim Sheldon





Jim Sheldon 🙌

@bitberk

I live in New Hampshire, USA

Principal Software Engineer @ Meltwater

- “Foundation Mission”
- meltwater.com

Drone in production for 3 years!

Launched in June 2018

Meltwater Drone pipelines per day

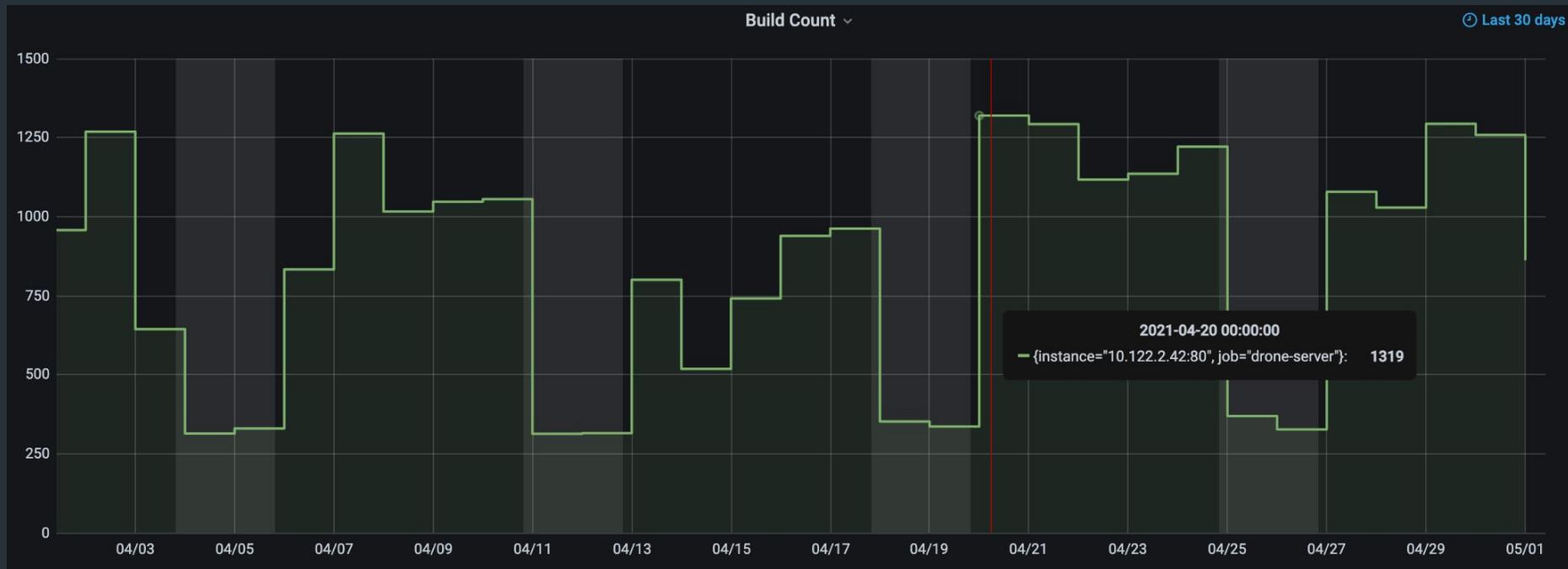
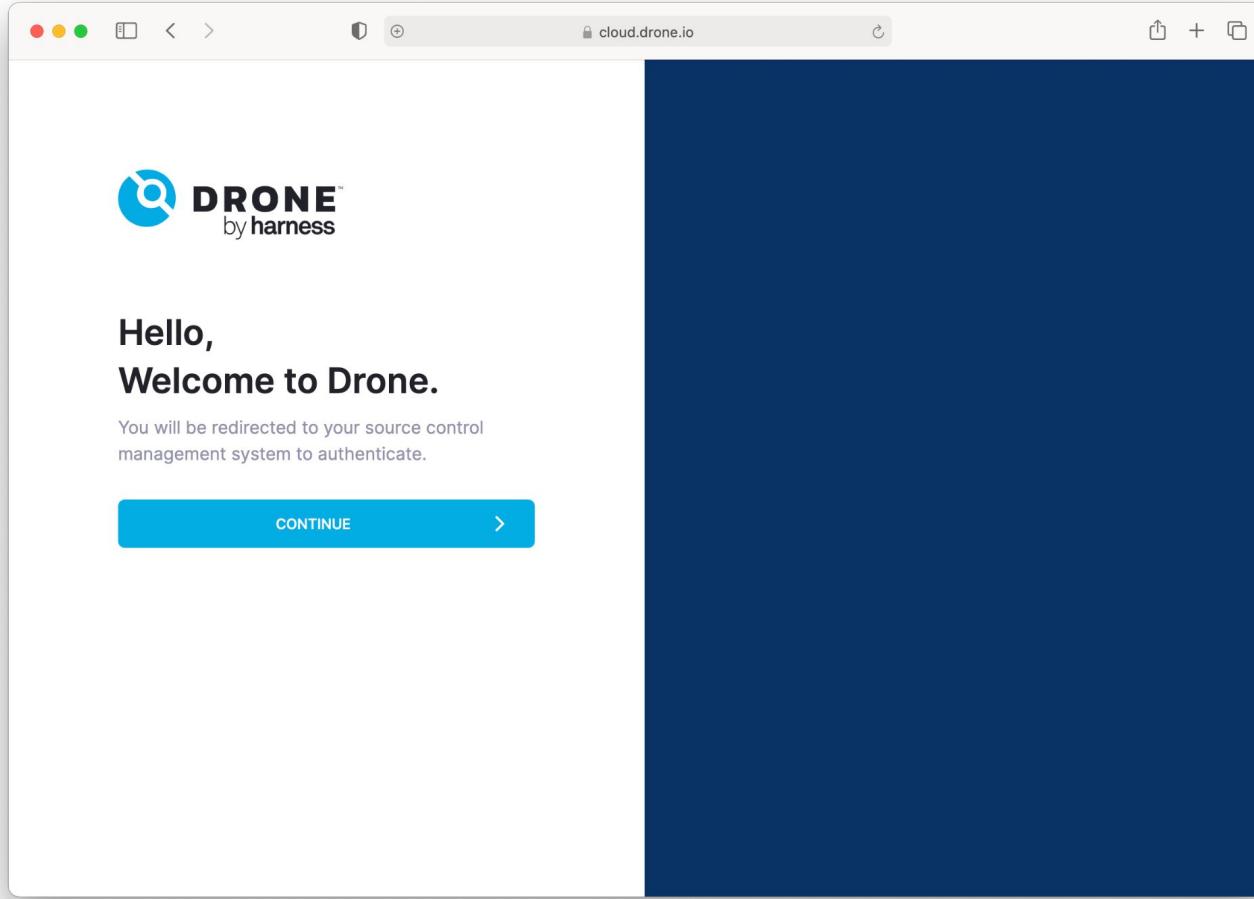




