
Kubernetes Cron Jobs

Chris Short

SJ Technologies

Detroit Kubernetes, Docker & all related things

whoami



DevOps'ish



CLOUD NATIVE
COMPUTING FOUNDATION
AMBASSADOR

open
source
.com

@ChrisShort

devopsish.com

Here's where every speaker tries to establish why they're standing in front of you.

Senior DevOps Advocate with SJ Technologies
We help people embrace DevOps practices like
Lean, Agile, and Infrastructure as Code

I'm a Cloud Native Computing Foundation
Ambassador

I'm a community moderator and DevOps lead at
opensource.com

I run a DevOps and Cloud Native newsletter called
DevOps'ish

I'm a frequent DevOpsDays speaker

And I'm a contributor to a book on IT Culture Change



Getting Started

Blah, blah, blah then hackery

- **Kubernetes Concepts**
Kubernetes, Jobs, and Cron Jobs
- **Use Case**
How are Kubernetes Cron Jobs useful
- **The Docker Bits**
A container to do the work
- **The Kubernetes Bits**
Use the yaml, Luke. —Yoda

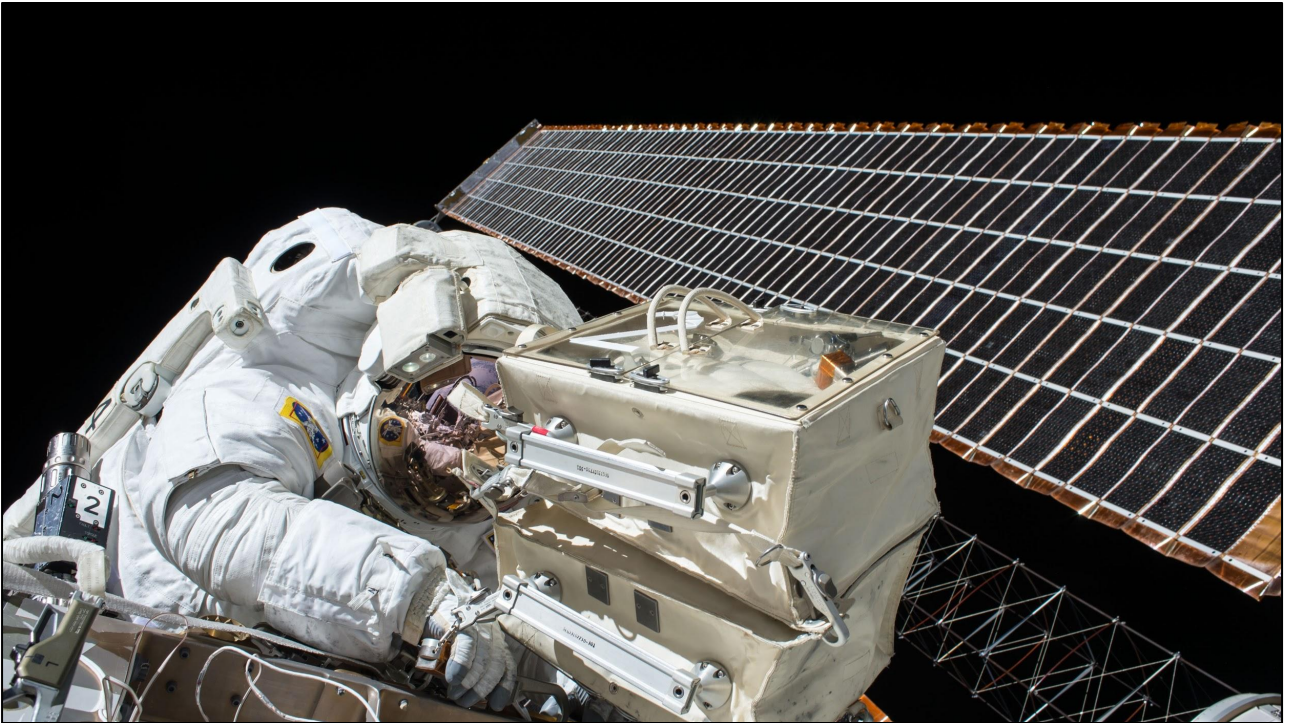


 @ChrisShort

devopsish.com

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

It groups containers that make up an application into logical units for easy management and discovery. Kubernetes builds upon 15 years of experience of running production workloads at Google, combined with best-of-breed ideas and practices from the community.