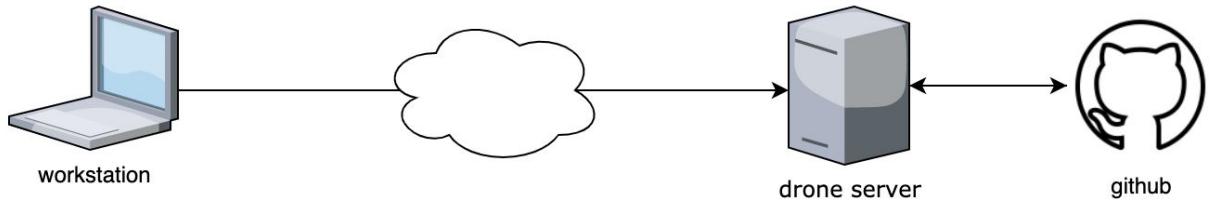
Photo by [Dino Reichmuth](#) on [Unsplash](#)



<https://cloud.drone.io>



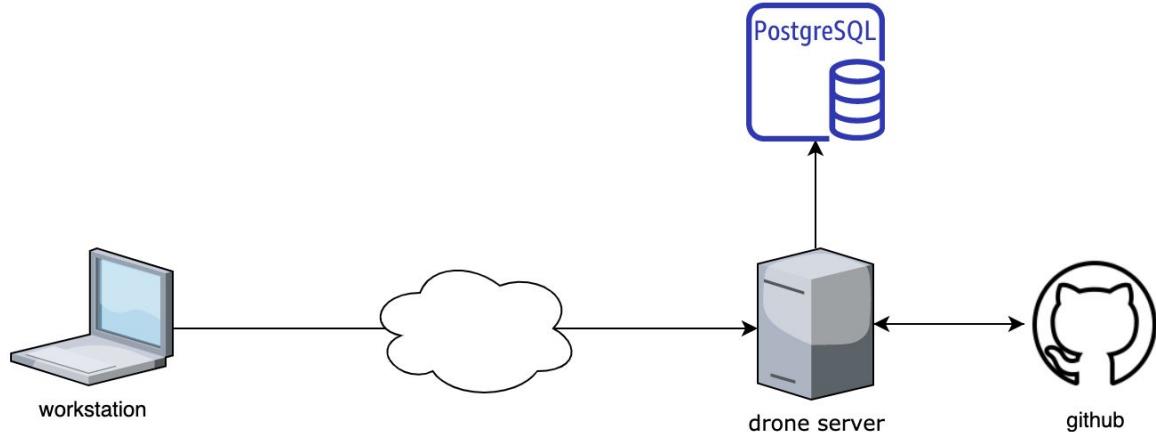
server



```
docker run \
--volume=/var/lib/drone:/data \
--env=DRONE_GITHUB_CLIENT_ID=${DRONE_GITHUB_CLIENT_ID} \
--env=DRONE_GITHUB_CLIENT_SECRET=${DRONE_GITHUB_CLIENT_SECRET} \
--env=DRONE_RPC_SECRET=${DRONE_RPC_SECRET} \
--env=DRONE_SERVER_HOST=${DRONE_SERVER_HOST} \
--env=DRONE_SERVER_PROTO=${DRONE_SERVER_PROTO} \
--publish=80:80 \
--publish=443:443 \
--restart=always \
--detach=true \
--name=drone \
drone/drone:1
```

<https://docs.drone.io/server/provider/github/>

server + database

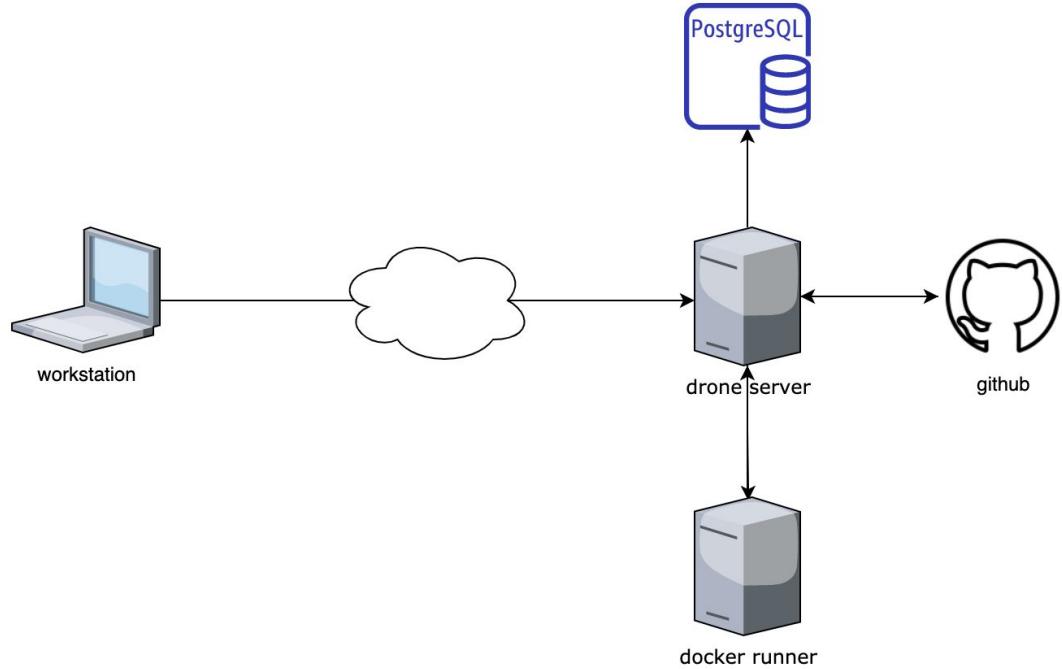


```
docker run \
--env=DRONE_GITHUB_CLIENT_ID=${DRONE_GITHUB_CLIENT_ID} \
--env=DRONE_GITHUB_CLIENT_SECRET=${DRONE_GITHUB_CLIENT_SECRET} \
--env=DRONE_RPC_SECRET=${DRONE_RPC_SECRET} \
--env=DRONE_SERVER_HOST=${DRONE_SERVER_HOST} \
--env=DRONE_SERVER_PROTO=${DRONE_SERVER_PROTO} \
--env=DRONE_DATABASE_DRIVER=postgres \
--env=DRONE_DATABASE_DATASOURCE=postgres://root:password@1.2.3.4:5432/postgres?sslmode=disable \
--publish=80:80 \
--publish=443:443 \
--restart=always \
--detach=true \
--name=drone \
drone/drone:1
```

<https://docs.drone.io/server/storage/database/>



server +
database +
docker runner

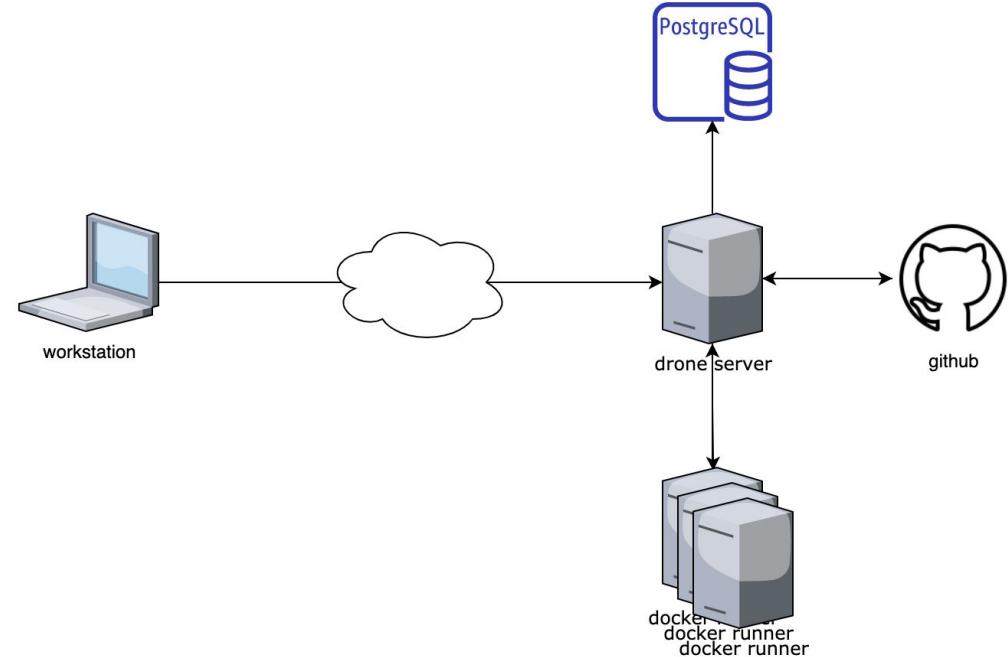


```
docker run -d \
    -v /var/run/docker.sock:/var/run/docker.sock \
    -e DRONE_RPC_PROTO=https \
    -e DRONE_RPC_HOST=drone.company.com \
    -e DRONE_RPC_SECRET=super-duper-secret \
    -e DRONE_RUNNER_CAPACITY=1 \
    -e DRONE_RUNNER_NAME=${HOSTNAME} \
    -p 3000:3000 \
    --restart always \
    --name runner \
    drone/drone-runner-docker:1
```