“A job creates one or more pods and ensures that a specified number of them successfully terminate.”

If you have a pod that needs to run until completion no matter what, a Kubernetes Job is for you.

Think of Jobs as a batch processor.

```
t.cshort@daffy: ~ (ssh)
root@daffy:/home/cshort# crontab -l
#Ansible: chrisshort.net Hijacking Check
*/5 * * * * /home/cshort/bin/dns_verify_chrisshort.net.sh > /dev/null 2>&1
#Ansible: cshort.co Hijacking Check
*/5 * * * * /home/cshort/bin/dns_verify_cshort.co.sh > /dev/null 2>&1
#Ansible: jules014.com Hijacking Check
*/5 * * * * /home/cshort/bin/dns_verify_jules014.com.sh > /dev/null 2>&1
#Ansible: Disk Utilization Check
*/15 * * * * /home/cshort/bin/graph_du.sh > /dev/null 2>&1
#Ansible: NTP Stratum Check
*/5 * * * * /home/cshort/bin/ntpcheck.sh > /dev/null 2>&1
#Ansible: Load Check
* * * * * /home/cshort/bin/graph_load.sh > /dev/null 2>&1
#Ansible: Free Memory Check
* * * * * /home/cshort/bin/graph_mem_free.sh > /dev/null 2>&1
#Ansible: Fever Feed Reader Refresh Cron Job
*/15 * * * * curl -k -L -s -m 840 --user-agent 'Fever Refresh Cron' 'https://fever.chrisshort.net/?refresh'
root@daffy:/home/cshort#
```

Show of hands...

Who runs cron jobs?

What happens when those cron jobs don't run?

Kubernetes Cron Jobs are a relatively new thing.

Since Cron Jobs build on top of the existing Job functionality, I know that the job will be run to completion.

Even if a pod is destroyed mid-job, it will spin up on another pod and run.

High-available cron jobs have been a beast I've tried to slay many times.

Use Case

Triggering Static Site Builds

I moved my newsletter, DevOps'ish, off of Medium and on to Netlify with Hugo as a static site generator.

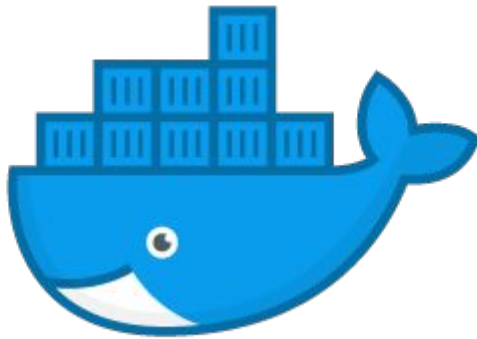
One piece of functionality lost in the move is the ability to schedule posts.

Netlify provides a build hook that will trigger builds when called.

I can write the newsletter and set it to a date in the future.

Hugo, by default, will not publish articles unless a build is completed after the specified date.

Calling the build hook URL via curl with a cron job is a way to implement scheduled posts with Hugo on Netlify.



docker.

We need a container

Lets build it and push it to Docker Cloud

```
docker build -t k8s-cron-job-demo
```

.

```
docker tag k8s-cron-job-demo  
chrisshort/k8s-cron-job-demo
```

```
docker push  
chrisshort/k8s-cron-job-demo
```



[@ChrisShort](#)

devopsish.com

Now to get kubernetes configured how we want it

We know we need a secret

```
minikube start
```

```
kubectl apply -f  
netlify-webhook-secret.yml
```

```
kubectl get secret netlify-webhook
```

```
kubectl apply -f netlify-cronjob.yml
```

```
kubectl get cronjob
```

```
kubectl get jobs --watch
```

Conclusion

Built a Docker container

Pushed the image to registry

Created secret to store the build hook

Created the CronJob

Imagine the possibilities of...

Having all cron jobs in version control

Storing secrets safely

Distributing the jobs across a cluster

Ensuring the jobs run to completion at every execution

References

Everything but the secret

[The Children's Illustrated Guide to Kubernetes](#)

[What is a Job? | Kubernetes](#)

[CronJob | Kubernetes](#)

[Secret | Kubernetes](#)

[Kubernetes: Cron Jobs](#)

[chris-short/k8s-cron-job-demo](#)