<https://docs.drone.io/runner/docker/installation/linux/>



server + database + docker runners



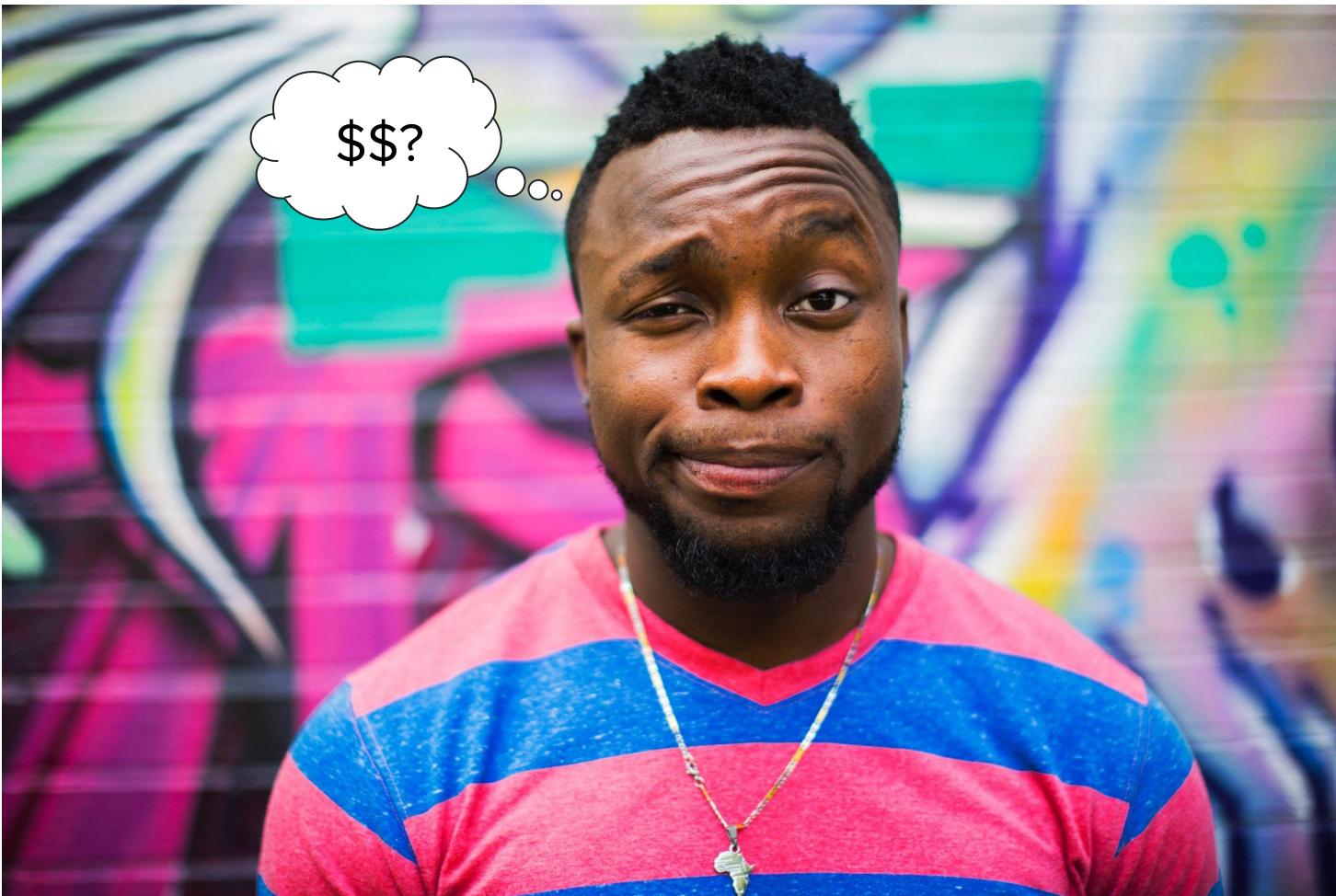
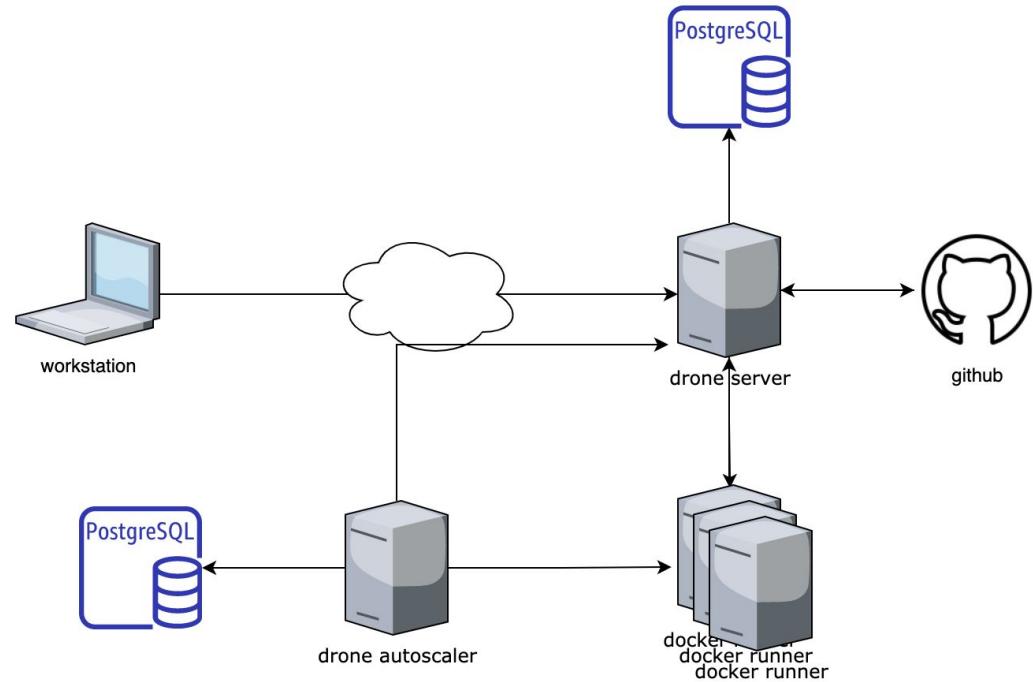
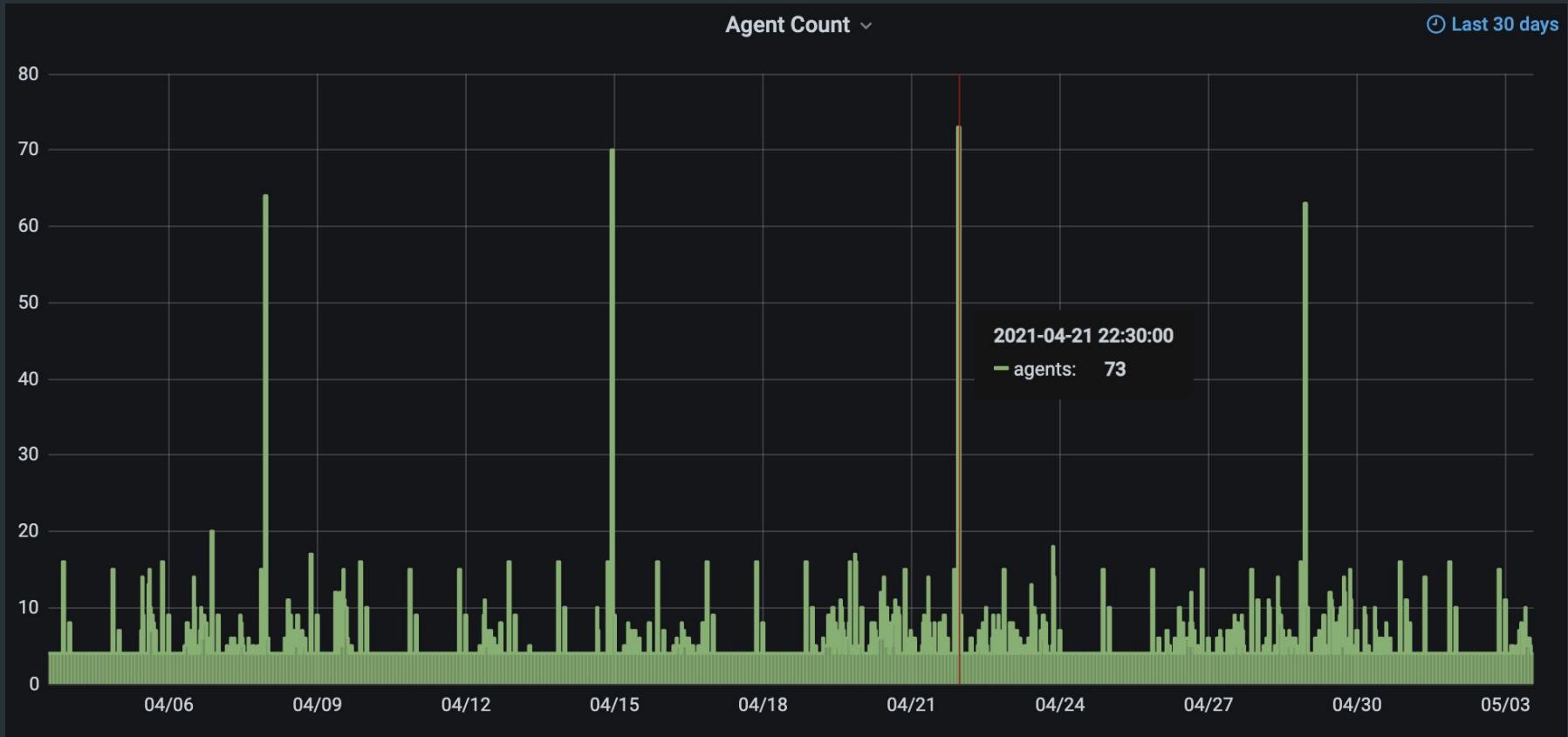


Photo by [Ayo Ogunseinde](#) on [Unsplash](#)

server +
database +
docker runners +
autoscaler



Meltwater docker runner count over time



```
docker run -d \
-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_DATABASE_DRIVER=${DRONE_DATABASE_DRIVER} \
-e DRONE_DATABASE_DATASOURCE=${DRONE_DATABASE_DATASOURCE} \
-e DRONE_AMAZON_INSTANCE=${DRONE_AMAZON_INSTANCE} \
-e DRONE_AMAZON_REGION=${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
```

<https://autoscale.drone.io/install/amazon/>

```
#cloud-config
apt_reboot_if_required: false
package_update: false
package_upgrade: false
apt:
  sources:
    docker.list:
      source: deb [arch=amd64] https://download.docker.com/linux/ubuntu $RELEASE stable
      keyid: 0EBFCD88
  packages:
    - docker-ce
  write_files:
    - path: /etc/systemd/system/docker.service.d/override.conf
      content: |
        [Service]
        ExecStart=
        ExecStart=/usr/bin/dockerd
    - path: /etc/default/docker
      content: |
        DOCKER_OPTS=""
    - path: /etc/docker/daemon.json
      content: |
        {
          "hosts": [ "0.0.0.0:2376", "unix:///var/run/docker.sock" ],
          "tls": true,
          "tlsv1": true,
          "tlscacert": "/etc/docker/ca.pem",
          "tlscert": "/etc/docker/server-cert.pem",
          "tlskey": "/etc/docker/server-key.pem"
        }
    - path: /etc/docker/ca.pem
      encoding: b64
      content: {{ .CACert | base64 }}
    - path: /etc/docker/server-cert.pem
      encoding: b64
      content: {{ .TLSCert | base64 }}
    - path: /etc/docker/server-key.pem
      encoding: b64
      content: {{ .TLSKey | base64 }}
  runcmd:
    - [ systemctl, daemon-reload ]
    - [ systemctl, restart, docker ]
```

```
docker run -d \
-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_INSTANCE=${DRONE_AMAZON_INSTANCE} \
-e DRONE_AMAZON_REGION=${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} \
-e DRONE_AMAZON_USERDATA_FILE=/path/in/container/init.yml \
--volume=/path/on/host/init.yml:/path/in/container/init.yml \
-p 8080:8080 \
--restart=always \
--name=autoscaler \
drone/autoscaler
```

<https://autoscale.drone.io/configure/cloud-init/>



```
steps:  
- name: Run tests  
image: node:10  
commands:  
- npm install -g gulp  
- npm install  
- npm test
```

```
steps:  
- name: Run tests  
image: node:10  
commands:  
- npm install -g gulp  
- npm install  
- npm test
```

```
steps:  
- name: Run tests  
image: node:10  
commands:  
- npm install -g gulp  
- npm install  
- npm test
```

Photo by [Helena Lopes](#) on [Unsplash](#)

The screenshot shows a web browser window with the URL docs.drone.io in the address bar. The page is titled "Plugins" and has a sub-section "Overview". The main content area contains an "Overview" section with a brief description of what plugins are and how they can be used. Below this is a code block showing an example pipeline configuration using the Docker and Slack plugins.

Plugins

[Overview](#)

[Explore](#)

EXAMPLES

[Example Go Plugin](#)

[Example Bash Plugin](#)

POPULAR

[Docker](#)

Overview

Plugins are Docker containers that perform pre-defined tasks and are configured as steps in your pipeline. Plugins can be used to deploy code, publish artifacts, send notification, and more.

Example pipeline using the Docker and Slack plugins:

```
1 kind: pipeline
2 type: docker
3 name: default
4
5 steps:
6 - name: build
7   image: golang
8   commands:
9     - go get
10    - go test
11    - go build
12
13 - name: publish
14   image: plugins/docker
15   settings:
16     username: kevinbacon
17     password: pa55word
18   repo: foo/bar
19   tags:
20     - 1.0.0
21     - 1.0
22
23 - name: notify
24   image: plugins/slack
25   settings:
```

<https://docs.drone.io/plugins/overview/>

Not Secure — plugins.drone.io

Plugins Documentation Support

Plugins

Browse our registry of community plugins to customize your continuous delivery pipeline.

AWS CloudFormation

by robertstettner

DEPLOY AMAZON AWS CLOUDFORMATION



AWS ECR

by drone-plugins

PUBLISH AMAZON AWS ECR DOCKER



AWS ECS

by josmo

DEPLOY AMAZON AWS ECS DOCKER



AWS ECS Deploy

by devops-israel

DEPLOY AMAZON AWS ECS DOCKER



<http://plugins.drone.io>

```
kind: pipeline
type: docker
name: default

steps:
- name: docker
  image: plugins/docker
  settings:
    repo: foo/bar
    username:
      from_secret: docker_username
    password:
      from_secret: docker_password
```

<http://plugins.drone.io/drone-plugins/drone-docker/>

github.com

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search Sign in Sign up

meltwater / drone-cache

Code Issues 24 Pull requests 2 Discussions Actions Projects Security 1 Insights

master 2 branches 15 tags Go to file Code

hhamalai and kakkoyun Fix assume role parameter passing (#... 40d8f10 on Feb 2 226 commits

.bingo	Use bingo for tool dependencies (#126)	11 months ago
.github	Fix github actions	11 months ago
archive	Use bingo for tool dependencies (#126)	11 months ago
cache	Use bingo for tool dependencies (#126)	11 months ago
docs	Merge release v1.1 (#125)	11 months ago
images	Add slack message image	2 years ago
internal	Use bingo for tool dependencies (#126)	11 months ago
key	Use bingo for tool dependencies (#126)	11 months ago
storage	Fix assume role parameter passing (#151)	3 months ago
test	Use bingo for tool dependencies (#126)	11 months ago
.dockerignore	Introduce GoReleaser action (#110)	13 months ago
.drone.yml	Add optional param for AWS IAM role assumption (#143)	6 months ago

About

A Drone plugin for caching current workspace files between builds to reduce your build times

plugin go golang aws
drone pipeline azure
cache s3
google-cloud-storage sftp
minio ceph drone-plugin
volume cicd hacktoberfest
drone-ci azure-blob-storage
droneio

Readme Apache-2.0 License

Releases 15

v1.1.0 Latest on Jun 11, 2020

<https://github.com/meltwater/drone-cache>

server +
database +
docker runners +
autoscaler

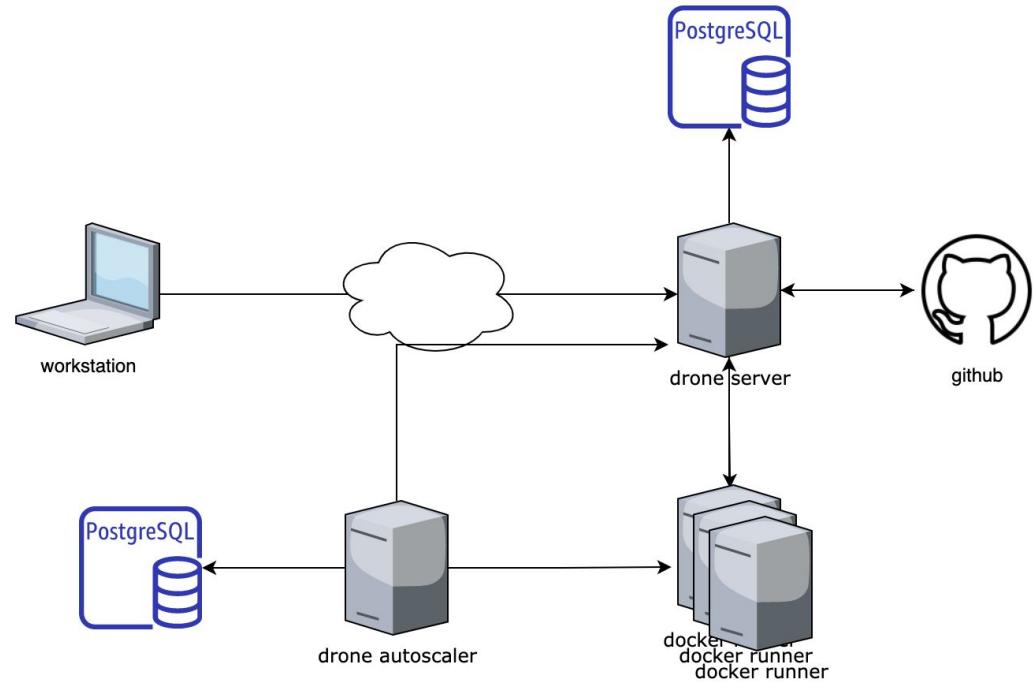






Photo by [bruce mars](#) on [Unsplash](#)



Photo by [Ameen Fahmy](#) on [Unsplash](#)

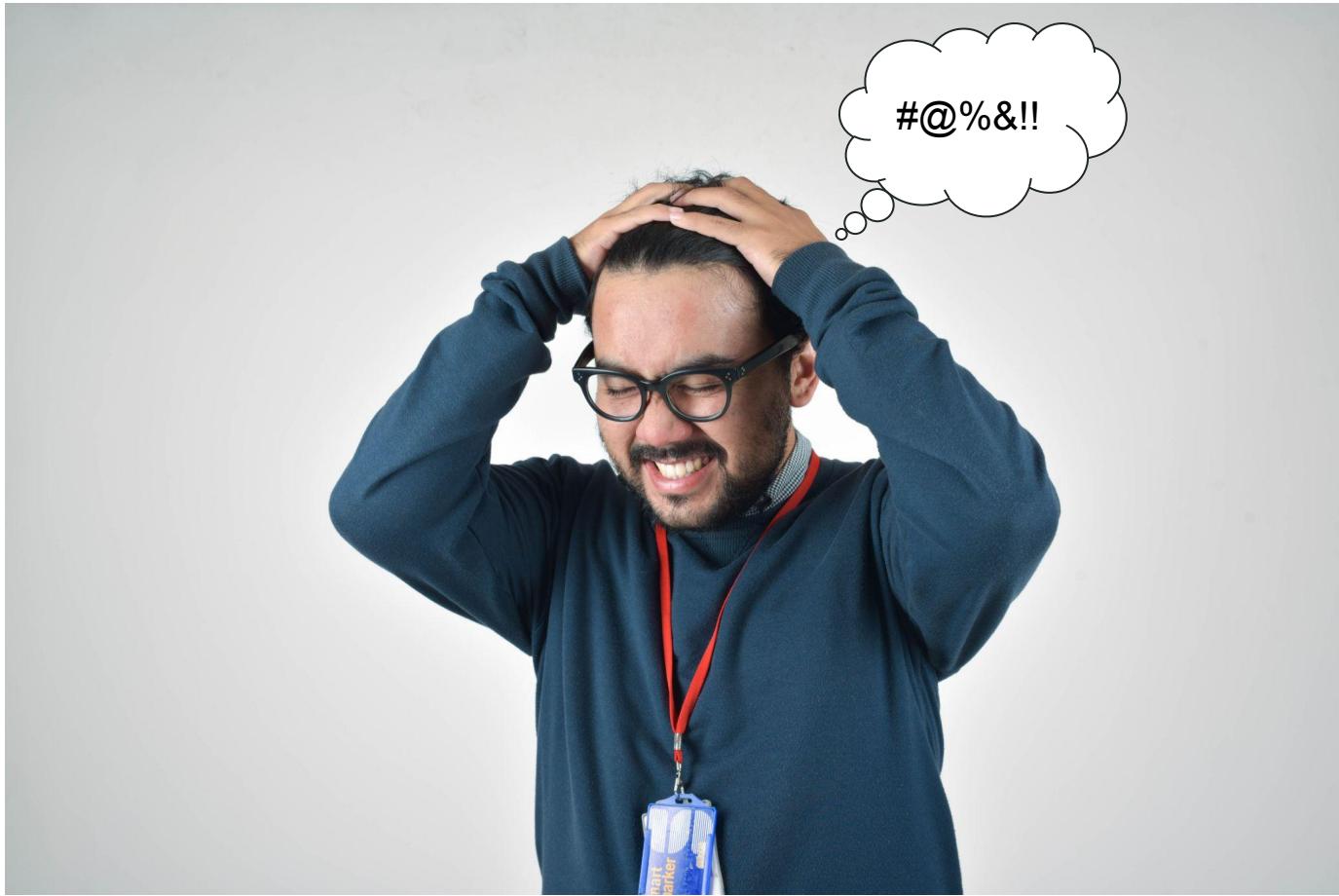
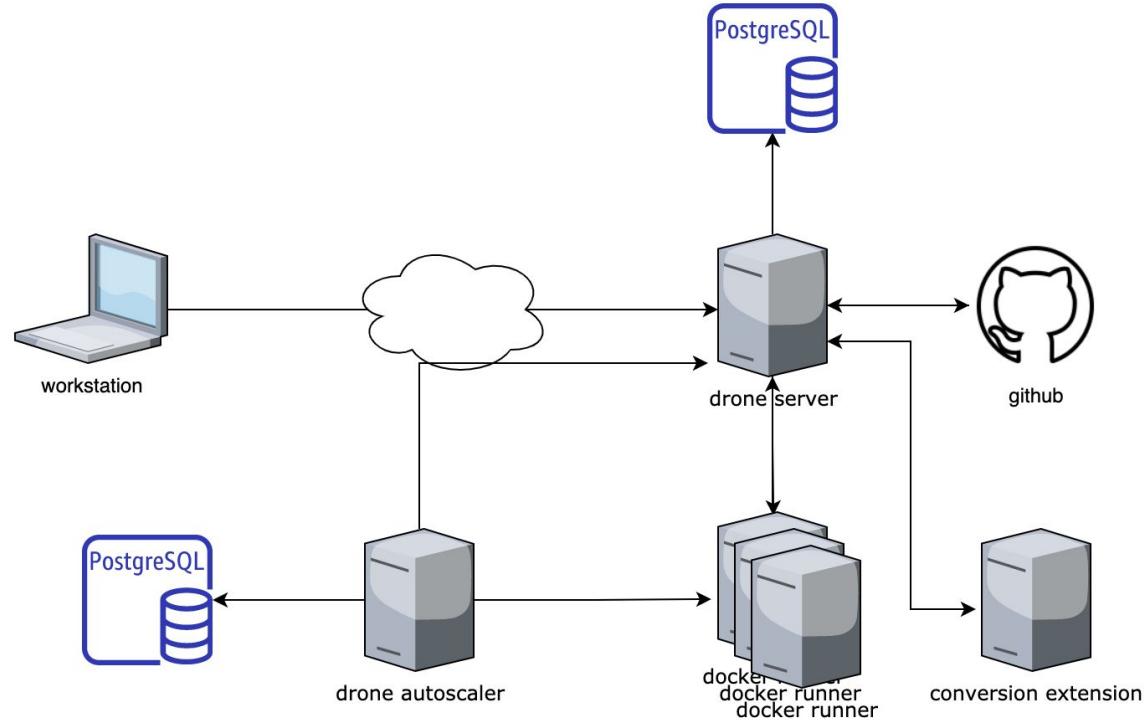


Photo by [ahmad gunnaivi](#) on [Unsplash](#)

server +
database +
docker runners +
autoscaler +
conversion extension



```
docker run -d \
--publish=3000:3000 \
--env=DRONE_DEBUG=true \
--env=DRONE_SECRET={{DRONE_SECRET}} \
--restart=always \
--name=converter {{DockerRepository}}
```

<https://github.com/drone/boilr-convert/tree/master/template>

```
docker run \
--env=DRONE_GITHUB_CLIENT_ID={{DRONE_GITHUB_CLIENT_ID}} \
--env=DRONE_GITHUB_CLIENT_SECRET={{DRONE_GITHUB_CLIENT_SECRET}} \
--env=DRONE_RPC_SECRET={{DRONE_RPC_SECRET}} \
--env=DRONE_SERVER_HOST={{DRONE_SERVER_HOST}} \
--env=DRONE_SERVER_PROTO={{DRONE_SERVER_PROTO}} \
--env=DRONE_DATABASE_DRIVER=postgres
--env=DRONE_DATABASE_DATASOURCE=postgres://root:password@1.2.3.4:5432/postgres?sslmode=disable
--env=DRONE_AGENTS_ENABLED=true \
--env=DRONE_CONVERT_PLUGIN_ENDPOINT=${EXTENSION_HOST} \
--env=DRONE_CONVERT_PLUGIN_SECRET=${EXTENSION_RPC_SECRET} \
--publish=80:80 \
--publish=443:443 \
--restart=always \
--detach=true \
--name=drone \
drone/drone:1
```

<https://docs.drone.io/extensions/conversion/>

github.com

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search Sign in Sign up

meltwater / drone-convert-pathschanged

Code Issues 5 Pull requests 2 Actions Security Insights

master 1 branch 9 tags Go to file Code

jimsheldon adding 0.5.0 to changelog [skip ci] f44e000 on Feb 12 56 commits

docker build images from 'scratch' (#47) 3 months ago

plugin handle commits with '--allow-empty' (#46) 3 months ago

providers use drone/go-scm in github provider (#36) 6 months ago

scripts adding arm64 support (#42) 4 months ago

.drone.yml adding docker manifest (#43) 4 months ago

.gitignore initial commit of alpha code 2 years ago

CHANGELOG.md adding 0.5.0 to changelog [skip ci] 3 months ago

LICENSE LICENSE is not markdown 2 years ago

README.md [SKIP CI] implicit include example 5 months ago

go.mod use drone/go-scm in github provider (#36) 6 months ago

go.sum handle commits with '--allow-empty' (#46) 3 months ago

main.go use drone/go-scm in github provider (#36) 6 months ago

About

Drone conversion extension to include/exclude pipelines and pipeline steps based on paths changed

golang drone drone-plugin conversion-extension

Readme Apache-2.0 License

Releases 9

0.5.0 Latest on Feb 12 + 8 releases

Contributors 4

<https://github.com/meltwater/drone-convert-pathschanged>



Photo by [Fab Lentz](#) on [Unsplash](#)

**default: Error response from daemon: toomanyrequests: You have reached your pull rate limit.
You may increase the limit by authenticating and upgrading:
<https://www.docker.com/increase-rate-limit>**



Photo by [June O](#) on [Unsplash](#)

The screenshot shows a web browser window with the Docker website open. The URL in the address bar is docker.com. The page title is "Understanding Docker Hub Rate Limiting". The navigation menu includes links for Products, Developers, Pricing, Blog, and About us. There is also a "Get Started" button. The main content area contains sections for "Learn More", "Related Resources", and a summary of the rate limit changes.

Understanding Docker Hub Rate Limiting

Learn More

On November 20, 2020, rate limits anonymous and free authenticated use of Docker Hub went into effect. Anonymous and Free Docker Hub users are limited to 100 and 200 container image pull requests per six hours. You can read [here](#) for more detailed information.

If you are affected by these changes you will receive this error message:

ERROR: toomanyrequests: Too Many Requests.

OR

You have reached your pull rate limit. You may increase the limit by authenticating and upgrading: <https://www.docker.com/increase-rate-limits>. You must authenticate your pull requests.

To increase your pull rate limits you can upgrade your account to a Docker Pro or Team subscription.

The rate limits of 100 container image requests per six hours for anonymous usage, and 200 container image requests per six hours for free Docker accounts are now in effect.

Related Resources

- Rate Limiting Documentation
- Resource Consumption Update FAQ
- Blog post on Resource Consumption and Image Retention Policy Updates
- Docker Subscription Pricing Page
- Pricing FAQ
- Guidance for Setting Up a Mirror
- Understanding Inner Loop Development and Pull Rates
- Docker Partner Information Request for Rate Limiting Exclusions

<https://www.docker.com/increase-rate-limits>

```
#cloud-config
apt_reboot_if_required: false
package_update: false
package_upgrade: false
apt:
  sources:
    docker.list:
      source: deb [ arch=amd64 ] https://download.docker.com/linux/ubuntu $RELEASE stable
      keyid: 0EBFCD88
  packages:
    - docker-ce
  write_files:
    - path: /root/.docker/config.json
      owner: root:root
      permissions: '0600'
      content: |
        {
          "auths": {
            "https://index.docker.io/v1/": {
              "auth": "{{docker_auth_token}}"
            }
          }
        }
    - path: /etc/systemd/system/docker.service.d/override.conf
      content: |
        [Service]
        ExecStart=
        ExecStart=/usr/bin/dockerd
    - path: /etc/default/docker
      content: |
        DOCKER_OPTS=""
    - path: /etc/docker/daemon.json
      content: |
        {
          "hosts": [ "0.0.0.0:2376", "unix:///var/run/docker.sock" ],
          "tls": true,
          "tlsverify": true,
          "tlscacert": "/etc/docker/ca.pem",
          "tlscert": "/etc/docker/server-cert.pem",
          "tlskey": "/etc/docker/server-key.pem"
        }
    - path: /etc/docker/ca.pem
      encoding: b64
      content: {{ .CACert | base64 }}
    - path: /etc/docker/server-cert.pem
      encoding: b64
      content: {{ .TLSCert | base64 }}
    - path: /etc/docker/server-key.pem
      encoding: b64
      content: {{ .TLSKey | base64 }}
  runcmd:
    - [ systemctl, daemon-reload ]
    - [ systemctl, restart, docker ]
```

```
docker run -d \
-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_AGENT_VOLUMES=/root/.docker/config.json:/root/.docker/config.json \
-e DRONE_AGENT_ENVIRON=DRONE_DOCKER_CONFIG=/root/.docker/config.json \
-e DRONE_AMAZON_INSTANCE=${DRONE_AMAZON_INSTANCE} \
-e DRONE_AMAZON_REGION=${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} \
-e DRONE_AMAZON_USERDATA_FILE=/path/in/container/init.yml \
--volume=/path/on/host/init.yml:/path/in/container/init.yml \
-p 8080:8080 \
--restart=always \
--name=autoscaler \
drone/autoscaler
```

<https://autoscale.drone.io/configure/cloud-init/>

The screenshot shows a web browser window with the URL discourse.drone.io in the address bar. The page title is "How to prevent DockerHub pull rate limit errors". The navigation bar includes links for "Documentation", "Get Started", and "Gitter". The header features the "DRONE by harness" logo, "Sign Up", "Log In", a search icon, and a menu icon.

How to prevent DockerHub pull rate limit errors

bradrydzewski · Nov 2020 · 3 replies · 1 / 2 · Nov 2020

DockerHub recently implemented rate limits that may cause pipeline to fail when Docker attempts to pull images. This failure can be identified by the following error message:

Error response from daemon: toomanyrequests: You have reached your pull rate limit. You may increase the limit by authenticating and upgrading: <https://www.docker.com/increase-rate-limit>

Pulling Pipeline Images

The docker daemon pulls images defined in your pipeline with no authentication, by default. We recommend creating a service account at DockerHub and purchasing a plan, which costs \$5 per month. Then provide the service account credentials to your runner

First, create the credentials json file and save to your machine
The Kubernetes team provides a [nice tutorial](#) for creating this file.

Next, mount this file as a volume into your runner container. Note that this is the path to the file that was mounted inside the container, not the path on the host machine.

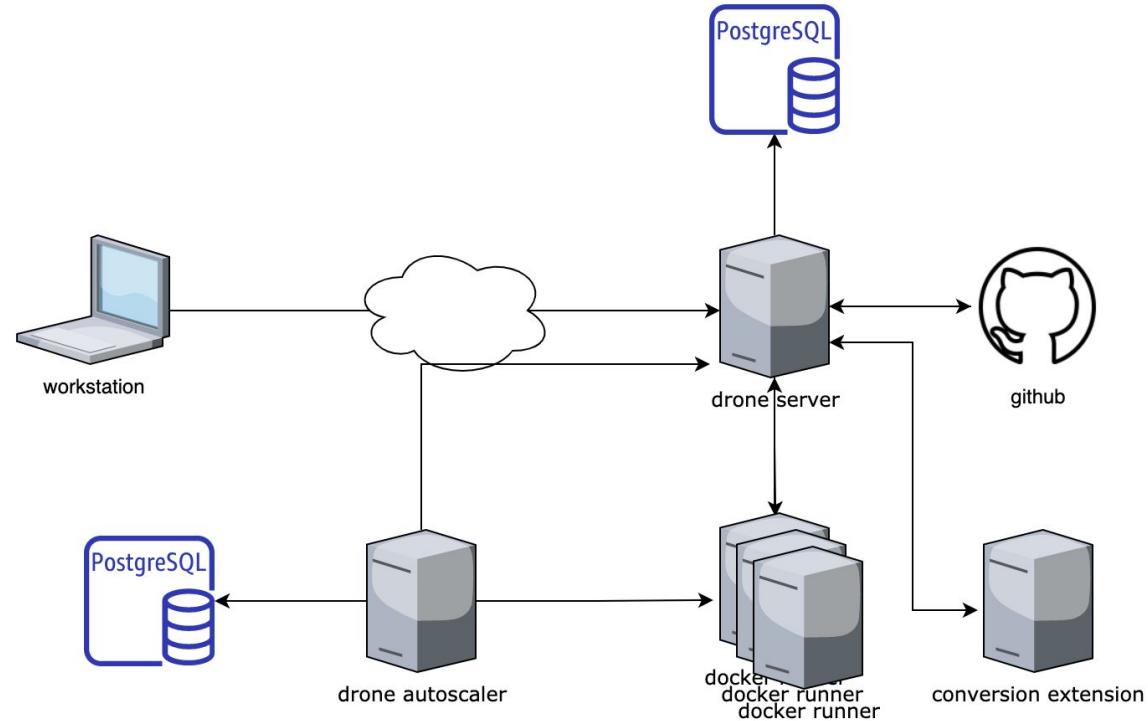
```
-v /root/.docker/config.json:/root/.docker/config.json
```

Last, provide the runner the path to the credentials json file. Note that this is the path to the file that was mounted inside the container, not the path on the host machine.

```
DRONE_DOCKER_CONFIG=/root/.docker/config.json
```

<https://discourse.drone.io/t/how-to-prevent-dockerhub-pull-rate-limit-errors/8324>

server +
database +
docker runners +
autoscaler +
conversion extension



The End



Thank you!

Meltwater Engineering Blog:
underthehood.meltwater.com

We're hiring!
careers.meltwater.com

Questions?